­

Odisee Technologiecampus Gent

Gebroeders de Smetstraat 1

9000 Gent

Smart frigobox

Realiseren van een zelfrijdende smart frigobox a.d.h.v. Arduino

Dries Bruggeman,

Oberon Lievens,

Aaron Lison &

Elian Van Cutsem

**Professionele bachelor ICT Mentor: Sabine Martens & Mario Wyns**

**2019-2020**



Odisee Technologiecampus Gent

Gebroeders de Smetstraat 1

9000 Gent

Smart frigobox

Realiseren van een zelfrijdende smart frigobox a.d.h.v. Arduino

Dries Bruggeman,

Oberon Lievens,

Aaron Lison &

Elian Van Cutsem

**Professionele bachelor ICT Mentor: Sabine Martens & Mario Wyns**

**2019-2020**

Abstract

Voorwoord

Inhoudsopgave

[Abstract 4](#_Toc38806311)

[Voorwoord 6](#_Toc38806312)

[Inhoudsopgave 8](#_Toc38806313)

[Figurenlijst 11](#_Toc38806314)

[Codefragmentenlijst 13](#_Toc38806315)

[Begrippenlijst 15](#_Toc38806316)

[Inleiding 16](#_Toc38806317)

[1 Opbouw 17](#_Toc38806318)

[1.1 Bouw van de frigobox 17](#_Toc38806319)

[1.2 Bouw van de frame 17](#_Toc38806320)

[2 Elektronica 17](#_Toc38806321)

[2.1 Elektonica van de frigobox 17](#_Toc38806322)

[2.2 Elektronica van de frame 17](#_Toc38806323)

[3 App 18](#_Toc38806324)

[3.1 Interface 18](#_Toc38806325)

[3.1.1 Onderzoek naar UI & UX 18](#_Toc38806326)

[3.1.2 Uiteindelijke UI 18](#_Toc38806327)

[3.1.3 Framework en code voor interface 19](#_Toc38806328)

[3.1.4 Code 19](#_Toc38806329)

[3.1.5 Gebruikte bibliotheken 19](#_Toc38806330)

[3.2 Connectiviteit 19](#_Toc38806331)

[3.2.1 Onderzoek naar beste keuze 19](#_Toc38806332)

[3.3 Joystick 20](#_Toc38806333)

[3.3.1 Onderzoek 20](#_Toc38806334)

[3.3.2 Gebruikte bibliotheek 20](#_Toc38806335)

[3.3.3 Uitlezen en verzenden van data 20](#_Toc38806336)

[3.4 Temperaturen 20](#_Toc38806337)

[3.4.1 Onderzoek 20](#_Toc38806338)

[3.4.2 Ontvangen van data 20](#_Toc38806339)

[4 Functies van de smart frigobox 20](#_Toc38806340)

[4.1 Temperatuurmeting 20](#_Toc38806341)

[4.2 Inventaris 20](#_Toc38806342)

[4.3 Verbinding 20](#_Toc38806343)

[4.4 Volgfunctie 20](#_Toc38806344)

[4.4.1 Onderzoek 21](#_Toc38806345)

[4.4.2 Richting bepalen 21](#_Toc38806346)

[4.4.3 Collision detection 22](#_Toc38806347)

[4.5 Smartlock 23](#_Toc38806348)

[5 Kostprijsberekening 23](#_Toc38806349)

[Conclusie 24](#_Toc38806350)

[Literatuurlijst 25](#_Toc38806351)

[Bijlagenoverzicht 27](#_Toc38806352)

[Bijlagen 28](#_Toc38806353)

Figurenlijst

Figuur 1: Interface van de startpagina 18

Figuur 2: Interface van een connecterende startpagina 18

Figuur 3: Interface van de temperaturen 18

Figuur 4: stelling van Pythagoras bij twee punten[20] 22

Figuur 5: ultrasone HC SR04 sensor 22

Codefragmentenlijst

[codefragment 1: De hoek tussen het kompas en het noorden bepalen 21](file:///C:\Git\verslag\hoofdstuk2ICTEO1.docx#_Toc38799244)

[codefragment 2: collision detection 23](file:///C:\Git\verslag\hoofdstuk2ICTEO1.docx#_Toc38799245)

Begrippenlijst

UI User Interface is een gebruikersomgeving of gebruikersinterface maakt interactie tussen mens en software mogelijk.[11]

UX User Experience of gebruikerservaring is een term die alle aspecten beschrijft van de ervaring van een gebruiker bij interactie met een product.[10]

Inleiding

In dit rapport wordt onderzocht hoe een zelfrijdende smart frigobox gevoed door draagbare batterij en gestuurd door twee gemotoriseerde wielen gerealiseerd kan worden. Deze realisatie gebeurt aan de hand van Arduino en een smartphoneapplicatie voor zowel Android als iOS. De frigobox zal zelf kunnen rijden aan de hand van een GPS-functie en echolocatie. De tijdspanne van dit onderzoek en de realisatie van het project is twaalf weken met als begindatum 9 februari 2020. Om dit project budget vriendelijk te houden is de maximale kostprijs €150.

De bedoeling van dit project is om de klassieke frigobox te vernieuwen. Het project start vanaf nul. De eerste stap is het ontwerpen van de smart frigobox. Deze bevat enkele functies waaronder de verbinding tussen een app en Arduino, een volgfunctie, temperatuurmeting en een inventaris. Vervolgens wordt de manier waarop de functionaliteiten tot stand kunnen komen onderzocht. Als laatste stap wordt een prototype gebouwd.

In dit rapport is de belangrijkste onderzoeksmethode een literatuurstudie. De voornaamste bronnen zijn handleidingen om de technische aspecten te begrijpen en te implementeren. Tijdens de opbouw van de frigobox wordt de werking geoptimaliseerd door diverse testen.

Het eerste hoofdstuk beschrijft de opbouw van de frigobox en het frame waarop de frigobox zich bevindt. Vervolgens wordt toegelicht hoe de elektronica werkt met alle functies die hierbij horen. Als derde, wordt de app besproken. Hier wordt uitgelegd hoe een connectie tot stand kan komen en hoe de juiste data van de gsm naar de Arduino geraakt. In het laatste hoofdstuk staan alle functies van de smart frigobox nog eens opgelijst met de nodige info. In de conclusie wordt verteld of de vooropgestelde doelstellingen behaald zijn en wat de mogelijke verbeterpunten zijn.

1. Opbouw
   1. Bouw van de frigobox
   2. Bouw van de frame
2. Elektronica
   1. Elektonica van de frigobox
   2. Elektronica van de frame
3. App
   1. Interface
      1. Onderzoek naar UI & UX

Allereerst wordt er onderzoek verricht naar hoe de applicatie in elkaar zit en welke functies deze bevat. Hierbij wordt bekeken welke functies nodig zijn om de smart frigobox te besturen via de app.

Uiteindelijk is beslist om volgende functies te voorzien:

* Een joystickcomponent om de frigobox handmatig besturen.
* Een pagina waarop alle temperaturen binnen de frigobox te zien zijn.
* Een knop om connectie via bluetooth met de frigobox te maken of te verbreken.
* Een knop om de volgfunctie van de frigobox in of uit te schakelen.

Naar deze functies is een schets uitgewerkt van hoe de app eruit ziet. Er zijn meer uitgewerkte prototypes gemaakt, die worden dan omgezet naar code.

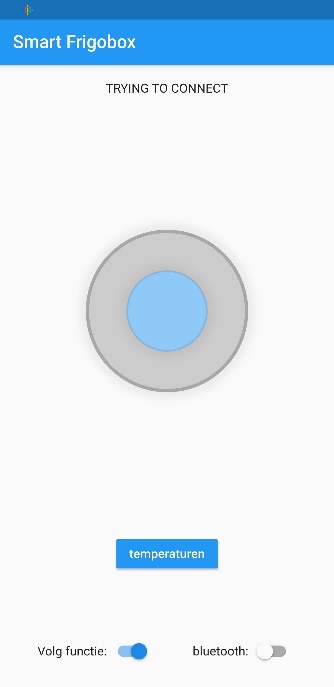
* + 1. Uiteindelijke UI

Het uiteindelijke streefdoel van de gebruikersinterface is natuurlijk dat deze gemakkelijk te gebruiken is en weinig of geen uitleg vereist voor het gebruik hiervan.

Afbeelding met schermafbeelding

Automatisch gegenereerde beschrijving

Figuur 1: Interface van de startpagina



Figuur 2: Interface van een connecterende startpagina

Figuur 3: Interface van de temperaturen

Bij de startpagina zijn de functies duidelijk vermeld aan de onderkant van het scherm (Figuur 1). Er zijn 2 aandachttrekkers te zien die de belangrijkste functies bevatten. De eerste is de joystick, die de frigobox controleert, de tweede is de knop om naar de temperaturen pagina te gaan (Figuur 3).

De joystick is uitgeschakeld door de transparantie een stuk lager te zetten (Figuur 2). De applicatie probeert automatisch te verbinden met de Arduino van de frigobox via bluetooth. Dit is bovenaan het scherm te zien (Figuur 2).

* + 1. Framework en code voor interface

Voor de interface en gehele app is gekozen om gebruik te maken van een framework genaamd Flutter. Dit framework is gekozen omdat dit een niet al te moeilijk te leren syntax heeft en heel hard lijkt op CSS en andere webcodeertalen.

Flutter is gebaseerd op Dart, werd ontwikkeld door Google en is volledig open-source waardoor er dus veel handige bibliotheken beschikbaar zijn. Er bestaan ook veel fora met handige tips en tricks. Flutter heeft ook de handige functie om geschreven code om te zetten naar Android en iOS compatibele code. Daardoor moet de code slechts éénmaal geschreven worden in Flutter en zal de compiler deze omzetten naar de juiste formaten voor de verschillende besturingsystemen. [1][2][3]

Flutter is een framework dat werkt met widgets. Deze zijn vergelijkbaar met componenten uit de webindustrie. Zo is het gemakkelijk een applicatie op te bouwen uit verschillende stukken en deze te hergebruiken op verschillende plaatsen binnen de applicatie.

* + 1. Code

Op dit moment bestaat de applicatie uit twee hoofddelen, die elk hun eigen pagina voorstellen. De thuispagina en het scherm met temperaturen dat uit 2 delen bestaat. In Flutter worden de pagina’s letterlijk genomen als elk hun eigen klasse. Verder is er een klasse nodig om de staten van objecten per pagina bij te houden. Ook is er een klasse die bijhoudt welke instellingen opgeslagen worden bij het starten van de app.

* + 1. Gebruikte bibliotheken

In deze code worden ook verschillende bibliotheken gebruikt om componenten te integreren of data te versturen. Hieronder is er een opsomming van welke bibliotheken gebruikt worden.

* Material theme: deze bibliotheek bevat componenten en kleuren die gebruikt kunnen worden om de applicatie het material theme van Google te geven [4].
* Control\_pad: een bibliotheek die een widget toont met daarin een joystick[5].
* Services library: deze bibliotheek wordt gebruikt om toegang te krijgen tot de verschillende gebruikte diensten van de telefoon zelf zoals Bluetooth en GPS [6].
* Flutter Serial Bluetooth library: deze bibliotheek verstuurt of ontvangt data via de bluetooth seriele connectie [7].
* Flutter toast: deze bibliotheek geeft feedback vanuit de applicatie naar de gebruiker weer via een toast message [8].
  1. Connectiviteit
     1. Onderzoek naar beste keuze

Vooraleer de applicatie kan gecodeerd worden, is er onderzoek verricht naar de beste technologie om te communiceren tussen de applicatie en de Arduino, die gebruikt wordt om de motoren van het onderstel te bedienen. De data moet in twee richtingen kunnen verstuurd worden, zowel van Arduino naar de applicatie en van de applicatie naar de Arduino. Dit is nodig om gegevens uit te wisselen over de locatie van de gebruiker en de temperaturen van de frigobox. Bluetooth blijkt de beste keuze omdat deze gemakkelijk te integreren is op zowel Arduino als in de applicatie. Bluetooth heeft ook een veel groter bereik dan andere alternatieven zoals wifi. Er is hoofdzakelijk niet voor wifi gekozen omdat de gebruiker dan niet zou kunnen verbinden met een ander netwerk tijdens het gebruik van de app. [9]

* 1. Joystick
     1. Onderzoek

Tijdens een eerste brainstorm kwam het idee om de automatisch volgfunctie uit te schakelen en de frigobox manueel te besturen via de app. Daarvoor is onderzoek gedaan naar een manier om dit te verwezelijken. Al snel is er ondervonden dat een bibliotheek bestaat die reeds bijna alle functies heeft die nodig zijn voor de manuele besturing van de frigobox.

* + 1. Gebruikte bibliotheek

De gebruikte bibliotheek voor deze feature is Control­ Pad van Artur Rynmarz. Deze bevat reeds een stuurknuppel die van 0 tot 360 graden kan uitlezen en een y-waarde van 0, 0.5 of 1. Deze waarde stelt voor hoe hard de binnenste cirkel verwijdert is van het middelpunt. Voor de applicatie is er nood aan een graduele verhoging. Deze is er gekomen door contact op te nemen met de auteur van de bibliotheek. Deze stemde in met de vraag voor graduele verhoging en kwam een kleine week later met de aangepaste bibliotheek waarin dit verwerkt zit. Nu kan een graduele waarde tussen nul en één uitgelezen worden. [5]

* + 1. Uitlezen en verzenden van data

Op het moment dat de gebruiker van de applicatie de joystick aanraakt, start een functie die de waarden van de joystick of control pad omzet naar verzendbare data. Eerst leest deze functie de twee waarden uit van de joystick, de richting naar waar de frigobox moet rijden en de snelheid ervan. Deze worden bepaald door de richting naar waar de gebruiker de binnenste cirkel van de joystick versleept en de afstand tot het centerpunt van de buitenste cirkel. Dit laatste wordt weergegeven als een waarde tussen nul en één. Nadat deze data is uitgelezen, wordt deze verpakt als een verzendbaar pakket om via bluetooth door te sturen naar de Arduino. Dit pakket wordt dan vertaald op de Arduino en uitgevoerd.

* 1. Temperaturen
     1. Onderzoek

Naar het weergeven van de temperaturen in de applicatie moet op zich niet veel onderzoek gebeuren aangezien deze zeer hard aansluit bij connectiviteit en het omgekeerde is van de joystick. In dit geval moet enkel het proces omgekeerd worden. Hier moet er data uitgelezen worden vanaf de Arduino en doorgestuurd worden over de bluetoothverbinding waar de applicatie deze ontvangt en omzet tot bruikbare data.

* + 1. Ontvangen van data

In het tweede scherm van de applicatie worden de temperaturen weergeven. Deze worden via sensoren uitgelezen op de Arduino en over een seriele bluetoothverbinding verzonden. De applicatie leest deze data uit en zet deze om tot een 2-dimentionale array en plaats deze in de interface waar de gebruiker deze kan bekijken.

1. Functies van de smart frigobox
   1. Temperatuurmeting
   2. Inventaris
   3. Verbinding
   4. Volgfunctie
      1. Onderzoek

Om de frigobox een smartphone te laten volgen zijn er twee voorwaarden. De eerste voorwaarde is dat de frigobox zijn positie ten opzichte van de smartphone kent. De andere voorwaarde is dat de frigobox zijn weet wat zijn richting is. Wanneer deze voorwaarden voldaan zijn kan de frigobox zijn richting aanpassen om zo de smartphone te volgen. Na onderzoek is er voor de combinatie van gps en een compas gekozen omdat dit de makkelijkste en de nauwkeurigste oplossing is. De gebruikte gps-module is een u-blox neo-6m. Deze gps is accuraat tot 2.5 meter. Een nadeel van de gps is dat deze enkel werkt als er verbinding met sattelieten is. Bijgevolg werkt de gps niet binnen. Voor het kompas wordt de Gy-273 HMC 5883L module gebruikt. Dit is een 3 assige magnetometer met een nauwkeurigheid tot 2°.[12][13]

Een ander onderdeel waar rekening mee moet gehouden worden is dat de frigobox niet botst. Dit is oplosbaar door collision detection. Na onderzoek is besloten om collision detection uit te voeren aan de hand van echolocatie. Echolocatie is een techniek om afstand te meten door een geluidsignaal te verzenden en terug op te vangen. Door de tijd tussen het verzenden en ontvangen te meten kan de afstand ontdekt worden. De module die hiervoor gebruikt wordt is een hc sr04 sensor. Deze sensor kan een afstand meten tussen 2cm en 400cm met een nauwkeurigheid van 3mm.[14]

* + 1. Richting bepalen

De richting bepalen gebeurt in verschillende stappen. De eerste stap is de hoek ten opzichte van het noorden te bepalen (codefragment 1). Dit gebeurt door het kompas en twee bibliotheken. De gebruikte bibliotheken zijn Adafruit\_Sensor en Adafruit\_HMC5883\_U. Als de graden gemeten zijn moet de magnetische declinatie worden afgetrokken. Dit is de afwijken tussen het magnetische noorden en het ware noorden. Magnetische declinatie hangt af van de plaats op aarde. Na de correctie wordt er een kleine controle uitgevoerd zodat de hoek tussen 0° en 360° is. Als dit zo is wordt de volgende stap uigevoerd, ander worden de graden aangepast.[15][16][17]

codefragment 1: De hoek tussen het kompas en het noorden bepalen

sensors\_event\_t event;

mag.getEvent(&event);

// Hold the module so that Z is pointing 'up' and you can measure the heading with x&y

// Calculate heading when the magnetometer is level, then correct for signs of axis.

float heading = atan2(event.magnetic.y, event.magnetic.x);

// Add 'Declination Angle', which is the 'Error' of the magnetic field in your location.

float declinationAngle = -0.01;

heading += declinationAngle;

// Correct for when signs are reversed.

if(heading < 0)

heading += 2\*PI;

// Check for wrap due to addition of declination.

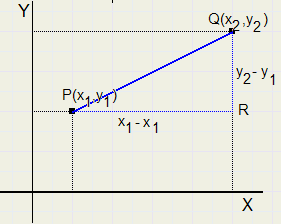
if(heading > 2\*PI)

heading -= 2\*PI;

// Convert radians to degrees for readability.

float myAngle = heading \* 180/M\_PI;

De volgende stap is de coördinaten van de frigobox te bepalen aan de hand van de gps-module. Door de bibliotheken TinyGPSPlus door Mikal Hart en SoftwareSerial van Arduino te gebruiken is het bepalen van de coördinaten makkelijker. Er bestaan namelijk functies voor om zowel de lengte- en breedtegraad. De coordinaten van de smartphone ontvangt de frigobox zoals beschreven in hoofdstuk 4.3.[18]

Wanneer beide coödinaten bekend zijn wordt door de stelling van Pythagoras (Figuur 4) de afstand tussen frigobox en smartphone berekent. Als deze afstand kleiner is dan twee meter dan stopt de frigobox met rijden. Is dit niet zo dan wordt aan de hand van de coördinaten de hoek ten opzichte van het noorden berekend. Dit gebeurd door het verschil te nemen van lengte- en breedtegraden van de coördinaten. Daarna wordt het verschil van de breedtegraden gedeeld door het verschil van de lengtegraden. Van deze breuk wordt de boogtangens genomen en dit resultaat geeft de gezochte hoek in radialen.[19]

Figuur 4: stelling van Pythagoras bij twee punten[20]

De frigobox heeft nu twee hoeken ten opzichte van het noorden. De eerste hoek is de richting van de frigobox zelf. De tweede hoek is de richting naar waar de frigobox moet rijden, ook de volghoek genoemd. Wanneer de twee hoeken hetzelfde zijn betekent dat de frigobox richting de smartphone rijdt en dus niet moet draaien. In dit geval rijdt de frigobox rechtdoor als de richting van de frigobox binnen een marge van 20° is ten opzichte van de volghoek. Als de richtingshoek van de frigobox groter is dan de volghoek en het verschil tussen de richtingshoek en volghoek is kleiner dan 180° dan draait de frigobox naar links. In een ander geval draait de frigobox naar rechts.

* + 1. Collision detection

Voor de frigobox echt rijdt wordt er aan collision detection gedaan zodat de frigobox nergens tegen botst. Als de frigobox rechtdoor moet rijden meet de voorste hc sr04 sensor (Figuur 5) de afstand tot een voorwerp vooraan (codefragment 2). Als deze afstand kleiner dan 20cm is dan rijdt de frigobox niet rechtdoor maar doet deze een controle met de linkse sensor. Als hier geen obstakel staat draait de sensor naar links. Als er wel een obstakel staat dan gebeurt het process nog eens maar nu rechts. Als hier ook een obstakel staat dan zal de frigobox stoppen. Deze maatregel is er om een oneindige loop tegen te gaan. Wanneer de frigobox naar links of rechts moet draaien meten de respectievelijke hc sr04 sensoren de afstand tot het dichtstbijzijnde obstakel. Als de afstand kleiner dan 20cm is dan zal de frigobox niet draaien maar rechtdoor rijden.

Figuur 5: ultrasone HC SR04 sensor

De manier waarop de sensoren de afstand meten met is a.d.h.v. echolocatie. De ultrasone sensor stuurt een geluid van 40kHz uit dat zich door de lucht beweegt. Als het geluid dat uitgestuurt is tegen een object botst, keert het terug en kan het opgevangen worden door de sensor. Als de tijd van wanneer een geluidssginaal is uitgezonden en wanneer een geluidssignaal wordt opgevangen vergeleken wordt kan hieruit de afstand berekend worden. Dit doen we door de tijd die we bekomen te delen door 2 aangezien het geluid zowel heen als terug moet en dan te vermenigvuldigen met 0.034. Dit laatste getal stelt de snelheid van het geluid voor. [1]

boolean checkCollision(int trigPin, int echoPin)

{

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance = duration\*0.034/2;

if(distance < 20){

return false;

}

else{

return true;

}

}

codefragment 2: collision detection

* 1. Smartlock

1. Kostprijsberekening

De geschatte prijs die tegen het einde van het project uitgegeven zal worden is maximaal €150.

Conclusie

**Geen indexgegevens gevonden.**

Literatuurlijst

|  |  |
| --- | --- |
| [1] | A. A. Jabbaar, "arduino project hub," 17 September 2019. [Online]. Available: https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6. [Accessed 26 April 2020]. |
| [2] | Flutter, „Flutter,” 12 Januari 2020. [Online]. Available: https://flutter.dev. [Geopend 12 Januari 2020]. |
| [3] | Flutter, „Flutter on Github,” Github, 12 Januari 2020. [Online]. Available: https://github.com/flutter. [Geopend 12 Januari 2020]. |
| [4] | Dart, „Dart,” Dart, 14 Januari 2020. [Online]. Available: https://dart.dev/. [Geopend 14 Januari 2020]. |
| [5] | Flutter, „Theme class,” Flutter, 20 Januari 2020. [Online]. Available: https://api.flutter.dev/flutter/material/Theme-class.html. [Geopend 20 Januari 2020]. |
| [6] | artrmz, „flutter\_control\_paf,” 1 Januari 2020. [Online]. Available: https://pub.dev/packages/control\_pad. [Geopend 23 Januari 2020]. |
| [7] | baseflow.com, „location\_permissions 2.0.5,” baseflow.com, 10 Maart 2020. [Online]. Available: https://pub.dev./packages/location\_permissions. [Geopend 12 Maart 2020]. |
| [8] | e. en p. , „flutter\_bluetooth\_serial 0.2.2,” 19 Augustus 2019. [Online]. Available: https://pub.dev/packages/flutter\_bluetooth\_serial. [Geopend 3 Maart 2020]. |
| [9] | huclengyue@gmail.com, „toast 0.1.5,” 16 Juli 2019. [Online]. Available: https://pub.dev/packages/toast. [Geopend 27 Februari 2020]. |
| [10] | M. Skawiński, „Bluetooth vs WiFi Comparison For the IoT Solutions,” Netguru, 19 Januari 2019. [Online]. Available: https://www.netguru.com/codestories/bluetooth-vs-wifi-comparison-for-the-iot-solutions. [Geopend 14 Februari 2020]. |
| [11] | M. Hein, „Gebruikersomgeving,” Wikipedia, 26 September 2018. [Online]. Available: https://nl.wikipedia.org/wiki/Gebruikersomgeving. [Geopend 22 Maart 2020]. |
| [12] | Verdel, „Gebruikerservaring,” Wikipedia, 3 april 2019. [Online]. Available: https://nl.wikipedia.org/wiki/Gebruikerservaring. [Geopend 22 Maart 2020]. |
| [13] | Robu.in, „U-blox NEO-6M GPS Module with EPROM,” 13 April 2020. [Online]. Available: https://robu.in/product/ublox-neo-6m-gps-module/. [Geopend 13 April 2020]. |
| [14] | Ben's electronics, „Gy-273 HMC 5883L 3 assige magnetometer (kompas sensor),” 16 April 2020. [Online]. Available: https://benselectronics.nl/gy-273-hmc-5883l-3-assige-magnetometer-kompas-sensor-/. [Geopend 16 April 2020]. |
| [15] | tinytronics, „Ultrasonische Sensor - HC-SR04,” 16 April 2020. [Online]. Available: https://www.tinytronics.nl/shop/nl/sensoren/afstand/ultrasonische-sensor-hc-sr04. [Geopend 16 April 2020]. |
| [16] | ladyada, „Adafruit\_Sensor,” 6 April 2020. [Online]. Available: https://github.com/adafruit/Adafruit\_Sensor. [Geopend 17 April 2020]. |
| [17] | tdicola, „Adafruit\_HMC5883\_Unified,” 27 Mei 2016. [Online]. Available: https://github.com/adafruit/Adafruit\_HMC5883\_Unified. [Geopend 17 April 2020]. |
| [18] | InternetArchiveBot, „Magnetic declination,” 22 Maart 2020. [Online]. Available: https://en.wikipedia.org/wiki/Magnetic\_declination. [Geopend 17 April 2020]. |
| [19] | M. Hart, „TinyGPSPlus,” 13 Oktober 2019. [Online]. Available: https://github.com/mikalhart/TinyGPSPlus. [Geopend 17 April 2020]. |
| [20] | Wikiwerner, „Stelling van Pythagoras,” 18 November 2019. [Online]. Available: https://nl.wikipedia.org/wiki/Stelling\_van\_Pythagoras. [Geopend 18 April 2020]. |
| [21] | „Cursus Algebra (3),” 26 April 2020. [Online]. Available: http://www.davdata.nl/algebra/alg3.html. [Geopend 26 April 2020]. |

Bijlagenoverzicht

Bijlagen

Logboek:

|  |  |  |
| --- | --- | --- |
| **Naam** | **Datum** | **Wat** |
| Dries, Oberon, Aaron, Elian | 13/02/2020 | Stand van zaken bekijken en nieuwe taken verdelen. |
| Oberon | 10/02/2020 | Aanmaken document |
| Oberon | 8/03/2020 | Document verbeteren |
| Oberon, Elian, Aaron | 15/03/2020 | Volledige verbetering na feedback + begin schrijven hoofdstuk |
| Elian | 16/03/2020 | Volledig ruwe versie schrijven van eerste hoofdstuk. |
| Oberon | 19/03/2020 | Figurenlijst automatisch laten genereren en gehele document nalezen. |
| Dries | 20/03/2020 | Verslag omgezet naar tegenwoordige tijd. Nagelezen op herhalingen en dt fouten. |
| Aaron | 21/03/2020 | Vergeten zinnen omgezet naar tegenwoordige tijd. Verbeteringen aan zinsbouw en spelling. |
| Elian | 23/03/2020 | Toevoegen van bronnen |
| Aaron | 23/03/2020 | Nakijken bronnen + toevoegen begrippen |
| Oberon | 29/03/2020 | Eerste verbetering van alle notities. |
| Aaron | 30/03/2020 | Verdere verbetering |
| Dries | 30/03/2020 | Nog eens overlezen. |
| Elian | 30/03/2020 | Laatste aanpassingen en eindcontrole |
| Aaron | 23/04/2020 | 4.4 geschreven |
| Aaron | 26/04/2020 | Toevoeging bronnen, figuren en codefragmenten + layout aanpassingen |