



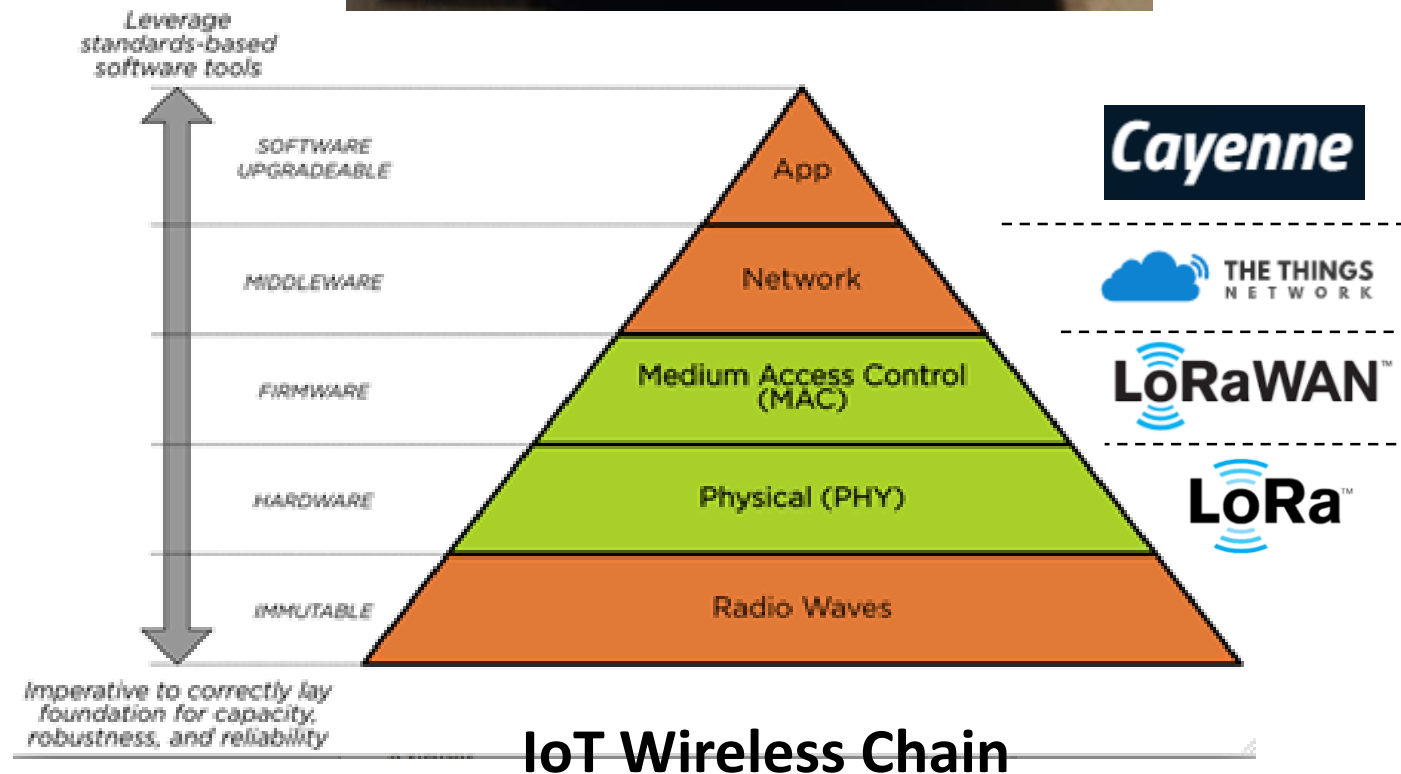
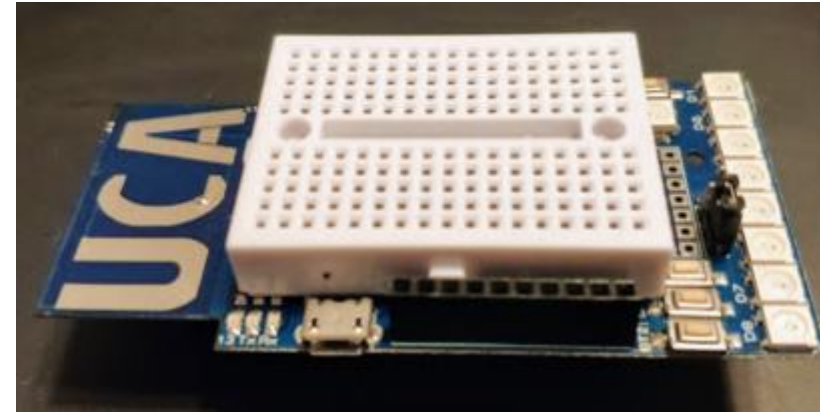
# LoRaWan tutorial : Register a device on TTN

F. Ferrero, Professor @UCA

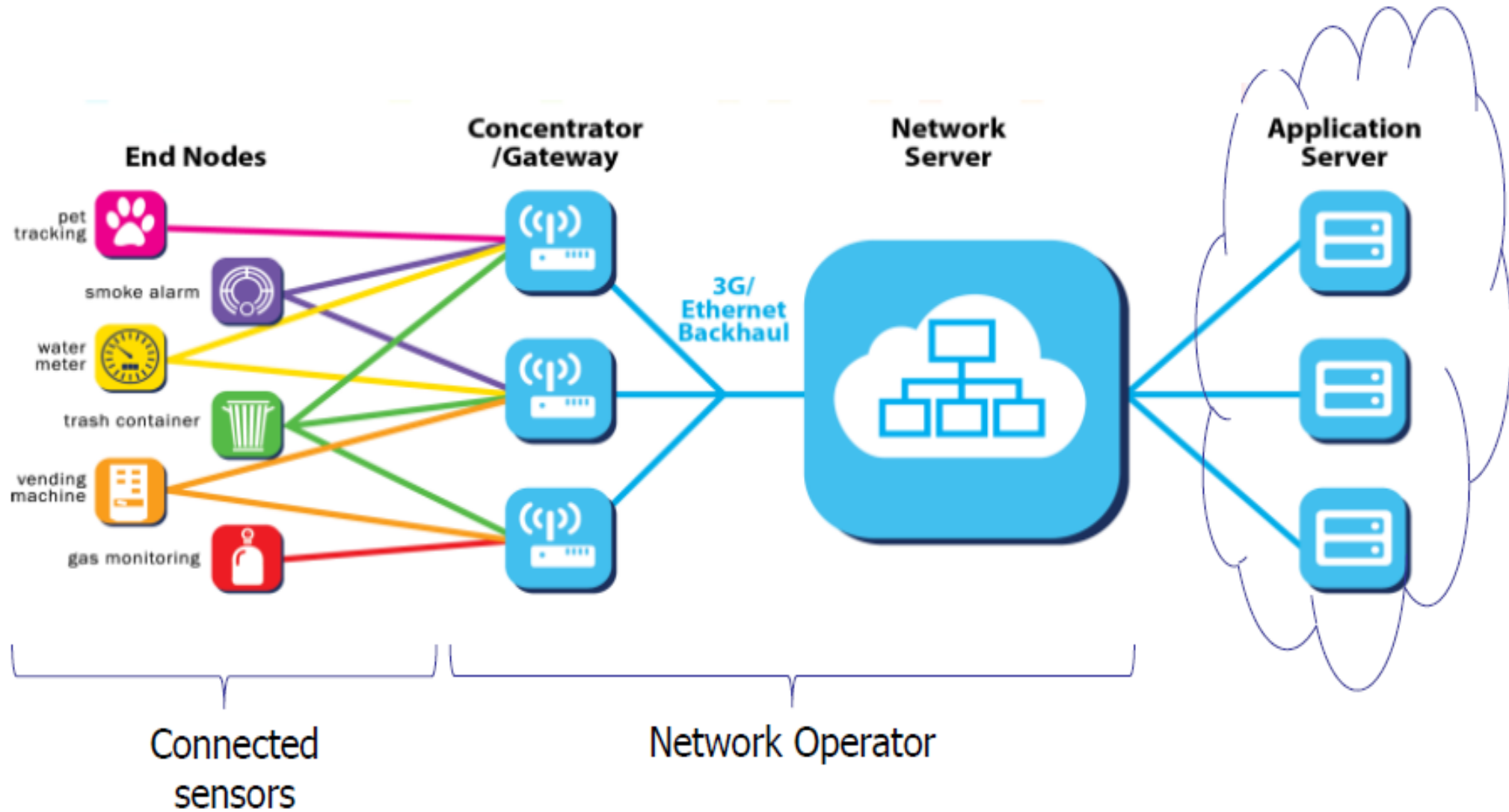
# LoRaWan Tutorial Objectives

In this tutorial, you will :

- Use the UCA Education Board
- Program a microcontroller in C with Arduino IDE
- Register the board to a network server
- Transmit data with LoRa modulation
- Push data to an application server



# LoRaWan Tutorial Objectives



# Downloading Arduino code on Github

- For this tutorial, you are going to use Arduino codes
- Codes are available on :  
[https://github.com/FabienFerrero/UCA Education Board](https://github.com/FabienFerrero/UCA_Education_Board)
- You can click on “Clone or Dowload” and “download zip”
- Then unzip it
- If you are using Github Desktop, you can use “open in Desktop”

Board for Education with LoRa

12 commits   1 branch   0 releases   1 contributor   MIT

Branch: master   New pull request   Find File   Clone or download

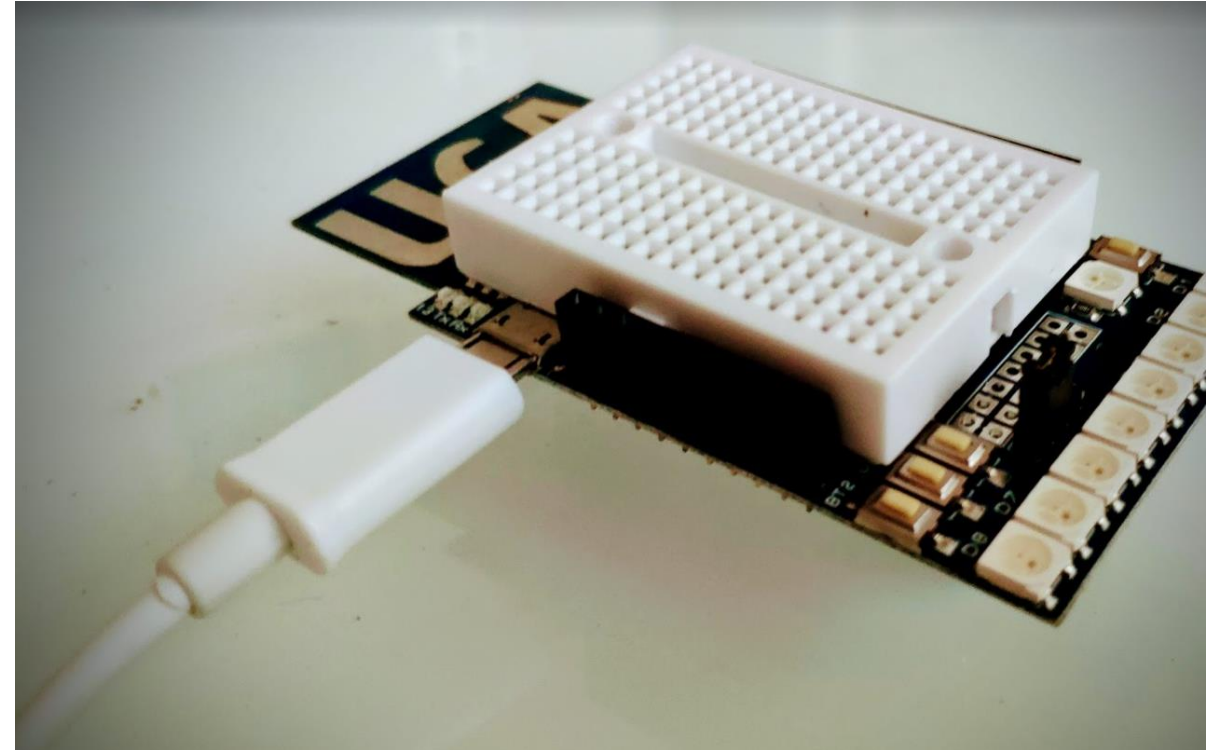
FabienFerrero add code		Latest commit ad15230 2 days ago
Antenna	code update	last month
Code	add code	2 days ago
Libraries	LED update	last month
Schematic	code update	last month
LICENSE	Initial commit	last month
README.md	Update README.md	last month

# Configuring your Arduino IDE

- After downloading the archive (.zip) and extracting the archive
- Copy the file from UCA\_Education\_Board\Libraries to /Document/Arduino/ Libraries/

It will install the libs needed during the tutorial

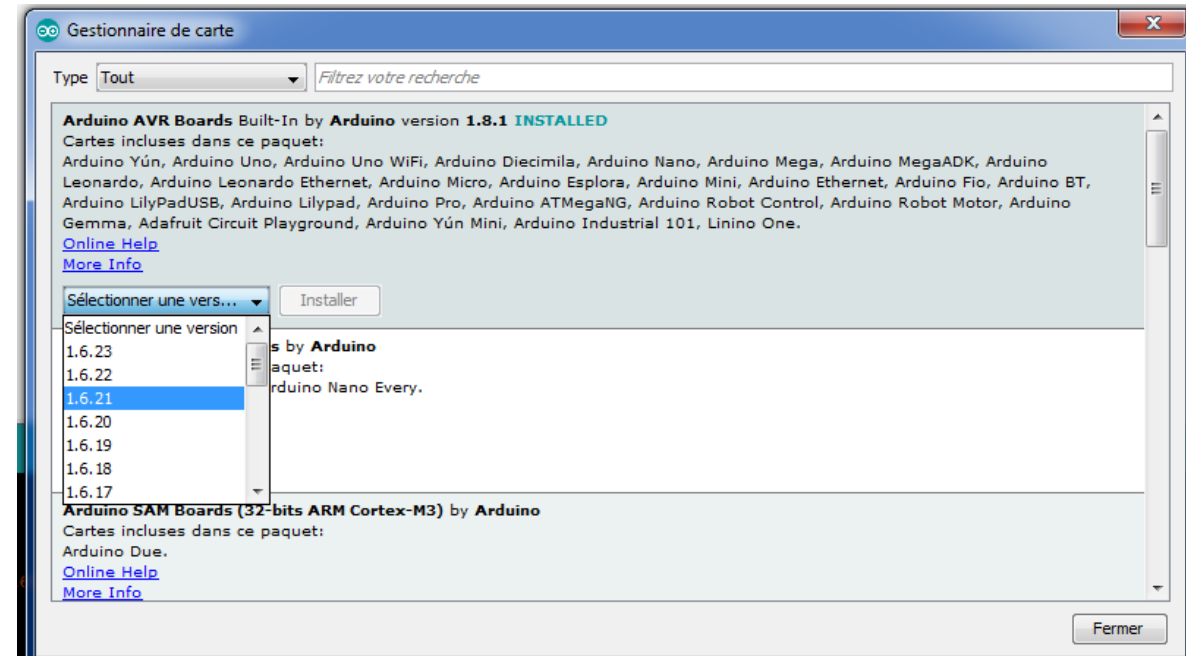
- If your using Windows or Mac, your may need to install the board USB driver (CH340C) : drivers are available [here](#)
- If Arduino IDE select in Tools (Outils)
  - Board : Arduino Pro or Pro Mini
  - Processor : AT328p 8MHz 3.3V
  - Port : Select your serial port





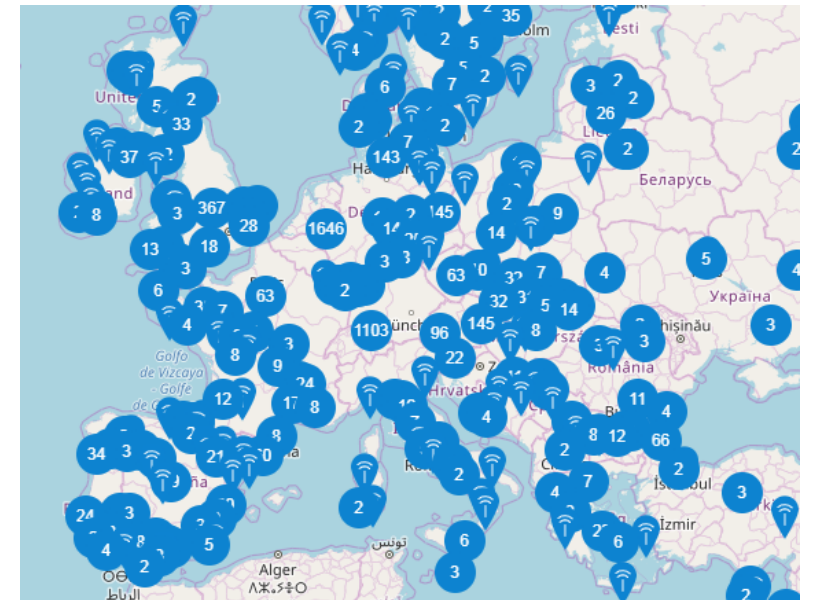
# Configuring your Arduino IDE

- **!Important!** We have an unsolved bug in Arduino
- To avoid it, you have to go to:
  - Tools\Board\Boards Manager
  - In Arduino AVR Boards, select **1.6.21** version and **Install**
  - You will probably need to close and re-open Arduino IDE



# 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
- TTN is free
- 10000 LoRa gateways are connected to TTN around the world
- Any TTN can use any GWs, it is a collaborative network




# Create a TTN account

- First, you have to [register](https://www.thethingsnetwork.org/) to <https://www.thethingsnetwork.org/>
- Then, give me your USERNAME, I will add you as a collaborator in our application
- You can also join a local community :

[TTN Côte d'Azur](#)


[TTN Da Nang](#)




## CREATE AN ACCOUNT

Create an account for The Things Network and start exploring the world of Internet of Things with us.


**USERNAME**  
This will be your username — pick a good one because you will **not** be able to change it.



**EMAIL ADDRESS**  
You will receive a confirmation email, as well as occasional account related emails. If this email address is managed by a third party (such as for corporate email addresses), this third party might block emails coming from The Things Network. This email address is not public.



**PASSWORD**  
Use at least 6 characters.



**NEWSLETTER**  
Subscribe to the newsletter. ☐

Create account



# Adding a new device

- Go to « application » and choose the available application
- Click on « register device »
- For ID and EUI, use the N° 50ff1aDA5A560XX and just increment XX.
- To remember it : « 50ff1a » is for « SOPHIA » and « DA5A56 » is for DaNang
- In DEVICE OVERVIEW, you get useful information on your device. Of course, status is : « Never Seen »
- First we will connect using ABP (Activation by Personalization )

**REGISTER DEVICE**[bulk import devices](#)

**Device ID**  
This is the unique identifier for the device in this app. The device ID will be immutable.

**Device EUI**  
The device EUI is the unique identifier for this device on the network. You can change the EUI later.  
 8 bytes

**App Key**  
The App Key will be used to secure the communication between you device and the network.

**App EUI**

**DEVICE OVERVIEW**

Application ID dniit

Device ID test

Activation Method OTAA

Device EUI

Application EUI

App Key

Status never seen


Frames up 0 [reset frame counters](#)

Frames down 0

# 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 **<>** **↕** 50 FF 1A 00 00 00 00 01 

Application EUI **<>** **↕** 70 B3 D5 7E D0 00 A3 90 

Device Address **<>** **↕** 26 01 11 44 

Network Session Key **<>** **↕**  msb { 0x14, 0x46, 0xEF, 0x00, 0x4B, 0xC9, 0x96, 0x95, 0xFE, 0x6A, 0x6F, 0x9F, 0xAC, ( 

App Session Key **<>** **↕**  msb { 0x18, 0xF8, 0xAA, 0x61, 0x31, 0x2C, 0x6C, 0x05, 0x66, 0x4F, 0xCB, 0xE1, 0x81, ( 

# Activation by Personalization (ABP)

- Open the code UCA\_Education\_Board\Code\LORAWAN\ABP\Basic\UCA-ABP\_Basic\UCA-ABP\_Basic.ino
- Copy/Paste DEVADDR from your TTN window with « 0x » for Hex style
- Copy/Paste NWKSKEY and APPSKEY using C-style from your TTN window

[illegible]

# Activation by Personalization (ABP)

- Compile and download the code on your board
- Look at the TTN device overview
- Frames up should increment each half minute as your board is sending an uplink 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 ...
- Click on the blue triangle

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 right button on your board)
- 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 can find :

`LMIC_setDrTxpow(DR_SF12,14);`

- You can change Spreading Factor(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.
- You can convert the payload in Hex to normal text using this online [tool](#)
- Change SF and Payload text ! What is the effect on Time on Air ?

# Over the Air Activation (OTAA)

- In TTN Settings of your device, select OTAA and save
- Open the code UCA\_Education\_Board\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

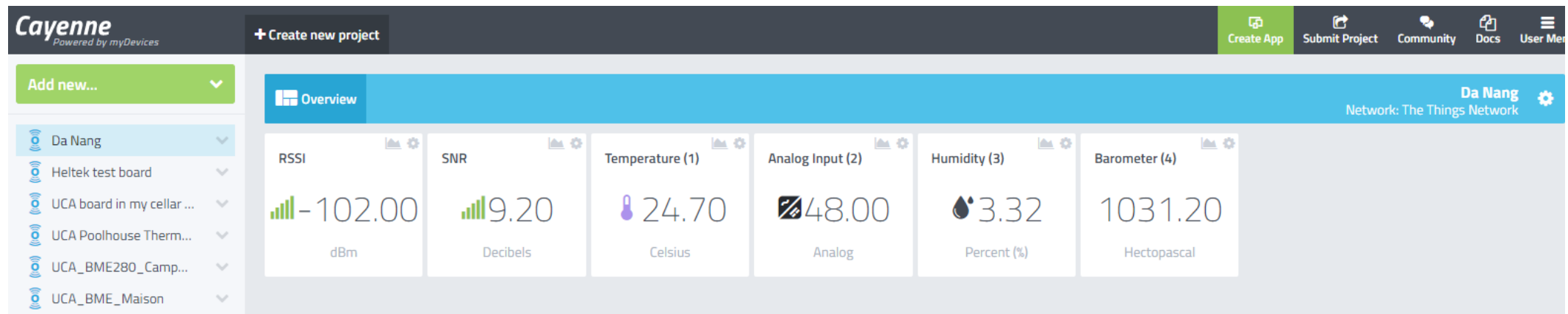
- Try now the code : UCA\_Education\_Board\Code\LORAWAN/OTAA/LP\_BME280/UCA-BME280.ino
- It use the sensor BME280 that measure T°C, Humidity and Pressure
- The code is using [Cayenne LPP format](#)
- Now you can see sensor data in the uplink packet

The screenshot displays the 'APPLICATION DATA' section of a Cayenne LPP format interface. It features a 'Filters' section with buttons for 'uplink', 'downlink', 'activation', 'ack', and 'error'. Below the filters, a table shows sensor data for an uplink packet. The table has columns for 'time', 'counter', 'port', and 'payload'. The 'payload' column contains the hex string '00 02 02 01 5A'. The 'time' column shows '22:16:49'. The 'counter' column shows '1'. The 'port' column shows '1'. The 'payload' column shows 'payload: 00 02 02 01 5A'. Above the table, there is a row of sensor data: '0 02 02 01 5A 03 68 43 04 73 27 65 05 02 00 3F analog\_in\_2: 3.46 analog\_in\_5: 0.63 barometric\_pressure\_4: 1008.5 relative\_humidity\_3: 33.5 temperature\_1: 22.4'. The interface also includes a 'pause' button and a 'clear' button in the top right corner.

time	counter	port	payload
22:16:49	1	1	payload: 00 02 02 01 5A

# Using Cayenne to see you data

- Go to <https://developers.mydevices.com/cayenne/features/> and sign up
- Add a device by selecting LoRa/TheThingsNetwork and Cayenne LPP.
- Just add your device EUI
- You should see your data





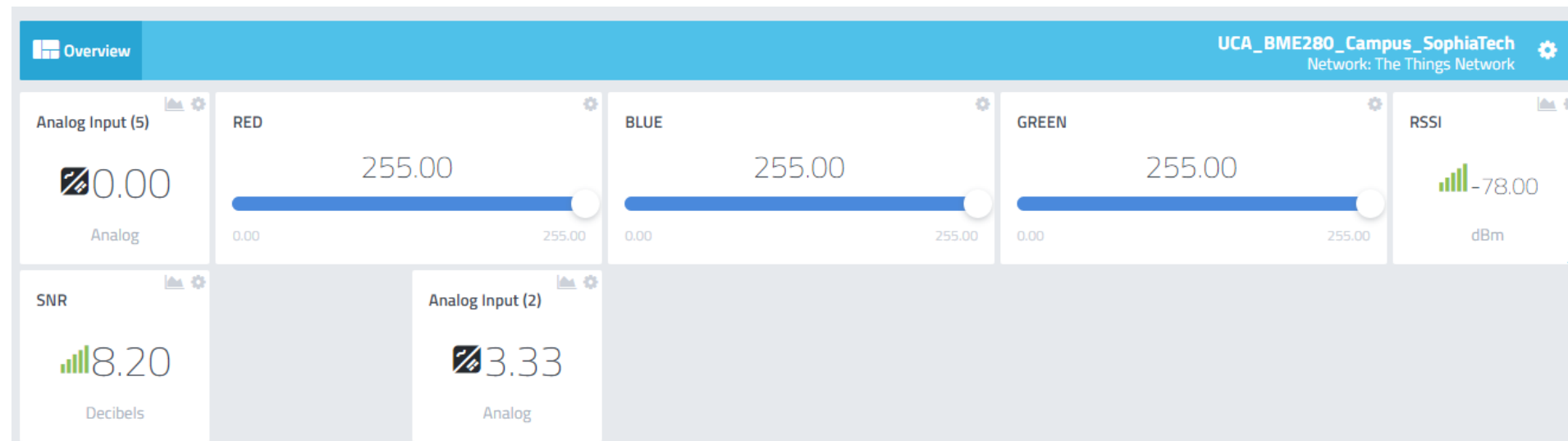
# Downlink with LoRaWAN

- You are now going to control the LED color from CAYENNE.
- Use the code in :

TO BE DONE

# 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

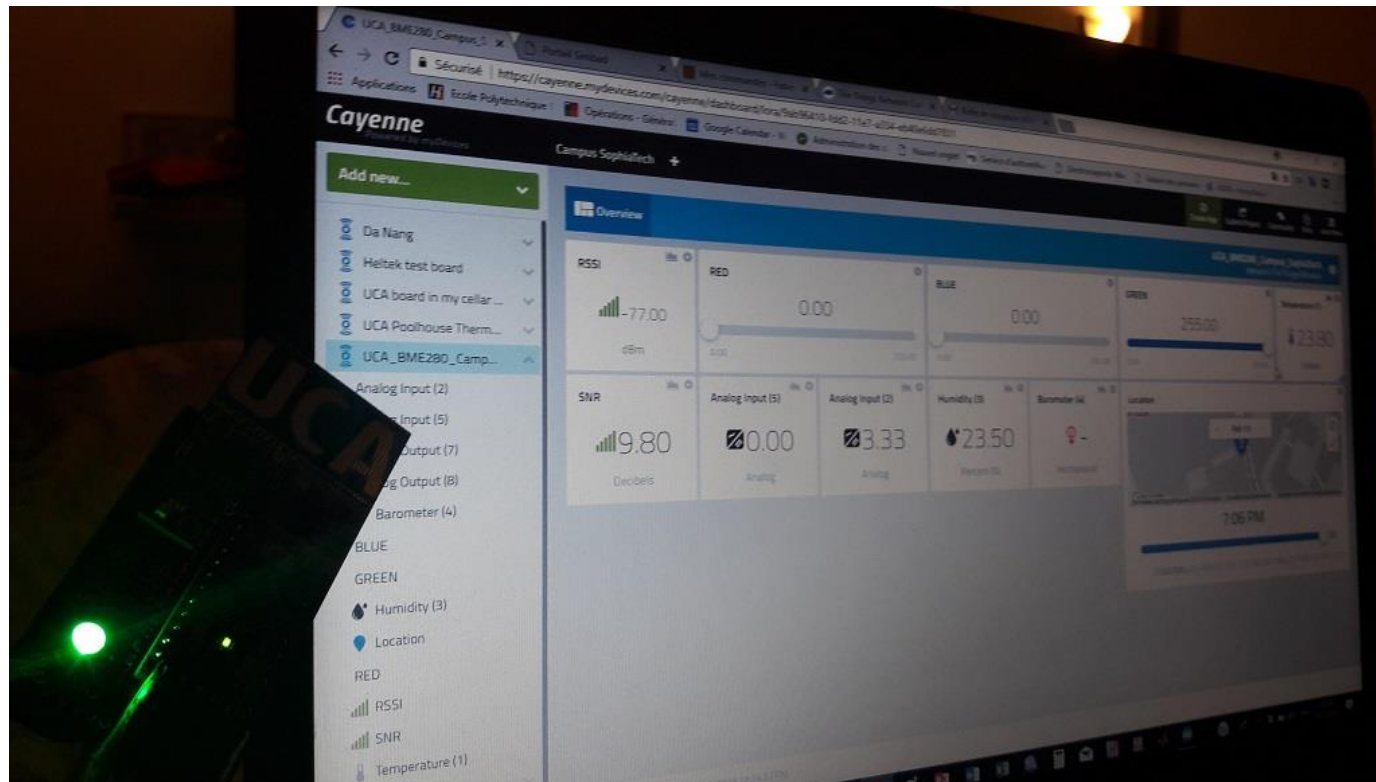
Filters

	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 opens a received window for downlink
- With Cayenne, only one color can be updated at the same time



Good luck for you projects !

This board as been funded by UCA

