

Masterclass Sensoren: een fijnstofsensor bouwen

HAN College of Technology
Radboud Pre-University College

Informatica / Natuur, Leven & Technologie



Deze Masterclass bestaat uit twee sessies:

Sessie 1:

Wat zijn sensoren en hoe bouw en configureer je je eigen sensor.

Onderzoek 1: Sensor

Onderzoek 2: Fijnstof

Lab 1: Teensy microcontroller

Lab 2: Analooq of digitaal

Lab 3: Fijnstof met de Teensy meten

Lab 4: Sensor informatie op internet zetten

Sessie 2:

Onderzoek 3: Internet of Things

Wat doe je nu uiteindelijk met je meetgegevens

J.A. Korten

Versie 0.2, juli 2015

Mansveld: sneller smogalarm

06-07-2015 19:07 | ANP



beeld AFP

Er zal sneller worden gewaarschuwd voor smog. Dat heeft staatssecretaris Wilma Mansveld (Milieu) maandag aan de Tweede Kamer laten weten. Ze werkt aan een nieuw smogalarm.

In het nieuwe waarschuwingssysteem zal naast de concentraties stikstofdioxide en ozon ook de concentraties fijnstof worden gemeld. Dat is vooral belangrijk voor de ruim een

miljoen mensen met luchtwegaandoeningen.

In ons land zal net als in België al voor smog worden gewaarschuwd als er meer dan 70 microgram fijn stof in de lucht zit. Op dit moment ligt die drempel nog op 200 microgram.

Deze zomer wordt er nog een nieuwe smogapp door het ministerie van Mansveld gelanceerd. Mensen kunnen daarmee een advies ontvangen over de luchtkwaliteit in de eigen buurt.

PvdA en D66 hadden om een nieuw smogalarm gevraagd.

Sessie 1:

In deze sessie leer je hoe wat sensoren zijn en gaan we aan de slag om een werkende fijnstofsensor te maken waarmee je de luchtkwaliteit bij je thuis of op school gaat meten.

Vorbereiding:

Nodig:

- 1x Fijnstofsensor
- 1x Teensy (pjrc.com)
- 1x ESP8266 draadloze module
- 1x LiPo batterij en omzetter (LiPo Rider)
- 1x Behuizing (optioneel)
- 1x knopje, 1x weerstand 10k, draadjes

Voor het programmeren en/of opladen:

- Micro USB kabel
- Arduino software met Teensyduino uitbreiding

Je krijgt eerst een kort college over wat sensoren zijn.

Daarna ga je met je groepje de onderzoeksopdrachten uitvoeren.

Tot slot programmeer je de sensor in een hands-on sessie (na de pauze).

N.b. Je neemt na afloop de sensor mee. In je groepje verdeel je wie de sensor wanneer heeft om metingen te doen.

De volgende keer (sessie 2) neem je de sensor weer mee terug.

Onderzoeksvraag 1.

Onderzoek wat een sensor in het algemeen is en ga hierbij onder andere in op:

- Wat is de definitie van een sensor en hoe gebruiken we een sensor?
- Wat is het verschil tussen analoge en digitale sensoren?
- Wat zouden geschikte manieren zijn om de kwaliteit van de lucht te meten?

Onderzoeksvraag 2.

Onderzoek de problemen en oorzaken van fijnstof. Is er onderscheid te maken tussen 'natuurlijk' en 'onnatuurlijk' fijnstof? Zo ja, waarin bestaat dat verschil?

N.b. een interessante discussie is het verschil tussen kernenergie en 'duurzame energie' (zoals windmolens en zonnepanelen) waarbij gas en kolencentrales altijd als secundaire bron nodig blijken te zijn.

Tip! De term voor fijnstof is "particulates" in het Engels.

Zie ook:

<http://www.ncbi.nlm.nih.gov/pubmed/?term=particulate+AND+health+risks>

Scan om direct in
PubMed te zoeken...



Onderzoeksvraag 3.

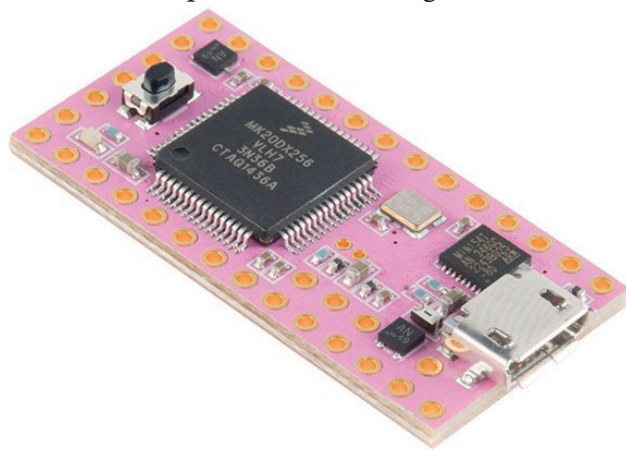
Onderzoek wat er voor manieren zijn om sensoren in een netwerk aan te sluiten. Geef aan wat de voor- en nadelen van deze oplossingen zijn.

Tip: je zou eens op de Sparkfun en Adafruit sites kunnen kijken. Denk onder andere na over zaken als connectiviteit, topologie, kosten. Je zou hier een tabel voor kunnen maken.

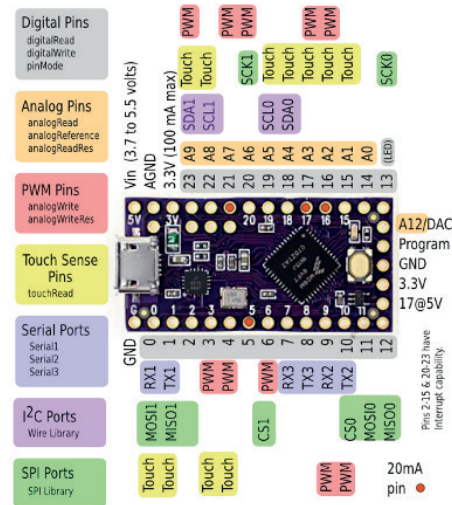


Hands-on Lab 1. Teensy microcontroller

Om een sensor te kunnen gebruiken heb je zoals je geleerd hebt in het masterclass college een verwerkingseenheid (microcontroller) nodig welke de gegevens van de sensor a. kan interpreteren en b. verder kan verwerken. Wij gebruiken hiervoor als basis een Teensy 3.1 met een ESP8266 SOC (System On a Chip). De ESP8266 is ook een microcontroller met ingebouwde WiFi optie (zie Afbeelding 3).



Afbeelding 1. Teensy 3.1 (de Anouk editie).



Afbeelding 2. Functies van de (voorkant van de) Teensy.

Je zou in theorie de ESP8266 ook los kunnen gebruiken, maar de Teensy is hiervoor prettiger omdat:

- a. je de Teensy direct kunt programmeren (is lastiger met de ESP8266)
- b. de pinnen van de Teensy 3.1 beter bestand zijn tegen 'gevaarlijke spanningen'
- c. je kunt het sensorsysteem zo mooi modulair op kunt zetten

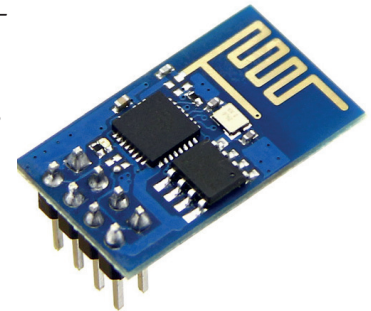
Afbeelding 2 geeft je een overzicht van de mogelijkheden van een Teensy. In deze workshop gaan we daar verder niet op in, maar een Teensy is door de vele mogelijkheden en kostprijs van zo'n 20€ heel goed te gebruiken voor allerlei projecten waaronder profielwerkstukken (Zie ook de link naar learn.sparkfun.com).

We gaan eigenlijk maar een paar opties van de Teensy gebruiken:

- a. fijnstofsensor uitlezen
- b. draadloze verbinding maken
- c. de Teensy zo programmeren dat de sensordata op internet komt

Opdracht Lab 1.1

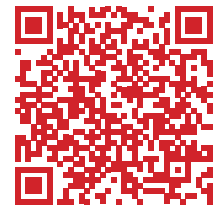
- A. Sluit de Teensy aan op je USB-poort, open de Arduino software en programmeer het programma "Blink" op de Teensy. Je ziet nu het LED-je op de Teensy knipperen.
- B. Verander de code zodat het LED-je twee keer zo snel knippert.



Afbeelding 3. ESP8266.

“Getting Started with the
Teensy”

Link: learn.sparkfun.com



Hands-on Lab 2. Sensoren intro

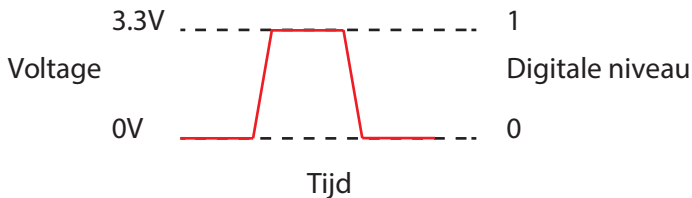
Om een microcontroller zoals van The Thing te kunnen programmeren zodat je sensor data uit kunt lezen en verwerken, moet je uiteraard eerst weten hoe sensoren werken en hoe een microcontroller sensoren kan lezen.

Achtergrond: Analooog of digitaal

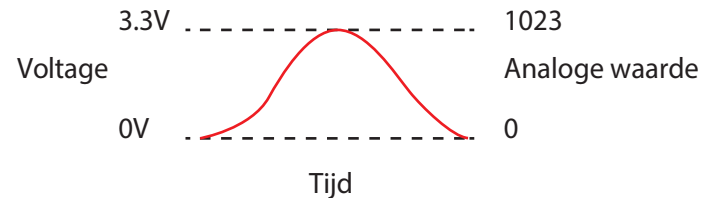
Als je met een microcontroller sensordata leest heb je te maken met:

- a. digitale inputs
- b. analoge inputs

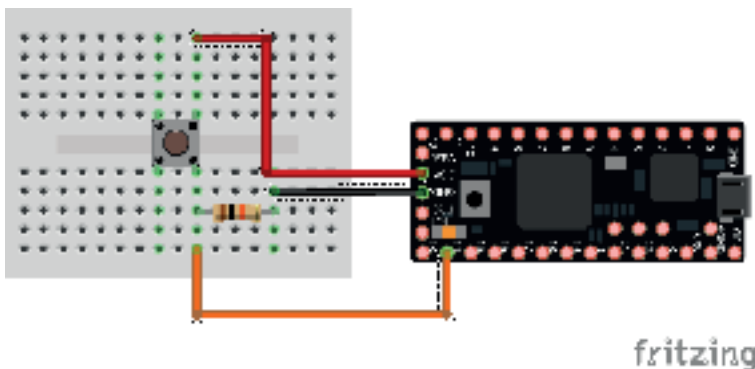
Een digitale input levert digitale sensordata op: óf 0, óf 1. Een analoge input levert analoge sensordata op. Om het verschil duidelijk te maken zie je het schema uit afbeelding 4.



Afbeelding 4. Een digitale input met bijv. een knop.



Afbeelding 5. Een analoge input.



- Je hebt drie draadjes voor een knopje:
1. een (rode) draad 3.3V (op sommige controllers 5V of soms 1.8V)
 2. een (zwarte) draad (met 10kΩ weerstand) naar 0V (heet ook GrouND)
 3. een (oranje) 'lees'-draad naar een analoge of digitale input.

Afbeelding 6. Knopje aan de Teensy μ C.

In afbeelding 4 en 5 zie je het verschil tussen een digitale en analoge input. Als je bijvoorbeeld een knopje aansluit zoals je in afbeelding 6 ziet, kun je de waarde van het knopje inlezen. De weerstand zorgt ervoor dat je geen kortsluiting krijgt, immers de stroom kiest de weg van de minste weerstand en dus ontstaat er geen kortsluiting tussen 0 en 3.3V.

Opdracht Lab 2.1

- Open de voorbeeldcode "Button" in je Arduino programma op en verander regel 29:
const int buttonPin = 2; naar *const int buttonPin = 14;*
- Upload de code naar je Teensy en observeer het resultaat (knopje indrukken / loslaten)

Reflectie op opdracht 2.1

Als je het knopje niet indrukt, lees je op je input pin 0V, immers de oranje draad is via een weerstand (bijv. 10 kilo-ohm) verbonden aan de ground-/GND-pin. Je leest nu in je programma een LOW oftewel 0. Die weerstand zit er om te zorgen dat het ook goed gaat als je het knopje indrukt: dan maakt de knop verbinding met de rode draad (3.3V) en lees je op de oranje draad dus opeens 3.3V. Als het een digitale pin is lees je nu een HIGH oftewel 1.

Bekijk het programma uit opdracht 2.1 nog eens? Begrijp je de programmeerregels met deze uitleg? Zo niet, vraag extra uitleg aan je practicumdocent/studentassistent.

Opdracht Lab 2.1: Een knop als analoge sensor.

We gaan nu een eenvoudige aanpassing doen waardoor je de knop ook als analoge sensor kunt gebruiken.

A. Pas je code aan naar de code van codevoorbeeld 1.

B. Upload je code naar de Teensy

C. Open de “Seriële monitor”

D. Observeer wat er gebeurt als je het knopje indrukt

Tip! bekijk de afbeeldingen 4 en 5 ook nog eens en betrek die bij je conclusie over labopdracht 2.1D.

```
int sensorPin = A0;    // select the input pin for the sensor
int ledPin = 13;       // select the pin for the LED
int sensorValue = 0;   // variable to store the value coming from the sensor

void setup() {
    // declare the ledPin as an OUTPUT:
    pinMode(ledPin, OUTPUT);
    Serial.begin(9600);
}

void loop() {
    sensorValue = analogRead(sensorPin); // read the value from the sensor:
    Serial.println(sensorValue);
    if (sensorValue < 100) {
        digitalWrite(ledPin, HIGH);
    } else if (sensorValue > 1000) {
        digitalWrite(ledPin, LOW);
    }
    delay(10);
}
```

Codevoorbeeld 1. Knopje analoog aan de Teensy μ C.

Reflectie op opdracht 2.2

Je hebt als het goed is waargenomen dat je als je de knop indrukt en je een analoge input gebruikt, een waarde leest die (bijna) 1023 is.

Voor een eenvoudige analoge sensor werkt het eigenlijk precies zo als met het knopje, hoewel het (natuurlijk) voor de hand ligt dat een knopje als digitale input gebruikt wordt: open/dicht, ingedrukt/niet-ingedrukt, je leest dus altijd maar óf het ene, óf het andere, wil je met andere soorten sensoren meer precisie hebben. Je wilt als je temperatuur meet liever het aantal graden meten dan 'heet' of 'koud', je wilt weten misschien wel weten hóe vochtig het is, hoe veel zonlicht je meet, hoe donker het is, hoe hard de geluiden zijn die je waarneemt, en... dus ook hoe schoon of 'smerig' de lucht is in de omgeving waarin je je fijnstofsensor hebt geplaatst.

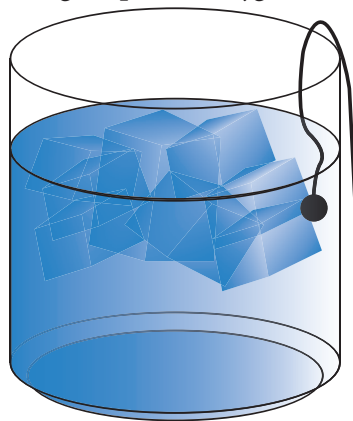
Achtergrond: Van analoog... naar digitaal

In afbeelding 5 zie je dat de meetwaarden daar tussen 0 en 1023 kunnen liggen. Écht analoog meten kunnen we helaas niet met een microcontroller: immers werkt dat ding intern zoals elke computer digitaal, dus heb je uiteindelijk altijd 0-en en 1-en waarmee je meet. Een microcontroller heeft hiervoor een ADC aan boord: een analoog naar digitaalomzetter. De resolutie van zo'n ADC is bij Arduino en ook bij de ESP8266 10-bit. Dat betekent dus dat het meetbereik (0..3.3V) opgeknipt wordt in 2^{10} -stappen (= 1024, oftewel van 0..1023). Je meet dan dus een laag getal als je sensor een lage waarde teruggeeft en een hoge waarde als je sensor een

hoog voltage uitstuurt. Afhankelijk van het soort en type sensor kun je die waarde tussen 0 en 1023 dan interpreteren: de data wordt omgezet in informatie.

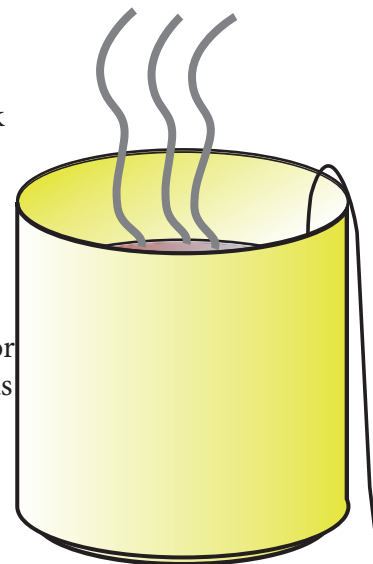
Achtergrond: Van data naar... informatie

Stel we hebben een lineaire temperatuursensor met een positieve temperatuurcoëfficiënt (oftwel: hoge temperatuur geeft je hoge meetwaarde). We lezen de analoge input en krijgen dan zeg 400 als waarde terug. Die 400 zegt ons eigenlijk niets: immers, we moeten de sensor eerst calibreren, op basis van vaste zogenaamde 'ijkwaarden' vaststellen wat de waarden zijn die bij die vaste punten horen.



Afbeelding 7a. IJswater met sensor.

In figuur 7a en 7b zie je de onze temperatuursensor in een glas ijswater van respectievelijk ca. 0° celcius en ca. 100° celcius. Als de sensor inderdaad lineair is en een bereik heeft van ≤ 0 graden en ≥ 100 graden, weet je nu dus genoeg om het hele bereik van de sensor te kunnen gebruiken. Stel de sensor geeft je de waarde 50 terug bij 0 graden en 750 bij 100 graden, dan weet je nu dat als je 350 leest dat



Afbeelding 7b. Kokend water met sensor.

de temperatuur 50°C zou moeten zijn.

Het op basis van vaste punten ijken van de sensor, noemen we dus calibreren. Helaas zijn de meeste temperatuursensoren niet lineair maar daar heb je dan weer speciale formules (zoals de Steinhart-Hart formule) voor.

Meer specialistische sensoren worden door de fabrikant gecalibreerd, dus daar hoeven wij ons niet druk over te maken, maar het is wel goed dit proces een beetje te begrijpen, zeker als je heel eenvoudige sensoren zoals temperatuur of lichtgevoelige weerstanden gebruikt.

Verwerking:

Je zou met een PTC of NTC (temperatuursgevoelige weerstand met respectievelijk een positieve of negatieve temperatuurcoëfficiënt) een testje kunnen doen zoals in afbeelding 7a en 7b te zien is. De code uit codevoorbeeld 1 is hier vrijwel helemaal voor te gebruiken.

Voor VWO-leerlingen is dit wellicht een interessante link:

https://en.wikipedia.org/wiki/Steinhart-Hart_equation

Hier is een toepassing hiervan voor Arduino:

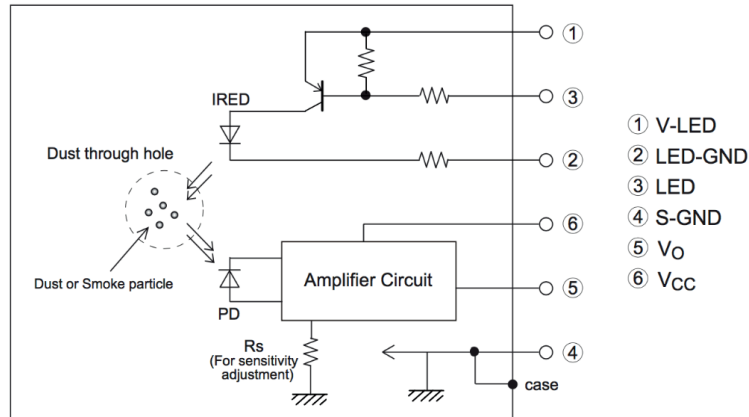
<http://playground.arduino.cc/ComponentLib/Thermistor2>

Hands-on Lab 3. De Fijnstofsensor

In deze lab-opdracht gaan we kijken naar onze fijnstofsensor en leer je deze programmeren met de Teensy.

Achtergrond: de fijnstofsensor GP2Y1010AU0F

De GP2Y1010AU0F is een sensor van het bedrijf Sharp waarmee we op een optische manier stof kunnen meten hoeveel stof er in de lucht aanwezig is.



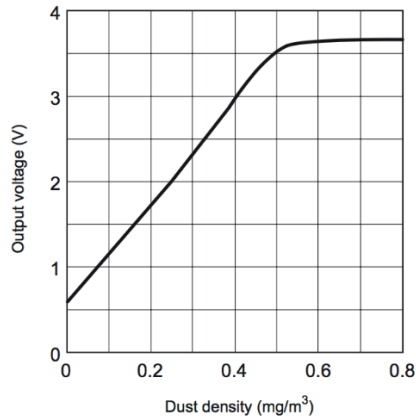
In afbeelding 8 is de interne schematische werking van de sensor te zien. De sensor zelf werkt eigenlijk heel eenvoudig.

De sensor bestaat uit:

- a. een lichtbron in de vorm van een infrarood LED
- b. een lichtgevoelige sensor

Afbeelding 8. De interne schematische werking van onze fijnstofsensor. (Sharp, 2006).

In de Sharp GP2Y1010AU0F datasheet staat naast het elektronische circuit uit afbeelding 8 ook een grafiek van hoe de sensor zich ongeveer gedraagt (afbeelding 9).



Afbeelding 9. Uitgangsspanning afgezet tegen stofdichtheid.

De tabel uit afbeelding 9 is een indicatie van welke output de sensor geeft bij welke 'stofdichtheid'.

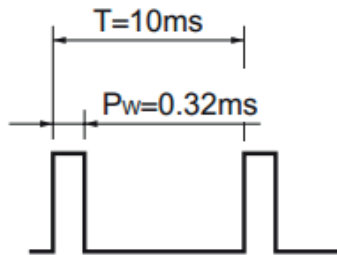
Wij voeden de sensor met een iets lager voltage (3.3V i.p.v. 5V), wat volgens de datasheet geen probleem zou moeten zijn, maar wat er wel voor zorgt dat de verdeling iets anders verloopt.

Technisch gezien zou onze meting zelfs iets nauwkeuriger moeten worden hierdoor (waarom?!). Wel zal de infrarood LED mogelijk iets minder fel branden waardoor je iets minder intensief meet. De inputs van de Teensy 3.1 zijn wel bestand tegen 5V, maar omdat de Teensy bedoeld is voor 3.3V is het beter de sensor iets minder spanning te geven.

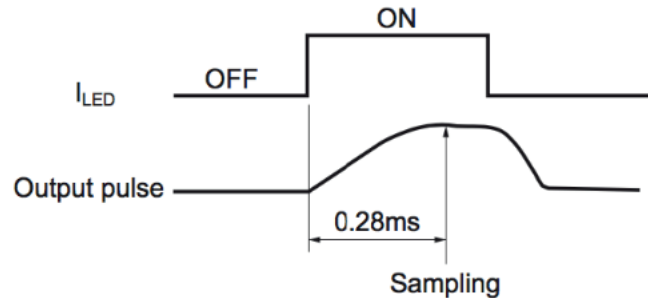
Programmeren: de datakant van de fijnstof sensor GP2Y1010AU0F

Volgens de datasheet moeten we de interne LED aanzetten (afbeelding 10), dan 280 μ s (microseconden) wachten (afbeelding 11) en dan het uitgangssignaal meten. De totale meettijd moet 320 μ s zijn, dus na het meten wachten we 40 μ s en beginnen we een nieuwe meetcyclus. In afbeelding 10 zie je hoe we dat de meetcyclus totaal 10ms in beslag neemt. Daarvan is de IR-LED 320 μ s aan: $(100\% / \text{totale tijd } 10) * 0,32 \text{ ms}$ geeft ons de duty cycle (3,2%).

Pulse-driven wave form



Afbeelding 10. De totale periode en de 'duty cycle' van de infrarood LED.



Afbeelding 11. De meting binnen de actieve fase van de cyclus.

De volgende stap is om dit daadwerkelijk te programmeren.

Inmiddels weet je hoe je op de Teensy een LED moet laten knipperen.

De infrarood LED hebben we aangesloten op pin digitale pin 12 en de meetpin op A6 (20).

```
/*  
  Sketch om een met een Teensy (Arduino compatible ARM) (www.pjrc.com)  
  de gegevens van een Sharp GP2Y1010AU0F stofsensor te lezen  
  Hogeschool van Arnhem en Nijmegen  
  Johan Korten, okt. 2015  
  v1.0  
  
  Based on code by Cyrille Médard de Chardon (serialC), Christophe Trefois (Trefex)  
  from  
  https://github.com/Trefex/arduino-airquality/  
  */  
  
#define SIDLE 0  
#define SIRON 1  
#define SDELTA 2  
#define SIROFF 3
```

Codevoorbeeld 2. De fijnstofsensor lezen met Teensy

```
const int measurePin = A6;
const int ledPower = 12;
const int heartBeat = 13;

const int samplingTime = 280;
const int deltaTime = 40;
const int sleepTime = 9680; // 10.000μS - samplingTime (280μS) - sleepTime (40μS)

const int sensorDelayTime = 1000; // na elke elke seconde lezen (in milliseconds)

float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;

int dustSenseState = 0; // 0: IDLE, 1: IR ON, 2: deltaTime, 3: IR LED off

long dustSenseTimer = 0;

unsigned long heartBeatTimer = 0;
int heartBeatTime = 10000;
int heartBeatTimeOff = 100;
```

Codevoorbeeld 2 - vervolg. De fijnstofsensor lezen met Teensy

```

void setup() {
    Serial.begin(9600);
    pinMode(ledPower, OUTPUT);
    pinMode(heartBeat, OUTPUT);
}

void loop() {
    takeSamples();
    //wifiAction();
    flashHeartBeat();
}

void flashHeartBeat() {
    if (millis() - heartBeatTimer > heartBeatTime) {
        digitalWrite(heartBeat, HIGH);
        if (millis() - heartBeatTimer > heartBeatTimeOff) {
            heartBeatTimer = millis();
            digitalWrite(heartBeat, LOW);
        }
    }
}

```

Codevoorbeeld 2 - vervolg. De fijnstofsensor lezen met Teensy

```

void takeSamples() {
    boolean transistion = false;
    if (dustSenseState == sIDLE) {
        if (millis() - dustSenseTimer > sensorDelayTime) { // slaaptijd tussen leescycles
            dustSenseState = sIRON;
            dustSenseTimer = micros();
            digitalWrite(ledPower, LOW); // sensor gaat dan AAN (via transistor, AAN = LOW)
        }
    } else if (dustSenseState == sIRON) {
        if (micros() - dustSenseTimer > samplingTime) {
            dustSenseState = sDELTA;
            dustSenseTimer = micros();
            voMeasured = analogRead(measurePin);
        }
    } else if (dustSenseState == sDELTA) {
        if (micros() - dustSenseTimer > deltaTime) {
            dustSenseState = sIROFF;
            dustSenseTimer = micros();
            digitalWrite(ledPower, HIGH); // sensor gaat dan UIT (via transistor, UIT = HIGH)
        }
    } else if (dustSenseState == sIROFF) {
        if (micros() - dustSenseTimer > sleepTime) { // minimale slaap tijd
            dustSenseState = sIDLE;
            dustSenseTimer = millis();
        }
    }
}

```

Codevoorbeeld 2 - vervolg. De fijnstofsensoren lezen met Teensy

```

    dustDensity = 0.17 * calcVoltage - 0.1;
    Serial.print("Raw Signal Value (0-1023): ");
    Serial.print(voMeasured);

    Serial.print(" - Voltage: ");
    Serial.print(calcVoltage);

    Serial.print(" - Dust Density: ");
    Serial.println(dustDensity);
  }
}
}

```

Codevoorbeeld 2 - vervolg. De fijnstofsensoren lezen met Teensy

In codevoorbeeld 2 zien we hoe de sensor geprogrammeerd kan worden op de Teensy om daadwerkelijk metingen te doen.

Is je dat gelukt is, is de belangrijkste stap voor elkaar: de sensor kan daadwerkelijk gebruikt worden.

```

// Er is blijkbaar nog iets niet helemaal goed want de sensoren geven nog een afwijkende waarde
// 1 nov. 2015

```


Masterclass
Een fijnstofsensoren bouwen.

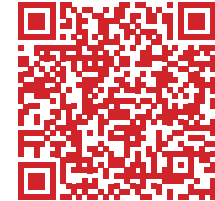
Betasteunpunt Arnhem/Nijmegen
HAN College of Technology / Radboud Pre-University College

Hands-on Lab 4. Internet of Things

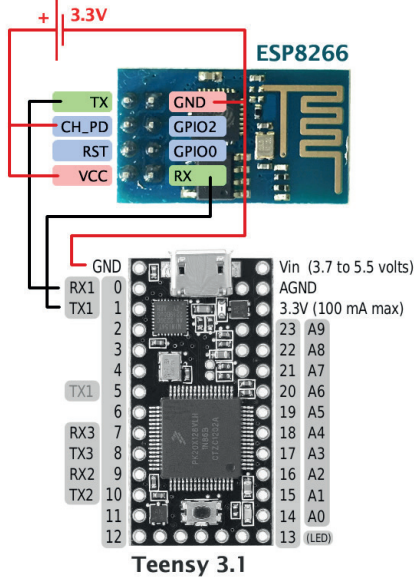
De laatste stap is de data van de fijnstofsensor op internet zien te krijgen.

We hebben hiervoor twee dingen nodig:

- A. Een internetverbinding naar de Teensy met fijnstofsensor.
- B. Een plek om de data naar toe te kunnen sturen en te delen.



Teensy 3 en ESP8266



Afbeelding 12. Zo sluit je de ESP8266 aan op de Teensy.

We beginnen met het aansluiten van de ESP8266 op de Teensy (afbeelding 12). We gebruiken seriële communicatie waarbij we de verzendlijn (TX) van de ene verbinden met de ontvanglijn van de andere (RX). De TX van de Teensy moet dus op de RX van de ESP8266 en andersom.

De ESP8266 moet gevoed worden met 3.3V. Omdat het zenden best veel stroom kost (ca. 300mA) kun je die niet direct van de Teensy afhalen. We hebben daarom een chipje tussen de LiPo rider en de ESP8266 gezet die 5V omzet naar 3.3V.

ToDo:

- testopstelling
- alles aan elkaar knopen
- intermediate server die sensor data op google maps zet of alternatief zoals thinkspeak / Pachube

Op het forum van de Teensy (forum.pjrc.com) is een “Guide To Using ESP8266 with Teensy 3” te vinden (<https://forum.pjrc.com/threads/27850-A-Guide-To-Using-ESP8266-With-TEENSY-3>).

Meer interessante links:

<http://www.cse.dmu.ac.uk/~sexton/ESP8266/>

<https://forum.pjrc.com/threads/26873-ESP8266-with-Teensy>

<https://forum.pjrc.com/threads/28846-Teensy-3-1-with-ESP8266-Web-Server>

<http://www.instructables.com/id/ESP8266-Wifi-Temperature-Logger/?ALLSTEPS>

Pachube

Masterclass
Een fijnstofsensoren bouwen.

Betastepunt Arnhem/Nijmegen
HAN College of Technology / Radboud Pre-University College

Literatuurlijst

Sharp (2006). *GP2Y1010AU0F*. Compact Optical Dust Sensor. Betrokken op 6 juli 2015 van http://www.sharpsma.com/webfm_send/1488

<http://www.howmuchsnow.com/arduino/airquality/>



Deze workshop is ontwikkeld door HAN ICA in opdracht van het Betastepunt Arnhem / Nijmegen. Het Betastepunt Arnhem en Nijmegen is een gezamenlijk initiatief van de HAN en Radboud Universiteit.
Licentie: Creative Commons Share Alike 3.0