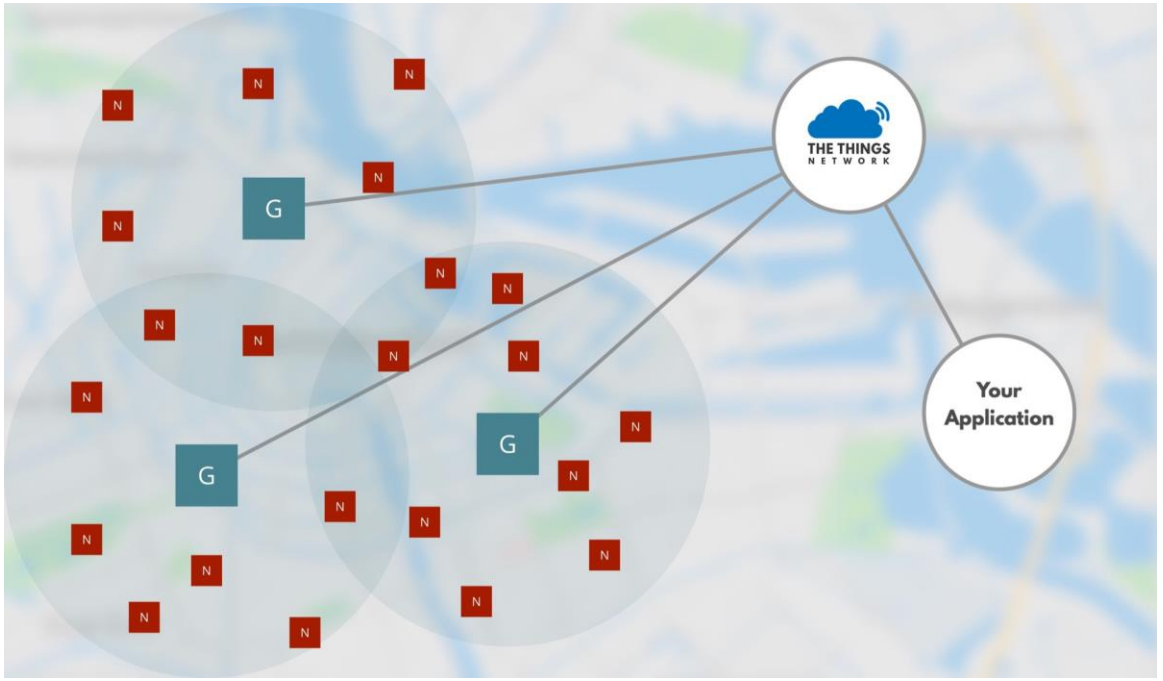
LoRaWan Register a device on TTN in Vietnam

F. Ferrero

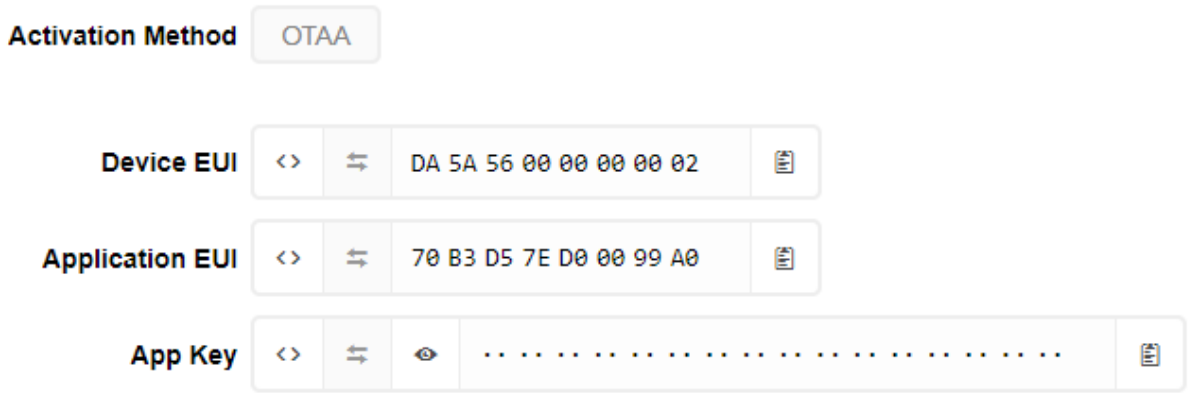
LoRaWan with The Thing Network

- The Things Network is a global, open, crowd-sourced Internet of Things data network.
- **The Things Network Backend** route messages from Nodes to the right Application, and back
- First, you have to register to https://www.thethingsnetwork.org/ , when it is done, tell me your ID, I will add you as a collaborator on the Da Nang application
- You can also join the Da Nang TTN community :



Adding a new device

- Just after receiving the autorisation for the application
- Go to « da_nang_tutorial » application and register device
- For ID and EUI, use the N° da5a56YYY00000XX and just incremant XX.
- YYY is your campus ID : DUT is D07, UTE is 07E and CIT is C17.
- To remember it : « da5a56 » is for « DANANG »
- It will provide Device EUI, Application EUI and App Key



Activation by Personalization (ABP)

- Go to settings
- Select ABP and save
- Go back to Overview
- You have now the Device Address and the two 128 AES keys
- You can click on Hex-C Style to have the key in the right format

Activation by Personalization (ABP)

- Go to my Github : https://github.com/FabienFerrero/UCA_Board
- Download the archive (.zip) and extract the archive
- Copy the file from Arduino_Code/Libraries/ to /Document/Arduino/Libraries/
- Open the code Arduino_Code/LORAWAN/ABP/Basic/UCA-ABP_Basic.ino
- Copy/Paste NWKSKEY, APPSKEY and DEVADDR with your IDs from TTN

```
// LoRaWAN NwkSKey, network session key
// This is the default Semtech key, which is used by the early prototype TTN
// network.
static const PROGMEM u1_t NWKSKEY[16] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };


// LoRaWAN AppSKey, application session key
// This is the default Semtech key, which is used by the early prototype TTN
// network.
static const u1_t PROGMEM APPSKEY[16] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };


// LoRaWAN end-device address (DevAddr)

static const u4_t DEVADDR = 0x00000000;
```

Activation by Personalization (ABP)

- Select the 923MHz Vietnamese band (AS923)
- In /Document/Arduino/ Libraries/arduino-lmic-custom/src/lmic
- Edit config.h file
- Comment the #define CFG_eu868 1 line
- Uncomment the #define CFG_as923 1 line

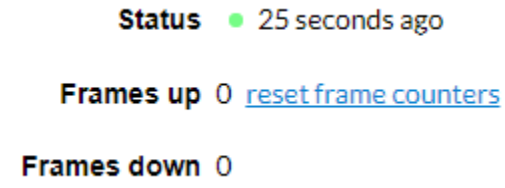
```
#ifndef _lmic_config_h_
#define _lmic_config_h_

// In the original LMIC code, these config values were defined on the
// gcc commandline. Since Arduino does not allow easily modifying the
// compiler commandline, use this file instead.

//#define CFG_eu868 1
//#define CFG_us915 1
#define CFG_as923 1
// This is the SX1272/SX1273 radio, which is also used on the HopeRF
// RFM92 boards.
//#define CFG_sx1272_radio 1
// This is the SX1276/SX1277/SX1278/SX1279 radio, which is also used on
// the HopeRF RFM95 boards.
#define CFG_sx1276_radio 1
```

Activation by Personalization (ABP)

- Compile and download the code on your board
- If it fails, install the [CP2102 usb driver]
- Look at the TTN device overview, status should be green
- Frames up should increment each 30s (« TX_INTERVAL »)
- Have look on Data
- For each uplink, you can look many details as RSSI, SNR, airtime, modulation, coding rate, GW ID, etc …
- Convert the payload from Hex to Text



time	counter	port		
21:45:35	3	1		payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
21:44:29	2	1		payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
21:43:22	1	1		payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
21:42:16	0	1	*retry*	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21

Activation by Personalization (ABP)

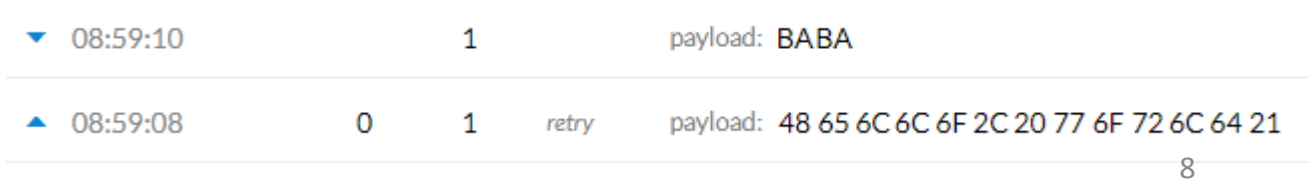
Frame counter security

- Now reset you board (click on the red button on the Arduino mini pro)
- TTN is no more receiving the data
- Click on « reset frame counters » and reset you board again
- As you can see, frame counter is a security features to avoid replay attack (done by capturing and re-transmitting the messages)
- Frame counter can be disabled for debug test in Settings

Downlink

- Open your serial monitor
- In TTN overview, go to downling, add a payload like « BABA » and click on send, and go to Data
- After the next uplink, you should see the number of byte received in downlink

```
Starting
Packet queued
150865: EV_TXCOMPLETE (includes waiting for RX windows)
Received
2
 bytes of payload
BABA
```



Activation by Personalization (ABP)

Change SF, power, payload …

- At the end of the arduino code, you have :

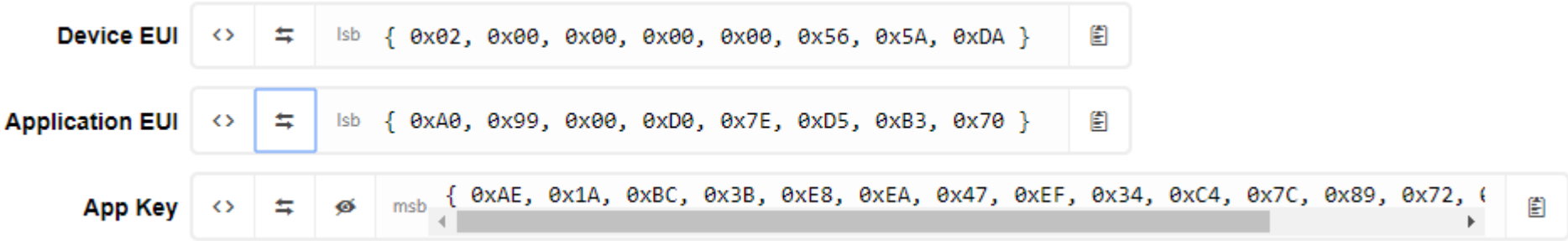
LMIC_setDrTxpow(DR_SF12,14);

- You can change SF from DR_SF7 to DR_SF12
- You can change the power from 2 dBm to 20 dBm
- Payload is in mydata[], and you can change the text.

- Do some test, what is the effect on the RSSI ?

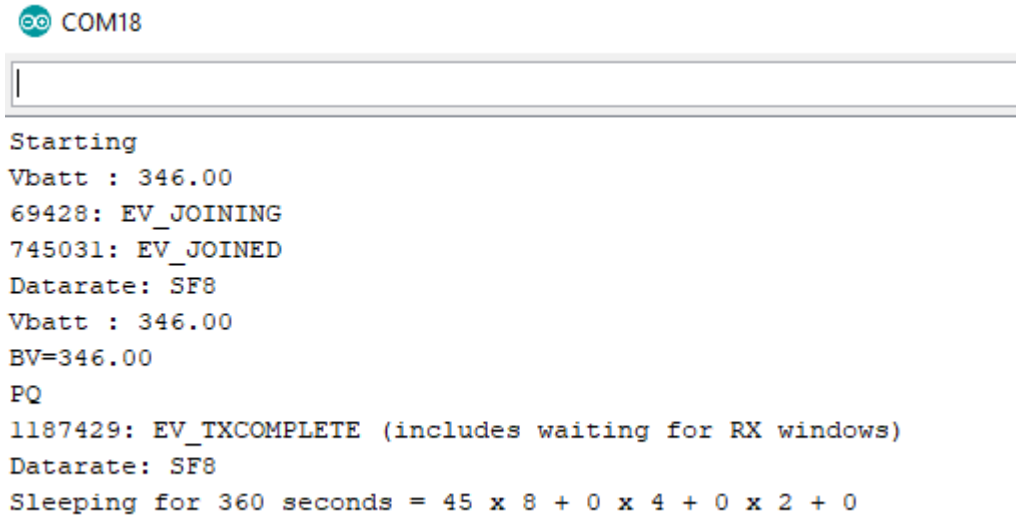
Over the Air Activation (OTAA)

- In TTN Settings of your device, select OTAA and save
- Open the code Arduino_Code/LORAWAN/OTAA/LP_Basic/UCA-OTAA_Basic.ino
- Copy paste after clicking on hexa-style the DEV-EUI, APP-EUI and App Key
- Be carefull !!!
  - Device EUI and Application EUI are **lsb**
  - App Key is **msb**



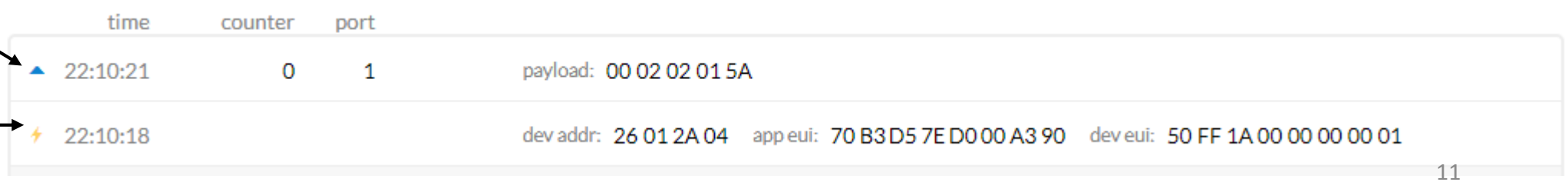
Over the Air Activation (OTAA)

- Look in data
- You should see a first uplink that request the connection
- And a second packet witht the first data
- On the serial monitor you can see the Joining process and then Joined and Tx.
- The device go to sleep after the Tx



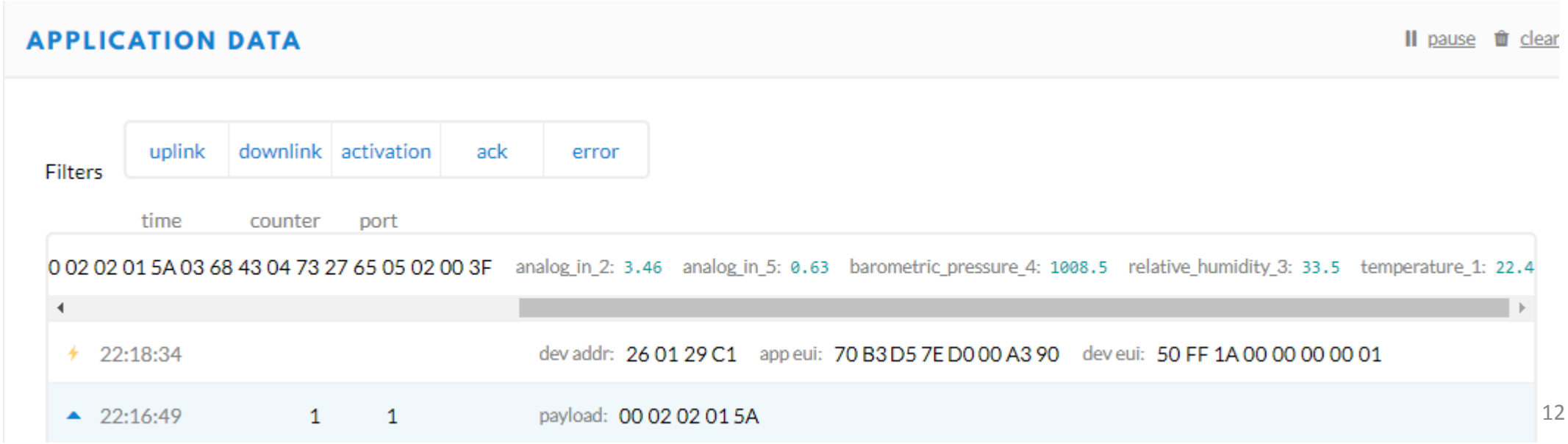
First packet with data

Board request for connection



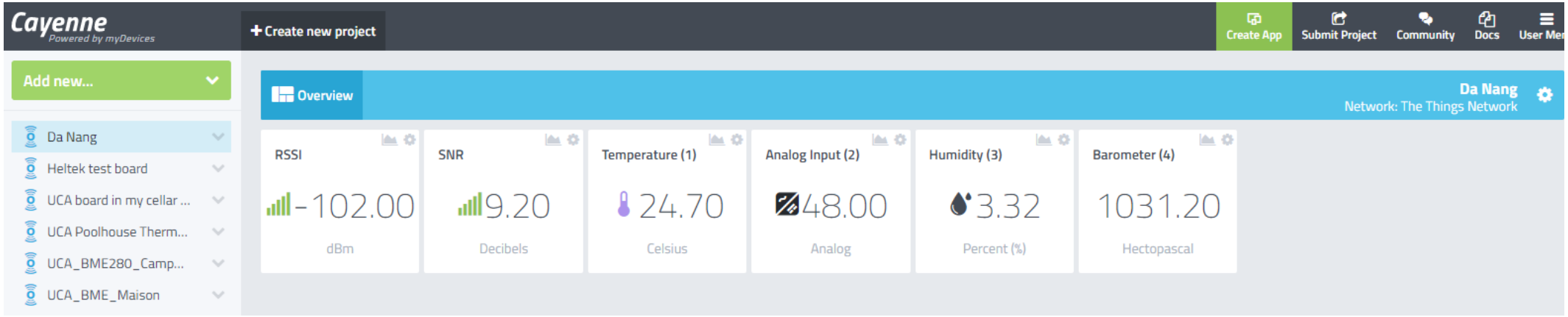
Over the Air Activation (OTAA) and data

- Try now the code Arduino_Code/LORAWAN/OTAA/LP_BME280/UCA-BME280.ino
- It use the sensor BME280 that measure T°c, Humidity and Pressure
- The code is using Cayenne LPP format
- Now you can see sensor data in the uplink packet



Using Cayenne to see you data

- Go to https://mydevices.com/ and create an account
- Add a device by selecting LoRa/TheThingNetwork and Cayenne LPP.
- Just add your device EUI
- You should see your data



Downlink with LoRaWAN

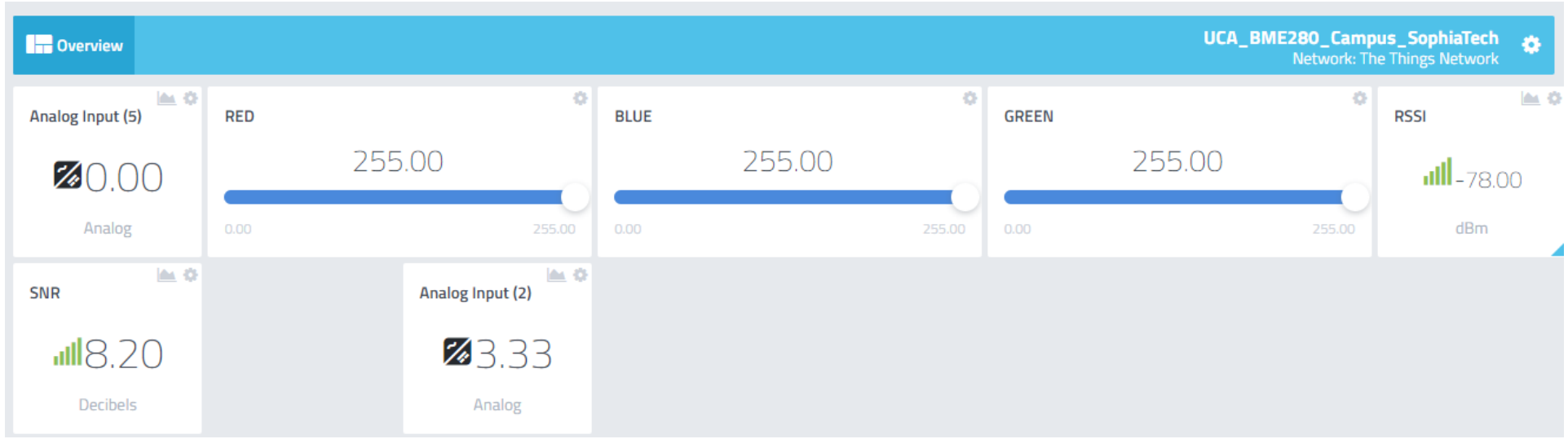
- You are not going to control the color of a LED from CAYENNE.
- Use the code in : https://github.com/FabienFerrero/UCA_Board/tree/master/Arduino_Code/LORAWAN/OTAA/LED_CONTROLLER

Downlink with LoRaWAN

- You are now going to control the color of a LED from CAYENNE.
- Use the code in : https://github.com/FabienFerrero/UCA_Board/tree/master/Arduino_Code/LORAWAN/OTAA/LED_CONTROLLER

Downlink with LoRaWAN

- Upload the code
- It will send uplink all 15 seconds
- The LED are controlled by a PWM with 255 states
- Change in your Cayenne Dashboard to have the slider from 0 to 255

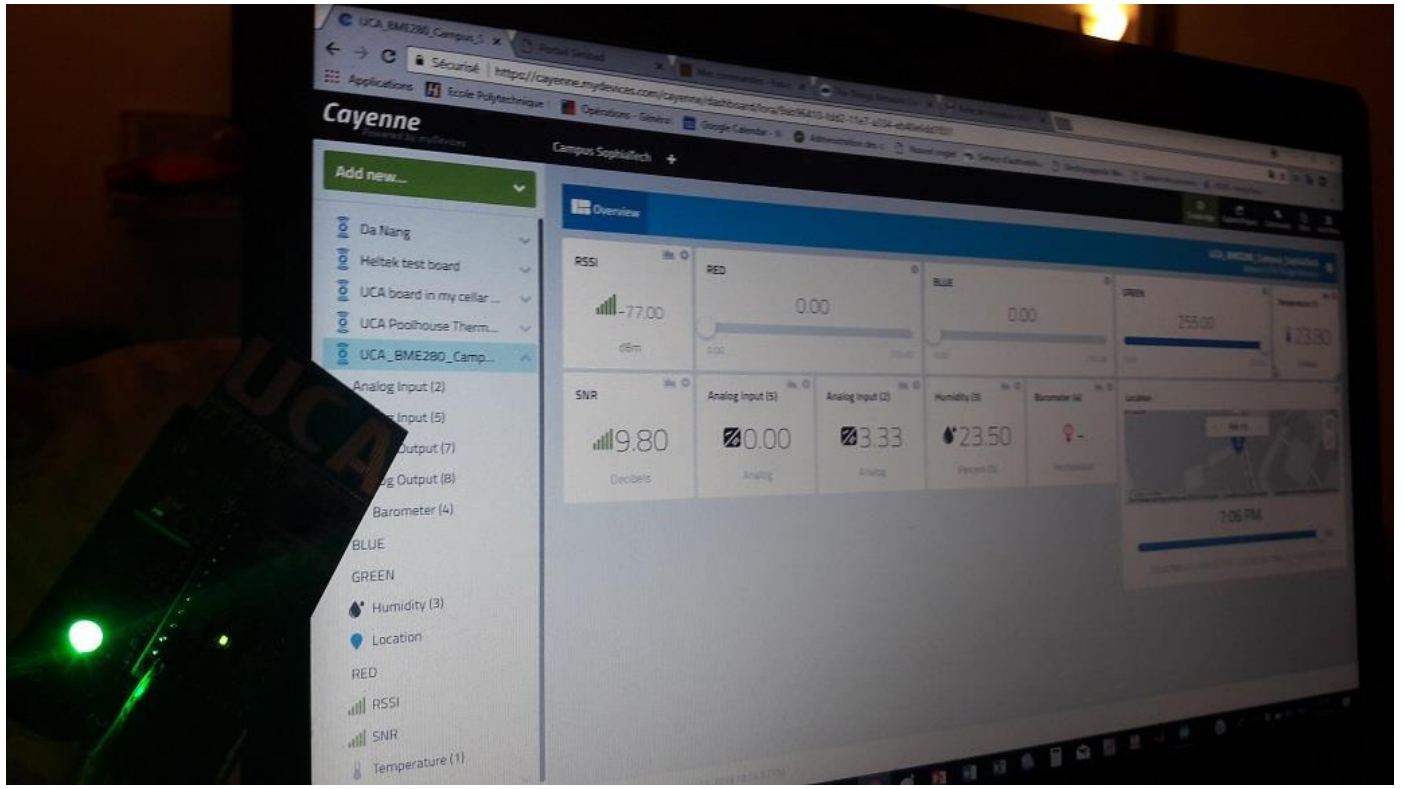


Downlink with LoRaWAN

- When you change the slider value, look at your data in TTN
- You should see that the downlink is scheduled
- Then it is confirmed and the node send an ack

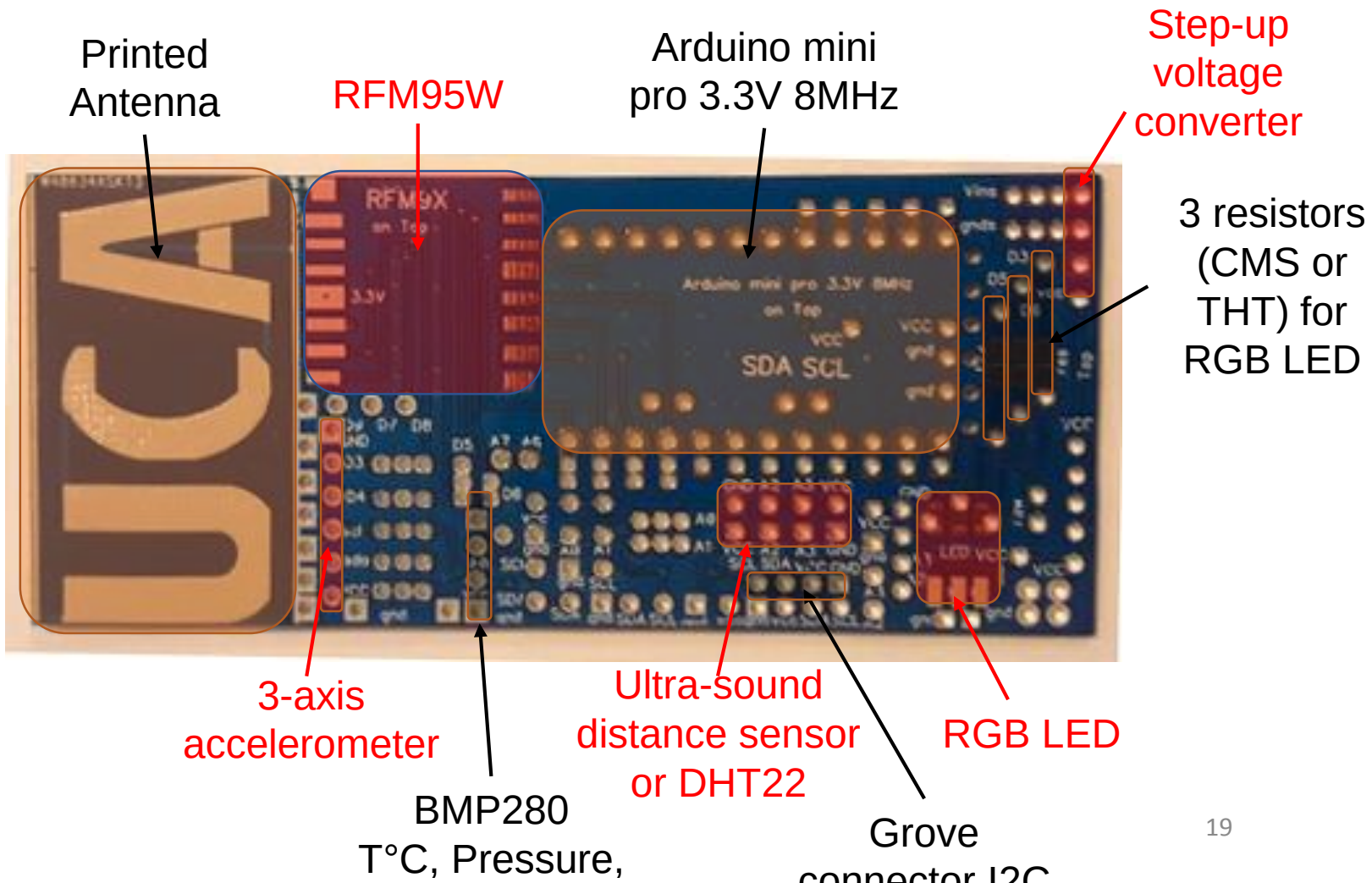
Downlink with LoRaWAN

- After each uplink, the node open a received window for downlink
- With Cayenne, only one color can be updated at the same time



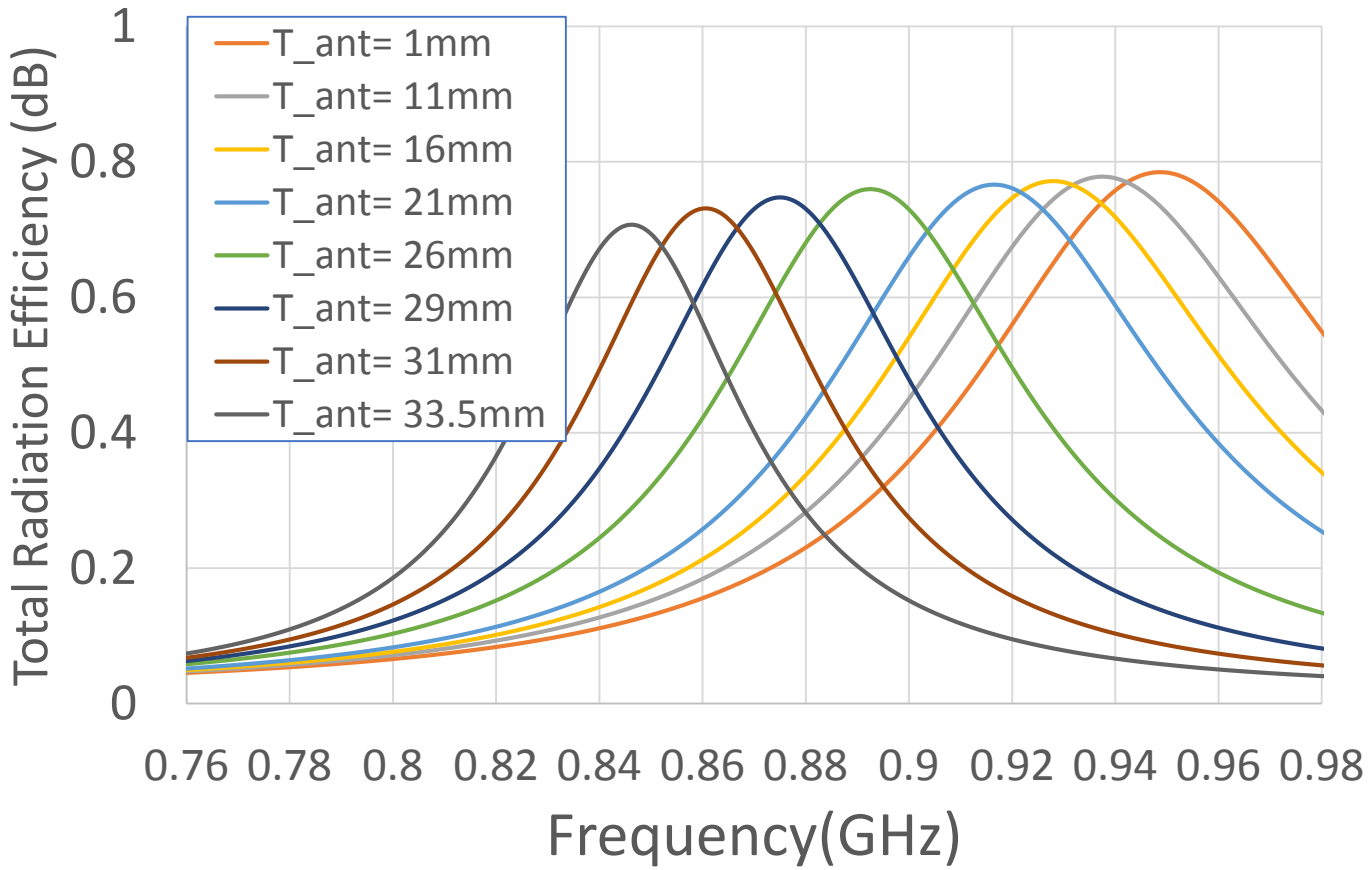
Use and Tune your board

- UCA PCB is provided to speed your developments
- Antenna as to be tuned for 923MHz !

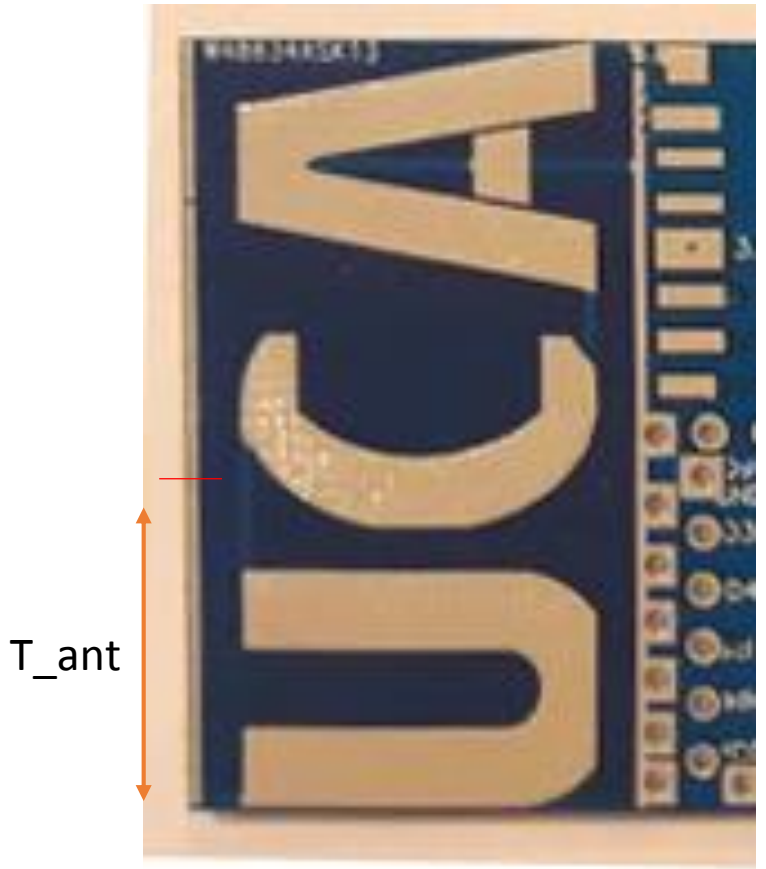


Use and Tune your board

- Antenna as to be tuned for 923MHz !
- More info on : http://github/fabien.ferrero/UCA



Printed Antenna



Good luck for you projects !