

Overdracht

Project Snuffelneus

| | | |
|--------------------------|---|---|
| Projectleider | : | Björn Braber |
| Projectleden | : | Peter Baan Pim van Dijk Raymen Scholten Edwin Schriek Paul Zoetewey |
| Begeleider | : | Maarten Dubbeld |
| Opdrachtgever | : | Bas Flipsen |
| Vak | : | ELEEEP71b |
| Datum van uitgave | : | 7 februari 2014 |

© 2013 - 2014 Projectgroep Snuffelneus

Versiebeheer

| Versie | Datum | Aanpassing | Auteur |
|--------|------------|--|---------------|
| 0.1 | 24-01-2014 | Opzetten document + eerste content | Pim van Dijk |
| 0.2 | 06-01-2014 | Aanvulling content & referenties | Björn Braber |
| 0.3 | 06-02-2014 | Handleiding app | Edwin Schriek |
| 1.0 | 07-02-2014 | Handleiding snuffelneus, toevoegen hardware en server gedeelte | Björn Braber |

Inhoudsopgave

| | | |
|----------|---|----------|
| 1 | Introductie | 1 |
| 1.1 | Doel van project | 1 |
| 1.2 | Project omvang en product eigenschappen | 1 |
| 2 | Beschrijving | 3 |
| 2.1 | Producten | 3 |
| 2.2 | Samenwerking Holstcentre | 3 |
| 3 | Handleiding Snuffelneus | 4 |
| 3.1 | Introductie | 4 |
| 3.2 | Knoppen | 4 |
| 3.3 | Led indicatie | 4 |
| 4 | Handleiding Android app | 6 |
| 4.1 | Gebruikers interactie | 6 |
| 4.2 | Pagina's | 7 |
| 4.2.1 | Google Maps pagina | 7 |
| 4.2.2 | Info pagina | 7 |
| 4.2.3 | Settings | 7 |
| 4.2.4 | Developer | 7 |
| 5 | Hardware ontwerp | 8 |
| 5.1 | USB & LiPo Charger | 8 |
| 5.2 | Power enable & DC-DC converter | 8 |
| 5.2.1 | Diode brug | 8 |
| 5.2.2 | Schmitt trigger | 8 |
| 5.2.3 | Verbetering 1 | 9 |
| 5.2.4 | Verbetering 2 | 9 |
| 5.3 | Microcontroller | 9 |
| 5.4 | Bluetooth | 9 |
| 5.5 | Programmer interface | 9 |
| 5.5.1 | Probleem | 9 |
| 5.6 | NO ₂ sensor meetcircuit | 9 |
| 5.6.1 | Stroombron | 10 |
| 5.6.2 | Referentiespanning | 11 |
| 5.6.3 | Instrumentatieversterker | 11 |
| 5.6.4 | ADC | 13 |
| 5.6.5 | Flexibiliteit | 13 |
| 5.6.6 | Problemen en verbeterpunten | 13 |
| 5.7 | NO ₂ -sensor-heater & Battery pack | 13 |
| 5.7.1 | Probleem | 13 |
| 5.8 | User(visual) | 13 |
| 5.9 | Humidity & temperature sensor | 14 |
| 5.10 | Overig | 14 |

| | | |
|----------|---|-----------|
| 5.10.1 | MOSFET, enable van modules | 14 |
| 5.11 | Samenvatting aanpassingen en verbeterpunten | 14 |
| 6 | Handleiding server | 15 |
| 6.1 | Introductie | 15 |
| 6.2 | Database | 15 |
| 6.3 | API | 15 |

1 Introductie

In dit document vindt u de overdracht van Project Snuffelneus. Hierin wordt uitgelegd hoe het systeem in elkaar zit en welke keuzes gemaakt zijn.

1.1 Doel van project

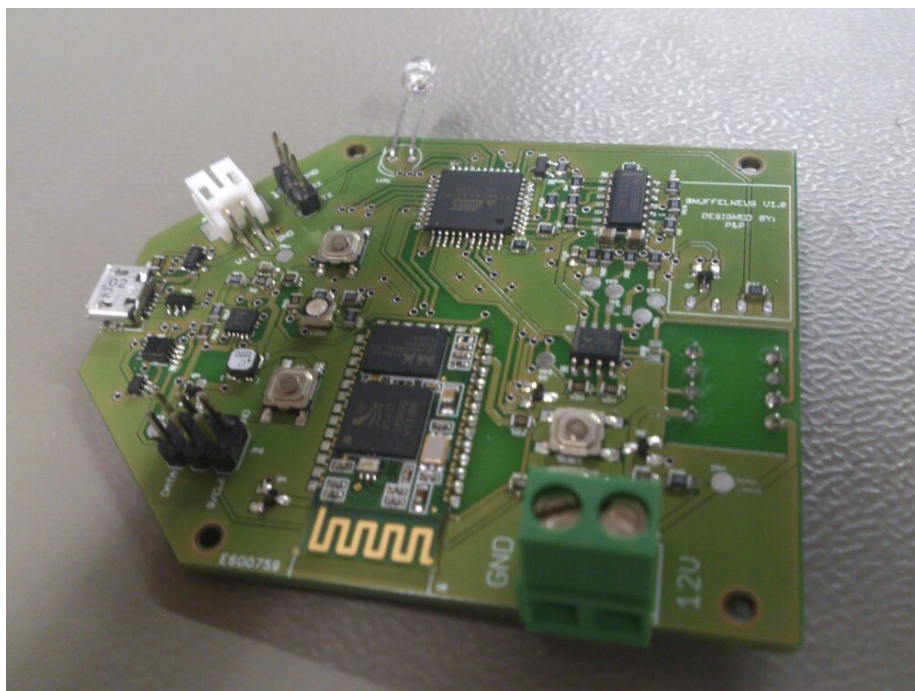
Het doel van het project is het maken van een systeem waarmee de luchtkwaliteit kan worden gemeten. Het systeem omvat een embedded system met een sensor, een Android telefoon applicatie en een webserver.

1.2 Project omvang en product eigenschappen

Het project omvat het maken van alle drie de onderdelen welke hierboven zijn benoemd. Het embedded system, "de Snuffelneus", stuurt de sensordata naar de Android telefoon applicatie. De telefoon bevat een app welke de gegevens opvangt. Vervolgens stuurt de app de sensordata door naar de webserver, waar een database op staat. Op deze manier kan alle data eenvoudig en overzichtelijk worden opgeslagen. De luchtkwaliteit wordt gemeten aan de hand van drie waarden, te weten de temperatuur, luchtvochtigheid en de hoeveelheid fijnstof(NO_2) in de [1].

De definitie voor luchtkwaliteit is vooralsnog niet gegeven. Aan de hand van een vergelijking tussen schadelijke gassen is het begrip luchtkwaliteit in deze contex gedefinieerd. Hierbij is gebleken dat NO_2 een zeer schadelijke stof is en om die reden relevant om te meten.[5] Voor het meten van luchtkwaliteit is na het formuleren van de definitie onderzocht welke hardware er gebruikt moet worden om dit te kunnen meten. Het onderzoek kan worden gevonden in [3]. Hierin staan ook welke keuzes zijn gemaakt betreffende communicatie en sensoren. Aan de hand van alle resultaten is het ontwerp van de snuffelneus gemaakt [4].

In de zip bestanden staan alle bestanden over dat betreffende onderdeel.



Figuur 1: Het printplaatje van de Snuffelneus.

2 Beschrijving

2.1 Producten

Het systeem omvat de Snuffelneus, een telefoon applicatie en een webserver. De Snuffelneus meet de temperatuur, luchtvochtigheid en NO_2 gehalte in de lucht. Dit wordt vervolgens doorgestuurt naar de telefoon. De telefoon stuurt de data vervolgens door, samen met andere informatie zoals gps locatie, tijd en gebruikers identifier, op naar de webserver. In het kort zijn dit de producten welke binnen ons project vallen:

- De gegevens moeten worden opgeslagen in een centrale database.
- Een telefoon applicatie moet worden gemaakt waarmee de gemeten gegevens kunnen worden weergegeven.
- De Snuffelneus moet de luchtkwaliteit meten, de te meten waarden moeten worden onderzocht.

De bovengenoemde punten zijn verder toegelicht in de komende hoofdstukken, waarbij ook wordt uitgelegd hoe alles in gebruik moet worden genomen.

2.2 Samenwerking Holstcentre

voor het meten van NO_2 is er gekozen een samenwerking aan te gaan met het bedrijf genaamd Holstcentre. De nadelen die andere NO_2 sensoren met zich meebrachten konden hierdoor worden weggenomen[5]. Desondanks verbruikt deze sensor in dit statium nog relatief veel energie. In de toekomst verwacht men deze sensor zeer zuinig te kunnen maken [2]. Met dit perspectief is gekozen om gebruik te maken van de Sensor van Holstcentre. Samen met dit bedrijf is de sensor in een verkleind formaat gemaakt voor dit project. Hierbij hebben wij zes stuks mogen ontvangen.

3 Handleiding Snuffelneus

3.1 Introductie

De snuffelneus is zo gebruiksvriendelijk mogelijk ontworpen. Hiermee wordt bedoelt dat de gebruiker door enkele handelingen in staat moet zijn een meting te kunnen doen. Om de gebruiksvriendelijkheid te bewaken tijdens de ontwikkeling werd er nauw samengewerkt met drie studenten van de opleiding industrieel product ontwerp. Het uiteindelijke product had één meerkleurige indicatie led en vier knoppen, waarvan twee beschikbaar voor de gebruiker.

3.2 Knoppen

Hieronder staat beschreven waar welke knop voor dient. Hierbij worden de posities van de knop beschreven wanneer men de usb aansluiting aan de linker kant houdt.

- Fiets licht (links boven)
De knop is bedoeld om het fietslicht aan of uit te schakelen. Uit het onderzoek van de studenten van IPO is gebleken dat het apparaat naast het meten van de luchtkwaliteit een extra functionaliteit moest bevatten. Dit omdat mensen anders niet worden gemotiveerd om dit apparaat op hun fiets te bevestigen.
- Aan/uit schakelaar (links onder)
Deze knop, samen met de knop voor het fietslicht is beschikbaar voor de gebruiker (rekening mee gehouden in de behuizing). Hiermee kan de snuffelneus worden ingeschakeld waarna er verbinding mee gemaakt kan worden. Het apparaat is direct na inschakelen gereed voor gebruik.
- Bootloader (rechts boven)
Deze knop is samen met de Reset knop niet bedoeld voor gebruikers. Wanneer men deze knop indrukt en tegelijk de usb kabel in de snuffelneus steekt zal deze geprogrammeerd kunnen worden. Deze functie is tijdens de realisatie echter niet meer geïmplementeerd vanwege tijdgebrek.
- Reset knop (rechts onder)
Met de reset knop is het mogelijk het apparaat te resetten wanneer nodig.

3.3 Led indicatie

Aan de hand van de meerkleurige led is het mogelijk om de gebruiker op de hoogte te stellen van een bepaalde status waarin het apparaat zich bevind. Bij de acceptatie test zijn de onderstaande kleuren/ combinaties gebruikt. Uiteraard zijn er meerdere mogelijkheden betreft de status indicatie.

- Rood knipperen twee maal
Wanneer het apparaat wordt opgestart zal deze de batterij meten of deze

vol genoeg is om op te functioneren. Wanneer het apparaat een lage spanning meet zal de LED twee maal rood knipperen en zal de snuffelneus zichzelf uitschakelen.

- Constant blauw
het apparaat is ingeschakeld.
- Constant Geel
De opwarm procedure is bezig. Om zo nauwkeurig mogelijk te meten wordt de desbetreffende sensor voorverwarmd. gedurende deze opwarming is de led geel gekleurd. Wanneer het verwamen klaar is zal de led weer blauw kleuren.

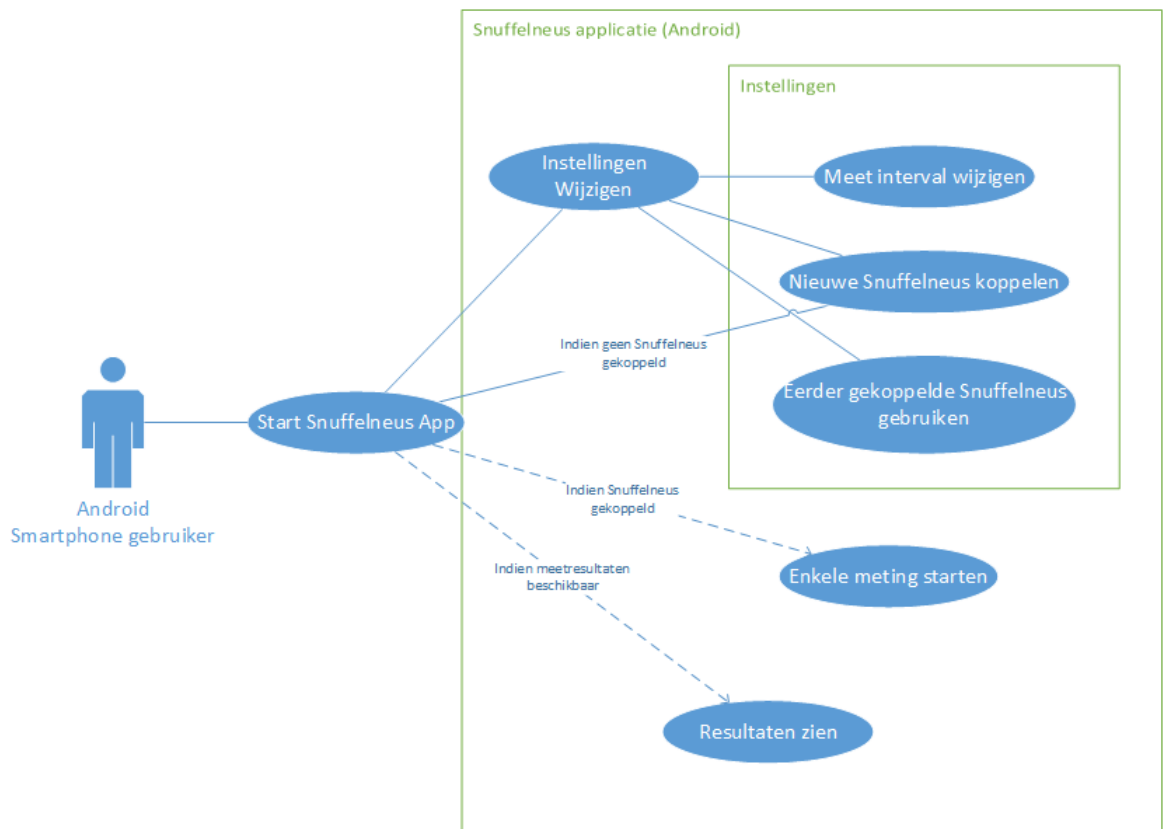
4 Handleiding Android app

In dit hoofdstuk zal kort maar krachtig de werking van de app uitgelegd worden. De app is gemaakt voor het Android platform voor versies 4.0 en hoger. Uitwerkingen van deze keuzes zijn te vinden in het design document van project Snuffelneus. Het idee van de app is simpel; de app moet als het ware met de Snuffelneus hardware kunnen communiceren en deze gecommuniceerde gegevens op slaan in een lokale en externe database.

Wat betreft de architectuur van de app kunt u het architectuur design document lezen.

4.1 Gebruikers interactie

De simpele gebruikers interactie kunnen wij weergeven met het volgende Use-Case diagram.



Figuur 2: Use-case diagram Snuffelneus App

Wat niet beschreven is in het Use-Case diagram is dat het meet proces zal starten wanneer de app opstart en indien er een Snuffelneus gekoppeld is. Het meet proces wordt uitgevoerd door de Backgroundservice, deze service zal ook draaien wanneer de Snuffelneus app niet in de voorgrond is. Dit meet proces zal door blijven gaan met een gefixeerd interval tot de menu optie "Stop scheduler" wordt ingedrukt. Dit interval is normaal in te stellen in de menu optie "Interval", echter doet deze menu optie nu nog niks.

4.2 Pagina's

4.2.1 Google Maps pagina

Op de pagina van de kaart zijn de locaties te vinden waar een meting is gedaan. Door middel van een marker wordt weergegeven waar een meting is gedaan en hoe goed de luchtkwaliteit is. Op de marker kan worden getikt, waardoor een scherm opent waar alle meetinformatie instaat. Op dit moment focust de kaart automatisch op Rotterdam.

4.2.2 Info pagina

Op de info pagina is informatie te vinden over de app.

4.2.3 Settings

In de settings kunnen verschillende instellingen worden aangepast. Zo kan een nieuwe snuffelneus worden gekoppeld, een oude snuffelneus worden gekoppeld en kan de scheduler (automatisch metingen uitvoeren) worden gestopt.

4.2.4 Developer

Developers kunnen commentaar vinden in de code. Hierin kan worden gevonden waar wat voor dient.

5 Hardware ontwerp

In dit hoofdstuk worden de hardware onderdelen uitgelegd met de problemen en verbeteringen. Bij het lezen van dit hoofdstuk is het handig om het elektrisch schema er naast te houden die in de bijlagen te vinden is.

5.1 USB & LiPo Charger

Via de USB wordt de module opgeladen en kan hij geprogrammeerd worden. Te zien in deze schakeling is dat er een ESD beveiliging zit om de achterliggende componenten te beschermen. De MCP chip is de oplader van de LI-PO batterij. De stroom waarmee hij oplaadt is net iets lager dan 500 mAh wat het maximum is wat een USB poort kan leveren.

5.2 Power enable & DC-DC converter

Een eis was dat de complete snuffelneus uit zichzelf kan uitschakelen. De werking van deze schakeling is als volgt: De DC-DC converter werkt doordat er een van de v4 ingangen hoog zijn:

- Button 1 is ingedrukt
- Button 2 is ingedrukt
- USB is ingeplugd
- De microcontroller houdt een pin hoog

Wanneer de knop bijvoorbeeld ingedrukt wordt gaat het signaal via de diode brug naar de enable van de DC-DC onverter. De microcontroller gaat aan en stuurt direct een hoog signaal via dezelfde diodebrug naar de DC-DC converter waardoor de modulen aan blijft. Het voordeel hiervan is dat de MCU nu zelf kan bepalen wanneer de hele modulen uit moet door simpelweg de enable pin laag te maken.

5.2.1 Diode brug

Er is gekozen voor een diode brug met 4 ingangen en 1 uitgang met een lage spanningsval omdat er een minimale spanning op de enable moet komen en wanneer er een de spanning van een bijna lege accu komt via de schakelaar via de diode brug moet deze spanning nog wel hoger dan de threshold van de enable pin zijn.

5.2.2 Schmitt trigger

De buttons en de USB gaan ook naar de microcontroller aangezien de spanning varieert is er gekozen oor een schmitt trigger om er zeker van te zijn dat de MCU aan de ingang altijd een spanning krijgt van 3.3 volt.

5.2.3 Verbetering 1

We hebben de DC-DC converter proberen in en uit te schakelen met een MOSFET tussen de min en de GND, dit werkte niet. Het signaal dat naar de GATE van de MOSFET gaat had direct naar de ENABLE van de DC-DC converter kunnen gaan. Dit hebben we dan ook aangepast op de printplaat.

5.2.4 Verbetering 2

Pull-down weerstanden toegevoegd voor de schmitt-trigger.

5.3 Microcontroller

Eisen waar op geselecteerd is:

- Energie zuinig
- Voldoende pinnen, ADC kanalen, USART
- USB enabled om te programmeren via USB
- AVR: makkelijk programmeerbaar (standaard)

5.4 Bluetooth

Om te communiceren met een mobieltje is er gekozen voor Bluetooth 2.0 omdat dit een standaard is en door ieder mobieltje wordt ondersteund. Ook is er gekozen voor een kant en klare kleine module die direct op de pcb gesoldeerd kon worden, dit is gedaan om dat het relatief goedkoop is en het een hoop problemen voorkomt.

5.5 Programmer interface

5.5.1 Probleem

De bootloader hebben we niet aan de praat gekregen dus hebben we altijd moeten programmeren via de 6 pin header met een externe programmer in plaats van de bootloader knop in te drukken en dan te programmeren via de USB.

5.6 NO_2 sensor meetcircuit

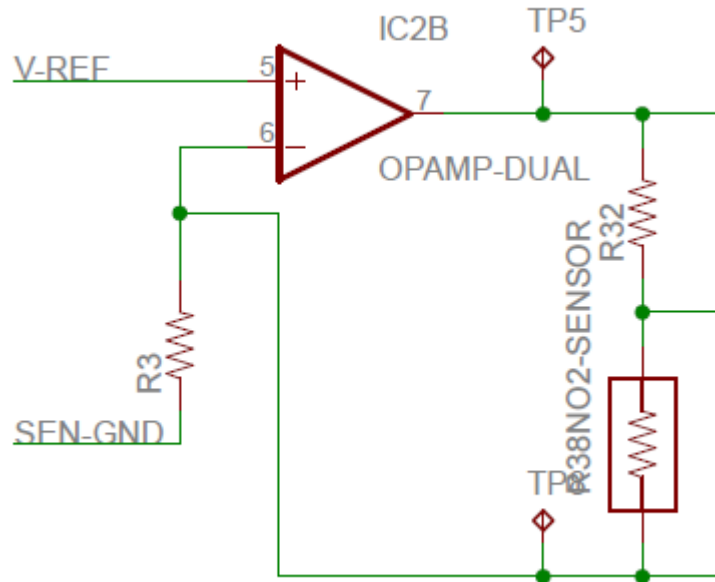
De gebruikte sensor is nog in ontwikkeling. Dit betekend dat er weinig bekend was/is over de technische specificaties van de sensor. Bij het ontwerp is hier rekening mee gehouden. Het resistieve bereik van de sensor hangt sterk af van de temperatuur en lichtval. Het circuit is zo flexibel mogelijk ontworpen zodat het meetbereik gemakkelijk kan worden aangepast met het veranderen van een aantal weerstanden.

5.6.1 Stroombron

Een constante stroombron zorgt voor een constante stroom door de sensor. Door de spanning te meten over de sensor kan de weerstand bepaald worden. In serie met de sensor is een nauwkeurige bekende weerstand (R_{32}) geplaatst. Door de spanning over deze bekende weerstand te meten kan de stroom bepaald worden. De stroom wordt ingesteld zie formule (1)

$$\frac{V_{ref} - Opam_offset}{R_3} \quad (1)$$

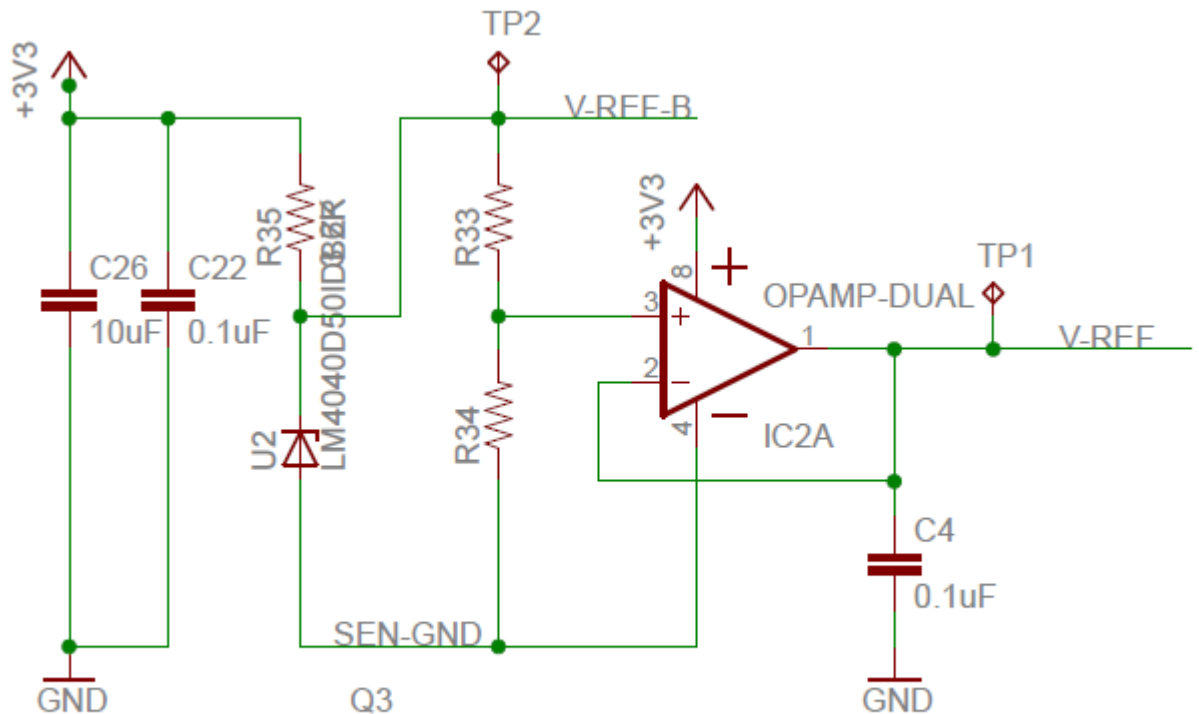
Het sensorbereik is gekozen op $20k\Omega$. De voedingsspanning is 3.3V, tussen TP5 en TP3 is maximaal 2.3V (3.3-vref) vanwege de spanningsdaling van de accu is dit tot 2V gekozen. Voor R_3 is een $12.7k\Omega$ 0.1% weerstand gekozen dit betekend dat er ongeveer 78uA gaat lopen. De stroom zal in de praktijk wat afwijken vanwege de opamp input bias spanning. Met behulp van R_{32} kan de daadwerkelijke stroom vrij precies bepaald worden. Voor R_{32} is een $5.36k\Omega$ 0,1% gekozen zo zal er ongeveer 0,42V over gaan staan. De spanningsval over sensor met maximale waarde van $20k\Omega$ wordt 1,57V. Dit past dus binnen de 2V. Zie figuur 3



Figuur 3: Constante stroombron

5.6.2 Referentiespanning

De referentiespanning moet stabiel zijn. Wanneer deze verandert zal dit direct meetfouten met zich meebrengen. Om dit te bereiken is een Vref IC gebruikt (U2). Deze IC heeft een nauwkeurige uitgangsspanning van 2,048V. Met de weerstandverhouding van $R33$ en $R34$ wordt de spanning omlaag gebracht naar 1V. Zie Figuur 4



Figuur 4: Referentiespanning (V-REF)

5.6.3 Instrumentatieversterker

Voordat de spanning gemeten wordt komt het langs de instrumentatieversterker. Er is gekozen voor de instrumentatieversterker om de volgende redenen:

- Aanpasbare versterker
- Hoge input impedantie
- Goede common mode rejectie

- Differentieel

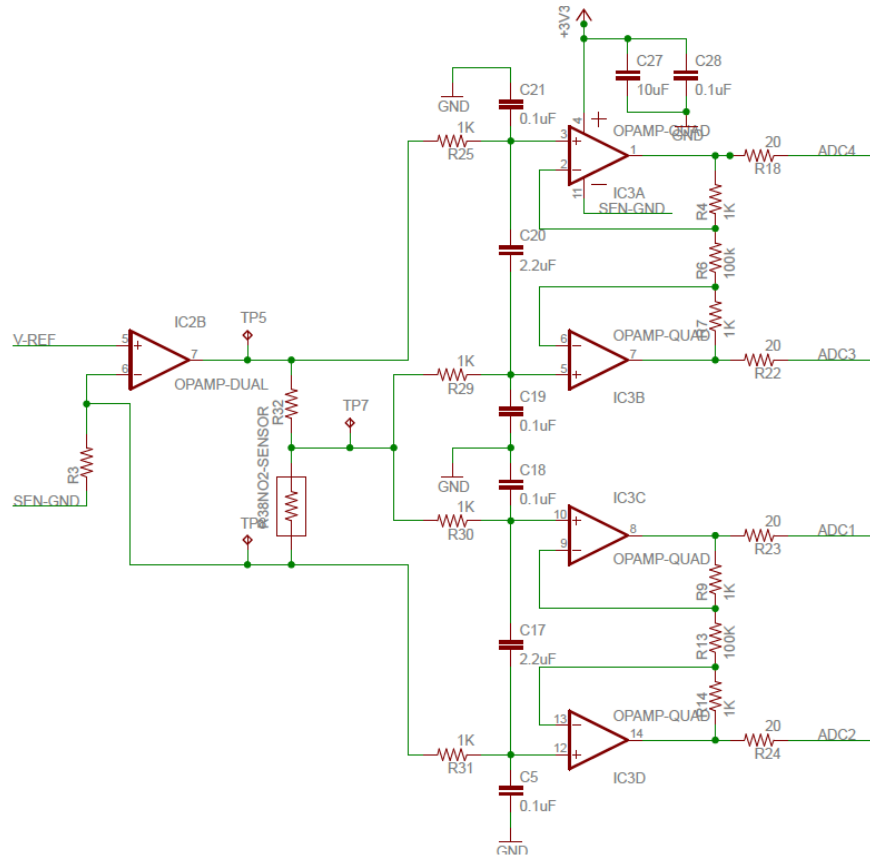
Om de kosten te drukken zijn de instrumentatieversterkers van losse OP-AMPS in plaats van een compleet IC. Dit heeft als nadeel dat de componenten niet gematched zijn. Het circuit is opgebouwd uit twee opamps als niet-inverterende versterkers. De input offset tussen (+) en (-) is erg klein. Omdat het systeem symmetrisch is kunnen deze offsetten van elkaar afgetrokken worden. Vanwege een kleine mismatch blijft er nog maar een kleine afwijking over. De versterking A_f kan met de formule (2) worden berekend

$$A_f = 1 + \frac{R_f}{R_G} \quad (2)$$

Waarbij:

R_f : gelijk aan R_4 of R_7

R_g : gelijk aan R_6



Figuur 5: Circuit met twee instrumentatieversterkers)

5.6.4 ADC

De spanning over de sensor wordt gemeten door de ADC's van de AVR microcontroller. Deze kunnen differentieel meten en bevat PGA's (programmable gain amplifiers).

5.6.5 Flexibiliteit

Zoals eerder vermeldt is het meetcircuit erg flexibel. De onderstaande onderdelen zijn gemakkelijk aan te passen:

- Referentiespanning door (R33 en R34)
- De constante stroom van stroombron door (R3)
- Het sensor meetbereik
- De gain van de instrumentatieversterker (R4, R6 en R7)
- Programmable gain van adc input.

5.6.6 Problemen en verbeterpunten

Allereerst zijn de ADC pinnen verkeerd aangesloten waardoor er geen gebruik gemaakt kon worden van de differentiële ADC meting. Nu wordt er twee keer single ended gemeten deze worden vervolgens van elkaar afgetrokken.

Verder heeft de meting een opmerkelijk grote afwijking van $\pm 6\%$ zelfs na calibratie. Dit is onderzocht het probleem lijkt volgens ons te zitten bij de kwaliteit van de ADC's. Het is aan te raden om hier nog even naar te kijken en mogelijk gebruik te maken van externe ADC ic's.

5.7 NO2-sensor-heater & Battery pack

De NO2 sensor die we hebben gebruikt zit een heater in die op 12 volt 200 mA verbruikt. Dit betekent dat er een externe voeding is toegevoegd om de heater van stoom te kunnen voorzien.

5.7.1 Probleem

Het toevoegen van een extra battery pack kost veel te veel ruimte waardoor hij niet echt draagbaar meer is, in het vervolg zal er een sensor komen die geen heater meer nodig heeft dus kan dit hele onderdeel wegelaten worden.

5.8 User(visual)

Om de gebruiker te informeren over de status van de snuffelneus en de luchtkwaliteit is er gekozen voor een RGB LED omdat hiermee alle kleuren gemaakt kunnen worden als deze met PWM aangestuurd worden.

5.9 Humidity & temperature sensor

Er is gekozen voor de DHT-11 module omdat hier niets meer aan kalibratie gedaan hoeft te worden en relatief goedkoop is. Ook is er een minimaal aantal aan componenten nodig. Een nadeel hiervan is dat hij een flinke afmeting heeft en niet standaard protocol om mee te communiceren.

5.10 Overig

5.10.1 MOSFET, enable van modules

De meeste modules hierboven zijn voorzien van een MOSFET om de module uit te kunnen schakelen. Deze keuze hebben we gemaakt om de snuffelneus energie zuinig te maken. De MOSFET wordt geschakeld door de microcontroller en zit tussen de min van de module of deelschakeling en de GND. Probleem: Bijna alle tekst op de PCB was veel te klein

5.11 Samenvatting aanpassingen en verbeterpunten

We hebben de DC-DC converter proberen in en uit te schakelen met een MOSFET tussen de min en de GND, dit werkte niet. Het signaal dat naar de GATE van de MOSFET gaat had direct naar de ENABLE van de DC-DC converter kunnen gaan. Dit hebben we dan ook aangepast op de printplaat.

Pull-down weerstanden moeten worden toegevoegd op de ingangen van de schmitt-trigger.

De bootloader hebben we niet aan de praat gekregen dus hebben we altijd moeten programmeren via de 6 pin header met een externe programmer in plaats van de bootloader knop in te drukken en dan te programmeren via de USB.

De ADC pinnen zijn verkeerd aangesloten waardoor er geen gebruik gemaakt kon worden van de differentiële ADC meting. Nu wordt er twee keer single ended gemeten deze worden vervolgens van elkaar afgetrokken.

Verder heeft de meting een opmerkelijk grote afwijking van +/- 6% zelfs na calibratie. Dit is onderzocht het probleem lijkt volgens ons te zitten bij de kwaliteit van de ADC's. Het is aan te raden om hier nog even naar te kijken en mogelijk gebruik te maken van externe ADC ic's.

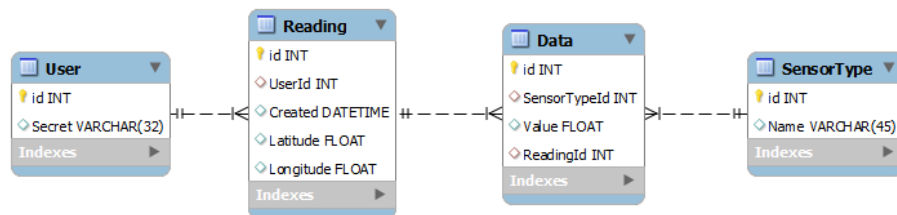
6 Handleiding server

6.1 Introductie

De server bestaat uit twee verschillende onderdelen om de data van alle snuffelneuzen gecontroleerd op te slaan. De database slaat alle gegevens op om later op te vragen en de API ontvangt aanvragen van de smartphone met meetinformatie die dan verwerkt worden.

6.2 Database

De database is gecreëerd aan de hand van de ERD te zien in Figuur 0.1. Het ontwerp bestaat uit vier tabellen die verbonden met elkaar zijn. De User tabel geeft elke aparte smartphone aan. Aan de hand van deze tabel wordt de binnengekomen data gekoppeld met een gebruiker. De Reading tabel wordt bijgevuld bij elke meting. Hierbij wordt de gebruiker, datum, tijd en locatie opgeslagen. De Data tabel bevat data van elke sensor apart en is verbonden met de Reading tabel en de SensorType tabel om aan te geven om welke sensor het gaat. In de SensorType tabel kunnen dus sensoren worden aangegeven. SQL Server 2012 R2 gehost vanuit Windows Azure is gebruikt als database voor de Snuffelneus.



Figuur 6:

6.3 API

Om de binnenkomende data te controleren wordt er gebruik gemaakt van een API. Via http aanvragen kan de Snuffelneus applicatie data versturen. Voor het toevoegen van data doet de smartphone een POST aanvraag naar 'http://linknaarsnuffelneuswebsite/api/Values' met als content-type: 'application/json'. Hierbij moet de data er als volgt uitzien (zie Listing 1)

```

1  {
2      "Secret": "ABCDEFGHJKLMOP",
3      "Latitude": 10.0,
4      "Longitude": 10.0,
5      "Measured": "2014-02-07T12:18:26.6399296Z",
6      "Values": [
7          {
8              "Sensor": "temp",
9              "Value": 44.44
10         },
11         {
12             "Sensor": "hum",
13             "Value": 55.55
14         },
15         {
16             "Sensor": "dust",
17             "Value": 66.66
18         }
19     ]
20 }

```

Listing 1: Json formaat

Deze data wordt vervolgens verwerkt en opgeslagen in de database. De API is in C# geschreven in combinatie met ASP.NET Web API. Tijdens de conceptperiode is de Web API gehost op Windows Azure, maar IIS (vanaf versie 8, mogelijk lager) wordt ook ondersteund.

Referenties

- [1] Oosterbaan a. “Kleine landschapselementen als invangers van fijn stof en ammoniak”. In: (2006).
- [2] S. Maas. *Environmental sensor for NOx detection*. URL: <http://www.holstcentre.com/en/Talent/ThesisOpportunities/Environmental%20sensor%20for%20NOx%20detection.aspx>.
- [3] Ontwikkelaars snuffelneus. “Onderzoek hardware”. In: (2014).
- [4] Ontwikkelaars snuffelneus. “Snuffelneus Design Description”. In: (2014).
- [5] Ontwikkelaars snuffelneus. “Vergelijkingen Gassen en sensoren”. In: (2014).