



*Figuur 1: Motor- en Servo knoppen*

Thijs Hulshof en Thijs Willemsen  
Embedded Systems Engineering  
Klas: ESE-2B-n  
HAN Solarboat  
Projectleider: Leonie Dijkhof  
Versie: 1.0

## Revisies

Versie	Wanneer	Wie	Wat
0.1	24-09-2024	Thijs en Thijs	Aanmaken van het bestand. Documentatie tot en met functioneel ontwerp.
0.2	7-10-2024	Thijs en Thijs	Feedback verwerken uitbreiden tot technische ontwerp.
0.3	22-10-2024	Thijs en Thijs	Volledig document tot en met technisch ontwerp
1.0	23-12-2024	Thijs en Thijs	Volledig document

## Voorwoord

In samenwerking met de HAN Solarboat hebben wij, Thijs Hulshof en Thijs Willemsen een project gedaan genaamd “Motorcontroller feedback”. Dit project is uitgevoerd in het tweede leerjaar van onze huidige opleiding Embedded Systems Engineering.

Dit rapport is opgesteld om een uitgebreid overzicht te bieden van de ontwikkeling en evaluatie van dit project. Het doel is om inzicht te geven in de belangrijkste ontwerpkeuzes, functionaliteiten, en resultaten van onze testfase.

Graag willen wij de HAN Solarboat en de HAN bedanken voor het mogelijk maken en verzorgen van dit project. Bijzonder willen wij Leonie Dijkhof, Maarten van Els en Yosri Tielbeke bedanken. Leonie voor het organiseren van het project, Maarten voor de begeleiding en kennis binnen dit semester en Yosri voor alle informatie en kennis over de boot.

## Samenvatting

Dit rapport beschrijft het ontwerp, de ontwikkeling en de realisatie van een innovatief feedbacksysteem voor de motorcontroller van de HAN Solarboat. De HAN Solarboat is een duurzaam, high-tech zonnevoertuig dat door professionals uit het bedrijfsleven en studenten van verschillende opleidingen wordt ontwikkeld en ingezet voor deelname aan nationale en internationale races.

Het primaire doel van het feedbacksysteem is het leveren van betrouwbare en heldere statusinformatie over de motorcontroller aan de piloot en het team. De huidige feedback is een groene led op de motorcontroller, dat is tijdrovend en potentieel onveilig. Dit wordt vervangen door een oplossing die gebruik maakt van de NUCLEO L432KC-microcontroller, het CAN-bus communicatieprotocol en diverse actuatoren. Het systeem is ontworpen om spatwaterdicht, lichtgewicht en gebruiksvriendelijk te zijn, met minimale stroomconsumptie, wat essentieel is voor het optimaliseren van de prestaties van de boot.

Naast een uitgebreide beschrijving van het functionele en technische ontwerp bevat het rapport een vernieuwd PCB-schema, hardware-specificaties en softwarefunctionaliteiten. Er wordt dieper ingegaan op het testproces, waarbij de nauwkeurigheid, betrouwbaarheid en duurzaamheid van het systeem zijn gevalideerd. Er zijn componenten hergebruikt en in combinatie met nieuwe componenten zijn de nieuwe functionaliteiten gerealiseerd.

## Inhoudsopgave

Revisies.....	1
Voorwoord .....	2
Samenvatting .....	3
1 Inleiding.....	6
1.1 Aanleiding .....	6
1.2 Probleemstelling.....	6
1.3 Doelstellingen .....	7
1.4 Betrokkenen .....	8
1.5 Structuur van het rapport.....	8
2 Vooronderzoek.....	9
2.1 Huidige situatie .....	9
2.2 CAN-bus .....	11
2.2.1 Algemeen .....	11
2.2.2 Solarboat.....	11
2.3 NUCLEO-L432KC microcontroller.....	12
2.3.1 Specificaties .....	12
2.3.2 Pinout.....	12
2.3.3 CAN-peripheral NUCLEO-L432KC .....	13
2.3.4 NUCLEO-L432KC software IDE .....	14
2.4 Feedback Systeem .....	15
2.4.1 Led (Licht).....	15
2.4.2 Buzzer (Geluid).....	15
2.4.3 Trilmotor (Haptisch).....	15
2.4.4 Conclusie.....	16
3 Functioneel ontwerp .....	17
3.1 Functionele specificatie .....	17
3.2 Datastroom .....	17
3.3 Software werking.....	18
3.4 Hardware .....	18
4 Technisch ontwerp .....	19
4.1 Technische specificatie .....	19
4.2 Architectuur .....	20
4.3 Interfaces .....	20
4.3.1 Microcontroller – CAN-bus .....	20
4.3.2 Microcontroller – Bedieningspaneel.....	21
4.3.3 Microcontroller – Actuator .....	21
4.4 Hardware .....	22
4.4.1 Benodigde hardware.....	22
4.4.2 Voedingspanning .....	22
4.4.3 PCB.....	22

4.4.4	Aansluitschema .....	23
4.4.5	Benodigdheden .....	24
4.5	Software.....	25
4.5.1	Flowchart microcontroller .....	25
4.5.2	Led aansturing.....	26
4.5.3	Timer-peripheral .....	26
4.5.4	CAN-peripheral .....	26
5	Realisatie .....	27
5.1	Aangebrachte verbeterpunten aansluitschema .....	27
5.1.1	Gebruik van een MOSFET .....	27
5.1.2	Componenten gebruik .....	27
5.2	PCB-ontwerp .....	28
5.3	Microcontroller L432KC software .....	29
5.3.1	Software werking .....	29
5.3.2	Prioriteiten in software .....	29
5.4	Behuizing .....	30
5.4.1	Verbeterpunten behuizing SOLIDWORKS .....	30
5.4.2	Knoppen.....	31
5.5	Eindresultaat.....	32
5.5.1	Eindproduct .....	32
5.5.2	Stroomverbruik.....	32
6	Verifiëren en testen.....	33
6.1	Test plan .....	33
6.1.1	(Voorbeeld) Test Tabel.....	33
6.1.2	(Voorbeeld) Resultaten Tabel .....	33
6.2	Testen en Resultaten .....	34
6.2.1	Test Tabel.....	34
6.2.2	Test Resultaten .....	35
7	Conclusies en aanbevelingen .....	36
7.1	Conclusie.....	36
7.2	Worst-case scenario .....	36
7.3	Aanbevelingen .....	37
7.3.1	Power detectie.....	37
7.3.2	Code voor de servo .....	37
8	Verwijzingen .....	38
9	Bijlage .....	39
9.1	Aansluitschema versie 2.1 .....	39
9.2	PCB-design .....	40
9.3	Aansluitschema versie 2.0 .....	43

# 1 Inleiding

In dit hoofdstuk staat beschreven hoe dit project is ontstaan. Daarnaast worden de problemen beschreven die de aanleiding zijn voor dit project.

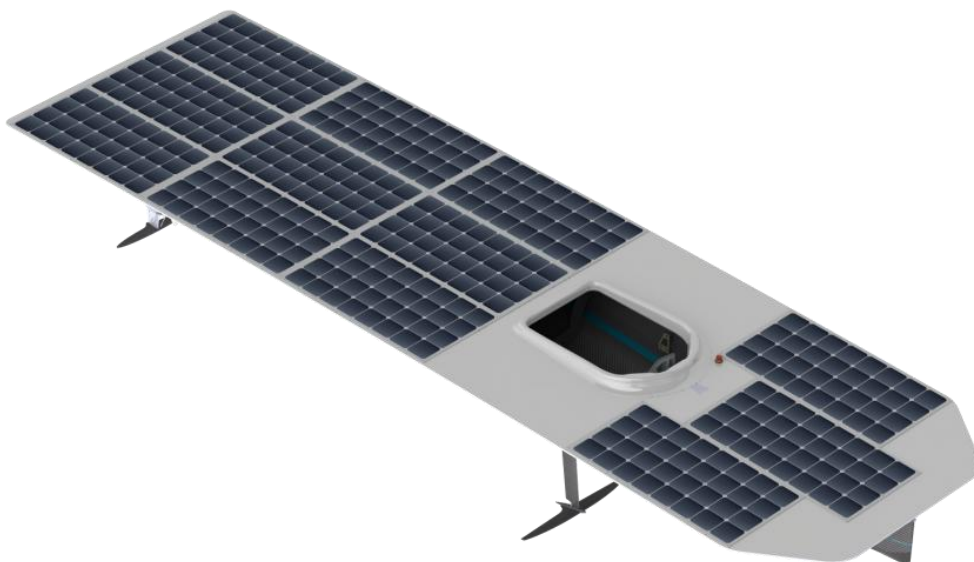
## 1.1 Aanleiding

Dit project wordt uitgevoerd in opdracht van de HAN solarboat, een boot die op zonne-energie vaart en wordt ontwikkeld door een team van enthousiaste studenten van verschillende opleidingen, in samenwerking met professionals van Qconcepts Design & Engineering. De HAN Solarboat neemt deel aan zowel nationale als internationale races.

Het IIME (Innovatieve Mobiliteit en Energie) stelt faciliteiten beschikbaar en biedt professionele ondersteuning uit het vakgebied. Elk semester werkt een nieuw team van studenten aan de verdere ontwikkeling van de HAN Solarboat. Bij IIME krijgen studenten de kans om zelfstandig bij te dragen aan de innovatie en technische vooruitgang van dit project.

## 1.2 Probleemstelling

Een solarboat is een boot die wordt aangedreven door zonne-energie. Zonnepanelen op de boot vangen de energie van de zon op en zetten deze om in elektriciteit. Deze elektriciteit wordt opgeslagen in een batterij. De opgeslagen energie kan de elektrische motor voeden die de boot aandrijft. Het concept van een solarboat combineert duurzaam varen met milieuvriendelijke technologie, aangezien er geen fossiele brandstoffen worden gebruikt en er geen uitstoot van schadelijke gassen is. Hieronder is de solarboat weergegeven:



*Figuur 2: Solarboat model*

De boot wordt dus aangedreven door een elektrische motor die wordt aangestuurd door een motorcontroller (SLRi-60-240)<sup>1</sup>. Deze bevindt zich achter in de boot en wordt tijdens het opstarten van de boot geïnitieerd. Tijdens het opstarten is het belangrijk voor het team om te weten of de controller goed functioneert. Momenteel wordt de motorcontroller getest door gas te geven. Gaat de motor draaien, dan doet de controller het. Zo niet, dan gaat het achterdek los. Dit is onhandig, onveilig en tijdrovend tijdens races. Dit zou makkelijker en veiliger kunnen door bijvoorbeeld in de cockpit feedback te geven over de status van de motorcontroller. Dit zou op verschillende manieren kunnen.

### 1.3 Doelstellingen

Het doel van dit project is het ontwikkelen van een betrouwbaar en gebruiksvriendelijk feedbacksysteem voor de motorcontroller van de HAN Solarboat. Het systeem moet ervoor zorgen dat zowel voor het team als de piloot duidelijk is wanneer de motorcontroller wordt opgestart. Het systeem moet fysieke controle van de motorcontroller overbodig maken. Hierdoor worden de huidige omslachtige en onveilige methoden vervangen door een eenvoudigere en veiligere oplossing.

Daarnaast zijn er enkele andere eisen gesteld waaraan de het eindproduct moet voldoen:

- **Waterdichtheid:** Het systeem moet bestand zijn tegen water, omdat de omgeving waarin de boot opereert vaak nat is.
- **Technische reglementen:** Het systeem moet voldoen aan de technische regels van de race-organisatie.
- **CAN-bus communicatieprotocol:** Het systeem maakt gebruik van het bestaande communicatieprotocol aan boord van de boot.
- **Minimaal stroomverbruik en gewicht:** Het ontwerp moet zo licht mogelijk zijn en zo min mogelijk stroom verbruiken, in overeenstemming met de eisen van de solarboat.

---

<sup>1</sup> (SLS-Sinusleistungssteller, n.d.)



## 1.4 Betrokkenen

Hieronder staat een tabel van alle personen die betrokken zijn bij dit project.

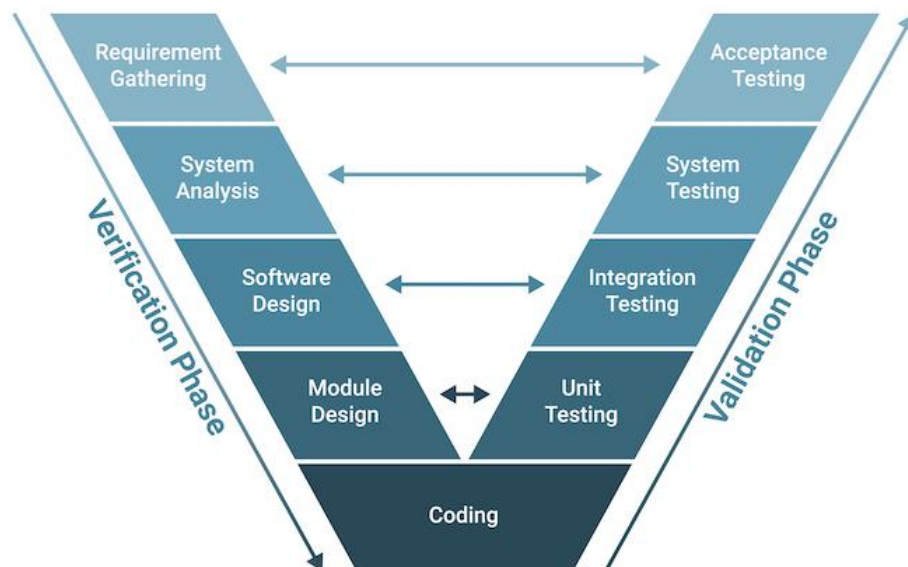
Naam:	Functie:	Studentnummer	E-mail:
Leonie Dijkhof	Projectleider	-	<a href="mailto:leonie@iime.nl">leonie@iime.nl</a>
Thijs Hulshof	Teamlid	2125571	<a href="mailto:t.hulshof2@student.han.nl">t.hulshof2@student.han.nl</a>
Thijs Willemsen	Teamlid	2122555	<a href="mailto:t.willemsen8@student.han.nl">t.willemsen8@student.han.nl</a>
Maarten van Els	Docent	-	<a href="mailto:maarten.vanels@han.nl">maarten.vanels@han.nl</a>

## 1.5 Structuur van het rapport

Dit rapport is opgebouwd volgens het V-model. Dit model is een methode om het project efficiënt uit te voeren. Aan het begin van het rapport wordt er uitleg gegeven over het vooronderzoek en de functionele eisen van de klant. Daarna worden alle technische eisen beschreven en in detail vertelt hoe het eindresultaat eruit komt te zien.

Het volgende deel in het rapport bestaat uit de realisatiefase. Hierin wordt beschreven hoe het eindproduct is gemaakt en hoe dit eindproduct is gerealiseerd. Om vervolgens het product te meten met de gesteld eisen wordt het product getest. De resultaten van de test worden na de realisatiefase in het rapport beschreven en toegelicht.

Tot slot is er een conclusie gemaakt en een aanbeveling gedaan voor verbeterpunten. Hieronder is voor de volledigheid het V-model weergegeven.



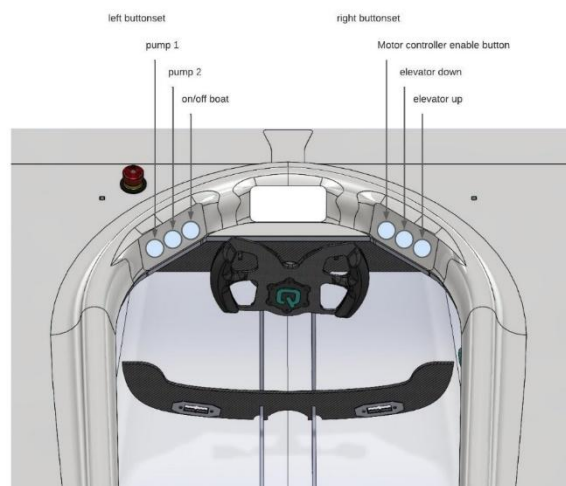
Figuur 3: V-Model

## 2 Vooronderzoek

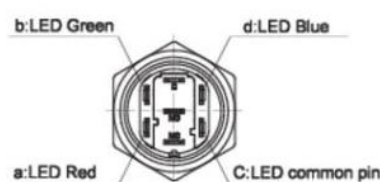
### 2.1 Huidige situatie

In de cockpit bevinden zich aan elke kant van het stuur 3 knoppen. Eentje hiervan is om de voeding van de Motorcontroller aan te zetten (zie Figuur 4). Als de voeding aan is gaat er in rode led rond het knopje branden. Er bevinden zich nog twee andere leds in de knop die groen en blauw zijn.

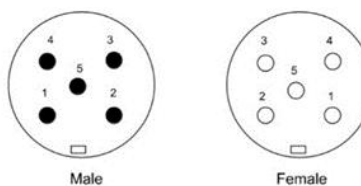
Twee andere knoppen zijn via een NUCLEO-L432KC<sup>2</sup> microcontroller op CAN-bus aangesloten (zie "Figuur 5"). De microcontroller stuurt op dit moment via CAN-bus commando's naar een servo moter achter in de boot. Dit is mogelijk doormiddel van een ttl naar CAN-bus converter. De converter bestaat uit een standaard mcp2562fd-e/sn in combinatie met een PESD1CAN. Deze componenten zitten op een speciaal ontworpen PCB (zie "Figuur 6: Huige Schema/PCB").



Figuur 4 Cockpit

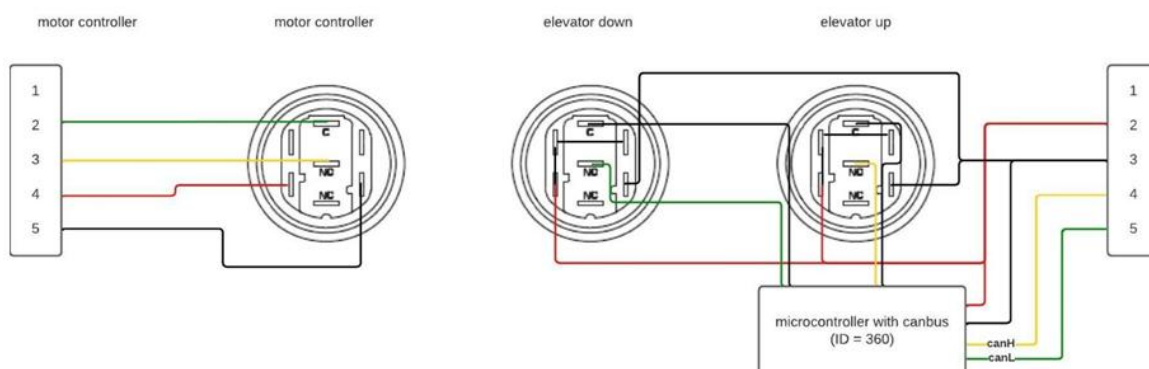


Button base diagram



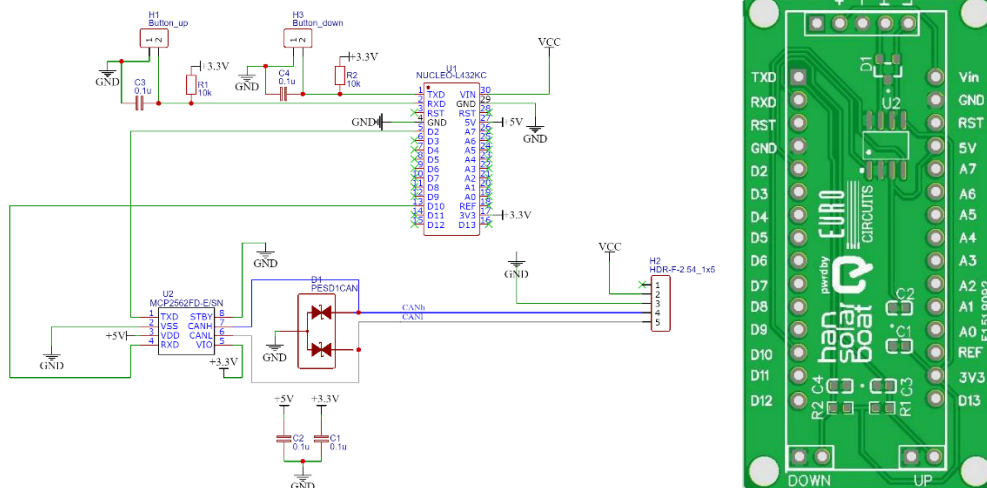
So-called 5-pin micro style connector (M12) shall use the following pinning applies:

Pin	Signal	Description
1	(CAN_SHLD)	Optional CAN Shield
2	(CAN_V+)	Optional CAN external positive supply (dedicated for supply of transceiver and optocouplers, if galvanic isolation of the bus node applies)
3	CAN_GND	Ground / 0V / V-
4	CAN_H	CAN_H bus line (dominant high)
5	CAN_L	CAN_L bus line (dominant low)



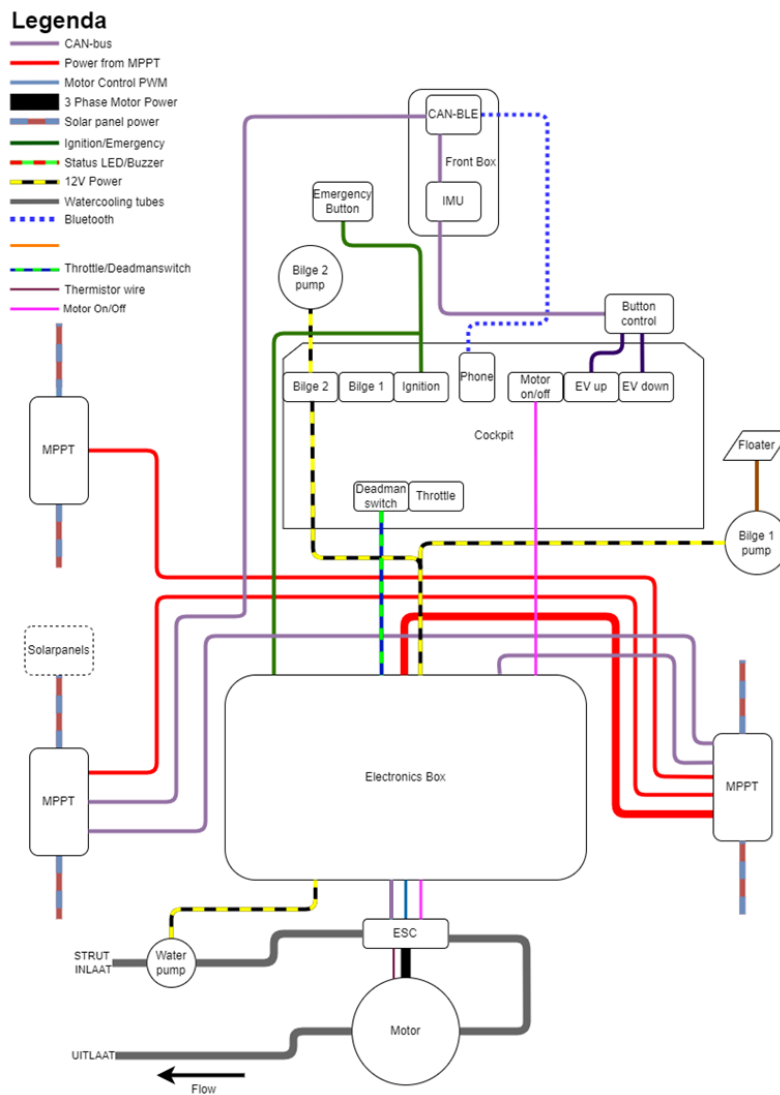
Figuur 5 Aansluiting knoppen cockpit

<sup>2</sup> (SLS-Sinusleistungssteller, n.d.)



Figuur 6: Huige Schema/PCB

In “Figuur 7” is het volledige elektrische schema weergegeven van de Solarboat. Hier zijn alle CAN-bus verbindingen zichtbaar en wordt de structuur van de boot duidelijk weergegeven.



Figuur 7: Volledig schema solarboat

## 2.2 CAN-bus

We doen een vooronderzoek naar CAN-bus omdat de Motorcontroller verbonden is met de CAN-bus. Door de CAN-bus in de gaten te houden kunnen we zien of de Motorcontroller aanstaan staat.

### 2.2.1 Algemeen

De CAN-bus (Controller Area Network) is een communicatieprotocol dat zorgt voor gegevensuitwisseling tussen verschillende elektronische componenten in een systeem, zoals sensoren en controllers. Het multi-master architectuurmodel van de CAN-bus zorgt ervoor dat meerdere apparaten op de bus kunnen communiceren zonder een centrale controle-eenheid. Elk apparaat op de bus kan op elk moment gegevens verzenden of ontvangen. Berichten op de CAN-bus hebben verschillende prioriteiten, wat betekent dat sommige berichten voorrang hebben op andere, afhankelijk van hun belangrijkheid.

### 2.2.2 Solarboat

In de solarboat wordt de CAN-bus gebruikt voor communicatie tussen verschillende subsystemen, waaronder de Motorcontroller. De solarboat maakt gebruik van meerdere CAN-bus protocollen, waarbij de motorcontroller specifiek het SLR-protocol<sup>3</sup> van SinusLeistungsSteller gebruikt voor de uitwisseling van gegevens. Dit protocol maakt het mogelijk om essentiële informatie zoals snelheden, stromen, en foutmeldingen in real-time te versturen.



Figuur 8: CAN-bus Protocol

Een bericht binnen het CAN-bus protocol bestaat uit:

ID (11 bits): Dit is verdeeld in twee delen:

4 bits om aan te geven wat voor soort bericht het is (bijvoorbeeld een statusupdate of een foutmelding).

7 bits voor het adres van de zender (het apparaat dat het bericht verstuurt).

Data (8 bytes): Dit is de daadwerkelijke informatie die het bericht draagt, zoals de status van de motorcontroller of foutinformatie.

De Motorcontroller van de solarboat heeft als Node ID 1 en stuurt elke 500 milliseconden zes verschillende berichten om de status van de motor en de controller te communiceren. Eén van deze berichten geeft aan of er een fout is opgetreden. Dit maakt het mogelijk om problemen in het systeem vroegtijdig op te sporen en actie te ondernemen.

<sup>3</sup> (NUCLEO-L432KC - STMicroelectronics, n.d.)

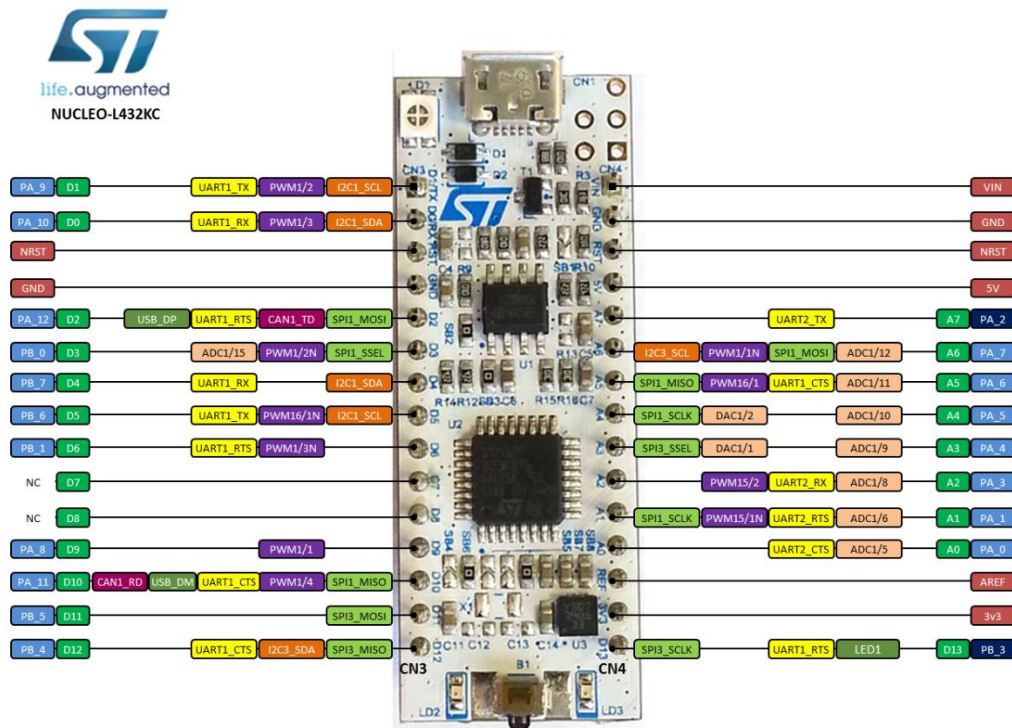
## 2.3 NUCLEO-L432KC microcontroller

Zoals in de huidige situatie is besproken gebruikt de solarboat op dit moment een NUCLEO-L432KC. Het mooiste zou natuurlijk zijn als deze hergebruikt kan worden voor dit project. Daarnaast is dit modulair, omdat deze microcontroller op meerdere plekken in de boot wordt gebruikt. Vanwege deze reden is er onderzoek naar gedaan. Een extra voordeel van deze microcontroller is dat deze momenteel al wordt gebruikt met CAN-communicatie voor de knoppen op het bedieningspaneel.

### 2.3.1 Specificaties

- STM32L432KC in UFQFPN32 package
- ARM®32-bit Cortex®-M4 CPU
- 80 MHz max CPU frequency
- VDD from 1.65 V to 3.6 V
- 256 KB Flash
- 64 KB SRAM
- Timers General Purpose (4)
- SPI/I2S (2)
- I2C (2)
- USART (2)
- 12-bit ADC with ten channels (1)
- GPIO (20) with external interrupt capability
- RTC
- Random Generator (TRNG for HW entropy)
- CAN-peripheral

### 2.3.2 Pinout



Figuur 9: Pinout microcontroller

### 2.3.3 CAN-peripheral NUCLEO-L432KC

In de NUCLEO die wordt gebruikt voor dit project zit een CAN-peripheral ingebouwd.

De CAN-peripheral ondersteunt:

- CAN-protocol versie 2.0 A, B Actief
- Bit snelheden tot 1 Mbit/s
- Transmissie
  - Drie transmit mailboxes
  - Configureerbare zendprioriteit
- Ontvangst
  - Twee ontvangs-FIFO's met drie niveaus
  - 14 schaalbare filterbanken
  - Identificatielijstfunctie
  - Configureerbare FIFO-overloop
- Optie voor tijd gestuurde communicatie
  - Uitschakelbare automatische heruitzendmodus
  - 16-bits vrijlopende timer
  - Tijdstempel verzonden in de laatste twee databytes
- Beheer
  - Maskerbare onderbrekingen
  - Software-efficiënte postvakttoewijzing op een uniek adresruimte

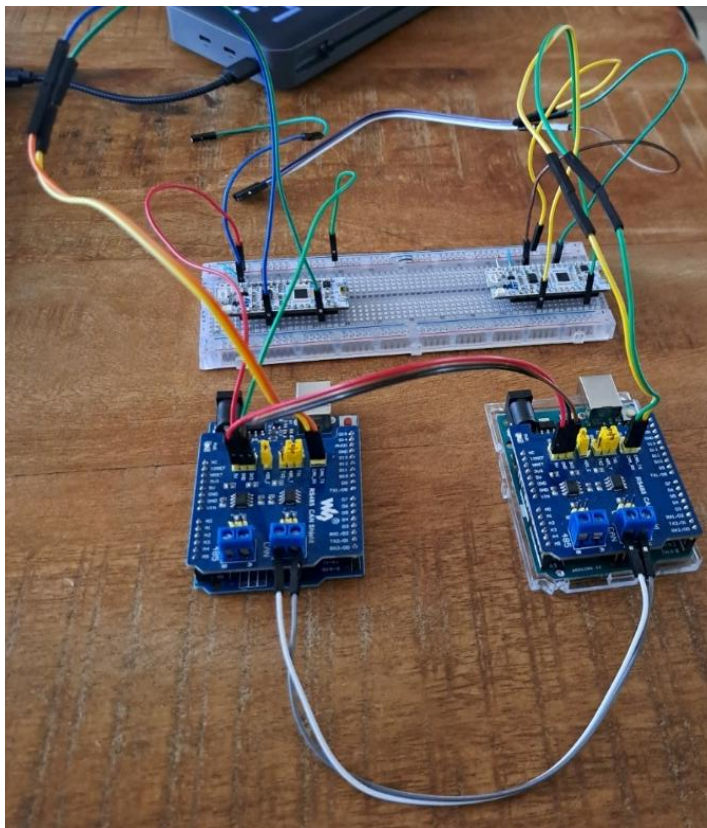
Een van de functies die handig zijn voor dit project is de identificatielijstfunctie. Deze kan de ID's in het systeem filteren en in dit geval alleen de berichten van de motorcontroller doorlaten. Dit is tijd efficiënt voor de microcontroller aangezien deze niet alle berichten hoeft te verwerken maar alleen degene van het juiste ID. Daarnaast is het basis protocol ook standaard geïmplementeerd, het ID-nummer en de data is daardoor al gescheiden. Voor de werking van het basis protocol zie: Wikipedia<sup>4</sup>.

Om dit systeem te testen is er een testopstelling gebouwd met twee L432KC boards en twee TTL naar CAN-converters. Zie “Figuur 10: Test opstelling CAN-peripheral” voor de testopstelling. De twee controllers vooraan zijn de converters. De linker microcontroller stuurt een bericht van een willekeurig karakter naar de converters. Hierdoor wordt de data omgezet van TTL naar CAN en vervolgens weer van CAN naar TTL. De data van de zender wordt ontvangen via de CAN-peripheral op de ontvanger. Tot slot wordt het ID dat ontvangen wordt en de data hiervan verstuurt via UART2 dat uitgelezen kan worden op een laptop/pc doormiddel van een USB naar TTL-converter. In “Figuur 10” wordt het eindresultaat weergegeven.

---

<sup>4</sup> (Wikipedia contributors, 2024)

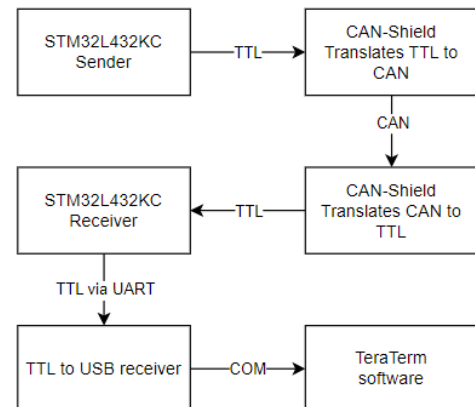




```

CAN ID: 0x123 DLC: 1 Data: p
CAN ID: 0x123 DLC: 1 Data: 0
CAN ID: 0x123 DLC: 1 Data: g
CAN ID: 0x123 DLC: 1 Data: l
CAN ID: 0x123 DLC: 1 Data: '
CAN ID: 0x123 DLC: 1 Data: C
CAN ID: 0x123 DLC: 1 Data: s
CAN ID: 0x123 DLC: 1 Data: 0
CAN ID: 0x123 DLC: 1 Data: M
CAN ID: 0x123 DLC: 1 Data: P
CAN ID: 0x123 DLC: 1 Data: @

```



Figuur 10: Test opstelling CAN-peripheral

### 2.3.4 NUCLEO-L432KC software IDE

Voor het programmeren van de L432KC wordt er gebruikt gemaakt van de Stm32CubeIDE<sup>5</sup>.

STM32CubeIDE integreert STM32-configuratie- en projectcreatiefuncties van STM32CubeMX om een alles-in-één gereedschapsservaring te bieden en de installatie- en ontwikkelingstijd te besparen. Na de selectie van een lege STM32 MCU of MPU, een vooraf geconfigureerde microcontroller of microprocessor, door het selecteren van een board of een voorbeeld, wordt het project aangemaakt en wordt de initialisatiecode gegenereerd. Op elk moment tijdens de ontwikkeling kan de gebruiker terugkeren naar de initialisatie en configuratie van de randapparatuur of middleware en de initialisatiecode opnieuw genereren zonder impact op de gebruikerscode.

STM32CubeIDE bevat build- en stackanalyzers die de gebruiker nuttige informatie verschaffen over de status van het project en de geheugeneisen.

STM32CubeIDE omvat ook standaard- en geavanceerde debuggingfuncties, waaronder weergaven van CPU-kernregisters, geheugen en randapparatuurregisters, evenals live variabelenbewaking, Serial Wire Viewer-interface of foutanalyse.

<sup>5</sup> (STM32CuBEIDE - STMicroelectronics, n.d.)

## 2.4 Feedback Systeem

Er is onderzoek gedaan naar verschillende actuatoren die gebruikt kunnen worden om feedback te geven over de status van de motorcontroller in de cockpit van de Solarboot. Hieronder worden drie mogelijke feedbackmechanismen besproken, elk met voor- en nadelen.

### 2.4.1 Led (Licht)

In de cockpit van de Solarboot zijn al meerdere leds aanwezig. Zoals vastgesteld in het vooronderzoek zijn er momenteel drie leds per drukknop. Deze kunnen worden aangepast om de status van de motorcontroller weer te geven, bijvoorbeeld door één led aan te laten gaan als de motorcontroller werkt, en een andere als deze niet werkt. De leds kunnen continu branden zonder storend te zijn.

#### **Voordelen:**

- Visuele indicatie zonder storende effecten.
- Makkelijk te integreren met de bestaande infrastructuur.

#### **Nadelen:**

- Verandering in status moet actief worden opgemerkt; de gebruiker moet hier alert op blijven.

### 2.4.2 Buzzer (Geluid)

Een buzzer kan worden geplaatst in de cockpit die een geluidssignaal geeft wanneer de motorcontroller aan- of uitgeschakeld wordt. De buzzer kan verschillende tonen genereren om de verschillende statussen van de motorcontroller aan te geven. Dit is handig omdat geluid waargenomen kan worden over grotere afstanden en door obstakels heen.

#### **Voordelen:**

- Goed hoorbaar, zelfs in een luidruchtige omgeving.
- Kan van op afstand worden waargenomen, bijvoorbeeld als de gebruiker zich niet direct in de cockpit bevindt.

#### **Nadelen:**

- Geluid kan storend zijn als het constant wordt gebruikt.

### 2.4.3 Trilmotor (Haptisch)

Een trilmotor kan worden gebruikt om zowel voelbare als hoorbare feedback te geven. De trilmotor creëert trillingen die de gebruiker kan voelen, wat een directere en meer persoonlijke vorm van feedback biedt dan licht of geluid. Dit kan vooral nuttig zijn wanneer visuele of auditieve signalen niet voldoende opvallen.

#### **Voordelen:**

- Directe, voelbare feedback die niet afhankelijk is van zicht of gehoor.
- Geschikt voor omgevingen met veel visuele en auditieve afleiding.

#### **Nadelen:**

- Trillingen kunnen als irritant worden ervaren als ze te vaak of te intens optreden.
- Minder geschikt voor situaties waar het over een afstand moet worden waargenomen.



#### 2.4.4 Conclusie

Het meest praktische is om een combinatie van feedbackmechanismen te gebruiken. Door zowel een buzzer als leds in te zetten, kunnen we de voordelen van beide systemen combineren en de nadelen compenseren. De buzzer biedt directe en hoorbare feedback, zelfs in lawaaierige omgevingen, terwijl de leds een continue visuele statusindicatie geven zonder storend te zijn. Hierdoor kan de gebruiker zowel op afstand als dicht bij de status van de motorcontroller in de gaten houden, wat leidt tot een robuust en veelzijdig feedbacksysteem.

### 3 Functioneel ontwerp

#### 3.1 Functionele specificatie

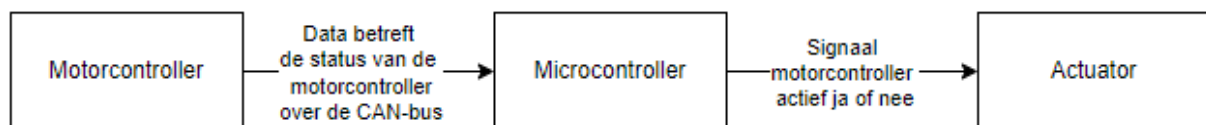
In Tabel 1 worden de eisen en wensen beschreven. Zowel de MoSCoW als de SMART methode wordt uitgeoefend op alle eisen. Wat inhoudt dat voor elke eis gespecificeerd wordt of dit moet, zal, kan of niet gebeurt. Daarnaast zijn alle beschrijvingen zo specifiek, meetbaar, acceptabel en realistische mogelijk beschreven.

Tabel 1: Functionele specificaties

SMART Functionele specificatie		
Nr.	MoSCoW	Beschrijving
<b>F1</b>	Must have	Er moet feedback gegeven worden aan de piloot/team over de motorcontroller.
<b>F1.1</b>	Could have	De piloot kan zien of de motorcontroller functioneert.
<b>F1.2</b>	Must have	Er wordt aan iedereen die zich rond de boot begeeft duidelijk gemaakt wanneer de motorcontroller aan gaat.
<b>F2</b>	Must have	De opstelling moet spatwaterdicht zijn.
<b>F3</b>	Must have	De opstelling voldoet aan de technische reglementen van de race organisatie.
<b>F4</b>	Must have	Er wordt gebruikt gemaakt van het bestaande communicatieprotocol over de CAN-bus in de boot.
<b>F5</b>	Should have	Er wordt gebruikt gemaakt van zo min mogelijk onderdelen als dat nodig is, in verband met het gewicht van de boot. Door de onderdelen te gebruiken die al aan boord zijn, wordt de hardware simpel gehouden en kan er minder kapotgaan.
<b>F6</b>	Should have	Er wordt zo min mogelijk stroom verbruikt door de opstelling.
<b>F7</b>	Won't have	Als de motorcontroller een fout heeft kan er een foutcode worden weergegeven.

#### 3.2 Datastroom

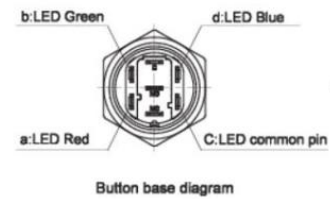
In "Figuur 11" is het functionele concept van het project weergegeven doormiddel van een datastroom:



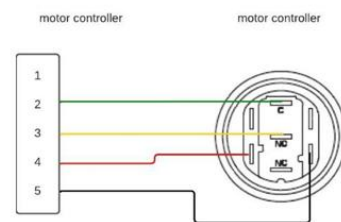
Figuur 11: Functionele datastroom motor controller feedback

### 3.3 Software werking

De software voor de motorcontroller zorgt voor een betrouwbare en efficiënte werking van de motorcontroller door bestaande systemen en componenten van de boot te gebruiken. Het geeft duidelijke feedback aan de piloot, het team en mensen rondom de boot, terwijl het systeem voldoet aan de technische reglementen en een minimaal stroomverbruik en gewicht garandeert.



De software dat zich in de microcontroller bevindt, leest de berichten van de motorcontroller via de CAN-bus. Deze berichten bevatten de juiste informatie om te bepalen of de motorcontroller aan staat en correct is geïnitieerd. De microcontroller stuurt aan de hand van deze informatie de actuatoren aan. Bijvoorbeeld: De RGB-led in de knop op het bedieningspaneel en/of een buzzer.



Figuur 12: Aansluiting drukknoppen

### 3.4 Hardware

Veel bestaande onderdelen aan de solarboot worden gebruikt voor dit project. De benodigde onderdelen voor dit project wordt hieronder weergegeven:

- Microcontroller, voor het uitvoeren van de functionaliteiten.
- Actuator, om feedback te generen.
- Bedieningspaneel, het bestaande paneel waar de actuator(en) geplaatst kan/kunnen worden.
- De motorcontroller zelf.
- CAN-bus
- Vertaler voor de microcontroller -> van CAN naar TTL
- Een spatwaterdichte behuizing voor de microcontroller.

Deze onderdelen zullen in het technisch ontwerp verder worden toegelicht.

## 4 Technisch ontwerp

### 4.1 Technische specificatie

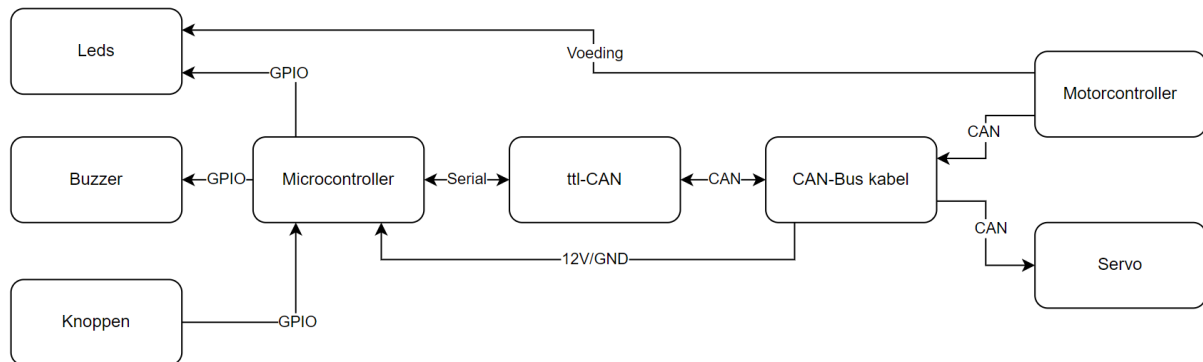
In Tabel 2 worden de technische eisen en wensen beschreven. Zowel de MoSCoW als de SMART methode wordt uitgeoefend op alle eisen. Wat inhoudt dat voor elke eis gespecificeerd wordt of dit moet, zal, kan of niet gebeurt. Daarnaast zijn alle beschrijvingen zo specifiek, meetbaar, acceptabel en realistische mogelijk beschreven.

Tabel 2: Technische specificaties

SMART Functionele specificatie		
Nr.	MoSCoW	Beschrijving
<b>T1</b>	Must have	Er wordt gebruik gemaakt van de al ingebouwde microcontroller, de STM32 NUCLEO L432KC
<b>T1.1</b>	Must have	De al bestaande functionaliteit van de microcontroller wordt behouden.
<b>T2.1</b>	Must have	Er wordt gebruik gemaakt van het communicatieprotocol CAN-bus om te communiceren met de motorcontroller
<b>T2.2</b>	Must have	De L432KC kan via de CAN-bus detecteren of de motorcontroller online is door het ID van de controller te herkennen in berichten
<b>T2.3</b>	Must have	Zodra de L432KC een bericht ontvangt van de motorcontroller via CAN-bus, wordt een timer in de microcontroller gereset. De timer zorgt voor de status van de motorcontroller (online of offline). Deze timer is ingesteld op nader te bepalen tijd.
<b>T2.4</b>	Could have	De L432KC kan via de berichten van de motorcontroller uitlezen en bepalen of de motorcontroller correct is geïnitieerd.
<b>T3</b>	Must have	De L432KC kan de led voor de motorcontroller in het huidige user interface aansturen om via deze weg feedback te geven.
<b>T4</b>	Must have	De L432KC kan een 3.3V buzzer activeren zodra er een event optreedt. Dit event kan zijn dat de motorcontroller online komt, foutcodes geeft of offline gaat.
<b>T4.1</b>	Must have	De buzzer komt in de spatwaterdichte behuizing die ook in de huidige situatie wordt gebruikt.

## 4.2 Architectuur

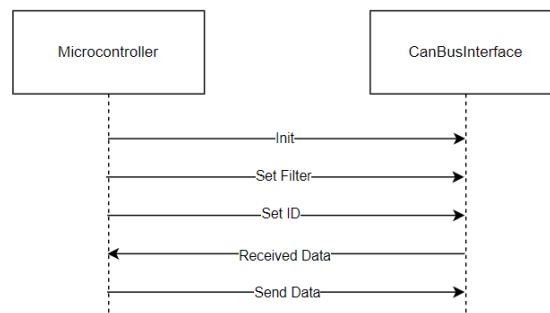
Hieronder zijn de CAN-bus aansluitingen weergegeven. Omdat er gebruik wordt gemaakt van een bestaande opstelling is in “Figuur 13” ook de servo van de boot opgegeven. De microcontroller die de functie van dit project erbij krijgt heeft momenteel als functie om de servo aan te sturen door middel van de knoppen op het bedieningspaneel. Voor het gedetailleerde aansluitschema, zie de kop “Aansluitschema”.



Figuur 13: Architectuur

## 4.3 Interfaces

### 4.3.1 Microcontroller – CAN-bus

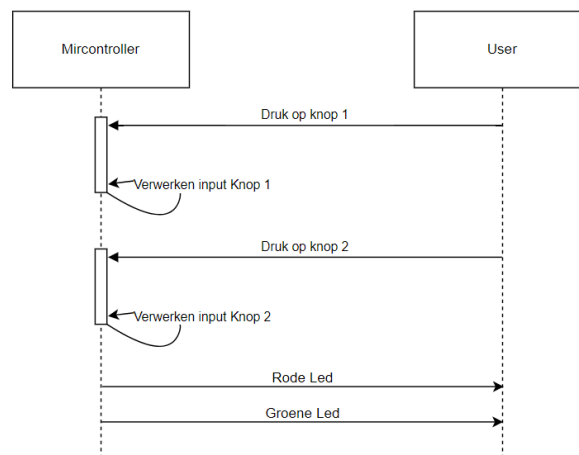


Figuur 14 : Interface Microcontroller - CAN-bus

De L432KC beschikt over ingebouwde hardware die de CAN-bus protocol ondersteunt. Het enige er gedaan moet worden is de bus initialiseren en een filter instellen om berichten te ontvangen via interrupts. Voor het verzenden van berichten moet er ook een eigen ID gespecificeerd worden.

De CAN-bus in de solarboat werkt met een bitrate van 250 kbps en maakt gebruik van een standaard ID-lengte.

#### 4.3.2 Microcontroller – Bedieningspaneel



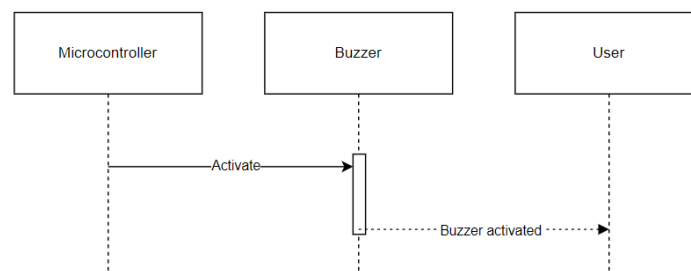
*Figuur 15 : Interface Microcontroller – Bedieningspaneel*

De gebruiker heeft twee knoppen om de hoek van de vleugel aan de achterkant van de boot aan te passen. Daarnaast is er een knop die direct op de motorcontroller is aangesloten om deze te starten. Deze knop is voorzien van een RGB led, waarbij de rode en groene kleuren worden gebruikt om de status van de motorcontroller aan de piloot door te geven.

- Rood betekent dat de motorcontroller geen voeding heeft en er geen CAN-bus berichten van de motorcontroller worden ontvangen.
- Oranje geeft aan dat de motorcontroller wel voeding heeft, maar nog geen berichten via de CAN-bus verstuurt.
- Groen betekent dat de motorcontroller voeding heeft en er ook CAN-bus berichten worden ontvangen.

De drukknoppen zijn 'normally open', en de leds werken op een spanning van 12 volt.

#### 4.3.3 Microcontroller – Actuator



*Figuur 16 : Interface Microcontroller - Actuator*

Om aan te geven dat de motorcontroller wordt ingeschakeld, maken we gebruik van een piezo buzzer. Deze wordt aangestuurd door de microcontroller en werkt op een spanning van 3,3 volt.

## 4.4 Hardware

### 4.4.1 Benodigde hardware

Voor de opstelling zijn de volgende onderdelen nodig:

- Bedieningspaneel: Knoppen en leds(12V)
- Buzzer (3.3V)
- NUCLEO L432KC microcontroller (3.3-12V)
- CAN-TTL converter
- CAN-bus
- Motorcontroller

Dit zijn de onderdelen die nodig zijn voor dit project. De andere onderdelen zijn al gerealiseerd op de boot en heeft geen extra waarde voor de opstelling.

### 4.4.2 Voedingsspanning

Er zijn verschillende soorten voedingsspanningen aanwezig op de boot. Deze komen uit één batterijpakket. De meest relevante voor dit systeem is de 12V. De L432KC microcontroller kan een ingang spanning hebben van 3.3V tot 12V. In de CAN-bus kabel in de solarboat is al een spanning aangesloten van 12V. Deze kan dus worden gebruikt voor de microcontroller en tevens voor de leds in het bedieningspaneel van de boot. Zie vooronderzoek “Huidige situatie” voor het aansluitschema.

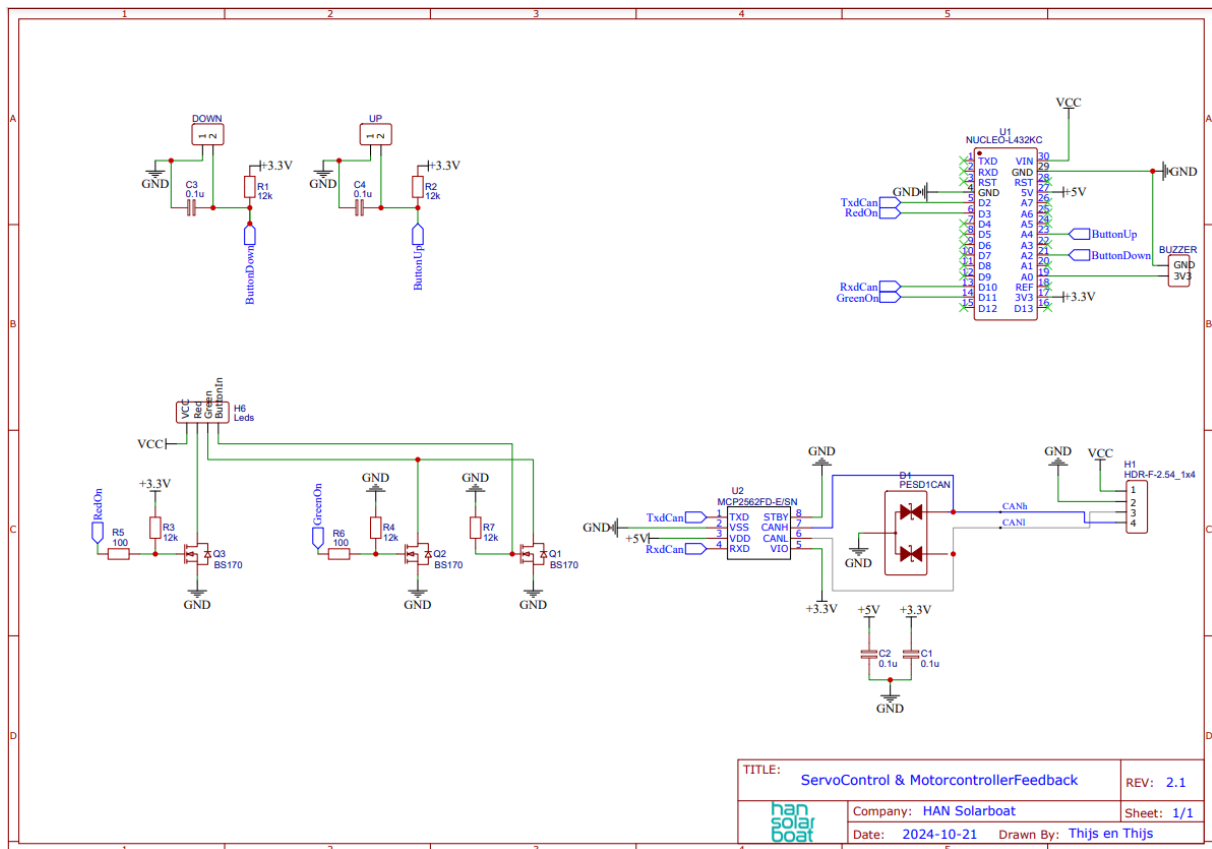
### 4.4.3 PCB

De PCB voor de microcontroller moet opnieuw worden ontworpen. Zie vooronderzoek “Huidige situatie” voor de PCB die is gebruikt voor alleen de servo functie. Deze PCB wordt uitgebreid met extra uitgang pinnen voor de buzzer en MOSFETS om de leds te schakelen met 12V. De reden dat de PCB wordt uitgebreid is, omdat de microcontroller ook de huidige functie moet blijven behouden. In de realisatie is het resultaat weergegeven.

#### 4.4.4 Aansluitschema

Voor het aansluitschema is er gebruik gemaakt van het ontwerpprogramma EasyEDA. Dit programma wordt namelijk binnen het team gebruikt.

Het aansluitschema voor de PCB moet bestaan uit de huidige situatie met de Servo werking en daarnaast de nieuwe functie voor de motorcontroller. De al eerdergenoemde hardware is aan dit schema toegevoegd. In "Figuur 17: Aansluitschema" is het resultaat weergegeven.

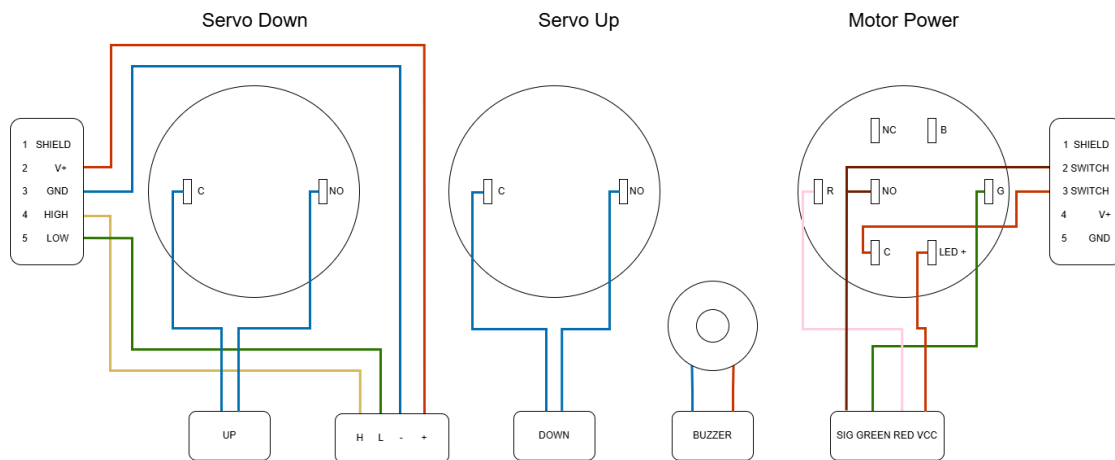


Figuur 17: Aansluitschema

Er is gekozen voor een extra MOSFET om het testen van de groene led mogelijk te maken. De groene led wordt volgens schema geschakeld door de voeding van de motorcontroller (de knop om de motorcontroller in te schakelen) en door een MOSFET van de microcontroller die schakelt wanneer de microcontroller operationeel is. De extra MOSFET schakelt alleen bij het opstarten zodat het zichtbaar is dat de leds werken. Het testen van de leds zijn belangrijk, aangezien deze de status van de motorcontroller aangeven. Zie "Led aansturing" voor welke kleur elke status van de motorcontroller aangeeft. Er is gekozen voor een N-channel (Negative) MOSFET. Zodra de microcontroller in storing valt, dan gaat de groene led uit en de rode led aan. In het volgende hoofdstuk is de stuklijst van het aansluitschema weergegeven.



Om een overzicht te creëren is er een aansluitschema gemaakt hoe de knoppen en CAN-aansluitingen aan de PCB gemaakt moeten worden. Deze is te zien in Figuur 18: Aansluitschema knoppen en CAN-connector.



Figuur 18: Aansluitschema knoppen en CAN-connector

#### 4.4.5 Benodigdheden

Hieronder zijn de onderdelen weergegeven voor de PCB. Aangezien er twee PCB's worden gemaakt voor dit project zijn er ook twee keer zoveel onderdelen besteld.

Tabel 3 : Benodigdheden

Nr.	Beschrijving	Artikelnummer	Leverancier	Aantal	Link
U1	NUCLEO L432KC microcontroller	-	Eigen voorraad	2	
U2	CAN-interface IC CAN Flexible Data Rate Transceiver	MCP2562FD-E/SN	Mouser	2	<a href="#">Link</a> <sup>6</sup>
C1-C2-C3-C4	Condensator 0.1uF SMD	-	Eigen voorraad	8	
D1	PESD1CAN Protection	PESD1CAN,215	Mouser	2	<a href="#">Link</a> <sup>7</sup>
Q1	MOSFET's BS170	MMBF170	Mouser	6	<a href="#">Link</a> <sup>8</sup>
R1-R2-R3-R4-R7	Resistor/Weerstand 12k Ohm SMD	-	Eigen voorraad	10	
R5-R6-R8	Resistor/Weerstand 100 Ohm SMD	-	Eigen voorraad	6	

<sup>6</sup> (Microchip MCP2562FD-E/SN, n.d.)

<sup>7</sup> (Nexperia PESD1CAN,215, n.d.)

<sup>8</sup> (Onsemi MMBF170, n.d.)

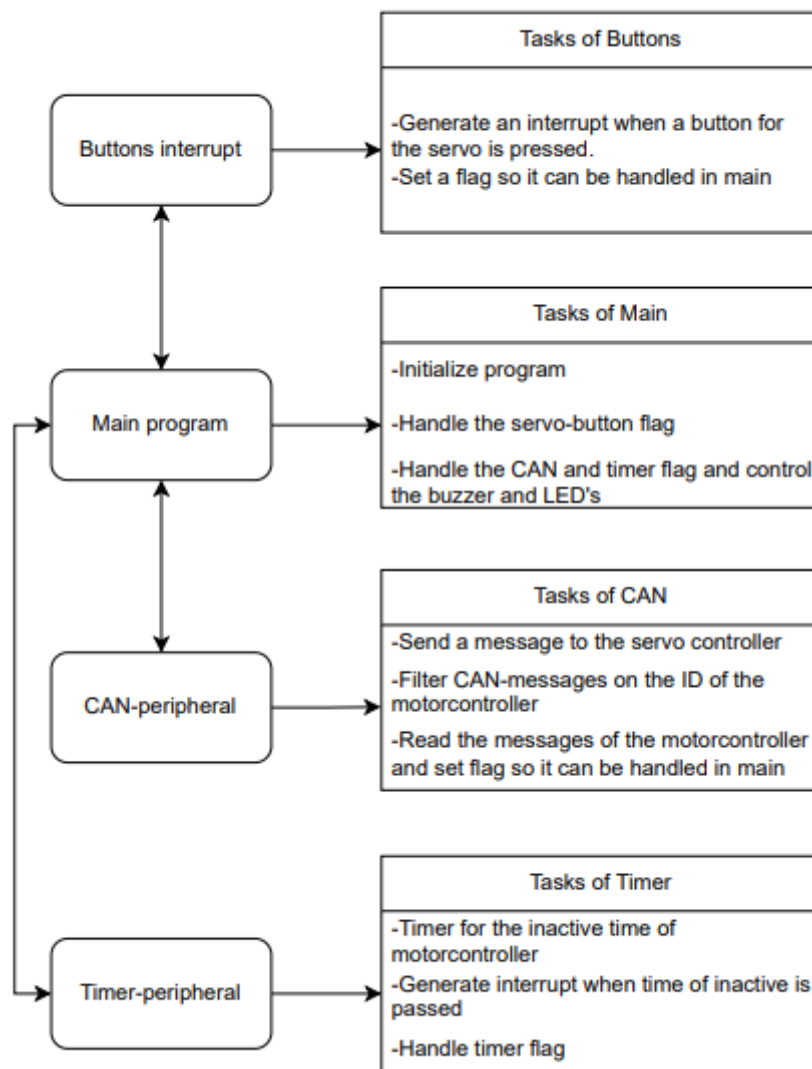
## 4.5 Software

Voor de software van de microcontroller is er gebruik gemaakt van het ontwikkelprogramma STM32CubeIDE. Dit is de aanbevolen IDE voor deze microcontroller en daarnaast is de huidige situatie ook gemaakt met dit programma.

Er kan gebruik worden gemaakt van het bestaande project waar de functie van de servo knoppen al is geïmplementeerd. In dit hoofdstuk wordt de software verder toegelicht.

### 4.5.1 Flowchart microcontroller

De taken van de microcontroller zijn overzichtelijk weergegeven in een flowchart. Aangezien er meerdere processen zijn binnen het programma van de microcontroller is de flowchart opgedeeld in verschillende deelprocessen met daarnaast de bijhorende taken. Zie “Figuur 19: Flowchart microcontroller”



Figuur 19: Flowchart microcontroller

#### 4.5.2 Led aansturing

Hieronder is een waarheidstabel weergegeven die de kleur van elke status van de motorcontroller toelicht.

Tabel 4 : Waarheidstabel aansturing

Motorcontroller voeding	Berichten over de CAN-bus	Led rood	Led groen	Status	Opmerking
0	0	1	0	Uit	Motorcontroller staat uit
0	1	0	0	-	Niet mogelijk, beide leds uit.
1	0	1	1	Opstarten	Oranje, wel voeding, geen CAN
1	1	0	1	Operationeel	Volledig werkend

#### 4.5.3 Timer-peripheral

De timer-peripheral wordt gebruikt om frequentie van de CAN-berichten te detecteren en om de het geluid van de buzzer aan te sturen. Door deze peripheral kunnen vertragingen in het programma worden ontweken. De timer peripheral kan na een gewenste periode een interrupt genereren die een variabele kan zetten of resetten. Hierdoor kan de microcontroller intussen tijd andere processen afhandelen. Er wordt gebruik gemaakt van twee timers. Eén voor de buzzer, deze genereert na de gewenste periode een interrupt die de buzzer activeert en deactiveert, om het juiste signaal te creëren. De andere timer wordt gebruikt om te controleren of de motorcontroller actief blijft. Dit wordt in het volgende kopje verder toegelicht.

#### 4.5.4 CAN-peripheral

De CAN-peripheral in de microcontroller kan zoals eerder genoemd, zie “CAN-peripheral NUCLEO-L432KC” CAN-berichten uitlezen en een interrupt genereren bij het ontvangen van data. Daarnaast kan de peripheral filteren op één of meerdere ID's van een apparaat. In dit geval de motorcontroller.

De interrupt kan in dit geval worden gebruikt om de timer van de microcontroller te resetten. Hierdoor wordt er elke keer dat er een bericht wordt ontvangen van de microcontroller de timer opnieuw gestart. Op het moment dat de motorcontroller is uitgeschakeld worden er ook geen nieuwe interrupts gegenereerd, aangezien er geen inkomende berichten meer zijn. Op het moment dat de timer de drempelwaarde heeft bereikt wordt de rode led ingeschakeld. Bij deze werking is gegeven dat de microcontroller om de 500 milliseconden een bericht stuurt over de CAN-bus. Zodra de CAN-peripheral geen berichten ontvangt voor 1,2 seconden is de drempelwaarde van de timer bereikt en gaat de rode led aan. De reden dat er voor 1,2 seconden is gekozen is om er zeker van te zijn dat er geen berichten over het hoofd worden gezien, of dat de motorcontroller door andere apparaten geen tijd heeft gehad om een bericht te sturen. Er is hierdoor een time-out van 2 berichten gecreëerd.

Daarnaast dient de peripheral voor de functie van de servo knoppen. De microcontroller stuurt berichten over de CAN-bus naar de servo met de juiste data afhankelijk van de knop die is ingedrukt op het bedieningspaneel.

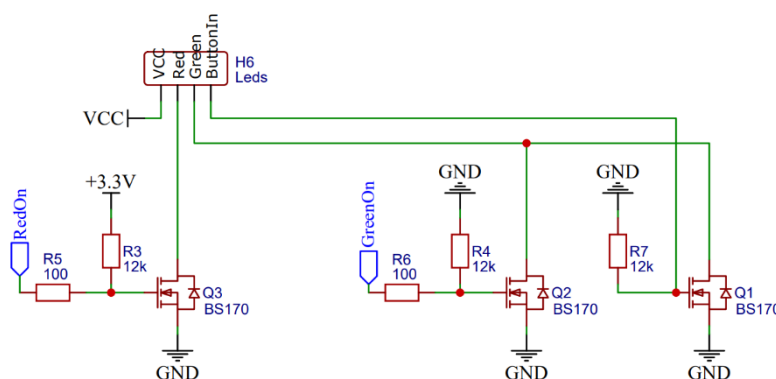
## 5 Realisatie

### 5.1 Aangebrachte verbeterpunten aansluitschema

#### 5.1.1 Gebruik van een MOSFET

Tijdens het realiseren van het ontwerp kwam de ontdekking dat er verkeerd gebruik wordt gemaakt van de MOSFET 's. De knop met RGB-leds die in de huidige situatie wordt gebruikt heeft de eigenschap dat alle leds één GND hebben. Hiervoor is er bedacht om 12V naar de juiste kleur led te sturen zodra deze nodig is. Echter is het de bedoeling dat een MOSFET pas wordt aangesloten NA de 'load' (de led). Met tot gevolg dat de MOSFET zoals in het eerdere schema was getekend niet werkt.

De oplossing hiervoor is om dezelfde knop te gebruiken, maar dan de versie waarbij de ledjes een gezamenlijke 12V heeft en aparte GND-pin per led. Het circuit is verbeterd en als volgt voor bijvoorbeeld de rode led:



Figuur 20: Correct aangesloten MOSFET

Hier is goed te zien dat de 'source' (output) van de MOSFET aan de GND zit verbonden.

Zie "Aansluitschema versie 2.0 en Aansluitschema versie 2.1" voor eerste en actuele aansluitschema

#### 5.1.2 Componenten gebruik

Bij het uitzoeken van de componenten tijdens de ontwerpfase is veel gekeken naar functionaliteit van de component. Aangezien het feit dat het aansluitschema aangepast moet worden om aan alle functionaliteiten te voldoen is er direct gekeken naar de keus van een aantal componenten. Er is gekeken naar de functionaliteit, vervangbaarheid, leverbaarheid en de kosten van een component. De volgende componenten zijn daardoor vervangen of verwijderd:

Tabel 5: Vervangen componenten

D2-D3	Gelijkrichters Hyperfst 2A 200V Fred Pt	VS-2EJH02-M3/6B	Mouser	4	<a href="#">Link<sup>9</sup></a>
Q1	MOSFET's PT5 150/25V Pch PowerTrench	FDN86265P	Mouser	2	<a href="#">Link<sup>10</sup></a>
U3	MOSFET's DUAL N-CHANNEL 40-V (D-S) 175C	SQJB46EP-T1_GE3	Mouser	2	<a href="#">Link<sup>11</sup></a>

De gelijkrichters zijn vanwege de andere knop overbodig geworden en dus verwijderd van het aansluitschema. De MOSFET's zijn vervangen voor een BS170 MOSFET. De reden hiervoor is dat de footprint voor de MOSFET veel overeenkomt met anderen, hierdoor is deze dus makkelijk aan te

<sup>9</sup> (Vishay VS-2EJH02-M3/6B, n.d.)

<sup>10</sup> (Onsemi FDN86265P, n.d.)

<sup>11</sup> (Vishay SQJB46EP-T1\_GE3, n.d.)

passen bij bijvoorbeeld het veranderen van microcontroller of knop, daarnaast is deze MOSFET heel goed leverbaar en kost per stuk maar 15 cent. Deze MOSFET wordt drie keer gebruikt op de PCB.

## 5.2 PCB-ontwerp

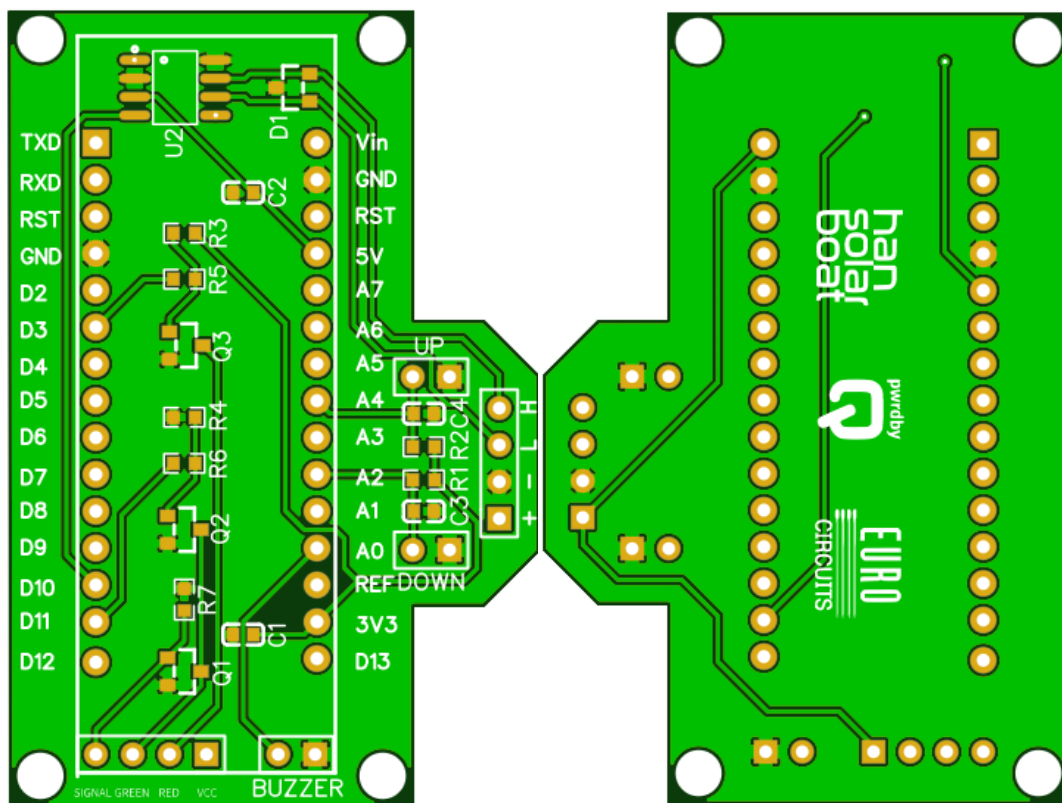
Het nieuwe PCB-ontwerp is gebaseerd op de originele PCB die we aantreffen aan het begin van het project (zie "Figuur 6: Huige Schema/PCB"). De afmetingen zijn zoveel mogelijk hetzelfde gehouden om de PCB in dezelfde behuizing te kunnen plaatsen.

De extra componenten zijn bewust verder uit elkaar geplaatst en in het midden om gemakkelijk de onderdelen te kunnen plaatsen tijdens het realiseren en eventueel te vervangen indien nodig.

De servo-buttons en CAN-functie is overgenomen van de originele PCB, hierbij is de buzzer en de aansturing van de leds toegevoegd. Daarnaast is er rekening gehouden met de kabels waar de PCB op aangesloten moet worden door aan de juiste kant de aansluiting te maken.

Er is gekozen om hetzelfde formaat weerstanden en condensatoren te gebruiken om de componenten van de originele pcb te kunnen hergebruiken. Daarnaast is voor de MOSFET's het formaat SOT-23 gebruikt, omdat deze maat erg gangbaar is voor MOSFET's.

Hieronder is het resultaat weergegeven:



Figuur 21 : PCB-ontwerp

### 5.3 Microcontroller L432KC software

### 5.3.1 Software working

De nieuwe functionaliteit is gerealiseerd door gebruik te maken van een timer en een twee interrupts die deze reset. De interrupt wordt gegenereerd zodra er een CAN-bericht van de motorcontroller afkomt. De code in de interrupt reset de timer. Dit zorgt ervoor dat als er geen berichten binnen komen vanaf de motorcontroller de timer niet gereset wordt en de timer een reset kan genereren. Deze interrupt zet een flag. Als deze flag hoog is zetten we de led in de knop op rood en anders op groen. Als de flag van hoog naar laag veranderd wordt er kort de buzzer aangezet.

Als er een error voorkomt in het programma wordt deze over de UART gestuurd en wordt de microcontroller via een software reset gereset. Dit voorkomt dat de servo knoppen onbruikbaar worden en zorgt ervoor dat als er een bootloop is, achterhaald kan worden wat er verkeerd is.

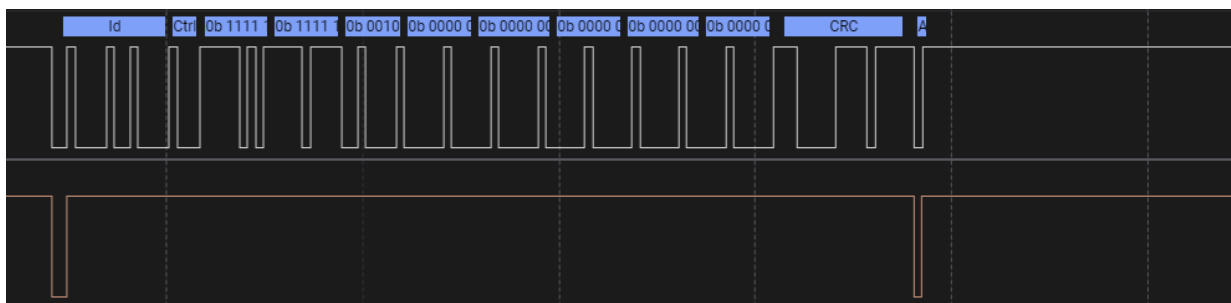
### 5.3.2 Prioriteiten in software

#### 5.3.2.1 Servo buttons

Om de werking van de servo zo goed mogelijk in stand te houden is er dezelfde code gebruikt als de huidige situatie. De enige aanpassing die er is gedaan is de prioriteit van de interrupts zo hoog mogelijk instellen in verband met de urgentie van deze functie. Zie “Testen en Resultaten” voor de testresultaten van deze functie.

#### 5.3.2.2 Auto re-transmission

Tijdens het realiseren is de ontdekking gedaan dat in de huidige software een bug bevindt, namelijk de auto re-transmissie. De knoppen van de servo worden periodiek geüpdatet, waarschijnlijk omdat eerdergenoemde functie uit staat. Deze periodieke update is waarschijnlijk overbodig geworden door het aanzetten van de functie, maar om geen onbekende functionaliteit te ontregelen is dit intact gehouden. Hieronder is de situatie gevisualiseerd met een logic analyzer:



*Figuur 22: Conflict tussen twee ID's waarbij de microcontroller heeft verloren omdat de microcontroller een lager ID heeft.*

Het onderste kanaal is van de microcontroller en de bovenste is de CAN-bus. Bij het versturen van de eerste bit wordt de ontdekking gedaan door de microcontroller dat er een ander apparaat, met hogere prioriteit, een bericht aan het versturen is. Dit is een normale situatie voor de CAN-bus. Bij het normale protocol zou het bericht van de microcontroller worden herhaald zodra de CAN-bus weer in de 'idle'-state is. Dit is dus niet het geval in deze afbeelding.

Deze configuratie is nu toegepast in het nieuwe softwarepakket voor de microcontroller. In “Testen en Resultaten” wordt deze functie getest.

## 5.4 Behuizing

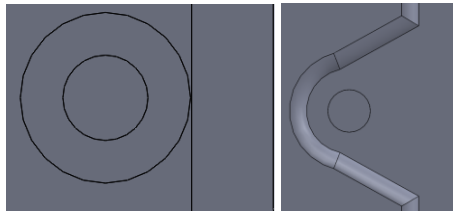
### 5.4.1 Verbeterpunten behuizing SOLIDWORKS

Er zijn verbeterpunten aangebracht aan de behuizing voor de microcontroller en de knoppen van dit project. Dit was nodig vanwege verschillende redenen, zoals: andere diameter van de knoppen, gaten klopten niet voor het monteren van de PCB en diameter voor de CAN-bus aansluitingen.

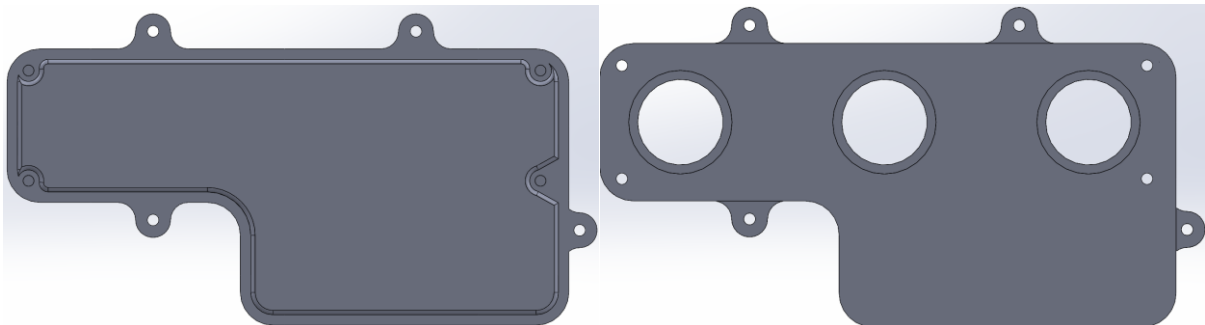
De volgende dingen zijn aangepast:

- Verstevinging bevestigingspunten. De bevestigingspunten zaten eerst alleen aan de bodemplaat, deze zijn nu ook aan de want vastgemaakt om extra stevigheid te bieden. Zie "Figuur 23".
- Verstevinging en spatwaterdicht maken van de randen van de behuizing. Door de randen van de behuizing rond te maken met een kleine rondingshoek van 2 graden zorgt voor extra stevigheid in de hoeken en maakt het hierdoor ook waterbestendiger.
- Gaten groter gemaakt. Voor de CAN-bus aansluitingen en voor de knoppen.
- Gaten voor de PCB kloppend gemaakt en verstevigd.

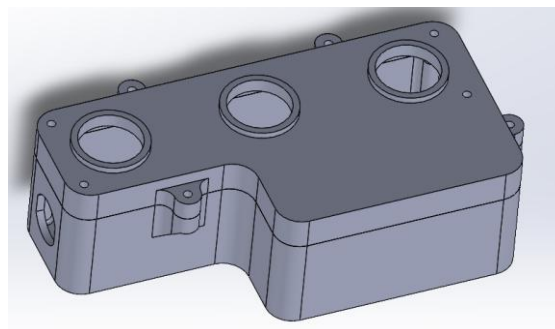
De behuizing is gemaakt in SOLIDWORKS. Hieronder is het eindresultaat weergegeven.



*Figuur 23: Voor en na aanpassing van verstevinging van de bevestigingspunten en randen*



*Figuur 24: Eindresultaat weergegeven in SOLIDWORKS*



*Figuur 25: Eindresultaat in geassembleerde staat*



### 5.4.2 Knoppen

De knoppen van de huidige situatie hadden geen standaardkleur en/of formaat. Er is daarom gekozen om over te stappen op andere soorten in verband met de functie voor de knop van de motorcontroller en voor het uiterlijk. De knop voor de motorcontroller heeft een diameter van 19 mm en is gemaakt van RVS. Deze eigenschappen zijn ook op de knoppen voor de servo toegepast.

Voor de servo knoppen zijn er knoppen gekozen met een opstaande rand, omdat de bestuurder van de boot makkelijk de knoppen moet kunnen bedienen tijdens de wedstrijden. Ook zijn deze knoppen momentary, terwijl de knop voor de motorcontroller latch is. De huidige knoppen voor de servo bevatten leds, deze zijn overbodig geworden omdat deze knoppen aan hetzelfde systeem vast zit als de motorcontroller knop, deze heeft wel leds.

De latch knop voor de motorcontroller is door de fabrikant zo ontworpen dat er maar een kleur te gelijk kan branden. Ook al staan er meerdere leds aan eentje geeft maar werkelijk licht. Na testen bleek dat rood prioriteit had over groen en groen prioriteit had over blauw. Dit zorgt ervoor dat er geen oranje getoond kan worden. In overleg met de opdrachtgever is de functionaliteit voor de kleur oranje samengevoegd met die van rood (Zie "Tabel 4 : Waarheidstabel aansturing").

Tabel 6: Nieuwe knoppen

Omschrijving	Artikel	Artikelnummer	Link
Knop voor de motorcontoller	TRU COMPONENTS TC-7494632 TC-LAS1-AGQ-11ZE/42RGB/	1873658 - 8J	<a href="#">Link<sup>12</sup></a>
Knop voor de servo	TRU COMPONENTS 701953 GQ19H-10/J/S	701953 - 8J	<a href="#">Link<sup>13</sup></a>

Hieronder zijn de gekozen knoppen weergegeven:



Figuur 26: Knop motorcontroller



Figuur 27: Knop servo

<sup>12</sup> (TRU COMPONENTS TC-7494632 TC-LAS1-AGQ-11ZE/42RGB/, n.d.)

<sup>13</sup> (TRU COMPONENTS 701953 GQ19H-10/J/S, n.d.)



## 5.5 Eindresultaat

### 5.5.1 Eindproduct

De nieuwe versie met alle aanpassingen hierboven genoemd ziet er als volgt uit:



Figuur 28: Eindresultaat

Deze versie heeft dezelfde formaten als de huidige versie, de knoppen zijn groter, er is een buzzer toegevoegd, de kabels zijn beter weggewerkt door een aangepast PCB-design en deze versie heeft dezelfde functionaliteiten met de extra motorcontroller detectie functie.

### 5.5.2 Stroomverbruik

Het stroomverbruik voor deze opstelling is ook belangrijk om te weten. De resultaten zijn hieronder weergegeven:

Tabel 7: Verbruik verschillende componenten

Onderdeel	Berekening	Vermogen
Motorcontroller leds	$12V \times 11 \text{ mA}$	0,13 Watt
Microcontroller	$12V \times 47\text{mA}$	0,564 Watt

De overige onderdelen op de PCB zoals de MOSFETS en de CAN-chip verbruiken in een totaal minder dan 1 milliwatt. Hetzelfde geldt voor het vermogen dat nodig is voor de communicatie over de CAN-bus. Het totale verbruik van deze opstelling is dus ongeveer: 0,75 Watt.

## 6 Verifiëren en testen

### 6.1 Test plan

In dit hoofdstuk wordt besproken hoe de testen worden geformuleerd en uitgevoerd.

De testen worden als volgt geformatteerd. Eerst krijgt de test een naam en omschrijving waarin staat wanneer de test geslaagd of gefaald is. Hierna wordt beschreven of de test fysiek of in software plaats vindt ook staat er in het test formaat wat de uitkomst van de test kan zijn.

Na het opstellen van de test wordt hij uitgevoerd en wordt het resultaat met mogelijke bijvoegingen in het Resultaten Tabel gezet.

De volgende tabel is een voorbeeld van zowel het Test tabel als het resultaten tabel.

#### 6.1.1 (Voorbeeld) Test Tabel

Tabel 8: Voorbeeld test tabel

Nr.	Test naam	Test Omschrijving	Test Uitvoering	Test formaat
<b>T1</b>	Knoppen Paneel	De volgende testen verifiëren de werking van het knoppen paneel door zowel software- als hardware matig de werking te testen.	Software/ Hardware testen.	
<b>T1.1</b>	Conformatie knop	Bij activering van de conformatie knop wordt er d.m.v. een softwareprotocol een signaal naar de microcontroller gestuurd. Als dit correct signaal aankomt bij de microcontroller is de test geslaagd	Software	Geslaagd/ Gefaald
<b>T2</b>	Ect.			

#### 6.1.2 (Voorbeeld) Resultaten Tabel

De testen worden gelijk met het Test Tabel geformatteerd en zowel de bijlage van de test als het resultaat zelf worden in deze tabel gezet. Op het moment dat een test faalt wordt in de bijlage een (mogelijk) rede en uitleg gegeven waarom de test niet is geslaagd. Bij geval van een test hoofdstuk (kopje met meerdere individuele testen) wordt er een V in het resultaat gezet om het begin van de testen weer te geven.

Tabel 9: Voorbeeld resultaten tabel

Nr.	Test naam	Bijlage Resultaat	Test Resultaat
<b>T1</b>	Knoppen Paneel	Alle testen rondom het knoppen paneel.	Geslaagd
<b>T1.1</b>	Conformatie knop	Bij activering van de conformatie knop wordt het juiste signaal ontvangen bij de microcontroller.	Geslaagd
<b>T2</b>	Ect.	(Bijlagen met (mogelijke) rede waarom de test gefaald is.)	Gefaald

## 6.2 Testen en Resultaten

Alle testen worden uitgebreid uitgelegd en verwerkt in de onderstaande tabel. In de tweede tabel worden alle resultaten, eventuele obstakels en onverwachte uitkomsten verwerkt.

### 6.2.1 Test Tabel

Tabel 10: Test tabel

Nr.	Test naam	Test Omschrijving	Test Uitvoering	Test format
<b>T1</b>	Detectie van de motorcontroller.	De microcontroller kan detecteren dat de motorcontroller CAN-berichten stuurt.	Hardware test. De groene lamp gaat branden	Geslaagd/ Gefaald
<b>T2</b>	Detectie van wegvallen van motorcontroller	De microcontroller kan detecteren dat de motorcontroller is gestopt met CAN-berichten sturen.	Hardware test. De rode lamp gaat branden	Geslaagd/ Gefaald
<b>T3</b>	Servo functionaliteit	De microcontroller stuurt nog steeds de servo correct aan.	Software. Uitlezen via CAN-bus	Geslaagd/ Gefaald
<b>T4</b>	Auto re-transmissie.	De microcontroller herhaalt het bericht op het moment dat deze is onderbroken door een ander apparaat op de CAN-bus	Software. Uitlezen via CAN-bus	Geslaagd/ Gefaald
<b>T5</b>	Testfunctie van de leds	De microcontroller doet bij opstarten de testfunctie en schakelt 4 keer tussen groen en rood.	Hardware test. De leds schakelen tussen rood en groen bij opstarten	Geslaagd/ Gefaald
<b>T6</b>	Buzzer werking	De buzzer maakt een geluid op het moment dat de motorcontroller wordt gedetecteerd op de CAN-bus	Hardware test. Bij het aansluiten van de CAN-bus gaat de buzzer aan.	Geslaagd/ Gefaald
<b>T7</b>	Microcontroller start opnieuw op	Microcontroller start uit zichzelf opnieuw op bij het optreden van een fout. Deze functie is ingebouwd vanwege veiligheid.	Hardware test. Loskoppelen van de CAN-bus.	Geslaagd/ Gefaald
<b>T8</b>	Servo knoppen	De servo knoppen worden gedetecteerd bij het indrukken.	Software test. Via CAN-bus kijken of de knoppen worden gedetecteerd.	Geslaagd/ Gefaald
<b>T9</b>	Motorcontroller knop	De knop voor de motorcontroller is functioneel en kan de motorcontroller en pomp in-/uitschakelen	Hardware test. De pomp gaat draaien bij het indrukken van de knop	Geslaagd/ Gefaald

## 6.2.2 Test Resultaten

Tabel 11: Resultaten tabel

Nr.	Test naam	Bijlage Resultaat	Test Resultaat
<b>T1</b>	Detectie van de motorcontroller.	De groene lamp gaat branden.	Geslaagd
<b>T2</b>	Detectie van wegvallen van motorcontroller	De rode lamp gaat branden.	Geslaagd
<b>T3</b>	Servo functionaliteit	 <p>De functionaliteit is niet veranderd vergeleken met de huidige versie.</p>	Geslaagd
<b>T4</b>	Auto re-transmissie.	De auto re-transmissie is gemeten door de logic-analyser en functioneert correct.	Geslaagd
<b>T5</b>	Testfunctie van de leds	De microcontroller schakelt met opstarten tussen rood en groen.	Geslaagd
<b>T6</b>	Buzzer werking	De buzzer ging aan op het moment dat de motorcontroller werd geactiveerd.	Geslaagd
<b>T7</b>	Microcontroller start opnieuw op	Bij het loskoppelen van de CAN-bus start de microcontroller net zolang opnieuw op tot de CAN-bus weer is aangesloten en hersteld zichzelf. Bij andere foutcodes gebeurt hetzelfde en de microcontroller loopt dus niet vast.	Geslaagd
<b>T8</b>	Servo knoppen	De knoppen zorgen voor verandering in data over de CAN-bus	Geslaagd
<b>T9</b>	Motorcontroller knop	De pomp gaat draaien bij het indrukken van de knop	Geslaagd

## 7 Conclusies en aanbevelingen

### 7.1 Conclusie

Kijkende naar de functionele en technische eisen zijn aan deze eisen voldaan. Er zijn nog wel een paar dingen die verbeterd zouden kunnen worden, zie “Aanbevelingen”.

Het eindresultaat is dus in staat om de motorcontroller te detecteren op de CAN-bus en heeft de vorige functies behouden. De vorige functies zijn het bedienen van de servo en de motorcontroller in-/uitschakelen. Daarnaast is de behuizing verstevigd en zijn de soort knoppen nu van hetzelfde model.

De functionele eisen zijn hier nogmaals weergegeven:

Tabel 12: Functionele eisen

SMART Functionele specificatie			
Nr.	MoSCoW	Beschrijving	Behaald?
<b>F1</b>	Must have	Er moet feedback gegeven worden aan de piloot/team over de motorcontroller.	<b>Ja</b>
<b>F1.1</b>	Could have	De piloot kan zien of de motorcontroller functioneert.	<b>Ja</b>
<b>F1.2</b>	Must have	Er wordt aan iedereen die zich rond de boot begeeft duidelijk gemaakt wanneer de motorcontroller aan gaat.	<b>Ja</b>
<b>F2</b>	Must have	De opstelling moet spatwaterdicht zijn.	<b>Ja</b>
<b>F3</b>	Must have	De opstelling voldoet aan de technische reglementen van de race organisatie.	<b>Ja</b>
<b>F4</b>	Must have	Er wordt gebruikt gemaakt van het bestaande communicatieprotocol over de CAN-bus in de boot.	<b>Ja</b>
<b>F5</b>	Should have	Er wordt gebruikt gemaakt van zo min mogelijk onderdelen als dat nodig is, in verband met het gewicht van de boot. Door de onderdelen te gebruiken die al aan boord zijn, wordt de hardware simpel gehouden en kan er minder kapotgaan.	<b>Ja</b>
<b>F6</b>	Should have	Er wordt zo min mogelijk stroom verbruikt door de opstelling.	<b>Ja</b>
<b>F7</b>	Won't have	Als de motorcontroller een fout heeft kan er een foutcode worden weergegeven.	<b>Nee</b>

### 7.2 Worst-case scenario

De worst-case scenario in deze situatie is dat de microcontroller geen spanning meer krijgt en uit gaat. In dit geval zou de led voor de motorcontroller uitgaan en is het niet meer mogelijk om de servo te bedienen. Dit is dezelfde worst-case scenario als de vorige versie, met bij het eindresultaat als verbetering dat de microcontroller niet in storing kan vallen, doordat deze nu automatisch opnieuw opstart.

### 7.3 Aanbevelingen

Eventuele verbeterpunten zouden zijn:

- Power detectie implementeren
- De code voor de servo optimaliseren

#### 7.3.1 Power detectie

Momenteel wordt er alleen gekeken naar de CAN-bus voor de motorcontroller. Zodra er berichten van de motorcontroller verschijnen, wordt de groene led weergegeven en gaat de buzzer af in de cockpit. Geen berichten van de motorcontroller van de CAN-bus, geeft een rode kleur op het bedieningspaneel. Voor extra veiligheid zou er gekeken kunnen worden of de motorcontroller wel is ingeschakeld. Hiermee kan je dus een extra foutcode implementeren die kan aangeven dat er iets mis is, of dat de motorcontroller simpelweg uit staat.

Deze optie is al wel op de vernieuwde PCB ingebouwd, maar wordt nog niet toegepast. Dit zou dus een makkelijke en goede verbetering zijn.

#### 7.3.2 Code voor de servo

Tijdens het vooronderzoek is er de ontdekking gedaan hoe de vorige versie communiceert met de servo achterin de boot. Deze functie is bewust hetzelfde gelaten, maar zou geoptimaliseerd kunnen worden. Nu wordt er een waarde tussen 0 en 12 stuurt, dit past binnen één byte, terwijl er acht bytes worden verstuurd. Dit neemt extra ruimte in op de CAN-bus en dat is eigenlijk wat er voorkomen moet worden. Een ander opvallende keuze is dat de byte waar de waarde instaat naar het tweede byte van de acht bytes wordt geschoven.

Het verbetervoorstel zou zijn om slecht één byte te sturen die de servo ontvangt en uitleest. Hiermee wordt ruimte bespaard op de CAN-bus, is de code overzichtelijker en zorgt dit voor optimalisatie.

Zie hieronder voor het huidige CAN-bericht, uitgelezen met de logic-analyzer:



*Figuur 29: CAN-bericht van de microcontroller naar de servo*

Volgens protocol wordt eerst het 11-bit ID gestuurd gevolgd door het control byte. Hierna volgt de data waarbij alleen de tweede byte gevuld is, met in dit geval 8, de servo waarde. Het zogenoemde bit-stuffing zorgt ervoor dat er extra flanken zijn, maar zijn eigenlijk onnodig.

## 8 Verwijzingen

SLS-Sinusleistungssteller. (n.d.).

[https://www.sinusleistungssteller.de/en\\_SLRI\\_Menu.html](https://www.sinusleistungssteller.de/en_SLRI_Menu.html)

NUCLEO-L432KC - STMicroelectronics. (n.d.). STMicroelectronics.

<https://www.st.com/en/evaluation-tools/nucleo-l432kc.html>

STM32CubeIDE - STMicroelectronics. (n.d.). STMicroelectronics.

<https://www.st.com/en/development-tools/stm32cubeide.html>

Microchip MCP2562FD-E/SN. (n.d.). Mouser Electronics.

<https://nl.mouser.com/ProductDetail/Microchip-Technology/MCP2562FD-E-SN?qs=YlpJWyuQ3POBNBSLSm7%2F1w%3D%3D>

Nexperia PESD1CAN,215. (n.d.). Mouser Electronics.

<https://nl.mouser.com/ProductDetail/Nexperia/PESD1CAN215?qs=LOCUfHb8d9vzREMT8aTFcw%3D%3D>

Onsemi MMBF170. (n.d.). Mouser Electronics. [https://nl.mouser.com/ProductDetail/onsemi-](https://nl.mouser.com/ProductDetail/onsemi-Fairchild/MMBF170?qs=FOlmdCx%252BAA2suHltufvdcg%3D%3D&mgh=1&utm_id=20331598300&gad_source=1&gclid=Cj0KCQiAIsy5BhDeARIsABRc6ZvPRb-83xFFr_lIi2as1MA2sJtjKrlLm_w7BJF-Fuxl5foKzbVbS_YaAjswEALw_wcB)

[Fairchild/MMBF170?qs=FOlmdCx%252BAA2suHltufvdcg%3D%3D&mgh=1&utm\\_id=20331598300&gad\\_source=1&gclid=Cj0KCQiAIsy5BhDeARIsABRc6ZvPRb-83xFFr\\_lIi2as1MA2sJtjKrlLm\\_w7BJF-Fuxl5foKzbVbS\\_YaAjswEALw\\_wcB](https://nl.mouser.com/ProductDetail/onsemi-Fairchild/MMBF170?qs=FOlmdCx%252BAA2suHltufvdcg%3D%3D&mgh=1&utm_id=20331598300&gad_source=1&gclid=Cj0KCQiAIsy5BhDeARIsABRc6ZvPRb-83xFFr_lIi2as1MA2sJtjKrlLm_w7BJF-Fuxl5foKzbVbS_YaAjswEALw_wcB)

Vishay VS-2EJH02-M3/6B. (n.d.). Mouser Electronics. [https://nl.mouser.com/ProductDetail/Vishay-](https://nl.mouser.com/ProductDetail/Vishay-Semiconductors/VS-2EJH02-M3-6B?qs=ei%2FvZTf36ZyE7L9vyeduYg%3D%3D)

[Semiconductors/VS-2EJH02-M3-6B?qs=ei%2FvZTf36ZyE7L9vyeduYg%3D%3D](https://nl.mouser.com/ProductDetail/Vishay-Semiconductors/VS-2EJH02-M3-6B?qs=ei%2FvZTf36ZyE7L9vyeduYg%3D%3D)

Onsemi FDN86265P. (n.d.). Mouser Electronics. [https://nl.mouser.com/ProductDetail/onsemi-](https://nl.mouser.com/ProductDetail/onsemi-Fairchild/FDN86265P?qs=wsLAKTqgg6igijlSDrHynQ%3D%3D)

[Fairchild/FDN86265P?qs=wsLAKTqgg6igijlSDrHynQ%3D%3D](https://nl.mouser.com/ProductDetail/onsemi-Fairchild/FDN86265P?qs=wsLAKTqgg6igijlSDrHynQ%3D%3D)

Vishay SQJB46EP-T1\_GE3. (n.d.). Mouser Electronics. [https://nl.mouser.com/ProductDetail/Vishay-](https://nl.mouser.com/ProductDetail/Vishay-Siliconix/SQJB46EP-T1_GE3?qs=DPoM0jnrROUm2bCVmiHcfig%3D%3D)

[Siliconix/SQJB46EP-T1\\_GE3?qs=DPoM0jnrROUm2bCVmiHcfig%3D%3D](https://nl.mouser.com/ProductDetail/Vishay-Siliconix/SQJB46EP-T1_GE3?qs=DPoM0jnrROUm2bCVmiHcfig%3D%3D)

TRU COMPONENTS TC-7494632 TC-LAS1-AGQ-11ZE/42RGB/. (n.d.). Conrad.

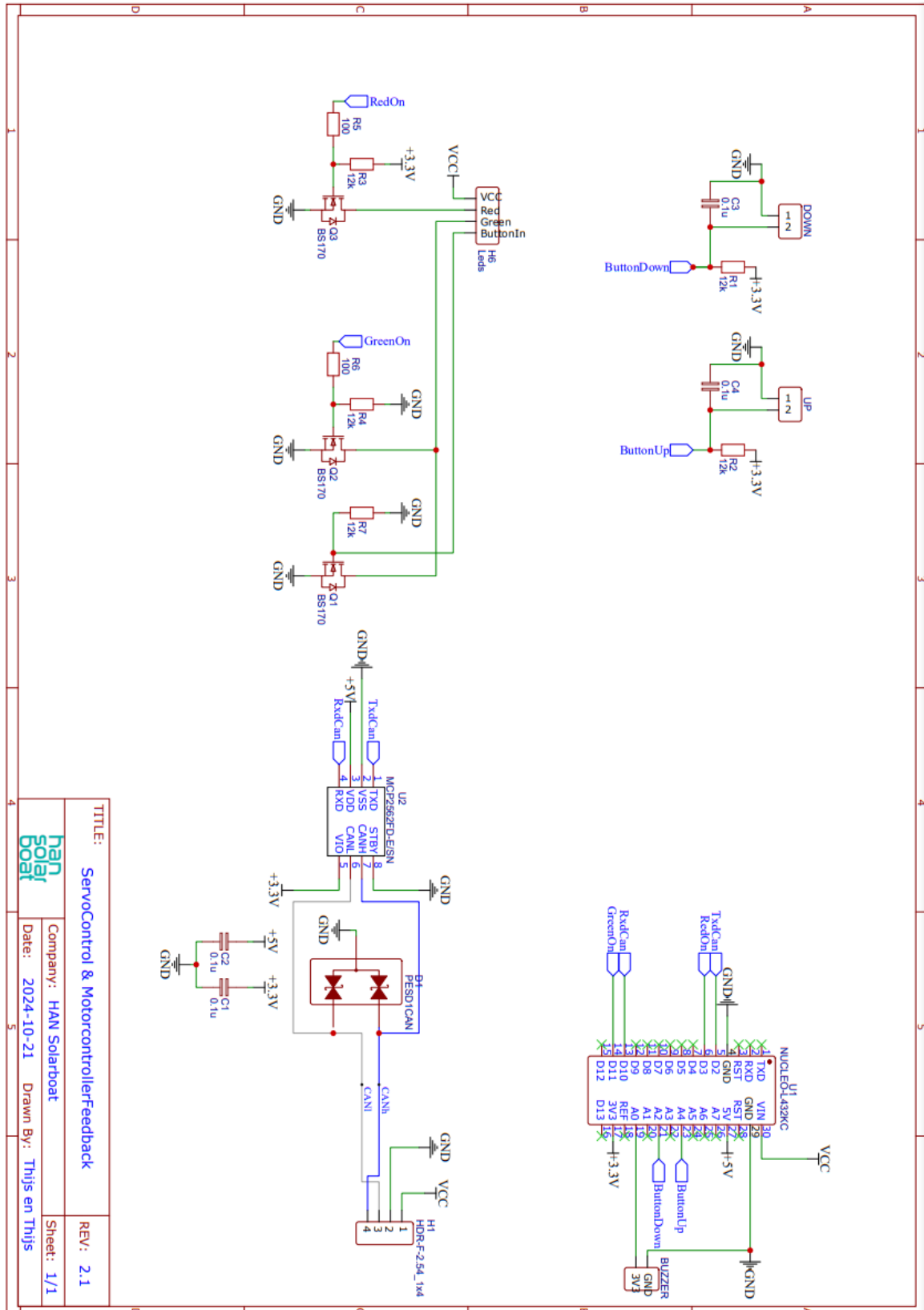
<https://www.conrad.nl/nl/p/tru-components-tc-7494632-tc-las1-agq-11ze-42rgb-vandalismebestandige-drukschakelaar-250-v-ac-3-a-1x-uit-aan-continu-ro-1873658.html?refresh=true>

TRU COMPONENTS 701953 GQ19H-10/J/S. (n.d.). Conrad Electronic.

<https://www.conrad.nl/nl/p/tru-components-701953-gq19h-10-j-s-vandalismebestandige-druktoets-48-v-dc-2-a-1x-uit-aan-moment-1-stuk-s-701953.html>

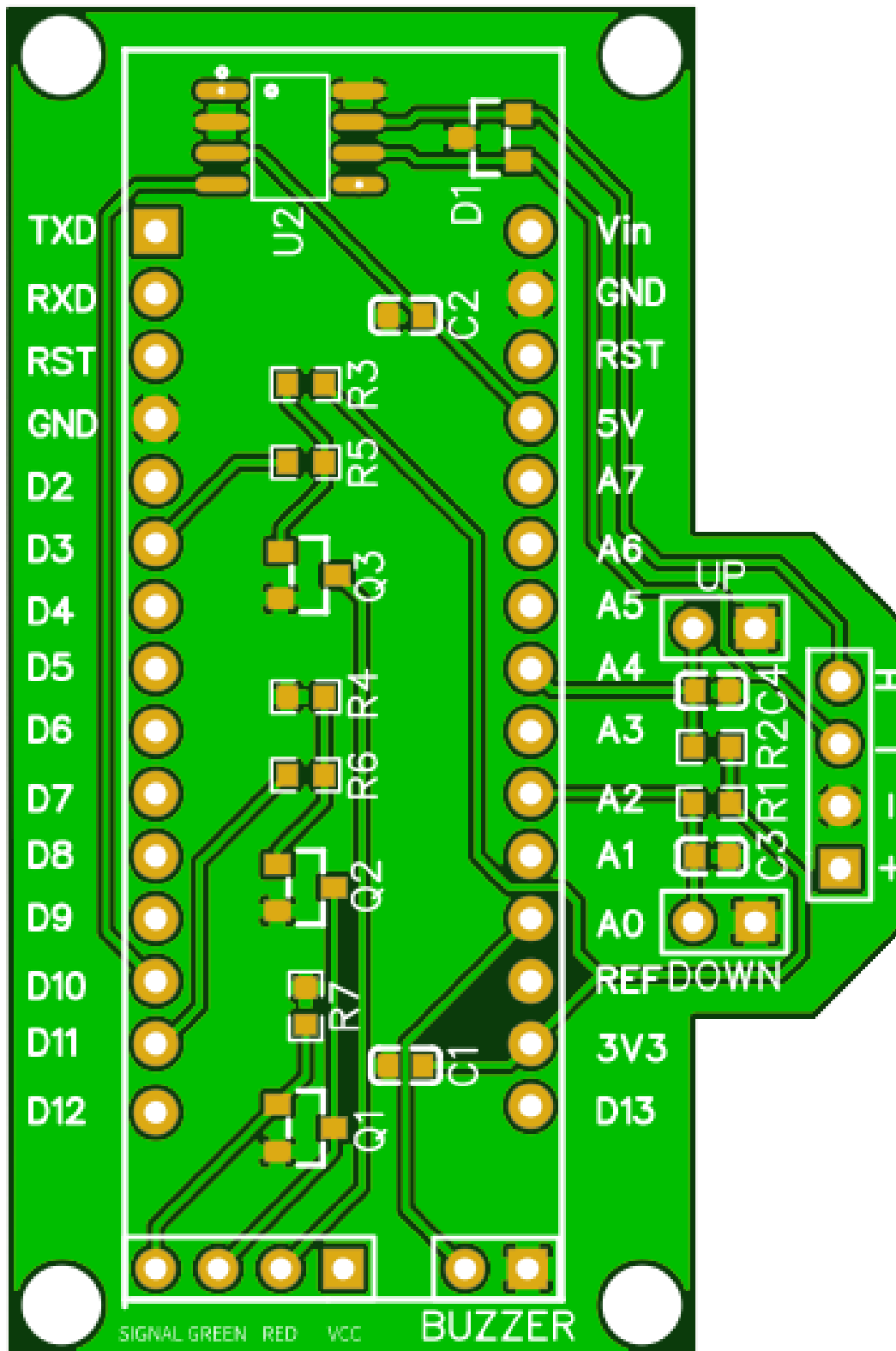
## 9 Bijlage

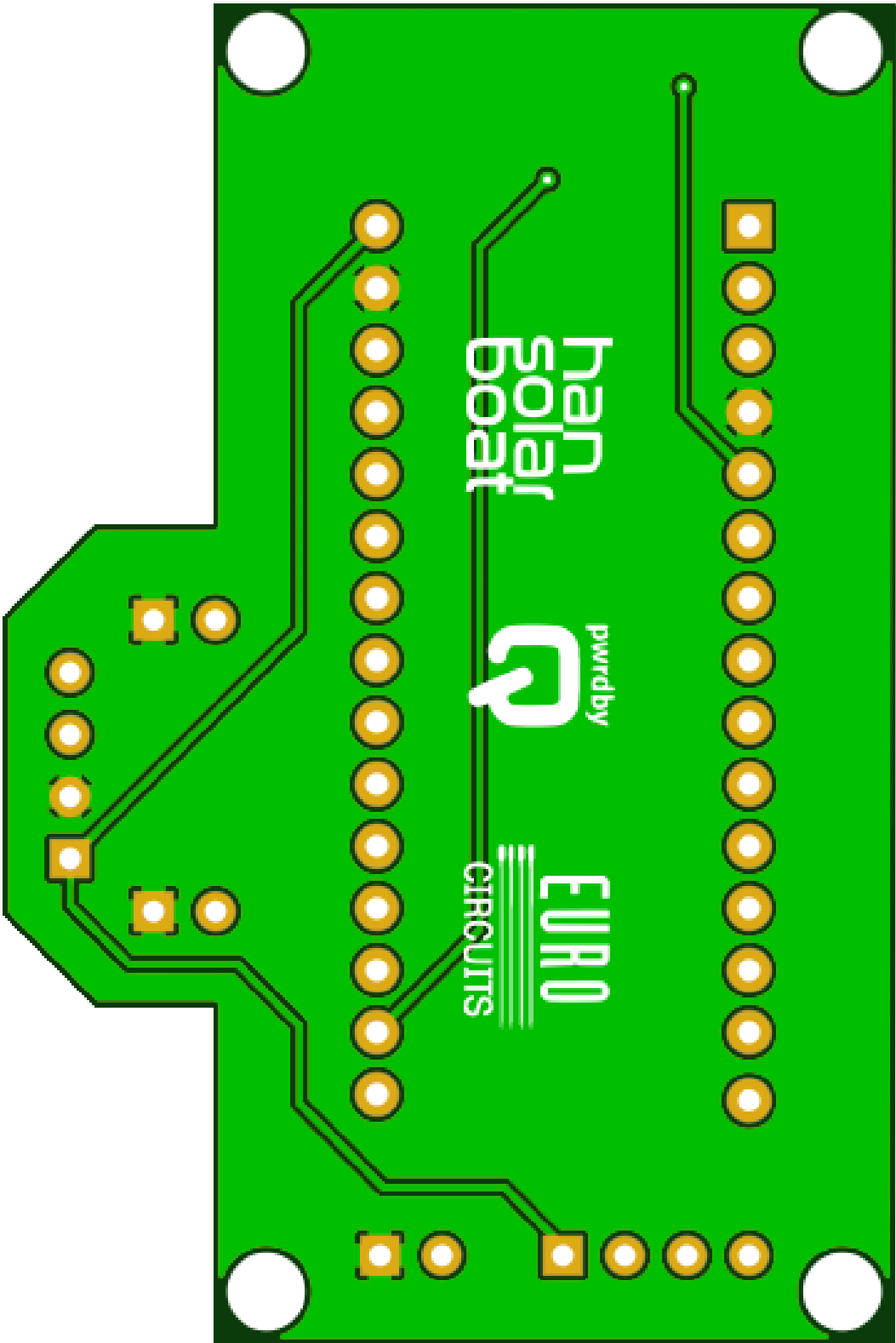
### 9.1 Aansluitschema versie 2.1

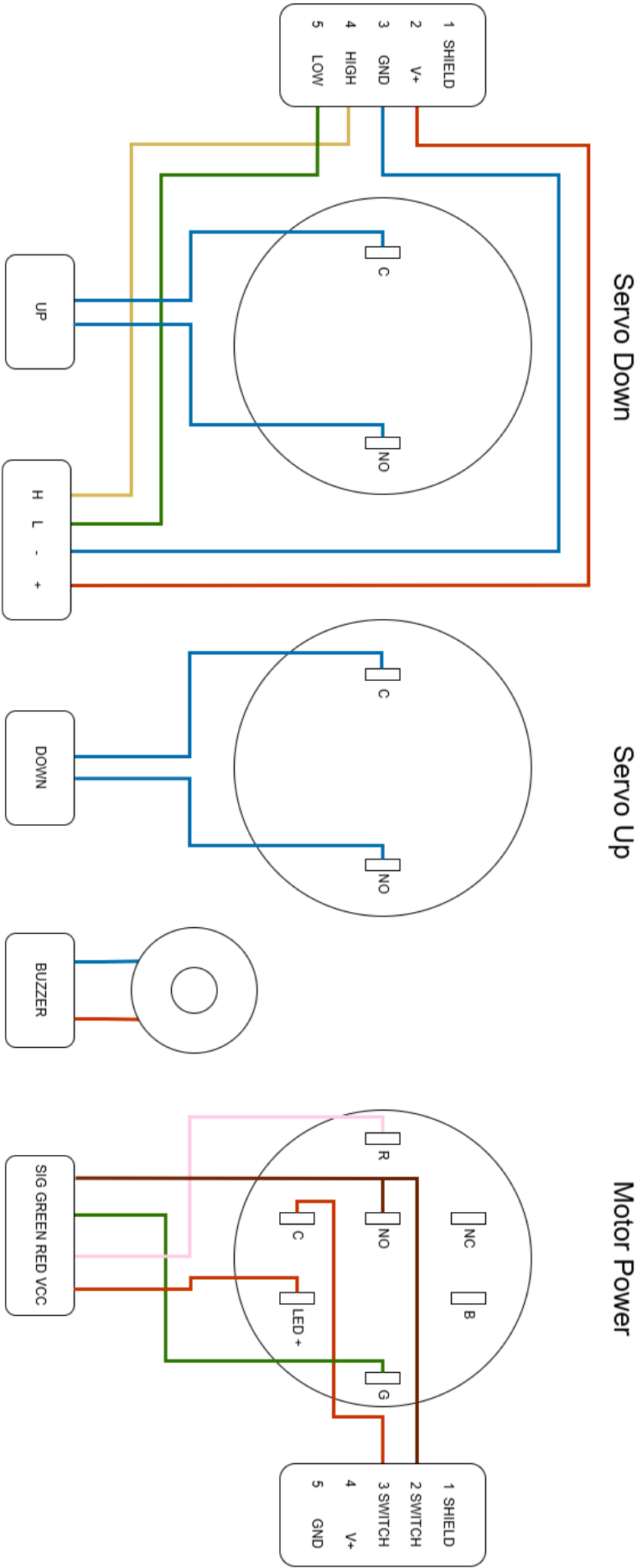




## 9.2 PCB-design







### 9.3 Aansluitschema versie 2.0

