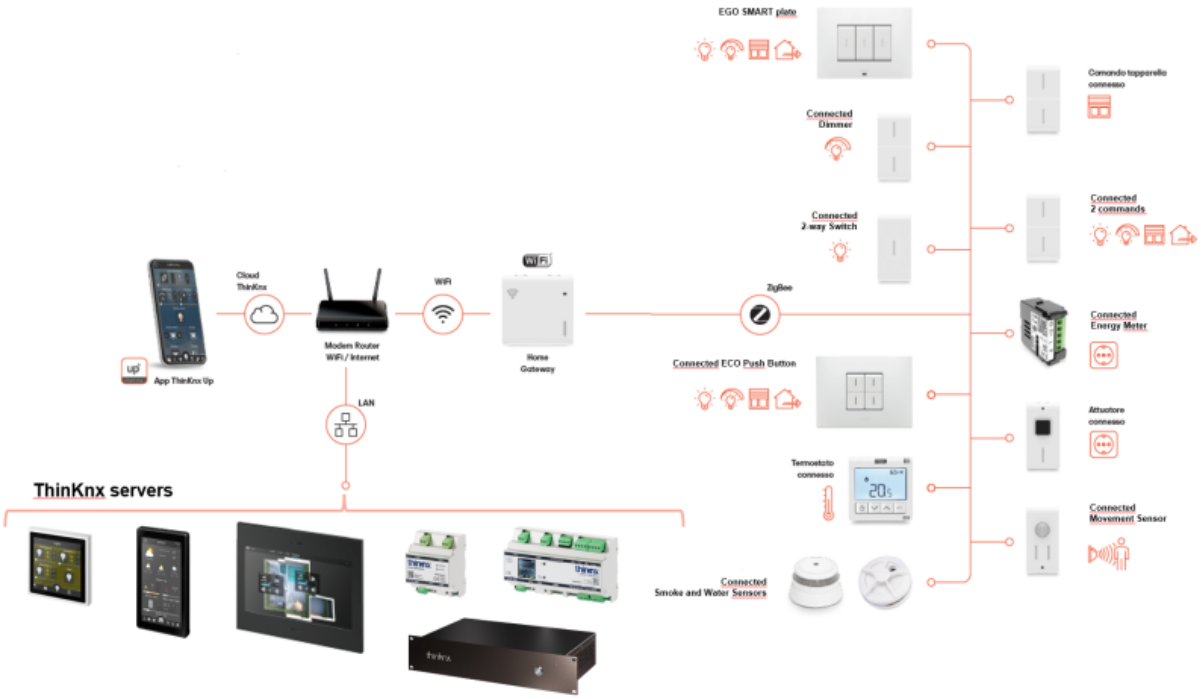
System architecture and limits



Maximum number of Home Gateways for each ThinKnx server:

- PICCOLO and K2: 1
- Other servers: 5

Maximum number of ZigBee devices for each Home Gateway:

- For each Home Gateway: 60

If the application requires to manage a higher number of Home Gateways per ThinKnx Server or a higher number of ZigBee devices per Home Gateway, please contact the technical assistance service for a deeper feasibility analysis.

Prerequisites

Requires 1.1.10.28 server firmware and 1.2.42 Thinknx App.

Available only on servers manufactured from 2021 onwards(**Micro20,Compact20,Rack25,Envision20**)

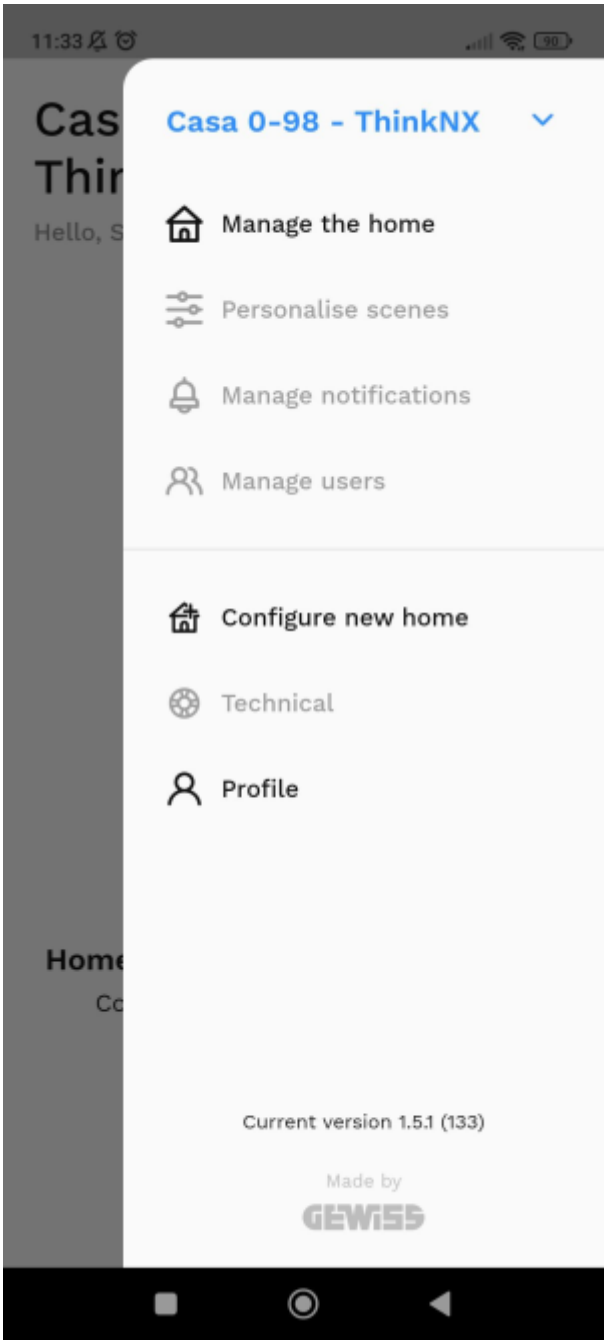
Configure the **Home Gateway** for communication with the **Thinknx server**.

Home Gateway LED indicators:

- **Purple** = Thinknx mode active
- **Blue** = Gewiss mode active (compatible with the Gewiss app)

Option 1 – Standard Home Gateway Configuration

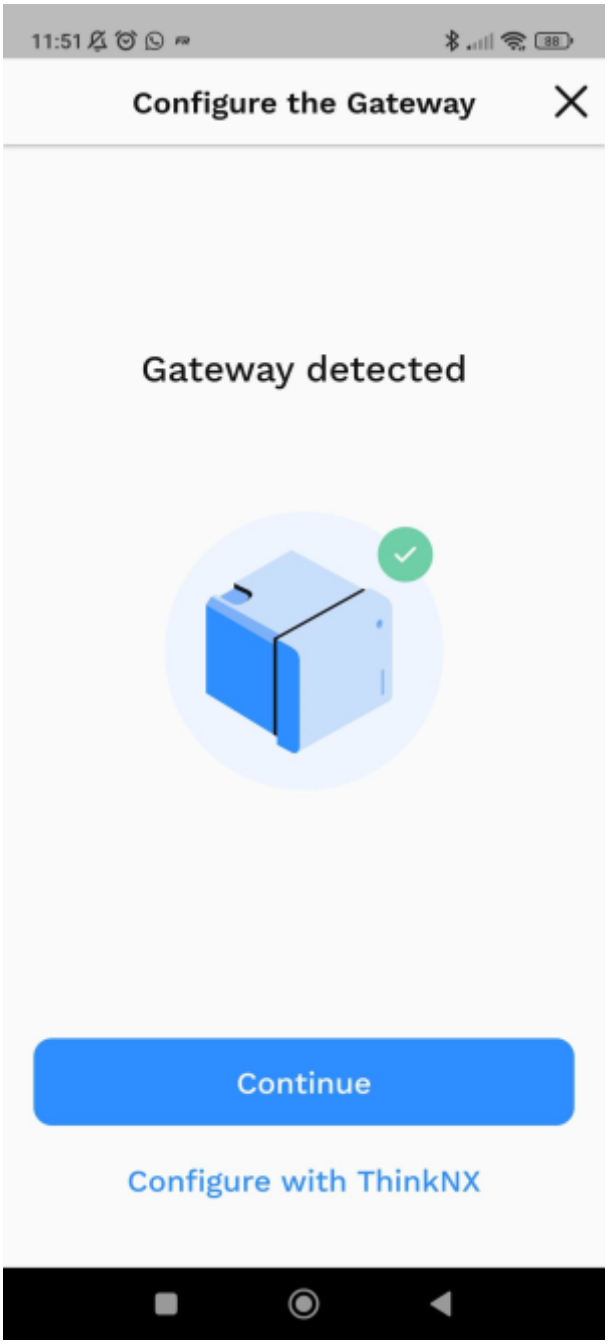
- Launch the **Home Gateway app**
- Select **Configure new home** from the “hamburger menu” (top right side)



- In order to identify the Home Gateway to be configured, follow the instructions guided by the app and then press **Continue**

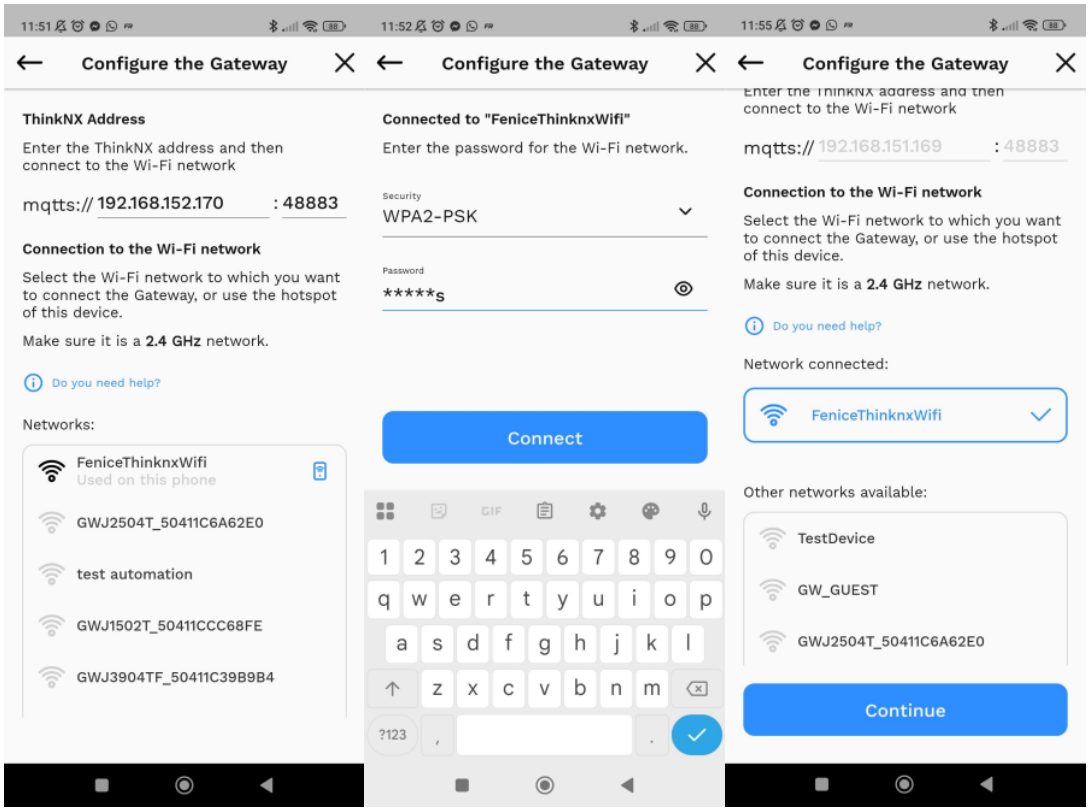
Wiki path: Option 1 – Standard Home Gateway configuration

- Press **Configure with Thinknx**



- Configure the following parameters to allow the Home Gateway to connect to the MQTT Broker (running on the Thinknx server) and to the local Wi-Fi network:

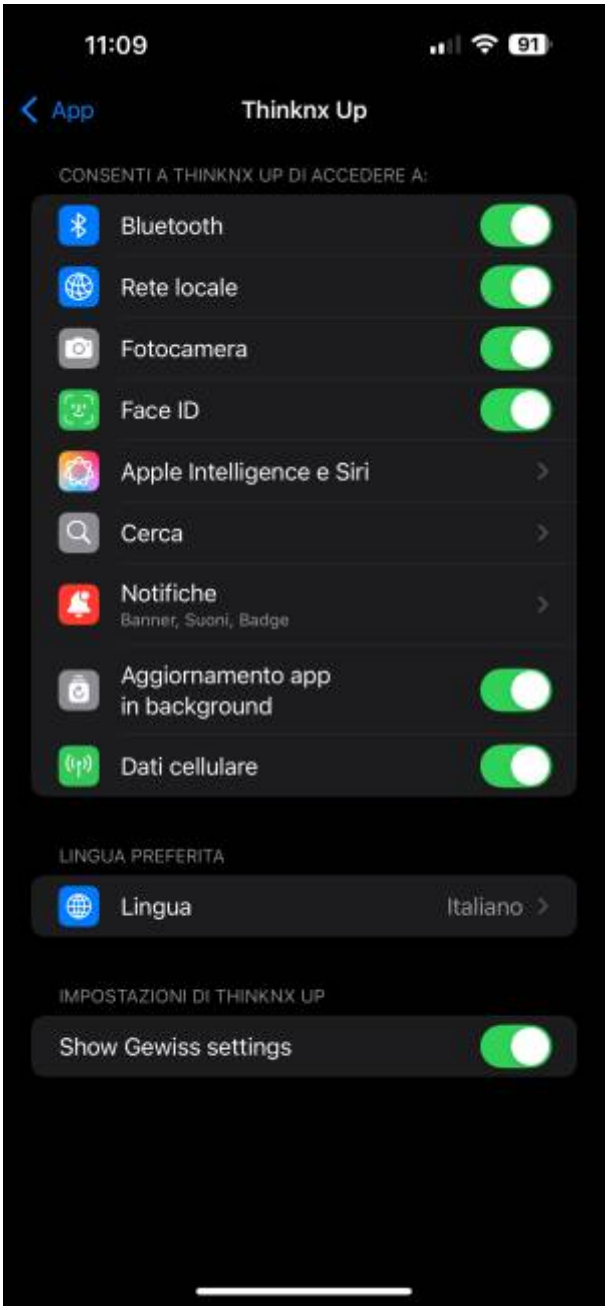
- **MQTT Broker IP address and port**
    - IP address: same IP address as the Thinknx server
    - Port: always **48883**
    - **Wi-Fi network**
        - Name: name of the Wi-Fi network
        - Security: always **WPA2-PSK**
        - Password: Wi-Fi network password



- At the end of this configuration phase:
  - The Home Gateway configuration is successful if the LED on the Home Gateway is **violet**
  - Then press the **Continue** button to proceed with the configuration using the **Thinknx Configurator** tool

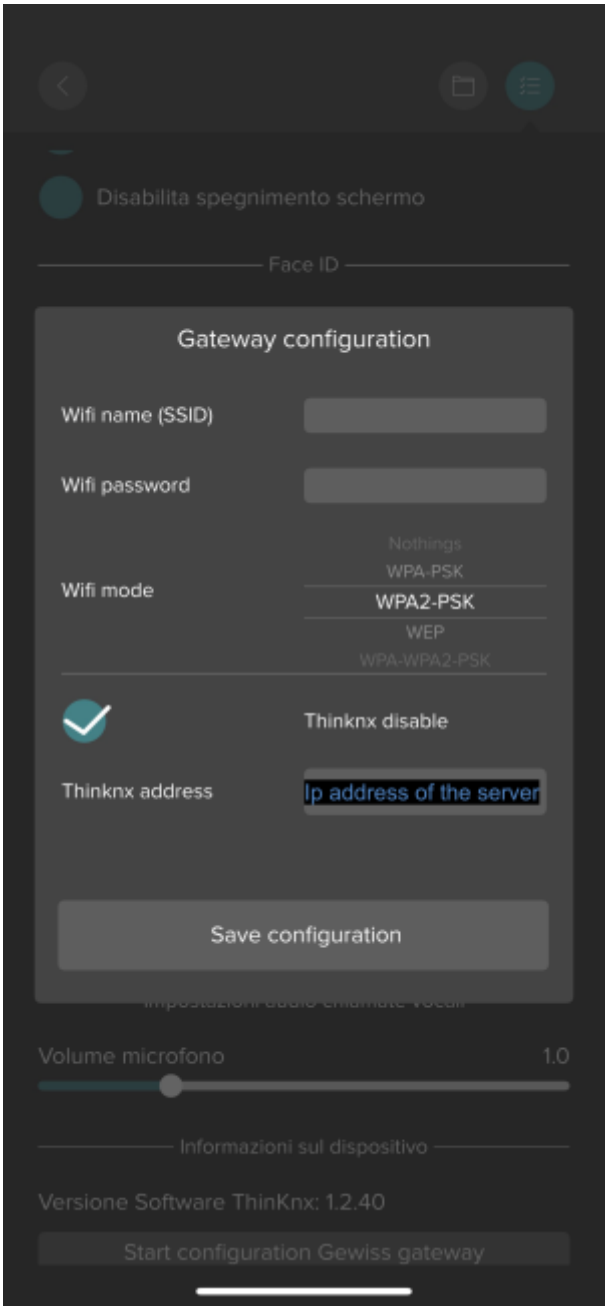
Option 2 – Configuration via Thinknx Up

- Enable the Gewiss configuration in the Thinknx Up app
  - Go to Settings → App → Thinknx Up → Show Gewiss settings



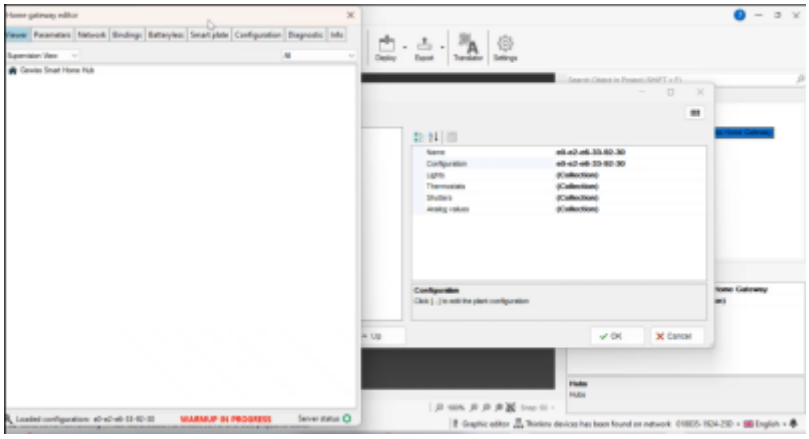
- Open the **Thinknx Up** app
- Go to the **Settings** menu
- Scroll to the bottom and select **Start Gewiss System Configuration**

- Press and hold the Home Gateway programming button for **4 seconds** until the LED starts blinking
- Wait for the blinking to stop to complete the association
- Enter the Wi-Fi network name (**SSID**), respecting uppercase and lowercase
- Enter the **Wi-Fi password**
- Select the security type (e.g., WPA2-PSK2)
- Enable **Thinknx** if available
- Enter the **Thinknx server IP address**

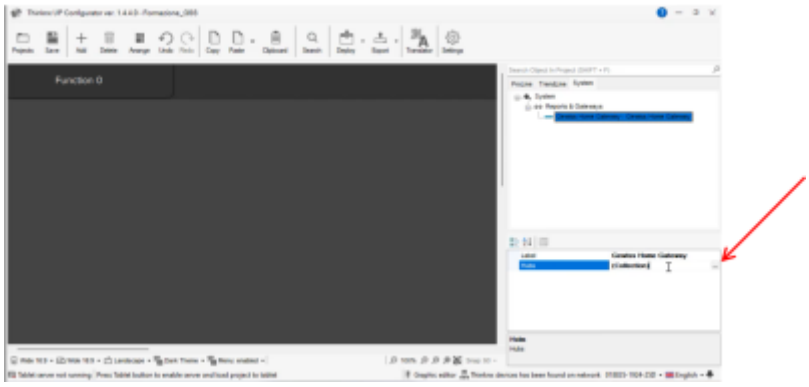


Creating a Gewiss Smart Home system in Thinknx

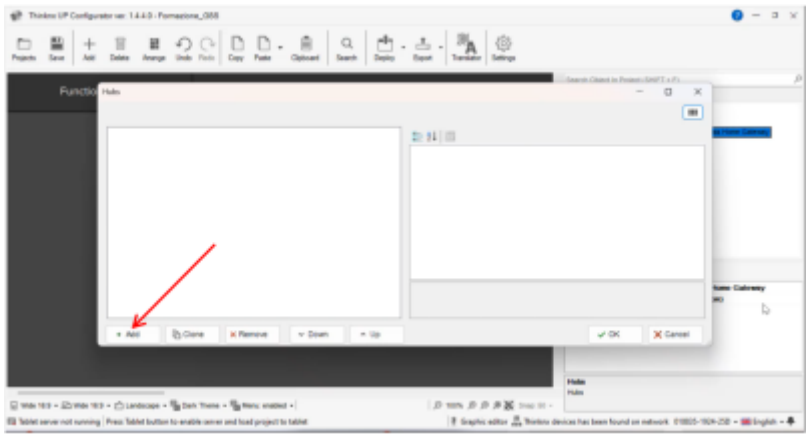
Add the **Gewiss Home Gateway** object in the **System** menu.



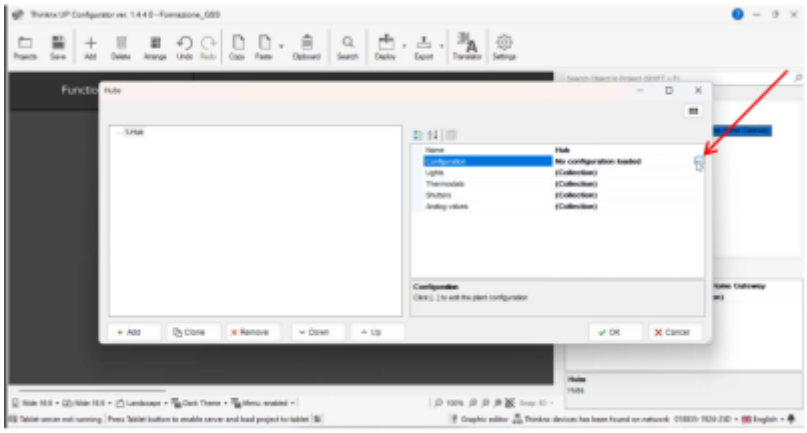
Start the gateway configuration: click on the button with the **three dots**.



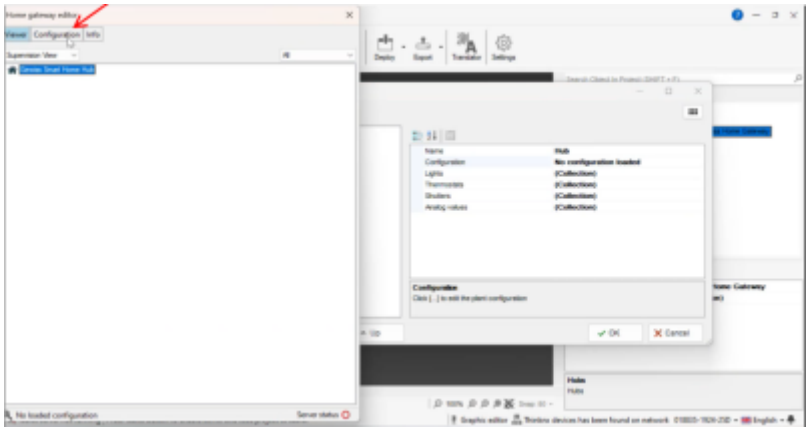
Click on the **Add** button.



Define the name: type in the **Name** field the name of your Smart Home system (e.g.: My Home). Click on the button with the **three dots** in the configuration field.



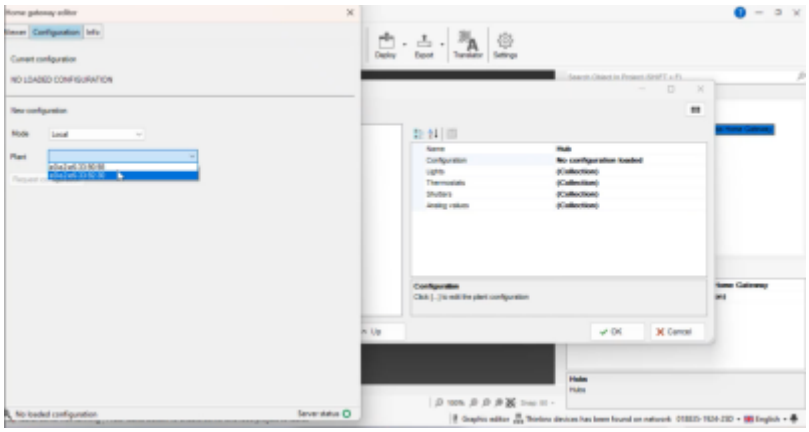
Click on the **Configuration** tab.



In the **Plant** field select the **mac-address** of the Home Gateway (select the mac-address of your Home Gateway from the list).

In the **Mode** field select:

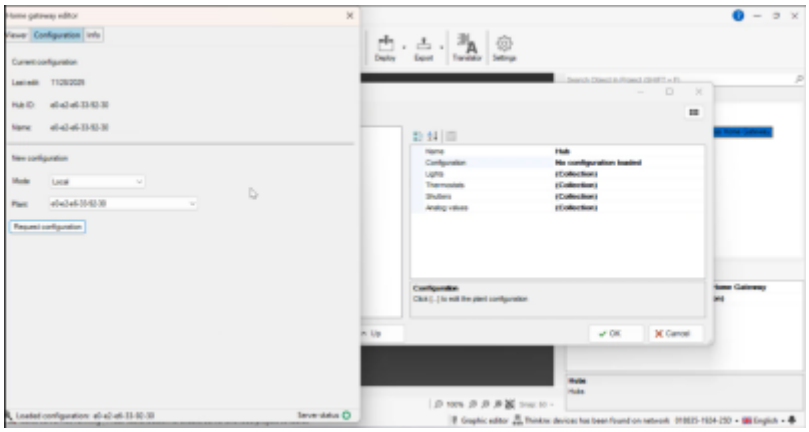
- **Local**: when you want to create the ZigBee network configuration from scratch (network creation, bindings, etc.)
- **Cloud**: when you want to upload a ZigBee configuration already created using the Home Gateway app and migrated to Thinknx (you will be asked for the account to access the system)



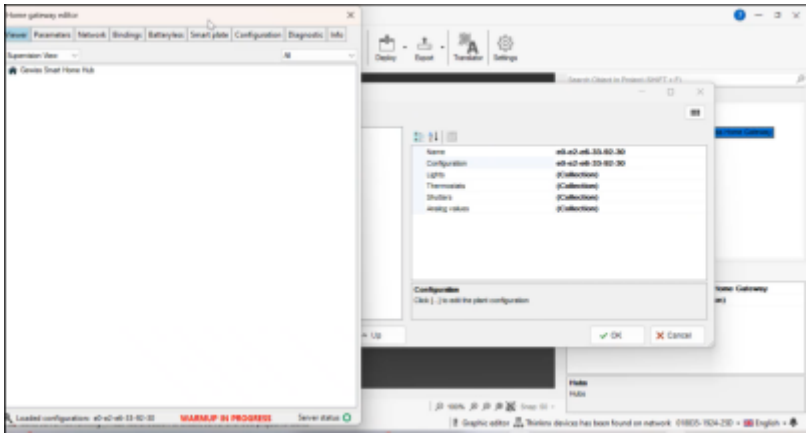
Below are the steps to migrate a system configured with the Home Gateway

app to Thinknx:

- From the Home Gateway app menu select **Manage the Home**
- Select **Managing the gateway**
- Select **Configure the gateway**
- Select **ThinKnx parameters**
- Click on **Modifying the parameters**
- Follow the instructions provided by the app



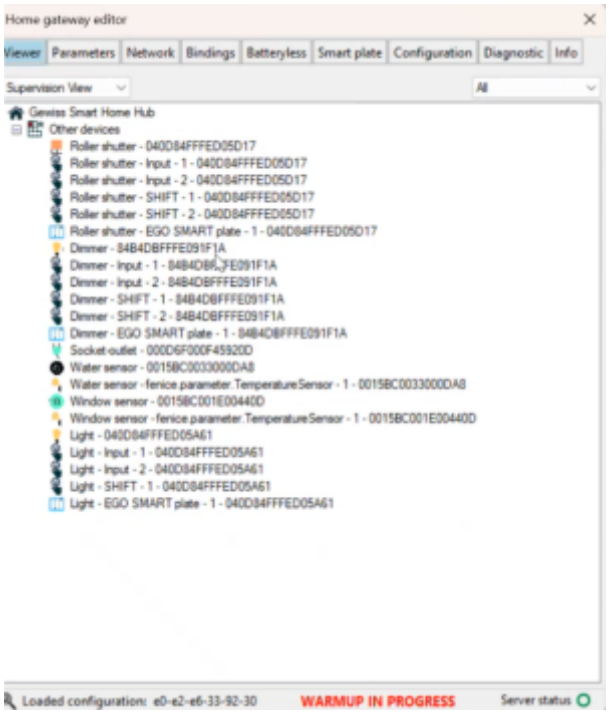
Click on the **Request Configuration** button.



At this point you can start configuring the ZigBee network (network creation, bindings definition, etc.), as described in the next chapter:

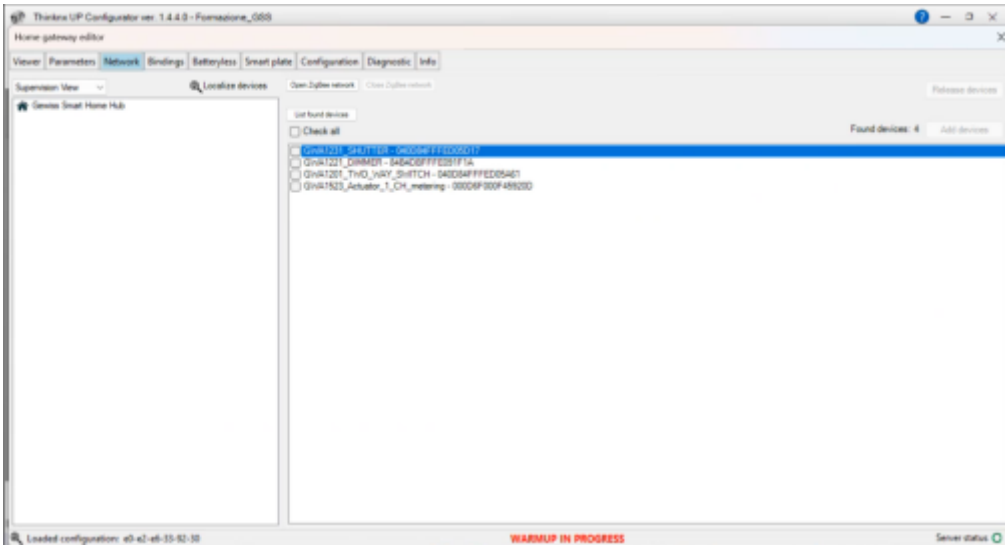
Configuration Tabs

Viewer The Viewer tab lists all ZigBee devices added to the project, the auxiliary input and the shift command.

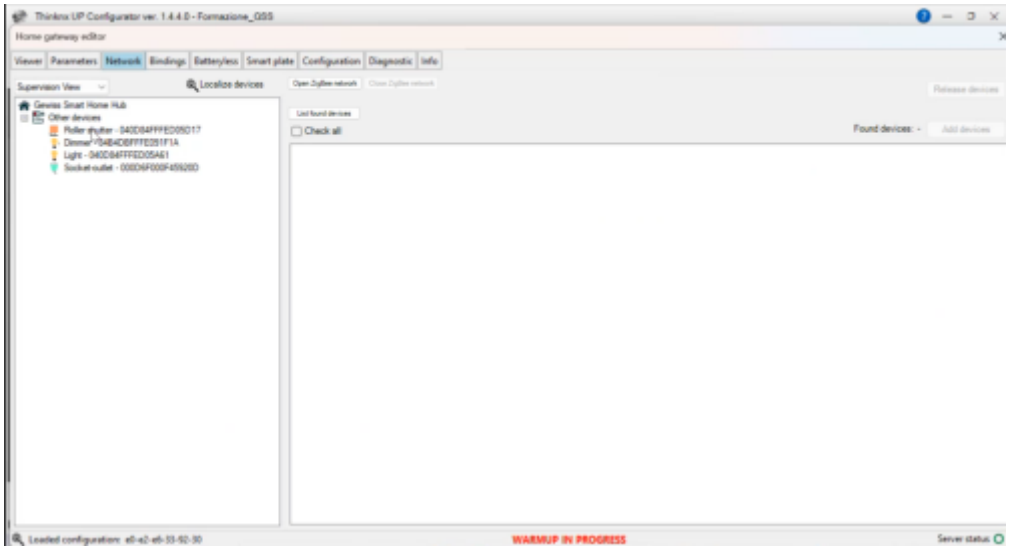


Network The Network tab allows the creation of the ZigBee network during the first commissioning or the addition of ZigBee devices at a later stage (e.g. maintenance).

- **Open ZigBee Network** – opens the ZigBee network (Coordinator) and allows ZigBee devices to join; joined devices are listed in the right window
- **Close ZigBee Network** – closes the ZigBee network (Coordinator)
- **List found devices** – lists all devices joined to the network

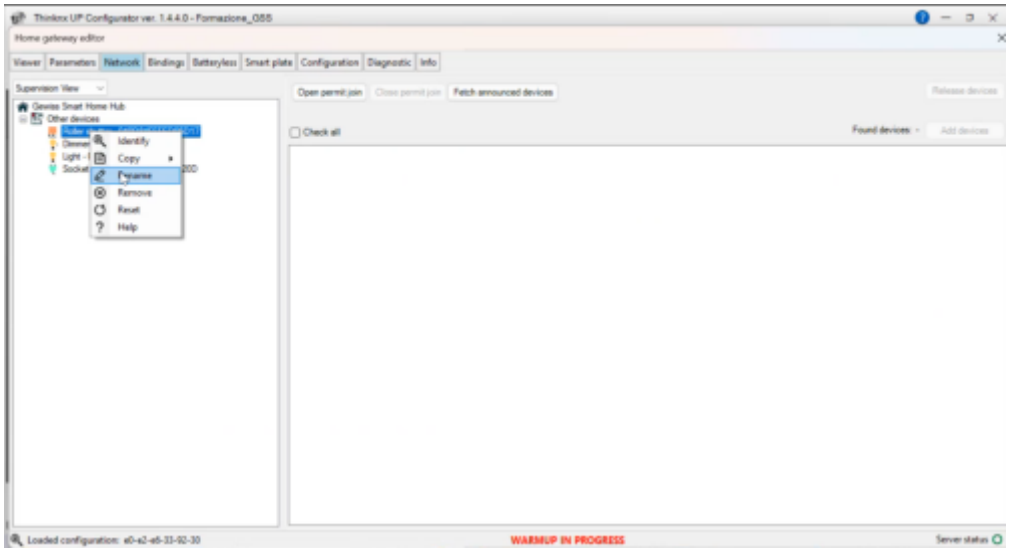


- **Add devices** – adds to the project all checked devices from the right window; added devices are then listed in the left window

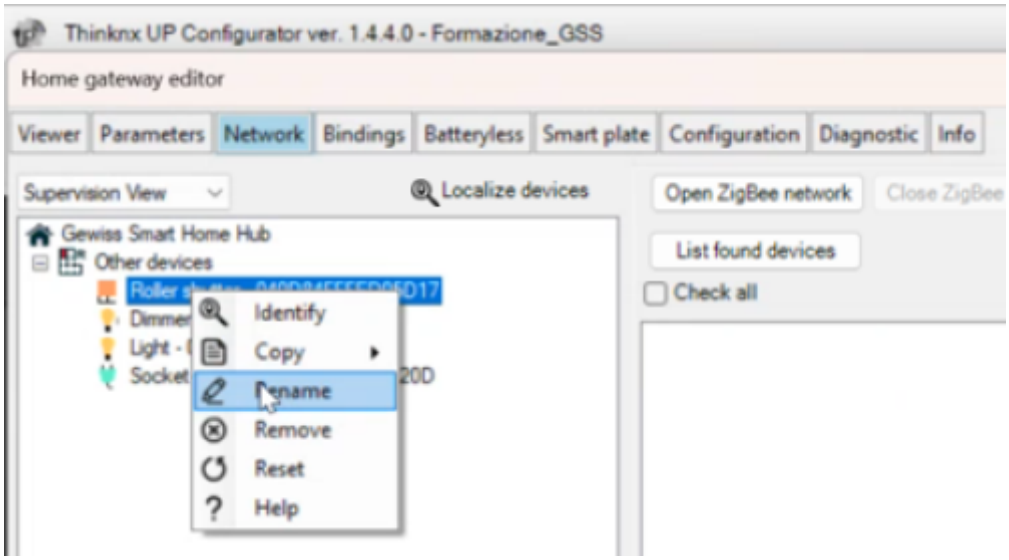


Special functions for ZigBee device management: The following functions are activated by left-clicking on a specific device:

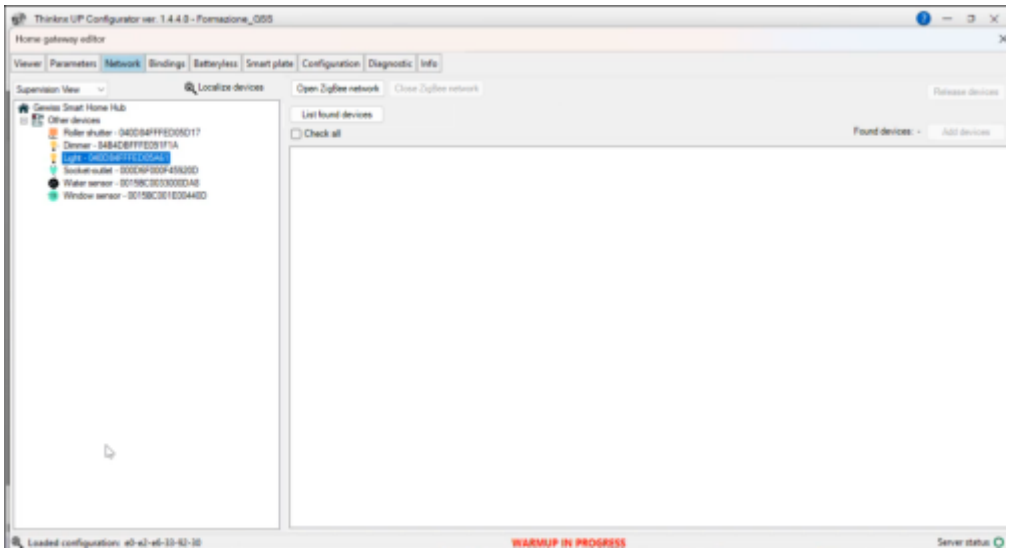
- **Identify** – activates the device to identify it on the field
- **Copy** – copies to the clipboard the following device information: ID, IEEE code, Model
- **Rename** – renames the device
- **Remove** – removes the device from the project while keeping it joined to the ZigBee network
- **Reset** – factory reset; the device is also removed from the ZigBee network
- **Help** – opens the default web browser to the Gewiss website related to the selected ZigBee device



By clicking on the **lens icon**, the function **Identify device from the field** is activated.



When a ZigBee device is activated on the field (e.g. pressing the local button of a dimmer), the corresponding device in the left window is highlighted. The function stops when clicking again on the lens icon.



To exit this feature, simply tap the magnifying glass again, or it will close automatically after a period of inactivity.

Parameters This tab allows defining and modifying parameters of ZigBee devices included in the project. Parameter values define device behaviour.

In the left window all ZigBee devices are listed. In the central window the parameters of the selected device are shown.

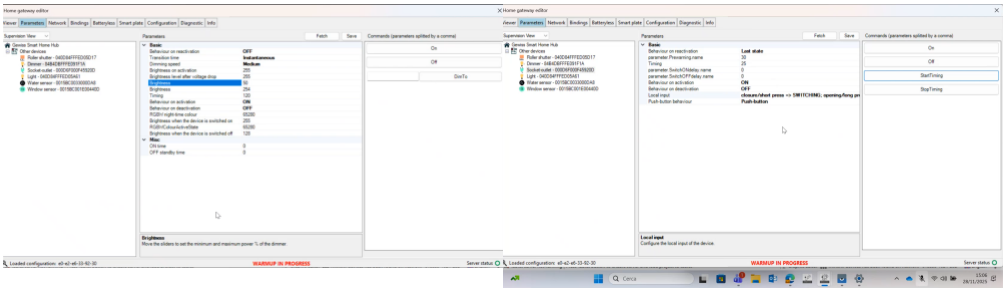
- **Fetch** – loads parameters from the selected ZigBee device
- **Save** – saves modified parameters

In the right window, commands can be sent directly to the selected device.

Example Dimmer: for a **Dimmer** the following commands are available:

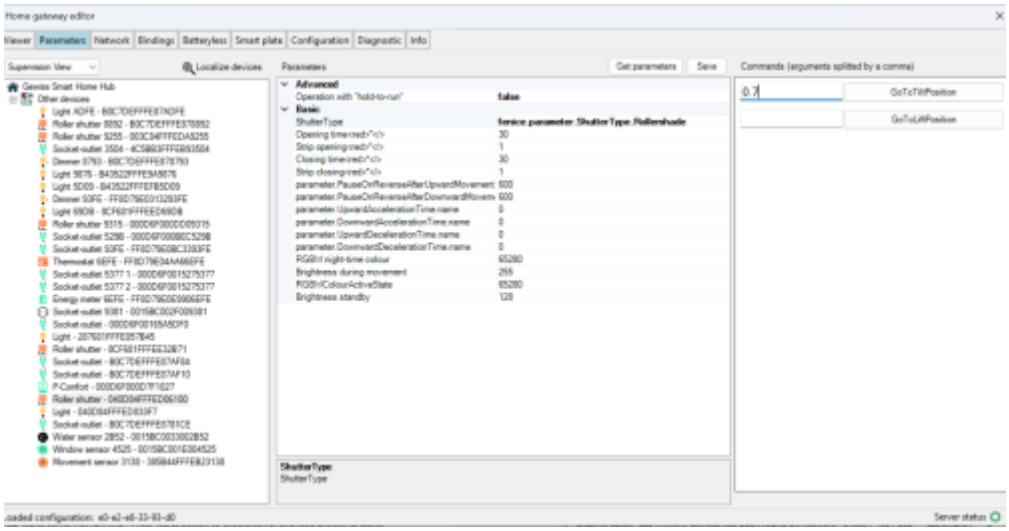
- **On** – switches the dimmer ON

- **Off** – switches the dimmer OFF
- **DimTo** – sets the illumination level (value between 0 and 1; 0.2 = 20%)



Example Shutter: for a **Roller Shutter** the following commands are available:

- **GoToTiltPosition** – sets the tilt angle of the slats when the shutter type is Blinds (value between 0 and 1; 0.2 = 20%)
- **GoToLiftPosition** – controls the lowering or raising of the roller shutter (value between 0 and 1; 0.2 = 20%)

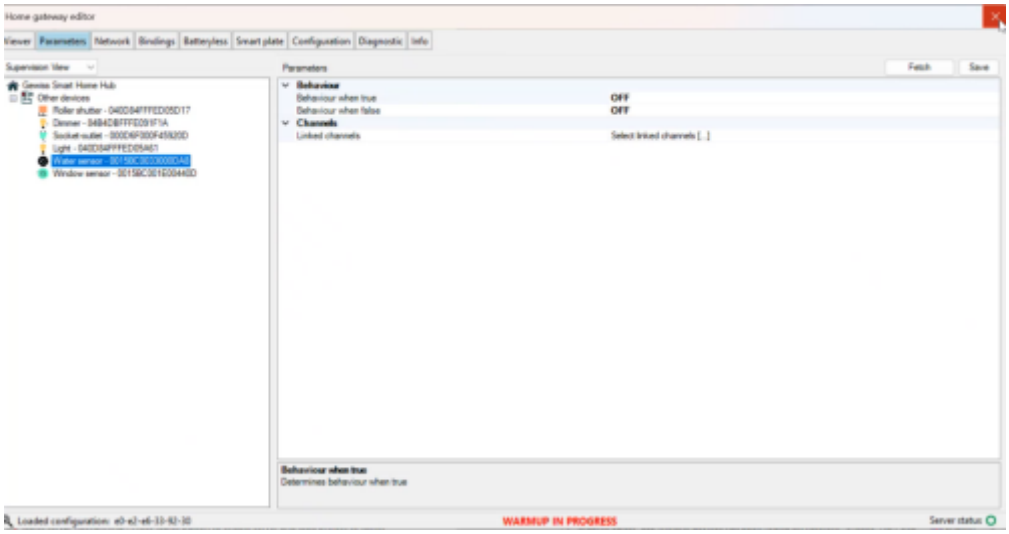


For sensors (water sensor, door sensor, movement sensor, etc.) commands cannot be sent, but actuators can be defined directly in this tab for alarm ON and OFF conditions.

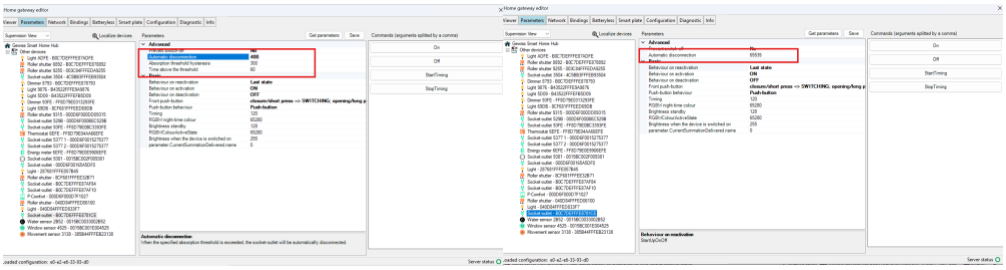
The window is split into two sections:

- **Behaviour** – defines sensor behaviour
- **Linked channels** – actuators commanded when alarm is ON or OFF

Example: If *behaviour when True = 0* and *behaviour when False = 1* are set, and the linked channel controls the water valve, the valve opens when a water leakage is detected and closes when the alarm ends.



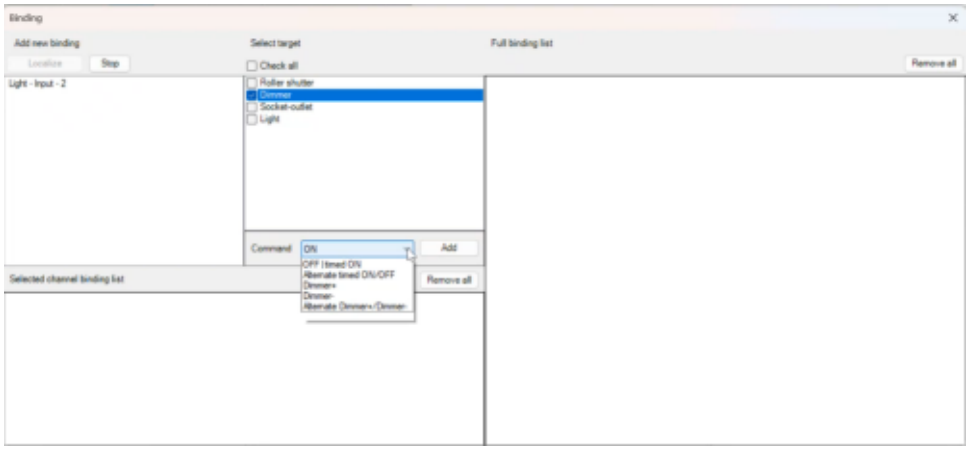
Example for GW1X826: The activation of ‘automatic disconnection’, in the case of the GW1X826 actuator, is performed by setting the value to 65535; otherwise, the disconnection function is disabled.



Bindings Bindings allow ZigBee devices to directly exchange ZigBee commands. The **“Check all”** option can be used only when the selected target devices support the same commands The **“Delete all”** option can be used to remove all created bindings either for the selected source device or for all source devices

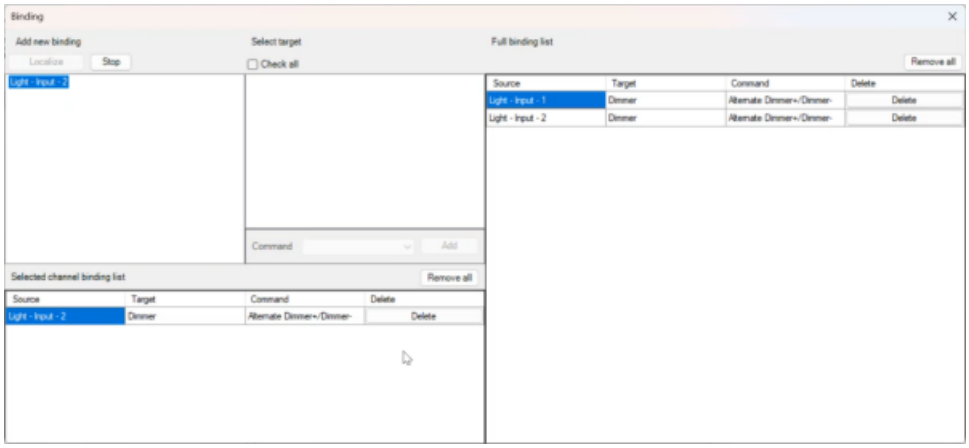
Example: to control a light using a ZigBee push button, a binding must be created between the push button and the actuator.

- **Localize**
  - Identify the command by pressing the physical button
  - The command name appears in the left window
  - Compatible actuators are listed in the central window
  - Select one or more actuators
  - Select the command type
  - Click **Add** to create the binding



After creation:

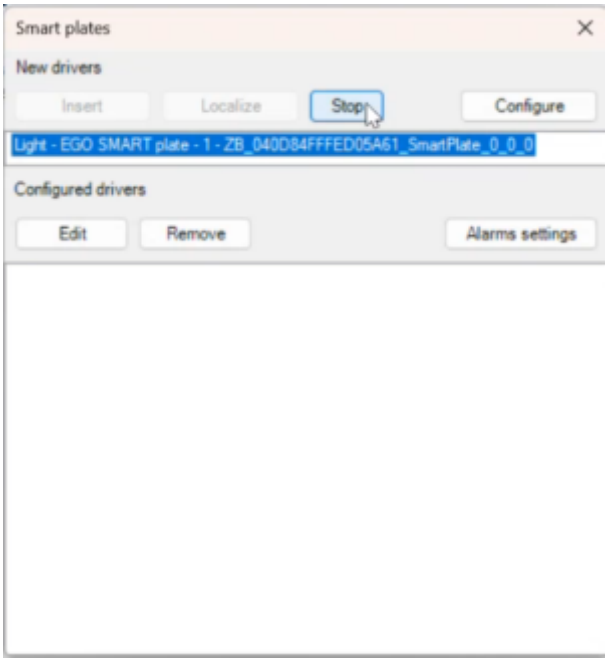
- The created binding is shown in the lower left window
- All bindings are listed in the right window (deletable only)
- Click **Stop** once all bindings are defined



Smart Plate This tab is used to configure Smart Plates.

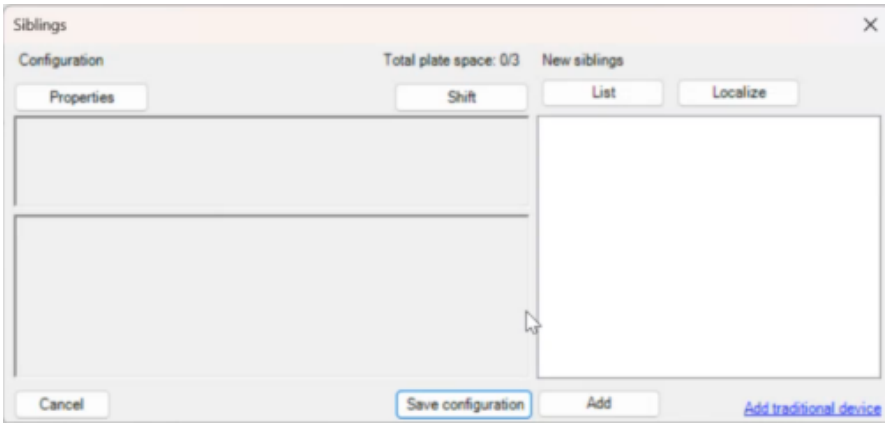
To add a Smart Plate:

- **Insert** – identify the Smart Plate by typing its ID(reported on the plate)
- **Localize** – recommended method:
  - Cover the proximity sensor for at least 5 seconds
  - The Smart Plate is identified
  - Click **Stop**

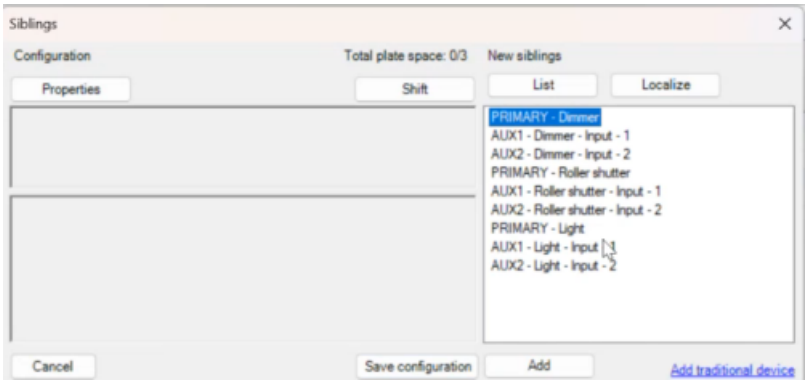


To configure the Smart Plate:

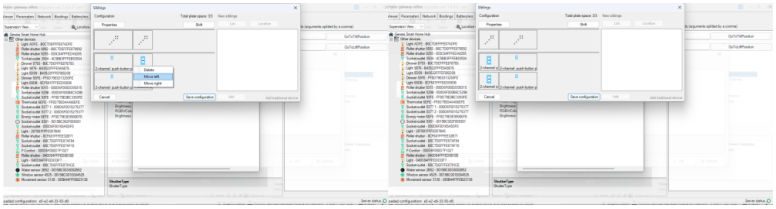
- **Configure** – opens the configuration window,that allows to configure all the function and parameters of the Smart Plate



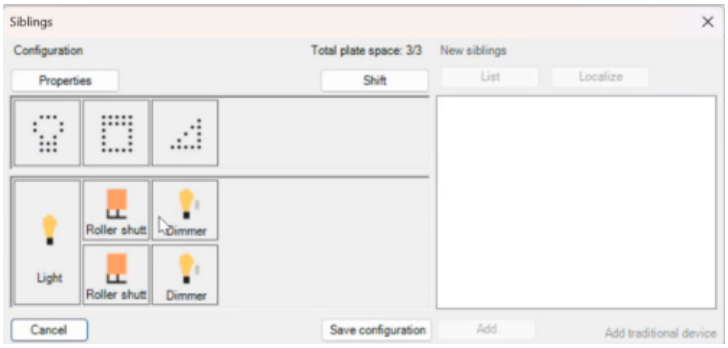
- **List** – defines the composition of devices installed inside the Smart Plate
  - Starting from the left of the Smart Plate devices must be selected among the ones that are listed in the left window



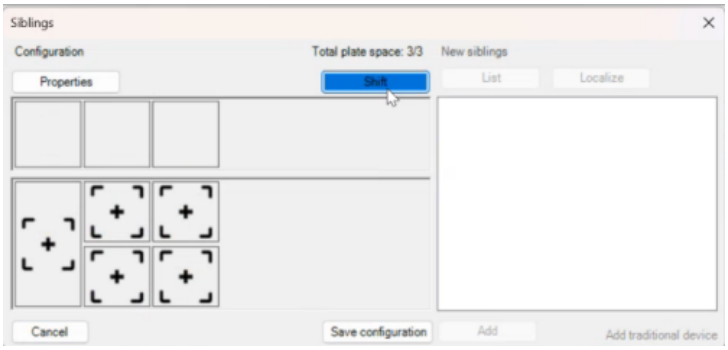
You can right-click a device and move it left or right to change its position.



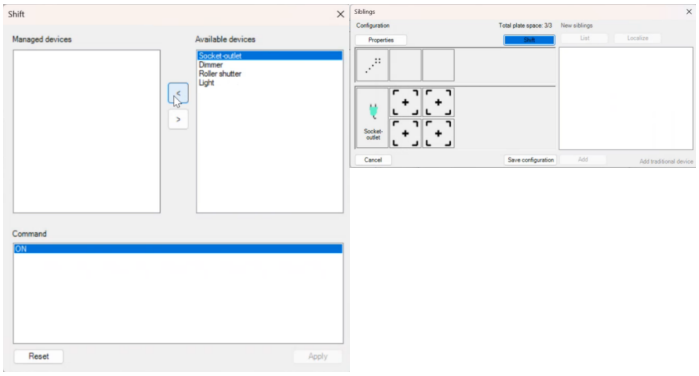
- **Localize** – identifies connected devices by pressing local buttons
  - Starting from the left of the Smart Plate it is enough to «push» (local push button) of each connected device included in the Smart Plate
  - If there are «traditional» devices it must be specified
  - The icons of the connected devices cannot be modified (because they depend on the function of the device)



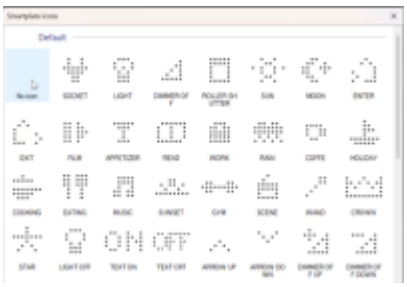
- **Shift** – allows to define the commands associated to the Shift function of the Smart Plate:



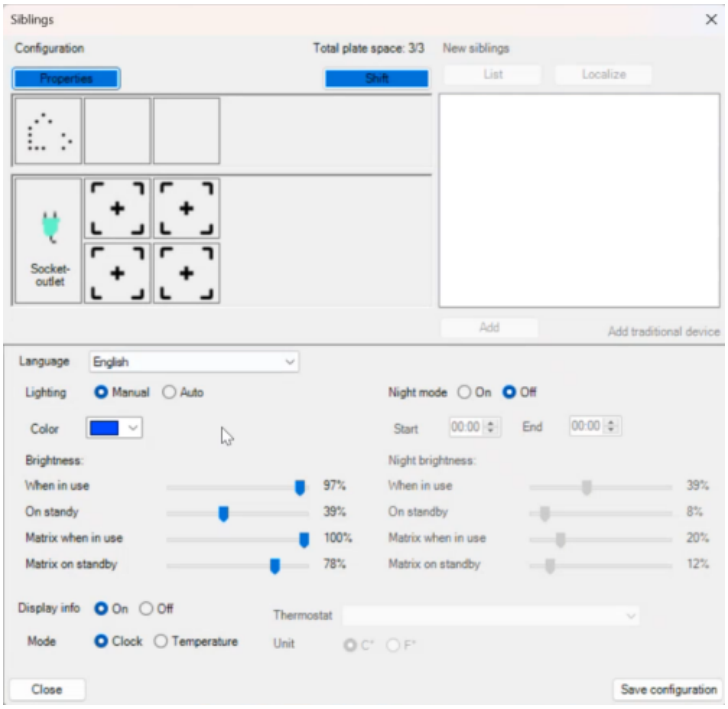
- Click on each «device» (a device can have one or two push bitton, see the symbol [+]) to define the actuator to be commanded and the command to be given, in the example below it is defined to give a ON command to the Smart Plug, by mean the first push button on the left:



- To define the icons displayed when the Shift function is activated, click on each icon and then select the desired icon among the library proposed:



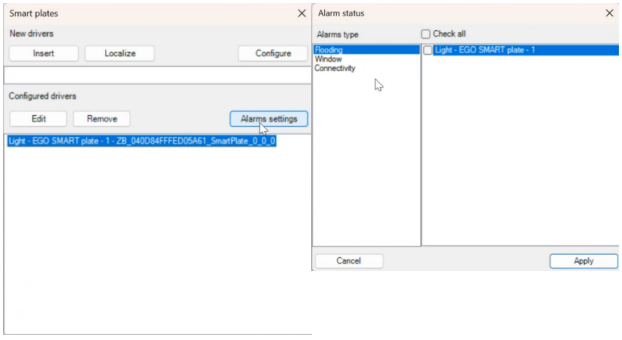
- **Parameters** – allows to define the behaviour of the Smart Plate: language, colour of the strip led, if in stand-by shall display the time or the temperature (for the temperature in the project there shall be at least one thermostat), etc



- **Alarms setting** – allows to define what are the alarms that has to be displayed on the Smart Plate, for each type of alarm a predefined message is displayed, the type of alarms are

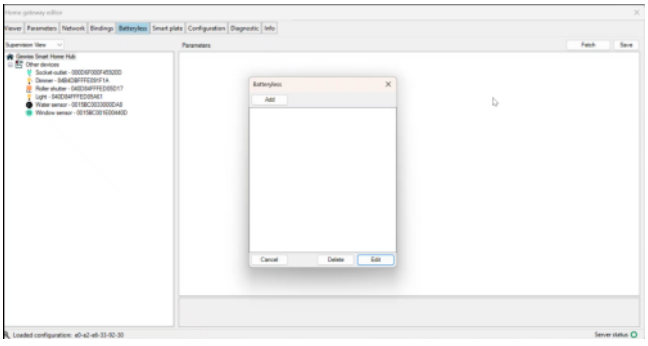
- Flooding – when one of the water sensors detects a water-leakage
- Window – when one of the window sensors detects a window-opened
- Movement – when one of the movement sensors detetcts movement
- Smoke – when one of the smoke sensors detetcs smoke
- Connectivity – when one of the ZigBee device has problem of connectivity

Alarm messages can be displayed on all Smart Plates or only on selected ones.



Batteryless This tab allows defining commands sent by Batteryless push buttons.

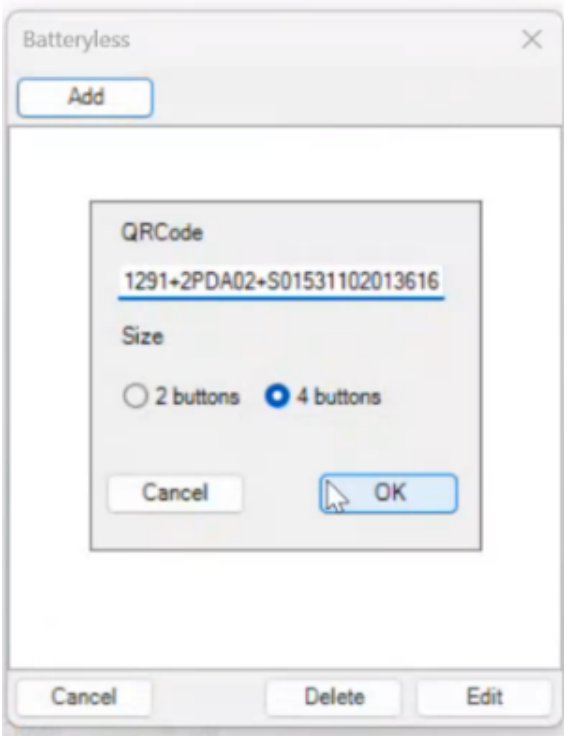
- **Add** – adds a Batteryless push button
- In the lower left window is reported the «binding created»
- In the rigt-window are reported all the «bindings» created», it is possible to modify/delete each one



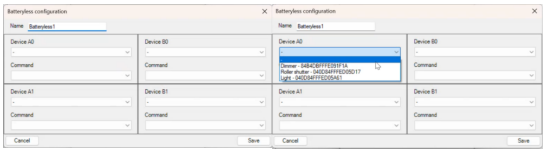
To configure a Batteryless push button:

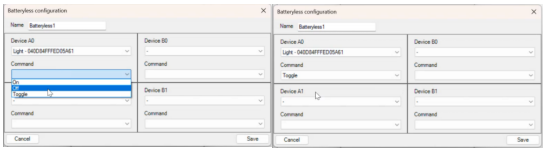
- Insert the QR-Code associated to the Batteryless push-button
  - Read by using your Smartphone the QR-Code attached on the back of

the Batteryless push-button
- Report the Qr-Code on the field named «Qr-Code»
- Select if you want to configure it as a 4 or 2 push buttons



- Give a name to the batteryless push button
- Configure each single push-button (A0, A1, B0, B1) defining the actuator to be commanded (e.g. Light) and the command to be given (e.g.: Toggle)



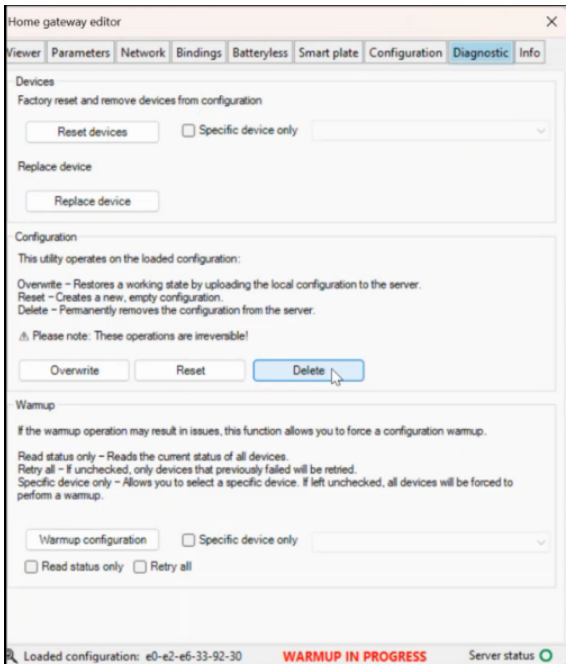


Diagnostic This tab is mainly used for maintenance operations.

* **Reset devices** – factory reset of one or all devices

- **Replace device** – replacement of a

faulty device



Associations for Thinknx Up and KNX

This section describes how to create associations between Thinknx objects and Gewiss Connected Smart Home objects.

Thinknx objects represent a logical link to Gewiss Smart Home devices and must be used whenever Smart Home data (commands, status, measures) are required inside Thinknx functions, such as:data logging, scenes, and logic functions.

To create the associations:

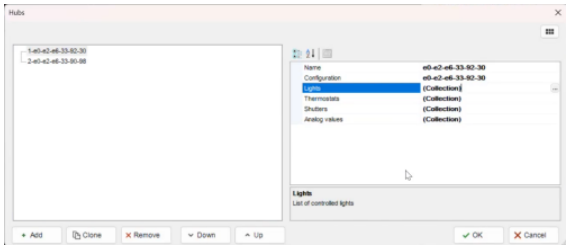
- Go to **System**
- Select **Gewiss Home Gateway**
- Select the desired **Hub**

The following sections allow the assignment of Gewiss smart home Devices to Thinknx and KNX group addresses.

Lights

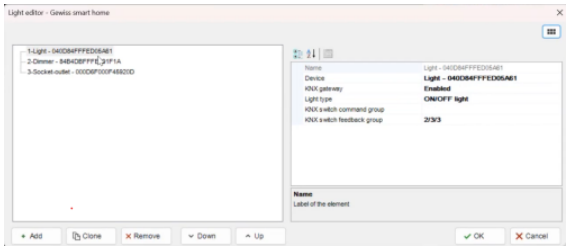
This section allows defining Thinknx objects associated with Smart Home light devices.

- Click on the **three dots** at the end of the Lights line
- Select **Add**



Define the following parameters:

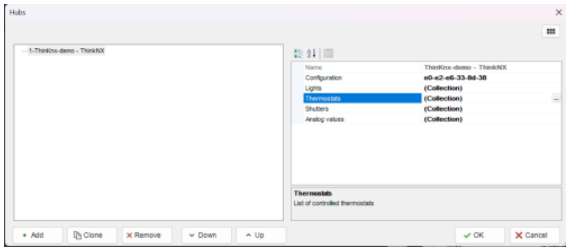
- **Name** – name of the Thinknx object
- **Device** – select the Smart Home light device
- **KNX Gateway** – enable to link the object to a KNX address
- **Light type** – On/Off, Dimmer, Any
- **KNX command and feedback groups** – KNX group addresses linked to the object



Thermostats

This section allows defining Thinknx objects associated with Smart Home thermostats.

- Click on the **three dots** at the end of the Thermostats line
- Select **Add**



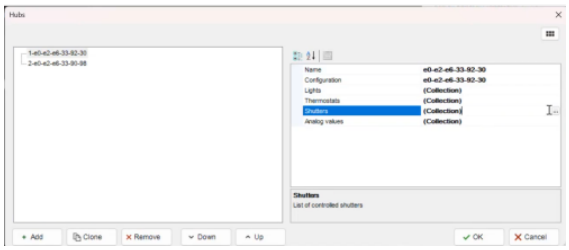
Define the following parameters:

- **Name** – name of the Thinknx object
- **Device** – select the Smart Home thermostat
- **KNX Gateway** – enable to link the object to a KNX address
- **KNX groups** – KNX group addresses linked to the object

Shutters

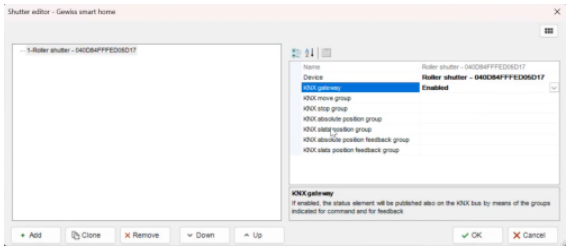
This section allows defining Thinknx objects associated with Smart Home shutter devices.

- Click on the **three dots** at the end of the Shutters line
- Select **Add**



Define the following parameters:

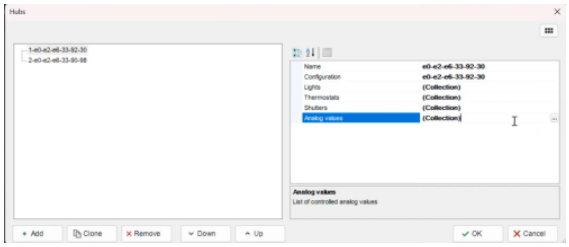
- **Name** – name of the Thinknx object
- **Device** – select the Smart Home shutter
- **KNX Gateway** – enable to link the object to a KNX address
- **KNX groups** – KNX group addresses linked to the object



Analog values

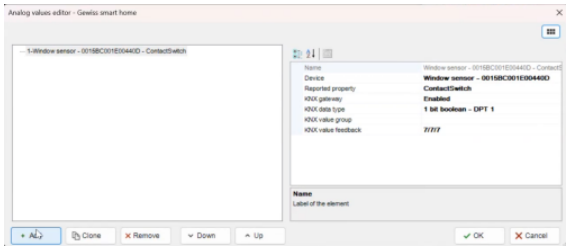
This section allows defining Thinknx objects associated with Smart Home values (status, measures, alarms, etc.).

- Click on the **three dots** at the end of the Analog values line
- Select **Add**



Define the following parameters:

- **Name** – name of the Thinknx object
- **Device** – select the Smart Home device
- **Reported property** – select the value to be associated (status, measure, alarm)
- **KNX Gateway** – enable to link the object to a KNX address
- **KNX value and feedback groups** – KNX group addresses linked to the object



Object Assignment for Control

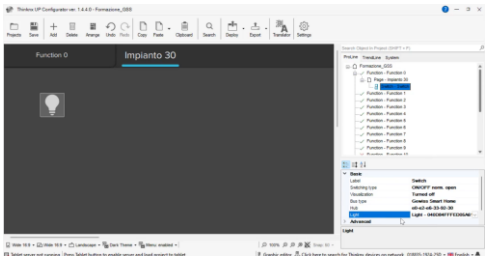
This section describes how to visualize and control Gewiss Smart Home objects in Thinknx supervision pages.

General steps:

- Add a Thinknx graphic object to the supervision page
- Associate the corresponding Thinknx object
- Configure object parameters

Control of an ON/OFF light

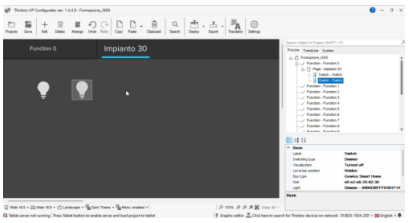
- Add **Object Switch** to the supervision page
- Configure the parameters:
  - Switching type – ON/OFF normally open
  - Bus type – Gewiss Smart Home
  - Hub – select the Home Gateway
  - Light – select the Thinknx light object



Control of a dimming light

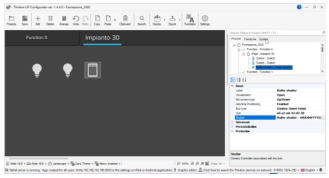
- Add **Object Switch** to the supervision page
- Configure the parameters:
  - Switching type – Dimmer
  - Bus type – Gewiss Smart Home
  - Hub – select the Home

Gateway
- Light – select the Thinknx dimmer object



Control of a roller shutter

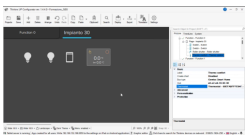
- Add **Object Roller Shutter** to the supervision page
- Configure the parameters:
  - Bus type – Gewiss Smart Home
  - Hub – select the Home Gateway
  - Shutter – select the Thinknx shutter object



Control of a thermostat

- Add **Object Thermo Comfort** to the supervision page
- Configure the parameters:
  - Bus type – Gewiss Smart Home
  - Hub – select the Home Gateway
  - Thermostat

– select the Thinknx thermostat object



Visualization of a value

This allows visualization of measures, alarms, or status values.

- Add **Object Analogue Value** to the supervision page
- Configure the parameters:
  - Data source – Gewiss Smart Home Hub – select the Home Gateway
  - Analog – select the Thinknx analog objec

t

Values can be analog (e.g. temperature) or binary (e.g. alarm).

Writing a parameter value

Writing a parameter value to a Gewiss Smart Home device. To write a parameter value:

- Add **Object Generic Command** to the supervision page
- Define the parameters associated with

this objec t

- **Command** – select the Gewiss Smart Home device and then th

evalue(e.g.parameter)tobewrittenin

When clicking on the **Command** property (see image above), the

configurator helps in identifying the value to be written by using a simple **navigation tree**, which allows browsing all objects defined in the system.

Follow these steps:

- Select the **Gew**

i
s
s
H
o
m
e
G
a
t
e
w
a
y
t
o
w
h
i
c
h
t
h
e
d
e
s
i
r
e
d
d
e
v
i
c
e
b
e
l
o
n
g
s

- S
e
l
e
c
t

t
h
e
d
e
s
i
r
e
d
d
e
v
i
c
e
(
i
n
t
h
e
e
x
a
m
p
l
e
,
a
t
h
e
r
m
o
s
t
a
t
)

- S
e
l
e
c
t
t
h

e
p
a
r
a
m
e
t
e
r
o
f
t
h
e
d
e
v
i
c
e
t
o
b
e
w
r
i
t
t
e
n
(
i
n
t
h
e
e
x
a
m
p
l
e
,
t
h
e
S
e

t
s
e
a
s
o
n
p
a
r
a
m
e
t
e
r
,
w
h
i
c
h
d
e
fi
n
e
s
h
o
w
t
h
e
t
h
e
r
m
o
s
t
a
t
a
l
g
o
r
i
t

h
m
o
p
e
r
a
t
e
s
:

- **C**
o
o
l
f
o
r
c
o
o
l
i
n
g
- **H**
e
a
t
f
o
r
h
e
a
t
i
n
g
)

U
s
i
n
g
t
h
i

s
a
p
p
r
o
a
c
h
,
i
t
i
s
p
o
s
s
i
b
l
e
t
o
d
e
fi
n
e
a
g
e
n
e
r
i
c
c
o
m
m
a
n
d
b
u
t
t
o
n
(

i
n
a
s
u
p
e
r
v
i
s
i
o
n
p
a
g
e
)
t
o
s
e
t
t
h
e
t
h
e
r
m
o
s
t
a
t
o
p
e
r
a
t
i
n
g
m
o
d
e
t

o
c
o
o
l
i
n
g
(
s
u
m
m
e
r
)
o
r
h
e
a
t
i
n
g
(
w
i
n
t
e
r
)
.

-

C
o
m
m
a
n
d
s
b
e

t
w
e
e
n
d
i
ff
e
r
e
n
t
H
o
m
e
G
a
t
e
w
a
y
s

A
T
h
i
n
k
n
x
s
e
r
v
e
r
c
a
n
m
a
n

agemorethanoneHomeGateway.Asforotherprotocolsandsys

t
e
m
s
m
a
n
a
g
e
d
b
y
T
h
i
n
k
n
x
,
i
t
i
s
p
o
s
s
i
b
l
e
t
o
u
s
e
a
c
o
m
m
a
n
d
c
o
m
i
n

g
f
r
o
m
a
d
e
v
i
c
e
c
o
n
n
e
c
t
e
d
t
o
o
n
e
H
o
m
e
G
a
t
e
w
a
y
(
e
.
g
.
a
p
u
s
h
b
u
t
t

o
n
)
t
o
c
o
n
t
r
o
l
a
d
e
v
i
c
e
o
r
a
n
a
c
t
i
o
n
b
e
l
o
n
g
i
n
g
t
o
a
d
i
ff
e
r
e
n
t
H
o

m
e
G
a
t
e
w
a
y
.

F
o
r
e
x
a
m
p
l
e
:
a
c
o
n
n
e
c
t
e
d
p
u
s
h
b
u
t
t
o
n
(
e
.
g
.
G
W
A
1

2
4
1
)
b
e
l
o
n
g
i
n
g
t
o
o
n
e
H
o
m
e
G
a
t
e
w
a
y
c
a
n
c
o
m
m
a
n
d
:

- a
c
o
n
n
e
c
t
e

d
a
c
t
u
a
t
o
r
(
e
.
g
.
G
W
1
0
8
2
6
)
,

- o
r
t
r
i
g
g
e
r
a
n
o
t
h
e
r
a
c
t
i
o
n
(
e
.
g
.

d
i
s
p
l
a
y
a
m
e
s
s
a
g
e
o
n
a
S
m
a
r
t
P
l
a
t
e
)

b
e
l
o
n
g
i
n
g
t
o
a
d
i
ff
e
r
e
n
t
H

omeGateway.

In the same way, a connected push button to can com

m
a
n
d
a
n
y
a
c
t
u
a
t
o
r
t
h
r
o
u
g
h
o
t
h
e
r
p
r
o
t
o
c
o
l
s
(
e
.
g
.
K
N
X
,
M
o
d
b
u
s

,
e
t
c
.
)
.

T
h
e
o
b
j
e
c
t
u
s
e
d
f
o
r
t
h
i
s
p
u
r
p
o
s
e
i
s
t
h
e
T
h
i
n
k
n
x
U
n
i
v

e r s a l G a t e w a y o b j e c t .

T h e s t e p s a r e :

- D e fi n e a T h i n k n x o b j e

c
t
a
s
s
o
c
i
a
t
e
d
w
i
t
h
t
h
e
d
e
s
i
r
e
d
p
u
s
h
b
u
t
t
o
n
o
f
t
h
e
G
e
w
i
s
s
S
m
a
r

t
H
o
m
e
(
e
.
g
.
a
u
x
i
l
i
a
r
y
c
o
m
m
a
n
d
,
c
o
n
n
n
e
c
t
e
d
p
u
s
h
b
u
t
t
o
n
,
c
o
n
n

e
c
t
e
d
b
a
t
t
e
r
y
l
e
s
s
b
u
t
t
o
n
)

- D
e
fi
n
e
a
**U
n
i
v
e
r
s
a
l
G
a
t
e
w
a
y**
o
b
j
e
c

tactivatedbytheconnectedcommandandandconfigure,asitsact

ion, the desired actuation on a device belonging to a different en

t
H
o
m
e
G
a
t
e
w
a
y

E x a m p l e : c o m m a n d b e t w e e n t w

o d i ff e r e n t H o m e G a t e w a y s

I n t h i s e x a m p l e , w h e n

a
p
u
s
h
b
u
t
t
o
n
w
i
r
e
d
t
o
a
n
a
u
x
i
l
i
a
r
y
i
n
p
u
t
o
f
a
c
o
n
n
e
c
t
e
d
d
e
v
i
c

e
i
s
p
r
e
s
s
e
d
,
a
m
e
s
s
a
g
e
i
s
d
i
s
p
l
a
y
e
d
o
n
a
S
m
a
r
t
P
l
a
t
e
b
e
l
o
n
g
i
n

g
t
o
a
d
i
ff
e
r
e
n
t
H
o
m
e
G
a
t
e
w
a
y
.

S
t
e
p
1
-
D
e
fi
n
e
t
h
e
A
n
a
l
o
g

Value object

Define a Thinknx **Analog Value** object associa

t
e
d
w
i
t
h
t
h
e
a
u
x
i
l
i
a
r
y
i
n
p
u
t
o
f
t
h
e
c
o
n
n
e
c
t
e
d
d
e
v
i
c
e
t
o
w
h
i
c
h

the push button to is wired.

Configure the following parameter

s
:

- **N a m e** – n a m e o f t h e o b j e c t
- **D e v i c e** – s e l e c t t h e c o n n e c t e d d

e
v
i
c
e
a
n
d
t
h
e
a
u
x
i
l
i
a
r
y
i
n
p
u
t
u
s
e
d
f
o
r
t
h
e
c
o
m
m
a
n
d
(
i
n
t
h
i
s
e
x

ample:auxiliaryinput1ofthedevice“Light”)

- **Reportedpr**

o
p
e
r
t
y
–
s
e
l
e
c
t
t
h
e
p
r
o
p
e
r
t
y
t
h
a
t
t
r
i
g
g
e
r
s
t
h
e
c
o
m
m
a
n
d
(
f
o
r
t

h
i
s
e
x
a
m
p
l
e
,
G
e
n
e
r
i
c
i
s
s
u
ffi
c
i
e
n
t
)

- **K**
N
X
d
a
t
a
t
y
p
e
–
d
e
fi
n
e
t
h
e
d

a
t
a
t
y
p
e
u
s
i
n
g
K
N
X
d
a
t
a
t
y
p
e
s
(
i
n
t
h
i
s
e
x
a
m
p
l
e
t
h
e
v
a
l
u
e
i
s
a
c
o

m
m
a
n
d
,
s
o
**B
o
o
l
e
a
n**
i
s
s
u
i
t
a
b
l
e
)

- **K
N
X
v
a
l
u
e
g
r
o
u
p**
–
n
o
t
u
s
e
d
i
n
t

his example

- **KNX value feedback** – KNX group address associat

e
d
w
i
t
h
t
h
i
s
c
o
m
m
a
n
d

W
h
e
n
e
v
e
r
t
h
e
p
u
s
h
b
u
t
t
o
n
i
s
p
r
e
s
s
e
d
o
r
r

e
l
e
a
s
e
d
,
a
t
e
l
e
g
r
a
m
i
s
s
e
n
t
o
n
t
h
i
s
K
N
X
a
d
d
r
e
s
s
:

- v
a
l
u
e
1
w
h
e

n
t
h
e
b
u
t
t
o
n
i
s
p
r
e
s
s
e
d

- v
a
l
u
e
0
w
h
e
n
t
h
e
b
u
t
t
o
n
i
s
r
e
l
e
a
s
e
d

T

hisKNXaddresswillbeusedtotriggertheUniversalGateway

o
b
j
e
c
t
d
e
fi
n
e
d
i
n
t
h
e
n
e
x
t
s
t
e
p
.

S t e p 2 – D e fi n e t h e U n i

v e r s a l G a t e w a y o b j e c t

D e fi n e t h e **U n i v e r s a l G a t e w a y** o b j

e
c
t
t
h
a
t
a
c
t
i
v
a
t
e
s
t
h
e
d
e
s
i
r
e
d
a
c
t
i
o
n
o
n
a
d
e
v
i
c
e
b
e
l
o
n
g
i
n
g
t

o
a
d
i
ff
e
r
e
n
t
H
o
m
e
G
a
t
e
w
a
y
(
i
n
t
h
i
s
e
x
a
m
p
l
e
:
d
i
s
p
l
a
y
i
n
g
a
m
e
s
s

a g e o n a S m a r t P l a t e) .

- G o t o t h e **S y s t e m** s e c t i o n
- A d d a **U n i v e r**

s a l G a t e w a y o b j e c t

A s i n g l e U n i v e r s a l G a t e w a y o b j e c t c a n

manageupto**threeindependentactions**.Inthisexample,onl

y
o
n
e
a
c
t
i
o
n
i
s
u
s
e
d
:
C
1
.

S
t
e
p
3
-
C
o
n
fi
g
u
r
e
t
h
e
U
n
i
v
e

rsalGatewayparameters

Configure the Universal Gatew

a y o b j e c t a s f o l l o w s :

- **N a m e** – n a m e o f t h e U n i v e r s a l G a t e w a y o

b
j
e
c
t

- **K
N
X
g
r
o
u
p** – K
N
X
g
r
o
u
p
a
d
d
r
e
s
s
o
n
w
h
i
c
h
t
h
e
t
r
i
g
g
e
r
i
n
g
t

e
l
e
g
r
a
m
i
s
r
e
c
e
i
v
e
d
(
i
n
t
h
i
s
e
x
a
m
p
l
e
:
6
/
6
/
6
,
t
h
e
s
a
m
e
a
d
d
r
e
s

s u s e d b y t h e A n a l o g V a l u e o b j e c t)

- **D a t a t y p e** – d e fi n e t h e d a t a t

ypeusingKNXdatatypes(inthisexample:**Boolean**)

- **C1enabl**

i
n
g
–
e
n
a
b
l
e
a
c
t
i
o
n
C
1

- **C**
1
c
o
m
m
a
n
d
–
s
e
l
e
c
t
t
h
e
c
o
m
m
a
n
d
/
a
c
t
i
o

n
t
o
b
e
e
x
e
c
u
t
e
d
(
i
n
t
h
i
s
e
x
a
m
p
l
e
:
d
i
s
p
l
a
y
t
h
e
m
e
s
s
a
g
e
“
B
u
o
n
g

i
o
r
n
o
”
s
c
r
o
l
l
i
n
g
f
r
o
m
l
e
f
t
t
o
r
i
g
h
t
o
n
a
S
m
a
r
t
P
l
a
t
e
,
w
i
t
h
t
h
e

LEDstripblinkinggreenat1Hz)

Configurethehealthcheckinterval

condition:

- **C1 condition** – select **Send if equal**
- **C1 value** – s

e
t
v
a
l
u
e
t
o
1

W
i
t
h
t
h
i
s
c
o
n
fi
g
u
r
a
t
i
o
n
,
t
h
e
m
e
s
s
a
g
e
i
s
d
i
s
p
l
a
y

e
d
o
n
t
h
e
S
m
a
r
t
P
l
a
t
e
w
h
e
n
t
h
e
v
a
l
u
e
r
e
c
e
i
v
e
d
f
r
o
m
t
h
e
K
N
X
a
d
d
r

e
s
s
6
/
6
/
6
i
s
1
,
t
h
a
t
i
s
w
h
e
n
t
h
e
p
u
s
h
b
u
t
t
o
n
c
o
n
n
e
c
t
e
d
t
o
t
h
e
a
u

x
i
l
i
a
r
y
i
n
p
u
t
i
s
p
r
e
s
s
e
d
.

C o m m a n d o f a G e w i s s S m a r t H

o
m
e
d
e
v
i
c
e
i
n
a
S
c
e
n
e

A
T
h
i
n
k
n
x
S
c
e
n
e
o
b
j
e
c
t
c
a
n
a
l
s
o
m
a
n

agecommandsandactionsonGewissSmartHomedevices.Withi

n
a
S
c
e
n
e
i
t
i
s
p
o
s
s
i
b
l
e
t
o
i
n
c
l
u
d
e
b
o
t
h
:

- c
o
m
m
a
n
d
s
f
o
r
c
o
n
n
e

connected devices (lights, shutters, actuators),

- and actions

u
c
h
a
s
m
e
s
s
a
g
e
s
d
i
s
p
l
a
y
e
d
o
n
o
n
e
o
r
m
o
r
e
S
m
a
r
t
P
l
a
t
e
s
.

T
h
e
s
t

e
p
s
t
o
d
e
fi
n
e
a
S
c
e
n
e
o
b
j
e
c
t
a
r
e
:

- D
e
fi
n
e
a
T
h
i
n
k
n
x
S
c
e
n
e
o
b
j
e
c

t
a
t
S
y
s
t
e
m
l
e
v
e
l
,
d
e
fi
n
i
n
g
t
h
e
l
i
s
t
o
f
a
c
t
i
o
n
s
t
o
b
e
e
x
e
c
u
t
e
d
,

i
n
c
l
u
d
i
n
g
c
o
m
m
a
n
d
s
/
a
c
t
i
o
n
s
f
o
r
G
e
w
i
s
s
S
m
a
r
t
H
o
m
e
d
e
v
i
c
e
s

Define how the Scene is triggered (e.g. by an alarm from a connne

c
t
e
d
d
e
v
i
c
e
o
r
b
y
a
b
u
t
t
o
n
o
n
a
s
u
p
e
r
v
i
s
i
o
n
p
a
g
e
)

T
h
e
f
o
l
l
o
w
i

n
g
e
x
a
m
p
l
e
d
e
s
c
r
i
b
e
s
a
S
c
e
n
e
t
h
a
t
i
n
c
l
u
d
e
s
a
c
t
i
o
n
s
o
n
:

- a
r
o

l
l
e
r
s
h
u
t
t
e
r
,

- a
d
i
m
m
e
r
o
f
t
h
e
G
e
w
i
s
s
S
m
a
r
t
H
o
m
e
,

a
n
d
i
s
a
c
t
i

v
a
t
e
d
b
o
t
h
b
y
:

- a
w
i
n
d
o
w
a
l
a
r
m
f
r
o
m
G
e
w
i
s
s
S
m
a
r
t
H
o
m
e
,
- a
n
d
a

b
u
t
t
o
n
o
n
a
s
u
p
e
r
v
i
s
i
o
n
p
a
g
e
.

S
t
e
p
1
–
D
e
fi
n
e
t
h
e
S
c
e
n
e
o

b
j
e
c
t

D
e
fi
n
e
t
h
e
T
h
i
n
k
n
x
S
c
e
n
e
o
b
j
e
c
t
.

- G
o
t
o
t
h
e
S
y
s
t
e
m
s
e

c
t
i
o
n

- A
d
d
a
S
c
e
n
e
o
b
j
e
c
t

  -
  C
o
n
fi
g
u
r
e
t
h
e
f
o
l
l
o
w
i
n
g
p
a
r
a
m
e
t

e
r
s
:

- **N a m e** – n a m e o f t h e S c e n e o b j e c t
- **K N X g r o u p** – K N X g r o u p a d

dressusedtotriggertheScene(theScenecanbetriggeredby

anyThinknxfunctionorbyanydevicewritingtothisgroup)

KNXdatatype – data type of the value written to the KNX group

Triggervalue – value written to the KNX group that triggers

h
e
S
c
e
n
e

- **R**
e
c
o
r
d – if enabled, the user can add and modify activi

onsautonomously(withoutsystemintegratorintervention

)

- **Restartable** – if enabled, the Scene can be retriggered before

o
r
e
t
h
e
p
r
e
v
i
o
u
s
e
x
e
c
u
t
i
o
n
h
a
s
fi
n
i
s
h
e
d

- **H
i
d
e
c
o
n
fi
g
u
r
a
t
i
o
n
a**

c
t
i
o
n
s
–
i
f
e
n
a
b
l
e
d
,
a
c
t
i
o
n
s
d
e
fi
n
e
d
d
u
r
i
n
g
c
o
n
fi
g
u
r
a
t
i
o
n
a
r
e

notvisibletotheuser

- **Collection** – list of actions execute

d
b
y
t
h
e
S
c
e
n
e

F
o
r
e
a
c
h
a
c
t
i
o
n
i
n
t
h
e
C
o
l
l
e
c
t
i
o
n
,
t
h
e
c
o
n
fi
g
u
r

a
t
o
r
p
r
o
v
i
d
e
s
a
s
i
m
p
l
e
n
a
v
i
g
a
t
i
o
n
t
r
e
e
t
o
i
d
e
n
t
i
f
y
t
h
e
o
b
j
e
c

t
t
o
b
e
c
o
m
m
a
n
d
e
d
a
m
o
n
g
a
l
l
o
b
j
e
c
t
s
d
e
fi
n
e
d
i
n
t
h
e
p
r
o
j
e
c
t
.

F
o

r
G
e
w
i
s
s
S
m
a
r
t
H
o
m
e
d
e
v
i
c
e
s
,
s
e
l
e
c
t
i
n
t
h
e
n
a
v
i
g
a
t
i
o
n
t
r
e
e
:

- t h e H o m e G a t e w a y t o w h i c h t h e d e v i c e b e l o n g s ,
- t h e d e v i c e ,
-

t
h
e
a
c
t
i
o
n
/
c
o
m
m
a
n
d
t
o
b
e
e
x
e
c
u
t
e
d
o
n
t
h
e
d
e
v
i
c
e
.

I
n
t
h
i
s
e
x
a

m
p
l
e
,
t
h
e
S
c
e
n
e
i
n
c
l
u
d
e
s
:

- a
c
o
m
m
a
n
d
o
n
a
r
o
l
l
e
r
s
h
u
t
t
e
r
:
c
o

m
m
a
n
d
o
f
m
o
v
i
n
g
u
p

- a
c
o
m
m
a
n
d
o
n
a
l
i
g
h
t
(
O
N
/
O
F
F
o
r
d
i
m
m
i
n
g
)
:
c

o
m
m
a
n
d
t
o
s
w
i
t
c
h
O
N

**S
t
e
p
2
-
T
r
i
g
g
e
r
t
h
e
S
c
e
n
e
f
r
o**

mawindowsensor

Defineathinknx **AnalogValue** object class

o
c
i
a
t
e
d
w
i
t
h
t
h
e
a
l
a
r
m
d
e
t
e
c
t
e
d
b
y
t
h
e
w
i
n
d
o
w
s
e
n
s
o
r
.

C
o
n
fi
g

u
r
e
t
h
e
f
o
l
l
o
w
i
n
g
p
a
r
a
m
e
t
e
r
s
:

- **N**
a
m
e
–
n
a
m
e
o
f
t
h
e
o
b
j
e
c
t
- **D**
e

vice – select the window sensor that triggers the Scene

- **Repo**

r t e d p r o p e r t y – s e l e c t **C o n t a c t s w i t c h** (a l a r m c o n d i t i o n)

- **K N**

X d a t a t y p e – d e fi n e t h e d a t a t y p e u s i n g K N X d a t a t y p e s (i n t h i s e

x
a
m
p
l
e
i
t
i
s
a
n
a
l
a
r
m
,
s
o
B
o
o
l
e
a
n
i
s
s
u
i
t
a
b
l
e
)

- **K**
N
X
v
a
l
u
e
g
r
o
u

p – not used in this example

- **KNX value feedback** – insert the KN

Xgroupaddressthattriggersthe Scene (in this example: **7/7**

/
7
,
previously defined for the Scene)

When the window sensor

detects an open window, the telegrams sent on this KNX address

t
r
i
g
g
e
r
s
t
h
e
S
c
e
n
e
.

**S
t
e
p
3
-
T
r
i
g
g
e
r
t
h
e
S
c
e
n
e
f
r
o**

masupervisionpage

Add a **SceneCommand** object to the su

p
e
r
v
i
s
i
o
n
p
a
g
e
.

C
o
n
fi
g
u
r
e
t
h
e
f
o
l
l
o
w
i
n
g
p
a
r
a
m
e
t
e
r
s
:

- **L**
a
b

e
l
–
n
a
m
e
o
f
t
h
e
S
c
e
n
e
C
o
m
m
a
n
d
o
b
j
e
c
t
s
h
o
w
n
o
n
t
h
e
s
u
p
e
r
v
i
s
i
o
n

page

- **Scene object** – name of the Scene object (System object)

to be triggered (in this example: **Scenario**, previously defin

e
d
)

W
h
e
n
t
h
e
u
s
e
r
p
r
e
s
s
e
s
t
h
e
S
c
e
n
e
C
o
m
m
a
n
d
b
u
t
t
o
n
o
n
t
h
e
s
u
p

e
r
v
i
s
i
o
n
p
a
g
e
,
t
h
e
a
s
s
o
c
i
a
t
e
d
S
c
e
n
e
i
s
e
x
e
c
u
t
e
d
.

o

TADeviceUpdate

Afterdeploying the proj

e
c
t
,
l
o
c
a
l
l
y
o
r
o
n
t
h
e
c
l
o
u
d
,
o
p
e
n
t
h
e
T
h
i
n
K
N
X
c
l
i
e
n
t
.

S
e
l
e
c

t
w
h
e
t
h
e
r
t
o
l
o
a
d
t
h
e
p
r
o
j
e
c
t
l
o
c
a
l
l
y
o
r
o
n
t
h
e
c
l
o
u
d
u
s
i
n
g
o
n
e

of the options shown in the following image.

- If the project

c
t
i
s
l
o
a
d
e
d
o
n
t
h
e
c
l
o
u
d
,
e
n
t
e
r
t
h
e
s
e
r
v
e
r
s
e
r
i
a
l
n
u
m
b
e
r
a
n
d
t

h
e
c
l
o
u
d
p
a
s
s
w
o
r
d
.

I
f
t
h
e
p
r
o
j
e
c
t
i
s
l
o
a
d
e
d
l
o
c
a
l
l
y
,
e
n
t
e
r
t

h
e
C
o
n
fi
g
u
r
a
t
o
r
U
R
L
,
v
i
s
i
b
l
e
i
n
t
h
e
C
o
n
fi
g
u
r
a
t
o
r
,
p
r
e
c
e
d
e
d
b
y

http:(seethexampleinthefigurewheretheURLisfirstshown

i n t h e C o n fi g u r a t o r a n d t h e n h o w t o e n t e r i t i n t h e c l i e n t) .

O
n
c
e
t
h
e
p
r
o
j
e
c
t
i
s
l
o
a
d
e
d
,
o
p
e
n
i
t
a
n
d
c
l
i
c
k
o
n
t
h
e
f
u
n
c
t
i

onsthatwerecreatedinthe Configurator.

Scrolldownun

t
i
l
y
o
u
r
e
a
c
h
t
h
e
s
e
t
t
i
n
g
s
s
e
c
t
i
o
n
,
w
h
e
r
e
t
h
e
t
h
r
e
e
d
o
t
s
a
r
e
d

isplayednexttoit,andclickonthem.

- In the page that opens, top

e
n
s
,
c
l
i
c
k
o
n
t
h
e
O
T
A
m
a
n
a
g
e
r
a
n
d
l
o
g
i
n
.
A
l
l
s
y
s
t
e
m
s
a
v
a
i
l
a
b
l

e
o
n
t
h
e
s
e
r
v
e
r
w
i
l
l
b
e
d
i
s
p
l
a
y
e
d
.
S
e
l
e
c
t
t
h
e
s
y
s
t
e
m
a
n
d
t
h
e
d
e

vicesyouwanttoupdatewithintheselectedsystem.

N.B

. I f t h e r e a r e n o d e v i c e s a v a i l a b l e f o r u p d a t e i n t h e s y s t e m , t

h
e
m
e
s
s
a
g
e
“
N
o
d
e
v
i
c
e
f
o
u
n
d
”
w
i
l
l
b
e
d
i
s
p
l
a
y
e
d
.

From:
http://www.thinknx.com/wiki/

-

LearningThinknx

Permanent link:
http://www.thinknx.com/wiki/doku.php?id=gw_manager

Last update:
2026/09/10 11:01