

Luchtkwaliteit sensors



November 28, 2017

Inhoudsopgave

Project beschrijving	3
Overzicht/analyse	4
Luchtkwaliteit meten	5
Luchtkwaliteit sensor	7
Opties	8
Uitleg prototype	9
Eigen firmware	10
PVC versie	14
Firmware uploaden	13
IoT-Platform	15
Dashboard	16
Conclusie	17
IoT Platform	17
Meten	17
Luchtkwaliteit	17

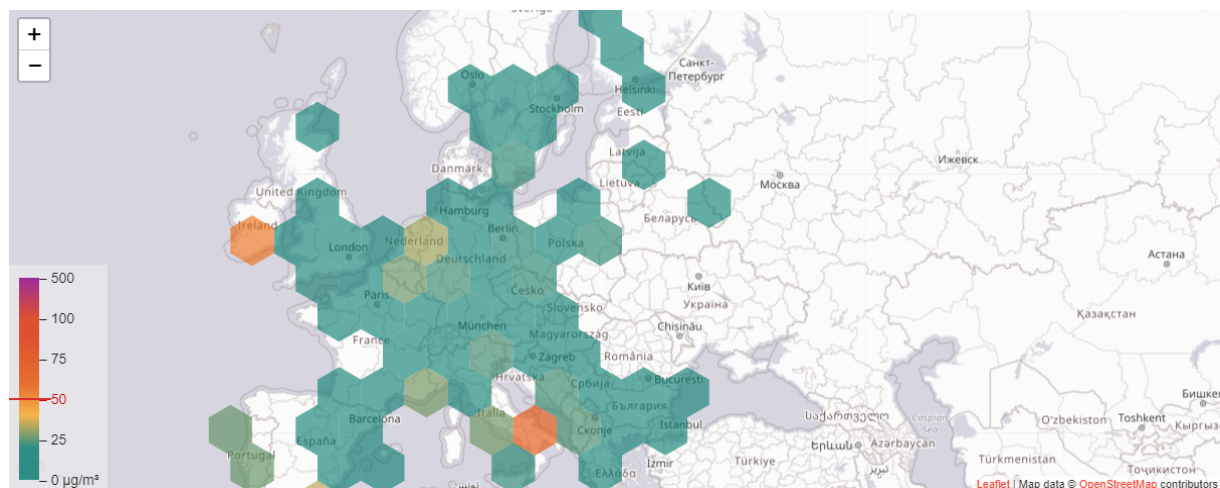
Project beschrijving

Na het project van OK LAB Stuttgart (<http://bit.ly/2FMLecL>) gezien te hebben kregen wij de wil om mee te doen.



OK LAB Stuttgart is een organisatie in Duitsland die zich momenteel bezighoudt met het meten van Fijnstof. Op hun website leggen ze uit hoe je een fijnstofmeter moet maken, ook hebben ze de software al gemaakt. Dus het enige wat je hoeft te doen is de sensors bestellen en het in elkaar zetten.

Als je mee doet met het project dan moet je wel jouw gegevens van jouw fijnstofmeter delen. Je geeft aan op welke locatie de fijnstof meter hangt. Het adres publiceren ze niet en op internet komt op 100m nauwkeurig te zien waar de meter hangt. Op deze website heb je een MAP van de wereld en je ziet waar mensen hun fijnstofmeters hebben hangen. <http://bit.ly/2BsAU4F>.



Zodra je alles af hebt kan je ook je eigen database toevoegen, hierdoor kan je de gemeten gegevens van jouw fijnstofmeter zelf gebruiken voor eigen doeleinden.



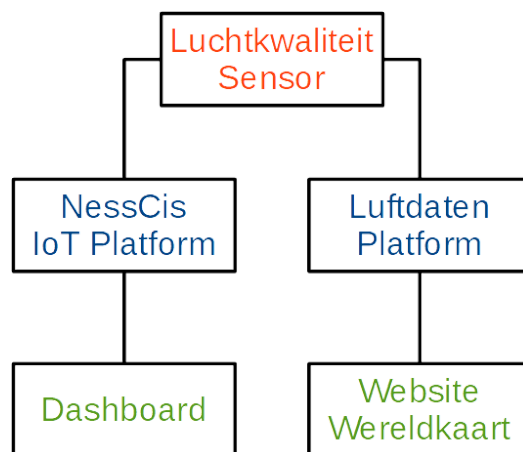
NessCis gebruikt dit project als voorbeeld om te laten zien waarvoor je hun IoT platform kan gebruiken.

Overzicht/analyse

We bouwen de sensor volgens de beschrijving van het OK LAB Stuttgart project. Deze sensor meet dan de luchtkwaliteit met fijnstof meters, luchtvochtigheid, temperatuur en luchtdruk.

Deze informatie gaat via een Wifi verbinding naar het internet.

De informatie wordt dan naar het Duitse project gestuurd maar ook naar het IoT platform van NessCis.



Luchtkwaliteit meten

De verschillende sensors meten de temperatuur, luchtvochtigheid, fijnstof en luchtdruk. Deze metingen worden als getallen doorgestuurd.

Om te begrijpen wat de getallen betekenen hebben we tabellen gemaakt met de minimum en maximum-waarden en hun betekenis.

Temperatuur			
\geq °C	< °C	Melding	Classificatie
	16	Onacceptabel	Koud
16	18	Waarschuwing	Fris
18	22	Ok	Comfortabel
22	26	Waarschuwing	Warm
26		Onacceptabel	Heet

<https://vasco.eu/nl-nl/blog/verwarming-algemeen/wat-de-ideale-kamertemperatuur-je-woon-bad-en-slaapkamer>

Luchtvochtigheid			
\geq %	< %	Melding	Classificatie
	30	Onacceptabel	Extreem droog
30	40	Waarschuwing	Droog
40	60	Ok	Comfortabel
60	70	Waarschuwing	Vochtig
80		Onacceptabel	Erg vochtig

<https://www.essent.nl/content/particulier/kennisbank/ventilatie/luchtkwaliteit-de-ideale-luchtvochtigheid-in-huis.html>

Fijnstof PM2.5			
\geq µg/m ³	< µg/m ³	Melding	Classificatie
	100	Ok	Comfortabel
100	150	Waarschuwing	Stoffig
150		Onacceptabel	Ongezond

<http://aqicn.org/faq/2013-02-02/why-is-pm25-often-higher-than-pm10/>

Fijnstof PM10			
$\geq \mu\text{g}/\text{m}^3$	$< \mu\text{g}/\text{m}^3$	Melding	Classificatie
	100	Ok	Comfortabel
100	150	Waarschuwing	Stoffig
150		Onacceptabel	Ongezond

<http://aqicn.org/faq/2013-02-02/why-is-pm25-often-higher-than-pm10/>

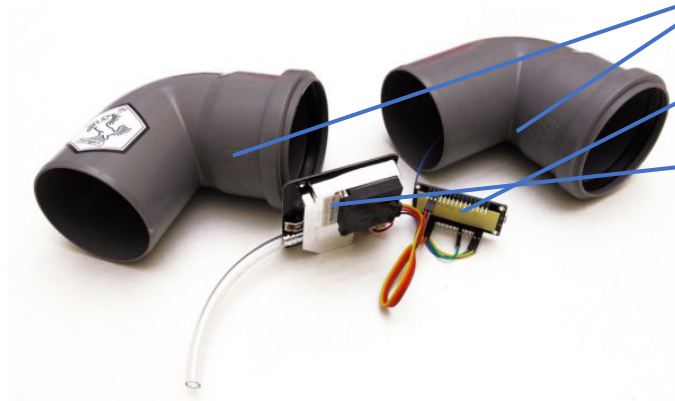
Luchtdruk			
$\geq \text{hPa}$	$< \text{hPa}$	Melding	Classificatie
	750	Onacceptabel	Gevaarlijk laag
750	900	Waarschuwing	Lage druk
900	1200	Ok	Comfortabel
1200	1300	Waarschuwing	Hoge druk
1300		Onacceptabel	Gevaarlijk hoog

<http://www.lakiere.info/AMS.htm>

<https://nl.wikipedia.org/wiki/Luchtdruk#Luchtdruk>

Luchtkwaliteit sensor

Standaard Benodigheden + kleine uitleg



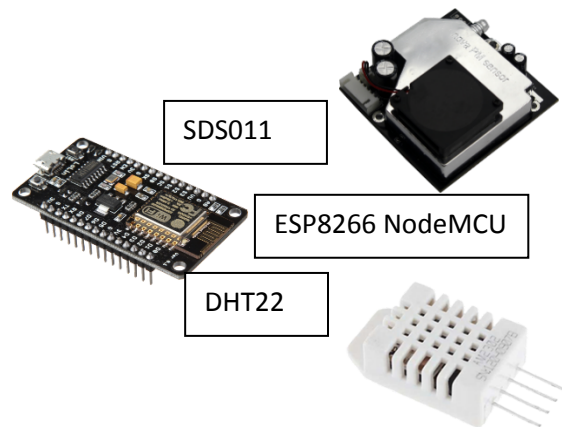
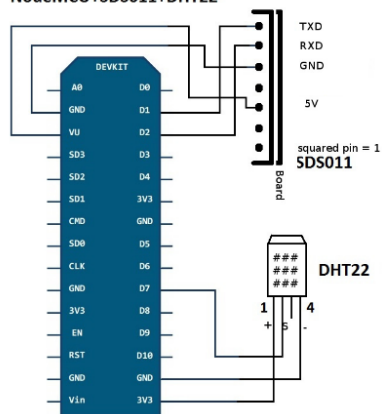
2x 70mm PVC

ESP8266 op een NodeMCU

SDS011

- SDS011 - PM2.5 / PM10
- ESP8266(NodeMCU) - De Wi-Fi transmitter
- DHT22 - Luchtvochtigheid & temperatuur
- USB kabel - stroom toevoer
- PCV - Bescherming tegen weersinvloeden
- Flexibel buisje - lucht van buiten de PVC toevoegen

NodeMCU+SDS011+DHT22



SDS011

ESP8266 NodeMCU

DHT22

Opties

Voor dit project zijn er verschillende hardware mogelijkheden.

- SDS011	Fijnstofmeter
- PMS1003, PMS5005, PMS6003, PMF7003	kleinere fijnstof meters
- PMS3003	het zelfde als bovengenoemde
- DHT22	luchtvochtigheid & temperatuur
- HTU21D	luchtvochtigheid & temperatuur
- PPD42NS	fijnstofmeter
- BMP180	temperatuur, luchtdruk & hoogte
- BMP280	luchtdruk & luchtdruk
- BME280	Temperatuur, luchtvochtigheid & luchtdruk
- DS18B20	waterdichte temperatuur meter
- GPS	Global Positioning System
- LCD 1602 (I2C: 0x27)	klein schermje om data af te kunnen lezen
- LCD 1602 (I2C: 0x3F)	wat de sensors versturen
-	

En als belangrijkste van al

- ESP8266 NodeMCU	Vertaler tussen de sensors en de database
--------------------------	--

De dik gedrukte hardware mogelijkheden gebruiken wij. Dat LCD 1602 wordt alleen gebruikt voor het prototype verder worden de SDS011, BME280 en de ESP8266 wel gebruikt voor die die buiten hangen.

Uitleg prototype sensor

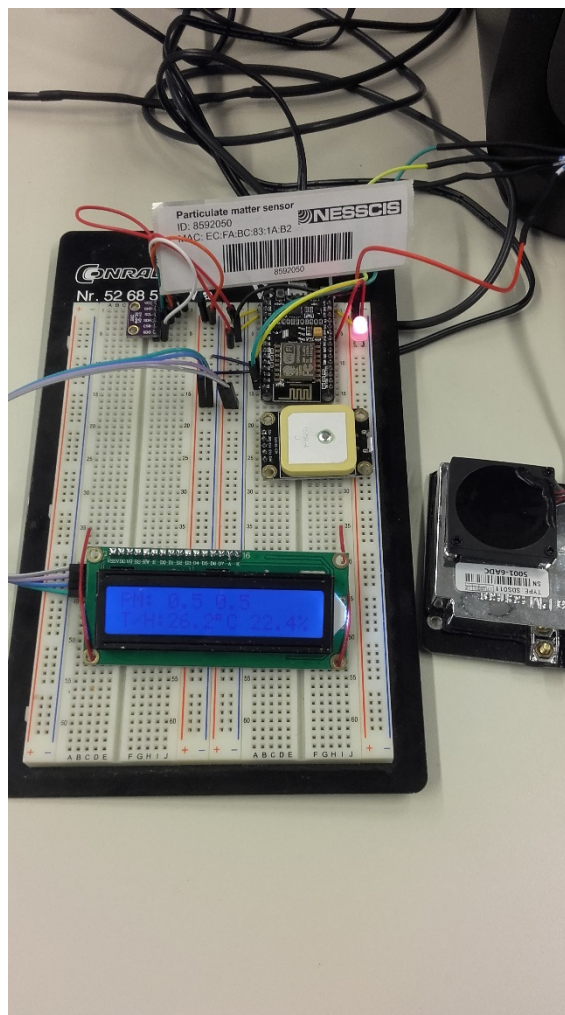
Het prototype is een beetje een test paneel. Kijken welke sensoren er goed op werken en welke niet. Daarnaast is dit prototype gemaakt om aangepast te worden. Niks is gesoldeerd en alles is verbonden via een breadboard en jumpwires.

Ook staat dit binnen en zijn de metingen niet echt van toepassing op de wereld om ons heen. Waarom heb je het dan gemaakt bedenkt u zich nu waarschijnlijk? Dat is best wel simpel eigenlijk. Door dit prototype kunnen we testen welke sensoren we wel en niet er op willen hebben, daarnaast ziet het er leuk uit.

Het prototype is makkelijker aanpasbaar dan een Fijnstofsensor die af is, hierdoor kunnen we makkelijker testen en nadenken hoe het er uit moet komen te zien.

Momenteel hebben we een BME280 en een SDS011 er op aangesloten zodat we wel gewoon metingen kunnen doen die te vinden zijn op het dashboard. Ook is er een LCD schermpje op aangesloten zodat je niet het dashboard er telkens bij moet houden om te zien of hij het überhaupt wel doet. Daarnaast ziet het er ook professioneler uit.

Hieronder is een foto te zien van het prototype.



Eigen firmware

Firmware is de software die in een apparaat zit. In het eerste prototype schrijf ik zelf de firmware om de sensors te testen.

Hieronder mijn programma code met de uitleg:

We moeten aangeven welke modules we willen gebruiken.

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <SDS011.h>
```

Hier staan de instellingen

```
//Constants
const char* mqtt_server    = "213.193.241.31";
const char* mqtt_user      = "mysql";
const char* mqtt_password  = "secret";
const char* ssid           = "NessCis";
const char* password       = "secret";

#define DHTPIN 13           // what pin we're connected to
#define DHTTYPE DHT22       // DHT 22 (AM2302)
```

Er worden variabelen gebruikt. Een variabele kan gebruikt worden om gegevens in op te slaan. Bij wiskunde heb je soms een variabele "x" waarvoor je verschillende getallen kan invullen.

```
int wifiStatus;
WiFiClient espClient;
PubSubClient client(espClient);

DHT dht(DHTPIN, DHTTYPE); //// Initialize DHT sensor for normal 16mhz
Arduino
SDS011 my_sds;

//Variables
int chk;
int error;
float hum; //Stores humidity value
float temp; //Stores temperature value
float p10;
float p25;
```

Hier worden de sensors opgestart en de Wifi verbinding gemaakt.

```
void setup() {
  my_sds.begin(D1,D2);
  Serial.begin(115200);
  dht.begin();
  delay(200);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println("Test 3-4-5");
  Serial.print("Je probeert verbinding te maken met;");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  client.setServer(mqtt_server, 1883);
}
```

Een programma functie om een verbinding te maken met de MQTT server.

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    // If you do not want to use a username and password, change next line
to
    // if (client.connect("ESP8266Client")) {
    if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

Het programma haalt alle sensor gegevens op en stuurt deze door via MQTT

```
void loop() {
    StaticJsonBuffer<300> JSONbuffer;
    JsonObject& JSONencoder = JSONbuffer.createObject();
    error = my_sds.read(&p25,&p10);
    hum = dht.readHumidity();
    temp= dht.readTemperature();
    JSONencoder["humidity"] = hum;
    JSONencoder["temperature"] = temp;
    JSONencoder["pm10"] = p10;
    JSONencoder["pm2_5"] = p25;

    Serial.println("");
    Serial.println("--+Begin+--");
    wifiStatus = WiFi.status();

    if(wifiStatus == WL_CONNECTED){
        Serial.println("");
        Serial.println("Je bent verbonden!");
        Serial.println("Je IP adres is: ");
        Serial.println(WiFi.localIP());
        reconnect();
    }
    else{
        Serial.println("");
        Serial.println("Kan niet verbinden met wifi");
    }
    Serial.println("-----");

    char JSONmessageBuffer[100];
    JSONencoder.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer));
    Serial.println("Sending message to MQTT topic..");
    Serial.println(JSONmessageBuffer);

    if (client.publish("airdata/0010fa6e384a", JSONmessageBuffer) == true)
    {
        Serial.println("Success sending message");
    } else {
        Serial.println("Error sending message");
    }
    client.loop();
    if (! error) {
        Serial.println("P2.5: "+String(p25));
        Serial.println("P10:  "+String(p10));
    }
    {
        //Print temp and humidity values to serial monitor
        Serial.print("Luchtvochtigheid: ");
        Serial.print(hum);
        Serial.print(" %, Temparatuur: ");
        Serial.print(temp);
        Serial.println(" Celsius");
        Serial.println("");
        Serial.println("--+End+--");
    }
    delay(300000); // elke 5 min
}
```

Luftdaten Firmware

In de firmware van Luftdaten zit alles wat we willen gebruiken dus die gebruiken we voor de echte versie van de sensor.

Ervan uitgaande dat Arduino al geïnstalleerd is samen met ESPTOOL. Indien dit niet het geval is moet dat eerst gebeuren want anders lukt het niet.

Om de software van Luftdaten te uploaden moet je eerst het .bin bestandje downloaden van de website. <http://bit.ly/2Iq1n6I>.

Als je dat gedaan heb moet je onthouden waar deze gedownload is want je hebt het pad nodig.

Druk dan windows-knop op je toetsenbord in tegelijkertijd met de 'r'. Nu open je UITVOEREN (Engels RUN) hier moet je de volgende 3 letters in type 'cmd' en klik dan op enter. Er verschijnt nu een klein zwart scherm; OpdrachtPrompt (Engels Command Prompt)

Persoonlijk vind ik het fijn om het volgende eerst in te type 'echo off' en natuurlijk enter, dit haalt het eerste stukje weg bij OpdrachtPrompt, zelf kan ik hierdoor makkelijk overzicht houden. Indien je dit toch gedaan heb en het terug wilt kan je 'echo on' type.

Maar hoe dan ook OpdrachtPrompt is nu geopend en dan hoeft er nog maar een paar dingetjes te gebeuren.

1. Check of de ESP8266 is aangesloten via usb aan de computer/laptop.
2. Type in
%USERPROFILE%\AppData\Local\Arduino15\packages\esp8266\tools\esptool\0.4.12\esptool.exe
-vv -cd nodemcu -cb 57600 -ca 0x00000 -cp **COM11** -cf
%USERPROFILE%\downloads\latest_**nl**.bin

Alles wat dik gedrukt is moet/kan eventueel aangepast worden. Die 0.4.12 is een versie wat bij sommige mensen kan verschillen, COM11 is welke poort dit is zo te checken bij Arduino zelf google kan veel makkelijker uitleggen hoe je dat moet checken. En last but not least, NL kan veranderen met welke taal je hem gedownload heb. Ook kan \downloads\ variëren maar zolang de downloadlocatie standaard blijft is hier geen probleem.

PVC versie sensor

Omdat we ook een sensor buiten willen hangen zonder dat de hardware aan de binnenkant beschadigd en vervolgens kapot gaat, maken we een PVC versie. We gebruiken hiervoor twee 90° hoeken van PVC.

In deze versie solderen we ook alles goed aan elkaar en maken we alles goed vast met ducttape aan de binnenkant.

In de pvc versie zitten alle hardware aan de binnenkant van twee 90° pvc hoeken. Hierdoor kan het buiten hangen zonder dat de hardware aan de binnenkant beschadigd, en vervolgens kapot gaat.

Hieronder is een foto te zien van de PVC versie.



IoT-Platform

In de firmware kan je instellen dat de meetwaarden naar Lufdaten gestuurd worden, maar ook naar een eigen systeem.

De meetwaarden worden via het HTTP protocol in JSON formaat naar het NessCis IoT platform gestuurd.

NessCis heeft een HTTP connector die deze data kan inlezen.

Alle meetwaarden van afgelopen jaar worden in een database opgeslagen. Die kunnen later weer ingelezen worden.

In het dashboard worden de waarden van de afgelopen 24 uur uitgelezen. Deze worden opgevraagd via een API (Application Programming Interface). Dit is een manier om data aan andere software door te geven.

Dashboard

Het dashboard is een website waar alle data van de sensors te zien is.

De data van de sensor('s) van NessCis is te zien op <http://bit.ly/2Imkr5J>.

Het originele project gebruikt 3 sensors een SDS011 een fijnstofmeter, een ESP8266 dit communiceert met het internet & DHT22 dit is een luchtvochtigheid en temperatuur meter. NessCis gebruikt bijna dezelfde sensoren. Ze hebben de DHT22 ingeruild voor een BME280 dit meet het zelfde maar ook de luchtdruk & op het prototype wat op kantoor staat is er een LCD schermpje aangesloten. Dan hoeft je niet het dashboard te openen om te zien wat het meet.

Zoals ik net al aangaf gebruikt NessCis de data ook voor hun eigen platform en om deze data uit te lezen hebben ze een dashboard gemaakt. Zie foto hieronder.

Donkerblauw;
Luchtdruk

Groen;
Temperatuur

Lichtblauw;
Luchtvochtigheid

Rood;
PM2.5

Geel;
PM10



In deze foto zijn er drie pieken te zien, dit heeft te maken dat er een valse meting was doordat de sensor geen stroom meer kreeg. Verder is het heel nauwkeurig.

PM10 en PM2.5 zijn fijnstof soorten, de 10 en de 2.5 staat voor de grote. Fijnstof bestaat in allemaal verschillende soorten, daarom wordt alleen de grote en hoeveelheid gemeten. Natuurlijk zijn deze waardes heel laag maar dat heeft er mee te maken dat het binnen in een kantoor staan en niet buiten in de lucht hangt.

Conclusie

IoT Platform

Het is gelukt om het IoT platform van NessCis te gebruiken voor de metingen van luchtkwaliteit. In het dashboard zijn de grafieken te zien van de metingen.

Meten

Door het dashboard kunnen we alle waarden zien van de afgelopen 24 uur. Hierin is bijvoorbeeld te zien dat er meer fijnstof gemeten wordt als er mensen op kantoor zijn.

Luchtkwaliteit

Het fijnstof viel erg mee op het kantoor. De lucht was wel droog en warm.