

## **5-Button Capacitive Touch Panel with Upper Graphic Display**

**ZVI-SQTMDD**

Application Program Version: [1.5]  
User Manual Version: [1.5]\_a

[www.zennio.com](http://www.zennio.com)

# CONTENTS

---

Contents .....	2
Document Updates .....	4
1 Introduction .....	5
1.1 Square TMD-Display .....	5
1.2 Installation.....	8
2 Configuration .....	10
2.1 Default Configuration.....	10
2.2 General .....	12
2.2.1 CONFIGURATION .....	12
2.2.2 Temperature Sensor .....	13
2.2.3 Screensaver.....	14
2.2.4 LEDs & Display .....	15
2.2.5 Sounds .....	18
2.2.6 ADVANCED.....	20
2.2.7 Translations .....	22
2.2.8 Touch Locking .....	24
2.2.9 Cleaning Function .....	26
2.2.10 Welcome Greeting.....	27
2.2.11 Welcome Back .....	28
2.3 Buttons .....	31
2.3.1 CONFIGURATION .....	31
2.3.2 Disabled .....	34
2.3.3 Pair .....	34
2.3.4 Individual .....	53
2.4 Display .....	66

---

2.4.1	CONFIGURATION .....	66
2.4.2	Disabled .....	69
2.4.3	Single.....	69
2.4.4	Double.....	82
2.4.5	orders.....	83
2.5	Inputs.....	85
2.5.1	Binary Input .....	85
2.5.2	Temperature Probe .....	85
2.5.3	Motion Detector .....	85
2.6	Thermostat .....	87
ANNEX I: LED Illumination Modes.....		88
ANNEX II: Communication Objects .....		91

## DOCUMENT UPDATES

---

Version	Changes	Page(s)
[1.5]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>• Optimisation of Thermostat module.</li> <li>• New DPT for the 1-byte object of the Translations functionality: <i>Language Selection</i>.</li> </ul>	-
[1.4]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>• Optimisation of the inputs, thermostat, motion detector and temperature probe modules.</li> </ul>	-
[1.3]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>• Optimisation of the start-up process.</li> </ul>	-
[1.2]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>• New option to select the temperature scale (Celsius or Fahrenheit).</li> </ul>	-
[1.1]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"> <li>• Internal optimisation of the motion detection function.</li> </ul>	-

# 1 INTRODUCTION

---

## 1.1 SQUARE TMD-DISPLAY

---

Square TMD-Display is an evolution of TMD-Display, the KNX **multifunction capacitive touch switch with an analogue display** from Zennio. While it aims at covering similar functional requirements, Square TMD-Display is offered at a **smaller size**.

Square TMD-Display provides **five capacitive buttons** aligned in a sole row, as well as **LEDs** to indicate user presses or states, and a **wide display** on the top of the front panel, as in ZAS and Roll-ZAS from Zennio.

Square TMD-Display 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.

Moreover, the **display** offers the possibility of showing useful information to the user about the control states and object values.



Figure 1 Square TMD-Display.

The most outstanding features of TMD-Display are:

- **1.8"** back-lit display with a resolution of **128x64** pixels.
- **Fully customisable** design of the front glass.
- **5 touch buttons** which can operate as individual or pair controls.
- **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 greeting on the display** and **Welcome Back object** (binary or scene).
- **Central touch area** with a customisable object sending when pressed.
- **Screensaver** function.
- Up to **5 switchable languages** for the displayed texts.
- **Two analogue/digital inputs** (for motion detectors, temperature probes, additional switches, etc.).
- **Thermostat** function.
- **Celsius and Fahrenheit** temperature scales for the on-screen indicators, being possible to select them in parameters or through communication object.
- Built-in **temperature sensor**.

The Square TMD-Display application program, as well as the TMD-Display One application program does, brings the user the possibility of configuring their Square TMD-Display as a **single-page user interface**. This means that the capacitive buttons and the LCD display will perform the same functions anytime, providing the user with an easy-to-use interface, with no menus or alternating pages.

Because of this, the design of the front glass can contain icons specific enough as to make the function of the touch buttons evident at a simple glance.

Some of the available functions are:

- On the touch buttons:
  - **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.
- On the display:
  - **Single-box indicator:** icon, number, text, time of day, climate, etc.
  - **Double-box indicator:** icon, number, text, time of day, climate, etc.

## 1.2 INSTALLATION

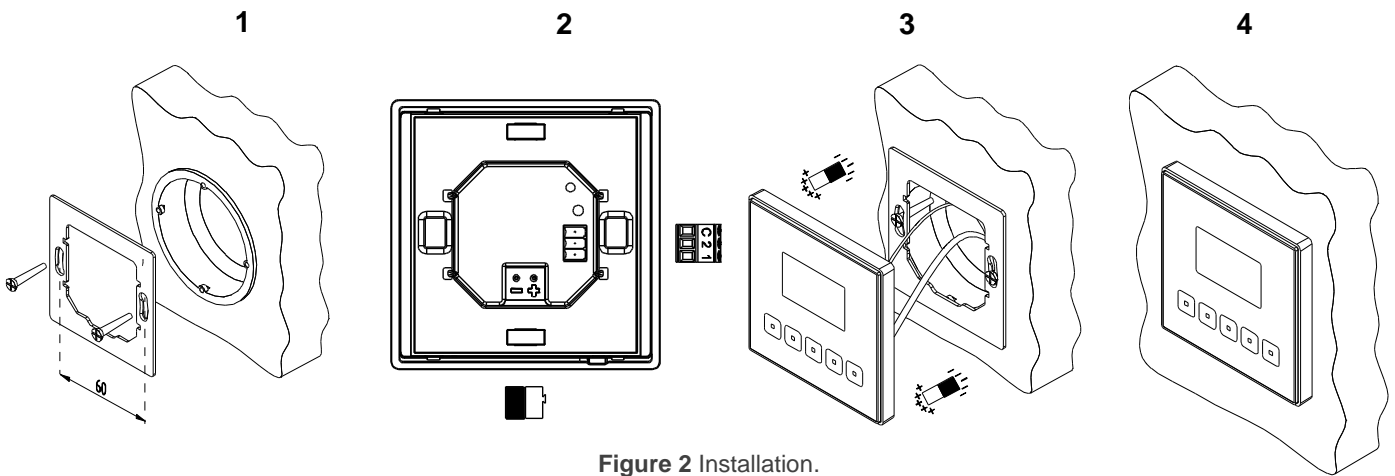


Figure 2 Installation.

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-Display 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.
3. Analogue-digital inputs.
4. Prog./Test. button.
5. Prog./Test LED.
6. Magnets.
7. Display.
8. Status LED.
9. Touch buttons.

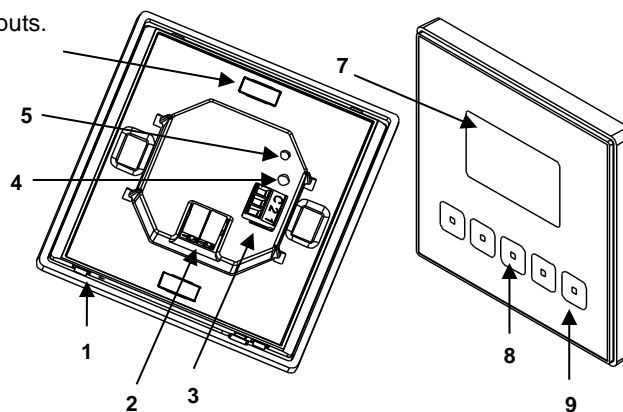


Figure 3 Element diagram.



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](http://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** (screensaver, sounds, LEDs and display brightness levels...) or to **advanced features** (lock procedure of the touch panel, cleaning function, welcome greeting on the display, welcome back object, etc.).

Some of the above options are common to other Zennio devices such as Touch-MyDesign, Roll-ZAS or Z41, while others are specific for TMD-Display and Square TMD-Display.

On the other hand, Square TMD-Display 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** or SQ-AmbientT **ZN1AC-SQAT**, both 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

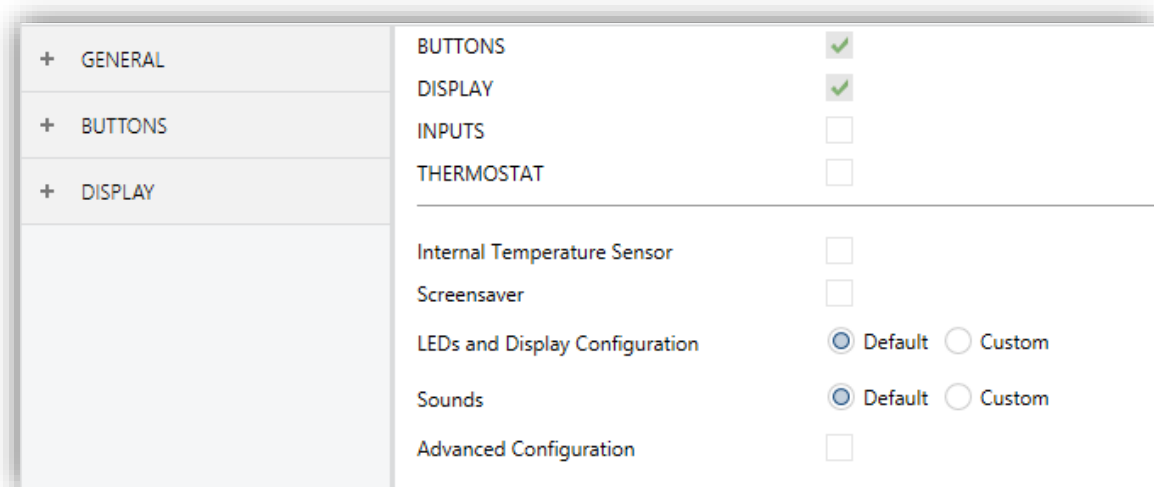


Figure 4 General.

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.
- **Display:** contains the parameters related to the configuration of the display of the device. See section 2.4 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

As shown in Figure 5, the “General” tab only contains one sub-tab by default: “Configuration”. However, it may include other tabs depending on the parameterisation.

### 2.2.1 CONFIGURATION

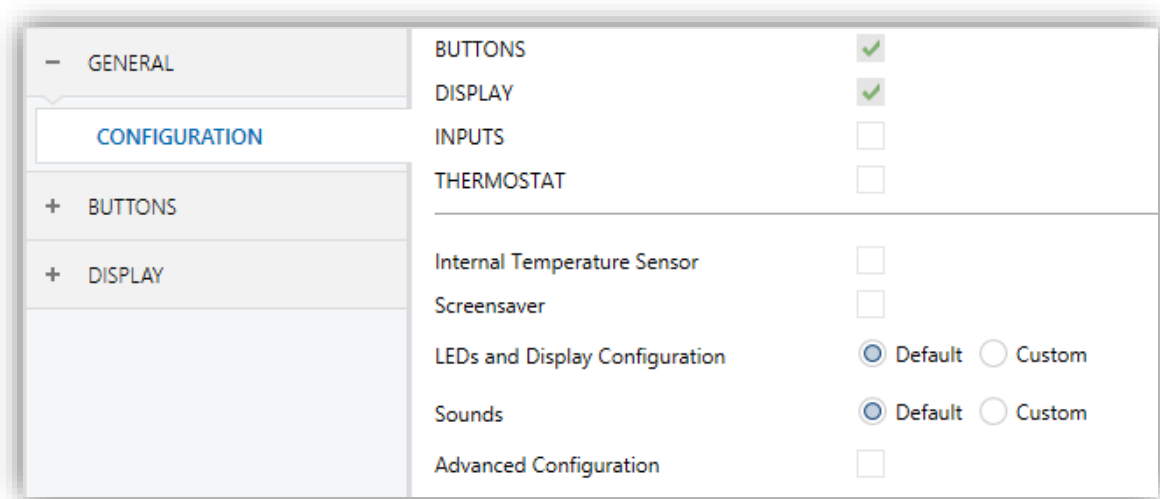


Figure 5. General - Configuration

This tab shows the following parameters:

- **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.
- **Display:** read-only parameter to make it evident that the “Display” tab is always enabled in the tab tree on the left. See section 2.4 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 2.5 for details.
- **Thermostat:** enables or disables the “Thermostat” tab in the tree on the left. See section 2.6 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.

- **Screensaver:** enables or disables the “Screensaver” tab in the tree on the left. See section 2.2.3 for details.
- **LEDs and Display Config.:** sets whether the LEDs and display 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. The display contrast level is set to 7.
  - “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. The display contrast level will also be configurable.

See section 2.2.4 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 2.2.5 for details.
- **Advanced Configuration:** enables or disables the “Advanced” tab in the tree on the left. See section 2.2.6 for details.

The project topology shows the following objects by default:

- **[General] Time of Day:** object for updating the time shown on the display.

**Important:** *the time of day must be set through the bus by means of an external reference. In the absence of power, the clock will become outdated.*
- **[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-Display 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-Display**” (available at the Zennio homepage, [www.zennio.com](http://www.zennio.com)) for detailed information about the functionality and the configuration of the related parameters.

### 2.2.3 SCREENSAVER

---

The screensaver is a special page that will only be shown after a period of inactivity, configurable by parameter.

It is possible to configure the screensaver to only show the current **Time**, the current **Temperature** (selecting the desired measurement source: the internal temperature sensor or an external value) or **both** (alternating every 5 seconds).

#### Notes:

- *The temperature shown will be truncated to the range [-99, 199] °C or to the equivalent [-142, 390] °F. See section 2.2.6 for details.*
- *The time of day must be set through the bus by an external reference. In addition, in the absence of power the clock will get outdated.*

The screensaver will disappear when if any of the enabled buttons is touched. It can be set by parameter whether, while the screensaver is active, this press should simply cause the screensaver interruption or, moreover –apart from hiding the screensaver–, the execution of the action corresponding to the pressed button.

### ETS PARAMETERISATION

---

This screen is divided into three parts: Activation, Content and Exit, containing the following parameters:

- **Activation timeout:** time to automatically show the screensaver after the last button touch. Range: 1 – 255 seconds, minutes or hours.
- **Time of Day:** sets whether to show the current time or not.
- **Temperature:** sets whether to show the current temperature or not. When enabled, the following parameter will show up:

- **Taken from:** defines the source of the temperature value: “Internal sensor” or “External reference”. The latter will bring a new two-byte object to the project topology (“**[General] Screensaver - External temperature**”) so that it the device can receive the required values from the bus.
- **Run button action on screensaver exit:** sets whether touching a button while the screensaver is active should cause the execution of the button action, apart from hiding the screensaver.

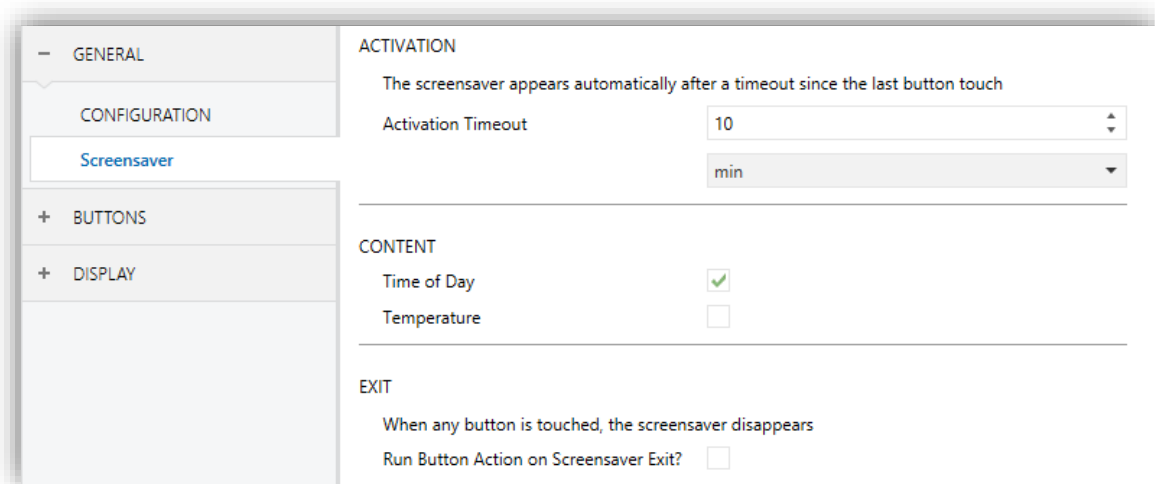


Figure 6 General – Screensaver.

## 2.2.4 LEDs & DISPLAY

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 and the display 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.

**Note:** a general parameter is provided to make the ETS parameterisation easier in case the night mode is not required and the user feels comfortable with the default on/off brightness levels.

Regarding the display, it is possible to parameterise a contrast level in parameters. The integrator can also enable an object to change the display contrast in real time.

## ETS PARAMETERISATION

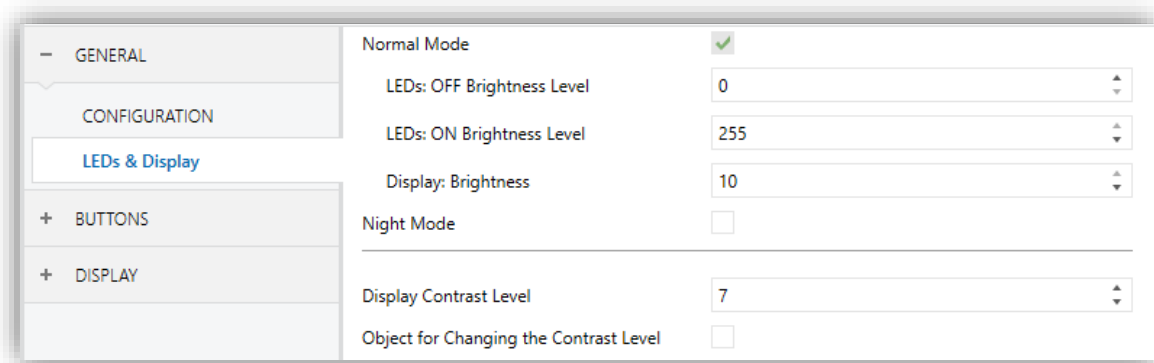


Figure 7 General – LEDs & Display configuration

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

The parameters shown in Figure 7 are equivalent to those for the “Default” configuration of parameter “**LEDs and Display config.**”.

The upper section of this new screen is for the LED brightness configuration while the lower is for the display brightness configuration.

### LED Brightness Levels

- **Normal Mode:**
  - **LEDs: OFF level brightness:** values between 0 (default) and 255.
  - **LEDs: ON level brightness:** values between 0 and 255 (default).
  - **Display: Brightness:** values between 0 and 10 (default).
- **Night Mode:** in case of being this mode necessary, this checkbox needs to be marked, which will incorporate some new parameters:
  - **LEDs: OFF level brightness:** values between 0 and 255 (1 by default).



- **LEDs: ON level brightness:** values between 0 and 255 (8 by default).
- **Display: Brightness:** values between 0 and 10 (default).

Normal Mode	<input checked="" type="checkbox"/>
LEDs: OFF Brightness Level	0
LEDs: ON Brightness Level	255
Display: Brightness	10
Night Mode	<input checked="" type="checkbox"/>
LEDs: OFF level brightness	1
LEDs: ON level brightness	8
Display: Brightness	10
Brightness Mode Control:	
Control Object: 1-Bit	<input type="checkbox"/>
Control Object: Scene	<input type="checkbox"/>
Idle Time to Change to Night Mode (0 = disabled)	0 x 1 s
Brightness Mode After ETS Download	<input checked="" type="radio"/> Normal <input type="radio"/> Night

Figure 8 General – LEDs & Display – LED Brightness.

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.
- **Idle time to change to Night Mode (0=disabled):** when the time set in this parameter elapses, Square TMD-Display will switch to Night Mode (provided that it was not already in this mode). The range is 0 to 65535 seconds (0 by default).

- **Brightness Mode after ETS download:** sets which of the two modes (“Normal” or “Night”) will be active after an ETS Download.

## Display

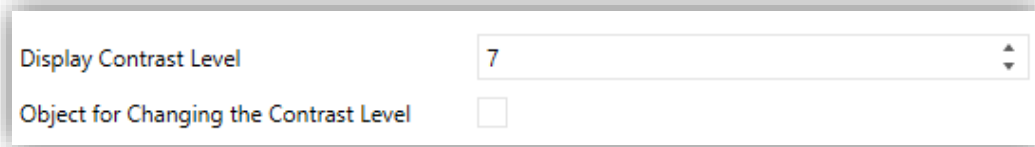


Figure 9 General – LEDs & Display – Display configuration.

- **Display contrast level:** value to set the display contrast level, from 1 to 10 (7 by default).
- **Object for changing the contrast level:** enables a one-byte object (“[General] Display - Contrast”) to change the display contrast level in real time.

### 2.2.5 SOUNDS

Apart from the behaviour of the LEDs, it is possible to configure the Square TMD-Display 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-Display 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.

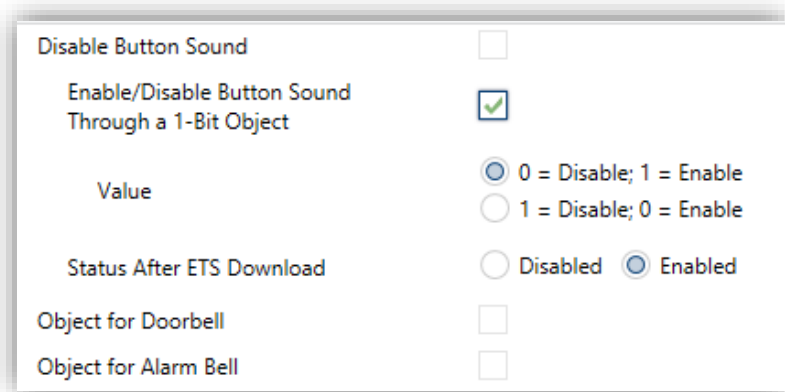


Figure 10 General – Sounds.

- **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”.

### 2.2.6 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 11) contains the following parameters:

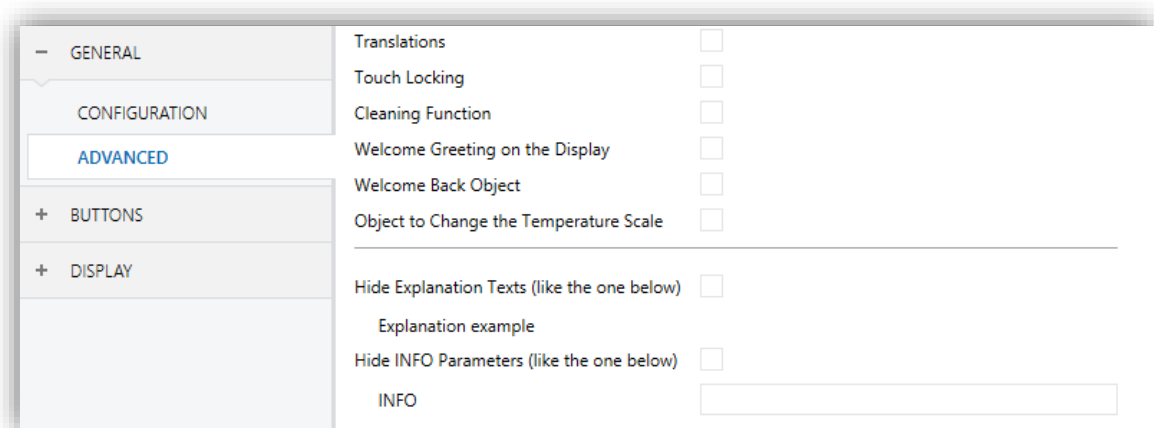


Figure 11 General - Advanced.

- **Translations:** enables or disables the “Translations” tab in the tree on the left. See section 2.2.7 for details.
- **Touch locking:** enables or disables the “Touch locking” tab in the tree on the left. See section 2.2.8 for details.
- **Cleaning function:** enables or disables the “Cleaning function” tab in the tree on the left. See section 2.2.9 for details.
- **Welcome greeting on the display:** enables or disables the “Welcome greeting” tab in the tree on the left. See section 2.2.10 for details.
- **Welcome back object:** enables or disables the “Welcome back” tab in the tree on the left. See section 2.2.11 for details.
- **Object to Change the Temperature Scale:** enables or disables the temperature scale selection object (“**[General] Temperature Scale**”), which permits changing in runtime the scale of the temperatures that may show on the screen. By receiving one ‘0’ through this object, the scale will switch to Celsius, while after receiving one ‘1’ it will switch to Fahrenheit.

The selected scale applies to any temperatures shown on the screen, such as:

- The **screensaver** temperatura.
- Indicators of **temperature controls** linked to a box in the display ([Climate] Temperature setpoint)
- **Temperature indicators** ([Climate] Temperature).

After enabling this functionality, the following parameter will also appear:

- **Scale After Programming:** sets the scale in use after download. The options are: “Celsius (°C)” (default option) and “Fahrenheit (°F)”

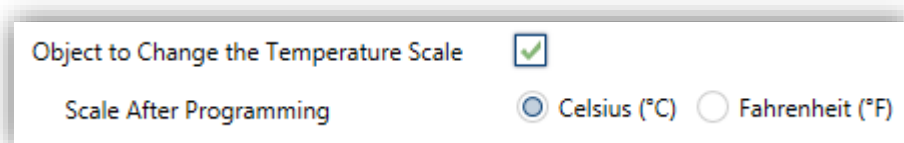


Figure 12 Temperature scale selection.

**Important:** selecting the Fahrenheit scale entails an internal conversion of the values shown on the screen. However, any parameters in ETS related to temperatures should still be configured in the Celsius scale. Objects related to temperatures should also send values in the Celsius scale, according to the KNX standard.

- **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.

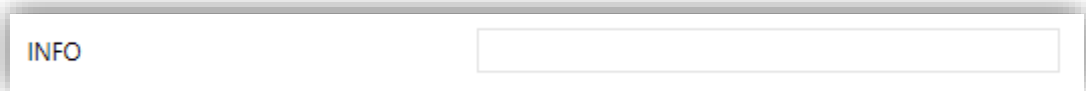


Figure 13 “Info” Textboxes.

## 2.2.7 TRANSLATIONS

---

The labels shown by the boxes on the screen, as well as the texts of the enumerated controls, can be translated into up to five different languages. Switching from language to another can be done through three types of communication objects:

- **Up to five one-bit objects**, one for each language. When receiving a “1” through one of them, the corresponding language is activated, if enabled (otherwise the main language is activated).
- **A one-byte object.** When, through this object, values from 0 to 4 are received, the language will be changed to the parameterized one.
- **A two-byte ASCII object.** The values expected by this object are two ASCII code characters corresponding to the ISO 639-1 standard. When receiving the value of a non-enabled language, the main language is selected.

**Note:** please refer to [http://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes) for a table with the language codes.

## ETS PARAMETERISATION

After enabling “Translations” in the “Advanced” configuration screen (see section 2.2.6), a new tab will be incorporated into the tree on the left.

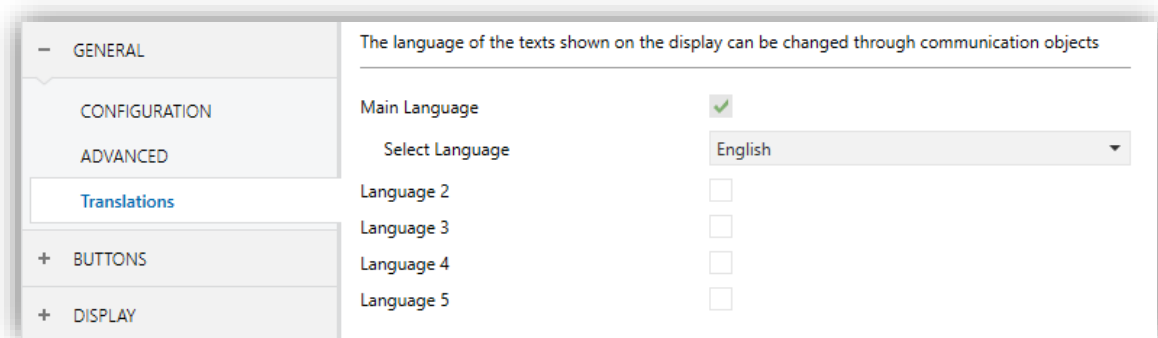


Figure 14 General – Translations.

This screen contains the following parameters:

- **Main language:** read-only parameter to make it evident that the main language is always enabled.
  - **Select language:** list of the available languages.
- **Language X:** enables or disables the additional language X.
  - **Select language:** list of available languages to select the language X.

Only the Main language is enabled by default.

While Translations stays enabled, the following objects are visible:

- “[General] Translations – Select language” (one-byte).
- “[General] Translations – Main language” (one-bit).
- “[General] Translations – Select language” (two-byte).

Up to four specific objects for the additional languages will be also shown, if required:

- “[General] Translations – Language X” (one bit).

These objects work accordingly to the behaviour mentioned above.

## 2.2.8 TOUCH LOCKING

The touch panel of Square TMD-Display 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, user presses on the touch buttons will be ignored: no actions will be performed (and no LEDs will change their states) when the user presses on any of the controls. However, if configured, a message will be shown on the display for three seconds if the user touches a button during the lock state. Furthermore, being in the lock state in the night mode, after pressing on a control the device will switch into normal mode (at least during the idle time configured to change back to night mode; see section 2.2.4), even if the control action is not performed in any case.

### ETS PARAMETERISATION

After enabling “Touch Locking” (see section 2.2.6), a new tab will be incorporated into the tree on the left.

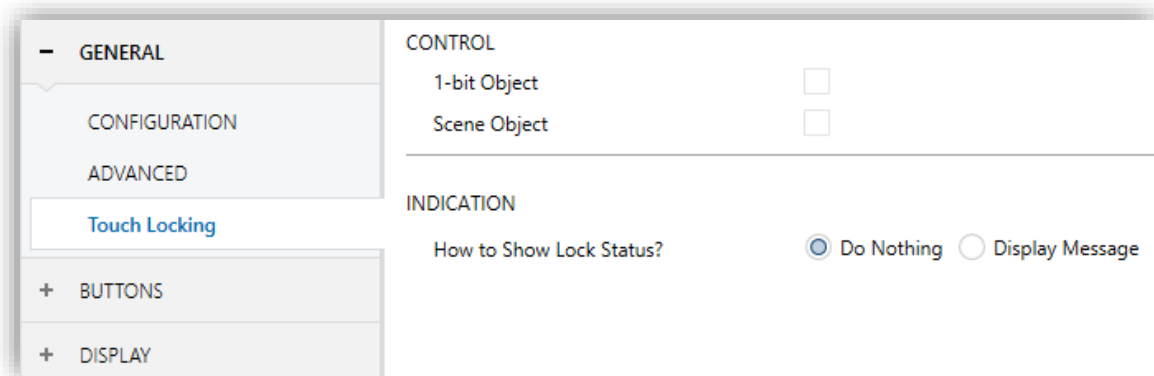


Figure 15 General - Touch Locking.

This screen is divided into two parts: **Control** and **Indication**.

**Control** 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  0 = Lock; 1 = Unlock

Scene Object

Lock: Scene Number

Unlock: Scene Number

Figure 16 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.

**Indication** comprises a parameter to select how to show the lock status:

- **How to show lock status?** Sets whether to display a message on the screen when the device is locked and the user attempts to touch a button (“Display message”) or not (“Do nothing”).

When the first one is selected, one **Message** textbox appears per enabled language (see section 2.2.7) to enter the desired message.

INDICATION

How to Show Lock Status?  Do Nothing  Display Message

Message

Figure 17 General - Touch Locking – Indication.

## 2.2.9 CLEANING FUNCTION

This feature is very similar to the touch locking, that is, it locks the touch area, thus discarding further button touches. The difference is that this function remains active only during a parameterisable time, and then stops.

This function is intended to let the user clean the touch area with the certainty of not triggering unwanted actions.

A message can be shown during the cleaning state. When the timeout is about to end, it is also possible to make this message blink or to make the device beep (or both).

### ETS PARAMETERISATION

After enabling “**Cleaning function**” (see section 2.2.9) a new tab will be incorporated into the tree on the left.

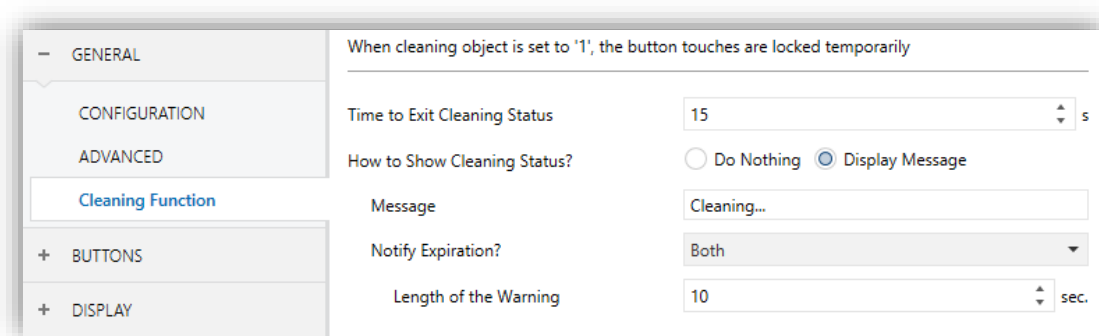


Figure 18 General – Cleaning function.

This screen contains the following parameters:

- **Time to exit cleaning status:** timeout to deactivate the cleaning function, once triggered. Range: from 5 to 65535 seconds (15 by default).
- **How to show cleaning status:** sets whether to show a message during the cleaning state. The options are: “Do nothing” and “Display message”.

When the latter is selected the following parameters appear:

- **Message:** textbox to enter the desired message. Note that one Message textbox will appear per enabled language (see section 2.2.7).

- **Notify expiration?:** sets whether to notify the timeout expiration or not. The options are: “No”, “Blink message”, “Play sound” and “Both”. When any of the three later options is selected, a new parameter shows up:
  - **Timer expiry warning time:** sets the ahead-time (5 to 65535 seconds; 10 by default) to start the notification prior to the end of the cleaning function.

The “[**General**] **Cleaning Function**” one-bit object triggers the cleaning function when it receives a “1” from the KNX bus.

### 2.2.10 WELCOME GREETING

This function permits showing the user a welcome message of up to four lines of text on the display, each of which can be object-dependant or set in parameters.

When a “1” is received through the welcome greeting one-bit object, the display will become blank and show the welcome text. The same will happen if any of the 14-byte objects that define the text lines receives a new value from the bus.

It is also possible to make the button LEDs flash during the welcome state.

#### ETS PARAMETERISATION

After enabling “**Welcome greeting**” (section 2.2.6), a new tab will be incorporated into the tab tree. A one-bit object named “[**General**] **Welcome Greeting**” will also be shown in the project topology, to trigger the welcome message by sending the value “1”.

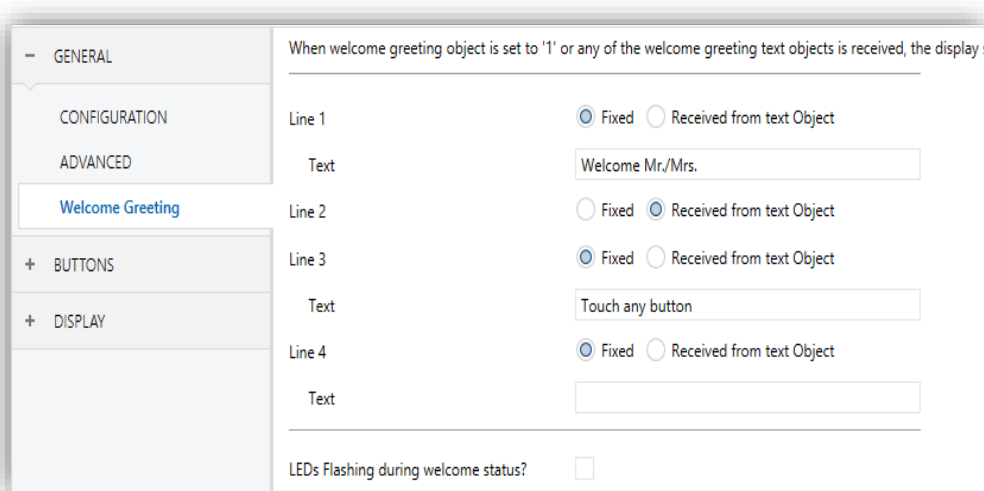


Figure 19 General – Welcome Greeting.

This screen (Figure 19) contains the following parameters:

- **Line X:** sets whether the corresponding text line will be pre-defined (“Fixed”) or object-dependent (“Received from text object”).

If “Fixed” is selected, the following parameter will appear:

- **Text:** textbox to enter the desired text for the corresponding line.

The 14-byte object “[**General**] **Welcome Greeting – Line X**” will be shown up to four times, depending on how many text lines have been assigned the option “Received from text object”.

- **LEDs Flashing During Welcome Status:** sets whether the LEDs of the buttons should blink or not during the welcome greeting.

With a parameterisation as the one shown in Figure 19, if “Mr. Marshall” is received through the “[**General**] **Welcome Greeting – Line 2**” object, the display will show the following message:

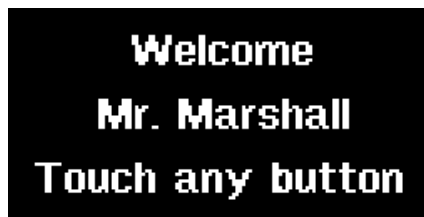


Figure 20 Welcome greeting message example.

### 2.2.11 WELCOME BACK

---

Square TMD-Display 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 the welcome back object to be 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 2.2.6), a new tab will be incorporated into the tree on the left.

This screen contains the following parameters:

Timeout (since last touch)	1
	h
Additional Condition	No additional condition
Welcome Back Object (1-Bit)	<input type="checkbox"/>
Welcome Back Object (Scene)	<input type="checkbox"/>

Figure 21 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)	<input checked="" type="checkbox"/>
Value	<input type="radio"/> Send 0 <input checked="" type="radio"/> Send 1
Welcome Back Object (Scene)	<input checked="" type="checkbox"/>
Scene Number	1

Figure 22 General - Welcome Back Object - Custom Configuration.

## 2.3 BUTTONS

---

### 2.3.1 CONFIGURATION

---

Square TMD-Display features **five capacitive buttons** 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.

Although the five buttons are laid out together in a single row, they can act as single-button controls or in up to two pairs, being possible to **combine any two of them** within one pair.

All the buttons are identical; this brings a high level of versatility for a wide variety of applications. 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, or Pair B**, being the function of such pair one of the following:
  - Switch (binary).
  - Two objects (short press / long press).
  - Light dimmer.
  - Shutter.
  - Scaling.
  - Counter.
  - Enumeration.
  - Float.
  - [Climate] Temperature setpoint.
  - [Climate] Mode.
  - [Climate] Fan.
- **Individual:**
  - LED indicator.
  - Switch (Binary).
  - Hold & release.
  - Two objects (short press / long press).
  - Scene.
  - Scaling constant.
  - Counter constant.
  - Float constant.

- Light dimmer.
- Shutter.
- Enumeration.
- [Climate] Fan.

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



Figure 23 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 (1 to 5).

- Disabled. See section 2.3.2.
- Pair A and Pair B. Sets that this touch button will belong to a two-button control (either A or B). Once one pair has been assigned to two buttons (and not before), a new tab (“**Pair A**” or “**Pair B**”) 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).

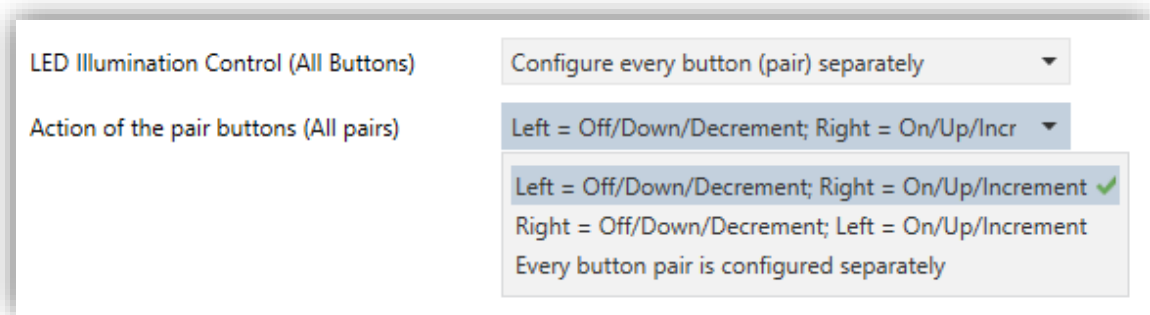


Figure 24 Buttons – Configuration – Advanced options.

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, unless it is configured as “Enumeration” or “[Climate] Mode”.

### 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-Display 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-Display 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.

- **Scaling, Counter and Float:** user touches over the buttons will trigger the sending of a certain numerical value to the bus (this value will be progressively increased or decreased every time the user touches one button or the other) through the control object.

These controls provide a parameter to **link them to a box in the display** so the current numerical value is shown on the display. The screen will reflect the new values as the user touches the buttons, however values are actually sent to the bus 1.5 seconds after the last press.

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.

- **Enumeration:** pressing the buttons will make Square TMD-Display send a one-byte (unsigned) value to the bus. It is possible to distinguish up to 6 different values. Successively pressing on the buttons will make Square TMD-Display walk through the different values.

As the above functions, this type of control provides parameter to **link the control with a box in the display** in order to show the current value on the display. Again, values are sent to the bus 1.5 seconds after the last press.

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.

- **[Climate] Temperature setpoint:** permits controlling the temperature setpoint of an external thermostat by means of a button pair and the display.

This function can also be **linked to a box in the display** in order to show the current setpoint value. The box will reflect the value (in °C or °F, depending on the selected scale; see section 2.2.6) of the status object, which gets automatically updated after sending control orders (that is, after pressing the buttons), being even possible to receive values from the bus, for example, from the corresponding setpoint status object from the external thermostat.

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.

- **[Climate] Mode:** pressing one of the two buttons will make Square TMD-Display send climate mode values to the bus. Two mode control types are possible: **Heat / Cool** and **Extended** (HVAC).

As above, it is possible to **link the control to a box in the display** in order to show the current mode on it.

The “**state-dependent**” and “**state-dependent (both LEDs)**” LED illumination modes are not available for the Extended mode (only the “**regular**” and “**dedicated object**” LED illumination are available). For the Heat / Cool mode all the illumination options are available. See ANNEX I: LED Illumination Modes for details.

- **[Climate] Fan:** pressing one of the two buttons will make Square TMD-Display send a binary value to the bus that allows changing the fan speed (up to **five levels** available). This value can be either **binary** (increase / decrease orders), **scaling** (0% - 100%) or **enumerated**.

Furthermore, it is possible to configure the **speed 0** as an additional fan level, and/or another level to activate the **Auto mode**.

The order to activate the auto mode can be sent over a one-bit object independent from the fan control object, or alternatively consist in the speed level 0. In the latter case, one ‘0’ will trigger the Auto mode.

The three control types available are:

- 1 bit (decrease/increase): one-bit values (increase or decrease) will be sent according to the action configured for each button.

In this case, switching from the current fan level to a non-consecutive level involves going through each of the intermediate levels.

- Scaling: button presses will make the device send percentage values through the corresponding one-byte object.

Every press will activate the subsequent (or preceding) fan level, depending on the current one. Therefore, the percentage values sent will depend on the number of the available levels, and on whether the Auto mode and the speed level 0 are available or not.

- Enumeration: this control is similar to the previous one, but integer values are sent instead of scaling values.

The table below shows the relation between the scaling values and the enumeration values, depending on the number of speed levels configured

(note that if the Auto mode or the speed level 0 are allowed one more level is added, corresponding to value 0).

% Values based on fan levels						
Speed Levels		1	2	3	4	5
Enumeration value	1	100	50,2	33,3	25,1	20
	2	-	100	66,7	50,2	40
	3	-	-	100	75,3	60
	4	-	-	-	100	80
	5	-	-	-	-	100

Table 1 Fan levels.

For the scaling and enumeration controls, the object is sent to the bus slightly delayed, once the desired value has been set by the user. For these two cases it is possible to **link the control with a box in the display** in order to show the current value on the display.

Under a “**state-dependent**” LED illumination, the LED of the corresponding button will remain on/off according to the current state of the fan. On the other hand, under a “**state-dependent (both LEDs)**” LED illumination, both of them will remain on or off while the fan level is greater than 0 or not, respectively. The “**regular**” and “**dedicated object**” LED illumination modes are also available. See ANNEX I: LED Illumination Modes for details.

## ETS PARAMETERISATION

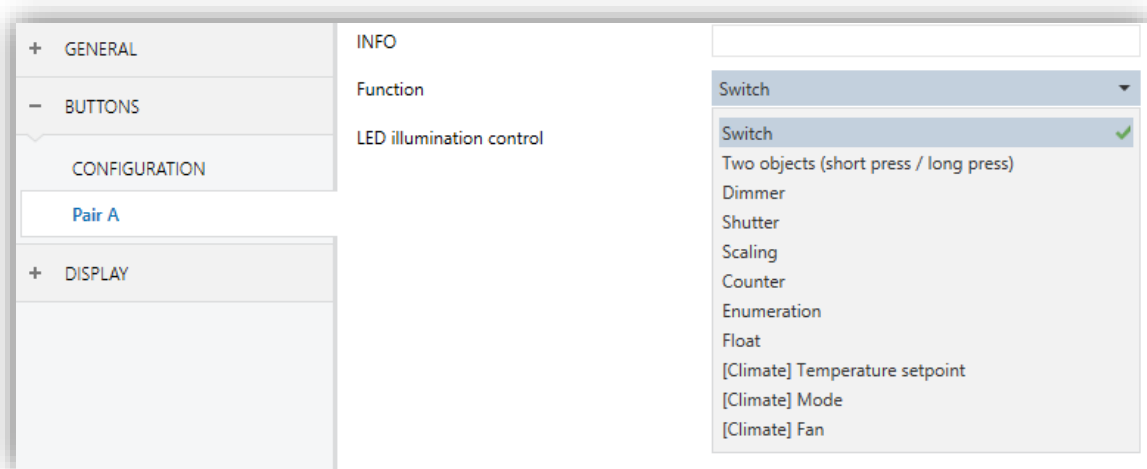


Figure 25 Buttons - Pair A.

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

**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 2.2.6.

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”, “Shutter”, “Scaling”, “Counter”, “Enumeration”, “Float”, “[Climate] Temperature Setpoint”, “[Climate] Mode” or “[Climate] Fan”.

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).

## Switch

Function	Switch
Action	<input checked="" type="radio"/> Left=0; Right=1 <input type="radio"/> Left=1; Right=0
LED illumination control	State-dependent

Figure 26 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:

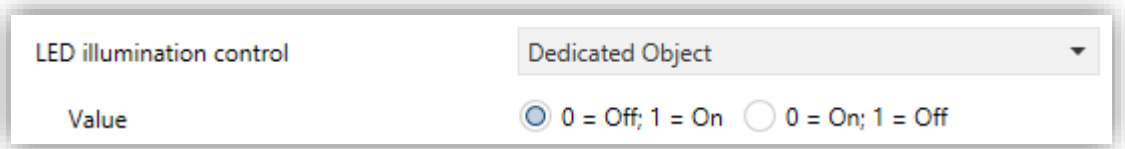


Figure 27 LED illumination control - Dedicated object.

**Note:** this parameter will remain hidden unless having selected “Configure every button (pair) separately” in “**LED Illumination Control (All Buttons)**” (see 2.1).

### Two Objects (short Press / long press)

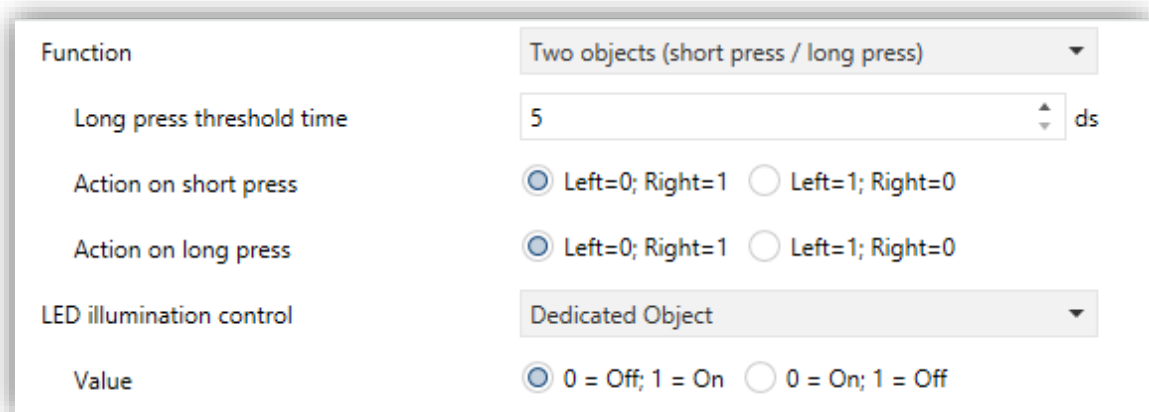


Figure 28 Pair Buttons - Two Objects (Short Press / Long Press).

- **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.
  - “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).



- **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”).

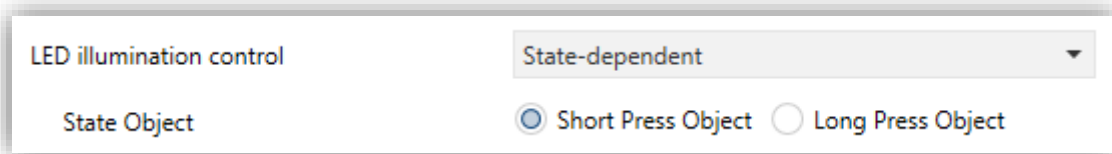


Figure 29 Pair Buttons - Two objects (short press / long press) - LED illumination control.

### Dimmer

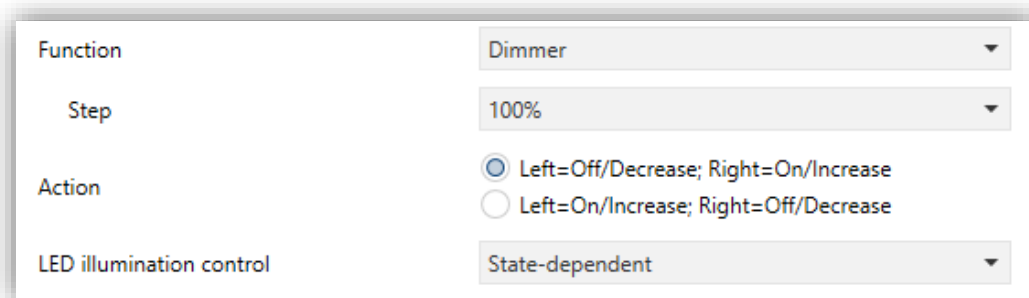


Figure 30 Pair Buttons – Dimmer.

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-Display 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	<input checked="" type="radio"/> Standard <input type="radio"/> Hold & Release
Action	<input checked="" type="radio"/> Left=Down; Right=Up <input type="radio"/> Left=Up; Right=Down
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 31 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”.

### Scaling

Function	Scaling
Minimum Value	0 %
Maximum Value	100 %
Increment on Short Press	1 %
Increment on Long Press	10 %
Action	<input checked="" type="radio"/> Left=Decrease; Right=Increase <input type="radio"/> Left=Increase; Right=Decrease
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 32 Pair Buttons – Scaling.

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

- “Left=Decrease; Right=Increase”.
- “Left=Increase; Right=Decrease”.

**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).*

- **Minimum Value:** sets which value from the available range will be the minimum value permitted by the control. The available range is 0-100.
- **Maximum Value:** sets which value from the available range will be the maximum value permitted by the control after. The available range is 0-100.
- **Increment on Short Press:** sets the increase or decrease to be applied to the current value on every short press over the increase or decrease buttons, respectively (i.e., the smaller the increment, the more presses will be required). The available range is 0-100.
- **Increment on Long Press:** sets the increase or decrease to be applied to the current value on every long press over the increase or decrease buttons, respectively (i.e., the smaller the increment, the more presses will be required). The available range is 0-100.
- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. The selected box must be configured with the same data type (see section 2.4 for details).
- **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”.

The percentage values will be sent to the bus through object “[Btn] [PX] Scaling”.

## Counter

Function	Counter
Size	<input checked="" type="radio"/> 1-byte <input type="radio"/> 2-bytes
Signed?	<input type="checkbox"/>
Minimum Value	0
Maximum Value	255
Increment on Short Press	1
Increment on Long Press	10
Action	<input checked="" type="radio"/> Left=Decrease; Right=Increase <input type="radio"/> Left=Increase; Right=Decrease
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 33 Pair Buttons – Counter.

- **Action:** assigns each of the two buttons the order to be sent:
  - “Left=Decrease; Right=Increase”.
  - “Left=Increase; Right=Decrease”.

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

- **Size:** sets the desired size of the object, “1 byte” or “2 bytes”.
- **Signed:** checkbox to set whether the object will be signed or unsigned.

The value ranges depend on the above two parameters:

- 1 byte unsigned: 0 – 255.
  - 1 byte signed: -128 – 127.
  - 2 bytes unsigned: 0 – 65535.
  - 2 bytes signed: -32768 – 32767.
- **Minimum Value:** sets which value from the available range will be the minimum value permitted by the control.

- **Maximum Value:** sets which value from the available range will be the maximum value permitted by the control.
- **Increment on Short Press:** sets the increase or decrease to be applied to the current value on every short press over the increase or decrease buttons, respectively (i.e., the smaller the increment, the more presses will be required).
- **Increment on Long Press:** sets the increase or decrease to be applied to the current value on every long press over the increase or decrease buttons, respectively (i.e., the smaller the increment, the more presses will be required).
- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. See section 2.4 for details.
- **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”.

The counter values will be sent to the bus through object “[Btn] [PX] Counter [size]”.

### Enumeration

Function	Enumeration
# Enums	1
Value 1	0
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 34 Pair Buttons – Enumeration.

- **# Enums:** number of states (up to 6) that will be distinguished. For every distinguished state, one parameter named **Value j** will become available, in order to set the value sent through the communication object.

- **Value  $j$ :** numerical value (0 – 255) that will be sent through the control object when the user, after pressing the buttons, sets the control to state  $j$ . The actual sending will not be performed until 1.5 seconds after the last press.
- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. See section 2.4 for details.
- **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”.

The values will be sent to the bus through object “[Btn] [PX] Enumeration”.

## Float

Function	Float
Minimum Value	-671088.625
Maximum Value	670760.9375
Increment on Short Press	0.5
Increment on Long Press	1
Action	<input checked="" type="radio"/> Left=Decrease; Right=Increase <input type="radio"/> Left=Increase; Right=Decrease
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 35 Pair Buttons – Float.

- **Action:** assigns each of the two buttons the order to be sent:
  - “Left=Decrease; Right=Increase”.
  - “Left=Increase; Right=Decrease”.

**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).*

- **Minimum Value:** sets which value from the available range (-671088.625 to 670760.9375) will be the minimum value permitted by the control.
- **Maximum Value:** sets which value from the available range (-671088.625 to 670760.9375) will be the maximum value permitted by the control.
- **Increment on Short Press:** sets the increase or decrease to be applied to the current value on every short press over the increase or decrease buttons, respectively (i.e., the smaller the increment, the more presses will be required). The available range is 0.1 – 670760.93750 (0.5 by default).
- **Increment on Long Press:** sets the increase or decrease to be applied to the current value on every long press over the increase or decrease buttons, respectively (i.e., the smaller the increment, the more presses will be required). The available range is 0.1 – 670760.93750 (1 by default).
- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. The selected box must be configured with the same data type (see section 2.4 for details).
- **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”.

The values will be sent to the bus through object “[Btn] [PX] Float”.

### [Climate] Temperature Setpoint

Function	[Climate] Temperature setpoint
Minimum Value	10 °C
Maximum Value	30 °C
Action	<input checked="" type="radio"/> Left=Decrease; Right=Increase <input type="radio"/> Left=Increase; Right=Decrease
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 36 Pair Buttons – [Climate] Temperature setpoint.



- **Action:** assigns each of the two buttons the order to be sent:
  - “Left=Decrease; Right=Increase”.
  - “Left=Increase; Right=Decrease”.

**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).*
- **Minimum Value:** sets which value from the available range (-671088.625 to 670760.9375; 10 by default) will be the minimum value permitted by the control.
- **Maximum Value:** sets which value from the available range (-671088.625 to 670760.9375; 30 by default) will be the maximum value permitted by the control after.
- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. The selected box must be configured with the same data type (see section 2.4 for details).
- **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”.

The values are sent to the bus through **“[Btn] [PX] (Climate) Temperature Setpoint”**.

### [Climate] Mode

Function	[Climate] Mode
Mode Type	<input checked="" type="radio"/> Heat/Cool <input type="radio"/> Extended
LED illumination control	State-dependent

Figure 37 Pair Buttons – [Climate] Mode.

- **Mode Type:** sets mode control type: “Heat/Cool” or “Extended” (HVAC).
  - “Heat/Cool”: pressing one of the two buttons will make Square TMD-Display send a one-bit value to the bus (Left=Cool, value “0”; Right=Heat, value “1”).

- **“Extended”**: pressing on the two buttons will make Square TMD-Display walk through the different HVAC modes and send a one-byte value to the bus according to the selected mode. Which of the modes should be available can be set through five specific checkboxes.

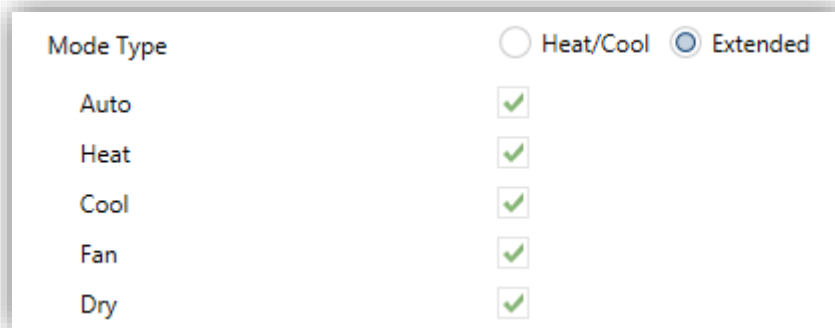


Figure 38 Pair Buttons – [Climate] Mode – Mode Type: Extended.

- **Link the control with a box in the display**: sets the area of the display where to show the indicator of this control, i.e., the current numeric value. The selected box must be configured with the same data type (see section 2.4 for details).

**Note:** *this option is only available for Extended mode type.*

- **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”.

The values will be sent to the bus through object **“[Btn] [PX] (Climate) Mode”**.

### [Climate] Fan

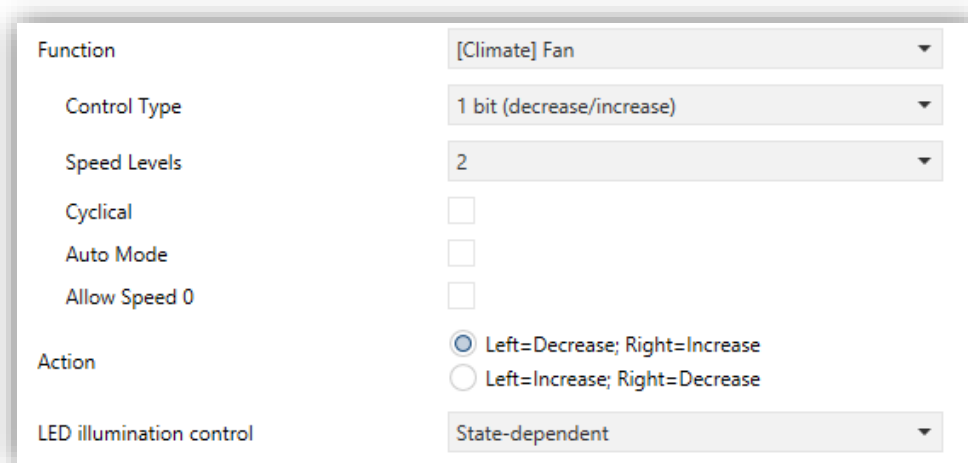


Figure 39 Pair Buttons – [Climate] Fan.

- **Control Type:** sets the type of the communications objects to control the fan level.
  - “1 bit (decrease/increase)”: orders of speed increase/decrease are sent through the one-bit object “[Btn] [PX] Fan control (1 bit)”.
  - “Scaling”: scaling values are sent through the one-byte object “[Btn] [PX] Fan control (scaling)”.
  - “Enumeration”: integer values are sent through the one-byte object “[Btn] [PX] Fan control enumerated”.
  
- **Speed levels:** sets how many speed levels will be available in the control. Between 1 and 5 levels.
  
- **Cyclical:** sets whether scrolling through the speed levels is circular or not. If checked, an increase order in the maximum level switches to the minimum and vice versa (in the minimum level a decrease order switches to the maximum).
  
- **Auto Mode:** sets whether the fan Auto mode will be available or not. If checked, the following parameters are also shown.
  - **Dedicated Object for Auto Mode:** sets how the Auto mode should be activated. In the absence of a dedicated object, the Auto fan mode will be activated by switching to fan level 0.

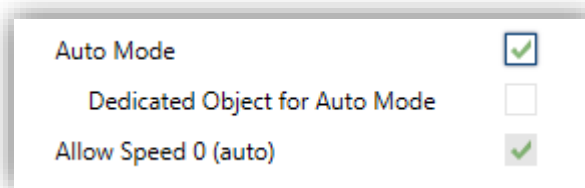


Figure 40 Pair Buttons – [Climate] Fan – Auto Mode.

In this case (supposing that Speed Levels has been set, for instance, to “3”), the fan levels that can be navigated through short presses are:

Auto Mode ( 0 )	Minimum	Medium	Maximum
-----------------	---------	--------	---------

On the other hand, marking the checkbox enables the one-bit object “[Btn] [PX] Fan control – Auto mode”, which will trigger the Auto mode when it receives the value “1” (or “0”, depending on the subsequent parameter, “Value to Set the Auto Mode”).

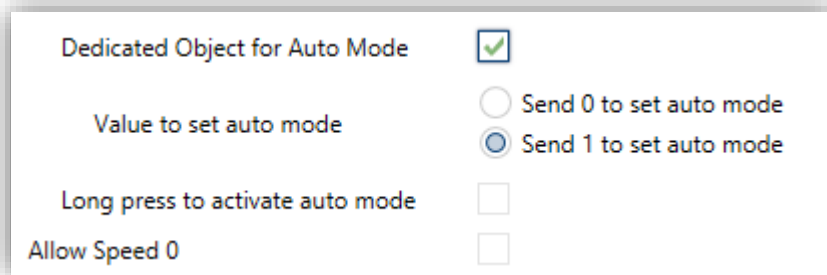


Figure 41 Pair Buttons – [Climate] Fan – Dedicated Object for Auto Mode.

However, two alternative (and mutually exclusive) methods are possible to activate this mode:

- By short press: the Auto mode can be reached as a further level above the maximum one. In this case the fan levels activated by short presses are (note that **speed 0 is optional**):

( 0 )	Minimum	Medium	Maximum	Auto Mode
-------	---------	--------	---------	-----------

- By long press on any of the buttons of the control (requires checking **Long Press to Activate Auto Mode**).

A further long press deactivates back the Auto mode and resumes the previous fan level. Instead, a short press deactivates the Auto mode and switches to the next level (or previous, depending on the button). In this case the fan levels reachable through short presses are (**speed 0 is optional**):

( 0 )	Minimum	Medium	Maximum
-------	---------	--------	---------

(The Auto mode here is only activated by long press).

- **Allow speed 0**: sets whether the speed level 0 will be present or not. When the **Auto Mode without a dedicated object** has been configured, this option will be necessarily activated.

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

- “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).*

- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current fan speed. The selected box must be configured with the same data type (see section 2.4 for details).

**Note:** *this parameter is only available for Scaling or Enumeration control types.*

- **LED illumination control:** analogous to the homonymous parameter for the previous functions (see above). The options are “Regular”, “State-dependent”, “State-dependent (both LEDs)” 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 to this alternation:

- 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-Display 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 as soon as 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-Display 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.

- **Enumeration:** implements a one-button enumeration control that sends one-byte (unsigned) values to the KNX bus. This control is completely analogous to the Enumeration control in pair buttons (see section 2.3.3), however the object is sent to the bus when the button is pressed (not 1.5 seconds later). Moreover, in this case there is no option to link it to the display.

The “**state-dependent**” LED mode is not available for this function.

- **[Climate] Fan:** implements a fan control which sends the KNX bus values in order to change the fan speed level. This function is similar to the pair button fan control described in section 2.3.3 (however in this case the object is sent to the bus immediately after the button is pressed).

## ETS PARAMETERISATION

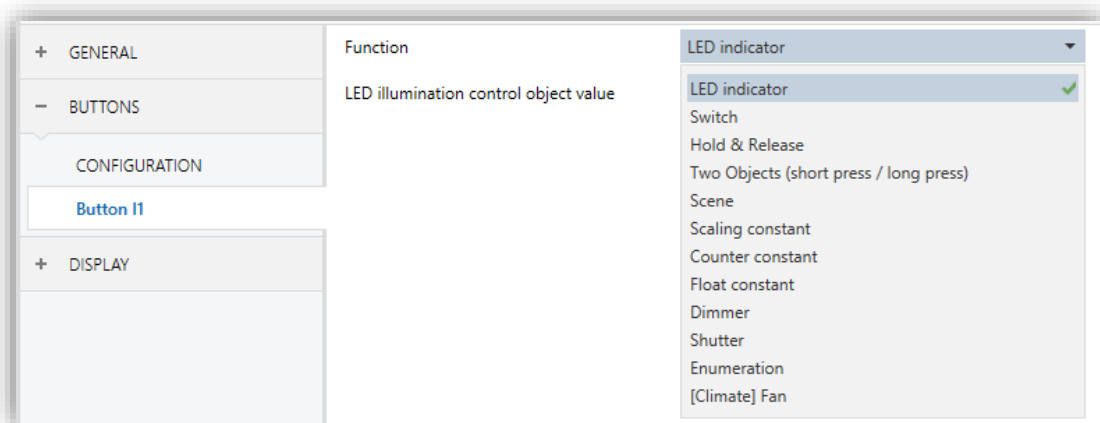


Figure 42 Individual Button.

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

**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 2.2.6.

The main parameter that needs to be configured is:

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

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 (1 through 5).

### LED Indicator

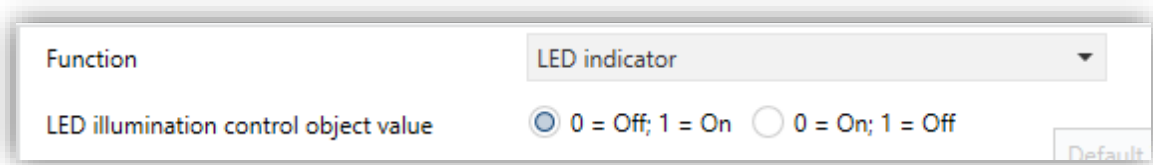


Figure 43 Individual Button - Disabled.

- **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 44 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	<input type="radio"/> Send 0 <input checked="" type="radio"/> Send 1
Action on Release	<input checked="" type="radio"/> Send 0 <input type="radio"/> Send 1
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated object

Figure 45 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 “LED Indicator” 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	<input checked="" type="radio"/> Short Press Object <input type="radio"/> Long Press Object

Figure 46 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 objects - 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 objects - 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 “LED Indicator”, “Switch” 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 objects - Short press” (“Short Press Object”) or to object “[Btn] [In] Two objects - Long press” (“Long Press Object”).

Scene

Function	Scene
Action	<input checked="" type="radio"/> Run scene <input type="radio"/> Run (short press) + Save (3 sec. press) scene
Scene number	1
LED illumination control	<input type="radio"/> Regular <input checked="" type="radio"/> Dedicated object
Value	<input checked="" type="radio"/> 0 = Off; 1 = On <input type="radio"/> 0 = On; 1 = Off

Figure 47 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 48 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
<b>Scaling Constant</b>	0% – 100%	[Btn] [In] Scaling
<b>Counter Constant</b>	0 – 255	[Btn] [In] Counter – 1 byte unsigned
	-128 – 127	[Btn] [In] Counter – 1 byte signed
	0 – 65535	[Btn] [In] Counter – 2 byte unsigned
	-32768 – 32767	[Btn] [In] Counter – 2 byte signed
<b>Float Constant</b>	-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

The screenshot shows a configuration window for a 'Dimmer' function. It contains four rows of settings:

- Function:** A dropdown menu with 'Dimmer' selected.
- Step:** A dropdown menu with '100%' selected.
- LED illumination control:** A dropdown menu with 'Dedicated object' selected.
- Value:** Two radio button options: '0 = Off; 1 = On' (which is selected) and '0 = On; 1 = Off'.

Figure 49 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 2.3.4, 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-Display 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	<input checked="" type="radio"/> Standard <input type="radio"/> Hold & Release
LED illumination control	<input type="radio"/> Regular <input checked="" type="radio"/> Dedicated object
Value	<input checked="" type="radio"/> 0 = Off; 1 = On <input type="radio"/> 0 = On; 1 = Off

Figure 50 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 2.3.4, 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 2.3.4.
- **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”.

## Enumeration

Function	Enumeration
# Enums	1
Value 1	0
LED illumination control	<input type="radio"/> Regular <input checked="" type="radio"/> Dedicated object
Value	<input checked="" type="radio"/> 0 = Off; 1 = On <input type="radio"/> 0 = On; 1 = Off

Figure 51 Individual Buttons – Enumeration.

This control is completely analogous to the Enumeration control in pair-button controls (see section 2.3.3), but without the parameter to link the control to the display (in this case, the value is sent to the bus when the button is pressed).

- **# Enums:** number of states (up to 6) that will be distinguished. For every distinguished state, one parameter **Value *j*** will become available, in order to set the value sent through the communication object.
- **Value *j*:** numerical value (0 – 255) that will be sent through the object “[Btn] [In] Enumeration” when the user, after pressing the buttons, sets the control box to state *j*.
- **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”.

## [Climate] Fan

Function	[Climate] Fan
Control Type	1 bit (decrease/increase)
Speed Levels	2
Sending Type	<input checked="" type="radio"/> Always send 1 <input type="radio"/> Send 0 or 1 according to the state object
Auto Mode	<input type="checkbox"/>
Allow Speed 0	<input type="checkbox"/>
LED illumination control	State-dependent

Figure 52 Individual Buttons – [Climate] Fan.



This control is analogous to the Fan control in pair buttons (see section 2.3.3).

The objects that depending on the configuration will send the orders to the bus are in this case the following:

- “[Btn] [In] Fan Control (1 bit)”
- “[Btn] [In] Fan Control Scaling”
- “[Btn] [In] Fan Control Enumerated”

The status object (“[Btn] [In] Fan - State”) will be percentage-type or enumerated-type depending on the configuration.

In case of opting for a **one-bit** fan control, one special parameter will show up instead of the “**Cyclical**” parameter (which in the single-button control only applies to the **scaling** and **enumerated** fan controls):

- **Sending Type:** “Always Send 1” (after every button press, one “1” will be sent to the bus, i.e., the direction of the speed regulation will always be upwards) or “Send 0 or 1 According to the State Object” (the direction of the fan control will be upwards or downwards depending on the current state, as explained below).

The first option makes sense when the fan actuator implements a **cyclical control** (after the maximum fan level, a further increase order switches to the minimum level).

Being single-button, the **direction (upwards/downwards) of the control** will always be determined by the current value of the status object (except in the aforementioned “Always Send 1” one-bit case):

- **Upwards** if already at the minimum level.
- **Downwards** if already at the maximum level.
- In any other case, the direction will be the same as in the previous press.

Note that depending on how many speed levels have been configured and on whether the Auto mode has been left disabled or has been enabled with a dedicated object or not, the minimum and maximum levels of the range may not always be the same.

## 2.4 DISPLAY

### 2.4.1 CONFIGURATION

Square TMD-Display features a **1.8” 128x64 pixel back-lit monochrome screen** with black background and white icons and fonts.

This screen can show useful information to the user, related to the current state of other devices or of its own controls.

The information provided by the display can consist in icons (with a resolution up to 28 x 22 pixels) or alphanumeric characters (texts). Please refer to the specific document **“Icons in Square TMD-Display”** (available at the Zennio homepage, [www.zennio.com](http://www.zennio.com)) to see the list of available icons.



Figure 53 Some icons from Square TMD-Display.

Three **different font sizes** for the alphanumeric characters are available: **Small**, **Big** and **Extra** (this one only for numeric characters). On the other hand, the built-in character set is:

- **Basic Latin**<sup>1</sup>, in small and big font size.
- **Extended Latin** (“À” to “ÿ”), only in small font size<sup>2</sup>.
- **Cyrilic** alphabet, in small and big font size.
- **Symbols**: € ° & - . ;

As Square TMD-Display offers the user a sole control page (instead of alternating, browsable pages), all the content of the display is shown permanently. It only gets hidden when the screensaver or some other special messages (welcome, locking or cleaning) are superimposed.

<sup>1</sup> According to the Unicode classification.

<sup>2</sup> In case the big size is selected, these characters will be represented in small size.

The screen is divided into 4 zones (**single** boxes –or *big* boxes-), named “a”, “b”, “c” and “d”, each of which can be further divided into two **double** boxes –or small boxes–. Each box consists of two areas:

- Lower area, where the box name (configurable through the parameter **Label**) is represented, with the Small text size. The text length depends on the box size: maximum 12 characters for big boxes and 6 characters for small boxes (if the text does not fit in the box, it will be truncated). If this label is left empty, then the indicator fills the box.
- Upper area, where the **Indicator** is represented. There are up to four representation modes, configurable in parameters for each indicator: **Permanent** (always shown), **Temporary** (shown for a moment when the value changes or after a bus recovery), **Intermittent** (intermittent indicator), and **Progressive** (the icon is progressively drawn from top to bottom).

The boxes in the screen can be assigned different functions, as described next.

### ETS PARAMETERISATION

An independent tab for the parameterisation of the Display is shown in ETS by default. While all the zones remain disabled, it will only contain a sub-tab named Configuration.

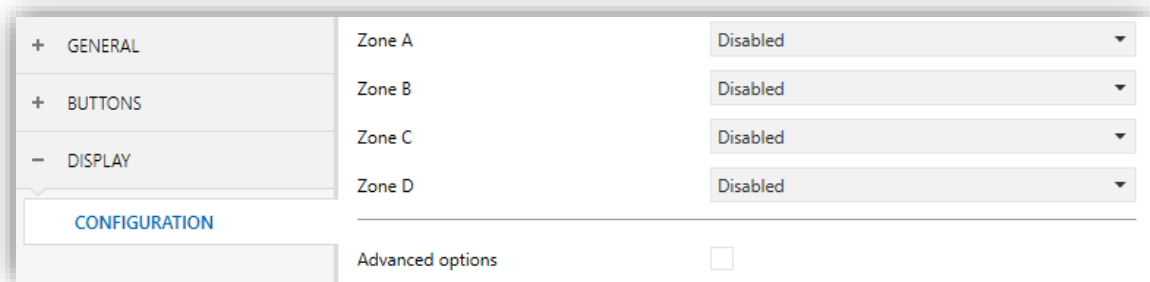


Figure 54 Display – Configuration.

One drop-down list is shown per zone. The options under it are:

- Disabled. See section 2.4.2.
- Single (one big box). Selecting this option brings a new tab to the tree on the left (named “**Box X**”, where X depends on the display zone). See section 2.4.3.

- **Double (two small boxes).** Selecting this brings the checkboxes “**Box X1**” and “**Box X2**” (where X depends on the display zone), which turn each of the two zones enabled or disabled. Depending on this, up to two new tabs (named “**Box Xn**”) will be included in the tree on the left. See section 2.4.4.

There is also a checkbox named **Advanced options**, which permit configuring the border lines and the representation of the indicators.

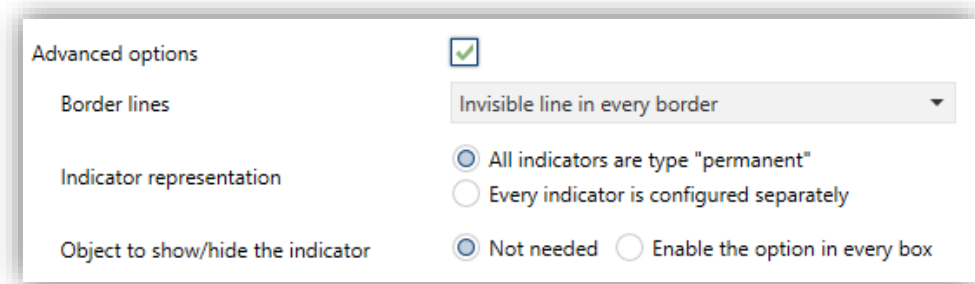


Figure 55 Display - Configuration - Advanced options.

- **Border lines:** lets selecting how the box border lines will be represented:

- Solid line in every border,
- Invisible line in every border,
- Broken line in every border,
- Customize every border.

In case of selecting “Customize every border”, an independent tab appears under Display to configure each box border line (see section 2.4.5).

- **Indicator representation:** defines the representation of the indicators on the display. The available options are:

- All indicators are type “permanent”,
- Every indicator is configured separately.

In case of selecting “Every indicator is configured separately”, there will be a specific parameter for every box to select the desired indicator representation.

- **Object to show/hide the indicator:** shows or hides a parameter in the configuration screen of each box to let the integrator add the project a one-bit object per box to show or hide its indicator in runtime.
  - Not needed,
  - Enable the option in every box.

## 2.4.2 DISABLED

---

While a display zone remains disabled, it does not represent anything and cannot be linked to any button – it will remain empty.

## ETS PARAMETERISATION

---

This function has no related parameters.

## 2.4.3 SINGLE

---

Single boxes have a size of 64 x 32 pixels. The display can be divided into up to four single boxes.

Single boxes can be assigned the following types of indicators:

- **Icon:** the indicator will represent an icon according to the value of an object. Depending on the type of object there are two types of icons:
  - One bit: an icon represents the On state and another one represents the Off state.
  - Enumeration: up to six different states can be represented through one icon each.
- **Number:** the indicator will represent a number (received through the associated object) and a three-character text representing the units. Different number types are available for the different object types: Scaling, Counter (1 byte unsigned), Counter (1 byte signed), Counter (2 bytes unsigned), Counter (2 bytes signed), Counter (4 bytes unsigned), Float (2 bytes).
- **Text:** the indicator will represent a text according to the value of an object. This indicator is totally analogous to the Icon indicator, but each value is associated to a text instead of an icon. The types are the same:

- One bit: a text represents the On state and another one represents the Off state.
- Enumeration: up to six different states can be represented through one text each.
- **Time of day**: represents the current time (“hh:mm” in 24-hour format). Note that the time of day must be set through the bus by an external reference. In addition, in the absence of power the clock will get outdated.
- **Text from object**: represents a string (up to 14 characters) received through the corresponding communication object.
- **[Climate] Temperature**: represents temperature values as received through the related communication object. Values in the range [-99, 199] °C and in the range [-142, 390] °F are available (depending on the selected temperature scale; see section 2.2.6). Values out of range will be replaced by the corresponding limit.

When the value contains decimals, only the first one will be shown, and it will be represented in a smaller size. Also, if the decimal part does not fit, the number will be rounded to the nearest integer.

- **[Climate] Mode**: represents climate modes as received through the associated communication object. There are two Mode Types:
  - Heat/Cool: only the Heat and Cool modes will be represented, according to the value of a one-bit object.
  - Extended: up to five HVAC climate modes will be represented, according to the value received through a specific one-byte object. The icons are represented with a progressive shift in the box, i.e., the first mode is shown at the left and the last at the right, with the exception of the Auto mode, which is represented in the centre of the box.

If the object value does not correspond to a valid mode, no icon will be displayed.

- **[Climate] Fan:** represents fan speeds, according to the value of the corresponding object. There are up to 4 fan speed levels: Off/Auto, Min, Med, Max. The number of levels can be customised, as explained later.

As with climate modes, the icons of the successive levels are represented with a progressive shift, but without hiding the preceding ones.

- **[Climate] On/Off + Mode + Status:** climate status indicator with three one-bit objects associated: On/Off, Mode and Status. The icon represented depends on the value of these three objects, as explained in next section.

### ETS PARAMETERISATION

When a display zone has been configured as Single, a specific tab (“**Box x**”) becomes available under “Display” in the tree on the left.

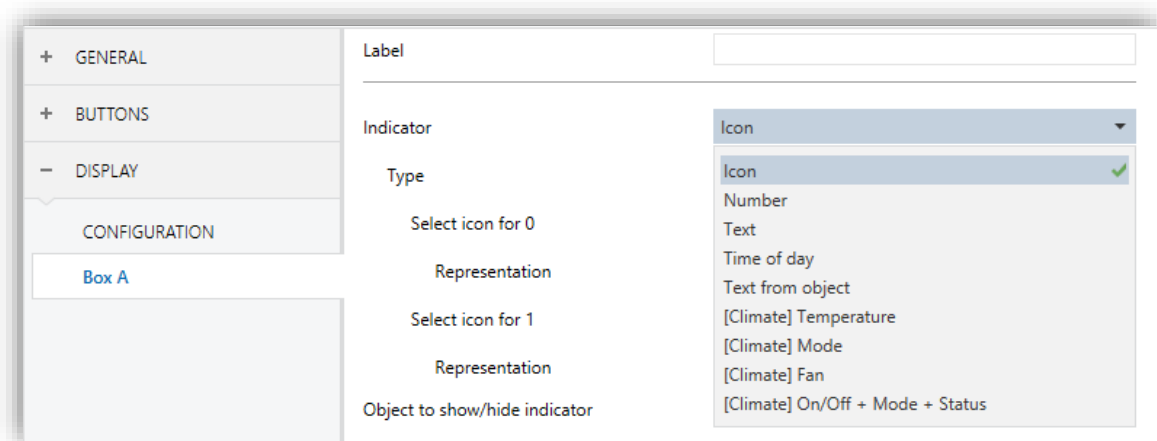


Figure 56 Display – Box a.

The main parameters that need to be configured are:

- **Label:** sets the box name (see section 2.4).
- **Indicator:** sets the desired indicator to display in the box: “Icon”, “Number”, “Text”, “Time of day”, “Text from object”, “[Climate] Temperature”, “[Climate] Mode”, “[Climate] Fan” or “[Climate] On/Off + Mode + Status”.

Depending on the indicator, 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 display zone (a, b, etc.).

## Icon

Figure 57 Display indicator – Icon – 1 bit.

• **Type:**

- “1-bit”: the icon represented depends on the value of the one-bit object “[Disp] [x] Icon – 1 bit”. When selecting this type, the following parameters need to be configured:

- **Select icon for 0/1:** sets the icon to display for each value.
- **Representation:** sets the icon representation. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “**Indicator representation**” (see 2.4.1).*

- “Enumeration”: the icon represented depends on the value of the one-byte object “[Disp] [x] Icon – Enumeration”. When selecting this type, the following parameters need to be configured.

- **How many values:** sets the number of states of the enumerated list. Up to 6 are possible. For each element the following three parameters need to be configured.
  - **Value:** sets the numerical value assigned to the state (0 to 255).
  - **Icon:** sets the icon to be displayed when the above value is received.
  - **Representation:** sets the icon representation. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected*



*“Every indicator is configured separately” in “Indicator representation” (see 2.4.1).*

The screenshot shows a configuration window for an indicator. The settings are as follows:

- Indicator:** Icon
- Type:** Enumeration (selected with a radio button)
- How many values?:** 1
- Value 1:** 0
- Icon:** No icon
- Representation:** Permanent
- Object to show/hide indicator:**

Figure 58 Display indicator – Icon – Enumeration.

- **Object to show/hide indicator:** enables or disables a one-bit object (“[Disp] [x] Show/Hide indicator”) to show or hide the indicator in runtime (1 = Show indicator; 0 = Hide indicator).

**Note:** *this parameter will remain hidden unless having selected “Enable the option in every box” in “Object to show/hide the indicator” (see 2.4.1).*

## Number

The screenshot shows a configuration window for a number indicator. The settings are as follows:

- Indicator:** Number
- Type:** Scaling
- Font size:** Big
- Representation:** Permanent
- Object to show/hide indicator:**

Figure 59 Display indicator – Number – Scaling.

- **Type:** sets the type of the number indicator. The options are: “Scaling”, “Counter (1 byte unsigned)”, “Counter (1 byte signed)”, “Counter (2 bytes unsigned)”, “Counter (2 bytes signed)”, “Counter (4 bytes unsigned)”, “Float (2 bytes)”.
- **Font size:** sets the font size of the number. The options are: “Small”, “Big” and “Extra”.

- **Suffix (units):** sets a label that will be shown next to the indicator. Up to three characters are possible.

**Note:** *this parameter is not available for Scaling, as the suffix is always “%”.*

- **Suffix font size:** sets the font size of the above suffix. The options are: “Small” and “Big”.

**Note:** *this parameter is not available for Scaling, as the suffix font size is always “Small”.*

- **Representation:** sets the indicator representation. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “**Indicator representation**” (see 2.4.1).*

- **Object to show/hide indicator:** enables or disables a one-bit object (“**[Disp] [x] Show/Hide indicator**”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** *this parameter will remain hidden unless having selected “Enable the option in every box” in “**Object to show/hide the indicator**” (see 2.4.1).*

## Text

Indicator	Text
Type	<input checked="" type="radio"/> 1-bit <input type="radio"/> Enumeration
Font size	<input checked="" type="radio"/> Small <input type="radio"/> Big
Value 0:	
Text	<input type="text"/>
Representation	Permanent
Value 1:	
Text	<input type="text"/>
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 60 Display indicator – Text – 1 bit.

• Type:

➤ 1\_bit: the text represented depends on the value of the one-bit object “[Disp] [x] Text – 1 bit”. When selecting this type, the following parameters need to be configured.

- **Value 0/1: Text**: sets the text to be displayed for each value. Maximum 12 characters allowed.
- **Representation**: sets the text representation. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note**: *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “**Indicator representation**” (see 2.4).*

➤ Enumeration: the icon represented depends on the value of the one-byte object “[Disp] [x] Text – Enumeration”. When selecting this type, the following parameters need to be configured.

The screenshot shows a configuration window for a display indicator. The 'Indicator' is set to 'Text'. The 'Type' is set to 'Enumeration' (selected with a radio button). The 'Font size' is set to 'Small' (selected with a radio button). The 'How many elements?' is set to '1'. The 'Value 1' is set to '0'. The 'Text' field is empty. The 'Representation' is set to 'Permanent'. The 'Object to show/hide indicator' checkbox is checked.

Figure 61 Display indicator – Text – Enumeration.

- **How many elements**: sets the number of states of the enumerated list. Up to 6 states are possible. For each state, the following three parameters need to be configured.
  - **Value**: sets the numerical value assigned to the state (0 to 255).
  - **Icon**: sets the text to be displayed when the above value is received.
  - **Representation**: sets the text representation. The options are:

“Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “**Indicator representation**” (see 2.4.1)*

- **Font size:** sets the font size of the texts. The options are: “Small” and “Big”.
- **Object to show/hide indicator:** enables or disables a one-bit object (“**[Disp] [x] Show/Hide indicator**”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** *this parameter will remain hidden unless having selected “Enable the option in every box” in “**Object to show/hide the indicator**” (see 2.4).*

### Time of day

Indicator	Time of day
Font size	Extra
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 62 Display indicator – Time of day.

Once the device starts, the time should be updated by an external clock through the “**[General] Time of day**” three-byte communication object.

- **Font size:** sets the font size of the time label. The options are: “Small”, “Big” and “Extra”.
- **Representation:** sets the representation type. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “**Indicator representation**” (see 2.4.1).*

- **Object to show/hide indicator:** enables or disables a 1 bit object (“**[Disp] [x] Show/Hide indicator**”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** *this parameter will remain hidden unless having selected “Enable the option in every box” in “**Object to show/hide the indicator**” (see 2.4.1).*

**Text from object**

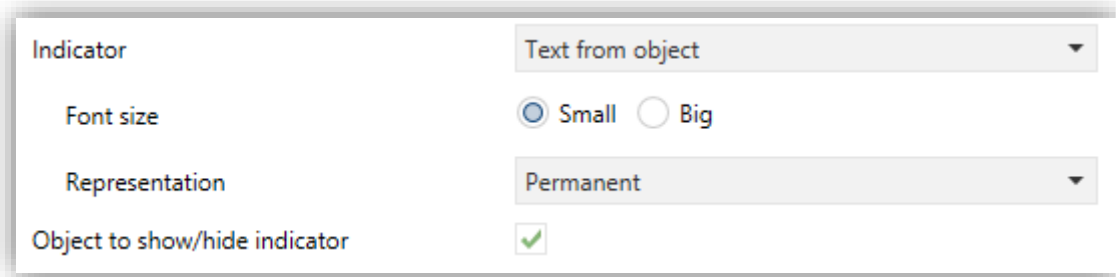


Figure 63 Display indicator – Text from object.

The text to be shown is to be received through the “[Disp] [x] Text from object” 14-byte communication object.

- **Font size:** sets the font size of the text. The options are: “Small” and “Big”.
- **Representation:** sets the text representation. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “**Indicator representation**” (see 2.4.1).*

- **Object to show/hide indicator:** enables or disables a 1 bit object (“[Disp] [x] Show/Hide indicator”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** *this parameter will remain hidden unless having selected “Enable the option in every box” in “**Object to show/hide the indicator**” (see 2.4).*

**[Climate] Temperature**

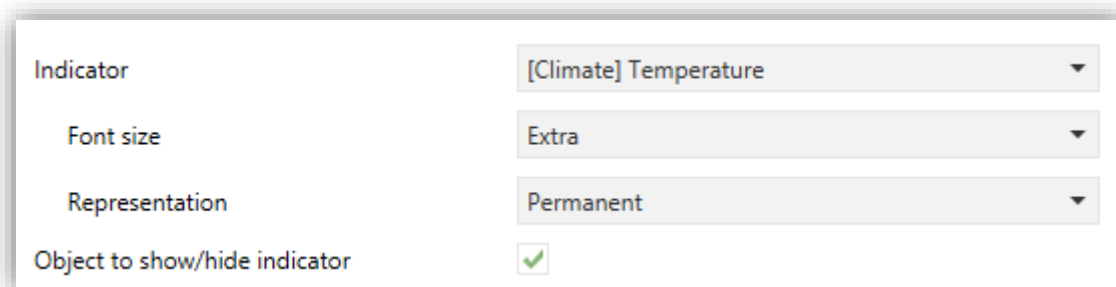


Figure 64 Display indicator – [Climate] Temperature.

The temperature needs to be updated through the “[Disp] [x] (Climate) Temperature” two-byte communication object.

- **Font size:** sets the font size of the temperature label. The options are: “Small”, “Big” and “Extra”.
- **Representation:** sets the temperature representation. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “**Indicator representation**” (see 2.4).*

- **Object to show/hide indicator:** enables or disables a one-bit object (“[Disp] [x] Show/Hide indicator”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** *this parameter will remain hidden unless having selected “Enable the option in every box” in “**Object to show/hide the indicator**” (see 2.4).*

### [Climate] Mode

Indicator	[Climate] Mode
Mode Type	<input checked="" type="radio"/> Heat/Cool <input type="radio"/> Extended
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 65 Display indicator – [Climate] Mode – Heat/Cool.

The climate mode needs to be updated through the “[Disp] [x] (Climate) Mode” one-bit object or the “[Disp] [x] (Climate) Mode - Extended” one-byte object, depending on the following parameterisation:

- **Mode Type:** sets whether only the Heat and Cool modes should be represented (“Heat/Cool”) or up to five HVAC modes (“Extended”).
  - “Heat/Cool”: the icon (Heat or Cool) will depend on the value of the “[Disp] [x] (Climate) Mode” one-bit object.
  - “Extended”: the icon represented will depend on the value of the “[Disp] [x] (Climate) Mode - Extended” one-byte object. When selecting this

type, five checkboxes appear to select which HVAC modes should be available (“Auto”, “Heat”, “Cool”, “Fan” and “Dry”, as shown in Figure 66).

Indicator	[Climate] Mode
Mode Type	<input type="radio"/> Heat/Cool <input checked="" type="radio"/> Extended
Auto	<input checked="" type="checkbox"/>
Heat	<input checked="" type="checkbox"/>
Cool	<input checked="" type="checkbox"/>
Fan	<input checked="" type="checkbox"/>
Dry	<input checked="" type="checkbox"/>
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 66 Display indicator – [Climate] Mode – Mode Type: Extended.

- **Representation:** sets the representation type. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** this parameter will remain hidden unless having selected “Every indicator is configured separately” in “Indicator representation” (see 2.4.1).

- **Object to show/hide indicator:** enables or disables a one-bit object (“[Disp] [x] Show/Hide indicator”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** this parameter will remain hidden unless having selected “Enable the option in every box” in “Object to show/hide the indicator” (see 2.4.1).

### [Climate] Fan

Indicator	[Climate] Fan
Off/Auto [0%]	<input type="checkbox"/>
Number of levels	<input type="radio"/> 2: Min. (0% - 50%); Max. (51% - 100%) <input checked="" type="radio"/> 3: Min. (0% - 33%); Med. (34% - 66%); Max. (67...
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 67 Display indicator – [Climate] Fan.

The value should be updated through the “[Disp] [x] Fan” one-byte communication object.

- **Off/Auto [0%]:** defines whether the Off/Auto level is required to be represented or not. If enabled, the following parameter shows up:

- **How to show Off/Auto status:** sets how to represent the Auto/Off state. The options are: “Auto-style (fan symbol + word “Auto”)”, “Off-style I (fan symbol + word “Off”)” and “Off-style II (clear the box)”.

- **Number of levels:** sets the number of fan speed levels. The available options depend on whether the Off/Auto level is enabled or not:

- **If Off/Auto is enabled:**

- “1: Max (1% - 100%)”,
- “2: Min (1% - 50%); Max (51% - 100%)”,
- “3: Min (1% - 33%); Med (34% - 67%); Max (68% - 100%)”

The value 0% does not correspond to any level, but to the Off/Auto state.

- **If Off/Auto is disabled:**

- “2: Min (0% - 50%); Max (51% - 100%)”,
- “3: Min (0% - 33%); Med (34% - 67%); Max (68% - 100%)”.

- **Representation:** sets the indicator representation. The options are: “Permanent”, “Temporary”, “Intermittent” and “Progressive”.

**Note:** *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “Indicator representation” (see 2.4).*

- **Object to show/hide indicator:** enables or disables a one-bit object (“[Disp] [x] Show/Hide indicator”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** *this parameter will remain hidden unless having selected “Enable the option in every box” in “Object to show/hide the indicator” (see 2.4.1).*



## [Climate] On/Off + Mode + Status



Figure 68 Display indicator – [Climate] On/Off + Mode + Status.

The icon to be represented depends on the value of three one-bit objects: “[Disp] [x] (Climate) On/Off”, “[Disp] [x] (Climate) On/Off - Mode” and “[Disp] [x] (Climate) On/Off - Status”:

- “[Disp] [x] (Climate) On/Off” = 0: no icon is represented.
- “[Disp] [x] (Climate) On/Off” = 1 and “[Disp] [x] (Climate) On/Off - Status” = 0, it means that the climate system is on but stopped. The icon represented will depend on the mode selected through “[Disp] [x] (Climate) On/Off - Mode” (1 = Heat or 0 = Cool). The representation type will be Permanent.
- If “[Disp] [x] (Climate) On/Off” = 1 and “[Disp] [x] (Climate) On/Off - Status” = 1, it means that the climate is on and running. The icon represented will depend on the mode selected through “[Disp] [x] (Climate) On/Off - Mode” (1 = Heat –with waves–, or 0 = Cool –with waves–). The representation type is Progressive.

**Note:** for this indicator, the representation type cannot be configured. It is always permanent or progressive, depending on the value of the objects, as explained above.

- **Object to show/hide indicator:** enables or disables a 1 bit object (“[Disp] [x] Show/Hide indicator”) to show or hide the indicator (1 = Show indicator; 0 = Hide indicator).

**Note:** this parameter will remain hidden unless having selected “Enable the option in every box” in “**Object to show/hide the indicator**” (see 2.4).

### 2.4.4 DOUBLE

Double boxes have a size of 32 x 32 pixels, so the display can be divided into up to 8 double boxes (two per zone).

Double boxes can be assigned exactly the same types of indicators as Single boxes (see section 2.4.3 for details).

The only differences between the two types of boxes are:

- The **maximum number of characters** allowed (up to 6 characters for double boxes; for single boxes the limit is 12 characters).
- The **font size** for some indicators. For example, the Extra size is not allowed for time and temperature indicators.

### ETS PARAMETERISATION

When a display zone has been set to Double, a specific tab (“**Box Xn**”) becomes available under “Display” in the tree on the left.

The ETS parameterisation of each indicator is exactly the same as for Single boxes. See section 2.4.3 for details.

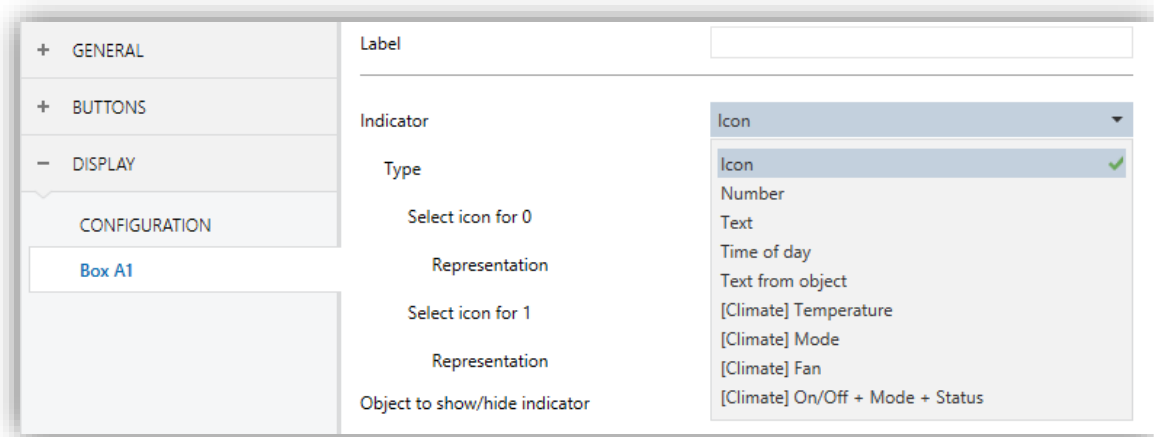


Figure 69 Display – Box a1.

### 2.4.5 ORDERS

Divisions between the boxes can be drawn in three different styles: with **solid lines**, **dashed** (dotted), or **invisible** (without division line). This configuration may be common to all lines or separate for each, as shown in Figure 70.

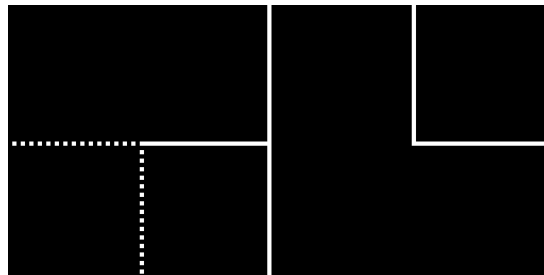


Figure 70 Borders – Different configuration for each border.

### ETS PARAMETERISATION

After selecting “Customize every border” in the **Border lines** parameter (see section 2.4.1), a new tab will be incorporated into the tree on the left.

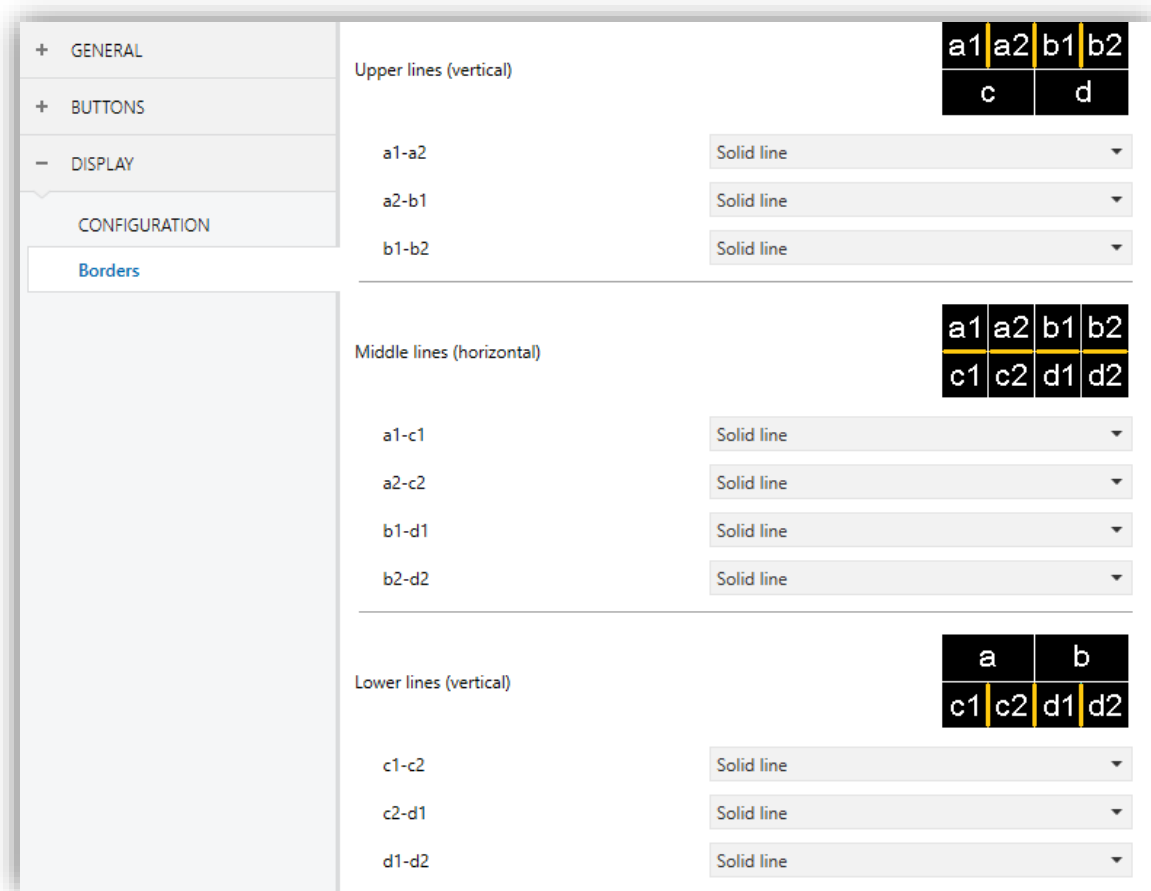


Figure 71 Display – Borders.

To simplify the configuration, this screen is divided into three parts (see Figure 71):

- **Upper lines (vertical):** configures the upper vertical lines **a1-a2**, **a2-b1** and **b1-b2**.
- **Middle lines (horizontal):** configures the middle horizontal lines **a1-c1**, **a2-c2**, **b1-d1** and **b2-d2**.
- **Lower lines (vertical):** configures the lower vertical lines **c1-c2**, **c2-d1** and **d1-d2**.

The options available for all the lines are: "Solid line", "Invisible" and "Dotted line".

## 2.5 INPUTS

---

Square TMD-Display incorporates **two analogue/digital inputs**, configurable as a:

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

**Important:** *older models of the Zennio motion detector (e.g., ZN1IO-DETEC and ZN1IO-DETEC-N) will not work properly with Squire TMD-Display.*

### 2.5.1 BINARY INPUT

---

Please refer to the specific “**Binary Inputs**” user manual, available in the Square TMD-Display product section at the Zennio website, <http://www.zennio.com>.

### 2.5.2 TEMPERATURE PROBE

---

Please refer to the specific “**Temperature Probe**” user manual, available in the Sqaure TMD-Display product section at the Zennio website, <http://www.zennio.com>.

### 2.5.3 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-Display. This brings the device with the possibility of monitoring motion and presence in the room. Depending on the detection, different response actions can be parameterised.

Please refer to the specific “**Motion Detector**” user manual (available in the Square TMD-Display product section at the Zennio website, [www.zennio.com](http://www.zennio.com)) for detailed information about the functionality and the configuration of the related parameters.

**Important:**

- *This device does not include **luminosity measurement capabilities**. Therefore, both motion detector models (ZN1IO-DETEC-P and ZN1IO-DETEC-X) offer equivalent functionality when connected to this device.*

- *Motion detectors with references ZN1IO-DETEC and ZN1IO-DETEC-N are **not compatible** with Square TMD-Display (they may report inaccurate measurements if connected to this device).*
- *When connected to Square TMD-Display, the rear micro-switch of model ZN1IO-DETEC-P should be set to position “**Type B**”.*

## 2.6 THERMOSTAT

---

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

Please refer to the specific manual “**Zennio Thermostat**” (available in the Square TMD-Display product section at the Zennio website, [www.zennio.com](http://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 2 illustrates which of the above are configurable for each function.

### **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).

		Disabled	Regular	State-dep.	State-dep. (both LEDs)	Dedicated object
PAIR	Switch		✓	✓	✓	✓
	Two Objects		✓	✓	✓	✓
	Dimmer		✓	✓	✓	✓
	Shutter		✓			✓
	Scaling		✓			✓
	Counter		✓			✓
	Enumeration		✓			✓
	Float		✓			✓
	[Climate] Temperature		✓			✓
	[Climate] Mode		✓	✓	✓	✓
	[Climate] Fan		✓			✓
INDIVIDUAL	Disabled					✓
	Switch		✓	✓		✓
	Hold & Rel.		✓			✓
	Two Objects		✓	✓		✓
	Scene		✓			✓
	Constants		✓	✓		✓
	Dimmer		✓	✓		✓
	Shutter		✓			✓
	Enumeration		✓			✓
DISABLED		✓				

Table 2 Functions vs. LED Illumination Options.

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 Illumination.**
- **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.

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	Name	Function
1	3 Bytes	I	C--W-	DPT_TimeOfDay	00:00:00 - 23:59:59	[General] Time of day	Time of day external reference
2	1 Byte	I	C--W-	DPT_SceneNumber	0-63	[General] Scene: receive	0-63 (Run scene 1-64)
3	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[General] Scene: send	0-63 / 128-191 (Run/Save scene 1-64)
4	1 Bit	I	C--W-	DPT_Enable	0/1	[General] Touch Locking	1 = Lock; 0 = Unlock
	1 Bit	I	C--W-	DPT_Enable	0/1	[General] Touch Locking	1 = Normal; 0 = Locked
5	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Cleaning Function	1 = Clean now; 0 = Nothing
6	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Welcome Greeting	1 = Welcome; 0 = Nothing
7	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[General] Screensaver - External temperature	Temperature to be shown on the screensaver
8	1 Bit	I	C--W-	DPT_Switch	0/1	[General] LEDs & Disp - Brightness mode	0 = Normal; 1 = Night
	1 Bit	I	C--W-	DPT_Switch	0/1	[General] LEDs & Disp - Brightness mode	0 = Night; 1 = Normal
9, 10, 11, 12	14 Bytes	I	C--W-			[General] Welcome Greeting - Line x	Text to show on the display at Line x
13	1 Bit		CT---	DPT_Switch	0/1	[General] Welcome back object	Switch object sent on Wake up
14	1 Byte	I	C--W-	DPT_Language_Selection	0-255	[General] Translations - Select Language	0 = Main; 1 = Lang. 2; ...; 4 = Lang. 5 (Scene format)
15	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Translations - Main language	1 = Choose this language; 0 = Nothing
16, 17, 18, 19	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Translations - Language x	1 = Choose this language; 0 = Nothing
20	2 Bytes	I	C--W-	DPT_LanguageCodeAlpha 2_ASCII		[General] Translations - Select Language	Language selection through ISO 639-1 two letters code
21	1 Byte	I	C--W-	DPT_SceneNumber	0-63	[General] Display - Contrast	0 = Level 1; ...; 9 = Level 10 (Scene format)
22	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
23	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Sounds - Doorbell	1 = Play a doorbell sound; 0 = Nothing
	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Sounds - Doorbell	0 = Play a doorbell sound; 1 = Nothing

24	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
25 - 29	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome back object - Additional condition	Additional condition object x
30	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Temperature Scale	0 = °C; 1 = °F
31, 37, 43, 49, 55	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] Switch	Send selected value on short press
	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] Hold & Release	Send selected values on hold and release presses
	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] Two Objects - Short press	Send selected value on short press
	1 Bit		CT - - -	DPT_Switch	0/1	[Btn] [Ix] Light - On/Off	(Short press) Switch between On and Off
	1 Bit		CT - - -	DPT_Step	0/1	[Btn] [Ix] Shutter - Stop/Step	(Short press) 0 = Stop shutter / Step up; 1 = Stop shutter / Step down
	1 Bit		CT - - -	DPT_Trigger	0/1	[Btn] [Ix] Shutter - Stop	(End pressing) Stop shutter
32, 38, 44, 50, 56	4 Bit	I	CT - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Btn] [Ix] Light - Dimming	(Long press) Switch between dimming up and down
	1 Bit		CT - - -	DPT_UpDown	0/1	[Btn] [Ix] Shutter - Move	(Long press) 0 = Up; 1 = Down
33, 39, 45, 51, 57	1 Bit		CT - - -	DPT_UpDown	0/1	[Btn] [Ix] Shutter - Move	(Start pressing) Switch between up and down
	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] Two Objects - Long press	Send selected value on long press
	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] Fan control - Auto mode	Switch auto mode on short press
	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] Fan control - Auto mode	Switch auto mode on long press
	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] LED On/Off	0 = Off; 1 = On
34, 40, 46, 52, 58	1 Bit	I	CT - W -	DPT_Switch	0/1	[Btn] [Ix] LED On/Off	0 = On; 1 = Off
	1 Byte	I	CT - W -	DPT_Scaling	0% - 100%	[Btn] [Ix] Scaling	Send selected percentage value on short press
35, 41, 47, 53, 59	1 Byte	I	CT - W -	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Counter - 1-byte unsigned	Send selected value on short press
	1 Byte	I	CT - W -	DPT_Value_1_Count	-128 - 127	[Btn] [Ix] Counter - 1-byte signed	Send selected value on short press
	2 Bytes	I	CT - W -	DPT_Value_2_Ucount	0 - 65535	[Btn] [Ix] Counter - 2-bytes unsigned	Send selected value on short press
	2 Bytes	I	CT - W -	DPT_Value_2_Count	-32768 - 32767	[Btn] [Ix] Counter - 2-bytes signed	Send selected value on short press
	2 Bytes	I	CT - W -	9.xxx	-671088.64 - 670760.96	[Btn] [Ix] Float	Send selected value on short press
	1 Byte	I	CT - W -	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Two Objects - Short press (x-byte)	Send selected 1-byte value on short press
	1 Byte	I	CT - W -	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Enumeration	Rotating value change
	1 Byte	I	CT - W -	DPT_Scaling	0% - 100%	[Btn] [Ix] Shutter Position	0 - 100 %

	1 Byte	I	<b>CT-W-</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Light Dimming (Status)	0 - 100 %
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	Auto, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	0%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	Auto, 50%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	0%, 50%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	50%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	33%, 67%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	0%, 33%, 67%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	Auto, 33%, 67%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	25%, 50%, 75%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	0%, 25%, 50%, 75%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	Auto, 25%, 50%, 75%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	20%, 40%, 60%, 80%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	0%, 20%, 40%, 60%, 80%, 100%
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan control (scaling)	Auto, 20%, 40%, 60%, 80%, 100%
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	1
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	0, 1
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	Auto, 1
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	1, 2
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	0, 1, 2
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	Auto, 1, 2
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	Auto, 1, 2, 3
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	0, 1, 2, 3
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	1, 2, 3
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	Auto, 1, 2, 3, 4
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	0, 1, 2, 3, 4
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	1, 2, 3, 4
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	Auto, 1, 2, 3, 4, 5
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	0, 1, 2, 3, 4, 5
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan Control (Enumerated)	1, 2, 3, 4, 5
36, 42, 48, 54, 60	1 Byte	I	<b>CT-W-</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Two Objects - Long press (1-byte)	Send selected 1-byte value on long press
	1 Byte	I	<b>CT-W-</b>	DPT_Scaling	0% - 100%	[Btn] [Ix] Fan - State	0-100%
	1 Byte	I	<b>CT-W-</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Ix] Fan - State	Enumerated Value
67, 73	1 Bit	I	<b>CT-W-</b>	DPT_Switch	0/1	[Btn] [Px] Switch	Left = 0; Right = 1
	1 Bit	I	<b>CT-W-</b>	DPT_Switch	0/1	[Btn] [Px] Two Objects - Short press	Left = 1; Right = 0

	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] Two Objects - Short press	Left = 0; Right = 1
	1 Bit		CT---	DPT_Switch	0/1	[Btn] [Px] Light - On/Off	(Short press) Left = Off; Right = On
	1 Bit		CT---	DPT_Step	0/1	[Btn] [Px] Shutter - Stop/Step	(Short press) Left = Stop/Step down; Right = Stop/Step up
	1 Bit		CT---	DPT_Trigger	0/1	[Btn] [Px] Shutter - Stop	(End pressing) Left = Stop-down; Right = Stop-up
	1 Bit	I	CT-W-	DPT_Heat_Cool	0/1	[Btn] [Px] (Climate) Mode	Left = Cool; Right = Heat
	1 Bit		CT---	DPT_Step	0/1	[Btn] [Px] Fan control (1 bit)	0 = Decrease; 1 = Increase
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] Switch	Left = 1; Right = 0
	1 Bit		CT---	DPT_Switch	0/1	[Btn] [Px] Light - On/Off	(Short press) Left = On; Right = Off
	1 Bit		CT---	DPT_Step	0/1	[Btn] [Px] Shutter - Stop/Step	(Short press) Left = Stop/Step up; Right = Stop/Step down
	1 Bit		CT---	DPT_Trigger	0/1	[Btn] [Px] Shutter - Stop	(End pressing) Left = Stop-up; Right = Stop-down
68, 74	4 Bit	I	CT-W-	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Btn] [Px] Light - Dimming	(Long press) Left = Darker; Right = Brighter
	4 Bit	I	CT-W-	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Btn] [Px] Light - Dimming	(Long press) Left = Brighter; Right = Darker
69, 75	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] Two Objects - Long press	Left = 0; Right = 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] Two Objects - Long press	Left = 1; Right = 0
	1 Bit		CT---	DPT_UpDown	0/1	[Btn] [Px] Shutter - Move	(Long press) Left = Down; Right = Up
	1 Bit		CT---	DPT_UpDown	0/1	[Btn] [Px] Shutter - Move	(Start pressing) Left = Down; Right = Up
	1 Bit		CT---	DPT_UpDown	0/1	[Btn] [Px] Shutter - Move	(Long press) Left = Up; Right = Down
	1 Bit		CT---	DPT_UpDown	0/1	[Btn] [Px] Shutter - Move	(Start pressing) Left = Up; Right = Down
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] Fan control - Auto mode	Switch auto mode on long press
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] Fan control - Auto mode	Switch Auto Mode on short press
70, 76	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] LED On/Off	0 = On; 1 = Off
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn] [Px] LED On/Off	0 = Off; 1 = On
71, 77	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn] [Px] Scaling	Left = Decrease percentage; Right = Increase percent.
	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn] [Px] Scaling	Left = Increase percentage; Right = Decrease percent.
	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Counter - 1-byte unsigned	Left = Decrease; Right = Increase
	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Counter - 1-byte unsigned	Left = Increase; Right = Decrease

2 Bytes	I	CT-W-	DPT_Value_2_Count	-32768 - 32767	[Btn] [Px] Counter - 2-bytes signed	Left = Decrease; Right = Increase
2 Bytes	I	CT-W-	DPT_Value_2_Count	-32768 - 32767	[Btn] [Px] Counter - 2-bytes signed	Left = Increase; Right = Decrease
1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Enumeration	Rotating value change on short press
2 Bytes	I	CT-W-	9.xxx	-671088.64 - 670760.96	[Btn] [Px] Float	Left = Decrease; Right = Increase
2 Bytes	I	CT-W-	9.xxx	-671088.64 - 670760.96	[Btn] [Px] Float	Left = Increase; Right = Decrease
2 Bytes	I	CT-W-	DPT_Value_Temp	-273.00 - 670760.00	[Btn] [Px] (Climate) Temperature setpoint	Left = Decrease; Right = Increase
2 Bytes	I	CT-W-	DPT_Value_Temp	-273.00 - 670760.00	[Btn] [Px] (Climate) Temperature setpoint	Left = Increase; Right = Decrease
1 Byte	I	CT-W-	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	[Btn] [Px] (Climate) Mode - Extended	Rotating change of selected modes on short press
1 Byte	I	CT-W-	DPT_Value_1_Count	-128 - 127	[Btn] [Px] Counter - 1-byte signed	Left = Decrease; Right = Increase
1 Byte	I	CT-W-	DPT_Value_1_Count	-128 - 127	[Btn] [Px] Counter - 1-byte signed	Left = Increase; Right = Decrease
2 Bytes	I	CT-W-	DPT_Value_2_Ucount	0 - 65535	[Btn] [Px] Counter - 2-bytes unsigned	Left = Decrease; Right = Increase
2 Bytes	I	CT-W-	DPT_Value_2_Ucount	0 - 65535	[Btn] [Px] Counter - 2-bytes unsigned	Left = Increase; Right = Decrease
1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn] [Px] Light Dimming (Status)	0 - 100 %
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	0%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	Auto, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	0%, 50%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	Auto, 50%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	50%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	Auto, 33%, 67%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	33%, 67%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	0%, 33%, 67%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	25%, 50%, 75%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	0%, 25%, 50%, 75%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	Auto, 25%, 50%, 75%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	20%, 40%, 60%, 80%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	0%, 20%, 40%, 60%, 80%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn] [Px] Fan control (scaling)	Auto, 20%, 40%, 60%, 80%, 100%
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	0, 1, 2, 3

	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	1, 2, 3
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	1, 2, 3, 4
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	0, 1, 2, 3, 4
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	Auto, 1, 2, 3, 4
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	Auto, 1, 2, 3, 4, 5
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	1, 2, 3, 4, 5
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	0, 1, 2, 3, 4, 5
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	Auto, 1, 2, 3
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	Auto, 1, 2
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	0, 1, 2
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	0, 1
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	1
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	1, 2
	1 Byte		<b>CT----</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan Control (Enumerated)	Auto, 1
72, 78	1 Byte	I	<b>CT-W-</b>	DPT_Scaling	0% - 100%	[Btn] [Px] Fan - State	0-100%
	1 Byte	I	<b>CT-W-</b>	DPT_Value_1_Ucount	0 - 255	[Btn] [Px] Fan - State	Enumerated Value
85, 101, 117, 133	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [X] Show/Hide indicator	0 = Hide indicator; 1 = Show indicator
85, 93, 101, 109, 117, 125, 133, 141	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [Xy] Show/Hide indicator	0 = Hide indicator; 1 = Show indicator
86, 102, 118, 134	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [X] Icon - 1-bit	Selected icon shown on new value reception
	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [X] Text - 1-bit	Parameterized text shown on new value reception
	1 Bit	I	<b>C--W-</b>	DPT_Heat_Cool	0/1	[Disp] [X] (Climate) Mode	Mode icon is shown (0 = Cool; 1 = Heat)
	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [X] (Climate) On/Off	0 = Off (hide indicator); 1 = On (show mode or status)
86, 94, 102, 110, 118, 126, 134, 142	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [Xy] Icon - 1-bit	Selected icon shown on new value reception
	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [Xy] Text - 1-bit	Parameterized text shown on new value reception
	1 Bit	I	<b>C--W-</b>	DPT_Heat_Cool	0/1	[Disp] [Xy] (Climate) Mode	Mode icon is shown (0 = Cool; 1 = Heat)
	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Disp] [Xy] (Climate) On/Off	0 = Off (hide indicator); 1 = On (show mode or status)
87, 103, 119, 135	1 Byte	I	<b>C--W-</b>	DPT_Value_1_Ucount	0 - 255	[Disp] [X] Icon - Enumeration	Selected icon shown on new value reception
	1 Byte	I	<b>C--W-</b>	DPT_Scaling	0% - 100%	[Disp] [X] Number - Scaling	Object numeric value is shown
	1 Byte	I	<b>C--W-</b>	DPT_Value_1_Ucount	0 - 255	[Disp] [X] Number - Counter (1-byte unsigned)	Object numeric value is shown
	1 Byte	I	<b>C--W-</b>	DPT_Value_1_Count	-128 - 127	[Disp] [X] Number - Counter (1-byte signed)	Object numeric value is shown
	1 Byte	I	<b>C--W-</b>	DPT_Value_1_Ucount	0 - 255	[Disp] [X] Text - Enumeration	Parameterized text shown on new value reception



	1 Byte	I	C - - W -	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	[Disp] [X] (Climate) Mode - Extended	Mode icon is shown
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [X] Fan	2 levels: Min. (0% - 50%); Max. (51% - 100%)
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [X] Fan	3 levels: Min. (0% - 33%); Med. (34% - 66%); Max. (67% - 100%)
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [X] Fan	Off/Auto + 1 level: Off/Auto = 0%; Max. (1% - 100%)
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [X] Fan	Off/Auto + 2 levels: Off/Auto = 0%; Min. (1% - 50%); Max. (51% - 100%)
87, 95, 103, 111, 119, 127, 135, 143	1 Byte	I	C - - W -	DPT_Value_1_Ucount	0 - 255	[Disp] [Xy] Icon - Enumeration	Selected icon shown on new value reception
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [Xy] Number - Scaling	Object numeric value is shown
	1 Byte	I	C - - W -	DPT_Value_1_Ucount	0 - 255	[Disp] [Xy] Number - Counter (1-byte unsigned)	Object numeric value is shown
	1 Byte	I	C - - W -	DPT_Value_1_Count	-128 - 127	[Disp] [Xy] Number - Counter (1-byte signed)	Object numeric value is shown
	1 Byte	I	C - - W -	DPT_Value_1_Ucount	0 - 255	[Disp] [Xy] Text - Enumeration	Parameterized text shown on new value reception
	1 Byte	I	C - - W -	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	[Disp] [Xy] (Climate) Mode - Extended	Mode icon is shown
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [Xy] Fan	2 levels: Min. (0% - 50%); Max. (51% - 100%)
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [Xy] Fan	3 levels: Min. (0% - 33%); Med. (34% - 66%); Max. (67% - 100%)
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [Xy] Fan	Off/Auto + 1 level: Off/Auto = 0%; Max. (1% - 100%)
	1 Byte	I	C - - W -	DPT_Scaling	0% - 100%	[Disp] [Xy] Fan	Off/Auto + 2 levels: Off/Auto = 0%; Min. (1% - 50%); Max. (51% - 100%)
88, 89, 104, 105, 120, 121, 136, 137	2 Bytes	I	C - - W -	DPT_Value_2_Ucount	0 - 65535	[Disp] [X] Number - Counter (2-bytes unsigned)	Object numeric value is shown
	2 Bytes	I	C - - W -	DPT_Value_2_Count	-32768 - 32767	[Disp] [X] Number - Counter (2-bytes signed)	Object numeric value is shown
88, 104, 120, 136	2 Bytes	I	C - - W -	9.xxx	-671088.64 -	[Disp] [X] Number - Float (2-bytes)	Object numeric value is shown

					670760.96		
	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[Disp] [X] (Climate) Temperature	Object value is shown (-99°C to 199°C)
88, 89, 96, 97, 104, 105, 112, 113, 120, 121, 128, 129, 136, 137, 144, 145	2 Bytes	I	C - - W -	DPT_Value_2_Ucount	0 - 65535	[Disp] [Xy] Number - Counter (2-bytes unsigned)	Object numeric value is shown
88, 96, 104, 112, 120, 128, 136, 144	2 Bytes	I	C - - W -	DPT_Value_2_Count	-32768 - 32767	[Disp] [Xy] Number - Counter (2-bytes signed)	Object numeric value is shown
	2 Bytes	I	C - - W -	9.xxx	-671088.64 - 670760.96	[Disp] [Xy] Number - Float (2-bytes)	Object numeric value is shown
	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[Disp] [Xy] (Climate) Temperature	Object value is shown (-99°C to 199°C)
90, 106, 122, 138	14 Bytes	I	C - - W -			[Disp] [X] Text from object	Received text is shown
90, 98, 106, 114, 122, 130, 138, 146	14 Bytes	I	C - - W -			[Disp] [Xy] Text from object	Received text is shown
91, 107, 123, 139	1 Bit	I	C - - W -	DPT_Heat_Cool	0/1	[Disp] [X] (Climate) On/Off - Mode	0 = Cool; 1 = Heat. (Mode icon is shown when On/Off = 1, and Status = 0)
91, 99, 107, 115, 123, 131, 139, 147	1 Bit	I	C - - W -	DPT_Heat_Cool	0/1	[Disp] [Xy] (Climate) On/Off - Mode	0 = Cool; 1 = Heat. (Mode icon is shown when On/Off = 1, and Status = 0)
92, 108, 124, 140	1 Bit	I	C - - W -	DPT_Switch	0/1	[Disp] [X] (Climate) On/Off - Status	0 = Halted; 1 = Running. (Status icon is shown when On/Off = 1, and Status = 1)
92, 100, 108, 116, 124, 132, 140, 148	1 Bit	I	C - - W -	DPT_Switch	0/1	[Disp] [Xy] (Climate) On/Off - Status	0 = Halted; 1 = Running. (Status icon is shown when On/Off = 1, and Status = 1)
149	1 Byte	I	C - - W -	DPT_SceneControl	0-63; 128-191	[Thermostat] Scene Input	Scene Value
150, 151	2 Bytes	I	C - - W -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Temperature Source x	External Sensor Temperature
152	2 Bytes	O	C T R - -	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Effective Temperature	Effective Control Temperature
153	1 Byte	I	C - - W -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode	1-byte HVAC Mode
154	1 Bit	I	C - - W -	DPT_Ack	0/1	[Tx] Special Mode: Comfort	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Special Mode: Comfort	0 = Off; 1 = On
155	1 Bit	I	C - - W -	DPT_Ack	0/1	[Tx] Special Mode: Standby	0 = Nothing; 1 = Trigger
	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] Special Mode: Standby	0 = Off; 1 = On
156	1 Bit	I	C - - W -	DPT_Ack	0/1	[Tx] Special Mode: Economy	0 = Nothing; 1 = Trigger

	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] Special Mode: Economy	0 = Off; 1 = On
157	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Special Mode: Protection	0 = Nothing; 1 = Trigger
	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] Special Mode: Protection	0 = Off; 1 = On
158	1 Bit	I	C--W-	DPT_Window_Door	0/1	[Tx] Window Status (Input)	0 = Closed; 1 = Open
159	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
160	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode Status	1-byte HVAC Mode
161	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Setpoint	Thermostat Setpoint Input
	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Basic Setpoint	Reference Setpoint
162	1 Bit	I	C--W-	DPT_Step	0/1	[Tx] Setpoint Step	0 = -0.5°C; 1 = +0.5°C
163	2 Bytes	I	C--W-	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Setpoint Offset	Float Offset Value
164	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Setpoint Status	Current Setpoint
165	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Basic Setpoint Status	Current Basic Setpoint
166	2 Bytes	O	CTR--	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Setpoint Offset Status	Current Setpoint Offset
167	1 Bit	I	C--W-	DPT_Reset	0/1	[Tx] Setpoint Reset	Reset Setpoint to Default
	1 Bit	I	C--W-	DPT_Reset	0/1	[Tx] Offset Reset	Reset offset
168	1 Bit	I	C--W-	DPT_Heat_Cool	0/1	[Tx] Mode	0 = Cool; 1 = Heat
169	1 Bit	O	CTR--	DPT_Heat_Cool	0/1	[Tx] Mode Status	0 = Cool; 1 = Heat
170	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] On/Off	0 = Off; 1 = On
171	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] On/Off Status	0 = Off; 1 = On
172	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] Control Variable (Cool)	PI Control (Continuous)
173	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] Control Variable (Heat)	PI Control (Continuous)
174	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Cool)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Cool)	PI Control (PWM)
175	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Heat)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Heat)	PI Control (PWM)
176	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Additional Cool	Temp >= (Setpoint+Band) => "1"
177	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Additional Heat	Temp <= (Setpoint-Band) => "1"
178	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] PI State (Cool)	0 = PI signal 0%; 1 = PI signal greater than 0%
179	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] PI State (Heat)	0 = PI signal 0%; 1 = PI signal greater than 0%
180, 184	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Ix] Current Temperature	Temperature sensor value
181, 185	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overcooling	0 = No Alarm; 1 = Alarm
182, 186	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overheating	0 = No Alarm; 1 = Alarm
183, 187	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Probe Error	0 = No Alarm; 1 = Alarm

188	2 Bytes	O	<b>CTR--</b>	DPT_Value_Temp	-273.00 - 670760.00	[Internal Probe] Current Temperature	Temperature sensor value
189	1 Bit	O	<b>CTR--</b>	DPT_Alarm	0/1	[Internal Probe] Overcooling	0 = No Alarm; 1 = Alarm
190	1 Bit	O	<b>CTR--</b>	DPT_Alarm	0/1	[Internal Probe] Overheating	0 = No Alarm; 1 = Alarm
191	1 Byte	I	<b>C--W-</b>	DPT_SceneControl	0-63; 128-191	[Motion Sensor] Scene Input	Scene Value
192	1 Byte		<b>CT---</b>	DPT_SceneControl	0-63; 128-191	[Motion Sensor] Scene Output	Scene Value
193, 217	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Ix] Luminosity	0-100%
194, 218	1 Bit	O	<b>CTR--</b>	DPT_Alarm	0/1	[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
195, 219	1 Bit	O	<b>CTR--</b>	DPT_Alarm	0/1	[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
196, 220	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Ix] Presence State (Scaling)	0-100%
197, 221	1 Byte	O	<b>CTR--</b>	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
198, 222	1 Bit	O	<b>CTR--</b>	DPT_Occupancy	0/1	[Ix] Presence State (Binary)	Binary Value
	1 Bit	O	<b>CTR--</b>	DPT_Ack	0/1	[Ix] Presence: Slave Output	1 = Motion Detected
199, 223	1 Bit	I	<b>C--W-</b>	DPT_Window_Door	0/1	[Ix] Presence Trigger	Binary Value to Trigger the Presence Detection
200, 224	1 Bit	I	<b>C--W-</b>	DPT_Ack	0/1	[Ix] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
201, 225	1 Bit	I	<b>C--W-</b>	DPT_Ack	0/1	[Ix] External Motion Detection	0 = Nothing; 1 = Motion detected by an external sensor
202, 207, 212, 226, 231, 236	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Ix] [Cx] Detection State (Scaling)	0-100%
203, 208, 213, 227, 232, 237	1 Byte	O	<b>CTR--</b>	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] [Cx] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
204, 209, 214, 228, 233, 238	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Ix] [Cx] Detection State (Binary)	Binary Value
205, 210, 215, 229, 234, 239	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Ix] [Cx] Channel Lock	According to parameters
206, 211, 216, 230, 235, 240	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Ix] [Cx] Force State	0 = No Detection; 1 = Detection
241, 247	1 Bit	I	<b>C--W-</b>	DPT_Enable	0/1	[Ix] Input Lock	0 = Unlock; 1 = Lock
242, 248	1 Bit		<b>CT---</b>	DPT_Switch	0/1	[Ix] [Short Press] 0	Sending of 0
	1 Bit		<b>CT---</b>	DPT_Switch	0/1	[Ix] [Short Press] 1	Sending of 1
	1 Bit	I	<b>CT-W-</b>	DPT_Switch	0/1	[Ix] [Short Press] 0/1 Switching	Switching 0/1

	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Ix] [Short Press] Brighter	Increase Brightness
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Ix] [Short Press] Darker	Decrease Brightness
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Ix] [Short Press] Brighter/Darker	Switch Bright/Dark
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] Light On	Sending of 1 (On)
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] Light Off	Sending of 0 (Off)
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Short Press] Light On/Off	Switching 0/1
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Run Scene	Sending of 0 - 63
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Save Scene	Sending of 128 - 191
	1 Bit	I/O	CTRW-	DPT_Switch	0/1	[Ix] [Switch/Sensor] Edge	Sending of 0 or 1
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Ix] [Short Press] Constant Value (Integer)	0 - 255
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Ix] [Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		CT---	DPT_Value_2_Ucount	0 - 65535	[Ix] [Short Press] Constant Value (Integer)	0 - 65535
	2 Bytes		CT---	9.xxx	-671088.64 - 670760.96	[Ix] [Short Press] Constant Value (Float)	Float Value
242, 243,	2 Bytes	0	CTRW-	DPT_Value_2_Ucount	0 - 65535	[Ix] [Pulse Counter] Counter	Number of Pulses

248, 249							
243, 249	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix] [Short Press] Shutter Status (Input)	0% = Top; 100% = Bottom
	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix] [Short Press] Dimming Status (Input)	0% - 100%
244, 250	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] 0	Sending of 0
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] 1	Sending of 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Long Press] 0/1 Switching	Switching 0/1
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Ix] [Long Press] Brighter	Long Pr. -> Brighter; Release -> Stop
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Ix] [Long Press] Darker	Long Pr. -> Darker; Release -> Stop
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0xE (Inc. by 3%) 0xF (Inc. by 1%)	[Ix] [Long Press] Brighter/Darker	Long Pr. -> Brighter/Darker; Release -> Stop
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] Light On	Sending of 1 (On)
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] Light Off	Sending of 0 (Off)
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Long Press] Light On/Off	Switching 0/1
1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Run Scene	Sending of 0 - 63	
1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Long Press] Save Scene	Sending of 128 - 191	
1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] [Switch/Sensor] Alarm: Breakdown or Sabotage	1 = Alarm; 0 = No Alarm	

	2 Bytes		CT---	9.xxx	-671088.64 - 670760.96	[Ix] [Long Press] Constant Value (Float)	Float Value
	2 Bytes		CT---	DPT_Value_2_Ucount	0 - 65535	[Ix] [Long Press] Constant Value (Integer)	0 - 65535
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Ix] [Long Press] Constant Value (Percentage)	0% - 100%
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Ix] [Long Press] Constant Value (Integer)	0 - 255
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Double Press] 0	Sending of 0
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Double Press] 1	Sending of 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Double Press] 0/1 Switching	Switching 0/1
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Double Press] Save Scene	Sending of 128 - 191
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Double Press] Run Scene	Sending of 0 - 63
245, 251	1 Bit		CT---	DPT_Trigger	0/1	[Ix] [Long Press/Release] Stop Shutter	Release -> Stop Shutter
	1 Bit	I	C--W-	DPT_Reset	0/1	[Ix] [Pulse Counter] Reset	0 = No Action; 1 = Reset
246, 252	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 Status (Input)	0% = Top; 100% = Bottom



Join and send us your inquiries  
about Zennio devices:

<http://support.zennio.com>

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

*Tel. +34 925 232 002*

*www.zennio.com*  
*info@zennio.com*



RoHS