

Graphic controllers

DS1XXXXX

# Programming manual



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

# Index

<b>1</b>	<b>INTRODUCTION</b> .....	<b>3</b>
<b>2</b>	<b>FIRST STEPS</b> .....	<b>4</b>
<b>3</b>	<b>GENERAL</b> .....	<b>5</b>
3.1	DEVICE .....	5
3.2	FIRMWARE UPDATE.....	5
3.3	GROUP ADDRESSES IMPORT .....	6
3.4	INDIVIDUAL ADDRESS ASSIGNMENT .....	7
<b>4</b>	<b>PROGRAMMING</b> .....	<b>8</b>
4.1	ADDING DRAWINGS TO THE PROJECT .....	8
4.2	ADDING ICONS.....	8
4.3	TYPE OF ICONS .....	9
4.3.1	<i>Button</i> .....	10
4.3.2	<i>Dimmer</i> .....	11
4.3.3	<i>Shutter/blind</i> .....	12
4.3.4	<i>Thermostat</i> .....	12
4.3.5	<i>Scene</i> .....	15
4.3.6	<i>IP camera</i> .....	15
4.3.7	<i>Colour picker</i> .....	16
4.4	OTHER FUNCTIONS .....	16
4.4.1	<i>Date and time</i> .....	16
4.4.2	<i>Technical alarms</i> .....	17
4.4.3	<i>Intrusion alarm</i> .....	17
4.4.4	<i>Macros</i> .....	19
4.4.5	<i>Timings</i> .....	20
4.4.6	<i>Values in labels</i> .....	21
4.4.7	<i>Logic gates</i> .....	21
4.4.8	<i>Scripts</i> .....	22
4.4.9	<i>Filtering group addresses</i> .....	26
4.4.10	<i>Screensaver</i> .....	26
<b>5</b>	<b>VIDEO INTERCOM</b> .....	<b>27</b>

# 1 Introduction

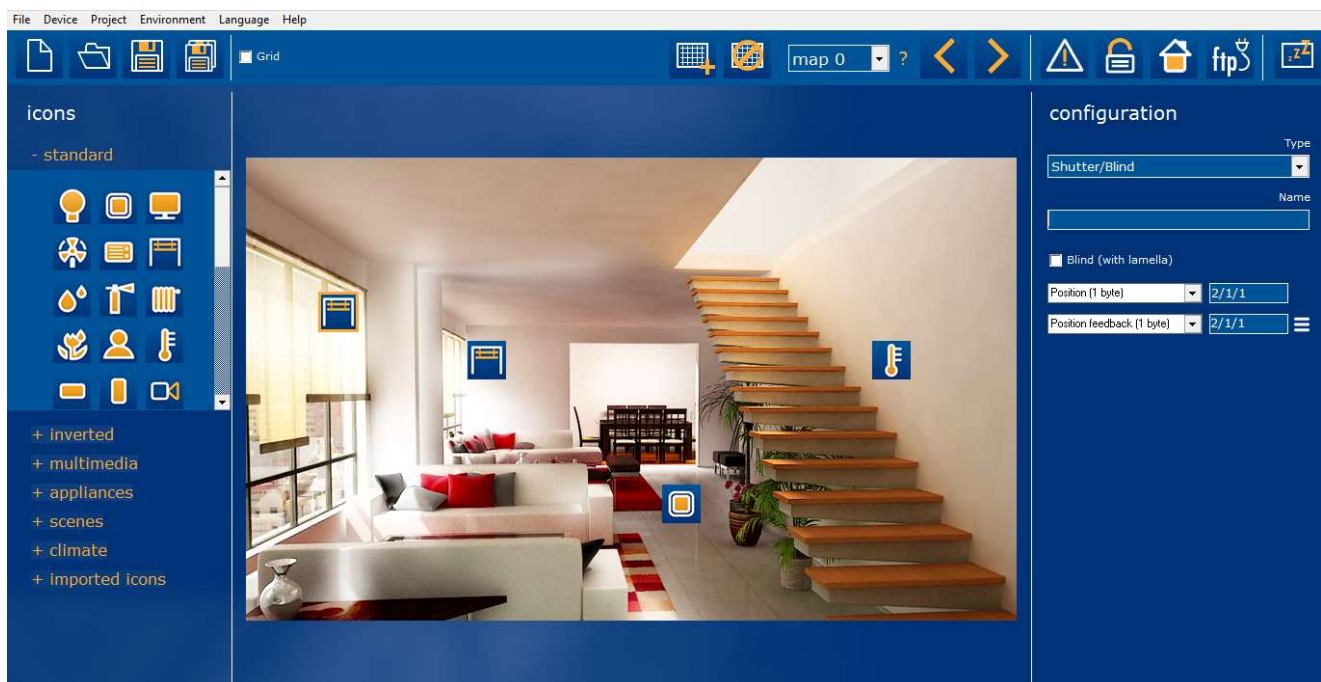
---

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.



## 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.

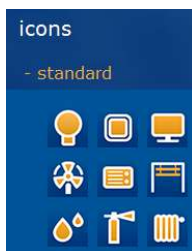
1. First select the type of project according to the target device by doing click on the button “new”.



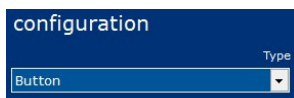
2. Select the background of the visualization. You can upload any jpg or bmp image, either a photograph or 3D design.



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.

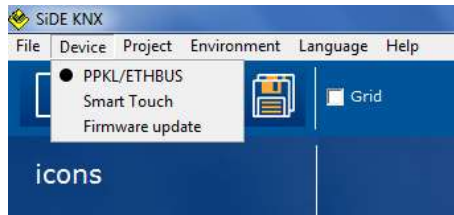
## 3 General

---

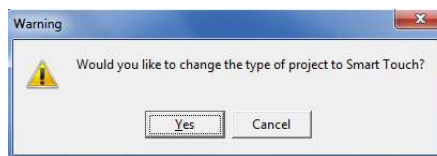
### 3.1 Device

---

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 VIIP or PPKLX or ETHBUS, it must be selected in the corresponding menu: Device -> PPKL/ETHBUS.



### 3.2 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 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.



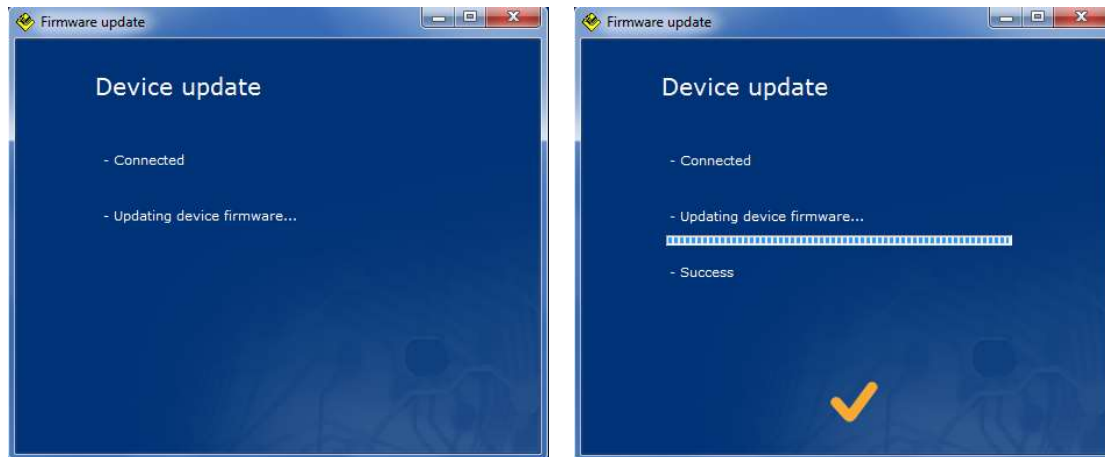
The SIDEKNX software will search for it and read the type of device.



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.

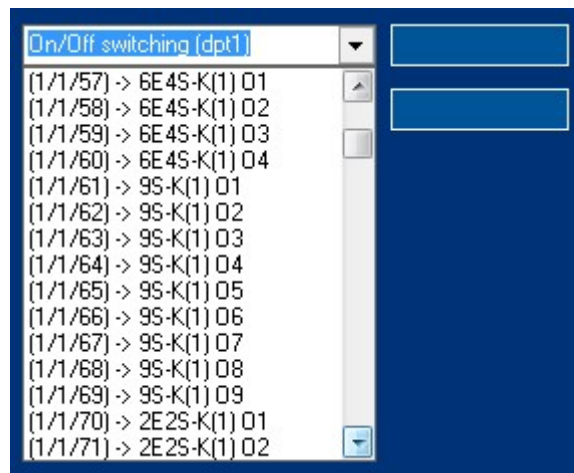


### 3.3 Group addresses import

---

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 delete 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.

### 3.4 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.

## 4 Programming

---

### 4.1 Adding drawings to the project

---

You can add and remove drawings or images (usually a picture or a 3D plan of the rooms of the house or areas of the installation) using the following icons.



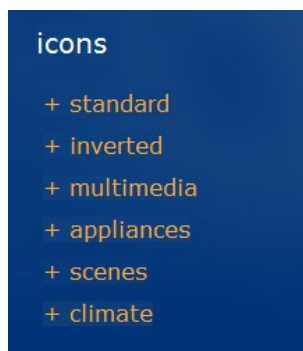
You can use the arrows or the selector on the right to navigate between the drawings of a project.

The question mark is used to know the path of the picture. When you move a project to another folder or to another computer, the locations of the pictures can be different and you must restore them. To do this, first search the file missed looking the file path in the question mark and add it again selecting the right file path.

### 4.2 Adding icons

---

Add icons to the current drawing with the tree on the left. Open each category and press over the icon, then do click on the drawing to place the icon where needed.





You can activate the grid and select the step to place icons aligned easily.



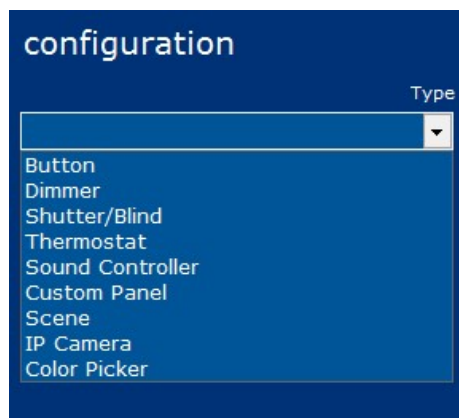
Do right click over the icon to copy, delete or change icon and select the new one if you want to change the icon appearance without changing its properties.




### 4.3 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 and their communication objects are explained next.



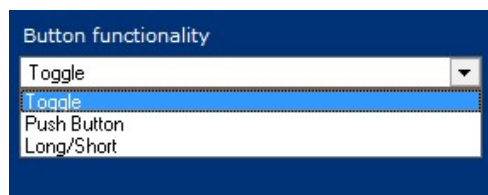
Use the name field to write the label that the icon will have on the screen and use write also the group addresses for each communication object.

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



### 4.3.1 Button

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



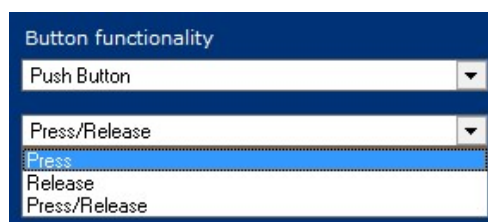
#### TOGGLE BUTTON

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:

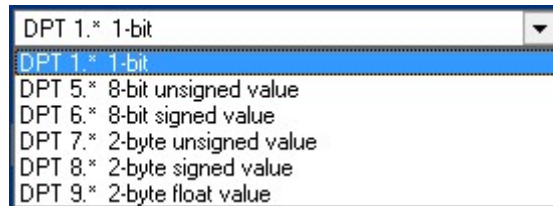
Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Button toggle: action	1 bit	1.001	•			•	
1	Button toggle: feedback	1 bit	1.001	•		•		•

#### 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:



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.

Detection time (ms.)

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	•		•		•

### 4.3.2 Dimmer

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

**configuration**

Type:

Name:

On/Off switching (dpt1)

On/Off feedback (dpt1)

Brightness value (dpt5)

Value feedback (dpt5)

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	•		•		•

### 4.3.3 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 bit	1.001	•			•	
1	Position feedback	1 bit	1.001	•		•		•
2	Lamella position	1 byte	5.001	•			•	
3	Lamella feedback	1 byte	5.001	•		•		•

### 4.3.4 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 value	2 bytes	9.001	●			●	
3	Set point temperature feedback	2 bytes	9.001	●		●		●
5	Room temperature feedback	2 bytes	9.001	●		●		●

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.

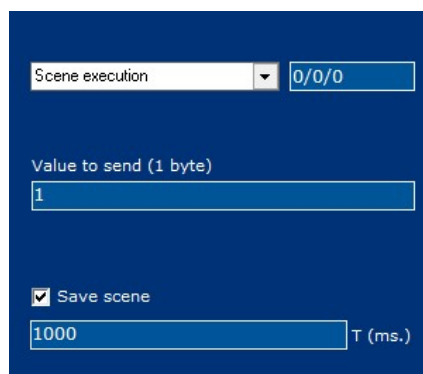


The parameter called Show limited setpoint allows to hide/show the real set point temperature chosen by the user. When this parameter is set to no, the set point scroll bar in the screen will show temperatures from 0°C to 51°C while internally is limiting the real set point according to the configuration.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

### 4.3.5 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.

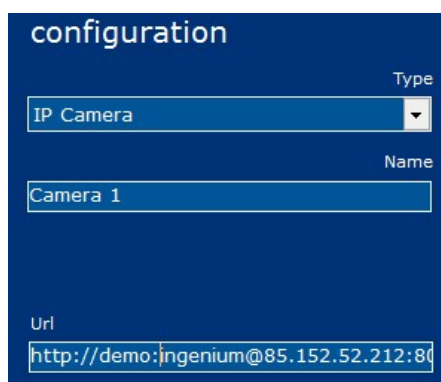
The “save scene” checkbox allows to execute a save function with long press of the icon. After the time programmed, a value from 128 to 192 (0x80 to 0xC0) is sent to the object and the KNX actuators will save their current status in the internal scene which number corresponds to it.

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

### 4.3.6 IP camera

---

If the IP camera icon type is selected, it will appear a field to enter the URL to access the images captured by the camera.



The URL must be written with the following format:

`http://usuario:password@IP:puerto/path`

For example: `http://demo:ingenium@85.152.52.212:8088/image.jpg`. The path where the camera saves the screenshots captured should be checked in the manufacturer documentation for each particular model.

### 4.3.7 Colour picker

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

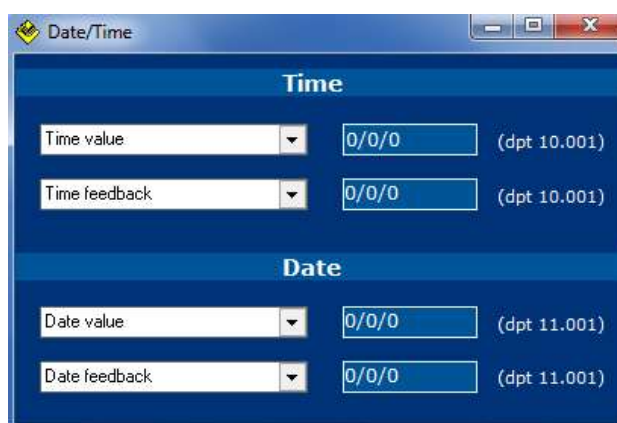
Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Red brightness value	1 byte	5.001	•			•	
1	Red brightness feedback	1 byte	5.001	•		•		•
2	Green brightness value	1 byte	5.001	•			•	
3	Green brightness feedback	1 byte	5.001	•		•		•
4	Blue brightness value	1 byte	5.001	•		•		•
5	Blue brightness feedback	1 byte	5.001	•		•		•
6	On/off switching	1 bit	1.001	•			•	
7	On/off switching feedback	1 bit	1.001	•		•		•

## 4.4 Other functions

### 4.4.1 Date and time

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

The device 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.



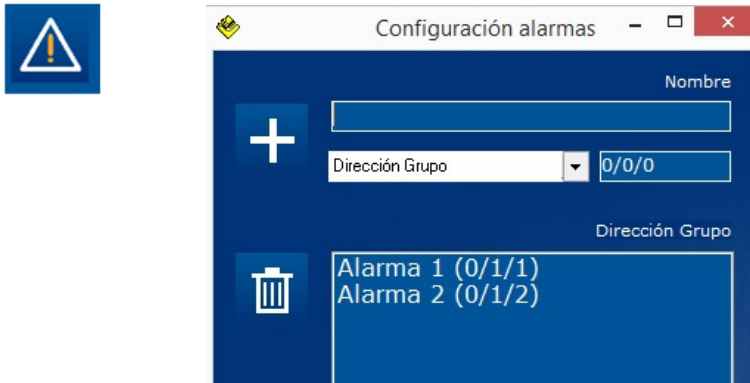
Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Time of day (input)	3 bytes	10.001	•		•		
1	Time of day feedback (output)	3 bytes	10.001	•			•	
2	Date (input)	3 bytes	11.001	•		•		
3	Date feedback (output)	3 bytes	11.001	•			•	



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).

#### 4.4.2 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.

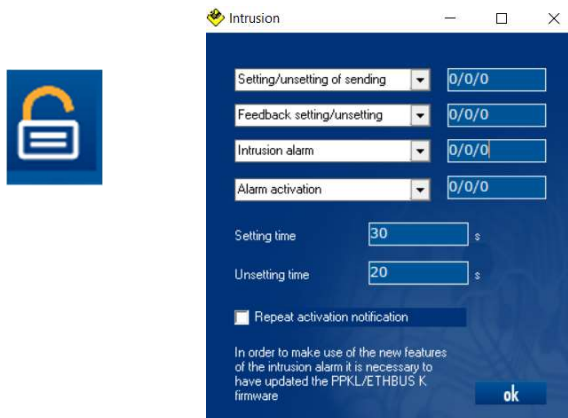


When receiving an alarm, the device displays a warning icon on the screen and can send an email or a notification if the service have been configured. If the device receives a "1" in the group address the alarm is triggered, if it receives a "0" the alarm is deleted.

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

#### 4.4.3 Intrusion alarm

Do click on the following button or the in menu *project* -> *intrusion alarm* to display the intrusion settings window.



Object	Name	Length	DPT	Flags				
				C	R	W	T	U
0	Setting/unsetting of sending	1 bit	1.001	•				•
1	Feedback setting/unsetting	1 bit	1.001	•				•
2	Intrusion alarm	1 bit	1.001	•		•		
3	Alarm activation	1 bit	1.001	•				•

The sending setting/unsetting group address is the one which sets/unsets the alarm. The device sends information about its state through the feedback setting/unsetting. The intrusion alarm object allows to trigger the alarm, shows a warning on the screen and sends email or notification if the service has been configured. If the alarm is triggered the alarm activation group address is activated.

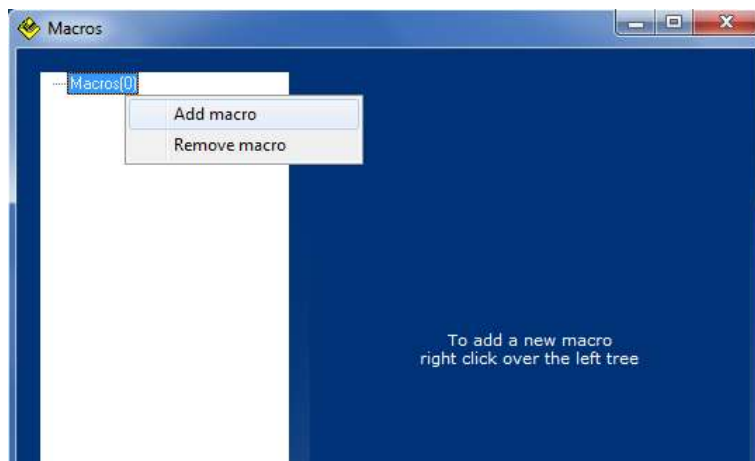
#### 4.4.4 Macros

---

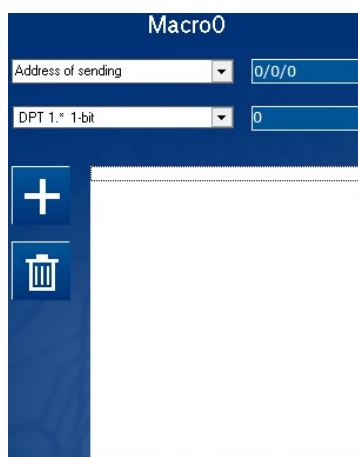
A macro can be an action or group of actions that are executed simultaneously. These macros can be scheduled weekly in the time menu of the device (see the device user manual) and they are programmed easily as explained next.

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.



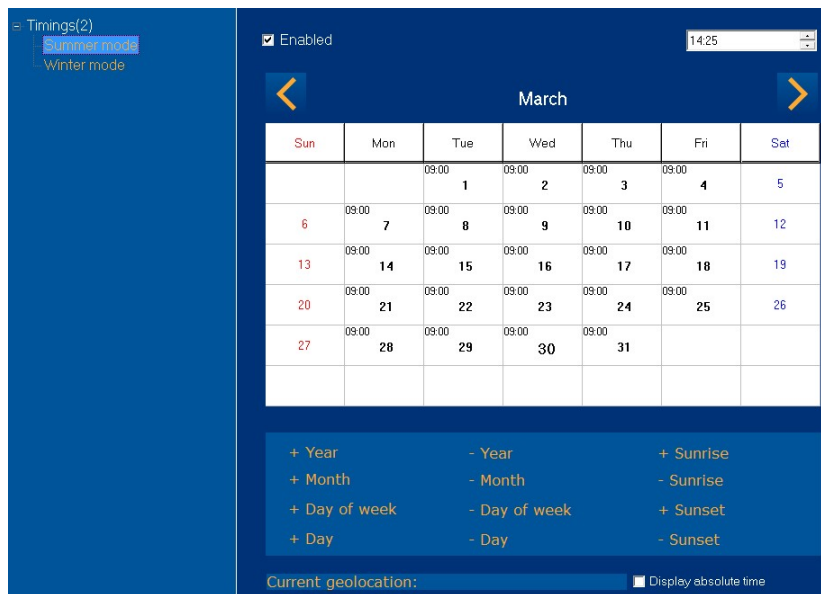
Assign a name/description to the Macro and then write the group address and the size of data sent. Do click on the plus button to add new events and the bin to delete. It is also possible to program delays between telegrams by selecting the corresponding function.



#### 4.4.5 Timings

The device allows to program annually macro execution that will be invisible and non-modifiable by the user if necessary. This does not exclude that the user can create their own macros and timers on the device itself but can be useful if you want the user not to change a setting done by the programmer.

After you create macros, click the menu Project -> Timings and select the macro to temporize to display the calendar.



First, use the clock on the top right to set the execution time and then the buttons at the bottom to add or delete the year, month, day of week or the specific day selected.

For example, if you want to temporize a macro to run every day of the year from Monday to Friday at 9 am, proceed as follows:

- 1- Set on the clock 09:00
- 2- Press + Year
- 3- Select and press any Saturday - Day of the week
- 4- Select and press any Sunday - Day of the week

At any time, it is possible to see the result of the scheduled macro on the calendar.

To configure a timing dependent of sunrise or sunset, first select the geolocation where the device will be installed, the VIIP will get it when is connected to the internet through the IP, then use the clock to set the hours and minutes to be added or subtracted and finally do click on +/- Sunrise or +/- Sunset . You can see the time at which the macro will be executed for the selected geolocation by clicking "display absolute time".

#### 4.4.6 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:

<b>%d</b>	1 bit (dpt 1.*)
<b>%hu</b>	8 bit unsigned (dpt 5.*)
<b>%hd</b>	8 bit signed (dpt 6.*)
<b>%u</b>	2 bytes unsigned (dpt 7.*)
<b>%d</b>	2 bytes signed (dpt 8.*)
<b>%f</b>	2 bytes float (dpt 9.*)

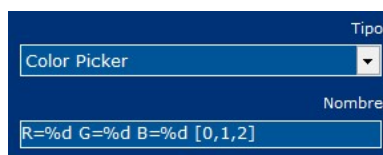
The label will show the value received through the feedback object of the icon formatted with the operators above.

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 **%3f** = 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 notification object (input) 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

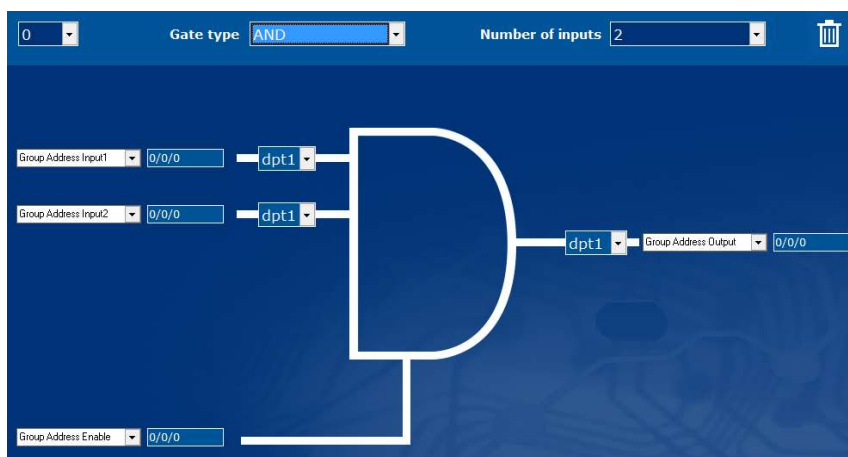


Example 2

#### 4.4.7 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.

#### 4.4.8 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;
If (param<=100)
    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.

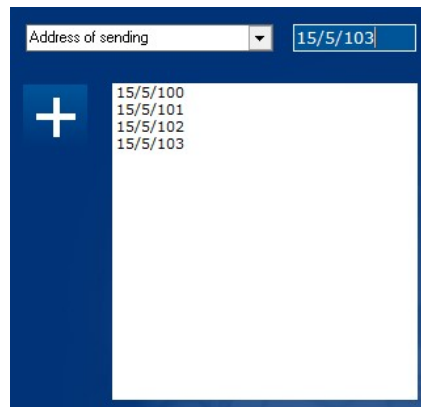
value – Is the value sent to the bus.

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

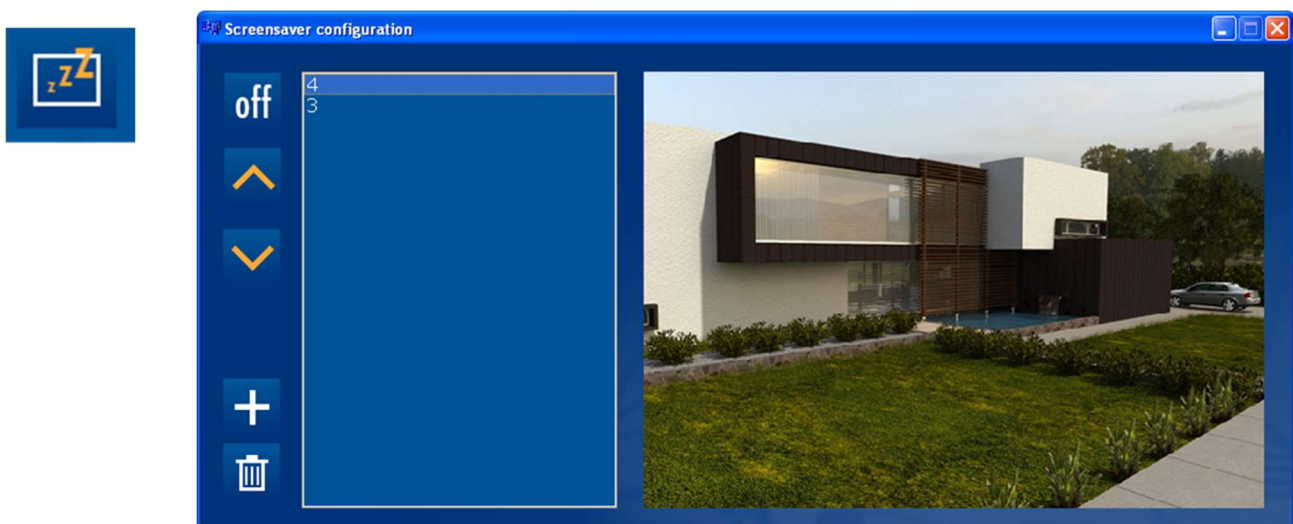
```
...
send_telegram(1/1/10,dpt5,150,write);
send_telegram(1/2/15,dpt9,21.50,write);
...
```

#### 4.4.9 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.



#### 4.4.10 Screensaver




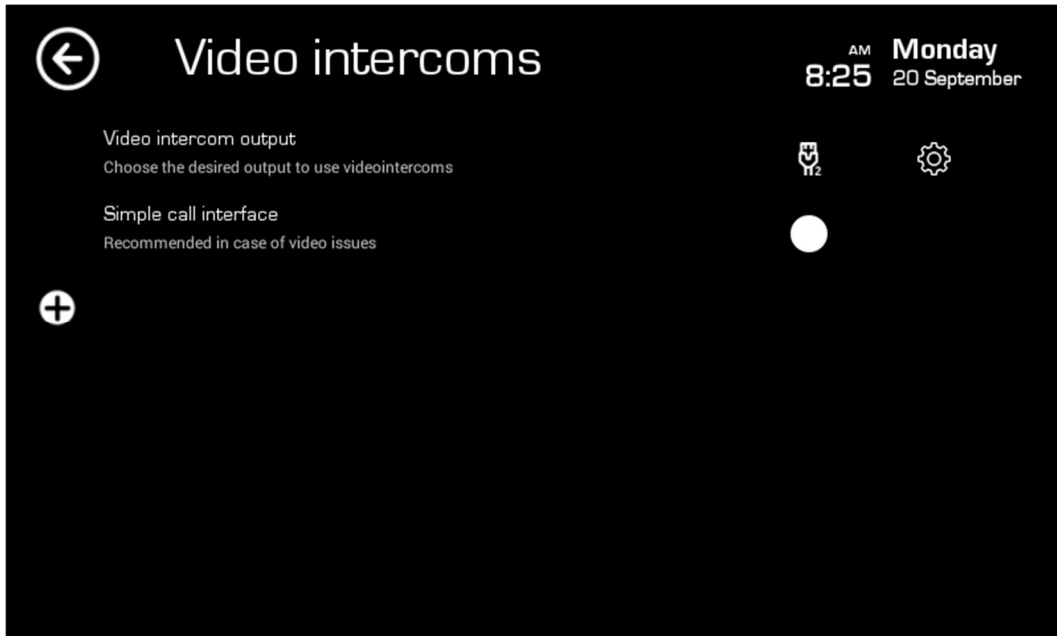
You can show pictures as screensavers in touchscreens, for that, you must first save the project and after that press the #19 button of the *Toolbar panel* to a window will be displayed. In this window you can add or delete the pictures for the screensaver. To add a picture to the list you must press the button with a plus sign on it and if you want to remove it from the list you must first select the picture in the list and then press the button with a bin icon on it. The arrows are for selecting a picture in the list which is displayed on the right.

Press the *Off/On* button in the *Screensaver configuration window* to choose if the screensaver pictures will be transferred with the project or not. If the button shows the *Off* text the pictures will not be transferred with the project.

## 5 Video intercom

---

For the video intercom configuration, on the VIIP, you must access the Settings / Connections / Video intercom configuration menu. To receive calls from the video intercom, click on the  button to add a video intercom and configure it.



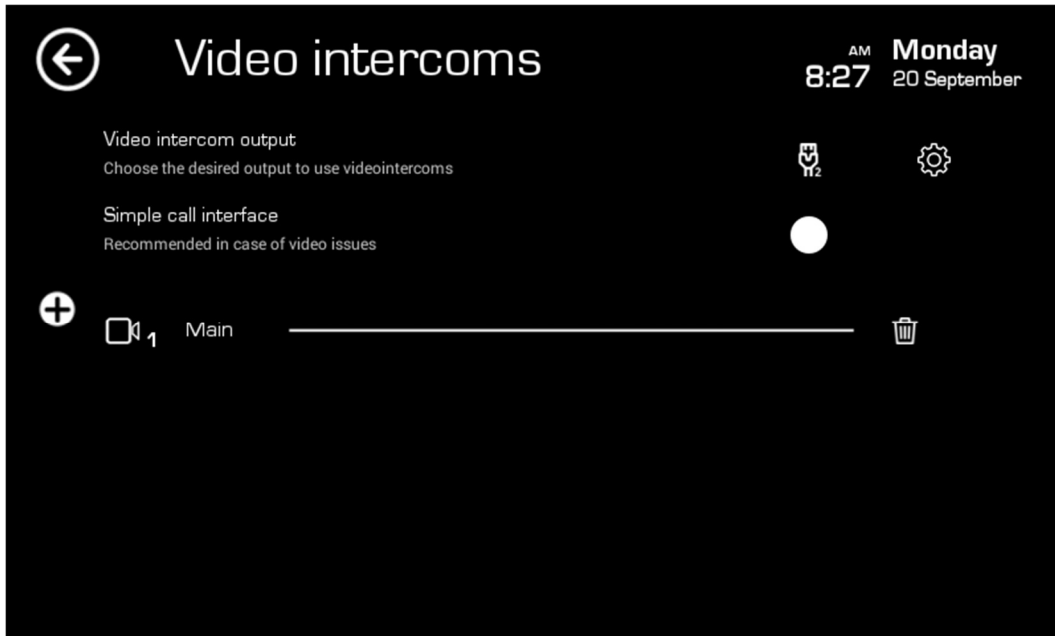
In the configuration window you must enter the following data:

- Name
- IP (from the video intercom)
- Extension (enter any 3-digit number eg: 101)
- Door opener code, which has to coincide with the one configured on the video intercom menu (some panels have restrictions in this code that must be taken into account when configuring).



The screenshot shows the 'Video Intercom' configuration window. At the top left is a back arrow icon. The title 'Video Intercom' is centered. On the top right, the time '8:26 AM' and date 'Monday 20 September' are displayed. Below the title, the text 'Video intercom configuration' is followed by a horizontal line. The configuration fields are: 'Name' with the value 'Main'; 'IP' with the value '192.168.2.220'; 'Extension' with the value '123'; 'Door code 1' with a text input field containing '1234'; 'Door code 2'; and 'Door code 3'. At the bottom right, there is a 'SAVE' button with a checkmark icon.

Once the data has been entered, SAVE button is pressed and we would have the VIIP configured to receive calls from the video intercom.



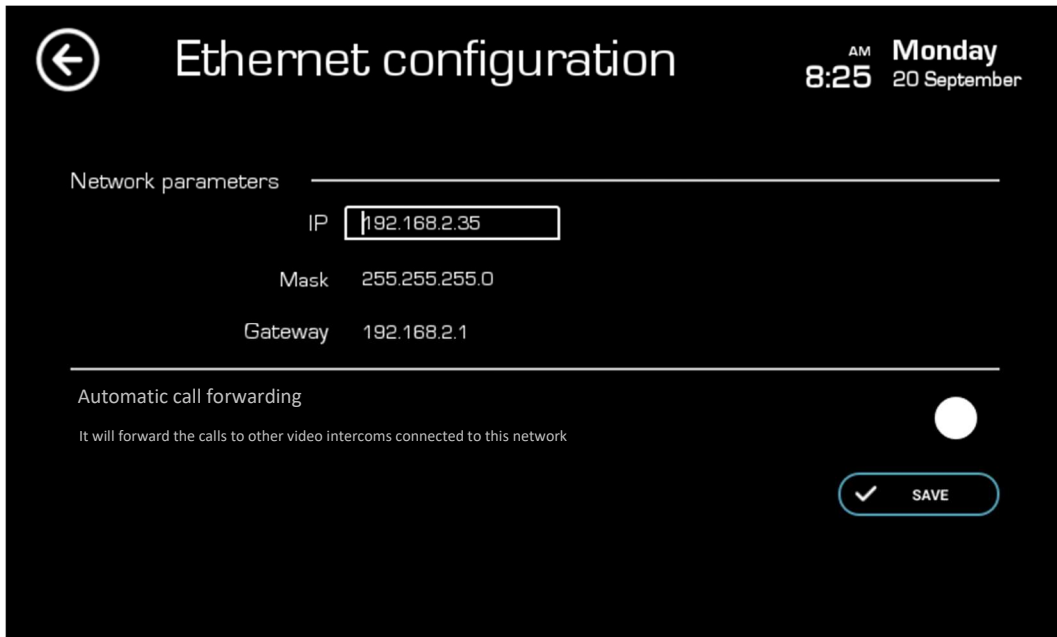
Once it is configured, you must configure the connection of the VIIP with the video intercom, click on the Video intercom output button until the icon of port 1, port 2 or Wi-Fi appears.

Then click on the  gear button to configure the connection.

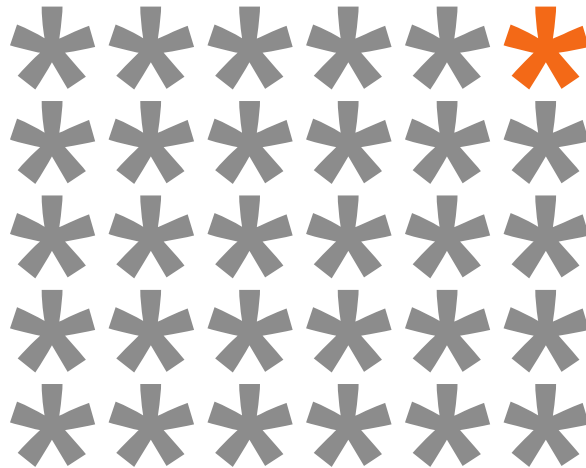
A window appears with the following options:

- IP: A fixed IP is entered that is in the same subnet as the panel.
- Subnet mask: value of the network's subnet mask.
- Gateway: network gateway value.

There is also a button to activate call forwarding. If the screen in which it is configured is activated, it will divert the calls to all the VIIPs that are connected to the same network.



To finish, press the SAVE button.



KNX products by ingenium



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

Parque Tecnológico de Asturias, Parcela 50

33428 Llanera, Asturias, España

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: v1.0*