

**Gemeentelijk Instituut Technisch Onderwijs Kalmthout**

**Technisch secundair onderwijs**

**Industriële Informatie en Communicatie Technologieën**

**LUCHTKWALITEIT**

**afbeelding**

**Geïntegreerde proef van**

Ruben Socquet & Arthur d’Hooge

**Promotor en mentor:**

m. D. Pauwels, m. G. Wagemans en mevr. T. Haentjes

**Kalmthout, 2018 - 2019**



**Gemeentelijk Instituut Technisch Onderwijs Kalmthout**

**Technisch secundair onderwijs**

**Industriële Informatie en Communicatie Technologieën**

**LUCHTKWALITEIT**

**afbeelding**

**Geïntegreerde proef van**

Ruben Socquet & Arthur d’Hooge

**Promotor en mentor:**

m. D. Pauwels, m. G. Wagemans en mevr. T. Haentjes

**Kalmthout, 2018 - 2019**

# Voorwoord

# Inhoudsopgave

Inhoud

[Voorwoord 4](#_Toc534806893)

[Inhoudsopgave 5](#_Toc534806894)

[Inleiding 7](#_Toc534806895)

[1.1 Definitiestudie (Project) 8](#_Toc534806896)

[1.1.1 Probleemstelling 8](#_Toc534806897)

[1.1.2 Specificaties 8](#_Toc534806898)

[1.1.3 Functionaliteiten 9](#_Toc534806899)

[1.1.4 Hardware 9](#_Toc534806900)

[1.1.5 Software 9](#_Toc534806901)

[1.1.6 Gebruikers 9](#_Toc534806902)

[1.1.7 Veiligheid 10](#_Toc534806903)

[1.2 Detailontwerp (project) 11](#_Toc534806904)

[1.2.1 Principewerking en principeschema (‘s) 11](#_Toc534806905)

[1.2.2 Hardware 12](#_Toc534806906)

[1.2.3 Dataverwerking 14](#_Toc534806907)

[1.2.4 Processturing 14](#_Toc534806908)

[1.2.5 Communicatie 15](#_Toc534806909)

[1.2.6 Garanties en veiligheid 15](#_Toc534806910)

[2. Schema’s en tekeningen 17](#_Toc534806911)

[2.1 Client zijde 17](#_Toc534806912)

[2.2 Netwerkdiagram 17](#_Toc534806913)

[2.3 gps-opstelling 18](#_Toc534806914)

[2.4 DHT11 opstelling 18](#_Toc534806915)

[2.5 CCS811 opstelling 19](#_Toc534806916)

[2.6 Schema BH1750 19](#_Toc534806917)

[3. Elektronica 20](#_Toc534806918)

[3.1 DHT11 20](#_Toc534806919)

[3.2 BH1750 21](#_Toc534806920)

[3.3 NEO-6M 22](#_Toc534806921)

[3.4 SIM900 23](#_Toc534806922)

[4. Logboek 24](#_Toc534806923)

[5. Reflectieverslag 25](#_Toc534806924)

[6. Bibliografie 26](#_Toc534806925)

# Inleiding

Dit GIP heeft als doel om een methode te ontwikkelen waarmee je eenvoudig de kwaliteit van de lucht in jouw omgeving te meten. Het doel is om het de gebruiker zo makkelijk en efficiënt mogelijk te maken. We gaan dit doen via een soort van cloud service. Gebruikers krijgen een meettoestel dat draagbaar, compact en voorzien is van een batterij zodat ze het mee kunnen nemen. Hiermee kunnen ze de lucht in hun omgeving in kaart brengen. Het meettoestel word voorzien van een 4g verbinding zodat het gemeten waardes naar de server kan sturen. De server word ook voorzien van een google maps kaart, daarmee kunnen gebruikers zien in welke plaatsen ze hebben gemeten en de daarbij horende waardes bekijken. Het doel is om plaatsen in kaart te brengen met nog veel vervuiling. Het toestel meet wel alleen CO2 waardes en hoeveelheden organische verbindingen. Het kan dus niet CO of fijnstof meten. Wel meet het de luchtvochtigheid en de lichtsterkte. 1. Het eigenlijke werk: Luchtkwaliteit

## 1.1 Definitiestudie (Project)

### 1.1.1 Probleemstelling

We zijn begonnen van het idee om een vliegtuig te maken en vanop het vliegtuig de luchtkwaliteit te meten. Om dit meer te richten op commercieel gebruik gaan we dit zonder een vliegtuig maken. Maar de sensoren in een behuizing integreren met een gps-tracker dat op verschillende voertuigen geplaatst kan worden. Vervolgens wordt de data via het mobiele netwerk verstuurd naar de server, deze server gaat de data verwerken en live op een website plaatsen. Die website zal openbaar toegankelijk zijn, wanneer je inlogt kan je volgen waar de luchtkwaliteit-meter zich bevindt en de gemeten data op die plaats bekijken. We hebben voor dit project gekozen omdat luchtvervuiling de laatste jaren een groter probleem wordt voor onze gezondheid. Met de data dat we verzamelen gaan we kunnen aantonen of de luchtkwaliteit daadwerkelijk achteruitgaat en of er iets aan veranderd moet worden. Doordat dit systeem eenvoudig in gebruik zal zijn kan het door iedereen gebruikt worden. Hierdoor zal er veel data verzameld kunnen worden om aan te tonen hoe de luchtomstandigheden evolueren. Dit systeem heeft als voordeel dat het overal kan gebruikt worden. Wanneer er geen data verstuurd kan worden via het mobiele netwerk zal er nog steeds data gemeten en lokaal bewaard worden. Deze data kan dan worden verzonden wanneer er terug een signaal is.

### 1.1.2 Specificaties

Het systeem zal uit 2 delen bestaan. Deel 1 is de Arduino Uno, de sensoren en de SIM900 shield deze afmetingen moeten compact blijven en mogen niet veel wegen. Het is een mobiele opstelling dat gemakkelijk ergens mee naartoe genomen kan worden en op voertuigen geplaatst kan worden. Het kan ook in een lokaal geplaatst kan worden om van op één plaats de metingen uit te voeren. Het systeem zal gebruikt worden met Arduino Uno om de data van de sensoren in te lezen, op te slaan en door te sturen. Voor de financiering hebben we afgesproken om dit onder ons 2 te verdelen.

### 1.1.3 Functionaliteiten

Een groot deel van het systeem zal het doorsturen van de data van de sensoren zijn via het mobiele netwerk door middel van de SIM900 met de Arduino. Wanneer deze data gaat doorgestuurd worden naar de website waar er dan verschillende gegevens te vinden zijn. Als output hebben we dan de data van de CO²-, temperatuur-, luchtvochtigheid-, lichtsterkte sensor en de locatie waar de data verzameld is. Om de hardware en sensors te voeden zal een batterij geïntegreerd worden in de behuizing.

### 1.1.4 Hardware

De hardware bestaat uit 2 grote delen, de server en het meettoestel. Voor de server gebruiken we een gewone pc waarop we IIS draaien (manier om webpagina’s te hosten) en een database met alle meetwaarden. Het meettoestel functioneert rond de Arduino waaraan alle sensors en de 4g module aangesloten zijn. De Arduino voert communicatie met de sensors via de I²C OneWire protocollen. De SIM900 is verbonden met de Arduino via een seriële verbinding.

### 1.1.5 Software

Er zal op de website door middel van registratie de luchtkwaliteitsmeter ingesteld worden en indien er al metingen gedaan zijn zullen er grafieken met de gemeten data te vinden zijn gebonden aan de locatie. Een server zal deze website hosten, samen met een console applicatie dat op de achtergrond zal draaien. De gebruikers zullen niets van deze applicatie te zien krijgen. Op de server zal ook een database aanwezig zijn. De functie van de console app is om de data van het meettoestel te ontvangen en in de database te zetten. De website kan dezelfde database aanspreken om de gegevens te tonen in de Web UI.

### 1.1.6 Gebruikers

De verschillende gebruikers kunnen we onderverdelen in de gebruikers die geen luchtkwaliteit meter hebben. Deze gebruikers kunnen niet inloggen op de website maar gaan wel de algemene data grafieken kunnen terugvinden van een gemeente op de website. Vervolgens hebben we de gebruikers dat wel een systeem hebben gekocht, zij kunnen een account aanmaken en dat account linken aan hun systeem. Zij kunnen de specifiek door hun gemeten waarden gelinkt aan de plaats bekijken. En dit vergelijken met andere plaatsen. Ook zal er aangegeven worden of dit gezonde of niet gezonde levensomstandigheden zijn. Ten slotte hebben we nog de administrators, dit zijn Arthur d’Hooge en Ruben Socquet, zij kunnen alle specifieke verzamelde data bekijken van elke gebruiker. De maximale hoeveelheid gebruikers hangt af van de server waarop de website wordt gehost.

### 1.1.7 Veiligheid

Er zal zeker rekening gehouden moeten worden naar privacy en bescherming van gegevens want wanneer de gebruikers gaan meten wordt live hun locatie doorgestuurd en dat is toch zeer gevoelige data. Daarom zal de data dat verstuurd wordt zeker versleuteld moeten worden. De website moet ook goed beveiligd worden zodat er geen vertrouwde gegevens van de gebruiker verloren gaan.

## 1.2 Detailontwerp (project)

### 1.2.1 Principewerking en principeschema (‘s)

De werking van ons GIP gaat als volgt: de sensoren aan de Arduino Uno voeren constant metingen uit. Die data slaan ze dan op en sturen ze via het mobiele netwerk door naar de server die de data zal verwerken en opslaan.

Grafische ontwerpen:

3D-model van de behuizing.

### 1.2.2 Hardware

De hardware die we gaan gebruiken:

|  |  |  |  |
| --- | --- | --- | --- |
| Lichtsensor | Lichtsterkte meten. | I2C-protocol, 1 - 65535 lux meetbereik, 5V. |  |
| CO2 sensor | CO2 gehalte in de lucht meten. | I2C-protocol, werkt op 3.3V. |  |
| Gsm-module | Communicatie met server. | Communicatie via seriële poort, heeft 12V voeding nodig, werkt op alle gsm-netwerken. |  |
| Vochtigheid en temperatuur sensor | Luchtvochtigheid en temperatuur meten. | Werkt op voeding van Arduino (5V), communiceert met OneWire protocol. |  |
| Arduino Uno | Microcontroller om meettoestel te besturen. |  |  |
| Batterij houder | Stroomvoorziening Meettoestel. | 6 AA Batterijen.  9V. |  |

Op de Arduino zullen we het programma laten draaien zodat we de sensoren kunnen besturen en de data van de sensoren via de SIM900 naar de server kunnen sturen. Dat programma zal gemaakt worden in C++ een taal die het makkelijk maakt om te interfacen met de Arduino zijn I/O pinnen. In het programma zal verbinding gemaakt worden met de SIM 900 shield, die shield gaat een SIM kaart hebben zodat er een verbinding met het mobiele netwerk gemaakt kan worden en de data naar de server gestuurd wordt. Die server zal een ASP.NET website draaien en een database beheren waarin alle gemeten data word opgeslagen.

### 1.2.3 Dataverwerking

De gemeten data zal voor het grootste deel verwekt worden in de server, dit is logisch want de server heeft de grootste rekenkracht. De Arduino moet enkel de data uitlezen en versturen. Met de data gaat de server dan een kaart plotten met locaties van het meettoestel, de CO2 gehaltes, luchtvochtigheid, lichtsterkte, temperatuur en het tijdstip van de meting bij elke locatie zetten. Alle data word ook in een database bewaard zodat als er later het zelfde traject word afgelegd er vergeleken kan worden. Er moet wel rekening gehouden worden met de privacy van de gegevens en in bijzondere maten de locatie.

### 1.2.4 Processturing

De besturing is heel eenvoudig, voor de meting kunnen parameters aangepast worden via de website die op de server draait. De parameters zijn dan welke soorten metingen (CO2, temperatuur, lichtsterkte…), de meetsnelheid en hoelang er gemeten moet worden. Deze worden dan doorgegeven naar het meettoestel. Op de website zal ook de meting gestart en gestopt kunnen worden. Voor de rest zal het systeem alles automatisch afhandelen. Op de website word er dan live-info getoond over het meettoestel, zoals de locatie en de huidige toestand van de sensoren.

### 1.2.5 Communicatie

Omdat wij veel verschillende toestellen (Arduino, Server, Database) gebruiken is communicatie een groot deel van ons GIP. Om te beginnen zullen de digitale sensoren die allemaal over I2C beschikken op een I2C bus worden aangesloten waarvan de master de Arduino is. De vochtigheid en temperatuur sensor werkt met het OneWire protocol en zal ook op de Arduino worden aangesloten. Door de SIM900 te verbinden met de microcontroller gaan we gebruik kunnen maken van het mobiele datanetwerk. Dit betekent dat de communicatie over het WAN gaat en we dus onze data moeten encrypteren. Om toegang te krijgen tot het mobiele netwerk moeten we een simkaart in de module steken. Hiervoor gaan we een prepaid simkaart gebruiken. Als het meettoestel data wilt verzenden zal een verbinding worden gemaakt met de server volgens het TCP protocol. Dit protocol heeft doorgedreven foutencontrole ingebouwd. Zo zijn we zeker dat er onderweg naar de server geen data verloren gaat of corrupt word.

### 1.2.6 Garanties en veiligheid

Mechanische veiligheid:

Alle hardware onderdelen worden in een afgesloten bakje gemonteerd dat het grootste vuil buitenhoudt. Dit bakje zal worden geprint met een 3D printer. Het zal gemaakt worden uit ABS. Dit plastiek kan temperaturen aan tot 85°C. Het is licht en hard, wat het ideaal maakt voor dit project.

Elektrische veiligheid:

De Arduino en gsm module hebben een ingebouwde voltage regulator. Dit betekend dat de 9V van de batterij naar 5V word gebracht. 9V en 5V zijn ‘veilige’ spanningen. Er is geen gevaar voor elektrocutie bij het aanraken van deze spanningen. Bij een kortsluiting kunnen er andere gevaren optreden zoals brand, batterijen die oververhitten, giftige gassen die vrijkomen,.. Gelukkig zijn zowel de Arduino als de GSM module beveiligt met een interne zekering.

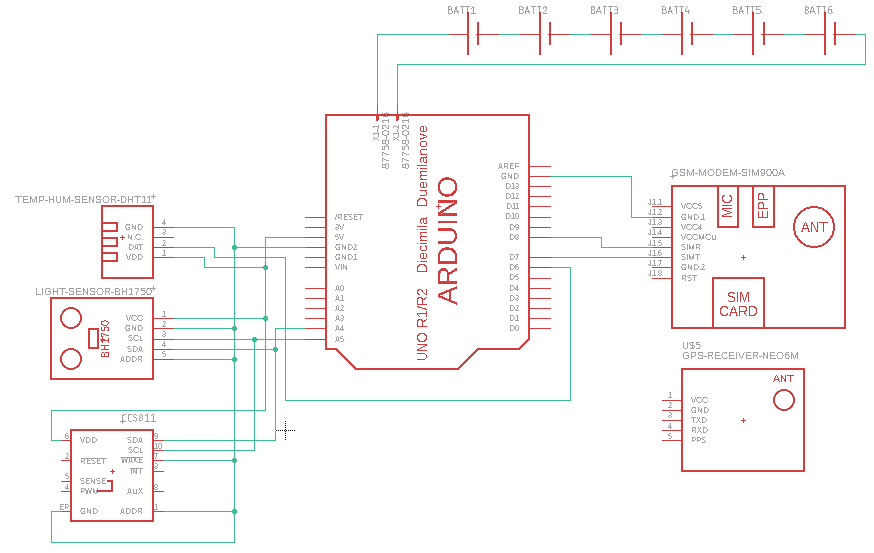
 *500mA smd zekering*

Privacy:

Locaties van gebruikers is een zeer gevoelig punt, we zullen in ons GIP moeten kijken of we aan de eisen van de wetgeving kunnen voldoen, hoelang we die data mogen houden, en andere vraagstukken hierrond moeten beantwoorden.

# 2. Schema’s en tekeningen

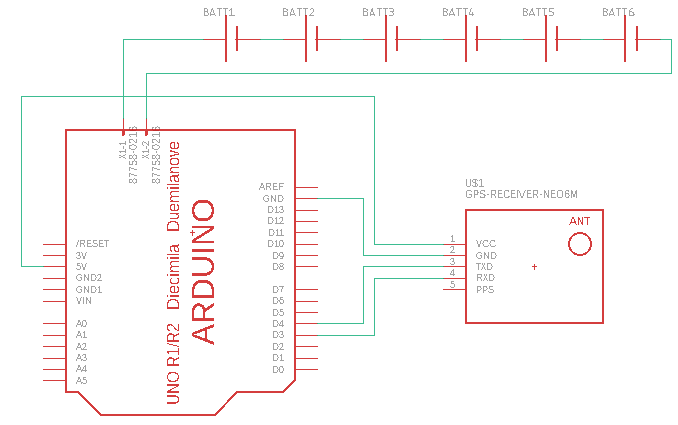
## 2.1 Client zijde



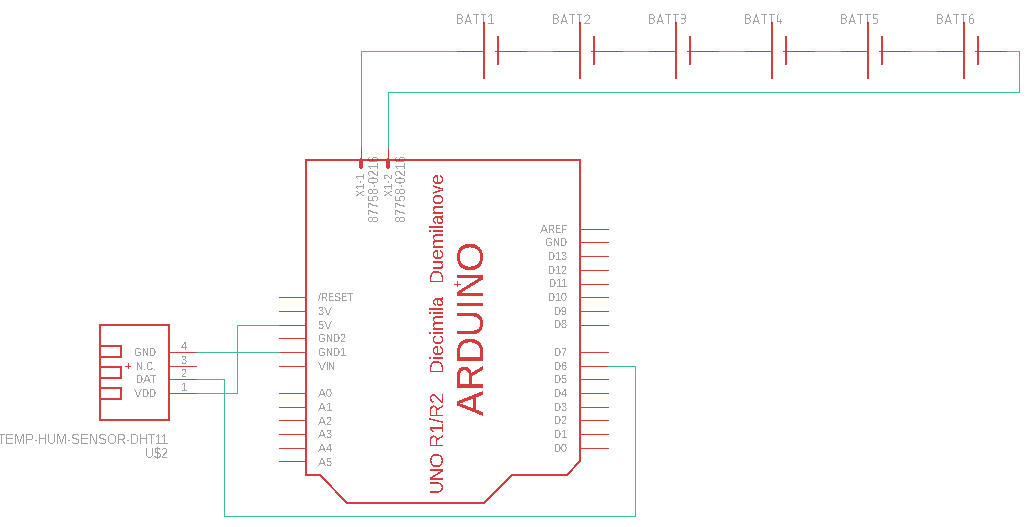
## 2.2 Netwerkdiagram



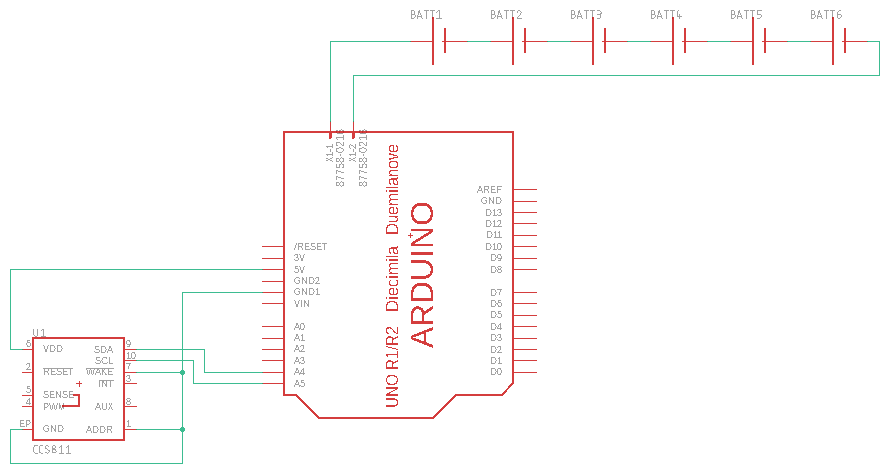
## 2.3 gps-opstelling



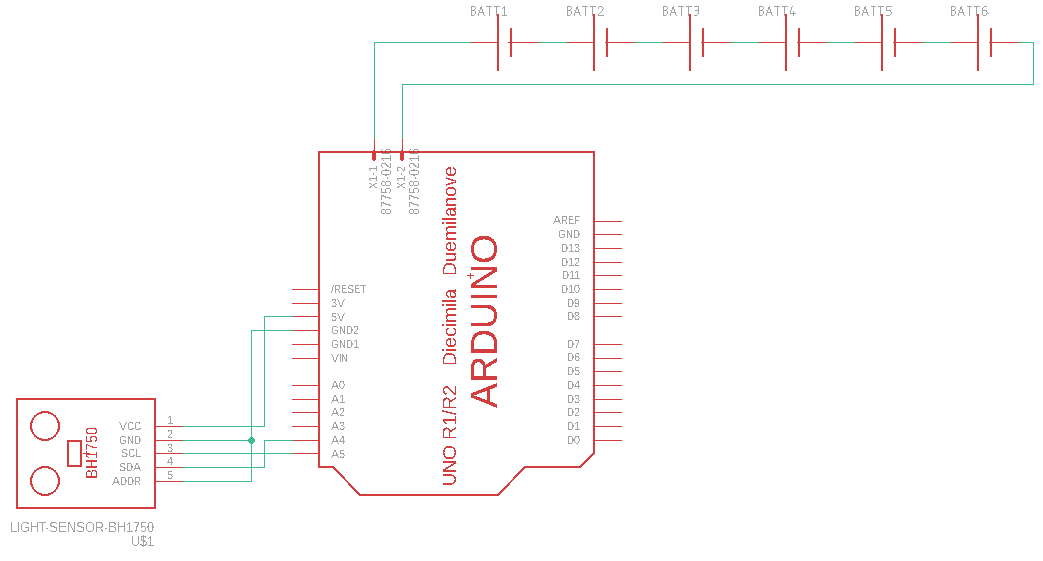
## 2.4 DHT11 opstelling



## 2.5 CCS811 opstelling



## 2.6 Schema BH1750

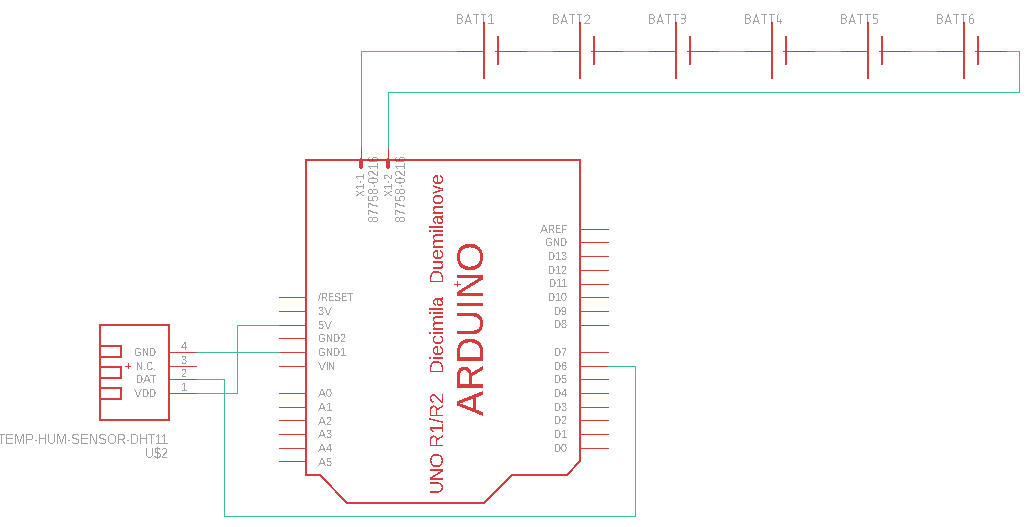


# 3. Elektronica

In dit hoofdstuk van het GIP gaan we kijken naar de hardwarecomponenten dat we gekozen hebben en hoe we ze hebben laten werken. We hebben in totaal 4 sensoren gebruikt waarmee we data gaan verzamelen. Deze waarden zijn de temperatuur, vochtigheid, lichtsterkte, locatie en CO² waarden.

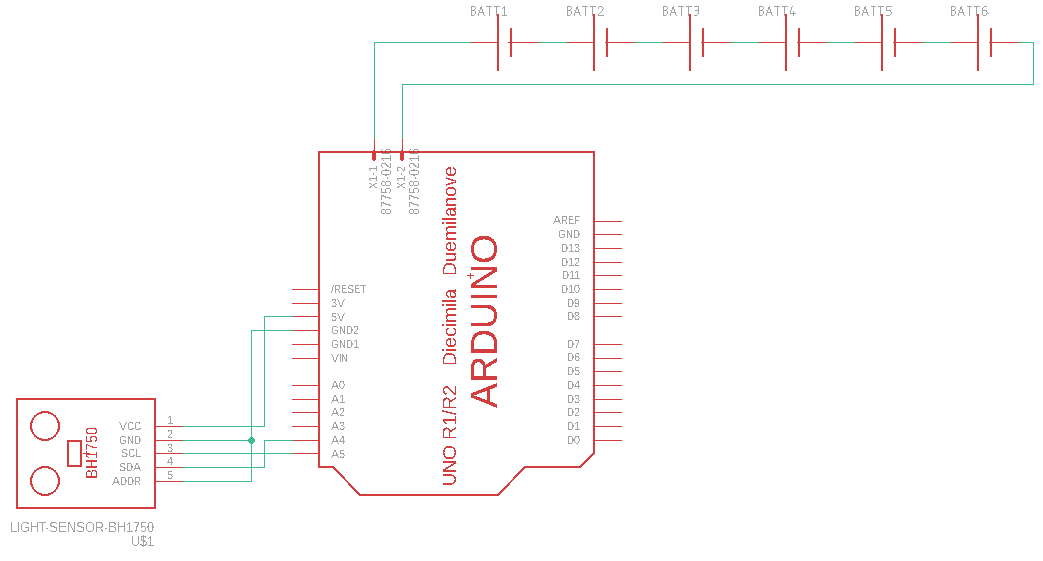
## 3.1 DHT11

Om de temperatuur en de vochtigheid te meten gebruiken we namelijk één sensor, de DHT11. Deze sensor heeft drie aansluitingen, twee voor 5V voeding van de Arduino en één voor de data door te sturen. Met deze sensor kunnen we temperaturen meten van 0°C tot 50°C met een nauwkeurigheid van +/- 2°C en een vochtigheid van 20% tot 90% met nauwkeurigheid van +/- 5%. De temperatuur wordt gemeten door een NTC-temperatuur sensor en de vochtigheid wordt gemeten door een vochtigheid afhankelijke weerstand.



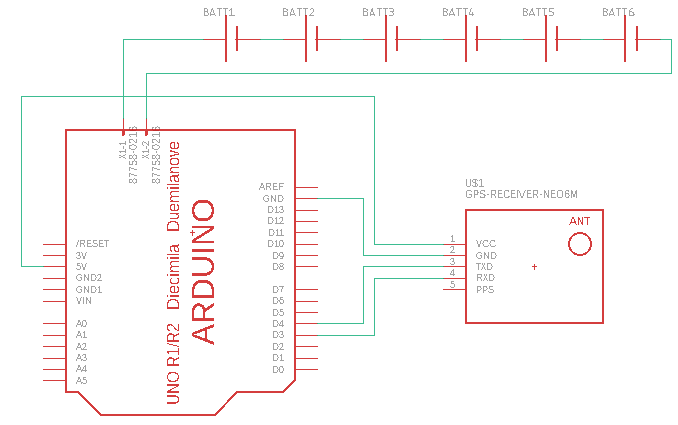
## 3.2 BH1750

Voor lichtsterkte te meten maken we gebruik van een BH1750 sensor dat lichtsterktes kan meten van 1lux tot 65535 lux. Door vijf aansluitingen, GND en VCC worden gebruikt voor de voeding van de sensor, namelijk 3,3V van de Arduino. De SCL pin van de sensor wordt verbonden met poort A5 van de Arduino, deze aansluiting gaat ervoor zorgen dat er een kloksignaal wordt gestuurd en de data volgens het kloksignaal verstuurd kan worden. Vervolgens hebben we de SDA, deze pin sluiten we aan op poort A4 van de Arduino en wordt gebruikt om data van de sensor naar de Arduino te sturen. Ten slotte is er nog de address aansluiting, deze wordt verbonden met de GND van de Arduino. Om de data naar de Arduino te sturen wordt een I²C bus gebruikt.



## 3.3 NEO-6M

Dit component in onze schakeling gaat er voor zorgen om gps coördinaten van zijn locatie te achterhalen en te sturen naar de Arduino. De gps heeft vier aansluitingen en een antenne. Het wordt gevoed door 3,3V van Arduino aan de VCC en GND pinnen. Dan hebben we nog de TX en RX pinnen. De TX pin zal de data versturen naar de Arduino en aan de RX pin zal de data ontvangen worden. Beide aansluitingen gaan serieel werken dus de data zal serieel ontvangen en verstuurd worden.



## 3.4 SIM900

Een ander deel van ons netwerk is de communicatie tussen de Arduino en de server. Hiervoor wordt een connectie via het mobiele netwerk gemaakt om het apparaat zo mobiel en gebruiksvriendelijk mogelijk te maken. Om dit te realiseren gebruiken we een SIM900 met een SIM kaart. Om de shield te voeden gebruiken we zes in serie geplaatste AA batterijen wat resulteert in een spanning van 9V wat genoeg is om de Arduino en de SIM900 shield te voeden. Om de data van de arduino naar de SIM900 te sturen gebruiken we drie pinnen, namelijk de GND pin aan de GND van de Arduino en een RX en TX pin waarover de daadwerkelijke data serieel in beide richtingen gestuurd kan worden. Om deze dataoverdracht te realiseren kunnen we via AT commando’s de shield aansturen en zo een verbinding met de serveropstellen en berichten vesturen. Op het SIM900 board kunnen we een antenne terugvinden. Deze antenne gaat ervoor zorgen dat er een verbinding met de GSM masten van de provider gemaakt kan worden. Wanneer we een verbinding met zo een GSM mast willen maken gaan we in ons programma de acces point name van de provider ingeven zodat er via de provider een toegang tot het WAN in en er een publiek IP-adres aan de shield wordt geven. Wanneer er toegang is tot het WAN kunnen we doormiddel van een vast IP-adres van de server dat in de code van de Arduino staat een verbinding met de server maken en de data van de sensoren doorsturen.

# 4. Logboek

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATUM: | WAT: | NAAM: | TIJD: | PLAATS: |
| 9/10/2018 | GIT repository | Samen | 1 uur | School |
| 9/10/2018 | Vocht sensor | Samen | 1 uur | School |
| 19/10/2018 | Plan & sensoren | Samen | 2 uur | School |
| 23/10/2018 | Sensoren | Samen | 2 uur | School |
| 24/10/2018 | Documentatie, to do list | Samen | 2 uur | School |
| 26/10/2018 | Licht sensor en CO² sensor | Samen | 2 uur | School |
| 3/11/2018 | Definitiestudie | Ruben | 2 uur | Thuis |
| 4/11/2018 | Definitiestudie | Ruben | 1 uur | Thuis |
| 5/11/2018 | Detailontwerp | Arthur | 3 uur | Thuis |
| 7/11/2018 | To do list, bestellijst, documentatie aanpassen | Samen | 2 uur | School |
| 13/11/2018 | DHT11 sensor en CO² sensor | Samen | 3 uur | School |
| 19/11/2018 | Bestellijst | Arthur | 30 min | Thuis |
| 17/12/2018 | Bundel | Arthur | 1uur | School |
| 26/12/2018 | Visual Studio projecten opgestart | Arthur | 3uur | Thuis |
| 27/12/2018 | Bundel bijwerken | Arthur | 2 uur | Thuis |
| 2/01/2019 | Communicatie Arduino en server | Ruben | 3 uur | Thuis |
| 3/01/2019 | Schema’s, libraries, bundel | Ruben | 2 uur | Thuis |
| 6/01/2019 | Bundel bijwerken | Ruben | 2 uur 30min | Thuis |
| 7/01/2019 | Bundel bijwerken | Arthur | 1 uur | Thuis |
| 7/01/2019 | Communicatie met server | Ruben | 3 uur | Thuis |
| 8/01/2019 | Bundel, website, sensoren, schema’s | Samen | 6 uur | School |
| 8/01/2019 | Bundel, Sensoren | Ruben | 3 uur | Thuis |
| 9/01/2019 | Bundel | Arthur | 15min | Thuis |
|  |  |  |  |  |
|  |  |  |  |  |

# 5. Reflectieverslag

# 6. Bibliografie