

Graphic controls

# SmartTouch

v1.5.1

## Programming manual



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

# Index

<b>1</b>	<b>GENERAL DESCRIPTION .....</b>	<b>3</b>
<b>2</b>	<b>TECHNICAL INFORMATION .....</b>	<b>4</b>
<b>3</b>	<b>PROGRAMMING .....</b>	<b>5</b>
3.1	INTRODUCTION .....	5
3.2	FIRST STEPS .....	6
3.3	DEVICE .....	7
3.4	FIRMWARE UPDATE.....	7
3.5	GROUP ADDRESSES IMPORT .....	8
3.6	INDIVIDUAL ADDRESS ASSIGNMENT .....	9
3.7	THEME CREATION .....	9
3.7.1	<i>Background pictures.....</i>	<i>9</i>
3.7.2	<i>Style edition palette .....</i>	<i>10</i>
3.8	ADDING NEW PAGES .....	12
3.9	ADDING NEW ICONS.....	12
3.10	TYPE OF ICONS .....	13
3.10.1	<i>Button .....</i>	<i>14</i>
3.10.2	<i>Dimmer .....</i>	<i>16</i>
3.10.3	<i>Shutter/blind.....</i>	<i>17</i>
3.10.4	<i>External thermostat.....</i>	<i>18</i>
3.10.5	<i>Internal thermostat.....</i>	<i>21</i>
3.10.6	<i>Scene.....</i>	<i>26</i>
3.10.7	<i>Colour picker.....</i>	<i>27</i>
3.11	OTHER FUNCTIONS .....	28
3.11.1	<i>Date and time .....</i>	<i>28</i>
3.11.2	<i>Technical alarms .....</i>	<i>29</i>
3.11.3	<i>Macros .....</i>	<i>30</i>
3.11.4	<i>Appearance change .....</i>	<i>30</i>
3.11.5	<i>Administrator password .....</i>	<i>31</i>
3.11.6	<i>Default actions.....</i>	<i>32</i>
3.11.7	<i>Values in labels .....</i>	<i>32</i>
3.11.8	<i>Logic gates.....</i>	<i>33</i>
3.11.9	<i>Scripts .....</i>	<i>33</i>
3.11.10	<i>Filtering group addresses.....</i>	<i>37</i>
3.12	DOWNLOADING THE PROJECT.....	38
<b>4</b>	<b>INSTALLATION .....</b>	<b>39</b>

# 1 General description

---

The SmartTouch (Ref. 34XX00) is a full colour capacitive 4.3" touch screen to control and monitor any KNX installation, using 3D plans, photographs, friendly-icons, etc.

It allows to program up to 4 pages with 8 icons in each page with which to control lights, shutters, thermostats, RGB leds, executing scenes and control any other KNX installation devices being ideal to work as a room controller.

It incorporates two user levels (normal user and administrator) and the appearance of the pictures, menus and icons shown is completely customizable by the programmer with the programming software SIDEKNX that allows creating different themes. The appearance theme can be also changed manually by the user or automatically by programmed hours.



## General characteristics:

- Wi-Fi connectivity with integrated web server for remote control via Android/iOS applications. (Basic version: only local control; Plus version: local and/or remote control).
- Admits up to 4 control pages (background: 3D plans, photographs, etc.).
- 8 control icons per plan. Up to 32 controls in total for scenes, on/off, dimming, shutter, thermostats, RGB Leds control, etc.
- It can store up to 16 programmable scenes from the software SIDEKNX. The user can program the time for the scenes through the screen.
- Internal temperature controller for one climate area. It also incorporates chronothermostat that allows user to schedule different configurations of the internal thermostat.
- Up to 6 fully customizable themes (icons, background and menus appearance).
- Programmable automatic theme change by hour.
- Two user level modes: administrator and standard user, to protect configuration settings.
- Possibly to execute 2 gesture macros by sliding up and down in stand-by.

## 2 Technical information

Power supply	18 - 30 Vdc from auxiliary power supply.
Current consumption	250 mA from auxiliary power supply 18 - 30 Vdc. 1 mA from KNX BUS.
Mounting	Surface. Flush mounting with box (included). On universal distribution box, screwed on wall. Easily mounting on plasterboard wall.
Size	129 x 88 x 4 mm (13 mm depth).
Connections	Quick micro-connector T. BUS and Supply connection terminal KNX.
Programming	Via Wi-Fi connectivity (FTP upload).
Supported software	SIDEKNX v1.3.6
Control pages	Up to 4.
Icons per page	Up to 8.
Environment temperature range	Operation: -10°C/55°C Storage: -30°C/60°C Transportation: -30°C/60°C
Regulation	According to the directives of electromagnetic compatibility and low voltage: EN 50090-2-2 / UNE-EN 61000-6-3:2007 / UNE-EN 61000-6-1:2007 / UNE-EN 61010-1.

## 3 Programming

---

### 3.1 Introduction

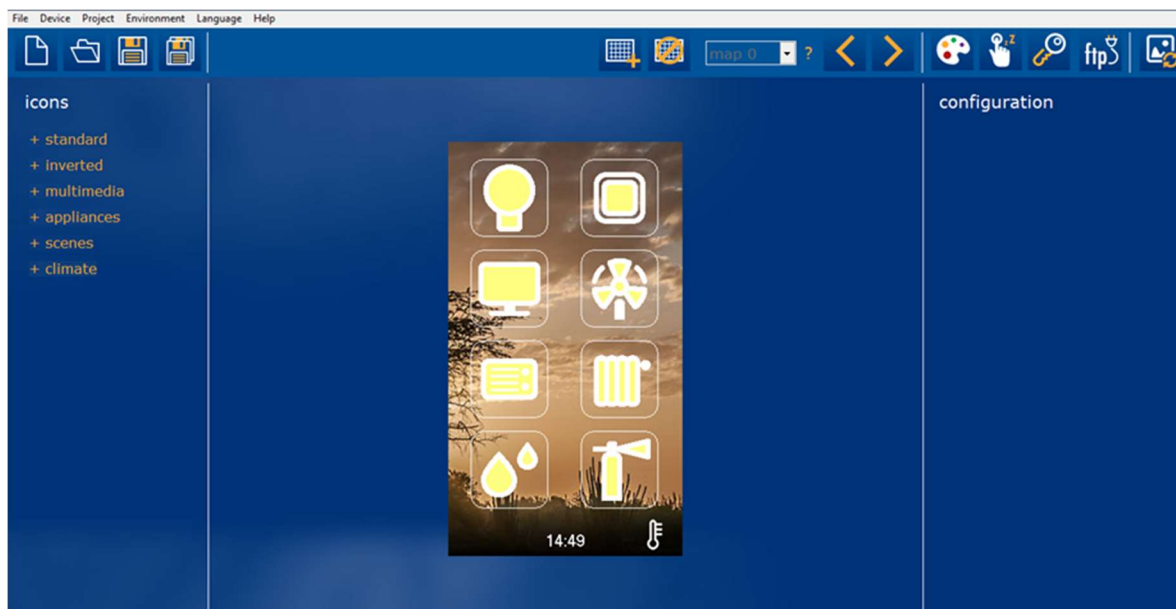
---

The Development System KNX® (SiDE KNX) is the software for programming Bes web servers and touchscreens.

Some of the general features of the SIDE KNX are the following:

- **Quick and intuitive programming:** It is a graphic tool with which what you see is what you program. All the devices that have been added to the project are shown intuitively over the plans and every icon is configured according to the programming needs.
- **Importing information from ETS:** The group addresses created in the ETS project can be imported in \*.xml to make the programming easier.
- **FTP programming:** The programming of the device is done quickly by downloading the project by FTP through Wi-Fi or Ethernet connection.
- **Easy and free updates:** Both the software and the device firmware is easily updated through the internet for free.

With this software, the programmer can load pictures (usually photos or 3D drawings of the house or plant areas) and place over them icons of controlled objects. Each icon is linked to different KNX control devices. When everything is programmed, the project should be transferred to touch screens or web servers, to do this the computer must be connected to the device via TCP/IP connection. After downloading, the device will automatically restart and when finished, it will be ready to operate.



## 3.2 First steps

---

If this is the first time you program a BES touch screen or web server you can read this quick guide to find a summary of how to develop a project with the SiDE KNX software.

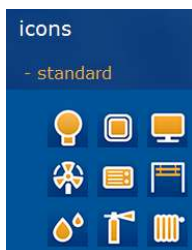
1. First select the type of project according to the target device by doing click on the button "new".



2. Select the background of the visualization. You can upload any jpg or bmp image, either a photograph or a 3D design.



3. Do click on the icon set available on the left side of the window to place them over the planes or pages as needed.



4. Select the type of control for each icon on the right side of the window to set its operation.



5. Link the icon to KNX group addresses. You can enter the addresses manually or use the selector if they have been previously imported from the ETS.



6. Connect the screen or web server to the same network as the PC to download the project through Wi-Fi or Ethernet and, finally, click the button to transfer the project to the IP address of the device.

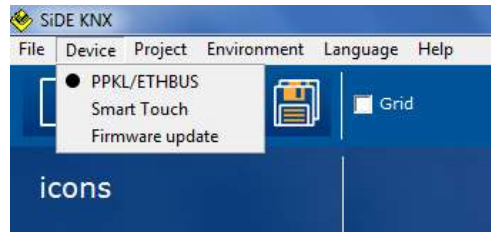


7. After downloading the project, the device will restart automatically and the new configuration will be ready.

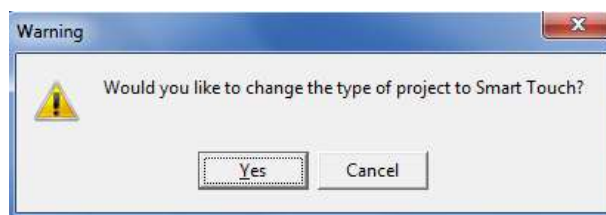
### 3.3 Device

---

The first thing to do when starting a programming project with the SIDEKNX is selecting the type of device that will be the target of the project. The software can be used for programming Ingenium KNX web servers (Ethbus) and different types of touchscreens (PPKL-x, Smart Touch, etc.).



When the target of the project is a Smart Touch, it must be selected in the corresponding menu: Device -> Smart Touch.



### 3.4 Firmware update

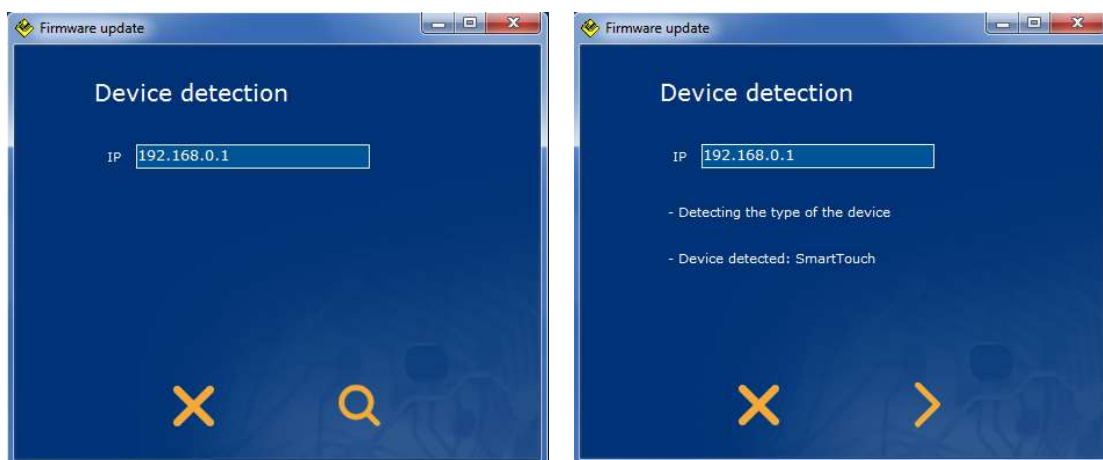
---

It is possible to update the firmware of the device by using this option from the device menu. The new firmware is downloaded through FTP by using the Wi-Fi or Ethernet connection of the device, so the computer and the device must be connected to the same network.

First, write the IP address of the device and do click on the search button at the bottom right of the window.



The SIDEKNX software will search for it and read the type of device.



Then, do click on the arrow button to download the new firmware



The newest firmware will be downloaded and the device will reboot automatically after that.

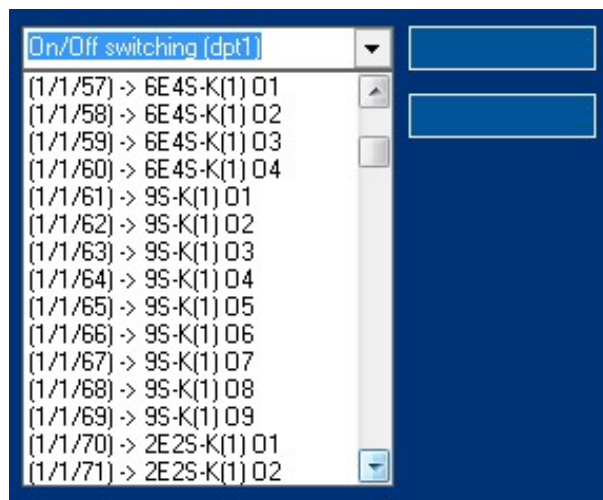


### 3.5 Group addresses import

---

The SIDEKNX allows importing the group addresses created in any ETS project by clicking on the environment menu and import group addresses. The file to import must be in XML format.

Once the group addresses have been imported they can be selected for each object when programming the type of icons.



Do click on *Reset group addresses* option in the environment menu to remove them from the SIDEKNX project. It is recommended to do so before importing new xml files.



### 3.6 Individual address assignment

The Smart Touch individual address it is not assigned by the ETS as other KNX devices. This touchscreen is completely programmed by the SIDEKNX software, including the individual address and group associations.

Do click on *Project > Individual address* from menu bar to show the individual address configuration window.



Here it is set the KNX individual address that the device takes when it is programmed. Write it down manually in the standard individual address format: x.x.x.

### 3.7 Theme creation

The appearance of the screen icons and menus can be changed by selecting different themes according to the customer likes in the configuration menu of the Smart Touch. It can be selected a fixed theme or a list of themes that will change automatically by programmed hours (see the device user manual).

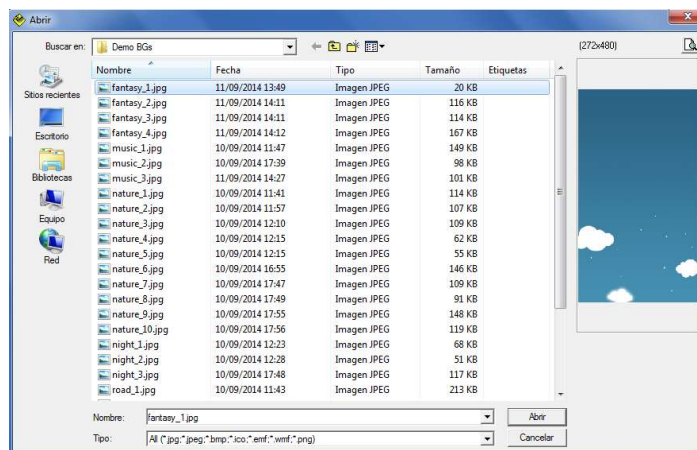
The Smart Touch incorporates 3 default themes to change the appearance of the screen with different colours and icon styles, but it can also incorporate another 3 themes that can be customized by the programmer/integrator by using the SIDEKNX theme creator tools.

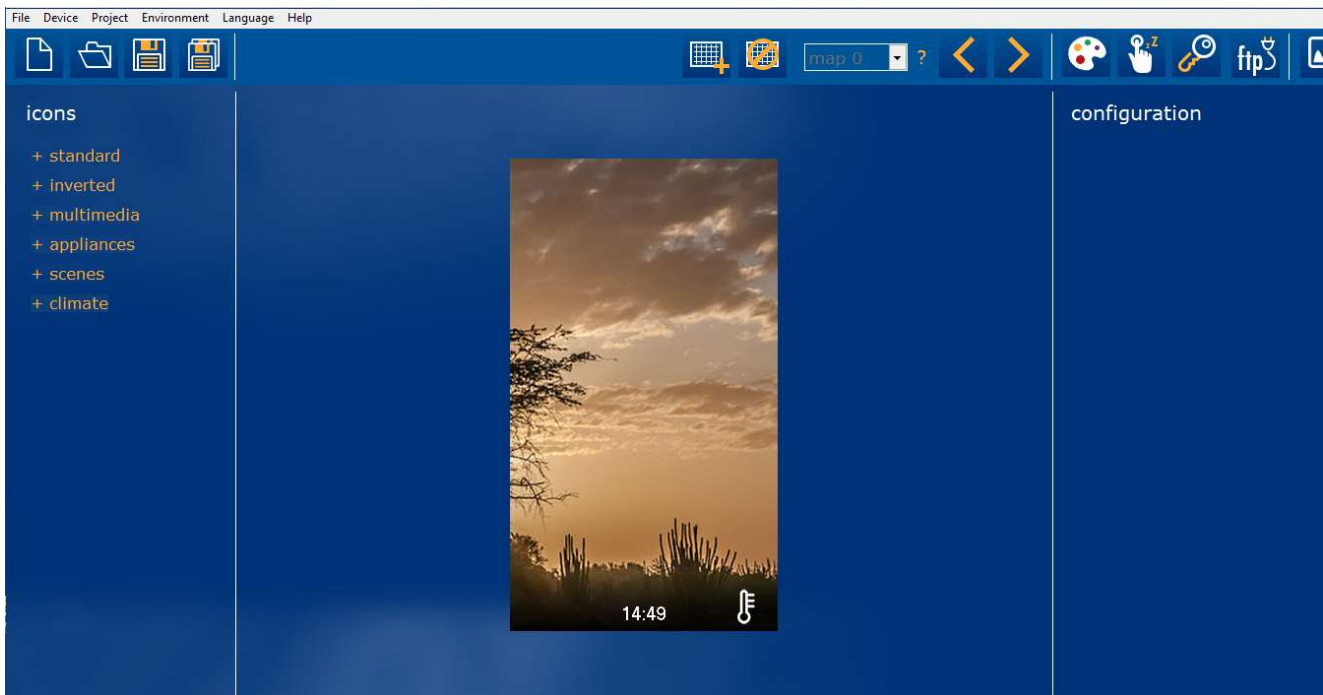
#### 3.7.1 Background pictures

Each theme must have a background picture that it is loaded with the following button:



In the installation path of the SIDEKNX (C:\Program Files (x86)\Ingenium\SiDE KNX\Demo BGs) there are some different pictures available, use the window that appears to select which one to be loaded in the current theme.



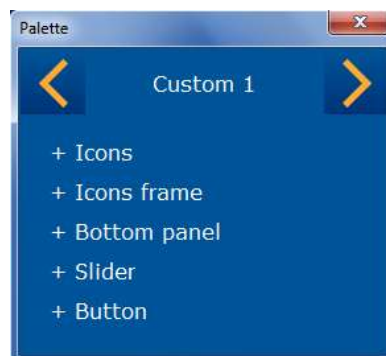


### 3.7.2 Style edition palette

Each appearance theme has its own style that can be loaded from the installation path (C:\Program Files (x86)\Ingenium\SiDE KNX\Demo Themes) by using the option menu Environment -> Import appearance theme. Also the style of the icons and menus of the Smart Touch can be edited by clicking on the palette icon.



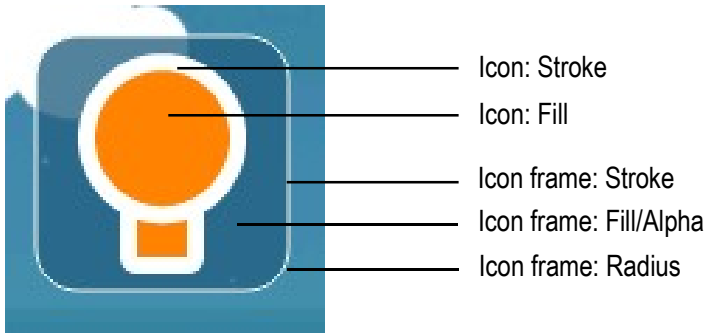
Use the right and left arrows at the top of the palette window to change between the 3 themes and editing their styles.



These are the part of icons, menus and control dialogues that can be edited for each theme. It can be selected any colour and in some cases a factor of transparency: alpha, which can vary from 0 to 255, being 0 the maximum transparency and 255 the maximum opacity.

Icons: Stroke, Fill and Main.

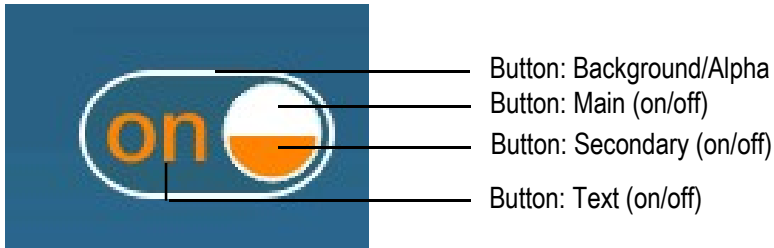
Icons frame: Stroke, Fill/Alpha, and Radius.



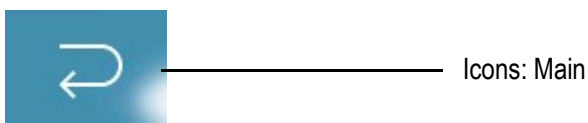
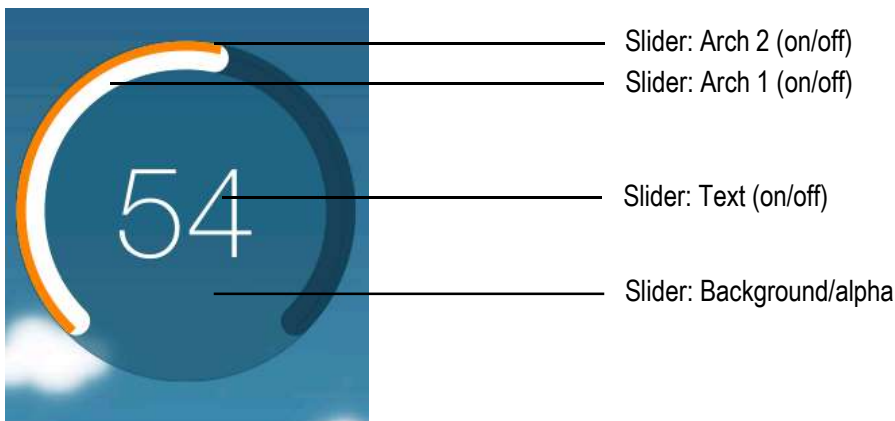
Bottom panel: Background, text.



Button: Background, Main, Secondary, Text.



Slider: Background, Arch 1, Arch 2, Text.



### 3.8 Adding new pages

---

Use the left and right arrows to add new pages of icons to the Smart Touch. It can be added up to 4 pages with 8 icons each.

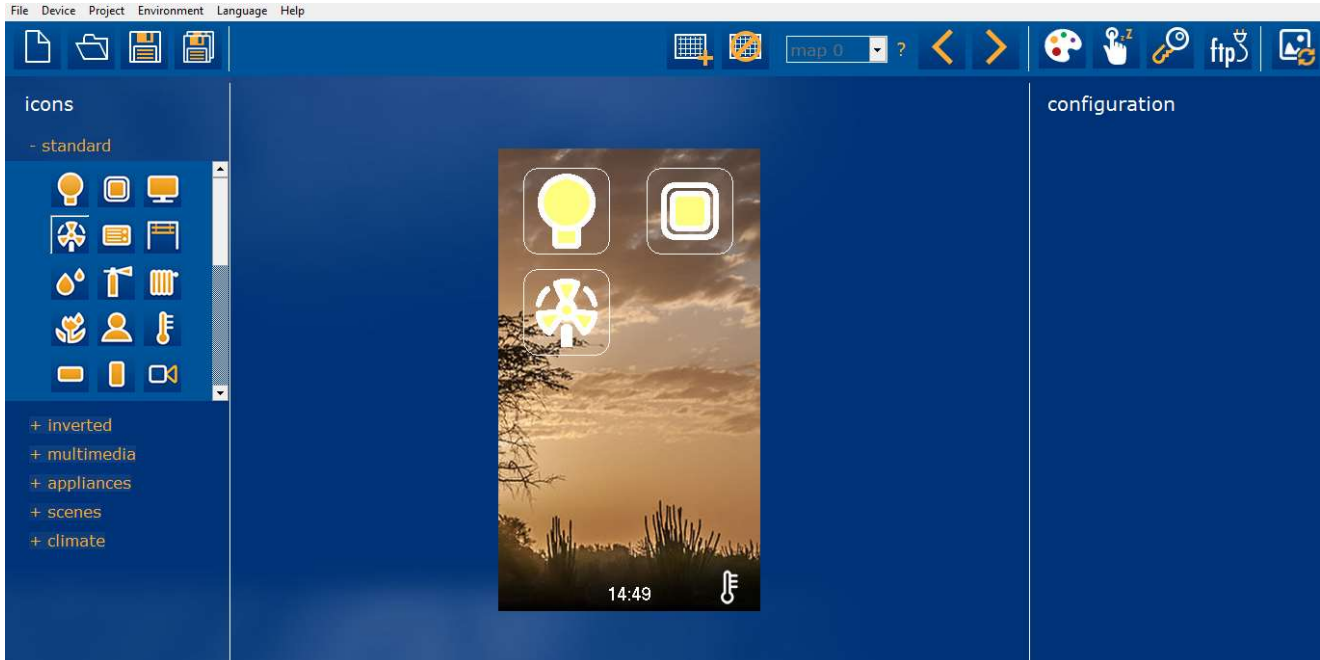
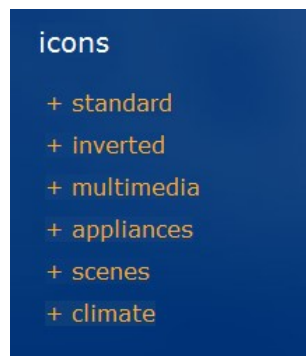


Do not confuse backgrounds with pages, only one background can be loaded per theme. It cannot be left empty pages in the programming project.

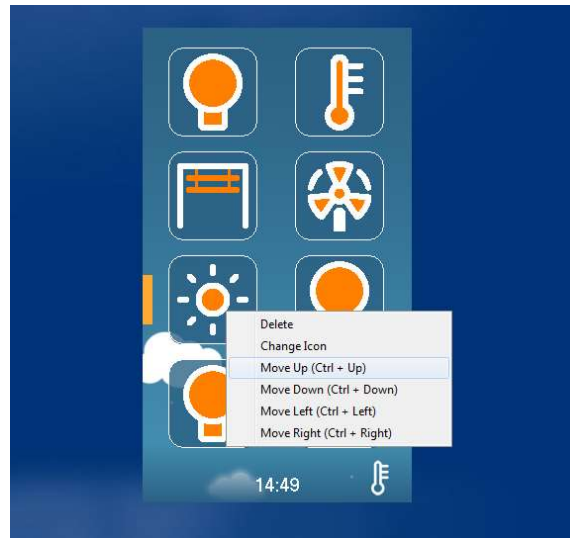
### 3.9 Adding new icons

---

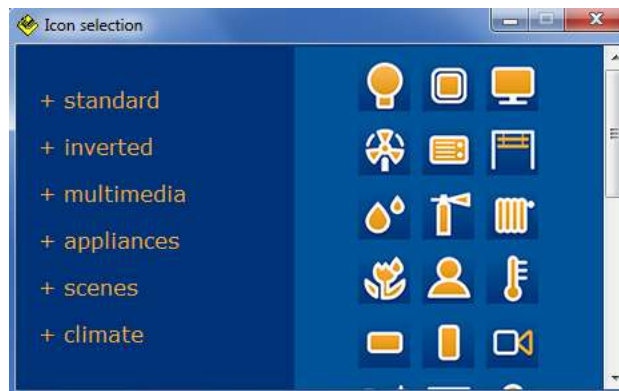
Add icons to the current page with the tree on the left. Open each category and press over the icons to include them to the project. Each icon will be added consecutively.



After adding every icon, they can be put in order by doing click with the right mouse button or by the keyboard shortcut (Ctrl+up/down/left/right).



Do click on change icon and select the new one if you want to change the icon appearance without changing its properties.



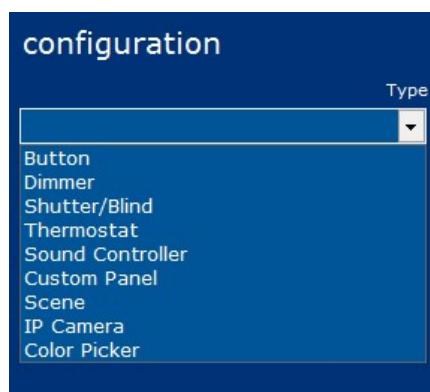
### 3.10 Type of icons

---


Do click on each icon to select it and then program the type using the combo box on the right.

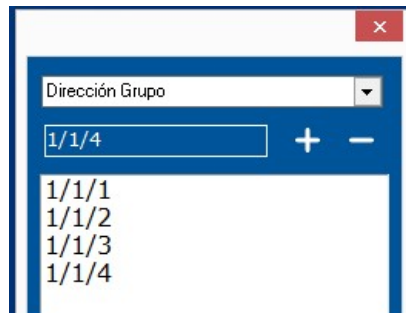


The type of icons available in the Smart Touch and their communication objects are explained next.



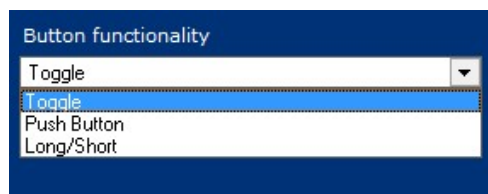
Use the name field to write the label that the icon will have on the screen and use write also the group addresses for each communication object.

Do click on the following button to link more than one group address to an object: 



### 3.10.1 Button

When the type *button* is selected the following options are shown:



#### TOGGLE BUTTON

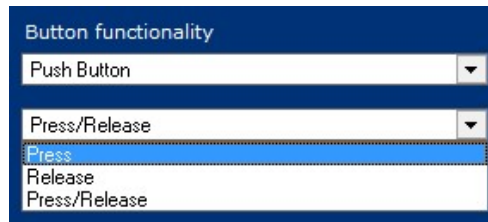
Use the toggle type button to control a light or any other output and perform a standard switch on / switch off control. The following communication objects can be programmed for the toggle type button:

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Button toggle: action	1 bit	1.001	•		•		
1	Button toggle: feedback	1 bit	1.001	•	•		•	

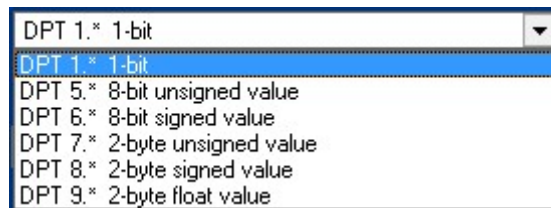
<b>Name</b>	<b>Button toggle: action</b>
<b>Function</b>	1 bit communication object to switch on or switch off an output.
<b>Description</b>	When it receives a “1” through this object the output switches on. When a “0” is received the output switches off
<b>Name</b>	<b>Button toggle: feedback</b>
<b>Function</b>	1 bit communication object to read or notify the output status
<b>Description</b>	When the output is “OFF” and receives a switch on telegram, it sends a “1” through this object. When the output is “ON” and receives a switch off telegram, it sends a “0” through this object.

## PUSH BUTTON (PRESS/RELEASE)

Use the push button type if it is necessary to execute different actions depending on the press and release events. When a push button is configured it can also be set if using press action, release action or both.



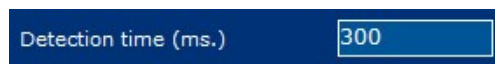
Then select the size and dpt of each object. Each dpt and size will allow you to do different actions:



Object	Name	Flags				
		C	R	W	T	U
0	Button press object: action	●		●		
1	Button release object: action	●		●		
2	Button press/release: icon feedback	●	●		●	

## LONG/SHORT BUTTON

Use the long/short button type if it is necessary to execute different actions with long and short pulsations of the icon. When this button is configured it must be set the time that the device uses to evaluate the type of pulsation.



It also allows to choose the size and the dpt of each object. Each dpt and size will allow you to do different actions

Object	Name	Flags				
		C	R	W	T	U
0	Button short press object: action	●			●	
1	Button long press object: action	●			●	
2	Button long/short: icon feedback	●		●		●

### 3.10.2 Dimmer

When the type *Dimmer* is selected the following objects are shown:

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	On/off switching	1 bit	1.001	•		•		
1	On/off feedback	1 bit	1.001	•	•		•	
2	Brightness value	1 byte	5.001	•		•		
3	Brightness value feedback	1 byte	5.001	•	•		•	

<b>Name</b>	<b>On/off switching</b>
<b>Function</b>	1 bit communication object to switch on and off the regulation channel
<b>Description</b>	When a “1” is received through this object the dimmer is switched on and the brightness level goes up to the last regulation value (non-zero). When a “0” is received the light is switched off.  By default, the behaviour of light when it is turned on through this object is to take the value it had before switching off.
<b>Name</b>	<b>On/off feedback</b>
<b>Function</b>	1bit communication object to read or notify the (on / off) state of the dimmer.
<b>Description</b>	When the light is off and receives a switch on telegram or a brightness value, a “1” is sent through this object. When the light is on and it receives a switch off telegram or a brightness value of 0%, a “0” is sent through this object.
<b>Name</b>	<b>Brightness value</b>
<b>Function</b>	1 byte communication object for precise control by setting a new brightness level directly.
<b>Description</b>	The brightness level will go up or down slowly according to the ramp speed configured.  By default, the behaviour when receiving a telegram with non-zero brightness value when the light is turned off, is to turn on the light and regulate the level received. Also, if the light is on you can turn it off by a telegram with the value 0.



<b>Name</b>	<b>Brightness value feedback</b>
<b>Function</b>	1byte communication object to notify the current brightness level of the dimmer.
<b>Description</b>	When a "0" is received through this object the device cannot be controlled by BUS telegrams (input is not disabled). When a "1" is received the device control by BUS is enabled. By default, this feature is enabled. No need to use this object to enable normal device function.

### 3.10.3 Shutter/blind

When the type *Shutter/Blind* is selected the following objects are shown:

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Position value	1 byte	5.001	•		•		
1	Position feedback	1 byte	5.001	•	•		•	
2	Lamella position	1 byte	5.001	•		•		
3	Lamella feedback	1 byte	5.001	•	•		•	

<b>Name</b>	<b>Position value</b>
<b>Function</b>	1 byte-Communication object to position the shutter/blind in a value directly
<b>Description</b>	When a value is sent through this object the shutter/blind moves to the received position, being 0 = completely closed and 255 = completely open
<b>Name</b>	<b>Position feedback</b>
<b>Function</b>	1 byte-Communication object to read or notify the shutter/blind position
<b>Description</b>	When a value is sent through this object the shutter/blind moves to the received position, being 0 = completely closed and 255 = completely open

<b>Name</b>	<b>Lamella position</b>
Function	1 byte-Communication object to position the lamella in a value directly
Description	When a value is sent through this object the lamella moves to the received position, being 0 = completely closed and 255 = completely open
<b>Name</b>	<b>Lamella feedback</b>
Function	1 byte-Communication object to read or notify the lamella position
Description	When a value is sent through this object the lamella moves to the received position, being 0 = completely closed and 255 = completely open

### 3.10.4 External thermostat

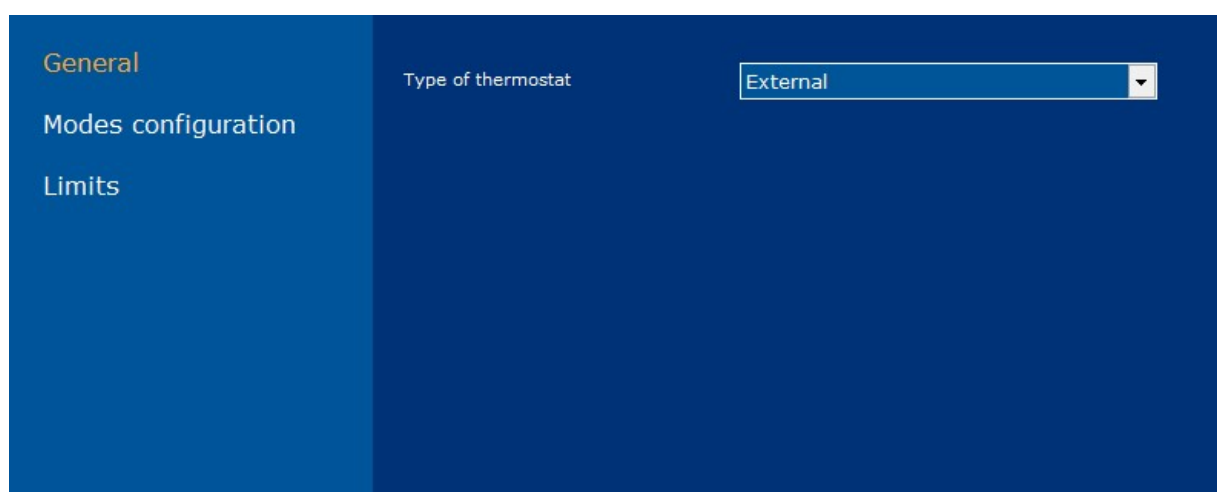
Use this type of icon to control a thermostat, temperature probe, gateway, etc. of the home automation system by using a thermostat control dialogue.

When the type *Thermostat* is selected the following objects are shown:

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	On/off switching	1 bit	1.001	•		•		
1	On/off feedback	1 bit	1.001	•	•		•	
2	Set point temperature	2 bytes	9.001	•		•		
3	Set point temperature feedback	2 bytes	9.001	•	•		•	
5	Room temperature feedback	2 bytes	9.001	•	•		•	

<b>Name</b>	<b>On/off switching</b>
<b>Function</b>	1 bit communication object to switch on or switch off the thermostat
<b>Description</b>	When a "1" is received through this object the thermostat switches on and starts to regulate the temperature When a "0" is received through this object the thermostat switches off
<b>Name</b>	<b>On/off feedback</b>
<b>Function</b>	1 bit communication object to read or notify the current status of the controller.
<b>Description</b>	When the output is switched off and it receives a switch on telegram, it sends a "1" through this object When the output is switched on and it receives a switch off telegram, it sends a "0" through this object
<b>Name</b>	<b>Set point temperature</b>
<b>Function</b>	2 bytes communication object to write a new set point controller.
<b>Description</b>	This communication object allows you to set a new temperature set point to the temperature controller. After a write, the controller will adjust its output to achieve and maintain the new temperature value of the room as programming.
<b>Name</b>	<b>Set point temperature feedback</b>
<b>Function</b>	2 bytes communication object for reading or notification of the current set point temperature.
<b>Description</b>	This communication object allows to read or notify the current set point temperature controller.
<b>Name</b>	<b>Room temperature feedback</b>
<b>Function</b>	2 bytes-Communication object to read or notify the room temperature.
<b>Description</b>	This object can read or notify the temperature that the sensor is measuring.

Do click on the settings button to set the thermostat configuration such as modes and limits.



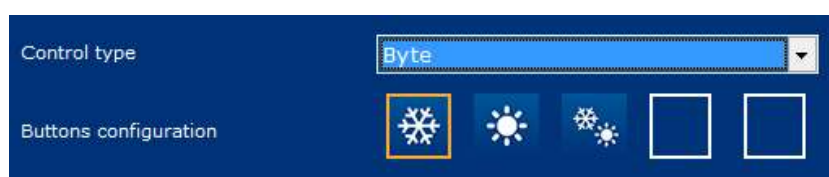
## General

Type of thermostat: Use the selector to choose if the icon will be associated to an external thermostat, temperature probe, gateway, etc., or to the internal thermostat of the device, in this case, the configuration is done in another window (see chapter Internal Thermostat).

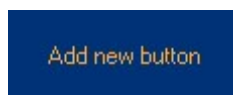
## Modes configuration

In the thermostat control dialogue, there can be two rows of icons that can be configured and parametrized for any programmer purpose, for example controlling modes, fan speeds, etc.

First select the behaviour of the group of icons and choose between the two types of controls: binary (individual bit communication objects for each icon) or byte (one byte communication object for the group of icons).



Do click on “add new button” and select one of the icons set, then do click into a box. Finally write the group addresses and values for each icon.



## Limits

It is possible to limit the set point temperature that can be modified by the user from the thermostat control dialogue. If not limited, the user will be able to send values from 0°C to 51°C. If limits are enabled, it is possible to set the minimum and the maximum value sent.



The parameter called Show limited setpoint allows to hide/show the real set point temperature chosen by the user. When this parameter is set to no, the set point scroll bar in the screen will show temperatures from 0°C to 51°C while internally is limiting the real set point according to the configuration. When the parameter is set to yes, the real limited set point temperature can be seen by the user.

### 3.10.5 Internal thermostat

Do click on the bottom of the edition window to program the internal thermostat of the Smart Touch.



When the *internal thermostat* is selected the following objects are shown:

**configuration**

Type: Thermostat

Name:

On/Off switching (1 bit):

On/Off feedback (1 bit):

Setpoint temperature (dpt9):

Setpoint temperature feedback (dpt9):

Room temperature (dpt9):

Room temperature feedback (dpt9):

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	On/off switching	1 bit	1.001	•		•		
1	On/off feedback	1 bit	1.001	•	•		•	
2	Set point temperature	2 bytes	9.001	•		•		
3	Set point temperature feedback	2 bytes	9.001	•	•		•	
4	Room temperature	2 bytes	9.001	•		•		
5	Room temperature feedback	2 bytes	9.001	•	•		•	

<b>Name</b>	<b>On/off switching</b>
<b>Function</b>	1 bit communication object to switch on or switch off the thermostat
<b>Description</b>	When a "1" is received through this object the thermostat switches on and starts to regulate the temperature When a "0" is received through this object the thermostat switches off
<b>Name</b>	<b>On/off feedback</b>
<b>Function</b>	1 bit communication object to read or notify the current status of the controller.
<b>Description</b>	When the output is switched off and it receives a switch on telegram, it sends a "1" through this object

	When the output is switched on and it receives a switch off telegram, it sends a "0" through this object
<b>Name</b>	<b>Set point temperature</b>
<b>Function</b>	2 bytes communication object to write a new set point controller.
<b>Description</b>	This communication object allows you to set a new temperature set point to the temperature controller. After a write, the controller will adjust its output to achieve and maintain the new temperature value of the room as programming.
<b>Name</b>	<b>Set point temperature feedback</b>
<b>Function</b>	2 bytes communication object for reading or notification of the current set point temperature.
<b>Description</b>	This communication object allows to read or notify the current set point temperature controller.
<b>Name</b>	<b>Room temperature</b>
<b>Function</b>	2 bytes communication object which is used to calculate the output
<b>Description</b>	The output value is calculated with the written temperature in this object . Furthermore, the written temperature in this object is shown in the Smarttouch. You can use a external temperature measured by a external thermostat placing here its group direction or, if you want to use the internal thermostat measured temperature, you must place as group direction "0/0/0"
<b>Name</b>	<b>Room temperature feedback</b>
<b>Function</b>	2 bytes-Communication object to read or notify the sensor temperature.
<b>Description</b>	This object can read or notify the temperature that the sensor is measuring.

Do click on the settings button to configure other parameters of the internal thermostat. The following pages are explained next.

## General

**Information in bottom panel:** This parameter allows configuring the behaviour of the screen's bottom panel.

- Only clock: The internal thermostat will not be accessible from the bottom panel and it will only show the time.
- Only thermostat: The bottom panel will show the thermostat information only.
- Clock and thermostat: Both clock and thermostat will be shown changing cyclically.

**Enable chronothermostats:** this parameter enables/disables the thermostat scheduling functionally in the time menu of the screen (see the device user manual).

**Lock / Unlock function:** If this parameter is enabled, it is possible to use an additional communication object that locks or unlocks the internal thermostat of the screen (lock = 1 / unlock = 0).

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Lock/unlock object	1 bit	1.001	•	•	•		•

The lock/unlock status is shown by a padlock at the bottom right corner of the thermostat control dialogue. When the thermostat is locked keeps working but it can only be controlled by bus telegrams.

### Modes configuration

**Controller enabled:** This parameter defines the type of controller for the internal thermostat. It can be heating, cooling or auto (both modes simultaneously).

**Controller change:** If auto mode is enabled, it is also possible to define whether the user can change manually between working modes by clicking on the corresponding icon or not (see the device user manual). Any time the working mode of the thermostat can be changed by bus telegrams by programming the bit group objects.

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Cool mode	1 bit	1.001	•		•		
1	Cool mode feedback	1 bit	1.001	•	•		•	
2	Heat mode	1 bit	1.001	•		•		
3	Heat mode feedback	1 bit	1.001	•	•		•	
4	Auto mode	1 bit	1.001	•		•		
5	Auto mode feedback	1 bit	1.001	•	•		•	

<b>Name</b>	<b>Cool mode</b>
Function	1 bit communication object to establish "Cooling" as the working mode of the controller.
Description	Writing a 1 in this order is possible to change the working mode to cooling. Writing a value of 0 does not have any action.
<b>Name</b>	<b>Cool mode feedback</b>
Function	Communication object 1 bit for reading or notification of the current status of this mode.
Description	Through this object is possible to read or notify if the controller is in cooling mode (1 = enabled cooling mode, 0= cooling mode disabled).
<b>Name</b>	<b>Heat mode</b>
Function	1 bit communication object to establish "Heating" as the working mode of the controller.
Description	Writing a 1 in this order is possible to change the working mode to heating. Writing a value of 0 does not have any action.
<b>Name</b>	<b>Heat mode feedback</b>
Function	Communication object 1 bit for reading or notification of the current status of this mode.
Description	Through this object is possible to read or notify if the controller is in heating mode (1 = enabled heating mode, 0= heating mode disabled).
<b>Name</b>	<b>Auto mode</b>
Function	1 bit communication object to establish "Auto" as the working mode of the controller.
Description	Writing a 1 in this order is possible to change the working mode to auto. Writing a value of 0 does not have any action.  In auto mode, the controller works both ways simultaneously. It automatically decides whether it is necessary to heat or cool. The temperature of the room is the same for both modes heat and cold.
<b>Name</b>	<b>Auto mode feedback</b>
Function	Communication object 1 bit for reading or notification of the current status of this mode.
Description	Through this object is possible to read or notify if the controller is in auto mode (1 = enabled auto mode, 0= auto mode disabled).

## Limits

For the internal thermostat, it is also possible to limit the set point temperature that can be modified by the user. If not limited, the user will be able to send values from 0°C to 51°C.

The screenshot shows a configuration screen for 'Setpoint limits'. At the top, there are two dropdown menus: 'Setpoint limits' is set to 'Absolute limits' and 'Show limited setpoint' is set to 'No'. Below these, the section 'Absolute limits' is expanded, showing two input fields: 'Minimum temperature' with the value '15' and 'Maximum temperature' with the value '25'.



There are two types of setpoint limitations that can be enabled individually or simultaneously:

- Absolute limits: Minimum and maximum limits that can be sent in any case.
- Relative limits: Defined by a deviation value above and below a temperature point that can be set by bus.

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Reference object	2 bytes	9.001	•	•	•		•

The parameter called Show limited setpoint allows to hide/show the real set point temperature chosen by the user. When this parameter is set to no, the set point scroll bar in the screen will show temperatures from 0°C to 51°C while internally is limiting the real set point according to the configuration. When the parameter is set to yes, the real limited set point temperature can be seen by the user.

### PID controller

In this window are programmed the Internal thermostat output communication objects, which are linked to the heating or cooling system (radiator, pump, electro-valve, fan-coil, etc.). The following control methods can be used individually or simultaneously: pwm (bit object) and continuous (byte object).

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Heat PWM	1 bit	1.001	•			•	
1	Cool PWM	1 bit	1.001	•			•	
2	Heat Continuous	1 byte	5.001	•			•	
3	Cold Continuous	1 byte	5.001	•			•	

Name	Heat PWM
------	----------

Function	1 bit communication object to control the heating system.
----------	---

Description	Through this object the heating system is controlled by a pulse width modulation control (PWM).
-------------	---

Name	Cold PWM
------	----------

Function	1 bit communication object to control the cooling system.
----------	---

Description	Through this object the cooling system is controlled by a pulse width modulation control (PWM).
-------------	---

Name	Heat continuous
------	-----------------

Function	1 byte communication object to control the heating system.
----------	--

Description	Through this object the heating system is controlled by a continuous PID control.
-------------	---

Name	Cold Continuous
------	-----------------

Function	1 byte communication object to control the heating system
----------	---

Description	Through this object the heating system is controlled by a continuous PID control .
-------------	--

## PID controller parameters

The thermostat temperature control method is based on a PI algorithm to obtain a better result and efficiency. Use the selector to set the better PI regulator parameters according to the heating/cooling system type:

- Radiant floor
- Water radiators
- Electric radiators
- Convection cooling
- Split systems
- PID disabled: 2 points control

It is also possible to tune in the PI regulator manually by selecting the type custom and using the Kp, Ki and T constants.

## Temperature sensor

Calibration: It is the group address to set the deviation value of the internal temperature probe. It must be set the difference between the temperature measured by the screen and the real temperature. Take into account that the deviation depends on the installation of the screen and backlight parameters.

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Calibration object (in °C)	2 bytes	9.001	•	•	•		•

External temperature probe: It is possible to use the measurement of an external temperature probe as reference for the internal thermostat or to obtain a weighted measure. Write the group address though which the external probe sends the measurement and select the weight of the value with the scroll.

Enable external sensor: Yes

External sensor address: Group address | 2/1/10

External measure weight: 0.5

Show temperature: Weighted measure

### 3.10.6 Scene

---

When the type *Scene* is selected the following options are shown:

The scene icon can be programmed to send a value from 1 to 64 (0x00 to 0x40) to the object. When receiving these values, KNX actuators will recall their internal scene which number corresponds to it.

The “save scene” checkbox allows to execute a save function with long press of the icon. After the time programmed, a value from 128 to 192 (0x80 to 0xC0) is sent to the object and the KNX actuators will save their current status in the internal scene which number corresponds to it.

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Scene save/recall	1 byte	18.001	●			●	

### 3.10.7 Colour picker

When the type *Colour picker* is selected the following objects are shown:

Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Red brightness value	1 byte	5.001	•			•	
1	Red brightness feedback	1 byte	5.001	•		•		•
2	Green brightness value	1 byte	5.001	•			•	
3	Green brightness feedback	1 byte	5.001	•		•		•
4	Blue brightness value	1 byte	5.001	•			•	
5	Blue brightness feedback	1 byte	5.001	•		•		•

**Name** Red brightness value

**Function** 1 bit communication object to set the level of red

**Description** Through this object you write the level of red for the controlled LED

**Name** Red brightness feedback

**Function** 1 bit communication object to read or notify the current level of red

**Description** Through this object you can read or notify the current level of red in the controlled LED

**Name** Green brightness value

**Function** 1 bit communication object to set the level of green

**Description** Through this object you write the level of green for the controlled LED

**Name** Green brightness feedback

**Function** 1 bit communication object to read or notify the current level of green

**Description** Through this object you can read or notify the current level of green in the controlled LED

**Name** Blue brightness value

**Function** 1 bit communication object to set the level of blue

**Description** Through this object you write the level of blue for the controlled LED

**Name** Blue brightness feedback

**Function** 1 bit communication object to read or notify the current level of blue

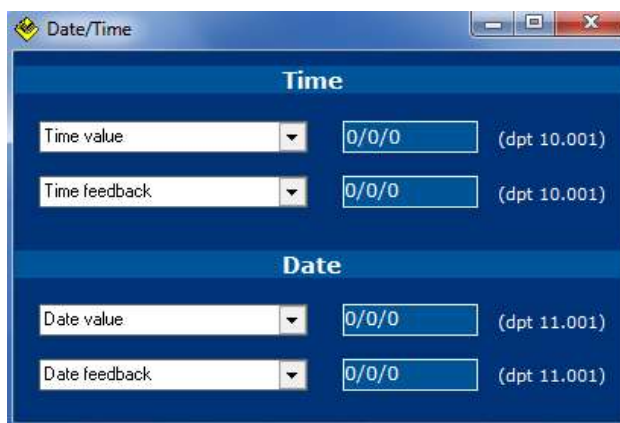
**Description** Through this object you can read or notify the current level of blue in the controlled LED

## 3.11 Other functions

### 3.11.1 Date and time

Do click on the project menu -> Date/time to show the settings window.

The Smart Touch can work as date and time master or slave in the KNX bus by sending or receiving the corresponding data by telegrams from any other KNX devices.



Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Time of day (input)	3 bytes	10.001	•		•		
1	Time of day feedback (output)	3 bytes	10.001	•			•	
2	Date (input)	3 bytes	11.001	•		•		
3	Date feedback (output)	3 bytes	11.001	•			•	

Write down the group addresses for objects 0 and 2 if the Smart Touch must be a date/time slave in the KNX bus. On the other hand, complete the group addresses for objects 1 and 3 if the Smart Touch must be a date/time master. In this case, use the time menu (options) to set the current configuration (see the device user manual).

### 3.11.2 Technical alarms

Do click on the following button or the in menu *project -> technical alarms* to display the technical alarms settings window. In this window should be filled the group address list of alarms.



When receiving an alarm, the device displays a warning icon on the screen and can send a notification if it has been uploaded to the cloud for remote control (see user manual). If the device receives a "1" in the group address the alarm is triggered, if it receives a "0" the alarm is deleted.

Objet	Name	Length	DPT	Flags				
				C	R	W	T	U
X	Technical alarms	1 bit	1.001	•		•	•	

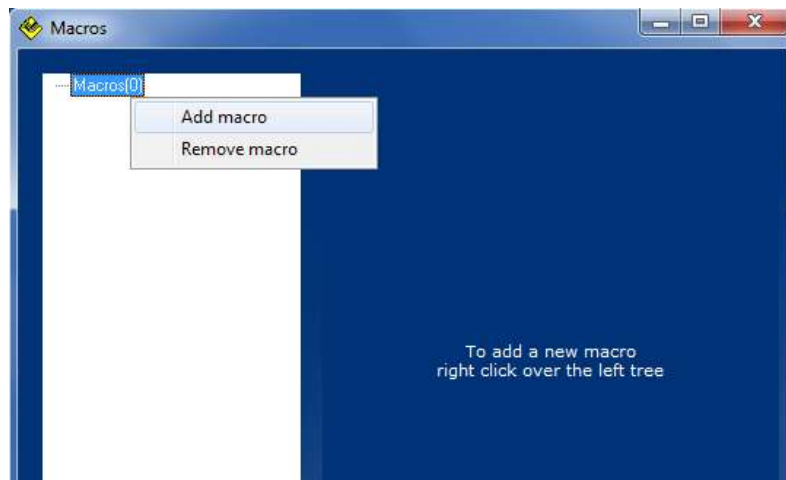
### 3.11.3 Macros

---

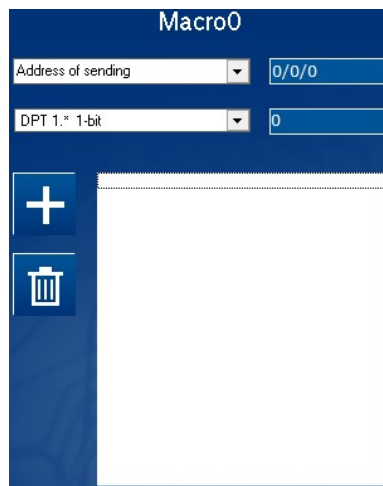
A macro can be an action or group of actions that are executed simultaneously in the home automation system. These macros can be scheduled weekly in the time menu of the device (see the device user manual) and they are programmed easily as explained next.

Do click on the project menu -> Macros to show the settings window.

Then, do right click over the tree on the left and add macro.



Assign a name/description to the Macro and then write the group address and the size of data sent. Do click on the plus button to add new events and the bin to delete. It is also possible to program delays between telegrams by selecting the corresponding function.



### 3.11.4 Appearance change

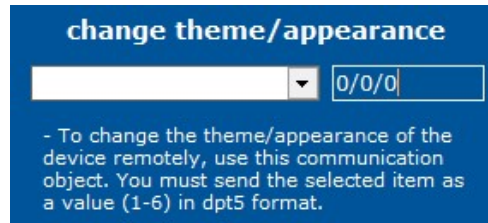
---

The appearance of the screen icons and colours can be changed by selecting different themes according to the customer likes. It can change automatically by programmed hours (see the device user manual) or by bus telegrams from any other device of the KNX system.

Do click on the following icon at the top right of the main window of SIDEKNX to program the corresponding group address.



Writing a value from 1 to 6 to this communication object will change the current theme of the Smart Touch to the desired one.



Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Theme change (appearance)	1 byte	5.001	•		•		

### 3.11.5 Administrator password

---

The options menu, which allows configuring the appearance, brightness, language, etc., can be hidden only for administrators.

Do click on the following icon at the top right of the main window of SIDEKNX to program the corresponding password for enabling/disabling it.



This function is enabled just by writing a password. If the menu is blocked (administrator mode enabled), press in the centre of the Smart Touch main window for 5 seconds approximately and use the keyboard to enter the access password.

Leave the admin password empty to disable this function. If the menu is unblocked (administrator mode disabled) any type of user will be able to change any configuration. In this case, it can be used the bar at the top of the main window to slide down the menu.



### 3.11.6 Default actions

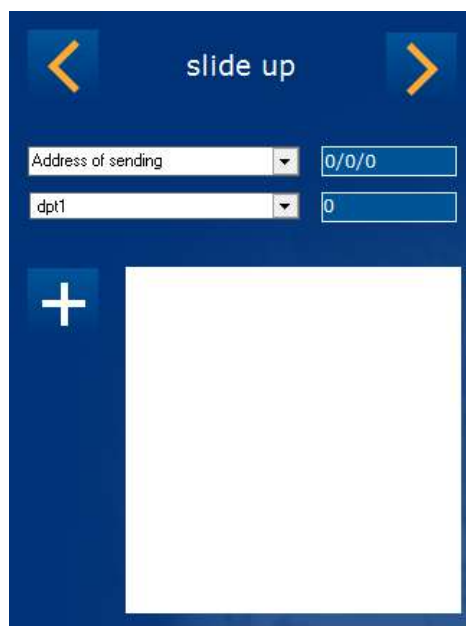
---

The Smart Touch allows executing some programmed actions by gesture control (sliding up or down) when the screen is off or locked.

Do click on the following icon at the top right of the main window of SIDEKNX to program the actions that will be executed for each gesture control.



Write the sending group address, set the size of the telegram and the value sent. It is also possible to program a sleep function in order to wait a time between programmed actions. Finally do click on the plus button to add the command to the macro.



### 3.11.7 Values in labels

---

It is possible to show values received from the bus in the name labels of each icon, for this it must be used the following formats in the field "name" of the icon:

- %d** 1 bit (dpt 1.\*)
- %hu** 8 bit unsigned (dpt 5.\*)
- %hd** 8 bit signed (dpt 6.\*)
- %u** 2 bytes unsigned (dpt 7.\*)
- %d** 2 bytes signed (dpt 8.\*)
- %f** 2 bytes float (dpt 9.\*)

The label will show the value received through the feedback object of the icon formatted with the operators above.

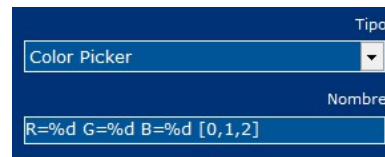
In the case of 2-byte floating, it can be defined the number of decimal places by adding ".x" for example: **%.1f** = 1 decimal or **%f** = 3 decimals. In the example 1, a button displays the value received in lux (dpt9) and the text "lux" is added.



If the icon type has more than one feedback object then you may select the object to show adding "[x, y, z]" at the end of the label. In the example 2, a color picker icon type shows its three values:



Example 1

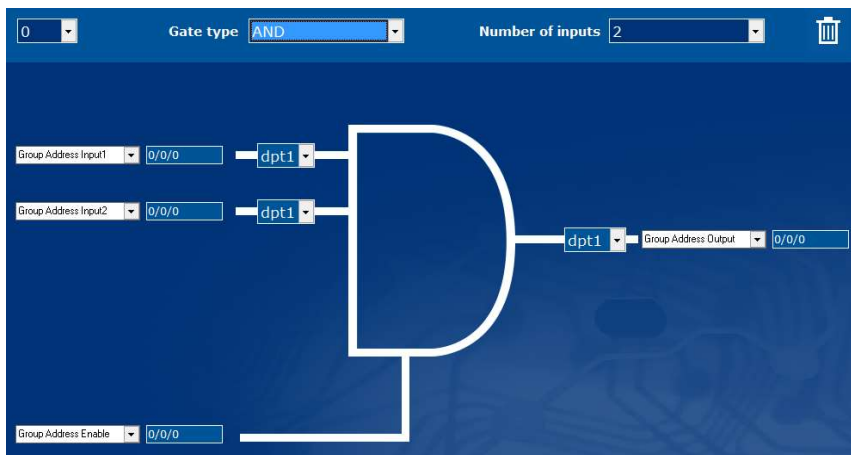


Example 2

### 3.11.8 Logic gates

This device incorporates an advanced Arithmetic and Logic Unit (UAL) that allows to do complex logic operation using internal or external variables with an intuitive and visual programming.

Each logic gate can include up to four inputs and one output being possible to select the size and format of each communication object and an auxiliary object to enable/disable the gate. There are up to 100 logic gates that can be linked by group addresses and each gate can be triggered or enabled/disabled by an auxiliary object.



Use the above options to select the number of gate among the 100 available, the type of operation and the number of inputs. The communication object at the bottom allows to enable/disable the gate. The result of a logic gate is sent to the bus when a new value is received in its inputs or when the gate is enabled.

### 3.11.9 Scripts

This device incorporates an advanced scripting programming method developed with its own programming language similar to other programming languages as C. The scripts allows the programmer to run a simple mood or scenario or to develop his own advanced program execution that can be run from the bus, sending telegrams or receiving parameters, counting, operating, etc.

The programming language used in the scripts is similar to other languages, for example, a semicolon indicates the end of every sentence. For better understanding each sentence can be written in a new line and we can use tabulations. There are several functions and reserved words that are explained next:

### - Parameter received from bus

#### **param**

It is the value received from the knx bus through the group address associated to the script. It can be a bit, a byte or 2bytes.

It can be used in operations, functions and other commands, even between operators of different size, for example:

```
...  
var1=(param*2)+234;  
...
```

We can also assign to it a new value directly or as a result of an operation during the script execution, for example:

```
...  
var1=(param*2)+234;  
param=0;  
...
```

The parameter will hold the value received or assigned until the end of the script execution. When receiving a floating point 2bytes value, the real value taken by the “param” is the binary one. Floating point operations are not available.

### - Variables

#### **varX, g\_varX**

There are up to 10 local variables “varX” in each script and 10 global variables “g\_varX”. The variables can be used in any operation, function and other commands, even between operators of different size. It is not necessary to declare or initialize them because with every script execution they take the value 0.

```
| var1=(param*2)+234;  
| g_var2=var2+var1;
```

In the same way as the parameter, we can assign to them a new value directly or as a result of an operation during the script execution. The variables hold the value assigned until the end of the script execution too.

The variables “varX” are local of its script and they can be used only in its script. The variables “g\_varX” are global and they can be used in every script, so if we need use a value from one script into another, we must send it through a parameter “param” or use a global variable.

### - Arithmetic and comparison operators

There are two types of operators: arithmetic and comparison.

**Arithmetic:** =, +, -, \*, / (assignment, addition, subtraction, multiplication and division).

These arithmetic operators can be used in any sentence with variables, constant values and also the parameter. The result of an arithmetic operation can be assigned to the parameter or an internal variable by the operator “=”. Parenthesis can be used to group several operations as common arithmetic rules.

```
var1=5;  
var2=(param/2)+(var1*2);  
var3=var3+var2;
```

Notice that all arithmetic operations are done with integers, floating point operations are not possible.

**Comparison:** ==, >=, <=, >, <, != (equal, major or equal, minor or equal, major, minor, different).

Comparison operators are used with the conditional function "if" as explained next.

#### - Functions: GOTO and IF

**Goto X;** Go to a line function

The "goto" function makes the program execution continue from a determined line that it is identified by a label "X" chosen by the programmer.

A label is defined with a combination of letters and/or numbers, but the first character must be a letter always, and the label finishes with ":". For example:

```
...  
goto next;  
    var1=0;  
    ...  
    Var10=0;  
next:  
    var1=param+1;  
...
```

The program continues from the label and the sentences after the "goto" function are not executed

**If (A X B)** Conditional function "if"

The conditional function "if" allows the programmer to execute or not one or more sentences of the script depending on the result of a comparison.

A and B are internal variables, constants or the parameter while X is a comparison operator.

If the comparison operation is true, the next line after the "if" is executed. If the comparison operation is false, the program jumps the sentence. For example:

```
If (param>100)  
    var1=param;  
var1=0;
```

If the parameter it is lower or equal to 100 the next instruction it is not executed

In the previous example, var1 will take the value of the parameter received only if it is greater than 100, if not var1 takes the value 0.

When the programmer needs to execute more than one sentence depending on a condition he can repeat the same “if” function, but it is better to use the opposite condition in combination with a “goto” function, for example:

```
If (param>100)
    var1=param;
If (param>100)
    var2=param*2;
If (param>100)
    var3=param*3;
If (param<=100)
    var1=0;
If (param<=100)
    var2=0;
If (param<=100)
    var3=0;
```

The previous script is the same than the following one using the opposite conditions:

```
If (param<=100)
    goto next;
    var1=param;
    var2=param*2;
    var3=param*3;
next:
If (param>100)
    goto next2;
    var1=0;
    var2=0;
    var3=0;
next2:
```

If the parameter it is greater than 100 the “goto” is not executed and but following instructions yes.

The programmer can use nested “if” functions to implement a composed condition. In the following example the script checks if the parameter value is between 5 and 10. If the condition is true the parameter is saved in the variable 1, if not the variable is 0.

```
If (param<5)
    If (param>10)
        var1=param;
var1=0;
```

If the parameter it is not lower than 5, the program jumps directly to the sentence var1=0.

When we use nested “if” functions, we can think that if the first “if” condition it is false, the following sentence it is not executed, so the second “if” condition it is not evaluated. It does not work like this.

In the previous example, the sentence that follows the first “if” function finishes in the first “;”. This means that if the parameter it is not lower than 5 the program continues with var1=0. We can understand the script better if it is written like this:

```
If (param<5)
    If (param>10) var1=param;
var1=0;
```

Tabulations and carriage returns are not necessary; they are used for better understanding of the script.

#### - Direct commands

There are several direct commands that are executed with reserved words as explained next:

**Sleep X;** - Being X the time in seconds from 0 to 65535. This command stops the execution of the script during the time configured.

#### - Sending telegrams to the bus

**Send\_telegram (X/X/X,dpt,value,command);**

X/X/X – Is the group address form 0/0/1 to 31/7/255.

dpt – Indicates the length of the data sent: dpt1, dpt5, dpt7 and dpt9.

value – Is the value sent to the bus.

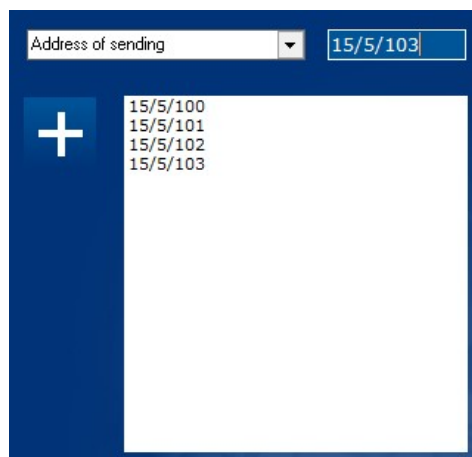
command – Is the command of the telegram that can be: write or read.

```
...
send_telegram(1/1/10,dpt5,150,write);
send_telegram(1/2/15,dpt9,21.50,write);
...
```

### 3.11.10 Filtering group addresses

---

It is possible to filter a list of group addresses in order to avoid any sending to the bus, for example when linking an output to an input of two logic gates. Do click on Project -> Filtered and write the group addresses to be filtered in the list.



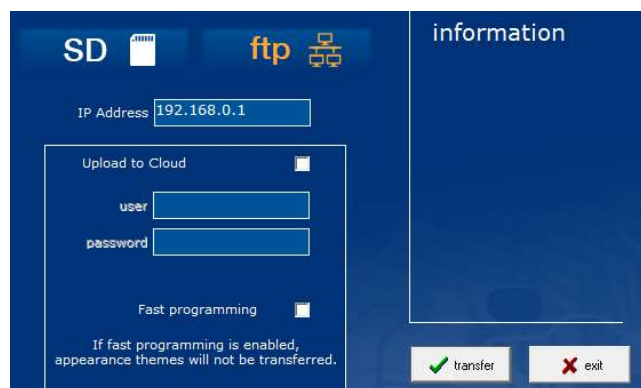
### 3.12 Downloading the project

---

The SIDE KNX allows programming Ingenium KNX web servers and touchscreens. In the case of the Ref. 342100/343100, it is programmed by connecting the computer to the same Wi-Fi network.



To program the Ingenium device, first save the project and then press the *transfer project* button from Toolbar panel or select *Project > Transfer* and the following window will be showed. Then do click on the corresponding button to select the ftp downloading method.



Enter the IP address of the screen (see the device user manual). The “upload to cloud” function is used only when programming the Smart Plus (that incorporates web server).

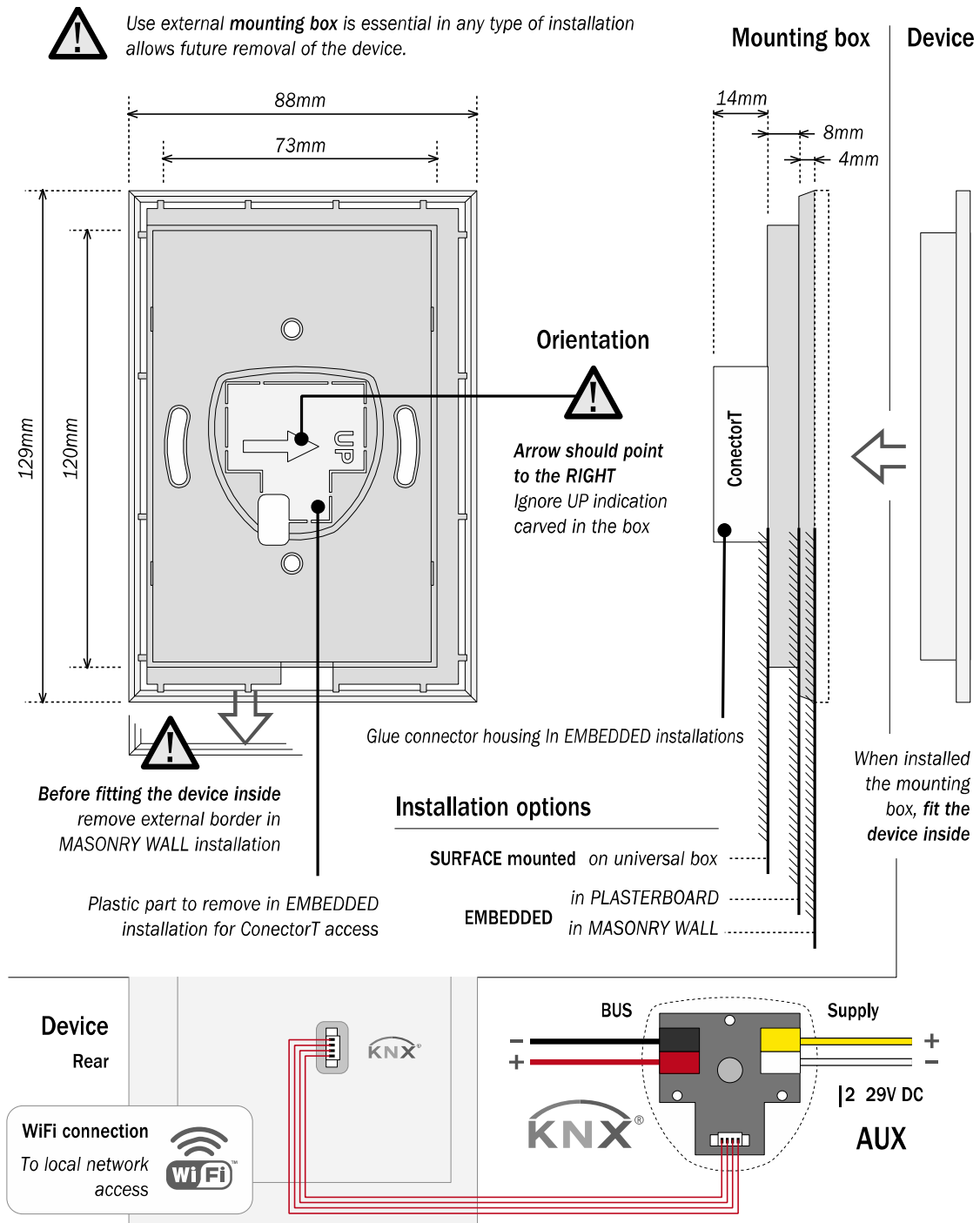


**Note:** The device and the programmer computer must be connected to the same network and configured in the same subnet. Check the correct connection of the screen by using ping network command before downloading.

Do click on “fast programming” to activate it if you do not want to change the appearance themes of the screen.

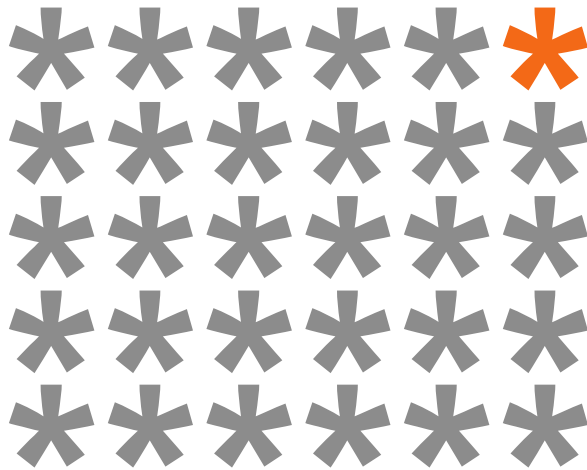
Do click on transfer to complete the process. The status will be shown in the information window. After the downloading process the screen will reboot automatically.

## 4 Installation



Feed low voltage lines (bus and inputs) in separate ducting to that of power (230V) and outputs to ensure there is enough insulation and avoid interferences.

Do not connect the main voltages (230V) or any other external voltages to any point of the bus or inputs.



KNX products by ingenium



**Ingenium, Ingeniería y Domótica S.L.**

Parque Tecnológico de Asturias, Parcela 50

33428 Llanera, Asturias, Spain

T (+34) 985 757 195

tec@besknx.com

www.besknx.com

www.ingeniumsl.com

*Liability limitation: The present document is subject to changes or excepted errors. The contents are continuously checked to be according to the hardware and software but deviations cannot be completely excluded. Consequently any liability for this is not accepted. Please inform us of any suggestion. Every correction will be incorporated in new versions of this manual.*

Manual version: v1.5.1