



## Z35

### KNX Capacitive Touch Panel

ZVI-Z35

Application Program Version: [1.1]  
User Manual Version: [1.1]\_a

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

# CONTENTS

---

Contents .....	2
Document updates.....	3
1 Introduction.....	4
1.1 Z35 .....	4
1.2 Functionality .....	6
1.3 Installation .....	7
2 Update Objects after Reset .....	8
3 Configuration.....	9
3.1 Main Configuration.....	9
3.1.1 GENERAL .....	9
3.1.2 Translations.....	13
3.1.3 Backlight.....	15
3.1.4 Security .....	17
3.1.5 Internal Temperature Sensor.....	20
3.1.6 Screensaver .....	20
3.1.7 Ambient Luminosity Sensor .....	22
3.1.8 Touch Locking .....	24
3.1.9 Sounds.....	25
3.1.10 ADVANCED .....	27
3.2 MENU.....	34
3.2.1 Configuration .....	36
3.3 Page <i>n</i> .....	37
3.3.1 Box <i>i</i> .....	38
3.4 Configuration Page .....	71
3.5 Inputs .....	74
3.5.1 Binary Input.....	74
3.5.2 Temperature Probe.....	74
3.5.3 Motion Detector .....	74
3.6 Thermostat .....	76
ANNEX I. Communication Objects.....	77

## DOCUMENT UPDATES

---

Version	Changes	Page(s)
[1.1]_a	<b>Changes in the application program:</b> <ul style="list-style-type: none"><li>• New parameter: Sound Type.</li></ul>	-
	New parameter: Sound Type.	25, 26
	Clarification about texts displayed in the screen.	13

# 1 INTRODUCTION

---

## 1.1 Z35

---

**Z35** is an easily and intuitively controllable high-performance **touch screens** from Zennio. The built-in features and functions make them the ideal solution for integral room control in hotels, offices or any other environments where controlling climate systems, lighting systems, shutters, scenes, etc. is required

The most outstanding features of Z35 are:

- **3.5 inch (3.5")** backlit **capacitive touch panel** with backlit 'Home' button incorporated.
- Multiple **direct action functions**, fully customisable.
- Control distribution across up to **7 customisable pages + 1 configuration page**.
- Full **climate** management.
- Programmable **timers**.
- **Scene** control.
- **Alarm** control.
- **Screensaver**.
- **Multi-Language**.
- 2 independent **thermostats**.
- Built-in **temperature** sensor.
- **Ambient luminosity sensor** for brightness automatic adjustment.
- **Proximity sensor** for quick start.

- **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)
- **Celsius and Fahrenheit** temperature scales for the on-screen indicators, being possible to select them in parameters or through communication object.
- 4 customisable **analogue-digital inputs**.
- **Heartbeat** or periodical “still-alive” notification.
- **Elegant** design, available in various **colours**.

## 1.2 FUNCTIONALITY

---

Application program feature the following functions:

- **7 General-Purpose Pages**, with up to 6 Fully-Customisable Boxes each, which the integrator may configure as indicators or controls.

- **Indicators:**

- Binary (icon, text).
- Enumerated (icon, text).
- Unsigned integer (1 / 2 bytes).
- Signed integer (1 / 2 / 4 bytes).
- Scaling (percentage).
- Temperature.
- Float (2 / 4 bytes).
- Text (14 bytes).

- **1-button Controls:**

- Switch (pre-set value, switch).
- Two objects (short press / long press)
- Hold & Release.
- Scene (run / save).
- Constant (counter, scaling, float).

- **2-button Controls:**

- Switch (icon, text).
- Switch + Indicator (counter, scaling, temperature)
- Two objects (short press / long press)
- Constant (counter, scaling, float).
- Enumeration.
- Shutters.
- Light dimming.
- Multimedia.

- **Climate Specific Controls:**

- Temperature Setpoint.
- Mode (cool/heat, extended).
- Special modes
- Fan.

- **Other Controls:**

- RGB.
- RGBW
- Daily / Weekly Timer.
- Alarm.

- **1 Configuration Page** (optional), which contains the brightness and sounds settings, the calibration of the built-in temperature probe, the programming button, Hour/Date settings and Reset configuration.

**Nota:** Z35 supports Latin, Chinese and Cyrillic characters for the on-screen, customisable labels.

## 1.3 INSTALLATION

Figure 1 shows the connection outline of the device:

1. Touch panel.
2. Backlit 'Home' button.
3. IR Receiver.
4. Luminosity and ProximitySensors.
5. Input connectors.
6. KNX connector.
7. Programation Button.
8. Programation LED.
9. Temperatura Probe.

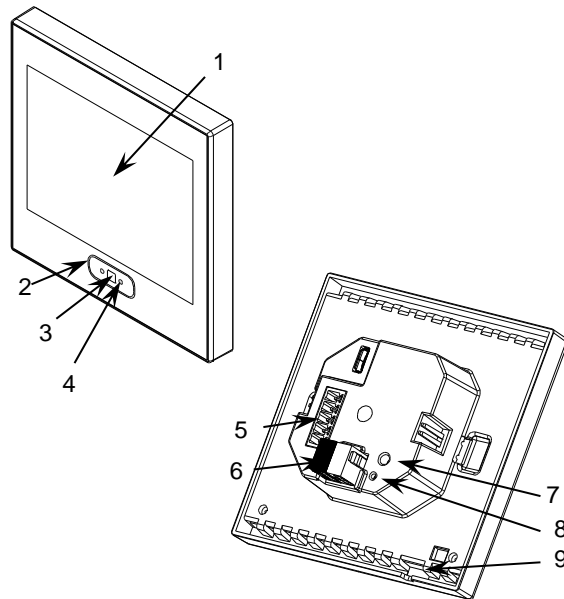


Figure 1. Schematic diagram.

Z35 is connected to the KNX bus through the built-in terminal (6). An external DC power supply is not needed.

A short press on the **Prog./Test button** (7) will make the device enter the programming mode. The **Prog./Test LED** (8) will then light in red. On the contrary, if this button is held while the device gets connected to the bus, Z35 will enter the **safe mode**. In such case, the programming LED will blink in red colour.

For detailed information about the technical features of Z35, as well as on security and installation procedures, please refer to the device **Datasheet**, bundled within the device packaging and also available at [www.zennio.com](http://www.zennio.com).

## 2 UPDATE OBJECTS AFTER RESET

---

The aim of this functionality is allowing the integrator to make a read request to the statuses of the device objects after a reset. There are two situations in which this functionality may be useful:

- In case of a Z35 reset, if there have been changes in the bus while the Z35 was off, after the reset, the objects in Z35 keep the same value as before the reset, but not their actual value in the installation.
- In addition, after programming from ETS, all objects are initialized to their default values, but not to their actual values in the installation.

When a bus failure or ETS programming occurs, read requests of all the following objects will be sent gradually (to prevent bus overload):.

- **General objects:** scene (reception), disabling pushbuttons, cleaning, external temperature, night mode, brightness, brightness sensor, proximity sensor, pulse sound enable and temperature scale.
- **Indicator objects.**
- **Timer** enabling objects.
- **Alarm confirmation** object.
- **Ventilation control:** Auto mode dedicated object.
- **14-byte Text indicator.**
- **RGB and RGBW** objects.

Objects that will NOT be updated are:

- **Control objects.**
- **Alarm trigger.**
- **4-Bit Light Dimming.**
- **4-Bit RGB/RGBW Dimming.**
- **Shutter: Stop/Step.**

The time and date objects of Z35 will be read from the bus always after a reset, regardless of whether this functionality is active or not.



## 3 CONFIGURATION

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

### 3.1 MAIN CONFIGURATION

This tab is divided into multiple screens, all of which contain a set of global parameters regarding the general functionality of the device, and therefore not specifically related to a particular page of the user interface.

#### 3.1.1 GENERAL

The "General" tab contains general settings. Most are checkboxes for enabling/disabling other functionalities.

#### ETS PARAMETERISATION

MAIN CONFIGURATION	INPUTS	<input type="checkbox"/>
	THERMOSTATS	<input type="checkbox"/>
<b>GENERAL</b>		
Translations	Heartbeat (Periodic Alive Notification)	<input type="checkbox"/>
Backlight	Show Time	<input checked="" type="checkbox"/>
Security	Show Temperature	Internal Temperature Probe
Internal Temp. Sensor	Button Box Style	Black Box
	First Weekday	<input checked="" type="radio"/> Monday <input type="radio"/> Sunday
	Weekdays Initials	MTWTFSS
	Hour Update Request Delay	<input type="checkbox"/>
	Update Objects	Disabled
	Screensaver	<input type="checkbox"/>
	Proximity Sensor Enabled	<input checked="" type="checkbox"/>
	Ambient Luminosity Sensor	<input type="checkbox"/>
	Touch Locking	<input type="checkbox"/>
	Sounds	<input checked="" type="radio"/> Default <input type="radio"/> Custom
	Advanced Configuration	<input type="checkbox"/>

Figure 2. GENERAL.

The following parameters are shown:

- **Inputs** [*disabled*<sup>1</sup>/*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. For more information, see section 3.5.
- **Thermostats** [*disabled*/*enabled*]: enables or disables the “Thermostat” tab in the tree on the left. For more information, see section 3.6.
- **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*).

The screenshot shows a configuration window for 'Heartbeat (Periodic Alive Notification)'. At the top, there is a checked checkbox. Below it, the 'Period' is set to '1' in a numeric input field. At the bottom, there is a dropdown menu currently showing 'min'.

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.*

- **Show Time** [*disabled*/*enabled*]: permits setting whether the current time (according to the internal clock) is displayed in the upper-left corner of any page or not.
- **Show Temperature** [*No/Internal Temperature Probe/External Value*]: sets whether the current temperature should show or not in the upper right corner of every page, being necessary in such case to choose the source of the temperature value: “Internal temperature probe” or “External value”. If the latter is chosen, an object named “**[General] External temperature**” will be enabled, so that it can be grouped with any other object that sends temperature values.
- **Button Box Style** [*No Box/White Box/Black Box*]: permits selecting a frame for the buttons, in order to distinguish them from the indicators.

<sup>1</sup> The default values of each parameter will be highlighted in blue in this document, as follows: [*default/rest of options*].



Figure 4. Button box (black, white, no box).

- **First Weekday** [[Monday/Sunday](#)]: sets the first day of week on calendar.
- **Weekdays initials** [[MTWTFSS](#)]: permits customising the label that will represent each of the weekdays on the screen. A seven-character string (including letters or numbers), ordered according to the above First Weekday parameter, must be entered – each of the characters will represent one weekday.
- **Hour Update Request Delay** [[disabled/enabled](#)]: sets a sending delay [[1...65535](#)] [[s/min/h](#)] for the date/time request when the device starts up.
- **Update Objects** [[Disabled/After Programming/After Reset/After Programming and Reset](#)]: enables the sending of read requests to update status objects and indicators (see section 2 for further details). There are four options available, some of them with a configurable **delay**:
  - “[Disabled](#)”: no read request, therefore objects are not updated.
  - “[After Programming](#)”: read requests are sent after a complete or partial download (or when pressing the reset button in the configuration page, if set as “Parameters Reset”, see section 3.4), after the parameterised **delay** (0-65535 s).
  - “[After Reset](#)”: read request are sent when a reset occurs (bus failure, the Reset Device ETS option or when pressing the reset button in the configuration page, if set as “Z35 Reboot”, see section 3.4), after the parameterised **delay** (0-65535 s).
  - “[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 3.1.6 for details.

- **Proximity Sensor** [*disabled/enabled*]: enables the proximity sensor. This functionality permits “waking up” the device display when detecting presence through the proximity sensor:
  - If the display is off or attenuated, it lights up with the corresponding luminosity of the current mode (normal mode or night mode).
  - If screensaver is active, it fades out and the default page is loaded.

If the proximity sensor is disabled, the display “wakes up” by touching it. When enabled, the object “[**General**] Proximity Sensor” is added to enable or disable this functionality at runtime.
- **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 3.1.7).
- **Touch Locking** [*disabled/enabled*]: enables or disables the possibility of locking the touch by object. When enabled, a new tab is added in the tree on the left (see section 3.1.8).
- **Sounds** [*Default/Custom*]: sets whether the sound functions (button beeps, alarm and doorbell) should work according to the pre-defined configuration (“Default”) or to a user-defined configuration (“Custom”) (see section 3.1.9).
- **Advanced Configuration** [*disabled/enabled*]: enables or disables the “Advanced” tab in the tree on the left (see 3.1.10).

The project topology shows the following objects by default:

- **[General] Time**: 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.

**Note:** *although the DPT of this object considers a field for setting the day of the week, Z35 calculates it from the date and therefore ignores that field.*

**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; see section 3.3.1.2.4).
- **[General] Display - Brightness:** 1-byte percentage object for changing the display brightness level.
- **[General] Illumination:** 1-bit object that, after the reception of the value “1” from the bus, will bring the backlight of the screen to the maximum level, while the reception of the value “0” will have no effect. This maximum illumination will last for a certain time, according to the explanation in section 3.1.3.
- **[General] Translations - Select Language:** 1 and 2-byte objects for changing the language showed in the screen when receiving a value through the bus (see section 3.1.2).
- **[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 3.1.2).
- **[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 3.1.5).

### 3.1.2 TRANSLATIONS

---

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

Texts of page titles, box titles, indicators, etc. 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 characters 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](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes) for a table with the language codes.

---

## ETS PARAMETERISATION

---

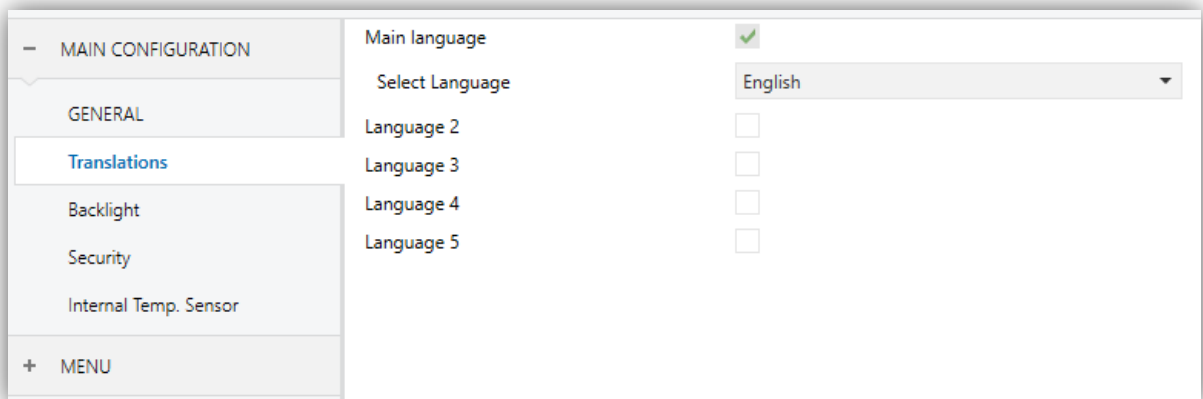


Figure 5. Translations.

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

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

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

These objects work accordingly to the behaviour mentioned above.

### 3.1.3 BACKLIGHT

---

It is possible to configure brightness user-defined levels for the display of Z35. Two operation modes are available: the **normal mode** and the **night mode**. The second one is optional. It is provided for temporary situations and environments where an excess of brightness may disturb the user. In such cases, it will be possible to switch the mode by means of a one-bit object and/or a scene object.

In order to prevent unnecessary power consumption while the device is idle, the backlight of the screen is partially faded out after a few minutes without any user interaction. After that, if the inactivity persists for a few more minutes, the backlight is completely turned off. While partially or totally faded out, any press on the screen will bring back the normal light level.

The timings for the above behaviour may differ in case a pop-up window is open, as well as depending on the configuration set in parameters.

---

#### ETS PARAMETERISATION

---

This screen entails certain aspects in relation to the backlight of the display:

- **Normal Mode** [[enabled](#)]: enabled by default.
  - **Backlight Level** [[1...100](#)]: display brightness percentage.

- **Backlight Attenuation** [[Turn Off the Screen After a Long Time of Inactivity/Attenuate the Screen After a Long Time of Inactivity/Max Illumination Always](#)]: sets the display behaviour after a no-activity time.

#### Notes:

- *This parameter is not available when the screensaver is enabling. However, it is possible to configure an attenuation when the screensaver activates (see section 3.1.6).*
- *Maintaining the default option is encouraged. Permanent illumination may affect adversely the device lifetime, and is only intended for special circumstances.*

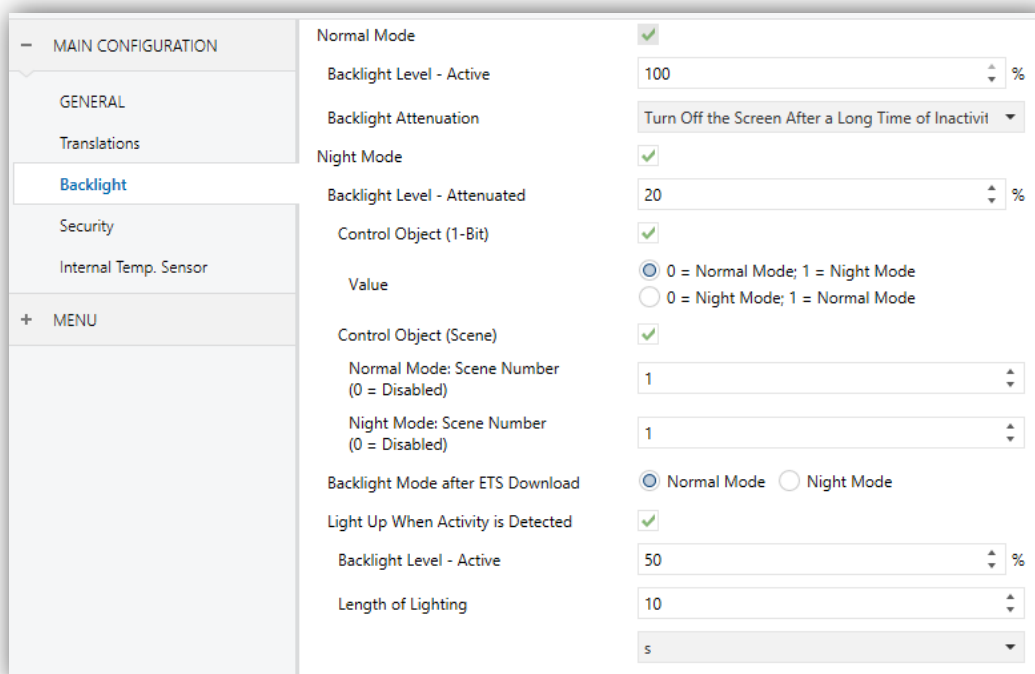


Figure 6. Illumination.

- **Night Mode** [[disabled/enabled](#)]: in case of being this mode necessary, this checkbox needs to be marked, which will incorporate some new parameters:
  - **Backlight Level - Attenuated** [[0...20...100](#)]: display brightness percentage.
  - **Control Object (1-Bit)** [[disabled/enabled](#)]: when marked, it will be possible to switch the mode by writing to a binary object (“**[General] Illumination**”). It is possible to select which value should trigger which mode [[0 = Normal mode, 1 = Night mode/0 = Night mode, 1 = Normal mode](#)]



- **Control Object (Scene)** [*disabled/enabled*]: when marked, it will be possible to switch the mode by writing a certain scene value to “[**General Scene: Receive**”]. Two specific textboxes will show up to enter what scenes [*0/1...64*] will trigger each mode.
- **Backlight Mode after ETS download** [*Normal Mode/Night Mode*]: sets which of the two modes will be active after an ETS Download.
- **Light Up when Activity is Detected** [*disabled/enabled*]: allows changing temporarily the brightness level when the screen is touched or presence is detected in night mode. To this end, two parameters:
  - **Backlight Level – Active** [*1...50...100*]: display brightness percentage.
  - **Length of Lighting** [*1...10...65535*] [*s/min/h*]: when the time set in this parameter elapses, Z35 will switch to Night Mode (provided that it was not already in this mode).

### 3.1.4 SECURITY

Any box or page will be given the possibility of **restricted access by password**. Setting one or two different passwords is possible, so the integrator can afterwards configure whether the access to a page or box will be protected by one password or another, or remain unprotected – every page can be independently configured.

Buttons that lead to a protected page or box will show a little **lock** icon overlaid on their lower left corner.

Figure 7 shows the “enter password” dialog shown to the user when trying to access a protected page.



Figure 7. Security Pop up.

In case of setting up **two levels**, the first one is assumed to be *enclosed* by the second one. This means that whenever the device asks the user to type password #1 (to enter a certain page); password #2 will also be accepted. On the contrary, password #1 cannot be used instead of password #2. This behaviour permits, therefore, making password #2 available to users with further privileges while password #1 is assigned to users with fewer privileges.

Moreover, when accessing to a protected page, all the boxes and pages with the same or lower access level of the introduced password, are automatically unlocked. It can be set if the elements are relocked after a time period or a page switch.

## ETS PARAMETERISATION

This screen permits selecting how many security levels (one or two) will be available for the configuration of the access to the control pages or the boxes.

Figure 8. Security.

- **Security Levels** [[One Level/Two Levels](#)]: dropdown list for selecting whether one (default option) or two security levels will be available.

**Note:** with independence of the option selected here, it will be necessary to establish the security level desired for each specific page of controls.

- **Protect Again** [[After a Time Period/After a Page Switch/After a Time Period or a Page Switch](#)]: sets when is re-activated the security of pages or boxes unlocked. When selecting the first or the last option, a new parameter **Time** [[10...65535](#)][s] [[1...65535](#)][min/h] appears to set the time period.
- **Password** [Level 1: [1234](#); Level 2: [5678](#)]: parameter made of four additional textboxes, each of which should contain one of the four consecutive digits [[0...9](#)] that will compose the password.

In case of enabling two security levels, the Password parameter will show twice, being the first one referred to the password of Level 1, and the second one to the password of Level 2.

LEVEL 1 PASSWORD	
- Default Password (1)	1
- Default Password (2)	2
- Default Password (3)	3
- Default Password (4)	4
LEVEL 2 PASSWORD	
- Default Password (1)	5
- Default Password (2)	6
- Default Password (3)	7
- Default Password (4)	8

Figure 9. Two security levels.

**Important:** the password insertion dialog features a specific option (lower left button) that lets the user change, in runtime, the passwords originally set by parameter. After accessing this option and prior to typing the new password, the user will be required to type the corresponding old password (level 1 or level 2). Note that although it will be possible to type password 2 even if the device asks for password 1, the new password typed afterwards will be anyway stored as the new password for level 1.

- **Security Pad Labels:** parameter consisting in six additional textboxes, intended for the customisation of the messages that the device shows (or may show) when the user interacts with the password insertion dialog.
  - **Label for 'Enter Password 1'** [[Enter Password 1](#)]: message shown when the user is required to type in the password for level 1.

- **Label for ‘Enter Password 2’** [[Enter Password 2](#)]: message shown when the user is required to type in the password for level 2.
- **Label for ‘ERROR’** [[ERROR](#)]: message shown to the user when the typed password is not valid.
- **Label for ‘New Password’** [[New Password](#)]: message shown to ask the user for a new password, during the password change process.
- **Label for ‘Repeat Password’** [[Repeat Password](#)]: message shown when the user is required to re-type the new password.
- **Label for ‘Updated’** [[Updated](#)]: message shown to the user as a confirmation of the password change.

### 3.1.5 INTERNAL TEMPERATURE SENSOR

---

Z35 is equipped with an internal temperature sensor for monitoring the ambient temperature of the room, so that the device can report it to the KNX bus and trigger several actions when the temperature reaches certain values.

Please refer to the specific documentation of the “**Temperature Prove**” available at the Zennio homepage, [www.zennio.com](http://www.zennio.com), for detailed information about the functionality and the configuration of the related parameters.

### 3.1.6 SCREENSAVER

---

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

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



Figure 10. Screensaver Pop up.

In addition, it is possible to configure if the screen will **attenuate** the brightness level while the screensaver is active or if the user wants the illumination to remain at the same level (maximum).

The time for the screen attenuation once the screensaver is activated will be 1 minute. In case of the 'night mode' is enabled, the illumination of the screen will depend on the mode in which the screen is located.

The screensaver will disappear when touching the screen screen or the HOME button or, if the proximity sensor is activated, when it detects presence.

### Notes:

- When showing the welcome greeting (see section 3.1.10.2), screensaver will not become active.
- When screensaver will be active and a pop-up will be activated (see touch locking 3.1.8, cleaning function 3.1.10.1 or welcome greeting 3.1.10.2) this pop-up will become show up over screensaver.

## ETS PARAMETERISATION

After enabling “Screensaver” from “General” screen (see section 3.1.1), a new tab will be incorporated into the tree on the left.

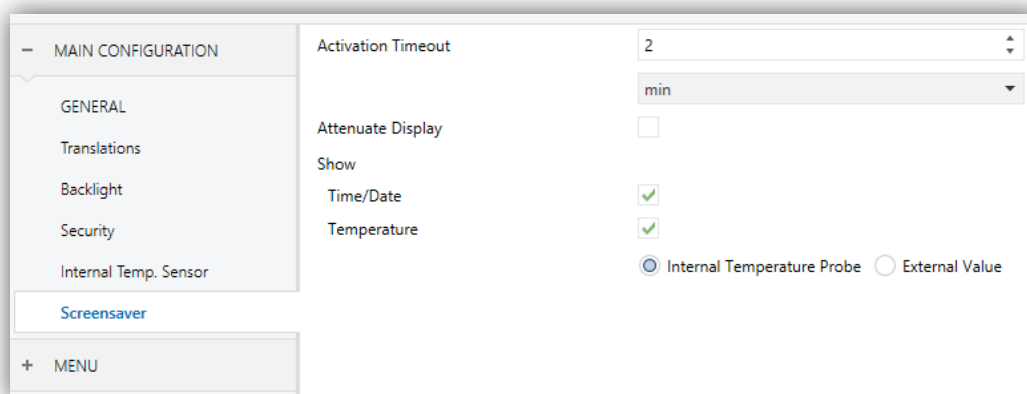


Figure 11. Screensaver.

- **Activation Timeout** [30...65535][s] [1...2...65535][min/h]: time to automatically show the screensaver after the last button touch.
- **Attenuate Display** [disabled/enabled]: sets whether to decrease the display brightness when showing the screensaver.
- **Time/Date** [disabled/enabled]: sets whether to show the current time or not.

- **Temperature** [*disabled/enabled*]: sets whether to show the current temperature or not. When enabled, it is possible to select the source of the temperature value [*Internal Temperature Probe/External Value*]. The latter option will enable a new two-byte object “[**General**] **External Temperature**” through which the device can receive the required values from the bus.

### 3.1.7 AMBIENT LUMINOSITY SENSOR

Z35 includes a sensor to measure the ambient luminosity level, so that the brightness of the display can be adjusted according to the current luminosity of the room.

To this end, it is possible to set a luminosity threshold and send a **binary object** or a **scene object** when the luminosity value is higher or lower than the threshold. In this way, if this object is linked with the one to control the backlight mode (see section 3.1.3), the normal mode could be activated if the brightness exceeds the threshold and the night mode if the brightness is below the threshold (plus the hysteresis).

#### Ejemplo:

1) ‘Backlight’ is parameterized as follow:

- Control Object (1-Bit)

Normal Mode = “0”

Night Mode = “1”

- Control Object (Scene)

Normal Mode = “1”

Night Mode = “64”

2) ‘Ambient Luminosity Sensor’ is parameterized as follow:

- Threshold: Ambient Luminosity Level = 25%

- Threshold: Hysteresis = 10%

- Control Object (1-Bit)

Normal Mode = “0”

Night Mode = “1”

- Control Object (Scene)

Normal Mode = “1”

Night Mode = “64”

Associating [**General**] Luminosity Object (1-bit) with [**General**] Backlight Mode:

- Luminosity > 25% → Normal Mode
- 25% > Luminosity > 15% → No mode change
- Luminosity < 15% → Night Mode

## ETS PARAMETERISATION

After enabling **Ambient Luminosity Sensor** from “General” screen (see section 3.1.1), a new tab will be incorporated into the tree on the left.

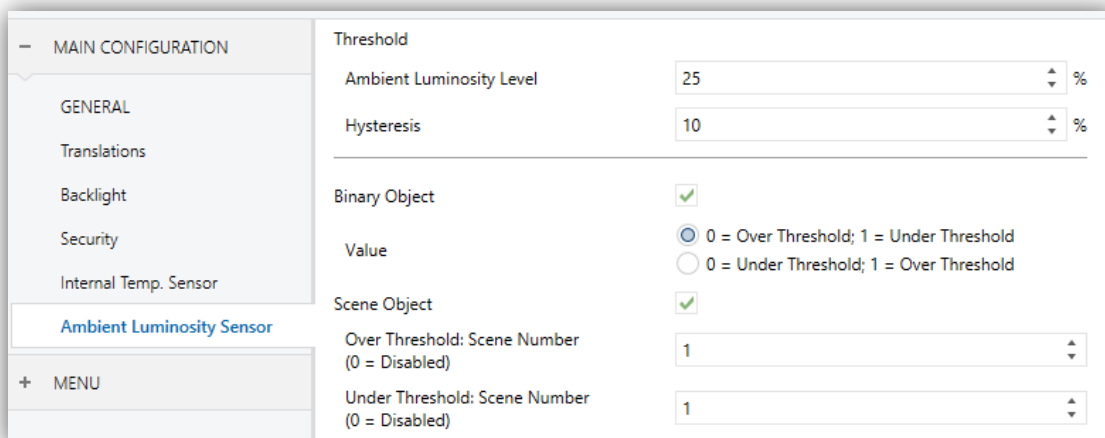


Figure 12. Ambient Luminosity Sensor.

- **Threshold:** configuration of the threshold of the luminosity sensor.
  - **Ambient Luminosity Level** [1...25...100]: luminosity porcentaje of the threshold value.
  - **Hysteresis** [0...10...100]: luminosity porcentaje for the hysteresis, i.e., margin around the threshold value.
- **Binary object** [disabled/enabled]: enables the binary object “[General] Luminosity Object (1-bit)” that will be sent to the bus with the corresponding value when the luminosity is over or under the threshold.
  - **Value** [0 = Over Threshold, 1 = Under Threshold/0 = Under Threshold, 1 = Over Threshold]: sets which value is sent when the luminosity is over or under the threshold.
- **Scene object** [disabled/enabled]: when enabled a scene value will be sent through the object “[General] Scene: send”, when the luminosity is over or under the threshold.
  - **Over Threshold: Scene Number (0 = Disabled)** [0/1...64]: scene number that is sent when a luminosity level higher than the threshold is reached.
  - **Under Threshold: Scene Number (0 = Disabled)** [0/1...64]: scene number that is sent when a luminosity level lower than the threshold is reached. Hysteresis must be considered.

### 3.1.8 TOUCH LOCKING

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

While locked, user presses on the touch buttons will be ignored: no actions will be performed 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.

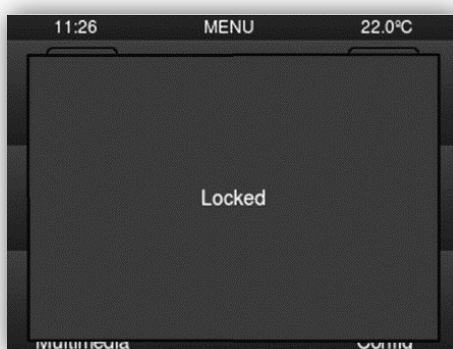


Figure 13. Touch Locking message.

**Note:** If an alarm with active lock is activated, it will be disabled and you can press the screen normally. If you want to re-activate it, you will need to lock it again.

### ETS PARAMETERISATION

After enabling **Touch Locking** from “General” screen (see section 3.1.1), a new tab will be incorporated into the tree on the left.

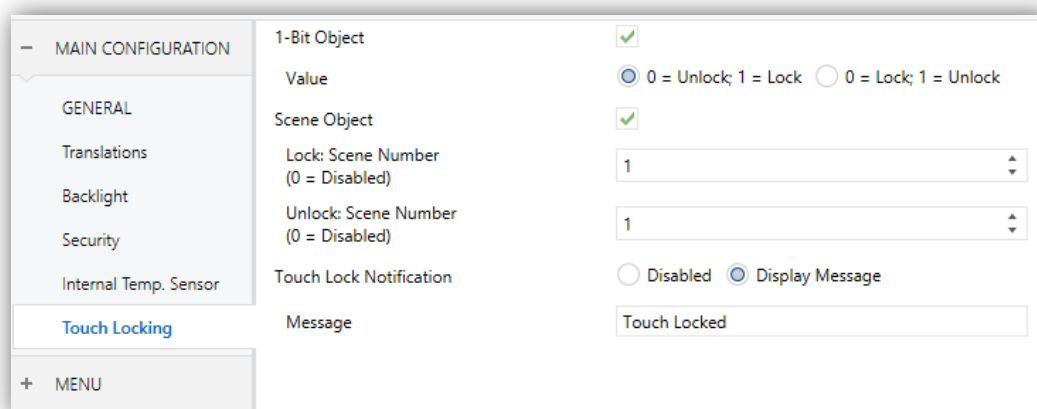


Figure 14. Touch Locking.

In this tab you can configure the blocking of the presses on the display.



- **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.
  
- **Push Button 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.

### 3.1.9 SOUNDS

---

Z35 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 these actions are performing 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.

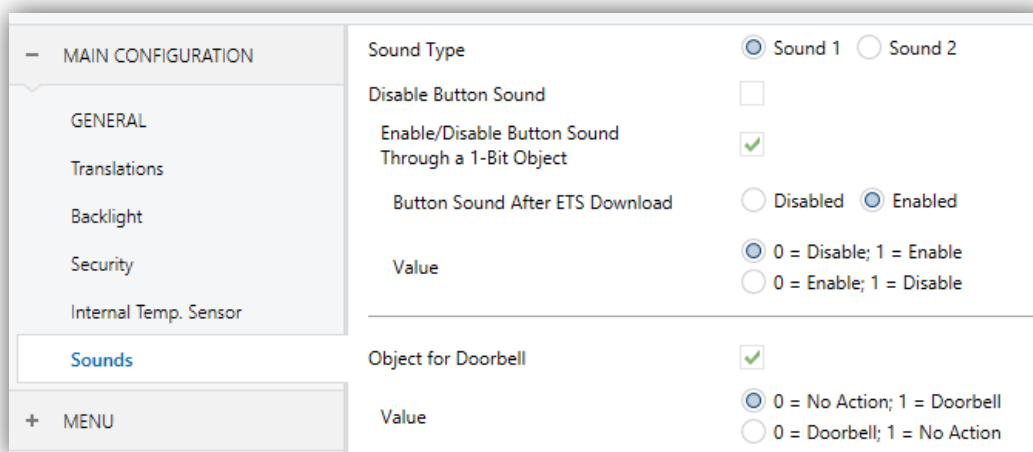
Pressing and sending confirmation sounds can be silenced using one of the following methods:

- **Parameterisation after ETS download.**
- **1-bit communication object.**
- **Checkbox in "Configuration Page".**

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

## ETS PARAMETERISATION

After enabling the “Custom” configuration of **Sounds** from “General” screen (see section 3.1.1), a new tab will be incorporated into the tree on the left.



**Figure 15.** Sounds.

The initial configuration of this screen is equivalent to the default option. However, the following parameters will be configurable.

- **Sound Type** [[Sound 1/Sound 2](#)]: sets which sounds range incorporates the device.
- **Disable Button Sounds** [[disabled/enabled](#)]: allows the user to ignore the push and to confirm sounds.

- **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

### 3.1.10 ADVANCED

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

After enabling the **Advanced configuration** from “General” screen (see section 3.1.1), a new tab will be incorporated into the tree on the left.

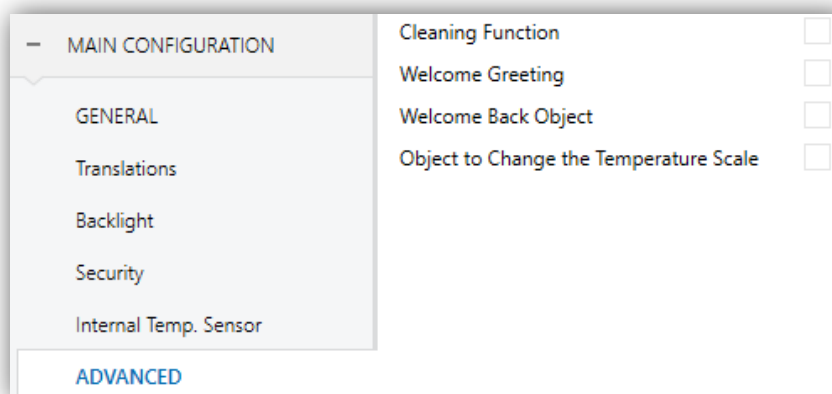


Figure 16. Advanced.

- **Cleaning Function** [[disabled/enabled](#)]: enables or disables the “Cleaning Function” tab. See section 3.1.10.1 for details.
- **Welcome Greeting** [[disabled/enabled](#)]: enables or disables the “Welcome Greeting” tab. See section 3.1.10.2 for details.
- **Welcome Back Object** [[disabled/enabled](#)]: enables or disables the “Welcome Back Object” tab. See section 3.1.10.3 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** temperatura.
- Indicators of **temperature controls** linked to a box in the display ([Climate] Temperature setpoint)
- **Temperature indicators** ([Climate] Temperature).

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

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

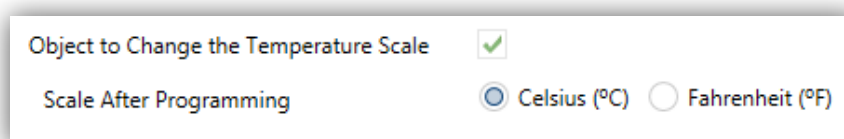


Figure 17. Temperature Scale After Programming.

### 3.1.10.1 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).



Figure 18. Cleaning Function Pop up.

## ETS PARAMETERISATION

After enabling **Cleaning Function** from “Advanced” screen (see section 3.1.10), a new tab will be incorporated into the tree on the left.

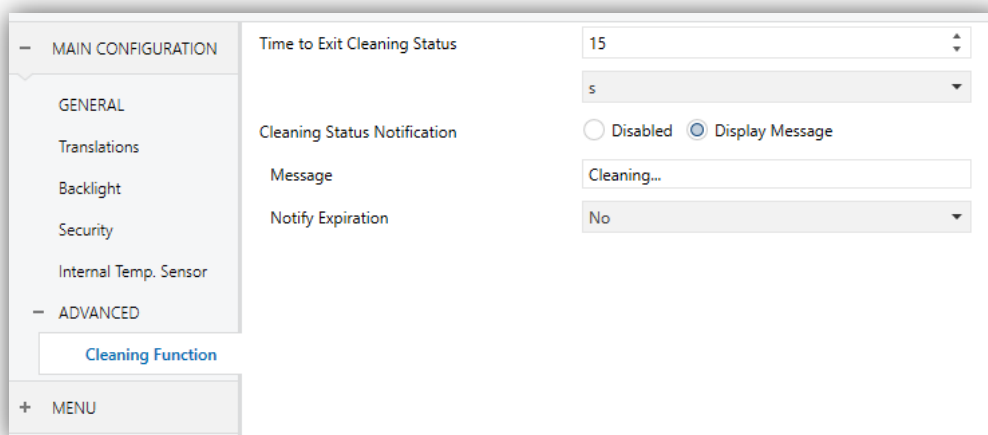


Figure 19. Cleaning Function.

- **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:
  - **Lenght 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.

### 3.1.10.2 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.



**Figure 20.** Welcome Greeting Pop up.

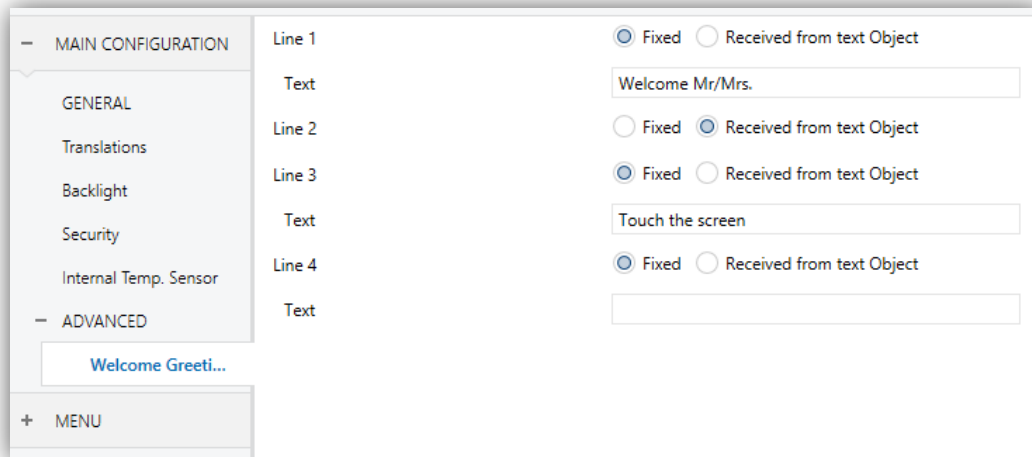
When a “1” is received through the welcome greeting one-bit object, the display will show the welcome text. The same will happen if any of the 14-byte objects that define the text lines receives a new value from the bus.

The message disappears when touching the screen.

**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** from “Advanced” screen (see section 3.1.10), a new tab will be incorporated into the tree on the left.



**Figure 21.** Welcome Greeting.

- **Line [1,4] [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.

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

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

### 3.1.10.3 WELCOME BACK

Z35 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 the welcome back object to be sent, the normal action of that button will not be triggered.

## ETS PARAMETERISATION

After enabling **Welcome Back Object** from “Advanced” screen (see section 3.1.10), a new tab will be incorporated into the tree on the left.

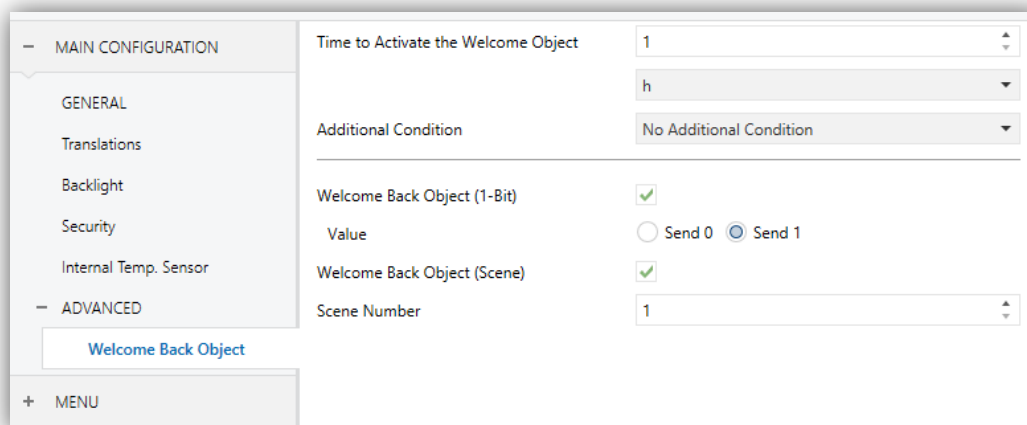


Figure 22. Welcome Back Object.

- **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.
- **Additional Condition** [\[No Additional Condition/Do Not Send Unless All Additional Conditions are 0/Do Not Send Unless All Additional Conditions are 1/Do Not Send at Less One of the Additional Conditions is 0/Do Not Send at Less One of the Additional Conditions is 1\]](#): condition that must be fulfilled for sending the welcome object. When selecting any condition, the following parameter appears:
  - **Number of Condition Objects** [\[1...5\]](#): up to 5 objects can be selected for the 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](#)]

## 3.2 MENU

The user interface is organised into **pages** (up to **seven** different pages, in addition to the “Configuration Page”), each of which can be accessed from the menu page, which (unless the contrary has been parameterised) is automatically shown after the start-up.

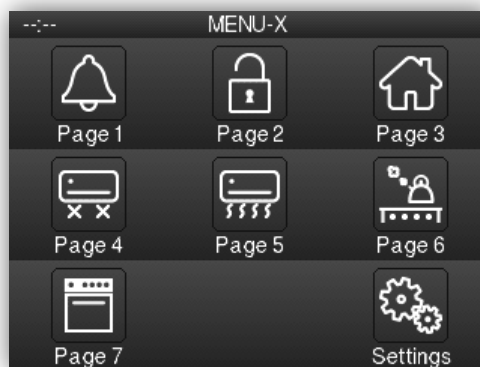


Figure 23. Menu

Password-protected pages (see section 3.1.4) will display a small icon with a lock next to the lower left corner of the associated button. On the other hand, if a page contains a box with an active alarm (see section 3.3.1.5.5), a small alarm icon will be displayed next to the lower right corner.



Figure 24. Menú with protection and alarms icons.

All the seven pages are **general-purpose**, at the entire disposal of the integrator, who may include up to six controls/indicators (with different functionalities) per page, being even possible to combine alarm, climate or any other controls within the same page.



Figure 25. Control page.

The remaining configuration Page is **specific-purpose**, as it is provided for user customisation of the device.

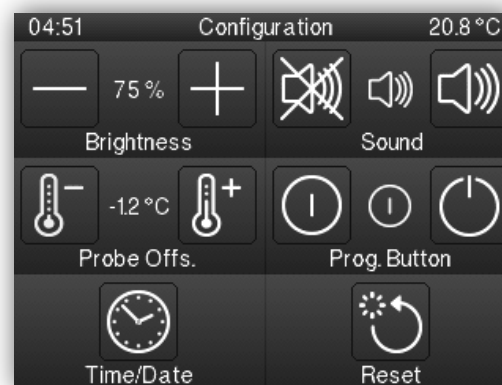


Figure 26. Configuration page.

On the other hand, the user interface will always show on top the title of the current page.

In addition, there is a button below the display, at the bottom of the touch, which allows the user to return to the home page. This button is known as the **'Home'** button.

### 3.2.1 CONFIGURATION

The Menu tab contains only one screen, Configuration.

#### ETS PARAMETERISATION

Title	
Main Language	Menu
Default Page	Menu
Page 1	<input type="checkbox"/>
Page 2	<input type="checkbox"/>
Page 3	<input type="checkbox"/>
Page 4	<input type="checkbox"/>
Page 5	<input type="checkbox"/>
Page 6	<input type="checkbox"/>
Page 7	<input type="checkbox"/>
Configuration Page	<input type="checkbox"/>

Figure 27. MENU Configuration.

The parameters available are:

- **Title:**
  - **Language X [Menu]:** text field that defines the title that will be shown on the top of the Menu page for the corresponding language.
- **Default Page [Menu]:** dropdown list that sets the page (Menu, or any of the general-purpose pages) that will behave as the default page. This page will be the one shown after one minute of inactivity, assuming that such page has been enabled and it is not protected with password.

In addition, one checkbox is shown per general-purpose page (that is, pages 1 to 7), as well as one more checkbox for the Configuration page. Each of these checkboxes will allow enabling or disabling the corresponding page in the device – a specific ETS tab will appear upon the activation of a page.

### 3.3 PAGE $n$

When any of the general-purpose pages is enabled from the “Configuration” screen under the “Menu” tab, a new tab named **Page  $n$**  will appear, where  $n$  is the number of the page.

Under this tab, one screen (**Configuration**) will be initially displayed to let the integrator enable or disable each of the boxes in the page. Depending on that, more parameter screens will appear.

**Note:** figures in this section will show the parameters of a certain page (for instance, page 1) or a certain box (for instance, box 1). The parameters for other pages or boxes are totally analogous.

#### ETS PARAMETERISATION

**Figure 28.** Page  $n$  Configuration.

This screen contains the following parameters:

- **Title:**
  - **Language X [Page 1]:** text field that defines the title that will be shown on the top of the Menu page for the corresponding language.
- **Icon [Home]:** sets the icon that will represent the page in the Menu page.

- **Protect:** sets whether the page will be password-protected or not. Depending on the security levels configured (one or two; see section 3.1.4) this list will contain the following options:

➤ **One Level:**

- **[No]:** the page will not be protected by password. All users can access it.
- **[Yes]:** the page will be protected by password. Users will be asked to type the password when trying to access it.

➤ **Two Levels:**

- **[No]:** the page will not be protected by password. All users can access it.
- **[Level 1]:** the page will implement security level 1. To access it, users will be required to enter password 1 or password 2.
- **[Level 2]:** the page will implement security level 2. To access it, users will be required to enter password 2.

Finally, six checkboxes are provided to let the integrator enable or disable each of the six boxes in the page. When enabled, every box will have its own parameter screen (**Box *i***) under the corresponding “Page *n*” tab. The next section explains the parameterisation process of these boxes.

### 3.3.1 BOX *i*

This screen contains the following parameters common to all type of boxes:

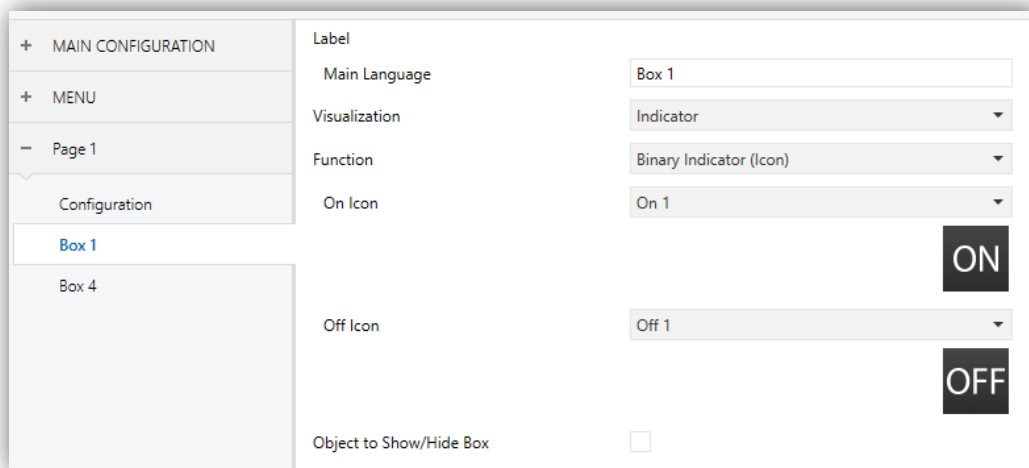


Figure 29. Box *i* Configuration.

- **Label:**
  - **Language X [Box 1]:** identifying title for the box for the corresponding language.
  
- **Visualization:** box format. The available box formats in Z35 are:
  - **[Indicator]:** the box will work as a status indicator.
  - **[1-button Control]:** the box will work as a one-button control.
  - **[2-button Control]:** the box will work not only as a status indicator, but also as a two-button control.
  - **[Climate Control]:** the box will act as a climate indicator and a climate control.
  - **[Other]:** the box will implement some other special functionality.
  
- **Function:** depending on the “Visualization” type selected, the parameters below will change. The following sections explain the available parameters depending on the visualization type selected.

The parameters described below are common to all types of controls:

- **Icon or Button:** drop-down list with the available icons to show in the indicator(s) and/or button(s) of the box.
  
- **Object to Show/Hide Box [disabled/enabled]:** enables or disables a 1-bit object (“[Pn][Bi] Show/Hide Box”) to show or hide the corresponding box.

Moreover, it is possible to protect with password boxes that are not indicators:

- **Protect:** sets whether the box will be password-protected or not. This function works in the same way as Page *n* security:
  - **One Level:**
    - **[No]:** the box will not be protected by password. All users can access it.
    - **[Yes]:** the box will be protected by password. Users will be asked to type the password when trying to access it.
  - **Two Levels:**

- [\[No\]](#): the box will not be protected by password. All users can access it.
- [\[Level 1\]](#): the box will implement security level 1. To access it, users will be required to enter password 1 or password 2.
- [\[Level 2\]](#): the box will implement security level 2. To access it, users will be required to enter password 2.

### 3.3.1.1 INDICATORS

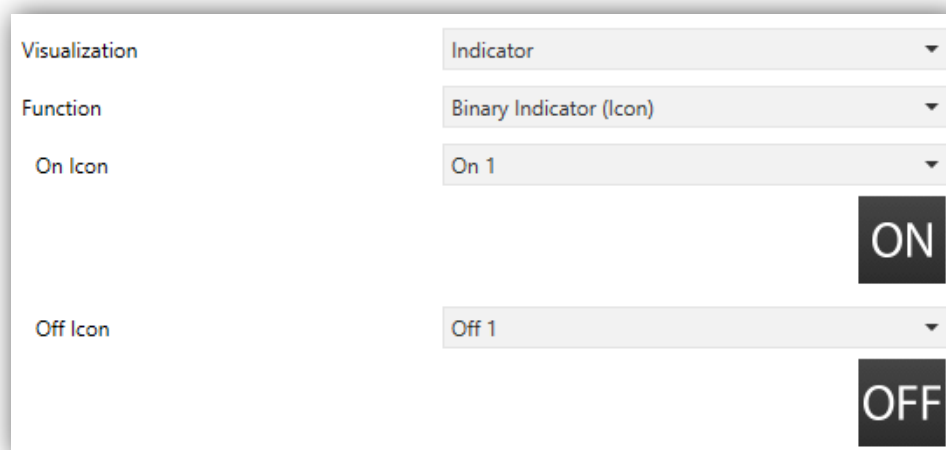
Boxes designed for displaying statuses, showing a numeric or text value, or displaying an icon representing the current value of a communication object.

The functions and the related parameters available in Z35 are:

#### 3.3.1.1.1 Binary Indicator (Icon)

The box will behave as a binary state indicator. Each of the two states will be shown in the box through the selected icon.

When this function is assigned to the box, the “[Pn][Bi] Binary indicator” communication object become available, as well as the parameters that permit selecting the icon to be displayed when the object acquires the value “0” (**Icon Off**) and the icon to be displayed when it acquires the value “1” (**Icon On**) will be displayed.



**Figure 30.** Binary Indicator (Icon).

Therefore, when the device receives the values “0” or “1” through the aforementioned object, the box will show one icon or another.



### 3.3.1.1.2 Binary Indicator (Text)

The box will behave as a binary state indicator. Each of the two states will be shown in the box through a different label.

When the box is assigned this function, the “[Pn][Bi] Binary indicator” object will become available, as well as the parameters that permit typing the text to be shown when the object receives a “0” (**Text Off**) and that to be shown when it becomes “1” (**Text On**) will be displayed.

Visualization	Indicator
Function	Binary Indicator (Text)
On Text	
Off Text	

Figure 31. Binary Indicator (Text).

Therefore, when the device receives the values “0” or “1” through the aforementioned object, the box will show one text or another.

### 3.3.1.1.3 Enumerated Indicator (Icon)

The box will behave analogously to the case of the Binary indicator, (Icon) however, it will be possible to distinguish up to 6 states (configurable through the **# Enums** parameter) instead of only two. The states will be determined by the reception of any values between 0 and 255.

Visualization	Indicator
Function	Enumerated Indicator (Icon)
# Enums	2
Value 1	0
Icon 1	Off 1
Value 2	0
Icon 2	Off 1

Figure 32. Enumerated Indicator (Icon).

When this function is assigned to the box, a 1-byte communication object, “[Pn][Bi] **Enumerated Indicator**”, will become available as well as two more parameters (**Value** and **Icon**) for each of the states to be distinguished will be available too. This allows setting which icon will be shown in the box upon the reception of which value through the communication object.

**Example:** a three-state enumerated indicator (“3” under “# Enum”) is parameterised as follows:

Value 1 = “1”	Icon 1 = “One”
Value 2 = “3”	Icon 2 = “Two”
Value 3 = “5”	Icon 3 = “Three”

When the [Pn][Bi] **Enumerated Indicator** object receives the value “1”, the box will show icon “One”. When it receives the value “3”, icon “Two” will be shown. And when it receives the value “5”, icon “Three” will show. In case of receiving any other value, no icon will be shown.

### 3.3.1.1.4 Enumerated Indicator (Text)

The box will behave analogously to the case of the Binary indicator (Text) however it will be possible to distinguish up to 6 states (configurable through the # **Enums** parameter) instead of only two. The states will be determined by the reception of any values between 0 and 255.

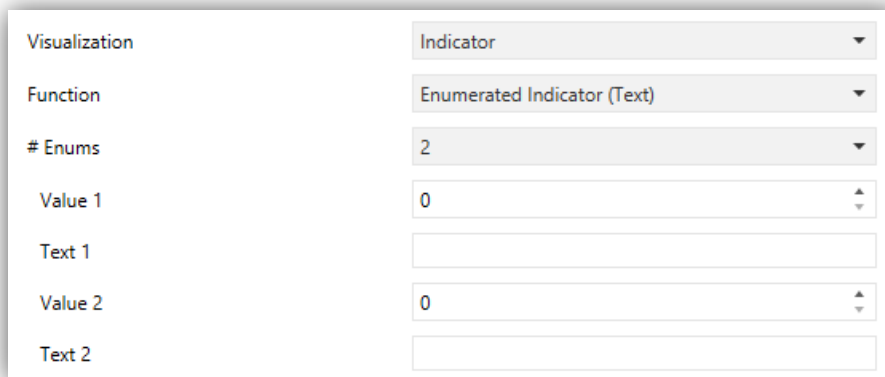


Figure 33. Enumerated Indicator (Text).

When this function is assigned to the box, one 1-byte communication object, “[Pn][Bi] **Enumerated Indicator**”, will become available as well as two additional parameters (**Value** and **Text**) will be displayed for each of the states to be distinguished. This

allows setting which texts will be displayed in the box upon the reception of which values through the communication object.

### 3.3.1.1.5 Numerical Indicators

The box will behave as a numerical state indicator that displays the value of the communication object enabled when the function is assigned to the box.

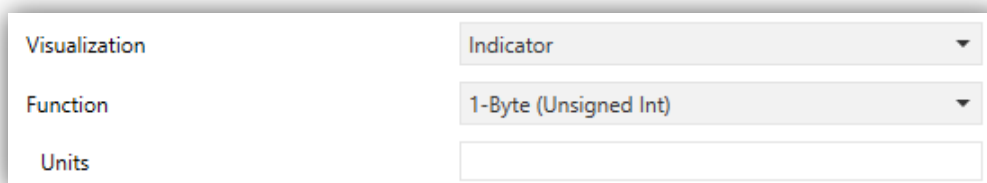


Figure 34. Numerical Indicator.

The range of values allowed for each type and the name of the corresponding object are included in the following table.

Function	Range	Related Object
1-Byte (Unsigned Int)	0– 255	[Pn][Bi] 1-byte Unsigned Int Indicator
1-Byte (Signed Int)	-128 – 127	[Pn][Bi] 1-byte Signed Int Indicator
Percentage Indicator	0 – 100	[Pn][Bi] Percentage Indicator
Temperature Indicator	-99 – 199	[Pn][Bi] Temperature Indicator
2-Byte (Unsigned Int)	0 – 65535	[Pn][Bi] 2-byte Unsigned Int Indicator
2 -Byte (Signed Int)	-32768 – 32767	[Pn][Bi] 2-byte Signed Int Indicator
2 -Byte (Float)	-671088,64 – 670760,96	[Pn][Bi] 2-byte Float Indicator
4-Byte (Signed Int)	-2147483648 – 2147483647	[Pn][Bi] 4-byte Signed Int Indicator
4-Byte (Float)	-2147483648 – 2147483647	[Pn][Bi] 4-byte Float Indicator

Table 1 Numerical Indicators.

In all cases the integrator will be shown a text field (**Unit**), empty by default that permits specifying the measuring units of the displayed value.

For percentage indicators, the symbol % always will be displayed as unit. Temperature indicators will be displayed in °C or °F depending on the selected scale (by the object to change scale, see section 3.1.10).

### 3.3.1.1.6 14-byte Text Indicator

The box will show the text received through the communication object “[Pn][Bi] 14-Byte Text Indicator”.

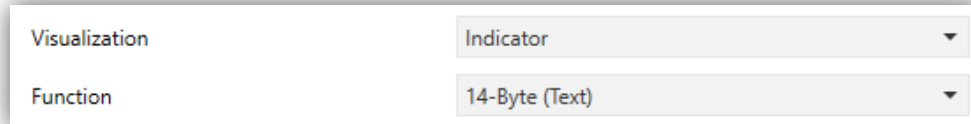


Figure 35. 14-byte text indicator.

**Note:** objects associated to the text indicators are stored in salved zone, so its value will be maintained after a restart.

### 3.3.1.2 1-BUTTON CONTROL

Boxes configured as 1-button controls show one centred button and a title. There is a parameter (**Function**) that will select the specific function that the box will play.

#### 3.3.1.2.1 Switch

The central button of the box will react to user presses by sending a binary value to the bus through the “[Pn][Bi] Binary control” object, which turns visible as soon as this function is assigned to the box.

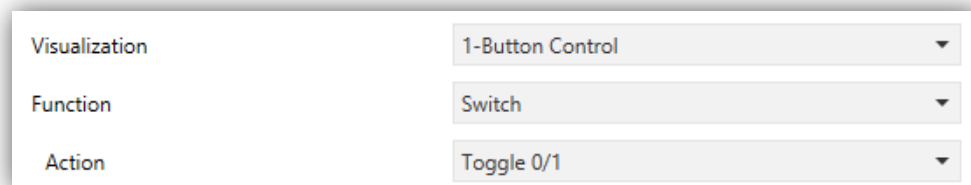


Figure 36. 1-Button Control: Switch.

On the other hand, **Action** permits setting what value will be sent to the bus through the mentioned object, and on what events. The options are:

- [\[Send 0\]](#): one “0” will be sent whenever the button is pressed.
- [\[Send 1\]](#): one “1” will be sent whenever the button is pressed.
- [\[Toggle 0/1\]](#): alternate sending of the values “1” and “0”.

### 3.3.1.2.2 Two Objects (Short Press/Long Press)

The central button in the box will react differently to a short press and to a long press, setting a time threshold to distinguish both types of press by parameter. The control responds to these presses by sending a binary value to the bus.

Different objects are used to send values for short and long pulsations: "[Pn][Bi] **Two objects - Short press**" and "[Pn][Bi] **Two objects - Long press**".

The screenshot shows a configuration window for a '1-Button Control'. The 'Visualization' is set to '1-Button Control'. The 'Function' is set to 'Two Objects (Short Press/Long Press)'. The 'Action on Short Press' is set to 'Send 0'. The 'Action on Long Press' is set to 'Send 1'. The 'Long Press Threshold Time' is set to 6, with a multiplier of 'x 1 ds'.

Visualization	1-Button Control
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

**Figure 37.** 1-Button Control: Two Objects (Short Press/Long Press).

On the other hand, the parameter called **Action** permits setting what value will be sent to the bus through the mentioned objects, and on what events. The actions available for each type of press are:

- [\[Send 0\]](#): a "0" will be sent whenever the button is pressed.
- [\[Send 1\]](#): a "1" will be sent whenever the button is pressed.
- [\[Toggle 0/1\]](#): alternate sending of the values "1" and "0".
- [\[Send 1-Bit Unsigned Int Value\]](#): the 1-Byte Unsigned Integer value indicated in **Value** will be sent whenever the button is pressed.

To distinguish optimally a short press and a long press, Z35 includes a parameter called **Long Press Threshold Time** [\[4...6...50\]\[ds\]](#), that sets the minimum time the user should hold the button in order to consider it a long press.

### 3.3.1.2.3 Hold & Release

This control function allows the user to configure sending a binary value on pressing and a different binary value on releasing the button, through the 1-bit object "[Pn][Bi] **Hold & Release**".

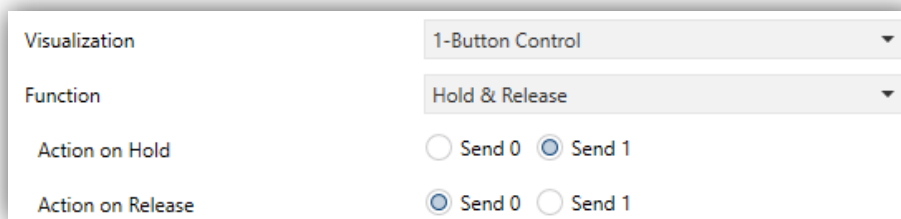


Figure 38. 1-Button Control: Hold & Release.

Through the parameters **Action on Hold** [[Send 0/Send 1](#)] and **Action on Release** [[Send 0/Send 1](#)] permit setting what value will be sent to the bus for each case.

#### 3.3.1.2.4 Scene

The central button of the box will react to the different pulses by sending a scene value to the KNX bus through the “[**General**] Scenes: send” object or through an individual scene object “[**Pn**][**Bi**] Scene: Send” for this box depending on the option selected in the parameter **Object to Use** [[General Scene Object/Individual Box Scene Object](#)].

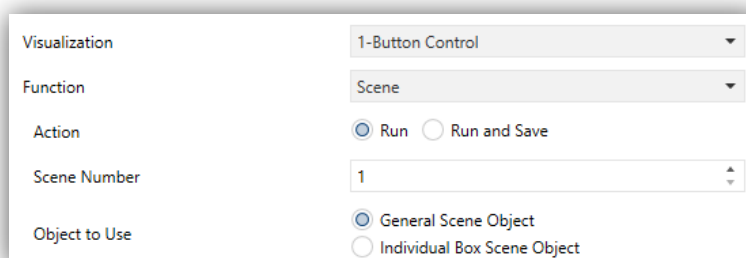


Figura 39. 1-Button Control: Scene.

Parameter **Scene number** [[1...64](#)] permits specifying the number of the scene to be sent. Moreover, through the parameter called **Action** [[Run Scene/Run and Save](#)], sets whether the device will only send scene execution orders (after a short press) or if it will be possible, in addition to sending execution orders upon short presses, to send scene save orders in the case of a long press.

#### 3.3.1.2.5 Numerical Constant Controls

Whether the box is assigned any of the remaining “constant” control options, the central button in the box will react to user presses by sending a certain numerical value, which is required to be specified under **Constant value**. This numerical value will depend on the constant control type selected to the box (**Function**).

**Figure 40.** Numerical Constant Control.

Table 2 shows, for every available Function, the permitted value range and the name of the object through which the values are sent to the bus.

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

**Table 2.** Numerical Constant Control.

### 3.3.1.3 2-BUTTON CONTROL

Boxes configured as 2-button controls consist in an indicator and two buttons that, when touched, trigger the sending of an action to the KNX bus through a certain object.

As a general rule, most of the 2-button controls permit configuring a pair of parameters, **Left button** and **Right button**, each containing a dropdown list for the selection of the icons to be displayed inside the buttons in the box.

**Note:** *When multiple presses are made consecutively on the buttons of a control that regulates (e.g. increases / decreases) the value of a certain variable, only the final value selected by the user will be sent to the bus, to prevent an unnecessary bus traffic due to all the intermediate values.*

On the other hand, the **Function** parameter contains a dropdown list for the selection of the particular two-button control type to be assigned to the box. The available options (and their related parameters) are:

### 3.3.1.3.1 Switch

When the user presses any of the buttons, Z35 will send a parameterized binary value to the bus through the object "[Pn][Bi] Switch".

Figure 41. 2-Button Control: Switch.

The parameters available are:

- **Action** [Left = 0; Right = 1/Right = 0; Left = 1]: sets the value to be sent when pressing each of the two buttons.
- **Indicator Type** [Text/Icon]: sets whether the indicator of the control will be a text indicator (two text fields will be displayed to introduce the corresponding texts for “0” and “1”) or an icon indicator (two drops lists will be displayed to select the corresponding icons for “0” and “1”).

### 3.3.1.3.2 Switch + Indicator

Like the previous control, when the user presses any of the buttons, Z35 will send a parameterized binary value to the bus through the object "[Pn][Bi] Switch". However, the indicator is independent; it will be updated according to the value received by the dedicated object.

Figure 42. 2-Button Control: Switch + Indicator.



- **Action** [Left = 0; Right = 1/Right = 0; Left = 1]: sets the value to be sent when pressing each of the two buttons.
- **Indicator Type** [Counter/Scaling/Temperature]: sets the indicator type. According to the selected indicator the objects “[Pn][Bi] x-Byte Signed Int Indicator”, “[Pn][Bi] Percentage Indicator”, “[Pn][Bi] Temperature Indicator” will be enabled respectively.

When selecting “Counter” type, the following parameters appear:

- **Size** [1 Byte/2 Bytes/4 Bytes Signed Int]: size of the indicator object.
- **Sign** [Signed/Unsigned]: sign of the indicator object.
- **Unit**: text field to set the measurement unit displayed next to the indicator.

### 3.3.1.3.3 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 buttons (i.e., they will work as a joint control; for independent buttons, please configure them as 1-button controls). Different objects will be used for the short and long presses: “[Pn][Bi] Two objects - Short press” and “[Pn][Bi] Two objects - Long press”.

Visualization	2-Button Control
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
Indicator Type	<input checked="" type="radio"/> Text <input type="radio"/> Icon
Text for '0'	<input type="text"/>
Text for '1'	<input type="text"/>

Figure 43. 2-Button Control: Two Objects (Short Press/Long Press).

- **Action on Short / Long press** [Left = 0; Right = 1/Right = 0; Left = 1]: sets the value to be sent when short / long pressing each of the two buttons.
- **Long Press Threshold Time** [4...6...50], sets the minimum time the user should hold the button in order to consider it a long press.

- **Indicator Type** [[Text/Icon](#)]: sets whether the indicator of the control will be a text indicator (two text fields will be displayed to introduce the corresponding texts for “0” and “1”) or an icon indicator (two drops lists will be displayed to select the corresponding icons for “0” and “1”).

#### 3.3.1.3.4 Numerical Controls (Counter, Scaling and Float)

If the box is assigned any of the numerical functions “[Counter](#)”, “[Scaling](#)” or “[Float](#)”, user touches over the buttons will trigger the sending of a certain numerical value to the bus. This value will be progressively increased or decreased with every touch on one button or the other, through the control object, while the box itself will permanently reflect the current value of the status object

Visualization	2-Button Control
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

**Figure 44.** 2-Button Numeric Control

The parameters available are:

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

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	<u>Signed</u>	<a href="#">[-128...127]</a>	<a href="#">[-128...127]</a>	<a href="#">[-128...1...127]</a>	<a href="#">[-128...10...127]</a>	[Pn][Bi] 1-Byte Signed Int Value
		<u>Unsigned</u>	<a href="#">[0...255]</a>	<a href="#">[0...255]</a>	<a href="#">[0...255]</a>	<a href="#">[0...255]</a>	[Pn][Bi] 1-Byte Unsigned Int Value
	2-Byte	<u>Signed</u>	<a href="#">[-32768...32767]</a>	<a href="#">[-32768...32767]</a>	<a href="#">[-32768...32767]</a>	<a href="#">[-32768...32767]</a>	[Pn][Bi] 2-Byte Signed Int Value
		<u>Unsigned</u>	<a href="#">[0...65535]</a>	<a href="#">[0...65535]</a>	<a href="#">[0...65535]</a>	<a href="#">[0...65535]</a>	[Pn][Bi] 2-Byte Unsigned Int Value
Scaling	1-Byte		<a href="#">[0... 100]</a>	<a href="#">[0... 100]</a>	<a href="#">[0... 1...100]</a>	<a href="#">[0... 10...100]</a>	[Pn][Bi] 1-Byte Scaling Value
Float	2-Byte		<a href="#">[-671088,64...670760,96]</a>	<a href="#">[-671088,64...670760,96]</a>	<a href="#">[-671088,64...0,5...670760,96]</a>	<a href="#">[-671088,64...1...670760,96]</a>	[Pn][Bi] 2-Byte Float Value

Tabla 3. 2-buttons Numeric Control.

### 3.3.1.3.5 Enumeration

The box will behave analogously to the case of the switch control, with the peculiarity that size of the communication object will be 1-Byte (“**[Pn][Bi] Enumerated Control**”). Moreover, it will be possible to distinguish up to 6 states (configurable through the **# Enums** parameter).

Visualization	2-Button Control
Function	Enumeration
# Enums	2
Indicator Type	<input checked="" type="radio"/> Text <input type="radio"/> Icon
Value 1	0
Text 1	
Value 2	0
Text 2	

Figure 45. 2-Button Control: Enumeration.

The parameters available are:

- **# Enums** [1...6]: number of states that will be distinguished. For every distinguished state, the parameter **Value** will become available together with the corresponding indicator (either **Text** or **Icon**).
- **Indicator Type** [Text/Icon]: sets whether the indicator of the control will be a text or an icon. If a **text indicator** is selected, a textbox will be added for each state. In the caso of an **icon indicator**, a drop list will be added for each state to set the corresponding icon.

### 3.3.1.3.6 Shutter

Shutter control permits sending move up, move down (“**[Pn][Bi] Move Shutter**”) or stop (“**[Pn][Bi] Stop Shutter**”) to control a shutter actuator connected to the bus by pressing the buttons in the box. In addition, the box will contain an indicator that will permanently show, as a percentage, the value of the status object (“**[Pn][Bi] Shutter Position**”).

Visualization	2-Button Control
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

Figure 46. 2-Button Control: 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.
- **Type parameter:**
  - [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). Once the button is released, it will send an order to stop the shutter.

### 3.3.1.3.7 Dimmer

The light control function permits making use of the two buttons in the box to send orders to a light dimmer, either through a binary object or through a 4-bit object. Moreover, the box will permanently display the current value of the dimming status object, which needs to be linked to the analogous object from the dimmer (as it does not get automatically updated on button presses).

Visualization	2-Button Control
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%

Figure 47. 2-Button Control: 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. Options are “Left Off, Right On” (default) and “Left On, Right Off”.

- **Dimming Step:** 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.

Setting up a box as a dimming control brings up the objects **[Pn][Bi] Light Indicator** (1 byte), **[Pn][Bi] Light On/Off** (1 bit) and **[Pn][Bi] Light Dimming** (4 bits). The first one acts as the dimming status object, which is required to be linked to the analogous object from the dimmer and whose value, in percentage, will be permanently reflected in the box. On their side, control orders will be sent through one of the two remaining objects:

- After a short press on the “turn on” button, the value “1” will be sent through the “**[Pn][Bi] Light On/Off**” binary object, while a short press on the “turn off” button will trigger the sending of the value “0”.
- When a long press on the “turn on” / “turn off” button takes place, the object “**[Pn][Bi] Light Dimming**” will send the light increment / decrement order corresponding to the parameterised dimming step, while a stop order (value “8” / value “0”) will be sent through the same object as soon as the user releases the button.

**Note:** *most light dimmers implement light step dimming progressively (i.e., sending a dimmer a step order of 25% typically does not imply that the light level is suddenly incremented/decremented by 25%, but a progressive increment or decrement of the light level by 25% which is in fact interrupted if a stop order arrives (such order is sent by Z35 when the user releases the button). Due to this behaviour, it is advised to parameterise dimming steps of 100%, so that the user can perform a complete dimming (from totally off to totally on, or vice versa) or a partial dimming by simply holding the button and then releasing it as soon as he gets the desired light level, therefore with no need of performing successive long presses for regulations greater than the parameterised step.*

#### 3.3.1.3.8 Multimedia

---

Tapping on one of the two buttons will make Z35 send a binary value to the bus, while pressing on the other will make it send the inverse binary value.

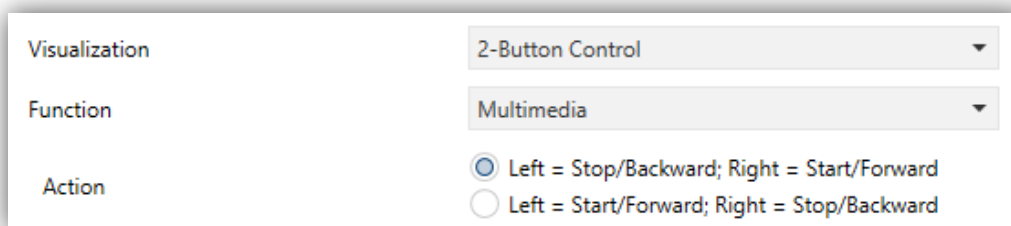


Figure 48. 2-Button Control: Multimedia.

- **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 object “[Pn] [X] 1-bit Multimedia Control”, while the indicator values should be received through the object “[Pn] [X] Multimedia Text Indicator”.

#### 3.3.1.4 CLIMATE CONTROL

This category covers a set of functions related to the climate control. The available options for **Function** (and for the dependent parameters) are detailed in the following sections.

##### 3.3.1.4.1 [Climate] Temperature Setpoint

This function permits **controlling the temperature setpoint of an external thermostat** by means of a two-button box and of the parameters enabled to that effect. When any box is configured as this control type, a single object will appear: “[Pn][Bi] (Climate) Temperature Setpoint”.

In addition, the box itself will permanently reflect the value in °C (o °F) of the status object, whose value 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.

Therefore, after every press on the temperature increment button, the bus will be sent (through the control object) a certain value, progressively increased on every button press until the parameterisable maximum setpoint value has been reached. Analogously, after every press on the temperature decrement button, the bus will be sent a progressively decreased value until the parameterisable minimum setpoint value has been reached.

Visualization	Climate Control
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

**Figure 49.** 2-Button Control: [Climate] Temperature Setpoint.

The available parameters are:

- **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.
- **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.

#### 3.3.1.4.2 [Climate] Mode

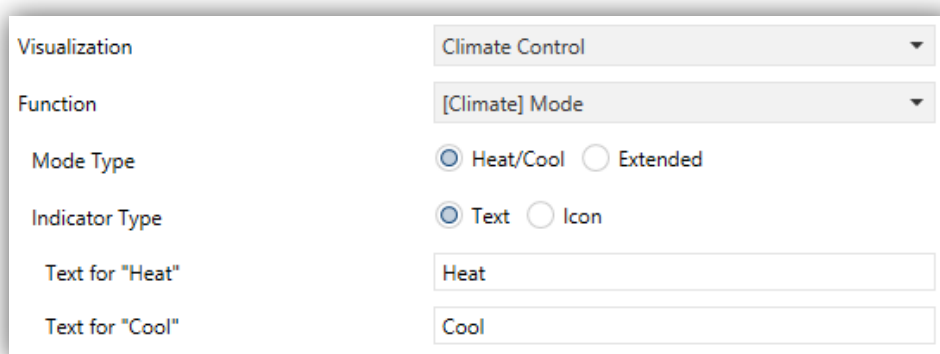
This function turns the box into a climate mode control.

Two mode control types are possible: **Heat/Cool** control type and **Extended (HVAC)** control type. Configuring any of them is possible through the **Mode Type** [Heat/Cool/Extended] parameter.





- **[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. A centred indicator will reflect, as an icon, the currently active mode.

One communication object is also enabled: the “[Pn][Bi] (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 (see Table 4). Also, the box indicator will automatically alternate between one icon and another when a mode change is performed.



**Figure 50.** 2-Button Control: [Climate] Mode (Heat/Cool).

Mode	Icon	Sent Value
Cool		0
Heat		1

**Table 4.** Heat/Cool Mode vs. Icons vs. Object Value.

- **[Extended]:** Selecting this type of mode control turns the box into a 2-button control, which permit sequentially commuting among the different HVAC climate modes. A centred indicator will reflect, as an icon, the currently active 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.

When this type of mode control is assigned to the box, one 1-byte communication object is enabled: the “[Pn][Bi] (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 (see Table 5), after which the box will display the icon that corresponds to the new mode. Additionally, if the status object receives from the bus a value that represents any of the modes, the box will adopt the corresponding icon, while if an unrecognised value is received, no icon will be shown.

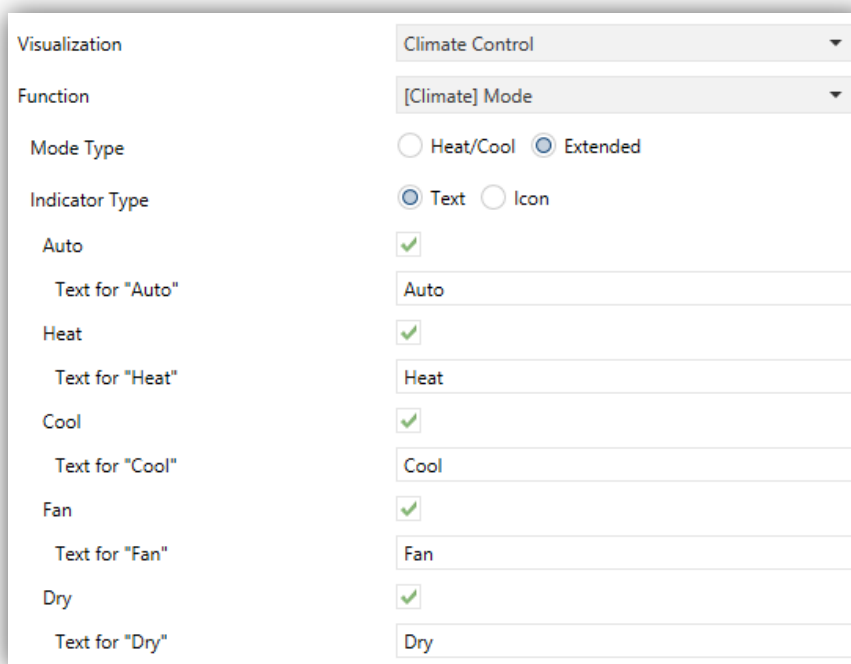


Figure 51. 2-Button Control: (Climate) Mode - Extended.






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

Table 5. HVAC Mode vs. Icon vs. Object Value.

In addition, it is possible to set the type of indicator, using the **Indicator Type** parameter [[Text/Icon](#)]: this setting allows configuring a text or icon indicator.

### 3.3.1.4.3 [Climate] Fan

This function implements a 2-button (increase/decrease) fan control, as well as an icon indicator.

When this function is assigned to the box, control object “[Pn][Bi] Fan Control” as well as the “[Pn][Bi] Fan Indicator” 1-byte status object are enabled. The status object (which needs to be linked to the status object of the fan actuator) will express, as a percentage, the value of the current fan level, which will be represented with a variable icon on the box.

Visualization	Climate Control
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"/>

Figure 52. 2-Button Control: [Climate] Fan.

- **Action** [Left = Decrease; Right = Increase/Left = Increase; Right = Decrease]: permits setting the increase or decrease actions to one button or another.
- **Speed Levels** [1...5]: sets how many speed levels will be available in the control.
- **Control Type**: sets the type of the communication objects that will control the fan level.
  - [1 bit (decrease/increase)]: orders of speed increase/decrease are sent through the one-bit object “[Pn] [Bi] Fan Control (Type: 1-Bit)”.
  - [Scaling]: scaling values are sent through the one-byte object “[Pn] [Bi] Fan Control (Type: Scaling)”.
  - [Enumeration]: integer values are sent through the one-byte object “[Pn] [Bi] Fan Control (Type: Enumeration)”.

- **Cyclical** [[disabled/enabled](#)]: sets whether scrolling through the speed levels is circular or not. If checked, an increase order in the maximum level switches to the minimum and vice versa (in the minimum level a decrease order switches to the maximum).
- **Auto Mode** [[disabled/enabled](#)]: sets whether the fan Auto mode will be available or not. If checked, the following parameters are also shown.
  - **Dedicated Object for Auto Mode:** sets how the Auto mode should be activated. In the absence of a dedicated object, the Auto fan mode will be activated by switching to fan level 0.

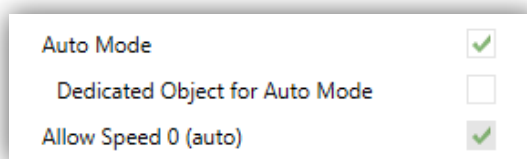
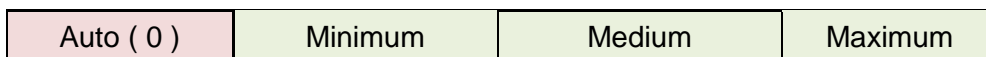


Figure 53. [Climate] Fan – Auto mode.

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



On the other hand, marking the checkbox enables the 1-bit object “[Pn][Bi] Fan Control – Auto Mode”, which will trigger the **Auto mode** when it receives the corresponding value (**Value to Set Auto Mode** [[Send 0 / Send 1](#)]).

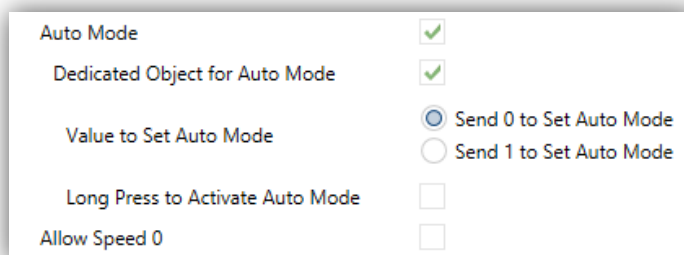


Figure 54. [Climate] Fan – Dedicated Object for Auto mode.

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
-------	---------	--------	---------	------

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

A further long press deactivates back the Auto mode and sends the minimum 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
-------	---------	--------	---------

- **Allow speed 0** [*disabled/enabled*]: 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.

### 3.3.1.4.4 [Climate] Special Mode

Boxes configured as **Special Mode** controls include two buttons that let the user sequentially commute between the different special climate modes, as well as an icon indicator of the the currently active special mode.

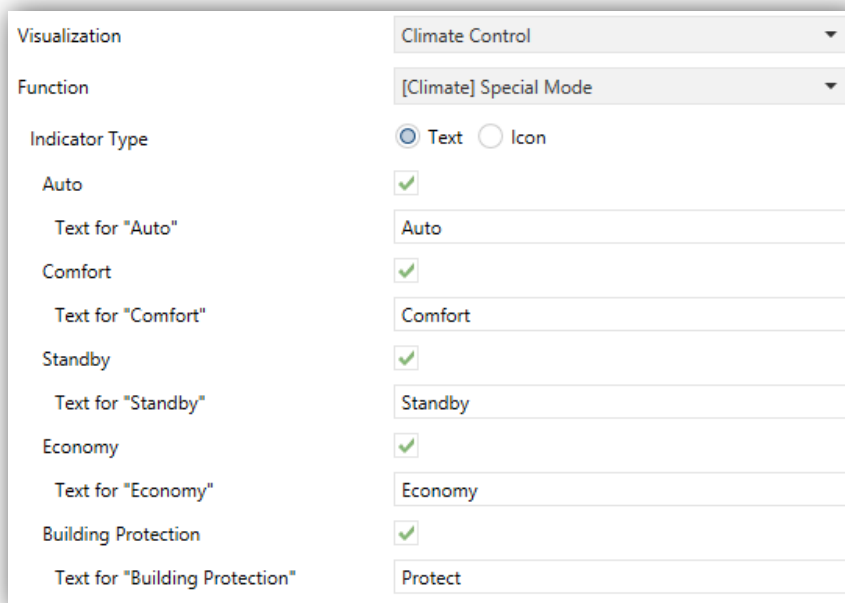







Figure 55. Climate Control: [Climate] Special Mode.

When this function is assigned to the box, the 1-Byte object “[Pn][Bi] (Climate) **Special Mode Control**” will be enabled. Through the former object will be sending the bus the value that corresponds to the mode that the user selects by touching the buttons, and the corresponding icon/text will be shown in the display.

If this object receives an unrecognised value from the bus, no icon will be displayed in the indicator.

Special Mode	Icon	Sent Value
Comfort		1 (0x001)
Standby		2 (0x002)
Economy		3 (0x003)
Protection		4 (0x004)
Auto Mode		5 (0x005)

**Tabla 6.** Special Modes vs. Icon vs. Object Value.

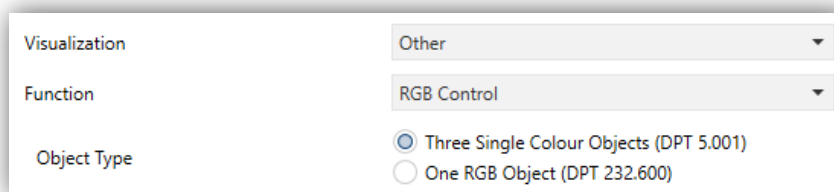
In addition, it is possible to set the type of indicator, using the **Indicator Type** parameter [[Text/Icon](#)], for each mode.

### 3.3.1.5 OTHER CONTROL TYPES

This category involves the following functions, which can be selected through the **Function** parameter:

#### 3.3.1.5.1 RGB Control

This function is intended for sending orders to three-colour LED light regulators, such as Lumento X3 from Zennio.



**Figure 56.** RGB Control.

When the function is assigned to the box, the following parameters come up:

- **Object Type** [[Three Single Colour Objects \(DPT 5.001\)](#)/[One RGB Object \(DPT 232.600\)](#)]: permits selecting what type of object will be used for controlling the light level of the RGB channels. If the first option is selected, three 1-byte objects will be enabled (with their Write and Read flags activated, so they can both send regulation orders and receive the status from the regulator) named: “[Pn][Bi] Red Channel”, “[Pn][Bi] Green Channel” and “[Pn][Bi] Blue Channel”. Control orders will consist in sending the light level (as a percentage) of the different channels through the corresponding objects.

If the second option is selected, only one 3-byte object will be enabled: “[Pn][Bi] RGB Colour”. In this case, the light levels of the three channels are sent (and received) concatenated into the above 3-byte object.

**Note:** *if statuses are received while the luminaire performs a regulation, this control may become hardly serviceable until such regulation ends.*

Regarding the RGB control boxes themselves, a central label will permanently show, as a percentage, the current light level (which is determined by the channel with the highest current light level). This **indicator** gets updated automatically as the user interacts with the box, but is also conditioned by the values received from the bus through the already described objects.



**Figure 57.** RGB Control (Box).

Moreover, two buttons can be found in the box:

- **Short-pressing** on the left button it permits commuting between a total absence of light in the three channels, and their respective states previous to the switch-off.
- **Long-pressing** it, on the other hand, will make the device send a 4-bit dimming order through object “[Pn][Bi] Light Dimming”, analogously as the light dimming control.
- The right button, when pressed, launches a pop-up containing a 5x6 **colour palette** that permits the selection of a light level for every channel by touching on any of the pre-set colours.



Figura 58. RGB/RGBW Colour Palette

### 3.3.1.5.2 RGBW Control

The RGBW control is **entirely analogous** to the above RGB control, although it also lets controlling a specific fourth channel for white (“[Pn][Bi] White Channel”), in case such feature is supported by the dimmer, as in Lumento X4 from Zennio.

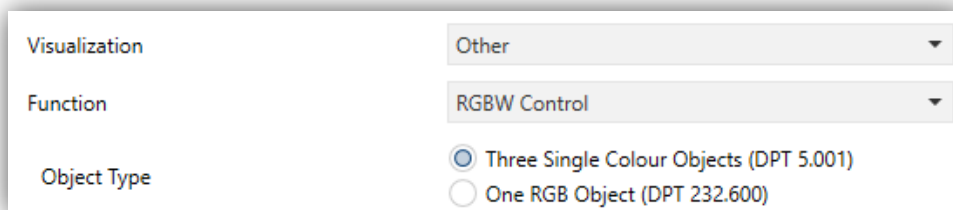


Figure 59. RGBW Control.

### 3.3.1.5.3 Daily Timer

Boxes configured as daily timers let the final user program a timed sending of a value to the bus, being possible to send binary or scene values (according to the parameterisation) at a certain time (either every day or only once) or at the end of a countdown. Hence, when the user presses on the box, a window containing the necessary controls for setting up a timed sending (either at a specific time or after a countdown) will pop up.



Figure 60. Daily Timer Pop up.



This window contains the following elements:

- **Timer Type Selector:** two-button selector to step through the following options:



			
Timer inactive.	Timer active (every day).	Timer active (no repetition).	Timer active (countdown).

Table 7. Timer Type.

- **Time Selector:** depending on the selected timer type, permits setting the time for the automated sending, or the length of the countdown.
- **Value to be Sent:** in case of having parameterised the sending of a binary value, this control lets the user determine the concrete value to be sent (“OFF” or “ON”). In the case of the sending of a scene, this control is not shown, as the scene number is set by parameter.

Hence, at the time set by the user or once the countdown ends, the bus will be automatically sent the corresponding value, through the “[Pn][Bi] Daily Timer Control” object (for binary values) or through the “[General] Scenes: Send” object (in case of scene values).

In both cases the “[Pn][Bi] Daily Timer Enabling” binary object will be available, making it possible to disable (by sending the value “0”) or to enable (value “1”) the execution of the timed sending set by the user. By default, timed sending is always enabled.

**Note:** in case of disabling a sending through “[Pn][Bi] Daily Timer Enabling”, the corresponding box will show the “forbidden” icon, although the box will not lose the configuration previously set by the user.

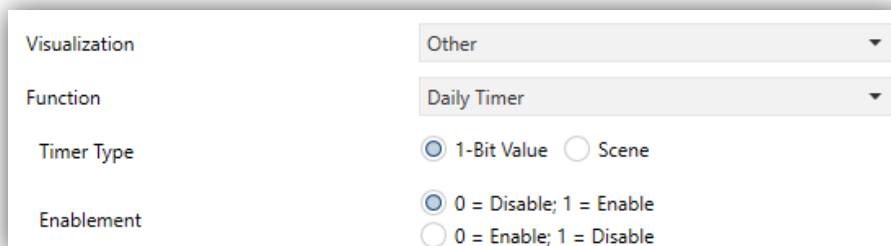


Figure 61. Daily Timer.

- **Timer Type** [*1-Bit Value/Scene*]: allows the user to select how the timer will be controlled.
  - **Scene Number** [*1...64*]: shown only after selecting “*Scene*” for the above parameter. This field defines the number of the scene that will be sent to the bus on the execution of the timed sending.
- **Enablement** [*0 = Disable; 1 = Enable/0 = Enable; 1 = Disable*]: sets the value to enable and disable the timer through the object “[Pn][Bi] **Daily Timer Enabling**”.

#### 3.3.1.5.4 Weekly Timer

Boxes configured as weekly timers let the final user program a timed sending of a binary or scene value to the bus (according to the parameterisation) at a certain time on certain days (on a weekly basis).

The difference between the daily and the weekly timers is that the latter does not permit the countdown-based sending, although it does permit that a time-based sending takes place every week, on the days selected by the user

Hence, when the user presses on the box, a window similar to Figure 62 will pop up.



Figure 62. Weekly Timer Pop-up.

This window contains the following elements:

- **Day Selector**: lets the user select the days the timed sending will take place on. The first day of the week depends on the parameterisation of the GENERAL tab (see section 3.1.1).
- **Value to be Sent (ON / OFF)**: lets the user set the value to be sent. It can be “ON” or “OFF”, or even both at different times. Once selector is provided for each of the two values.

**Note:** if instead of the sending of a binary value, the sending of a scene value has been parameterised, only one checkbox and once selector will be shown, so the user can enable/disable the sending of the scene value and, if enabled, set the desired time.

- **Time Selector:** let the user set the time at which the automatic sending will take place.

Hence, at the specified time on the days selected by the user, the bus will be automatically sent the corresponding value through the “[Pn][Bi] Weekly Timer Control” object (in the case of sending a binary value) or through the “[General] Scenes: Send” object (in the case of sending a scene number).

In both cases the “[Pn][Bi] Weekly Timer Enabling” binary object will be available, making it possible to disable (by sending the value “0”) or to enable (value “1”) the execution of the timings set by the user. By default, user timings are always enabled.

**Note:** in case of disabling a sending through “[Pn][Bi] Weekly Timer Enabling”, the corresponding box will show the “forbidden” icon, although the box will not lose the configuration previously set by the user.

Visualization	Other
Function	Weekly Timer
Timer Type	<input checked="" type="radio"/> 1-Bit Value <input type="radio"/> Scene
Label for "Time ON"	Time ON
Label for "Time OFF"	Time OFF
Enablement	<input checked="" type="radio"/> 0 = Disable; 1 = Enable <input type="radio"/> 0 = Enable; 1 = Disable

Figure 63. Weekly Timer.

This function offers the following parameters in ETS:

- **Timer Type:** [[1-Bit Value](#)/[Scene](#)]: allows the user to select how the timer will be controlled.
- **Scene Number** [[1...64](#)]: shown only after selecting “[Scene](#)” for the above parameter. This field defines the number of the scene that will be sent to the bus on the execution of the timed sending.

- **Label for “Scene”** [[Scene](#)]: permits customising the label shown next to the checkbox that enables/disables the scene sending.
- **Label for “Time ON”** [[Time ON](#)]: permits customising the label shown next to the checkbox that enables/disables the sending of the switch-on order.
- **Label for “Time OFF”** [[Time OFF](#)]: permits customising the label shown next to the checkbox that enables/disables the sending of the switch-off order.
- **Enablement** [[0 = Disable; 1 = Enable/0 = Enable; 1 = Disable](#)]: sets the value to enable and disable the timer through the object “[Pn][Bi] Weekly Timer Enabling”.

**Note:** *the execution of user-defined timers is triggered on the detection of state transitions, even if the clock does not specifically pass through the specified time. In other words, supposing one timed sending of the value “ON” at 10:00h am and one timed sending of the value “OFF” at 8:00h pm, if the internal clock is set to 9:00h pm just after 10:00h am (and therefore after having sent the value “ON”), then the “OFF” order corresponding to 8:00h pm will take place immediately after the time change.*

### 3.3.1.5.5 Alarm

Boxes configured as alarms perform user warnings on anomalous events. For this purpose, boxes of this type are related to the binary object “[Pn][Bi] Alarm Trigger” that permits receiving alarm messages from the bus, thus making Z35 emit a continuous beep while the screen light blinks. In addition, the screen will automatically browse to the page containing the alarm box that has been triggered. Any pop-up, message or function that has been activated will be deactivated and the alarm will be given priority.

The **blinking warning icon** shows that the alarm is active and not confirmed. This icon will also appear in the lower right corner of the button of the page where the alarm is located.

When an alarm has been activated, there are two different ways to silence the sound notification and the luminous blinking of the display.

- Pressing the ‘Home’ button: it will silence the sound notification and end the flashing, but it will not confirm the alarm. The alarm box icon will continue blinking.

- Press the 'OK' button in the alarm box: this will silence the sound notification, end the flashing light, confirm the alarm, and cause the icon to stop blinking. When this button is pressed, the binary object "[Pn] [Bi] Alarm Confirmation" with the value "1" will be sent through the bus. If this object receives from the bus the value "1", the alarm will also be confirmed, having the same effects.

The alarm will definitely become inactive once it has been confirmed and, in addition, once the "[Pn][Bi] Alarm Trigger" object goes back to the "no alarm" state (the order of these two events is irrelevant), after which the box icon and the page icon will finally turn off.

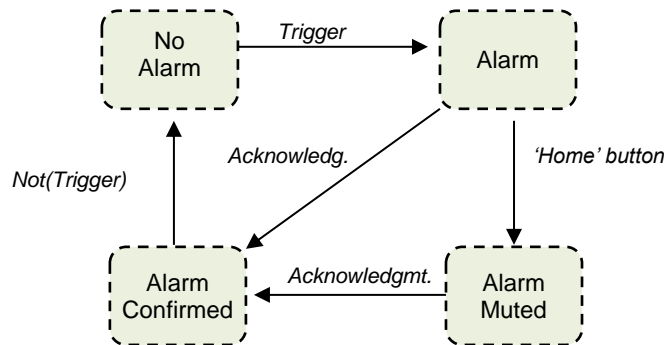


Figure 64 Alarm state diagram (case of confirmation previous to the alarm end).

There is also the possibility of **periodically monitoring** the status of the trigger object, for situations where this object is periodically received from the bus. This will let Z35 automatically assume the alarm situation if the "no alarm" value does not get sent through the trigger object after a certain time, for example upon failures of the transmitter. The maximum time window can be defined by parameter.

Consequently, alarm boxes permit configuring the following parameters:

Visualization	Other
Function	Alarm
Trigger	<input type="radio"/> 0 <input checked="" type="radio"/> 1
Periodic monitoring	<input checked="" type="radio"/> No <input type="radio"/> Yes
Protect	<input checked="" type="radio"/> No <input type="radio"/> Yes

Figure 65. Alarm.

- **Trigger** [1/0]: defines the value that will trigger the alarm (“0” or “1”; it is “1” by default), i.e., the value that, when received through “[Pn][Bi] Alarm Trigger”, should be interpreted by Z35 as an alarm situation. Implicitly, this parameter also defines the inverse “no alarm” value.
- **Periodic Monitoring** [No/Yes]: activates or deactivates periodic monitoring of the alarm trigger object. When activated, the following parameter appears:
  - **Cycle Time** [30...65535][s] [1...65535][min/h]: sets the maximum accepted time space without receiving the “no alarm” value before Z35 adopts the alarm situation.

**Example I:** *supposing a certain sensor responsible for sending the value “1” (once) to the bus when a flood is detected, and the value “0” (again, only once) when such situation terminates, if the object sent is linked to the alarm trigger object from an alarm box in Z35 where no periodical monitoring has been parameterised and where the value “1” has been set as the alarm value, then Z35 will notify an emergency every time a flood takes place. The notification will stop as soon as the user touches any button, although a warning icon will still blink on the screen. After that, if the user acknowledges the alarm, the icon will stop blinking; switching then off once the sensor sends a “0”.*

**Example II:** *suppose a certain CO<sub>2</sub> sensor responsible for sending the value “0” to the bus every two minutes, except when it detects a dangerous CO<sub>2</sub> level; in such case, it will immediately send the value “1”. If the object being sent is linked to the alarm trigger object from an alarm box in Z35 where periodic monitoring has been parameterised and where the value “1” is configured as the alarm trigger value, then Z35 will notify the user about the emergency both if the CO<sub>2</sub> level reaches a dangerous level and if the sensor stops sending the value “0”, for example due to failures or sabotage. The user should proceed in the same manner as in the above example to manage the alarm notification.*

## 3.4 CONFIGURATION PAGE

---

The Configuration page lets the user to know or adjust certain technical details about the device, as well as to configure the visual and sound settings.

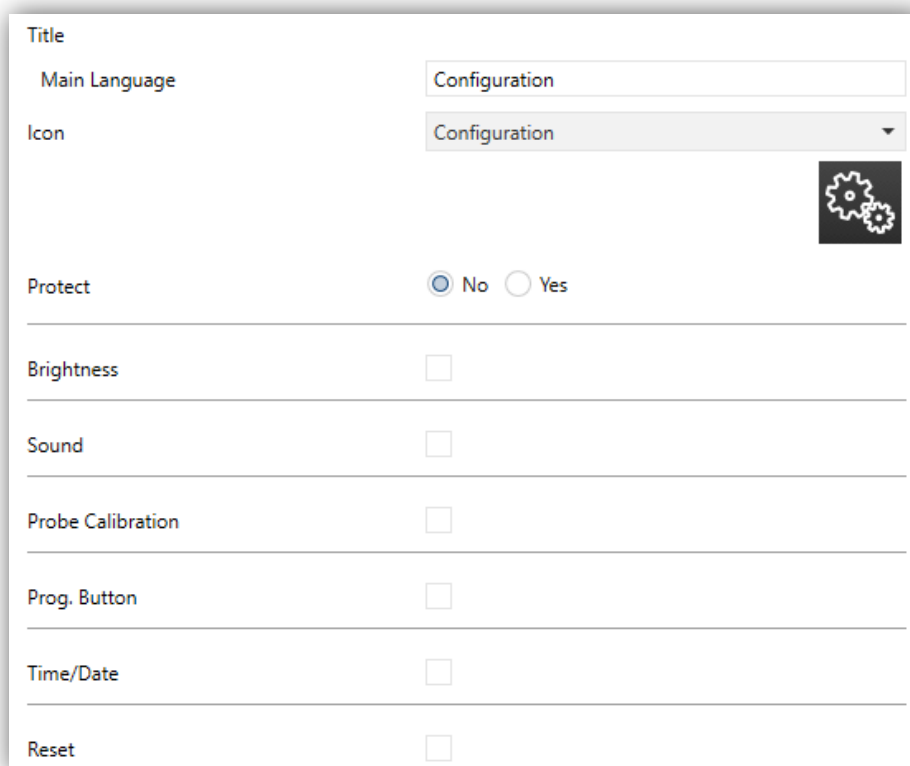
The integrator has previously had to enable this page on the "MENU" page.

---

### ETS PARAMETERISATION

---

After enabling the **Configuration Page** from "Configuration" screen under Menu (see section 3.1.1), a new tab will be incorporated into the tree on the left.



The screenshot shows a configuration page with the following fields and options:

- Title:** Configuration
- Main Language:** Configuration
- Icon:** Configuration (with a gear icon)
- Protect:**  No  Yes
- Brightness:**
- Sound:**
- Probe Calibration:**
- Prog. Button:**
- Time/Date:**
- Reset:**

Figure 66 Configuration Page.

The parameters of the page itself are:

- **Title:**

- **Language X [Configuration]:** text field that defines the title that will be shown on the top of the Configuration page for the corresponding language.

- **Icon** [[Configuration](#)]: sets the icon that will represent the page in the Menu page.
- **Protect**: sets whether the page will be password-protected or not. Depending on the security levels configured (one or two; see section 3.1.4) this list will contain the following options:
  - **One Level:**
    - [[No](#)]: the page will not be protected by password. All users can access it.
    - [[Yes](#)]: the page will be protected by password. Users will be asked to type the password when trying to access it.
  - **Two Levels:**
    - [[No](#)]: the page will not be protected by password. All users can access it.
    - [[Level 1](#)]: the page will implement security level 1. To access it, users will be required to enter password 1 or password 2.
    - [[Level 2](#)]: the page will implement security level 2. To access it, users will be required to enter password 2.

In addition, the specific controls that can be enabled for the Configuration page are:

- **Brightness** [[disabled/enabled](#)]: enables or disables the box to adjust the brightness of the display. In addition, the following settings can be configured:
- **Sound** [[disabled/enabled](#)]: enables or disables the box to deactivate/activate the device sounds.
- **Probe Calibration** [[disabled/enabled](#)]: enables or disables the box to configure the offset of the internal temperature probe.
- **Prog. Button** [[disabled/enabled](#)]: enables or disables the control/indicator that shows the status of the Prog./Test LED of the device. In particular, it permits entering/leaving the programming mode as by pressing the actual programming button of the device (see 1.3).
- **Time/Date** [[disabled/enabled](#)]: enables or disables the box to configure Z35 time and date.



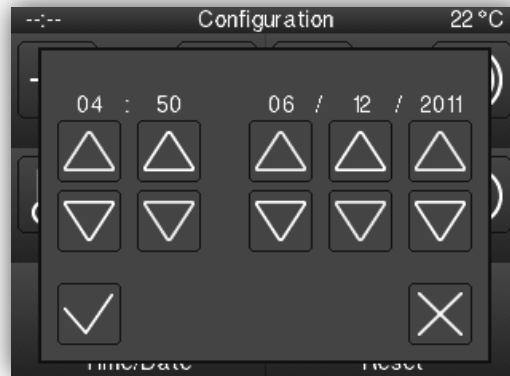


Figure 67 Time and date pop-up

- **Reset** [[disabled/enabled](#)]: enables or disables the box to perform a reset on the device, by pressing more than 3 seconds the button:
  - **Reset Type** [[Parameters Reset/Z35 Reboot](#)]: sets the reset type required:
    - [[Parameters Reset](#)]: similar to restoring the device to the just-parameterised state, with the subsequent reset of the object values, alarm controls, timers, etc.
    - [[Z35 Reboot](#)]: simple device reset, with no data loss.

Moreover, for all these controls, the following parameters can be configured:

- **Label** [[Brightness](#)]: text that will appear to identify the box. Again, new parameters will appear if several translations get enabled.
- **Protection** [[No/Yes](#)] / [[No/Level 1/Level 2](#)]: exactly the same as the page protection explained above. In this case, the user can enable the protection of the box.
- **Object to Show/Hide de Box** [[disabled/enabled](#)]: When this option is enabled, a new 1 bit object appears (“**[Config.][Bi] Show/Hide Box**”) that allows hiding / displaying the box through the KNX bus.

## 3.5 INPUTS

---

Z35 incorporates **four 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 (models ZN1IO-DETEC-P and ZN1IO-DETEC-X from Zennio).

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

### 3.5.1 BINARY INPUT

---

Please refer to the specific user manual “**Binary Inputs**”, available in the Z35 product section, at the Zennio website ([www.zennio.com](http://www.zennio.com)).

### 3.5.2 TEMPERATURE PROBE

---

Please refer to the specific user manual “**Temperature Probe**”, available in the Z35 product section, at the Zennio website ([www.zennio.com](http://www.zennio.com)).

### 3.5.3 MOTION DETECTOR

---

To connect motion detectors (models **ZN1IO-DETEC-P** and **ZN1IO-DETEC-X** from Zennio) to the input ports of Z35 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 Z35 product section at the Zennio website, [www.zennio.com](http://www.zennio.com)) for detailed information about the functionality and the configuration of the related parameters.

**Important:**

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

- *Motion detectors with references ZN1IO-DETEC and ZN1IO-DETEC-N are **not compatible** with Z35 (they may report inaccurate measurements if connected to this device).*
- *When connected to Z35, the rear micro-switch of model ZN1IO-DETEC-P should be set to position “**Type B**”.*

## 3.6 THERMOSTAT

---

Z35 implements **two Zennio thermostats** which can be enabled and fully customised.

Please refer to the specific manual “**Zennio Thermostat**” (available in the Z35 product section at the Zennio website, [www.zennio.com](http://www.zennio.com)) for detailed information about the functionality and the configuration of the related parameters.

## ANNEX I. 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	CT-WU	DPT_SceneNumber		[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	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
7	1 Bit	I	CT-WU	DPT_Ack	0/1	[General] Cleaning Function	0 = Nothing; 1 = Clean Now
8	2 Bytes	I	CT-WU	DPT_Value_Temp	-273.00 - 670760.00	[General] External Temperature	Temperature to Be Shown
9	1 Bit	I	CT-WU	DPT_Switch	0/1	[General] Backlight Mode	0 = Night Mode; 1 = Normal Mode
	1 Bit	I	CT-WU	DPT_Switch	0/1	[General] Backlight Mode	0 = Normal Mode; 1 = Night Mode
10	1 Byte	I	CT-WU	DPT_Scaling	0% - 100%	[General] Display - Brightness	0% ... 100%
11	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Illumination	0 = Nothing; 1 = Illuminate Display
12	1 Bit	I	CT-WU	DPT_Enable	0/1	[General] Luminosity Sensor	0 = Disable; 1 = Enable
13	1 Bit	O	CTR--	DPT_Switch	0/1	[General] Luminosity Object (1-bit)	0 = Over Threshold; 1 = Under Threshold
	1 Bit	O	CTR--	DPT_Switch	0/1	[General] Luminosity Object (1-bit)	0 = Under Threshold; 1 = Over Threshold
14	1 Byte	O	C-R--	DPT_Scaling	0% - 100%	[General] Luminosity Object (Percentage)	0% ... 100%
15	1 Bit	I	CT-WU	DPT_Enable	0/1	[General] Proximity Sensor	0 = Disable; 1 = Enable
16	1 Bit	I	C--W-	DPT_Ack	0/1	[General] Welcome Greeting	0 = Nothing; 1 = Welcome
17-20	14 Bytes	I	C--W-			[General] Welcome Greeting - Line X	Text to Show On the Display at Line X

21	1 Byte	I	C - - W -		0-5	[General] Translations - Select Language	0 = Main; 1 = Lang. 2; ...; 4 = Lang. 5
22	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Translations - Main Language	0 = Nothing; 1 = Choose this Language
23	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Translations - Language 2	0 = Nothing; 1 = Choose this Language
24	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Translations - Language 3	0 = Nothing; 1 = Choose this Language
25	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Translations - Language 4	0 = Nothing; 1 = Choose this Language
26	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Translations - Language 5	0 = Nothing; 1 = Choose this Language
27	2 Bytes	I	C - - W -	DPT_LanguageCodeAlpha2_ASCII		[General] Translations - Select Language	Language Selection Through ISO 639-1 Two Letters Code
28	1 Bit	I	C T - W U	DPT_Enable	0/1	[General] Sounds - Disabling Button Sound	0 = Disable Sound; 1 = Enable Sound
	1 Bit	I	C T - W U	DPT_Enable	0/1	[General] Sounds - Disabling Button Sound	0 = Enable Sound; 1 = Disable Sound
29	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
30	1 Bit	O	C T R - -	DPT_Switch	0/1	[General] Welcome Back Object	Switch Object Sent on Wake Up
31	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome Back Object - Additional Condition	Additional Condition Object 1
32	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome Back Object - Additional Condition	Additional Condition Object 2
33	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome Back Object - Additional Condition	Additional Condition Object 3
34	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome Back Object - Additional Condition	Additional Condition Object 4
35	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Welcome Back Object - Additional Condition	Additional Condition Object 5
36	1 Bit	I	C T - W U	DPT_Switch	0/1	[General] Temperature Scale	0 = °C; 1 = °F
37, 38, 39, 40, 41, 42	1 Bit	I	C - - W -	DPT_Switch	0/1	[Config.][By] Show/Hide Box	0 = Hide Box; 1 = Show Box
43, 58, 73, 88, 103, 118, 133, 148, 163, 178, 193, 208, 223, 238, 253, 268, 283, 298, 313,	1 Bit	I	C - - W -	DPT_Switch	0/1	[Px][By] Show/Hide Box	0 = Hide Box; 1 = Show Box

328, 343, 358, 373, 388, 403, 418, 433, 448, 463, 478, 493, 508, 523, 538, 553, 568, 583, 598, 613, 628, 643, 658							
44, 59, 74, 89, 104, 119, 134, 149, 164, 179, 194, 209, 224, 239, 254, 269, 284, 299, 314, 329, 344, 359, 374, 389, 404, 419, 434, 449, 464, 479, 494, 509, 524, 539, 554, 569, 584, 599, 614, 629, 644, 659	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Switch: "0"	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Switch: "1"	1-Bit Generic Control
	1 Bit	I/O	<b>CTRWU</b>	DPT_Switch	0/1	[Px][By] Switch: "0/1"	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Hold & Release	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Two Objects - Short Press: "1"	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Two Objects - Short Press: "0"	1-Bit Generic Control
	1 Bit	I/O	<b>CTRWU</b>	DPT_Switch	0/1	[Px][By] Two Objects - Short Press: "0/1"	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Switch	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Light On/Off	0 = Off; 1 = On
	1 Bit	O	<b>CTR--</b>	DPT_UpDown	0/1	[Px][By] Move Shutter	0 = Up; 1 = Down
	1 Bit	I/O	<b>CTRWU</b>	DPT_Switch	0/1	[Px][By] Two Objects - Short Press	1-Bit Generic Control
	1 Bit	I/O	<b>CTRWU</b>	DPT_Heat_Cool	0/1	[Px][By] (Climate) Mode	Left = Cool; Right = Heat
	1 Bit	O	<b>CTR--</b>	DPT_Step	0/1	[Px][By] Fan Control (Type: 1-Bit)	0 = Decrease; 1 = Increase
	1 Bit	I	<b>CT-WU</b>	DPT_Enable	0/1	[Px][By] Daily Timer Enabling	0 = Disable; 1 = Enable
	1 Bit	I	<b>CT-WU</b>	DPT_Enable	0/1	[Px][By] Daily Timer Enabling	0 = Enable; 1 = Disable
	1 Bit	I	<b>CT-WU</b>	DPT_Enable	0/1	[Px][By] Weekly Timer Enabling	0 = Disable; 1 = Enable
	1 Bit	I	<b>CT-WU</b>	DPT_Enable	0/1	[Px][By] Weekly Timer Enabling	0 = Enable; 1 = Disable
	1 Bit	I	<b>C--WU</b>	DPT_Alarm	0/1	[Px][By] Alarm trigger	Trigger: 0
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] 1-Bit Multimedia Control	0 = Stop/Backward; 1 = Start/Forward
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] 1-Bit Multimedia Control	0 = Start/Forward; 1 = Stop/Backward
1 Bit	I	<b>C--WU</b>	DPT_Alarm	0/1	[Px][By] Alarm trigger	Trigger: 1	
45, 60, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255, 270, 285, 300, 315, 330, 345, 360, 375, 390, 405, 420, 435, 450, 465, 480, 495,	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Two Objects - Long Press: "1"	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Two Objects - Long Press: "0"	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Two Objects - Long Press: "0/1"	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Step	0/1	[Px][By] Stop Shutter	0 = Stop/Step Up; 1 = Stop/Step Down
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Two Objects - Long Press	1-Bit Generic Control
	1 Bit	I/O	<b>CTRWU</b>	DPT_Switch	0/1	[Px][By] Fan Control - Auto Mode	Switch Auto Mode on Short Press

510, 525, 540, 555, 570, 585, 600, 615, 630, 645, 660	1 Bit	I/O	<b>CTRWU</b>	DPT_Switch	0/1	[Px][By] Fan Control - Auto Mode	Switch Auto Mode on Long Press
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Daily Timer Control	1-Bit Generic Control
	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Px][By] Weekly Timer Control	1-Bit Generic Control
	1 Bit	I	<b>CT-WU</b>	DPT_Ack	0/1	[Px][By] Alarm Confirmation	0 = No Action; 1 = Confirm
46, 61, 76, 91, 106, 121, 136, 151, 166, 181, 196, 211, 226, 241, 256, 271, 286, 301, 316, 331, 346, 361, 376, 391, 406, 421, 436, 451, 466, 481, 496, 511, 526, 541, 556, 571, 586, 601, 616, 631, 646, 661	4 Bit	O	<b>CTR--</b>	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Px][By] Light Dimming	4-Bit Dimming Control
47, 62, 77, 92, 107, 122, 137, 152, 167, 182, 197, 212, 227, 242, 257, 272, 287, 302, 317, 332, 347, 362, 377, 392, 407, 422, 437, 452, 467, 482, 497, 512, 527, 542, 557, 572, 587, 602, 617, 632, 647, 662	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] 1-Byte Unsigned Int Value	0 ... 255
	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Count	-128 - 127	[Px][By] 1-Byte Signed Int Value	-128 ... 127
	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Two Objects - Short Press: (1-Byte)	Send Selected 1-Byte Value on Short Press
	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] 1-Byte Scaling Value	0% ... 100%
	1 Byte	I/O	<b>CTRWU</b>	DPT_HVACContrMode	0 = Auto 1 = Heat 3 = Cool 9 = Fan 14 = Dry	[Px][By] (Climate) Mode - Extended	Auto, Heat, Cool, Fan and Dry
	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	100%
	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	50%, 100%
	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	33%, 67%, 100%
	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	25%, 50%, 75%, 100%
	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	20%, 40%, 60%, 80%, 100%
	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	1
	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	1, 2
	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	1, 2, 3
	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	1, 2, 3, 4
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	1, 2, 3, 4, 5	



					Enumeration)	
1 Byte	I/O	<b>CTRWU</b>	DPT_Scaling	0% - 100%	[Px][By] Red Channel	0% ... 100%
1 Byte	I/O	<b>CTRWU</b>	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Px][By] (Climate) Special Mode Control	Auto, Comfort, Standby, Economy and Building Protection
1 Byte	I/O	<b>CTRWU</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Enumerated Control	0 ... 255
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	0, 1
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	0, 1, 2, 3, 4, 5
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	0, 1, 2, 3, 4
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	0, 1, 2, 3
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	0, 1, 2
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	0%, 20%, 40%, 60%, 80%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	0%, 33%, 67%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	0%, 50%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	0%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	0%, 25%, 50%, 75%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	Auto, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	Auto, 20%, 40%, 60%, 80%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	Auto, 25%, 50%, 75%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	Auto, 33%, 67%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Control (Type: Scaling)	Auto, 50%, 100%
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	Auto, 1
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	Auto, 1, 2
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	Auto, 1, 2, 3
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	Auto, 1, 2, 3, 4
1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Control (Type: Enumeration)	Auto, 1, 2, 3, 4, 5
1 Byte		<b>CT---</b>	DPT_SceneControl	0-63; 128-191	[Px][By] Scene: Send	0-63 (Run Scene 1-64)
1 Byte		<b>CT---</b>	DPT_SceneControl	0-63; 128-191	[Px][By] Scene: Send	0-63 / 128-191 (Run/Save Scene 1-64)

48, 63, 78, 93, 108, 123, 138, 153, 168, 183, 198, 213, 228, 243, 258, 273, 288, 303, 318, 333, 348, 363, 378, 393, 408, 423, 438, 453, 468, 483, 498, 513, 528, 543, 558, 573, 588, 603, 618, 633, 648, 663	1 Byte	O	<b>CTR--</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Two Objects - Long Press: (1-Byte)	Send Selected 1-Byte Value on Long Press
	1 Byte	I/O	<b>CTRWU</b>	DPT_Scaling	0% - 100%	[Px][By] Green Channel	0% ... 100%
49, 64, 79, 94, 109, 124, 139, 154, 169, 184, 199, 214, 229, 244, 259, 274, 289, 304, 319, 334, 349, 364, 379, 394, 409, 424, 439, 454, 469, 484, 499, 514, 529, 544, 559, 574, 589, 604, 619, 634, 649, 664	1 Byte	I/O	<b>CTRWU</b>	DPT_Scaling	0% - 100%	[Px][By] Blue Channel	0% ... 100%
50, 65, 80, 95, 110, 125, 140, 155, 170, 185, 200, 215, 230, 245, 260, 275, 290, 305, 320, 335, 350, 365, 380, 395, 410, 425, 440, 455, 470, 485, 500, 515, 530, 545, 560, 575, 590, 605, 620, 635, 650, 665	1 Byte	I/O	<b>CTRWU</b>	DPT_Scaling	0% - 100%	[Px][By] White Channel	0% ... 100%
51, 66, 81, 96, 111, 126, 141, 156, 171, 186, 201, 216, 231, 246, 261, 276,	2 Bytes	O	<b>CTR--</b>	DPT_Value_2_Ucount	0 - 65535	[Px][By] 2-Byte Unsigned Int Value	0 ... 65535
	2 Bytes	O	<b>CTR--</b>	DPT_Value_2_Count	-32768 - 32767	[Px][By] 2-Byte Signed Int Value	-32768 ... 32767
	2 Bytes	O	<b>CTR--</b>	9.xxx	-671088.64 - 670760.96	[Px][By] 2-Byte Float Value	-671088.64 ... 670760.96
	2 Bytes	I/O	<b>CTRWU</b>	DPT_Value_Temp	-273.00 - 670760.00	[Px][By] (Climate) Temperature Setpoint	-99°C ... 199°C

291, 306, 321, 336, 351, 366, 381, 396, 411, 426, 441, 456, 471, 486, 501, 516, 531, 546, 561, 576, 591, 606, 621, 636, 651, 666							
52, 67, 82, 97, 112, 127, 142, 157, 172, 187, 202, 217, 232, 247, 262, 277, 292, 307, 322, 337, 352, 367, 382, 397, 412, 427, 442, 457, 472, 487, 502, 517, 532, 547, 562, 577, 592, 607, 622, 637, 652, 667	3 Bytes	I/O	<b>CTRWU</b>	DPT_Colour_RGB	[0 - 255] * 3	[Px][By] RGB Color	Red, Green and Blue Components
53, 68, 83, 98, 113, 128, 143, 158, 173, 188, 203, 218, 233, 248, 263, 278, 293, 308, 323, 338, 353, 368, 383, 398, 413, 428, 443, 458, 473, 488, 503, 518, 533, 548, 563, 578, 593, 608, 623, 638, 653, 668	1 Bit	I	<b>CT-WU</b>	DPT_Switch	0/1	[Px][By] Binary Indicator	1-Bit Indicator
54, 69, 84, 99, 114, 129, 144, 159, 174, 189, 204, 219, 234, 249, 264, 279, 294, 309, 324, 339, 354, 369, 384, 399, 414, 429, 444, 459, 474, 489, 504,	1 Byte	I	<b>CT-WU</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Enumerated Indicator	0 ... 255
	1 Byte	I	<b>CT-WU</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] 1-Byte Unsigned Int Indicator	0 ... 255
	1 Byte	I	<b>CT-WU</b>	DPT_Value_1_Count	-128 - 127	[Px][By] 1-Byte Signed Int Indicator	-128 ... 127
	1 Byte	I	<b>CT-WU</b>	DPT_Scaling	0% - 100%	[Px][By] Percentage Indicator	0% ... 100%
	1 Byte	I	<b>CT-WU</b>	DPT_Scaling	0% - 100%	[Px][By] Light Indicator	0% = Off; 100% = On
	1 Byte	I	<b>CT-WU</b>	DPT_Scaling	0% - 100%	[Px][By] Shutter Position	0% = Top; 100% = Bottom
	1 Byte	I	<b>CT-WU</b>	DPT_Scaling	0% - 100%	[Px][By] Fan Indicator	0% ... 100%
	1 Byte	I	<b>CT-WU</b>	DPT_Value_1_Ucount	0 - 255	[Px][By] Fan Indicator	Enumerated Value

519, 534, 549, 564, 579, 594, 609, 624, 639, 654, 669							
55, 70, 85, 100, 115, 130, 145, 160, 175, 190, 205, 220, 235, 250, 265, 280, 295, 310, 325, 340, 355, 370, 385, 400, 415, 430, 445, 460, 475, 490, 505, 520, 535, 550, 565, 580, 595, 610, 625, 640, 655, 670	2 Bytes	I	CT - W U	DPT_Value_2_Ucount	0 - 65535	[Px][By] 2-Byte Unsigned Int Indicator	0 ... 65535
	2 Bytes	I	CT - W U	DPT_Value_2_Count	-32768 - 32767	[Px][By] 2-Byte Signed Int Indicator	-32768 ... 32767
	2 Bytes	I	CT - W U	9.xxx	-671088.64 - 670760.96	[Px][By] 2-Byte Float Indicator	-671088.64 ... 670760.96
56, 71, 86, 101, 116, 131, 146, 161, 176, 191, 206, 221, 236, 251, 266, 281, 296, 311, 326, 341, 356, 371, 386, 401, 416, 431, 446, 461, 476, 491, 506, 521, 536, 551, 566, 581, 596, 611, 626, 641, 656, 671	2 Bytes	I	CT - W U	DPT_Value_Temp	-273.00 - 670760.00	[Px][By] Temperature Indicator	-99°C ... 199°C
	4 Bytes	I	CT - W U	DPT_Value_4_Count	-2147483648 - 2147483647	[Px][By] 4-Byte Signed Int Indicator	-2147483648 ... 2147483647
57, 72, 87, 102, 117, 132, 147, 162, 177, 192, 207, 222, 237, 252, 267, 282, 297, 312, 327, 342, 357, 372, 387, 402, 417, 432, 447, 462, 477, 492, 507, 522, 537, 552, 567, 582, 597, 612, 627, 642, 657, 672	4 Bytes	I	CT - W U	14.xxx	-2147483648 - 2147483647	[Px][By] 4-Byte Float Indicator	-2147483647 ... 2147483647
	14 Bytes	I	CT - W U			[Px][By] 14-Byte Text Indicator	Text String
673, 677, 681,	14 Bytes	I	CT - W U			[Px][By] Multimedia Indicator	14-Bytes Text String
	2 Bytes	O	CTR --	DPT_Value_Temp	-273.00 - 670760.00	[Ix] Current Temperature	Temperature sensor value

685							
674, 678, 682, 686	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overcooling	0 = No Alarm; 1 = Alarm
675, 679, 683, 687	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Overheating	0 = No Alarm; 1 = Alarm
676, 680, 684, 688	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Probe Error	0 = No Alarm; 1 = Alarm
689	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Internal Temp. Probe] Current Temperature	Temperature sensor value
690	1 Bit	O	CTR--	DPT_Alarm	0/1	[Internal Temp. Probe] Overcooling	0 = No Alarm; 1 = Alarm
691	1 Bit	O	CTR--	DPT_Alarm	0/1	[Internal Temp. Probe] Overheating	0 = No Alarm; 1 = Alarm
692	1 Bit	O	CTR--	DPT_Bool	0/1	[Internal Temp. Probe] Probe Error	0 = No Alarm; 1 = Alarm
693, 699, 705, 711	1 Bit	I	C--W-	DPT_Enable	0/1	[Ix] Input Lock	0 = Unlock; 1 = Lock
694, 700, 706, 712	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)	[Ix] [Short Press] Brighter/Darker	Switch Bright/Dark	

					0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)		
	1 Bit		<b>CT---</b>	DPT_Switch	0/1	[Ix] [Short Press] Light On	Sending of 1 (On)
	1 Bit		<b>CT---</b>	DPT_Switch	0/1	[Ix] [Short Press] Light Off	Sending of 0 (Off)
	1 Bit	I	<b>CT-W-</b>	DPT_Switch	0/1	[Ix] [Short Press] Light On/Off	Switching 0/1
	1 Byte		<b>CT---</b>	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Run Scene	Sending of 0 - 63
	1 Byte		<b>CT---</b>	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Save Scene	Sending of 128 - 191
	1 Bit	I/O	<b>CTRW-</b>	DPT_Switch	0/1	[Ix] [Switch/Sensor] Edge	Sending of 0 or 1
	1 Byte		<b>CT---</b>	DPT_Value_1_Ucount	0 - 255	[Ix] [Short Press] Constant Value (Integer)	0 - 255
	1 Byte		<b>CT---</b>	DPT_Scaling	0% - 100%	[Ix] [Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		<b>CT---</b>	DPT_Value_2_Ucount	0 - 65535	[Ix] [Short Press] Constant Value (Integer)	0 - 65535
	2 Bytes		<b>CT---</b>	9.xxx	-671088.64 - 670760.96	[Ix] [Short Press] Constant Value (Float)	Float Value
695, 701, 707, 713	1 Byte	I	<b>C--W-</b>	DPT_Scaling	0% - 100%	[Ix] [Short Press] Shutter Status (Input)	0% = Top; 100% = Bottom
	1 Byte	I	<b>C--W-</b>	DPT_Scaling	0% - 100%	[Ix] [Short Press] Dimming Status (Input)	0% - 100%
696, 702, 708, 714	1 Bit		<b>CT---</b>	DPT_Switch	0/1	[Ix] [Long Press] 0	Sending of 0
	1 Bit		<b>CT---</b>	DPT_Switch	0/1	[Ix] [Long Press] 1	Sending of 1
	1 Bit	I	<b>CT-W-</b>	DPT_Switch	0/1	[Ix] [Long Press] 0/1 Switching	Switching 0/1
	1 Bit		<b>CT---</b>	DPT_UpDown	0/1	[Ix] [Long Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		<b>CT---</b>	DPT_UpDown	0/1	[Ix] [Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		<b>CT---</b>	DPT_UpDown	0/1	[Ix] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		<b>CT---</b>	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step Up)
	1 Bit		<b>CT---</b>	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step Down)
	1 Bit		<b>CT---</b>	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Shutter (Switched)	Switching of 0/1 (Stop/Step Up/Down)
	4 Bit		<b>CT---</b>	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
697, 703, 709, 715	1 Bit		CT---	DPT_Trigger	0/1	[Ix] [Long Press/Release] Stop Shutter	Release -> Stop Shutter
698, 704, 710, 716	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
717	1 Byte	I	C--W-	DPT_SceneControl	0-63; 128-191	[Motion Detector] Scene Input	Scene Value
718	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Motion Detector] Scene Output	Scene Value
719, 748, 777, 806	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix] Luminosity	0-100%
720, 749, 778, 807	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
721, 750, 779, 808	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error

722, 751, 780, 809	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Ix] Presence State (Scaling)	0-100%
723, 752, 781, 810	1 Byte	O	<b>CTR--</b>	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
724, 753, 782, 811	1 Bit	O	<b>CTR--</b>	DPT_Occupancy	0/1	[Ix] Presence State (Binary)	Binary Value
	1 Bit	O	<b>CTR--</b>	DPT_Ack	0/1	[Ix] Presence: Slave Output	1 = Motion Detected
725, 754, 783, 812	1 Bit	I	<b>C--W-</b>	DPT_Window_Door	0/1	[Ix] Presence Trigger	Binary Value to Trigger the Presence Detection
726, 755, 784, 813	1 Bit	I	<b>C--W-</b>	DPT_Ack	0/1	[Ix] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
727, 756, 785, 814	2 Bytes	I	<b>C--W-</b>	DPT_TimePeriodSec	0 - 65535	[Ix] Presence: Waiting Time	0-65535 s.
728, 757, 786, 815	2 Bytes	I	<b>C--W-</b>	DPT_TimePeriodSec	1 - 65535	[Ix] Presence: Listening Time	1-65535 s.
729, 758, 787, 816	1 Bit	I	<b>C--W-</b>	DPT_Enable	0/1	[Ix] Presence: Enable	According to parameters
730, 759, 788, 817	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Ix] Presence: Day/Night	According to parameters
731, 760, 789, 818	1 Bit	O	<b>CTR--</b>	DPT_Occupancy	0/1	[Ix] Presence: Occupancy State	0 = Not Occupied; 1 = Occupied
732, 761, 790, 819	1 Bit	I	<b>C--W-</b>	DPT_Ack	0/1	[Ix] External Motion Detection	0 = Nothing; 1 = Motion detected by an external sensor
733, 738, 743, 762, 767, 772, 791, 796, 801, 820, 825, 830	1 Byte	O	<b>CTR--</b>	DPT_Scaling	0% - 100%	[Ix] [Cy] Detection State (Scaling)	0-100%
734, 739, 744, 763, 768, 773, 792, 797, 802, 821, 826, 831	1 Byte	O	<b>CTR--</b>	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] [Cy] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
735, 740, 745, 764, 769, 774, 793, 798, 803, 822, 827, 832	1 Bit	O	<b>CTR--</b>	DPT_Switch	0/1	[Ix] [Cy] Detection State (Binary)	Binary Value
736, 741, 746, 765, 770, 775, 794, 799, 804, 823, 828, 833	1 Bit	I	<b>C--W-</b>	DPT_Enable	0/1	[Ix] [Cy] Enable Channel	According to parameters
737, 742, 747, 766, 771, 776, 795, 800, 805, 824, 829, 834	1 Bit	I	<b>C--W-</b>	DPT_Switch	0/1	[Ix] [Cy] Force State	0 = No Detection; 1 = Detection



835	1 Byte	I	C--W-	DPT_SceneControl	0-63; 128-191	[Thermostat] Scene Input	Scene Value
836, 866	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Temperature Source 1	External Sensor Temperature
837, 867	2 Bytes	I	C--W-	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Temperature Source 2	External Sensor Temperature
838, 868	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Effective Temperature	Effective Control Temperature
839, 869	1 Byte	I	C--W-	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode	1-byte HVAC Mode
840, 870	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
841, 871	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
842, 872	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
843, 873	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
844, 874	1 Bit	I	C--W-	DPT_Window_Door	0/1	[Tx] Window Status (Input)	0 = Closed; 1 = Open
845, 875	1 Bit	I	C--W-	DPT_Ack	0/1	[Tx] Comfort Prolongation	0 = Nothing; 1 = Timed Comfort
846, 876	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode Status	1-byte HVAC Mode
847, 877	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
848, 878	1 Bit	I	C--W-	DPT_Step	0/1	[Tx] Setpoint Step	0 = -0.5°C; 1 = +0.5°C
849, 879	2 Bytes	I	C--W-	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Setpoint Offset	Float Offset Value
850, 880	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Setpoint Status	Current Setpoint
851, 881	2 Bytes	O	CTR--	DPT_Value_Temp	-273.00 - 670760.00	[Tx] Basic Setpoint Status	Current Basic Setpoint
852, 882	2 Bytes	O	CTR--	DPT_Value_Tempd	-670760.00 - 670760.00	[Tx] Setpoint Offset Status	Current Setpoint Offset
853, 883	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
854, 884	1 Bit	I	C--W-	DPT_Heat_Cool	0/1	[Tx] Mode	0 = Cool; 1 = Heat
855, 885	1 Bit	O	CTR--	DPT_Heat_Cool	0/1	[Tx] Mode Status	0 = Cool; 1 = Heat
856, 886	1 Bit	I	C--W-	DPT_Switch	0/1	[Tx] On/Off	0 = Off; 1 = On
857, 887	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] On/Off Status	0 = Off; 1 = On
858, 888	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] Control Variable (Cool)	PI Control (Continuous)
859, 889	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Tx] Control Variable (Heat)	PI Control (Continuous)
860, 890	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Cool)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Cool)	PI Control (PWM)

861, 891	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Heat)	2-Point Control
	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Control Variable (Heat)	PI Control (PWM)
862, 892	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Additional Cool	Temp >= (Setpoint+Band) => "1"
863, 893	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] Additional Heat	Temp <= (Setpoint-Band) => "1"
864, 894	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] PI State (Cool)	0 = PI signal 0%; 1 = PI signal greater than 0%
865, 895	1 Bit	O	CTR--	DPT_Switch	0/1	[Tx] PI State (Heat)	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