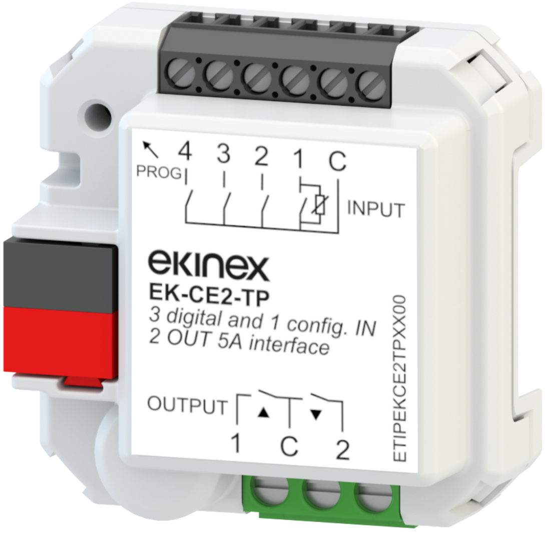
ekinex
CONTROL YOUR LIVING SPACE

APPLICATION MANUAL
EK-CE2-TP



Universal Interface
4 DIN / 1 NTC
2 OUT Relay 5A

Summary

Revision	Changes	Date
1.0.0	First draft	04/10/2017
1.1.0	Fixes for switching to Ekinex SpA	20/08/2018
1.2.0	Errors correction (dimming definition for DI)	21/10/2019
2.0	Update to version 2024 with WE output terminal 691213510003	11/03/2024

1 Scope of the document

The present Manual describes the application details for version 01.003 of the ekinex® KNX temperature / contacts universal interface EK-CE2-TP.

Document	Filename (## = revision)	Version	Device Rel.	Last update
Technical Datasheet	STEKCE2TP_IT.pdf	EK-CE2-TP	A2	03 / 2024
Application Manual	MAEKCE2TP_IT.pdf	EK-CE2-TP		
Application Program	APEKCE2TP ##. Knxprod	EK-CE2-TP		

2 Product description

The module EK-CE2-TP ekinex® can be programmed as [DI] to perform the following functions:

- on / off control of single and group utilities;
- dimming control;
- roller blind operation (rolling shutters, blinds, curtains, etc.);
- Signal contacts status detection (from safety devices, alarms, etc.);
- Scenarios store and recall;
- Values sending on the bus (temperature, brightness, etc.);
- switching in forced operation (blocking);
- pulse and drive counts.

The first input is also configurable as a temperature probe for sending values, or as a thermostat on / off type for the management of control valves for heating / cooling or fan coil.

The output channels of the device can be programmed as [DO].

As binary outputs, available operations are:

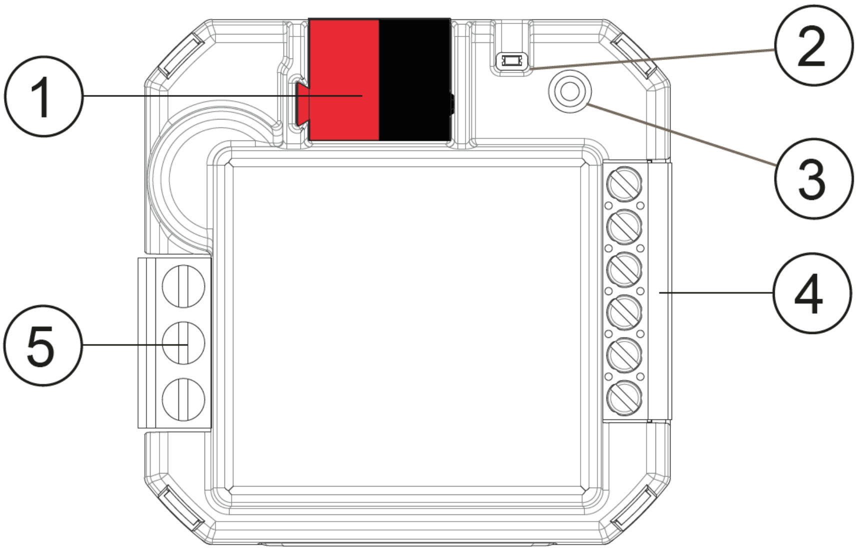
- On / Off control of single or group utilities;
- Configuration as a normally open or normally closed switch;
- Logic ports and forced operation for each channel;
- Block function for each channel;
- Timing: on / off delay, staircase light function with signaling of notice;
- Integration into scenarios;
- Energy/time counters.

As shutter actuator (coupled), the following functions are available:

- Full stroke of climb and descent
- Partial stroke with stop to 0 to 100% of stroke length
- Position setting (change during operation of the preset position)
- Adjustment of tilt slats (for venetian blinds)
- Insertion into scenarios
- Automatic control for direct sunlight protection
- Automatic control for protection against atmospheric conditions (rain, wind, frost).

3 Connection elements

The following elements are necessary for the device connection:



1. KNX bus line connection terminal
2. LED indication of the programming mode
3. Programming button
4. Terminal block for digital / analog inputs
5. Terminal block 2 for bistable relays (5A) outputs

4 Configuration

The device functionality is set via ETS software.

In order to configure the device, the development tool ETS4 (or later) and the ekinex® application program specific for the device are required.

The application program allows to access, within the environment ETS4 / 5, the configuration of all the device operating parameters. The program must be uploaded into ETS (alternatively you can download in a single operation the whole ekinex® products database), then any device of the same type can be added to the project under development.

The configurable parameters for the device will be described in detail in the next chapters.

The configuration is tipically fully defined in off-line mode; the upload of the configuration into the device will then take place in the programming phase, which will be described in the next paragraph.

Code product	N. of inputs	N. 5A relay outputs	Program ETS application (## = revision)	Communication objects (Nr. max)	THEGroup ndirizzi (Nr. max)
EK-CE2-TP	4	2	APEKCE2TP ## .knxprod	196	254

5 Programming and 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.

Resetting the device

In order to reset the device, please remove the bus network connection by pulling out the bus terminal from its seat. While keeping the programming button pressed, reinsert the bus terminal in its seat; immediately after, the programming LED should start blinking at fast rate.

Then, release the programming button and pull out the clamp again; the reset has been done.

After the reset operation, it is necessary to perform again the addressing and the configuration of the device via the ETS software.

Warning! The reset brings back the device to the factory configuration. The address and the parameter values previously configured will be lost!

6 Description of the features

The KNX bus performs also the power supply function; when this is connected, the device enters in a full activity state after a short period of time (a few tens of ms), which is necessary for re-initialization. It is possible to define an additional delay, to avoid traffic overload on the bus during the phase system start-up.

In case of voltage drop on the bus (that is, voltage lower than 19Vdc for 1s or more), the device switches off automatically; before the supply becomes insufficient, the state at the time of shutdown is stored internally. The timer functions are stopped and the device doesn't answer anymore to the associated group of addresses.

Once the voltage level is up again, the device resumes the operations by restoring the state stored at power off, except for those parameters for which a different value of power-on initialization has been configured.

6.1 Off-line operation

A non-programmed device does not operate in standby mode. Since the operation capability relies entirely on the information exchange through communication objects, the device can't operate at all independently from a KNX bus.

6.2 On-line operation

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

6.3 Software Operation

The activities performed by the software involving the button are the following ones:

- Detect button presses by the user and consequently generate the telegrams on the bus, depending on the device programming;
- Implement interlocking functions and coordination of inputs and timings;
- Manage the incoming telegrams, to update the communication objects and LED outputs status;
- React to telegrams on the bus, depending on the status request of the inputs or local variables.

The device status, and specifically of its interface elements (input activation status and LED indicators) is based on KNX communication objects. They can be defined by the application program and connected in different ways to the physical elements of the device; these communication objects act as state variables for the device.

Special events are also foreseen, to activate additional features (e.g. bus voltage drop or restore, uploading of a new configuration by ETS).

6.4 Input buttons

The input press may be associated with different effects on a state variable.

6.4.1 Events on buttons

The pressure of a button can be handled either as events of "on-off" type (where "on" refers to the pressure, off" to the release") or as events of "long / short pressure" type. The time period which discriminates between "long" and "short" can be defined.

In both cases, for each of the two available events, a different action that acts on a different variable can be assigned (and in some cases, even more than one; see in the next paragraphs for further details).

6.4.2 Lock Function

For each input (or channel, if inputs are coupled), a blocking function can be enabled seperately. This function disables the input operation via a telegram from the bus; in the same way, it can be de-activated through a telegram.

If in locked state, the input is effectively disabled.

It is possible to assign a communication object with a specific value, that matches each of the transitions of input or output to/from the block.

The locked state can also be automatically activated at the bus reset.

6.4.3 State variables (communication objects)

If a variable is modified by any input event, its data-type can be selected among those made available by the KNX standard for the communication objects; eg. a 1-bit value (on-off), a 2-bit value or an integer value of higher dimension.

In any case, each of the two events can:

- modify the variable value with one of two values, among those allowed for the selected data-type (the trivial case is 1-bit type);
- switch alternatively to the second value (see the previous case);
- do nothing (the value is unchanged).

When the state variable is assigned a group address, it becomes a KNX communication object; therefore, it takes over the usual properties of the communication objects, e.g., the possibility to be modified by other devices via a telegram, or the use of flags to determine how an object modification impacts on its transmission over the bus.

6.4.4 Events and Communication Objects link

The description above has been slightly simplified for calrification purposes; each event may be associated to several communication objects (up to a maximum of 8), even of different types. Each of these communication objects can have its specific behavior (in terms of accessibility to KNX) and its associated value.

6.4.5 Cyclic sending

For most functionality, it is possible to configure the telegram sending not only if the value associated to a state changes (typically as a result of the inputs transition), but also at regular intervals when the state is active.

This behavior, called “cyclic sending”, can be set separately for each of the two states associated with either an input or a channel.

If an input type is set to "enter values or sequences", the cyclic sending is available only if the input is associated to a single communication object.

6.4.6 Inputs coupling

The 4 inputs previously described can be considered as independent; given the physical structure of the device and the nature of the mostocommonly performed functions, the inputs can be associated in pairs. A pair of inputs will be briefly named as “channel”, since it is physically associated to a channel.

Since the inputs are numbered from 1 to 4, they can be identified as Input 1 / Input 2 for channel 1 and Input 2 / Input 4 for channel 2. For consistency, the same naming convention is used, regardless of the fact that some of the inputs are coupled together or not.

To specify whether an input is used in coupled mode or not, a specific option has been foreseen in the configuration panel: the input can be “independent or single”, or “coupled”. This selection is done at channel level, because only inputs that physically belong to the same channel can be coupled. Therefore, the only allowed coupling options are: Input 1 with Input 2, and Input 3 with Input 4.

If the “Independent or single” mode is selected, each of the inputs operates independently and has its own parameters and communication objects. This is the mode described so far.

If the “coupled” mode is selected, the two inputs are linked under the same channel for common functionality; accordingly, these inputs operate on shared communication objects.

Obviously it is possible to configure some of the inputs as independent and some others as coupled, with the link constraints described above.

In the next paragraphs, a description of the features associated to the inputs is reported. The “independent or single” and the “coupled” modes have similar functions, but configuration is different and for this reason they will be described separately.

6.4.7 Independent or single input

Each independent input can be configured for one of the following input types:

a. Send values or sequences

An event triggers the transmission over the bus of a configurable value, or sequences of values . The type for such values can be either logical or numeric, with different datatypes or sizes.

A sequence can be made by a maximum of 8 communication objects, each one of different type and value.

The values of the sequence can be separated by configurable delays.

b. Dimmer control

This input type is used in combination with KNX dimming actuators, for lighting devices control purposes.

The function is activated only with long / short 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.

c. Shutters or venetian blinds control

This function is used in combination with KNX dimming actuators, to control blinds or motorized dampers. These actuators are able to to open and close the connected shutters; it is possible to select two types of movements, continuous or discrete.

Depending on the input events, the device sends the actuator appropriate telegrams. The configuration parameters are as follows:

- 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.

d. Scene function output

Scene mode is used in combination with KNX units that support the scene function (e.g., dimmers).

The feature allows you to store and retrieve a communication object for scenario setting; in particular, the device sends a scenario "store" or "invoke” command to the actuators, as a result of a short / long press event.

Available configuration options are as follows:

- Activate pre-set scene on short press, and store current setting as scene value on long press
- Activate two different scenes on long and short press.

e. Counter

This input type allows the cyclical sending of a communication object, with data-type configurable as 1, 2, or 4 bytes. It is also possible to set a maximum value.

6.4.8 Coupled input mode

Each pair of inputs, that correspond to the two sides of the same channel, can be configured to run one of the following features (only the differences with respect to the independent mode description are mentioned).

a. Switch control

The two coupled inputs are connected to the same communication object; unlike the independent or single mode, however, the object can only be of 1 bit type (on-off), i.e. a conventional switching behaviour.

The user can select which input is associated to the action of "turn on" or "off".

b. 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.

c. Shutters or venetian blinds

The two inputs of the channel are assigned to opposite movement directions; in particular, input 1 or 3 can open / move up the shutter, while input 2 or 4 can close / move down, or vice versa.

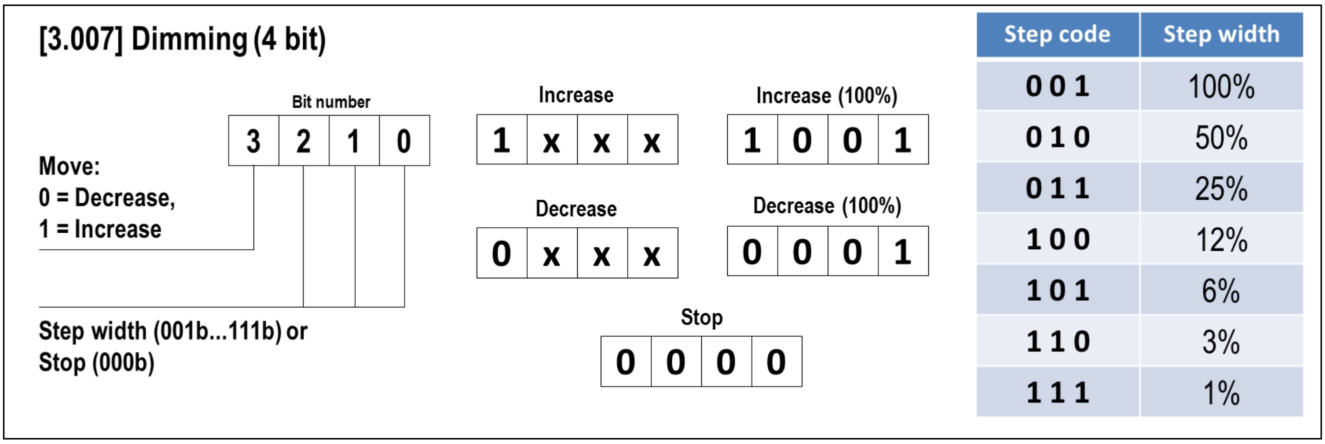
The venetian blind mode can be set to work in the same way as independent or single inputs.

In coupled mode, there is no provision for a *scene* control feature.

6.4.9 Dimming function

The "dimmer" feature is an application profile, described in the KNX specifications. These specifications define the basic requirements related to the interface control; however, some aspects of the operating modes, that are device specific (eithet a control device or an actuator) must be taken into account.

The control for a "dimmer" type is based essentially on a 4-bit communication object, whose data have the format shown in Figure 2:



Step code	Step width
0 0 1	100%
0 1 0	50%
0 1 1	25%
1 0 0	12%
1 0 1	6%
1 1 0	3%
1 1 1	1%

Figure 1 - "dimmer" type control

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

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

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

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

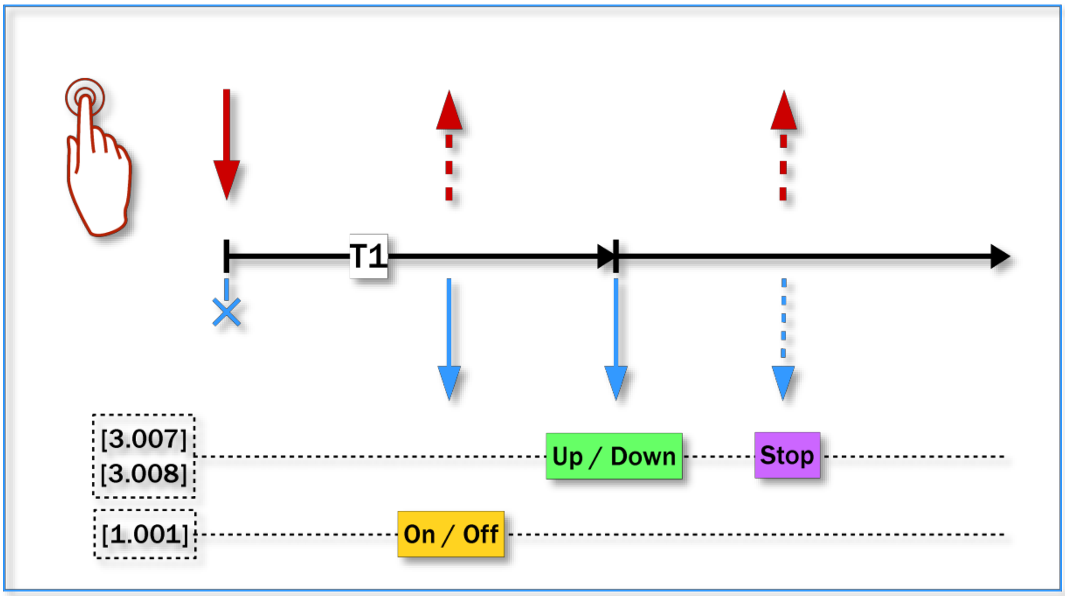


Figure 2 – "dimmer" commands sequence

The defined operations and the associated commands are as follows:

- Short press: instantaneous switch on / off (toggle on / off on a switch object)
- Long press: Increase / decrease value up to 100% / down to 0%
- Release: Stop Up / Down.

The same logic can be applied for the control of roller shutters or slats for venetian blinds (where "maximum / minimum intensity" should be replaced by "open / close"). For this purpose, a dedicated datapoint (DPT) 3,008 is foreseen, which has identical structure and values compared with the above described ones.

In order to control a roller shutter in the same way described above, it is therefore possible to connect a 3.007 type (on the command side) communication object, with a 3,008 type (on the actuator side) communication object, if this is available; in this case, the "On / Off" type object, which allows the instantaneous on / off switching, is not used.

6.4.10 Shutter / venetian blind function

The "shutter / venetian blind" fuction is a set of application profiles for devices, covered by the KNX specifications. As in the case of the dimmer function, these specifications define the basic requirements related to the interface logic, in addition, the features related to the specific operating mode of the device (control device or actuator) must be considered.

For roller shutters, the actuator moves a mechanical component from a end limit position to another point in a gradual manner, with the possibility to stop at intermediate points; the command is sent out through two lines: when activated (only one at a time), they move the actuator in the corresponding direction.

A venetian blind is basically 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 roller shutter or a venetian blind is essentially based on 3 communication objects (each of 1-bit data-type):

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

The commands associated with these objects behave as follows:

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

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

In the simplest version, on command side:

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

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

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

In case of Ekinex input devices, two modes are available – indicated as "Shutter" and "Venetian blind" based on their typical destination – which are illustrated in Figure 4.

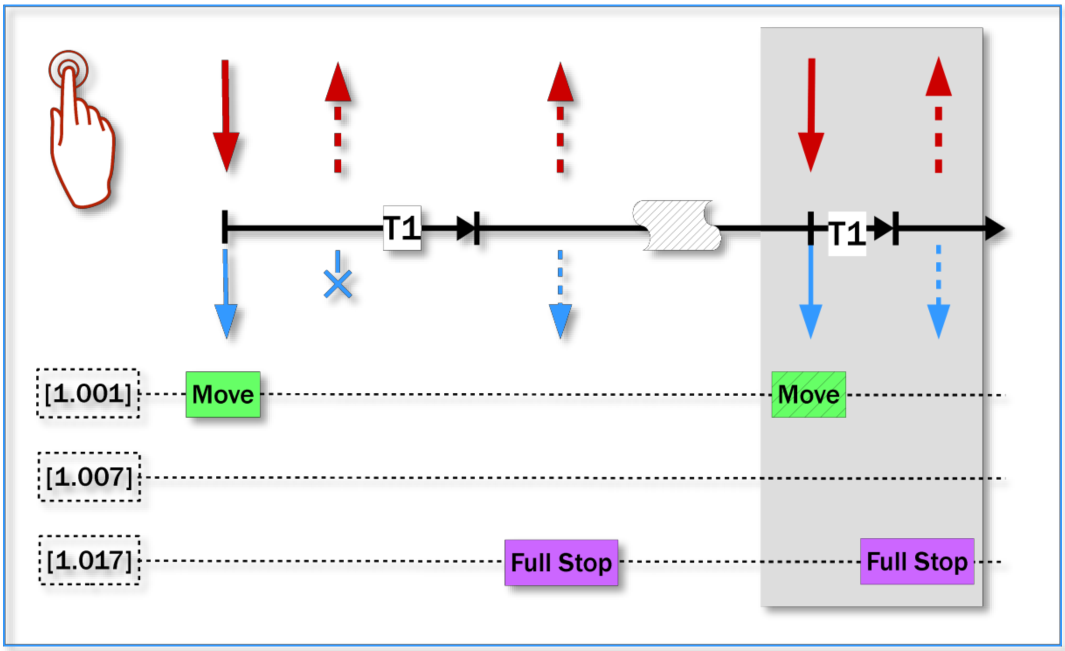


Figure 3 - sequence of commands in a "shutter"

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

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

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

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

The sequence is illustrated in Figure 5.

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

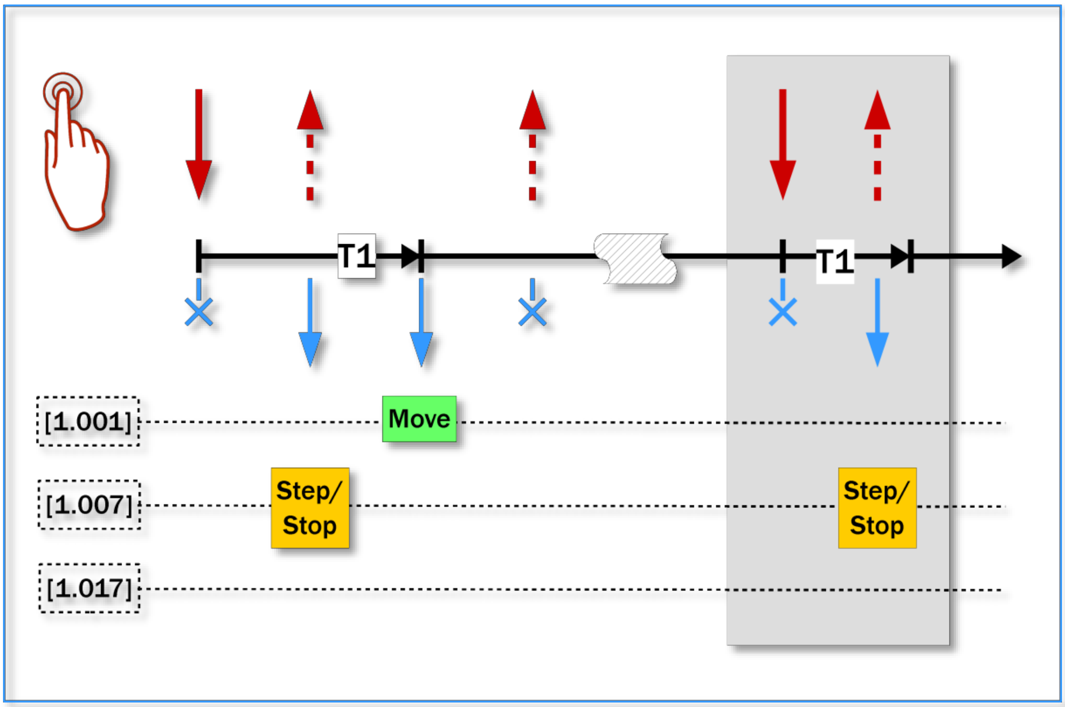


Figure 4 - sequence of commands in a "Venetian"

7 Logical 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 6:

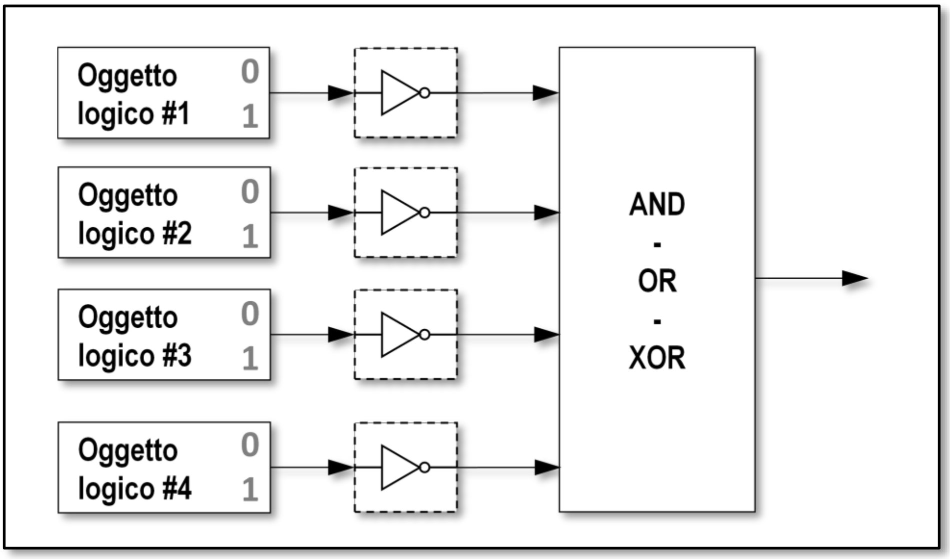


Figure 5 - Combination Logic function

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 following figures show the basic logic functions, assuming 2 inputs and only one logic communication object:

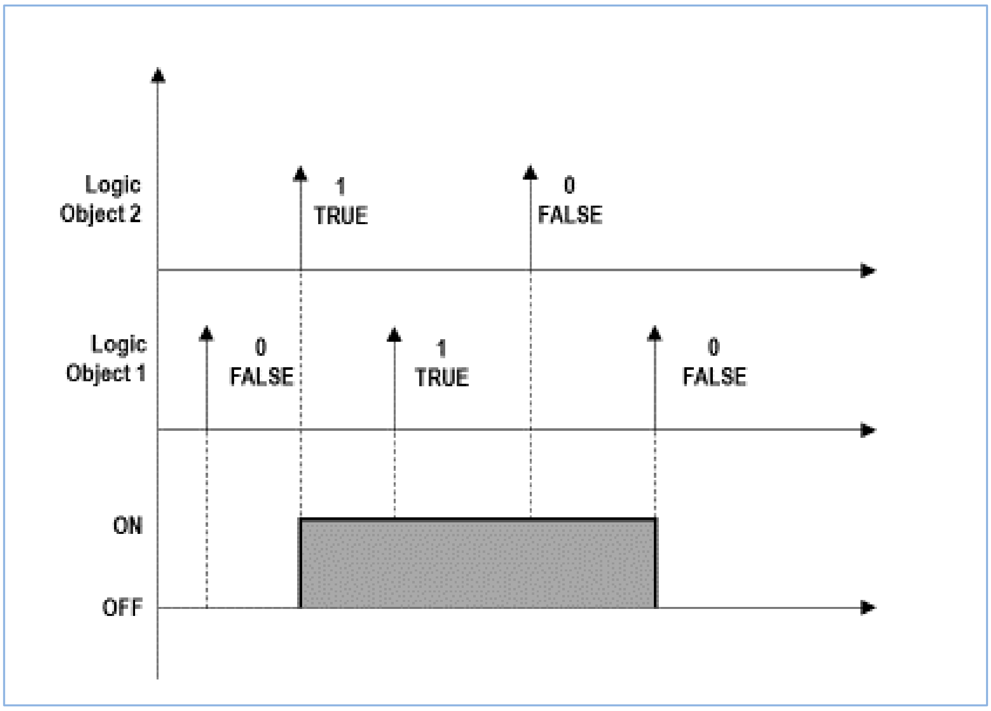


Figure 6 - logic function OR

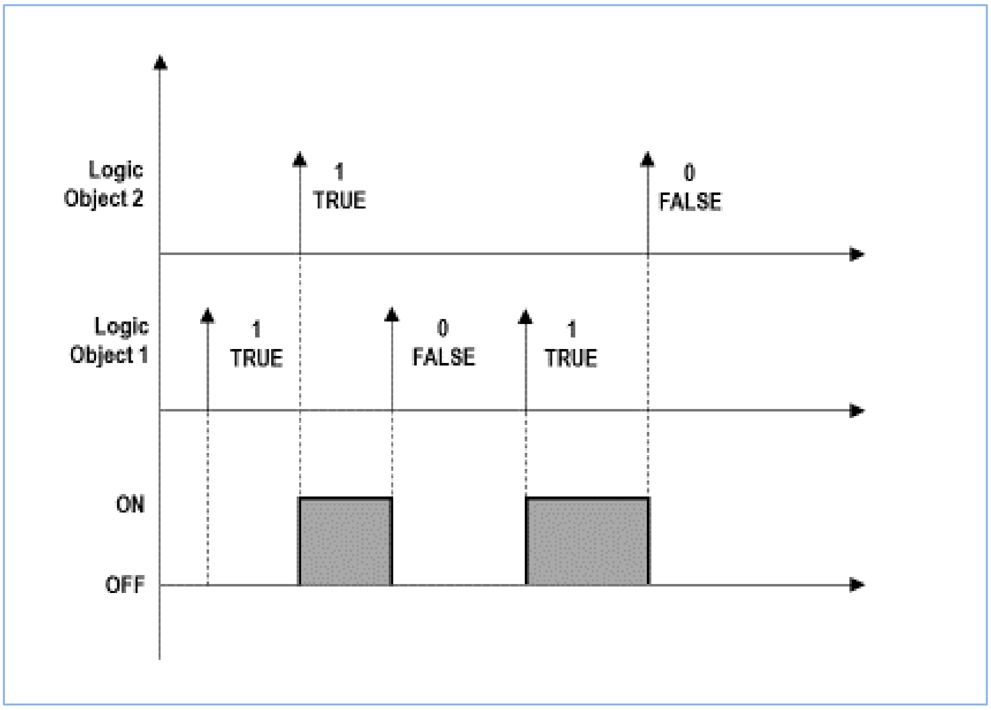


Figure 7 - Logic AND function

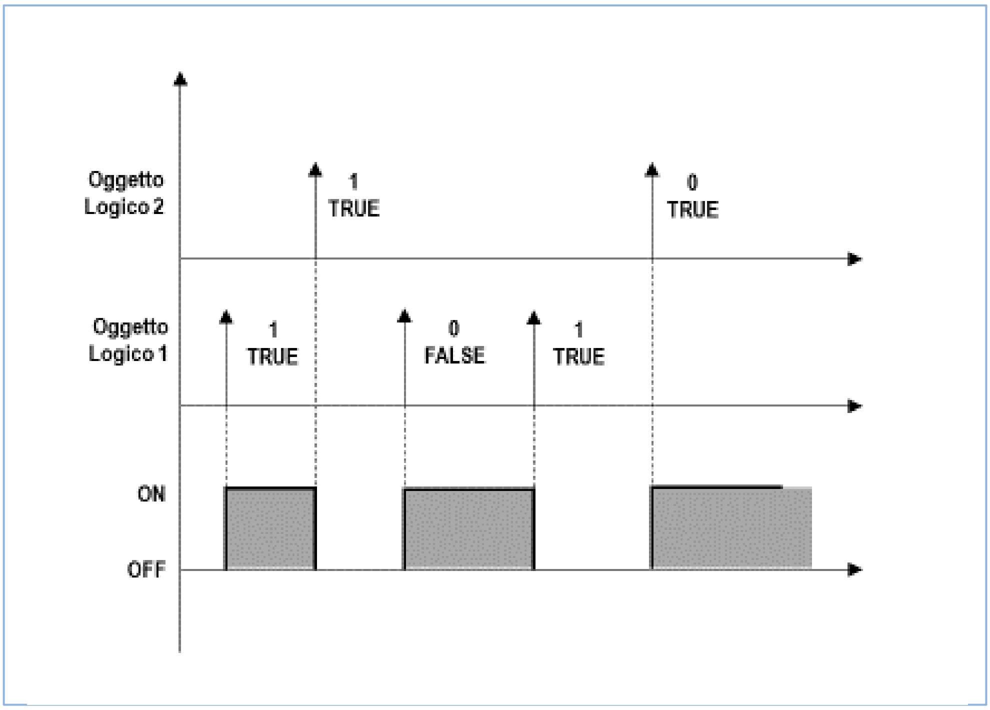


Figure 8 - Logic function Exclusive OR (XOR)

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.

8 Use as a regulator

The control of indoor temperature in home environment is implement by opening and closing of the shut-on/off valves in the distribution system, with ON / OFF control algorithm or PWM. For valves control, electrothermal actuators and / or servo motors can be used.

8.1 Control Algorithms

Figure 9 shows the components of a generic control system for the ambient temperature. The temperature controller (thermostat) detects the current value of the ambient air mass temperature (Teff) and compares it with the desired temperature value or setpoint (Tset).

below shows the components of a common generic control system for ambient temperature. The room temperature controller (thermostat) measures the current value of the air mass temperature (Teff) and compares it to the setpoint value (Tset)[1].

[1] Throughout this manual, the terms “desired value” or “setpoint value” are used interchangeably.

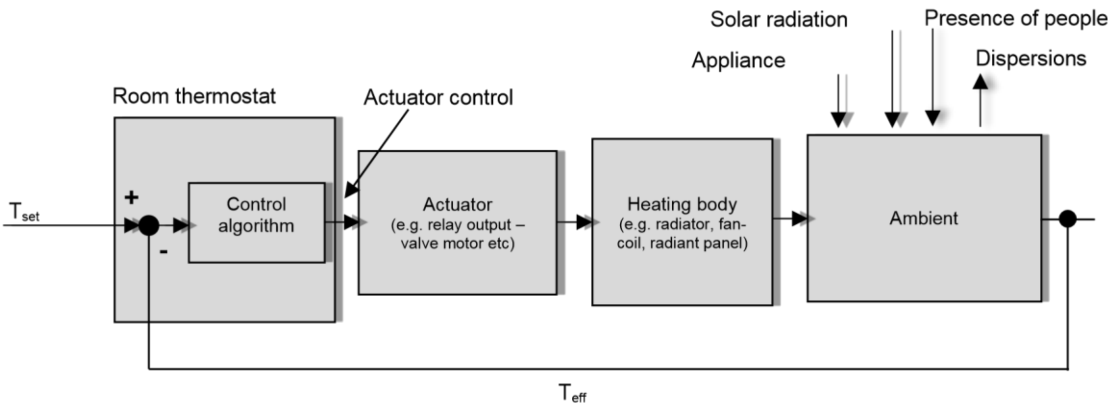


Figure 9 - Generic control system for the ambient temperature

The control algorithm, basing on the difference between the setpoint and measured temperature values, processes a command value which can be of analog or On / Off type; the command is represented by a CO that is transmitted via bus, either periodically or event based, to a KNX actuator device.

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

The device offers the following temperature control algorithms:

- Two-point control with hysteresis
- Continuous P-I controller (with analogue output)
- Continuous P-I controller with PWM output

These algorithms will be detailed in the following sections.

8.2 Two-point control with hysteresis

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

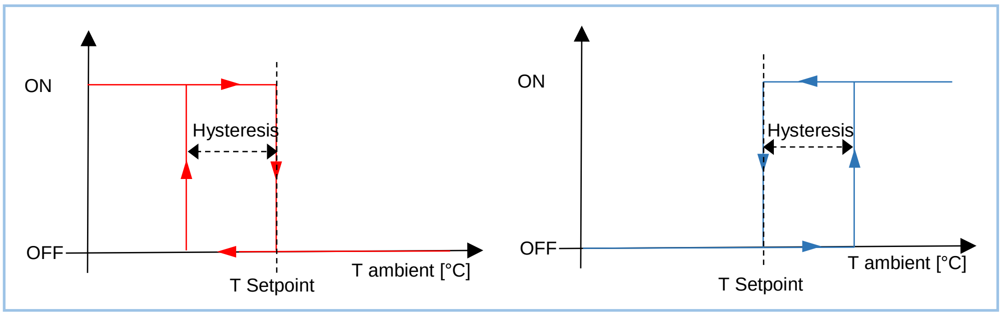


Figure 10 - Temperature control

Heating mode: when the measured temperature is lower than the value of the difference (Tset - ΔTHysteresis), where ΔTHysteresis identifies the differential adjustment of the boilers, the thermostat 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 value Tset, the thermostat turns off the heating system by sending a new message. In this way, there are two decision thresholds for activation and deactivation of the heating, the first being the level (Tset - ΔTHysteresis), below which the device activates the system, whereas the second is the desired temperature above which which the heating system is deactivated.

Cooling mode: when the measured temperature is higher than the value of the difference (Tset + ΔTHysteresis), where ΔTHysteresis identifies the cooling control differential, the thermostat activates the air conditioning system by sending its telegram to the actuator that handles the terminal; when the measured temperature falls below the value Tset, the device turns off the air conditioning system by sending another telegram. The mechanism uses two decision thresholds for the activation and deactivation of the cooling system: the first is made by (Tset + ΔTHysteresis), above which the thermostat activates the system, the second is made by Tset, below which the thermostat deactivates the system.

For applications where floor or ceiling radiant panels are adopted, it is possible to implement a zone temperature control with 2 different points. This type of control has to be combined with a control system (for the water flow temperature) that takes into account the internal conditions or with an optimizer that exploits the thermal capacity of the building, in order to defer the energy contributions.

With this type of control, the hysteresis (ΔTHysteresis) or the room temperature limit (Tset + ΔTHysteresis) represent the level of deviation from the desired condition that the user is willing to accept during the plant operation.

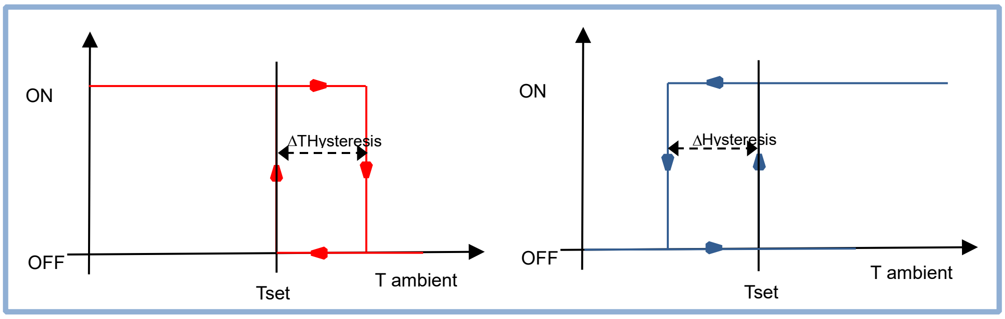


Figure 11 – Two-points temperature control

Heating mode: when the measured temperature is lower than the value of Tset, the thermostat 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 value (Tset + ΔTHysteresis), where ΔTHysteresis identifies the differential adjustment of the boilers, the thermostat turns off the heating system by sending a new message. In this way, there are two decision thresholds for activation and deactivation of the heating, the first being the value Tset, below which the device activates the system, whereas the second is (Tset + ΔTHysteresis), above which which the heating system is deactivated.

Cooling mode: when the measured temperature is higher than the value of Tset, the thermostat activates the air conditioning system by sending its telegram to the actuator that handles the terminal; when the measured temperature equals the value (Tset – ΔTHysteresis), where ΔTHysteresis identifies the cooling control differential, the device turns off the air conditioning system by sending another telegram. The mechanism uses two decision thresholds for the activation and deactivation of the cooling system: the first is made by Tset, above which the thermostat activates the system, the second is made by (Tset – ΔTHysteresis), below which the thermostat deactivates the system.

In the application program ETS, the Two-points control algorithm with hysteresis is proposed by default; it configures the lower hysteresis for heating and the upper hysteresis for cooling. In applications with radiant systems it is possible to select the position of the hysteresis, according to the second described mode, i.e. with the upper hysteresis for heating and the lower one for cooling. The hysteresis values for heating and cooling are differentiated: for the identification of the correct values is necessary to consider the characteristic inertia of the system.

The desired temperature (Tset) it is generally different for each of the four operating modes and the two heating/cooling seasonal modes. The values are defined for the first time during configuration with ETS and can be modified later. To optimize the energy-saving (for each degree above ambient temperature, the dispersions towards the outside and energy consumption increases by approximately 6%), it is possible to easily take advantage of the multifunctionality of the domotic system, for example with:

- hourly programming with automatic switching of the operating mode, performed by a KNX device with supervisor function;
- automatic-switched operating mode, to open windows for air refresh;
- stop the circulator as soon as the thermostats reach the setpoint;
- reduction of the flow temperature in partial load conditions.

8.3 PWM Control

The proportional-integral PWM (Pulse Width Modulator) controller uses an analog 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 picture below, by varying the ratio between the ON time and the OFF time, the average time of activation of the output varies, and consequently the average intake of heating or cooling power supplied to the environment.

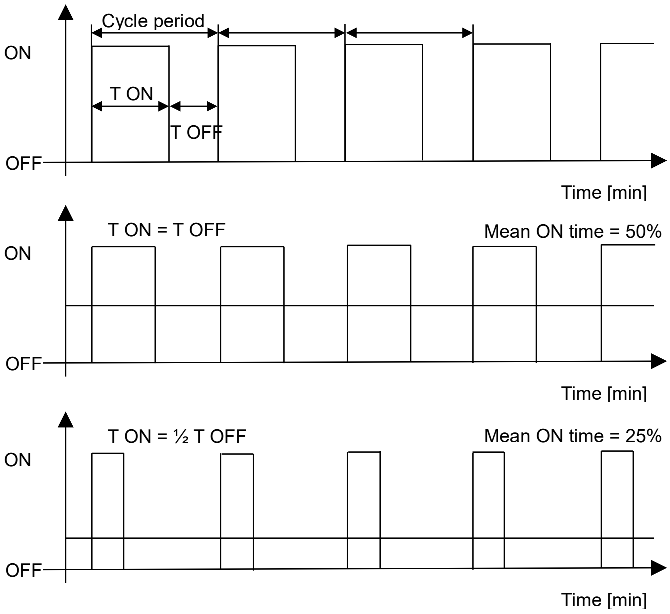


Figure 12 - PWM control scheme

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

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

Terminal type	Proportional band [K]	Integral time [min]	PWM cycle time[min]
Fancoils	4	90	15-20
Floor radiant panels	5	240	15-20
Ceiling radiant panels	5	100	15-20

i	The following guidelines are useful for the choice of parameters of a proportional-integral PWM type regulator. **Cycle time**: The cycle time should be chosen at least one order of magnitude (10 times) higher than the approximate characteristic time constants of the ambient; for systems with low inertia, such as air convection heating and cooling, short periods (10-15 min or lower, compatibly with the capabilities and performance of the heating / cooling system) should be chosen to avoid wide fluctuations in temperature. **Proportional band**: a narrow P.B. yields quick settling time to setpoint, but broad and continuous fluctuations in ambient temperature**. A** wide P.B, conversely, yields small to practically no oscillations of ambient temperature, but a slightly larger settling time. **Integral time**: a short I.T. yields short settling time but continuous oscillations around the setpoint; a long I.T. yields longer settling time but no oscillations.

8.4 Setpoint management

The unit does not have a local interface for integrated room temperature controllers: any changes of temperature setpoint values must therefore be done by means of another KNX device, specifically configured (with supervisor function) and transferred to the device by means of communication objects.

Three types of setpoint values management are foreseen:

- Single setpoint;
- Relative setpoint;
- Absolute setpoint.

8.4.1 Single setpoint mode

In this mode, a single communication object (input Setpoint) is used to modify the desired temperature. This object can be updated cyclically or on an variation event, by the supervisor device. In case of power failure the last value is saved in the non-volatile memory of the controller. In case the object is not updated, the temperature controller operates anyway on the default setpoints (for heating and cooling) that are set in the application program during the commissioning phase.

i	In the case the temperature control is configured in both heating and cooling modes, it is necessary for the supervisor device to update also the conduction mode input object (Heating / cooling state, [1.100] DPT_Heat_Cool). This is necessary to switch the action of the regulator consistently.

In case the window contacts are used to activate the energy-saving function, once the open window state is detected, the input setpoint is suspended and the building protection Setpoint previously set is activated (the related communication object is exposed and differentiated between heating and cooling down).

8.4.2 Relative setpoint mode

In this mode 4 communication objects for each of the operation modes:

- comfort setpoint;
- standby offset;
- economy offset;

- building protection setpoint.

The standby and economy setpoints are represented as attenuations with respect to the comfort setpoint, in order to facilitate the management by the supervisor: just by modifying the comfort setpoint value, all the references are automatically translated to the attenuated modes. The values changed from the bus are kept in the non-volatile memory of the controller.

With relative setpoint mode, the supervisor device may introduce a programmed time slots by sending the controller the current operating mode (communication object HVAC mode in [2.102] DPT_HVACMode). The default value for the HVAC mode object corresponds to the recall of the comfort setpoint.

Like the single setpoint mode, in case of a temperature control both in heating and in cooling mode with switching configuration from bus, it is necessary that the supervisor device also updates the conduction mode input object (Heating / cooling state, [1.100] DPT_Heat_Cool), to switch the action of the regulator consistently.

8.4.3 Absolute setpoint mode

In this mode, three communication objects are available, for each of the plant operation modes:

- comfort setpoint;
- standby offset;
- economy offset;
- building protection setpoint.

All the setpoints are represented as absolute values: by setting these values from the bus via communication objects, it is necessary to maintain consistency between the values of the attenuated operational modes.

With this mode, the supervisor device may introduce programmed time slots by sending the device the current operating mode (communication object HVAC mode in - [20.102] DPT_HVACMode). The default value for the object *HVAC mode in* corresponds to the recall of the comfort setpoint.

Like the single setpoint mode, in case of a temperature control both in heating and in cooling mode with switching configuration from bus, it is necessary that the supervisor device also updates the conduction mode input object (Heating / cooling state, [1.100] DPT_Heat_Cool), to switch the action of the regulator consistently.

8.5 Operating Modes

In single setpoint mode, two levels for each of the conduction modes are available:

- temperature setpoint;
- building protection setpoint.

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

Both in relative and absolute setpoint modes, 4 different operating modes (that are mutually exclusive) are available:

- comfort;
- standby;
- economy;
- building protection.

Each of these operating modes identifies a different temperature setpoint (for heating; there is a separate setpoint set for cooling), which can be assigned through the ETS program.

Each of these setpoints is accessible through communication objects. The setpoint and attenuation changes can be carried out remotely via the communication objects. The only exception is represented by the building protection setpoint, which affects the safe operation in the protection of system components (particularly in heating mode). This parameter must therefore be configured in advance through ETS or can be changed, if necessary, by another supervisor device via the bus.

8.6 Seasonal mode switching (Heating / cooling)

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

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

8.6.1 Switch over from KNX bus

Mode 1 envisages that the switching command comes from the KNX bus, i.e. from another “master” KNX device, such as the control and display unit ekinex® Touch & See or the room thermostat ekinex® EK-EP2-TP. The temperature controller integrated into the device behaves as a "slave": the switch over is carried out by the input communication object [DPT 1.100 heat / cool].

8.6.2 Automatic switch over, based on the room temperature

This mode is only suitable in applications for heating / cooling hydraulic systems with a 4-pipe configuration. Even in this case, the information can be sent on the bus with the output communication object [DPT 1,100 heat / cool]; the difference with mode 1 is that the switch over is performed automatically by the device, basing on the values of current temperature and setpoint. The automatic switch over is achieved by introducing a dead band, as shown in Figure 13**.**

.

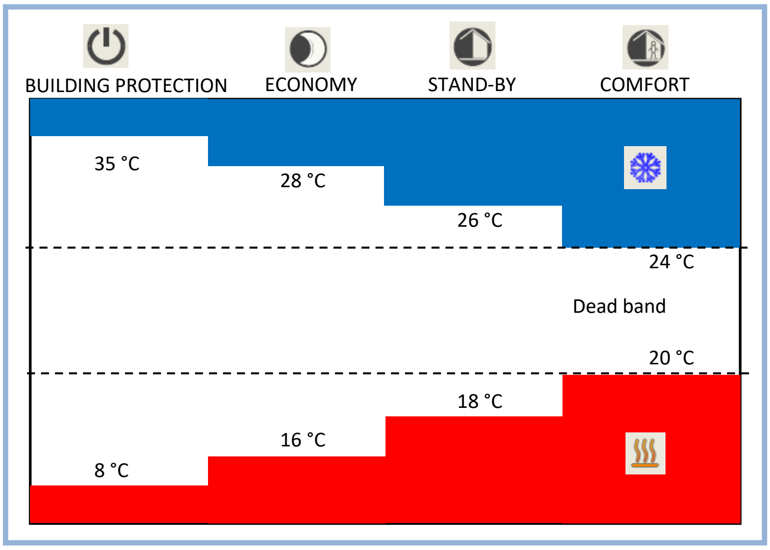


Figure 13 - Automatic switch over, based on the room temperature

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.

8.7 Alarms temperature control

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*.

9 Inputs from bus

9.1 General and timeout

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

9.2 Probes environment (input) and weighted average (ext. Obj.)

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

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

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

9.3 Surface temperature limitation function (ext. Obj.)

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:

- TSmax ≤ 29°C for zones of normal occupancy;
- TSmax ≤ 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 NTCtype sensor which is in contact with the floor surface.

The surface temperature limitation may be realized for several purposes:

- physiological compatibility (correct temperature at the height of the legs);
- when the system is used as auxiliary stage for heating. In this case, the heat losses to the exterior of the building are handled by the main heating stage, while the auxiliary stage only works to keep the floor temperature at a comfortable level (for example in bathrooms of residential buildings, sports centers , spas and thermal baths, etc.);
- protection against damages of the final coating due to an accidental overheating. Note that the warm water radiant panels are usually already equipped with a safety thermostat (with intervention on the hydraulic mixing group), while in the case of electrical power this device is not usable and it is common practice to realize a temperature limitation with a surface temperature sensor connected to the device.

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

9.4 Anticondensation protection function (ext. Obj.)

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

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

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

9.5 Window contacts (ext. Obj.)

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

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

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

9.6 Presence Sensor (ext. Obj.)

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 if the room temperature controller is enabled; these objects are then synchronized to the situations detected by the sensors.

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

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

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

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

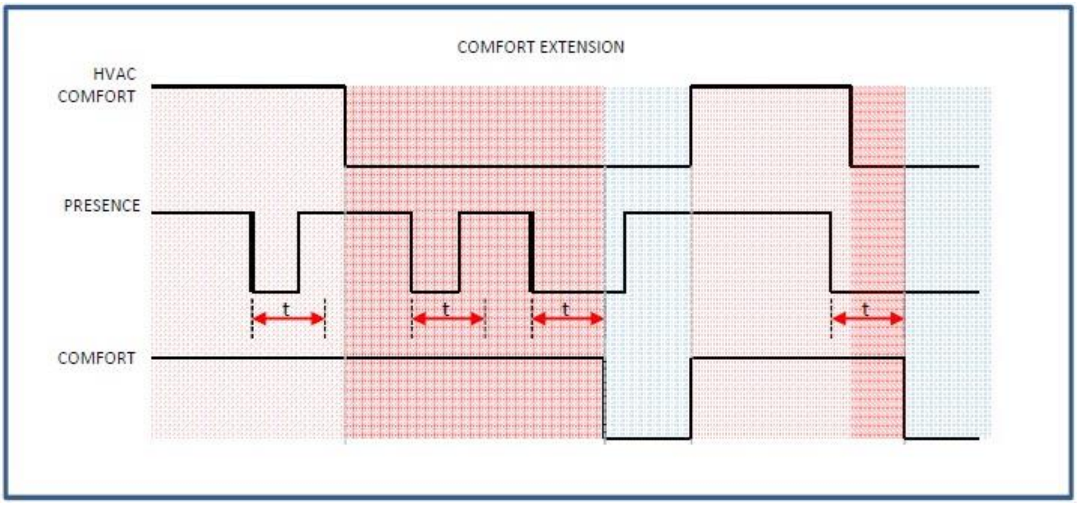


Figure 14 - Diagram for the comfort extension function

Figure 14 shows that, even if a presence is detected while the operating mode forced by the time scheduling function is not comfort, no change of operating mode is performed until the next programmed comfort event.

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

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

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

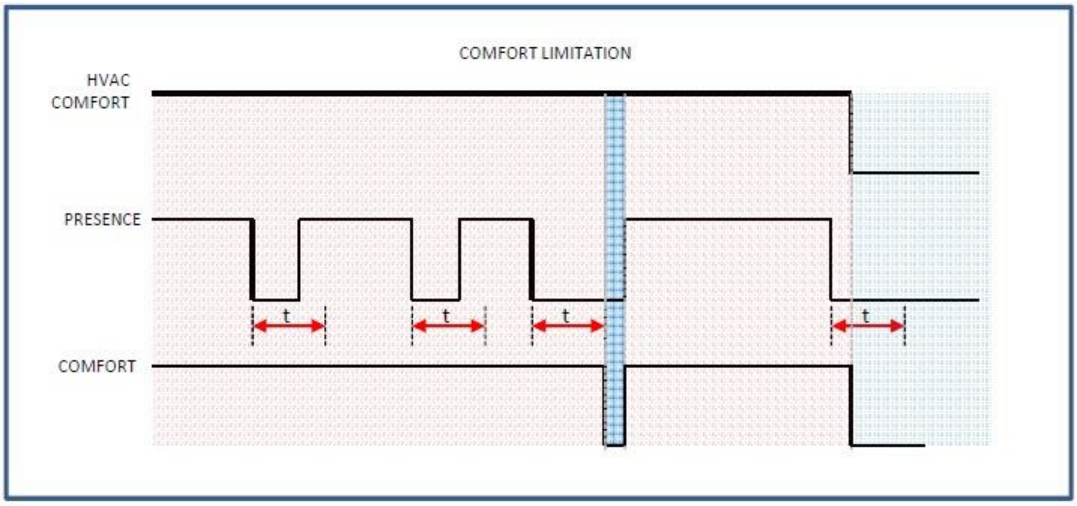


Figure 15 - Diagram for the comfort limitation function

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.

10 Application program for ETS

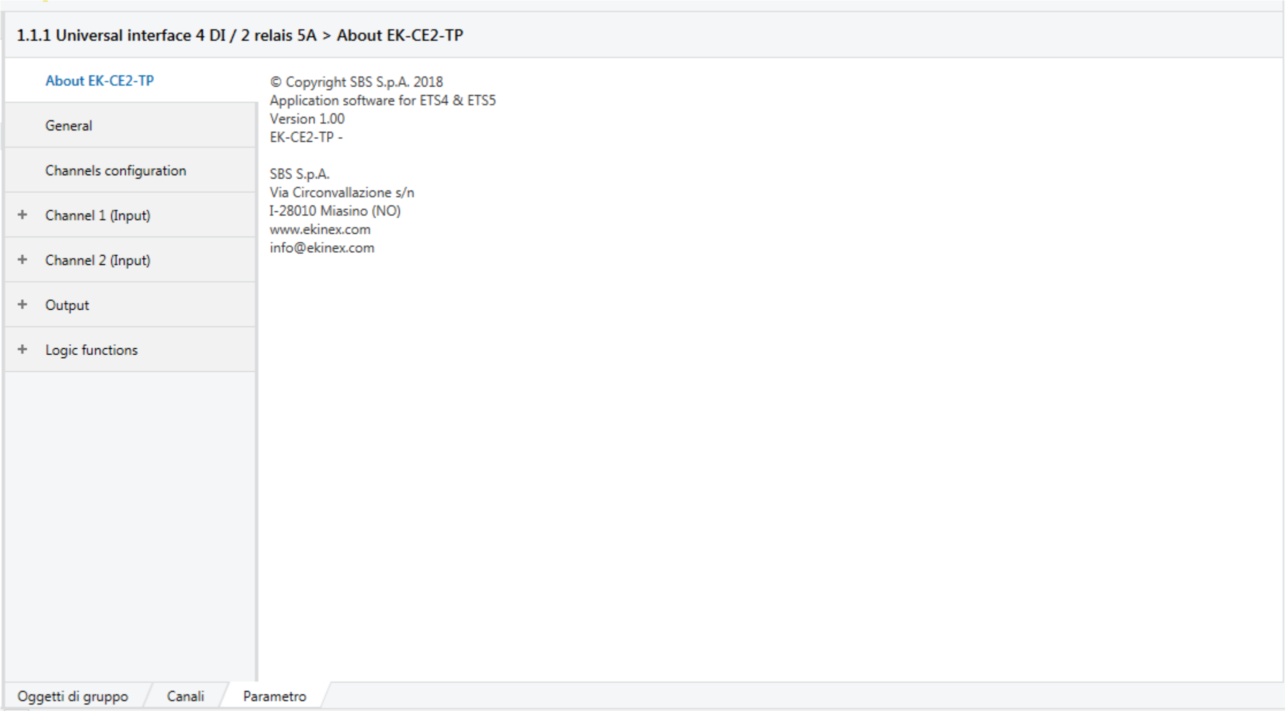
This section of the manual lists all the configurable parameters and simultaneously describes the relevant communication objects.

Every channel, and every input (or coupled inputs of the same channel) offers the same parameters and the same types of communication objects, but they may all be independently configured.

10.1 General aspects

In the next paragraphs, all settings are grouped by channel or input (depending on how applicable): a generic channel number is referenced as “x” (where x = 1,2), while a generic input is referenced as “xx” (xx = 1,2,3,4).

i	The values of the parameters highlighted in bold are the default ones.



© Copyright SBS S.p.A. 2018
Application software for ETS4 & ETS5
Version 1.00
EK-CE2-TP -

SBS S.p.A.
Via Circonvallazione s/n
I-28010 Miasino (NO)
www.ekinex.com
info@ekinex.com

10.1.1 About EK-CE2-TP

The folder named ***About EK-CE2-TP*** is for information purposes only and does not contain parameters to be set.

The information provided is:

Copyright © 2018 Ekinex SpA
Application software for ETS4 / 5
Version 1.00 (or later)
EK-CE2-TP

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

10.1.2 General parameters

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	Values
Delay after bus voltage recovery	-	hh:mm: ss.fff **(00: 00: 04:000)**

	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.	
Debounce time	-	hh:mm: ss.fff **(00: 00: 00:050)**
	Time frame between the first level switch for a signal, and the instant where it stays stable, without further ambiguity.	
Technical alarm		**disabled** / enabled
	Enables any technical alarm and the related sending over the bus.	

10.1.3 Channels configuration

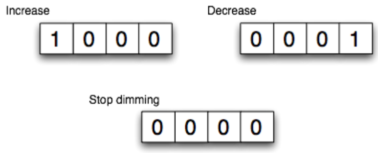
Parameter name	Conditions	Values
Channel x (Input)	-	disabled **independent or single** coupled
	Sets the operating mode for inputs associated with Channel x. *The channel and physical inputs identification is done trhough a specific number (1, 2, 3, 4).*	
Input 1	Channel 1 = independent or single	Disabled / **enabled DIN** / enabled NTC
	Enables or disables the possibility to generate events for Input 1.	
Input xx	Channel x = independent or single (x = 1,2)	**disabled** enabled
	Enables or disables the possibility to generate events for Input xx (xx = 2, 3, 4).	
Type	Channel x = independent or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4)	**sending values or sequences** dimming shutter or venetian blind scene counter
	Set the functionality associated with input xx of the selected channel. *Additional parameters for the selected function are listed in the individual configuration of each input (see following paragraphs).*	
Output	-	**disabled** 2 binary outputs valve / venetian blind / shutter
	Set the functionality for output. *Additional parameters for the selected function appear in the individual configuration of output (see following paragraphs).*	
Output 1A	Output = 2 binary outputs	**Disabled** enabled

Parameter name	Conditions	Values
	Set the functionality associated with Output 1A, when 2 binary output configuration is selected.	
Output 1B	Output = 2 binary outputs	disabled enabled **copy parameters from function 1A**
	Enables or disables the ability to generate events for the second Output. *If it is not disabled, the second output can be assigned its own independent function, or perform the same type of function as Output 1A.*	
Use	Output = valve / venetian blind / shutter	valve shutter **venetian blind**
	Set the functionality associated with the second Output. *Additional parameters for the selected function appear in the individual configuration of each input (see following paragraphs).*	

10.1.4 Independent or single: send values or sequences

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Channel x – Switching status [type], object n	Channel x = independent or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4)	according to the configuration (1-bit)	C-WTU	according to the configuration **([1.001] switch)**	3, 20 (1, 2) 37, 54 (3, 4)
	Up to 8 objects can be defined and associated to the same event. *The listed COs refer to the first of these objects 8 (for each of the inputs); the CO after the first one are sequential. To obtain the CO number for the n-th object, simply add (n-1) to the numbers given.* *Example: the COs associated to input 3 have numbers starting at 37. The number of the 5th CO associated with that input will then be 37 + (5-1) = 41.* *Types and dimensions of the individual objects can be configured as described in the next paragraphs.*				

10.1.5 Independent or single: dimming

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx - Switching command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) type = dimming	1 bit	C-WTU	[1.001] switch	11, 28 (1, 2) 45, 62 (3, 4)
	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 xx – Dimming up / down / stop command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) type = dimming	4 bit	C--T-	[3.*] 3-bit control	12, 29 (1, 2) 46, 63 (3, 4)
	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 1 0 0 0 Decrease 0 0 0 1 Stop dimming 0 0 0 0 *Increase/decrease values are sent when a long press action occurs and stop value on press release.* *The value sent can be a fixed value or it can be toggled at each input activation.*				

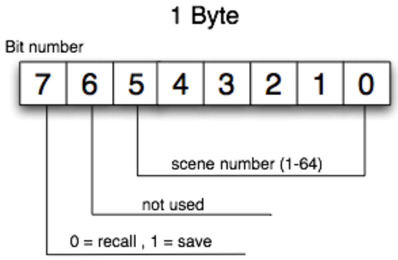
10.1.6 Independent or single: shutter or venetian blind

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx – dedicated stop command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) type = shutter or venetian blind	1 bit	C- -T-	[1.017] trigger	11, 28 (1, 2) 45, 62 (3, 4)
	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 - Stop – step up/down command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = shutter or venetian blind venetian blind mode = enabled	1 bit	C--T-	[1.007] step	14, 31 (1, 2) 48, 65 (3, 4)
	Move the blind to fully open or fully closed position. The object is sent at the end of a long press.				
Input xx - Move up / down command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = shutter or venetian blind venetian blind mode = enabled	1 bit	C--T-	[1.008] up / down	15, 32 (1, 2) 49, 66 (3, 4)
	Increase or decrease the opening of the blind stepwise. *The object is sent on a short press.*				

10.1.7 Independent or single: scene

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx - Scene number	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = scene	1 Byte	C--T-	[17.*] Scene number [18.*] Scene control	16, 33 (1, 2) 50, 67 (3, 4)

Store or recall a scene. The lowest 6 bits in the byte form the code of the scene, while the highest bit is the operation code (store or recall).



10.1.8 Coupled: switch

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx - Switching command	Channel x = coupled Type = switching	1-bit	C-WTU	[1.001] switch	11 45

10.1.9 Coupled: dimming

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx - Switching command	Channel x = coupled Type = dimming	1-bit	C-WTU	[1.001] switch	11 45
Input xx - Dimming up / down / stop command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = dimming	4 bits	CR-T-	[3.*] 3-bit control	12 46
	See notes for independent or single input.				

10.1.10 Coupled: shutter or venetian blind

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx - Dedicated stop command	Channel x = coupled Type = shutter or venetian blind Venetian blind mode = disabled	1 bit	C- -T-	[1.017] trigger	11 45
	See notes for independent or single input.				
Input xx - Stop – step up/down command	Channel x = coupled Type = shutter or venetian blind Venetian blind mode = enabled	1 bit	C- -T-	[1.007] step	14 48
	See notes for independent or single input.				
Input xx - Move up / down command	Channel x = coupled Type = shutter or venetian blind Venetian blind mode = enabled or disabled	1 bit	C- -T-	[1008] up / down	15 49
	See notes for independent or single input.				

10.2 Input xx: functions setup

10.2.1 Independent or single

For the independent or single channel setting, all parameters listed below are referred to the corresponding functions.

In the following sections, it is implicitly understood that for the listed parameters to appear, the corresponding Input xx (where xx = 2, 3, 4) 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 values of type:

Parameter name	Condition	Values
Lock Function	-	enabled / **disabled**
	Enables or disables the capability of locking the input through a remote command (telegram).	
Lock function – Invert lock device signal	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = send values or sequences	**not inverted** / inverted
	Allows interpreting a "lock activate" telegram as unlock and vice-versa.	
Lock function – Lock after bus recovery	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = send values or sequences	**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.	

10.2.2 Independent or single: Lock function enabled

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input x - Lock Function	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Lock function = enabled	1 bit	C-W-U	[1.003] enable	2, 19 (1, 2) 36, 53 (3, 4)

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

Details are shown in the following sections; a resume of all options is shown in the table below

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

10.2.3 Independent or single: send values or sequences

Parameter name	Conditions	Values
Number of communication objects	Input xx = independent or single Type = send values or sequences	1 ... 8 **(1)**
	Number of communication objects configured in association with the button event.	
Lock Function - Behaviour at locking	Input xx = independent or single Type = send values or sequences	**none** as close or short press as open or long press
	Allows performing the operation associated to the specified event when a locking command is received. You can choose between operations linked to two possible closing (or short press, depending on the configuration) or opening (or long press) events.	
Lock Function - Behaviour at unlocking	Input xx = independent or single Type = send values or sequences	**none** as close or short press as open or long press
	Allows performing the operation associated to the specified event when a locking command is received. You can choose between operations linked to two possible closing (or short press, depending on the configuration) or opening (or long press) events.	
Event	Input xx = 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	Input xx = 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 to discriminate between a short and a long press.	
Object n - Send delay	Input xx = 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	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8	nobody **off / value 1** on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated. *The cyclical transmission is only available if the number of communication objects to link is 1.*	

Parameter name	Conditions	Values
Object n - Cyclic sending interval	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Cyclical sending ≠ none	hh: mm: ss **(00:02:00)**
	Interval between cyclican transmissions.	
Object n - communication object dimension	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8	**1 bit value** 2 bits value 1 byte unsigned value 1 byte percentage value 1 byte signed value 2 bytes unsigned value 2 bytes signed value 2 bytes floating value
	Defines size and type of the values to be sent when an event occurs.	
Object n - Activation / release	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = 1 bit value Event = activation / release	none on off **toggle**
	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = 2 bits value Event = activation / release	none disable enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = *any byte value* Event = activation / release	none **send value 1** send value 2 send value 1 ↔ send value 2
	Value change behaviour caused by either an activation or release event (according to event configuration)	
Object n - long or short press	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = 1 bit value Event = long / short press	none on off **toggle**
	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = 2 bits value Event = long / short press	none disable enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down

Parameter name	Conditions	Values
	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = *any byte value* Event = long / short press	nons **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	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = *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	Input xx = independent or single Type = send values or sequences Number of com. objects = 1,…,8 Comm. Object dim. = *any byte value*	*Same as value 1*
	Second value available for association in send events.	

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx – Switching status *[type]* Object *n*	Input xx = independent or single Type = send values or sequences	see table below	C-WTU	see table below	3, 20 (1, 2) 37, 54 (3, 4)
	Up to 8 objects can be defined and associated to the same event. *The listed COs refer to the first of these objects 8 (for each of the inputs); the CO after the first one are sequential. To obtain the CO number for the n-th object, simply add (n-1) to the numbers given.* *Example: the COs associated to input 3 have numbers starting at 37. The number of the 5th CO associated with that input will then be 37 + (5-1) = 41.* *Types and dimensions of the individual objects can be configured as described in the next paragraphs.*				

The data size and Point Data Types are as follows:

Data 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 with sign	[6.*] 8-bit signed value
2 bytes unsigned	[7.*] 2-byte unsigned value
2 bytes with sign	[8.*] 2-byte signed value
2 bytes floating point	[9.*] 2-byte float value

10.2.4 Independent or single: dimming

Parameter name	Conditions	Values
Long press time	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = dimming	hh: mm: ss.fff **(00: 00: 08.000)**
	Minimum push time for a press, to discriminate between short or long press.	
Toggle mode	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) 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 = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = dimming Toggle mode = enabled	**darker** brighter darker ↔ brighter
	It defines the function to be assigned to prolonged pressure. *If toggle mode is enabled, the short press is already assigned the function toggle.*	
Short / long action	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) 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 = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) 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 = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = dimming Send cyclically ≠ none	hh: mm: ss **(00:02:00)**
	Interval between cyclical transmissions.	
Lock Function - Behavior at locking	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = dimming Lock function = enabled	**none** off on toggle

Parameter name	Conditions	Values
	Value to be assigned to the object when a locking command is received.	
Lock Function - Behavior at unlocking	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = dimming Lock function = enabled	**none** off on as previous
	Value to be assigned to the object when an unlocking command is received.	

10.2.5 Independent or single: scene

Parameter name	Conditions	Values
First scene number	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) 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 = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) 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 = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) 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 = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) 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 = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = scene Learning Mode = disabled Activation scenery = toggle between 2 scenes	1 ... 64 **(2)**

Parameter name	Conditions	Values
	Alternate scene number to be assigned to button press.	
Lock Function - Behavior at locking	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = scene Lock function = enabled	**none** send first scene send second scene
	Operation to perform when a locking command is received.	
Lock Function - Behavior at unlocking	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = scene Lock function = enabled	**none** send first scene send second scene
	Operation to perform when an unlocking command is received.	

10.2.6 Coupled

For a coupled channel, all the parameters are referred to the single menu entry for Input 1-2 and Input 3-4. For all Type values:

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

10.2.7 Coupled: Lock function enabled

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Input xx - Lock Function	Channel x = coupled Lock function = enabled	1 bit	C-W-U	[1.003] enable	2, 36

10.2.8 Coupled: switch

Parameter name	Conditions	Values
1 and 2 use	Channel x = coupled Type = switching	**A on, B off** A off, B on
	Allows to invert side A and side B functionalities	

Parameter name	Conditions	Values
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 Lock function = enabled	**none** on off toggle
	Value to be assigned to the object when a locking command is received..	
Lock Function - Behaviour at unlocking	Channel x = coupled Type = switching Lock function = enabled	**none** on off as previous
	Value to be assigned to the object when an unlocking command is received.	

10.2.9 Coupled: dimming

Parameter name	Conditions	Values
Long press time	Channel x = coupled Type = dimming	hh: mm: ss.fff **(00: 00: 00.800)**
	Minimum push time for a press, to discriminate between short or long press.	
1 and 2 use	Channel x = coupled Type = dimming	**A increases, B decreases** A decreases, B increments
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.	
Cyclical sending interval	Channel x = coupled Type = dimming Cyclical sending ≠ none	hh: mm: ss **(00:02:00)**
	Interval between cyclical transmissions.	
Lock Function - Behavior at locking	Channel x = coupled Type = dimming Lock function = enabled	**none** on off toggle
	Value to be assigned to the object when a locking command is received.	
Lock Function - Behavior at unlocking	Channel x = coupled Type = dimming Lock function = enabled	**none** on off as previous
	Value to be assigned to the object when an unlocking command is received.	

10.2.10 Coupled: shutter or venetian blind

Parameter name	Conditions	Values
Long press time	Channel X = coupled Type = shutter or venetian blind	hh: mm: ss.fff **(00: 00: 00: 8.000)**
	Minimum push time for a press, to discriminate between short or long press.	
1 and 2 use	Channel X = coupled Type = shutter or venetian blind	**A up, B down** A down, B up
Venetian blind mode	Channel X = coupled Type = shutter or venetian blind	**enabled** / disabled
	If venetian 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 - Behavior at locking	Channel x = coupled Type = shutter or venetian blind Lock function = enabled	**none** on off toggle
	Value to be assigned to the object when a locking command is received.	
Lock Function - Behavior at unlocking	Channel x = coupled Type = shutter or venetian blind Lock function = enabled	**none** on off as previous
	Value to be assigned to the object when an unlocking command is received.	

For other communication objects related to the coupled mode, please refer to paragraph 6.4.

10.3 Temperature sensor

The Temperature Sensor tab contains the following parameters:

- Enable the sensor
- Filter Type on internal data processing
- Temperature offset correction
- Minimum change of value to send [K]
- Cyclic sending interval
- Threshold 1
- Threshold 2

10.3.1 Parameters and communication objects

Parameter name	Conditions	Values
Temperature sensor		**enabled** disabled
Filter Type	Temperature sensor = enabled	low **medium** high
	Low = average value every 4 measurements *Average = average every 16 measurements* *High = average value every 64 measurements*	
Temperature offset	Temperature sensor = enabled	**0 ° C** [Range -5 ° C ... + 5 ° C]
Minimum change of value to send [K]	Temperature sensor enabled =	**0.5** [0 ... 5]
	If the parameter is set to 0 (zero), no value is sent after a change.	
Cyclical sending interval	Temperature sensor = enabled	**no sending** [Other values in 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]
Threshold 2	Temperature sensor = enabled	**not active** below above
Value [° C]	Temperature sensor = enabled, Threshold 2 = below or above	**45** [range 0 ... 50]

Parameter name	Conditions	Values
Hysteresis	Temperature sensor = enabled, Threshold 1 and / or threshold 2 = below or above	**0.4 K** [Other values between 0.2 K and 3 K]
Cyclical sending interval	Temperature sensor = enabled, Threshold 1 and / or threshold 2 = below or above	**no sending** [Other values in the range 30 s ... 120 min]

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Temperature value	Input x = ind. or single, Input 1 = NTC	2 Byte	CR-T--	[9.001] temperature (° C)	90
Temperature Threshold 1 - Switch	Input x = ind. or single, Input 1 = NTC, Threshold 1 = below or above	1 Bit	CR-T--	[1.001] switch	91
Temperature Threshold 2 - Switch	Input x = ind. or single, Input 1 = NTC, Threshold 2 = below or above	1 Bit	CR-T--	[1.001] switch	92

10.3.2 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.

10.3.3 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).

10.3.4 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. Enabling an external sensor, without connecting the corresponding communication object, generates a permanent alarm on the display and suspends the thermoregulation function.

The external sensors board (the bus) contains the following parameters:

- Room temperature
- Relative humidity
- Antistratification temperature
- Outdoor temperature
- (Heat exchange) coil temperature
- Floor surface temperature
- Flow temperature

- Analog sensors timeout

- Anticondensation
- Window contact X (X = 1, 2)
- Presence Sensor X (X = 1, 2)
- Contact card holder pocket

- Digital sensors timeout

10.3.5 Parameters and communication objects

Parameter name	Conditions	Values
Room temperature		**disabled** / enabled
	It enables a bus temperature sensor. The measured value can be used to calculate a weighted average value in combination with the temperature sensor integrated into the device or a temperature sensor connected to a device input.	
Cyclic reading interval	Room temperature = enabled	**no reading** [Other values in 30 s ... 120 min range]
	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 bytes (DPT 9.007)**
Cyclic reading interval	Relative humidity = enabled	**no reading** [Other values in 30 s ... 120 min range]

Parameter name	Conditions	Values
Antistratification temperature		**disabled** / enabled
	It enables a temperature bus sensor to carry out the antistratification function.	
Cyclic reading interval	Temperature stratification = enabled	**no reading** [Other values in 30 s ... 120 min range]
Outdoor temperature		**disabled** / enabled
	Enables a bus outdoor temperature sensor.	
Cyclic reading interval	Outdoor temperature = enabled	**no reading** [Other values in 30 s ... 120 min range]
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 30 s ... 120 min]
Floor surface 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	Floor surface temperature = enabled	**no reading** [Other values in 30 s ... 120 min range]
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 30 s ... 120 min range]
Analog sensors timeout		**0:05:00 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
	enables a bus sensor for detecting the condensation.	
Signal	Anticondensation = enabled	**not inverted** / inverted
Cyclic reading interval	Anticondensation = enabled	**no reading** [Other values in 30 s ... 120 min range]

Parameter name	Conditions	Values
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 30 s ... 120 min range]
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 30 s ... 120 min range]
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 30 s ... 120 min range]
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 30 s ... 120 min range]
Card holder pocket		**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

Parameter name	Conditions	Values
Cyclic reading interval	Card holder contact = enabled	**no reading** [Other values in 30 s ... 120 min range]
Timeout digital sensors		**0:05:00 hh: mm: ss** [Field ... 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.	

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Room temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] Temperature (° C)	93
Humidity (2 bytes, from bus)	Relative Humidity Sensor = enabled, Humidity CO size = 2 bytes	2 Byte	C-WTU	[9.007] humidity (%)	94
Humidity (1 byte, from bus)	Relative Humidity Sensor = enabled, Humidity CO size = 1 byte	1 Byte	C-WTU	[5.001] percentage (0..100%)	95
Antistratification temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] Temperature (° C)	96
Outdoor temperature (from the bus)	enabled	2 Byte	C-WTU	[9.001] temperatures ° C	97
Coil temperature (from the bus)	enabled	2 Byte	C-WTU	[9.001] Temperature (° C)	98
Floor temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] Temperature (° C)	99
Flow temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] Temperature (° C)	100
Anticondensation (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	106
Windows contact sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.019] window / door	101
Windows contact sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.019] window / door	102

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
Presence Sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.018] occupancy	103
Presence Sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.018] occupancy	104
Contact of card holder (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	105

10.3.6 *Note for sensors 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 related alarm code (see also the alarms list in the Diagnostics paragraph).

10.4 Weighted temperature value

The weighted temperature value tab appears only if two sensors for measuring the room temperature are enabled; it contains the following parameters:

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

10.4.1 Parameters and communication objects

Parameter name	Conditions	Values
Relative weight		100% main sensor 90% / 10% 80% / 20% 70% / 30% 60% / 40% **50% / 50%** 40% / 60% 30% / 70% 20% / 80% 10% / 90% 100% sensor from bus
Minimum change of value to send [K]	Relative weight ≠ 100% for main sensor and relative weight ≠ 100% for sensor from bus	**0.5** [Other values in 0 ... 5 K range]
	If the parameter is set to 0 (zero), no value is sent at the change.	
Cyclic sending interval	Relative weight ≠ 100% for main sensor and relative weight ≠ 100% for sensor from bus	**no sending** [Other values in 30 s ... 120 min range]

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Weighted temperature	Cyclic sending interval ≠ no sending	2 Byte	CR-T-	[9.001] temperature ° C	107

10.4.2 Note for weighted temperature

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

1) from the integrated temperature sensor;
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.

10.5 Temperature control

The temperature control tab contains the following sub-tabs:

- Settings
- Heating
- Cooling
- Ventilation

The **cooling** tab appears only if in “Temperature control – Settings” tab is set to either “both heating and cooling” value or “cooling”.La sub-tab appears only if scenarios in the Scenarios parameter settings tab is set to Enabled value.

The **ventilation** tab is available only if Heating or Cooling type is set to “fancoils”.

10.5.1 Settings

The Settings tab contains the following parameters:

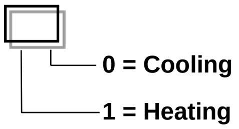
- Setpoint type
- Thermostat function
- Command Communication Object unique or separated (2 or 4 pipes systems)
- Heating/Cooling changeover
- Heating/Cooling cyclic sending interval
- HVAC mode after download
- Setpoint cyclic sending interval
- End of manual operation
- Disable temperature controller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

10.5.2 Parameters and communication objects

Parameter name	Conditions	Values
Setpoint type	Thermostat = enabled	**single** absolute relative
Thermostat function	Thermostat = enabled	**heating** cooling both heating and cooling
Command Communication Object	Thermostat function = both heating and cooling	**separated** / unique
Heating/Cooling changeover	Thermostat function = heating and cooling	**from bus** / automatic

Parameter name	Conditions	Values
Heating/Cooling cyclic sending interval	Thermostat function = both heating and cooling	**no sending** [Other values in 30 s ... 120 min range]
HVAC mode after download	Thermostat function = both heating and cooling	**no change** heating cooling
Setpoint cyclic sending interval		**no sending** [Other values in 30 s ... 120 min range]
	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		**at first telegram from bus** [Other values in 30 min ... 48 h range]
	It defines how to exit from manual mode.	
Disable temperature controller from bus		**no** / yes
	It defines if the temperature controller can be disabled from bus.	
Signal from bus	Disable temperature control from bus = yes	**not inverted** inverted
Transmission delay after mode change	Disable temperature control from bus = yes	hh: mm: ss.fff **(00: 00: 04.000)**
	It defines the delay for transmission on the bus, after an HVAC mode changeover. *A null value (00:00:00) means that the transmission is immediate.*	
Valve protection function		**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 5 s ... 20 min range]

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Actual setpoint		2 Byte	CR-T-	[9.001] Temperature (° C)	115

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Manual setpoint		2 Byte	C-W--	[9.001] Temperature (° C)	116
Heating / cooling status out	Always visible	1 Bit	CR-T-	[1.100] cooling / heating	108
	The communication object is updated on the bus, on event of internal change processed by the controller. The object is always exposed and contains the information about the current conduction mode of the internal temperature controller. **[1.100] DPT Heating / Cooling 1** 				
Heating / cooling status in	Thermostat function = both heating and cooling; heating / cooling switchover = from bus	1 Bit	C-W--	[1.100] cooling / heating	109
	The communication object is received from the bus. On switching event, the internal controllers of primary and auxiliary stages (if enabled) switch their operating mode. The actual operating mode is displayed by the corresponding icon.				
HVAC mode in		1 Byte	C-W--	[20.102] HVAC mode	110
	The device receives the operating mode (HVAC mode) from a bus device with the function of supervisor. The operating mode received through this communication object can be later modified by the user (in this case the room thermostat switches to manual mode).				
HVAC forced mode in		1 Byte	C-W--	[20.102] HVAC mode	111
	The communication object allows to receive the operating mode similarly to "HVAC mode in" communication object. The difference is that the operating mode received via this object (with the exception of AUTO) can no longer be modified by the user. The user can modify the operating mode only after "HVAC forced mode in" communication object has sent the AUTO command.				
HVAC mode out		1 Byte	CR-T-	[20.102] HVAC mode	112
HVAC manual mode		1 Byte	C-WTU	[20.102] HVAC mode	113
Chrono active status		1 Bit	CR-T-	[1.011] state	114
Manual / forced setpoint active status		1 Bit	CRWTU	[1.011] state	138
Building protection HVAC mode active		1 Bit	CR-T-	[1.011] state	155

11.6.1.1 Note on system terminals for heating and cooling

The application functions of the room temperature controller that can be configured via ETS, are particularly suitable to the command / control by means of KNX actuators (generic or dedicated) of the following heating/cooling terminal units:

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

10.5.3 Heating

The Heating tab contains the following parameters:

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

10.5.4 Parameters and communication objects

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

Parameter name	Conditions	Values
Comfort temperature setpoint [° C]		**21** [Range 10 ... 50]
Standby temperature setpoint [° C]		**-30** [Range -50 ... -10]
	For a correct operation of the device, the standby temperature setpoint has to be lower than the comfort temperature setpoint.	
Economy temperature setpoint [° C]		**-50** [Range -80 ... -10]
	For a correct operation of the device, the economy temperature setpoint has to be lower than the standby temperature setpoint.	
Building protection temperature setpoint [° C]		**7** [Range 2 ... 10]
Heating type		**radiators** electric fancoils 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 points hysteresis** PWM (pulse width modulation to) continuous
Hysteresis	Control Type = 2 points hysteresis	**0.3 K** [Other values in the 0.2 ... 3 K range]
Hysteresis position	Heating type = floor radiant panels, ceiling radiant panels Control Type = 2 points hysteresis, continuous	**below** / above
	The above hysteresis is suitable in case of special applications requiring mixing group control.	
Cyclic sending interval	Control Type = 2 points hysteresis, continuous	**no sending** [Other values in 30 s ... 120 min range]
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]

Parameter name	Conditions	Values
Proportional band [0.1 K]	control type = continuous or PWM	**50** [Other values in the 5 ... 100 range]
	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	**240** [0 ... 255]
Min control value [%]	control type = continuous or PWM	**15** [Range 5 ... 240]
Max control value [%]	control type = continuous or PWM	**85** [Range 70 ... 100]
Floor temperature limitation	Heating type = floor radiant panels, External sensors: floor surface temperature sensor = 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 overtemperature protection usually installed in hydronic floor systems, realized with the proper safety thermostat.*	
Temperature limit [° C]	Limitation floor temperature = enabled	**29** [Range 20 ... 40]
	According to standard EN 1264, a maximum allowable temperature is prescribed for the surface of a floor heating system: • *T (sup) max ≤ 29 ° C for normal occupation zones;* • *T (sup) max ≤ 35 ° C for the peripheral areas of the environments.* *National regulations may also limit these temperatures to lower values. Peripheral zones are generally located along the walls of the environment bands facing the exterior of the building with a maximum width of 1 m.*	
Hysteresis [K]		**0.3 K** [Other values in the 0,2 ... 3 K range]
	Before quitting from the alarm status, the device waits until the surface temperature decreases under the hysteresis threshold set.	
Auxiliary heating	Auxiliary heating = enabled	**Disabled** / enabled
Communication object	Auxiliary heating = enabled	**separated** / unique
	It enables the activation and deactivation of the function through a telegram sent on the bus by a supervising device.	
Disabled from bus	Auxiliary heating = enabled	**No** / yes

Parameter name	Conditions	Values
	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 0 ... 3 K range]
Hysteresis	Auxiliary heating = enabled	**0.3 K** [Other values in the range 0.2 ... 3 K K]
Cyclic sending interval	Auxiliary heating = enabled	**no sending** [Other values in 30 s ... 120 min range]
Ventilation for auxiliary heating	Heating type = 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	Size	Flags	DPT	*Comm. Obj. N.*
Comfort setpoint (heating)		2 Byte	CRWTU	[9.001] Temperature (°C)	117
Standby setpoint (heating)		2 Byte	CRWTU	[9.001] Temperature (°C)	119
Economy setpoint (heating)		2 Byte	CRWTU	[9.001] Temperature (°C)	121
Building protection setpoint (heating)		2 Byte	CRWTU	[9.001] Temperature (°C)	123
Heating out command	Control type = 2 points hysteresis or PWM Command communication object = separated	1 Bit	CR-T-	[1.001] switch	126
Heating out command	Control type = continuous Command communication object = separated	1 Byte	CR-T-	[5.001] percentage (0..100%)	126
Heating and cooling out command	Control type = 2 points hysteresis or PWM Command communication object = unique	1 Bit	CR-T-	[1.001] switch	126

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Heating and cooling out command	Control type = continuous Command communication object = unique	1 Byte	CR-T-	[5.001] percentage (0..100%)	126
Auxiliary heating out command	Auxiliary heating = enabled Command communication object = separated	1 Bit	CR-T-	[1.001] switch	128
Auxiliary heating and cooling output command	Auxiliary heating = enabled Command communication object = unique	1 Bit	CR-T-	[1.001] switch	128
Auxiliary heating disable	Auxiliary heating = enabled Disabled from bus = yes	1 Bit	CW---	[1.003] enable	130

11.5.4.1 Note on floor temperature limitation function

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 (TSmax) for the surface of the floor that is physiologically correct defined as:

- T_{Smax} ≤ 29 ° C for zones of normal occupancy;
- T_{Smax} ≤ 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 implemented 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.

10.5.5 Cooling

The cooling tab includes the following parameters:

- Comfort temperature setpoint [° C]
- Standby temperature setpoint [° C]
- Economy temperature setpoint [° C]
- Building protection temperature setpoint [° C]
- Cooling 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 [%]
- Anticondensation with probe
- Active anticondensation
- Flow temperature (project)
- Anticondensation hysteresis range
- Delay for alarm signal
- Auxiliary cooling
- Disabled from bus
- Offset from setpoint
- Hysteresis
- Cyclic sending interval
- Ventilation for auxiliary cooling

10.5.6 Parameters and communication objects

Conditions: General ⇒ Thermostat function = cooling or heating and cooling.

Parameter name	Conditions	Values
Comfort temperature setpoint [° C]		**23** [Range 10 ... 50]
Standby temperature setpoint [° C]		**30** [Range 10 ... 50]
	For a correct operation of the device the standby temperature setpoint has to be greater than the comfort temperature setpoint.	

Parameter name	Conditions	Values
Economy temperature setpoint [° C]		**50** [Range 10 ... 80]
	For a correct operation of the device the standby temperature setpoint has to be greater than the economy temperature setpoint.	
Building protection temperature setpoint [° C]		**36** [Range 30 ... 50]
Cooling type		Fancoils, **floor radiant panels,** ceiling radiant panels
	If in Settings Tab 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	Command communication object = separated	**2 points hysteresis,** PWM (pulse width modulation), continuous
	If in Settings Tab the parameter Thermostat function = both heating and cooling and Command communication object = unique, the parameter Control type is bound to the choice done for Heating.	
Hysteresis	Control Type = 2 points hysteresis	**0.3 K** [Other values in 0.2 ... 3 K range]
Hysteresis position	Cooling type = floor radiant panels, ceiling radiant panels Control Type = 2 points hysteresis	**below** / above
Cyclic sending interval	Control type = 2 points hysteresis, continuous	**no sending** [Other values in 30 s ... 120 min range]
Minimum change of value to send [%]	Control type = continuous	**10** [range 0 ... 100]
PWM cycle time	Control type = PWM (pulse width modulation)	**15 min** [Range 5 ... 240 min]
Proportional band [0.1 K]	Control type = continuous or PWM	**50** [range 0 ... 255]

Parameter name	Conditions	Values
	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	**240** [0 ... 255 min]
	**) The field contains a preset value depending on the selected cooling type (the value can be modified):* • *fancoils: 90 min* • *floor radiant panels: 240 min* • *ceiling radiant panels: 180 min* *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.*	
Minimum control value [%]	Control type = continuous or PWM	**15** [range 0 ... 30]
Maximum 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
Active anticondensation	Cooling type = floor radiant panel or ceiling radiant panels, External sensors (from bus) ⇒ Flow temperature sensor = disabled	**disabled** enabled (project temperature)
	Cooling type = floor radiant panel or ceiling radiant panels, External sensors (from bus) ⇒ Flow temperature sensor = enabled	disabled **enabled (flow and dew point temperature comparison)**
	If the flow temperature is lower than the calculated dew point temperature, the operating mode is cooling and the room thermostat is in flow request, then the thermostat will close the valve and display an alarm condition.	
Flow temperature (project)	Cooling type = floor radiant panel or ceiling radiant panels, External sensors (from bus) ⇒ Flow temperature sensor = disabled Active anticondensation = enabled (project temperature)	**14 ° C** [Other values in 14.5 ° C ... 20 ° C range]
	Only displayed if the flow temperature from an external sensor (from bus) is not available.	

Parameter name	Conditions	Values
Anticondensation hysteresis range	Active anticondensation = enabled	**0.5 K** [Other values in 0.2 ... 3 K range]
Delay for alarm signal	Anticondensation with probe = enabled	30s, 1 min, 2 min, 3 min, 4 min, **5 min**, 7 min, 10 min, 15 min, 20 min, 30 min, 45 min, 60 min, 90 min, 120 min
	Before exiting the alarm condition, it waits for a configurable delay	
Auxiliary cooling		**disabled** / enabled
Disabled from bus	Auxiliary cooling = enabled	**no** / yes
	The parameter enables the activation and deactivation of the function through a telegram from a bus device with supervisor function.	
Offset from setpoint	Auxiliary cooling = enabled	0.2 K / **0.3 K** / K 0.4 / 0.5 / 0.6 K 0.8 K / 1 K / 1.5 K / 2 K / 2.5 K / 3 K
Hysteresis [K]	Auxiliary cooling = enabled	0.2 K / **0.3 K** / K 0.4 / 0.5 / 0.6 K 0.8 K / 1 K / 1.5 K / 2 K / 2.5 K / 3 K
Cyclic sending interval	Auxiliary cooling = enabled	**no sending**, 30s, 1 min, 2 min, 3 min, 4 min, 5 min, 7 min, 10 min, 15 min, 20 min, 30 min, 45 min, 60 min, 90 min, 120 min
	The "no sending" value means that cyclic sending is not enabled.	
Ventilation for auxiliary cooling	Cooling type = floor radiant panel or ceiling radiant panels	**disabled** / enabled
	This option allows combining a high-inertial system as the floor radiant panels to a low-inertial one as the fancoils.	

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Comfort setpoint (cooling)		2 Byte	CRWTU	[9.001] Temperature (° C)	118
Standby Setpoint (cooling)		2 Byte	CRWTU	[9.001] Temperature (° C)	120
Economy setpoint (cooling)		2 Byte	CRWTU	[9.001] Temperature (° C)	122
Building protection setpoint (cooling)		2 Byte	CRWTU	[9.001] Temperature (° C)	124
Cooling out command	Control type = hysteresis 2-point or PWM	1 Bit	CR-T-	[1.001] switch	127

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Cooling out command	Control type = continuous	1 Byte	CR-T-	[5.001] percentage (0..100%)	127
Auxiliary cooling output command	Auxiliary cooling = enabled	1 Bit	CR-T--	[1.001] switch	129
Auxiliary cooling disable	Auxiliary cooling = enabled Disabled from bus = yes	1 Bit	C-W--	[1.003] enable	131
Anticondensation alarm	Active anticondensation = enabled	1 Bit	CR-T-	[1.005] alarm	153

11.5.6.1 Note on 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 achieved:

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

Nr.	Type	Denomination	Description
1a	Passive	Anticondensation protection by probe (via bus)	The thermostat receives the information about condensation via bus from a different KNX device through communication object 27: 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 object involved is 45: 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 acts 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: 100 - Flow temperature (from bus) [DPT 9.001 temperature °C] as input and 127 - Cooling out command [DPT 1.001 switch].
3	Active	Anticondensation protection with dew-point temperature sending over the bus and adjustment of the flow temperature	Software protection that foresees the sending on the bus of the dew-point temperature calculated by the room thermostat using temperature and relative humidity values to a KNX device capable of controlling the mixing condition of the conveying fluid for the cooling circuit. The regulation is performed by the KNX device receiving the dew-point temperature sent by the thermostat. The communication object involved is 144: Dew-point temperature [DPT 9.001 temperature °C].

Figure 16 – Achievable 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 the table).

If you are using the comparison between the dew point temperature calculated by the thermostat and the flow temperature of the heat transfer fluid, there are three possibilities:

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

10.5.7 Main and auxiliary ventilation

The Ventilation tab contains the following parameters:

- Ventilation function
- Control type
- Threshold first speed [0.1 K]
- Threshold second speed [0.1 K]
- Threshold third speed [0.1 K]
- Speed control hysteresis [K]
- Speed proportional band [0.1 K]
- Minimum change of value to send [%]
- Manual operation

- Hot start
- Min. temperature to start ventilation [°C]
- Antistratification function
- Antistratification temperature 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 Tab are:

- Heating ⇒ Heating type = fancoils, or Cooling type = fancoils or a combination of the two conditions;

- Heating ⇒ Heating type = floor radiant panels or ceiling radiant panels and Heating ⇒ Ventilation ⇒ Auxiliary heating = enabled
- Cooling ⇒ Cooling type = floor radiant panels or ceiling radiant panels and Cooling ⇒ Ventilation ⇒ Auxiliary cooling = enabled

In this way it is possible to control two types of systems:

- fan-coil terminals, or
- radiant panels as main stage and fancoil terminals as auxiliary stage.

10.5.8 Parameters and communication objects

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** [0 ... 255]
	The value is represented in tenths of Kelvin degrees. If the parameter Thermostat function = both heating and cooling, the threshold value is valid for both seasonal modes.	
Threshold second speed [0.1 K]	Control type ≥ 2 speeds	**10** [0 ... 255]
	The value is represented in tenths of Kelvin degrees. If the parameter Thermostat function = both heating and cooling, the threshold value is valid for both seasonal modes. For a correct operation of the ventilation, Threshold second speed > Threshold first speed.	

Parameter name	Conditions	Values
Threshold third speed [0.1 K]	Control type = 3 speeds	**20** [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 [K]	Control type = 1, 2 or 3 speeds	**0.3 K** [Other values in 0.2 ... 3 K range]
Proportional band [0.1 K]	Control type = continuous regulation	**30** [0 ... 255 range]
	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.	
Minimum change of value to send [%]	Control type = continuous regulation	**10** [2 ... 40 range]
	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, 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 must be enabled. To this purpose an external sensor (from bus) can be used.	
Min. temperature to start ventilation [° C]	Hot start = yes	**28** [28 ... 40 range]
	If enabled, the function is active in heating mode only.	
Antistratification function	External sensors from bus ⇒ antistratification temperature = enabled	**disabled** / enabled
	For carrying 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 0.25 ... 4.00 range]
	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 0.2 ... 3 K range]

Parameter name	Conditions	Values
Disable ventilation from bus		**no** / yes
Signal from bus	Disable ventilation from bus = yes	**not inverted** inverted
Fan start delay		**0 s** [Other values in the 10 s ... 12 min range]
	It is displayed also 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 30 s ... 120 min range]

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Fan continuous speed	Control type = continuous regulation	1 Byte	CR-T-	[5.001] percentage (0..100%)	132
Fan speed 1	Control type ≥ 1 speed	1 Bit	CR-T-	[1.001] switch	133
Fan speed 2	Control type ≥ 2 speeds	1 Bit	CR-T-	[1.001] switch	134
Fan speed 3	Control type = 3 speeds	1 Bit	CR-T-	[1.001] switch	135
Fan control disable	Disable ventilation from bus = yes	1 Bit	C-W--	[1.002] boolean	136
Fan manual speed		1 Byte	CRWTU	[5.010] counter pulses (0 ... 255)	139
Fan speed status		1 Byte	CR-T-	[5.010] counter pulses (0 ... 255)	140
Fan manual active status		1 Bit	CRWTU	[1.011] state	141

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Fan manual speed percentage		1 Byte	CR-T-	[5.001] percentage (0..100%)	156
Fan manual speed off status		1 Bit	CR-T-	[1.011] state	157

10.5.9 Delay fan start function ("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 (approximated function).

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

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

10.5.10 Antistratification function

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

The antistratification function requires:

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

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

10.5.11 2-stage configuration with fancoils 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 selected, the parameter Cooling ⇒ Cooling type does not appear)
4. Heating ⇒ Auxiliary heating = enabled
5. Auxiliary heating ⇒ Communication object = separated
6. Heating ⇒ Ventilation for auxiliary heating = enabled
7. Cooling ⇒ Cooling type = fancoils

Important!

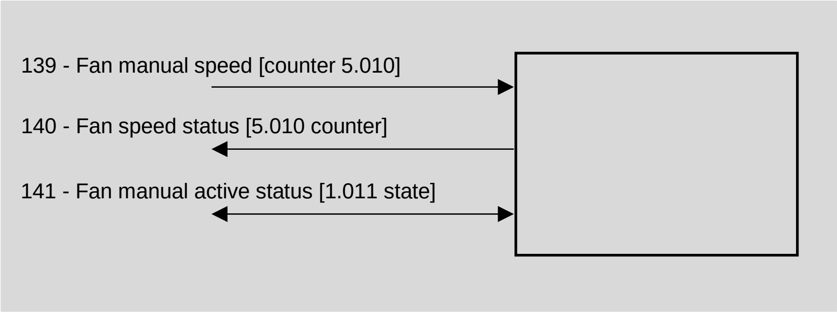
If the fan-coil system has a 2-pipe hydraulic configuration, the objects Auxiliary heating output command (1 bit) and Cooling out command (1 byte) have to be set in logical OR in the actuator for controlling the fan-coil which in this case is unique.

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.

10.5.12 Remote fan speed modification

The communication objects shown in figure allow to monitor 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.


139 - Fan manual speed [counter 5.010]
140 - Fan speed status [5.010 counter]
141 - Fan manual active status [1.011 state]

The C.O. *140 – Fan speed (status)* allows to evaluate the actual fan speed; the C.O. *141 – Fan manual active status* contains the information about automatic (=0, not active) or manual (=1, active) operating mode. By modifying the C.O. *139 – 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. *141 – Fan manual active status* (=0, not active).

Accepted values for C.O.s 139 and 140 depend on the number of speeds set in ETS.

If *Control Type* parameter in Ventilation tab is set to 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%

10.6 Relative humidity control

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

- Dehumidification
- Humidification

The secondary Tabs *Dehumidification* and *Humidification* appear if the humidity sensor is enabled.

The acquisition of relative humidity is done via bus from a KNX relative humidity sensor.

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

- Local display and sending on the bus (for information purpose) through DPT [9.007] percentage (%);
- Use of detected value for derivated psycrometric values calculations and sending on the bus through corresponding DPTs;
- Calculation of correlated indexes (perceived temperature) for local display or sending on the bus;
- 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.

10.6.1 Dehumidification

The secondary Tab *humidification* includes the following parameters:

- Operating modes where dehumidification is active (heating, cooling or both);
- Relative humidity setpoint for dehumidification control [%];
- Dehumidification control hysteresis [%];
- Dehumidification subordinated to temperature control;
- Dehumidification start delay;
- Integration;
- Function of integration of sensible heat (temperature difference for integration);
- Hysteresis for integration;
- Cyclic sending interval;
- Disable dehumidification control from bus;
- Signal from bus.

10.6.2 Parameters and communication objects

Parameter name	Conditions	Values
Dehumidification function		**disabled** cooling only heating only both cooling and heating
	Parameter that sets the dehumidification function.	
Humidity setpoint [%]	Dehumidification function ≠ disabled	**55** [Range 30 ... 60]
Hysteresis	Dehumidification function ≠ disabled	**0.8%** [Other values in the 0.5 ... 4% range]
Subordinated to temperature control	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ type = floor radiant panels or ceiling radiant panels Relative humidity control ⇒ dehumidification ⇒ dehumidification cooling only = function	**no** / yes
Dehumidification start delay	Subordinate to temperature control = no	**00:05:00 hh: mm: ss** [00:00:00 … 18:12:15 range]
	Value 00:00:00 means that the start delay is not enabled.	
Integration		**no** / yes
Temperature difference for integration	Integration = yes	**1.5 ° C** [Other values in 0.5 ... 3 ° C range]
Hysteresis for integration	Integration = yes	**0.5 K** [Other values in 0.2 ... 3 K range]
Cyclic sending interval		**no sending** [Other values in 30 s ... 120 min range]

Parameter name	Conditions	Values
Disable dehumidification control from bus		**no** / yes
Signal from bus	Disable dehumidification control from bus = yes	**not inverted** / inverted

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Relative humidity setpoint for dehumidification		2 Byte	CRWTU	[9.007] humidity (%)	145
Dehumidification command		1 Bit	CR-T-	[1.001] switch	147
Dehumidification water battery command	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ Cooling type = floor radiant panels or ceiling radiant panels, Relative humidity control ⇒ dehumidification ⇒ dehumidification function = cooling only Subordinated to temperature control = No	1 Bit	CR-T-	[1.001] switch	148
Integration dehumidification 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	149
	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 from bus = yes	1 Bit	C-W--	[1.002] boolean	150

10.6.3 Humidification

The secondary Tab ***Humidification*** includes the following parameters:

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

10.6.4 Parameters and communication objects

Parameter name	Conditions	Values
Humidification function		**disabled** cooling only heating only both cooling and heating
	Parameter that selects the humidification function.	
Humidity setpoint [%]	Humidification ≠ disabled	**35** [25 ... 45% range]
Humidity hysteresis [%]		**0.8%** [Other values in 0.5 ... 4% range]
Cyclic sending interval		**no sending** [Other values in 30 s ... 120 min range]
Disable humidification control from bus		**no** / yes
Signal from bus	Disable humidification control from bus = yes	**not inverted** / inverted

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Relative humidity setpoint for humidification		2 Byte	CRWTU	[9.007] humidity (%)	146
Humidification command		1 Bit	CR-T-	[1.001] switch	151
Humidification control disable	Disable humidification control from bus = yes	1 Bit	C-W--	[1.002] boolean	152

10.6.5 Calculated psychrometric values

The sub-tab *Calculated psychrometric values* contains the following parameters:

- Dew Point [° C]
- Cyclic sending interval
- Min. change of value to send [K]

Condition for Tab displaying: Internal sensors ⇒ Relative Humidity Sensor = enabled.

10.6.6 Parameters and communication objects

Parameter name	Conditions	Values
Dew-point temperature		**disabled** / enabled
	The value of dew-point temperature sent over the bus, allows to implement an active condensation protection with a recalibration of the flow conditions of the heat transfer fluid, in case a control device for the mixing group is connected. If the thermostat is installed in an room where cooling function is not foreseen (e.g., a bathroom), it is appropriate to exclude the room from the control by setting the parameter "dew-point temperature = disabled".	
Cyclical transmission interval	Dew-point temperature = enabled	**no sending** [Other values in 30 s ... 120 min range]
Min. change of value to send [K]	Dew-point temperature = enabled	**0.2 K** / No sending [Other values in the 0.2 ... 3 K range]

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Dew-point temperature	Dew-point temperature = enabled	2 Byte	CR-T-	[9.001] temperatures ° C	144

10.7 Energy saving

In order to activate the 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*** tab contains the following sub-tabs:

- Window contacts
- Presence sensors
- Card holder

10.7.1 Window contacts

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

External sensors (from bus) ⇒ Windows contact sensor 1 or 2 (from bus) = enabled

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

- Window contacts function
- Wait time to building protection mode

10.7.2 Parameters and communication objects

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** [00:00:00 18:12:15 range]
	Time interval before the automatic switching of the device to the Building protection operating mode	

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Window contact sensor 1 (from bus)	Window contacts function = enabled, Window contact 1 = enabled	1 Bit	C-WTU	[1.019] window / door	101
Window contact sensor 2 (from bus	Window contacts function = enabled, Window contact 2 = enabled	1 Bit	C-WTU	[1.019] window / door	102

10.8 Presence sensors

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

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

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

External sensors (from bus) ⇒ Presence sensor 1 (from bus) or Presence sensor 2 (from bus) = enabled

10.8.1 Parameters and communication objects

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
	Parameter to indicate whether the presence has to extend the comfort mode, or if the absence has to limit the mode, or both behaviors.	

Parameter name	Conditions	Values
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 the HVAC mode	Presence sensor function = enabled	**00:01:00 hh: mm: ss** [00:00:00 18:12:15 range]
	Time interval before the automatic switching of the operating mode set in the Thermostat modes parameter.	

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Presence Sensor 1 (from bus)	Presence sensor function = enabled	1 Bit	C-WTU	[1.018] occupancy	103
Presence Sensor 2 (from bus)	Presence sensor function = enabled	1 Bit	C-WTU	[1.018] occupancy	104

10.9 Card holder

The **Card holder** secondary Tab is displayed 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** Tab includes the following parameters:

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

10.9.1 parameters and communication objects

Parameter name	Conditions	Values
Card holder function		**disabled** / enabled
	This parameter enables the card holder function.	
On card insertion switch HVAC mode to	Card holder function = enabled	None, **comfort** standby economy
	This parameter defines the operating mode the device should automatically switch to, by inserting the card into the holder.	
Activation delay on card insertion	Card holder function = enabled	**00:00:00 hh: mm: ss** [00:00:00 … 18:12:15 range]
	Time interval before the automatic switching of the operating mode, after inserting the card into the holder.	

Parameter name	Conditions	Values
On card removal switch HVAC mode to	Card holder function = enabled	None, **standby** economy building protection
	This parameter defines the operating mode the device should automatically switch to, by removing the card from the holder.	
Activation delay on card removal	Card holder function = enabled	**00:00:00 hh: mm: ss** [00:00:00 … 18:12:15 range]
	Time interval before the automatic switching of the operating mode, after the card has been removed from the holder.	

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Contact of card holder (from bus)	Card holder function = enabled	1 Bit	C-WTU	[1.018] occupancy	105

11.9.1.1 Note to the 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).

11 Logic Functions

The EK-CE2-TP device allow to use some useful logic functions (AND, OR, NOT and exclusive OR) in order to implement complex functions in the building automation system.

The following items can be configured:

- 4 channels of logical functions;
- 4 inputs (logic objects) for each channel.

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

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.

> **i** In case of wrong connection of the input communication objects 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.

11.1 Generality

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

11.1.1 Parameters and communication objects tables

The following condition has to be true: General ⇒ Logical 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		**00: 00: 04,000 hh: mm: ss.fff** [00: 00: 00,000 ... 00: 10: 55,350 range]
	Time interval between the bus voltage recovery and the first reading of the input communication objects for evaluating the logic functions	
Output cyclic transmission delay		**no sending** [Other values in 30 s ... 120 min range]
	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	
Logic object x	x = 1,2,3,4	**disabled** / enabled
Logic object 1 negated	Logic object x = enabled	**no** / yes

Parameter name	Conditions	Values
	By denying the logic object state, it is possible to implement complex combinatorial logic. Example: Output = (NOT (logic object 1) OR logic object 2).	
Logic object x read at startup	Logic object x = enabled	**no** / yes
Logic object x default value	Logic object x = enabled	**none** / off / on
Output send	Logic function = enabled	**both values** only value 0 only value 1
Output update	Logic function = enabled	**on value change** on value or input change

Object name	Conditions	Size	Flags	DPT	*Comm. Obj. N.*
Logic function X - Input 1	Logic function X = enabled Logic object 1 = enabled	1 Bit	C-WTU	[1.001] switch	70, 75, 80, 85
	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	71, 76, 81, 86
	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	72, 77, 82, 87
	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	73, 78, 83, 88
	X= 1,2,3,4				
Logic function X - Output	logic function X = enabled	1 Bit	C-WTU	[1.001] switch	74, 79, 84, 89
	X= 1,2,3,4				

12 Outputs configuration

12.1 State variables (communication objects)

The state of the device, and specifically of its interface elements (physical outputs) is based on internal state variables, that are automatically defined by the application program. These state variables, once assigned a group address, are actually KNX communication objects, which allows other devices on the bus to exploit the features of the device.

State variables undergo 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.

12.2 Outputs: independently and coupled

Outputs can be driven independently, or they can be coupled; the features available in both modes will be explained in detail in following chapters.

For a variety of application functions which the device is intended, the outputs can be managed in pairs: in this case, two outputs are coupled to form a Channel. The outputs can be coupled only if they are physically close on the terminal block

> *In order to maintain a consistent naming convention, the outputs are numbered in the same way regardless whether the channel pairing is used or not.*

In order to specify the channel pairing, each input can be configured in two ways: independent (or single) and coupled mode.

- In *independent or single mode*, each output operates independently, has its own parameters and communication objects. This is the mode of operation described so far.
- In *coupled mode*, the two outputs operate logically grouped under the same channel in order to perform a common functionality. Accordingly, these inputs operate on shared communication objects.

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

12.2.1 Output features in independent mode

In the simplest case there is only one communication object per channel, “On-Off command”, that switches each channel output directly with a message.

By setting the device parameters, it is possible to activate additional features, most of which will also affect the outputs. These features are:

- Relay inversion: allows to short contacts on the Off logical value and disconnect on the On value.
- Feedback: sends message on each switching operation or cyclically each period of time
- Time delay block: allows to perform the actual relay switch with a programmable delay. It is available (with separate delay settings) both for the On-Off and for the Off-On transition.
- Staircase function: performs a retriggerable time period activation of an output.
- Logic function: allows to compute the output value as a logic function based on the value of several communication objects.
- Lock and Force: these functions can temporarily force the output to fixed values and also perform high priority switching operations.

- Scene management: - allows to save and recall a combination of state and values with a single telegram.
- Operating hours / Energy consumption counter: allows a limited tracking of energy consumption by accumulating “On” period durations over time.

The most significant functional blocks for an output in independent operation are described in the following scheme.

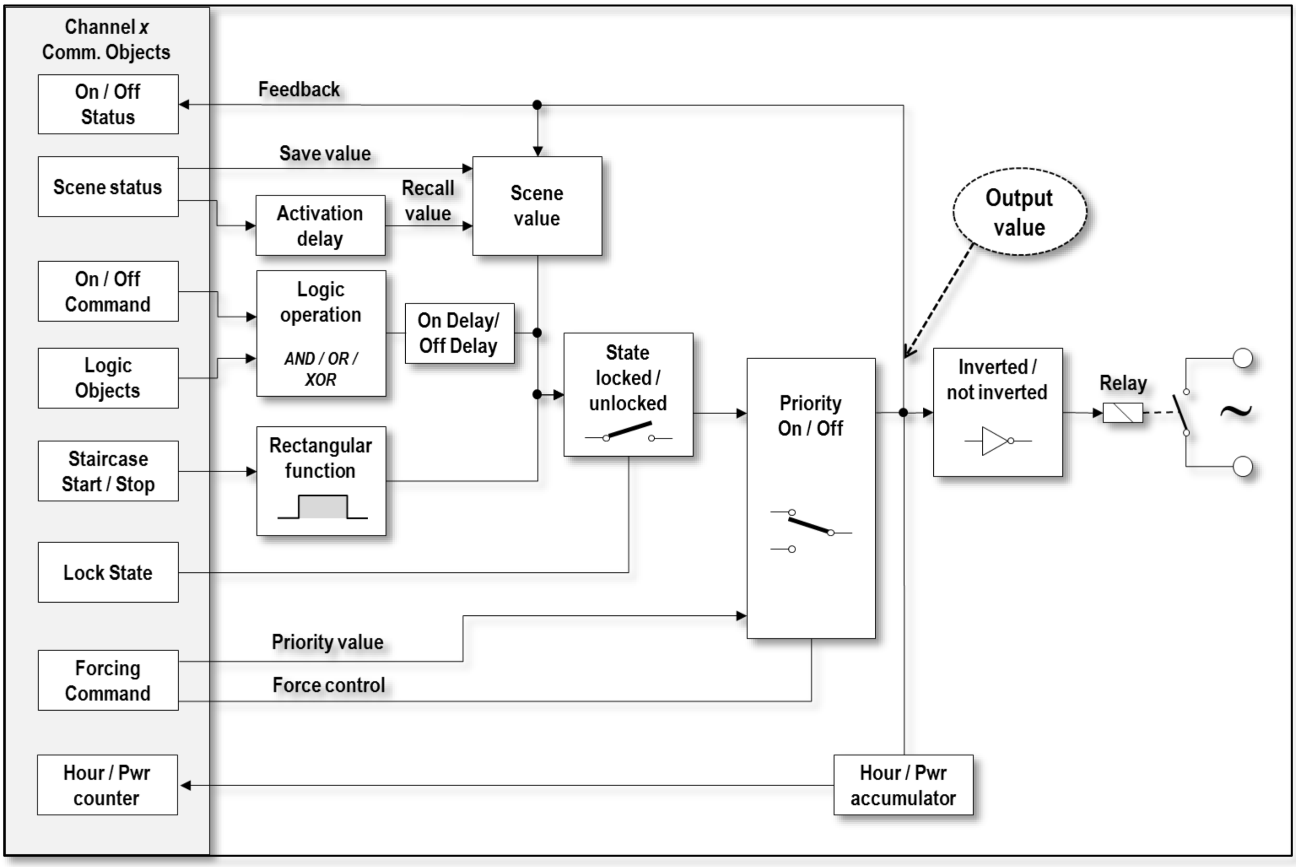


Figure 17 - Functional blocks - independent mode (referring to the single output)

It must be noted that, as can be seen from the above diagram, the different features of the output channel can be activated and operated in parallel at the same time; the configurator has the responsibility of taking care that any interference between different functions does not produce unintended effects on the way device outputs are managed.

- *Relay inversion*

This feature inverts the status of the physical contact of a channel with respect to the exit status.
NOTE: regardless of the “inversion” parameter setting, the following sections will always take “on” and “off” to be a reference to the logical status of the output, not the status of the relay contact switch.

- *Feedback*

When feedback is enabled, a communication object corresponding to the status of the output is made available for reading by other devices on the bus. This object carries the actual state of the logic output, which is likely to be different from the command value because it includes the effect of all additional functions which may be active at the time.

If this communication object is defined, it is also transmitted on every state change, so it can be used to trigger events following the actual state change of an output; it is also possible to configure transmission at regular intervals.

Feedback telegrams are not sent if the outputs are operated manually.

- *Time delay*

The actual change of state of an output can be set to take place after a configurable delay, from the change of the value of the corresponding communication object; this applies both to the on-off and the off-on transitions, each with its individually configurable delay value (T_{on} and T_{off} respectively).

These delays are applied to the switching command via direct and / or logical objects, but not to those caused by other functions (eg. Stair light or scenario).

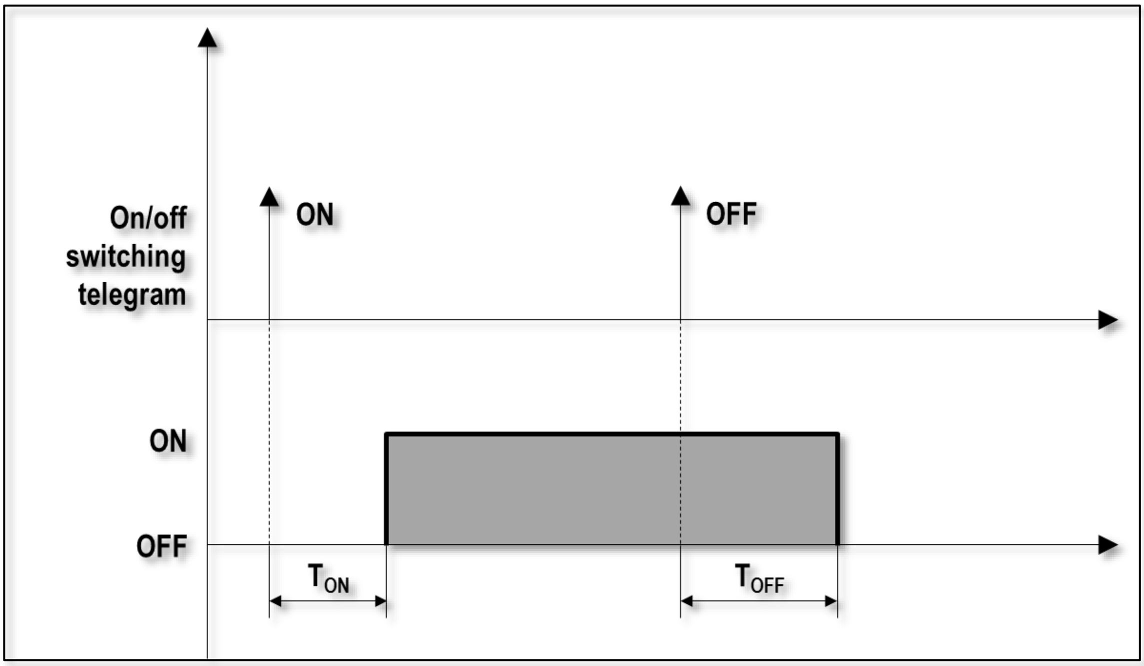


Figure 18 – Time delay

- *Staircase light function*

This function is intended to provide a simple and flexible way to manage the switching of staircase lights. These have following peculiar requirements:

- The light is activated by a “start” command (e.g. through a pushbutton or a presence sensor), and normally remain lit for a programmed time duration;
- There is a provision to enable a “stop” (Manual Off) command, again through a pushbutton or other events, that allows to switch the light off before the programmed time expires (e.g. because the person who triggered the presence sensor has surely left the building through an exit);
- There is a provision to allow another “start” command (Retriggering), received during activation, to restart the time duration counter;
- A further optional “pre-warning” function allows to briefly switch off the load a certain time before expiration (both times, i.e. pause duration and time before expiration, are configurable) in order to warn the user that the activation time is about to end.

The following figures illustrate the operation of the manual switch off function:

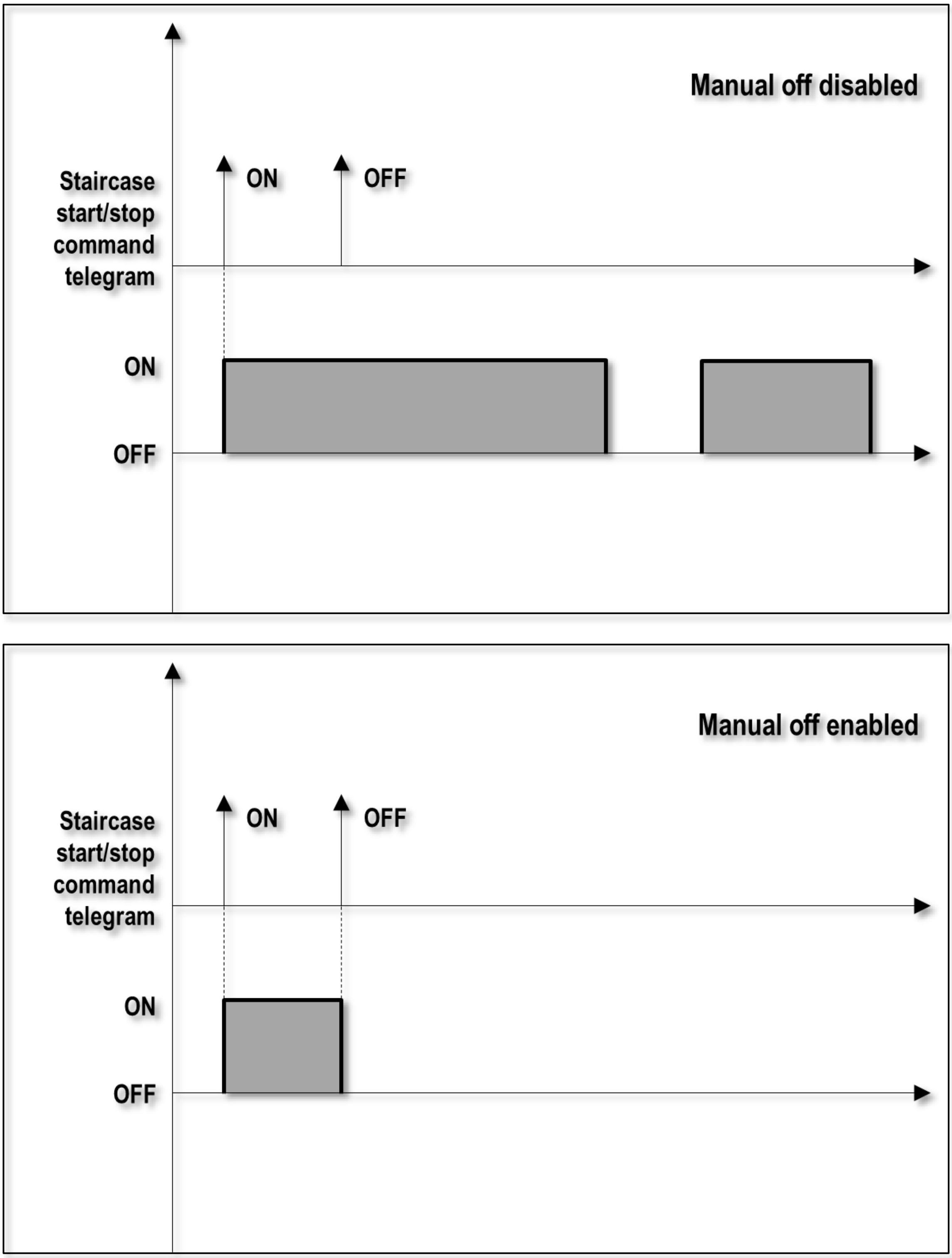

Manual off disabled
Staircase start/stop command telegram
ON
OFF
ON
OFF
Manual off enabled
Staircase start/stop command telegram
ON
OFF
ON
OFF

Figure 19 - Manual switch-off function

The following figures illustrate the operation of the restart function:

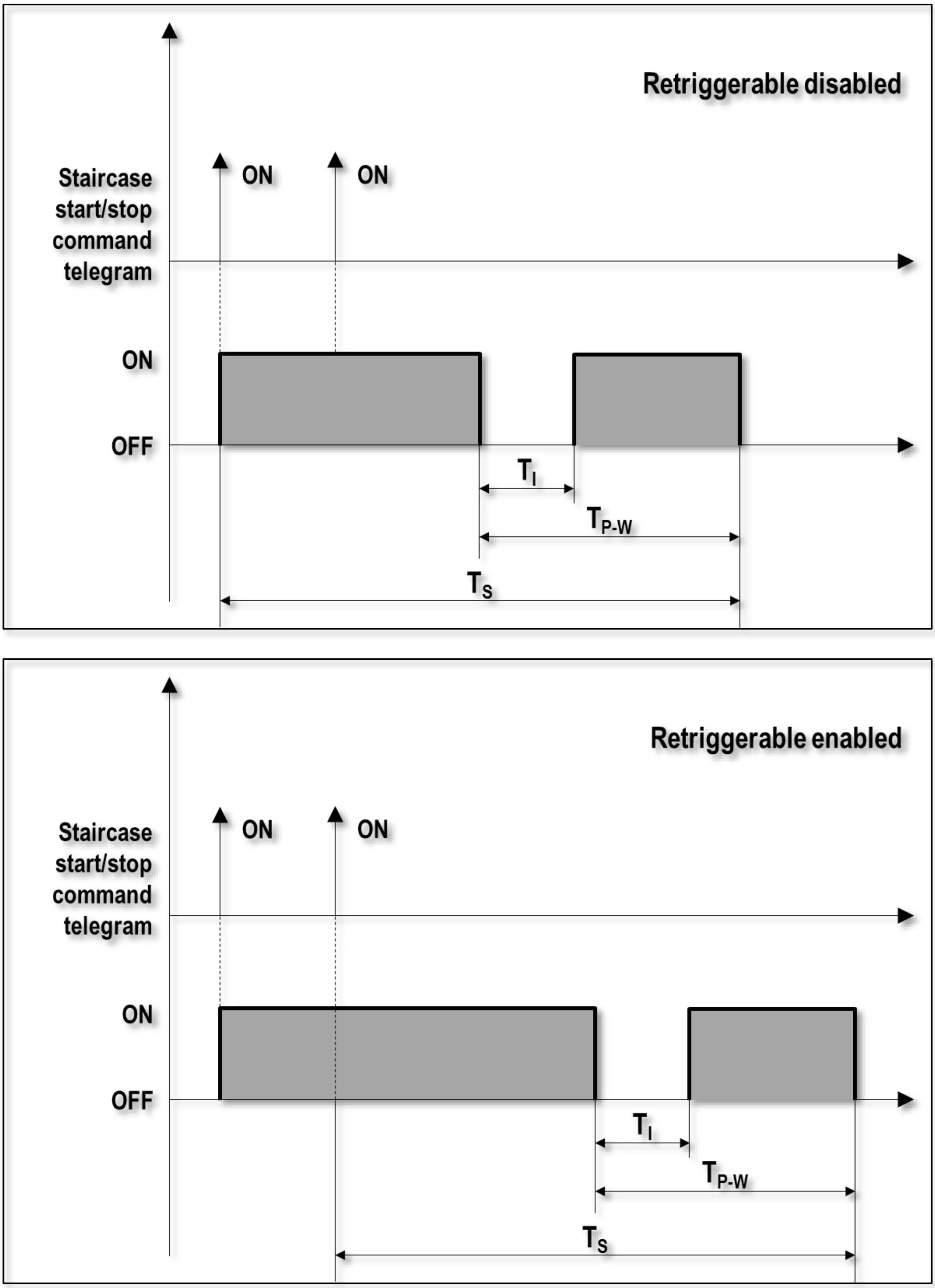


Figure 20 – Retrigger function

The following figures illustrate the operation of the warning function:

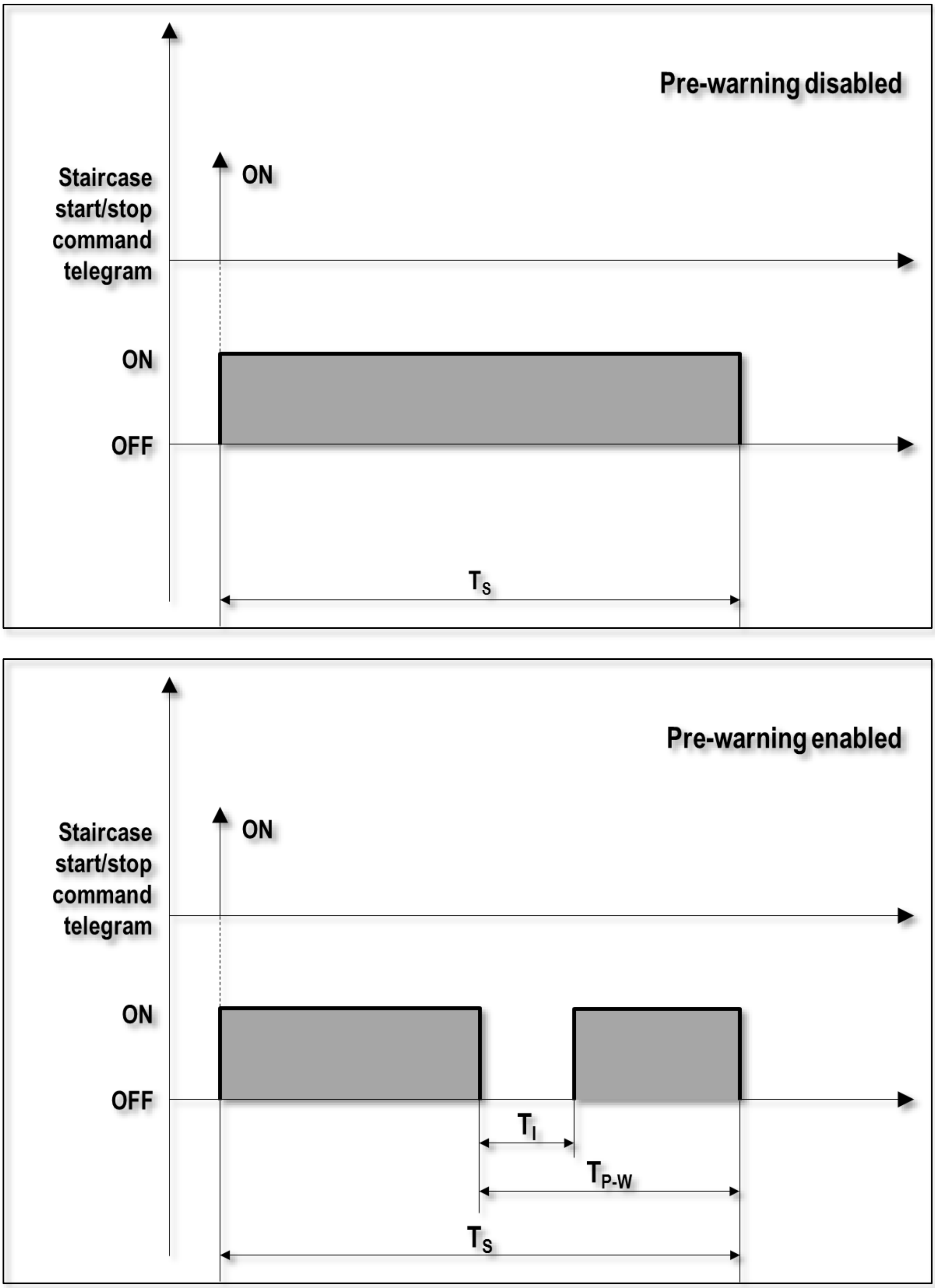


Figure 21 – Pre-warning function

- *Lock Function*

If the locking feature is enabled, the operation of a channel can be inhibited by writing a value in a communication object. The value written is of the KNX type “enable” (active); please beware that the meaning of this value is “activate lock”, which is not to be confused either with “enable locking function” or with “enable output”. The meaning of the value can be optionally inverted through a configuration parameter (an “enable on” value can be interpreted as “lock off”).

A locked output ignores the switching commands that are received for the duration of the lock, thereby maintaining the status it has upon lock entry. The status of the output can be set to a particular value both when the lock is set and when it is released; it is also possible to determine whether the lock status should be maintained or changed on recovery after a bus power-off.

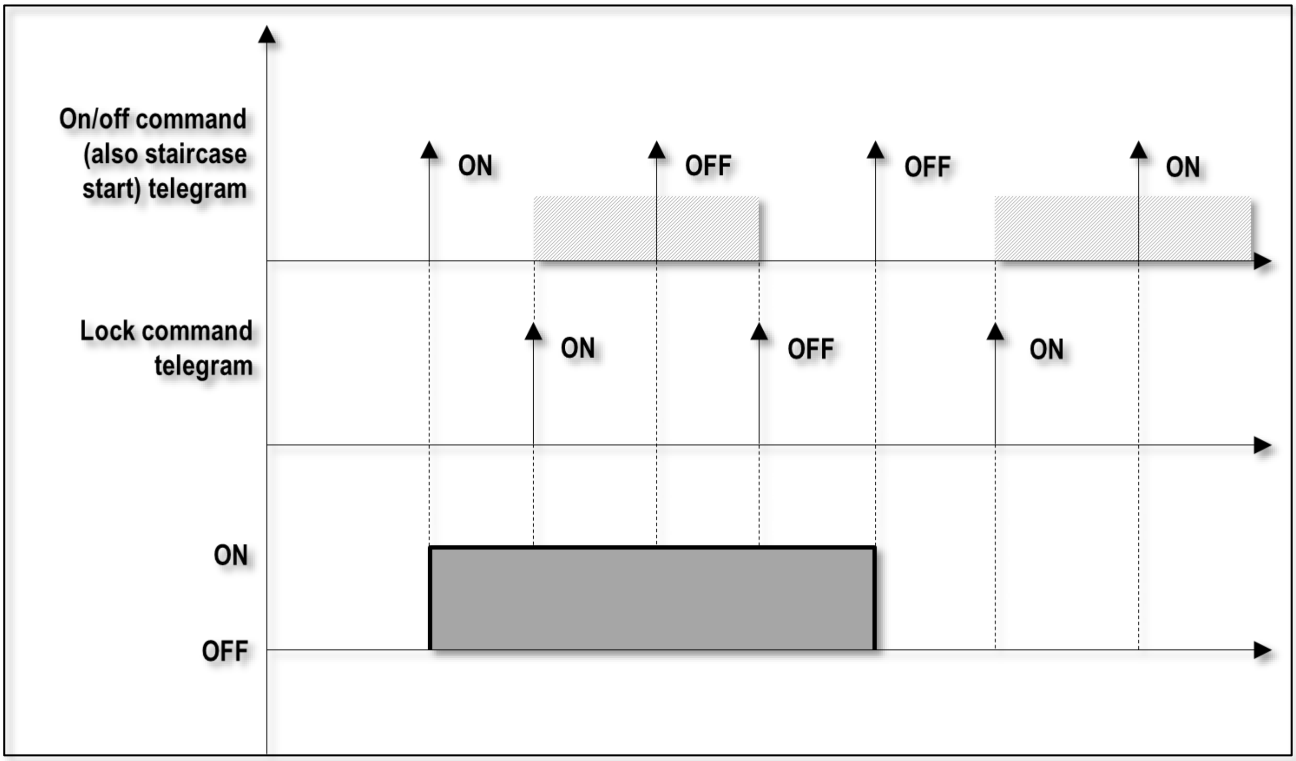


Figure 22 - Lock Function

- *Forcing Function*

The forcing function is very similar to the normal output direct control, but with the particularity of having priority is compared to the value set in an "ordinary" both with respect to the value conditioned by any other function (ie logic functions, timings light stairs etc .)

In addition to force a desired value, it is possible to establish the value that the output must assume both the release of the force, both the power recovery after a lack of voltage on the bus, in case there was an active force at the time of ' break.

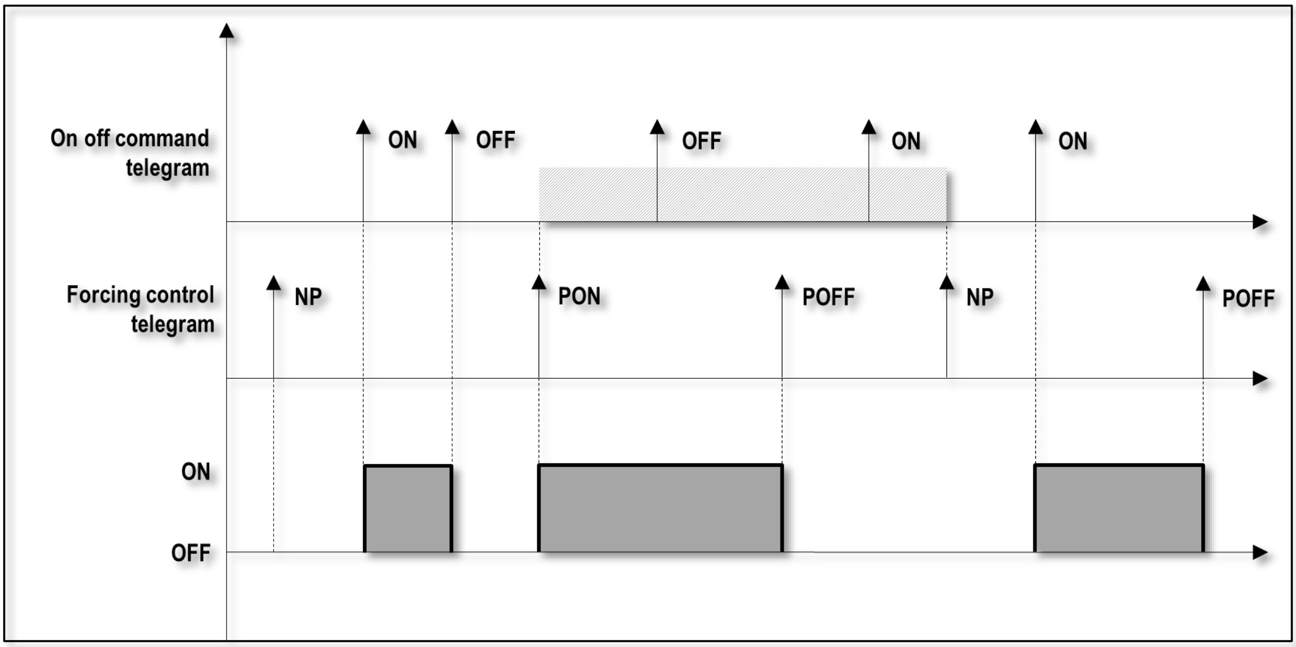


Figure 23 - Forcing Function

The “Force” command has priority over Locking (which acts on the ordinary on-off command); therefore, a locked output can still be operated through “Force” commands.

The KNX command code for the “Force” operation is a 2 bit value; the priority bit determines whether the output value must be forced, in which case the value bit is assigned to the output.

In the figure above, NP means that the priority bit is 0 (No Priority), while the PON and POFF codes indicate the values with priority = 1 and value respectively 1 or 0.

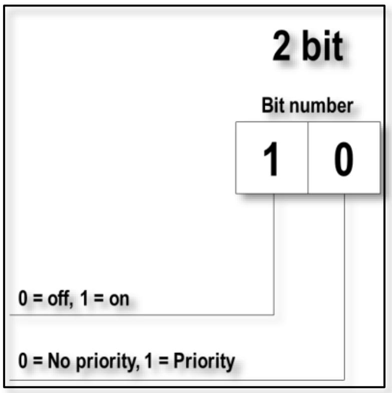


Figura 24 - Force command bits

- *Scenes function management*

Each output can be linked to up to 8 scene codes; when one of these scene codes is recalled through a bus command originated by any controller device, the output will assume a preset value. An additional delay can be defined for the output activation (or deactivation) from the moment the scene code is recalled.

The output value for a scene can either be fixed and chosen in the configuration phase, or it can be defined as reprogrammable through a Scene Learning command.

If this latter option is enabled (for each single output), whenever a Scene Learning command is received on the bus for a specific scene code to which the output has an association, the device will store the current output status value for that scene. This value will then be recalled in subsequent scene activations.

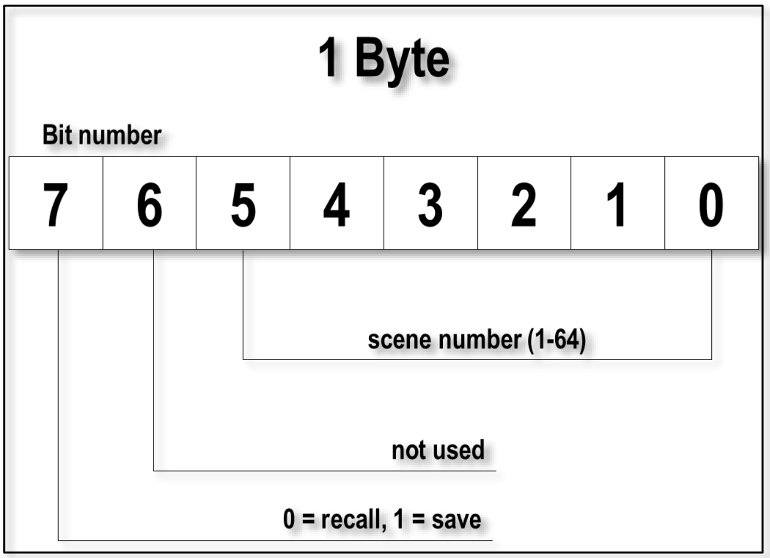


Figure 25 - Scene store / recall command code

- *Operating energy/time counter*

For each output, an activation counter can be associated which accumulates the count of hours that the output passed in the “on” state. In terms of communication objects, this counter has the format of a KNX hour counter, thus it also has a “reset” command ad a “runout” alarm in case the maximum value is overflowed. Both of these commands are constituted by additional communication objects.

The power counter also has an associated KNX object of type "energy meter (kWh)", with its own reset command. A special parameter is used to define a conventional value of electric power in W associated with the load.

Please note that this is not a physical electric power measurement, but just the definition of a proportionality factor between the operating time and the estimated conventional consumption. However, this feature allows to get an indication of maximum consumption for user surveillance, particularly in the case of resistive loads or power fixed as in the case of lights and many other residential or office equipment.

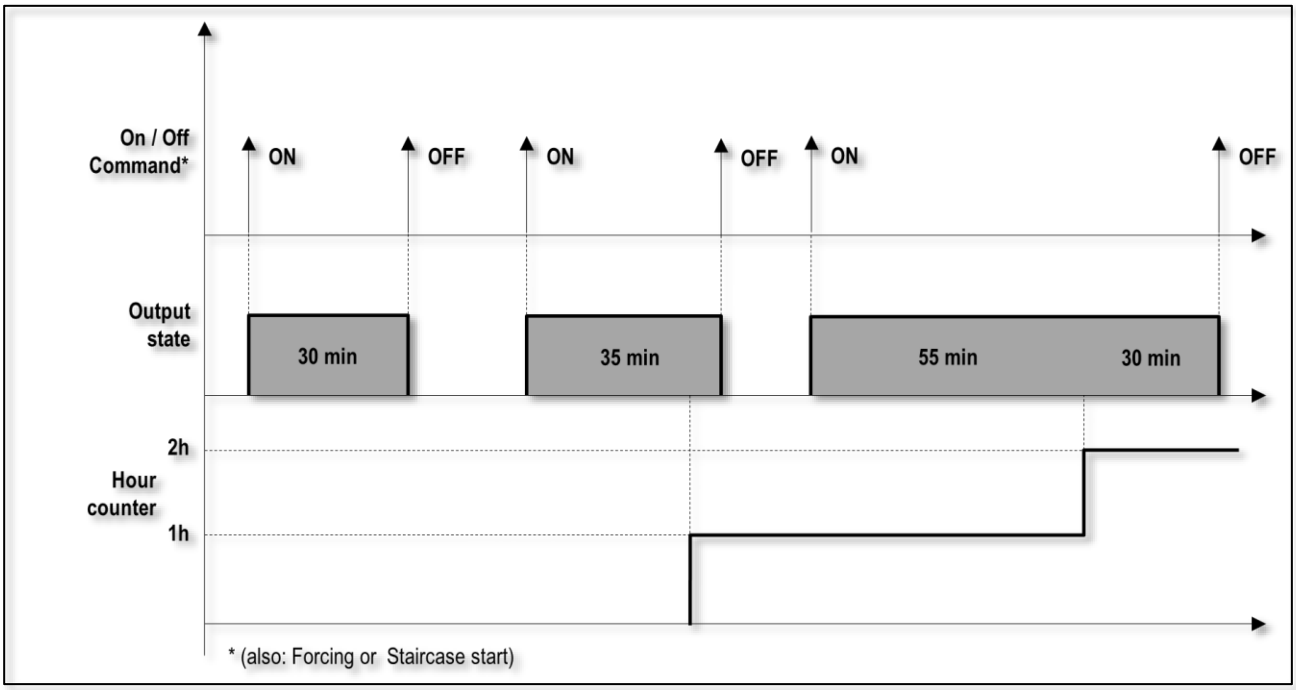


Figure 26 - Operating hours / Energy consumption counter diagram

- *Output restore values*

As already mentioned, the status of the device after some significant events (see “Output restore” paragraph for description) can be assigned to a specific value during the configuration phase.

The available values for the independent outputs restore are:

- On;
- Off;
- no change;
- previous value / state *

(* This option is not available for events "off the bus" and "after downloading")

The difference between *"no change"* and *"previous value / state"* is as follows:

- *"no change"* refers to before the event itself (e.g., for the “bus on” event, an output which was “off” before bus recovery will remain “off” thereafter);
- *"previous value / state"* refers to the condition that is terminated by the event (e.g. for the “bus on” event, an output which was “on” before bus failure will return “on” after bus recovery).

For more details, please refer to the description section of the device settings.

12.2.2 Output features in coupled mode

In coupled mode, output pairs can be used to drive three categories of devices: these are grouped under the denomination of Valve actuators (2- or 3-way), Shutters and Venetian Blinds.
These categories have basically a similar operation mode, that is, they move a physical device from one to another endpoint; this can happen stepwise, with full stroke, or possibly stopping at given intermediate positions. The mentioned actuators, in the order they are listed, could be seen – apart from minor details - an increasingly sophisticated version of the same basic mechanism. Anyway, all three of them are driven through two lines, one for each direction.
For any single channel, one of these three types of behavior can be chosen.
Beside the distinctive features of these categories, there are further features common to all of them, like the locking and forcing functions, meteo alarms and scene management, that will be described below. Some of these features are similar to those described for those of single outputs in independent mode; in these cases, the corresponding sections in the previous paragraph are referenced.

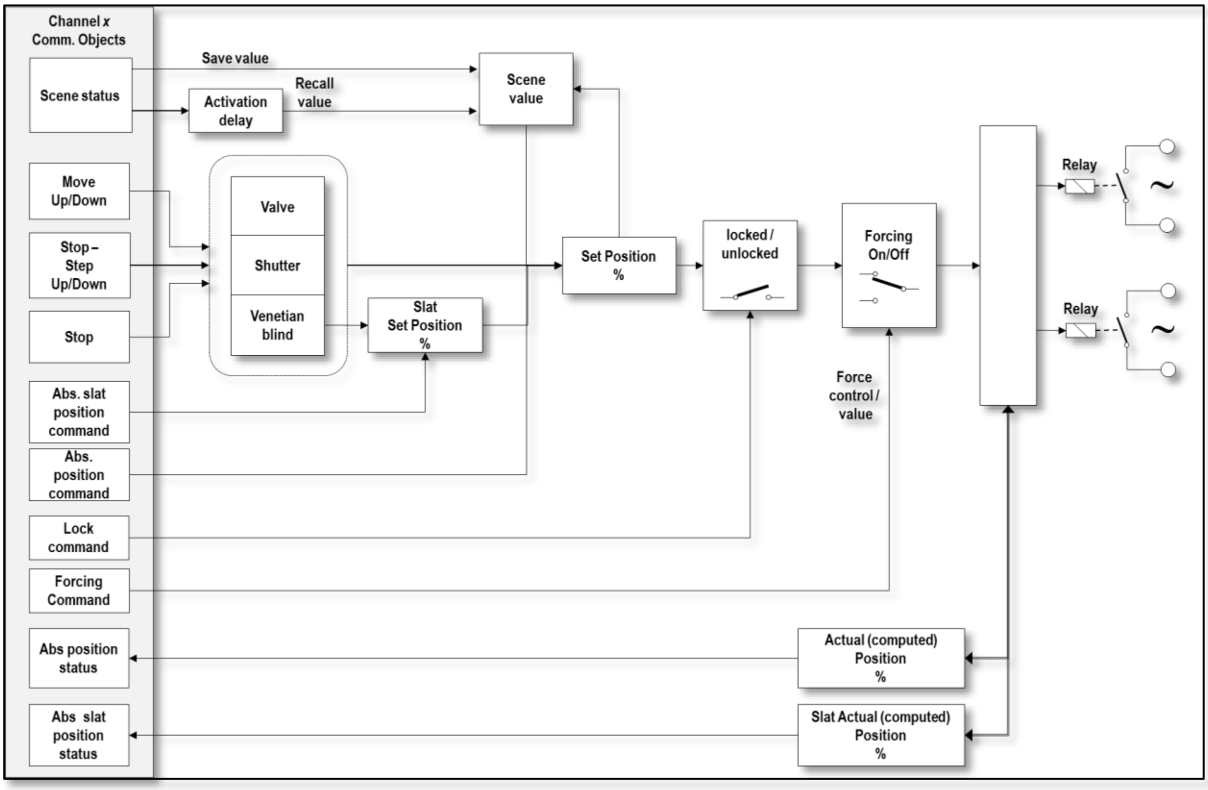


Figure 27 - Functional blocks – Coupled mode (referred to a single output)

- *General information on coupled output control*

The control with coupled outputs is based on three main telegrams, all of which are 1-bit values and thus can convey up to two commands each:

Move Up (Open) / Down (Close)	When the telegram is received, the actuator starts moving all the way towards the specified endpoint.
Dedicated stop	When the telegram is received, the actuator stops any movement and remains in the current position
Stop – Step Up / Down	This command allows a gradual or stepwise movement of the actuator. It actually has a dual purpose: • when the actuator is at rest, it acts similarly to the Move Up/Down command. When the telegram is received, the actuator moves in the specified direction, but just by one “step” (i.e. a length predefined by timing); • when the actuator is moving, it stops in the current position.

In most actual systems, as also defined by KNX standards, the difference between “Move” and “Step” (aside from the additional “Stop” function of the latter) is just the length of the time interval: in principle, a “Move” command is just a “Step” command which duration is guaranteed to be long enough to allow the actuator to reach the endpoint.

Looking at it another way, the same timing that in the case of stepping defines the Step duration, in the case of the Move command has the role of a timeout that deactivates the output when it is no longer necessary to drive it (of course there are different parameters for these timings). Actuators, anyway, will normally have electrical end switches that will prevent overloads caused by unnecessarily applying power when at the endpoints.

Since no position feedback is available from the mechanical actuator, the shutter position is determined through movement timing: given the full-scale movement time value (i.e. the exact time the shutter / actuator takes to move from one endpoint to the other), a partial movement expressed in a percent fraction of the full stroke will then correspond to the same fraction of movement time. The device keeps an internal position counter which is realigned whenever a full Move up/down command is issued.

In order to have the correct timing to be applied to output switches, the full-scale movement time value must be set through a parameter.

This is just a basic generic description; actual actuator types may not have the same control possibilities (e.g. they might not be capable of stopping in positions other than the two endpoints) or they may have more options and features. This will be described below in the explanation of specific functions.

- *Valve control*

The valve control is the most basic of the three controls available; the control can be configured for both 2- and 3-way actuators.

A 2-way actuator has two command lines: one line brings the valve in one (say "open") position, while the other moves it the opposite way. There are no intermediate rest positions.

A 3-way actuator works almost the same way, except that the movement between the two endpoints is gradual (and slower); therefore, if both command lines are de-energized while the actuator is travelling between the endpoints, it will stop in the current intermediate rest position.

Since a 3-way actuator works exactly like a Shutter control, which is described in the next section, only the 2-way actuator will be described here.

This control supplies the three basic commands already described in the "basics" section; however, the "Stop/Step" command is provided because it is required by KNX specifications, but it has no practical effect because no gradual movement is possible. The Stop command also has no practical effect on the movement (other than de-energizing both outputs immediately).

The standard way of driving a 2-way valve requires therefore just the "Move" command to be issued with either direction set in order to switch the valve to either position.

An additional communication object is available to query the movement status of the actuator (i.e. it indicates whether the valve is moving or at rest).

- *Shutters / blind control*

The shutter control is the most similar to the typical control described in the "basics" section; the description of its operation also applies exactly to the 3-way valve.

This control supplies the three basic commands already described in the "basics" section; however, the "Stop/Step" command is provided because it is required by KNX specifications, but only acts when used as a "Stop" command (it has no effect when the actuator is not moving).

The standard way of driving a shutter channel is therefore the following:

- issue the "Move" command with either direction set, in order to start the motion of the shutter;
- either leave the shutter to arrive to the endpoint (the output will be deactivated after a timeout anyway, see below) or issue either a "Stop" or a "Step/Stop" command as soon as the shutter has reached the desired intermediate position.

In order to better exploit the possibility of intermediate positioning, this control has additional ways to specify the actuator position:

- the position can be specified as "absolute position" (in percentage); a feedback value for the actual current position and a telegram of "valid position" (setpoint reached) are also available;
- if enabled, a dimmer-type control for the position is also available, as illustrated in figure below. Please refer to the parameter description section for more details.

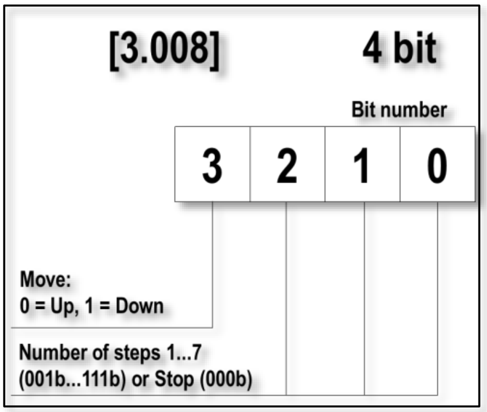


Figure 28 – Dimmer-type blind control

As already mentioned, the full-scale movement time value must be set; there are two parameters for this purpose, one for the upward and one for the downward direction. Times in two directions may be different for mechanical reasons (e.g. heavy shutters) or functional reasons.

The time amount to be specified is the actual and exact stroke time frome one endpoint to another; this will be used to compute the timings for the requested movement stretches. If a movement must be effected that guarantees that the endpoint is surely reached, its duration will be set to 120% of the specified value.

Another parameter which must be defined for the shutter movement is the reversion pause time, i.e. a pause to be made when a movement command in one direction is issued while the shutter is moving the opposite direction. This is mainly made to allow the shutter to correctly stop without excessive strain on mechanical organs.

- *Venetian blind control*

The Venetian blind has the same features as the Shutter control, but with a few additional parameters dedicated to the management of slats (or louvers).

In terms of available commands and parameters, Venetian blinds differ from Shutters for the following features:

- the "Step" command is now meaningful. A step movement is referred to the slats (not to the blinds panel opening); there is a corresponding parameter to define the step time, i.e. the activation time for the outputs corresponding to the movement of a desired step;
- a further set of communication object for "absolute position", "absolute position status" and "Valid position" is available for slats;
- a further dimmer-type control is also available for the slats.

Since slats also have their own absolute positioning feature, a parameter for the total movement time of the slats, similar to the one defined for the blinds, is also provided (but in this case common to both directions, since little or no mechanical asymmetry is to be expected). An internal position counter, similar to the one for the blinds or shutter position, is managed to guarantee the best possible precision in positioning.

Standard blinds' actuators control both blind and slat movement through only two interface lines, the same as shutters discussed in previous paragraph; in order to achieve control of both movements, they are driven as described below. Please bear in mind that this is a principle description of a simplified, albeit realistic, mechanism just for illustration purpose; actual devices may employ different or more sophisticated solutions to realize the same functionalities.

As a general description, each of the driving lines (for respectively upward and downward movement) of the actuator motor directly moves the blind panel towards the corresponding direction. In doing so, the slats are "dragged" in the same direction as the panel (i.e. opening or closing) until they reach their fully open or fully closed position.

We first assume that the blinds start in fully closed position. Activating the "open" line, the motor starts to drag the blinds' array upwards; the slats also move towards the open position. Once these have reached their endpoint, the further action of the motor just continues to lift the blinds.

Assuming now that the blind is stopped halfway, we have a partially open blind with fully open slats; we may naturally continue from here all the way until fully open. If we now activate the downward driver line, though, the slats are moved towards the closed position while the blinds' panel begins to move. The slats are eventually fully closed and the blinds continue to move downwards.

If the activation time of the downward driver line was brief, i.e. not long enough to have the slats span all the way to the closed position, we would obtain a situation where the blind has moved down slightly, but the slats are in an intermediate position; in fact, by alternating the activation of the up / down lines, they can be brought in any desired intermediate position

The following picture illustrates how the blinds react to a command sequence:

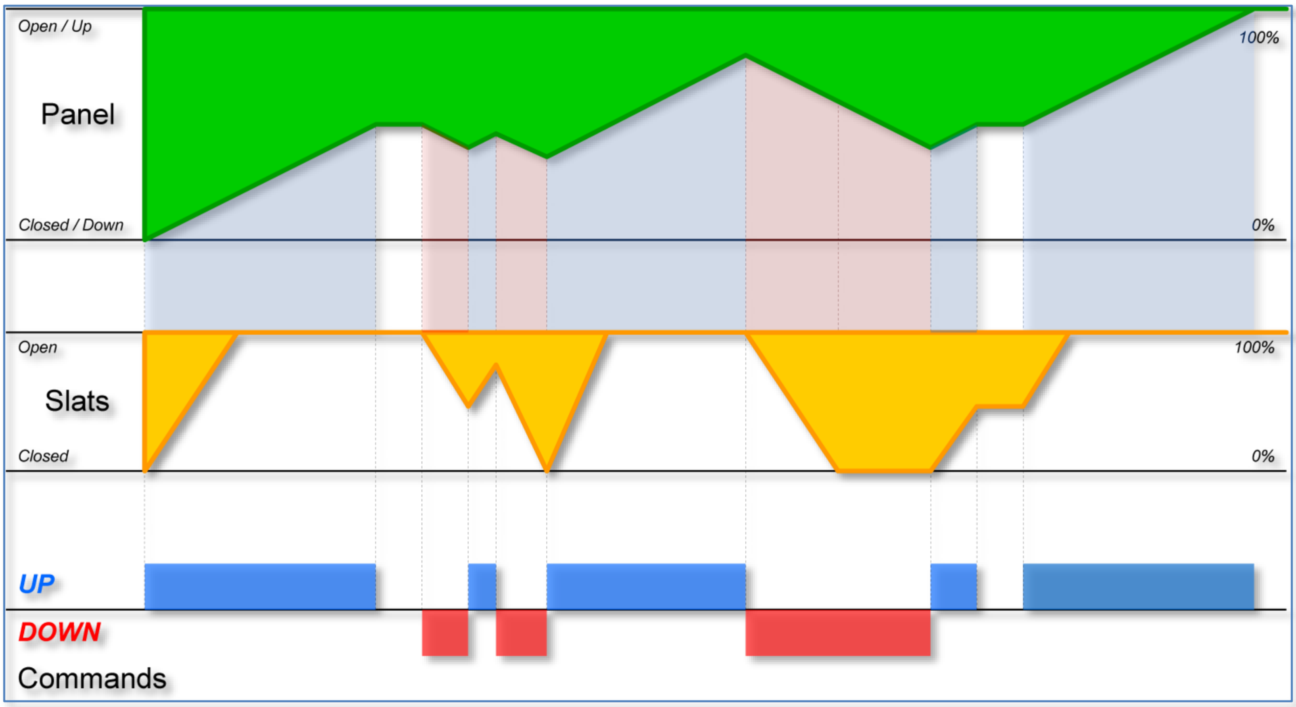


Figure 29 - Venetian Blind diagram

As shown in Figure 29, the slats cannot be moved independently from the blinds' array, i.e. small drive pulses do move the slats as desired but also modify the blinds' position slightly. In order to compensate for this effect and achieve a slat movement without changing the blinds' position (unless temporarily), a "recovery" movement is effected, much like the backlash recovery in automated tools.

This recovery works as follows. Let's assume for example that we would like to lower (close) the slats starting from a 50% position to a 70% position. When the downward line is activated, the blinds' panel is also lowered a little (length "L1" in the picture below). The actual movement is therefore corrected as illustrated in the second part of the picture (which is shown from the original starting position for clarity's sake).

The blinds are initially raised until the slats are fully open (0%, length L2), and then further to compensate for the mentioned length L1. After that, the downward line is activated for the time necessary to go from 0% to the desired position (70%). The final result is as expected.

All the lengths (and corresponding movement times) are computed by the device according to the defined time values for full-range movement times for both slats and blinds' panel; both of these times must be configured for the actuator in use as precisely as possible. The compensation mechanism is automatically managed and does not need being accounted for either by the configurator or the final user.

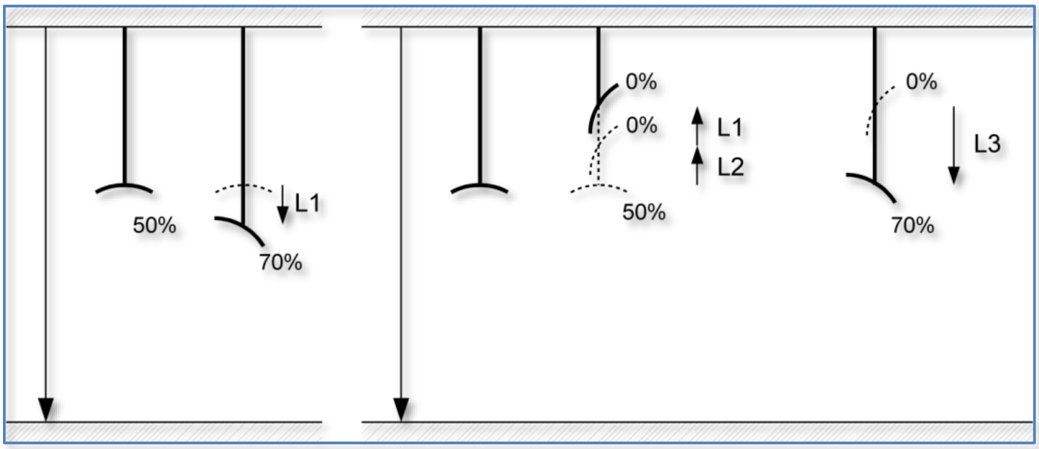


Figure 30 - Compensation of slats movement

- *Lock Function*

The locking feature is similar to the case of independent inputs; the only actual difference is in the wider range of values that can be assigned to the actuator position with respect to simple binary outputs. In particular, these values include stopping current motion, moving the actuator to one of the endpoints, to a programmed position or to the position the actuator had before locking.

Further details can be found in the configuration section.

- *Forcing Function*

The forced control is basically similar to the case of independent inputs; the very same considerations apply as for the case of the Lock function.

- *Meteo alarms*

The Meteo Alarms allow to pre-program an actuator deployment in case of meteorological events detected by a meteo sensor unit (which must be separately purchased and interfaced).

Three types of Meteo alarms can be handled independently, namely for Wind, Frost and Rain. The name is actually just descriptive, since the three alarms are perfectly equivalent and can be used even for different events altogether.

For each of these alarms, a behavior can be defined for the actuator when the alarm is received (go to full "up / open" position, go to full "down / closed" position, or do nothing). Another behavior can be associated to the ceasing of all alarms (all choices above, plus return in the state the actuator had before the alarm).

If more than one alarm becomes active, only the action associated to the first received alarm is executed.

A KNX alarm has an optional "heartbeat" function, i.e. the telegram associated with the alarm can possibly be repeated (and usually it is) at regular intervals; this has a double purpose, in that it assures that an active alarm is not missed if a telegram is lost for whatever reason, and it also confirms that the alarm source is "alive" and that no alarm condition is active if this is the case (alarm telegrams are transmitted with an “Alarm condition clear” value even if the alarm is not active).

For each of the three available alarms, a timeout can be defined for the heartbeat function; if an Alarm information telegram is not received within the timeout duration, the alarm is assumed active and the actuator is correspondingly set. A timeout which occurs when the alarm is already active has no effect.

The heartbeat timeout can of course be disabled; it is important to mention, though, that if it is enabled the device that originates the alarm must be configured for the periodic transmission of alarm information telegrams (furthermore with a period compatible with the timeout interval).

- *Scene management*

Scene management function is similar to the case of independent inputs; the same considerations apply as for the case of the Lock function. The values that can be assigned to a scene are the two endpoints, a specified intermediate position, or a stop (the scene interrupts any current movement).

- *Output restore values*

As already mentioned in earlier paragraphs, the status of the device after some significant events (see “Output restore” paragraph for description) can be defined by configuration.

The values available for restore after system events for coupled inputs are:

- None
- Up / Open
- Down / Close
- Stop
- Move to position

Further details can be found in the configuration section.

12.3 Device settings

This section lists all configurable parameters and describes related communication objects.

Every channel offers the same set of communication objects and parameters, but they may all be independently configured.

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

The device parameters are divided into general parameters and channel-specific parameters.

The general parameters are described in the next paragraphs.

The remaining configuration parameters are divided into two groups: the general configuration parameters of the channels and the parameters for the individual configuration of each channel.

12.3.1 Channels configuration

These settings configure which channels of the device are activated and in which mode.

Activating a channel causes the creation of a few communication objects in the minimal number required to switch the output relays through a bus telegram.

For Output 1B, instead of being explicitly defined, the channel configuration can be copied from Output 1A. If this option is selected, the corresponding channel can be made to perform the exact same kind of function as the source channel.

This allows to spare time in configuring the device, at the same time assuring that there is no inconsistency between two channels that are meant to be configured in exactly the same way.

It must be noted that the configuration copy from the previous output is just a shortcut for the selection of configuration options; it is in no way implied that the two output share any of the involved communication objects. If the configuration of the original output is varied, then so is the “derived” output; in the same fashion, if the original output is disabled, so is also the derived one.

Parameter name	Conditions	Values
Output	-	disabled **2 binary outputs** valve / venetian blind / shutter
	Enable Output Channel.	
Output 1A	Output = 2 binary outputs	disabled / **enabled**
	Enable the first of the two outputs of the channel.	
Output 1B	Output = 2 binary outputs	disabled qualified **copy parameters from output 1A**
	Enable the second of the two outputs of the channel.	
Output - Use	Output = valve / venetian blind / shutter	**valve** shutter venetian blind
	Defines the functionality associated with the pair of outputs.	

Parameter name	Conditions	Values
Three-way mode	Output = valve / venetian blind / shutter Use = valve	disabled / **enabled**
	Configures a valve for three-way mode (same functionality as for a shutter).	

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
Output 1A [1B] - On / Off command	Output = 2 binary outputs	1 bit	CRWTU	[1.001] switch	158, 169
	This communication object is the direct command to the output setting.				
Channel 1 – Move up-down command	Output = valve / venetian blind / shutter	1 bit	C-W--	[1.008] up / down [1.009] open / close	180
	Trigger object for continuous movement: when received, it starts continuous movement in the specified direction.				
Channel 1 - Stop-step up-down command	Output = valve / venetian blind / shutter	1 bit	C-W--	[1.007] step	181
	Trigger object for step movement: when received, and the actuator is at rest, it starts a step movement in the specified direction. If the actuator is not at rest, just stops current movement.				
Channel 1 - Dedicated stop command	Output = valve / venetian blind / shutter	1 bit	C-W--	[1.017] trigger	182
	Stop any ongoing movement when received.				
Channel 1 - Info move	Output = valve / venetian blind / shutter	1 bit	CR-T-	[1.008] up / down	183
	Allows to query the current movement direction				
Channel 1 - valid current abs position	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	1 bit	CR-T-	[1.002] boolean	184
	Signals that the actuator has reached the requested absolute position. *Issued on absolute position movement commands.*				
Channel 1 - Abs [valve / shutter / blind] position command	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	1 bit	C-W--	[5.001] percentage (0..100%)	192
	Sets the target absolute position to reach and starts actuator movement *For the venetian blinds, the position refers to the blinds' panel.*				

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
Channel 1 - Abs [valve / shutter / blind] position status	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	1 bit	CR-T-	[5.001] percentage (0..100%)	193
	Returns the current absolute position of the actuator. *The position is computed from the sequence of requested movements and realigned whenever an endpoint is reached.* *For the venetian blinds, the position refers to the blinds' panel.*				
Channel 1 - Abs slat position control	Output = valve / venetian blind / shutter Use = venetian blind	1 bit	C-W--	[5.001] percentage (0..100%)	195
	Sets the target absolute position for the slats to reach and starts actuator movement.				
Channel 1 - Abs slat position status	Output = valve / venetian blind / shutter Use = venetian blind	1 bit	CR-T-	[5.001] percentage (0..100%)	196
	It returns the current absolute position of the slats. *The position is computed from the sequence of requested movements and realigned whenever an endpoint of the slats' rotation is reached.*				

12.3.2 Independent outputs: Output 1A / 1B configuration

This section lists the configuration parameters of the output channels when used as independent outputs.

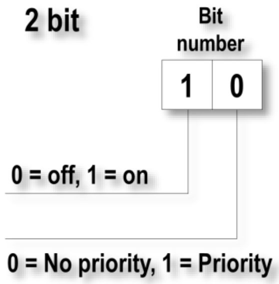
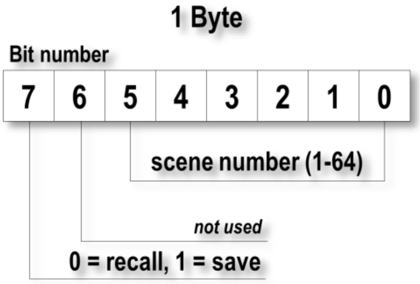
- *Main parameters*

Parameter name	Conditions	Values
Relay operation	Output = 2 binary outputs	**not inverted** reverse
	In the "not inverted" mode, the relay contacts (i.e. the physical output terminals) are shorted when the output is On (active).	
Behavior at bus off	Output = 2 binary outputs	off on **no change**
	It allows to determine the state of the output when the auxiliary power is restored.	
Behavior at bus on	Output = 2 binary outputs	off on **previous state**
	It allows to determine the state of the output after bus recovery.	
Behavior after download	Output = 2 binary outputs	off on **no change**
	It allows to determine the state of the output when the device resumes operation after a new parametrization has been downloaded.	

Parameter name	Conditions	Values
Status feedback telegram	Output = 2 binary outputs	disabled / **enabled**
	Enables or disables the output change notification through a bus telegram. *Updating the object from "ON" to "ON" or from "OFF" to "OFF" has no influence on the switching status feedback.*	
Status feedback telegram - Delay after bus voltage recovery	Output = 2 binary outputs Status feedback telegram = enabled	hh: mm: ss.fff **(00: 00: 03.000)**
	Time after bus voltage recovery before status feedback telegrams begin to be sent. *The delay has no effect on the behaviour of the outputs; only the feedback telegrams are delayed. The outputs can therefore be activated during the delay after a bus voltage recovery.* *During this delay, no feedback telegram will be transmitted even if a switching occurs; the feedback telegram for a switch during the delay period is lost.*	
Status feedback telegram - Transmission cycle time	Output = 2 binary outputs Status feedback telegram = enabled	hh: mm: ss **(00:00:00)**
	Interval between cyclical transmissions. *A zero value (00:00:00) means no cyclical transmission (feedback telegrams are only sent on value change).* *Values less than “00:00:10” (ten seconds) are considered by the firmware in any case as 10 (ten) seconds; the maximum value is 18:12:15.*	
On delay time	Output = 2 binary outputs	hh: mm: ss.fff **(00: 00: 00.000)**
	Delay between the “On” command telegram and the actual output activation. *This time delay does not affect the output of the staircase and forced control functions.* *For the scene function the delay can be set separately.* *Updating the object from "ON" to "ON" or from "OFF" to "OFF" retrigger the delay time.*	
Off delay time	Output = 2 binary outputs	hh: mm: ss.fff **(00: 00: 00.000)**
	Delay between the “Off” command telegram and the actual output deactivation. *Same comments as for the “On delay time” parameter apply.*	
Staircase lighting function	Output = 2 binary outputs	enabled / **disabled**
	Enables or disables the staircase lighting feature. *For further details and parameter descriptions see the corresponding section below.*	
Locking function	Output = 2 binary outputs	enabled / **disabled**
	Enables or disables the capability of locking the input through a remote command. *For further details and parameter descriptions see the corresponding section below.*	
Forcing function	Output = 2 binary outputs	enabled / **disabled**
	Enables or disables the capability of forcing the input through a remote command. *For further details and parameter descriptions see the corresponding section below.*	

Parameter name	Conditions	Values
Forcing operation - Behavior end forced control	Output = 2 binary outputs Forcing function = enabled	off on **no change** previous value
	Allows to determine the state of the output when the forcing is released.	
Forcing operation - Behavior after bus recovery	Output = 2 binary outputs Forcing function = enabled	off on **no change** previous value
	Allows to determine the state of the output when the device resumes operation after bus voltage recovery. *Please notice that this is the status of the output, not the forcing status: forcing is maintained over bus failure and bus recovery.*	
Scenes function	Output = 2 binary outputs	enabled / **disabled**
	Enables or disables the Scenes function. *For further details and parameter descriptions see the corresponding section below.*	
Operating energy / time counter	Output = 2 binary outputs	enabled / **disabled**
	Enables or disables the Hour / Energy counter function. *For further details and parameter descriptions see the corresponding section below.*	

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
Output 1A [1B] - on / off status	Output = 2 binary outputs Status feedback telegram = enabled	1 bit	CR-T-	[1.001] switch	159, [170]
	Sent at any change of the output state and also periodically, as configured.				
Output 1A [1B] - Staircase lighting start stop command	Output = 2 binary outputs staircase lighting function = enabled	1 bit	C-W--	[1.001] On / off	160, [171]
	Starts the staircase light timing with an On value. *The timed activation automatically stops at the end of the preset time.* *If “Manual off” is enabled, the communication object will stop the timing with an Off value.*				
Output 1A [1B] - Lock command	Output = 2 binary outputs locking function = enabled	1 bit	C-W--	[1.003] enable	161, [172]
	Inhibits the switching commands for the output when an “enable” telegram is received, and unlocks them when a “disable” telegram is received.				
Output 1A [1B] - Forcing Command	Output = 2 binary outputs forcing function = enabled	2 bits	C-W--	[2.001] control switch	162, [173]

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
	Allows to force the status of an output. *It is composed of 2 bits: the first one is used for the priority value (i.e. defines whether the forcing is in effect, "Priority", or not) and the second one for the imposed value (which is not considered if forcing is not effective).*  2 bit Bit number 1 0 0 = off, 1 = on 0 = No priority, 1 = Priority				
Output 1A [1B] - Scene Number	Output = 2 binary outputs scenes function = enabled	1 Byte	C-W--	[17.001] Scene number [18.001] Scene control	163, [174]
	Allows to recall a scene setting for the status of the output, and to store current status in association to the specified scene.  1 Byte Bit number 7 6 5 4 3 2 1 0 scene number (1-64) not used 0 = recall, 1 = save				
Output 1A [1B] - kWh counter	Output = 2 binary outputs Operating time/energy counter = enabled	4-byte signed counter	CR-T-	[13.013] active energy [kWh]	164, [175]
	Stores the current counter value of the accumulated energy.				
Output 1A [1B] - kWh counter reset command	Output = 2 binary outputs Operating time/energy counter = enabled	1 bit	C-W--	[1.015] reset	165, [176]
	Resets the energy counter to 0.				
Output 1A [1B] - Operating time counter	Output = 2 binary outputs Operating time/energy counter = enabled	2-byte unsigned counter	CR-T-	[7.007] Time [h]	166, [177]
	Stores the current counter value of the accumulated operating time.				
Output 1A [1B] - Operating time counter reset command	Output = 2 binary outputs Operating time/energy counter = enabled	1 bit	C-W--	[1.015] reset	167, [178]
	Resets the operating hour counter to 0.				

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
Output 1A [1B] - Operating time counter runout	Output = 2 binary outputs Operating time/energy counter = enabled	1 bit	CR-T-	[1.005] alarm	168, [179]
	1-bit alarm sent when the time counter reaches the maximum value of 65535 hours.				

12.3.3 Staircase lighting function

Parameter name	Conditions	Values
Staircase lighting time	Output = 2 binary outputs staircase lighting function = enabled	hh: mm: ss **(00:01:00)**
	Duration of staircase lighting time. *This time is the one shown on the time diagram in the descriptive section of this manual as “T_s”*	
Manual off	Output = 2 binary outputs staircase lighting function = enabled	**enabled** / disabled
	When enabled, it allows an “Off” command to terminate the lighting time. *The “Off” command can be sent at any time with the same effect, including when the pre-warning is activated.*	
Retriggerable	Output = 2 binary outputs staircase lighting function = enabled	**enabled** / disabled
	When enabled, it allows a new “On” command to restart the timing. *The “On” command can be sent at any time with the same effect, including when the pre-warning is activated.*	
Pre-warning	Output = 2 binary outputs staircase lighting function = enabled	**enabled** / disabled
	Activates the pre-warning feature. *For a detailed description see the corresponding section of this manual.*	

Pre-warning time	Output = 2 binary outputs
	staircase lighting function = enabled
	Pre-warning = enabled
		hh: mm: ss
		(00:00:10)

	Specifies how much time before the end of the timing a pre-warning light interruption will be carried out. *The time interval specified includes the interruption time.* *The maximum value is 18:12:15.* *This time is the one shown on the time diagram in the descriptive section of this manual as “T_{p-w}”.*	

Interruption time	Output = 2 binary outputs
	staircase lighting function = enabled
	Pre-warning = enabled
		hh: mm: ss.fff
		(00: 00: 00.500)

	Specifies the duration of the pre-warning interruption. *This time is the one shown on the time diagram in the descriptive section of this manual as “T_i”.*	

- The pre-warning time should be shorter than the staircase time ($T_{P-W} < T_S$) and the interruption time shorter than the pre-warning time ($T_I < T_{P-W}$).
- Time delays have no influence on the staircase function (if enabled).
- A staircase timing in progress will be terminated by a reset of the actuator (bus voltage recovery or ETS reprogramming) or by using any function that affects the output (i.e. normal switching, forced control, logic function, scene recall), even if the function does not cause an actual change in the output value.
- On a forced termination, the value of the output remains unchanged; the same that is true also if the termination occurs during pre-warning time.

12.3.4 Locking Function

parameter name	Conditions	Values
Lock device signal	Output = 2 binary outputs Locking function = enabled	**not inverted** / inverted
	It allows to translate a “lock activate” telegram as unlock and vice-versa.	
After bus recovery	Output = 2 binary outputs Locking function = enabled	unlock lock **previous state**
	It defines how to set the lock status after bus voltage recovery.	
Behavior at locking	Output = 2 binary outputs Locking function = enabled	**off** on no change
	It defines how to set the output value when the lock is activated	
Behavior at unlocking	Output = 2 binary outputs Locking function = enabled	off on no changes updated value **value before locking**
	It how to set the output value when the lock is deactivated. • ***Updated value** is the latest one that the output would assume if it had not been locked, i.e. it includes the output value change generated by whatever other function in the meantime.* • ***Value before locking** is the value that the output had before the lock was activated.*	

12.3.5 Scenes function

Parameter name	Conditions	Values
Download overwrites learned behavior	Output = 2 binary outputs scenes function = enabled	No / **yes**
	Defines whether the download of a program on the device should erase and overwrite the stored scene output values previously learned and stored in the device. *When the device is put into operation for the first time, this parameter should be set to "yes" (default value) so that the output is initialized with valid scene values. Otherwise, the values are set to "0" (off) for all scenes.*	
Scene *n*	Output = 2 binary outputs scenes function = enabled	enabled / **disabled**
	Enables or disables a new scene code to be assigned to the output.	
Scene *n* – Scene number	Output = 2 binary outputs scenes function = enabled Scene n = enabled	1 ... 64 **(1)**
	Scene number to be assigned to the output. The output will respond to scene commands that match the specified number.	
Scene *n* – Output behavior	Output = 2 binary outputs scenes function = enabled Scene n = enabled	off / **on**
	(Initial) output value for the selected scene. This value will be possibly overwritten by a scene "store" command if the "Learning mode" option is enabled.	
Scene *n* – Activation delay	Output = 2 binary outputs scenes function = enabled Scene n = enabled	hh: mm: ss.ff **(00: 00: 00.00)**
	Delay between a scene "recall" command and the actual output switching. *The maximum value is 01:49:13.50.*	
Scene *n* – Learning mode	Output = 2 binary outputs scenes function = enabled Scene n = enabled	**disabled** / enabled
	When disabled, the scene "store" commands are ignored and only the output values set in the configuration are used.	

- Each scene recall telegram restarts the activation delay.
- If a new scene recall telegram is received while a delay is active (scene recall not yet executed), the old - and not yet recalled - scene will be rejected and the newest scene value will be in effect.
- The scene recall delay has no influence on the saving of scene values when the learning mode is active.
- If the same scene number is set for several scene entries, only the scene with the lowest entry number (1...8) will be considered. The other internal scenes will be ignored in this case.
- The scene recall can be overridden by a *forced control* or a *lock* function.

12.3.6 Operating energy / time counter

parameter name	Conditions	Values
Output load [W]	Output = 2 binary outputs Operating energy / time counter = enabled	-671088,64 ... +670760,96 **(1000)**
	It defines the nominal rated power to be considered in computing the accumulated power consumption for the load connected to this output. *The total energy consumed [kWh] is calculated as the product of the specified value [W] and the operating hours [h].*	
Energy / time cyclic sending	Output = 2 binary outputs Operating energy / time counter = enabled	hh: mm: ss **(00:00:00)**
	It defines the time interval for the cyclic retransmission of the counter values (both for accumulated time and energy). *A value of zero (00:00:00) disables cyclic transmission.*	

During either ETS programming or bus voltage failure, even if the output is in the "On" position, the counter is stopped.

12.3.7 Coupled outputs configuration

This section lists all detail settings for the output channels when used as coupled outputs.

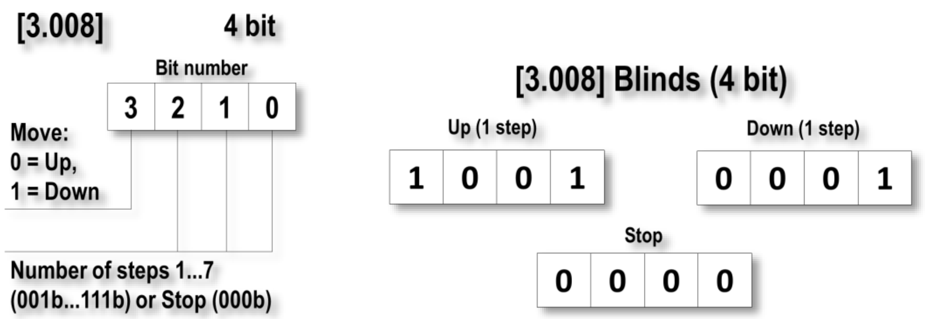
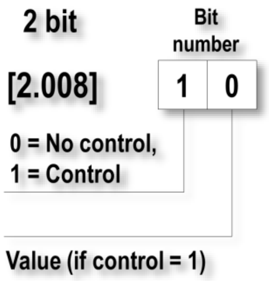
For all entries in this section, the condition "Output = Valve / shutter / Venetian" is implicitly assumed but not shown, for clarity.

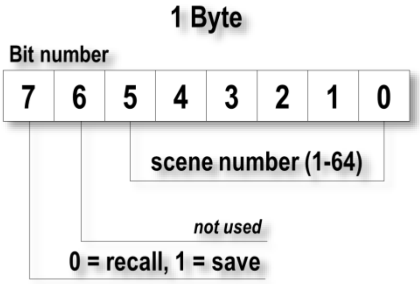
- *Main parameters*

In this section most of the configurable parameters for the output are listed.

Parameter name	Conditions	Values
Reversion pause time	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	0 ... 65535 [milliseconds] **(300 ms)**
	The minimum pause time between contact activation when switching from one output to another.	
Open time	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	hh: mm: ss **(00:00:15)**
	The time for the actuator to run the full stroke between the endpoints, in the opening direction. *It is important that the specification of this time is particularly accurate, since the accuracy of positioning depends heavily on it.*	
Close time	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	hh: mm: ss **(00:00:15)**
	The time for the actuator to run the full stroke between the endpoints, in the closing direction. *It is important that the specification of this time is particularly accurate, since the accuracy of positioning depends heavily on it.*	
Position control with dimmer	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	**no** / yes
	If this option is selected, a dimmer-type communication object is made available for the control of the actuator. It can be used, as an alternative, at the same time as the other standard control mechanisms.	
Slat movement time	Output = valve / venetian blind / shutter Use = venetian blind	hh: mm: ss **(00:00:15)**
	The time for the actuator to run the slats over the full stroke between the endpoints. *Unlike the main panel movement, there are no separate times for the two directions, because no relevant mechanical asymmetry is to be expected.* *It is important that the specification of this time is particularly accurate, since the accuracy of positioning depends heavily on it.*	
Slat step time	Output = valve / venetian blind / shutter Use = venetian blind	0 ... 65535 [milliseconds] **(100 ms)**
	The activation time corresponding to a desired step span for the slats.	
Slats control with dimmer	Output = valve / venetian blind / shutter Use = venetian blind	**no** / yes
	If this option is selected, a dimmer-type communication object is made available for the control of the actuator. It can be used, as an alternative, at the same time as the other standard control mechanisms.	

Parameter name	Conditions	Values
Behavior at bus on	-	**none** up / open down / close stop move to position
	It allows to determine the state of the output after bus recovery.	
Behavior after download	-	**none** up / open down / close stop move to position
	It allows to determine the state of the output when the device resumes operation after a new parametrization has been downloaded..	
Locking function	-	enabled / **disabled**
	It enables or disables the capability of locking the input through a remote command. *For further details and parameter descriptions see the corresponding section below.*	
Forcing function	-	enabled / **disabled**
	It enables or disables the capability of forcing the input through a remote command. *For further details and parameter descriptions see the corresponding section below.*	
Forcing function – behavior end forced control	Forcing function = enabled	**none** up / open down / close stop move to position previous
	It allows to determine the state of the output when the forcing is released..	
Forcing function – behavior after bus recovery	Forcing function = enabled	**not forced** forced up / open forced down / closed stop move to position previous
	It allows to determine the state of the output when the device resumes operation after bus voltage recovery.	
Meteo alarms	-	enabled / **disabled**
	Enables or disables the Meteo alarm processing feature. *For further details and parameter descriptions see the corresponding section below.*	
Scene function	-	enabled / **disabled**
	Enables or disables the Scene function. *For further details and parameter descriptions see the corresponding section below.*	

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
Channel 1 - Dimmer blind position command	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled position control with dimmer = yes	4-bit controlled	C-W--	[3.008] blind control	191
	It allows to command the actuator through a dimmer-style command.  [3.008] 4 bit Bit number 3 2 1 0 Move: 0 = Up, 1 = Down Number of steps 1...7 (001b...111b) or Stop (000b) [3.008] Blinds (4 bit) Up (1 step) 1 0 0 1 Down (1 step) 0 0 0 1 Stop 0 0 0 0				
Channel 1 - Dimmer slats command	Output = valve / venetian blind / shutter Use = venetian blind Slats control with dimmer = yes	4-bit controlled	C-W--	[3.008] blind control	194
	It allows to command the slats position through a dimmer-style command. *See previous entry for bit field details.*				
Channel 1- Lock Command	Locking function = enabled	1 bit	C-W--	[1.003] enable	185
	It inhibits the switching commands for the output when an "enable" telegram is received, and unlocks them when a "disable" telegram is received.				
Channel 1 - Forcing command	Forcing operation = enabled	2 bits	C-W--	[2.008] 1 direction control	186
	Allows to force the status of an output pair. The command is a "direction control" telegram, which can force movement in one direction, the other, or release forcing.  2 bit Bit number [2.008] 1 0 0 = No control, 1 = Control Value (if control = 1)				
Channel 1- Wind alarm	Meteo alarms = enabled	1 bit	C-W--	[1.005] alarm	187
	If this alarm is enabled, writing an active alarm value here will set the corresponding alarm condition; the alarm will be released by writing a "clear alarm" value. *If the heartbeat timeout" is set, even in absence of an alarm condition, the "clear alarm" value must be regularly written at intervals not higher than the timeout period.*				

Object name	Conditions	Size	Flags	DPT	Comm. Obj. N.
Channel 1 - Frost alarm	Meteo alarms = enabled	1 bit	C-W--	[1.005] alarm	188
	Same considerations as for previous alarm apply.				
Channel 1 - Rain alarm	Meteo alarms = enabled	1 bit	C-W--	[1.005] alarm	189
	Same considerations as for previous alarm apply.				
Channel 1 - Scene Number	Scene function = enabled	1 Byte	C-W--	[17.001] Scene number [18.001] Scene control	190
	Allows to recall a scene setting for the status of the output, and to store current status in association to the specified scene. 				

12.3.8 Locking Function

Parameter name	Conditions	Values
Lock device signal	Locking function = enabled	**not inverted** / inverted
	It allows to interpret a "lock activate" telegram as unlock and vice-versa.	
After recovery bus	Locking function = enabled	unlock lock **previous state**
	It defines how to set the lock status after bus voltage recovery.	
Behavior at locking	Locking function = enabled	**none** up / open down / close stop move to position
	It defines how to set the output value when the lock is activated.	
Behavior at unlocking	Locking function = enabled	**none** up / open down / close stop move to position previous
	It defines how to set the output value when the lock is deactivated.	

12.3.9 Meteo alarms

Parameter name	Conditions	Values
Reaction to [Wind / frost / rain]	Meteo Alarms = enabled	**none** up / open down / close
	It defines the position to be reached by the actuator when the alarm is active.	
[wind / frost / rain] heartbeat timeout	Meteo Alarms = enabled	0 ... 65535 [minutes] **(10 Min.)**
	It defines the timeout for the alarm heartbeat. *If a heartbeat timeout is set, the alarm telegrams are required to be sent at regular intervals (shorter that the specified timeout), even with the alarm is not active, in order to be sure that the alarm communication is effective. If a “no alarm” telegram is not received in time, the alarm condition is set.* *A timeout value of zero (0) disables the heartbeat monitoring function.*	
End of alarm action	Meteo Alarms = enabled	**none** up / open down / close previous
	It defines the position to be reached by the actuator when the alarm ceases.	

12.3.10 Scenes function

Parameter name	Conditions	Values
Download overwrites learned behavior	scenes function = enabled	No / **yes**
	It defines whether the download of a program on the device should erase and overwrite the stored scene output values previously learned and stored in the device. *When the device is put into operation for the first time, this parameter should be set to "yes" (default value) so that the output is initialized with valid scene values. Otherwise, the values are set to “0” (off) for all scenes.*	
Scene *n*	scenes function = enabled	enabled / **disabled**
	It enables or disables a new scene code to be assigned to the output.	
Scene *n* - Scene number	scenes function = enabled Scene *n* = enabled	1 ... 64 **(1)**
	Scene number to be assigned to the output. The output will respond to scene commands that match the specified number.	
Scene *n* - output behavior	scenes function = enabled Scene *n* = enabled	stop **fully opened** fully closed move to position
	(Initial) output value for the selected scene. *This value will be possibly later overwritten by a scene “store” command if the “Learning mode” option is enabled.*	

Parameter name	Conditions	Values
Scene *n* - Scene position	scenes function = enabled Scene *n* = enabled Output behavior = move to position	*(cursor control 0 ... 100%)*
	Absolute position value for the blinds for the selected scene. *This value will be possibly later overwritten by a scene "store" command if the "Learning mode" option is enabled.*	
Scene *n* - Scene slat position	scenes function = enabled Scene *n* = enabled Output behavior = move to position Use = venetian blind	*(cursor control 0 ... 100%)*
	Absolute position value for the slats for the selected scene. *This value will be possibly later overwritten by a scene "store" command if the "Learning mode" option is enabled.*	
Scene *n* - Activation delay	scenes function = enabled Scene *n* = enabled	hh: mm: ss.ff **(00: 00: 00.00)**
	Delay between a scene "recall" command and the actual output switching. *The maximum value is 01:49:13.50.*	
Scene *n* - learning mode	scenes function = enabled Scene *n* = enabled	enabled / **disabled**
	When disabled, the scene "store" commands are ignored and only the output values set in the configuration are used.	

i

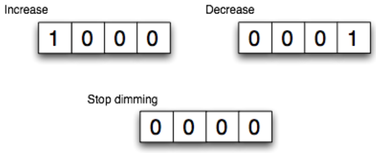
- Each scene recall telegram restarts the activation delay.
- If a new scene recall telegram is received while a delay is active (scene recall not yet executed), the old - and not yet recalled - scene will be rejected and the newest scene value will be in effect.
- The scene recall delay has no influence on the saving of scene values when the learning mode is active.
- If the same scene number is set for several scene entries, only the scene with the lowest entry number (1...8) will be considered. The other internal scenes will be ignored in this case.
- The scene recall can be overridden by a forced control or a lock function.

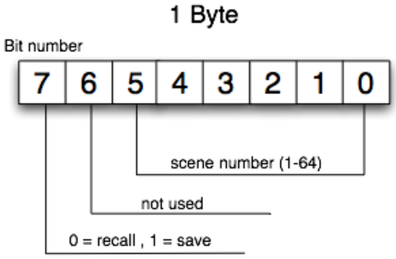
Appendix

A.1 Summary of KNX communication objects by function

Below is a list of the KNX communication objects, along with the related Point Data Types (DPT) defined by the application program, depending on the selected configurations.

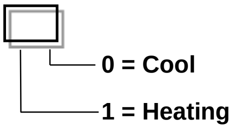
The list is generically ordered by function.

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Input xx - switching command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) type = dimming	1 bit	C-WTU	[1.001] switch	11, 28 (1, 2) 45, 62 (3, 4)
	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 xx – Dimming up / down / stop command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) type = dimming	4 bits	C--T-	[3. *] 3-bit control	12, 29 (1, 2) 46, 63 (3, 4)
	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 1 0 0 0 Decrease 0 0 0 1 Stop dimming 0 0 0 0 *Increase/decrease values are sent when a long press action occurs and stop value on press release.* *The value sent can be a fixed value or it can be toggled at each input activation.*				
Input xx - Dedicated stop command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) type = shutter or venetian blind	1 bit	C--T-	[1.017] triggers	11, 28 (1, 2) 45, 62 (3, 4)
	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.				

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Input xx - Stop – step up/down command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = shutter or venetian blind venetian blind mode = enabled	1 bit	C--T-	[1.007] step	14, 31 (1, 2) 48, 65 (3, 4)
	Move shutter in the fully open or closed position. The object is sent to the end of a long pressure.				
Input xx - Move up / down command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = shutter or venetian blind venetian blind mode = enabled	1 bit	C--T-	[1.008] up / down	15, 32 (1, 2) 49, 66 (3, 4)
	Increase or decrease the opening of the blind stepwise. The object is sent on a short press.				
Input xx - Scene Number	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = scene	1 Byte	C--T-	[17 *] Scene number [18. *] Scene control	16, 33 (1, 2) 50, 67 (3, 4)
	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).  1 Byte Bit number 7 6 5 4 3 2 1 0 scene number (1-64) not used 0 = recall , 1 = save				
Input xx - Switching command	Channel x = coupled Type = switching	1-bit	C-WTU	[1.001] switch	11, 45
Input xx - Switching command	Channel x = coupled Type = dimming	1 bit	C-WTU	[1.001] switch	11, 45

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Input xx - Dimming up / down / stop command	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Type = dimming	4 bits	CR-T-	[3.*] 3-bit control	12, 46
	See notes for independent or single input.				
Input xx - Dedicated stop command	Channel x = coupled Type = shutter or venetian blind Venetian blind mode = disabled	1 bit	C--T-	[1.017] trigger	11, 45
	See notes for independent or single input.				
Input xx - Stop – step up/down command	Channel x = coupled Type = shutter or venetian blind Venetian blind mode = enabled	1 bit	C--T-	[1.007] step	14, 48
	See notes for independent or single input.				
Input xx - Move up / down command	Channel x = coupled Type = shutter or venetian blind	1 bit	C--T-	[1.008] up/down	15, 49
	See notes for independent or single input.				
Input x - Lock Function	Channel x = indep. or single Input 1 = enabled DIN Input xx = enabled (xx = 2,3,4) Lock function = enabled	1 bit	C-W-U	[1.003] enable	2, 19 (1, 2) 36, 53 (3, 4)
Input xx – Switching status *[type]* Object *n*	Input xx = independent or single Type = send values or sequences	variable	C-WTU	variable	3, 20 (1A, 1B) 37, 54 (2A, 2B)
	Up to 8 objects can be defined and associated to the same event. *The listed COs refer to the first of these objects 8 (for each of the inputs); the CO after the first one are sequential. To obtain the CO number for the n-th object, simply add (n-1) to the numbers given.* *Example: the COs associated to input 3 have numbers starting at 37. The number of the 5th CO associated with that input will then be 37 + (5-1) = 41.*				

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Temperature value	Input x = ind. or single, Input 1 = NTC	2 Byte	CR-T-	[9.001] temperature (° C)	90
Temperature Threshold 1 - Switch	Input x = ind. or single, Input 1 = NTC, Threshold 1 = below or above	1 Bit	CR-T--	[1.001] switch	91
Temperature Threshold 2 - Switch	Input x = ind. or single, Input 1 = NTC, Threshold 2 = below or above	1 Bit	CR-T--	[1.001] switch	92
Room temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	93
Humidity (2 bytes, from bus)	Relative Humidity Sensor = enabled, Humidity CO size = 2 bytes	2 Byte	C-WTU	[9.007] humidity (%)	94
Humidity (1 byte, from bus)	Relative Humidity Sensor = enabled, Humidity CO size = 1 byte	1 Byte	C-WTU	[5.001] percentage (0,…, 100%)	95
Antistratification temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	96
Outdoor temperature (from the bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	97
Coil temperature (from the bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	98
Floor temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	99
Flow temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature (°C)	100

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Anticondensation (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	106
Windows contact sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.019] window/door	101
Windows contact sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.019] window/door	102
Presence Sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.018] occupancy	103
Presence Sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.018] occupancy	104
Contact of card holder (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	105
Weighted temperature	Cyclic sending interval ≠ no sending	2 Byte	CR-T-	[9.001] temperature °C	107
Actual setpoint		2 Byte	CR-T-	[9.001] temperature (°C)	115
Manual setpoint		2 Byte	C-W--	[9.001] temperature (°C)	116
Heating / cooling status out	Always visible	1 Bit	CR-T-	[1.100] heating/cooling	108
	The communication object is updated on the bus, on event of internal change processed by the controller. The object is always exposed and contains the information about the current conduction mode of the internal temperature controller. **[1.100] DPT Heat/Cool 1 Bit**  				

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Heating / cooling status in	Thermostat function = both heating and cooling; heating / cooling switchover = from bus	1 Bit	C-W--	[1.100] heating/cooling	109
	The communication object is received from the bus. On switching event, the internal controllers of primary and auxiliary stages (if enabled) switch their operating mode. The actual operating mode is displayed by the corresponding icon.				
HVAC mode in		1 Byte	C-W--	[20.102] HVAC mode	110
	The device receives the operating mode (HVAC mode) from a bus device with the function of supervisor. The operating mode received through this communication object can be later modified by the user (in this case the room thermostat switches to manual mode).				
HVAC forced mode in		1 Byte	C-W--	[20.102] HVAC mode	111
	The communication object allows to receive the operating mode similarly to "HVAC mode in" communication object. The difference is that the operating mode received via this object (with the exception of AUTO) can no longer be modified by the user. The user can modify the operating mode only after "HVAC forced mode in" communication object has sent the AUTO command.				
HVAC mode out		1 Byte	CR-T-	[20.102] HVAC mode	112
HVAC manual mode		1 Byte	C-WTU	[20.102] HVAC mode	113
Chrono active status		1 Bit	CR-T-	[1.011] state	114
Manual / forced setpoint active status		1 Bit	CRWTU	[1.011] state	138
Building protection HVAC mode active		1 Bit	CR-T-	[1.011] state	155
Comfort setpoint (heating)		2 Byte	CRWTU	[9.001] temperature (°C)	117
Standby setpoint (heating)		2 Byte	CRWTU	[9.001] temperature (°C)	119
Economy setpoint (heating)		2 Byte	CRWTU	[9.001] temperature (°C)	121
Building protection setpoint (heating)		2 Byte	CRWTU	[9.001] temperature (°C)	123

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Heating out command	Control type = 2 points hysteresis or PWM Command communication object = separated	1 Bit	CR-T-	[1.001] switch	126
Heating out command	Control type = continuous Command communication object = separated	1 Byte	CR-T-	[5.001] percentage (0..100%)	126
Heating and cooling out command	Control type = 2 points hysteresis or PWM Command communication object = unique	1 Bit	CR-T-	[1.001] switch	126
Heating and cooling out command	Control type = continuous Command communication object = unique	1 Byte	CR-T-	[5.001] percentage (0..100%)	126
Auxiliary heating out command	Auxiliary heating = enabled Command communication object = separated	1 Bit	CR-T-	[1.001] switch	128
Auxiliary heating and cooling output command	Auxiliary heating = enabled Command communication object = unique	1 Bit	CR-T-	[1.001] switch	128
Auxiliary heating disable	Auxiliary heating = enabled Disabled from bus = yes	1 Bit	C-W--	[1.003] enable	130
Room temperature controller status		1 Bit	CR-T-	[1.003] enable	125
Comfort setpoint (cooling)		2 Byte	CRWTU	[9.001] temperature (°C)	118

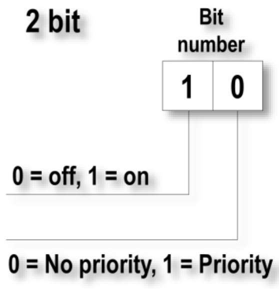
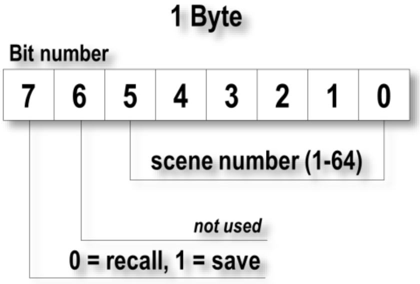
Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Standby Setpoint (cooling)		2 Byte	CRWTU	[9.001] temperature (°C)	120
Economy setpoint (cooling)		2 Byte	CRWTU	[9.001] temperature (°C)	122
Building protection setpoint (cooling)		2 Byte	CRWTU	[9.001] temperature (°C)	124
Cooling out command	Control type = hysteresis 2-point or PWM	1 Bit	CR-T-	[1.001] switch	127
Cooling out command	Control type = continuous	1 Byte	CR-T-	[5.001] percentage (0..100%)	127
Auxiliary cooling output command	Auxiliary cooling = enabled	1 Bit	CR-T-	[1.001] switch	129
Auxiliary cooling disable	Auxiliary cooling = enabled Disabled from bus = yes	1 Bit	C-W--	[1.003] enable	131
Anticondensation alarm	Active anticondensation = enabled	1 Bit	CR-T-	[1.005] alarm	153
Fan continuous speed	Control type = continuous regulation	1 Byte	CR-T-	[5.001] percentage (0..100%)	132
Fan speed 1	Control type ≥ 1 speed	1 Bit	CR-T-	[1.001] switch	133
Fan speed 2	Control type ≥ 2 speeds	1 Bit	CR-T-	[1.001] switch	134
Fan speed 3	Control type = 3 speeds	1 Bit	CR-T-	[1.001] switch	135

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Fan control disable	Disable ventilation from bus = yes	1 Bit	C-W--	[1.002] boolean	136
Fan manual speed		1 Byte	CRWTU	[5.010] counter pulses (0…255)	139
Fan speed status		1 Byte	CR-T-	[5.010] counter pulses (0…255)	140
Fan manual active status		1 Bit	CRWTU	[1.011] state	141
Fan manual speed percentage		1 Byte	CR-T-	[5.001] percentage	156
Fan manual speed off status		1 Bit	CR-T-	[1.011] state	157
Relative humidity setpoint for dehumidification		2 Byte	CRWTU	[9.007] humidity (%)	145
Dehumidification command		1 Bit	CR-T-	[1.001] switch	147
Dehumidification water battery command	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ Cooling type = floor radiant panels or ceiling radiant panels, Relative humidity control ⇒ dehumidification ⇒ dehumidification function = cooling only Subordinated to temperature control = No	1 Bit	CR-T-	[1.001] switch	148

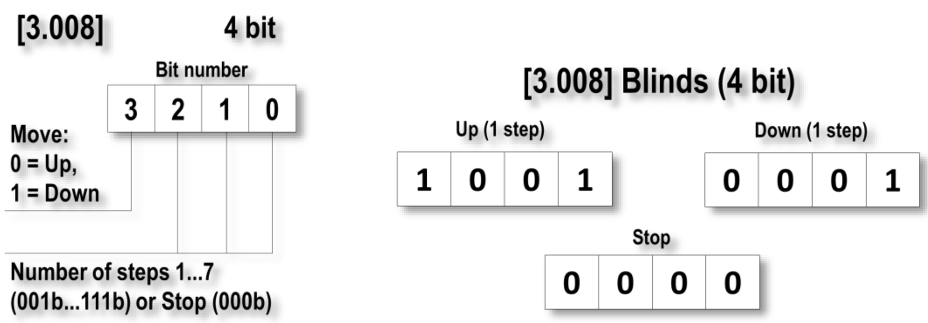
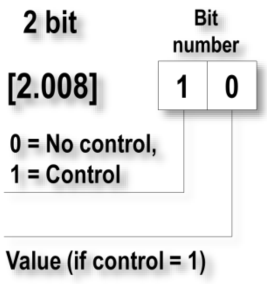
Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Integration dehumidification control	Temperature control ⇒ Settings ⇒ Thermostat function = both heating and cooling, Temperature control ⇒ cooling ⇒ Cooling type = floor radiant panels or ceiling radiant panels, Relative humidity control ⇒ dehumidification ⇒ dehumidification function = cooling only Integration = yes	1 Bit	CR-T-	[1.001] switch	149
	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 from bus = yes	1 Bit	C-W--	[1.002] boolean	150
Relative humidity setpoint for humidification		2 Byte	CRWTU	[9.007] humidity (%)	146
Humidification command		1 Bit	CR-T-	[1.001] switch	151
Humidification control disable	Disable humidification control from bus = yes	1 Bit	C-W--	[1.002] boolean	152
Dew-point temperature	Dew-point temperature = enabled	2 Byte	CR-T-	[9.001] temperatures ° C	144
Window contact sensor 1 (from bus)	Window contacts function = enabled, Window contact 1 = enabled	1 Bit	C-WTU	[1.019] window/door	101
Window contact sensor 2 (from bus	Window contacts function = enabled, Window contact 2 = enabled	1 Bit	C-WTU	[1.019] window/door	102

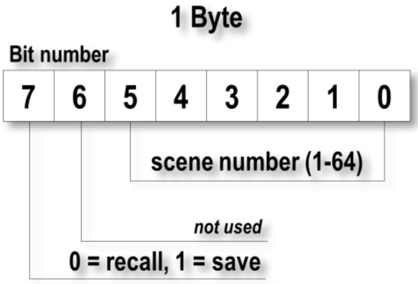
Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Presence Sensor 1 (from bus)	Presence sensor function = enabled	1 Bit	C-WTU	[1.018] occupancy	103
Presence Sensor 2 (from bus)	Presence sensor function = enabled	1 Bit	C-WTU	[1.018] occupancy	104
Contact of card holder (from bus)	Card holder function = enabled	1 Bit	C-WTU	[1.018] occupancy	105
Logic function X - Input 1	Logic function X = enabled Logic object 1 = enabled	1 Bit	C-WTU	[1.001] switch	70, 75, 80, 85
	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	71, 76, 81, 86
	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	72, 77, 82, 87
	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	73, 78, 83, 88
	x = 1,2,3,4				
Logic function X - Output	logic function X = enabled	1 Bit	C-WTU	[1.001] switch	74, 79, 84, 89
	x = 1,2,3,4				
Output 1A [1B] - On / Off command	Output = 2 binary outputs	1 Bit	CRWTU	[1.001] switch	158, 169
	This communication object is the direct command to the output setting.				
Channel 1 – Move up-down command	Output = valve / venetian blind / shutter	1 Bit	C-W-	[1.008] up/down [1.009] open/close	180
	Trigger object for continuous movement: when received, it starts continuous movement in the specified direction.				
Channel 1 - Stop-step up-down command	Output = valve / venetian blind / shutter	1 Bit	C-W-	[1.007] step	181
	Trigger object for step movement: when received, and the actuator is at rest, it starts a step movement in the specified direction. If the actuator is not at rest, just stops current movement.				

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Channel 1 - Dedicated stop command	Output = valve / venetian blind / shutter	1 Bit	C-W-	[1.017] trigger	182
	Stop any ongoing movement when received.				
Channel 1 - Info move	Output = valve / venetian blind / shutter	1 Bit	CR-T-	[1.008] up/down	183
	Allows to query the current movement direction.				
Channel 1 - valid current abs position	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	1 Bit	CR-T-	[1.002] boolean	184
	Signals that the actuator has reached the requested absolute position. *Issued on absolute position movement commands.*				
Channel 1 - Abs [valve / shutter / blind] position command	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	1 Bit	C-W-	[5.001] percentage (0..100%)	192
	Sets the target absolute position to reach and starts actuator movement *For the venetian blinds, the position refers to the blinds' panel.*				
Channel 1 - Abs [valve / shutter / blind] position status	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled	1 Bit	CR-T-	[5.001] percentage (0..100%)	193
	Returns the current absolute position of the actuator. *The position is computed from the sequence of requested movements and realigned whenever an endpoint is reached.* *For the venetian blinds, the position refers to the blinds' panel.*				
Channel 1 - Abs slat position control	Output = valve / venetian blind / shutter Use = venetian blind	1 Bit	C-W--	[5.001] percentage (0..100%)	195
	Sets the target absolute position for the slats to reach and starts actuator movement.				
Channel 1 - Abs slat position status	Output = valve / venetian blind / shutter Use = venetian blind	1 Bit	CR-T-	[5.001] percentage (0..100%)	196
	It returns the current absolute position of the slats. *The position is computed from the sequence of requested movements and realigned whenever an endpoint of the slats' rotation is reached.*				
Output 1A [1B] - on / off status	Output = 2 binary outputs Status feedback telegram = enabled	1 Bit	CR-T-	[1.001] switch	159, [170]

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
	Sent at any change of the output state and also periodically, as configured.				
Output 1A [1B] - Staircase lighting start stop command	Output = 2 binary outputs staircase lighting function = enabled	1 Bit	C-W--	[1.001] on/off	160, [171]
	Starts the staircase light timing with an On value. *The timed activation automatically stops at the end of the preset time.* *If "Manual off" is enabled, the communication object will stop the timing with an Off value.*				
Output 1A [1B] - Lock command	Output = 2 binary outputs locking function = enabled	1 Bit	C-W--	[1.003] enable	161, [172]
	Inhibits the switching commands for the output when an "enable" telegram is received, and unlocks them when a "disable" telegram is received.				
Output 1A [1B] - Forcing Command	Output = 2 binary outputs forcing function = enabled	2 Bit	C-W--	[2.001] switch control	162, [173]
	Allows to force the status of an output. *It is composed of 2 bits: the first one is used for the priority value (i.e. defines whether the forcing is in effect, "Priority", or not) and the second one for the imposed value (which is not considered if forcing is not effective).*  2 bit Bit number 1 0 0 = off, 1 = on 0 = No priority, 1 = Priority				
Output 1A [1B] - Scene Number	Output = 2 binary outputs scenes function = enabled	1 Byte	C-W--	[17.001] scene number [18.001] scene control	163, [174]
	Allows to recall a scene setting for the status of the output, and to store current status in association to the specified scene.  1 Byte Bit number 7 6 5 4 3 2 1 0 scene number (1-64) not used 0 = recall, 1 = save				

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Output 1A [1B] - kWh counter	Output = 2 binary outputs Operating time/energy counter = enabled	4-byte signed counter	CR-T-	[13.013] active energy [kWh]	164, [175]
	Stores the current counter value of the accumulated energy.				
Output 1A [1B] - kWh counter reset command	Output = 2 binary outputs Operating time/energy counter = enabled	1 Bit	C-W--	[1.015] reset	165, [176]
	Resets the energy counter to 0.				
Output 1A [1B] - Operating time counter	Output = 2 binary outputs Operating time/energy counter = enabled	2-byte unsigned counter	CR-T-	[7.007] time [h]	166, [177]
	Stores the current counter value of the accumulated operating time.				
Output 1A [1B] - Operating time counter reset command	Output = 2 binary outputs Operating time/energy counter = enabled	1 Bit	C-W--	[1.015] reset	167, [178]
	Resets the operating hour counter to 0.				
Output 1A [1B] - Operating time counter runout	Output = 2 binary outputs Operating time/energy counter = enabled	1 Bit	CR-T-	[1.005] alarm	168, [179]
	1-bit alarm sent when the time counter reaches the maximum value of 65535 hours.				

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Channel 1 - Dimmer blind position command	Output = valve / venetian blind / shutter Use = all except 3-way valve enabled position control with dimmer = yes	4-bit controlled	C-W--	[3.008] blind control	191
	It allows to command the actuator through a dimmer-style command.  [3.008] 4 bit Bit number 3 2 1 0 Move: 0 = Up, 1 = Down Number of steps 1...7 (001b...111b) or Stop (000b) [3.008] Blinds (4 bit) Up (1 step) 1 0 0 1 Down (1 step) 0 0 0 1 Stop 0 0 0 0				
Channel 1 - Dimmer slats command	Output = valve / venetian blind / shutter Use = venetian blind Slats control with dimmer = yes	4-bit controlled	C-W--	[3.008] blind control	194
	It allows to command the slats position through a dimmer-style command. *See previous entry for bit field details.*				
Channel 1- Lock Command	Locking function = enabled	1 Bit	C-W--	[1.003] enable	185
	It inhibits the switching commands for the output when an "enable" telegram is received, and unlocks them when a "disable" telegram is received.				
Channel 1 - Forcing command	Forcing operation = enabled	2 Bit	C-W--	[2.008] direction 1 control	186
	Allows to force the status of an output pair. The command is a "direction control" telegram, which can force movement in one direction, the other, or release forcing.  2 bit Bit number [2.008] 1 0 0 = No control, 1 = Control Value (if control = 1)				

Object Name	Conditions	Size	Flags	DPT	Comm Obj. N.
Channel 1- Wind alarm	Meteo alarms = enabled	1 Bit	C-W--	[1.005] alarm	187
	If this alarm is enabled, writing an active alarm value here will set the corresponding alarm condition; the alarm will be released by writing a "clear alarm" value. *If the heartbeat timeout" is set, even in absence of an alarm condition, the "clear alarm" value must be regularly written at intervals not higher than the timeout period.*				
Channel 1 - Frost alarm	Meteo alarms = enabled	1 Bit	C-W--	[1.005] alarm	188
	Same considerations as for previous alarm apply.				
Channel 1 - Rain alarm	Meteo alarms = enabled	1 bit	C-W--	[1.005] alarm	189
	Same considerations as for previous alarm apply.				
Channel 1 - Scene Number	Scene function = enabled	1 Byte	C-W--	[17.001] scene number [18.001] scene control	190
	Allows to recall a scene setting for the status of the output, and to store current status in association to the specified scene.  1 Byte Bit number 7 6 5 4 3 2 1 0 scene number (1-64) not used 0 = recall, 1 = save				

A.2 Summary of KNX communication objects by progressive number

Below is a list of the KNX communication objects, along with the related Point Data Types (DPT) defined by the application program, depending on the selected configurations.

The list is generically ordered by CO progressive number. In case of similar objects related to different inputs, only the first input is mentioned.

CO	ComObject	Text	Dimen s.	Flag	DataPoint Typr
1	TechnicalAlarm	TechnicalAlarm	1 Bit	C-W--	Variable(**)
2	LockCommand_1	Input 1 - LockCommand	1 Bit	C-W-U	Variable(**)
3	SwitchingStatus_1_1	Input 1 - SwitchingStatus1	2 Bytes	C-WTU	Variable(**)
4	SwitchingStatus_1_2	Input 1 - SwitchingStatus2	2 Bytes	C-WTU	Variable(**)

5	SwitchingStatus_1_3	Input 1 - SwitchingStatus3	2 Bytes	C-WTU	Variable(**)
6	SwitchingStatus_1_4	Input 1 - SwitchingStatus4	2 Bytes	C-WTU	Variable(**)
7	SwitchingStatus_1_5	Input 1 - SwitchingStatus5	2 Bytes	C-WTU	Variable(**)
8	SwitchingStatus_1_6	Input 1 - SwitchingStatus6	2 Bytes	C-WTU	Variable(**)
9	SwitchingStatus_1_7	Input 1 - SwitchingStatus7	2 Bytes	C-WTU	Variable(**)
10	SwitchingStatus_1_8	Input 1 - SwitchingStatus8	2 Bytes	C-WTU	Variable(**)
11	SwitchingCommand_1	Input 1 - SwitchingCommand-DedicatedStop	1 Bit	C-WTU	Variable(**)
12	DimmingUpDownStopCommand_1	Input 1 - DimmingUpDownStopCommand	4 Bit	C- -T-	Variable(**)
13	InfoStatus_1	Dummy - Input 1 - InfoStatus	1 Bit	-	Internal Only
14	StopStepUpDownCommand_1	Input 1 - StopStepUpDownCommand	1 Bit	C- -T-	Variable(**)
15	MoveUpDownCommand_1	Input 1 - MoveUpDownCommand	1 Bit	C- -T-	Variable(**)
16	CounterValue_1	Input 1 - CounterValue-SceneNumberControl	4 Bytes	C- -T-	Variable(**)
17	CounterResetCommand_1	Dummy - Input 1 - Counter reset command	1 Bit	-	Internal Only
18	CounterRunout_1	Dummy - Counter runout	1 Bit	-	Internal Only
19	LockCommand_2	Input 2 - LockCommand	1 Bit	C-W-U	Variable(**)
20	SwitchingStatus_2_1	Input 2 - SwitchingStatus1	2 Bytes	C-WTU	Variable(**)
21	SwitchingStatus_2_2	Input 2 - SwitchingStatus2	2 Bytes	C-WTU	Variable(**)
22	SwitchingStatus_2_3	Input 2 - SwitchingStatus3	2 Bytes	C-WTU	Variable(**)
23	SwitchingStatus_2_4	Input 2 - SwitchingStatus4	2 Bytes	C-WTU	Variable(**)
24	SwitchingStatus_2_5	Input 2 - SwitchingStatus5	2 Bytes	C-WTU	Variable(**)
25	SwitchingStatus_2_6	Input 2 - SwitchingStatus6	2 Bytes	C-WTU	Variable(**)

26	SwitchingStatus_2_7	Input 2 - SwitchingStatus7	2 Bytes	C-WTU	Variable(**)
27	SwitchingStatus_2_8	Input 2 - SwitchingStatus8	2 Bytes	C-WTU	Variable(**)
28	SwitchingCommand_2	Input 2 - SwitchingCommand-DedicatedStop	1 Bit	C-WTU	Variable(**)
29	DimmingUpDownStopCommand_2	Input 2 - DimmingUpDownStopCommand	4 Bit	C- -T-	Variable(**)
30	InfoStatus_2	Dummy - Input 2 - InfoStatus	1 Bit	-	Internal Only
31	StopStepUpDownCommand_2	Input 2 - StopStepUpDownCommand	1 Bit	C- -T-	Variable(**)
32	MoveUpDownCommand_2	Input 2 - MoveUpDownCommand	1 Bit	C- -T-	Variable(**)
33	CounterValue_2	Input 2 - CounterValue-SceneNumberControl	4 Bytes	C- -T-	Variable(**)
34	CounterResetCommand_2	Dummy - Input 2 - Counter reset command	1 Bit	-	Internal Only
35	CounterRunout_2	Dummy - Input 2 - Counter runout	1 Bit	-	Internal Only
36	LockCommand_3	Input 3 - LockCommand	1 Bit	C-W-U	Variable(**)
37	SwitchingStatus_3_1	Input 3 - SwitchingStatus1	2 Bytes	C-WTU	Variable(**)
38	SwitchingStatus_3_2	Input 3 - SwitchingStatus2	2 Bytes	C-WTU	Variable(**)
39	SwitchingStatus_3_3	Input 3 - SwitchingStatus3	2 Bytes	C-WTU	Variable(**)
40	SwitchingStatus_3_4	Input 3 - SwitchingStatus4	2 Bytes	C-WTU	Variable(**)
41	SwitchingStatus_3_5	Input 3 - SwitchingStatus5	2 Bytes	C-WTU	Variable(**)
42	SwitchingStatus_3_6	Input 3 - SwitchingStatus6	2 Bytes	C-WTU	Variable(**)
43	SwitchingStatus_3_7	Input 3 - SwitchingStatus7	2 Bytes	C-WTU	Variable(**)
44	SwitchingStatus_3_8	Input 3 - SwitchingStatus8	2 Bytes	C-WTU	Variable(**)
45	SwitchingCommand_3	Input 3 - SwitchingCommand-DedicatedStop	1 Bit	C- -T-	Variable(**)

46	DimmingUpDownStopCommand_3	Input 3 - DimmingUpDownStopCommand	4 Bit	C- -T-	Variable(**)
47	InfoStatus_3	Dummy - Input 3 - InfoStatus	1 Bit	-	Internal Only
48	StopStepUpDownCommand_3	Input 3 - StopStepUpDownCommand	1 Bit	C- -T-	Variable(**)
49	MoveUpDownCommand_3	Input 3 - MoveUpDownCommand	1 Bit	C- -T-	Variable(**)
50	CounterValue_3	Input 3 - CounterValue-SceneNumberControl	4 Bytes	C- -T-	Variable(**)
51	CounterResetCommand_3	Input 3 - Counter reset command	1 Bit	-	Internal Only
52	CounterRunout_3	Input 3 - Counter runout	1 Bit	-	Internal Only
53	LockCommand_4	Input 4 - LockCommand	1 Bit	C-W-U	Variable(**)
54	SwitchingStatus_4_1	Input 4 - SwitchingStatus1	2 Bytes	C-WTU	Variable(**)
55	SwitchingStatus_4_2	Input 4 - SwitchingStatus2	2 Bytes	C-WTU	Variable(**)
56	SwitchingStatus_4_3	Input 4 - SwitchingStatus3	2 Bytes	C-WTU	Variable(**)
57	SwitchingStatus_4_4	Input 4 - SwitchingStatus4	2 Bytes	C-WTU	Variable(**)
58	SwitchingStatus_4_5	Input 4 - SwitchingStatus5	2 Bytes	C-WTU	Variable(**)
59	SwitchingStatus_4_6	Input 4 - SwitchingStatus6	2 Bytes	C-WTU	Variable(**)
60	SwitchingStatus_4_7	Input 4 - SwitchingStatus7	2 Bytes	C-WTU	Variable(**)
61	SwitchingStatus_4_8	Input 4 - SwitchingStatus8	2 Bytes	C-WTU	Variable(**)
62	SwitchingCommand_4	Input 4 - SwitchingCommand-DedicatedStop	1 Bit	C-WTU	Variable(**)
63	DimmingUpDownStopCommand_4	Input 4 - DimmingUpDownStopCommand	4 Bit	C- -T-	Variable(**)
64	InfoStatus_4	Dummy - Input 4 - InfoStatus	1 Bit	-	Internal Only
65	StopStepUpDownCommand_4	Input 4 - StopStepUpDownCommand	1 Bit	C- -T-	Variable(**)
66	MoveUpDownCommand_4	Input 4 - MoveUpDownCommand	1 Bit	C- -T-	Variable(**)

67	CounterValue_4	Input 4 - CounterValue-SceneNumberControl	4 Bytes	C- -T-	Variable(**)
68	CounterResetCommand_4	Input 4 - Counter reset command	1 Bit	-	Internal Only
69	CounterRunout_4	Input 4 - Counter runout	1 Bit	-	Internal Only
70	LogicFunctionIn_1_1	Logic function 1 - Input 1	1 Bit	C-WTU	[1.1] DPT_Switch
71	LogicFunctionIn_1_2	Logic function 1 - Input 2	1 Bit	C-WTU	[1.1] DPT_Switch
72	LogicFunctionIn_1_3	Logic function 1 - Input 3	1 Bit	C-WTU	[1.1] DPT_Switch
73	LogicFunctionIn_1_4	Logic function 1 - Input 4	1 Bit	C-WTU	[1.1] DPT_Switch
74	LogicFunctionOut_1	Logic function 1 - Output	1 Bit	C-WTU	[1.1] DPT_Switch
75	LogicFunctionIn_2_1	Logic function 2 - Input 1	1 Bit	C-WTU	[1.1] DPT_Switch
76	LogicFunctionIn_2_2	Logic function 2 - Input 2	1 Bit	C-WTU	[1.1] DPT_Switch
77	LogicFunctionIn_2_3	Logic function 2 - Input 3	1 Bit	C-WTU	[1.1] DPT_Switch
78	LogicFunctionIn_2_4	Logic function 2 - Input 4	1 Bit	C-WTU	[1.1] DPT_Switch
79	LogicFunctionOut_2	Logic function 2 - Output	1 Bit	C-WTU	[1.1] DPT_Switch
80	LogicFunctionIn_3_1	Logic function 3 - Input 1	1 Bit	C-WTU	[1.1] DPT_Switch
81	LogicFunctionIn_3_2	Logic function 3 - Input 2	1 Bit	C-WTU	[1.1] DPT_Switch
82	LogicFunctionIn_3_3	Logic function 3 - Input 3	1 Bit	C-WTU	[1.1] DPT_Switch
83	LogicFunctionIn_3_4	Logic function 3 - Input 4	1 Bit	C-WTU	[1.1] DPT_Switch
84	LogicFunctionOut_3	Logic function 3 - Output	1 Bit	C-WTU	[1.1] DPT_Switch
85	LogicFunctionIn_4_1	Logic function 4 - Input 1	1 Bit	C-WTU	[1.1] DPT_Switch
86	LogicFunctionIn_4_2	Logic function 4 - Input 2	1 Bit	C-WTU	[1.1] DPT_Switch
87	LogicFunctionIn_4_3	Logic function 4 - Input 3	1 Bit	C-WTU	[1.1] DPT_Switch

88	LogicFunctionIn_4_4	Logic function 4 - Input 4	1 Bit	C-WTU	[1.1] DPT_Switch
89	LogicFunctionOut_4	Logic function 4 - Output	1 Bit	C-WTU	[1.1] DPT_Switch
90	TempValue	Thermostat - Temperature value	2 Bytes	CR-T-	[9.1] DPT_Value_Temp
91	TempTreshold1Switch	Thermostat - Temperature threshold 1 - Switch	1 Bit	CR-T-	[1.1] DPT_Switch
92	TempTreshold2Switch	Thermostat - Temperature threshold 2 - Switch	1 Bit	CR-T-	[1.1] DPT_Switch
93	RoomTempFB	Thermostat - Room temperature (from bus)	2 Bytes	C-WTU	[9.1] DPT_Value_Temp
94	Humidity2bytesFB	Thermostat - Humidity value (2 bytes, from bus)	2 Bytes	C-WTU	[9.7] DPT_Value_Humidity
95	Humidity1byteFB	Thermostat - Humidity value (1 byte, from bus)	1 Byte	C-WTU	[5.1] DPT_Scaling
96	AntistratificationTempFB	Thermostat - Antistratification temperature (from bus)	2 Bytes	C-WTU	[9.1] DPT_Value_Temp
97	OutdoorTempFB	Thermostat - Outdoor temperature (from bus)	2 Bytes	C-WTU	[9.1] DPT_Value_Temp
98	CoilTempFB	Thermostat - Coil temperature (from bus)	2 Bytes	C-WTU	[9.1] DPT_Value_Temp
99	FloorTempFB	Thermostat - Floor temperature (from bus)	2 Bytes	C-WTU	[9.1] DPT_Value_Temp
100	FlowTempFB	Thermostat - Flow temperature (from bus)	2 Bytes	C-WTU	[9.1] DPT_Value_Temp
101	WindowsContactSensor1FB	Thermostat - Windows contact sensor 1 (from bus)	1 Bit	C-WTU	[1.19] DPT_Window_Door
102	WindowsContactSensor2FB	Thermostat - Windows contact sensor 2 (from bus)	1 Bit	C-WTU	[1.19] DPT_Window_Door

103	PresenceSensor1FB	Thermostat - Presence sensor 1 (from bus)	1 Bit	C-WTU	[1.18] DPT_Occupancy
104	PresenceSensor2FB	Thermostat - Presence sensor 2 (from bus)	1 Bit	C-WTU	[1.18] DPT_Occupancy
105	ContactHolderFB	Thermostat - Contact of card holder (from bus)	1 Bit	C-WTU	[1.18] DPT_Occupancy
106	AnticondensationFB	Thermostat - Anticondensation (from bus)	1 Bit	C-WTU	[1.1] DPT_Switch
107	WeightedTemp	Thermostat - Weighted temperature	2 Bytes	CR-T-	[9.1] DPT_Value_Temp
108	HeatingCoolingStatusOut	Thermostat - Heating/cooling status out	1 Bit	CR-T-	[1.100] DPT_Heat_Cool
109	HeatingCoolingStatusIn	Thermostat - Heating/cooling status in	1 Bit	C-W- -	[1.100] DPT_Heat_Cool
110	HVACModeIn	Thermostat - HVAC mode in	1 Byte	C-W- -	[20.102] DPT_HVACMode
111	HVACForcedModeIn	Thermostat - HVAC forced mode in	1 Byte	C-W- -	[20.102] DPT_HVACMode
112	HVACModeOut	Thermostat - HVAC mode out	1 Byte	CR-T-	[20.102] DPT_HVACMode
113	HVACManualMode	Thermostat - HVAC manual mode	1 Byte	C-WTU	[20.102] DPT_HVACMode
114	ChronoModeStatus	Thermostat - Chrono status	1 Bit	CR-T-	[1.11] DPT_State
115	ActualSetpoint	Thermostat - Actual setpoint	2 Bytes	CR-T-	[9.1] DPT_Value_Temp
116	ManualSetpoint	Thermostat - Manual setpoint	2 Bytes	C-W- -	[9.1] DPT_Value_Temp
117	SetpointComfort_1	Thermostat - Comfort setpoint (heating)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp

118	SetpointComfort_2	Thermostat - Comfort setpoint (cooling)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp
119	SetpointStandby_1	Thermostat - Standby setpoint (heating)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp
119	OffsetStandby_1	Thermostat – Standby offset (heating)	2 Bytes	CRWTU	[9.2] DPT_Value_Tempd
120	SetpointStandby_2	Thermostat - Standby setpoint (cooling)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp
120	OffsetStandby_2	Thermostat - Standby offset (cooling)	2 Bytes	CRWTU	[9.2] DPT_Value_Tempd
121	SetpointEconomy_1	Thermostat - Economy setpoint (heating)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp
121	OffsetEconomy_1	Thermostat – Economy offset (heating)	2 Bytes	CRWTU	[9.2] DPT_Value_Tempd
122	SetpointEconomy_2	Thermostat - Economy setpoint (cooling)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp
122	OffsetEconomy_2	Thermostat – Economy offset (cooling)	2 Bytes	CRWTU	[9.2] DPT_Value_Tempd
123	SetpointBuildProtection_1	Thermostat - Building protection setpoint (heating)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp
124	SetpointBuildProtection_2	Thermostat - Building protection setpoint (cooling)	2 Bytes	CRWTU	[9.1] DPT_Value_Temp
125	RoomTempControllerStatus	Thermostat - Room temperature controller status	1 Bit	CR-T-	[1.3] DPT_Enable
126	HeatingOut	Thermostat - Heating out command	1 Byte	CR-T-	[5.1] DPT_Scaling
126	HeatingOut1bit	Thermostat - Heating out command 1bit	1 Bit	CR-T-	[1.1] DPT_Switch

126	HeatingCoolingOut1byte	Thermostat - Heating and Cooling out command 1 byte	1 Byte	CR-T-	[5.1] DPT_Scaling
126	HeatingCoolingOut1bit	Thermostat - Heating and Cooling out command 1 bit	1 Bit	CR-T-	[1.1] DPT_Switch
127	CoolingOut	Thermostat - Cooling out command	1 Byte	CR-T-	[5.1] DPT_Scaling
127	CoolingOut1bit	Thermostat - Cooling out command 1 bit	1 Bit	CR-T-	[1.1] DPT_Switch
128	AuxiliaryHeatingOut	Thermostat - Auxiliary heating output command	1 Bit	CR-T-	[1.1] DPT_Switch
128	AuxiliaryHeatingCoolingOut	Thermostat - Auxiliary heating and cooling output command	1 Bit	CR-T-	[1.1] DPT_Switch
129	AuxiliaryCoolingOut	Thermostat - Auxiliary cooling output command	1 Bit	CR-T-	[1.1] DPT_Switch
130	AuxiliaryHeatingDisable	Thermostat - Auxiliary heating disable	1 Bit	C-W- -	[1.3] DPT_Enable
131	AuxiliaryCoolingDisable	Thermostat - Auxiliary cooling disable	1 Bit	C-W- -	[1.3] DPT_Enable
132	FanContinuousSpeed	Thermostat - Fan continuous speed	1 Byte	CR-T-	[5.1] DPT_Scaling
133	FanSpeed_1	Thermostat - Fan speed 1	1 Bit	CR-T-	[1.1] DPT_Switch
134	FanSpeed_2	Thermostat - Fan speed 2	1 Bit	CR-T-	[1.1] DPT_Switch
135	FanSpeed_3	Thermostat - Fan speed 3	1 Bit	CR-T-	[1.1] DPT_Switch
136	FanControlDisable	Thermostat - Fan control disable	1 Bit	C-W- -	[1.2] DPT_Bool
137	ThermAlarmText	Thermostat - Text alarm	14 Bytes	CR-T-	[16.0] DPT_String_ASCII
138	ManualModeStatus	Thermostat - Manual mode status	1 Bit	CRWTU	[1.11] DPT_State
139	FanStepManualSpeed	Thermostat - Fan step manual speed	1 Byte	CRWTU	[5.10] DPT_Value_1_Ucount
140	FanStepSpeedStatus	Thermostat - Status fan step speed	1 Byte	CR-T-	[5.10] DPT_Value_1_Ucount

141	VentManualOperation	Thermostat - Ventilation manual operation	1 Bit	CRWTU	[1.11] DPT_State
142	TempControllerAlarm	Thermostat - Room temperature controller alarm	1 Bit	CR-T-	[1.5] DPT_Alarm
143	DisableTempController	Thermostat - Disable room temperature controller	1 Bit	C-W- -	[1.1] DPT_Switch
144	DewPointTemp	Thermostat - Dew-point temperature	2 Bytes	CR-T-	[9.1] DPT_Value_Temp
145	HumiditySetpointForDehum	Thermostat - Relative humidity setpoint for dehumidification	2 Bytes	CRWTU	[9.7] DPT_Value_Humidity
146	HumiditySetpointForHum	Thermostat - Relative humidity setpoint for humidification	2 Bytes	CRWTU	[9.7] DPT_Value_Humidity
147	DehumCommand	Thermostat - Dehumidification command	1 Bit	CR-T-	[1.1] DPT_Switch
148	DehumWaterBatteryCommand	Thermostat - Dehumidification water battery command	1 Bit	CR-T-	[1.1] DPT_Switch
149	DehumIntegrationControl	Thermostat - Dehumidification integration control	1 Bit	CR-T-	[1.1] DPT_Switch
150	DehumControlDisable	Thermostat - Dehumidification control disable	1 Bit	C-W- -	[1.2] DPT_Bool
151	HumidCommand	Thermostat - Humidification command	1 Bit	CR-T-	[1.1] DPT_Switch
152	HumidControlDisable	Thermostat - Humidification control disable	1 Bit	C-W- -	[1.2] DPT_Bool
153	AnticondensationAlarm	Thermostat - Anticondensation alarm	1 Bit	CR-T-	[1.5] DPT_Alarm
154	ThermalGeneratorLock	Thermostat - Thermal generator lock	1 Bit	C-W- -	[1.5] DPT_Alarm
155	HVACProtectionModeStatus	Thermostat - HVAC building protection mode activated	1 Bit	CR-T-	[1.1] DPT_State
156	FanManualSpeedPercentage	Thermostat - Fan manual speed percentage	1 Byte	CR-T-	[5.1] DPT_Scaling
157	FanManualOffStatus	Thermostat - Fan manual off status	1 Bit	CR-T-	[1.1] DPT_State
158	O_OnOffCommand_1	Output 1A – On/Off Command	1 Bit	CRWTU	[1] 1.001 DPT_Switch

159	O_OnOffStatus_1	Output 1A – On/Off Status	1 Bit	CR-T-	[1] 1.001 DPT_Switch
160	O_StaircaseStartStopCommand_1	Output 1A – Staircase lighting start stop command	1 Bit	C-W- -	[1] 1.001 DPT_Switch
161	O_LockCommand_1	Output 1A – Lock command	1 Bit	C-W- -	[1] 1.003 DPT_Enable
162	O_ForcingCommand_1	Output 1A – Forcing command	2 Bit	C-W- -	[2] 2.001 DPT_Switch_Control
163	O_SceneNumberControl_1	Output 1A – scene number	1 Byte	C-W- -	[8] [17.1] DPT_SceneNumber, [18.1] DPT_SceneControl
164	O_kWhCounter_1	Output 1A – kWh counter	4 Bytes	CR-T-	[32] 13.013 DPT_ActiveEnergy_kWh
165	O_kWhCounterResetCommand_1	Output 1A – kWh counter reset command	1 Bit	C-W- -	[1] 1.015 DPT_Reset
166	O_SecondsCounter_1	Output 1A – Operating time counter	2 Bytes	C-W- -	[16] 7.007 DPT_TimePeriodHrs
167	O_SecondsCounterResetCommand_1	Output 1A – Operating time counter reset command	1 Bit	C-W- -	[1] 1.015 DPT_Reset
168	O_SecondsCounterRunout_1	Output 1A – Operating time counter runout	1 Bit	CR-T-	[1] 1.005 DPT_Alarm
169	O_OnOffCommand_2	Output 1B – On/Off Command	1 Bit	CRWTU	[1] 1.001 DPT_Switch
170	O_OnOffStatus_2	Output 1B – On/Off Status	1 Bit	CR-T-	[1] 1.001 DPT_Switch
171	O_StaircaseStartStopCommand_2	Output 1B – Staircase lighting start stop command	1 Bit	C-W- -	[1] 1.001 DPT_Switch
172	O_OutLockCommand_2	Output 1B – Lock command	1 Bit	C-W- -	[1] 1.003 DPT_Enable
173	O_ForcingCommand_2	Output 1B – Forcing command	2 Bit	C-W- -	[2] 2.001 DPT_Switch_Control
174	O_SceneNumberControl_2	Output 1B – scene number	1 Byte	C-W- -	[8] [17.1] DPT_SceneNumber, [18.1]

					DPT_Scene Control
175	O_kWhCounter_2	Output 1B – kWh counter	4 Bytes	CR-T-	[32] 13.013 DPT_Active Energy_kWh
176	O_kWhCounterResetCommand_2	Output 1B – kWh counter reset command	1 Bit	C-W- -	[1] 1.015 DPT_Reset
177	O_SecondsCounter_2	Output 1B – Operating time counter	2 Bytes	CR-T-	[16] 7.007 DPT_TimePeriodHrs
178	O_SecondsCounterResetCommand_2	Output 1B – Operating time counter reset command	1 Bit	C-W- -	[1] 1.015 DPT_Reset
179	O_SecondsCounterRunout_2	Output 1B – Operating time counter runout	1 Bit	CR-T-	[1] 1.005 DPT_Alarm
180	O_MoveUpDownCommandChn_1	Channel 1 - Move up-down command	1 Bit	C-W- -	[1] 1.008 DPT_UpDown, 1.009 DPT_OpenClose
181	O_StopStepUpDownCommandChn_1	Channel 1 - Stop-step up-down command	1 Bit	C-W- -	[1] 1.007 DPT_Step
182	O_DedicatedStopCommandChn_1	Channel 1 - Dedicated stop command	1 Bit	C-W- -	[1] 1.017 DPT_Trigger
183	O_InfoMoveStatusChn_1	Channel 1 - Info move	1 Bit	CR-T-	[1] 1.008 DPT_UpDown
184	O_ValidCurrentAbsPositionChn_1	Channel 1 - Valid current abs position	1 Bit	CR-T-	[1] 1.002 DPT_Bool
185	O_LockCommandChn_1	Channel 1 - Lock command	1 Bit	C-W- -	[1] 1.003 DPT_Enable
186	O_ForcingCommandChn_1	Channel 1 - Forcing command	2 Bit	C-W- -	[2] 2.001 DPT_Switch_Control
187	O_WindAlarmChn_1	Channel 1 - Wind alarm	1 Bit	C-W- -	[1] 1.005 DPT_Alarm
188	O_FrostAlarmChn_1	Channel 1 - Frost alarm	1 Bit	C-W- -	[1] 1.005 DPT_Alarm
189	O_RainAlarmChn_1	Channel 1 - Rain alarm	1 Bit	C-W- -	[1] 1.005 DPT_Alarm
190	O_SceneNumberChn_1	Channel 1 - Scene number	1 Byte	C-W- -	[8] [17.1] DPT_Scene

					Number, [18.1] DPT_Scene Control
191	O_DimmerPosCommandChn_1	Channel 1 - Dimmer blind/shutter/valve position command	4 Bit	C-W- -	[4] 3.007 DPT_Control_Dimming
192	O_AbsPositionCommandChn_1	Channel 1 - Abs blind/shutter/valve position command	1 Byte	C-W- -	[8] 5.001 DPT_Scaling
193	O_AbsPositionStatusChn_1	Channel 1 - Abs blind/shutter/valve position status	1 Byte	CR-T-	[8] 5.001 DPT_Scaling
194	O_DimmerSlatsCommandChn_1	Channel 1 - Dimmer slats command	4 Bit	C-W- -	[4] 3.007 DPT_Control_Dimming
195	O_AbsSlatsPositionCommandChn_1	Channel 1 - Abs slats position command	1 Byte	C-W- -	[8] 5.001 DPT_Scaling
196	O_AbsSlatsPositionStatusChn_1	Channel 1 - Abs slats position status	1 Byte	CR-T-	[8] 5.001 DPT_Scaling

Data 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 with sign	[6.*] 8-bit signed value
2 bytes unsigned	[7.*] 2-byte unsigned value
2 bytes with sign	[8.*] 2-byte signed value
2 bytes floating point	[9.*] 2-byte float value

Table A1 - Size and DPT for CO configured as single or independent inputs

Warnings

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

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

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
- ekinex® is a registered trademark of SBS S.p.A.
- KNX® and ETS® are registered trademarks of KNX Association cvba, Brussels