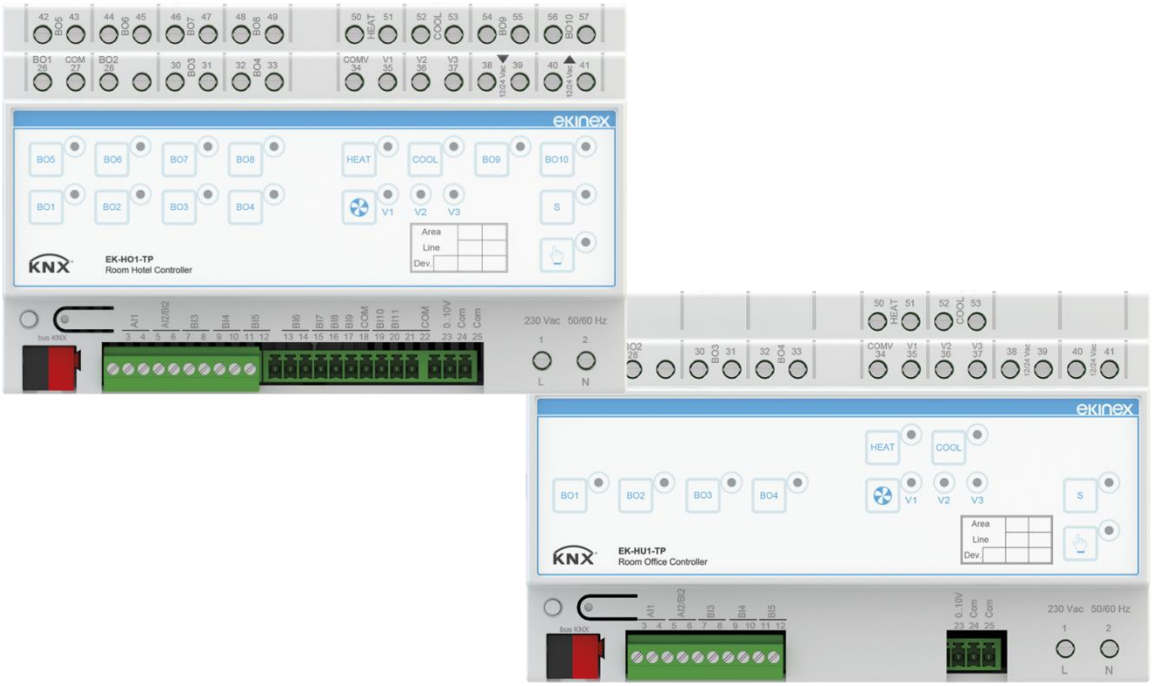
ekinex

CONTROL YOUR LIVING SPACE



Application manual
EK-HO1-TP / EK-HU1-TP
Hotel / Hospitality / Office Controllers

Summary

Revision	Changes	Date
1.2.0	EK-HU1-TP office module integration	18/02/2019
1.1.0	Introduction of the error message list in the communication object 10 Alarm text [16.0] DPT_String_ASCII	08/11/2018
1.0.0	Draft	21/07/2018

1 Document purpose

This manual describes the application details for version A1.0 of the ekinex® controllers EK-HO1-TP and EK-HU1-TP. The document is aimed at the system configurator as a description and reference guide for device functionalities and application programming. For mechanical and electrical details of the devices, please refer to the technical data sheet of the device.

This application manual and application programs for the ETS development environment are available for download at www.ekinex.com.

Document	File name (## = version)	Version	Device review	Last update
Product datasheet	STEKHOHU1TP##_IT.pdf		A1.0	07/2018
Application manual	MAEKHOHU1TP##_IT.pdf		A1.0	07/2018
Application program	APEKHO1TP##. knxprod	EK-HO1-TP	A1.0	07/2018
Application program	APEKHU1TP##. knxprod	EK-HU1-TP	A1.0	02/2019

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

EK-HO1-TP

EK-HU1-TP

2 General information

The ekinex® controllers EK-HO1-TP and EK-HU1-TP are KNX devices for mounting on a profiled DIN rail, optimized for the complete management respectively of a hotel room/accommodation facility or an office unit. A single product allows the management of the electric ON/OFF room lighting users, the control of blinds, blinds or shading devices, the command of signaling outputs and the management of the thermoregulation in heating and cooling for control of a convector or 3-speed fan coil with 2 or 4-pipe distribution.

The controllers can be integrated with wall-mounted KNX devices such as push-button controls and has binary inputs that allow easy interfacing of traditional push-buttons for on/off functions, roller shutter control and dimming (the implementation of the latter is carried out via communication objects sent on the bus).

About EK-HO1-TP. the interface to the badge-holder device inside the room and to the external access control device can be implemented either via dedicated binary inputs or via communication objects on the KNX bus: the controller has a set of functions dedicated to the management of courtesy lights, the general shutdown of the room and the command of an electric door.

For temperature control, the devices can be operated as simple actuators, in combination with an external controller consisting of another KNX device (eg an ekinex® room thermostat), or as true room temperature controllers, detecting the temperature environment by means of another KNX device (eg an ekinex® push-button control) or a traditional NTC temperature probe connected to one of the 2 analog inputs, calculating the control variable internally and carrying out the corresponding actuation.

The products have a membrane keyboard for manual control of the outputs, the LED indicators provide indications on the manual status of the outputs inserted; the function is useful for checking the connections during the electrical commissioning of the system.

The devices are equipped with an integrated communication module for the KNX bus and are suitable for mounting on a 35 mm profiled DIN rail according to EN 60715. The power supply to the logic part is supplied by the KNX bus (SELV, 30 Vdc); the power supply to the relay outputs is provided by the 230 Vac, 50-60 Hz mains voltage; in addition, for the electric lock command, SELV 12 Vdc / ac or 24 Vdc / ac auxiliary power supply terminals are available.

The supply includes inside the package:

- a device;
- a connection terminal for the KNX bus line;
- an instruction sheet.

2.1 EK-HO1-TP main functional features

Functions managed by the product:

- 1-2 shutter controls and / or dimming devices
- 8-10 lighting controls and ON / OFF signals
- 3 dimming controls for lighting from coupled (raise / lower) binary inputs that can be implemented by sending communication objects on the bus
- 6 dimming controls for lighting from single binary inputs (short / long press) that can be activated by sending communication objects on the bus
- 1 command for electric door lock of the room managed by binary input or by transponder via bus
- 1 fan-coil zone (with 3-speed fan or with 0 ... 10V control)
- 1 binary input or 1 communication object for cardholder contact inside the camera. Function for sending a maximum of 8 communication objects on the bus for power off / general activation + courtesy lights of the room

- 1 binary input or 1 communication object for access control contact outside the camera. Function for sending a sequence of up to 8 communication objects on the bus to manage the electric lock + courtesy lights of the room
- 4 logic functions

Input/outputs:

- [AI1/BI1]: freely configurable input as analogue input (ambient temperature probe, external probe, anti-stratification, probe on the heat-exchanger coil of the fan-coil, thermovector fluid delivery probe or generic) or as single-track input (sending commands and sequences, dimming, input for shutters / blinds commands and sending scenarios) or coupled (switching, dimming or input for shutters / blinds)
- [AI2/BI2]: freely configurable input as analogue input (ambient temperature probe, external probe, anti-stratification, probe on the heat-exchange coil of the fan-coil, thermovector fluid delivery probe or generic) or as single-track input (sending commands and sequences, dimming, input for shutters / blinds commands and sending scenarios) or coupled (switching, dimming or input for shutters / blinds)
- [BI3-BI4]: single binary inputs (sending commands and sequences, dimming, input for shutters / blinds and sending scenarios) or coupled (switching, dimming or input for shutters / blinds)
- [BI5-BI6]: single binary inputs (sending commands and sequences, dimming, input for shutters / blinds and sending scenarios) or coupled (switching, dimming or input for shutters / blinds)
- [BI7]: Binary input configurable as internal card-holder contact
- [BI8]: Binary input configurable as an external access control contact
- [BI9]: Binary input configurable as a window contact or as a generic contact
- [BI10]: Binary input configurable as assistance request contact
- [BI11]: Binary input configurable as room reset contact

- [BO1-BO2]: 2 outputs coupled to relays with a capacity of 5 (3) A, for control of 1 roller shutter, curtain or darkening device, or for lighting ON/OFF command or sending of signals; the outputs can be activated directly from the binary inputs and / or from communication objects on the bus
- [BO3-BO4]: 2 outputs coupled to relays with a capacity of 5 (3) A, for control of 1 roller shutter, curtain or darkening device, or for lighting ON / OFF command or sending of signals; the outputs can be activated directly from the binary inputs and / or from communication objects on the bus
- [BO5-BO10]: 6 relay outputs with a capacity of 16 (10) A, single for control of ON / OFF loads; the outputs can be activated directly from the binary inputs and / or from communication objects on the bus
- [DL]: door lock, 1 relay output powered at 12/24 Vac (contact capacity 5 (3) A), for sending pulses to an electric lock. The activation can be configured by binary input, to interface a traditional transponder installed outside the camera with an output contact, or from a communication object on the bus
- [BO1]: 1 output 0 ... 10V, for control of a fan-coil fan with brushless motor or for direct commands from the bus via communication objects

2.2 EK-HO1-TP Technical data

Part number	EK-HO1-TP
Supply voltage (microcontroller logic)	from bus KNX (30 Vdc)
Auxiliary power supply (for output relays)	230 Vac
Auxiliary power supply for electric strike command	SELV, 12 Vdc/ac / 24 Vdc/ac
AI1÷AI2: 2 analog inputs with passive NTC sensor	NTC, 10 kΩ a 25°C
BI1÷BI11: 11 inputs configurable as voltage free contacts	binary imputs, free contacts
AO1: 1 analog outputs 0... 10Vdc	to be combined with high impedance inputs
BO1÷BO4: 4 relay binary outputs, voltage-free contacts	monostable relay, SPST, 5(3) A / 250 Vac
V1-V2-V3: 3 relay binary outputs, voltage free contacts	monostable relay, SPST, 5(3) A / 250 Vac
1 relay binary output, live contact, for electric strike command	monostable relay, SPST, 5(3) A / 250 Vac
HEAT, COOL: 2 relay binary outputs, voltage-free contacts, for heating and cooling valve control	monostable relay, SPST, 16(10) A / 250 Vac
BO5÷BO10: 6 relay binary outputs, voltage-free contacts	monostable relay, SPST, 16(10) A / 250 Vac
Current consumption from the bus	< 30 mA
Operating temperature	0°C… +45°C
Degree of protection	IP20
Dimensions	144 x 90 x 60 mm (L x H x P)

2.3 EK-HU1-TP main functional features

Functions managed by the product:

- 1-2 shutter controls and / or dimming devices
- 2-4 lighting controls and ON / OFF signals
- 2 dimming controls for lighting from coupled (raise / lower) binary inputs that can be implemented by sending communication objects on the bus
- 5 dimming controls for lighting from single binary inputs (short / long press) that can be activated by sending communication objects on the bus
- 1 command for electric door lock of the room managed by binary input or by transponder via bus
- 1 fan-coil zone (with 3-speed fan or with 0 ... 10V control, 2-4 pipes)
- 4 logic functions

Input/outputs:

- [AI1/BI1]: freely configurable input as analogue input (ambient temperature probe, external probe, anti-stratification, probe on the heat-exchanger coil of the fan-coil, thermovector fluid delivery probe or generic) or as single-track input (sending commands and sequences, dimming, input for shutters / blinds commands and sending scenarios) or coupled (switching, dimming or input for shutters / blinds)
- [AI2/BI2]: freely configurable input as analogue input (ambient temperature probe, external probe, anti-stratification, probe on the heat-exchange coil of the fan-coil, thermovector fluid delivery probe or generic) or as single-track input (sending commands and sequences, dimming, input for shutters / blinds commands and sending scenarios) or coupled (switching, dimming or input for shutters / blinds) [BI3-BI4]: single binary inputs (sending commands and sequences, dimming, input for shutters / blinds and sending scenarios) or coupled (switching, dimming or input for shutters / blinds)
- [BI5]: single binary inputs (sending commands and sequences, dimming, input for shutters / blinds and sending scenarios)

- [BO1-BO2]: 2 outputs coupled to relays with a capacity of 5 (3) A, for control of 1 roller shutter, curtain or darkening device, or for lighting ON/OFF command or sending of signals; the outputs can be activated directly from the binary inputs and / or from communication objects on the bus
- [BO3-BO4]: 2 outputs coupled to relays with a capacity of 5 (3) A, for control of 1 roller shutter, curtain or darkening device, or for lighting ON / OFF command or sending of signals; the outputs can be activated directly from the binary inputs and / or from communication objects on the bus
- [DL]: door lock, 1 relay output powered at 12/24 Vac (contact capacity 5 (3) A), for sending pulses to an electric lock. The activation can be configured by binary input, to interface a traditional transponder installed outside the camera with an output contact, or from a communication object on the bus
- [BO1]: 1 output 0 ... 10V, for control of a fan-coil fan with brushless motor or for direct commands from the bus via communication objects

2.4 EK-HU1-TP Technical data

Part number	EK-HU1-TP
Supply voltage (microcontroller logic)	from bus KNX (30 Vdc)
Auxiliary power supply (for output relays)	230 Vac
Auxiliary power supply for electric strike command	SELV, 12 Vdc/ac / 24 Vdc/ac
AI1÷AI2: 2 analog inputs with passive NTC sensor	NTC, 10 kΩ a 25°C
BI1÷BI5: 5 inputs configurable as voltage free contacts	binary imputs, free contacts
AO1: 1 analog outputs 0... 10Vdc	to be combined with high impedance inputs
BO1÷BO4: 4 relay binary outputs, voltage-free contacts	monostable relay, SPST, 5(3) A / 250 Vac
V1-V2-V3: 3 relay binary outputs, voltage-free contacts	monostable relay, SPST, 5(3) A / 250 Vac
1 relay binary output, live contact, for electric strike command	monostable relay, SPST, 5(3) A / 250 Vac
HEAT, COOL: 2 relay binary outputs, voltage-free contacts, for heating and cooling valve control	monostable relay, SPST, 16(10) A / 250 Vac
Current consumption from the bus	< 30 mA
Operating temperature	0°C… +45°C
Degree of protection	IP20
Dimensions	144 x 90 x 60 mm (L x H x P)

3 EK-HO1-TP Switching, display and connection elements

The EK-HO1-TP device has 15 membrane keys on the front panel, 17 signaling LEDs and electrical connection terminals. By switching the device to manual operation via the dedicated menu on the display, it is possible to perform function tests during system commissioning. The button for activating the programming mode with the corresponding LED and the housing for the connection terminal to the KNX bus line are also located on the front panel.

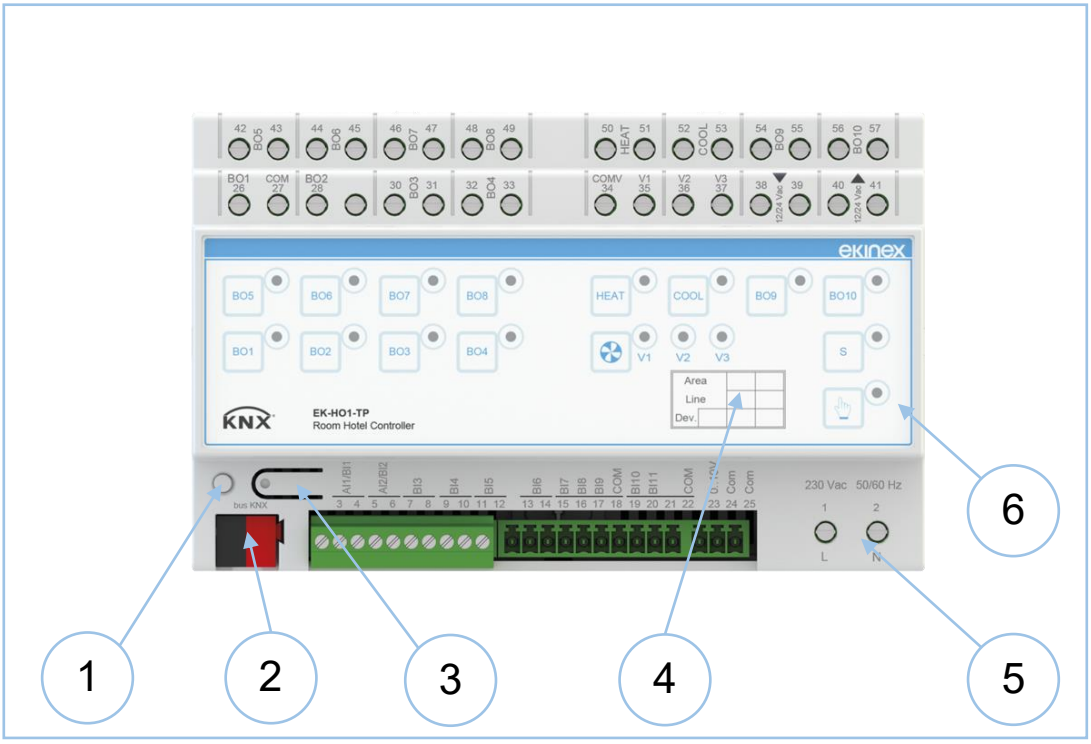


Figure 1 - Switching, display and connection elements version EK-HO1-TP

1. Programming LED 2. Terminal block for KNX bus line 3. Programming pushbutton 4. Registration fields for physical address 5. Terminal block for 230 Vac power supply	6. LED indicator for manual mode active

4 EK-HU1-TP Switching, display and connection elements

The EK-HU1-TP device has 9 membrane keys on the front panel, 11 signaling LEDs and electrical connection terminals. By switching the device to manual operation via the dedicated menu on the display, it is possible to perform function tests during system commissioning. The button for activating the programming mode with the corresponding LED and the housing for the connection terminal to the KNX bus line are also located on the front panel.

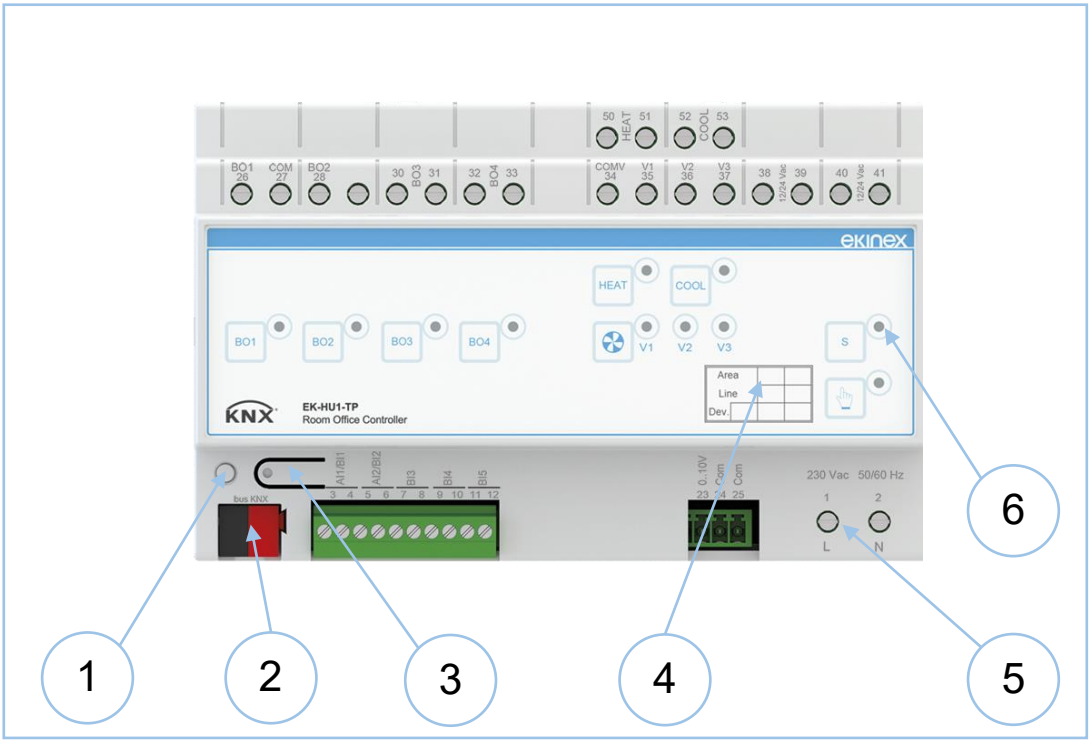


Figure 2 - Switching, display and connection elements version EK-HU1-TP

6. Programming LED 7. Terminal block for KNX bus line 8. Programming pushbutton 9. Registration fields for physical address 10. Terminal block for 230 Vac power supply	6. LED indicator for manual mode active

5 Configuration

The exact functionality of the device depends on the software settings. In order to configure and commission the device you need ETS4 or later releases and the ekinex® application program, APEKHO1TP##.knxprod or APEKHU1TP##.knxprod, which can be downloaded from the ekinex website www.ekinex.com.

The application program allows the configuration of all working parameters for the device. The device-specific application program has to be loaded into ETS or, as alternative, the whole ekinex® product database can be loaded; at this point, all the instances of the selected device type can be added to the project.For every single device, ETS allows to set the operating parameters separately for each function as described in detail in the following chapters.

Product code	EAN	No. of channels	ETS application software (## = release)	Communication objects (max no.)	Group addresses (max no.)
EK-HO1-TP	8018417221217	11	APEKHO1TP##. knxprod	293	254
EK-HU1-TP	8018417221705	5	APEKHU1TP##. knxprod	293	254

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

For further information: www.knx.org.

6 Commisioning

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 rear 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.

7 Function description

The device works as a controlled switch, detecting inputs' statuses and activating the outputs based on received command from bus as KNX frames, according to the temperature controller's logic.

The logic outputs are binary (or digital), i.e. can only assume two values, "On" and "Off"; each output is equipped with a unipolar relay with a contact sized to carry 5 A at 230 Vac.

7.1 Switching on

After connecting the bus line, the device becomes fully functional after a short time (tenths of ms) needed for reinitialization. A delay is programmable for the device to become active on the bus in order to avoid a bus traffic overload during the first moments of start-up of the whole network.

If the auxiliary power is already present (or when it will be), the device is ready.

7.2 Offline operation

The device will be partially functional in case one of the two power supplies (KNX bus line or auxiliary 230 Vac) shoud be missing. The internal circuit dedicated to logic and communication is powered by KNX bus line; output relays, for consumption reasons, are powered by auxiliary supply only.

Should both power supply be off, the device will be completely not functional.

7.2.1 Operation with bus power only

In case of no auxiliary power, all functions not related to outputs are active; however, relay switching will not be active.

In order to detect this normally undesired situation, it is possible to enable a communication object which activates an alarm, so that other devices on the bus can take all proper countermeasures and/or display the anomaly to the user.

In order to show a visual indication of lack of auxiliary power, all LEDs on the panel will blink.

7.2.2 Operation with auxiliary power only

When KNX bus is disconnected, or in case of bus voltage failure (voltage less than 19 V for more than 1 s), all device functions are stopped, particularly those related to timers.

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

7.3 Manual operation

Manual operation constitutes an alternative to input switching through bus commands; this mode is meant for test or maintenance only.

7.3.1 Output status when mode changes

When manual mode is activated, outputs' statuses are not modified. When manual mode is active, the frames coming from the bus do not affect the physical outputs; the output contacts can be switched only if the corresponding membrane key on the front side is pressed.

The manual activation/deactivation of the outputs does not generate any feedback frame. The LED linked to the outputs, however, will continue to display their status nonetheless.

Even when manual mode is deactivated, the actual output status remains unaltered.

From another point of view, the situation could be explained by saying that during manual mode it is like the variables were temporarily "unconnected" from group addresses. When "reconnecting" them (exit from manual mode) their value remains unaltered until a new command from bus does not alter them.

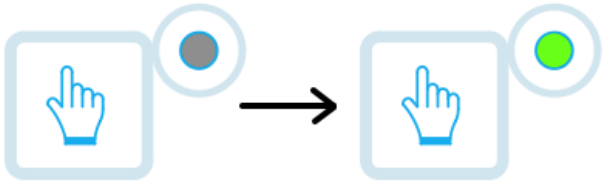
The same considerations made for the command from bus are valid for state changes caused by internal timing functions (e.g. ctivation delays or stairs light function): those state changes have no effect while manual mode is active.

7.3.2 Manual mode activation

The controllers EK-HO1-TP and EK-HU1-TP are fitted with a membrane keyboard to perform manual commands which are useful during commissioning phase.

In order to switch to manual mode you have to proceed as follows:

1) press the manual mode pushbutton on the front side of thee device. During normal operation, this LED is off; when this LED goes on, the membrane keys become active and manual mode is activated;



2) press the key corresponding to the channel you wish to activate (in the example: DO1). Pressing it several times perform a toggle between On and Off states.

3) when testing phase is over, deactivate manual mode by pressing again the manual mode pushbutton. Returning to normal mode, the indicator LED will display the output state again, which will be restored as described.

The switch to manual mode through frontal panel can be inhibited in two ways, both configurable:

- by completely disabling manual mode functionality;
- by a command from bus.

Please note that the command from bus prevent the device from changing mode through the proper pushbutton, but does not actually changes the mode.

If manual mode is not inhibited from configuration nor defined as bus controlled, through a different parameter it is possible to set a timeout after that, if the device has been left in manual mode, it is automatically brought back to normal operation. This prevents the device from being left in an uncontrollable state by mistake.

7.4 Online operation

All functions described below presume that the device has been correctly configured with ETS. An unprogrammed device does not perform any task on the bus; however, it can be activated through the membrane keyboard, making it switch to manual mode.

7.4.1 Software working cycle

The tasks performed by the software are the following:

- update the internal state variables based on KNX frames;
- implement the functions related to timing and other integrated functions to determine the state of the outputs;
- activate the output relays based on the logical outputs' status;
- answer to the requests related to the communication objects received via bus

In addition, there are particular events that can trigger additional features. These events are, for example, a bus power failure or restore, or an ETS new configuration load.

7.4.2 State variables (communication objects)

The device status, with particular attention to its interface elements (outputs) is based on *state variables* which are automatically defined by the application program. When a state variable is assigned to a group address, this variable automatically becomes a KNX communication object; therefore, it inherits all the usual characteristics of communication objects, such as the use of *flags* to determine the impact of the object modification on its bus transmission.

8 Input configuration

The EK-HO1-TP controller has 11 physical inputs, while the EK-HU1-TP office controller has 5 inputs.

The inputs AI1 and AI2 for both devices can be configured as analog inputs for the acquisition of temperature probes (see the paragraph concerning the thermoregulation for an in-depth analysis of possible uses).

All inputs configured as binary inputs can be used as:

- Independent inputs
- Coupled inputs

The couplings allowed by the ETS application are exclusively between contiguous inputs (for example BI1 and BI2). In independent or single mode, each of the inputs operates independently and has its own parameters and communication objects.

In coupled mode, the two inputs are instead grouped for a common functionality; as a consequence, these inputs operate on shared communication objects.

It is possible to configure some inputs as independent and others as coupled, with the association constraints described above.

8.1 Independent or single input

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

8.1.1 Send values or sequences

An event activates the transmission on the bus of a values or sequences of configurable values. These values can be of logical or numerical type with different dimensions.

A sequence can be made up of a maximum of 8 communication objects each of different type and value. Configurable delays can be entered between the sequence values.

8.1.2 Dimming

This mode is used in conjunction with KNX dimmer actuators for controlling luminaires. The function is activated only with long / short pressure events on traditional devices that activate the input contacts of the controllers. At short press, the appliance sends the on / off commands to the dimmer; at prolonged pressure, the percentage of dimming is increased or decreased until release of the contact.

8.1.3 Shutter or venetian blinds

This mode is used in conjunction with KNX dimmer actuators for the control of motorized shutters or rolling shutters or similar.

These actuators have functions for opening and closing the shutters; it is possible to select two types of movements, continuous or at times. Following the input events, the device sends the appropriate telegrams to the actuator. The configuration parameters are as follows:

- if the toggle mode is enabled, each time a certain input is activated, the direction of movement is reversed;
- if the toggle mode is not enabled, the direction is fixed and can be set to "raise" or "lower";

- if the Venetian blind mode is enabled, the appliance sends a "raise / lower all" command for a prolonged pressure, and a "step" (step) to the short pressure; if instead it is disabled, the command for prolonged pressure is the same but at the short pressure a "stop" command is sent.

8.1.4 Scenes

This mode is used in conjunction with KNX units that support the scenario function.

The function allows to memorize and recall a scene setting communication object; in particular, the device sends a "store" or "recall scenario" command to the actuators as a consequence of a short / long pressure event.

The configuration options are as follows:

- Activate the selected scenario with short pressure, and store the current configuration as a selected scenario with a long press;
- Activate a scenario with a short pressure, and another with a long press.

8.1.5 Courtesy light management

The controller for hotels and accommodation facilities EK-HO1-TP easily allows the use of the configuration with independent inputs with sending of values or the scene function to realize timed and customized sequences during the entry and exit of the client from the room.

The configurations mainly concern:

- BI7 input: internal badge holder
- BI8 input: external transponder

In case of use of the controller in combination with KNX standard pushbutton commands, the sending of values and sequences can also be performed by locking the functions during the absence of the customer from the room.

8.2 Coupled inputs

Each pair of coupled inputs can be configured for one of the following functionalities (only the differences with respect to that described for the independent mode are highlighted):

8.2.1 Switching

The two inputs of the pair are connected to the same communication object; unlike the single mode, however, the object can only be of the 1-bit type (on-off), thus constituting a conventional switching. The system integrator can choose which of the two inputs to associate with the "turn on" or "switch off" action.

8.2.2 Dimming

The dimming function uses the long / short pressure events on the inputs for activation. The system integrator can configure which of the two inputs corresponds to the "increase" or "decrease" action.

With a short pressure of a traditional button on the input configured as "increase", the device sends a "turn on" command, while on the other hand the "lower" side sends the "switch off" command.

With a long press, the percentage of dimming is changed to increase or decrease until the input is released.

8.2.3 Shutter or venetian blinds

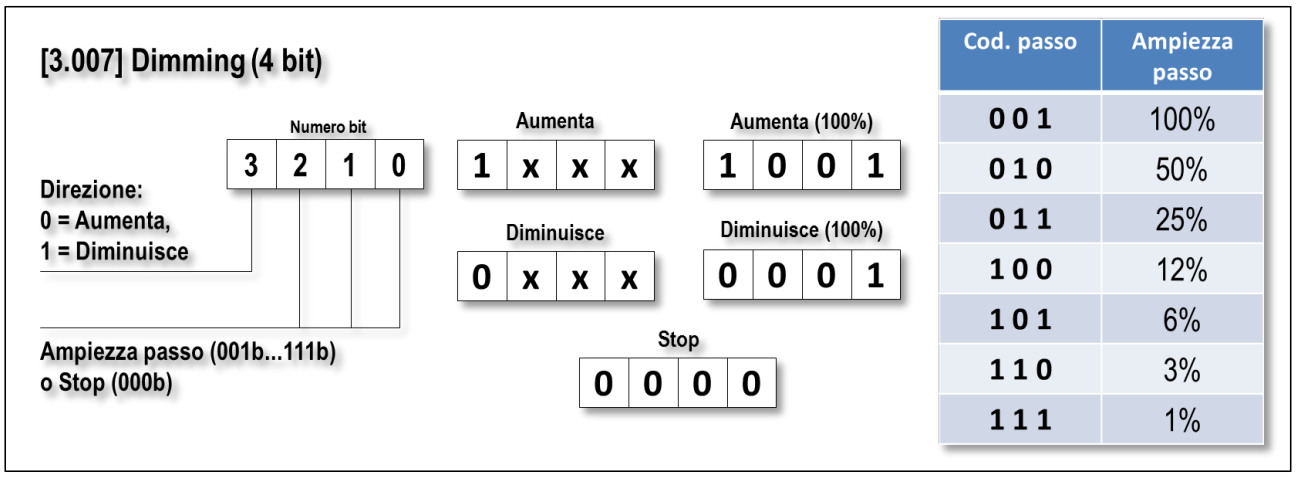
The two coupled binary inputs are assigned to opposite and configurable directions of movement, ie A open / rise and B close / down or vice versa.

It is possible to set the "venetian" mode, which works exactly as for the independent inputs. In the coupled inputs mode, the Scenario functionality is not available.

8.3 Dimmer control

The "dimmer" function is an application profile for devices contemplated by KNX specifications. These specifications define basic requirements related to the interface mechanisms, in addition to which some aspects concerning the operating modes that are specific to the device (be it a control device or an actuator) must be considered.

The "dimmer" type control is essentially based on a 4-bit communication object whose data has the format indicated in the figure:



Cod. passo	Ampiezza passo
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%

The transmission of telegrams containing data of this format communicates to the actuator:

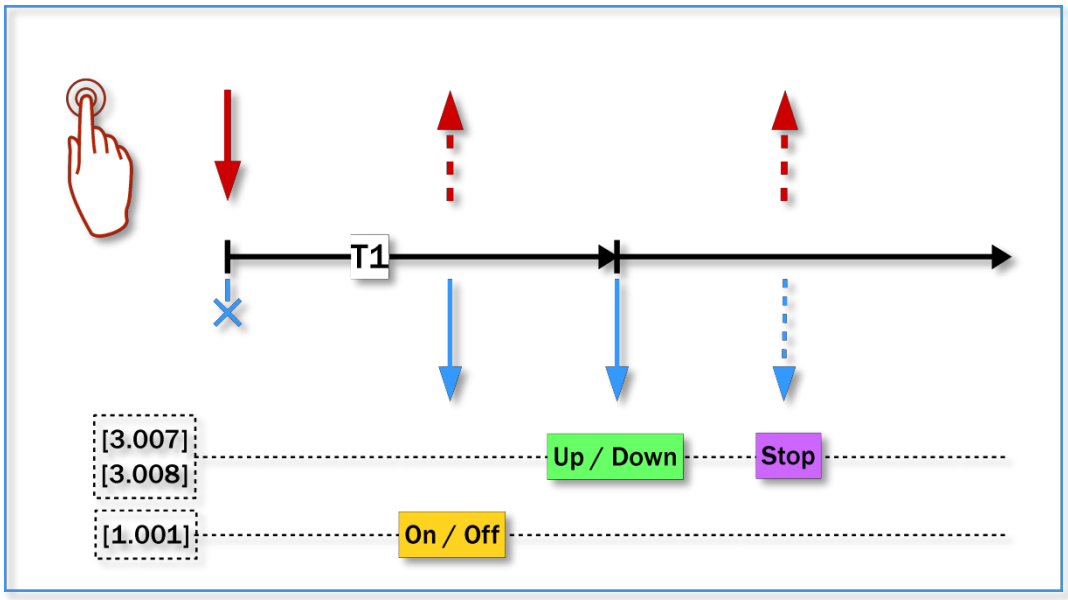
- to make an increase or decrease, of amplitude equal to the specified step,
- an output value,
- to interrupt a variation in progress.

The increase or decrease of the intensity value by the actuator is not instantaneous but gradual; consequently, an increase / decrease command with an interval equal to the maximum possible range has the effect of starting the variation of the intensity in the indicated direction, which will continue until the maximum (or minimum) value is reached. This variation can then be interrupted once the desired intensity value has been reached, by sending a "Stop" command.

It is normally also possible to have the possibility to instantly switch the load on or off (ie bring the intensity instantly to 0% or 100%). To achieve this, we use a command based on another object, of the type "On / Off"; this is nothing more than the same object used for normal load switching, normally present even in the absence of a dimming mechanism.

The controllers define the operations to generate a sequence of these commands in order and with the timing appropriate to obtain the desired command effect.

The operations defined and the associated associated commands are the following:



Short press: instant on / off (toggle on / off on switch object)

Long press: Increase / decrease value up to 100%

Release: Stop increase / decrease.

The same mechanism can be applied for the control of shutters or Venetian blinds (in this case "maximum / minimum intensity" must be replaced with "opening / closing"). For this purpose, the data type [3.8] DPT_Control_Blinds exists, which has the same structure and values as those described above; to control a shutter using the same methods as above, it is therefore possible to connect a communication object [3.7] DPT_Control_Dimming on the command side to an object [3.8] DPT_Control_Blinds on the actuator side. In this case obviously the "On / Off" type object is not used, which allows instant on / off.

8.4 Shutter/venetian blinds control

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

In the case of roller shutters, the actuator carries a mechanical member from one end of travel point to another in a gradual manner, with the possibility of stopping at intermediate points; the command takes place through two lines which, when activated (only one at a time), move the actuator in the corresponding direction.

The Venetian blind is a type of shutter that, in addition to raising / lowering, is also equipped with slats that are opened or closed with the same shutter mode (gradual movement between the two ends). The particularity is given by the fact that normally the movement of the lamellas and that of raising / lowering are controlled with

the same two lines, so that the activation of the electromechanical device must take place according to particular sequences.

The basic control for a shutter or blind is based essentially on a set of communication objects (all 1 bit size):

- [1.008] Move Up / Down (Move Up / Down)
- [1.007] Up / Down Step - Stop (Stop - Step Up / Down)
- [1.017] Unconditional Stop (Dedicated Stop)

The effect of the commands associated with these objects is as follows:

- The "Move" command, upon reception, starts the movement of the shutter in the indicated direction.
- The "Step / Stop" command has two functions: if the shutter is stopped, it takes a step in the indicated direction (the duration is set in the actuator), otherwise it stops the movement in progress and does nothing else.
- The "Stop" command stops the movement in progress.

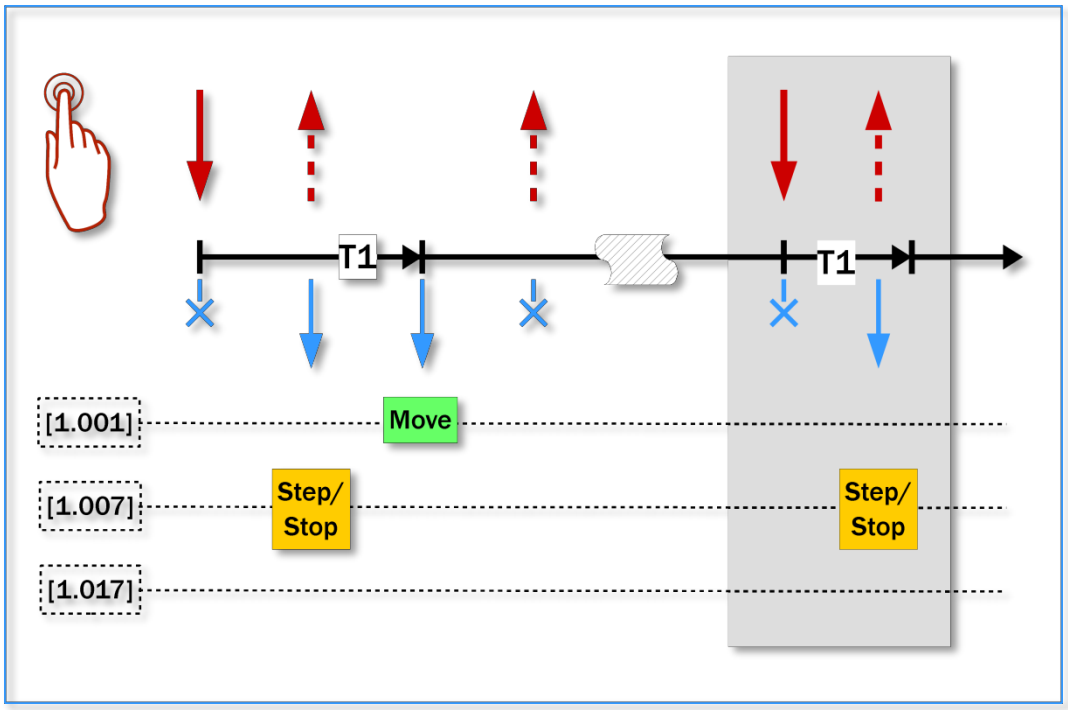
In the simplest version, from the command side:

for the control of a shutter, at least the "Move" and "Stop" objects are required (and present);

for the control of a Venetian blind, at least the "Move" and "Pass / Stop" objects are required (and present).

As regards the operations to be carried out on the control device, in our case the button unit, to generate a sequence of these commands in the order and with the opportune timing, the possible variations are multiple.

In the case of ekinex input devices, two modes are available - referred to as "Shutter" and "Venetian" based on their typical destination - shown in the following figure.

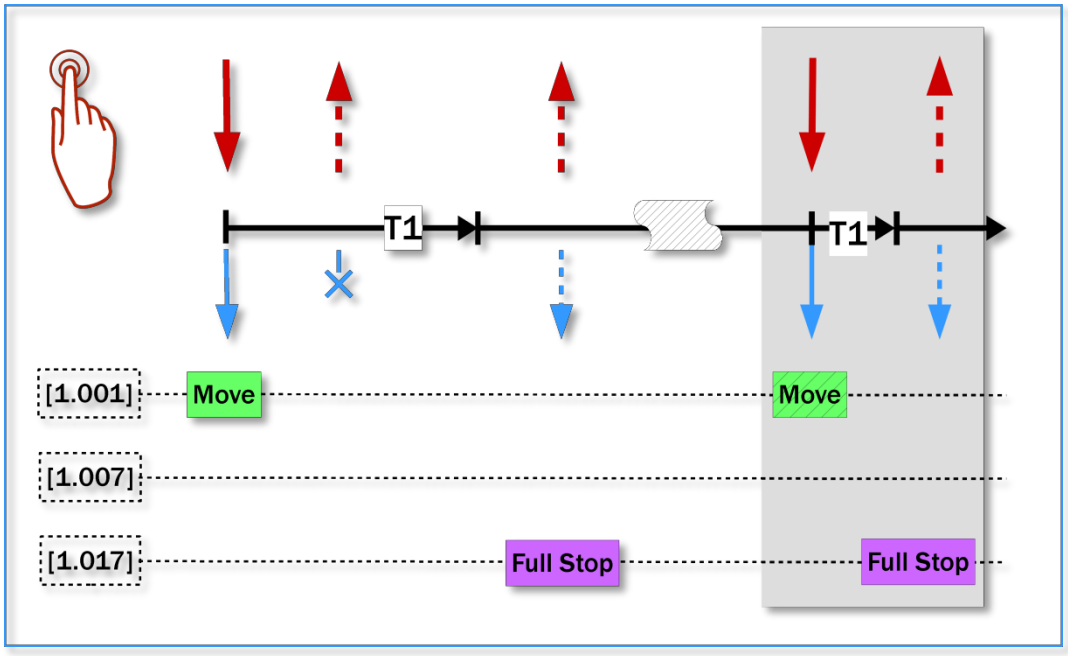


In "Shutter" mode, when a binary input is activated, the shutter starts moving in the corresponding direction (which can be alternatively in the two directions if the input is in independent mode and configured in toggle).

If the entry is released quickly, the shutter will continue running until it is completely closed or opened; you can still stop it by pressing the input again with a long press.

If, on the other hand, the pressure is prolonged, when the input is released, which will occur at the desired intermediate position, the shutter stops.

In "Venetian" mode, at the short pressure of an input (in correspondence with the release) the shutter performs a movement step; this operation is normally used for the regulation of the lamellae.



Keeping the input pressed for a longer time, when a threshold time is reached, a "Move" command is sent, which will bring the shutter up to complete closure or opening. In case you want to stop it at an intermediate point, simply activate the input again with a short press.

9 Output control

he EK-HO1-TP hotel and accommodation controller has 10 relay outputs, while the EK-HU1-TP office controller has 4 outputs. The available outputs can be driven either independently or coupled.
For several of the application functionalities to which the device is intended, the outputs must be managed in pairs: in this case, two coupled outputs form a channel. The outputs that are in an adjacent position can be coupled on the two rows of the output terminal blocks.
To specify the couplings, each output can be configured in two ways: independent (or single) and coupled.

- In independent or single mode, each of the outputs operates independently and has its own parameters and communication objects.
- In coupled mode, two outputs are grouped under the same channel for a common functionality; as a consequence, these outputs operate on shared communication objects. Each output has the possibility of coupling only with the adjacent output.

It is possible to configure some outputs as independent and others as coupled, with the association constraints described above.

9.1 Independent binary outputs

In the simplest application, the presence of a single communication object per channel, "Command On / Off", which switches the channel directly to the value received via a KNX telegram is sufficient.
By configuring the parameters of the device, it is possible to activate some more advanced functions, most of which have an effect on the switching of the outputs.
These functions are as follows:

- Output inversion: allows to associate the contact position (physical) closed to the "On" logical state and vice versa.
- Feedback: automatically transmits the status information to the switching or, if the output is active, at regular intervals.
- Switching delay: allows delaying the actual switching of the contact with respect to the command with a settable delay. Two separate timings are available in opening and closing.
- Operation Lock and Forcing Functions: these functions allow to inhibit the operation of the output or to force its value in different ways.
- Scenario management: allows to recall or set a default value for output in association with scene codes.

9.1.1 Inverted relais command

With this function it is possible to invert the status of the physical contact of the output with respect to the value of the logic output.
Note: *Regardless of this setting, in the continuation of the manual with "On" and "Off" we will always refer to the status of the logic output, not to the state of the relay output contact*.

9.1.2 Feedback

When the feedback is enabled, a communication object corresponding to the output status is made available for reading by other devices on the bus. This object shows the actual status of the logical output, which is probably different from the one set by the command as it includes the effect of any other features currently

active. When this communication object is defined, it is automatically transmitted at each change of state, so that it can generate events at each effective variation of the output. It is also possible to configure the object so that transmission takes place even at regular intervals.

However, the feedback telegrams are not transmitted if the outputs are manually activated.

9.1.3 Switching delay

It is possible to set delays between the status change command of an output and the actual switching. A separate delay time can be set for each transition in activation and deactivation (or, with electrotechnical terminology, in excitation and in de-energization); these times in the following figure are indicated respectively as Ton and Toff. These delays apply to the commutations by direct command and / or logical objects, but not to those caused by other functions (eg staircase lighting or scenario).

9.1.4 Lock output

If the block function is enabled, the activation of a channel can be inhibited by the bus by writing a value in a communication object. The object has the type of datapoint KNX "enable" ("active"); it is important to note that the meaning of this value to On is "active block", not to be confused with "active blocking" or with "active output". In addition, a parameter can be inverted by means of a parameter, so that an "enable = On" value is interpreted as "disable block".

A block output ignores direct switching commands for the duration of the block, keeping (except for the intervention of other functions) the value in force at the time of the block entry. It is possible to assign the output status to a particular value both at the block input and at the block release; it is also possible to indicate whether the blocking status must persist or decline when the power supply is restored after a power failure on the bus.

9.1.5 Forcing function

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

In addition to forcing a desired value, it is possible to establish the value that the output must take on both the release of the forcing and the restart of the power supply after a lack of voltage on the bus, if there is an active forcing at the time of break.

The forcing command has priority both with respect to the blocking function (which acts on the ordinary direct command); this means that a block output can in any case be controlled by forcing commands.

The code of the forcing KNX command is a 2-bit value; the priority bit determines whether the output should be forced, in which case the value bit will be assigned to the output.

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

9.1.6 Scenes

Each output can be associated with a maximum of 8 scene codes; when a scenario with one of these codes is called by a control device, the output will assume the configured value. It is possible to define an additional

delay for the effective activation (or deactivation) of the output with respect to the instant in which the scene is called.

The value to be assigned to the output in correspondence with the scenario can be defined either as fixed (selected during configuration) or reprogrammable by bus through a "save scene" command.

If the latter option is enabled, when the device receives a save command for a scene code associated with the output, the current value of the output itself will be memorized in association with that scenario. This value will be the one that will be recalled in the subsequent activations of the scenario.

9.1.7 Recovery values

As already mentioned, the status of the outputs following some significant events (see paragraph "Resetting the outputs") can be assigned to a value specified in the configuration phase.

The values available for the outputs used as independent are:

- On
- Off
- no changes
- previous value / status *

  (* this option is not available for "bus off" and "after download" events)

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

- "no change" refers to before the event itself (eg for the "bus on" event, if the output was "off" before the bus reset, it remains "off" after reset);
- "previous state" refers to before the condition terminated by the event (eg for the "bus on" event, if the output was "off" before the bus is dropped it remains "off" after reset) ;

For further details refer to the description section of the device settings.

9.2 Coupled binary outputs

In coupled mode, the output channels can drive three categories of actuators: these are grouped under the names of Valves (2 or 3-way), Shutters and Venetian Blinds.

All these categories have a similar mode of operation, ie they move a mechanical device from one end of stroke to another; this can be done in steps, at full stroke, and possibly with the possibility of stopping the race at intermediate points. The listed actuators can be seen, ignoring minor details, as progressively more complex versions of the same basic mechanism; in any case, all are characterized by two pilot lines associated with the two directions.

Each of the individual channels can be associated with one of these categories.

In addition to the main features, these categories of actuators also have common characteristics, such as blocking and forcing functions and scenario management. Several of these characteristics are quite similar to those seen for independent outputs; in these cases, the description will be referred to the relevant sections of the previous paragraph.

9.2.1 Summary on coupled outputs functions

The coupled outputs control is based on three main telegrams, each of which consists of a 1-bit value and therefore can carry two different commands:

Handling Ascent (Opens) / Descent (Closes)	When the telegram is received, the actuator starts moving in the specified direction until the limit switch is reached.
Dedicated stop	When the telegram is received, the actuator stops any movement by stopping at the current position.
Stop - step up / down	This command causes a pulsed movement, that is, steps. It has two different purposes: • if the actuator is at rest, it behaves similarly to the movement command. Upon reception, the actuator moves in the specified direction, but only one step (ie a predefined length by timing); • if the actuator is in motion, it stops at the current position.

In most real systems, and on the other hand defined by KNX standards, the difference between "Movement" and "Step" (apart from the additional stop function of the latter) is only the temporal duration of the movement: in in principle, a "movement" command is simply a "step" command whose duration is guaranteed to be long enough to allow the actuator to reach the limit switch in any case.

Seen in another perspective, the same timing that in the case of the step defines the length of the latter, in the case of the complete movement has the meaning of a timeout that deactivates the output when it is no longer necessary to drive it (obviously these intervals are defined by different parameters). In any case it should be noted that the actuators must always be equipped with electric limit switches to prevent overloads due to the forcing of piloting at the ends of the stroke.

Since mechanical feedback is not available to determine the position of the actuator, this is determined by timing the movements: given the exact duration of the stroke between the two extremes, a movement expressed as a percentage fraction of the total stroke will correspond with an excellent approximation to the same percentage of total travel time. The device maintains an internal position counter which is resynchronized whenever it is known that the actuator reaches one of the ends (following a complete movement command).

Because the timing above - to be used for driving the output contacts - is correct, the total travel time must be measured and set by means of a parameter.

The one just made is a general description of base: real examples of actuators may not have the same possibilities of control (for example, they may not have the possibility to stop in different positions from the two extremes) or may have further options and functionality. These will be illustrated below in the specific description of the actuator categories.

9.2.2 Shutter blinds control

The shutter control is the one most similar to the generic one described in the paragraph "general"; its description also applies exactly to the valve with 3-way actuator.

The commands used are the three fundamental; however, the "Stop / Step" command, implemented because required by KNX specifications, actually has the only effect of "Stop" (it has no effect if the actuator is at rest), thus being equivalent to the third command already available.

The standard way of controlling a damper channel is therefore as follows:

- send the "Movement" command with the desired direction to set the damper in motion;
- let the damper reach the extreme position (the output will be deactivated when a timeout expires) or send a "Stop" or "Step / Stop" command when the damper has reached the desired position.

To better exploit the possibilities of intermediate positioning, the control has further ways to specify the position of the actuator:

- the "absolute position" can be specified (expressed as a percentage); a feedback value is available for the current position and a "position reached" telegram;
- a "dimmer" type control can be enabled as shown in the following figure. Refer to the section describing the parameters for more details.

As already mentioned, it is necessary to specify the total travel time by means of a parameter; for this purpose there are two parameters, one for the rising direction and the other for the descent. In fact, the times are probably different in the two cases due to mechanical reasons (eg heavy shutters).

The time to be specified is the actual and precise running time from extreme to extreme, which will be used for timing calculations; in the cases in which a movement must be made that guarantees the reaching of the limit switch, the duration of the movement will be set equal to 120% of the specified value.

Another parameter that must be defined is the inversion time, that is the pause that must occur between two movement commands in different directions sent in succession. The purpose is to allow the damper to stop and reverse the motion correctly without excessive stress on the mechanical parts.

9.2.3 Venetian blinds control

The Veneziana differs from the shutter / damper due to the presence of the fins, whose management requires some additional parameters.

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

- the "Step" command is now actually used. A step movement refers to the fins (not to the up / down movement); there is a specific parameter to define the time associated with the step, ie the activation time of the outputs that causes the movement relative to the desired step;
- an additional set of communication objects is available for setting and reading the absolute position of the fins, in addition to a "position reached" object;
- an object for dimmer control can also be enabled for the fins.

Since the fins also have their own positioning function, as for the up / down movement there is a parameter to configure the total travel time (but this time it is unique, given that no significant mechanical asymmetries are to be expected). Another internal position counter is then maintained to guarantee even in this case the best possible positioning performance.

The common Venetian actuators control both the up / down movement and the inclination of the fins using the same two pilot lines; to allow the separation of movements, these are used as described below. Note that this is the description of a mechanism of principle (however realistic) for illustrative purposes; Actual actuators could achieve the same functionalities with different or more sophisticated mechanical solutions, without prejudice to the control mechanism.

Let us first assume that the Venetian blind is in a completely closed position (including wings). By activating the opening command line, the motor starts to bring the flaps into open position; when these have reached their end of stroke, the further action of the engine drags the Venetian panel upwards.

Imagining now to stop the Venetian blind halfway, we obtain a partially open panel with completely open wings; of course we could resume the climb up to full opening. However, if we now activate the closing command line, the fins will first begin to close; the panel will not start moving until the wings are completely closed and in turn will drag the panel downwards.

If the activation time of the closing command line had been short, ie not long enough to let the fins reach the completely closed position, we would have obtained a condition in which the panel would remain in the same position, but we would have obtained a different regulation of opening of the fins. Furthermore, in this condition, by alternating the activation of the two command lines, the fins can be brought to any intermediate position provided they do not exceed their end of stroke (in which case the panel will start to move in raising or lowering).

9.2.4 Lock outputs

The blocking function is similar to the case of independent outputs; the only significant difference lies in the wider range of values for the position of the actuator compared to the case of simple binary outputs. In particular, the possible values include the two end-of-travel ends, an arbitrary programmed position, or the position that the actuator had before the block, in addition to the stopping of any movement in progress.

For further details refer to the configuration description section.

9.2.5 Forcing function

The forcing function is similar to the case of independent outputs; the considerations made just above regarding the blocking function are valid.

For each of the three alarms available, you can define a separate timeout for the heartbeat function. If no alarm information telegram is received for a time duration exceeding the timeout, the alarm is assumed to be active and the actuator is controlled accordingly; a timeout that occurs when an active alarm has no effect.

The heartbeat function can of course be disabled; note, however, that if it is enabled, the device that originates the alarms must be configured to support the periodic sending with times compatible with the value chosen for the timeout.

9.2.6 Scenes

Scenario management is completely analogous to the case of independent outputs; the same considerations made above for the blocking function apply here.

The values that can be assigned to the outputs are specified in terms of the damper position; in particular, there are the two stroke ends, a specified intermediate position, or the stop (the activation of the scenario interrupts only any movements in progress).

9.2.7 Recovery values

As already mentioned, the status of the outputs following some significant events (see paragraph "Resetting the outputs") can be assigned to a value specified in the configuration phase.

The values available for the outputs used in coupled mode are:

- None (remains unaffected)
- Ascent / opening
- Descent / closure
- Stop
- Move to position

9.3 Electric door lock control

Both EK-HO1-TP and EK-HU1-TP controllers have an additional voltage-controlled output for controlling an electric lock. The output is externally powered (see the technical sheet for details of the terminals for external power supply), depending on the model of electric lock used. The pulse duration on the DL output can be configured in the ETS application according to the particular needs, with a resolution of 1 ms. The electric lock command can be automated through a communication object on the bus.

10 Heating, cooling and ventilation

10.1 Operation as actuator: choosing the control variable type

When operating as actuator, in order to make the integration with the temperature controller device easier, it is possibleto use several Data Point Types for control communication objects. Fan speed control can be performed through the following type, which can be selected in the ETS application:

- [DPT 1.001] switch – Each speed is associated to a 1-bit communication object; the communication objects are mutually interlocked by the software. The speed associated to the communication object modified during the last receiving event prevails.
- [DPT 5.010] counter pulses – This 1-byte communication object can assume several values based on selected discrete speed (0=OFF, 1=Speed1, 2= Speed2 and 3= Speed3) or continuous speed (0=OFF, 1=20%, 2=40%, 3=60%, 4=80% 5=100%). Vaues of the communication object which are coherent with the performed configuration (1-3 or continuous speed) are not taken into considerations by the actuator.
- [DPT 5.001] percentage (0 ..100%) – This 1-byte communication object allows actuating both a 3-speed fan (the percentage speed thresholds are set in the application software) and a continuous fan with 0-10V control signal.

In convector applications, the heat exchange coils command in realized through 1-bit communication objects ([DPT 1.001] switch): the temperature controller can send both ON/OFF and PWM commands. Otherwise, in fan-coil applications the coil command can be single and separated from the fan command: in case of single command, the valves open when at least Speed1 is selected; in case of separated commands, the same considerations of convector applications apply.

Diagrams displaying the transfer functions between percentage command value and discrete set speed or continouous speed percentage (linearized on 0-10V control signal) are shown in *Fan* section, inside the chapter about ETS application program and use of the device as actuator.

10.2 Control variable timeout alarm

In order to guarantee the reliability of the frame exchange between controller and actuator on the bus, it is possible to add a time check when every command is received: when the preset time expires, if no new commands are received, the actuator's outputs can be forced in predefined positions.

When setting a timeout different from 0, make sure to set cyclic sending of commands on the device acting as temperature controller. In order to work correctly, the cyclic sending must assume values less than the preset timeout.

10.3 Operation as controller and actuator

The room air temperature control is performed thanks to the intercept valve(s) on the heat exchange coil(s), with an ON/OFF or PWM control algorithm. In order to control the intake air flow, different modes are available, according to the control needs and the kind of actuator used for the fan unit.

10.4 Control algorithms

10.4.1 1-2-3 speed ON/OFF control

It is the most common algorithm used to control air flow in air terminal units and it is available in combination with fan units equipped with an asinchronous motor with 3 independent windings. In case of motors with 5

windings, it is suggested to connect 3 windings to the minimum, medium and maximum speed according to the treated air flows. It is also possible to configure the algorithm with 1 or 2 speeds only.

This simple algorithm is also used in devices with a 0-10V control output: in ETS applicaton program, it is possible to set the output percentage linked to each air flow threshold.

The algorithm perform an ON/OFF control on 3 different air flow windows based on the error between the setpoint temperature and the actual measured room temperature. The threshold values, as well as the activation hysteresis, do not change from heating to cooling mode. However, the error has a different meaning in those 2 conduction modes:

- Heating: *error = (T setpoint – T measured)*
- Cooling: *error = (T measured – T setpoint)*

10.4.2 Proportional control with continuous output

This algorithm is available in combination with fan units equipped with a brushless motor and 0-10V control signal. This algorithm performs a more accurate control of the room temperature; the power consumption of the actuator is also reduced, as well as the noise caused by the fan unit rotation. To avoid a temperature error while at steady condition, a minimum rotation speed is applied, as well as an hysteresis cycle for reboot after shutdown, when the setpoint temperature has been reached.

10.4.3 Proportional-integral control with continuous output

It is a variation of the previously explained algorithm, which is available in combination with fan units equipped with a a brushless motor and 0-10V control signal. Compared to the proportional control with continuous output, a contribution which is proportional to the integral of the temperature error is added; this allows an even more accurate control, especially when combined with PWM control of the valves on heat exchange coils. Please note that this solution makes the fan unit work continuously.

Diagrams displaying the transfer functions between temperature error and discrete set speed or continouous speed percentage (linearized on 0-10V control signal) are shown in *Fan* section, inside the chapter about ETS application program and use of the device as controller.

10.5 Setpoint management

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

Three setpoint management modes are foreseen:

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

10.5.1 Single setpoint mode

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

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

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

10.5.2 Relative setpoints mode

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

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

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

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

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

10.5.3 Absolute Setpoint mode

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

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

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

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

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

10.6 Operating modes

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

- Temperature setpoint
- Building protection setpoint

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

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

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

Through ETS application program, it is possible to assign 2 different setpoint values to each operating mode, for comfort and building protection level, and two different attenuation levels for stand-by and economy, corresponding to both heating and cooling.

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

10.7 Heating/cooling switch over

The switch over between both heating and cooling mode can take place in 3 ways:

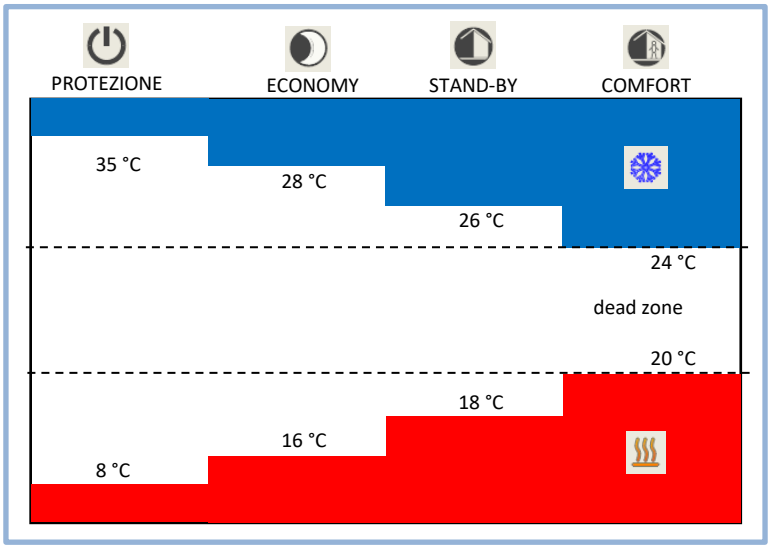
1. from KNX bus, through a communication object;
2. automatically, based on the room temperature;
3. automatically, based on the heat exchange coil temperature

10.7.1 Switchover from bus

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

10.7.2 Automatical switch over, based on the room temperature

Mode 2 is suitable for applications with heating / cooling systems with a 4-pipe configuration. In addition, the information can be transmitted on the bus through an output communication object [DPT 1.100 heat/cool]; the difference with mode 1 is that the switch over is performed automatically by the machine, basing on the values of current temperature and setpoint. The automatic switch over is achieved by introducing a dead band as shown in the following figure.



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

10.7.3 Automatical switch over, based on the heat exchange coil temperature

In case of heating / cooling systems with a 2-pipe configuration, it is possible to set an automatic conduction mode switch over (from heating to cooling and vice versa) by measuring the heat exchange fluid temperature with a proper temperature sensor, which has to be installed on the intake pipe of the heat exchange coil and needs to be connected to an analog input on the device (or to another KNX device equipped with an analog input).

In order to achieve that, 2 temperature thresholds are defined: if the heat exchange temperature is less than a *low threshold* (e.g. 18°C), the device switches to cooling mode; if it is greater than a *high threshold* (e.g. 28°C), the device switches to heating mode

10.8 Temperature control alarm

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

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

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

10.9 External inputs and inputs from bus

The devices are equipped with 2 inputs freely programmable as analog or binary. Moreover, when using the device as integrated temperature controller, variables acquired by bus through communication objects are also available. All external and bus inputs allow extending the device functionality.

10.9.1 Room temperature or return air temperature for control

In case no external controller is used (for example a KNX room thermostat), the device can alternatively use:

1) the temperature value of the air mass measured by a sensor connected to an analog input, installed on an internal wall at approx. 1,50 m.
2) the temperature value measured by a sensor connected to an analog input and positioned next to the return air grid.

10.9.2 External climate compensation

When the integrated controller is in cooling mode, if an external temperature sensor is connected to an analog input or if a value from bus is acquired from bus through a communication object, it is possible to perform a climate compensation on the desired room temperature.

This compensation allows automatically raising the desired temperature as the external summer climate becomes too warm, thus avoiding discomfort when passing from outside to inside. The curve is set first by selecting an initial external temperature value and then selecting the growth gradient of the desired temperature.

10.9.3 Heat exchange coil temperature for hot and cold start

In both heating and cooling modes, in order to avoid the possible discomfort caused by an air flow whose temperature is too different from the actual one, the device does not start the fan unit until the heat exchange fluid inside the heat exchange coil has not reached a suitable temperature value. This situation normally happens at first start or after long inactivity periods. This function can be performed through:

1) a temperature control through temperature sensor (minimum/maximum sensor) installed on the heat exchange coil of the fan-coil unit;
2) a delayed start through a proper time interval setting (approximation function).

10.9.4 Minimum/maximum sensor.

In the first case, the temperature of the heat exchange fluid flowing inside the coil is measured: the function performs an actual temperature control. For the operation it is necessary that the minimum/maximum sensor is connected to an analog input. Alternatively, the temperature value can be received via bus from a different KNX device equipped with analog inputs, where the sensor is connected.

10.9.5 Antistratification temperature

Fan-coil units are realized in different costruction shapes for floor, wall or ceiling installations. In particular cases, such as rooms with height and volume much higher than usual (atriums, fitness facilities, commercial buildings, etc.) during heating season warm air can accumulate in the highest part of the room; the air stratification phenomenon can cause energy waste and discomfort.

In order to find a remedy for this situation, the device is equipped with an antistratification function, which forces the fan unit in first speed. This function requires 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.

10.9.6 Water intake temperature for automatic heating/cooling switch over

The heat exchange fluid temperature measurement can be done by means of a proper temperature sensor to be installed next to the intake pipe of the heat exchange coil, connected to an analog input of the device (or to a different KNX device equipped with analog inputs).

10.9.7 Generic temperature acquisition

The analog input can be used to acquire a generic temperature value by means of a traditional NTC (10 kΩ at 25°C) sensor. The measured value can be sent on the bus and used by other KNX devices, for example for display purposes or calculation of a weighted mean average by a room thermostat.

10.10 Window contacts

In order to realise 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 actuator/controller fan-coil 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.

10.11 Presence sensors

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.

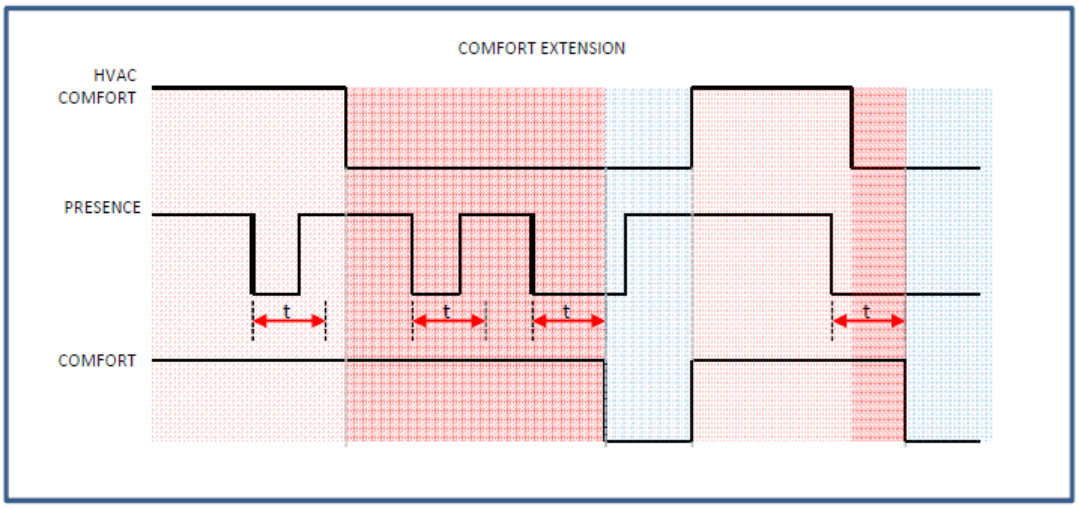
The occupancy status detection is performed by presence sensors which can be connected to KNX devices equipped with binary inputs; the actuator/controller for fan-coil units exposes up to 2 1-bit communication objects which can be synchronized to the situations detected by the sensors. An internal logic performs a

logical OR or the states of the connected sensors: in order to activate the energy saving function, at least one sensor needs to detect a presence. 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.



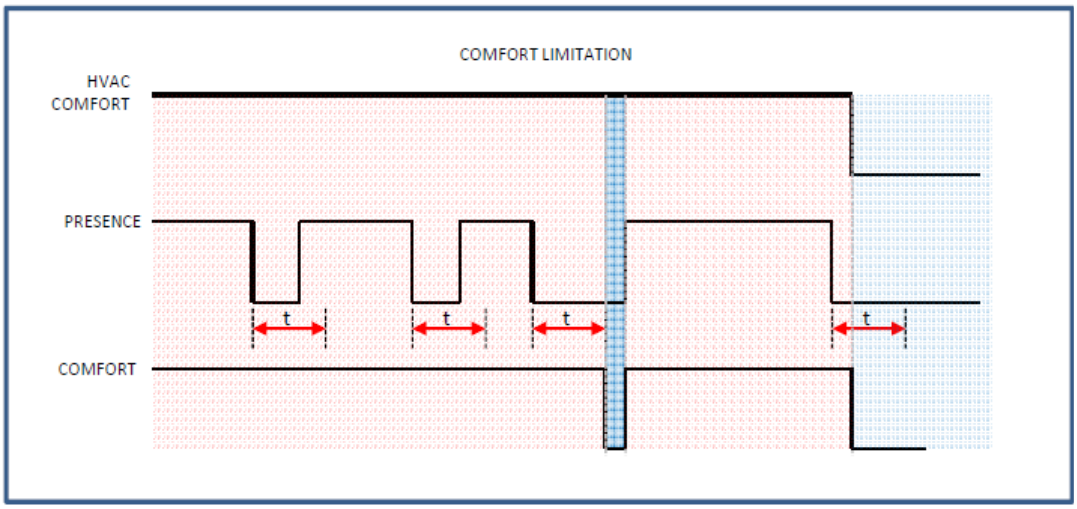
The figure above shows that, even if a presence is detected while the operating mode forced by the time scheduling function is not comfort, no change of operating mode is performed until the next programmed comfort event.

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

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

Comfort limitation. This function is only active if the actual operating mode is set on comfort; if, during this time, a presence is not detected for a time period greater than a preset time, the operating mode shifts to economy

or standby. The attenuation modes can be selected in the application program and are independent from the modes foreseen by the time scheduling function.



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.

10.12 Drip tray control

When in cooling mode, drip water collected inside the proper basin can be discharged by means of a discharge pump. When the level inside the basin reaches the safety threshold, a proper communication object changes its state; this communication object can be used as standalone or in logical OR with other objects of the same type, in order to activate a binary output that activates the drip tray discharge pump.

10.13 Filter monitoring

Fan-coil units are equipped with a filter that absorbs and hold the suspended dust before air is sent into the environment. The filter is extractable for cleaning and substitution operations. In order to execute the monitoring function, the device is equipped with a working hour counter; the fan group needs to be set at least to 1st speed in order for the counter to work. When the time interval set in the proper parameter has been reached, a communication object warning about the filter subsititution is issued. The same object can be used to suppress the warning and simultaneously reset the counter.

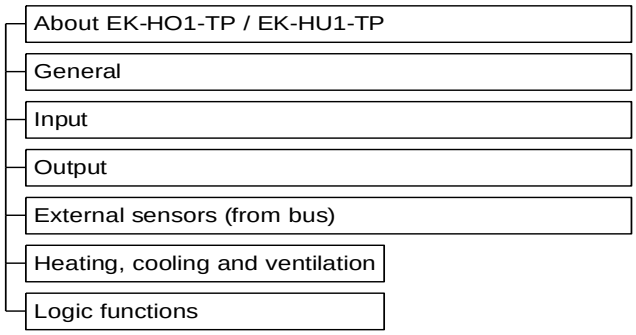
10.14 Valve protection

The thermal plant where the fan-coil units are installed uses water as heat exchange fluid and units are equipped with motorized valves to intercept the hydraulic circuits. Under particular conditions, long inactivity periods can block those valves: to prevent this problem, the device can periodically activate an open/close cycle for the valves.

To perform such function, the device is equipped with a separated counter for each valve, which is activated every time the actuator completely closes the valve. When that counter reaches the value set in parameter *Frequency*, the valve is opened to prevent it from getting stuck. The duration of this opening depends on the value set in parameter *Time interval*. If the actuator brings the valve to fully open position before tah time interval is reached, the counter is reset and then rebooted when the valve is closed again. The valve protection function is available when the device is configured both as actuator and as controller.

11 Application program for ETS

At its opening, the application program tree includes the following main items:



11.1 About EK-HO1-TP

The Info tab on the EK-HO1-TP is for informational purposes only and contains no parameters to set. The information shown is:

© Copyright Ekinex S.p.A. 2019
Application software for ETS4 & ETS5
Version 1.00 (or later)
Hotel Room Controller - KNX EK-HO1-TP

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

11.2 About EK-HU1-TP

The Info tab on the EK-HU1-TP is for informational purposes only and contains no parameters to set. The information shown is:

© Copyright Ekinex S.p.A. 2019
Application software for ETS4 & ETS5
Version 1.00 (or later)
Office Controller - KNX EK-HO1-TP

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

11.3 General

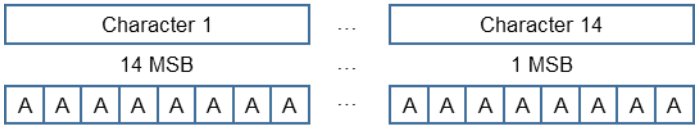
In the General tab the settings are made that concern:

- Manual operation setting parameters
- Delay after bus voltage recovery
- Alarm activation

Parameter name	*Conditions*	*Values*
Manual operations		disabled /**enabled**
Disable from bus	Manual operations = enabled	**no** / si
Restore auto mode time (0 means no automatic restoring)	Manual operations = enabled and Disable from bus = no	**00:15:00** hh:mm:ss [range 00:00:00 … 18:12:15]
[…]		
Delay after bus voltage recovery		**00:00:05** hh:mm:ss [range 00:00:00 … 18:12:15]
	The parameter sets the delay between the instant in which the bus line is powered and the moment when the transmission of data by the device begins (sending status feedback, control output, etc.). This delay must be carefully planned to avoid that after a power failure of the bus power supply line and subsequent reset, all the devices simultaneously start sending telegrams, causing excessive occupation of the available signal band.	
[…]		
Technical alarm		disabled /**enabled**
Device power off alarm		disabled /**enabled**

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Manual mode active	Manual operations = enabled	1 Bit	R-CT--	[1.3] DPT_Enable	1
Disable front pushbuttons	Manual operations = enabled and Disable from bus = yes	1 Bit	-WC---	[1.2] DPT_Bool	2
Technical alarm	Technical alarm = enabled	1 Bit	R-CT--	[1.5] DPT_Alarm	3
Communication alarm		1 Bit	R-CT--	[1.5] DPT_Alarm	4
Device power off alarm	Device power off alarm = enabled	1 Bit	R-CT--	[1.5] DPT_Alarm	5
Thermal generator lock alarm	Enabled from Heating, cooling and ventilation tab	1 Bit	R-CT--	[1.5] DPT_Alarm	6
Temperature control alarm		1 Bit	R-CT--	[1.5] DPT_Alarm	7

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Filter change warning	Enabled from Heating, cooling and ventilation tab	1 Bit	R-CT--	[1.5] DPT_Alarm	8
Drip tray status	Enabled from Heating, cooling and ventilation tab	1 Bit	R-CT--	[1.5] DPT_Alarm	9
Alarm text		14 Bytes	R-CT--	[16.0] DPT_String_ASCII	10



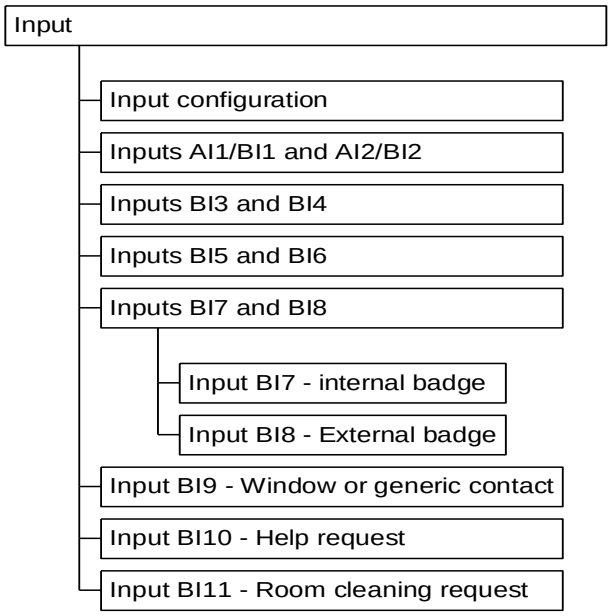
This Data Point Type is used to transmit the alarm signal as a sequence of ASCII characters. The maximum string length is fixed at 14 characters (14 octets). Content is transferred starting from the most significant character (14 MSB). If the string to be transmitted is shorter than 14 characters, the unused bytes are filled with the NULL character (00h).
Example: "EKINEX is OK" is represented as follows:

45h 4Bh 49h 4Eh 45h 58h 20h 69h 73h 20h 4Fh 4Bh 00h 00h

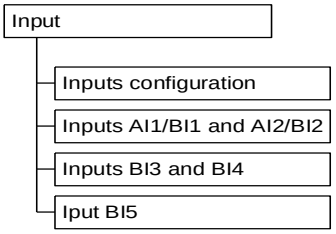
11.4 Inputs

The following parameters are set in the Inputs tab:

- For inputs 1-2 it is possible to select the analog acquisition mode for temperature probes
- For all inputs in binary mode, it is possible to configure them as single or coupled inputs
- The single binary inputs can be used to send values or sequences, for dimming, to activate blinds, blinds or blackout elements and to send scenarios on the bus
- The coupled binary inputs can instead be used to perform a switching, by dimming or by driving blinds, blinds or blackout elements



EK-HO1-TP



EK-HU1-TP

11.4.1 Input Configuration

Parameter name	*Conditions*	*Values*
Inputs BI1 and BI2		**disabled** independent or single coupled
Input AI1/BI1	Inputs BI1 and BI2 = independent or single	**disabled** enabled DI enabled NTC
Type	Inputs BI1 and BI2 = independent or single and Input AI1/BI1 = enabled DI	**send values or sequences** dimming shutter or venetian blinds scene
Type	Inputs BI1 and BI2 = independent or single and Input AI1/BI1 = enabled NTC	**[AI] room temperature sensor** [AI] outdoor temperature sensor [AI] heat exchange coil battery temperature sensor [AI] antistratification temperature sensor [AI] flow temperature sensor [AI] generic (NTC) sensor
Input AI2/BI2	Inputs BI1 and BI2 = independent or single	**disabled** enabled DI enabled NTC
Type	Inputs BI1 and BI2 = independent or single and Input AI2/BI2 = enabled DI	**send values or sequences** dimming shutter or venetian blinds scene
Type	Inputs BI1 and BI2 = independent or single and Input AI2/BI2 = enabled NTC	**[AI] room temperature sensor** [AI] outdoor temperature sensor [AI] heat exchange coil battery temperature sensor [AI] antistratification temperature sensor [AI] flow temperature sensor [AI] generic (NTC) sensor
Type	Inputs BI1 and BI2 = coupled	**switching** dimming shutter or venetian blind
[…]		
Inputs BI3 and BI4		**disabled** independent or single coupled
Input BI3	Inputs BI3 and BI4 = independent or single	**disabled** / enabled

Parameter name	*Conditions*	*Values*
Type	Inputs BI3 and BI4 = independent or single and Input BI3 = enabled	**send values or sequences** dimming shutter or venetian blinds scene
Input BI4	Inputs BI3 and BI4 = independent or single	**disabled** / enabled
Type	Inputs BI3 and BI4 = independent or single and Input BI4 = enabled	**send values or sequences** dimming shutter or venetian blinds scene
Type	Inputs BI3 and BI4 = coupled	**switching** dimming shutter or venetian blind
[…]		
Inputs BI5 and BI6		**disabled** independent or single coupled
	(only for EK-HO1-TP; for EK-HU1-TP only Input BI5 is available)	
Input BI5	Inputs BI5 and BI6 = independent or single	**disabled** / enabled
Type	Inputs BI5 and BI6 = independent or single and Input BI5 = enabled	**send values or sequences** dimming shutter or venetian blinds scene
Input BI6	Inputs BI5 and BI6 = independent or single	**disabled** / enabled
	(only for EK-HO1-TP)	
Type	Inputs BI5 and BI6 = independent or single and Input BI6 = enabled	**send values or sequences** dimming shutter or venetian blinds scene
Type	Inputs BI5 and BI6 = coupled	**switching** dimming shutter or venetian blind
[…]		
Input BI7		**disabled** / enabled
	(only for EK-HO1-TP)	
Type	Input BI7 = enabled	**send values or sequences** dimming shutter or venetian blinds scene

Parameter name	*Conditions*	*Values*
Input BI8		**disabled** / enabled
	(only for EK-HO1-TP)	
Type	Input BI8 = enabled	**send values or sequences** dimming shutter or venetian blinds scene
[…]		
Input BI9		**disabled** window contact generic contact
	(only for EK-HO1-TP)	
[…]		
Input BI10		**disabled** / enabled
	(only for EK-HO1-TP)	
[…]		
Input BI11		**disabled** / enabled
	(only for EK-HO1-TP)	

11.4.2 AI1/AI2 – Single inputs: temperature acquisition

The tab allows you to make the following settings:

- Temperature probe function (linked to specific thermoregulation functions)
- Activation of 2 maximum or minimum comparators
- Enabling of the comparator threshold that can be modified via communication objects (this function makes it easy to introduce additional temperature on / off controls, for example additional zones)

Parameter name	*Conditions*	*Values*
Input AI1 type		**[AI] room temperature sensor** [AI] outdoor temperature sensor [AI] heat exchange coil battery temperature sensor [AI] antistratification temperature sensor [AI] flow temperature sensor [AI] generic (NTC) sensor
Filter type	Input AI1 = [AI1] …	low **medium** high
	Values settable: *Low = average value every 4 measurements* *Medium = average value every 16 measurements* *High = average value every 64 measurements*	
Temperature offset	Input AI1 = [AI1] …	**0°C** [range -5,0°C … +5,0°C]
Minimum change of value to send [K] (0 means no values sent on change)	Input AI1 = [AI1] …	**0,5** [range from 0 to 5]
	If set to the value 0, no value is sent to the change.	
Cyclic sending interval	Input AI1 = [AI1] …	**no sending** [other values in the range 30 s … 120 min]
Threshold 1	Input AI1 = [AI1] …	**not active** / below / above
Value [°C]	Input AI1 = [AI1] … Threshold 1 = below or above	**7** [range from 0 to 50]
Temperature threshold 1 enabled from bus	Input AI1 = [AI1] …	**disabled** / enabled
Threshold 2	Input AI1 = [AI1] …	**not active** / below / above
Value [°C]	Input AI1 = [AI1] … Threshold 1 = below or above	**45** [range from 0 to 50]
Temperature threshold 2 enabled from bus	Input AI1 = [AI1] …	**disabled** / enabled
Hysteresis	Input AI1 = [AI1] … Threshold 1 = below or above Threshold 2 = below or above	**0,4 K** [other values in the range 0,2 K … 3 K]

Parameter name	*Conditions*	*Values*
Cyclic sending interval	Input AI1 = [AI1] … Threshold 1 = below or above Threshold 2 = below or above	**no sending** [other values in the range 30 s … 120 min]

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input AI1 Temperature value		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	11
Input AI1 outdoor temperature value		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	11
Input AI1 heat exchange coil battery sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	11
Input AI1 antistratification temperature sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	11
Input AI1 flow temperature sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	11
Input AI1 generic temperature sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	11
Input AI1 Temperature threshold 1 – Switch	Threshold 1 ≠ not active	1 Bit	R-CT--	[1.1] DPT_Switch	12
Input AI1 Temperature threshold 2 – Switch	Threshold 2 ≠ not active	1 Bit	R-CT--	[1.1] DPT_Switch	13
Input AI1 Temperature threshold 1 – Value	Threshold 1 ≠ not active and Temperature threshold 1 enabled from bus = enable	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	14
Input AI1 Temperature threshold 2 - Value	Threshold 2 ≠ not active and Temperature threshold 2 enabled from bus = enable	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	15

Parameter name	*Conditions*	*Values*
Input AI2 type		**[AI] room temperature sensor** [AI] outdoor temperature sensor [AI] heat exchange coil battery temperature sensor [AI] antistratification temperature sensor [AI] flow temperature sensor [AI] generic (NTC) sensor
Filter type	Input AI2 = [AI2] …	low **medium** high
	Values settable: *Low = average value every 4 measurements* *Medium = average value every 16 measurements*	

Parameter name	*Conditions*	*Values*
	High = average value every 64 measurements	
Temperature offset	Input AI2 = [AI2] …	**0°C** [range -5,0°C … +5,0°C]
Minimum change of value to send [K] (0 means no values sent on change)	Input AI2 = [AI2] …	**0,5** [range from 0 to 5]
	If set to the value 0, no value is sent to the change.	
Cyclic sending interval	Input AI2 = [AI2] …	**no sending** [other values in the range 30 s … 120 min]
Threshold 1	Input AI2 = [AI2] …	**not active** / below / above
Value [°C]	Input AI2 = [AI2] … Threshold 1 = below or above	**7** [range from 0 to 50]
Temperature threshold 1 enabled from bus	Input AI2 = [AI2] …	**disabled** / enabled
Threshold 2	Input AI2 = [AI2] …	**not active** / below / above
Value [°C]	Input AI2 = [AI2] … Threshold 1 = below or above	**45** [range from 0 to 50]
Temperature threshold 2 enabled from bus	Input AI2 = [AI2] …	**disabled** / enabled
Hysteresis	Input AI2 = [AI2] … Threshold 1 = below or above Threshold 2 = below or above	**0,4 K** [other values in the range 0,2 K … 3 K]
Cyclic sending interval	Input AI2 = [AI2] … Threshold 1 = below or above Threshold 2 = below or above	**no sending** [other values in the range 30 s … 120 min]

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input AI2 Temperature value		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	16
Input AI2 outdoor temperature value		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	16
Input AI2 heat exchange coil battery sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	16
Input AI2 antistratification temperature sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	16
Input AI2 flow temperature sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	16
Input AI2 generic temperature sensor		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	16

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input AI2 Temperature threshold 1 – Switch	Threshold 1 ≠ not active	1 Bit	R-CT--	[1.1] DPT_Switch	17
Input AI2 Temperature threshold 2 – Switch	Threshold 2 ≠ not active	1 Bit	R-CT--	[1.1] DPT_Switch	18
Input AI2 Temperature threshold 1 – Value	Threshold 1 ≠ not active and Temperature threshold 1 enabled from bus = enable	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	19
Input AI2 Temperature threshold 2 - Value	Threshold 2 ≠ not active and Temperature threshold 2 enabled from bus = enable	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	20

11.4.3 BI1.. BI6 – Independent or single inputs: binary inputs

Independent or single inputs: send values or sequences

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Number of communication objects		from **1** to 8
Event		**activation / release** short / long press
Long press time	Event = short / long press	**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** as close or short press as open or long press
Behavior at unlocking	Function lock = enabled	**none** as close or short press as open or long press

Tab *Object(n)*

Parameter name	*Conditions*	*Values*
Send delay		**00:00:00.00** hh:mm:ss.ff [range from 00:00:00.00 to 01:49:13.50]
[…]		
Send ciclically	Number of communication objects = 1	**none** off / value 1 on / value 2 both off and on / both values

Parameter name	*Conditions*	*Values*
Cyclic sending interval	Number of communication objects = 1 and Send ciclically ≠ none	**00:00:02** hh:mm:ss [range 00:00:00 … 18:12:15]
[…]		
Communication object dimension		**1 bit value** 2 bit values 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned values 2 bytes signed values 2 bytes floating value
Reaction to short press	Communication object dimension = 1 bit value	none on off **toggle**
Reaction to long press	Communication object dimension = 1 bit value	none on off **toggle**
Reaction to short press	Communication object dimension = 2 bit values	**none** disabled enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
Reaction to long press	Communication object dimension = 2 bit values	**none** disabled enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
Reaction to short press	Communication object dimension ≠ 1 bit value and 2 bit values	**none** send value 1 send value 2 send value 1 ↔ send value 2
Reaction to long press	Communication object dimension ≠ 1 bit value and 2 bit values	**none** send value 1 send value 2 send value 1 ↔ send value 2
Value 1	Communication object dimension ≠ 1 bit value and 2 bit values	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 byte unsigned value) -32768... 32767 (2 byte signed value) -671088.64...670760.96 (2 bytes floating value)

Parameter name	*Conditions*	*Values*
Value 2	Communication object dimension ≠ 1 bit value and 2 bit values	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 byte unsigned value) -32768... 32767 (2 byte signed value) -671088.64...670760.96 (2 bytes floating value)

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BIx – Lock command	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	21, 35, 49, 63, 77, 91
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				
Input BIx – Switching status *1-bit*, object *n*	Communication object dimension = 1 bit value	1 Bit	-WCTU-	[1.1] DPT_Switch	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				
Input BIx – Switching status 2-bits, object *n*	Communication object dimension = 2 bits value	2 Bit	-WCTU-	[2.1] DPT_Switch_Control, [2.8] DPT_Direction1_Control	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				
Input BIx – Switching status 1-unsigned byte, object *n*	Communication object dimension = 1 byte unsigned value	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				
Input BIx – Switching status 1-unsigned byte, object *n*	Communication object dimension = 1 byte percentage	1 Byte	-WCTU-	[5.1] DPT_Scaling	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				
Input BIx – Switching status 1-signed byte, object *n*	Communication object dimension = 1 byte signed value	1 Byte	-WCTU-	[6.1] DPT_Percent_V8 [6.10] DPT_Value_1_Count	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				
Input BIx – Switching status 2-unsigned bytes, object *n*	Communication object dimension = 2 bytes unsigned value	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BIx – Switching status 2-signed bytes, object *n*	Communication object dimension = 2 bytes signed value	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				
Input BIx – Switching status 2-bytes floating value, object *n*	Communication object dimension = 2 bytes floating value	2 Bytes	-WCTU-	[9.x] DPT_Value_Temp	22.. 29 36.. 43 50.. 57 64.. 71 78.. 85 92.. 99
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP.				

Independent or single inputs: dimming

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Long press time		**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
Toggle mode		**disabled** / enabled
Short / long action	Toggle mode = disabled	off / darker on / brighter off / darker ↔ brighter on / darker ↔ brighter
Reaction to long press	Toggle mode = enabled	darker brighter darker ↔ brighter
Send ciclically		**none** off / value 1 on / value 2 both off and on / both values
Cyclic sending interval	Send ciclically ≠ none	**00:00:02** hh:mm:ss [range 00:00:00 … 18:12:15]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** off on toggle
Behavior at unlocking	Function lock = enabled	**none** off on as previous

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BIx – Lock command	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	21, 35, 49, 63, 77, 91
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				
Input BIx Dimming – Switching command		1 Bit	-WCTU-	[1.1] DPT_Switch	30, 44, 58, 72, 86, 92
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				
Input BIx – Dimming up / down / stop command		4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds	31, 45, 59, 73, 87, 93
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				

Independent or single inputs: shutter or venetian blind

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Long press time		**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
Toggle mode		**disabled** / enabled
Up / down action		**up** down
Venetian blind mode		**disabled** / enabled

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** up down
Behavior at unlocking	Function lock = enabled	**none** up down

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BIx – Lock command	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	21, 35, 49, 63, 77, 91
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				
Input BIx – Move up / down command	Venetian blind mode = enabled	1 Bit	--CT--	[1.8] DPT_UpDown	33, 47, 61, 75, 89, 103
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				
Input BIx – Dedicated stop command	Venetian blind mode = enabled	1 Bit	--CT--	[1.17] DPT_Trigger	30, 44, 58, 72, 86, 100
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				
Input BIx – Stop-step up / down command	Venetian blind mode = enabled	1 Bit	--CT--	[1.7] DPT_Step	32, 46, 60, 74, 88, 102
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				

Independent or single inputs: scenes

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[…]		
First scene number		**1** [range 1 … 64]
Learning mode		**disabled** /enabled
Scene activation	Learning mode = disabled	**send first scene only** toggle between 2 scenes
Second scene number	Learning mode = disabled	**2** [range 1 … 64]
Long press time	Learning mode = enabled	**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** send first scene send second scene
Behavior at unlocking	Function lock = enabled	**none** send first scene send second scene

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BIx – Lock command	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	21, 35, 49, 63, 77, 91
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				
Input BIx – Scene number		1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	34, 48, 62, 76, 90, 104
	[BIx] x from 1 to 6. BI6 is available only for EK-HO1-TP				

11.4.4 BI1/2, BI3/4, BI5/6 – Coupled Inputs

Coupled inputs: switching

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Long press time		**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
Use		**BI(x) on, BI(x+1) off** BI(x) off, BI(x+1) on
Send ciclically		**none** off / value 1 on / value 2 both off and on / both values
Cyclic sending interval	Send ciclically ≠ none	**00:00:02** hh:mm:ss [range 00:00:00 … 18:12:15]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** off on toggle
Behavior at unlocking	Function lock = enabled	**none** off on as previous

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Inputs BI(x) and BI(x+1) – Lock command	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	21, 49, 77
	[BIx] x = 1,3,5				

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Inputs BI(x) and BI(x+1) – Switching command		1 Bit	-WCTU-	[1.1] DPT_Switch	30, 58, 86
	[BIx] x = 1,3,5				

Coupled inputs: dimming

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Long press time		**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
Use		**BI(x) increases, BI(x+1) decreases** BI(x) decreases, BI(x+1) increases
Send ciclically		**none** off / value 1 on / value 2 both off and on / both values
Cyclic sending interval	Send ciclically ≠ none	**00:00:02** hh:mm:ss [range 00:00:00 … 18:12:15]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** off on as previous
Behavior at unlocking	Function lock = enabled	**none** off on toggle

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Inputs BI(x) and BI(x+1) – Lock command	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	21, 49, 77
	[BIx] x = 1,3,5				
Inputs BI(x) and BI(x+1) Dimming – Switching command	Type = dimming	1 Bit	-WCTU-	[1.1] DPT_Switch	30, 58, 86
	[BIx] x = 1,3,5				

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Inputs BI(x) and BI(x+1) – Dimming up / down / stop command	Type = dimming	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds	31, 59, 87
	[BIx] x = 1,3,5				

Coupled inputs: shutter or venetian blind

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Long press time		**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
Use		**BI(x) up, BI(x+1) down** BI(x) down, BI(x+1) up
Venetian blind mode		**disabled** / enabled

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** up down
Behavior at unlocking	Function lock = enabled	**none** up down

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Inputs BI(x) and BI(x+1) – Lock command	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	21, 49, 77
	[BIx] x = 1,3,5				
Inputs BI(x) and BI(x+1) – Move up / down command		1 Bit	--CT--	[1.8] DPT_UpDown	33, 61, 89
	[BIx] x = 1,3,5				
Inputs BI(x) and BI(x+1) – Dedicated stop command	Venetian blind mode = disabled	1 Bit	--CT--	[1.17] DPT_Trigger	30, 58, 86
	[BIx] x = 1,3,5				
Inputs BI(x) and BI(x+1) – Stop-step up / down command	Venetian blind mode = enabled	1 Bit	--CT--	[1.7] DPT_Step	32, 60, 88
	[BIx] x = 1,3,5				

11.4.5 BI7 – Internal badge Input

Send values or sequences

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Number of communication objects		from **1** to 8
Event		**activation / release** short / long press
Long press time	Event = short / long press	**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** as close or short press as open or long press
Behavior at unlocking	Function lock = enabled	**none** as close or short press as open or long press

Tab *Object(n)*

Parameter name	*Conditions*	*Values*
Send delay		**00:00:00.00** hh:mm:ss.ff [range from 00:00:00.00 to 01:49:13.50]
[…]		
Send ciclically	Number of communication objects = 1	**none** off / value 1 on / value 2 both off and on / both values

Parameter name	*Conditions*	*Values*
Cyclic sending interval	Number of communication objects = 1 and Send ciclically ≠ none	**00:00:02** hh:mm:ss [range 00:00:00 … 18:12:15]
[…]		
Communication object dimension		**1 bit value** 2 bit values 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned values 2 bytes signed values 2 bytes floating value
Reaction to short press	Communication object dimension = 1 bit value	none on off **toggle**
Reaction to long press	Communication object dimension = 1 bit value	none on off **toggle**
Reaction to short press	Communication object dimension = 2 bit values	**none** disabled enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
Reaction to long press	Communication object dimension = 2 bit values	**none** disabled enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
Reaction to short press	Communication object dimension ≠ 1 bit value and 2 bit values	**none** send value 1 send value 2 send value 1 ↔ send value 2
Reaction to long press	Communication object dimension ≠ 1 bit value and 2 bit values	**none** send value 1 send value 2 send value 1 ↔ send value 2
Value 1	Communication object dimension ≠ 1 bit value and 2 bit values	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 byte unsigned value) -32768... 32767 (2 byte signed value) -671088.64...670760.96 (2 bytes floating value)

Parameter name	*Conditions*	*Values*
Value 2	Communication object dimension ≠ 1 bit value and 2 bit values	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 byte unsigned value) -32768... 32767 (2 byte signed value) -671088.64...670760.96 (2 bytes floating value)

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BI7 – Lock command (* only EK-HO1-TP device)	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	105
Input BI7 – Switching status *1-bit*, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 bit value	1 Bit	-WCTU-	[1.1] DPT_Switch	106.. 113
Input BI7 – Switching status 2-bits, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bits value	2 Bit	-WCTU-	[2.1] DPT_Switch_Control, [2.8] DPT_Direction1_Control	106.. 113
Input BI7 – Switching status 1-unsigned byte, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 byte unsigned value	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount	106.. 113
Input BI7 – Switching status 1-unsigned byte, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 byte percentage	1 Byte	-WCTU-	[5.1] DPT_Scaling	106.. 113
Input BI7 – Switching status 1-signed byte, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 byte signed value	1 Byte	-WCTU-	[6.1] DPT_Percent_V8 [6.10] DPT_Value_1_Count	106.. 113
Input BI7 – Switching status 2-unsigned bytes, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bytes unsigned value	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount	106.. 113
Input BI7 – Switching status 2-signed bytes, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bytes signed value	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count	106.. 113
Input BI7 – Switching status 2-bytes floating value, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bytes floating value	2 Bytes	-WCTU-	[9.x] DPT_Value_Temp	106.. 113

Scenes function

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[…]		
First scene number		**1** [range 1 … 64]
Learning mode		**disabled** /enabled
Scene activation	Learning mode = disabled	**send first scene only** toggle between 2 scenes
Second scene number	Learning mode = disabled	**2** [range 1 … 64]
Long press time	Learning mode = enabled	**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** send first scene send second scene
Behavior at unlocking	Function lock = enabled	**none** send first scene send second scene

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BI7 – Lock command (* only EK-HO1-TP device)	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	105
Input BI7 – Scene number (* only EK-HO1-TP device)		1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	114

11.4.6 BI8 – External transponder Input

Send values or sequences

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[,,,]		
Number of communication objects		from **1** to 8
Event		**activation / release** short / long press
Long press time	Event = short / long press	**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** as close or short press as open or long press
Behavior at unlocking	Function lock = enabled	**none** as close or short press as open or long press

Tab *Object(n)*

Parameter name	*Conditions*	*Values*
Send delay		**00:00:00.00** hh:mm:ss.ff [range from 00:00:00.00 to 01:49:13.50]
[…]		
Send ciclically	Number of communication objects = 1	**none** off / value 1 on / value 2 both off and on / both values

Parameter name	*Conditions*	*Values*
Cyclic sending interval	Number of communication objects = 1 and Send ciclically ≠ none	**00:00:02** hh:mm:ss [range 00:00:00 … 18:12:15]
[…]		
Communication object dimension		**1 bit value** 2 bit values 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned values 2 bytes signed values 2 bytes floating value
Reaction to short press	Communication object dimension = 1 bit value	none on off **toggle**
Reaction to long press	Communication object dimension = 1 bit value	none on off **toggle**
Reaction to short press	Communication object dimension = 2 bit values	**none** disabled enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
Reaction to long press	Communication object dimension = 2 bit values	**none** disabled enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
Reaction to short press	Communication object dimension ≠ 1 bit value and 2 bit values	**none** send value 1 send value 2 send value 1 ↔ send value 2
Reaction to long press	Communication object dimension ≠ 1 bit value and 2 bit values	**none** send value 1 send value 2 send value 1 ↔ send value 2
Value 1	Communication object dimension ≠ 1 bit value and 2 bit values	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 byte unsigned value) -32768... 32767 (2 byte signed value) -671088.64...670760.96 (2 bytes floating value)

Parameter name	*Conditions*	*Values*
Value 2	Communication object dimension ≠ 1 bit value and 2 bit values	0...255 (1 byte unsigned value) 0…100 (1 byte percentage) -128...127 (1 byte signed value) 0...65535 (2 byte unsigned value) -32768... 32767 (2 byte signed value) -671088.64...670760.96 (2 bytes floating value)

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BI8 – Lock command (* only EK-HO1-TP device)	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	115
Input BI8 – Switching status *1-bit*, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 bit value	1 Bit	-WCTU-	[1.1] DPT_Switch	116.. 123
Input BI8 – Switching status 2-bits, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bits value	2 Bit	-WCTU-	[2.1] DPT_Switch_Control, [2.8] DPT_Direction1_Control	116.. 123
Input BI8 – Switching status 1-unsigned byte, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 byte unsigned value	1 Byte	-WCTU-	[5.10] DPT_Value_1_Ucount	116.. 123
Input BI8 – Switching status 1-unsigned byte, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 byte percentage	1 Byte	-WCTU-	[5.1] DPT_Scaling	116.. 123
Input BI8 – Switching status 1-signed byte, object *n* (* only EK-HO1-TP device)	Communication object dimension = 1 byte signed value	1 Byte	-WCTU-	[6.1] DPT_Percent_V8 [6.10] DPT_Value_1_Count	116.. 123
Input BI8 – Switching status 2-unsigned bytes, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bytes unsigned value	2 Bytes	-WCTU-	[7.1] DPT_Value_2_Ucount	116.. 123
Input BI8 – Switching status 2-signed bytes, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bytes signed value	2 Bytes	-WCTU-	[8.1] DPT_Value_2_Count	116.. 123
Input BI8 – Switching status 2-bytes floating value, object *n* (* only EK-HO1-TP device)	Communication object dimension = 2 bytes floating value	2 Bytes	-WCTU-	[9.x] DPT_Value_Temp	116.. 123

Scenes function

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]
[…]		
First scene number		**1** [range 1 … 64]
Learning mode		**disabled** /enabled
Scene activation	Learning mode = disabled	**send first scene only** toggle between 2 scenes
Second scene number	Learning mode = disabled	**2** [range 1 … 64]
Long press time	Learning mode = enabled	**00:00:00.800** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** send first scene send second scene
Behavior at unlocking	Function lock = enabled	**none** send first scene send second scene

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BI8 – Lock command (* only EK-HO1-TP device)	Function lock = enabled	1 Bit	-WC---	[1.3] DPT_Enable	115
Input BI8 – Scene number (* only EK-HO1-TP device)		1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	124

11.4.7 BI9.. BI11 – Generic inputs

Parameter name	*Conditions*	*Values*
Contact type		**NO (normally open)** NC (normally closed)
Lock function		**disabled** /enabled
Debounce time		**00:00:00.050** hh:mm:ss.fff [range from 00:00:00.000 to 00:10:55.350]

Tab *Locking function*

Parameter name	*Conditions*	*Values*
Invert lock device signal	Function lock = enabled	**not inverted** inverted
Lock after bus recovery	Function lock = enabled	**no** / yes
Behavior at locking	Function lock = enabled	**none** as close as open
Behavior at unlocking	Function lock = enabled	**none** as close as open

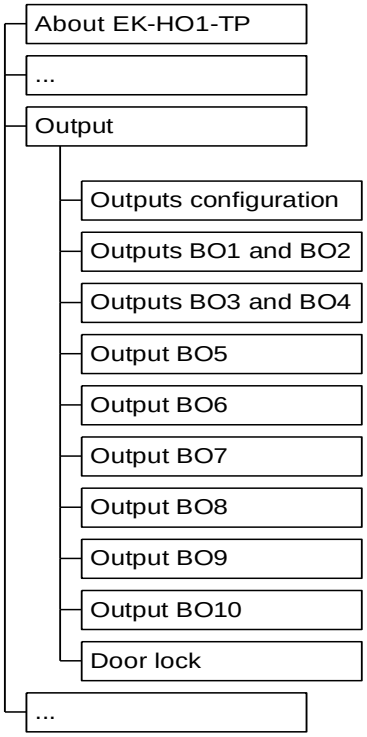
Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BI9 – Lock command (* only EK-HO1-TP device)	Lock function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	125
Input BI9 – Windows contact status (* only EK-HO1-TP device)		1 Bit	R-CT--	[1.19] DPT_Window_Door	126
Input BI9 – Generic contact status (* only EK-HO1-TP device)		1 Bit	R-CT--	[1.1] DPT_Switch	126
Input BI10 – Lock command (* only EK-HO1-TP device)	Lock function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	127
Input BI10 – Help request status (* only EK-HO1-TP device)		1 Bit	R-CT--	[1.1] DPT_Switch	128
Input BI11 – Lock command (* only EK-HO1-TP device)	Lock function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	129
Input BI11 – Room cleaning request status (* only EK-HO1-TP device)		1 Bit	R-CT--	[1.1] DPT_Switch	130

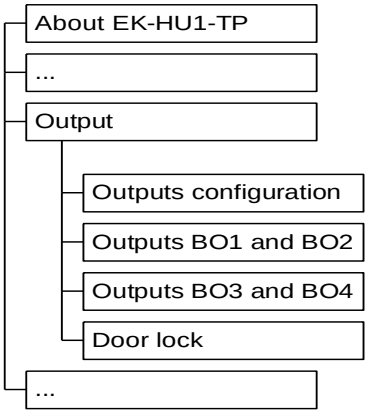
11.5 Outputs

The following parameters are set in the *Outputs* tab:

- Enable outputs
- The outputs BO1-BO2 and BO3-BO4 can be individually configured as single binary outputs or in coupled mode for the control of shutters, blinds or dimming devices
- The remaining outputs (only for EK-HO1-TP controller) can be configured individually as binary outputs

In the board it is also possible to enable the output dedicated to the electric strike, available on both the EK-HO1-TP and EK-HU1-TP controllers.





11.5.1 Outputs setting

Parameter name	*Conditions*	*Values*
Outputs BO1 and BO2		**disabled** 2 binary outputs venetian blind / shutter
Use	Outputs BO1 and BO2 = venetian blind / shutter	**shutter** venetian blind
Output BO1	Outputs BO1 and BO2 = 2 binary outputs	**disabled** / enabled
Output BO2	Outputs BO1 and BO2 = 2 binary outputs	**disabled** / enabled
[…]		
Outputs BO3 and BO4		**disabled** 2 binary outputs venetian blind / shutter
Output BO3	Outputs BO3 and BO4 = 2 binary outputs	**disabled** / enabled
Output BO4	Outputs BO3 and BO4 = 2 binary outputs	**disabled** / enabled
Use	Outputs BO3 and BO4 = venetian blind / shutter	**shutter** venetian blind
[…]		
Output BO5 (*)		**disabled** / enabled
	(only for EK-HO1-TP)*	
Output BO6 (*)		**disabled** / enabled
	(only for EK-HO1-TP)*	
Output BO7 (*)		**disabled** / enabled
	(only for EK-HO1-TP)*	
Output BO8 (*)		**disabled** / enabled
	(only for EK-HO1-TP)*	
Output BO9 (*)		**disabled** / enabled
	(only for EK-HO1-TP)*	
Output BO10 (*)		**disabled** / enabled
	(only for EK-HO1-TP)*	
[…]		
Door lock		**disabled** / enabled

11.5.2 BO1/2 – BO3/4 – Single or coupled outputs

Parameter name	*Conditions*	*Values*
Reversion pause time		**300** ms [other values in the range 50 ms ... 65535 ms]
Move up time		**00:00:15** hh:mm:ss [range 00:00:00 … 18:12:15]
Move down time		**00:00:15** hh:mm:ss [range 00:00:00 … 18:12:15]
Position control with dimmer		**no** / yes
Slat movement time	Use = venetian blinds	**00:00:15** hh:mm:ss [range 00:00:00 … 18:12:15]
Slat step time	Use = venetian blinds	**100** ms 0…65535 [milliseconds]
Slats control with dimmer	Use = venetian blinds	**no** / yes
[…]		
Behavior at device power on		**none** up / down down / close stop move to position
Device power on position	Behavior at device power on = move to position	**0**.. 100 %
Device power on slats position	Use = venetian blinds and Behavior at device power on = move to position	**0**.. 100 %
Behavior at bus on		**none** up / down down / close stop move to position
Bus on position	Behavior at bus on = move to position	**0**.. 100 %
Bus on slats position	Use = venetian blinds and Behavior at bus on = move to position	**0**.. 100 %
Behavior after download		**none** up / down down / close stop move to position

Parameter name	*Conditions*	*Values*
After download position	Behavior after download = move to position	**0**.. 100 %
After download slats position	Use = venetian blinds and Behavior after download = move to position	**0**.. 100 %
[…]		
Locking function		**disabled /**enabled
Forcing function		**disabled /**enabled
Behavior end forced control	Forcing function = enabled	**none** up / down down / close stop move to position previous
End forced position	Forcing function = enabled and Behavior end forced control = move to position	**0**.. 100 %
End forced slat position	Use = venetian blinds and Forcing function = enabled and Behavior end forced control = move to position	**0**.. 100 %
Behavior after bus recovery	Funzionamento forzato = abilitato	**none** up / down down / close stop move to position previous
Bus recovery forced position	Forcing function = enabled and Behavior after bus recovery = move to position	**0**.. 100 %
Bus recovery forced slat position	Use = venetian blinds and Forcing function = enabled and Behavior after bus recovery = move to position	**0**.. 100 %
[…]		
Scenes function		**disabled /**enabled

Locking function

Parameter name	***Conditions***	***Values***
Lock device signal		**not inverted** / inverted
After bus recovery		unlock lock **previous state**
Behavior at locking		**none** up / down down / close stop move to position
Lock start position	Behavior at locking = move to position	**0**.. 100 %
Lock start slats position	Use = venetian blinds and Behavior at locking = move to position	**0**.. 100 %
Behavior at unlocking		**none** up / down down / close stop move to position previous
Lock end position	Behavior at unlocking = move to position	**0**.. 100 %
Lock end slat position	Use = venetian blinds and Behavior at unlocking = move to position	**0**.. 100 %

Scenes function

Parameter name	***Conditions***	***Values***
Download overwrites learned behavior		**yes** / no
[…]		
Scene x (x from 1 to 8)		**disabled** /enabled
Scene number	Scene x = enabled	**1** [range from 1 to 64]
Output behavior	Scene x = enabled	stop **fully opened** fully closed move to position

Parameter name	*Conditions*	*Values*
Scene position	Scene x = enabled Output behavior = move to position	**0**.. 100 %
Scene slats position	Use = venetian blinds and Scene x = enabled Output behavior = move to position	**0**.. 100 %
Activation delay	Scene x = enabled	**00:00:00.00** hh:mm:ss.ff [range from 00:00:00.00 to 01:49:13.50]
Learning mode	Scene x = enabled	**disabled** /enabled

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Output BO1 – On/off command	Output BO1 and BO2 = 2 binary inputs	1 Bit	-WC---	[1.1] DPT_Switch	192
Output BO1 – On/off status	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	193
Output BO1 – Lock command	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	194
Output BO1 – Forcing command	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	195
Output BO1 – Scene number	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	144
Output BO2 – On/off command	Output BO1 and BO2 = 2 binary inputs	1 Bit	-WC---	[1.1] DPT_Switch	145
Output BO2 – On/off status	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	146
Output BO2 – Lock command	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	147
Output BO2 – Forcing command	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	148
Output BO2 – Scene number	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	149
Output BO3 – On/off command	Output BO3 and BO4 = 2 binary inputs	1 Bit	-WC---	[1.1] DPT_Switch	168
Output BO3 – On/off status	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	169
Output BO3 – Lock command	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	170
Output BO3 – Forcing command	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	171

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Output BO3 – Scene number	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	172
Output BO4 – On/off command	Output BO3 and BO4 = 2 binary inputs	1 Bit	-WC---	[1.1] DPT_Switch	173
Output BO4 – On/off status	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	174
Output BO4 – Lock command	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	175
Output BO4 – Forcing command	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	176
Output BO4 – Scene number	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	177
[…]					
Outputs BO1 and BO2 – Lock command	Lock function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	159
Outputs BO1 and BO2 – Forcing command	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	160
Outputs BO1 and BO2 – Scene number	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	161
Outputs BO1 and BO2 – Move up-down command	Use = shutter Use = venetian blind	1 Bit	-WC---	[1.8] DPT_UpDown [1.9] DPT_OpenClose	154
Outputs BO1 and BO2 – Stop-step up-down command	Use = shutter Use = venetian blind	1 Bit	-WC---	[1.7] DPT_Step	155
Outputs BO1 and BO2 – Dedicated stop command	Use = shutter Use = venetian blind	1 Bit	-WC---	[1.17] DPT_Trigger	156
Outputs BO1 and BO2 – Info move	Use = shutter Use = venetian blind	1 Bit	R-CT--	[1.8] DPT_UpDown	157
Outputs BO1 and BO2 – Valid current abs position	Use = shutter Use = venetian blind	1 Bit	R-CT--	[1.2] DPT_Bool	158
Outputs BO1 and BO2 – Abs position command	Use = shutter Use = venetian blind	1 Byte	-WC---	[5.1] DPT_Scaling	163
Outputs BO1 and BO2 – Abs position status	Use = shutter Use = venetian blind	1 Byte	R-CT--	[5.1] DPT_Scaling	164
Outputs BO1 and BO2 – Dimmer position command	Use = shutter Position control with dimmer = yes	4 Bit	-WC---	[3.8] DPT_Control_Blinds	162
Outputs BO1 and BO2 – Dimmer slats command	Use = venetian blind Slats control with dimmer	4 Bit	-WC---	[3.8] DPT_Control_Blinds	165

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Outputs BO1 and BO2 – Abs slat position command	Use = venetian blind	1 Byte	-WC---	[5.1] DPT_Scaling	166
Outputs BO1 and BO2 – Abs slat position status	Use = venetian blind	1 Byte	R-CT--	[5.1] DPT_Scaling	167
[…]					
Outputs BO3 and BO4 – Lock command	Lock function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	183
Outputs BO3 and BO4 – Forcing command	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	184
Outputs BO3 and BO4 – Scene number	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	185
Outputs BO3 and BO4 – Move up-down command	Use = shutter Use = venetian blind	1 Bit	-WC---	[1.8] DPT_UpDown [1.9] DPT_OpenClose	178
Outputs BO3 and BO4 – Stop-step up-down command	Use = shutter Use = venetian blind	1 Bit	-WC---	[1.7] DPT_Step	179
Outputs BO3 and BO4 – Dedicated stop command	Use = shutter Use = venetian blind	1 Bit	-WC---	[1.17] DPT_Trigger	180
Outputs BO3 and BO4 – Info move	Use = shutter Use = venetian blind	1 Bit	R-CT--	[1.8] DPT_UpDown	181
Outputs BO3 and BO4 – Valid current abs position	Use = shutter Use = venetian blind	1 Bit	R-CT--	[1.2] DPT_Bool	182
Outputs BO3 and BO4 – Abs position command	Use = shutter Use = venetian blind	1 Byte	-WC---	[5.1] DPT_Scaling	187
Outputs BO3 and BO4 – Abs position status	Use = shutter Use = venetian blind	1 Byte	R-CT--	[5.1] DPT_Scaling	188
Outputs BO3 and BO4 – Dimmer position command	Use = shutter Position control with dimmer = yes	4 Bit	-WC---	[3.8] DPT_Control_Blinds	186
Outputs BO3 and BO4 – Dimmer slats command	Use = venetian blind Slats control with dimmer	4 Bit	-WC---	[3.8] DPT_Control_Blinds	189
Outputs BO3 and BO4 – Abs slat position command	Use = venetian blind	1 Byte	-WC---	[5.1] DPT_Scaling	190
Outputs BO3 and BO4 – Abs slat position status	Use = venetian blind	1 Byte	R-CT--	[5.1] DPT_Scaling	191

11.5.3 BO5.. BO10 – Single Outputs

The outputs BO5 to BO10, which can only be configured as single outputs, are available for the EK-HO1-TP controller.

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Output BO5 – On/off command (* only EK-HO1-TP device)		1 Bit	-WC---	[1.1] DPT_Switch	192
Output BO5 – On/off status (* only EK-HO1-TP device)	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	193
Output BO5 – Lock command (* only EK-HO1-TP device)	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	194
Output BO5 – Forcing command (* only EK-HO1-TP device)	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	195
Output BO5 – Scene number (* only EK-HO1-TP device)	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	196
Output BO6 – On/off command (* only EK-HO1-TP device)		1 Bit	-WC---	[1.1] DPT_Switch	197
Output BO6 – On/off status (* only EK-HO1-TP device)	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	198
Output BO6 – Lock command (* only EK-HO1-TP device)	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	199
Output BO6 – Forcing command (* only EK-HO1-TP device)	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	200
Output BO6 – Scene number (* only EK-HO1-TP device)	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	201
Output BO7 – On/off command (* only EK-HO1-TP device)		1 Bit	-WC---	[1.1] DPT_Switch	202
Output BO7 – On/off status (* only EK-HO1-TP device)	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	203
Output BO7 – Lock command (* only EK-HO1-TP device)	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	204
Output BO7 – Forcing command (* only EK-HO1-TP device)	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	205

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Output BO7 – Scene number (* only EK-HO1-TP device)	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	206
Output BO8 – On/off command (* only EK-HO1-TP device)		1 Bit	-WC---	[1.1] DPT_Switch	207
Output BO8 – On/off status (* only EK-HO1-TP device)	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	208
Output BO8 – Lock command (* only EK-HO1-TP device)	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	209
Output BO8 – Forcing command (* only EK-HO1-TP device)	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	210
Output BO8 – Scene number (* only EK-HO1-TP device)	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	211
Output BO9 – On/off command (* only EK-HO1-TP device)		1 Bit	-WC---	[1.1] DPT_Switch	212
Output BO9 – On/off status (* only EK-HO1-TP device)	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	213
Output BO9 – Lock command (* only EK-HO1-TP device)	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	214
Output BO9 – Forcing command (* only EK-HO1-TP device)	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	215
Output BO9 – Scene number (* only EK-HO1-TP device)	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	216
Output BO10 – On/off command (* only EK-HO1-TP device)		1 Bit	-WC---	[1.1] DPT_Switch	217
Output BO10 – On/off status (* only EK-HO1-TP device)	Status feedback telegram = enabled	1 Bit	R-CT--	[1.1] DPT_Switch	218
Output BO10 – Lock command (* only EK-HO1-TP device)	Locking function = enabled	1 Bit	-WC---	[1.3] DPT_Enable	219
Output BO10 – Forcing command (* only EK-HO1-TP device)	Forcing function = enabled	2 Bit	-WC---	[2.8] DPT_Direction1_Control	220

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Output BO10 – Scene number (* only EK-HO1-TP device)	Scenes function = enabled	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	221

11.5.4 Door lock

Parameter name	*Conditions*	*Values*
Pulse length		**00:00:00.600** hh:mm:ss.fff [from 00:00:00.000 to 00:10:55.350]

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Door Lock – Pulse command		1 Bit	-WC---	[1.1] DPT_Switch	222
Door Lock – Pulse status		1 Bit	R-CT--	[1.1] DPT_Switch	223

11.6 External sensors (from bus)

The card allows configuring:

- The temperature probes from the bus required to implement the temperature control functions
- Communication objects to realize energy saving functions (window contacts and presence sensors from the bus)
- Communication objects concerning the internal pocket and the transponder outside the camera.

Parameter name	*Conditions*	*Values*
Room temperature 1		**disabled** / enabled
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Room temperature 2		**disabled** / enabled
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Outdoor temperature		**disabled** / enabled
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Coil temperature		**disabled** / enabled
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Antistratification temperature		**disabled** / enabled
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Flow temperature		**disabled** / enabled
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]

Parameter name	*Conditions*	*Values*
Analog sensors timeout (0 means timeout)		**00:05:00** hh:mm:ss [range 00:00:00 … 18:12:15]
	The field has the format hh: mm: ss (hours: minutes: seconds): the default value of 00:05:00 therefore corresponds to a timeout of 5 minutes. The value 00:00:00 means that the timeout of the analog sensors is deactivated.	
[…]		
Window contact 1		**disabled** / enabled
Signal	enabled	**not inverted** / inverted
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Window contact 2		**disabled** / enabled
Signal	enabled	**not inverted** / inverted
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
External badge		**disabled** / enabled
Signal	enabled	**not inverted** / inverted
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Internal badge		**disabled** / enabled
Signal	enabled	**not inverted** / inverted
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Presence sensor 1		**disabled** / enabled
Signal	enabled	**not inverted** / inverted
Read on init	enabled	**no** / yes

Parameter name	*Conditions*	*Values*
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Presence sensor 2		**disabled** / enabled
Signal	enabled	**not inverted** / inverted
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Drip tray level		**disabled** / enabled
Signal	enabled	**not inverted** / inverted
Read on init	enabled	**no** / yes
Cyclic reading interval	enabled	**no reading** [other values in the range 30 s … 120 min]
Digital sensors timeout (0 means no timeout)		**00:05:00 hh:mm:ss** [range 00:00:00 … 18:12:15]
	The field has the format hh: mm: ss (hours: minutes: seconds): the default value of 00:05:00 therefore corresponds to a timeout of 5 minutes. The value 00:00:00 means that the digital sensors timeout is deactivated.	

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Room temperature 1 (from bus)		2 Bytes	-WC---	[9.1] DPT_Value_Temp	131
Room temperature 2 (from bus)		2 Bytes	-WC---	[9.1] DPT_Value_Temp	132
Outdoor temperature (from bus)		2 Bytes	-WC---	[9.1] DPT_Value_Temp	133
Coil battery temperature (from bus)		2 Bytes	-WC---	[9.1] DPT_Value_Temp	134
Antistratification temperature (from bus)		2 Bytes	-WC---	[9.1] DPT_Value_Temp	135
Flow temperature (from bus)		2 Bytes	-WC---	[9.1] DPT_Value_Temp	136
Drip tray contact (from bus)		1 Bit	-WC---	[1.5] DPT_Alarm	137
Windows contact 1 (from bus)		1 Bit	-WC---	[1.1] DPT_Window_Door	138
Windows contact 2 (from bus)		1 Bit	-WC---	[1.1] DPT_Window_Door	139
Presence sensor 1 (from bus)		1 Bit	-WC---	[1.18] DPT_Occupancy	140
Presence sensor 2 (from bus)		1 Bit	-WC---	[1.18] DPT_Occupancy	141

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
External badge (from bus)		1 Bit	-WC---	[1.1] DPT_Switch	142
Internal badge (from bus)		1 Bit	-WC---	[1.1] DPT_Switch	143

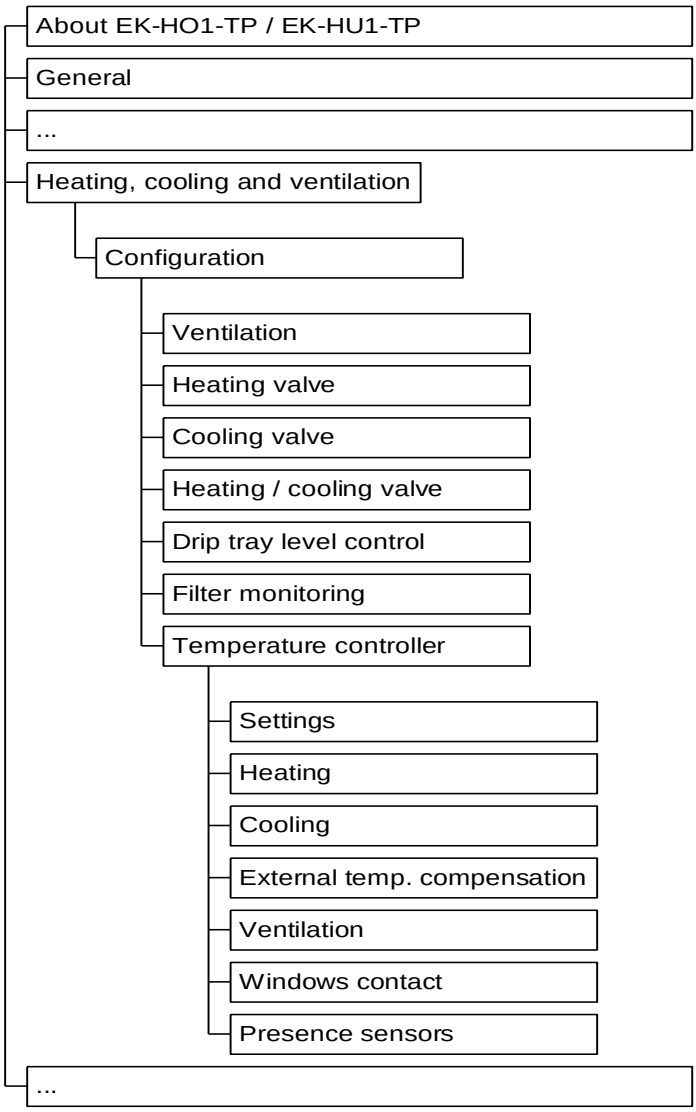
11.7 HVAC – Heating, cooling and ventilation

The controllers are designed to offer a complete thermoregulation solution to an area for a hotel room or for offices.

- Use of HO1 / HU1 device as actuator in combination of a room thermostat via KNX bus: the thermostat can be ekinex or a third-party product
- Use of the HO1 / HU1 device with the internal controller and with an external interface consisting of a room thermostat via bus (ekinex or third-party). The thermostat sends the value of the room temperature sensor to the HO1 / HU1 device; alternatively, the room temperature is acquired directly by the HO1 / HU1 device via AI1 and / or AI2 and can be visualized or not on the room thermostat. The room thermostat allows to modify the Setpoint and this is sent to the HO1 / HU1 device as a single Setpoint.
- Use of the HO1 / HU1 device with the internal controller complete with the 8 Setpoints, controlled by a supervision system.

It is possible to keep the operating modes that, in a hotel application, have another meaning:

- BUILDING PROTECTION: room not booked
- ECONOMY mode: preset camera
- STAND-BY mode: room reserved and not occupied
- COMFORT mode: room booked and occupied



11.7.1 Settings

This folder contains the general settings for the device configuration:

- Application type: ventilation, convector or fan-coil
- Function: heating, cooling and both heating and cooling
- Installation type: 2-pipe or 4-pipe
- Device use: as actuator with external controller or as actuator/controller with internal controller.

Parameter name	*Conditions*	*Values*
Application		ventilation convector **fan-coil**

Parameter name	*Conditions*	*Values*
Function		**heating** cooling both heating and cooling
Installation Type	Function = both heating and cooling, Application = convector, fan-coil	**2 pipes** 4 pipes
Controller		**external** internal
Communication object	Controller = external, Application = fan-coil	unique **separated**
Ventilation object format	Controller = external, Application = ventilation, Application = fan-coil	output status [DPT 1.001] counter [DPT 5.010] **percentage [DPT 5.001]**
Valve object format	Controller = external, Application = convector, Application = fan-coil and Communication object = separated	output status [DPT 1.001]
	In the current version of the devices, only this option is available.	
Communication object timeout	Controller = external	**00:05:00** hh:mm:ss [range 00:00:00 … 18:12:15]
	This parameter allows disabling the actuator outputs and generating a communication alarm if the object or command objects are not updated within the timeout set. The field has the format hh:mm:ss (hours:minutes:seconds). The default 0:05:00 therefore corresponds to a 5 minute timeout. The 00:00:00 means that the update control for command objects is disabled.	
Manual operation		**disabled** enabled
Disable from bus	Manual operation = enabled	**no**/yes
Restore auto mode time	Manual operation = enabled	**00:15:00** hh:mm:ss [range 00:00:00 … 18:12:15]
Power status feedback		**disabled** enabled
Report filter change		**no**/yes
Delay after bus voltage recovery		**00:00:05** hh:mm:ss [range 00:00:00 … 18:12:15]
	Time interval after which the transmission of the telegrams on the bus starts after the power supply is restored. The delay affects both the event-driven transmission and the cyclic transmission of a telegram. Regarding the latter, the counting of the pause interval for retransmission starts at the end of the time of initial delay. *The field has format hh:mm:ss:fff (hours : minutes : seconds .milliseconds): the default value 00:00:05.000 corresponds to 5 seconds.*	

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Thermal generator lock alarm		1 Bit	-WC---	[1.5] DPT_Alarm	6
	This C.O. is used from an external device to stop operation both as actuator and as regulator. If the controller = external, when the alarm is received the outputs and ventilation stop. If controller = internal, also the internal temperature controller is stopped.				
Temperature control alarm		1 Bit	R-CT--	[1.5] DPT_Alarm	7
	The internal temperature controller alarm is active in one of the following conditions: • *Failure of one of the temperature sensors used for control* • *Timeout reception of a temperature sensor used for control* • *Alarm reception from C.O. Thermal generator locked* • *Receiving a full condensate pan state (if configured ≠ only reporting).*				
Filter change warning	Report filter change = yes	1 Bit	R-CT--	[1.5] DPT_Alarm	8
Output V1 from bus	Application = convector, Ventilation ⇒ Control type = 3 speeds (0…10V) (*) o Ventilation ⇒ Control type = continuous regulation (0…10V)	1 Bit	-WC---	[1.1] DPT_Switch	268
	The output is not used for the ventilation command; the C.O. allows to control the relay output and dedicate it to other use (eg. command dedicated to lighting or the ON/OFF drive).				
Output V2 from bus	Application = convector, Ventilation ⇒ Control type = 1 speed (relay) o Ventilation ⇒ Control type = 3 speeds (0…10V) o Ventilation ⇒ Control type = continuous regulation (0…10V)	1 Bit	-WC---	[1.1] DPT_Switch	269
	The output is not used for the ventilation command; the C.O. allows to control the relay output and dedicate it to other use (eg. command dedicated to lighting or the ON/OFF drive).				
Output V3 from bus	Application = convector, Ventilation ⇒ Control type = 2 speeds (relays) or Ventilation ⇒ Control type = 3 speeds (0…10V) or Ventilation ⇒ Control type = continuous regulation (0…10V)	1 Bit	-WC---	[1.1] DPT_Switch	270
	The output is not used for the ventilation command; the C.O. allows to control the relay output and dedicate it to other use (eg. command dedicated to lighting or the ON/OFF drive).				
Output heating valve from bus	Application = ventilation	1 Bit	-WC---	[1.1] DPT_Switch	271
	The output is not used for the ventilation command; the C.O. allows to control the relay output and dedicate it to other use (eg. command dedicated to lighting or the ON/OFF drive).				
Output cooling valve from bus	Application = ventilation, Application = convector, fan-coil and Installation type = 2 pipes	1 Bit	-WC---	[1.1] DPT_Switch	272
	The output is not used for the ventilation command; the C.O. allows to control the relay output and dedicate it to other use (eg. command dedicated to lighting or the ON/OFF drive).				
Output 0-10V from bus		1 Byte	-WC---	[5.1] DPT_Scaling	273
	The output is not used for the ventilation command; the C.O. allows to control the 0-10V output and dedicate it to other use.				

11.7.2 External controller

Ventilation

The ventilation folder contains:

- the setting parameters for relays or continuous command outputs (for EK-HB1-TP and EK-HC1-TP versions) and the parameters for the output behaviour based on the values of the commands received from bus;
- the default values for the outputs when the timeout after receiving a command is reached;
- the fan activation and deactivation delays;
- the speed limitation activation (e.g. at night);

The ventilation folder is active if: Application = ventilation or fan-coil and Controller = external.

If Controller = internal, the folder is not enabled and an equivalent folder is activated inside the Temperature control folder.

Parameter name	***Conditions***	***Values***
Control type	General ⇒ Communication object = output status or counter or General ⇒ Ventilation object format = output status or counter	**1 speed (relay)** 2 speeds (relays) 3 speeds (relays) 3 speeds (0…10V)
	The continuous control option is not active in this condition because the command signal has only discrete values.	
Control type	General ⇒ Communication object = percentage or General ⇒ Ventilation object format = percentage	**1 speed (relay)** 2 speeds (relays) 3 speeds (relays) 3 speeds (0…10V) continuous regulation (0… 10V)
[…]		
	Setting parameters depending on the adopted configuration. Please see the different situation in the following sections.	
Speed at timeout	General ⇒ Communication object = output status or counter or General ⇒ Ventilation object format = output status or counter	stopped **speed 1** speed 2 speed 3
	If Control type *= n speeds, this parameter has n+1 options. It is not possible to set a timeout for an unavailable speed.*	
Speed at timeout	General ⇒ Communication object = percentage or General ⇒ Ventilation object format = percentage	stopped from **10%** to 100%
Disable from bus		**no**/yes
Signal	Disable from bus = yes	**not inverted** inverted
Fan speed limit from bus	Control type > 1 speed (relay)	**not limited** speed 1 speed 2
Start delay		**0 s** [other values in the range 10 s … 12 min]
	Also displayed if warm start function mode through water temperature measurement on heat exchange coil is used. This function is active in both heating and cooling modes.	

Parameter name	*Conditions*	*Values*
Stop delay		**0 s** [other values in the range 10 s … 12 min]
	This function allows extending the fan operation, dissipating heat in the environment and residual cool in the heat exchange coil. This function is active in both heating and cooling modes.	

There are other parameters to be added, which allow configuring the physical outputs' behaviour (relays or 0 … 10V signal) depending on:

Control type and *General ⇒ Ventilation object format* or *General ⇒ Command object format*

Five different configuration cases are identified. Please see the table below to identify the proper configuration.

	Ventilation object format	
Control type	output status or counter	percentage
1-2-3 speeds (relays)	C1	C3
3 speeds (0…10V)	C2	C4
continuous regulation (0… 10V)		C5

C1 configuration

General ⇒ Ventilation object format = output status or counter

Control type =1 speed (relay) or 2 speeds (relays) or 3 speeds (relays)

In this case, NO parameter has to be added: the speed is already determined by the output values or by the counter. If the counter is set to a higher speed than the set value, the value is ignored; e.g. if Control type = 2 speeds (relays) and counter value = 3, this value is ignored.

C2 configuration

General ⇒ Ventilation object format = output status or counter

Control type = 3 speeds (0… 10V)

In this case, the voltage value for each speed to be assigned to the output signals must be provided.

Parameter name	*Conditions*	*Values*
Output value at the first speed	*General⇒Ventilation object format = output* status or counter Control type = 3 speeds (0… 10V)	**20%** [range 0.. 100%]
Output value at the second speed	*General⇒ Ventilation object format = output* status or counter Control type = 3 speeds (0… 10V)	**40%** [range 0.. 100%]
Output value at the third speed	*General⇒ Ventilation object format = output* status or counter Control type = 3 speeds (0… 10V)	**70%** [range 0.. 100%]

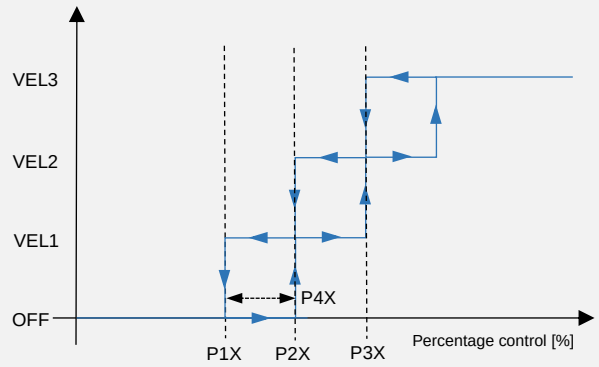
C3 configuration

General ⇒ Ventilation object format = percentage

Control type = 1 speed (relay) or 2 speeds (relays) or 3 speeds (relays)

In this case, in order to activate a speed, the percentage value of the activation threshold must be provided; it is also necessary to specify a hysteresis value.

Parameter name	*Conditions*	*Values*
First speed threshold [%]	*General⇒ Ventilation object format = percentage* Control type=1-2-3 speeds (relays)	**10%** [range 0.. 100%]
Second speed threshold [%]	*General⇒ Ventilation object format = percentage* Control type=2-3 speeds (relays)	**40%** [range 0.. 100%]
Third speed threshold [%]	*General⇒ Ventilation object format = percentage* Control type = 3 speeds (relays)	**70%** [range 0.. 100%]
Hysteresis [%]	*General⇒ Ventilation object format = percentage* Control type=1-2-3 speeds (relays)	**10%** [range 0.. 20%]



Parameters table	
P1X	First speed threshold [%]
P2X	Second speed threshold [%]
P3X	Third speed threshold [%]
P4X	Hysteresis [K]

Configurazione C4

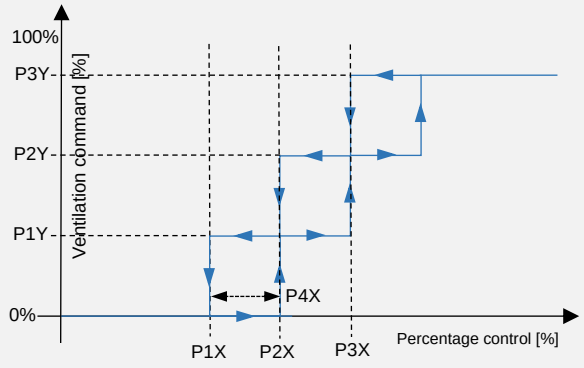
General ⇒ Ventilation object format = percentage

Control type = 3 speeds (0… 10V)

In this case, in order to activate a speed, the percentage value of the activation threshold must be provided; it is also necessary to specify a hysteresis value (like in C3 configuration). An output percentage for each speed (i.e. voltage to be sent on the output for each speed) must be also provided.

Parameter name	*Conditions*	*Values*
First speed threshold [%]	*General⇒ Ventilation object format = percentage* Control type = 3 speeds (0… 10V)	**10%** [range 0.. 100%]
Second speed threshold [%]	*General⇒ Ventilation object format = percentage* Control type = 3 speeds (0… 10V)	**40%** [range 0.. 100%]

Parameter name	*Conditions*	*Values*
Third speed threshold [%]	*General⇒ Ventilation object format = percentage* Control type = 3 speeds (0… 10V)	**70%** [range 0.. 100%]
Hysteresis [%] (*)	*General⇒ Ventilation object format = percentage* Control type = 3 speeds (0… 10V)	**10%** [range 0.. 20%]
Output value at the first speed	*General⇒ Ventilation object format* = percentage Control type = 3 speeds (0… 10V)	**20%** [range 0.. 100%]
Output value at the second speed	*General⇒ Ventilation object format* = percentage Control type = 3 speeds (0… 10V)	**40%** [range 0.. 100%]
Output value at the third speed	*General⇒ Ventilation object format* = percentage Control type = 3 speeds (0… 10V)	**70%** [range 0.. 100%]



Parameters table	
P1X	First speed threshold [%]
P2X	Second speed threshold [%]
P3X	Third speed threshold [%]
P4X	Hysteresis [%]
P1Y	Output value at the first speed [%]
P2Y	Output value at the second speed [%]
P3Y	Output value at the third speed [%]

C5 configuration

General ⇒ Ventilation object format = percentage

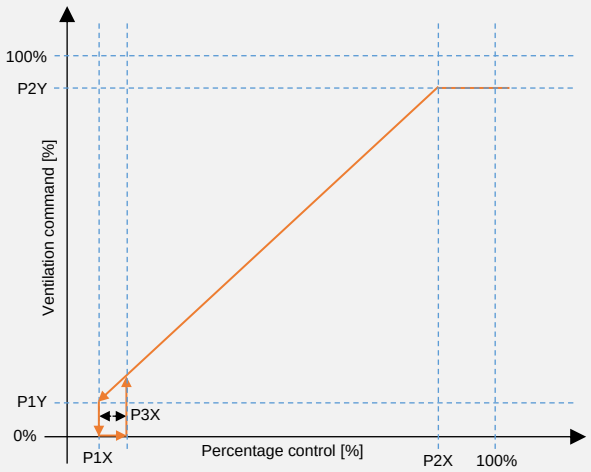
Control type =continuous regulation (0… 10V)

Below a specific command CO value, the inverter is switched off; above a specific command CO value, the inverter is brought at 100%.

Parameter name	*Conditions*	*Values*
Min value to control [%]	*General⇒ Ventilation object format* = percentage Control type = continuous regulation (0… 10V)	**15 %** [range 0 … 30 %]

Parameter name	*Conditions*	*Values*
Max value to control [%]	*General⇒ Ventilation object format* = percentage Control type = continuous regulation (0… 10V)	**85 %** [range 70 … 100 %]
Low limit continuous speed	*General⇒ Ventilation object format* = percentage Control type = continuous regulation (0… 10V)	**20 %** [range 0 … 30 %]
High limit continuous speed	*General⇒ Ventilation object format* = percentage Control type = continuous regulation (0… 10V)	**90%** [range 70 … 100 %]

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Fan continuous speed command	*General ⇒ Ventilation object format* = percentage	1 Byte	-WC---	[5.1] DPT_Scaling	226
Fan continuous speed command counter	*General ⇒ Ventilation object format* = counter	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount	226
First fan speed command	*General ⇒ Ventilation object format* = output status, Control type >= 1 speed (relay)	1 Bit	-WC---	[1.1] DPT_Switch	226
Second fan speed command	*General ⇒ Ventilation object format* = output status, Control type >= 2 speeds (relays)	1 Bit	-WC---	[1.1] DPT_Switch	227
Third fan speed command	*General ⇒ Ventilation object format* = output status, Control type = 3 speeds (relays)	1 Bit	-WC---	[1.1] DPT_Switch	228
Fan disable from bus	Disable from bus = yes	1 Bit	-WC---	[1.3] DPT_Enable	262
Fan speed limit from bus	Fan speed limit from bus ≠ not limited	1 Bit	-WC---	[1.3] DPT_Enable	274
Output V2 from bus	Control type = 1 speed (relay)	1 Bit	-WC---	[1.1] DPT_Switch	269
	This CO is automatically exposed in case the fan unit is configured to have only 1 speed, in order to assign the relay output to other functions: e.g. lighting control or ON/OFF actuations.				
Output V3 from bus	Control type = 1 speed (relay) o Control type = 2 speeds (relays)	1 Bit	C-W---	[1.001] switch	270
	This CO is automatically exposed in case the fan unit is configured to have 1-2 speeds, in order to assign the relay output to other functions: e.g. lighting control or ON/OFF actuations.				



Parameters table	
P1X	Min value to control [%]
P2X	Max value to control [%]
P3X	Hysteresis [%]
P1Y	Low limit continuous speed [%]
P2Y	High limit continuous speed [%]

Heating valve

This folder allows setting the following parameters:

- the output default value in case of timeout after a command is issued;
- the activation of the valve protection function during inactivity periods.

This folder is enabled if Controller = external and

Application = convector or fan-coil and Function = heating or

Application = convector or fan-coil and Function = both heating and cooling and Installation type = 4 pipes

If Controller = internal, this folder is not enabled and an equivalent folder is activated inside the Temperature control folder.

Parameter name	***Conditions***	***Values***
Valve position after timeout		**OFF**/ON
Control can be disabled from bus		**no**/yes
Signal	Control can be disabled from bus = yes	**not inverted** inverted
Antiscuff protection		**disabled** enabled
	It enables the function that activates the drive for the valve control during periods of inactivity of the system.	
Frequency	Antiscuff protection = enabled	once a day, **once a week**, once a month
Time interval	Antiscuff protection = enabled	**10 s** [other value in the range 5 s … 20 min]

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm Obj. No.***
Heating valve command (on/off)		1 Bit	-WC---	[1.1] DPT_Switch	229
Heating valve disable from bus	Disable valve from bus = yes	1 Bit	-WC---	[1.3] DPT_Enable	264

Cooling valve

This folder allows setting the following parameters:

- the output default value in case of timeout after a command is issued;
- the activation of the valve protection function during inactivity periods.

This folder is enabled if *Controller* = external and

Application = convector or fan-coil and *Function* = cooling or

Application = convector or fan-coil and *Function* = both heating and cooling and *Installation type* = 4 pipes

If *Controller* = internal, this folder is not enabled and an equivalent folder is activated inside the *Temperature control* folder.

Parameter name	***Conditions***	***Values***
Valve position after timeout		**OFF**/ON
Control can be disabled from bus		**no**/yes
Signal	Control can be disabled from bus = yes	**not inverted** inverted
Antiscuff protection		**disabled** enabled
	It enables the function that activates the drive for the valve control during periods of inactivity of the system.	
Frequency	Antiscuff protection = enabled	once a day, **once a week**, once a month
Time interval	Antiscuff protection = enabled	**10 s** [other value in the range 5 s … 20 min]

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm Obj. No.***
Cooling valve command (on/off)		1 Bit	-WC---	[1.1] DPT_Switch	230
Cooling valve disable from bus	Disabilita valvola dal bus = si	1 Bit	-WC---	[1.3] DPT_Enable	265

Heating/cooling valve

This folder allows setting the following parameters:

- the output default value in case of timeout after a command is issued;
- the activation of the valve protection function during inactivity periods.

This folder is enabled if *Controller* = external and

Application = convector or fan-coil and *Function* = cooling or

Application = convector or fan-coil and *Function* = both heating and cooling and *Installation type* = 2 pipes

If *Controller* = internal, this folder is not enabled.

Parameter name	***Conditions***	***Values***
Valve position after timeout		**OFF**/ON
Control can be disabled from bus		**no**/yes
Signal	Control can be disabled from bus = yes	**not inverted** inverted
Antiscuff protection		**disabled** enabled
	It enables the function that activates the drive for the valve control during periods of inactivity of the system.	
Frequency	Antiscuff protection = enabled	once a day, **once a week**, once a month
Time interval	Antiscuff protection = enabled	**10 s** [other value in the range 5 s … 20 min]

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm Obj. No.***
Heating/cooling valve command (on/off)		1 Bit	-WC---	[1.1] DPT_Switch	229
Heating/cooling valve disable from bus	Disabilita valvola dal bus = si	1 Bit	-WC---	[1.3] DPT_Enable	264

11.7.3 Internal controller

This folder is enabled if *Controller* = internal.

If *Controller* = external, the folder is active but empty.

Settings

The Settings folder contains the parameter allowing to perform the basic configuration of the room temperature controller:

- setpoint type: single or relative;
- operating mode change over;
- antiscuff protection activation.

Parameter name	***Conditions***	***Values***
Setpoint type		single **relative**
	In case the option "Single" is selected and Function = heating, the temperature controller acts on heating mode; in case Function = cooling, the temperature controller acts on cooling mode. *In case Function = both heating and cooling, the current seasonal mode needs to be specified by the proper communication object.*	
Setpoint CO type	Setpoint type = Relative	**absolute** relative
Cyclic sending interval		**no sending** [other values in the range 30 s … 120 min]
	In case Setpoint type = single, the actual setpoint value takes only into account the actual state of the contacts window (if the corresponding function is enabled). *In case Setpoint type = relative, the actual setpoint value also depends on the operating mode set manually by the user or automatically by another KNX supervising device with the possibility of time scheduling.*	
Heating/cooling changeover	Function = both heating and cooling, Setpoint type = Relative	**from bus** automatic from room temperature (*) automatic from flow temperature
	In case Setpoint type = single, the heating-cooling changeover must be carried out from bus.	
Temperature for changeover (heating)	Heating/cooling changeover = automatic from flow temperature; Input (X) = flow temperature sensor or External sensors (from bus) ⇒ Flow temperature = enabled,	**35** [range 20°C … 50°C]
Temperature for changeover (cooling)	Heating/cooling changeover = automatic from flow temperature; Input (X) = flow temperature sensor or External sensors (from bus) ⇒ Flow temperature = enabled,	**16** [range 5°C … 20°C]
Changeover cyclic sending interval	Function = both heating and cooling	**no sending** [other values in the range 30 s … 120 min]
Antiscuff protection		**disabled** enabled
	It enables the function that activates the drive for the valve control during periods of inactivity of the system.	

Parameter name	*Conditions*	*Values*
Frequency	Antiscuff protection = enabled	once a day **once a week** once a month
Time interval	Antiscuff protection = enabled	**10 s** [other values in the range 5 s … 20 min]

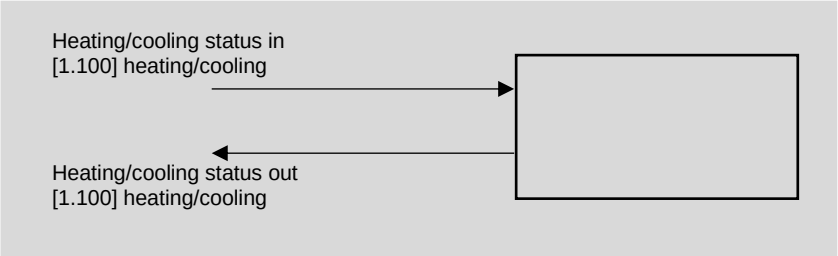
Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Thermal generator lock alarm		1 Bit	-WC---	[1.5] DPT_Alarm	6
	This CO is used by an external device to stop the operation as actuator or controller. In case Controller = internal, when receiving the alarm, the internal temperatre controller is deactivated.				
Temperature control alarm	Controller = internal	1 Bit	R-CT--	[1.5] DPT_Alarm	7
	Internal temperature controller alarm with regulation disabling. The alarm is activated on one of the following conditions: • *failure on one of the temperature sensors used for control* • *timeout when receiving the value of one of the temperature sensors used for control* • *drip tray alarm (if configured differently from simple warning)*				
Actual setpoint		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	244
Heating/cooling status out	Always visible	1 Bit	R-CT--	[1.100] DPT_Heat_Cool	240
	The communication object is updated on the bus to the switching event internally processed by the controller. The object is always exposed and contains information on the current conduction mode of the internal temperature controller. **[1.100] DPT Heat/Cool 1 Bit** **0 = Cool** **1 = Heating**				
Heating/cooling changeover command	Function = both heating and cooling, Heating/cooling changeover = from bus	1 Bit	-WC---	[1.100] DPT_Heat_Cool	241
	The communication object is received from the bus. At the switching event, the internal regulator switches the conduction mode.				
HVAC mode in	Setpoint type = Relative	1 Byte	-WC---	[20.102] DPT_HVACMode	242
	Bits 5 to 8 are reserved. **[20.102] DPT HVAC Mode 1 Byte** AUTO 0 0 0 0 — COMFORT 0 0 0 1 — STAND-BY 0 0 1 0 ECONOMY 0 0 1 1 — PROTECTION 0 1 0 0				
HVAC manual mode	Setpoint type = Relative	1 Byte	-WC---	[20.102] DPT_HVACMode	243

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
HVAC mode out	Setpoint type = Relative	1 Byte	R-CT--	[20.102] DPT_HVACMode	245
HVAC chrono mode active status	Setpoint type = Relative	1 Bit	RWCTU-	[1.11] DPT_State	244

Monitoring and remote control of the conduction mode

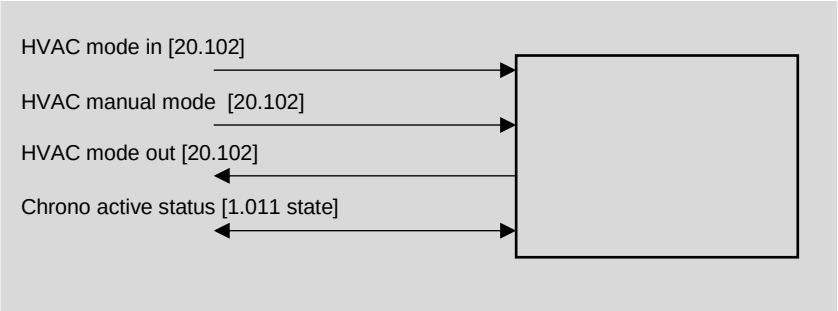
The communication objects indicated in the block diagram allow you to monitor and modify the current conduction mode imposed on the internal temperature controller of the device. The object Heating / cooling out status is always exposed, even when the function of the regulator is only heating or cooling only. In the event that the Function is both heating and cooling, the cyclic sending of the object on the bus can be enabled; in all cases the information on the current conduction mode can be acquired with a read request to this communication object.

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



Remote operative mode modification

The communication objects shown in figure allow monitoring the operating mode (comfort, standby, economy and building protection) modifications performed by a supervisor software or the operating mode forced by chrono program.



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

Heating

The *Heating* folder allows setting:

- the default value for single or relative setpoint (comfort setpoint and standby / economy attenuations);
- the type of regulation algorithm

This folder is active if *General* ⇒ *Controller* =internal and

General ⇒ *Function* = heating or both heating and cooling.

Parameter name	***Conditions***	***Values***
Temperature setpoint [°C]	Setpoint type = Single	**21** [range 10 … 50]
Comfort setpoint [°C]	Setpoint type = Relative	**21** [range 10 … 50]
Standby offset [0,1 K]	Setpoint type = Relative,	**- 30** [range -10 … -50]
Economy offset [0,1 K]	Setpoint type = Relative	**-50** [range -10 … -50]
Building protection [°C]		**7** [range 2 … 10]
[…]		
	Parameters about the type of regulation algorithm for the valves.	
Control type		**on/off**, PWM (pulse width modulation)
Hysteresis	Control type = 2-point hysteresis	**0,3 K** [other values in the range 0,2 K … 3 K]
PWM cycle time	Control type = PWM (pulse width modulation)	**15 min** [range 5 … 240 min]
Min. control value [%]	Control type = PWM (pulse width modulation)	**15 %** [range 0 %…30 %]
Max. control value [%]	Control type = PWM (pulse width modulation)	**85 %** [range 70 %…100 %]
Proportional band [0,1 K]	Control type = PWM (pulse width modulation)	**30** [range 0 … 255]
Integral time [min]	Control type = PWM (pulse width modulation)	**0** [range 0 … 255 min]
Min. change of value to send [%]	Control type = PWM (pulse width modulation)	**10 %** [range 0 %…100 %]
	The duration of the ON command during a PWM cycle time is modified when the output percentage of the controller changes inside the range specified by this parameter.	

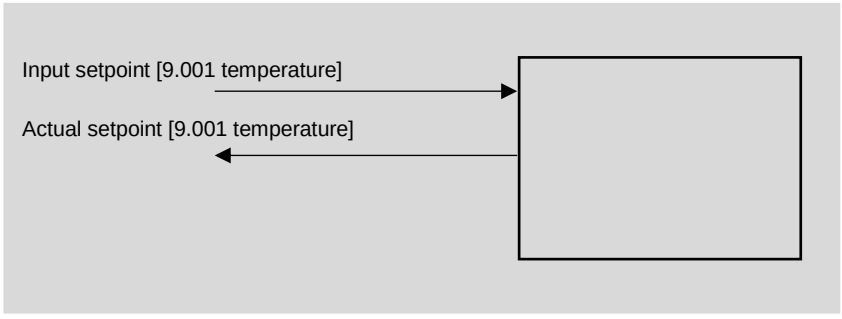
Parameter name	*Conditions*	*Values*
[…]		
Cyclic sending interval		**no sending** [other values in the range 30 s … 120 min]
	This parameter allows cyclically sending on the bus the value of the temperature controller output (CO 61). Through this parameter, it is possible to send, in parallel, the output value to a different fan-coil device used as a simple actuator.	
Forced mode		**no**/yes
	This parameter allows commanding the controller output in manual / forced mode.	
Control can be disabled from bus		**no**/yes
Signal	Control can be disabled from bus = yes	**not inverted** inverted
Valve position feedback		**no**/yes
	In case of position feddback enabling without cyclic sending, the CO is updated when the device starts and on change of state.	
Cyclic sending interval	Valve position feedback = yes	**no sending** [other values in the range 30 s … 120 min]

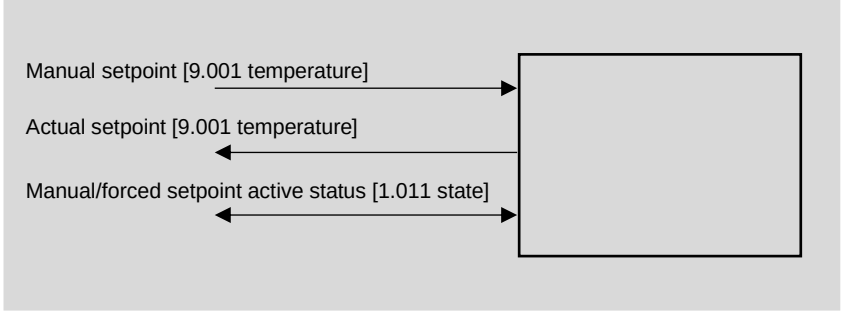
Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Actual setpoint		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	246
Input setpoint	Setpoint type = Single	2 Bytes	-WC---	[9.1] DPT_Value_Temp	247
Comfort setpoint (heating)	Setpoint type = Relative	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	248
Standby offset (heating)	Setpoint type = Relative, Setpoint CO type = relative	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd	250
Standby setpoint (heating)	Setpoint type = Relative, Setpoint CO type = absolute	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	250
Economy offset (heating)	Setpoint type = Relative, Setpoint CO type = relative	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd	252
Economy setpoint (heating)	Setpoint type = Relative, Setpoint CO type = absolute	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	252
Building protection (heating)		2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	254
Controller output forced from bus	Heating ⇒ Forced mode = yes or Cooling ⇒ Forced mode = yes	1 Byte	-WC---	[5.1] DPT_Scaling	266
Controller output automatic/forced from bus	Heating ⇒ Forced mode = yes or Cooling ⇒ Forced mode = yes	1 Bit	-WC---	[1.3] DPT_Enable	267

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Manual/forced setpoint active status	Setpoint type = Relative	1 Bit	RWCTU-	[1.11] DPT_State	256
Manual setpoint	Setpoint type = Relative	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	257
Heating valve disable from bus	Control can be disabled from bus = yes and Function = Heating or both heating and cooling 4 pipes	1 Bit	-WC---	[1.3] DPT_Enable	264
Heating/cooling valve disable from bus	Control can be disabled from bus = yes and Function = both heating and cooling 2 pipes	1 Bit	-WC---	[1.3] DPT_Enable	264
Heating valve status	Valve position feedback = yes and Function = Heating or both heating and cooling 4 pipes	1 Bit	R-CT--	[1.1] DPT_Switch	238
Heating/cooling valve status	Valve position feedback = yes and Function = both heating and cooling 2 pipes	1 Bit	R-CT--	[1.1] DPT_Switch	238

Remote Setpoint modification

The communication objects shown in figure allow to monitor the Setpoint forced modifications performed remotely, for example from a supervisor software.

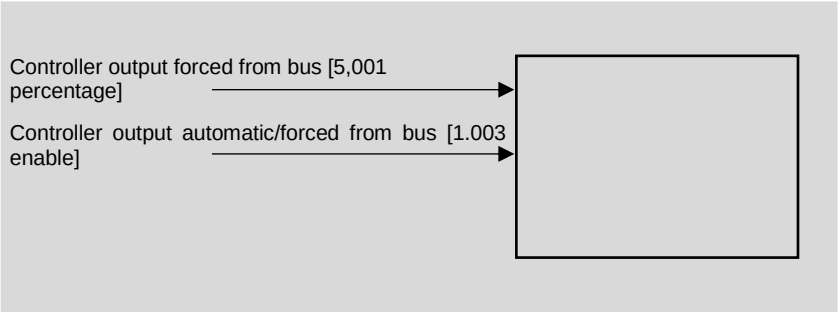




Those objects are about the Setpoint forced modification: alternatively, the supervisor can act directly on the operating mode setpoints. The value of the C.O. *Actual setpoint* represents the current operative setpoint, which the control algorithms are based on. O.C. *Manual/forced setpoint active status* indicates if the forced mode is active. The supervisor can force at any time the actual setpoint by writing a new value directly into the C.O. *Manual setpoint*. The C.O. *Manual/forced setpoint active status* can also be used in writing to exit from the active forced mode.

Output manual command

It is possible to manually force the controller output to a desired percentage value in order to test the ventilation. The forcing requires to take the controller output to Forced mode in the first place, then it is possible to select a controller output in a 0-100% range. Likewise, in order to go back to automatic operation, the CO *Controller output automatic/forced from bus* must be written.



Cooling

The *Cooling* folder allows setting:

- the default values for single or relative setpoints (comfort setpoint and standby / economy attenuations) in case of manual heating-cooling changeover;
- the default value for the dead-band for changeover and for standby / economy attenuations in case of automatic, based on internal conditions heating-cooling changeover;
- the type of regulation algorithm (2-point hysteresis, PWM) and internal parameters to control the valve.

This folder is active if *General* ⇒ *Controller* = internal and

General ⇒ *Function* = cooling or both heating and cooling.

Parameter name	***Conditions***	***Values***
Temperature setpoint	Setpoint type = Single	**23** [range 10 … 50]
Dead Band for change-over [0,1 K]	Setpoint type = Relative, Heating/cooling changeover = automatic	**20** [range 10 … 40]
Comfort setpoint [°C]	Setpoint type = Relative, Heating/cooling changeover = from bus	**23** [range 10 … 50]
Standby offset [0,1 K]	Setpoint type = Relative,	**30** [range 10 … 50]
Economy offset [0,1 K]	Setpoint type = Relative	**50** [range 10 … 80]
Building protection [°C]		**36** [range 20 … 50]
[…]		
	Parameters about the type of regulation algorithm for the valves.	
Control type		**on/off**, PWM (pulse width modulation)
Hysteresis	Control type = 2-point hysteresis	**0,3 K** [other value in the range 0,2 K … 3 K]
PWM cycle time	Control type = PWM (pulse width modulation)	**15 min** [range 5 … 240 min]
Min. control value [%]	Control type = PWM (pulse width modulation)	**15 %** [range 0 %…30 %]
Max. control value [%]	Control type = PWM (pulse width modulation)	**85 %** [range 70 %…100 %]
Proportional band [0,1 K]	Control type = PWM (pulse width modulation)	**30** [range 0 … 255]

Parameter name	*Conditions*	*Values*
Integral time [min]	Control type = PWM (pulse width modulation)	**0** [range 0 … 255 min]
Min change of value to send [%]	Control type = PWM (pulse width modulation)	**10 %** [range 0 %…100 %]
	The duration of the ON command during a PWM cycle time is modified when the output percentage of the controller changes inside the range specified by this parameter.	
[…]		
Cyclic sending interval		**no sending** [other values in the range 30 s … 120 min]
	This parameter allows cyclically sending on the bus the value of the temperature controller output. Through this parameter, it is possible to send, in parallel, the output value to a different fan-coil device used as a simple actuator.	
Forced mode		**no**/yes
	This parameter allows commanding the controller output in manual / forced mode.	
Control can be disabled from bus		**no**/yes
Signal	Control can be disabled from bus = yes	**not inverted** inverted
Valve position feedback		**no**/yes
	In case of position feddback enabling without cyclic sending, the CO is updated when the device starts and on change of state.	
Cyclic sending interval	Valve position feedback = yes	**no sending** [other values in the range 30 s … 120 min]

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Actual setpoint		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	246
Input setpoint	Setpoint type = Single	2 Bytes	-WC---	[9.1] DPT_Value_Temp	247
Comfort setpoint (cooling)	Setpoint type = Relative	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	249
Standby offset (cooling)	Setpoint type = Relative, Setpoint CO type = relative	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd	251
Standby setpoint (cooling)	Setpoint type = Relative, Setpoint CO type = absolute	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	251
Economy offset (cooling)	Setpoint type = Relative, Setpoint CO type = relative	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd	253
Economy setpoint (cooling)	Setpoint type = Relative, Setpoint CO type = absolute	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	253
Building protection (cooling)		2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	255

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Controller output forced from bus	Heating ⇒ Forced mode = yes or Cooling ⇒ Forced mode = yes	1 Byte	-WC---	[5.1] DPT_Scaling	266
Controller output automatic/forced from bus	Heating ⇒ Forced mode = yes or Cooling ⇒ Forced mode = yes	1 Bit	RWCTU-	[1.3] DPT_Enable	267
Manual/forced setpoint active status	Setpoint type = Relative	1 Bit	RWCTU-	[1.11] DPT_State	256
Manual setpoint	Setpoint type = Relative	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp	257
Cooling valve disable from bus	Control can be disabled from bus = yes and Function =Heating or both heating and cooling 4 pipes	1 Bit	-WC---	[1.3] DPT_Enable	265
Cooling valve status	Valve position feedback = yes and Function =Heating or both heating and cooling 4 pipes	1 Bit	R-CT--	[1.1] DPT_Switch	239

Setpoint remote modification and output manual command use the same settings as the Heating conduction mode; for a correct use of the exposed C.O.'s, please refer to function blocks reported in the *Heating* section.

External temperature compensation

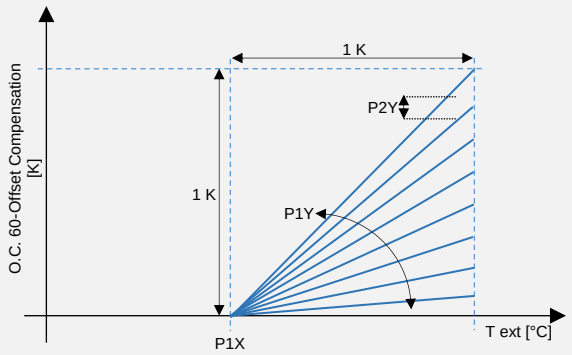
Temperature compensation can be performed in two ways:

- acquiring the external temperature by means of a sensor connected to an analog input. The device can send the acquired value to other fan-coil controller on the bus;
- acquiring the external temperature from bus by means of a KNX temperature sensor suitable for outdoor installation.

The folder is active if *General* ⇒ *Function* = Cooling or both heating and cooling and if an external temperature sensor is connected.

Parameter name	***Conditions***	***Values***
Summer compensation		**disabled** enabled
Lowest external compensation [°C]		**25 [°C]** [range 25 … 40 °C]
Coefficient		1 K each 8 K external temperature change 1 K each 7 K external temperature change 1 K each 6 K external temperature change 1 K each 5 K external temperature change **1 K each 4 K external temperature change** 1 K each 3 K external temperature change 1 K each 2 K external temperature change 1 K each 1 K external temperature change
Min change to send value		**0,6 K** [range 0 … +5 K]
	If the parameter is set to the value 0, no value is sent to the change.	
Cyclic sending interval		**no sending** [other values in the range 30 s … 120 m]

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm Obj. No.***
Actual setpoint		2 Bytes	R-CT--	[9.1] DPT_Value_Temp	246
Cooling compensation offset		2 Bytes	R-CT--	[9.2] DPT_Value_Tempd	258



Parameters table	
P1X	Lowest external temperature [°C]
P1Y	Coefficient
P2Y	Min change to send value [K]

Ventilation

The Ventilation tab contains:

the parameters for setting the relay or continuous control outputs and the behavior parameters of the outputs according to the value of the commands received from the bus

- Cold start settings
- Hot start settings
- Anti-stratification function settings
- start-up delay and shutdown of the fan
- activation of speed limitation (for example during night hours)
- activation of ventilation speed feedback

The situations that are generated are different from the situations and parameters that are found in the case of use as an actuator. The major difference is that in the case of use of the device as an actuator reference is made to the N speed threshold [%] in percent, with internal regulator reference is made to the speed threshold N [K] in degrees kelvin as an error between the operating Set and the measured T.

This tab is active if *Heating, cooling and ventilation* ⇒ *Configuration* ⇒ *Controller* = *internal* and *Application* = *fan-coil or ventilation* .

Parameter name	***Conditions***	***Values***
Control type		**1 speed (relais)** 2 speed (relais) 3 speed (relais) 3 speed (0.. 10V) continuous regulation
[…]		
	Setting parameters that depend on the configurations adopted. Consult the different situations later.	
Warm start	Configuration ⇒ Function = heating or heating and cooling, Inputs configuration ⇒ Input X ⇒ [AI] heat exchange coil battery temperature sensor or External sensors (from bus) ⇒ coil temperature = enabled	**no** / yes
	To perform the function, at least one sensor must be enabled to measure the water temperature at the heat exchanger coil of the fan-coil. Optionally, it can be an input configured as analog or an external sensor (from the bus).	
Heat exchange coil temperature [°C]	Warm start = yes	**35** [range 28 … 40]
Cold start	Configuration ⇒ Function = heating or heating and cooling, Inputs configuration ⇒ Input X ⇒ [AI] heat exchange coil battery temperature sensor or External sensors (from bus) ⇒ coil temperature = enabled	**no** / yes
Heat exchange coil temperature [°C]	Cold start = yes	**12** [range 7 … 18]

Parameter name	*Conditions*	*Values*
Antistratification function	Inputs configuration ⇒ Input X ⇒ [AI] antistratification temperature sensor or External sensors (from bus) ⇒ antistratification temperature sensor = enabled	**disabled /**enabled
	For carrying out the function, at least one sensor must be enabled to measure a second room temperature value at a different level from that of the thermostat. Optionally, it can be an input configured as analog or an external sensor (from the bus).	
Temperature differential	Antistratification function = enabled	**2 [K/m]** [other values in the range 0,25 … 4,00]
	The DIN 1946 standard recommends not exceeding 2 K / m for rooms with an ordinary height (between 2.70 and 3 m).	
Hysteresis	Antistratification function = enabled	**0,6 K** [other values in the range 0,2 … 3 K]
Disable from bus		**no** / yes
Signal	Disable from bus = yes	**not inverted** inverted
Fan speed limit from bus	Control type > 1 speed (relais)	**not limited** [other values in the range 10% … 90%]
	The parameter enables the possibility of forcing a predetermined and fixed speed from the bus. The typical application is in the hotel environment to limit noise during night hours.	
Start delay		**0 s** [other values in the range 10 s … 12 min]
	It also appears if you use the hot start mode by measuring the water temperature at the heat exchanger coil of the fan coil. The function is active in both conduction modes.	
Stop delay		**0 s** [other values in the range 10 s … 12 min]
	The function allows to prolong the operation of the fan, dissipating in the environment the heat or the residual cold present in the heat exchange coil. The function is active in both conduction modes.	
Control feedback		**no** / yes
Cyclic sending interval	Control feedback = yes	**no sending** [other values in the range 30 s … 120 min]
Time-scheduled antistratification	Control feedback = yes	**disabled** heating only cooling only
Frequency	Control feedback = yes	**30 min** [other values in the range 5 min … 60 min]
Time interval	Control feedback = yes	**3 min** [other values in the range 30 s … 5 min]

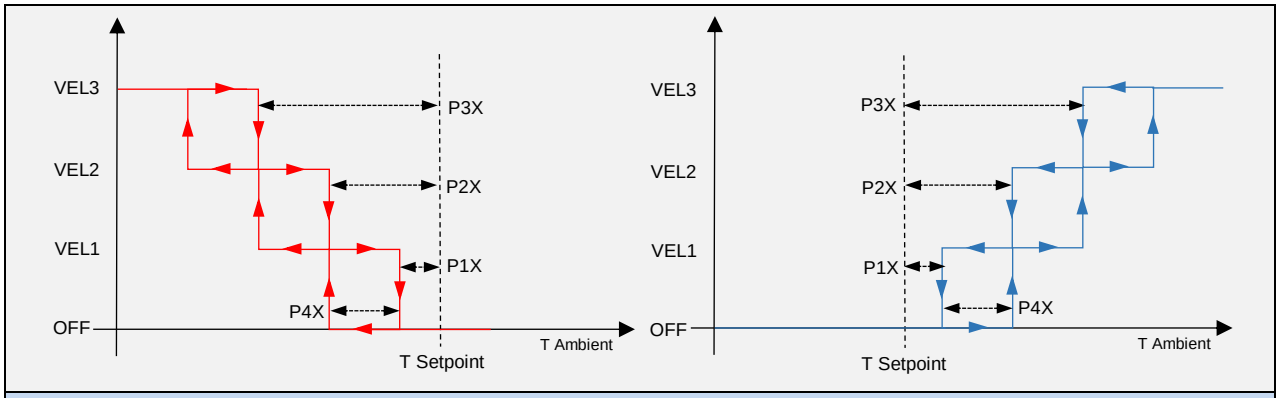
To these parameters must be added other parameters that allow to configure the behavior of the physical outputs (relay or 0 ... 10V signal).
A series of 3 different configurations is identified. Consult the table to identify the appropriate configuration.

Output type	
1-2-3 speed (relais)	C1
3 speed (0.. 10V)	C2
Continuous regulation (0.. 10V)	C3

Setting C1

Control type = 1 speed (relais), 2 speed (relais) and 3 speed (relais).

Parameter name	***Conditions***	***Values***
First speed threshold [0,1K]	Control type = 1-2-3 speed (relais)	**0** [range 0 … 255]
Second speed threshold [0,1K]	Control type = 2-3 speed (relais)	**10** [range 0 … 255]
Third speed threshold [0,1K]	Control type = 3 speed (relais)	**20** [range 0 … 255]
Hysteresis [K]	Control type = 1-2-3 speed (relais)	**0,3 K** [other value in the range 0,2 K … 3 K]



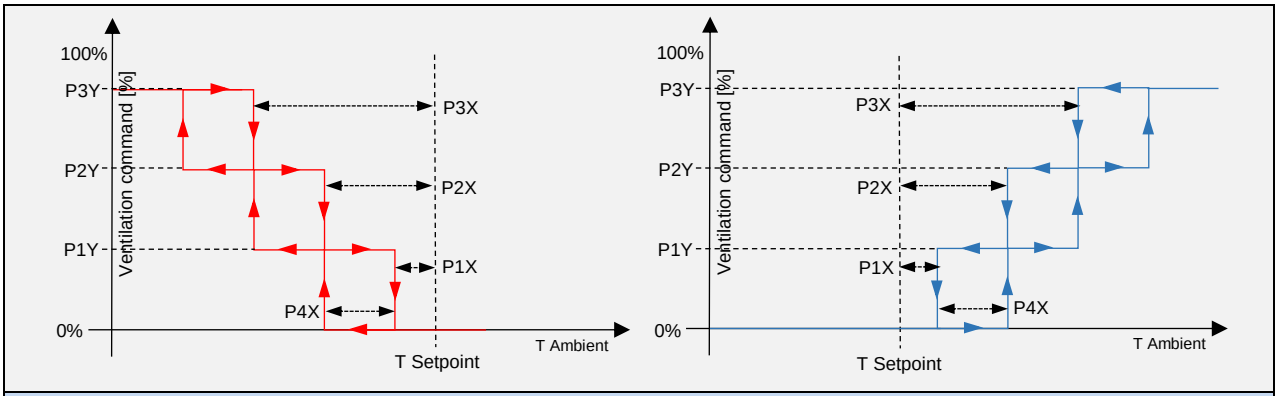
Parameters table	
P1X	First speed threshold [0,1 K]
P2X	Second speed threshold [0,1 K]
P3X	Third speed threshold [0,1 K]
P4X	Hysteresis [K]

Setting C2

Control type = 3 speed (0.. 10V)

In this case, simply provide the activation threshold to activate a speed and enter a hysteresis value as in Situation 3. However, it is also necessary to enter the output percentage at a speed (what voltage to put on the output signal at a speed).

Parameter name	*Conditions*	*Values*
First speed threshold [0,1K]	*Control type = 3 speed (0.. 10V)*	**0** [range 0 … 255]
Second speed threshold [0,1K]	*Control type = 3 speed (0.. 10V)*	**10** [range 0 … 255]
Third speed threshold [0,1K]	*Control type = 3 speed (0.. 10V)*	**20** [range 0 … 255]
Hysteresis [0,1K]	*Control type = 3 speed (0.. 10V)*	**0,3 K** [other values in the range 0,2 K … 3 K]
Output value at the first speed	*Control type = 3 speed (0.. 10V)*	**20%** [range 0.. 100%]
Output value at the second speed	*Control type = 3 speed (0.. 10V)*	**40%** [range 0.. 100%]
Output value at the third speed	*Control type = 3 speed (0.. 10V)*	**70%** [range 0.. 100%]



Parameters table	
P1X	First speed threshold [0,1 K]
P2X	Second speed threshold [0,1 K]
P3X	Third speed threshold [0,1 K]
P4X	Hysteresis [K]
P1Y	Output value at the first speed [%]
P2Y	Output value at the second speed [%]
P3Y	Output value at the third speed [%]

Setting C3

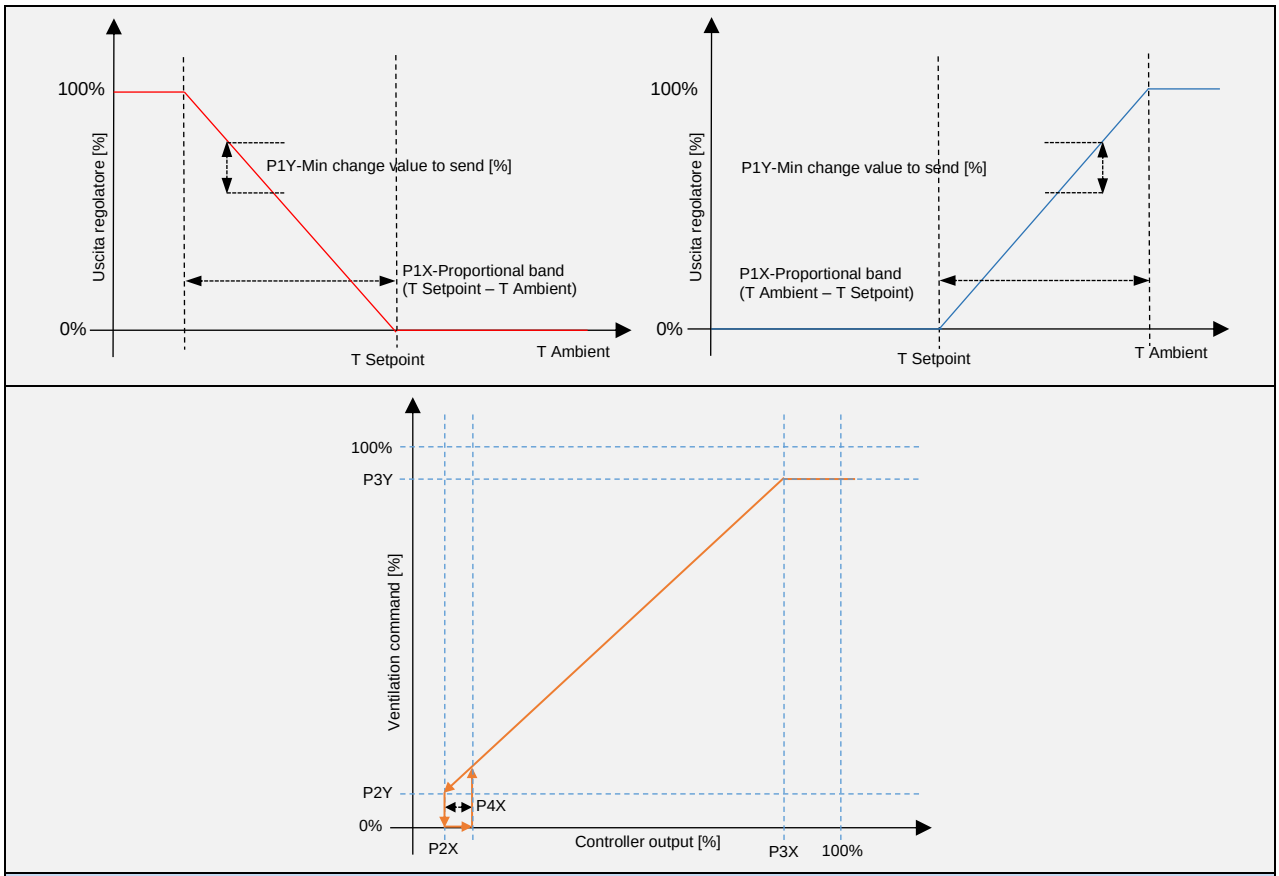
Control type = continuous regulation (0.. 10V).

Below a certain value of the command comunication object, the inverter is switched off; above a certain value of the command C.O., the inverter is brought to maximum.

Proportional Band [0,1 K]	*Control type = continuous regulation [0… 10V]*	**30** [range 0 … 255]

Integral time [min]	*Control type = continuous regulation [0… 10V]*	**0** [range 0 … 255]
Speed control low limit [%]	*Control type = continuous regulation [0… 10V]*	**0 %** [range 0 … 100 %]
Speed control high limit [%]	*Control type = continuous regulation [0… 10V]*	**100 %** [range 0 … 100 %]
Hysteresis [%]		**5 %** [range 0 … 30 %]
Speed output low limit [%]	*Control type = continuous regulation [0… 10V]*	**0 %** [range 0 … 100 %]
Speed output high limit [%]	*Control type = continuous regulation [0… 10V]*	**100%** [range 0 … 100 %]

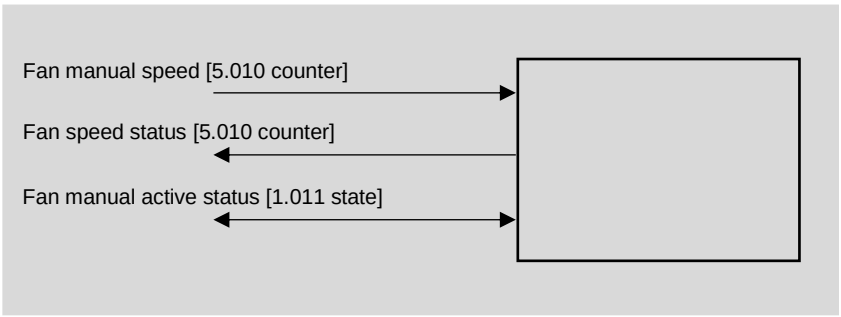
Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm Obj. No.***
Fan stopped status	Control feedback = yes	1 Bit	R-CT--	[1.1] DPT_Switch	233
First fan speed status	Control feedback = yes	1 Bit	R-CT--	[1.1] DPT_Switch	234
Second fan speed status	Control feedback = yes	1 Bit	R-CT--	[1.1] DPT_Switch	235
Third fan speed status	Control feedback = yes	1 Bit	R-CT--	[1.1] DPT_Switch	236
Counter speed fan status	Control feedback = yes	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount	237
Fan continuous speed command		1 Byte	-WC---	[5.1] DPT_Scaling	226
Fan continuous speed command counter		1 Byte	-WC---	[5.10] DPT_Value_1_Ucount	226
Fan manual/auto status		1 Bit	RWCTU-	[1.3] DPT_Enable	263
Fan speed limit enable	Fan speed limit from bus ≠ not limited	1 Bit	-WC---	[1.3] DPT_Enable	274
Fan disable from bus	Fan speed limit from bus ≠ not limited	1 Bit	-WC---	[1.3] DPT_Enable	262

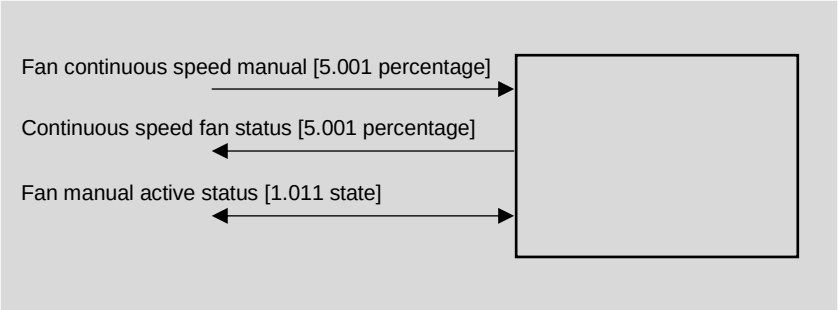


Parameters table	
P1X	Proportional band [0,1 K]
P2X	Min value to control [%]
P3X	Max value to control [%]
P4X	Hysteresis [%]
P1Y	Min change value to send [%]
P2Y	Low limit continuous speed [%]
P3Y	High limit continuous speed [%]

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 manually. The communication objects also allow performing the same modifications remotely, for example from a supervisor software.





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

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

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

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

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

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

Windows contacts

The Tab is active if the internal temperature controller is set and if a window contact sensor is connected to one of the inputs on the terminal board (Inputs tab) or if the contact status is detected via 1-2 communication objects (External inputs tab from bus). In the case of connecting several window contact sensors to multiple inputs on the terminal board or to multiple inputs and by acquiring a status via a communication object, the device processes the open window status with the call of the building protection operating mode by executing the logical OR of all inputs.

Parameter name	***Conditions***	***Values***
Window contact function		**disabled**/enabled
Time to wait before HVAC change	Window contact function = enable	**00:01:00 hh:mm:ss** [range 00:00:00 … 18:12:15]
	Time interval before automatic switching of the device in the building protection operating mode.	

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm Obj. No.***
Input BI9 – Windows contact status		1 Bit	R-CT--	[1.19] DPT_Window_Door	126
	(only EK-HO1-TP)*				
Windows contact 1 (from bus)		1 Bit	C-W--	[1.019] window/door	138
Windows contact 2 (from bus)		1 Bit	C-W--	[1.019] window/door	139

Presence sensors

The Tab is active if the internal temperature controller is set and if 1-2 presence sensors are connected via the related communication objects (External inputs from bus Tab). In the case of connection of 2 presence sensors, the device processes the extension and / or comfort limitation status by executing the logical OR of all the O.C. states input.

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

Object name	***Conditions***	***Dim.***	***Flags***	***DPT***	***Comm Obj. No.***
Presence sensor 1 (from bus)	Inputs (from bus) ⇒ Presence sensor 1 = enabled	1 Bit	-WCTU-	[1.18] DPT_Occupancy	140
Presence sensor 2 (from bus)	Inputs (from bus) ⇒ Presence sensor 2 = enabled	1 Bit	-WCTU-	[1.18] DPT_Occupancy	141

11.7.4 Drip tray control

The board allows you to customize the management of the condensate collection system during cooling operation with the following parameters:

- Type of reaction to reach the maximum level in the collection system (action on the exchange coil valve and / or the fan)
- Cyclical sending of the status (for example, to activate an emptying pump for the collection system)

The tab is active if the fan-coil application is selected: *General* ⇒ *Application* = *fan-coil*.

The function can be activated if a condensate level control sensor is connected to the inputs (Inputs tab) in the terminal board or a communication object is acquired (External sensors card from the bus).

Parameter name	*Conditions*	*Values*
Drip control		**disabled** enabled
	If no condensate level sensor is enabled, the following information message appears: "For the condensate level control function, enable the relative sensor in Inputs or in External sensors (from the bus)".	
In case of drip	Drip control = enabled	**close valve and switch fan off** close valve and fan to min speed close valve and fan to max speed warning only

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Input BIx-Generic contact status		1 Bit	R-CT--	[1.1] DPT_Switch	
Drip tray contact		1 Bit	-WCTU-	[1.5] DPT_Alarm	137
Drip tray status		1 Bit	RWCTU-	[1.5] DPT_Alarm	9

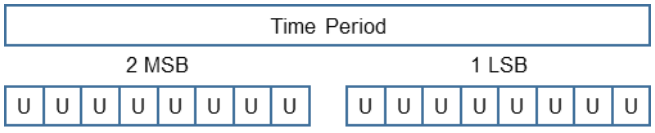
In the case of connection of several condenser level sensors to several inputs in the terminal board or to several inputs and through acquisition of a status via a communication object, the device processes the state of the O.C. through the logical OR of all inputs.

11.7.5 Filter monitoring

The filter monitoring function calculates the ventilation usage time and allows you to send an alert after a set period of operation (in weeks) providing an indication of the need to replace the air purification filters. It is also possible to send an object on the bus that indicates the total time of operation of the ventilation (in hours or seconds).

Tab Filter monitoring activation condition: Heating, cooling and ventilation ⇒ Report filter change = yes.

Parameter name	*Conditions*	*Values*
Change filter warning after [weeks]		**16** [range 1…128 weeks]
Send warning		**on filter change only** cyclic always
Send usage time		**no sending (only reading)** on change only on change and cyclic sending
Cyclic sending interval		1 h / 2 h / 4 h / **24 h** / 2 times for week / once a week
Communication object type		**seconds [DPT 13.100]** hours [DPT 7.007]

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Fan working time (hours)	Heating, cooling and ventilation ⇒ Report filter change = yes, Communication object type = hours [DPT 7.7]	2 Bytes	R-CT--	[7.7] DPT_Time_Hours	260
	 *The usage time is represented as an unsigned integer and has a range of [0 ... 65535] hours.*				
Fan working time (seconds)	Heating, cooling and ventilation ⇒ Report filter change = yes, Communication object type = seconds [DPT 13.1]	4 Bytes	R-CT--	[13.1] DPT_LongDeltaTimeSec	261
Filter change warning	Heating, cooling and ventilation ⇒ Report filter change = yes,	1 Bit	RWCTU-	[1.5] DPT_Alarm	8
	The communication object has 2 functions. As a transmission object, it sends cyclically or on event an ON binary status when the number of weeks of ventilator use set has been exceeded. As a receiving object, only when it is in the ON binary state, it can be changed to OFF: the effect is to reset the Fan use time object.				

11.8 Logic Functions

The EK-HO1-TP and EK-HU1-TP products provide useful combinational functions of the AND, OR, NOT and OR type, exclusive to perform complex functions in the building automation system.

They are available and configurable:

- 4 channels of logical functions
- 4 inputs for each channel

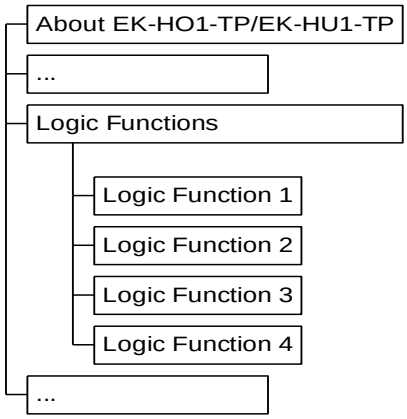
To each of these objects can be individually applied, if desired, a denial operator that reverses its value.

For each of the 8 channels, the parameter *Delay after bus voltage recovery* has been inserted: this parameter represents the time interval between the bus voltage recovery and the first reading of the input communication objects for the evaluation of the logic functions.

In case of incorrect connection of input communication objects or electrical problems on the bus for which the input reading request does not provide a positive result, the logic output of the corresponding channel can be calculated by setting the default Values for the inputs.

The communication object that represents the output of the logic function is sent on the bus on the event, at each variation of its state; alternatively, cyclic sending can be set at fixed intervals.

Activation condition: *Logic functions = Logic function x = 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** [range 00:00:00.000 … 00:10:55.350]
	Time interval between the bus voltage recovery and the first reading of the input communication objects for the evaluation of the logic functions.	
Output cyclic transmission interval		**no sending** [other values in the range 30 s … 120 min]
	No sending means that the output status of the logic function is updated on the bus only to a change. Different intervals imply cyclic sending of the output status on the bus.	
Output send		**both values** only value 0

Parameter name	*Conditions*	*Values*
		only value 1
Output updating		**on value changes** on change of value or input
Logic object x		**disabled /**enabled
Logic object x negated	Logic object x = enabled	**no** / yes
	By denying the logic state of the corresponding input, it is possible to carry out complex combinational logics. Example: Output = (NOT (Logical Object 1) OR Logical 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

Object name	*Conditions*	*Dim.*	*Flags*	*DPT*	*Comm Obj. No.*
Logic function X – Input 1	Logic function = enabled Logic object x = enabled	1 Bit	-WC---	[1.1] DPT_Switch	276, 281, 286, 291
Logic function X – Input 2	Logic function = enabled Logic object x = enabled	1 Bit	-WC---	[1.1] DPT_Switch	277, 282, 287, 292
Logic function X – Input 3	Logic function = enabled Logic object x = enabled	1 Bit	-WC---	[1.1] DPT_Switch	278, 283, 288, 293
Logic function X – Input 4	Logic function = enabled Logic object x = enabled	1 Bit	-WC---	[1.1] DPT_Switch	279, 284, 289, 294
Logic function X – Output	Logic function = enabled	1 Bit	-WC---	[1.1] DPT_Switch	280, 285, 290, 295

12 Appendix

12.1 List of communication objects

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

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

The communication objects marked with (*) refer only to the EK-HO1-TP device. All other communication objects refer to both the EK-HO1-TP and EK-HU1-TP devices.

Nr.	Name	Size	Flags	Datapoint type
1	Manual mode active	1 Bit	R-CT--	[1.3] DPT_Enable
2	Disable front pushbuttons	1 Bit	-WC---	[1.2] DPT_Bool
3	Technical alarm	1 Bit	R-CT--	[1.5] DPT_Alarm
4	Communication alarm	1 Bit	R-CT--	[1.5] DPT_Alarm
5	Device power off alarm	1 Bit	R-CT--	[1.5] DPT_Alarm
6	Thermal generator lock alarm	1 Bit	-WC---	[1.5] DPT_Alarm
7	Temperature control alarm	1 Bit	R-CT--	[1.5] DPT_Alarm
8	Filter change warning	1 Bit	RWCTU-	[1.5] DPT_Alarm
9	Drip tray status	1 Bit	RWCTU-	[1.5] DPT_Alarm
10	Alarm text	14 Bytes	R-CT--	[16.0] DPT_String_ASCII
AI1 Input				
11	Input AI1 Temperature value	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
11	Input AI1 outdoor temperature value	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
11	Input AI1 heat exchange coil battery sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
11	Input AI1 antistratification temperature sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
11	Input AI1 flow temperature sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
11	Input AI1 generic temperature sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
12	Input AI1 Temperature threshold 1 - Switch	1 Bit	R-CT--	[1.1] DPT_Switch
13	Input AI1 Temperature threshold 2 - Switch	1 Bit	R-CT--	[1.1] DPT_Switch
14	Input AI1 Temperature threshold 1 - Value	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
15	Input AI1 Temperature threshold 2 - Value	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
AI2 Input				
16	Input AI2 Temperature value	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
16	Input AI2 outdoor temperature value	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
16	Input AI2 heat exchange coil battery sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
16	Input AI2 antistratification temperature sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
16	Input AI2 flow temperature sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
16	Input AI2 generic temperature sensor	2 Bytes	R-CT--	[9.1] DPT_Value_Temp

Nr.	Name	Size	Flags	Datapoint type
17	Input AI2 Temperature threshold 1 – Switch	1 Bit	R-CT--	[1.1] DPT_Switch
18	Input AI2 Temperature threshold 2 – Switch	1 Bit	R-CT--	[1.1] DPT_Switch
19	Input AI2 Temperature threshold 1 – Value	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
20	Input AI2 Temperature threshold 2 - Value	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
BI1-BI2 Inputs				
21	Input BI1 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable
21	Inputs BI1 and BI2 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable
22.. 29	Input BI1 – Switching status *[type]*, object *n*	*table A1*	-WCTU-	*table A1 at the end of paragraph*
30	Input BI1 Dimming – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
30	Inputs B1 and BI2 Dimming – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
30	Input BI1 – Dedicated stop command	1 Bit	--CT--	[1.17] DPT_Trigger
30	Inputs BI1 and BI2 – Dedicated stop command	1 Bit	--CT--	[1.17] DPT_Trigger
30	Inputs BI1 and BI2 – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
31	Input BI1 – Dimming up / down / stop command	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
31	Inputs BI1 and BI2 – Dimming up / down / stop command	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
32	Input BI1 – Stop-step up / down command	1 Bit	--CT--	[1.7] DPT_Step
32	Inputs BI1 and BI2 – Stop-step up / down command	1 Bit	--CT--	[1.7] DPT_Step
33	Input BI1 – Move up / down command	1 Bit	--CT--	[1.8] DPT_UpDown
33	Inputs BI1 and BI2 – Move up / down command	1 Bit	--CT--	[1.8] DPT_UpDown
34	Input BI1 – Scene number	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
35	Input BI2 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable
36.. 43	Input BI2 – Switching status *[type]*, object *n*	*table A1*	-WCTU-	*table A1 at the end of paragraph*
44	Input BI2 Dimming – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
44	Input BI2 – Dedicated stop command	1 Bit	--CT--	[1.17] DPT_Trigger
45	Input BI2 – Dimming up / down / stop command	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
46	Input BI2 – Stop-step up / down command	1 Bit	--CT--	[1.7] DPT_Step
47	Input BI2 – Move up / down command	1 Bit	--CT--	[1.8] DPT_UpDown
48	Input BI2 – Scene number	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
BI3-BI4 Inputs				
49	Input BI3 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable
49	Inputs BI3 and BI4 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable

Nr.	Name	Size	Flags	Datapoint type
50.. 57	Input BI3 – Switching status *[type]*, object *n*	*table A1*	-WCTU-	*table A1 at the end of paragraph*
58	Input BI3 Dimming – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
58	Inputs B3 and BI4 Dimming – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
58	Input BI3 – Dedicated stop command	1 Bit	--CT--	[1.17] DPT_Trigger
58	Inputs BI3 and BI4 – Dedicated stop command	1 Bit	--CT--	[1.17] DPT_Trigger
58	Inputs BI3 and BI4 – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
59	Input BI3 – Dimming up / down / stop command	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
59	Inputs BI3 and BI4 – Dimming up / down / stop command	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
60	Input BI3 – Stop-step up / down command	1 Bit	--CT--	[1.7] DPT_Step
60	Inputs BI3 and BI4 – Stop-step up / down command	1 Bit	--CT--	[1.7] DPT_Step
61	Input BI3 – Move up / down command	1 Bit	--CT--	[1.8] DPT_UpDown
61	Inputs BI3 and BI4 – Move up / down command	1 Bit	--CT--	[1.8] DPT_UpDown
62	Input BI3 – Scene number	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
63	Input B4 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable
64.. 71	Input BI4 – Switching status *[type]*, object *n*	*table A1*	-WCTU-	*table A1 at the end of paragraph*
72	Input BI4 Dimming – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
72	Input BI4 – Dedicated stop command	1 Bit	--CT--	[1.17] DPT_Trigger
73	Input BI4 – Dimming up / down / stop command	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
74	Input BI4 – Stop-step up / down command	1 Bit	--CT--	[1.7] DPT_Step
75	Input BI4 – Move up / down command	1 Bit	--CT--	[1.8] DPT_UpDown
76	Input BI4 – Scene number	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
BI5-BI6 Inputs				
77	Input BI5 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable
77	Inputs BI5 and BI6 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable
78.. 85	Input BI5 – Switching status *[type]*, object *n*	*table A1*	-WCTU-	*table A1 at the end of paragraph*
86	Input BI5 Dimming – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch
86 (*)	Inputs B5 and BI6 Dimming – Switching command (* only EK-HO1-TP device)	1 Bit	-WCTU-	[1.1] DPT_Switch
86	Input BI5 – Dedicated stop command	1 Bit	--CT--	[1.17] DPT_Trigger
86 (*)	Inputs BI5 and BI6 – Dedicated stop command (* only EK-HO1-TP device)	1 Bit	--CT--	[1.17] DPT_Trigger
86 (*)	Inputs BI5 and BI6 – Switching command	1 Bit	-WCTU-	[1.1] DPT_Switch

Nr.	Name	Size	Flags	Datapoint type
	(* only EK-HO1-TP device)			
87	Input BI5 – Dimming up / down / stop command	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
87 (*)	Inputs BI5 and BI6 – Dimming up / down / stop command (* only EK-HO1-TP device)	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
88	Input BI5 – Stop-step up / down command	1 Bit	--CT--	[1.7] DPT_Step
88 (*)	Inputs BI5 and BI6 – Stop-step up / down command (* only EK-HO1-TP device)	1 Bit	--CT--	[1.7] DPT_Step
89	Input BI5 – Move up / down command	1 Bit	--CT--	[1.8] DPT_UpDown
89 (*)	Inputs BI5 and BI6 – Move up / down command (* only EK-HO1-TP device)	1 Bit	--CT--	[1.8] DPT_UpDown
90	Input BI5 – Scene number	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
91 (*)	Input B6 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC-U-	[1.3] DPT_Enable
92.. 99 (*)	Input BI6 – Switching status *[type]*, object *n* (* only EK-HO1-TP device)	*table A1*	-WCTU-	*table A1 at the end of paragraph*
100 (*)	Input BI6 Dimming – Switching command (* only EK-HO1-TP device)	1 Bit	-WCTU-	[1.1] DPT_Switch
100 (*)	Input BI6 – Dedicated stop command (* only EK-HO1-TP device)	1 Bit	--CT--	[1.17] DPT_Trigger
101 (*)	Input BI6 – Dimming up / down / stop command (* only EK-HO1-TP device)	4 Bit	--CT--	[3.7] DPT_Control_Dimming [3.8] DPT_Control_Blinds
102 (*)	Input BI6 – Stop-step up / down command (* only EK-HO1-TP device)	1 Bit	--CT--	[1.7] DPT_Step
103 (*)	Input BI6 – Move up / down command (* only EK-HO1-TP device)	1 Bit	--CT--	[1.8] DPT_UpDown
104 (*)	Input BI6 – Scene number (* only EK-HO1-TP device)	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
BI7 Inputs				
105 (*)	Input BI7 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC-U-	[1.3] DPT_Enable
106.. 113 (*)	Input BI7 – Switching status *[type]*, object *n* (* only EK-HO1-TP device)	*table A1*	-WCTU-	*table A1 at the end of paragraph*
114 (*)	Input BI7 – Scene number (* only EK-HO1-TP device)	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
BI8 Inputs				
115 (*)	Input BI8 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC-U-	[1.3] DPT_Enable
116.. 123 (*)	Input BI8 – Switching status *[type]*, object *n* (* only EK-HO1-TP device)	*table A1*	-WCTU-	*table A1 at the end of paragraph*
124 (*)	Input BI8 – Scene number (* only EK-HO1-TP device)	1 Byte	--CT--	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
BI9 Inputs				
125 (*)	Input BI9 – Lock command	1 Bit	-WC-U-	[1.3] DPT_Enable

Nr.	Name	Size	Flags	Datapoint type
	(* only EK-HO1-TP device)			
126 (*)	Input BI9 – Windows contact status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.19] DPT_Window_Door
126 (*)	Input BI9 – Generic contact status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
BI10 Inputs				
127 (*)	Input BI10 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC-U-	[1.3] DPT_Enable
128 (*)	Input BI10 – Help request status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
BI11 Inputs				
129 (*)	Input BI11 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC-U-	[1.3] DPT_Enable
130 (*)	Input BI11 – Room cleaning request status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
Sensors from bus				
131	Room temperature 1 (from bus)	2 Bytes	-WC---	[9.1] DPT_Value_Temp
132	Room temperature 2 (from bus)	2 Bytes	-WC---	[9.1] DPT_Value_Temp
133	Outdoor temperature (from bus)	2 Bytes	-WC---	[9.1] DPT_Value_Temp
134	Coil battery temperature (from bus)	2 Bytes	-WC---	[9.1] DPT_Value_Temp
135	Antistratification temperature (from bus)	2 Bytes	-WC---	[9.1] DPT_Value_Temp
136	Flow temperature (from bus)	2 Bytes	-WC---	[9.1] DPT_Value_Temp
137	Drip tray contact (from bus)	1 Bit	-WC---	[1.5] DPT_Alarm
138	Windows contact 1 (from bus)	1 Bit	-WC---	[1.1] DPT_Window_Door
139	Windows contact 2 (from bus)	1 Bit	-WC---	[1.1] DPT_Window_Door
140	Presence sensor 1 (from bus)	1 Bit	-WC---	[1.18] DPT_Occupancy
141	Presence sensor 2 (from bus)	1 Bit	-WC---	[1.18] DPT_Occupancy
142	External badge (from bus)	1 Bit	-WC---	[1.1] DPT_Switch
143	Internal badge (from bus)	1 Bit	-WC---	[1.1] DPT_Switch
[BO1-BO2] Outputs				
144	Output BO1 – On/off command	1 Bit	-WC---	[1.1] DPT_Switch
145	Output BO1 – On/off status	1 Bit	R-CT--	[1.1] DPT_Switch
146	Output BO1 – Lock command	1 Bit	-WC---	[1.3] DPT_Enable
147	Output BO1 – Forcing command	2 Bit	-WC---	[2.1] DPT_Switch_Control
148	Output BO1 – Scene number	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
149	Output BO2 – On/off command	1 Bit	-WC---	[1.1] DPT_Switch
150	Output BO2 – On/off status	1 Bit	R-CT--	[1.1] DPT_Switch
151	Output BO2 – Lock command	1 Bit	-WC---	[1.3] DPT_Enable
152	Output BO2 – Forcing command	2 Bit	-WC---	[2.1] DPT_Switch_Control
153	Output BO2 – Scene number	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
154	Outputs BO1 and BO2 – Move up-down command	1 Bit	-WC---	[1.8] DPT_UpDown [1.9] DPT_OpenClose
155	Outputs BO1 and BO2 – Stop-step up-down command	1 Bit	-WC---	[1.7] DPT_Step

Nr.	Name	Size	Flags	Datapoint type
156	Outputs BO1 and BO2 – Dedicated stop command	1 Bit	-WC---	[1.17] DPT_Trigger
157	Outputs BO1 and BO2 – Info move	1 Bit	R-CT--	[1.8] DPT_UpDown
158	Outputs BO1 and BO2 – Valid current abs position	1 Bit	R-CT--	[1.2] DPT_Bool
159	Outputs BO1 and BO2 – Lock command	1 Bit	-WC---	[1.3] DPT_Enable
160	Outputs BO1 and BO2 – Forcing command	2 Bit	-WC---	[2.1] DPT_Switch_Control
161	Outputs BO1 and BO2 – Scene number	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
162	Outputs BO1 and BO2 – Dimmer position command	4 Bit	-WC---	[3.8] DPT_Control_Blinds
163	Outputs BO1 and BO2 – Abs position command	1 Byte	-WC---	[5.1] DPT_Scaling
164	Outputs BO1 and BO2 – Abs position status	1 Byte	R-CT--	[5.1] DPT_Scaling
165	Outputs BO1 and BO2 – Dimmer slats command	4 Bit	-WC---	[3.8] DPT_Control_Blinds
166	Outputs BO1 and BO2 – Abs slat position command	1 Byte	-WC---	[5.1] DPT_Scaling
167	Outputs BO1 and BO2 – Abs slat position status	1 Byte	R-CT--	[5.1] DPT_Scaling
[BO3-BO4] Outputs				
168	Output BO3 – On/off command	1 Bit	-WC---	[1.1] DPT_Switch
169	Output BO3 – On/off status	1 Bit	R-CT--	[1.1] DPT_Switch
170	Output BO3 – Lock command	1 Bit	-WC---	[1.3] DPT_Enable
171	Output BO3 – Forcing command	2 Bit	-WC---	[2.8] DPT_Direction1_Control
172	Output BO3 – Scene number	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
173	Output BO4 – On/off command	1 Bit	-WC---	[1.1] DPT_Switch
174	Output BO4 – On/off status	1 Bit	R-CT--	[1.1] DPT_Switch
175	Output BO4 – Lock command	1 Bit	-WC---	[1.3] DPT_Enable
176	Output BO4 – Forcing command	2 Bit	-WC---	[2.1] DPT_Switch_Control
177	Output BO4 – Scene number	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
178	Outputs BO3 and BO4 – Move up-down command	1 Bit	-WC---	[1.8] DPT_UpDown [1.9] DPT_OpenClose
179	Outputs BO3 and BO4 – Stop-step up-down command	1 Bit	-WC---	[1.7] DPT_Step
180	Outputs BO3 and BO4 – Dedicated stop command	1 Bit	-WC---	[1.17] DPT_Trigger
181	Outputs BO3 and BO4 – Info move	1 Bit	R-CT--	[1.8] DPT_UpDown
182	Outputs BO3 and BO4 – Valid current abs position	1 Bit	R-CT--	[1.2] DPT_Bool
183	Outputs BO3 and BO4 – Lock command	1 Bit	-WC---	[1.3] DPT_Enable
184	Outputs BO3 and BO4 – Forcing command	2 Bit	-WC---	[2.1] DPT_Switch_Control
185	Outputs BO3 and BO4 – Scene number	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl

Nr.	Name	Size	Flags	Datapoint type
186	Outputs BO3 and BO4 – Dimmer position command	4 Bit	-WC---	[3.8] DPT_Control_Blinds
187	Outputs BO3 and BO4 – Abs position command	1 Byte	-WC---	[5.1] DPT_Scaling
188	Outputs BO3 and BO4 – Abs position status	1 Byte	R-CT--	[5.1] DPT_Scaling
189	Outputs BO3 and BO4 – Dimmer slats command	4 Bit	-WC---	[3.8] DPT_Control_Blinds
190	Outputs BO3 and BO4 – Abs slat position command	1 Byte	-WC---	[5.1] DPT_Scaling
191	Outputs BO3 and BO4 – Abs slat position status	1 Byte	R-CT--	[5.1] DPT_Scaling
[BO5] Output				
192 (*)	Output BO5 – On/off command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.1] DPT_Switch
193 (*)	Output BO5 – On/off status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
194 (*)	Output BO5 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.3] DPT_Enable
195 (*)	Output BO5 – Forcing command (* only EK-HO1-TP device)	2 Bit	-WC---	[2.1] DPT_Switch_Control
196 (*)	Output BO5 – Scene number (* only EK-HO1-TP device)	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
[BO6] Output				
197 (*)	Output BO6 – On/off command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.1] DPT_Switch
198 (*)	Output BO6 – On/off status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
199 (*)	Output BO6 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.3] DPT_Enable
200 (*)	Output BO6 – Forcing command (* only EK-HO1-TP device)	2 Bit	-WC---	[2.1] DPT_Switch_Control
201 (*)	Output BO6 – Scene number (* only EK-HO1-TP device)	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
[BO7] Output				
202 (*)	Output BO7 – On/off command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.1] DPT_Switch
203 (*)	Output BO7 – On/off status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
204 (*)	Output BO7 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.3] DPT_Enable
205 (*)	Output BO7 – Forcing command (* only EK-HO1-TP device)	2 Bit	-WC---	[2.1] DPT_Switch_Control
206 (*)	Output BO7 – Scene number (* only EK-HO1-TP device)	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
[BO8] Output				
207 (*)	Output BO8 – On/off command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.1] DPT_Switch
208 (*)	Output BO8 – On/off status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
209 (*)	Output BO8 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.3] DPT_Enable
210 (*)	Output BO8 – Forcing command (* only EK-HO1-TP device)	2 Bit	-WC---	[2.1] DPT_Switch_Control

Nr.	Name	Size	Flags	Datapoint type
211 (*)	Output BO8 – Scene number (* only EK-HO1-TP device)	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
[BO9] Output				
212 (*)	Output BO9 – On/off command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.1] DPT_Switch
213 (*)	Output BO9 – On/off status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
214 (*)	Output BO9 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.3] DPT_Enable
215 (*)	Output BO9 – Forcing command (* only EK-HO1-TP device)	2 Bit	-WC---	[2.1] DPT_Switch_Control
216 (*)	Output BO9 – Scene number (* only EK-HO1-TP device)	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
[BO10] Output				
217 (*)	Output BO10 – On/off command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.1] DPT_Switch
218 (*)	Output BO10 – On/off status (* only EK-HO1-TP device)	1 Bit	R-CT--	[1.1] DPT_Switch
219 (*)	Output BO10 – Lock command (* only EK-HO1-TP device)	1 Bit	-WC---	[1.3] DPT_Enable
220 (*)	Output BO10 – Forcing command (* only EK-HO1-TP device)	2 Bit	-WC---	[2.1] DPT_Switch_Control
221 (*)	Output BO10 – Scene number (* only EK-HO1-TP device)	1 Byte	-WC---	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl
222	Door Lock – Pulse command	1 Bit	-WC---	[1.1] DPT_Switch
223	Door Lock – Pulse status	1 Bit	R-CT--	[1.1] DPT_Switch
Actuator commands				
224	Room temperature controller status	1 Bit	R-CT--	[1.3] DPT_Enable
225	Disable room temperature controller	1 Bit	-WC---	[1.1] DPT_Switch
226	Fan continuous speed command	1 Byte	-WC---	[5.1] DPT_Scaling
226	Fan continuous speed command counter	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
226	First fan speed command	1 Bit	-WC---	[1.1] DPT_Switch
227	Second fan speed command	1 Bit	-WC---	[1.1] DPT_Switch
227	Fan continuous speed manual counter	1 Byte	-WC---	[5.10] DPT_Value_1_Ucount
228	Third fan speed command	1 Bit	-WC---	[1.1] DPT_Switch
229	Heating valve command (on/off)	1 Bit	-WC---	[1.1] DPT_Switch
229	Heating/cooling valve command (on/off)	1 Bit	-WC---	[1.1] DPT_Switch
230	Cooling valve command (on/off)	1 Bit	-WC---	[1.1] DPT_Switch
231	Heating/cooling changeover command	1 Bit	-WC---	[1.100] DPT_Heat_Cool
240	Heating/cooling status out	1 Bit	R-CT--	[1.100] DPT_Heat_Cool
States				
233	Fan stopped status	1 Bit	R-CT--	[1.1] DPT_Switch
234	First fan speed status	1 Bit	R-CT--	[1.1] DPT_Switch
235	Second fan speed status	1 Bit	R-CT--	[1.1] DPT_Switch
236	Third fan speed status	1 Bit	R-CT--	[1.1] DPT_Switch
237	Counter speed fan status	1 Byte	R-CT--	[5.10] DPT_Value_1_Ucount

Nr.	Name	Size	Flags	Datapoint type
238	Heating valve status	1 Bit	R-CT--	[1.1] DPT_Switch
238	Heating/cooling valve status	1 Bit	R-CT--	[1.1] DPT_Switch
239	Cooling valve status	1 Bit	R-CT--	[1.1] DPT_Switch
240	Heating/cooling status out	1 Bit	R-CT--	[1.100] DPT_Heat_Cool
241	Heating/cooling changeover command	1 Bit	-WC---	[1.100] DPT_Heat_Cool
HVAC Operative modes				
242	HVAC mode IN	1 Byte	-WC---	[20.102] DPT_HVACMode
243	HVAC manual mode	1 Byte	-WC---	[20.102] DPT_HVACMode
244	HVAC chrono mode active status	1 Bit	RWCTU-	[1.11] DPT_State
245	HVAC mode OUT	1 Byte	R-CT--	[20.102] DPT_HVACMode
Internal regulator				
246	Actual setpoint	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
247	Input setpoint	2 Bytes	-WC---	[9.1] DPT_Value_Temp
248	Comfort setpoint (heating)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
249	Comfort setpoint (cooling)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
250	Standby setpoint (heating)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
250	Standby offset (heating)	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd
251	Standby setpoint (cooling)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
251	Standby offset (cooling)	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd
252	Economy setpoint (heating)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
252	Economy offset (heating)	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd
253	Economy setpoint (cooling)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
253	Economy offset (cooling)	2 Bytes	RWCTU-	[9.2] DPT_Value_Tempd
254	Building protection (heating)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
255	Building protection (cooling)	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
256	Manual/forced setpoint active status	1 Bit	RWCTU-	[1.11] DPT_State
257	Manual setpoint	2 Bytes	RWCTU-	[9.1] DPT_Value_Temp
258	Cooling compensation offset	2 Bytes	R-CT--	[9.2] DPT_Value_Tempd
259	Temperature controller output	1 Byte	R-CT--	[5.1] DPT_Scaling
260	Fan working time (hours)	2 Bytes	R-CT--	[7.7] DPT_Time_Hours
261	Fan working time (seconds)	4 Bytes	R-CT--	[13.100] DPT_LongDeltaTimeSec
Disable output from bus				
262	Fan disable from bus	1 Bit	-WC---	[1.3] DPT_Enable
263	Fan manual/auto status	1 Bit	RWCTU-	[1.3] DPT_Enable
264	Heating valve disable from bus	1 Bit	-WC---	[1.3] DPT_Enable
264	Heating/cooling valve disable from bus	1 Bit	-WC---	[1.3] DPT_Enable
265	Cooling valve disable from bus	1 Bit	-WC---	[1.3] DPT_Enable
266	Controller output forced from bus	1 Byte	-WC---	[5.1] DPT_Scaling
267	Controller output automatic/forced from bus	1 Bit	-WC---	[1.3] DPT_Enable
Output from bus				

Nr.	Name	Size	Flags	Datapoint type
268	Output V1 from bus	1 Bit	-WC---	[1.1] DPT_Switch
269	Output V2 from bus	1 Bit	-WC---	[1.1] DPT_Switch
270	Output V3 from bus	1 Bit	-WC---	[1.1] DPT_Switch
271	Output heating valve from bus	1 Bit	-WC---	[1.1] DPT_Switch
272	Output cooling valve from bus	1 Bit	-WC---	[1.1] DPT_Switch
273	Output 0-10V from bus	1 Byte	-WC---	[5.1] DPT_Scaling
Additional features				
274	Fan speed limit enable	1 Bit	-WC---	[1.3] DPT_Enable
275	Weighted room temperature	2 Bytes	R-CT--	[9.1] DPT_Value_Temp
Logic functions				
276, 281, 286, 291	Logic function X – Input 1	1 Bit	-WC---	[1.1] DPT_Switch
277, 282, 287, 292	Logic function X – Input 2	1 Bit	-WC---	[1.1] DPT_Switch
278, 283, 288, 293	Logic function X – Input 3	1 Bit	-WC---	[1.1] DPT_Switch
279, 284, 289, 294	Logic function X – Input 4	1 Bit	-WC---	[1.1] DPT_Switch
280, 285, 290, 295	Logic function X – Output	1 Bit	R-CT--	[1.1] DPT_Switch

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

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

12.2 Diagnostic

The last active alarm in the device can be consulted with a textual description through the communication object Text Alarms (OC 10): when an alarm occurs the string "ALARM Exx" is sent, when the alarm is returned the string is sent " NO ALARM Exx ". Below is the list with the alarm code xx and the description of the alarms managed.

Error code	Description
Alarm Code 01	Superficial temperature overcome
Alarm Code 02	Condensation
Alarm Code 03	Thermal generator lock
Error Code 06	Analog input 1: generic NTC failure
Error Code 07	Analog input 1: added temperature sensor failure
Error Code 08	Analog input 1: fan-coil temperature sensor failure
Error Code 10	Analog input 1: outdoor temperature sensor failure
Error Code 11	Analog input 1: antistratification temperature sensor failure
Error Code 14	Analog input 2: generic NTC failure
Error Code 15	Analog input 2: added temperature sensor failure
Error Code 16	Analog input 2: fan-coil temperature sensor failure
Error Code 18	Analog input 2: outdoor temperature sensor failure
Error Code 19	Analog input 2: antistratification temperature sensor failure
Error Code 22	Analog input 3: generic NTC failure
Error Code 23	Analog input 3: added temperature sensor failure
Error Code 24	Analog input 3: fan-coil temperature sensor failure
Error Code 26	Analog input 3: outdoor temperature sensor failure
Error Code 27	Analog input 3: antistratification temperature sensor failure
Error Code 30	Air Quality from bus failure
Error Code 31	Outdoor temperature from bus failure
Error Code 32	Added temperature from bus failure
Error Code 33	Fan-coil temperature from bus failure
Error Code 34	Superficial temperature from bus failure
Error Code 35	Flow temperature from bus failure
Error Code 36	Relative humidity from bus failure
Error Code 37	Antistratification temperature from bus failure
Error Code 41	Air Quality from bus timeout
Error Code 42	Outdoor temperature from bus timeout
Error Code 43	Added temperature from bus timeout
Error Code 44	Fan-coil temperature from bus timeout
Error Code 45	Superficial temperature from bus timeout
Error Code 46	Flow temperature from bus timeout
Error Code 47	Relative humidity from bus timeout
Error Code 48	Antistratification temperature from timeout
Error Code 49	Anti-condensation sensor from bus timeout
Error Code 50	Windows contact 1 from bus timeout
Error Code 51	Windows contact 2 from bus timeout
Error Code 52	Presence sensor 1 from bus timeout
Error Code 53	Presence sensor 2 from bus timeout
Error Code 54	Card holder from bus timeout
Error Code 56	Drip tray level sensor from bus timeout
Error Code 57	External regulator timeout
Error Code 58	Internal badge from bus timeout
Error Code 59	External badge from bus timeout

Table of error codes and alarms that can be consulted

13 Warnings

- Installation, electrical connection, configuration and commissioning of the device can only be carried outby qualified personnel in compliance with the applicable technical standards and laws of the respective countries
- Opening the housing of the device causes the immediate end of the warranty period
- In case of tampering, the compliance with the essential requirements of the applicable directives, for which the device has been certified, is no longer guaranteed
- ekinex® KNX defective devices must be returned to the manufacturer at the following address:

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

14 Other information

- The instruction sheet must be delivered to the end customer with the project documentation
- For further information on the product, please contactthe ekinex® technical support at the e-mail address: support@ekinex.com or visit the website www.ekinex.com
- Each ekinex® device has a unique serial number onthe label. The serial number can be used by installersor system integrators for documentation purposes and has to be added in each communication addressed to the Ekinex technical support in case of malfunctioning of the device
- ekinex® is a registered trademark of Ekinex S.p.A.
- KNX® and ETS® are registered trademarks of KNX Association cvba, Brussels