



VRIJE
UNIVERSITEIT
BRUSSEL



BACHELORPROEF: SFERISCHE ROBOT

Eindverslag

Louis Croonen, Seppe Goossens

May 24, 2022

Academic year 2021-2022
Industriële wetenschappen

Contents

| | | |
|----------|--|-----------|
| 1 | Opdracht Beschrijving | 4 |
| 2 | Literatuurstudie | 5 |
| 2.1 | Soorten Sferische Robots | 5 |
| 3 | Aanpak | 7 |
| 3.1 | Eerdere Ideeën | 7 |
| 3.1.1 | Moduleren op basis van elektromagneten | 7 |
| 3.1.2 | Moduleren op basis van solenoïdes | 8 |
| 3.2 | Huidige Aanpak | 10 |
| 3.2.1 | Montage | 10 |
| 3.2.2 | Werking | 11 |
| 4 | Design evolutie | 14 |
| 4.1 | Ring | 14 |
| 4.2 | (Tand)wiel | 15 |
| 4.3 | Middenstuk | 16 |
| 4.4 | Binnenring | 16 |
| 5 | Elektronica | 17 |
| 5.1 | Micro Controller | 17 |
| 5.1.1 | ESP32 | 17 |
| 5.1.2 | Lineaire Voltage Regulator | 18 |
| 5.2 | Batterijen | 19 |
| 5.2.1 | Gebruikte Batterijen | 19 |
| 5.2.2 | Battery Level Indicators | 21 |
| 5.3 | Power Circuit + Reverse Voltage Protection Circuit | 23 |
| 5.3.1 | Fuse | 23 |
| 5.3.2 | Reverse Voltage Protection | 23 |
| 5.4 | Servo Circuit | 26 |
| 5.4.1 | Servo's | 26 |
| 5.4.2 | DC-DC Step Down Buck Converter | 28 |
| 5.4.3 | Optocouplers | 29 |
| 5.5 | Sensors | 31 |
| 5.5.1 | Absolute Orientatie Sensor | 31 |
| 5.6 | Besturing | 34 |
| 6 | Ondervonden Problemen | 35 |
| 6.1 | IO Pinnen ESP32 | 35 |
| 6.2 | Keuze 2 Batterijen | 36 |
| 6.3 | Extra Binnenring | 37 |
| 6.4 | Stijfheid Constructie | 37 |
| 6.5 | Servo's | 41 |
| 6.6 | Design | 41 |
| 7 | Performantie/Test | 42 |
| 7.1 | Algemene Specificaties | 42 |
| 7.2 | Accuraatheid Test | 42 |
| 8 | Zelfreflectie | 45 |
| 9 | Conclusie | 46 |

| | |
|--|-----------|
| A Appendix | 47 |
| A.1 Exploded Views | 47 |
| A.2 Componentenlijst | 48 |
| A.3 Technische Tekeningen | 50 |
| A.4 Schematic, Protoboard en PCB | 62 |
| A.4.1 Schematic | 62 |
| A.4.2 Protoboard en PCB | 63 |
| A.5 Blok Schema Elektronica | 65 |
| A.6 Algemeen Blok Schema | 65 |
| A.7 Code | 66 |
| A.7.1 Main Code | 66 |
| A.7.2 Battery Level Indicators | 66 |
| A.7.3 BNO055 Info | 67 |
| A.7.4 Movements | 67 |
| A.8 Flow Chart | 69 |
| A.9 Logboek | 70 |
| A.10 Foto's en Video's | 71 |
| A.10.1 Foto's | 71 |
| A.10.2 Video's | 73 |
| Sources | 74 |

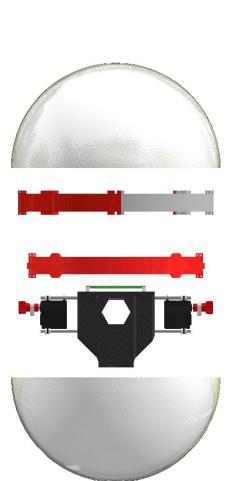


Figure 1: Onze Sferische Robot

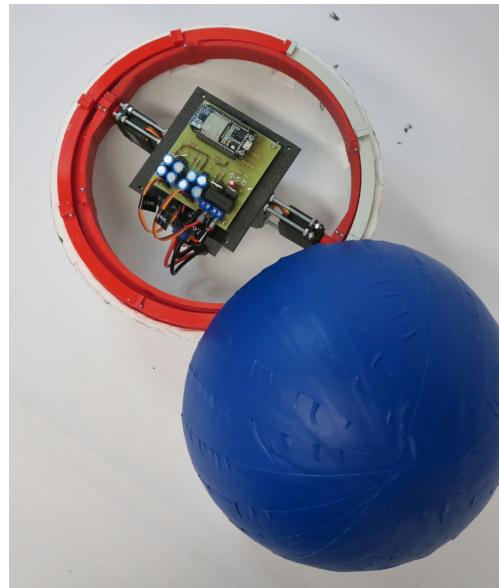
1 Opdracht Beschrijving

Door onze promotores werden we uitgedaagd om een sferische robot te bouwen dat kan bewegen d.m.v. het moduleren van een massa. De meeste reeds bestaande sferische robots maken gebruik van het concept van een hamster in een wiel. Door in de binnenkant van de sfeer de wanden op te klimmen en ze voorwaarts te duwen zal de sfeer en dus de robot beginnen rollen. Omdat dit een reeds gerealiseerde en werkende methode is, zijn we opzoek gegaan naar een alternatieve methode om een sferische robot te bouwen, zie sectie 3. De meest voor de hand liggende methode is door het moduleren van een massa dat zich binnenin de sfeer bevindt. Het moduleren van dat massamiddelpunt zal zo de robot in beweging brengen en doen rollen. Dit is gelijkaardig aan de hamster in de ring methode maar we moduleren effectief een massa om een beweging in gang te brengen. Hiernaast moet de robot gerealiseerd worden met low cost onderdelen.

In sectie 3 gaan we onze huidige maar ook eerdere methodes toelichten als ook de voor- en nadelen van deze methodes. In sectie 4 overlopen we welke aanpassingen die we hebben gemaakt in het design van de robot met de bijhorende redenen, sectie 5 bekijken we de elektronica die we hebben gebruikt om de robot te realiseren en in sectie 6 bespreken we de ondervonden problemen bij het bouwen van de robot. Verder voeren we in sectie 7 een test uit met de robot om een beeld te schetsen van de performantie van de robot. We voeren in sectie 8 een zelfreflectie uit waarin we aspecten toelichten waar er zeker nog verbetering mogelijk is. Tot slot volgt het documentatie gedeelte in sectie A waar de technische tekeningen, logboek, schema's en code zich bevinden.



(a) Exploded View Onze Sferische Robot



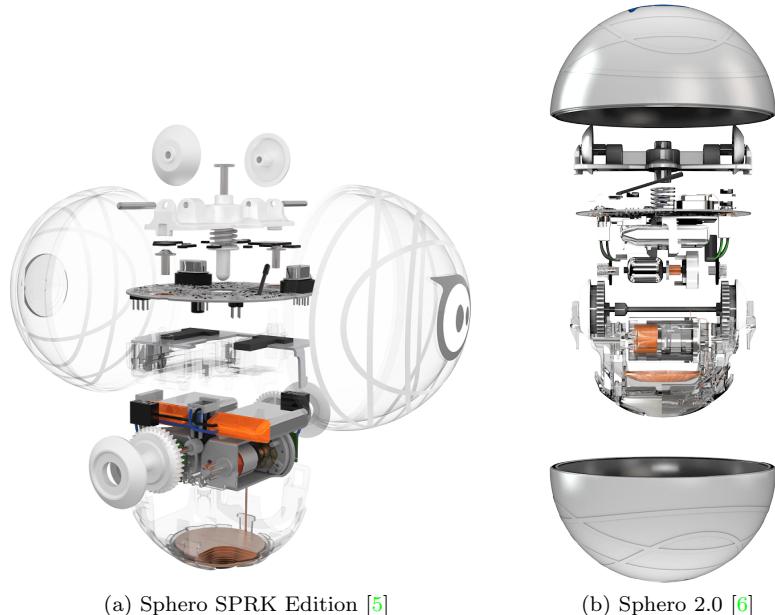
(b) Top View Onze Sferische Robot

Figure 2: Onze Sferische Robot

2 Literatuurstudie

2.1 Soorten Sferische Robots

Een sferische robot bestaat typisch uit een bolvormige schaal die gebruikt wordt als het lichaam van de robot in combinatie met een aandrijving. Deze aandrijving kan diverse vormen aannemen bij sferische robots. Dit is dus ook meestal het criteria waarin sferische robots zich onderscheiden van elkaar. We kunnen voornamelijk 2 categorieën onderscheiden: d.m.v. directe aandrijving of d.m.v. de zwaartekracht. In de eerste categorie bevindt zich de meest gekende methode, de "hamster car". Hierin wordt een autootje in een sfeer geplaatst zoals een hamster in een ring. Door de auto in de binnenvanden van de sfeer te laten opritzen zal het massamiddelpunt van de robot zich verschuiven en zal de robot beginnen rollen. Sphero is hier een voorbeeld van. De Sphero kan rond zijn als draaien door de wielen in tegengestelde richting te doen draaien. Om vooruit te bewegen moeten de wielen dezelfde richting uit gaan. Hierdoor zal het platform binnenvan de sfeer beginnen rijden en zal het zwaartepunt van de totale robot verschuiven. Dit laat op zijn beurt de robot rollen in de gewenste richting. Verder gebruikt de Sphero een elektronische gyrometer om te weten wat de bovenkant van de robot is en zo vlak mogelijk te blijven tijdens het rijden. Ook wordt er een accelerometer gebruikt om de snelheid en acceleratie bij te houden. [1] [2] [3] Met deze informatie zal de Sphero dan zo stabiel mogelijk blijven rijden. Hiernaast focust Sphero zich ook op het hele ecosysteem rond de Sphero producten. De Sphero SPRK+ laat bijvoorbeeld toe om de Sphero te programmeren via hun eigen app. Ook stelt Sphero API's ter beschikking voor developers om verder apps te ontwikkelen. Dit kan gebruikt worden om bijvoorbeeld meerdere Sphero's samen een taak te doen uitoefenen. [4] Vele projecten



(a) Sphero SPRK Edition [5]

(b) Sphero 2.0 [6]

Figure 3: Binnenkant van verschillende Sphero's

op het internet zijn gebaseerd op de Sphero. Vaak zijn deze wel minder ingewikkeld en maken ze gebruik van een speelgoed auto in een sfeer. Het principe blijft wel hetzelfde.

De tweede categorie maakt gebruik van het moduleren van het massamiddelpunt van de robot om zo een rollende beweging te creëren. Het idee bij deze methode is dat er een massa op een offset van het centrum van de robot geplaatst wordt. Deze zal dan door de zwaartekracht aangetrokken worden en dit zal de robot doen rollen. Dit mechanisme kan bijvoorbeeld bereikt worden door motoren die het massamiddelpunt van de robot gaan moduleren. Hierbij worden de motoren niet gebruikt voor een directe aandrijving van bijvoorbeeld een auto of de sfeer zelf maar voor het moduleren van een massa. Deze robots hebben vaak een laag hangend gewicht in het center van de sfeer. Dit gewicht kan dan gemoduleerd worden door gebruik te maken van motoren. Zo kan het gewicht naar links of naar rechts geleund worden zodat de robot tijdens het rollen naar links of naar rechts beweegt. Een groot verschil hierbij is dat de robot niet rond zijn as draait zoals bij de Sphero. De robot kan rollen door 2 tegenovergestelde punten op de sfeer te verbinden met elkaar. De verbinding wordt op zijn beurt vastgemaakt aan de hangende massa. Door de massa te verplaatsen zoals een pendelum zal de sfeer beginnen rollen. Het nadeel van dit type robot is dat het niet erg compact is, dit vooral omdat de krachtarm van de hangende massa lang genoeg moet zijn om een groot genoeg moment te creëren zodat de robot kan beginnen rollen. Bovendien kan de robot niet rond zijn as draaien wat leidt tot een minder precieze besturing. Dit type robot heeft zo wel zijn voordeelen tegenover bijvoorbeeld de Sphero maar aangezien het design en de uitvoering veel complexer is, wordt deze methode niet vaak toegepast.



Figure 4: Sferische Robot gebaseerd op een pendelum[7]

3 Aanpak

In de volgende secties gaan we ons onderzoek toelichten en uitleggen waarom we juist voor ons huidig idee hebben gekozen.

3.1 Eerdere Ideeën

3.1.1 Moduleren op basis van elektromagneten

Het eerste idee waarmee we van start gingen was het moduleren van het gewicht op basis van elektromagneten. De hoofdgedachte van deze aanpak is dat we een sferisch gewicht hebben in het midden van een grotere sfeer die bestaat uit een binnen en buiten sfeer met ertussen elektromagneten. Met behulp van deze elektromagneten zou de metalen bal in het midden kunnen verplaatst worden en de robot in zijn geheel doen rollen. Zie figuur 5 voor een schets.

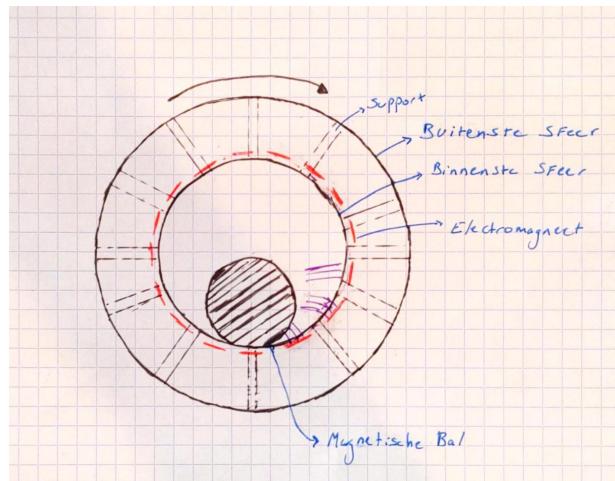


Figure 5: Sketch Idee op Basis van Elektromagneten

Er werden al snel een aantal problemen duidelijk, voornamelijk dat elektromagneten in het algemeen een zeer kort bereik hebben. De kracht in functie van de afstand voor een punt lading is: $F \sim r^{-2}$, de kracht neemt dus exponentieel af met de afstand. Ze zijn wel zeer sterk, maar pas als het voorwerp in direct contact staat met de elektromagneet. In principe is de bal altijd in contact met de elektromagneet die op de bodem ligt, maar om de bal te doen rollen zouden de naastliggende elektromagneten de bal moeten aantrekken. Het probleem hierbij is dat de elektromagneten enkel werken als het voorwerp in een directe lijn is met de elektromagneet. Als deze daarvan afwijkt zal het voorwerp niet aangetrokken worden. Ook de afstand tussen de bal en de elektromagneten zal steeds te groot zijn om een grote kracht te kunnen realiseren.

Een oplossing voor het rechte lijn probleem zou zijn om meer elektromagneten te plaatsen zodat er wel altijd één elektromagneet in directe lijn staat met de middelste bal. Dit vergt wel meer totale stroom, met als gevolg een grotere batterij. Dit resulteert in een volgende complicatie. Een omvangrijkere robot zal een zwaarder beweegbaar massamiddelpunt nodig hebben om te bewegen, wat zijn beurt nog meer stroom zou vragen om aan te trekken. Een vicieuze cirkel dus. Er werd ook gekeken naar hoe een elektromagneet sterker kan gemaakt worden. Er waren twee opties voor het verkrijgen van een elektromagneet. De eerste eenvoudigste en meest compacte

optie is er een aankopen. De andere optie is er zelf één bouwen wat meer vrijheid levert bij het dimensioneren van de elektromagneet. Je kan een elektromagneet bouwen met koperdraad rond een schroef te winden. Dit is geen compacte oplossing. Om een zelfgemaakte elektromagneet sterker te maken kan je drie dingen doen. Maar deze hebben direct ook een trade-off:

- Meer stroom door de wikkelingen → grotere voeding → grotere batterij → grotere robot
- Groter aantal wikkelingen → grotere elektromagneet → grotere robot
- Elektromagneet buigen in hoefijzer vorm → grotere elektromagneet → grotere robot

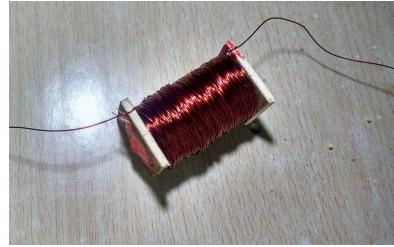


Figure 6: DIY Elektromagneet [8]

Een andere aanpak waar we ook over hebben nagedacht is het gebruiken van een accelerator of "coilgun". Een accelerator bestaat uit een buis waarrond ijzeren windingen zijn gedraaid. Door het precies timen van het aan en uitschakelen van de spoelen kan je bijvoorbeeld een ijzeren bal doen voortbewegen. We hadden het idee om dit te implementeren in een cirkel zodat het in een sfeer zou passen.

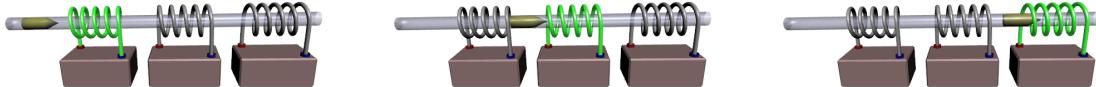


Figure 7: Werking Coilgun [9]

Als we dan één of meerdere ballen zouden gebruiken en genoeg spoelen zouden we in principe de ballen in de buis kunnen doen voortbewegen en dus ook de grote buitenste sfeer. Hiermee worden de problemen van de elektromagneten geëlimineerd. Anderzijds laat deze aanpak je wel maar toe om in één richting te bewegen en met beperkte controle. Ook heb je altijd een start-duw nodig zodat de balletjes in de buis begint te rollen. Ook deze aanpak leek niet volledig geschikt voor het voortbewegen van de robot d.m.v. modulatie van de massa. In theorie zou dit misschien werken maar in praktijk had dit een zeer moeilijke oplossing geweest.

3.1.2 Moduleren op basis van solenoïdes

Solenoides en elektromagneten zijn zeer gelijkaardig. Het grootste verschil is dat een solenoïde een ijzeren wikkeling/spoel is waardoor een stroom wordt gestuurd. Een elektromagneet bevat naast een ijzeren wikkeling ook een ijzeren kern, die ervoor gaat zorgen dat de magnetische

veldlijnen geconcentreerd worden. Het grote verschil is dus dat een elektromagneet een sterker magnetisch veld opwekt. Men dacht eraan om dan een solenoïde te gebruiken in combinatie met een beweegbare ijzeren kern. Wanneer er een stroom op de wikkeling wordt gezet zal het geproduceerde magnetisch veld de beweegbare ijzeren kern aantrekken. De ijzeren kern zou terug naar zijn originele positie gaan met behulp van een veer. Dit laat dus toe om elektrische energie om te zetten in mechanische energie.



Figure 8: Elektromagneet versus Solenoïde [10]

Omdat solenoïde een sterke trekkracht hebben en breed beschikbaar zijn in verschillende dimensies, dachten we dat dit misschien een optie kon zijn om het massa middelpunt te moduleren. Het idee was om 6 solenoïdes in een kruis te plaatsen, zie sketch 9 die een 2D versie illustreert. Aan het uiteinde van elke solenoïde zou een extra massa gehangen worden. Door de solenoïdes aan te sturen wordt het gewicht verder van het middelpunt verschoven. Dit resulteert in een langere kracht arm en dus een groter moment. Op die manier zou de bal moeten rollen. Het is belangrijk dat de solenoïdes perfect in het midden zouden staan, wat redelijk moeilijk te realiseren is.

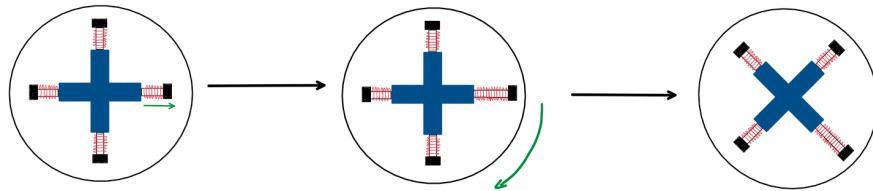


Figure 9: Sketch Idee op Basis van Solenoïde

3.2 Huidige Aanpak

Na het verwerpen van de vorige concepten, werd er besloten om te werken met servomotoren. Deze aanpak sluit het dichtste aan bij de 2de categorie dat besproken is in sectie 2. We gebruiken motoren om een massa te moduleren en zo een beweging te creëren. Servomotoren hebben een hoog vermogen voor hun compacte vorm en ze leveren veel koppel wat geschikt is voor het moduleren van de massa. De ring laat toe om via een rotatie de gewenste rol richting uit te kiezen.

3.2.1 Montage

In het midden is er een omhullende 3D print met plaats voor de batterijen en een PCB. Deze 3D print wordt bevestigd aan twee servomotoren met vier draadstangen zoals weergegeven in figuur 10. Aan de as van beide servo's is een tandwiel bevestigd die ronddraait in een ring. In deze binnenring is een tandwiel aangebracht, zie figuur 10 (d). Rond de kleinere tandwielen die aan de servo's bevestigd zijn, is een lager bevestigd die voor de verbinding zorgt tussen de kleine tandwielen en een binnenring. Deze binnenring is ontworpen met de bedoeling dat de tandwielen gestationeerd blijven binnen de buitenste ring. De buitenste ring is dan bevestigd in een bol uit piepschuim.

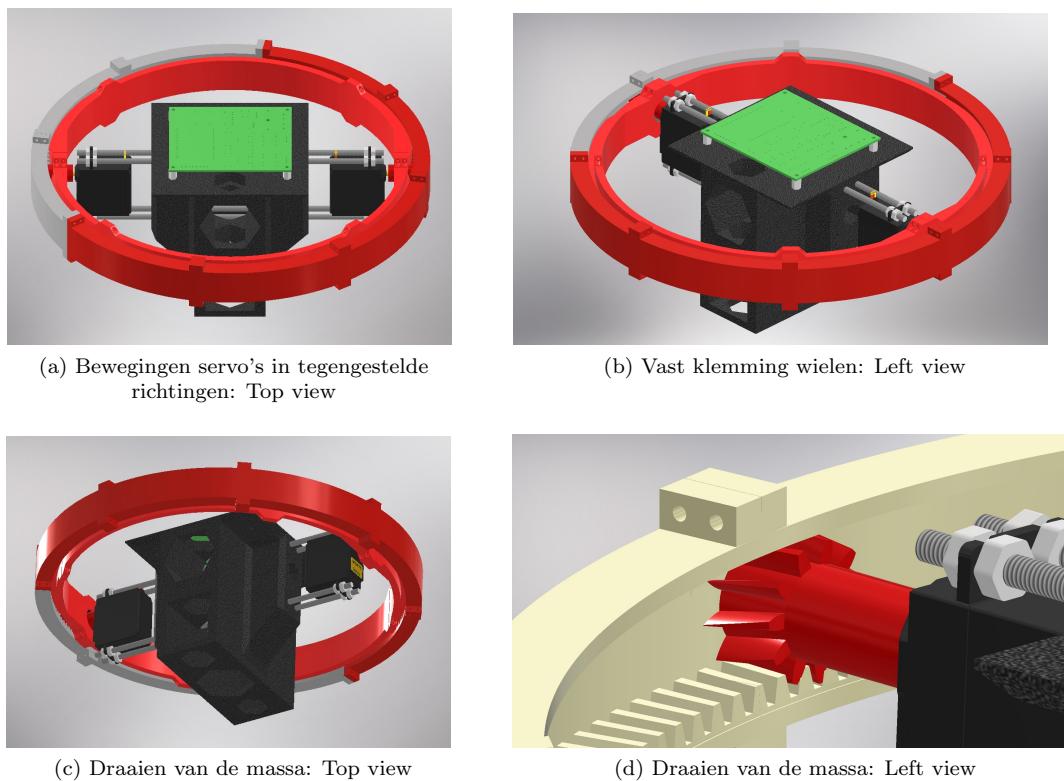


Figure 10: 3D-model final design

3.2.2 Werking

Het effectief moduleren van de massa gebeurt door het activeren van de servo's in tegengestelde richtingen. Met dezelfde richting wordt bedoeld, beide met de klok mee of tegen de klok in als je naar de servo kijkt, zie figuur 11. Als we de linkse en de rechtse servo beide met de klok mee (of beide tegen de klok in) laten draaien dan zal de massa rond zijn als draaien. Hiermee kan de rij-richting van de robot bepaald worden, zie figuur 12.

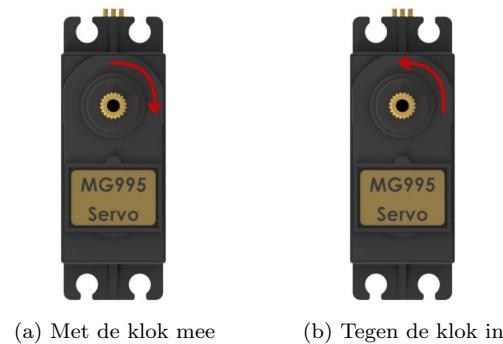


Figure 11: Draairichtingen servo

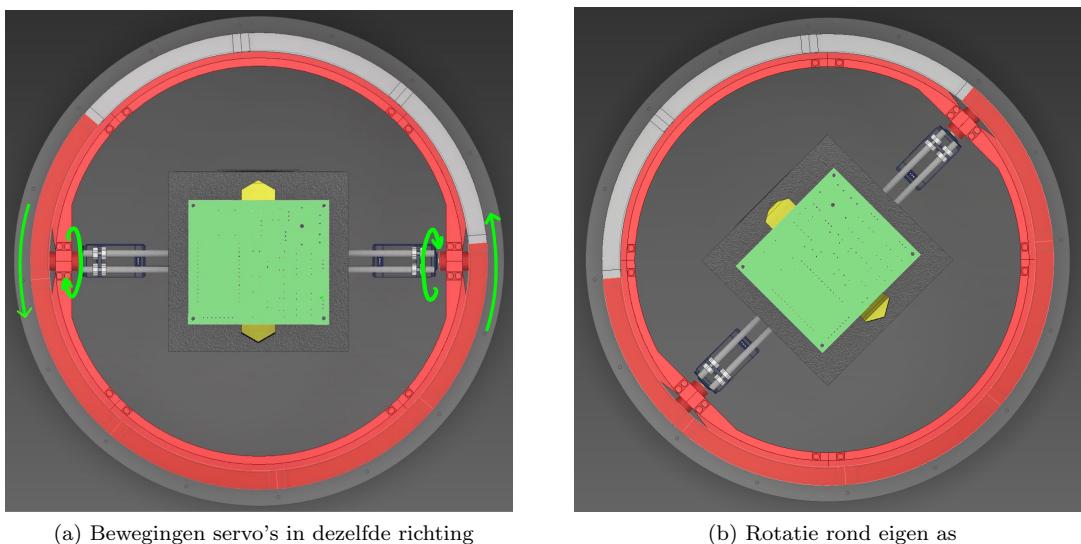


Figure 12: Rotatie van de robot rond eigen as

Onderstaand stappenplan overloopt nog eens de werking van de robot, ook geïllustreerd in figuur 13 en figuur 14.

1. Beide servo's draaien in tegengestelde richting (één met de klok mee en de andere tegen de klok in), ook zichtbaar in figuur 14 (a,b).
2. De kleine tandwielen willen dezelfde richting uit, hetgeen gehinderd wordt door de binnerring die tegen de buitenring aanduwt.
3. De massa gaat naar omhoog dus het massamiddelpunt ligt niet meer in het midden, ook zichtbaar in figuur 14 (c,d).
4. De sfeer rolt om het massamiddelpunt weer in het midden te brengen.

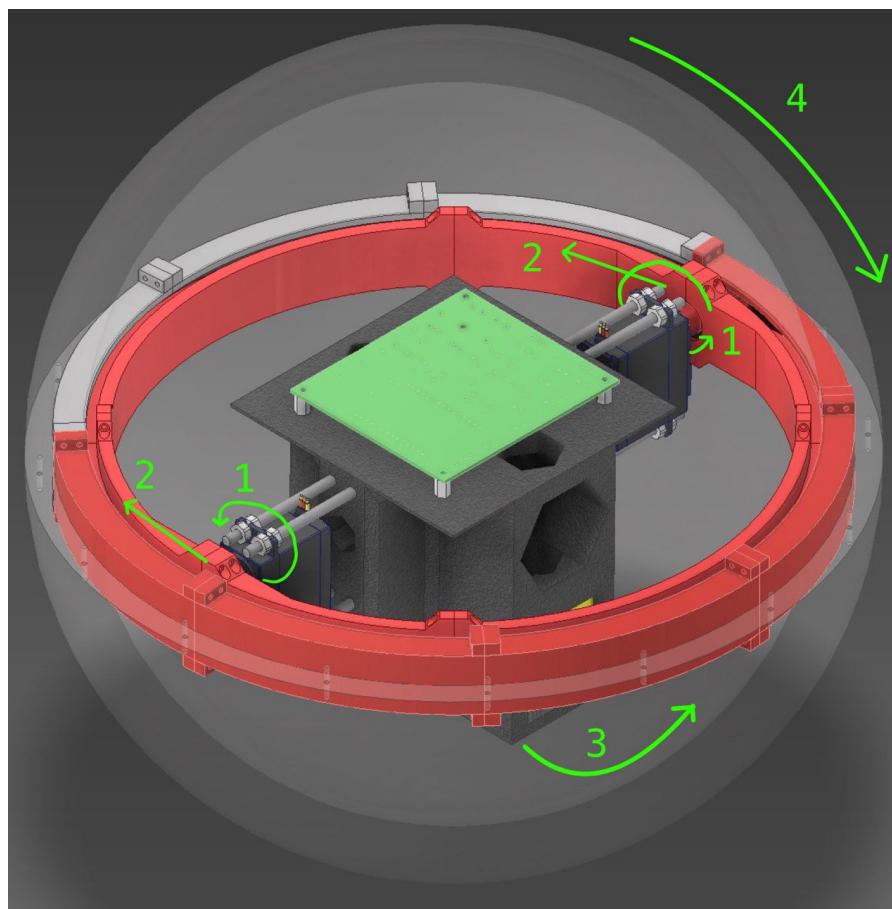
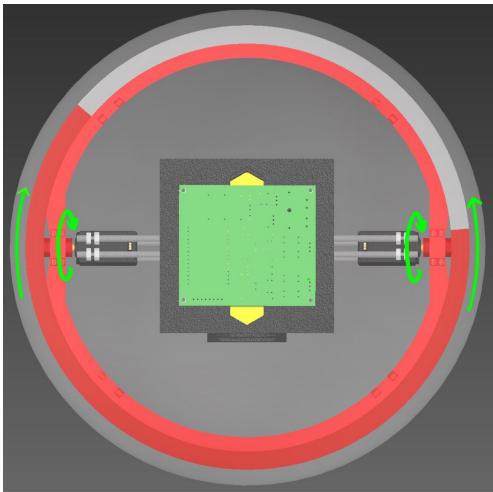
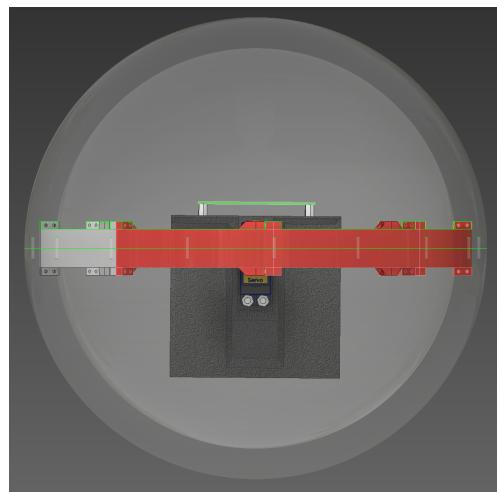


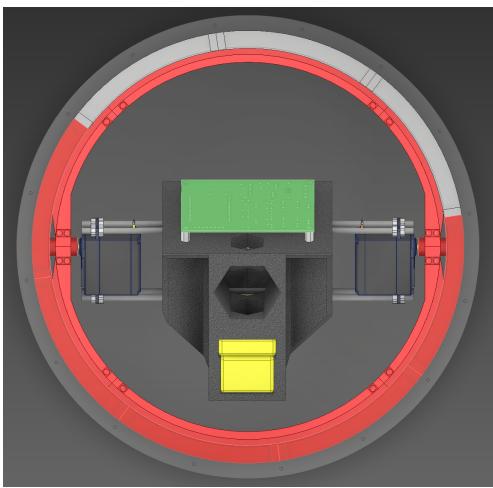
Figure 13: Mechanisme voor het Rollen van de Robot: Perspectief



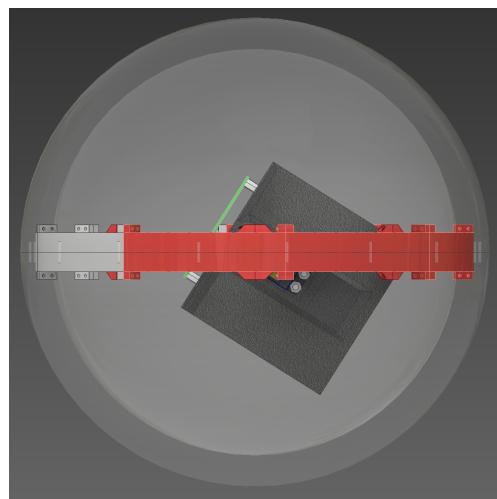
(a) Bewegingen servo's in tegengestelde richtingen:
Top view



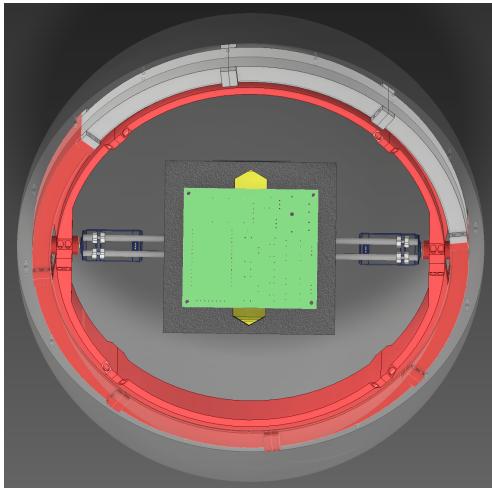
(b) Vast klemming wielen: Left view



(c) Draaien van de massa: Top view

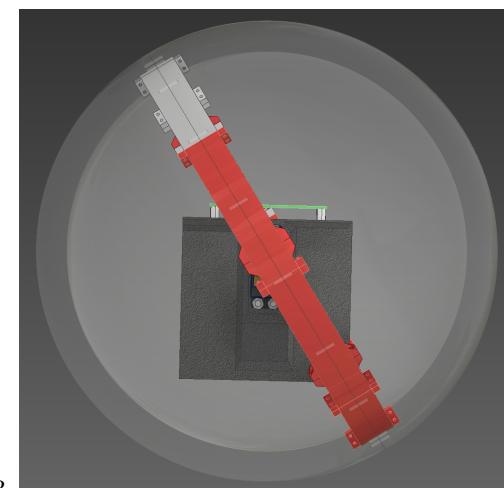


(d) Draaien van de massa: Left view



(e) Robot na het rollen: Top view

13



(f) Robot na het rollen: Left view

Figure 14: Rollen van de robot

4 Design evolutie

In dit hoofdstuk wordt de evolutie van de robot weergegeven. De belangrijkste upgrades en problemen dat opdoken met de bijbehorende oplossingen zullen hier besproken worden. De onderdelen die aan bod zullen komen zijn de ring, het (tand)wiel, het middenstuk en de binnenring. Ook de redeneringen waarom we bepaalde onderdelen zo hebben ontworpen worden omschreven.

4.1 Ring

In figuur 15 zijn vier afbeeldingen weergegeven, men kan er twee verschillende ringen onderscheiden. De eerste ring (a,b) was het oorspronkelijke idee waarbij de servo's rond draaien met een wieltje zoals in figuur 16 (a). Het wiel blijft op zijn plaats door de bult die in de gleuf van het wiel past. Dit was in theorie een goed concept, maar in de praktijk was dit minder vanwege de precisie die nodig is om de gleuf en de bult perfect in elkaar te doen passen. Dit 3D-printen was ingewikkeld waardoor de prints een beetje moesten bijgeschuurd worden zodat ze goed zouden passen. Deze methode was heel inefficiënt, de wielen draaiden soms rond hun as door gebrek aan grip met als gevolg dat het ontwerp aangepast moest worden.

Vervolgens werden er tandwielen gebruikt. In tegenstelling tot de vorige methode, zorgt dit ervoor dat de servo's veel meer grip hebben en dus hun kracht beter kunnen overdragen. Na dit te hebben toegepast zijn we tot de conclusie gekomen dat dit een geschikte manier is om te servo's te verplaatsen. Deze manier is nog steeds van toepassing in de finale robot.

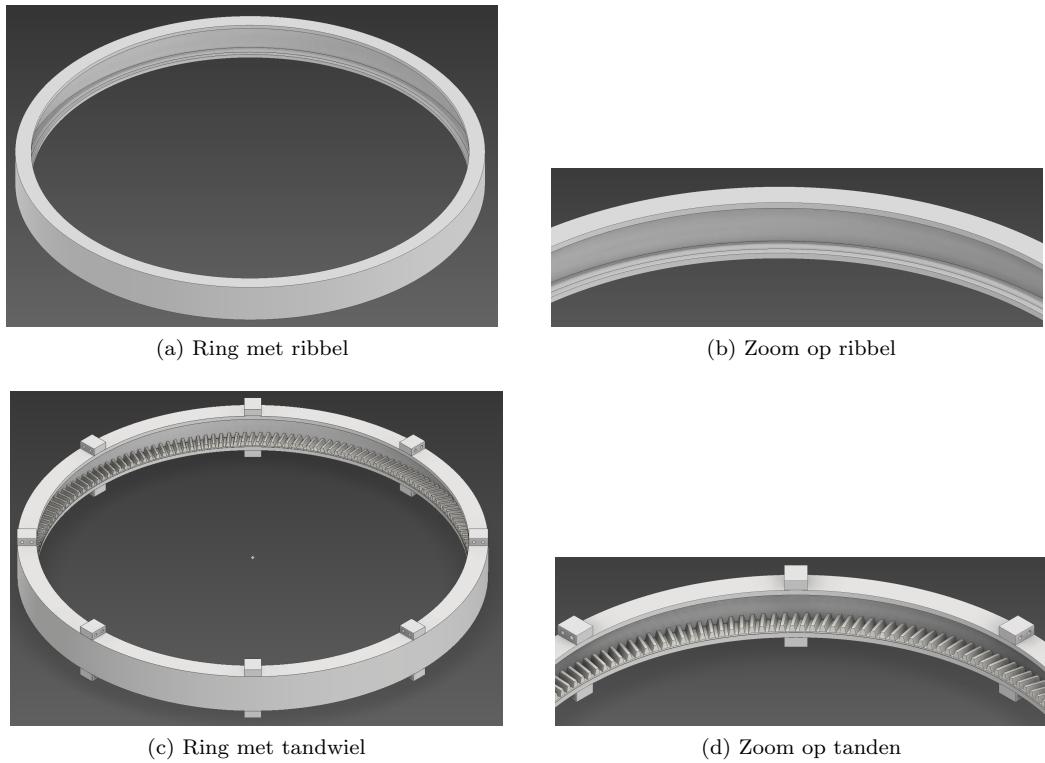


Figure 15: Buiterring van de sferische robot

4.2 (Tand)wiel

Aangezien er veranderingen werden toegepast aan de ring, spreekt het voor zich dat ook het wiel sommige veranderingen is ondergaan. In figuur 16 ziet men drie verschillende designs. Waarom het eerste wiel (a) niet geschikt was wordt uitgelegd in het sectie 4.1. Onze eerste versie van het tandwiel is te zien op afbeelding b en c. Het tandwiel (c) werd met behulp van schroeven vastgemaakt aan het onderdeel (b). In het finale tandwiel (d) wordt er gebruik gemaakt van de kleine tandwielen op de servo om de koppel over te dragen en een bout voor de stevigheid.

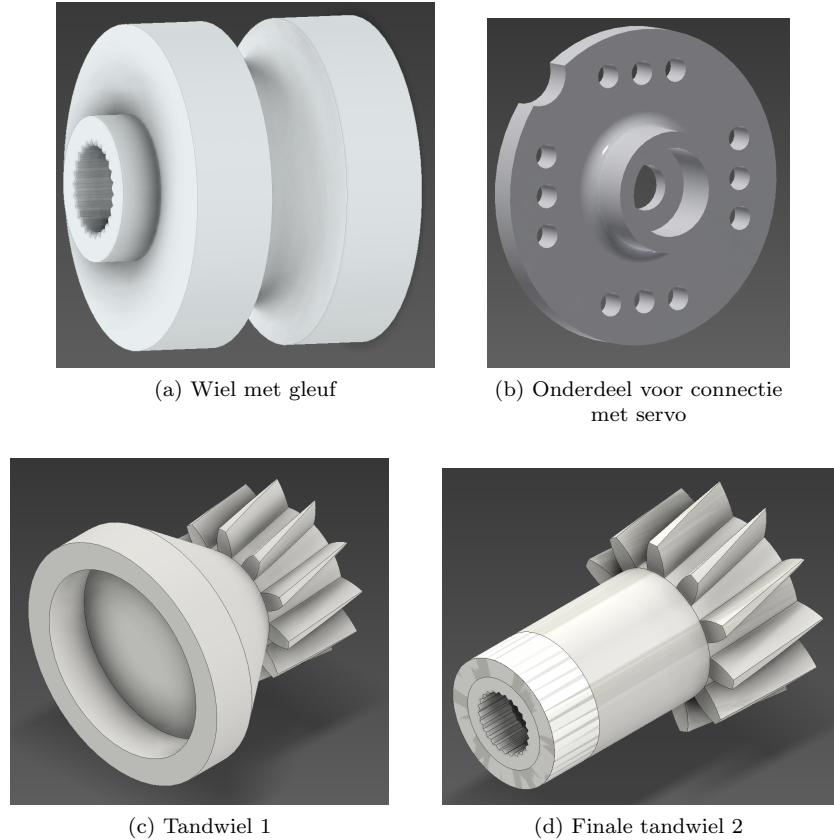


Figure 16: Gebruikte onderdelen voor het wiel

4.3 Middenstuk

Door sommige ondervonden problemen, in detail beschreven in sectie 6.4 heeft men besloten om de stijfheid van de constructie te verbeteren. Zichtbaar in figuur 17, waar afbeelding b robuuster is dan afbeelding a. Men merkt ook op dat het robuuste middenstuk plaats heeft voor twee batterijen. Dit is een oplossing voor een probleem uitgelegd in sectie 6.2.

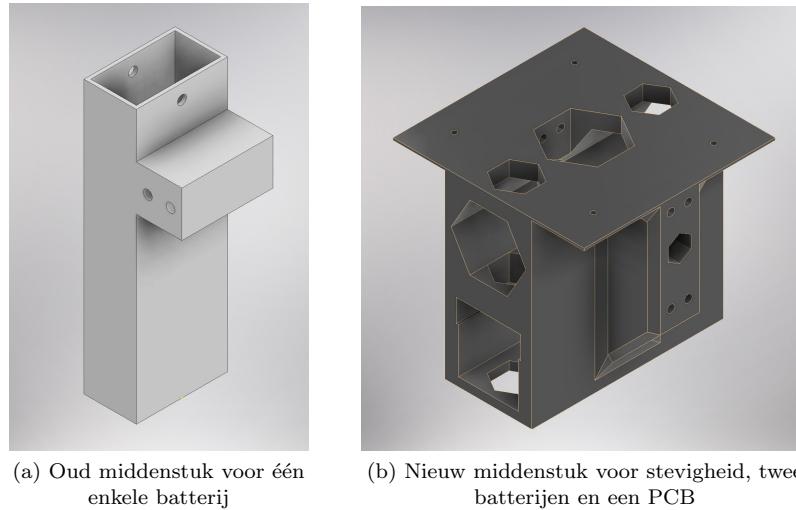


Figure 17: Middenstuk van de robot

4.4 Binnenring

De binnenring heeft een cruciale rol gespeeld in het ontwerpen van de robot. Het is de oplossing van al de laatste getroffen problemen. Dit is meer in detail besproken in sectie 6.3.

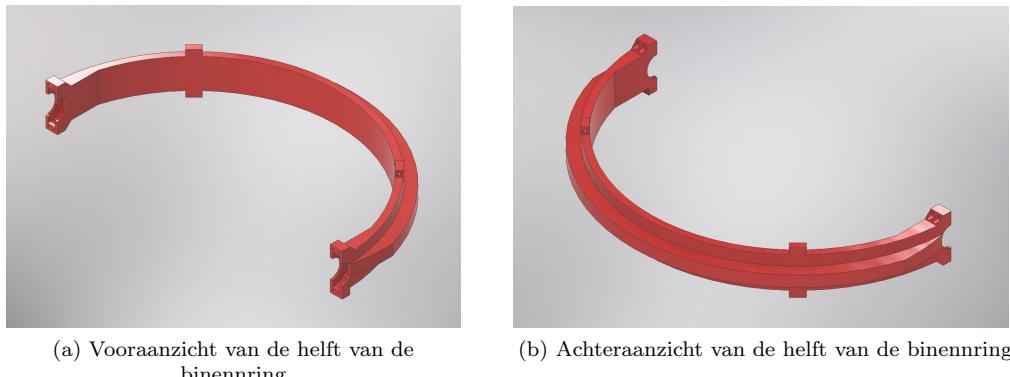


Figure 18: Binnenring van de robot

5 Elektronica

5.1 Micro Controller

5.1.1 ESP32

De micro controller die we gebruikt hebben is de ESP-WROOM-32. Specifieker de NodeMCU module.

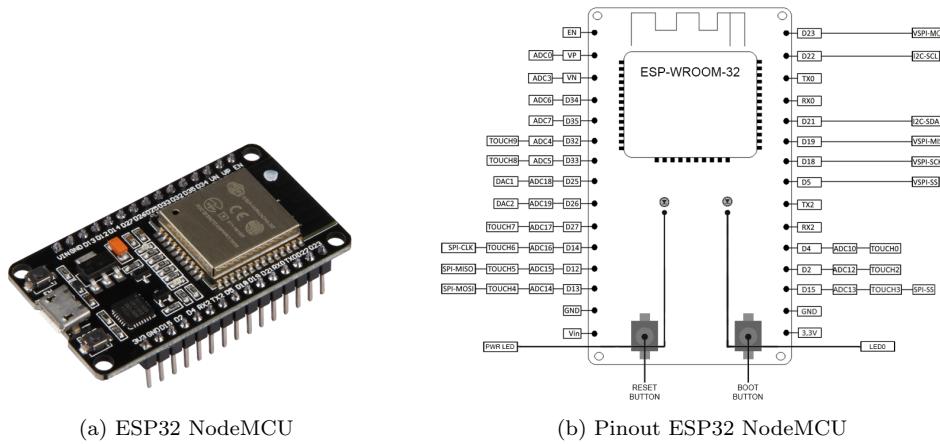


Figure 19: ESP-WROOM-32 module op NodeMCU bord [11]

Waarom zijn we dan voor de ESP32 gegaan? Grotendeels voor de ingebouwde Bluetooth functies. Vermits we de robot draadloos moeten besturen, is de ingebouwde Bluetooth functie van de ESP32 zeer handig. Via een app op de smartphone kunnen we dan op een eenvoudige manier met de ESP32 communiceren over Bluetooth. Met een Arduino library kan de Bluetooth geactiveerd worden. Ook bevat de NodeMCU ESP32 genoeg IO pinnen voor onze toepassing en kunnen we de vertrouwde Arduino IDE gebruiken.

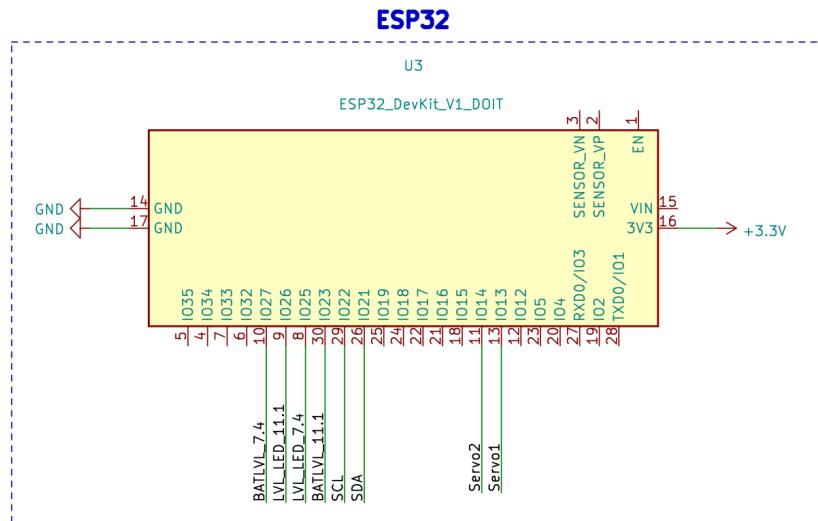


Figure 20: ESP32 Schematic

IO pinnen 27 en 26 zijn verbonden met de spanningsdeler voor het controleren van het batterij niveau. IO pinnen 25 en 23 zijn verbonden met de bijhorende status LED's. De datapinnen van de servo's zijn verbonden met IO pinnen 14 en 13. De ESP32 wordt gevoed met 3.3V van de lineaire voltage regulator.

5.1.2 Lineaire Voltage Regulator

7.4V → 3.3V Circuit

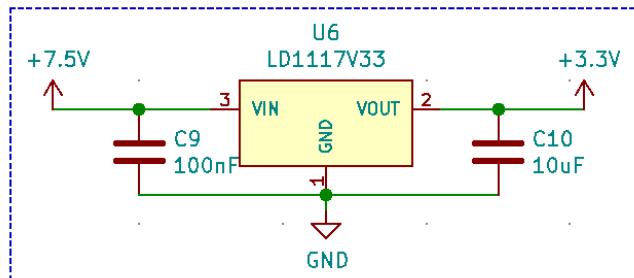


Figure 21: Lineaire Voltage Regulator Schema

Omdat we met batterijen werken kan de ESP32 niet gevoed worden via zijn micro-USB poort. We hebben gebruik gemaakt van een lineaire voltage regulator namelijk de LD1117V33. Deze zet de ingang spanning om in 3.3V. Volgens de datasheet van de ESP32 is het aangeraden om een voeding spanning van 3.3V te gebruiken die constante uitgang stroom van minstens 500mA kan leveren.

- The operating voltage of ESP32 ranges from 2.3 V to 3.6 V. When using a single-power supply, the recommended voltage of the power supply is 3.3 V, and its recommended output current is 500 mA or more.

Figure 22: Aangeraden Voeding ESP32 [12]

Volgens de datasheet van de LD1117V33 is de maximum ingang spanning 15V en kan de lineaire voltage regulator een constante uitgang stroom van minstens 800mA leveren. Dit is dus voldoende voor het voeden van de ESP32.

| | | | | | | |
|----------|-------------------------|--|--|--|----|---|
| V_{in} | Operating Input Voltage | | | | 15 | V |
|----------|-------------------------|--|--|--|----|---|

Figure 23: Maximum Ingang Spanning LD1117V33 [13]

| | | | | | | | |
|-------|----------------|------------------------------|--------------------------|-----|-----|------|----|
| I_o | Output Current | $V_{in} - V_O = 5 \text{ V}$ | $T_J = 25^\circ\text{C}$ | 800 | 950 | 1300 | mA |
|-------|----------------|------------------------------|--------------------------|-----|-----|------|----|

Figure 24: Uitgang Stroom LD1117V33 [13]

5.2 Batterijen

5.2.1 Gebruikte Batterijen

De huidige opstelling van de robot gebruikt 2 batterijen, een voor het voeden van de ESP32 en bijhorende componenten en 1 voor het voeden van de servo's. In sectie 6.2 motiveren we onze keuze voor het gebruiken van 2 batterijen.

De voeding voor de ESP32 en bijhorende componenten is een Zippy Compact 1000mAh 2S 25C LiPo batterij. De Zippy Compact is een LiPo batterij met een gewicht van 67g. De capaciteit van de batterij kunnen we uitdrukken in Watt-uur. Dit wordt dan:

$$Wh = V * Ah = 7.4V * 1Ah = 7.4Wh \quad (1)$$



Figure 25: Zippy Compact 1000mAh 2S 25C LiPo Batterij [14]

De C-rating van de batterij vertelt ons hoeveel stroom de batterij veilig kan leveren zonder dat de batterij hierbij beschadigd wordt. De Zippy Compact heeft een C-rating van 25C. Dit

betekent dus dat de batterij veilig kan ontladen aan 25 * capaciteit of:

$$\begin{aligned} 1000mAh / 1000 &= 1A \\ \Rightarrow 25C * 1A &= 25A \text{ beschikbaar} \\ \Rightarrow 60/25 &= 2.4 \text{ minuten} \end{aligned}$$

De batterij kan dus veilig 25A voor 2.4 minuten leveren zonder dat de batterij hierbij schade op loopt. Dit is zeker voldoende voor onze toepassing

Deze Zippy Compact batterij is voldoende voor het voeden van de ESP32 en bijhorende componenten. De ESP32 verbruikt in actieve mode tussen de 160mA en 260mA [15]. Bij pieken kan de ESP32 tot 500mA verbruiken. Stel dat de ESP32 en de bijhorende componenten te samen een constant verbruik van 500mA hebben. Dan kan de Zippy Compact batterij de ESP32 voeden voor 2uur:

$$\begin{aligned} 1000mAh &= 1A \text{ voor 1 uur} \\ \Leftrightarrow 0.5A &= 2 \text{ uur} \end{aligned}$$

Dit is dus voldoende voor onze toepassing.

De batterij voor de servo's is een MCM Power 3S 11.1V 2200mAh 30C batterij. Deze weegt 172g. De capaciteit van de batterij in Wh is:

$$\begin{aligned} Wh &= V * Ah \\ &= 11.1V * 2.2Ah \\ &= 24.42Wh \end{aligned}$$

We zien dat de MCM Power batterij een veel grotere capaciteit heeft in vergelijking met de Zippy Compact. De MCM Power is ook fysiek groter. De C-rating is 30C of m.a.w:

$$\begin{aligned} 2200mAh / 1000 &= 2.2A \\ \Rightarrow 30C * 2.2 &= 66A \text{ beschikbaar} \\ \Rightarrow 60/25 &= 2 \text{ minuten} \end{aligned}$$

De MCM Power batterij kan dus veilig 66A leveren voor 2 minuten zonder schade op te lopen. Deze batterij volstaat voor het voeden van de servo's. De servo's verbruiken samen tussen de 1A-1.8A. De stall current van beide servo's kunnen oplopen tot 5A [16]. De batterij kan zeker de servo's voor een uur voeden: 2200mAh = 2.2A voor 1 uur. Zelfs bij de stall current van de servo's kan de batterij het nodige leveren voor minstens 25 minuten terwijl we veilig binnen de C-rating blijven.



Figure 26: MCM Power 3S 11.1V 2200mAh 30C LiPo Batterij

Uit testen bleek dat het voeden van de servo's met de Zippy Compact niet werkte door de te grote stroom pieken. Een batterij met een grotere capaciteit was dus nodig. Maar het voeden van zowel de ESP32 en als de servo's uit deze grotere batterij gaf ook een aantal problemen. In sectie [6.2](#) leggen we deze problemen uit en motiveren we de keuzes die we gemaakt hebben.

5.2.2 Battery Level Indicators

Verder zijn er ook 2 LED's aanwezig (1 per batterij), die aangeven of de batterijen onder een bepaald spanning level is. Dit is om te voorkomen dat we de batterijen beschadigen door ze op een te lage spanning te gebruiken. Het tijdig alarmeren kan dus beschadiging voorkomen en het helpt ook om te weten wanneer de batterijen opgeladen moeten worden.

Hiervoor hebben we gebruik gemaakt van een spanningsdeler. Deze zorgt ervoor dat we via de ADC (Analog to Digital Converter) pins van de ESP32 de spanning van de batterij kunnen lezen. De ADC pinnen hun maximum spanning is 3.3V. We kunnen dus niet rechtstreeks de spanning van de batterij aflezen daarom gebruiken we de spanningsdeler. Deze zorgt ervoor dat wanneer de batterij opgeladen is, er 3.3V op de ADC pin staat en wanneer de batterij bijna leeg is er een lagere spanning gelezen wordt en we de LED kunnen doen branden.

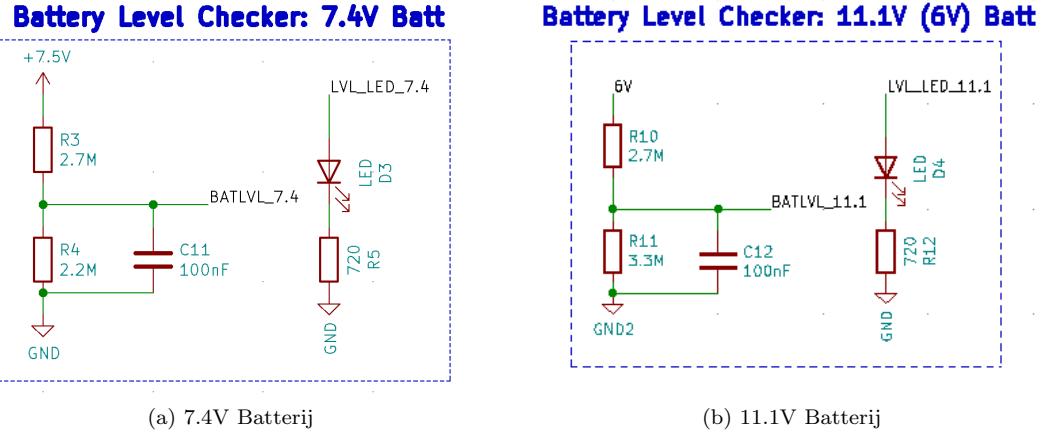


Figure 27: Schematics Battery Level Checkers

Voor de 7.4V batterij willen we dus dat, wanneer de batterij volledig opgeladen is en dus de nominale spanning 7.4V bedraagt, er 3.3V op de ADC pin terechtkomt. De spanning van de batterij op welke de LED's gaan branden is 4V. We weten dat de formule voor de spanningsdeler er als volgt uitziet:

$$V_{out} = V_{in} \frac{R_2}{R_2 + R_1}$$

Met R_1 gelijk aan R_3 en R_2 aan R_4 en R_{11} in figuur 27. Als we weten dat we bij 7.4V, 3.3V op de ADC pin willen lezen dan kunnen we de verhouding tussen R_1 en R_2 als volgt berekenen:

$$\begin{aligned} V_{out} &= V_{in} \frac{R_2}{R_2 + R_1} \\ \Leftrightarrow \frac{V_{out}}{V_{in}} &= \frac{R_2}{R_2 + R_1} \\ \Leftrightarrow \frac{3.3V}{7.4V} &= \frac{R_2}{R_2 + R_1} \\ \Leftrightarrow 0.446 &= \frac{R_2}{R_2 + R_1} \\ \Leftrightarrow 0.446(R_1 + R_2) &= R_2 \\ \Leftrightarrow 0.446R_1 + 0.446R_2 &= R_2 \\ \Leftrightarrow 0.446R_1 &= R_2 - 0.446R_2 \\ \Leftrightarrow 0.446R_1 &= (1 - 0.446)R_2 \\ \Leftrightarrow 0.446R_1 &= 0.554R_2 \\ \Leftrightarrow R_1 &= 1.242R_2 \end{aligned}$$

We kiezen de weerstand waarden best zo hoog mogelijk zodat de stroom die verloren gaat zo laag mogelijk is. We hebben gekozen voor $2.7M\Omega$ en $2.2M\Omega$. Hiernaast kunnen we ook een capaciteit toevoegen die de ruis op de ADC pin vermindert. De waarde hiervan is 100nF.

De 11.1V van de batterij wordt omgezet in 6V door de buck converter. Hierdoor bedraagt de verhouding tussen de weerstanden 0.82. De gekozen weerstanden zijn voor R10 en R11 en zijn respectievelijk $2.7\text{M}\Omega$ en $3.3\text{M}\Omega$.

5.3 Power Circuit + Reverse Voltage Protection Circuit

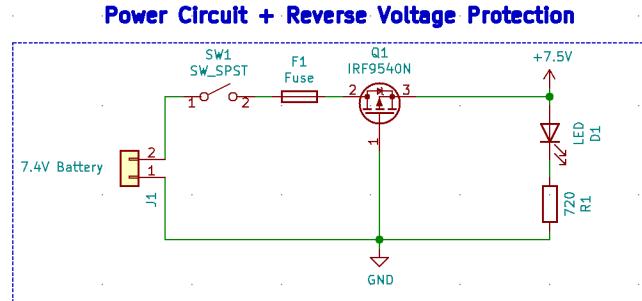


Figure 28: Power Circuit + Reverse Voltage Protection Circuit

Dit circuit is ter beveiliging van de microcontroller. De P-channel mosfet zorgt ervoor dat enkel en alleen als de batterij juist aangesloten wordt met de juiste polariteit, de stroom wordt doorgelaten naar de microcontroller. Hiernaast bevat het schema verder ook een on/off switch, fuse en een status LED.

5.3.1 Fuse

De fuse of smeltzekering beschermt de achterliggende componenten tegen een te hoge stroom. De waarde van de fuse is 2A.

5.3.2 Reverse Voltage Protection

P-Channel mosfet

Voor het beschermen van de microcontroller tegen omgekeerde polariteit maken we gebruik van een P-channel mosfet, namelijk de IRF9540N.

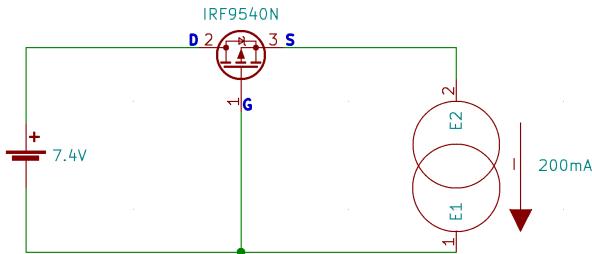


Figure 29: Reverse Voltage Protection met P-Channel mosfet

Wanneer de batterij juist wordt aangesloten zal de mosfet in doorlaat staan. Als de batterij fout wordt aangesloten zal de mosfet zich gedragen als een open gate en dus geen stroom doorlaten.

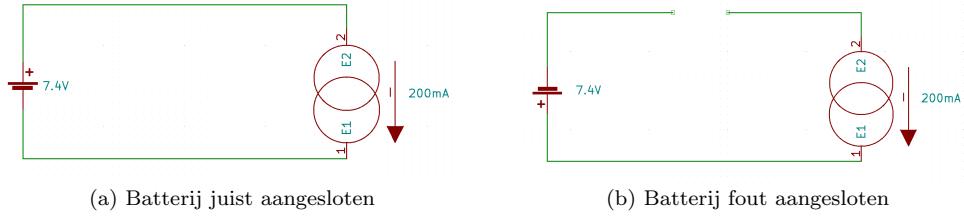


Figure 30: Gedrag P-Channel mosfet i.f.v. aansluiting batterij

Wanneer de batterij juist wordt aangesloten zal de mosfet in doorlaat staan als de drain-source spanning $V_{GS_{th}}$ kleiner is dan -4V, de gate threshold voltage. Dit omdat we met een P-channel mosfet werken. Deze waarde kunnen we terug vinden in de datasheet:

| | | | | | | |
|--------------|------------------------|------|---|------|---|---------------------------------------|
| $V_{GS(th)}$ | Gate Threshold Voltage | -2.0 | — | -4.0 | V | $V_{DS} = V_{GS}$, $I_D = -250\mu A$ |
|--------------|------------------------|------|---|------|---|---------------------------------------|

Figure 31: Drain Source spanning IRF9540N [17]

Ook de spanningsval V_{DS} over de Body diode van de mosfet kunnen we terug vinden:

| | | | | | | |
|----------|-----------------------|---|---|------|---|---|
| V_{SD} | Diode Forward Voltage | — | — | -1.6 | V | $T_J = 25^\circ C$, $I_S = -11A$, $V_{GS} = 0V$ ④ |
|----------|-----------------------|---|---|------|---|---|

Figure 32: Spanningsval Body diode IRF9540N [17]

De source spannings van de mosfet kan dan als volgt berekend worden:

$$\begin{aligned} V_S &= V_{Batt} - V_{DS} \\ &7.4V - 1.6V \\ &= 5.8V \end{aligned}$$

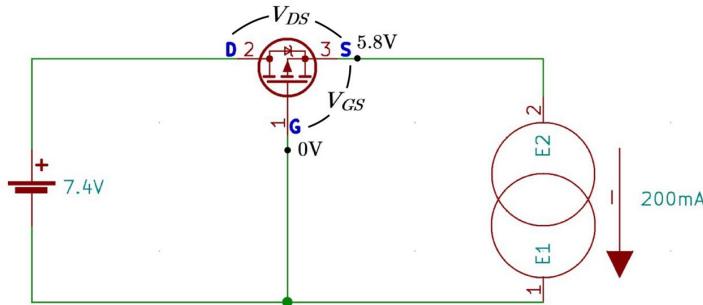


Figure 33: Spanningen bij juiste aansluiting van batterij

Met deze informatie kunnen we dan de spanning over de gate en de source berekenen:

$$\begin{aligned} V_{GS} &= V_G - V_S \\ 0V - 5.8V \\ &= -5.8V \end{aligned}$$

We bekomen dus een gate-source spanning van -5.8V. Dit betekent dus dat:

$$V_{GS} < V_{GS_{th}} \Leftrightarrow -5.8V < -4V$$

M.a.w. de mosfet zal in doorlaat zijn als de batterij juist wordt aangesloten. Het voordeel van het gebruiken van een mosfet in deze situatie is dat eens dat de mosfet in doorlaat staat, de mosfet een zeer lage $R_{DS(on)}$ heeft van 0.117Ω . Dit betekent dat de weerstandswaarde van de diode tussen drain en source wanneer de mosfet in doorlaat is, bijna 0 bedraagt en dus zeer efficiënt is. Ook deze waarde kunnen we terug vinden in de datasheet van de IRF9540N:

| | | | | | | | |
|--------------|--------------------------------------|---|---|-------|----------|-------------------------------|--|
| $R_{DS(on)}$ | Static Drain-to-Source On-Resistance | — | — | 0.117 | Ω | $V_{GS} = -10V, I_D = -11A$ ④ | |
|--------------|--------------------------------------|---|---|-------|----------|-------------------------------|--|

Figure 34: Weerstandswaarde tussen drain en source wanneer IRF9540N in doorlaat is [17]

We kunnen dezelfde redenering maken wanneer de batterij fout wordt aangesloten. De gate spanning is nog steeds 0V omdat deze, zoals hiervoor, verbonden is met ground.

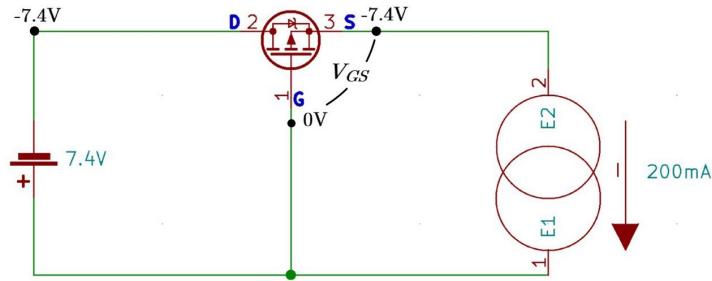


Figure 35: Spanningen bij foute aansluiting van batterij

Stel dat de mosfet toch aan zou staan in deze situatie. Doordat er $R_{DS(on)}$ zeer laag is kunnen we voor deze redenering de spanningsval even weg laten. Hierdoor wordt de source spanning gelijk aan -7.4V. Als we dan opnieuw de gate source spanning berekenen zoals hierboven dan krijgen we:

$$\begin{aligned} V_{GS} &= V_G - V_S \\ 0V - (-7.4V) \\ &= 7.4V \end{aligned}$$

En dus:

$$V_{GS} < V_{GS_{th}} \Leftrightarrow 7.4V < -4V \Rightarrow \text{klopt niet!}$$

Dit betekent dat de mosfet nooit aan kan staan in deze situatie omdat de gate source spanning groter is dan de threshold waarde van de mosfet. M.a.w. de initiële stelling dat de mosfet aan zou staan in deze situatie klopt niet!

5.4 Servo Circuit

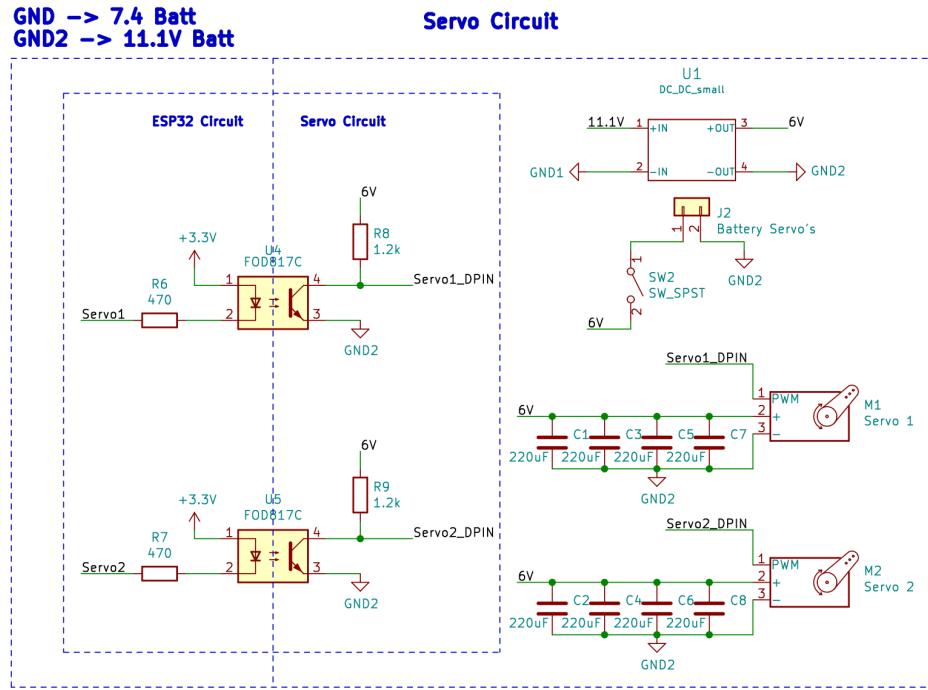
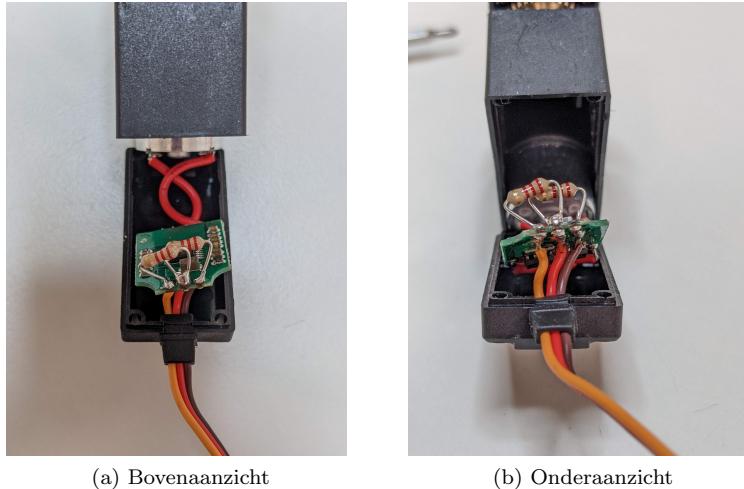


Figure 36: Servo Circuit

Het servo circuit bestaat grotendeels uit 2 onderdelen. Het logische gedeelte en het voeding gedeelte. Deze 2 worden optisch gescheiden d.m.v. optocouplers. Verder bevatt het schema een buck converter om van 11.1V naar de gewenste 6V voor de servo's te gaan.

5.4.1 Servo's

Tot hertoe hebben we altijd gezegd dat we servo's gebruiken voor het besturen van de robot. Dit klinkt misschien als een speciale aanpak aangezien een servo gebruikt wordt voor precieze bewegingen tussen 0° en 180° . We hebben gebruik gemaakt van servo's aangezien deze eigenlijk compacte DC motoren zijn met ingebouwde H-bruggen en een eenvoudige gearbox. Door de servo's te modifieren kan je ze continu doen bewegen en heb je eigenlijk een PWMbare DC motor. Hiervoor hebben we de potentiometer verwijderd en deze vervangen door een spanningsdeler. Door de weerstanden zodanig te kiezen dat de uitgang spanning van de spanningsdeler altijd de helft is van de voeding spanning, zal het controle circuit denken dat de feedback altijd op 90° staat. We hebben de originele potentiometer van servo gemeten en deze had een waarde van $3.6\text{ k}\Omega$. Hierdoor hebben we gekozen voor $2 \times 1.8\text{ k}\Omega$ weerstanden. We hebben de potentiometer los gesoldeerd en deze vervangen door de 2 weerstanden tussen de juiste pinnen. Het probleem met deze methode is dat er een marge zit op de weerstandswaarden. Hierdoor kan het zijn dat de neutrale positie niet exact op 90° ligt. Bij ons ligt deze op 100° .



(a) Bovenaanzicht

(b) Onderaanzicht

Figure 37: Spanningsdeler Gemodificeerde MG996R

De gebruikte servo's zijn de MG996R. Dit is een redelijk compacte maar stevige servo door de metalen tandwielen. Volgens de datasheet kan de MG996R gevoed worden tussen de 4.8V en 7.2V. Vermits we een buck converter gebruiken kunnen we vrij kiezen welke spanning we gebruiken. We hebben gekozen om de servo op 6V te voeden omdat alle specificaties in de datasheet van de MG996R hierop zijn afgesteld. Bovendien zal de servo meer kracht kunnen leveren indien gevoed op 5V. Ook hebben we bij het testen gemerkt dat wanneer we de servo voedden op 7.2V, er zeer hoge piekstromen waren bij het aansturen van de servo's.

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA - 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 µs
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

Figure 38: Specificaties MG996R [16]

Om deze piekstromen tegen te gaan hebben we capaciteiten ($4 \times 220\mu\text{F}$) geplaatst tussen de + en - pinnen van de servo. Bij het opstarten van de servo's kunnen ze een zeer hoge stroom vragen. Ook bij het gewoon besturen van de servo's trekken de servo's veel stroom in korte pieken. De capaciteiten werken als een buffer, ze zorgen ervoor dat bij het opstarten en het besturen van de servo, de voeding de gemiddelde stroom levert en de capaciteiten de stroom pieken afvlakken. Dit zorgt ervoor dat de servo's soepeler werken en de stroom pieken de batterij niet beschadigen. De gemiddelde stroom die de servo trekt op 6V is rond de 500mA - 900mA. Vermits de gebruikte voeding een 11.1V 3C batterij is, zal deze zoals eerder besproken de benodigde stroom kunnen leveren.



Figure 39: MG996R Servo [18]

5.4.2 DC-DC Step Down Buck Converter

Om van 11.1V naar 6V te gaan hebben we gebruik gemaakt van een DC-DC step down buck converter. Specifieker de LM2596 module. Deze module is zoals de naam zegt, gebaseerd op de LM2596. De module voegt een potentiometer toe waarmee je de uitgang spanning manueel kan regelen. Volgens de datasheet kan de ingang spanning tussen de 4V en 40V liggen, de uitgang spanning tussen de 1.25V en 35V met een maximum uitgang stroom van 3A [19]. Dit voldoet dus ook aan onze toepassingen.

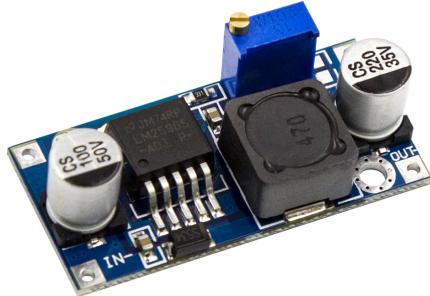


Figure 40: LM2596 DC-DC Buck Converter Module [20]

Maar waarom geen lineaire voltage regulator gebruiken? Buck converters zijn veel efficiënter in vergelijking met een lineaire voltage regulator zoals een LD1117V50 of MCC7805CT die we in het FabLab vinden. Stel dat we de 11.1V van de batterij willen transformeren naar 6V en dat we 2 servo's gebruiken die te samen 2A verbruiken, dan kunnen we de vermogensdissipatie als volgt berekenen:

$$\begin{aligned}
 P_{verloren} &= (V_{in} - V_{out}) * I_{out} \\
 &= (11.1V - 6V) * 2A \\
 &= 10.2W
 \end{aligned}$$

Zeker bij grote projecten zal een grote vermogensdissipatie die omgezet wordt in warmte tot problemen leiden, zoals het beschadigen van de voltage regulator. Daarom is het een goede

gewoonte om bij projecten waar je hoge efficiëntie, lage warmte productie en "hoge" spanningen gebruikt, geen lineaire voltage regulator te gebruiken.

5.4.3 Optocouplers

Bij sommige toepassingen is het noodzakelijk dat circuits geïsoleerd zijn terwijl deze wel nog met elkaar moeten interageren. Bij het besturen van de servo's via een micro controller is het belangrijk dat deze 2 circuits gescheiden zijn. Stel dat we de servo voeden via de 5V pin van de micro controller, dan zouden de grote piekstromen van de servo's spanningsverschillen opwekken. Vermits de servo rechtstreeks in verbinding staat met de micro controller kunnen deze piekstromen en spanningsverschillen de micro controller sterk beïnvloeden. Hierdoor is het een goede gewoonte om componenten die de logica van het systeem kunnen beïnvloeden zoals servo's, te isoleren. In het geval van een optocoupler gebeurt dit optisch. We hebben gebruik gemaakt van de FOD817C. Voor onze toepassing willen we de logische 5V dan de ESP32 en de 6V voeding en de data-lijnen van de servo's gescheiden houden. Omdat we 2 batterijen gebruiken zijn de voedingslijnen al gescheiden. Om het micro controller circuit en het servo circuit volledig te scheiden moeten dus we de datalijnen van de servo's en de ESP32 scheiden.

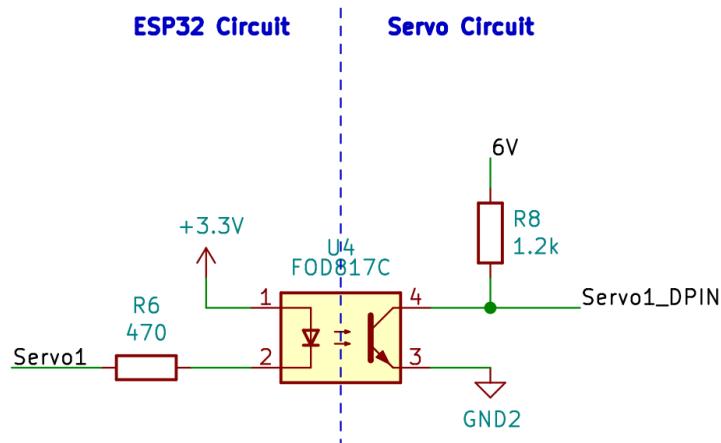


Figure 41: Schema Optocoupler

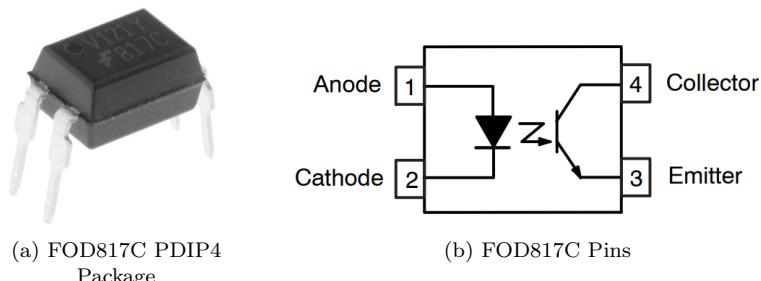


Figure 42: FOD817C [21]

Het idee van dit circuit is dat een logische hoog van de micro controller overeenkomt met een

logische hoog aan de servo's kant. We kunnen dus 2 staten onderscheiden: een logisch hoog en een logisch laag signaal van de micro controller. Wanneer de micro controller een logische hoog op cathode van de LED zet zal er geen spanningsval over de LED zijn vermits de anode verbonden is met 5V en dus altijd logisch hoog staat. Hierdoor zal de LED niet branden en dus als gevolg zal de foto-gevoelige sensor ook niets ontvangen. De emitter van de detector is verbonden met ground en de collector is via een pull-up weerstand verbonden met 6V. De collector zal dus op hoog staan als de detector niets ontvangt. De data lijn van de servo is ook verbonden met de collector. Dit betekent dus dat wanneer de micro controller een hoog signaal geeft, de LED niet gaat branden, de detector niets ontvangt en dus de data lijn van de servo op hoog zal staan door de pull-up weerstand.

Als de micro controller daarentegen een logisch laag signaal op de cathode van de LED zet zal er wel een spanningsval zijn over de LED en dus zal deze branden. De detector zal dit licht ontvangen en bijgevolg in doorlaat staan. Hierdoor zal de stroom door de weerstand, via de collector en emitter naar ground vloeien. De data lijn van de servo zal logisch laag staan.

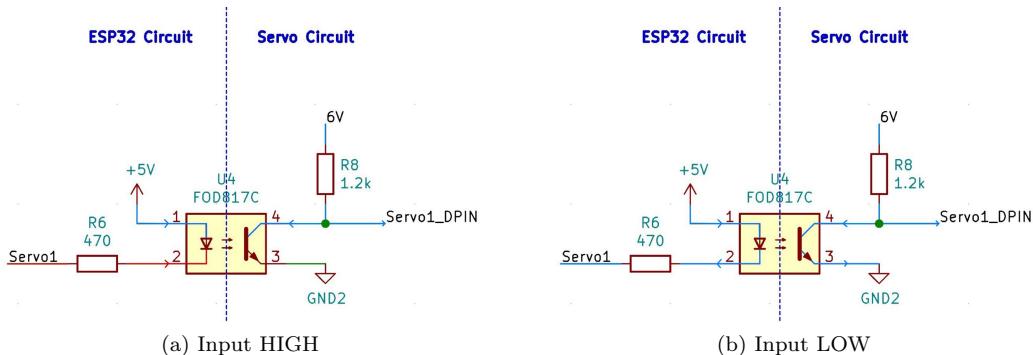


Figure 43: Staten Optocoupler

| Input | LED/Detector | Output |
|-------|--------------|--------|
| HIGH | OFF - OFF | HIGH |
| LOW | ON - ON | LOW |

Table 1: Waarheidstabell Optocoupler Circuit

Voor het berekenen van de weerstandswaarde voor de LED moeten we naar de voorwaartse spanningsval van de LED kijken. Dit kunnen we terug vinden in de datasheet:

| VF | Forward Voltage | FOD814 | IF = ±20 mA | - | 1.2 | 1.4 | V |
|----|-----------------|--------|-------------|---|-----|-----|---|
| | | FOD817 | IF = 20 mA | - | 1.2 | 1.4 | |

Figure 44: Voorwaarde Spanningsval FOD817C LED [21]

Deze bedraagt dus typisch 1.2V bij een voorwaartse stroom van 20mA. De weerstandswaarde wordt dan:

$$R_{LED} = \frac{V_{ESP32} - V_F}{I_F} = \frac{5V - 1.2V}{0.02A} = 190\Omega$$

We hebben een meer voorkomende weerstandswaarde gekozen van $470\ \Omega$. De waardes van pull-up weerstanden liggen meestal ergens tussen de $1\ k\Omega$ en $10\ k\Omega$. De $1.2\ k\Omega$ waarde is eerder tegen de kant van de sterk pull-up weerstanden omdat we zeker willen zijn dat de data lijn van de servo hoog is als deze hoog moet zijn.

5.5 Sensors

5.5.1 Absolute Orientatie Sensor

De robot is voorzien van een 3-in-1 accelerometer, gyrometer en magnetometer sensor of korter een IMU. Hiermee kan de robot tegelijkertijd zijn tangentiële versnelling en hoeksnelheid meten als ook de sterkte van het magnetisch veld. Via deze 3 informatie bronnen kan de robot zijn absolute positie weten en beslissingen maken. We hebben de robot niet geprogrammeerd om autonoom te bewegen maar dit is wel mogelijk via deze sensor, deze is voorzien voor eventuele toekomstige toepassing.

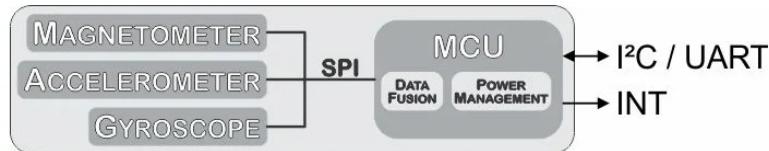


Figure 45: BNO055 Schema [22]

De sensor in kwestie is de BNO055. Naast de 3 sensoren bevat de module ook een microcontroller. Met als functies te filteren, de data te combineren en door te sturen naar de gebruiker via I²C of UART. Bij het opstarten kalibreert de sensor zich. Hierna zal elke verandering tegenover de gekalibreerde waarden weergegeven worden. De waardes die vooral nuttig zijn voor het optimaliseren van de besturing zijn de yaw, pitch en roll waardes. Wanneer we deze uitlezen en plotten in de Serial Monitor van de Arduino krijgen we het volgende:

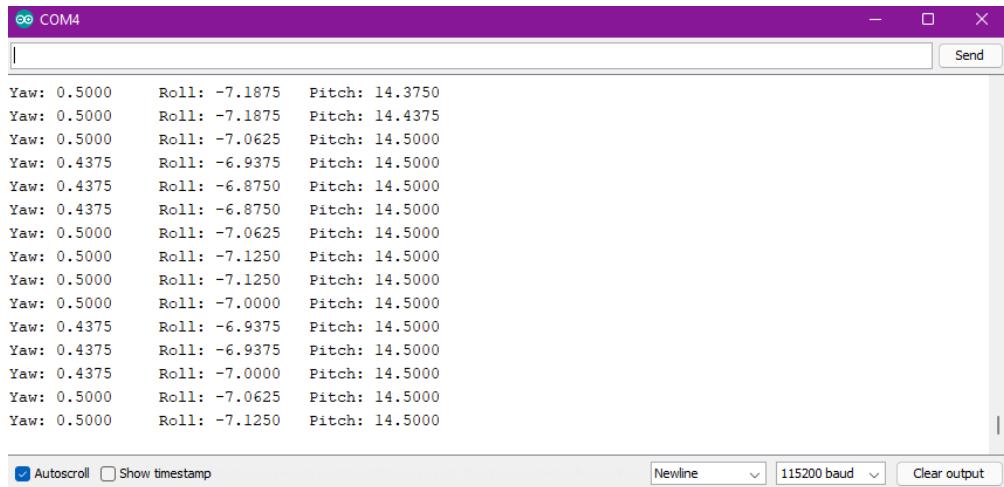


Figure 46: BNO055 Serial Monitor

We de 3 waardes ook bekijken op de Serial Plotter:

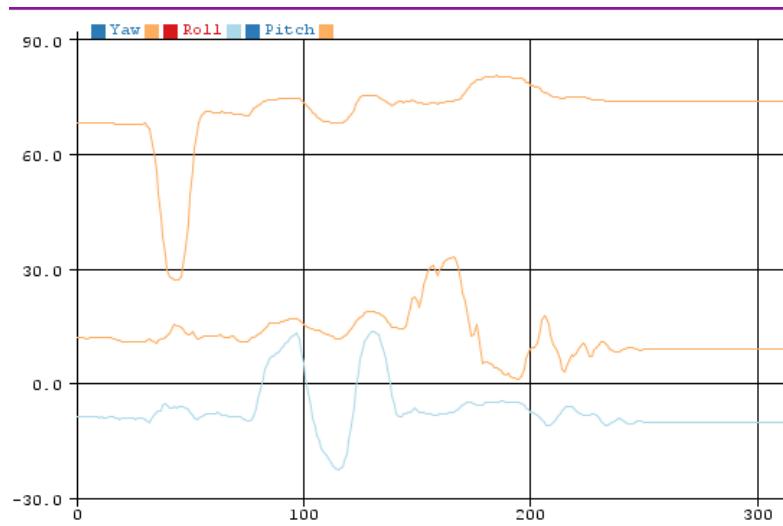


Figure 47: BNO055 Serial Plotter

Dit geeft een beter beeld van het verloop van de verschillende waardes. Bij het constant monitoren van de waardes terwijl de robot beweegt, constateren we een verwachte verandering van de waardes. De BNO055 is accuraat tot op 1 graden. Voor onze toepassing is dat voldoende. Ook is de leessnelheid voldoende. De robot zou dus eventueel bestuurd kunnen worden via de BNO055. Het probleem hierbij is dat de bewegingen van onze robot niet accuraat zijn en deze dus een bottleneck vormen voor het autonoom besturen van de robot. Wat wel een oplossing zou kunnen zijn is de sensor gebruiken bij het stabiliseren van de robot. Zo kan de robot a.d.h.v. de sensor zich terug naar de gekalibreerde waardes brengen.

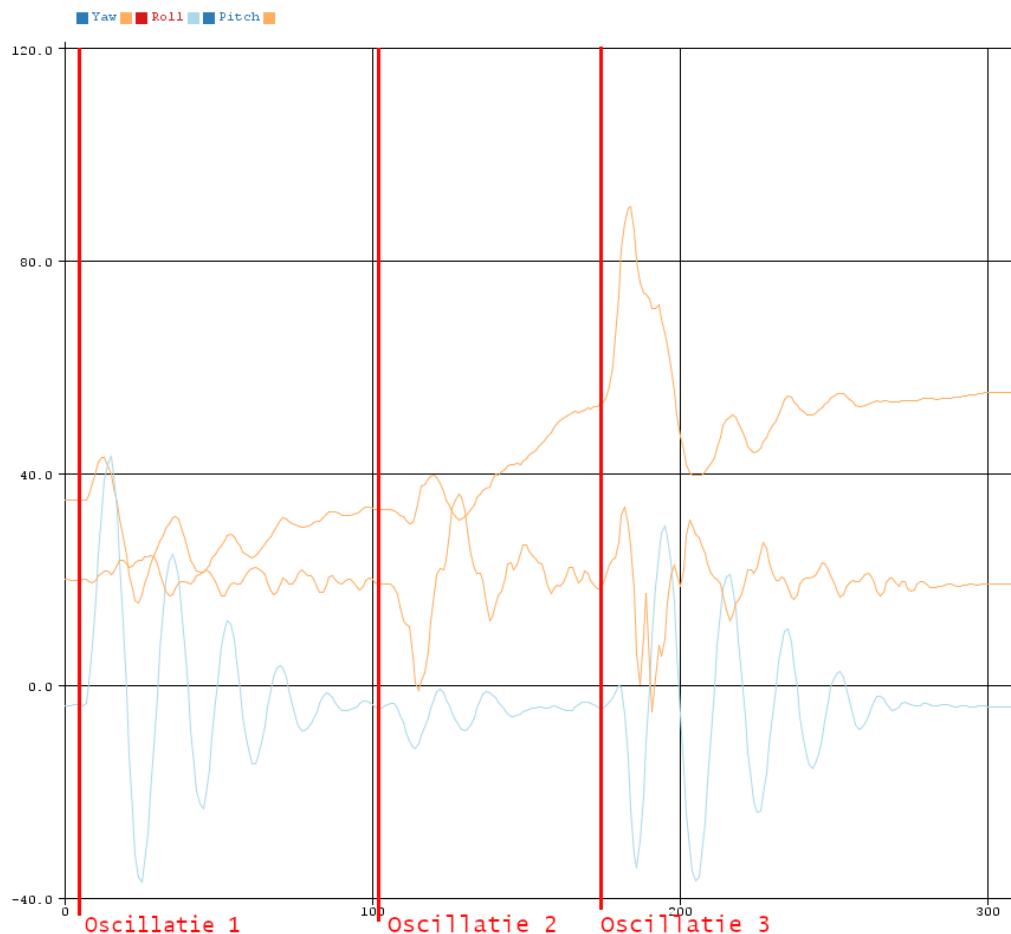


Figure 48: BNO055 Serial Plotter bij Stabilisatie

Hierboven zijn de 3 waarden geplot op de Serial Plotter bij het loslaten van de robot op een willekeurige positie, dit werd 3 keer herhaald bij elke stabilisatie. We zien duidelijk de oscillaties van de roll en pitch waarden bij het stabiliseren van de robot.

5.6 Besturing

Voor het besturen van de robot maken we gebruik van Dabble. Dit is een app dat geïnstalleerd kan worden op een smartphone. Dabble bevat verschillende functies die via Bluetooth communicerbaar zijn met de ESP32. Wij gebruiken van de gamepad:

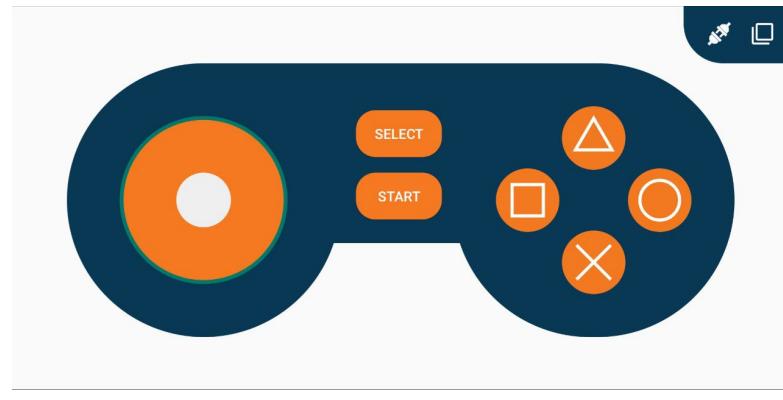


Figure 49: Dabble Gamepad

Wij maken enkel gebruik van de joystick om de robot te bewegen. Het eerste icoon dat rechtsboven staat dient om de verbinding met de ESP32 in stand te krijgen. Eens dat de smartphone en de ESP32 verbonden zijn, kan je signalen doorsturen. Naast de app moeten er in code ook enkele libraries ingeladen worden. Door de app te gebruiken is het zeer eenvoudig om de ESP32 draadloos te besturen, dit is de reden waarom we kozen voor de app in combinatie met een ESP32.

6 Ondervonden Problemen

6.1 IO Pinnen ESP32

Een probleem waar we redelijk wat tijd mee verloren hebben, is dat sommige IO pinnen van de ESP32 geen servo's kunnen besturen. Uit ervaring met de ESP32 dev kit V4 weten we dat IO pinnen 12 en 13 compatibel zijn voor het besturen van een servo. Het probleem was dat we gebruik maakten van verschillende soorten ESP's. Bij het testen op breadbord maakten we gebruik van de ESP32 dev kit V4. Pas bij het maken van het protoboard hebben we gebruik gemaakt van de NodeMCU ESP32, vermits de levering dan pas was toegekomen. Hiermee dachten we dat het gebruiken van IO pinnen 12 en 13 hetzelfde zou zijn voor de beide versies van de ESP32. Bij het testen van het protoboard bleek dit dus niet het geval. IO pin 12 werkt niet voor het besturen van een servo op de NodeMCU ESP32. Het interessante was dat de beide servo's niet wouden werken, dus ook pin 13 niet. Bij het testen van het protoboard zijn we dan met een multimeter elk deel van het circuit afgegaan om te zien waar we onverwachte signalen te zien kregen. We constateerden dat IO pin 12 constant laag was en pin 13 constant hoog. Achteraf gezien hadden we beter gebruik gemaakt van een oscilloscoop i.p.v. een multimeter voor het nakijken van de signalen. Uiteindelijk hebben we dus gebruik gemaakt van IO pin 14 i.p.v. 12. Hierna werkten zowel IO pin 13 als pin 14 en konden we de servo's aansturen.

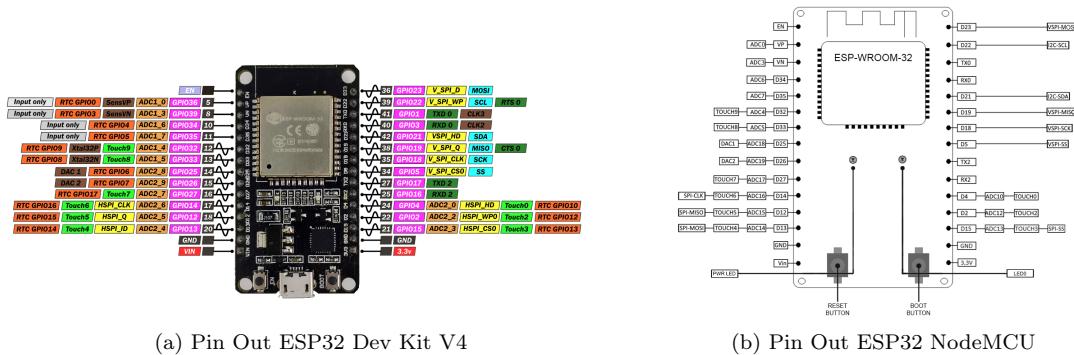


Figure 50: Pin outs gebruikte ESP32's [23]

IO pinnen 12 en 13 lijken nochtans gelijkaardig op beide ESP32's. Net zoals pin 12 en 14 gelijkaardig lijken op de NodeMCU ESP32. We begrijpen dus nog steeds niet echt waarom pin 12 niet werkt maar pin 14 wel.

6.2 Keuze 2 Batterijen

Zoals eerder aangehaald maken we gebruik van 2 batterijen. De reden hiervoor is wat we noemen "Brown Out" van de microcontroller. Dit gebeurt wanneer de voedingsspanning van de microcontroller tijdelijk onder de nodige voedingsspanning van de microcontroller ligt. Hierbij kan het zijn dat de microcontroller onverwacht gedrag vertoont of dat er schade aangericht wordt. De microcontroller zal zichzelf resetten om dit te voorkomen. Om dit te doen bevat de microcontroller een zogenaamd Brown-Out Detection circuit of BOD. Dit circuit maakt gebruik van een comparator om de inkomende voedingsspanning te vergelijken met de nodige voedingsspanning. Op figuur 51 staat de voedingsspanning afgebeeld i.f.v. de tijd. V₁ is de normale voedingsspanning, V₂ is het punt waar de microcontroller onverwacht gedrag kan vertonen en V₃ is het punt waar de microcontroller stopt met werken. Tussen V₂ en V₃ is de gevaren zone.

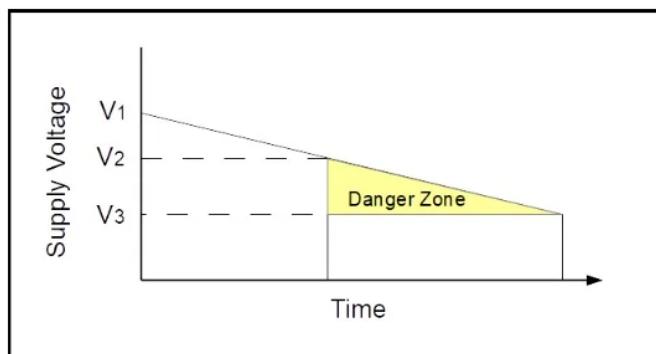


Figure 51: Brown Out: Spanning i.f.v. de tijd [24]

We zijn dit probleem tegen gekomen bij het testen van het protoboard. Wanneer we de servo's aanstuarden, draaiden ze voor ongeveer 2 tot 3 seconden. Daarna stoppen ze met draaien. Ook bij het veranderen van de draai-richting stoppen ze met draaien na 2 tot 3 seconden. Na wat research kwamen we uit op het Brown Out fenomeen. Dit kwam volledig overeen met wat we waarnamen. Wanneer de servo's beginnen met draaien en zeker bij het veranderen van de draairichting trekken de servo's zeer veel stroom uit de batterij waardoor de voedingsspanning tijdelijk zakt. De spanning in de batterij zal zakken door de interne en inherente weerstand van de batterij. Hierdoor kreeg de ESP32 tijdelijk niet genoeg voedingsspanning, waardoor nadat het BOD circuit dit detecteerde, de ESP32 zichzelf resette.

We hebben meerdere dingen geprobeerd om dit probleem op te lossen. Een capaciteit ($470\mu F$) plaatsen tussen de voeding en ground van de ESP32. Een capaciteit ($470\mu F$) plaatsen tussen de enable pin en ground van de ESP32 zodat er meer tijd zou zijn bij boot en de ESP32 deftig kan opstarten zonder dat hij direct weer moet resetten. Gebruik maken van acceleratie en de-acceleratie voor het aansturen van de servo's zodat ze niet instant een grote stroom vragen maar geleidelijk aan meer en meer stroom vragen. Deze acceleratie en de-acceleratie kan puur in code geïmplementeerd worden d.m.v. een for-loop met daarin een kleine delay die telkens de snelheid verhoogt. We hebben ook de grotere MCU Power batterij proberen gebruiken maar zelf dan bleef het probleem zicht voortdoen. Hierdoor hebben we beslist om gebruik te maken van 2 aparte voedingen. 1 kleinere batterij voor de ESP32 en 1 grotere batterij met genoeg capaciteit voor de servo's. Hierdoor beïnvloeden de stroom pieken van de servo's de ESP32 niet. Bij het testen met 2 batterijen werkte dit goed en konden we de servo's besturen in de gewenste richting en snel

de richting veranderen zonder problemen. Het gebruik van 2 batterijen voor onze toepassing is eigenlijk nuttig gewicht dat we kunnen moduleren. Hierdoor is het gebruiken van 2 batterijen in onze toepassing zeker geen nadeel.

6.3 Extra Binnenring

Er zijn twee hoofdredenen voor het plaatsen van een extra binnenring. De eerste reden is dat het originele concept waarbij we de tandwielen van de servo's laten knellen in de ring waardoor de robot zou rollen werkt niet. Dit is grotendeels te wijten aan de verbinding tussen de kleine tandwielen en de servo's. Deze verbinding was niet sterk genoeg, waardoor de verbinding brak op het moment dat de tandwielen gekneld werden in de ring. Ze braken voordat ze genoeg gekneld werden om te roteren. De tweede reden is dat de servo's niet dezelfde rotatie snelheid hebben, waardoor de verbinding tussen de 2 servo's na enkele rotaties niet meer om zijn als draaide maar asymmetrisch draaide. Dit had als gevolg dat de tandwielen blokkeerden en er geen rotatie meer mogelijk is.

Door deze 2 hoofdredenen hebben we in overleg met onze promotores beslist om een extra binnenring te plaatsen. Dit zorgt ervoor dat de tandwielen geblokkeerd zitten wanneer ze beiden in dezelfde richting gaan en dus rotatie als gevolg heeft waardoor de robot kan rollen. Bovendien houdt deze ring de servo's en tandwielen grotendeels gedwongen om op 180° te blijven.

6.4 Stijfheid Constructie

Voor het verbinden van de 2 servo's maken we gebruik van 4 bouten. In de eerste versies (figuur 52) van het prototype gebruikten we de bovenste paar stangen om de elektronica op te bevestigen. De onderste paar stangen werden gebruikt om de batterij aan te bevestigen. Bij het testen van deze constructie merkten we al snel dat de servo's onderling te veel speling hadden. Dit is dus een gevolg van een gebrek aan stijfheid van de constructie. De 4 stangen waren onderling niet genoeg verbonden waardoor er torsie kon optreden.

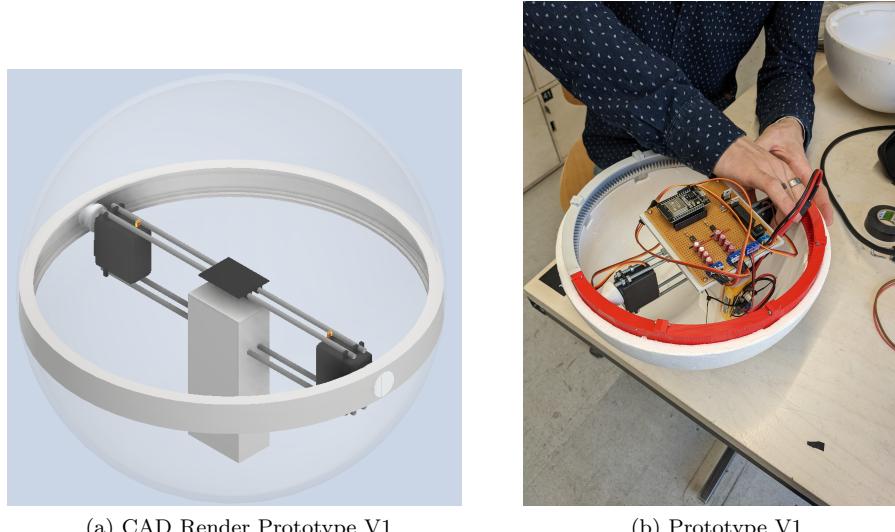


Figure 52: Prototype Robot V1

Om dit op te lossen moesten we dus de 4 stangen onderling zien te verbinden om de stijfheid te vergroten. Vervolgens hebben we een middenstuk ontworpen waardoor de 4 stangen lopen, het protoboard bovenop zit en de batterijen aan de onderkant. Dit middenstuk verhoogt de stijfheid van de robot vermits de 4 stangen onderling verbonden zijn. Ook hebben we nu een centrale plaats waar de elektronica en batterijen dicht bij elkaar zitten. Hiernaast is het ook belangrijk om al het gewicht van de elektronica en de batterijen te centreren in het midden van de robot. Het toevoegen van het middenstuk brengt dus veel voordelen met zich mee.

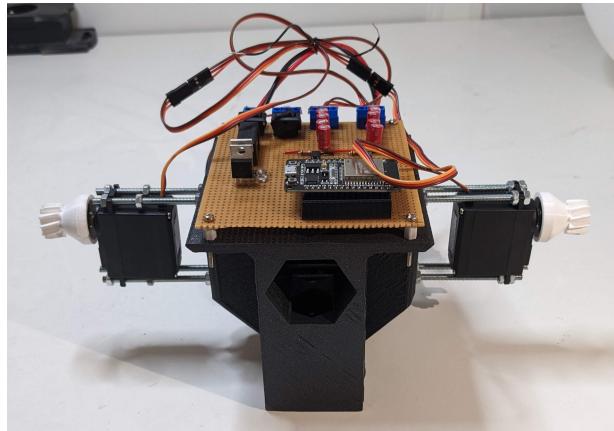


Figure 53: Middenstuk V2

Hiernaast was de verbinding tussen de servo en het tandwiel niet sterk genoeg. De afstand tussen de tandwielen en de servo was te groot waardoor er te veel stress op de verbinding stond. Voor de effectieve verbinding maakten we gebruik van de meegeleverde opzetstukken van de servo. Het voordeel hiervan is dat de stukken perfect op het aanzet stuk van de servo passen. Het nadeel is dat deze enkel d.m.v. frictie erop vast zitten. Bij het minste dat de servo's geblokkeerd werden, brak de verbinding tussen de tandwielen en de servo's.

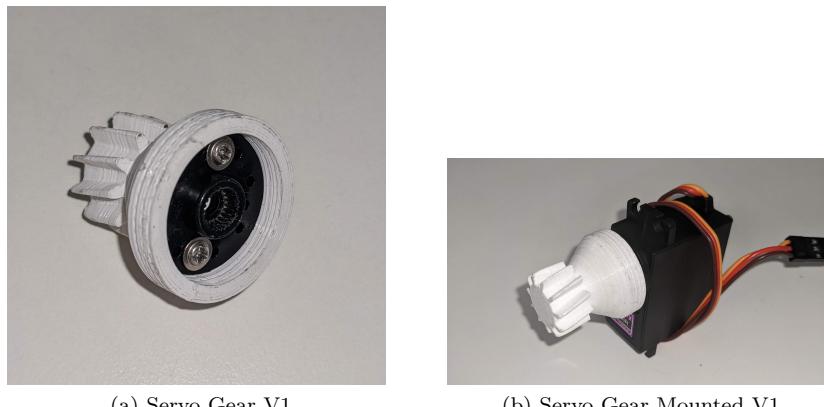


Figure 54: Servo Gear V1

Om dit op te lossen hebben we de hele verbinding hermaakt. Er is plaats voorzien voor de lager van de binnenring en de afstand tussen het tandwiel en de servo is kleiner. Voor de

verbinding maken we gebruik van zowel het tandwiel op de servo, als de schroefdraad binnendit tandwiel. D.m.v. een bout door het tandwiel te gebruiken, hebben we een sterkere verbinding. Hieronder foto's van de finale versie van het tandwiel en middenstuk.

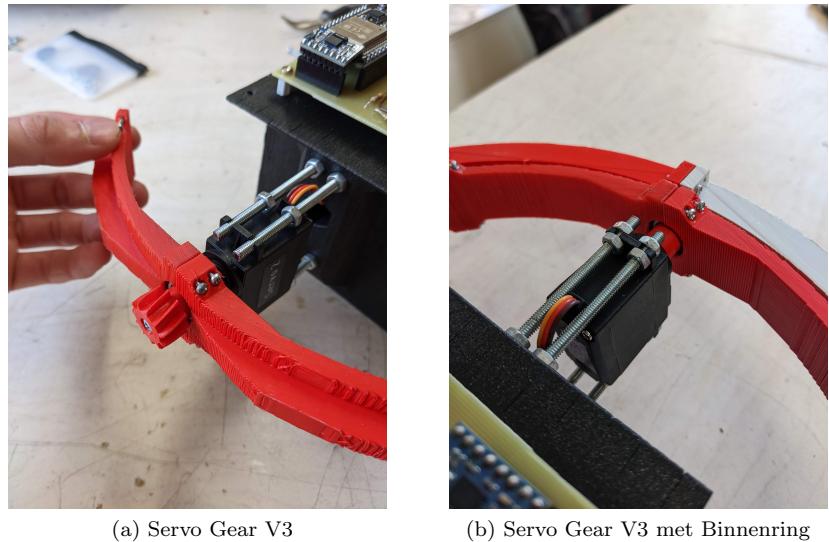


Figure 55: Servo Gear V3

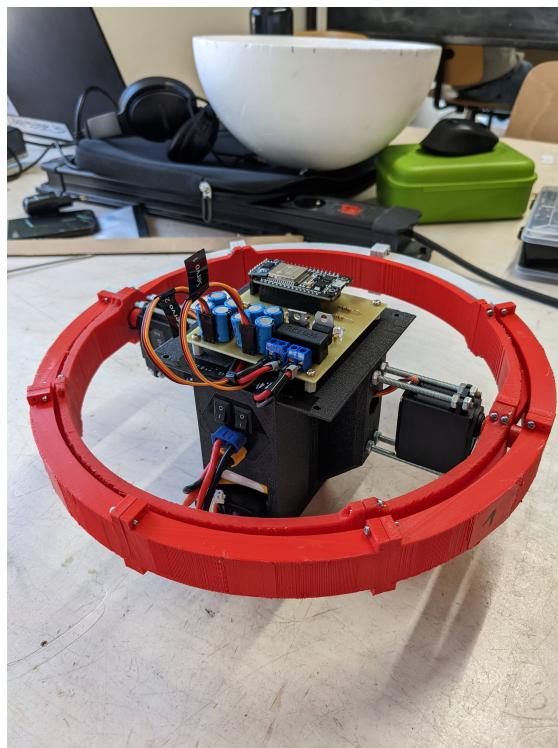


Figure 56: Zijaanzicht Middenstuk V3

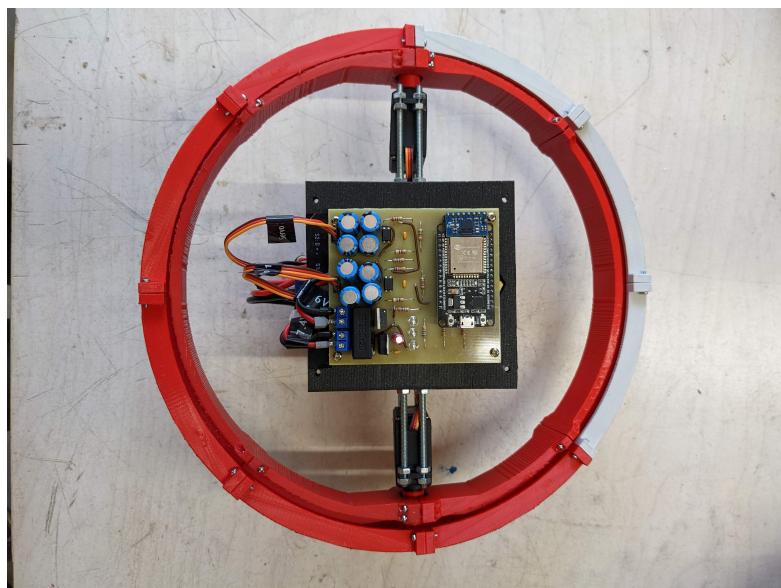


Figure 57: Bovenaanzicht Middenstuk V3

6.5 Servo's

Een bron van de meeste problemen die we zijn tegengekomen zijn de servo's. Zoals eerder aangehaald hebben we gebruik gemaakt van servo's voor verschillende redenen. Deze voordelen wegen grotendeels op tegen de nadelen maar de nadelen blijven wel significant groot genoeg. Het eerste nadeel is dat de goedkope hobby servo's redelijk inaccuraat zijn. Ze kunnen wel accuraat tussen 0 en 180 graden bewegen, maar de verschillen tussen servo's van hetzelfde model zijn merkbaar. Dit is een van de redenen dat we in de code, het verschil in snelheid tussen de 2 servo's manueel moeten aanpassen. De servo's, al zijn ze technisch gezien hetzelfde, verschillen nog steeds in hun performantie. Dit verschil is ook te wijten aan het modifieren van de servo's om ze continu te doen laten draaien. Dit is iets waarvan we op voorhand wisten dat dit een invloed zou hebben. Het gevolg van het snelheidsverschil is dat wanneer de robot rond zijn as draait, de binnenring na een tijdje gekneld raakt. Omdat de ene servo sneller gaat dan de andere zal de rotatie niet meer perfect rond het centrum van de robot draaien. Omdat de binnenring de servo's beperkt probeert de houden op 180° zal deze gekneld raken met de buitenring.

Een mogelijke oplossing voor het snelheidsverschil is om een feedback lus te creëren dat de snelheden van de servo's gaan meten en aanpassingen maakt waar nodig. Een Hall sensor of een rotary encoder zouden een oplossingen kunnen zijn. Het probleem hiermee is dat we gemerkt hebben dat de snelheden van de servo's niet lineair zijn. Wat we hiermee bedoelen is dat een snelheidsverandering dichter bij het neutrale punt van de servo een grotere impact zal hebben op de snelheid van de servo dan bij de uiteinden van het interval. Zo zal een verandering van 1550ms \Rightarrow 1600ms een grotere verandering in snelheid teweeg brengen dan bijvoorbeeld een verandering van 1800ms \Rightarrow 2000ms. Vermits deze veranderingen niet lineair zijn, is het moeilijk om een feedback lus te gebruiken die aanpassingen maakt aan de input snelheid van de servo terwijl we niet weten hoe deze snelheidsveranderingen zich gedragen.

6.6 Design

Een van de hoofdredenen waarom de robot niet accuraat rolt is door de keuze van het design. Omdat we met een ring werken die op de evenaar van de sfeer zit en mee beweegt tijdens het rollen is het moeilijk om bij sommige hoeken een rotatie uit te voeren. Als de ring bijvoorbeeld verticaal staat zal het moeilijk zijn om de gewenste rotatie uit te voeren. Hiernaast rolt de robot niet in een rechte lijn. We hebben opgemerkt bij het testen dat dit komt doordat de massa van de robot niet homogeen verdeeld is. Het gewicht is nu niet gecentreerd genoeg. Ideaal zou het gewicht uniform verdeeld zijn over de hele oppervlakte aan de binnenkant van de onderste sfeer. Ook is er heel wat oscillatie na het stoppen met rollen. Het duurt zeker een aantal seconden eer dat de robot gestopt is met oscilleren en op een vaste plaats blijft staan. Dit is ook te wijten aan de niet uniforme verdeling van de massa.

7 Performantie/Test

7.1 Algemene Specificaties

- Afmetingen: D=30cm
- Totaal Gewicht: 1004.2g
- Rolsnelheid: 0.062 deg/s
- Rotatiesnelheid: 1.25 m/s

7.2 Accuraatheid Test

Om een idee te krijgen over hoe accuraat de robot kan bewegen hebben we een test uitgevoerd. De robot moet altijd hetzelfde punt bereiken door dezelfde bewegingen uit te voeren. De test bestaat uit 2 bewegingen. Eerst een rotatie naar links en vervolgens rollen om het gewenste punt te bereiken. Het test parcour ziet er als volgt uit:

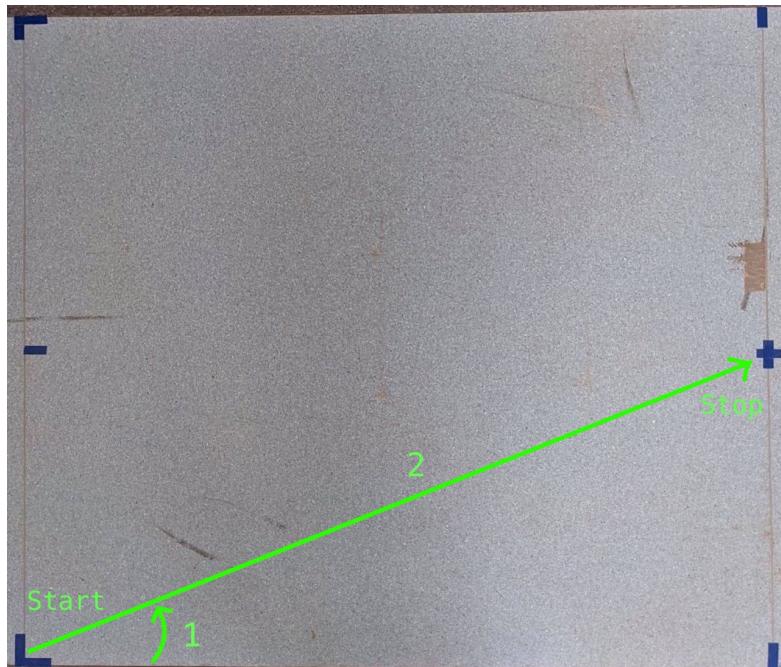


Figure 58: Test Veld

De dimensies van het test veld zijn: 1.8m x 1.8m. De robot start op het punt (0m;0m). Vervolgens maakt de robot een rotatie van 26.57° . Hierna moet de bal 2m rollen om het eindpunt op coördinaat (1.8m;0.9m) te bereiken. Om ervoor te zorgen dat de robot altijd dezelfde beweging zou maken en we effectief de robot zijn accuraatheid meten hebben we de bewegingen vastgelegd in code. Volgens de gemeten rol-en rotatie snelheid van de robot zouden we theoretisch 2.45s moeten roteren en 3.63s moeten rollen. In de praktijk was dit een groot verschil. We zijn de beste resultaten bekomen met een rotatie duur van 3.20s en een rol duur van 8.00s.

| Test | Resultaat (X, Y) in cm | Afstand tot Eindpunt in cm |
|------|------------------------|----------------------------|
| 1 | 151; 163 | 78.55 |
| 2 | 162; 72 | 25.46 |
| 3 | 113; 157 | 94.75 |
| 4 | 143; 132 | 55.97 |
| 5 | 175; 43 | 47.27 |
| 6 | 131; 124 | 59.64 |
| 7 | 178; 81 | 9.22 |
| 8 | 146; 145 | 64.66 |
| 9 | 128; 146 | 76.42 |
| 10 | 163; 105 | 22.67 |

Table 2: Resultaten Test

We hebben deze test 10 keer uitgevoerd. Telkens de robot op zijn start positie gezet met de juiste oriëntatie, de robot geactiveerd en genoteerd waar hij terecht kwam. Dit zijn de resultaten:

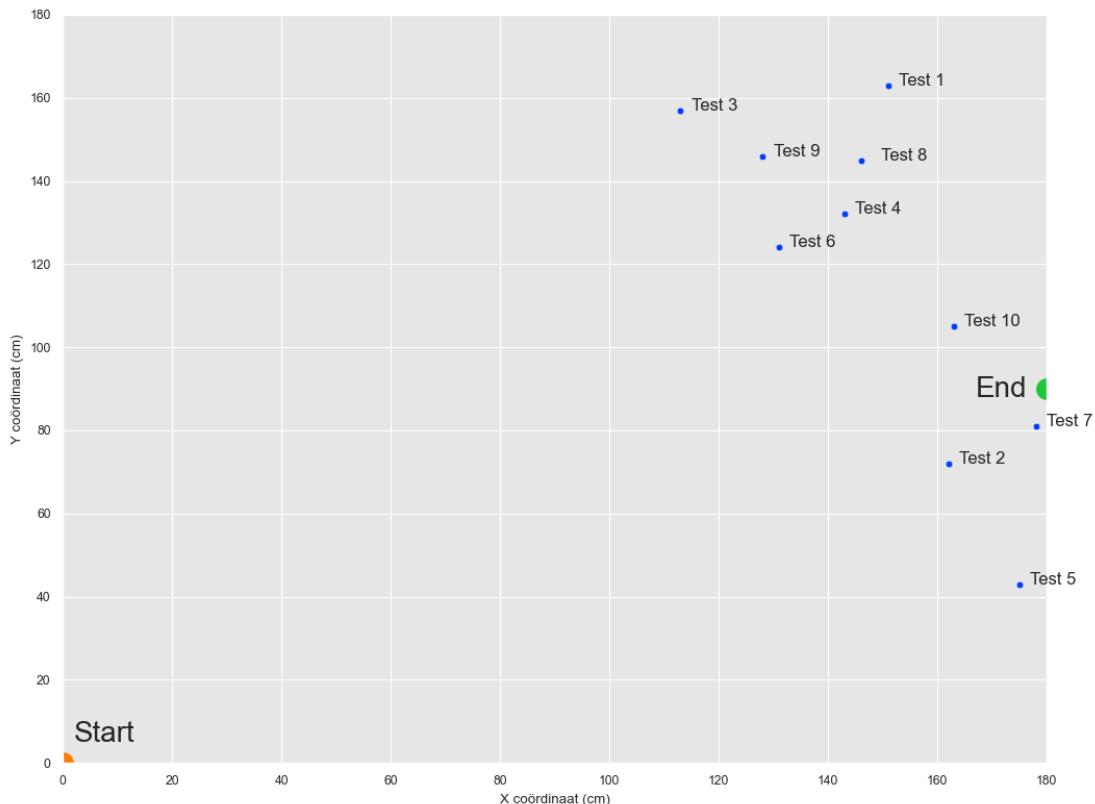


Figure 59: Resultaten Test Plot

Vervolgens kunnen we de gemiddelde afstand tot het eindpunt berekenen. Deze bedraagt 53.46cm. Hieruit kunnen we dan de standaard deviatie berekenen. Deze bedraagt 27.40cm. We

kunnen ook nagaan hoeveel van de tests binnen 1 standaard deviatie van het gemiddelde liggen. Het interval wordt dan: (26.06cm, 80.86cm). 60% van de tests liggen binnen dit interval. 100% van de tests liggen binnen 2 standaard deviaties van het gemiddelde. We kunnen dus besluiten dat de uitkomsten van de tests relatief groot verspreid zijn t.o.v. het eind put. Dit is te wijten aan het beperkte vermogen van de robot om zich accuraat voor te bewegen en te draaien. In sectie ... leggen we verder uit hoe dit komt.

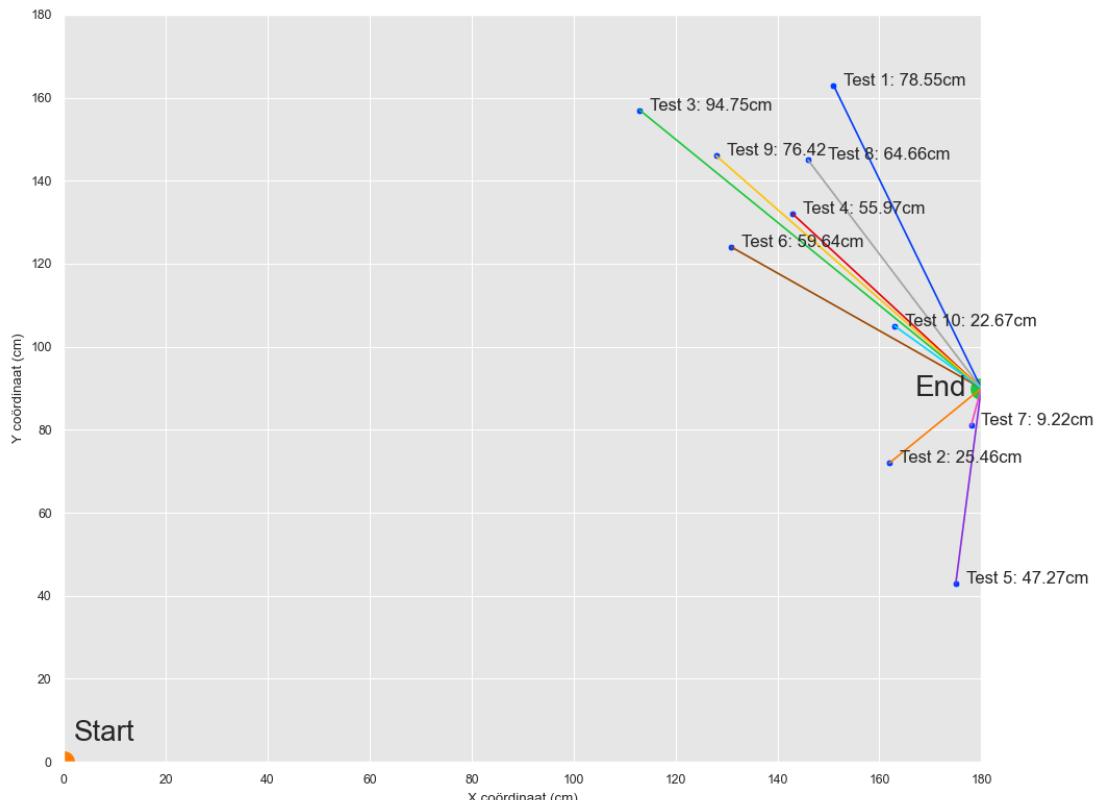


Figure 60: Resultaten Test Plot met Afstanden

8 Zelfreflectie

Na de afloop van de bachelorproef zijn er een aantal zaken dat we hebben bijgeleerd. Bovendien, in een eventuele remake van de robot zouden we een aantal zaken zeker anders aanpakken. Omdat we wisten dat het project veel tijd en werk zou vergen, was het op voorhand plannen of toch een ruwe schatting maken van waaraan we welke weken moesten werken vrij belangrijk. We merkte ook op dat dit niet altijd makkelijk was om dit bij te houden. Sommige problemen die voorkwamen waar we niet op gerekend hadden zorgden voor kleine achterstanden. Hoewel door aanpassingen te maken aan het plan en sommige weken gedurende het jaar harder te werken, lukte het ons vrij goed het ruw plan te volgen. Dit is grotendeels te wijten aan het feit dat we bij het opstellen van de planning, ingepland hebben dat we tegenslagen gingen hebben en hier dus genoeg ruimte en tijd voor hebben gelaten. Opnieuw wordt de kracht van het goed plannen van projecten bewezen. Daarbovenop leerden we vooral goed werken met 3D-printers. Dit lijkt vrij eenvoudig om aan te leren maar er zijn een talrijk parameters die je kan aanpassen om je prints te verbeteren/ te versnellen. Hoewel we door trial and error veel tijd hebben verloren met de prints, zijn we tevreden deze vaardigheid mee te nemen naar volgende projecten.

Wat we nog bijlerden was in welke mate het samenwerken en goede communicatie je project kan bevorderen. Het feit dat je veel kan bijleren door het kritisch denken samen met een collega student, vooral omdat we beiden in een verschillende keuze-richting zitten heeft dit begunstigd. Dit heeft echter ook een nadeel, de ene heeft een beknoptere kennis over wat de andere aan het doen is wat het helpen bemoeilijkt.

De robot die we nu hebben rolt maar is niet accuraat bestuurbaar. Dit komt, zoals eerder besproken, grotendeels door 2 redenen: het concept en de servomotoren. Indien een tweede versie gebouwd zou worden, raden we aan om het gebruik van servomotoren te herbekijken. We hebben namelijk de meeste problemen daaraan te danken, zie sectie 6.5. Men zou als oplossing meer accurate motoren gebruiken zoals bijvoorbeeld steppermotoren. Dit zou al veel ondervonden problemen wegwerken. Hiernaast zorgt het design dat we gekozen hebben ook voor een aantal problemen, zie sectie . Zo kan het bijvoorbeeld zijn dat na het rollen, de ring zich bijna verticaal bevindt waardoor een rotatie van de robot niet meer zal resulteren in de gewenste rotatie. Dit is duidelijk een ontwerp fout. We zouden ook graag meer tijd hebben willen spenderen aan een goede gewichtsverdeling binnen de bol door meer tests uit te voeren. Doordat de gewichtsverdeling niet optimaal is, heeft de robot een inaccurate beweging tijdens het besturen. Het stabiliseren van de robot na het stoppen met rollen wordt ook beïnvloed door de matige gewichtsverdeling. Als laatste, omdat we zo compact mogelijk wouden werken is het zeer lastig om de robot uit elkaar halen. Waardoor we steeds wachten op meerdere redenen om de robot te demonteren. We vinden de robot nogal stroef.

We hebben gedurende de bachelorproef in het begin lang zitten brainstormen over het concept van de robot. Dit was nodig omdat we niet genoeg tijd zouden hebben gehad om twee compleet verschillende versies te bouwen. Hierdoor moesten we één concept kiezen en alles op alles zetten om ervoor te zorgen dat dit concept zou werken.

De delen waar de wel tevreden mee zijn is het gebruik van een polystyrenen bol, dit gaf vrijheid om aanpassingen te doen en steeds dezelfde bol te gebruiken. Ook het gebruik van een al bestaande applicatie heeft ons veel moeite bespaard voor een draadloze connectie te realiseren. Deze applicatie maakt zeer goed gebruik van de Bluetooth functionaliteit van de ESP32. De aanpak van eerst een protoboard te maken en vervolgens een echte PCB zorgt ook voor minder problemen en tijdverspilling. Hiernaast zijn we over het algemeen wel tevreden over de 3D prints die we gemaakt hebben. In bijzonder het middengedeelte dat alle verschillende onderdelen mooi

incorporeert. We hebben geprobeerd om zo een afgewerkt mogelijke robot te maken ook al werkt deze functioneel niet perfect. Naar onze mening is dit redelijk goed gelukt. Al deze zaken nemen we zeker en vast mee naar andere projecten.

9 Conclusie

Al bij al hebben we zeer veel bijgeleerd bij dit project. Van werkmatig plannen tot goed samenwerken met een student collega, het uitvoeren van een uitgebreide brainstorm over het concept tot het realiseren van dit concept met alle moeilijkheden en tegenslagen die dit met zich meebrengt. De sferische robot die we gerealiseerd hebben is zeker niet perfect en er absoluut ruimte voor verbetering. Maar we zijn wel van mening dat we een interessante alternatieve manier hebben gedemonstreerd waar zeker op verder gebouwd kan worden.

Verder wensen we ook onze promotoren, Jan Lemeire en Colas Schretter te bedanken voor de goede feedback en interessante ideeën die ze ons meegaven.

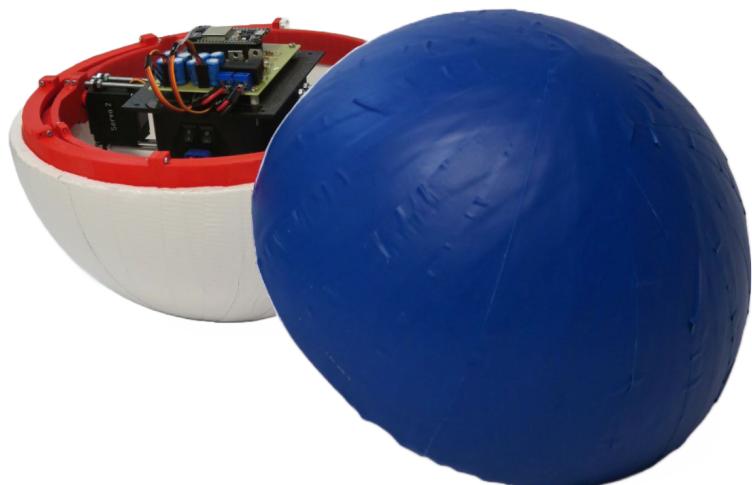


Figure 61: Onze Sferische Robot

A Appendix

A.1 Exploded Views

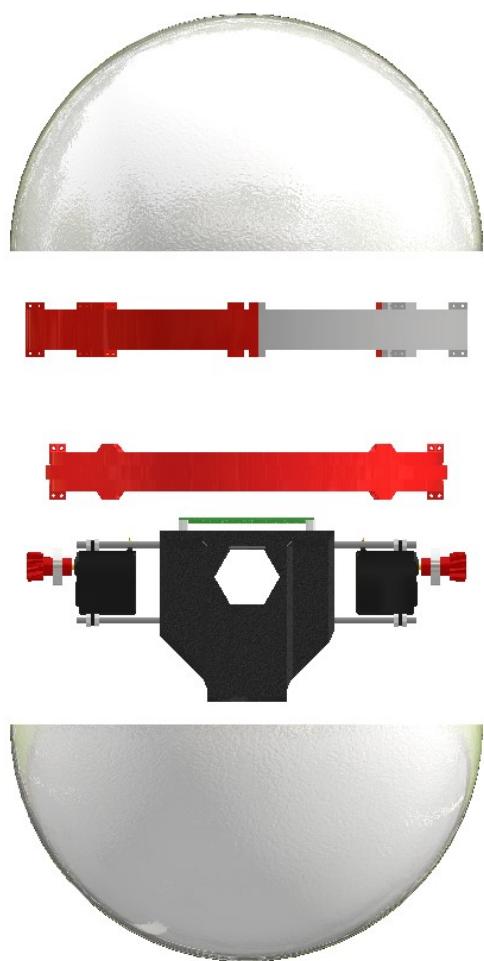


Figure 62: Exploded View 1



Figure 63: Exploded View 2

A.2 Componentenlijst

Link naar de componentenlijst: https://vub-my.sharepoint.com/:x/g/personal/seppo_goossens-vub_be/ERClRyIUbHRPkvrL34qMdzEBD3QMmxcaq5XX6PZtLk3dJA?e=7yMl38

A.3 Technische Tekeningen

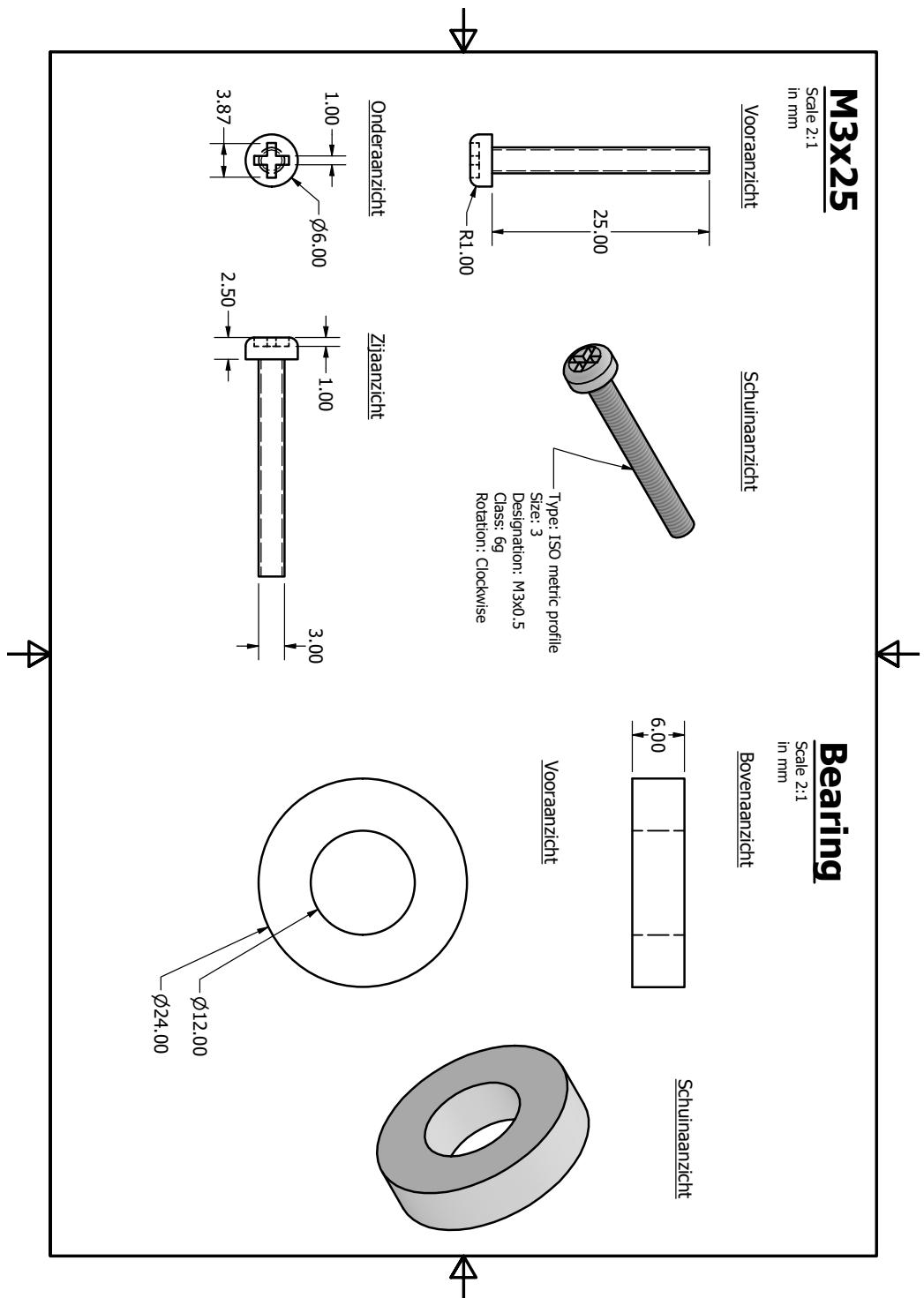


Figure 64: M3x25 en Bearing

Stang met tred

Scale 1/2
in mm

Vooraanzicht

215.00

Onderaanzicht
 $\phi 4.00$

Type: ISO metric profile
Size: 4
Designation: M4x0.7
Class: 6g
Rotation: Clockwise

Schuinaanzicht

Standoff

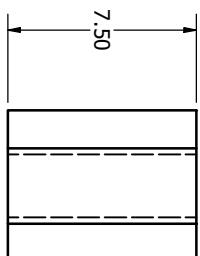
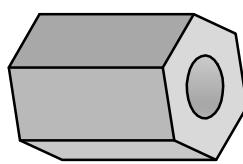
Scale 5:1
in mm

Bovenaanzicht

1.35

$\phi 2.50$

Schuinaanzicht



Vooraanzicht



Figure 65: Stang en Standoff

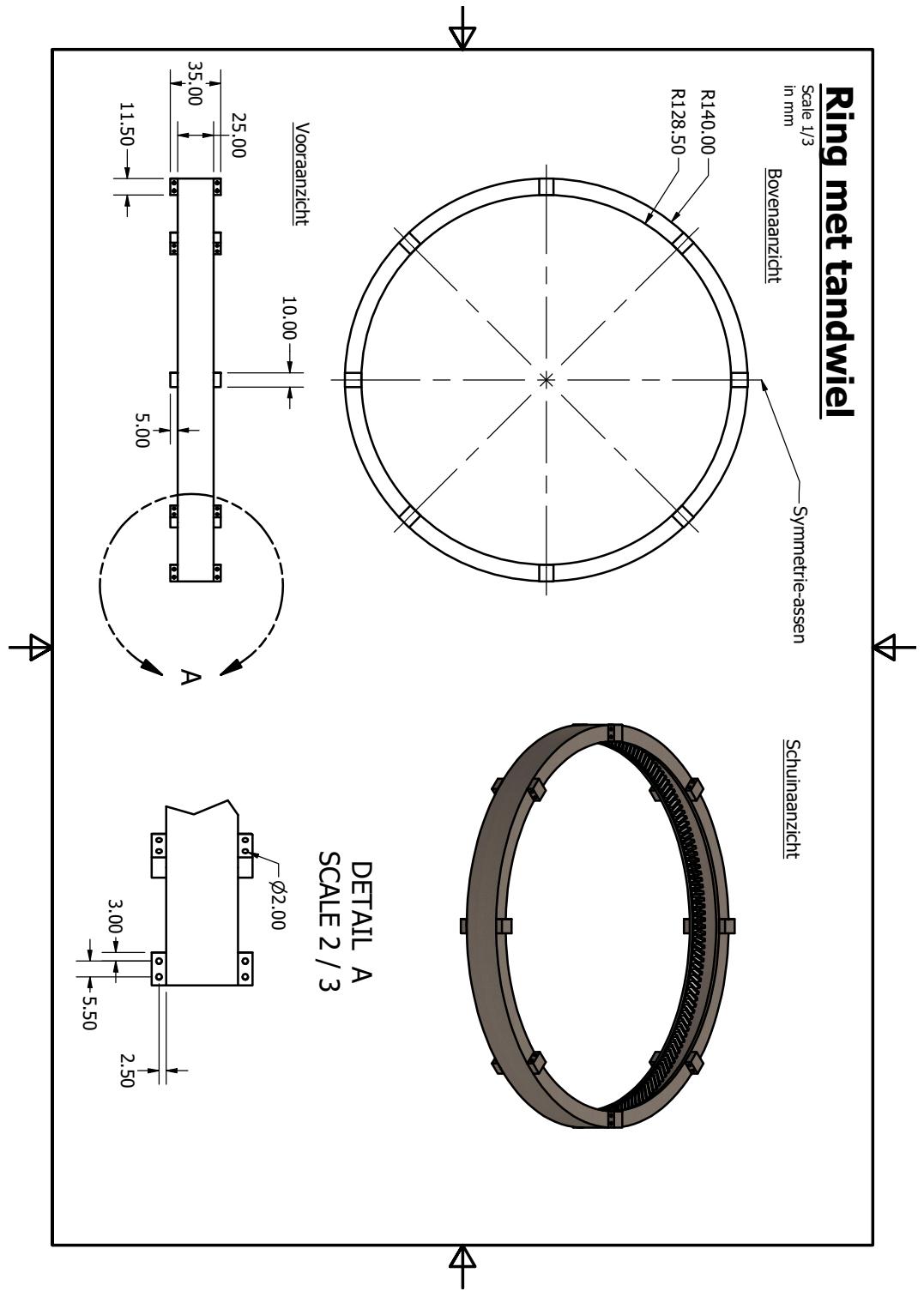
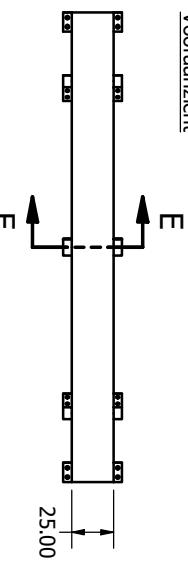


Figure 66: Ring met Tandwiel

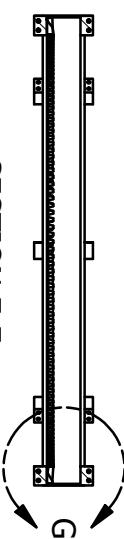
Ring met tandwiel

Scale 1/3
in mm

Vooranzicht



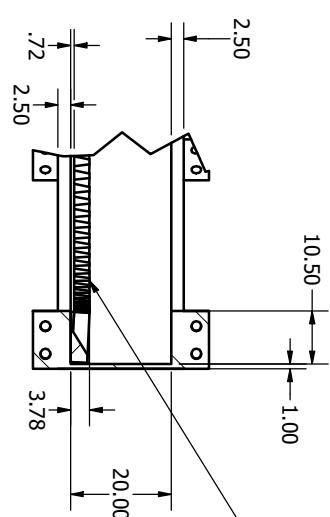
SECTION E-E
SCALE 1 / 3



Stukje ring met tandwiel

Scale 1/2
in mm

Schuinaanzicht



Tandwheelspecificaties

Gear ratio: 16.8182 ul
Facewidth: 10 mm
Module 1.5 mm
Shaft angle: 90 deg
Pressure angle: 15 deg
Helix angle: 10 deg
Number of teeth: 185 ul

DETAIL G
SCALE 1

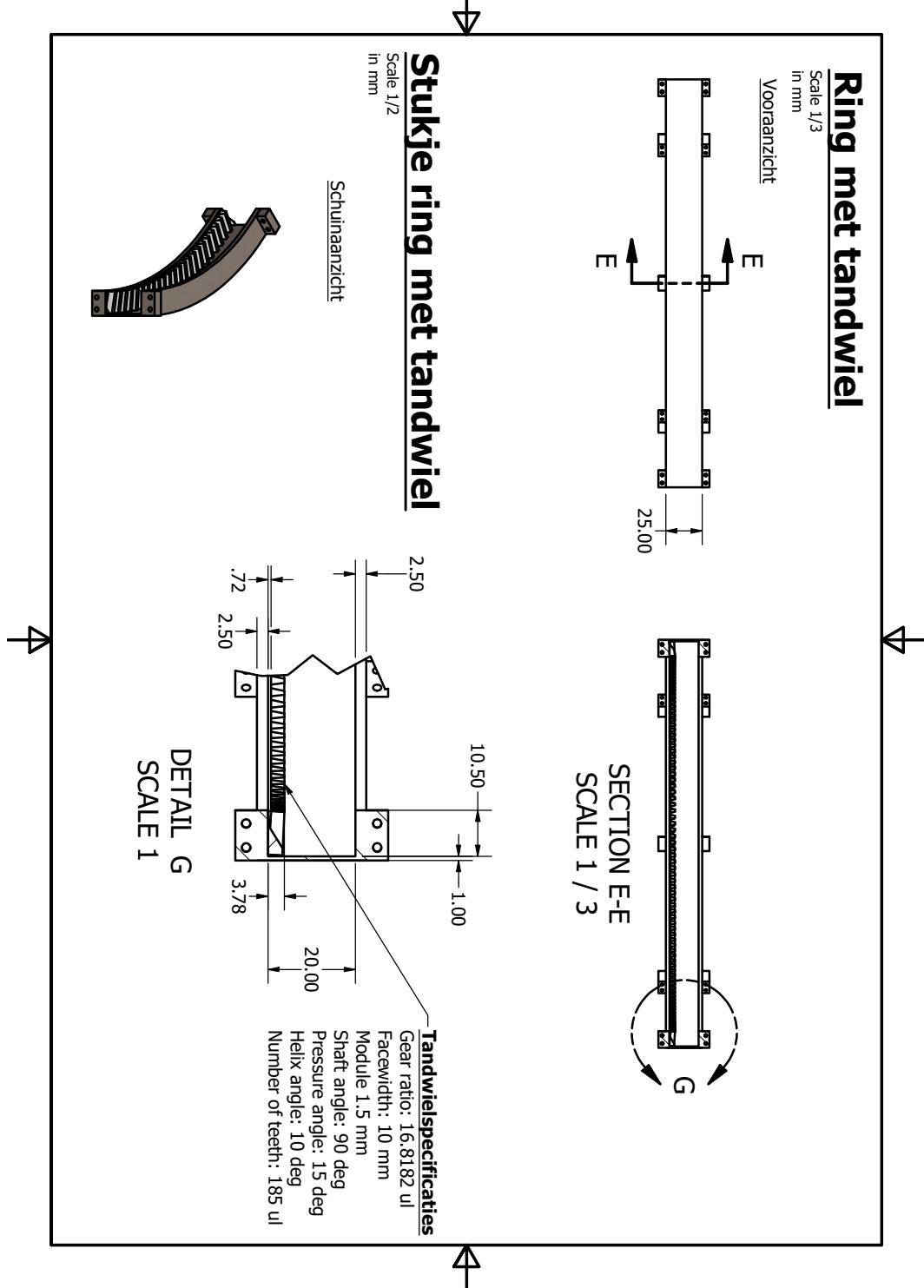


Figure 67: Ring met tandwiel en stuk ring

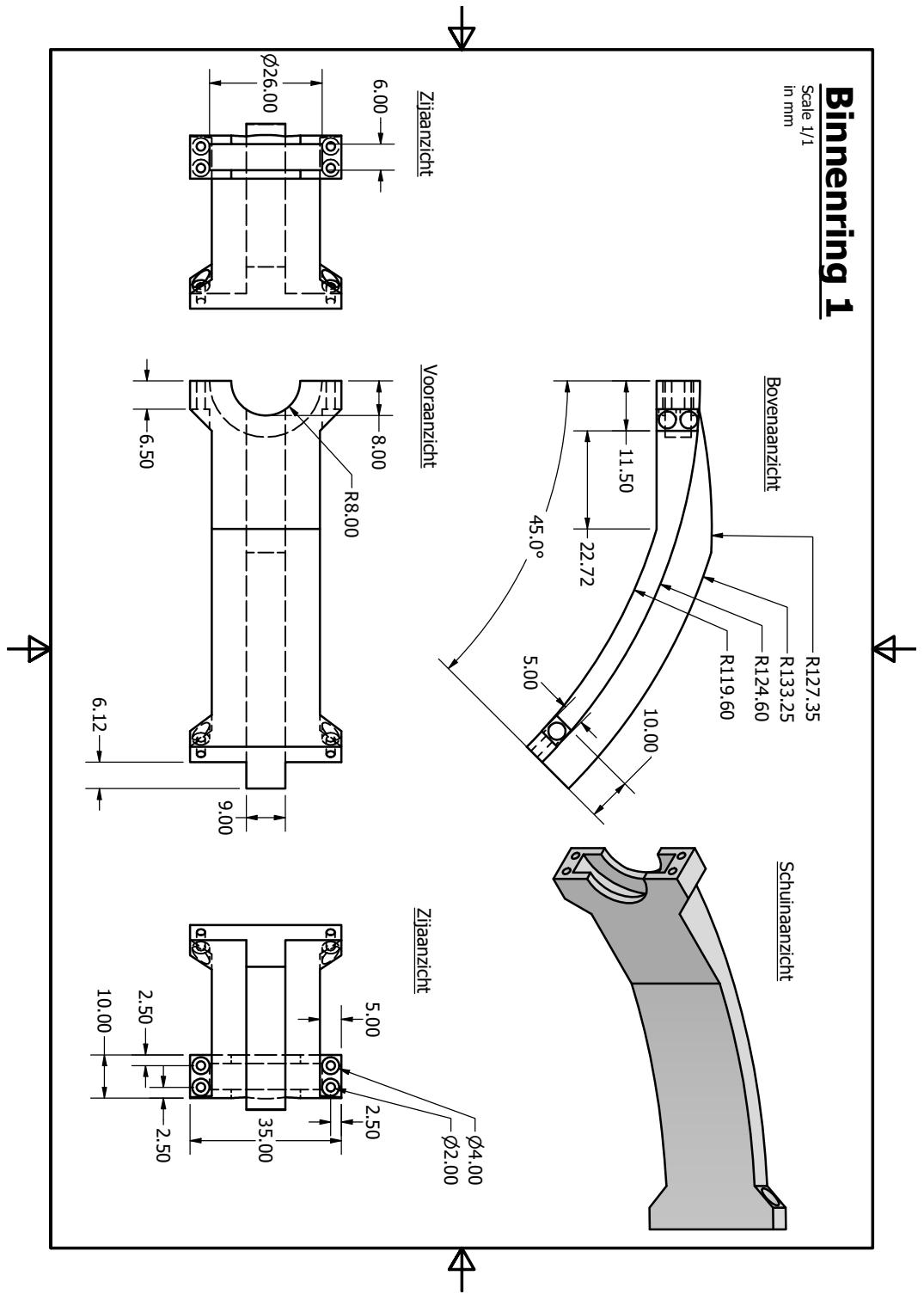
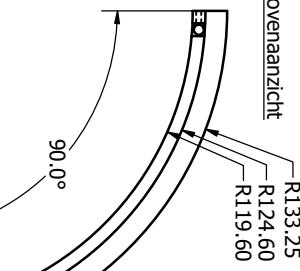


Figure 68: Binnenring aangrijpstuk

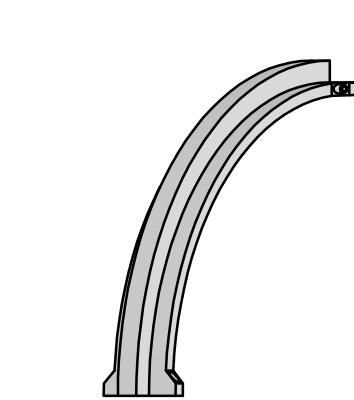
Binnenring middenstuk

Scale 1/2
in mm

Schuinaanzicht



Vooraanzicht



Zijaanzicht

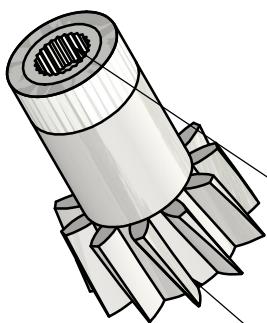
Ring met tandwiel

Scale 2:1
in mm

Bovenaanzicht



Schuinaanzicht



Specifieaties tandwiel van de servo

Tandwielspecificaties

Gear ratio: 16.8182 ul
Facewidth: 10 mm
Module 1.5 mm
Shaft angle: 90 deg
Pressure angle: 15 deg
Helix angle: 10 deg
Number of teeth: 11 ul

Vooraanzicht

Ø11.98
Ø8.50

Zijaanzicht

6.00



Figure 70: Tandwiel servo

Middenstuk

Scale 1/3
in mm

De gaten in vorm van hexagons en de chamfers zijn weggelegd,
omdat deze niet bijdragen tot het design maar alleen het 3D-printen vergemakkelijkt.

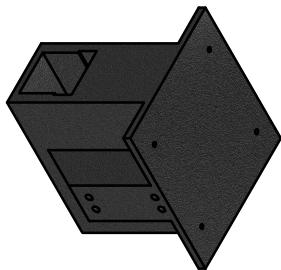
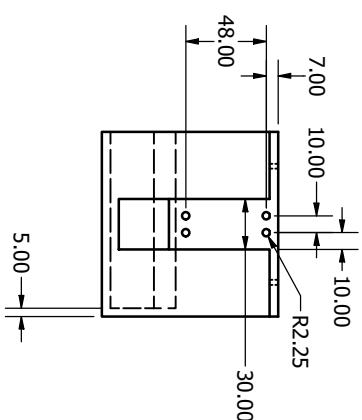
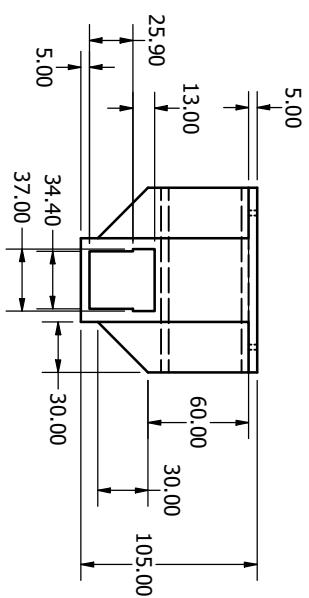
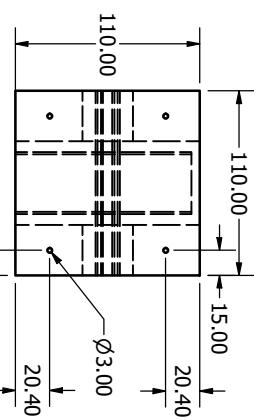


Figure 71: Middenstuk

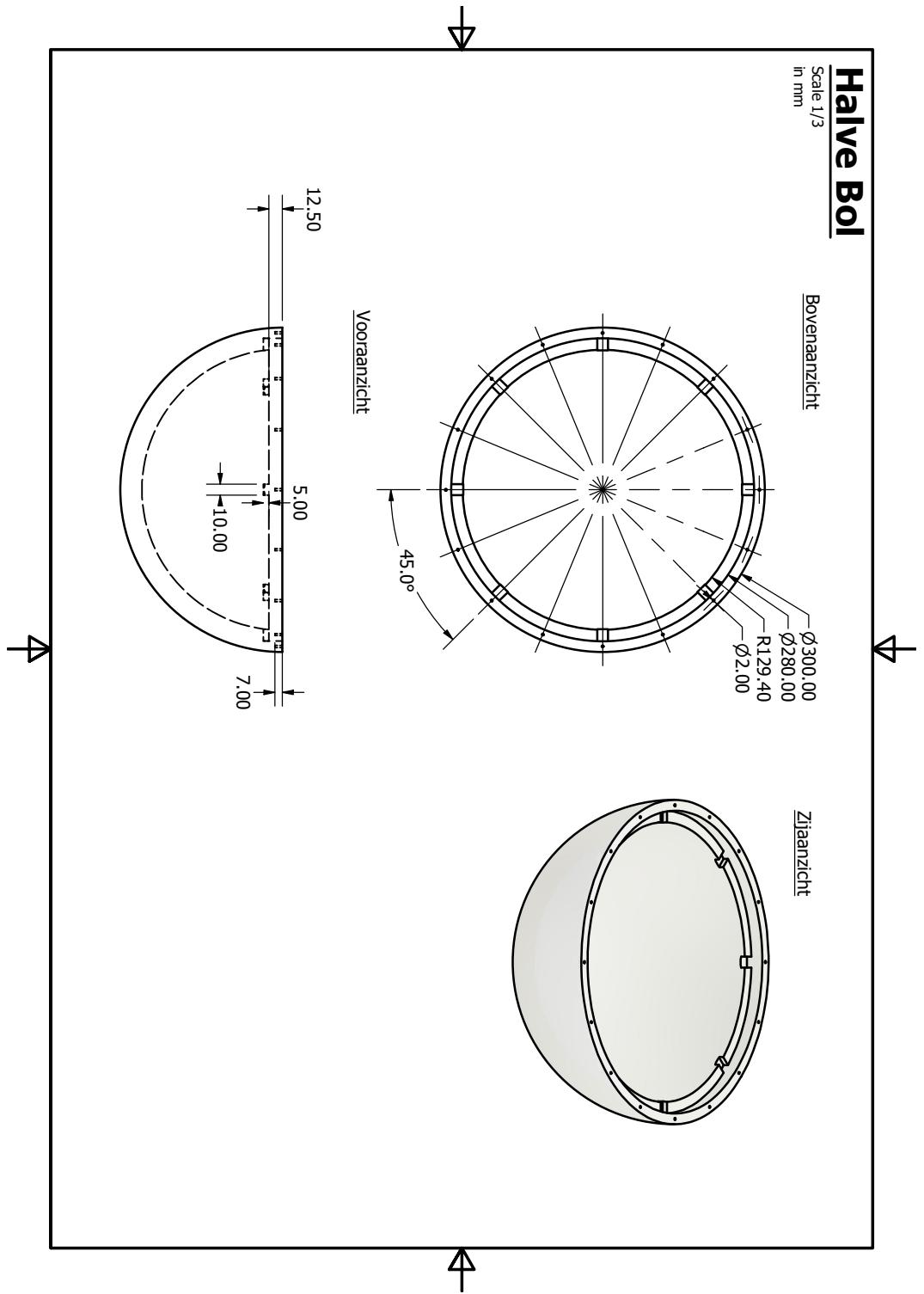


Figure 72: Halve bol

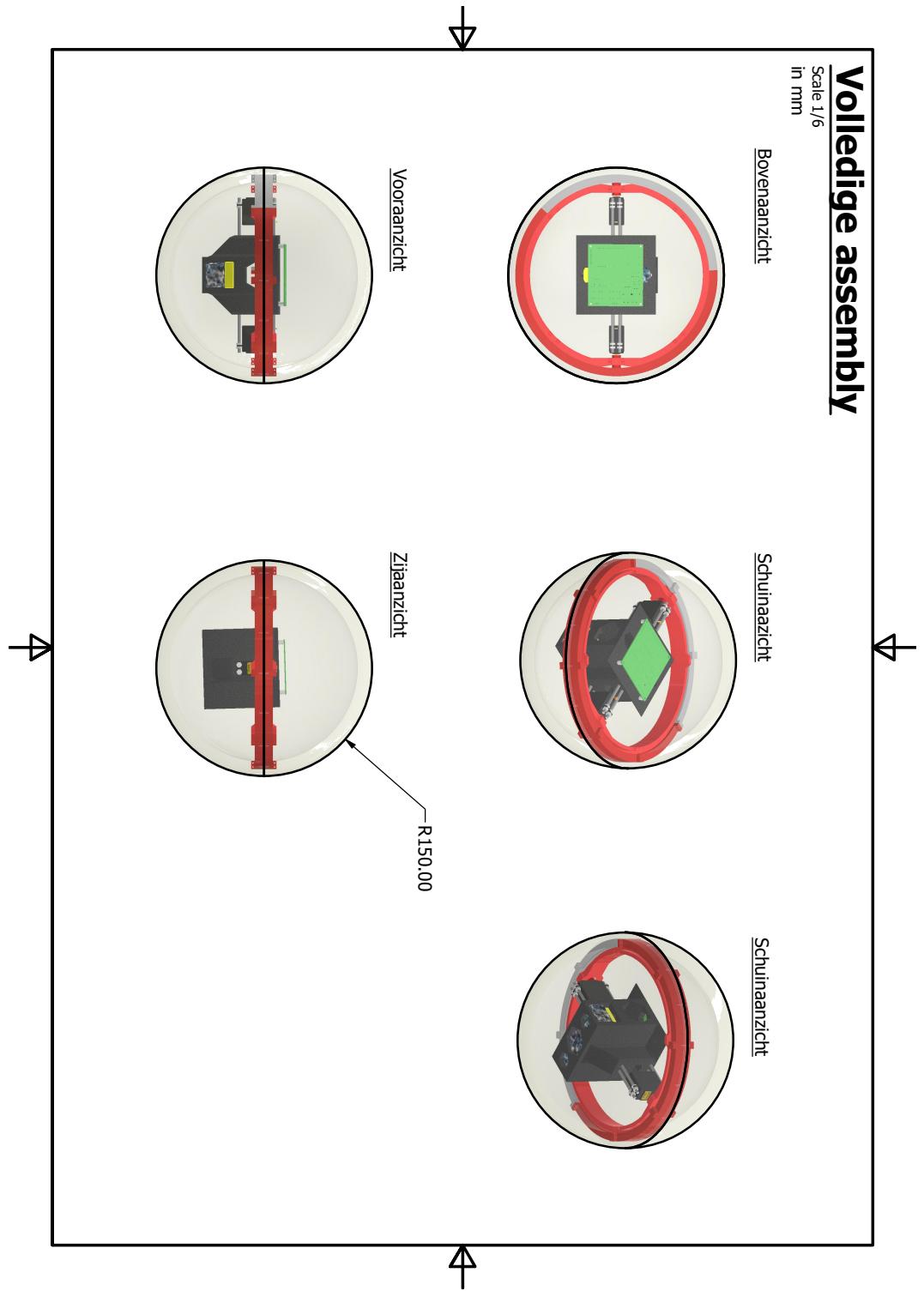


Figure 73: Volleidige Assembly

Exploded view

Scale 1/5

| TABLE | | |
|-------------------|-------------|--------|
| Naam | Hoeveelheid | Nummer |
| M3x25 | 2 | 1 |
| Lager | 2 | 2 |
| Stang met tread | 4 | 3 |
| Standoff | 4 | 4 |
| Ring met tandwiel | 2 | 5 |
| Binnerring | 2 | 6 |
| Servo tandwiel | 2 | 7 |
| Middenstuk | 1 | 8 |
| Halve bol | 2 | 9 |
| PCB | 1 | 10 |
| Batterijen | 2 | 11 |
| M3 moeren | 16 | 12 |
| Servomotor | 2 | 13 |

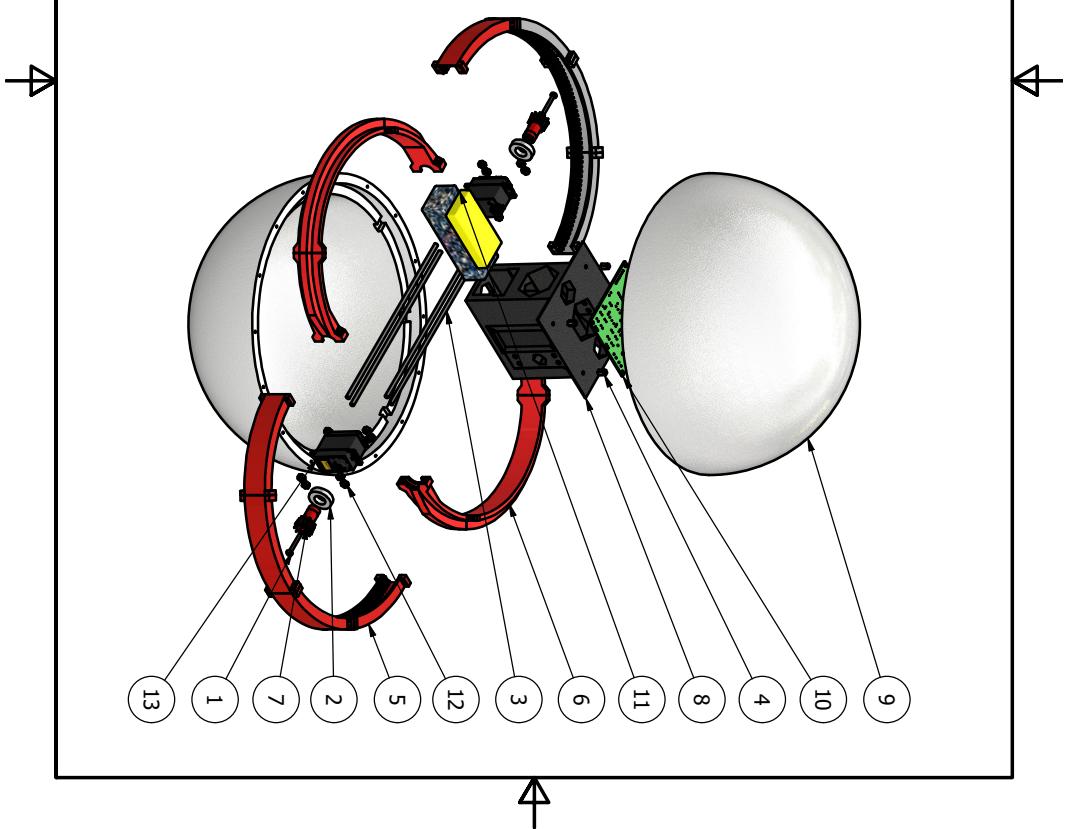


Figure 74: Volledige Assembly Exploded View

A.4 Schematic, Protoboard en PCB

A.4.1 Schematic

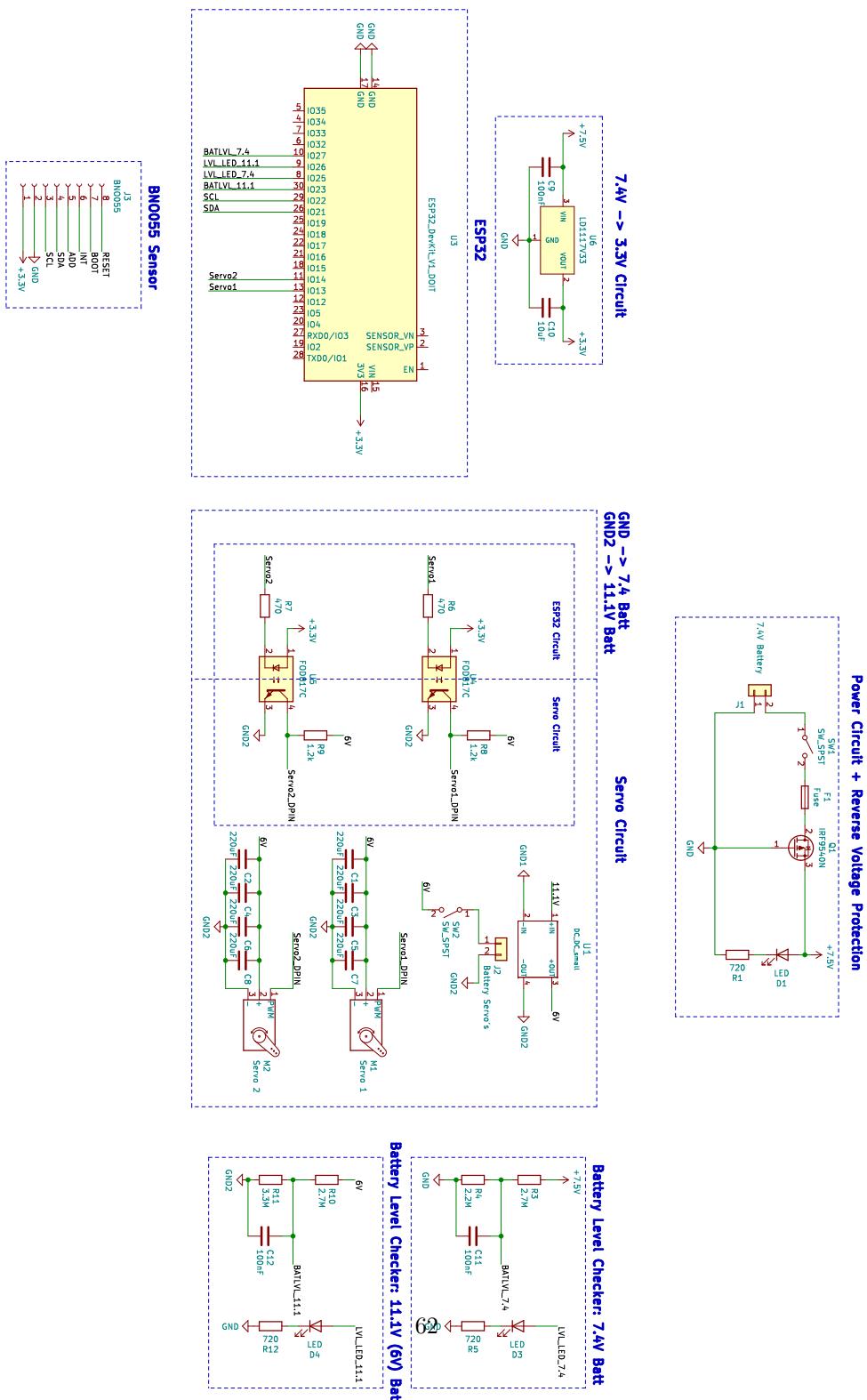


Figure 75: Volledige Schematic

A.4.2 Protoboard en PCB

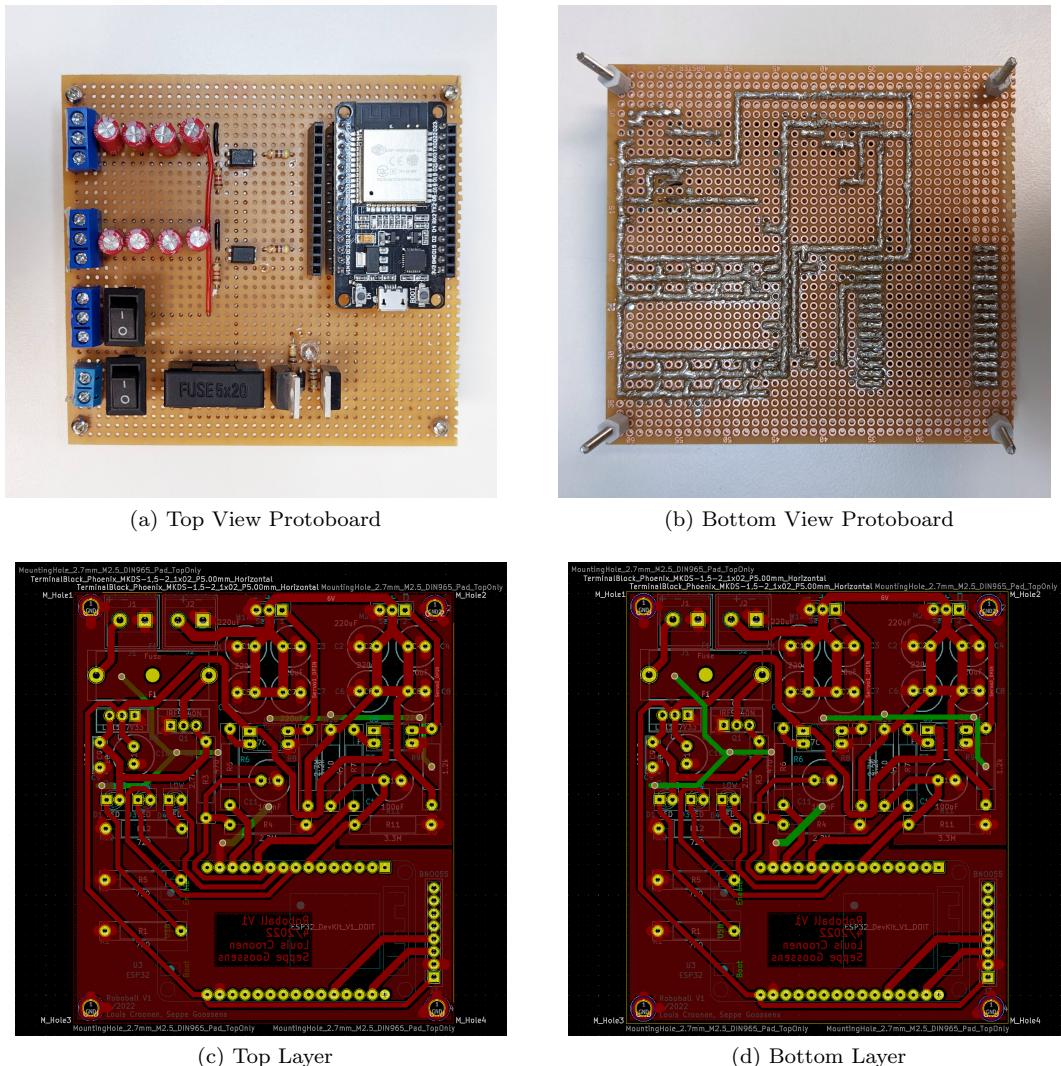
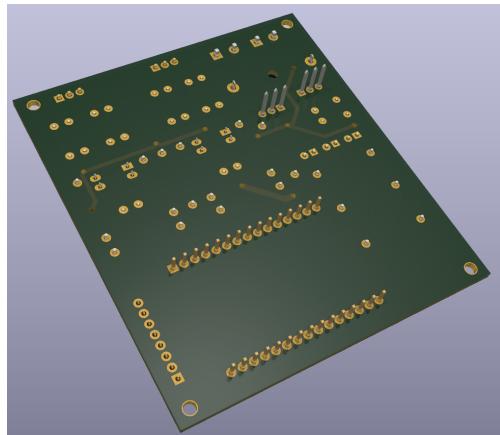


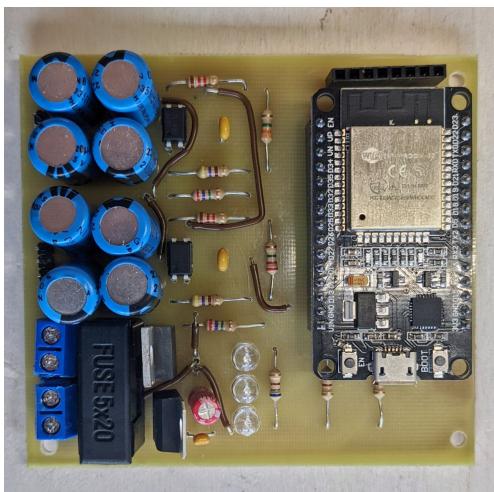
Figure 76: Protoboard en PCB in Kicad



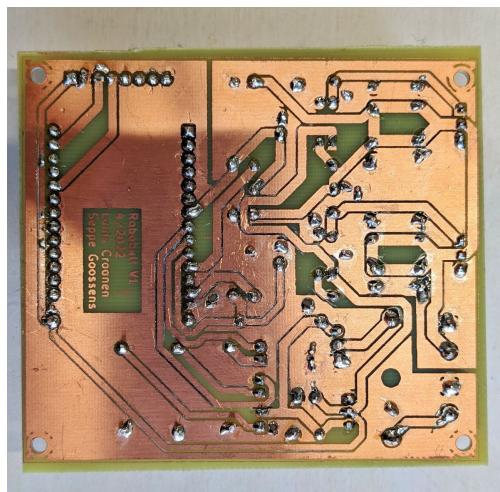
(a) Kicad PCB Renders Top Layer



(b) Kicad PCB Renders Bottom Layer



(c) Top View PCB



(d) Bottom View PCB

Figure 77: Kicad render en PCB

A.5 Blok Schema Elektronica

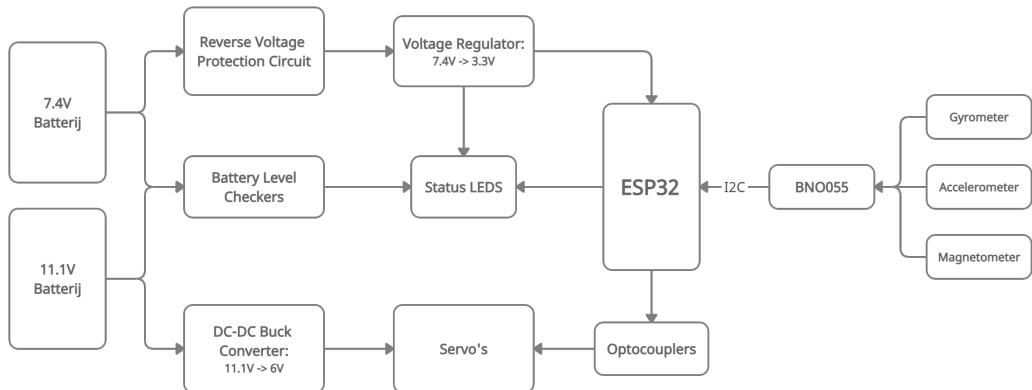


Figure 78: Blok Schema Elektronica

A.6 Algemeen Blok Schema

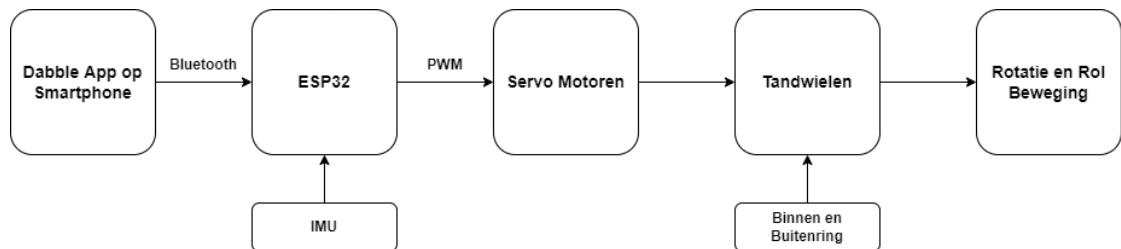


Figure 79: Algemeen Blok Schema Elektronica en Mechanica

A.7 Code

A.7.1 Main Code

```
1 void loop() {
2
3     //Battery Level Checker
4     check_batt_level();
5
6     //IMU
7     IMU_info();
8     //Bluetooth and Servo Control
9     Dabble.processInput();
10    if (GamePad.getRadius() <= 1) {
11        neutral();
12    }
13    if (GamePad.getAngle() >= 60 && GamePad.getAngle() <= 120) {
14    {
15        forward();
16    }
17    if (GamePad.getAngle() >= 240 && GamePad.getAngle() <= 300)
18    {
19        backward();
20    }
21    if (GamePad.getAngle() >= 150 && GamePad.getAngle() <= 210)
22    {
23        left();
24    }
25    if (GamePad.getXaxisData() >= 5.20
26        && GamePad.getYaxisData() >= -3
27        && GamePad.getYaxisData() <= 3)
28    // angle cant be >= 330 and <= 30 at the same time
29    {
30        right();
31    }
32 }
```

Listing 1: Loop

In de loop wordt er constant gecheckt of er nieuwe signalen ontvangen worden via Bluetooth met de `Dabble.processInput();` command. Afhankelijk van de input van de gamepad zal de corresponderende functie aangeroepen worden voor het besturen van de servo's. Ook wordt de batterij level checker functie aangeroepen als ook de functie voor het verkrijgen van de info van de BNO055.

A.7.2 Battery Level Indicators

```
1 void check_batt_level(){
2     //7.4V Battery
3     value_74 = analogRead(lvl_74) * resolutionVoltage;
4     voltage_74 = value_74 * ((R1_74 + R2_74) / R1_74);
5     Serial.println(voltage_74);
6     if(voltage_74 <= limit_74){
7         digitalWrite(led_74, HIGH);
8         Serial.println("7.4V LOW VOLTAGE");
9     }
10    else{
11        digitalWrite(led_74, LOW);
12    }
13 }
```

```

14 //11.1V Battery
15 value_11 = analogRead(lvl_11) * resolutionVoltage;
16 voltage_11 = value_11 * ((R1_11 + R2_11) / R1_11);
17 Serial.println(voltage_11);
18 if(voltage_11 <= limit_11){
19   digitalWrite(led_11, HIGH);
20   Serial.println("11.1V LOW VOLTAGE");
21 }
22 else{
23   digitalWrite(led_11, LOW);
24 }
25 }
```

Listing 2: Battery Level Checker Functie

In deze functie wordt er per batterij gekeken wat het spanningsniveau is en of deze al dan niet te laag is. Eerst wordt de analoge waarde omgezet in de corresponderende spanning via de `resolutionVoltage`. Deze waarde is zo dat 0V op de IO pin overeenkomt met een 0 als analoge waarde en 3.3V (max spanning op IO pin ESP32) overeenkomt met een analoge waarde van 4095 (max analoge waarde ESP32). Vervolgens wordt de verkregen waarde via de spanningsdeler omgerekend naar de effectieve spanning van de batterij. Dit kunnen we bereken via de spanningsdeler met de gekende weerstandswaarden. Als deze spanning onder een gekozen limiet ligt dan zal de corresponderende LED beginnen branden.

A.7.3 BNO055 Info

```

1 void IMU_info() {
2   /* Get a new sensor event */
3   sensors_event_t event;
4   bno.getEvent(&event);
5
6   /* Display the floating point data */
7   Serial.print("Yaw: ");
8   Serial.print(event.orientation.x, 4);
9   Serial.print("\tRoll: ");
10  Serial.print(event.orientation.y, 4);
11  Serial.print("\tPitch: ");
12  Serial.print(event.orientation.z, 4);
13
14  /* New line for the next sample */
15  Serial.println("");
16
17  /* Wait the specified delay before requesting next data */
18  delay(BNO055_SAMPLERATE_DELAY_MS);
19 }
```

Listing 3: IMU info

Hierin wordt de data verkregen van de BNO055 geprint in de Serial Monitor. Deze code kan terug gevonden worden in de voorbeeldcode van de BNO055 library.

A.7.4 Movements

```

1 void neutral() {
2   //Serial.println("STOP");
3   s1_pos = s1_npos;
4   s2_pos = s2_npos;
5   servol1.writeMicroseconds(s1_pos);
```

```

6     servo2.writeMicroseconds(s2_pos);
7 }
8
9 void forward() {
10    //Serial.println("VOORUIT");
11    s1_pos = s1_bpos;
12    s2_pos = s2_fpos;
13    servo1.writeMicroseconds(s1_pos);
14    servo2.writeMicroseconds(s2_pos);
15 }
16
17 void backward() {
18    //Serial.println("ACHTERUIT");
19    s1_pos = s1_fpos;
20    s2_pos = s2_bpos;
21    servo1.writeMicroseconds(s1_pos);
22    servo2.writeMicroseconds(s2_pos);
23 }
24
25 void left() {
26    //Serial.println("LINKS");
27    s1_pos = s1_fpos;
28    s2_pos = s2_fpos;
29    servo1.writeMicroseconds(s1_pos);
30    servo2.writeMicroseconds(s2_pos);
31 }
32
33 void right() {
34    //Serial.println("RECHTS");
35    s1_pos = s1_bpos;
36    s2_pos = s2_bpos;
37    servo1.writeMicroseconds(s1_pos);
38    servo2.writeMicroseconds(s2_pos);
39 }

```

Listing 4: Movement Functions

Per richting zullen de servo's aangestuurd worden met de corresponderende richtingen en snelheden. We maken hier gebruik van `.writeMicroseconds()` omdat dit meer nauwkeurigheid biedt dan `.write()`.

A.8 Flow Chart

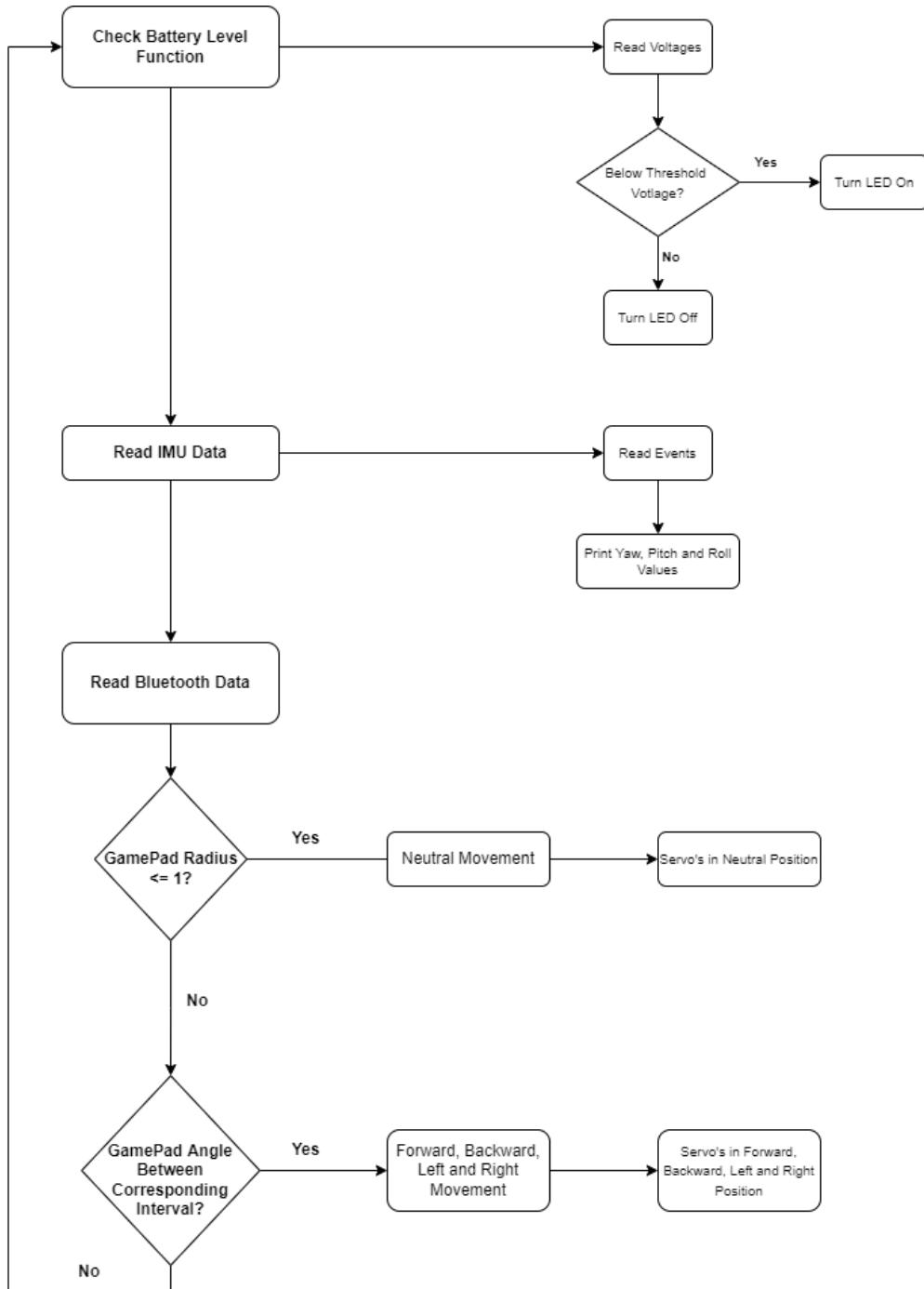


Figure 80: Flow Chart Code

A.9 Logboek

Voor het bijhouden, plannen en documenteren van het werk hebben we gebruik gemaakt van Trello. Link naar de Trello: <https://trello.com/invite/b/EQnFq7HD/d20867c98e79759346c6c189eb306458/bachelorproef>

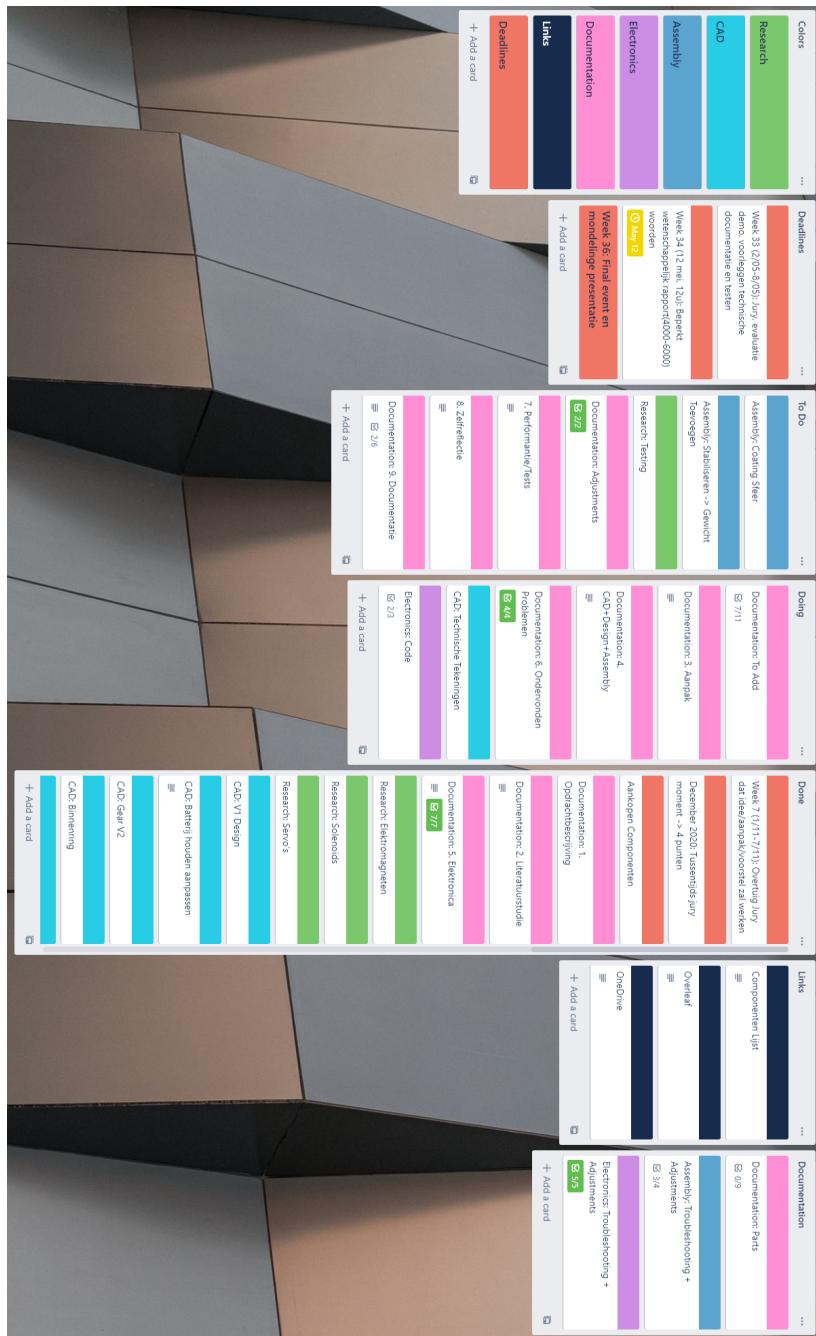


Figure 81: Trello Board

A.10 Foto's en Video's

A.10.1 Foto's



Figure 82: Onze Sferische Robot 1



Figure 83: Onze Sferische Robot 2

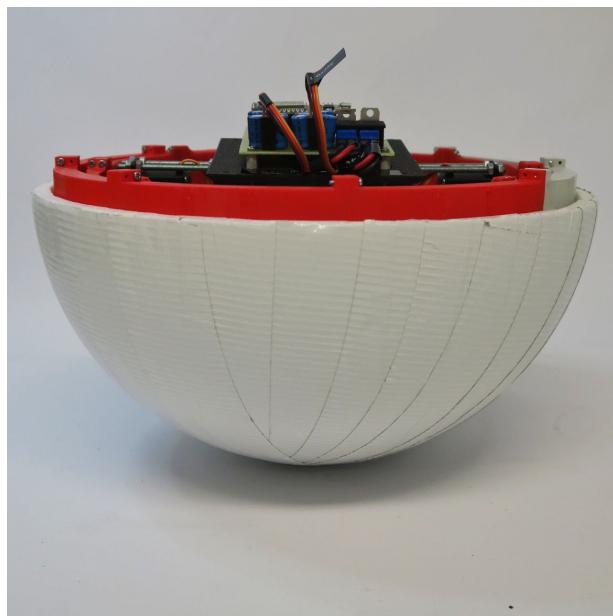


Figure 84: Onze Sferische Robot 3

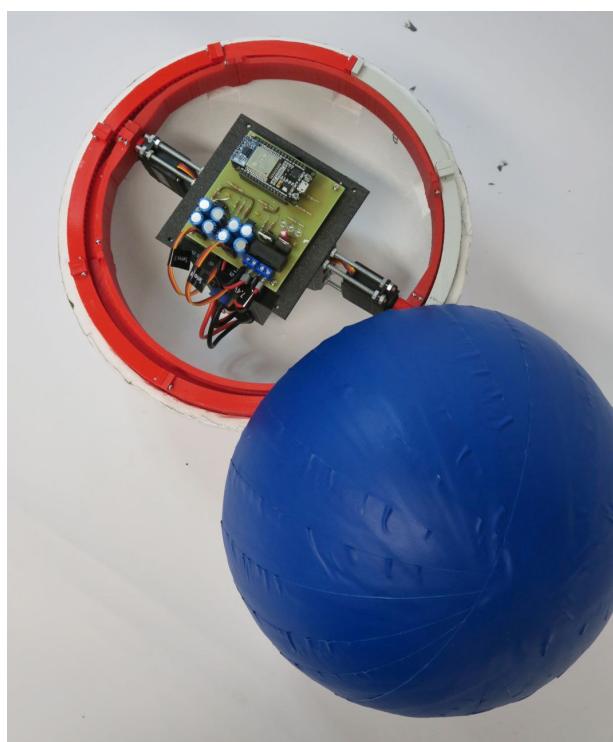


Figure 85: Onze Sferische Robot 4

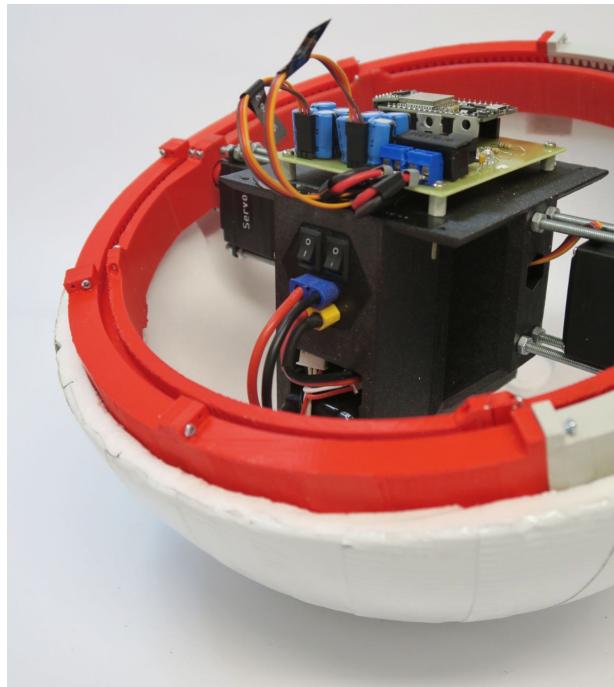


Figure 86: Onze Sferische Robot 5

A.10.2 Video's

Test: https://vub-my.sharepoint.com/:v/g/personal/seppe_goossens_vub_be/EaxPyH3Nj8JFscVyfRp4EaEBzyX54Ce=rHdRyU

Rotatie van de robot: https://vub-my.sharepoint.com/:v/g/personal/seppe_goossens_vub_be/ERiJWm6qOv1MpeWf64Ykay8BdOOHueWiVz5foP2pNh_Sgw?e=zTrVFA

Voorwaartse en achterwaardse beweging: https://vub-my.sharepoint.com/:v/g/personal/seppe_goossens_vub_be/EdQIF5d93bZJg8oHexfdLjcBckIPgarK09WytNdFqcgtqA?e=6A1Ob6

Rollen van de robot 1: https://vub-my.sharepoint.com/:v/g/personal/seppe_goossens_vub_be/Ef0gtQZV6ulNnwvHF-QF1uYBwoUGjXq_tmnnqQ2szplUD2w?e=EiwaM9

Rollen van de robot 2: https://vub-my.sharepoint.com/:v/g/personal/seppe_goossens_vub_be/ERvZdsqXoMFMt8qfqAxeWwcBcMMEgykt9nqokduy7aFsVA?e=mY80my

Sources

- [1] *What's in the bot? – Sphero.* <https://sphero.com/blogs/news/whats-in-the-bot>. (Accessed on 02/22/2022).
- [2] CNET. *Cracking Open - Sphero BB-8 Star Wars toy.* Youtube. 2015. URL: <https://www.youtube.com/watch?v=bdsZvzz7oIM>.
- [3] Adam Savage's Tested. *How the BB-8 Sphero Toy Works.* Youtube. 2015. URL: <https://www.youtube.com/watch?v=5FHtcR78GA0&t=379s>.
- [4] *Details on the Sphero API · Sphero Public SDK.* https://sdk.sphero.com/docs/api_spec/general_api. (Accessed on 02/22/2022).
- [5] José Rodríguez Corral et al. "A Study on the Suitability of Visual Languages for Non-Expert Robot Programmers". In: *IEEE Access* 7 (Jan. 2019), pp. 17535–17550. DOI: [10.1109/ACCESS.2019.2895913](https://doi.org/10.1109/ACCESS.2019.2895913).
- [6] Wesley Fabry. *Review van Sphero 2.0, bestuurbare robotbal!* <https://www.appletips.nl/sint-kerst-cadeautip-sphero-veelzijdige-robotbal-bestuurbaar-met-je-smartphone/>. (Accessed on 02/22/2022). Mar. 2016.
- [7] Wikipedia contributors. *Spherical robot — Wikipedia, The Free Encyclopedia.* [Online; accessed 22-December-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Spherical_robot&oldid=1061482552.
- [8] *POWERFUL ELECTROMAGNET (5KG LIFTING) : 6 Steps (with Pictures) - Instructables.* <https://www.instructables.com/POWERFUL-ELECTROMAGNET-5KG-LIFTING/>. (Accessed on 12/22/2021). 2017.
- [9] Wikipedia contributors. *Coilgun — Wikipedia, The Free Encyclopedia.* <https://en.wikipedia.org/w/index.php?title=Coilgun&oldid=1060067604>. [Online; accessed 22-December-2021]. 2021.
- [10] Mammoth Memory. *The difference between an electromagnet and a solenoid.* <https://mammothmemory.net/physics/magnets-and-electromagnetism/electromagnetism/the-difference-between-an-electromagnet-and-a-solenoid.html>. (Accessed on 12/22/2021).
- [11] *Microcontroller Development Board.* NodeMCU ESP32. Joy-it. 2018. URL: [%5Curl%7Bhttps://cdn-reichelt.de/documents/datenblatt/A300/SBC-NODEMCU-ESP32-DATASHEET-V1.2.pdf%7D](https://cdn-reichelt.de/documents/datenblatt/A300/SBC-NODEMCU-ESP32-DATASHEET-V1.2.pdf).
- [12] *ESP32 Series Datasheet.* ESP32. Espressif Systems. URL: [%5Curl%7Bhttps://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf%7D](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
- [13] *LOW DROP FIXED AND ADJUSTABLE POSITIVE VOLTAGE REGULATORS.* LD1117 SERIES. ST. 2005. URL: [%5Curl%7Bhttps://www.sparkfun.com/datasheets/Components/LD1117V33.pdf%7D](https://www.sparkfun.com/datasheets/Components/LD1117V33.pdf).
- [14] HobbyKing.com. *Pack ZIPPY Compact 1000mAh 2S 25C Lipo.* https://hobbyking.com/nl_nl/zippy-compact-1000mah-2s-25c-lipo-pack.html?__store=nl_nl. (Accessed on 05/24/2022).
- [15] Last Minute Engineers. *Insight Into ESP32 Sleep Modes & Their Power Consumption.* <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>. (Accessed on 05/04/2022).
- [16] *Servo MG996R Datenblatt.* MG996R. Az-Delivery. URL: [%5Curl%7Bhttps://cdn.shopify.com/s/files/1/1509/1638/files/Servo_MG996R_Datenblatt.pdf?5470426105647302493%7D](https://cdn.shopify.com/s/files/1/1509/1638/files/Servo_MG996R_Datenblatt.pdf?5470426105647302493%7D).

- [17] *International IOR Rectifier*. IRF9540N. HEXFET Power MOSFET. 1998. URL: %5Curl%7Bhttps://datasheetspdf.com/pdf-file/284171/InternationalRectifier/IRF9540N/1%7D.
- [18] HobbyKing. *TowerPro MG996R 10kg Servo 10kg / 0.20sec / 55g*. https://hobbyking.com/nl_nl/towerpro-mg996r-10kg-servo-10kg-0-20sec-55g.html?__store=nl_nl. (Accessed on 05/04/2022).
- [19] *LM2596 DC-DC Adjustable PSU Module*. LM2596. URL: %5Curl%7Bhttp://tpelectronic.ir/datasheets/20150123144301750.pdf%7D.
- [20] Kuongshun. *LM2596 LM2596S DC-DC Step-Down Power Supply Module – Kuongshun Electronic Shop*. <https://kuongshun.com/products/lm2596-lm2596s-dc-dc-step-down-power-supply-module>. (Accessed on 05/04/2022).
- [21] *4-Pin DIP Phototransistor Optocouplers*. FOD814, FOD817. Onsemi. URL: %5Curl%7Bhttps://www.onsemi.com/pdf/datasheet/fod814-d.pdf%7D.
- [22] Mark Hughes. *Capturing IMU Data with a BNO055 Absolute Orientation Sensor - Projects*. <https://www.allaboutcircuits.com/projects/bosch-absolute-orientation-sensor-bno055/>. (Accessed on 05/09/2022). Mar. 2017.
- [23] Renzo Mischianti. *ESP32 DevKitC v4 high resolution pinout and specs – Renzo Mischianti*. <https://www.mischianti.org/2021/07/17/esp32-devkitc-v4-high-resolution-pinout-and-specs/>. (Accessed on 05/04/2022). July 2021.
- [24] Stephen Colley. *What Is Brown Out Reset in Microcontrollers? How to Prevent False Power-Downs - Technical Articles*. <https://www.allaboutcircuits.com/technical-articles/what-is-brown-out-reset-microcontroller-prevent-false-power-down/>. (Accessed on 05/04/2022). Oct. 2019.