UNIVERSITÉ CÔTE D'AZUR

DNIIT DANANG INTERNATIONAL INSTITUTE OF TECHNOLOGY

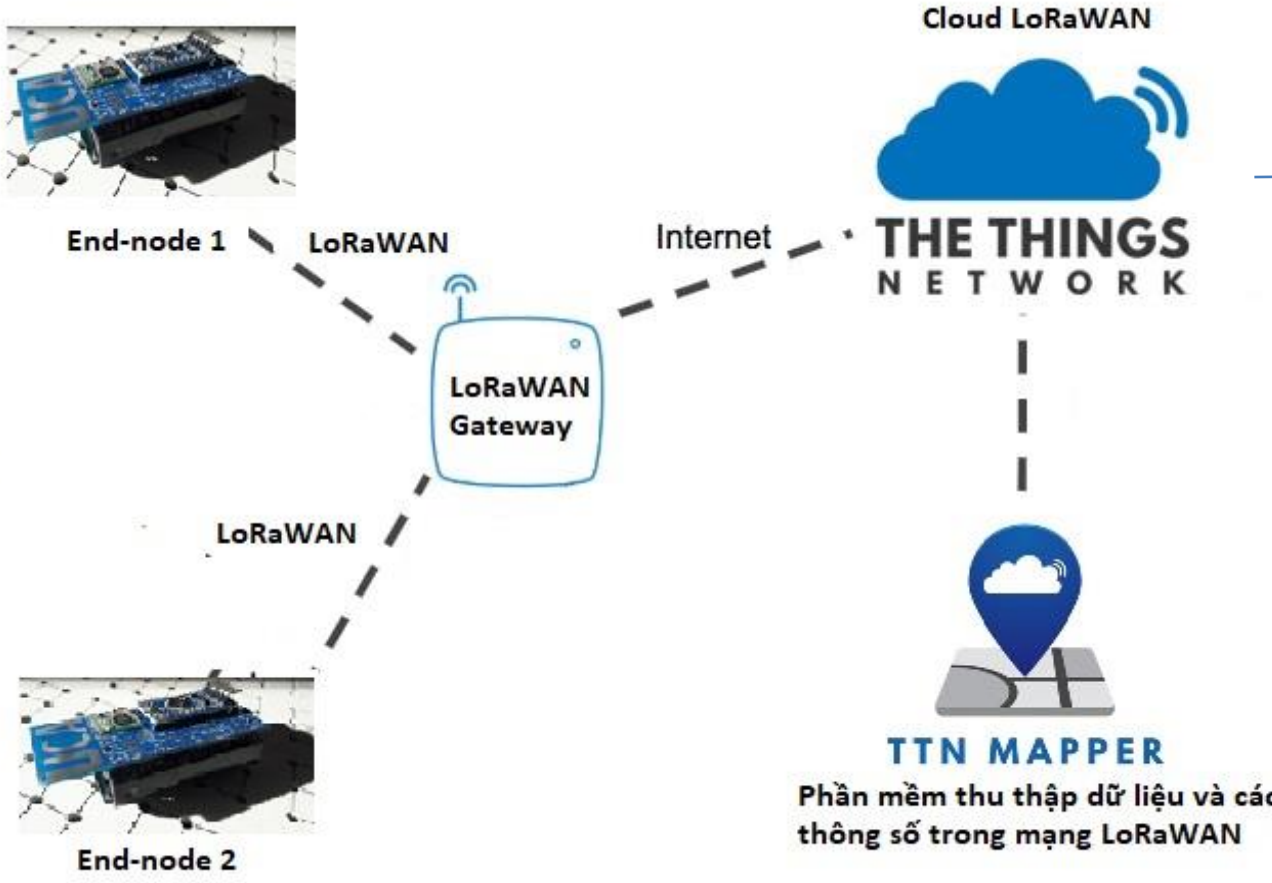
INITIATIVE OF THE EXCELLENCE - INVESTMENT FOR THE FUTURE

LoRaWan
Deploy LoRaWAN Application Using UCA Education Board

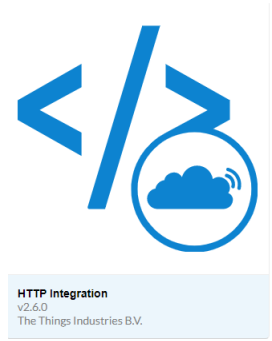
Tran Van Lic, F. Ferrero

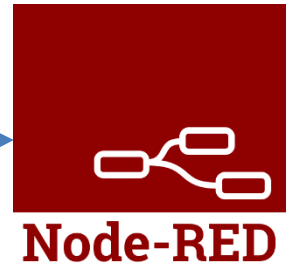
Hue, 03/2021

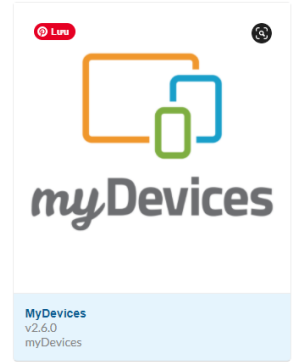
LoRaWAN Architecture



HTTP/MQTT







Application Server

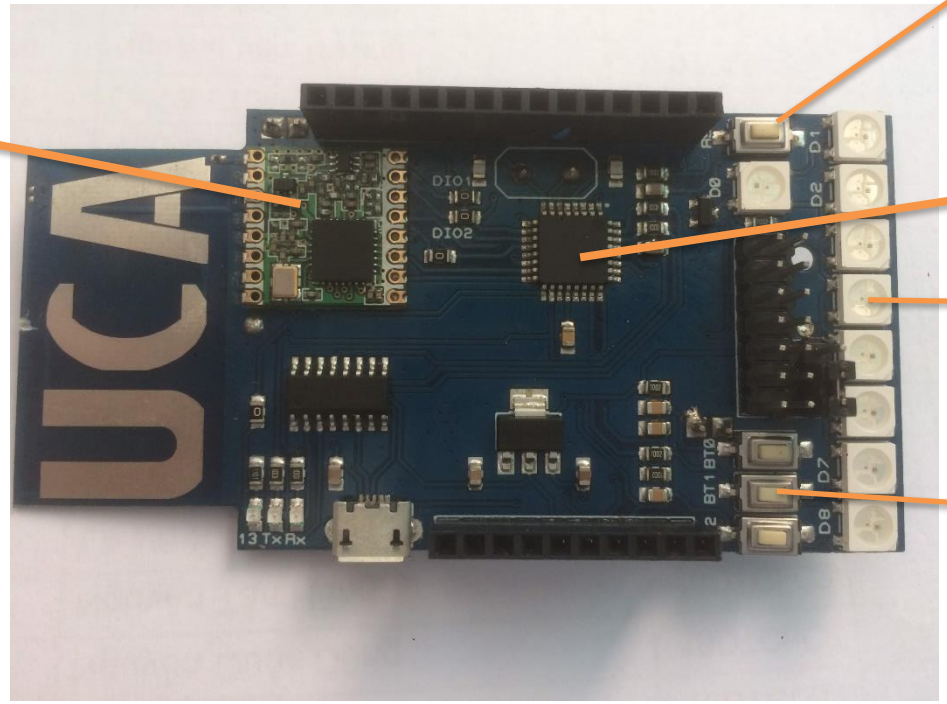
UCA Education Board

RFM95 Module

Specifications:

Power:DC 1.8V ~ 3.7V
Transmit power: 20dBm
Working frequency: 868/915
Modulation mode: LoRA/FSK/GFSK/00K MHz

High sensitivity: down to -148 dBm.
Data Rate:300kbps

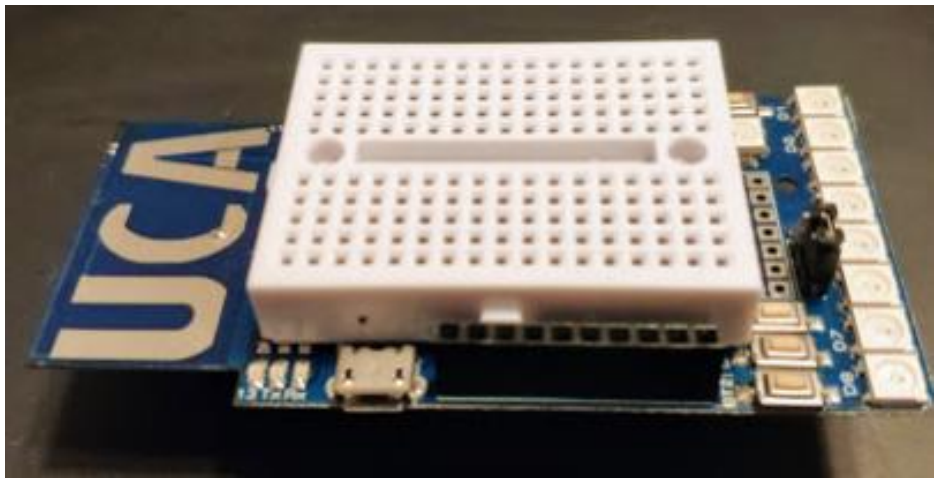


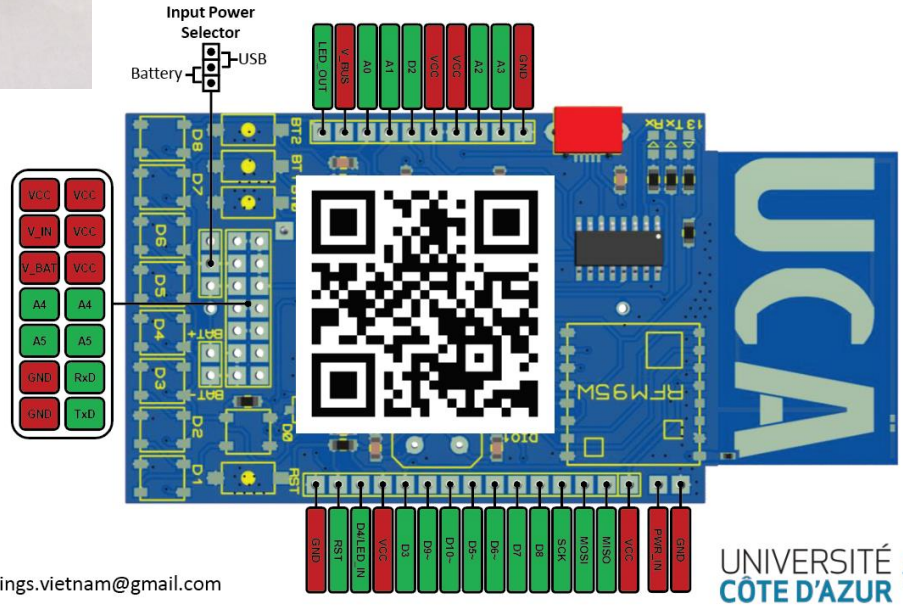
RESET button

RGB LED

Chip AT MEGA 328P

Button 1,2,3





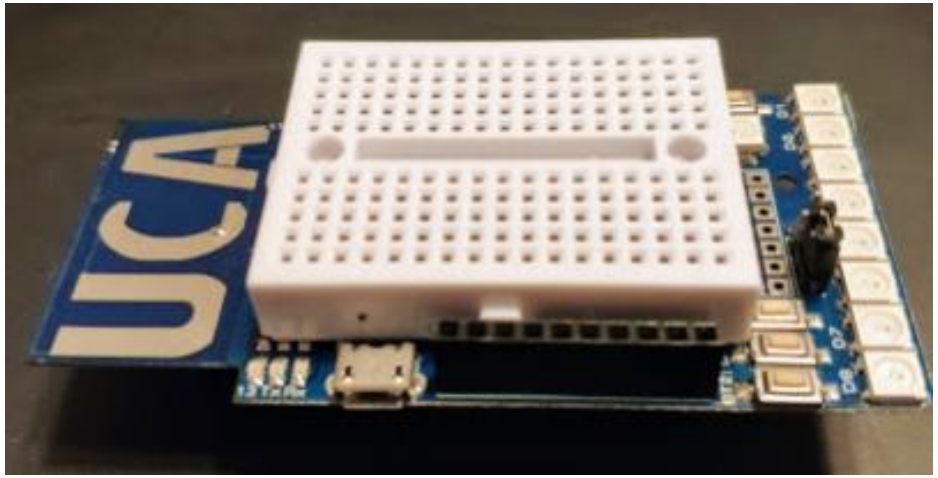
rfthings.vietnam@gmail.com

UNIVERSITÉ CÔTE D'AZUR

Preparing

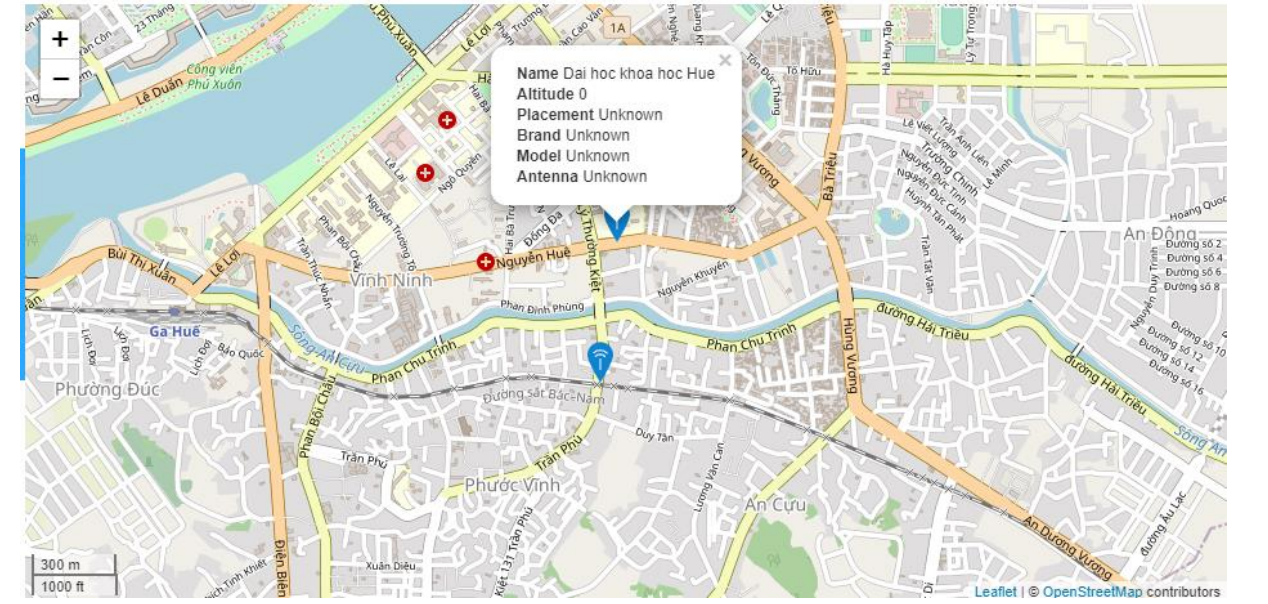
- Software
  - Ardiuno IDE: https://www.arduino.cc/en/software
  - Codes are available on : https://github.com/FabienFerrero/UCA_Education_Board
  - The board is using CH340C chip for USB to You may need to install the driver to use

    https://sparks.gogo.co.nz/ch340.html
- Hardware
  - UCA Education Board



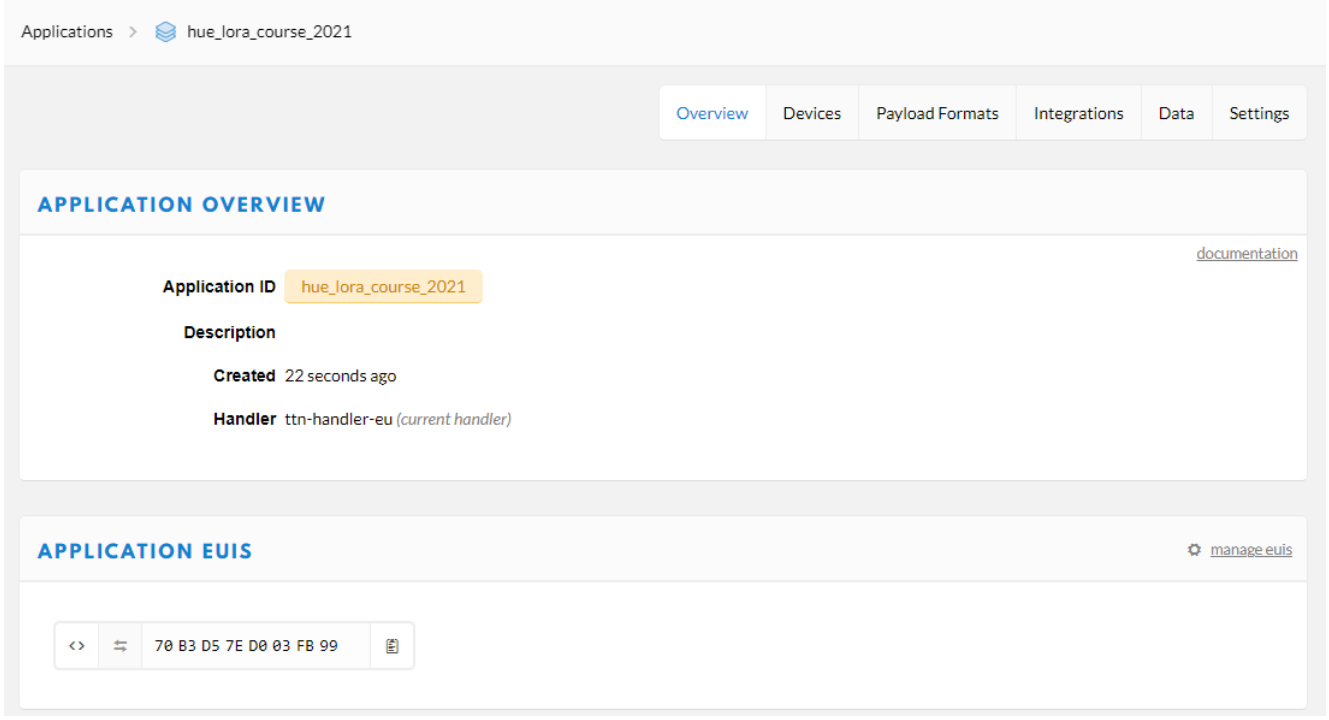
LoRaWan with The Thing Network

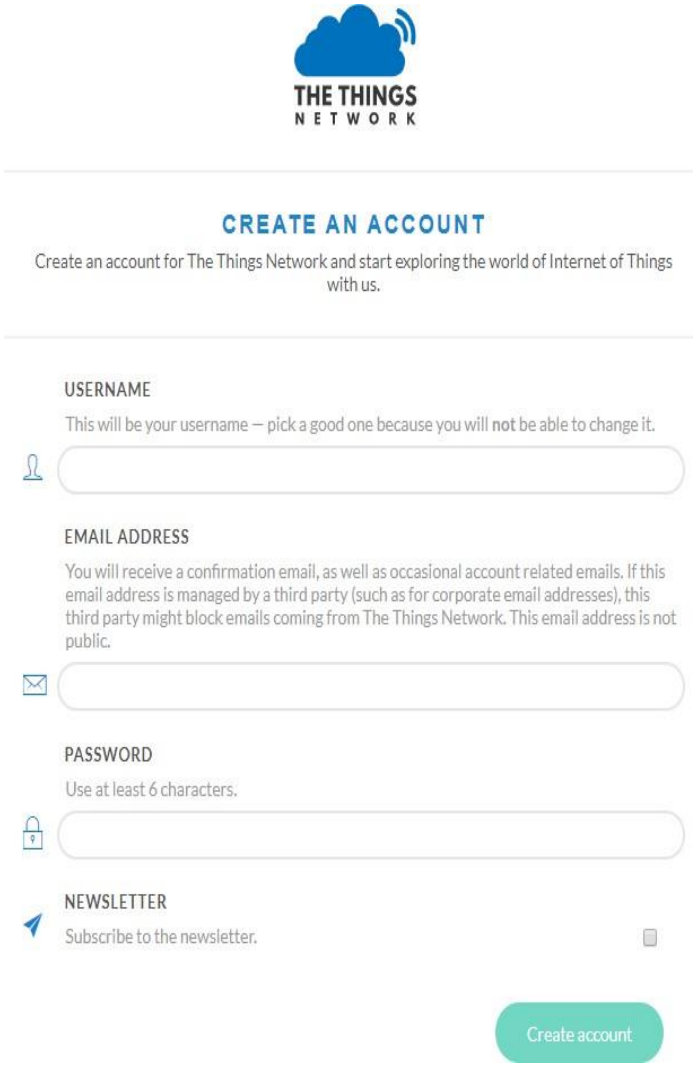
- The Things Network is a **global**, open, crowd-sourced Internet of Things data network.
- **The Things Network Backend** route messages from Nodes to the right Application, and back
- TTN is free
- 17541 LoRa gateways are connected to TTN around the world
- Any TTN can use any GWs, it is a collaborative network



Create a TTN account

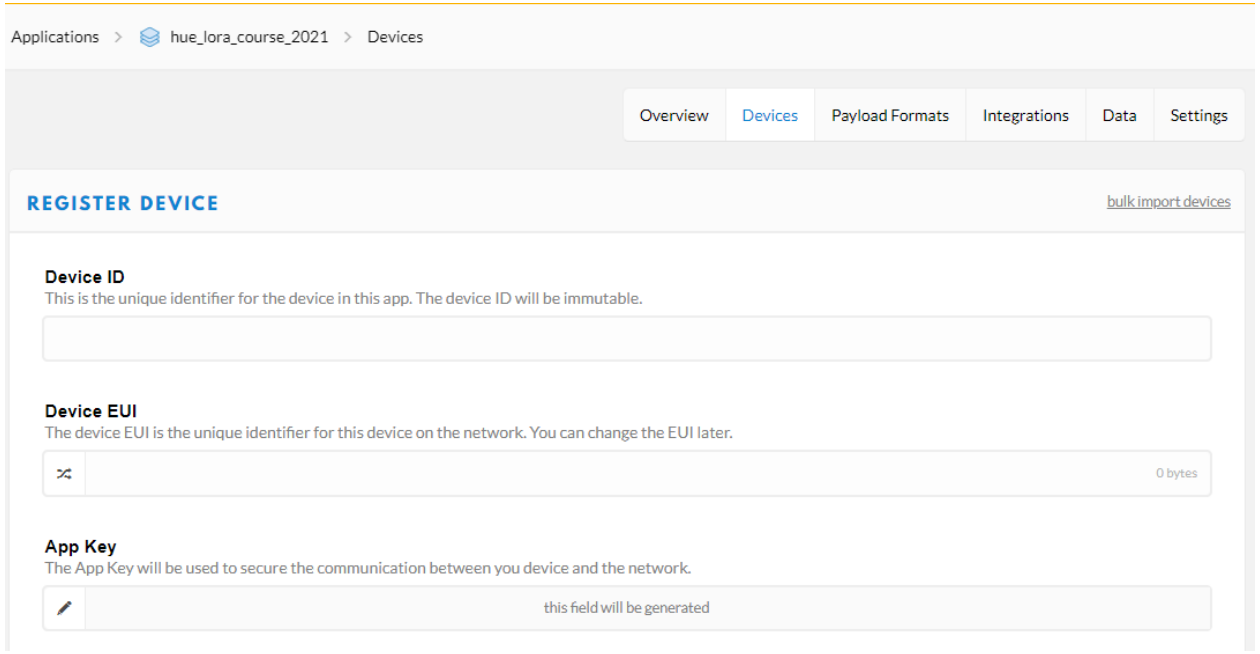
- First, you have to register to https://www.thethingsnetwork.org/
- Then, give me your USERNAME, I will add you as a collaborator in our application – hue_lora_course 2021

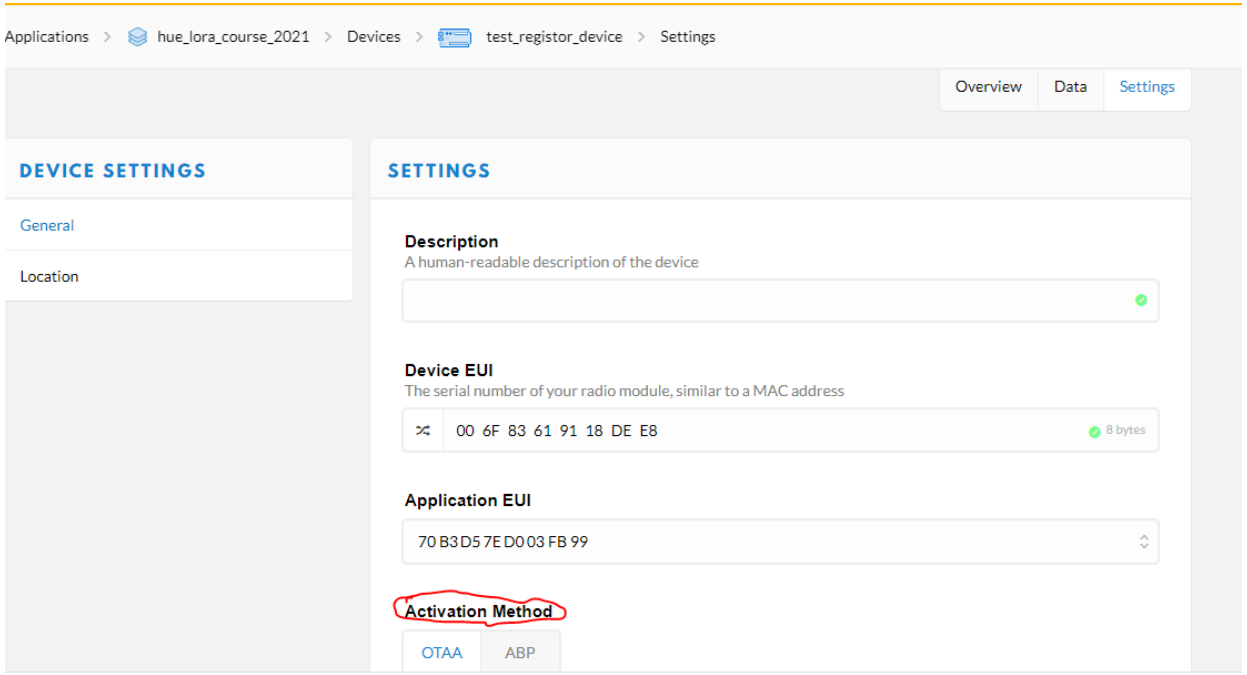


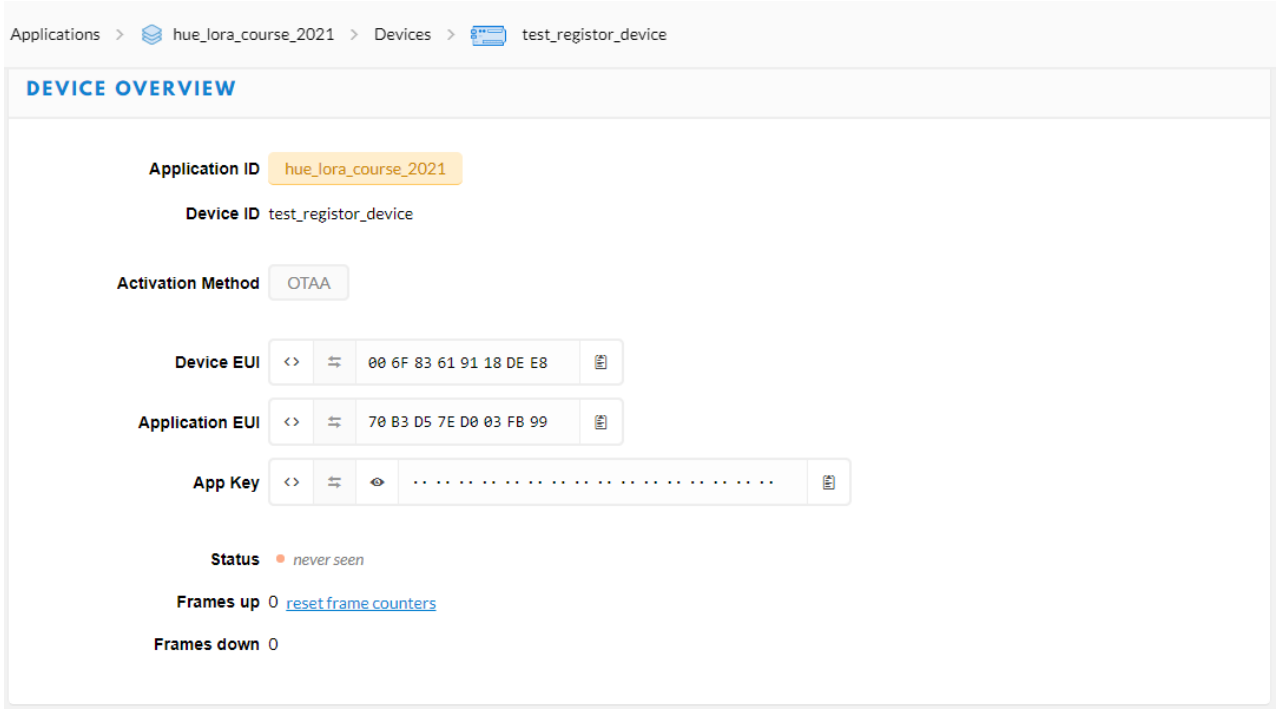


Adding a new device

- Go to **« application »** and choose the available application – **hue_lora_course_2021**
- Click on « **register device** »
- In DEVICE OVERVIEW, you get usefull information on your device. Of course, status is : « Never Seen »
- First, we will connect using ABP (Activation by Personalization)





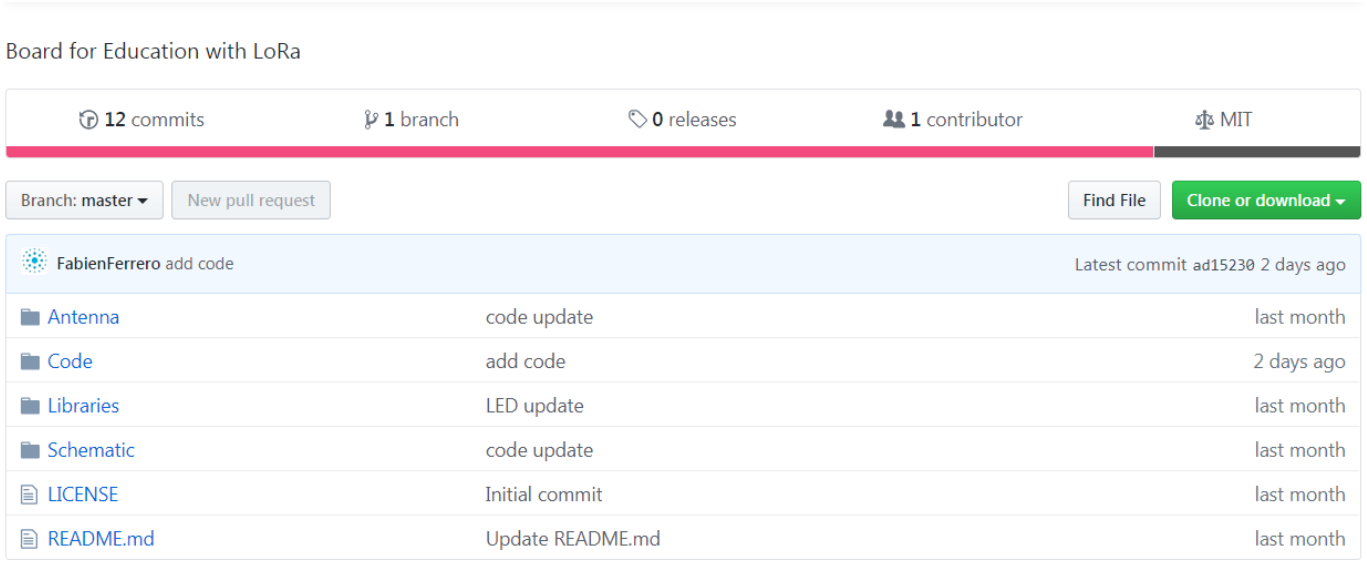


Downloading Arduino code on Github

- For this tutorial, your are going to use Arduino codes
- Codes are available on :

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”

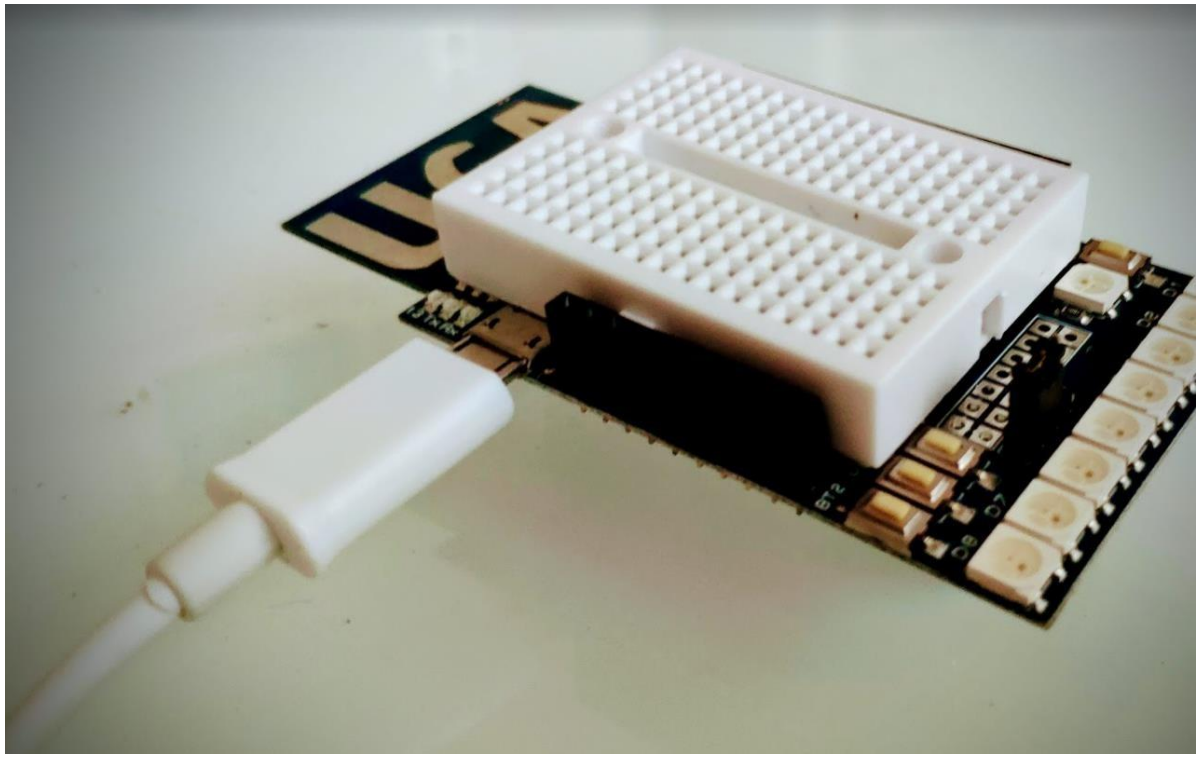


Configuring your Arduino IDE

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

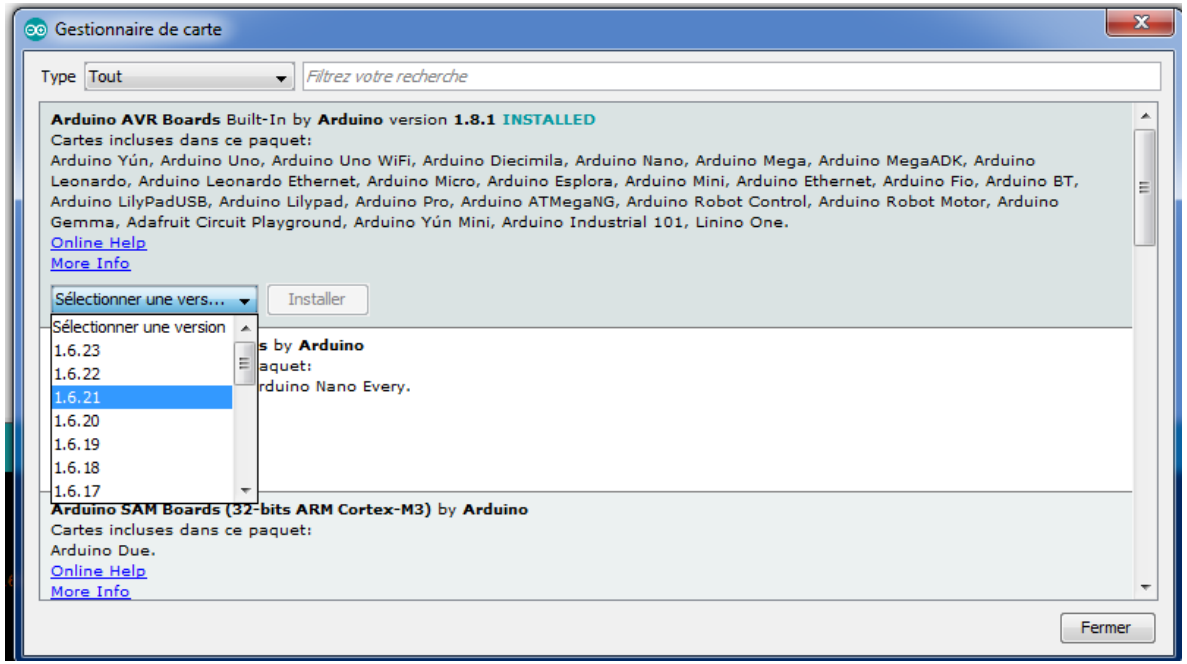
It will install the libs needed during the tutorial

- 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



UCA_Education_Board/Code/LORAWAN/ABP/Basic/**UCA-ABP_Basic/**

```
/*******************************************************************************/
// Region definition (will change de frequency bands
// Define only 1 country
//
//#define CFG_EU 1
#define CFG_VN 1

/*******************************************************************************/

#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>

// LoRaWAN end-device address (DevAddr)

static const u4_t DEVADDR = 0x00000000;

// 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 };
```

Switch to frequency plan in Vietnam

Device address

Network Sesstion Key

App Sesstion Key

```
#if defined(CFG_EU)
// Set up the 8 channels used
LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI);      // g-band
LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK,  DR_FSK),  BAND_MILLI);      // g2-band

#elif defined(CFG_VN)
// Set up the 8 channels used
LMIC_setupChannel(0, 921400000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(1, 921600000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI);      // g-band
LMIC_setupChannel(2, 921800000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(3, 922000000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(4, 922200000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(5, 922400000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(6, 922600000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(7, 922800000, DR_RANGE_MAP(DR_SF12, DR_SF7),  BAND_CENTI);      // g-band
LMIC_setupChannel(8, 922700000, DR_RANGE_MAP(DR_FSK,  DR_FSK),  BAND_MILLI);      // g2-band
#endif
```

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

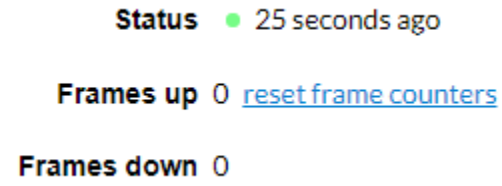
Activation by Personalization (ABP)

- 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

```
#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>

// LoRaWAN end-device address (DevAddr)

static const u4_t DEVADDR = 0x00000000;

// 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 };
```

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



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)

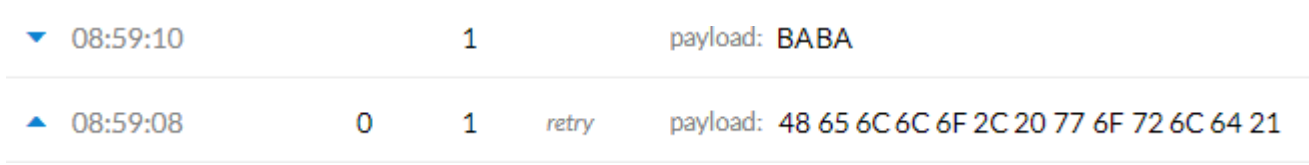
Frame counter security

- 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
```



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 ?

OTAA | ABP

Device Address

26 04 11 88 — 4 bytes

Network Session Key

84 55 4B 72 82 1B CB 22 2D 41 83 96 B7 43 64 B5 — 16 bytes

App Session Key

70 73 2D 4B 54 76 0C 42 A7 59 F8 D5 37 BC FA 24 — 16 bytes

Frame Counter Width

16 bit | 32 bit

Frame Counter Checks

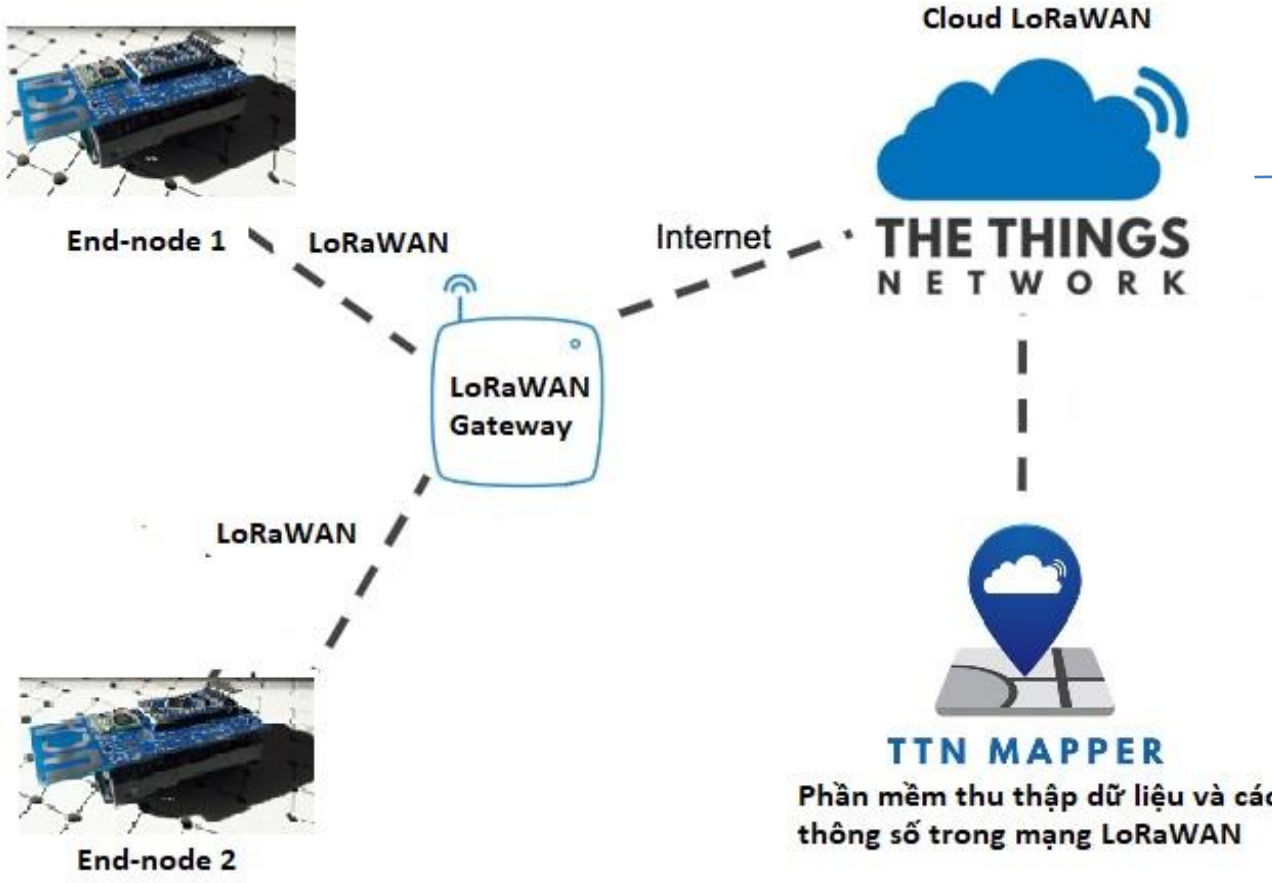
Disabling frame counter checks drastically reduces security and should only be used for development purposes

UCA-ABP_Basic

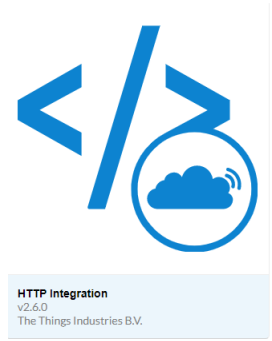
```
static const PROGMEM u1_t NWKSKEY[16] = { 0x84, 0x55, 0x4B, 0x72, 0x82, 0x1B, 0xCB, 0x22, 0x2D, 0x41, 0x83, 0x96, 0xB7, 0x43, 0x64, 0xB5 };


// 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] = { 0x70, 0x73, 0x2D, 0x4B, 0x54, 0x76, 0x0C, 0x42, 0xA7, 0x59, 0xF8, 0xD5, 0x37, 0xBC, 0xFA, 0x24 };


// These callbacks are only used in over-the-air activation, so they are
// left empty here (we cannot leave them out completely unless
// DISABLE_JOIN is set in config.h, otherwise the linker will complain).
void os_getArtEui (u1_t* buf) { }
void os_getDevEui (u1_t* buf) { }
void os_getDevKey (u1_t* buf) { }

static uint8_t mydata[] = "Hello, world!";
static osjob_t sendjob;

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).
const unsigned TX_INTERVAL = 30;

// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 8,
    .dio = {3, 7, 6},
};

void onEvent (ev_t ev) {
```

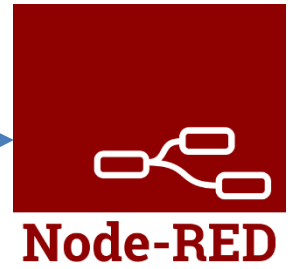
Thay đổi dữ liệu gửi đi
Ví dụ: mydata[] = “Hello, Toi la nhom abc”;

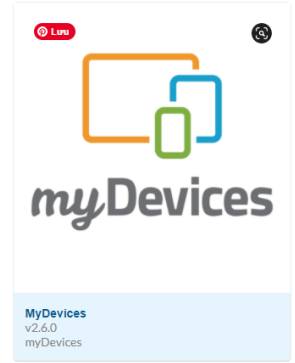
LoRaWAN Architecture



HTTP/MQTT

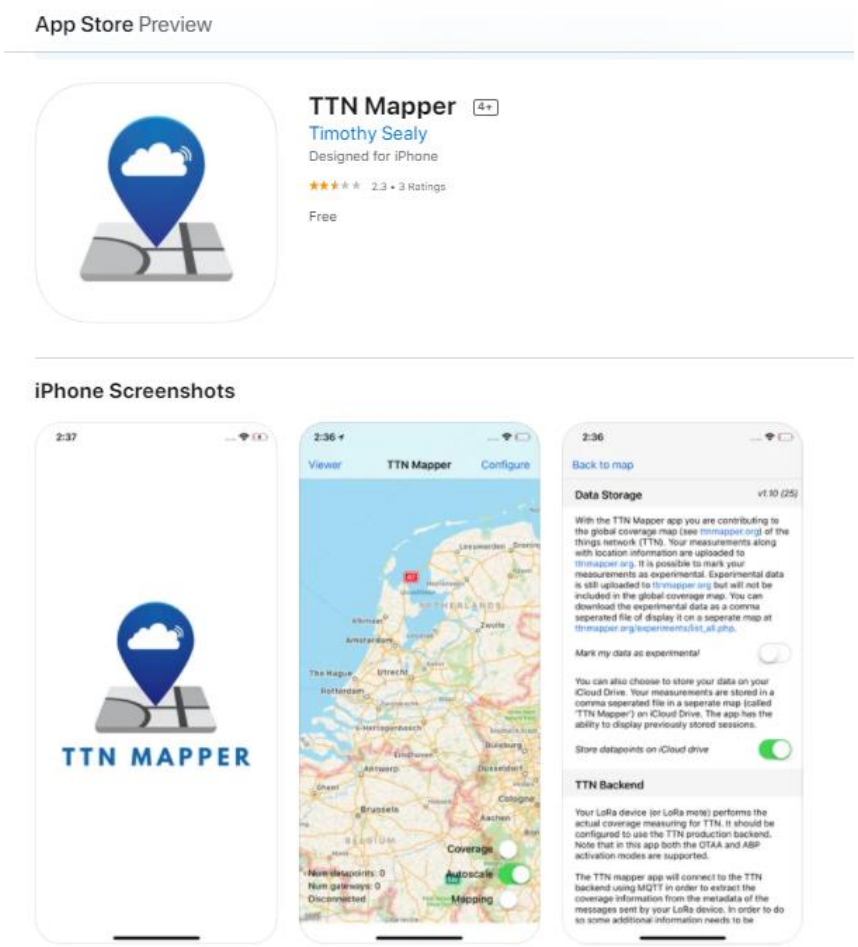


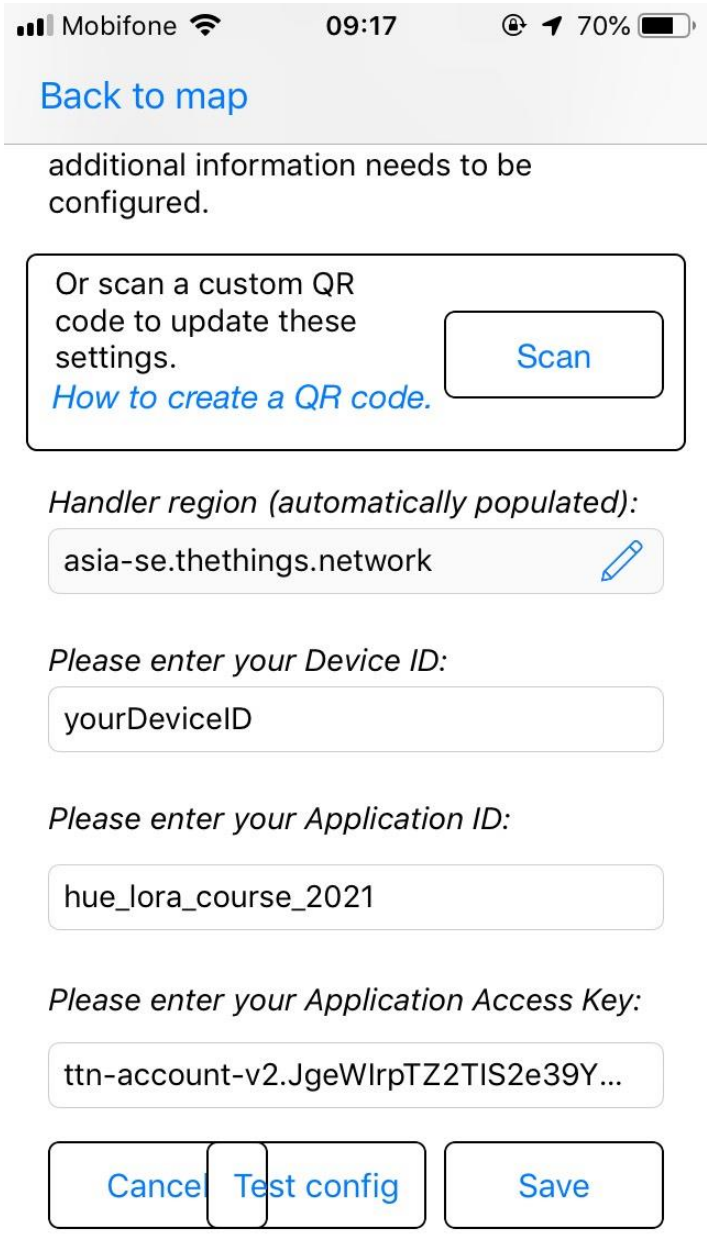




Application Server

Install TTN Mapper

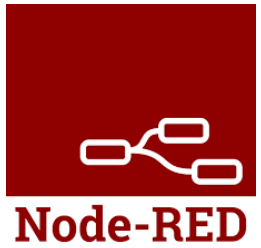




Scan HERE
QR CODE – Configure TTN mapper for hue_lora_course_2021

Test TTN Mapper

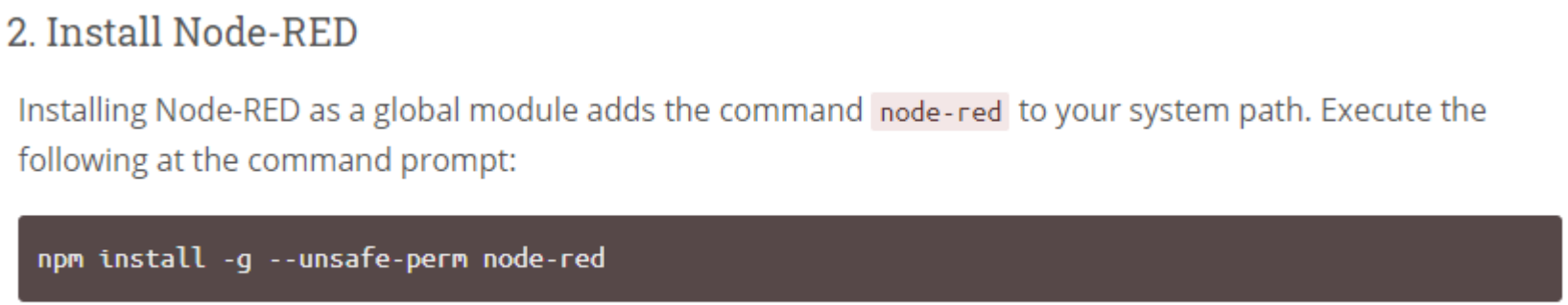
Application server - Node Red



- Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.
- It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.
- Built on Node.js
  - The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

Install Nodejs and Node-Red

- Install nodejs: https://nodejs.org/en/download/
- Install git : https://git-scm.com/downloads
- Then install Node Red, follow this tutorial :
  - https://nodered.org/docs/getting-started/windows



- Install package in Node Red :

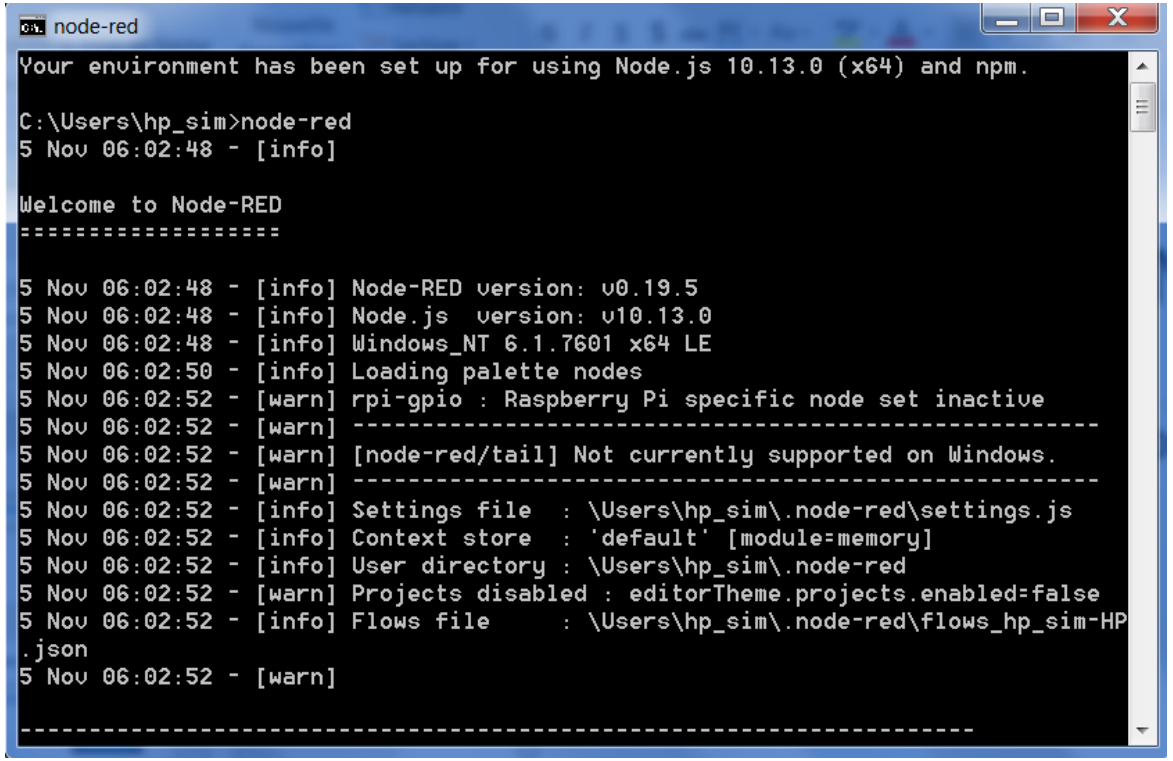
  node-red-contrib-ttn

  cd $HOME/.node-red

  npm install node-red-contrib-ttn

Connecting to TTN

- Start NODE.js command prompt
- Run : node-red
- Open your web browser and go to http://127.0.0.1:1880



•On the editor, click here

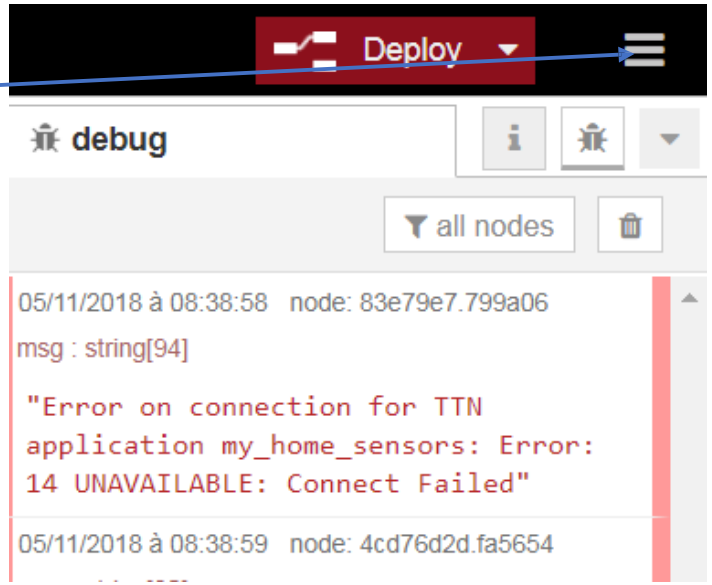
And go to palette editor

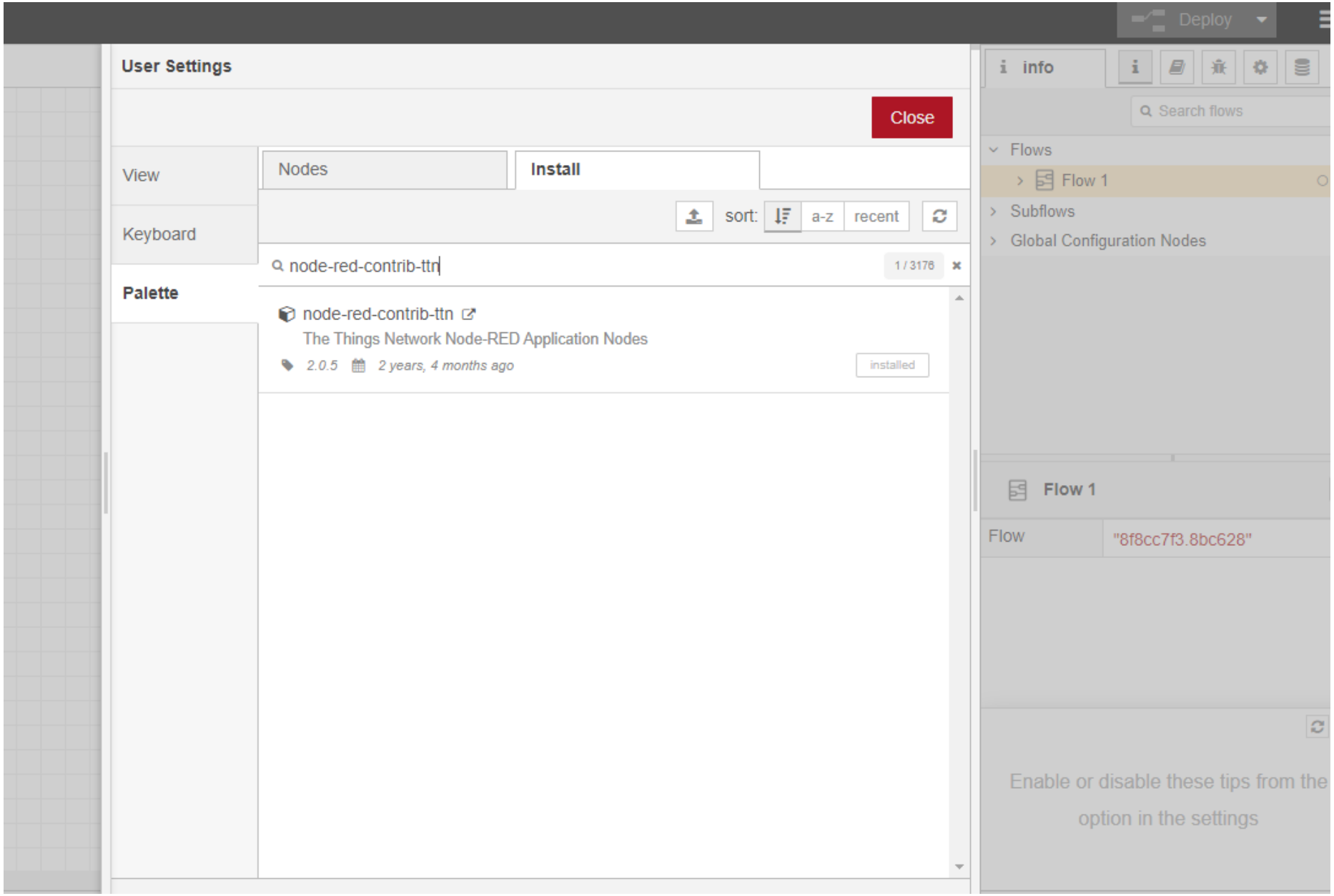
Install :

- node-red-contrib-ttn

OR npm install node-red-contrib-ttn

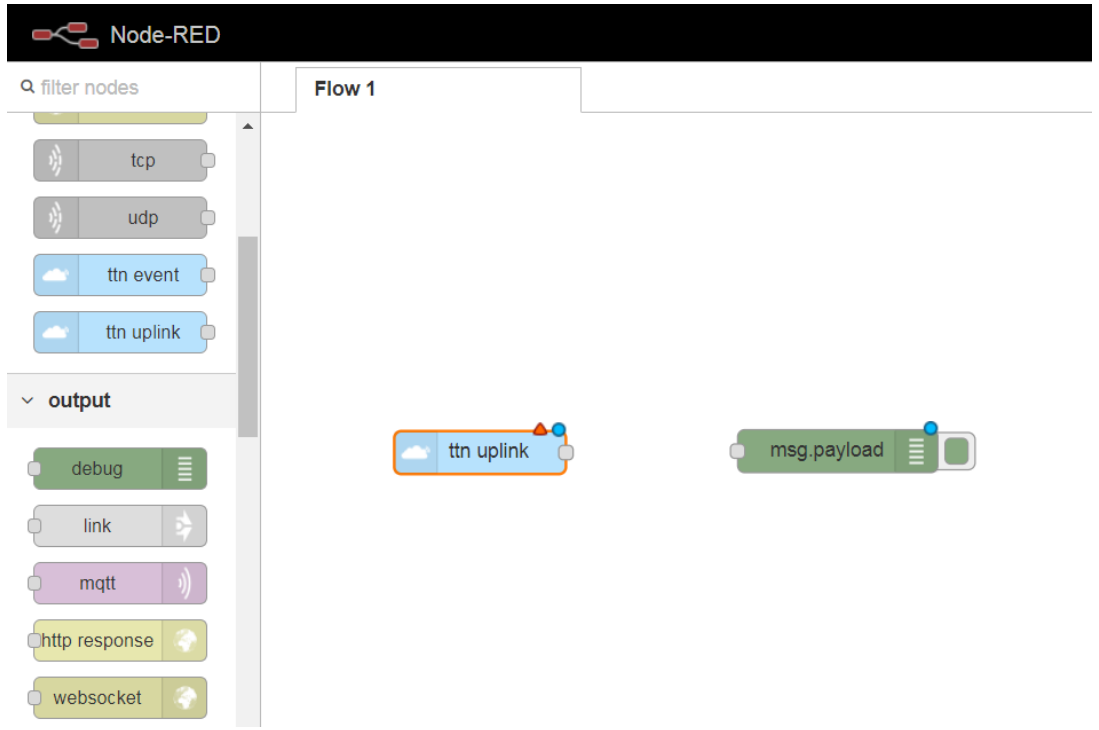
OR npm i node-red-contrib-ttn@1.0.0.

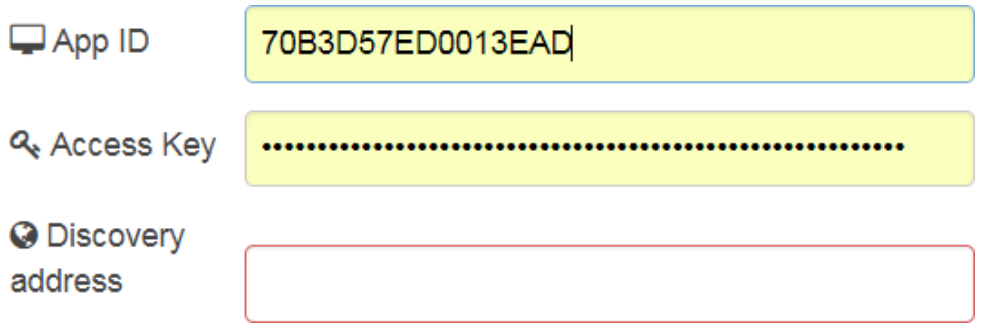


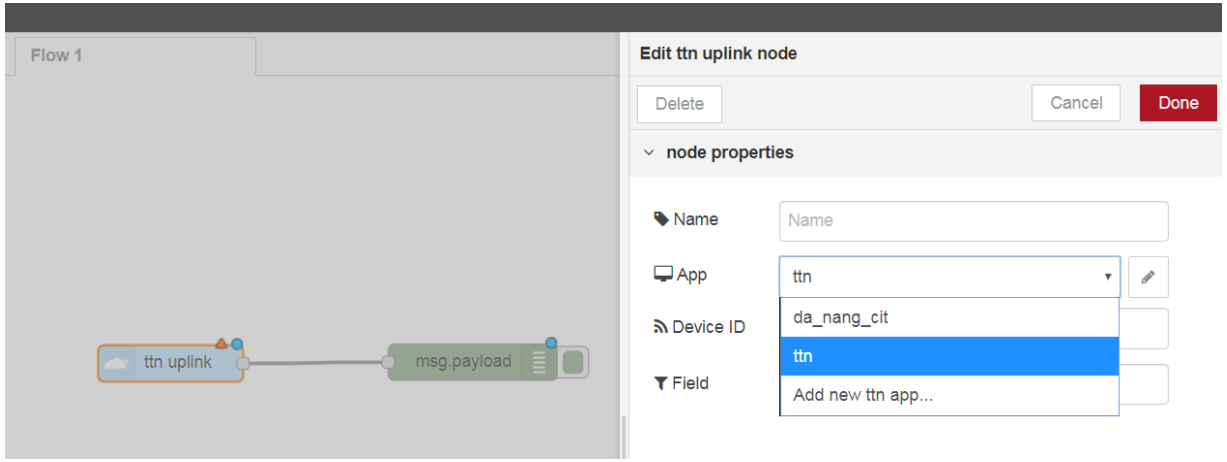

Deploy
User Settings
Close
View
Keyboard
Palette
Nodes
Install
sort:
a-z
recent
node-red-contrib-ttn
1 / 3176
node-red-contrib-ttn
The Things Network Node-RED Application Nodes
2.0.5
2 years, 4 months ago
installed
info
Search flows
Flows
Flow 1
Subflows
Global Configuration Nodes
Flow 1
Flow
"8f8cc7f3.8bc628"
Enable or disable these tips from the
option in the settings

Connecting to TTN

- You have the graphical Node-red editor
- Add ttn uplink and a debug output
- Edit TTN uplink
- Choose « Add new ttn app …» in App and click on edit





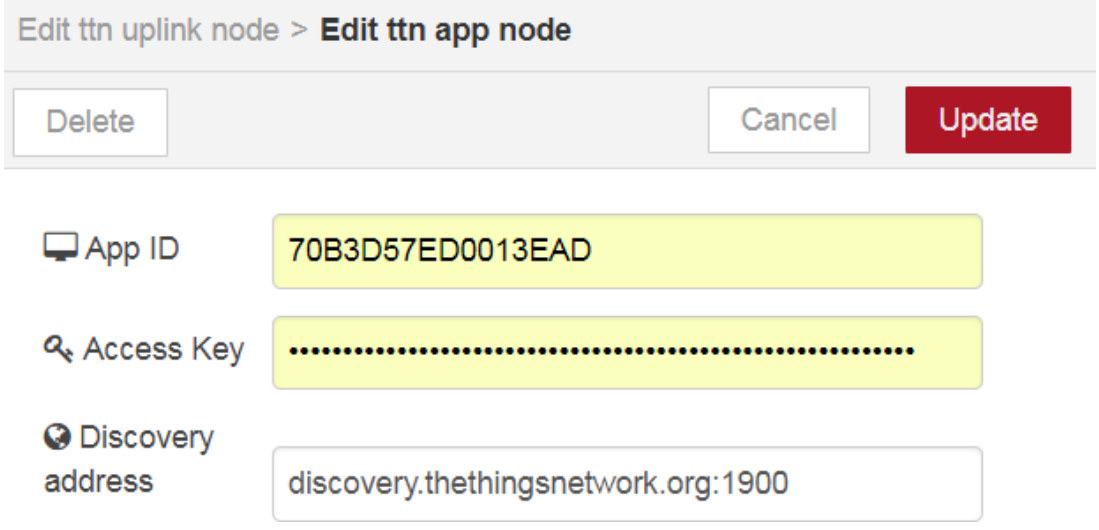


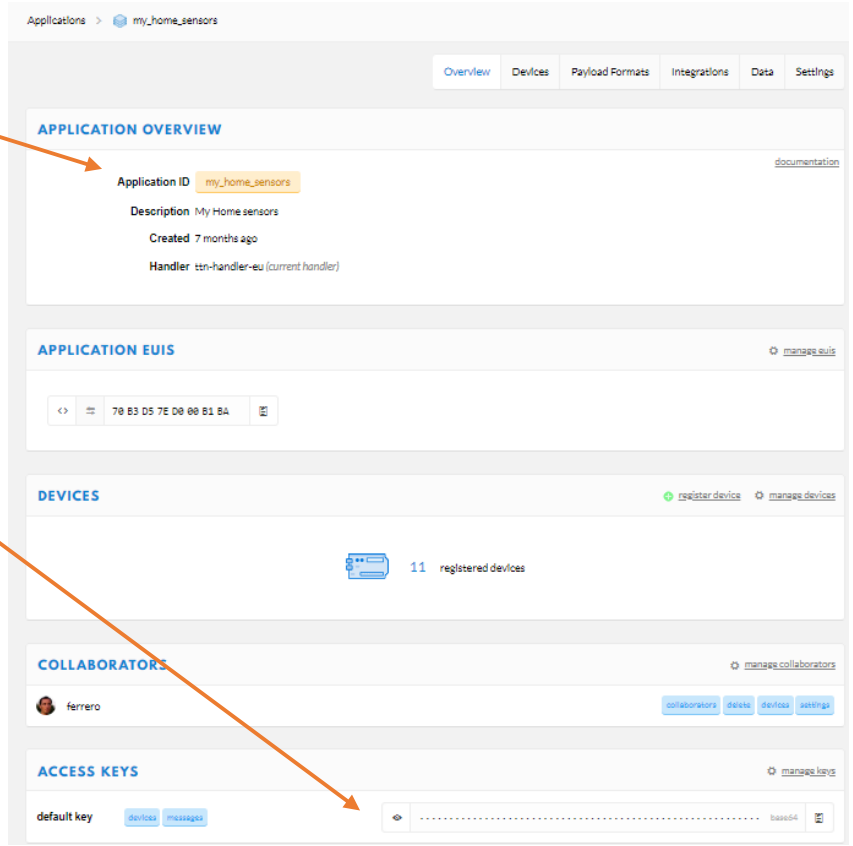
Connecting to TTN

- You need :
  - App ID :
  - Access Key :
  - Discovery adress :

discovery.thethingsnetwork.org:1900

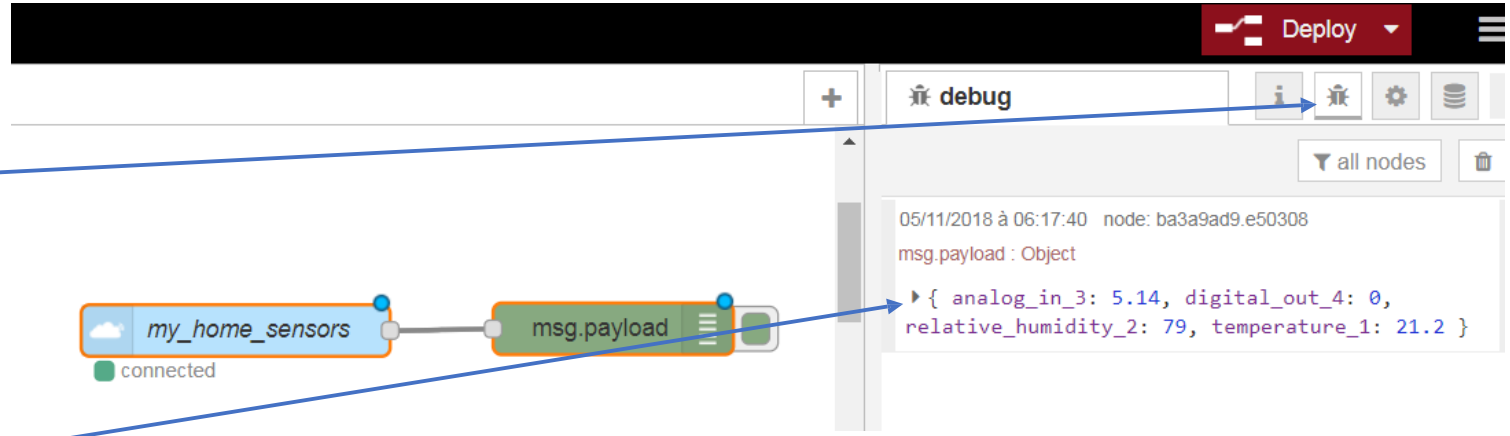
- Go to you application in TTN
- Copy paste the Application ID and Access Key

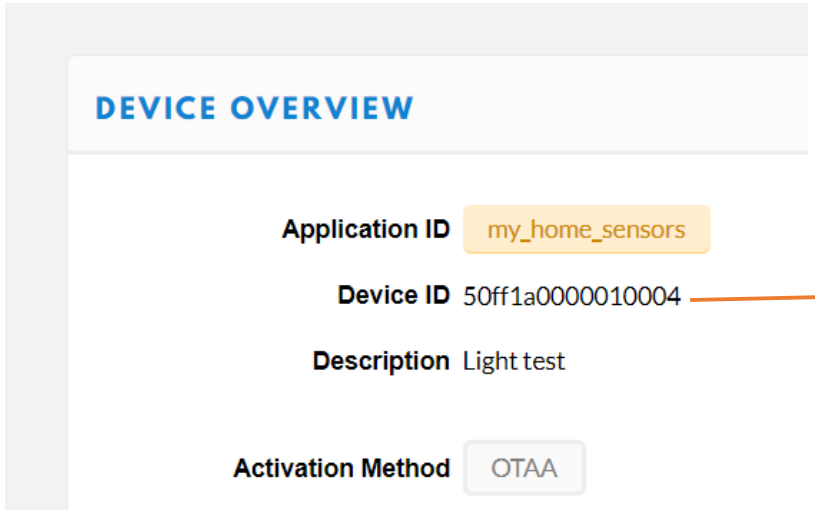


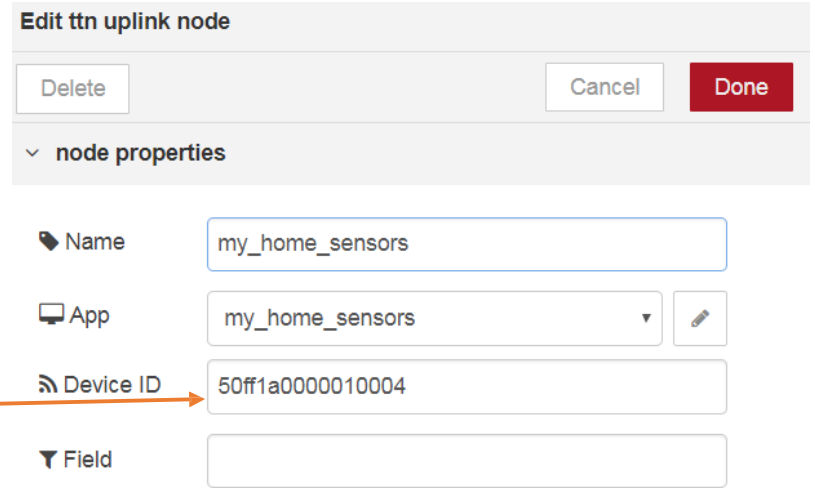


Connecting to TTN

- Click on Deploy
- You uplink TTN should be connected
- Click on debug window
- You will receive the packet of the application
- If you want to filter only your device, add your device ID
- Click here :

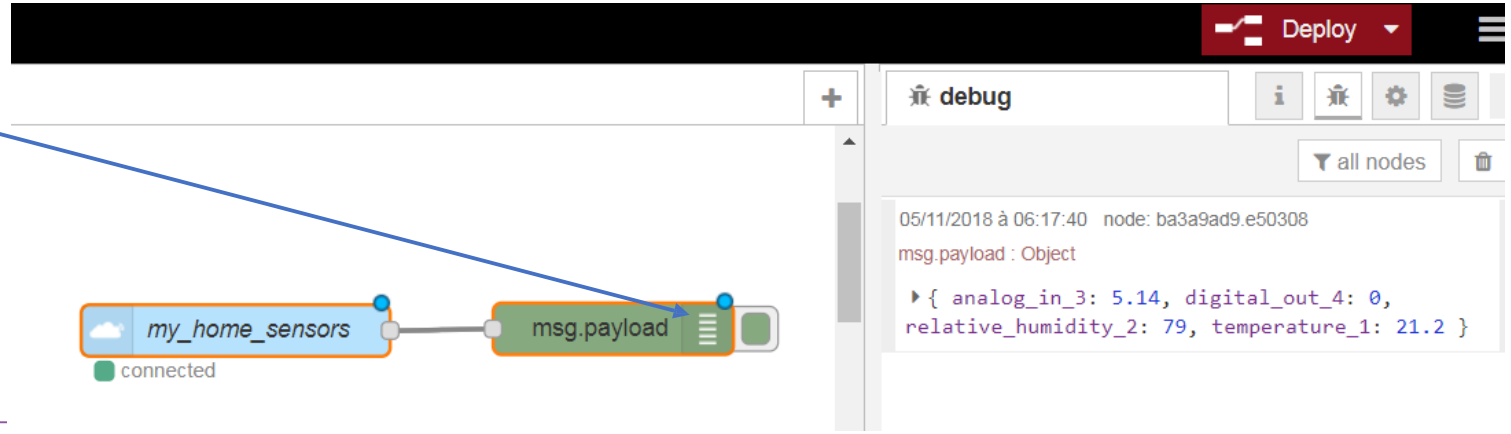


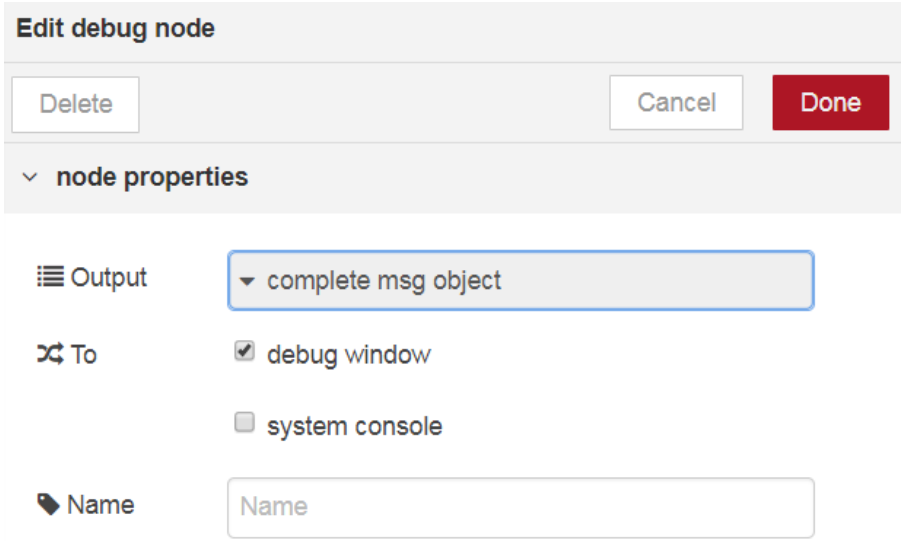




Connecting to TTN

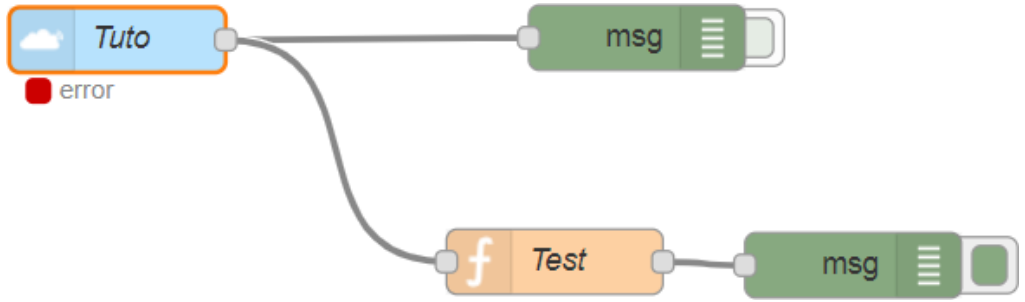
- Click here :
- Choose « complete msg object »
- And Deploy
- You have now more information of your uplink





Connecting to TTN

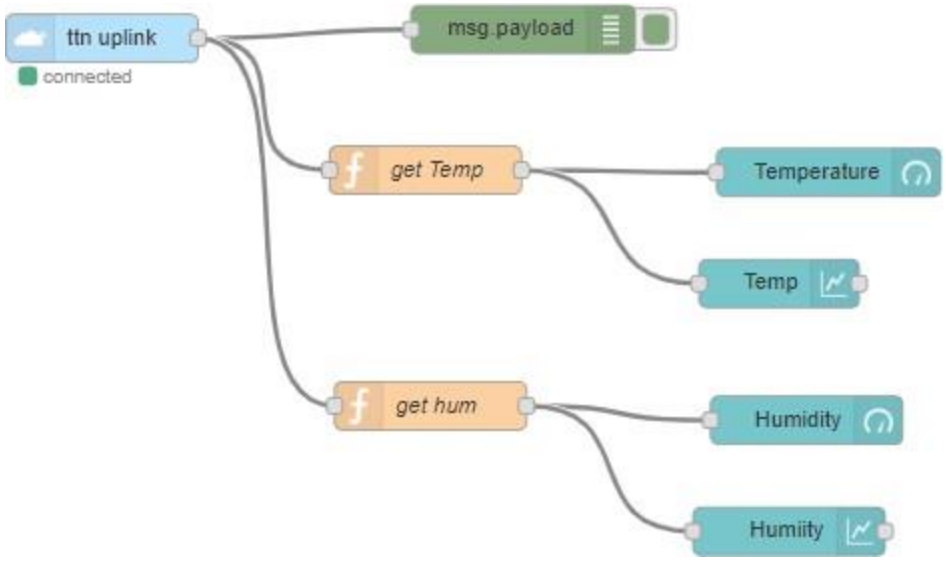
- If you want to extract only 1 data,
- As an exemple the RSSI (received signal Strength indicator
- Use a function to extract the wanted data




```
var gateways = msg.metadata.gateways;

return {
 // Some fields from the metadata freq:
msg.metadata.frequency,
cr: msg.metadata.cr,
dr: msg.metadata.dr,

// Combine RSSI and SNR of all gateways into two  arrays:
rssi: gateways.map(gw => gw.rssi),
snr: gateways.map(gw => gw.snr),

};
```

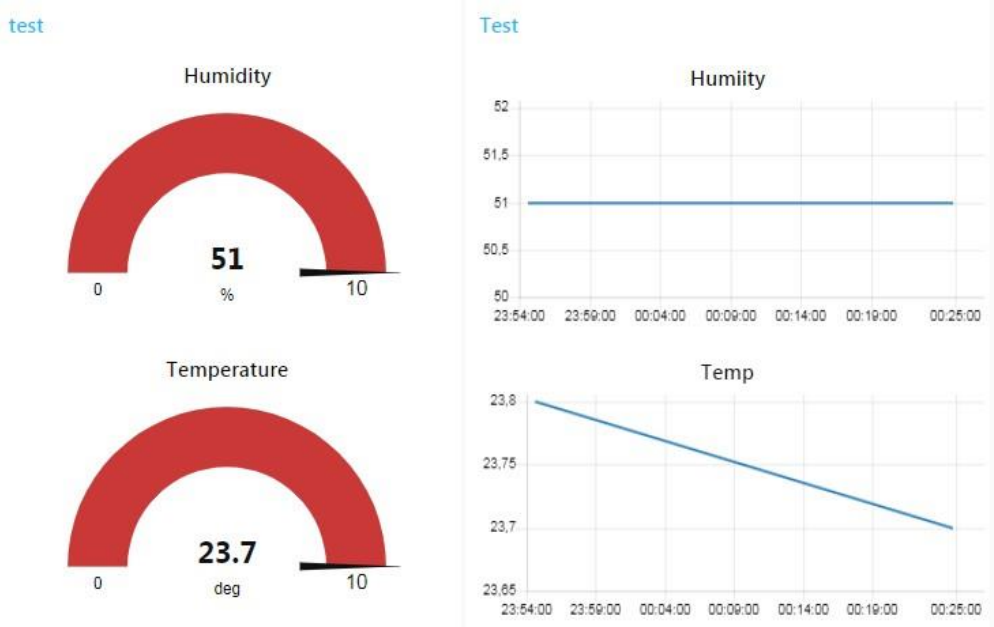
Add a Dashboard

- Go to Manage Palette, select Install
- Install : node-red-dashboard
- Add a function to extract sensor values (Temp, Hum, luminosity…)
- Add Gauge and Graph for Dashboard section
- Go to : http://127.0.0.1:1880/ui/



Function

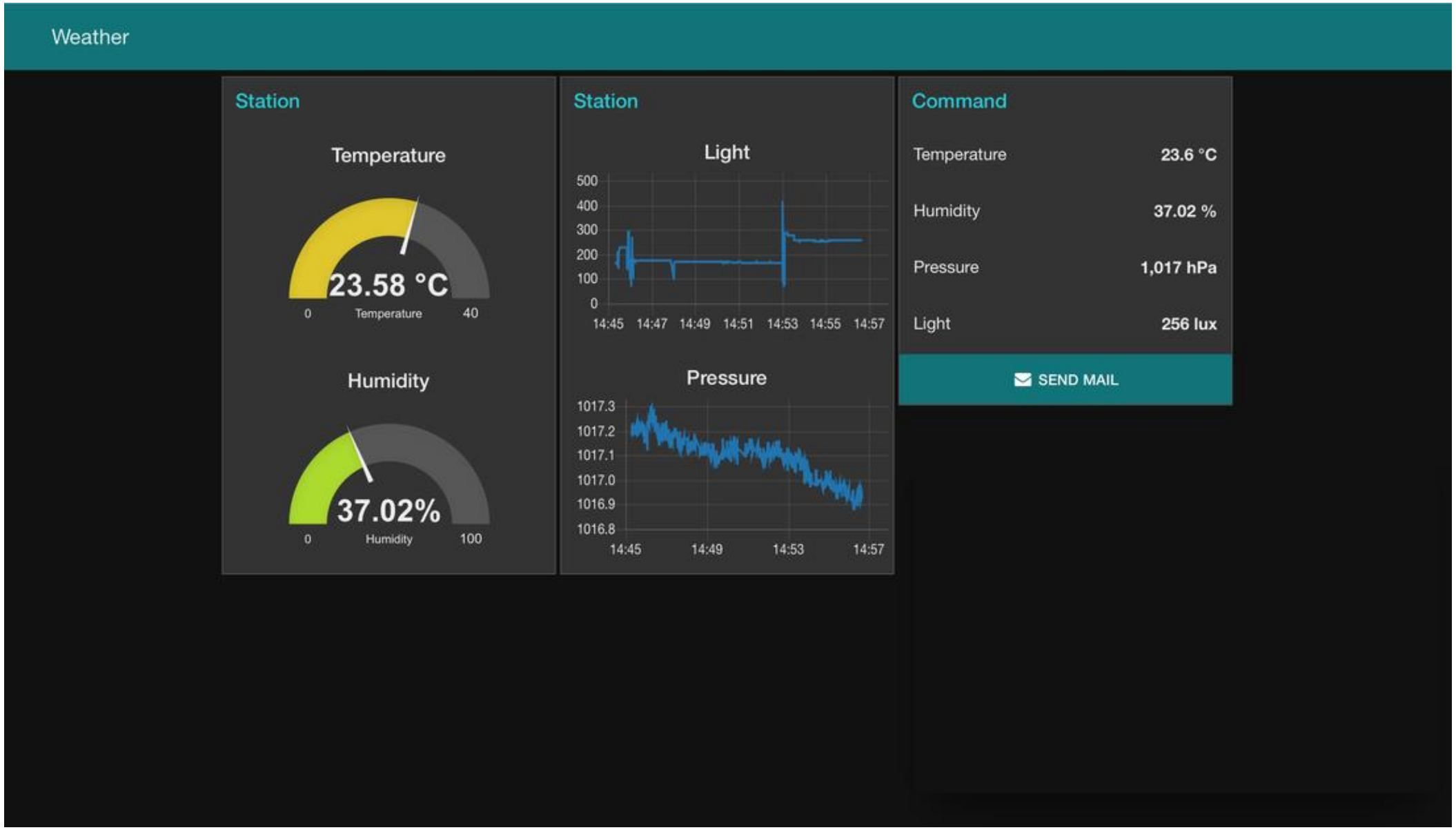
```
1  var temp = {}
2  temp.payload = msg.payload.temperature_4;
3  return temp;
```



UCA Education Board with sensor

- Depending on the sensor available from your instructor, you may use :
  - BME280 : T°c, Humidity and Pressure with I2C connection
  - SI7021 : T°c and Humidity with I2C connection
  - TEMT6000 : Ambiant light with analog output
  - SRC4+ : Distance with digital connection
  - Or else …
- You have to wire the sensor on the breadboard on the UCA board

Example: Weather monitoring



Good luck for you projects!

UNIVERSITÉ CÔTE D'AZUR

DNIIT DANANG INTERNATIONAL INSTITUTE OF TECHNOLOGY

INITIATIVE OF THE EXCELLENCE - INVESTMENT FOR THE FUTURE