Zennio®
Passion for KNX innovation

Touch-MyDesign
Plus

KNX Capacitive Touch Switch
(4/6/8 Main Buttons + 5 Additional Buttons)

ZVI-TMDP4
ZVI-TMDP6
ZVI-TMDP8

Application Program Version: [1.2]
User Manual Version: [1.2]_a

USER MANUAL

CONTENTS

DOCUMENT UPDATES

Version	Changes	Page(s)

[1.2]_a	**Changes in the application program**:
	• Individual buttons:
	   o New functionality (non-alternating control) in the dimmer and shutter controls.
	   o Dependence of the dimming and shutter motion directions to the corresponding status objects.
		-

	Minor text changes.	-
	Extension of the functionality of the 1-button dimming control.	25, 33
	Extension of the functionality of the 1-button shutter control.	26, 34
	Update of the communication object table.	57-66
[1.1]_a	**Changes in the application program**: • Internal optimisation of the ETS project.	-

1 INTRODUCTION

1.1 TOUCH-MYDESIGN PLUS

Touch-MyDesign Plus is an updated version of the original Touch-MyDesign, the KNX **multifunction capacitive touch switch** from Zennio. Like the classic model, Touch-MyDesign Plus provides 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**, **temperature sensor** and **thermostat** function, and by 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.

The inexistence of a screen, menus or complex user interaction beyond simply touching the buttons confers the device a **notable ease of use**.
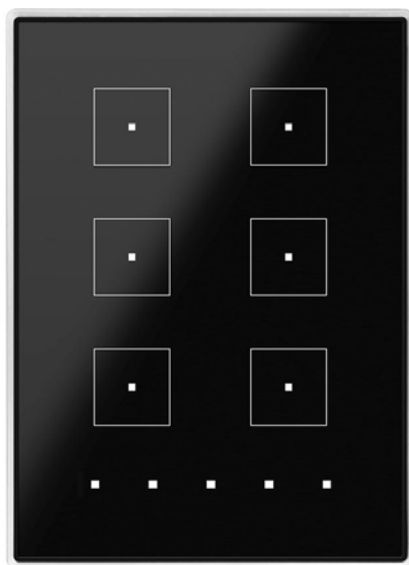

Figure 1 Touch-MyDesign Plus (6-Button Model)

The most outstanding features of Touch-MyDesign Plus are shown next.

- **Fully customisable** design of the front glass.

- **4 / 6 / 8 main touch buttons** (depending on the model), which can operate as individual or pair controls.

- **5 additional touch buttons**, which can operate as individual or joint controls.

- **Horizontal** or **vertical** orientation.

- **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.

- **Welcome Back object** (binary or scene), which will be sent to the bus on the first touch of the button panel after a certain (parameterisable) standby period.

- **Two analogue/digital inputs** (for motion sensors, temperature probes, additional switches, etc.).

- **Thermostat** function.

- Built-in **temperature sensor.**
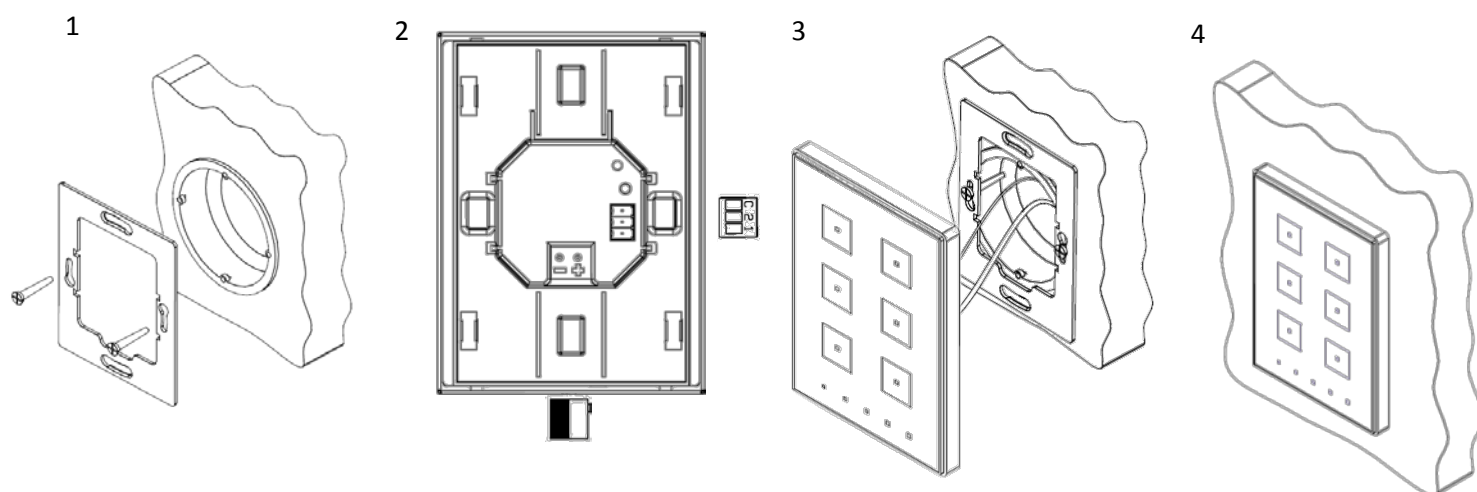
1.2 INSTALLATION

1   2   3   4

Figure 2 Touch-MyDesign Plus. Installation Process.

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, Touch-MyDesign Plus 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 the profile of the device becomes visible from above, from below and from both sides (the metallic plate should be completely hidden).

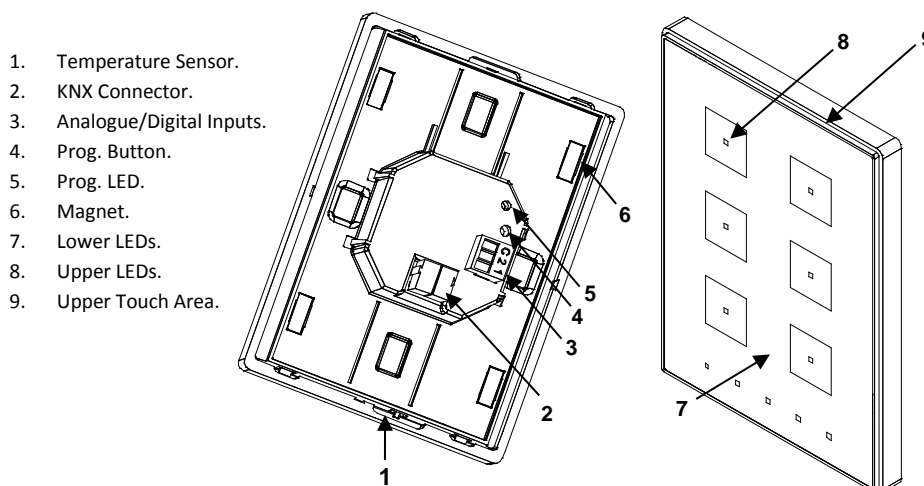This device does not need any external supply, as it is powered through the KNX bus.

1. Temperature Sensor.
2. KNX Connector.
3. Analogue/Digital Inputs.
4. Prog. Button.
5. Prog. LED.
6. Magnet.
7. Lower LEDs.
8. Upper LEDs.
9. Upper Touch Area.

Figure 3 Element diagram.

The programming button (4) may be pressed with the help of a thin screw to set Touch-MyDesign Plus 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 making pressure over 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 Touch-MyDesign Plus and for safety instructions for the installation process, please refer to the **Datasheet** bundled with the original packaging of the device and also available at www.zennio.com.

2 CONFIGURATION

2.1 GENERAL CONFIGURATION

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

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

ETS PARAMETERISATION

Figure 4 General Configuration.

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 parameters:

- **Main Buttons**: read-only parameter to make it evident that the Main Buttons configuration tab (containing all the parameters involved in the configuration of the main buttons of the device) is always enabled in the tab tree on the left. See section 0 for details.

- **Additional Buttons**: read-only parameter to make it evident that the Additional Buttons configuration tab (containing all the parameters related to the configuration of the additional buttons of the device) is always enabled in the tab tree on the left. See section 0 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 2.7 for details.

- **Thermostat**: enables or disables the Thermostat tab in the tree on the left, depending on whether this function is needed or not. See section 0 for details.

- **Touch-MyDesign Plus Orientation**: defines the orientation ("Horizontal" / "Vertical") of the device, with the aim of implementing a logical behaviour in the two-button controls.
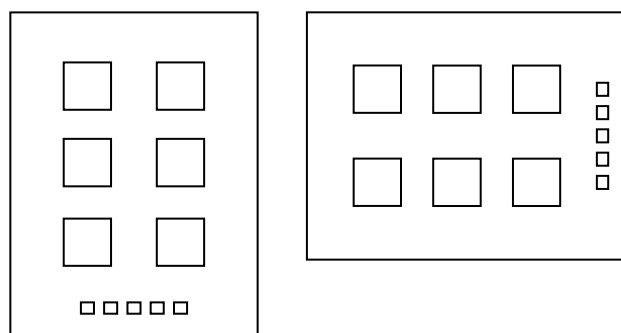

Figure 5 Device Orientation

Note: *the particular behaviour of the two-button controls (which one does what) is defined through **Button Pair Action**, as explained below.*

- **LED Brightness**: sets whether the LEDs should make use of the pre-defined brightness levels ("<u>Default</u>") or of a user-defined configuration ("<u>Custom</u>").

  - ➢ In "<u>Default</u>", the LEDs will remain off while in the "off" state, and at the maximum light level while in the "on" state.

  - ➢ In "<u>Custom</u>", a specific tab will be included in the tab tree on the left so the integrator can set the desired light levels for the "off" and "on" states and whether to use the Night Mode or not.

  See section 0 for details.

- **Touch Locking**: enables or disables the Touch Locking tab in the menu on the left, depending on whether this function is needed or not. See section 2.5 for details.

- **Welcome Back Object**: enables or disables the Welcome Back Object tab in the tree on the left, depending on whether this function is needed or not. See section 2.4 for details.

- **Sounds**: sets whether the sound functions (button beeps, alarm and doorbell) should work according to the pre-defined configuration ("<u>Default</u>") or to a user-defined configuration ("<u>Custom</u>"). See section 2.6 for details.

- **Internal Temperature Sensor**: enables or disables the Internal Temperature Sensor tab in the tree on the left, depending on whether this function is required or not. See section 2.9 for details.

- **Button Pair Action**: lets selecting how the two-button controls should behave depending on the device orientation (see above):

Vertically-oriented device	**Left**: 0 / Off / Decrease / Down. **Right**: 1 / On / Increase / Up
	Left: 1 / On / Increase / Up **Right**: 0 / Off / Decrease / Down.
	Every Button Pair is Configured Separately
Horizontally-oriented device	**Down**: 0 / Off / Decrease / Down **Up**: 1 / On / Increase / Up
	Down: 1 / On / Increase / Up **Up**: 0 / Off / Decrease / Down
	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 (see section 2.2.1.2).

- **LED Illumination Control (All Buttons)**: sets how the LED indicators of all the touch buttons should behave. The available options are (see section 2.3.1 for details):

  - ➢ Regular,
  - ➢ State-Dependent (where available),
  - ➢ Dedicated Object,
  - ➢ Every button (pair) is configured separately.

  In case of selecting "Every Button (Pair) is Configured Separately", there will be a specific parameter in every control for specifically selecting the desired behaviour of the LED (or LEDs).

- **Hide Info Parameters**: displays or hides the INFO textboxes on the top of the parameter screen of each function. These textboxes have no functionality – they are offered for the convenience of the integrator, as they allow assigning a description to each control, which may be useful in case of later re-parameterisation of the project.
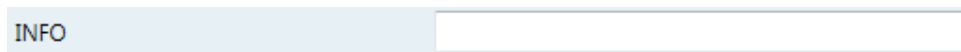
INFO	

Figure 6 "Info" Textboxes.

Apart from the above parameters, the project contains the following objects by default:

- **[General] Scene: Receive** and **[General] Scene: Send**: objects for respectively receiving and sending scene values from/to the KNX bus whenever it is necessary (e.g., when the user touches a button that has been configured to send scene commands; see section 2.2).

- **[Z] Temperature Setpoint**: object for sending temperature values when the user touches the additional push buttons, which by default are assigned the "Temperature Setpoint" function (see section 2.2).

2.2 TOUCH BUTTONS

Touch-MyDesign Plus features **four, six or eight main capacitive buttons** (depending on the model) at the user's disposal for the execution of actions. **Five additional capacitive buttons** can be found at the bottom of the front panel (if mounted vertically) or on the right side of the front panel (if mounted horizontally).

Each will perform a specific function at any time, as their functions do not depend on alternating menus, pages, etc.
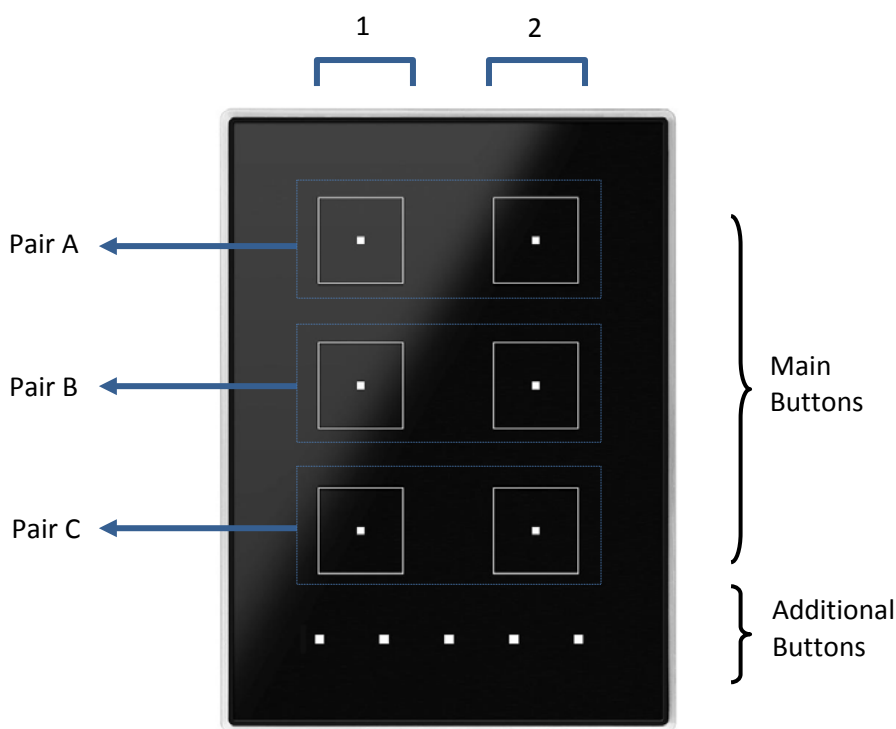

Figure 7 Touch panel.

As Figure 7 shows, two areas can be distinguished in the touch panel:

- **The main buttons**, grouped in pairs.

- **The additional buttons**, aligned and separated from the main buttons.

All the main buttons are identical, and also every additional button is identical to the others, which brings a high level of versatility for a wide variety of applications.

2.2.1 MAIN BUTTONS

The following is a list of the functions that can be assigned to each pair of main buttons.

- **Disabled** (none of the two buttons will react to user presses)**.**

- **Pair Buttons** (the two buttons will work jointly):
  - ➢ Switch.
  - ➢ Dimmer.
  - ➢ Shutter.
  - ➢ Two Objects (Short Press / Long Press).

- **Individual Buttons** (each button in the pair will work independently):
  - ➢ Disabled.
  - ➢ Binary.
  - ➢ Hold & Release.
  - ➢ Two Objects (Short Press / Long Press).
  - ➢ Scene.
  - ➢ Scaling Constant.
  - ➢ Counter Constant.
  - ➢ Float Constant.
  - ➢ Dimmer.
  - ➢ Shutter.

The next sections detail each of the above options.

ETS PARAMETERISATION

As stated in section 2.1, an independent tab for the parameterisation of the main buttons is shown in ETS by default. While all the main buttons remain disabled, it will only contain a sub-tab named Configuration.
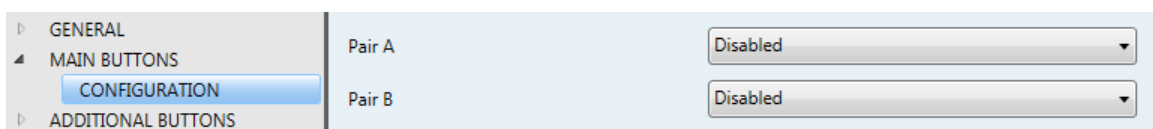
Figure 8 Main Buttons - Configuration.

One drop-down list is shown per pair of main buttons. The options under it are:

- Disabled. See section 2.2.1.1.

- Pair. Selecting this option brings a new tab to the tree on the left (named "Pair X", where X depends on the button pair). See section 2.2.1.2.

- Individual. Selecting this brings the checkboxes "**Button X1**" and "**Button X2**" (where X depends on the button pair), which turn each of the two buttons functional or non-functional. Depending on this, up to two new tabs (named "**Button Xn**") will be included in the tree on the left. See section 0.

    Note: *if a button pair is set to "Individual" but the "Button Xn" checkbox is left unmarked, that button will behave as if the pair had been set to "Disabled".*

2.2.1.1 DISABLED

While a button pair remains disabled, none of the two buttons will be functional: touching them will not cause the execution of any action, nor will make the associated LED light (whatever the option selected for "**Button Pair Action**"; see section 2.1).

ETS PARAMETERISATION

This function has no related parameters.

2.2.1.2 PAIR

Pair buttons configured to work jointly can be assigned the following control functions:

- **Switch**: pressing one of the two buttons will make Touch-MyDesign Plus 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 section **¡Error! No se encuentra el origen de la referencia.**), 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.

- **Dimmer**: short-pressing one of the two buttons will make Touch-MyDesign Plus 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 section **¡Error! No se encuentra el origen de la referencia.** for details), the LED of the corresponding button will remain on/off according to the current state of the dimmer (i.e., according to whether the on/off status object from the dimmer sends the value "0" or "1" and on the light level, which should be reported through the proper objects from the dimmer to Touch-MyDesign Plus). On the other hand, under a "**state-dependent (both LEDs)**" LED illumination, both of them will remain on or off is depending on whether the dimmer is "on" or "off" and whether the light level is greater than 0% or not.

- **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 section **¡Error! No se encuentra el origen de la referencia.** for details.

- **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 section **¡Error! No se encuentra el origen de la referencia.** for details) depend on either one object or the other. However, if "**LED Illumination Control (All Buttons)**" (section 2.1) has been set to "state-dependent (where available)" only the short press object will be considered.

ETS PARAMETERISATION

When a button pair has been configured to work as a joint control, a specific tab ("**Pair X**") becomes available under "Main Buttons" in the tree on the left.
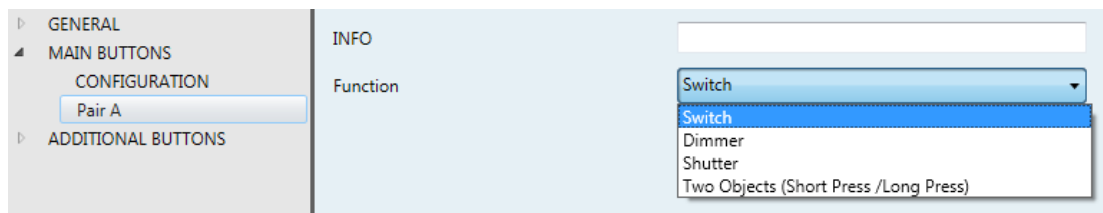
Figure 9 Main Buttons - Pair A.

Note: *the INFO textboxes are functionless (they simply bring the option to add a description to each button of the project) and can be hidden if desired. See section 2.1.*

The main parameter that needs to be configured is:

- **Function**: sets the desired function for the button pair: "Switch", "Dimmer", "Shutter" or "Two Objects (Short Press / Long Press)".

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, etc.).

Switch
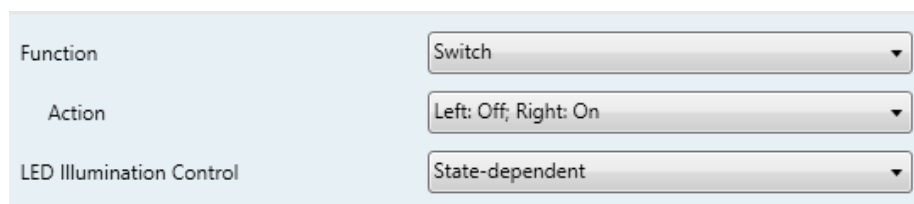
Figure 10 Pair Buttons – Switch.

- **Action**: assigns each of the two buttons the value to be sent through "**[X] Binary Control**" (which has the Write flag enabled, so the state of the switch can be updated from external devices). The options are "Left: Off; Right: On" and "Left: On; Right: Off" (or "Down: Off; Up: On" and "Down: On; Up: Off" under a horizontally-oriented configuration; see section 2.1).

  Note: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in* **Button Pair Action** *(see section 2.1).*

- **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 "**[X] LED On/Off**" will be included in the project topology.

  Note: *this parameter will remain hidden unless having selected "Every button (pair) is configured separately" in "LED Illumination Control (All Buttons)" (see 2.1).*

Dimmer

Function	Dimmer
Action	Left: Off/Decrease; Right: On/Increase
Dimming Step	25%
LED Illumination Control	Dedicated Object

Figure 11 Pair Buttons – Dimmer.

The switch orders will be sent through the "**[X] Light On/Off**" one-bit object, while the increase/decrease orders will be through the "**[X] Light Dimming**" four-bit object. The first one has the *write* flag enabled so it can receive feedback from the dimmer about the on/off status. Similarly, the "**[X] Light Dimming (Status)**" one-byte object can 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 2.2.1.2, the alternation of the on/off and step requests will be conditioned to these statuses, to prevent sending ineffective requests.

The parameters for this function are:

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

  ➢ "<u>Left: Off / Decrease; Right: On / Increase</u>" (or "Down: Off / Decrease; Up: On / Increase" under a horizontally-oriented configuration),

  ➢ "<u>Left: On / Increase; Right: Off / Decrease</u>" (or "Down: On / Increase; Up: Off / Decrease" under a horizontally-oriented configuration).

  Note: *this parameter will remain hidden unless having selected "<u>Every button pair is configured separately</u>" in* **Button Pair Action** *(see section 2.1).*

- **Dimming Step**: defines the dimming step ("<u>100%</u>", "<u>50%</u>", "<u>25%</u>", "<u>12%</u>", "<u>6%</u>", "<u>3%</u>" or "<u>1%</u>") 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 Touch-MyDesign Plus 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 the "Switch" function (see above).

Shutter

Function	Shutter
Action	Left: Upwards; Right: Downwards
Type	Standard
LED Illumination Control	Regular

Figure 12 Pair Buttons – Shutter.

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

The parameters for this function are:

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

  ➤ "Left: Downwards; Right: Upwards" (or "Down: Downwards; Up: Upwards" under a horizontally-oriented configuration).

  ➤ "Left: Upwards; Right: Downwards" (or "Down: Upwards; Up: Downwards" under a horizontally-oriented configuration).

  Note: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in* **Button Pair Action** *(see section 2.1).*

- **Type**: sets the desired behaviour of the buttons, "Standard" or "Hold & Release" (note: the differences between them have been explained in previous pages).

- **LED Illumination Control**: analogous to the homonymous parameter for the "Switch" and "Dimmer" functions (see above). In this case, the only options are "Regular" and "Dedicated Object".

Two Objects (Short Press / Long Press)

Figure 13 Pair Buttons - Two Objects (Short Press / Long Press).

- **Action on Short Press**: sets the value that will be sent through "**[X] Two Objects: Short Press**" after the user short-presses one of the two buttons.

  ➤ "Left: 0; Right: 1" (or "Down: 0; Up: 1" under a horizontally-oriented configuration.

  ➤ "Left: 1; Right: 0" (or "Down: 1; Up: 0" under a horizontally-oriented configuration.

  Note: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in* **Button Pair Action** *(see section 2.1).*

- **Action on Long Press**: sets the value that will be sent through "**[X] Two Objects: Long Press**" after the user long-presses one of the two buttons.

  - ➢ "Left: 0; Right: 1" (or "Down: 0; Up: 1" under a horizontally-oriented configuration.

  - ➢ "Left: 1; Right: 0" (or "Down: 1; Up: 0" under a horizontally-oriented configuration.

  Note: *this parameter will remain hidden unless having selected "Every button pair is configured separately" in* **Button Pair Action** *(see section 2.1).*

- **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 4 to 50 tenths of a second, being 6 tenths the default value.

- **LED Illumination Control**: analogous to the homonymous parameter for the Switch and Dimmer functions (see above). In this case, however, selecting "State-dependent" or "State-dependent (both LEDs)" brings an additional parameter, "**LED Illumination State Object**", to make the state of the LEDs correspond either to object "**[X] Two Objects: Short Press**" ("Short Press Object") or to object "**[X] Two Objects: Long Press**" ("Long Press Object").
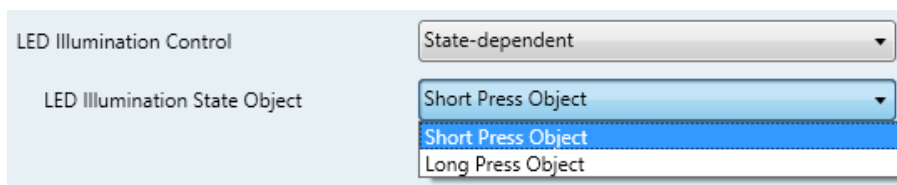
Figure 14 Pair Buttons - Two Objects (Short Press / Long Press) - LED Illumination Control.

2.2.1.3 INDIVIDUAL

Buttons of pairs configured to work as individual (separate) controls can be assigned any of the following control functions:

- **Disabled**: user presses will not trigger any function although the LED will still react to them, if configured.

- **Binary**: 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 illumination 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 2.1) 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 -671088.64 to 670760.96.

  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:

  ➢ <u>Switch-on / Switch-off orders</u> (on short presses).

  ➢ <u>Step dimming orders</u> (on long presses) and the subsequent stop order once the button is released.

  The specific on / off orders which the button will be able to send may be chosen from the three following available actions:

  ➢ <u>Off / Decrease</u>: a short press will always send a switch-off order, while a long press will always send a decrease order.

  ➢ <u>On / Increase</u>: a short press will always send a switch-on order, while a long press will always send an increase order.

  ➢ <u>Toggle</u>: this option **will alternate** the switch-on and switch-off orders after every short press, and so will do the step dimming orders (increase / decrease) for every long press. However, some exceptions must be taken into account:

    • <u>On a long press</u>: an increase order will be sent if the current light level is found to be 0%. On the other hand, a decrease order will be sent if the current light level is found to be 100%.

    • <u>On a short press</u>: a switch-on order will be sent if the current light level is found to be 0%. On the other hand, a switch-off order will be sent if the current light level is found to be 100%.

  Note that the device considers that the **current light level** is the value of a specific objected provided to be written from the KNX bus (i.e., to receive feedback from the dimmer). The light dimmer is therefore responsible of keeping its value up to date.

Under a "**state-dependent**" LED illumination, the LED will remain on/off according to the following:

➢ After sending a switch-on order to the bus, the LED will turn on. After sending a switch-off order to the bus, the LED will turn off.

➢ After sending a dimming order greater than 0%, the LED will turn on. After sending a dimming order equal to 0%, the LED will turn off.

➢ After receiving from the dimmer the on/off status object, the LED will turn on/off depending on the value of the object.

➢ After receiving from the dimmer the dimming status object, the LED will turn on/off depending on whether the value is greater than 0% or not.

Note: after a bus failure, the LED will remain off unless after the bus recovery the light dimmer sends back the on/off status object or the dimming status object updated.

● **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**).

   The specific raise / lower orders which the button will be able to send may be chosen from the three following available actions:

   • Up: a short press will always send a step-up / stop order, while a long press will always send a *move up* order.

   • Down: a short press will always send a step-down / stop order, while a long press will always send a *move down* order.

   • Toggle: choosing this option, the direction of the motion will **alternate** (upwards / downwards) for both the move and the step

orders after every press. However, there are some exceptions to this alternation:

- <u>On a short press</u>: a step-up order will be sent if the current position is found to be 100%. On the other hand, a step-down order will be sent if the current position is found to be 0%.

- <u>On a long press</u>: a move-up order will be sent if the current position is found to be 100%. On the other hand, a move-down order will be sent 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).

Touch-MyDesign Plus 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.

➢ <u>Hold & Release</u>: 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.

Alike the standard control, the individual button may be configured to perform three possible actions:

- <u>Up</u>: pressing the button will always send a *move up* order. Releasing the button will always send a *stop* order.

- <u>Down</u>: pressing the button will always send a *move down* order. Releasing the button will always send a *stop* order.

- <u>Toggle</u>: 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.

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

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

ETS PARAMETERISATION

When an individual button has been enabled, a specific tab ("**Button X1**" or "**Button X2**") becomes available under "Main Buttons" in the tree on the left.

Figure 15 Individual Button.

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 2.1.*

The main parameter that needs to be configured is:

- **Function**: sets the desired function for the button: "Disabled", "Binary", "Hold & Release", "Two Objects (Short Press / Long Press)" or "Scene", "Scaling Constant", "Counter Constant", "Float Constant", "Dimmer" and "Shutter".

Depending on the function, some more parameters are involved (as described next). Please note that in the next pages "**[Xn]**" is used as a general notation for the

communication objects, as "X" depends on the button pair (A, B, etc.) and "n" depends on the particular button within the pair (1 or 2).

Disabled

INFO	
Function	Disabled
LED Illumination Control	Dedicated Object
Object Value	0 = Off; 1 = On

Figure 16 Individual Button - Disabled.

- **LED Illumination Control**: sets the behaviour of the LED on the button. The options are "Disabled" and "Dedicated Object" (see section **¡Error! No se encuentra el origen de la referencia.** for details about each option).

  In case of selecting the latter, object "**[Xn] LED On/Off**" will be included in the project topology (the desired values to make the LED turn on/off must be set through "**Object Value**").

- **Note**: *this parameter will remain hidden unless having selected "Every button (pair) is configured separately" in "**LED Illumination Control (All Buttons)**" (see 2.1).*

Binary

INFO	
Function	Binary
Action	Send 0
LED Illumination Control	State-dependent

Figure 17 Individual Button - Binary.

- **Action**: sets the value to be sent to the bus (through objects "**[Xn] Binary Control: B**" (where "B" depends on the action selected) 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 "<u>Regular</u>", "<u>State-dependent</u>" and "<u>Dedicated Object</u>".

Hold & Release

INFO	
Function	Hold & Release
Action on Hold	Send 1
Action on Release	Send 0
LED Illumination Control	Regular

Figure 18 Individual Button – Hold & Release.

- **Action on Hold**: sets the value to be sent to the bus (through "**[Xn] Binary Control, Press/Release**") when the user touches the button. The options are "<u>Send 0</u>" and "<u>Send 1</u>" (default).

- **Action on Release**: sets the value to be sent to the bus (again, through "**[Xn] Binary Control, Press/Release**") when the user stops touching the button. The options are "<u>Send 0</u>" (default) and "<u>Send 1</u>"

- **LED Illumination Control**: analogous to the homonymous parameter for the "Disabled" and "Binary" functions (see above). The options in this case are "<u>Regular</u>" and "<u>Dedicated Object</u>".

Two Objects (Short Press / Long Press)

Figure 19 Individual Button – Two Objects (Short Press / Long Press).

- **Action on Short Press**: sets the value to be sent to the bus (through "**[Xn] Two Objects, Short Press: B**") when the user short-presses the button. The options are "Send 0", "Send 1", "Toggle 0/1" and "Send 1-Byte Value (Unsigned Int.)". 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 "**[Xn] Two Objects, Long Press: B**") when the user long-presses the button. The options are the same as for the short 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 4 to 50 tenths of a second, being 6 tenths the default value

- **LED Illumination Control**: analogous to the homonymous parameter for the "Disabled", "Binary" and "Hold & Release" functions (see previous pages). The options in this case are "Regular", "State-dependent" and "Dedicated Object".

  Selecting "State-dependent" brings an additional parameter, "**LED Illumination State Object**", to make the state of the LED correspond either to object "**[X] Two Objects: Short Press**" ("Short Press Object") or to object "**[X] Two Objects: Long Press**" ("Long Press Object").

Scene

Figure 20 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

Figure 21 Individual Button - Scaling Constant.

- **Object 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:

	Available Values	**Name of the Object**
Scaling Constant	0% – 100%	[Xn] 1-Byte Value (Scaling)
Counter Constant	0 – 255	[Xn] 1-Byte Value (Unsigned Int)
	-128 – 127	[Xn] 1-Byte Value (Signed Int)
	0 – 65535	[Xn] 2-Byte Value (Unsigned Int)
	-32768 – 32767	[Xn] 2-Byte Value (Signed Int)
Float Constant	-671088.64 – 670760.96	[Xn] 2-Byte Value (Float)

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.

- **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

Figure 22 Individual Button - Dimmer.

The switch orders will be sent through the "**[Xn] Light On/Off (Toggle)**", "**[Xn] Light On**" or "**[Xn] Light Off**" one-bit objects (depending on the selected action: "Toggle", "On/Increase" or "Off/Decrease", respectively), while the increase/decrease/stop orders will be through the "**[Xn] Light Dimming (Toggle)**", "**[Xn] Light Dimming (Increase)**" or "**[Xn] Light Dimming (Decrease)**" four-bit objects (also depending on the selected action).

The "**[Xn] Light Dimming (Status)**" one-byte object should be linked to the light level status object of the dimmer. As explained in 2.2.1.3, the alternation of the on/off and

the step orders will be conditioned to this status, to prevent sending ineffective requests.

The parameters for this function are:

- **Action**: defines the order type that the button will send once a short or long press is performed ("Toggle", "Off/Decrease" or "On/Increase").

- **Dimming Step**: defines the dimming step ("100%", "50%", "25%", "12%", "6%", "3%" or "1%") to be sent (through "**[Xn] Light Dimming (Toggle/Increase/Decrease)**") 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 Touch-MyDesign Plus 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

INFO	
Function	Shutter
Type	Standard
Action	Toggle
LED Illumination Control	Dedicated Object
Object Value	0 = Off; 1 = On

Figure 23 Individual Button - Shutter.

The move up/down orders will be sent through the "**[Xn] Move Shutter (Toggle Direction)**", "**[Xn] Move Shutter Up**" or "**[Xn] Move Shutter Down**" one-bit objects (depending on the selected action: "Toggle", "Up" or "Down", respectively), while the step up/down orders will be through the "**[Xn] Stop Shutter / Step**", "**[Xn] Stop**

Shutter / Step Up" or "**[Xn] Stop Shutter / Step Down**" one-bit objects (also depending on the selected action: "Toggle", "Up" or "Down", respectively).

Additionally, if the "Toggle" action is selected, a one-byte object ("**[Xn] Shutter Position**") will be provided to link it to the position status object from the shutter actuator. As explained in 2.2.1.3, the alternation of the move up/down and the step up/down orders will be conditioned to this status, 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 2.2.1.3.

- **Action**: defines the order type that the button will send once a short or long press is performed ("Toggle", "Up" or "Down").

- **LED Illumination Control**: analogous to the homonymous parameter for all the previous functions (see previous pages). The options in this case are "Regular" and "Dedicated Object".

2.2.2 ADDITIONAL BUTTONS

The following is a list of the functions (joint or individual) that can be assigned to the five additional buttons of the device.

- **Temperature Setpoint.**

- **1-Byte Control (Unsigned Int.).**

- **1-Byte Control (Scaling).**

- **Individual Buttons** (each additional button will work independently):

  - ➢ Disabled.
  - ➢ Binary.
  - ➢ Hold & Release.
  - ➢ Two Objects (Short Press / Long Press).
  - ➢ Scene.
  - ➢ Scaling Constant.
  - ➢ Counter Constant.
  - ➢ Float Constant.
  - ➢ Dimmer.
  - ➢ Shutter.

The next sections detail each of the above options.

ETS PARAMETERISATION

Figure 24 Additional Buttons - Configuration
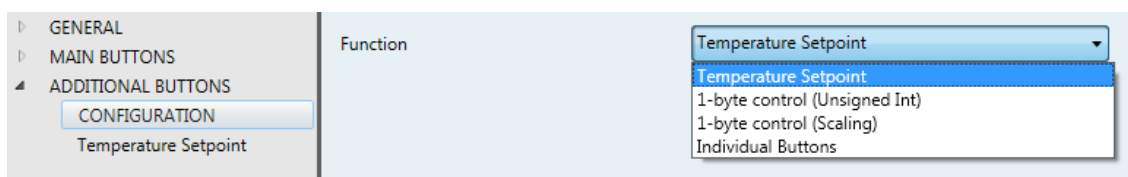
As stated in section 2.1, an independent tab for the parameterisation of the additional buttons is shown in ETS by default. This tab comprises another one named "**Configuration**", containing one parameter:

- **Function**: sets the desired function for the additional buttons. The options are "Temperature Setpoint" (default selection), "1-Byte Control (Unsigned Int)", "1-

Byte Control (Scaling)" and "Individual Buttons". In case of selecting the latter, five checkboxes (one per additional button) will be displayed to individually turn functional / non-functional each additional button.

Depending on these checkboxes, the "Additional Buttons" tab will comprise some more additional subsections.

Each of these functions will be described next.

2.2.2.1 TEMPERATURE SETPOINT

This function lets the user control the setpoint of a climate system by simply touching the appropriate additional button, which will result in sending a two-byte floating point value to the KNX bus. Each button can be assigned a different value, but all of them will be sent through the same communication object, thus working as an easy-to-configure, joint control.

On the other hand, the illumination of the LEDs will be state-dependent. Therefore, the LED of the last button pressed will remain on, and the others off. If the object that sends the setpoints receives a value from the bus and it falls into one of those parameterised, the corresponding LED will turn on (and the others off).

ETS PARAMETERISATION

When the additional buttons have been assigned the "Temperature Setpoint" joint function, a specific tab ("**Temperature Setpoint**") will be available under "Additional Buttons" in the tree on the left.

Figure 25 Additional Buttons - Temperature Setpoint.

This window contains five textboxes ("**Button n: Value**") to enter the particular setpoint

that will be sent to the KNX bus through the "**[Z] Temperature Setpoint**" two-byte object when the user touches each button. The accepted range is -20 to 150 ºC.

As already mentioned, the illumination of the LEDs for this function is state-dependent. Moreover, if "**[Z] Temperature Setpoint**" receives a value that matches that of one of the buttons, the LED of such button will turn on (and the others off).

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 2.1.*

2.2.2.2 1-BYTE CONTROL (UNSIGNED INTEGER)

This is completely analogous to the above "Temperature Setpoint" function, although the values that can be sent to the bus will be in this case one-byte integers (0 to 255).

The illumination of the LEDs is also state-dependent, as for "Temperature Setpoint".

ETS PARAMETERISATION

Figure 26 Additional Buttons – 1-Byte Control (Unsigned Int)

This window contains five textboxes ("**Button n: Value**") to enter the particular unsigned integer value (0 to 255) that will be sent to the KNX bus through "**[Z] 1-Byte Value (Unsigned Int)**" when the user touches each of the buttons.

As already mentioned, the illumination of the LEDs for this function is state-dependent. Moreover, if "**[Z] 1-Byte Value (Unsigned Int)**" receives a value that matches that of one of the buttons, the LED of such button will turn on (and the others off).

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 2.1.*

2.2.2.3 1-BYTE CONTROL (SCALING)

This is completely analogous to the above "Temperature Setpoint" and "1-Byte Control (Unsigned Integer)" functions, although the values that can be sent to the bus in this case are percentage values (0% to 100%).

The illumination of the LEDs is also state-dependent, as for the previous functions.

ETS PARAMETERISATION

Figure 27 Additional Buttons - 1-Byte Control (Scaling).

This window contains five textboxes ("**Button n: Value**") to enter the particular percentage value (0% to 100%) that will be sent to the KNX bus through "**[Z] 1-Byte Value (Scaling)**" when the user touches one of the additional buttons.

As already mentioned, the illumination of the LEDs for this function is state-dependent. Moreover, if "**[Z] 1-Byte Value (Scaling)**" receives a value that matches that of one of the buttons, the LED of such button will turn on (and the others off).

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 2.1.*

2.2.2.4 INDIVIDUAL BUTTONS

This function brings the option to use the additional buttons (all or just the ones required) not as a joint control but as separate controls, each with a specific function.

The functions that can be assigned to the additional buttons once they have been configured to work as individual controls are entirely analogous to those for the main buttons (except for "*Dimmer*" and "*Shutter*", which won't allow individual action selection and will always perform a "*Toggle*" behaviour) working as such:

- Disabled.

- Binary.

- Hold & Release.

- Two Objects (Short Press / Long Press).

- Scene.

- Scaling Constant.

- Counter Constant.

- Float Constant.

- Dimmer.

- Shutter.

All the above functions and their ETS parameterisation have already been described in section 2.2.1.3.

ETS PARAMETERISATION

Additional buttons working as individual controls are entirely analogous to the main buttons working as such. Therefore, please refer to section 2.2.1.3 for details about the functionality and the ETS parameterisation of these functions, keeping in mind that when they are assigned to an additional button, **the name of the involved communication objects will start with "[Zn]"** (with "n" being the number of the button: 1, 2, 3, 4 or 5) **and not with "[Xn]"** (with "X" depending on the button pair: A, B, etc.).

2.3 LED ILLUMINATION

2.3.1 ILLUMINATION MODES

Every main or additional button incorporates a central LED indicator which 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 main buttons configured as pair controls. The two LEDs of the control will light or not, depending on the value of the related object and on the particular control type parameterised for that pair of buttons. The only difference compared to the previous case is that, under "both LEDs", the two LEDs will always turn off or on simultaneously, as if it were a unique indicator consisting of two LEDs.

- **Dedicated Object**: the LED will light or not depending on the value ("0" or "1", configurable) of a binary, independent object. In the case of the pair controls, the value "0" will make one of the LEDs light (leaving the other one off), while the value "1" will make them switch their states.

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

Note:

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

- *Button Pair configured as "Disabled": the two LEDs will be functionless (off).*

- *Button Pair Configured as "Individual", but buttons themselves disabled: the LEDs will still be functionless (as above).*

- *Button Pair Configured as "Individual", and buttons configured as "Disabled": the buttons will remain functionless, while the behaviour of the LEDs will be configurable (being also possible to leave them off), as shown in the table.*

		Disabled	Regular	State-dep.	State-dep. (both LEDs)	Dedicated object
PAIR (Main Buttons)	Switch		✓	✓	✓	✓
	Dimmer		✓	✓	✓	✓
	Shutter		✓			✓
	Two Objects		✓	✓	✓	✓
INDIVIDUAL (Main or Additional Buttons)	Disabled	✓	✓			✓
	Binary		✓	✓		✓
	Hold & Rel.		✓			✓
	Two Objects		✓	✓		✓
	Scene		✓			✓
	Constants		✓	✓		✓
	Dimmer		✓	✓		✓
	Shutter		✓			✓
JOINT (Additional Buttons)	Setpoint			✓		
	Unsigned			✓		
	Scaling			✓		

Table 1 Functions vs. LED Illumination Options.

Although the behaviour of the LEDs can be configured independently for each control (see section 2.1), it is also possible to define a **general behaviour for all of them**, 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.

- **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 2.2).

In case of desiring a **similar behaviour for all of the LEDs**, please find the parameter "**LED Illumination Control (All Buttons)**" in the General configuration screen. This parameter is explained in section 2.1.

2.3.2 BRIGHTNESS LEVELS

As anticipated in section 2.1, 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"). When they switch the state depends on the illumination mode, as already explained in section 2.3.1.

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

To sum up, it is possible to configure the following:

- **Normal Mode**:

  - "Off" Brightness Level: allowed values are from 0 (default value) to 255.

  - "On" Brightness Level: allowed values are from 0 to 255 (default value).

- **Night Mode** (not switchable by default):

  - "Off" Brightness Level: allowed values are from 0 to 255 (1 by default).

  - "On" Brightness Level: allowed values are from 0 to 255 (8 by default).

A general parameter is provided to make the ETS parameterisation easier in case the night mode is not required and the integrator feels comfortable with the default on/off brightness levels. On the other hand, customising these modes and levels involves a set of parameters, as explained next.

ETS PARAMETERISATION

In case the default brightness levels of the LEDs match the requirements of the installation, the "**LED Brightness**" parameter in the general **Configuration** tab (see section 2.1) can be set to "Default". In such case, the night mode will not be available and the LEDs will remain off while in the *off* state, and at the maximum brightness level while in the *on* state.

On the other hand, if that parameter is set to "Custom", a specific tab named "**LED Brightness**" will show up in the tree on the left.
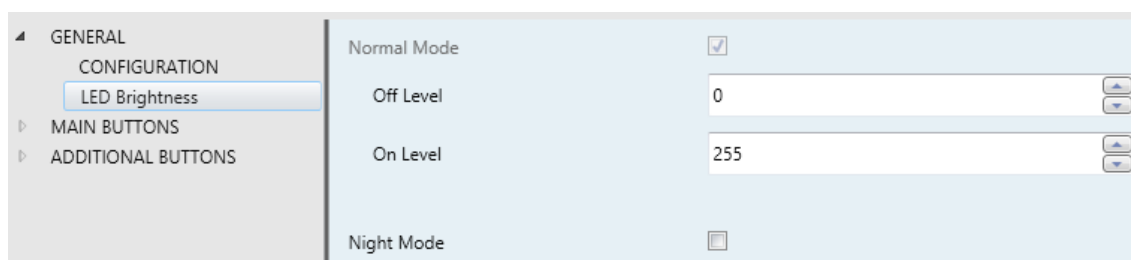
Figure 28 General - LED Brightness.

The initial configuration of this screen is equivalent to the aforementioned default LED brightness. However, the following parameters can be customised:

- **Normal Mode**:

  - ➢ **Off Level**: values from 0 (default) to 255.

  - ➢ **On Level**: values from 0 to 255 (default).

- **Night Mode**: in case of being this mode necessary, the corresponding checkbox needs to be marked. This will incorporate some new parameters:

  - ➢ **Off Level**: values from 0 to 255 (1 by default).

  - ➢ **On Level**: values from 0 to 255 (8 by default).

In case of enabling the night mode, it is possible to select which of the two modes ("Normal" or "Night") will be active after an ETS Download (parameter "**LED Brightness Mode after ETS Download**").

In addition, two non-exclusive checkboxes (under "**LED Brightness Control**") let the integrator select how to switch from one mode to another:

- **1-Bit Object**: when marked, it will be possible to switch the mode by writing to a binary object ("**[General] LEDs: 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").

- **Scene Object**: 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.

Figure 29 General - LED Brightness: Night Mode.

2.4 WELCOME BACK OBJECT

Touch-MyDesign Plus can send a specific object (called "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 if the welcome back object is sent to the bus. Thus, if the user presses a button and this makes Touch-MyDesign Plus send the welcome back object, 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 as normally. Hence, the action corresponding to the button touch will be executed.

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

ETS PARAMETERISATION

After enabling "**Welcome Back Object**" in the general Configuration screen (see section 2.1), a new tab will be incorporated into the tree on the left.
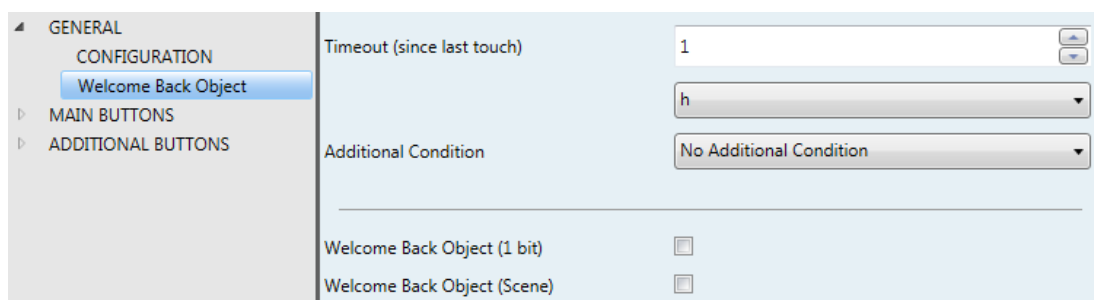
Figure 30 General - Welcome Back Object.

This screen contains the following parameters:

- **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 condition objects 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 condition objects 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 condition objects 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 condition objects 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 Object: B**") when the welcome back function is triggered and the condition (if any) evaluates to true. The desired value "B" (0 or 1) should to be entered 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**".

Figure 31 General - Welcome Back Object - Custom Configuration.

2.5 LOCKING THE TOUCH PANEL

The touch panel of Touch-MyDesign Plus can be optionally locked and unlocked in runtime 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.

ETS PARAMETERISATION

After enabling "**Touch Locking**" in the general Configuration screen (see section 2.1), a new tab will be incorporated into the tree on the left.
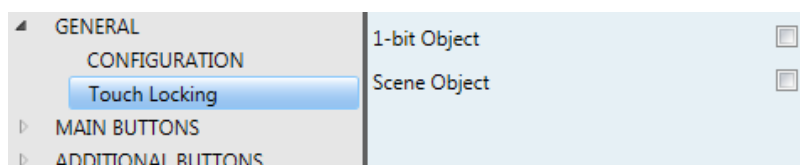

Figure 32 General - Touch Locking.

This screen comprises two simple checkboxes to select whether the touch panel lock/unlock should be done upon the reception of a one-bit value and/or a scene value:
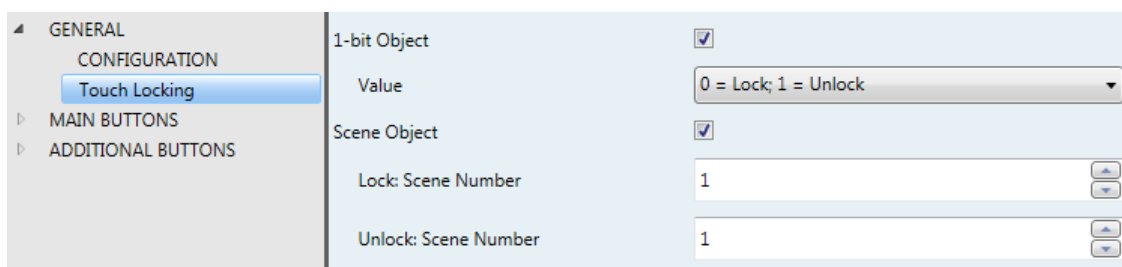
Figure 33 General - Touch Locking - Objects

- **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**".

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

2.6 SOUNDS

Apart from the behaviour of the LEDs, it is possible to make Touch-MyDesign Plus emit a **brief beep** as an acoustic feedback when an action is triggered after a button touch. 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. A specific object has also been included for externally enabling or disabling these **beeps**, provided that they had not been disabled in parameters.

With independence of the aforementioned button sounds, Touch-MyDesign Plus can also emit **doorbell** (a single beep on request) and **alarm** (a sequence of brief beeps with a higher pitch) sounds on the reception of certain values through specific objects. 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 integrator 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 2.1) 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 that parameter is 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.
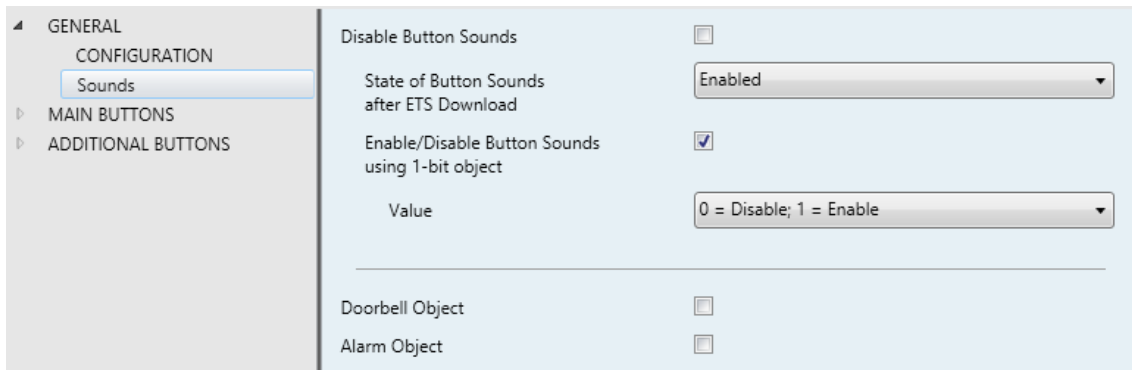
Figure 34 General – Sounds.

- **Disable Button Sounds**: enables or disables the beeps on the execution of an action after the user touches a button. If disabled (default option) the following parameters will also be available:

  - **State of Button Sounds 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.

  - **Enable/Disable Button Sounds Using 1-Bit Object**: enables or disables the option to stop / resume the button beeping function in runtime by writing to a specific object ("**[General] Sounds: Enable**"). The values (0 or 1) that will disable the beeps after the button touches and then re-enable them are parameterisable through "**Value**".

- **Doorbell Object**: 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 entered in "**Value**".

- **Alarm Object**: 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**".

Figure 35 General - Sounds: custom configuration.

2.7 INPUTS

Touch-MyDesign Plus incorporates **2 analogue/digital inputs**, each configurable as a:

- **Binary Input**, for the connection of a pushbutton or a switch/sensor.

- **Temperature Probe**, to connect a temperature sensor (model ZN1AC-NTC68 S/E/F from Zennio).

- **Motion Detector**, to connect a motion detector (models ZN1IO-DETEC-P and ZN1IO-DETEC-X from Zennio).

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

For detailed information about the functionality and the configuration of the related parameters, please refer to the following specific manuals, all of them available at the Zennio homepage (www.zennio.com):

- Binary Inputs in Touch-MyDesign Plus.

- Temperature Probe in Touch-MyDesign Plus.

- Motion Detector in Touch-MyDesign Plus.

2.8 THERMOSTAT

Touch-MyDesign Plus implements **one Zennio thermostat** which can be enabled and fully customised.

Please refer to the specific manual "**Zennio Thermostat in Touch-MyDesign Plus**" (available at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

2.9 INTERNAL TEMPERATURE SENSOR

Touch-MyDesign Plus 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 certain values.

Please refer to the specific manual "**Internal Temperature Sensor in Touch-MyDesign Plus**" (available at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

ANNEX I: 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.

- **"1st boot"** shows the cases where an object is assigned a certain value by the application program after a device download or a full reset. In case the value of such assignment can be parameterised, √ is shown in column "**P**". Objects showing a hyphen (-) are not assigned a particular value and therefore can be assumed to be initialised with the value "0" or with the corresponding updated value in case they depend on an external element (sensors, etc.). Moreover, if the object is sent (or is there an option to send it) to the bus (write or read requests) after a download or a device reset from ETS, the marks **(W)** or **(R)** will be shown, respectively for transmissions or read requests.

- **"Reboot"** shows the cases where an object is assigned a certain value by the application program after a bus power failure. In case the value of such assignment can be parameterised, √ is shown in column "**P**". Objects showing a hyphen (-) are not assigned a particular value and therefore can be assumed to maintain their previous value after the failure or with the corresponding updated value in case they depend on external elements (sensors, etc.). Moreover, if the object is sent (or is there an option to send it) to the bus (write or read requests) after a bus failure, the marks **(W)** or **(R)** will be shown, respectively for transmissions or read requests.

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	1stBoot	P	Reboot	P	Name	Function
1	1 Byte	E	C - - W -	DPT_SceneControl	0-63; 128-191	-		-		[General] Scene: Receive	0-63 (Run Scene 1-64)
2	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-		[General] Scene: Send	0-63 / 128-191 (Run or Save Scene 1-64)
3	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[General] LEDs: Brightness Mode	0 = Night; 1 = Normal
	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-		[General] LEDs: Brightness Mode	0 = Normal; 1 = Night
4	1 Bit	E	C - - W -	DPT_Enable	0/1	-		-		[General] Lock Touch	0 = Unlock; 1 = Lock
	1 Bit	E	C - - W -	DPT_Enable	0/1	-		-		[General] Lock Touch	0 = Lock; 1 = Unlock
5	1 Bit		C T - - -	DPT_Switch	0/1	-		-		[General] Welcome Back Object: "1"	1-bit Generic Control
	1 Bit		C T - - -	DPT_Switch	0/1	-		-		[General] Welcome Back Object: "0"	1-bit Generic Control
6 - 10	1 Bit	E	C - - W -	DPT_Bool	0/1	-		-		[General] Welcome Back Object: Xst Condition	1-bit Generic Control
11	1 Bit	E	C - - W -	DPT_Alarm	0/1	-		-		[General] Sounds: Alarm	0 = Start Alarm; 1 = Stop Alarm
	1 Bit	E	C - - W -	DPT_Alarm	0/1	-		-		[General] Sounds: Alarm	0 = Stop Alarm; 1 = Start Alarm
12	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[General] Sounds: Doorbell	0 = No Action; 1 = Doorbell
	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-		[General] Sounds: Doorbell	0 = Doorbell; 1 = No Action
13	1 Bit	E	C - - W -	DPT_Enable	0/1	-		-		[General] Sounds: Enable	0 = Enable; 1 = Disable
	1 Bit	E	C - - W -	DPT_Enable	0/1	-		-		[General] Sounds: Enable	0 = Disable; 1 = Enable
14, 24, 34, 44,	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-		[Xn] Binary Control: "0"	1-bit Generic Control

19, 29, 39, 49	1 Byte	E	C T - W -	DPT_Value_1_Ucount	0 - 255	-		-	[Xn] Two Objects, Short Press: 1-byte Value (Unsigned Int)	0 – 255
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Binary Control: "1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Binary Control: "0/1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Binary Control, Press/Release	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[X] Light On/Off	0 = Off; 1 = On
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Light On	0 = On
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Light Off	1 = On
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Light On/Off (Toggle)	0 = Off; 1 = On
	1 Bit	E	C T - W -	DPT_UpDown	0/1	-		-	[Xn] Move Shutter Up	0 = Up
	1 Bit	E	C T - W -	DPT_UpDown	0/1	-		-	[Xn] Move Shutter Down	1 = Down
	1 Bit	E	C T - W -	DPT_UpDown	0/1	-		-	[Xn] Move Shutter (Toggle Direction)	0 = Up; 1 = Down
	1 Bit	E	C T - W -	DPT_UpDown	0/1	-		-	[X] Move Shutter	0 = Up; 1 = Down
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[X] Two Objects: Short Press	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Two Objects, Short Press: "0"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Two Objects, Short Press: "1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Two Objects, Short Press: "0/1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[X] Binary Control	1-bit Generic Control
15, 25, 35, 45, 20, 30, 40, 50	1 Byte	E	C T - W -	DPT_Value_1_Ucount	0 - 255	-		-	[Xn] Two Objects, Long Press: 1-byte Value (Unsigned Int)	0 – 255
	1 Bit	E	C T - W -	DPT_Step	0/1	-		-	[X] Stop Shutter / Step	0 = Stop Shutter / Step Up; 1 = Stop Shutter / Step Down
	1 Bit	E	C T - W -	DPT_Step	0/1	-		-	[Xn] Stop Shutter / Step Up	0 = Stop Shutter / Step Up
	1 Bit	E	C T - W -	DPT_Step	0/1	-		-	[Xn] Stop Shutter / Step Down	1 = Stop Shutter / Step Down
	1 Bit	E	C T - W -	DPT_Step	0/1	-		-	[Xn] Stop Shutter / Step	0 = Stop Shutter / Step Up; 1 = Stop Shutter / Step Down
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[X] Two Objects: Long Press	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Two Objects, Long Press: "0"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Two Objects, Long Press: "1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Xn] Two Objects, Long Press: "0/1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Trigger	0/1	-		-	[X] Stop Shutter	0, 1 = Stop Shutter
	1 Bit	E	C T - W -	DPT_Trigger	0/1	-		-	[Xn] Stop Shutter	0 = Stop Shutter
	1 Bit	E	C T - W -	DPT_Trigger	0/1	-		-	[Xn] Stop Shutter	1 = Stop Shutter
	1 Bit	E	C T - W -	DPT_Trigger	0/1	-		-	[Xn] Stop Shutter	0, 1 = Stop Shutter
16, 26, 36, 46, 21, 31, 41, 51	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) 0x2 (Dec.by 50%) 0x3 (Dec.by 25%)	-		-	[X] Light Dimming	4-bit Dimming Control

#	Size		Flags	DPT	Range			Function	Description
					0x4 (Dec.by 12%) 0x5 (Dec.by 6%) 0x6 (Dec.by 3%) 0x7 (Dec.by 1%) 0x8 (Stop) 0x9 (Inc.by 100%) 0xA (Inc.by 50%) 0xB (Inc.by 25%) 0xC (Inc.by 12%) 0xD (Inc.by 6%) 0xE (Inc.by 3%) 0xF (Inc.by 1%)				
	4 Bit	E	C T - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-	-	[Xn] Light Dimming (Toggle)	4-bit Dimming Control
17, 27, 37, 47, 22, 32, 42, 52	2 Bytes	E	C T - W -	DPT_Value_2_Ucount	0 - 65535	-	-	[Xn] 2-byte Value (Unsigned Int)	0 – 65535
	1 Byte	E	C T - W -	DPT_Value_1_Ucount	0 - 255	-	-	[Xn] 1-byte Value (Unsigned Int)	0 – 255
	1 Byte	E	C T - W -	DPT_Scaling	0% - 100%	-	-	[Xn] 1-byte Value (Scaling)	0 – 100 %
	2 Bytes	E	C T - W -	9.xxx	-671088.64 - 670760.96	-	-	[Xn] 2-byte Value (Float)	-671 088.64 – 670 760.96
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	-	-	[X] Light Dimming (Status)	0 – 100 %
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	50%	50%	[Xn] Light Dimming (Status)	0 – 100 %
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	50%	50%	[Xn] Shutter Position	0 – 100 %
	1 Byte	E	C T - W -	DPT_Value_1_Count	-128 - 127	-	-	[Xn] 1-byte Value (Signed Int)	-128 - 127
	2 Bytes	E	C T - W -	DPT_Value_2_Count	-32768 - 32767	-	-	[Xn] 2-byte Value (Signed Int)	-32768 - 32767
18, 28, 38, 48, 23, 33, 43, 53	1 Bit	E	C - - W -	DPT_Switch	0/1	-	-	[X] LED On/Off	0 = Off; 1 = On
	1 Bit	E	C - - W -	DPT_Switch	0/1	-	-	[X] LED On/Off	0 = On; 1 = Off
	1 Bit	E	C - - W -	DPT_Switch	0/1	-	-	[Xn] LED On/Off	0 = Off; 1 = On
54, 59, 64, 69, 74	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Binary Control: "0"	1-bit Generic Control
	1 Byte	E	C T - W -	DPT_Value_1_Ucount	0 - 255	-	-	[Zn] Two Objects, Short Press: 1-byte Value (Unsigned Int)	0 – 255
	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Binary Control: "1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Binary Control: "0/1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Binary Control, Press/Release	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Light On/Off (Toggle)	0 = Off; 1 = On
	1 Bit	E	C T - W -	DPT_UpDown	0/1	-	-	[Zn] Move Shutter (Toggle Direction)	0 = Up; 1 = Down
	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Two Objects, Short Press: "0"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Two Objects, Short Press: "1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-	-	[Zn] Two Objects, Short Press: "0/1"	1-bit Generic Control
55, 60, 65, 70, 75	1 Byte	E	C T - W -	DPT_Value_1_Ucount	0 - 255	-	-	[Zn] Two Objects, Long Press: 1-byte Value (Unsigned Int)	0 – 255
	1 Bit	E	C T - W -	DPT_Step	0/1	-	-	[Zn] Stop Shutter / Step	0 = Stop Shutter / Step Up; 1 = Stop

										Shutter / Step Down
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Zn] Two Objects, Long Press: "0"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Zn] Two Objects, Long Press: "1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[Zn] Two Objects, Long Press: "0/1"	1-bit Generic Control
	1 Bit	E	C T - W -	DPT_Trigger	0/1	-		-	[Zn] Stop Shutter	0, 1 = Stop Shutter
56, 61, 66, 71, 76	4 Bit	E	C T - W -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-		-	[Zn] Light Dimming (Toggle)	4-bit Dimming Control
57, 62, 67, 72, 77	2 Bytes	E	C T - W -	DPT_Value_Temp	-273.00°C - 670760.00°C	-		-	[Z] Temperature Setpoint	-20.0°C – 150.0°C
	1 Byte	E	C T - W -	DPT_Value_1_Ucount	0 - 255	-		-	[Z] 1-byte Value (Unsigned Int)	0 – 255
	1 Byte	E	C T - W -	DPT_Scaling	0% - 100%	-		-	[Z] 1-byte Value (Scaling)	0 – 100%
	2 Bytes	E	C T - W -	DPT_Value_2_Ucount	0 - 65535	-		-	[Zn] 2-byte Value (Unsigned Int)	0 – 65535
	1 Byte	E	C T - W -	DPT_Value_1_Ucount	0 - 255	-		-	[Zn] 1-byte Value (Unsigned Int)	0 – 255
	1 Byte	E	C T - W -	DPT_Scaling	0% - 100%	-		-	[Zn] 1-byte Value (Scaling)	0 – 100 %
	2 Bytes	E	C T - W -	9.xxx	-671088.64 - 670760.96	-		-	[Zn] 2-byte Value (Float)	-671 088.64 – 670 760.96
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	-		-	[Zn] Light Dimming (Status)	0 – 100 %
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	-		-	[Zn] Shutter Position	0 – 100 %
	1 Byte	E	C T - W -	DPT_Value_1_Count	-128 - 127	-		-	[Zn] 1-byte Value (Signed Int)	-128 - 127
	2 Bytes	E	C T - W -	DPT_Value_2_Count	-32768 - 32767	-		-	[Zn] 2-byte Value (Signed Int)	-32768 - 32767
58, 63, 68, 73, 78	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-	[Zn] LED On/Off	0 = On; 1 = Off
	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-	[Zn] LED On/Off	0 = Off; 1 = On
79, 83	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00°C – 670760.00°C	-		-	[In] Current Temperature	Temperature sensor value
80, 84	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-	[In] Overcooling	0 = No Alarm; 1 = Alarm
81, 85	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-	[In] Overheating	0 = No Alarm; 1 = Alarm
82, 86	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-	[In] Probe Error	0 = No Alarm; 1 = Alarm
87	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00°C – 670760.00°C	-		-	[Internal Sensor] Current Temperature	Temperature sensor value
88	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-	[Internal Sensor] Overcooling	0 = No Alarm; 1 = Alarm
89	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-	[Internal Sensor] Overheating	0 = No Alarm; 1 = Alarm
91, 97	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-	[In] Input Lock	1 = Locked; 0 = Unlocked
92, 98	1 Bit		C T - - -	DPT_Switch	0/1	-		-	[In] [Short Press] 0	Sending of 0
	1 Bit		C T - - -	DPT_Switch	0/1	-		-	[In] [Short Press] 1	Sending of 1
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[In] [Short Press] 0/1 Switching	Switching 0/1
	1 Bit		C T - - -	DPT_UpDown	0/1	-		-	[In] [Short Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		C T - - -	DPT_UpDown	0/1	-		-	[In] [Short Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		C T - - -	DPT_UpDown	0/1	-		-	[In] [Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)

	1 Bit		C T - - -	DPT_Step	0/1	-		-	[In] [Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)
	1 Bit		C T - - -	DPT_Step	0/1	-		-	[In] [Short Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step down)
	1 Bit		C T - - -	DPT_Step	0/1	-		-	[In] [Short Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-		-	[In] [Short Press] Brighter	Increase Brightness
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-		-	[In] [Short Press] Darker	Decrease Brightness
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-		-	[In] [Short Press] Brighter/Darker	Switch Bright/Dark
	1 Bit		C T - - -	DPT_Switch	0/1	-		-	[In] [Short Press] Dimmer ON	Sending of 1 (ON)
	1 Bit		C T - - -	DPT_Switch	0/1	-		-	[In] [Short Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[In] [Short Press] Dimmer ON/OFF	Switching 0/1
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-	[In] [Short Press] Run Scene	Sending of 0 - 63
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-	[In] [Short Press] Save Scene	Sending of 128 - 191
	1 Bit	E/S	C T R W -	DPT_Switch	0/1	-		-	[In] [Switch/Sensor] Edge	Sending of 0 or 1
	1 Byte		C T - - -	DPT_Value_1_Ucount	0 - 255	-		-	[In] [Short Press] Constant Value (Integer)	0 - 255
	1 Byte		C T - - -	DPT_Scaling	0% - 100%	-		-	[In] [Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		C T - - -	DPT_Value_2_Ucount	0 - 65535	-		-	[In] [Short Press] Constant Value (Integer)	0 - 65535
	2 Bytes		C T - - -	9.xxx	-671088.64 - 670760.96	-		-	[In] [Short Press] Constant Value (float)	Float value
93, 99	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	-		-	[In] [Short Press] Shutter Status (input)	0% = Top; 100% = Bottom
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	-		-	[In] [Short Press] Dimming Status (input)	0% - 100%
94, 100	1 Bit		C T - - -	DPT_Switch	0/1	-		-	[In] [Long Press] 0	Sending of 0
	1 Bit		C T - - -	DPT_Switch	0/1	-		-	[In] [Long Press] 1	Sending of 1
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-	[In] [Long Press] 0/1 Switching	Switching 0/1
	1 Bit		C T - - -	DPT_UpDown	0/1	-		-	[In] [Long Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		C T - - -	DPT_UpDown	0/1	-		-	[In] [Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		C T - - -	DPT_UpDown	0/1	-		-	[In] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		C T - - -	DPT_Step	0/1	-		-	[In] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)

	Size		Flags	Data Type (DPT)	Functional Range					Function	Description
	1 Bit		C T - - -	DPT_Step	0/1	-		-		[In] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step down)
	1 Bit		C T - - -	DPT_Step	0/1	-		-		[In] [Long Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-		-		[In] [Long Press] Brighter	Long Pr. -> Brighter; Release -> Stop
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-		-		[In] [Long Press] Darker	Long Pr. -> Darker; Release -> Stop
	4 Bit		C T - - -	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec.by 100%) ... 0x8 (Stop) 0x9 (Inc.by 100%) ... 0xF (Inc.by 1%)	-		-		[In] [Long Press] Brighter/Darker	Long Pr. -> Brighter/Darker; Release -> Stop
	1 Bit		C T - - -	DPT_Switch	0/1	-		-		[In] [Long Press] Dimmer ON	Sending of 1 (ON)
	1 Bit		C T - - -	DPT_Switch	0/1	-		-		[In] [Long Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	E	C T - W -	DPT_Switch	0/1	-		-		[In] [Long Press] Dimmer ON/OFF	Switching 0/1
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-		[In] [Long Press] Run Scene	Sending of 0 - 63
	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-		[In] [Long Press] Save Scene	Sending of 128 - 191
	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[In] [Switch/Sensor] Alarm: Breakdown or sabotage	1 = Alarm; 0 = No Alarm
	2 Bytes		C T - - -	9.xxx	-671088.64 - 670760.96	-		-		[In] [Long Press] Constant Value (float)	Float value
	2 Bytes		C T - - -	DPT_Value_2_Ucount	0 - 65535	-		-		[In] [Long Press] Constant Value (Integer)	0 - 65535
	1 Byte		C T - - -	DPT_Scaling	0% - 100%	-		-		[In] [Long Press] Constant Value (Percentage)	0% - 100%
	1 Byte		C T - - -	DPT_Value_1_Ucount	0 - 255	-		-		[In] [Long Press] Constant Value (Integer)	0 - 255
95, 101	1 Bit		C T - - -	DPT_Trigger	0/1	-		-		[In] [Long Press/Release] Stop Shutter	Release -> Stop Shutter
96, 102	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	50%		-		[In] [Long Press] Dimming Status (input)	0% - 100%
	1 Byte	E	C - - W -	DPT_Scaling	0% - 100%	50%		-		[In] [Long Press] Shutter Status (input)	0% = Top; 100% = Bottom
103	1 Byte	E	C - - W -	DPT_SceneControl	0-63; 128-191	-		-		[Motion Sensor] Scene Input	Scene Value
104	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	-		-		[Motion Sensor] Scene Output	Scene Value
105, 129	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[In] Luminosity	0-100%
106, 130	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[In] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
107, 131	1 Bit	S	C T R - -	DPT_Alarm	0/1	-		-		[In] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
108, 132	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-		[In] Presence State (Scaling)	0-100%
109, 133	1 Byte	S	C T R - -	DPT_HVACMode	1=Comfort 2=Standby	-		-		[In] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection

					3=Economy 4=Protection					
110, 134	1 Bit	S	C T R - -	DPT_Occupancy	0/1	-		-	[In] Presence State (Binary)	Binary Value
	1 Bit	S	C T R - -	DPT_Trigger	0/1	-		-	[In] Presence: Slave Output	1 = Motion Detected
111, 135	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-	[In] Presence Trigger	Binary Value to Trigger the Presence Detection
112, 136	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-	[In] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
113, 137	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-	[In] External Motion Detection	0 = Nothing; 1 = Motion detected by an external sensor
114, 119, 124, 138, 143, 148	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-	[In][Cm] Detection State (Scaling)	0-100%
115, 120, 125, 139, 144, 149	1 Byte	S	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Protection	-		-	[In][Cm Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
116, 121, 126, 140, 145, 150	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[In][Cm] Detection State (Binary)	Binary Value
117, 122, 127, 141, 146, 151	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-	[In][Cm] Channel Lock	According to parameters
118, 123, 128, 142, 147, 152	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-	[In][Cm] Force State	0 = No Detection; 1 = Detection
153	1 Byte	E	C - - W -	DPT_SceneControl	0-63; 128-191	-		-	[Thermostat] Scene Input	Scene Value
154	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00°C – 670760.00°C	25.0		-	[T1] Temperature Source 1	External Sensor Temperature
155	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00°C – 670760.00°C	25.0		-	[T1] Temperature Source 2	External Sensor Temperature
156	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00°C – 670760.00°C	-		-	[T1] Effective Temperature	Effective Control Temperature
157	1 Byte	E	C - - W -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Protection	-	√	-	[T1] Special Mode	1-byte HVAC Mode
158	1 Bit	E	C - - W -	DPT_Trigger	0/1	-	√	-	[T1] Special Mode: comfort	0 = Nothing; 1 = Trigger
	1 Bit	E	C - - W -	DPT_Switch	0/1	-	√	-	[T1] Special Mode: comfort	0 = Off; 1 = On
159	1 Bit	E	C - - W -	DPT_Trigger	0/1	-	√	-	[T1] Special Mode: standby	0 = Nothing; 1 = Trigger
	1 Bit	E	C - - W -	DPT_Switch	0/1	-	√	-	[T1] Special Mode: standby	0 = Off; 1 = On
160	1 Bit	E	C - - W -	DPT_Trigger	0/1	1	√	-	[T1] Special Mode: economy	0 = Nothing; 1 = Trigger
	1 Bit	E	C - - W -	DPT_Switch	0/1	1	√	-	[T1] Special Mode: economy	0 = Off; 1 = On
161	1 Bit	E	C - - W -	DPT_Trigger	0/1	-	√	-	[T1] Special Mode: protection	0 = Nothing; 1 = Trigger
	1 Bit	E	C - - W -	DPT_Switch	0/1	-	√	-	[T1] Special Mode: protection	0 = Off; 1 = On
162	1 Bit	E	C - - W -	DPT_Window_Door	0/1	-		-	[T1] Window Status (input)	0 = Closed; 1 = Open
163	1 Bit	E	C - - W -	DPT_Trigger	0/1	-		-	[T1] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
164	1 Byte	S	C T R - -	DPT_HVACMode	1=Comfort 2=Standby	3 W	√	- W	[T1] Special Mode Status	1-byte HVAC Mode

					3=Economy 4=Protection					
165	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00°C – 670760.00°C	-		-	[T1] Setpoint	Thermostat Setpoint Input
	2 Bytes	E	C - - W -	DPT_Value_Temp	-273.00°C – 670760.00°C	-		-	[T1] Basic Setpoint	Reference Setpoint
166	1 Bit	E	C - - W -	DPT_Step	0/1	-		-	[T1] Setpoint Step	0 = -0.5°C; 1 = +0.5°C
167	2 Bytes	E	C - - W -	DPT_Value_Tempd	-273.00°C – 670760.00°C	-		-	[T1] Setpoint Offset	Float Offset Value
168	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00°C – 670760.00°C	- W	√	- W	[T1] Setpoint Status	Current Setpoint
169	2 Bytes	S	C T R - -	DPT_Value_Temp	-273.00°C – 670760.00°C	- W	√	- W	[T1] Basic Setpoint Status	Current Basic Setpoint
170	2 Bytes	S	C T R - -	DPT_Value_Tempd	-273.00°C – 670760.00°C	- W	√	- W	[T1] Setpoint Offset Status	Current Setpoint Offset
171	1 Bit	E	C - - W -	DPT_Reset	0/1	-		-	[T1] Setpoint Reset	Reset Setpoint to Default
	1 Bit	E	C - - W -	DPT_Reset	0/1	-		-	[T1] Offset Reset	Reset offset
172	1 Bit	E	C - - W -	DPT_Heat_Cool	0/1	-		-	[T1] Mode	0 = Cool; 1 = Heat
173	1 Bit	S	C T R - -	DPT_Heat_Cool	0/1	- W	√	- W	[T1] Mode Status	0 = Cool; 1 = Heat
174	1 Bit	E	C - - W -	DPT_Switch	0/1	-		-	[T1] On/Off	0 = Off; 1 = On
175	1 Bit	S	C T R - -	DPT_Switch	0/1	- W	√	- W	√	[T1] On/Off Status	0 = Off; 1 = On
176	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-	[T1] Control Variable (Cool)	PI Control (Continuous)
177	1 Byte	S	C T R - -	DPT_Scaling	0% - 100%	-		-	[T1] Control Variable (Heat)	PI Control (Continuous)
178	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] Control Variable (Cool)	2-Point Control
	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] Control Variable (Cool)	PI Control (PWM)
179	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] Control Variable (Heat)	2-Point Control
	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] Control Variable (Heat)	PI Control (PWM)
180	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] Additional Cool	Temp >= (Setpoint+Band) => "1"
181	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] Additional Heat	Temp <= (Setpoint-Band) => "1"
182	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] PI State (Cool)	0 = PI signal 0%; 1 = PI signal greater than 0%
183	1 Bit	S	C T R - -	DPT_Switch	0/1	-		-	[T1] PI State (Heat)	0 = PI signal 0%; 1 = PI signal greater than 0%

Join and send us your inquiries
about Zennio devices:

http://zennioenglish.zendesk.com