



INTEL-IRRIS

Intelligent Irrigation System for Low-cost Autonomous Water Control
in Small-scale Agriculture



Intelligent Irrigation System for Low-cost Autonomous Water Control in Small-scale Agriculture



Building the INTEL-IRRIS LoRa IoT platform Part 2: edge-enabled gateway



Prof. Congduc Pham
<http://www.univ-pau.fr/~cpham>
Université de Pau, France



Review: Technology components



Review: Low-cost sensors



- Build on low-cost, low-power IoT expertise
- Increase accuracy of low-cost sensors by automatic and remotely controlled procedures for advanced calibration
- Enable deployment of several complementary low-cost sensors
- Include agricultural models / knowledge with corrective & predictive analytics

Review: Smart embedded control

- Build on low-cost embedded & open IoT gateway expertise
- Implement the “Intelligent Irrigation in-the-box” with "plug-&-sense" approach
- Model complex water-soil-plant interaction
- Embed Decision Support System (DSS) and disruptive Artificial Intelligence (AI)
- Integration of various knowledge streams
- Fully autonomous



INTEL-IRRIS starter-kit

- "Intelligent Irrigation in-the-box", "plug-&-sense"
- From idea to reality!



WaziGate

- WaziGate is an IoT LoRa Gateway developed by WAZIUP
- WaziGate implements the edge-enabled IoT gateway approach
 - customized applications can be directly hosted in the gateway
 - the gateway can easily work without Internet connectivity
 - data are available to end-users in an embedded database
 - web-based visualization module provides graphical user interface
- You can find all the WaziGate documentation on the [WaziGate documentation page](#). There are 4 main sections describing the generic WaziGate main features:
 - [Quick start:](#) https://www.waziup.io/documentation/wazigate/v2/quick_start/
 - [Installation:](#) <https://www.waziup.io/documentation/wazigate/v2/install/>
 - [LoRaWAN:](#) <https://www.waziup.io/documentation/wazigate/v2/lorawan/>
 - [WaziApps:](#) <https://www.waziup.io/documentation/wazigate/v2/waziapps/>
 - Look at the generic installation video: <https://youtu.be/DvGdmdsGZHA>

Install your gateway (1)

● INTEL-IRRIS gateway distribution (RPI3B/3B+/4B)

- comes pre-configured with a soil sensor device
- will work out-of-the box with the INTEL-IRRIS soil sensor device
- will be updated to host the INTEL-IRRIS irrigation application
- Download the INTEL-IRRIS gateway SD card image from project website
- <http://intel-iris.eu/results>
- Image uses EU868 frequency band
- EU533 can be configured afterwards
- Flash the SD card image on 8GB SD card class 10

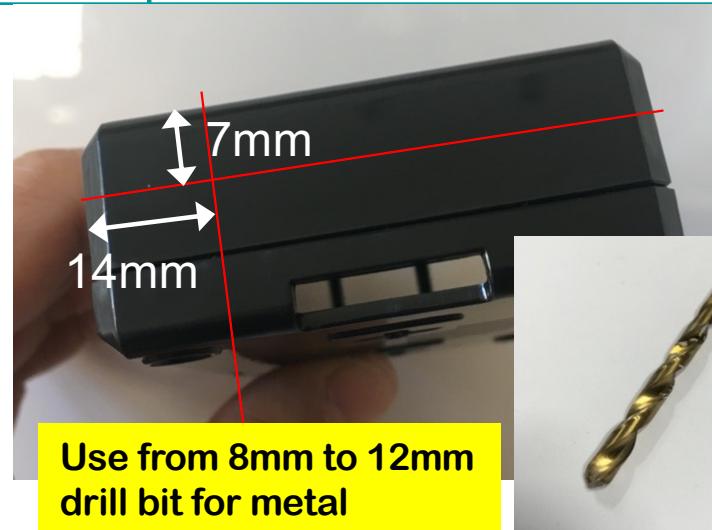


Look at the INTEL-IRRIS gateway video

Video n°4 at t=124s: <https://youtu.be/j-1Nk0tv0xM?t=124>

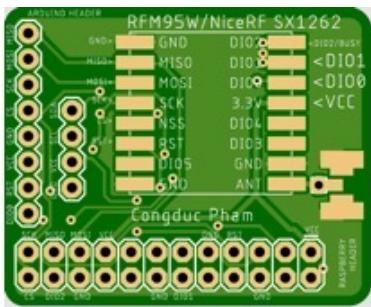
Install your gateway (2)

- Recommended RPI model is RPI3B
 - plug in the radio LoRa hat
 - The OLED screen is optional but highly recommended
- Drill a hole for the SMA connector on the RPI enclosure, at the SD card slot side
- <https://fr.aliexpress.com/item/32718435597.html>

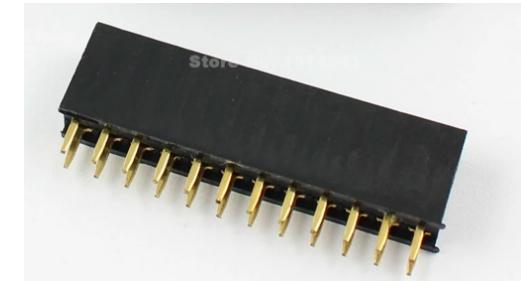
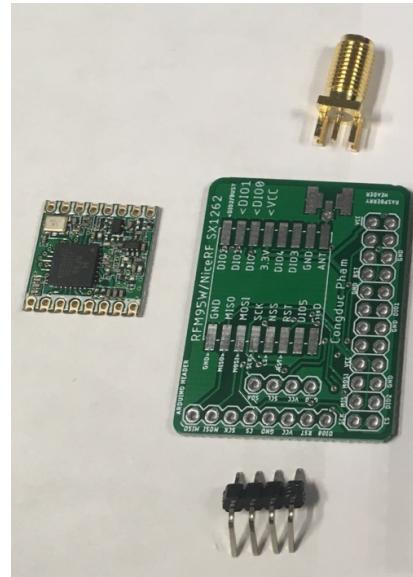


The LoRa radio hat

<https://github.com/CongducPham/PRIMA-Intel-Irris/blob/main/Tutorials/Intel-Irris-PCB.pdf>



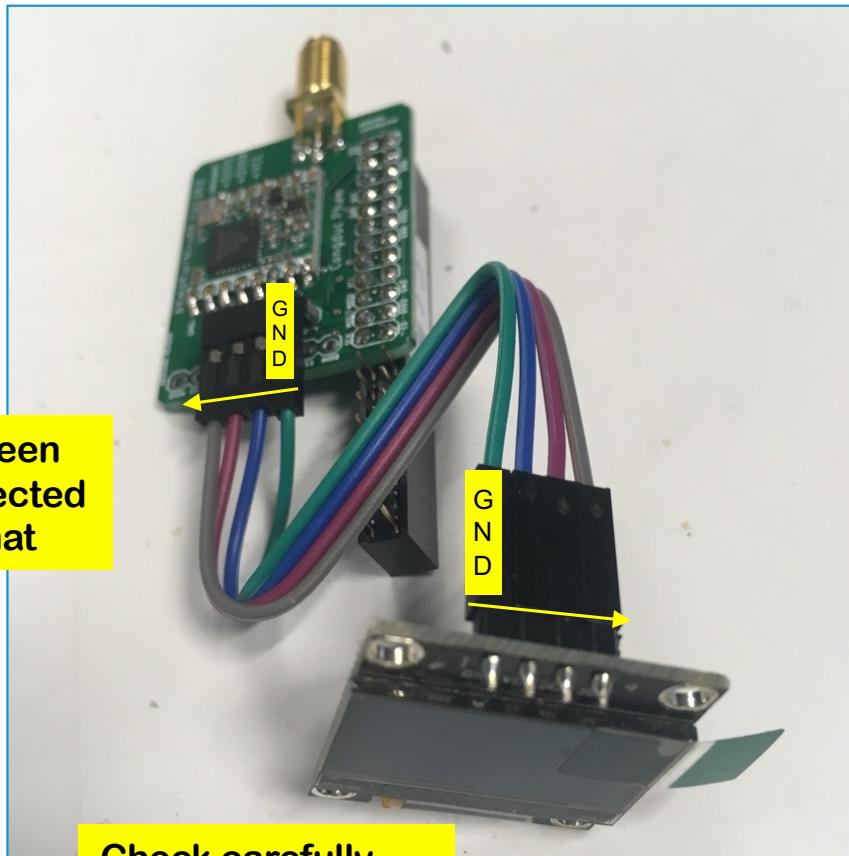
RFM95W/NiceRF SX1262
breakout



2-row 12-pin header
or
2 X 1-row 12-pin header



Installing OLED

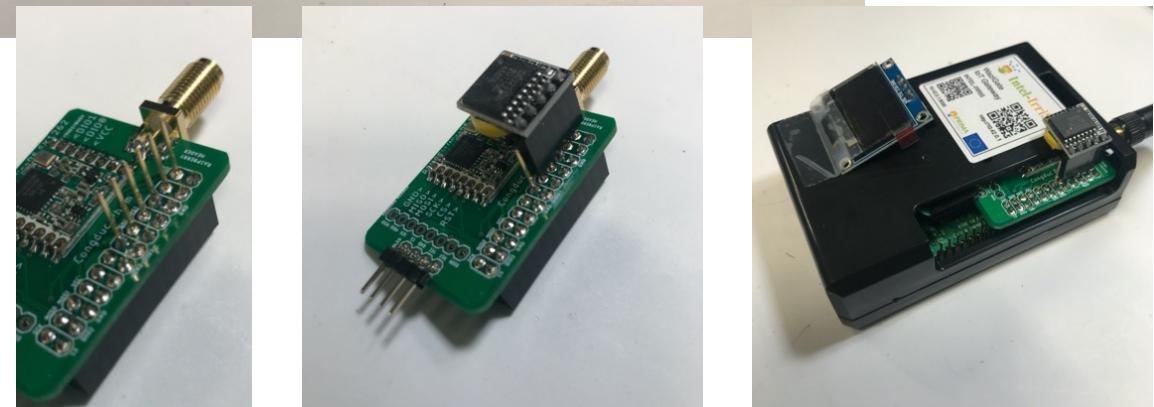
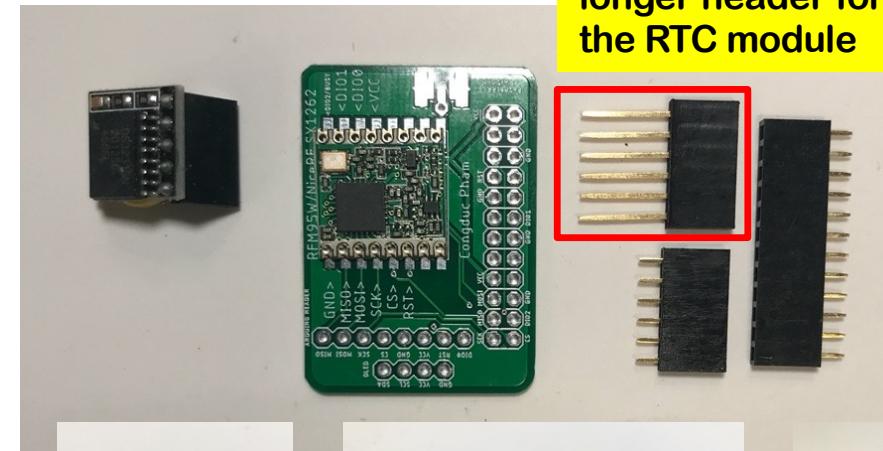


Connecting an RTC module

- Several possibilities depending on availability of components



Solder a header with longer pins to connect the RTC module. You can then cut the remaining pins if you want



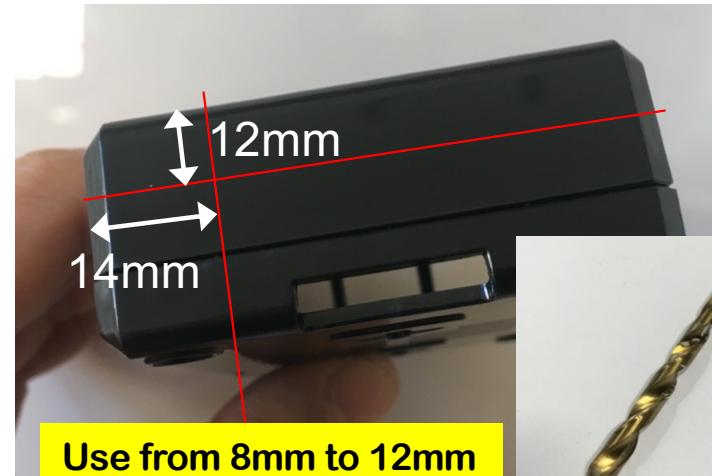
With WAZIUP LoRa gateway hat



LoRa gateway hat
with on-board RTC,
no need for external
RTC module



An OLED screen can
directly be connected
to the radio hat



Gateway power consumption



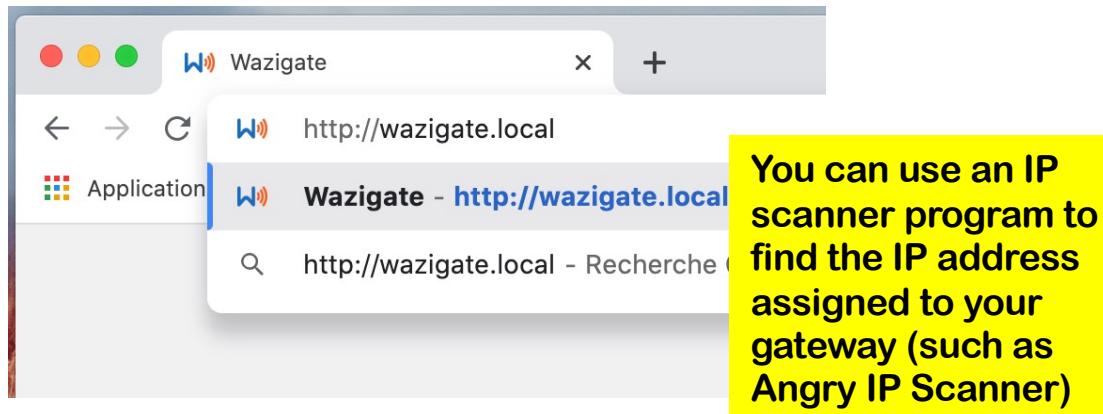
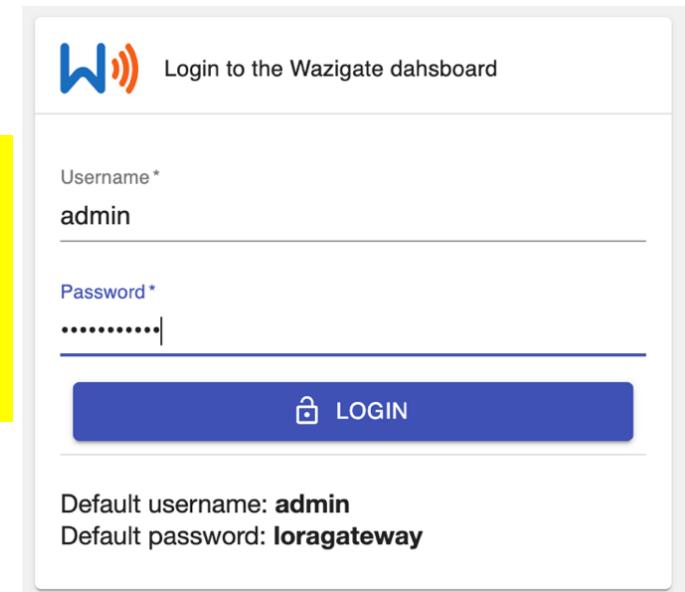
RPI3B consumes less than 300mA

There can be peak up to 800mA when booting

However, a 2.5A micro USB charger is needed to avoid undervoltage

Checking the gateway (Ethernet)

- Connect the gateway to your laptop which has Internet
- Enable Internet sharing, laptop provides IP address to gateway
- Power the gateway, wait 3-4mins for boot process
- Open web navigator. Go to <http://wazigate.local> or use IP address

Login to the Wazigate dashboard

Username*
admin

Password*
.....|

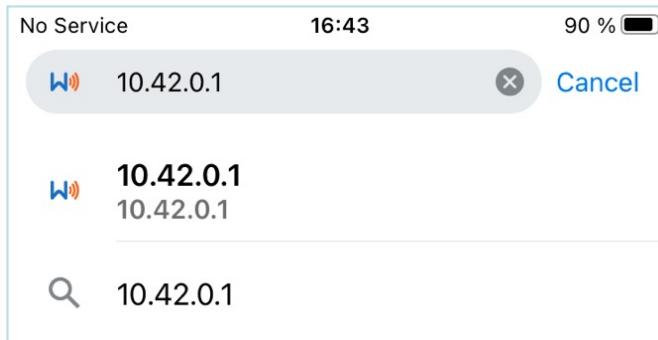
LOGIN

Default username: admin
Default password: loragateway

- Use default login to connect
 - User: admin
 - Password: loragateway

Checking the gateway (WiFi)

- Use a smartphone to check access to gateway through WiFi
- Connect to WAZIGATE_XXXXXXXXXXXX WiFi network
 - default WiFi password is loragateway
- Open web navigator. Go to <http://wazigate.local> or <http://10.42.0.1> or flash QR code



Login to the Wazigate dashboard

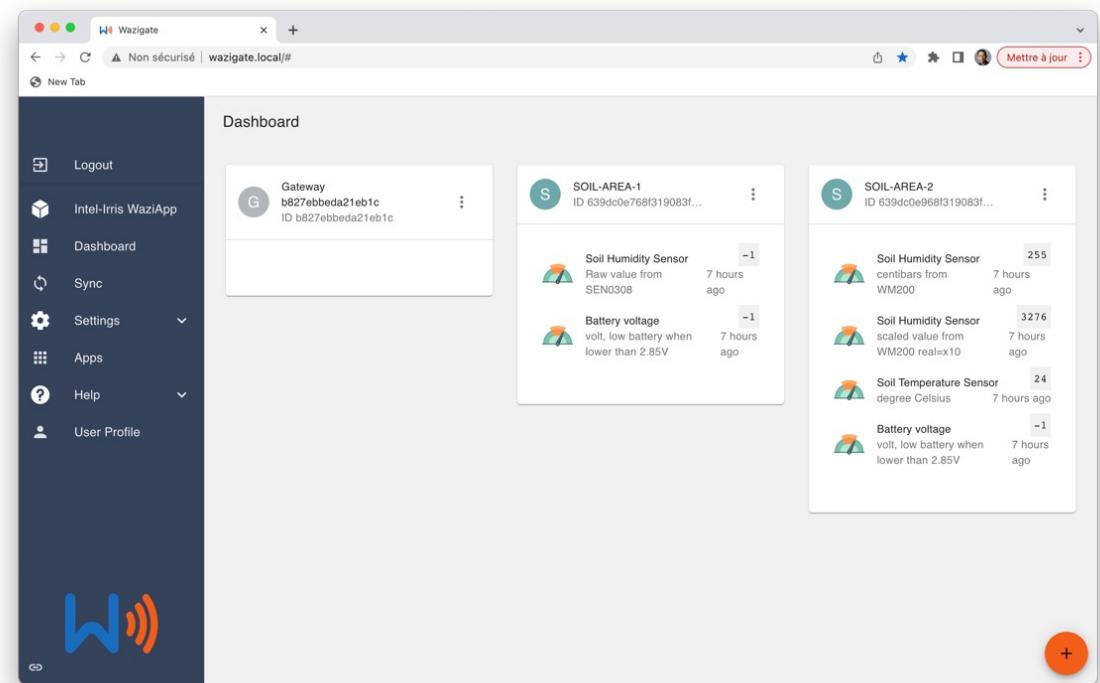
Username *	<input type="text" value="admin"/>
Password *	<input type="password" value="....."/>
<input type="button" value="LOGIN"/>	
Default username: admin	
Default password: loragateway	

- Use default login to connect
 - User: admin
 - Password: loragateway

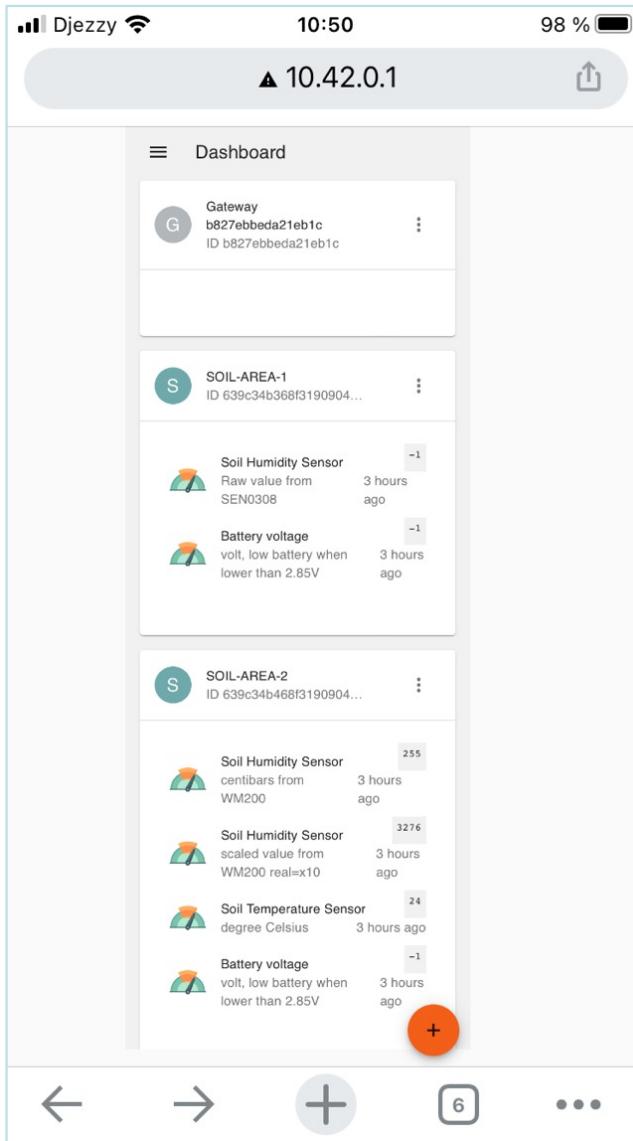
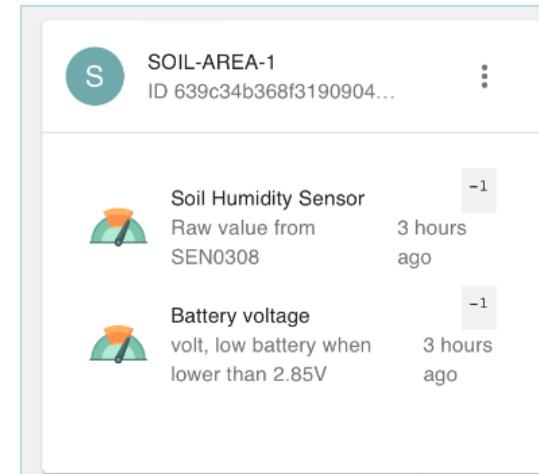
Default gateway configuration (1)

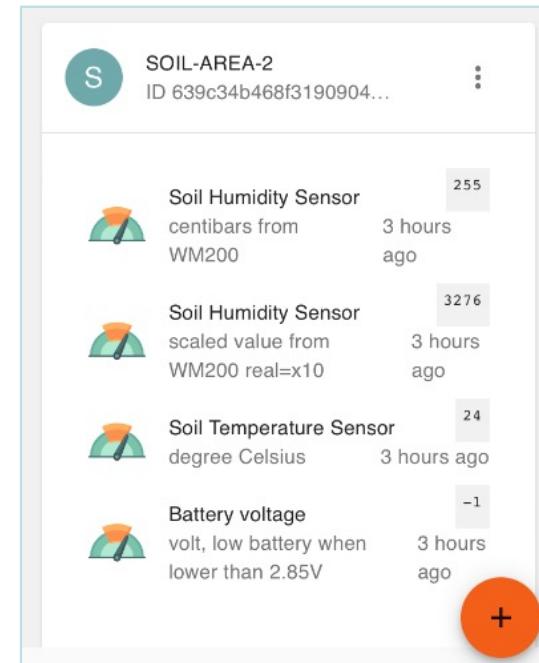


- For the starter-kit, the INTEL-IRRIS gateway will be ready for
 - 1 capacitive sensor named SOIL-AREA-1 with address 26011DAA
 - 1 tensiometer sensor named SOIL-AREA-2 with address 26011DB1
- Capacitive device will show humidity and battery values
- Tensiometer device will show centibar, raw resistance, soil temperature and battery values



Default gateway configuration (2)



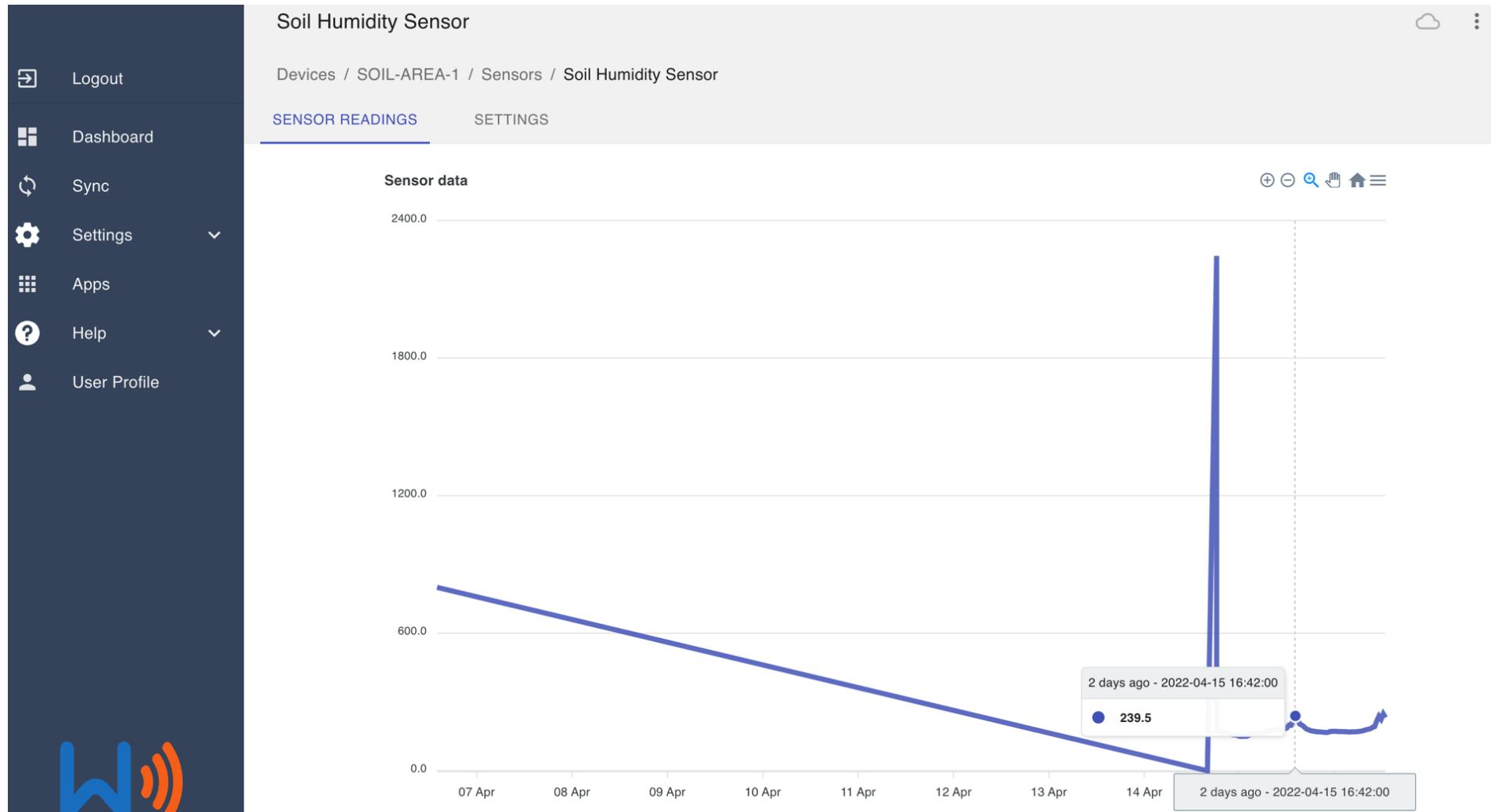
Default values for the SEN0308 capacitive sensor



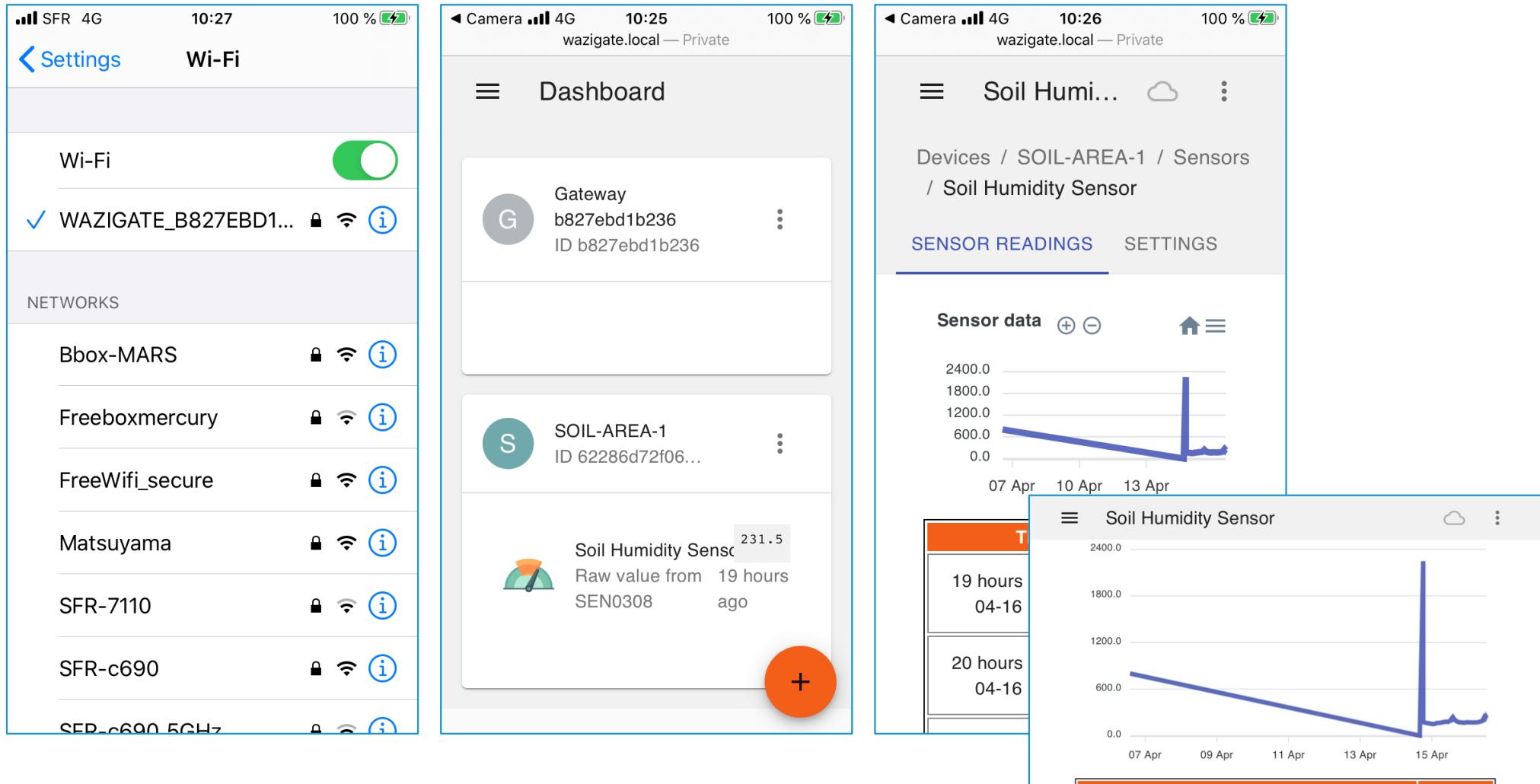
Default values for the WM200 tensiometer sensor



Display sensor data



All these steps with a smartphone



The figure displays three mobile application screens illustrating the integration of various smart home and IoT components:

- Wi-Fi Settings Screen:** Shows the device is connected to "WAZIGATE_B827EBD1..." via 4G at 10:27. The battery level is 100%. The screen lists several available networks including "Bbox-MARS", "Freeboxmercury", "FreeWiFi_secure", "Matsuyama", "SFR-7110", "SFR-c690", and "SFR_c690_5GHz".
- Device Dashboard Screen:** Shows the device is connected to "wazigate.local" via 4G at 10:25. The screen displays a "Dashboard" section with a "Gateway" entry (ID b827ebd1b236) and a "SOIL-AREA-1" entry (Raw value from SEN0308 231.5, 19 hours ago). A large orange "+" button is visible at the bottom right.
- Sensor Data Screen:** Shows the device is connected to "wazigate.local" via 4G at 10:26. The screen displays "Soil Humi..." data for "SOIL-AREA-1 / Sensors / Soil Humidity Sensor". It shows a "Sensor data" graph from April 7 to April 15, 2016, with values ranging from 0.0 to 2400.0. The graph shows a sharp drop in value around April 14, followed by a recovery. Below the graph, a sidebar shows "19 hours 04-16" and "20 hours 04-16". A detailed view of the "Soil Humidity Sensor" data for April 15 is shown in a modal window.

QR code for connecting to WiFi

- The gateway WiFi is WAZIGATE_XXXXXXXXXXXX where XXXXXXXXXXXX is the MAC address of the RPI
- For instance WAZIGATE_B827EBD1B236
- With the OLED, a QR code for joining the WiFi network is dynamically generated at boot time and displayed for 10s before the main screen so that users can automatically join with a smartphone
- Then, users can scan the static QR code on the gateway sticker to connect to the gateway's dashboard or the INTEL-IRRIS IIWA App



Intelirris_Soil_Sensor | Arduino 1.8.13

```
Intelirris_Soil_Sensor DS18B20.cpp DS18B20.h RadioSettings.h SX126X_RadioSettings.h SX127X_RadioSettings.h SX128X_RadioSettings.h

27 */
28
29曰 ****
30 -----
31 / --\      /--\      /--\      /--\
32 | / \---| / |---| / |---| / |---| / |---|
33 | | / _\| / _\| / _\| / _\| / _\| / _\| / _\|
34 | \_/\_|| / | / | / | / | / | / | / | / | / |
35 \_/\_/_|| / | / | / | / | / | / | / | / |
36           /| |
37           |__/
38 ****
39
40 /////////////////
41 // sends data to INTEL-IRRIS WaziGate edge-gateway
42 #define TO_WAZIGATE
43
44 /////////////////
45 // Frequency band - do not change in SX127X_RadioSettings.h anymore
46 //#define BAND868
47 //#define BAND900
48 #define BAND433
49
50 /////////////////
51 // Test device
52 //#define TEST_DEVICE_RANDOM
53
54 /////////////////
55 // uncomment to have a soil tensiometer watermark sensor
56 //#define WITH_WATERMARK
57
```

Be sure that

```
#define TO_WAZIGATE
```

is uncommented

Configuring for EU433 band

Intelirris_Soil_Sensor | Arduino 1.8.13

```

  DS18B20.cpp DS18B20.h RadioSettings.h SX126X_RadioSettings.h SX127X_RadioSettings.h SX128X_RadioSettings.h

27 /*
28 */
29 //*****
30 //----\ /----\ /----\ /----\
31 // / \ \ / \ / \ / \ / \ / \ / \
32 // | | | | | | | | | | | | | |
33 // | \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ |
34 // | \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ |
35 // \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ |
36 // \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ |
37 // \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ |
38 //*****/
39 /////////////////
40 // sends data to INTEL-IRRIS WaziGate edge-gateway
41 #define TO_WAZIGATE
42
43 /////////////////
44 // Frequency band - do not change in SX127X_RadioSettings.h anymore
45 //##define BAND868
46 //##define BAND900
47 #define BAND433
48
49 /////////////////
50 // Test device
51 ##define TEST_DEVICE_RANDOM
52
53 /////////////////
54 // uncomment to have a soil tensiometer watermark sensor
55 ##define WITH_WATERMARK
56
57

```

48 Arduino Pro or Pro Mini, ATmega328P (3.3V, 8 MHz) on /dev/cu.usbserial-AK05C49Q

If you use the EU433 band

make sure that

#define BAND433

is the only uncommented
band option

Intelirris_Soil_Sensor | Arduino 1.8.13

```
Intelirris_Soil_Sensor DS18B20.cpp DS18B20.h RadioSettings.h SX126X_RadioSettings.h SX127X_RadioSettings.h x12

1/* 
2 * INTEL_IIRRIS soil humidity sensor platform
3 * extended version with AES and custom Carrier Sense features
4 *
5 * Copyright (C) 2016-2021 Congduc Pham, University of Pau, France
6 *
7 * This program is free software: you can redistribute it and/or modify
8 * it under the terms of the GNU General Public License as published by
9 * the Free Software Foundation, either version 3 of the License, or
10 * (at your option) any later version.
11 *
12 * This program is distributed in the hope that it will be useful,
13 * but WITHOUT ANY WARRANTY; without even the implied warranty of
14 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15 * GNU General Public License for more details.
16 *
17 * You should have received a copy of the GNU General Public License
18 * along with the program. If not, see <http://www.gnu.org/licenses/>.
19 *
20 ****
21 * last update: February 2nd, 2022 by C. Pham
22 *
23 * NEW: LoRa communicain library moved from Libelium's lib to StuartProject's lib
24 * https://github.com/StuartsProjects/SX12XX-LoRa
25 * to support SX126X, SX127X and SX128X chips (SX128X is LoRa in 2.4GHz band)
26 *
27 */
28 ****
29 */

30 _____
31 / \ \ \
32 | | | |
33 | | | |
34 | | | |
```

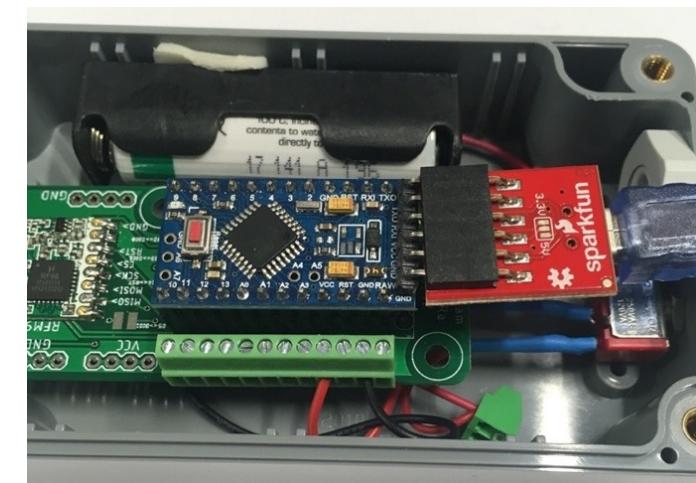
Done uploading.

```
Using library LowPower at version 1.0 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries
Using library OneWire at version 2.3.2 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries
Using library Dallas-Temperature at version 3.7.7 in folder: /Users/cpham/Dropbox/Arduino/sketch
Using library AES-128_V10 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries/AES-128_V10
Using library Base64 at version 1.0.0 in folder: /Users/cpham/Dropbox/Arduino/sketch/libraries
/Users/cpham/Library/Arduino15/packages/arduino/tools/avr-gcc/7.3.0-atmel3.6.1-arduino7/bin/avr
Sketch uses 14208 bytes (46%) of program storage space. Maximum is 30720 bytes.
Global variables use 476 bytes (23%) of dynamic memory, leaving 1572 bytes for local variables.
```

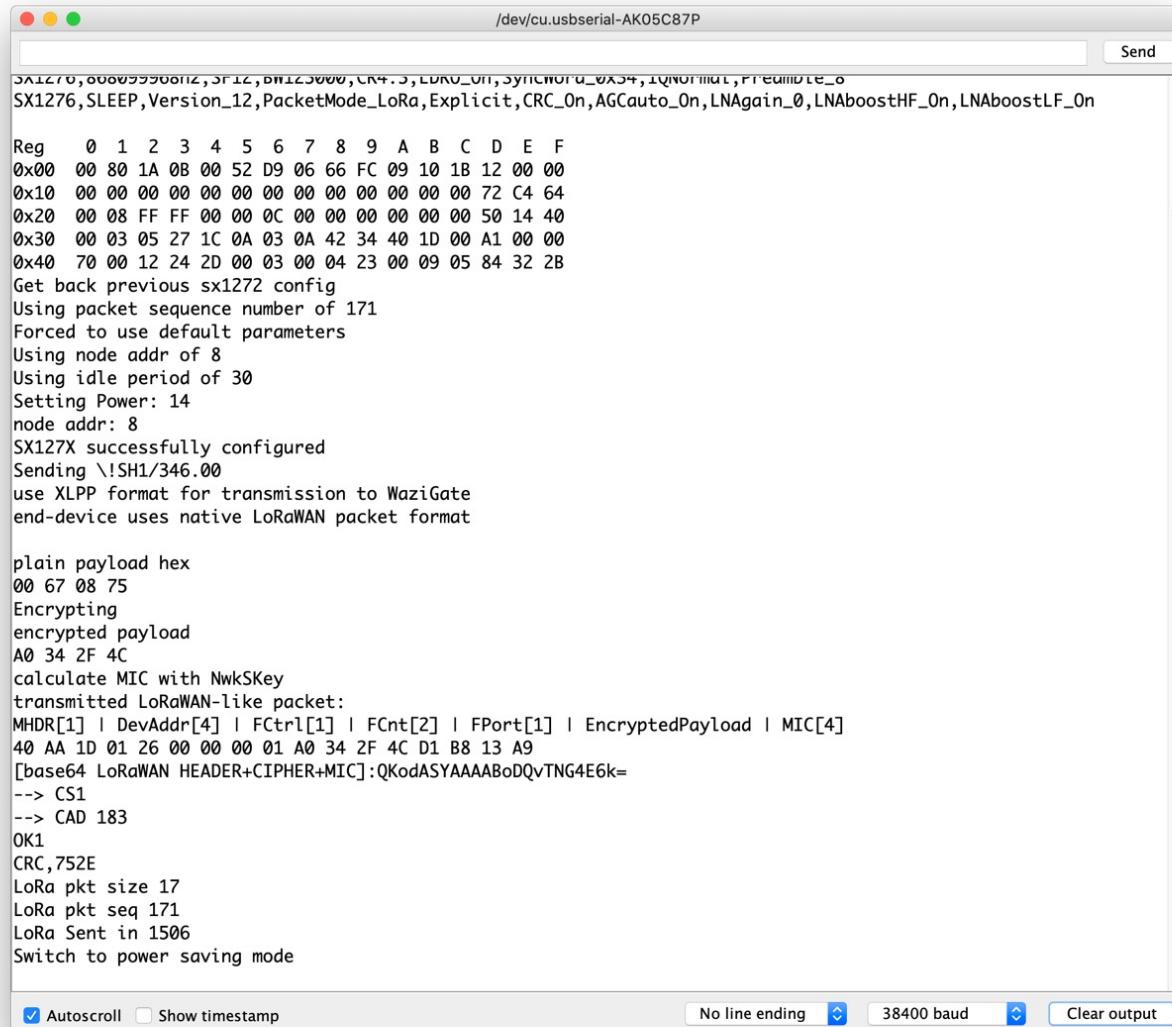
Click on the "upload" button



And wait until upload is completed



Checking that device is operational



```

/dev/cu.usbserial-AK05C87P
Send

SX1276,0000000000000000,3F12,B0123000,CR4.5,LDRU_0H,SYNCH0D_0X34,TQNormal,Preamble_0
SX1276,SLEEP,Version_12,PacketMode_LoRa,Explicit,CRC_On,AGCAuto_On,LNAgain_0,LNAboostHF_On,LNAboostLF_On

Reg 0 1 2 3 4 5 6 7 8 9 A B C D E F
0x00 00 80 1A 0B 00 52 D9 06 66 FC 09 10 1B 12 00 00
0x10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 72 C4 64
0x20 00 08 FF FF 00 00 0C 00 00 00 00 00 00 00 50 14 40
0x30 00 03 05 27 1C 0A 03 0A 42 34 40 1D 00 A1 00 00
0x40 70 00 12 24 2D 00 03 00 04 23 00 09 05 84 32 2B
Get back previous sx1272 config
Using packet sequence number of 171
Forced to use default parameters
Using node addr of 8
Using idle period of 30
Setting Power: 14
node addr: 8
SX127X successfully configured
Sending \!SH1/346.00
use XLPP format for transmission to WaziGate
end-device uses native LoRaWAN packet format

plain payload hex
00 67 08 75
Encrypting
encrypted payload
A0 34 2F 4C
calculate MIC with NwkSKey
transmitted LoRaWAN-like packet:
MHDR[1] | DevAddr[4] | FCtrl[1] | FCnt[2] | FPort[1] | EncryptedPayload | MIC[4]
40 AA 1D 01 26 00 00 00 01 A0 34 2F 4C D1 B8 13 A9
[base64 LoRaWAN HEADER+CIPHER+MIC]:QKodASYAAABoDQvTNG4E6k=
--> CS1
--> CAD 183
OK1
CRC,752E
LoRa pkt size 17
LoRa pkt seq 171
LoRa Sent in 1506
Switch to power saving mode

 Autoscroll  Show timestamp
  No line ending  Clear output

```

Open serial monitor

Set baud rate to 38400

See output from board

**Check that
transmission is OK**

Transmission to gateway



Parameters for
INTEL-IRRIS gateway



SF12BW125
868.1MHz | 433.175MHz
Node id is 26011DAA
1 msg/60mins
1 sensor
XLPP data



This dedicated video will show all these steps, from connecting the SEN0308 to testing transmission to the gateway
Video n°4: <https://youtu.be/j-1Nk0tv0xM>

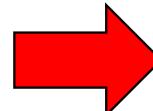


Check data reception on OLED

YOU CAN ALSO VIEW ON DASHBOARD

Default values for the
SEN0308 capacitive sensor

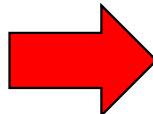
Testing with
capacitive device



Default values for the WM200
tensiometer sensor

VALUES ARE ONLY INDICATIVE

Testing with
tensiometer device



Soil sensor information on OLED

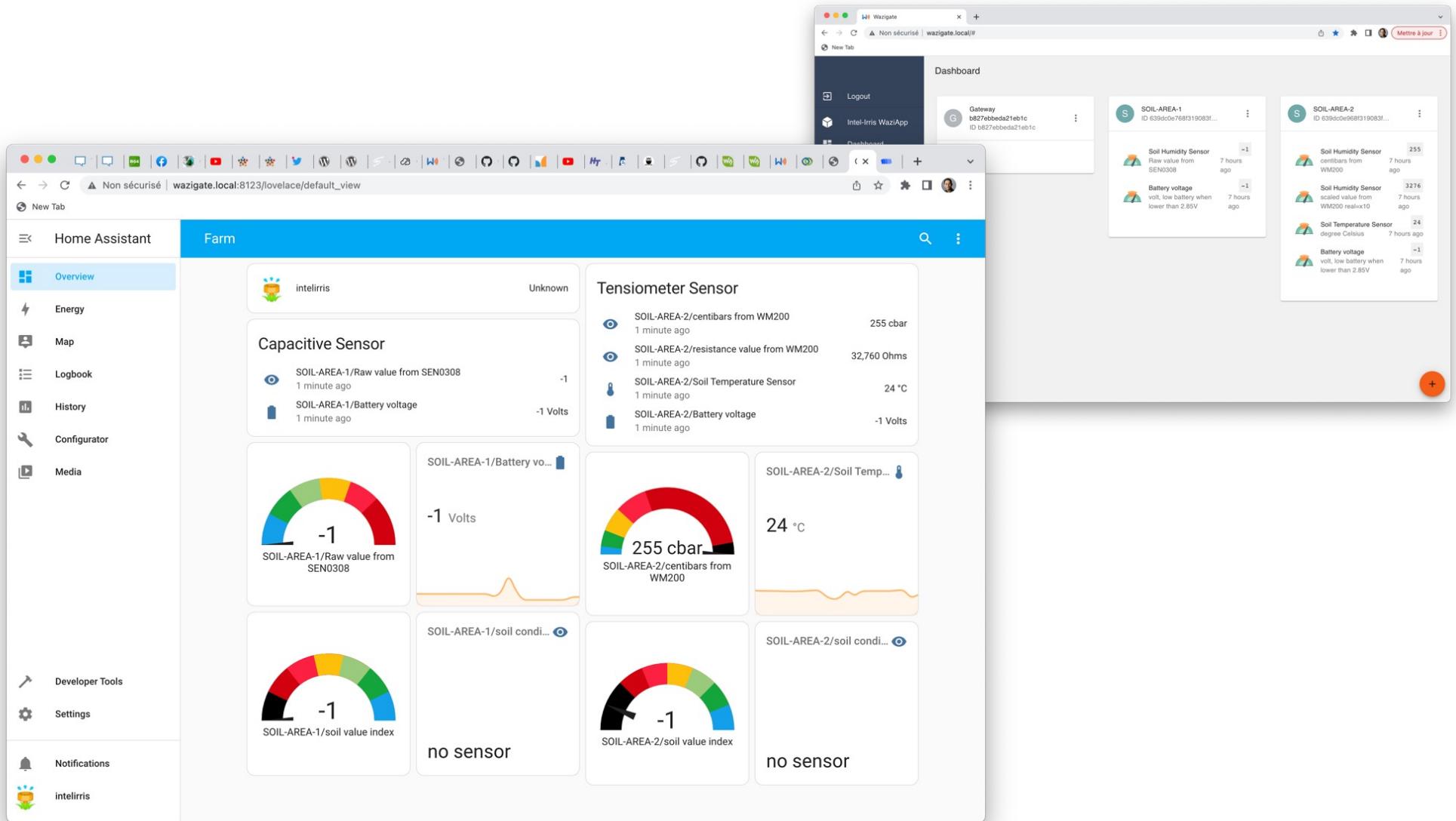
- The OLED displays the latest received sensor data for the end-user: the device name, the time of last received data, the sensor raw value and the soil condition
- The main screen is displayed for 6s every 30s. Then a screen saver display will show a shorter version of these information with a 5-bar visual
- 5 bars: saturated | 4 bars: wet
- 3 bars: wet | 2 bars: dry
- 1 bar: dry | 0 bar: very dry



Home Assistant integration

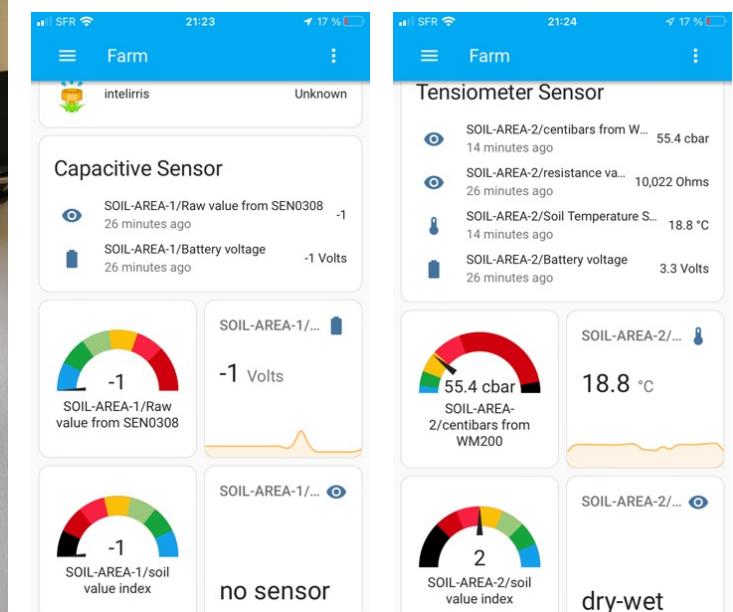
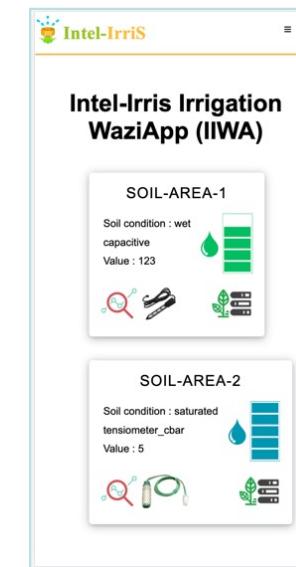
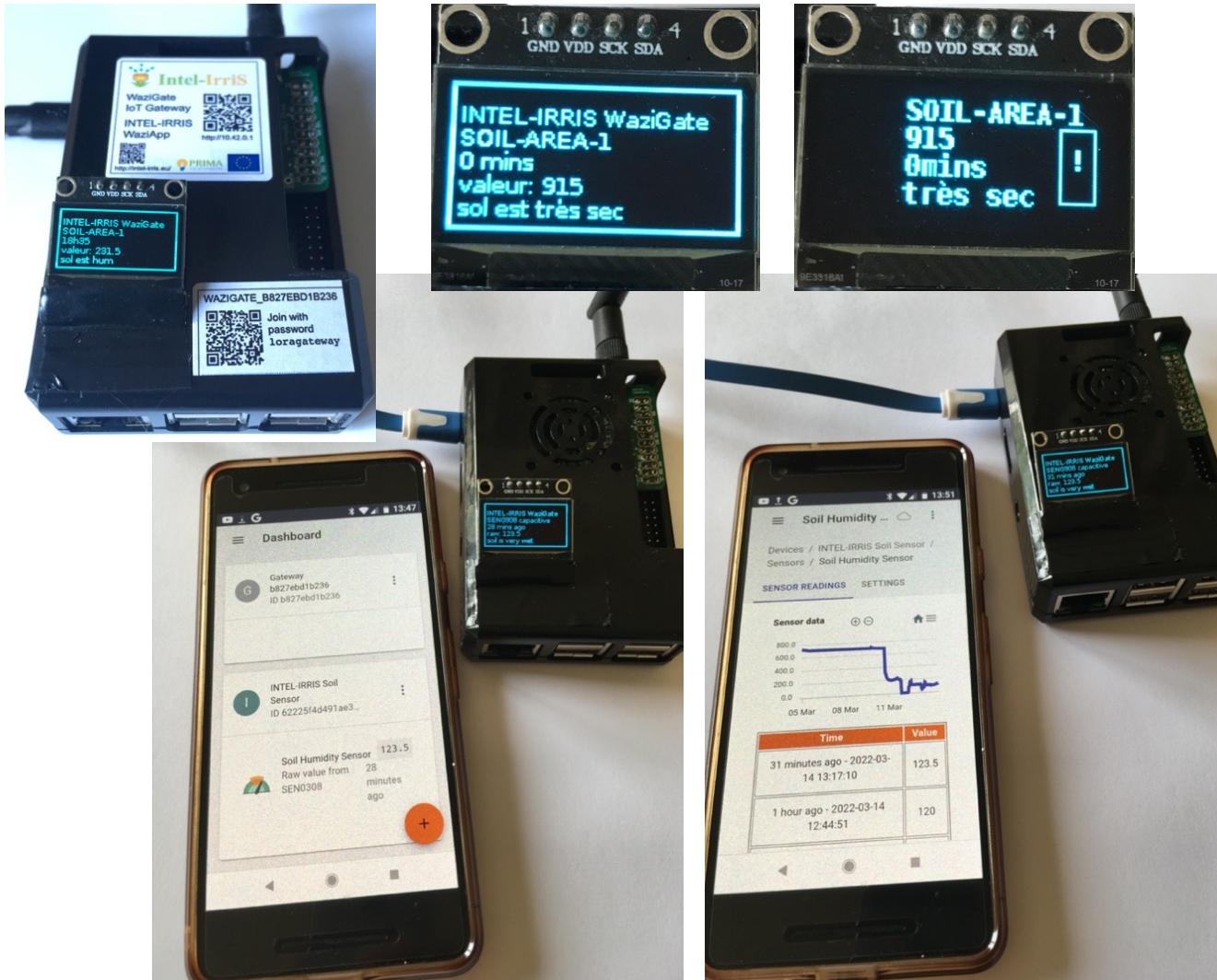
- Download the INTEL-IRRIS gateway **w/HA integration** SD card image from the project website: <http://intel-irris.eu/results>
- Power & boot your RPI
- When connected to the gateway (either with wired Ethernet or through the gateway's WiFi), open a web browser and open
 - `http://wazigate.local:8123` if wired Ethernet
 - `http://10.42.0.1:8123` if through gateway's WiFi
- The HA login page should appear
 - User: inteliris
 - Password: inteliris
- The HA default dashboard shows the default configuration of the starter-kit
- Of course, the gateway default dashboard is still available

Home Assistant dashboard



The screenshot displays two browser windows side-by-side. The left window shows the Home Assistant interface for a "Farm" entity. It includes a sidebar with options like Overview, Energy, Map, Logbook, History, Configurator, Media, Developer Tools, Settings, Notifications, and intelirris. The main area shows a "Tensiometer Sensor" card with four data points from SOIL-AREA-2 and SOIL-AREA-1. Below it are four cards: SOIL-AREA-1/Battery voltage (-1 Volts), SOIL-AREA-2/centibars from WM200 (255 cbar), SOIL-AREA-1/soil cond... (no sensor), and SOIL-AREA-2/soil cond... (no sensor). The right window shows the Wazigate gateway status for "intel-Irris WaziApp" with a "Dashboard" tab open, displaying sensor data for SOIL-AREA-1 and SOIL-AREA-2.

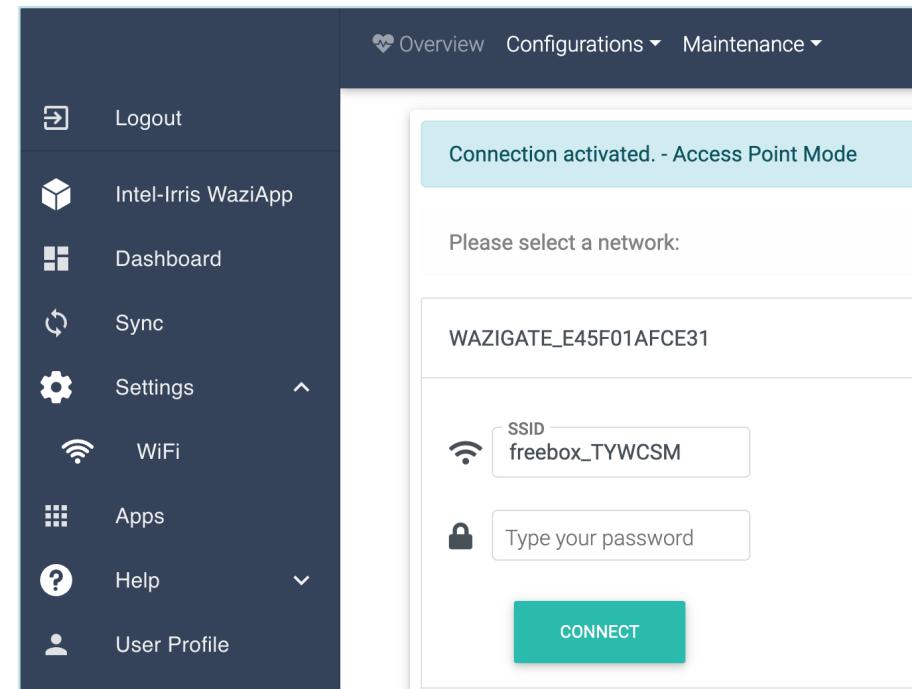
Summary of INTEL-IRRIS gateway various User Interfaces



Advanced configuration

connect gateway to a WiFi network

- By default, the gateway acts as a WiFi Access Point
- To connect the gateway to a WiFi network, go to Setting/WiFi to list all available WiFi networks
- Then select the one you want in order to provide the WiFi password
- Ex: connect to freebox_TYWCSM
- Once connected, gateway is in WiFi Client mode



Advanced configuration

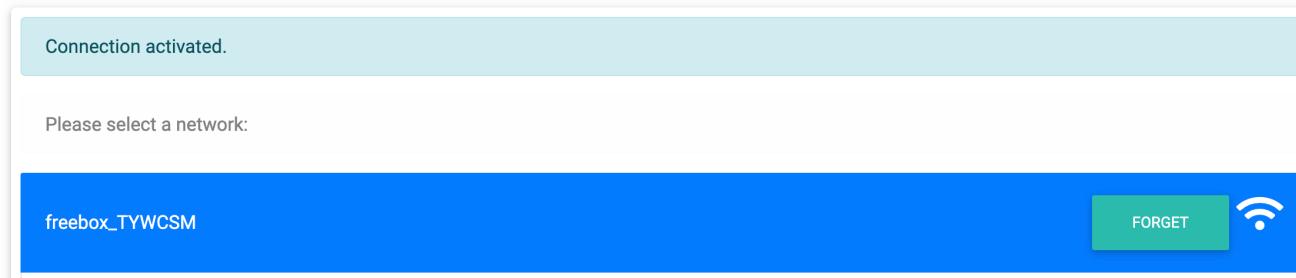
connect gateway to a WiFi network, con't

- You can connect to several WiFi networks, one after another, to have a list of known WiFi networks
- They will be memorized and if the current WiFi network is not available, another available network in the list of known WiFi networks will be selected
- If there are no available WiFi networks in the list of known WiFi networks anymore, then the WaziGate switches back to Access Point mode

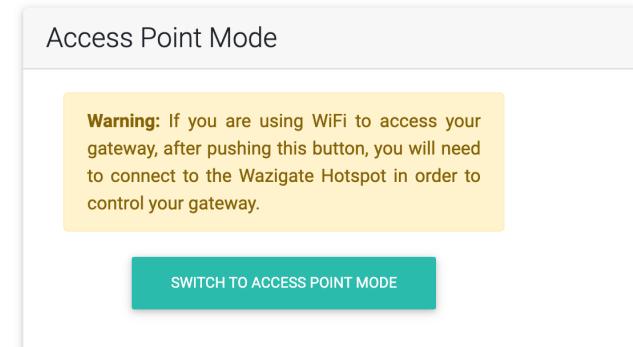
Advanced configuration

switch gateway back to WiFi access point mode

- To get back to Access Point mode, go to Setting/WiFi and simply click on "Forget" for the current WiFi network



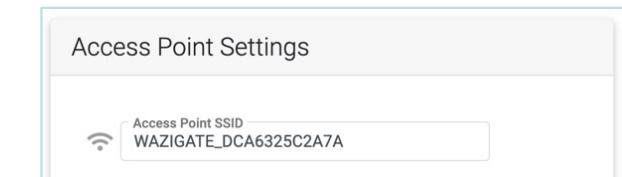
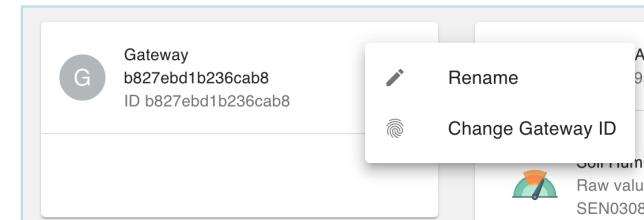
- If you previously added several WiFi networks, click on "Forget" for **ALL** known & memorized WiFi networks
- **DO NOT USE** the "SWITCH TO ACCESS POINT MODE" option
- **IT IS NOT STABLE!**



Advanced configuration

Synching your gateway to the cloud

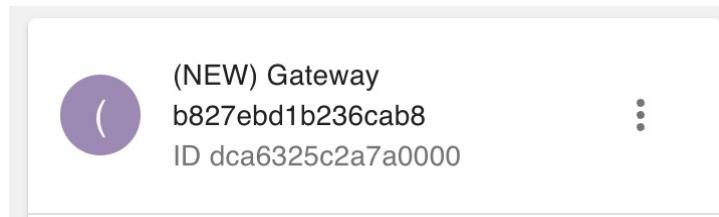
- If you want to sync your gateway to the Waziup Cloud, look at this tutorial page
 - <https://www.waziup.io/documentation/wazigate/v2/install/#registration-with-the-cloud>
- You will need an account on Waziup Cloud dashboard
 - If you don't have one, you need to create one first
 - <https://dashboard.waziup.io/>
- Then, you NEED to change your gateway id
 - Use the unique MAC address of your gateway that appears in Settings/Configuration (it is used as your gateway's WiFi hotspot)
 - Here: DCA6325C2A7A
 - Add 0000 at the end to have 16 digits
 - -> DCA6325C2A7A0000



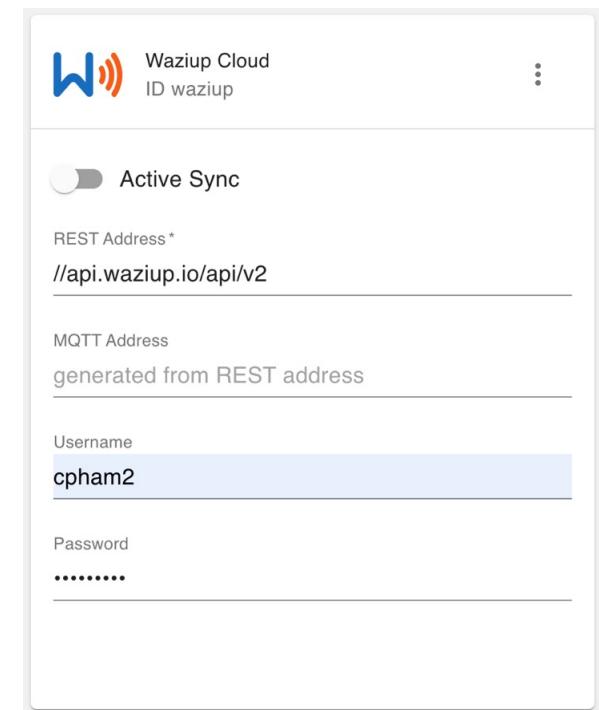
Advanced configuration

Synching your gateway to the cloud, con't

- You should have a new gateway on your dashboard with the new ID



- Enter your Waziup Cloud credentials in the Sync menu
- Then, just activate sync on your gateway which needs to be connected to Internet
- Log in the Waziup Cloud dashboard and check that you see your gateway and your device
- You can activate/deactivate synchronization at anytime



Advanced configuration

MQTT integration from WaziCloud

- With sensor data on WaziCloud, it is possible to subscribe to those data with MQTT protocol
- With command line mosquitto_sub
 - mosquitto_sub
-L "mqtt://api.waziup.io/devices/<deviceID>/sensors/<sensorID>/value"
 - mosquitto_sub
-h api.waziup.io -t devices/<deviceID>/sensors/<sensorID>/value
- With other MQTT integration client/platform
 - Host: api.waziup.io
 - Topic: devices/<deviceID>/sensors/<sensorID>/value
- Output
 - { "value": 34, "timestamp": "2022-12-23T10:23:54Z" }

Advanced configuration

MQTT integration – MQTT client on a smartphone

- Example with an MQTT client (EasyMQTT) on an iPhone7

