

Bouw je eigen
weerstation workshop!

18 apr 2018 TheThingsNetwork Enschede

Introductie

Wat heb je nodig voor deze workshop ?

Voor deze workshop heb je het volgende nodig :

- Laptop (of PC) met daarop een actuele Arduino programmeer omgeving. Deze kun je vinden op : <https://www.arduino.cc/en/Main/Software>
- Een kopie van de broncode die je kunt vinden op : https://github.com/ph2lb/ttnenschede_env_node
- De libraries zoals benoemd in de broncode :
 - 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>
- Een TheThingsNetwork account (login of maak aan op) : <https://account.thethingsnetwork.org/users/login>
- Als je hebt een 3.3V USB UART kabeltje of adapter die je kunt gebruiken om een Arduino ProMini te programmeren.

Verdere info kun je vinden op :

<https://www.meetup.com/The-Things-Network-Enschede/events/248921307/>

Welke hardware heb je gekregen?

De hardware bestaat uit een compleet gemonteerde node met :

- Arduino Mini Pro met 1.8V BOD
- CH2I Mini LoRa PCB voor I2C sensors
- RFM95W met eenvoudige draad antenne
- BME280
- Batterij houder voor 2 AA batterijen

Deze node is Low current* en batterij gevoed.

** op voorwaarde dat je de spanningsregulator en power led verwijderd. Dit hebben we bewust nog niet gedaan omdat mogelijk niet iedereen dit wil (en je kunt nu nog zien dat hij aan staat). Zie laatste pagina van deze handleiding.*

De node Arduino source code

De software voor de node is geschreven in Arduino. Hoewel de software eigenlijk nog beta is (totaal 0,0 garantie) werkt hij al een hele tijd tot volle tevredenheid. Een kopie van de broncode die je kunt vinden op : https://github.com/ph2lb/ttnenschede_env_node

Lets get hands on

Stap 1 : Keuzes, keuzes en keuzes

Voordat we kunnen beginnen moet je bepalen wat je met de berichten van de node wilt doen. Je kunt deze laten sturen naar de “ENV omgeving”, naar de Cayenne omgeving of zelfs naar je eigen omgeving.

De ENV omgeving is een opensource platform welke is ontwikkeld door Lex Bolkesteijn en wordt op dit moment bij hem op een prive server gehost. Dit systeem is gepubliceerd op het forum van TheThingsNetwork (zie :

<https://www.thethingsnetwork.org/forum/t/lora-bme280-environmental-node-with-webbased-backend/9264>)

De Cayenne omgeving is een IoT platform welke door MyDevices is ontwikkeld en naadloos met TheThingsNetwork integreert. Hierover zijn veel tutorial te vinden op TheThingsNetwork. Als je hier mee wilt starten raden wij je aan de stappen van deze LAB door te lopen :

<https://www.thethingsnetwork.org/docs/applications/cayenne/>

In de Arduino broncode zijn 2 soorten berichten gedefinieerd, namelijk het bericht formaat voor de “ENV omgeving” en voor een “Cayenne omgeving”. Wil je je eigen bericht definiëren? Dat kan natuurlijk ook, maar voor deze workshop gaan we uit dat je een van de twee vooraf gedefinieerde berichten formaten kiest.

Wat is je keuze ?

☐ ENV omgeving

☐ Cayenne

Keuze gemaakt ? Top, dan gaan we eens met de Arduino source code aan de gang.

Stap 2 : De node Arduino source code

Zoals al gezegd, kan de Arduino source code 2 soorten berichten versturen, namelijk het bericht formaat voor de Env omgeving en voor een Cayenne omgeving en als het goed heb je hierboven een keuze gemaakt welk pad je wilt bewandelen.

Als je de broncode ophaalt van bovenstaande github dan staat hij default ingesteld op de “ENV omgeving” en kun je door naar de volgende stap.

Heb je gekozen voor de “Cayenne omgeving” dan moet je in de broncode onderstaande regel even opzoeken :

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

En veranderen in :

```
#define USE_CAYENNE 1
```

Dus de // even weghalen.

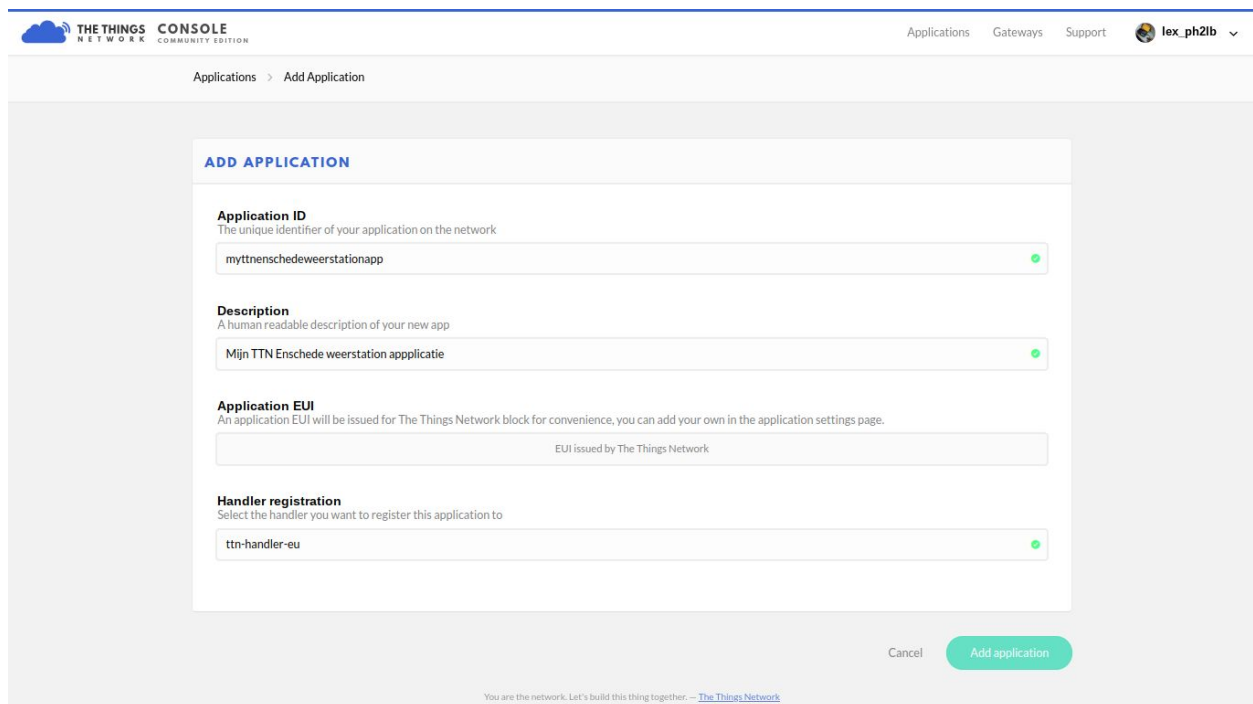
Stap 3 : TTN Console

Voordat je de node kunt gebruiken dien je in het TTN Console een applicatie aan te maken. Dit doe je door in te loggen met je TheThingsNetwork account op :

<https://account.thethingsnetwork.org/users/login>

Stap 3.1 : aanmaken applicatie

Maak een applicatie aan door achtereenvolgens Application - Add Application te kiezen en vul de gevraagde gegevens in.



The screenshot shows the 'ADD APPLICATION' form in the TTN Console. The form is titled 'ADD APPLICATION' and has a blue header. It contains four sections, each with a title, a description, and a text input field with a green checkmark indicating successful input.

- Application ID**: The unique identifier of your application on the network. Input: myttnschedeweestationapp.
- Description**: A human readable description of your new app. Input: Mijn TTN Enschede weerstation applicatie.
- 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. Input: EUI issued by The Things Network.
- Handler registration**: Select the handler you want to register this application to. Input: ttn-handler-eu.

At the bottom right, there are two buttons: 'Cancel' and 'Add application' (in a green rounded rectangle). At the bottom center, there is a footer text: 'You are the network. Let's build this thing together. — The Things Network'.

En klik op [Add Application]. Je komt dan weer in het Application Overview scherm.

The screenshot shows the 'Application Overview' page in the The Things Network Console. The page is titled 'APPLICATION OVERVIEW' and displays details for the application 'myttnschedeweerstation'. The details include the Application ID, Description ('Mijn TTN Enschede weerstation'), Created time ('1 minute ago'), and Handler ('ttn-handler-eu (current handler)'). There are links for 'documentation' and 'manage euis'. Below the details, the 'APPLICATION EUIS' section shows a list of EUIs with a search bar and a 'manage euis' link. The 'DEVICES' section shows a list of registered devices with a 'register device' link and a 'manage devices' link. The page has a top navigation bar with 'Applications', 'Gateways', and 'Support' links, and a user profile 'lex_ph2lb'.

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications Gateways Support lex_ph2lb

Applications > myttnschedeweerstation

Overview Devices Payload Formats Integrations Data Settings

APPLICATION OVERVIEW

Application ID myttnschedeweerstation [documentation](#)

Description Mijn TTN Enschede weerstation

Created 1 minute ago

Handler ttn-handler-eu (current handler)

APPLICATION EUIS [manage euis](#)

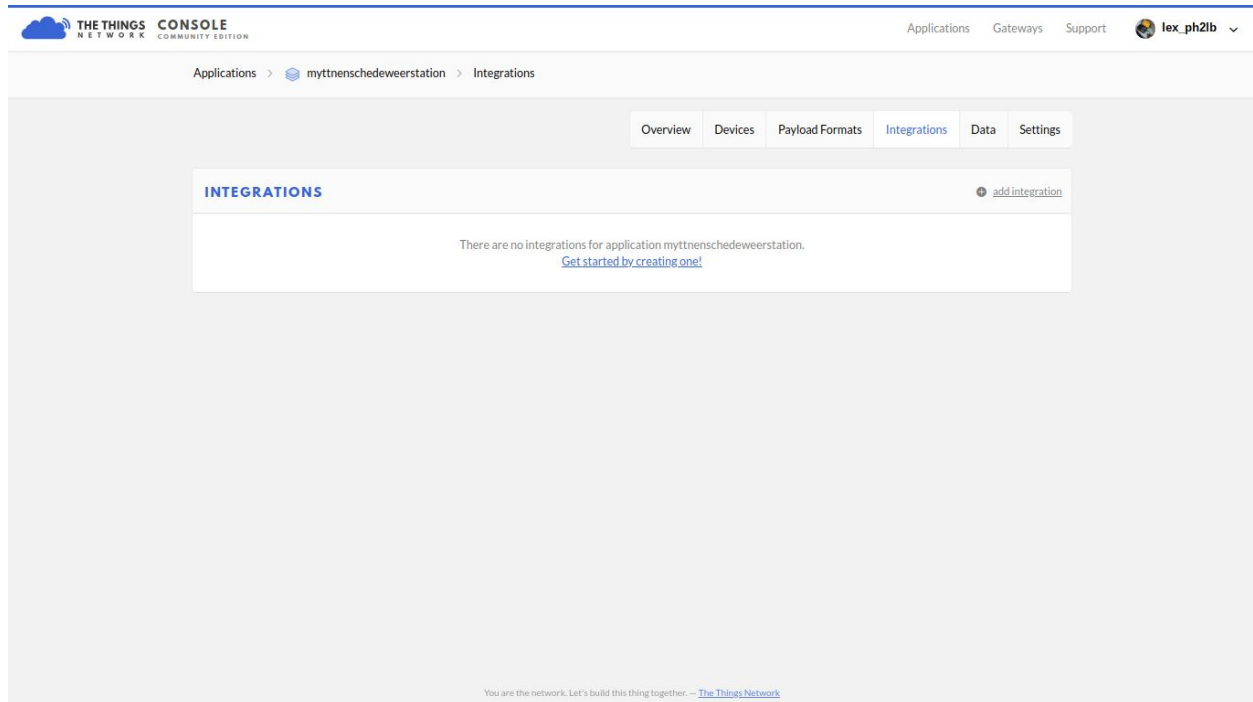
<> 70B3D57ED000BC02

DEVICES [register device](#) [manage devices](#)

0 registered devices

Stap 3.2 : integratie instellen (let op : de keuze die je hierboven hebt gemaakt).

Klik op het tabblad [Integrations]



Oh oh keuze stress. Weet je nog welke omgeving je had gekozen ?

Heb je gekozen voor de “ENV omgeving” ga dan naar stap 3.2a.

Heb je gekozen voor de “Cayenne omgeving” ga dan naar stap 3.2b.

Stap 3.2a : Ik heb gekozen voor de “ENV omgeving”

Klik op [Add Integration] en kies uit de lijst “HTTP Integration”.

Kies een mooie Process ID b.v “myttnschedeweerstation”.

Access 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
COMMUNITY EDITION

CONSOLE

[Applications](#)
[Gateways](#)
[Support](#)
 lex_ph2lb

[Applications](#) > [myttnschedeweestation](#) > [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

Access Key
The access key used for downlink

[devices](#)
[messages](#)

URL
The URL of the endpoint

Method
The HTTP method to use

THE THINGS NETWORK
COMMUNITY EDITION

CONSOLE

[Applications](#)
[Gateways](#)
[Support](#)
 lex_ph2lb

[Applications](#) > [myttnschedeweestation](#) > [Integrations](#)

URL
The URL of the endpoint

Method
The HTTP method to use

Authorization
The value of the Authorization header

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

Custom Header Value
The value of the custom Header

[Cancel](#)
[Add integration](#)

klik op [Save]

Klik vervolgens op het tabblad [Payload Functions] en kies “Custom” en voer voor de “Decoder” onderstaande functie over (zie ook broncode 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;
}

```

Voor de test voer “00 00 00 00 00 00 00 00 00 00 00 00 00 00 00” en klik op [Test]

Je zult dan als resultaat

```

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

```

Krijgen. Klopt dit? Klik dan op [Save Payload Functions]

Ga door naar Stap 3.3

Stap 3.2b : Ik heb gekozen voor de “Cayenne omgeving”

Klik op [Add Integration] en kies uit de lijst “Cayenne”.

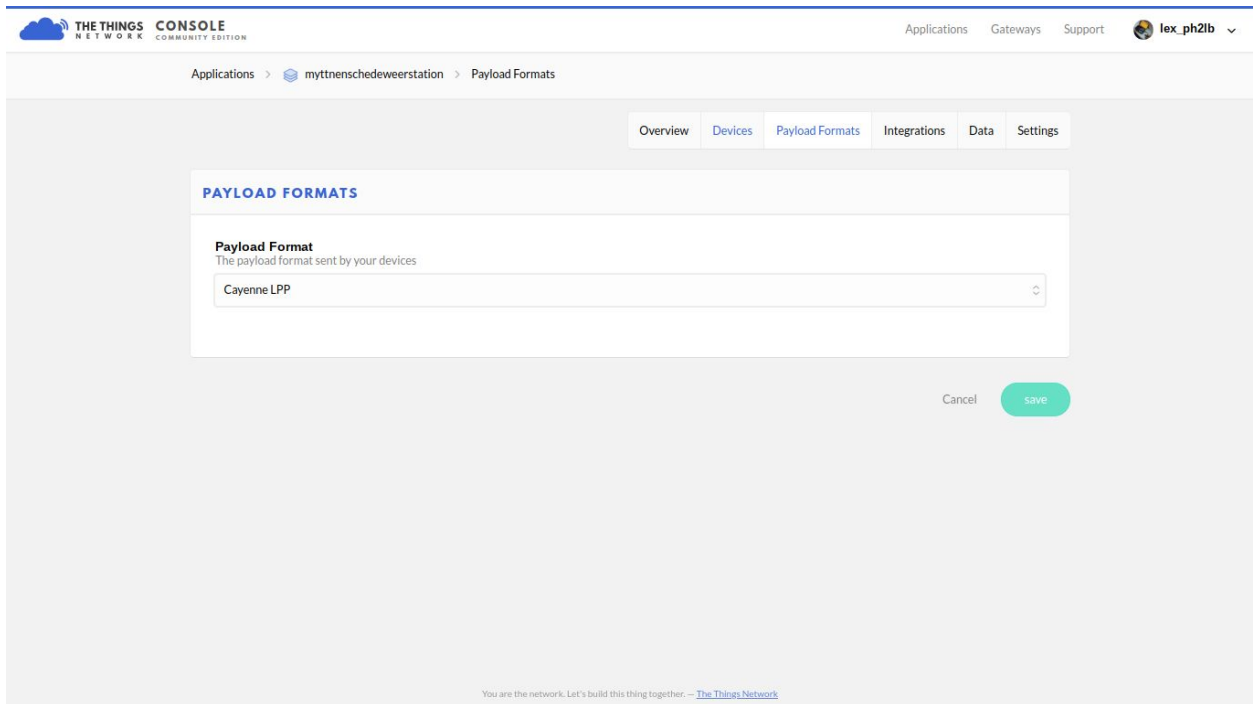
Is de URL van je Cayenne dashboard :

<https://cayenne.mydevices.com/cayenne/dashboard/lora/00000000-0000-0000-0000-000000000000>
0000 dan is je ProcessID 00000000-0000-0000-0000-000000000000

The screenshot shows the 'ADD INTEGRATION' form in the The Things Network Console. The form is for adding a 'Cayenne' integration (v2.6.0) by 'myDevices'. It includes a description: 'Quickly design, prototype and commercialize IoT solutions with myDevices Cayenne' and a link to 'documentation'. The 'Process ID' field is pre-filled with '00000000-0000-0000-0000-000000000000'. The 'Access Key' dropdown is set to 'default key', with 'devices' and 'messages' as options. At the bottom right, there are 'Cancel' and 'Add integration' buttons.

Kies voor de acces key “default key” en klik op [Add Integration]

Klik vervolgens op het tabblad [Payload Functions] en kies “Cayenne LPP”



klik op [Save]

Ga door naar Stap 3.3

Stap 3.3 : device registratie

Klik op het tabblad “Devices” en klik op [Register Device].

Kies voor de “Device ID” een stoere naam en klik op het knopje naast het invul veld “Device EUI” zodat de tekst “this field will be generated” komt te staan.

The screenshot shows the 'REGISTER DEVICE' form in the The Things Network Console. The form is titled 'REGISTER DEVICE' and has a link for 'bulk import devices'. It contains four sections: 'Device ID' with a text input field containing 'myttnschedeweestationnode_001'; 'Device EUI' with a text input field containing 'this field will be generated' and a pencil icon; 'App Key' with a text input field containing 'this field will be generated' and a pencil icon; and 'App EUI' with a text input field containing '70 B3 D57E D000 BC 02'. At the bottom right, there are 'Cancel' and 'Register' buttons.

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications Gateways Support lex_ph2lb

Applications > myttnschedeweestation > Devices

Overview Devices Payload Formats Integrations Data Settings

REGISTER DEVICE [bulk import devices](#)

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

myttnschedeweestationnode_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 D57E D000 BC 02

Cancel Register

Klik vervolgens op [Register] en je krijgt onderstaande scherm.

THE THINGS NETWORK CONSOLE COMMUNITY EDITION Applications Gateways Support lex_ph2lb

Applications > myttenschedeweestation > Devices > myttenschedeweestationnode_001

Overview Data Settings

DEVICE OVERVIEW

Application ID myttenschedeweestation

Device ID myttenschedeweestationnode_001

Activation Method OTAA

Device EUI 00 F8 63 7E 6D F5 7E CE

Application EUI 70 B3 D5 7E D0 00 BC 02

App Key

Status ● never seen

Frames up 0 [reset frame counters](#)

Frames down 0

DOWNLINK

Maar nu is hij nog OTAA Activation Method en deze workshop gaat uit van de ABP Activation Method. Dit kunnen we wijzigen in het tabblad “Settings”. Klik op het tabblad “Settings”

THE THINGS NETWORK CONSOLE COMMUNITY EDITION Applications Gateways Support lex_ph2lb

Applications > myttenschedeweestation > Devices > myttenschedeweestationnode_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
70 B3 D5 7E D0 00 BC 02

Activation Method
OTAA ABP

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

Network Session Key

Klik op ABP en als je hier toch bent voer een store Description in. Scroll even verder naar beneden en zet het vinkje bij “Frame Counter Checks” uit (de gele waarschuwing mag je negeren).

THE THINGS NETWORK CONSOLE COMMUNITY EDITION

Applications Gateways Support lex_ph2lb

Applications > myttnschedeweestation > Devices > myttnschedeweestationnode_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

Delete Device Cancel Save

Klik vervolgens op Save. Je komt dan weer terug in het Device Overview scherm. Klik op de oogjes naast “Network Session Key” en “App Session Key” zodat de keys zichtbaar worden en vervolgens op de “<>” naast “Device Address”, “Network Session Key” en “App Session Key”. Het scherm ziet er dan zo uit :

THE THINGS NETWORK
COMMUNITY EDITION

CONSOLE

[Applications](#)
[Gateways](#)
[Support](#)
 lex_ph2lb

[Applications](#) >
[myttnschedeweestation](#) >
[Devices](#) >
[myttnschedeweestationnode_001](#)

DEVICE OVERVIEW

Application ID myttnschedeweestation

Device ID myttnschedeweestationnode_001

Description Mijn TTN Enschede weerstation

Activation Method ABP

Device EUI <> 00F8637E6DF57ECE

Application EUI <> 70B3D57E000BC02

Device Address <> msb { 0x26, 0x01, 0x13, 0x13 }

Network Session Key <> msb { 0x56, 0xEC, 0x0D, 0x77, 0xEE, 0x90, 0x9F, 0xE4, 0xD7, 0xC8, 0x83, 0x32, 0x61, 0xEE, 0xFD, 0xDC }

App Session Key <> msb { 0x27, 0x6E, 0x57, 0x80, 0x91, 0x99, 0x84, 0x83, 0x95, 0x59, 0x6D, 0xEC, 0x5E, 0xA1, 0x16, 0xC1 }

Status ● never seen

Frames up 0 [reset frame counters](#)

Frames down 0

Je hebt nu de “Device Address”, “Network Session Key” en “App Session Key” zichtbaar die je nodig hebt in je Arduino bron code. Je kunt deze velden eenvoudig kopiëren door op het clipboard te klikken want deze heb je nodig in de volgende stap.

```
{ 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 }
```

Heb je je :

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

Stap 4 : De node Arduino source code part 2

Om een LoRa node in TheThingsNetwork te kunnen gebruiken en te identificeren heb je in de vorige stap de “Network session key”, “Application session key” en het “Device Address” aangemaakt en als het goed is gekopieerd.

Nu moeten die moet “Network session key”, “Application session key” en het “Device Address” worden opgenomen in de arduino source code. Zoek in de broncode onderstaande sleutels en vervang ze door jou sleutels die je gemaakt hebt in Stap 3.3.

```
// Keys for ttn_env_test_001

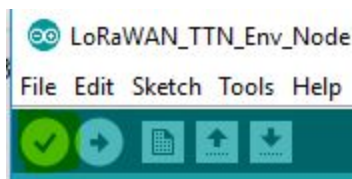
static const PROGMEM u1_t NWKSKY[16] = { 0xEC, 0x88, 0x17, 0x61, 0xFB, 0x51, 0xB7, 0x27,
0x2D, 0xEE, 0x47, 0xCE, 0xE8, 0xC8, 0x6C, 0xCB }; // LoRaWAN NwkSKey, network session key

static const u1_t PROGMEM APPSKY[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)
```

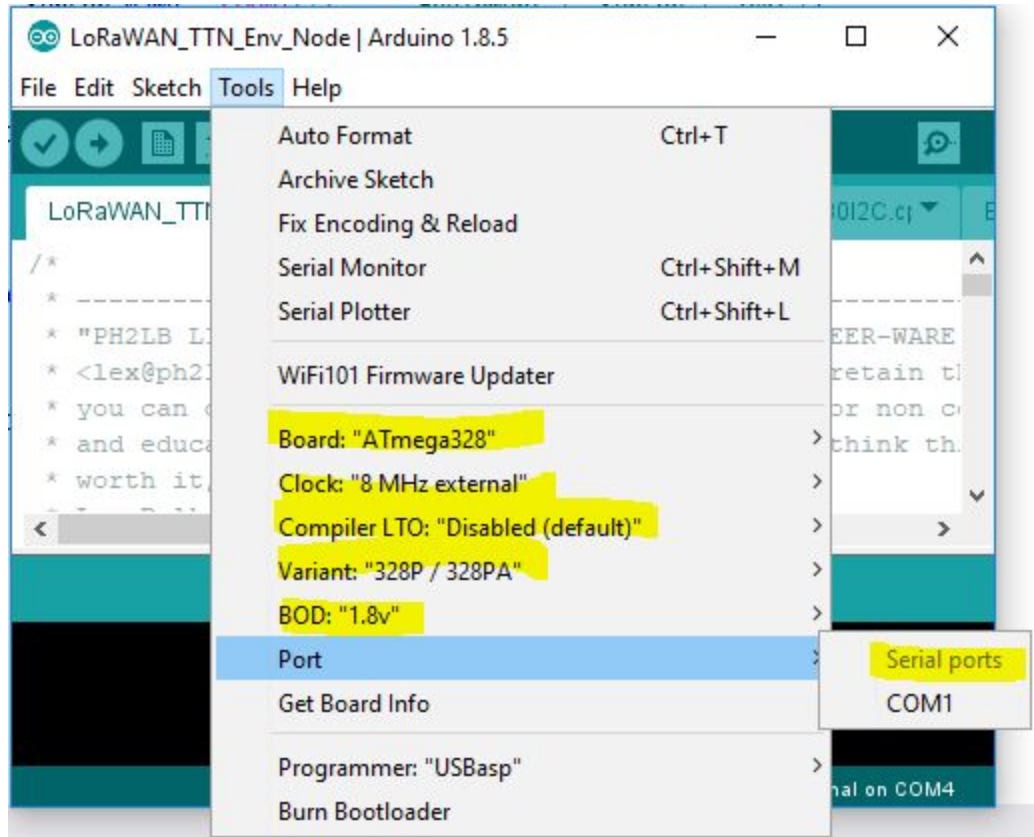
Als je dit allemaal gedaan hebt en je hebt je keuze gemaakt en eventueel doorgevoerd in Stap 2, kun je de software programmeren in de Arduino.

Eerst kijk of de broncode compileert door op het Vinkje in de Arduino studio te klikken.



Krijg je geen fouten, dan kunnen we beginnen met het uploaden naar de Arduino.

Zorg je de MiniCore hardware library hebt geïnstalleerd en je de Arduino studio goed hebt ingesteld. Sluit de je programmer aan op je laptop en zorgt dat ook de Serial Port goed ingesteld staat.

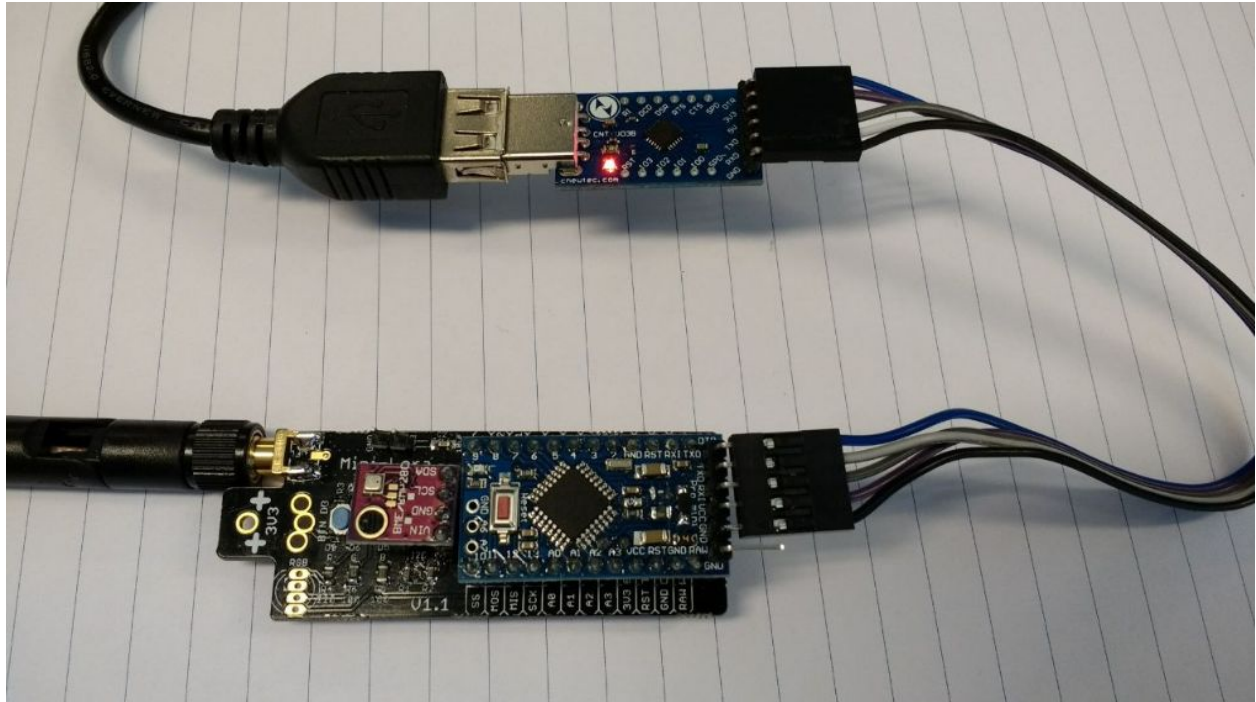


WARNING

Let op : zorg dat je een goede programmer hebt die 3.3V geschikt is anders loop je het risico de boel op te blazen. Heb je die niet, kun je eventueel de voedings lijn tussen de programmer en de Aduino los laten en de node met de batterijen te voeden. Je kunt hiervoor headers met lange pinnen gebruiken en de voedings pin er uit trekken. Vertrouw je het niet of weet je niet zeker wat er bedoeld wordt, vraag even een van de organisatoren.

Stap 4.1 : ik heb een 3.3V programmer.

Die ik bij mijn vorige workshop er bij had zitten en ingesteld had op 3.3V.

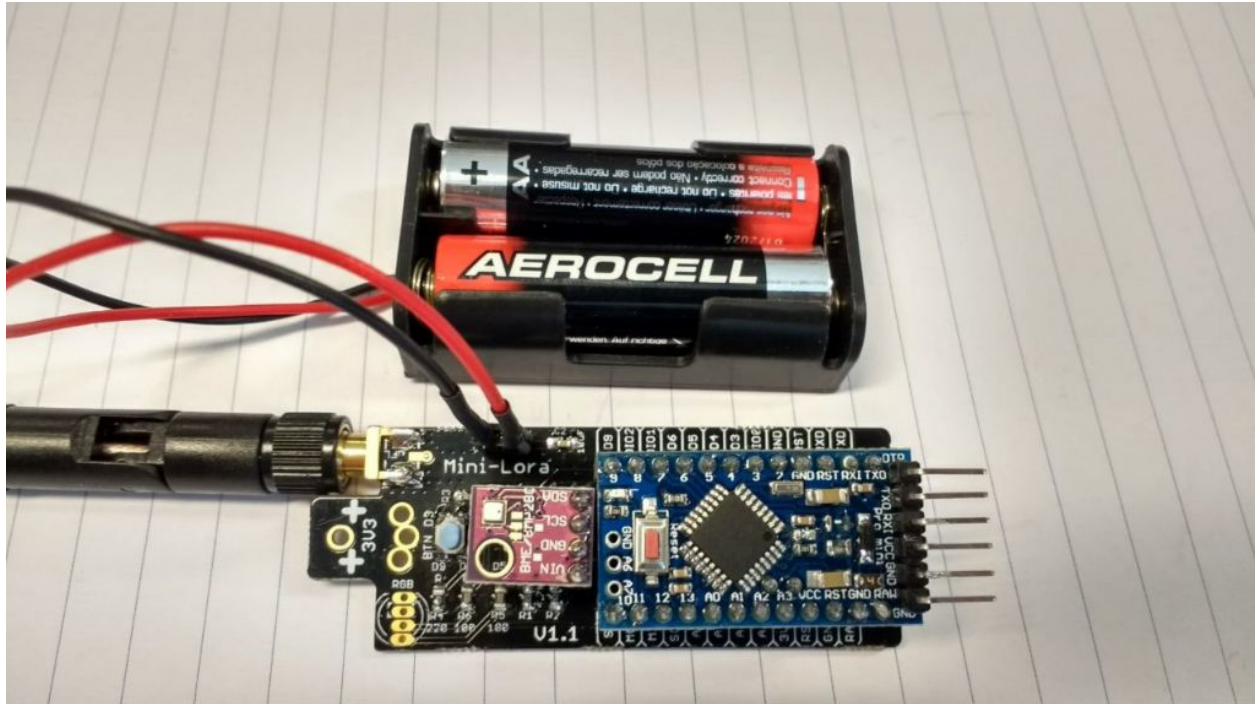


Ga naar stap 4.3.

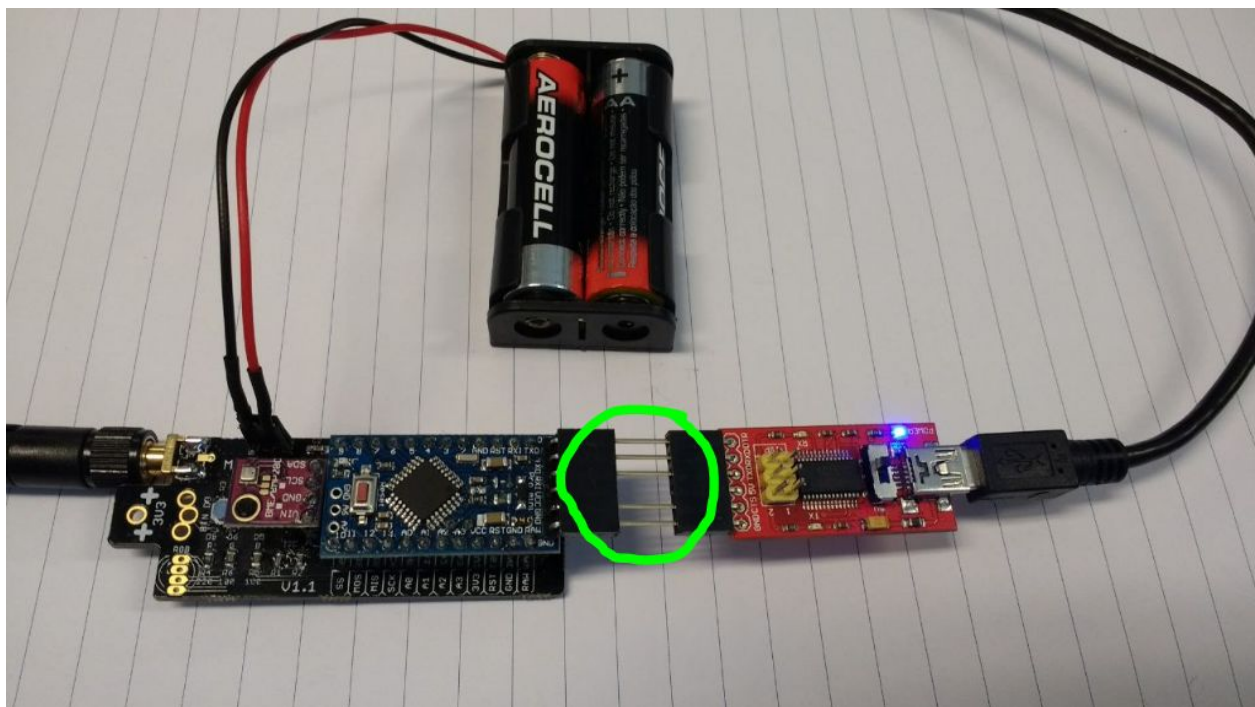
Stap 4.2 : ik heb een 5V programmer.

Maar ik heb alleen een 5V programmer. Ok geen paniek dat kan allemaal prima.

Sluit nu de batterij aan op de Node zoals weergegeven op onderstaande foto (dus zwart aan de kant het woord “Mini” en rood aan de kant van het woord “Lora”).



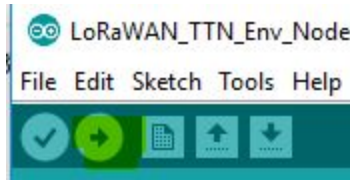
Vraag even aan de organisatoren om een verlengde header waarbij de VCC pin is losgekoppelt en plaatst die tussen de Arduino en de programmer.



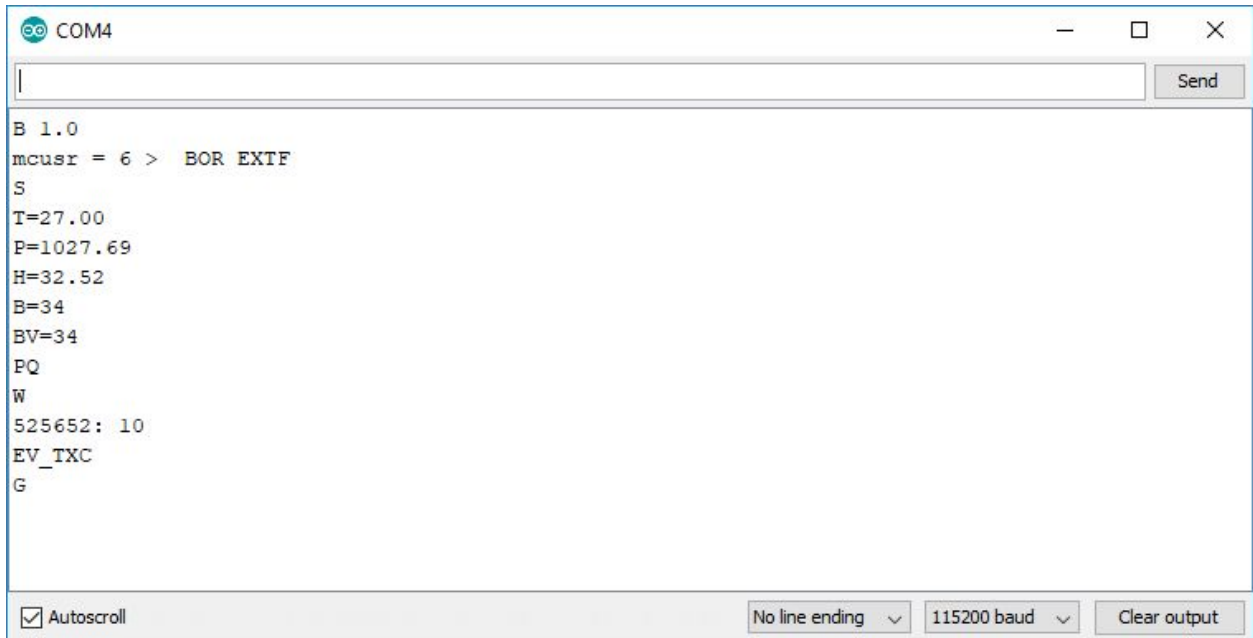
Ga naar stap 4.3.

Stap 4.3 : programmeren

Je hebt nu alles goed aangesloten dus op het pijltje naar rechts in de Arduino studio.



Als alles goed is gegaan en je hebt het programma in de Arduino ge-upload dan kun je in de seriële monitor kijken. Je ziet dan naar beknopte boot en debug informatie zoals meetwaarden en informatie als je berichten verstuurd.



En nu wordt het echt leuk.

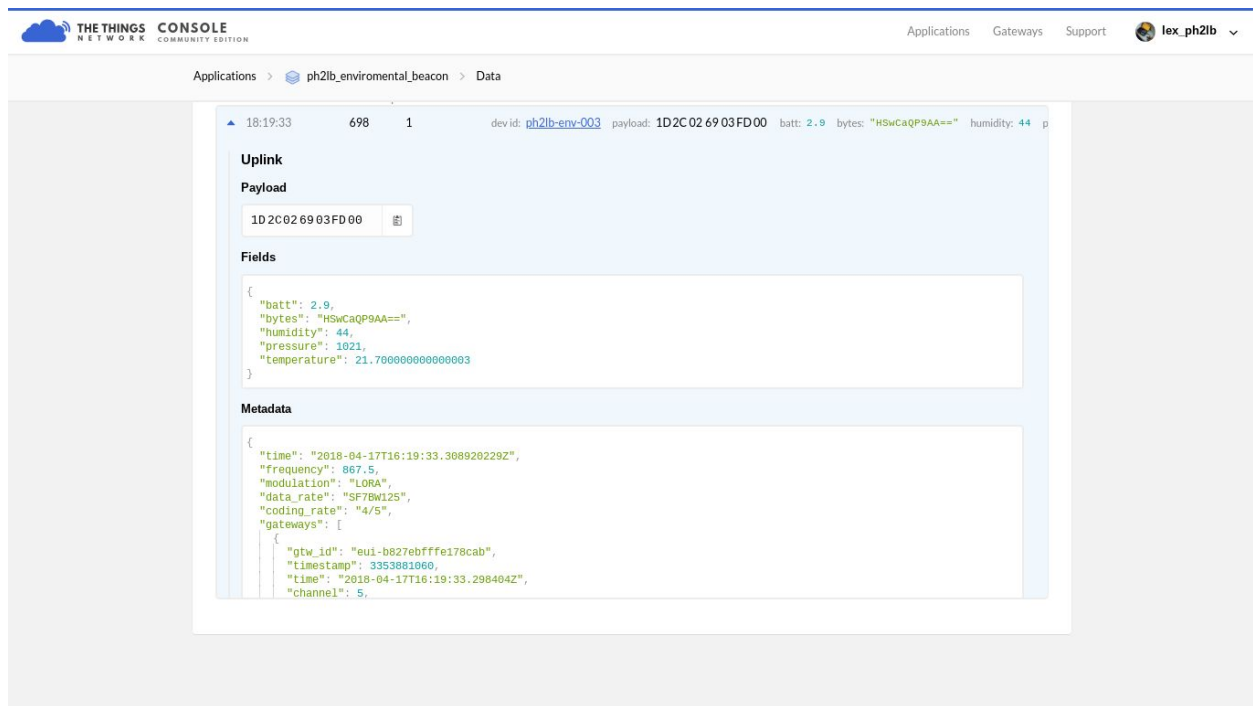
Stap 5 : TTN Console part 2

Zorg dat je ingelogd bent op het TTN Console (<https://account.thethingsnetwork.org/users/login>) en je applicatie hebt geselecteerd en klik het tabblad “Data”.

Als je nu je node reset zul je data voorbij zien komen in het door jou ingestelde format.

Stap 5.1 : Ik heb gekozen de “ENV omgeving”

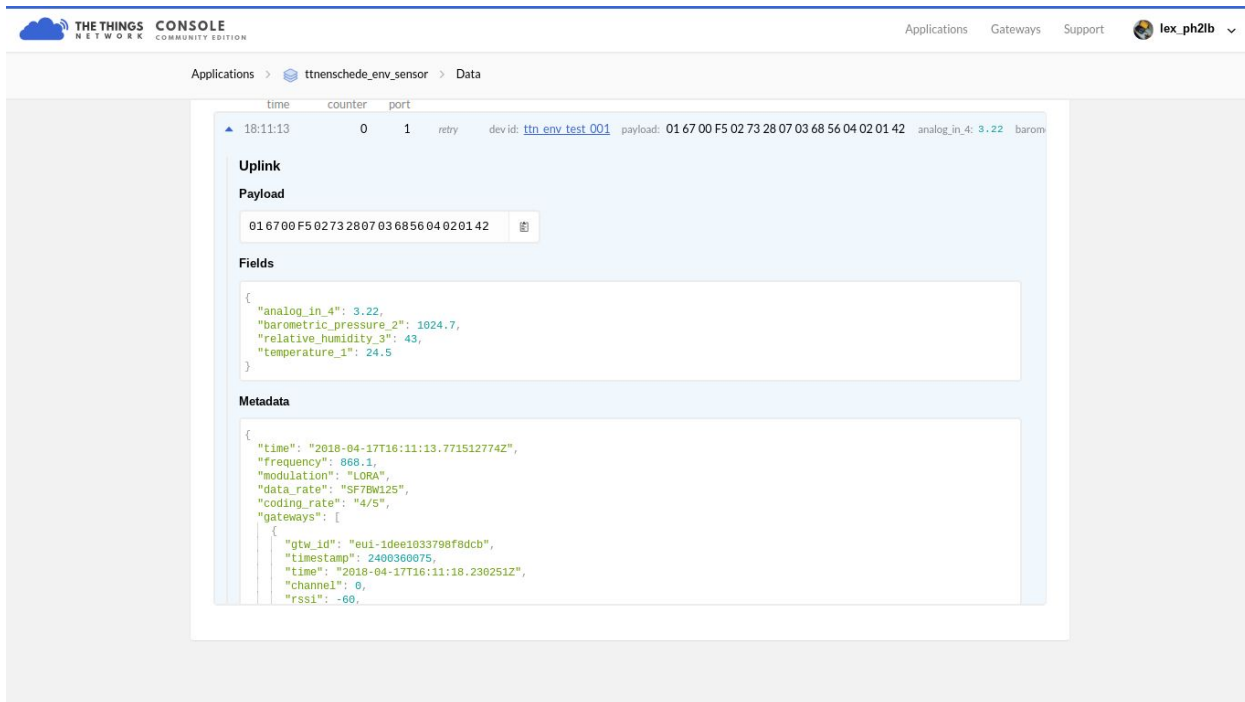
Had je voor de “ENV omgeving” gekozen zal een data packet er ongeveer zo uit zien.



The screenshot displays the TTN Console interface for the 'ph2lb_enviromental_beacon' application. The 'Data' tab is active, showing a list of data packets. The selected packet has a timestamp of 18:19:33, a size of 698 bytes, and a device ID of 'ph2lb-env-003'. The payload is '1D2C026903FD00'. The 'Fields' section shows a JSON object with the following data: 'batt': 2.9, 'bytes': 'HSwCaQP9AA==', 'humidity': 44, 'pressure': 1021, and 'temperature': 21.700000000000003. The 'Metadata' section shows a JSON object with the following data: 'time': '2018-04-17T16:19:33.308920229Z', 'frequency': 867.5, 'modulation': 'LORA', 'data_rate': 'SF7BW125', 'coding_rate': '4/5', and 'gateways': [{ 'gtw_id': 'eu1-b827ebfffe178cab', 'timestamp': 3353881060, 'time': '2018-04-17T16:19:33.298404Z', 'channel': 5 }].

Stap 5.2 : Ik heb gekozen voor de “Cayene omgeving”

Had je voor de “Cayene omgeving” gekozen zal een data packet er ongeveer zo uit zien.



Step 6 : Enjoy your data

Stap 6.1 : Ik heb gekozen de “ENV omgeving”

Heb je gekozen voor de “ENV omgeving”, geef dan even het volgende door aan Lex

Device ID : (de tekstuele device ID)
 Owner : (je naam en e-mailadres)
 Area : (plaats naam)
 Latitude : (nodig voor het tonen op een kaartje)
 Longitude : (nodig voor het tonen op een kaartje)
 Omschrijving :

Hij zal dan je node registreren in de “ENV omgeving” en dan kun je je data inzien op :
<https://ssl.bolkesteijn.nl:8943/env/> en op <https://ssl.bolkesteijn.nl:8943/env/angular/>
 (experimentele Angular omgeving).

Stap 6.2 : Ik heb gekozen voor de “Cayene omgeving”

Kijk dan op <https://cayenne.mydevices.com/cayenne/dashboard> naar de binnen gekomen data.

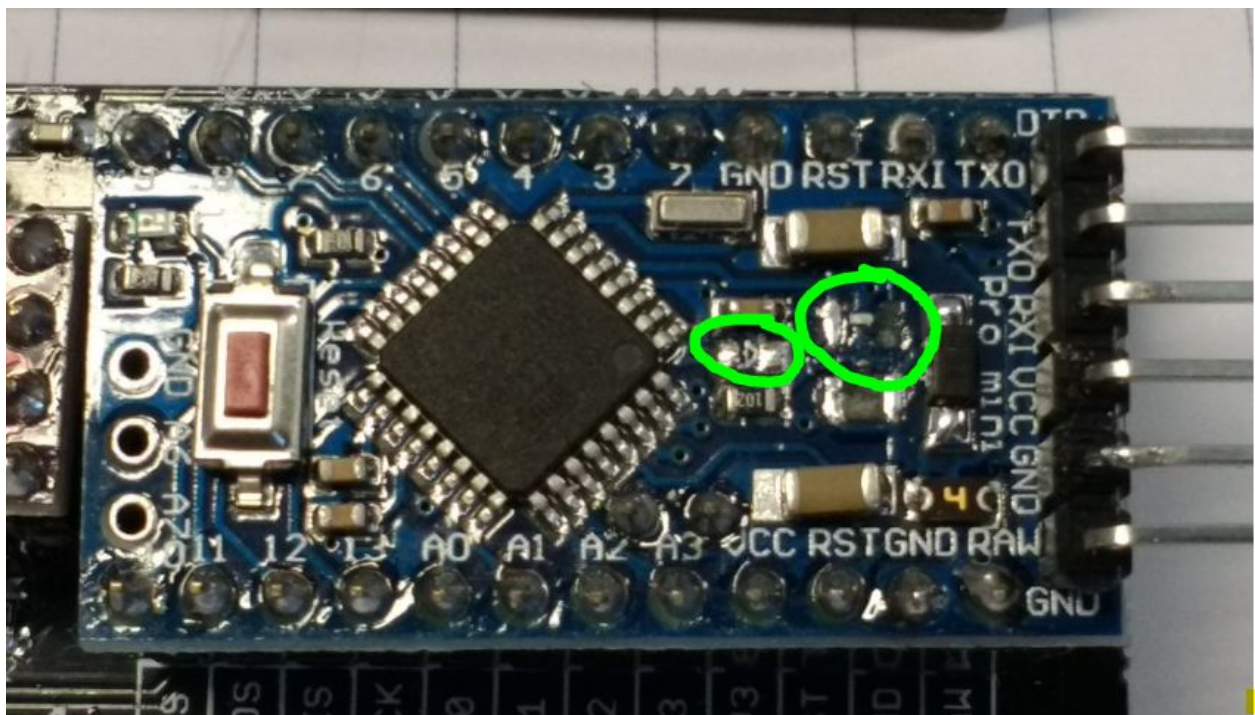
Wat nu verder?

Er zijn nog een paar dingen die beter kunnen zoals bijvoorbeeld brownout / reset detectie.

Zie functie : `showBootStatus(mcusr);`

Nog meer low power ?

Nu zal de Arduino niet optimaal low-power zijn. In deepsleep zal hij nog steeds ongeveer 1.5mA verbruiken. Wil je een meer optimale deepsleep dan kun je dit bereiken door voorzichtig het power ledje en eventueel daarna nog de spannings regulator te verwijderen.



Te vergelijking :

- Deepsleep met power led en met regulator = 1.5mA
- Deepsleep zonder power led en met regulator = 29.6uA
- Deepsleep zonder power led en zonder regulator = **4.1uA** :-)

Verder nuttige 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>