



Z41 LITE
TOUCH PANEL

KNX Capacitive Colour Touch Panel

ZVI-Z41LIT

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

www.zennio.com

Contents

1	Introduction	4
1.1	Z41 Lite	4
1.2	Functionality.....	5
1.3	Installation.....	6
1.3.1	Firmware Updates.....	8
2	Configuration.....	9
2.1	Menu Page	9
2.2	General-Purpose Pages	10
2.3	Profile Page	10
2.4	Configuration Page.....	12
2.5	Backlight	14
2.6	Page Shaping	15
3	ETS Parameterisation	17
3.1	Default Configuration.....	17
3.2	Main Configuration	19
3.2.1	General.....	20
3.2.2	Security.....	22
3.2.3	Touch Lock	25
3.2.4	Internal Temperature Sensor	27
3.2.5	Energy Monitor Objects	28
3.2.6	White-Channel Objects	29
3.2.7	Firmware Update	29
3.3	Menu	30
3.3.1	Configuration	30
3.4	Page <i>n</i>	31
3.4.1	Configuration	31
3.4.2	Box <i>i</i>	33
3.4.2.1	Indicator	33
3.4.2.2	1-button Control.....	38
3.4.2.3	2-button Control.....	41

3.4.2.4 Climate Control..... 50

3.4.2.5 Other..... 58

3.5 Profile Page 74

3.5.1 Configuration 74

3.6 Configuration Page..... 75

3.6.1 Configuration 75

3.7 Thermostat n..... 76

3.8 Inputs..... 77

ANNEX I. Communication Objects..... 78

1 INTRODUCTION

1.1 Z41 LITE

Z41 Lite is an easily and intuitively controllable high-performance **colour touch screen**. Its built-in features and functions make it 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 Z41 Lite are:

- **4.1 inch** backlit **capacitive** touch panel.
- User interface with various pre-set **colour patterns** and themes.
- Multiple **direct action functions**, fully customisable.
- Control distribution across up to **6 customisable pages**.
- Full **climate** management.
- Programmable **timers**.
- **Scene** control.
- **Alarm** control.
- 2 independent **thermostats**.
- Built-in temperature sensor.
- **2 customisable analogue-digital inputs**.
- **Elegant** design, available in various **colours**.

Note: overall, Z41 Lite offers similar functionality to other Z41 products from Zennio, although some specific design features have been slimmed down to make it an even more competitive product. Please note that Z41 Lite **does not incorporate an Ethernet interface**, and therefore is not intended for installations where a remote control (i.e., through the Z41 Remote application) of the device is required.

1.2 FUNCTIONALITY

The Z41 Lite application program features the following functions:

- **6 General-Purpose Pages, with up to 8 Fully-Customisable Boxes each,** which the integrator may configure as indicators or controls.

- **Indicators:**

- Binary (icon or text).
- Enumerated (icon or text).
- Unsigned integer (8 / 16 bits).
- Signed integer (8 / 16 bits).
- Scaling (percentage).
- Floating point (16 bits).

- **1-button Controls:**

- Binary (pre-set value; switch; hold & release; short press, long press...).
- Unsigned integer (8 / 16 bits).
- Signed integer (8 / 16 bits).
- Scaling (percentage).
- Floating point (16 bits).
- Scene (run / save).

- **2-button Controls:**

- Binary (icon, text, number).
- Enumerated (icon, text).
- Unsigned integer (8 / 16 bits).
- Signed integer (8 / 16 bits).
- Percentage (Scaling).
- Floating point (16 bits).
- 2 scenes (run / save).
- Shutters.
- Light dimming.

- **Climate Specific Controls:**

- Temperature setpoint control.
- Mode (cool/heat or extended).
- Special modes
- Fan.

- **Other Controls:**

- RGB.
- RGBW.
- Energy consumption (4 Bytes).
- Daily timer.
- Weekly timer.
- Alarm.

- **1 Profile Page** (optionally), which lets the user modify, directly from the user interface, parameters such as the brightness, contrast and colour pattern of the screen, and the sound of the button press beeps.
- **1 Configuration Page** (optionally), which contains the time/date setting, the built-in temperature probe calibration and the alarm log, among others.

1.3 INSTALLATION

Figure 1 shows the connection outline of Z41 Lite:

- 1.- Mini-USB connector.
- 2.- External power input.
- 3.- Temperature sensor.
- 4.- KNX bus connection.
- 5.- Analogue/digital inputs.
- 6.- Slot for the button battery.
- 7.- Prog./Test button.
- 8.- Prog./Test LED.
- 9.- Magnet.

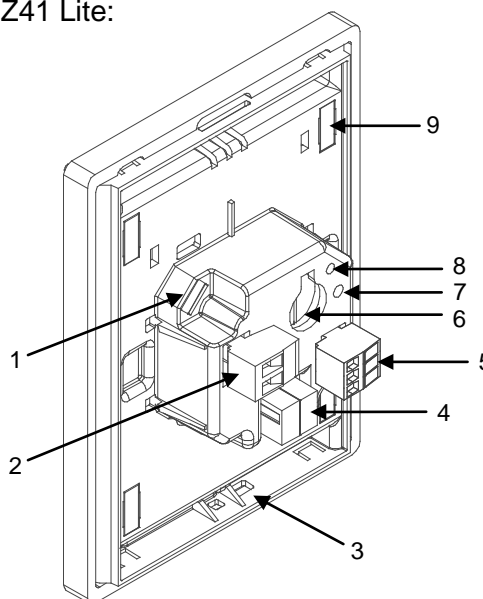


Figure 1 Schematic diagram

Z41 Lite connects to the KNX bus through the built-in terminal (4). An external **DC power supply** (12v / 24v / 29v) which delivers at least 150 mA of current is also required. The KNX power and the external power must be electrically isolated and provided by separate supplies, in order to avoid interference to the KNX bus. This external power must be supplied to the device through the corresponding terminal (2) provided within the original packaging, ensuring that each cable is properly connected and respecting the electrical polarities –positive and negative– from the supply to the device.

In addition to the external supply, Z41 Lite makes use of a **button battery** (LR44), located at the corresponding slot (6), to help maintain the time and date updated in case of a failure of the external power supply.

Note: immediately after connecting the external supply, the device will perform an automatic calibration, which may result deviated if it takes place while any type of pressure is being made over the front panel. In case of an unexpected behaviour in touch recognition, it will be necessary to disconnect and connect the device back, avoiding making pressure over the panel, in order to perform a new calibration.

After the connection of the external power supply, the screen will blink for an instant. A few seconds later, a **welcome screen** containing the Zennio logo and the “Loading...” label will be shown. Once the load process ends, the **Status page** will come up (see Figure 2).



12:45	Zennio Z41 Lite	29.7°C
01.01.001 KNX	OK Software	
0% Sync	1.0.0 SW Version	

Figure 2 Status Page

This screen is divided into boxes, which contain the following information:

- The “**KNX**” box shows the individual address assigned to the screen (in case of no assignment, this will show as “x.x.x”).
- The “**Software**” box will show empty in absence of an application program. In general, it will indicate if the download software is correct (“OK”) or if it is found to be corrupt due to download issues (“ERROR”).
- The “**Sync**” box shows, as a percentage, the progress of the internal synchronisation of the device during the start-up. Once it reaches 100%, the Status page will disappear and the Menu page will come up.
- Last, the “**SW Version**” box shows the version of the application program currently loaded to the device.

The Status page will also show up during bus failures in case the external power (12-24-29v) is not interrupted.

Note: *although the device will not turn functional until it is powered with the external supply, the KNX bus power should be enough to perform downloads from ETS (application program, group addresses, etc.).*

A short press on the **programming button** (7) will make the device enter the programming mode. The **programming LED** (8) will then light in red. On the contrary, if this button is held while the device gets connected to the bus, Z41 Lite will enter the **safe mode**. In such case, the programming LED will blink in red colour.

The Mini-USB connector (1) is provided for firmware updates (see section 1.3.1).

For detailed information about the technical features of Z41 Lite, as well as on security and installation procedures, please refer to the device **Datasheet**, included within the device packaging and also available at www.zennio.com.

1.3.1 FIRMWARE UPDATES

Z41 Lite incorporates **two separate microcontrollers**. One of them is focused on interfacing with the KNX bus and on running the application program itself, while the second one is dedicated to running the firmware that implements the operating system and the management of the peripherals.

Therefore, an occasional update of Z41 Lite may consist in two phases:

- Downloading the new **application program** (from ETS),
- Downloading the new **firmware**.

For the latter, it is necessary to connect of a **flash memory USB drive** to the Mini-USB port (see 5 in Figure 1), through the proper adapter cable. Please note that Z41 Lite does not provide an Ethernet interface for firmware update purposes.

For further details on the update process, please refer to the specific document “**Z41 – USB firmware update**”, available at www.zennio.com.

2 CONFIGURATION

2.1 MENU PAGE

The user interface of the Z41 Lite touch screen is organised into **pages** (up to eight different pages), each of which can be accessed from the **Menu page**, which (unless the contrary has been parameterised) is automatically shown after the start-up.

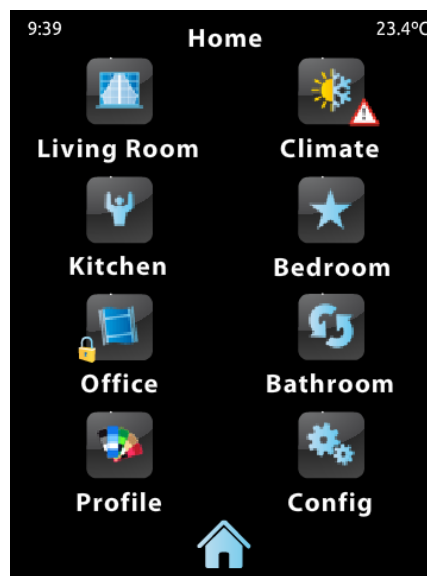


Figure 3 Menu Page (example)

As Figure 3 shows, the access to the pages is provided by a set of icons, which may show other (smaller) icons overlaid, such as an **exclamation** or a **lock**. Exclamations reflect that there are active alarms (see epigraph g) in section 3.4.2.5) within the page, while a lock reflects that the access to the page is password-protected (see section 3.2.2).

On the other hand, 6 out of the 8 pages are **general-purpose** pages at the entire disposal of the integrator, who may include up to 8 controls/indicators (with different functionalities) per page, being even possible to combine alarm, climate or any other controls within the same page. The remaining two pages (**Profile** and **Configuration**) are specific-purpose, as they are provided for user customisation of the device in runtime.

The user interface permanently shows a top bar (containing the name of the current page as well as the current time of day and temperature) and a button at the bottom of every page to get back to the Menu page:



Figure 4 Menu button

2.2 GENERAL-PURPOSE PAGES

The integrator can make use of up to 6 general-purpose pages, each of which can hold up to 8 different indicators or controls (with no restrictions for combining them) which will show inside the pages, automatically distributed or not, according to page parameterisation. For more details on the box distribution, refer to section 2.6.

The parameterisable controls for the general-purpose pages and their behaviour will be described in detail in section 3.

2.3 PROFILE PAGE

The Profile page lets the final user perform certain adjustments on the visual and audible side of the user interface:

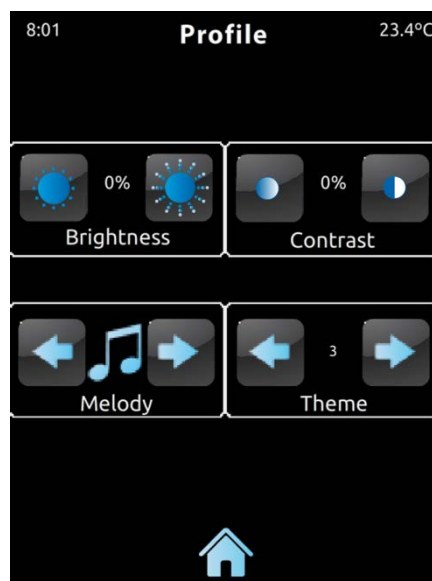


Figure 5 Profile Page

The specific controls that can be made available for the Profile page are:

- **Contrast.**
- **Brightness.**
- **Theme:** desired colour combination, among eight different options:



Figure 6 Themes

- **Melody:** desired tune (among three different options) for the beeps emitted on button presses or as a feedback on action executions. One more possibility is to make these beeps silent (alarm beeps will still sound).

Note that the labels of the controls in the Profile page, as well as the title of the page itself, are customisable in ETS. Moreover, it is possible to make any of the described

controls remain hidden for the final user. These details are explained in section 3, together with the entire parameterisation process.

2.4 CONFIGURATION PAGE

The Configuration page lets the user know or adjust certain technical values about the device.

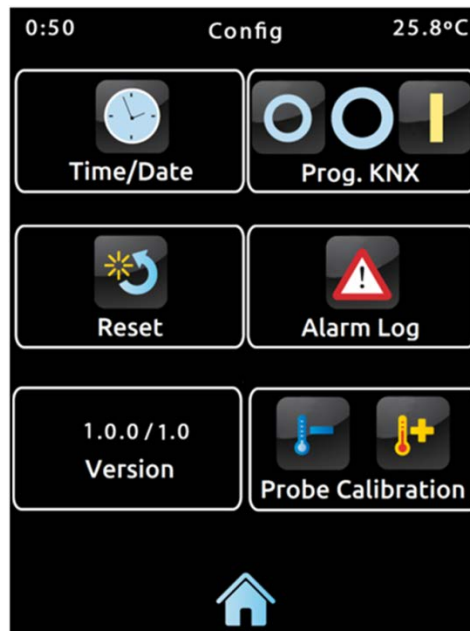


Figure 7 Configuration Page

- Date / Time:** even though the device implements two communication objects (see section 3.1) for setting the internal time and date from an external device, they can also be adjusted (if such option has been enabled by parameter) directly from the user interface. Therefore, by pressing this button, a set of controls will pop up to let the user set the current hour (0-23), minutes, day, month and year.



Figure 8 Time Set Control

Z41 features an internal clock powered by a battery (see section 1.3), because of which the time/date will not be lost after downloads or reboots.

- **Programming Button:** control/indicator that shows the status of the programming LED of the device. In particular, it permits entering/leaving the programming mode as by pressing the actual programming button of the device (see section 1.3) in case the rear side of Z41 Lite cannot be accessed.
- **Reset:** holding this button for a few seconds (i.e., a long press is required) sets the device back to a certain state, which can be set in ETS:
 - ETS Reset Device: similar to a device reset from ETS, with the subsequent reset of the object values, alarm controls, timers, etc.
 - Z41 Reboot: simple device reset, with no data loss (zeroing).
 - ETS Reset Device and Z41 Reboot: the above two, combined.

Note: *alarm logs, consumption logs or any other historic logs are not reset to zero in any case (such thing takes only place after a firmware download; see section 1.3.1). Nevertheless, a device reset from ETS does reset any adjustment performed by the final user from the Profile and Configuration pages (excepting the time / date, which is controlled by an internal clock).*

- **Alarm Log:** log file where alarm events will be incorporated as soon as they occur. This centralised file may be useful when the user needs to check the status of a set of alarm controls distributed across several pages.

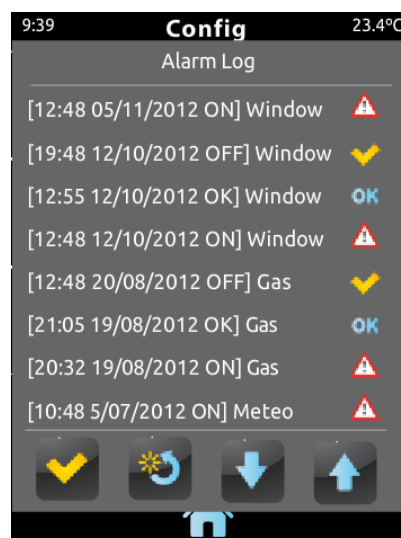


Figure 9 Alarm Log

Hence, the alarm log will contain a listing, in reverse chronological order, where the name, time and date of every alarm event will be shown. In addition, the log will show the current status of each of them, which may be:

- **“ON”**: the alarm is still active or has not been acknowledged by the user.
- **“OK”**: the user has acknowledged the alarm event, but the corresponding object is still in the alarm state.
- **“OFF”**: the user has acknowledged the alarm event and the corresponding object has already left the alarm state.

For more information on alarms, see epigraph g) in section 3.4.2.5.

- **Software Version**: indicator that shows the version of the device firmware currently installed on the device (see section 1.3.1).
- **Probe Calibration**: control for setting a certain correction over the values thrown by the internal temperature probe. This way, if the value of the measurement is found to be slightly over (or below) the actual temperature due to external factors, the final user will have the option to set an adjustment of between -5.0°C and +5.0°C, in steps of 0.1°C. In addition, it is possible to set an initial value for such correction by parameter.

Note: any changes made on the calibration of the internal probe will be applied beginning with the next measurement performed. Therefore, they may not be perceived immediately.

The labels of the controls in the Configuration page, as well as the title of the page itself, are customisable in ETS. Moreover, it is possible to make any of the described controls stay hidden for the final user. These details are explained in section 3, together with the entire parameterisation process.

2.5 BACKLIGHT

In order to prevent unnecessary power consumption while the device is idle, Z41 automatically fades partially out the backlight of the screen 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 (see section 3.2.1).

2.6 PAGE SHAPING

Each of the pages of the user interface –including Menu and Profile– can be independently parameterised to display its control/indicator boxes statically distributed or automatically ordered.

- **Static (grid) Distribution:** all the 8 boxes of the page will be shown, remaining empty those not having been assigned a function.

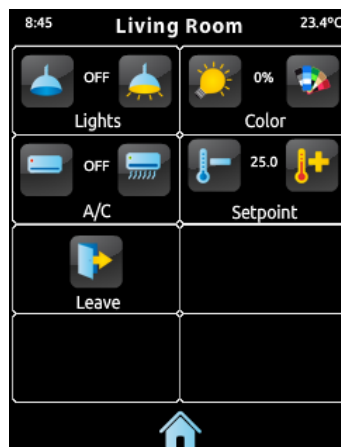


Figure 10 Static Distribution

- **Automatic Distribution:** the active boxes will be dynamically ordered, so that they make use of all the available space. The final distribution of the page will therefore depend on the number of boxes with a function assigned.

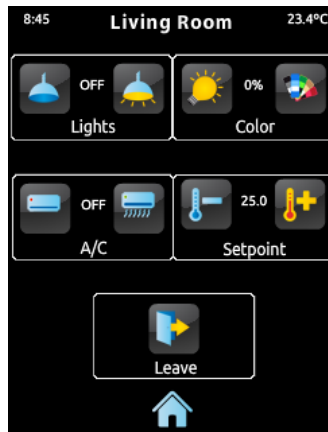


Figure 11 Dynamic Distribution of the enabled boxes

3 ETS PARAMETERISATION

To begin with the parameterisation of Z41 Lite, once the ETS program is running it is necessary to import the product database (**Z41 Lite** application program). Next, the device is added to the project and, after right-clicking on the name of the device, the option “Edit parameters” must be selected to start the configuration process.

The next sections explain the ETS parameterisation of the device in depth.

Note: *the amount of characters permitted for text parameters (labels, etc.) by ETS may differ depending on whether they contain special characters (symbols, accents...) coded with more than one byte. It is advisable to verify that these texts fit properly on the screen, with independence of whether they reach the maximum text length allowed by ETS or not.*

Important: *whenever the device is updated to a new version of the application program from ETS, it is also necessary to update the firmware of the secondary microcontroller (responsible for the operating system; see section 1.3.1) in case an updated version exists. If the installed versions of the application program and the firmware do not match, the device may not work correctly (the Software box in the Status page will show the word “Error”).*

3.1 DEFAULT CONFIGURATION

This section illustrates the default initial state of the device.

1	[General] Time	Current Time	3 Bytes	C	R	W	T	U	time of day
2	[General] Date	Current Date	3 Bytes	C	R	W	T	U	date
3	[General] Scenes: send	Scene Value	1 Byte	C	-	-	T	-	scene control
4	[General] Internal Temperature	Current Temperature (built-in sensor)	2 Bytes	C	R	-	T	-	temperature (°C)
6	[General] Illumination	0=No action; 1=Light Display	1 bit	C	-	W	-	-	acknowledge
7	[General] Touch Lock	0=Unlocked; 1=Locked	1 bit	C	-	W	-	-	switch

Figure 12 Default Topology

The general objects enabled by default are:

- **[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, Z41 Lite calculates it from the date and therefore ignores that field.*

- **[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] Scenes: send:** 1-byte object through which scene run/save orders (1 to 64 according to the KNX standard) can be sent to the bus in case any of the boxes is configured as a scene control or as a timer.
- **[General] Internal 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 3.2.4).
- **[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 2.5.
- **[General] Touch Lock:** 1-bit object that, after the reception of the value “1” from the bus, will lock the touch panel so that user presses are ignored from that moment. Touch locking will be interrupted as soon as the value “0” is received through this object. See section 3.2.3.

During the parameter configuration, a screen similar to Figure 13 will be shown.

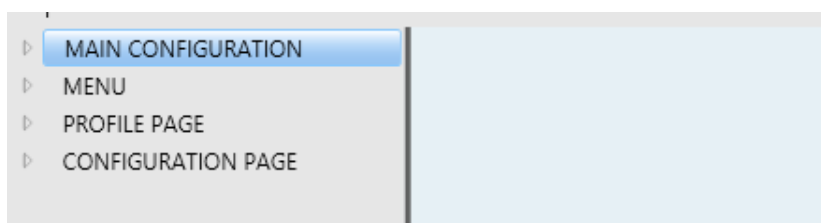


Figure 13 Parameter Screen shown by default

As shown in Figure 13, the parameter window is initially divided into four main tabs:

- **Main Configuration**, which contains a few more tabs:
 - General,
 - Security,
 - Touch lock,
 - Internal Temperature Sensor,
 - Energy Monitor Objects,
 - White Channel Objects.
 - Firmware Update.

- **Menu**, which contains one more screen by default:
 - Configuration, from which it is possible to activate and configure each of the six general-purpose pages of the device (see section 2.2).

- **Profile Page**, which itself will contain one more screen:
 - Configuration, from which it is possible to activate and configure the Profile page (see section 2.3).

- **Configuration Page**, which itself will contain one more screen:
 - Configuration, from which it will be possible to activate and configure the Configuration page (see section 2.4).

The following sections will detail the parameterisation process of each of the above tabs, including the initially hidden screens, which will become visible depending on the selected options.

3.2 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.

3.2.1 GENERAL

Power Supply Voltage	12 V.
Note: this parameter is only used for adjusting the temperature measured by the internal temperature probe	
Default Theme (after programming)	Ocean
Screen Illumination	Turn off the screen after a long time of inactivity
Show Time	Yes
Show Temperature	Internal Temperature Probe
Time/Date Sending Period	10 x 1 min. (0=Disabled)
Thermostat 1	<input type="checkbox"/>
Thermostat 2	<input type="checkbox"/>
Inputs	<input type="checkbox"/>
WEEKDAYS Initials (Mon...Sun) [e.g. MTWTFSS]	MTWTFSS
Label for "Time ON"	Time ON
Label for "Time OFF"	Time OFF
Label for "Scene"	Scene

Figure 14 General (Main Configuration)

- The General screen contains the following parameters:
 - Power Supply Voltage:** lets the integrator specify the particular voltage of the external supply powering Z41 Lite. The available values are 12 (by default), 24 and 29 volts. Selecting one option or another will only cause an internal correction over the temperature value measured by the built-in probe.
 - Default Theme (after programming):** permits selecting the user interface colour scheme (out of 8 different options) to be loaded after the ETS download. The final user will have the option to switch the scheme from the Profile page (see section 2.3), unless such option is disabled by parameter.
 - Screen Illumination:** lets defining whether the backlight should behave during inactivity according to section 2.5 (Turn off screen), or if on the contrary the screen should never turn off entirely (Attenuate the screen) or not even attenuate the light level (Max illumination always).

Note: *the default option is encouraged. Permanent illumination may affect adversely the device lifetime, and is only intended for special circumstances.*

- **Show Time:** sets whether the current time (according to the internal clock) should show in the upper left corner of every page. The default is “Yes”.
- **Show Temperature:** 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.
- **Time/Date Sending Period:** when enabled (value different from zero), whenever the device is restarted, the date and time objects (“**[General] Date**” and “**[General] Time**”) will be sent to the bus, and thereafter, they will be sent again every time the period expires. Setting a new time to the device either through object or from the date/time configuration window will not restart the count of the sending period. Range: [0-65535] x 1 min. The default value is 10 min.
- **Thermostat 1:** checkbox (unselected by default) that activates/deactivates the Thermostat 1 parameter screen. See section 3.7.
- **Thermostat 2:** idem for Thermostat 2. See section 3.7.
- **Weekdays Initials (Mon... Sun):** field that permits customising the label that represents each of the weekdays within the configuration pages of the weekly timers (see epigraph e) in section 3.4.2.5). A 7-character string (including letters or numbers) should be entered, so each of the characters will represent one weekday. The default value is “MTWTFSS”, according to the initial letters of the day names in English.
- **Label for “Time ON”:** field that permits customising the label that will be shown within the configuration pages of the weekly timers (see epigraph e) in section 3.4.2.5), next to the checkbox that enables/disables the sending of the switch-on order. The default value is “Time ON”.
- **Label for “Time OFF”:** field that permits customising the label that will be shown within the configuration pages of the weekly timers (see epigraph e) in

section 3.4.2.5), next to the checkbox that enables/disables the sending of the switch-off order. The default value is “Time OFF”.

- **Label for “Scene”**: field that permits customising the label that will be shown within the configuration pages of the weekly timers (see epigraph e) in section 3.4.2.5), next to the checkbox that enables/disables the sending of scenes. The default value is Scene”.

3.2.2 SECURITY

Figure 15 Security (Main Configuration)

This screen permits selecting how many security levels (one or two) will be available for the configuration of the access to the control pages. In other words, it permits setting one or two different passwords, so the integrator can afterwards configure whether the access to each page will be protected by one password or another, or remain unprotected – every page can be independently configured. Figure 16 shows the “enter password” dialog shown to the user when trying to access a protected page.

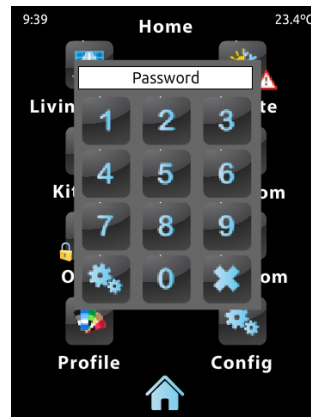


Figure 16 Password Insertion Dialog (for password-protected pages)

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.

The parameters on the Security screen are:

- **Security Levels:** dropdown list for selecting whether one (default option) or two security levels will be available. Depending on the selection, the parameter below (Password) may show once or twice.

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

- **Password:** parameter made of four additional textboxes, each of which should contain one of the four consecutive digits (0-9) that will compose the password. The default password is "1234".

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. The default password for Level 2 is "5678".

Security Levels: Two Levels

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 17 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 Levels:** 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. Up to 15 characters are permitted per label.
 - **Label for “Password 1”:** message shown when the user is required to type in the password for level 1. By default, “Password 1”.
 - **Label for “Password 2”:** message shown when the user is required to type in the password for level 2. By default, “Password 2”.
 - **Label for “Error”:** message shown to the user when the typed password is not valid. By default, “ERROR”.
 - **Label for “New password”:** message shown to ask the user for a new password, during the password change process. By default, “New password”.

- **Label for “Repeat password”:** message shown when the user is required to re-type the new password. By default, “Repeat password”.
- **Label for “Updated”:** message shown to the user as a confirmation of the password change. By default, “Updated”.

Buttons that lead to a protected page from the Menu page (see section 2.1) will show a little lock icon overlaid on their lower left corner.

3.2.3 TOUCH LOCK

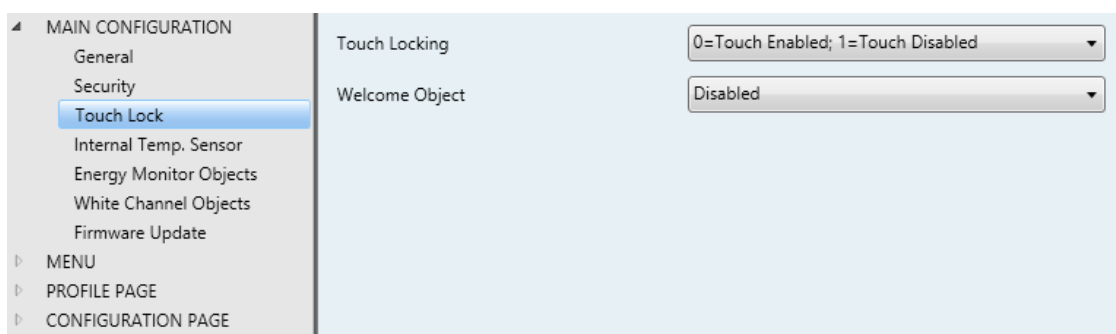


Figure 18 Touch Lock (Main Configuration)

This screen is provided for the configuration of the “welcome” and “touch lock” objects. By default two parameters are shown:

- **Touch Locking:** dropdown list with the following options: “0=Touch enabled; 1=Touch disabled” (default value) and “0=Touch disabled; 1=Touch enabled”. Depending on the selection, sending one value or another (“0” or “1”) to object **[General] Touch lock** will lock/unlock the touch panel. The detailed behaviour of this object is described in section 3.1.
- **Welcome Object:** dropdown list with the following options: “Disabled” (default value), “1-bit object” and “Scene”.
 - If “Disabled” is selected, this functionality will remain inactive. Therefore, nothing will happen when, after a certain standby period, the user resumes the interaction with the touch panel.
 - If “1-bit object” is selected, the **[General] Welcome Object** binary object is enabled and two more parameters turn visible:
 - **Value:** value (“0” or “1”) that will be sent to the bus. By default it is “1”.

- **Time to Activate the Welcome Object:** minimum time of inactivity (in minutes) before a new press on the touch panel triggers the sending of the welcome object. By default, it is 1 minute.

Touch Locking	0=Touch Enabled; 1=Touch Disabled
Welcome Object	1-bit object
Value	1
Time to activate the welcome object	1 x 1min.

Figure 19 Touch Lock (Time to activate the welcome object configuration – 1 bit object)

- If “Scene” is selected, the **[General] Scenes: Send** object, already enabled in the beginning, will work as the welcome object. In this case, the following two parameters come up:

- **Scene Number:** number of the scene (1-64) that will be sent to the bus. The default value is “1”.
- **Time to Activate the Welcome Object:** minimum time of inactivity (in minutes) before a new press on the touch panel triggers the sending of the parameterised scene. By default, it is 1 minute.

Touch Locking	0=Touch Enabled; 1=Touch Disabled
Welcome Object	Scene
Scene Number	1
Time to activate the welcome object	1 x 1min.

Figure 20 Touch Lock (Time to activate the welcome object configuration - Scene)

3.2.4 INTERNAL TEMPERATURE SENSOR

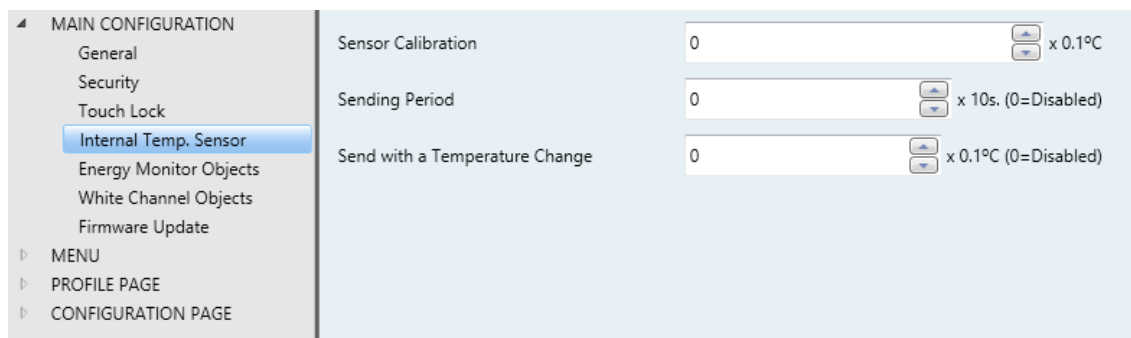


Figure 21 Internal Temperature Sensor (Main Configuration)

This screen permits configuring the internal temperature sensor of the device. Three parameters are provided for this:

- **Sensor Calibration:** permits setting a certain correction over the values thrown by the internal temperature probe. This way, if the value of the measurement is found to be slightly over (or below) the actual temperature due to external factors, it is possible to set an adjustment between -5.0°C and 5.0°C (0.0°C by default), in steps of 0.1°C . The value set here may be optionally changed, in runtime, from the Configuration page (see section 2.4).
- **Sending Period:** cycle time (in tens of a second) for sending the value of the measured temperature to the bus. This sending is made through the **[General] Internal temperature** object (see section 3.1), enabled by default. The permitted values are 0 to 100 tens of a second. The value “0” (set by default) deactivates this sending to the bus.
- **Send with a Temperature Change:** permits performing a specific sending of the temperature value to the bus –through the **[General] Internal Temperature** object– whenever an increase or decrease greater or equal than a certain value is detected between two consecutive measurements, no matter if a periodic sending has been enabled or not. Such value (unsigned) must be entered here, in terms of tenths of a degree. To avoid having this extra sending on temperature changes, simply leave the default value (“0”).

3.2.5 ENERGY MONITOR OBJECTS

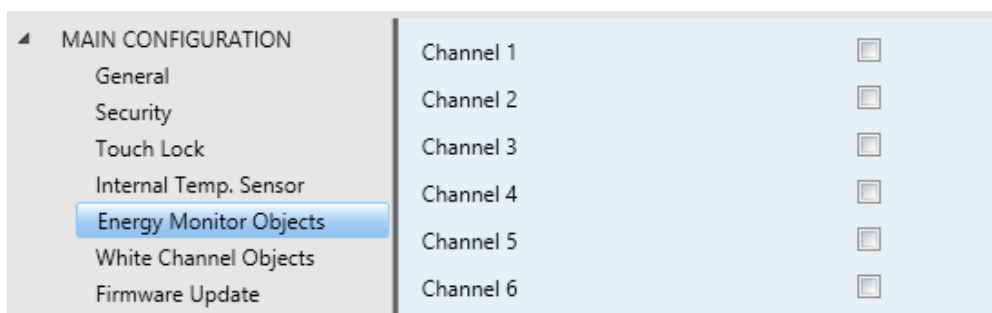


Figure 22 Energy Monitor Objects (Main Configuration)

This screen contains six checkboxes, each of which allows enabling or disabling one communication “channel” that may be used for bus reception of energy consumption data.

Every channel implements four communication objects:

- **[Energy x] Power:** 2-byte object to be linked to external objects that send power values, such as the power measuring objects implemented by **KES**, the energy saver device from Zennio.
- **[Energy x] Energy:** 4-byte object to be linked to external objects that send energy values, such as the energy measuring objects implemented by **KES**, the energy saver device from Zennio.
- **[Energy x] CO2:** 2-byte object to be linked to external objects that reflect CO₂ volumes, such as those implemented in **KES**, the energy saving device from Zennio, which represent in floating-point format the kilograms of CO₂ emitted to the atmosphere due to a certain energetic consumption.
- **[Energy x] Cost:** 2-byte object to be linked to external objects that express, as floating point values, the cost of the energy consumption, in terms of the local currency. For example, the cost estimation objects implemented by **KES**, the energy saver device from Zennio.

So, when any of the six channels is enabled, the four communication objects corresponding to that channel will show up.

For further information about these objects and about power consumption monitoring in Z41 Lite, please refer to epigraph c) in section 3.4.2.5.

3.2.6 WHITE-CHANNEL OBJECTS

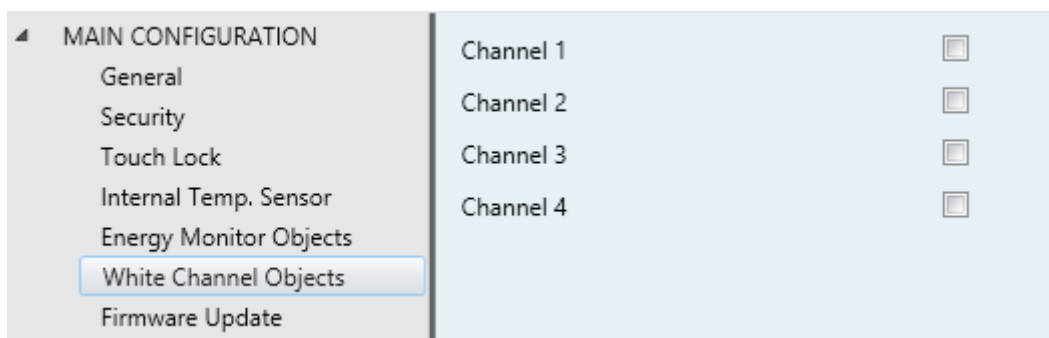


Figure 23 White-Channel Objects (Main Configuration)

This screen contains four checkboxes that allow the integrator enable or disable up to four 1-byte communication objects independently, named **[RGBW] White Channel x**, through which regulation orders (expressed as percentage values) may be sent to the white colour channels of one or more RGBW light dimmers.

For further information about the usage of these objects and about the per-channel dimming controls implemented by Z41 Lite, please refer to epigraph b) in section 3.4.2.5.

3.2.7 FIRMWARE UPDATE

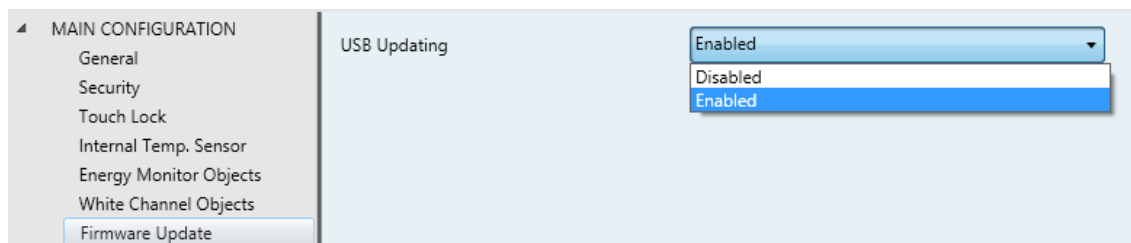


Figure 24 Firmware Update (Main Configuration)

This tab lets the integrator enable or disable the device firmware update function (see section 1.3.1) through the USB port. The only parameter is:

- **USB Updating:** “Enabled” (default option) or “Disabled”.

Note: *Z41 Lite does not provide an Ethernet interface for firmware update purposes.*

3.3 MENU

The Menu tab contains only one screen, Configuration.

3.3.1 CONFIGURATION

MAIN CONFIGURATION	Title	<input type="text"/>
MENU	Automatic page shaping	No
Configuration	Default Page	Menu
PROFILE PAGE	Page 1	<input type="checkbox"/>
CONFIGURATION PAGE	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"/>
	Profile Page	<input checked="" type="checkbox"/>
	Configuration Page	<input checked="" type="checkbox"/>

Figure 25 Configuration (Menu)

The Configuration screen, under the Menu tab, permits configuring the Menu page itself, as well as a set of general options related to the enabled pages of controls that can be accessed from the Menu page.

The parameters available are:

- **Title:** text field that defines the title that will be shown on the top of the Menu page.
- **Automatic Page Shaping:** dropdown list that allows choosing whether the boxes in the Menu page should be automatically distributed (option “Yes”) depending on the number of boxes configured, or be displayed as a static 2x4 grid (option “No”). See section 2.6.
- **Default Page:** 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 6), as well as two more checkboxes for the Profile and Configuration pages. 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. The next sections of this manual describe these tabs in detail.

3.4 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.

3.4.1 CONFIGURATION

MAIN CONFIGURATION	Title	<input type="text"/>
MENU	Automatic page shaping	No
PAGE 1	Icon	Home
Configuration	Protected	No
PROFILE PAGE	Box 1	<input type="checkbox"/>
CONFIGURATION PAGE	Box 2	<input type="checkbox"/>
	Box 3	<input type="checkbox"/>
	Box 4	<input type="checkbox"/>
	Box 5	<input type="checkbox"/>
	Box 6	<input type="checkbox"/>
	Box 7	<input type="checkbox"/>
	Box 8	<input type="checkbox"/>

Figure 26 Configuration (Page *n*)

This screen contains the following parameters:

- **Title:** text field that sets the title shown on the top of the Menu page.

- **Automatic Page Shaping:** dropdown list that allows choosing whether the boxes in the Menu page should be automatically distributed (option “Yes”) depending on the number of boxes configured, or be displayed as a static 2x4 grid (option “No”). See section 2.6.

- **Icon:** sets the icon that will represent the page in the Menu page.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

- **Protected:** dropdown list that allows setting whether the access to the page will be protected by password or not. Depending on the security levels configured (one or two; see section 3.2.2) 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, eight checkboxes are provided to let the integrator enable or disable each of the eight 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.

3.4.2 BOX *i*

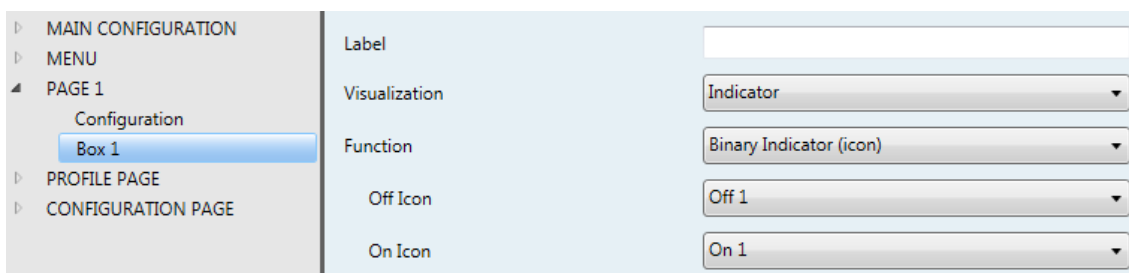


Figure 27 Box *i* (Page *n*)

This screen contains the following parameters:

- **Label:** identifying title for the box.
- **Visualization:** box format, which may be: Indicator (default option; the box will simply 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) or Other (the box will implement some other functionality).

Depending on the Visualization type selected, the parameters below will change. The following sections explain the available parameters depending on the visualization type selected.

3.4.2.1 INDICATOR



Figure 28 Indicator (Box)

Boxes configured as indicators permit implementing a set of different functions. The desired function must be selected through the **Function** parameter. The available functions and the related parameters are:

a) 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**).

Visualization	Indicator
Function	Binary Indicator (icon)
Off Icon	Off 1
On Icon	On 1

Figure 29 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.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

b) 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**).

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

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

Figure 30 Binary Indicator (Text)

c) 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, instead of only values 0 and 1.

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. This allows setting which icon will be shown in the box upon the reception of which value through the communication object.

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

Figure 31 Enumerated Indicator (Icon)

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.

d) 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, instead only by values 0 and 1. 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**) 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.

Figure 32 Enumerated Indicator (Text)

e) Numerical Indicators

Figure 33 1-Byte Indicator (Unsigned Integer)

The remaining functions under Indicator are shown in Table 1. In all of these cases, the box will behave as a numerical state indicator that permanently displays the value of a certain communication object, which is enabled when the function is assigned to the box.

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
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 (floating point)	-671088.64 – 670760.96	[Pn][Bi] 2-byte float indicator

Table 1 Numerical indicators

The table contains the allowed value ranges and the name of the object for each case.

In the particular case of the 2-byte floating point indicator, the integrator will be shown a text field (**Float Unit**), empty by default, that permits specifying the measuring units (up to 6 characters) of the displayed value, for instance “°C”. Moreover, for this particular case, the device will dynamically adjust the number of the decimal figures shown depending on the order of magnitude of the

current value: two for values greater than 0 and lower than 0.1; one for values between 0.1 and 100; and none for other cases (and analogously for negative values).

All these numerical indicators incorporate an additional parameter, **Save Log?**. In case of enabling this option (“Yes”), any changes in the value of the indicator will be continuously saved to an internal log file, so the final user can be afterwards offered a visual representation that reflects the chronological evolution of such value. Boxes for which this option has been enabled will show a small icon overlaid on their upper left corner.

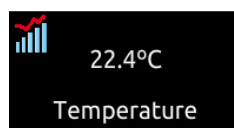


Figure 34 Indicator implementing the Log Function

Boxes showing this icon let the user press them, which will bring up a pop-up window similar to Figure 35.

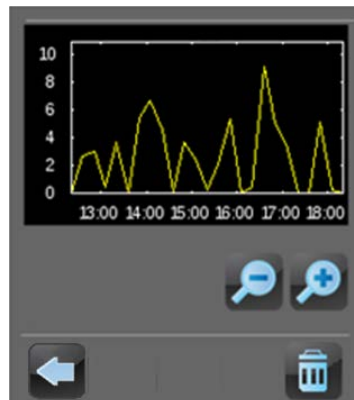


Figure 35 Graphical representation of the Log File

The graph (where the vertical scale will dynamically adapt to the values being represented) shows the chronological evolution of the value, with higher or lower detail (hourly, daily, weekly, monthly; see Figure 36) depending on the zoom level set by the user by means of the on-screen magnifying glass control. In the case of the lower detail levels, the graph will show the average values per time unit (day or month) as well as the maximum and minimum values.

Finally, the bin icon (lower right corner of the screen) lets the user, by performing a long press, delete the contents of the log file, thus setting the values of the monitor to zero.

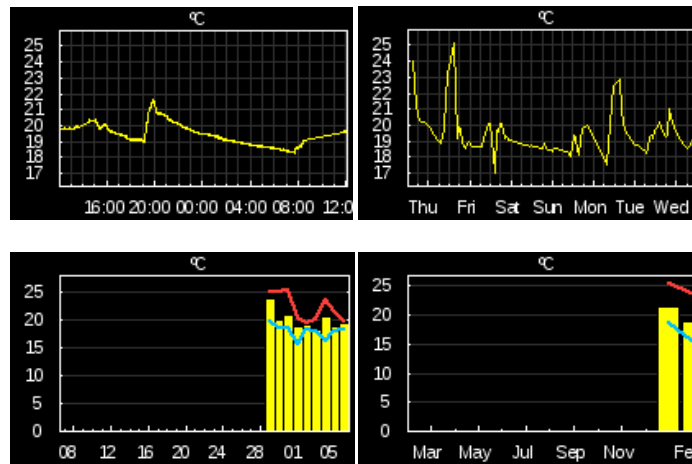


Figure 36 Different Detail Levels (log file)

3.4.2.2 1-BUTTON CONTROL

Boxes configured as 1-button controls show one centred button and a title.

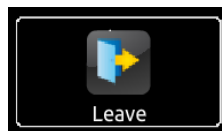


Figure 37 1-Button Control (Box)

The desired function must be selected through the **Function** parameter. The available functions and the parameters related to each are:

a) Binary Control

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.

Visualization	1-button Control
Function	Binary Control
Action	Toggle
Button (In the middle)	Off 1

Figure 38 Binary control

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:

- **Toggle:** pressing the button will cause an alternate sending of the values “1” and “0”. Particularly, on the first press one “1” will be sent; on the second, one “0”; on the third, one “1”; and so on.
- **0:** one “0” will be sent whenever the button is pressed.
- **1:** one “1” will be sent whenever the button is pressed.
- **Short 1 – Long 0:** short presses will cause the sending of the value “1”, while long presses will cause the sending of the value “0”.
- **Short 0 – Long 1:** inverse to the previous case.
- **Hold 0 – Release 1:** one “0” will be sent when the button press starts, and one “1” as soon as the button is released.
- **Hold 1 – Release 0:** one “1” will be sent when the button press starts, and one “0” as soon as the button is released.

Finally, the dropdown list next to **Button (in the middle)** permits selecting the icon to be displayed inside the button.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

b) Numerical Constant Controls

If 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**.

The screenshot shows a configuration panel with the following settings:

Visualization	1-button Control
Function	1-byte Constant (unsigned int)
Constant Value	0
Button (In the middle)	Scene

Figure 39 1-byte constant (unsigned int)

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	Range	Object
1-byte constant (unsigned int.)	0 – 255	[Pn][Bi] 1-byte unsigned int control
1-byte constant (signed int.)	-128 – 127	[Pn][Bi] 1-byte signed int control
Constant percentage value	0% – 100%	[Pn][Bi] Percentage control
2-byte constant (unsigned int.)	0 – 65535	[Pn][Bi] 2-byte unsigned int control
2-byte constant (signed int.)	-32768 – 32767	[Pn][Bi] 2-byte signed int control
2-byte constant (float)	-671088.64 – 670760.96	[Pn][Bi] 2-byte float control

Table 2 Numerical constant controls

Finally, the dropdown list next to **Button (in the middle)** permits selecting an icon to be displayed inside the button.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

c) Scene Control

In this case, the central button of the box will react to user presses by sending a scene value to the KNX bus through the **[General] Scenes: send** object, which is enabled by default. Parameter **Scene number** permits specifying the number of the scene (1 – 64) to be sent. Moreover, **Action** (which shows the options “Run” and “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.

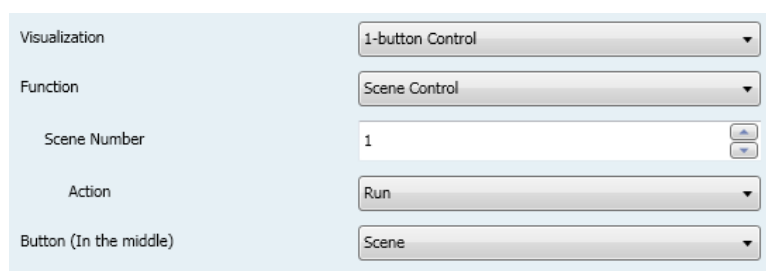


Figure 40 Scene control

Finally, the dropdown list next to **Button (in the middle)** permits selecting an icon to be displayed inside the button.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

3.4.2.3 2-BUTTON CONTROL



Figure 41 2-button control (Box)

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. This category also includes some specific controls such as **light dimming** and **shutter control**.

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.

Notes:

- A list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.
- *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:

a) Binary Control (Icon)

Visualization	2-button Control
Function	Binary Control (icon)
Action	Left 0, Right 1
Off Icon	Off 1
On Icon	On 1
Left Button	Off 1
Right Button	On 1

Figure 42 Binary Control (Icon)

When the user touches the buttons, a binary value will be sent through the **[Pn][Bi] Binary control** object, while the status object, **[Pn][Bi] Binary indicator**, will determine the icon shown in the box.

The parameters available are:

- **Action:** sets the value to be sent on button touches. This can be "Left 0, Right 1" (by default) or "Left 1, Right 0".
- **Off Icon** and **On Icon:** dropdown lists for the selection of the icons that will represent the two states (0 / 1) of the **[Pn][Bi] Binary Indicator** object, which gets automatically updated whenever an order is sent through the control object. It may also receive values from the bus.

b) Binary Control (Text)

Visualization	2-button Control
Function	Binary Control (text)
Action	Left 0, Right 1
Off Text	
On Text	
Left Button	Off 1
Right Button	On 1

Figure 43 Binary Control (Text)

When the user touches the buttons, a binary value will be sent through the **[Pn][Bi] Binary Control** object, while the status object, **[Pn][Bi] Binary Indicator**, will determine the text that will be shown in the box.

The parameters available are:

- **Action:** sets the value to be sent on button touches. This can be "Left 0, Right 1" (default) or "Left 1, Right 0".
- **Text Off** and **Text On:** indicative texts that will be shown in the box depending on the state (0 / 1) of the **[Pn][Bi] Binary Indicator** object, which gets automatically updated whenever an order is sent through the control object. It may also receive values from the bus.

c) Enumerated Control (Icon)

The box will behave analogously to the case of the Binary control (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, as the control (**[Pn][Bi] Enumerated Control**) and the status (**[Pn][Bi] Enumerated Indicator**) objects are not binary but 1-byte objects.

Visualization	2-button Control
Function	Enumerated Control (icon)
# Enums	1
Value 1	0
Icon 1	Off 1
Left Button	Arrow Left
Right Button	Arrow Right

Figure 44 Enumerated Control (Icon)

The parameters available are:

- **# Enums**: number of states (up to 6) that will be distinguished. For every distinguished state, the parameters **Value** and **Icon** will become available, in order to relate every significant value of the communication object to a certain icon that will represent it.
- **Value *j***: numerical value (0 – 255) that will be sent through the control object when the user, after pressing the buttons, sets the control box to state *j*.
- **Icon *j***: indicative icon that will be displayed in the box when the status object (which gets automatically updated whenever a control order is sent, although it may also receive values from the bus) acquires the numerical value defined in the above parameter.

d) Enumerated Control (Text)

The box will behave analogously to the case of the Binary control (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, as the control (**[Pn][Bi] Enumerated Control**) and the status (**[Pn][Bi] Enumerated Indicator**) objects are not binary but 1-byte objects.

Visualization	2-button Control
Function	Enumerated Control (text)
# Enums	1
Value 1	0
Text 1	
Left Button	Arrow Left
Right Button	Arrow Right

Figure 45 Enumerated Control (Text)

The parameters available are:

- **# Enums**: number of states (up to 6) that will be distinguished. For every distinguished state, the parameters **Value** and **Icon** will become available, in order to relate every significant value of the communication object to a certain icon that will represent it.
- **Value *j***: value (0 – 255) that will be sent through the control object when the user, after pressing the buttons, sets the control box to state *j*.
- **Text *j***: indicative text that will be displayed in the box when the status object (which gets automatically updated whenever a control order is sent, although it may also receive values from the bus) acquires the numerical value defined in the above parameter.

e) Numerical Controls

If the box is assigned any of the six numerical functions, user touches over the buttons will trigger the sending of a certain numerical value to the bus (which 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 (which gets automatically updated after a control order is sent, although it may also receive values from the bus).

Table 3 shows, for each of the six functions, the permitted value range and the name of the control and status objects

Range	Control object	Status object
1-byte (unsigned int.)		
0 – 255	[Pn][Bi] 1-byte unsigned int control	[Pn][Bi] 1-byte unsigned int indicator
1-byte (signed int.)		
-128 – 127	[Pn][Bi] 1-byte signed int control	[Pn][Bi] 1-byte signed int indicator
Percentage control		
0% – 100%	[Pn][Bi] Percentage control	[Pn][Bi] Percentage indicator
2-byte (unsigned int.)		
0 – 65535	[Pn][Bi] 2-byte unsigned int control	[Pn][Bi] 2-byte unsigned int indicator
2-byte (signed int.)		
-32768 – 32767	[Pn][Bi] 2-byte signed int control	[Pn][Bi] 2-byte signed int indicator
2-byte (float)		
-671088.64 – 670760.96	[Pn][Bi] 2-byte float control	[Pn][Bi] 2-byte float indicator

Table 3 Numerical Controls

For all the six functions, the parameters available are:

Figure 46 1-Byte (Unsigned Int.) 2-Button Control

- **Action:** sets which of the two buttons will increase the current numerical value and which will decrease it on user presses. The allowed options are “Left Decrease, Right Increase” (default) and “Left increase, Right Decrease”.

- **Initial Value:** sets which value from the permitted range will be assumed by the control prior to any button presses. It is always 0 by default.
- **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.
- **Short Increment:** sets the increase or decrease to be applied to the current value on every short press over the increase or decrease buttons, respectively (i.e., the smaller the increment is, the more presses will be required). It is 1 by default, except for the case of the 2-byte floating point control, where it is 0.5.
- **Long Increment:** sets the increase or decrease to be applied to the current value on every long press over the increase or decrease buttons, respectively (i.e., the smaller the increment is, the more presses will be required). It is 10 by default, except for the case of the 2-byte floating point control, where it is 1.

f) Two-button Scene Control

Label	<input type="text"/>
Visualization	2-button Control
Function	Two-button Scene Control
Left Scene	1
Action	Run and Save
Button Icon	Scene
Right Scene	1
Action	Run and Save
Button Icon	Scene

Figure 47 2-Button Scene Control

Analogously to the 1-button scene control, 2-button scene controls allows sending the KNX bus a scene value through the **[General] Scenes: send**

object, however in this case there will be two buttons in the box, each of which can be configured independently.

Through the **Scene** parameter it is possible to assign each of the two buttons (left / right) the number of the scene to be sent (1 – 64). In addition, **Action** (which permits the options “Run” and “Run and save”) lets the integrator set whether the device will only send execution orders for the specified scene (after a short press), or it will possible, apart from sending execution orders on short presses, to also send save orders for such scene, in the case of performing a long press on the button

Last, the dropdown list under **Button Icon** sets the icon to be displayed by the buttons.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

g) Shutter Control

Shutter controls permit sending move up, move down, stop or step shutter orders to the KNX 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.

Visualization	2-button Control
Function	Shutter Control
Action	Left Down, Right Up
Left Button	Shutter Down 1
Right Button	Shutter Up 1

Figure 48 Shutter Control

The only specific parameter for this function is:

- **Action:** sets which of the two buttons will send the move up orders and which the move down orders. Options are “Left Down, Right Up” (default) and “Left Up, Right Down”.

Setting up a box as a shutter control enables three communication objects, **[Pn][Bi] Shutter Position** (1 byte), **[Pn][Bi] Move Shutter** and **[Pn][Bi] Stop**

Shutter (both, binary objects). The first one acts as the status object, which determines the value shown in the box; it will be necessary to link it to the analogous object from the shutter actuator (as it does not get automatically updated on user presses) so it receives the value 100% when the shutter is completely down, and the value 0% when it is totally up. On their side, control orders will be sent through one of the two remaining objects:

- Orders to move the shutter up/down (completely) are sent through **[Pn][Bi] Move Shutter** after a long press on the corresponding button. Value “0” represents the move up order, while “1” represents the move down order.
- Orders to perform a step movement (value “0” for steps up and “1” for steps down) are sent through **[Pn][Bi] Stop Shutter** after a short press on the corresponding button. If the shutter was already in movement, these orders are interpreted by the actuator as a stop order.

h) Light Dimming

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	Light Dimming
Action	Left Off, Right On
Dimming Step	100%
Left Button	Light Off 1
Right Button	Light On 1

Figure 49 Light Dimming

The parameters available are:

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

Table 4 shows the number of steps (that is, of long presses) required for a complete light regulation (from totally off to totally on, or vice versa) depending on the value parameterised as the dimming step.

Dimming Step	Button Presses Required for Complete Dimming (0 – 100%)
(1) 100%	1
(2) 50%	2
(3) 25%	4
(4) 12,5%	8
(5) 6,25%	16
(6) 3,1%	32
(7) 1,5%	64

Table 4 Dimming Steps

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 Z41 Lite 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.

3.4.2.4 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 as follows:

a) Setpoint Control

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: **[Pn][Bi] Temperature Control** for the control and **[Pn][Bi] Temperature Indicator** for the status. In addition, the box itself will permanently reflect the value (in °C) 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.

Visualization	Climate Control
Function	Setpoint Control
Action	Left Decrease, Right Increase
Setpoint Type	Absolute
Minimum Value	10 x 1°C
Maximum Value	30 x 1°C
Left Button	Temp Decrease
Right Button	Temp Increase

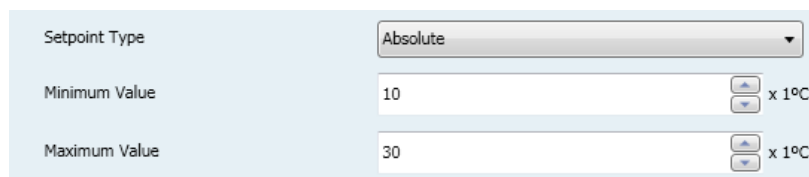
Figure 50 Setpoint

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.

The parameters available are:

- **Action:** sets which of the two buttons will permit increasing the setpoint value and which one will permit decreasing it. Options are “Left decrease, Right increase” (default) and “Left increase, Right decrease”.
- **Setpoint Type:** it may be “Absolute” (default), “Relative (1-bit object)” or “Relative (Float Object)”.

If the first is selected, the control object will be a 2-byte floating point object and its value will correspond to the actual setpoint value to be sent to the thermostat, being progressively higher or lower as the user touches the control buttons. On the other hand, the permitted temperature setpoints will be delimited by a **maximum value** and a **minimum value**, defined by the so-named parameters.



Setpoint Type	Absolute
Minimum Value	10 x 1°C
Maximum Value	30 x 1°C

Figure 51 Absolute Setpoint Control

If the second option is selected (1-bit object setpoint control), the control object will be binary, and it will throw a “0” to the bus whenever the user asks for a setpoint decrement, and a “1” when the user asks for an increment. How much it is incremented or decreased after every order is defined through the **offset** parameter (which accepts values between 1 and 255 tenths of a degree), thus permitting a consistent self-update of the label in the box after every user press, although the status object will also admit the reception of feedback from the bus, i.e., the updated absolute setpoint value sent from the external thermostat.

Setpoint Type: Relative (1-bit object)

Offset: 5 x 0.1°C

Figure 52 Relative (1 Bit) Setpoint Control

Finally, when “Relative (float object)” is selected, the control will send 2-byte floating point values. However, these values will correspond to the particular increment or decrement (or offset) to be applied each time. This offset refers to a base temperature and will always be a multiple of 0.5°C. As the user touches the increase/decrease buttons in the box, this offset will vary, always within a certain range defined through the **Minimum Offset** and **Maximum Offset** parameters.

Setpoint Type: Relative (float object)

Minimum Offset: -10 x 1°C

Maximum Offset: 10 x 1°C

Figure 53 Relative (Float Object) Setpoint Control

Example: suppose the following parameterisation:

- Relative setpoint with a floating point object,
- Minimum offset = -10°C,
- Maximum offset = 10°C.

The objects are linked to those from an external thermostat, which can be supposed to have a base temperature setpoint of 25°C. After the parameter download:

1. The first touch on the increase button will trigger the sending of the value “0.5”, so that the thermostat sets the setpoint to **25.5°C**. This value will be then sent through the status object, and afterwards shown by Z41 Lite in the corresponding box.
2. A second touch will trigger the value “1”, after which the setpoint will become **26°C**.
3. On the third touch, the value sent will be “1.5”, so the setpoint will be set to **26.5°C**.

And so on until the offset sent becomes 10 (setpoint of **35°C**). Any further presses on the increase button will send the value “10” again, so the setpoint will remain at **35°C**, while touching the decrease button for the first time will trigger the sending of the value “9.5”, setting the temperature back to **34.5°C**. On the second button press, the value

“9” will be sent, and the setpoint will become **34°C**. And so on until an offset of “-10” (setpoint of **15°C**) is reached. From that moment, any further presses on the decrease button will re-send the value “-10” and the setpoint will remain **at 15°C**.

Important: the control object for the floating point relative setpoint control has its Write flag enabled with the intention of permitting the reception of feedback from the thermostat, which may reset or modify the status of the offset for any reason at any time. This feedback allows that Z41 Lite updates its own value to equal that from the external thermostat. Thinking of the above example, if Z41 Lite receives the value “0” after having the user performed three presses on the increase button (therefore, after having been sent the value “1.5” to the bus), it will happen that on the forth press the value sent will be “0.5” and not “2”.

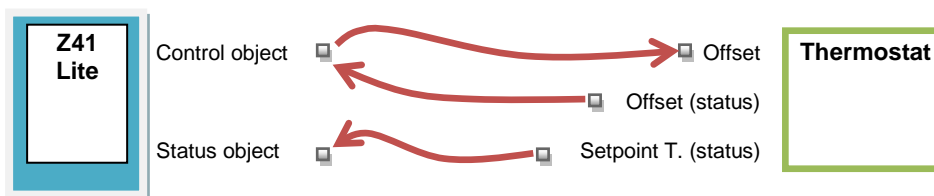


Figure 544 Relative (Float) Setpoint Control. Object Linking.

Note: the parameterised setpoint type becomes transparent to the final user, as the aspect of the on-screen box remains the same in all cases. In addition, the setpoint status object is always a 2-byte floating point object and will represent, in any case, the absolute value of the current temperature setpoint.

- **Left Button** and **Right Button:** permit selecting the desired icon to be displayed inside each button in the box.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

b) Mode Control

This function turns the box into a climate mode control.

Visualization	Climate Control
Function	Mode Control
Mode Type	Heat/Cool
Left Button	Mode Cool
Right Button	Mode Heat

Figure 55 Heat/Cool Mode Type

Two mode control types are possible: the **Heat / Cool** control type and the **Extended** (HVAC) control type. Selecting the former or the latter is possible through the **Mode Type** 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.



Figure 56 Heat/cool Mode (Box)

Two communication objects are also enabled: the **[Pn][Bi] Mode Control** control object and the **[Pn][Bi] Mode Indicator** status object. When the user activates the Cool mode, the device will send the value “0” through the control object, while on the activation of the Heat mode the value “1” will be sent. Moreover, the box will automatically commute between one icon and the other every time the user executes a mode change, and also on the reception of a new value (from the bus) through the status object.

➤ **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.



Figure 57 Extended Mode (Box)

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

Figure 58 Extended Mode

When this type of mode control is assigned to the box, two 1-byte communication objects are enabled: the **[Pn][Bi] Mode Control** control object, and the **[Pn][Bi] Mode Indicator** 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.

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

Table 5 HVAC Mode vs. Object Value

Finally, the **Left Button** and **Right Button** parameters permit selecting the icon that will be displayed inside the buttons of the box.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

c) Fan Control

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

Visualization	Climate Control
Function	Fan Control
Action	Left Decrease, Right Increase
Fan Type	Min/Max
Left Button	Minus
Right Button	Plus

Figure 59 Fan Control

When this function is assigned to the box, a binary control object (**[Pn][Bi] Fan Control**) and a 1-byte status object (**[Pn][Bi] Fan Indicator**) are enabled. The former will be sending the value “1” to the bus when the user touches the increase button, while the value “0” will be sent when the decrease button is touched. On the other hand, 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.

The parameters available are:

- **Action:** permits setting the increase or decrease actions to one button or another. Options are: “Left: decrease, Right increase” (default) and “Left increase, Right decrease”.
- **Fan Type:** defines the number of states to be distinguished depending on the value of the status object, so more or less different icons will be used to represent the fan level in the box.
 - Min / Max: the first state (**minimum**) will correspond to any value within the range 0% to 49%, both included; the second state (**maximum**) will correspond to all values greater than 49%.

- Min / Med / Max: levels between 0% and 33%, both included, correspond to the **minimum** state; values between 34% and 66%, to the **medium** state; and values over 67% to the **maximum** state.
 - Auto / Min / Med / Max: a level of 0% will be interpreted as the **Auto** state; levels between 1% and 33%, both included, correspond to the **minimum state**; levels between 34% and 66% will be considered as the **medium** state; and a value of 67% or more will correspond to the **maximum** state.
 - Off / Min / Med / Max: 0% will be associated to the **off** fan state; between 1% and 33%, both included, to the **minimum** state; values between 34% and 66% will be interpreted as the **medium** state; and 67% and over will correspond to the **maximum** state.
- **Left Button** and **Right Button**: permit selecting the icon that will be displayed inside the buttons of the box.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

d) Special Mode Control



Figure 60 Special Mode Control (Box)

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 that will adopt an aspect or another depending on the currently active special mode.

Visualization	Climate Control
Function	Special Mode Control
Allow Protection Mode	No
Left Button	Arrow Left
Right Button	Arrow Right

Figure 61 Special Modes

When this function is assigned to the box, two 1-byte objects (**[Pn][Bi] Special Mode Control** as the control object, and **[Pn][Bi] Special Mode Indicator** as the status object) will be enabled. The former will be sending the bus the value that corresponds to the mode that the user selects by touching the buttons (see Table 6), while the status object (which can receive values from the bus although it gets self-updated as well on button presses) will determine the indicative icon represented in the box at any time. If this object receives an unrecognised value from the bus, no icon will be displayed in the indicator.





Special Mode	Icon	Object Value
Comfort		1 (0x001)
Standby		2 (0x002)
Economy		3 (0x003)
Protection		4 (0x004)

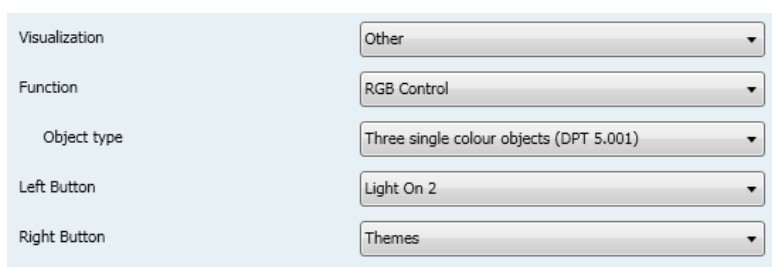
Table 6 Special Modes vs. Icon vs. Object Value

The remaining parameters for this function are:

- **Allow Protection Mode:** this parameter permits excluding (“No”, default option) or including (“Yes”) the “building protection” mode from the special modes listed in the sequential scrolling performed by the buttons.
- **Left Button** and **Right Button:** permit selecting the icon that will be displayed inside the buttons in the box.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

3.4.2.5 OTHER



The image shows a configuration panel with five rows, each with a label on the left and a dropdown menu on the right:

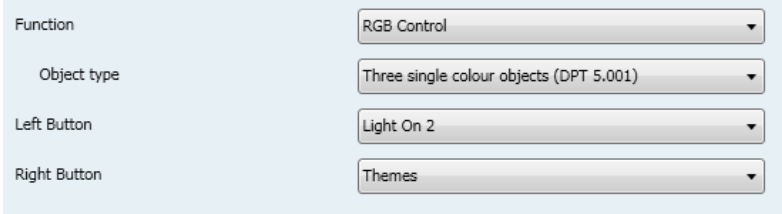
- Visualization: Other
- Function: RGB Control
- Object type: Three single colour objects (DPT 5.001)
- Left Button: Light On 2
- Right Button: Themes

Figure 62 Other

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

a) **RGB**

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



Function	RGB Control
Object type	Three single colour objects (DPT 5.001)
Left Button	Light On 2
Right Button	Themes

Figure 63 RGB

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

- **Object Type:** permits selecting what type of object will be used for controlling the light level of the RGB channels. It may be “Three single colour objects (DPT 5.001)” (by default) or “One RGB object (DPT 232.600)”. If the former is selected, three 1-byte objects will be enabled (with their Write and Read flags activated, so they can work as control and status objects) named:

- **[Pn][Bi] Red Channel**
- **[Pn][Bi] Green Channel**
- **[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.

- **Left Button** and **Right Button:** permit selecting the icon that will be displayed inside the buttons in the box.

Note: a list with all icons available in Z41 Lite can be found in document “Z41 Lite Icon list”, available at www.zennio.com.

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 64 RGB (Box)

Moreover, two buttons can be found in the box:

- The button on the left, which is an **On/Off** switch, permits commuting between a total absence of light in the three channels, and their respective states previous to the switch-off.
- The button on the right, when pressed, launches a **colour palette** that permits the selection of a light level for every channel, either by touching on any of the pre-set colours, or by separately selecting the level of each channel (R, G, B).



Figure 65 RGB Colour Palette

A set of recently used colours is shown at the bottom of the palette. These colours refer to those recently used within that particular box.

b) **RGBW**

Although totally analogous to the function just described, the RGBW control additionally permits controlling one extra channel specifically for white light, intended for dimmers that implement such function, such as LUMENTO X4 RGBW from Zennio.

Visualization	Other
Function	RGBW Control
Object type	Three single colour objects (DPT 5.001)
RGBW Channel	Channel 1
Enable the selected channel in "Main Config./White Channel Objects" tab	
Left Button	Light On 2
Right Button	Themes

Figure 66 RGBW

Therefore, the functionality is similar to that of the RGB control (including the possibility of controlling the RGB channels both through 1-byte objects or through a sole 3-byte object), however one additional parameter (**RGBW Channel**) is provided to let the integrator select which of the four white channels available will be regulated by the box. The channel selected here must have been specifically enabled from the **White Channel Objects** screen (see section 3.2.6) under the Main configuration tab, which will make the **[RGBW] White Channel x** available for sending dimming orders to the white channel.

Important: *although it is possible to assign the "RGBW control" function to any box from any page, it is only permitted to control as many as four different white channels, as it is not possible to have a different **[RGBW] White Channel x** for each of the boxes. This restriction does not apply to the R, G and B channels from the RGB or RGBW controls: each box will make use of its own communication objects for regulating such channels.*

Finally, RGBW control boxes look similarly to RGB control boxes:



Figure 67 RGBW (Box)

Nevertheless, when the colour palette of an RGBW box is brought up, a forth increase/decrease regulator is shown together with the three regulators shown in the case of the RGB control boxes:

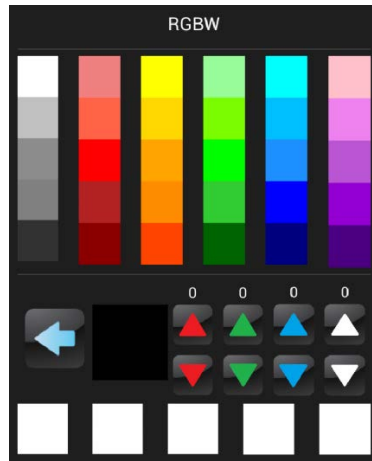


Figure 68 RGBW Colour Palette

c) **Energy Monitor (KES)**

This function permits making use of one of the boxes of Z41 Lite as a monitor of information related to energy consumption –which may be reported by devices such as KES from Zennio– and in particular, as a monitor of the evolution of the instant power.

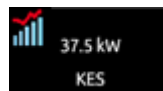


Figure 69 Energy Monitor (KES). Box.

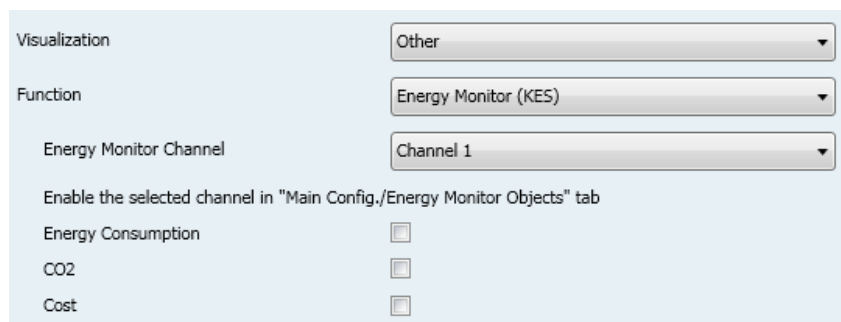


Figure 70 Energy Monitor (KES)

When the user touches any of these boxes, a pop-up window similar to Figure 71 will be shown.

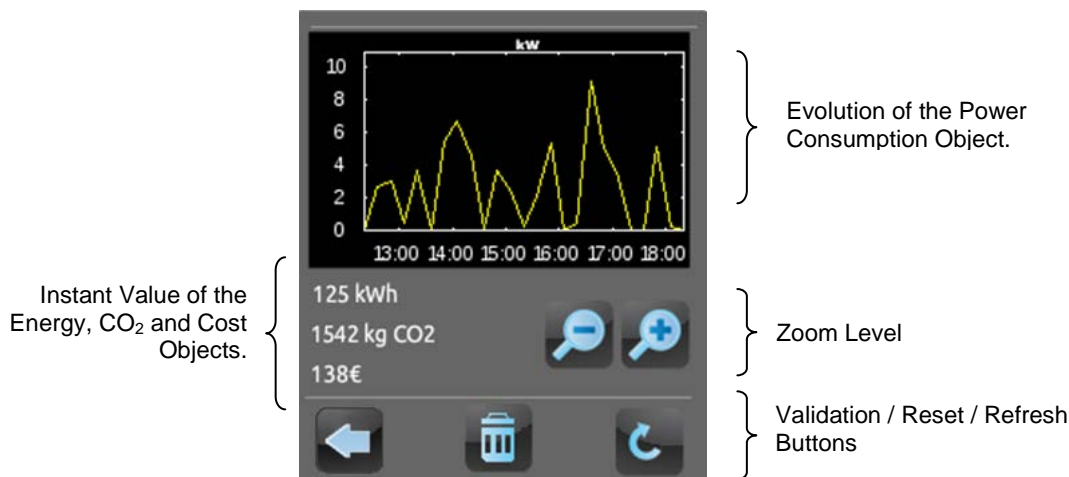


Figure 71 **Energy Monitor (KES). Pop-Up Window.**

Prior to detailing the elements shown in the pop-up window, the ETS parameters for this type of function will be described:

- **Energy Monitor Channel:** dropdown list for the selection of the channel (1 – 6) that will be associated to the box. The channel selected here must have been enabled from the **Energy Monitor Objects** screen (see section 3.2.5) under the Main configuration tab. This way, the box will be monitoring the values received through the **[Energy x] Power**, **[Energy x] Energy**, **[Energy x] CO2** and **[Energy x] Cost** objects, where “x” is the number of the assigned channel.

Important: *although it is possible to assign the “Energy monitor (KES)” function to any box from any page, it is only possible to control as many as six different energy channels, as it is not possible to have an exclusive channel for each of the boxes.*

- **Energy Consumption:** determines whether the pop-up window should show the instant value of the **[Energy x] Energy** object or not.
- **CO2:** determines whether the pop-up window should show the instant value of the **[Energy x] CO2** object or not.
- **Cost:** determines whether the pop-up window should show the instant value of the **[Energy x] Cost** object or not.

When this function is assigned to the box, two 1-bit communication objects show up: **[Pn][Bi] Energy Monitor: Request** and **[Pn][Bi] Energy Monitor:**

Reset. The first of them is intended for sending a request from Z41 Lite to KES asking for the updated values of the energy measurement, which requires having the object properly linked to the **request** object from KES, while the second object will permit sending reset orders from Z41 Lite to KES, so that all the energy consumption, CO₂ and cost variables are set to zero, which requires having the object properly linked to the **reset** object from KES. Please refer to the user manual of KES for further information.

Note: *the energy monitor objects in Z41 Lite have no restrictions about the meaning of the data they receive from the bus, so it is entirely at the integrator's choice whether the box will monitor estimations, average consumptions, real consumptions, etc.*

Finally, in the pop-up window that shows up when the box is pressed, the following elements are present:

- Evolution of the Power Consumption object: shows, as a graph, the chronological evolution of the value of the power object (**[Energy x] Power**). The horizontal axis will show different time lapses (hours, days, or months), depending on the selected zoom level.
- Instant Value of the Energy, CO₂ and Cost objects: indicator that, depending on whether the **Energy consumption, CO₂ and Cost** parameters have been enabled, will show the current value of, respectively, the **[Energy x] Energy, [Energy x] CO₂ and [Energy x] Cost** objects.
- Zoom Level: two-button control that lets the user change the time scale, so more / less detail is shown in the graph of the evolution of the power object. The behaviour and the representation of the data are similar to those of the numeric data log (see epigraph e) in section 3.4.2.1).
- Validation / Refresh / Reset buttons: permit, respectively, closing the pop-up window, sending a reset order (value "1" through the **[Pn][Bi] Energy Monitor: Reset** object) and requesting an updated consumption report (value "1" through the **[Pn][Bi] Energy Monitor: Request** object).

d) Daily Timer

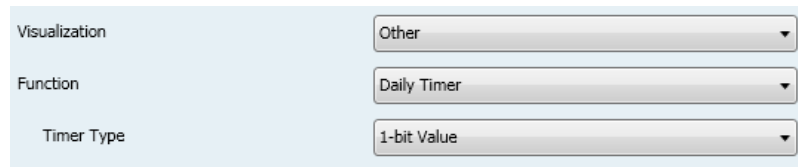


Figure 72 Daily Timer

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.



Figure 73 Daily Timer (Box)

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.

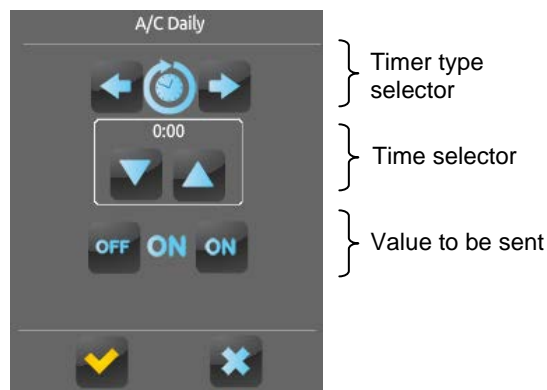


Figure 74 Daily Timer (Pop-Up Window)

This window contains the following elements:

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

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

- **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.

Finally, this function offers the following parameters in ETS:

- **Timer Type:** switches between “1-bit value” (default option) and “Scene”.
- **Scene Number:** shown only after selecting “Scene” for the above parameter. This field defines the number of the scene (1-64) that will be sent to the bus on the execution of the timed sending.

e) Weekly Timer

Visualization	Other
Function	Weekly Timer
Timer Type	1-bit Value

Figure 75 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.



Figure 76 Weekly Timer (Box).

Hence, when the user presses on the box, a window similar to Figure 77 will pop up

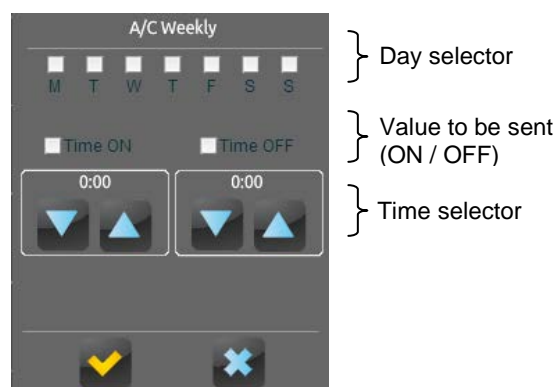


Figure 77 Weekly Timer (Pop-Up Window).

This window contains the following elements:

- Day Selector: lets the user select the days on which the timed sending will take place.
- 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. One time 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 one time 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 timed sending set by the user. By default, timed sending is 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.*

Finally, this function offers the following parameters in ETS:

- **Timer Type:** switches between “1-bit value” (default option) and “Scene”.
- **Scene Number:** shown only after selecting “Scene” for the above parameter. This field defines the number of the scene (1-64) that will be sent to the bus on the execution of the timed sending.

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.*

f) **Chrono-thermostat:**

Boxes configured as chrono-thermostats let the final user schedule the timed sending of **temperature setpoint orders** (preceded by switch-on orders) or **switch-off orders** to a thermostat through the KNX bus.

To that end, it is necessary to configure the following two parameters:

Label	<input type="text"/>
Visualization	Other
Function	Chrono-thermostat
Minimum Setpoint Value	18 <input type="button" value="↑"/> <input type="button" value="↓"/> x 1°C
Maximum Setpoint Value	30 <input type="button" value="↑"/> <input type="button" value="↓"/> x 1°C

Figure 78 Chrono-thermostat

- **Minimum Setpoint Value:** minimum setpoint value the user will be able to set. Range: [-20, 95] x 1°C. Default value: 18°C.
- **Maximum Setpoint Value:** maximum setpoint value the user will be able to set. Range: [-20, 95] x 1°C. Default value: 30°C.

When a box is configured as a chrono-thermostat, it looks as follows:

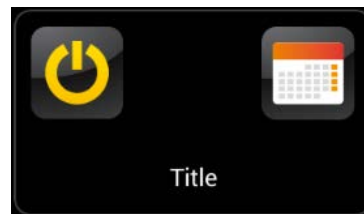



Figure 79 Chrono-thermostat (box)

The **switch button** on the left activates / deactivates the chrono-thermostat, that is, resumes or stops the execution of the timed actions that may have been configured (the configuration is not lost during the deactivation). While the chrono-thermostat is not active, icon  is shown next to the switch button.

On the other hand, when the user touches on the **calendar icon**, a new window will pop up to let the user set the actions to be executed:

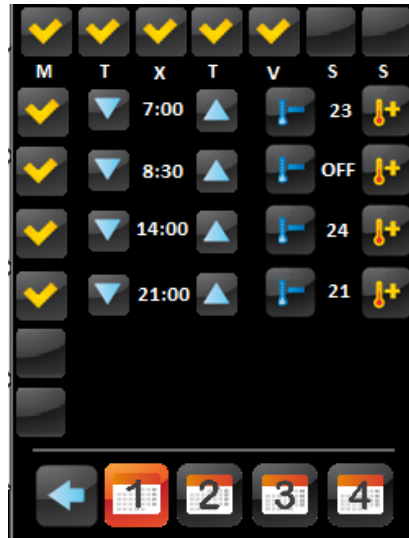


Figure 80 Chrono-thermostat (pop up)

Each box configured as a chrono-thermostat contains **four pages** with **six customisable timers** each, being possible for the user to configure up to 24 timed actions.

The buttons on the bottom of the window let switching across the four pages, as well as closing the pop-up window (button ←).

For each of the six timed actions of each page it is necessary to select, on the first hand, the **days of the week** on which the action should be performed, as well as the specific **time of day**.

After that, the user should select (through controls 🌡+ and 🌡-) the **temperature setpoint that will be sent**. The range of this value is restricted by parameters “**Minimum Setpoint Value**” and “**Maximum Setpoint Value**”.


Note: *the timed sending of a temperature setpoint is always preceded by a switch-on order.*

Instead of a temperature setpoint, it is also possible to send a **switch-off** order to the thermostat. In such case, the user must successively press on the above control until one of the range limits is reached, after which the word “**OFF**” will show on the screen instead of a temperature value.

On the other hand, it is possible to **activate or deactivate** each action separately by marking its corresponding checkbox. While this checkbox remains unmarked, the action configured will not take place.

On **power failures and ETS downloads**, chrono-thermostat boxes behave the same as other user-defined timers: the user configuration will not be lost, although after an ETS download it will be necessary to enable the chrono-thermostat manually.

Each chrono-thermostat box entails the following three objects:

- “[Pn][Bi] Chrono-thermostat: **Enabling**”: permits activating (value “1”) or deactivating (value “0”) the chrono-thermostat analogously as from the on-screen **switch button** of the box, which will show the icon  while deactivated.
- “[Pn][Bi] Chrono-thermostat: **Temperature Setpoint**”: this is the object through which the timed temperature setpoints will be sent.
- “[Pn][Bi] Chrono-Thermostat: **On/Off**”: this is the object through which the timed switch-off orders will be sent.

g) Alarm

Visualization	Other
Function	Alarm
Trigger	1
Periodic Monitoring	No

Figure 81 Alarm

Boxes configured as alarms perform user warnings on anomalous events. For this purpose, boxes of this type are related to a binary object (**[Pn][Bi] Alarm Trigger**) that permits receiving alarm messages from the bus, thus making Z41 Lite 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. After that moment, pressing on any button of the screen will interrupt the beeps and blinks although the box will still show an intermitting icon, which means that the alarm is still active and unconfirmed. Additionally, the icon of the page that contains the box will also show an exclamation.



Figure 82 Alarm (Box).

The intermitting exclamation means that the alarm is active and unconfirmed.

However, alarm boxes implement an additional binary object, **[Pn][Bi] Alarm Confirmation**, intended for receiving the value “1” from the bus, which will be interpreted as an acknowledgement of the alarm, thus making the icon intermittence stop. Alternatively, the alarm can be acknowledged by touching on the “OK” button in the box itself, which will cause the sending of the value “1” through the mentioned object.

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 icon will finally turn off.

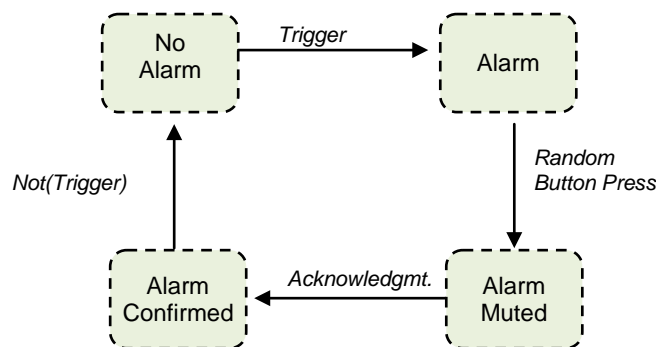


Figure 83 Alarm state diagram (case of confirmation previous to the alarm end)

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 Z41 Lite 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:

- **Trigger:** 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 Z41 Lite as an alarm situation. Implicitly, this parameter also defines the inverse “no alarm” value.

- **Periodic Monitoring:** activates (“Yes”) or deactivates (“No”, default option) periodic monitoring of the alarm trigger object.
- **Cycle Time (x 1min.):** in case of having selected “Yes” under the above parameter, sets the maximum accepted time space without receiving the “no alarm” value before Z41 Lite 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 Z41 Lite where no periodical monitoring has been parameterised and where the value “1” has been set as the alarm value, then Z41 Lite 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”.*

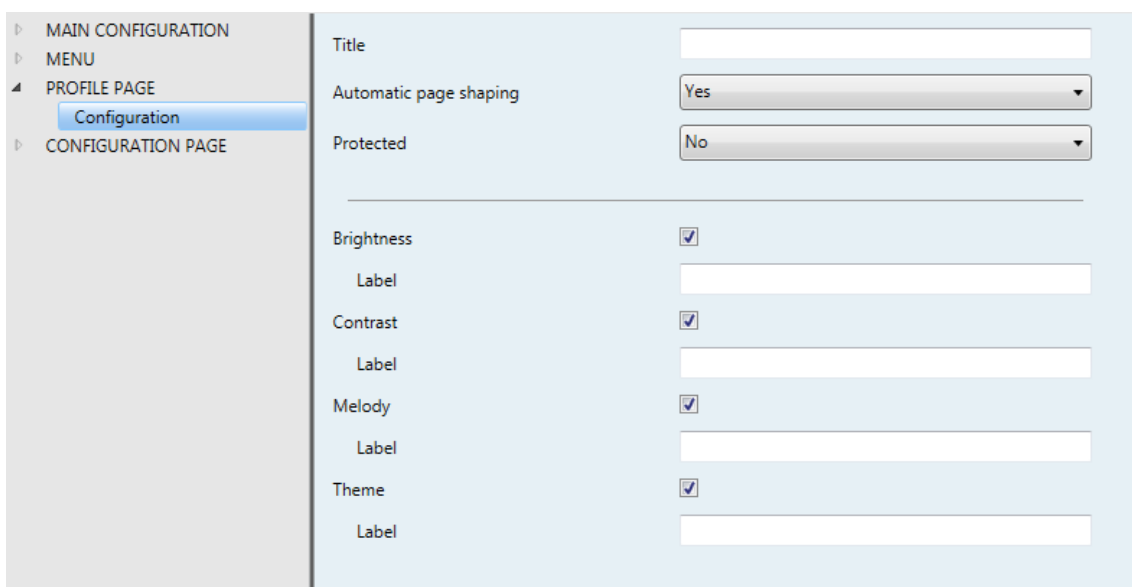
Example II: *suppose a certain CO₂ sensor responsible for sending the value “0” to the bus every two minutes, except when it detects a dangerous CO₂ level; in such case, it will immediately send the value “1”. If the object being sent is linked to the alarm trigger object from an alarm box in Z41 Lite where periodic monitoring has been parameterised and where the value “1” is configured as the alarm trigger value, then Z41 Lite will notify the user about the emergency both if the CO₂ 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.*

3.5 PROFILE PAGE

The Profile Page tab contains a sole screen, Configuration.

3.5.1 CONFIGURATION

This screen permits the integrator configure the controls that will be available for the final user within the Profile page (see section 2.3), as well as the label that will identify them on the screen. Note that the Profile page itself can be activated or hidden from the Configuration screen, under the Menu tab (see section 3.3.1).



▷ MAIN CONFIGURATION	Title	<input type="text"/>
▷ MENU	Automatic page shaping	Yes ▾
▲ PROFILE PAGE	Protected	No ▾
Configuration	<hr/>	
▷ CONFIGURATION PAGE	Brightness	<input checked="" type="checkbox"/>
	Label	<input type="text"/>
	Contrast	<input checked="" type="checkbox"/>
	Label	<input type="text"/>
	Melody	<input checked="" type="checkbox"/>
	Label	<input type="text"/>
	Theme	<input checked="" type="checkbox"/>
	Label	<input type="text"/>

Figure 84 Configuration (Profile Page)

By default, all controls are marked as active. Please refer to section 2.3 for details.

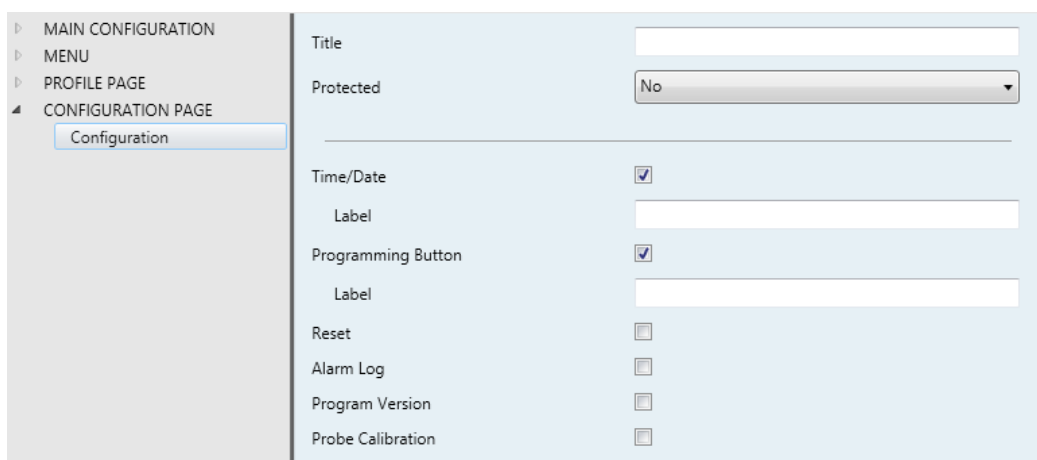
On the other hand, for the Profile page –as for any other page–, a label (parameter **Title**), a box shaping (parameter **Automatic Page Shaping**; see section 2.6) and a security level (parameter **Protected**; see section 3.2.2) can be set.

3.6 CONFIGURATION PAGE

The Configuration Page tab contains a sole screen, Configuration.

3.6.1 CONFIGURATION

This screen permits the integrator configure the controls that will be available for the final user within the Configuration page (see section 2.4), as well as the label that will identify them on the screen. Note that the Configuration page itself can be activated or hidden from the Configuration screen, under the Menu tab (see section 3.3.1).



The screenshot displays the Configuration Page interface. On the left, a navigation menu lists: MAIN CONFIGURATION, MENU, PROFILE PAGE, and CONFIGURATION PAGE (with 'Configuration' selected). The main area contains the following settings:

Title	<input type="text"/>
Protected	<input type="text" value="No"/>
Time/Date	<input checked="" type="checkbox"/>
Label	<input type="text"/>
Programming Button	<input checked="" type="checkbox"/>
Label	<input type="text"/>
Reset	<input type="checkbox"/>
Alarm Log	<input type="checkbox"/>
Program Version	<input type="checkbox"/>
Probe Calibration	<input type="checkbox"/>

Figure 85 Configuration Screen (Configuration Page)

By default, the **Time/Date** and **Programming Button** controls are marked as active. Please refer to section 2.4 for details.

On the other hand, for the Configuration page –as for any other page–, a label (parameter **Title**), a box shaping (parameter **Automatic Page Shaping**; see section 2.6), and a security level (parameter **Protected**; see section 3.2.2) can be set.

3.7 THERMOSTAT n

Thermostat Function	Heating
Reference Temperature	Temperature Source 1
Thermostat always ON?	No
Startup setting (on Bus voltage recovery)	Last (before Bus failure)
Automatic ON when a new special mode arrives	Disabled
Sending Statuses on BUS voltage recovery	No

Figure 86 Configuration Screen (Thermostat n)

When the functions “Thermostat 1” and “Thermostat 2” from the General screen under the Main configuration tab are enabled, the integrator will be shown two more tabs, both similar to each other, for the parameterisation of these functions independently. Under each of the tabs, a set of screens (Configuration, Setpoint, etc.) is provided to define the type of the thermostatic control the user will be performing from Z41 Lite.

Note: for further information about the behaviour and the parameterisation of the Zennio “Building” thermostat implemented by Z41 Lite, please refer to the specific documentation (“**Building Thermostat**”) available at www.zennio.com.

3.8 INPUTS

Z41 Lite incorporates **two analogue/digital inputs**, each configurable as a:

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

Important: *older models of the Zennio motion detector (e.g., ZN1IO-DETEC and ZN1IO-DETEC-N) will not work properly with Z41 Lite, even if they do with different Z41 models.*

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

- **Binary Inputs in Z41 Lite.**
- **Temperature Probe in Z41 Lite.**
- **Motion Detector in Z41 Lite.**

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	3 Bytes	I/O	C T R W U	DPT_TimeOfDay	0:00:00 – 23:59:59	[General] Time	Current Time
2	3 Bytes	I/O	C T R W U	DPT_Date	01/01/1990 - 31/12/2089	[General] Date	Current Date
3	1 Byte		C T - - -	DPT_SceneControl	0-63; 128-191	[General] Scenes: send	Scene Value
4	2 Bytes	O	C T R - -	DPT_Value_Temp	-273,00 - 670760,00	[General] Internal Temperature	Current Temperature (built-in sensor)
5	2 Bytes	I	C - - W U	DPT_Value_Temp	-273,00 - 670760,00	[General] External Temperature	Temperature to show in the display
6	1 Bit	I	C - - W -	DPT_Ack	0/1	[General] Illumination	0=No action; 1=Light Display
7	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Touch Lock	0=Unlocked; 1=Locked
	1 Bit	I	C - - W -	DPT_Switch	0/1	[General] Touch Lock	0=Locked; 1=Unlocked
8	1 Bit	O	C T R - -	DPT_Switch	0/1	[General] Welcome Object	1-bit Generic Control
9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90, 93, 96, 99, 102, 105, 108, 111, 114, 117, 120, 123, 126, 129, 132, 135, 138, 141, 144, 147, 150	1 Bit	I	C - - W U	DPT_Switch	0/1	[Px][By] Binary Indicator	1-bit Indicator
	1 Byte	I	C - - W U	DPT_Value_1_Ucount	0 - 255	[Px][By] Enumerated Indicator	0...255
	1 Byte	I	C - - W U	DPT_Value_1_Ucount	0 - 255	[Px][By] 1-byte Unsigned Int Indicator	0...255
	1 Byte	I	C - - W U	DPT_Value_1_Count	-128 - 127	[Px][By] 1-byte Signed Int Indicator	-128...127
	1 Byte	I	C - - W U	DPT_Scaling	0% - 100%	[Px][By] Percentage Indicator	0-100%
	2 Bytes	I	C - - W U	DPT_Value_2_Ucount	0 - 65535	[Px][By] 2-byte Unsigned Int Indicator	0...65535
	2 Bytes	I	C - - W U	DPT_Value_2_Count	-32768 - 32767	[Px][By] 2-byte Signed Int Indicator	-32768...32767
	2 Bytes	I	C - - W U	9.xxx	-671088,64 – 670760,96	[Px][By] 2-byte Float Indicator	-671088.64...670760.96
	1 Byte	I	C - - W U	DPT_Scaling	0% - 100%	[Px][By] Shutter Position	0%=Top; 100%=Bottom
	1 Byte	I	C - - W U	DPT_Scaling	0% - 100%	[Px][By] Light Indicator	0%=Off; 100%=On
	2 Bytes	I	C - - W U	DPT_Value_Temp	-273,00 - 670760,00	[Px][By] Temperature Indicator	Absolute Float Value
	1 Bit	I	C - - W U	DPT_Heat_Cool	0/1	[Px][By] Mode Indicator	0 = Cool; 1 = Heat
	1 Byte	I	C - - W U	DPT_HVACContrMode	0 = Auto	[Px][By] Mode Indicator	Heat, Cool, Auto, Fan and Dry

					1 = Heat 3 = Cool 9 = Fan 14 = Dry		
	1 Byte	I	C - - W U	DPT_Scaling	0% - 100%	[Px][By] Fan Indicator	0-49%=Min; 50-100%=Max
	1 Byte	I	C - - W U	DPT_Scaling	0% - 100%	[Px][By] Fan Indicator	0-33%=Min; 34-66%=Med; >66%=Max
	1 Byte	I	C - - W U	DPT_Scaling	0% - 100%	[Px][By] Fan Indicator	0=Auto; 1-33%=Min; 34-66%=Med; >66%=Max
	1 Byte	I	C - - W U	DPT_Scaling	0% - 100%	[Px][By] Fan Indicator	0=Off; 1-33%=Min; 34-66%=Med; >66%=Max
	1 Byte	I	C - - W U	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Px][By] Special Mode Indicator	Comfort, Standby, Economy and Building Protection
	1 Byte	I/O	C T R W U	DPT_Scaling	0% - 100%	[Px][By] Red Channel	0-100%
	1 Bit	I	C - - W U	DPT_Enable	0/1	[Px][By] Alarm Trigger	Trigger
	1 Bit	I	C - - W U	DPT_Alarm	0/1	[Px][By] Weekly Timer Enabling	0 = Disabled; 1 = Enabled
	1 Bit	I	C - - W U	DPT_Enable	0/1	[Px][By] Daily Timer Enabling	0 = Disabled; 1 = Enabled
	1 Bit	I	C - - W U	DPT_Enable	0/1	[Px][By] RGB Color	Red, Green and Blue components
	3 Bytes	I/O	C T R W U	DPT_Colour_RGB	[0 - 255] * 3	[Px][By] Chrono-thermostat: enabling	0 = Disabled; 1 = Enabled
10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 49, 52, 55, 58, 61, 64, 67, 70, 73, 76, 79, 82, 85, 88, 91, 94, 97, 100, 103, 106, 109, 112, 115, 118, 121, 124, 127, 130, 133, 136, 139, 142, 145, 148, 151	1 Bit	O	C T R - -	DPT_Switch	0/1	[Px][By] Binary Control	1-bit Control
	1 Byte	O	C T R - -	DPT_Value_1_Ucount	0 - 255	[Px][By] 1-byte Unsigned Int Control	0...255
	1 Byte	O	C T R - -	DPT_Value_1_Count	-128 - 127	[Px][By] 1-byte Signed Int Control	-128...127
	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Px][By] Percentage Control	0-100%
	2 Bytes	O	C T R - -	DPT_Value_2_Ucount	0 - 65535	[Px][By] 2-byte Unsigned Int Control	0...65535
	2 Bytes	O	C T R - -	-32768 - 32767	-32768 - 32767	[Px][By] 2-byte Signed Int Control	-32768...32767
	2 Bytes	O	C T R - -	9.xxx	-671088,64 - 670760,96	[Px][By] 2-byte Float Control	-671088.64...670760.96
	1 Byte	O	C T R - -	DPT_Value_1_Ucount	0 - 255	[Px][By] Enumerated Control	0...255
	1 Bit	O	C T R - -	DPT_UpDown	0/1	[Px][By] Move Shutter	0=Up; 1=Down
	1 Bit	O	C T R - -	DPT_Switch	0/1	[Px][By] Light On/Off	0=Off; 1=On
	2 Bytes	O	C T R - -	DPT_Value_Temp	-273,00 - 670760,00	[Px][By] Temperature Control	Absolute Float Value
	2 Bytes	I/O	C T R W -	DPT_Value_Tempd	-670760,00 - 670760,00	[Px][By] Temperature Offset Control	Relative Float Value
	1 Bit	O	C T R - -	DPT_Step	0/1	[Px][By] Temperature Offset Control	0=Decrease; 1=Increase
	1 Bit	O	C T R - -	DPT_Heat_Cool	0/1	[Px][By] Mode Control	0 = Cool; 1 = Heat
1 Byte	O	C T R - -	DPT_HVACContrMode	0 = Auto 1 = Heat	[Px][By] Mode Control	Heat, Cool, Auto, Fan and Dry	

					3 = Cool 9 = Fan 14 = Dry			
	1 Bit	O	CTR--	DPT_Step	0/1	[Px][By] Fan Control	0=Decrease; 1=Increase	
	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Px][By] Special Mode Control	Comfort, Standby, Economy and Building Protection	
	1 Byte	I/O	CTRWU	DPT_Scaling	0% - 100%	[Px][By] Green Channel	0-100%	
	1 Bit	O	CTR--	DPT_Switch	0/1	[Px][By] Daily Timer Control	1-bit Control	
	1 Bit	O	CTR--	DPT_Switch	0/1	[Px][By] Weekly Timer Control	1-bit Control	
	2 Bytes	O	CTR--	DPT_Value_Temp	-20 – 95 °C	[Px][By] Alarm Confirmation	0=No Action; 1=Confirm	
	1 Bit	I	CT-W-	DPT_Ack	0/1	[Px][By] Energy Monitor: request	0=No Action; 1=Request	
	1 Bit	O	CTR--	DPT_Trigger	0/1	[Px][By] Chrono-thermostat: temperature setpoint	Absolute Float Value	
11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 42, 47, 50, 53, 56, 59, 62, 65, 68, 71, 74, 77, 80, 83, 86, 89, 92, 95, 98, 101, 104, 107, 110, 113, 116, 119, 122, 125, 128, 131, 134, 137, 140, 143, 146, 149, 152	1 Bit	O	CTR--	DPT_Step	0/1	[Px][By] Stop Shutter	0 or 1 -> Stop	
	4 Bit	O	CTR--	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	
	1 Bit	O	CTR--	DPT_Switch	0/1	[Px][By] Blue Channel	0-100%	
	1 Byte	I/O	CTRWU	DPT_Scaling	0% - 100%	[Px][By] Energy Monitor: reset	0=No Action; 1=Reset	
	1 Bit	O	CTR--	DPT_Reset	0/1	[Px][By] Chrono-thermostat: on/off	0 = Off; 1 = On	
	153, 180	2 Bytes	I	C--W-	DPT_Value_Temp	-273,00 - 670760,00	[Tx] Temperature Source 1	External Sensor Measure
	154, 181	2 Bytes	I	C--W-	DPT_Value_Temp	-273,00 - 670760,00	[Tx] Temperature Source 2	External Sensor Measure
155, 182	1 Byte	I	C--W-	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode	1-byte HVAC Mode	
156, 183	1 Bit	I	C--W-	DPT_Trigger	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	
157, 184	1 Bit	I	C--W-	DPT_Trigger	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	
158, 185	1 Bit	I	C--W-	DPT_Trigger	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	
159, 186	1 Bit	I	C--W-	DPT_Trigger	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	

160, 187	1 Bit	I	C - - W -	DPT_Window_Door	0/1	[Tx] Window Status (input)	0=Closed; 1=Open
161, 188	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Tx] Comfort Prolongation	0=Nothing; 1=Timed Comfort
162, 189	1 Byte	O	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Tx] Special Mode Status	1-byte HVAC Mode
163, 190	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
164, 191	1 Bit	I	C - - W -	DPT_Step	0/1	[Tx] Setpoint Step	0=-0.5°C; 1=+0.5°C
165, 192	2 Bytes	I	C - - W -	DPT_Value_Tempd	-670760,00 - 670760,00	[Tx] Setpoint Offset	Float offset value
166, 193	2 Bytes	O	C T R - -	DPT_Value_Temp	-273,00 - 670760,00	[Tx] Setpoint Status	Current setpoint
167, 194	2 Bytes	O	C T R - -	DPT_Value_Temp	-273,00 - 670760,00	[Tx] Basic Setpoint Status	Current basic setpoint
168, 195	2 Bytes	O	C T R - -	DPT_Value_Tempd	-670760,00 - 670760,00	[Tx] Setpoint Offset Status	Current setpoint offset
169, 196	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
170, 197	1 Bit	I	C - - W -	DPT_Heat_Cool	0/1	[Tx] Mode	0 = Cool; 1 = Heat
171, 198	1 Bit	O	C T R - -	DPT_Heat_Cool	0/1	[Tx] Mode Status	0 = Cool; 1 = Heat
172, 199	1 Bit	I	C - - W -	DPT_Switch	0/1	[Tx] On/Off	0=Off; 1=On
173, 200	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] On/Off Status	0=Off; 1=On
174, 201	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Control Variable (Cool)	2-Point Control
	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Control Variable (Cool)	PI Control (PWM)
175, 202	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Control Variable (Heat)	2-Point Control
	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Control Variable (Heat)	PI Control (PWM)
176, 203	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Tx] Control Variable (Cool)	PI Control (Continuous)
177, 204	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Tx] Control Variable (Heat)	PI Control (Continuous)
178, 205	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Additional Cool	Temp >= (Setpoint+Band) => "1"
179, 206	1 Bit	O	C T R - -	DPT_Switch	0/1	[Tx] Additional Heat	Temp <= (Setpoint-Band) => "1"
207, 211, 215, 219, 223, 227	2 Bytes	I	C - - W -	DPT_Power	-670760,00 - 670760,00	[Energy X] Power	kW
208, 212, 216, 220, 224, 228	4 Bytes	I	C - - W -	DPT_ActiveEnergy_kWh	-2147483648 - 2147483647	[Energy X] Energy	kWh
209, 213, 217, 221, 225, 229	2 Bytes	I	C - - W -	CO2 emission (non-standardized datapoint type)	-670760,00 - 670760,00	[Energy X] CO2	kg CO2
210, 214, 218, 222, 226, 230	2 Bytes	I	C - - W -	Currency (non-standardized datapoint type)	-670760,00 - 670760,00	[Energy X] Cost	Local Currency
231-234	1 Byte	I/O	C T R W U	DPT_Scaling	0% - 100%	[RGBW] White Channel 4	0-100%
235, 241	1 Bit	I	C - - W -	DPT_Switch	0/1	[Ix] Input Lock	1 = Locked; 0 = Unlocked

236, 242	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] 0	Sending of 0
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] 1	Sending of 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Short Press] 0/1 Switching	Switching 0/1
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Short Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Short Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Brighter	Increase Brightness
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Darker	Decrease Brightness
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Short Press] Brighter/Darker	Switch Bright/Dark
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] Dimmer ON	Sending of 1 (ON)
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Short Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Short Press] Dimmer ON/OFF	Switching 0/1
1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Run Scene	Sending of 0 - 63	

	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Ix] [Short Press] Save Scene	Sending of 128 - 191
	1 Bit	I/O	CTRW-	DPT_Switch	0/1	[Ix] [Switch/Sensor] Edge	Sending of 0 or 1
	1 Byte		CT---	DPT_Value_1_Ucount	0 - 255	[Ix] [Short Press] Constant Value (Integer)	0 - 255
	1 Byte		CT---	DPT_Scaling	0% - 100%	[Ix] [Short Press] Constant Value (Percentage)	0% - 100%
	2 Bytes		CT---	DPT_Value_2_Ucount	0 - 65535	[Ix] [Short Press] Constant Value (Integer)	0 - 65535
	2 Bytes		CT---	9.xxx	-671088.64 - 670760.96	[Ix] [Short Press] Constant Value (float)	Float value
237, 243	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix] [Short Press] Shutter Status (input)	0% = Top; 100% = Bottom
	1 Byte	I	C--W-	DPT_Scaling	0% - 100%	[Ix] [Short Press] Dimming Status (input)	0% - 100%
238, 244	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] 0	Sending of 0
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] 1	Sending of 1
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Long Press] 0/1 Switching	Switching 0/1
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Up Shutter	Sending of 0 (Up)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Down Shutter	Sending of 1 (Down)
	1 Bit		CT---	DPT_UpDown	0/1	[Ix] [Long Press] Move Up/Down Shutter	Switching 0/1 (Up/Down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Up Shutter	Sending of 0 (Stop/Step up)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Down Shutter	Sending of 1 (Stop/Step down)
	1 Bit		CT---	DPT_Step	0/1	[Ix] [Long Press] Stop/Step Shutter (switched)	Switching of 0/1 (Stop/Step up/down)
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%) ... 0xF (Inc. by 1%)	[Ix] [Long Press] Brighter	Long Pr. -> Brighter; Release -> Stop
	4 Bit		CT---	DPT_Control_Dimming	0x0 (Stop) 0x1 (Dec. by 100%) ... 0x7 (Dec. by 1%) 0x8 (Stop) 0x9 (Inc. by 100%)	[Ix] [Long Press] Darker	Long Pr. -> Darker; Release -> Stop

					0xF (Inc. by 1%)		
	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] Dimmer ON	Sending of 1 (ON)
	1 Bit		CT---	DPT_Switch	0/1	[Ix] [Long Press] Dimmer OFF	Sending of 0 (OFF)
	1 Bit	I	CT-W-	DPT_Switch	0/1	[Ix] [Long Press] Dimmer 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
239, 245	1 Bit		CT---	DPT_Trigger	0/1	[Ix] [Long Press/Release] Stop Shutter	Release -> Stop Shutter
240, 246	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
247	1 Byte	I	C--W-	DPT_SceneControl	0-63; 128-191	[Motion Sensor] Scene Input	Scene Value
248	1 Byte		CT---	DPT_SceneControl	0-63; 128-191	[Motion Sensor] Scene Output	Scene Value
249, 273	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix] Luminosity	0-100%
250, 274	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Open Circuit Error	0 = No Error; 1 = Open Circuit Error
251, 275	1 Bit	O	CTR--	DPT_Alarm	0/1	[Ix] Short Circuit Error	0 = No Error; 1 = Short Circuit Error
252, 276	1 Byte	O	CTR--	DPT_Scaling	0% - 100%	[Ix] Presence State (Scaling)	0-100%
253, 277	1 Byte	O	CTR--	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] Presence State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
254, 278	1 Bit	O	CTR--	DPT_Occupancy	0/1	[Ix] Presence State (Binary)	Binary Value

	1 Bit	O	C T R - -	DPT_Trigger	0/1	[Ix] Presence: Slave Output	1 = Motion Detected
255, 279	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Ix] Presence Trigger	Binary Value to Trigger the Presence Detection
256, 280	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Ix] Presence: Slave Input	0 = Nothing; 1 = Detection from slave device
257, 281	1 Bit	I	C - - W -	DPT_Trigger	0/1	[Ix] External Motion Detection	0 = Nothing; 1 = Motion detected by an external sensor
258, 263, 268, 282, 287, 292,	1 Byte	O	C T R - -	DPT_Scaling	0% - 100%	[Ix] [Cy] Detection State (Scaling)	0-100%
259, 264, 269, 283, 288, 293	1 Byte	O	C T R - -	DPT_HVACMode	1=Comfort 2=Standby 3=Economy 4=Building Protection	[Ix] [Cy] Detection State (HVAC)	Auto, Comfort, Standby, Economy, Building Protection
260, 265, 270, 284, 289, 294	1 Bit	O	C T R - -	DPT_Switch	0/1	[Ix] [Cy] Detection State (Binary)	Binary Value
261, 266, 271, 285, 290, 295	1 Bit	I	C - - W -	DPT_Switch	0/1	[Ix] [Cy] Channel Lock	According to parameters
262, 267, 272, 286, 291, 296	1 Bit	I	C - - W -	DPT_Switch	0/1	[Ix] [Cy] Force State	0 = No Detection; 1 = Detection
297, 301	2 Bytes	O	C T R - -	DPT_Value_Temp	-273,00 - 670760,00	[Ix] Current Temperature	Temperature sensor value
298, 302	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Overcooling	0 = No Alarm; 1 = Alarm
299, 303	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Overheating	0 = No Alarm; 1 = Alarm
300, 304	1 Bit	O	C T R - -	DPT_Alarm	0/1	[Ix] Probe Error	0 = No Alarm; 1 = Alarm

Join and send us your inquiries
about Zennio devices:

<http://zennioenglish.zendesk.com>

Zennio Avance y Tecnología S.L.

C/ Río Jarama, 132. Nave P-8.11
45007 Toledo (Spain).

Tel. +34 925 232 002.

Fax. +34 925 337 310.

www.zennio.com

info@zennio.com



RoHS