­

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**

**Realising a self driving smart coolbox using Arduino**

D. Bruggeman, O. Lievens, A. Lison, E. Van Cutsem

The purpose of this project is to reinvent the idea of a classic coolbox for less then €100,00. This is done by adding some smart functions to a coolbox and making it self-driving. The smart functions used are: a follow function, temperature measurement and an inventory. These are controlled by a Bluetooth connection between an app and an Arduino.

Firstly the possibilities of connecting an Arduino to control the coolbox and electronics to a mobile application are researched. The conclusion of this study directs to a Bluetooth connection over which all needed data is transferred by a serial Bluetooth connection. The Arduino is used to control the electronics in the coolbox and the frame. The motors in the frame are controlled by a function which compares the coordinates of the coolbox to the application and controls the motors to drive in the needed direction. The frame is also capable of watching the sides. when the frame comes to close to a wall, the Arduino reconfigures the motors to avoid a collision. The Arduino is constantly reading the temperatures inside the coolbox, so the mobile application is capable of showing the individual or average temperature in the coolbox. The application is constructed using a native mobile framework developed by Google called Flutter. The application uses libraries to control the Bluetooth-state and manage Bluetooth connections. The application consists of three main screens which include navigation, statistics and settings.

Concluding can be stated that a prototype of the self-driving coolbox controlled by a mobile application and an Arduino is completely functional. The total cost of this project was €00,00 which is less than the maximum amount set for this project. The project still can be extended by new smart-functions controlled by an Arduino.

Keywords: Internet of Things, Arduino, Electronics, coolbox, self-driving, App, Flutter, smart-technology, GPS, echolocation, Bluetooth

**Voorwoord**

In het tweede jaar van de opleiding Professionele Bachelor in Elektronica / ICT aan hogeschool Odisee is het maken van een verslag een deelopdracht voor het opleidingsonderdeel Project. Hier moeten de studenten zich verdelen in teams van vier en zelf een projectvoorstel en projectbegeleider voorstellen.

Wij hebben ervoor gekozen om een eigen voorstel aan te brengen omdat het een grotere motivator is en we onze eigen creativiteit wouden laten zien. In het begin wisten we al dat we iets met eten of drinken zouden doen aangezien we dit alle vier leuk vinden. Onze creativiteit heeft ons bij verschillende ideeën gebracht waarvan deze de meest realtische was om te kunnen volbrengen.

Een frigobox die vol zit kan veel wegen waardoor wij met het idee kwamen om de frigobox zelf te laten rijden. Niet alleen kan de frigobox zelf rijden maar heeft hij ook verschillende andere functies die het gebruik ervan verbeteren. Hierdoor wordt de frigobox ergonomischer en makkelijker in gebruik.

In ons project hebben we de kennis van verschillende vakken bijeengebracht en veel nieuwe kennis opgedaan. We hebben vaardigheden gebruikt die wij normaal gezien niet veel nodig zouden hebben waardoor we onze horizon verbreedt hebben. Niet alleen het project op zich was een hele uitdaging maar het Coronavirus heeft er een extra moeilijkheidgraad bijgevoegd. We hebben nog meer geleerd hoe belangrijk de communicatie binnen een team is en hoe we op een positieve manier een goed eindresultaat kunnen bekomen.

Graag willen we Mario Wyns en Sabine Martens

Inhoudsopgave

[Inhoudsopgave 5](#_Toc40950801)

[Figurenlijst 8](#_Toc40950802)

[Codefragmentenlijst 10](#_Toc40950803)

[Begrippenlijst 12](#_Toc40950804)

[Inleiding 13](#_Toc40950805)

[1 Opbouw 14](#_Toc40950806)

[1.1 Bouw van de frigobox 14](#_Toc40950807)

[1.2 Bouw van de frame 14](#_Toc40950808)

[2 Elektronica 14](#_Toc40950809)

[2.1 Elektonica van de frigobox 14](#_Toc40950810)

[2.2 Elektronica van de frame 14](#_Toc40950811)

[3 App 15](#_Toc40950812)

[3.1 Interface 15](#_Toc40950813)

[3.1.1 Onderzoek naar UI & UX 15](#_Toc40950814)

[3.1.2 Uiteindelijke UI 16](#_Toc40950815)

[3.1.3 Framework en code voor interface 17](#_Toc40950816)

[3.1.4 Code 18](#_Toc40950817)

[3.1.5 Gebruikte bibliotheken 18](#_Toc40950818)

[3.2 Connectiviteit 18](#_Toc40950819)

[3.2.1 Onderzoek naar beste keuze 18](#_Toc40950820)

[3.3 Joystick 19](#_Toc40950821)

[3.3.1 Onderzoek 19](#_Toc40950822)

[3.3.2 Gebruikte bibliotheek 19](#_Toc40950823)

[3.3.3 Uitlezen en verzenden van data 19](#_Toc40950824)

[3.4 Temperaturen 19](#_Toc40950825)

[3.4.1 Onderzoek 19](#_Toc40950826)

[3.4.2 Ontvangen van data 19](#_Toc40950827)

[3.5 Locatie 19](#_Toc40950828)

[4 Functies van de smart frigobox 20](#_Toc40950829)

[4.1 Verbinding 20](#_Toc40950830)

[4.1.1 Vooronderzoek 20](#_Toc40950831)

[4.1.2 Realisatie 20](#_Toc40950832)

[4.1.3 Ontvangen en verzenden van data 20](#_Toc40950833)

[4.2 Temperatuurmeting 20](#_Toc40950834)

[4.2.1 Onderzoek 20](#_Toc40950835)

[4.2.2 Implementatie 21](#_Toc40950836)

[4.3 Inventaris 21](#_Toc40950837)

[4.3.1 Onderzoek 21](#_Toc40950838)

[4.3.2 Implementatie 21](#_Toc40950839)

[4.4 Volgfunctie 21](#_Toc40950840)

[4.4.1 Onderzoek 21](#_Toc40950841)

[4.4.2 Richting bepalen 22](#_Toc40950842)

[4.4.3 Botsingsdetectie 23](#_Toc40950843)

[4.5 Smartlock 24](#_Toc40950844)

[5 Kostprijsberekening 24](#_Toc40950845)

[Conclusie 25](#_Toc40950846)

[Literatuurlijst 26](#_Toc40950847)

[Bijlagenoverzicht 29](#_Toc40950848)

[Bijlagen 30](#_Toc40950849)

Figurenlijst

Figuur 1: overzicht van het prototype uitgewerkt in Figma 15

Figuur 2: Startscherm van het prototype 15

Figuur 3: ruwe versie voor het verbinden met een al geconnecteerd toestel 16

Figuur 4: ruwe versie voor het verbinden met een nieuw Bluetooth toestel 16

Figuur 5: ruwe versie van het cherm om de frigobox te besturen 16

Figuur 6: ruwe versie van startscherm van de applicatie met alle funties 16

Figuur 7: Het uiteindelijke ontwerp van het instellingen-scherm in de applicatie 17

Figuur 8: Het uiteindelijke ontwerp van het temperatuur-scherm in de applicatie 17

Figuur 9: Het uiteindelijke ontwerp van het startscherm in de applicatie 17

Figuur 10: SparkFun BlueSMiRFmodule [27] 20

Figuur 11: 470ohm NTC-thermistor [15] 20

Figuur 12: stelling van Pythagoras bij twee punten [26] 22

Figuur 13: ultrasone HC SR04 sensor [25] 23

Codefragmentenlijst

[codefragment 1: De hoek tussen het kompas en het noorden bepalen 23](file:///C:\Users\obero\Downloads\2hoofdstukICTEO1.docx#_Toc39507213)

[codefragment 2: Botsingsdetectie 25](file:///C:\Users\obero\Downloads\2hoofdstukICTEO1.docx#_Toc39507214)

Begrippenlijst

UI User Interface is een gebruikersomgeving of gebruikersinterface maakt interactie tussen mens en software mogelijk. [1]

UX User Experience of gebruikerservaring is een term die alle aspecten beschrijft van de ervaring van een gebruiker bij interactie met een product. [2]

ADC Analog to Digital Converter. Dit berekend een digitale waarde uit een analoge spanning.

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
      1. Onderzoek

Eerst wordt er onderzocht hoe een frigobox gerealiseerd kan worden. Deze frigobox is net als de klassieke frigobox. Het doel hiervan is om eten of drinken fris te houden. Hier komen twee mogelijkheden tevoorschijn. Mogelijkheid één is om een kant en klare frigobox te kopen en deze te verbouwen. Het voordeel hiervan is dat er niet veel werk nodig is om de frigobox om te bouwen voor het einddoel. Het nadeel is dat een frigobox redelijk duur is en dat deze gaten bevat voor de elektronica in te krijgen. Een andere mogelijkheid is om zelf een frigobox te bouwen. Hier kruipt meer tijd in, maar de materialen zijn goedkoper en de electronica kan makkelijk geïnstalleerd worden. Om deze redenen is mogelijkheid twee de keuze geworden.

De nodige materialen voor deze frigobox zijn mdf-plaaten, isomo, twee scharnieren en een aantal vijzen. Deze materialen zijn goedkoop en goed isolerend waardoor de binnenkant van de frigobox steeds koel blijft.

* + 1. Bouw

De opbouw van de frigobox is niet voltooid door de coronacrisis. Hieronder is de theoretische opbouw beschreven.

De frigobox is opgebouwd uit drie verschillende lagen. De buitenste laag is een box, opgebouwd uit mdf-platen. De afmetingen van de box zijn 40 cm in de breedte, 60 cm in de lengte en 40 cm in de hoogte. De platen zijn vast gemaakt door middel van vijzen, behalve de bovenste plaat. De bovenste plaat hangt vast door twee scharnieren. Dit zorgt ervoor dat de frigobox makkelijk open en dicht kan.

De middenste laag is de isolerende laag. Deze laag bestaat uit 5 cm isomo aan elke zijde. Door de dikke laag isomo blijft de binnenkant van de box koel.

De laatste en binnenste laag bestaat terug uit MDF-platen. Deze laag dient als afwerking en om de isomo laag te beschermen.

Door alle lagen zijn kleine gaatjes geboord om de kabels van de elektronica van binnen de frigobox naar buiten te krijgen.

* 1. Bouw van de frame

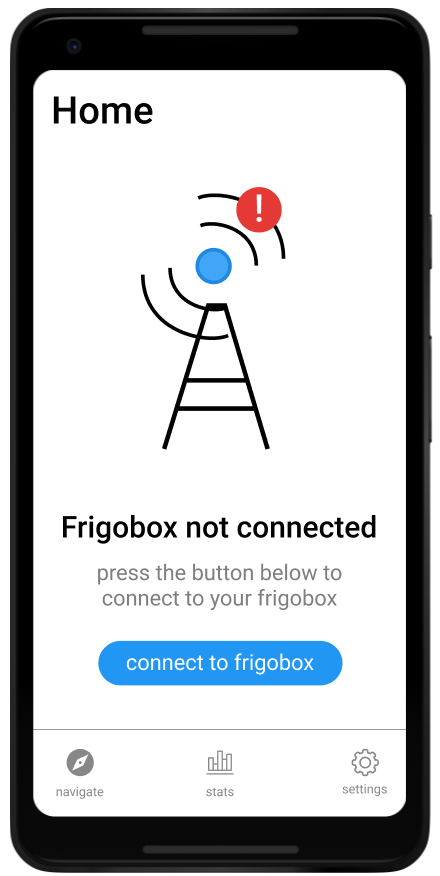
1. Elektronica
   1. Elektronica van de frigobox
   2. Elektronica van de frame
2. App
   1. Interface
      1. Onderzoek naar UI & UX

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 knop om te verbinden met een nieuw Bluetooth toestel.
* Een knop om te verbinden met een verbonden Bluetooth toestel.
* 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.

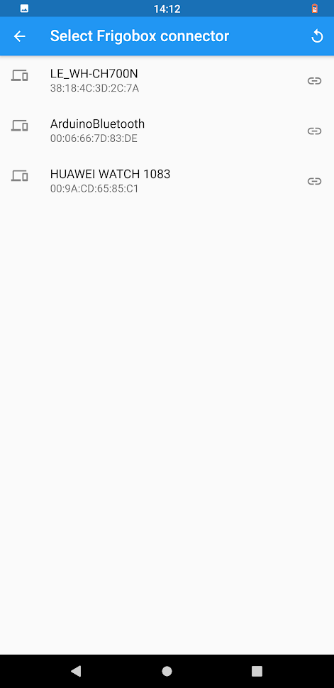
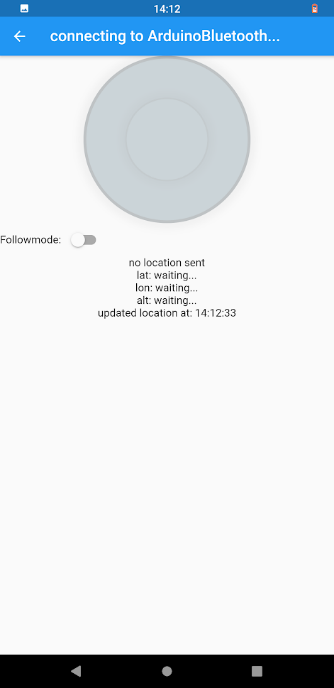


Figuur 1: overzicht van het prototype uitgewerkt in Figma

Figuur 2: Startscherm van het prototype

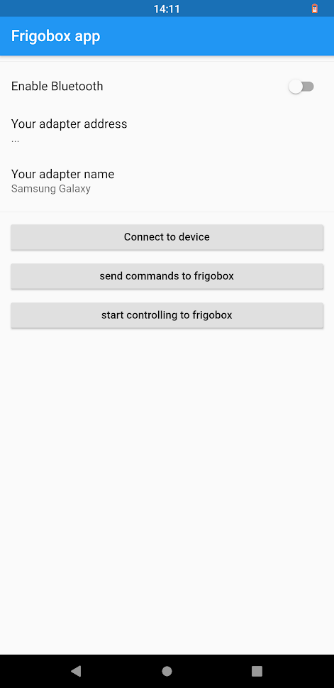
* + 1. Uiteindelijke UI

Het uiteindelijke streefdoel van de gebruikersinterface is natuurlijk dat deze gemakkelijk te gebruiken is en weinig of geen uitleg vereist voor het gebruik hiervan. Eerst is er een heel simpele versie van de applicatie ontwikkeld waarin alle functies in de basis van de applicatie aanwezig zijn. Deze versie is gebruikt om alle technologiën te testen en uit te werken alvorens aandacht te besteden aan het uiteindelijke ontwerp van de applicatie. Aangezien er met Bluetooth gewerkt wordt, is er vertrokken van de voorbeeld-applicatie van de gebruikte Bluetooth bibliotheek die gratis te gebruiken is.



Figuur 3: ruwe versie voor het verbinden met een al geconnecteerd toestel

Figuur 4: ruwe versie voor het verbinden met een nieuw Bluetooth toestel



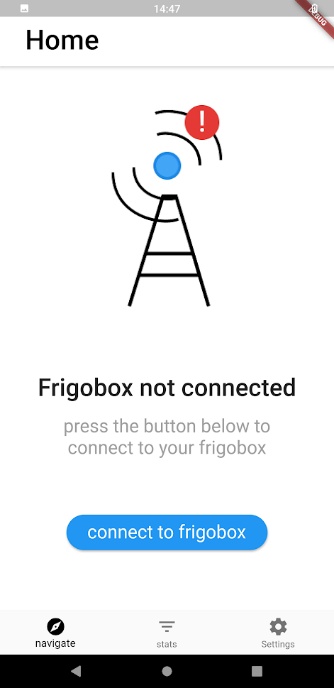
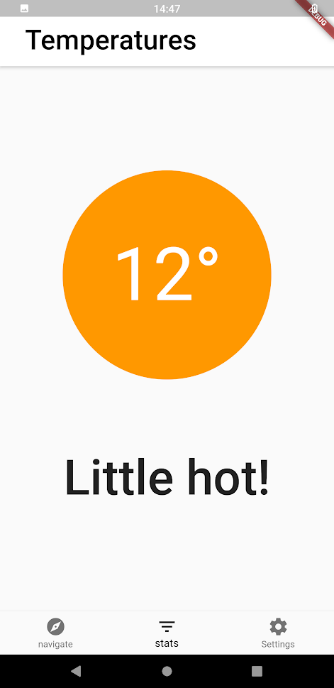
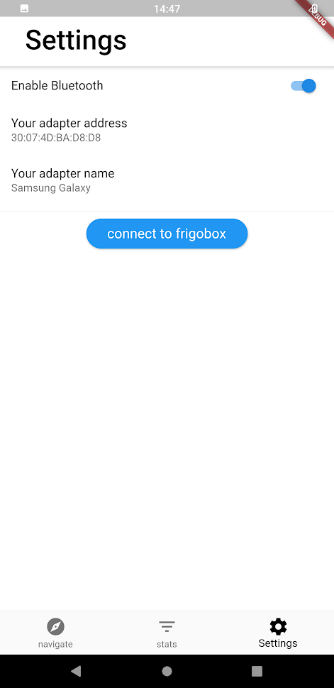
Figuur 5: ruwe versie van het cherm om de frigobox te besturen

Figuur 6: ruwe versie van startscherm van de applicatie met alle funties

Bij de startpagina zijn bovenaan de Bluetooth-instellingen van het toestel te zien, daar net onder zijn er 3 knoppen, deze stellen de hoofdfuncties van de applicatie voor. De functies staan duidelijk vermeld op de knoppen op het scherm (Figuur 6). Er zijn 3 aandachttrekkers te zien die de belangrijkste functies bevatten. De eerste is de functie om te verbinden met een nieuw Bluetooth toestel (Figuur 4), de tweede is een functie om zelf strings door te sturen naar de seriele verbinding van een Bluetooth-toestel (Figuur 3), de laatste is een functie om met een geconnecteerd Bluetooth-toestel te verbinden en het controle-scherm te tonen (Figuur 5).

Er is ook meteen rekening gehouden met de volg-funcite. De joystick wordt uitgeschakeld als de volg-functie ingeschakeld is. Dit wordt weergeven door de transparantie van de joystick een stuk lager te zetten. De applicatie probeert automatisch te verbinden met de Arduino van de frigobox via bluetooth. Dit is bovenaan het scherm te zien (Figuur 5).

Nadat al deze functies functioneel werken, is het ontwikkelingsproces van de uiteindelijke UI gestart.

In bovenstaande figuren is het uiteindelijk ontwerp van de applicatie naar het prototype-ontwerp te zien. Op het startscherm (Figuur 9) is te zien dat de applicatie niet verbonden is met een toestel. Als er op de blauwe knop gedrukt wordt, komt er een een scherm tevoorschijn waarmee de gebruiker een toestel kan selecteren uit een lijst met reeds verbonden Bluetooth-toestellen (Figuur 3). Eens geconnecteerd wordt de gebruiker naar een scherm gestuurd waar de gebruiker het verbonden toestel kan besturen met een Joystick. Aan de onderkant van elk scherm is een menu te zien waarmee genavigeerd kan worden naar de drie hoofdschermen: het bestuur-scherm (Figuur 9), statistieken-scherm (Figuur 8) en het instellingen-scherm (Figuur 7). De laatste heeft exact dezelfde info als te zien in Figuur 6. Deze info is enkel wat herwerkt om makkelijker bruikbaar te zijn. Het statistieken-scherm heeft nog 2-subschermen die bereikt kunnen worden door naar links of rechts te vegen. Links bevindt zich het Details scherm waar eventuele details van de Arduino in de frigobox weergeven kunnen worden. Rechts bevind zich een waterniveau-scherm dat niet afgewerkt is wegens de corona-crisis. Op deze pagina kan weergegeven worden of er water of vloeistof gelekt is in de frigobox.

Figuur 7: Het uiteindelijke ontwerp van het instellingen-scherm in de applicatie

Figuur 8: Het uiteindelijke ontwerp van het temperatuur-scherm in de applicatie

Figuur 9: Het uiteindelijke ontwerp van het startscherm in de applicatie

* + 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. [3] [4] [5]

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

De code van de applicatie is verdeeld is verchillende klassen. De eerste klasse die gebruikt wordt is de main.dart klasse. Deze klasse bevat alle code van het startscherm van de applicatie. Verder is er voor elk scherm een klasse. Er zijn ook klassen voor elk stuk hergebruikte code zoals het menu en de hoofding, deze komen voor op elke pagina gevarieerd voor.

Het navigatie-menu gebruikt de voorgeprogammeerde widget BottomNavigationBar [6]. Deze is aangepast om de juiste iconen en tesksten te tonen. De hoofding is ook gebaseerd op een voorgeprogammeerd widget: AppBar [7]. Deze is licht aangepast naar het prototype ontwerp.

* + 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. [7]
* Control\_pad: een bibliotheek die een widget toont met daarin een joystick. [8]
* Services library: deze bibliotheek wordt gebruikt om toegang te krijgen tot de verschillende gebruikte diensten van de telefoon zelf zoals Bluetooth en GPS. [9]
* Flutter Serial Bluetooth library: deze bibliotheek verstuurt of ontvangt data via de bluetooth seriele connectie. [10]
* Flutter toast: deze bibliotheek geeft feedback vanuit de applicatie naar de gebruiker weer via een toast message. [11]
* Background location libray: deze bibliotheek verzameld GPS gegevens in de achtergrond. [12]
  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. [13]

* 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. [7]

* + 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. Locatie

Voor de locatie word gebruik gemaakt van de interne GPS van het toestel waar de applicatie op draait. Hiervoor is echter toestemming van de gebruiker verreist. Deze wordt aangevraagd bij de eerste opening van de applicatie. Als deze verleend wordt, zal de applicatie naar behoren werken. Als de permissie-aanvraag geweigerd wordt, zal de applicatie niet goed werken en zal de applicatie bij elke opening opnieuw vragen om toestemming.

De applicatie zal elke 2 seconden een update vragen aan de GPS van het toestel. Deze update bevat de longitude, latitude, hoogte en snelheid van de GPS in het toestel. Als er een update beschikbaar is, zal deze doorgegeven worden aan de gebruikte bibliotheek [13]. Deze wordt dan doorgestuurd over de seriele Bluetooth-connectie naar de Arduino.

1. Functies van de smart frigobox
   1. Verbinding
      1. Vooronderzoek

Om een verbinding mogelijk te maken tussen de smartphone en de Arduino wordt gebruik gemaakt van bluetoothtechnologie. Deze technologie is gekozen wegens de gemakkelijke integratie met Arduino en smartphones, zoals vermeld in hoofdstuk 3.2. De gebruikte manier om dit te realiseren binnen dit project is om een bluetoothmodule aan te sluiten op de Arduino. Daarmee kan dan via de mobiele applicatie binnenkomende data gebruikt worden om bijvoorbeeld de frigobox naar de smartphone te doen navigeren of de sensors van de Arduino uit te lezen op de app.

* + 1. Realisatie

Figuur 10: SparkFun BlueSMiRFmodule [27]

Om de verbinding te realiseren wordt gebruik gemaakt van de SparkFun BlueSMiRFmodule (Figuur 4). Deze module beschikt over een bluetoothzendontvanger waardoor deze in staat is om zowel data te versturen als te ontvangen. De module beschikt ook over een serieele aansluiting, waarmee bluetoothzendingen kunnen vertaald worden naar tekst. Dit zorgt voor een gemakkelijke integratie in dit project.

* + 1. Ontvangen en verzenden van data

De communicatie tussen de bluetoothmodule en de mobiele applicatie verloopt in dit project helemaal via een seriele connectie. Daarvoor wordt er gebruik gemaakt van een bibliotheek SoftwareSerial. De Arduino beschikt zelf over een seriele connectie, maar die wordt al gebruikt als Seriele Monitor voor de Arduino. Een Arduino Mega beschikt wel over meerdere seriele connecties, maar een Arduino Uno niet; het is dus best om een seriele connectie te maken met deze bibliotheek, zodat het programma ook op een Arduino Uno kan draaien.

De SoftwareSerialbibliotheek is ontworpen om een nieuwe softwarematige seriele connectie te maken. Het is ook mogelijk om op deze seriele poorten verschillende snelheden in te stellen. Deze bibliotheek werkt enkel op digitale poorten.

* 1. Temperatuurmeting
     1. Onderzoek

De smart frigobox kan de temperatuur binnen de frigobox weergeven. Deze functie gaat hand in hand met de inventatisfunctie van de frigobox. Om temperatuur uit te meten wordt gebruik gemaakt van temperatuursgevoelige weerstanden, ook wel thermistors genoemd. Deze weerstand die deze thermistors bieden, is veranderlijk in fucntie van de temperatuur. Er bestaan twee soorten thermistors namelijk NTC en PTC. Bij de NTC-veriant daalt de weerstand wanneer de temperatuur stijgt. Bij de PTC-variant stijgt de weerstand wanneer de temperatuur stijgt. [12]

Om de temperatuur te meten wordt met de thermistor een spanningsdeler gemaakt. Er komt een spanning van 5 V over de spanningsdeler te staan. Wanneer een weerstand met dezelfde weerstandswaarde als de nominale weerstandswaarde van de thermistor gebruikt wordt, dan is de spanning aan de uitgang van de spanningsdeler bij nominale temperatuur 2,5 V. Dit wil zeggen dat bij een NTC-thermistor, wanneer de temperatuur daalt, de spanning aan de uitgang van de spanningsdeler ook zal dalen.

Figuur 11: 470ohm NTC-thermistor [15]

Deze spanning wordt ingelezen door de ADC van de Arduino. Deze zet de spanning om in een 10 bit binaire waarde of een decimale waarde tussen 0 en 1023. [13] Elke weerstandcombinatie moet gecalibreerd worden. Dit kan door de weerstand in koud water te leggen met een gekende temperatuur en die te noteren samen met de decimale waarde. Deze stap wordt herhaald met warm water en dan wordt een mapping gemaakt tussen deze twee waarden en kunnen we temperatuur inter- en extrapoleren. [14]

* + 1. Implementatie

In dit project worden tien thermistoren gebruikt met een weerstand van 470 Ω met een tolerantie van 10%. De nominale temperatuur ervan is plus minus 25°C, dit wil zeggen dat wanneer de temperatuur van de thermistor 25°C is, de weerstand 470 Ω zal zijn. [15]

In de twee coldpackhouders wordt telkens één thermistor bevestigd. Daarnaast wordt er in elke houder voor een blikje één thermistor geplaatst. In totaal zijn er acht houders beschikbaar. Zo kan elke component zijn temperatuur uitgemeten worden.

* 1. Inventaris
     1. Onderzoek

Om aan inventarisatie te doen, is er nood aan een sensor die kan detecteren of er een voorwerp op een plek aanwezig is. Door de bouw van de frigobox kunnen enkel blikjes gebruikt worden. Blikjes zijn gemaakt van aluminium, dit is een geleidend materiaal. Wanneer een blikje contact heeft met twee geleidende oppervlakken, zou er een kortsluiting kunnen ontstaan. Dit is hetzelfde als het sluiten van een shakelaar. Hierdoor kan een Arduino detecteren dat er een blikje is. Na verschillende testen blijkt dat dit heel inconsequent werkt en niet gebruikt kan worden. Dit kan wel vervangen worden door een soort van druksensor. Deze kan makkelijk zelf gemaakt worden. [16]

* + 1. Implementatie

Zo een druksensor kan makkelijk op dezelfde manier geïmplementeerd worden als een gewone drukknop. Een draad wordt aangesloten aan een digitale poort op de Arduino en door gebruik van een pullupweerstand dus ook aan 5 V. De andere draad wordt aangesloten aan GND. Dit wil zeggen dat wanneer een blikje in een slot geplaatst wordt, de druksensor intern kortsluit. Wanneer de code van de Arduino de inventarisatiestatus opvraagt, ziet de Arduino dat er in een bepaalde houder een blikje geplaatst is.

* 1. 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 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 kompas 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°. [17] [18]

Een ander onderdeel waar rekening mee gehouden wordt, is dat de frigobox niet botst. Dit is oplosbaar door botsingsdetectie. Na onderzoek is besloten om botsingsdetectie 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 berekend worden. De module die hiervoor gebruikt wordt, is een hc sr04-sensor. Deze sensor kan een afstand meten tussen 2 cm en 400 cm met een nauwkeurigheid van 3 mm. [19]

* + 1. Richting bepalen

De richting bepalen gebeurt in verschillende stappen. De eerste stap is de hoek ten opzichte van het noorden bepalen (codefragment 1). Dit gebeurt door het gebruik van het kompas en twee bibliotheken. De gebruikte bibliotheken zijn Adafruit\_Sensor en Adafruit\_HMC5883\_U. Als de graden gemeten zijn, wordt de magnetische declinatie afgetrokken. Dit is de afwijking 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 of de hoek tussen 0° en 360° is. Als dit zo is, wordt de volgende stap uitgevoerd, anders worden de graden aangepast. [20] [21] [22]

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.

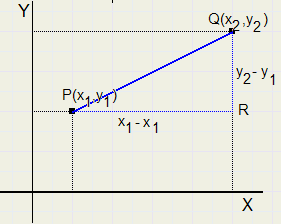
if(heading > 2\*PI)

heading -= 2\*PI;

// Convert radians to degrees for readability.

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

De volgende stap is de coördinaten van de frigobox 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 om zowel de lengte- en breedtegraad te verkrijgen. De frigobox ontvangt de coördinaten van de smartphone zoals beschreven in hoofdstuk 4.1.3. [23]

Wanneer beide coödinaten bekend zijn, wordt door de stelling van Pythagoras (Figuur 4) de afstand tussen de frigobox en de smartphone berekent. Als deze afstand kleiner is dan twee meter, dan stopt de frigobox met rijden. Is de afstand groter dan twee meter, dan wordt de hoek ten opzichte van het noorden aan de hand van de coördinaten berekend. Dit gebeurt door het verschil te nemen van de 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. [24]

Figuur 12: stelling van Pythagoras bij twee punten [26]

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 dat de frigobox richting de smartphone rijdt en dus niet moet draaien. In praktijk rijdt de frigobox rechtdoor als de richtingshoek van de frigobox binnen een marge van 20° ligt ten opzichte van de volghoek. Als de richtingshoek van de frigobox groter is dan de volghoek en het verschil tussen de richtingshoek en volghoek kleiner is dan 180° dan draait de frigobox naar links. In het andere geval draait de frigobox naar rechts.

* + 1. Botsingsdetectie

Voor de frigobox echt rijdt, wordt er aan botsingsdetectie gedaan zodat de frigobox nergens tegen botst. Hc sr04 sensoren (Figuur 7) meten afstand aan de hand van echolocatie. Deze sensoren zijn ultrasone sensoren. Dit wil zeggen dat ze geluidsgolven kunnen verzenden en opvangen. De ultrasone sensor stuurt een geluid van 40kHz uit dat zich door de lucht beweegt. Als het geluid dat uitgestuurt wordt, tegen een object botst, keert het terug en wordt het opgevangen door de sensor. De afstand tot dit object wordt berekend door te meten hoelang het duurt voor het uitgestuurde geluidsignaal terug opgevangen wordt door de sensor. De gemeten tijd wordt door twee gedeeld, aangezien het geluid zowel heen als weer gaat. Tot slot wordt het bekomen resultaat vermenigvuldigd met 0.034, dit stelt de snelheid van het geluid voor. Na deze vermingvuldiging is de afstand tussen de sensor en het object bekent. [25]

Figuur 13: ultrasone HC SR04 sensor [25]

Als de frigobox rechtdoor moet rijden meet de hc sr04 sensor (Figuur 7) vooraan de frigobox de afstand tot een voorwerp (codefragment 2). Als deze afstand kleiner is dan 20cm, rijdt de frigobox niet rechtdoor, maar doet deze een controle met de linkse sensor. Als hier geen obstakel staat, draait de frigobox naar links. Als er wel een obstakel staat, dan gebeurt het proces opnieuw maar dan 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 signaal krijgt van de gps om naar links of naar rechts te draaien, meten de respectievelijke hc sr04 sensoren de afstand tot het dichtstbijzijnde obstakel. Als de afstand kleiner is dan 20cm, draait de frigobox niet maar rijdt hij rechtdoor tot hij het obstakel voorbij is en de bocht kan nemen.

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: Botsingsdetectie

* 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] | Verdel, „Gebruikerservaring,” Wikipedia, 3 april 2019. [Online]. Available: https://nl.wikipedia.org/wiki/Gebruikerservaring. [Geopend 22 Maart 2020]. |
| [2] | M. Hein, „Gebruikersomgeving,” Wikipedia, 26 September 2018. [Online]. Available: https://nl.wikipedia.org/wiki/Gebruikersomgeving. [Geopend 22 Maart 2020]. |
| [3] | Flutter, „Flutter,” 12 Januari 2020. [Online]. Available: https://flutter.dev. [Geopend 12 Januari 2020]. |
| [4] | Flutter, „Flutter on Github,” Github, 12 Januari 2020. [Online]. Available: https://github.com/flutter. [Geopend 12 Januari 2020]. |
| [5] | Dart, „Dart,” Dart, 14 Januari 2020. [Online]. Available: https://dart.dev/. [Geopend 14 Januari 2020]. |
| [6] | Flutter, „BottomNavigationBar class,” 13 May 2020. [Online]. Available: https://api.flutter.dev/flutter/material/BottomNavigationBar-class.html. [Geopend 17 May 2020]. |
| [7] | Flutter, „AppBar class,” Flutter API, 13 May 2020. [Online]. Available: https://api.flutter.dev/flutter/material/AppBar-class.html. [Geopend 16 May 2020]. |
| [8] | Flutter, „Theme class,” Flutter, 20 Januari 2020. [Online]. Available: https://api.flutter.dev/flutter/material/Theme-class.html. [Geopend 20 Januari 2020]. |
| [9] | artrmz, „flutter\_control\_paf,” 1 Januari 2020. [Online]. Available: https://pub.dev/packages/control\_pad. [Geopend 23 Januari 2020]. |
| [10] | baseflow.com, „location\_permissions 2.0.5,” baseflow.com, 10 Maart 2020. [Online]. Available: https://pub.dev./packages/location\_permissions. [Geopend 12 Maart 2020]. |
| [11] | 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]. |
| [12] | huclengyue@gmail.com, „toast 0.1.5,” 16 Juli 2019. [Online]. Available: https://pub.dev/packages/toast. [Geopend 27 Februari 2020]. |
| [13] | ali.almoullim@gmail.com, „background\_location 0.0.9+3,” pub.dev, 2 March 2020. [Online]. Available: https://pub.dev/packages/background\_location. [Geopend 13 March 2020]. |
| [14] | 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]. |
| [15] | „Thermistor - Wikipedia,” Wikipedia, 4 april 2020. [Online]. Available: https://en.wikipedia.org/wiki/Thermistor. [Geopend 27 april 2020]. |
| [16] | Nate, „Analog to Digital Convertion - SparkFun,” SparkFun, [Online]. Available: https://learn.sparkfun.com/tutorials/analog-to-digital-conversion/all. [Geopend 27 april 2020]. |
| [17] | S. Bolt, „YouTube - Arduino with water proof, 2 Wire temperature Gauge (PT100),” 13 februari 2016. [Online]. Available: https://www.youtube.com/watch?v=GCNyZt\_vZW4. [Geopend 27 april 04]. |
| [18] | Gotron, „Gotron | NTC Thermistor,” EPCOS, [Online]. Available: https://www.gotron.be/ntc-thermistor-470ohm-10-tolerance.html. [Geopend 27 april 2020]. |
| [19] | H. STEM, „YouTube,” 19 maart 2019. [Online]. Available: www.youtube.com/watch?v=VPHv9FtoFlI. [Geopend 27 april 2020]. |
| [20] | 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]. |
| [21] | 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]. |
| [22] | 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]. |
| [23] | ladyada, „Adafruit\_Sensor,” 6 April 2020. [Online]. Available: https://github.com/adafruit/Adafruit\_Sensor. [Geopend 17 April 2020]. |
| [24] | tdicola, „Adafruit\_HMC5883\_Unified,” 27 Mei 2016. [Online]. Available: https://github.com/adafruit/Adafruit\_HMC5883\_Unified. [Geopend 17 April 2020]. |
| [25] | InternetArchiveBot, „Magnetic declination,” 22 Maart 2020. [Online]. Available: https://en.wikipedia.org/wiki/Magnetic\_declination. [Geopend 17 April 2020]. |
| [26] | M. Hart, „TinyGPSPlus,” 13 Oktober 2019. [Online]. Available: https://github.com/mikalhart/TinyGPSPlus. [Geopend 17 April 2020]. |
| [27] | Wikiwerner, „Stelling van Pythagoras,” 18 November 2019. [Online]. Available: https://nl.wikipedia.org/wiki/Stelling\_van\_Pythagoras. [Geopend 18 April 2020]. |
| [28] | 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]. |
| [29] | „Cursus Algebra (3),” 26 April 2020. [Online]. Available: http://www.davdata.nl/algebra/alg3.html. [Geopend 26 April 2020]. |
| [30] | SparkFun, „SparkFun - BlueSMiRF,” SparkFun, [Online]. Available: https://www.sparkfun.com/products/12577. [Geopend 27 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 |
| Oberon | 24/04/2020 | Toevoeging hoofdstuk 4 |
| Aaron | 26/04/2020 | Toevoeging bronnen, figuren en codefragmenten + layout aanpassingen |
| Oberon | 26/04/2020 | Nalezen en kleine toevoegingen |
| Elian | 27/04/2020 | Schrijven van 4.1 & nalezen 4.4 |
| Dries | 27/04/2020 | Schrijven 4.2, 4.3. Nalezen 4.1 en 4.4. |
| Dries & Aaron | 27/04/2020 | In orde brengen automatische verwijzing bronvermelding. |
| Aaron | 27/04/2020 | Nalezen + aanpassingen zinsconstructies + verbetering spellingsfouten |
| Aaron, Dries, Elian, Oberon | 02/05/2020 | Alles nalezen en verbeteren feedback. |
| Elian | 17/05/2020 | Herschrijven hoofdstuk 3 |
| Oberon | 17/05/2020 | Schrijven inleiding en controle hoofdstuk 3. |
| Aaron | 21/05/2020 | Begin schrijven abstract |
| Elian | 21/05/2020 | Abstract verbeteren, aanvullen en herschrijven |
| Aaron | 21/05/2020 | Schrijven hoofdstuk 1.1 |