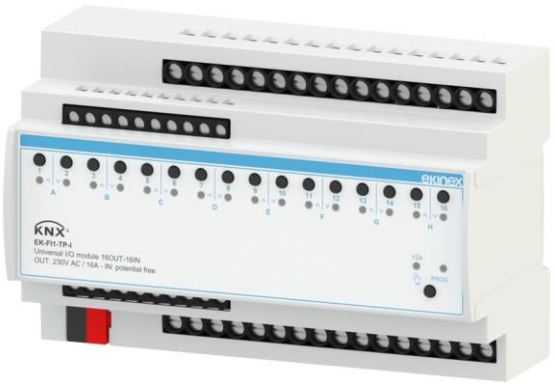
ekinex

CONTROL YOUR LIVING SPACE

Application manual



KNX 8/16-output 8/16-input universal I/O modules ÆVO Series

EK-FG1-TP-I 8 output / 8 input

EK-FI1-TP-I 16 output 16 input

Contents

Release	Update	Date	Author	Revised by
1.0	First issue	27/10/2025	G. Schiochet	V. Cappelli

1 Scope of the document

This manual describes the application details for version 1.0 of the ekinex® EK-FG1-TP-I universal actuator module with 8 outputs (4/8 channels) and 8 inputs and the ekinex® EK-FI1-TP-I universal actuator module with 16 outputs (8/16 channels) and 16 inputs.
The document is intended for system configurators as a description and reference guide for device functionality and application programming. For mechanical and electrical details of the installation device, please refer to the device datasheet.
This application manual and the application programs for the ETS development environment are available for download at www.ekinex.com.

Item	*File name (## = release)*	*Version*	*Device rel.*	*Update*
Technical datasheet	STEKFG1TPI_EN.pdf STEKFI1TPI_EN.pdf	1.0 and later	1.0	10/2025
Application manual	MAEKFG1FI1TPI_EN.pdf	1.0 and later		
Application program	APEKFG1TPI##.knxprod APEKFI1TPI##.knxprod	0.1 and later		

2 Product description

The ekinex® EK-FG1-TP-I actuator module with 8 outputs (4/8 channels) and 8 inputs is a modular DIN-rail device for panel mounting, allowing the independent switching of 4 or 8 electrical loads, respectively. It also includes 8 separate digital/analog inputs, each configurable as a binary input or an NTC temperature sensor.

Similarly, the ekinex® EK-FI1-TP-I actuator module with 16 outputs (8/16 channels) and 16 inputs is a modular DIN-rail device for panel mounting, allowing the independent switching of 8 or 16 electrical loads, respectively. It also includes 16 separate digital inputs, each configurable as a binary input or, in the case of digital/analog inputs 9 to 16, also as an NTC temperature sensor.

The outputs consist of relay contacts.

Each device is equipped with an integrated KNX bus interface module and is designed for mounting on a standard DIN rail inside electrical panels.

During operation, the module receives communication telegrams from the KNX bus sent by another device (e.g., a manual control point, a sensor, a timer, etc.). These telegrams activate or deactivate the outputs, applying a series of programmable utility functions.

Manual control of the outputs is also possible using the membrane buttons on the front of the unit; LED indicators allow you to check the status of the outputs.

The device draws its power exclusively from the KNX bus line with a SELV voltage of 30 VDC. Therefore, no auxiliary power supply is required.

For further technical information, please also refer to the product datasheet STEKFG1TPI_EN.pdf or STEKFI1TPI_EN.pdf available on the ekinex website www.ekinex.com.

2.1 Technical data

2.1.1 Supply

- Power supply 21-30 Vdc via KNX bus
- Bus current draw at start-up: 54 mA @30 Vdc, 49 mA @21 Vdc
- Bus current draw in stand-by: 10 mA @30 Vdc, 13 mA @21 Vdc
- Bus current draw during relay movement: 32 mA @21-30 Vdc

2.1.2 Outputs

- EK-FG1-TP-I: 8 independent outputs, 4 combined in channels (depending on use)
- EK-FI1-TP-I: 16 independent outputs, 8 combined in channels (depending on use)
- Bistable relays with tungsten pre-contact
- Rated current (I_n) per output: AC 16(6) A @250 Vac (4000 VA); DC 7 A @30 Vdc (210 W)
- Maximum load per output: resistive 4000 W, inductive 1500 VA, LED lamps 90-230 Vac max. 400 W

- Maximum inrush current: 800 A / 200 µs, 165 A / 20 ms
- Maximum capacitive load: 200 µF
- Possibility of connecting different phases in adjacent outputs
- Total maximum current in device: 96 A
- Short-circuit protection: not present
- Overload protection: not present
- Connection method: screw terminal block (0.5 Nm torque max.)
- Cable cross-section: 4 mm² / 2 x 2,5 mm² (0,5 mm² min.)
- Outputs per common: 1

2.1.3 Inputs

- EK-FG1-TP-I / EK-FI1-TP-I: 8/16 independent digital inputs [DI] for potential-free contacts, e.g., for connecting traditional switches or buttons for on/off control of appliances.
- 8 inputs individually configurable as analog [AI] for measuring room temperature using a passive temperature probe (NTC 10 kΩ at 25°C) to be connected to the input with the option of sending the value to the bus and thermostat functionality.
- EK-FG1-TP-I / EK-FI1-TP-I: 4/8 independent channels (4/8 coupled inputs), e.g., for connecting double buttons to control dimmers or motorized drives.
- On/off control of individual and group appliances.
- Detection of the status of signal contacts (from safety devices, alarms, etc.).
- Scene recall and storage.
- Sending values (temperature, brightness, etc.) to the bus.
- Lock and shift register function

2.1.4 Environmental conditions

- Operating temperature: - 5 ... + 45°C
- Storage temperature: - 25 ... + 55°C
- Transport temperature: - 25 ... + 70°C
- Relative humidity: 95% not condensing)

2.1.5 Other characteristics

- Housing in plastic material
- Mounting on 35 mm rail (according to EN 60715)
- Protection degree IP20 (installed device)
- Overvoltage class III (according to EN 60664-1)
- Classification climatic 3K5 and mechanical 3M2 (according to EN 50491-2)

- Pollution degree 2 (according to IEC 60664-1)
- Mechanical lifetime (min. cycles): 3 000 000
- Electrical lifetime (min. cycles): 100000 @8A / 25000 @16 A (Vac)
- Dimensions (WxHxD): 71 x 90 x 63 mm (for EK-FG1-TP-I), 142 x 90 x 63 mm (for EK-FI1-TP-I)
- Weight: 277 g (for EK-FG1-TP-I), 505 g (for EK-FI1-TP-I)

3 Switching, display and connection elements

The device is equipped with:

- membrane buttons for manual activation of the outputs;
- a button to switch the device to programming mode (short press) or to switch between manual mode or control line of the outputs (long press > 2s);
- LED indicators for the status of the outputs and for the manual mode;
- LED indicator for the programming mode;
- screw terminal blocks for connecting the output loads;
- screw terminal blocks for connecting the inputs;
- plug-in terminal block for connecting the KNX bus line.

The terminals of the outputs that can be used as coupled are located side by side on the upper or lower terminal block; the corresponding outputs are marked with letters from A to D (corresponding to the channels) and numbers from 1 to 8 for EK-FG1-TP-I, or with letters from A to H and numbers from 1 to 16 for EK-FI1-TP-I. The up/down arrows are intended to recall the typical function for roller shutters/blinds (raise/lower).

The terminals of the inputs that can be used in coupled mode are located side by side on the central terminal blocks; the corresponding inputs are marked with numbers 1 to 8 for EK-FG1-TP-I, or with numbers 1 to 16 for EK-FI1-TP-I. In ETS, the coupled inputs are marked with letters A to D for EK-FG1-TP-I and A to H for EK-FI1-TP-I.

Further details on this can be found in the following chapters.

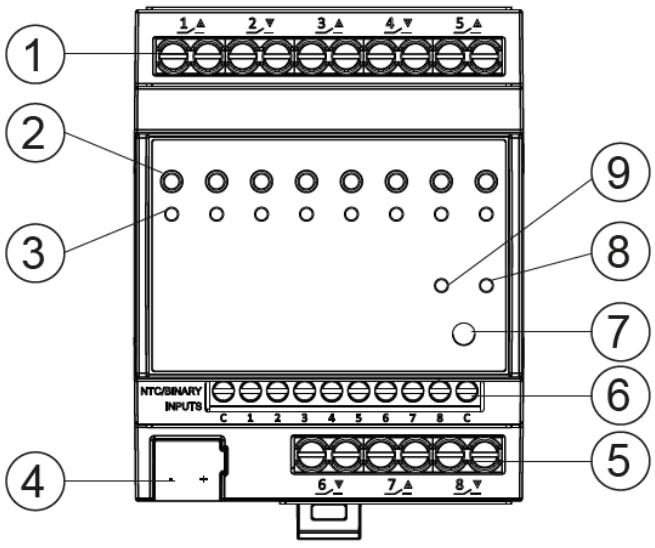


Figure 1 - Switching, display and connection elements for EK-FG1-TP-I

1. Connection terminals for outputs A-1 to C-5	6. Connection terminals for inputs A-1 to D-8 / NTC probes

2. Buttons for forced operation of the outputs 3. Green LEDs for indicating the status of the outputs 4. Connection terminal for KNX bus line 5. Connection terminals for outputs C-6 to D-8	6. Button for switching between forced and automatic operation of the outputs / KNX programming 7. Red LED for indicating KNX programming mode 8. Green LED for indicating the operating mode (on = forced operation, off = automatic operation)

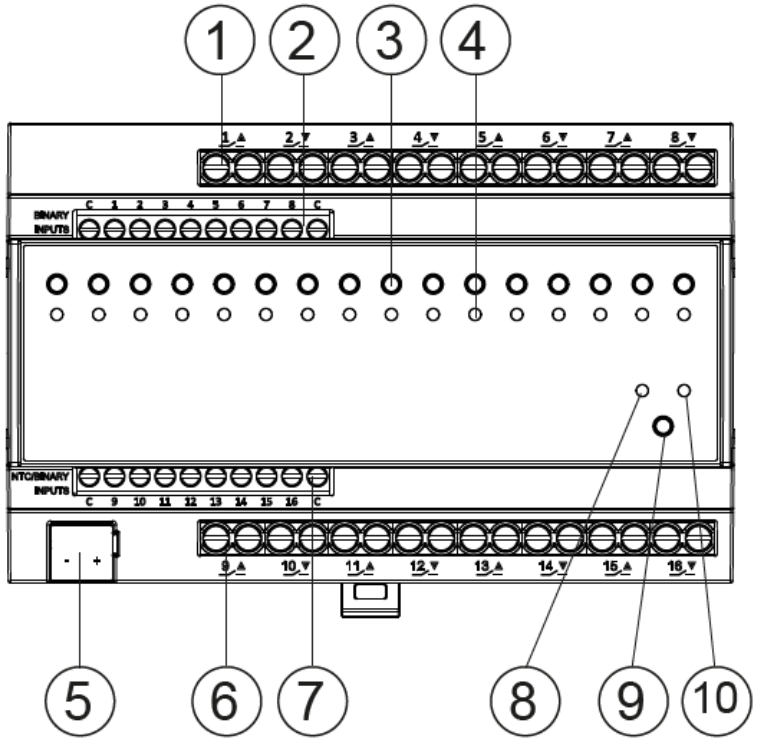


Figure 2 - Switching, display and connection elements for EK-FI1-TP-I

1. Connection terminals for outputs A-1 to D-8 2. Morsetti di collegamento ingressi da A-1 a D-8 3. Buttons for forced operation of the outputs 4. Green LEDs for indicating the status of the outputs 5. Connection terminal for KNX bus line	6. Connection terminals for outputs E-9 to H-16 7. Connection terminals for inputs E-9 to H-16 / NTC probes 8. Button for switching between forced and automatic operation of the outputs / KNX programming 9. Red LED for indicating KNX programming mode 10. Green LED for indicating the operating mode (on = forced operation, off = automatic operation)

4 Configuration

The functionality of the device on the settings done via software.

In order to configure the device, the ETS5 development tool (or later versions) and the dedicated application program for the device, called **APEKFG1TP##.knxprod** for EK-FG1-TP-I or **APEKFI1TP##.knxprod** for EK-FI1-TP-I, are required; the latter can be downloaded from the ekinex website www.ekinex.com.

The application program allows access, within the ETS environment, to the configuration of all the working parameters of the device. The program must be loaded into ETS (alternatively, it is possible to load the entire ekinex® product database in a single operation), after which all the device specimens of the considered type can be added to the project being defined.

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

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

Product code	No. of channels/outputs	No. of digital inputs/NTC	ETS application software (## = release)	Communication objects (max nr.)	Group adresses (max nr.)
EK-FG1-TP-I	4 / 8	8 / 8	APEKFG1TP##.knxprod	1057	254
EK-FI1-TP-I	8 / 16	16 / 8	APEKFI1TP##.knxprod	1337	254

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

For further information: www.knx.org

5 Commissioning

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

- electrically connect the device, as described in the technical data sheet, to the KNX bus in the final destination system or in a reduced system, composed specifically for programming. The system will in any case contain an interface device to the PC on which the KNX environment is installed;
- apply power to the bus;
- activate the programming mode on the device by pressing the appropriate button located on the front. The programming mode indicator LED must light up with a fixed light;
- from the ETS environment, start programming (which in the case of first configuration must include the physical address to be given to the device).

Once the program has been downloaded, the device automatically returns to operational mode; the programming LED must be off. The device is now programmed and ready for operation in the system.

5.1 Device reset

To reset the device, press and hold the programming button for at least 10 seconds, until the programming LED flashes quickly: the reset is complete. You will now need to re-address and re-configure the device via ETS.

Warning! The reset restores the device to the factory delivery state. The addressing and the value of the parameters set in the configuration phase are lost.

6 Function description

The device works as a controlled switch, which activates its outputs according to the commands received from the bus in the form of KNX telegrams.

In addition to direct activation, it also has auxiliary functions such as timing functions and logical combination of inputs. These functions are described in detail in the following paragraphs.

The logic outputs are of the binary (or digital) type, that is, they can only assume the two values “On” and “Off”; each output is equipped with a bistable relay. Inputs can be digital or analog for NTC probe.

6.1 Start-up

When the bus is connected, the device enters a fully active state after a short period (of the order of tens of ms) necessary for reinitialization. It is possible to define a longer additional delay to avoid overloading the bus during the system start-up phase.

At this point the device is ready for operation.

To operate, the device receives a telegram from the bus, sent by a KNX sensor or another KNX control device, which determines the opening or closing of one or more relays. Each output uses a bistable relay and can also be controlled either via the KNX bus or manually via buttons on the front of the device. There are also LEDs on the front that indicate the status of each output. The device requires only power from the KNX bus at 21-30Vdc.

6.2 Resetting outputs

In any mode, it is possible to define the behavior of the device following some relevant events. In particular, these events are:

- Bus On, i.e. the restoration of the KNX bus;
- downloading a new configuration from ETS.

Further events are then given by the cessation of particular functions, such as the block function or the forcing of the outputs.

For each of these events, the state of the outputs (or pairs of outputs) can be defined by configuration among a set of values depending on how each output is configured. These sets of values will be indicated later in the paragraphs describing the relative functions.

It should be noted that in all the cases listed above it is assumed that the voltage is present; otherwise, even if the internal state of the outputs is the expected one, the physical switching of the output relays cannot take place.

6.3 Manual operation

Manual operation is an alternative possibility to switching inputs via bus commands; this mode is intended for test or maintenance situations.

6.3.1 Output status when changing mode

When manual mode is activated, the status of the outputs is not altered. When manual mode is active, telegrams from the bus do not affect the physical outputs; the output contacts can only be switched via the membrane keys on the front.

Manual activation / deactivation of the outputs does not cause any status feedback telegram to be generated on the bus. The LEDs associated with the outputs will continue to indicate their status in any case.

Even when returning online from manual mode, the status of the outputs remains the one currently set.

From another point of view, the situation could be illustrated by saying that while remaining in manual mode, it is as if the internal variables were temporarily "disconnected" from the group addresses. When "reconnected" (exiting manual) their value remains unchanged until a new bus command alters it.

The same considerations made for bus commands apply to switching due to internal timing functions (for example, activation delays or stair light function): status changes due to internal functions have no effect as long as manual mode remains active.

6.3.2 Activating manual mode

If manual operation has been enabled by ETS, to switch to manual operation proceed as follows:

1) Press the manual mode button on the front of the device for at least 2 seconds. In normal operation, the LED is off; when the LED lights up green, the forced operation buttons are active, and manual mode is activated.

2) Press the keypad button corresponding to the channel to be activated (in the example: A-1). Repeated pressing toggles between On and Off.

3) When the need is over, deactivate manual mode by pressing the mode change button again for at least 2 seconds. When switching to normal mode, the indicator LED turns off.

Switching to manual mode via the front panel can be prevented in two ways, both configurable:

- by completely disabling the manual operation functionality from the ETS interface;
- by a bus command.

It should be noted that the bus command just mentioned prevents the mode change via the appropriate button, but does not allow the mode to be changed.

If manual mode is neither inhibited by configuration nor defined as bus controllable, another parameter can be used to set a timeout period (*Restore automatic mode*) after which, if the device is left in manual mode, it is returned to online mode. This prevents the device from being left in an uncontrollable state by mistake.

6.4 Online operation

All features described below assume the device has been correspondingly programmed by means of the ETS tool. A fully unprogrammed device causes no activity on the bus; it can be switched to manual mode and operated through the membrane keys on the front panel.

6.4.1 Software working cycle

The software working cycle can be described as follows:

- Handle incoming telegrams from the KNX bus to update internal state variables
- Implement timing functions and other inbuilt functions to determine effect on physical outputs;
- Drive output relays outputs according to output status
- Handle the key presses from the membrane key on the front.
- Respond to bus messages requesting feedback on the status of the outputs and of the device.

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

6.4.2 State variables (Communication objects)

The determination of the status of physical outputs or i puts is made basing on internal state variables. These state variables, once assigned a group address, are actually KNX communication objects, which allows other devices on the bus to exploit the features of the device.

State variables undergo the usual rules for communication objects, among which – for instance – the effect of flags to determine how the change of value affects the transmission of the objects.

6.4.3 Output independent mode and coupling

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

Due to the nature of the functions this device most frequently performs, the outputs can be grouped in pairs. In this case, each channel is made of a pair of outputs which are physically close on the terminal block.

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

Since the channels are identified by the letters A,B,C,D (for EK-FG1-TP-I) or A,B,C,D,E,F,G,H (for EK-FI1-TP-I) and the outputs with the numbers 1 to 8 (for EK-FG1-TP-I) or 1 to 16 (for EK-FI1-TP-I), the outputs are indicated as 1, 2 for channel A, 3, 4 for channel B and so on. For uniformity, the same naming is used even if the outputs are used as independent.

In order to specify channel pairings, each output can be configured in two modes: independent (or single) and coupled.

- In *independent or single mode*, each of the outputs operates independently and has its own parameters and communication objects. This is the mode described so far.
- In *3-way valve/Shutter/Venetian blind* mode, two outputs are grouped under the same channel for a common functionality; consequently, these inputs operate on shared communication objects. Each

output has the possibility of coupling only with the adjacent output, hence the coupling possibilities 1 2, 3 with 4 etc.

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

6.4.4 Output features in independent mode

In the simplest application, a single communication object per channel, “On/Off Command”, is sufficient, which switches the channel directly based on the value received via a KNX telegram.

By configuring the device parameters, 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:

- relay operation (output inversion): allows you to associate the closed (physical) contact position with the logical “On” state and vice versa;
- behavior with bus ON: defines the value of the outputs when the KNX bus is powered;
- status feedback telegram: automatically transmits the status information upon switching;
- behavior after download: defines the value of the outputs after downloading the ETS application;
- delay after bus recovery: defines the time within which the device is operational, after the bus on command;
- valve function: to enable or disable it;
- delay at power-on: to set a delay in the transition of the output to the on state;
- delay in switching off: to set a delay in switching the output to the off state;
- staircase light function: manages a restartable and resettable timer on the output;
- lock and forced operation functions: these functions allow you to inhibit the operation of the output or force its value in different ways;
- scenario function: allows you to recall or set a default value for the output in association with scenario codes;
- energy counter / operating time: allows an approximate count of the energy consumed by accumulating the activity time of an output.The most significant functional blocks for an output in independent operation are described in the following scheme.

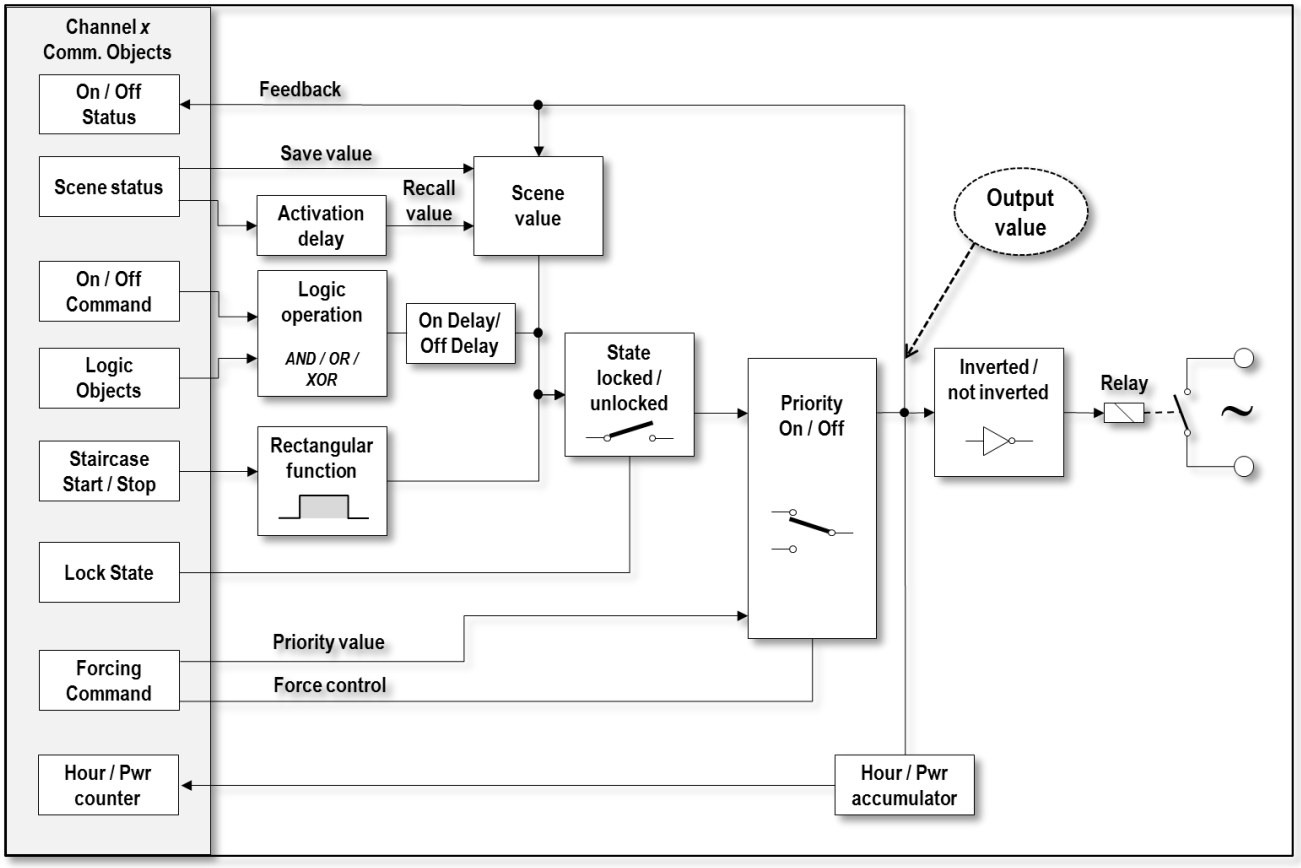


Figure 3 - Functional blocks – Independent mode (referred to a single output)

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

6.4.4.1 Relay operation

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

Note: Regardless of this setting, in the remainder of this manual "On" and "Off" will always refer to the state of the logic output, not the state of the relay output contact.

6.4.4.2 Status feedback telegram

When feedback is enabled, a communication object corresponding to the output state is made available for reading by other devices on the bus. This object reports the actual state of the logical output, which is likely to be different from the state set by the command as it includes the effect of any other functions that are currently active.

When this communication object is defined, it is automatically transmitted at each change in state, so that events can be generated at each actual change in the output.

Feedback telegrams are **not** transmitted, however, if the outputs are activated manually.

6.4.4.3 ON/OFF delay time

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

These delays apply to switching via direct command and/or logical objects, but not to those caused by other functions (e.g. staircase light or scene function).

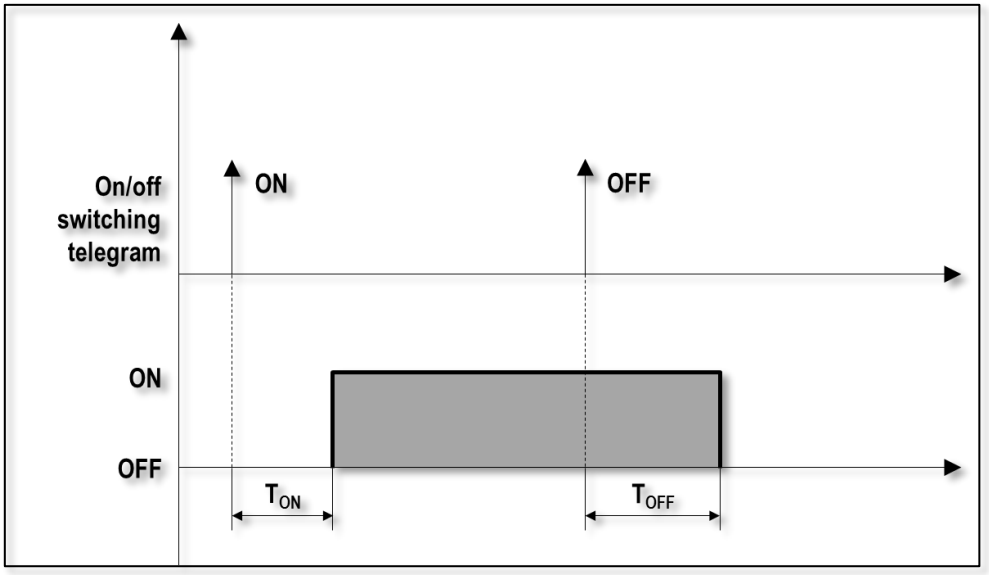


Figure 4 – ON/OFF Time delay

6.4.4.4 Staircase function

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

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

- The warning time must be shorter than the staircase light time ($T_{P-W} < T_S$) and the interruption time must be shorter than the warning time ($T_I < T_{P-W}$).
- The set on / off delay times have no influence on the staircase light function.
- A running timing will be terminated by a reset of the device (bus voltage drop and recovery or reprogramming from ETS) or by using any function that influences the output (e.g. direct command, forced command, logic function, scene recall), even if the on / off value of the output is not changed by the function used.
- In the event of forced termination of the timing, the value of the output remains the one active at the time of termination; this also applies if termination occurs during the warning time.

The following pictures show the *Manual Off* feature:

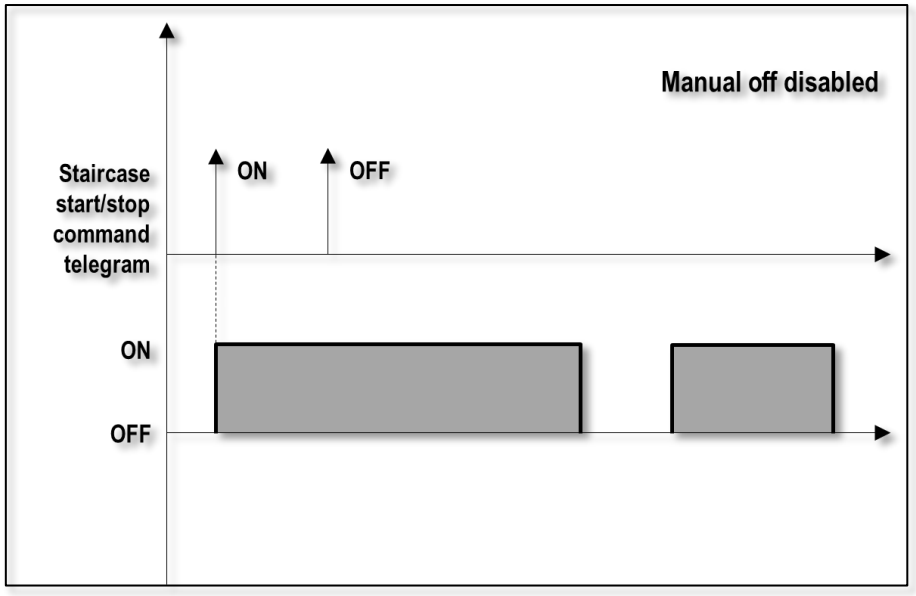


Figure 5 - Manual Off feature (enabled)

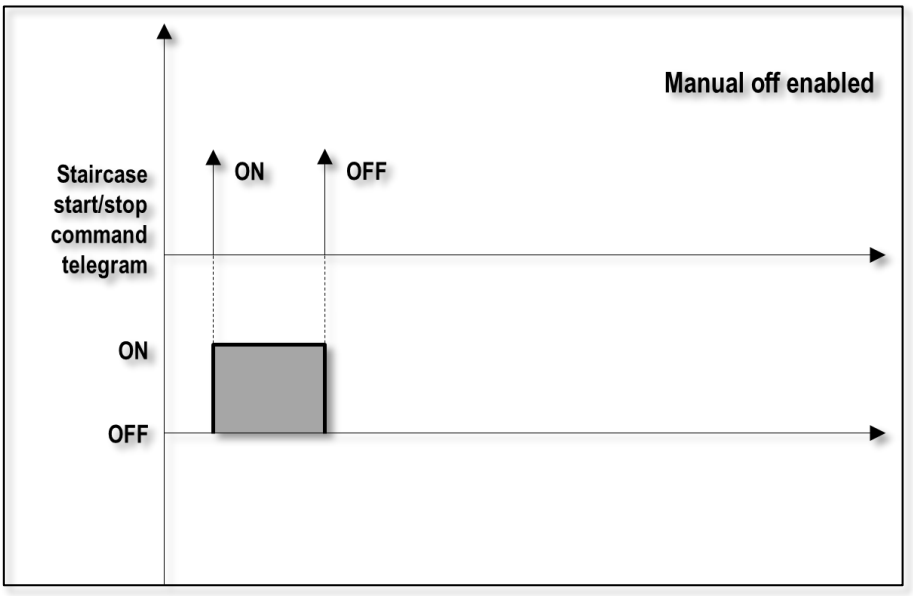


Figure 6 - Manual Off feature (disabled)

Following pictures show the *Retrigger* feature:

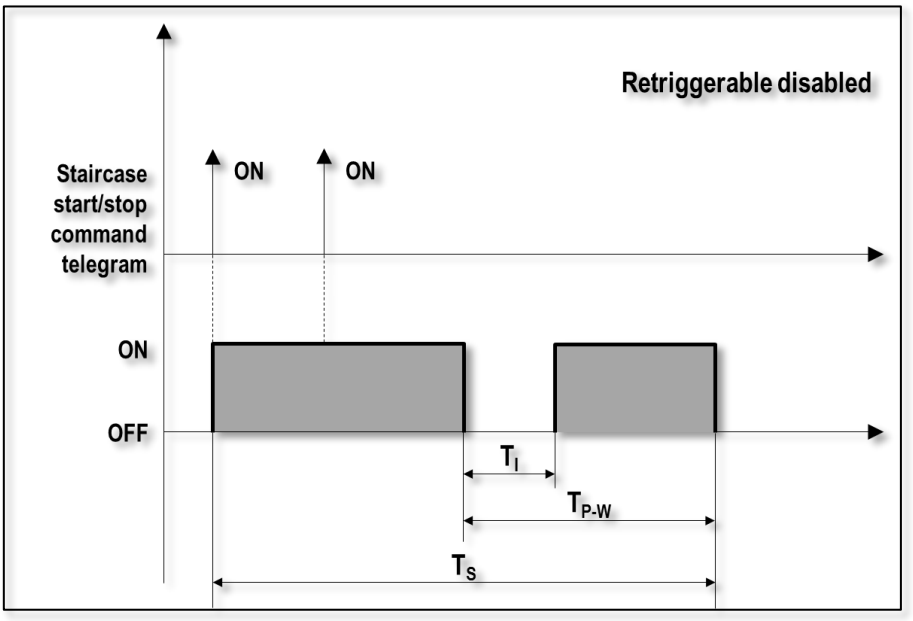


Figure 7 - Retrigger feature (disabled)

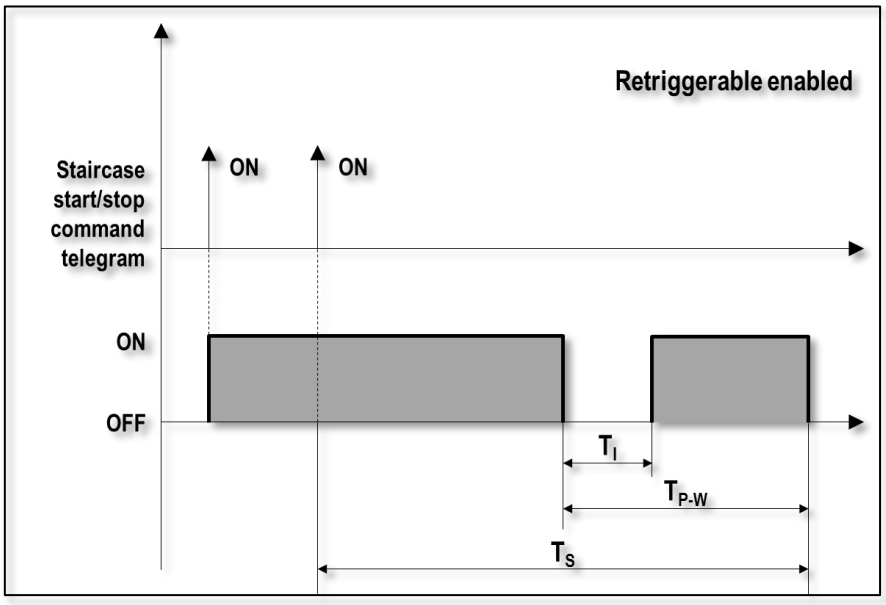


Figure 8 - Retrigger feature (enabled)

Following pictures show the *Pre-warning* feature:

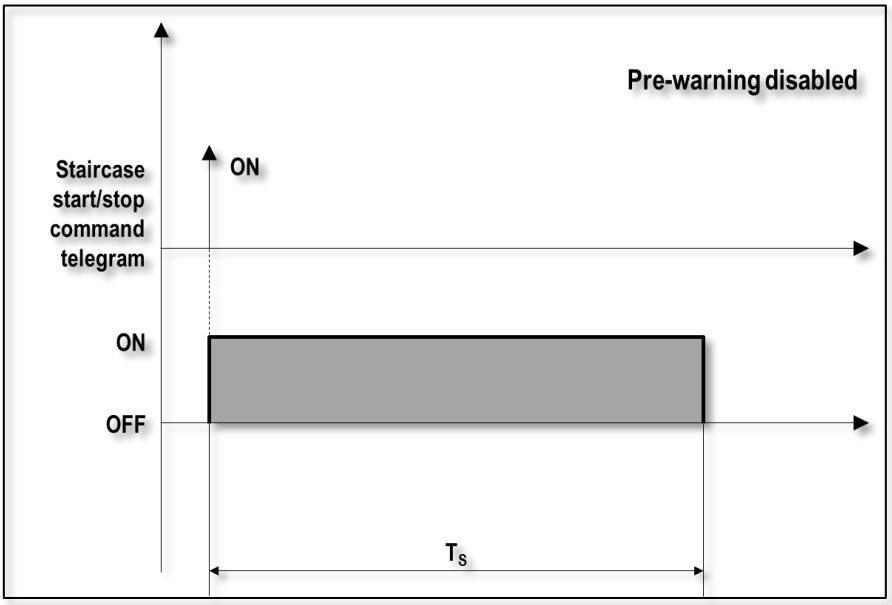


Figure 9 - Pre-warning feature (disabled)

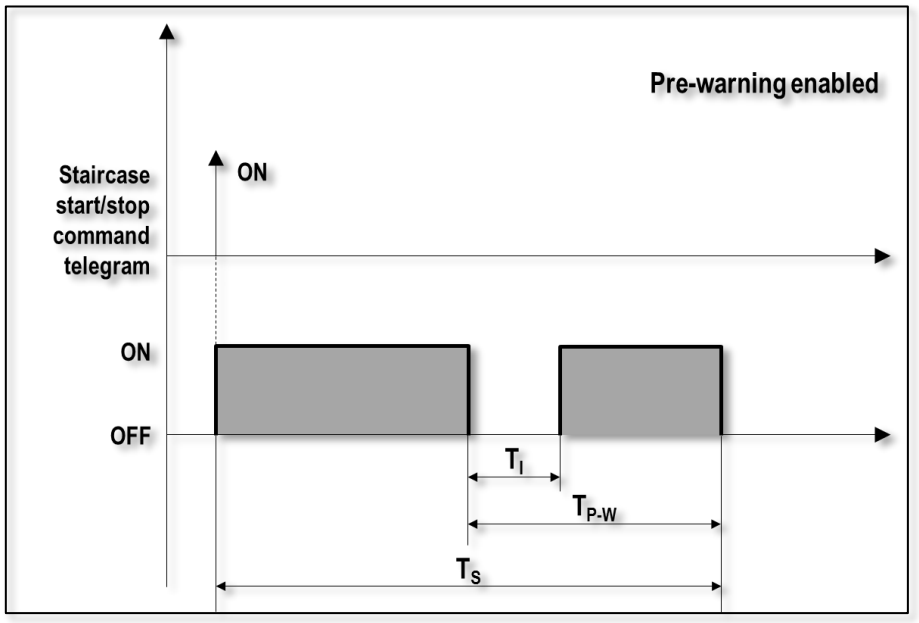


Figure 10 - Pre-warning feature (enabled)

6.4.4.5 Valve function

Each output has a Valve Function (*Use valve*), which allows you to set the output to work as a 3-way valve with PWM regulation. The parameters that can be set are:

- Valve position after timeout;
- Valve command timeout: when this time expires, without any operation being carried out, the valve positions itself as set in the previous parameter;
- PWM cycle time (min): is the period in which the output is maintained at the ON value for a time proportional to the value of the control variable;
- Minimum (control) valve value (%): allows you to adapt the PWM cycle to use with electrothermal actuators with very long opening times. For values of the control variable lower than the set value, the NC actuator remains unpowered, the NO actuator remains unpowered;
- Maximum (control) valve value (%): allows you to adapt the PWM cycle to use with electrothermal actuators with very long opening times. For values of the control variable higher than the set value, the NC actuator remains powered, the NO actuator remains unpowered.

Note: The minimum and maximum (contro) valve values are not taken into account if you set the valve position after timeout below or above these values.

6.4.4.6 Lock function

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

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

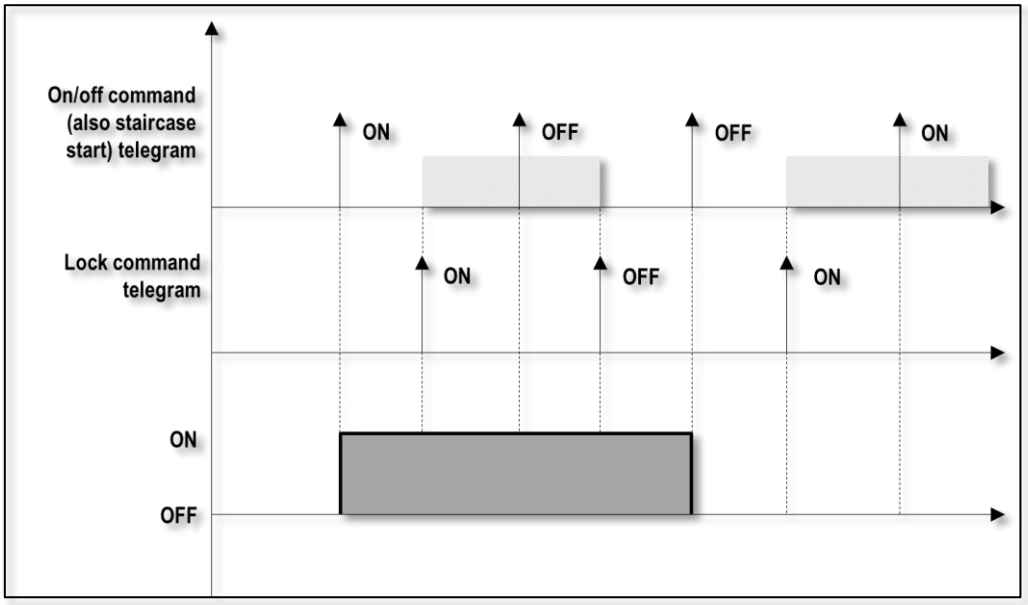


Figure 11- Lock function

6.4.4.7 Forcing function

The forced control is very similar to the basic direct command of the output value, but with the peculiarity that it overrides both the “regular” set value and every other value conditioning feature (i.e. logic function, staircase timing etc.).

It is possible to set what value the output should assume both when the output forcing is released and also on recovery after a bus power-off if forcing was previously in effect.

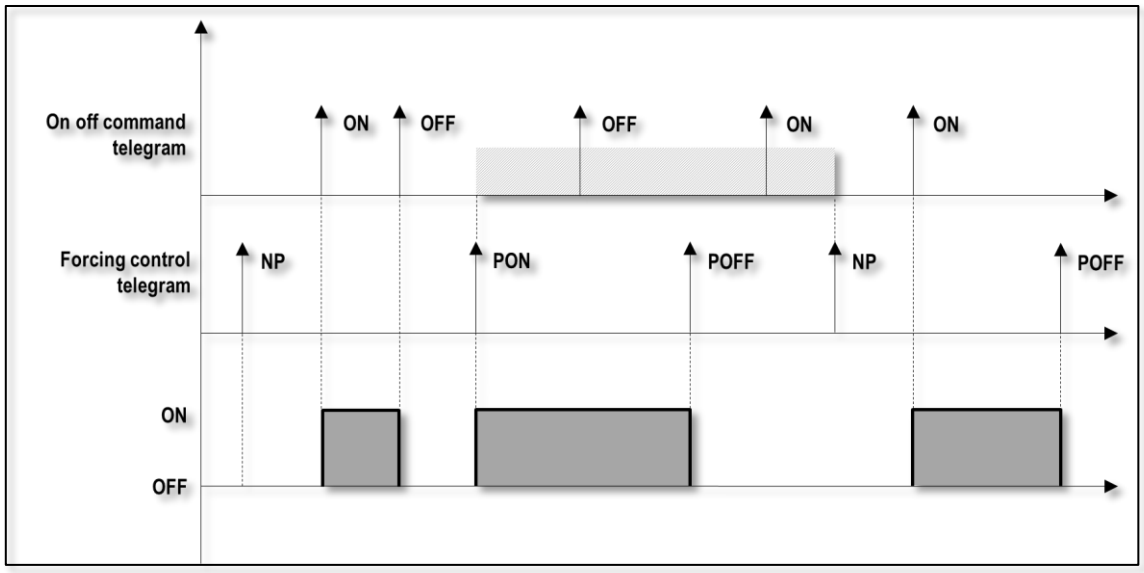


Figure 12 - Forcing function

The lock command has priority over the forcing function; this means that an output in the lock state cannot be controlled via the force commands.

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

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

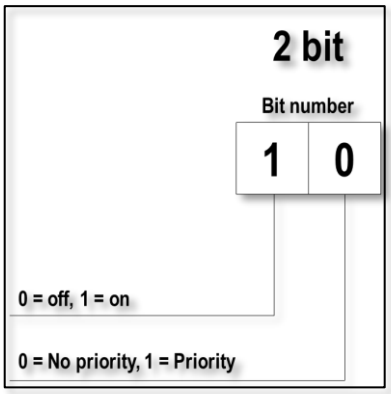


Figure 13 - Force command bits

6.4.4.8 Scene management

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

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

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

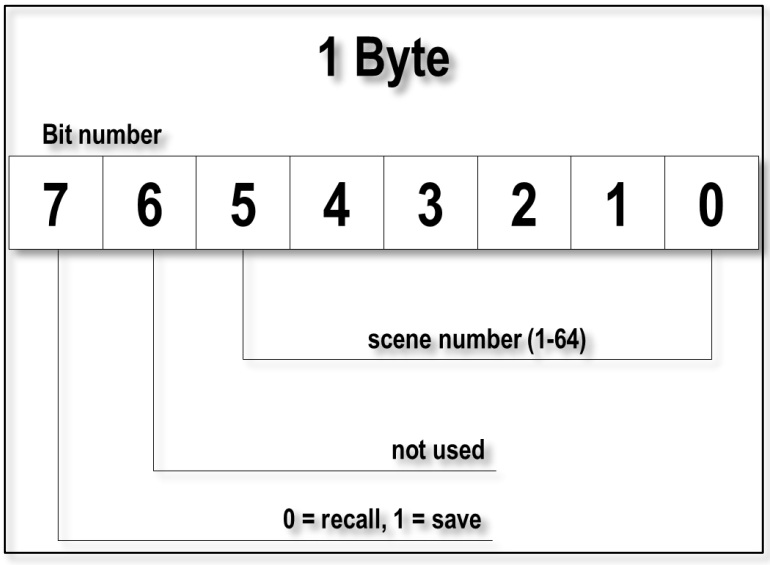


Figure 14 - Scene store / recall command code

6.4.4.9 Operating hours / Energy counter function

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

Together with the hour counter, a KNX object of the “energy counter (Wh)” type is created, which also has a communication object with a reset command. A specific parameter allows you to define a conventional value of electrical power in W associated with the load.

An additional parameter allows to define a conventional electrical power which is associated to the load; although this is not a “real” power metering, but merely a conversion factor between activation time and the estimated consumed power, nonetheless it can supply a useful indication for approximate power monitoring, particularly for resistive or fixed-power loads like lights or many other home or office appliances.

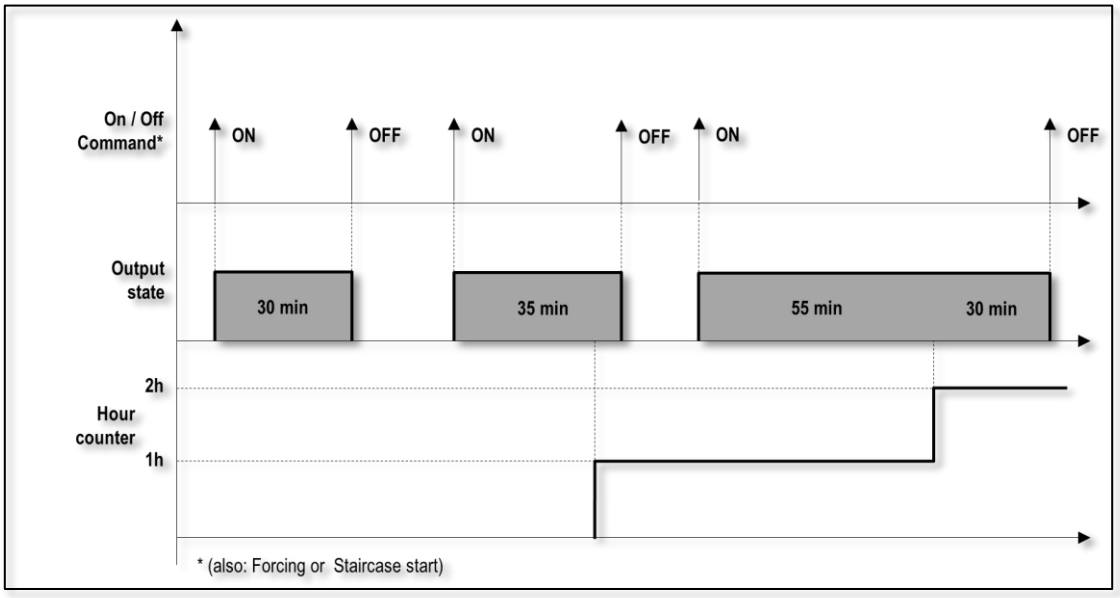


Figure 15 - Operating hours and energy counter

6.4.4.10 Output restore values for independent mode

As mentioned in an earlier paragraph, the status of the device after some significant events (see paragraph 6.2 for description) can be defined by configuration.

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

- On
- Off

- no change[1]
- previous value[2]
  (* this option is not available for either “bus off” or “after download” events).

The difference between “no change” and “previous state” is following:

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

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

[1] This option is not available for the “Behavior after download” event.
[2] This option is not available for the “Behavior at bus on” and “Behavior after download” events.

6.4.5 Output features in coupled mode

In coupled mode, output pairs can be used to drive three categories of devices: these are grouped under the denomination of 3-way v*alve*, *Shutters* and *Venetian Blinds.*

These categories have basically a similar operation mode, that is, they move a physical device from one to another endpoint; this can happen stepwise, with full stroke, or possibly stopping at given intermediate positions. The mentioned actuators, in the order they are listed, could be seen – apart from minor details - an increasingly sophisticated version of the same basic mechanism. Anyway, all three of them are driven through two lines, one for each direction.

For any single channel, one of these three types of behavior can be chosen.

Beside the distinctive features of these categories, there are further features common to all of them, like the locking and forcing functions, the meteo alarms and the scene management, which will be described below. Some of these features are similar to those described for those of single outputs in independent mode; in these cases, the corresponding sections in the previous paragraph are referenced.

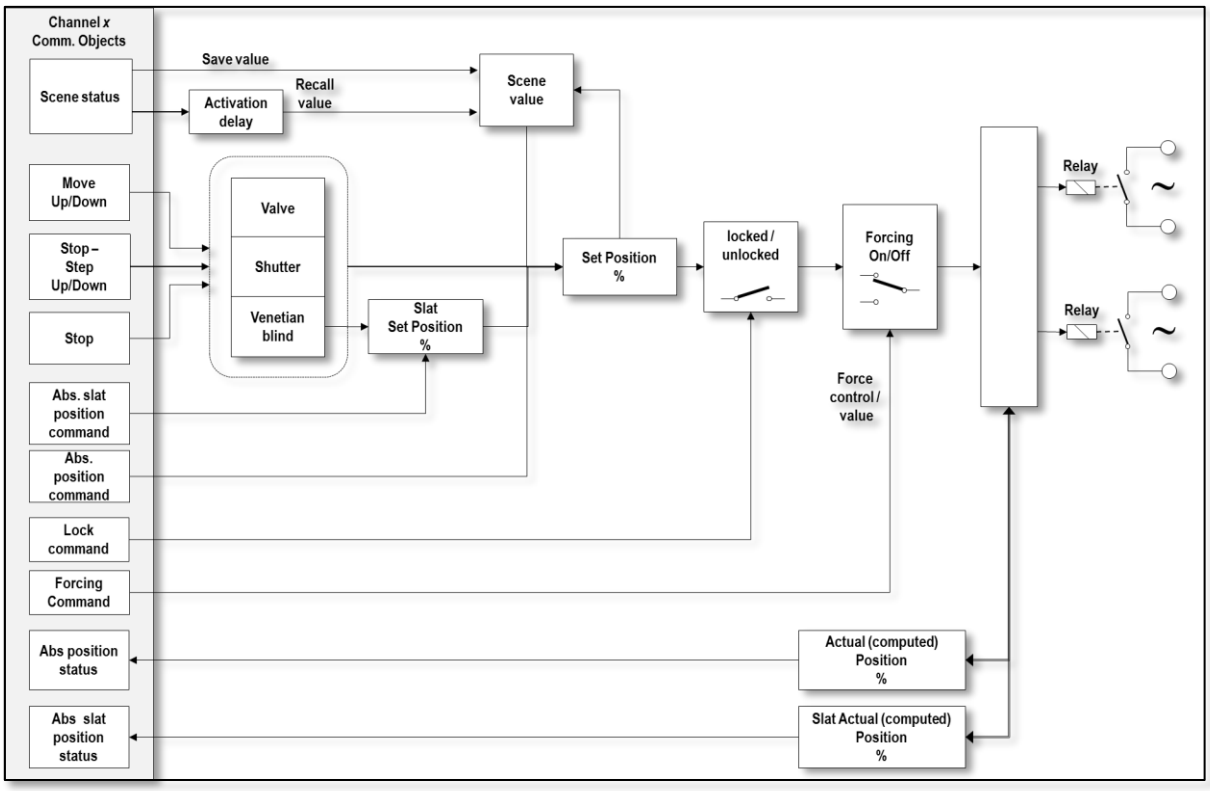


Figure 16 - Functional blocks – Coupled mode (referred to a single output)

6.4.5.1 Coupled output control basics

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

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

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

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

In cases where a movement must be performed to ensure that the end of travel is reached, to better characterize the up/down times it is possible to enable an “extra time mode” (full scale), which allows you to set two distinct time intervals:

- Time over full scale when moving up / opening (for the up / opening movement);
- Time over full scale in descent / closing (for the down / closing movement).

These intervals are to be understood as two additional optional time periods, in addition to the opening/closing times.

Warning:

Make sure your motor has adequate limit switches, otherwise this option may damage the operation.

Since no position feedback is available from the mechanical actuator, the shutter position is determined through movement timing: given the full-scale movement time value (i.e. the exact time the shutter / actuator

takes to move from one endpoint to the other), a partial movement expressed in a percent fraction of the full stroke will then correspond to the same fraction of movement time. The device keeps an internal position counter which is realigned whenever a full Move up/down command is issued.

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

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

6.4.5.2 Valve control

Valve control can be configured for 2-way or 3-way actuators.

A 2-way actuator has two control lines, each of which moves the valve to one of the extreme positions; the movement may not be instantaneous, but intermediate rest positions are not possible.

A 3-way actuator works in much the same way, except that the stroke is gradual (and usually slower); therefore, if both control lines are deactivated while the actuator is running, it will stop at the position reached at that moment.

The operating mode of the 3-way actuator is similar to that of the roller shutter (or shutter), except that it does not provide the On/Off type opening/closing command but only the absolute position command. Since this operation is described in the next paragraph, only the control for the 2-way actuator is described below.

This control provides the three basic commands already explained in paragraph 6.4.5.1; however, the “Stop / Step” command is implemented because it is required by the KNX specifications, but it has no effect since progressive movements are not possible. Even the “Stop” command has no real effect on the movement (it simply immediately deactivates both output lines).

The ordinary way of controlling a 2-way valve therefore requires the use of only the “Movement” command with one or the other direction specified, to move the valve in the two possible positions.

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

6.4.5.3 Shutter control

The shutter control applies exactly to the valve with 2-way actuator (for single outputs).

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

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

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

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

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

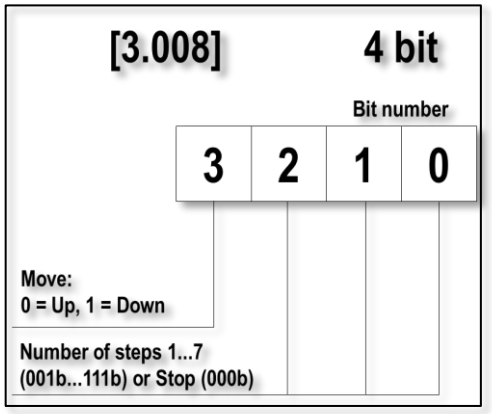


Figure 17 - Shutter control with dimming

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

The time to be specified is the actual and precise time of travel from end to end, which will be used for timing calculations. In cases where a movement must be made that guarantees reaching the end of travel, the full scale times can be used.

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

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

6.4.5.4 Venetian blind control

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

In terms of available commands and parameters, Venetian blinds differ from Shutters in following respects:

- The movement of the slats is defined by 3 parameters:
  - Number of slat steps
  - Opening time of the slat step
  - Closing time of the slat step

A step movement refers to the slats (not to the up/down movement); there is a specific parameter to define the time associated with the step (*opening/closing time of the slat step*), i.e. the activation time of the outputs that

causes the movement relative to the desired step; this period is repeated for a number of times specified by the *Number of slats step* parameter;

- an additional set of communication objects is available for setting and reading the absolute position of the slats, in addition to a “position reached” object;
- it is also possible to enable a dimmer control object for the slats.

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

As in the case of the roller shutter, also in the case of the venetian blind it is possible to enable a “extra time mode” (full scale), with the same methods described in the previous paragraphs.

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

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

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

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

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

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

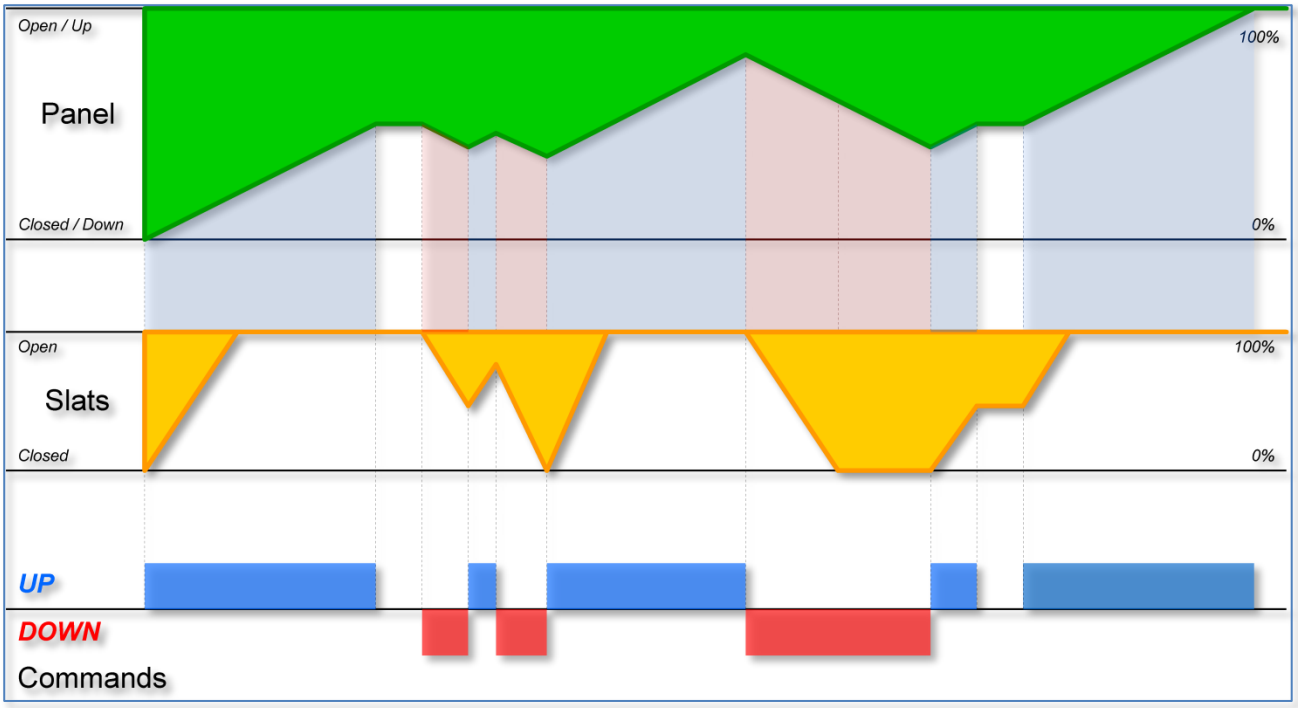


Figure 18 - Venetian blind control diagram

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

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

The blinds are initially raised until the slats are fully open (length L2), and then further to compensate for the mentioned length L1. After that, the downward line is activated for as long as necessary to bring the slats to their desired position. The final result is as intended.

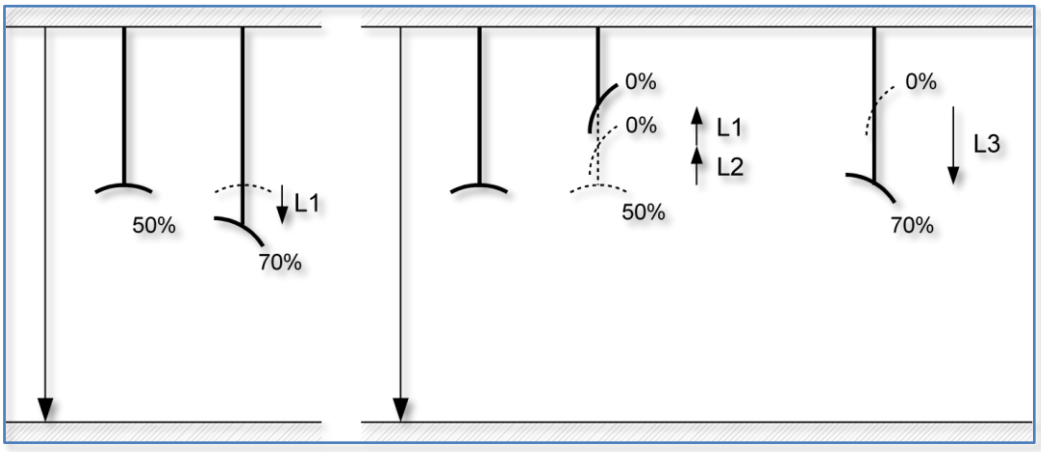


Figure 19 - Compensation for slat movement

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

Finally, you can enable a parameter to restore the slat position, or to leave it unchanged after repositioning, following an up or down movement.

Example of use:

- send a command to move the Venetian blind Down 100% (closed);
- with the Venetian blind closed, set the slats to 50%;
- send a command to move the Venetian blind Up (opening): at this point, the slats are positioned at 0% and the Venetian blind rises.

When the Venetian blind stops:

- if Restore slat position = YES, then the slats return to 50%;
- if Restore slat position = NO, then the slats remain at 0%.
-

6.4.5.5 Lock function

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

Further details can be found in the configuration section.

6.4.5.6 Forcing function

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

6.4.5.7 Scene management

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

6.4.5.8 Meteo alarms

Meteo alarms allow you to define pre-programmed positioning in the event of particular weather conditions detected by a weather sensor unit interfaced on the KNX bus (which must be purchased and installed separately).

Three types of meteo alarms can be managed independently, in particular for wind, frost and rain. The attribution is in fact only descriptive, since these three alarms are perfectly equivalent and could even be used for different types of events.

It is possible to define a specific behavior of the actuator upon receipt of each of these alarms: movement to the fully closed / down position, to the fully open / up position, or no action. Another behavior can be associated with the cessation of the alarm (all the previous options, plus the return to the state prior to the alarm).

If more than one alarm is activated, only the actions corresponding to the first alarm received are performed.

KNX alarms include an optional *heartbeat* function, which monitors at regular intervals the reception of the alarm telegram (whether the alarm is active or not), sent by an external source (e.g. a weather station). The purpose of this function is twofold: on the one hand it decreases the probability that an alarm condition will not be detected in case of a telegram loss, and on the other hand it confirms that the source is communicating and that no alarm is in progress if this is the case (alarm telegrams must be sent by the source with an “alarm inactive” information).

For each of the three available alarms, a separate timeout can be defined for the *heartbeat* function. When the source sends an alarm telegram (active or inactive), the timeout countdown starts at the same time. If no alarm information telegram is received within the configured timeout interval, the alarm is assumed to be active and the actuator is controlled accordingly. In practice, the alarm condition is a logical OR between the “alarm active” condition and the timeout expiration.

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

6.4.5.9 Output restore values in coupled mode

As mentioned in an earlier paragraph, the status of the device after some significant events (see 6.4.4.10 paragraph for description) can be defined by configuration.

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

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

Further details can be found in the configuration section.

6.4.6 Input actions

When an input contact closes, the device sends the telegram (or sequence) associated with it during programming to the KNX bus. It's similar to pressing a button and can be managed with "on-off" events (where "on" refers to pressing, "off" refers to releasing) or with "long/short press" events (for which a duration value is defined to distinguish between "long" and "short").

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

In the most common case, for example, one input could send an "ON" status telegram for a light point, while the adjacent one could send an "OFF" status telegram. Other typical application examples include increasing and decreasing the brightness of a lamp controlled by a dimmer unit, or raising/lowering commands for a motorized roller shutter or awning, and so on.

6.4.6.1 Types of input

The state of the inputs is determined by the state of the connected physical contacts.

The closing of an input contact can be associated with different effects on a state variable.

Each type of analog or digital contact on the device can be configured in two modes, so that it can be interfaced with either NO (normally open) or NC (normally closed) contacts.

From a logical perspective, these modes modify the interpretation of the "active" and "inactive" states of an input as follows:

• In NO mode, an open connection between the two input terminals (open contact) is associated with the inactive state of the input, and a closed connection is associated with the active state; this is obviously the most intuitively common mode;

• In NC mode, an open connection between the two input terminals (open contact) is associated with the active state of the input, and a closed connection is associated with the inactive state.

The two modes are therefore named according to the type of contact presumably intended to be connected.

6.4.6.2 Input events

The device recognizes two types of input events (or rather, pairs of events): "contact closed/open" and "short/long press."

The first type of event is a simple logical value: "Open" is a synonym for "inactive," while "Closed" is a synonym for "active."

> *It is important to note that the terms "OPEN" and "CLOSED," being common terms for indicating the states of an input, are to be interpreted logically and should not be confused with the state of the associated physical contact as described for the NO and NC input modes.*
>
> *For example, a NO contact in the active position is electrically and logically "closed," while a NC contact in the active position is electrically "open" but logically "closed."*

The second type of event that can be detected on an input is "short/long press." The term "press" refers to user-activated buttons, but the mechanism can obviously be applied to contacts on other devices.

The distinction is as follows:

- If an input remains active for a period shorter than a definable duration, the "short press" event is generated upon release;

- If the input remains active for longer than the defined duration, the "long press" event is generated upon expiration of the duration. After this point, the input can remain active for any length of time, and no further events will be generated either during the remainder of the activation or upon release (the next event can only be generated after a subsequent activation).

Refer to the timing diagrams in the figures below for an illustration of the differences between the various events; note that these diagrams represent the closing/opening conditions of the electrical contacts at the terminals as the triggers.

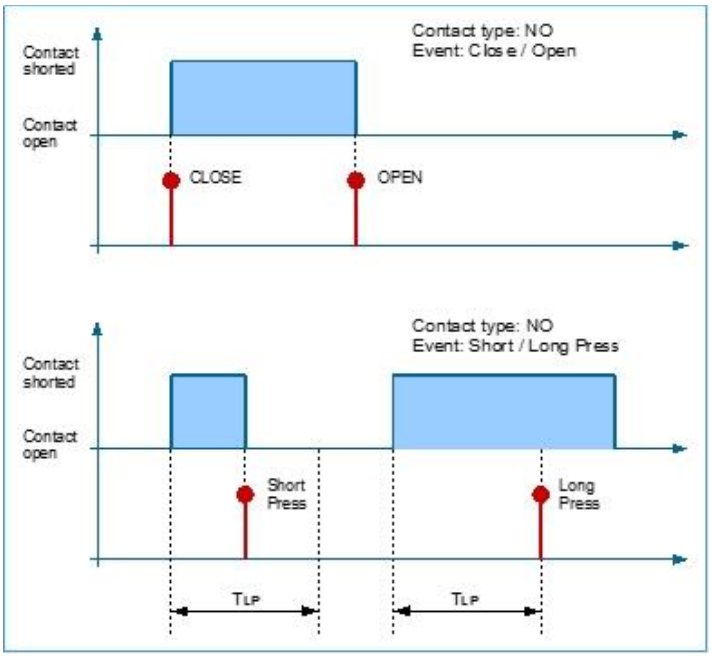


Figure 20 - time diagrams for Normally Open (NO) mode

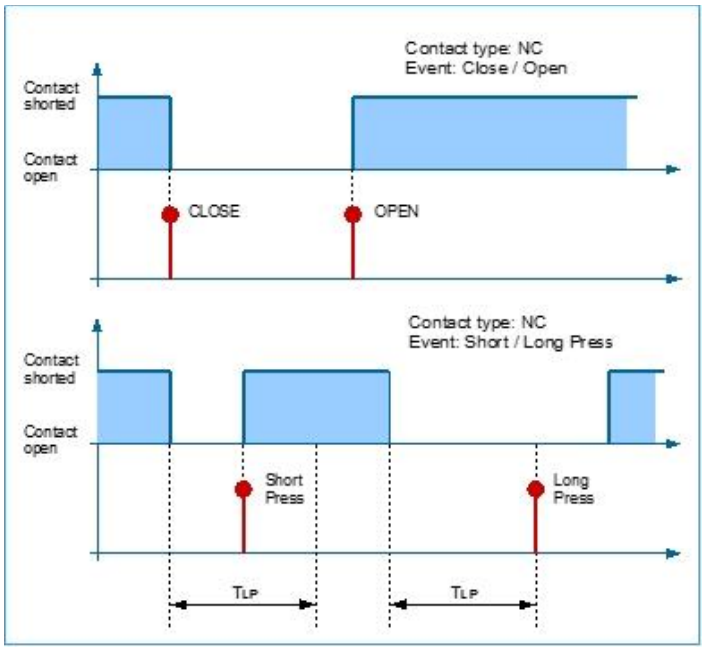


Figure 21 - time diagrams for NC (Normally Closed) mode

6.4.6.3 Lock function

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

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

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

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

6.4.6.4 Generic temperature acquisition

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

By configuring one of the 8 digital/analog inputs as a sensor via ETS, the device allows you to enable and configure up to eight thermostats for 8 independent zones.

For instructions on using the device as a temperature controller, please refer to section 6.4.11.

6.4.6.5 Input configuration and usage

The inputs described can be considered and used as independent; however, given the physical structure of the device and the nature of the functions it most commonly performs, the inputs can be associated in pairs. In this case, a pair can consist of two inputs located in adjacent positions on the terminal block (even input - odd input in the ETS application).

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

- In *single mode*, each input operates independently, has its own parameters and communication objects. This is the mode of operation described so far.
- In *coupled mode*, 2 inputs operate logically grouped under the same channel in order to perform a common functionality. Each input has only one possibility of coupling with a given other input: for example, inputs 1 and 2 for channel A, 3 and 4 for channel B, etc.

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

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

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

Throughout this document, as well as within the application program, for clarity, the term "Channel" will be used to indicate the pair of inputs.

Single (uncoupled) inputs will be referred to generically as "inputs."

6.4.7 Independent or single input

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

1. Send values or sequences

An event triggers the transmission of configurable values or sequences of values on the bus. These values can be logical or numeric with different dimensions.

A sequence can consist of up to 8 communication objects, each with a different type and value.

Configurable delays can be inserted between the values in the sequence.

2. Dimming

This mode is used in combination with KNX dimmer actuators to control lighting devices.

The function is activated only with long/short contact events:

- with a short contact, the device sends on/off commands to the dimmer;
- with a long contact, the dimming percentage is varied – up or down – until the input is released.

3. Shutters or Venetian Blinds

This mode is used in combination with KNX dimmer actuators to control motorized shutters, blinds, or similar devices.

These actuators have functions for opening and closing shutters; two types of movement can be selected: continuous or intermittent.

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

- If toggle mode is enabled, each time a specific input is activated, the direction of movement is reversed; if it is disabled, the direction is fixed and can be set to "raise" or "lower";
- If Venetian blind mode is enabled, the device sends a "raise/lower all" command for a long press, and a "step" command for a short press; if it is disabled, the command for a long press is the same, but a "stop" command is sent for a short press.

4. Scenario

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

This feature allows you to save and recall a scenario setting communication object; specifically, the device sends a "store" or "recall scenario" command to the actuators following a short/long press.

The configuration options are as follows:

- Activates the selected scenario with a short contact, and stores the current configuration as the selected scenario with a long contact;
- Activates one scenario with a short contact, and another with a long contact.

5. Shift register

Closing the input contact activates a countdown, configurable in terms of start/stop position, increment/decrement step, and reset.

6.4.7.1 Dimming

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

Dimmer control is essentially based on a 4-bit communication object whose data has the format shown in the figure:

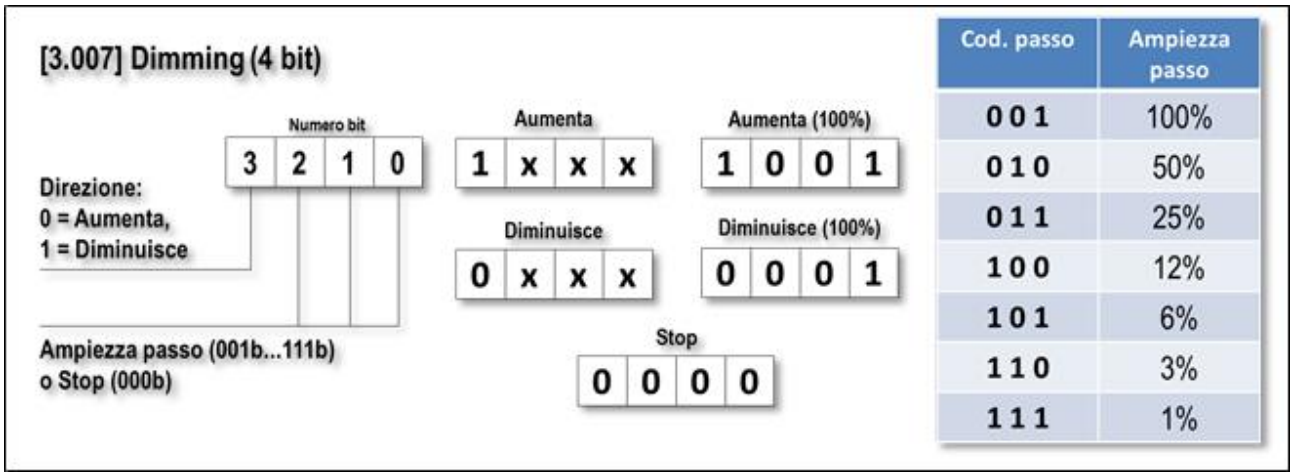


Figure 22 - Dimming communication object

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

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

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

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

The defined operations and related commands are the following ones:

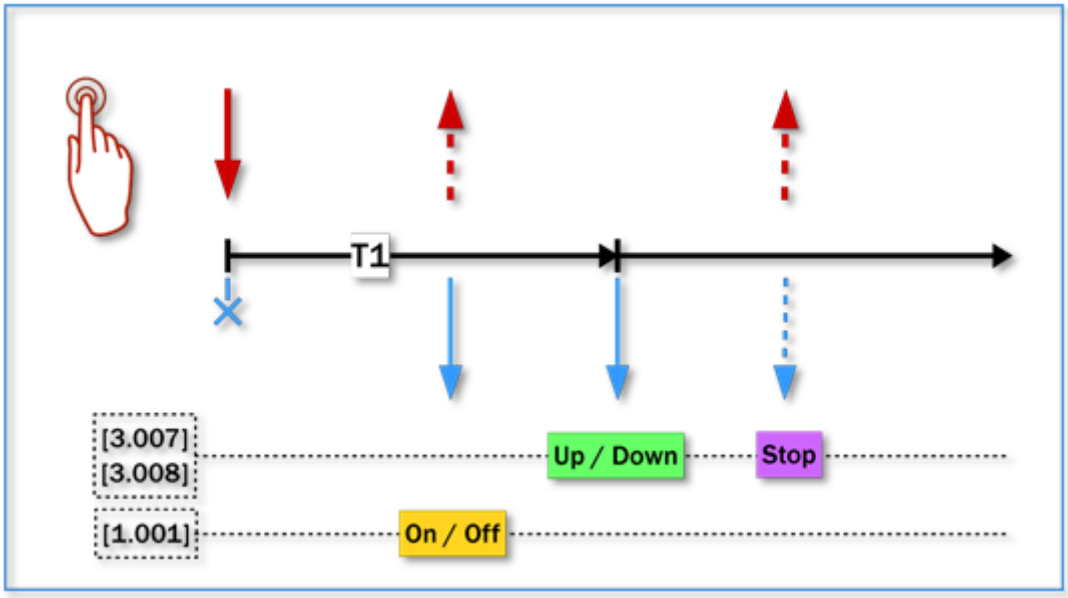


Figure 23 - Dimmer mode command sequence

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

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

6.4.7.2 Shutter or Venetian blind control

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

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

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

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

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

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

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

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

In the simplest version, on command side:

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

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

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

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

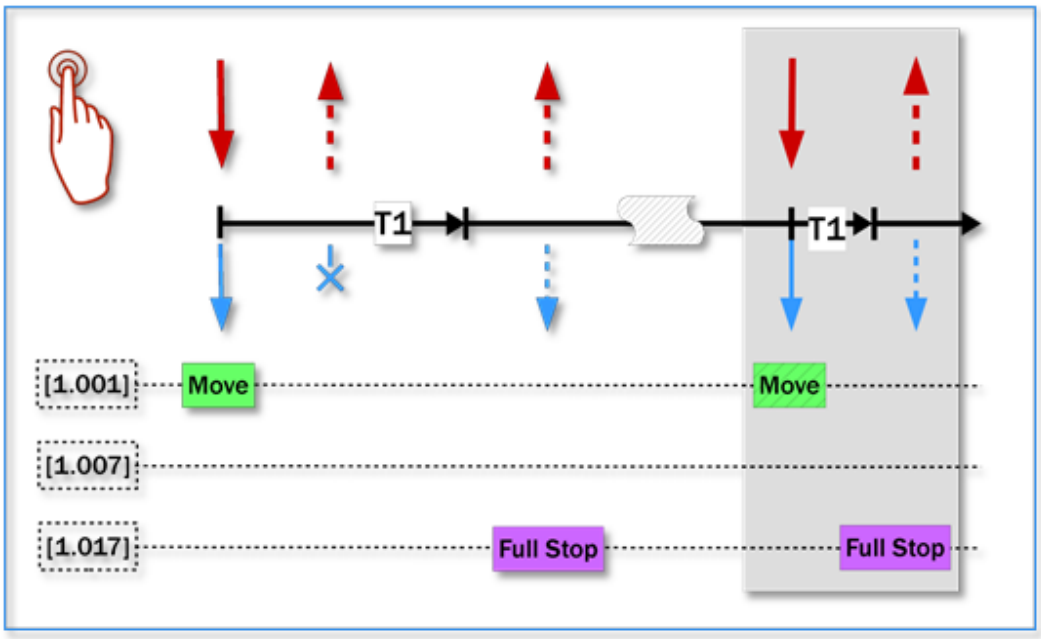


Figure 24 - Shutter” mode command sequence

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

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

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

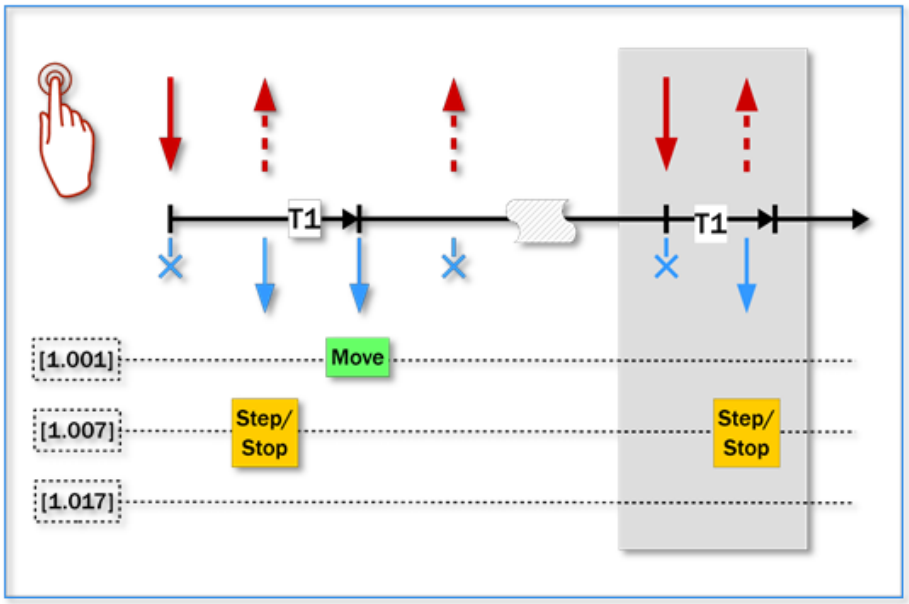


Figure 25 -" Venetian blind” mode command sequence

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

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

6.4.8 Logic functions

The device has a limited provision for the logic processing of internal variables in order to condition the status of outputs.

The direct command can be applied to the input of a block with a logical operation selectable between:

- *OR;*
- *AND;*
- *XOR (Exclusive OR);* the output is ON if the number of inputs ON is odd;

- *NOR (Negated OR);*
- *NAND (Negated AND);*
- *XNOR;* the output is ON if the number of inputs ON is even;
- *IDN (identity operator);* always returns the ON value.

Up to other 8 objects can be provided as inputs to the same block, which are connected to communication objects accessible via bus to other external devices. Each of these objects can be individually applied, if desired, a negation operator that inverts its value.

The input objects are logically combined as in following picture:

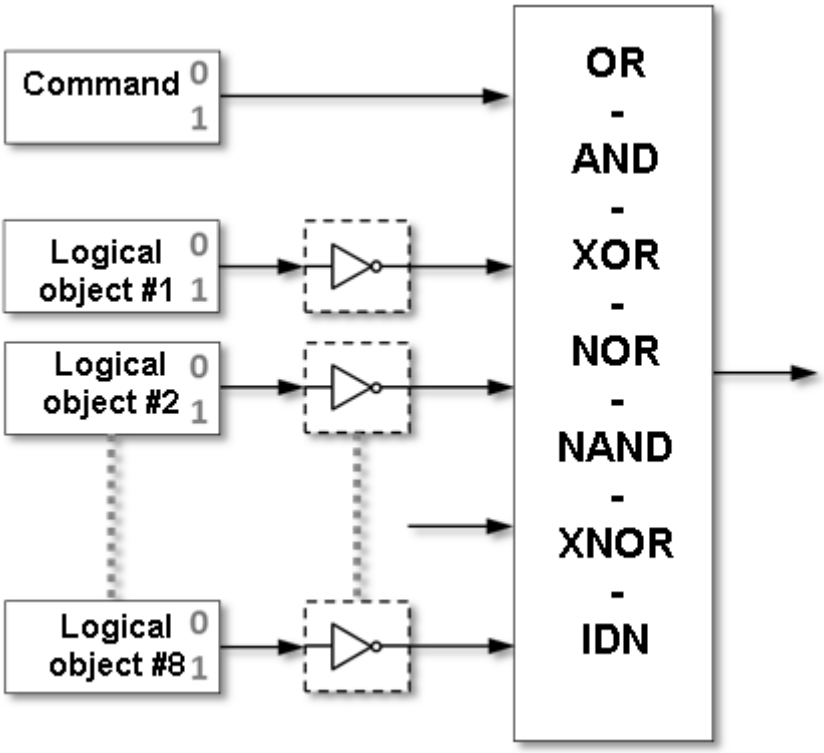


Figure 26 - Logical combination function

The logic combination block on the right works as follow according to which logical operation is selected:

It must be noted that, in the above description, "input" and "output" are referred to the logical block; for the purpose of operation, the actual "inputs" are the logic objects, thus the optional inverters must be factored in.
This structure allows to implement fairly complex logical combinations; a more generic and powerful programming capability would add more complexity and therefore it would be far beyond the scope of an output module that is simple to use.
In the following pictures, the basic logic functions are illustrated, assuming the output command and one logic object are used:

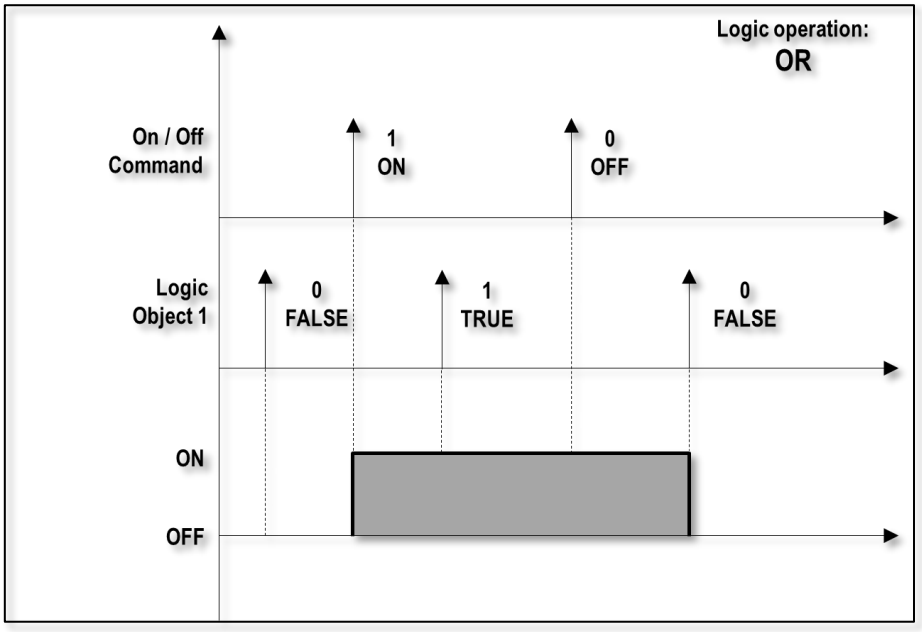


Figure 27 - Logic OR function

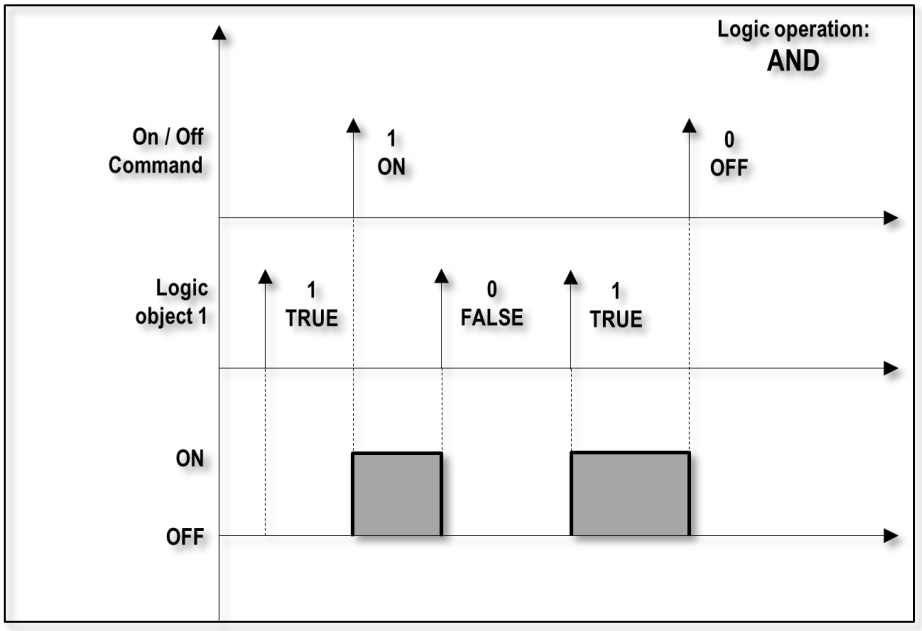


Figure 28 - Logic AND function

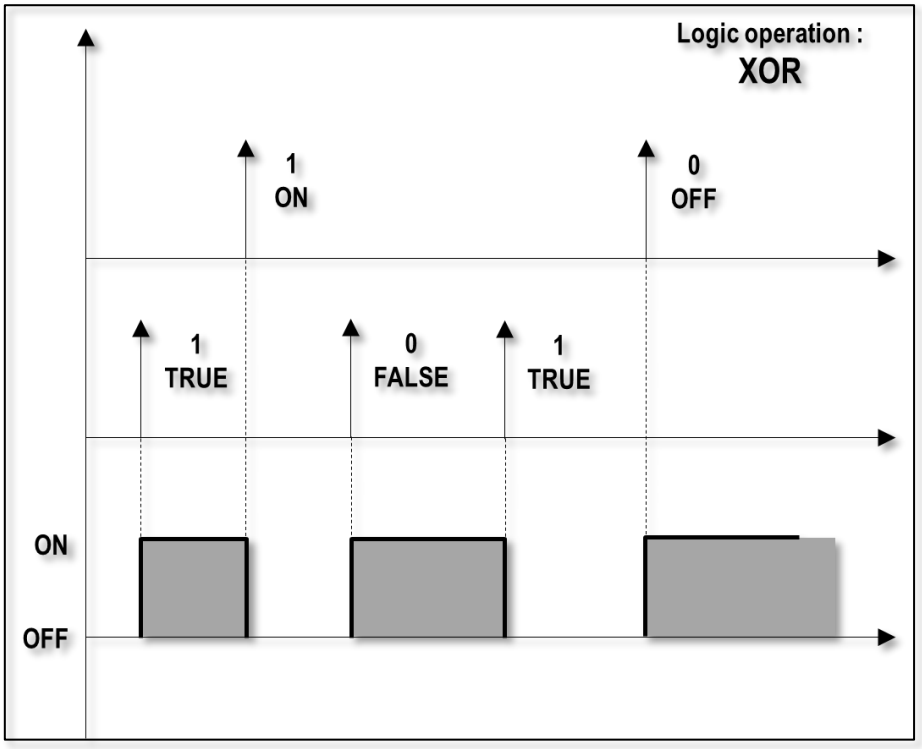


Figure 29 - Logic XOR function

6.4.9 Comparison functions

Up to 8 comparison functions are available to compare two objects with assigned data types. The output of the operation can be True or False.

The comparison objects can be:

- Communication objects, with the following data types:
  - 1 bit;
  - 2 bit;
  - 1 byte percentage;
  - 1 byte signed;
  - 1 byte unsigned;
  - 2 byte signed;
  - 2 byte unsigned;
  - 2 byte float.
- Objects internal to the device (only for Comparison 1 object), namely:
  - Value of the output relay (from 1 to 8 for EK-FG1-TP-I, from 1 to 16 for EK-FI1-TP-I);
  - Value of the output channel (A,B,C,D for EK-FG1-TP-I, or A,B,C,D,E,F,G,H for EK-FI1-TP-I).
  - Numeric value (only for Comparison 2 object), expressed in 2 byte float, in the interval [-671088,…,670597].

The available comparison operations are:

- Equality;
- Diversity;
- Less than;
- Greater than;
- Less than or equal;
- Greater than or equal.

It is also possible to assign a hysteresis for the comparison, in percentage value. This means that the comparison operation is true or false, unless the assigned hysteresis: for example, the comparison of equality between the numeric values 100 and 90, with a hysteresis of 10%, returns the value "True", because this hysteresis gives the value 90 to all values between 90±15% = [76.5 ÷ 103.5] and 100.

Note: Hysteresis applies to the second comparison object.

Note: Due to the limitations imposed by the KNX protocol, for the 2 bytes floating point data type, values greater than 670100 or less than -670100 are not considered valid. For this reason the results of operations performed on them are not reliable.

6.4.10 Arithmetic functions

Up to 8 arithmetic functions are available to perform mathematical or statistical operations on up to 8 input objects.

The result of the operation is made available in a specific object.

These functions are:

- Minimum
- Maximum
- Average
- Sum
- Difference
- Multiplication
- Division

The available data types, both for individual input arithmetic objects and for the result, are as follows:

- 1-bit value
- 2-bit value
- 4-bit value
- 1-byte percentage
- 1-byte signed
- 1-byte unsigned
- 2-byte signed
- 2-byte unsigned
- 2-byte floating-point value
- 4-byte signed
- 4-byte unsigned
- 4-byte floating-point value

Note: The division operation takes only 2 input objects (dividend and divisor)

6.4.11 Use as a regulator (room thermostat)

Temperature controllers integrated into the input behavior allow room temperature acquisition via bus from another KNX device, such as an ekinex® pushbutton with a temperature sensor, or from an NTC sensor connected to the input.

To optimize or correct room temperature regulation in specific cases (in large rooms, in the presence of strong asymmetry in temperature distribution, when the pushbutton is installed in a position unsuitable for room temperature measurement, etc.), the device can therefore use a weighted average between two temperature values. The weights are assigned using the Relative weight parameter, which assigns a proportion to the two values.

Note on mounting position

If using the integrated temperature controller, the NTC sensor should preferably be installed on an interior wall at a height of 1.5 m and at least 0.3 m from doors. The sensor should not be installed near heat sources such as radiators or household appliances, or in locations exposed to direct sunlight. If necessary, a weighted average of the temperature value detected by the NTC sensor connected to the input and a value received via the bus from another KNX device can be used for regulation.

Applications

The configurable uses are specific to single-stage heating systems and include the following terminals: radiators, electric radiators, and radiant panel systems.

Temperature control can be:

• two-point control with hysteresis, ON-OFF control type;

• proportional integral, with ON-OFF, PWM, or continuous control type.

Room air temperature is controlled by opening and closing the shut-off valve(s) on the circuits that make up the distribution manifold, with an ON/OFF or PWM control algorithm. Electrothermal actuators and/or servomotors can be used to control the valves.

Figure 30 shows the components of a generic room temperature control system. The temperature controller (thermostat) detects the current temperature of the room air mass (Teff) and compares it with the desired temperature value or setpoint (Tset).

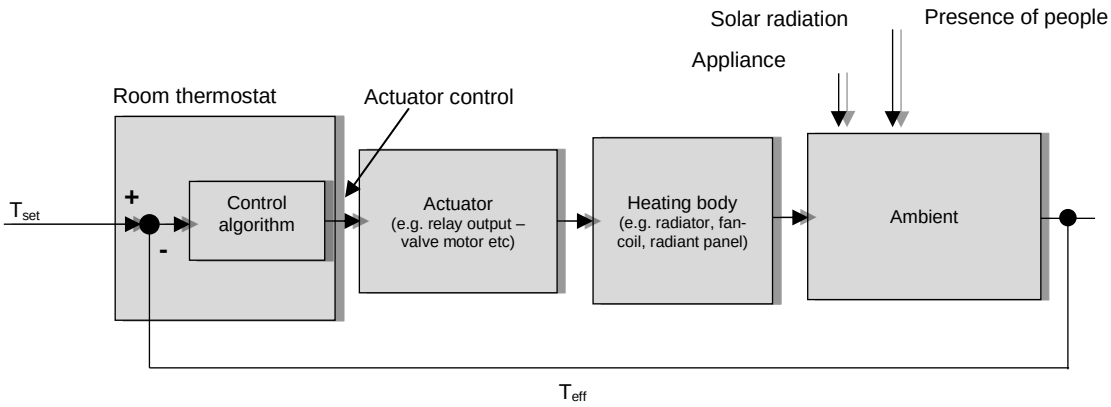


Figure 30 - Generic control system for room temperature regulation

6.4.11.1 Control algorithm

The control algorithm, based on the difference between Tset and Teff, processes a command that can be a percentage or on/off command; the command is represented by a communication object that is transmitted via bus to an actuator device periodically or upon switching events. The output of the actuator device is the manipulable variable of the control system, which can be, for example, a water or air flow rate. The control system implemented by the room thermostat is of the feedback type (or closed loop); the algorithm takes into account the effects on the system to modify the amount of control itself.

6.4.11.2 Continuous Proportional-Integral control

The Proportional-Integral control (PI) is described by the following relation:

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

where:

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

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

$Kp = proportional\ constant$

$Ki = integral\ constant$

The control variable is composed of a part that depends proportionally from the error and a part that depends on the integral of the error.

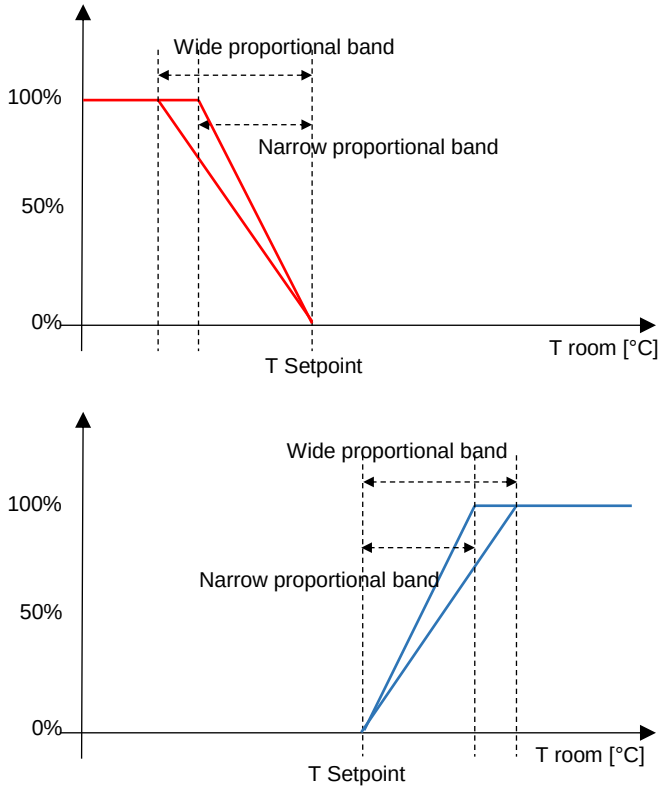
In practice derived values with a more intuitive meaning are used.

$$Proportional\ band\ BP\ [K] = \frac{100}{Kp}$$

$$Integral\ time\ Ti\ [min] = \frac{Kp}{Ki}$$

The proportional band is the error value which determines the maximum excursion output at 100%.

For example, a regulator with proportional band of 5 K provides a 100% control output when the Setpoint = 20°C and the measured temperature is ≤ 15°C in heating; in the cooling conduction mode, it provides a 100% control output when the Setpoint = 24°C and the measured temperature is ≥ 29°C. As shown in the figure, a regulator with a small proportional band tends to provide higher values of the control variable for small errors than a regulator with a higher proportional band.



The integral time is the time required to repeat the value of the control variable of a purely proportional regulator, when the error remains constant in time.

For example, with a purely proportional controller in heating and with a value of proportional band of 4 K, if the setpoint is = 20°C and the measured temperature is = 18°C, the control variable assumes the value of 50%. With an integral time = 60 minutes, if the error remains constant, the control variable will take the value = 100% after 1 hour, i.e. a contribution equal to the value given by only proportional contribution will be added to the control variable.

In heating and air conditioning systems, a purely proportional controller is not able to guarantee the achievement of the setpoint. You should always introduce an integrated action for achieving the Setpoint: that is why the integral action is also called automatic reset.

6.4.11.3 Two-point control with hysteresis

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

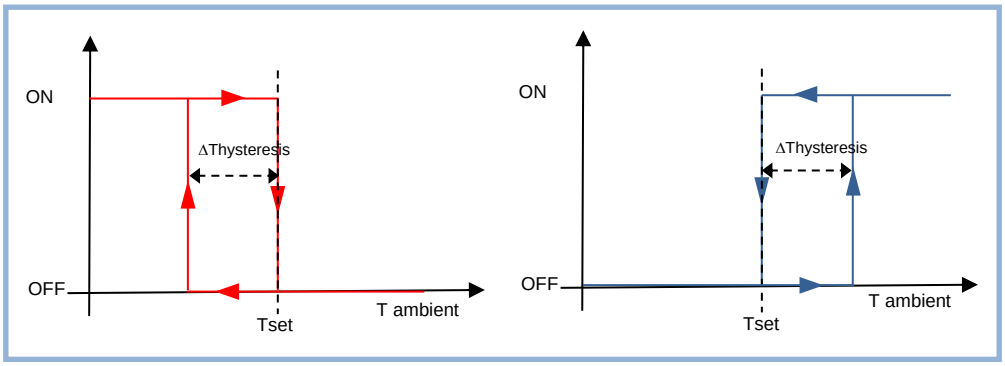


Figure 31 - Temperature control with hysteresis

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

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

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

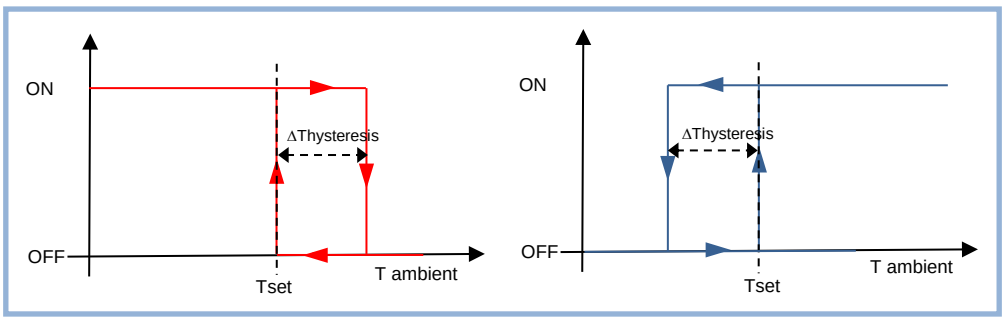


Figure 32 - Two-points temperature regulation

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

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

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

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

The desired temperature (T_{set}) is generally different for each one of the 4 operating modes and for heating/cooling modes. The different values are defined for the first time during ETS configuration and can be modified later on. In order to optimize energy saving (for each extra degree of room temperature, outbound dispersions and energy consumption go up 6%), it is possible to take advantage of the multi-functionality of the domotic system, for example with:

- Hour programming with automatic commutation of the operating mode by means of KNX supervisor;
- Automatic commutation of the operating mode according to presence of people in the room;
- Automatic commutation of the operating mode according to window opening for air refreshment;
- Circuit deactivation when desired temperature is reached;
- Flow temperature reduction in case of partial load.

6.4.11.4 PWM Proportional-Integral control

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

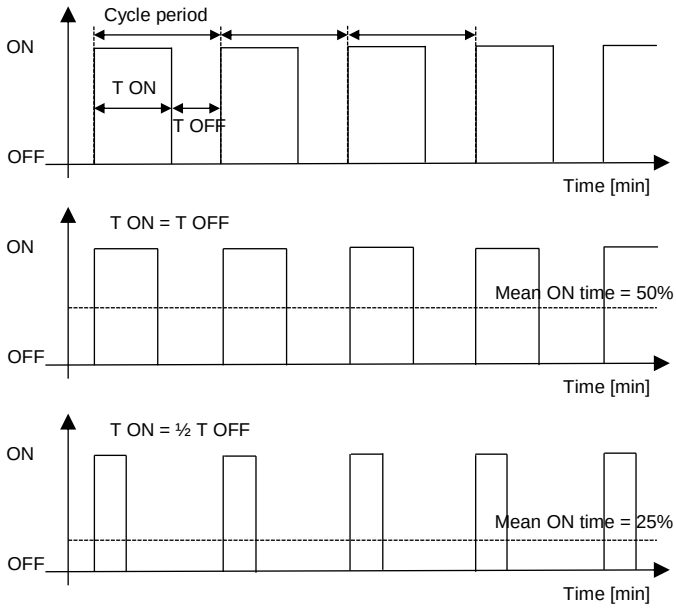


Figure 33 - PWM regulation scheme

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

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

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

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

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

6.4.11.5 Setpoint management

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

Three setpoint management modes are foreseen:

- Single setpoint
- Relative setpoints;
- Absolute setpoints.

6.4.11.5.1 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).

6.4.11.5.2 Relative setpoints

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

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

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

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

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

6.4.11.5.3 Absolute setpoints

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

- Comfort setpoint
- Stand-by setpoint
- Economy setpoint
- Biulding protection setpoint

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

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

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

6.4.11.6 Operating modes

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

- Temperature setpoint
- Building protection setpoint

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

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

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

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

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

6.4.11.7 Heating/Cooling switch over

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

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

6.4.11.7.1 Switch over 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 or the EK-EP2-TP room thermostat. The integrated temperature controller acts as a "slave": the switch over is carried out by input communication object [DPT 1.100 heat/cool].

6.4.11.7.2 Automatic switch over, based on the room temperature

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

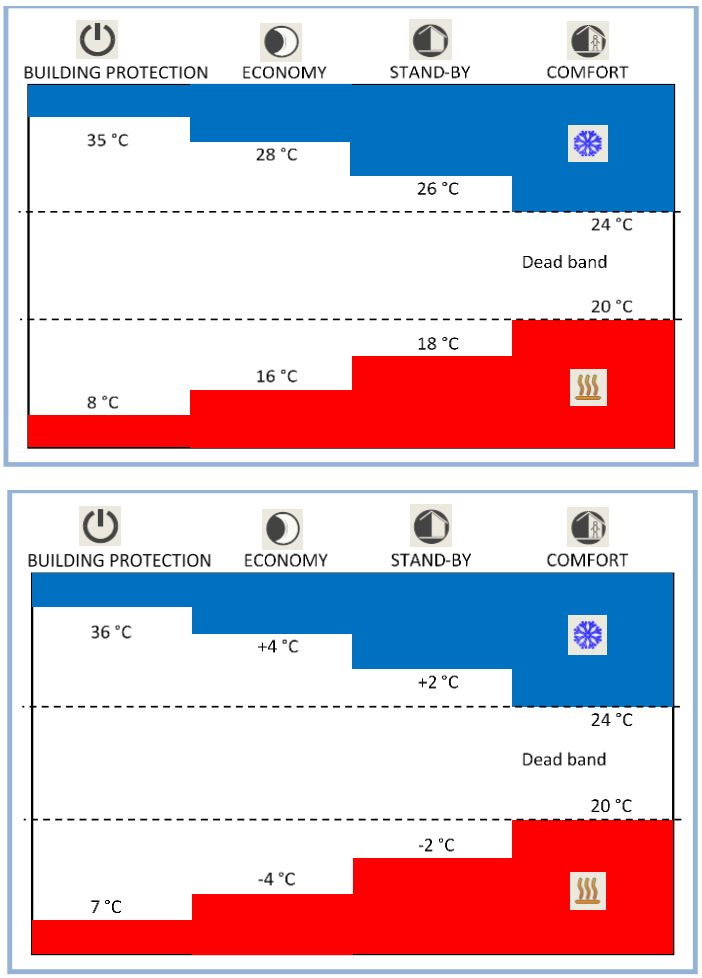


Figure 34 - Automatic switch-over, based on room temperature

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

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

6.4.11.8 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*.

6.4.11.9 Input from bus

6.4.11.9.1 Characteristics and timeout

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

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

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

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

6.4.11.9.3 Surface temperature limitation function (external object)

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

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

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

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

The surface temperature limitation may be realized for several purposes:

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

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

The surface temperature limitation function closes the intercept valves on the distribution manifold when the temperature measured on the panel rises above set threshold (default value 29°C). The regular operation of

the room thermostat resumes when the measured surface temperature drops below the hysteresis threshold (29°C - 0,3 K).

6.4.11.9.4 Anticondensation protection function (external object)

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

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

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

6.4.11.9.5 Window contacts (external object)

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

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

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

6.4.11.9.6 Presence sensors (external objects)

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

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

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

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

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

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

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

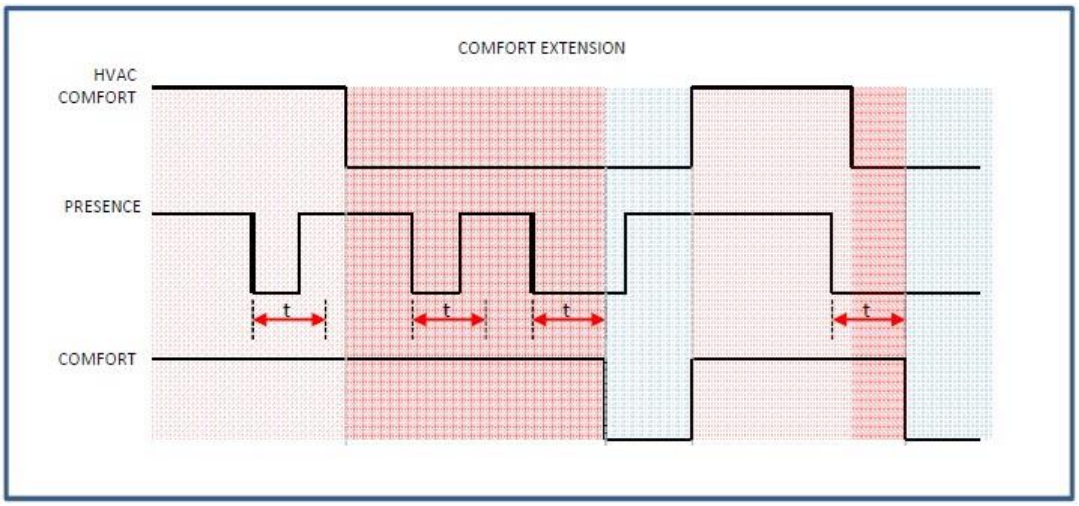


Figure 35 - Comfort extension

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

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

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

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

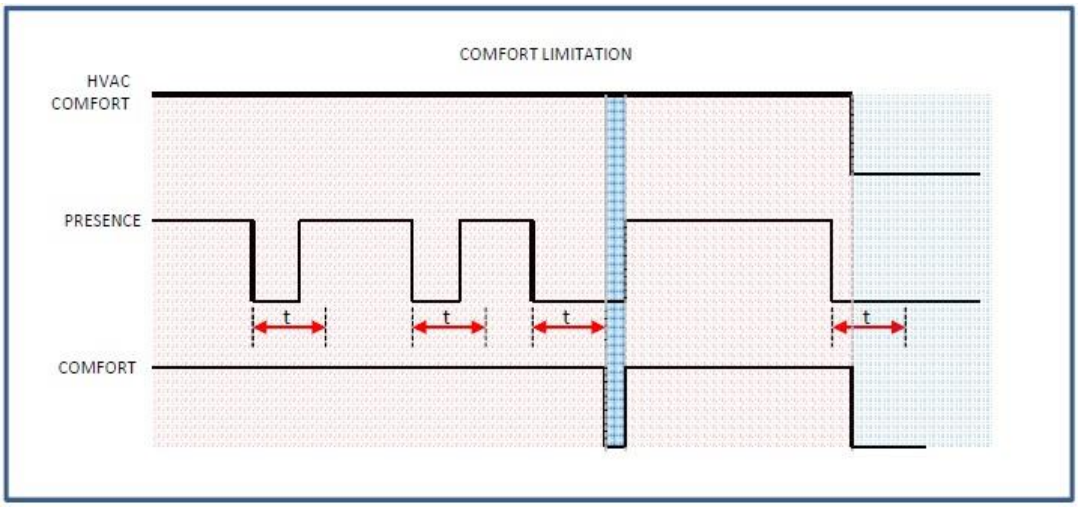


Figure 36 - Comfort limitation

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

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

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

7 Firmware upgrade

It is possible to update the device firmware via KNX, using the Ekinex Firmware Updater tool.

Consult the specific guide of the update software and consult the Ekinex website of the product to always have the latest versions available.

8 ETS application program

This section of the manual lists all configurable parameters and describes the related communication objects at the same time.

Each channel/output has the same parameters and makes the same types of communication objects available, but the configuration is independent for each of them.

In the following, a generic output will be referred to with “x” (where x = 1…8 for EK-FG1-TP-I or x = 1…16 for EK-FI1-TP-I).

The parameter values highlighted in bold represent the default value.

The device parameters are divided into general parameters and specific parameters, grouped into tabs.

Figure 37 shows the tree structure of the application program with the main tabs.

The numbers of communication objects in the “CO(s) numbers” column of each table are divided into 2 groups. The objects of the second group (in *italics*) are related to the 16-output device EK-FI1-TP-I only.

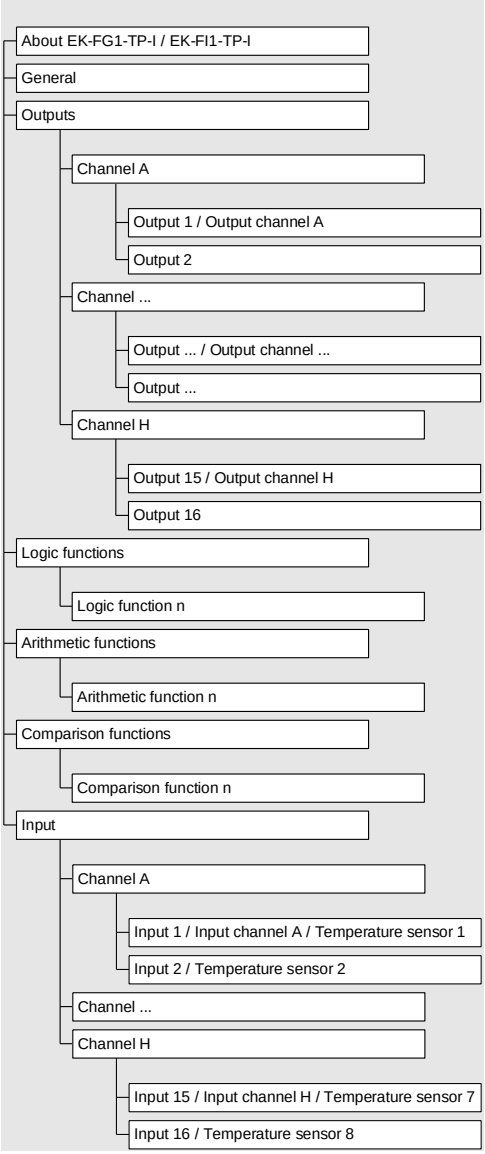


Figure 37 - ETS application program parameters

8.1 About

The **About EK-FG1-TP-I / EK-FI1-TP-I** sheet is for informational purposes only and does not contain any parameters to be set. The information provided is:

Application software per ETS5 e ETS6
Versione 1.00 (or later)
EK-FG1-TP-I – KNX 8-output universal actuator, or

EK-FI1-TP-I – KNX 16-output universal actuator

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

8.2 General

The following parameters are available in the General tab:

Parameter name	*Conditions*	*Settings*
Product type	-	**EK-FG1-TP-I**, or **EK-FI1-TP-I**
	It allows the user to select the device type. The field is currently fixed, as only 1 device is supported.	
Manual operation	-	**disabled** enabled
	Enables the front button that allows you to switch to manual mode.	
Manual disable from bus	Manual operation = enabled	**disabled** enabled
	Allows you to disable switching to manual mode via a bus command.	
Enable logic module	-	**disabled** enabled
	Enable or disable the section for logical functions.	
Number of logic functions	Enable logic module= enabled	**0**…8
	Sets the number of logic functions to enable.	
Number of compare functions	Enable logic module= enabled	**0**…8
	Sets the number of compare functions to enable.	
Number of arithmetic functions	Enable logic module= enabled	**0**…8
	Sets the number of arithmetic functions to enable.	
Restore auto mode	Manual operation = enabled Manual disable from bus = disabled	hh:mm:ss (**00:15:00**)
	Sets the time after which the device returns to automatic mode.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Manual commands disabled	Manual operation = enabled Manual disable from bus = enabled	1 bit	C-W--	[1.001] switch	1

The remaining device settings are divided in two main groups: the general channel configuration settings and the channel-specific settings.

8.3 Outputs configuration

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

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

For outputs 2 and above, instead of being explicitly defined, the output configuration can be copied from any of the preceding channels. If this option is selected, the corresponding channel can be made to perform the exact same kind of function as the source channel. This allows the user to spare time in configuring the device, at the same time assuring that there is no inconsistency between two channels that are meant to be configured in exactly the same way.

Additionally, it is also possible to copy a channel: for example, channel B can be copied from channel A. Please note that a channel can only be a copy of a channel that is not itself copied, otherwise a message like the following will be displayed:

It must be noted that to copy the configuration from another output or channel is just a shortcut for the selection of configuration options; it is in no way implied that the two channels share any of the involved communication objects.

If the configuration of the original channel is varied, then so is the “derived” channel; in the same fashion, if the original channel is disabled, so is also the derived one.

8.3.1.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Output channel *n*	-	disabled **independent or single** 3-way valve / shutter / venetian blind Copy parameters from channel*
	Enable output channel n (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I) ** This option is only available for channels B,C,D for EK-FG1-TP-I and B,C,D,E,F,G,H for EK-FI1-TP-I.*	
Output channel *n* Source channel	Output channel n = copy parameters from channel	**A** / B / C for EK-FG1-TP-I **A** / B / C / D / E / F / G for EK-FI1-TP-I
	For output channel n (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I), copy the parameters from one of the previous channels, provided that it is not already a copied channel.	
Output channel *n* - Output 1 Output 3 Output 5 Output 7	Output channel n = independent or single	disabled **enabled**
	Enables the first of the two outputs of channel n (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).	
Output channel *n* - Output 2 Output 4 Output 6 Output 8	Output channel n = independent or single	disabled enabled **copy parameters from output (A1/B3/C5/D7)** or **copy parameters from output (A1/B3/C5/D7/E9/F11/G13/H15)** for EK-FI1-TP-I

Parameter name	*Conditions*	*Settings*
	Enables the second of the two outputs of channel n (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I),	
Output channel *n* - Use	Output channel n = 3-way valve / shutter / venetian blind	**valve** shutter venetian blind
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Defines the functionality associated with the output pair.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output x – On/Off command	Output channel n = independent or single valve function = disabled	1 bit	C-W--	[1.001] switch	170, 173, 176, 179, 182,185, 188, 191, *194, 197, 200, 203, 206, 209, 212, 215.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *This communication object is the direct command for setting the output (x=1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				
Output x – On/Off status	Output channel n = independent or single AND Status feedback telegram = enabled	1 bit	CR-T-	[1.001] switch	172, 175, 178, 181, 184, 187, 190, 193, *196, 199, 202, 205, 208, 211, 214, 217.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Sent at each output state change, and also periodically if configured (x=1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				

8.3.2 Independent outputs: channel n / output x configuration

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

In general, the following nomenclature is intended for channels and outputs: n=A,B,C,D and x=1,…,8 for EK-FG1-TP-I, or n=A,B,C,D,E,F,G,H and x=1,…,16 for EK-FI1-TP-I.

8.3.2.1 Parameters and communication objects

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

Parameter name	*Conditions*	*Settings*
Relay operation	Output channel n = independent or single Ouput x = enabled	**not inverted** inverted
	In the "not inverted" mode, the relay contacts (i.e. the physical output terminals) are shorted when the output is On (active). (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). (x= *1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	
Behaviour at bus on	Output channel n = independent or single Ouput x = enabled	off on **no change**
	Allows to determine the state of the output after bus recovery. (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	
Status feedback telegram	Output channel n = independent or single Ouput x = enabled	disabled / **enabled**
	Enables or disables the output change notification through a bus telegram. *Updating the object from "ON" to "ON" or from "OFF" to "OFF" has no influence on the switching status feedback.* (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	
Transmission cycle time	Output channel n = independent or single Ouput x = enabled Status feedback telegram = enabled	hh:mm:ss (**00:00:00**)
	Frequency with which the output status telegram is sent on the bus. *If set to 00:00:00, the telegram is sent only if there is a change in the status.* (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	
Behaviour after download	Output channel n = independent or single Ouput x = enabled	**Off** on
	Allows to determine the state of the output when the device resumes operation after a new parametrization has been downloaded. (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	

Parameter name	*Conditions*	*Settings*
Delay after bus recovery	Output channel n = independent or single Ouput x = enabled	hh:mm:ss.fff (**00:00:03.000**)
	Time after bus voltage recovery before status feedback telegrams begin to be sent. *The delay has no effect on the behaviour of the outputs; only the feedback telegrams are delayed. The outputs can therefore be activated during the delay after a bus voltage recovery.* *During this delay, no feedback telegram will be transmitted even if a switching occurs; the feedback telegram for a switch during the delay period is lost.* (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	

Parameter name	*Conditions*	*Settings*
Valve function	Output channel n = independent or single Output x = enabled	**Disabled** enabled
	Allows you to enable the valve function on independent output. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Valve position after timeout	Output channel n = independent or single Ouput x = enabled Valve function = enabled	**50 %** [other values in the OFF…100 % range]
	Expresses the valve position, as a percentage of the stroke, in the absence of command update and upon reaching the timeout. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Valve command timeout	Output channel n = independent or single Ouput x = enabled Valve function = enabled	hh:mm:ss (**00:00:00**) [other values in the 00:00:00 … 04:39:37 range]
	Sets the timeout after which the valve is no longer controlled, if no other commands intervene. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
PWM cycle time	Output channel n = independent or single Ouput x = enabled Valve function = enabled	hh:mm:ss (**00:01:00**) [other values in the 00:01:00 … 04:39:37 range]
	Allows you to set the period in which the output is maintained at the ON value for a time proportional to the value of the control variable. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Minimum valve value [%]	Output channel n = independent or single Ouput x = enabled Valve function = enabled	**0 %** [other values in the 0 %…30 % range]
	It allows to adapt the PWM cycle for use with electrothermal actuators with very long opening times. For control variable values lower than the set one, the NC actuator remains unpowered, the NO actuator remains powered. *The set value does not affect the valve position after timeout.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Maximum valve value [%]	Output channel n = independent or single Ouput x = enabled Valve function = enabled	**100 %** [other values in the 70 %…100 % range]
	It allows to adapt the PWM cycle for use with electrothermal actuators with very long opening times. For values of the control variable higher than the set one, the NC actuator remains powered, the NO actuator remains not powered. *The set value does not affect the valve position after timeout.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Parameter name	*Conditions*	*Settings*
On delay time	Output channel n = independent or single Ouput x = enabled Valve function = disabled	hh:mm:ss.ff (**00:00:00.00**) [other values in the 00:00:00.00 … 04:39:37.21 range]
	Delay between the "On" command telegram and the actual output activation. *This time delay does not affect the output of the staircase and forced control functions.* *For the scene function the delay can be set separately.* *Updating the object from "ON" to "ON" or from "OFF" to "OFF" retrigger the delay time.* (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	
Off delay time	Output channel n = independent or single Ouput x = enabled Valve function = disabled	hh:mm:ss.fff (**00:00:00.000**) [other values in the 00:00:00.00 … 04:39:37.21 range]
	Delay between the "Off" command telegram and the actual output deactivation. *Same comments as for the "On delay time" parameter apply.* (n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for* EK-FG1-TP-I*, or x=1,…,16 for* EK-FI1-TP-I).	
Staircase lighting function	Output channel n = independent or single Ouput x = enabled	**disabled** enabled
	Enables or disables the staircase lighting feature. *For further details and parameter descriptions see the corresponding section (par. 6.4.4.4).* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Locking function	Output channel n = independent or single Ouput x = enabled	**disabled** enabled
	Enables or disables the capability of locking the input through a remote command. *For further details and parameter descriptions see the corresponding section (par. 6.4.4.6).* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Forcing function	Output channel n = independent or single Ouput x = enabled	**disabled** enabled
	Enables or disables the capability of forcing the input, i.e. the ability to force a value on the output with precedence over other functions. *For further details and parameter descriptions see the corresponding section (par.6.4.4.7).* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Scenes function	Output channel n = independent or single Ouput x = enabled	**disabled** enabled
	Enables or disables the Scene function. *For further details and parameter descriptions see the corresponding section (par. 6.4.4.8).* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Parameter name	*Conditions*	*Settings*
Operating energy / time counter	Output channel n = independent or single Ouput x = enabled	**disabled** enabled
	Enables or disables the Hour / Energy counter function. *For further details and parameter descriptions see the corresponding section (par. 6.4.4.9).* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output x – Valve continuous command	Output channel n = independent or single valve function = enabled	8 bit unsigned	C–W– –	[5.001] percentage (0…100%)	171, 174, 177, 180, 183, 186, 189, 192, *195, 198, 201, 204, 207, 210, 213, 216.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *This communication object is the direct command for setting the output (x=1,…,8 or x=1,…,16) when the valve function has been enabled.*				

8.3.2.2 Locking function

These parameters are found under the Tab *Channel n (n=A,B,C,D for EK-FG1-TP-I, or n=A,B,C,D,E,F,G,H for EK-FI1-TP-I) – Output x (x=1,…,8 for EK-FG1-TP-I, x=1,…,16 for EK-FI1-TP-I).*

The lock function allows you to inhibit the operation of the output of a specific channel.

For a comprehensive discussion of the lock function, refer to paragraph 6.4.4.6.

Parameter name	*Conditions*	*Settings*
Invert lock device signal	Output channel n = independent or single Ouput x = enabled Locking function = enabled	**not inverted** / inverted
	Allows to set a "lock activate" telegram as "unlock" and vice-versa. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Lock after bus recovery	Output channel n = independent or single Ouput x = enabled Locking function = enabled	unlock lock **no change**
	Defines how to set the lock status after bus voltage recovery. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* • *If llock after bus recovery = yes, the input is always blocked after a bus off/on, regardless of how it was before the bus off;* • *if lock after bus recovery = no, then if it was blocked before the bus off, it remains blocked after the bus on.*	
Behaviour at locking	Output channel n = independent or single Ouput x = enabled Locking function = enabled	off on **no change**
	Defines how to set the output value when the lock is activated. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Behaviour at unlocking	Output channel n = independent or single Ouput x = enabled Locking function = enabled	off on **no change** previous updated value
	Defines how to set the output value when the lock is deactivated. ***Previous*** *is the value that the output had before the lock was activated (see par.6.4.4.10).* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output x – Lock command	Output channel n = independent or single Ouput x = enabled Locking function = enabled	1 bit	C–W– –	[1.003] enable	242, 243, 244, 245, 246, 247, 248, 249, *250, 251, 252, 253, 254, 255, 256, 257.*
	Inhibits the switching commands for the output when an "enable" telegram is received, and unlocks them when a "disable" telegram is received. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				

8.3.2.3 Forcing function

These parameters are found under the Tab *Channel n (n=A,B,C,D for EK-FG1-TP-I, or n=A,B,C,D,E,F,G,H for EK-FI1-TP-I) – Output x (x=1,…,8 for EK-FG1-TP-I, x=1,…,16 for EK-FI1-TP-I).*

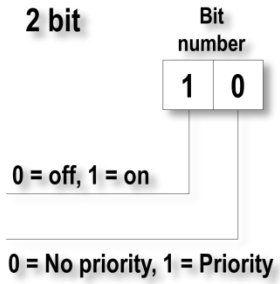
The forcing function allows you to force a value on the output of a specific channel, with precedence over other functions.

For a comprehensive discussion of the forcing function, please refer to paragraph 6.4.4.7.

Parameter name	Conditions	Settings
Forced function - Behaviour end forced control	Output channel n = independent or single Ouput x = enabled Forcing function = enabled	on off **no change** previous
	Allows to determine the state of the output when the forcing is released. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Forced function - Behaviour after bus recovery	Output channel n = independent or single Ouput x = enabled Forcing function = enabled	Not forced Forced up / open Forced down / close Previous **No change**
	Allows to determine the state of the output when the device resumes operation after bus voltage recovery. *Please notice that this is the status of the output, not the forcing status: forcing is maintained over bus failure and bus recovery.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output *x* – Forcing command	Output channel n = independent or single Ouput x = enabled Forcing function = enabled	2 bit	C-W--	[2.001] switch control	266, 267, 268, 269, 270, 271, 272, 273, *274, 275 276, 277, 278, 279, 280, 281.*

(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).
(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).
Allows to force the status of an output.
It is composed of 2 bits: the first one is used for the priority value (i.e. defines whether the forcing is in effect, "Priority", or not) and the second one for the imposed value (which is not considered if forcing is not effective).



8.3.2.4 Staircase lighting function

These parameters are found under the Tab *Channel n (n=A,B,C,D for EK-FG1-TP-I, or n=A,B,C,D,E,F,G,H for EK-FI1-TP-I) – Output x (x=1,…,8 for EK-FG1-TP-I, x=1,…,16 for EK-FI1-TP-I).*

For a comprehensive discussion of the stair light function, refer to paragraph 6.4.4.4.

Parameter name	Conditions	Settings
Staircase lighting time	Output channel n = independent or single Ouput x = enabled Staircase function = enabled	hh:mm:ss (**00:00:30**) [range 00:00:03…04:39:37]
	Duration of staircase lighting time. *This time is the one shown on the time diagram in the descriptive section of this manual as "Ts" in paragraph 6.4.4.4.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Manual off	Output channel n = independent or single Ouput x = enabled Staircase function = enabled	disabled **enabled**
	When enabled, it allows an "Off" command to terminate the lighting time. *The "Off" command can be sent at any time with the same effect, including when the pre-warning is activated.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Parameter name	*Conditions*	*Settings*
Retriggerable	Output channel n = independent or single Ouput x = enabled Staircase function = enabled	disabled **enabled**
	When enabled, it allows a new "On" command to restart the timing. *The "On" command can be sent at any time with the same effect, including when the pre-warning is activated.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Pre-warning	Output channel n = independent or single Ouput x = enabled Staircase function = enabled	disabled **enabled**
	Activates the pre-warning feature. *For a detailed description see in paragraph 6.4.4.4 of this manual.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Pre-warning – Pre-warning time	Output channel n = independent or single Ouput x = enabled Staircase function = enabled Pre-warning = enabled	hh:mm:ss (**00:00:10**) [range 00:00:02…04:39:37]
	Specifies how much time before the end of the timing a pre-warning light interruption will be carried out. *The time interval specified includes the interruption time.* *This time is the one shown on the time diagram in the descriptive section of this manual as "**Tp-w**" in paragraph 6.4.4.4.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Pre-warning – Interruption time	Output channel n = independent or single Ouput x = enabled Staircase function = enabled Pre-warning = enabled	ss.fff (**00.250**) [range 00.250…01.000]
	Specifies the duration of the pre-warning interruption. *This time is the one shown on the time diagram in the descriptive section of this manual as "**Ti**" in paragraph 6.4.4.4.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output *x* – Staircase lighting start stop command	Output channel n = independent or single Ouput x = enabled Staircase function = enabled	1 bit	C-W--	[1.010] start/stop	290, 291, 292, 293, 294, 295, 296, 297, *298, 299, 300, 301, 302, 303, 304, 305.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *Starts the staircase light timing with an On value.* *The timed activation automatically stops at the end of the preset time.* *If "Manual off" is enabled, the communication object will stop the timing with an Off value.*				

8.3.2.5 Scenes function

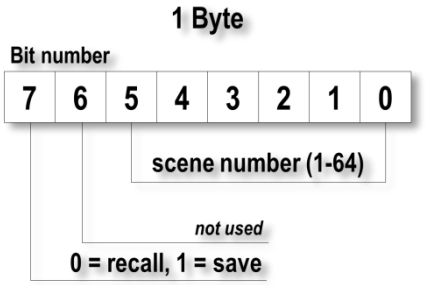
These parameters are found under the Tab *Channel n (n=A,B,C,D for EK-FG1-TP-I, or n=A,B,C,D,E,F,G,H for EK-FI1-TP-I) – Output x (x=1,…,8 for EK-FG1-TP-I, x=1,…,16 for EK-FI1-TP-I).*

For a comprehensive discussion of the scenario function, refer to paragraph 6.4.4.8.

Parameter name	Conditions	Settings
Learning mode	Output channel n = independent or single Ouput x = enabled Scenes function = enabled	**disabled** enabled
	When disabled, the "save scenario" commands are simply ignored and only the values assigned in configuration are used for the scenarios. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Download overwrites learned behavior	Output channel n = independent or single Ouput x = enabled Scenes function = enabled Learning mode = enabled	no / **yes**
	Defines whether the download of a program on the device should erase and overwrite the stored scene output values previously learned and stored in the device. *When the device is put into operation for the first time, this parameter should be set to "yes" (default value) so that the output is initialized with valid scene values. Otherwise, the values are set to "0" (off) for all scenes.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Scene k	Output channel n = independent or single Ouput x = enabled Scenes function = enabled	enabled / **disabled**
	Enables or disables a new scene code to be assigned to the output. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *k=1,…,8.*	

Parameter name	*Conditions*	*Settings*
Scene k – Scene number	Output channel n = independent or single Ouput x = enabled Scenes function = enabled Scene *n* = enabled	1…64 (**1**)
	Scene number to be assigned to the output. The output will respond to scene commands that match the specified number. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *k=1,…,8.*	
Scene k – Activation delay	Output channel n = independent or single Ouput x = enabled Scenes function = enabled Scene *n* = enabled	hh:mm:ss.ff (**00:00:00.0**) [range 00:00:00.0…04:39:37.2]
	Delay between a scene "recall" command and the actual output switching. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *k=1,…,8.*	
Scene *k* – Output behavior	Output channel n = independent or single Ouput x = enabled Scenes function = enabled Scene *n* = enabled	off / **on**
	(Initial) output value for the selected scene. This value will be possibly overwritten by a scene "store" command if the "Learning mode" option is enabled. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *k=1,…,8.*	

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

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output *x* – Scene number	Output channel n = independent or single Ouput x = enabled Scene function = enabled	1 Byte	C-W--	[17.001] scene number [18.001] scene control	314, 315, 316, 317, 318, 319, 320, 321, *322, 323, 324, 325, 326, 327, 328, 329.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *Allows to recall a scene setting for the status of the output, and to store current status in association to the specified scene.* 				

8.3.2.6 Operating energy / Time counter

These parameters are found under the Tab *Channel n (n=A,B,C,D for EK-FG1-TP-I, or n=A,B,C,D,E,F,G,H for EK-FI1-TP-I) – Output x (x=1,…,8 for EK-FG1-TP-I, x=1,…,16 for EK-FI1-TP-I).*

For a comprehensive discussion of the scenario function, refer to paragraph 6.4.4.9.

Parameter name	Conditions	Settings
Output load [W]	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	**50** [W] [range 0…5000]
	Defines the nominal rated power to be considered in computing the accumulated power consumption for the load connected to this output. *The total energy consumed [Wh] is calculated as the product of the specified value [W] and the operating hours [h].* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Energy / time cyclic sending	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	hh:mm:ss (**00:00:00**) [range 00:00:00…04:39:37]
	Defines the time interval for the cyclic retransmission of the counter values (both for accumulated time and energy). *A value of zero (00:00:00) disables cyclic transmission.*	

- During ETS programming or bus voltage failure, the counter stops counting.

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output *x* – Energy counter (Wh)	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	4-byte signed value	CR-T-	[13.013] active energy [kWh]	338, 344, 350, 356, 362, 368, 374, 380, *386, 392, 398, 404, 410, 416, 422, 428.*
	Stores the current counter value of the accumulated energy. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				
Output *x* - Energy counter reset command	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	1 bit	C-W--	[1.015] reset	339, 345, 351, 357, 363, 369, 375, 381, *387, 393, 399, 405, 411, 417, 423, 429.*
	Resets the energy counter to 0. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				
Output *x* - Hours counter	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	2-byte unsigned value	CR-T-	[7.007] time [h]	340, 346, 352, 358, 364, 370, 376, 382, *388, 394, 400, 406, 412, 418, 424, 430.*
	Stores the current value of the operating time counter, expressed in hours. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				
Output *x* - Hours counter reset	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	1 bit	C-W--	[1.015] reset	341, 349, 353, 361, 365, 373, 377, 385, *389, 397, 401, 409, 413, 421, 425, 433.*
	Resets the operating hour counter to 0. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output *x* - Hours counter overflow	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	1 bit	CR-T-	[1.005] alarm	342, 347, 354, 359, 366, 371, 378, 383, *390, 395, 402, 407, 414, 419, 426, 431.*
	1-bit alarm sent when the time counter reaches the maximum value of 65535 hours. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				
Output *x* - Operating seconds counter	Output channel n = independent or single Ouput x = enabled Operating energy/time counter = enabled	4-byte signed value	CR-T-	[13.100] time lag [s]	343, 348, 355, 360, 367, 372, 379, 384, *391, 396, 403, 408, 415, 420, 427, 432.*
	Stores the current value of the operating time counter, expressed in seconds. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*				

8.3.3 Coupled outputs: channel *n* configuration

This paragraph lists the configuration parameters of the output channels when they are used as coupled outputs.

In general, the following nomenclature is intended for channels and outputs: n=A,B,C,D for EK-FG1-TP-I, or n=A,B,C,D,E,F,G,H for EK-FI1-TP-I.

For all the items in this section, the condition *"Output channel n = 3-way valve/Shutter/Venetian blind"* is implicitly assumed but not indicated, for greater clarity.

8.3.3.1 Parameters and communication objects

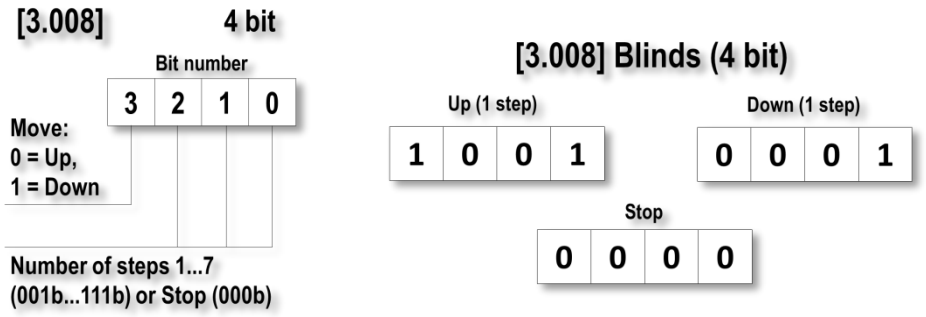
In this section most of the configurable parameters for the output are listed.

Parameter name	*Conditions*	*Settings*
Relay operation	Use = all	**Not inverted** inverted
Status feedback telegram	Use = all	Disabled **enabled**
Reversion pause time	Use = all	hh:mm:ss.fff **00:00:00.300** [range 00:00:00.000 … 04:39:37.215]
	The minimum pause time between contact activation when switching from one output to another.	
Open time	Use = all	hh:mm:ss (**00:00:15**) [range 00:00:00 … 04:39:37]
	The time for the actuator to run the full stroke between the endpoints, in the opening direction. *It is important that the specification of this time is particularly accurate, since the accuracy of positioning depends heavily on it.*	
Close time	Use = all	hh:mm:ss (**00:00:15**) [range 00:00:00 … 04:39:37]
	The time for the actuator to run the full stroke between the endpoints, in the closing direction. *It is important that the specification of this time is particularly accurate, since the accuracy of positioning depends heavily on it.*	
Position control with dimmer	Use = all	**no** / yes
	If this option is selected, a dimmer-type communication object is made available for the control of the actuator. It can be used, as an alternative, at the same time as the other standard control mechanisms.	

Parameter name	*Conditions*	*Settings*
Enable extra time	Use = all	Disabled **enabled**
	Enable or disable additional time to reach the full scale of the movement.	
Time over full scale when moving up / opening	Use = all Enable extra time = enabled	0…60 s **(3 s)**
	Allows the user to set the additional time period for reaching the full scale of the travel during move up/opening.	
Time over full scale in descent / closing	Use = all Enable extra time = enabled	0…60 s **(3 s)**
	Allows the user to set the additional time period for reaching the full scale of the travel during move down/closing.	
Number of slat steps	Use = venetian blind	1…64 **5**
	Number of steps necessary for the motorized actuator to run the slats over the full movement between the endpoints.	
Opening time of the slat step	Use = venetian blind	00:00:00.000…04:39:37.215 hh:mm:ss.fff (**100 ms**)
	Output activation time corresponding to the elementary step of desired amplitude for the opening of the slats.	
Closing time of the slat step	Use = venetian blind	00:00:00.000…04:39:37.215 hh:mm:ss.fff (**100 ms**)
	Output activation time corresponding to the elementary step of desired amplitude for the closing of the slats.	
Slats control with dimmer	Use = venetian blind	**no** / yes
	If this option is selected, a dimmer-type communication object is made available for the control of the actuator. It can be used, as an alternative, at the same time as the other standard control mechanisms.	
Restore slats position	Use = venetian blind	**No** / Yes
	If this option is selected, after an ascent, descent, or absolute position command, the slats return to their previous position. If not selected, they remain in the position they assumed during movement.	
Behaviour at bus on	Use = all	**No change** up / open down / close move to position
	Allows to determine the state of the output after bus recovery.	
Delay after bus recovery	Use = all	00:00:00.00…04:39:37.21 hh:mm:ss.ff (**00:00:03.00**)
	Time, starting from bus recovery, after which feedback telegrams can start to be transmitted. *The delay has no effect on the behavior of the outputs, but only affects the transmission of telegrams. The outputs can therefore be activated during the delay time.* *The transmission of telegrams is not delayed but inhibited; for any switching that occurs during the delay time, no feedback is therefore generated.*	
Locking function	Use = all	enabled / **disabled**

Parameter name	*Conditions*	*Settings*
	Enables or disables the capability of locking the input through a remote command. *For further details and parameter descriptions see the corresponding section 6.4.5.5*	
Forcing function	Use = all	enabled / **disabled**
	Enables or disables the capability of forcing the input through a remote command. *For further details and parameter descriptions see the corresponding section 6.4.5.6.*	
Scenes function	Use = all	enabled / **disabled**
	Enables or disables the Scene function. *For further details and parameter descriptions see the corresponding section 6.4.5.7.*	
Meteo alarms	Use = shutter or venetian blind	enabled / **disabled**
	Enables or disables the meteo alarms management function. *For further details and parameter descriptions see the corresponding section 6.4.5.8.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output channel *n* – Move up-down command	Output channel n = 3-way valve / shutter / venetian blind Use = shutter or venetian blind	1 bit	C-W--	[1.008] up/down	2, 16, 30, 44, *58, 72, 86, 100.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Continuous motion command object: if received, starts motion in the specified direction.*				
Output channel *n* – Stop-step up-down command	Output channel n = 3-way valve / shutter / venetian blind Use = shutter or venetian blind	1 bit	C-W--	[1.007] step	3, 17, 31, 45, *59, 73, 87, 101.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Command object for step motion: if received (and there are no movements in progress), starts a movement of a predetermined duration in the specified direction. If the actuator is already moving, instead, stops the movement in progress.*				
Output channel *n* – Dedicated Stop command	Output channel n = 3-way valve / shutter / venetian blind Use = all	1 bit	C-W--	[1.017] trigger	4, 18, 32, 46, *60, 74, 88, 102.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Upon receipt, stops the current movement.*				
Output channel *n* – Info move	Output channel n = 3-way valve / shutter / venetian blind Use = all	1 bit	CR-T-	[1.011] state	5, 19, 33, 47, *61, 75, 89, 103.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Returns information about the current direction of movement.*				
Output channel *n* – Valid current abs position	Output channel n = 3-way valve / shutter / venetian blind Use = all	1 bit	CR-T-	[1.002] boolean	6, 20, 34, 48, *62, 76, 90, 104.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Indicates that the actuator has reached the requested absolute position. Transmitted following absolute positioning commands.*				

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output channel n – Dimmer blind position command	Use = venetian blind Position control with dimmer = yes	4 bit	C-W--	[3.008] blind control	7, 21, 35, 49, *63, 77, 91, 105.*
	Allows to command the actuator through a dimmer-style command. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* 				
Output channel n – Dimmer shutter position command	Use = shutter Position control with dimmer = yes	4 bit	C-W--	[3.008] blind control	7, 21, 35, 49, *63, 77, 91, 105.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).				
Output channel n – Dimmer valve position command	Use = valve Position control with dimmer = yes	4 bit	C-W--	[3.008] blind control	7, 21, 35, 49, *63, 77, 91, 105.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).				
Output channel *n* – Abs [valve / shutter / blind] position command	Output channel n = 3-way valve / shutter / venetian blind Use = all	1 bit	C-W--	[5.001] percentage (0..100%)	8, 22, 36, 50, *64, 78, 92, 106.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Sets the absolute position to reach and starts the movement in the appropriate direction.* *In the case of Venetian blinds, the absolute position to which it refers is that of the panel.*				
Output channel *n* – Abs [valve / shutter / blind] position status	Output channel n = 3-way valve / shutter / venetian blind Use = all	1 bit	CR-T-	[5.001] percentage (0..100%)	9, 23, 37, 51, *65, 79, 93, 107.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Returns the current absolute position of the actuator.* *The absolute position is calculated based on the sequence of requested movements and is realigned whenever the actuator reaches a limit switch.* *In the case of Venetian blinds, the absolute position to which it refers is that of the panel.*				
Output channel n – Dimmer slats command	Use = venetian blind Slats control with dimmer = yes	4 bit	C-W--	[3.008] blind control	10, 24, 38, 52, *66, 80, 94, 108.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Allows to command the slats position through a dimmer-style command.* *See previous entry for bit field details.*				

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output channel *n* – Abs slats position command	Output channel n = 3-way valve / shutter / venetian blind **Use = venetian blind**	1 bit	C-W--	[5.001] percentage (0..100%)	11, 25, 39, 53, *67, 81, 95, 109.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Sets the absolute position to be reached for the slats and starts the movement in the appropriate direction.*				
Output channel *n* – Abs slats position status	Output channel n = 3-way valve / shutter / venetian blind **Use = venetian blind**	1 bit	CR-T-	[5.001] percentage (0..100%)	12, 26, 40, 54, *68, 82, 96, 110.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Returns the current absolute position of the slats.* *The absolute position is calculated based on the sequence of requested movements and is realigned whenever the slats reach the end of their travel; this happens when the uninterrupted duration of movement in one direction is at least equal to the full travel time specified as a parameter.*				
Output channel *n* - Direction status output x	Output channel n = 3-way valve / shutter / venetian blind AND Status feedback telegram = enabled	1 bit	CR-T-	[1.001] switch	172, 175, 178, 181, 184, 187, 190, 193, *196, 199, 202, 205, 208, 211, 214, 217.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).* *Sent at each output state change.*				

8.3.3.2 Locking function

Parameter name	Conditions	Settings
Lock device signal	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Locking function = enabled	**not inverted** / inverted
	Allows to interpret a "lock activate" telegram as unlock and vice-versa. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).*	
After bus recovery	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Locking function = enabled	unlock lock **no change**
	Defines how to set the lock status after bus voltage recovery. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).*	

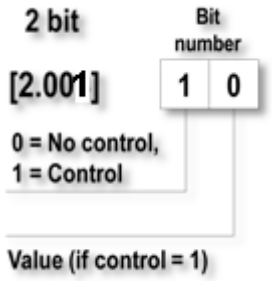
Parameter name	*Conditions*	*Settings*
Behaviour at locking	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Locking function = enabled	**no change** up / open down / close stop move to position
	Defines how to set the output value when the lock is activated. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).*	
Behaviour at unlocking	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Locking function = enabled	**no change** up / open down / close stop move to position previous updated value
	Defines how to set the output value when the lock is deactivated. *For the meaning of the "previous" option, see the details in par. 6.4.4.10.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* ***Previous** is the value that the output had when the block was activated.* ***Updated value** behaves as follows:* *1) If during the block the user makes an up or down movement, when the block is removed the shutter/venetian blind/valve is positioned at the bottom of the scale fully up or down, depending on the direction set during the block;* *2) If during the block the user sends an absolute position command, for example 50%, when the block is removed the actuator moves exactly to that percentage value.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output channel n – Lock command	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Locking function = enabled	1 bit	C–W– –	[1.003] enable	242, 244, 246, 248, *250, 252,* *254, 256.*
	Inhibits the switching commands for the output when an "enable" telegram is received, and unlocks them when a "disable" telegram is received. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).*				

8.3.3.3 Forcing function

Parameter name	*Conditions*	*Settings*
Forced function - Behaviour on forced control end	Output channel n = independent or single Ouput x = enabled Forcing function = enabled	**no change** up / open down / close stop move to position previous

Parameter name	Conditions	Settings
	Allows to determine the state of the output when the forcing is released. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	
Forced function - Behaviour after bus recovery	Output channel n = independent or single Ouput x = enabled Forcing function = enabled	Not forced Forced up / open Forced down / close Previous **no change**
	Allows to determine the state of the output when the device resumes operation after bus voltage recovery. *Please notice that this is the status of the output, not the forcing status: forcing is maintained over bus failure and bus recovery.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output channel n – Forcing command	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Forcing function = enabled	2 bit	C–W– –	[2.001] switch control	266, 268, 270, 272, *274, 276, 278, 280.*
	Allows to force the status of an output pair. The command is a "direction control" telegram, which can force movement in one direction, the other, or release forcing. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).*  				

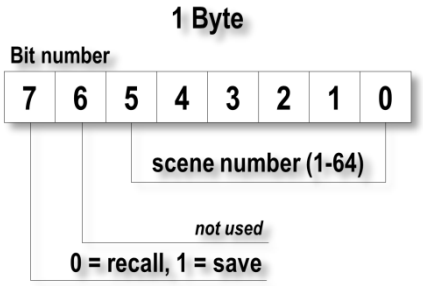
8.3.3.4 Scenes function

Parameter name	Conditions	Settings
Learning mode	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled	**disabled** enabled
	When disabled, the "save scenario" commands are simply ignored and only the values assigned in configuration are used for the scenarios. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *(x= 1,…,8 for EK-FG1-TP-I, or x=1,…,16 for EK-FI1-TP-I).*	

Parameter name	*Conditions*	*Settings*
Download overwrites learned behavior	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled Learning mode = enabled	no / **yes**
	Defines whether the download of a program on the device should erase and overwrite the stored scene output values previously learned and stored in the device. *When the device is put into operation for the first time, this parameter should be set to "yes" (default value) so that the output is initialized with valid scene values. Otherwise, the values are set to "0" (off) for all scenes.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).*	
Scene *k*	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled	enabled / **disabled**
	Enables or disables a new scene code to be assigned to the output. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *k=1,…,8.*	
Scene *k* – Scene number	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled Scene *n* = enabled	1…64 (**1**)
	Scene number to be assigned to the output. The output will respond to scene commands that match the specified number. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *k=1,…,8.*	
Scene *k* – Activation delay	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled Scene *n* = enabled	hh:mm:ss.f (**00:00:00.0**) [range 00:00:00.0…04:39:37.2]
	Delay between a scene "recall" command and the actual output switching. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *k=1,…,8.*	
Scene *k* – Output behaviour	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled Scene *n* = enabled	**Stop** Open Close Move to position
	Value to assign to the output for the selected scene. *This is an initialization value that can remain fixed or, if the learning mode is enabled, be overwritten by a "save scene" command.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *k=1,…,8.*	
Scene *k* - position	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled Scene *n* = enabled Output behaviour = move to position	**0** [0…100%]

Parameter name	*Conditions*	*Settings*
	Absolute position that the actuator must reach for the selected scene. *This is an initialization value that can remain fixed or, if the learning mode is enabled, be overwritten by a "save scene" command.* *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *k=1,…,8.*	
Scene *k* – slats position	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scenes function = enabled Scene *n* = enabled Output behaviour = move to position	**0** [0…100%]
	Absolute position to assign to the slats for the selected scene. This is an initialization value that can remain fixed or, if the learning mode is enabled, be overwritten by a "save scene" command. *(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I).* *k=1,…,8.*	

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

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output channel n – Scene number	Output channel n = 3-way valve/Shutter/Venetian blind Use = all Scene function = enabled Scene *k* = enabled	1 Byte	C–W––	[17.001] scene number [18.001] scene control	314, 316, 318, 320, *322, 324,* *326, 328.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *k=1,…,8.* *Allows to recall a scene setting for the status of the output, and to store current status in association to the specified scene.*  				

8.3.3.5 Meteo alarms

Parameter name	*Conditions*	*Settings*
Reaction to [wind / frost / rain]	Meteo alarms = enabled	**none** Up / Open Down / Close
	Defines the movement that the actuator must perform when the corresponding alarm is activated.	
[wind / frost / rain] timeout	Meteo alarms = enabled	0…65535 [minutes] (**10 Min.**)
	Defines the timeout interval for the heartbeat function. *If a timeout is set, "Alarm not active" telegrams must be sent regularly by the source with a period shorter than the specified timeout, to prove that alarm communication is occurring properly. If an "Alarm not active" telegram is not received for a period longer than the timeout, the alarm condition is activated.* *A null value (0) disables the heartbeat function.*	
Action at the end of the alarm	Meteo alarms = enabled	**none** Up / Open Down / Close Previous
	Defines the movement that the actuator must perform when all alarms cease.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output channel *n* – Wind alarm	Meteo alarms = enabled	1 bit	C–W– –	[1.005] alarm	13, 27, 41, 55, *69, 83, 97, 111.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *Enable meteo alarm reception. If reception is enabled, the alarm will be activated by writing an "alarm active" command, and will be deactivated by writing a "cancel alarm" command.* *If using the Alarm Heartbeat feature (see section 6.4.5.8), the "cancel alarm" command must be sent regularly at intervals shorter than the maximum programmed timeout.*				
Output channel *n* – Frost alarm	Meteo alarms = enabled	1 bit	C–W– –	[1.005] alarm	14, 28, 42, 56, *70, 84, 98, 112.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *See previous parameter.*				
Output channel *n* – Rain alarm	Meteo alarms = enabled	1 bit	C–W– –	[1.005] alarm	15, 29, 43, 57, *71, 85, 99, 113.*
	(n = A,B,C,D for EK-FG1-TP-I, or n = A,B,C,D,E,F,G,H for EK-FI1-TP-I). *See previous parameter.*				

8.4 Input configuration

The ***Input configuration*** tab allows you to configure each input channel independently or in pairs. It contains the following parameters:

- Input channel A
  - Input 1
    - Type
  - Input 2
    - Type

…

- Input channel D
  - Input 7
    - Type
  - Input 8
    - Type
-

…

- Input channel E, F, G, H (only for EK-FI1-TP-I)
  - Input 9, 11, 13, 15
    - Type
  - Input 10, 12, 14, 16
    - Type

8.4.1 Input type

8.4.1.1 Parameters and communication objects

Parameter name	***Conditions***	***Settings***
Input channel x	-	disabled **independent or single** coupled
	x = A, B, C, D (E, F, G, H for EK-FI1-TP-I). *Sets the operating mode for the inputs associated with Channel x.*	
Input xx (xx = 1, …, 8 for EK-FG1-TP-I) (xx = 1, …, 16 for EK-FI1-TP-I)	**Input channel x = independent or single**	Disabled / **digital input**
	x = A, B, C, D (E, F, G, H for EK-FI1-TP-I). *Enables or disables a channel input as a digital input.*	
Type	**Input channel x = independent or single** **Input xx = digital input**	**Switching** Dimming Shutter or venetian blind Scene shift register

Parameter name	*Conditions*	*Settings*
	x = A, B, C, D (E, F, G, H for EK-FI1-TP-I). *(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).* *Sets the functionality associated with the input of the channel* *Additional parameters for the selected function appear in the individual configuration of the single channel/input (see following paragraphs).*	
Input xx (xx = 1, …, 8 for EK-FG1-TP-I) (xx = 9, …, 16 for EK-FI1-TP-I)	**Input channel x = independent or single**	Disabled / **digital input** / NTC
	x = A, B, C, D (E, F, G, H for EK-FI1-TP-I). *Enables or disables a channel input as a digital input or NTC sensor.* *The option to enable an input as an NTC sensor applies to:* • *All inputs, for EK-FG1-TP-I;* • *Only inputs 9 to 16, for EK-FG1-TP-I;*	
Type	**Input channel x = coupled**	Switching **Dimming** Shutter or venetian blind
	x = A, B, C, D (E, F, G, H for EK-FI1-TP-I). *Sets the functionality associated with the two coupled inputs of the channel.* *Additional parameters for the selected function appear in the individual configuration of each input (see following paragraphs).*	

8.4.2 Independent or single - Digital input - switching

8.4.2.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Contact type	Input xx = digital input Type = switching	**NO (normally open)** NC (normally closed)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the idle state.*	
Number of communication objects	Input xx = digital input Type = switching	1…8 (**1**)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Number of communication objects to be associated with the button event.*	
Lock function	Input xx = digital input Type = switching	**disable** enable
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Enables or disables the lock function.*	
Lock function – Invert lock device signal	Input xx = digital input Type = switching Lock function = enabled	**Not inverted** Inverted

Parameter name	*Conditions*	*Settings*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows you to interpret an "enable lock" command code as "disable lock" and vice versa.*	
Lock function – Lock after bus recovery	Input xx = digital input Type = switching Lock function = enabled	**No** Yes
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	
Lock function – Behaviour at locking	Input xx = digital input Type = switching Lock function = enabled	**None** as closed or short press as open or long press
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the operation to be performed when locking occurs. The choice is between the operations associated with the two possible events of closing (or short press, depending on the configuration) or opening (or long press).*	
Lock function – Behaviour at unlocking	Input xx = digital input Type = switching Lock function = enabled	**None** as closed or short press as open or long press
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the operation to be performed upon unlocking. The choice is between the operations associated with the two possible events of closing (or short press, depending on the configuration) or opening (or long press).*	
Number of communication objects	Input xx = digital input Type = switching	1…8 (**1**)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Number of communication objects to be associated with the contact closure event.*	
Event	Input xx = digital input Type = switching	**activation / release** short / long press
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Type of event to be used as an action trigger.*	
Long press tume	Input xx = digital input Type = switching Event = short/prolonged press	hh:mm:ss.fff (**00:00:00.800**)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Minimum pressure holding time to distinguish between short and long pressure.*	
Object *n* – communication object dimension	Input xx = digital input Type = switching	**1 bit value** 2 bit value 1 byte scaling 1 byte signed value 1 byte unsigned value 2 bytes signed value 2 bytes unsigned value 2 byte floating value
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the type and size of individual communication objects.* *n = 1,…,8.*	
Object *n* – Send delay	Input xx = digital input Type = switching	hh:mm:ss.ff (**00:00:00.00**)

Parameter name	*Conditions*	*Settings*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Delay between the event and the transmission of the value on the bus.* *By defining an individual delay for each object, it is possible to form a defined sequence of telegrams to be associated with the event.* *n = 1,…,8.*	
Object *n* – Send cyclically	Input xx = digital input Type = switching Number of communication objects ≥ 1	None **off / value 1** on / value 2 off and on / both values
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines which values, if any, should be transmitted periodically if they are in the activated state.* *Cyclical transmission is only available if the number of communication objects to be associated is 1.* *n = 1,…,8.*	
Object *n* – Cyclical sending interval	Input xx = digital input Type = switching Number of communication objects ≥ 1 Send cyclically ≠ none	hh:mm:ss (**00:02:00**) [range 00:00:10 … 04:39:37]
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Interval between periodic retransmissions.* *n = 1,…,8.*	
Object *n* – Reaction to activation *or* Reaction to short press	Input xx = digital input Type = switching o.c. size = 1 bit	none on off **toggle**
	Input xx = digital input Type = switching o.c. size = 2 bits	none disable enable off/up **enable on/down** enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Input xx = digital input Type = switching o.c. size = *all byte values*	none **send value 1** send value 2 send value 1 ↔ send value 2
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Value change triggered by a Closed or Short Press event (depending on the event configuration).* *n = 1,…,8.*	
Object *n* – Reaction to release *or* Reaction to prolonged press	Input xx = digital input Type = switching o.c. size = 1 bit	**none** on off toggle
	Input xx = digital input Type = switching o.c. size = 2 bits	none **disable** enable off / up enable on / down enable off / up ↔ disable enable on / down ↔ disable enable off / up ↔ enable on / down
	Input xx = digital input Type = switching o.c. size = *all byte values*	**none** send value 1 send value 2 send value 1 ↔ send value 2

Parameter name	*Conditions*	*Settings*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Value change triggered by an Open or Long press event (depending on the event configuration).* *n = 1,…,8.*	
Object *n* – Value 1	Input xx = digital input Type = switching o.c. size = *all byte values*	0…100 (1 byte scaling) -128...127 (1 byte signed) 0…255 (1 byte unsigned) -32,768... 32,767 (2 bytes signed) 0...65535 (2 bytes unsigned) -671088.64...670760.96 (2 bytes floating)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *First value configured for association with events.* *n = 1,…,8.*	
Object *n* – Value 2	Input xx = digital input Type = switching o.c. size = *all byte values*	*as for value 1*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Second value configured for association with events.* *n = 1,…,8.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – switching Status *[type]*, object *no.*	Input *xx* = digital input Type = switching	*depending on configuration* (**1-bit**)	C-WTU	*depending on configuration* (**[1.001] switch**)	749, 757, 765, 773, 781, 789, 797, 805 *813, 821, 829, 837, 845, 853, 861, 869*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Up to 8 objects can be defined to be associated with the same event (n = 1,…,8).* *The OC numbers listed refer to the first of these 8 objects (for each of the inputs); the OCs of the subsequent objects are sequential. To obtain the OC number for the nth object, simply add (n-1) to the numbers shown.* *E.g.: the OCs associated with input 3 have numbers starting from 765. The number of the 5th OC associated with this input will therefore be 765 + (5-1) = 769.* ***The types and sizes of individual objects can be configured as described below.***				
Input *xx* – Lock command	Input *xx* = digital input Type = switching Lock function = enabled	1 bit	C-W-U	[1.003] enable	653, 659, 665, 671, 677, 683, 689, 695 *701, 707, 713, 719, 725, 731, 737, 743*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).				

The data dimensions and *Data Point Types* are as follows:

Dimension	*DPT*
1 bit	[1,001] switch
2 bits	[2.*] 1-bit control
1 byte unsigned	[4.*] character [5.*] 8-bit unsigned value [20.*] 1 byte

Dimension	*DPT*
1 byte percentage	[4.*] character [5.*] 8-bit unsigned value [20.*] 1-byte
1 byte with sign	[6.*] 8-bit signed value
2 bytes without sign	[7.*] 2-byte unsigned value
2 bytes with sign	[8.*] 2-byte signed value
2 bytes floating point	[9.*] 2-byte floating value

Table 1 - Sizes and Data Types for Communication Objects

Note on the 2-byte floating point value range

When the data is of the *2-byte floating point* type, the value range is between -273...670760.96, instead of between -671088.64...670760.96. The reason for this is that by default it is considered as a temperature range in °C, so the lower limit is absolute zero.

To obtain negative values < -273, open the *Properties* menu of the communication object from the Group Objects tab and manually change the DPT, choosing the most suitable one from those available for group 7.* 2-byte unsigned value, as shown in the example at Figure 38.

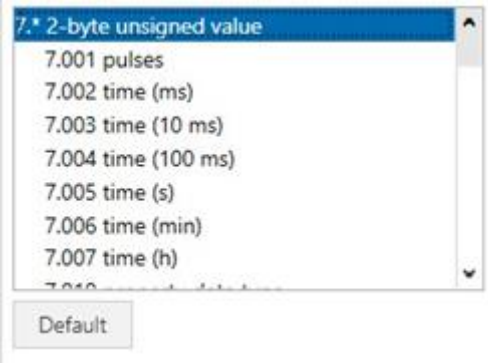


Figure 38 - DPT for group 7.* 2-bytes floating point

8.4.3 Independent or single - Digital input - dimming

8.4.3.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Contact type	Input xx = digital input Type = dimming	**NO (normally open)** NC (normally closed)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the idle state.*	
Lock function	Input xx = digital input Type = dimming	**disable** enable
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Enables or disables the lock function.*	

Parameter name	*Conditions*	*Settings*
Long press time	Input xx = digital input Type = dimming	hh:mm:ss.fff (**00:00:03.000**) [range 00:00:00.215 04:39:37.215]
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Minimum pressure holding time to distinguish between short and long pressure.*	
Toggle mode	Input xx = digital input Type = dimming	**enabled**/disabled
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *When enabled, a short press reverses the on/off state of the associated communication object; otherwise, a short press is associated with a fixed state between the two.*	
Reaction to long press	Input xx = digital input Type = dimming Toggle mode = enabled	**darker** brighter darker ↔ brighter
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the function to be assigned to a long press.* *If* Toggle mode *is enabled, the Toggle function is already assigned to short press.*	
Short/long (press) action	Input xx = digital input Type = dimming Toggle mode = disabled	**off / darker** on / brighter off / darker ↔ brighter on / darker ↔ brighter
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the function to be assigned to short and long presses.*	
Cyclic sending	Input xx = digital input Type = dimming	**none** off / value 1 on / value 2 both off and on / both values
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines which values, if any, should be transmitted periodically if they are in the activated state.*	
Cyclical sending interval	Input xx = digital input Type = dimming Cyclical sending ≠ none	hh:mm:ss (**00:02:00**) [range 00:00:10 … 04:39:37]
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Interval between periodic retransmissions.*	
Lock function – Invert lock device signal	Input xx = digital input Type = dimming Lock function = enabled	**not inverted** / inverted
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows a "lock active" command code to be interpreted as "lock deactivated" and vice versa.*	
Lock function – Lock after bus recovery	Input xx = digital input Type = dimming Lock function = enabled	**no** / yes
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	
Lock function – Behaviour at locking	Input xx = digital input Type = dimming Lock function = enabled	**None** off on toggle
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the status to be set for the communication object at the block input.*	

Parameter name	*Conditions*	*Settings*
Lock function – Behaviour at unlocking	Input xx = digital input Type = dimming Lock function = enabled	**None** off on as previous
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the status to be set for the communication object upon unlocking.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Switch command	Input xx = digital input Type = dimming	1 bit	C-WTU	[1.001] switch	654, 660, 666, 672, 678, 684, 690, 696 *702, 708, 714, 720, 726, 732, 738, 744.*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Sends an on/off command to a dimmer actuator.* *The command is sent following a short contact closure event.* *The value sent can be a fixed value or alternate between the two possible values each time it is activated.*				
Input *xx* – Dimming up / down / stop command	Input xx = digital input Type = dimming	4 bits	C--T-	[3.*] 3-bit control	655, 661, 667, 673, 679, 685, 691, 697 *703, 709, 715, 721, 727, 733, 739, 745.*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Sends a command to change the intensity (increase or decrease) to a dimmer actuator.* *Three values are used, corresponding to the start increase, start decrease and stop variation commands.* Increase 1 0 0 0 Decrease 0 0 0 1 Stop dimming 0 0 0 0 *The increase/decrease commands are sent following a long closure; the stop command is sent following the opening of the contact.* *The value sent upon activation can be a fixed value or alternate between the two possible values (increase/decrease) upon each activation.*				
Input *xx* – Lock command	Input xx = digital input Type = dimming Lock function = enabled	1 bit	C-W--	[1.003] enable	653, 659, 665, 671, 677, 683, 689, 695 *701, 707, 713, 719, 725, 731, 737, 743.*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).				

8.4.4 Independent or single - Digital input - shutters or venetian blinds

8.4.4.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Contact type	Input xx = digital input Type = Shutter or venetian blind	**NO (normally open)** NC (normally closed)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the idle state.*	
Lock function	Input xx = digital input Type = Shutter or venetian blind	**disable** enable
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Enables or disables the lock function.*	
Long press time	Input xx = digital input Type = Shutter or venetian blind	hh:mm:ss.fff (**00:00:00.800**) [range 00:00:00.215 04:39:37.215]
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Minimum pressure holding time to distinguish between short and long pressure.*	
Toggle mode	Input xx = digital input Type = Shutter or venetian blind	**Disabled** Enabled
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *When enabled, briefly pressing the button reverses the direction of movement; otherwise, briefly pressing the button assigns a fixed value between the two.*	
Up / down action	Input xx = digital input Type = Shutter or venetian blind Toggle mode = disabled	**Up/open** Down/close
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the direction of movement to be associated with the closing of the contact.*	
Venetian blind mode	Input xx = digital input Type = Shutter or venetian blind	enabled/**disabled**
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *If Venetian blind mode is enabled, the device sends "full movement" commands following prolonged contact closure, and "step" commands following brief closure;* *if it is disabled, it still sends "full movement" commands when the contact is closed for a prolonged period, but "stop" commands when the contact is closed briefly.*	
Lock function – Invert lock device signal	Input xx = digital input Type = Shutter or venetian blind Lock function = enabled	**not inverted** / inverted
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows a "lock active" command code to be interpreted as "lock inactive" and vice versa.*	
Lock function – Lock after bus recovery	Input xx = digital input Type = Shutter or venetian blind Lock function = enabled	**no** / yes
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	
Lock function – Behaviour at locking	Input xx = digital input Type = Shutter or venetian blind Lock function = enabled	**None** Up Down

Parameter name	Conditions	Settings
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the action to be performed when locking occurs.*	
Lock function – Behaviour at unlocking	Input xx = digital input Type = Shutter or venetian blind Lock function = enabled	**None** Up Down
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the action to be performed when unlocked.*	

Object name	Conditions	Size	Flags	DPT	CO number(s)
Input xx - Dedicated stop command	Input xx = digital input Type = Shutters or venetian blinds Venetian blind mode = disabled	1 bit	C- -T-	[1.017] trigger	654, 660, 666, 672, 678, 684, 690, 696 *702, 708, 714, 720,* *726, 732, 738, 744.*
	Immediately stops any movement of the roller shutter. The object is sent to short closing if "Venetian blind" mode is disabled, or to long closing if "Venetian blind" mode is enabled. *(xx = 1, ..., 8 for EK-FG1-TP-I, or xx = 1, ..., 16 for EK-FI1-TP-I).*				
Input xx - Stop/step up/down command	Input xx = digital input Type = Shutters or venetian blinds Venetian blind mode = enabled	1 bit	C--T-	[1.007] step	654, 660, 666, 672, 678, 684, 690, 696 *702, 708, 714, 720,* *726, 732, 738, 744*
	Moves the roller shutter to the fully open or closed position. *The object is sent at the end of a long closing.* *(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).*				
Input xx - Move up / down command	Input xx = digital input Type = Shutters or venetian blinds	1 bit	C--T-	[1.008] up/down	656, 662, 668, 674, 680, 686, 692, 698 *704, 710, 716, 722,* *728, 734, 740, 746.*
	Opens or closes the roller shutter in steps. *The object is sent to the short contact.* *(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).*				
Input xx - Lock command	Input xx = digital input Type = Shutters or venetian blinds Lock function = enabled	1 bit	C-W--	[1.003] enable	653, 659, 665, 671, 677, 683, 689, 695 *701, 707, 713, 719,* *725, 731, 737, 743.*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).				

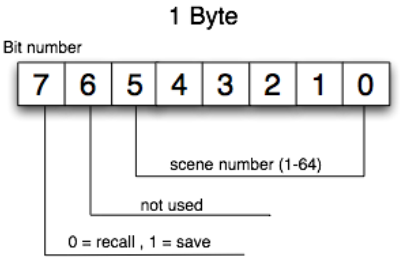
8.4.5 Independent or single - Digital input – Scene

8.4.5.1 Parameters and communication objects

Parameter name	Conditions	Settings
Contact type	Input xx = digital input Type = scene	**NO (normally open)** NC (normally closed)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the idle state.*	

Parameter name	*Conditions*	*Settings*
Lock function	Input xx = digital input Type = scene	**disabled** enabled
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Enables or disables the lock function.*	
First scene number	Input xx = digital input Type = scene	1…64 (**1**)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Number of the main scene to be assigned to the input.* *It is referred to as "first" because a second alternative scene number can be defined.*	
Learning mode	Input xx = digital input Type = scene	**Disabled** Enabled
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *When enabled, it allows you to activate the storage of the current configuration with the assigned scenario number by pressing and holding.*	
Long press time	Input xx = digital input Type = scene Learning mode = enabled	hh:mm:ss.fff (**00:00:00.800**) [range 00:00:00.215 04:39:37.215]
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Minimum pressure maintenance time to distinguish between short and long pressure.*	
Scene activation	Input xx = digital input Type = scene Learning mode = disabled	**send first scene only** Toggle between 2 scenes
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows the input contact to be used to alternate between two different scenes.*	
Second scene number	Input xx = digital input Type = scene Learning mode = disabled Scenario activation = Toggle between 2 scenarios	1…64 (**2**)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Number of selectable alternative scenes.*	
Lock function – Invert lock device signal	Input xx = digital input Type = scene Lock function = enabled	**not inverted** / inverted
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows a "lock active" code from a command to be interpreted as "lock deactivated" and vice versa.*	
Lock function – Lock after bus recovery	Input xx = digital input Type = scene Lock function = enabled	**no** / yes
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	
Lock function – Behaviour at locking	Input xx = digital input Type = scene Lock function = enabled	**None** send first scenario send second scenario
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the action to be performed when the lock is activated.*	

Parameter name	*Conditions*	*Settings*
Lock function – Behaviour at unlocking	Input xx = digital input Type = scene Lock function = enabled	**None** send first scenario Send second scenario
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Defines the action to be performed upon unlocking.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Scene number	Input xx = digital input Type = scene	1 Byte	C--T-	[17.*] Scene number [18.*] Scene control	657, 663, 669, 675, 681, 687, 693, 699 *705, 711, 717, 723, 729, 735, 741, 747.*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Stores or recalls a scee.* *The 6 lowest bits in the code byte represent the scene number, while the highest bit is the operation code (store or recall).*  				
Input *xx* – Lock command	Input xx = digital input Type = scene Lock function = enabled	1 bit	C-W--	[1.003] enable	653, 659, 665, 671, 677, 683, 689, 695 *701, 707, 713, 719, 725, 731, 737, 743.*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).				

8.4.6 Independent or single - Digital input - Shift register

8.4.6.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Contact type	Input xx = digital input Type = shift register	**NO (normally open)** NC (normally closed)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the rest state.*	
Lock function	Input xx = digital input Type = shift register	**disabled** enabled
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Enables or disables the lock function.*	
Shift register maximum value	Input xx = digital input Type = shift register	0…255 (**0**)

Parameter name	*Conditions*	*Settings*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Maximum value for the counter.*	
Shift register minimum value	Input xx = digital input Type = shift register	0…255 (**0**)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Minimum value for the counter.*	
Shift register step value	Input xx = digital input Type = shift register	-128…127 (**0**)
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Increment/decrement step of the register.*	
Shift register direction	Input xx = digital input Type = shift register	**From lowest to highest value** From highest value to lowest value
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows you to choose the increment/decrement direction of the register.*	
Shift register reset enable	Input xx = digital input Type = shift register Shift register reset enable = enabled	**Disabled** enabled
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Enable or disable register reset.* *If enabled, a long contact (press) of the input resets the counter.*	
Long press time	Input xx = digital input Type = shift register	hh:mm:ss.fff (**00:00:00.800**) [range 00:00:00.215 04:39:37.215]
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Minimum pressure maintenance time to distinguish between short and long pressure.*	
Lock function – Invert lock device signal	Input xx = digital input Type = shift register Lock function = enabled	**not inverted** / inverted
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows a command's "lock active" code to be interpreted as "lock inactive" and vice versa.*	
Lock function – Lock after bus recovery	Input xx = digital input Type = shift register Lock function = enabled	**no** / yes
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Shift register	Input xx = digital input Type = shift register	1 byte	CR-T-	[5.010] counter pulses (0.255)	658, 664, 670, 676, 682, 688, 694, 700 *706, 712, 718, 724, 730, 736, 742, 748.*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).				

8.4.7 Coupled: switching

8.4.7.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Contact type	Input channel x = coupled Type = switching	**NO (normally open)** NC (normally closed)
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the rest state.*	
Lock function	Input channel x = coupled Type = switching	**disable** enable
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Enables or disables the lock function.*	
1 and 2 use	Input channel x = coupled Type = switching	**Input odd increases, even decreases** Input odd decreases, even increases
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows the functionality of the odd and even inputs to be reversed.*	
Send cyclically	Input channel x = coupled Type = switching	none **off / value 1** on / value 2 both off and on / both values
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines which values, if any, must be transmitted periodically if they are in the activated state.*	
Cyclic sending interval	Input channel x = coupled Type = switching Send cyclically ≠ none	hh:mm:ss (**00:02:00**) [range 00:00:10 … 04:39:37]
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Interval between periodic retransmissions.*	
Lock function – Invert lock device signal	Input channel x = coupled Type = switching Lock function = enabled	**Not inverted**/inverted
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows a "lock active" command code to be interpreted as "lock deactivated" and vice versa.*	
Lock function – Lock after bus recovery	Input channel x = coupled Type = switching Lock function = enabled	**no** / yes
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	
Lock function – Behaviour at locking	Input channel x = coupled Type = switching Lock function = enabled	**none** off on toggle
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines the operation to be performed when the lock is activated. The choice is between the operations associated with the two possible events of closure (or short contact, depending on the configuration) or opening (or prolonged contact).*	

Parameter name	*Conditions*	*Settings*
Lock function – Behaviour at unlocking	Input channel x = coupled Type = switching Lock function = enabled	**None** off on as previous
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines the operation to be performed upon release. The choice is between the operations associated with the two possible events of closure (or short contact, depending on the configuration) or opening (or prolonged contact).*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input channel *x* – Switching command	Input channel x = coupled Type = switching	1-bit	C-WTU	[1.001] switch	658, 670, 682, 694, *706, 718, 730, 742.*
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).				
Input channel *x* – Lock command	Input channel x = coupled Type = switching Lock function = enabled	1 bit	C-W-U	[1.003] enable	653, 665, 677, 689, *701, 713, 725, 737.*
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).				

8.4.8 Coupled: dimming

8.4.8.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Contact type	Input channel x = coupled Type = dimming	**NO (normally open)** NC (normally closed)
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the idle state.*	
Lock function	Input channel x = coupled Type = dimming	**disabled** enabled
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Enables or disables the lock function.*	
Send cyclically	Input channel x = coupled Type = dimming	None **off / value 1** on / value 2 both off and on / both values
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines which values, if any, should be transmitted periodically if they are in the activated state.*	
Cyclic sending interval	Input channel x = coupled Type = dimming Send cyclically ≠ none	hh:mm:ss (**00:02:00**) [range 00:00:10 … 04:39:37]
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Period for cyclical transmission of the value to the bus.*	

Parameter name	*Conditions*	*Settings*
Long press time	Input channel x = coupled Type = dimming	hh:mm:ss.fff **(00:00:08.000)** [range 00:00:00.215 04:39:37.215]
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Minimum pressure maintenance time to distinguish between short and long pressure.*	
Usage 1 and 2	Input channel x = coupled Type = dimming	**Input odd increases, even decreases** Input odd decreases, even increases
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows the functionality of the odd input and even input to be reversed.*	
Lock function – Invert lock device signal	Input channel x = coupled Type = dimming Lock function = enabled	**not inverted** / inverted
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows a command's "lock active" code to be interpreted as "lock inactive" and vice versa.*	
Lock function – Lock after bus recovery	Input channel x = coupled Type = dimming Lock function = enabled	**no** / yes
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	
Lock function – Behaviour at locking	Input channel x = coupled Type = dimming Lock function = enabled	**none** off on toggle
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines the operation to be performed when the lock is activated. The choice is between the operations associated with the two possible events of closure (or short contact, depending on the configuration) or opening (or prolonged contact).*	
Lock function – Behaviour at unlocking	Input channel x = coupled Type = dimming Lock function = enabled	**none** off on as previous
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines the operation to be performed upon release. The choice is between the operations associated with the two possible events of closure (or short contact, depending on the configuration) or opening (or prolonged contact).*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input channel x - Switch command	Input channel x = coupled Type = dimming	1 bit	C-WTU	[1.001] switch	654, 666, 678, 690, *702, 714, 726, 738*.
	See notes for independent input. *x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).*				
Input channel *x* – Dimming up / down / stop command	Input channel x = coupled Type = dimming	4 bits	C--T-	[3.*] 3-bit control	655, 667, 679, 691, *703, 715, 727, 739.*
	See notes for independent input. *x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).*				

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input channel *x* – Lock command	Input channel x = coupled Type = dimming Lock function = enabled	1 bit	C-W-U	[1.003] enable	653, 665, 677, 689, *701, 713, 725, 737.*
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).				

8.4.9 Coupled: shutter or venetian blinds

8.4.9.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Contact type	Input channel x = coupled Type = shutter or venetian blinds	**NO (normally open)** NC (normally closed)
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows you to choose whether the input contacts are separated (NO) or joined (NC) in the rest state.*	
Lock function	Input channel x = coupled Type = shutter or venetian blinds	**disable** enable
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Enables or disables the lock function.*	
Long press time	Input channel x = coupled Type = shutter or venetian blinds	hh:mm:ss.fff **(00:00:08.000)** [range 00:00:00.215 04:39:37.215]
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Minimum pressure maintenance time to distinguish between short and long pressure.*	
1 and 2 use	Input channel x = coupled Type = shutter or venetian blinds	**A up, B down** A up, B down
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows you to reverse the functionality of the odd and even inputs.*	
Venetian blind mode	Input channel x = coupled Type = shutter or venetian blinds	**Disabled** Enabled
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *If Venetian blind mode is enabled, the device sends "full movement" commands when pressed and held, and "step" commands when pressed briefly;* *if disabled, it still sends "full movement" commands when pressed and held, but "stop" commands when pressed briefly.*	
Lock function – Invert lock device signal	Input channel x = coupled Type = shutter or venetian blinds Lock function = enabled	**not inverted** / inverted
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Allows a "lock active" command code to be interpreted as "lock deactivated" and vice versa.*	
Lock function – Lock after bus recovery	Input channel x = coupled Type = shutter or venetian blinds Lock function = enabled	**no** / yes
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	

Parameter name	*Conditions*	*Settings*
Lock function – Behaviour at locking	Input channel x = coupled Type = shutter or venetian blinds Lock function = enabled	**none** up down
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines the operation to be performed when locking occurs. The choice is between the operations associated with the two possible events of closing (or short contact, depending on the configuration) or opening (or prolonged contact).*	
Lock function – Behaviour at unlocking	Input channel x = coupled Type = shutter or venetian blinds Lock function = enabled	**none** up down
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I). *Defines the operation to be performed when unlocked. The choice is between the operations associated with the two possible events of closing (or short contact, depending on the configuration) or opening (or prolonged contact).*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input channel *x* – Dedicated stop command	Input channel x = coupled Type = shutter or Venetian blinds Venetian blind mode = disabled	1 bit	C—T-	[1.017] trigger	657, 669, 681, 693, *705, 717, 729, 741.*
	See notes for independent input. *x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).*				
Input channel *x* – Stop-step up/down command	Input channel x = coupled Type = shutter or Venetian blinds Venetian blind mode = enabled	1 bit	C--T-	[1.007] step	657, 669, 681, 693, *705, 717, 729, 741.*
	See notes for independent input. *x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).*				
Input channel *x* – Move up/down command	Input channel x = coupled Type = shutter or Venetian blinds	1 bit	C—T-	[1.008] up/down	656, 668, 680, 692, *704, 716, 728, 740.*
	See notes for independent input. *x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).*				
Input channel *x* – Lock command	Input channel x = coupled Type = shutter or Venetian blinds Lock function = enabled	1 bit	C-W-U	[1.003] enable	653, 665, 677, 689, *701, 713, 725, 737.*
	x = A, B, C D (E, F, G, H only for EK-FI1-TP-I).				

8.4.10 Lock function

When the lock function is enabled, a behaviour can be defined for each input to be performed when a lock or unlock command is received.

Details are provided in the following paragraphs; a summary of the various options is provided in Table 2.

Function type	*Behaviour on lock*	*Behaviour When unlocked*
Switching	**none** off on toggle	**none** off on as previous
Dimming	**none** off on toggle	**none** off on as previous
shutters or blinds	**none** up down	**none** up down
scene	**none** send first scene send second scene	
Shift register	NA	

Table 2 - Block function behaviours

8.4.10.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Lock function	-	enabled/**disabled**
	Enables or disables the ability to lock an input via remote command (bus telegram).	
Lock function – Invert lock device signal	Input *xx* = enabled Type = any Lock function = enabled	**not inverted** / inverted
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). *Allows a command's "lock active" code to be interpreted as "lock deactivated" and vice versa.*	
Lock function – Lock after bus recovery	Input *xx* = enabled Type = any Lock function = enabled	**no** / yes
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I). • *If Lock after bus recovery = yes, after a bus off/on the input is always locked, regardless of how it was before the bus off;* • *if Lock after bus recovery = no, then if it was locked before the bus off, it remains locked after the bus on.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input *xx* – Lock command	Input *xx* = enabled Lock function = enabled	1 bit	C-W--	[1.003] enable	653, 659, 665, 671, 677, 683, 689, 695 *701, 707, 713, 719, 725, 731, 737, 743*
	(xx = 1, …, 8 for EK-FG1-TP-I, or xx = 1, …, 16 for EK-FI1-TP-I).				

8.4.11 Analogue input - NTC

Setting an input channel as independent or single, and a single input as NTC, allows you to activate the ***Temperature Sensor x*** tab for that specific channel.

Note: for the EK-FGI-TP-I device, this setting is only possible for channels E, F, G, H (inputs 9 to 16).

The ***Temperature Sensor x*** tab contains the following parameters:

- Filter type on internal data processing
- Temperature offser
- Minimum change of value to send [K]
- Cyclic sending interval
- Threshold 1
- Threshold 2
- Thermostat functionality

8.4.11.1 Parameters and communication objects

Parameter name	***Conditions***	***Settings***
Temperature sensor x – Filter type	Temperature sensor x = enabled	low **medium** high
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I). *Low = average value every 4 measurements* *Medium = average value every 16 measurements* *High = average value every 64 measurements*	
Temperature sensor x – Temperature offset	Temperature sensor x = enabled	**0°C** [range -5°C … +5°C]
	Sets an offset to be applied to the measured temperature value. *(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).*	
Temperature sensor x – Minimum change of value to send [K]	Temperature sensor x = enabled	**5** [range 0 … 5]
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I). *Sets the minimum temperature change for the value to be sent to the bus.* *If the parameter is set to 0, no value is sent when the change occurs.*	
Temperature sensor x – Cyclic sending interval	Temperature sensor x = enabled	**no sending** [other values in the range 30 s … 120 min]
	Period for cyclical transmission of the value to the bus. *(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).*	
Temperature sensor x – Threshold 1	Temperature sensor x = enabled	**not active** below above
	Allows you to set a threshold to compare the measured value. *(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).*	
Temperature sensor x – Value [°C]	Temperature sensor x = enabled, Threshold 1 = below or above	**45** [range 0 … 50]
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	

Parameter name	*Conditions*	*Settings*
Temperature sensor x – Threshold value update from bus	Temperature sensor x = enabled, Threshold 1 = below or above	**no** / yes
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Threshold lock enable	Temperature sensor x = enabled, Threshold 1 = below or above	**no** / yes
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Behaviour at lock	Temperature sensor x = enabled, Threshold 1 = below or above Behaviour at lock = yes	**none** / off / on
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Behaviour at bus recovery	Temperature sensor x = enabled, Threshold 1 = below or above Behaviour at lock = yes	**previous state** / lock / unlock
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Hysteresis	Temperature sensor x = enabled, Threshold 1 = below or above	**0.4 K** [other values between 0.2 K and 3 K]
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I). *Sets a hysteresis value for comparing the temperature value and the threshold.*	
Temperature sensor x – Cyclic sending interval	Temperature sensor x = enabled, Threshold 1 = below or above	**no sending** [other values in the range 30 s … 120 min]
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Threshold 2	Temperature sensor x = enabled	**not active** below above
	Allows you to set a threshold to compare the measured value. *(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).*	
Temperature sensor x – Value [°C]	Temperature sensor x = enabled, Threshold 2 = below or above	**45** [range 0 … 50]
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Threshold value update from bus	Temperature sensor x = enabled, Threshold 2 = below or above	**no** / yes
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Threshold lock enable	Temperature sensor x = enabled, Threshold 2 = below or above	**no** / yes
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Behaviour at lock	Temperature sensor x = enabled, Threshold 2 = below or above Behaviour at lock = yes	**none** / off / on
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Behaviour at bus recovery	Temperature sensor x = enabled, Threshold 2 = below or above Behaviour at lock = yes	**previous state** / lock / unlock
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Hysteresis	Temperature sensor x = enabled, Threshold 2 = below or above	**0.4 K** [other values between 0.2 K and 3 K]
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I). *Sets a hysteresis value for comparing the temperature value and the threshold.*	

Parameter name	*Conditions*	*Settings*
Temperature sensor x – Cyclic sending interval	Temperature sensor x = enabled, Threshold 2 = below or above	**no sending** [other values in the range 30 s … 120 min]
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).	
Temperature sensor x – Thermostat function	Temperature sensor x = enabled	**Disabled** Enabled
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I). Enables or disables thermostat functions on the selected input.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input x – Temperature value	Temperature sensor x = enabled	2 bytes	CR-T-	[9.001] temperature (°C)	877, 884, 891, 898, 905, 912, 919, 926
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).				
Temperature threshold 1 - switch	Temperature sensor x = enabled, Threshold 1 = below or above	1 bit	CR-T-	[1.001] switch	878, 885, 892, 899, 906, 913, 920, 927
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).				
Temperature threshold 1 - Lock	Temperature sensor x = enabled, Threshold 1 = below or above Activate threshold block = yes	1 bit	C-W--	[1.001] switch	880, 887, 894, 901, 908, 915, 922, 929
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).				
Temperature threshold 1 – Value (from bus)	Temperature sensor x = enabled, Threshold 1 = below or above Update threshold value from bus = yes	2 bytes	C-W--	[9.001] temperatures (°C)	882, 889, 896, 903, 910, 917, 924, 931
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).				
Temperature threshold 2 - switch	Temperature sensor x = enabled, Threshold 2 = below or above	1 bit	CR-T-	[1.001] switch	879, 886, 893, 900, 907, 914, 921, 928
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).				
Temperature threshold 2 – Lock	Temperature sensor x = enabled, Threshold 2 = below or above Activate threshold block = yes	1 bit	C-W--	[1.001] switch	881, 888, 895, 902, 909, 916, 923, 930
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).				
Temperature threshold 2 – Value (from bus)	Temperature sensor x = enabled, Threshold 2 = below or above Update threshold value from bus = yes	2 bytes	C-W--	[9.001] temperatures (°C)	883, 890, 897, 904, 911, 918, 925, 932
	(x = 1, …, 8 for EK-FG1-TP-I, or x = 9, …, 16 for EK-FI1-TP-I).				

Acquisition filter

The acquisition filter calculates an average between a series of acquired values of the measured quantity before sending it to the bus. The parameter can take the following values:

- low (average value calculated every 4 measurements);
- average (average value calculated every 16 measurements);
- high (average value calculated every 64 measurements).

Measured temperature correction (offset)

The temperature value is sampled (indicatively) every 10 seconds; during configuration with ETS, it is possible to correct the measured temperature value within the offset range - 5 °C ... + 5 °C (0.1 K step).

8.5 Thermostat function

When an input is configured to read an NTC temperature sensor, it is also possible to enable or disable the thermostat function.

The ***Thermostat*** tab allows you to configure the input as a room temperature controller and also allows you to filter the main sensor (NTC probe) reading with the reading of an additional sensor acquired from the bus, using a weighted average.

The tab is active if the thermostat function has been enabled in the *Temperature Sensor x* tab.

The following secondary tabs are included:

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

8.5.1 External sensors (from bus)

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

The device's internal control system cyclically monitors the update status of the external sensor values (from the bus) when the set timeout value expires. If no updated value has been received, the control function is suspended and the actuator valves are closed. An alarm is sent to the bus via communication objects *987, 1062, 1137, 1212, 1287, 1362, 1437, 1512 - Room temperature control alarm* (refer to the *Settings* tab).

The *External Sensors (from bus)* tab is always active and contains the following parameters:

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

8.5.1.1 Parameters and communication objects

Parameter name	Conditions	Settings
Room temperature		**Disabled**/Enabled
	Enables a temperature bus sensor. The measured value can be used to calculate a weighted average value in combination with the temperature sensor integrated in the device.	
Cyclical reading each	Room temperature = enabled	**No reading** [other values in the range 30 s … 120 min]
	If the parameter is set to "no reading", the corresponding communication object must be updated by the remote device sending the data. With other values, the data is updated with a read request from the room thermostat.	
Relative humidity		**disabled** / enabled
Humidity CO dimension	Relative humidity = enabled	1 byte (DPT 5.001) **2 bytes (DPT 9.007)**
Cyclical reading each	Relative humidity = enabled	**No reading** [other values in the range 30 s … 120 min]
Antistratification temperature		**disabled** / enabled
	Enables a temperature bus sensor to perform the anti-stratification function.	
Cyclical reading each	Anti-stratification temperature = enabled	**No reading** [other values in the range 30 s … 120 min]
Outdoor temperature		**disabled** / enabled
	Enables an external temperature bus sensor	
Cyclical reading each	External temperature = enabled	**No reading** [other values in the range 30 s … 120 min]
Coil (exchange battery) temperature		**disabled** / enabled
	Enables a bus sensor for measuring the temperature of the heat transfer fluid at the heat exchange coil. The value acquisition allows the fan hot-start function to be performed.	
Cyclical reading each	Coil temperature = enabled	**No reading** [other values in the range 30 s ... 120 min]
Floor surface temperature		**disabled** / enabled
	Enables a bus sensor for measuring the surface temperature of a heated floor. The value acquisition allows the surface temperature limitation function to be implemented.	
Cyclical reading each	Floor surface temperature = enabled	**No reading** [other values in the range 30 s … 120 min]
Flow temperature		**disabled** / enabled
	Enables a bus sensor for measuring the flow temperature of the heat transfer fluid. The value acquisition allows the dew point temperature to be calculated and the active anti-condensation protection function to be implemented in surface cooling systems (floor or ceiling).	

Parameter name	Conditions	Settings
Cyclical reading each	Flow temperature = enabled	**No reading** [other values in the range 30 s ... 120 min]
Analog sensors timeout		**00:05:10** hh:mm:ss [range 00:00:00 … 04:39:37]
	The field has the format hh:mm:ss (hours : minutes : seconds): the default value 00:05:00 therefore corresponds to a timeout of 5 minutes. The value 00:00:00 means that the analogue sensor timeout is deactivated.	
Anticondensation probe		**disabled** / enabled
	Enables a bus sensor for detecting condensation.	
Signal	Anti-condensation probe = enabled	**Not inverted** / inverted
Cyclical reading each	Anti-condensation probe = enabled	**No reading** [other values in the range 30 s … 120 min]
Window contact 1		**disabled** / enabled
	Enables a bus sensor to detect whether a window or door is open or closed.	
Signal	Window contact 1 = enabled	**Not inverted** / inverted
Cyclical reading each	Window contact 1 = enabled	**No reading** [other values in the range 30 s … 120 min]
Window contact 2		**disabled** / enabled
	Enables a bus sensor for detecting the open/closed status of a window or door.	
Signal	Window contact 2 = enabled	**Not inverted** / inverted
Cyclical reading each	Window contact 2 = enabled	**No reading** [other values in the range 30 s … 120 min]
Presence sensor 1		**disabled** / enabled
	Enables a bus sensor to detect movement/presence of people within the environment.	
Signal	Presence sensor 1 = enabled	**not inverted** / inverted
Cyclical reading each	Presence sensor 1 = enabled	**No reading** [other values in the range 30 s … 120 min]
Presence sensor 2		**disabled** / enabled
	Enables a bus sensor to detect movement/presence of people within the environment.	
Signal	Presence sensor 2 = enabled	**not inverted** / inverted

Parameter name	Conditions	Settings
Cyclical reading each	Presence sensor 2 = enabled	**No reading** [other values in the range 30 s … 120 min]
Card holder contact		**disabled** / enabled
	Enables a bus sensor to detect the presence/absence of people in the room.	
Signal	Card holder pocket contact = enabled	**not inverted** / inverted
Cyclical reading each	Card pocket contact = enabled	**no reading** [other values in the range 30 s … 120 min]
Digital sensor timeout		**00:05:10** hh:mm:ss [range 00:00:00 … 04:39:37]
	The field has the format hh:mm:ss (hours : minutes : seconds): the default value 00:05:00 therefore corresponds to a timeout of 5 minutes. The value 00:00:00 means that the digital sensor timeout is deactivated.	

About sensor timeout

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

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Room temperature (from bus)	Room temperature sensor = enabled	2 bytes	C-WTU	[9.001] temperatures (°C)	933, 1008, 1083, 1158, 1233, 1308, 1383, 1458.
Thermostat - Relative humidity (2 bytes, from bus)	Relative humidity sensor = enabled, Communication object size humidity = 2 bytes	2 bytes	C-WTU	[9.007] humidity (%)	934, 1009, 1084, 1159, 1234, 1309, 1384, 1459.
Thermostat - Relative humidity (1 byte, from bus)	Relative humidity sensor = enabled, Communication object size humidity = 1 byte	1 byte	C-WTU	[5.001] percentage (0..100%)	935, 1010, 1085, 1160, 1235, 1310, 1385, 1460.
Thermostat - Antistratification temperature (from bus)	enabled	2 bytes	C-WTU	[9.001] temperatures (°C)	936, 1011, 1086, 1161, 1236, 1311, 1386, 1461.
Thermostat - Outdoor temperature (from bus)	enabled	2 bytes	C-WTU	[9.001] temperature °C	937, 1012, 1087, 1162, 1237, 1312, 1387, 1462.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat – Coil temperature (from bus)	enabled	2 bytes	C-WTU	[9.001] temperatures (°C)	938, 1013, 1088, 1163, 1238, 1313, 1388, 1463.
Thermostat - Floor temperature (from bus)	enabled	2 bytes	C-WTU	[9.001] temperatures (°C)	939, 1014, 1089, 1164, 1239, 1314, 1389, 1464.
Thermostat - Flow temperature (from bus)	enabled	2 bytes	C-WTU	[9.001] temperatures (°C)	940, 1015, 1090, 1165, 1240, 1315, 1390, 1465.
Thermostat - Anticondensation (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	946, 1021, 1096, 1171, 1246, 1321, 1396, 1471.
Thermostat - Window contact sensor 1 (from bus)	enabled	1 Bit	C-WTU	[1.019] window/door	941, 1016, 1091, 1166, 1241, 1316, 1391, 1466.
Thermostat - Window contact sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.019] window/door	942, 1017, 1092, 1167, 1242, 1317, 1392, 1467.
Thermostat - Presence sensor 1 (from bus)	enabled	1 bit	C-WTU	[1.018] occupancy	943, 1018, 1093, 1168, 1243, 1318, 1393, 1468.
Thermostat - Presence sensor 2 (from bus)	enabled	1 Bit	C-WTU	[1.018] occupancy	944, 1019, 1094, 1169, 1244, 1319, 1394, 1469.
Thermostat - Card holder contact (from bus)	enabled	1 Bit	C-WTU	[1.001] switch	945, 1020, 1095, 1170, 1245, 1320, 1395, 1470.

8.5.2 Weighted temperature value

The *Weighted temperature value* tab can only be used if the acquisition of the ambient temperature is enabled both from the main sensor (NTC probe connected to the input) and from the additional sensor. It contains the following parameters:

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

8.5.2.1 Parameters and communication objects

Parameter name	Conditions	Settings
Relative weight		100% main sensor 90% / 10% 80% / 20% 70% / 30% 60% / 40% **50% / 50%** 40% / 60% 30% / 70% 20% / 80% 10% / 90% 100% additional sensor
Minimum variation for sending value [K]	Relative weight ≠ 100% main sensor and Relative weight ≠ 100% sensor from bus	**0.5** [other values in the range 0 … 5 K]
	If the parameter is set to 0, no value is sent upon change.	
Cyclical transmission interval	Relative weight ≠ 100% main sensor and Relative weight ≠ 100% sensor from bus	**no sending** [other values in the range 30 s … 120 min]

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Weighted temperature	Room temperature (from bus) = enabled	2 bytes	CR-T--	[9.001] temperatures °C	947, 1022, 1097, 1172, 1247, 1322, 1397, 1472.

Note on weighted temperature

The thermostat allows the ambient temperature to be acquired in two ways:

1. from the temperature NTC probe connected to an input;
2. via bus from another KNX device, for example from an ekinex push button [External sensors (from bus) - Room temperature = enabled];

To optimise or correct the room temperature setting in special cases (in large rooms, in the presence of significant temperature distribution asymmetry, when the thermostat is installed in an unsuitable position, etc.), the device can therefore use a weighted average of two temperature values. The weights are assigned using the *Relative weight* parameter, which assigns a proportion to the two values.

Note: the value for the "Weighted temperature" communication object is set to 7F FF if the actual value cannot be read from the bus.

8.5.3 Temperature control

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

- Settings
- Heating
- Cooling
- Ventilation

The *Cooling* sub-tab only appears if the Thermostat function parameter in the Settings tab is set to *"heating and cooling"* or *"cooling"*.

The *Ventilation* sub-tab appears in the following cases:

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

8.5.4 Settings

The *Settings* tab contains the following parameters:

- Setpoint type
- Thermostat function
- Single or separate command communication object (2-pipe or 4-pipe systems)
- Heating – cooling switching type
- Heating/cooling cyclic sending interval
- Thermostat function after download
- Setpoint cyclic sending interval
- End of manual operation
- Disable temperature controller from bus
- Signal from bus
- Transmission delay after mode change
- Valve protection function
- Frequency
- Time interval

8.5.4.1 Parameters and communication objects

Parameter name	Conditions	Settings
Setpoint type	Thermostat function = enabled	**single** absolute relative
Thermostat function		**heating** cooling both heating and cooling
Command communication object	Thermostat function = both heating and cooling	**separate** / single

Parameter name	Conditions	Settings
Heating - cooling changeover	Setpoint type ≠ single Command communication object = separate Thermostat function = both heating and cooling	**from bus** / automatic
	If the setpoint type is single, or the command communication object is unique, the heating-cooling switching command can only be sent via bus.	
Heating-cooling cyclic sending interval	Thermostat function = both heating and cooling	**no sending** [other values in the range 30 s … 120 min]
Thermostat function after download	Thermostat function = both heating and cooling Heating – cooling changeover = from bus	**no change** Heating Cooling
Setpoint cyclic sending interval		**no sending** [other values in the range 30 s … 120 min]
	The setpoint value that can be sent cyclically is the actual value, depending on the operating mode set manually by the user or automatically by another KNX supervisor device with time programming capability. The actual setpoint value also takes into account the status of any window contacts and presence detection (provided that the corresponding functions have been enabled).	
End of manual operation		**At first telegram from bus** [other values in the range 30 min ... 48 h]
	Defines the mode for exiting manual/forced mode	
Disable temperature control from bus		**no** / yes
	Defines the mode for exiting temperature control via the bus	
Signal from bus	Disable temperature control from bus = yes	**Not inverted** inverted
Transmission delay after mode change		**00:00:04.00** [interval 00:00:00.00 … 04:39:37.21 hh:mm:ss.ff]
	Defines a delay in transmission on the bus after an HVAC mode change. A zero value (00:00:00) means that transmission is immediate.	
Valve protection function	Heating type ≠ electric and Thermostat function ≠ heating	**disabled** / enabled
	Enables the function that activates the valve control drives during periods of prolonged system inactivity.	
Frequency	Valve protection function = enabled	once a day, **once a week**, once a month
Time interval	Valve protection function = enabled	**10 s** [other values in the range 5 s … 20 min]

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Actual setpoint		2 Bytes	CR-T--	[9.001] temperatures (°C)	955, 1030, 1105, 1180, 1255, 1330, 1405, 1480.
Thermostat - Manual setpoint	Setpoint type = absolute or relative	2 bytes	C-W---	[9.001] temperatures (°C)	956, 1031, 1106, 1181, 1256, 1331, 1406, 1481.
Thermostat - Heating/cooling status out		1 bit	CR-T--	[1.100] heating/cooling	948, 1023, 1098, 1173, 1248, 1323, 1398, 1473.
	The communication object is updated on the bus at the switching event processed internally by the controller. The object is always exposed and contains information about the current operating mode of the internal temperature controller. **[1.100] DPT Heat/Cool 1 Bit** **0 = Cooling** **1 = Heating**				
Thermostat - Heating/cooling status in	Thermostat function = both heating and cooling; Heating/cooling changeover = from bus	1 Bit	C-W--	[1,100] heating/cooling	949, 1024, 1099, 1174, 1249, 1324, 1399, 1474.
	The communication object is received by the bus. At the switching event, the internal controllers of the primary and auxiliary stages (if enabled) switch the conduction mode.				
Thermostat - HVAC mode in	Setpoint type = absolute or relative	1 Byte	C-W--	[20.102] HVAC mode	950, 1025, 1100, 1175, 1250, 1325, 1400, 1475.
	The device receives the operating mode (HVAC mode) from a bus device with a supervisory function. The operating mode received via this communication object can be subsequently changed by the user (in this case, the room thermostat switches to manual control).				
Thermostat – HVAC forced mode in	Setpoint type = absolute or relative Thermostat function = both heating and cooling	1 byte	C-W--	[20.102] HVAC mode	951, 1026, 1101, 1176, 1251, 1326, 1401, 1476.
	The communication object allows the operating mode to be received in the same way as with the HVAC mode in communication object; the difference is that the operating mode received via this object (with the exception of the AUTO command) cannot be subsequently changed by the user. The user can only change the mode after the forced HVAC mode in has sent the AUTO command.				

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - HVAC Mode out	Setpoint type = absolute or relative	1 Byte	CR-T-	[20.102] HVAC mode	952, 1027, 1102, 1177, 1252, 1327, 1402, 1477.
Thermostat - HVAC manual mode	Setpoint type = absolute or relative	1 Byte	C-WTU	[20.102] HVAC mode	953, 1028, 1103, 1178, 1253, 1328, 1403, 1478.
Thermostat - Chrono active status	Setpoint type = absolute or relative	1 Bit	CRWTU	[1.011] state	954, 1029, 1104, 1179, 1254, 1329, 1404, 1479.
Thermostat - Room temperature control status		1 bit	CR-T-	[1.003] enable	970, 1045, 1120, 1195, 1270, 1345, 1420, 1495.
Thermostat - Alarm text		14 bytes	CR-T-	[16,000] Character string (ASCII)	987, 1062, 1137, 1212, 1287, 1362, 1437, 1512.
Thermostat - Manual setpoint active status	Setpoint type ≠ single	1 bit	CRWTU	[1.011] state	988, 1063, 1138, 1213, 1288, 1363, 1438, 1513.
Thermostat - Input setpoint	Setpoint type = single	2 bytes	CRWTU	[9.001] temperature (°C)	957, 1032, 1107, 1182, 1257, 1332, 1407, 1482.
	The communication object allows you to set and/or read the setpoint status (manual/forced).				
Thermostat - Room temperature control alarm		1 bit	CR-T--	[1.005] alarm	992, 1068, 1142, 1217, 1292, 1367, 1442, 1517.
Thermostat – disable room temperature controller (from bus)	Disable temperature controller from bus = yes	1 bit	C-W--	[1.001] switch	993, 1068, 1143, 1218, 1293, 1368, 1443, 1518.
Thermostat – Thermal generator lock alarm		1 bit	C-W--	[1.005] alarm	1004, 1079, 1154, 1229, 1304, 1379, 1454, 1529.
Thermostat - Building protection HVAC mode active		1 bit	CR-T--	[1.011] state	1005, 1080, 1155, 1230, 1305, 1380, 1455, 1530.

Note on heating and cooling system terminals

The thermostat application functions that can be configured with the ETS application are particularly suitable for command/control via KNX actuators (generic or dedicated) of the following system terminals:

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

8.5.4.2 Heating/cooling changeover

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

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

Automatic changeover, based on the room temperature (mode 1)

This mode is only suitable in applications for heating / cooling hydraulic systems with a 4-pipe configuration. Even in this case, the information can be sent on the bus with the output communication object [DPT 1,100 heat / cool]; the changeover is performed automatically by the device, basing on the values of current temperature and setpoint. The automatic switch over is achieved by introducing a dead band, as shown in Figure 39**.**

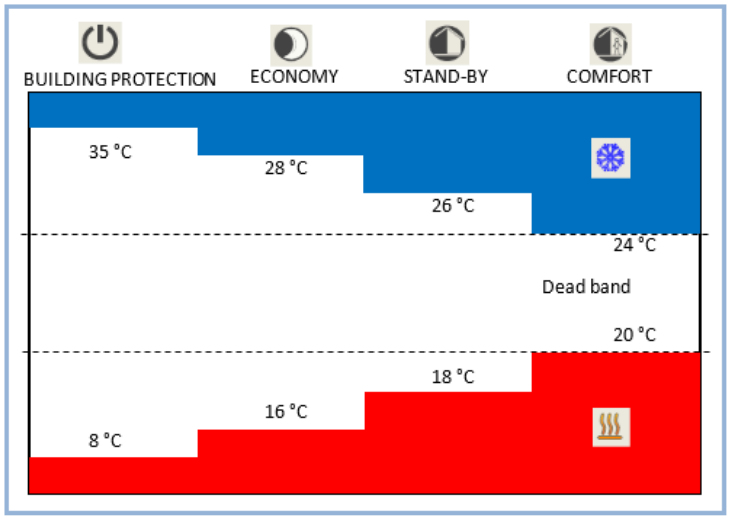


Figure 39 - Dead zone and example of consistently distributed setpoint values

As long as the actual (measured) temperature is below the setpoint value for heating, the operating mode remains heating; similarly, if the actual (measured) value is above the setpoint value for cooling, the operating mode is cooling. If the actual (measured) temperature value is within the dead zone, the operating mode remains the same as before; the switching point for the heating/cooling operating mode must be at the current setpoint of the set HVAC mode, and similarly, the changeover from cooling to heating must be at the set heating setpoint.

Switching from the KNX bus (mode 2)

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

Note: for convenience, the rest of this discussion will refer only to the group objects of input 1 for EK-FG1-TP-I (9 for EK-FGI1-TP-I). The same considerations apply to the other inputs not mentioned.

The communication objects shown in the block diagram allow you to monitor and change the current operating mode set on the temperature controller. Object 948 - *Heating/cooling status out* is always displayed, even when the thermostat function is only heating or only cooling. If the function is both heating and cooling, the cyclical sending of the object on the bus can be enabled; in all cases, information on the current control mode can be acquired by requesting a read operation on this communication object.

Object 949 - *Heating/cooling status in* is only displayed when the function is both heating and cooling and switching between modes is performed by the bus.

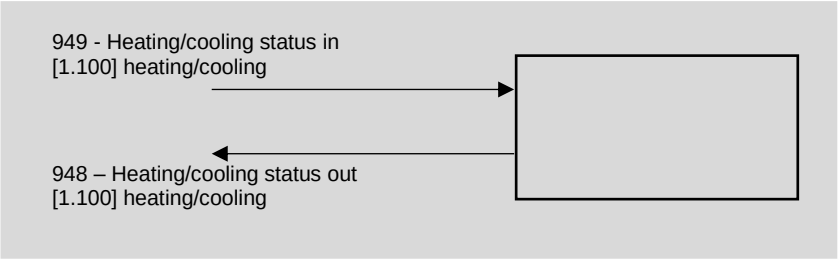


Figure 40 - Bus switching diagram

8.5.4.3 Valve protection function

This function is suitable for heating and cooling systems that use water as a heat transfer fluid and have motorised valves for shutting off a zone or a single room. Long periods of system inactivity can cause the valves to jam: to prevent this, the thermostat can periodically send a valve open/close command during periods of system inactivity. This option is available in the application programme via the "Valve protection function" parameter, which is further defined by the frequency and duration of valve operation.

Note: this function is not available if the heating type is electric and the thermostat function is set to heating only.

8.5.4.4 Remote setpoint modification

The communication objects (O.C. below) indicated in the block diagram of Figure 41 allow the setpoint to be monitored and modified manually.

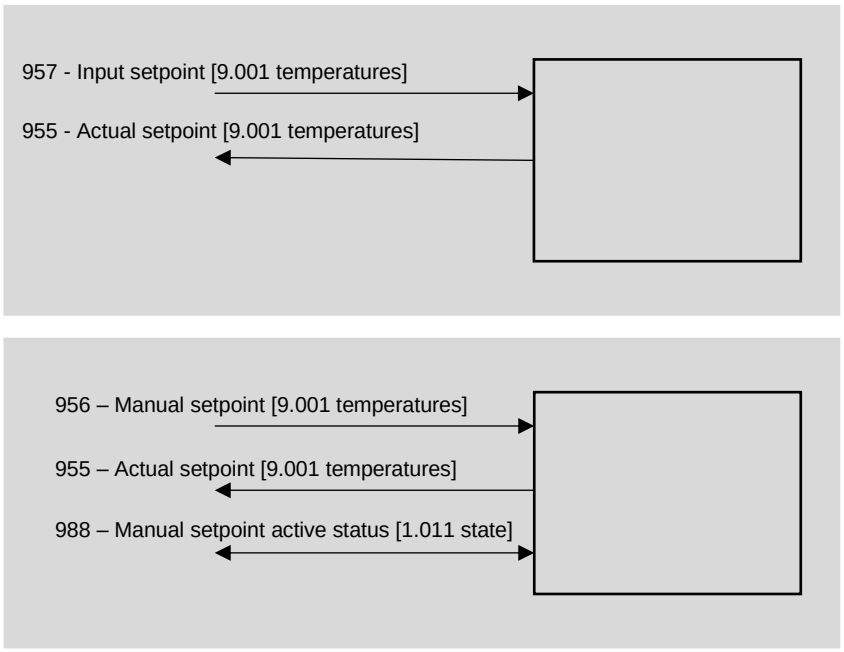


Figure 41 - Manual setpoint modification

The objects refer to manual setpoint modification: alternatively, the supervisor can act directly on the operating mode setpoints (O.C. with indices from 959 to 969). The value of O.C. *955-Current setpoint* represents the current operating setpoint on which the control algorithms operate. O.C. *988-Manual setpoint active* status indicates whether forced mode is enabled. The supervisor can force the current setpoint at any time by writing a new value directly to O.C. *956-Manual setpoint.* O.C. *988-Manual setpoint active status* can also be used in writing to exit forced mode.

8.5.4.5 Remote modification of operating modes

The communication objects shown in the block diagram allow you to monitor changes in operating mode (comfort, standby, economy and building protection) made in manual/forced mode, or the operating mode imposed by the time schedule. The communication objects (O.C. below) also allow you to make the same changes remotely, for example via a system supervisor.

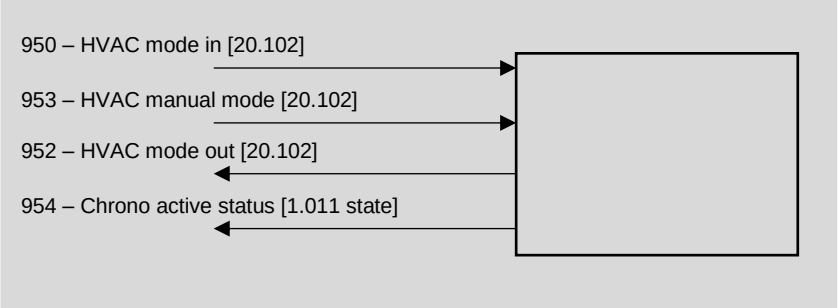


Figure 42 - Remote modification of operating modes

O.C. *950-HVAC mode in* is associated with the system time programme. O.C. *952-HVAC mode out* and *954-Crono active status* allow the remote supervisor to reconstruct the active mode on the room thermostat and understand whether the time programme is enabled or attenuation is managed manually. The supervisor can set a manual operating mode at any time via O.C. *953-Manual HVAC mode;* to activate the current time programme remotely, simply set O.C. 953 to 0 = Automatic.

8.5.5 Heating

The ***Heating*** tab allows you to set:

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

Activation conditions: Settings⇒ thermostat function = heating or heating and cooling.

This tab contains the following parameters:

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

8.5.5.1 Parameters and communication objects

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

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

Parameter name	Conditions	Settings
Comfort temperature setpoint [°C]	Setpoint type = absolute or relative	**21** [range 10 … 50]
Standby temperature setpoint [°C]	Setpoint type = absolute	**18** [range 10 … 50]
	For the device to function correctly, the standby temperature setpoint must be lower than the comfort temperature setpoint.	
Economy temperature setpoint [°C]	Setpoint type = absolute	**16** [range 10 … 50]
	For the device to function correctly, the economy temperature setpoint must be lower than the standby temperature setpoint.	
Standby temperature offset [0.1 K]	Setpoint type = relative	**-30** [range -50 … -10]
Economy temperature offset [0.1 K]	Setpoint type = relative	**-50** [range -80 … -10]
	For the device to function correctly, the economy temperature offset must be less than the standby temperature offset.	
Building protection temp. setpoint [°C]		**7** [range 2 … 10]
Heating type		**Radiators** Electric Fancoils Floor radiant panels Ceiling radiant panels
	Defines the terminal used for heat exchange in the environment. The choice determines the default parameters of the PWM control algorithm (proportional band and integral time) and the control options.	
Control type		**2-point hysteresis** PWM (pulse width modulation) continuous
Hysteresis	Control type = 2-point hysteresis, or continuous	**0.3 K** [other values in the range 0.2 K … 3 K]

Parameter name	Conditions	Settings
Hysteresis position	Control type = continuous	**below**/above
	The upper hysteresis is recommended for special applications that also require control of the mixing unit.	
Cyclic sending interval	Control type = 2-point hysteresis, continuous	**no sending** [other values in the range 30 s … 120 min]
PWM cycle time	Control type = PWM (pulse width modulation)	**15 min** [range 5 … 240 min]
Proportional band [0.1 K]	Control type = continuous or PWM	**50** [range 0 … 255]
	The value is represented in tenths of a degree Kelvin (K). *The field contains a preset value that depends on the selected heating type (the value can be changed):* *radiators: 50 (5 K)* *electric: 40 (4 K)* *fancoil: 40 (4 K)* *floor radiant panels: 50 (5 K)* *ceiling radiant panels: 50 (5 K)* *The value of the Proportional band parameter represents the maximum deviation between the desired temperature and the measured temperature that determines the maximum control output.*	
Integral time [min]	Control type = continuous or PWM	**240** [other values in the range 0 … 255 min]
	The field contains a preset value that depends on the selected heating type (the value can be changed): *radiators: 150 min* *electric: 100 min* *fancoil: 90 min* *floor radiant panels: 240 min* *ceiling radiant panels: 180 min*	
Min control value [%]	Control type = continuous or PWM	**15** [range 0 … 30]
Max control value [%]	Control type = continuous or PWM	**85** [range 70 … 100]

Parameter name	Conditions	Settings
Floor temperature limitation	Heating type = floor radiant panels External sensors from the bus⇒ Floor surface temperature probe = enabled	**disabled** / enabled
	This parameter enables the function that limits the surface temperature of a heated floor. For this function to work, the surface temperature of the floor must be measured by enabling the corresponding temperature sensor in the External sensors (from bus) tab. *Important. This function does not replace the over-temperature protection normally provided in hydronic floor heating systems, which is achieved by means of a special safety thermostat.*	
Temperature limit (upper) [°C]	Floor temperature limitation = enabled	**29** [range 20 … 40]
	According to standard EN 1264, a maximum permissible temperature is prescribed for the surface of radiant flooring: *T(sup) max ≤ 29°C for areas of normal occupancy;* *T(sup) max ≤ 35°C for peripheral areas of rooms.* *National regulations may also limit these temperatures to lower values. Peripheral areas are generally defined as strips located along the walls of the room facing the outside of the building, with a maximum width of 1 m.*	
Hysteresis [K]	Floor temperature limitation = enabled	**0.3 K** [other values in the range 0.2 K ... 3 K]
	The surface temperature must fall below the set threshold by an offset equal to the hysteresis value before the alarm status is cleared.	
Auxiliary heating		**disabled** / enabled
Communication object	Auxiliary heating = enabled	Unique / **separated**
Disabled from bus	Auxiliary heating = enabled	**No** / Yes
	Enables activation and deactivation of the function via a telegram from the supervisory device on the bus.	
Offset from setpoint	Auxiliary heating = enabled	**0.6 K** [other values in the range 0 … 3 K]
Hysteresis [K]	Auxiliary heating = enabled	**0.3 K** [other values in the range 0.2 K … 3 K]
Cyclic sending interval	Auxiliary heating = enabled	**No transmission** [other values in the range 30 s … 120 min]

Parameter name	Conditions	Settings
Ventilation for auxiliary heating	Heating type ≠ fancoils	**disabled** / enabled
	This option allows you to combine a high-inertia system such as underfloor heating (in the water-fed version) with a low-inertia system such as fancoils.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat – Input setpoint	Setpoint type = single	2 bytes	CRWTU	[9.001] temperature (°C)	957, 1032, 1107, 1182, 1257, 1332, 1407, 1482.
Thermostat - Comfort setpoint (heating)	Setpoint type = absolute or relative	2 bytes	CRWTU	[9.001] temperatures (°C)	957, 1032, 1107, 1182, 1257, 1332, 1407, 1482.
Thermostat - Standby setpoint (heating)	Setpoint type = absolute	2 bytes	CRWTU	[9.001] temperatures (°C)	960, 1035, 1110, 1185, 1260, 1335, 1410, 1485.
Thermostat - Standby offset (heating)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	960, 1035, 1110, 1185, 1260, 1335, 1410, 1485.
Thermostat - Economy setpoint (heating)	Setpoint type = absolute	2 bytes	CRWTU-	[9.001] temperature (°C)	964, 1039, 1114, 1189, 1264, 1339, 1414, 1489.
Thermostat - Economy offset (heating)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	964, 1039, 1114, 1189, 1264, 1339, 1414, 1489.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Building protection setpoint (heating)		2 bytes	CRWTU-	[9.001] temperatures (°C)	968, 1043, 1118, 1193, 1268, 1343, 1418, 1493.
Thermostat - Heating out command	Control type = 2-point hysteresis or PWM, Command communication object = separated	1 Bit	CR-T-	[1.001] switch	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.
Thermostat - Heating out command	Control type = continuous, Command communication object = separated	1 byte	CR-T-	[5.001] percentage (0..100%)	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.
Thermostat - Heating and cooling out command	Control type = 2-point hysteresis or PWM, Command communication object = unique	1 Bit	CR-T-	[1.001] switch	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.
Thermostat - Heating and cooling out command	Control type = continuous, Command communication object = unique	1 Byte	CR-T--	[5.001] percentage (0..100%)	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.
Thermostat - Auxiliary heating output command	Auxiliary heating = enabled, Command communication object = separated	1 Bit	CR-T--	[1.001] switch	977, 1052, 1127, 1202, 1277, 1352, 1427, 1502.
Thermostat - Auxiliary heating and cooling output command	Auxiliary heating = enabled (Auxiliary heating) Communication object = unique	1 Bit	CR-T--	[1.001] switch	977, 1052, 1127, 1202, 1277, 1352, 1427, 1502.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Auxiliary heating disable	Auxiliary heating = enabled, Disabled from bus = yes	1 Bit	C-W---	[1.003] enable	980, 1055, 1130, 1205, 1280, 1355, 1430, 1505.

8.5.6 Cooling

The ***Cooling*** tab contains the following parameters:

- Temperature setpoint [°C]
- Comfort temperature setpoint [°C]
- Standby temperature setpoint [°C]
- Economy temperature setpoint [°C]
- Standby temperature offset [0.1 K]
- Economy temperature offset [0.1 K]
- Building protection temperature setpoint [°C]
- Cooling type
- Control type
- Hysteresis [K]
- Hysteresis position
- Cyclic sending interval
- Proportional band [0.1 K]
- Integral time [min]
- PWM cycle time
- Min. change value for sending [%]
- Min control value [%]
- Max control value [%]
- Protection with probe (anti-condensation)
- Active anticondensation
- Flow temperature (project)
- Anti-condensation hysteresis range
- Alarm signal delay
- Auxiliary cooling
- Disabled by bus
- Deviation from setpoint
- Hysteresis [K]
- Cyclic sending interval
- Ventilation for auxiliary cooling

8.5.6.1 Parameters and communication objects

Conditions: *Temperature control⇒ Settings⇒* = cooling or both heating and cooling.

Parameter name	*Conditions*	*Settings*
Temperature setpoint [°C]	Setpoint type = single	**23** [range 10 … 50]
Temperature setpoint (comfort) [°C]	Setpoint type = absolute or relative	**23** [range 10 … 50]
Standby temperature setpoint [°C]	Setpoint type = absolute	**26** [range 10 … 50]
	For correct operation, the standby temperature setpoint must be greater than the comfort temperature setpoint.	
Economy temperature setpoint [°C]	Setpoint type = absolute	**28** [range 10 … 50]
	For correct operation, the economy temperature setpoint must be greater than the standby temperature setpoint.	
Standby temperature offset [0.1 K]	Setpoint type = relative	**30** [range 10 … 50]
Economy temperature offset [0.1 K]	Setpoint type = relative	**50** [range 10 … 80]
	For the device to function correctly, the economy temperature offset must be greater than the standby temperature offset.	
Building protection temperature (setpoint) [°C]		**36** [range 10 … 50]
Neutral zone [0.1 K]	⇒ settings Thermostat function = both heating and cooling ⇒ settings Communication command object = separate ⇒ settings Heating – cooling changeover = automatic Setpoint type = relative	**20** [range 10 … 80]
	Defines the width of the neutral zone, in case you want to switch automatically between heating and cooling starting from the Comfort heating setpoint, when this zone is exceeded.	
Cooling type		**fancoil**, floor radiant panels, ceiling radiant panels
	If in Settings the parameter Thermostat function = heating and cooling and Communication command object = single, the Cooling type is bound to the choice made in Heating.	
Control type		**2-point hysteresis**, PWM (pulse width modulation), continuous
Hysteresis [K]	Control type = 2-point hysteresis	**0.3 K** [other values in the range 0.2 K … 3 K]

Parameter name	*Conditions*	*Settings*
Hysteresis position	Cooling type = radiant floor panels, radiant ceiling panels, Control type = 2-point hysteresis	**below**/above
	The upper hysteresis is recommended for special applications that also require control of the mixing unit.	
Cyclic sending interval	Command communication object = separated Control type = 2-point hysteresis, continuous	**no sending** [other values in the range 30 s … 120 min]
Proportional band [0.1 K]	Control type = continuous or PWM	**50** [range 5 … 100]
	The value is represented in tenths of a degree Kelvin (K). **) The field contains a preset value that depends on the selected heating type (the value can be changed):* • *fancoil: 40 (4 K)* • *radiant floor: 50 (5 K)* • *radiant ceiling: 50 (5 K)* *The value of the Proportional Band parameter represents the maximum deviation between the desired temperature and the measured temperature that determines the maximum control output.*	
Integral time [min]	Control type = continuous or PWM	**100** [range 0 … 255 min]
	**) The range contains a preset value that depends on the selected heating type (the value can be changed):* ▪ *fancoil: 90 min* ▪ *underfloor heating: 240 min* ▪ *radiant ceiling: 180 min*	
PWM cycle time	Control type = PWM (pulse width modulation)	**15 min** [range 5 … 240 min]
Minimum change in value for sending [%]	Control type = continuous	**10** [range 0 … 100]
Min.control value [%]	Control type = continuous or PWM	**15** [range 0 … 30]
Max control value [%]	Control type = continuous or PWM	**85** [range 70 … 100]
Anti-condensation with probe	Cooling type = radiant floor panels, radiant ceiling panels, External sensors (from bus)⇒ Anti-condensation probe = enabled	**disabled** / enabled
Active anticondensation	Cooling type = floor radiant panels, ceiling radiant panels, External sensors (from bus)⇒ relative humidity = enabled	**disabled** enabled (design temperature)
	Cooling type = floor radiant panels, ceiling radiant panels, External sensors (from bus)⇒ Relative humidity = enabled External sensors (from bus)⇒ Flow temperature = enabled	**disabled** enabled (flow and dew point temperatures comparison)

Parameter name	*Conditions*	*Settings*
	If the flow temperature is lower than the calculated dew point temperature, the conduction mode is in cooling and the room thermostat is requesting flow, the thermostat closes the valve and signals the alarm condition via bus.	
Flow temperature (project)	Active anticondensation = enabled (project temperature) External sensors (from bus) ⇒ Flow temperature = disabled	**14 °C** [other values in the range 14 °C … 20°C]
	Only appears if the flow temperature from an external sensor (from the bus) is not activated.	
Anticondensation hysteresis range	Active anticondensation = enabled (project temperature) External sensors (from the bus)⇒ Flow temperature = disabled	0.2 K / 0.3 K / 0.4 K / **0.5 K** / 0.6 K 0.8 K / 1 K / 1.5 K / 2 K / 2.5 K / 3 K
	Before exiting the alarm condition, the calculated dew point temperature must fall below the flow temperature by an offset equal to the hysteresis value.	
Delay for alarm signal	Active anticondensation = enabled (flow and dew temperatures comparison), or Anticondensation with probe = enabled	**5 min** [other values in the range 30 s … 120 min]
Auxiliary cooling		**disabled** / enabled
Disabled from bus	Auxiliary cooling = enabled	**no** / yes
	This parameter enables the function to be activated and deactivated via a telegram from a bus device with a supervisory function.	
Offset from setpoint	Auxiliary cooling = enabled	0 K / 0.2 K / 0.4 K / **0.6 K** 0.8 K / 1 K / 1.5 K / 2 K / 2.5 K / 3 K
Hysteresis [K]	Auxiliary cooling = enabled	0.2 K / **0.3 K** / 0.4 K / 0.5 K / 0.6 K 0.8 K / 1 K / 1.5 K / 2 K / 2.5 K / 3 K
Cyclic sending interval	Auxiliary cooling = enabled	**no sending** [other values in the range 30 s … 120 min]
Ventilation for auxiliary cooling	Cooling type = floor radiant panels, ceiling radiant panels	**disabled** / enabled
	This option allows you to combine a high-inertia system such as underfloor heating (in the water-fed version) with a low-inertia system such as fancoils.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Comfort setpoint (cooling)	Setpoint type = absolute or relative	2 bytes	CRWTU	[9.001] temperature (°C)	959, 1034, 1109, 1184, 1259, 1334, 1409, 1484.
Thermostat - Standby setpoint (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.001] temperature (°C)	962, 1037, 1112, 1187, 1262, 1337, 1412, 1487.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Standby offset (cooling)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	962, 1037, 1112, 1187, 1262, 1337, 1412, 1487.
Thermostat - Economy setpoint (cooling)	Setpoint type = absolute	2 bytes	CRWTU	[9.001] temperature (°C)	966, 1041, 1116, 1191, 1266, 1341, 1416, 1491.
Thermostat - Economy offset (cooling)	Setpoint type = relative	2 bytes	CRWTU	[9.002] temperature difference (K)	966, 1041, 1116, 1191, 1266, 1341, 1416, 1491.
Thermostat - Building protection setpoint (cooling)		2 bytes	CRWTU	[9.001] temperatures (°C)	969, 1044, 1119, 1194, 1269, 1344, 1419, 1494.
Thermostat - Cooling out command	Communication object control = separate Control type = 2-point hysteresis or PWM	1 bit	CR-T-	[1.001] switch	976, 1051, 1126, 1201, 1276, 1351, 1426, 1501.
Thermostat - Cooling out command	Communication object control = separate Control type = continuous	1 byte	CR-T-	[5.001] percentage (0..100%)	976, 1051, 1126, 1201, 1276, 1351, 1426, 1501.
Thermostat - Auxiliary cooling output command	Auxiliary cooling = enabled	1 bit	CR-T-	[1.001] switch	979, 1054, 1129, 1204, 1279, 1354, 1429, 1504.
Thermostat - Auxiliary cooling disable	Auxiliary cooling = enabled, Disabled by bus = yes	1 bit	C-W--	[1.003] enable	981, 1056, 1131, 1206, 1281, 1356, 1431, 1506.
Thermostat - Anticondensation alarm	Active anticondensation or Protection with anticondensation probe = enabled	1 bit	CR-T-	[1.005] alarm	1003, 1078, 1153, 1228, 1303, 1378, 1453, 1528.

About anticondensation protection function

The purpose of this function is to prevent the condensation on the thermal exchange surfaces of the installation or building when cooling is working. This function is mainly used in systems with thermal exchange consisting in surface terminals such as for the floor and ceiling cooling radiant systems. In this case the hydraulic circuits

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

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

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

Nr.	Type	Denomination	Description
1a	Passive	Anticondensation protection by probe (via bus)	The thermostat receives the information about condensation via bus from a different KNX device through communication object 33: Anticondensation (from bus) [DPT 1.001 switch].
2a	Active	Anticondensation protection with comparison between flow temperature (constant projected value, set as parameter on ETS and dew-point temperature (calculated by the thermostat)	Software protection that intervenes by closing down the room cooling circuit when the flow temperature defined in the hydronic project (as set in the corresponding ETS parameter) is lower than dew-point temperature calculated by the room thermostat using temperature and relative humidity values. The communication object involved is 57: Cooling out command [DPT 1.001 switch].
2b	Active	Anticondensation protection with comparison between flow temperature (constant projected value, set as parameter on ETS and dew-point temperature (calculated by the thermostat)	Software protection that intervenes by closing down the room cooling circuit when the actual measured flow temperature and received via bus from a different KNX device is lower than dew-point temperature calculated by the room thermostat using temperature and relative humidity values. The communication objects involved are 27 at input: Flow temperature (from bus) [DPT 9.001 temperature °C] and 57: Cooling out command [DPT 1.001 switch].
3	Active	Anticondensation protection with dew-point temperature sending over the bus and adjustment of the flow temperature	Software protection that foresees the sending on the bus of the dew-point temperature calculated by the room thermostat using temperature and relative humidity values to a KNX device capable of controlling the mixing condition of the conveying fluid for the cooling circuit. The regulation is performed by the KNX device receiving the dew-point temperature sent by the thermostat. The communication object involved is 73: Dew-point temperature [DPT 9.001 temperature °C].

Table 3 - Anticondensation protection modes

If an anticondensation sensor is used, it is necessary use a device provided with a potential-free signalling contact. It is possible to connect the signalling contact to an input channel of another KNX device, e.g. a pushbutton interface or a binary input (External sensors (from bus) ⇒ Anticondensation sensor = enabled). In this case the signal of the sensor is transmitted to the room temperature controller through the status of a communication object (case 1a described in Table 3).

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

- if the flow temperature value is not available (case 2a of Table 3), you can insert the value used in the project (parameter Flow temperature (projected));

- if the flow temperature value is available (case 2b of Table 3), you enable the Anticondensation Active parameter for comparison;
- if an bus actuator capable of intervention on the conveying fluid's mixing is available, the thermostat sends on the bus the calculated value of the dew-point temperature; this parameter has to be enabled in the *Relative humidity control ⇒ Calculated psychrometric values* tab.

  The actuator compares this value with the flow temperature and, if necessary, modifies the mixing conditions in order to prevent the risk for condensation formation.

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

8.5.7 Main and auxiliary ventilation

The ***Ventilation*** tab contains the following parameters:

- Control type
- Threshold first speed [0.1 K]
- Threshold second speed [0.1 K]
- Threshold third speed [0.1 K]
- Speed control hysteresis [K]
- Speed proportional band [0.1 K]
- Minimum change of value to sending [%]
- Manual operation
- Hot start (use of temperature probe on heat exchanger for fan start-up)
- Antistratification function
- Disable ventilation from bus
- Fan start delay
- Fan stop delay

The conditions for the ventilation tab to appear are:

Heating⇒ Type of heating = fan-coil or

Cooling type = fancoil or a combination of the two conditions:

⇒ *heating* Heating type = floor radiant or ceiling radiant *and*⇒ *heating* Auxiliary heating fan = enabled

Cooling⇒ Cooling type = floor radiant or ceiling radiant *and* Cooling⇒ Auxiliary cooling fan = enabled

This allows you to control two types of systems: i) fancoil terminals or ii) radiant panel terminals as the main stage and fancoils as the secondary stage.

8.5.7.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Control type		**1 speed** 2 speeds 3 speeds Continuous regulation
Threshold first speed [0.1 K]	Control type ≥ 1 speed	**0** [range 0 … 255]
	The value is represented in tenths of °C. If the Thermostat function parameter = heating and cooling, the threshold value is valid for both operating modes.	
Threshold second speed [0.1 K]	Control type ≥ 2 speeds	**10** [range 0 … 255]
	The value is represented in tenths of °C. If the Thermostat function parameter = heating and cooling, the threshold value is valid for both operating modes. For the fancoil to operate correctly, the following constraint must be respected: Second speed threshold > First speed threshold.	
Threshold third speed [0.1 K]	Control type = 3 speeds	**20** [range 0 … 255]
	The value is represented in tenths of °C. If the Thermostat function parameter = heating and cooling, the threshold value is valid for both operating modes. For the fancoil to operate correctly, the following constraint must be respected: Third speed threshold > Second speed threshold.	
Speed control hysteresis [K]	Control type = 1, 2 or 3 speeds	**0.3 K** [other values in the range 0.2 K … 3 K]
Speed proportional band [0.1 K]	Control type = continuous regulation	**30** [range 5 … 100]
	The value is represented in tenths of °C. If the Thermostat function parameter = heating and cooling, the threshold value is valid for both operating modes.	
Min. change of value to send [%]	Control type = continuous regulation	**28** [range 2 … 40]
	See also the chapter Control algorithms for further information on the meaning of the parameter.	
Manual operation		**Not depending on the temperature** Depending on the temperature
	If the parameter Manual operation = independent of temperature, the fan remains at the speed set by the user even when the temperature setpoint is reached; if, on the other hand, Manual operation = dependent on temperature, the fan stops when the temperature setpoint is reached.	
Hot start	Thermostat function = heating or both heating and cooling, Heating type = fancoils External sensors from the bus⇒ heat exchanger temperature = enabled	**no** / yes
	For this function to work, at least one bus sensor must be enabled to measure the water temperature at the fancoil heat exchange coil.	
Min. temp. to start ventilation [°C]	Hot start = yes	**28** [range 28 … 40]
	If enabled, the function is only active during heating mode.	

Parameter name	*Conditions*	*Settings*
Anti-stratification function	External sensors from the bus⇒ Anti-stratification temperature = enabled	**disabled** / enabled
	The bus sensor must be enabled to measure the anti-stratification temperature in order for the function to work.	
Antistratification temp. differential	Anti-stratification function = enabled	**2 [K/m]** [other values in the range 0.25 … 4.00]
	DIN 1946 recommends not exceeding a value of 2 K/m for rooms of normal height (between 2.70 and 3 m).	
Hysteresis	Anti-stratification function = enabled	**0.5 K** [other values in the range 0.2 … 3 K]
Disable ventilation from bus		**no** / yes
Signal from bus	Disable ventilation from bus = yes	**Not inverted** inverted
Fan start delay		**0 s** [other values in the range 10 s … 12 min]
	Also appears if hot start mode is used by measuring the water temperature at the fancoil heat exchange coil. The function is active in both conduction modes.	
Fan stop delay		**0 s** [other values in the range 10 s … 12 min]
	This function allows the fan to continue operating, dissipating any residual heat or cold present in the heat exchange coil into the room. The function is active in both operating modes.	
Cyclic sending interval		**no sending** [other values in the range 30 s … 120 min]

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Fan continuous speed	Control type = continuous regulation	1 byte	CR-T-	[5.001] percentage (0..100%)	982, 1057, 1132, 1207, 1282, 1357 1432, 1507
Thermostat - Fan speed 1	Control type ≥ 1 speed	1 bit	CR-T-	[1.001] switch	983, 1058, 1133, 1208, 1283, 1358, 1433, 1508.
Thermostat - Fan speed 2	Control type ≥ 2 speeds	1 bit	CR-T-	[1.001] switch	984, 1059, 1134, 1209, 1284, 1359 1434, 1509.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Fan speed 3	Control type = 3 speeds	1 bit	CR-T--	[1.001] switch	985, 1060, 1135, 1210, 1285, 1360 1435, 1510.
Thermostat - Fan control disable	Disable ventilation from bus = yes	1 bit	C-W---	[1.002] boolean	986, 1061, 1136, 1211, 1286, 1361, 1436, 1511.
Thermostat – Fan manual speed		1 byte	CRWTU	[5,010] counter pulses (0…255)	989, 1064, 1139, 1214, 1289, 1364, 1439, 1514
Thermostat - Fan speed		1 byte	CR-T-	[5.010] counter pulses (0…255)	990, 1065, 1140, 1215, 1290, 1365, 1440, 1515.
Thermostat – Fan manual active status		1 bit	CRWTU	[1.011] state	991, 1066, 1141, 1216, 1291, 1366, 1441, 1516.
Thermostat - Fan manual speed percentage		1 byte	CR-T-	[5.001] percentage	1006, 1081, 1156, 1231, 1306, 1381, 1456, 1531.
Thermostat – Fan manual speed off status		1 bit	CR-T-	[1.011] state	1007, 1082, 1157, 1232, 1307, 1382, 1457, 1532.

8.5.7.2 Delayed fan start function (“hot-start”)

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

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

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

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

8.5.7.3 Antistratification function

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

The antistratification function requires:

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

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

8.5.7.4 2-stage configuration with fan-coils as auxiliary stage

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

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

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

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

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

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

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

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

8.5.7.5 Remote fan speed modification

The communication objects shown in Figure 43 allow to monitor the actual fan speed forced automatically by the temperature controller or or by means of a supervisor system. The communication objects (from now on: C.O.) also allow to perform the same modifications remotely, for example from a supervisor software.

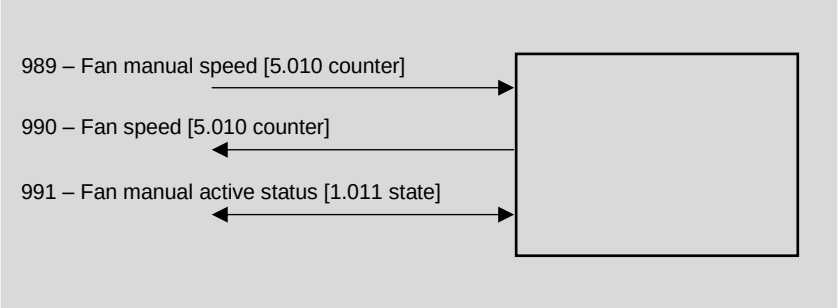


Figure 43 - Remote fan speed adjustment

O.C. *990 - Fan speed* allows you to evaluate the current fan speed; O.C. 991 - *Fan manual active status* contains the information about automatic (=0, not active) or manual (=1, active) operating mode. By modifying the O.C. *989 – 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 O.C. 991 (= 0, not active).

Accepted values for O.C.s with index 989, 990 and 991 depend on the number of speeds set in ETS for the Ventilation Control type.

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

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

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

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

8.5.8 Relative humidity control

The ***Relative Humidity Control*** tab contains the following sub-tabs:

- Dehumidification
- Humidification
- Calculated psychrometric values

The ***Dehumidification***, ***Humidification*** and **Calculated Psychrometric Values** sub-tabs only appear if an external (from bus) relative humidity sensor is enabled.

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

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

8.5.8.1 Dehumidification

When enabled, the ***Dehumidification*** secondary tab contains the following parameters:

- Dehumidification function (operating modes where dehumidification is active)
- Relative humidity setpoint for dehumidification control [%]
- Dehumidification control hysteresis [%]
- Cyclic sending interval
- Disable dehumidification control from bus
- Signal from bus
- Dehumidification subordinated to temperature control
- Dehumidification start delay
- Sensible heat integration function
- Temperature difference for integration
- Hysteresis for integration

8.5.8.1.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Dehumidification function	⇒ temperature control⇒ settings Thermostat function = both heating and cooling	**Disabled** cooling only heating only both heating and cooling
	Temperature control⇒ Settings⇒ Thermostat function = heating	**disabled** / heating only
	Temperature control⇒ Settings⇒ Thermostat function = cooling	**disabled** / cooling only
	Parameter that selectively enables the dehumidification function.	

Parameter name	*Conditions*	*Settings*
Humidity setpoint (for dehumidification control)[%]	Dehumidification function ≠ disabled	**55** [range 20 … 80]
Humidity hysteresis	Dehumidification function ≠ disabled	**0.8** [other values in the range 0.5 … 4%]
Cyclic sending interval	Dehumidification function ≠ disabled	**No transmission** [other values in the range 30 s … 120 min]
Disable dehumidification control from bus	Dehumidification function ≠ disabled	**no** / yes
Signal from bus	Disable dehumidification control from the bus = yes	**not inverted** / inverted
Subordinated to temperature control	⇒ temperature control⇒ settings Thermostat function = cooling or both heating and cooling, ⇒ temperature control⇒ cooling type = floor radiant panel or ceiling radiant panel, Relative humidity control⇒ dehumidification⇒ Dehumidification function = cooling only	**no** / yes
Dehumidification start delay	Subordinate to temperature control = no	**00:05:00 hh:mm:ss** [range 00:00:00 … 04:39:37]
	The value 00:00:00 means that the start delay is not enabled.	
Integration		**no** / yes
Temperature difference for integration	Integration = yes	**1.5** [other values in the range 0.5 … 3 °C]
Hysteresis for integration	Integration = yes	**0.5 K** [other values in the range 0.2 K … 3 K]

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Relative humidity setpoint for dehumidification	Dehumidification function ≠ disabled	2 bytes	CRWTU	[9.007] humidity (%)	995, 1070, 1145, 1220, 1295, 1370, 1445, 1520.
Thermostat - Dehumidification command	Dehumidification function ≠ disabled	1 bit	CR-T-	[1.001] switch	997, 1072, 1147, 1222, 1297, 1372, 1447, 1522.
Thermostat - Dehumidification command - Water battery	⇒ temperature control⇒ settings Thermostat function = both heating and cooling, ⇒ temperature control⇒ cooling type = floor radiant panel or ceiling radiant panel,	1 bit	CR-T-	[1.001] switch	998, 1073, 1148, 1223, 1298, 1373, 1448, 1523.

Object name	Conditions	Size	Flags	DPT	CO number(s)
	Relative humidity control⇒ dehumidification⇒ Dehumidification function = cooling only				
Thermostat - Dehumidification integration control	⇒ temperature control⇒ settings Thermostat function = heating and cooling, Temperature control⇒ Cooling⇒ Cooling type = floor radiant or ceiling radiant panel Relative humidity control⇒ dehumidification⇒ Dehumidification function = cooling only Integration = yes	1 bit	CR-T--	[1.001] switch	999, 1074, 1149, 1224, 1299, 1374, 1449, 1524.
	The object is set to ON if the relative humidity detected exceeds the setpoint and the room temperature exceeds the setpoint of the temperature difference value for integration activation.				
Thermostat - Dehumidification control disable	Disable dehumidification control from the bus = yes	1 bit	C-W--	[1.002] boolean	1000, 1075, 1150, 1225, 1300, 1375, 1450, 1525.

8.5.8.2 Humidification

The ***Humidification*** sub-tab contains the following parameters:

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

8.5.8.3.1 Parameters and communication objects

Parameter name	Conditions	Settings
Humidification function		**disabled** heating only cooling only Both heating and cooling
	Parameter that selectively enables the humidification function.	
Relative humidity setpoint (for humidification control) [%]	Humidification ≠ disabled	**35** [range 20 … 80 %]
Humidity hysteresis	Humidification ≠ disabled	**0.8** [other values in the range 0.5 … 4%]
Cyclic sending interval	Humidification ≠ disabled	**No transmission** [other values in the range 30 s … 120 min]

Parameter name	Conditions	Settings
Disable humidification control from the bus	Humidification ≠ disabled	**no** / yes
Signal from bus	Humidification ≠ disabled Disable humidification control from bus = yes	**not inverted** / inverted

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Relative humidity setpoint for humidification	Humidification ≠ disabled	2 bytes	CRWTU	[9.007] humidity (%)	996, 1071, 1146, 1221, 1296, 1371, 1446, 1521.
Thermostat - Humidification command	Humidification ≠ disabled	1 bit	CR-T-	[1.001] switch	1001, 1076, 1151, 1226, 1301, 1376, 1451, 1526.
Thermostat - Humidification control disable	Disable humidification control from bus = yes	1 bit	C-W--	[1.002] boolean	1002, 1077, 1152, 1227, 1302, 1377, 1452, 1527

8.5.8.3 Calculated psychrometric values

The ***Calculated Psychrometric Values*** secondary tab contains the following parameters:

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

Tab display condition: External sensors⇒ Relative humidity sensor = enabled.

8.5.8.3.2 Parameters and communication objects

Parameter name	Conditions	Settings
Dew point temperature		**disabled** / enabled
	Sending the dew point temperature value to the bus allows active anti-condensation protection to be implemented by recalibrating the heat transfer fluid delivery conditions if there is a control device for the mixing unit on the bus. If the thermostat is installed in an environment where cooling is not required (e.g. bathroom), it is advisable to exclude the environment from control by setting the Dew point temperature parameter to disabled.	
Cyclic sending interval	Dew point temperature = enabled	**no sending** [other values in the range 30 s ... 120 min]

Parameter name	*Conditions*	*Settings*
Min. value change for sending [K]	Dew point temperature = enabled	**0.2 K** / no transmission [other values in the range 0.2 … 3 K]

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat – Dew-point temperature	Dew-point temperature = enabled	2 bytes	CR-T-	[9.001] temperature °C	994, 1069, 1144, 1219, 1294, 1369, 1444, 1519.

8.5.9 Energy saving

To implement energy saving functions, contacts can be used to detect when windows are opened, presence sensors and card holders.

The ***Energy Saving*** tab contains the following secondary tabs:

- Window contacts
- Presence sensors
- Card holder

Tab display condition:

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

8.5.9.1 Windows contacts

The secondary ***Windows Contacts*** tab can be configured if at least one sensor dedicated to this function is enabled, i.e. if the following condition is met:

- *External sensors (from the bus)⇒ Window contact 1 and/or 2* = enabled

The ***Window contacts*** tab contains the following parameters:

- Window contact function
- Wait time to building protection mode

8.5.9.1.1 Parameters and communication objects

Parameter name	***Conditions***	***Settings***
Window contacts function		**disabled** / enabled
	Parameter that enables the window contact function.	
Wait time to building protection mode	Window contacts function = enabled	**00:01:00 hh:mm:ss** [range 00:00:00 … 04:39:37]
	Time interval before the device automatically switches to building protection mode.	

Object name	***Conditions***	***Size***	***Flags***	***DPT***	***CO number(s)***
Thermostat – Windows contact sensor 1 (from bus)	Window contact function = enabled	1 bit	C-WTU	[1.019] window/door	941, 1016, 1091, 1166, 1241, 1316, 1391, 1466.
Thermostat – Windows contact sensor 2 (from bus)	Window contact function = enabled	1 bit	C-WTU	[1.019] window/door	942, 1017, 1092, 1167, 1242, 1317, 1392, 1467.

8.5.9.2 Presence sensors

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

- Presence sensor function
- Use of presence sensors
- Thermostat modes
- Absence time for switching the HVAC mode

External sensors (from the bus) such as the EK-SG2-TP-M motion sensor or the EK-SG2-TP-P presence sensor can be used for this function.

The following condition must therefore be verified:

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

8.5.9.3.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Presence sensor function		**disabled** / enabled
	Parameter that enables the presence sensor function.	
Presence sensors use	Presence sensor function = enabled	**comfort extension** comfort limitation Comfort extension and comfort limitation
Thermostat modes	Presence sensor function = enabled Presence sensors use = comfort extension and comfort limitation, or comfort limitation	**comfort-standby** comfort-economy
Absence time to switch HVAC mode	Presence sensor function = enabled	**00:01:00 hh:mm:ss** [range 00:00:00 … 04:39:37]
	Time interval before automatic mode switching set in the Thermostat modes parameter.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Presence sensor 1 (from bus)	Presence sensor function = enabled	1 bit	C-WTU	[1,018] occupancy	943, 1018, 1093, 1168, 1243, 1318, 1393, 1468.
Thermostat - External presence sensor 2 (from bus)	Presence sensor function = enabled	1 bit	C-WTU	[1.018] occupancy	944, 1019, 1094, 1169, 1244, 1319, 1394, 1469.

8.5.9.3 Card holder pocket

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

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

The ***Card holder*** tab contains the following parameters:

- Card holder function
- HVAC mode to switch to when the card is inserted
- Activation delay when the card is inserted
- HVAC mode to switch to when the card is removed
- Activation delay when the card is removed

8.5.9.3.2 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Card holder function		**disabled** / enabled
	Parameter that enables the card holder function.	
On card insertion, switch HVAC mode to (*)	Card holder function = enabled	none **comfort** standby economy
	Parameter that defines which operating mode the device should automatically switch to when the card is inserted into the card holder. *(*) Note: if Setpoint type = single has been selected in the "Temperature control" tab, this parameter is set to "none", as operating modes are not managed.*	
Activation delay on card insertion	Card holder function = enabled	**00:00:00 hh:mm:ss** [range 00:00:00 … 04:39:37]
	Time interval before automatic switching of the operating mode when the card is inserted into the card holder.	
On card removal, switch HVAC mode to (*)	Card holder function = enabled	None **standby** economy building protection
	Parameter that defines which operating mode the device should automatically switch to when the card is removed from the card holder. *(*) Note: if Setpoint type = single has been selected in the "Temperature control" tab, this parameter is set to "Building protection", as operating modes are not managed.*	
Activation delay on card removal	Card holder function = enabled	**00:00:00 hh:mm:ss** [range 00:00:00 … 04:39:37]
	Time interval before automatic switching of the operating mode when the card is removed from the card holder.	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Card holder contact (from bus)	Card holder function = enabled	1 bit	C-WTU	[1.018] occupancy	945, 1020, 1095, 1170, 1245, 1320, 1395, 1470.

Note on card holder function

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

Conventional (not KNX) card holder

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

KNX card holder

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

8.6 Logic functions

For a comprehensive discussion of the logic function, refer to paragraph 6.4.8.

The communication objects for a specific logic function are available only if the Logic Module is enabled, the logic function is enabled and finally if at least one input has been enabled.

8.6.1.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Logic function n	Number of logic functions ≠ 0	**disabled** enabled
	It allows to enable a specific logic function. *n=1,…,8.*	
Logic operation type	Logic function n = enabled	**OR** AND XOR NOR NAND XNOR IDN
	Defines the logical operation to be performed on the input objects. *n=1,…,8.*	
Read delay after bus recovery	Logic function n = enabled	mm:ss.fff (**00:01.000**) [range 00.00.000 … 05.00.000]
	After bus recovery, the device waits the specified time before evaluating the logical objects configured as inputs; for those for which a value has not yet been received at the end of the time, a request is sent on the bus. *The maximum value is 05:00.000.* *n=1,…,8.*	
Logic object *x*	Logic function n = enabled	**disabled** / enabled
	Defines which of the available logical objects to use as inputs. *Logical objects configured as disabled are ignored and their communication objects are not generated.* *n=1,…,8. x= 1,…,8.*	
Logic object *x* – Logic object *x negated*	Logic function n = enabled Logic object x = enabled	**no** / yes
	Applies a logical negation to the value of the logical object. *n=1,…,8. x= 1,…,8.*	

The Logic function calculation is performed only if and when at least one of the input logic objects is updated.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Logic function n - Logic object *x*	Logic function n = enabled Logic object x = enabled	1 bit	C-W--	[1.001] switch	Function 1: 477,…,484 Function 2: 486,…,493 Function 3: 495,…,502 Function 4: 504,…,511 Function 5: 513,…,520 Function 6: 522,…,529 Function 7: 531,…,538 Function 8: 540,…,547
	n=1,…,8. x= 1,…,8. *These are the objects related to the inputs for each logical function.*				
Logic function n – Logic object status	Logic function n = enabled Logic object x = enabled	1 bit	C-W--	[1.001] switch	485, 494, 503, 512, 521, 530, 539, 548.
	n=1,…,8. x= 1,…,8. *These are the objects related to the outputs for each logical function..*				

8.7 Comparison functions

For a comprehensive discussion of comparison functions, see section 6.3.7.

The status object for a specific comparison function is only available if the Logic Module is enabled and the comparison function n (n=1,…,8) is enabled.

Additionally, depending on the comparison type set, new communication objects can be enabled, internal objects or numeric values can be used.

8.7.1.1 Parameters and communication objects

Parameter name	Conditions	Settings
Comparison function n	Number of compare functions ≠ 0	**disabled** enabled
	It allows to enable a specific comparison function. *n=1,…,8.*	
Comparison operation	Comparison function n = enabled	**=** != < > >= <=
	It defines the comparison operation to be calculated. *n=1,…,8.*	
Hysteresis	Comparison function n = enabled	0… 15 % (**0**)
	Defines a hysteresis value to be used in the comparison operation between the 2 values.. *n=1,…,8.*	

Parameter name	*Conditions*	*Settings*
Comparison 1	Comparison function n = enabled	**Communication object** Internal object
	Allows you to set whether the first comparison object is a communication or internal object. *n=1,…,8.*	
DPT	Comparison function n = enabled Comparison 1 = communication object	**1 bit (DPT 1.001)** 2 bit (DPT 2.001) 4 bit (DPT 3.008) 1 byte scaling (DPT 5.001) 1 byte signed (DPT 6.010) 1 byte unsigned (DPT 5.010) 2 byte signed (DPT 8.001) 2 byte unsigned (DPT 7.001) 2 byte float (DPT 9.002)
	Expresses the data type of comparison term 1, when it is set as a communication object. *n=1,…,8.*	
Internal value	Comparison function n = enabled Comparison 1 = internal object	**Internal type**
	By selecting "internal object", the type matches that of the object selected subsequently.	
Internal type	Comparison function n = enabled Comparison 1 = internal object	**Output value** Output channel value
	It allows you to select the type of internal object, between independent output and channel (coupled output).	
Output number	Comparison function n = enabled Comparison 1 = internal object Internal type = output value	1…8 **(1)**
	It selects the independent output as the internal object.	
Output channel number	Comparison function n = enabled Comparison 1 = internal object Internal type = output channel value	A, B, C, D **(A)**
	It selects the channel (coupled output) as the internal object.	
Comparison 2	Comparison function n = enabled	**Communication object** Numeric value
	It allows to set whether the second comparison object is a communication object or a numeric value. *n=1,…,8.*	

Parameter name	*Conditions*	*Settings*
DPT	Comparison function n = enabled Comparison 2 = communication object	**1 bit (DPT 1.001)** 2 bit (DPT 2.001) 4 bit (DPT 3.008) 1 byte scaling (DPT 5.001) 1 byte signed (DPT 6.010) 1 byte unsigned (DPT 5.010) 2 byte signed (DPT 8.001) 2 byte unsigned (DPT 7.001) 2 byte float (DPT 9.002)
	Expresses the data type of comparison term 2, when it is set as a communication object. *n=1,…,8.*	
Valore	Comparison function n = enabled Comparison 2 = numeric value	**0** [-671088…670597]
	Expresses the data type of comparison term 2, when it is set as a numeric value. *n=1,…,8.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Compare n - Object 1 *[DataType]*	Comparison function n = enabled Comparison 1 = communication object	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	549, 552, 555, 558, 561, 564, 567, 570.
	n=1,…,8. *Types and sizes of individual objects can be configured as described in Table 1.*				
Compare n - Object 2 *[DataType]*	Comparison function n = enabled Comparison 2 = communication object	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	550, 553, 556, 559, 562, 565, 568, 571.
	n=1,…,8. *Types and sizes of individual objects can be configured as described in Table 1*				
Status compare *n*	Comparison function n = enabled	1 bit	CR-T-	[1.001] switch	551, 554, 557, 560, 563, 566, 569, 572.
	n=1,…,8. *It expressed the result of the comparison operation, such as True or False.*				

The data sizes and Data Point Types are as follows:

Dimension	*DPT*
1 bit	[1.001] 1-bit, switch
2 bit	[2.001] 1-bit controlled, switch control
4 bit	[3.008] blind control
1 byte scaling	[5.001] 8-bit unsigned value, counter pulses
1 byte signed	[6.010] 8-bit signed value, counter pulses
1 byte unsigned	[5.010] 8-bit unsigned value, counter pulses
2 byte signed	[8.001] 2-byte signed value, pulses difference
2 byte unsigned	[8.001] 2-byte unsigned value, pulses
2 byte floating	[9.002] 2-byte float value, temperature difference

Table 4 - DPT for comparison objects

8.8 Arithmetic functions

For a comprehensive discussion of arithmetic functions, see section 6.4.10.

The status object for a specific arithmetic function is only available if the Logic Module is enabled and the arithmetic function n (n=1,…,8) is enabled.

Additionally, depending on the type of operation set, you can enable new communication objects, use internal objects, or use numeric values.

8.8.1.1 Parameters and communication objects

Parameter name	*Conditions*	*Settings*
Arithmetic function n	Number of arithmetic functions ≠ 0	**disabled** enabled
	Allows you to enable a specific arithmetic function. *n=1,…,8.*	
Operation type	Arithmetic function n = enabled	**Minimum** Maximum Average Sum Difference Multiplication Division
	Defines the arithmetic operation to be performed. *n=1,…,8.*	
Read delay after bus recovery	Arithmetic function n = enabled	00:00:.000 … 05:00.000 (**00:01.000**)
	Defines a delay in reading operands when the bus restarts. *n=1,…,8.*	

Parameter name	*Conditions*	*Settings*
DPT of the result	Arithmetic function n = enabled	**1 bit value (DPT 1.001)** 2 bit value (DPT 2.001) 4 bit value (DPT 3.008) 1 byte scaling (DPT 5.001) 1 byte signed value (DPT 6.010) 1 byte unsigned value (DPT 5.010) 2 byte signed value (DPT 8.001) 2 byte unsigned value (DPT 7.001) 2 byte floating value (DPT 9.002) 4 byte signed value (DPT 13.001) 4 byte unsigned value (DPT 12.001) 4-byte floating value (DPT 14.056)
	Defines the data type for the result of the operation. *n=1,…,8.*	
Arithmetic object x	Arithmetic function n = enabled	**disabled** enabled
	n=1,…,8. *x=1,…,8.* *Enables the operand x.* *For the division operation, only arithmetic objects 1 and 2 can be enabled.*	
DPT	Arithmetic function n = enabled Arithmetic object x = enabled	**1 bit value (DPT 1.001)** 2 bit value (DPT 2.001) 4 bit value (DPT 3.008) 1 byte scaling (DPT 5.001) 1 byte signed value (DPT 6.010) 1 byte unsigned value (DPT 5.010) 2 byte signed value (DPT 8.001) 2 byte unsigned value (DPT 7.001) 2 byte floating value (DPT 9.002) 4 byte signed value (DPT 13.001) 4 byte unsigned value (DPT 12.001) 4-byte floating value (DPT 14.056)
	n=1,…,8. *x=1,…,8.* *Defines the data type for the operand x.*	

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Arithmetic function n - Object x value	Arithmetic function n = enabled Arithmetic object x = enabled	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	573, 574, 575. 576, 577, 578. 579, 580. 583, 584, 585, 586, 587, 588, 589. 590. 593, 594, 595, 596, 597, 598, 599, 600. 603, 604, 605, 606, 607, 608, 609, 610. 613, 614, 615, 616, 617, 618, 619, 620. 623, 624, 625, 626, 627, 628, 629, 630. 633, 634, 635, 636, 637, 638, 639, 640. 643, 644, 645, 646, 647, 648, 649. 650.
	n=1,…,8. *x=1,…,8.* *The types and sizes of individual objects can be configured as described below.*				
Arithmetic function n - Result	Arithmetic function n = enabled	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	581, 591, 601, 611, 621, 631, 641, 651.
	n=1,…,8. *x=1,…,8.* *The types and sizes of individual objects can be configured as described below..*				
Arithmetic function n - Error	Arithmetic function n = enabled	1 bit	CR-T-	[1.001] alarm	582, 592, 602, 612, 622, 632, 642, 652.
	n=1,…,8. *Triggers an alarm if the operation results in an error.*				

The data dimensions and data point types are as follows:

Dimension	*DPT*
1 bit	[1.001] 1-bit, switch
2 bit	[2.001] 1-bit controlled, switch control
4 bit	[3.008] blind control
1 byte scaling (percentage)	[5.001] 8-bit unsigned value, counter pulses
1 byte signed	[6.010] 8-bit signed value, counter pulses
1 byte unsigned	[5.010] 8-bit unsigned value, counter pulses
2 byte signed	[8.001] 2-byte signed value, pulses difference
2 byte unsigned	[8.001] 2-byte unsigned value, pulses
2 byte floating	[9.002] 2-byte float value, temperature difference
4 byte signed	[13.001] 4-byte signed value, counter pulses (signed)
4 byte unsigned	[12.001] 4-byte unsigned value, counter pulses

Dimension	*DPT*
	(unsigned)
4 byte floating	[14.056] 4-byte float value, power (W)

Table 5 - DPT arithmetic objects

9 Appendix

9.1 Communication objects table

Following is a list of KNX communication objects with the corresponding Data Point Types (DPT) defined by the application program based on the configurations made.

The list is generally sorted by object number.

The communication object numbers in the "Comm. Object No." column of each table are divided into two groups. The objects in the second group (in italics) relate only to the 16-output EK-FI1-TP-I device.

Indexes:

- n= A, B, C, D (E, F, G, H for the EK-FI1-TP-I device).
- x=1, 2, 3, 4, 5, 6, 7, 8 (9, 10, 11, 12, 13, 14, 15, 16 for the EK-FI1-TP-I device).

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Manual commands disabled	Manual operation = enabled Manual disable from bus = enabled	1 bit	C-W--	[1.001] switch	1
Output channel n – Move up/down command	Output channel n = 3-way valve / shutter / Venetian blind Use = shutter or venetian blind	1 bit	C-W--	[1.008] up/down	2, 16, 30, 44, *58, 72,* *88, 100.*
Output channel n – stop-step up/down command	Output channel n = 3-way valve / shutter / Venetian blind Use = shutter or venetian blind	1 bit	C-W--	[1.007] step	3, 17, 31, 45, *59, 73,* *87, 101.*
Output channel n – Dedicated stop command	Output channel n = 3-way valve / shutter / Venetian blind Use = all	1 bit	C-W--	[1.017] trigger	4, 18, 32, 46, *60, 74,* *88, 102.*
Output channel n – Info move	Output channel n = 3-way valve / shutter / Venetian blind Use = all	1 bit	CR-T-	[1.011] state	5, 19, 33, 47, *61, 75,* *89, 103.*
Output channel n – Valid current abs position	Output channel n = 3-way valve / shutter / Venetian blind Use = all	1 bit	CR-T-	[1.002] boolean	6, 20, 34, 48, *62, 76,* *90, 104.*
Output channel n – Dimmer blind position command	Output channel n = 3-way valve / shutter / Venetian blind Use = Venetian blind Position control with dimmer = yes	4 bit	C-W--	[3.008] blind control	7, 21, 35, 49, *63, 77,* *91, 105.*
Output channel n – Dimmer shutter position command	Output channel n = 3-way valve / shutter / Venetian blind Use = shutter Position control with dimmer = yes	4 bit	C-W--	[3.008] blind control	7, 21, 35, 49, *63, 77,* *91, 105.*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output channel n – Dimmer valve position command	Output channel n = 3-way valve / shutter / Venetian blind Use = Valve Position control with dimmer = yes	4 bit	C-W--	[3.008] blind control	7, 21, 35, 49, *63, 77, 91, 105.*
Output channel n – Abs [valve/shutter/blind] position command	Output channel n = 3-way valve / shutter / Venetian blind Use = all	1 bit	C-W--	[5.001] percentage (0..100%)	8, 22, 36, 50, *64, 78, 92, 106.*
Output channel n – Abs [valve/shutter/blind] status command	Output channel n = 3-way valve / shutter / Venetian blind Use = all	1 bit	CR-T-	[5.001] percentage (0..100%)	9, 23, 37, 51, *65, 79, 93, 107.*
Output channel n – Dimmer slats command	Output channel n = 3-way valve / shutter / Venetian blind Use = Venetian blind Slats control with dimmer = yes	4 bit	C-W--	[3.008] blind control	10, 24, 38, 52, *66, 80, 94, 108.*
Output channel n – Abs slat position command	Output channel n = 3-way valve / shutter / Venetian blind Use = Venetian blind	1 bit	C-W--	[5.001] percentage (0..100%)	11, 25, 39, 53, *67, 81, 95, 109.*
Output channel n – Abs slat position status	Output channel n = 3-way valve / shutter / Venetian blind Use = Venetian blind	1 bit	CR-T-	[5.001] percentage (0..100%)	12, 26, 40, 54, *68, 82, 96, 110.*
Output channel n – Wind alarm	Output channel n = 3-way valve / shutter / Venetian blind Use = shutter or venetian blind Meteo alarms = enabled	1 bit	C-W--	[1.005] alarm	13, 27, 41, 55, *69, 83, 97, 111.*
Output channel n – Frost alarm	Output channel n = 3-way valve / shutter / Venetian blind Use = shutter or venetian blind Meteo alarms = enabled	1 bit	C-W--	[1.005] alarm	14, 28, 42, 56, *70, 84, 98, 112.*
Output channel n – Rain alarm	Output channel n = 3-way valve / shutter / Venetian blind Use = shutter or venetian blind Meteo alarms = enabled	1 bit	C-W--	[1.005] alarm	15, 29, 43, 57, *71, 85, 99, 113.*

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output x – On/off command	Output channel n = independent or single valve function = disabled	1 bit	C-W--	[1.001] switch	170, 173, 176, 179, 182,185, 188, 191, *194, 197, 200, 203, 206, 209, 212, 215.*
Output x – Valve continuous command	Output channel n = independent or single valve function = enabled	8 bit unsigned	C-W--	[5.001] percentage (0…100%)	171, 174, 177, 180, 183, 186, 189, 192, *195, 198, 201, 204, 207, 210, 213, 216.*
Output x – On/off status	Output channel n = independent or single AND Status feedback telegram = enabled	1 bit	CR-T-	[1.001] switch	172, 175, 178, 181, 184, 187, 190, 193, *196, 199, 202, 205, 208, 211, 214, 217.*
Output channel n – Direction status output x	Output channel n = independent or single AND Status feedback telegram = enabled Use = all	1 bit	CR-T-	[1.001] switch	172, 175, 178, 181, 184, 187, 190, 193, *196, 199, 202, 205, 208, 211, 214, 217.*
Output *x* – Lock command	Output channel n = independent or single Output x = enabled Locking function = enabled	1 bit	C-W--	[1.003] enable	242, 243, 244, 245, 246, 247, 248, 249, *250, 251, 252, 253, 254, 255, 256, 257.*
Output channel n – Lock command	Output channel n = 3-way valve / shutter / Venetian blind Use = all Locking function = enabled	1 bit	C-W--	[1.003] enable	242, 244, 246, 248, *250, 252, 254, 256.*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Output *x* – Forcing command	Output channel n = independent or single Output x = enabled Forcing function = enabled	2 bit	C-W--	[2.001] switch control	266, 267, 268, 269, 270, 271, 272, 273, *274, 275 276, 277, 278, 279, 280, 281.*
Output channel n – Forcing command	Output channel n = 3-way valve / shutter / Venetian blind Use = all Forcing function = enabled	2 bit	C-W--	[2.001] switch control	266, 268, 270, 272, *274, 276, 278, 280.*
Output *x* – Staircase lighting start stop command	Output channel n = independent or single Output x = enabled Staircase function = enabled	1 bit	C-W--	[1.010] start/stop	290, 291, 292, 293, 294, 295, 296, 297, *298, 299, 300, 301, 302, 303,* 304, 305.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output *x* – Scene number	Output channel n = independent or single Output x = enabled Scenes function = enabled	1 Byte	C-W--	[17.001] scene number [18.001] scene control	314, 315, 316, 317, 318, 319, 320, 321, *322, 323, 324, 325, 326, 327, 328, 329.*
Output channel n – Scene number	Output channel n = 3-way valve / shutter / Venetian blind Use = all Scenes function = enabled	1 Byte	C-W--	[17.001] scene number [18.001] scene control	314, 316, 318, 320, *322, 324, 326, 328.*
Output *x* – Energy counter (Wh)	Output channel n = independent or single Output x = enabled Operating energy/time counter = enabled	4-byte signed value	CR-T-	[13.0130 active energy [Wh]	338, 344, 350, 356, 362, 368, 374, 380, *386, 392, 398, 404, 410, 416, 422, 428.*
Output *x* – Energy counter reset command	Output channel n = independent or single Output x = enabled Operating energy/time counter = enabled	1 bit	C-W--	[1.015] reset	339, 345, 351, 357, 363, 369, 375, 381, *387, 393, 399, 405, 411, 417, 423, 429.*
Output *x* – Operating hours counter	Output channel n = independent or single Output x = enabled Operating energy/time counter = enabled	2-byte unsigned value	CR-T-	[7.007] time [h]	340, 346, 352, 358, 364, 370, 376, 382, *388, 394, 400, 406, 412, 418,* 424, 430.
Output *x* – Operating seconds counter	Output channel n = independent or single Output x = enabled Operating energy/time counter = enabled	4-byte signed value	CR-T-	[13.100] time lag [s]	341, 349, 353, 361, 365, 373, 377, 385, *389, 397, 401, 409, 413, 421, 425, 433.*
Output *x* – Hours counter reset	Output channel n = independent or single Output x = enabled Operating energy/time counter = enabled	1 bit	C-W--	[1.015] reset	342, 347, 354, 359, 366, 371, 378, 383, *390, 395, 402, 407, 414, 419, 426, 431.*

Object name	Conditions	Size	Flags	DPT	CO number(s)
Output *x* – Hours counter overflow	Output channel n = independent or single Output x = enabled Operating energy/time counter = enabled	1 bit	CR-T-	[1.005] alarm	343, 348, 355, 360, 367, 372, 379, 384, *391, 396, 403, 408, 415, 420,* 427, 432.
Logic function 1 - Logic object *x*	Logic function 1 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	477, 478, 479, 480, 481, 482, 483, 484.
Logic function 1 - Logic object status	Logic function 1 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	485
Logic function 2 - Logic object *x*	Logic function 2 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	486, 487, 488, 489, 490, 491, 492, 493.
Logic function 2 - Logic object status	Logic function 2 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	494
Logic function 3 - Logic object *x*	Logic function 3 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	495, 496, 497, 498, 499, 500, 501, 502.
Logic function 3 - Logic object status	Logic function 3 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	503
Logic function 4 - Logic object *x*	Logic function 4 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	504, 505, 506, 507, 508, 509, 510, 511.
Logic function 4 - Logic object status	Logic function 4 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	512
Logic function 5 - Logic object *x*	Logic function 5 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	513, 514, 515, 516, 517, 518, 519, 520.
Logic function 5 - Logic object status	Logic function 5 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	521
Logic function 6 - Logic object *x*	Logic function 6 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	522, 523, 524, 525, 526, 527, 528, 529.
Logic function 6 - Logic object status	Logic function 6 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	530
Logic function 7 - Logic object *x*	Logic function 7 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	531, 532, 533, 534, 535, 536, 537, 538.

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Logic function 7 - Logic object status	Logic function 7 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	539
Logic function 8 - Logic object *x*	Logic function 8 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	C-W--	[1.001] switch	540, 541, 542, 543, 544, 545, 546, 547.
Logic function 8 - Logic object status	Logic function 8 = enabled Logic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] switch	548
Compare n – Object 1 *[Data Type]* *n=1,…,8*	Comparison function n = enabled Comparison 1 = communication object	See *Table 1*	C-W--	See *Table 1*	549, 552, 555, 558, 561, 564, 567, 570.
Compare n – Object 2 *[Data Type]* *n=1,…,8*	Comparison function n = enabled Comparison 2 = communication object	See **Errore. L'origine riferimento non è stata trovata.**	C-W--	See **Errore. L'origine riferimento non è stata trovata.**	550, 553, 556, 559, 562, 565, 568, 571.
Comparison state n *n=1,…,8*	Comparison function n = enabled	1 bit	CR-T-	[1.001] switch	551, 554, 557, 560, 563, 566, 569, 572.
Arithmetic function 1 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	573, 574, 575. 576, 577, 578. 579, 580.
Arithmetic function 1 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	581
Arithmetic function 1 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	582
Arithmetic function 2 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	583, 584, 585, 586, 587, 588, 589. 590.
Arithmetic function 2 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	591
Arithmetic function 2 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	592

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Arithmetic function 3 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	593, 594, 595, 596, 597, 598, 599, 600.
Arithmetic function 3 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	601
Arithmetic function 3 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	602
Arithmetic function 4 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	603, 604, 605, 606, 607, 608, 609, 610.
Arithmetic function 4 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	611
Arithmetic function 4 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	612
Arithmetic function 5 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	613, 614, 615, 616, 617, 618, 619, 620.
Arithmetic function 5 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	621
Arithmetic function 5 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	622
Arithmetic function 6 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	623, 624, 625, 626, 627, 628, 629, 630.
Arithmetic function 6 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	631
Arithmetic function 6 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	632

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Arithmetic function 7 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	633, 634, 635, 636, 637, 638, 639, 640.
Arithmetic function 7 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	641
Arithmetic function 7 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	642
Arithmetic function 8 - Object x value	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	C-W--	Depending on the configuration **([1.001] switch)**	643, 644, 645, 646, 647, 648, 649. 650.
Arithmetic function 8 - Result	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	Depending on the configuration **1 bit**	CR-T-	Depending on the configuration **([1.001] switch)**	651
Arithmetic function 8 - Error	Arithmetic function 1 = enabled Arithmetic object x = enabled (x = 1,…,8)	1 bit	CR-T-	[1.001] alarm	652
Input x – Lock command	Input x = digital input Lock function = enabled	1 bit	C-W-U	[1.003] enable	653, 659, 665, 671, 677, 683, 689, 695 *701, 707, 713, 719, 725, 731, 737, 743*
Input channel *x* – Lock command	Input channel x = coupled Lock function = enabled	1 bit	C-W-U	[1.003] enable	653, 665, 677, 689, *701, 713, 725, 737.*
Input x – Switch command	Input x = digital input Type = dimming	1 bit	C-WTU	[1.001] switch	654, 660, 666, 672, 678, 684, 690, 696 *702, 708, 714, 720, 726, 732, 738, 744.*
Input channel *x* – Switch command	Input channel x = coupled Type = dimming	1 bit	C-WTU	[1.001] switch	654, 666, 678, 690, 702, 714, 726, 738.
Input *x* – Dedicated stop command	Input channel x = coupled Type = shutter or Venetian blinds Venetian blind mode = disabled	1 bit	C—T-	[1.017] trigger	654, 660, 666, 672, 678, 684, 690, 696 *702, 708, 714, 720,* 726, 732, 738, 744.
Input x - Stop/step up/down command	Input x = digital input Type = Shutters or venetian blinds Venetian blind mode = enabled	1 bit	C--T-	[1.007] step	654, 660, 666, 672, 678, 684, 690, 696 *702, 708, 714, 720, 726, 732, 738, 744*
Input channel *x* – Dimming up / down / stop command	Input channel x = coupled Type = dimming	4 bit	C--T-	[3.*] 3-bit control	655, 667, 679, 691, *703, 715, 727, 739.*
Input *x* – Dimming up / down / stop command	Input x = digital input Type = dimming	4 bit	C--T-	[3.*] 3-bit control	655, 661, 667, 673, 679, 685, 691, 697 *703, 709, 715, 721, 727, 733, 739, 745.*

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Input x - Move up / down command	Input x = digital input Type = Shutters or venetian blinds	1 bit	C--T-	[1.008] up/down	656, 662, 668, 674, 680, 686, 692, 698 *704, 710, 716, 722, 728, 734, 740, 746.*
Input channel *x* – Move up/down command	Input channel x = coupled Type = shutter or Venetian blinds	1 bit	C--T-	[1.008] up/down	656, 668, 680, 692, *704, 716, 728, 740.*
Input *x* – Scene number	Input x = digital input Type = scene	1 Byte	C--T-	[17.*] Scene number [18.*] Scene control	657, 663, 669, 675, 681, 687, 693, 699 *705, 711, 717, 723, 729, 735, 741, 747.*
Input channel *x* – Dedicated stop command	Input channel x = coupled Type = shutter or Venetian blinds Venetian blind mode = disabled	1 bit	C--T-	[1.017] trigger	657, 669, 681, 693, *705, 717, 729, 741.*
Input channel x - Stop/step up/down command	Input x = digital input Type = Shutters or venetian blinds Venetian blind mode = disabled	1 bit	C--T-	[1.007] step	657, 669, 681, 693, *705, 717, 729, 741.*
Input *x* – Shift register	Input x = digital input Type = shift register	1 Byte	CR-T-	[5.010] counter pulses (0.255)	658, 664, 670, 676, 682, 688, 694, 700 *706, 712, 718, 724, 730, 736, 742, 748.*
Input channel *x* – Switching command	Input channel x = coupled Type = switching	1-bit	C-WTU	[1.001] switch	658, 670, 682, 694, *706, 718, 730, 742.*
Input *x* – switching Status *[type]*, object *no.*	Input *x* = digital input Type = switching	*depending on configuration* (**1-bit**)	C-WTU	*depending on configuration* (**[1.001] switch**)	749, 757, 765, 773, 781, 789, 797, 805 *813, 821, 829, 837, 845, 853, 861, 869*
Input x – Temperature value	Temperature sensor x = enabled	2 Byte	CR-T-	[9.001] temperature (°C)	877, 884, 891, 898, 905, 912, 919, 926
Temperature threshold 1 - switch	Temperature sensor x = enabled, Threshold 1 = below or above	1 bit	CR-T-	[1.001] switch	878, 885, 892, 899, 906, 913, 920, 927
Temperature threshold 2 - switch	Temperature sensor x = enabled, Threshold 2 = below or above	1 bit	CR-T-	[1.001] switch	879, 886, 893, 900, 907, 914, 921, 928
Temperature threshold 1 - Lock	Temperature sensor x = enabled, Threshold 1 = below or above Activate threshold block = yes	1 bit	C-W--	[1.001] switch	880, 887, 894, 901, 908, 915, 922, 929
Temperature threshold 2 – Lock	Temperature sensor x = enabled, Threshold 2 = below or above Activate threshold block = yes	1 bit	C-W--	[1.001] switch	881, 888, 895, 902, 909, 916, 923, 930
Temperature threshold 1 – Value (from bus)	Temperature sensor x = enabled, Threshold 1 = below or above Update threshold value from bus = yes	2 Byte	C-W--	[9.001] temperatures (°C)	882, 889, 896, 903, 910, 917, 924, 931
Temperature threshold 2 – Value (from bus)	Temperature sensor x = enabled, Threshold 2 = below or above Update threshold value from bus = yes	2 Byte	C-W--	[9.001] temperatures (°C)	883, 890, 897, 904, 911, 918, 925, 932
Thermostat - Room temperature (from bus)	Room temperature sensor = enabled	2 Byte	C-WTU	[9.001] temperatures (°C)	933, 1008, 1083, 1158, 1233, 1308, 1383, 1458.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Relative humidity (2 bytes, from bus)	Relative humidity sensor = enabled, Communication object size humidity = 2 bytes	2 Byte	C-WTU	[9.007] humidity (%)	934, 1009, 1084, 1159, 1234, 1309, 1384, 1459.
Thermostat - Relative humidity (1 byte, from bus)	Relative humidity sensor = enabled, Communication object size humidity = 1 byte	1 Byte	C-WTU	[5.001] percentage (0..100%)	935, 1010, 1085, 1160, 1235, 1310, 1385, 1460.
Thermostat - Antistratification temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperatures (°C)	936, 1011, 1086, 1161, 1236, 1311, 1386, 1461.
Thermostat - Outdoor temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperature °C	937, 1012, 1087, 1162, 1237, 1312, 1387, 1462.
Thermostat – Coil temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperatures (°C)	938, 1013, 1088, 1163, 1238, 1313, 1388, 1463.
Thermostat - Floor temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperatures (°C)	939, 1014, 1089, 1164, 1239, 1314, 1389, 1464.
Thermostat - Flow temperature (from bus)	enabled	2 Byte	C-WTU	[9.001] temperatures (°C)	940, 1015, 1090, 1165, 1240, 1315, 1390, 1465.
Thermostat - Window contact sensor 1 (from bus)	enabled	1 bit	C-WTU	[1.019] window/door	941, 1016, 1091, 1166, 1241, 1316, 1391, 1466.
Thermostat - Window contact sensor 2 (from bus)	enabled	1 bit	C-WTU	[1.019] window/door	942, 1017, 1092, 1167, 1242, 1317, 1392, 1467.
Thermostat - Presence sensor 1 (from bus)	enabled	1 bit	C-WTU	[1.018] occupancy	943, 1018, 1093, 1168, 1243, 1318, 1393, 1468.
Thermostat - Presence sensor 2 (from bus)	enabled	1 bit	C-WTU	[1.018] occupancy	944, 1019, 1094, 1169, 1244, 1319, 1394, 1469.
Thermostat - Card holder contact (from bus)	enabled	1 bit	C-WTU	[1.001] switch	945, 1020, 1095, 1170, 1245, 1320, 1395, 1470.
Thermostat - Anticondensation (from bus)	enabled	1 bit	C-WTU	[1.001] switch	946, 1021, 1096, 1171, 1246, 1321, 1396, 1471.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Weighted temperature	Room temperature (from bus) = enabled	2 Byte	CR-T--	[9.001] temperatures °C	947, 1022, 1097, 1172, 1247, 1322, 1397, 1472.
Thermostat - Heating/cooling status out		1 bit	CR-T--	[1.100] heating/cooling	948, 1023, 1098, 1173, 1248, 1323, 1398, 1473.
Thermostat - Heating/cooling status in	Thermostat function = both heating and cooling; Heating/cooling changeover = from bus	1 bit	C-W--	[1,100] heating/cooling	949, 1024, 1099, 1174, 1249, 1324, 1399, 1474.
Thermostat - HVAC mode in	Setpoint type = absolute or relative	1 Byte	C-W--	[20.102] HVAC mode	950, 1025, 1100, 1175, 1250, 1325, 1400, 1475.
Thermostat – HVAC forced mode in	Setpoint type = absolute or relative Thermostat function = both heating and cooling	1 Byte	C-W--	[20.102] HVAC mode	951, 1026, 1101, 1176, 1251, 1326, 1401, 1476.
Thermostat - HVAC Mode out	Setpoint type = absolute or relative	1 Byte	CR-T-	[20.102] HVAC mode	952, 1027, 1102, 1177, 1252, 1327, 1402, 1477.
Thermostat - HVAC manual mode	Setpoint type = absolute or relative	1 Byte	C-WTU	[20.102] HVAC mode	953, 1028, 1103, 1178, 1253, 1328, 1403, 1478.
Thermostat - Chrono active status	Setpoint type = absolute or relative	1 bit	CRWTU	[1.011] state	954, 1029, 1104, 1179, 1254, 1329, 1404, 1479.
Thermostat - Actual setpoint		2 Byte	CR-T-	[9.001] temperatures (°C)	955, 1030, 1105, 1180, 1255, 1330, 1405, 1480.
Thermostat - Manual setpoint	Setpoint type = absolute or relative	2 Byte	C-W--	[9.001] temperatures (°C)	956, 1031, 1106, 1181, 1256, 1331, 1406, 1481.
Thermostat - Input setpoint	Setpoint type = single	2 Byte	CRWTU	[9.001] temperature (°C)	957, 1032, 1107, 1182, 1257, 1332, 1407, 1482.
Thermostat - Comfort setpoint (heating)	Setpoint type = absolute or relative	2 Byte	CRWTU	[9.001] temperatures (°C)	957, 1032, 1107, 1182, 1257, 1332, 1407, 1482.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Comfort setpoint (cooling)	Setpoint type = absolute or relative	2 Byte	CRWTU	[9.001] temperature (°C)	959, 1034, 1109, 1184, 1259, 1334, 1409, 1484.
Thermostat - Standby setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU	[9.001] temperatures (°C)	960, 1035, 1110, 1185, 1260, 1335, 1410, 1485.
Thermostat - Standby offset (heating)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	960, 1035, 1110, 1185, 1260, 1335, 1410, 1485.
Thermostat - Standby setpoint (cooling)	Setpoint type = absolute	2 Byte	CRWTU	[9.001] temperature (°C)	962, 1037, 1112, 1187, 1262, 1337, 1412, 1487.
Thermostat - Standby offset (cooling)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	962, 1037, 1112, 1187, 1262, 1337, 1412, 1487.
Thermostat - Economy setpoint (heating)	Setpoint type = absolute	2 Byte	CRWTU-	[9.001] temperature (°C)	964, 1039, 1114, 1189, 1264, 1339, 1414, 1489.
Thermostat - Economy offset (heating)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	964, 1039, 1114, 1189, 1264, 1339, 1414, 1489.
Thermostat - Economy setpoint (cooling)	Setpoint type = absolute	2 Byte	CRWTU	[9.001] temperature (°C)	966, 1041, 1116, 1191, 1266, 1341, 1416, 1491.
Thermostat - Economy offset (cooling)	Setpoint type = relative	2 Byte	CRWTU	[9.002] temperature difference (K)	966, 1041, 1116, 1191, 1266, 1341, 1416, 1491.
Thermostat - Building protection setpoint (heating)		2 Byte	CRWTU-	[9.001] temperatures (°C)	968, 1043, 1118, 1193, 1268, 1343, 1418, 1493.
Thermostat - Building protection setpoint (cooling)		2 Byte	CRWTU	[9.001] temperatures (°C)	969, 1044, 1119, 1194, 1269, 1344, 1419, 1494.
Thermostat - Room temperature control status		1 bit	CR-T-	[1.003] enable	970, 1045, 1120, 1195, 1270, 1345, 1420, 1495.
Thermostat - Heating out command	Control type = 2-point hysteresis or PWM, Command communication object = separated	1 bit	CR-T-	[1.001] switch	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.
Thermostat - Heating out command	Control type = continuous, Command communication object = separated	1 Byte	CR-T-	[5.001] percentage (0..100%)	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Heating and cooling out command	Control type = 2-point hysteresis or PWM, Command communication object = unique	1 bit	CR-T-	[1.001] switch	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.
Thermostat - Heating and cooling out command	Control type = continuous, Command communication object = unique	1 Byte	CR-T-	[5.001] percentage (0..100%)	973, 1048, 1123, 1198, 1273, 1348, 1423, 1498.
Thermostat - Cooling out command	Communication object control = separate Control type = 2-point hysteresis or PWM	1 bit	CR-T-	[1.001] switch	976, 1051, 1126, 1201, 1276, 1351, 1426, 1501.
Thermostat - Cooling out command	Communication object control = separate Control type = continuous	1 Byte	CR-T-	[5.001] percentage (0..100%)	976, 1051, 1126, 1201, 1276, 1351, 1426, 1501.
Thermostat - Auxiliary heating output command	Auxiliary heating = enabled, Command communication object = separated	1 bit	CR-T-	[1.001] switch	977, 1052, 1127, 1202, 1277, 1352, 1427, 1502.
Thermostat - Auxiliary heating and cooling output command	Auxiliary heating = enabled (Auxiliary heating) Communication object = unique	1 bit	CR-T-	[1.001] switch	977, 1052, 1127, 1202, 1277, 1352, 1427, 1502.
Thermostat - Auxiliary cooling output command	Auxiliary cooling = enabled	1 bit	CR-T-	[1.001] switch	979, 1054, 1129, 1204, 1279, 1354, 1429, 1504.
Thermostat - Auxiliary heating disable	Auxiliary heating = enabled, Disabled from bus = yes	1 bit	C-W--	[1.003] enable	980, 1055, 1130, 1205, 1280, 1355, 1430, 1505.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Auxiliary cooling disable	Auxiliary cooling = enabled, Disabled by bus = yes	1 bit	C-W--	[1.003] enable	981, 1056, 1131, 1206, 1281, 1356, 1431, 1506.
Thermostat - Fan continuous speed	Control type = continuous regulation	1 Byte	CR-T-	[5.001] percentage (0..100%)	982, 1057, 1132, 1207, 1282, 1357 1432, 1507
Thermostat - Fan speed 1	Control type ≥ 1 speed	1 bit	CR-T-	[1.001] switch	983, 1058, 1133, 1208, 1283, 1358, 1433, 1508.
Thermostat - Fan speed 2	Control type ≥ 2 speeds	1 bit	CR-T-	[1.001] switch	984, 1059, 1134, 1209, 1284, 1359 1434, 1509.
Thermostat - Fan speed 3	Control type = 3 speeds	1 bit	CR-T-	[1.001] switch	985, 1060, 1135, 1210, 1285, 1360 1435, 1510.
Thermostat - Fan control disable	Disable ventilation from bus = yes	1 bit	C-W--	[1.002] boolean	986, 1061, 1136, 1211, 1286, 1361, 1436, 1511.
Thermostat - Alarm text		14 Byte	CR-T-	[16,000] Character string (ASCII)	987, 1062, 1137, 1212, 1287, 1362, 1437, 1512.
Thermostat - Manual setpoint active status	Setpoint type ≠ single	1 bit	CRWTU	[1.011] state	988, 1063, 1138, 1213, 1288, 1363, 1438, 1513.
Thermostat – Fan manual speed		1 byte	CRWTU	[5,010] counter pulses (0…255)	989, 1064, 1139, 1214, 1289, 1364, 1439, 1514
Thermostat - Fan speed		1 Byte	CR-T-	[5.010] counter pulses (0…255)	990, 1065, 1140, 1215, 1290, 1365, 1440, 1515.
Thermostat – Fan manual active status		1 bit	CRWTU	[1.011] state	991, 1066, 1141, 1216, 1291, 1366, 1441, 1516.
Thermostat - Room temperature control alarm		1 bit	CR-T-	[1.005] alarm	992, 1068, 1142, 1217, 1292, 1367, 1442, 1517.
Thermostat - Room temperature control alarm		1 bit	CR-T-	[1.005] alarm	992, 1068, 1142, 1217, 1292, 1367, 1442, 1517.
Thermostat – disable room temperature controller (from bus)	Disable temperature controller from bus = yes	1 bit	C-W--	[1.001] switch	993, 1068, 1143, 1218, 1293, 1368, 1443, 1518.
Thermostat – Dew-point temperature	Dew-point temperature = enabled	2 Bytes	CR-T-	[9.001] temperature °C	994, 1069, 1144, 1219, 1294, 1369, 1444, 1519.

Object name	Conditions	Size	Flags	DPT	CO number(s)
Thermostat - Relative humidity setpoint for dehumidification	Dehumidification function ≠ disabled	2 Bytes	CRWTU	[9.007] humidity (%)	995, 1070, 1145, 1220, 1295, 1370, 1445, 1520.
Thermostat - Relative humidity setpoint for humidification	Humidification ≠ disabled	2 Bytes	CRWTU	[9.007] humidity (%)	996, 1071, 1146, 1221, 1296, 1371, 1446, 1521.
Thermostat - Dehumidification command	Dehumidification function ≠ disabled	1 bit	CR-T-	[1.001] switch	997, 1072, 1147, 1222, 1297, 1372, 1447, 1522.
Thermostat - Dehumidification command - Water battery	⇒ temperature control⇒ settings Thermostat function = both heating and cooling, ⇒ temperature control⇒ cooling type = floor radiant panel or ceiling radiant panel, Relative humidity control⇒ dehumidification⇒ Dehumidification function = cooling only	1 bit	CR-T-	[1.001] switch	998, 1073, 1148, 1223, 1298, 1373, 1448, 1523.
Thermostat - Dehumidification integration control	⇒ temperature control⇒ settings Thermostat function = heating and cooling, Temperature control⇒ Cooling⇒ Cooling type = floor radiant or ceiling radiant panel Relative humidity control⇒ dehumidification⇒ Dehumidification function = cooling only Integration = yes	1 bit	CR-T--	[1.001] switch	999, 1074, 1149, 1224, 1299, 1374, 1449, 1524.
Thermostat - Dehumidification control disable	Disable dehumidification control from the bus = yes	1 bit	C-W--	[1.002] boolean	1000, 1075, 1150, 1225, 1300, 1375, 1450, 1525.
Thermostat - Humidification command	Humidification ≠ disabled	1 bit	CR-T-	[1.001] switch	1001, 1076, 1151, 1226, 1301, 1376, 1451, 1526.
Thermostat - Humidification control disable	Disable humidification control from bus = yes	1 bit	C-W--	[1.002] boolean	1002, 1077, 1152, 1227, 1302, 1377, 1452, 1527

Object name	*Conditions*	*Size*	*Flags*	*DPT*	*CO number(s)*
Thermostat - Anticondensation alarm	Active anticondensation or Protection with anticondensation probe = enabled	1 bit	CR-T-	[1.005] alarm	1003, 1078, 1153, 1228, 1303, 1378, 1453, 1528.
Thermostat – Thermal generator lock alarm		1 bit	C-W--	[1.005] alarm	1004, 1079, 1154, 1229, 1304, 1379, 1454, 1529.
Thermostat - Building protection HVAC mode active		1 bit	CR-T-	[1.011] state	1005, 1080, 1155, 1230, 1305, 1380, 1455, 1530.
Thermostat - Fan manual speed percentage		1 Byte	CR-T-	[5.001] percentage	1006, 1081, 1156, 1231, 1306, 1381, 1456, 1531.
Thermostat – Fan manual speed off status		1 bit	CR-T-	[1.011] state	1007, 1082, 1157, 1232, 1307, 1382, 1457, 1532.

10 Warning

- Installation, electrical connection, configuration and commissioning of the device may only be carried out by qualified personnel.
- Opening the device container causes the immediate expiration of the warranty.

10.1 Return of defective devices

Devices that have problems or defects can be returned for repair or replacement by following the procedure described below.

10.1.1 Devices purchased directly from ekinex®

It is necessary to first contact ekinex® technical support by sending an e-mail to support@ekinex.com with the following information (mandatory):

- Exact model of the device;
- Serial number (found on the label applied to the product);
- Date and/or references of the purchase document;
- Precise description, and as detailed as possible, of the fault or problem.

ekinex® technical support will promptly contact the customer, depending on the case, to investigate the problem, suggest possible solutions or authorize the return of the device for repair or replacement.

The devices must be shipped to the following address:

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

Any further instructions will be agreed with the technical support during the support phase.

10.1.2 Devices purchased through resellers

For assistance relating to devices purchased through resellers, it is necessary to contact the technical support structure of the latter.

Depending on the type of problem and any other factors, at the sole decision of ekinex® and in agreement with the dealer, the customer may be advised to contact ekinex® directly according to the same procedure as above.

10.2 Other information

This application manual is intended for installers, system integrators and system configurators.

For further information on the product, please contact the ekinex® technical support service at support@ekinex.com or visit the website www.ekinex.com.