

Workshop location tracking using different techniques (WiFi and LoRaWAN)

Building a localisation node (using different technologies for localisation)

Identifying the components

Our bill of materials (Complete Workshop set available at [TinyTronics](#)) → zelf maken

- Wemos Lora PCB= RFM Module RFM95W (868 Mhz)
(<https://github.com/hallard/WeMos-Lora>)
- Attached BME280 sensor en of screen?
- Wemos D1 mini v3 – ESP8266
- Micro USB cable
- 3D print

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Starting up the Arduino Environment and login (HOMEWORK)

You may skip this step if you have:

- Arduino installed with ESP8266 support
- LMIC 'Charles' version installed
- Account made on The Things Network (please do this well before. There is a waiting period)
<https://thethingsnetwork.org>
- Account made on LoRaCLOUD (please do this well before. There is a waiting period)
<https://www.loracloud.com>
- Account made on FRED (for NODE-RED)
<https://fred.sensetecnic.com>
- Account made on HERE developers portal(for WiFi geolocation services)
<https://developer.here.com>

Arduino

Setup the Arduino Environment. To use the ESP8266 we need the ESP toolchain. Installation instructions can be found here: <https://github.com/esp8266/Arduino/>

Installing with Boards Manager

Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager. We have packages available for Windows, Mac OS, and Linux (32 and 64 bit).

- Install the current upstream Arduino IDE at the 1.8 level or later. The current version is at the [Arduino website](#).
 - Start Arduino and open Preferences window.
 - Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
 - Open Boards Manager from Tools > Board menu and install esp8266 platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).
- Close your Arduino IDE software. Remove (if already installed) the current LMIC library. We will use the 'Charles' version of the library. This one is compatible with your other projects, but has some special functionality needed for the WeMos LoRa board (supporting one input pin for the three DIO pins).

Download the new library and install the .zip file with library manager:

1. Goto <https://github.com/ch2i/arduino-lmic> or <https://github.com/turtlecrazy/WorkshopLocationWiFiLoRaWAN2020/>
2. Choose 'Clone or download'
3. Download ZIP
4. Mark the location
5. Go to Arduino IDE
6. Select 'Sketch->Include Library->Add .ZIP Library'
7. Select the downloaded Arduino-lmic-master.zip file from your download location

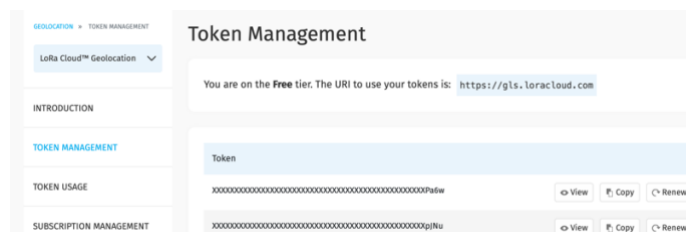
```

// #define CFG_us915 1
#define CFG_eu868 1
#define CFG_sx1276_radio 1

```


To use the Things network data dashboard, dashboard, you need a The Things Network account. You can create an account here: <https://account.thethingsnetwork.org/users/login> .

LoRaCloud



FRED is for running Node-red. You need this login to so some codeless programming. You can ask for a login at: <https://fred.sensetecnic.com> and make an node-red instance

HERE is for asking yourself where you are... (HERE):- You can apply for a login at: <https://developer.here.com>



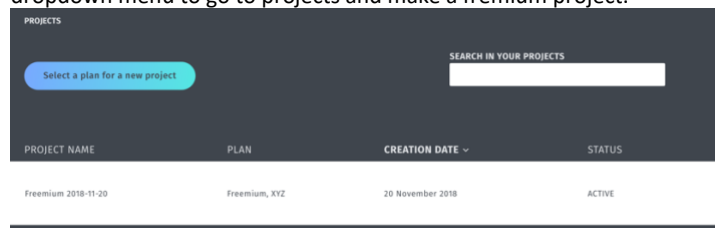
Name	Access Key ID	Password Reset	Last Used
root	AKIAI44QH8DHBVS7GALH	Yes	2017-07-20 10:00:00


root

Access Key ID: AKIAI44QH8DHBVS7GALH

Secret Access Key: [REDACTED]

Root user is the only user in the account.






REST

These application credentials can be used with [HERE Location Services REST API](#). Please read the [documentation](#) about authentication for details.

NEW


APP ID



API Keys

Create API key

There is a maximum of two keys per app. 1 of 2 keys created.

API KEY	CREATED	STATUS
	15 January 2020	Enabled

WORKSHOP START

Introduction



In this workshop we will show you all the steps how you will get from sensor (with microprocessor) to display results in a online dashboard.

This workshop will use Node-red as the dashboard source. With integrations of The Things Network there are however a lot more possibilities of visualizing and analysing your sensor data.

First we will start at configuring the sensor.

Sensor configuration

If you have done your homework you can easily download the sketch I prepared for you at <https://github.com/turtlecrazy/WorkshopLocationWiFiLoRaWAN2020/blob/master/CrazyTurtleTrackerUpload.ino>

As you will see in the sketch, you will need to fill in keys in order for the sensor to communicate with the things network.

For this you need to make an application and add devices to it on the things network account.

The Things Network Dashboard



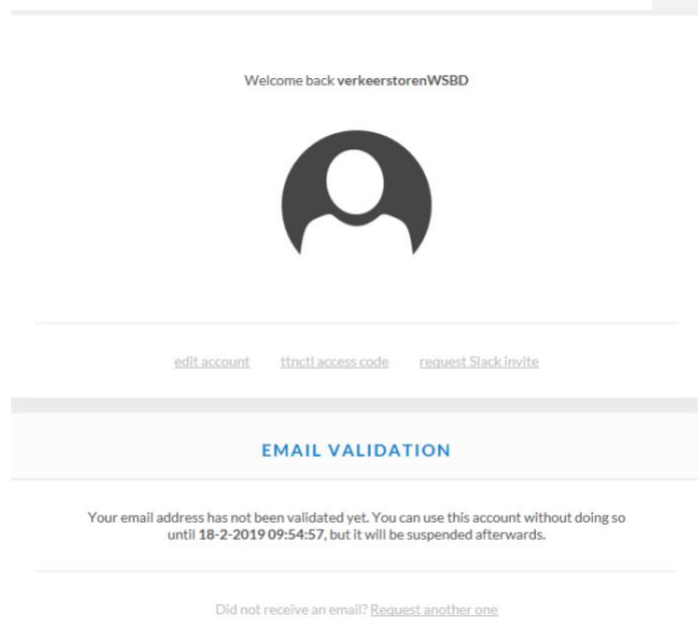
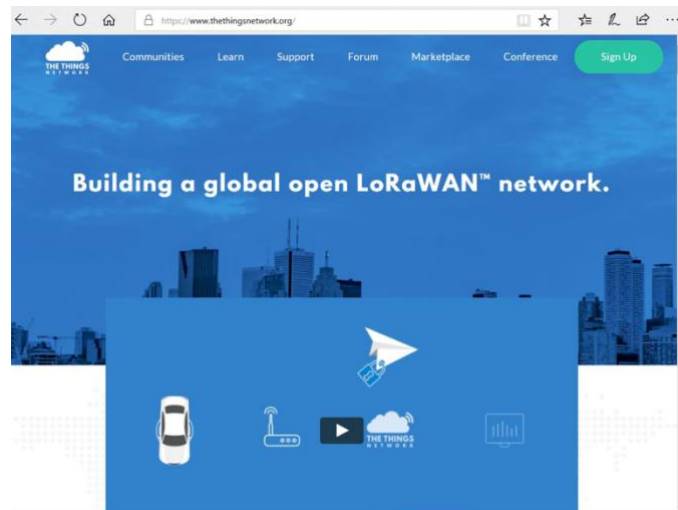
In order for the nodes to be able to communicate with the things network backend via the gateways. You need to have an account on the things network to be able to add devices. If you did your homework you are well prepared and have the highest chance of success rate to finish a complete app this evening.

Your applications and devices can be managed by The Things Network Dashboard.

Create an Account

To use the dashboard, you need a The Things Network account. You can create an account here <https://account.thethingsnetwork.org/users/login>.

After registering and validating your email address, you will be able to log in to The Things Network Dashboard.



Create an Application



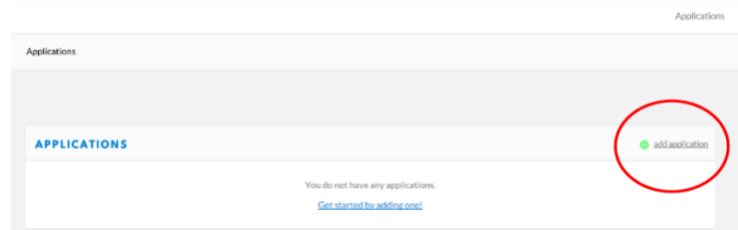
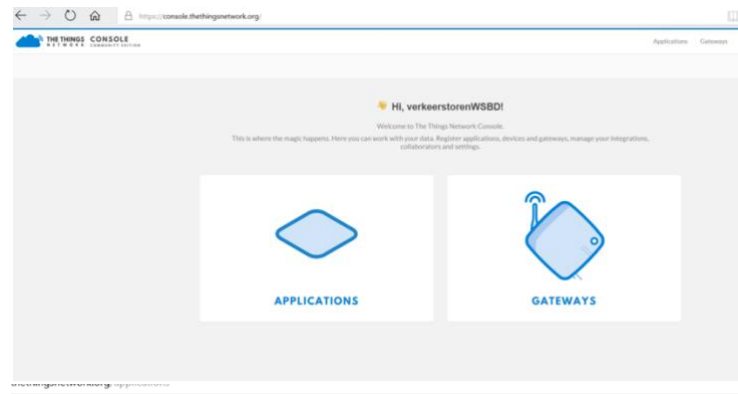
<https://console.thethingsnetwork.org/applications>

Choose 'add application'

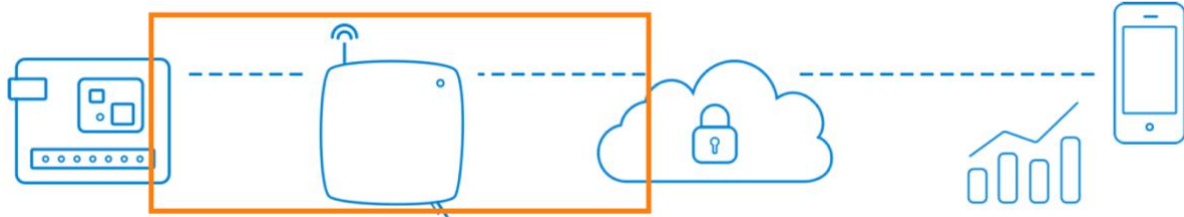
Give your Application a unique ID. You can use ONLY lowercase! You can add an unique number to get uniqueness over the ttn network (this is a global ID). If you already use an application you better

add another one, because of the specific Payload Function needed here.

Your description can be any description you like.

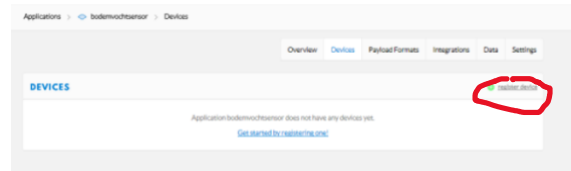


Registering a device (as an ABP Node)

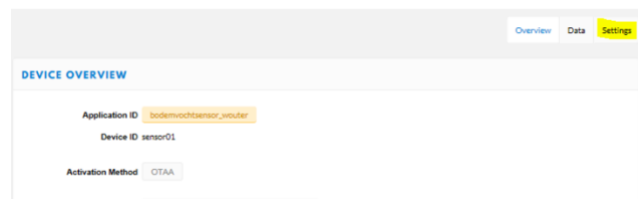
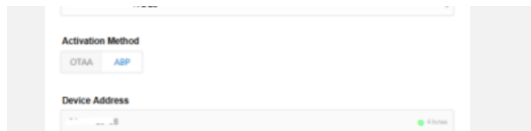


Activation by Personalization (ABP) is a method where the security keys are stored in the device. Not as safe as the OTAA method, but for experiments it works OK. There is no join procedure, nodes will work right away.

Choose Register the device.



Now edit the settings of the device and choose ABP (OTAA will be selected by default)



The addresses found can be added in the right format to the arduino sketch as stated earlier.

Device Address	<>	26 01 1A 32	hex	
Network Session Key	<>	{ 0x21, 0xEE, 0x1E, 0x77, 0x58, 0xBD, 0xFE, 0x47, 0x45, 0x34, 0x...	msb	
App Session Key	<>	{ 0xFE, 0xC6, 0xE8, 0x6E, 0x4D, 0x31, 0x35, 0x4D, 0xA5, 0x65, 0x...	msb	

Figure 1: TTN console ABP device keys

- Copy the Device Address as a HEX value to DEVADDR in the example, so 26 01 1A 32 will be 0x26011A32.
- Copy the Network Session Key as MSB to NWSKEY.
- Copy the App Session Key as MSB to APPSKEY.

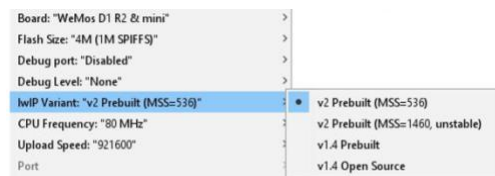
```
// LoRaWAN NwkSKey, network session key, AppSKey, application session key, end-device address
static const PROGMEM u1_t NWSKEY[16] = { 0x21, 0xEE, 0x1E, 0x77, 0x58, 0xBD, 0xFE, 0x47, 0x45, 0x34, 0x...
static const PROGMEM u1_t APPSKEY[16] = { 0xFE, 0xC6, 0xE8, 0x6E, 0x4D, 0x31, 0x35, 0x4D, 0xA5, 0x65, 0x...
// LoRaWAN end-device address (DevAddr)
static const u4_t DEVADDR = 0x26011A32 ; // <-- Change this address for every node!
```

Figure 2: Invoer in de Arduino sketch

In the Arduino IDE apply the right settings for the device:

Select the Wemos board in the Arduino IDE (no debug port, lwIP v2 Prebuilt, CPU 80 Mhz, Upload speed 921200)

Also check if the right comport is in use for the upload of the sketch.



After this: Compile and upload the code. Check in the dashboard the working.
You might want to uncheck the frame counter check.

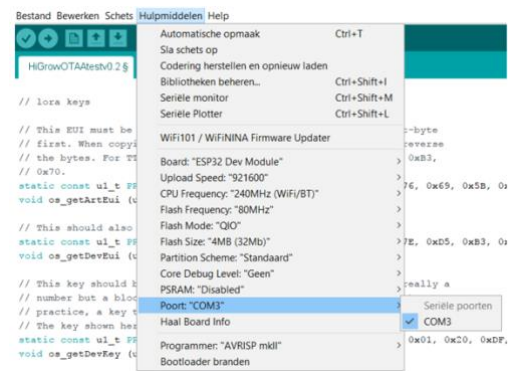
If the frame counter is checked, you must respect the sequence number, and probably copied packages are refused on the network. Though restarting you node, and thereby resetting the frame counter, will disable your node. So to get this working, you have to disable this check by unchecking this box.

Frame counter width

16 bit

32 bit

☒ Frame counter checks



Payload functions

Privacy first... We don't want people to know where we are ofcourse. It's level 2 privacy information. Luckily, The Things Network enrolled Stack V3 of the LoRa Alliance this year which adds to more secure data exchange. We will not go into this before everyone has a working dashboard.



To get the right display format, we can create a decoder in our application. Select your application and open the 'Payload Functions':

And overwrite the payload function with the text to be found here:

PAYLOAD FUNCTIONS

decoder converter validator encoder

```
1 function Decoder(bytes, port) {
2   // Decode an uplink message from a buffer
3   // (array) of bytes to an object of fields.
4   var decoded = {};
5
6   // if (port == 1) decoded.led = bytes[0];
7
8   return decoded;
9 }
```

<https://github.com/turtlecrazy/WorkshopLocationWiFiLoRaWAN2020/blob/master/payloadfunction.rtf>

decoder converter validator encoder

```
1 var hexChar = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E", "F"];
2
3 function byteToHex(b) {
4   return hexChar[(b >> 4) & 0x0F] + hexChar[b & 0x0F];
5 }
6
7 function hexToInt(hex) {
8   var num=hex;
9   if (num>0x7F) {
10    num=num-0x100;
11 }
12 return num;
```

First test the function with two dummy bytes CE D0, And then Save the function. Return to your data and look what happens:

APPLICATION DATA

II pause

cte

Filters

uplink downlink activation ack error

	time	counter	port			
▲	22:12:14	11	1	dev id: wemos_01	payload: 78 8A 20 4C 04 C8 D0 7A 8A 20 4C 04 C8 D0	macaddress.mac_1: "78:8A:20:4C:04:C8:D0:7A"
◀						
▲	22:11:06	10	1	dev id: wemos_01	payload: 78 8A 20 4C 04 C8 D0 7A 8A 20 4C 04 C8 CF	macaddress.mac_1: "78:8A:20:4C:04:C8:CF:7A"
◀						

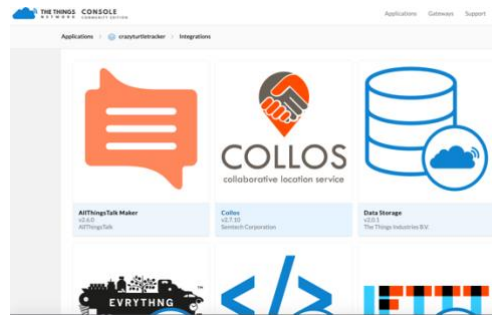
Application setup



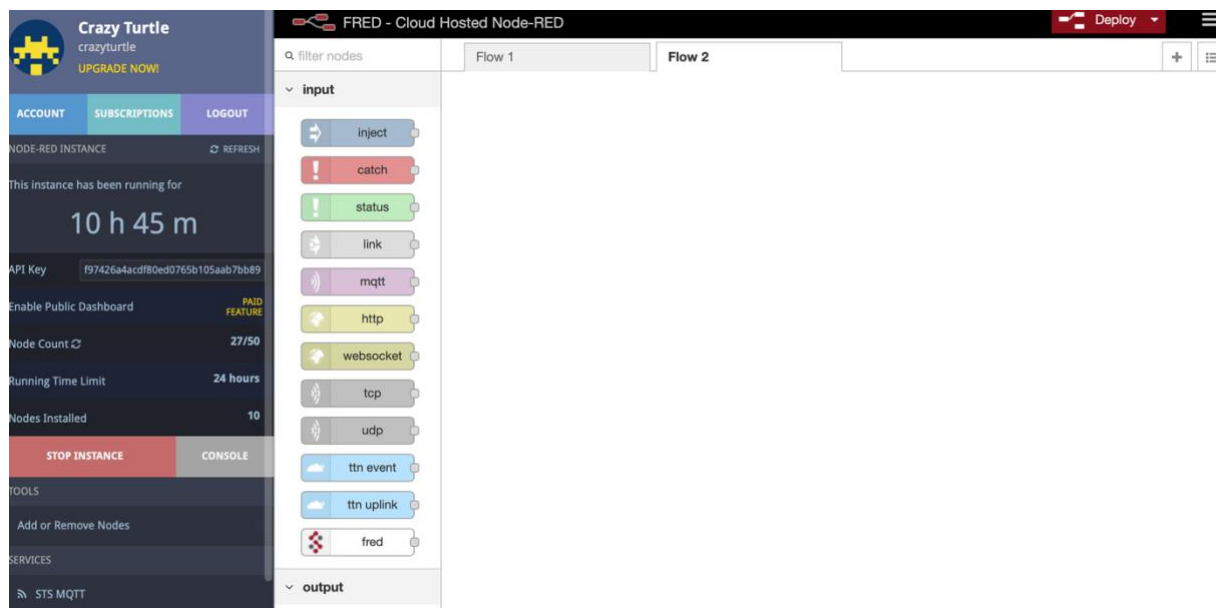
As stated earlier, with The Things Network, there are a lot of possibilities to add dashboard visualisations and analysis of the data via integrations.

In this workshop we chose Node-red. You can have a look yourself at your application integrations which possibilities there are.

We will not discuss possibilities in this workshop. In other workshops we might use different options.

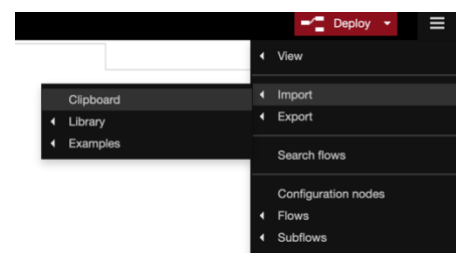


The node-red integrations we use via MQTT. A communication protocol for sensors. If you did your homework you already have a node-red instance received from FRED.



Via clicking on the laminated icon a dropdown menu opens with the possibilities to import a clipboard.

Select this option and import the tekst from this link

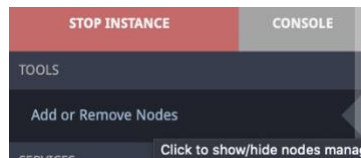


<https://github.com/turtlecrazy/WorkshopLocationWiFiLoRaWAN2020/blob/master/node-red.rtf>

You will now see the node-red no coding program to eventually plot the locations of the node on a map using both wifi (top “mindmap”) and lorawan (bottom “mindmap”).

It is possible some nodes are giving errors. This is because they are not installed by default by FRED.

You can do this via add or remove nodes



And search for the name of the node you are missing.

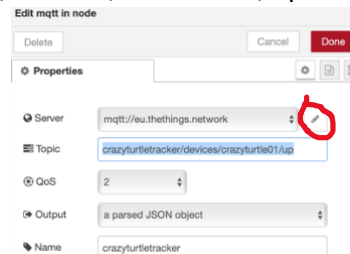
Also the credentials of the different services from The Things Network, HERE and LoRaCLOUD need to be applied for the “mindmaps” to be able to run.

If you click on the MQTT node a menu opens for you to fill in credentials.

First fill in the right topic. My topic for mqtt to retrieve data from is: crazyturtletracker/devices/crazyturtle01/up

Which consists of MYAPPLICATION/devices/MYDEVICE/up.

Please apply your own application and device name here.

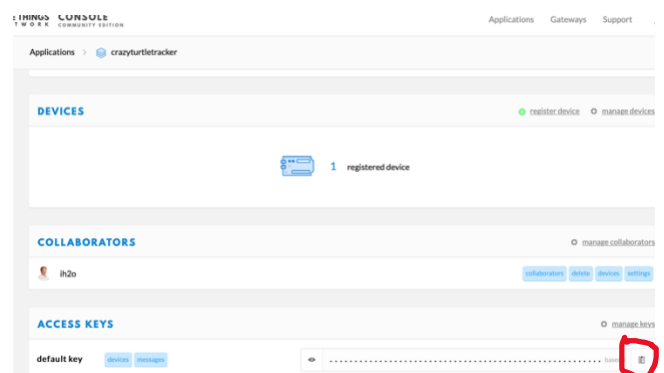
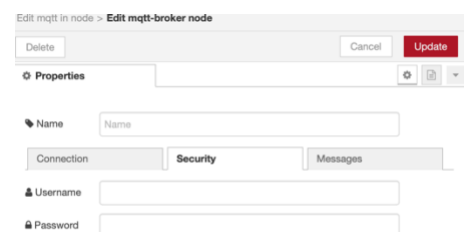
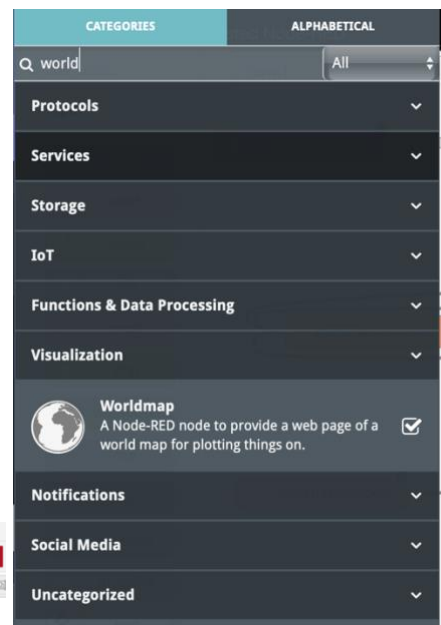
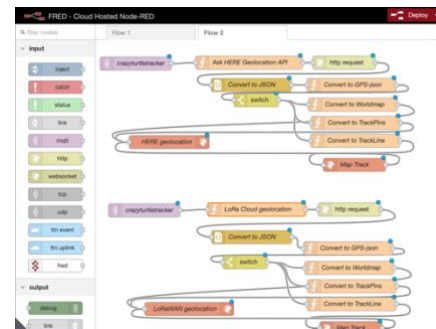


Of course you also need to apply your security settings. This can be done by clicking on the **potlood**. A new screen will open.

For username you can use your application name and the password consists of the access keys.

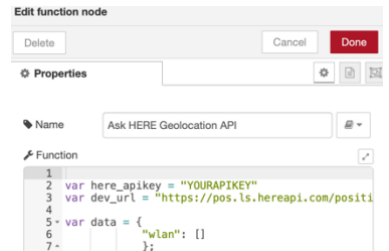
You can copy them from the console at your application on the bottom of the page.

Once you managed this and clicked on deploy. You will see the MQTT node displaying “connected”.



Now we connected The Things Network with MQTT. We still need to apply keys for both LoRaCLOUD and HERE for the location services to work.

You Obtained the HERE key as described in the homework assignment and can paste it at the "YOURAPIKEY" text of you open up the ASK HERE Geolocation API function.

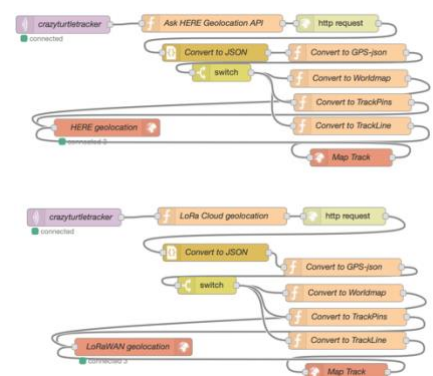
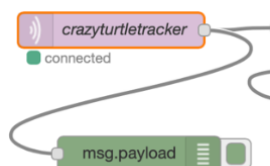
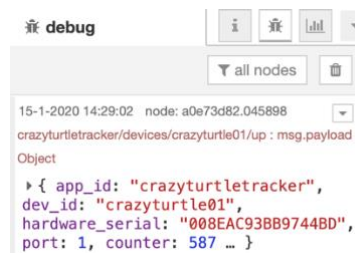


Same procedure applies to connecting to the LoRaCLOUD geolocation service. Here the keys are called tokens.

If all goes well you will see the MQTT nodes connected as well as the HERE geolocation and the LoRaWAN geolocation.

If not there are probably bugs involved.... Which need to be exterminated by using tools...

The debug node.. In the screen on the right plane with the bug you can find information for extermination



However if all goes well and you worked clean, there will be no bugs involved. If you click on either the LoRaWAN geolocation (worldmap nodes) or the HERE geolocation to open the nodes. And click: `%m - ctrl-shift-m`. You will see the maps displaying the nodes location using the different techniques. (Left a map with wifi locations and right with LoRaWAN locations)



P.s. you might have to have some patience before node-red starts showing the dots on the maps.

To do. Inrichten dashboard BME 280 en sensor aansluit etc