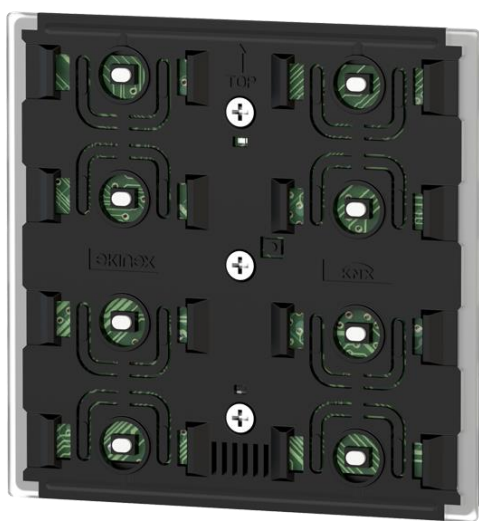# EKINEX
CONTROL YOUR LIVING SPACE

**Application manual
Wall mounting KNX Pushbuttons
BASIC WL (Without Led)**

**EK-E32-TP   2-4 rockers Series 'FF
EK-E23-TP   1-4 rockers Series '71**

Summary

Revision	Modifications	Author	Date
1.0.0	Emission	R. Rocco	23/05/2017
2.3	Added note for plastic screws mounting (page 6)	R. Rocco	04/05/2022
3.0	Updates from E22 to E32	G. Schiochet	27/05/2022

1 Scope of the document

This application manual describes application details for ekinex® pushbutton interface EK-E32-TP (2-4 rockers) and for ekinex® pushbutton interface EK-E23-TP (1-4 rockers).

The document is aimed at the system configurator as a description and reference of device features and application programming. For installation, mechanical and electrical details of the device please refer to the technical description datasheet.

Application manual and application programs for ETS are available for download at www.ekinex.com.

Item	File name (## = release)	Version	Device rel.	Update
Product datasheet	STEKED2E32TP_EN.pdf	EK-E32-TP		
Application manual	MAEKE32E23TP_EN.pdf	EK-E32-TP	A1.0	05 / 2022
Application program	APEKE32TP##.knxprod	EK- E32-TP		

Item	File name (## = release)	Version	Device rel.	Update
Product datasheet	STEKE13E23TP_IT.pdf	EK-E23-TP		
Application manual	MAEKE32E23TP_IT.pdf	EK-E23-TP	A1.0	05 / 2022
Application program	APEKE23TP##.knxprod	EK-E23-TP		

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

2-4 rocker interface EK-E32-TP:

1-4 rocker interface EK-E23-TP:

2 Product description

The ekinex® KNX 4-rocker pushbutton unit is a wall-mounting device for on/off switching of loads, dimming of lighting devices, control of motor drives or other programmable switching and control functions.

This unit is equipped with an integrated KNX bus communication module and is designed for wall installation; commands are constituted by rockers with 2 active plus a neutral rest position.

For final use, this unit must be completed with frontal plates for commands and a frame, which must be ordered separately in order to obtain the desired aesthetic look; regardless of the detail, several kinds of plates are available (square or rectangular) which can be combined in order to obtain different rockers' combinations.

The device is powered by the KNX bus line with a 30 VDC SELV voltage and does not require auxiliary power.

Product code	Number and type of rockers	Rocker size	Frame
EK-E32-TP	2 rectangular vertical 4 square 4 rectangular horizontal	40 x 80 mm 40 x 80 mm 80 x 20 mm	*Form* or *Flank* series, or 'NF (No Frame)

Codice prodotto	Numero e tipo di tasti	Dimensione dei tasti	Cornice
EK-E23-TP	1 single square 2 rectangular vertical 4 square 4 rectangular horizontal	60x60 mm 30x60 mm 30x30 mm 15x60 mm	*Form* or *Flank* series, or 'NF (No Frame)

The supply includes, inside the box:

- 2 pairs of fixing screws;
- a KNX terminal block for the connection of the bus line.

i	**Mounting information**
	1. The screws supplied in the package are suitable for standard installations. For particular applications, in which the screws must be replaced, these must be of the flat-head type.
	2. The supplied plastic screws (# 2) must only be used to hold the push-button panel in position, therefore they must not be tightened with excessive force (max. torque 0.4 Nm).
	3. The screws for the metal support must be tightened with a max. torque of 1.0 Nm.

For full installation and operation, the unit must be completed with:

- A rocker faceplate (according to the chosen number and disposition);
- A plastic adapter;
- A metal support;
- An ekinex® *Form* or *Flank* series 1-place square or 2-place rectangular frame (with the exception of the rocker interfaces of No Frame series NF);
- An ekinex® 1-window square or 2-window rectangular plate

Rocker placement for EK-E32-TP

Combining the 3 available models of rockers (rectangular vertical, square horizontal and rectangular horizontal) different configurations are allowed, as shown in the following picture.
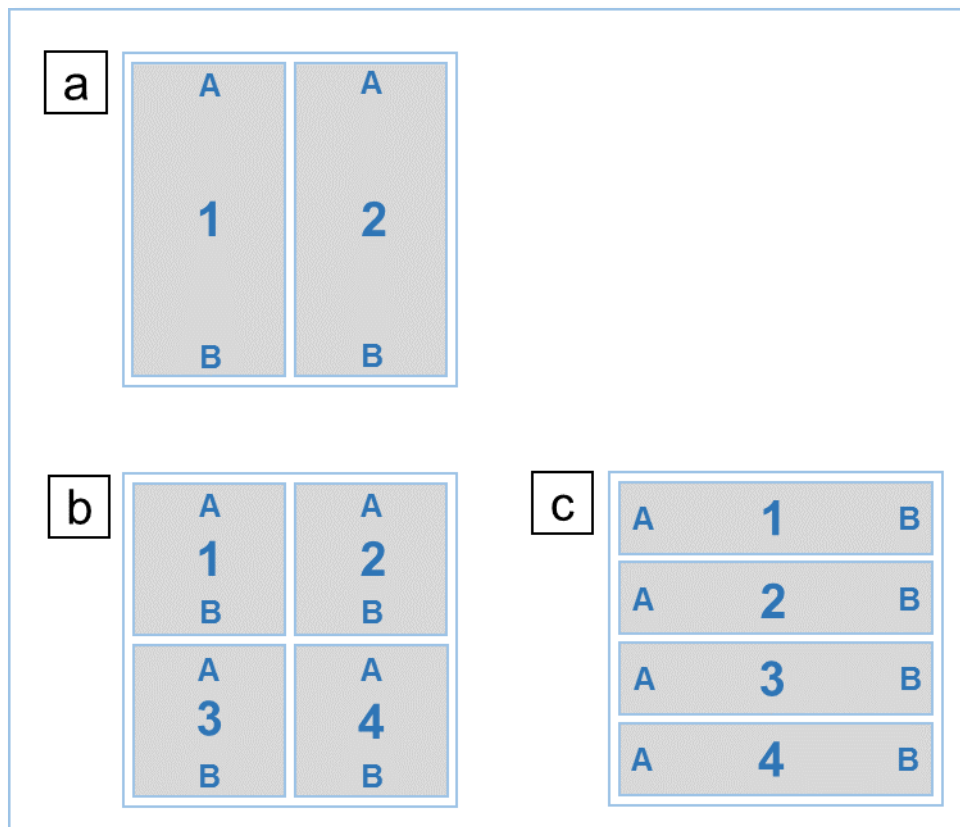

Figure 1A - Rocker combination for EK-E32-TP

Disposizione dei tasti EK-E23-TP

Combining the 4 available models of rockers (square single, rectangular vertical, square horizontal and rectangular horizontal) different configurations are allowed, as shown in the following picture.
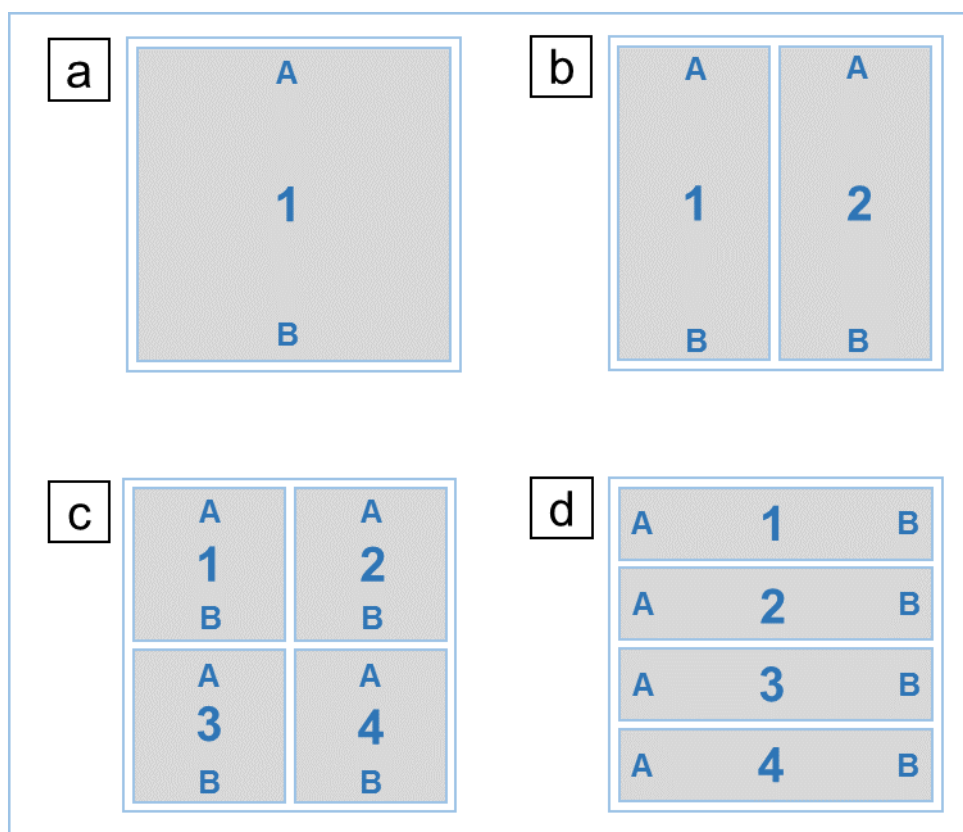

Figure 2B - Rocker combination for EK-E23-TP

2.1 Rocker functions

Each one of two active positions of the rockers (side for rectangular rockers, superior and inferior for square ones) corresponds to an *action*, i.e. an input, or physical pushbutton, of the device. Such actions, in relation to a single rocker, will be labelled with letters A and B.

When one side of a rocker is pressed, the device sends on the KNX bus the telegram (or sequence) associated to the corresponding function according to how the device is programmed.

In the most common situation, for instance, one side of the rocker might send an "ON" telegram for a lighting unit, while the other side would send the "OFF" telegram for the same unit. Another typical application would be for one side of the rocker to increase the brightness of a dimmed light (and respectively decrease it for the opposite side), or to raise / lower a curtain or blind and so on.

The two functions associated with a rocker can also be programmed to perform exactly the same operation, thereby effectively causing one rocker to act as a single pushbutton.

Note for EK-E32-TP

> **i** The use of the entire activation surface of the rocker as if it is a single rocker is programmed by defining the function B as "in parallel with function A as a single function". This use is configurable only in combinations a), b) and c) (see Figure 1 in the previous page): for instance, it is possible to link only one function to combination a) with a single square rocker.
>
> In combination d) with 4 rectangular horizontal rockers it is possible to link only functions other than function A and function B.

2.2 Customization of rocker plates

Rocker plates can be customized with predefined symbols and texts. On request, a customization is also possible with symbols and texts chosen by the customer. For more information see the standard library on the ekinex® catalogue or the website www.ekinex.com.

> **i** For further technical information, please also refer to the product datasheet available on the website www.ekinex.com.

3   Switching, display and connection elements

The front side of the device is fitted with mounting hooks for the rocker faceplates; between the hooks there are the pushbuttons.

On the rear side, the device is equipped with a programming pushbutton, a programming status LED and terminals for connecting the KNX bus line.
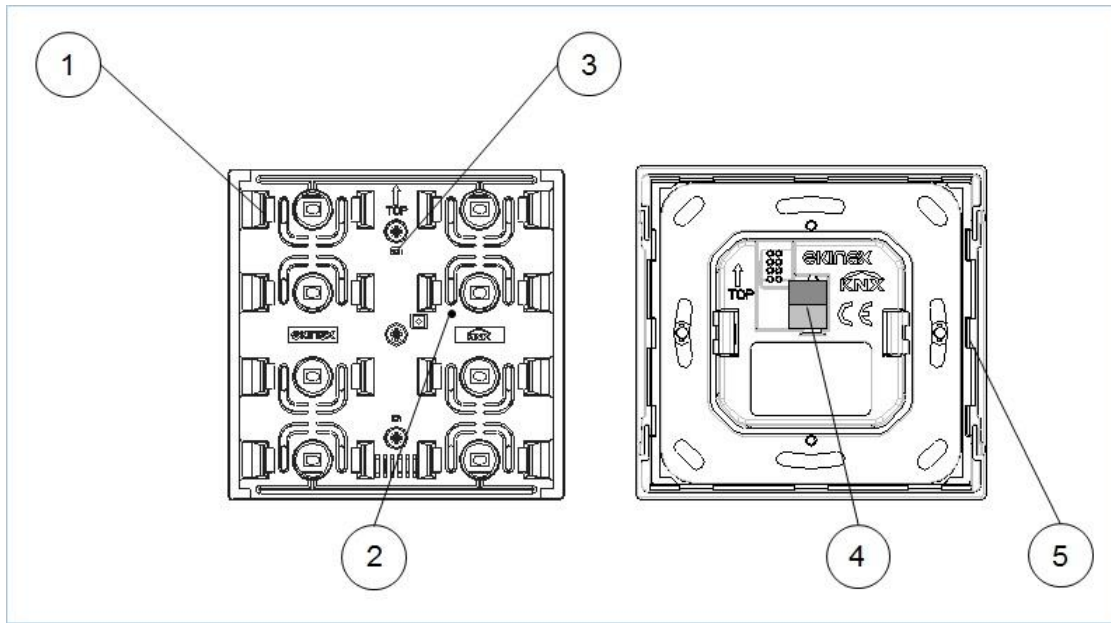
Fig. 2A - Switching, display and connection elements for EK-E32-TP

1.   Rocker faceplate hooks

2.   Programming pushbutton

3.   Programming LED

4.   Terminal block for KNX bus line
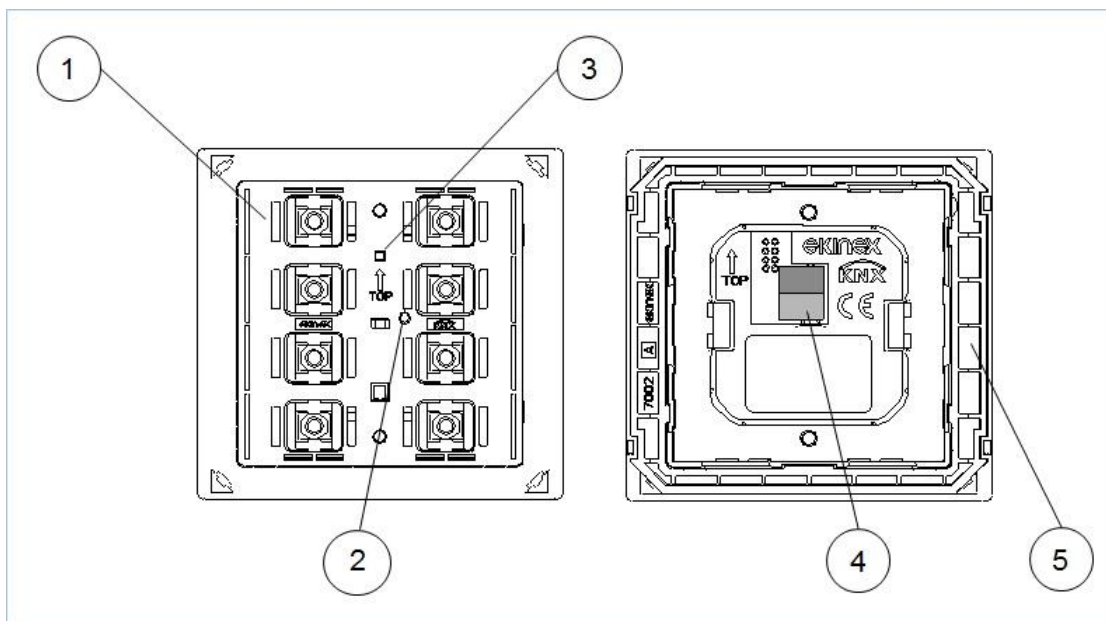
5.   Adapter

Fig. 2B- Switching, display and connection elements for EK-E23-TP

1. Rocker faceplate hooks
2. Programming pushbutton
3. Programming LED
4. Terminal block for KNX bus line
5. Adapter

4 Configuration

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

In order to configure and commission the device you need ETS4 or later releases and the ekinex® application program, (namely **APEKE32TPxx.knxprod for EK-E32-TP** and **APEKE23TPxx.knxprod for EK-E23-TP**); which can be downloaded from the ekinex website www.ekinex.com.

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

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

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

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

Product code	EAN	No. of channels	ETS application software (## = release)	Communication objects (max no.)	Group addresses (max no.)
EK-E32-TP		8	APEKE32TP##.knxprod	229	254

Product code	EAN	No. of channels	ETS application software (## = release)	Communication objects (max no.)	Group addresses (max no.)
EK-E23-TP		8	APEKE23TP##.knxprod	229	254

> **i** Configuration and commissioning of KNX devices require specialized skills; to acquire these skills, you should attend training courses at a training centre certified by KNX.
>
> For further information: www.knx.org.

5   Commissioning

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

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

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

6   Function description

After switching on the bus, which also acts as a power supply, the device becomes fully functional after a very short time needed for reinitialization. A delay is programmable for the device to become active on the bus in order to avoid a bus traffic overload during the first moments of start-up of the whole network.

In case of a bus power failure (voltage lower than 19 V for 1 s or more), the device becomes unreactive: before the power supply becomes insufficient, the status is internally stored. The timing functions are not active, neither are the programmed group addresses.

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

6.1   Offline operation

A fully unprogrammed device does not operate in standby mode. Since the operation relies entirely on the exchange of information through communication objects, there is no part of the device that can operate independently from a KNX bus.

6.2   Online operation

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

6.3   Software working cycle

The main purpose of the software is following:

- Handle user pushbutton presses and generate bus telegrams according to the assigned functions;
- Implement pushbutton interlock and timing functions;
- reagire ai telegrammi sul bus di richiesta dello stato degli ingressi o delle variabili locali.
- Respond to bus messages requesting feedback on the status of the inputs.

The status of the device and specifically of its entities (input activation status) relies on KNX *communication objects*, which can be freely defined and bound in various ways to the physical elements of the device; these communication objects acts as *state variables* for the device.

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

6.4   Pushbutton inputs

The press of a pushbutton can be bound to different effects on a state variable.

6.4.1   Pushbutton input events

A button press can be handled either as an "on-off" event ("on" means when the button is pushed, "off" when it is released), or as a "short press - long press" event (whereby a time period can be defined to discriminate the duration of the "long" from the "short" press).

In both cases, for each of the two available events a separate action can be assigned that operates on a selected variable (actually, more than one; see below for details).

6.4.2   Lock function

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

When in a locked state, the input is effectively disabled.

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

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

6.4.3   State variables (communication objects)

The variable that is changed by the input events can be one of the types available for KNX communication objects, i.e. for instance a 1-bit value (on-off), a 2-bit value or an integer value of larger size.

In all cases, each of the two events can:

- change the value of the variable to one of two definable values within its range (which is trivial in the case of the 1-bit value);
- toggle between the two defined values
- do nothing (value is unaffected)

This state variable, once assigned a group address, is actually a **KNX communication object**; as such, it undergoes the usual rules for communication objects, among which – for instance – the effect of flags to determine how the change of value affects the transmission of the objects.

6.4.4   Binding between Events and Communication objects

The above description is a little simplified in order to ease comprehension; as a matter of fact, to each event can be assigned not just one, but several communication objects (up to 8), even of different types. Each of these communication objects can have its own behaviour and its own associated value set.

6.4.5   Repeated send

For most features, is it possible to set the device to send a telegram not just when a value changes as a consequence of an input transition, but also at regular intervals whenever that value setting is active.

This behaviour, also referred to as Cyclical Transmission, can be set separately for each of the two values that are associated to an input (or both, or none of them).

If an input is set to "*send values or sequences*" mode, repeated send is not available if more than 1 Communication Object is assigned to that input.

6.4.6   Accoppiamento ingressi

The 8 pushbutton inputs described can be considered, and used, as independent; however, due to the physical structure of the device and the nature of the functions it most frequently performs, these inputs can be naturally grouped in pairs, which in the application program are referred to as *channels*. Each channel is made of a pair of inputs, and is physically associated to a rocker.

Since the channels of the device are labelled 1 to 4, the inputs are labelled 1A / 1B for channel 1, 2A / 2B for channel 2 and so on. The same numbering is used whether the channel pairing is used or not.

In order to specify channel pairings, each rocker can be configured in two ways: single mode and coupled mode. This setting appears among rocker-level settings rather than input-level settings, because only inputs belonging to the same rocker can be coupled. The only combinations allowed for coupling are in fact 1A with 1B, 2A with 2B, and so on.

- In *single or independent 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 a channel in order to perform a common functionality; therefore, they operate on shared communication objects.

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

It must be mentioned that there is actually a third way to configure an input pair, which lies somehow halfway between the two modes above (although it is considered as a variation of the single mode): each second input, i.e. inputs 1B, 2B, 3B etc., can be configured to perform exactly the same function as its first input. In this fashion, both pushbuttons associated with a rocker are effectively operated "in parallel", so as to operate the whole rocker as a single, larger control (either pushbutton or switch, according to programmed operation).

Following there is a description of all possible features of the channels. *Single or independent* and *coupled* modes have a similar functionality, but differ for the configuration and will be therefore be treated separately.

6.4.7 Single or independent input mode

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

1. Send values or sequences
An event triggers the transmission on the bus of configurable values or sequence of values.
These values can be of a logical type or a numerical type with a different size.
A sequence of values can be made of up to 8 communication objects of different value types.
Time delays can set between values in the sequence.

2. Dimmer control
This mode is intended to be used with dimming actuators for the control of lighting devices.
The functionality is triggered on short press and long press events.
On short press events, the device sends on/off telegrams to the dimming actuator.
On long press events, the dimming percentage is varied up or down until the button is released.

3. Shutter or Venetian blind control
This mode is intended to be used together with actuators for the control of motorized blinds, shutters and similar devices. These actuators have functions for blind opening and closing; two movement types are selectable, i.e. continuous movement and stepwise movement. On input events, the device sends operation telegrams to the actuators.
The operation is configurable through following parameters:
- If *toggle* mode is enabled, on each activation of the same input the movement direction is inverted; if it is disabled, the movement direction is fixed and it can be set to "up" or "down".
- If *blinds* mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.

4. Scene function output
This mode is intended to be used together with several KNX actuators that support using a scene function; this function allows storing and recalling a communication object value on an actuator.
In this mode, the role of the device is to send a "store / recall scene" telegram to the actuator on a long / short press event.
This mode has two possible configurations:
- Activate pre-set scene on short press, and store current setting as scene value on long press
- Activate two different scenes on long and short press.

6.4.8 Coupled input mode

Each pair of coupled inputs, corresponding to the two sides of a same rocker, can be configured for one of following different features (only the differences from the single mode are highlighted):

1. Switch control
Both inputs in a pair are bound to the same communication object; unlike single mode, the object can only be of the 1-bit type (on-off), therefore building a conventional switching behaviour.
The user can configure which of the two inputs sets the "off" or resp. "on" value.

2. Dimmer control
The functionality is triggered on short press and long press events of the inputs in the pair.
The user can configure which of the two inputs sets the "up" or resp. "down" value.
On short press events, the input configured as "up" sends an "on" switching telegram to the dimming actuator; while the "down" input sends an "off" telegram.

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

3. *Shutter or Venetian blind control*
    The two inputs of a pair are assigned to opposite movement directions; these can be assigned to inputs as desired, i.e. A up / B down or the other way around.
    The *blinds* mode can also be set, and it works exactly as in single mode.

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

6.4.9   Dimming function

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

> **i**
>
> *All the information contained in this section have the purpose of illustrating specific device functions, and therefore are not to be necessarily considered exhaustive or applicable to other cases. In order to obtain a complete or generally applicable documentation, please refer to the official KNX documentation.*
>
> *For further information, visit the website www.knx.org.*

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

Cod. passo	Ampiezza passo
0 0 1	100%
0 1 0	50%
0 1 1	25%
1 0 0	12%
1 0 1	6%
1 1 0	3%
1 1 1	1%
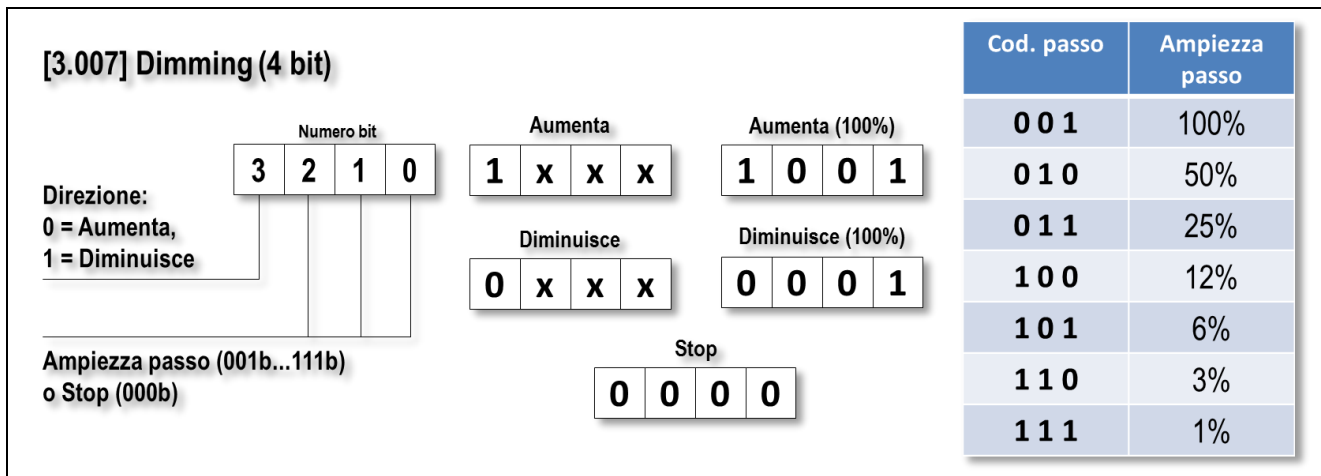
The transmission of telegrams containing data of such format tells the actuator to perform an increase or a decrease, by an amplitude equal to the specified step, or to stop an ongoing variation.

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

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

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

In case of unit EK-Ex2-TP, the defined operations and related commands are the following:
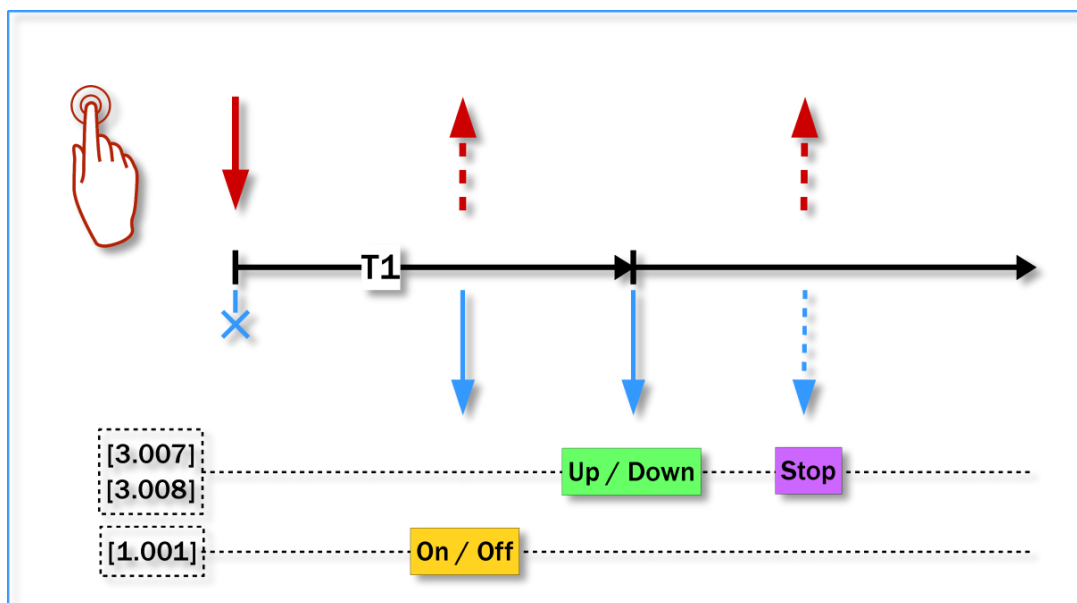
- Short press: instantaneous switch on / off (toggle on / off on a switch object);
- Long press: increase / decrease value until 100% / 0%;
- Release: stop increase / decrease.

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

6.4.10 Shutter / venetian blind function

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) are to be considered.

> **i**
>
> *All the information contained in this section have the purpose of illustrating specific device functions, and therefore are not to be necessarily considered exhaustive or applicable to other cases. In order to obtain a complete or generally applicable documentation, please refer to the official KNX documentation.*
>
> *For further information, visit the website www.knx.org.*

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 rocker unit, in order to generate a sequence of these commands with the proper order and time interval, there are multiple possibilities.

In case of ekinex input devices, two modes are available – indicated as "Shutter" and "Venetian blind" based on their typical destination – which are illustrated in the following figure.
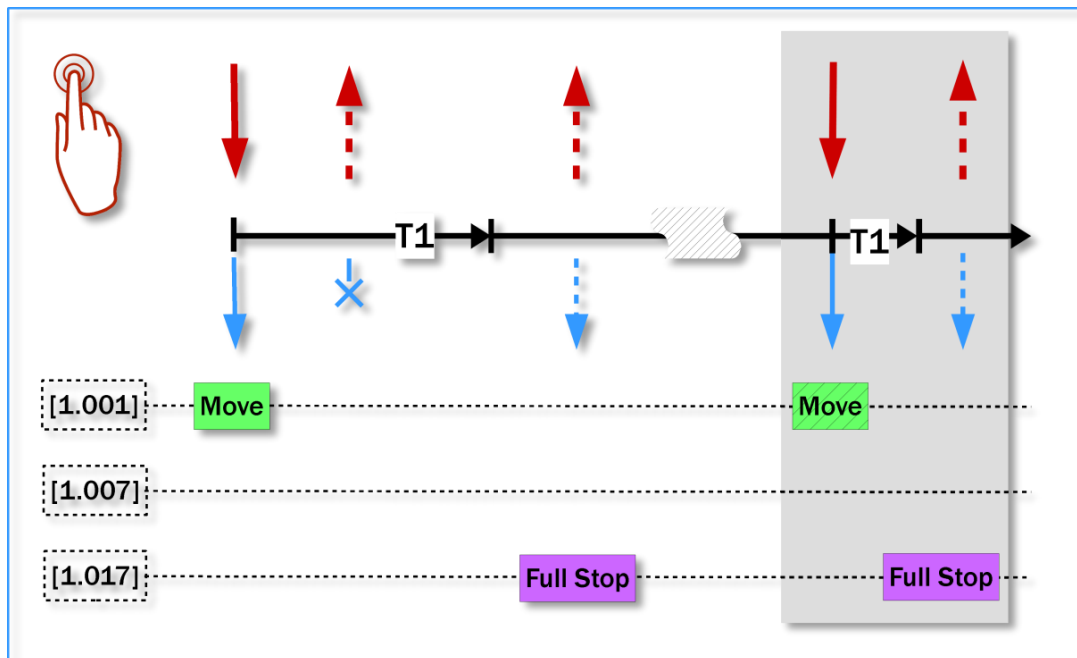
Figure 4 - "Shutter" mode command sequence

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

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

If the rocker is pressed with a long press, when it is released – which will be in correspondence with the desired position – the shutter will stop.
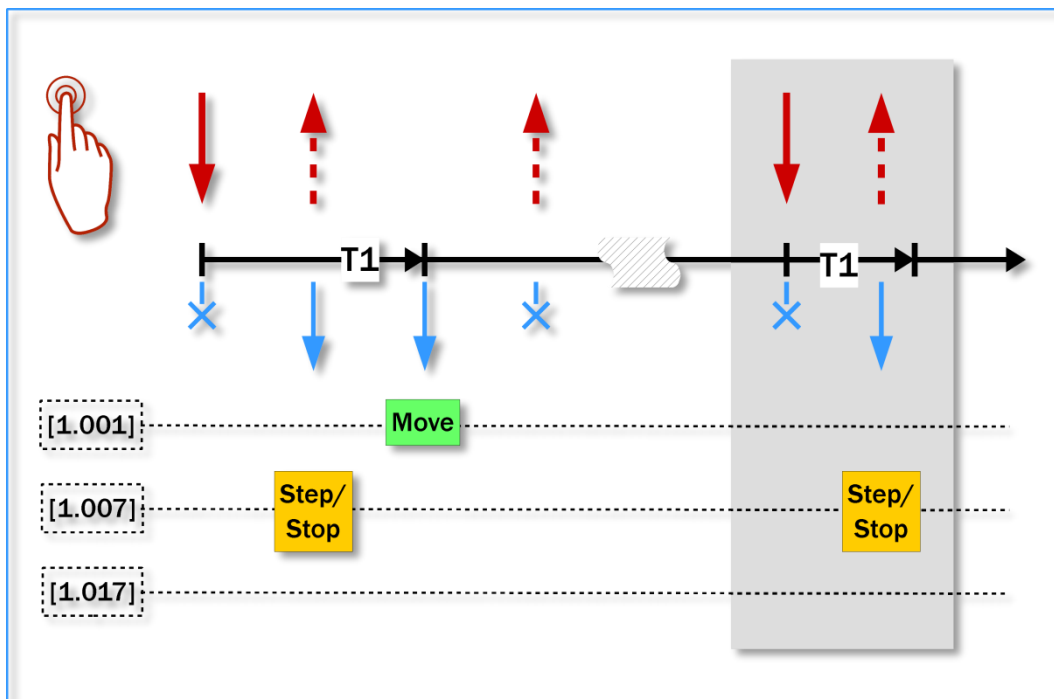
Figure 5 - "Venetian blind" mode command sequence

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

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

6.5   Logic functions

The KNX pushbutton allows to use some useful logic functions (AND, OR, NOT and exclusive OR) in order to implement complex functions in the building automation system.

You can configure:

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

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

The inputs created by the objects are then logically combined as shown in the following figure:

Figure 6 – Logic combination function
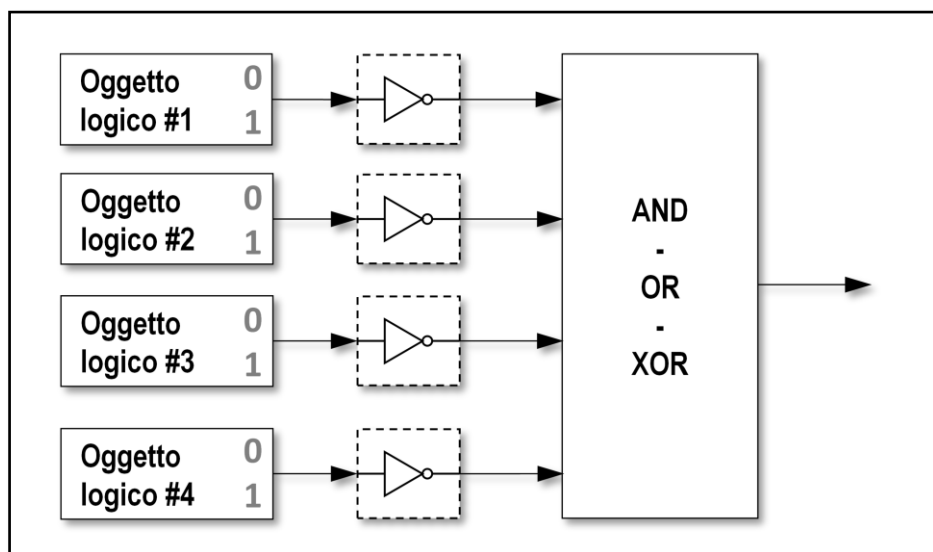
The logic block on the right side of the figure has the following function, based on the selected operation:

- OR – the output is ON if at least one input is ON;
- AND – the output is ON if all inputs are ON;
- XOR – the output is ON if an odd number of inputs is ON;

This last function is more intuitive when there are only 2 inputs: in this case, the output is ON when one input or the other one is ON, but not the two of them simultaneously.
Please note that in this description, with "input" and "output" we refer only to the logic block; for the device operation, the effective "inputs" are given by communication objects, so also the possible activation of NOT logic operators has to be considered.

The following figures show the basic logic functions, assuming 2 inputs and only one logic communication object:
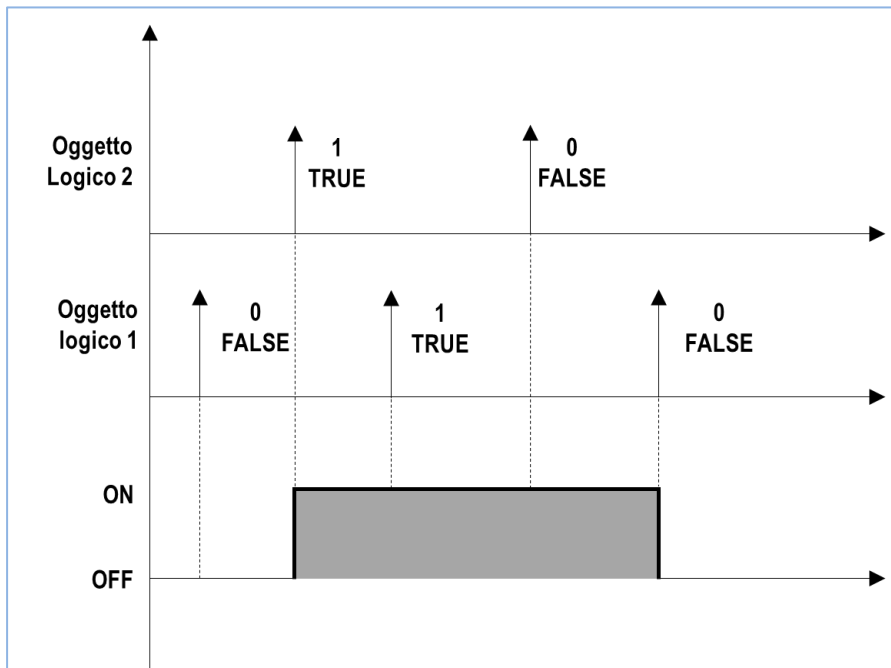
Figure 7 – Logic function OR
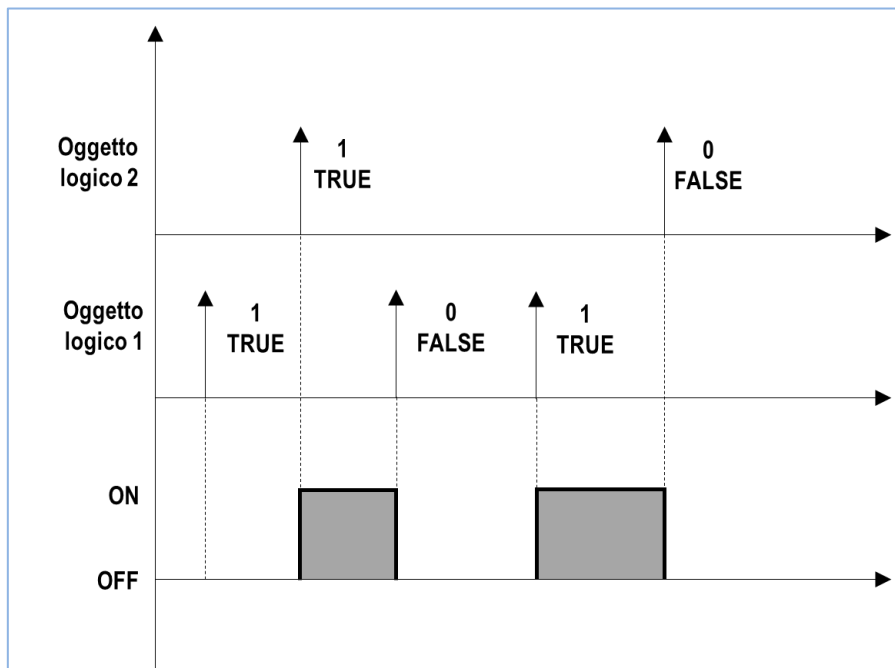
Figure 8 – Logic function AND
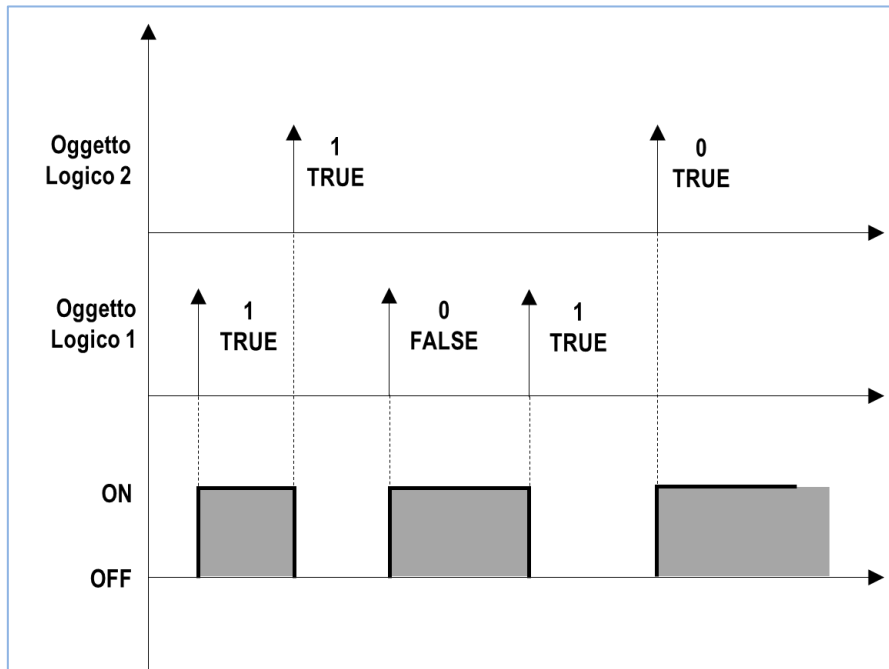
Figure 9 – Logic function Exclusive OR (XOR)

For each channel, a parameter *Delay after bus voltage recovery* is available: this parameter represents the time interval between the bus voltage recovery and the first reading of the input communication objects for evaluating the logic functions.

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

7   Application program for ETS

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

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

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

---

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

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

```
─ Info su EK-EX2-TP

─ Generale

─ Sensori interni

─ Configurazione tasti

─ Tasto 1
      ─ Funzione A
            ─ Oggetto 1
      ─ Funzione B
            ─ Oggetto 1

─ ....

─ Tasto 4

─ Funzioni logiche
```
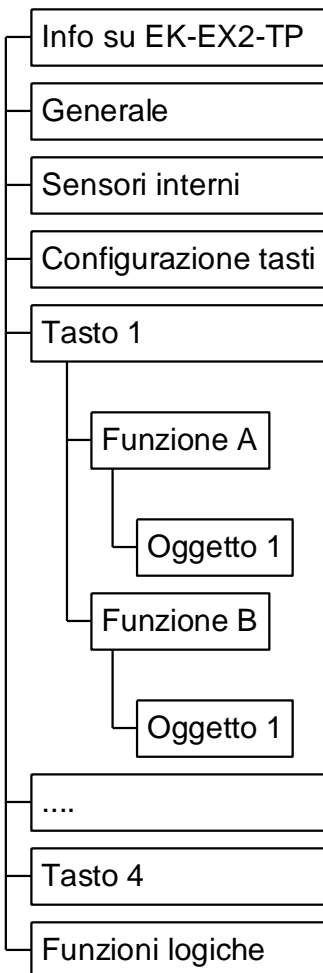
7.1   About EK-E32-TP and EK-E23-TP

The folder *About EK-E32-TP* is for information purposes only and does not contain parameters to be set. The information given is:

© Copyright EKINEX S.p.A. 2017
Application software for ETS4
Version 1.00 (or later)
4 rockers pushbutton WL series 'FF

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

The folder *About EK-E23-TP* is for information purposes only and does not contain parameters to be set. The information given is:

© Copyright EKINEX S.p.A. 2017
Application software for ETS4
Version 1.00 (or later)
4 rockers pushbutton WL series 71

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

7.2   General settings

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

Parameter name	Conditions	Values
Rockers configuration		*See Fig. 9 and 10* *for available options*
	Specifies the configuration of installed rocker plates, thereby determining how physical pushbuttons will be associated to logical inputs and coupled in rocker pairs.	
Delay after bus voltage recovery	-	hh:mm:ss.fff (**00:00:04.000**)
	Delay before bus telegrams can be sent after a recovery of the bus voltage. The delay time affects the transmission generated by an event as well as the cyclical transmission. For the cyclical transmission: after the delay time finished, the cycle restarts and the first telegram will be sent after the cycle time.	
Logic functions		enabled / **disabled**
	Enables the folders to configure AND, OR e XOR logic functions and their relative input and output communication objects.	
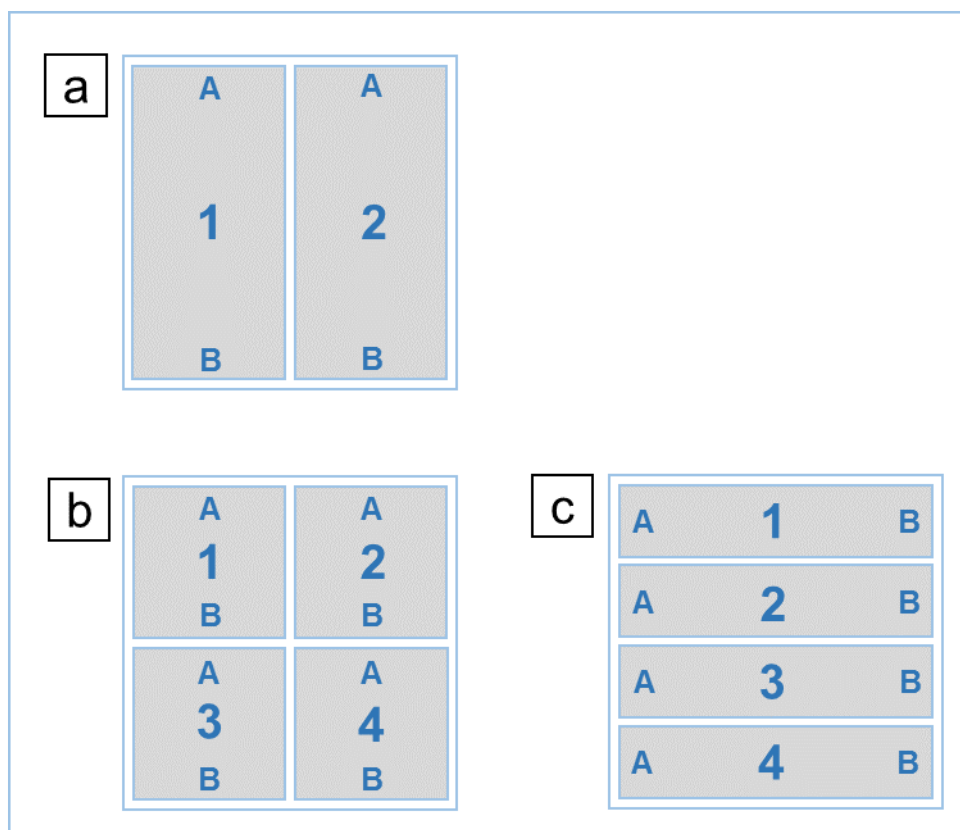
Figure 10A – Rockers' combination version for EK-E32-TP

Available rocker plate configuration options for 2-4 rocker pushbutton:

a.   2 vertical rectangular rockers
b.   4 square rockers
c.   4 rectangular horizontal rockers
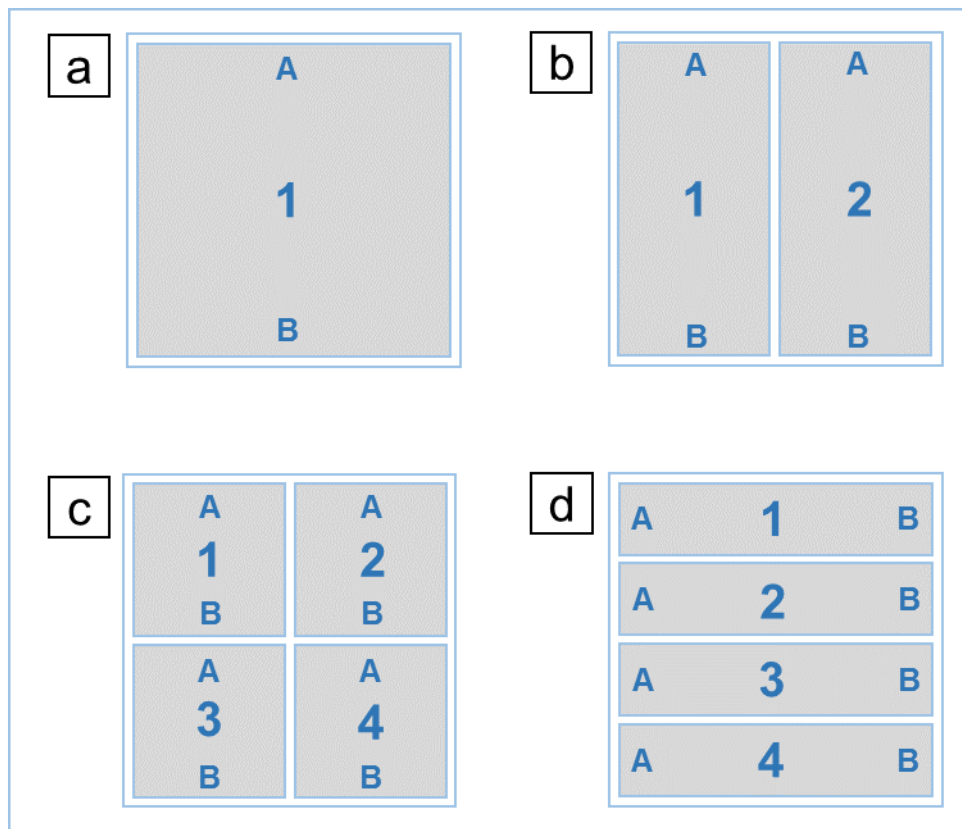
Figure 11B – Rockers' combination version for EK-E23-TP

Available rocker plate configuration options for 1-4 rocker pushbutton:

d.   1 single square rocker*
e.   2 vertical rectangular rockers
f.   4 square rockers
g.   4 rectangular horizontal rockers

7.3 Rockers configuration

Parameter name	Conditions	Values
Rocker x	-	disabled **independent or single** coupled copy parameters from rocker*
	Set operation mode for inputs corresponding to Rocker x. The identification of which Rocker and associated input pushbuttons are corresponding to a given number (e.g. 1A – 2B – 4A etc.) is done according to parameter "General / Rockers configuration". * This option is only available for rockers no. 2 and above. If selected, the corresponding rocker can be made to perform the exact same kind of function as another specified rocker, but basing on different communication objects. This allows sparing time in configuring the device, at the same time assuring that there is no inconsistency between two rockers that are meant to be configured in exactly the same way. To assign the same configuration is just a shortcut for the selection of configuration options; it is in no way implied that the two rockers share any of the involved communication objects (each rocker has its own independent objects).	
Function A	Rocker x = independent or single	**enabled /** disabled
	Enables or disables the capability to generate events for the first pushbutton of the rocker.	
Type	Rocker x = independent or single Function A = enabled	**send values or sequences** dimming shutter or venetian blind scene
	Determines the kind of function performed by the FIRST rocker input. Further parameters for the selected function will appear in the individual rocker configuration sections (see below).	
Function B	Rocker x = independent or single	disabled enabled in parallel with function A, as a single function **copy parameters from function A**
	Enables or disables the capability to generate events for the second pushbutton of the rocker. If not disabled, the pushbutton can be given an own independent function (enabled),used as an "alias" of the first input (in parallel), or perform the exact same kind of function as first input (copy parameters), but possibly basing on a different communication object.	
Type	Rocker x = independent or single Function B = enabled	**send values or sequences** dimming shutter or venetian blind scene
	Determines the kind of function performed by the SECOND rocker input. Further parameters for the selected function will appear in the individual rocker configuration sections (see below).	
Type	Rocker x = coupled	switch **dimming** shutter or venetian blind
	Determines the kind of function performed by the FIRST and SECOND rocker input. Further parameters for the selected function will appear in the individual rocker configuration sections (see below).	
Rocker to copy from	Rocker x = copy parameters from rocker (x > 1)	1..4*
	* The values that can be chosen obviously do not include the number of the rocker for which the selection is made.	

7.3.1 Independent or single: send values or sequences

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker x – Switching status [type], object n	Rocker x = independent or single Function x = enabled Type = send values or sequences	*according to the configuration* (**1-bit**)	CRWTU	*according to the configuration* (**[1.001] switch**)	5, 22 (1A, 1B) 39, 56 (2A, 2B) 73, 90 (3A, 3B) 107, 124 (4A, 4B)
	Up to 8 objects can be defined for binding with the same event. *The listed CO numbers are those referring to object nr.1; the COs for each subsequent object are following in sequence.* *To obtain the CO numbers for object number n, just add (n-1) to the listed numbers.* *E.g.: COs associated to input 3A (of Rocker 3) have numbers from 81 to 89. The number of CO no. 5 is therefore 81+(5-1) = 85.* **The size and type of the individual objects can be configured as described in following sections**.				

7.3.2 Independent or single: dimming

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker x – Switching command	Rocker x = independent or single Function x = enabled Type = dimming	1 bit	CRWTU	[1.001] switch	13, 30 (1A, 1B) 47, 64 (2A, 2B) 81, 98 (3A, 3B) 115, 132 (4A, 4B)
	Send a command to a dimming actuator to switch the light on or off. *The command is triggered by a short press on the input.* The value sent can be a fixed value or it can be toggled at each input activation.				
Rocker x – Dimming up / down / stop command	Rocker x = independent or single Function x = enabled Type = dimming	4 bit	CR-T-	[3.*] 3-bit control	14, 31 (1A, 1B) 48, 65 (2A, 2B) 82, 99 (3A, 3B) 116, 133 (4A, 4B)
	Send a command to a dimming actuator to change dimming intensity (brighter or darker). *Three values are used which mean start increase, start decrease or stop the change.*				

Increase: 1 0 0 0   Decrease: 0 0 0 1

Stop dimming: 0 0 0 0

Increase/decrease values are sent when a long press action occurs and stop value on press release.
The value sent can be a fixed value or it can be toggled at each input activation.

7.3.3   Independent or single: shutter or venetian blind

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker x – Dedicated stop command	Rocker x = independent or single Function x = enabled Type = shutter or venetian blind	1 bit	CRWTU	[1.017] trigger	13, 30 (1A, 1B) 47, 64 (2A, 2B) 81, 98 (3A, 3B) 115, 132 (4A, 4B)
	Immediately stop any movement of the blind. The object is sent on a short press if the blind mode is disabled and at the end of a long press if the venetian blind mode is enabled.				
Rocker x – Stop – step up/down command	Rocker x = independent or single Function x = enabled Type = shutter or venetian blind Blind mode = enabled	1 bit	CR-T-	[1.007] step	16, 33 (1A, 1B) 50, 67 (2A, 2B) 84, 101 (3A, 3B) 118, 135 (4A, 4B)
	Move the blind to fully open or fully closed position. The object is sent at the end of a long press.				
Rocker x – Move up / down command	Rocker x = independent or single Function x = enabled Type = shutter or venetian blind	1 bit	CRWTU	[1.008] up/down	17, 34 (1A, 1B) 51, 68 (2A, 2B) 85, 102 (3A, 3B) 119, 136 (4A, 4B)
	Increase or decrease the opening of the blind stepwise. The object is sent on a short press.				

7.3.4   Independent or single: scene

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker x – Scene number	Rocker x = independent or single Function x = enabled Type = scene	1 Byte	CR-T-	[17.*] Scene number [18.*] Scene control	18, 35 (1A, 1B) 52, 69 (2A, 2B) 86, 103 (3A, 3B) 120, 137 (4A, 4B)
	Store or recall a scene. The lowest 6 bits in the byte form the code of the scene, while the highest bit is the operation code (store or recall). 				

7.3.5 Coupled: switch

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker x – Switching command	Rocker x = coupled Function x = enabled Type = switch	1-bit	CRWTU	[1.001] switch	13, 47, 81, 115

7.3.6 Coupled: dimming

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker x – Switching command	Rocker x = coupled Function x = enabled Type = dimming	1 bit	CRWTU	[1.001] switch	13, 47, 81, 115
See notes for independent input.					
Rocker x – Dimming up / down / stop command	Rocker x = independent or single Function x = enabled Type = dimming	4 bit	CR-T-	[3.*] 3-bit control	14, 48, 82, 116
See notes for independent input.					

7.3.7 Coupled: shutter or venetian blind

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker x – Dedicated stop command	Rocker x = coupled Function x = enabled Type = shutter or venetian blind Blind mode = disabled	1 bit	CRWTU	[1.017] trigger	13, 47, 81, 115
See notes for independent input.					
Rocker x – Stop – step up/down command	Rocker x = coupled Function x = enabled Type = shutter or venetian blind Blind mode = enabled	1 bit	CR-T-	[1.007] step	16, 50, 84, 118
See notes for independent input.					
Rocker x – Move up / down command	Rocker x = coupled Function x = enabled Type = shutter or venetian blind	1 bit	CRWTU	[1.008] up/down	17, 51, 85, 119
See notes for independent input.					

7.4 Rocker *x: Function A/B* configuration

7.4.1   Independent or single

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

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

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

For all Type values:

Parameter name	Conditions	Values
Lock function	-	enabled / **disabled**
	Enables or disables the capability of locking the input through a remote command (telegram).	
Lock function – Invert lock device signal	Rocker x = independent or single Type = send values or sequences	**not inverted** / inverted
	Allows interpreting a "lock activate" telegram as unlock and vice-versa.	
Lock function – Lock after bus recovery	Rocker x = independent or single Type = send values or sequences	**no** / yes
	If active, after returning from a bus failure or power-off the device will retain the lock status it had before. Otherwise (in the default case), the device will restart in the non-locked condition.	

7.4.2 Independent or single: Lock function enabled

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker xx – Lock function	Rocker x = Independent or single Lock function = enabled	1 bit	C-W--	[1.003] enable	4, 21 (1A, 1B) 38, 55 (2A, 2B) 72, 89 (3A, 3B) 106, 123 (4A, 4B)

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

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

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

7.4.3 Independent or single: send values or sequences

Parameter name	Conditions	Values
Number of communication objects	Rocker x = independent or single Type = send values or sequences	1…8 (**1**)
	Number of communication objects configured in association with the button event.	
Lock function – Behaviour at locking	Rocker x = independent or single Type = send values or sequences	**none** as close or short press as open or long press
	Allows performing the operation associated to the specified event when a locking command is received. You can choose between operations linked to two possible closing (or short press, depending on the configuration) or opening (or long press) events.	
Lock function – Behaviour at unlocking	Rocker x = independent or single Type = send values or sequences	**none** as close or short press as open or long press
	Allows performing the operation associated to the specified event when an unlocking command is received. You can choose between operations linked to two possible closing (or short press, depending on the configuration) or opening (or long press) events.	
Event	Rocker x = independent or single Type = send values or sequences	**close / open contact** short / long press
	Type of event that should be used as trigger for an action.	
Long press time	Rocker x = independent or single Type = send values or sequences Event = short / long press	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Object n – Send delay	Rocker x = independent or single Type = send values or sequences	hh:mm:ss.ff (**00:00:00.00**)
	Delay before the object is transmitted on the bus. *By defining a delay after the event occurs and before the object value is sent, it is possible to associate a time defined sequence of values to an input event.*	
Object n – Send cyclically	Rocker x = independent or single Type = send values or sequences Number of communication objects = 1	none **off / value 1** on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated. *The cyclical transmission is only available if the number of communication objects to link is 1.*	
Object n – Cyclic sending interval	Rocker x = independent or single Type = send values or sequences Number of communication objects = 1 Send cyclically ≠ none	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Object n – send dimension	Rocker x = independent or single Type = send values or sequences	**1 bit value** 2 bits value 1 byte unsigned value 1 byte percentage 1 byte signed value 2 bytes unsigned value 2 bytes signed value 2 bytes floating value
	Defines size and type of the values to be sent when an event occurs.	
Object n – Close or Short press	Rocker x = independent or single Type = send values or sequences send dimension = 1 bit	none on off **toggle**

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

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker xx – Switching status [type] Object n	Rocker x = Independent or single Type = send values or sequences	*vedi tabella di seguito*	CR-TU	*vedi tabella di seguito*	5, 22 (1A, 1B) 39, 56 (2A, 2B) 73, 90 (3A, 3B) 107, 124 (4A, 4B)
	The listed CO numbers are those referring to object nr.1; the COs for each subsequent object are following in sequence. *To obtain the CO numbers for object number n, just add (n-1) to the listed numbers.* E.g.: COs associated to input 3A (of Rocker 3) have numbers from 81 to 89. The number of CO no. 5 is therefore 81+(5-1) = 85.				

Sizes and DPTs are as follows:

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

7.4.4 Independent or single: dimming

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

7.4.5   Independent or single: shutter or venetian blind

Parameter name	Conditions	Values
Long press time	Rocker x = independent or single Type = shutter or venetian blind	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Toggle mode	Rocker x = independent or single Type = shutter or venetian blind	**enabled /** disabled
	When enabled, causes the short press to toggle the on-off status of the destination CO; otherwise, a fixed status can be assigned to the short press.	
Up / Down action	Rocker x = independent or single Type = shutter or venetian blind Toggle mode = disabled	**up** down
	Defines the movement direction to be assigned to the button press.	
Blind mode	Rocker x = independent or single Type = shutter or venetian blind	enabled / **disabled**
	If blinds mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.	
Lock function – Behaviour at locking	Rocker x = independent or single Type = shutter or venetian blind	**none** up down
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Rocker x = independent or single Type = shutter or venetian blind	**none** up down
	Operation to perform when an unlocking command is received.	

7.4.6 Independent or single: scene

Parameter name	Conditions	Values
First scene number	Rocker x = independent or single Type = scene	1…64 (**1**)
	Main scene number to be assigned to button press. It is named "first" for the case that an alternative scene number is used.	
Learning mode	Rocker x = independent or single Type = scene	enabled / **disabled**
	When enabled, a long key press can be used to program the selected scene by storing the current parameters.	
Long press time	Rocker x = independent or single Type = scene Learning mode = enabled	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
Scene activation	Rocker x = independent or single Type = scene Learning mode = disabled	**send first scene only** toggle between two scenes
	Allows the key to be used to alternate between two different scenes.	
Second scene number	Rocker x = independent or single Type = scene Learning mode = disabled Scene activation = toggle between two scenes	1…64 (**2**)
	Alternate scene number to be assigned to button press.	
Lock function – Behaviour at locking	Rocker x = independent or single Type = scene	**none** send first scene send second scene
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Rocker x = independent or single Type = scene	**none** send first scene send second scene
	Operation to perform when an unlocking command is received.	

7.4.7   Coupled

For a *coupled* channel, all the parameters are referred to the single menu entry for Function xA and xB.

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

For all Type values:

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

7.4.8   Coupled: Lock function enabled

Object name	Conditions	Size	Flags	DPT	No. Comm. Obj.
Rocker xx – Lock function	Rocker x = coupled Lock function = enabled	1 bit	C-W--	[1.003] enable	4, 38, 72, 106

7.4.9   Coupled: switch

Parameter name	Conditions	Values
xA and xB use	Rocker x = coupled Type = switch	**A on, B off** A off, B on
	Allows to invert side A and side B functionalities	
Send cyclically	Rocker x = coupled Type = switch	none **off / value 1** on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclic sending interval	Rocker x = coupled Type = switch Send cyclically ≠ none	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at locking	Rocker x = coupled Type = switch	**none** on off toggle
	Value to be assigned to the object when a locking command is received.	
Lock function – Behaviour at unlocking	Rocker x = coupled Type = switch	**none** on off as previous
	Value to be assigned to the object when an unlocking command is received.	

7.4.10 Coupled: dimming

Parameter name	Conditions	Values
Long press time	Rocker x = coupled Type = dimming	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
xA and xB use	Rocker x = coupled Type = dimming	**A increases, B decreases** A decreases, B increases
Send cyclically	Rocker x = coupled Type = dimming	none **off / value 1** on / value 2 both off and on / both values
	Defines which of the values, if any, must be cyclically retransmitted whenever activated.	
Cyclic sending interval	Rocker x = coupled Type = dimming Send cyclically ≠ no	hh:mm:ss (**00:02:00**)
	Interval between cyclical transmissions.	
Lock function – Behaviour at locking	Rocker x = coupled Type = dimming	**none** on off toggle
	Value to be assigned to the object when a locking command is received.	
Lock function – Behaviour at unlocking	Rocker x = coupled Type = dimming	**none** on off as previous
	Value to be assigned to the object when an unlocking command is received.	

7.4.11 Coupled: shutter or venetian blind

Parameter name	Conditions	Values
Long press time	Rocker x = coupled Type = shutter or venetian blind	hh:mm:ss.fff (**00:00:03.000**)
	Minimum push time for a press in order to be recognized as a long press.	
xA and xB use	Rocker x = coupled Type = shutter or venetian blind	**A up, B down** A down, B up
Blind mode	Rocker x = coupled Type = shutter or venetian blind	enabled / **disabled**
	If blinds mode is enabled, the device sends "full movement" telegrams on long press and "step" telegrams on short press; if it is disabled, the device sends "full movement" telegrams on long press and "stop" telegrams on short press.	
Lock function – Behaviour at locking	Rocker x = coupled Type = shutter or venetian blind	**none** up down
	Operation to perform when a locking command is received.	
Lock function – Behaviour at unlocking	Rocker x = coupled Type = shutter or venetian blind	**none** up down
	Operation to perform when an unlocking command is received.	

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

7.5   Logic functions

The pushbutton interfaces EK-E32-TP and EK-E23-TP allow to use some useful logic functions (AND, OR, NOT and exclusive OR) in order to implement complex functions in the building automation system.

You can configure:

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

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

For each channel, a parameter *Delay after bus voltage recovery* is available: this parameter represents the time interval between the bus voltage recovery and the first reading of the input communication objects for evaluating the logic functions.

> **i**   In case of uncorrect connection of the input communication object or electrical trouble on bus resulting in a failed input reading request, the logic output of the corresponding channel can be calculated by setting the input values to default.

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

7.5.1   Parameter and communication object tables

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

Parameter name	Conditions	Values
Logic function		**disabled** / enabled
Logic operation	Logic function = enabled	**OR** / AND / XOR
	XOR (eXclusive OR)	
Delay after bus voltage recovery		**00:00:04.000 hh:mm:ss.fff** [range 00:00:00.000 … 00:10:55.350]
	Time interval between the bus voltage recovery and the first reading of the input communication objects for evaluating the logic functions	
Output cyclic transmission delay		**no sending** [other value in range 30 s … 120 min]
	No sending means that the output state of the logic function is updated on the bus only on change. Different values imply cyclic sending on the bus of the output state.	
Logic object x		**disabled** / enabled
Negated	Logic object x = enabled	**no** / yes
	Negando lo stato logico dell'ingresso corrispondente, è possibile realizzare logiche combinatorie articolate. Esempio: Output=(NOT(Oggetto logico 1) OR Oggetto logico 2).	
Read at startup	Logic object x = enabled	**no** / yes
Default value	Logic object x = enabled	**none** / off / on

Object name	Conditions	Dim.	Flags	DPT	Comm. Obj. No.
Logic function X – Input 1	Logic function X = enabled Logic object 1 = enabled	1 Bit	C-W--	[1.001] switch	189, 194, 199, 204, 209, 214, 219, 224
Logic function X – Input 2	Logic function X = enabled Logic object 2 = enabled	1 Bit	C-W--	[1.001] switch	190, 195, 200, 205, 210, 215, 220, 225
Logic function X – Input 3	Logic function X = enabled Logic object 3 = enabled	1 Bit	C-W--	[1.001] switch	191, 196, 201, 206, 211, 216, 221, 226
Logic function X – Input 4	Logic function X = enabled Logic object 4 = enabled	1 Bit	C-W--	[1.001] switch	192, 197, 202, 207, 212, 217, 222, 227
Logic function X – Output	Logic function X = enabled	1 Bit	C-W--	[1.001] switch	193, 198, 203, 208, 213, 218, 223, 228

8 Appendix

8.1 Summary of KNX communication objects

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

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

Object name	Size	Flags	DPT	No. Comm. Obj.
Rocker xx – Lock function	1 Bit	C-W--	[1.3] DPT_Enable	4, 21, 38, 55, 72, 89, 106, 123
Rocker x – Switching status [type], object n *	*Vedi tabella A1*	CR-TU	*Vedi tabella A1*	5..12, 22.. 29, 39.. 46, 56.. 63, 73.. 80, 90.. 97, 107.. 114, 124.. 131
	* I numeri degli O.C. elencati sono riferiti al primo di questi 8 oggetti (per ciascuno degli ingressi); gli O.C. degli oggetti successivi sono sequenziali. Per ottenere il numero dell'O.C. per l'n-esimo oggetto, aggiungere semplicemente (n-1) ai numeri riportati. *Es.: gli O.C. associati all'ingresso 3A hanno numeri a partire da 81. Il numero del 5°O.C. associato a tale ingresso sarà quindi 81+ (5-1) = 85.*			
Rocker xx – Switching command	1 Bit	CRWTU	[1.1] DPT_Switch	13, 30, 47, 64, 81, 98, 115, 132
Rocker xx – Dedicated stop command	1 Bit	CRWTU	[1.17] DPT_Trigger	13, 30, 47, 64, 81, 98, 115, 132
Rocker xx – Dimming up / down / stop command	4 Bit	CR-T-	[3.*] DPT_Control_Dimming, DPT_Control_Blinds	14, 31, 48, 65, 82, 99, 116, 133
Rocker xx – Stop – step up/down command	1 Bit	CR-T-	[1.7] DPT_Step	16, 33, 50, 67, 84, 101, 118, 135
Rocker xx – Move up / down command	1 Bit	CRWTU	[1.8] DPT_UpDown	17, 34, 51, 68, 85, 102, 119, 136
Rocker xx – Scene number	1 Byte	CR-T-	[17.1] DPT_SceneNumber [18.1] DPT_SceneControl	18, 35, 52, 69, 86, 103, 120, 137
Logic function X – Input 1	1 Bit	-WCTU-	[1.1] DPT_Switch	189, 194, 199, 204, 209, 214, 219, 224
Logic function X – Input 2	1 Bit	-WCTU-	[1.1] DPT_Switch	190, 195, 200, 205, 210, 215, 220, 225
Logic function X – Input 3	1 Bit	-WCTU-	[1.1] DPT_Switch	191, 196, 201, 206, 211, 216, 221, 226
Logic function X – Input 4	1 Bit	-WCTU-	[1.1] DPT_Switch	192, 197, 202, 207, 212, 217, 222, 227
Logic function X – Output	1 Bit	R-CT--	[1.1] DPT_Switch	193, 198, 203, 208, 213, 218, 223, 228

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

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

8.2 Warning

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

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

8.3 Other information

• This application manual is aimed at installers, system integrators and planners
• For further information on the product, please contact the ekinex® technical support at the e-mail address: support@ekinex.com or visit the website www.ekinex.com
• ekinex® is a registered trademark of EKINEX S.p.A.
• KNX® and ETS® are registered trademarks of KNX Association cvba, Brussels

© EKINEX S.p.A. 2014. The company reserves the right to make changes to this documentation without notice.