THE THINGS
NETWORK
ENSCHEDE

Build your own
weather station workshop!

18 apr 2018   TheThingsNetwork Enschede

Introduction

What do you need for this workshop ?

For this workshop you need the following :

- Laptop (or PC) with an up to date Arduino programming environments. You can find it here : https://www.arduino.cc/en/Main/Software
- A copy of the sources, which you can find here : https://github.com/ph2lb/ttnenschede_env_node
- The libraries used in the sources :
  - LMIC https://github.com/matthijskooijman/arduino-lmic
  - LowPower library https://github.com/rocketscream/Low-Power
  - MiniCore loader https://forum.arduino.cc/index.php?topic=412070 https://github.com/MCUdude/MiniCore
- A TheThingsNetwork account (login or create a new one) : https://account.thethingsnetwork.org/users/login
- If you have one, a 3.3V USB UART cable or adapter to program an Arduino Mini.

For more information go to the meetup page : https://www.meetup.com/The-Things-Network-Enschede/events/248921307/

What hardware did you receive?

The hardware consists of an assemble LoRa node with :

- Arduino Mini Pro with 1.8V BOD
- CH2I Mini LoRa PCB for I2C sensors
- RFM95W with a simple wire antenne
- BME280
- Battery holder for 2 AA batteries

This node is Low current* and battery powered.

** provided you remove the voltage regulator and power led. We have not done this yet because possibly not everyone wants this (and you can now see that it is on). See last page of this manual.*

The node Arduino source code

The software for the node is written in Arduino. Although the software is actually still beta (no warranty whatsoever), it has been working satisfactorily for a long time. A copy of the source code that you can find on : https://github.com/ph2lb/ttnenschede_env_node

Lets get hands on

Step 1 : Choices, choices, choices

Before we can start you have to decide what you want to do with the messages from the node. You can have these sent to the "ENV environment", to the Cayenne environment or even to your own environment.

The ENV environment is an open source platform developed by Lex Bolkesteijn and is currently hosted on a private server. This system has been published on the TheThingsNetwork forum (see : https://www.thethingsnetwork.org/forum/t/lora-bme280-environmental-node-with-webbased-backend/9264) [Dutch]

The Cayenne environment is an IoT platform that has been developed by MyDevices and integrates seamlessly with TheThingsNetwork. Many tutorials can be found on TheThingsNetwork. If you want to start with this, we advise you to follow the steps of this LAB: https://www.thethingsnetwork.org/docs/applications/cayenne/

In the Arduino source code 2 types of messages are defined, namely the message format for the "ENV environment" and for a "Cayenne environment". Do you want to define your own message? Of course you can, but for this workshop we assume that you choose one of the two pre-defined message formats.

What is your choice ?

[ ] ENV environment
[ ] Cayenne

Made a choice? Awesome, then let's start with the Arduino source code.

Step 2 : The node Arduino source code

As already mentioned, the Arduino source code can send 2 types of messages, namely the message format for the Env environment and for a Cayenne environment and if you have made a choice above which path you want to walk.

If you retrieve the source code from the above github then it is default set to the "ENV environment" and you can proceed to the next step.

If you have opted for the "Cayenne environment" then you have to look up the following line in the source code:

```
// #define USE_CAYENNE 1
```

And change it to :

```
#define USE_CAYENNE 1
```

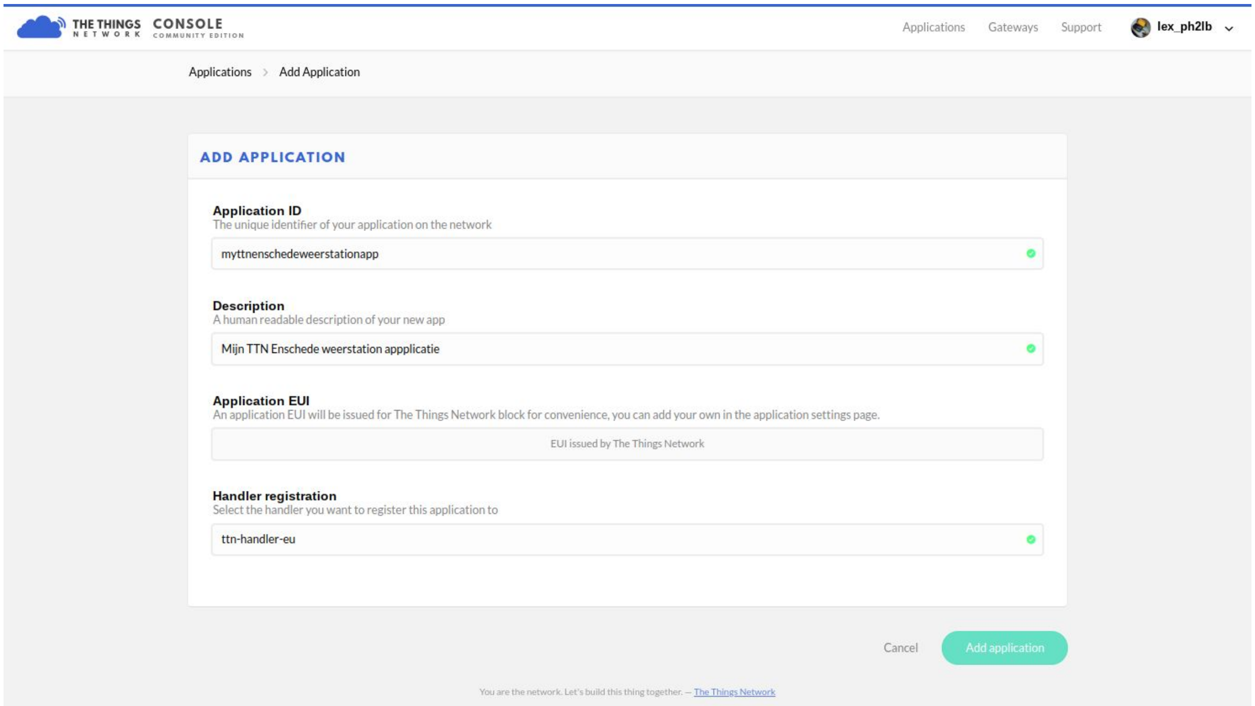
So leave out the //.

Step 3 : TTN Console

Before you can use the node you need to create an application in the TTN Console. You do this by logging in with your TheThingsNetwork account on: https://account.thethingsnetwork.org/users/login

Step 3.1 : create an application

Create an application by choosing Application - Add Application and fill in the requested data.


THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > Add Application

ADD APPLICATION

Application ID
The unique identifier of your application on the network
myttnenschedeweerstationapp

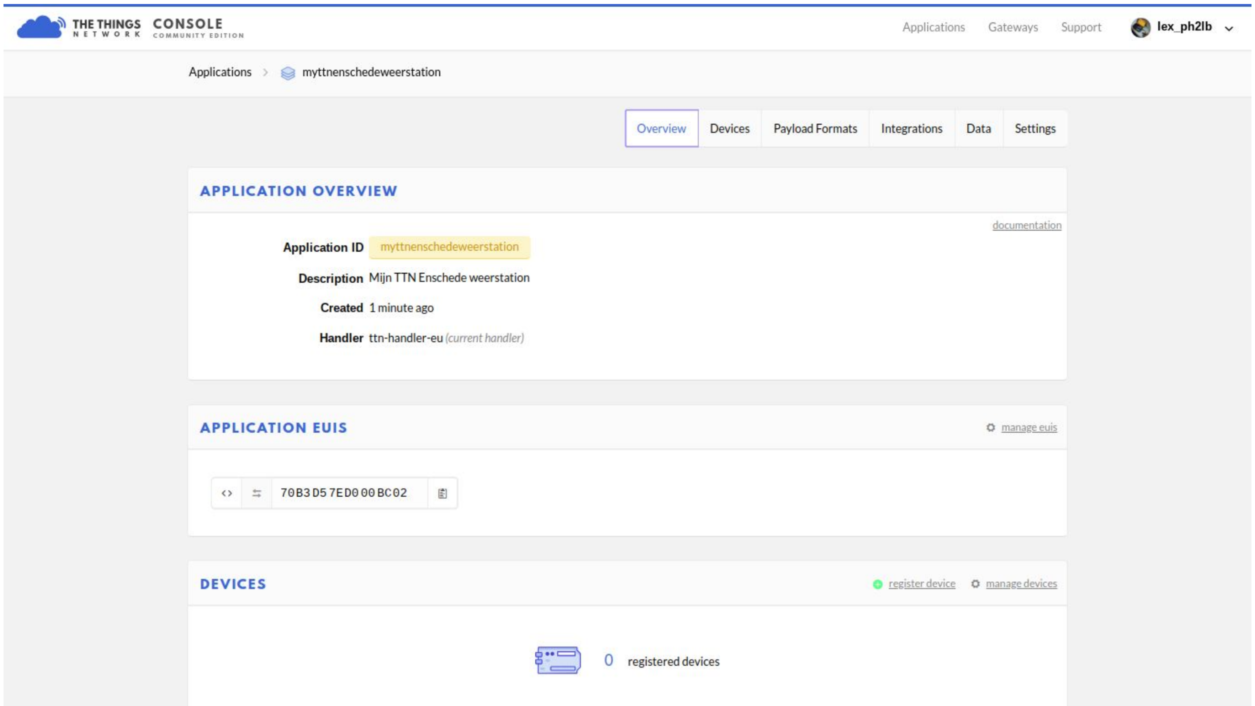
Description
A human readable description of your new app
Mijn TTN Enschede weerstation appplicatie

Application EUI
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.
EUI issued by The Things Network

Handler registration
Select the handler you want to register this application to
ttn-handler-eu

Cancel
Add application
You are the network. Let's build this thing together. – The Things Network

And click on [Add Application]. You will then enter the Application Overview screen again.



APPLICATION OVERVIEW

Application ID myttnenschedeweerstation

Description Mijn TTN Enschede weerstation

Created 1 minute ago

Handler ttn-handler-eu *(current handler)*

APPLICATION EUIS

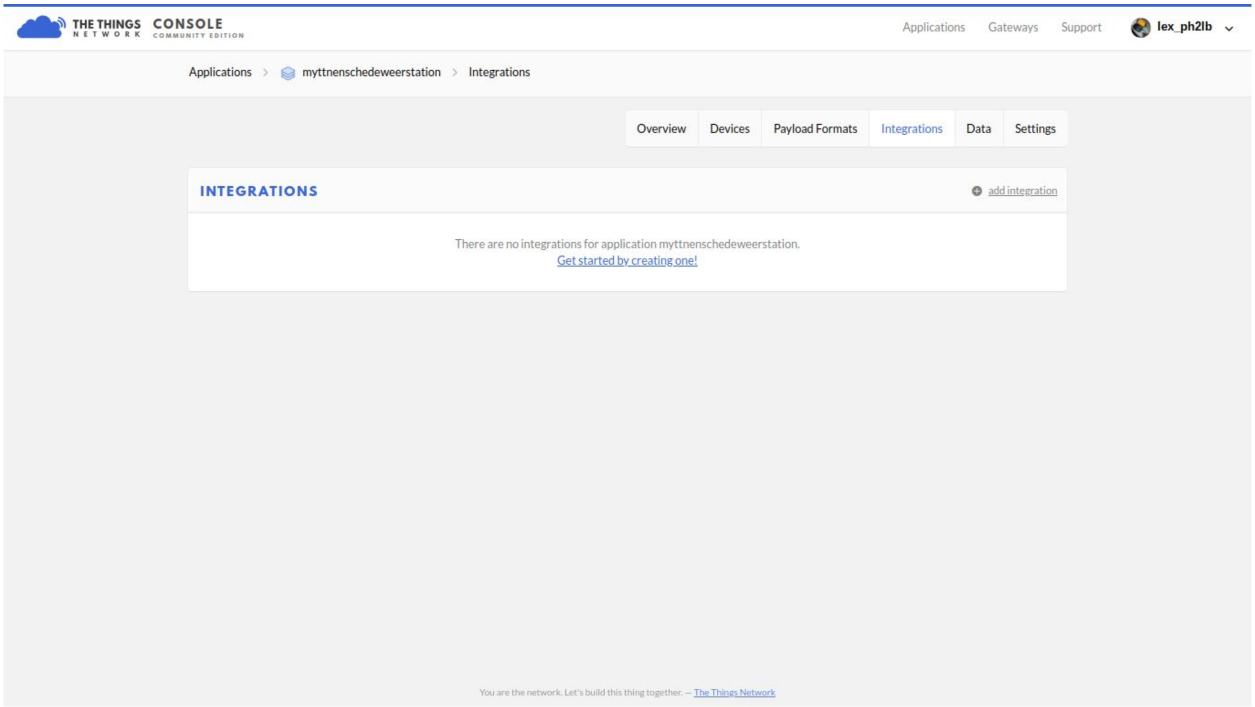
70B3D57ED000BC02

DEVICES

0 registered devices

Step 3.2 : integration configuration (note: the choice you made above is important).

Click on the tab [Integrations]



There are no integrations for application myttnenschedeweerstation.
Get started by creating one!

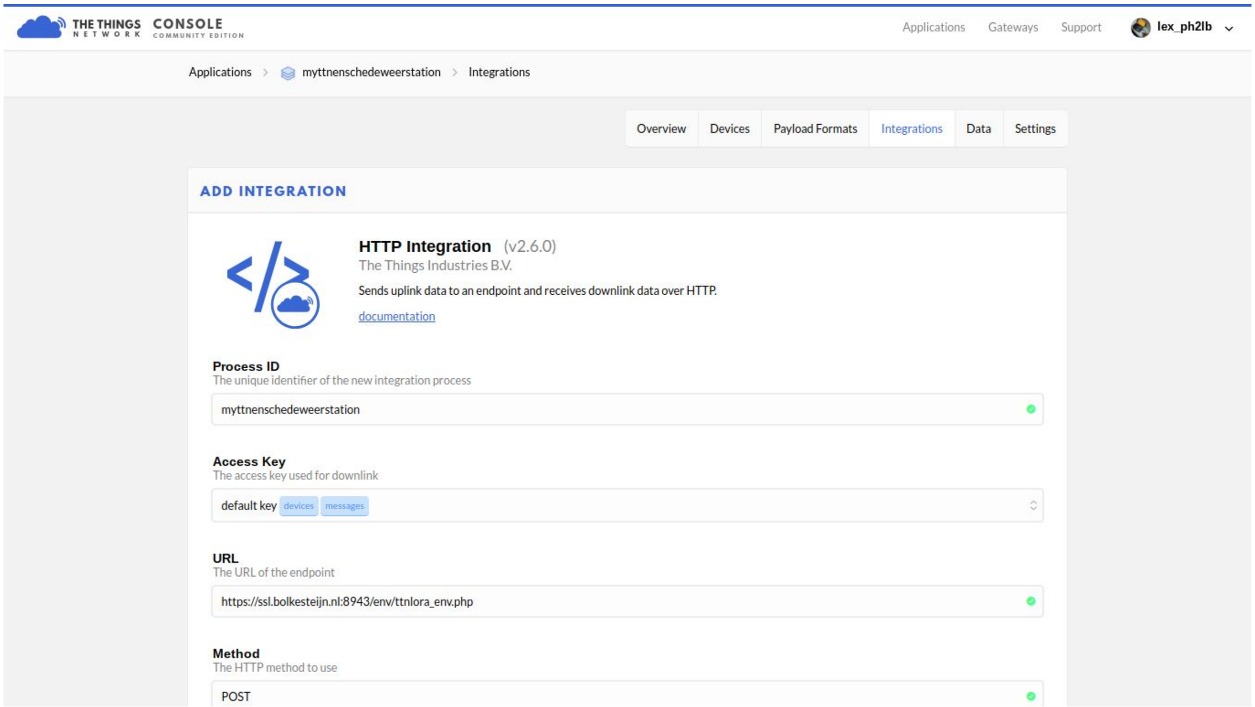
You are the network. Let's build this thing together. — The Things Network

Oh oh choice stress. Do you remember which area you had chosen?

If you have chosen the "ENV environment" go to step 3.2a.
If you have chosen the "Cayenne environment" go to step 3.2b.

Step 3.2a : I have chosen the “ENV environment”

Click on [Add Integration] and select the “HTTP Integration” from the list.
Select a Process ID for example “myttnenschedeweerstation”.
Acces Keys : “Default key”
URL : https://ssl.bolkesteijn.nl:8943/env/ttnlora_env.php
Methode : POST
Custom Header Name : api
Custom Header Value : 482a186d58b26613a4a0145d00b8a3ff


THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > myttnenschedeweerstation > Integrations
Overview
Devices
Payload Formats
Integrations
Data
Settings

ADD INTEGRATION

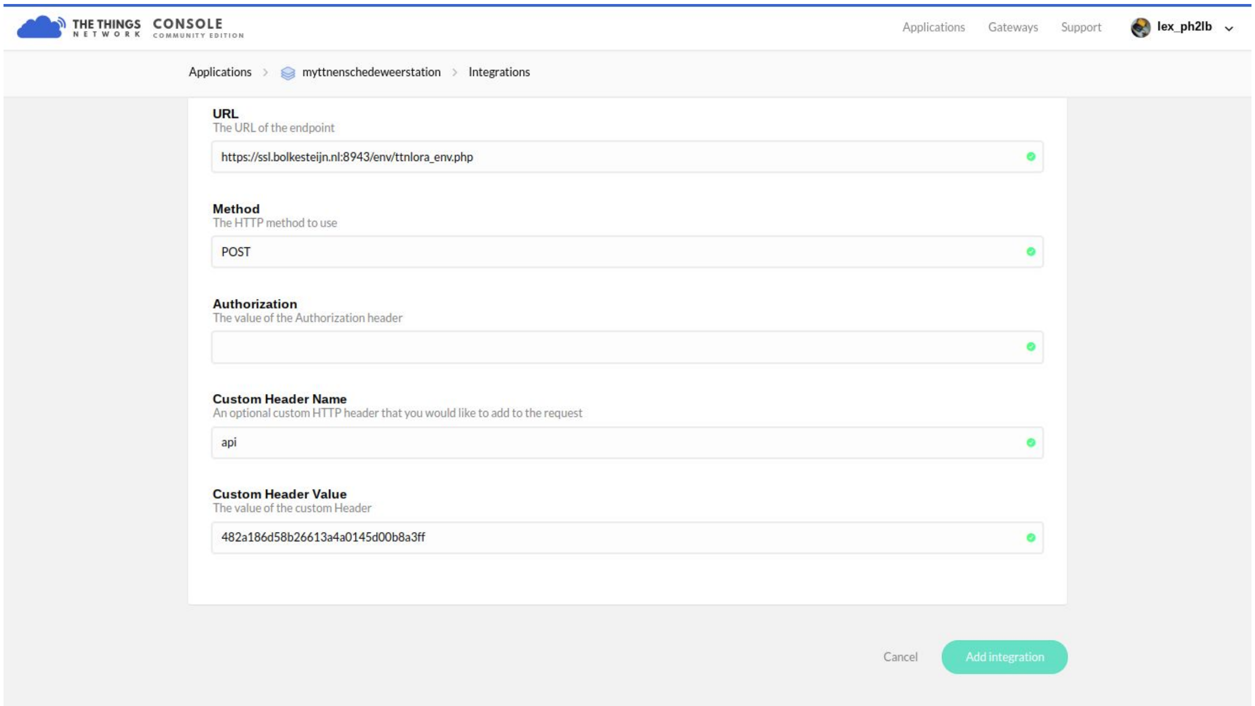
HTTP Integration (v2.6.0)
The Things Industries B.V.
Sends uplink data to an endpoint and receives downlink data over HTTP.
documentation

Process ID
The unique identifier of the new integration process
myttnenschedeweerstation

Access Key
The access key used for downlink
default key devices messages

URL
The URL of the endpoint
https://ssl.bolkesteijn.nl:8943/env/ttnlora_env.php

Method
The HTTP method to use
POST


THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > myttnenschedeweerstation > Integrations

URL
The URL of the endpoint
https://ssl.bolkesteijn.nl:8943/env/ttnlora_env.php

Method
The HTTP method to use
POST

Authorization
The value of the Authorization header

Custom Header Name
An optional custom HTTP header that you would like to add to the request
api

Custom Header Value
The value of the custom Header
482a186d58b26613a4a0145d00b8a3ff

Click on [Save]

Then click on the [Payload Functions] tab and select "Custom" and enter the function below for the "Decoder" (also see source code Arduino Software)

```
function Decoder(bytes, port)
```

```
{
  var retValue =   {
    bytes: bytes
  };

  retValue.batt = bytes[0] / 10.0;
  if (retValue.batt === 0)
     delete retValue.batt;

  if (bytes.length >= 2)
  {
    retValue.humidity = bytes[1];
    if (retValue.humidity === 0)
      delete retValue.humidity;
  }
  if (bytes.length >= 3)
  {
    retValue.temperature = (((bytes[2] << 8) | bytes[3]) / 10.0) - 40.0;
  }
  if (bytes.length >= 5)
  {
    retValue.pressure = ((bytes[4] << 8) | bytes[5]);
    if (retValue.pressure === 0)
      delete retValue.pressure;
  }

  return retValue;
}
```

To test fill in “00 00 00 00 00 00 00 00 00 00 00 00 00 00 00” and click on [Test]

You should see the following result:

```
{
  "bytes": "AAAAAAAAAAAAAAAAAAAA",
  "temperature": -40
}
```

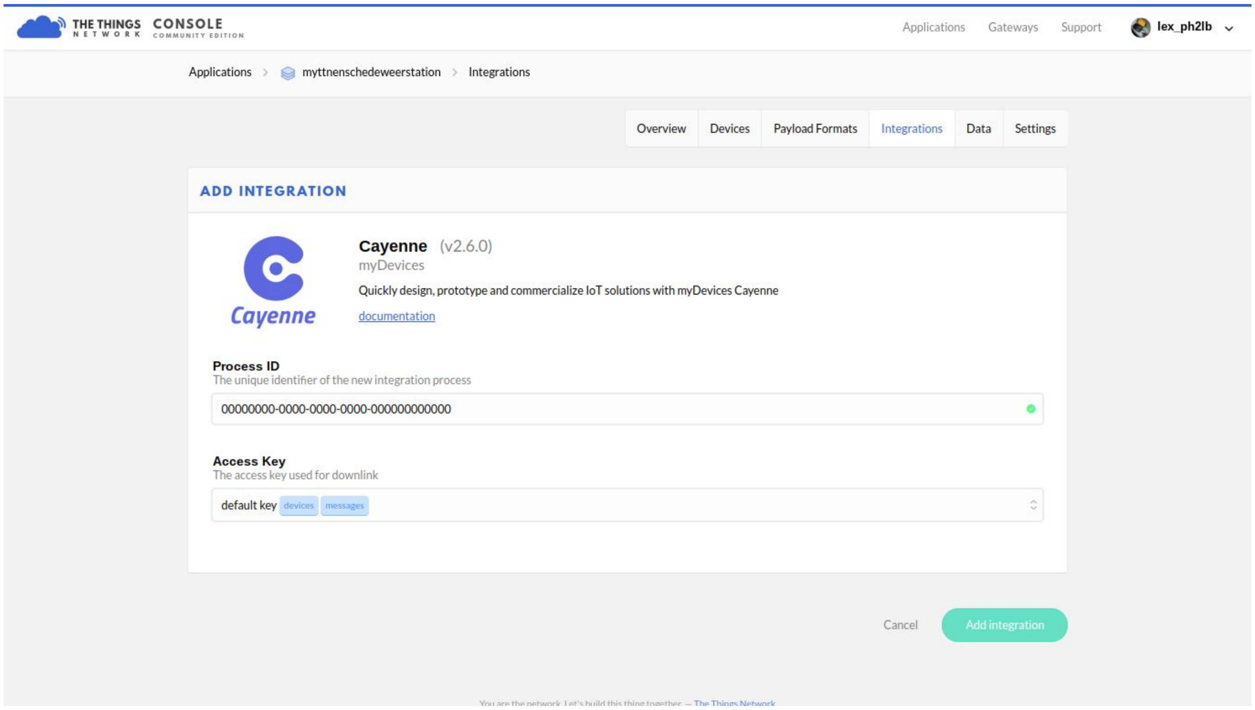
To get. Is this correct? Then click [Save Payload Functions]

Proceed to Step 3.3

Step 3.2b : I have chosen the “Cayene environment”

Click on [Add Integration] and chose “Cayenne” from the list.

This it the URL of your Cayenne dashboard : *https://cayenne.mydevices.com/cayenne/dashboard/lora/00000000-0000-0000-0000-000000000000 where you fill in your ProcessID on the 00000000-0000-0000-0000-000000000000 part*



ADD INTEGRATION

Cayenne (v2.6.0)
myDevices
Quickly design, prototype and commercialize IoT solutions with myDevices Cayenne
documentation

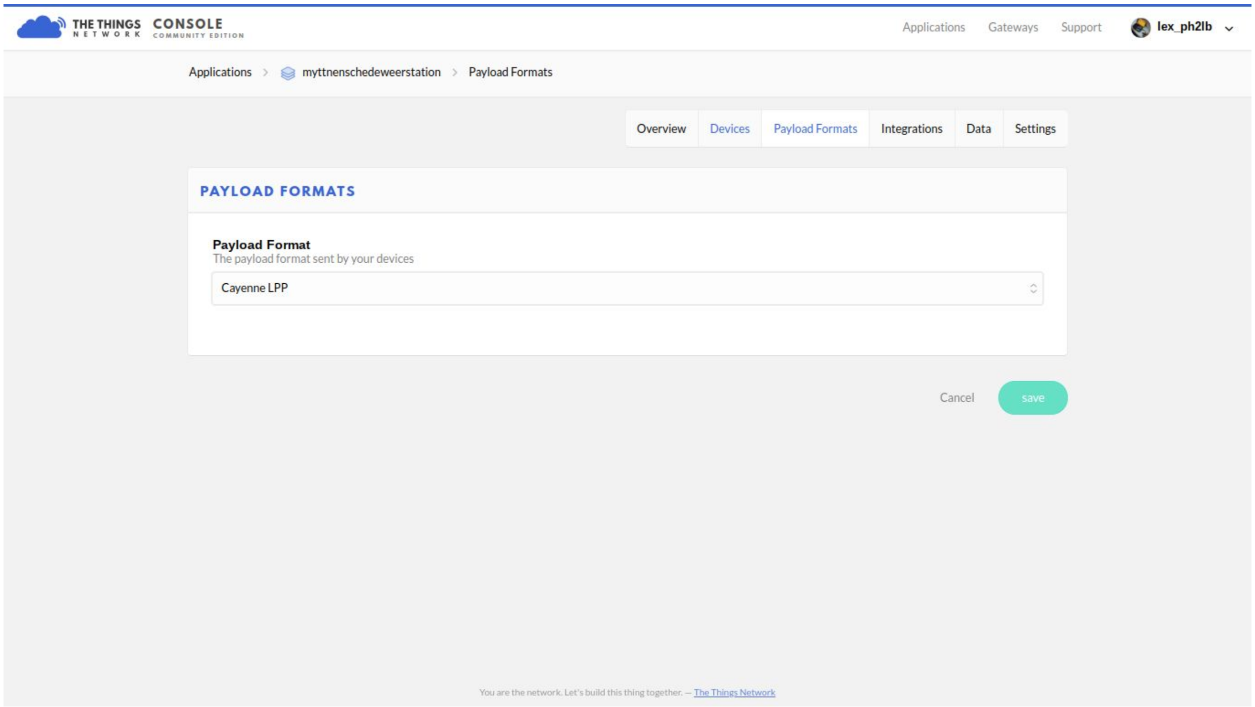
Process ID
The unique identifier of the new integration process
00000000-0000-0000-0000-000000000000

Access Key
The access key used for downlink
default key devices messages

Cancel
Add integration
You are the network. Let's build this thing together. — The Things Network

Select the default key for the access key and click on [Add Integration]

Then click on the [Payload Functions] tab and select "Cayenne LPP"



THE THINGS NETWORK
CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > myttnenschedeweerstation > Payload Formats
Overview
Devices
Payload Formats
Integrations
Data
Settings
PAYLOAD FORMATS

Payload Format
The payload format sent by your devices
Cayenne LPP

Cancel
save
You are the network. Let's build this thing together. — The Things Network

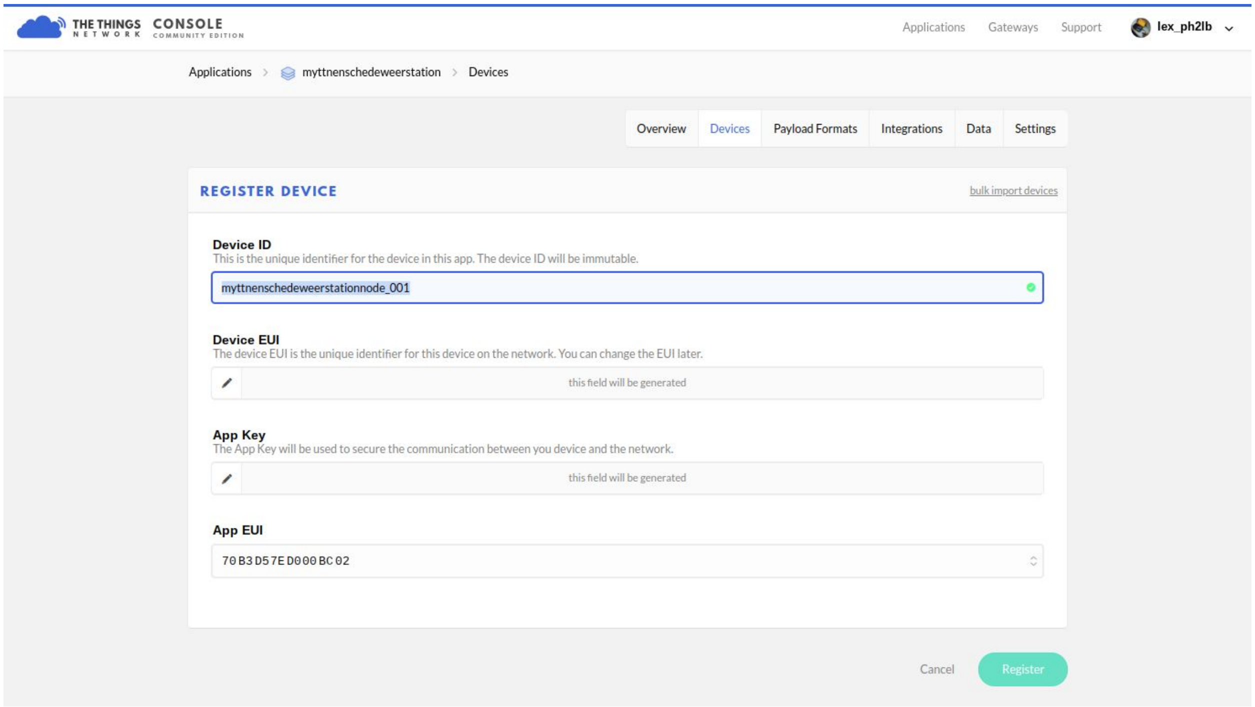
Click on [Save]

Go to Step 3.3

Step 3.3 : device registration

Click on the tab “Devices” and click on [Register Device].

Choose a sturdy name for the "Device ID" and click on the button next to the field "Device EUI" so that the text "this field will be generated" will be displayed.



REGISTER DEVICE

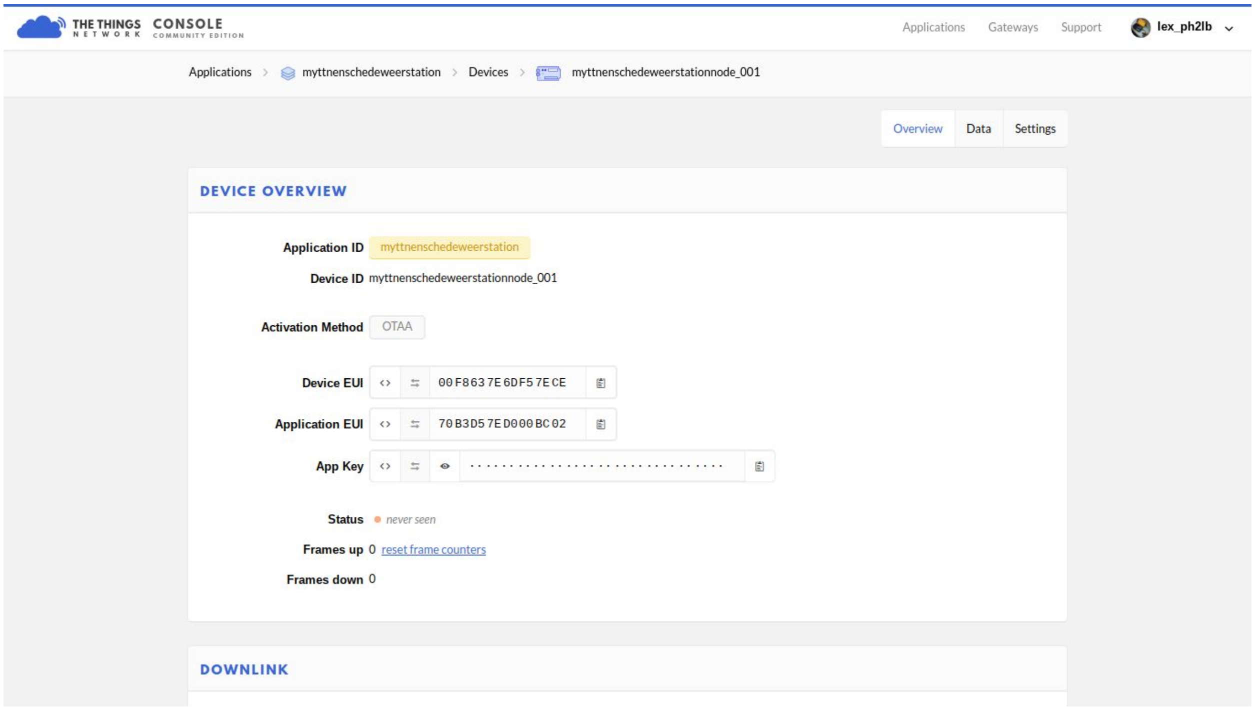
Device ID
This is the unique identifier for the device in this app. The device ID will be immutable.
myttnenschedeweerstationnode_001

Device EUI
The device EUI is the unique identifier for this device on the network. You can change the EUI later.
this field will be generated

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

App EUI
70 B3 D5 7E D0 00 BC 02

Then click on [Register] and you will see the screen below.


THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > myttnenschedeweerstation > Devices > myttnenschedeweerstationnode_001
Overview
Data
Settings

DEVICE OVERVIEW

Application ID myttnenschedeweerstation

Device ID myttnenschedeweerstationnode_001

Activation Method OTAA

Device EUI 00F8637E6DF57ECE

Application EUI 70B3D57ED000BC02

App Key ································

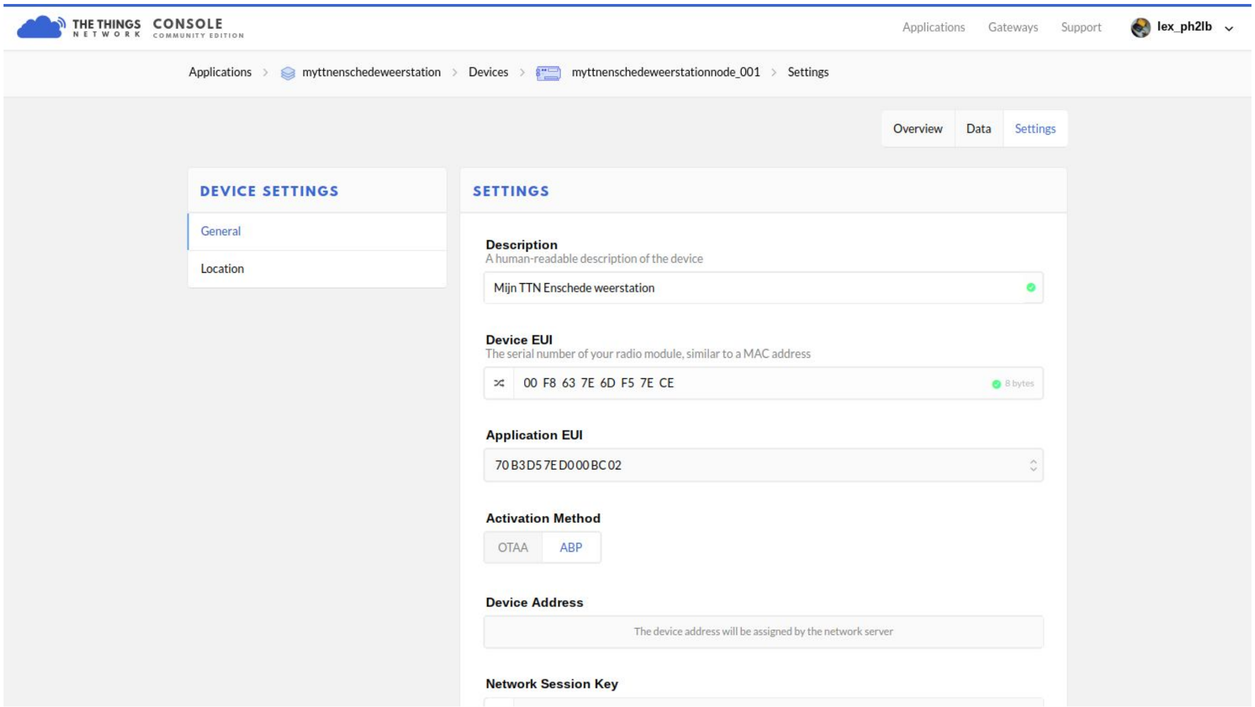
Status never seen

Frames up 0 reset frame counters

Frames down 0

DOWNLINK

But now he is still OTAA Activation Method and this workshop is based on the ABP Activation Method. We can change this in the tab "Settings". Click on the tab "Settings"


THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > myttnenschedeweerstation > Devices > myttnenschedeweerstationnode_001 > Settings
Overview
Data
Settings
DEVICE SETTINGS
General
Location

SETTINGS

Description
A human-readable description of the device

Mijn TTN Enschede weerstation

Device EUI
The serial number of your radio module, similar to a MAC address

00 F8 63 7E 6D F5 7E CE 8 bytes

Application EUI

70B3D57ED000BC02

Activation Method

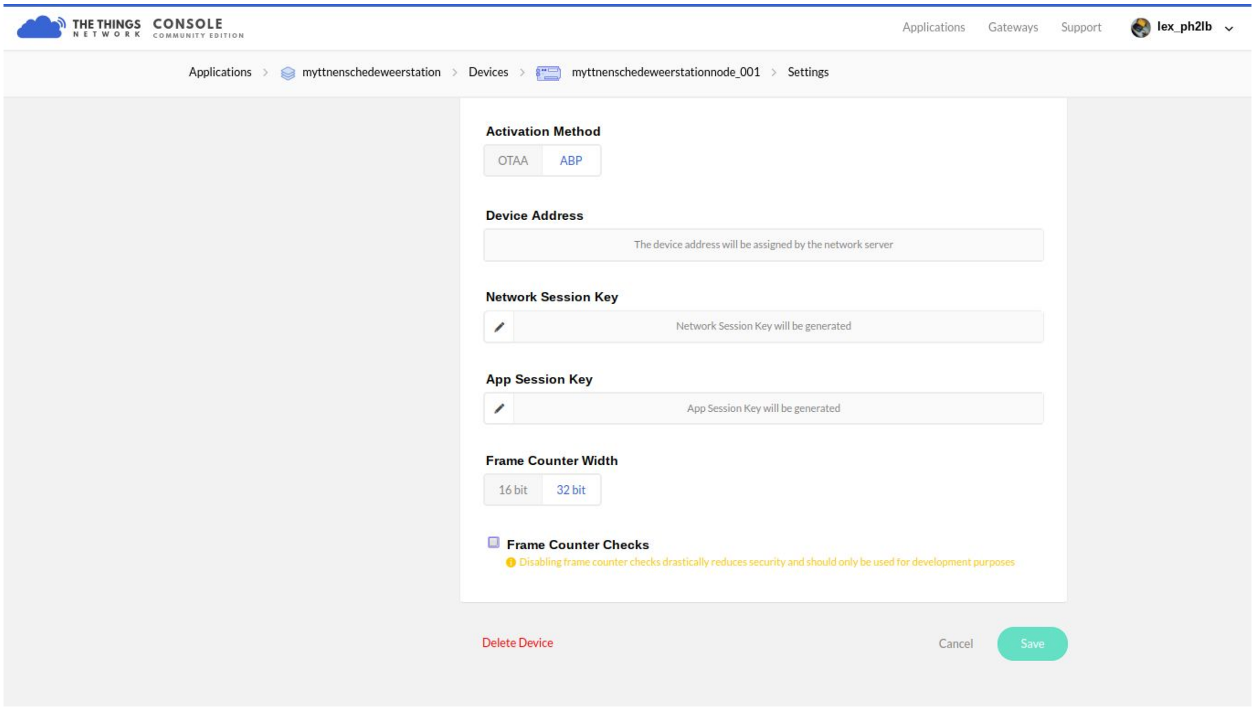
OTAA ABP

Device Address

The device address will be assigned by the network server

Network Session Key

Click ABP and if you are here enter a store description. Scroll down further and uncheck "Frame Counter Checks" (you may ignore the yellow warning).


THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > myttnenschedeweerstation > Devices > myttnenschedeweerstationnode_001 > Settings

Activation Method
OTAA | ABP

Device Address
The device address will be assigned by the network server

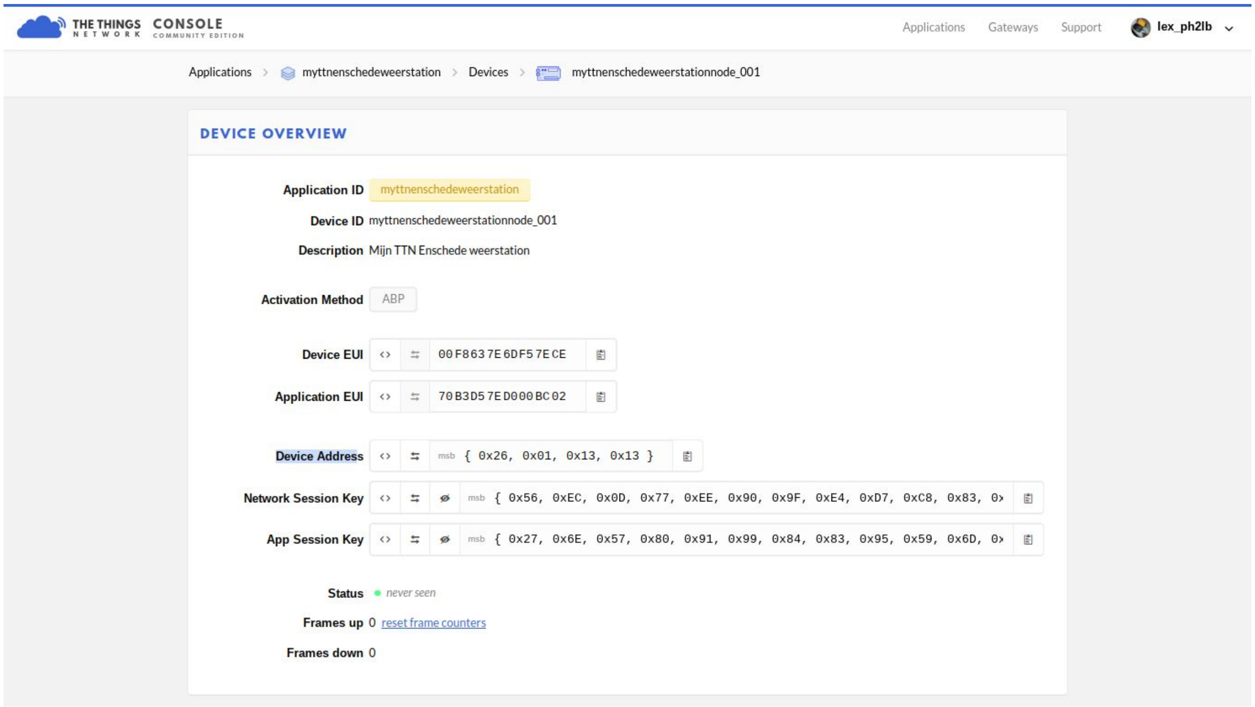
Network Session Key
Network Session Key will be generated

App Session Key
App Session Key will be generated

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

Then click Save. You will then return to the Device Overview screen.
Click on the eyes next to "Network Session Key" and "App Session Key" so that the keys are visible and then on the "<>" next to "Device Addres", "Network Session Key" and "App Session Key". The screen looks like this:



Applications > myttnenschedeweerstation > Devices > myttnenschedeweerstationnode_001

DEVICE OVERVIEW

Application ID myttnenschedeweerstation
Device ID myttnenschedeweerstationnode_001
Description Mijn TTN Enschede weerstation

Activation Method ABP

Device EUI 00 F8 63 7E 6D F5 7E CE
Application EUI 70 B3 D5 7E D0 00 BC 02

Device Address msb { 0x26, 0x01, 0x13, 0x13 }
Network Session Key msb { 0x56, 0xEC, 0x0D, 0x77, 0xEE, 0x90, 0x9F, 0xE4, 0xD7, 0xC8, 0x83, 0x
App Session Key msb { 0x27, 0x6E, 0x57, 0x80, 0x91, 0x99, 0x84, 0x83, 0x95, 0x59, 0x6D, 0x

Status never seen
Frames up 0 reset frame counters
Frames down 0

You now have the "Device Addres", "Network Session Key" and "App Session Key" visible that you need in your Arduino source code. You can easily copy these fields by clicking on the clipboard because you need them in the next step.

{ 0x26, 0x01, 0x13, 0x13 }

{ 0x56, 0xEC, 0x0D, 0x77, 0xEE, 0x90, 0x9F, 0xE4, 0xD7, 0xC8, 0x83, 0x32, 0x61, 0xEE, 0xFD, 0xDC }

{ 0x27, 0x6E, 0x57, 0x80, 0x91, 0x99, 0x84, 0x83, 0x95, 0x59, 0x6D, 0xEC, 0x5E, 0xA1, 0x16, 0xC1 }

Checklist. Do you have the following :

[ ] Network session key
[ ] Application session key
[ ] Device Address

Step 4 : The node Arduino source code part 2

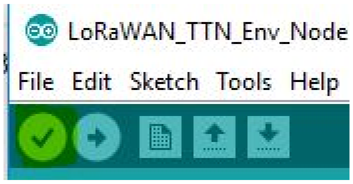
In order to use and identify a LoRa node in TheThingsNetwork, you have created the "Network session key", "Application session key" and the "Device Address" in the previous step and if it has been copied correctly.

Now the "Network session key", "Application session key" and the "Device Address" must be included in the arduino source code. Search the keys below for the source code and replace them with your keys that you have made in Step 3.3.

```
// Keys for ttn_env_test_001

static const PROGMEM u1_t NWKSKEY[16] = { 0xEC, 0x88, 0x17, 0x61, 0xFB, 0x51, 0xB7, 0x27,
0x2D, 0xEE, 0x47, 0xCE, 0xE8, 0xC8, 0x6C, 0xCB }; // LoRaWAN NwkSKey, network session key

static const u1_t PROGMEM APPSKEY[16] = { 0x32, 0xE9, 0x80, 0x3D, 0x5C, 0x26, 0x84, 0xF6,
0xE8, 0x44, 0x36, 0x05, 0x2E, 0x3A, 0x59, 0xFD }; // LoRaWAN AppSKey, application session key

static const u4_t DEVADDR = 0x260117FF ;   // LoRaWAN end-device address (DevAddr)
```

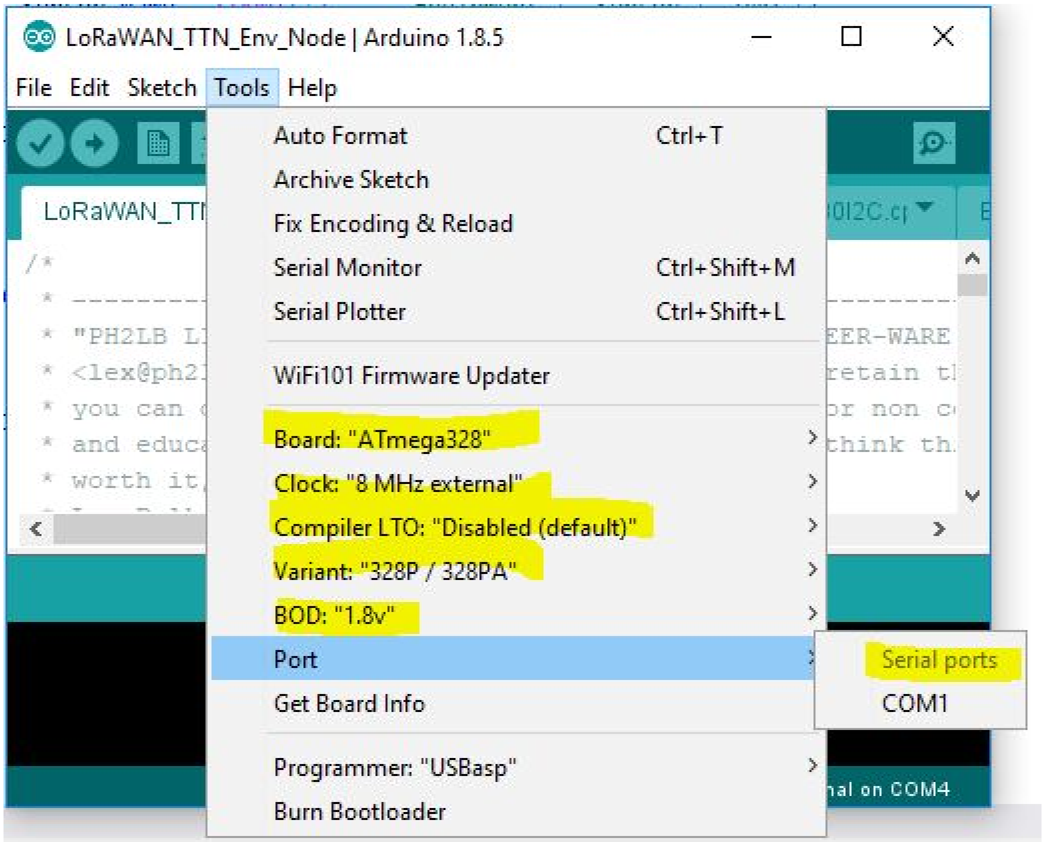
nce you have done all this and you have made your choice and possibly implemented in Step 2, you can program the software in the Arduino.

First check whether the source code compiles by clicking on the check mark in the Arduino studio.



If you do not get any errors, we can start uploading to the Arduino.

Make sure you have installed the MiniCore hardware library and you have set up the Arduino studio correctly. Connect the programmer to your laptop and ensure that the Serial Port is properly set up.



Board: "ATmega328"

Clock: "8 MHz external"

Compiler LTO: "Disabled (default)"

Variant: "328P / 328PA"

BOD: "1.8v"

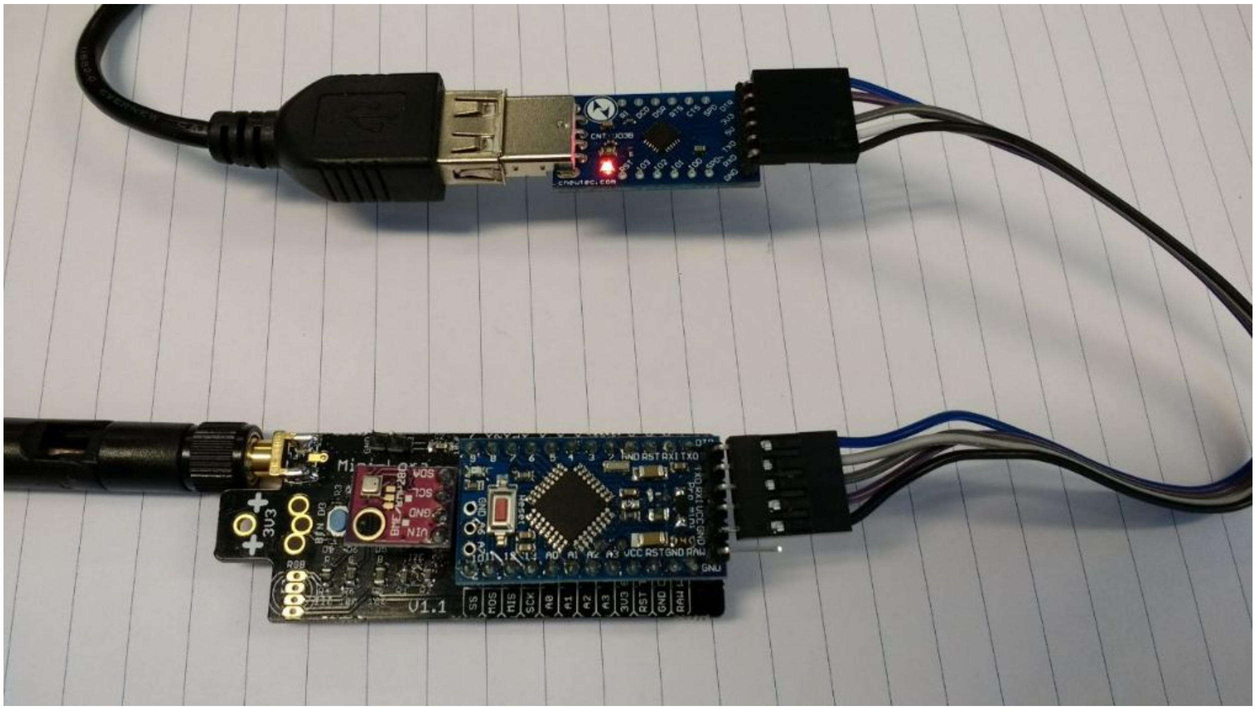
Programmer: "USBasp"

WARNING

Note: *make sure you have a good programmer that is 3.3V suitable otherwise you run the risk of blowing up everything. If you do not have one, you can possibly disconnect the power supply between the programmer and the Adruino and feed the node with the batteries. You can use headers with long pins and pull the power pin out. If you do not trust it or you are not sure what is meant, ask one of the organizers.*

Step 4.1 : I have a 3.3V programmer.

The one I had at my last workshop was set at 3.3V.

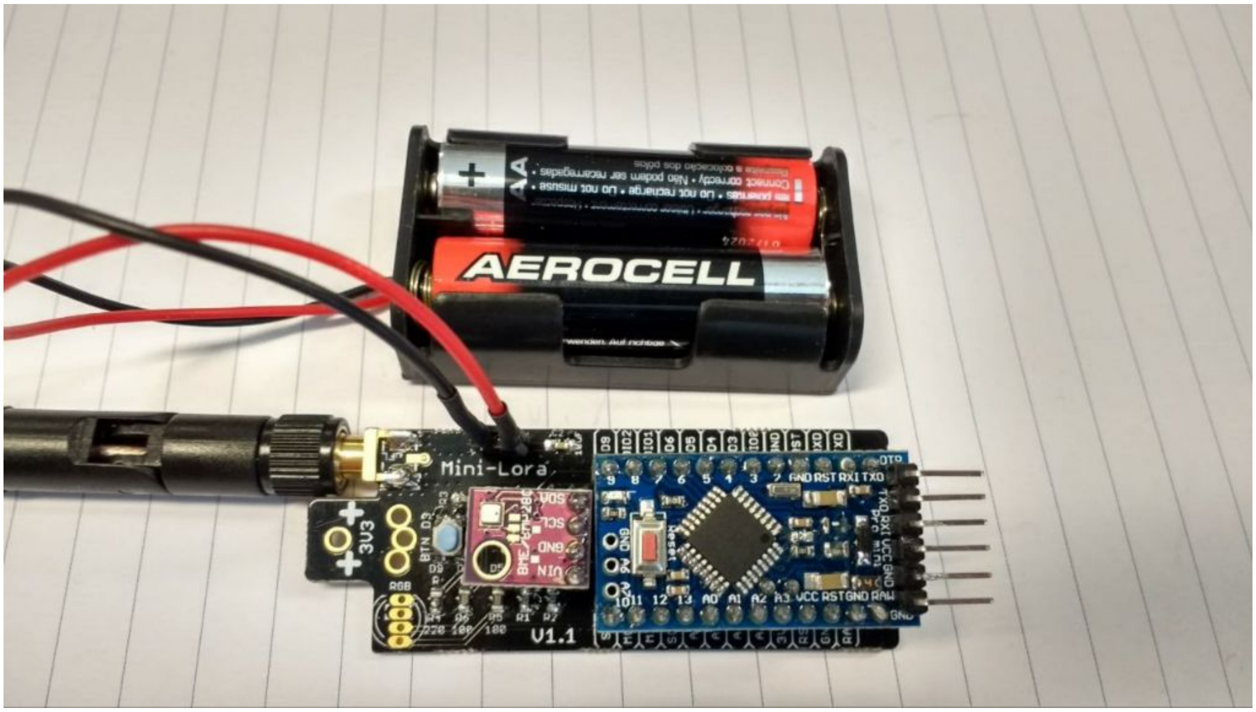


Go to step 4.3.

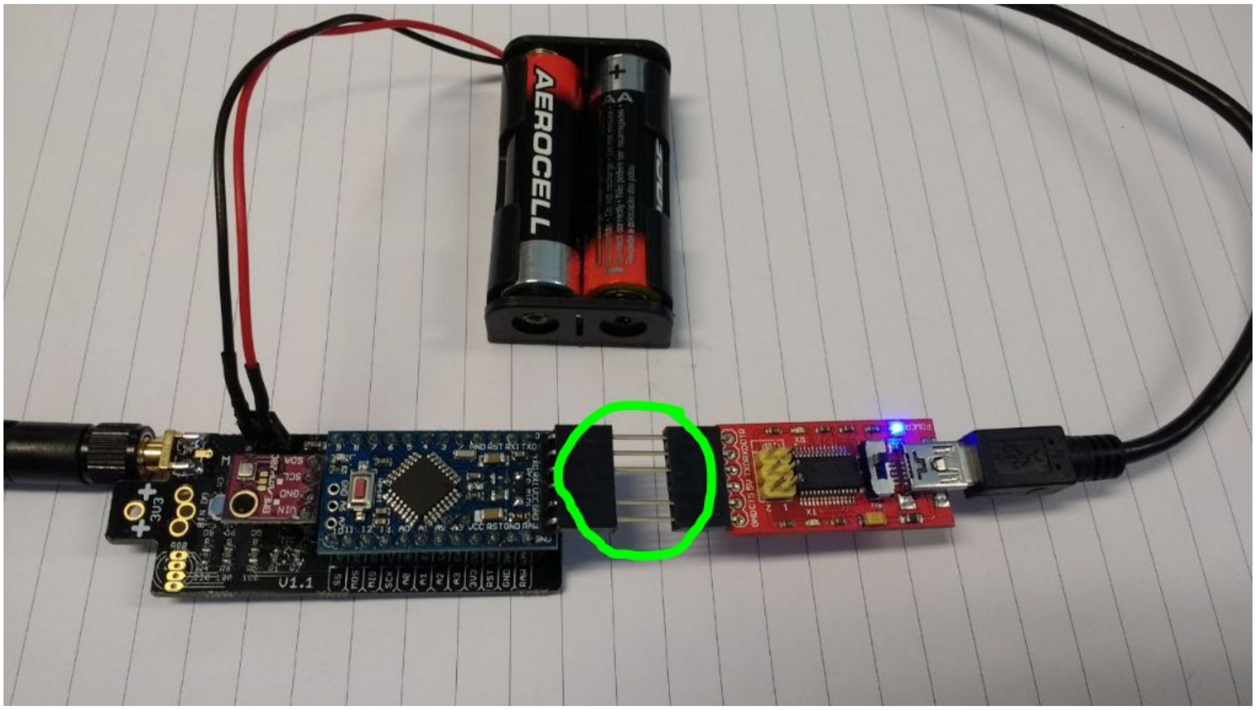
Step 4.2 : I have a 5V programmer.

But I only have a 5V programmer. Ok no panic that can all be fine.

Now connect the battery to the Node as shown in the picture below (so black on the side the word "Mini" and red on the side of the word "Lora".



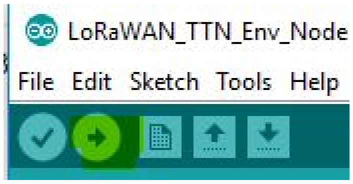
Ask the organizers for an extended header where the VCC pin is disconnected and places it between the Arduino and the programmer.



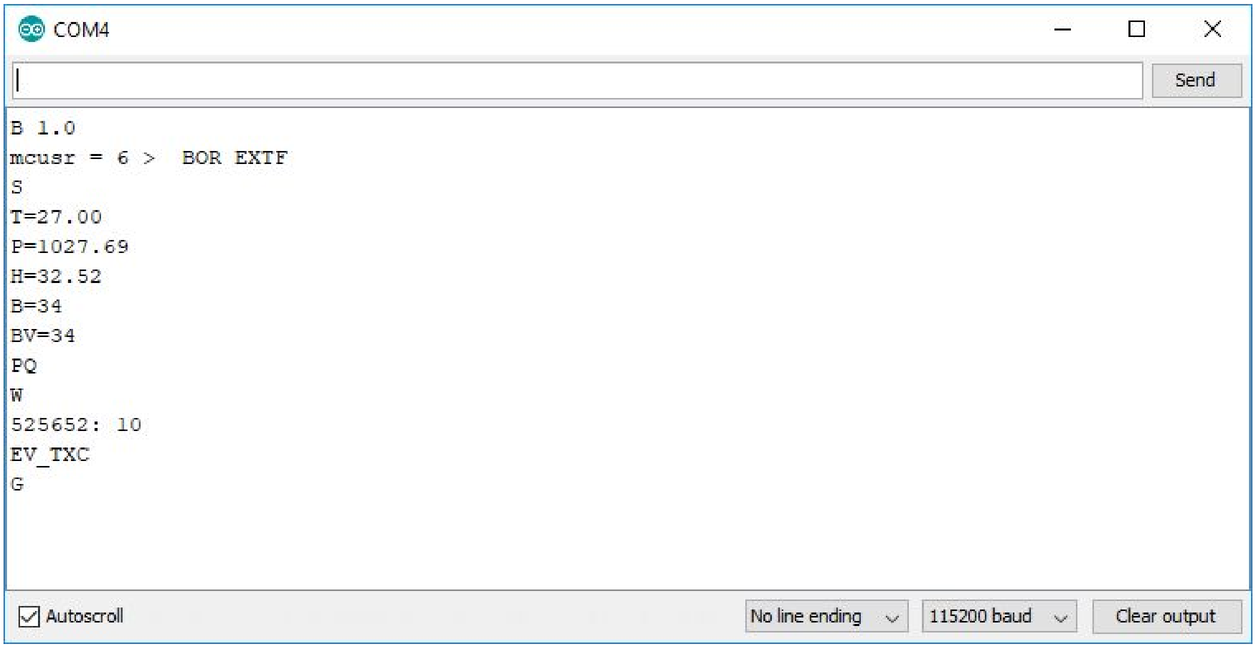
Go to step 4.3.

Step 4.3 : programming

You now have everything connected properly so on the arrow to the right in the Arduino studio.



If everything went well and you have uploaded the program in the Arduino you can look in the serial monitor. You will then see a concise boat and debug information such as measured values and information when you send messages.



```
B 1.0
mcusr = 6 >  BOR EXTF
S
T=27.00
P=1027.69
H=32.52
B=34
BV=34
PQ
W
525652: 10
EV_TXC
G
```

☑ Autoscroll   No line ending   115200 baud   Clear output

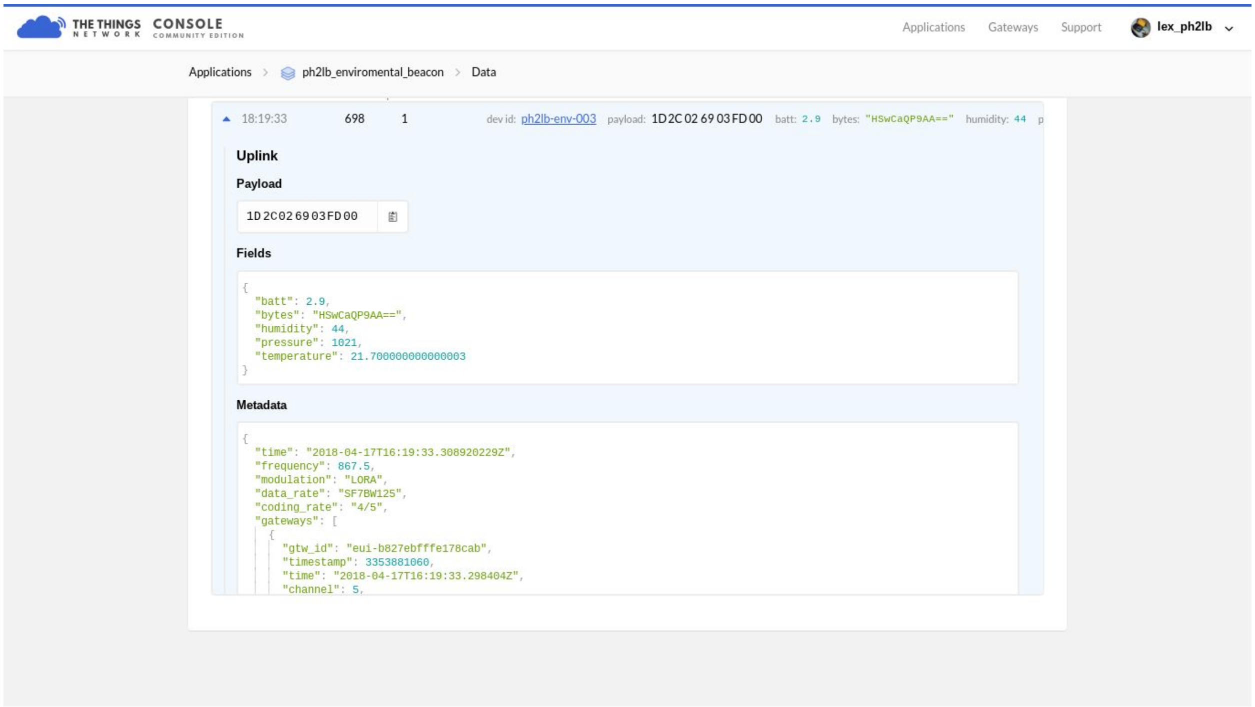
And now the fun starts.

Step 5 : TTN Console part 2

Make sure you are logged in to the TTN Console (https://account.thethingsnetwork.org/users/login) and you have selected your application, then click on the tab “Data”.

If you reset your node now, you will see data passing by in the format you have set.

Step 5.1 : I have chosen the “ENV environment”
If you had chosen the "ENV environment", a data packet would look something like this.



THE THINGS NETWORK
CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
lex_ph2lb
Applications > ph2lb_enviromental_beacon > Data

18:19:33 698 1 dev id: ph2lb-env-003 payload: 1D 2C 02 69 03 FD 00 batt: 2.9 bytes: "HSwCaQP9AA==" humidity: 44 p

Uplink

Payload

1D 2C 02 69 03 FD 00

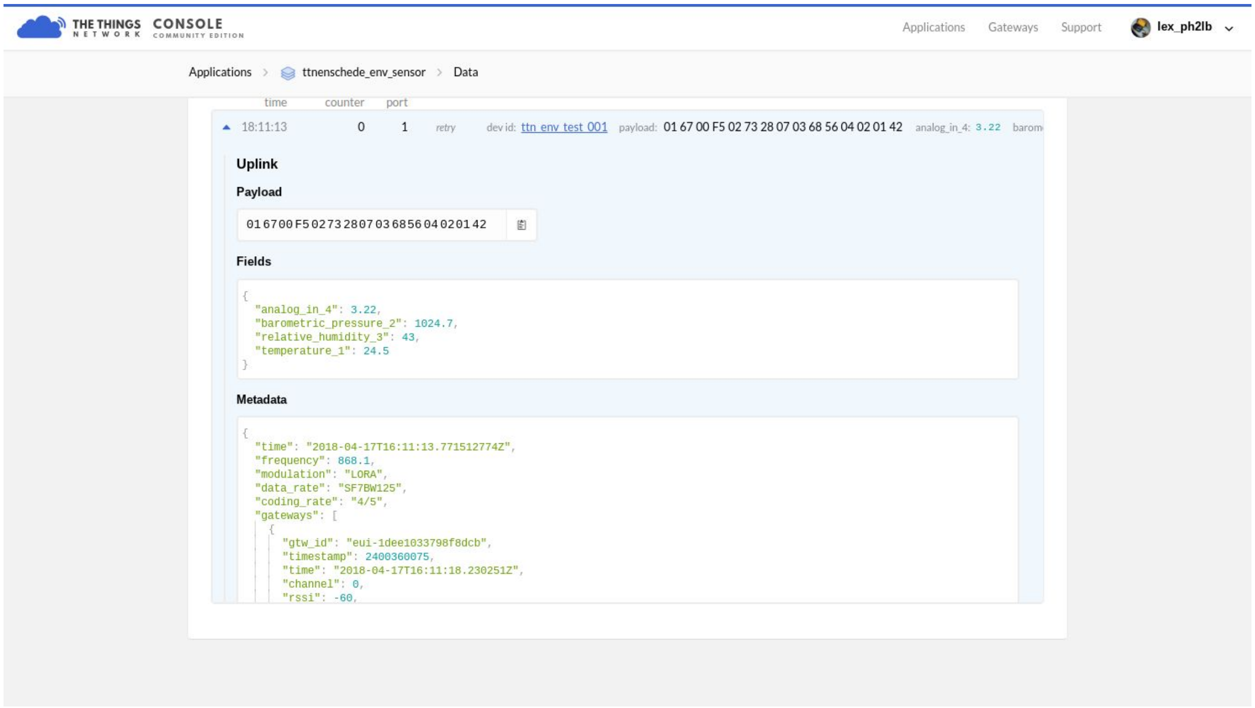
Fields

```
{
  "batt": 2.9,
  "bytes": "HSwCaQP9AA==",
  "humidity": 44,
  "pressure": 1021,
  "temperature": 21.700000000000003
}
```

Metadata

```
{
  "time": "2018-04-17T16:19:33.308920229Z",
  "frequency": 867.5,
  "modulation": "LORA",
  "data_rate": "SF7BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-b827ebfffe178cab",
      "timestamp": 3353881060,
      "time": "2018-04-17T16:19:33.298404Z",
      "channel": 5,
```

Step 5.2 : I have chosen the “Cayene environment”
If you had chosen the "Cayenne environment", a data packet would look something like this.



```
     time        counter   port
▲ 18:11:13          0        1   retry   dev id: ttn env test 001   payload: 01 67 00 F5 02 73 28 07 03 68 56 04 02 01 42   analog_in_4: 3.22   barom
```

Uplink

Payload

```
01 67 00 F5 02 73 28 07 03 68 56 04 02 01 42
```

Fields

```
{
  "analog_in_4": 3.22,
  "barometric_pressure_2": 1024.7,
  "relative_humidity_3": 43,
  "temperature_1": 24.5
}
```

Metadata

```
{
  "time": "2018-04-17T16:11:13.771512774Z",
  "frequency": 868.1,
  "modulation": "LORA",
  "data_rate": "SF7BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "eui-1dee1033798f8dcb",
      "timestamp": 2400360075,
      "time": "2018-04-17T16:11:18.230251Z",
      "channel": 0,
      "rssi": -60,
```

Step 6 : Enjoy your data

Step 6.1 : I have chosen the “ENV environment”

If you have chosen the "ENV environment", please pass on the following to Lex

Device ID : . . . . (the text device ID)
Owner : . . . . (your name and e-mailadres)
Area : . . . . (place)
Latitude : . . . . (needed to display results on the map)
Longitude : . . . . (needed to display results on the map)
Omschrijving : : . . . .

He will then register your node in the "ENV environment" and then you can view your data on : https://ssl.bolkesteijn.nl:8943/env/ and on https://ssl.bolkesteijn.nl:8943/env/angular/ (experimentele Angular environment).

Stap 6.2 : I have chosen the “Cayene envrionment”

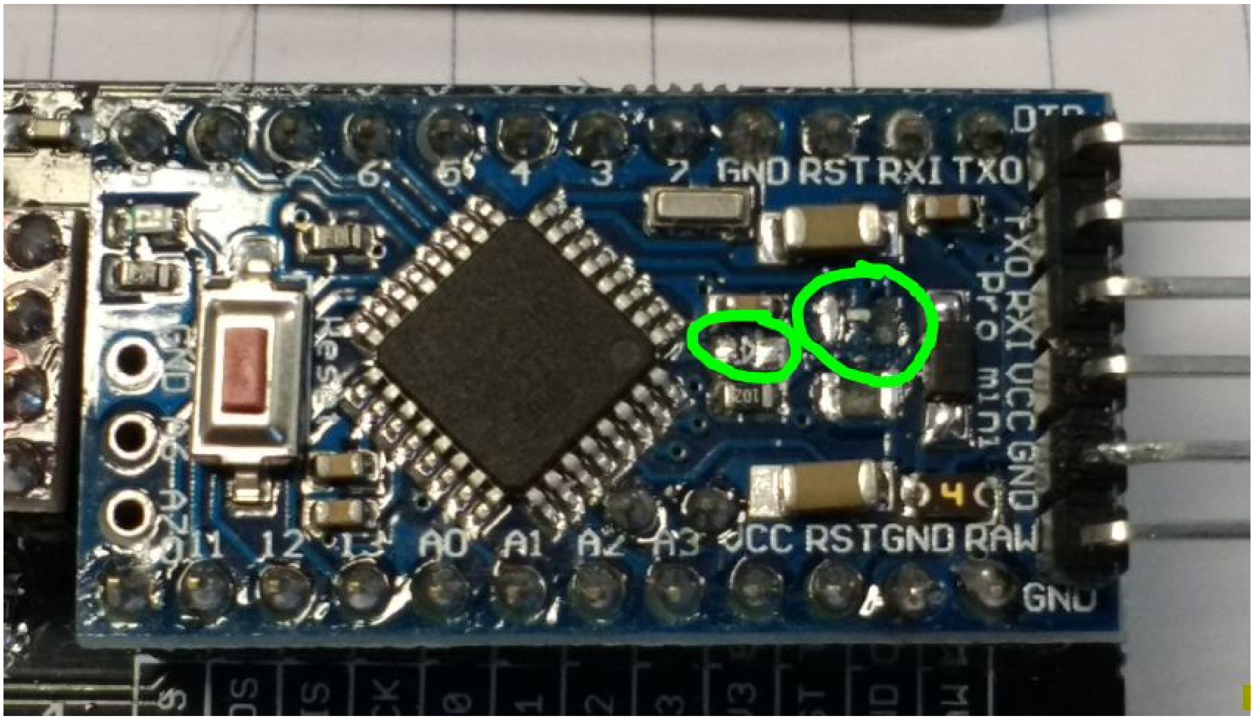
Go to https://cayenne.mydevices.com/cayenne/dashboard and see the incoming data.

Now what?

There are still a few things that can be improved, such as for example brownout / reset detection. See function : `showBootStatus(mcusr);`

Even lower power ?

Now the Arduino will not be optimally low-power. In deep trail he will still consume about 1.5mA. If you want a more optimal deep trail, you can achieve this by carefully removing the power LED and possibly the voltage regulator afterwards.



As a comparison :

- Deepsleep with power led and regulator = 1.5mA
- Deepsleep without power led and with regulator = 29.6uA
- Deepsleep without power led and without regulator = **4.1uA** :-)

Other useful info

https://www.thethingsnetwork.org/forum/t/full-arduino-mini-lorawan-and-1-3ua-sleep-mode/8059
https://www.thethingsnetwork.org/forum/t/lora-bme280-environmental-node-with-webbased-backend/9264