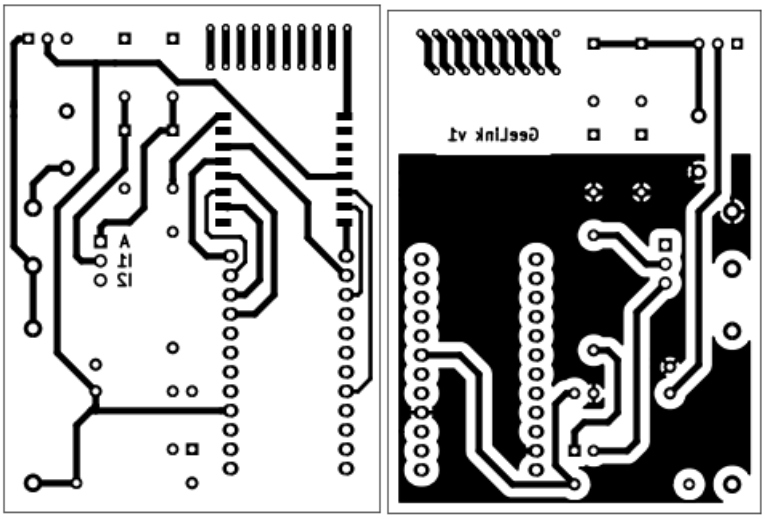
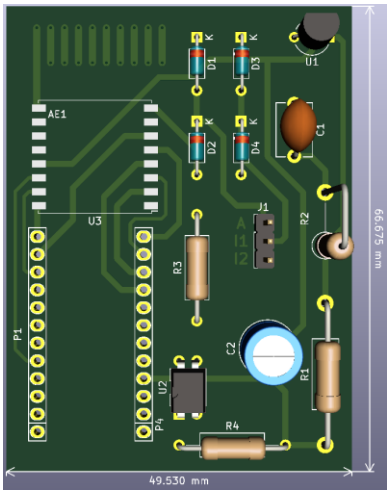
❖ GeeLink Tutorials

Hardware:

Print PCB design by KiCad https://github.com/hovuduybao/GeeLink/tree/master/KiCad/GeeLink_v1/PCB



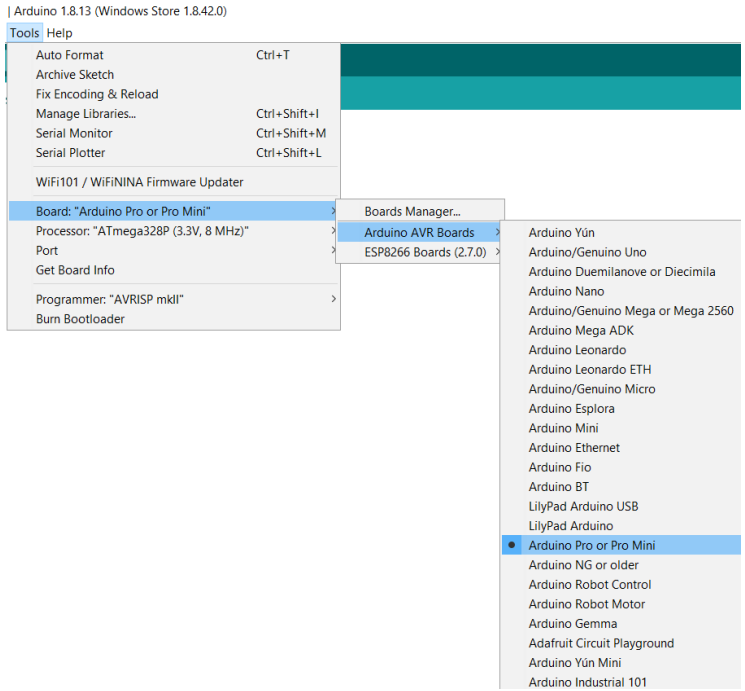
Solder the components following this 3D model



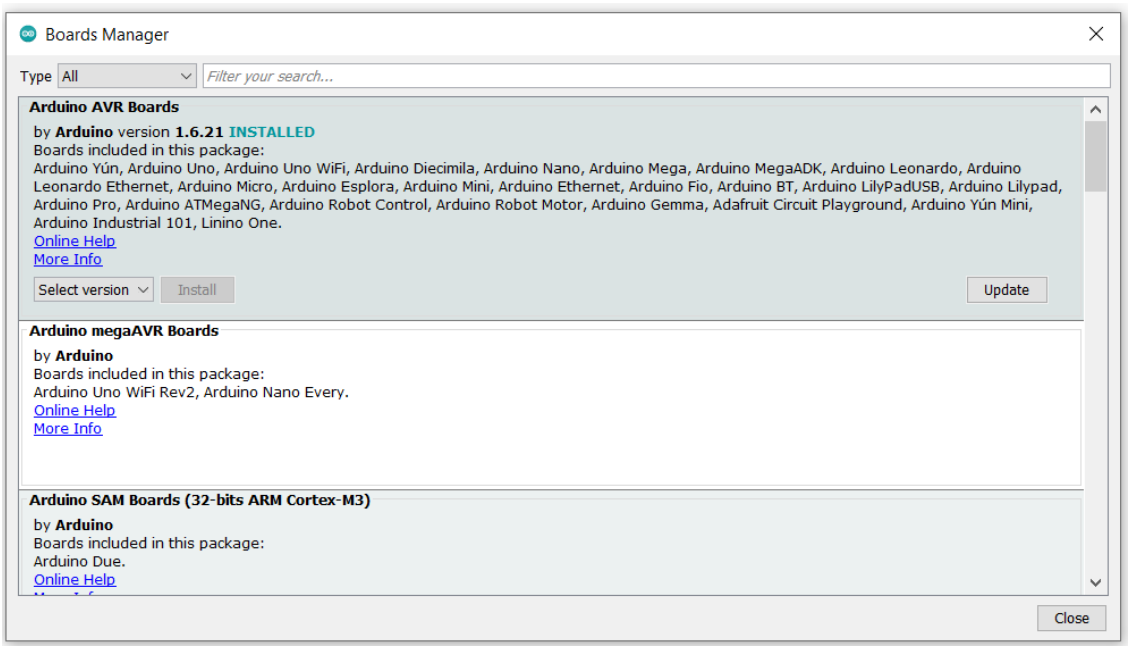
Connect 3 pins (I1, I2 and A) from TIC to the PCB board.

Software:

- Arduino IDE:
  - Setup board information:



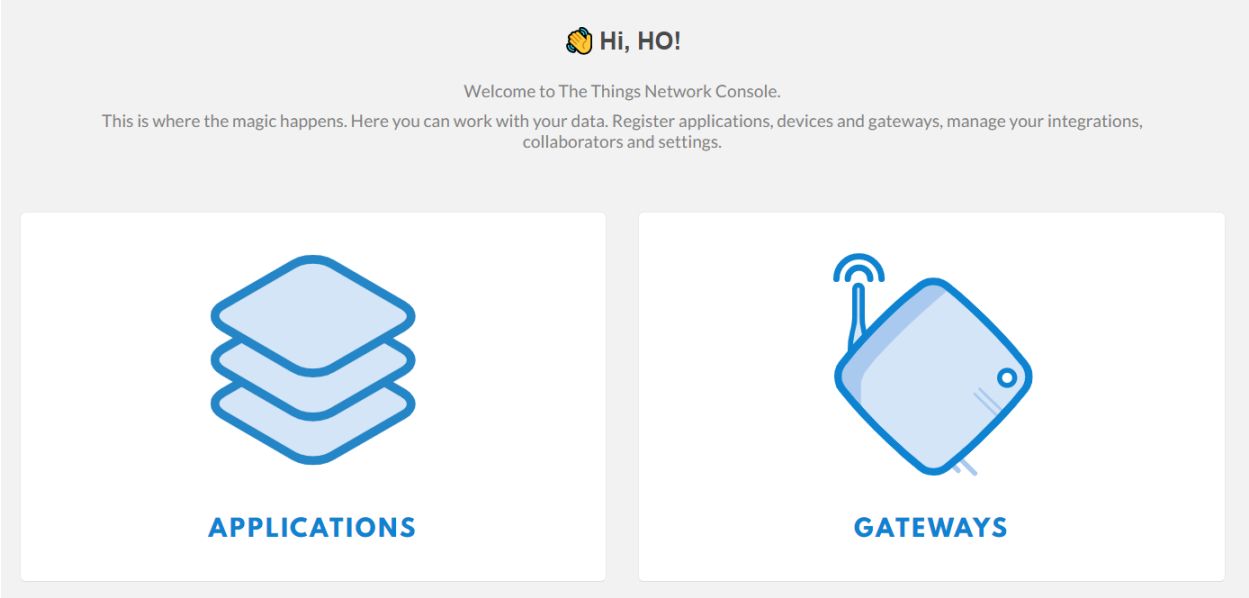
  - Install ver **1.6.21** on Boards Manager (the higher versions have the bugs for LMiC library)



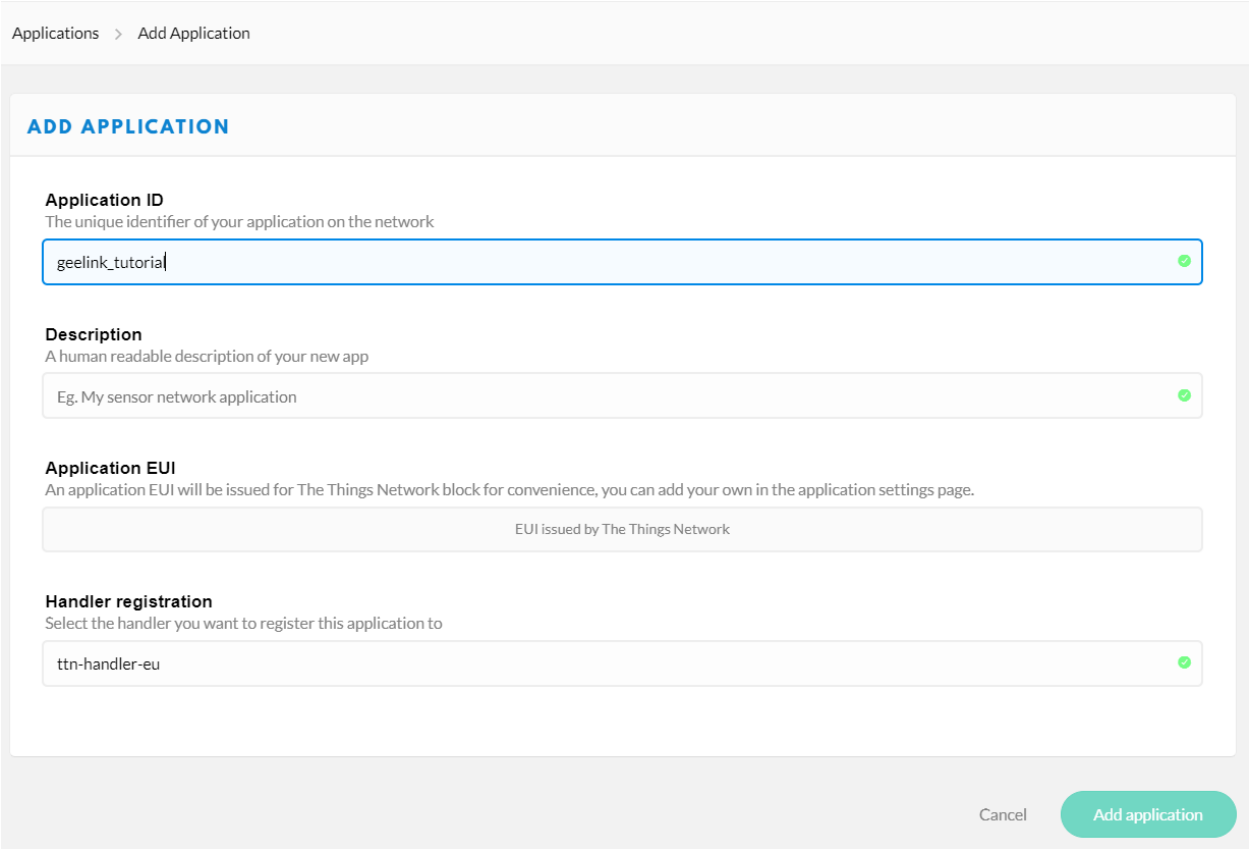
  - Select Processor: “ATmega328P (3.3V, 8MHz)” and Port that is connecting to ProMini
  - Install Libraries following this instruction: https://www.arduino.cc/en/guide/libraries

The Things Network (TTN)

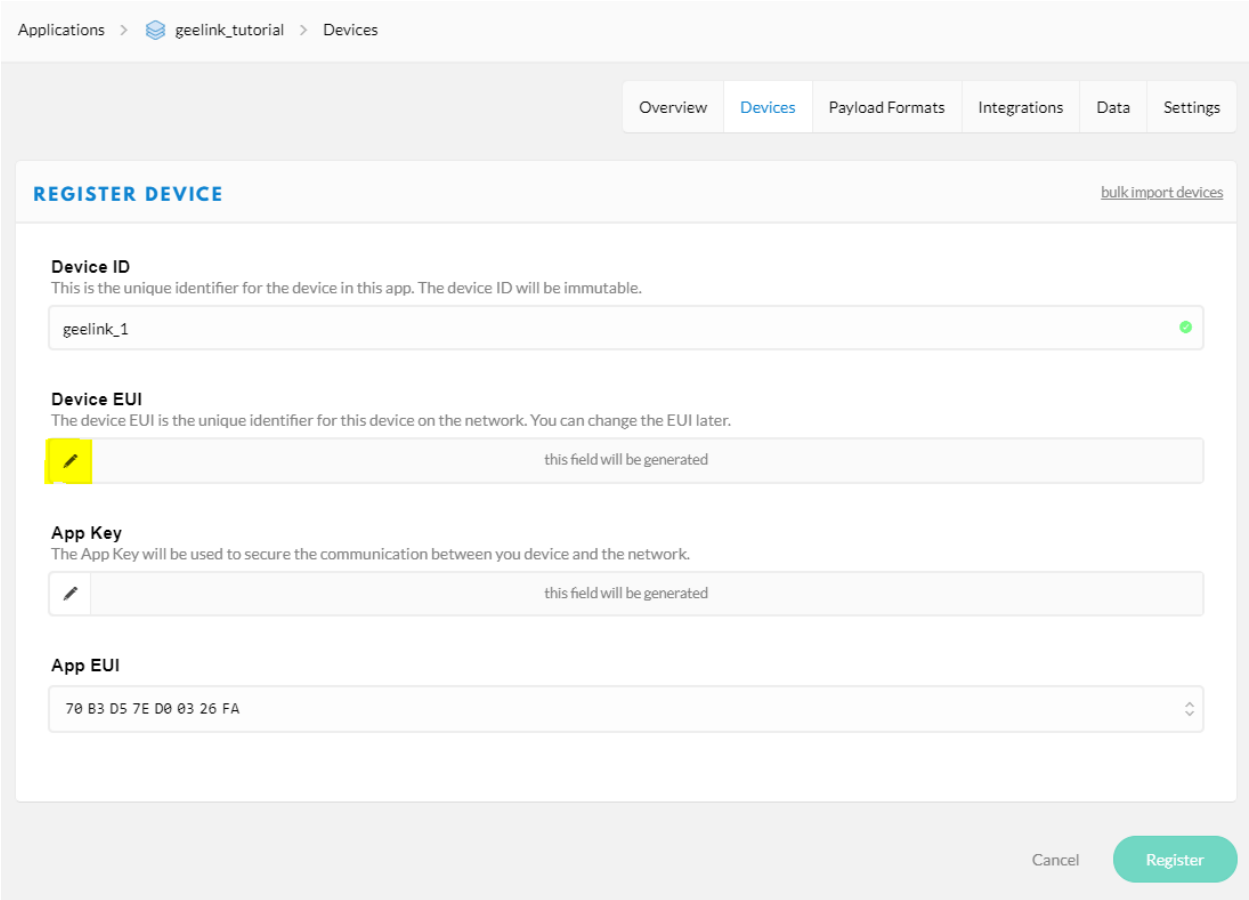
- Login to TTN console: https://console.thethingsnetwork.org/ and select Application



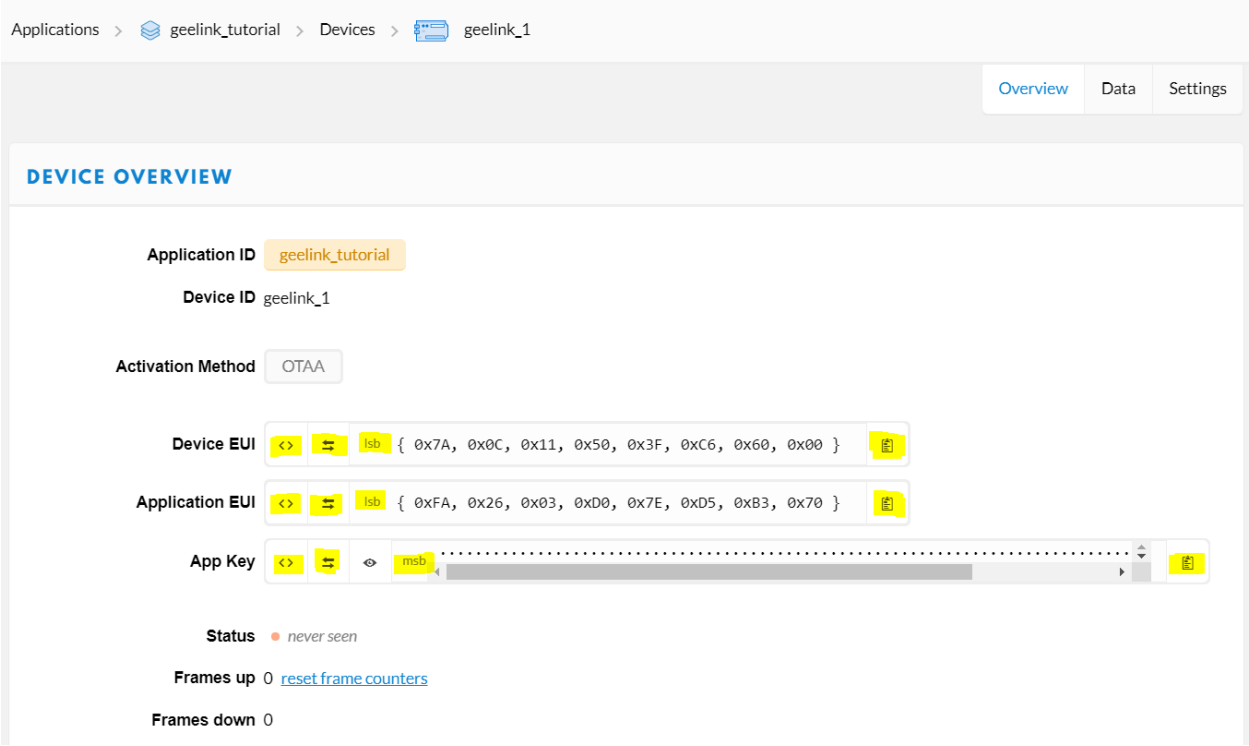
- Add a new Application



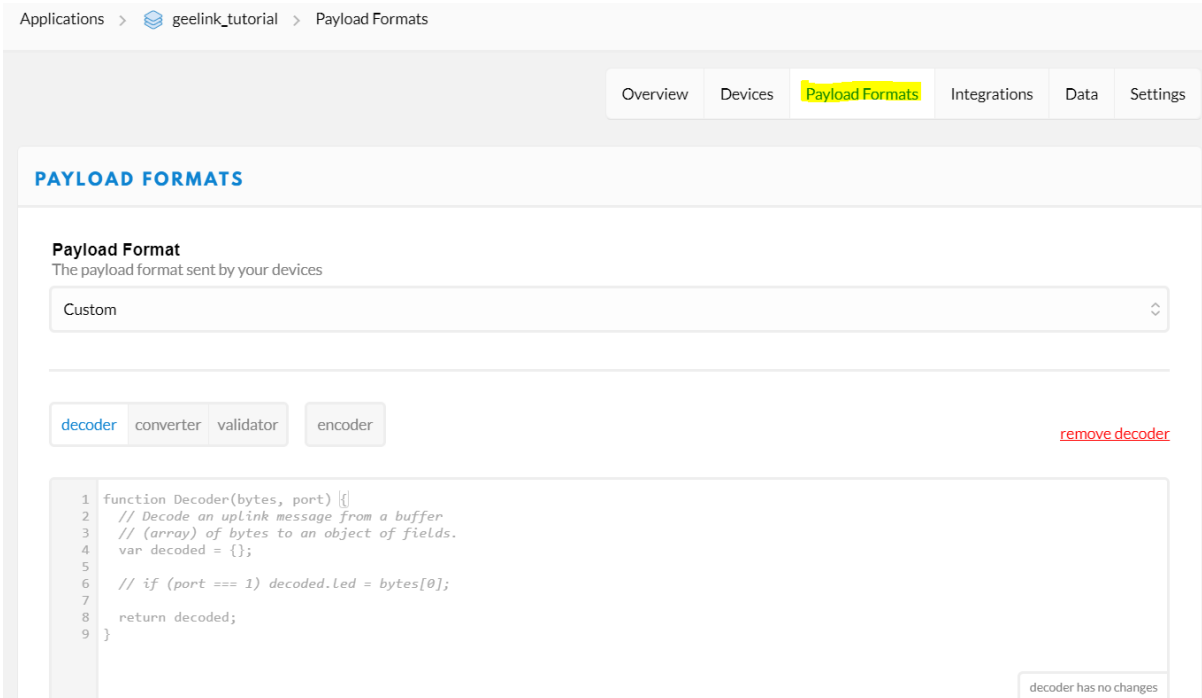
- Register a device



- After registering a device, adjust the keys into this format



Payload decoder on TTN

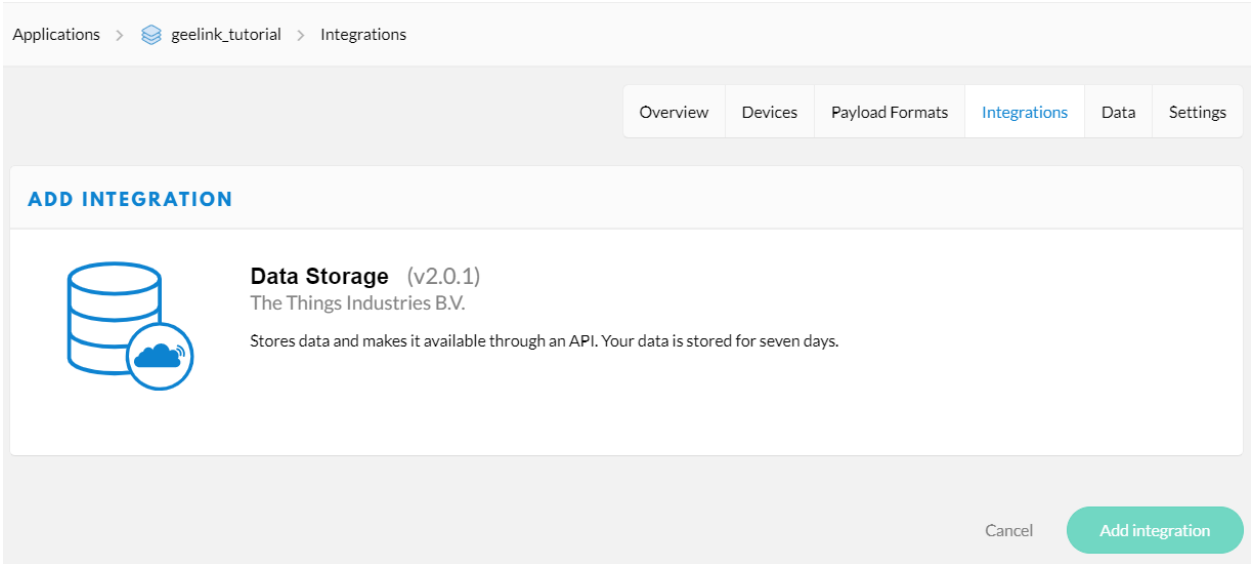


- Change the decoder by the GeeLink_decoder following this link and *remember* to **Save**
  https://github.com/hovuduybao/GeeLink/blob/master/GeeLink_Decoder_TTN/GeeLink_Decoder_TTN.js

- Useful Links:
  - https://github.com/myDevicesIoT/cayenne-docs/blob/master/docs/LORA.md
  - https://doc.rakwireless.com/rak7200-lora-tracker/analyzing-the-data-from-rak7200

Integrations

- Add this integration:



Application Server

- Install InfluxDB and Grafana following this instruction:

https://github.com/ITU-PITLab/public/blob/master/TheThingsNetwork%2Bnode-red%2Binfluxdb%2Bgrafana.md

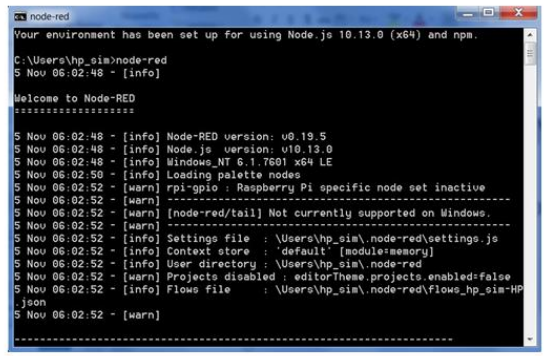
- Install Node-RED

If Git is not installed on your PC: https://git-scm.com/downloads After that, run this command on Gitbash

npm install -g --unsafe-perm node-red

Connecting to TTN

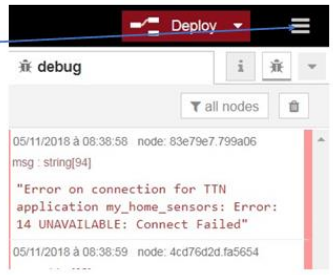
- Start NODE.js command prompt
- Run: node-red
- Open your web browser and go to http://127.0.0.1:1880



- On the editor, click here and go to palette editor

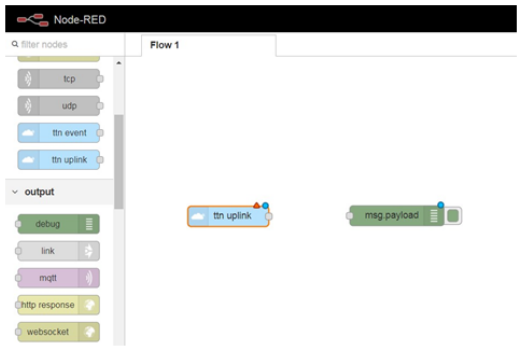
Install:

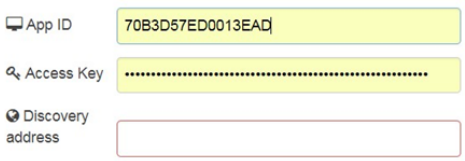
- node-red-contrib-ttn
- node-red-contrib-influxdb

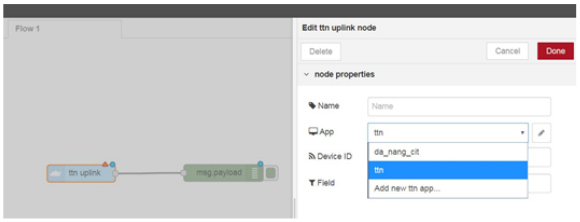


Connecting to TTN

- 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





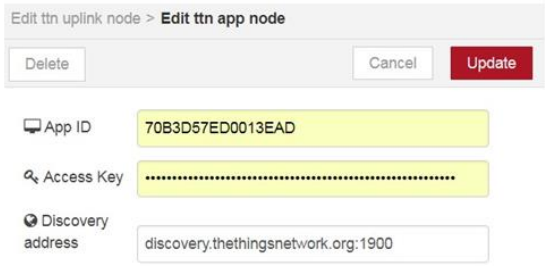


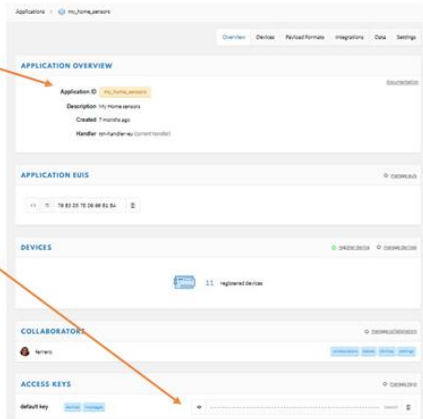
Connecting to TTN

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

discovery.thethingsnetwork.org:1900

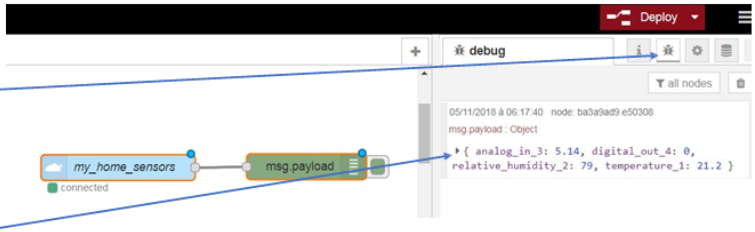
- Go to your application in TTN
- Copy past the Application ID and Access Key

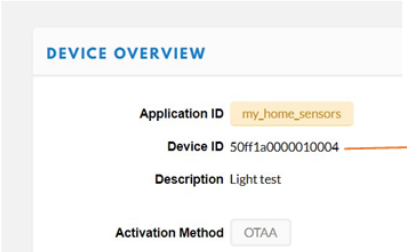


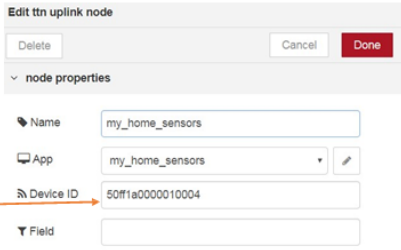


Connecting to TTN

- Click on Deploy
- Your 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:

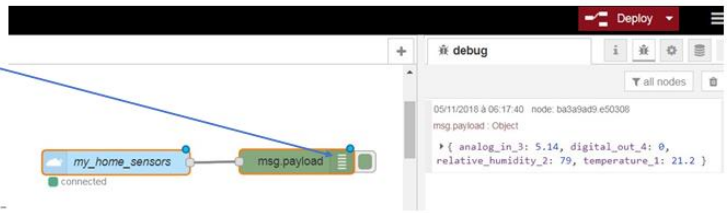


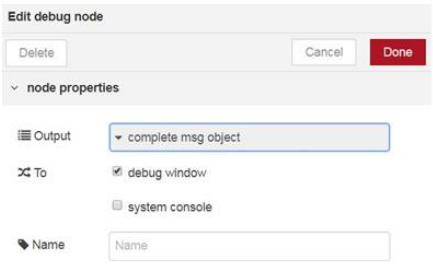




Connecting to TTN

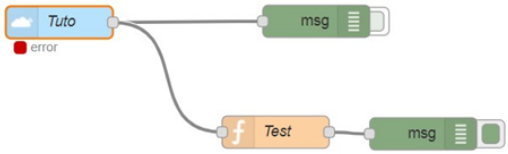
- Click here:
- Choose « complete msg object »
- And Deploy
- You have now more information of your uplink





Connecting to TTN

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



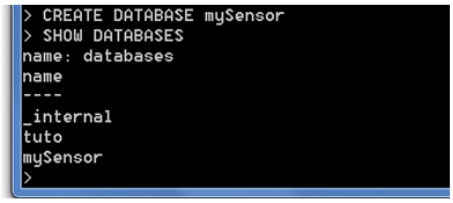

```
return {
 // Some fields from the metadata  freq:
msg.metadata.frequency,
cr: msg.metadata.cr,
dr: msg.metadata.dr,

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

};
```

InfluxDB

- Run « influxd.exe », it will start the database
- Run « influx.exe », it will open a shell
- Write: « CREATE DATABASE mySensor »
- Then write: « SHOW DATABASES »

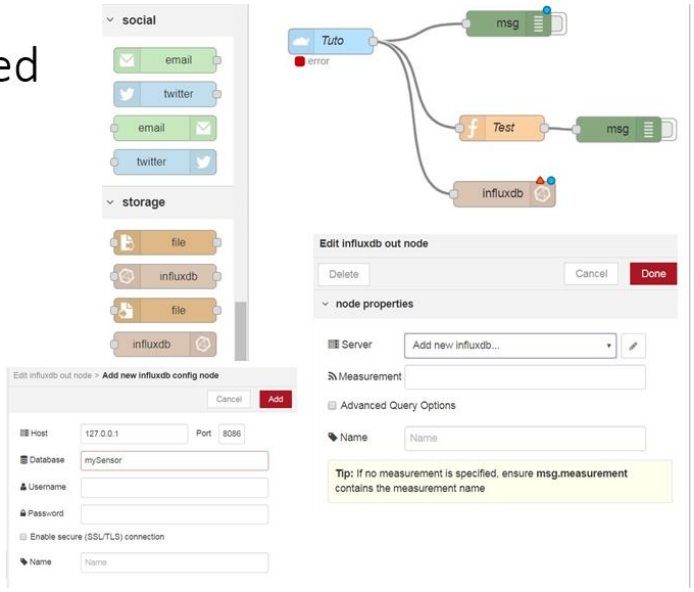


Your database is created

InfluxDB – Node Red

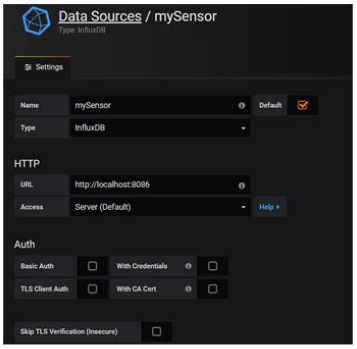
- How to store data in your database?
- Add an influxdb storage and connect it to your uplink
- Define a server, just add the Database name: mySensor
- Add
- In measurement field, add a name for your device: device1

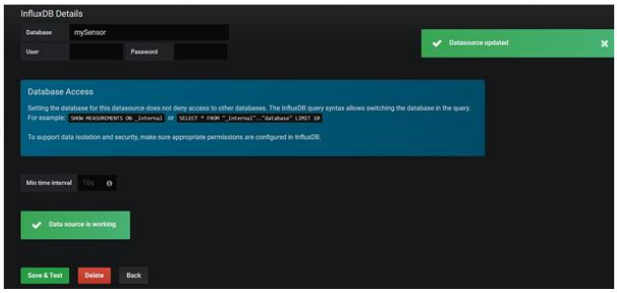
- Go to InfluxDB shell
- Run : SHOW SERIES ON mySensor



Grafana

- Go to yours unzip Grafana directory/bin
- Start grafana-server.exe
- Go to : http://127.0.0.1:3000
- User name and password is: admin
- Provide a new password
- Click « Add data source »
- Add a name
- Choose InfluxDB type
- Define Database name « mySensor »
- Click on Save and Test





Grafana

- Create a new dashboard
- Click on Graph
- Panel Title / Edit
- Select your data source and measurement, field temperature, time 1s, fill linear
- Change to the last 5mn
- Put your finger on the sensor
- Look at your curve