

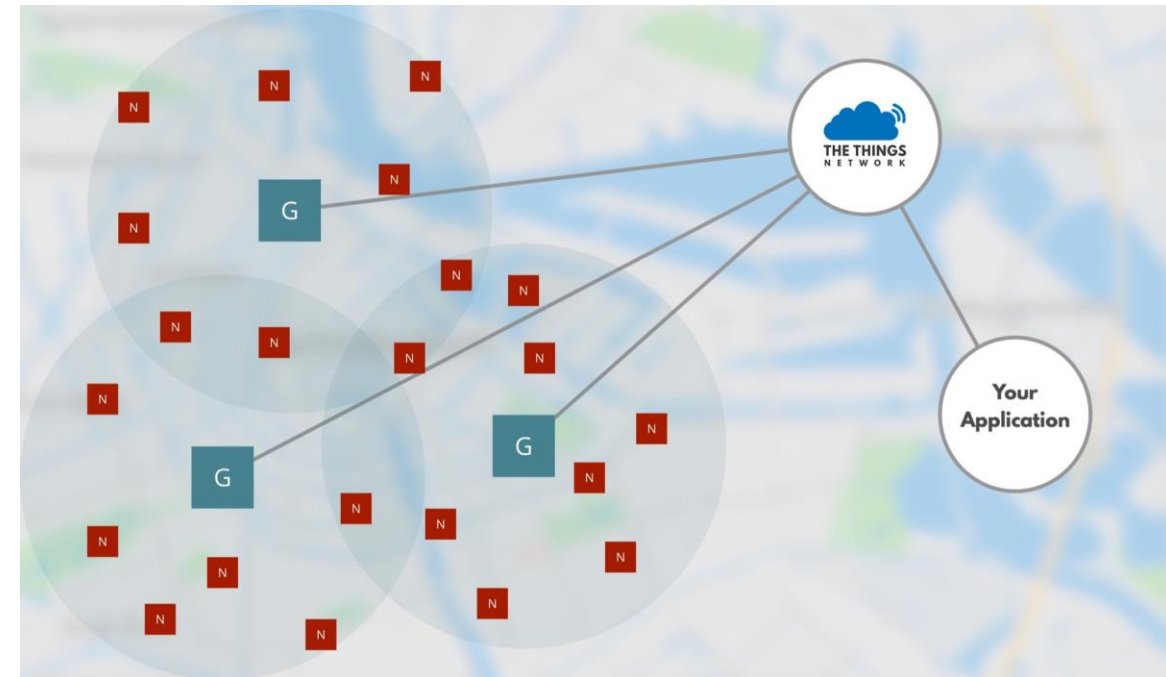
LoRaWan Register a device on TTN in Vietnam

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 increment 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 Method

Device EUI DA 5A 56 00 00 00 00 02

Application EUI 70 B3 D5 7E D0 00 99 A0

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 Method **ABP**

Device EUI

Application EUI

Device Address

Network Session Key

App Session Key

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 NWKSKY, APPSKY 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 ul_t NWKSKY[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 ul_t PROGMEM APPSKY[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

Status ● 25 seconds ago

Frames up 0 [reset frame counters](#)

Frames down 0

	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
```

▼ 08:59:10	1		payload: BABA
▲ 08:59:08	0	1	retry payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21

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**

Device EUI	<>	⇄	lsb	{ 0x02, 0x00, 0x00, 0x00, 0x00, 0x56, 0x5A, 0xDA }	📋
Application EUI	<>	⇄	lsb	{ 0xA0, 0x99, 0x00, 0xD0, 0x7E, 0xD5, 0xB3, 0x70 }	📋
App Key	<>	⇄	👁	msb { 0xAE, 0x1A, 0xBC, 0x3B, 0xE8, 0xEA, 0x47, 0xEF, 0x34, 0xC4, 0x7C, 0x89, 0x72, 0x...	📋

Over the Air Activation (OTAA)

- Look in data
- You should see a first uplink that request the connection
- And a second packet with 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

COM18

```
Starting
Vbatt : 346.00
69428: EV_JOINING
745031: EV_JOINED
Datarate: SF8
Vbatt : 346.00
BV=346.00
PQ
1187429: EV_TXCOMPLETE (includes waiting for RX windows)
Datarate: SF8
Sleeping for 360 seconds = 45 x 8 + 0 x 4 + 0 x 2 + 0
```

First packet
with data

Board
request for
connection

	time	counter	port	
First packet with data	▲ 22:10:21	0	1	payload: 00 02 02 01 5A
Board request for connection	⚡ 22:10:18			dev addr: 26 01 2A 04 appeui: 70 B3 D5 7E D0 00 A3 90 dev eui: 50 FF 1A 00 00 00 00 01

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

The screenshot displays the 'APPLICATION DATA' section of a Cayenne LPP format interface. At the top, there are controls for 'pause' and 'clear'. Below this, a 'Filters' section contains buttons for 'uplink', 'downlink', 'activation', 'ack', and 'error'. The main data area shows a table with columns for 'time', 'counter', and 'port'. The first row of data shows a timestamp of '22:18:34' with a lightning bolt icon, and a payload of '00 02 02 01 5A'. The second row shows a timestamp of '22:16:49' with an upward arrow icon, a counter of '1', a port of '1', and a payload of '00 02 02 01 5A'. The payload is expanded to show sensor data: 'analog_in_2: 3.46', 'analog_in_5: 0.63', 'barometric_pressure_4: 1008.5', 'relative_humidity_3: 33.5', and 'temperature_1: 22.4'. The device address is '26 01 29 C1', the application EUI is '70 B3D5 7ED000 A3 90', and the device EUI is '50 FF 1A 00 00 00 00 01'.

time	counter	port
22:18:34		
22:16:49	1	1

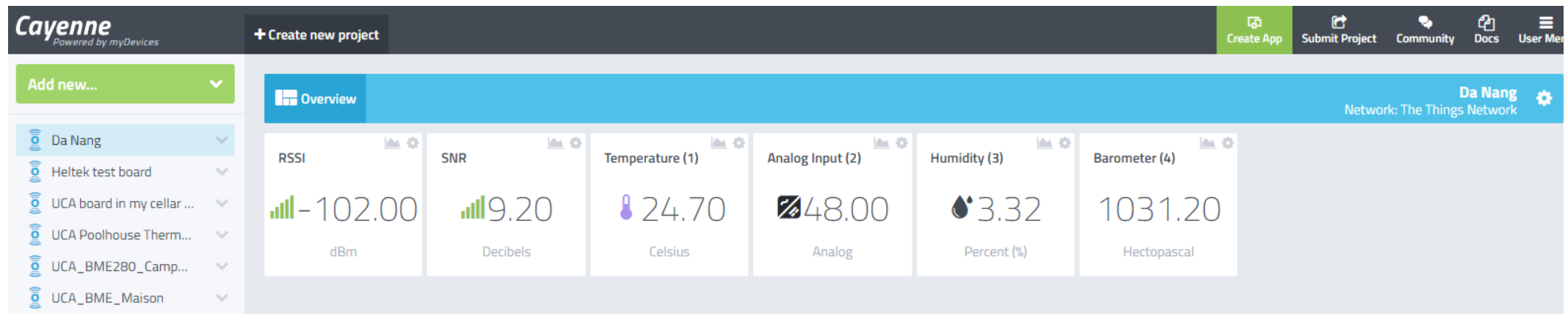
payload: 00 02 02 01 5A

analog_in_2: 3.46 analog_in_5: 0.63 barometric_pressure_4: 1008.5 relative_humidity_3: 33.5 temperature_1: 22.4

dev addr: 26 01 29 C1 app eui: 70 B3D5 7ED000 A3 90 dev eui: 50 FF 1A 00 00 00 00 01

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

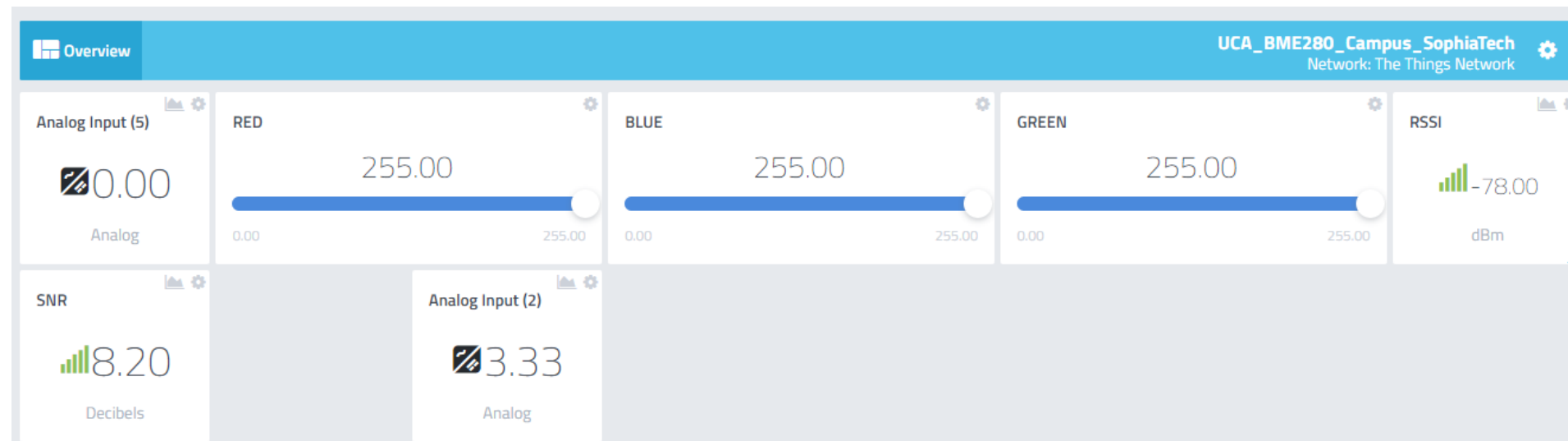
Downlink with LoRaWAN

- 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

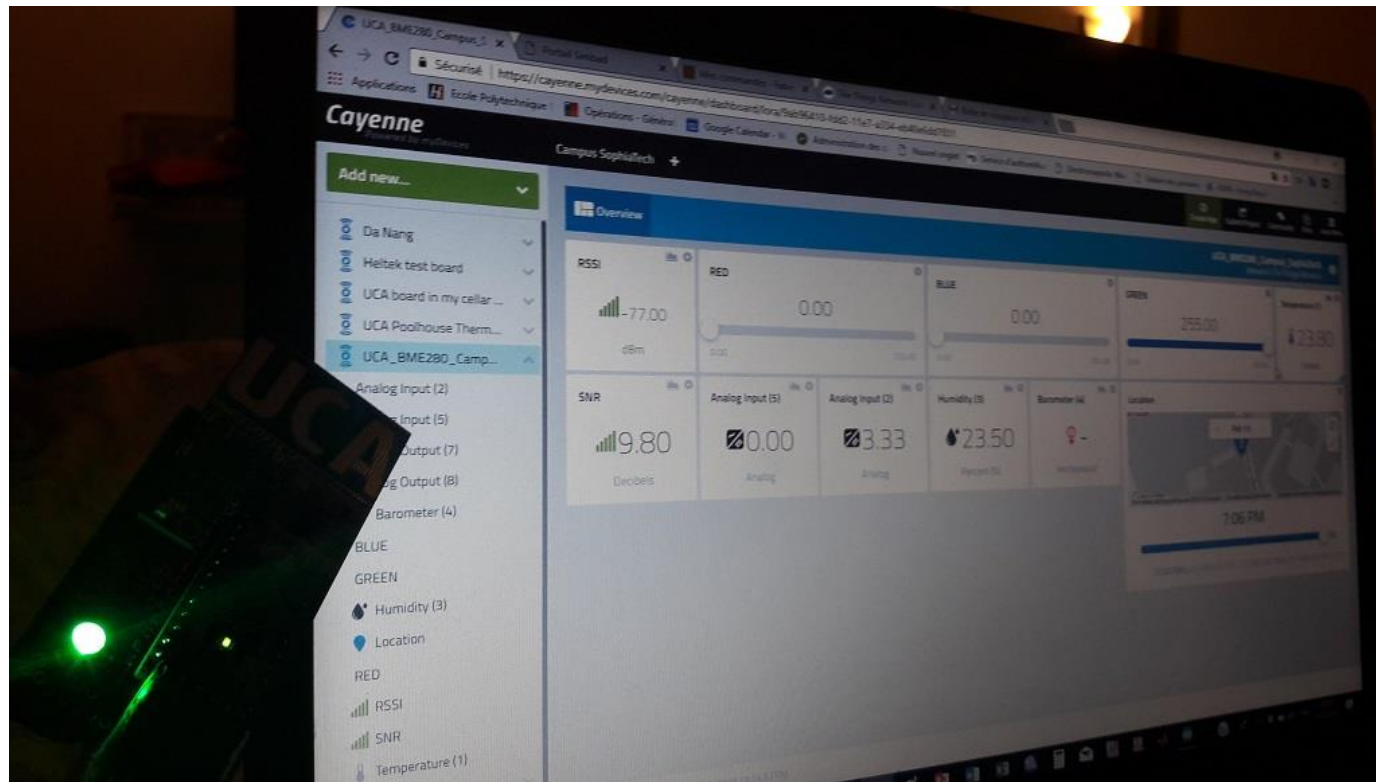
Filters

	uplink	downlink	activation	ack	error
--	--------	----------	------------	-----	-------

	time	counter	port	
▲	14:13:28	8	1	payload: 02 02 01 51 06 03 00 00 07 03 00 FF 08 03 00 00 analog_in_2: 3.37 analog_out_6: 0 analog_out_
▼	14:13:28		99	confirmed ack app id: campus_sophiatech
▼	14:13:16		99	confirmed payload: 07 63 9C FF
▲	14:13:15	7	1	payload: 02 02 01 52 06 03 00 00 07 03 00 00 08 03 00 00 analog_in_2: 3.38 analog_out_6: 0 analog_out_
▼	14:13:11		99	scheduled confirmed payload: 07 63 9C FF
▲	14:13:03	6	1	payload: 02 02 01 51 06 03 00 00 07 03 00 00 08 03 00 00 analog_in_2: 3.37 analog_out_6: 0 analog_out_

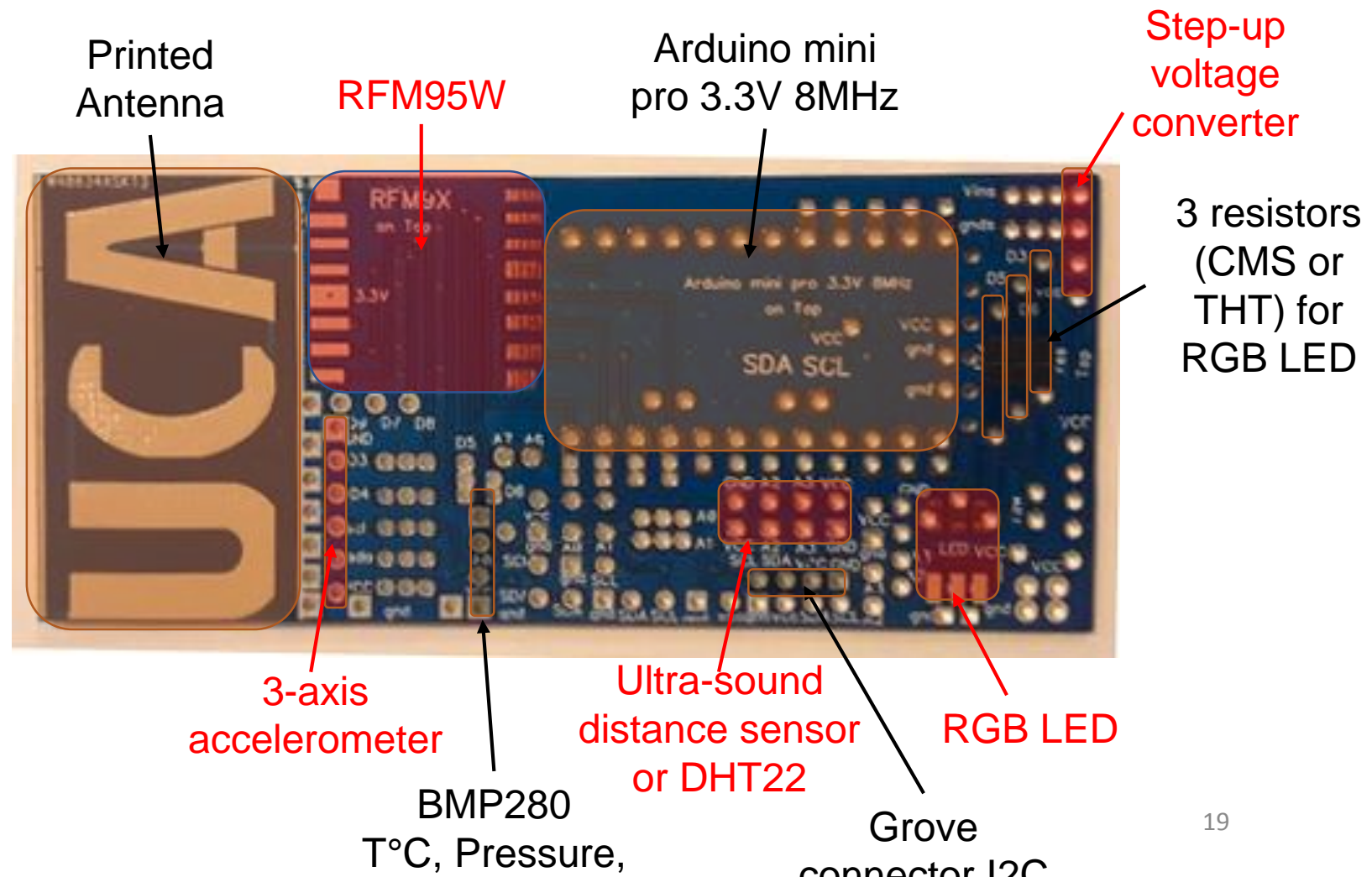
Downlink with LoRaWAN

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



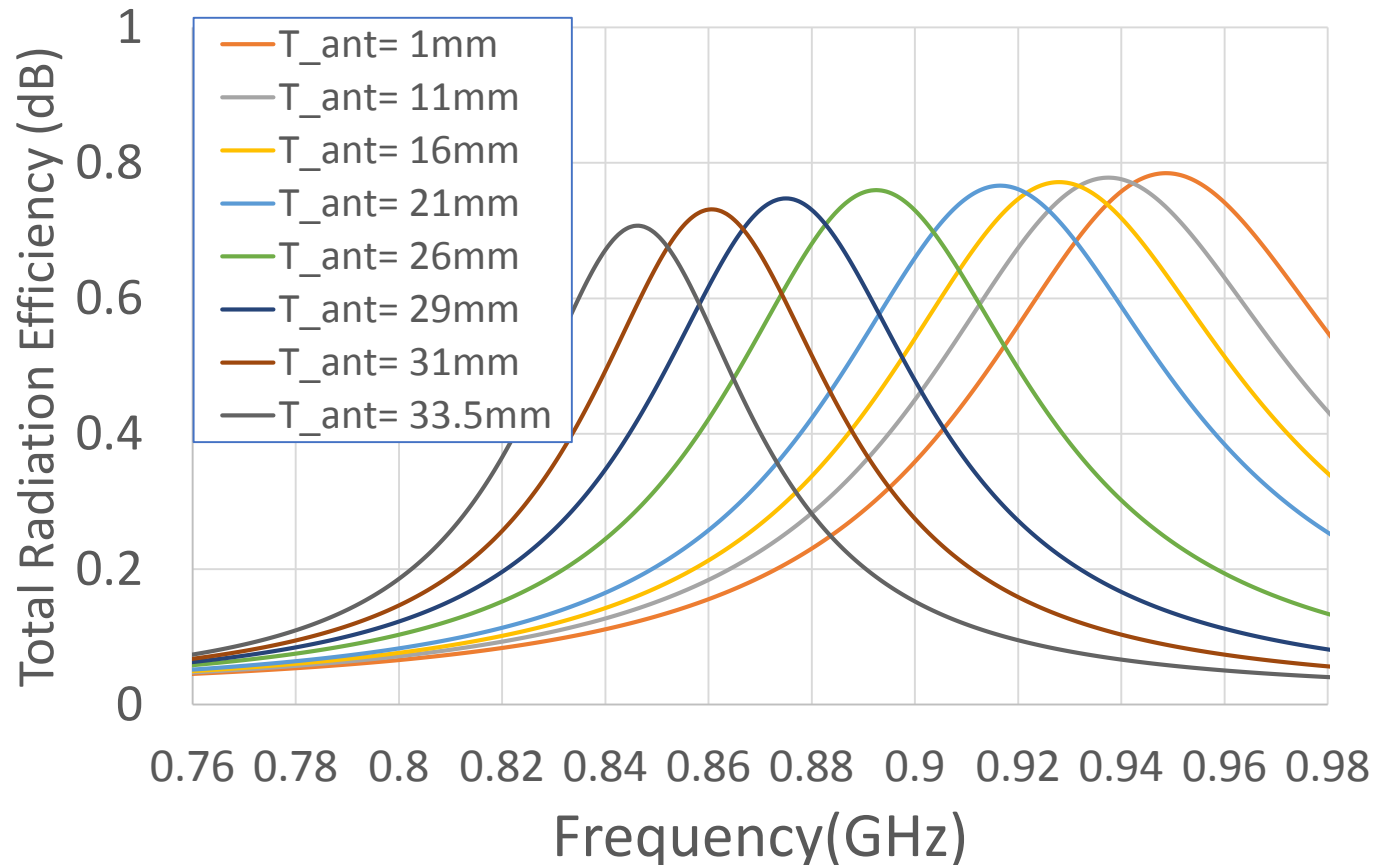
Use and Tune your board

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

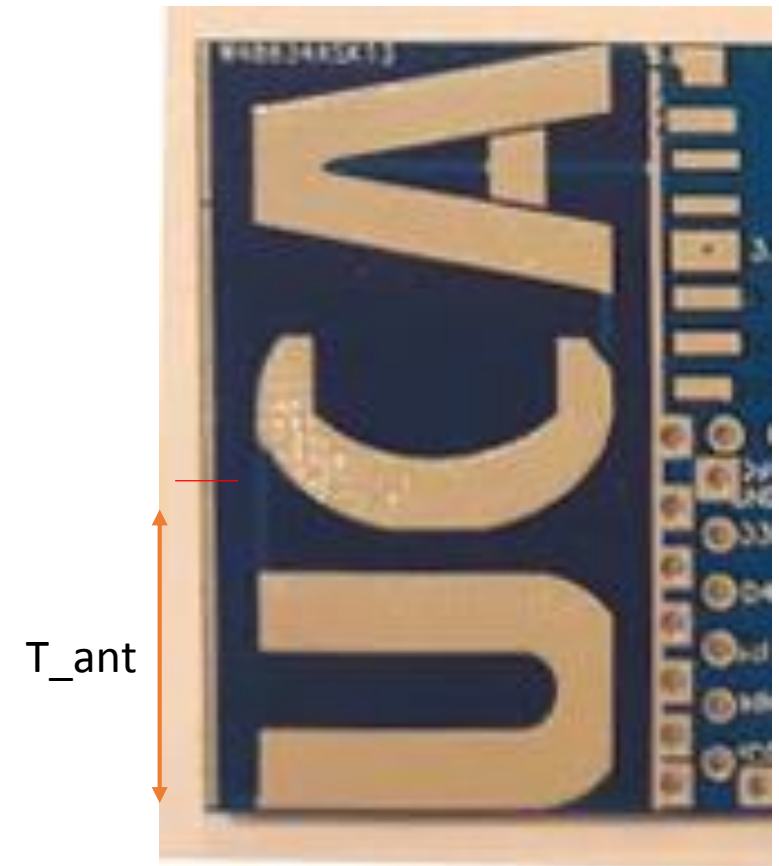


Use and Tune your board

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



Printed
Antenna



Good luck for you projects !

