



Flat Display

**Capacitive touch panel
with 5 buttons and 2.4" display**

ZVI-FD

Application Programme Version: [2.0]
User Manual Version: [2.0]_a

CONTENTS

Contents	2
Document updates	3
1 Introduction	4
1.1 Flat Display.....	4
1.2 Installation	6
1.3 Start-Up and Power Loss	7
2 Configuration	8
2.1 General	8
2.1.1 Configuration	8
2.1.2 Advanced Configuration	19
2.2 Buttons	25
2.2.1 Configuration	25
2.2.2 Disable	27
2.2.3 Individual	28
2.2.4 Pair.....	43
2.3 Display	58
2.3.1 Configuration	58
2.3.2 Disabled	61
2.3.3 Single	62
2.3.4 Double	72
2.3.5 Setpoint	73
2.3.6 Fan	75
2.3.7 Borders	76
2.4 Inputs.....	79
2.4.1 Binary Input.....	79
2.4.2 Temperature Probe	79
2.4.3 Motion Detector	79
2.5 Thermostat	81
ANNEX I. LED Illumination Modes.....	82
ANNEX II. Communication Objects	85

DOCUMENT UPDATES

Version	Changes	Page(s)
[2.0]_a	Changes in the application program: <ul style="list-style-type: none">• New page style: Thermostat.• Optimization of the Luminosity and Proximity Sensor module.	-
	New page style: Thermostat	59 - 62
[1.1]_a	Changes in the application program: <ul style="list-style-type: none">• Optimization of the Luminosity and Proximity Sensor, Heartbeat and Temperature Probe module.	-

1 INTRODUCTION

1.1 FLAT DISPLAY

Flat Display is a KNX **multifunction capacitive touch switch with an analogue display** from Zennio with proximity sensor, luminosity sensor and backlighted buttons and display.

It is offered at **a reduced size and weight**, with **five capacitive touch buttons** on the bottom, with **LED backlight** to confirm the press of the buttons or showing states, and **a display** on the top.

Flat Display is a fully customisable solution for the room control where the user needs to control climate systems, lighting, blinds, scenes, etc.

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

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

The most outstanding features of Flat Display are:

- **2.4-inch (2.4")** back-lit OLED display with a resolution of **128 x 64** pixels.
- **Fully customisable** design of the front glass.
- **5 touch buttons** which can operate as individual or pair controls.
- **Light indicator (LED)** for every button.
- **Buzzer** for an audible acknowledgement of user actions (with the possibility of disabling it either by parameter or by object).
- Possibility of **locking / unlocking the touch panel** through binary orders or scenes, and of setting a timed/automatic locking of the device (**cleaning function**).

- **Welcome greeting on the display** and **Welcome Back object** (binary or scene).
- **Screensaver** function.
- Up to **5 switchable languages** for the displayed texts.
- **Two analogue/digital inputs** (for motion detectors, temperature probes, additional switches, etc.).
- **Thermostat** function.
- Built-in **temperature sensor**.
- **Celsius and Fahrenheit** temperature scales for the on-screen indicators, being possible to select them in parameters or through communication object.
- **Ambient luminosity sensor** for brightness automatic adjustment.
- **Proximity sensor** for quick start.
- **Heartbeat** or periodical “still-alive” notification.

1.2 INSTALLATION

Figure 1 shows the connection outline of the device:

1. Internal Temperature Probe.
2. KNX connector
3. Prog./Test LED.
4. Prog./Test Button.
5. Attachment Clips.
6. Inputs Connector
7. Display.
8. Touch Area.
9. Luminosity and Proximity Sensor.

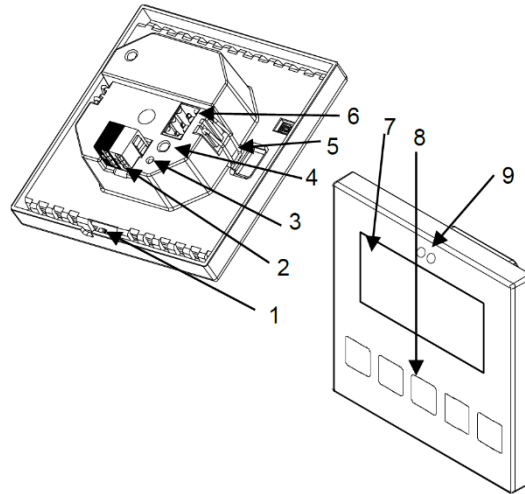


Figure 1. Schematic diagram.

Flat Display is connected to the KNX bus through the built-in terminal (2). An external DC power supply is not needed.

A short press on the **Prog./Test button** (4) will make the device enter the programming mode. The **Prog./Test LED** (3) will light in red. On the contrary, if this button is held while plugging the device into the KNX bus, the device will enter the **Safe Mode**. In such case, the programming LED will then blink in red.

For detailed information about the technical features of Flat Display, as well as on security and installation procedures, please refer to the **Datasheet**, bundled within the device packaging of also available at www.zennio.com.

1.3 START-UP AND POWER LOSS

After download or device reset it is necessary **to wait for about 2 minutes without performing any action** in order to make it possible a proper calibration of:

- Proximity sensor.
- Luminosity sensor.
- Button presses.

For a correct calibration of the proximity and brightness sensors it is recommended not to approach less than 50 cm from the device during this time and to avoid that the light strikes directly.

2 CONFIGURATION

After importing the corresponding database in ETS and adding the device into the topology of the project, the configuration process begins by entering the Parameters tab of the device.

2.1 GENERAL

In order to allow the device to perform the desired functions, a number of options must be parameterized, either related to its **general behaviour** (screensaver, sounds, lock procedure of the touch panel...) or to **advanced features** (cleaning function, welcome back object, welcome greeting...).

2.1.1 CONFIGURATION

In the "Configuration" tab, the general settings are displayed.

ETS PARAMETERISATION

Buttons	<input checked="" type="checkbox"/>
Display	<input checked="" type="checkbox"/>
Inputs	<input type="checkbox"/>
Thermostat	<input type="checkbox"/>
Heartbeat (Periodic Alive Notification)	<input type="checkbox"/>
<hr/>	
Time and Date Request Delay	<input type="checkbox"/>
Update Objects	Disabled
<hr/>	
Screensaver	<input type="checkbox"/>
Run Button Action on Screensaver Exit or Alarm Mute	<input type="checkbox"/>
Time to Consider Inactivity	1
	min
Touch Locking	<input type="checkbox"/>
Proximity Sensor	<input checked="" type="checkbox"/>
Ambient Luminosity Sensor	<input type="checkbox"/>
Sounds	<input checked="" type="radio"/> Default <input type="radio"/> Custom
Advanced Configuration	<input type="checkbox"/>

Figure 2. Configuration

This tab shows the following parameters:

- **Buttons** [[enabled](#)]¹: read-only parameter to make it evident that the “Buttons” tab is always enabled in the tab tree on the left. See section 2.2 for details.
- **Display** [[enabled](#)]: read-only parameter to make it evident that the “Display” tab is always enabled in the tab tree on the left. See section 2.3 for details.
- **Inputs** [[disabled/enabled](#)]: enables or disables the “Inputs” tab in the tree on the left, depending on whether the device will or will not be connected any external accessories. See section 2.4 for details
- **Thermostat** [[disabled/enabled](#)]: enables or disables the “Thermostat” tab in the tree on the left. See section 2.5 for details.
- **Heartbeat (Periodic Alive Notification)** [[disabled/enabled](#)]: incorporates a one-bit object to the project (“**[Heartbeat] Object to Send ‘1’**”) that will be sent periodically with value “1” to notify that the device is still working (*still alive*).

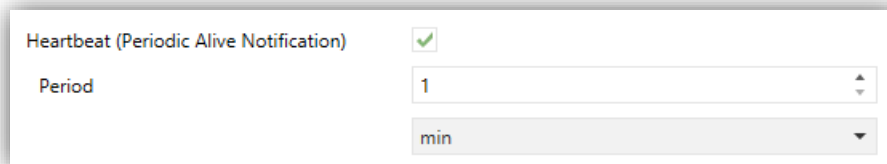


Figure 3. Heartbeat

Note: *the first sending after download or bus failure takes place with a delay of up to 255 seconds, to prevent bus overload. The following sendings march the period set.*

- **Time and Date Request Delay** [[disabled/enabled](#)]: sets a sending **delay** [[1...65535](#)] [[s/min/h](#)] for the date and time request when the device starts up.
- **Update Objects:** enables the sending of read requests to update status objects and indicators. There are four options available, some of them with a configurable delay:

¹ The default values of each parameter will be highlighted in blue in this document, as follows: [[default/rest of options](#)].

- [\[Disabled\]](#): no read request, therefore objects are not updated.
 - [\[After Programming\]](#): read requests are sent after a complete or partial download after the parameterised **delay** [\[1...10...65535\]](#) [\[s/min/h\]](#).
 - [\[After Reset\]](#): read request are sent when a reset occurs (bus failure, the Reset Device ETS option), after the parameterised **delay** [\[1...10...65535\]](#) [\[s/min/h\]](#).
 - [\[After Programming and Reset\]](#): combination of the two above options.
- **Screensaver** [\[disabled/enabled\]](#): enables or disables the “Screensaver” tab in the tree on the left. See section 2.1.1.4 for details.
 - **Run Button Action on Screensaver Exit or Alarm Mute** [\[disabled/enabled\]](#): sets whether, touching a button while the screensaver or an alarm is active should cause the execution of the button action, apart from hiding the screensaver or muting the alarm.
 - **Time to Consider Inactivity** [\[1...30...255\]](#) [\[s/min/h\]](#): time that must elapse since the last press and/or proximity detection to consider inactivity state. Then the display and the button LEDs backlight will dim.
 - **Touch Locking** [\[disabled/enabled\]](#): enables or disables the “Touch locking” tab in the tree on the left. See section 2.1.1.5 for details.
 - **Proximity Sensor** [\[disabled/enabled\]](#): enables the proximity sensor. This functionality permits “waking up” the device display when detecting presence.

Please refer to the specific manual “**Proximity and Luminosity Sensor**” (available in the Flat Display product section at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.
 - **Ambient Luminosity Sensor** [\[disabled/enabled\]](#): enables or disables the ambient luminosity sensor. When enabled, a new tab is added in the tree on the left. See section 2.1.1.6 for details.
 - **Sounds** [\[Default / Custom\]](#): sets whether the sound functions (button beeps, alarm and doorbell) should work according to the pre-defined configuration or to a user-defined configuration (see section 2.1.1.7).

- **Advanced Configuration** [[disabled/enabled](#)]: enables or disables the “Advanced” tab in the tree on the left. See section 2.1.2 for details.

The project topology shows the following objects by default:

- **[General] Time of Day**: 3-byte object for setting the internal time of the device, for example, by linking it to a KNX clock. This object also allows read requests, so the current time of the device can be checked. It is also automatically sent after time changes made by the user from the screen itself.

Important: *the time of day must be set through the bus by means of an external reference. In the absence of power, the clock will become outdated.*

- **[General] Date**: 3-byte object for setting the internal date of the device, for example, by linking it to a KNX clock. This object also allows read requests, so the current date of the device can be checked. It is also automatically sent after date changes made by the user from the screen itself.
- **[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)
- **[General] Activity**: 1-bit object to force activity or inactivity state in the device. See “**Proximity and luminosity Sensor**” and “**Brightness**” user manuals (available in the Flat Display product section at the Zennio homepage, www.zennio.com) for more information.
- **[General] Display - Brightness**: 1-byte percentage object for changing the display brightness level.
- **[General] Proximity Sensor**, **[General] External Proximity Detection** and **[General] Proximity Detection**: 1-bit object whose functionality is tied to the proximity sensor. For further information, please refer to the user manual “**Proximity and Luminosity Sensor**” (available in the Flat Display product section at the Zennio homepage, www.zennio.com).
- **[General] Translations - Select Language**: 1-byte and 2-byte objects for changing the language showed in the screen when receiving a value through the bus (see section 2.1.1.1).

- **[General] Translations - Main Language:** 1-bit object that, after the reception of the value “1” from the bus, will load the main language (see section 2.1.1.1).
- **[Internal Temp. Probe] Current Temperature:** 2-byte object through which the value of the current measurement of the built-in sensor will be sent to the bus, according to the parameterisation (see section 2.1.1.3).

2.1.1.1 TRANSLATIONS

Texts shown on the screen can be translated into up to **five different languages**.

Texts of the box titles are entered by parameter in the corresponding configuration tab. For each language enabled, an additional textbox is displayed to enter the translation.

Note: *depending on the space occupied by the character on the screen, the full text entered may not be displayed.*

Switching from language to another can be done through two types of communication objects:

- **Up to five 1-bit objects**, one for each language. If this is enabled, when a “1” is received through any of this, the corresponding language is activated in the device.
- **A 1-byte scene object.** The expected values in this object are fixed, from 0 to 4, to select the language. If the corresponding language to the received value is not enabled, the texts will be displayed in the main. If an out-of-range value is received, it does not change the active language.
- **A 2-byte ASCII object.** The expected values for this object are two characters of the ASCII code, corresponding to ISO 639-1. If the received character pair does not correspond to any enabled language, the texts will be represented in the main language.

Note: *please refer to http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes for a table with the language codes.*

ETS PARAMETERISATION

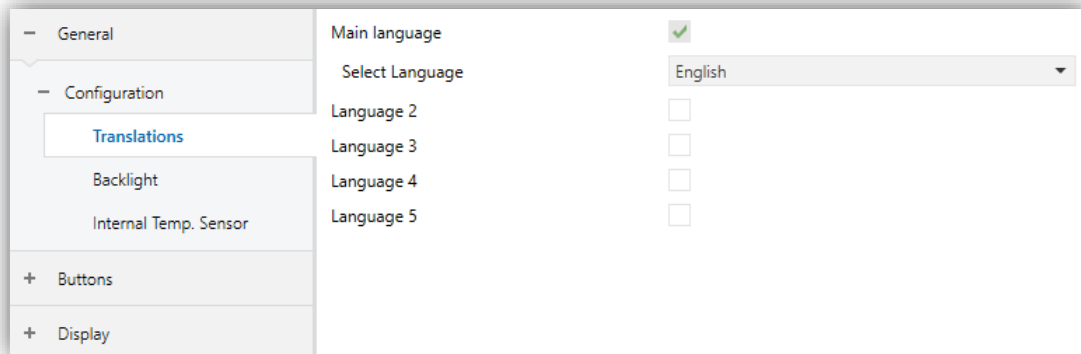


Figure 4. Configuration - Translations

- **Main language** [[enabled](#)]: read-only parameter to make it evident that the main language is always enabled.
 - **Select language:** list of the available languages.
- **Language X** [[enabled/disabled](#)]: enables the additional language X.
 - **Select language:** list of available languages to select the language X.

Only the Main language is enabled by default.

While Translations stays enabled, the following objects are visible:

- “[General] Translations – Select language” (1-byte).
- “[General] Translations – Main language” (1-bit).
- “[General] Translations – Select language” (2-byte).

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

- “[General] Translations – Language X” (1-bit).

These objects work accordingly to the behaviour mentioned above.

2.1.1.2 BACKLIGHT

Flat Display allows managing the brightness of the display and the LED according to two operating modes: normal mode and night mode.

Note: *Contrast is not a configurable feature in this device.*

Please refer to the specific manual “**Brightness**” (available in the Flat Display product section at the Zennio website, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

2.1.1.3 TEMPERATURE SENSOR

Flat Display is equipped with **one internal temperature probe** which can monitor the ambient temperature of the room, thus making the device capable of reporting it to the KNX bus and of triggering certain actions when the temperature reaches specific values.

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

2.1.1.4 SCREENSAVER

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

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

To exit the screen saver, press on the screen or approach the screen if the proximity sensor is activated. It can be set by parameter if, while the screen saver is active, pressing the touch panel will only cause the screen saver to exit, or if the action corresponding to the button pressed must also be executed (see section 2.1.1).

Notes:

- *When showing the welcome greeting (see section 2.1.2.2), screensaver will not become active.*
- *When screensaver is active and a pop-up shows up (see touch locking 2.1.1.5, cleaning function 2.1.2.1 or welcome greeting 2.1.2.2) this pop-up will overlay the screensaver.*

PARAMETRIZACIÓN ETS

After enabling "**Screensavers**" from the "Configuration" tab (section 2.1.1), a new tab is added to the left tree:

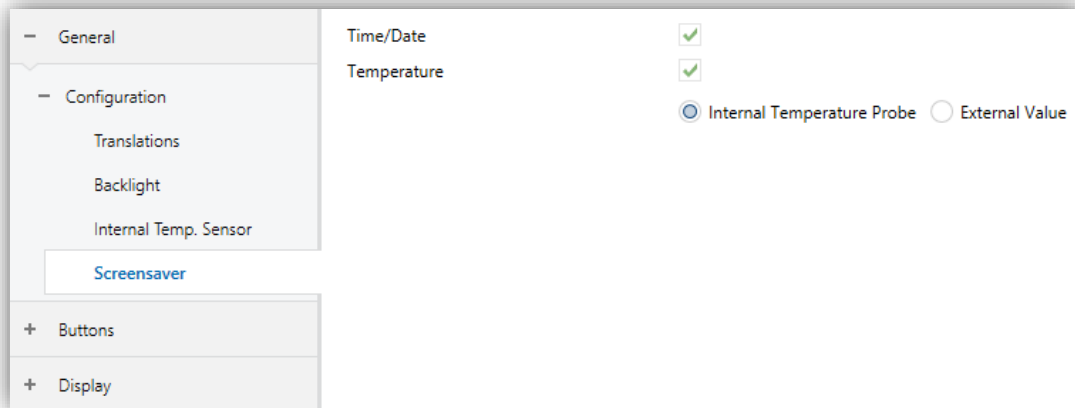


Figure 5. Configuration – Screensaver

- **Time/Date** [[enabled/disabled](#)]: determines whether to display the current time and date or not.
- **Temperature** [[enabled/disabled](#)]: determines whether to display the current temperature or not. The source of the temperature value can be selected [[Internal Temperature Probe/External Value](#)]. Selecting "External value" will enable a new communication object "**[General] External temperature**", whereby the device can receive the values from the bus.

2.1.1.5 TOUCH LOCKING

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

While locked, user presses on the touch buttons will be ignored: no actions will be performed when the user presses on any of the controls. However, if configured, a message will be shown on the display for three seconds if the user touches a button during the lock state.

ETS PARAMETERISATION

After enabling "Touch Locking" from the "Configuration" tab (section 2.1.1), a new tab is added to the left tree:

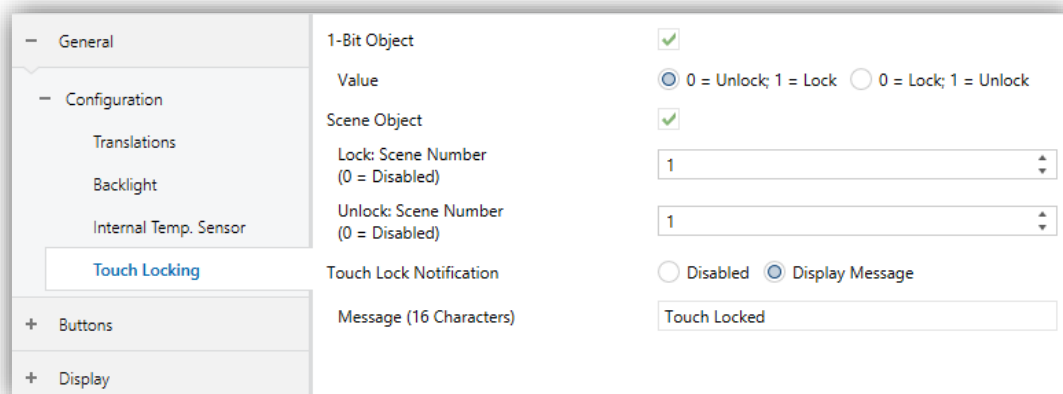


Figure 6. Configuration - Touch Locking

In this tab the blocking of the presses on the display can be configured.

- **1-Bit Object** [*disabled/enabled*]: enables the 1-bit object "[General] Touch Locking" to trigger the touch lock.
 - **Value** [*0 = Unlock, 1 = Lock / 0 = Lock, 1 = Unlock*]: parameter to select which value should trigger which action when received through the indicated object.
- **Scene Object** [*disabled/enabled*]: enables the touch locking and unlocking when receiving the configured scene value through the object ("[General] Scene: receive").
 - **Lock: Scene Number (0 = Disabled)** [*0/1...64*]: scene number that locks the touch.
 - **Unlock: Scene Number (0 = Disabled)** [*0/1...64*]: scene number that unlocks the touch.
- **Touch Lock Notification** [*Disabled / Display Message*]: sets whether to display a message on the screen or not when the device is locked and the user attempts to touch a button. When the first one is selected, one textbox (**Message**) appears to enter the desired message.

2.1.1.6 AMBIENT LUMINOSITY SENSOR

Flat Display incorporates a **luminosity sensor** to receive and monitor ambient brightness measurement.

Please refer to the specific manual “**Luminosity and Proximity Sensor**” (available in the Flat Display product section at the Zennio homepage, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

2.1.1.7 SOUNDS

Flat Display emits **3 types of sounds**, depending on the action performed:

- **Press Confirmation:** short beep indicating that the user has pressed a button. This only applies to step controls, i.e., controls that walk through a certain range of values and that do not send a value after every touch, but only the final value after the last press. For this action, the user can choose between two different sounds.
- **Sending Confirmation:** a slightly longer and sharper beep than the previous one. It indicates the sending of an object to the bus as a result of a press.
- **Alarm:** sharp and longer beep than the previous one, high intensity, which is typically used as alarm or bell.

The range of sounds emitted when performing these actions will be different depending on the sound type selected.

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.

Note: *Under no circumstances the alarm or the ring tone will be muted.*

ETS PARAMETERISATION

After enabling the “Custom” configuration of **Sounds** from “Configuration” screen (see section 2.1.1), a new tab will be incorporated into the tree on the left.

The initial configuration of this screen is equivalent to the default setting. However, the following parameters can be customized:

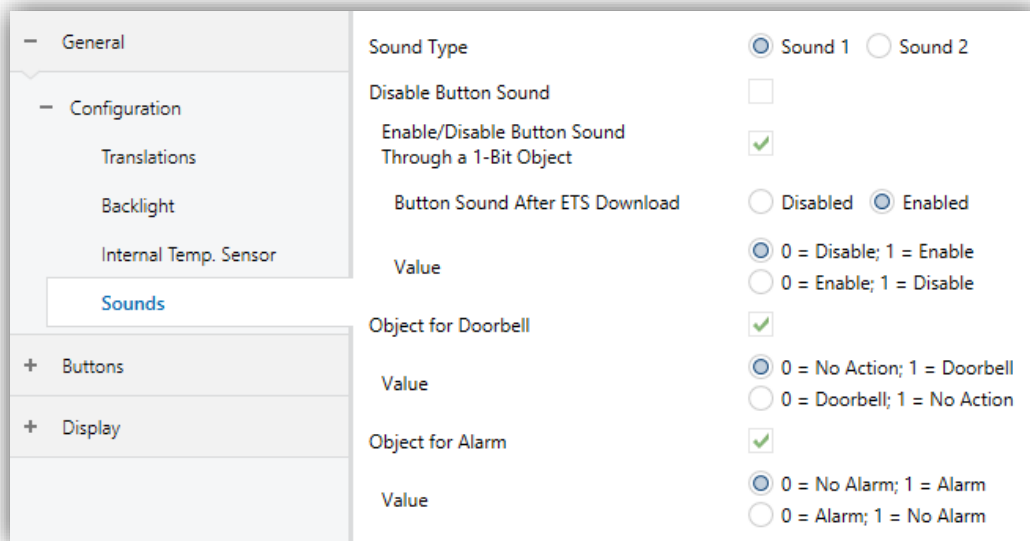


Figure 7. Configuration - Sounds.

- **Sound Type** [[Sound 1](#) / [Sound 2](#)]: parameter to select which sounds range incorporates the device.
- **Disable Button Sounds** [[disabled/enabled](#)]: enables or disables the acoustic signals when the actions derived from the button clicks are executed. If enabled, the following parameters will appear:
 - **Enable/Disable Button Sound Through a 1-bit Object** [[disabled/enabled](#)]: makes it possible to disable / resume the button beeping function in runtime by writing to a specific object (“**[General] Sounds – Disabling Button Sound**”).
 - **Button Sound After ETS Download** [[enabled/disabled](#)]: sets whether the button beeping function should start up enabled (default option) or disabled after an ETS download.
 - **Value** [[0 = Disabled, 1 = Enabled](#) / [0 = Enabled, 1 = Disabled](#)]: parameter to select which value should trigger which action when received through the indicated object

- **Object for Doorbell** [[disabled/enabled](#)]: enables or disables the doorbell function. If enabled, a specific object (“**[General] Sounds - Doorbell**”) will be included into the project topology.
 - **Value** [[0 = No Action, 1 = Doorbell](#) / [0 = Doorbell, 1 = No Action](#)]: parameter to select which value should trigger which action when received through the indicated object.
- **Object for Alarm** [[disabled/enabled](#)]: enables or disables the doorbell function. If enabled, a specific object (“**[General] Sounds - Alarm**”) will be included into the project topology.
 - **Value** [[0 = No Alarm, 1 = Alarm](#) / [0 = Alarm, 1 = No Alarm](#)]: parameter to select which value should trigger which action when received through the indicated object.

2.1.2 ADVANCED CONFIGURATION

Tab for the parameterisation of some advanced functions is shown in ETS if enabled from the “Configuration” tab. These functions are explained next.

ETS PARAMETERISATION

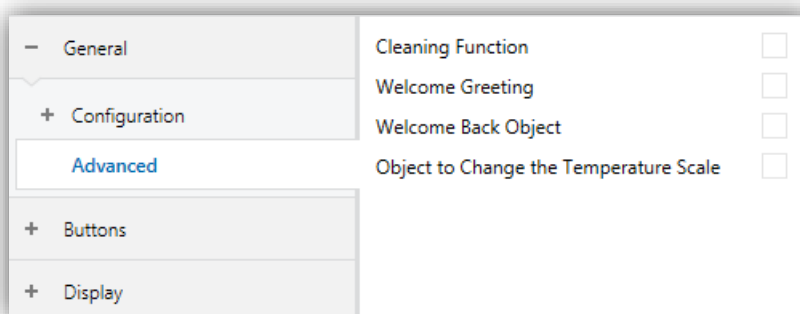


Figure 8. Advanced.

- **Cleaning Function** [[disabled/enabled](#)]: enables or disables the “Cleaning Function” tab. See section 2.1.2.1 for details.
- **Welcome Greeting** [[disabled/enabled](#)]: enables or disables the “Welcome Greeting” tab. See section 2.1.2.2 for details.

- **Welcome Back Object** [[disabled/enabled](#)]: enables or disables the “Welcome Back Object” tab. See section 2.1.2.3 for details.
- **Object to Change the Temperature Scale** [[disabled/enabled](#)]: enables the 1-bit object “[General] Temperature Scale”), which permits changing in runtime the scale of the temperatures that may show on the screen. By receiving one ‘0’ through this object, the scale will switch to Celsius, while after receiving one ‘1’ it will switch to Fahrenheit.

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

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

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

- **Scale After Programming** [[Celsius \(°C\)](#) / [Fahrenheit \(°F\)](#)]: sets the scale in use after download.

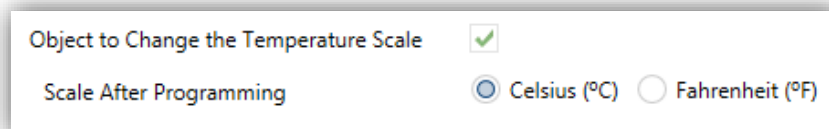


Figure 9. Temperature Scale After Programming.

2.1.2.1 CLEANING FUNCTION

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

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

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

ETS PARAMETERISATION

After enabling **Cleaning Function** from “Advanced” screen (see section 2.1.2), a new tab will be incorporated into the tree on the left.

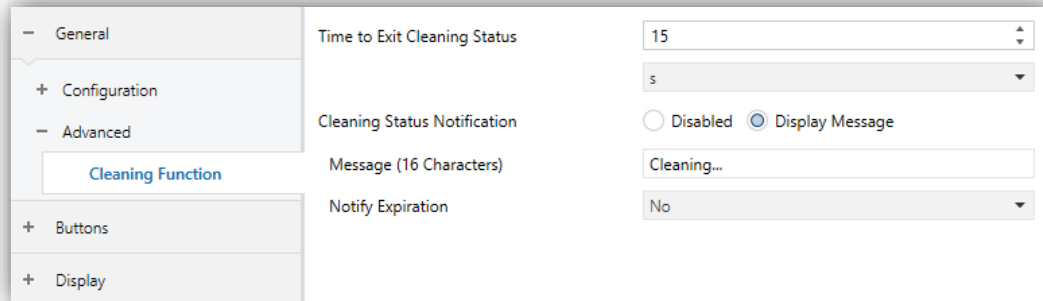


Figure 10. Advanced - Cleaning Function.

- **Time to Exit Cleaning Status** [5...15...65535][s] [1...65535][min/h]: timeout to deactivate the cleaning function once triggered.
- **Cleaning Status Notification** [Disabled / Display Message]: sets whether to show a message during the cleaning state. When “Display Message” is selected, the following parameters appear:
 - **Message** [Cleaning...]: textbox to enter the desired message.
 - **Notify Expiration** [No / Blink Message / Play Sound / Both]: sets whether to notify the timeout expiration or not. When any of the three later options is selected, a new parameter shows up:
 - **Length of the Warning** [1...5...65535][s] [1...65535][min/h]: sets the ahead-time to start the notification prior to the end of the cleaning function.

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

2.1.2.2 WELCOME GREETING

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

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

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

The message disappears when touching the screen or receiving a “0” through the welcome greeting object.

Note: *Welcome greeting takes precedence over screensaver. When the welcome greeting is displayed, the screensaver will be disabled until the first one disappears.*

ETS PARAMETERISATION

After enabling “**Welcome greeting**” (section 2.1.2), a new tab will be incorporated into the tab tree. A one-bit object named “[**General**] **Welcome Greeting**” will also be shown in the project topology, to trigger the welcome message by sending the value “1” or to end it by sending the value “0”.

Line	Mode	Text
Line 1	<input checked="" type="radio"/> Fixed	Welcome Mr/Mrs.
Line 2	<input type="radio"/> Fixed <input checked="" type="radio"/> Received from text Object	
Line 3	<input checked="" type="radio"/> Fixed	Touch the screen
Line 4	<input checked="" type="radio"/> Fixed	

Run Welcome Greeting with Text Object Update ☐

LEDs Flashing During Welcome Status ☐

Figure 11. Advanced– Welcome Greeting.

This screen contains the following parameters:

- **Line X** [Fixed / Received from text Object]: sets whether the corresponding text line will be pre-defined or object-dependent.

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

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

If “Received from text Object” is selected, the following parameter will appear:

- **Run Welcome Greeting with Text Object Update** *[enabled/disabled]*: allows choosing whether run the welcome greeting when writing on the 14-byte object “[General] Welcome Greeting – Line X”. Up to four objects will appear, depending on how many lines have been assigned the “Received from text Object” option.
- **LEDs Flashing During Welcome Status** *[enabled/disabled]*: sets whether the LED of the buttons should blink or not during the welcome greeting.

2.1.2.3 WELCOME BACK OBJECT

Flat Display can send a specific object (a **one-bit** value, a **scene** value or both, depending on the parameterisation) to the KNX bus when the user presses a touch button after a significant amount of time since the last press or presence detection (when the proximity sensor is enabled). Sending it or not can also depend on an **additional, configurable** condition consisting in the evaluation of up to five binary objects.

Any actions that in normal operation may be executed will not be if the welcome back object is sent to the bus. Thus, if the user presses a button and this causes that the welcome back object is sent, the normal action of that button will not be triggered.

ETS PARAMETERISATION

After enabling **Welcome Back Object** from “Advanced” screen (see section 2.1.2), a new tab will be incorporated into the tree on the left.

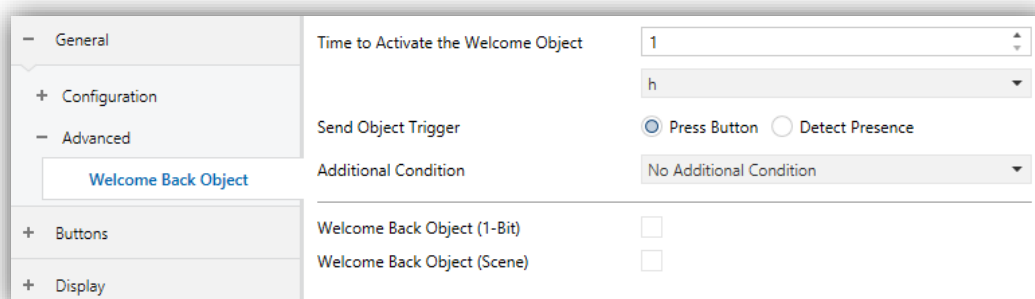


Figure 12. Advanced - Welcome Back Object.

- **Time to Activate the Welcome Object** [1...65535][s] [1...65535][min/h]: sets the minimum time that should elapse after the last button touch (or presence detection, when the proximity sensor is enabled) before the next one triggers the execution of the welcome back function.
- **Send Object Trigger** [Press Button / Detect Presence]: sets whether the welcome back object is sending after a touch in the screen or when the proximity sensor detects presence.
- **Additional Condition:** sets if sending the welcome back object should also depend on an external condition. The option by default is [No Additional Condition]. The following are available too:
 - [Do not send unless all additional conditions are 0]: the welcome back object will only be sent if all the condition objects are found to have the value "0".
 - [Do not send unless all additional conditions are 1]: the welcome back object will only be sent if all the condition objects are found to have the value "1".
 - [Do not send unless at least one of the additional conditions is 0]: the welcome back object will only be sent if at least one of the condition objects is found to have the value "0".
 - [Do not send unless at least one of the additional conditions is 1]: the welcome back object will only be sent if at least one of the condition objects is found to have the value "1".
 - **Number of Condition Objects** [1...5]: up to 5 objects can be selected for the additional condition ("[General] Welcome Back Object – Additional Condition").
- **Welcome Back Object (1-Bit)** [disabled/enabled]: checkbox to enable the sending of a 1-bit value (through "[General] Welcome back") when the welcome back function is triggered and the condition (if any) evaluates to true. The desired value should to be set in **Value** [Send 0/Send 1]
- **Welcome Back Object (Scene)** [disabled/enabled]: 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 value should to be set in **Scene Number** [1...64].

2.2 BUTTONS

2.2.1 CONFIGURATION

Flat Display features **five capacitive buttons** at the user's disposal for the execution of actions. Each will perform a specific function any time, as their functions do not depend on alternating menus, pages, etc.

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

All the buttons are identical; this brings a high level of versatility for a wide variety of applications. The following is a list of the functions that can be assigned to each button.

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

- **Individual:**

- LED indicator.
- Switch.
- Hold & release.
- Two objects (short press / long press).
- Scene.
- Scaling constant.
- Counter constant.
- Float constant.
- Dimmer.
- Shutter.
- Enumeration.
- [Climate] Fan.
- Room State.

- **Pair A, or Pair B**, being the function of such pair one of the following:

- Switch.
- Two objects (short press / long press).
- Dimmer.
- Shutter.
- Scaling.

- Counter.
- Enumeration.
- Float.
- Multimedia.
- [Climate] Temperature setpoint.
- [Climate] Mode.
- [Climate] Fan.
- [Climate] Mode Special.

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

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

ETS PARAMETERISATION

An independent tab for the parameterisation of the buttons is shown in ETS by default, initially containing only a sub-tab named “Configuration”.

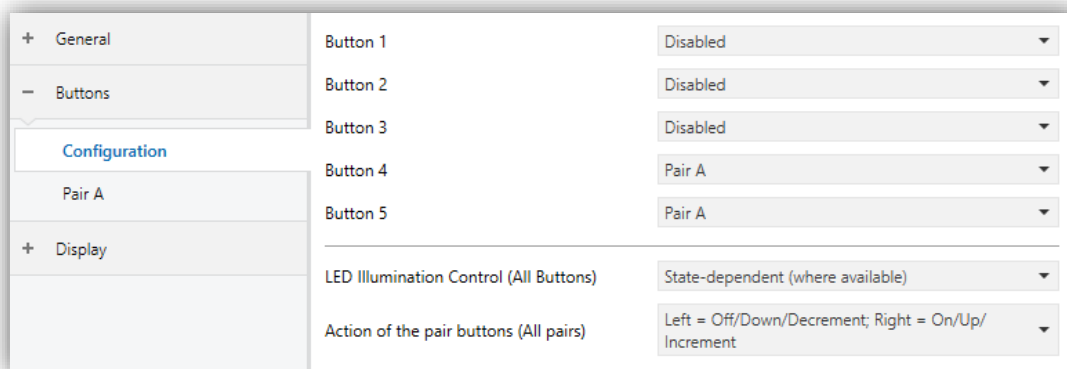


Figure 13. Buttons – Configuration

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

- [[Disabled](#)]. See section 2.2.2 for details.
- [[Individual](#)]. Selecting this option brings a new tab (“Button X”), which will make it possible to configure the functionality of that particular touch button. See section 2.2.3 for details.

- [Pair X]. Sets that this touch button will belong to a two-button control. Once a pair has been assigned to two buttons (and not before), a new tab ("Pair X") will show up in the tab tree, in order to configure the desired functionality. See section 2.2.4 for details.
- **LED Illumination Control (All buttons)**: jointly sets the LEDs illumination behaviour. The options are (please refer to ANNEX I. LED Illumination Modes for details):
 - [Regular]
 - [State-Dependent (where available)]
 - [State-Dependent (where available) (both LEDs)]
 - [Dedicated Object]
 - [Configure every button (pair) separately]: in case of selecting the last option, there will be a specific parameter **for each control** to specifically select the desired behaviour of the LED (or LEDs):

If at least one two-button control is being configured (either Pair A or Pair B), an additional parameter will be available. The options are:

- **Action of the pair buttons (All pairs)**: to determine an operation criterion:
 - [Left = Off/Down/Decrement; Right = On/Up Increment]
 - [Right = Off/Down/Decrement; Left = On/Up Increment]
 - [Every button pair is configured separately]: there will be a specific parameter in every two-button control for selecting the desired behaviour, unless it is configured as "Enumeration" or "[Climate] Mode".

2.2.2 DISABLE

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

ETS PARAMETERISATION

This function has no related parameters.

2.2.3 INDIVIDUAL

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

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

Under a “State-dependent” LED illumination, the LED will remain on/off according to the current state (on/off) of the object.

- **Hold & Release:** as soon as the user touches the button, a binary value will be sent to the KNX bus. And as long as the user releases the button, another value will be sent through the same object.

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

- **Two Objects (short press/long press):** specific binary values will be sent both after a short or a long press (a different object will be used in each case).

Under a “State-dependent” LED illumination, the LED will remain on/off according to the current state (on/off) of either one object or the other, which can be configured in parameters. However, if **LED Illumination Control (All Buttons)** 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

Table 1 Value range – Counter type constant

The “State-dependent” LED illumination mode is analogous as for the Scaling Constant function.

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

The “State-dependent” LED illumination mode is analogous as for the Scaling Constant and Counter Constant functions.

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

- Switch-on/Switch-off orders (on short presses).
- Step dimming orders (on long presses) and the subsequent stop order after the button is released.

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

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

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

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

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

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

- Standard: the device will react to both long and short presses, being possible to send the bus the following commands:
 - Move (raise/lower) orders (on **long presses**).
 - Stop/Step orders (on **short presses**).

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

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

As usual in the KNX standard, stop/step orders are interpreted by the actuators as a request to move the slats one step up or down (in case the

shutter is still) or as a request to interrupt the motion of the shutter (in case it is already moving up or down).

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

The direction of this motion (upwards or downwards) will **alternate** with every touch, according to the following sequence: downwards → upwards → downwards → ... However, there are some exceptions to this alternation:

- If the position of the shutter is found to be 0%, the next order will lower the shutter.
- If the position of the shutter is found to be 100%, the next order will raise the shutter.

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

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

- **Enumeration:** sends one-byte (unsigned) numeric value to the KNX bus from a list of up to 6 different values. Each press advances the list to the next item.

This control provides a parameter to link the control to a box on the screen so that the current numeric value is displayed at all times on the screen.

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

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

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

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

The table below shows the relation between the scaling values and the enumeration values, depending on the number of speed levels configured (note that if the Auto mode or the speed level 0 are allowed one more level is added, corresponding to value 0).

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

Table 2 Fan levels.

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

Under a "state-dependent" LED illumination, the LED will remain on/off according to the current state of the fan.

- **Room State:** allows controlling the room states (normal, make-up request, do not disturb). Pressing the button will activate the Do Not Disturb or Make-Up Request status (as configured) or deactivate it to return to Normal status.

Depending on the parameterisation and the current value of object, after a short press, the following values will be transmitted:

Parameterisation	Current Object Value	Transmitted Value
Make-Up Request	Do Not Disturb / Normal	Make-Up Request
	Make-Up Request	Normal
Do Not Disturb	Normal / Make-Up Request	Do Not Disturb
	Do Not Disturb	Normal

Table 3 Room States

If the LED illumination is "State-dependent", LED will light up when the current object value coincides with the parameterized value.

ETS PARAMETERISATION

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

The main parameter to be configured is:

- **Function** [*LED indicator* / *Switch* / *Hold & Release* / *Two Objects (short press / long press)* / *Scene* / *Scaling constant* / *Counter constant* / *Float constant* / *Dimmer* / *Shutter* / *Enumeration* / *[Climate] Fan* / *Room State*]: sets the desired function for the button.

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

- **LED Illumination Control:** sets the behaviour of the LEDs on the buttons. Depending on the Function selected for each button, the options may be: [*Regular* / *State-Dependent* / *Dedicated object*]. (See [ANNEX I. LED Illumination Modes](#) for details).

In case of selecting “Dedicated Object”, the object “[Btn] [In] LED On/Off” will be included in the project topology and a new parameter:

- **Value** [*0 = Off; 1 = On* / *0 = On; 1 = Off*] to select the value to switch off and on the LED shows up.

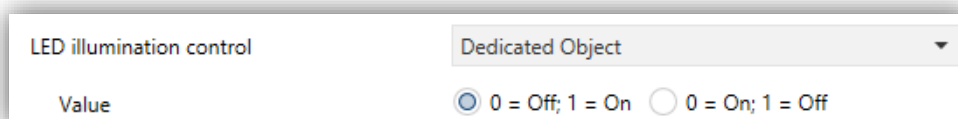


Figure 14 LED illumination control - Dedicated object.

Note: *this parameter will remain hidden unless having selected “Configure every button (pair) separately” in “LED Illumination Control (All Buttons)” (see 2.2.1).*

Moreover, the textbox **INFO** allows changing the default name of the tab in the left menu, as shows the following figure.

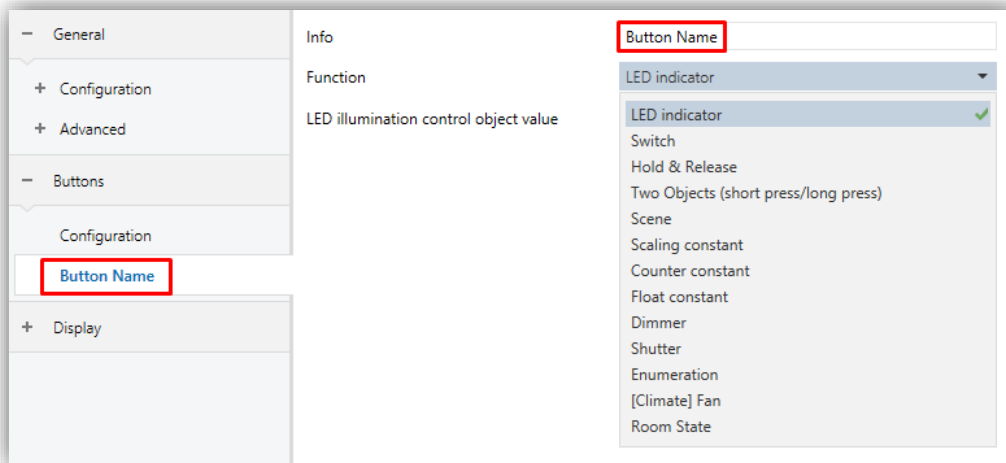


Figure 15. Individual Button.

LED indicator

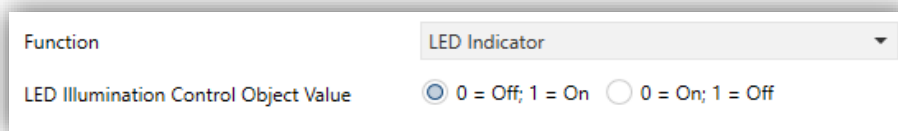


Figure 16. Individual Button – LED indicator.

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

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

Switch

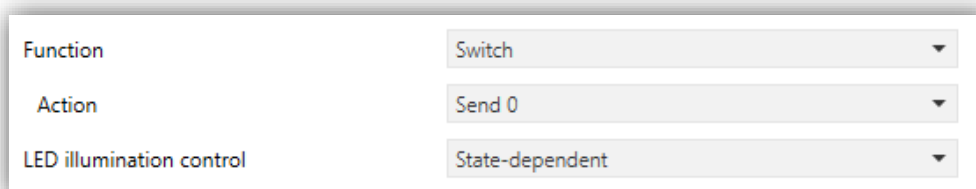
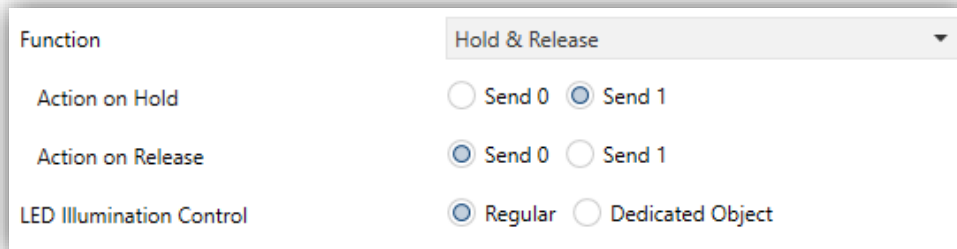


Figure 17. Individual Button - Switch.

- **Action** [[Send 0](#) / [Send 1](#) / [Toggle 0/1](#)]: sets the value to be sent to the bus (through object “[Btn] [In] Switch”) when the user touches the button.
- **LED illumination control** [[Regular](#) / [State-Dependent](#) / [Dedicated object](#)].

Hold & Release

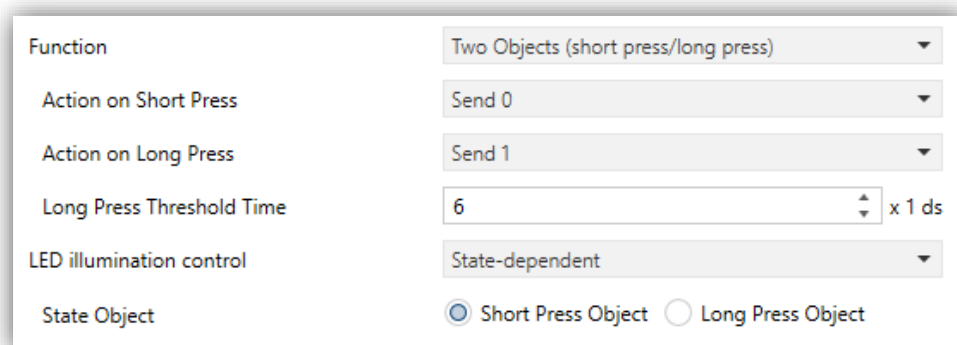


Function	Hold & Release
Action on Hold	<input type="radio"/> Send 0 <input checked="" type="radio"/> Send 1
Action on Release	<input checked="" type="radio"/> Send 0 <input type="radio"/> Send 1
LED Illumination Control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 18. Individual Button - Hold & Release.

- **Action on Hold** [[Send 0](#) / [Send 1](#)]: sets the value to be sent to the bus (through “[Btn] [In] Hold & Release”) when the user touches the button.
- **Action on Release** [[Send 0](#) / [Send 1](#)]: sets the value to be sent to the bus (again, through “[Btn] [In] Hold & Release”) when the user stops touching the button.
- **LED illumination control** [[Regular](#) / [Dedicated object](#)].

Two Objects (short press/long press)



Function	Two Objects (short press/long press)
Action on Short Press	Send 0
Action on Long Press	Send 1
Long Press Threshold Time	6 x 1 ds
LED illumination control	State-dependent
State Object	<input checked="" type="radio"/> Short Press Object <input type="radio"/> Long Press Object

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

- **Action on Short Press** [[Send 0](#) / [Send 1](#) / [Toggle 0/1](#) / [Send 1-byte Unsigned Int Value](#)]: sets the value to be sent to the bus (through “[Btn] [In] Two switches - Short Press”) when the user short-presses the button.

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** [*Send 0 / [Send 1](#) / Toggle 0/1 / Send 1-byte Unsigned Int Value*]: sets the value to be sent to the bus (through “[**Btn**] [**In**] **Two switches - Long Press**”) when the user long-presses the button.

In case of selecting the latter, an additional parameter (“**Value**”) will be displayed to enter the desired one-byte value [0...255].

- **Long Press Threshold Time** [0...[5](#)...50] [ds]: sets the minimum time the user should hold the button in order to consider it a long press.
- **LED illumination control** [[State dependent](#) / *Regular* / *Dedicated object*]:
Selecting “[State dependent](#)” brings an additional parameter:
 - **LED Illumination State Object** [[Short Press Object](#) / *Long Press Object*] to make the state of the LED correspond either to object “[**Btn**] [**In**] **Two switches - Short press**” or to object “[**Btn**] [**In**] **Two switches - Long press**” respectively.

Scene

Figure 20. Individual Button - Scene.

- **Action** [[Run](#) / *Run and Save*]: sets whether the value to be sent to the KNX bus when the user touches the button will always be a scene run request or – depending on the length of button press– a scene run or save request (3s press).
- **Scene number** [[1](#)...64]: number of the scene to be sent to the bus, both in the case of the run requests and the save requests.

- **Object to Use** [[General Scene Object](#) / [Individual Control Scene Object](#)]: commands are sent via the general scene object (“[General] Scene: Send”) or through a control object (“[Btn][I1] Scene: Send”).
- **LED illumination control** [[Regular](#)/[Dedicated object](#)].

Scaling Constant / Counter Constant / Float Constant

Figure 21. Individual Button - Scaling Constant

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

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

Function	Size	Sign	Object Value	Object Name
Scaling	1-Byte		[0...100]	[Btn][In] 1-Byte Scaling Value
Counter	1-Byte	<u>Signed</u>	[-128...0...127]	[Btn][In] 1-Byte Signed Int Value
		<u>Unsigned</u>	[0...255]	[Btn][In] 1-Byte Unsigned Int Value
	2-Bytes	<u>Signed</u>	[-32768...32767]	[Btn][In] 2-Byte Signed Int Value
		<u>Unsigned</u>	[0...65535]	[Btn][In] 2-Byte Unsigned Int Value
Float	2-Bytes		[-671088.64...0...670760.96]	[Btn][In] 2-Byte Float Value

Tabla 1 Constant type numerical control.

- **LED illumination control** [[Regular](#) / [State dependent](#) / [Dedicated object](#)].

Dimmer

Function	Dimmer
Step	100%
LED illumination control	State-dependent

Figure 22. Individual Button - Dimmer

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

On the other hand, the “[Btn] [In] Light - Dimming (Status)” one-byte object may be linked to the light level status object of the dimmer.

The parameters for this function are:

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

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

- **LED illumination control** [[State-Dependent](#) / Regular / Dedicated object].

Shutter

Function	Shutter
Type	<input checked="" type="radio"/> Standard <input type="radio"/> Hold & Release
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated object

Figure 23. Individual Button - Shutter

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

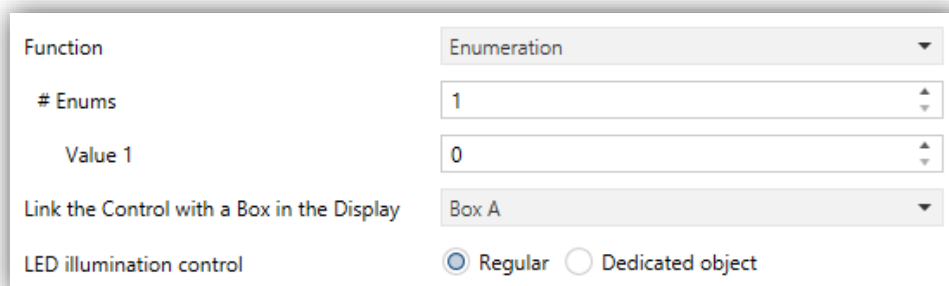
Additionally, a one-byte object (“[Btn] [In] Shutter Position”) is provided to link it to the position status object of the shutter actuator (in fact, this object is only intended to receive values from the bus, not to send them).

The parameters for this function are:

- **Type:** sets the desired control type:
 - [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.
 - [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). After the button is released, it will send an order to stop the shutter.
- **LED illumination control** [Regular / Dedicated object].

Enumeration

This control is analogous to the Switch control, with the particularity that the communication object will be one byte (“[Btn][In] Enumeration”) and that up to 6 states can be distinguished depending on the object value.



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

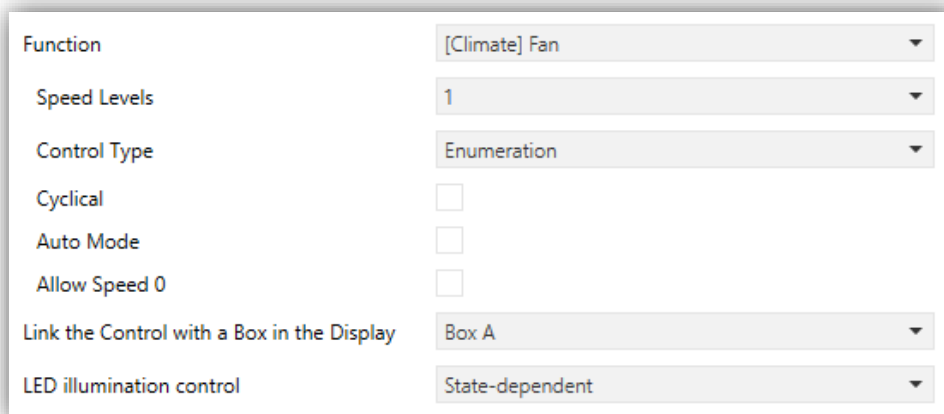
Figure 24. Individual Buttons – Enumeration.

- **# Enums** [1...6]: number of states that will be distinguished. For every distinguished state, the following parameter will become available:

- **Value j [0...255]**: numerical value that will be sent through the control object when the user, after pressing the buttons, sets the control to state j . The actual sending will not be performed until 1.5 seconds after the last press.
- **Link the control with a box in the display**: sets the area of the display where to show the indicator of this control, i.e., the current numeric value. See section 2.3 for details.
- **LED illumination control** [[Regular](#) / [Dedicated object](#)].

[Climate] Fan

When this function is assigned to the button, one object (“[Btn][In] Fan Control”) for each type of control and 1-byte status object (“[Btn][In] Fan Indicator”) are enabled. The status object (which must be linked to the status object of the fan actuator) will indicate, in percent, the value of the current speed level.



Function	[Climate] Fan
Speed Levels	1
Control Type	Enumeration
Cyclical	<input type="checkbox"/>
Auto Mode	<input type="checkbox"/>
Allow Speed 0	<input type="checkbox"/>
Link the Control with a Box in the Display	Box A
LED illumination control	State-dependent

Figure 25 Individual Buttons – [Climate] Fan

- **Speed Levels [1...5]**: allows selecting the speed levels available for control.
- **Control Type**: sets the type of the communications objects to control the fan level.
 - [[1 bit \(decrease/increase\)](#)]: orders of speed increase/decrease are sent through the one-bit object “[Btn] [PX] Fan Control (Type: 1-Bit)”.
 - [[Scaling](#)]: scaling values are sent through the one-byte object “[Btn] [PX] Fan Control (Type: Scaling)”.

- **[Enumeration]**: integer values are sent through the one-byte object “[Btn] [PX] Fan Control (Type: Enumeration)”.
- **Cyclical** [\[enabled/disabled\]](#): sets whether scrolling through the speed levels is circular or not. If checked, a press switches to the minimum level. On the other hand, if cyclical is disabled, the level increases to the last one and then decrease. At levels other than the first and last, the last action (increase or decrease) is maintained until reaching the first or last.
- **Auto Mode** [\[enabled/disabled\]](#): sets whether the fan Auto mode will be available or not. If checked, the following parameters are also shown.
- **Dedicated Object for Auto Mode** [\[enabled/disabled\]](#): sets how the Auto mode should be activated. In the absence of a dedicated object, the Auto fan mode will be activated by switching to fan level 0.

Auto Mode	<input checked="" type="checkbox"/>
Dedicated Object for Auto Mode	<input type="checkbox"/>
Allow Speed 0 (auto)	<input checked="" type="checkbox"/>

Figure 26. Pair Buttons – [Climate] Fan – Auto Mode.

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

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

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

Dedicated Object for Auto Mode	<input checked="" type="checkbox"/>
Value to set auto mode	<input type="radio"/> Send 0 to set auto mode <input checked="" type="radio"/> Send 1 to set auto mode
Long press to activate auto mode	<input type="checkbox"/>
Allow Speed 0	<input type="checkbox"/>

Figure 27. Pair Buttons – [Climate] Fan – Dedicated Object for Auto Mode.

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

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

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

- By long press on any of the buttons of the control (requires checking “**Long Press to Activate Auto Mode**” [[enabled/disabled](#)]).

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

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

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

- **Allow speed 0** [[enabled/disabled](#)]: sets whether the speed level 0 will be present or not. When the “**Auto Mode without a dedicated object**” has been configured, this option will be necessarily activated.
- **Link the control with a box in the display**: sets the area of the display where to show the indicator of this control, i.e., the current fan speed. The selected box must be configured with the same data type (see section 2.3 for details).

Note: *this parameter is only available for Scaling or Enumeration control types.*
- **LED illumination control** [[Regular](#) / [State-dependent](#) / [Dedicated Object](#)].

Room State

When this function is assigned to the button, the object for the control ““**[Btn][In] Room State**” is enabled. This object will also be a status indicator.

Function	Room State
State	<input checked="" type="radio"/> Make-Up Request <input type="radio"/> Do Not Disturb
LED illumination control	State-dependent

Figure 28. Individual Button – Room State.

- **State** [[Make-up Request](#) / [Do not Disturb](#)]: sets the state that is activated with this button. Commutes between Normal (“0”) and the selected state: Make-up room (“1”) and Do not disturb (“2”).
- **LED illumination control** [[Regular](#) / [State-Dependant](#) / [Dedicated Object](#)].

2.2.4 PAIR

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

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

Under a “[state-dependent](#)” LED illumination (see [ANNEX I. LED Illumination Modes](#)), the LED of the corresponding button will remain on/off according to the current state (on/off) of the switch.

On the other hand, under a “[state-dependent \(both LEDs\)](#)” LED illumination, both of them will remain on while the switch is in the “on” state, and off while in the “off” state.

- **Two objects (short press/long press):** control for sending specific binary values both after a short or a long press [on any of the two](#). Different objects will be used for the short and long presses.

Moreover, it is possible (in parameters) to make the “[state-dependent](#)” and “[state-dependent \(both LEDs\)](#)” LED illumination modes (see [ANNEX I. LED Illumination Modes](#)) depend on either one object or the other. However, if “**LED Illumination Control (All Buttons)**” (section 2.2.1) has been set to “[state-dependent \(where available\)](#)” only the short press object will be considered.

- **Dimmer:** sends 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). Setting which button does what is possible.

Under a “state-dependent” LED illumination (see ANNEX I. LED Illumination Modes), the LED of the corresponding button will remain on/off according to whether the current value of the light level status object (which should be updated by the actual dimmer) is greater than 0% or not.

On the other hand, under a “state-dependent (both LEDs)” LED illumination, both together will remain on or off depending on such value.

- **Shutter:** This option permits making use of the two buttons to control a shutter actuator connected to the bus. Two alternative control methods are possible:
 - Standard: a long press will make the device send to the KNX bus an order to start moving the shutter (upwards or downwards, depending on the button), while a short press will make it send a stop order (which will be interpreted as an order to step up or to step down –depending on the button– if the shutter was not in motion and such function is available).
 - Hold & Release: as soon as the button is held, the device will send the KNX bus an order to start moving the shutter (upwards or downwards, depending on the button). Once the button is released, it will send an order to stop the shutter.

The “state-dependent” and “state-dependent (both LEDs)” LED illumination modes are not available for this function. See ANNEX I. LED Illumination Modes for details.

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

These controls provide a parameter to link them to a box in the display so the current numerical value is shown on the display. The screen will reflect the

new values as the user touches the buttons, however values are actually sent to the bus 1.5 seconds after the last press.

The “state-dependent” and “state-dependent (both LEDs)” LED illumination modes are not available for this function. See ANNEX I. LED Illumination Modes for details.

- **Enumeration:** pressing the buttons will make Flat Display send a one-byte (unsigned) value to the bus. There were up to 6 different values. Successively pressing on the buttons will make Flat Display walk through the different values.

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

The “state-dependent” and “state-dependent (both LEDs)” LED illumination modes are not available for this function. See ANNEX I. LED Illumination Modes for details.

- **Multimedia:** pressing one of the two buttons will make Flat Display send a binary value to the bus, while pressing on the other will make it send the inverse binary value. It is possible to set by parameter whether the left button sends "Stop/Reverse" and the right button "Start/Forward", or vice versa.

Under a “state-dependent” LED illumination (see ANNEX I. LED Illumination Modes), the LED of the corresponding button will remain on/off according to whether the current state of the binary object.

On the other hand, under a “state-dependent (both LEDs)” LED illumination, both together will remain on depending on while the multimedia control is in “Start/Forward” status, or off while in “Stop/Backward” status.

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

This function can also be linked to a box in the display in order to show the current setpoint value. The box will reflect the value (in °C or °F, depending on the selected scale; see section 2.1.2) of the status object, which gets automatically updated after sending control orders (that is, after pressing the

buttons), being even possible to receive values from the bus, for example, from the corresponding setpoint status object from the external thermostat.

The “state-dependent” and “state-dependent (both LEDs)” LED illumination modes are not available for this function. See ANNEX I. LED Illumination Modes for details.

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

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

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

- **[Climate] Fan:** implements a two-button fan control which sends the KNX bus values in order to change the fan speed level. This function is analogous to the ventilation control on individual button described in section 2.2.3.
- **[Climate] Special Mode:** lets the user sequentially commute between the different special climate mode: Auto, Comfort, Standby, Economy and Building Protection. This function is analogous to the modes control in its Extended version.

ETS PARAMETERISATION

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

The main parameter that needs to be configured is:

- **Function** [Switch / Two Objects (short press / long press) / Dimmer / Shutter / Scaling / Counter / Enumeration / Float / Multimedia / [Climate] Temperature Setpoint / [Climate] Mode / [Climate] Fan / [Climate] Mode Special]: sets the desired function for the button.

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).

- **LED Illumination Control:** sets the behaviour of the LEDs on the buttons. Depending on the type of control selected for each buttons pair, the options may be: [Regular / State-Dependent / State-dependent (both LEDs) / Dedicated object]. (See ANNEX I. LED Illumination Modes for details)

In case of selecting “Dedicated Object”, the object “[Btn] [PX] LED On/Off” will be included in the project topology and a new parameter:

- **Value** [0 = Off; 1 = On / 0 = On; 1 = Off] to select the value to switch off and on the LED shows up.

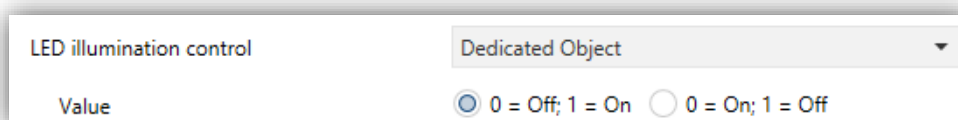


Figure 29 LED illumination control - Dedicated object.

Note: *this parameter will remain hidden unless having selected “Configure every button (pair) separately” in “LED Illumination Control (All Buttons)” (see 2.2.1).*

Moreover, the textbox **INFO** allows changing the default name of the tab in the left menu, as shows the following figure.

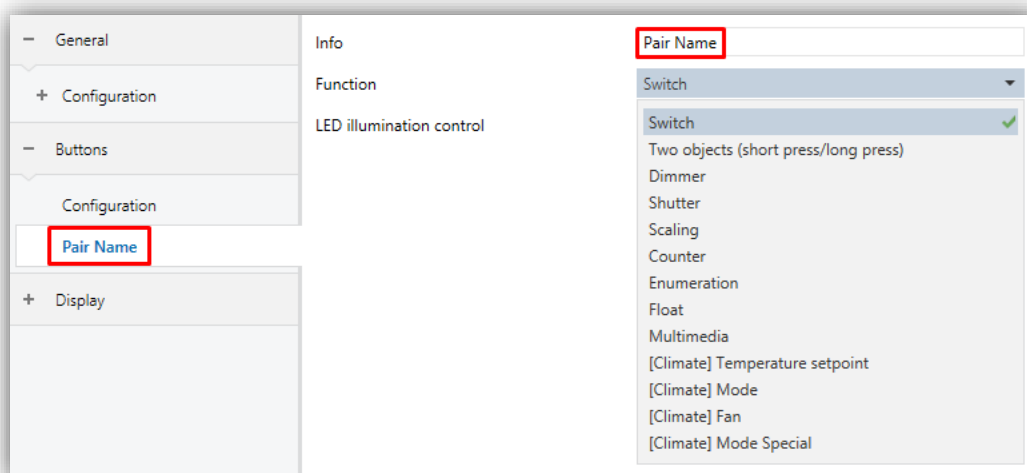


Figure 30. Pair A.

Switch

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

Figure 31. Pair Buttons – Switch

- **Action** [Left = 0; Right = 1 / Left = 1; Right = 0]: assigns each of the two buttons the value to be sent through “[Btn] [PX] Switch”.

Note: *this parameter will remain hidden unless having selected “Every button pair is configured separately” in **Action of the pair buttons** (see section 2.2.1).*

- **LED illumination control** [Regular / State-dependent / State-dependent (both LEDs) / Dedicated Object].

Two objects (short press / long press)

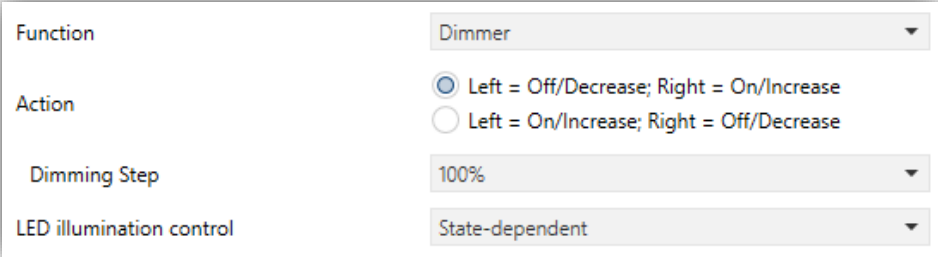
Function	Two objects (short press/long press)
Action on Short Press	<input checked="" type="radio"/> Left = 0; Right = 1 <input type="radio"/> Right = 0; Left = 1
Action on Long Press	<input checked="" type="radio"/> Left = 0; Right = 1 <input type="radio"/> Right = 0; Left = 1
Long Press Threshold Time	6 x 1 ds
LED illumination control	State-dependent
State Object	<input checked="" type="radio"/> Short Press Object <input type="radio"/> Long Press Object

Figure 32 Pair Buttons - Two objects (short press/long press).

- **Action on Short Press** [Left = 0; Right = 1 / Right = 0; Left = 1]: sets the value to be sent through the object “[Btn][PX] Two Objects – Short Press” when short pressing each of the two buttons.
- **Action on Long Press** [Left = 0; Right = 1 / Right = 0; Left = 1]: sets the value to be sent through the object “[Btn][PX] Two Objects – Long Press” when long pressing each of the two buttons.

- **Long Press Threshold Time** [4...6...50] [ds]: sets the minimum time the user should hold the button in order to consider it a long press.
- **LED illumination control** [*Regular* / *State-dependent* / *State-dependent (both LEDs)* / *Dedicated Object*]. Selecting “*State-dependent*” or “*State-dependent (both LEDs)*” brings an additional parameter:
 - **State Object** [*Short Press Object* / *Long Press Object*], to make the state of the LEDs correspond either to object “[Btn] [PX] Two switches: Short press” or to object “[Btn] [PX] Two switches: Long press” respectively.

Dimmer



Function	Dimmer
Action	<input checked="" type="radio"/> Left = Off/Decrease; Right = On/Increase <input type="radio"/> Left = On/Increase; Right = Off/Decrease
Dimming Step	100%
LED illumination control	State-dependent

Figure 33 Pair Buttons - Dimmer

The parameters available are:

- **Action** [*Left = Off/Decrease; Right = On/Increase* / *Left = On/Increase; Right = Off/Decrease*]: sets which of the two buttons will send the “turn on” orders and which the “turn off” orders.

Note: *this parameter will remain hidden unless having selected “Every button pair is configured separately” in **Action of the pair buttons** (see section 2.2.1).*

- **Dimming Step** [*100%* / *50%* / *25%* / *12.5%* / *6.25%* / *3.1%* / *1.5%*]: sets the increase or decrease in the light level (in terms of percentage) the dimmer must perform with every step order. How step dimming works is detailed below.

Note: *since dimmers typically do not apply the new light level immediately (i.e., the step regulation is performed progressively) and since Flat Display sends an order to interrupt the step dimming once the user releases the button, it is advisable to configure a step of 100%. This way, the user can*

perform any dimming step by simply leaving the button pressed and then releasing it without needing to make successive button presses.

- **LED illumination control** [*Regular* / *State-dependent* / *State-dependent (both LEDs)* / *Dedicated Object*].

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

On the other hand, the “[Btn] [PX] Light Dimming (Status)” one-byte object may be linked to the light level status object of the dimmer (in fact, this object is only intended to receive values from the bus, not to send them).

Shutter

Function	Shutter
Action	<input checked="" type="radio"/> Left = Downwards; Right = Upwards <input type="radio"/> Left = Upwards; Right = Downwards
Type	<input checked="" type="radio"/> Standard <input type="radio"/> Hold & Release
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 34. Pair Buttons - Shutter.

- **Action** [*Left = Downwards; Right = Upwards* / *Right = Upwards; Left = Downwards*]: sets which of the two buttons will send the move up orders and which the move down orders.

Note: this parameter will remain hidden unless having selected “*Every button pair is configured separately*” in **Action of the pair buttons** (see section 2.2.1).

- **Type** [*Standard*/Hold & Release]: sets the desired behaviour of the buttons.
- **LED illumination control** [*Regular* / *Dedicated Object*].

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

Scaling, Counter, Float

Function	Counter
Action	<input checked="" type="radio"/> Left = Decrease; Right = Increase <input type="radio"/> Left = Increase; Right = Decrease
Size	<input checked="" type="radio"/> 1 Byte <input type="radio"/> 2 Bytes
Sign	<input checked="" type="radio"/> Signed <input type="radio"/> Unsigned
Minimum Value	-128
Maximum Value	127
Increment on Short Press	1
Increment on Long Press	10
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 35. Pair Buttons - Counter.

- **Action** [Left = Decrease; Right = Increase / Left = Increase; Right = Decrease]: sets which of the two buttons will increase the current numerical value and which will decrease it on user presses.

Note: *this parameter will remain hidden unless having selected “Every button pair is configured separately” in **Action of the pair buttons** (see section 2.2.1).*

- **Minimum Value:** sets which value from the available range will be the minimum value permitted by the control after a number of presses on the decrease button.
- **Maximum Value:** sets which value from the available range will be the maximum value permitted by the control after a number of presses on the increase button.
- **Increment on Short Press:** sets the increase or decrease to be applied to the current value on every short press over the increase or decrease buttons, respectively.
- **Increment on Long Press:** sets the increase or decrease to be applied to the current value on every long press over the increase or decrease buttons, respectively.

- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. See section 2.3 for details
- **LED illumination control** [[Regular](#) / [Dedicated Object](#)].

Whether the selected function is **counter** type, two additional options will be enabled:

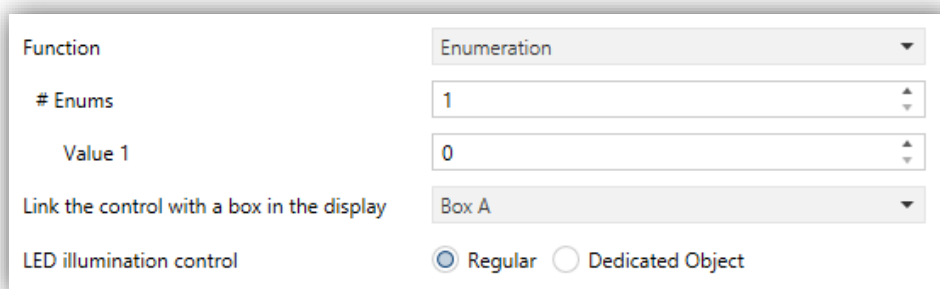
- **Size** [[1 Byte](#) / [2 Bytes](#)]: size of the indicator object.
- **Sign** [[Signed](#) / [Unsigned](#)]: sets whether the range includes negative values or only positive values

The different types of configurable 2-buttons controls are listed in the following table:

Function	Size	Sign	Minimum Value	Maximum Value	Increment short press	Increment long press	Related Object
Counter	1-Byte	Signed	[-128...127]	[-128...127]	[-128...1...127]	[-128...10...127]	[Btn][PX] Counter - 1-Byte signed
		Unsigned	[0...255]	[0...255]	[0...255]	[0...255]	[Btn][PX] Counter - 1-Byte unsigned
	2-Byte	Signed	[-32768...32767]	[-32768...32767]	[-32768...32767]	[-32768...32767]	[Btn][PXi] Counter - 2-Byte signed
		Unsigned	[0...65535]	[0...65535]	[0...65535]	[0...65535]	[Btn][PX] Counter - 2-Byte unsigned
Scaling	1-Byte		[0...100]	[0...100]	[0...1...100]	[0...10...100]	[Btn][PX] Scaling
Float	2-Byte		[-671088.64...670760.96]	[-671088.64...670760.96]	[-671088.64...0.5...670760.96]	[-671088.64...1...670760.96]	[Btn][PX] Float

Table 4 2-buttons Numeric Control

Enumeration



Function: Enumeration

Enums: 1

Value 1: 0

Link the control with a box in the display: Box A

LED illumination control: ☒ Regular ☐ Dedicated Object

Figure 36. Pair Buttons - Enumeration.

- **# Enums** [[1...6](#)]: number of states that will be distinguished. For every distinguished state, the following parameter will become available:
 - **Value j** [[0...255](#)]: numerical value that will be sent through the control object “[Btn][PX] Enumeration” when the user, after pressing the buttons,

sets the control to state j . The actual sending will not be performed until 1.5 seconds after the last press.

- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. See section 2.3.
- **LED illumination control** [[Regular](#) / [Dedicated Object](#)].

Multimedia

Function	Multimedia
Action	<input checked="" type="radio"/> Left = Stop/Backward; Right = Start/Forward <input type="radio"/> Left = Start/Forward; Right = Stop/Backward
LED illumination control	State-dependent

Figure 37. Pair Buttons - Multimedia

- **Action** [[Left = Stop/Backward; Right = Start/Forward](#) / [Left = Start/Forward; Right = Stop/Backward](#)]: sets which of the two buttons will send the Stop/Backward orders and which one the Start/Forward orders. The orders will be sent to the bus through the 1-bit object "[Btn] [PX] Multimedia.
- **LED illumination control** [[Regular](#) / [State-dependent](#) / [State-dependent \(both LEDs\)](#) / [Dedicated Object](#)].

[Climate] Temperature Setpoint

Function	[Climate] Temperature setpoint
Action	<input checked="" type="radio"/> Left = Decrease; Right = Increase <input type="radio"/> Left = Increase; Right = Decrease
Minimum Value	10 °C
Maximum Value	30 °C
Increment on Short Press	0,5 °C
Increment on Long Press	1 °C
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 38. Pair Buttons - [Climate] Temperature setpoint.

- **Action** [[Left = Decrease](#); [Right = Increase](#) / [Right = Increase](#); [Left = Decrease](#)]: sets which of the two buttons will permit increasing the setpoint value and which one will permit decreasing it. The values are sent to the bus through “[Btn] [PX] (Climate) Temperature Setpoint”.

Note: *this parameter will remain hidden unless having selected “[Every button pair is configured separately](#)” in **Action of the pair buttons** (see section 2.2.1).*

- **Minimum Value** [-99...[10](#)...199]: minimum value than can be reached by the control after a number of presses on the decrease button
- **Maximum Value** [-99...[30](#)...199]: maximum value than can be reached by the control after a number of presses on the increase button.
- **Increment on Short Press** [[0.1](#)...[0.5](#)...10]: sets the increase or decrease step to be applied to the current value on every short press over the increase or decrease buttons, respectively.
- **Increment on Long Press** [[0.1](#)...[1](#)...10]: sets the increase or decrease step to be applied to the current value on every long press over the increase or decrease buttons, respectively.

Note: *Long and short increments are applied in °C regardless of the scale used.*

- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. The selected box must be configured with the same data type (see section 2.3 for details).
- **LED illumination control** [[Regular](#) / [Dedicated Object](#)].

[Climate] Mode

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

Figure 39. Pair Buttons – [Climate] Mode (Heat/Cool).

• **Mode Type:** sets mode control type:

- [Heat/Cool]: Selecting this type of mode control turns the box into a 2-button control one of which will activate the Heat mode, while the other one will activate the Cool mode.

One communication object is also enabled: the “[Btn][PX] (Climate) **Mode**” control and status object. Depending on the mode selected by the user, the control object will be sent to the bus a certain value (‘0’ for Cool and ‘1’ for Heat).

- [Extended]: pressing on the two buttons will make Flat Display walk through the different HVAC modes and send a one-byte value to the bus according to the selected mode.

Up to five modes **Heat / Cool / Auto / Fan / Dry** are available, each of which can be in ETS by means of the proper checkbox, which permits setting which of all the five extended modes will be included into the sequential scrolling implemented by the buttons.

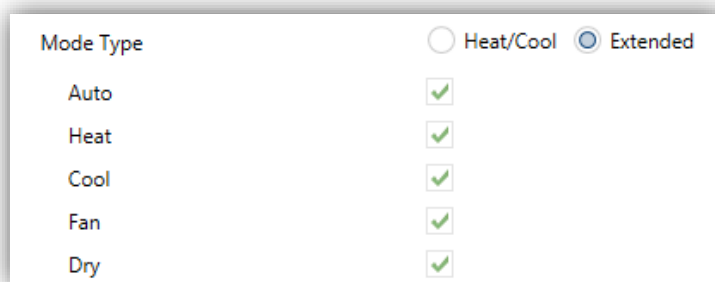


Figure 40. Pair Buttons – [Climate] Mode (Extended).

When this type of mode control is assigned to the box, one 1-byte communication object is enabled: the “[Btn][PX] (Climate) **Mode – Extended**” control and status object. Depending on the mode selected by the user, the control object will be sent to the bus a certain value.

Mode	Sent Value
Auto	0 (0x00)
Heat	1 (0x01)
Cool	3 (0x03)
Fan	9 (0x09)
Dry	14 (0x0E)

Table 2. HVAC Mode vs. Object Value.

- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current numeric value. The selected box must be configured with the same data type (see section 2.3 for details).

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

- **LED illumination control** [*Regular* / *State-dependent* / *State-dependent (both LEDs)* / *Dedicated Object*].

Note: *for Extended mode type is only available the options [*Regular* / *Dedicated Object*].*

[Climate] Fan

This control is analogous to the Fan control in individual buttons (see section 2.2.3). The objects that depending on the configuration will send the orders to the bus are in this case the following:

- “[Btn] [PX] Fan Control (Type: 1-Bit)”
- “[Btn] [PX] Fan Control (Type: Scaling)”
- “[Bnt] [PX] Fan Control (Type: Enumeration)”

The status object (“[Btn] [PX] Fan Indicator”) will be percentage-type.

Function	[Climate] Fan
Action	<input checked="" type="radio"/> Left = Decrease; Right = Increase <input type="radio"/> Left = Increase; Right = Decrease
Speed Levels	1
Control Type	1 bit (Decrease/Increase)
Cyclical	<input type="checkbox"/>
Auto Mode	<input type="checkbox"/>
Allow Speed 0	<input type="checkbox"/>
LED illumination control	State-dependent

Figure 41. Pair Buttons - [Climate] Fan.

When two buttons are available, the "direction" of increasing or decreasing the ventilation level or switch the automatic mode is determined by the current ventilation level reflected in the status object, by the "Cyclic" parameter, by whether or not there is a dedicated object and the action associated to each buttons of the pair:

- **Action** [[Left = Decrease; Right = Increase](#) / [Left = Increase; Right = Decrease](#)]: sets which of the buttons to use to send an order to decrease and which to increase.

Note: *this parameter will remain hidden unless having selected “[Every button pair is configured separately](#)” in **Action of the pair buttons** (see section 2.2.1).*

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

- **Link the control with a box in the display:** sets the area of the display where to show the indicator of this control, i.e., the current fan speed. The selected box must be configured with the same data type (see section 2.3 for details).

Note: *this parameter is only available for [Scaling](#) or [Enumeration](#) control types.*

- **LED illumination control** [[Regular](#) / [State-dependent](#) / [State-dependent \(both LEDs\)](#) / [Dedicated Object](#)].

[Climate] Mode Special

This control is exactly the same as **[Climate] Mode** but with the modes: **Auto**, **Comfort**, **Standby**, **Economy**, **Building Protection**. Which of the modes should be available can be set through five specific checkboxes.

The climate mode will be sent through the object “**[Btn][Px] (Climate) Mode Special**”.

Function	[Climate] Mode Special
Auto	<input checked="" type="checkbox"/>
Comfort	<input checked="" type="checkbox"/>
Standby	<input checked="" type="checkbox"/>
Economy	<input checked="" type="checkbox"/>
Building Protection	<input checked="" type="checkbox"/>
Link the control with a box in the display	Box A
LED illumination control	<input checked="" type="radio"/> Regular <input type="radio"/> Dedicated Object

Figure 42. Pair Buttons - [Climate] Mode Special

2.3 DISPLAY

2.3.1 CONFIGURATION

Flat Display features a **2.4” 128 x 64 pixel back-lit monochrome screen** with black background and white icons and fonts.

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

The information provided by the display can consist in icons or alphanumeric characters (texts). Please refer to the specific document “**Icons in Flat Display**” (available at the Zennio homepage, www.zennio.com) to see the list of available icons.

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

- **Basic Latin**², in small and big font size.
- **Extended Latin** (“À” to “ÿ”), in small and big font size.
- **Cyrillic** alphabet, in small and big font size.
- **Symbols**: € ° & - . ;

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

The screen is divided into different zones according to the chosen page style:

- **Normal Page**: 4 zones (**single** boxes –or *big* boxes–), named “A”, “B”, “C” and “D”, each of which can be further divided into two **double** boxes –or *small* boxes–.

² According to the Unicode classification.



Figure 43. Display – Thermostat Page

- **Thermostat:** the display is divided into 3 zones, named "A", "B" and "C", distributed as follows:
 - **Zone A:** occupies the left side of the screen and can be configured as a **double** box.
 - **Zone B:** **setpoint** box is displayed centred on the screen and larger than the other boxes.
 - **Zone C:** occupies the right side of the screen and can be configured as a **fan** box or as a **double** box.

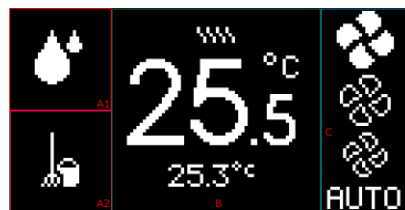


Figure 44. Display – Normal Page

Each box consists of two areas:

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

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

ETS PARAMETERISATION

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

This screen contains the following parameters:

- **Page Style** [[Normal Page](#) / [Thermostat](#)]: sets the layout and functionality of the display.

The screenshot shows the 'Configuration' sub-tab selected in the left sidebar. The main panel is titled 'Page Style' and has two radio buttons: 'Normal Page' (selected) and 'Thermostat'. Below this is a 'Layout' section with a 2x2 grid of boxes labeled A, B, C, and D. Underneath the layout are four dropdown menus for 'Zone A', 'Zone B', 'Zone C', and 'Zone D', all of which are currently set to 'Disabled'. At the bottom, there is a 'Border lines' dropdown set to 'Invisible line in every border' and an 'Indicator representation' section with two radio buttons: 'All indicators are type "permanent"' (selected) and 'Every indicator is configured separately'.

Figure 45. Display – Configuration (Normal Page)

The screenshot shows the 'Configuration' sub-tab selected in the left sidebar. The main panel is titled 'Page Style' and has two radio buttons: 'Normal Page' and 'Thermostat' (selected). Below this is a 'Layout' section with a 1x3 grid of boxes labeled A, B, and C. Underneath the layout are three dropdown menus for 'Zone A', 'Zone B', and 'Zone C'. 'Zone A' has two radio buttons: 'Disabled' (selected) and 'Double (two small boxes)'. 'Zone B' has a 'Setpoint' dropdown set to 'Disabled'. 'Zone C' has a dropdown set to 'Disabled'. At the bottom, there is a 'Border lines' dropdown set to 'Invisible line in every border' and an 'Indicator representation' section with two radio buttons: 'All indicators are type "permanent"' (selected) and 'Every indicator is configured separately'.

Figure 46. Display – Configuration (Thermostat Page)

One drop-down list is shown per zone. Depend on the page style and the zone, the options may be:

- [[Disabled](#)]. See section 2.3.2.

- [Single (one big box)]. Selecting this option brings a new tab to the tree on the left (named “**Box X**”, where X depends on the display zone). See section 2.3.3.

Note: *This option will be available in all zones when the selected page style is Normal Page.*

- [Double (two small boxes)]. Selecting this option brings the checkboxes “**Box X1**” and “**Box X2**” (where X depends on the display zone), which turn each of the two zones enabled or disabled. Depending on this, up to two new tabs (named “**Box Xn**”) will be included in the tree on the left. See section 2.3.4.
- [Fan (one big box)]. Selecting this option brings a new tab to the tree on the left named “**Box C**”. See section 2.3.6.

Note: *Option only available for **zone C** of Thermostat Page.*

- [Setpoint]: Selecting Thermostat Page, this option will always be activated in **zone B**. The “**Box B**” tab is displayed in the left tree.

- **Border lines** [Solid line in every border / Invisible line in every border / Dotted line in every border / Customize every border]: lets selecting how the box border lines will be represented.

In case of selecting “Customize every border”, an independent tab appears under Display to configure each box border line (see section 2.3.7).

- **Indicator representation** [All indicators are type “permanent” / Every indicator is configured separately]: defines the representation of the indicators on the display.

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

2.3.2 DISABLED

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

ETS PARAMETERISATION

This function has no related parameters.

2.3.3 SINGLE

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

Single boxes can be assigned the following types of indicators:

- **Icon:** the indicator will represent an icon according to the value of an object. Depending on the type of object there are two types of icons:
 - One bit: an icon represents the On state and another one represents the Off state.
 - Enumeration: up to six different states can be represented through one icon each.
- **Number:** the indicator will represent a number (received through the associated object) and a three-character text representing the units. Different number types are available for the different object types: Scaling, Counter (1-Byte unsigned), Counter (1-Byte signed), Counter (2-Bytes unsigned), Counter (2-Bytes signed), Counter (4-Bytes unsigned), Counter (4-Bytes signed), Float (2-Bytes).
- **Text:** The indicator will represent a text according to the value of an object. This indicator is totally analogous to the Icon indicator, but each value is associated to a text instead of an icon. The types are the same:
 - One bit: a text represents the On state and another one represents the Off state.
 - Enumeration: up to six different states can be represented through one text each.
- **Text from object:** Represents a string (up to 14 characters) received through the corresponding communication object.

- **[Climate] Temperature:** represents temperature values as received through the related communication object. Values in the range [-99, 199] °C and in the range [-142, 390] °F are available (depending on the selected temperature scale; see section 2.1.2). Values out of range will be replaced by the corresponding limit.

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

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

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

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

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

- **[Climate] On/Off + Mode + Status:** Climate status indicator with three one-bit objects associated: On/Off, Mode and Status. The icon represented depends on the value of these three objects.
- **[Climate] Mode Special:** works in the same way as **[Climate] Mode** in its extended version with the values mentioned in section 2.2.4.

ETS PARAMETERISATION

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

Moreover, the textbox **INFO** allows changing the default name of the tab in the left menu, as shows the following figure, as happen with buttons (see section 2.2.3).

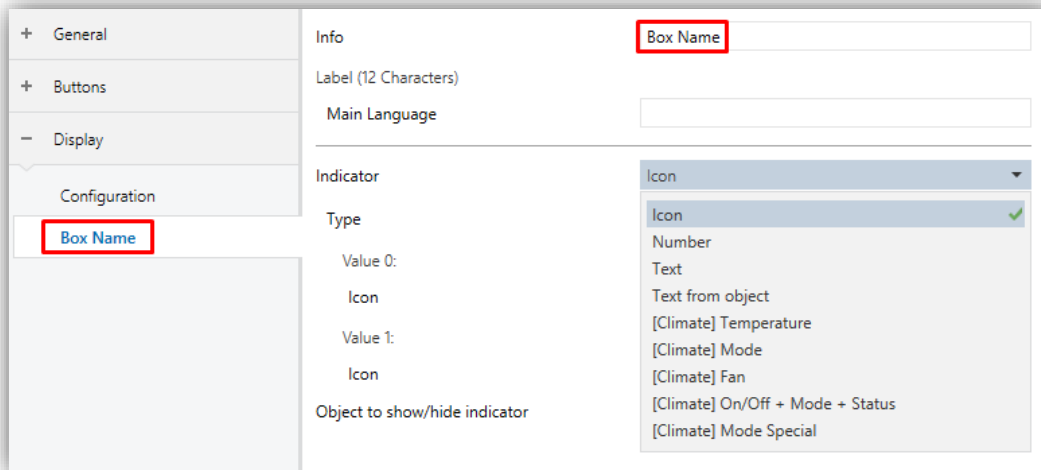


Figure 47. Display – Box a.

The main parameters that need to be configured are:

- **Label:**
 - **Language X:** identifying title for the box for the corresponding language
- **Indicator** [[Icon](#) / [Number](#) / [Text](#) / [Text from object](#) / [\[Climate\] Temperature](#) / [\[Climate\] Mode](#) / [\[Climate\] Fan](#) / [\[Climate\] On/Off + Mode + Status](#)]: sets the desired indicator to display in the box.
- **Representation** [[Permanent](#) / [Temporary](#) / [Intermittent](#) / [Progressive](#)]: sets the icon representation.

Note: *this parameter will remain hidden unless having selected “Every indicator is configured separately” in “Indicator representation” (see 2.3.1).*
- **Object to show/hide indicator** [[enabled/disabled](#)]: enables or disables a one-bit object (“[Disp] [x] Show/Hide indicator”) to show or hide the indicator in runtime.

Depending on the indicator, some more parameters are shown, as described next. Please note that in the next pages the general notation “[Disp][X]” is used for the name of the communication objects, as “x” depends on the display zone (a, b, etc.).

Icon

The screenshot shows the configuration window for the 'Icon' indicator. The 'Indicator' dropdown is set to 'Icon'. Under 'Type', the '1-bit' radio button is selected. For 'Value 0', the 'Icon' dropdown shows a power-off icon and the text 'Off', and the 'Representation' dropdown is set to 'Permanent'. For 'Value 1', the 'Icon' dropdown shows a power-on icon and the text 'On', and the 'Representation' dropdown is also set to 'Permanent'. The 'Object to show/hide indicator' checkbox is checked.

Figure 48. Display indicator – Icon – 1 bit.

Type:

- [1-bit]: the icon represented depends on the value of the one-bit object “[Disp] [X] Icon – 1-Bit”.

When selecting this type, the following parameters need to be configured:

- **Icon**: sets the icon to display for each value.

- [Enumeration]: the icon represented depends on the value of the one-byte object “[Disp] [X] Icon – Enumeration”.

When selecting this type, the following parameters need to be configured:

The screenshot shows the configuration window for the 'Icon' indicator with the 'Enumeration' type selected. The 'Indicator' dropdown is set to 'Icon'. Under 'Type', the 'Enumeration' radio button is selected. The '# Enums' spinner is set to 1. The 'Value 1' spinner is set to 0. The 'Icon' dropdown shows a power-off icon and the text 'Off', and the 'Representation' dropdown is set to 'Permanent'. The 'Object to show/hide indicator' checkbox is checked.

Figure 49. Display indicator – Icon – Enumeration.

- **# Enums** [[1...6](#)]: sets the number of states of the enumerated list.
 - **Value** [[0...255](#)]: sets the numerical value assigned to the state.
 - **Icon**: sets the icon to be displayed when the above value is received.

Number

Figure 50. Display indicator – Number – Scaling.

- **Type** [[Scaling](#) / *Counter (1-Byte unsigned)* / *Counter (1-Byte signed)* / *Counter (2-Bytes unsigned)* / *Counter (2-Bytes signed)* / *Counter (4-Bytes unsigned)* / *Counter (4-Bytes signed)* / *Float (2 bytes)*]: sets the type of the number indicator.
- **Font size** [[Small](#) / *Big* / *Extra*]: sets the font size of the number.

Text

Figure 51. Display indicator – Text – 1 bit.

• **Type:**

- [\[1 bit\]](#) the text represented depends on the value of the one-bit object “[Disp] [X] Text – 1-Bit”.

When selecting this type, the following parameters need to be configured.

- **Text for value 0/1:** sets the text to be displayed for each value.

- [\[Enumeration\]](#): the icon represented depends on the value of the one-byte object “[Disp] [x] Text – Enumeration”.

When selecting this type, the following parameters need to be configured.

The screenshot shows a configuration window titled 'Indicator' with a dropdown menu set to 'Text'. Below this, there are several settings:

- Type:** Two radio buttons are present: '1-bit' (unselected) and 'Enumeration' (selected).
- Font size:** Two radio buttons are present: 'Small' (selected) and 'Big' (unselected).
- How many elements?:** A numeric input field containing the value '1'.
- Value 1:** A numeric input field containing the value '0'.
- Text:** An empty text input field.
- Representation:** A dropdown menu set to 'Permanent'.
- Object to show/hide indicator:** A checkbox that is checked, indicated by a green checkmark icon.

Figure 52. Display indicator – Text – Enumeration.

- **How many elements?** [\[1...6\]](#): sets the number of states of the enumerated list. For each state, the following parameters need to be configured.
 - **Value** [\[0...255\]](#): sets the numerical value assigned to the state.
 - **Icon:** sets the text to be displayed when the above value is received.
- **Font size** [\[Small \(12 Characters\) / Big \(8 Characters\)\]](#): sets the font size of the texts.

Text from object

Indicator	Text from object
Font size	<input checked="" type="radio"/> Small <input type="radio"/> Big
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 53. Display indicator – Text from object.

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

- **Font size** [[Small](#) / [Big](#)]: sets the font size of the text.

[Climate] Temperature

Indicator	[Climate] Temperature
Font size	Extra
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 54 Display indicator – [Climate] Temperature.

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

- **Font size** [[Small](#) / [Big](#) / [Extra](#)]: sets the font size of the temperature label.

[Climate] Mode

Indicator	[Climate] Mode
Mode Type	<input checked="" type="radio"/> Heat/Cool <input type="radio"/> Extended
Representation	<input checked="" type="radio"/> Text <input type="radio"/> Icon
Font size	<input checked="" type="radio"/> Small (12 Characters) <input type="radio"/> Big (8 Characters)
Text for "Heat"	Heat
Text for "Cool"	Cool
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 55. Display indicator – [Climate] Mode – Heat/Cool.

● **Mode Type:**

- [\[Heat/Cool\]](#): the icon (Heat or Cool) will depend on the value of the “[Disp] [X] (Climate) Mode” one-bit object.
- [\[Extended\]](#): the icon represented will depend on the value of the “[Disp] [x] (Climate) Mode - Extended” one-byte object. When selecting this type, five checkboxes appear to select which HVAC modes should be available: **Auto / Heat / Cool / Fan / Dry** (as shown in Figure 56).

The screenshot shows a configuration window titled 'Indicator' with a dropdown menu set to '[Climate] Mode'. The 'Mode Type' section has two radio buttons: 'Heat/Cool' (unselected) and 'Extended' (selected). The 'Representation' section has two radio buttons: 'Text' (selected) and 'Icon' (unselected). The 'Font size' section has two radio buttons: 'Small (12 Characters)' (selected) and 'Big (8 Characters)' (unselected). Below these are five rows, each with a checked checkbox and a text input field: 'Auto' (input: Auto), 'Heat' (input: Heat), 'Cool' (input: Cool), 'Fan' (input: Fan), and 'Dry' (input: Dry). At the bottom, there is a 'Representation' dropdown set to 'Permanent' and an 'Object to show/hide indicator' checkbox which is checked.

Figure 56. Display indicator – [Climate] Mode – Mode Type: Extended.

● **Representation** [\[Text / Icon\]](#): sets whether the mode is displayed by means of a preset text or icon. the representation. If “[Text](#)” is selected, the following parameters will appear:

- **Font size** [\[Small \(12 Characters\) / Big \(8 Characters\)\]](#): sets the font size of the texts.
- **Text for each value**: sets the text to be displayed for each value.

[Climate] Fan

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

Figure 57. Display indicator – [Climate] Fan.

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

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

- **How to show Off/Auto status?** [\[Show Auto / Show Off / Clear the Box\]](#): sets how to represent the Auto/Off state.

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

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

- [\[1: Max. \(1% - 100%\)\]](#)
- [\[2: Min. \(1% - 50%\); Max. \(51% - 100%\)\]](#)
- [\[3: Min. \(1% - 33%\); Med. \(34% - 66%\); Max. \(67% - 100%\)\]](#)

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

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

- [\[2: Min. \(0% - 50%\); Max. \(51% - 100%\)\]](#)
- [\[3: Min. \(0% - 33%\); Med. \(34% - 67%\); Max. \(68% - 100%\)\]](#)

[Climate] On/Off + Mode + Status

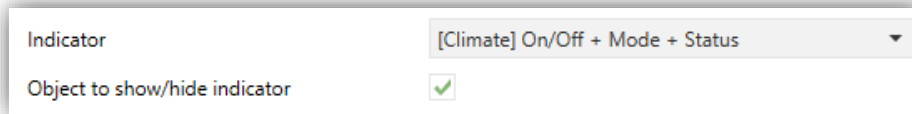


Figure 58. Display indicator – [Climate] On/Off + Mode + Status.

The icon to be represented depends on the value of three one-bit objects:

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

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

[Climate] Mode Special

- **Representation** [Text / Icon]: Sets whether the mode is displayed on screen by means of text or a preset icon. This representation depends on the value of the object in one byte “**[Disp] [x] (Climate) Mode Special**”. If “Text” is selected, the following parameters will appear:
 - **Font size** [Small (12 Characters) / Big (8 Characters)]: sets the font size of the texts.
 - **Text for each value**: sets the text to be displayed for each value.

Indicator	[Climate] Mode Special
Representation	<input checked="" type="radio"/> Text <input type="radio"/> Icon
Font size	<input checked="" type="radio"/> Small (12 Characters) <input type="radio"/> Big (8 Characters)
Auto	<input checked="" type="checkbox"/>
Text for "Auto"	Auto
Comfort	<input checked="" type="checkbox"/>
Text for "Comfort"	Comfort
Standby	<input checked="" type="checkbox"/>
Text for "Standby"	Standby
Economy	<input checked="" type="checkbox"/>
Text for "Economy"	Economy
Building Protection	<input checked="" type="checkbox"/>
Text for "Building Protection"	Protect
Representation	Permanent
Object to show/hide indicator	<input checked="" type="checkbox"/>

Figure 59. [Climate] Mode Special

- **Auto / Comfort / Standby / Economy / Building Protection**
[disabled/enabled]: five checkboxes to select which of the special modes will be available.

2.3.4 DOUBLE

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

Double boxes can be assigned exactly the same types of indicators as Single boxes (see section 2.3.3 for details).

The only differences between the two types of boxes are:

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

ETS PARAMETERISATION

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

The textbox **INFO** also allows changing the default name of the tab in the left menu.

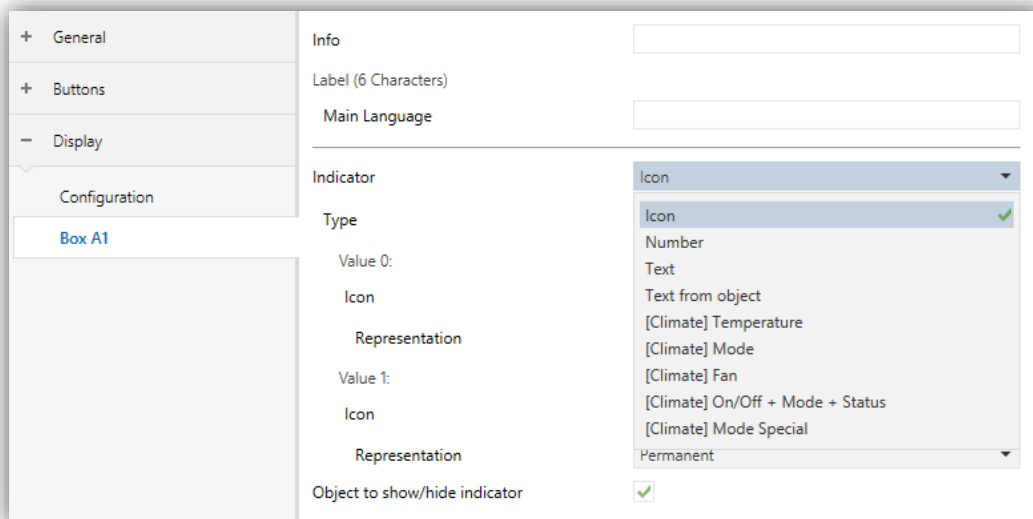


Figure 60. Display – Box a1.

The ETS parameterisation of each indicator is exactly the same as for Single boxes. See section 2.3.3 for details.

2.3.5 SETPOINT

Thermostat type display will have a larger central box (always enabled) for showing the setpoint temperature of an external thermostat. Displaying a second temperature, real temperature, is also possible.

On the other hand, at the top of the display, a moving icon is shown to indicate whether the thermostat is operating. The icon is displayed when a "1" is received through the binary object "**[Disp][Setpoint] Thermostat status**".

The setpoint box also has an alarm to link, for example, to a window opening sensor. The alarm trigger is received through the object "**[Disp][Setpoint] Alarm**", causing the thermostatic indicator to disappear and an icon and text to appear. Enabling and disabling the alarm functionality at runtime is also possible through the object "**[Disp][Setpoint] Alarm Enable**".

ETS PARAMETERISATION

When the display is configured as thermostat, a specific tab ("Box B") becomes available under "Display" in the tree on the left.

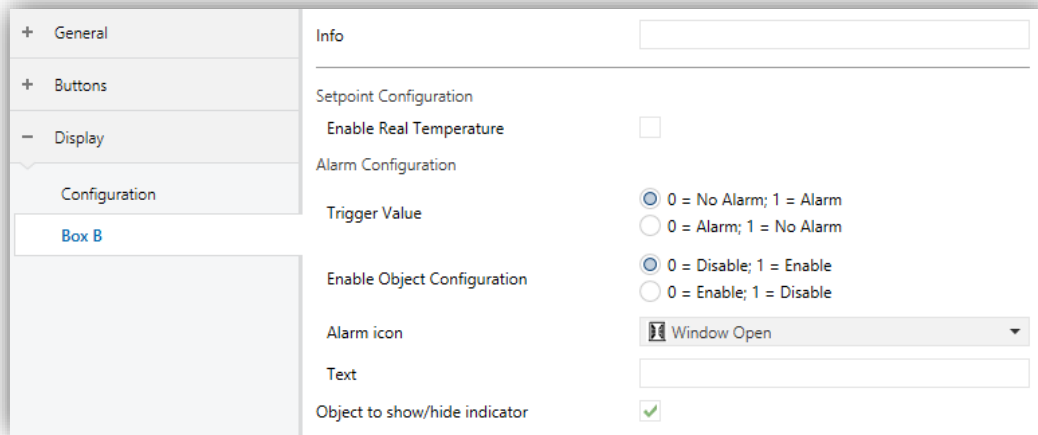


Figure 61. Display - Thermostat - Setpoint (Box B)

This tab contains the following parameters:

● Setpoint Configuration:

- **Enable Real Temperature** [*disabled* / *enabled*]: when enabled the display will show, in addition to the setpoint temperature, the actual temperature. The following parameter will also appear:
 - **Main Temperature to Show** [*Setpoint* / *Real*]: sets whether the temperature displayed centered and larger is the setpoint or the actual temperature.

The setpoint temperature is received through the object "[Disp][Setpoint] Setpoint Temperature" and the real temperature through the object "[Disp][Setpoint] Real temperature".

● Alarm Configuration:

- **Trigger Value** [*0 = No Alarm; 1 = Alarm* / *0 = Alarm; 1 = No Alarm*]: polarity of the binary alarm binary object "[Disp][Setpoint] Alarm".

- **Enable Object Configuration** [*0 = Disable; 1 = Enable* / *0 = Enable; 1 = Disable*]: polarity of the alarm enabling object "[Disp][Setpoint] Alarm Enable".
- **Alarm icon** [*Window Open*]: drop-down menu to choose the icon shown when the alarm is triggered.
 - **Representation** [*Permanent* / *Temporary* / *Intermittent* / *Progressive*].
- **Text**: text that appears in the display when the alarm is activated.
- **Object to show/hide indicator** [*enabled/disabled*].

2.3.6 FAN

Fan boxes show ventilation speed indicators on the right side of the screen in a vertical arrangement.

When this box is enabled, the 1-byte status object "[Disp][Fan] Fan Indicator" appears. The status object (which needs to be linked to the status object of the fan actuator) will express, in percentage, the value of the current fan level, which also determine the icon that will be shown on the box.

ETS PARAMETERISATION

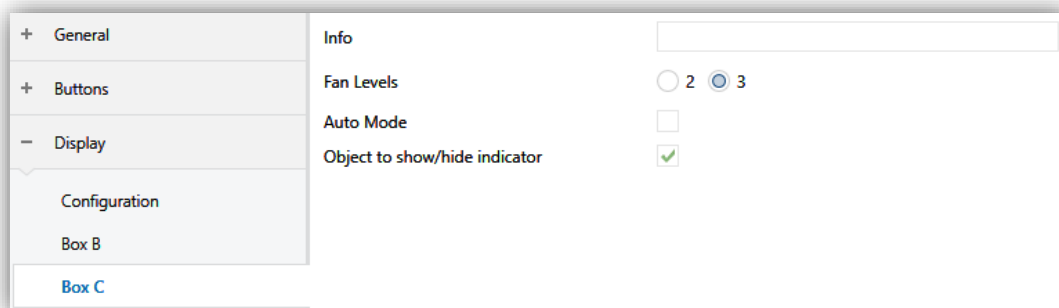


Figure 62. Display - Thermostat - Fan (Box C)

This tab contains the following parameters:

- **Fan Levels** [*2* / *3*]: sets how many speed levels will be available in the indicator.

- **Auto Mode** [[disabled/enabled](#)]: sets whether the fan Auto mode will be available. If checked, the Auto fan mode will be activated by switching to fan level 0% and the following parameter shows up:
 - **Dedicated Object for Auto Mode** [[disabled / enabled](#)]: marking the checkbox enables the 1-bit object “[Disp][Fan] Auto Mode Value”, which will trigger the auto mode when it receives the corresponding value set in **Value to Activate Auto Mode** [[Receive 0 to Set Auto Mode / Receive 1 to Set Auto Mode](#)].

Figure 63. Thermostat Page - Fan - Auto Mode

- **Object to show/hide indicator** [[disabled / enabled](#)].

2.3.7 BORDERS

Divisions between the boxes can be drawn in three different styles: with **solid lines**, **dashed** (dotted), or **invisible** (without division line). This configuration may be common to all lines or separate for each, as shown in Figure 64.

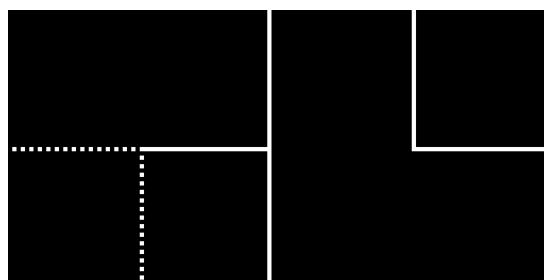


Figure 64. Borders – Different configuration for each border.

ETS PARAMETERISATION

After selecting “Customize every border” in the **Border lines** parameter (see section 2.3.1), a new tab will be incorporated into the tree on the left.

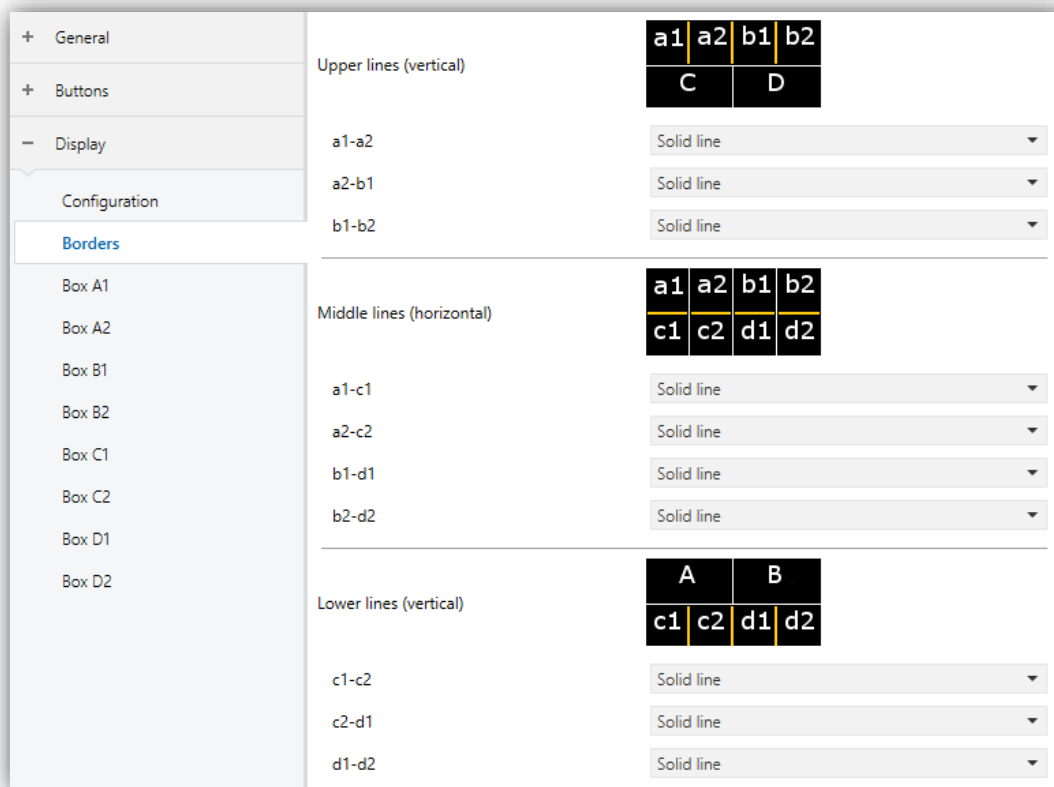


Figure 65. Display – Borders (Normal Page)

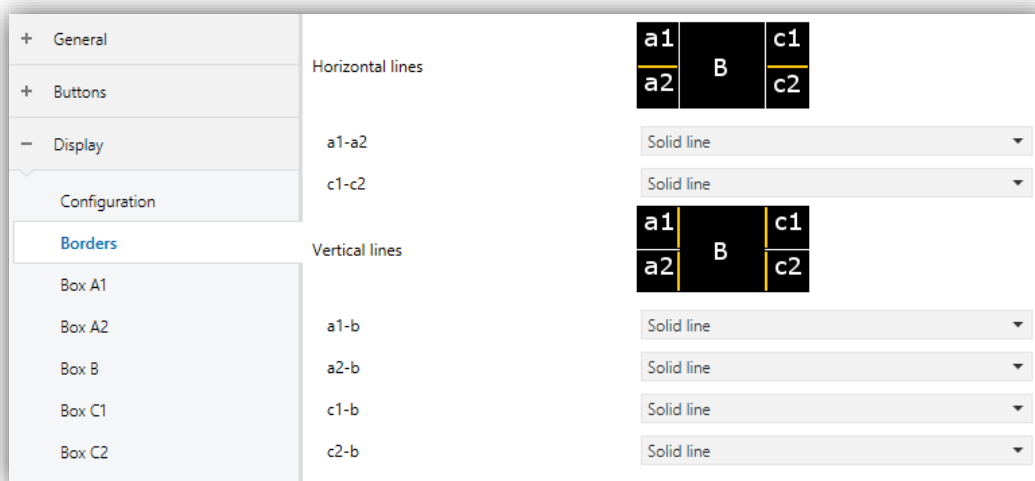


Figure 66. Display – Borders (Thermostat Page)

Depending on the page style selected in the “Configuration” tab (see section 2.3.1), this screen will be divided into the following parts:

● Normal Page:

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

● **Thermostat Page:**

- **Horizontal lines:** configures the horizontal lines **a1-a2** y **c1-c2**.
- **Vertical lines:** configures the vertical lines **a1-b**, **a2-b**, **c1-b** y **c2-b**.

The options available for all the lines are: [[Solid line](#) / *Invisible* / *Dotted line*].

2.4 INPUTS

Flat Display incorporates **two 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 from Zennio.
- **Motion Detector**, to connect a motion detector (suitable for the new motion detector model from Zennio).

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

2.4.1 BINARY INPUT

Please refer to the specific user manual “**Binary Inputs**”, available in the Flat Display product section, at the Zennio website (www.zennio.com).

2.4.2 TEMPERATURE PROBE

Please refer to the specific user manual “**Temperature Probe**”, available in the Flat Display product section, at the Zennio website (www.zennio.com).

2.4.3 MOTION DETECTOR

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

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

Notes:

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

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

2.5 THERMOSTAT

Flat Display implements **one Zennio thermostat** which can be enabled and fully customised.

Please refer to the specific manual “**Zennio Thermostat**” (available in the Flat Display product section at the Zennio website, www.zennio.com) for detailed information about the functionality and the configuration of the related parameters.

ANNEX I. LED ILLUMINATION MODES

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

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

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

Table 5 illustrates which of the above are configurable for each function.

		Disabled	Regular	State-dep.	State-dep. (both LEDs)	Dedicated object
INDIVIDUAL	LED Indicator					✓
	Switch		✓	✓		✓
	Hold & Release		✓			✓
	Two Objects		✓	✓		✓
	Scene		✓			✓
	Constants		✓	✓		✓
	Dimmer		✓	✓		✓
	Shutter		✓			✓
	Enumeration		✓			✓
	Fan		✓	✓		✓
	Room State		✓	✓		✓
PAIR	Switch		✓	✓	✓	✓
	Two Objects		✓	✓	✓	✓
	Dimmer		✓	✓	✓	✓
	Shutter		✓			✓
	Scaling, Counter, Float		✓			✓
	Enumeration		✓			✓
	Multimedia		✓	✓	✓	✓
	Temperature Set-Point		✓			✓
	Mode: Heat/Cool		✓	✓	✓	✓
	Mode: Extended		✓			✓
	Mode Special		✓			✓
	Fan		✓	✓	✓	✓
DISABLED		✓				

Table 5 Functions vs. LED Illumination Options.

Note:

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

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

Although the behaviour of the LEDs can be configured independently for each control, it is also possible to define a **general behaviour for all of them** thus not being then necessary to configure the same option multiple times.

In case of opting for a general configuration, the options are:

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

ETS PARAMETERISATION

For details on the parameterisation of the LED illumination modes please refer to the pages that cover the specific function being assigned to the button (see 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 options of the buttons general configuration (section 2.2.1).

ANNEX II. COMMUNICATION OBJECTS

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

Number	Size	I/O	Flags	Data type (DPT)	Functional Range	Name	Function
1	1 Bit		CT---	DPT_Trigger	0/1	[Heartbeat] Object to Send '1'	Sending of '1' Periodically
2	3 Bytes	I	CT-WU	DPT_TimeOfDay	00:00:00 - 23:59:59	[General] Time of Day	Time of Day External Reference
3	3 Bytes	I	CT-WU	DPT_Date	01/01/1990 - 31/12/2089	[General] Date	Date External Reference
4	1 Byte	I	C--W-	DPT_SceneNumber	0-63	[General] Scene: Receive	0-63 (Run Scene 1-64)
5	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[General] Scene: Send	0-63/128-191 (Run/Save Scene 1-64)
6	1 Bit	I	C--W-	DPT_State	0/1	[General] Activity	0 = Inactivity; 1 = Activity
7	1 Bit	I	CT-WU	DPT_Enable	0/1	[General] Touch Locking	0 = Unlock; 1 = Lock
	1 Bit	I	CT-WU	DPT_Enable	0/1	[General] Touch Locking	0 = Lock; 1 = Unlock
8	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Cleaning Function	0 = Nothing; 1 = Clean Now
9	2 Bytes	I	CT-WU	DPT_Value_Temp	-273.00 - 670760.00	[General] External Temperature	Temperature to Be Shown
10	1 Bit	I	C--W-		0/1	[General] Backlight Mode	0 = Night Mode; 1 = Normal Mode
	1 Bit	I	C--W-		0/1	[General] Backlight Mode	0 = Normal Mode; 1 = Night Mode
11	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[General] Display - Brightness	0% ... 100%
12	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[General] Display - Contrast	0% ... 100%
13	1 Bit	I	C--W-	DPT_Enable	0/1	[General] Proximity Sensor	0 = Disable; 1 = Enable
14	1 Bit	I	C--W-	DPT_Ack	0/1	[General] External Proximity Detection	1 = Detection
15	1 Bit		CT---	DPT_Ack	0/1	[General] Proximity Detection	Send 1 when Proximity is Detected
16	1 Bit		CT---	DPT_Bool	0/1	[General] Luminosity (1-Bit)	0 = Over Threshold; 1 = Under Threshold
	1 Bit		CT---	DPT_Bool	0/1	[General] Luminosity (1-Bit)	0 = Under Threshold; 1 = Over Threshold
17	1 Byte	O	C-R--	DPT_Scaling	0% - 100%	[General] Luminosity (Percentage)	0% ... 100%
19	1 Bit	I	CT-W-	DPT_Switch	0/1	[General] Welcome Greeting	0 = Finish Welcome; 1 = Run Welcome
20, 21, 22, 23	14 Bytes	I	C--W-			[General] Welcome Greeting - Line x	Text to Show On the Display at Line x
24	1 Byte	I	C--W-		0 - 5	[General] Translations - Select Language	0 = Main; 1 = Lang. 2; ...; 4 = Lang. 5
25	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Translations - Main Language	0 = Nothing; 1 = Choose this Language

26, 27, 28, 29	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Translations - Language x	0 = Nothing; 1 = Choose this Language
30	2 Bytes	I	C - - W -	DPT_LanguageCodeAlpha2_ASCII		[General] Translations - Select Language	Language Selection Through ISO 639-1 Two Letters Code
31	1 Bit	I	C - - W -	DPT_Enable	0/1	[General] Sounds - Disabling Button Sound	0 = Disable Sound; 1 = Enable Sound
	1 Bit	I	C - - W -	DPT_Enable	0/1	[General] Sounds - Disabling Button Sound	0 = Enable Sound; 1 = Disable Sound
32	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Sounds - Doorbell	0 = Play Doorbell Sound; 1 = Nothing
	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Sounds - Doorbell	0 = Nothing; 1 = Play Doorbell Sound
33	1 Bit	I	C - - W -	DPT_Alarm	0/1	[General] Sounds - Alarm	0 = Start Alarm; 1 = Stop Alarm
	1 Bit	I	C - - W -	DPT_Alarm	0/1	[General] Sounds - Alarm	0 = Stop Alarm; 1 = Start Alarm
34	1 Bit		C T - - -	DPT_Switch	0/1	[General] Welcome Back Object	Switch Object Sent on Wake Up
35, 36, 37, 38, 39	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome Back Object - Additional Condition	Additional Condition Object x
40	1 Bit	I	C T - W U	DPT_Switch	0/1	[General] Temperature Scale	0 = °C; 1 = °F
41, 48, 55, 62, 69	1 Bit		C T - - -	DPT_Switch	0/1	[Btn][Ix] Light - On/Off	(Short press) Switch between On and Off
	1 Bit		C T - - -	DPT_Step	0/1	[Btn][Ix] Shutter - Stop/Step	(Short press) 0 = Stop shutter/Step up; 1 = Stop shutter/Step down
	1 Bit		C T - - -	DPT_Trigger	0/1	[Btn][Ix] Shutter - Stop	(End pressing) Stop shutter
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Switch: "0"	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Switch: "1"	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Switch: "0/1"	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Hold & Release	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Two Objects - Short Press: "1"	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Two Objects - Short Press: "0"	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Two Objects - Short Press: "0/1"	1-Bit Generic Control
42, 49, 56, 63, 70	1 Bit		C T - - -	DPT_Step	0/1	[Btn][Ix] Fan Control (Type: 1-Bit)	0 = Decrease; 1 = Increase
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn][Ix] Shutter - Move	(Long press) 0 = Up; 1 = Down
	1 Bit		C T - - -	DPT_UpDown	0/1	[Btn][Ix] Shutter - Move	(Start pressing) Switch between up and down
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Two Objects - Long Press: "1"	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Two Objects - Long Press: "0"	1-Bit Generic Control
	1 Bit	I	C T - W -	DPT_Switch	0/1	[Btn][Ix] Two Objects - Long Press: "0/1"	1-Bit Generic Control

43, 50, 57, 64, 71	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Ix] Fan Control - Auto Mode	Switch Auto Mode on Short Press
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Ix] Fan Control - Auto Mode	Switch Auto Mode on Long Press
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Ix] LED On/Off	0 = Off; 1 = On
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Ix] LED On/Off	0 = On; 1 = Off
44, 51, 58, 65, 72	4 Bit	I	CT-W-	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Btn][Ix] Light - Dimming	(Long press) Switch between dimming up and down
45, 52, 59, 66, 73	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Enumeration	Rotating value change
	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Ix] Shutter Position	0 - 100 %
	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Ix] Light Dimming (Status)	0 - 100 %
	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] 1-Byte Unsigned Int Value	0 ... 255
	1 Byte	I	CT-W-	DPT_Value_1_Count	-128 - 127	[Btn][Ix] 1-Byte Signed Int Value	-128 ... 127
	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Two Objects - Short Press: (1-Byte)	Send Selected 1-Byte Value on Short Press
	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Ix] 1-Byte Scaling Value	0% ... 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	50%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	33%, 67%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	25%, 50%, 75%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	20%, 40%, 60%, 80%, 100%
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	1
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	1, 2
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	1, 2, 3
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	1, 2, 3, 4
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	1, 2, 3, 4, 5
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	0, 1
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	0, 1, 2, 3, 4, 5
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	0, 1, 2, 3, 4
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	0, 1, 2, 3

						Enumeration)	
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	0, 1, 2
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	0%, 20%, 40%, 60%, 80%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	0%, 33%, 67%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	0%, 50%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	0%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	0%, 25%, 50%, 75%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	Auto, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	Auto, 20%, 40%, 60%, 80%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	Auto, 25%, 50%, 75%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	Auto, 33%, 67%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Control (Type: Scaling)	Auto, 50%, 100%
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	Auto, 1
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	Auto, 1, 2
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	Auto, 1, 2, 3
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	Auto, 1, 2, 3, 4
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Control (Type: Enumeration)	Auto, 1, 2, 3, 4, 5
	1 Byte	I	CT-W-	DPT_SceneControl	0-63; 128-191	[Btn][Ix] Scene: Send	0-63 (Run Scene 1-64)
	1 Byte	I	CT-W-	DPT_SceneControl	0-63; 128-191	[Btn][Ix] Scene: Send	0-63/128-191 (Run/Save Scene 1-64)
	1 Byte	I	CT-W-		0 - 2	[Btn][Ix] Room State	0 = Normal; 1 = Make-up room; 2 = Do not disturb
46, 53, 60, 67, 74	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Two Objects - Long Press: (1-Byte)	Send Selected 1-Byte Value on Long Press
	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Ix] Fan Indicator	0% ... 100%
	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Ix] Fan Indicator	Enumerated Value
47, 54, 61, 68, 75	2 Bytes	I	CT-W-	DPT_Value_2_Ucount	0 - 65535	[Btn][Ix] 2-Byte Unsigned Int Value	0 ... 65535
	2 Bytes	I	CT-W-	DPT_Value_2_Count	-32768 - 32767	[Btn][Ix] 2-Byte Signed Int Value	-32768 ... 32767
	2 Bytes	I	CT-W-	9.xxx	-671088.64 - 670760.96	[Btn][Ix] 2-Byte Float Value	-671088.64 ... 670760.96
76, 83	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Switch	Left = 0; Right = 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Two Objects - Short press	Left = 1; Right = 0
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Two Objects - Short press	Left = 0; Right = 1
	1 Bit		CT---	DPT_Switch	0/1	[Btn][Px] Light - On/Off	(Short press) Left = Off; Right = On
	1 Bit		CT---	DPT_Step	0/1	[Btn][Px] Shutter - Stop/Step	(Short press) Left = Stop/Step down; Right = Stop/Step up

	1 Bit		CT---	DPT_Trigger	0/1	[Btn][Px] Shutter - Stop	(End pressing) Left = Stop-down; Right = Stop-up
	1 Bit	I	CT-W-	DPT_Heat_Cool	0/1	[Btn][Px] (Climate) Mode	Left = Cool; Right = Heat
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Switch	Left = 1; Right = 0
	1 Bit		CT---	DPT_Switch	0/1	[Btn][Px] Light - On/Off	(Short press) Left = On; Right = Off
	1 Bit		CT---	DPT_Step	0/1	[Btn][Px] Shutter - Stop/Step	(Short press) Left = Stop/Step up; Right = Stop/Step down
	1 Bit		CT---	DPT_Trigger	0/1	[Btn][Px] Shutter - Stop	(End pressing) Left = Stop-up; Right = Stop-down
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Multimedia	Left = Stop/Backward; Right = Start/Forward
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Multimedia	Left = Start/Forward; Right = Stop/Backward
	1 Bit		CT---	DPT_Step	0/1	[Btn][Px] Fan Control (Type: 1-Bit)	0 = Decrease; 1 = Increase
77, 84	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Two Objects - Long press	Left = 0; Right = 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Two Objects - Long press	Left = 1; Right = 0
	1 Bit		CT---	DPT_UpDown	0/1	[Btn][Px] Shutter - Move	(Long press) Left = Down; Right = Up
	1 Bit		CT---	DPT_UpDown	0/1	[Btn][Px] Shutter - Move	(Start pressing) Left = Down; Right = Up
	1 Bit		CT---	DPT_UpDown	0/1	[Btn][Px] Shutter - Move	(Long press) Left = Up; Right = Down
	1 Bit		CT---	DPT_UpDown	0/1	[Btn][Px] Shutter - Move	(Start pressing) Left = Up; Right = Down
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Fan Control - Auto Mode	Switch Auto Mode on Long Press
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] Fan Control - Auto Mode	Switch Auto Mode on Short Press
78, 85	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] LED On/Off	0 = On; 1 = Off
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Btn][Px] LED On/Off	0 = Off; 1 = On
79, 86	4 Bit	I	CT-W-	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Btn][Px] Light - Dimming	(Long press) Left = Darker; Right = Brighter
	4 Bit	I	CT-W-	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Btn][Px] Light - Dimming	(Long press) Left = Brighter; Right = Darker
80, 87	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Px] Scaling	Left = Decrease percentage; Right =

							Increase percent.
1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Px] Scaling		Left = Increase percentage; Right = Decrease percent.
1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Counter - 1-Byte unsigned		Left = Decrease; Right = Increase
1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Counter - 1-Byte unsigned		Left = Increase; Right = Decrease
1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Enumeration		Rotating value change on short press
1 Byte	I	CT-W-	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	[Btn][Px] (Climate) Mode - Extended		Rotating change of selected modes on short press
1 Byte	I	CT-W-	DPT_Value_1_Count	-128 - 127	[Btn][Px] Counter - 1-Byte signed		Left = Decrease; Right = Increase
1 Byte	I	CT-W-	DPT_Value_1_Count	-128 - 127	[Btn][Px] Counter - 1-Byte signed		Left = Increase; Right = Decrease
1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Px] Light Dimming (Status)		0 - 100 %
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)		100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)		50%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)		33%, 67%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)		25%, 50%, 75%, 100%
1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)		20%, 40%, 60%, 80%, 100%
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		1
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		1, 2
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		1, 2, 3
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		1, 2, 3, 4
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		1, 2, 3, 4, 5
1 Byte	I	CT-W-	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Btn][Px] (Climate) Special Mode Control		Auto, Comfort, Standby, Economy, Building Protection
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		0, 1
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		0, 1, 2, 3, 4, 5
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		0, 1, 2, 3, 4
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		0, 1, 2, 3
1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)		0, 1, 2

	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	0%, 20%, 40%, 60%, 80%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	0%, 33%, 67%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	0%, 50%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	0%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	0%, 25%, 50%, 75%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	Auto, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	Auto, 20%, 40%, 60%, 80%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	Auto, 25%, 50%, 75%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	Auto, 33%, 67%, 100%
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Btn][Px] Fan Control (Type: Scaling)	Auto, 50%, 100%
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)	Auto, 1
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)	Auto, 1, 2
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)	Auto, 1, 2, 3
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)	Auto, 1, 2, 3, 4
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Control (Type: Enumeration)	Auto, 1, 2, 3, 4, 5
81, 88	1 Byte	I	CT-W-	DPT_Scaling	0% - 100%	[Btn][Px] Fan Indicator	0% ... 100%
	1 Byte	I	CT-W-	DPT_Value_1_Ucount	0 - 255	[Btn][Px] Fan Indicator	Enumerated Value
82, 89	2 Bytes	I	CT-W-	DPT_Value_2_Count	-32768 - 32767	[Btn][Px] Counter - 2-Byte signed	Left = Decrease; Right = Increase
	2 Bytes	I	CT-W-	DPT_Value_2_Count	-32768 - 32767	[Btn][Px] Counter - 2-Byte signed	Left = Increase; Right = Decrease
	2 Bytes	I	CT-W-	9.xxx	-671088.64 - 670760.96	[Btn][Px] Float	Left = Decrease; Right = Increase
	2 Bytes	I	CT-W-	9.xxx	-671088.64 - 670760.96	[Btn][Px] Float	Left = Increase; Right = Decrease
	2 Bytes	I	CT-W-	DPT_Value_Temp	-273.00 - 670760.00	[Btn][Px] (Climate) Temperature setpoint	Left = Decrease; Right = Increase
	2 Bytes	I	CT-W-	DPT_Value_Temp	-273.00 - 670760.00	[Btn][Px] (Climate) Temperature setpoint	Left = Increase; Right = Decrease
	2 Bytes	I	CT-W-	DPT_Value_2_Ucount	0 - 65535	[Btn][Px] Counter - 2-Byte unsigned	Left = Decrease; Right = Increase
	2 Bytes	I	CT-W-	DPT_Value_2_Ucount	0 - 65535	[Btn][Px] Counter - 2-Byte unsigned	Left = Increase; Right = Decrease
90, 98, 106, 114, 122, 130, 138, 146	1 Bit	I	C--W-	DPT_Switch	0/1	[Disp][ax] Show/Hide Box	0 = Hide Box; 1 = Show Box
91, 99, 107, 115, 123, 131, 139, 147	1 Bit	I	CT-WU	DPT_Switch	0/1	[Disp][ax] Icon - 1-Bit	Selected icon shown on new value reception
	1 Bit	I	CT-WU	DPT_Heat_Cool	0/1	[Disp][ax] (Climate) Mode	Mode Indicator Is Shown (0 = Cool; 1 = Heat)
	1 Bit	I	CT-WU	DPT_Switch	0/1	[Disp][ax] Text - 1-Bit	Parameterized text shown on new

							value reception
	1 Bit	I	CT - W U	DPT_Switch	0/1	[Disp][ax] (Climate) On/Off	0 = Off (hide indicator); 1 = On (show mode or status)
92, 100, 108, 116, 124, 132, 140, 148	1 Bit	I	CT - W U	DPT_Heat_Cool	0/1	[Disp][ax] (Climate) On/Off - Mode	0 = Cool; 1 = Heat. (Mode Indicator Is Shown when On/Off = 1, and Status = 0)
93, 101, 109, 117, 125, 133, 141, 149	1 Bit	I	CT - W U	DPT_Switch	0/1	[Disp][ax] (Climate) On/Off - Status	0 = Halted; 1 = Running. (Status icon is shown when On/Off = 1, and Status = 1)
94, 102, 110, 118, 126, 134, 142, 150	1 Byte	I	CT - W U	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	[Disp][ax] (Climate) Mode - Extended	Mode Indicator Is Shown
	1 Byte	I	CT - W U	DPT_Scaling	0% - 100%	[Disp][ax] Fan	3 levels: Min. (0% - 33%); Med. (34% - 66%); Max. (67% - 100%)
	1 Byte	I	CT - W U	DPT_Scaling	0% - 100%	[Disp][ax] Fan	2 levels: Min. (0% - 50%); Max. (51% - 100%)
	1 Byte	I	CT - W U	DPT_Value_1_Count	-128 - 127	[Disp][ax] Number - Counter (1-Byte signed)	Object numeric value is shown
	1 Byte	I	CT - W U	DPT_Value_1_Ucount	0 - 255	[Disp][ax] Number - Counter (1-Byte unsigned)	Object numeric value is shown
	1 Byte	I	CT - W U	DPT_Scaling	0% - 100%	[Disp][ax] Number - Scaling	Object numeric value is shown
	1 Byte	I	CT - W U	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Disp][ax] (Climate) Mode Special	Mode Indicator Is Shown
	1 Byte	I	CT - W U	DPT_Value_1_Ucount	0 - 255	[Disp][ax] Text - Enumeration	Parameterized text shown on new value reception
	1 Byte	I	CT - W U	DPT_Scaling	0% - 100%	[Disp][ax] Fan	Off/Auto + 2 levels: Off/Auto = 0%; Min. (1% - 50%); Max. (51% - 100%)
	1 Byte	I	CT - W U	DPT_Scaling	0% - 100%	[Disp][ax] Fan	Off/Auto + 3 levels: Off/Auto = 0%; Min. (1% - 33%); Med. (34% - 66%); Max. (67% - 100%)
	1 Byte	I	CT - W U	DPT_Value_1_Ucount	0 - 255	[Disp][ax] Icon - Enumeration	Selected icon shown on new value reception
	1 Byte	I	CT - W U	DPT_Scaling	0% - 100%	[Disp][ax] Fan	Off/Auto + 1 level: Off/Auto = 0%; Max. (1% - 100%)
95, 103, 111, 119, 127, 135, 143, 151	2 Bytes	I	CT - W U	DPT_Value_2_Ucount	0 - 65535	[Disp][ax] Number - Counter (2-Byte unsigned)	Object numeric value is shown
	2 Bytes	I	CT - W U	DPT_Value_2_Count	-32768 - 32767	[Disp][ax] Number - Counter (2-Byte signed)	Object numeric value is shown
	2 Bytes	I	CT - W U	DPT_Value_Temp	-273.00 - 670760.00	[Disp][ax] (Climate) Temperature	Object value is shown (-99°C to 199°C)

	2 Bytes	I	CT-WU	9.xxx	-671088.64 - 670760.96	[Disp][ax] Number - Float (2-Byte)	Object numeric value is shown
96, 104, 112, 120, 128, 136, 144, 152	4 Bytes	I	CT-WU	DPT_Value_4_Count	-2147483648 - 2147483647	[Disp][ax] Number - Counter (4-Byte signed)	Object numeric value is shown
	4 Bytes	I	CT-WU	DPT_Value_4_Ucount	0 - 4294967295	[Disp][ax] Number - Counter (4-Byte unsigned)	Object numeric value is shown
97, 105, 113, 121, 129, 137, 145, 153	14 Bytes	I	CT-WU			[Disp][ax] Text from object	Received text is shown
106	1 Bit	I	C--W-	DPT_Switch	0/1	[Disp][Setpoint] Show/Hide Box	0 = Hide Box; 1 = Show Box
111, 119	2 Bytes	I	CT-WU	DPT_Value_Temp	-273.00 - 670760.00	[Disp][Setpoint] Real Temperature	-99°C ... 199°C
	2 Bytes	I	CT-WU	DPT_Value_Temp	-273.00 - 670760.00	[Disp][Setpoint] Setpoint Temperature	-99°C ... 199°C
122	1 Bit	I	C--W-	DPT_Switch	0/1	[Disp][Fan] Show/Hide Box	0 = Hide Box; 1 = Show Box
123	1 Bit	I	CT-WU	DPT_Switch	0/1	[Disp][Fan] Auto Mode Value	Auto Mode With 0
	1 Bit	I	CT-WU	DPT_Switch	0/1	[Disp][Fan] Auto Mode Value	Auto Mode With 1
126	1 Byte	I	CT-WU	DPT_Scaling	0% - 100%	[Disp][Fan] Fan Indicator	3 levels: Min. (0% - 33%); Med. (34% - 66%); Max. (67% - 100%)
	1 Byte	I	CT-WU	DPT_Scaling	0% - 100%	[Disp][Fan] Fan Indicator	2 levels: Min. (0% - 50%); Max. (51% - 100%)
	1 Byte	I	CT-WU	DPT_Scaling	0% - 100%	[Disp][Fan] Fan Indicator	Auto + 3 levels: Auto = 0%; Min. (1% - 33%); Med. (34% - 66%); Max. (67% - 100%)
	1 Byte	I	CT-WU	DPT_Scaling	0% - 100%	[Disp][Fan] Fan Indicator	Auto + 2 levels: Auto = 0%; Min. (1% - 50%); Max. (51% - 100%)
139	1 Bit	I	C--W-	DPT_Enable	0/1	[Disp][Setpoint] Alarm Enable	0 = Disable; 1 = Enable
	1 Bit	I	C--W-	DPT_Enable	0/1	[Disp][Setpoint] Alarm Enable	0 = Enable; 1 = Disable
140	1 Bit	I	CT-WU	DPT_Alarm	0/1	[Disp][Setpoint] Alarm	0 = No Alarm; 1 = Alarm
	1 Bit	I	CT-WU	DPT_Alarm	0/1	[Disp][Setpoint] Alarm	0 = Alarm; 1 = No Alarm
147	1 Bit	I	CT-WU	DPT_Switch	0/1	[Disp][Setpoint] Thermostat State	0 = Off; 1 = On
154, 158	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Ix] Current Temperature	Temperature Sensor Value
155, 159	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overcooling	0 = No Alarm; 1 = Alarm
156, 160	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overheating	0 = No Alarm; 1 = Alarm
157, 161	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Probe Error	0 = No Alarm; 1 = Alarm
162	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Internal Temp. Probe] Current Temperature	Temperature Sensor Value
163	1 Bit	O	CTR--	DPT_Alarm	0/1	[Internal Temp. Probe] Overcooling	0 = No Alarm; 1 = Alarm
164	1 Bit	O	CTR--	DPT_Alarm	0/1	[Internal Temp. Probe] Overheating	0 = No Alarm; 1 = Alarm
165, 171	1 Bit	I	C--W-	DPT_Enable	0/1	[Ix] Input Lock	0 = Unlock; 1 = Lock
166, 172	1 Bit		CT---	DPT_Switch	0/1	[Ix][Short Press] 0	Sending of 0
	1 Bit		CT---	DPT_Switch	0/1	[Ix][Short Press] 1	Sending of 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix][Short Press] 0/1 Switching	Switching 0/1
	1 Bit		CT---	DPT_UpDown	0/1	[Ix][Short Press] Move Up Shutter	Sending of 0 (Up)

	1 Bit		CT---	DPT_UpDown	0/1	[Ix][Short Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix][Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix][Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Bit		CT---	DPT_Step	0/1	[Ix][Short Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix][Short Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix][Short Press] Brighter	Increase Brightness
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix][Short Press] Darker	Decrease Brightness
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix][Short Press] Brighter/Darker	Switch Bright/Dark
	1 Bit		CT---	DPT_Switch	0/1	[Ix][Short Press] Light On	Sending of 1 (On)
	1 Bit		CT---	DPT_Switch	0/1	[Ix][Short Press] Light Off	Sending of 0 (Off)
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix][Short Press] Light On/Off	Switching 0/1
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix][Short Press] Run Scene	Sending of 0 - 63
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix][Short Press] Save Scene	Sending of 128 - 191
	1 Bit	I/O	CTRW-	DPT_Switch	0/1	[Ix][Switch/Sensor] Edge	Sending of 0 or 1
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Ix][Short Press] Constant Value (Integer)	0 - 255
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Ix][Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		CT---	DPT_Value_2_Ucount	0 - 65535	[Ix][Short Press] Constant Value	0 - 65535

	2 Bytes		CT---	9.xxx	-671088.64 - 670760.96	(Integer) [Ix][Short Press] Constant Value (Float)	Float Value
167, 173	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix][Short Press] Shutter Status (Input)	0% = Top; 100% = Bottom
	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix][Short Press] Dimming Status (Input)	0% - 100%
168, 174	1 Bit		CT---	DPT_Switch	0/1	[Ix][Long Press] 0	Sending of 0
	1 Bit		CT---	DPT_Switch	0/1	[Ix][Long Press] 1	Sending of 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix][Long Press] 0/1 Switching	Switching 0/1
	1 Bit		CT---	DPT_UpDown	0/1	[Ix][Long Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix][Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix][Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix][Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Bit		CT---	DPT_Step	0/1	[Ix][Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix][Long Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix][Long Press] Brighter	Long Pr. -> Brighter; Release -> Stop
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix][Long Press] Darker	Long Pr. -> Darker; Release -> Stop
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix][Long Press] Brighter/Darker	Long Pr. -> Brighter/Darker; Release -> Stop
	1 Bit		CT---	DPT_Switch	0/1	[Ix][Long Press] Light On	Sending of 1 (On)

	1 Bit		CT---	DPT_Switch	0/1	[Ix][Long Press] Light Off	Sending of 0 (Off)
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix][Long Press] Light On/Off	Switching 0/1
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix][Long Press] Run Scene	Sending of 0 - 63
	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix][Long Press] Save Scene	Sending of 128 - 191
	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix][Switch/Sensor] Alarm: Breakdown or Sabotage	1 = Alarm; 0 = No Alarm
	2 Bytes		CT---	9.xxx	-671088.64 - 670760.96	[Ix][Long Press] Constant Value (Float)	Float Value
	2 Bytes		CT---	DPT_Value_2_Ucount	0 - 65535	[Ix][Long Press] Constant Value (Integer)	0 - 65535
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Ix][Long Press] Constant Value (Percentage)	0% - 100%
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Ix][Long Press] Constant Value (Integer)	0 - 255
169, 175	1 Bit		CT---	DPT_Trigger	0/1	[Ix][Long Press/Release] Stop Shutter	Release -> Stop Shutter
170, 176	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix][Long Press] Dimming Status (Input)	0% - 100%
	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix][Long Press] Shutter Status (Input)	0% = Top; 100% = Bottom
177	1 Byte	I	C--W-	DPT_SceneNumber	0-63	[Motion Detector] Scene Input	Scene Value
178	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Motion Detector] Scene Output	Scene Value
179, 208	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix] Luminosity	0-100%
180, 209	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
181, 210	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
182, 211	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix] Presence State (Scaling)	0-100%
183, 212	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
184, 213	1 Bit	O	CTR--	DPT_Occupancy	0/1	[Ix] Presence State (Binary)	Binary Value
	1 Bit	O	CTR--	DPT_Ack	0/1	[Ix] Presence: Slave Output	1 = Motion Detected
185, 214	1 Bit	I	C--W-	DPT_Window_Door	0/1	[Ix] Presence Trigger	Binary Value to Trigger the Presence Detection
186, 215	1 Bit	I	C--W-	DPT_Ack	0/1	[Ix] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
187, 216	2 Bytes	I	C--W-	DPT_TimePeriodSec	0-65535	[Ix] Presence: Waiting Time	0-65535 s.
188, 217	2 Bytes	I	C--W-	DPT_TimePeriodSec	1-65535	[Ix] Presence: Listening Time	1-65535 s.
189, 218	1 Bit	I	C--W-	DPT_Enable	0/1	[Ix] Presence: Enable	According to parameters
190, 219	1 Bit	I	C--W-		0/1	[Ix] Presence: Day/Night	According to parameters
191, 220	1 Bit	O	CTR--	DPT_Occupancy	0/1	[Ix] Presence: Occupancy State	0 = Not Occupied; 1 = Occupied
192, 221	1 Bit	I	C--W-	DPT_Ack	0/1	[Ix] External Motion Detection	0 = Nothing; 1 = Motion detected by

							an external sensor
193, 198, 222, 227	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix][ax] Detection State (Scaling)	0-100%
194, 199, 223, 228	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix][ax] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
195, 200, 224, 229	1 Bit	O	CTR--	DPT_Switch	0/1	[Ix][ax] Detection State (Binary)	Binary Value
196, 201, 225, 230	1 Bit	I	C--W-	DPT_Enable	0/1	[Ix][ax] Enable Channel	According to parameters
197, 202, 226, 231	1 Bit	I	C--W-	DPT_Switch	0/1	[Ix][ax] Force State	0 = No Detection; 1 = Detection
203, 232	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix][Cx] Detection State (Scaling)	0-100%
204, 233	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix][Cx] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
205, 234	1 Bit	O	CTR--	DPT_Switch	0/1	[Ix][Cx] Detection State (Binary)	Binary Value
206, 235	1 Bit	I	C--W-	DPT_Enable	0/1	[Ix][Cx] Enable Channel	According to parameters
207, 236	1 Bit	I	C--W-	DPT_Switch	0/1	[Ix][Cx] Force State	0 = No Detection; 1 = Detection
237	1 Byte	I	C--W-	DPT_SceneControl	0-63; 128-191	[Thermostat] Scene Input	Scene Value
238	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Temperature Source 1	External Sensor Temperature
239	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Temperature Source 2	External Sensor Temperature
240	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Effective Temperature	Effective Control Temperature
241	1 Byte	I	C--W-	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode	1-Byte HVAC Mode
242	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Special Mode: Comfort	0 = Nothing; 1 = Trigger
	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] Special Mode: Comfort	0 = Off; 1 = On
243	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Special Mode: Standby	0 = Nothing; 1 = Trigger
	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] Special Mode: Standby	0 = Off; 1 = On
244	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Special Mode: Economy	0 = Nothing; 1 = Trigger
	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] Special Mode: Economy	0 = Off; 1 = On
245	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Special Mode: Protection	0 = Nothing; 1 = Trigger
	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] Special Mode: Protection	0 = Off; 1 = On
246	1 Bit	I	C--W-	DPT_Window_Door	0/1	[Tx] Window Status (Input)	0 = Closed; 1 = Open
247	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
248	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy	[Tx] Special Mode Status	1-Byte HVAC Mode

					4=Building Protection		
249	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Setpoint	Thermostat Setpoint Input
	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Basic Setpoint	Reference Setpoint
250	1 Bit	I	C--W-	DPT_Step	0/1	[Tx] Setpoint Step	0 = Decrease Setpoint; 1 = Increase Setpoint
251	2 Bytes	I	C--W-	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Setpoint Offset	Float Offset Value
252	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Setpoint Status	Current Setpoint
253	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Basic Setpoint Status	Current Basic Setpoint
254	2 Bytes	O	CTR--	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Setpoint Offset Status	Current Setpoint Offset
255	1 Bit	I	C--W-	DPT_Reset	0/1	[Tx] Setpoint Reset	Reset Setpoint to Default
	1 Bit	I	C--W-	DPT_Reset	0/1	[Tx] Offset Reset	Reset Offset
256	1 Bit	I	C--W-	DPT_Heat_Cool	0/1	[Tx] Mode	0 = Cool; 1 = Heat
257	1 Bit	O	CTR--	DPT_Heat_Cool	0/1	[Tx] Mode Status	0 = Cool; 1 = Heat
258	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] On/Off	0 = Off; 1 = On
259	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] On/Off Status	0 = Off; 1 = On
260	1 Bit	I/O	C-RW-	DPT_Switch	0/1	[Tx] Main System (Cool)	0 = System 1; 1 = System 2
261	1 Bit	I/O	C-RW-	DPT_Switch	0/1	[Tx] Main System (Heat)	0 = System 1; 1 = System 2
262	1 Bit	I	C--W-	DPT_Enable	0/1	[Tx] Enable/Disable Secondary System (Cool)	0 = Disable; 1 = Enable
263	1 Bit	I	C--W-	DPT_Enable	0/1	[Tx] Enable/Disable Secondary System (Heat)	0 = Disable; 1 = Enable
264, 270	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] [Sx] Control Variable (Cool)	PI Control (Continuous)
265, 271	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] [Sx] Control Variable (Heat)	PI Control (Continuous)
	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] [Sx] Control Variable	PI Control (Continuous)
266, 272	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] Control Variable (Cool)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] Control Variable (Cool)	PI Control (PWM)
267, 273	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] Control Variable (Heat)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] Control Variable (Heat)	PI Control (PWM)
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] Control Variable	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] Control Variable	PI Control (PWM)
268, 274	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] PI State (Cool)	0 = PI Signal 0%; 1 = PI Signal Greater than 0%
269, 275	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] PI State (Heat)	0 = PI Signal 0%; 1 = PI Signal Greater than 0%
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] [Sx] PI State	0 = PI Signal 0%; 1 = PI Signal Greater than 0%

Join and send us your inquiries
about Zennio devices:

<http://support.zennio.com>

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

Tel. +34 925 232 002.

www.zennio.com
info@zennio.com



RoHS