

Project Sauna

Joachim Pham
joachimpham@hotmail.com

Inhoudstabel

Inleiding	1
Opdracht	1
Benodigdheden.....	1
1) Wat moet het kunnen?	2
2) Beetje code schrijven met voorwaarde.....	3
3) Zorg dat alle instrumenten data weergeven in Serial Monitor met Arduino Uno.....	3
4) Zoek documentatie op voor instrument, aansluiting.....	4
5) De juiste USB poort kiezen en het juiste board.....	4
6) Schrijf zelf code of begin met een voorbeeld.....	5
7) Valideer of direct uploaden naar board, pas foutmeldingen aan in jouw code als die voorkomen.....	5
8) Bekijk resultaat in Seriële monitor.....	5
9) Als alles naar wens is, begin je andere programma's te combineren tot één.....	6
10) Om op lange afstand data te krijgen van de node, hebben we eerst een Gateway nodig.....	7
a. Vooraleer je de RPi kunt gebruiken, heb je een opstartprogramma nodig.....	8
b. Zorg dat alle software up-to-date is op de Raspi.....	8
c. Installeer git	9
d. Hardware aansluiten: Raspi aan de Dragino LoRa/GPS HAT.....	10
e. Start de service.....	10
f. The Things Network Simple Gateway.....	11
g. Maak een node met Arduino UNO	11
h. The Things Network Application.....	12
i. Open LMIC library met voorbeeld ttn-abp, pas info aan en GO	13
j. CayenneLPP integratie.....	13
k. Cayenne Dashboard	13
l. Om data door te sturen naar Cayenne Dashboard, gaan we een bericht schrijven met een vaste structuur.....	14
m. Nadat alle software geschreven is, kunnen we testen en fine-tunen.....	16
11) Bronnen	

Herkansing IOT project

Opdracht:

We willen een poolhouse voorzien een automatische verlichting en temperatuurscontrole die we kunnen monitoren vanop afstand.

Hiervoor willen we een Arduino en RaspBerry Pi gebruiken met de Lora technologie.

Eerst en vooral willen we de temperatuur en luchtvochtigheid meten van onze poolhouse, dit moet overal ter wereld bekeken kunnen worden.

Extra add-ons

We willen ook dat indien de temperatuur te hoog wordt er een ventilatie systeem wordt ingeschakeld.

Indien de temperatuur te laag wordt zal de verwarming moeten aanslaan.

We willen natuurlijk ook van op afstand zien wanneer deze ingeschakeld zijn.

Aan de poolhouse is er natuurlijk ook verlichting binnen en buiten. De buiten verlichting mag aanspringen zodra het donker wordt, de binnenverlichting kan handgeschakeld worden met een drukknop. Ook dit moeten we overal ter wereld kunnen raadplegen.
Dit project dient te worden gedocumenteerd op github!

(DHT11, TTN en Cayenne Cloud App)

```
DHT11 library,  
const int temp = A0
```

```
if temp < threshold temp:  
    turn on heating();  
    turn off ventilation();
```

```
elif temp > threshold temp:  
    turn off heating();  
    turn on ventilation();
```

```
else:  
    turn off heating();  
    turn off ventilation();
```

Cayenne.begin() en Cayenne.loop()

```
const int knop = D0;  
const int licht = D1;  
const int motion/LDR = A0;
```

```
if light < threshold light: {  
    turn on outlight()  
} else: {  
    turn off outlight() }
```

Moet aanwezig zijn in het project:

- ***Arduino + Hardware***
- ***Raspberry Pi***
- ***Lora***
- ***TTN (The things network)***
- ***Visualisatie van data***

Evaluatie moment:

Presentatie op 31/08 (uur nader te bepalen)

Veel succes!

in te leveren documenten:

- *Volledige documentatie van het project*
- *Presentatie*
- *Filmpje van het project*
- *Link naar github documentatie.*

1) Lees de opdracht: Wat moet het kunnen?

Welke instrumenten hebben we nodig? Leerkracht heeft al opgegeven welke toestellen vereist waren.



RaspBerry Pi (4b) [2GB](#) / [4GB](#) / [8GB](#) RAMemory



Arduino UNO [WiFi Rev2](#)



Kabels

MicroHDMI-naar-HDMI (meegeleverd aankoop RaspBerry Pi)

USB-B naar USB-A (meegeleverd aankoop Arduino UNO)

Male-naar-Female bedrading

LoRa Shields:

- Dragino LoRa / GPS HAT 868Mhz voor RaspBerry Pi
- Dragino LoRa Shield 868Mhz voor Arduino UNO

koop [hier](#)

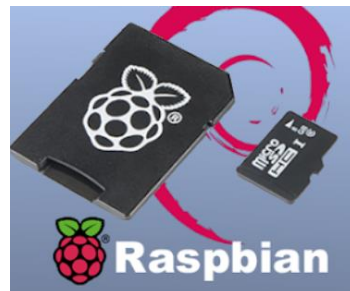
koop [hier](#)

Raspberry Pi LoRa/GPS HAT - 868MHz



Dragino LoRa Shield - 868MHz

MicroSD kaart met Raspbian OS



2) Beetje C++ code schrijven met voorwaarde:

Ex. : De ventilator moet aangaan als een temperatuur boven de 25°C wordt gemeten.

Ex. : De buitenverlichting moet aangaan als de lichtsterkte onder 500 lux is.

```
int thresholdTemp = 25;
```

```
if (temp > thresholdTemp) {
```

```
    turn on ventilator }
```

```
if (temp < thresholdTemp) {
```

```
    turn on heater }
```

```
if (temp == thresholdTemp) {
```

```
    turn off ventilator and heater }
```

```
int thresholdLicht = 500;
```

```
if (licht > thresholdLicht) {
```

```
    turn on lichtBuiten }
```

```
if (licht < thresholdLicht) {
```

```
    turn off lichtBuiten }
```

```
if (licht == thresholdLicht) {
```

```
    return; }
```

3) Zorg dat alle instrumenten data weergeven in Serial Monitor met Arduino Uno:

Download via internet Arduino IDE via Arduino.cc onder;

"Software" > "Downloads" > "Download IDE for your OS"

(of <https://www.arduino.cc/en/Main/Software>)

Na installatie, open IDE en Download de juiste libraries voor instrumenten onder;

"Hulpmiddelen" > "Bibliotheken beheren"

(of Ctrl+Shift+I)

Voor de temp → *"DHT Sensor Library"*

Voor de LoRa → *"TheThingsNetwork"* , *"TheThingsNode"*

Voor de knop → *"Bestand"* > *"Voorbeelden"* > *"02.Digital"* > *"Debounce"*

4) Zoek documentatie op voor instrument, aansluiting:

Voor de DHT11 KY-015 hebben we 3 pinnen;

S voor signaal

+ voor voltage = 3,3V (of 5V) --> Altijd dubbel checken welke voltage!!!!

- voor negatief = GND

Ik ga de DHT aansluiten op D2 van de Arduino (Uno) , GND en 3,3V.

Voor de LDR:

5v --> LDR pin1

LDR pin2 --> Analog 0

LDR pin2 --> Resistor 1000 Ohm --> GND

Voor de ventilatie en verwarming:

Een relais voor schakeling aan/uit:

Digital IN

5V

GND

Voor de RGB-Led:

B = Blauw → Digital IN

G = Groen → Digital IN

R = Rood → Digital IN

-- = Ground

Voor de knop:

Digital IN



Resistor 1000 Ohm → GND

5 Volt

5) De juiste USB poort kiezen en het juiste board:

Dat doe je onder;

```
"Hulpmiddelen" > "Board" > "Board Beheer" > "Arduino AVR Boards"
en
"Poort" > "COM....."
```

Klik op uploaden

6) Schrijf zelf code of begin met voorbeelden:

Nu kun je beginnen met schrijven van jouw eigen code.

Meestal begin je met **de libraries** op te sommen,

daarna jouw **constanten** zoals wat waar aangesloten is; `const int ledbuiten = D5;`

dan jouw **void_setup()** schrijven zoals `Serial.begin(baudrate);`

dan jouw **void_loop()** schrijven zoals `digitalRead(sensor)` of `Serial.print(tekst);`

dan kun je nog jouw eigen functies schrijven zoals **void_x()** `{ reading = digitalRead(A0); }`

Als dat nog te moeilijk is, kun je voorbeelden vinden in de libraries die je downloadde;

"Bestand" > "Voorbeelden" > "Library voor instrument"

7) Valideer of direct uploaden naar board, pas foutmeldingen aan in jouw code als die voorkomen:

Internet is jouw vriend, hij weet alles en probeer het uit!

Kopieer de **foutcode** en zoek het op in het **Engels**,

meer forums en documentatie beschikbaar.

8) Bekijk resultaat in Seriële monitor:

Alles wat je met `Serial.print()` wilt, zie je in "Seriële monitor"

(of Ctrl+Shift+M)

```
temp = dht.readTemperature();  
vocht = dht.readHumidity();  
  
Serial.println("");  
Serial.print("De temperatuur is: ");  
Serial.print(temp);  
Serial.println("°C");  
Serial.print("De vochtigheid is: ");  
Serial.print(vocht);  
Serial.print("%");
```


9) *Als alles naar wens is,
begin je andere programma's te combineren tot één:*

Omdat wij LoRa nodig hebben, downloadde we ook de "*TheThingsNetwork*" library.

Ze bevat voorbeelden van een simpele "**Hello, World**" bericht.

Omdat we sensor data gaan doorsturen, gebruiken we de integratie van "*CayenneMQTT*"

en "*CayenneLPP*" libraries.

Ze bevatten voorbeelden van data doorsturen.

Ik nam het voorbeeld: "*Actuator*" in "*CayenneMQTT*" library:

code die een lichtschakelaar bestuurt:

Open deze, probeer de code te begrijpen en kopie-paste code die je nodig hebt.

Cayenne heeft ook zijn eigen platform om data te visualiseren.

Ga naar cayenne.mydevices.com, maak een account aan,
login en open rechtsboven "Docs" voor meer info.

Omdat we met Lora gaan werken, kies je links in de overview op

"Lora" > "*Cayenne Low Power Payload*"

voor info over hoe je data stockeert en doorstuurt naar het platform.

Op de [site](#), kun je bv een toestel toevoegen, "een Arduino",
rechtstreeks data laten versturen naar Cayenne servers via het internet van jouw laptop.
Dit is enkel handig bij het ontwikkelen van een toepassing.

- Voeg het toestel dat je hebt, toe
- Sluit hem aan op de pc en kies Serial USB
- Kopie-paste de code met auth gegevens in een nieuwe IDE sketch
- Upload deze op jouw board
- Open "cmd.exe" op de pc
- Navigeer naar en vervang [YourName] naar de naam van jouw pc:
" `cd C:\Users\[YourUserName]\Documents\Arduino\libraries\CayenneMQTT\extras\scripts` "
- typ " `cayenne-ser.bat -c COM4` " als de COM-poort hetzelfde is
--> dit zorgt ervoor dat de Arduino data wordt doorgestuurd via het internet van jouw pc.

10) Om op lange afstand data te krijgen van de node, hebben we eerst een Gateway nodig:

Deze bouwen we met een;

Raspberry Pi (4b),

Dragino "Lora/GPS HAT met 868Mhz Wireless RPi Extender",

MicroSD-kaart met adapter

RaspBian OS

Balena Etcher Software: SD-kaart brander

Klavier, muis en beeldkabel.

a) Vooraleer je de RPi kunt gebruiken,
heb je een opstartprogramma nodig.

Ik gebruikte de Raspbian OS, te downloaden op

(raspberrypi.org/downloads/raspberry-pi-os)

Download "BalenaEtcher" voor het branden van de RaspberryPi .ZIP image op jouw MicroSD-kaart.

Steek de voorbereide MicroSD in de Raspberry Pi,

start hem op door de stekker in te steken en

schermkabel aan te sluiten op een beeldscherm.

Geef het ook een keyboard en muis.

Na de opstartprocedure, maak een account aan met een wachtwoord dat enkel jij kent. Vergeet deze niet!!

Voor het openen van de geavanceerde opties, typ `"raspi-config"`.

Voor de gateway moet "SPI" aanstaan, staat onder optie "interfaces" van de config.

Na de login, connecteer je met WiFi en

kijk je welk IP-adres hij heeft aangenomen

door met jouw muis op het wifi-icoon te zweven. (192.168.0.....)

Op een andere computer die op dezelfde wifi zit,

kun je "Putty" downloaden en

connecteren met het IP-adres van de RPi, login = pi en paswoord.

b) Zorg dat alle software up-to-date is op de Raspi:

```
"sudo apt-get update"
```

```
"Do you want to proceed with the upgrade?"
```

=

```
"sudo apt-get upgrade"
```

```
"wil je voortgaan met de upgrade?" (y/n) : y ENTER
```

c) Installeer git met `"sudo apt-get install git"`

Maak een nieuwe map voor de github software:

```
[naam naar keuze]      "mkdir "gateway"
[Navigeer naar map]     "cd "gateway"
                        "git clone https://github.com/tftelkamp/single_chan_pkt_fwd"
```

Edit `"main.cpp"` met commando; `"nano main.cpp"` en

pas de `"configure these values!"` aan:

```
int ssPin = 6;
int dio0  = 7;
int RST   = 0;

SF7
freq = 868100000; // in Mhz for EU

No location needed

#define SERVER1      "52.169.76.203"
#define SERVER2      "router.eu.thethings.network"
#define PORT 1700
```

`"CTRL+X"` om af te sluiten, save op dezelfde naam en exit.

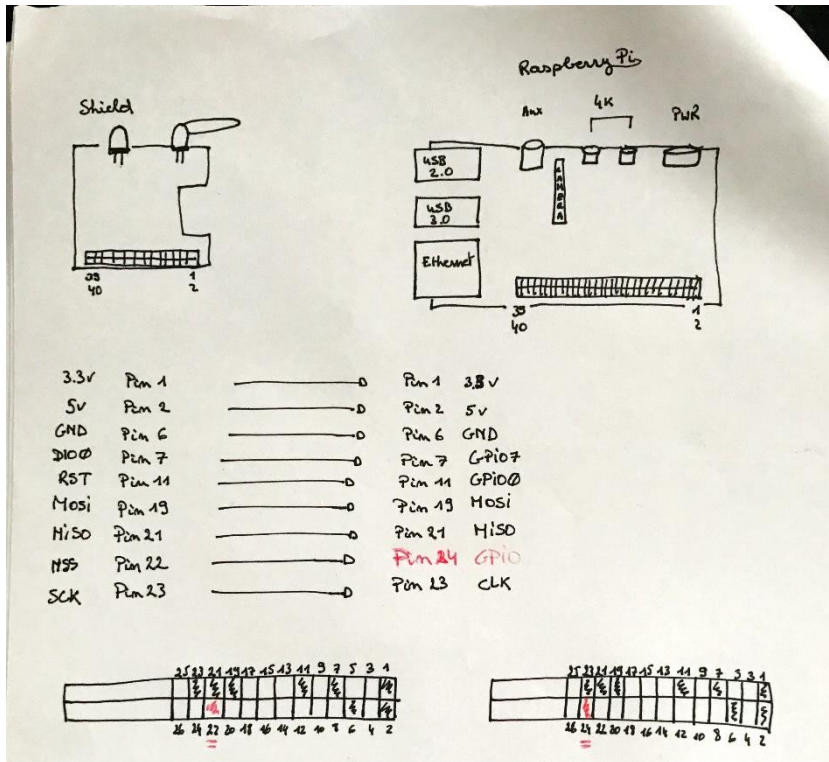
```
"make" en "sudo make install"
```

Start de service in de map waar alle software staat met: `"./single_chan_pkt_fwd"`

Manuele commando:

```
"systemctl start single_chan_pkt_fwd"
"systemctl stop single_chan_pkt_fwd"
"systemctl status single_chan_pkt_fwd"
```

d) Hardware aansluiten: Raspi aan de Dragino LoRa/GPS HAT



(bekijk de afbeelding van RPi pinouts en LoRa HAT)

Of typ in de RPi console commando `"pinout"` om alle pins te zien.

Aansluitingen gaan van RPi naar de HAT;

aanbevolen om dit met male-to-female draden te doen,

één aanpassing = **HAT pin22 naar RPi pin24**

e) Start de service:

Navigeer met `"cd gateway"` naar de map gateway

en start de service met `"./single_chan_pkt_fwd"`

Als de service `"unrecognized transceiver"` geeft, is er een fout met de hardware aansluiting.

Normaal gezien krijg je:

SX1276 detected, starting.

Gateway ID: dc..... → deze ID hebben we nodig voor het The Things Network.

Listening at SF7 on 868.100000 Mhz

stat update:

Onderbreek de service door `"CTRL+C"`

f) The Things Network Simple Gateway:

Maak een account aan op "[TheThingsNetwork](#)" en login.

in de console krijg je 2 opties: "application" en "gateway".

met de Gateway ID van hierboven gaan we luisteren naar binnen komende berichten.

Klik op "gateway" > "register gateway" en voer de "ID" in.

zorg dat je de optie: *"i'm using a legacy packet forwarder"* aanklikt!

Normaal krijg je na het registreren en het activeren van de service,

groen licht op TTN; connected.

Als je op *"traffic"* klikt,

kun je de binnen komende berichten zien verschijnen.

Als dat niet zo is, dan moet je nog een node aanmaken.

g) Maak een node met Arduino Uno:

De node is een apart toestel dat gegevens zal doorsturen via LoRa.

Klik de LoRa Shield voor Arduino OP de pinouts en

gebruik de voorbeelden uit:

download TheThingsNode library	→ via Arduino IDE > 'CTRL+SHIFT+I' > "TheThingsNetwork"
download TheThingsNetwork library	→ via Arduino IDE > ...
download LMIC library	→ via RPi " <i>git clone</i> "
	https://github.com/Kokojuice2/LoRa-Sauna.git

In dit geval gaan we temperatuur/vochtigheid doorsturen van de DHT,

Lichtintensiteit van de LDR

Relaisstatus

download DHT sensor library	→ via Arduino IDE > 'CTRL+SHIFT+I' > "DHT sensor library"
download SPI library	→ via Arduino IDE > ...
download CayenneMQTT library	→ via Arduino IDE > ...
download CayenneLPP library	→ via Arduino IDE > ...

h) The Things Network Application:

Login op TTN Console en klik op application.

voeg er een nieuwe aan toe met info:

'App ID' in kleine letters voor jezelf:

'Description' om te beschrijven wat dit gaat doen

'App EUI' zal voorzien worden door TTN

'Handler' voor EU = *ttn-handler-eu*

add application

In de app die je net aanmaakte, registreer je een nieuw toestel

'Device ID' in kleine letters voor jezelf:

'Device EUI' kan gegenereerd worden door shuffle icoon


add device

In de *'Device Overview'*, zou *'Activation Methode'* op *'ABP'* moeten staan.

met de *'NetworkKey'*, *'ApplicationKey'* en *'DeviceAddress'* info kun je

het voorbeeld uit *'LMIC library' > 'ttn-abp'* vervolledigen.

Let wel op dat deze in HEX moet staan.

Kopieer de HEX door het klikken op  in TTN en op copy.

Plak ze in de sketch en upload ze op de Arduino.

In de *'Payload Formats'*, zorg je dat CayenneLPP aangeduid staat.

In de *'Integrations'*, voeg je “myDevices” toe,

met *'ProcessID'* naar keuze voor jezelf en *'AccessKey'* default key.

In *'Settings'*, kun je nog aanpassingen doen zoals extra EUIs toevoegen,

eigendom delen en access keys aanmaken voor verschillende services.

i) Open LMIC library met voorbeeld ttn-abp, pas info aan en GO:

Zorg dat alle info juist is en upload deze naar jouw Arduino Board.

*Start de `"/single_chan_pkt_fwd"` service in de gateway map, op de RPi
en zie de pakketten binnen komen:

```
"Packet RSSI: ....
```

*Verifieer de binnekomst op TTN bij `application > device of via gateway > traffic.`

j) CayenneLPP integratie:

*Open cayenne.mydevices.com op een andere pc, login en voeg nieuw toestel toe: "RaspBerry Pi".

*Open "Putty" en activeer SSH met de IP van de RaspberryPi, login en paswoord.

*Volg de stappen op "Cayenne Dashboard" om control te krijgen over de RPi.

```
wget https://cayenne.mydevices.com/dl/.... .sh ENTER
```

*Nadien `sudo ...`

k) Cayenne Dashboard:

Om verschillende data weer te geven;

```
add Widget > Custom Widgets:  
welk 'Device'  
welke 'Data'  
welke 'Unit'
```

Sketch file om code te voorzien, upload en volg de stappen.

```
nadien add widget
```

*Omdat wij werken met de Dragino LoRa Shield voor Arduino, voeg je deze toe door:

```
add Device > LoRa > The Things Network > "Dragino Technology LoRa Shield"
```

*Kopiëer de "Device EUl" van de Arduino op TTN:

zie stap **h)**

1) Om data door te sturen naar Cayenne Dashboard, gaan we een bericht schrijven met een vaste structuur:

Bij de benodigdheden van LoRa hebben we een bericht:

```
static uint8_t mydata[17] = {0x01, 0x67, 0x00, 0x00, 0x02, 0x68, 0x00, 0x03, 0x65, 0x00, 0x00, ..., ..}
```

in vak 1, data type 67, temp 1:00, temp 2:00, in vak 2, data type 68, vochtigheid 000, in vak 3, ...

data type HEX = type, data size (= grootte in bytes)

data type 67 = temperatuur, krijgt 2 bytes (= 16 bits) ← 0,1°C signed <== als waarde 1 is = 0,1°C , als 100 is = 10°C

data type 68 = vochtigheid, krijgt 1 byte (= 8 bits) ← 0,5% unsigned <== als waarde 1 is = 0,5% , als 100 is = 50%

data type 65 = lichtsterkte, krijgt 2 bytes (= 16 bits) ← 1 lux unsigned <== als waarde 1 is = 1 lux , als 100 is = 100 lux

"data resolution per bit" = wilt zeggen hoeveel het is voor LPP als de waarde 1 is bij elke type.

meer info over data type op [CayenneLPP-data-types](#)

Arrays beginnen vanaf 0 te tellen! mydata[17](=array[16]+1)

```
"uint8_t mydata[17] =  
{0x01 - 0x67 - 0x00 - 0x00 - 0x02 - 0x68 - 0x00 - 0x03 - 0x65 - 0x00 - 0x00 - 0x04 - 0x0 - 0x00 - 0x05 - 0x0 - 0x00}"  
[0] - [1] - [2] - [3] - [4] - [5] - [6] - [7] - [8] - [9] - [10] - [11] - [12] - [13] - [14] - [15] - [16]
```

We willen de 0x00 vervangen door data die we hierboven lezen met functie temp, vocht, licht

Temperatuur gaat bv 22°C zijn → 0,1°C per bit → temp*10 = 220

Vochtigheid gaat bv 50 % zijn → 0,5% per bit → lux * 2 = 100

Lichtsterkte gaat bv 500 lux zijn → 1 per bit → 500

Het inlezen van deze gegevens doen we met:

```
"float temp = dht.readTemperature();"
"float vocht = dht.readHumidity();"
"float licht = analogRead(A0);"
```

Omdat temperatuur data 2 bytes (=16 bit) krijgt, maken we een conversie:

```
"int16_t temp_LPP, lux_LPP;"
"temp_LPP = temp*10; "
```

Omdat we willen doorgeven wanneer er een licht aangaat, gebruiken we data type digital out:

```
"lichtBinnen = digitalRead(D5);"
"lichtBuiten = digitalRead(D8);"
```

Na de conversie gaan we de 0x00 in de juiste arrays vervangen:

```
"mydata[2] = temp_LPP>>8;"
"mydata[3] = temp_LPP;"
"mydata[6] = vocht*2;"
"mydata[9] = lux_LPP>>8;"
"mydata[10] = lux_LPP;"
"mydata[13] = lichtBinnen;"
"mydata[16] = lichtBuiten;"
```

$temp_LPP \gg 8 = 00000000\ 01010101$
 $temp_LPP = 01010101\ 00000000$

m) Nadat alle software geschreven is, kunnen we testen en fine-tunen:

We hebben een probleem als de Arduino IDE aangeeft dat het 81% van de geheugen heeft gebruikt.

Hierdoor kunnen dingen niet functioneren als NU

```
if (temp > thresholdTemp) {  
    digitalWrite(Rood, HIGH);  
    digitalWrite(Groen, LOW);  
    digitalWrite(Blauw, LOW); }  
  
if (licht > thresholdLicht) {  
    digitalWrite(lichtBinnen, LOW);  
    digitalWrite(lichtBuiten, HIGH); }  
  
const int lichtBinnen = D10  
const int lichtBuiten = D11
```

→ wilt geen rood geven op pin 6,7,9
→ wilt ALTIJD Groen geven
→ flikkert bij LoRa verzending

→ hij flikkert om de 3 seconden
→ hij gaat niet aan bij de knop indrukking

Verlaag de hoeveelheid in het programma:

→ de if-voorwaardes vervangen door die van Cayenne Dashboard

Bronnen:

[Simpele voorbeelden](https://www.arduino.cc/en/Tutorial/BuiltInExamples) → <https://www.arduino.cc/en/Tutorial/BuiltInExamples>

[Single channel pkt fwd](https://github.com/tftelkamp/single_chan_pkt_fwd) → https://github.com/tftelkamp/single_chan_pkt_fwd

[Documentatie RPi HAT](http://wiki.dragino.com/index.php?title=Lora/GPS_HAT) → http://wiki.dragino.com/index.php?title=Lora/GPS_HAT

[Bouw Single channel gateway 1](http://wiki.dragino.com/index.php?title=Use_Lora/GPS_HAT_%2B_RaspberryPi_to_set_up_a_Lora_Node#Build_a_single_channel_LoRaWAN_gateway) → http://wiki.dragino.com/index.php?title=Use_Lora/GPS_HAT_%2B_RaspberryPi_to_set_up_a_Lora_Node#Build_a_single_channel_LoRaWAN_gateway

[Bouw Single channel gateway 2](https://www.instructables.com/id/Use-Lora-Shield-and-RPi-to-Build-a-LoRaWAN-Gateway/) → <https://www.instructables.com/id/Use-Lora-Shield-and-RPi-to-Build-a-LoRaWAN-Gateway/>

[Docs Cayenne Low Power Payload](https://developers.mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload-examples) → <https://developers.mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload-examples>

[Cayenne Dashboard add device](https://cayenne.mydevices.com/cayenne/dashboard/add) → <https://cayenne.mydevices.com/cayenne/dashboard/add>

[Mijn Github](https://github.com/Kokojuice2/LoRa-Sauna.git) → <https://github.com/Kokojuice2/LoRa-Sauna.git>