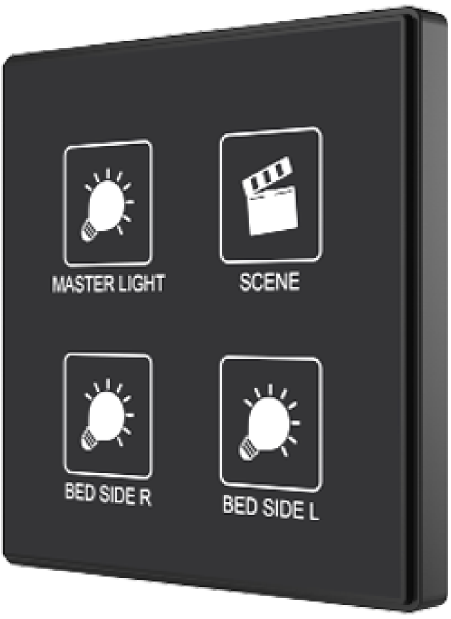


SQUARE TMD

1/2/4/6-Button Capacitive Touch Panel

ZVI-SQTMD1
ZVI-SQTMD2
ZVI-SQTMD4
ZVI-SQTMD6

Application Program Version: [1.0]
User Manual Version: [1.0]_a

www.zennio.com

USER MANUAL

CONTENTS

1 INTRODUCTION

1.1 SQUARE TMD

Square TMD is an evolution of Touch-Mydesign Plus, the KNX **multifunction capacitive touch switch** from Zennio. While it aims at covering similar functional requirements, Square TMD is offered at **a smaller size**, with either one, two, four or six capacitive touch buttons (according to the user's needs) together with LED indicators to confirm the press of the buttons as well as showing states.

Square TMD is a fully customisable solution for the room control, including hotel rooms, offices or any other environment where the user needs to control climate systems, lighting, blinds, scenes, etc.

The versatility of the above functions is enhanced by the built-in **analogue/digital inputs**, the **internal temperature sensor** and the **thermostat** function, as well as an elegant and **fully customisable design of the front glass** – customers can choose their button icons, texts and colours and even personalise the background with their pictures, logos, etc.

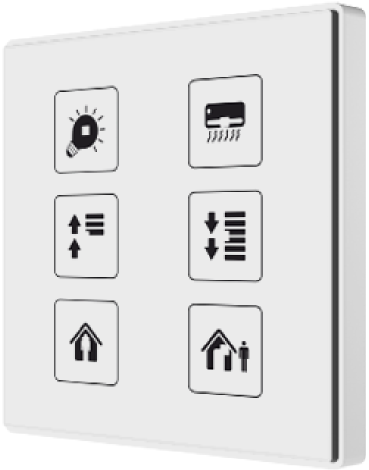


Figure 1 Square TMD.

The most outstanding features of Square TMD are:

- **Fully customisable** design of the front glass.
- **1 / 2 / 4 / 6 touch buttons**, which can operate as individual or pair controls:
  - **Pair button functions**: numeric counter, light dimmer, shutter controller, temperature setpoint, etc.
  - **Individual button functions**: binary switch, hold & release, scenes, light dimmer, shutter controller, numeric constants, etc.
- **Horizontally or vertically-oriented** configuration (only available for the two-button and six-button models).
- **Light indicator (LED)** for every button, the state of which (on or off) can depend on different factors according to the user’s needs.
- **Buzzer** for an audible acknowledgement of user actions (with the possibility of disabling it either by parameter or by object).
- Possibility of **locking / unlocking the touch panel** through binary orders or scenes, and of setting a timed/automatic locking of the device (**cleaning function**).
- **Welcome Back object** (binary or scene) which is sent to the KNX bus when a pulsation is detected after a certain period (configurable) of intactivity.
- **Two analogue/digital inputs** (for motion detectors, temperature probes, additional switches, etc.).
- **Thermostat** function.
- Built-in **temperature sensor**.

1.2 INSTALLATION

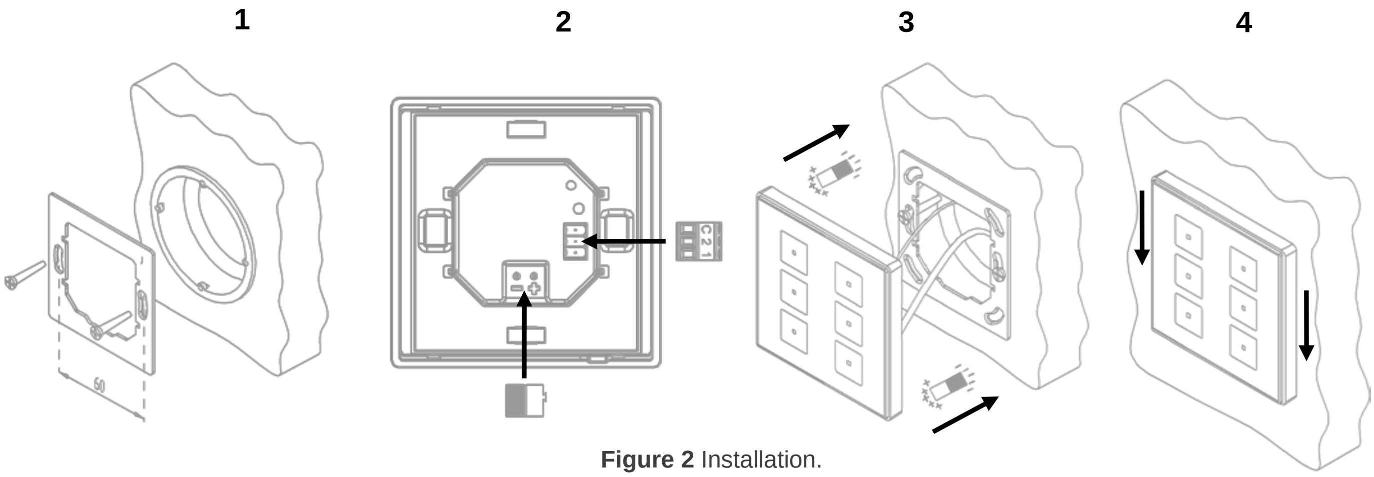


Figure 2 Installation.

To install the device, it is first necessary to mount the metallic plate into a square/round standard appliance box through the suitable screws. Next, the Square TMD is connected to the KNX bus through the corresponding terminal on the rear side of the device, and then the input terminal is as well connected to the rear of the device.

Once the input terminal and the KNX terminal are connected, the device can be easily mounted on the metallic plate by the action of the built-in magnets. After that, it is necessary to slide it downwards to fix it through the security anchorage system.

Finally, it is advisable to check that the device is properly installed, and that only its profile becomes visible from above, from below and from both sides (the metallic plate should be completely hidden).

This device does not need any external supply, as it is powered through the KNX bus.

1. Temperature sensor.
2. KNX bus connector.
3. Analogue-digital inputs.
4. Prog./Test button.
5. Prog./Test LED.
6. Magnets.
7. Status LED.
8. Touch buttons.

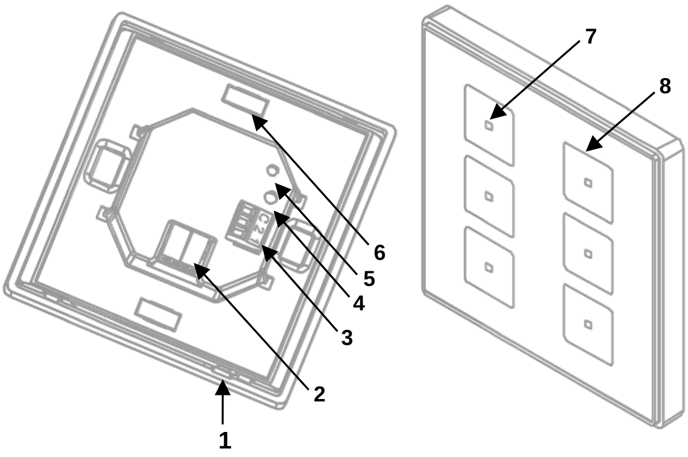


Figure 3 Element diagram.

The programming button (4) can be pressed with the help of a thin screw to set the device into the **Programming Mode**. After a short press, the programming LED (5) will light in red. Note that if this button is held while plugging the device into the KNX bus, the device will enter the **Safe Mode**. The LED will then blink in red.

Note: *whenever the device recovers the bus power, an immediate self-calibration process of the touch panel takes place. Please ensure to avoid touching the front glass while powering the device. If undesired effects arise during normal use, please disconnect the device from the bus and connect it again, making sure that the front glass is not touched during this process.*

For detailed information about the technical features of the device and for safety instructions and on the installation process, please refer to the **Datasheet** bundled with the original packaging of the device and also available at www.zennio.com.

2 CONFIGURATION

2.1 DEFAULT CONFIGURATION

To make the device perform the desired functions, several options need to be parameterised, either related to its **general behaviour** (horizontal/vertical orientation, sounds, LED brightness levels…) or to **advanced features** (lock procedure of the touch panel, cleaning function, welcome back object).

On the other hand, Square TMD features two opto-coupled inputs, each of which may be independently configured as **switches/sensors**, **push-buttons**, **temperature probes** or **motion detectors**, so that external accessories can be connected to the device. In the particular case of an external temperature probe (such as model **ZN1AC-NTC68** from Zennio), it will be possible to use it with independence of the built-in temperature sensor of the device, which implements its own communication objects and can be enabled and disabled in parameters.

ETS PARAMETERISATION

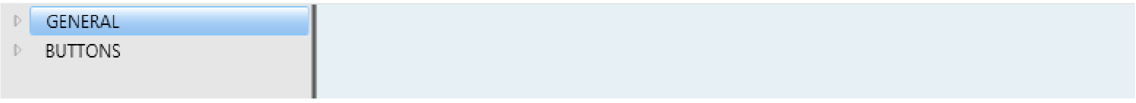


Figure 4 General.

After importing the corresponding database in ETS and adding the device into the topology of the project, the configuration process begins by right-clicking into the device and selecting *Edit parameters*.

This will bring the window shown in Figure 4, which contains the following tabs:

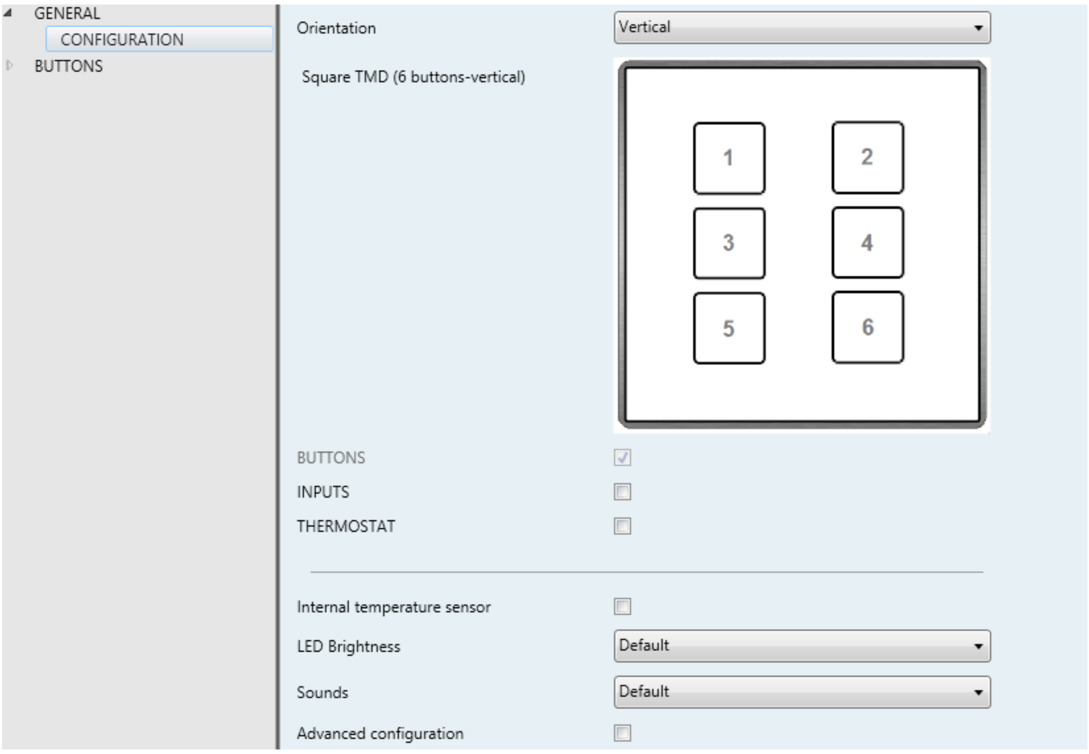
- **General**: contains the parameters that define the general behaviour of the device and that enable/disable additional features such as the thermostat or the external inputs. See section 2.2 for details.

- **Buttons**: contains the parameters involved in the configuration of the touch buttons of the device. See section 2.3 for details.

Depending on the configuration, other tabs may show in the tab tree on the left. These tabs will be explained in the next sections.

2.2 GENERAL

2.2.1 CONFIGURATION



- GENERAL
  - CONFIGURATION
- BUTTONS

Orientation: Vertical

Square TMD (6 buttons-vertical)

BUTTONS ☑

INPUTS □

THERMOSTAT □

Internal temperature sensor □

LED Brightness: Default

Sounds: Default

Advanced configuration □

Figure 5. General - Configuration

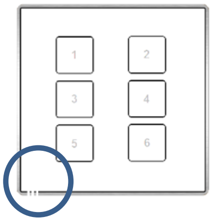
Note: *depending on the number of buttons of the device being configured there may be slight differences between the figures shown in this manual and the actual configuration sreens*.

This tab shows the following parameters:

- **Orientation**: allows assigning a **horizontal** or **vertical** orientation to the device, making it easier to identify the push-buttons during the configuration process (ETS will show a figure with the final distribution of the push-buttons). To prevent inconsisce in the configuration, please note the following criterium:

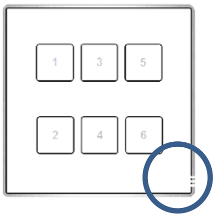
  Vertical (normal):

  Temperature probe hole on the left of the bottom side.



  Horizontal (rotated):

  Temperature probe hole on the bottom of the right side.



  Note: *this parameter is only available in some models (see section 2.3).*

- **Buttons**: read-only parameter to make it evident that the "Buttons" tab is always enabled in the tab tree on the left. See section 2.3 for details.

- **Inputs**: enables or disables the "Inputs" tab in the tree on the left, depending on whether the device will or will not be connected any external accessories. See section 2.4 for details.

- **Thermostat**: enables or disables the "Thermostat" tab in the tree on the left. See section 2.5 for details.

- **Internal Temperature Sensor**: enables or disables the "Temperature Sensor" tab in the tree on the left. See section 2.2.2 for details.

- **LEDs lighting**: sets whether the LEDs should make use of the pre-defined brightness levels ("Default") or of a user-defined configuration ("Custom").

  - "Default": the LEDs will remain off while in the "off" state, and at the maximum light level while in the "on" state.

  - "Custom": a specific tab will be included in the tab tree on the left so the integrator can set the desired light levels for the "off" and "on" states and whether to use the Night Mode or not. See section 2.2.3 for details.

- **Sounds**: sets whether the sound functions (button beeps, alarm and doorbell) should work according to the pre-defined configuration ("Default") or to a user-defined configuration ("Custom"). See section 2.2.4 for details.

- **Advanced Configuration**: enables or disables the "Advanced" tab in the tree on the left. See section 2.2.5 for details.

The project topology shows the following objects by default:

- **[General] Scene: Receive** and **[General] Scene: Send**: objects for respectively receiving and sending scene values from/to the KNX bus whenever it is necessary (e.g., when the user touches a button that has been configured to send scene commands; see section 2.3.4).

2.2.2 TEMPERATURE SENSOR

Square TMD is equipped with **one internal temperature probe** which can monitor the ambient temperature of the room, thus making the device capable of reporting it to the

KNX bus and of triggering certain actions when the temperature reaches specific values.

Please refer to the specific manual “**Temperature Sensor in Square TMD**” (available at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

2.2.3 LED LIGHTING

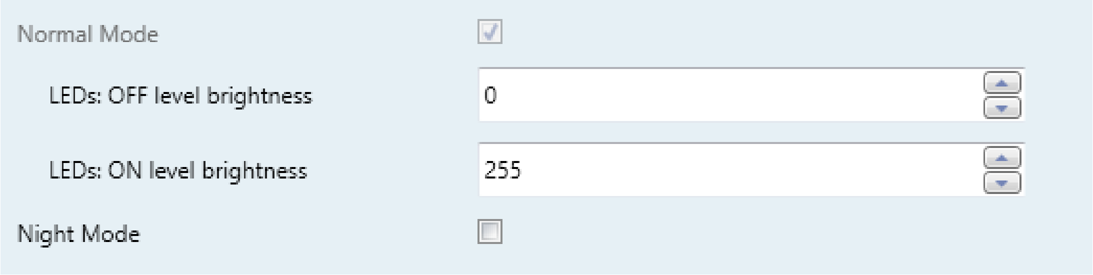
As anticipated in section 2.2, the integrator can configure the **LEDs of the buttons** to make use of the pre-defined brightness levels or of user-defined levels.

To begin with, each LED can commute between two states: **off** (which not necessarily means “no light”) and **on** (which not necessarily means “light on”). They will switch their states according to the options explained in ANNEX I: LED Illumination Modes.

Moreover, the LEDs can also commute between two operation modes: the **normal mode** and the **night mode**. The second one is optional. It is provided for temporary situations and environments where an excess of brightness may disturb the user. In such cases, it will be possible to switch the mode by means of a one-bit object and/or a scene object.

ETS PARAMETERISATION

After selecting “Custom” for “**LEDs Brightness**” (see section 2.2.1), a new tab will be incorporated into the tree on the left.



Normal Mode ☑

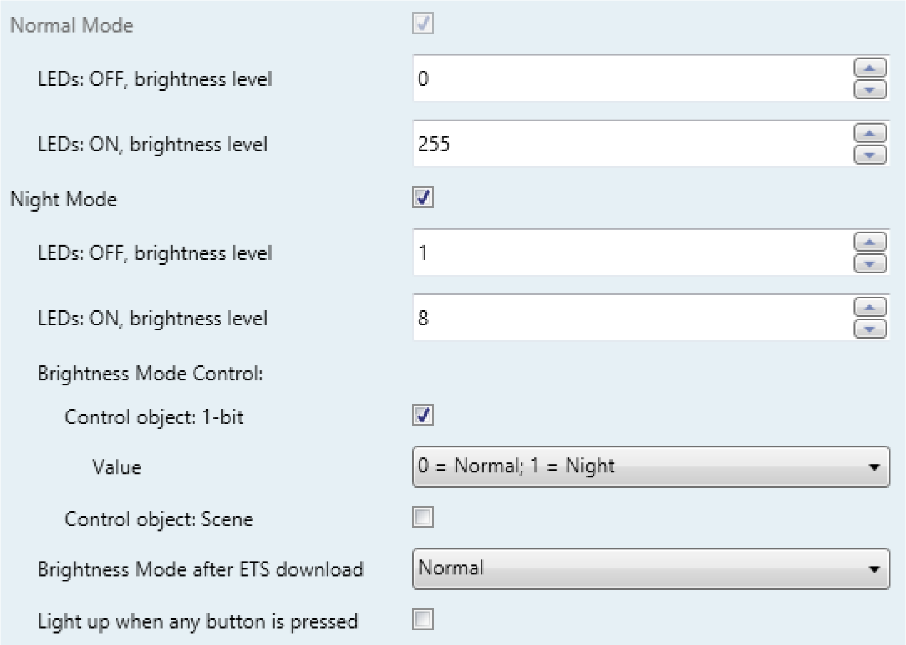
LEDs: OFF level brightness 0

LEDs: ON level brightness 255

Night Mode □

Figure 6 General – LEDs

The parameters shown in Figure 6 are equivalent to those for the “Default” configuration of parameter “**LEDs Brightness**”.



Normal Mode ☑

LEDs: OFF, brightness level: 0

LEDs: ON, brightness level: 255

Night Mode ☑

LEDs: OFF, brightness level: 1

LEDs: ON, brightness level: 8

Brightness Mode Control:

Control object: 1-bit ☑

Value: 0 = Normal; 1 = Night

Control object: Scene □

Brightness Mode after ETS download: Normal

Light up when any button is pressed □

Figure 7 General – LED Brightness.

- **Normal Mode**:
  - **LEDs: OFF, brightness level**: values between 0 (default) and 255.
  - **LEDs: ON, brightness level**: values between 0 and 255 (default).
- **Night Mode**: in case of being this mode necessary, this checkbox needs to be marked, which will incorporate some new parameters:
  - **LEDs: OFF, brightness level**: values between 0 and 255 (1 by default).
  - **LEDs: ON, brightness level**: values between 0 and 255 (8 by default).

In case of enabling the night mode, some more options can be configured:

- **Control object: 1 bit**: when marked, it will be possible to switch the mode by writing to a binary object ("**[General] LEDs & Disp - Brightness Mode**"). A specific drop-down list will show up to select which value should trigger which mode ("0 = Normal; 1 = Night" or "0 = Night; 1 = Normal").
- **Control object: Scene**: when marked, it will be possible to switch the mode by writing a certain scene value to "**[General] Scene: Receive**". Two specific textboxes will show up to enter what scenes (1 through 64) will trigger each mode.
- **Brightness Mode after ETS download**: sets which of the two modes ("Normal" or "Night") will be active after an ETS Download.

- **Light up when any button is pressed**: allows the LED brightness temporarily switch to the normal mode whenever a button previously enabled is touched by the user. If enabled, the following parameter will also show:
  - **Length of lighting**: time period after which the brightness switches back to night mode. The range is from 1 to 65535 seconds.

2.2.4 SOUNDS

Apart from the behaviour of the LEDs, it is possible to configure Square TMD so that a **brief beep** is emitted as an acoustic feedback in the following cases:

- **When a button is pressed**: short beep indicating that the user has pressed a button. This only applies to step controls, i.e., controls that walk through a certain range of values and that do not send a value after every touch, but only the final value after the last press.

- **When an action has been triggered**: sharper beep than the above one, indicating that a value is being sent to the KNX bus.

Enabling and disabling the button sounds can be done in parameters or through an object, being also possible to define in parameters whether the button sounds should be initially enabled or not.

On the other hand, Square TMD can also emit the following sounds on request (through the corresponding communication objects):

- **Doorbell sounds** (a single beep),

- **Alarm sounds** (a sequence of brief beeps with a higher pitch),

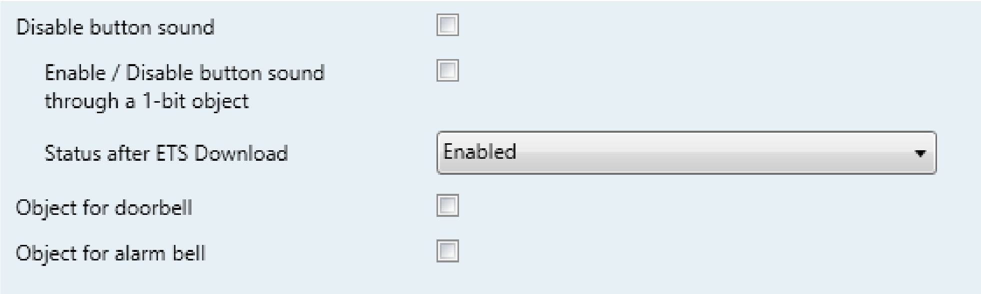
The alarm sequence will only stop when the alarm object gets deactivated or when the user touches any of the buttons (this will not trigger any action, only the alarm deactivation). Note that doorbell requests are ignored while the alarm is active.

A general parameter is provided to make the ETS parameterisation easier in case the doorbell and alarm functions are not required and the user feels comfortable with the default button beeps. On the other hand, customising the button, doorbell and alarm sounds involves a set of parameters, as explained next.

ETS PARAMETERISATION

In case the default button beep sound matches the requirements of the installation and the doorbell and alarm functions are not necessary, the "**Sounds**" parameter in the general **Configuration** tab (see section 2.2.1) can be set to "Default". This will also imply that the button beeps will be unconditional, as it will not be possible to disable this function through an object.

On the other hand, if set to "Custom", a specific tab named "**Sounds**" will show up in the tab tree on the left. The initial configuration of this screen is equivalent to the aforementioned default option. However, the following parameters will be configurable.



Disable button sound □

Enable / Disable button sound through a 1-bit object □

Status after ETS Download: Enabled

Object for doorbell □

Object for alarm bell □

Figure 8 General – Sounds.

- **Disable button sound**: enables or disables the button and action beeps. If enabled (default option) the following parameters will also be available:

  - **Enable / Disable button sounds through a 1-bit object**: makes it possible to disable / resume the button beeping function in runtime by writing to a specific object ("**[General] Sounds – Disabling button sound**").

    The values (0 or 1) that will disable / resume it are parameterisable through "**Value**".

  - **Button sound after ETS Download**: sets whether the button beeping function should start up enabled (default option) or disabled after an ETS download. This makes sense if the "**Enable/Disable Button Sounds Using 1-Bit Object**" checkbox (see below) is enabled too.

- **Object for doorbell**: enables or disables the doorbell function. If enabled, a specific object ("**[General] Sounds: Doorbell**") will be included into the project topology. The value that will trigger the sound (1 or 0) must be set in "**Value**".

- **Object for alarm bell**: enables or disables the alarm function. If enabled, a specific object ("**[General] Sounds: Alarm**") will be included into the project topology. The values (1 or 0) that will trigger the alarm sequence and then stop it must be entered in “**Value**”.

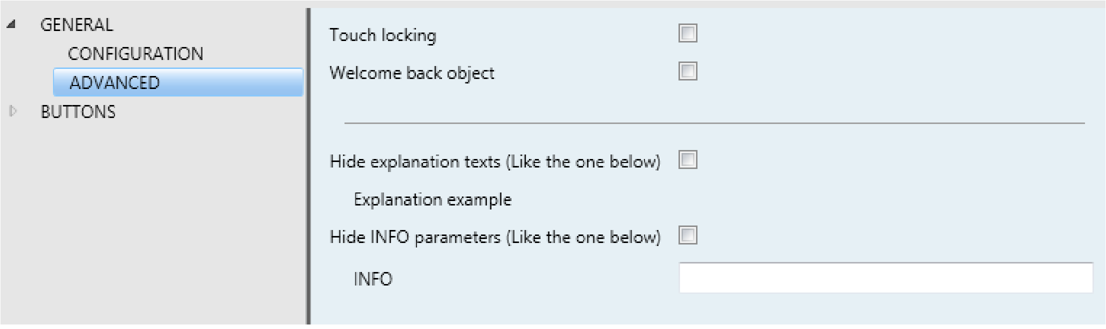
2.2.5 ADVANCED

As stated in section 2.2.1, an independent tab for the parameterisation of some advanced functions is shown in ETS if enabled from the “Configuration” tab.

These functions are explained next.

ETS PARAMETERISATION

The “Advanced” tab (Figure 9) contains the following parameters:



Touch locking □
Welcome back object □

Hide explanation texts (Like the one below) □
Explanation example
Hide INFO parameters (Like the one below) □
INFO

Figure 9 General - Advanced.

- **Touch locking**: enables or disables the “Touch locking” tab in the tree on the left. See section 2.2.6 for details.
- **Welcome back object**: enables or disables the “Welcome back” tab in the tree on the left. See section 2.2.7 for details.
- **Hide explanation texts**: displays or hides the explanation texts on the top of the parameter screen of some functions. Experienced users may prefer to hide these texts to have simpler parameter screens.
- **Hide INFO parameters**: displays or hides the INFO textboxes on the top of the parameter screen of each control. These textboxes have no functionality – they are offered for the convenience of the integrator, as they allow assigning a description to each control, which may be useful in case of later re-parameterisation of the project.

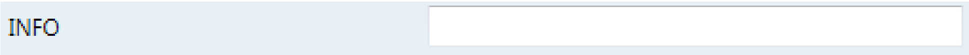

INFO

Figure 10 “Info” Textboxes.

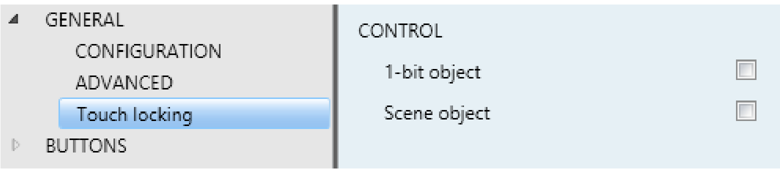
2.2.6 TOUCH LOCKING

The touch panel of Square TMD can be optionally locked and unlocked anytime by writing a configurable one-bit value to a specific object provided for this purpose. It can also be done through scene values.

While locked, pressing on the buttons will be ignored: no actions will be performed (and no LEDs will change their states) when the user touches on any of the controls.

ETS PARAMETERISATION

After enabling "**Touch Locking**" (see section 2.2.5), a new tab will be incorporated into the tree on the left.



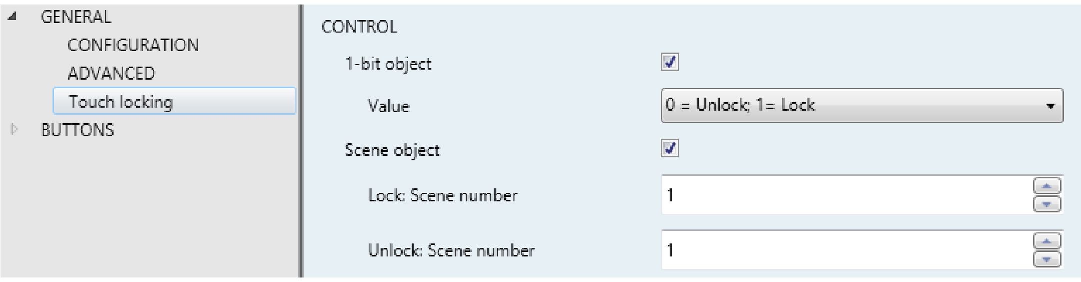
CONTROL

1-bit object □

Scene object □

Figure 11 General - Touch Locking.

Control of this function comprises two simple, non-exclusive checkboxes to select how the touch panel lock/unlock should be performed:



CONTROL

1-bit object ☑

Value: 0 = Unlock; 1= Lock

Scene object ☑

Lock: Scene number: 1

Unlock: Scene number: 1

Figure 12 General - Touch Locking: Control.

- Upon the reception of a one-bit value ("**1 bit object**").

  When marked, a specific drop-down list will show up to select which value should trigger which action ("0 = Unlock; 1 = Lock" or "0 = Lock; 1 = Unlock"). These values are to be received through object "**[General] Lock Touch**".

- Upon the reception of a scene value ("**Scene object**").

  When marked, two specific textboxes will show up to enter the scene numbers (1 - 64) that should trigger each action. These values are to be received through the general "**[General] Scene: Receive**" object.

2.2.7 WELCOME BACK OBJECT

Square TMD can send a specific object (the *welcome back object*) to the KNX bus when the user presses a touch button after a significant amount of time since the last press. Sending it or not can also depend on an **additional, configurable condition** consisting in the evaluation of up to five binary objects.

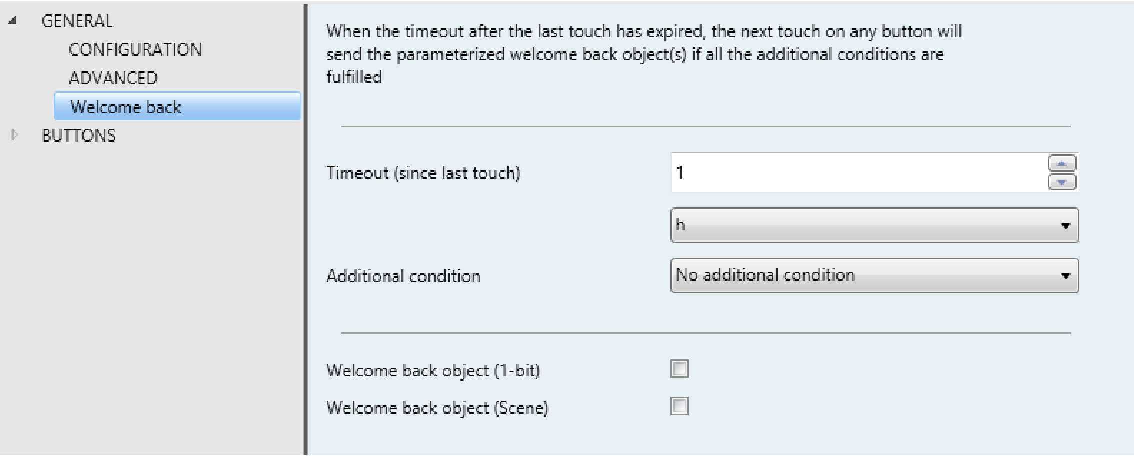
Any actions that in normal operation may be executed will not be if the welcome back object is sent to the bus. Thus, if the user presses a button and this causes that the welcome back object is sent, the normal action of that button will not be triggered. On the other hand, if the additional condition is not evaluated to true, the device will react normally. Hence, the action corresponding to the button touch will be executed.

The welcome back object can consist in a **one-bit** value or a **scene** value (or both), depending on the parameterisation.

ETS PARAMETERISATION

After enabling "**Welcome Back Object**" (see section 2.2.5), a new tab will be incorporated into the tree on the left.

This screen contains the following parameters:



When the timeout after the last touch has expired, the next touch on any button will send the parameterized welcome back object(s) if all the additional conditions are fulfilled

Timeout (since last touch): 1 h

Additional condition: No additional condition

Welcome back object (1-bit) □

Welcome back object (Scene) □

Figure 13 General - Welcome Back Object.

- **Timeout (since last touch)**: sets the minimum time (1 to 255 seconds, 1 to 255 minutes, or 1 to 255 hours) that should elapse after the last button touch before the next one triggers the execution of the welcome back function.

- **Additional Condition**: sets if sending the welcome back object should also depend on an external condition. The option by default is "No Additional Condition". The following are available too:

  - Do not send unless all additional conditions are 0: the welcome back object will only be sent if all the condition objects are found to have the value "0".

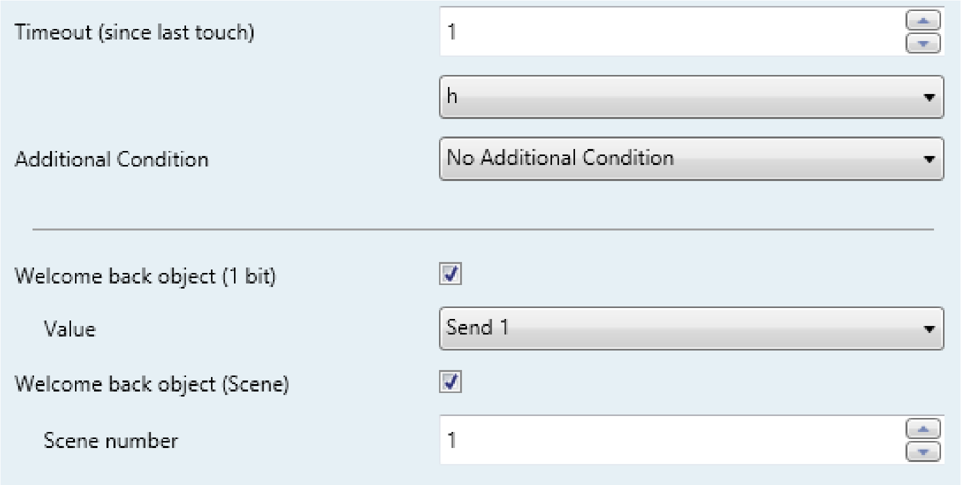
  - Do not send unless all additional conditions are 1: the welcome back object will only be sent if all the condition objects are found to have the value "1".

  - Do not send unless at least one of the additional conditions is 0: the welcome back object will only be sent if at least one of the condition objects is found to have the value "0".

  - Do not send unless at least one of the additional conditions is 1: the welcome back object will only be sent if at least one of the condition objects is found to have the value "1".

- **Welcome Back Object (1 Bit)**: checkbox to enable the sending of a 1-bit value (through "**[General] Welcome back**") when the welcome back function is triggered and the condition (if any) evaluates to true. The desired value (0 or 1) should to be set in "**Value**".

- **Welcome Back Object (Scene)**: checkbox to enable the sending of a scene run request (through "**[General] Scene: send**") when the welcome back function is triggered and the condition (if any) evaluates to true. The desired scene number (1 through 64) should to be entered in "**Value**".



Timeout (since last touch): 1

h

Additional Condition: No Additional Condition

Welcome back object (1 bit): ☑

Value: Send 1

Welcome back object (Scene): ☑

Scene number: 1

Figure 14 General - Welcome Back Object - Custom Configuration.

2.3 BUTTONS

As indicated in previous sections, Square TMD features **one, two, four or six capacitive buttons** (depending on the model) at the user's disposal for the execution of actions. Each will perform a specific function any time, as their functions do not depend on alternating menus, pages, etc.

The distribution of the buttons will depend on the model, being possible to configure them as single-button controls or in pairs by **combining any two of them** within one pair.

There are some differences in the button configuration depending on the model:

- **Square TMD 1**: due to the existence of only one push button, only one individual control is possible (two-button controls are not available). Moreover, it can only be configured under the normal (vertical) orientation (see section 2.2.1).

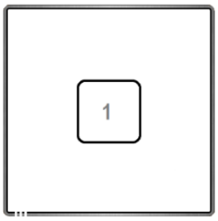


Figure 15 Square TMD 1.

- **Square TMD 2**: up to two one-button controls can be configured, or either one two-button control, under any of the two orientations (see section 2.2.1).

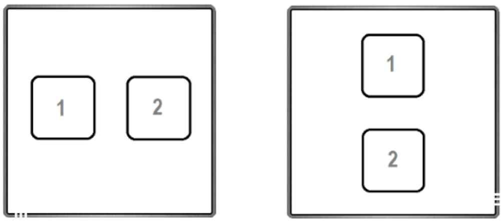


Figure 16 Square TMD 2. Orientación normal (izquierda) y rotada (derecha).

- **Square TMD 4** up to four one-button controls can be configured, or up to two two-button controls (by combining any two buttons for each pair). The figure shows which number identifies each button during the configuration process.

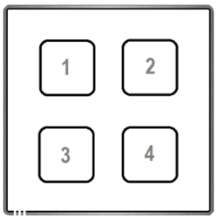


Figure 17 Square TMD 4.

- **Square TMD 6**: up to six one-button controls, or three two-button controls (by combining any two push buttons for each pair) can be configured, under any of the two orientations (see section 2.2.1). The figure shows which number identifies each button during the configuration process

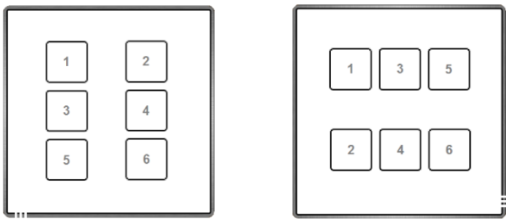


Figure 18 Square TMD 6. Orientación normal (izquierda) y rotada (derecha).

2.3.1 CONFIGURATION

The following is a list of the functions that can be assigned to each button.

- **Disabled** (the button will not react to user presses)**.**

- **Pair A**, **B or C** (the amount of available pairs depends on each model), being the function of such pair one of the following:

  - Switch (binary).
  - Two objects (short press / long press).
  - Light dimmer.
  - Shutter.

- **Individual** (one-button control):

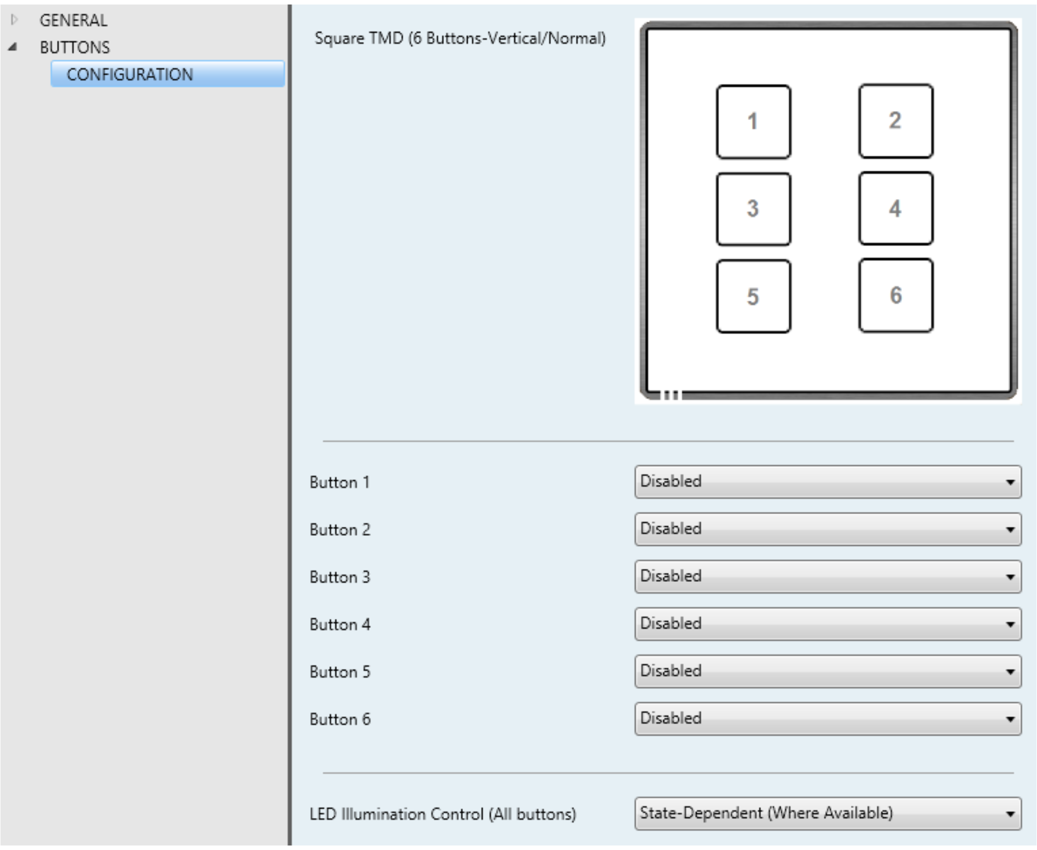
  - LED indicator.
  - Switch (Binary).
  - Hold & release.
  - Two objects (short press / long press).
  - Scene.
  - Scaling constant.
  - Counter constant.

- Float constant.
- Light dimmer.
- Shutter.

Apart from the button function itself, the integrator can select the desired behaviour of the button **LEDs**. The different **illumination modes** have been detailed in ANNEX I: LED Illumination Modes.

The next sections explain the configuration involved for each of the above functions.

ETS PARAMETERISATION



Square TMD (6 Buttons-Vertical/Normal)

Button 1: Disabled
Button 2: Disabled
Button 3: Disabled
Button 4: Disabled
Button 5: Disabled
Button 6: Disabled

LED Illumination Control (All buttons): State-Dependent (Where Available)

Figure 19 Buttons - Configuration.

As stated in section 2.1, an independent tab for the parameterisation of the buttons is shown in ETS by default, initially containing only a sub-tab named Configuration.

One drop-down list with the following options is shown per button:

- Disabled. See section 2.3.2.
- Pair X. Sets that this touch button will belong to a two-button control (where X is A, B or C, depending on the model). Once one pair has been assigned to

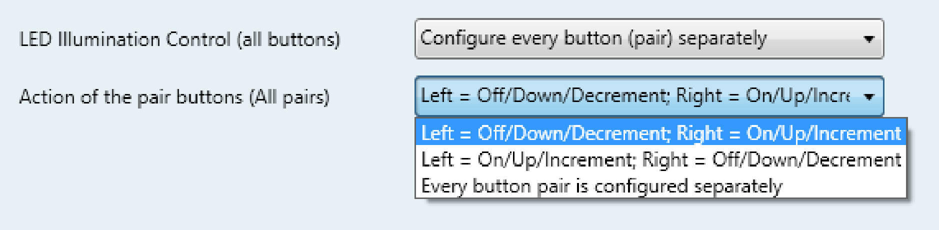
two buttons (and not before), a new tab ("**Pair X**") will show up in the tab tree, in order to configure the desired functionality. See section 2.3.3.

- Individual. Selecting this option brings a new tab ("**Button Ix**", where "x" depends on the button), which will make it possible to configure the functionality of that particular touch button. See section 2.3.4.

A drop-down list ("**LED Illumination Control (all buttons)**") is provided at the bottom of the window so a joint behaviour can be configured for the illumination of the LEDs. The options are (please refer to ANNEX I: LED Illumination Modes for details):

- Regular,
- State-Dependent (where available),
- State-Dependent (where available) (both LEDs),
- Dedicated Object,
- Configure every button (pair) separately.

In case of selecting "Configure every button (pair) separately", there will be a specific parameter **for each control** to specifically select the desired behaviour of the LED (or LEDs).



LED Illumination Control (all buttons): Configure every button (pair) separately

Action of the pair buttons (All pairs): Left = Off/Down/Decrement; Right = On/Up/Incre

Left = Off/Down/Decrement; Right = On/Up/Increment
Left = On/Up/Increment; Right = Off/Down/Decrement
Every button pair is configured separately

Figure 20 Buttons – Configuration – Advanced options.

Finally, if at least one two-button control is being configured (either Pair A or Pair B), an additional parameter ("**Action of the pair buttons (all pairs)**") will be available to determine an operation criterion. The options are:

- Left = Off/Down/Decrement; Right = On/Up Increment,
- Right = Off/Down/Decrement; Left = On/Up Increment,
- Every button pair is configured separately.

In case of selecting "Every Button Pair is Configured Separately", there will be a specific parameter in every two-button control for specifically selecting the desired behaviour.

2.3.2 DISABLED

While a button stays disabled, it will not be functional: touching on it will not cause the execution of actions, nor will make the associated LED light.

ETS PARAMETERISATION

This function has no related parameters.

2.3.3 PAIR

Buttons configured to work as a joint control can be assigned the following functions:

- **Switch**: pressing one of the two buttons will make Square TMD send a binary value to the bus, while pressing on the other will make it send the inverse binary value. It is possible to configure which one does what.

  Under a "**state-dependent**" LED illumination (see ANNEX I: LED Illumination Modes), the LED of the corresponding button will remain on/off according to the current state (on/off) of the switch. On the other hand, under a "**state-dependent (both LEDs)**" LED illumination, both of them will remain on while the switch is in the "on" state, and off while in the "off" state.

- **Two Objects (Short Press / Long Press)**: permits sending specific binary values both after a short or a long press on any of the two buttons (i.e., they will work as a joint control; for independent buttons, please configure them as individual). Different objects will be used for the short and long presses.

  Moreover, it is possible (in parameters) to make the "**state-dependent**" and "**state-dependent (both LEDs)**" LED illumination modes (see ANNEX I: LED Illumination Modes) depend on either one object or the other. However, if "**LED Illumination Control (All Buttons)**" (section 2.3.1) has been set to "state-dependent (where available)" only the short press object will be considered

- **Dimmer**: short-pressing one of the two buttons will make Square TMD send a switch-on order to the bus, while doing so on the other button will make it send a switch-off order.

Long presses will make it send a step dimming order (the value of which is configurable) to make a dimmer increase or decrease the light level (and a stop order as soon as the user releases the push button). It is possible to configure which button does what.

Under a "**state-dependent**" LED illumination (see ANNEX I: LED Illumination Modes), the LED of the corresponding button will remain on/off according to whether the current value of the light level status object (which should be updated by the actual dimmer) is greater than 0% or not. On the other hand, under a "**state-dependent (both LEDs)**" LED illumination, both together will remain on or off depending on such value.

- **Shutter**: this option permits making use of the two buttons to control a shutter actuator connected to the bus. Two alternative control methods are possible:

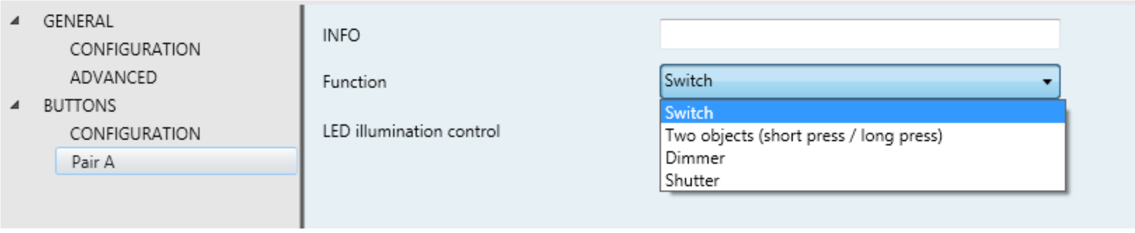
  - Standard: a long press will make the device send to the KNX bus an order to start moving the shutter (upwards or downwards, depending on the button), while a short press will make it send a stop order (which will be interpreted as an order to step up or to step down –depending on the button– if the shutter was not in motion and such function is available).

  - Hold & Release: as soon as the button is held, the device will send the KNX bus an order to start moving the shutter (upwards or downwards, depending on the button). Once the button is released, it will send an order to stop the shutter.

  The "**state-dependent**" and "**state-dependent (both LEDs)**" LED illumination modes are not available for this function (only the "**regular**" and "**dedicated object**" LED illumination are available). See ANNEX I: LED Illumination Modes for details.

ETS PARAMETERISATION

Once two buttons have been assigned to a particular pair, a specific tab ("**Pair X**") becomes available under "Buttons" in the tab tree.



GENERAL
CONFIGURATION
ADVANCED
BUTTONS
CONFIGURATION
Pair A

INFO

Function: Switch

Switch
Two objects (short press / long press)
Dimmer
Shutter

LED illumination control

Figure 21 Buttons - Pair A.

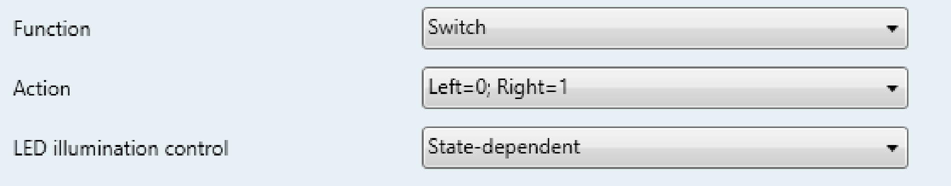
__Note__: *the INFO textboxes are functionless (they simply bring the option to add a description to each button of the project) and can be hidden if desired. See section 2.2.5*.

The main parameter that needs to be configured is:

- **Function**: sets the desired function for the button pair: "Switch", "Two objects (short press / long press)", "Dimmer" or "Shutter".

Depending on the function, some more parameters are shown, as described next. Please note that in the next pages the general notation "**[X]**" is used for the name of the communication objects, as "X" depends on the button pair (A, B or C).

Switch



Function: Switch

Action: Left=0; Right=1

LED illumination control: State-dependent

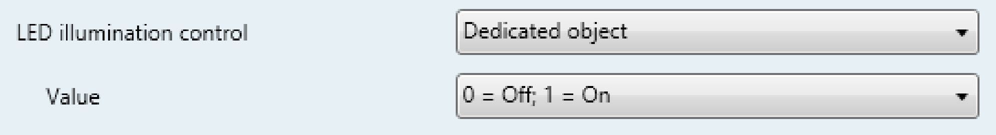
Figure 22 Pair Buttons – Switch.

- **Action**: assigns each of the two buttons the value to be sent through "**[Btn] [PX] Switch**" (which has the Write flag enabled, so the state of the switch can be updated from external devices). The options are "Left=0; Right=1" and "Left=1; Right=0".

  __Note__: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in **Action of the pair buttons** (see section 2.3.1)*.

- **LED illumination control**: sets the behaviour of the LEDs on the buttons. The options are "Regular", "State-dependent", "State-dependent (both LEDs)" and "Dedicated Object".

In case of selecting the latter, the object "**[Btn] [PX] Led On/Off**" will be included in the project topology and a new parameter to select the value to switch off and on the LED shows up:

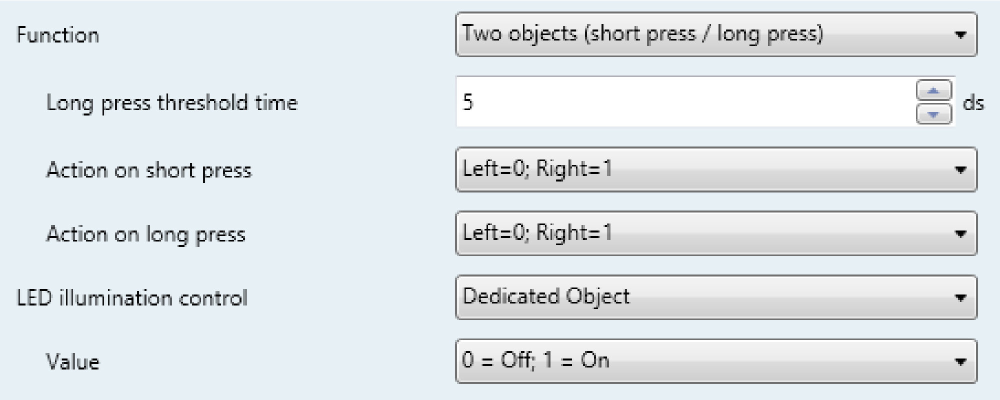


LED illumination control	Dedicated object
Value	0 = Off; 1 = On

Figure 23 LED illumination control - Dedicated object.

<u>**Note**</u>: *this parameter will remain hidden unless having selected "<u>Configure every button (pair) separately</u>" in "**LED Illumination Control (All Buttons)**" (see 2.3.1)*.

Two Objects (short Press / long press)



Function	Two objects (short press / long press)
Long press threshold time	5 ds
Action on short press	Left=0; Right=1
Action on long press	Left=0; Right=1
LED illumination control	Dedicated Object
Value	0 = Off; 1 = On

Figure 24 Pair Buttons - Two Objects (Short Press / Long Press).

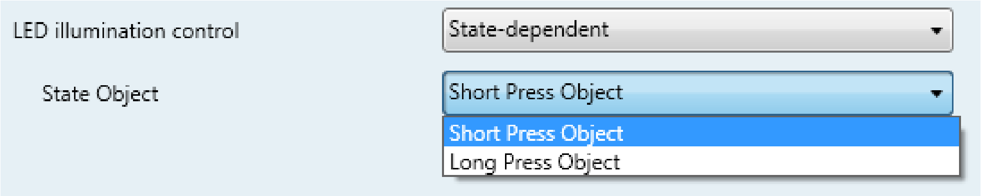
- **Long press threshold time**: sets the minimum time the user should hold the button in order to consider it a long press. The available range is 5 to 50 tenths of a second, being 5 tenths the default value.
- **Action on short press**: sets the value that will be sent through "**[Btn] [PX] Two switches - Short press**" after the user short-presses one of the two buttons.
  - "<u>Left=0; Right=1</u>".
  - "<u>Left=1; Right=0</u>".

<u>**Note**</u>: *this parameter will remain hidden unless having selected "<u>Every button pair is configured separately</u>" in **Action of the pair buttons** (see section 2.3.1)*.

- **Action on long press**: sets the value that will be sent through "**[Btn] [PX] Two switches: Long press**" after the user long-presses one of the two buttons.
  - "Left=0; Right=1".
  - "Left=1; Right=0".

  Note: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in* ***Action of the pair buttons*** *(see section 2.3.1)*.

- **LED illumination control**: analogous to the homonymous parameter for the "Switch" function (see above). In this case, however, selecting "State-dependent" or "State-dependent (both LEDs)" brings an additional parameter, "**State Object**", to make the state of the LEDs correspond either to object "**[Btn] [PX] Two switches: Short press**" ("Short Press Object") or to object "**[Btn] [PX] Two switches: Long press**" ("Long Press Object").

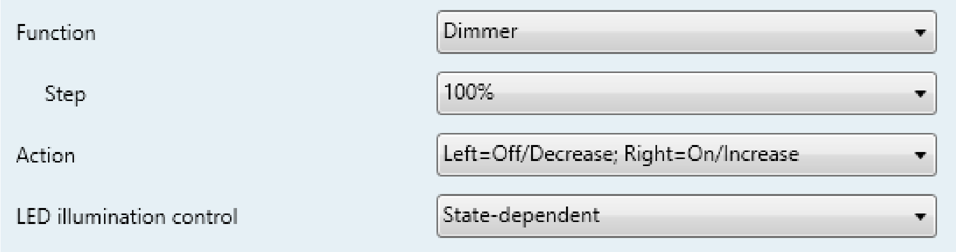


LED illumination control: State-dependent

State Object: Short Press Object

Figure 25 Pair Buttons - Two objects (short press / long press) - LED illumination control.

Dimmer



Function: Dimmer

Step: 100%

Action: Left=Off/Decrease; Right=On/Increase

LED illumination control: State-dependent

Figure 26 Pair Buttons – Dimmer.

The switch orders will be sent through the "**[Btn] [PX] Light - On/Off**" one-bit object, while the increase/decrease orders will be through the "**[Btn] [PX] Light - Dimming**" four-bit object.

On the other hand, the "**[Btn] [PX] Light Dimming (Status)**" one-byte object may be linked to the light level status object of the dimmer (in fact, this object is only intended to receive values from the bus, not to send them). As explained in 2.3.3, the state-dependent LED lighting will be determined by the value of this object (LED off at 0% and on at any other level).

The parameters for this function are:

- **Step**: defines the dimming step ("100%", "50%", "25%", "12.5%", "6.25%", "3.1%" or "1.5%") to be sent to the light dimmer with every increase / decrease order.

  Note: *since dimmers typically do not apply the new light level immediately (i.e., the step regulation is performed progressively) and since Square TMD sends an order to interrupt the step dimming once the user releases the button, it is advisable to configure a step of 100%. This way, the user can perform any dimming step by simply leaving the button pressed and then releasing it without needing to make successive button presses*.

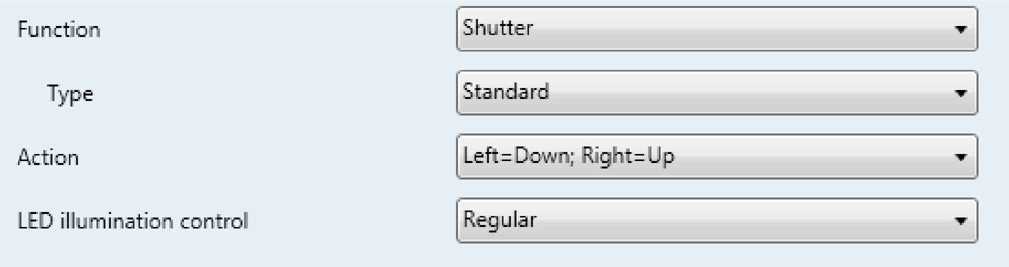
- **Action**: assigns each of the two buttons the order to be sent:

  - "Left=Off; Right=On".
  - "Left=On; Right=Off".

  Note: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in **Action of the pair buttons** (see section 2.3.1)*.

- **LED illumination control**: analogous to the homonymous parameter for the "Switch" and "Two Objects (short press / long press)" function (see above).

Shutter



Function: Shutter

Type: Standard

Action: Left=Down; Right=Up

LED illumination control: Regular

Figure 27 Pair Buttons – Shutter.

The move orders will be sent through "**[Btn] [PX] Shutter - Move**", while the stop orders will be sent through "**[Btn] [X] Shutter Stop/Step**" (for Standard type) or "**[Btn] [PX] Shutter - Stop**" (for Hold & Release type).

The parameters for this function are:

- **Type**: sets the desired behaviour of the buttons, "Standard" or "Hold & Release" (the differences between them have been explained in previous pages).
- **Action**: assigns each of the two buttons the order to be sent:
  - "Left=Down; Right=Up".
  - "Left=Up; Right=Down".

  Note: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in* ***Action of the pair buttons*** *(see section 2.3.1).*
- **LED illumination control**: analogous to the homonymous parameter for the previous functions (see above). In this case, the only options are "Regular" and "Dedicated Object".

2.3.4 INDIVIDUAL

Buttons configured to work as individual (separate) controls can be assigned any of the following control functions from the corresponding **Button IX** tab:

- **LED indicator**: user presses will not trigger any function although the LED will turn on or off depending on the values received from the bus.
- **Switch**: whenever the user touches the button, a binary value will be sent to the KNX bus. This value is configurable and may be 0 or 1, or alternate with every touch according to the sequence 1 → 0 → 1 → …

  Under a "**state-dependent**" LED illumination, the LED will remain on/off according to the current state (on/off) of the object.
- **Hold & Release**: as soon as the user touches the button, a binary value ("0" or "1", configurable) will be sent to the KNX bus. And as long as the user releases the button, another value ("0" or "1", also configurable) will be sent through the same object.

  The "**state-dependent**" LED mode is not available for this function.
- **Two Objects (Short Press / Long Press)**: specific binary values will be sent both after a short or a long press (a different object will be used in each

case). Under a "**state-dependent**" LED illumination, the LED will remain on/off according to the current state (on/off) of either one object or the other, which can be configured in parameters. However, if "**LED Illumination Control** (**All Buttons)**" (see section 2.3.1) has been set to "state-dependent (where available)", only the short press object will apply.

- **Scene**: after the user touches the button, an order to run a specific scene (configurable) will be sent to the bus. If enabled in parameters, orders to save the scene can also be sent to the bus after a three-second press on the button. The "**state-dependent**" LED mode is not available for this function.

- **Scaling Constant**: sends a percentage value (configurable) to the bus when the user touches the button. Under a "**state-dependent**" LED illumination, the LED will remain on/off depending on whether the current value of the object matches the one parameterised. This object can also be written from the bus, which will update the LED according to the new value.

- **Counter Constant**: sends an integer value (configurable) to the bus when the user touches the button. This value can be one-byte or two-byte sized, as well as signed or unsigned. The available ranges are shown next:

	1-byte	2-byte
Unsigned	0 – 255.	0 – 65535.
Signed	-128 – 127.	-32768 – 32768.

  The "**state-dependent**" LED illumination mode is analogous as for the Scaling Constant function.

- **Float Constant**: sends a two-byte floating point value (configurable) to the bus when the user touches the button. The available range is -671088.625 to 670760.9375.

  The "**state-dependent**" LED illumination mode is analogous as for the Scaling Constant and Counter Constant functions.

- **Dimmer**: implements a one-button light control that sends orders to the KNX bus, which can then be executed by light dimmers. These orders consist in:

  - Switch-on / Switch-off orders (on short presses).

- Step dimming orders (on long presses) and the subsequent stop order once the button is released.

Being a one-button control, the **switch orders will alternate** (on / off) for every short press, and so will do the step dimming orders (increase / decrease) for every long press. However, there are some exceptions:

- On a long press: an increase dimming order will be sent if the light is found to be off (according to the status object). On the other hand, a decrease order will be sent if it is found to be 100%.

- On a short press: a switch-on order will be sent if the light is found to be off (according to the status object). On the other hand, a switch-off order will be sent if it is found to be on (value greater than 0%).

Note that the device considers that the **current light level** is the value of a specific one-byte object provided to be written from the KNX bus (i.e., to receive feedback from the dimmer). This object is internally updated after a short or long press, but linking it to the real dimmer status is highly advisable.

Under a "**state-dependent**" LED illumination, the LED will remain on/off according to the value of the aforementioned status object (i.e., off when the value is 0% and on in any other case).

Note: *after a bus recovery, the light dimmer should send back the status object so the control and the LED update their own state, instead of simply recovering the previous one.*

- **Shutter**: implements a one-button shutter control that sends orders to the KNX bus, which can then be executed by an actuator.

  Two control types can be configured:

  - Standard: the device will react to both long and short presses, being possible to send the bus the following commands:

    - Move (raise / lower) orders (on **long presses**).

    - Stop / Step orders (on **short presses**).

Being a one-button control, the direction of the motion will alternate (upwards / downwards) for both the move and the step orders after every long press. However, there are some exceptions to this alternation:

- On a short press: a step-up order will be sent if the last long press made the shutter move up, or if the current position is found to be 100%. On the other hand, a step-down order will be sent if the last long press made the shutter move down or if the current position is found to be 0%.

- On a long press: a move-up order will be sent if the last short press caused a step-down order or if the current position is found to be 100%. On the other hand, a move-down order will be sent if the last short press caused a step-up order or if the current position is found to be 0%.

As usual in the KNX standard, **stop/step orders** are interpreted by the actuators as a request to move the slats one step up or down (in case the shutter is still) or as a request to interrupt the motion of the shutter (in case it is already moving up or down).

Square TMD is aware of the current **position of the shutter** through a specific object which should be linked to the analogous object of the shutter actuator in order to receive feedback.

➢ Hold & Release: the device will send an order to move the shutter when the button is touched, and the order to stop it as soon as it is released. Hence, short or long touches have the same effect: the shutter will remain in motion as long as the user keeps holding the button.

The direction of this motion (upwards or downwards) will **alternate** with every touch, according to the following sequence: downwards → upwards → downwards → …

However, there are some exceptions to this alternation:

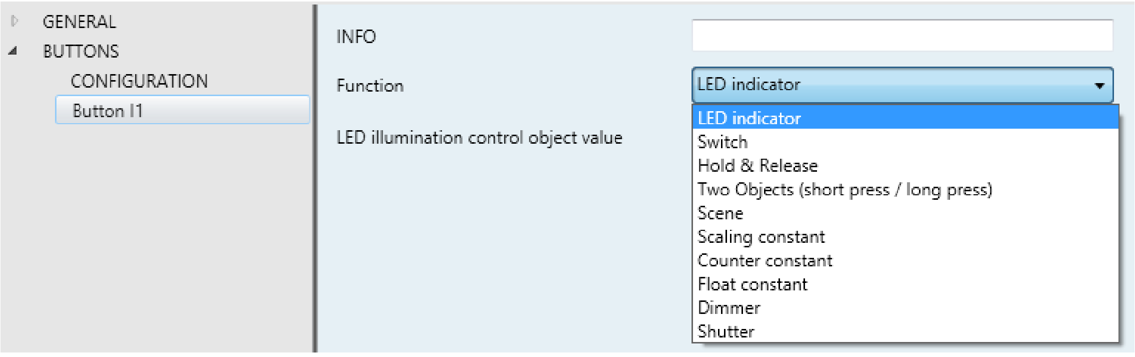
- If the position of the shutter is found to be 0%, the next order will lower the shutter.

- If the position of the shutter is found to be 100%, the next order will raise the shutter.

Square TMD is aware of the current **position of the shutter** through a specific object which should be linked to the analogous object of the shutter actuator, in order to receive feedback. This object is initialised with value "50%" after a download or a bus failure; therefore, the actuator is required to update it with the real value after the bus recovery.

The "**state-dependent**" LED illumination mode is not available for this function.

ETS PARAMETERISATION



INFO

Function: LED indicator

LED illumination control object value

Figure 28 Individual Button.

When an individual button has been enabled, a specific tab ("**Button In**") becomes available under "Buttons" in the tree on the left.

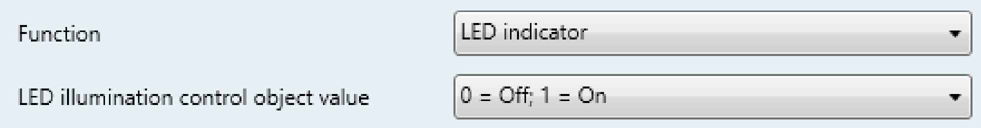
<u>**Note**</u>: *the INFO textboxes are functionless (they simply bring the option to add a description to each button of the project) and can be hidden if desired. See section 2.2.5*.

The main parameter that needs to be configured is:

- **Function**: sets the desired function for the button: "<u>LED indicator</u>", "<u>Switch</u>", "<u>Hold & Release</u>", "<u>Two Objects (short press / long press)</u>" or "<u>Scene</u>", "<u>Scaling constant</u>", "<u>Counter constant</u>", "<u>Float constant</u>", "<u>Dimmer</u>", "<u>Shutter</u>" and "<u>Enumeration</u>".

Depending on the function, some more parameters are involved (as described next). Please note that in the next pages "**[In]**" is used as a general notation for the communication objects, where "n" depends on the particular button pair.

LED Indicator



Function	LED indicator
LED illumination control object value	0 = Off; 1 = On

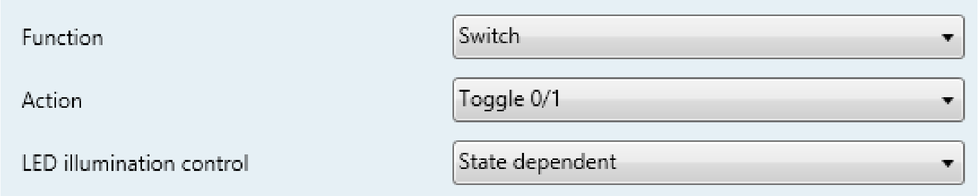
Figure 29 Individual Button - Disabled.

- **Function**: sets the behaviour of the LED of the button. The options are similar to those of the dedicated-object LED illumination available for other control types: "0=Off; 1=On" and "0=On; 1=Off"".

  Note: *this parameter does not depend on the option selected for "**LED Illumination Control (All Buttons)**" (see section 2.3.1).*

  After assigning this function to the button, object "**[Btn] [In] LED On/Off**" is included in the project topology, so that the values that determine the state of the LED at a given time can be received from the bus.

Switch



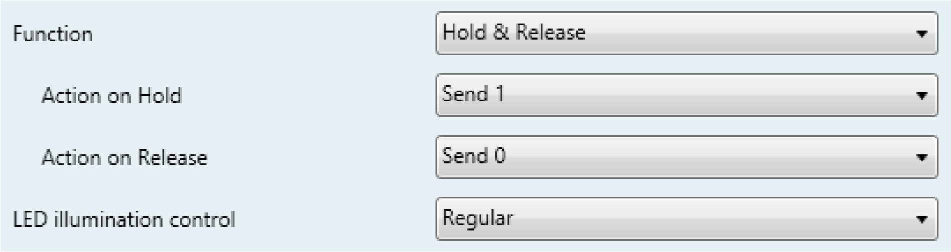
Function	Switch
Action	Toggle 0/1
LED illumination control	State dependent

Figure 30 Individual Button - Switch.

- **Action**: sets the value to be sent to the bus (through object "**[Btn] [In] Switch**") when the user touches the button. The options are "Send 0", "Send 1" and "Toggle 0/1".

- **LED illumination control**: analogous to the homonymous parameter for the "Disabled" function (see above). The options in this case are "Regular", "State dependent" and "Dedicated object".

Hold & Release



Function	Hold & Release
Action on Hold	Send 1
Action on Release	Send 0
LED illumination control	Regular

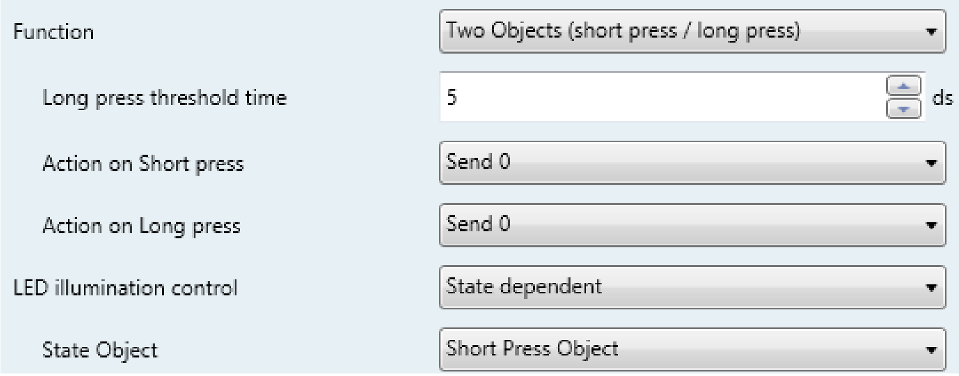
Figure 31 Individual Button – Hold & Release.

- **Action on Hold**: sets the value to be sent to the bus (through "**[Btn] [In] Hold & Release**") when the user touches the button. The options are "Send 0" and "Send 1" (default).

- **Action on Release**: sets the value to be sent to the bus (again, through "**[Btn] [In] Hold & Release**") when the user stops touching the button. The options are "Send 0" (default) and "Send 1".

- **LED illumination control**: analogous to the homonymous parameter for the "Disabled" and "Switch" functions (see above). The options in this case are "Regular" and "Dedicated object".

Two Objects (Short Press / Long Press)



Function	Two Objects (short press / long press)
Long press threshold time	5 ds
Action on Short press	Send 0
Action on Long press	Send 0
LED illumination control	State dependent
State Object	Short Press Object

Figure 32 Individual Button – Two Objects (short press / long press).

- **Long Press Threshold Time**: sets the minimum time the user should hold the button in order to consider it a long press. The available range is 5 to 50 tenths of a second, being 5 tenths the default value.

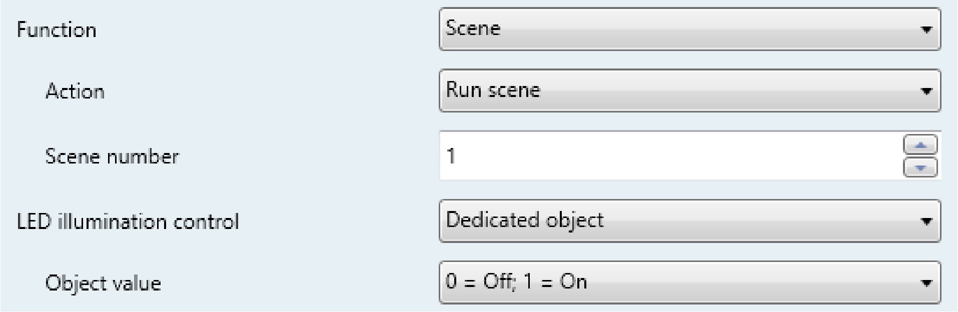
- **Action on Short press**: sets the value to be sent to the bus (through "**[Btn] [In] Two switches - Short press**") when the user short-presses the button. The options are "Send 0", "Send 1", "Toggle 0/1" and "Send 1-byte value". In case of selecting the latter, an additional parameter (“**Value**”) will be displayed to enter the desired one-byte value (0 - 255).

- **Action on Long press**: sets the value to be sent to the bus (through “**[Btn] [In] Two switches - Long press**”) when the user long-presses the button. The options are the same as for the short press.

- **LED illumination control**: analogous to the homonymous parameter for the “Disabled”, “Binary” and “Hold & Release” functions (see previous pages). The options in this case are “Regular”, “State dependent” and “Dedicated object”.

  Selecting “State dependent” brings an additional parameter, “**LED Illumination State Object**”, to make the state of the LED correspond either to object “**[Btn] [In] Two switches - Short press**” (“Short Press Object”) or to object “**[Btn] [In] Two switches - Long press**” (“Long Press Object”).

Scene



Function	Scene
Action	Run scene
Scene number	1
LED illumination control	Dedicated object
Object value	0 = Off; 1 = On

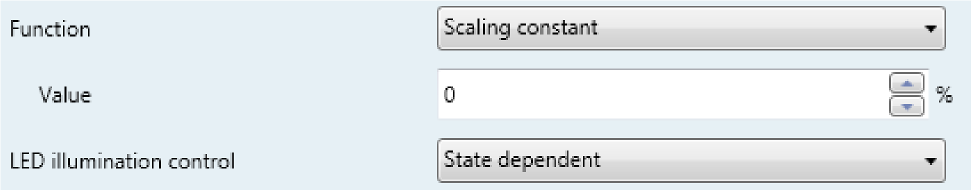
Figure 33 Individual Button – Scene.

- **Action**: sets whether the value to be sent to the KNX bus (through “**[General] Scene: send**”) when the user touches the button will always be a scene run request (“Run scene”) or –depending on the length of button press– a scene run or save request (“Run (short press) + Save (3s press) scene”).

- **Scene number**: number of the scene (1 – 64) to be sent to the bus, both in the case of the run requests and the save requests.

- **LED illumination control**: analogous to the homonymous parameter for all the above functions (see previous pages). The options in this case are “Regular”, and “Dedicated object”.

Scaling Constant / Counter Constant / Float Constant



Function: Scaling constant
Value: 0 %
LED illumination control: State dependent

Figure 34 Individual Button - Scaling Constant.

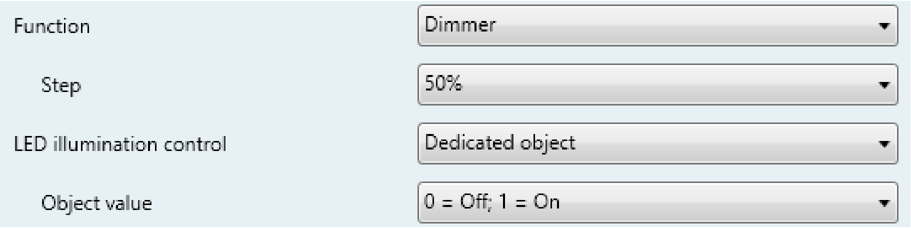
- **Value**: sets the value to be sent to the KNX bus when the user touches the button. The available range and the object through which the value is sent depends for each case, as the table below shows.

  In case of selecting Counter Constant, two specific parameters ("**Size**" and "**Signed**") will be displayed to respectively define the size of the constant ("1 byte" or "2 bytes") and whether it is a signed value or an unsigned value. Depending on that, the range and the name of the object will vary.

	Available Values	Name of the Object
Scaling Constant	0% – 100%	[Btn] [In] Scaling
Counter Constant	0 – 255 -128 – 127 0 – 65535 -32768 – 32767	[Btn] [In] Counter – 1 byte unsigned [Btn] [In] Counter – 1 byte signed [Btn] [In] Counter – 2 byte unsigned [Btn] [In] Counter – 2 byte signed
Float Constant	-671088.64 – 670760.96	[Btn] [In] Float

- **LED illumination control**: analogous to the homonymous parameter for all the above functions (see previous pages). The options in this case are "Regular", "State dependent" and "Dedicated object".

Dimmer



Function: Dimmer
Step: 50%
LED illumination control: Dedicated object
Object value: 0 = Off; 1 = On

Figure 35 Individual Button - Dimmer.

The (alternating) switch orders will be sent through the "**[Btn] [In] Light - On/Off**" one-bit object, while the (alternating) increase/decrease/stop orders will be through the "**[Btn] [In] Light - Dimming**" four-bit object.

On the other hand, the "**[Btn] [In] Light Dimming (Status)**" one-byte object may be linked to the light level status object of the dimmer (in fact, this object is only intended to receive values from the bus, not to send them). As explained in 2.3.42.3.3, the state-dependent LED lighting will be determined by the value of this object (LED off at 0% and on at any other level)

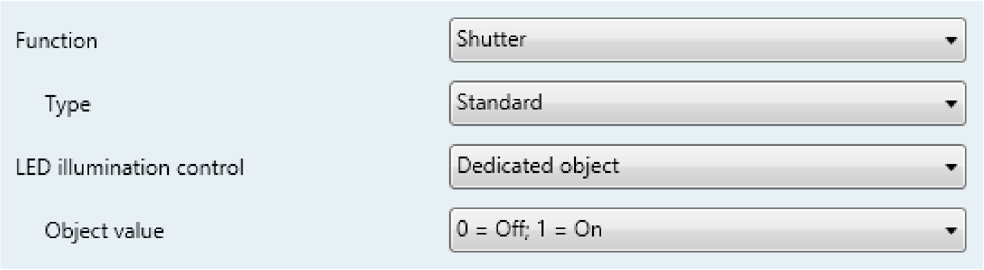
The parameters for this function are:

- **Step**: defines the dimming step ("100%", "50%", "25%", "12.5%", "6.25%", "3.1%" or "1.5%") to be sent (through "**[Btn] [In] Light - Dimming**") to the light dimmer with every long press.

  Note: *since dimmers typically do not apply the new light level immediately (i.e., the step is performed progressively) and since Square TMD sends an order to interrupt the step dimming once the user releases the button, it is advisable to configure a step of 100%. This way, the user can perform any dimming step by simply leaving the button pressed and then releasing it, without needing to make successive button presses*.

- **LED illumination control**: analogous to the homonymous parameter for all the above functions (see previous pages). The options in this case are "Regular", "State dependent" and "Dedicated object".

Shutter



Function: Shutter
Type: Standard
LED illumination control: Dedicated object
Object value: 0 = Off; 1 = On

Figure 36 Individual Button - Shutter.

The (alternating) move up/down orders will be sent through the "**[Btn] [In] Shutter - Move**" one-bit object, while the (alternating) step up/down orders will be through the "**[Btn] [In] Shutter - Stop / Step**" one-bit object.

Additionally, a one-byte object ("**[Btn] [In] Shutter Position**") is provided to link it to the position status object of the shutter actuator (in fact, this object is only intended to receive values from the bus, not to send them). As explained in 2.3.3, the alternation of the move up/down and the step up/down requests will be conditioned to these statuses, to prevent sending ineffective requests.

The parameters for this function are:

- **Type**: sets the desired control type: "Standard" or "Hold & Release", which have been described in section 2.3.3.
- **LED illumination control**: analogous to the homonymous parameter for all the previous functions (see previous pages). The options in this case are "Regular" and "Dedicated object".

2.4 INPUTS

Square TMD incorporates **2 analogue/digital inputs**, each configurable as a:

- **Binary Input**, for the connection of a pushbutton or a switch/sensor.
- **Temperature Probe**, to connect a temperature sensor (model ZN1AC-NTC68 S/E/F from Zennio).
- **Motion Detector**, to connect a motion detector (models ZN1IO-DETEC-P and ZN1IO-DETEC-X from Zennio).

Notes:

- *Older models of the Zennio motion detector (e.g., ZN1IO-DETEC and ZN1IO-DETEC-N) will not work properly with Square TMD.*

2.4.1 BINARY INPUT

Please refer to the specific user manual "**Binary Inputs in Square TMD**" , available at www.zennio.com.

2.4.2 TEMPERATURE PROBE

Please refer to the specific user manual "**Temperature Probe in Square TMD**", available at www.zennio.com.

2.4.3 MOTION DETECTOR

It is possible to connect motion detectors (models **ZN1IO-DETEC-P** and **ZN1IO-DETEC-X** from Zennio) to the input ports of Square TMD. This brings the device with the possibility of monitoring motion and presence in the room, as well as the light level. Depending on the detection, different response actions can be parameterised.

Please refer to the specific user manual "**Motion Detector in Square TMD**" (available at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

Notes:

- The ZN1IO-DETEC-P motion detector is compatible with a variety of Zennio devices. However, depending on the device it is actually being connected to, the functionality may differ slightly. Therefore, please refer specifically to the aforementioned user manual “**Zennio Motion Detector in Square TMD**”.

- Motion detectors with references ZN1IO-DETEC and ZN1IO-DETEC-N are **not compatible** with Square TMD (may report inaccurate measurements if connected to this device).

- When connected to Square TMD, the rear voltage selection micro-switch of model ZN1IO-DETEC-P should be set to position “**Type B**”.

2.5 THERMOSTAT

Square TMD implements **one Zennio thermostat** which can be enabled and fully customised.

Please refer to the specific manual “**Zennio Thermostat in Square TMD**” (available at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

ANNEX I: LED ILLUMINATION MODES

The central LED indicator of every button, by default (in most functions), will turn on for a brief instant whenever the button is touched. This behaviour is referred to as the "**Regular Illumination**".

However, in most cases it is possible to assign different behaviours to the LEDs. Which options are available will depend on the function parameterised for the button, but will always include some of the following:

- **Regular Illumination**: the LED will light for an instant once the button is touched.
- **State-Dependent Illumination**: the LED will or will not light, depending on the value of the communication object that corresponds to the function implemented by the button. The exact correspondence between the different values of the object and the different states of the LED may be slightly different from one type of control to another, and is detailed for each function.
- **State-Dependent Illumination (both LEDs)**: only applies to buttons configured as pair controls. The two LEDs of the control will light or not, depending on the value of the related object and on the particular control type parameterised for that pair of buttons. The only difference compared to the previous case is that, under "both LEDs", the two LEDs will always turn off or on simultaneously, as if it were a unique indicator consisting of two LEDs.
- **Dedicated Object**: the LED will light or not depending on the value ("0" or "1", configurable) of a binary, independent object. In the case of the pair controls, the value "0" will make one of the LEDs light (leaving the other one off), while the value "1" will make them switch their states.

Table 1 illustrates which of the above are configurable for each function.

		Disabled	Regular	State-dep.	State-dep. (both LEDs)	Dedicated object
PAIR	Switch		✓	✓	✓	✓
	Two Objects		✓	✓	✓	✓
	Dimmer		✓	✓	✓	✓
	Shutter		✓			✓
INDIVIDUAL	LED Indicator					✓
	Switch		✓	✓		✓
	Hold & Release		✓			✓
	Two Objects		✓	✓		✓
	Scene		✓			✓
	Constants		✓	✓		✓
	Dimmer		✓	✓		✓
	Shutter		✓			✓
DISABLED		✓				

Table 1 Functions vs. LED Illumination Options.

Note:

Regarding the LEDs, it is interesting to distinguish the following cases:

- *Disabled button: the LED will remain off, and the button will have no function.*
- *Button configured as "Individual" with "LED Indicator" function: the button will still have no function. The LED may be turned on/off through a binary object.*
- *Button configured as any other control type: the behaviour of the LED will be configurable according to the following table (being also possible to leave it turned off).*

Although the behaviour of the LEDs can be configured independently for each control, it is also possible to define a **general behaviour for all of them** (see section 2.3.1) thus not being then necessary to configure the same option multiple times. In case of opting for a general configuration, the options are:

- **Regular**.
- **State-Dependent (where available)**. Functions where “state-dependent” is not available will use the regular illumination.
- **State-Dependent (where available) (both LEDs)**. Functions where “state-dependent” is not available will use the regular illumination.
- **Dedicated Object**. One binary communication object per control will be included in the project topology so that the LED of every control turns on/off depending on its own object.

ETS PARAMETERISATION

For details on the parameterisation of the LED illumination modes please refer to the pages that cover the specific function being assigned to the button (section 2.3).

In case of desiring a **similar behaviour for all of the LEDs**, please find the parameter “**LED Illumination Control (all buttons)**” in the options of the General configuration screen. This parameter is explained in section 2.3.1.

ANNEX II: COMMUNICATION OBJECTS

- "**Functional range**" shows the values that, with independence of any other values permitted by the bus according to the object size, may be of any use or have a particular meaning because of the specifications or restrictions from both the KNX standard or the application program itself.

- The objects shown in this table are from model **Square TMD 6**. Please note that certain objects will not be available in models with less push buttons.

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	Name	Function
1	1 Byte	I	C - - W -	DPT_SceneNumber	0-63	[General] Scene: receive	0-63 (Run scene 1-64)
2	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	[General] Scene: send	0-63 / 128-191 (Run/Save scene 1-64)
3	1 Bit	I	C - - W -	DPT_Enable	0/1	[General] Touch Locking	0 = Unlock; 1 = Lock
	1 Bit	I	C - - W -	DPT_Enable	0/1	[General] Touch Locking	0 = Lock; 1 = Unlock
4	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] LEDs - Brightness mode	0 = Normal; 1 = Night
	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] LEDs - Brightness mode	0 = Night; 1 = Normal
5	1 Bit		C T - - -	DPT_Switch	0/1	[General] Welcome back	Switch object sent on Wake up
6	1 Bit	I	C - - W -	DPT_Enable	0/1	[General] Sounds - Disabling button sound	0 = Disable sound; 1 = Enable sound
	1 Bit	I	C - - W -	DPT_Enable	0/1	[General] Sounds - Disabling button sound	0 = Enable sound; 1 = Disable sound
7	1 Bit	I	C - - W -	DPT_Trigger	0/1	[General] Sounds - Doorbell	1 = Play a doorbell sound; 0 = Nothing
	1 Bit	I	C - - W -	DPT_Trigger	0/1	[General] Sounds - Doorbell	0 = Play a doorbell sound; 1 = Nothing
8	1 Bit	I	C - - W -	DPT_Alarm	0/1	[General] Sounds - Alarm	1 = Play alarm intermittent sounds; 0 = Stop alarm sounds
	1 Bit	I	C - - W -	DPT_Alarm	0/1	[General] Sounds - Alarm	0 = Play alarm intermittent sounds; 1 = Stop alarm sounds
9-13	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome back - Additional condition	Additional condition object X
14, 20, 26, 32, 38, 44	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [Ix] Switch	Send selected value on short press
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [Ix] Hold & Release	Send selected values on hold and release presses
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [Ix] Two objects - Short press	Send selected value on short press
	1 Bit		C T - - -	DPT_Switch	0/1	[Btn] [Ix] Light - On/Off	(Short press) Switch between On and Off
	1 Bit		C T - - -	DPT_Step	0/1	[Btn] [Ix] Shutter - Stop/Step	(Short press) 0 = Stop shutter / Step up; 1 =

							Stop shutter / Step down
	1 Bit		C T - - -	DPT_Trigger	0/1	[Btn] [Ix] Shutter - Stop	(End pressing) Stop shutter
15, 21, 27, 33, 39, 45	4 Bit	I	C T - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Btn] [Ix] Light - Dimming	(Long press) Switch between dimming up and down
16, 22, 28, 34, 40, 46	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [Ix] Shutter - Move	(Long press) 0 = Up ; 1 = Down
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [Ix] Shutter - Move	(Start pressing) Switch between up and down
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [Ix] Two objects - Long press	Send selected value on long press
17, 23, 29, 35, 41, 47	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [Ix] LED On/Off	0 = Off; 1 = On
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [Ix] LED On/Off	0 = On; 1 = Off
18, 24, 30, 36, 42, 48	1 Byte	I	C T - W -	DPT_Scaling	0% - 100%	[Btn] [Ix] Scaling	Send selected percentage value on short press
	1 Byte	I	C T - W -	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Counter - 1-byte unsigned	Send selected value on short press
	1 Byte	I	C T - W -	DPT_Value_1_Count	-128 - 127	[Btn] [Ix] Counter - 1-byte signed	Send selected value on short press
	2 Bytes	I	C T - W -	DPT_Value_2_Ucount	0 - 65535	[Btn] [Ix] Counter - 2-bytes unsigned	Send selected value on short press
	2 Bytes	I	C T - W -	DPT_Value_2_Count	-32768 - 32767	[Btn] [Ix] Counter - 2-bytes signed	Send selected value on short press
	2 Bytes	I	C T - W -	9.xxx	-671088.64 - 670760.96	[Btn] [Ix] Float	Send selected value on short press
	1 Byte	I	C T - W -	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Two objects - Short press (1-byte)	Send selected 1-byte value on short press
	1 Byte	I	C T - W -	DPT_Scaling	0% - 100%	[Btn] [Ix] Shutter Position	0 - 100 %
	1 Byte	I	C T - W -	DPT_Scaling	0% - 100%	[Btn] [Ix] Light Dimming (Status)	0 - 100 %
19, 25, 31, 37, 43, 49	1 Byte	I	C T - W -	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Two objects - Long press (1-byte)	Send selected 1-byte value on long press
50, 56, 62	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Switch	Left = 0; Right = 1
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Short press	Left = 1; Right = 0
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Short press	Left = 0; Right = 1

	1 Bit		C T - - -	DPT_Switch	0/1	[Btn] [PX] Light - On/Off	(Short press) Left = Off; Right = On
	1 Bit		C T - - -	DPT_Step	0/1	[Btn] [PX] Shutter - Stop/Step	(Short press) Left = Stop/Step down; Right = Stop/Step up
	1 Bit		C T - - -	DPT_Trigger	0/1	[Btn] [PX] Shutter - Stop	(End pressing) Left = Stop-down; Right = Stop-up
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Switch	Left = 1; Right = 0
	1 Bit		C T - - -	DPT_Switch	0/1	[Btn] [PX] Light - On/Off	(Short press) Left = On; Right = Off
	1 Bit		C T - - -	DPT_Step	0/1	[Btn] [PX] Shutter - Stop/Step	(Short press) Left = Stop/Step up; Right = Stop/Step down
	1 Bit		C T - - -	DPT_Trigger	0/1	[Btn] [PX] Shutter - Stop	(End pressing) Left = Stop-up; Right = Stop-down
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Switch	Lower = 0; Upper = 1
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Switch	Lower = 1; Upper = 0
	1 Bit		C T - - -	DPT_Switch	0/1	[Btn] [PX] Light - On/Off	(Short press) Lower = Off; Upper = On
	1 Bit		C T - - -	DPT_Switch	0/1	[Btn] [PX] Light - On/Off	(Short press) Lower = On; Upper = Off
	1 Bit		C T - - -	DPT_Step	0/1	[Btn] [PX] Shutter - Stop/Step	(Short press) Lower = Stop/Step down; Upper = Stop/Step up
	1 Bit		C T - - -	DPT_Step	0/1	[Btn] [PX] Shutter - Stop/Step	(Short press) Lower = Stop/Step up; Upper = Stop/Step down
	1 Bit		C T - - -	DPT_Trigger	0/1	[Btn] [PX] Shutter - Stop	(End pressing) Lower = Stop-down; Upper = Stop-up
	1 Bit		C T - - -	DPT_Trigger	0/1	[Btn] [PX] Shutter - Stop	(End pressing) Lower = Stop-up; Upper = Stop-down
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Short press	Lower = 0; Upper = 1
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Short press	Lower = 1; Upper = 0
51, 57, 63	4 Bit	I	C T - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) … 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) … 0xF (Inc. by 1%)	[Btn] [PX] Light - Dimming	(Long press) Left = Darker; Right = Brighter
	4 Bit	I	C T - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) … 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%)	[Btn] [PX] Light - Dimming	(Long press) Left = Brighter; Right = Darker

					... 0xF (Inc. by 1%)		
	4 Bit	I	C T - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Btn] [PX] Light - Dimming	(Long press) Lower = Darker; Upper = Brighter
	4 Bit	I	C T - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Btn] [PX] Light - Dimming	(Long press) Lower = Brighter; Upper = Darker
52, 58, 64	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Long press	Left = 0; Right = 1
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Long press	Left = 1; Right = 0
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Long press) Left = Down; Right = Up
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Start pressing) Left = Down; Right = Up
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Long press) Left = Up; Right = Down
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Start pressing) Left = Up; Right = Down
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Long press) Lower = Down; Upper = Up
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Long press) Lower = Up; Upper = Down
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Start pressing) Lower = Down; Upper = Up
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn] [PX] Shutter - Move	(Start pressing) Lower = Up; Upper = Down
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Long press	Lower = 0; Upper = 1
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] Two objects - Long press	Lower = 1; Upper = 0
53, 59, 65	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] LED On/Off	0 = On; 1 = Off
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn] [PX] LED On/Off	0 = Off; 1 = On
54, 60, 66	1 Byte	I	C T - W -	DPT_Scaling	0% - 100%	[Btn] [PX] Light Dimming (Status)	0 - 100 %
68	1 Byte	I	C - - W -	DPT_SceneControl	0-63; 128-191	[Thermostat] Scene Input	Scene Value
69	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[T1] Temperature Source 1	External Sensor Temperature

70	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[T1] Temperature Source 2	External Sensor Temperature
71	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[T1] Effective Temperature	Effective Control Temperature
72	1 Byte	I	C - - W -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[T1] Special Mode	1-byte HVAC Mode
73	1 Bit	I	C - - W -	DPT_Trigger	0/1	[T1] Special Mode: comfort	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[T1] Special Mode: comfort	0 = Off; 1 = On
74	1 Bit	I	C - - W -	DPT_Trigger	0/1	[T1] Special Mode: standby	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[T1] Special Mode: standby	0 = Off; 1 = On
75	1 Bit	I	C - - W -	DPT_Trigger	0/1	[T1] Special Mode: economy	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[T1] Special Mode: economy	0 = Off; 1 = On
76	1 Bit	I	C - - W -	DPT_Trigger	0/1	[T1] Special Mode: protection	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[T1] Special Mode: protection	0 = Off; 1 = On
77	1 Bit	I	C - - W -	DPT_Window_Door	0/1	[T1] Window Status (input)	0 = Closed; 1 = Open
78	1 Bit	I	C - - W -	DPT_Trigger	0/1	[T1] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
79	1 Byte	O	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[T1] Special Mode Status	1-byte HVAC Mode
80	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[T1] Setpoint	Thermostat Setpoint Input
	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[T1] Basic Setpoint	Reference Setpoint
81	1 Bit	I	C - - W -	DPT_Step	0/1	[T1] Setpoint Step	0 = -0.5ºC; 1 = +0.5ºC
82	2 Bytes	I	C - - W -	DPT_Value_Tempd	-670760.00 - 670760.00	[T1] Setpoint Offset	Float Offset Value
83	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[T1] Setpoint Status	Current Setpoint
84	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[T1] Basic Setpoint Status	Current Basic Setpoint
85	2 Bytes	O	C T R - -	DPT_Value_Tempd	-670760.00 - 670760.00	[T1] Setpoint Offset Status	Current Setpoint Offset
86	1 Bit	I	C - - W -	DPT_Reset	0/1	[T1] Setpoint Reset	Reset Setpoint to Default
	1 Bit	I	C - - W -	DPT_Reset	0/1	[T1] Offset Reset	Reset offset
87	1 Bit	I	C - - W -	DPT_Heat_Cool	0/1	[T1] Mode	0 = Cool; 1 = Heat

88	1 Bit	O	C T R - -	DPT_Heat_Cool	0/1	[T1] Mode Status	0 = Cool; 1 = Heat
89	1 Bit	I	C - - W -	DPT_Switch	0/1	[T1] On/Off	0 = Off; 1 = On
90	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] On/Off Status	0 = Off; 1 = On
91	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[T1] Control Variable (Cool)	PI Control (Continuous)
92	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[T1] Control Variable (Heat)	PI Control (Continuous)
93	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] Control Variable (Cool)	2-Point Control
	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] Control Variable (Cool)	PI Control (PWM)
94	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] Control Variable (Heat)	2-Point Control
	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] Control Variable (Heat)	PI Control (PWM)
95	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] Additional Cool	Temp >= (Setpoint+Band) => "1"
96	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] Additional Heat	Temp <= (Setpoint-Band) => "1"
97	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] PI State (Cool)	0 = PI signal 0%; 1 = PI signal greater than 0%
98	1 Bit	O	C T R - -	DPT_Switch	0/1	[T1] PI State (Heat)	0 = PI signal 0%; 1 = PI signal greater than 0%
99, 103	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[Ix] Current Temperature	Temperature sensor value
100, 104	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Overcooling	0 = No Alarm;1 = Alarm
101, 105	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Overheating	0 = No Alarm;1 = Alarm
102, 106	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Probe Error	0 = No Alarm;1 = Alarm
107	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[Internal Probe] Current Temperature	Temperature sensor value
108	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Internal Probe] Overcooling	0 = No Alarm;1 = Alarm
109	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Internal Probe] Overheating	0 = No Alarm;1 = Alarm
110	1 Byte	I	C - - W -	DPT_SceneControl	0-63; 128-191	[Motion Sensor] Scene Input	Scene Value
111	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	[Motion Sensor] Scene Output	Scene Value
112, 136	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Ix] Luminosity	0-100%
113, 137	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
114, 138	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
115, 139	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Ix] Presence State (Scaling)	0-100%
116, 140	1 Byte	O	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
117, 141	1 Bit	O	C T R - -	DPT_Occupancy	0/1	[Ix] Presence State (Binary)	Binary Value
	1 Bit	O	C T R - -	DPT_Trigger	0/1	[Ix] Presence: Slave Output	1 = Motion Detected
118, 142	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Ix] Presence Trigger	Binary Value to Trigger the Presence Detection
119, 143	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Ix] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
120, 144	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Ix] External Motion Detection	0 = Nothing; 1 = Motion detected by an external

							sensor
121, 145, 126, 150, 131, 155	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Ix][Cy] Detection State (Scaling)	0-100%
122, 146, 127, 151, 132, 156	1 Byte	O	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix][Cy] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
123, 147, 128, 152, 133, 157	1 Bit	O	C T R - -	DPT_Switch	0/1	[Ix][Cy] Detection State (Binary)	Binary Value
124, 148, 129, 153, 134, 158	1 Bit	I	C - - W -	DPT_Switch	0/1	[Ix][Cy] Channel Lock	According to parameters
125, 149, 130, 154, 135, 159	1 Bit	I	C - - W -	DPT_Switch	0/1	[Ix][Cy] Force State	0 = No Detection; 1 = Detection
160, 166	1 Bit	I	C - - W -	DPT_Switch	0/1	[Ix] Input Lock	1 = Locked; 0 = Unlocked
161, 167	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Short Press] 0	Sending of 0
	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Short Press] 1	Sending of 1
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Ix] [Short Press] 0/1 Switching	Switching 0/1
	1 Bit		C T - - -	DPT_UpDown	0/1	[Ix] [Short Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		C T - - -	DPT_UpDown	0/1	[Ix] [Short Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		C T - - -	DPT_UpDown	0/1	[Ix] [Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		C T - - -	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)
	1 Bit		C T - - -	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step down)
	1 Bit		C T - - -	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) … 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) …	[Ix] [Short Press] Brighter	Increase Brightness

					0xF (Inc. by 1%)		
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Darker	Decrease Brightness
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Brighter/Darker	Switch Bright/Dark
	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Short Press] Dimmer ON	Sending of 1 (ON)
	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Short Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Ix] [Short Press] Dimmer ON/OFF	Switching 0/1
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Run Scene	Sending of 0 - 63
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Save Scene	Sending of 128 - 191
	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Switch/Sensor] Edge	Sending of 0 or 1
	1 Byte		C T - - -	DPT_Value_1_Ucount	0 - 255	[Ix] [Short Press] Constant Value (Integer)	0 - 255
	1 Byte		C T - - -	DPT_Scaling	0% - 100%	[Ix] [Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		C T - - -	DPT_Value_2_Ucount	0 - 65535	[Ix] [Short Press] Constant Value (Integer)	0 - 65535
	2 Bytes		C T - - -	9.xxx	-671088.64 - 670760.96	[Ix] [Short Press] Constant Value (float)	Float value
162, 168	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Ix] [Short Press] Shutter Position (Status input)	0% = Top; 100% = Bottom
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Ix] [Short Press] Dimming Status (input)	0% - 100%
163, 169	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Long Press] 0	Sending of 0
	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Long Press] 1	Sending of 1
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Ix] [Long Press] 0/1 Switching	Switching 0/1
	1 Bit		C T - - -	DPT_UpDown	0/1	[Ix] [Long Press] Move Up	Sending of 0 (Up)

						Shutter	
	1 Bit		C T - - -	DPT_UpDown	0/1	[Ix] [Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		C T - - -	DPT_UpDown	0/1	[Ix] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		C T - - -	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)
	1 Bit		C T - - -	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step down)
	1 Bit		C T - - -	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Long Press] Brighter	Long Pr. -> Brighter; Release -> Stop
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Long Press] Darker	Long Pr. -> Darker; Release -> Stop
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Long Press] Brighter/Darker	Long Pr. -> Brighter/Darker; Release -> Stop
	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Long Press] Dimmer ON	Sending of 1 (ON)
	1 Bit		C T - - -	DPT_Switch	0/1	[Ix] [Long Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Ix] [Long Press] Dimmer ON/OFF	Switching 0/1
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Run Scene	Sending of 0 - 63
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Save Scene	Sending of 128 - 191
	1 Bit		C T - - -	DPT_Alarm	0/1	[Ix] [Switch/Sensor] Alarm:	1 = Alarm; 0 = No Alarm

						Breakdown or sabotage	
	2 Bytes		C T - - -	9.xxx	-671088.64 - 670760.96	[Ix] [Long Press] Constant Value (float)	Float value
	2 Bytes		C T - - -	DPT_Value_2_Ucount	0 - 65535	[Ix] [Long Press] Constant Value (Integer)	0 - 65535
	1 Byte		C T - - -	DPT_Scaling	0% - 100%	[Ix] [Long Press] Constant Value (Percentage)	0% - 100%
	1 Byte		C T - - -	DPT_Value_1_Ucount	0 - 255	[Ix] [Long Press] Constant Value (Integer)	0 - 255
164, 170	1 Bit		C T - - -	DPT_Trigger	0/1	[Ix] [Long Press/Release] Stop Shutter	Release -> Stop Shutter
165, 171	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Ix] [Long Press] Dimming Status (input)	0% - 100%
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Ix] [Long Press] Shutter Position (Status input)	0% = Top; 100% = Bottom

Join and send us your inquiries
about Zennio devices:

http://zennioenglish.zendesk.com

Zennio Avance y Tecnología S.L.
C/ Río Jarama, 132. Nave P-8.11
45007 Toledo (Spain).

Tel. +34 925 232 002.
Fax. +34 925 337 310.
www.zennio.com
info@zennio.com