



Gemeentelijk Instituut Technisch Onderwijs Kalmthout

Technisch secundair onderwijs

Industriële Informatie en Communicatie Technologieën

# LUCHTKWALITEIT

**Geïntegreerde proef van**

Ruben Socquet & Arthur d'Hooge

**Promotor en mentor:**

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

**Kalmthout, 2018 - 2019**





Gemeentelijk Instituut Technisch Onderwijs Kalmthout

Technisch secundair onderwijs

Industriële Informatie en Communicatie Technologieën

# LUCHTKWALITEIT

**Geïntegreerde proef van**

Ruben Socquet & Arthur d'Hooge

**Promotor en mentor:**

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

**Kalmthout, 2018 - 2019**

# Voorwoord

Om de luchtkwaliteitsmeter te realiseren hebben we meerdere obstakels gehad. Om die problemen op te lossen konden we steeds terecht bij onze mentor en promotor. Daarom zou ik graag m. D. Pauwels, m. G. Wagemans. We willen graag mevr. T. Haentjens bedanken voor het nalezen en uitgebreid verbeteren van de bundel.

# Inhoudsopgave

## Inhoudsopgave

Voorwoord.....	1
Inhoudsopgave.....	2
Inleiding.....	4
1. Het eigenlijke werk.....	5
1.1 Definitiestudie (Project) .....	5
1.1.1 Probleemstelling.....	5
1.1.2 Specificaties .....	6
1.1.3 Functionaliteiten .....	6
1.1.4 Hardware .....	6
1.1.5 Software .....	7
1.1.6 Gebruikers.....	7
1.1.7 Veiligheid .....	7
1.2 Detailontwerp .....	8
1.2.1 Principewerking en principeschema ('s).....	8
1.2.2 Hardware .....	9
1.2.3 Dataverwerking .....	12
1.2.4 Processturing .....	12
1.2.5 Communicatie .....	13
1.2.6 Garanties en veiligheid.....	14
1.2.7 Software .....	15
2. Schema's en tekeningen .....	17
2.1 Lay-out website .....	17
.....	18
2.2 Meettoestel .....	20
.....	20
2.3 PCB.....	21
2.3 Netwerkdigram .....	22
3. Elektronica .....	23
3.1 DHT11 .....	23

3.2 BH1750 .....	26
3.3 NEO-6M.....	28
3.4 SIM900 .....	31
3.5 CCS 811 .....	34
3.6 MicroSD card adapter .....	36
3.7 Arduino Uno.....	38
3.8 Elektromagnetische compatibiliteit.....	39
4. Elektriciteit .....	41
4.1 Vitale 8.....	41
4.2 Elektrische veiligheid.....	42
5. Logboek.....	44
6. Reflectieverslag.....	49
7. Bibliografie.....	50
8. Bronnen.....	51

## Inleiding

Dit GIP heeft als doel om een methode te ontwikkelen waarmee je eenvoudig de kwaliteit van de lucht in de omgeving kan meten. Het doel is om het de gebruiker zo makkelijk en efficiënt mogelijk te maken. We gaan dit doen aan de hand van een 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 wordt voorzien van een mobiele verbinding tot het internet zodat het de gemeten waarden naar de server kan sturen. De server wordt ook voorzien van een map, daarmee kunnen gebruikers zien in welke plaatsen er metingen zijn uitgevoerd en de daarbij horende waarden bekijken. Het doel is om locaties in kaart te brengen waar er veel luchtvervuiling is zodat er aangetoond kan worden dat er maatregelen genomen moeten worden. Het toestel meet CO<sup>2</sup> waardes en hoeveelheden organische verbindingen. Daarnaast wordt er ook temperatuur, luchtvochtigheid en lichtsterkte gemeten.

# 1. Het eigenlijke werk

## 1.1 Definitiestudie (Project)

### 1.1.1 Probleemstelling

We zijn vertrokken 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 vliegtuig doen. We integreren 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 zal de data verwerken en live op een website plaatsen. Die website zal openbaar toegankelijk zijn, wanneer je inlogt kan je volgen waar de luchtkwaliteitsmeter 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 die we verzamelen kunnen we aantonen of de luchtkwaliteit daadwerkelijk achteruitgaat en er iets aan veranderd moet worden. Doordat dit systeem eenvoudig in gebruik is, 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 geüpload worden wanneer de gebruiker inlogt op de website.



### 1.1.2 Specificaties

Het systeem zal uit twee delen bestaan. Deel één bestaat uit de Arduino Uno, de sensoren en de SIM900 shield. Deze afmetingen moeten compact blijven en ze mogen niet veel wegen. Het is een mobiele opstelling die gemakkelijk ergens mee naartoe genomen kan worden en op voertuigen geplaatst kan worden. Het kan ook lokaal geplaatst 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 twee te verdelen.

### 1.1.3 Functionaliteiten

Een groot deel van het project is het doorsturen van de data gemeten door de sensoren via het mobiele netwerk door middel van de SIM900 naar de Arduino. Wanneer deze data doorgestuurd wordt naar de website hebben we als output de CO<sup>2</sup>-, temperatuur-, luchtvochtigheid-, lichtsterkte sensor, locatie waar de data verzameld is en de tijd van de meting. Omdat er op de hiervoor beschreven manier een verbinding gemaakt moet worden met de zendmasten van een provider, is er ook de mogelijkheid om de data lokaal op een SD-kaart te bewaren en later te uploaden naar de website. Om de hardware en sensoren te voeden zal een batterij geïntegreerd worden in de behuizing.

### 1.1.4 Hardware

De hardware bestaat uit twee onderdelen, 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 sensoren aangesloten zijn, de 4G module, de gps en de SD-kaarthouder.

### 1.1.5 Software

De luchtkwaliteitsmeter wordt ingesteld door deze te registreren op de website. Indien er al metingen gedaan zijn, zullen er grafieken met de gemeten data te vinden zijn gebonden aan de locatie.

### 1.1.6 Gebruikers

Zowel gebruikers als niet-gebruikers van de luchtkwaliteitsmeter kunnen de website bezoeken. De niet-gebruikers kunnen niet inloggen op de website maar kunnen wel de algemene datagrafieken terugvinden van een gemeente op de website. De gebruikers die wel een systeem hebben gekocht, kunnen een account aanmaken en dat account linken aan hun systeem. Zij kunnen de specifiek door hen gemeten waarden, gelinkt aan de plaats, bekijken en dit vergelijken met andere plaatsen. Ook zal er aangegeven worden of dit gezonde levensomstandigheden zijn of niet. Ten slotte hebben we nog de administrators, zij kunnen alle specifieke verzamelde data bekijken van elke gebruiker.

De maximale hoeveelheid gebruikers is afhankelijk van de server waarop de website wordt gehost.

### 1.1.7 Veiligheid

Er zal zeker rekening gehouden moeten worden met 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 die 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


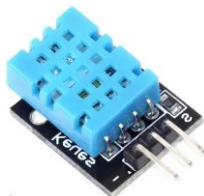

### 1.2.1 Principewerking en principeschema ('s)



De werking van de luchtmeter begint bij de Arduino die de metingen van de verschillende sensoren opvraagt. Waarna vervolgens de data van de sensoren tijdelijk in de Arduino zal worden opgeslagen tot er een verbinding met de server wordt gemaakt met de SIM900 module. Die module zal door een vaste code in de Arduino eerst een IP-adres opvragen in het WAN (Wide Area Network) en vervolgens verbinding maken met de APN (Access Point Name) van de provider van de SIM kaart. Wanneer die verbinding is gemaakt, zal er een verbinding gemaakt worden met de server dat een vast IP-adres en vaste poort zal hebben zodat de Arduino weet met welke server verbinding gemaakt moet worden. Al de commando's die naar de SIM900 shield gestuurd worden, zijn AT commando's. Wanneer het meetapparaat verbonden is met de server kan de data van de sensoren verstuurd worden. In de server wordt de data opgeslagen in een Acces database zodat de data overzichtelijk en makkelijk toegankelijk opgeslagen is en vervolgens verwerkt en bezichtigd kan worden via de website. Op de website zal er een optie zijn om te registreren en in te loggen. Wanneer de bezoeker van de website een meter heeft, kan hij deze registreren via de site om de door hem gemeten metingen en zijn afgelegde route te bekijken.

## 1.2.2 Hardware

De hardware die we gaan gebruiken:

Lichtsensoren	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.		

Batterijhouder	Stroomvoorziening meettoestel.	6 AA Batterijen.  9V.	
microSD card adapter Catalex	Lokale opslag op een SD-kaart	5V, SPI protocol	

Het programma draait op Arduino 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 interfaceren met de I/O pinnen van de Arduino. In het programma zal verbinding gemaakt worden met de SIM900 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 op een ASP.NET website draaien en een database beheren waarin alle gemeten data wordt opgeslagen.

### 1.2.3 Dataverwerking

De gemeten data zal voor het grootste deel verwerkt 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 CO<sub>2</sub>-gehalten, luchtvochtigheid, lichtsterkte, temperatuur en het tijdstip van de meting bij elke locatie zetten. Alle data wordt ook in een database bewaard zodat, wanneer er later hetzelfde traject wordt afgelegd, er vergeleken kan worden. Er moet wel rekening gehouden worden met de privacy van de gegevens en in het bijzonder 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 (CO<sub>2</sub>, 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. Verder zal het systeem alles automatisch afhandelen. Op de website wordt live-info getoond over het meettoestel, zoals de locatie en de huidige toestand van de sensoren wanneer de data overgedragen wordt via de SIM900. Bij gebruik van de SD-kaart opslagmethode wordt de data pas op de website zichtbaar nadat het wordt geüpload.

### 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 temperatuursensor 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 onze data dus 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 wordt.

Omdat de SIM900 mobiele data gebruikt voor het versturen van de data en ook een verbinding met een gsm-mast nodig heeft is er ook de mogelijkheid om data lokaal op te slaan. De gemeten data gaat dan door een SD-adapter naar een SD-kaart gestuurd worden en in een tekstdocument geplaatst worden. Na de metingen zal de gebruiker de SD-kaart moeten verbinden met een computer om tenslotte het tekstbestand te uploaden naar de website.



## 1.2.6 Garanties en veiligheid

### 1.2.6.1 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 plastic kan temperaturen aan tot 85°C. Het is licht en hard, wat het ideaal maakt voor dit project.

### 1.2.6.4 Privacy

De locatie 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.

## 1.2.7 Software

### 1.2.7.1 Serverstructuur

De software voor de server wordt geschreven in Visual Studio 2017. We gebruiken C# als programmeertaal. In de Solution hebben we twee projecten: 'Data\_Listener' en 'Website'. In het project 'Data\_Listener' schrijven we de code om gemeten waarden van sensoren die van het meettoestel komen weg te schrijven naar de database (zie 1.2.7.3 Database). Het project 'Website' is een ASP.NET website. Door twee projecten in één Solution te steken kunnen deze tegelijk getest en gedebugd worden. In beide projecten zit een C# klasse. Deze dient om de communicatie met de database te regelen.

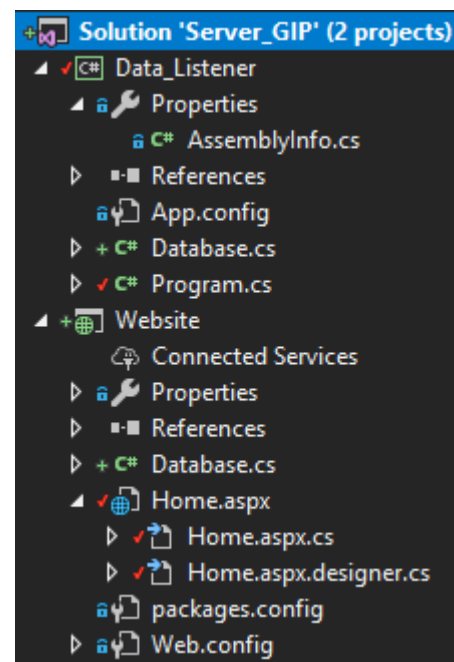


FOTO 1

### 1.2.7.2 Data\_Listener

Dit project is een gewone consoleapplicatie. Het luistert op een TCP-poort naar inkomende verbindingen. Wanneer het meettoestel verbinding maakt zal het de data ontvangen en testen of deze geldig is. De inkomende data moet dus voldoen aan bepaalde voorwaarden. Er moet bijvoorbeeld getest worden of sommige sensorwaarden geen NULL bevatten. De data wordt dan in de database gezet. Dit gebeurt via de database klasse. Deze consoleapplicatie dient dus enkel en alleen om data te ontvangen van het meettoestel.

### 1.2.7.3 Website

De website is een ASP.NET project. Met ASP kunnen we een website bouwen met HTML en CSS, achter de gebruiksvriendelijke UI draait dan een code behind file geschreven in C#. Hiermee kunnen we de website interactief maken. De code behind draait volledig op de server. Om toch variabelen mee te sturen maken we gebruik van de ViewState methode. Maar omdat we gebruikersnamen en wachtwoorden versturen moeten we deze encrypteren. Gelukkig kunnen pagina's in ASP voorzien worden van encryptie, door de op Foto 2 de eerste regel de in het rood onderlijnde parameters toe te voegen:

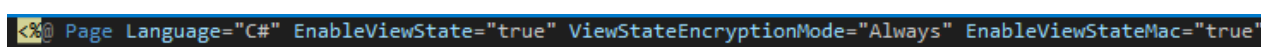
A screenshot of a code editor showing a line of ASP.NET code. The code is: <%@ Page Language="C#" EnableViewState="true" ViewStateEncryptionMode="Always" EnableViewStateMac="true" %>. The parameters EnableViewState="true", ViewStateEncryptionMode="Always", and EnableViewStateMac="true" are underlined in red.

Foto 2

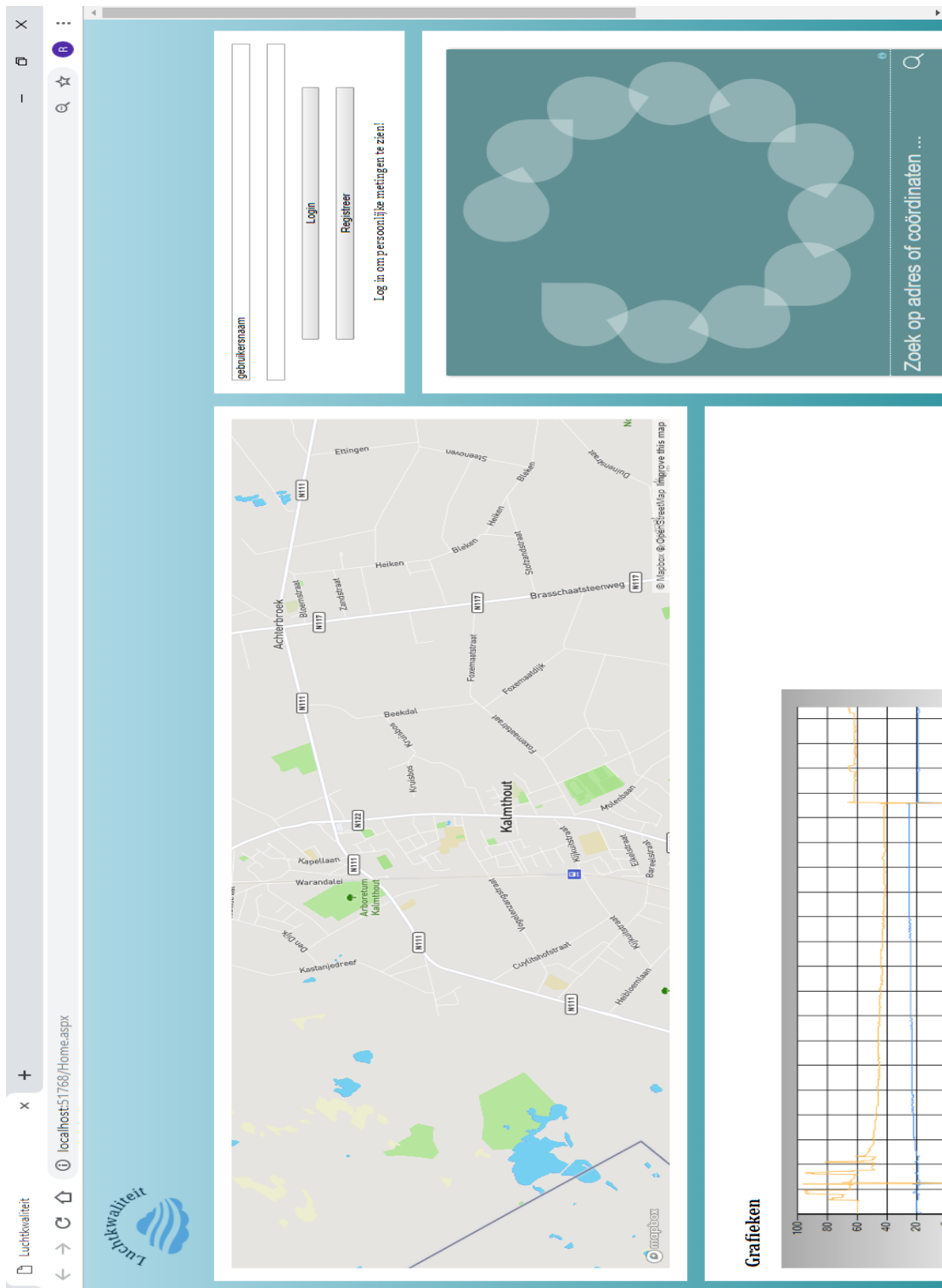
Het grafische ontwerp van de site staat bij 2.1 Lay-out website.

### 1.2.7.4 Client side

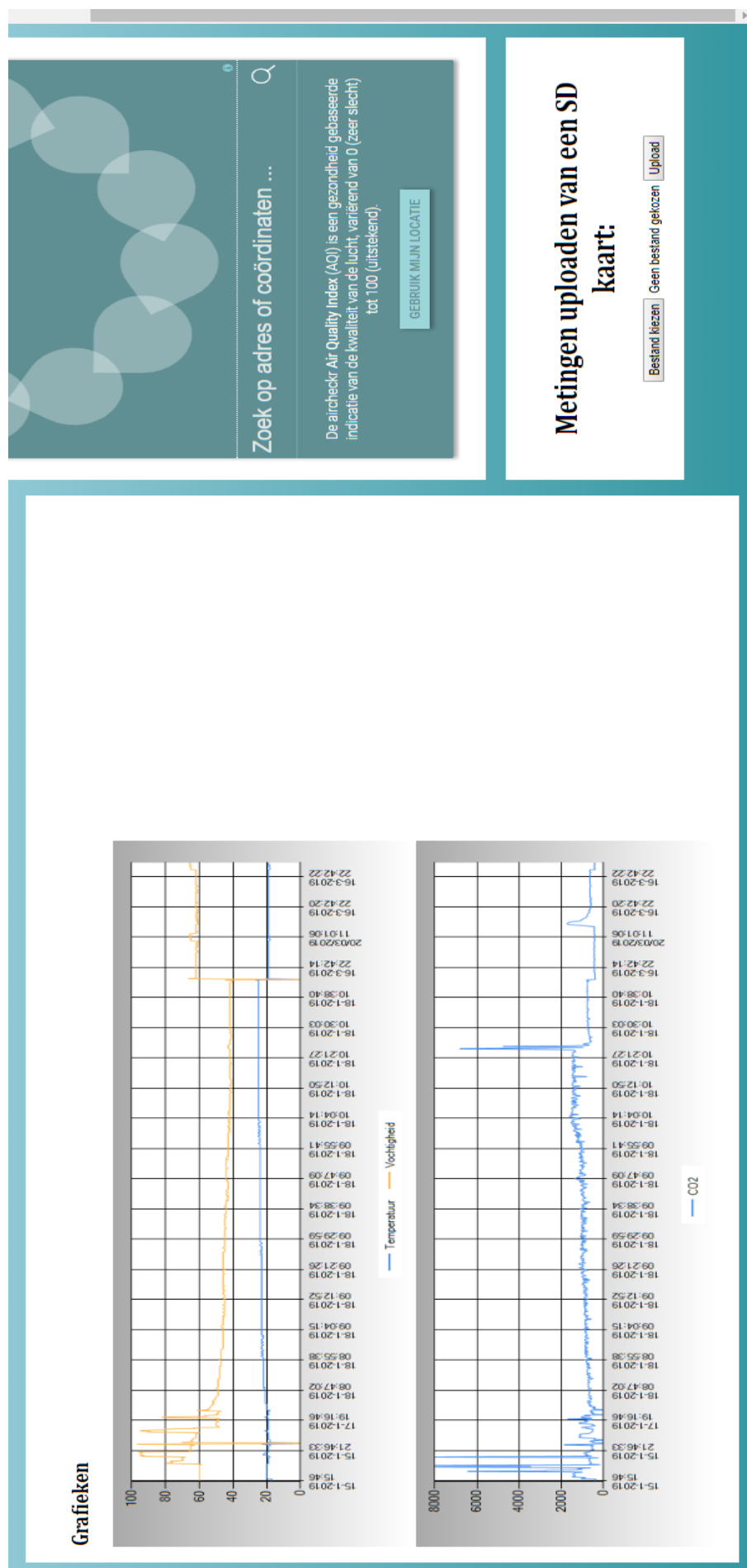
De webpagina draait op de client Javascript. Dit is een programmeertaal waarmee we communicatie met de server en client kunnen voeren zonder de pagina te vernieuwen. Ook gebruiken we dit om de kaart te tekenen. Javascript kan in de html-file staan maar ook in een apart bestand. We steunen op verschillende functies die deze taal biedt, waaronder de JQuery, Ajax en JSON onderdelen. JQuery en Ajax worden gebruikt om HTTP requests te sturen en te ontvangen. Deze worden meestal beantwoord met een response in JSON formaat. JSON betekend Javascript Object Notation. Het is een manier om classes van bijvoorbeeld C# om te zetten naar een string. Dit gebeurt met een serializer. De inhoud van de class wordt eigenlijk gewoon omgezet naar een string. JSON wordt enorm veel gebruikt. In het geval van dit project zal een functie in de client side javascript worden gecalled. In deze functie sturen we een request naar de server via http. Deze kan beantwoord worden via een WebMethod of een API. Het antwoord bestaat uit een JSON-string met daarin de gemeten waardes van de user die de pagina opvraagt. De response wordt doorgegeven aan de functie die de kaart zal renderen op het scherm van de user.

## 2. Schema's en tekeningen

### 2.1 Lay-out website

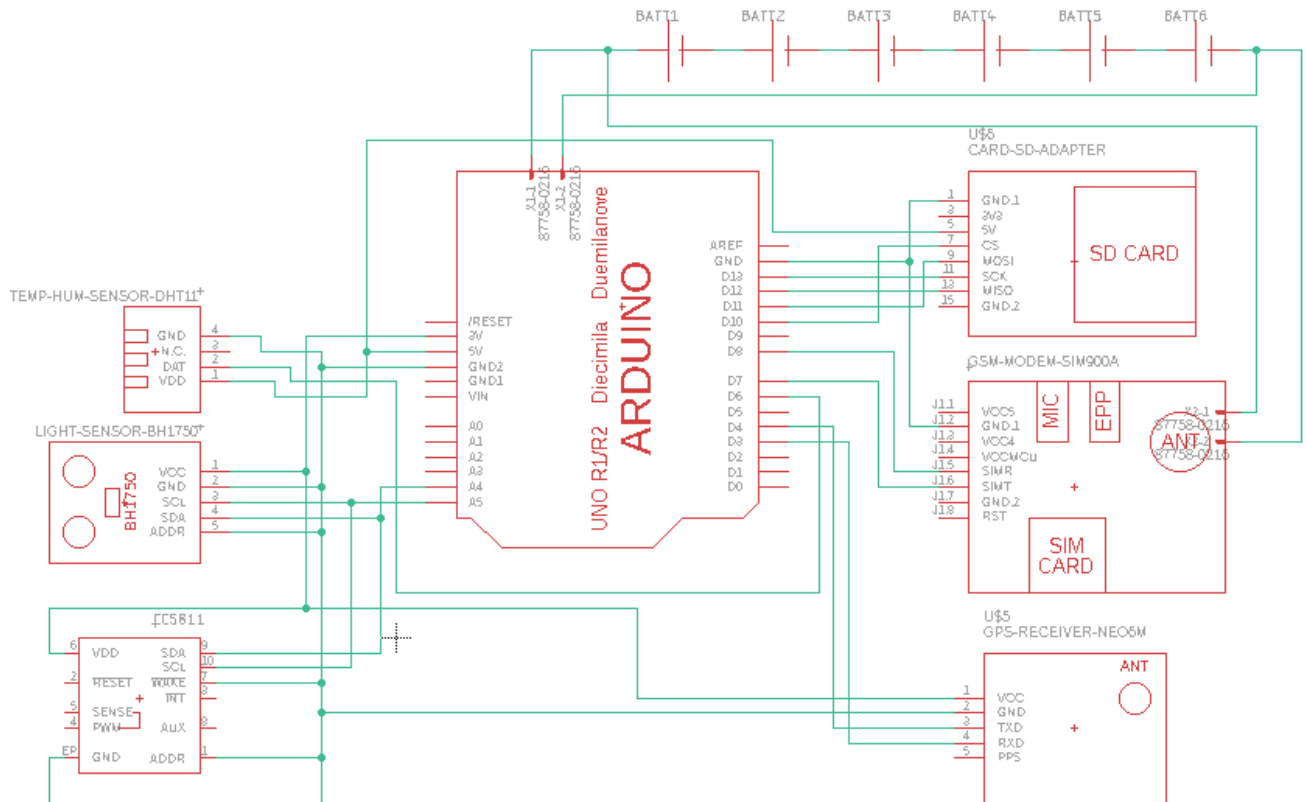


FIGUUR 3



Op de grafische weergave van de website die u kan terugvinden op foto 3 en 4 kan deze opgedeeld worden in 5 delen. Namelijk de map die onder het logo is terug te vinden. Onder de map staan grafieken van de CO<sub>2</sub>, temperatuur en vochtigheid in de functie van tijd. Bovenaan in de rechter kolom kan een al reeds geregistreerde gebruiker inloggen of een account aanmaken door op registreren te drukken. Daaronder staat een API van aircheckr, hiermee kan de gebruiker de luchtinformatie van de ingegeven locatie bekijken. Onder de API kan er wanneer de gebruiker ingelogd is meetwaarden uploaden dat opgeslagen werden op de SD-kaart van het meetapparaat.

## 2.2 Meettoestel



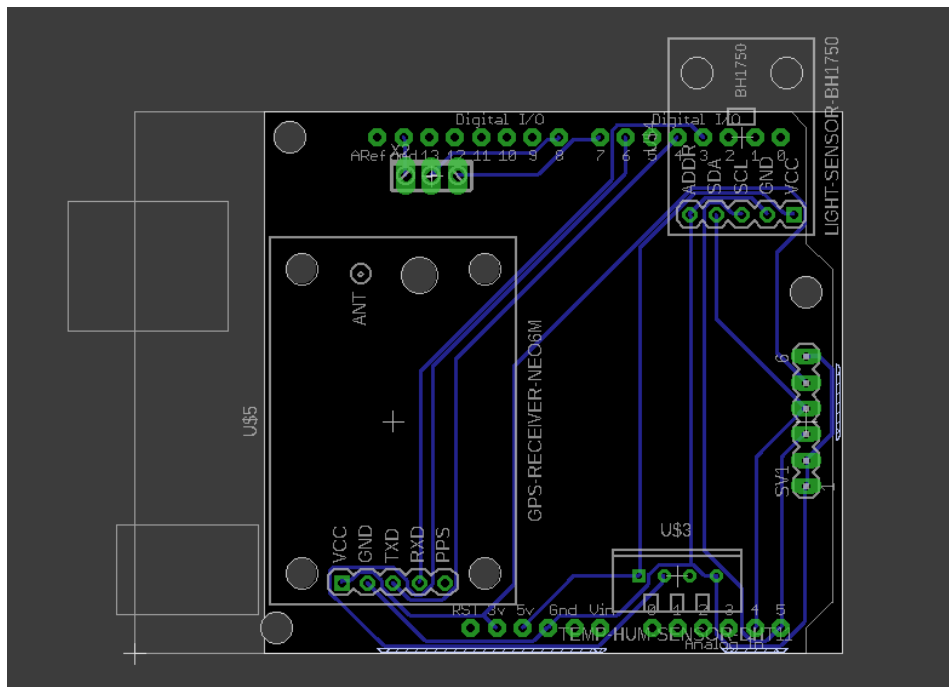
FIGUUR 5

Op figuur 5 kan u het volledige schema van het meet toestel zien. Verdere informatie over de verschillende sensoren en hoe deze werken kan u in hoofdstuk 3 vinden.

## 2.3 PCB

De pcb is gebaseerd op een arduino uno bord. Dit maakt het makkelijk om het te stacken met onze microcontroller. Niet alle onderdelen staan op de pcb. Sommige gaan we verbinden met draden. We solderen uiteraard alle verbindingen. De draden dat we zelf nog gaan verbinden gaan wikkelen om EMI-storingen te verminderen. Meer uitleg over EMC en EMI kan u terugvinden in 3.8 Elektromagnetische comptabiliteit.

*Figuur 6 – PCB desing in Eagle*





## 2.3 Netwerkdigram

Op foto 6 kan u zien dat er via een mobiele dataverbinding een connectie met de server gemaakt zal worden. Wanneer de gebruiker kiest om de gemeten waarden live via de 4G module te versturen.

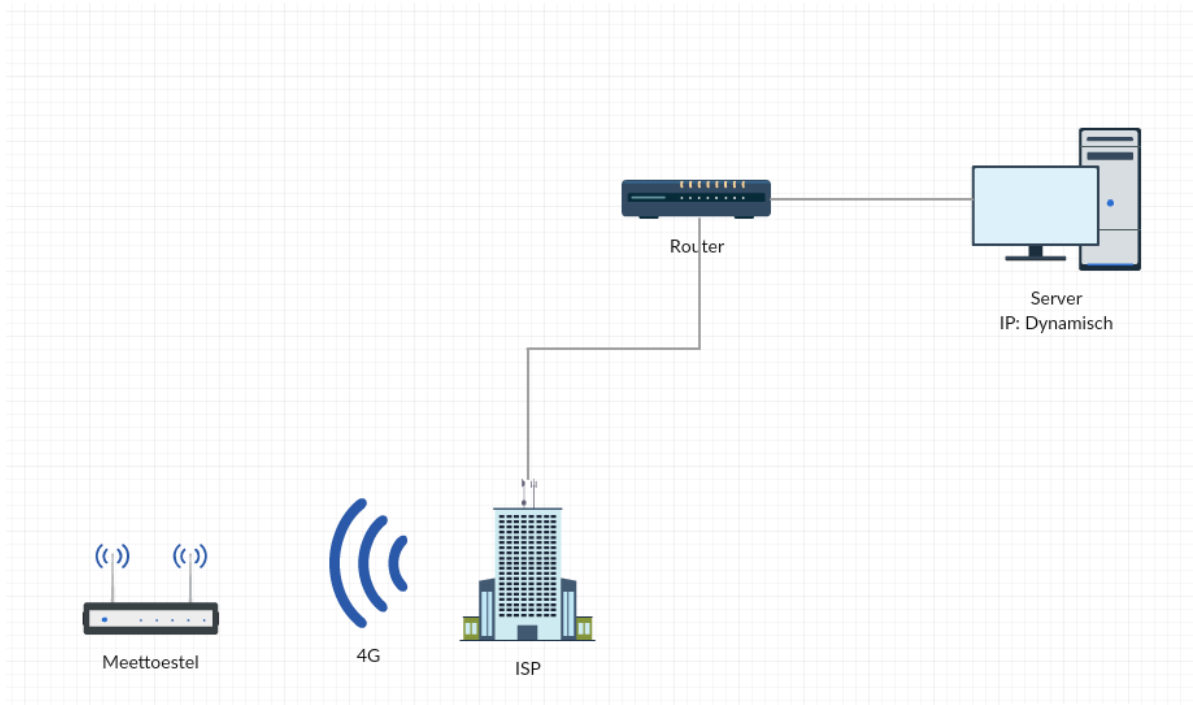


FOTO 6

## 3. Elektronica

In dit hoofdstuk van het GIP gaan we kijken naar de hardwarecomponenten die we gekozen hebben en bespreken hoe ze werken. We hebben in totaal vier 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 één sensor, de DHT11. Deze sensor heeft drie aansluitingen, VCC en GND voor 5V voeding van de Arduino en DAT om de data over te dragen. Om de dataoverdracht via één draad te laten verlopen wordt een seriële enkele draad protocol gebruikt. De Arduino zal hierdoor zijn aanvraag sturen en de data terugkrijgen over dezelfde draad. Dit gaat als volgt te werk: de Arduino zal eerst een aanvraag doen om data te krijgen, dit gebeurt door de DAT pin naar een logisch niveau “0” te trekken voor iets langer dan 18ms, dan wordt deze pin terug naar een logisch niveau “1” opgetrokken. Vervolgens zal de DHT11 een reactie geven van ~54µs een logische “0” en daarna ~80µs een logische “1”. Tenslotte worden de gegevens overgedragen in vijf segmenten van acht bits. Deze methode hebben we zelf kunnen schrijven door de bibliotheek “dht.h”, anders had dit proces op bit niveau geprogrammeerd moeten worden. De DHT11 is een digitale sensor, daarom zal het ook aan een digitale pin aangesloten worden, namelijk op pin 6 van de Arduino. De temperatuur wordt gemeten door een NTC temperatuur sensor en de vochtigheid wordt gemeten door een vochtigheidsafhankelijke weerstand.

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	± 5 %RH	± 2°C	1	4 Pin Single Row

FOTO 7

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%.

VDD=5V, T = 25°C (unless otherwise stated)

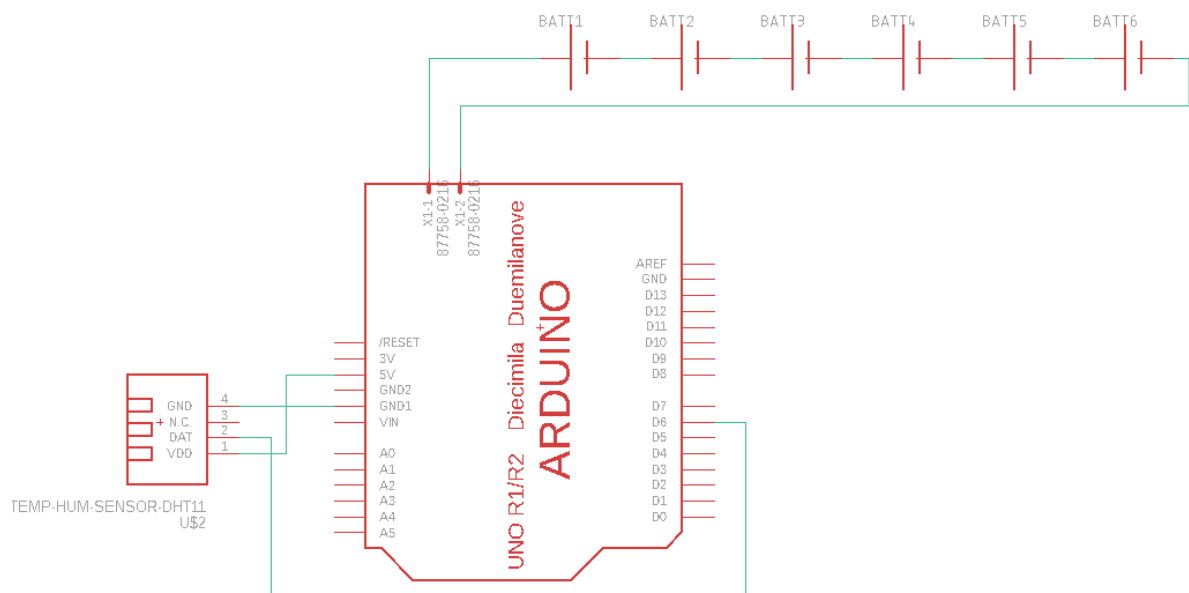
	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

FOTO 8

In de datasheet vinden we ook welke voedingsspanning er gebruikt moet worden. Hiervoor kunnen we best de kenmerkende spanning gebruiken dus in dit geval is dat 5V. Deze waarde wordt geleverd door de Arduino en om er zeker van te zijn kunnen we berekenen hoeveel vermogen de sensor maximaal vraagt. We weten dat het maximaal vermogen niet hoger mag zijn dan ¼ W dus:

$$P = U \times I = 5V \times 2,5mA = 12,5mW .$$

Deze waarde is dus onder het maximaal vermogen van de Arduino en kan dus gevoed worden. Ook is het belangrijk om rekening te houden met de sample periode, dit wil zeggen hoe snel er een meting achter elkaar uitgevoerd kan worden. We mogen de meting dus niet herhalen binnen één seconde.



FIGUUR 9

In figuur 9 ziet u de schematische opstelling van hoe de DHT11 sensor op de Arduino Uno wordt aangesloten. De Arduino Uno wordt gevoed door zes in serie geschakelde AA-batterijen.

## 3.2 BH1750

Om lichtsterkte te meten maken we gebruik van een BH1750 sensor die lichtsterkte kan meten van 1lux tot 65535 lux. De sensor wordt aangesloten met vijf aansluitingen om data te versturen en de communicatie te onderhouden met de Arduino daarnaast is er nog de GND en VCC dat worden gebruikt om de sensor te voeden, dit bedraagt 3,3V dat van de Arduino afkomstig is. 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 adresaansluiting, deze wordt verbonden met de GND van de Arduino zodat de sensor weet dat het een slaaf is. Om de data naar de Arduino te sturen wordt een I<sup>2</sup>C-bus gebruikt. Een I<sup>2</sup>C-bus bevindt zich enkel tussen de pinnen A4 en A5, maar er kunnen nog sensoren aan deze pinnen gekoppeld worden, want dit protocol ondersteund meerde slaven (multiple slaves) over dezelfde datalijnen. Om deze overdrachtsmethode te gebruiken moet er aan het programma een bibliotheek genaamd "BH1750FVI.h" toegevoegd worden. Deze bibliotheek zal er voor zorgen dat de overdracht geprogrammeerd kan worden met woorden die we kunnen gebruiken en niet op bitniveau.

### ● Operating Conditions

Parameter	Symbol	Ratings			Units
		Min.	Typ.	Max.	
Vcc Voltage	Vcc	2.4	3.0	3.6	V

FIGUUR 10

In de datasheet vinden we terug met welke spanning de sensor gevoed moet worden. Hiervoor kunnen we best de kenmerkende spanningswaarde gebruiken, dit is 3V. We zouden graag als voeding de Arduino gebruiken maar die heeft geen 3V uitgang. Wat we wel op de Arduino vinden is een 3,3V uitgang en dan kunnen we zien dat deze waarde ligt binnen het werkspanningsgebied.

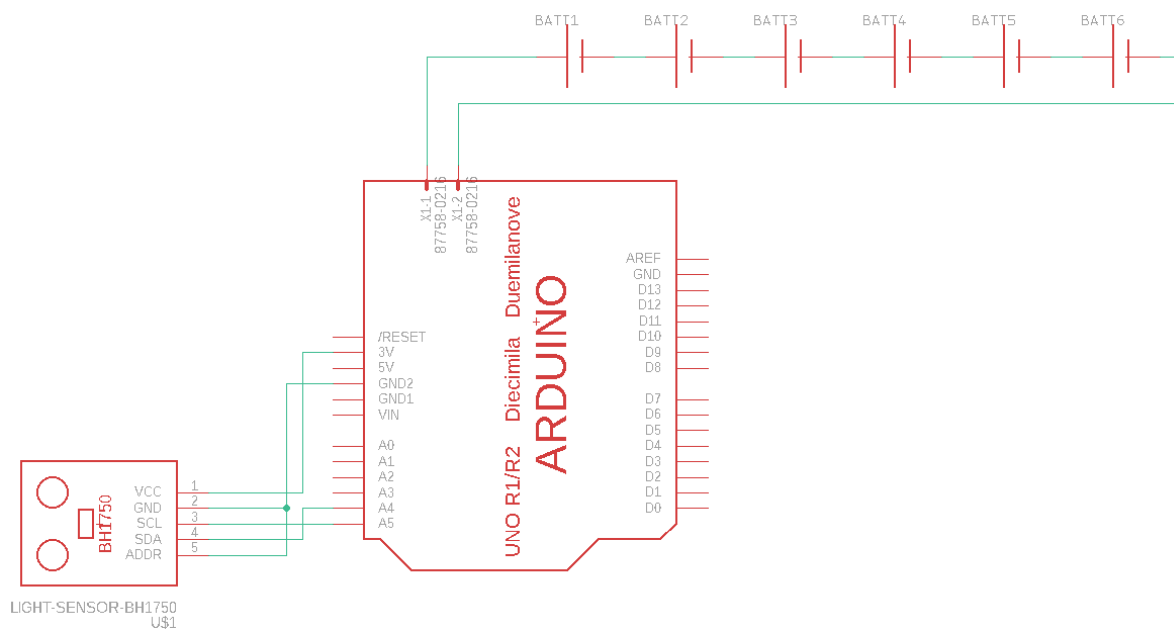
Parameter	Symbol	Limits			Units
		Min.	Typ.	Max.	
Supply Current	Icc1	—	120	190	µA

FOTO 11

Om er zeker van te zijn dat de sensor door Arduino gevoed kan worden, gaan we moeten nagaan of het maximaal vermogen niet overschreven wordt. Deze waarde is ¼ W dus:

$$P = U \times I = 3,3V \times 190\mu A = 627\mu W.$$

Deze waarde is onder het maximaal vermogen van de Arduino dus er moeten geen extra maatregelen genomen worden.



FIGUUR 12

Op figuur 12 zien we de testopstelling van de lichtsensor. Hierop is te zien dat de sensor verbonden en gevoed is door de Arduino Uno. De Arduino zelf wordt gevoed door zes in serie geplaatste AA-batterijen.

### 3.3 NEO-6M

De NEO-6M in onze schakeling gaat gps coördinaten van zijn locatie achterhalen en sturen naar de Arduino. Er zal met de gps ook snelheid gemeten worden om te bepalen hoe snel metingen uitgevoerd worden. Hiermee wordt ook de tijd achterhaald van wanneer een meting werd uitgevoerd. 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, dit is volgens het Universal Receiver/Transmitter protocol. Dit soort communicatie wordt ook gebruikt bij USB-verbindingen. Om dit soort verbinding te gebruiken wordt de “SoftwareSerial.h” bibliotheek gebruikt zodat de gebruiker niet op bit-niveau moet werken. Verder kan hij door deze bibliotheek ook een transmissiesnelheid instellen, kiezen welke pinnen er gebruikt worden als TX en RX poort en nog meerdere commando's. Verder wordt er ook de “TinyGPS++.h” gebruikt, deze bibliotheek zal ervoor zorgen dat de inkomende data omgezet wordt naar de breedte- en lengtegraad die we daarna tonen op de map.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	1.75	1.8	1.95	V	
		NEO-6Q/M	2.7	3.0	3.6	V	
		NEO-6P/V/T					

FIGUUR 13

Om te weten hoeveel spanning er nodig is om de sensor te voeden kunnen we dit terugvinden in de datasheet. De beste spanning waarmee we de sensor voeden is 3V, maar zoals bij de BH1750 valt de 3,3V van de Arduino ook tussen de minimale en maximale spanningswaarden. Dus kunnen we de 3,3V van de Arduino gebruiken als voeding.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Max. supply current <sup>15</sup>	I <sub>ccp</sub>	All			67	mA	V <sub>CC</sub> = 3.6 V <sup>16</sup> / 1.95 V <sup>17</sup>

FIGUUR 14

Om er zeker te zijn dat de Arduino dit kan voeden, gaan we berekenen wat het maximale vermogen van de gps is. In de datasheet hebben we gevonden dat de maximale voedingsstroom 67mA bedraagt. Die maximale waarde is ¼ W dus:

$$P = U \times I = 3,3V \times 67mA = 221,1mW$$

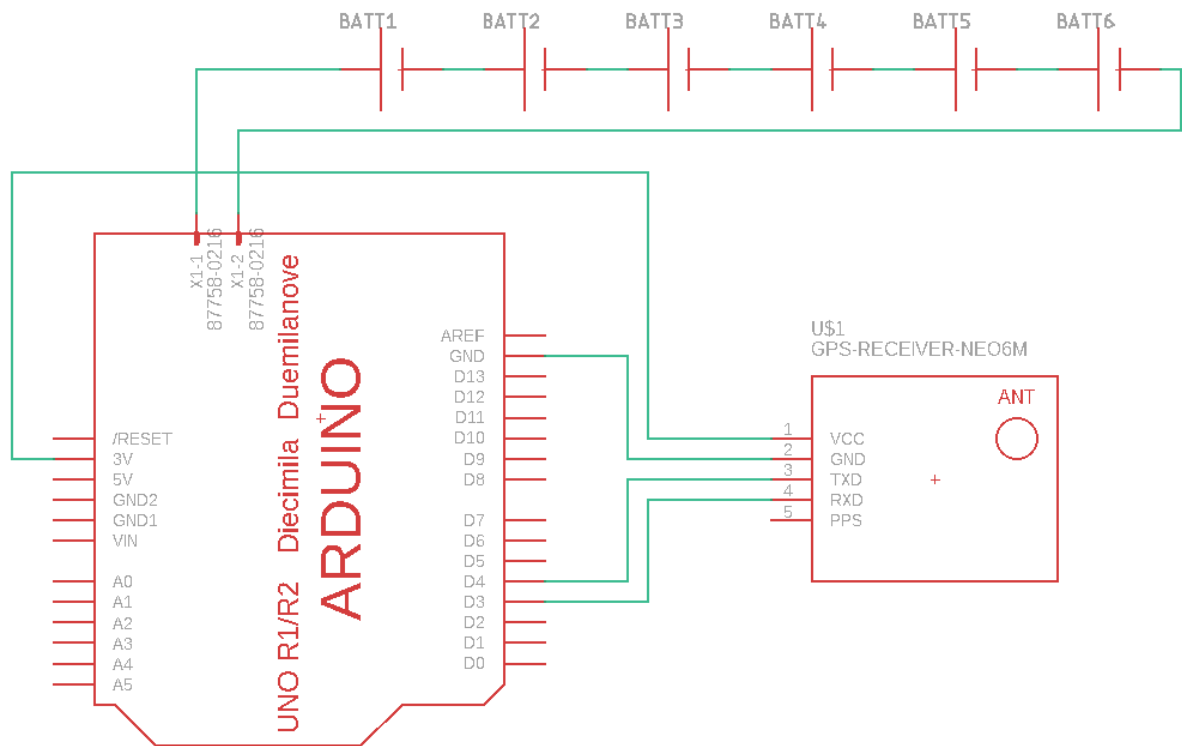
Hierdoor kunnen we concluderen dat de gps door de Arduino gevoed kan worden.

Interface	Settings
Serial Port 1 Output	9600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: <b>GGA, GLL, GSA, GSV, RMC, VTG, TXT</b> (In addition to the 6 standard NMEA messages the NEO-6T includes <b>ZDA</b> ).

FIGUUR 15

Bij het schrijven van de code is het ook handig om te weten wat er reeds ingesteld is in de GPS-module. Op de screenshot hierboven is te zien dat er al een Baud rate is ingesteld. Dit bepaald hoe snel de dataoverdracht naar de Arduino gaat, dit hoeft niet meer in de code ingesteld worden.





FIGUUR 16

Op figuur 16 ziet u de GPS-module aangesloten op de Arduino Uno. De Arduino is gevoed door zes AA-batterijen.

## 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. Dit soort overdracht wordt Universal Receiver/Transmitter of UART genoemd. Dit soort communicatie wordt ook gebruikt bij USB-verbindingen. Om dit soort verbinding te gebruiken wordt de “SoftwareSerial.h” bibliotheek gebruikt zodat de gebruiker niet op bit niveau moet werken. Verder kan hij door deze bibliotheek ook een transmissiesnelheid instellen, kiezen welke pinnen er gebruikt worden als TX en RX poort en nog meerdere commando's. Om de dataoverdracht te realiseren kunnen we via AT-commando's de shield aansturen en zo een verbinding met de server opstellen en berichten versturen. 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 een GSM-mast willen maken, gaan we in ons programma de access point name van de provider ingeven zodat er via de provider een toegang tot het WAN is en er een publiek IP-adres aan de shield wordt geven. Wanneer er toegang is tot het WAN kunnen we door middel 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.

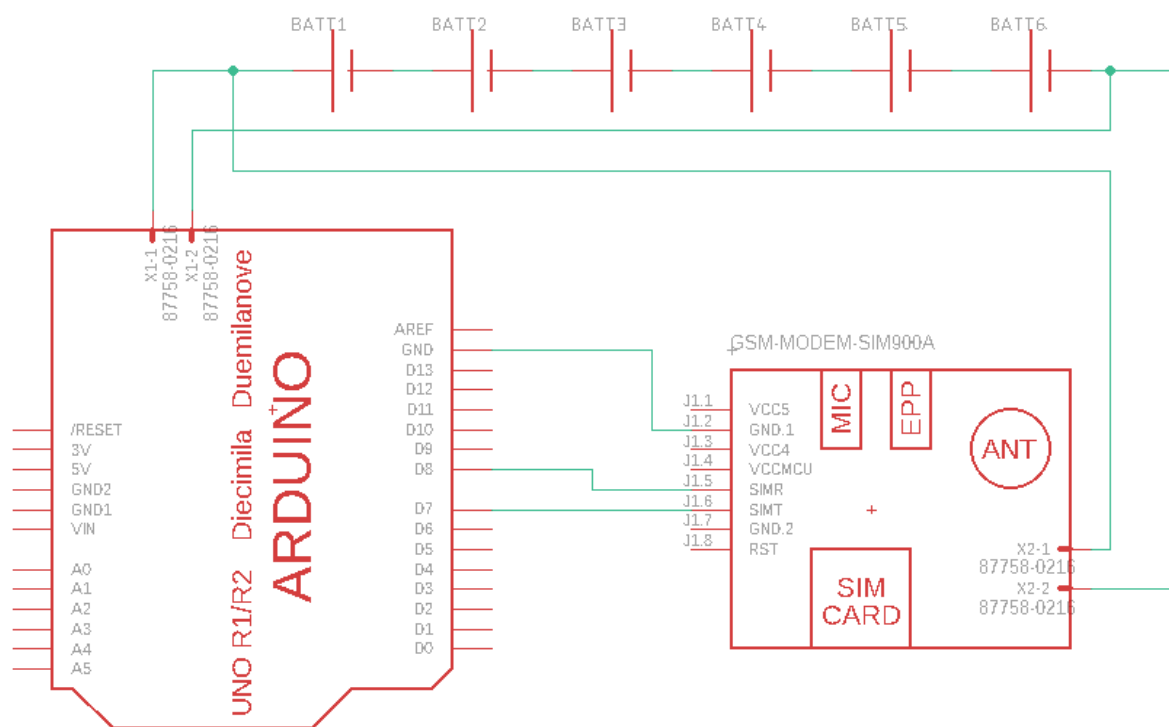
Parameter	Condition	Min.	Typ.Max.	Max.	Uints
<b>Vin</b>		5		12	V
<b>Current</b>	Sleep Mode			1.5	mA
	Voice Call			250	
	GPRS Data Mode	76		440	
	During Tx Burst			2	A
<b>Dimension</b>		55*81			mm

FIGUUR 17

Om de SIM 900 shield te voeden gaan we gebruik maken van zes AA-batterijen van elks 1,5V die in serie geplaatst zijn. Volgens de formule  $U_{tot} = U_1 + U_2 + U_3 + \dots$  wat een totale spanning maakt van 9V. Die spanning valt binnen het bereik van de werkspanning van de shield. We gaan de SIM900 enkel gebruiken voor een GPRS verbinding met de server, in de tabel kan je zien dat de maximale stroom bij "GPRS Data Mode" 440mA zal zijn.

Hiervoor moeten de batterijen een vermogen kunnen lever van :

$$P = U \times I = 9V \times 440mA = 3,96W$$



FIGUUR 18

Op figuur 18 ziet u de SIM900 shield verbonden met de Arduino Uno. Beide zijn gevoed door zes in serie geplaatste AA batterijen.

## 3.5 CCS 811

De CCS 811 sensor gaat de hoeveelheid CO<sub>2</sub> hoeveelheid per m<sup>2</sup> meten. De sensor heeft zeven aansluitingen waarvan de GND en VCC gebruikt worden door een 3,3V voeding van de Arduino Uno. Daarnaast is voor de dataoverdracht pin “SCL” en “SCA” voorzien. 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” pin, deze pin sluiten we aan op poort “A4” van de Arduino en wordt gebruikt om data van de sensor naar de Arduino te sturen. Er is ook nog een “WAKE” aansluiting, die wordt aangesloten aan de “GND” zodat de sensor begint te werken. De aansluitingen “RST” en “INT” worden niet gebruikt. Wanneer die pinnen zijn aangesloten wordt er met de Arduino een verbinding gemaakt door gebruik van het I<sup>2</sup>C-protocol. Een I<sup>2</sup>C-bus bevindt zich enkel tussen de pinnen A4 en A5, maar er kunnen nog sensoren aan deze pinnen gekoppeld worden want dit protocol ondersteund meerde slaven over dezelfde datalijnen. Om deze overdrachtsmethode te gebruiken moet er aan het programma een bibliotheek genaamd “SparkFunCCS811.h” toegevoegd worden. Deze bibliotheek zal er voor zorgen dat de overdracht geprogrammeerd kan worden met woorden dat we kunnen begrijpen en niet op bit-niveau. Ook zal er door deze bibliotheek data naar een voor ons begrijpbare taal omgezet worden.

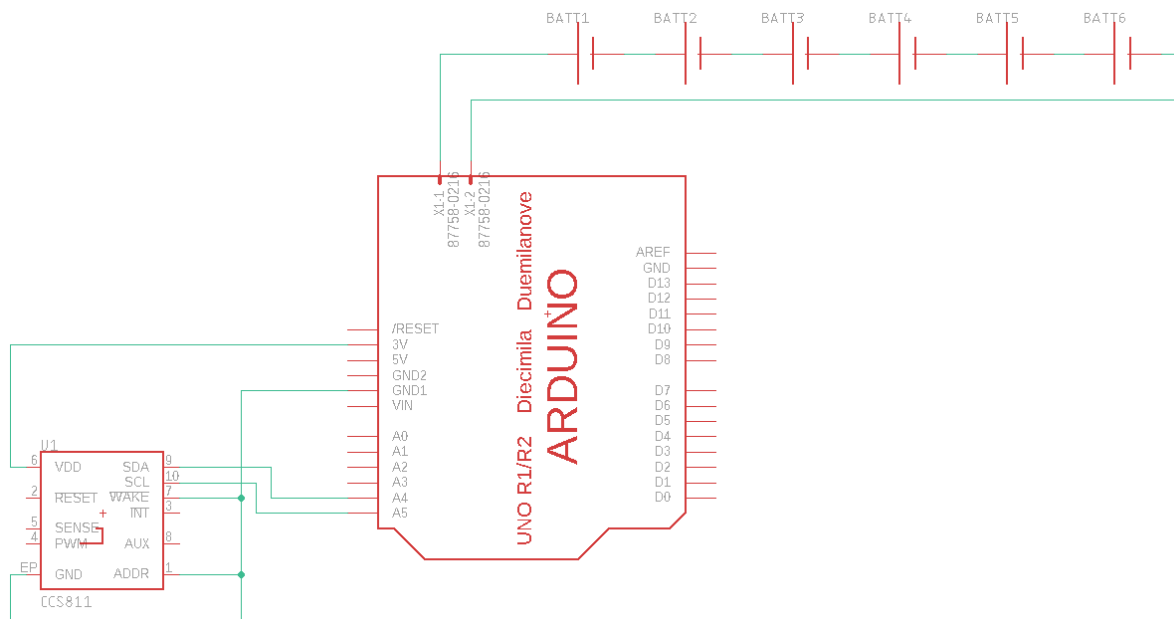
Symbol	Parameter	Min	Max	Units	Comments
<b>Electrical Parameters</b>					
V <sub>DD</sub> <sup>(1)</sup>	Supply Voltage	1.8 <sup>(2)</sup>	3.6	V	
I <sub>DD</sub>	Supply Current		30	mA	In mode 1

FIGUUR 19

Om de spanningswaarde van de voeding te bepalen gaan we de datasheet raadplegen. Hierin gaan we zoeken naar de waarden op figuur 19. Daarop kunnen we zien dat de waarde van voedingsspanning zich moet bevinden tussen 1,8V en 3,6V. Dit wil zeggen dat de Arduino 3,3V geschikt is voor de voeding voor de sensor. Om te controleren of de Arduino het maximaal gevraagde vermogen kan leveren gaan we dit bepalen met de formule:

$$P = U \times I = 3,3V \times 30mA = 99mW$$

Het maximaal vermogen dat de Arduino Uno kan leveren is ¼ W dus de manier van voeden is mogelijk.



FIGUUR 20

Op figuur 20 kan u de CCS 811 sensor zien die verbonden is met de Arduino Uno. De Arduino Uno wordt gevoed door zes in serie geplaatste batterijen.

## 3.6 MicroSD card adapter

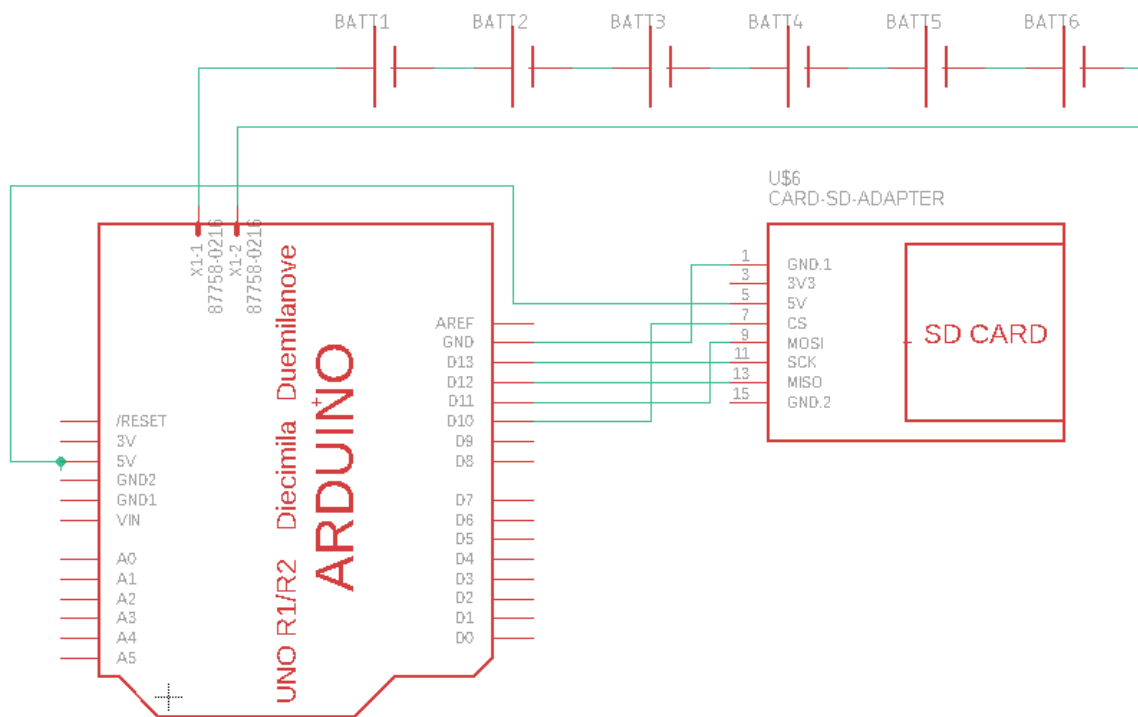
Om de gebruiker een keuze te geven hoe hij zijn gemeten waarden naar de server kan overzetten kan hij kiezen om dit via de 4G-module door te sturen of om dit lokaal op een SD-kaart te bewaren. Wanneer er wordt gekozen om dit op een SD-kaart te bewaren is het aangeraden om de metingen nadien op de website te uploaden. De adapter dat wordt gebruikt is de “MicroSD card adapter” van Catalex. Hierop zijn zes aansluitpinnen te vinden waarvan twee (VCC en GND) voor voeding zorgen. De adapter wordt aangesloten op 5V van de Arduino maar heeft voor de SD-kaart enkel 3,3V nodig, dit vanwege een spanningsregulator geïntegreerd op het board. Communicatie met de Arduino wordt gevoerd via het SPI protocol, waarbij de adapter de slaaf is. De master (de Arduino) zal het kloksignaal naar de adapter sturen om de dataoverdracht snelheid te bepalen. Daarnaast is er nog de “slave select” aansluiting, om communicatie te voeren met de desbetreffende slaaf. Voor de data over te dragen van en naar de Arduino worden de “Mosi” (master output, slave input) en “Miso” (master input, slaver output) pinnen gebruikt. Dit betekend dat er aan de Mosi-pin data naar de adapter wordt gestuurd en aan de Miso-pin omgekeerd.

### Features

regulator circuit; Supports Micro SD up to 2GB; Micro SDHC up to 32GB; Leads: GND, VCC, MISO, MOSI, SCK, CS; SPI

FIGUUR 21

Op figuur 21 kan u een zien welke micro-SD kaarten worden ondersteund door de adapter. Om tekst op te slaan tot het wordt geüpload naar de server is er niet veel opslagplaats nodig. Dus voor deze toepassing is de micro-SD adapter geschikt.

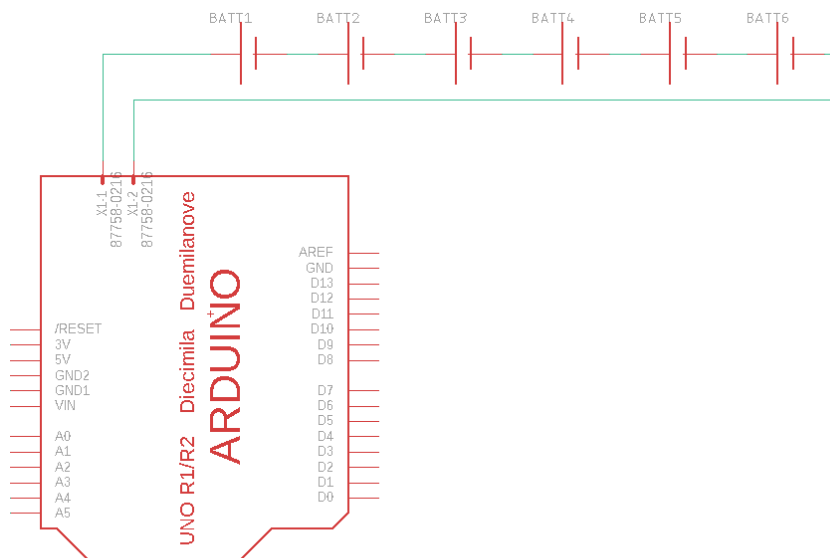


FIGUUR 22

Op figuur 22 kan u in schematische weergave zien hoe we de SD-card adapter aangesloten hebben aan de Arduino Uno.



### 3.7 Arduino Uno



FIGUUR 23

Eén van de belangrijkste componenten van het meetapparaat is de Arduino Uno. Dit is een microcontroller waar een code ingeladen kan worden via een USB-verbinding met een computer. De code is geschreven in de programmeeromgeving Arduino IDE in de taal C++. De Arduino gaat zijn code lijn per lijn doorlopen, bovenaan staan de gebruikte bibliotheken die nodig zijn om bepaalde sensoren in te lezen en hun data om te zetten naar voor ons begrijpbare metingen.



FIGUUR 24

Op de site van Arduino kunnen we bij het product “Arduino Uno” de specificaties terugvinden. Op de figuur 22 zien we dat de externe voeding tussen 7V en 12V moet kunnen leveren. De zes AA-batterijen dat we willen gebruiken leveren elks 1,5V wat in totaal een spanning van 9V levert. Die opstelling kan dus gebruikt worden.

## 3.8 Elektromagnetische compatibiliteit

Elektromagnetische compatibiliteit ofwel EMC zorgt er voor dat een apparaat de elektromagnetische omgeving niet zodanig beïnvloedt dat de functies van andere apparaten en systemen nadelig worden beïnvloed. Deze manier van beïnvloeden is onder andere bekend als EMI (Elektromagnetische interferentie). Bij het bouwen van ons project is het belangrijk om rekening te houden met zowel de opgewekte EMI en de bestendigheid er tegen. Wanneer hier geen rekening mee wordt gehouden gaan er problemen optreden tussen de samenwerking van de verschillende sensoren modules in de behuizing.

EMI kan veroorzaakt worden door de mens maar ook door de natuur. Door de mens veroorzaakte EMI komt voornamelijk voort uit elektronische circuits of door het schakelen van grote stromen. Wanneer dit door de natuur is veroorzaakt kan dit ontstaan zijn door kosmische ruis, bliksem en andere atmosferische soorten lawaai.

Om EMI tegen te gaan werden EMC-voorwaarden opgesteld, onder de vereisten bevonden zich de EMC-richtlijn. Hierin stond aan welke voorwaarden alle apparatuur die "elektromagnetische storingen kon veroorzaken of waarvan de werking door dergelijke storingen kan worden aangetast" moeten voldoen.

Om een circuit aan de EMC-eisen te voldoen moet er met een aantal zaken rekening gehouden worden. Ten eerste moet de bekabeling voorzien zijn van EMC-maatregelen, dit kan gebeuren door verschillende draden te draaien. De draden kunnen ook nog gescreend worden voor minder interferentie. De signalen op de printbanen kunnen gefilterd worden, dit kan de kwaliteit van het signaal beïnvloeden. Het is ook aangeraden om de EMC-kritieke delen te onderscheiden van de andere delen. Een belangrijk onderdeel in de ontwikkeling van een print is de aarding, deze moet steeds een lage ingangsimpedantie hebben.



## 4. Elektriciteit

### 4.1 Vitale 8

De vitale 8 zijn regels die nageleefd moeten worden wanneer een er werken of een aanpassing gebeurt aan een elektrische installatie. Deze regels zorgen voor de veiligheid van de mensen die werken aan de installatie en in de omgeving.

#### 1. Werkzaamheden voorbereiden

- Nagaan welke maatregelen er genomen moeten worden om veilig te werk te gaan.

#### 2. Vrij schakelen

- De installatie waaraan je wil werken loskoppelen van alle voedingsbronnen.

#### 3. Voorkom herinschakeling

- Indien er hulpbronnen aanwezig zijn moeten deze ook buiten werking gesteld worden.

#### 4. Controleer de spanningsafwezigheid

- Ga met een meettoestel na of er nog spanning aanwezig is op de geleider binnen het gebied waar er werken of aanpassingen moeten gebeuren.

#### 5. Aarden, ontladen en kortsluiten

- Binnen de werkzone moeten alle hoogspanningsinstallaties geaard zijn en laagspanningsinstallaties dat ongewild spanning kunnen voeren moeten ook geaard zijn.

## 6. Afbakenen en/of afschermen van de werkzone

- Op werkplaatsen waar in de werkzone andere elektrische installaties zijn moet er een afbakening of een afscherming gebeuren volgens het AREI.

## 7. De elektrische installatie vrijgeven

- Een werkverantwoordelijke is de enige persoon dat de toestemming aan het uitvoerend personeel kan geven om de werken te starten.

## 8. Terug onder spanning brengen

- Wanneer de werkverantwoordelijke zeker is dat de elektrische installatie terug op veilige wijze onder spanning gebracht kan worden zal hij dit aan de installatieverantwoordelijke melden. Wanneer dat is gebeurd zal onder verantwoordelijkheid van de installatieverantwoordelijke de spanning terug ingeschakeld worden.

## 4.2 Elektrische veiligheid

De Arduino en GSM-module hebben een ingebouwde voltageregulator. Dit betekent dat de 9V van de batterij naar 5V wordt 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, etc. Gelukkig zijn zowel de Arduino als de GSM-module beveiligd met een interne zekering zoals op foto 1.



*.500mA smd zekering*

FOTO 1



## 5. Logboek

DATUM:	WAT:	NAAM:	TIJD:	PLAATS:
9/10/2018	GIT repository	Samen	1 uur	School
9/10/2018	Vochtsensor	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 CO <sup>2</sup> 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 CO <sup>2</sup> sensor	Samen	3 uur	School
19/11/2018	Bestellijst	Arthur	30 min	Thuis
17/12/2018	Bundel	Arthur	1 uur	School
26/12/2018	Visual Studio projecten opgestart	Arthur	3 uur	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 30 min	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, Sensoren, schema's, bronnen	Ruben	6 uur	Thuis
10/01/2019	Bundel	Arthur	2 uur 45 min	Thuis
10/01/2019	PPT, Bundel	Ruben	2 uur	Thuis
12/01/2019	Website	Arthur	3 uur 40 min	Thuis
10/3/2019	JSON toevoegen aan website	Arthur	3uur 20 min	Thuis
12/3/2019	SD-adapter aan Arduino	Ruben	3 uur	Thuis
14/3/2019	SD-card data uploaden naar website	Ruben	4 uur	Thuis
15/3/2019	Grafieken SD- card data uploaden	Ruben	4 uur	Thuis
16/3/2019	Bundel SD- adapter, aanpassingen 1 <sup>ste</sup> evaluatie	Ruben	5 uur	Thuis
19/3/2019	Bundel EMC toegevoegd	Ruben	4 uur	Thuis

20/3/2019	bundel, to do list aangevuld	Ruben	1 uur 30min	Thuis
21/3/2019	Bundel aanpassingen	Ruben	3 uur	Thuis


## 6. Reflectieverslag

## 7. Bibliografie

## 8. Bronnen

Bron: [www.slideshare.net](http://www.slideshare.net)

Op de website slideshare heb ik een volledige lijst gevonden van de AT commando's met uitleg.

Bron: [www.deviceplus.com](http://www.deviceplus.com)

Om de protocollen beter te begrijpen heb ik een uitleg gelezen van veelgebruikte protocollen bij Arduino.

Bron: [www.c-sharpcorner.com](http://www.c-sharpcorner.com)

Informatie over ViewState en encryptie.

Bron: [www.sparkfun.com](http://www.sparkfun.com)

Bibliotheek voor de CCS811 sensor.

Bron: [www.volta-org.be](http://www.volta-org.be)

Op deze site heb ik de betekenis van de vitale 8 gevonden en heb ik aan de hand van deze bron kunnen verwoorden.

Bron: [www.electronics-notes.com](http://www.electronics-notes.com)

Met behulp van electronics-notes hebben we de betekenis en rol van EMC in ons project kunnen plaatsen.