IoT Deployment with WAZIUP

* * *

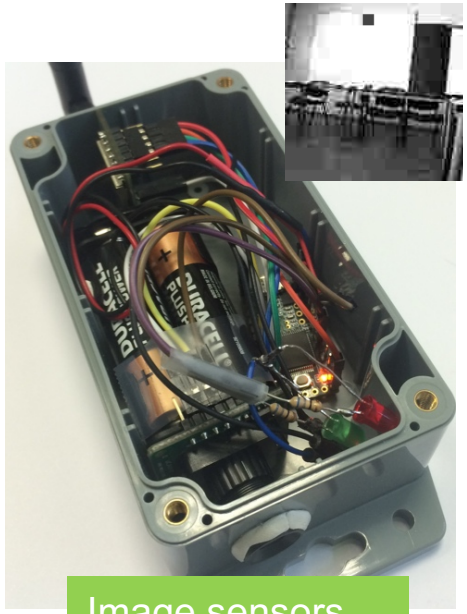
Guidelines and Best practices

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

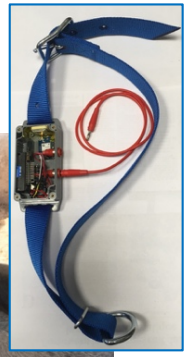
Reading instructions

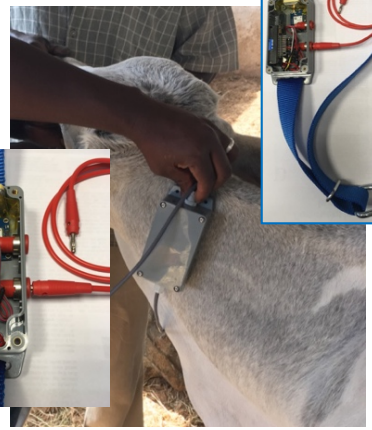
- Recommended reading:
  - Low-cost-LoRa-IoT-step-by-step.pdf
  - Low-cost-LoRa-IoT-outdoor-step-by-step.pdf
  - Low-cost-LoRa-GW-step-by-step.pdf
  - WAZIUP FAQ
  - https://github.com/CongducPham/tutorials
- This document specifically focuses on deployment issues while the above mentioned documents provide more general and broader information on the WAZIUP long-range IoT platform.

Sensing Devices



GPS collar





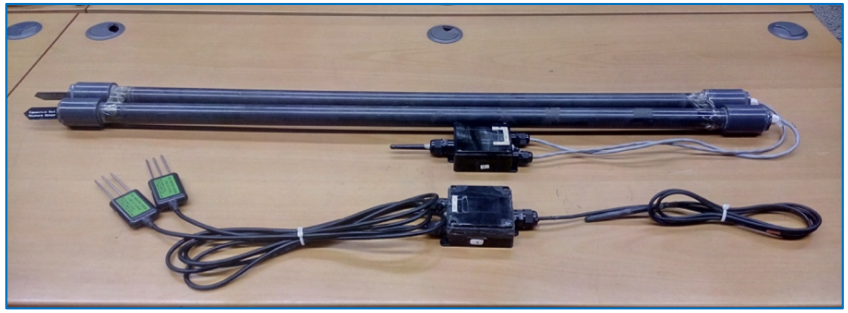


Image sensors

Soil Moisture

Photo from Unparallel

Bin presented at Woelab

Waste Mngt

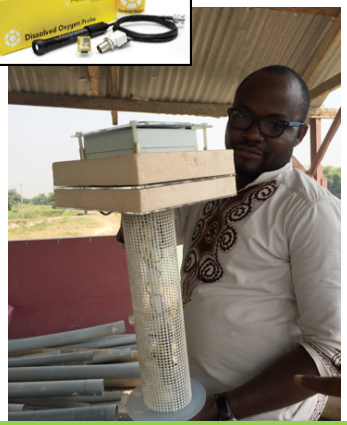


Photo from EGM

Buoy for water quality

Weather Station

Sensing device Important issues

- Don't forget that you should never transmit without an antenna!
- When a device has been flashed and each time you switch it on, it is going to transmit, so don't forget the antenna in any case!
- Put a name tag on the case to remember the device's address, see next slide.

Sensing device
Arduino_LoRa_Simple_temp

- For each sensor node that you will install you have to change the device's address, starting at 2 for instance. Address 1 is reserved for the gateway.

```
//////////////////////////////////////////////////////////////////
// CHANGE HERE THE LORA MODE, NODE ADDRESS
#define LORAMODE  1
// you need to change the node address for each sensor in the same organization/farm
// node address starts at 2 and ends at 255
#define node_addr 2
//////////////////////////////////////////////////////////////////
```

- If needed, change the frequency for measure and transmission, in minutes.

```
//////////////////////////////////////////////////////////////////
// CHANGE HERE THE TIME IN MINUTES BETWEEN 2 READING & TRANSMISSION
unsigned int idlePeriodInMin = 60;
//////////////////////////////////////////////////////////////////
```

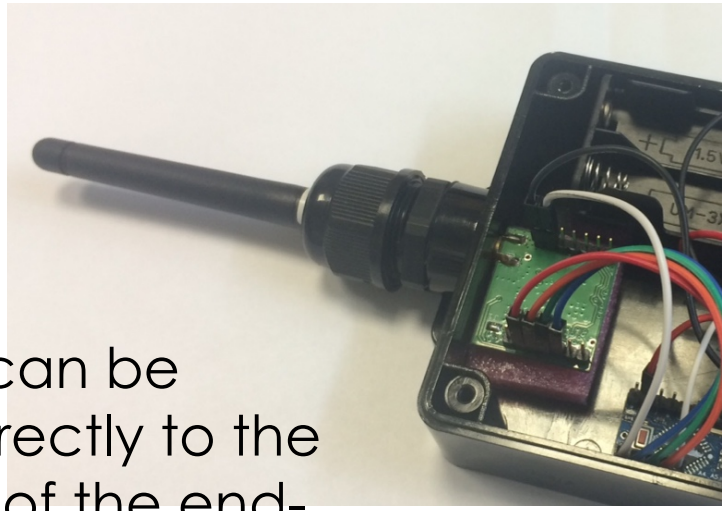
Sensing device Deployment

If you case has hole, do not put these holes in the up position to limit water getting into the case! Always use cable gland to also void dust

- Strongly tighten all cable glands, especially the one of the antenna cable so that the cable cannot turn and get disconnected from the radio module!
- Remember to put a shade cover to protect from direct sun!

CONNECTING AN ANTENNA TO THE SENSING DEVICE

- Many low-cost antenna that you can buy are usually simple ¼ wave whip/monopole antenna with connector (usually SMA-male)



- The antenna can be connected directly to the radio module of the end-device, using a larger cable gland to connect the antenna through the cable gland.

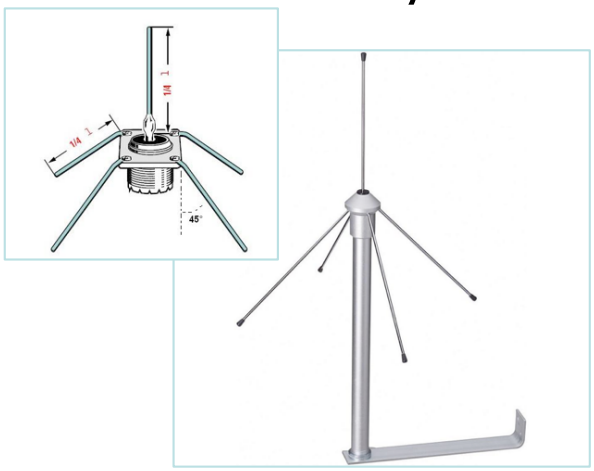
Use a coaxial antenna cable

- However, when the antenna is connected directly to the radio module, placing the device may be difficult as the antenna should be placed at a high location such as on top of a mast.

- Using an extension coaxial cable between the antenna and the radio module greatly ease the deployment of device **but**:
  - The antenna cable should not be too long to avoid high attenuation: 2m-3m
  - A ¼ wave monopole antenna will not provide good performance

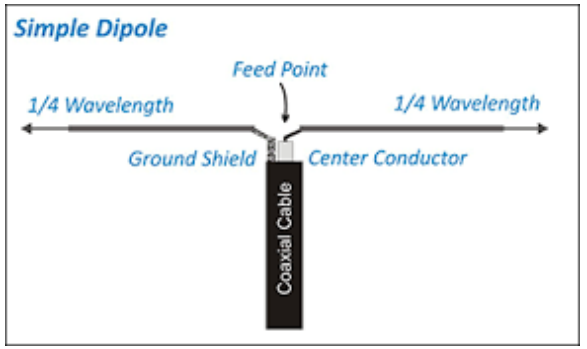
ANTENNA FOR DEVICE WITH A COAXIAL CABLE

- At the end of a coaxial cable, it is possible to connect a ground plane antenna (usually ¼ wave) or a ½ wave dipole antenna.



Ground plane

Sleeve dipole



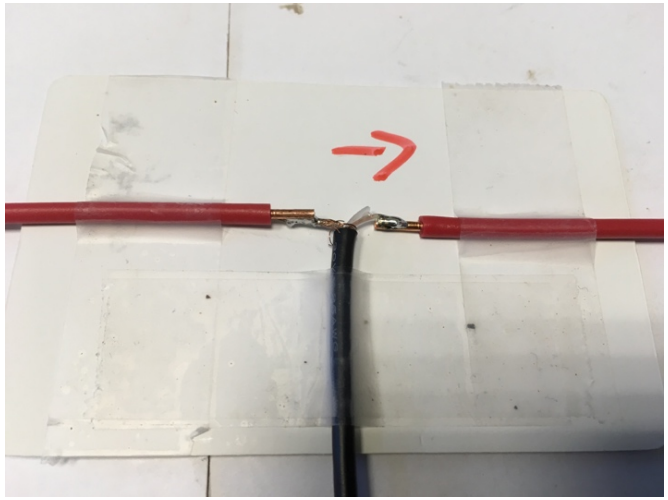
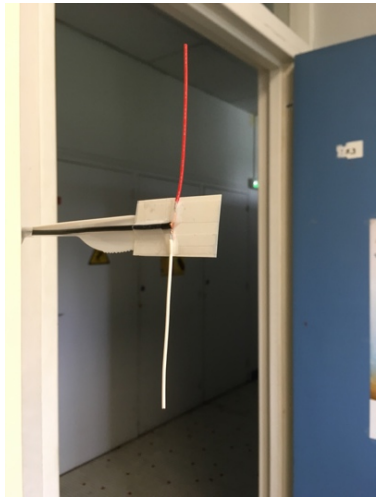
Simple dipole

More complex: collinear, array,…

- Some of them are easy to build (ground plane and simple dipole) and there are many tutorials.

Simple ½ wave dipole antenna

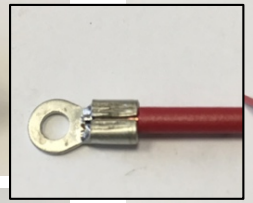
- Very simple dipole can be made with 2 pieces of ¼ wave wires. ¼ wave in 868 is about 8.2cm (16.4cm for 433MHz).

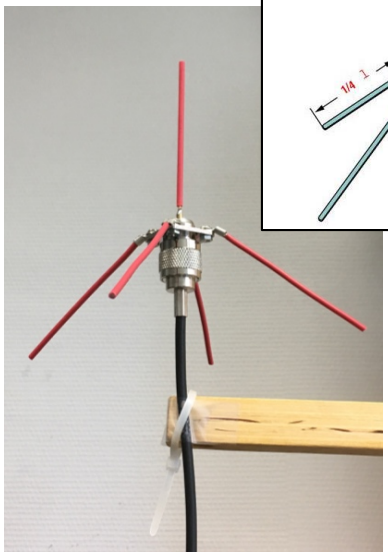



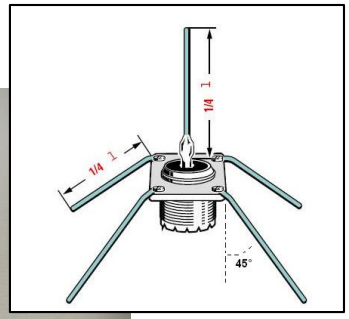

- There is no balun here but it is still better than the ¼ wave monople if a coaxial cable is used.
- You can buy a 3m **RG58** cable (SMA-m to SMA-f for instance), keep the male side, cut the female side and solder the core conductor and the braid as shown.

Simple ¼ wave ground plane antenna

- The ground plane antenna can be made with 5 pieces of ¼ wave wires.





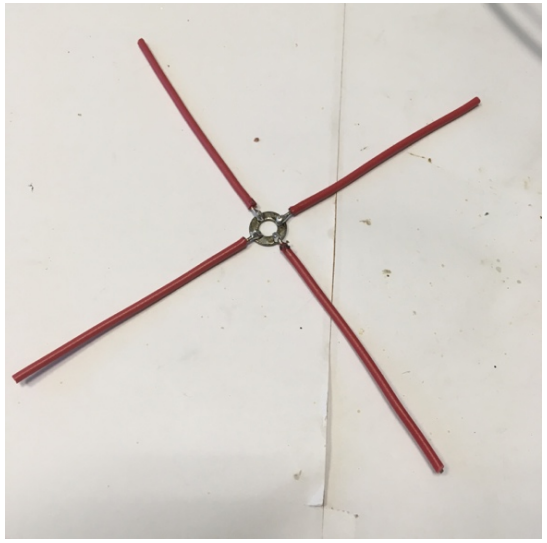
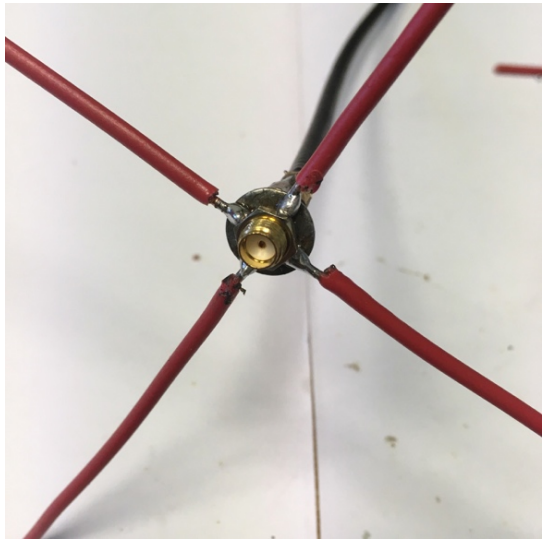
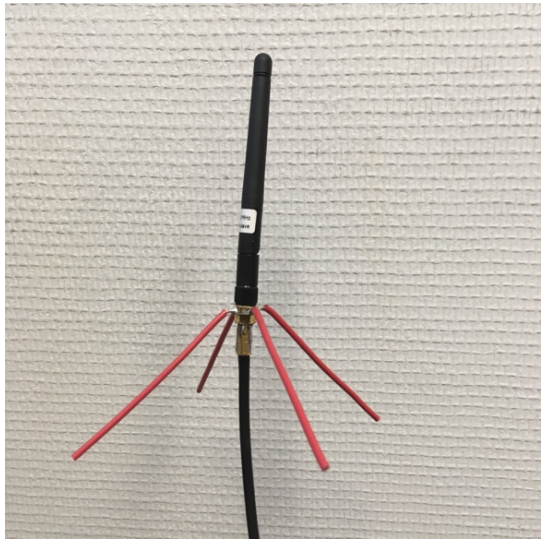


N Female Panel Mount Chassis

- You can buy a 3m RG58 cable with an SMA-male at one end and a male N-connector at the other end. Or build your own cable.

Even simpler ¼ wave ground plane antenna

- With an existing SMA-m/SMA-f cable, you can also build a ground plane antenna by adding 4 radiant wires to the ¼ wave monopole.

- This is a cheaper solution for sensing devices.

Some cable links

https://www.aliexpress.com/item-img/SMA-M-le-SMA-Femelle-Connecteur-Extension-C-ble-RG58-2-M/32543987605.html

2m RG58 SMA male to SMA female

2m RG58 N male to SMA male

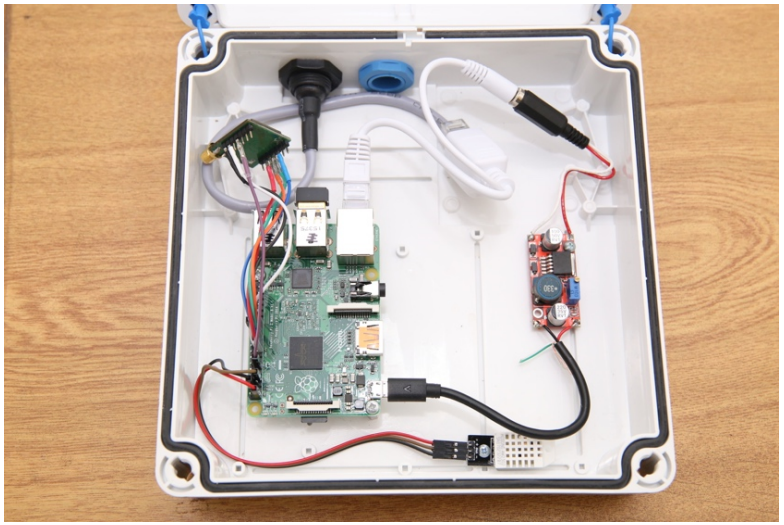
https://www.aliexpress.com/item-img/RG58-2m-N-Male-Jack-to-SMA-Male-M-M-RF-Coax-Pigtail-WLAN-Adapter-Adaptor/32616929641.html#

SUMMARY OF ANTENNAS FOR SENSING DEVICE

- The easiest solution would be to buy a general purpose antenna for the frequency range you are operating.
- If you want to try the DIY approach, try first the simple dipole and then see if the range is acceptable.
- A ground plane antenna can be purchased or also made. You can test both solutions.
- RF transmissions depend a lot on the antenna location, the environment and many other factors!

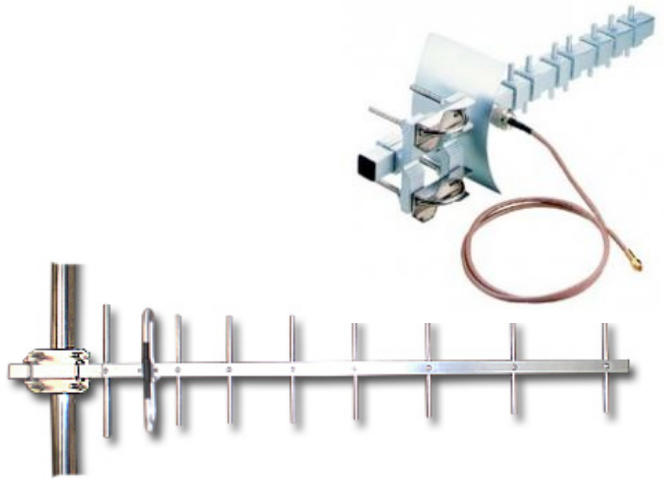
INSTALLING THE GATEWAY

- Power the gateway either with PoE or directly with a 5V USB adaptor.
- If possible, it is much better to put the gateway indoor.
- Try avoiding long antenna cable: 2m to 5m max.
- If the gateway needs to be put outdoor because of the antenna cable constraints, don't forget to protect it from direct sun!
- Get Internet access by connecting the Ethernet cable to a DSL or 4G router that will assign an IP address with DHCP.



ANTENNAS FOR GATEWAY

- Antennas for gateways can be placed on a building, at a high location.
- As for end-devices, you can easily use ground plane or sleeve dipole antenna. More complex high gain antenna or a directional Yagi antenna can be purchased depending on your budget and whether the device deployment allows it.



Gateway software installation

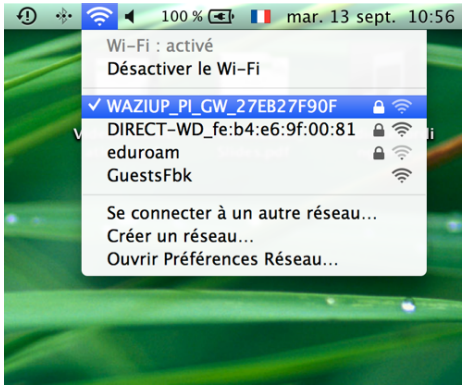
- An SD card image with a Raspberry Raspbian Jessie version is provided.
- You will need an 8GB SD card. Be careful, some SD cards will not work. This one has been successfully tested. It has to be class 10.

- Look at https://www.raspberrypi.org/documentation/installation/installing-images/ to see the procedure depending on your OS. 7948206080 bytes should be written, otherwise you may have a problem.
- Once flashed, insert the SD card and power-up the Raspberry-based gateway.

SSH TO THE GATEWAY WITH WIFI

- The gateway is also configured as a WiFi access point with address `192.168.200.1`
- Select the `WAZIUP_PI_GW_xxxxxxxxxx` WiFi
- WiFi password is `loragateway`
- Then `ssh pi@192.168.200.1`
- Login password is `loragateway`

You can use an iOS or Android smartphone or tablet to connect to the gateway with an SSH client app! See next slide.




```
cpham — pi@raspberrypi: ~/lora_gateway —
MacBookProRetina-de-Congduc-Pham:~ cpham$ ssh pi@192.168.200.1
pi@192.168.200.1's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug  4 17:19:00 2016 from 192.168.200.102
pi@raspberrypi:~ $ cd lora_gateway/
pi@raspberrypi:~/lora_gateway $ ll
total 864
-rw------- 1 pi   pi    44155 Aug  3 16:55 arduPi.cpp
-rw------- 1 pi   pi    16715 Aug  3 16:55 arduPi.h
-rw-r--r-- 1 pi   pi    35164 Aug  3 17:01 arduPi.o
-rw------- 1 pi   pi    43310 Aug  3 16:55 arduPi_pi2.cpp
-rw------- 1 pi   pi    14043 Aug  3 16:55 arduPi_pi2.h
-rw------- 1 pi   pi    77976 Aug  3 16:55 bcm2835.h
```

Gateway access & configuration interfaces

- There are 2 interfaces
  - A web admin interface
  - A command line interface that needs ssh
- Look at the gateway tutorial
  - https://github.com/CongducPham/tutorials/blob/master/Low-cost-LoRa-GW-step-by-step.pdf
- The web interface is sufficient for most users
  - Easy basic configuration and easy update
  - Pre-defined cloud configuration
  - dedicated tutorial: https://github.com/CongducPham/tutorials/blob/master/Low-cost-LoRa-GW-web-admin.pdf

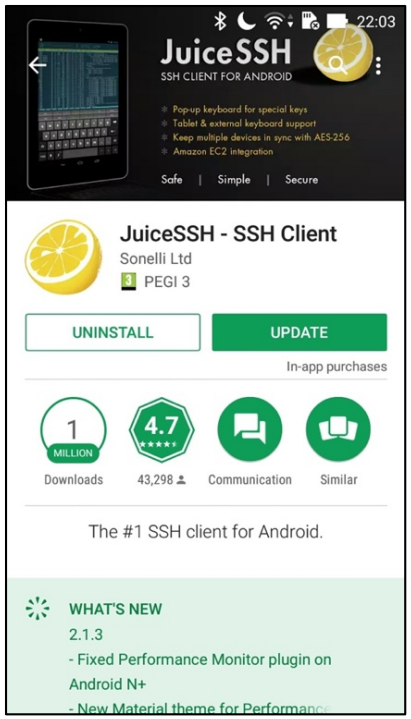
GATEWAY'S SIMPLE COMMAND INTERFACE

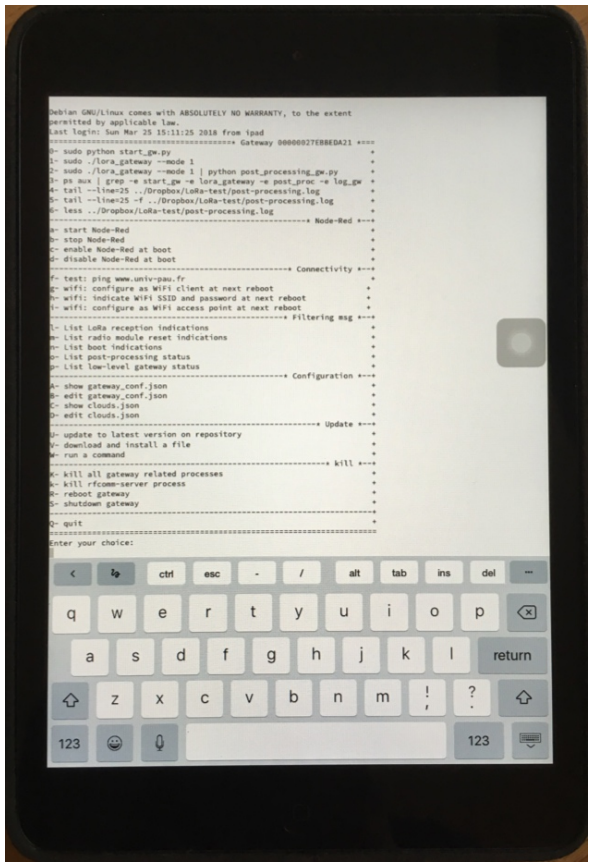
- Once logged on the gateway, you may directly enter in a simple command interface
- This command interface consists in a `cmd.sh` shell script
- **In image versions after May 2017, this script is launched when you log into the gateway with ssh**
- If this happens, select Q and hit RETURN to quit this interface
- You should be in the `lora_gateway` folder

```
pi@raspberrypi:~/lora_gateway $ ./cmd.sh
=====================================* Gateway 00000027EB84C456 *===
0- sudo python start_gw.py                                          +
1- sudo ./lora_gateway --mode 1                                     +
2- sudo ./lora_gateway --mode 1 | python post_processing_gw.py      +
3- ps aux | grep -e start_gw -e lora_gateway -e post_proc -e log_gw +
4- tail --line=25 ../Dropbox/LoRa-test/post-processing.log          +
5- tail --line=25 -f ../Dropbox/LoRa-test/post-processing.log       +
6- less ../Dropbox/LoRa-test/post-processing.log                    +
-----------------------------------------------* Connectivity *--+
f- test: ping www.univ-pau.fr                                       +
g- wifi: configure as WiFi client at next reboot                    +
h- wifi: indicate WiFi SSID and password at next reboot             +
i- wifi: configure as WiFi access point at next reboot              +
-----------------------------------------------* Filtering msg *--+
l- List LoRa reception indications                                  +
m- List radio module reset indications                              +
n- List boot indications                                            +
o- List post-processing status                                      +
p- List low-level gateway status                                    +
-----------------------------------------------* Configuration *--+
A- show gateway_conf.json                                           +
B- edit gateway_conf.json                                           +
C- show clouds.json                                                 +
D- edit clouds.json                                                 +
-------------------------------------------------------* ngrok *--+
M- get and install ngrok                                            +
N- ngrok authtoken                                                  +
O- ngrok tcp 22                                                     +
-------------------------------------------------------* Update *--+
U- update to latest version on repository                           +
V- download and install a file                                      +
W- run a command                                                    +
-------------------------------------------------------* kill *--+
K- kill all gateway related processes                               +
k- kill rfcomm-server process                                       +
R- reboot gateway                                                   +
S- shutdown gateway                                                 +
-------------------------------------------------------------------+
Q- quit                                                             +
=====================================================================
Enter your choice:
```

Using iOS/Android smartphone or tablet

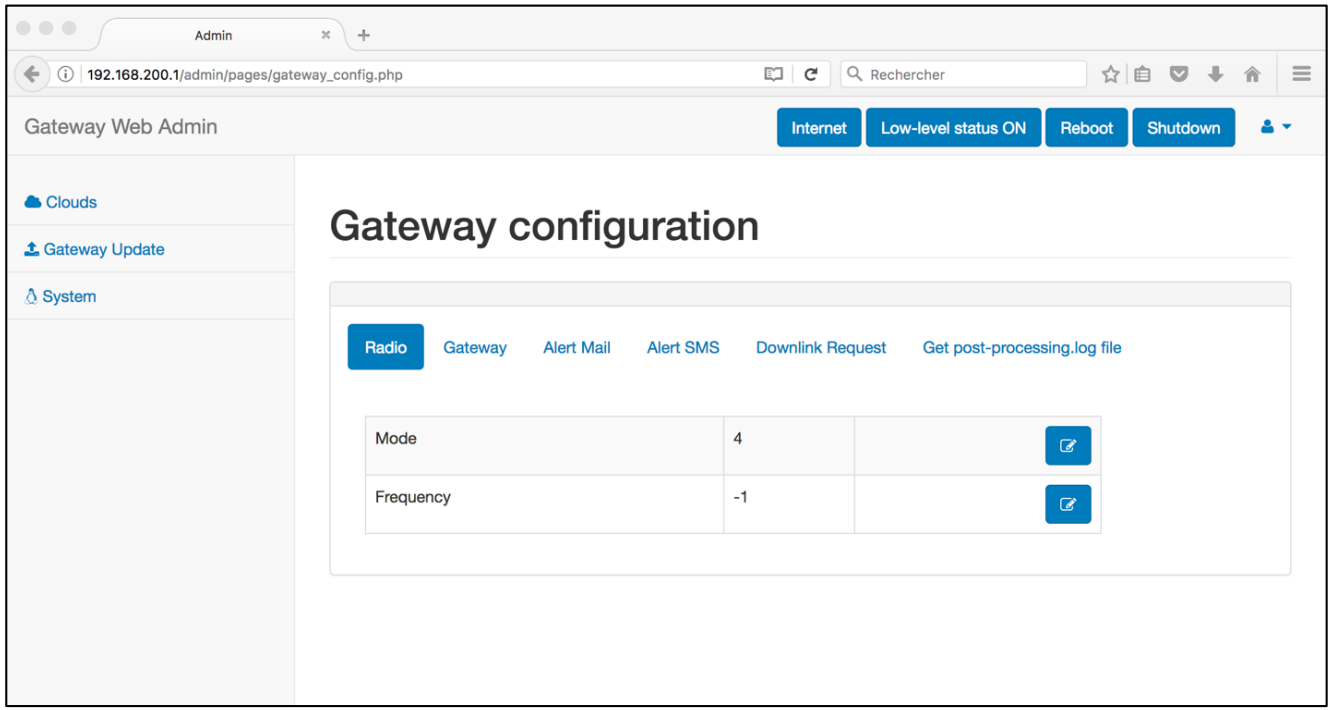
- On iOS we tested Termius
- On Android we tested JuiceSSH





Gateway web admin interface

- http://192.168.200.1/admin
  - Login: admin
  - Password: loragateway

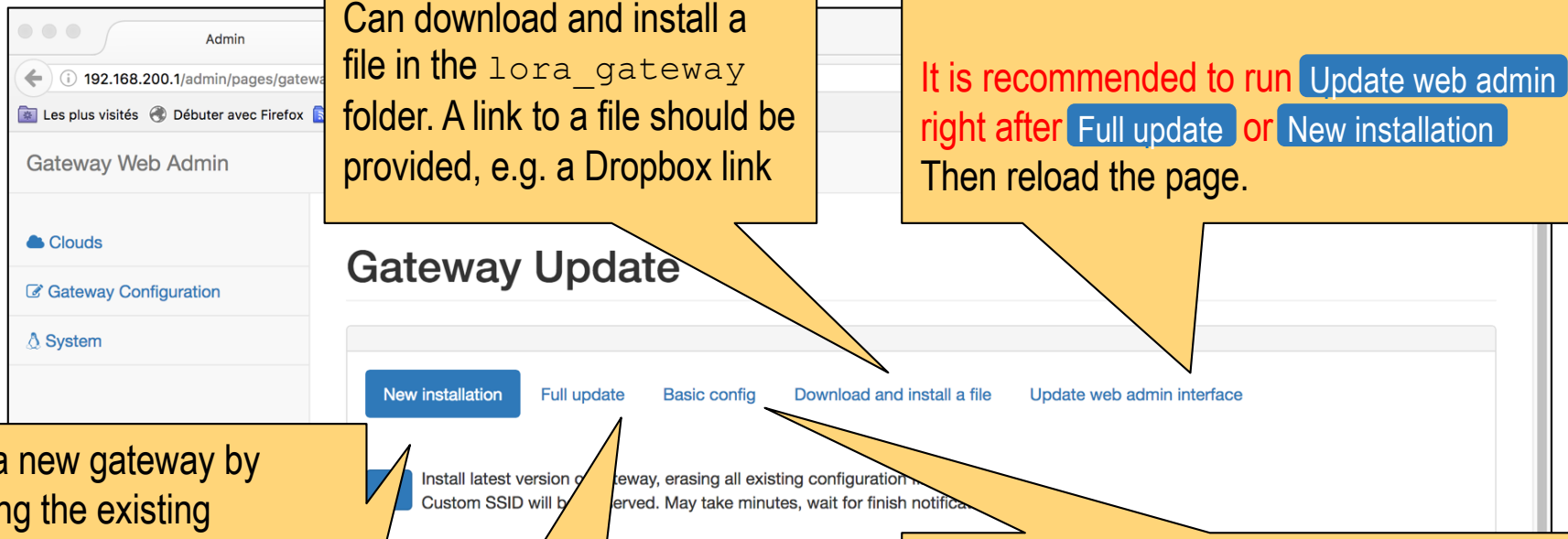


Gateway update

- The gateway must be updated to the latest version.
- It is better to do the update procedure at the university office so that the gateway can have easy internet access.
- The update procedure can easily be done with the web admin interface, connect to the gateway WiFi first
- The update steps are
  1. Full Update
  2. Basic Config
  3. Update Web Interface

GATEWAY UPDATE PAGE

❑ Gateway update section



Can download and install a file in the `lora_gateway` folder. A link to a file should be provided, e.g. a Dropbox link

Update the web admin interface after an update of the distribution to install the last version of the web admin interface.

It is recommended to run Update web admin right after Full update or New installation
Then reload the page.

Install a new gateway by removing the existing `lora_gateway` folder, all existing configuration files will be overwritten.

If you install a new gateway with our SD card image, you can use this option.

Update with latest version on github, all your configuration files will be kept. This is the recommended option.

Compile and configure the gateway (to set the gateway id & the WiFi access point SSID). This is also required if you install a new gateway using the provided SD card image. It is recommended to run Basic config right after Full update or New installation

Configuring data management

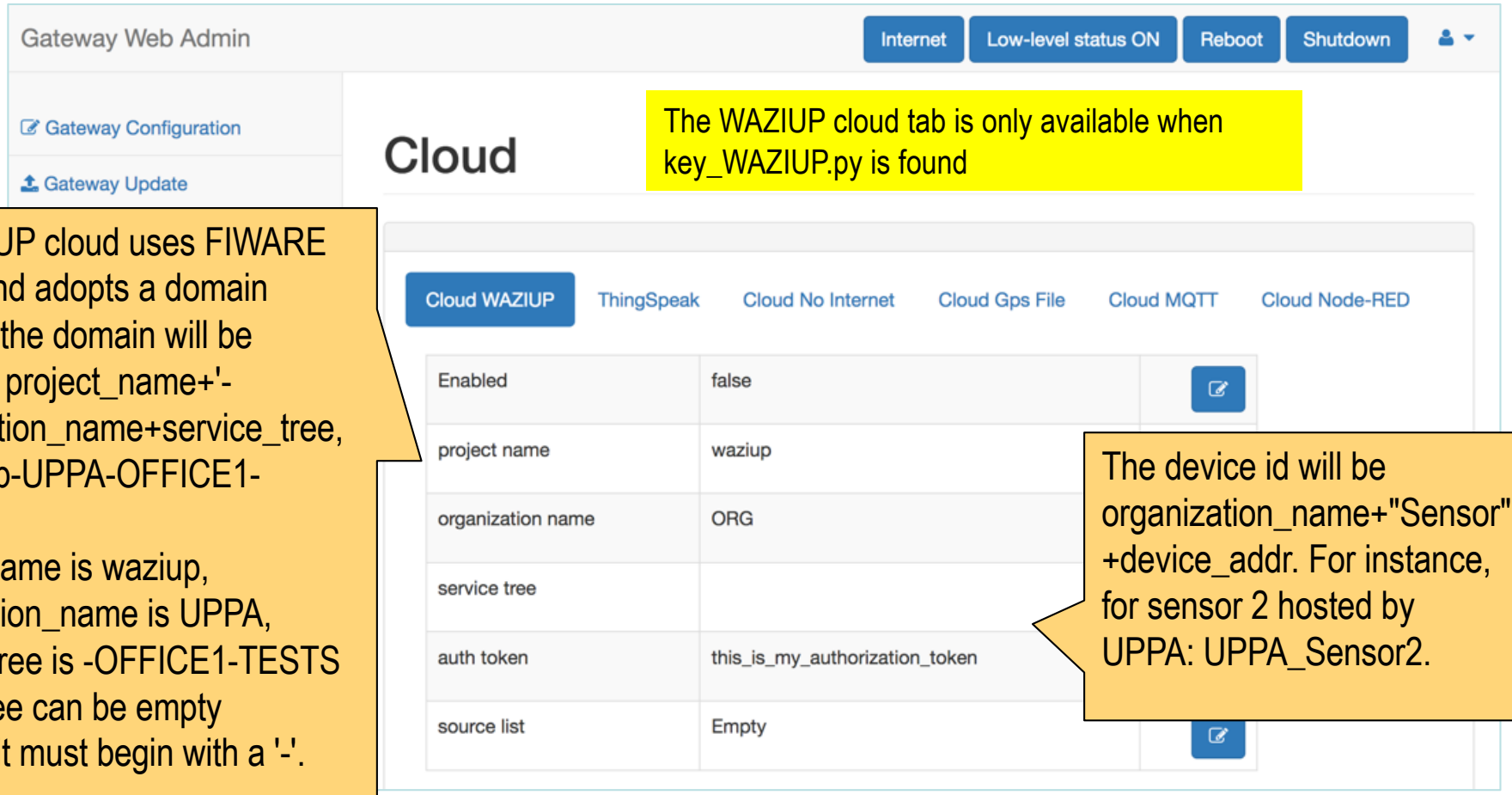
- Received data from devices will be uploaded to the WAZIUP data platform.

```
{
  "name": "WAZIUP Orion cloud new API",
  "script": "python CloudWAZIUP.py",
  "type": "iotcloud",
  "enabled": true
},
```

- Modify `clouds.json` according to your need if necessary
- `CloudWAZIUP.py` script will use information from `key_WAZIUP.py` to configure data management for each organization

WAZIUP Cloud Configuration

- Configuring WAZIUP cloud



KEY_WAZIUP.PY

```
#####################################################
#server: CAUTION must exist
orion_server="http://api.waziup.io/api/v1"

#project name
project_name="waziup"

#your organization: CHANGE HERE
#choose one of the following: "DEF", "UPPA", "EGM", "IT21", "CREATENET", "CTIC", "UI", "ISPACE",
"UGB", "WOELAB", "FARMERLINE", "C4A", "PUBD"
organization_name="ORG"

#service tree: CHANGE HERE at your convenience
#should start with -
#service_tree='-TESTS'
#can be empty
service_tree=''

#sensor name: CHANGE HERE but maybe better to leave it as Sensor
#the final name will contain the sensor address
sensor_name="Sensor"

#service path: DO NOT CHANGE HERE
service_path='-'+organization_name+service_tree

#SUMMARY
#with a domain vision, the domain will be project_name+service_path, e.g waziup-UPPA-TESTS
#the entity name will then be organization_name+"_"+sensor_name+scr_addr, e.g. "UPPA_Sensor2"

source_list=[]
```

You need to change the organization_name.

service_tree is optional

EDITING KEY_WAZIUP.PY

Use `nano` to edit the file:

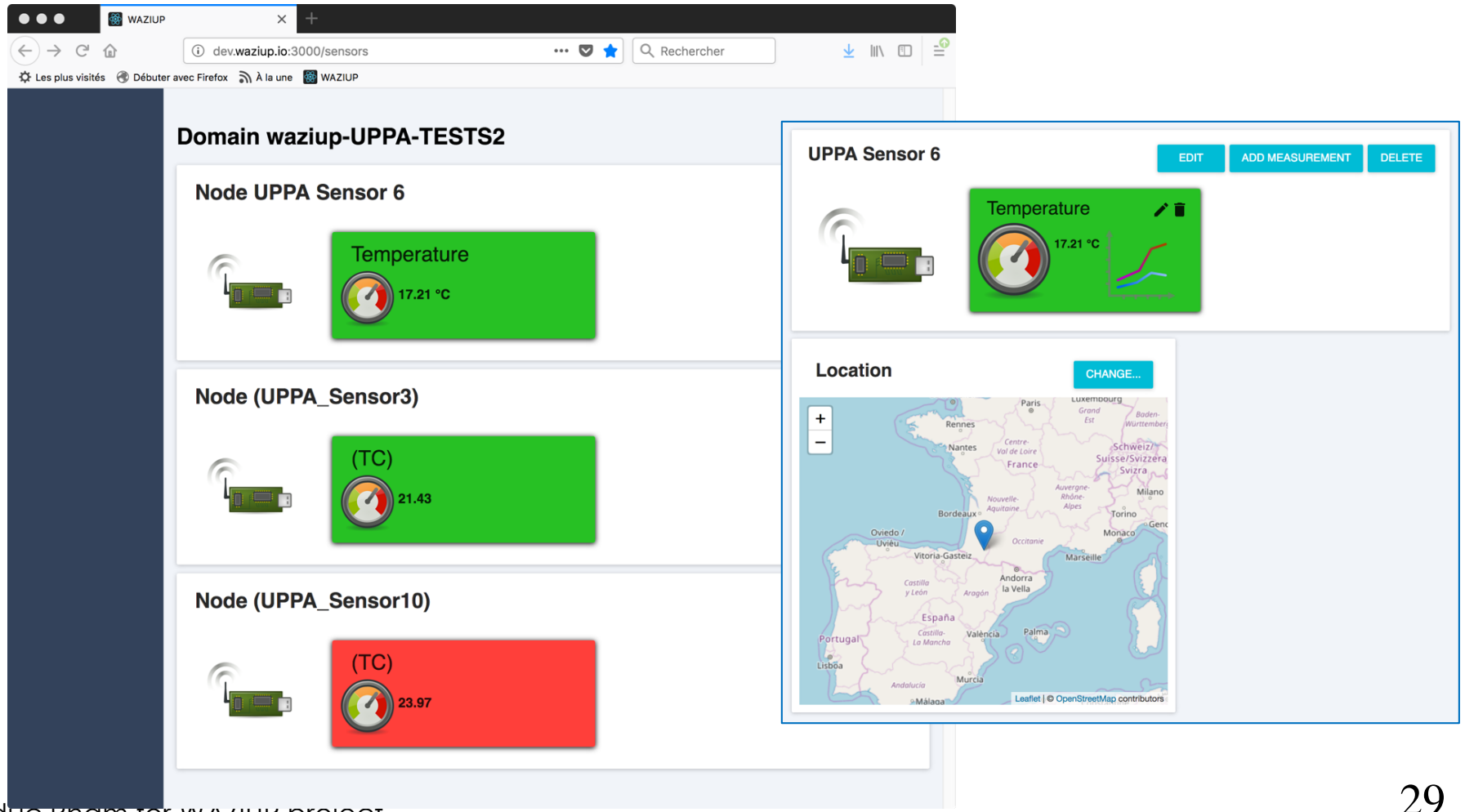
```
> nano key_WAZIUP.py
```

Then CTRL-O + RETURN to save

CTRL-X to quit

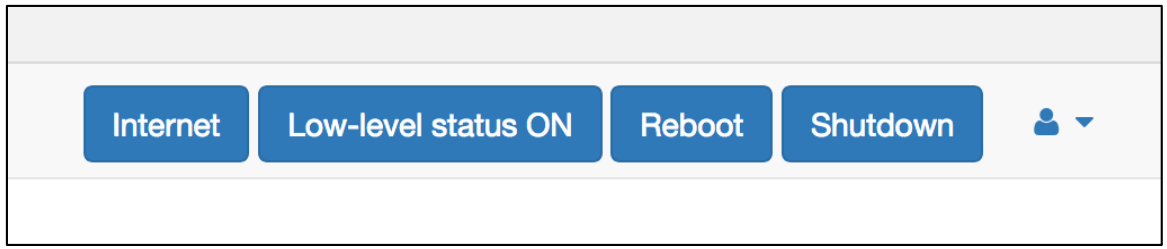
The WAZIUP cloud platform

- dashboard.waziup.io



REBOOTING THE GATEWAY

- Your gateway is now updated and configured
- You can now reboot the gateway



- After reboot, check the WiFi SSID which now should meet your gateway's id
- In general, try to avoid unplugging power cable to shutdown your gateway. Use the web admin interface instead
- Your gateway is now ready to be deployed.

Testing the connectivity between devices and gateway

- When deploying the gateway and the devices, the first step is to check connectivity and adjust the gateway/antenna location.
- It is recommended to use an device programmed to send a message every 1 minute for instance (the Ping-Pong example with a small OLED LCD for instance).
- Place the device at the planned location in the field, with the mast, as for a definitive setting.

Securing with Application Key (1)

- End-device can use application key (app key) on 4 bytes to allow filtering mechanisms at the gateway side.
- The app key is defined in the end-device sketch (`Arduino_LoRa_Simple_temp`) and the feature is activated by uncommenting `#define WITH_APPKEY`

```
#ifdef WITH_APPKEY
////////////////////////////////////////////////////////////////////
// CHANGE HERE THE APPKEY, BUT IF GW CHECKS FOR APPKEY, MUST BE
// IN THE APPKEY LIST MAINTAINED BY GW.
uint8_t my_appKey[4]={5, 6, 7, 8};
////////////////////////////////////////////////////////////////////
#endif
```

- At the gateway side, post_processing_gw.py has a list of allowed app key

```
app_key_list = [
          #change/add here your application keys
          '\x01\x02\x03\x04',
          '\x05\x06\x07\x08' ]
```

Securing with Application Key (2)

- With app key enforcement at gateway, all LoRa data to be uploaded on clouds will need a valid app key, otherwise the data will be discarded as shown below:

```
--- rxlora. dst=1 type=0x12 src=6 seq=136 len=17 SNR=9 RSSIpkt=-56
rcv ctrl pkt info (^p): 1, 18, 6,136,17,9,-56
splitted in:  [1, 18, 6, 136, 17, 9, -56]
(dst=1 type=0x12 src=6 seq=136 len=17 SNR=9 RSSI=-56)
got first framing byte
--> got app key sequence
app key is: [9, 10, 11, 12]
not in app key list
invalid app key: discard data
```

- This is configured in the `gateway_conf.json` file. Set to true

```
    "freq": 433.3
  },
  "gateway_conf": {
    "gateway_ID": "000000XXXXXXXXXX",
    "ref_latitude": "my_lat",
    "ref_longitude": "my_long",
    "wappkey": false,
    "raw": false,
    "aes": false,
    "log_post_processing": true,
```

How to use app key

- App key can be used to differenciate data from one organization to another
  - Sensing devices of a given organization will use the same app key
  - The gateway is configured to only accept this app key
- App key can be used to distribute the gateway task in case several gateways in the same organization are deployed
  - Sensing devices will be caterorized with 2 app key
  - Each gateway will allow only one of these 2 app key
  - In this way, data that can be received by 2 gateways will be processed by only 1 gateway

Securing by encryption (1)

- `Arduino_LoRa_temp` is an extended version of `Arduino_LoRa_Simple_temp` with data encryption feature.
- Data will be encrypted using 128-bit AES algorithm following the LoRaWAN encryption method.
- Uncomment `#define WITH_AES`

```
//////////////////////////////////////////////////////////////////
// COMMENT OR UNCOMMENT TO CHANGE FEATURES.
// ONLY IF YOU KNOW WHAT YOU ARE DOING!!! OTHERWISE LEAVE AS IT IS
#if not defined _VARIANT_ARDUINO_DUE_X_ && not defined __SAMD21G18A__
#define WITH_EEPROM
#endif
#define WITH_APPKEY
#define LOW_POWER
#define LOW_POWER_HIBERNATE
#define WITH_AES
```

Securing by encryption (2)

- Encryption ensures confidentiality. The two 16-byte encryption keys are defined in the end-device sketch (`Arduino_LoRa_temp`)

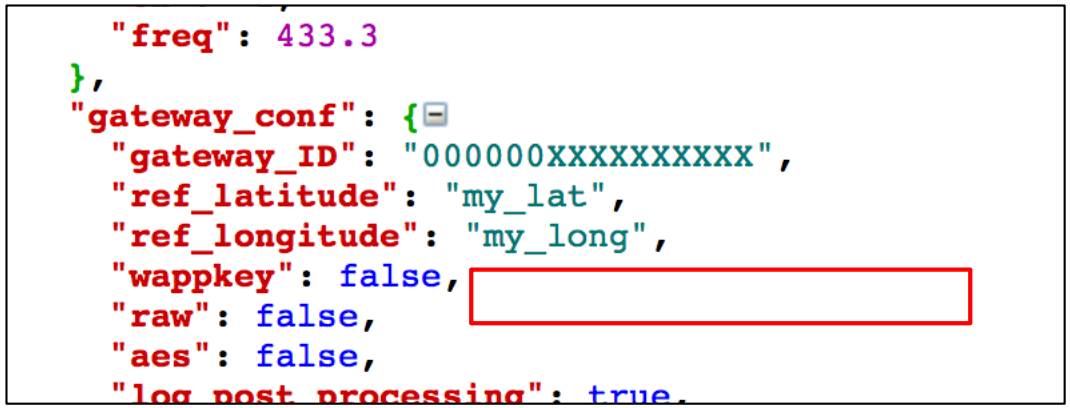
```
unsigned char AppSkey[16] = {
  0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,
  0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F, 0x3C
};

unsigned char NwkSkey[16] = {
  0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,
  0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F, 0x3C
};
```

- And should also be declared in the `loraWAN.py` script on the gateway

```
AppSKey = '2B7E151628AED2A6ABF7158809CF4F3C'
NwkSKey = '2B7E151628AED2A6ABF7158809CF4F3C'
```

Securing by encryption (3)

- With encryption at device and decryption at gateway, there is more robust integrity check of the messages.
- Note that app key can still be used with AES, even if different gateways may have different encryption keys.
- To enable decryption at gateway, AES feature should be activated (set to true) in the `gateway_conf.json` file. You can also use the web admin interface



- Otherwise, the gateway will not be able to decrypt and therefore will not be able to push meaningfull data to clouds

Further readings

- A web page explaining our low-cost gateway
  - http://cpham.perso.univ-pau.fr/LORA/RPIgateway.html
- Specific README files on the github, especially those on cloud management and encryption
  - https://github.com/CongducPham/WaterSense/blob/master/WaterSenseGateway/README-NewCloud.md
  - https://github.com/CongducPham/WaterSense/blob/master/WaterSenseGateway/README-aes_lorawan.md