IOT ONLINE COURSE

Developing low-cost & open-source IoT solutions

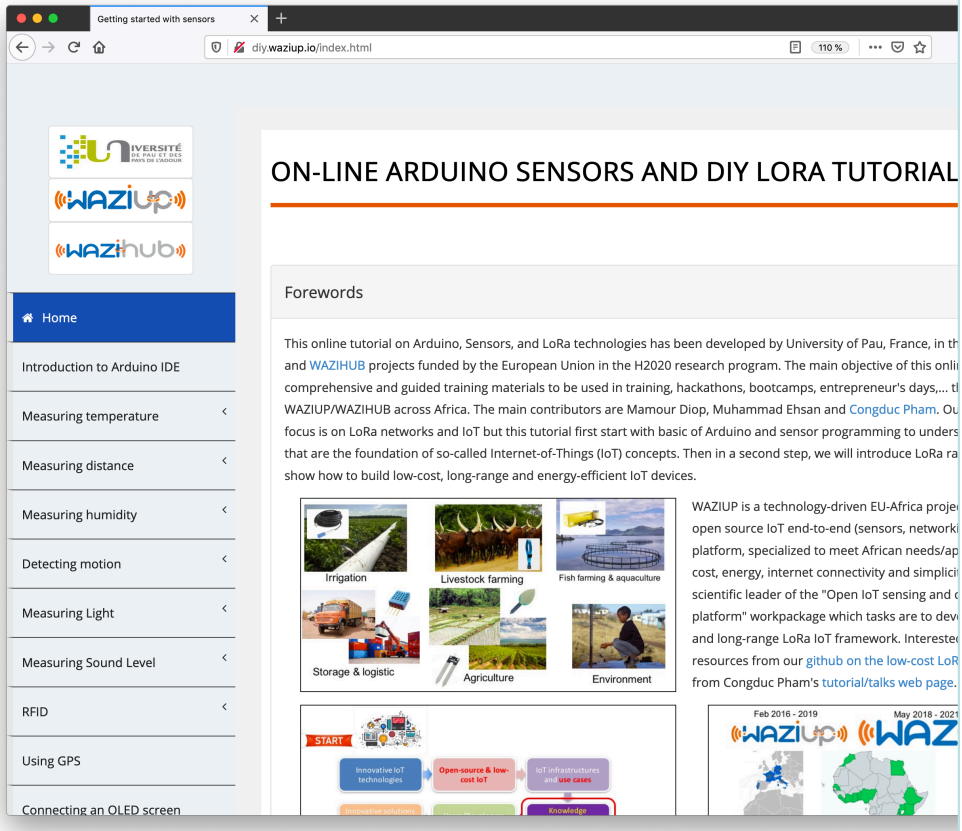
D-GW-5: Device/Gateway Deployment & Configuration Guidelines

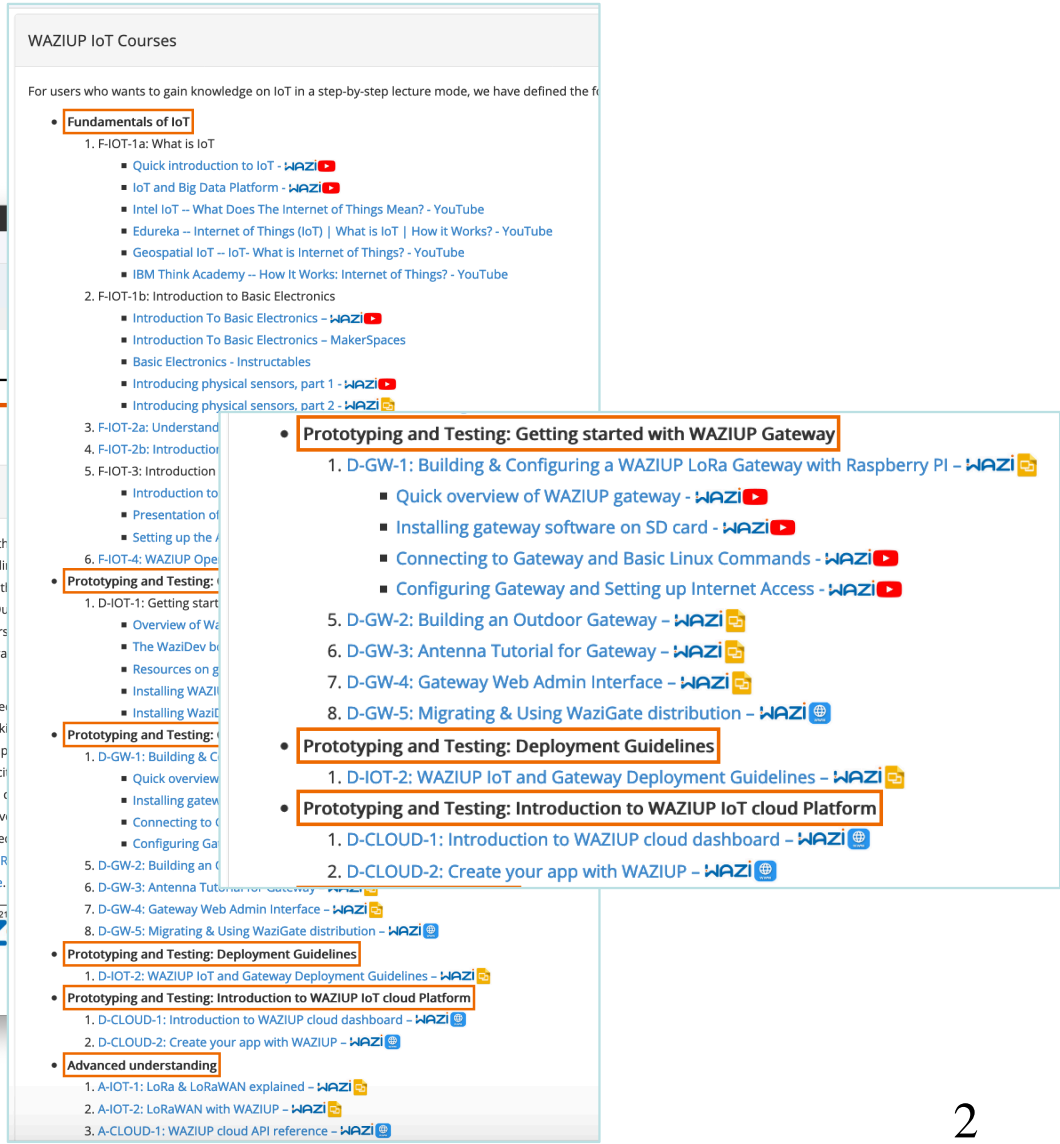
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France

IoT – from idea to reality

IoT Online Course

- http://diy.waziup.io





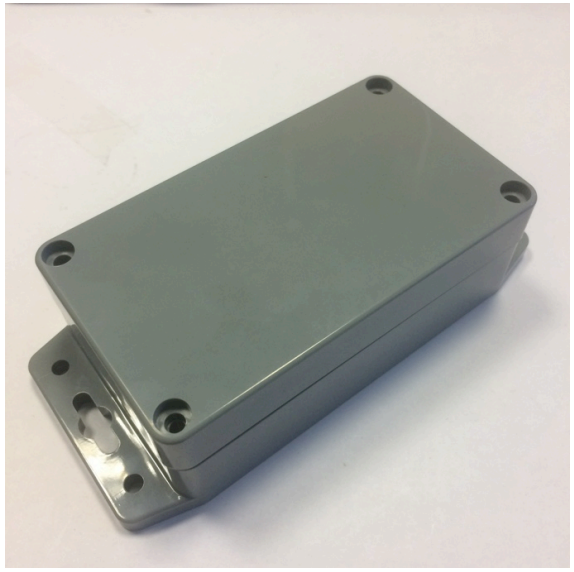
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Reading instructions

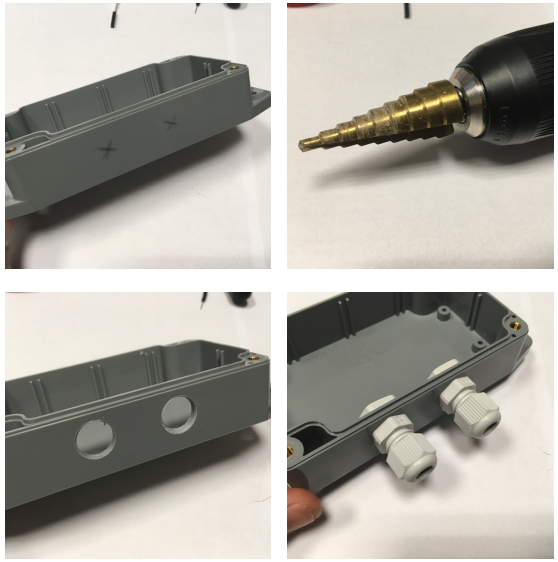
- Additional recommended tutorials:
  - Low-cost-LoRa-IoT-step-by-step.pdf
  - Low-cost-LoRa-IoT-outdoor-step-by-step.pdf
  - Low-cost-LoRa-GW-step-by-step.pdf
  - Low-cost-iot-hardware-parts.pdf
  - Low-cost LoRa IoT/GW FAQ
  - https://github.com/CongducPham/tutorials
- This document specifically focuses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

Packaging your IoT device: casing, sensors, antennas,…

Get a case for outdoor usage



The case should be water-resistant (not necessarily water-proof). Some cases (like the black one on the right) already have cable gland but most of them are just simple case and you may need to add your own cable gland by drilling appropriate holes.

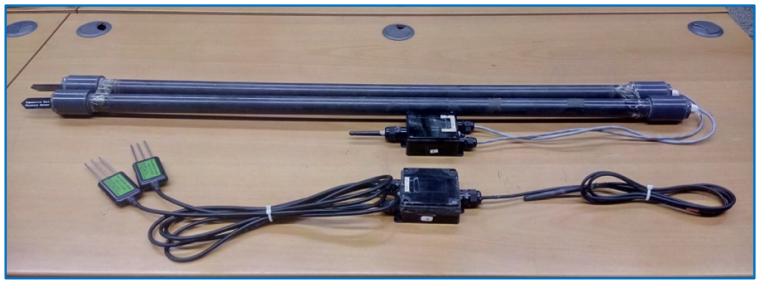


Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Connecting physical sensors

- When connecting sensor, you have to adapt the design so that your sensor is not going to be damaged by water, humidity, dust, sun, …
- In many cases, using a cable gland is enough

- In some cases, more complex design is needed to get the measure you need: e.g. soil humidity at larger depth



Protecting from rain, dust & sun

If you case has holes or cable glands, avoid placing them upward to limit water infiltration! Always tighten all cable glands.

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

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Sensing devices examples

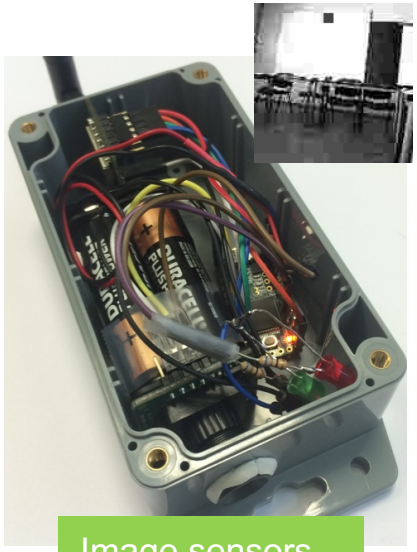
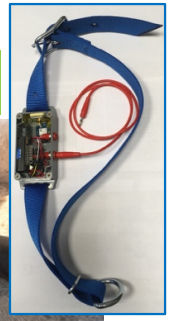
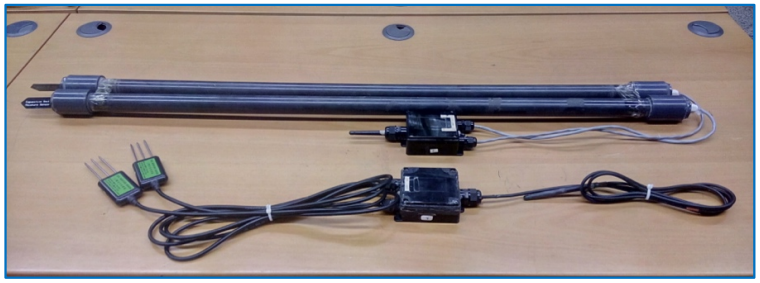


Image sensors

GPS collar





Soil Moisture

Photo from Unparallel

Bin presented at Woelab

Waste Mngt

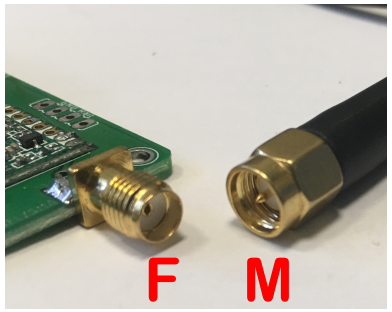
Photo from EGM

Buoy for water quality

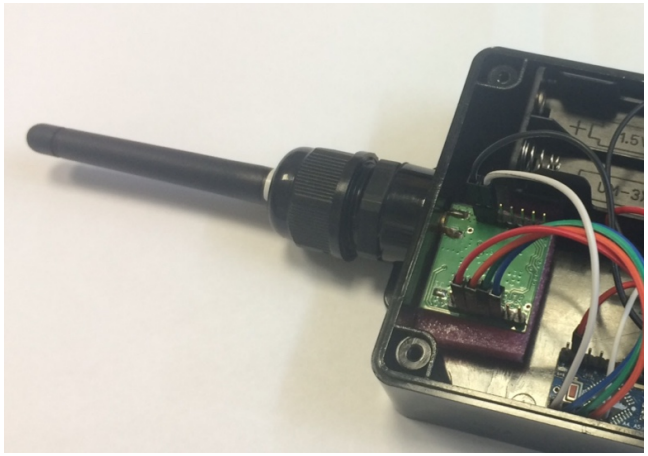
Weather Station

Connecting an antenna

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



Make sure that you have matching connectors!



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

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

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-5m
  - A ¼ wave monopole antenna WILL NOT provide good performance

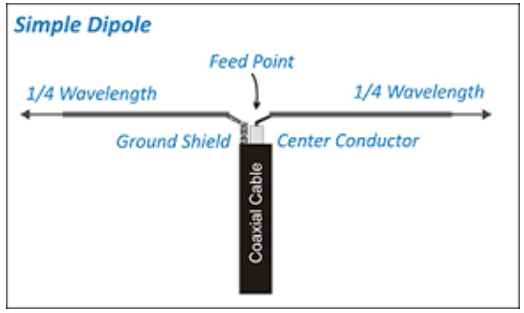
¼ wave monopole at the end of a coax cable is not good! Use a dipole or ground plane.

Connecting 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



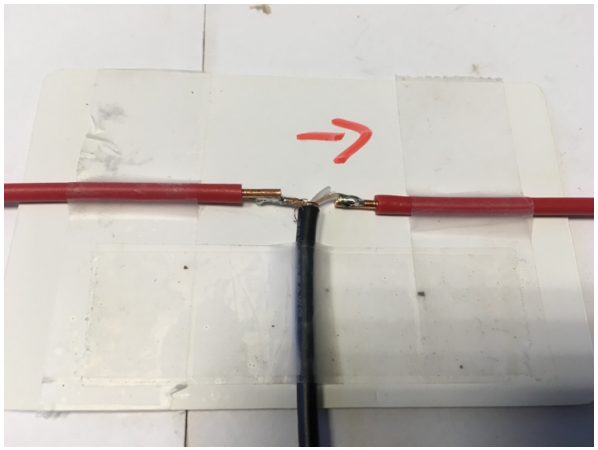
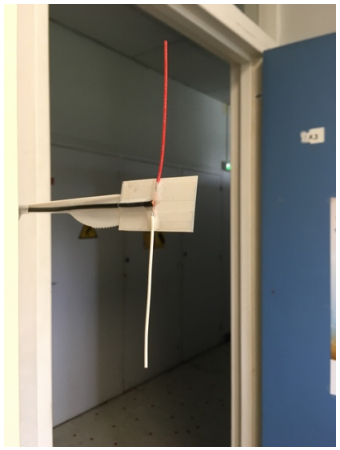
Simple dipole

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

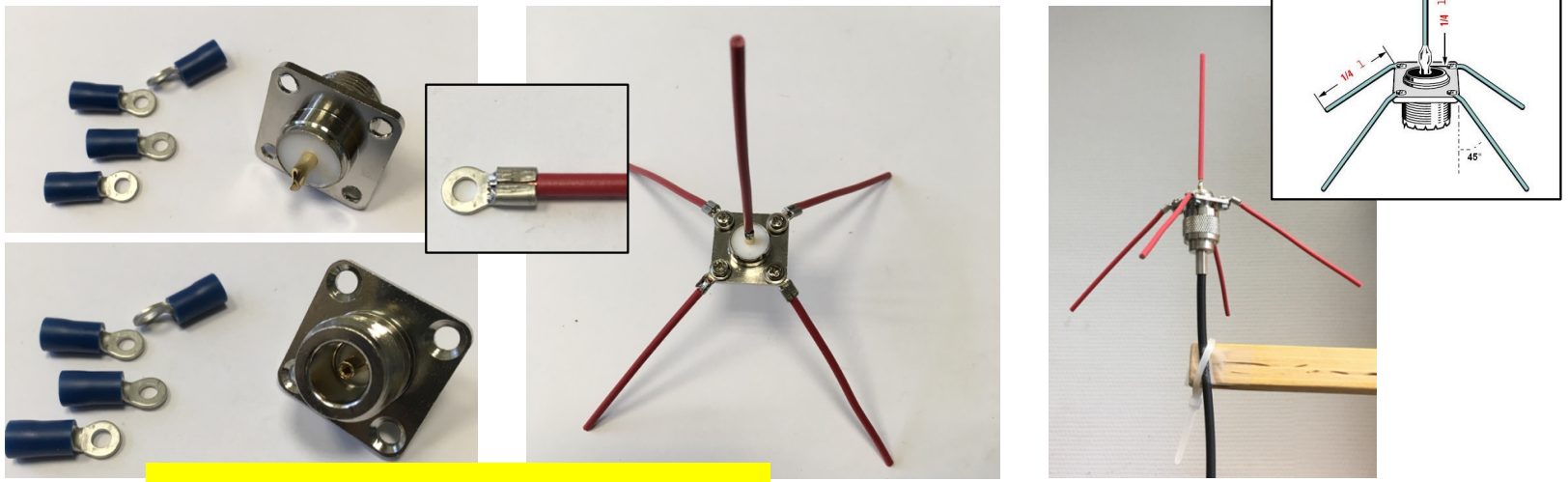
- Very simple dipole can be made with 2 pieces of ¼ wave wires. ¼ wave in 868 is about 8.2cm (16.4cm for 433MHz).


- 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.

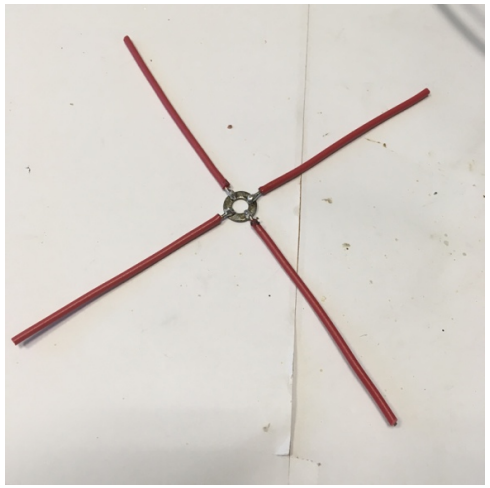
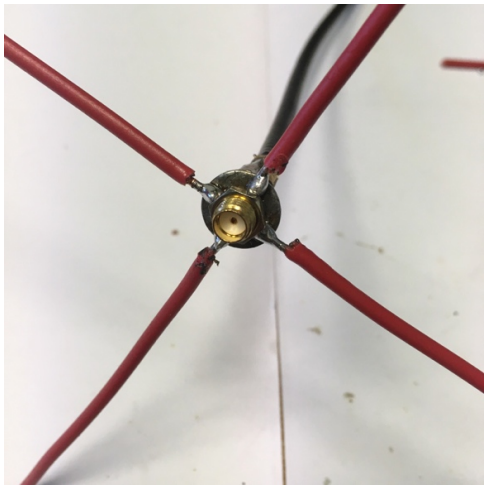
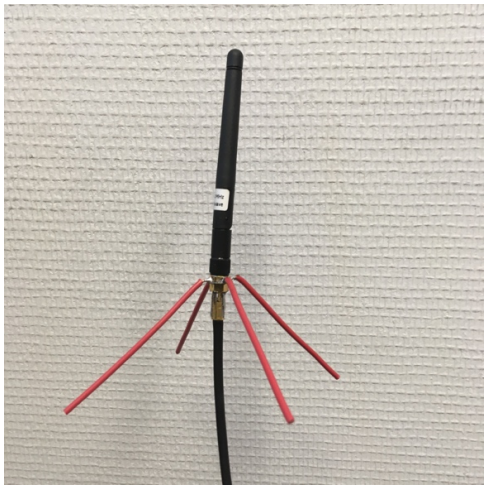


N Female Panel Mount Chassis

- 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 N male to SMA male

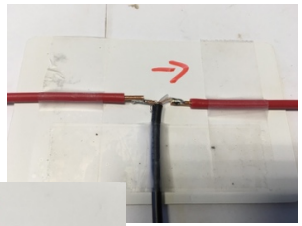
https://www.aliexpress.com/item-img/RG58-2m-N-Male-Jack-to-SMA-Male-M-M-RF-Coax-Pigtail-WLAN-Adapter-Adaptor/32616929641.html#

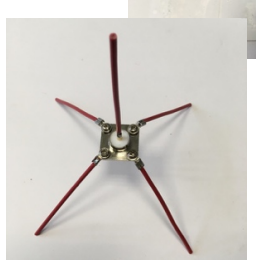
2m RG58 SMA male to SMA female

Summary of antennas for sensing device

- The easiest solution would be to buy a general purpose antenna for the frequency range you are operating but these antennas have very low performance
- If you want to try the DIY approach, try first the simple and easy-to-make dipole and 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!
- See D-GW-3 "Antenna Tutorial for Gateway"

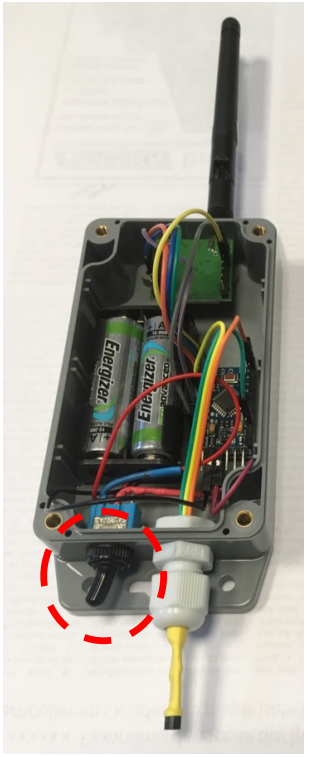
General purpose but far from optimal: will certainly introduce high attenuation in RF signal





Before powering your device

- 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 slides
- It can be useful to have a switch to easily set the device ON/OFF. Take a water resistant switch, see the Low-cost-iot-hardware-parts.pdf



Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Device software configuration

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Template configuration

[Arduino|WaziDev]_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 and 0 is for broadcast

```
////////////////////////////////////////////////////////////////////
// CHANGE HERE THE NODE ADDRESS
#define node_addr 8
////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////
// CHANGE HERE THE LORA MODE
#define LORAMODE  1
////////////////////////////////////////////////////////////////////
```

- If needed, change the measure interval (and transmission), in minutes.

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

Setting PA_BOOST

- The Semtech SX1272/76 has actually 2 lines of RF power amplification (PA): a high efficiency PA up to 14dBm (RFO) and a high power PA up to 20dBm (PA_BOOST)
- Some radio modules only wire the PA_BOOST and not the RFO: RFM95 for instance has only PA_BOOST line

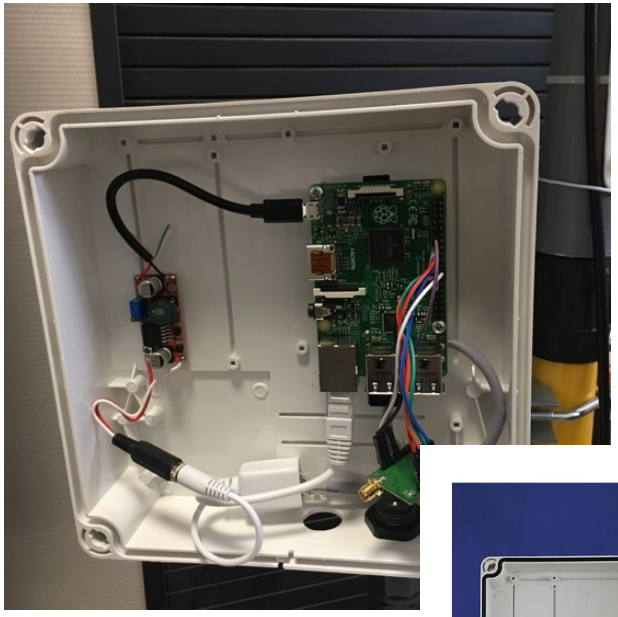
```
// IMPORTANT
////////////////////////////////////////////////////////////////////////////////////////////////////
//
// uncomment if your radio is an HopeRF RFM92W, HopeRF RFM95W, Modtronix inAir9B, NiceRF1276
// or you known from the circuit diagram that output use the PABOOST line instead of the RFO line
//#define PABOOST
////////////////////////////////////////////////////////////////////////////////////////////////////
```

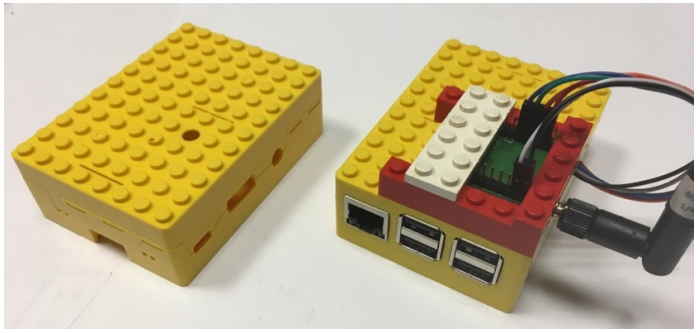
Uncomment the « #define PABOOST » line, compile and upload again

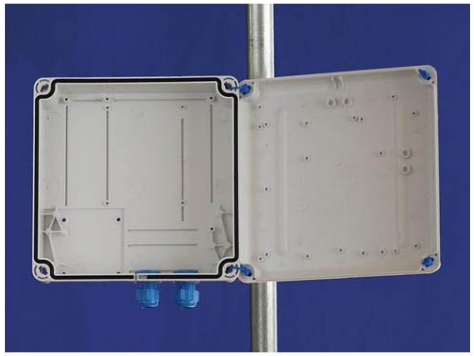
Packaging your gateway: casing, antenna, power, internet connection,…

Get a case for your gateway

- Various cases can be used: from very simple ones for an indoor gateway to more robust cases for outdoor usage

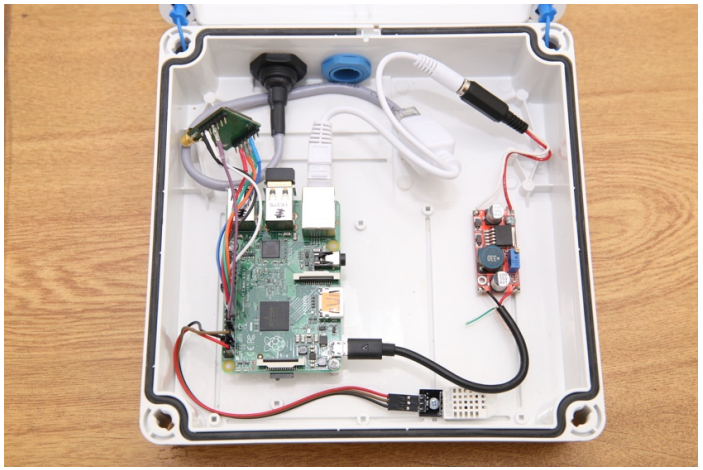






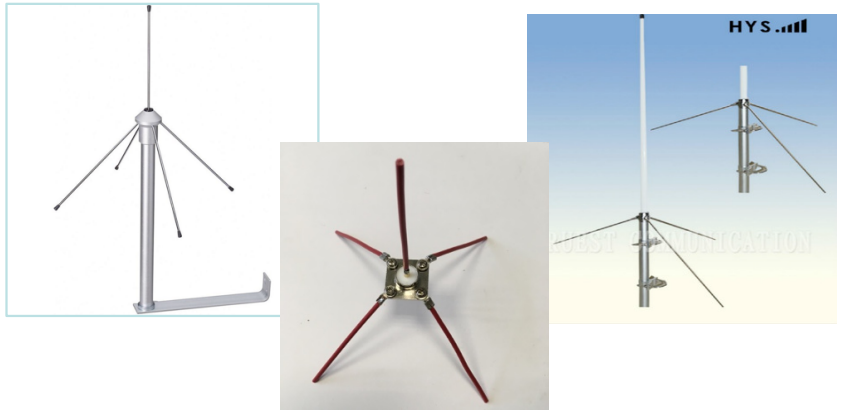
Installing the gateway

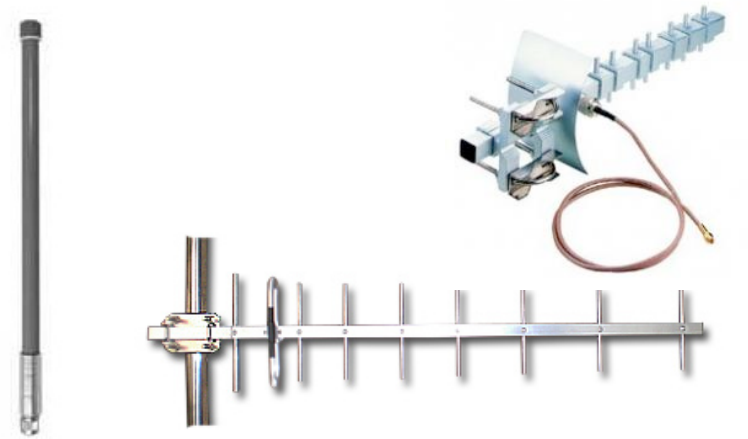
- Power the gateway either with PoE or directly with a 5V USB adaptor.
- If possible, it is much better to put the gateway indoor
- Try avoiding long antenna cable: 2m to 5m max
- If the gateway needs to be put outdoor because of the antenna cable constraints, don't forget to protect it from direct sun!
- 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
- You can easily use ground plane or dipole antennas (e.g. sleeve dipole). More complex high gain antenna or a directional Yagi antenna can be purchased depending on your budget and whether the deployment allows it





Internet connection

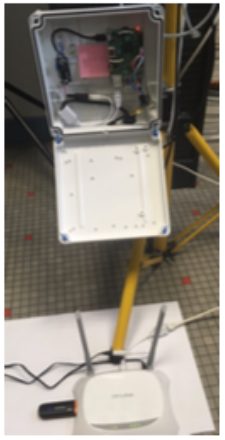
- See the gateway booklet

Connecting the gateway to the Internet

The best way to provide Internet to the gateway is through Ethernet via a DSL router for instance

The DSL router can be replaced by a 3G router. This solution is better than using a USB 3G dongle because of power issues.

The Loranga hat mentioned above is a great solution that provides high flexibility of deployment. We have collaboration with the Loranga development team and support of the board is included in the github distribution



External 3G router + Ethernet | 3G USB dongle | 2G/3G Loranga hat | Loranga hat on an RPI0

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Gateway software installation & configuration

Gateway software installation

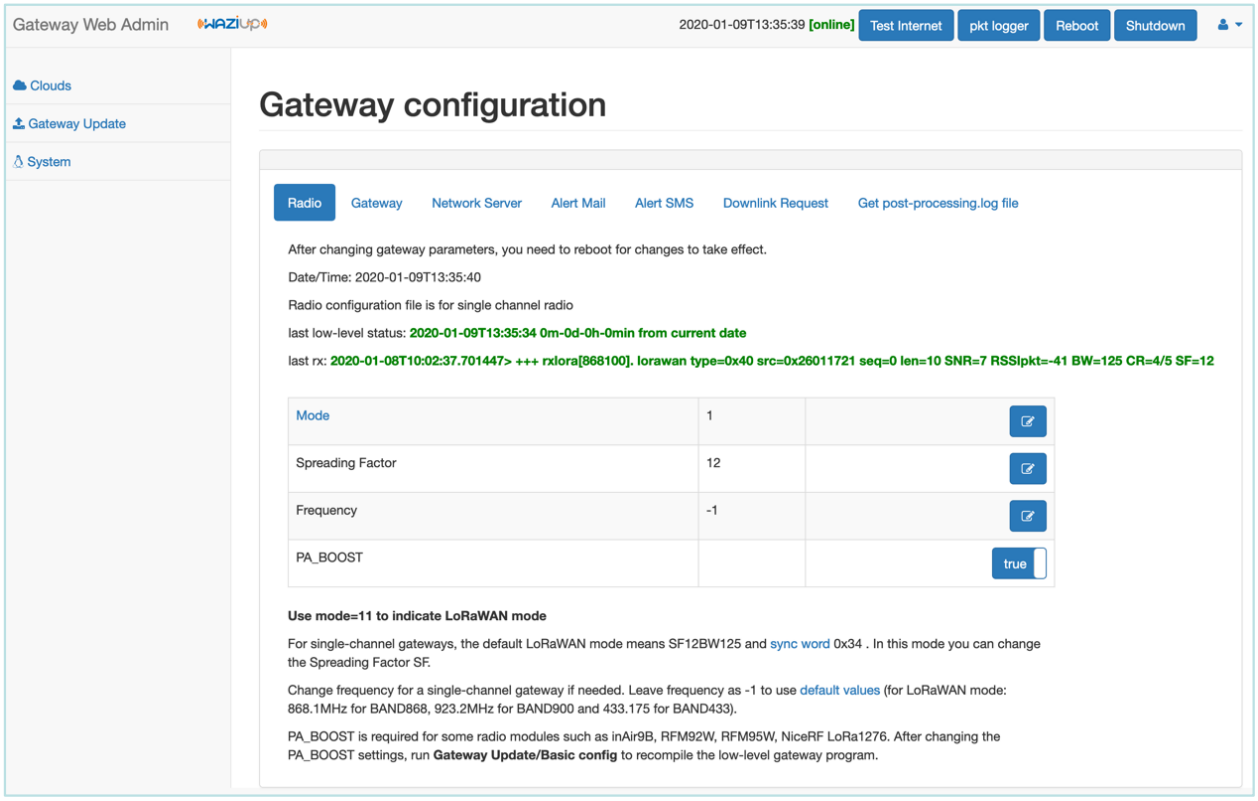
- An SD card image with a Raspberry Raspbian Jessie version is provided.
- You will need an 8GB SD card. Be careful, some SD cards will not work. This one has been successfully tested. It has to be class 10.
- Look at https://www.raspberrypi.org/documentation/installation/installing-images/ to see the 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.

Gateway access & configuration interfaces

- There are 2 gateway configuration interfaces
  - A web admin interface
  - A command line interface that needs ssh
- Look at the gateway course
  - D-GW-1: Building & Configuring a Low-Cost LoRa Gateway with Raspberry PI
- The web interface is sufficient for most users
  - Easy basic configuration and easy update
  - Pre-defined cloud configuration
  - dedicated course: D-GW-4 Gateway Web Admin Interface

Gateway web admin interface

- http://192.168.200.1/admin (with WiFi connection)
  - Login: `admin`
  - Password: `loragateway`

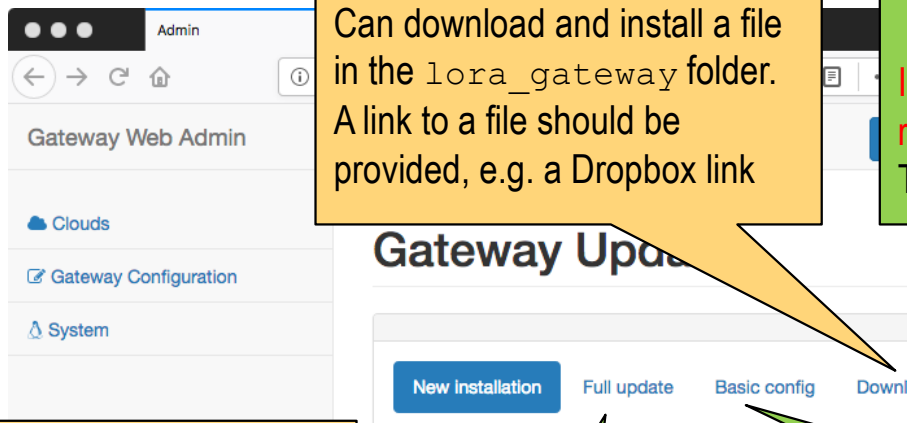


Gateway update

- The gateway must be updated to the latest version.
- Internet access for the gateway is necessary
- 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.

3

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.

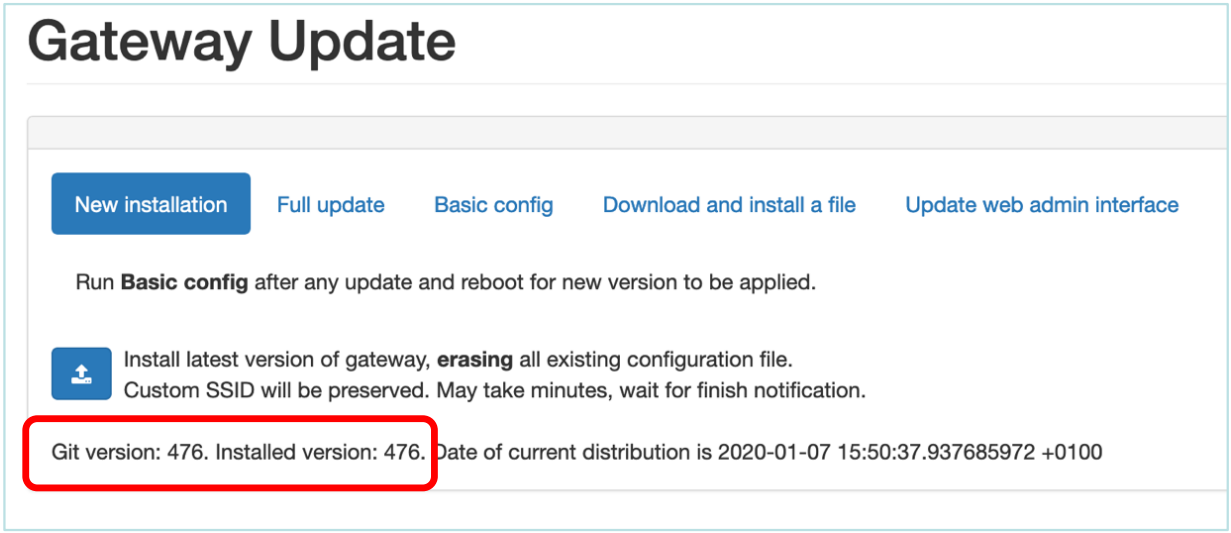
1

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

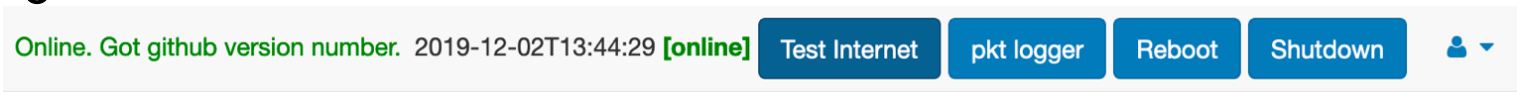
2

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

Software version number



- The software version number on github and the installed version number are displayed
- Click on Test Internet to obtain the latest software version number on github

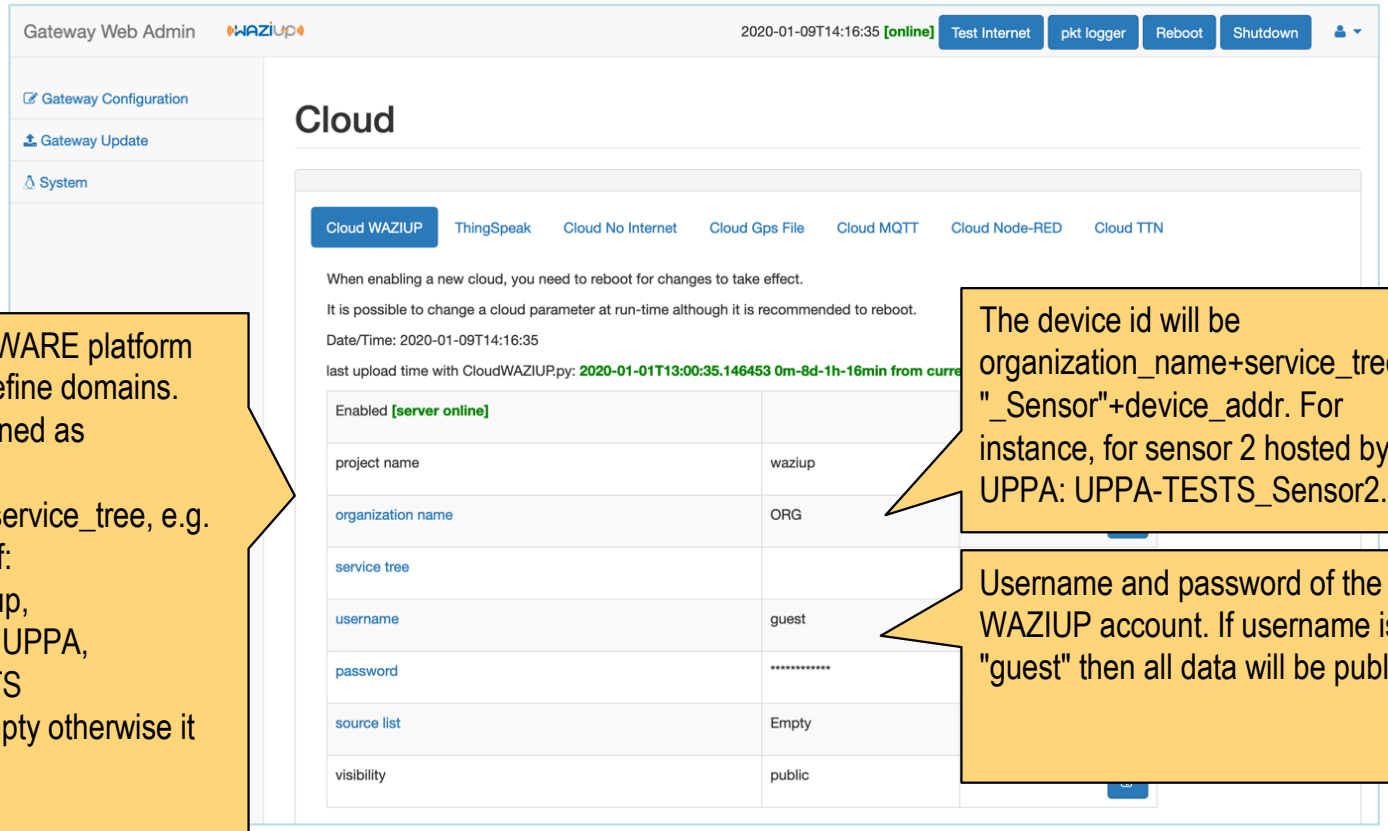


Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

WAZIUP Cloud configuration

Configuring WAZIUP cloud

The WAZIUP cloud tab is only available when key_WAZIUP.py is found



WAZIUP cloud uses FIWARE platform with the possibility to define domains. The domain will be defined as project_name+'-'+organization_name+service_tree, e.g. waziup-UPPA-TESTS if:
- project_name is waziup,
- organization_name is UPPA,
- service_tree is –TESTS

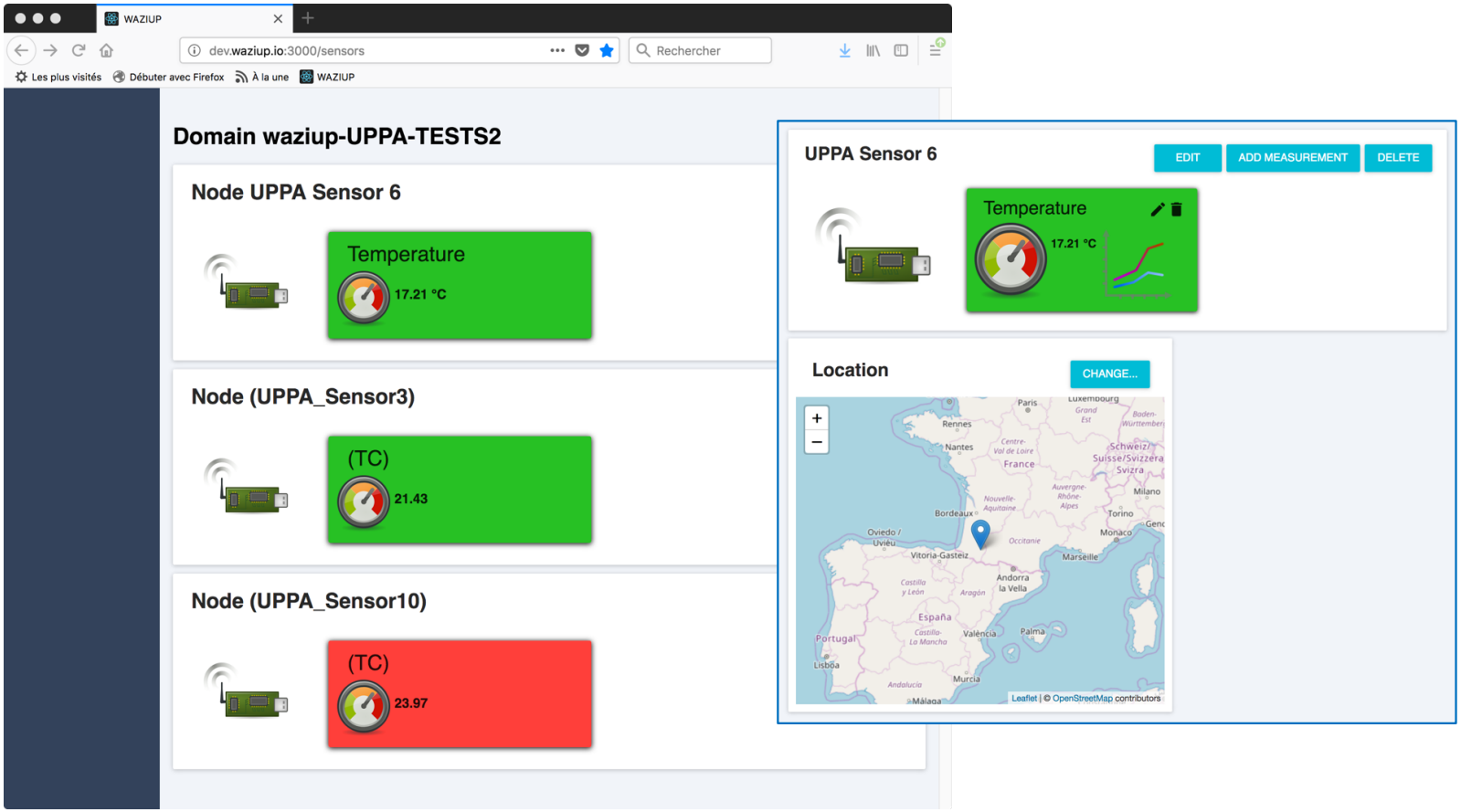
service_tree can be empty otherwise it must begin with a '-'.

The device id will be organization_name+service_tree+"_Sensor"+device_addr. For instance, for sensor 2 hosted by UPPA: UPPA-TESTS_Sensor2.

Username and password of the WAZIUP account. If username is "guest" then all data will be public

The WAZIUP cloud platform

- dashboard.waziup.io



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
```

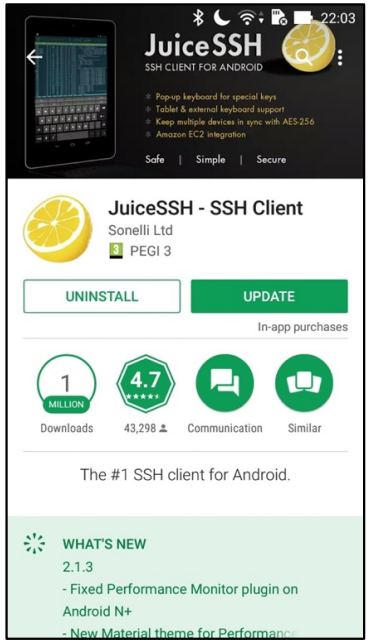
Gateway's simple command interface

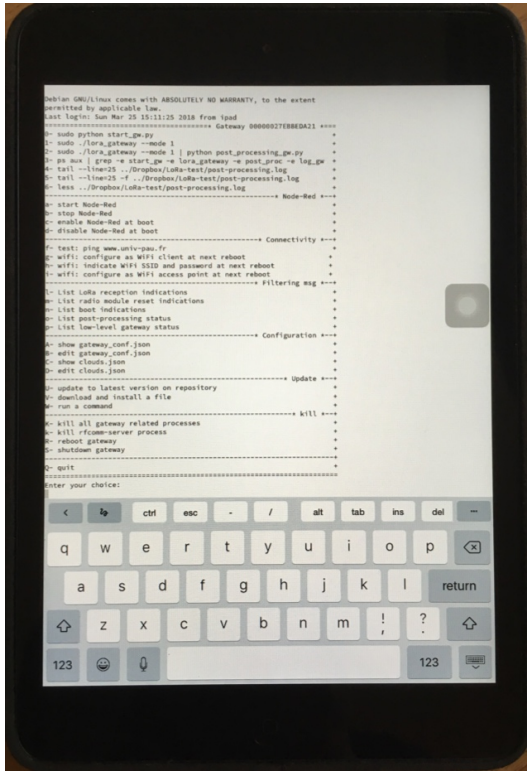
- Once logged on the gateway, you may directly enter in a simple command interface
- This command interface consists in a `cmd.sh` shell script
- In image versions after May 2017, this script is launched when you log into the gateway with ssh
- If this happens, select Q and hit RETURN to quit this interface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

iOS/Android smartphone or tablet

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





Manual 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

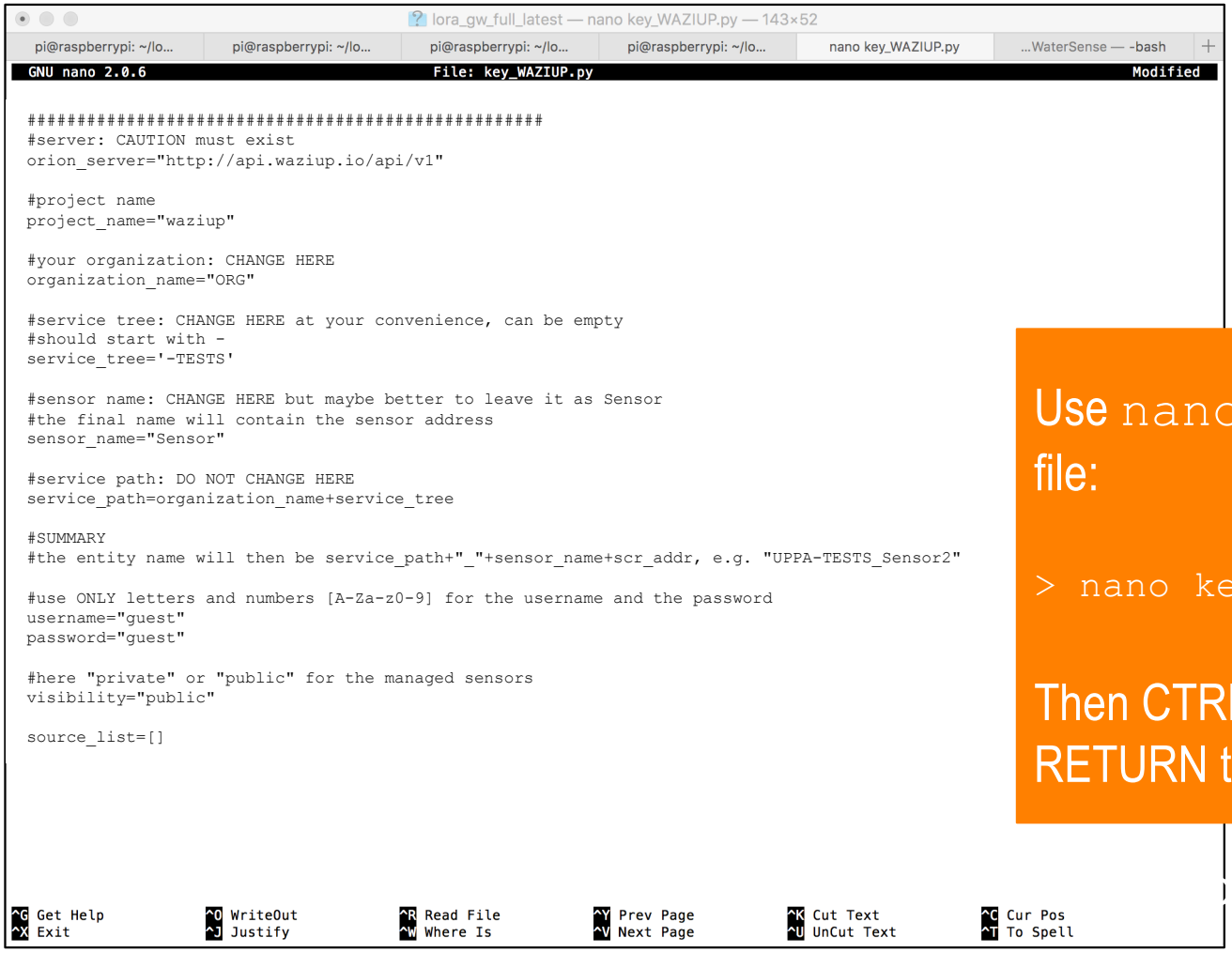
key_WAZIUP.py

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

#project name
project_name="waziup"

#your organization: CHANGE HERE
organization_name="ORG"

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

#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
#the entity name will then be service_path+"_"+sensor_name+scr_addr, e.g. "UPPA-TESTS_Sensor2"

#use ONLY letters and numbers [A-Za-z0-9] for the username and the password
username="guest"
password="guest"

#here "private" or "public" for the managed sensors
visibility="public"

source_list=[]
```

You MUST 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

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Connecting your gateway to the Internet

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

1-DHCP solution

- The simplest way to connect your gateway to Internet is through a DHCP-enabled network
- If you use a DSL/3G router, the router will be the DHCP server
- If your company/organization has a local network, it is most likely that there is a DHCP server somewhere
- You can also connect your gateway to a computer/laptop which will share its Internet connection, thus acting as a DHCP server. That usually needs some simple configuration on the computer/laptop

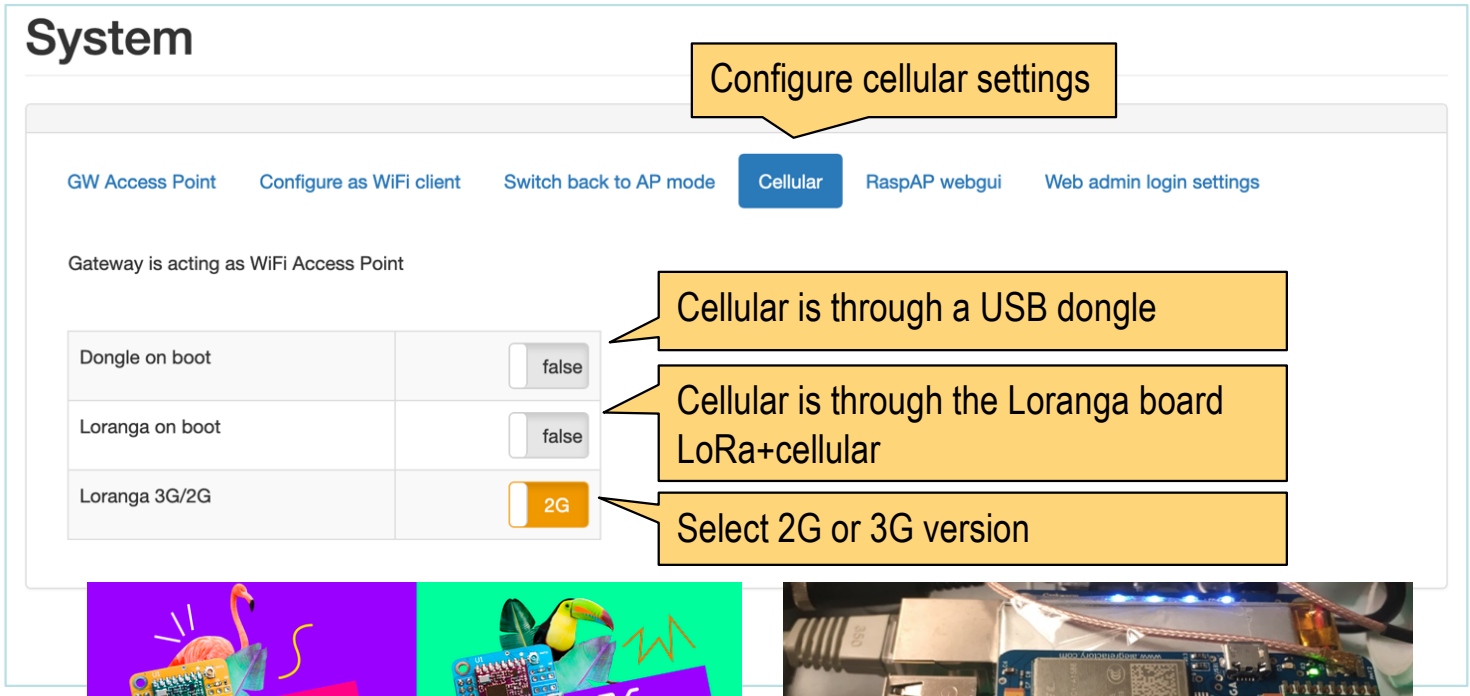
2-Using a 3G dongle (1)

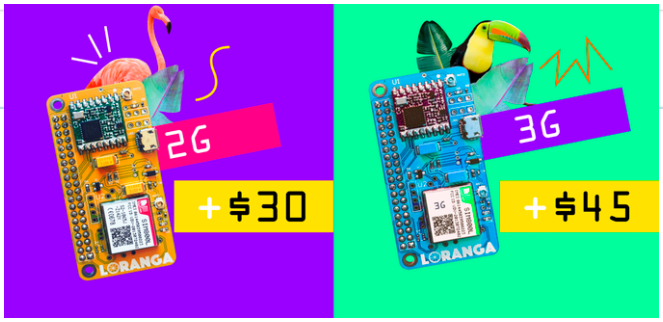
- If you use a 3G dongle that you directly connect to the gateway to get Internet connectivity, be sure to use a DC adaptor that provides at least 2A
- Beware that not all dongle work. The Huawei E220-based dongles have been tested successfully
- Use the web admin interface to configure the gateway for cellular access
- Then reboot

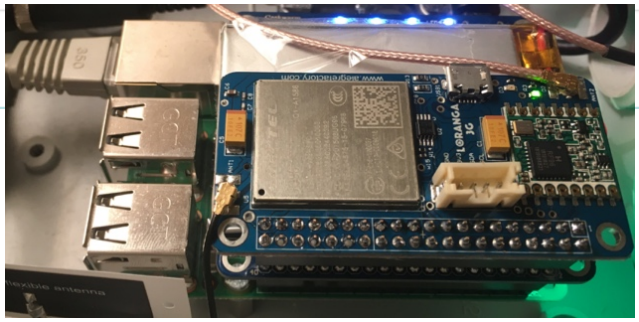
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

2-Using a 3G dongle (2)

- Configure cellular for Internet access

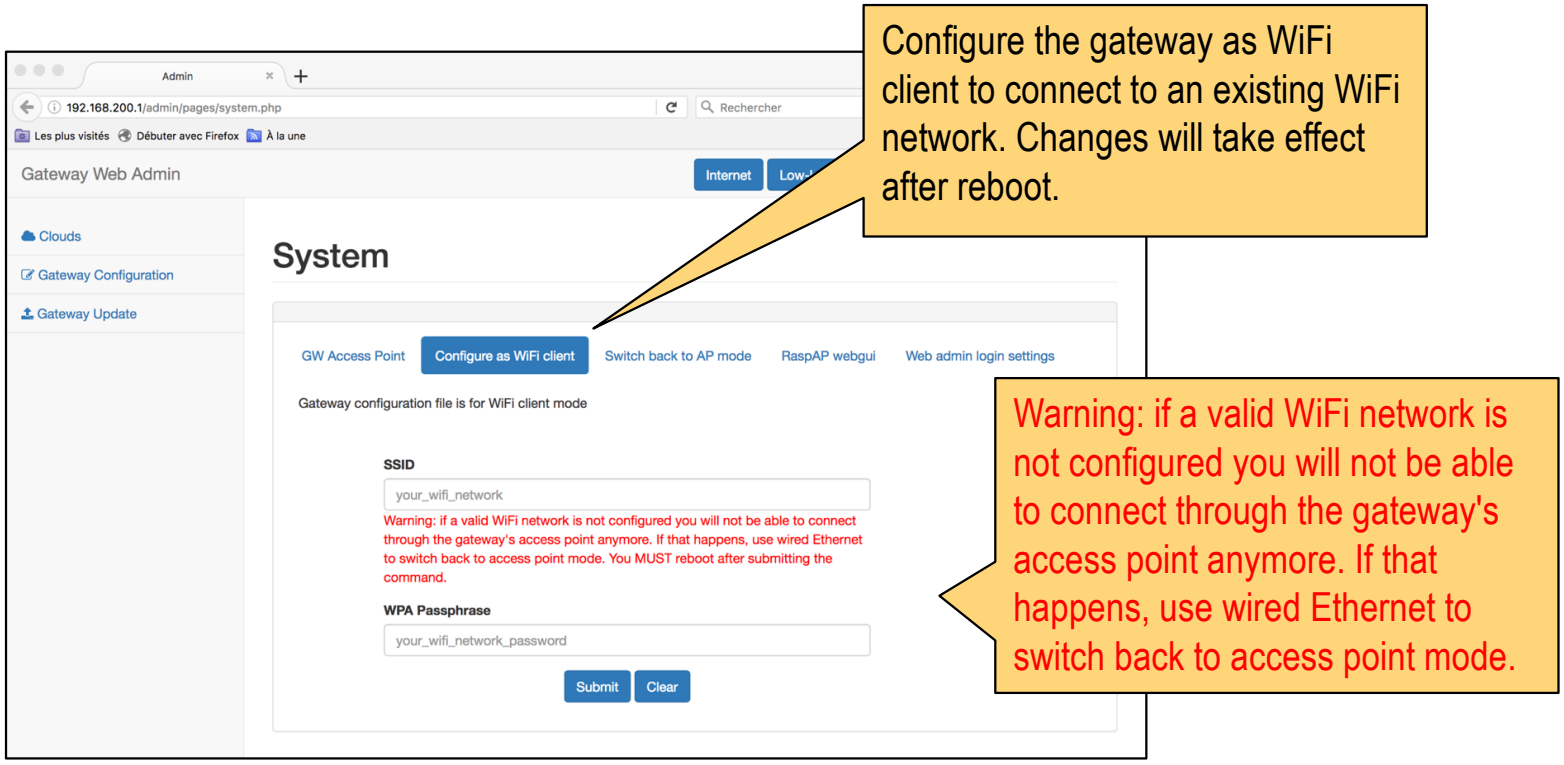






3-Connect to an existing WiFi

- Configure the gateway as WiFi client to have Internet connection through an existing WiFi network



When everything is ready…

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Last step before deployment

- 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

Gateway advanced configuration

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

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, `key_AppKey.py` contains a list of allowed app keys

```
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/WaziDev]_LoRa_temp` is an extended version of `[Arduino/WaziDev]_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 device sketch (`Arduino_LoRa_temp`)

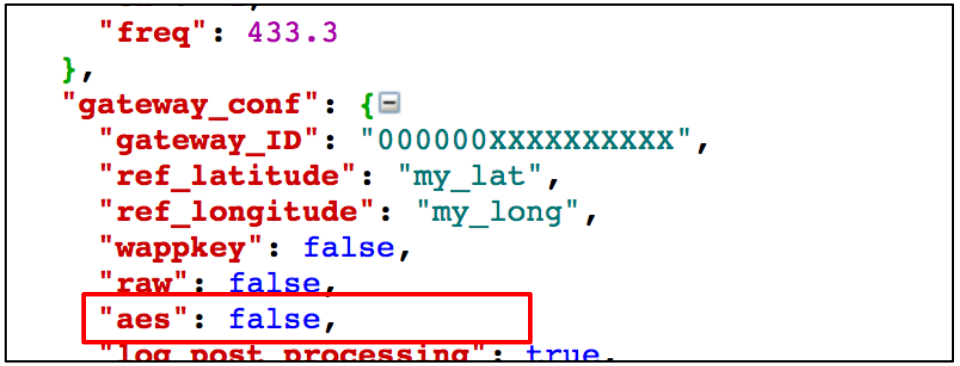
```
//this is the default as LoRaWAN example
unsigned char AppSkey[16] = { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F, 0x3C };

//this is the default as LoRaWAN example
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 `key_AES.py` script on the gateway if local decryption is needed

```
device_key = {
                #do not remove default entry
                #replace default key by your default key if necessary
                #use 4-byte hex format, e.g. 26014821
                "default" : {
                                "AppSKey" : '2B7E151628AED2A6ABF7158809CF4F3C',
                                "NwkSKey" : '2B7E151628AED2A6ABF7158809CF4F3C'
                },
                "00000006" : {
                                "AppSKey" : '2B7E151628AED2A6ABF7158809CF4F3C',
                                "NwkSKey" : '2B7E151628AED2A6ABF7158809CF4F3C'
                },
                "000000FF" : {
                                "AppSKey" : '0540AC89349E0C60650D50CF00F01C0D',
                                "NwkSKey" : '0110FF0060BA0AE08712606B0A508F01'
                }
}
```

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.



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

Testing connectivity and performing range tests

Testing device<-> gateway connectivity

- 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.

Remote access to your gateway

Remote access

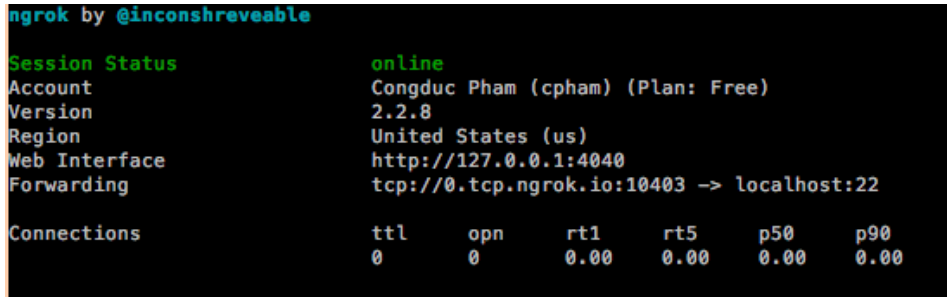
- It is usefull to be able to remotely get access to your deployed gateway for management and configuration purposes
- There are many remote access possibilities and the current distribution proposes 2 alternatives
  - `ngrok` from https://ngrok.com
  - `weavedconnectd` (remot3.it) from https://www.remot3.it
- `ngrok` is simpler but requires an interactive scenario
  - You need somebody to log on the gateway locally to provide remote access information to the gateway
- `weavedconnectd` uses centralized web platform to provide remote access information
  - If you have an account you will be able to connect to your gateway remotely without interacting with a physical person

Using ngrok

- The `ngrok` binary is already included in the github distribution (`/home/pi/lora_gateway/ngrok`)
- You need to create an account on https://ngrok.com/signup. Then go to https://dashboard.ngrok.com to get your authentication token
- On the Raspberry, run the following commands
  - `> cd /home/pi/lora_gateway`
  - `> ./ngrok authtoken <your_auth_token>`
- Replace <your_auth_token> by the token indicated in your `ngrok` dashboard

Running ngrok

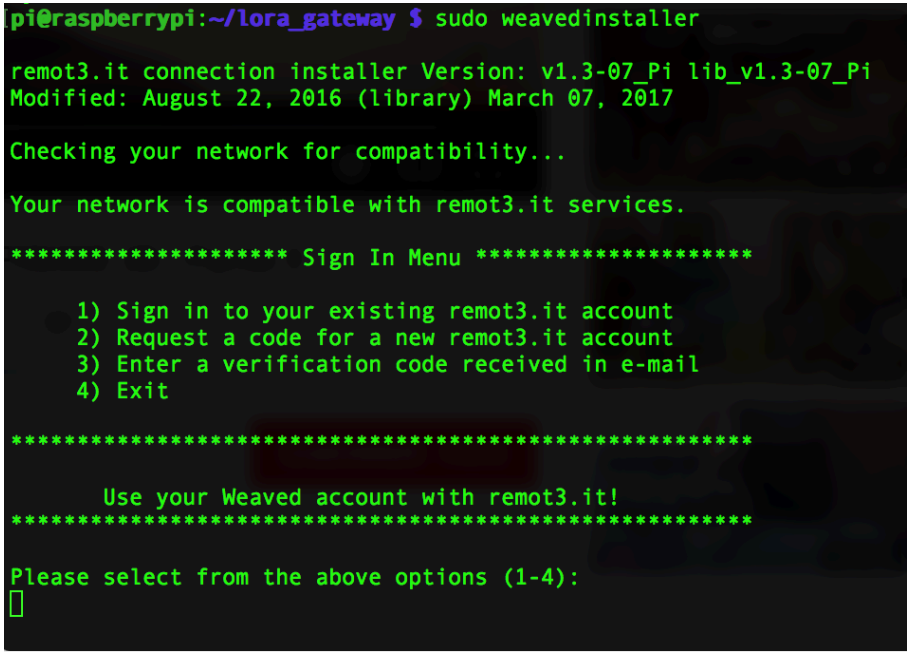
- To start ssh remote access from anywhere in the world, even if your gateway is behind a local LAN, a physical person must run the following command on the gateway:
  - `> cd /home/pi/lora_gateway`
  - `> ./ngrok tcp 22`
- `ngrok` will display something similar to



- Then, the local person should provide (using mail, SMS, ...) the forwarding `info 0.tcp.ngrok.io:10403` for you to use ssh
  - `>ssh -p 10403 pi@0.tcp.ngrok.io`

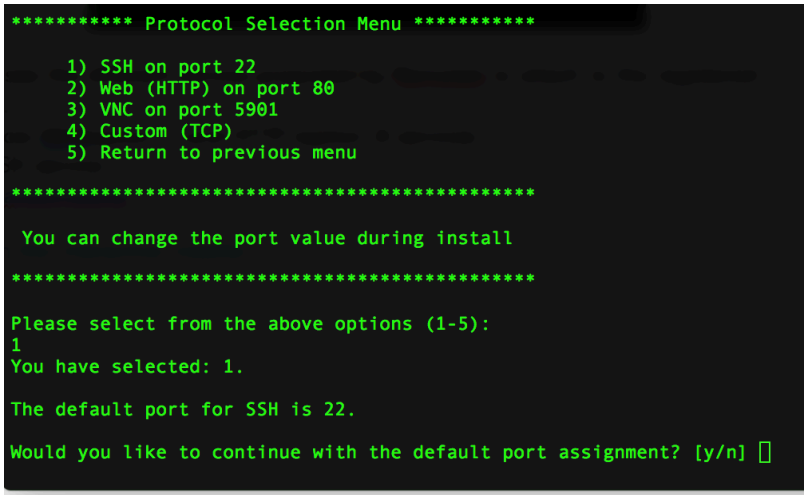
Using weavedconnectd (remot3.it)

- The `weavedconnectd` and `weavedinstaller` binaries from remot3.it platform are already included in the SD card image
- You need to create an account on https://www.remot3.it/web/index.html. An account allows as many gateways as you want
- Then you have to register a gateway with `weavedinstaller`
  - `> sudo weavedinstaller`
- Use option 1 to log with your account
- You will then need to give a name to your gateway
  - For instance UPPA_TEST_GW_96



Create services

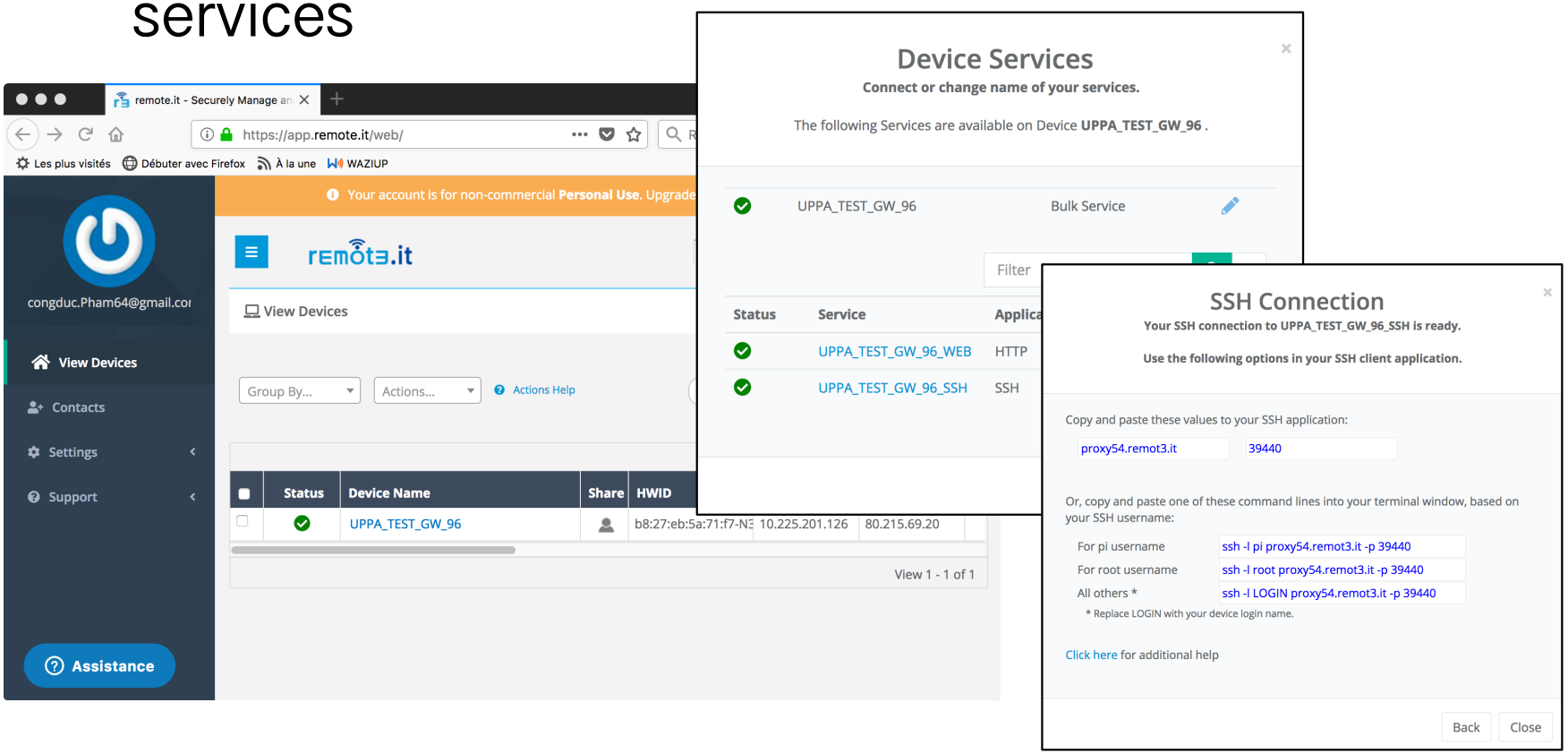
- You will then need to create a service (SSH or TCP for instance)
- Use `weavedinstaller` to install services that you want to use
- Here, the SSH service
  - You have to name the service, e.g.
  - UPPA_TEST_GW_96_SSH
  - SSH service allows ssh remote login



- Install a TCP service to allow access to the gateway's web admin interface

The remot3.it dashboard

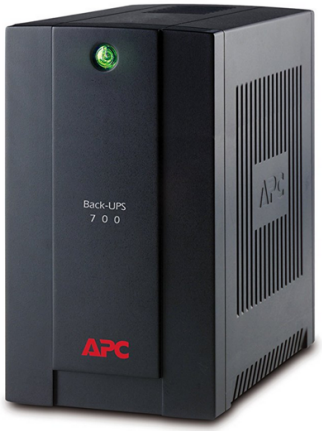
- If you log on your remot3.it account you should see your gateways and get access info to the installed services



Advanced power options for your gateway

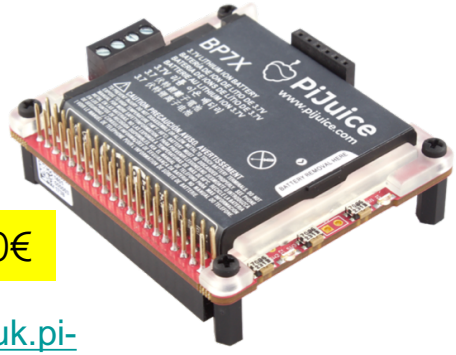
Dealing with unstable power sources

- If your premise suffers from frequent power shortage or unstable power, you can invest in an office UPS (uninterruptible power source) system
- These systems are very common and are quite affordable. Designed to protect computers or other electronic devices, they can of course protect and power your Raspberry gateway as well
- Autonomy depends on internal battery capacity (which has price impact on the UPS system)



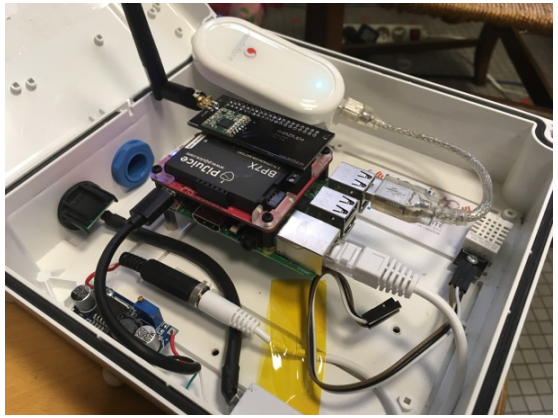
UPS hats for RPI

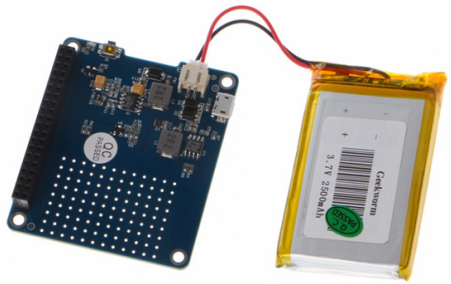
- There are UPS hats for Raspberry that can be a good alternative to office UPS if you only need to protect your gateway
- The PIJuice with its 1820mAh battery shipped with the hat can power your gateway for about 2 hours
- There are also cheaper, simpler UPS hats from China manufacturer (2500mAh battery included



About 50€

https://uk.pi-supply.com/products/pijuice-standard



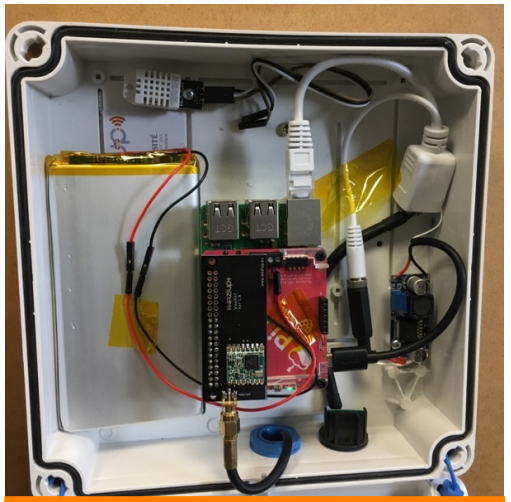


About 15€

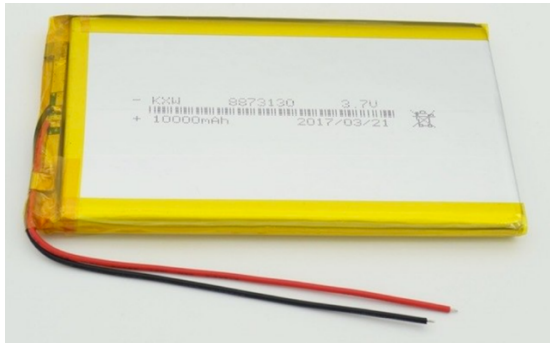
https://fr.aliexpress.com/item/UPS-HAT-Board-Module-2500mAh-Lithium-Battery-For-Raspberry-Pi-3-Model-B-Pi-2B-B/32882666003.html?spm=a2g0s.9042311.0.0.40696c37G298M7

Using higher capacity battery with UPS hats

- For the PIJuice, there is a 2300mAh battery to replace the default 1820mAh
- Both aforementioned hats can also charge any 3.7v **single cell** LiIon/LiPo battery
- You can therefore also connect most of smartphone LiIon batteries, small LiPo batteries designed for drone&quadcopter (middle) or high-capacity (>10000mAh) batteries.
- For the PIJuice, more info at https://github.com/PiSupply/PiJuice/tree/master/Hardware



About 11h of autonomy with the single cell LiIon 10000mAh battery connected to the PIJuice

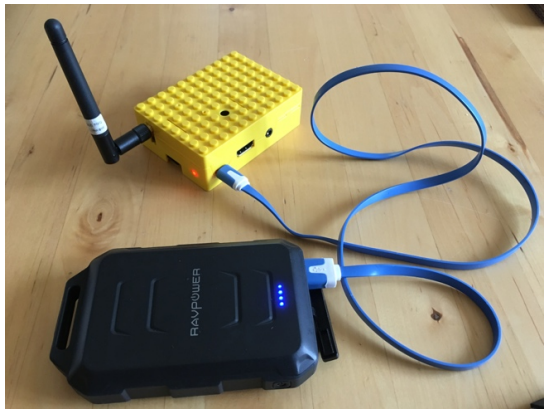


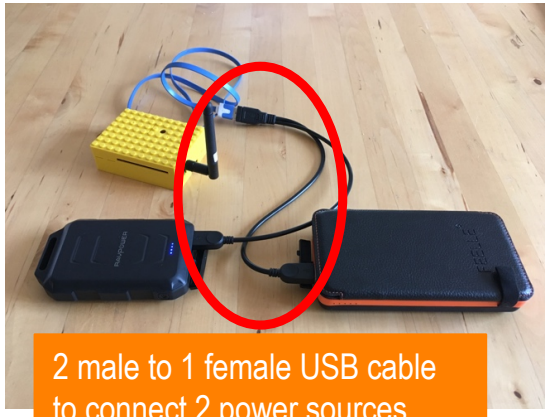
Autonomous gateway: Using a 12V car battery

- A 12V car battery is actually a very high capacity battery (from 60000mAh to 90000mAh)
- You can use simple 12V-5V car USB converter that are easily available to power your Raspberry gateway
- Take at least a 2A converter to avoid insufficient current issue
- You can expect about 5 days of autonomy when battery is fully charged

Autonomous gateway: mobility scenario

- You can use an external USB power bank that is easily available. Take a 10000 or even 20000mAh pack (left)
- You can additionaly use a dual USB cable to switch battery without interrupting your gateway (middle). After connecting the second battery, you need to switch it ON before removing the first battery
- These battery packs can also be solar-charged to be charged during the day (right)





2 male to 1 female USB cable to connect 2 power sources

About 40h of autonomy with a 20000mAh battery

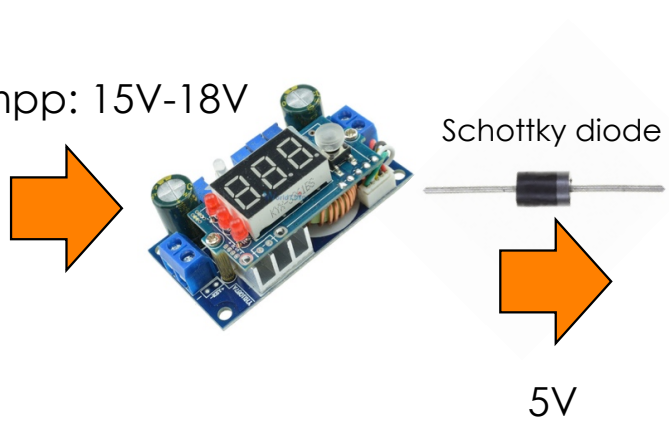
Powering gateway with solar panels

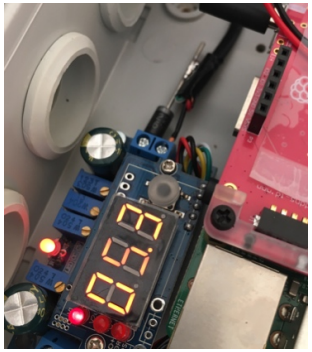
- The UPS hats can also be used to power your gateway with a solar panel
- More info on solar panel https://www.solar-facts.com/
- Most portable solar panel (left) have USB output (5V) and can therefore be directly plugged into the hat's onboard micro USB (middle for PIJuice hat) which accepts voltage in the range of 4.2V-10V. See video link.

https://raspi.tv/2017/pijuice-testing-the-software-and-hardware-plus-6w-40w-solar-panels-video

Connecting larger solar panels

- To really achieve a fully uninterruptible power supply, a larger solar panel and a high capacity LiIon/LiPo battery (e.g. 10000mAh) should be used (more info: https://www.solartechnology.co.uk/support-centre/calculating-your-solar-requirments)
- Large solar panels are usually 12V/24V systems (more info on solar panel specifications: https://www.altestore.com/blog/2016/04/how-do-i-read-specifications-of-my-solar-panel/)
- You probably need to regulate output from the solar panel to an acceptable range for your UPS hat, e.g. 4.2V-10V for the PIJuice, by using an MPPT (Maximum Power Point Tracking) controller with DC-DC stepdown
- More info on MPPT: http://www.leonics.com/support/article2_14j/articles2_14j_en.php

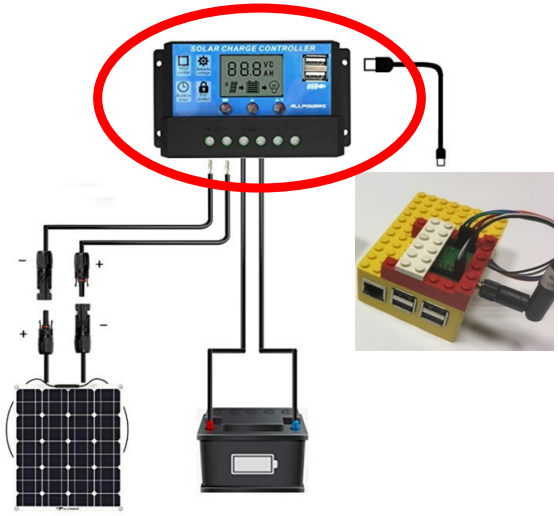


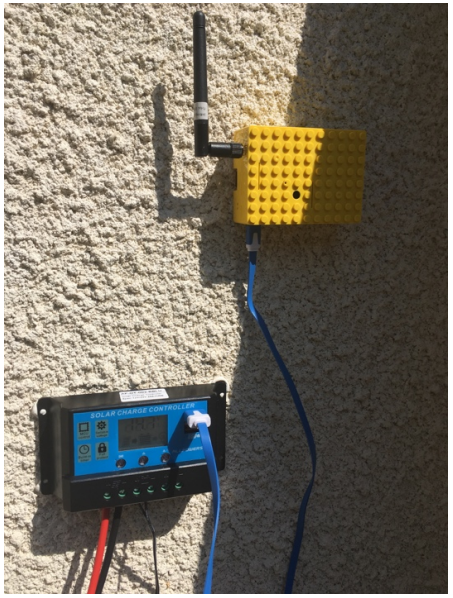


Using 12V battery and 12V solar panel

- There are very affordable solar charge controller to connect a 12V solar panel to your 12V car battery to power your gateway

https://fr.aliexpress.com/item/-/32904671590.html?spm=a2g0s.13010208.99999999.259.1dd23c00BnOdQd





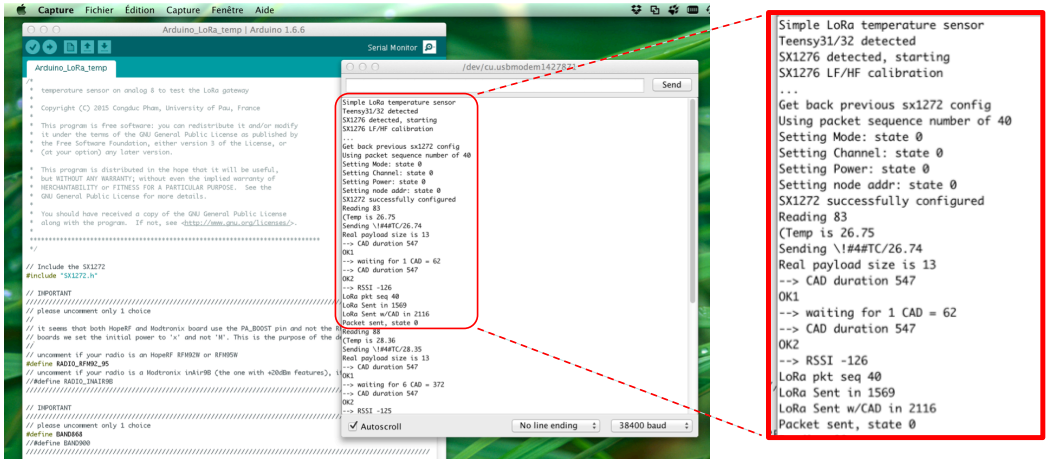
https://fr.aliexpress.com/item/Dokio-Brand-Solar-Panel-China-100W-Monocrystalline-Silicon-18V-1175x530x25MM-Size-Top-quality-Solar-battery-China/32802702078.html?spm=a2g0s.13010208.99999999.259.a8d33c00HQzpLB

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham

Troubleshooting & FAQ

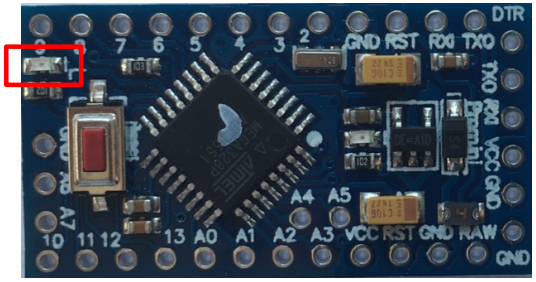
How can I know the sensor node is sending data?

- If you can connect the sensor node to a computer to use a serial tool (e.g. Arduino IDE serial monitor)



Otherwise, check that when powered on, the activity led goes through the following sequence:

- Fast blinking (booting)
- Off for some seconds (working)
- On for about 1s (transmitting)
- Off (sleeping)

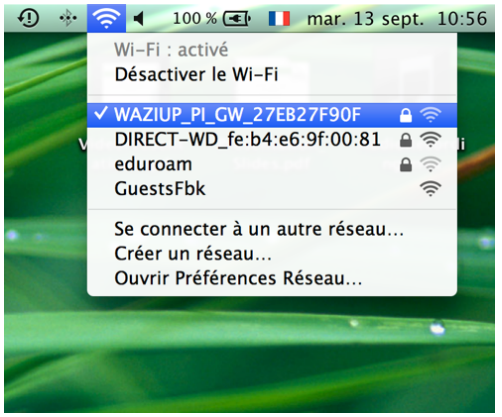


You can see the output from the sensor if it is connected to your computer. Use the Arduino IDE « serial monitor » to get such output, just to verify that the sensor is running fine, or to debug new code. Be sure to use 38400 baud. If you get the "Packet sent, state 0" result, it is most likely that your device is sending OK, but to be sure, you need to check data reception on the gateway.

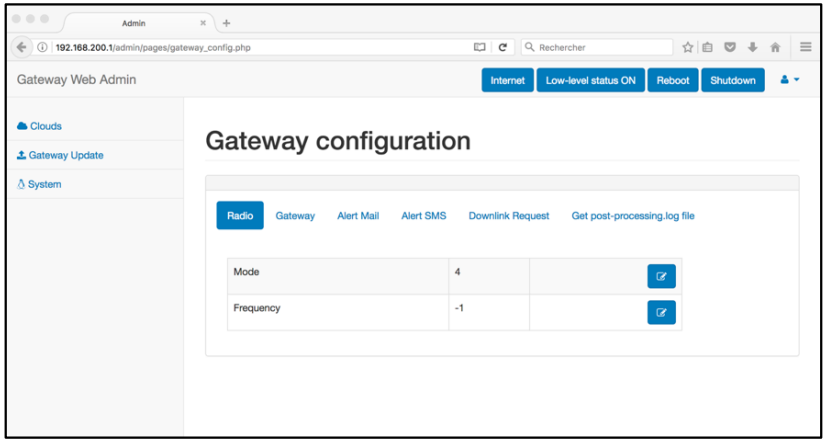
How can I check my gateway is booting properly?

- If your gateway is booting properly, you should see the gateway's WiFi:

  `WAZIUP_PI_GW_xxxxxxxxxx`



- Connect to that WiFi and then check that you can connect to the web admin interface
- http://192.168.200.1/admin
  - Login: admin
  - Password: loragateway

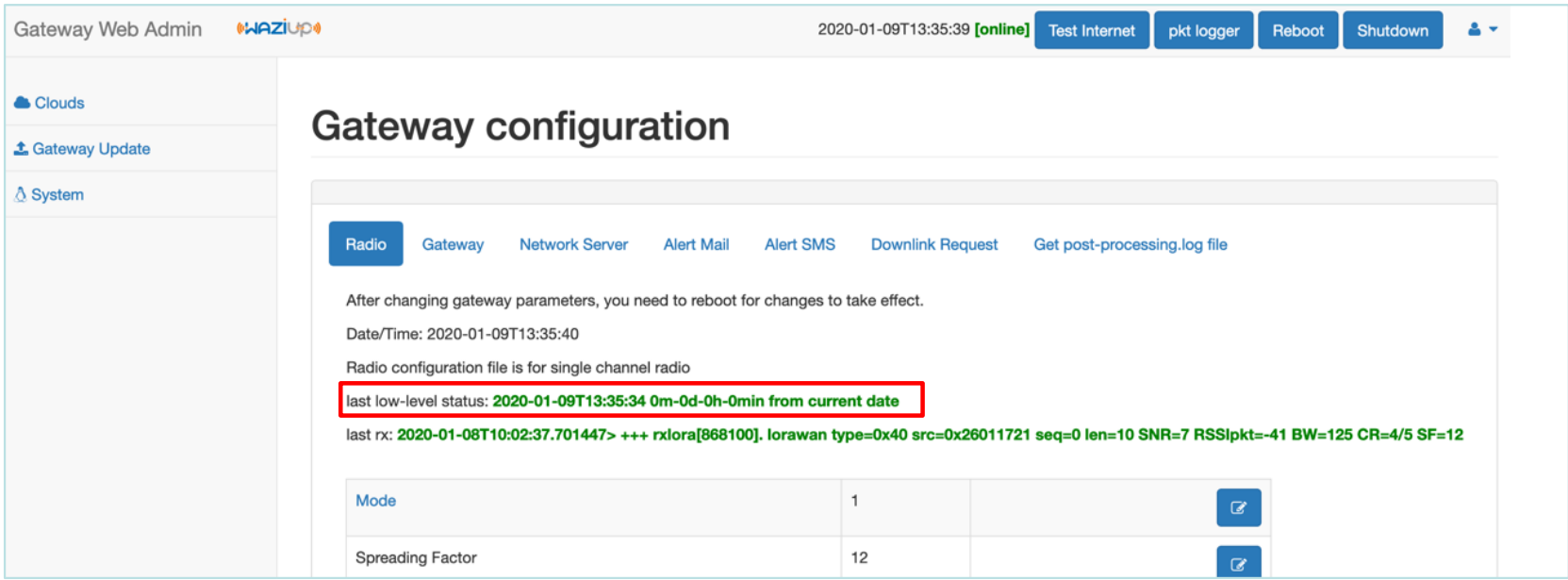


How can I know if gateway is connected to Internet?

- Use the web admin interface and check whether it is indicating online or offline

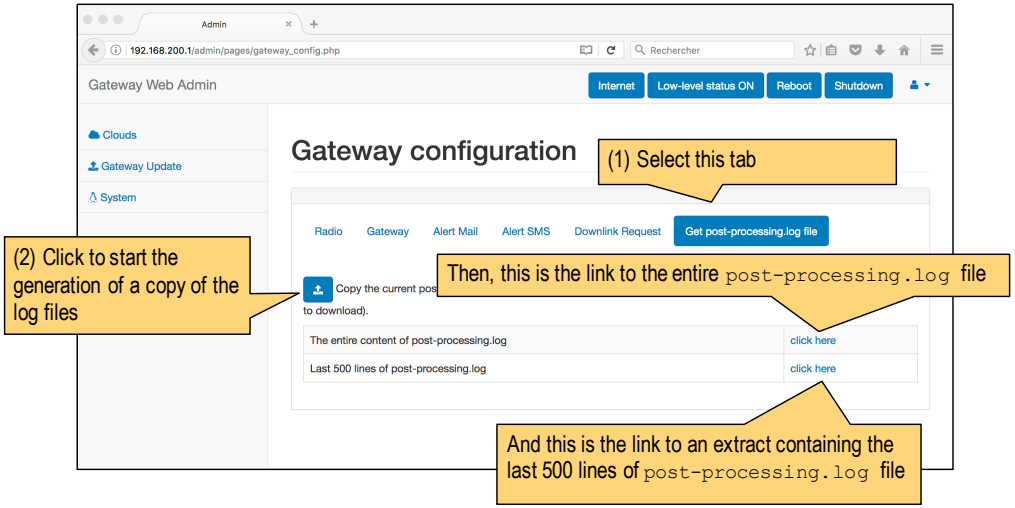
How can I know the LoRa module of the gateway works properly? First method

- Use the web interface and look for the low-level status indication
  - The low-level gateway issues every 10min a status
  - After boot, the gateway clock may not be already synchronized, so wait for more than 10mins



How can I know the LoRa module of the gateway works properly? Second method

- Use the web admin interface to get the gateway's log file after the gateway has booted
- You should see something similar to



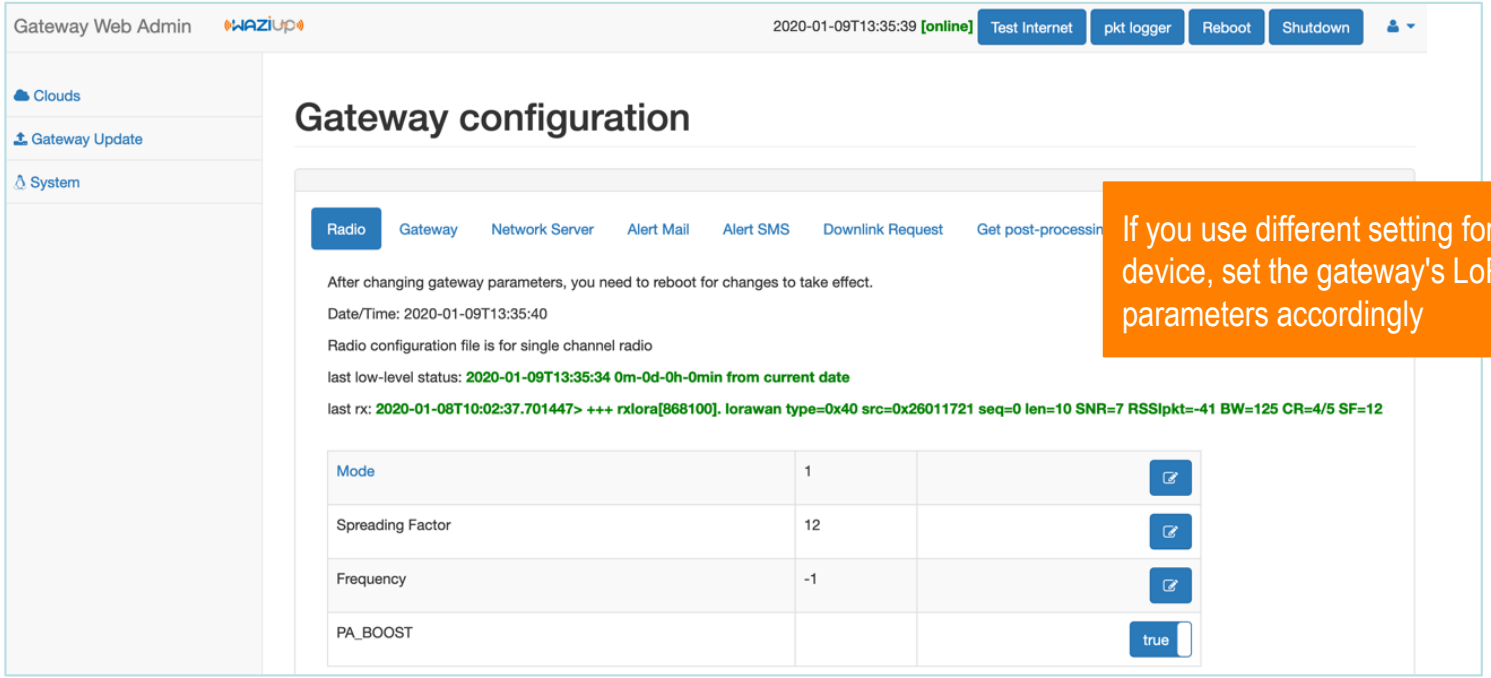

```
2018-08-14T23:06:12.579672> SX1276 detected, starting.
2018-08-14T23:06:12.579834> SX1276 LF/HF calibration
2018-08-14T23:06:12.579993> ...
2018-08-14T23:06:12.580151> **********Power ON: state 0
2018-08-14T23:06:12.580309> Default sync word: 0x12
2018-08-14T23:06:12.580465> LoRa mode 1
2018-08-14T23:06:12.580619> Setting mode: state 0
2018-08-14T23:06:12.580777> Channel CH_10_868: state 0
2018-08-14T23:06:12.580935> Set LoRa power dBm to 14
2018-08-14T23:06:12.581091> Power: state 0
2018-08-14T23:06:12.581245> Get Preamble Length: state 0
2018-08-14T23:06:12.581403> Preamble Length: 8
2018-08-14T23:06:12.581558> LoRa addr 1: state 0
2018-08-14T23:06:12.581715> SX1272/76 configured as LR-BS. Waiting RF input for transparent RF-serial bridge
2018-08-14T23:06:12.581884> Low-level gw status ON
```

```
Unrecognized transceiver
...
...
...
...
...
...
...
...
...
```

If you see something like this output then check how your radio module is connected to the Raspberry

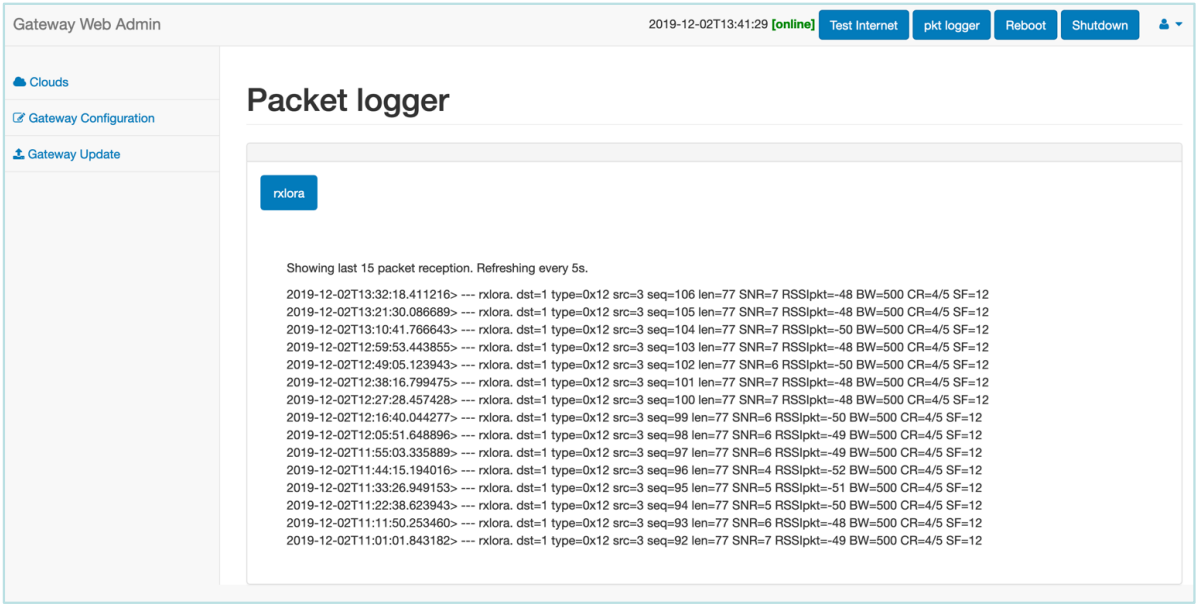
How can I check that LoRa parameters are the same for device and gateway?

- By default, all our examples use LoRa mode 1 (BW=125kHz, SF=12) with frequency of 865.2MHz (CH_10_868)
- By default, the gateway LoRa parameters are similar, -1 for frequency means default frequency, e.g. 865.2MHz



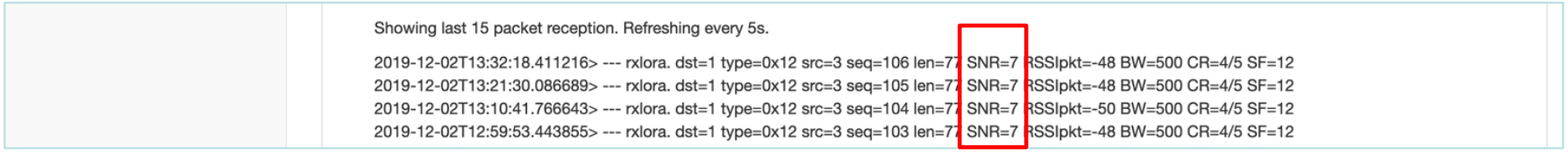
How can I know if gateway receives data or not?

- First, check that LoRa parameters for the device and gateway are the same
- Then, use the web admin interface to display the simple packet logger and switch ON a device
- You should see a new packet reception



What is PA_BOOST and how do I know PA_BOOST is set correctly?

- The Semtech SX1272/76 has actually 2 lines of RF power amplification (PA): a high efficiency PA up to 14dBm (RFO) and a high power PA up to 20dBm (PA_BOOST)
- Some radio modules only wire the PA_BOOST and not the RFO: RFM95 for instance has only PA_BOOST line
- If you are not sure, then check packet reception at gateway (simple packet logger) and if the SNR is negative at short range then it is most likely that the PABOOST setting must be inverted at the device side
- Here it is ok



How can I know if gateway pushes data to the cloud?

- First, check that the targeted cloud is enabled (either with the web interface or by editing `clouds.json`)
- Look at the gateway's log file and check that the cloud script is called and executed correctly
- Here is an exampe with the WAZIUP cloud

```
2018-08-17T16:33:16.652691> --- rxlora. dst=1 type=0x10 src=6 seq=34 len=10 SNR=8 RSSIpkt=-45 BW=125 CR=4/5 SF=12
2018-08-17T16:33:16.653027> 2018-08-17T16:33:16.650293
2018-08-17T16:33:16.653191> rcv ctrl pkt info (^p): 1,16,6,34,19,8,-45
2018-08-17T16:33:16.653353> splitted in:  [1, 16, 6, 34, 10, 8, -45]
2018-08-17T16:33:16.653513> (dst=1 type=0x10(DATA) src=6 seq=34 len=10 SNR=8 RSSI=-45)
2018-08-17T16:33:16.653676> rcv ctrl radio info (^r): 125,5,12
2018-08-17T16:33:16.653835> splitted in:  [125, 5, 12]
2018-08-17T16:33:16.653991> (BW=125 CR=5 SF=12)
2018-08-17T16:33:16.654144> rcv timestamp (^t): 2018-08-17T16:33:16.649
2018-08-17T16:33:16.654303>
2018-08-17T16:33:16.654452> got first framing byte
2018-08-17T16:33:16.654605> --> got LoRa data prefix
2018-08-17T16:33:16.654759> valid app key: accept data
2018-08-17T16:33:16.654914> number of enabled clouds is 1
2018-08-17T16:33:16.655069> --> cloud[0]
2018-08-17T16:33:16.655220> uploading with python CloudWAZIUP.py
2018-08-17T16:33:16.656730> WAZIUP: uploadingCloud
2018-08-17T16:33:16.656945> WAZIUP: will issue requests with
2018-08-17T16:33:16.657060> url: http://api.waziup.io/api/v1/domains/waziup-UPPA-
TESTS2/sensors/UPPA_Sensor6/measurements/TC/values
2018-08-17T16:33:16.657120> data: {"value":"24.60","timestamp":"2018-08-17T16:33:16.649"}
2018-08-17T16:33:16.657250> WAZIUP: returned msg from server is  200
2018-08-17T16:33:16.657420> WAZIUP: upload success
…
```

- Also, check on the cloud web page for your data

How can I activate local MongoDB storage?

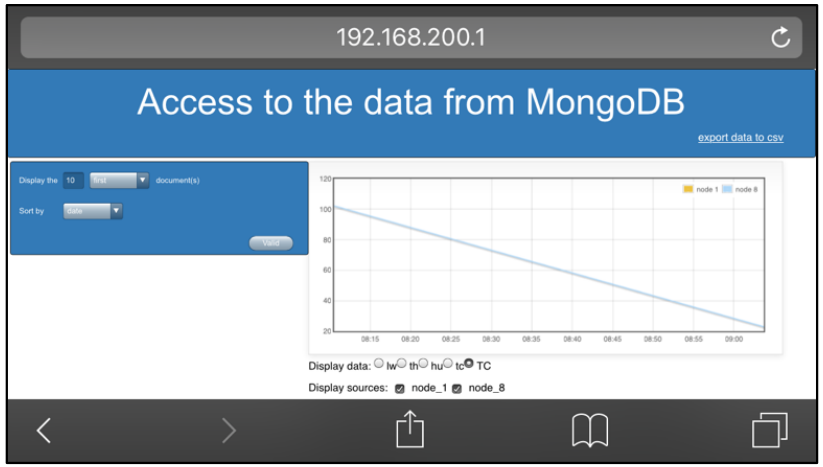
- Log in with `ssh` on your gateway, edit `clouds.json` and check that the local MongoDB cloud is enabled: set to true if necessary


```
{
  "clouds": [
    {
      "name": "Local gateway MongoDB",
      "notice": "do not remove the MongoDB cloud declaration, just change en
      "script": "python CloudMongoDB.py",
      "type": "database",
      "max_months_to_store": 2,
      "enabled": false
    },
```


- If you changed the setting, reboot your gateway for changes to take effect

What if the data web interface is not showing anything?

- When connected to the gateway's WiFi, opening http://192.168.200.1 will display the data web interface where data stored in the local MongoDB database are displayed
- If the graph section is displayed but your data is not displayed, check that local MongoDB is enabled
- If the graph section is not displayed, you need to repair the MongoDB database
  - Log in with `ssh` on your gateway, quit the text command interface if needed (option Q)
  - Go into `scripts` folder: `cd scripts`
  - Run `./mongo_repair.sh` script
  - Reload the data web interface



Further readings

- Specific README files on the github, especially those on cloud management and encryption
  - https://github.com/CongducPham/LowCostLoRaGw/blob/master/gw_full_latest/README-NewCloud.md
  - https://github.com/CongducPham/LowCostLoRaGw/blob/master/gw_full_latest/README-aes_lorawan.md