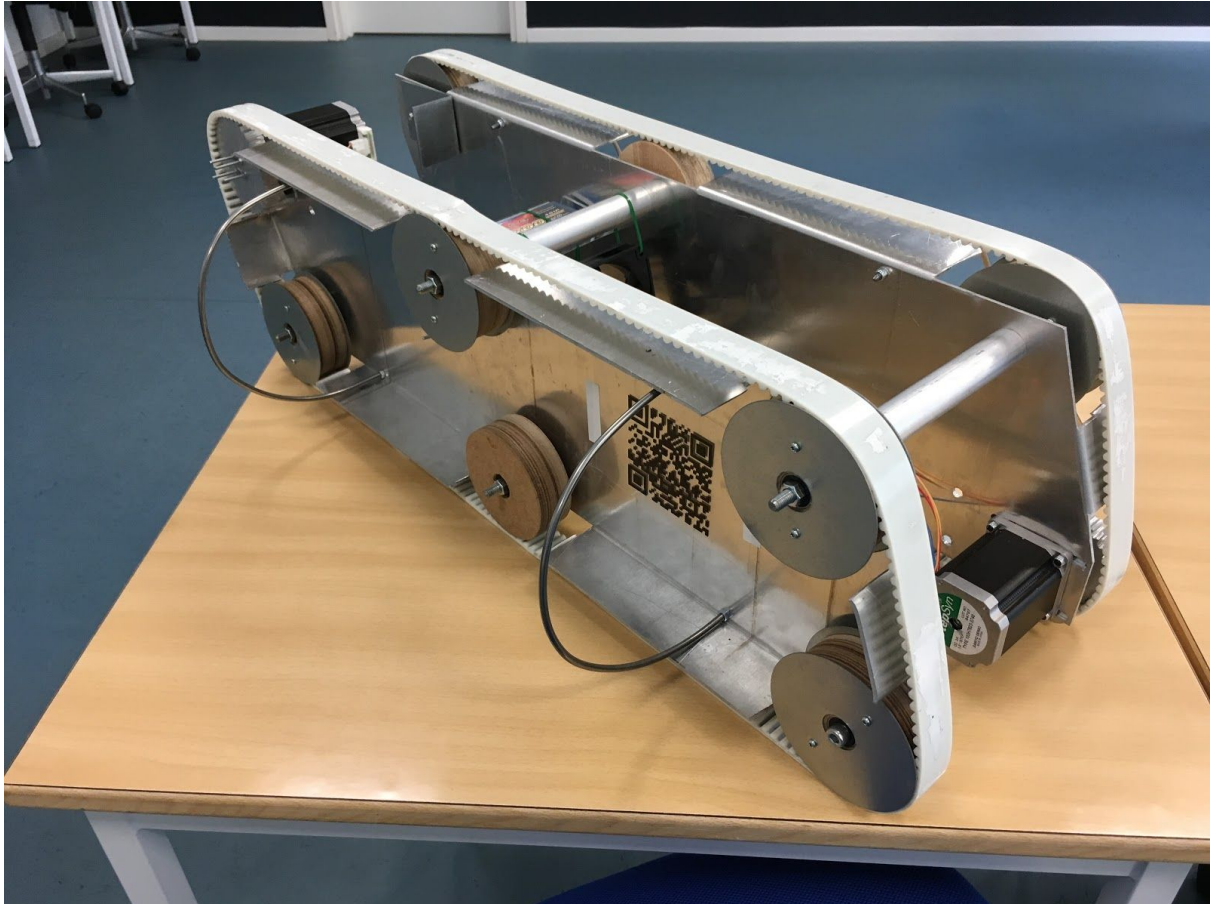


Teknisk dokumentation

E4PRO4 - Bæltedrevet køretøj



Navn - Team 1	AU-ID
Anders Reiche	201604873
Jens Chr. True	20081266
Mikkel Christiansen	201607042
Tobias Valbjørn	201208849

Kunde	Start: Per Lysgaard. Slut: vejledere
Vejledere	Morten Opprud Jacobsen Klaus Kolle

1. Læsevejledning	3
2. Introduktion	5
3. Krav	5
Krav fastlæggelse	5
Endelige Krav	6
4. Overordnet arkitektur	9
5. Mekanik	9
Robot	9
Stykliste	12
Motor	14
Evaluerings af trappekørsel	16
6. Hardware	18
Batteri	18
Platform	19
PSU	19
Stepmotor driver	20
BeagleBone Cape	23
7. Software	28
Etablering af udviklingsmiljø	28
Kommunikation	30
Snitflader	30
webinterface - webserver - ws_rclient	30
ws_rclient - stepdriver	31
ws_rclient - LED interface	31
stepdriver - GPIO output	32
webinterface	32
Hosting	35
ws_rclient	35
Stepdriver	36
Opstartsløsning	39
8. Konklusion	40
Opsummering	40
Produkt evaluering	40
Procesevaluering	43
Kundens afsluttende kommentarer	44
9. Referencer	45
10. Appendix	46

1. Læsevejledning

Denne rapport beskriver den tekniske dokumentation for projekt 3 og 4 for Team 1 på EDE uddannelsen i Herning. Der er taget udgangspunkt i EUDP-metoden, der angiver retningslinjer for udfærdigelsen af en "Technical Report" i uddannelsesøjemed [1]. Målet er, at rapporten skal give en opsummering af udviklingsarbejdet, herunder introduktion, konklusion, arkitektur og beskrivelse af blokkene i det færdige system. Ønskes der en uddybning af de valg, der er foretaget bør man søge mod den udviklingsplan, der ses nedenfor, denne plan viser output af hver timebox.

Box	Uge	Forventet opgave	Output
#1	44 & 45	Proof-of-concept på styring via websockets	Demo af køretøjsstyring. QR-kode
#2	46 & 47	Færdiggørelse af hardware specifikationer. Batteri, motor driver, stepmotor mm.	Demo af stepmotor stepmotorer
#3	48 & 49	Begrundet valg af specifik Beagle Bone Black platform. Færdig DC-regulering til forsyning af Beagle Bone Black platform. PCB cape extension board i form af design og layout.	Valg af BBB platform Færdig DC to regulator PCB udlæg
#4	5 & 6	Test bench opsætning Færdiggørelse af første revision af PCB cape extension board.	Test bench Test og færdiggørelse af PCB cape extension board. Første revision af mekanisk byggeplan. Indkøbsliste mekanik
#5	7 & 8	Mekanisk design Robot software	BBB opretter forbindelse til server. Drive status LED med aktuel tilstand. Undersøgelse af Stepmotor elektromekaniske egenskaber

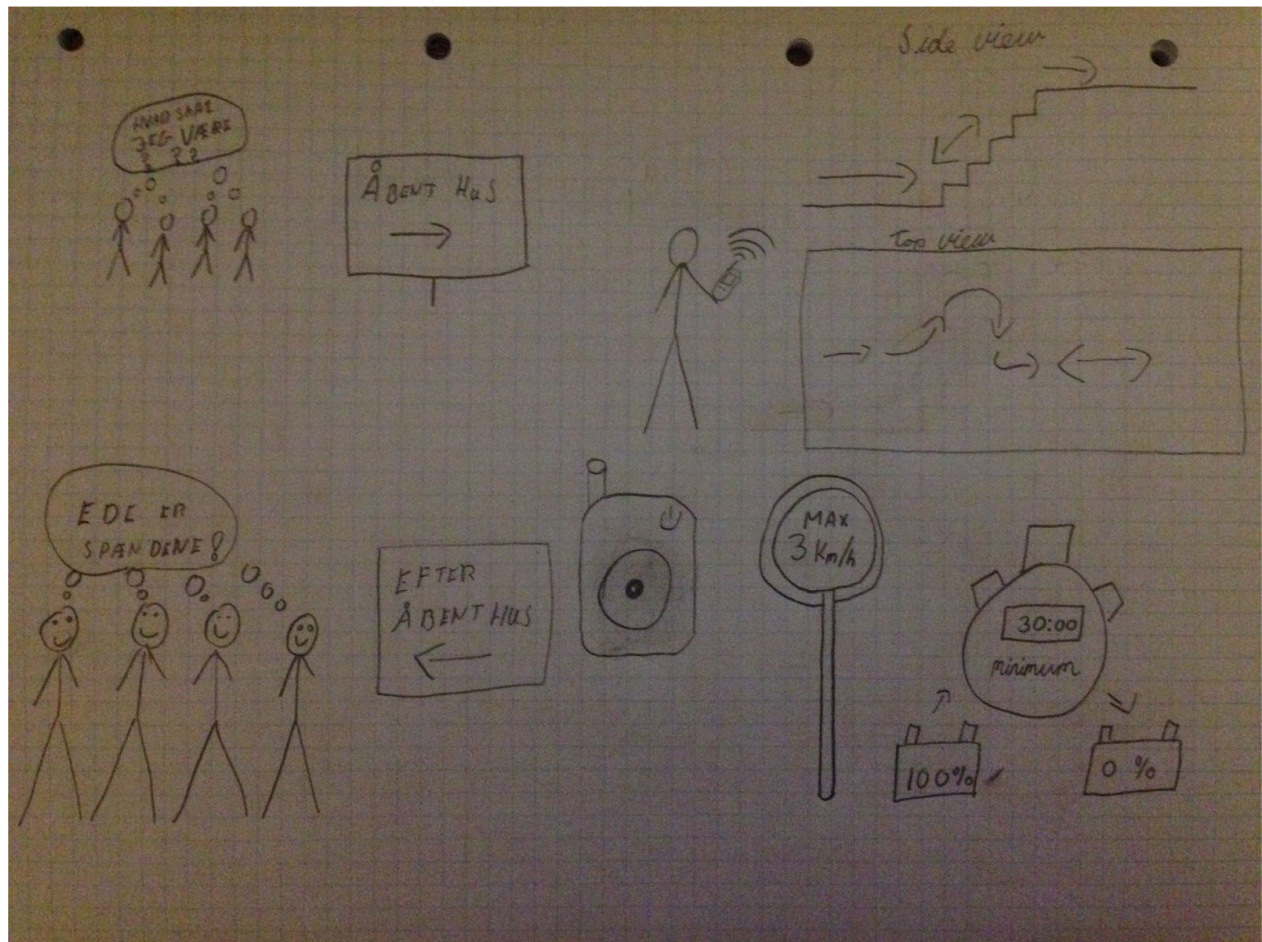
#6	9 & 10	Mekanisk design Robot software	Færdig mekanisk framework til robot Styring af stepmotorer via webinterface med motor driver og controller.
#7	11 & 12	Mekanisk konstruering Server software, samt afbrydning af webinterface Strøm målinger af system Robot software	Færdiggørelse af robot prototype, der kan køre på gulv. Test og dokumentation af nye stepdriver. Bestilling af batteri Køsystem
#8	14 & 15	Samling og montering af bælter Kabling til forsyning Kalibrering, fejlfinding og test af robot Opsætning af system, så robotten bliver klar til styring ved tænd.	Kørende prototype Kabling Forbedring af styring. Opstartsløsning.
#9	16 & 17	Implementering af batteri Videre arbejde med mekanisk design. Verifikation af robot	Fastmontering af batteri. Montering af sidebøjler Test af løsninger til trappe kørsel. Specificering af evne til trappe kørsel.
#10	18 & 19	Teknisk dokumentation, levering, postproject.	Samlet aflevering af produkt + dokumentation Interview med kunde Postproject

Figur 1.1 - De forskellige timeboxes med ugenummer, arbejdsopgaver og det produkt kunden kan forvente.

2. Introduktion

Projektet dækkede 3. og 4. semester på Elektronik i Herning Efterår 2017/ Forår 2018. Oplægget til projektet er baseret på "E3PRO3 Projektforslag -Bæltedrevet køretøj" [2] samt efterfølgende samtale med kunden [3], hvor en prioriteret liste over krav blev udformet.

I løbet af dette projekt, er der blevet arbejdet med at bygge en robot, som kan køre rundt på gulvet, samt komme op og ned af trapper. En bruger skal nemt kunne styre robotten med sin android mobiltelefon. Den er blevet designet ud fra kundens ønske om, at den skal vække begejstring og interesse hos en tilskuerskare. Systemet skal skabe interesse for studier indenfor elektronik og vise hvad en studerende kan efter 2 års studium.



Figur 2.1. Rich picture, udarbejdet med kunden.

3. Krav

Krav fastlæggelse

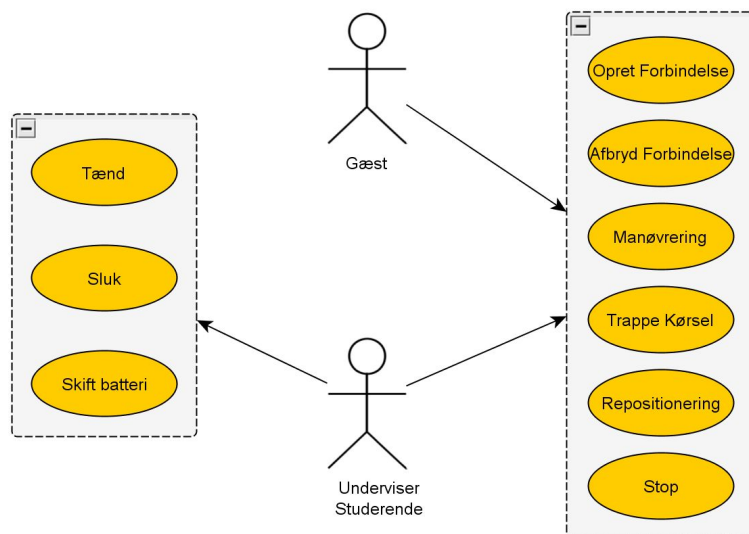
Efter E3PRO3 Projektforslag Bæltedrevet køretøj [2] og et møde med kunden [3], blev der lavet en prioriteret liste, så udviklerne og kunden er enige om hvad der er vigtigt.

Resume af kundemøde:

Citater og uddrag fra kundens ønsker og tanker omkring produktet.

- Essensen af systemet
 - “Robotten skal kunne køre frem, dreje og bakke. Det er essensen. Det kunne være fedt, hvis den kan køre op af trapper. Det er et ”må”, det er ikke et ”skal”, men det kunne være godt at få undersøgt om det overhovedet er muligt.”
- Brugerinterface
 - “Det kunne være sjovt, hvis man kunne bruge en QR code, der kan bruges til at koble op på ens egen mobiltelefon.”
- Robusthed
 - “Robotten skal kunne tåle at falde ned fra et bord.”
- Angående evnen til at vende sig
 - “Hvis den kan køre på hovedet, efter at have inverteret styringen, så vil det være okay, så svarer det til at vende sig. “
- Adfærd ved lavt batteri
 - “Den skal bare gå i stå. Så skal det skiftes.”

Use case diagrammer der beskriver hvem, der benytter systemet, samt hvordan de hver især skal benytte det.



Figur 3.1 - Use case diagram.

Endelige Krav

Fastlagte krav i overensstemmelse med kunde og vejledere i prioriteret rækkefølge.

Skal

1. Brugeren skal kunne styre bælterne på robotten i begge retninger uafhængigt af hinanden.
 - 1.1 Stepmotorerne skal kunne styres fra en beagle bone cape, med en frekvens som input.

- 1.2 Stepmotorerne skal kunne køres i begge retninger uafhængigt af hinanden.
- 1.3 Stepmotorerne kan drive bælte gennem et drivhjul.
2. Robotten skal kunne køre 28 cm/s (1 km/h). Køretøjet må ikke kunne køre hurtigere end 83 cm/s (3 km/h). Der er +/- 1 cm/s præcision.
3. Robotten skal kunne betjenes fra en almindelig moderne Android smartphone minimum Android version 5.0 (Lollipop).
 - 3.1 Webinterfacet skal kunne betjenes fra en almindelig moderne Android smartphone minimum Android version 5.0 (Lollipop).
 - 3.2 Robotten skal modtage kommandoer fra webinterface.
 - 3.3 Joystick kommandoerne skal drive stepmotorerne.
4. Batteriet skal kunne udskiftes på mindre end 5 min.
5. Uanset hvordan robotten placeres på et fladt indendørs underlag, skal den vippe om på sine larvefødder, og være i stand til at fortsætte kørslen.
6. Brugeren skal kunne afbryde betjeningen af *robotten*.
 - 6.1 Brugeren skal kunne afbryde betjeningen af *webinterfacet*.
 - 6.2 Brugeren skal blive disconnected efter 30 sekunders inaktivitet
7. Robotten skal stoppe når betjeningen ikke berøres.
 - 7.1 Webinterfacet skal sende en "END" kommando når joysticket ikke længere berøres.
 - 7.2 Akslen på motoren skal stå stille, når joysticket ikke berøres
8. Robotten skal være kortere end 85 cm.
9. Robotten skal have en driftstid på minimum 30 minutters uafbrudt drift.

Forventet

10. Robotten skal kunne overleve at få blokeret sine hjul med motorerne i fremdrift i 10 sekunder.
 - 10.1 Stepmotorerne skal kunne blokeres i 10 sekunder uden, at hverken cape eller motorer ødelægges.
 - 10.2 Mekanikken skal kunne holde til, at hjulene blokeres i 10 sekunder.

Bør

11. Bruger må kunne scanne en QR kode på robotten for at betjene dette.
 - 11.1 Brugeren må kunne scanne en QR kode for at blive linket til webinterfacet, for at betjene dette.
12. Brugeren skal kunne se på status LED på robotten, når den er klar til at oprette forbindelse og når den er oprettet.
13. Robotten må gerne kravle op af en trappe med stødplader.
 - 13.1. Robotten skal kunne strække sig over to trin.
 - 13.2. Motorerne skal dimensioneres til at kunne løfte en vægt på 8 kg op af trappen.
 - 13.3. Ved kørsel op af trappen skal bælteerne ikke glide.
 - 13.4. Robotten skal komme op på det første trin.
 - 13.5. Robotten skal kunne komme hele vejen op af trappen.

Må

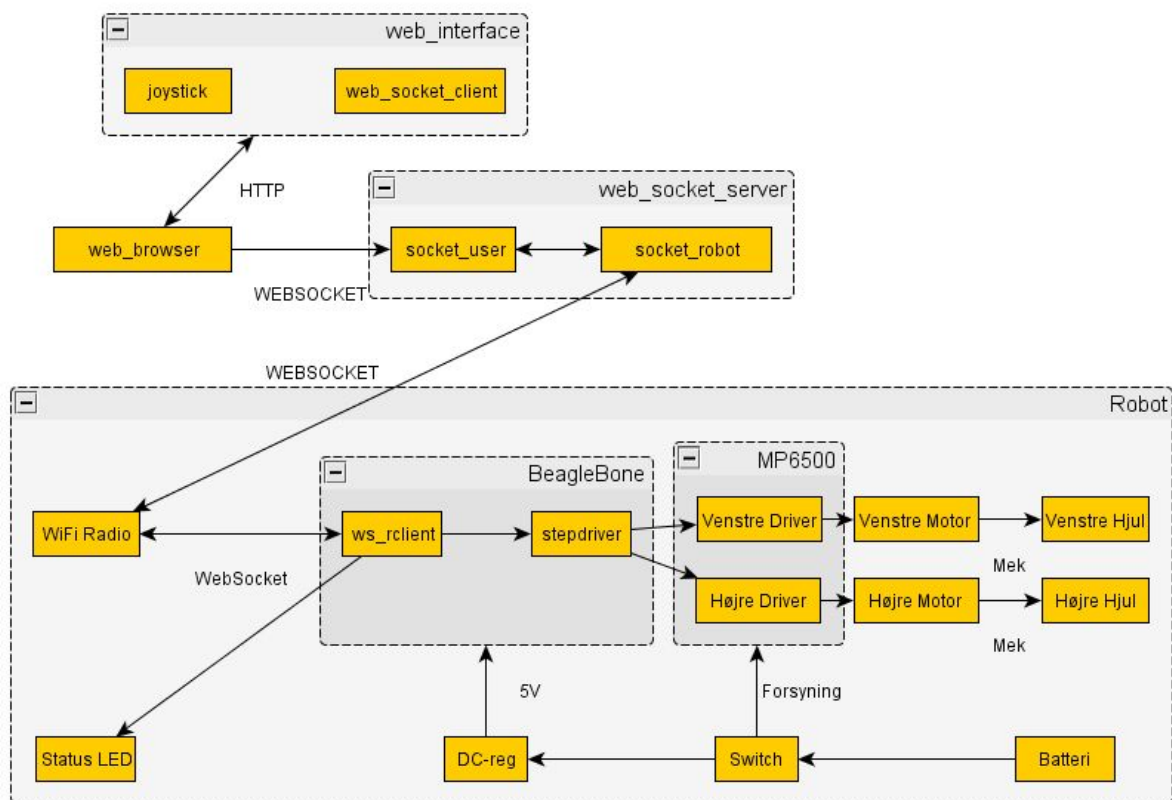
14. Robotten bør overleve et frit fald på 90 cm ned på et fast linoleumsunderlag.
15. Der efterlades minimum 50% CPU ressourcer.

16. Der må max benyttes 80% af den samlede strøm kapacitet (max current, IKKE maH)
17. Stepmotorer skal have 20% ekstra kraft, end hvad der kræves for at bestige trappen

Specifikationer for testudførsel er tilgængelig i dokumentet "Requirements and Verification"[29]

4. Overordnet arkitektur

Projektet består af 3 overordnede dele. Et webinterface del, som er den brugergrænseflade brugeren præsenteres for. Webserveren er bindeled mellem brugeren og robotten. Til sidst findes robotten som består af både hardware, software og mekanik. Denne dokumentation vil først beskrive den mekaniske opbygning af robotten, da dette danner grundlag for mange af de senere designovervejelser.



Figur 4.1 Arkitektur over system

5. Mekanik

Robot

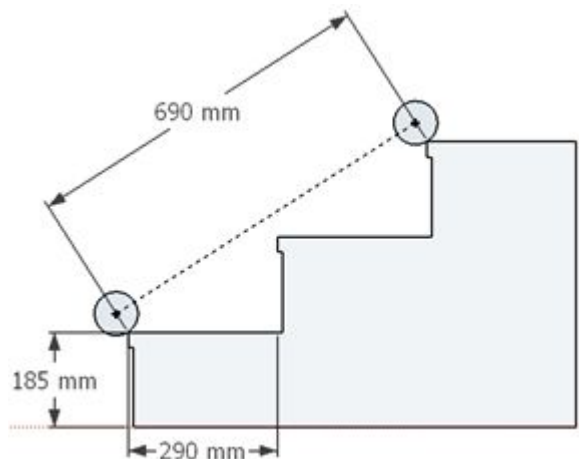
Funktionaliteten "at køre på trapper" er opfyldes at designe robotten som en "Mark V-tank type" [4]. Robotten skal være høj nok til at kunne "bestige" første trin, og når den kører på trapperne, skal den altid skal have kontakt med mindst to trappekanter.

Sammen med kravet om maksimal længde fastlægger dette robottens størrelse.

Dette er skitseret i figurerne herunder.

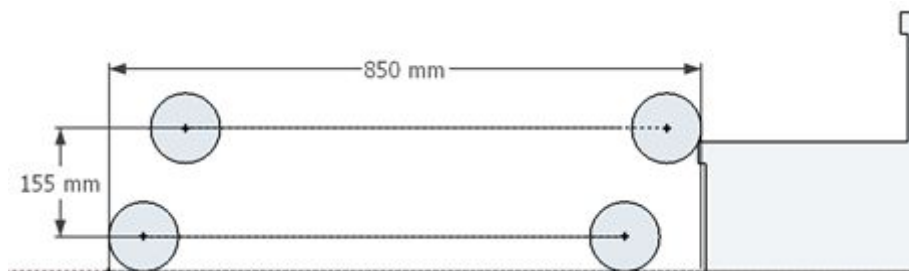
Motoren har et drivhjul på $\varnothing 100$ mm så det "kommer ud" over motorernes baseplade, da robotten skal køre "direkte" på drivhjulet.

Figur 5.1 viser en skitse over trappen med de af kunden, angivne dimensioner, og den mindste afstand mellem et sæt aksler, der gør det muligt for robotten hele tiden at have kontakt med mindst 2 trappekanter, og dermed stabilt at kunne køre på trappestigningen.

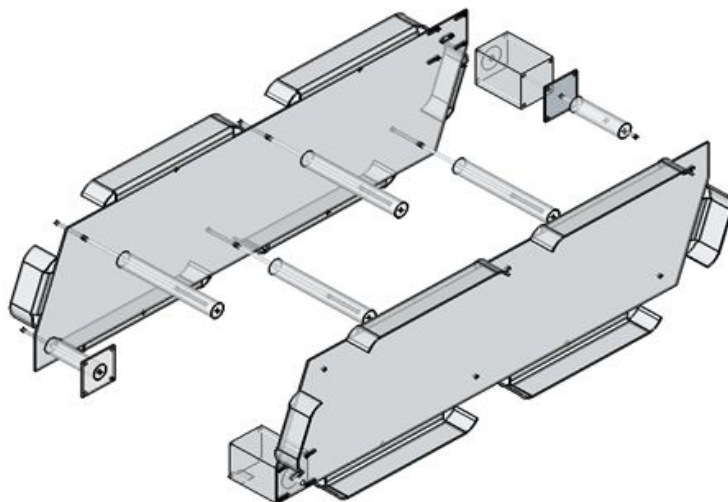


Figur 5.1 - Trappedimensioner og bestemmelse af afstand mellem for og bagaksel.

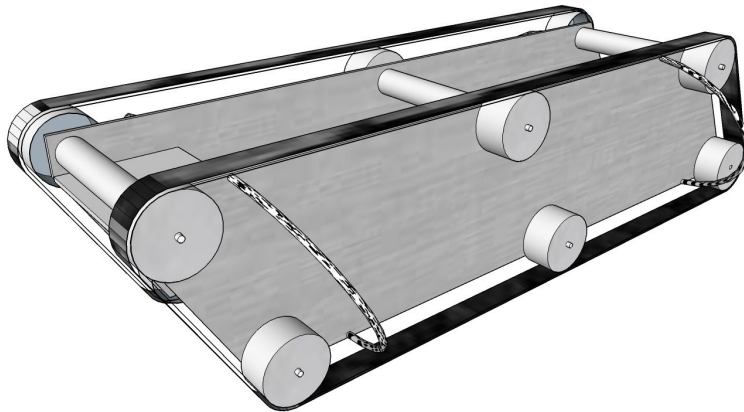
Figur 5.2 viser den højde robotten skal have, så bælterne (som senere monteres på drivhjul/føringshjul), kan få et godt "greb" om første trappetrin, så den kan forceres. Samtidig vises den størst mulige vinkling af robottens for-/bagende, når længden er begrænset til 85 cm. Det antages, at jo større vinkling, jo nemmere vil robotten have ved at forcere første trappetrin.



Figur 5.2 - Bestemmelse af robottens højde.

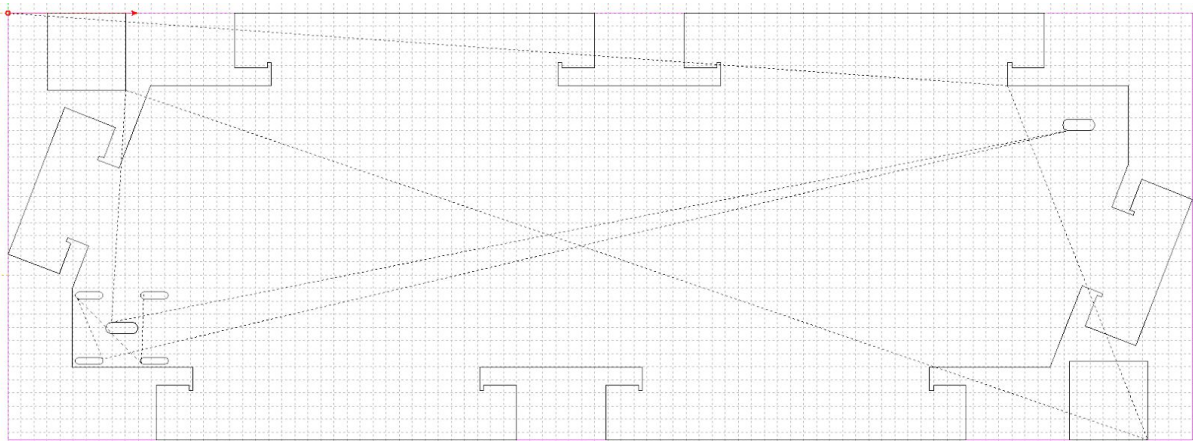


Figur 5.3 - "Exploded view" af robots chassis.

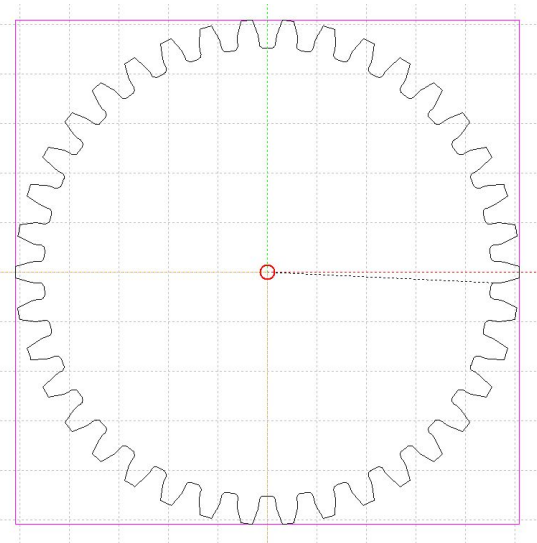


Figur 5.4 - 3D skitse af design forslag

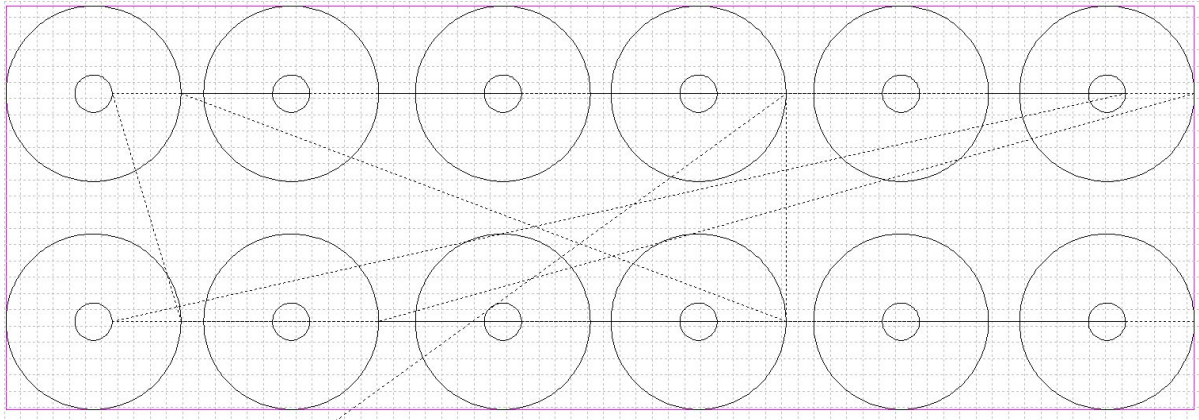
For yderligere information om designprocessen, se bilag[5]. Ud fra det overordnede design, er der udviklet filer til vandskærer, så grunddelene kan skæres:



Figur 5.5 - Tegning til udskæring af sideplader (2stk)



Figur 5.6 - Tegning til udskæring af tandhjul (2stk) [6]



Figur 5.7 - Tegning til udskæring af sidestykker

Siderne på robotten er lavet i 3 mm aluminiums plader, ydermere for at robotten mere stabil, er der lavet en række 90 graders buk, som afstiver robotten og samtidig giver, bælterne en flade at glide på. Der er 2 tandhjul, et til hver motor, som sikrer en god kraftoverførsel fra motorens aksel, og til bælterne monteret modsat hinanden. Der er fremstillet 10 drivhjul i træ, med kuglelejer, som minimerer friktionen fra bælternes bevægelse. Der er monteret sideplader på drivhjulene, for at sikre at bælterne ikke kører skævt og falder af. For at holde de 2 sider sammen, er der brugt 8 mm gevindstænger med 25 mm aluminiumsrør udenover. Disse rør styrker konstruktionen. Sidebøjler sikrer, at robotten altid lander på et af de to underlag, hvor den kan fortsætte kørslen.

Til elektronik er der placeret en bakke inde midt i robotten, hvor styringsenheden og batteriet er monteret. Motorene er spændt direkte på de 3 mm aluminiumsplader, der er laver slidser både ved motoren og drivhjulet overfor, så disse kan flyttes når bæltet spændes op.

Der en fysisk ON/OFF knap på selve køretøjet. Knappen afbryder forbindelsen til batteriet. Robotten er 85 cm lang og vejer 12.38 kg. For mere information om det mekaniske arbejde, se bilag[6][7][8].

Stykliste

Komplet stykliste for en robot.

Beskrivelse	Dimension	Bemærkninger
2 stk. 3 mm aluminiumsplade til sidevanger	minimum 2 stk: 326 x 915 mm	Til skæring i vandskærer.
4 stk afstandsrørstykker mellem sidevanger	Ø25 mm alu-rør, 1,5 mm godstykkelse Længde: 176 mm	Ø25 diameter er valgt for at give god stivhed i konstruktionen.
4 stk aksler	Ø8 mm gevindstang Længde: 280 mm	Udgør også aksel for kørehjul
2 stk bolte	Ø8 mm 70mm bolt	Aksler for kørehjul modsat stepmotor

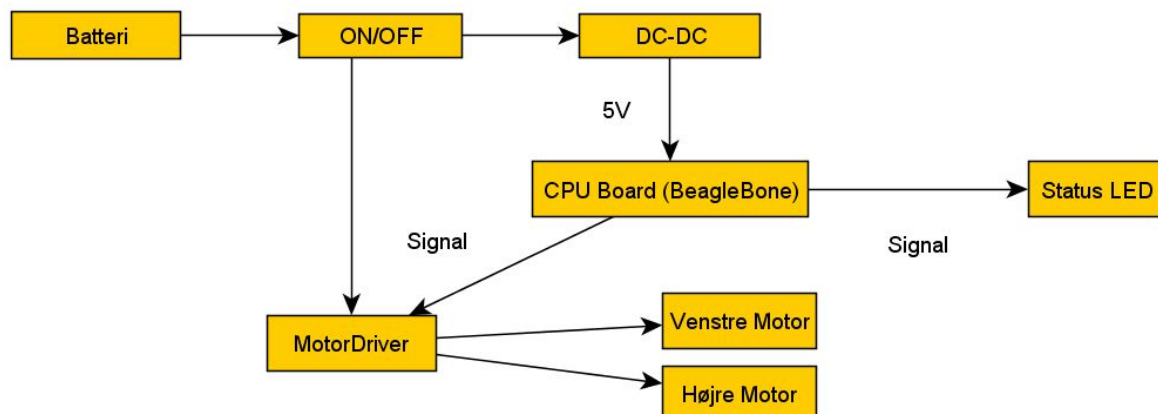
2 stk. plade til opstramning af bælte	120 * 60 * i 2 mm plade med et midt over 120.	Til opspænding af løs aksel
1 stk. Sammenspænding af sidevanger	10 stk alm M8 10 stk store skiver M6 med ydre diameter ø25. 20 stk M6 låsemøtrikker	De store skiver bruges når sidevangerne spændes fast mod afstandsstykkerne. De små skiver bruges til montering af føringshjulene.
2 stk. Aksel adapter	1 stk adapter 8 styk 4M låsemøtrikker 4 styk 4M gevindstang 60mm	Adapter fra stepmotorens D aksel der muliggør montering af tandhjul.
1 stk. Platform til batteri og elektronik	2 mm zink-plade. 216 mm x 140 mm.	Monteres mellem sidevangerne.
1 stk. Montering af platform	4 styk M4 bolte 4 styk M4 låsemøtrik 4 styk M3 stag 4 styk M3 skruer	Diverse bolte og møtrikker til montering af platform med beaglebone montering
4 stk Styrerbøjler	Ø6 mm rundjern 2 stk. møtrik 2 stk. låsemøtrik.	Der er skåret gevind i enderne, og rundjernet er bukket i passende bue, og monteres fra top til bund i hver ende, af de to sidevanger.
2 stk. Bælter	Bredde 30 mm. 2meter. 8M deling	Som bælte benyttes to tandremme, der samles via syning med ståltråd.
2 stk tandremsskiver	Ca. ø100 diameter til 8M bælttestørrelse	Skal passe til 8 mm akslerne på stepmotorerne samt aksel adapter. Skåret selv.
10 stk kørehjul	Aksel-diameter ø8 mm. 2 stk. sideplader. 4 skruer til montering M3 2 stk. kugleleje. 1 stk afstandsrør. 1 stk træ hjul.	Samme diameter som tandremsskiven. Der skal være lejer indbygget i føringshjulene. sideplader monteret for at holde remmen på plads.
1 stk. Blæser	12V blæser. 4 styk plastikstrips	Blæser til køling af elektronik monteret via strips på afstandsrør.
2 stk stepmotor	Sanyo Denki 103H7823-5740 4 stk. 30 mm M5 bolte. 4 stk. M5 skiver. 4 stk. M5 låsemøtrikker	http://docs-europe.electrocomponents.com/webdocs/141d/0900766b8141d510.pdf
2 stk stepmotor kabel	Sanyo kabel til bipolar stepmotor	
1 stk. Batteri	Gens ace 2700mAh 14.8V	LiPo batteri. Oplading ikke del af projekt

	35C Lipo	
1 stk. Batteri montering	5 styk plastikstrips han-hun bananstik ON-OFF afbryder 10x20 cm skummåtte	Batteri indpakkes i skum og monteres via plastikstrips. Han-Hun bananstik monteres på kabel ender og forsynes med afbryder.
1 stk. BeagleBone	BeagleBone Black Wireless 1 stk MicroSD kort min 4gb	http://dk.farnell.com/element14/bbone-black-wireless/beaglebone-black-wireless/dp/2671597
1 stk. Stepdriver cape	Beaglebone cape til stepdriver	Custom cape til beagle
2 stk. stepdriver	MP6500	Monteres i sokler på beagle bone cape

Motor

Stepmotorer har nogle egenskaber, der er ønskværdige ift. system-to-be. De har en god hastighedskontrol, som vil lette processen, med at overholde kravene til, hvor hurtigt robotten må køre, samt eventuelt, at indstille et hastighedstrin. Derudover kan en stepmotor, i modsætning til DC-motorer, udvikle et højt torque ved lave hastigheder. Dette er relevant under bestigningen af trapper. Stepmotorerne giver muligheder for kraftigt holdemoment.

Stepmotorer er ikke særlig energieffektive. De bruger samme mængde energi konstant, uanset belastning.



Figur 5.8 - Hardware Architecture Diagram tegnet i yEd

Der blev konfereret med Jan Møller Nielsen omkring den anslåede vægt af designet. Denne viden blev brugt til at dimensionere motorstørrelsen. Der regnedes med en vægt på 5 kg. Estimatet inkluderede siderne på robotten, tværgående afstivning, bøjler på siden, drivhjul og vejhjul. Der blev tillagt 3.1 kg til robotchassiset, som et estimat, for at dække motor, batteri, elektronik og bæltet. Køretøjets samlede vægt blev estimeret til 8,1 kg. Der er valgt den mindst mulige radius på de hjul, der skal sidde på stepmotorernes aksel, da den kraft

der kan leveres falder ved en større radius. (Vægten endte i sidste ende op i 12.38 kg med en radius på 4,5 cm.)

Motorerne blev dimensioneret efter den estimerede vægt, samt kravene til at komme op af trappen.

Figur 5.9 viser den oprindelige udregning med henblik på dimensioneringen af stepmotoren. Trappens dimensioner er angivet til beregning af stigningsvinkel. En typisk bipolar stepmotor, har en effektivitet mellem 50% og 75%, derfor er torque outputet beregnet ved 3 forskellige motoreffektivitetsgrader.

I det værste tilfælde regnedes det at der skulle bruges to stepmotorer med en torque på 1.7Nm.

Udregning af trappens vinkel:

$$h_{trappe} := 0.185 \text{ m} \quad b_{trappe} := 0.29 \text{ m}$$

Phytagaros til udregning af den sidste side:

$$c := \sqrt{(h_{trappe})^2 + (b_{trappe})^2} = 0.344 \text{ m}$$

Cosinusrelationen bruges til at finde vinklen som robotten skal bestige:

$$\theta := \arcsin\left(\frac{h_{trappe}}{c}\right) = 32.535 \text{ deg}$$

$$g := 9.816 \frac{\text{m}}{\text{s}^2} \quad m_{køretøj} := 8100 \text{ gm}$$

$$force := m_{køretøj} \cdot g \cdot \sin(\theta) = 42.762 \text{ N}$$

$$wheel_r := 0.04 \text{ m} = 4 \text{ cm}$$

Torque required

$$torque := \frac{force \cdot wheel_r}{2} = 0.855 \text{ J}$$

Motor efficiency

$$M_{e1} := 50\% \quad M_{e2} := 65\% \quad M_{e3} := 75\%$$

$$\frac{torque}{M_{e1}} = 1.71 \text{ J} \quad \frac{torque}{M_{e2}} = 1.316 \text{ J} \quad \frac{torque}{M_{e3}} = 1.14 \text{ J}$$

Compactly stated, torque = wheel radius force and force = $m \cdot g \cdot \sin(\theta)$, the component of the gravitational force on the vehicle, parallel to the slope.

Figur 5.9 Bestemmelse af maksimalt torque-behov pr. hjul.

Der er valgt en bipolar motor. Den giver et mere torque, men øget kompleksitet i forhold til motor driveren. Der er valgt en stepopløsning på 1.8 grader. Det giver en tilpas opløsning til at kunne justere hastigheden præcist, samtidig med, at der kan leveres det nødvendige høje torque output.

Ud fra de givne constraints er der valgt 2 styks Sanyo Denki 103H7823-5740

Maksimal torque fås ved pulsfrekvens på 100 Hz. Her kan stepmotoren yde op til 2,7 Nm. Dette er der behov for, når robotten køres op af trapper, eller over andre lignende forhindringer.

Evaluering af trappekørsel

Forsøg til at fastlægge hvilken stigningsgrad robotten er i stand til at passere, og hvor tæt projektet således er på at opfylde kravet til trappekørsel. Kravet om trappekørsel er ikke opfyldt.

Stignings setup

Robotten blev placeret på en plade, som løftes til forskellige vinkler, og det blev undersøgt hvilken stigningsgrad robotten er i stand til at køre op ad. Der benyttes en 2 meter lang plade, og robotten blev placeret midt på stykket. Robotten måtte i stilstand ikke glide, og når der langsomt blev navigeret frem, skulle robotten køre op af rampen. Stigningsgraden på rampen ændredes herefter ved at løfte pladen i den ene ende mens den anden holdes mod gulvet.

Højde	Vinkel	Resultat	Bemærkning
47 cm	14°	Ja	
50 cm	15°	Ja	
55 cm	16°	Ja	Ja. Med let reduceret hastighed
59 cm	17°	Ja	Ved meget forsigtig navigering kan robotten bevæge sig op af rampen
88 cm	26°	Nej	Robotten kan holde stille men glider baglæns hvis motorene aktiveres
108 cm	33°	Nej	Robotten kan ikke holde stille

Trappen der ønskes kørsel på har en stigning på 33 grader. Et løseligt forsøg blev udført, hvor robottens vægt blev reduceret via lodret løft med en bagagevægt. Dette forsøg burde vise om robotten ville være i stand til kørsel på den vinkel, med en lavere totalvægt. Forsøget blev udført flere gange, for at give et bedre resultat. Der vurderes ud fra disse

målinger at en vægt reducere på 4½ kg burde gøre robotten i stand til at køre op af en skråning på 33 grader.

Ydermere blev robotten placeret på gulvet, og en måling af robottens trækraft. Ved kørsel på fladt underlag, blev aflæst med en bagagevægt. Robotten udførte en trækraft på 2.92 kg. Ud fra denne trækraft er den teoretiske maksimale vægt, og den teoretiske maksimale vinkel for den givne robot vægt beregnet.

Robot trækraft

$$m_{pull} := 2.92 \text{ kg}$$

Robot masse

$$m_{robot} := 12.3 \text{ kg}$$

Trækraft omregnet til moment

$$F_{motor} := m_{pull} \cdot g = 28.635 \text{ N}$$

Beregn den maksimale vægt der kan løftes 33grader

$$m_{max} := \frac{F_{motor}}{g \cdot \sin(33^\circ)} = 5.361 \text{ kg}$$

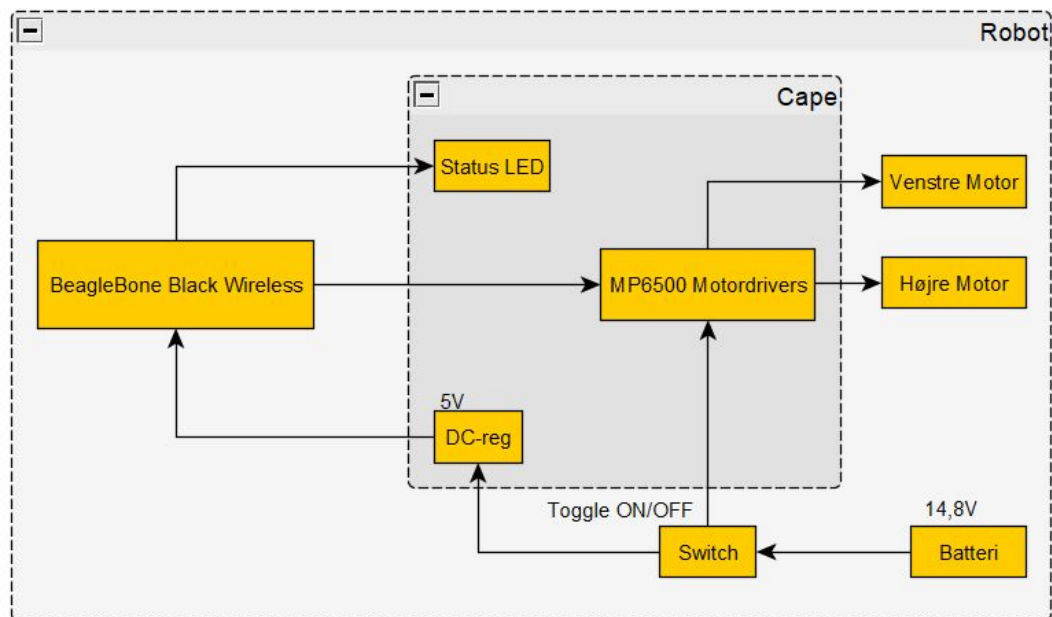
Beregn den maksimale vinkel robottens masse kan løftes

$$\phi_{max} := \arcsin\left(\frac{F_{motor}}{g \cdot m_{robot}}\right) = 13.733 \text{ deg}$$

Figur 5.10 - Robot udregninger

Motoren er dimensioneret efter at have et drivhjul på 4 cm i radius. Efterfølgende i designfasen endte vi med, at have et drivhjul på ca. 4.5 cm. Robotten er endt med at veje ca. 4 kg mere end først estimeret. Derudover var der mere energi, der blev afsat internt i robotten end vi havde regnet med, i selve konstruktionen. Begge dele øger kravet til torque yderligere. Hvis vi ser bort fra friktion, og vi bruger den samme torqueudregning fra motordimensioneringen, så vil behovet ved 70% effektivitet være 2.1 Nm. Det er altså under hvad vores motor er rated til ved 100 Hz, der var relevant i forhold til trappekørsel. Det vil sige, at tab af effekt i den mekaniske konstruktion, udgør en stor del af problemet.

6. Hardware



Figur 6.1 blokdiagram over hardware.

Hardwaren består af en beaglebone black wireless inklusiv et specialfremstillet cape med PSU og motor driver, og batteriforbindelser.

Batteri

Der er valgt et batteri med følgende Specifikationer :

Bredde	42 mm
Længde	138 mm
Tykkelse	23 mm
Vægt	311 gram
Spænding	14.8V
Kapacitet	2700 mAH
aflade/lade hastighed	35C

Efter udregningerne, kan hele systemet køre i 1 time og 30 min, på et fuldt opladt batteri, hvilket svarer til 300% af kravspecifikationerne.



Figur 6.2 - Montering af batteri

Der er konstrueret en plexiglasholder til montering, og til beskyttelse af LiPo batteriet. Dette skal holde batteriet, når det er monteret i robotten. Batteriet er monteret med standard bananstik, som gør det let at oplade, eller midlertidig forsyne robotten under test. De to strips kan trækkes til siden for hurtig udskiftning af batteriet. For mere information om batterivalg se bilag[9].

Platform

Hardware platformen er baseret på et embedded Linux board. BeagleBone Black Wireless. MicroSD kortet indeholder en debian linux distribution opdateret til version 9.3 og det brugte image er "bone-debian-9.3-iot-armhf-2018-01-28-4gb.img"

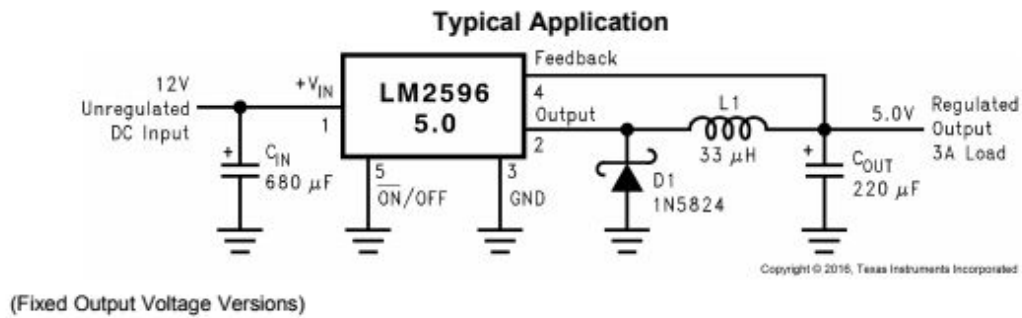
PSU

DC-regulator

Der er følgende "design constraints" til DC-regulatoren:

- Med input fra 16.8 V og ned til 12.8 V (henholdsvis maksimalt og minimalt opladning på et 4 cell LiPo batteri med en nominal spænding på 14.8V) skal den kunne regulere ned til en spænding på 5.0V +/- 250 mV, og "rated" til 2A (BBB specifikationer)[13].

Der er benyttet den typiske applikation fra databladet (figur 6.3).



Figur 6.3 - Schematic for en typisk applikation med LM2596. Eksemplet er med 5 volt. [10]

Komponent værdier er ændret ifht. 2A applikation af kredsløbet.

L1: 68mH

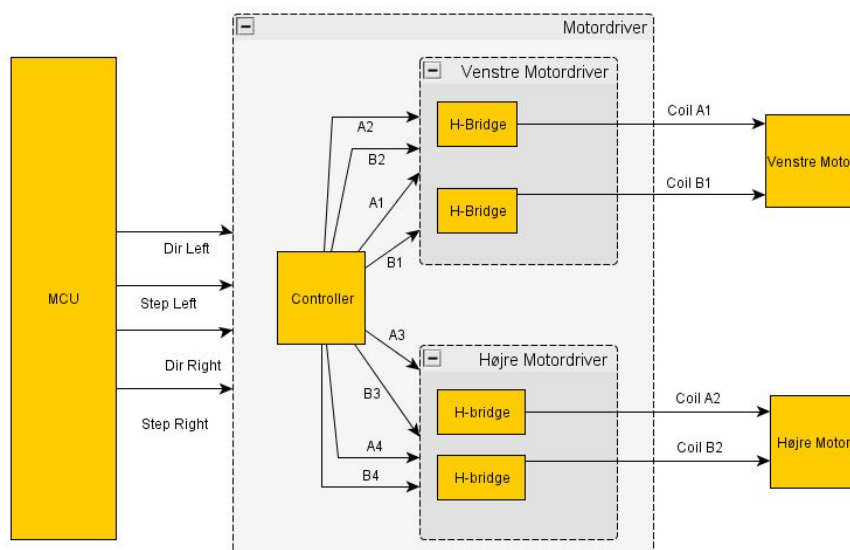
Cout: 180µF

For mere information om PSU, se bilag[11].

Steppmotor driver

Kravene til motor driveren er følgende:

Vcc:	$\geq 5V$
Control input voltage:	0-3.3V
Control input current:	0-4mA (Maks output fra BeagleBone GPIO pins)
Output voltage:	5V
Output current:	2A

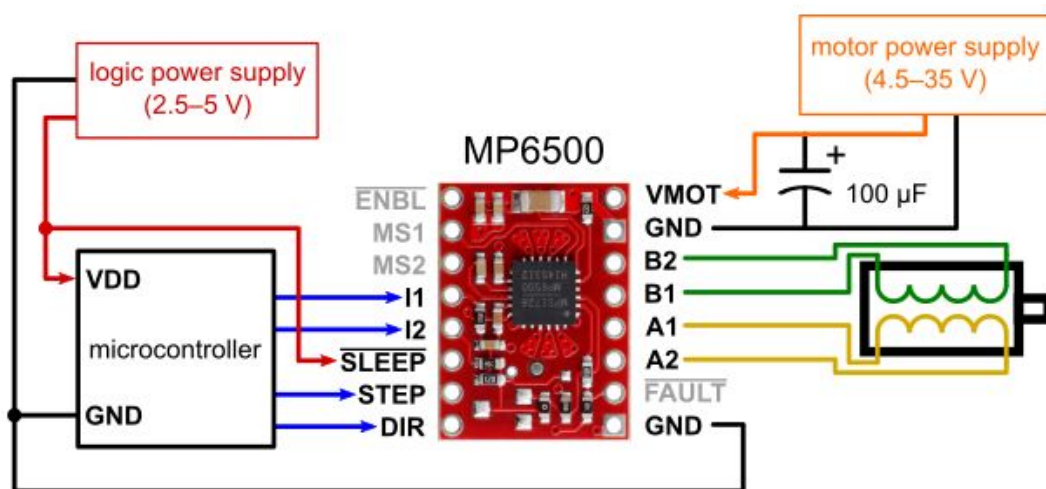


Figur 6.4 - Motor Driver interface diagram.

Overordnet interface

Signal	Voltage	Max Current
Dir Left	3.3V	4mA
Dir Right	3.3V	4mA
Step Left	3.3V	4mA
Step Right	3.3V	4mA
A1,A2,A3,A4,B1,B2,B3,B4	3.3-5V	Afhænger af H-bro hFE
Coil A1, Coil A2, Coil B1, Coil B2	Batteri spænding	2A

Figur 6.5 - Interface for motor-driver.



Basic wiring diagram for connecting a microcontroller to an MP6500 Stepper Motor Driver Carrier, Digital Current Control (full-step mode).

Figur 6.6 Basalt diagram for MP6500 med pinout. Microcontroller styrer driver i full step mode.

PIN-beskrivelse for MP6500:

PIN	Beskrivelse	Tilstand	Input	Input fra
ENABLE	Enable pin for kredsløb. Hvis pin er high deaktiveres MOSFET på udgangene.	Lav for aktiv driver	Lav < $V_{DD} \times 0.3$ Høj > $V_{DD} \times 0.7$	BBB-GPIO
MS1, MS2	Microstepping selector pins.	Sættes alle lav for full stepping.	Lav < $V_{DD} \times 0.3$ Høj > $V_{DD} \times 0.7$	BBB-GPIO
I1	Current limiting control 1. Lav for 2Amp	Lav	Lav < $V_{DD} \times 0.3$ Høj > $V_{DD} \times 0.7$	BBB-GPIO

I2	Current limiting control 1 Lav for 2Amp	Lav	Lav < $V_{DD} \times 0.3$ Høj > $V_{DD} \times 0.7$	BBB-GPIO
SLEEP	Sleep mode pin. Sætter kredsen i en strømbesparende mode.	Sæt høj for at holde ude af sleep.	Lav < $V_{DD} \times 0.3$ Høj > $V_{DD} \times 0.7$	BBB-GPIO
STEP	Advancer stepmotoren med et step.	Trigger på opgående flanke	Lav < $V_{DD} \times 0.3$ Høj > $V_{DD} \times 0.7$	BBB-PWM
DIR	Retningsvælger pin	Lav/Høj vælger retning	Lav < $V_{DD} \times 0.3$ Høj > $V_{DD} \times 0.7$	BBB-GPIO
VMOT	Forsyning til stepmotor.	Forsyning	8-35V	Direkte fra batteri
Fault	Open drain output. Pin forbindes ikke.	NC	NC	NC
GND	Ground		0 V	
B2, A2, A1, B1	Stepmotor output		VMOT	

Figur 6.7 - Pin beskrivelse af MP6500 motor driver

Forbindelse mellem Beaglebone og MP6500 stepdriver:

Pin nr. på BBB-P8-header	Pin nr. på Motor driver 1	Pin nr. på BBB-P9-header	Pin nr. på Motor driver 2
17	ENABLE	15	ENABLE
7	MS1	23	MS1
9	MS2	25	MS2
18	I1	27	i1
14	I2	30	I2
16	SLEEP	41	SLEEP
13 (dedikeret PWM-port)	STEP	14 (dedikeret PWM-port)	STEP
10	DIR	12	DIR

Figur 6.8 - Pin-planning BBB<-->2 * MP6500 motor driver.

For mere information om stepdriver se bilag[12].

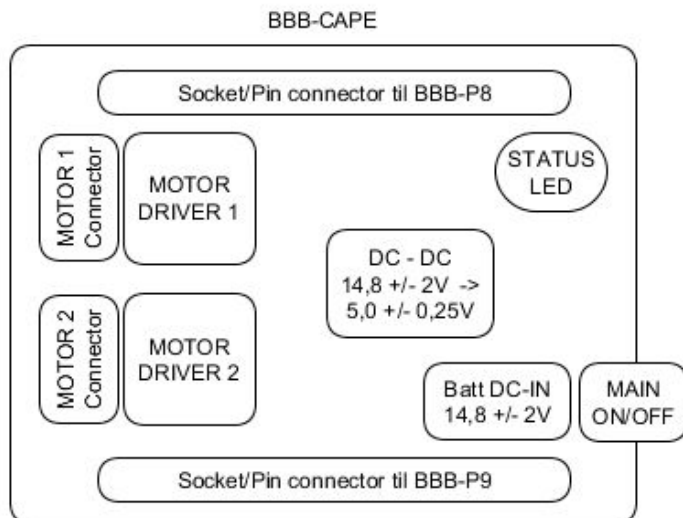
Under implementeringen fandt vi ud af, at vores drivers blev meget varme. Vi har implementeret aktiv køling med en blæser, der forsynes direkte fra batterispænding. Dette har løst varmeproblemet. Der er ikke foretaget yderligere overvejelser omkring dette område.

BeagleBone Cape

For at minimere mængden af kabler på robotten, samt gøre denne mere robust, er der udviklet et "beaglebone cape" som integrerer en stor del af de eksterne komponenter.

Følgende funktionalitet er implementeret på boardet.

- BeagleBone strømforsyning
 - Stepdown converter, som skal levere 5V til BeagleBone fra batteriet.
- Stepmotor driver
 - Sokler til 2 stk monterede MP6500 moduler.
- Status LED
 - LED til indikation af robottens status. Synlig farve med 3 mulige tilstande. Tændt, slukket og blink med varierende interval.
- Connectors til
 - Batteri
 - 2 stk. Bipolar Stepmotor
 - BeagleBone Black Wireless



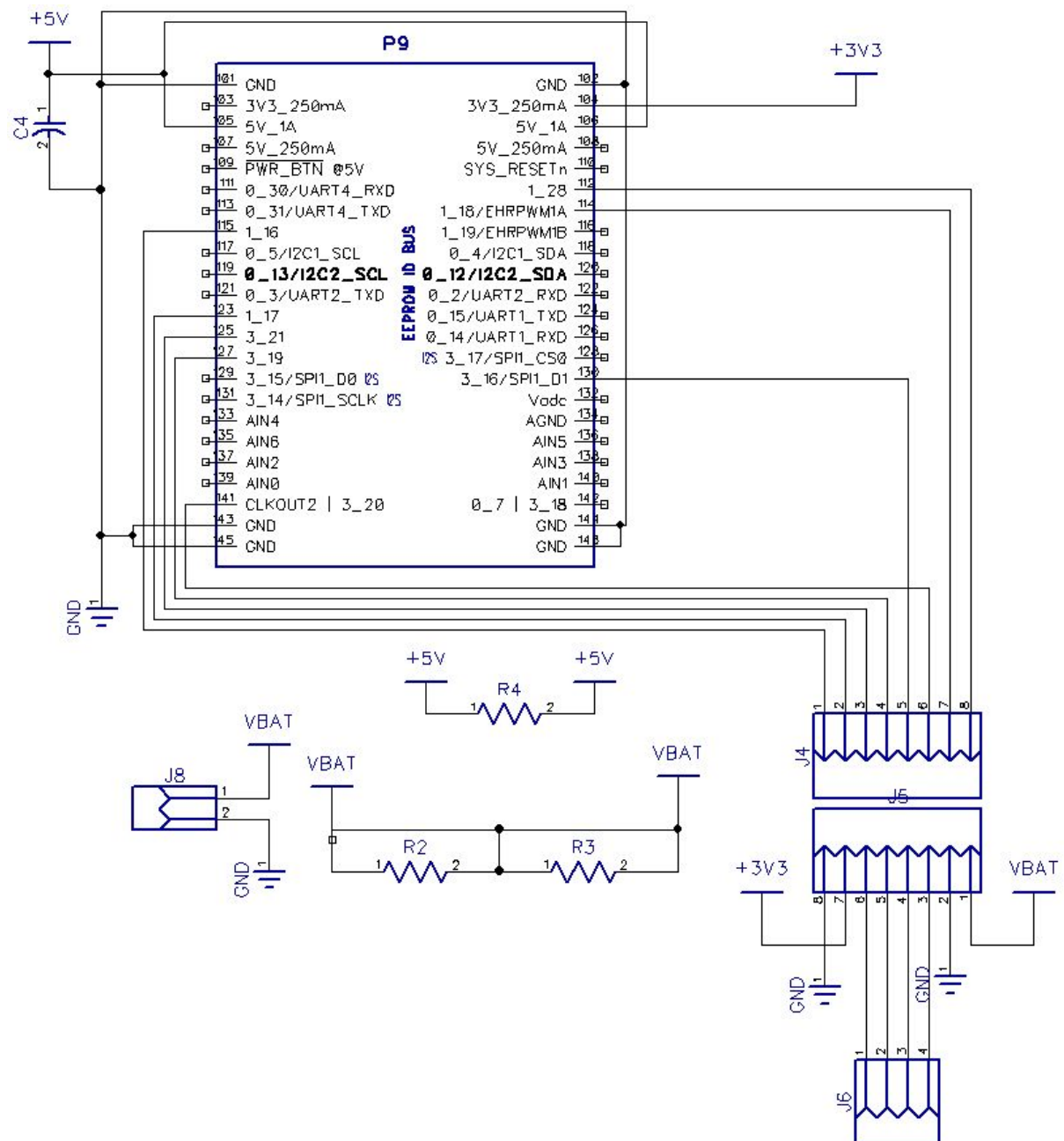
Figur 6.9 - Funktionalitetsdiagram for PCB: BBB-cape.

PCB er blevet designet efter de design constraints, der er blevet opstillet i Analysis & Design afsnittet i Timebox 3. Der er blevet benyttet en PCB calculator[15] til at beregne current carrying capacity, for de relevante traces.

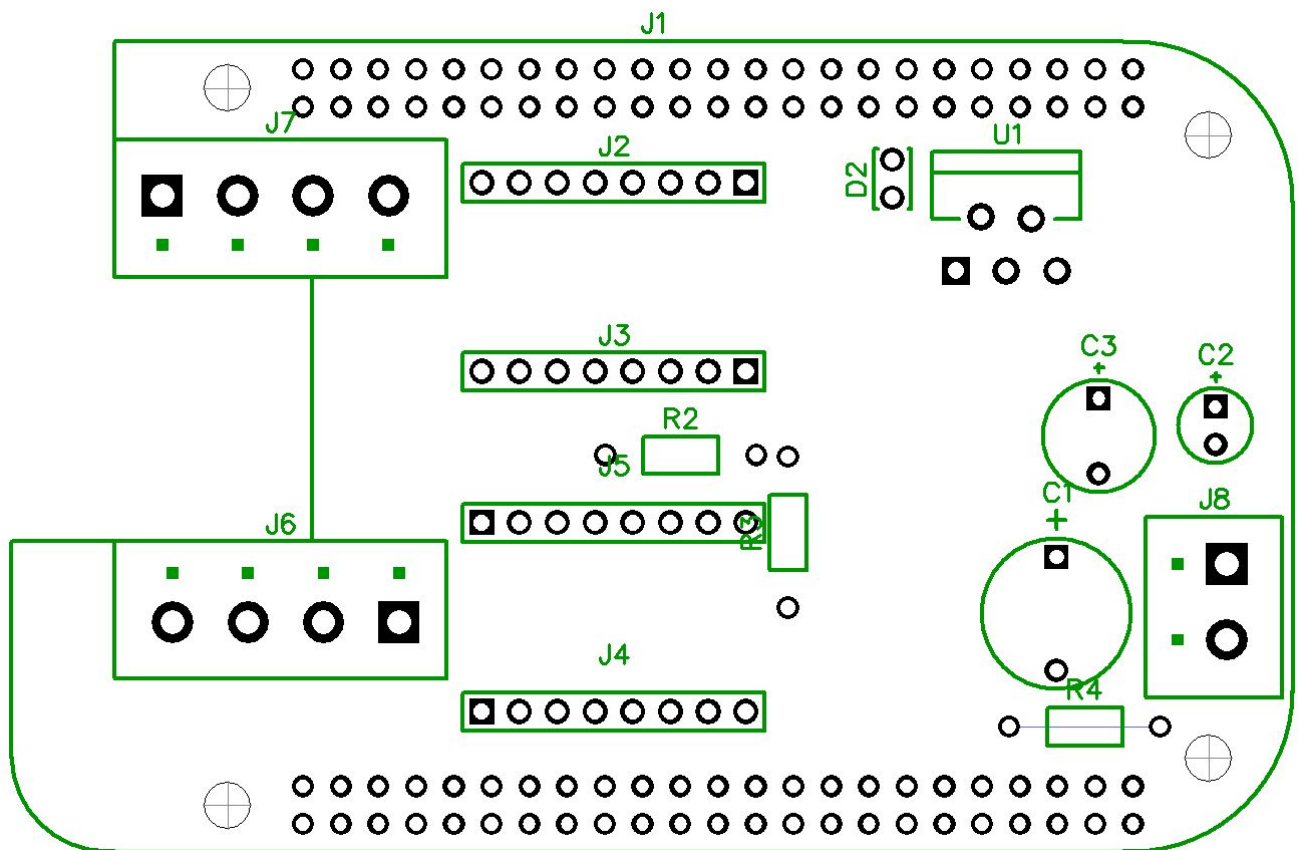
I designet er der lagt vægt på at få korte forsynings baner, dvs., at der er lagt 0 ohm jumpers modsat lange baner på printet. Ligeledes holdes stelplanet solidt.

Printet er udlagt som et single layer print, dette kan fremstilles hurtigt og billigt via direkte ætsning.

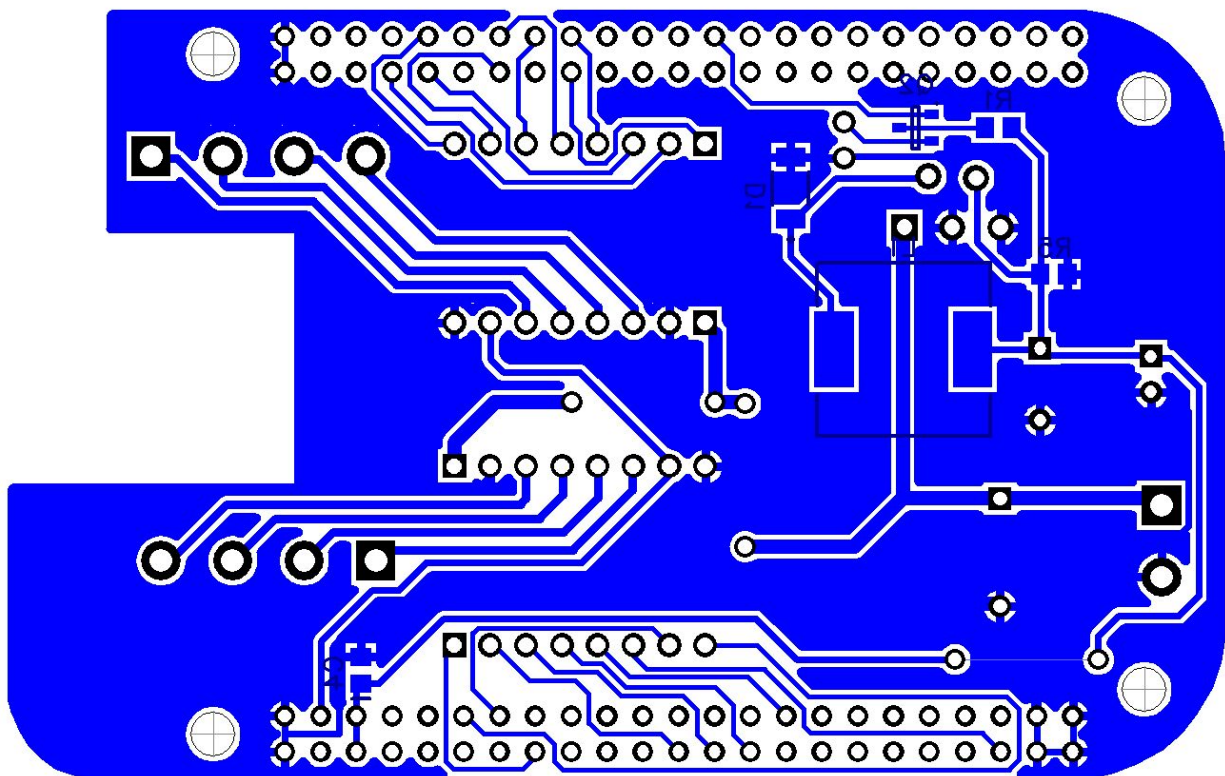
Diagram & PCB



Figur 6.10 - Beaglebone Black Cape Schematic (1/2)



Figur 6.12 - Beaglebone Black Cape PCB (Top Layer)

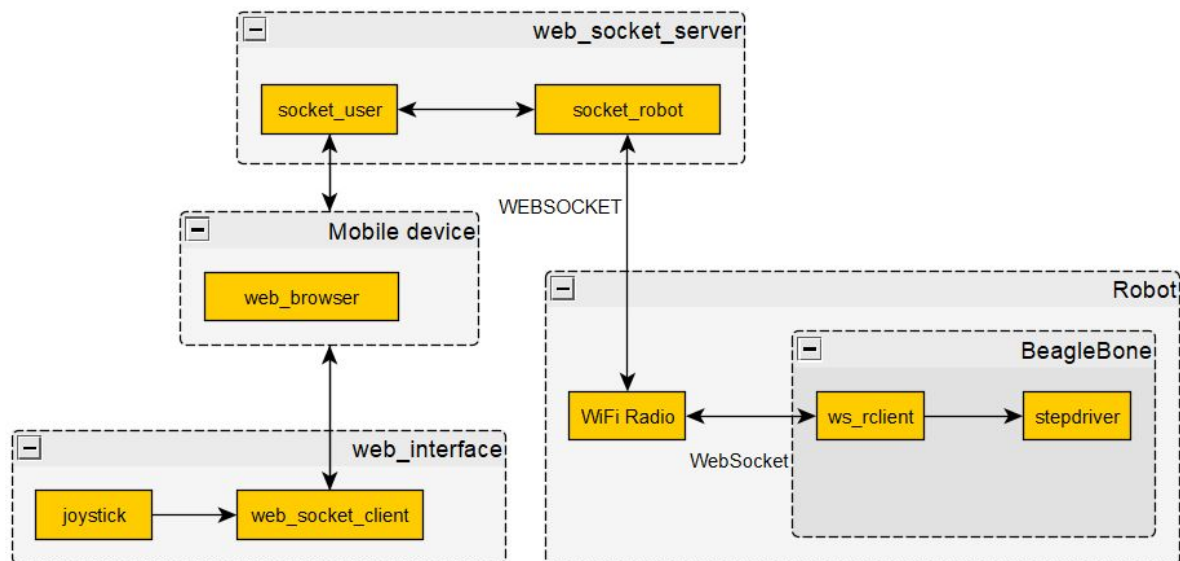


Figur 6.13 - Beaglebone Black Cape PCB (Bottom Layer)

For mere information om design af Beaglebone Black Cape [14].

7. Software

Softwareplatformen er baseret på 2 hovedkomponenter. Et webserver modul, som tilbyder et webinterface, hvorfra robotten kan betjenes med en webbrowser, samt et robot modul, der modtager kommandoer fra serveren og driver hardwaren.



Figur 7.1 -Diagram over det overordnede software design

Oversigt over software moduler. Alle moduler er open source og hostet på github.[16]

Software	Beskrivelse
webinterface[17]	Webserver modul, som indeholder webinterfacet i HTML og en websocket server ansvarlig for videresendelse af kommandoer til robotten. JavaScript/HTML.
ws_rclient[18]	Client på robot som opretter forbindelse til webserver og modtager kommandoer. Desuden håndteres status LED'en fra dette program. Skrevet i JavaScript.
stepdriver[19]	Linux program der tilgår GPIO/PWM interfacet. Modtager joystick positioner og beregner relative PWM værdier ud fra dette. Skrevet i C++.

Etablering af udviklingsmiljø

Følgende kommandoer benyttes første gang, for at etablere et udviklingsmiljø under Debian Linux.

Robot client (webinterface)

- `Sudo apt-get install nodejs`
- `Git clone https://github.com/PRO3-Team1/webinterface.git`
- `Cd webinterface`
- `Npm install`

For at starte programmet

- `node main.js`

Robot client (ws_rclient)

- *Sudo apt-get install nodejs*
- *Git clone https://github.com/PRO3-Team1/ws_rclient.git*
- *Cd ws_rclient*
- *Npm install*

For at starte programmet

- *Sudo node main.js*

(Bemærk brugen af sudo for tilgang til GPIO pins)

Robot client (stepdriver)

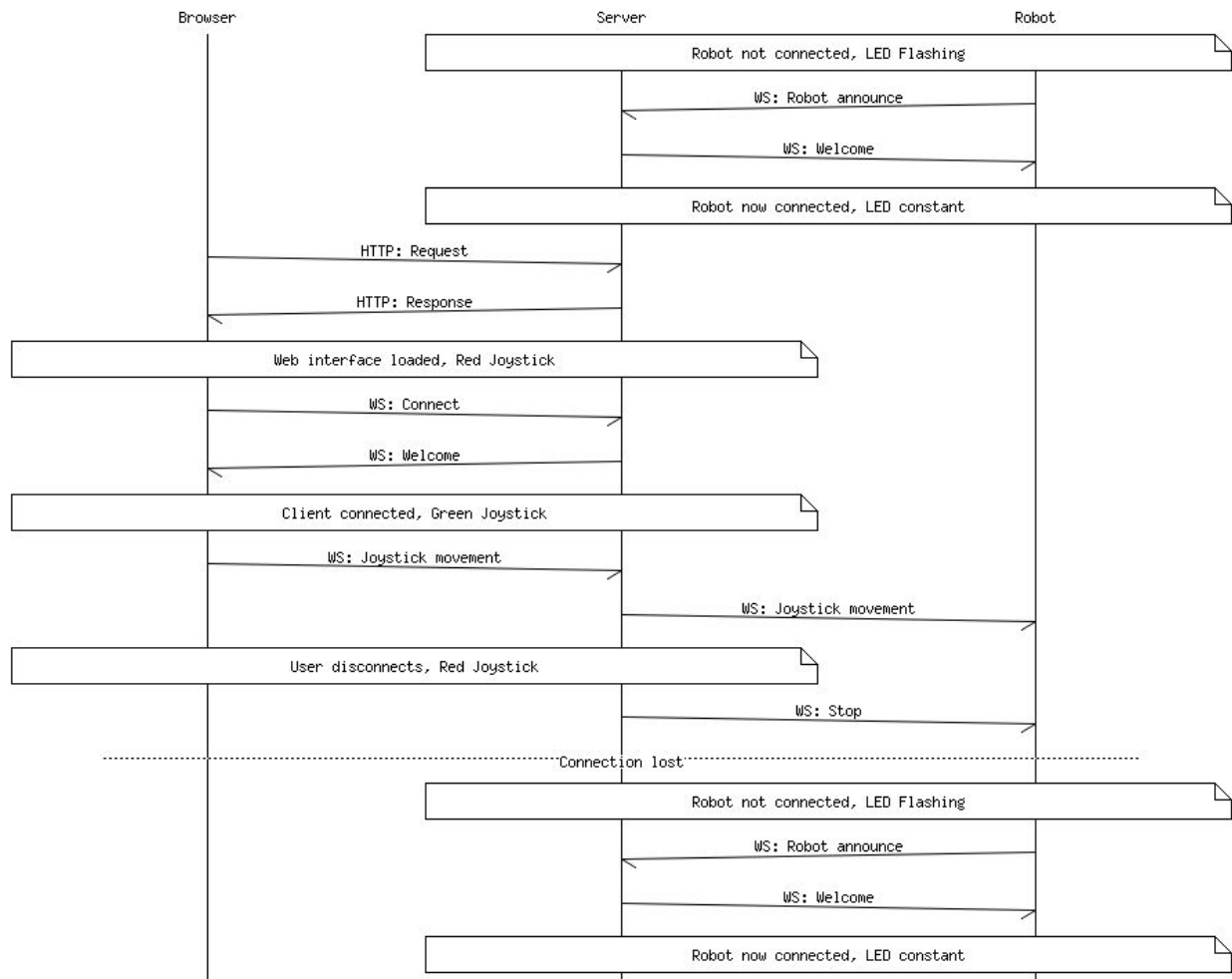
- *Git clone <https://github.com/PRO3-Team1/stepdriver.git>*
- *Cd ws_rclient*
- *make*

For at starte programmet

- *Sudo dist/Release/GNU-Linux/stepdriver*

(Bemærk brugen af sudo for tilgang til GPIO pins)

Kommunikation



Figur 7.2. Message sequence diagram for kommunikation mellem klient på robot og server.

Snitflader

Software snitflader er beskrevet i detaljer i bilaget[20]. Følgende snitflader benyttes:

webinterface - webserver - ws_rclient

Der benyttes en færdig implementation af WebSockets, som er testet og understøttet på tværs af platforme. Socket.io implementerer både backend og client side som begge benyttes. Socket.io tillader at sende data-pakker under såkaldte "eventName". Socket.io håndterer leveringen, reconnect og modtagelse.

Socket.io er skrevet under MIT licensen som tillader inkludering i dette projekt[21].

Følgende events er fastlagt under snitfladen fra webbrowseren til serveren, og fra webserver til ws_rclient på robotten.

Event	Data	Beskrivelse
start	-	Brugeren påbegynder bevægelse af joysticket
end	-	Brugeren bevæger ikke joysticket
move	Angle, Force	Joysticket bevæges, og positionen af joysticket sendes som en vektor med vinklen (i radianer) og længden (0-1)
connect	-	En forbindelse oprettes
disconnect	-	En forbindelse afbrydes

Figur 7.3 - Event tabel

ws_rclient - stepdriver

Der benyttes stdout fra ws_rclient til stdin på stepdriver. Dette giver en stor fleksibilitet i forhold til testning, og er en simpel måde at implementere kommunikation mellem 2 processer. Stepdriver accepterer stdin. Her benyttes et CSV format for at simplificere interfacet så meget som muligt.

1. Argument= Joystick tilstand: START, MOVE, END
2. Argument= Angle mellem 0-6.28 (Vinkel i radianer)
3. Argument= Relative kraft, float mellem 0 og 1.

- Kommandoer adskilles med newline.
- Linjer der begynder med # ignoreres af modtageren, således debug beskeder kan inkluderes i output.

Eksempel på tre telegrammer, i et case scenarie, hvor joysticket flyttes kort til 6.28 i angle og 0.95 i force og derefter slippes. Der vedhæftes en debug kommentar:

```
START
MOVE,6.28,0.95
#User released the joystick
END
```

ws_rclient - LED interface

Client til LED interfacet implementeres via Linux GPIO subsystem.

- Logisk HIGH på GPIO61 resulterer i en tændt LED
- Logisk LOW på GPIO61 resulterer i en slukket LED

Se yderligere dokumentation i Timebox 3 dokumentation.

stepdriver - GPIO output

GPIO snitflade fra stepdriver modul i henhold til Hardware afsnittet. Der benyttes standard Linux GPIO api'er gennem det virtuelle filsystem.

webinterface

Brugerinterfacet er fastlagt som et software joystick, hvorved larvefødderne kan betjenes.

- Joysticket er én-finger betjent fra enhver moderne webbrowser.
- Interfacet er minimalt, og har ingen forstyrrende elementer.
- Feedback er begrænset til farvevalg og interaktion.
- Baggrundsfarven er grøn, når der er oprettet forbindelse.
- Hvis brugeren ikke benytter joysticket i 30 sekunder afbrydes forbindelsen.
- Baggrundsfarven er rød ved manglende forbindelse til server, eller tilsvarende fejlsituationer.
- Betjening af robotten er ligetil uden indledende introduktion.

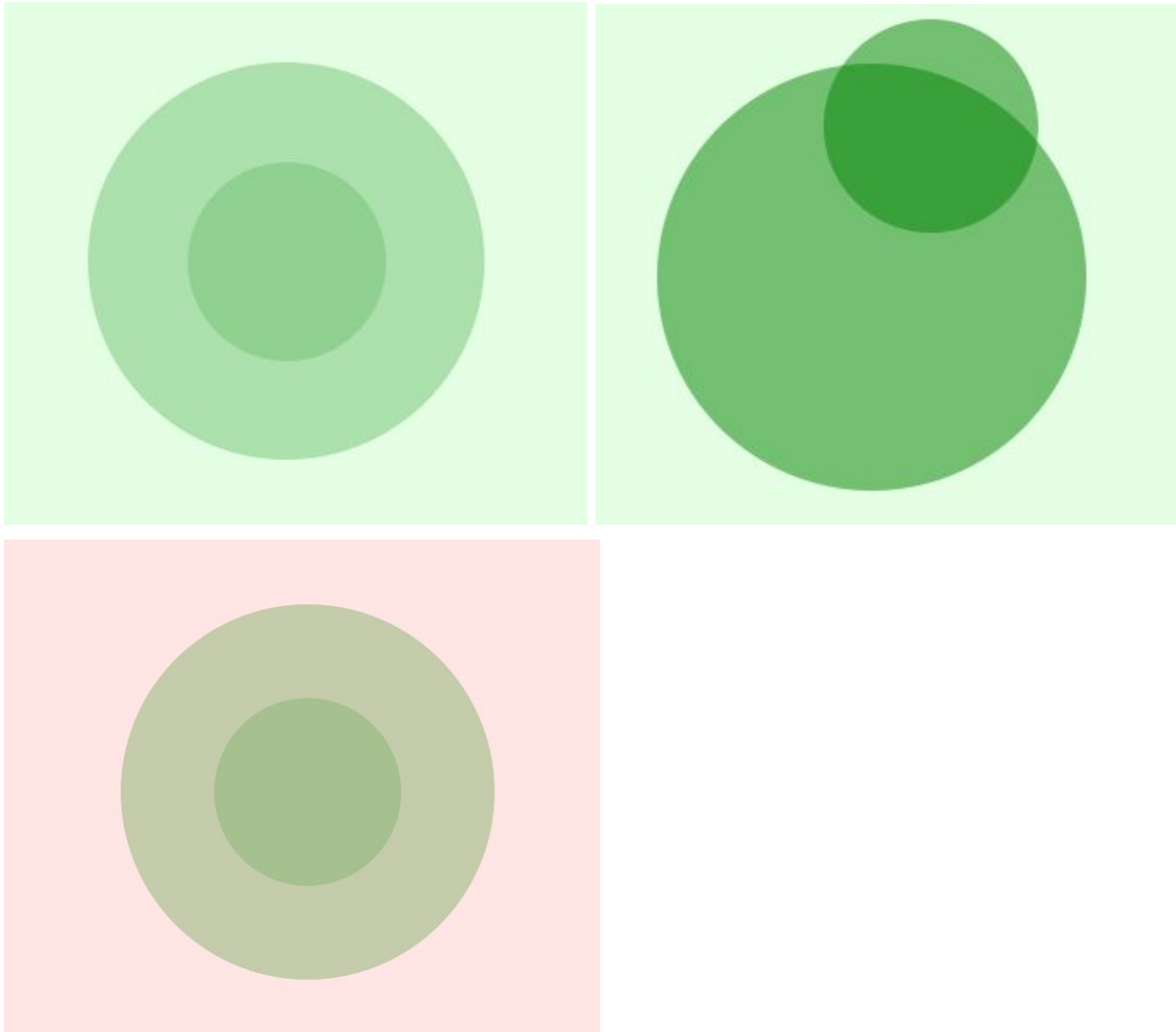
Server side implementering foretages i NodeJS. Da udviklingsplatformen primært vil være Windows og deployment platformen ikke er fastlagt, er det for projektet vigtigt at server side implementeringen kan eksekveres på både Windows og Linux baserede platforme.

NippleJS er skrevet under MIT licensen som tillader inkludering i dette projekt.[22]

Joystick

Interaktivt joystick[23]

Debug output fra joystick til udvikling[24]



Figur 7.4 - Joystick i hhv. 'neutral' aktiv' og 'inaktiv' modes

Websocket

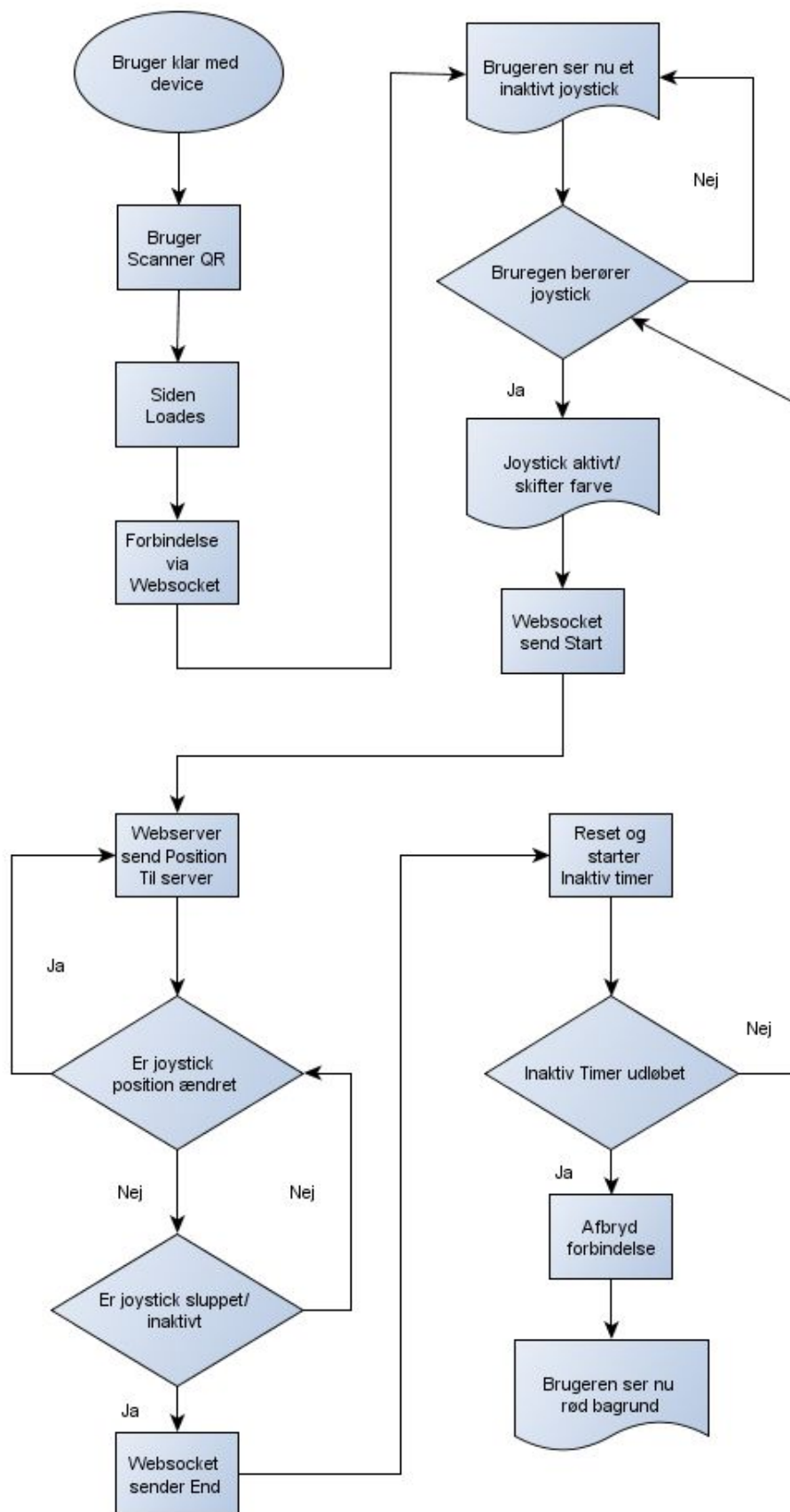
Der benyttes en færdig implementering af WebSockets som er testet og understøttet på tværs af browser platforme. Socket.io implementerer både backend og client side som begge benyttes. Socket.io tillader at sende datapakker under såkaldte "eventName".

Socket.io håndterer leveringen, reconnect og modtagelse.

Socket.io er skrevet under MIT licensen som tillader inkludering i dette projekt.

<http://socket.io>

I figur 7.5 er det valgt at illustrere virkemåden/interaktionen med et flow-diagram.



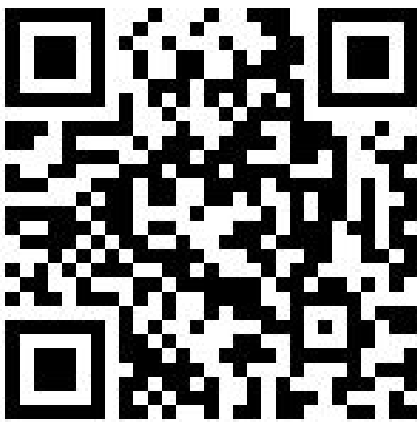
Figur 7.5 - Flow diagram over "behavioural design".

Hosting

Løsningen er designet til deployment til Herokus gratis service. Heroku's gratis plan tillader hosting op til 18 timer i døgnet. Herokus backend er automatisk i stand til at trække seneste commit fra git. Der startes automatisk en instans af serveren når en bruger anmoder om interfacet.

Interface hostet på Herokus platform.[23]

En QR kode er blevet autogenereret til at henvise til linket:



Figur 7.6 - QR kode med link til joystick

ws_rclient

Opgaver for ws_rclient til hvert interface.

1. ws_rclient og web_socket_server gennem WIFI radio
 - a. Initialiser netværksforbindelse og opretholder forbindelse.
 - b. Robot annoncerer forbindelse til server.
 - c. Detektion af forbindelsesstatus.
 - d. Genoprettelse af forbindelse ved udfald.
 - e. Modtag float angle, float force og joystick tilstand (MOVE, STOP).
 - f. Korrekt nedlukning af system.
2. ws_rclient og controller i beaglebone
 - a. Output float angle, float force og joystick tilstand (MOVE, STOP).
 - b. Output led-status.
3. ws_rclient og LED-status
 - a. Forbindelsesstatus omdannes til et signal.
 - b. Ud fra signalet dannes et LED-mønster, der signalerer de forskellige tilstande.

Stepdriver

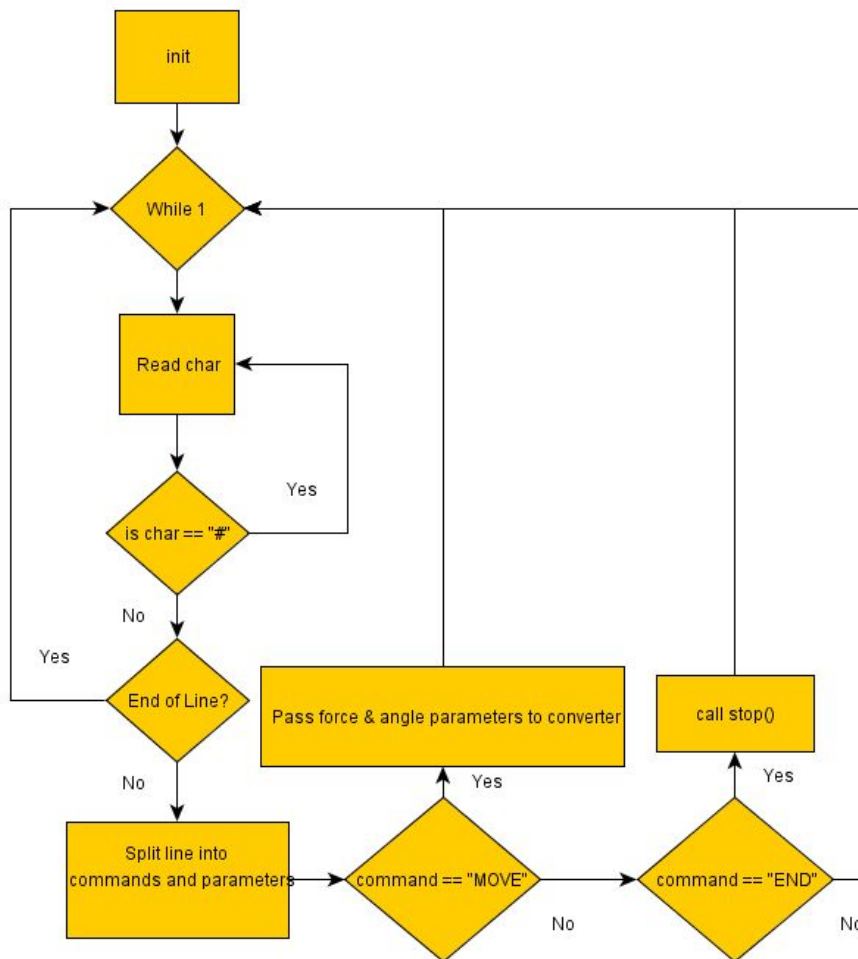
Opgaver er defineret for controller til hvert interface.

1. Controller og ws_rclient
 - a. Modtag et kommasepareret telegram fra ws_rclient
 - b. Læs hvilken kommando der er blevet modtaget
 - c. Kald konverterings algoritme hvis kommandoen er "MOVE"
 - d. Kald stepmotor.stop() hvis kommandoen er "END"
2. Controller og venstre/højre driver
 - a. Konverter Force og Angle parametre til Dir og Step på højre motor og Dir og Step på venstre motor.

Step output styres med PWM pins på BeagleBone Black. Til det benyttes "EBBLibrary"[25], koden til dette er inkluderet i projektet.

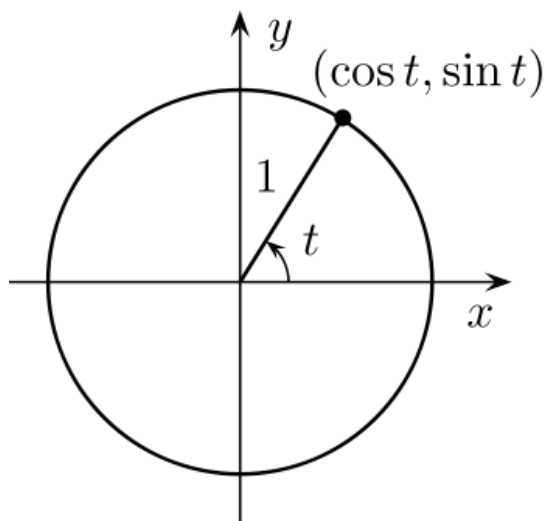
Programmet reagerer på telegrammer beskrevet i interfacet fra ws_rclient. Der er tre mulige kommandoer, og kun MOVE kommandoen har parametre der skal håndteres, og START kommandoen bruges ikke i nuværende implementering. END kommandoen modtages, og håndteres som stop af stepmotorerne.

Når programmet modtager en MOVE kommando er den efterfulgt af parametrene "ANGLE" og "FORCE". På formatet "MOVE,<0-2*pi>,<0-1>" Programmet konverterer disse parametre til en step frekvens på hhv. højre og venstre stepmotor.



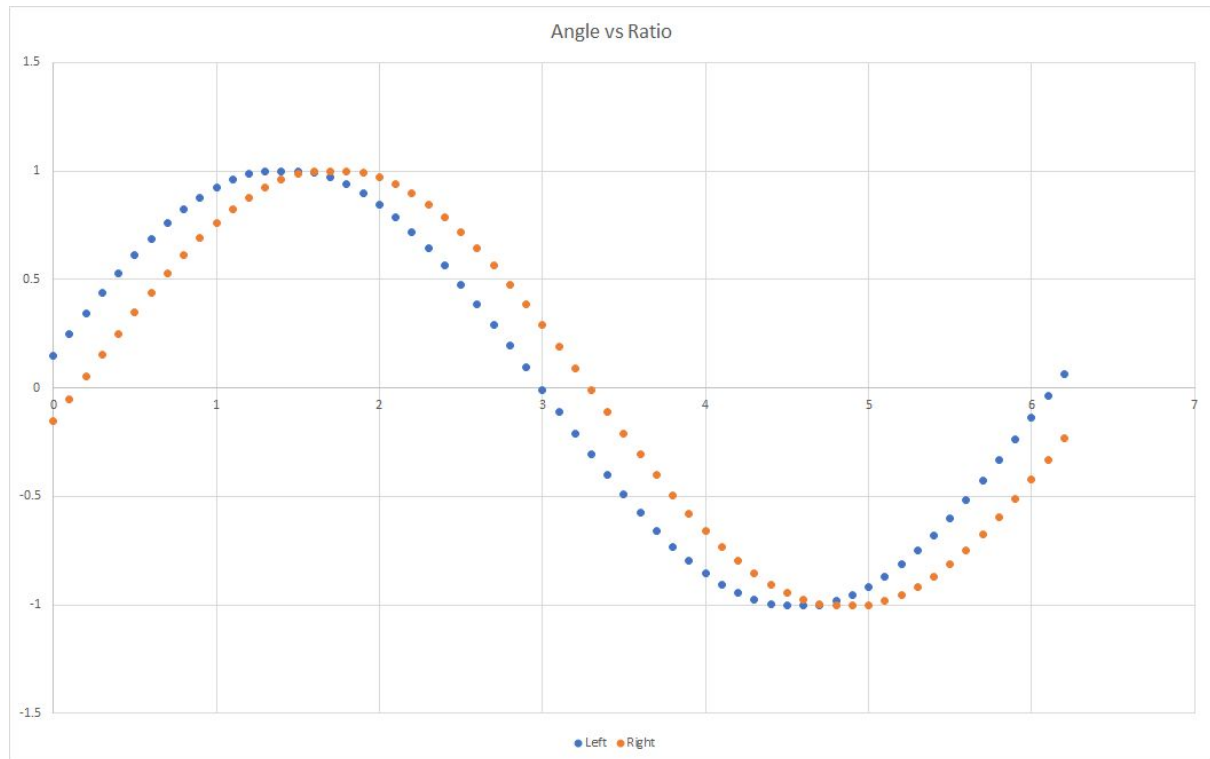
Figur 7.7 - Stepdriver flowchart

Algoritmen tager udgangspunkt i en enhedscirkel. Ud fra denne cirkel blev de intuitive hastigheder for de to motorer fastlagt. Her observeres at sinus til angle svarer til den fremadrettet komponent og cosinus til angle svarer til den hastighed hvormed der skal drejes.



Figur 7.8 - Enhedscirkel med cos og sin.

Der er indført en SPEED og en TURN factor, som kan bruges til at justere hastigheden og motorernes hastighed når de drejer. Der er oprettet et Excel dokument som visualiserer effekten af SPEED og TURN factors grafisk. Se reference [26].



Figur 7.9 - Forhold mellem Left og Right ratio med turn factor 0.15. X-aksen er radianer og y-aksen er hastighedsfaktor for hver motor.

Grafen går fra 0 - 2π . Ved $\pi/2$ (1.57) vil vi være imellem 1. og 2. Kvadrant på enhedscirklen og her skulle begge motorer køre fremad med 100%, hvilket også ses.

Implementation

```
#define SPEED_FACTOR 1.0
#define TURN_FACTOR 0.15

extern double lRatio;
extern double rRatio;

int angleCalc(double Ang)
{
    double forward = sin(Ang) * SPEED_FACTOR;
    double leftright = cos(Ang) * TURN_FACTOR;

    lRatio = max(-1.0,min(1.0, forward + leftright));
    rRatio = max(-1.0,min(1.0, forward - leftright));

    return 0;
}
```

Figur 7.10 - kodeafsnit med udregninger til styringen af motorer.

Via build-measure-learn loops blev TURN_FACTOR fastlagt til 0.15. Dette gav en naturlig brugeroplevelse i forhold til navigation når robotten skal foretage drejninger på stedet. Arbejdet med den nye driver er dokumenteret i bilaget [27]

Programmet skal pipes i linux terminalen så den modtager stdout fra ws_rclient ind på stdin på følgende måde:

```
debian@beaglebone:~$ node ws_rclient/main.js | sudo ./stepdriver listen
```

Figur 7.11 kode til eksekvering af programmer.

Programmet skriver direkte til PWM og GPIO virtuelle filer i filsystemet.

Opstartsløsning

Robotten kan bruges uden forbindelse til en PC, hvor den tændes og styres fra en telefon. Programmerne til styring af robotten fra et webinterface starter op automatisk, og output og input forbindes, når BBB booter ved tilføring af strøm.

Designkrav til program:

1. Program eksekverer automatisk ved opstart.
2. Programmet starter op med root tilladelse, så der er adgang til at styre I/O, PWM m.m.
3. Punkt 1 og 2 udføres uden periodiske undtagelser.

Opstartsløsning er implementeret via systemd. En unit file for servicen er oprettet under /etc/systemd/system/team1robot.service. Den peger mod et linux shell script som eksekverer de 2 programmer og piper output imellem dem.

```
debian@beaglebone:/etc/systemd/system$ cat team1robot.service
[Unit]
Description=TEAM1 Robot

[Service]
ExecStartPre=/bin/sleep 5
ExecStart=/bin/sh /home/debian/robotstart.sh
Restart=on-failure
User=root
Group=root

[Install]
WantedBy = multi-user.target
```

Figur 7.12 - Systemd unit file

```
debian@beaglebone:~$ cat robotstart.sh
/usr/bin/node /home/debian/ws rclient/main.js | /home/debian/stepdriver
```

Figur 7.13 - Start up script

Robotten er konfigureret til AU-Gadget netværket, således den automatisk opnår forbindelse til internet i AU's bygninger.

8. Konklusion

Opsummering

Systemet består af et batteridrevet robotbælte køretøj. Robotten er udstyret med en Beaglebone Black wireless. Motorerne styres gennem en applikation, der får informationer fra brugeren via et klientprogram. Brugeren kommunikerer med robotten via et webinterface med et joystick på en vilkårlig mobiltelefon, eller anden enhed, som kan load webinterfacet ind. Selve køretøjet kan manøvreres frem, tilbage og til siderne. Hastigheden holder sig inden for specifikationerne. Køretøjet kan uden fysisk menneskekontakt, bringe sig i stand til at køre videre, efter at være væltet, eller på anden måde være bragt ud af normal køreposition. Robotten kan bruges i mindst 30 minutter, og evt. batteriskift kan foretages på mindre end 5 minutter.

Herudover var det målet, at den skulle være i stand til at køre op af trapper og, at køretøjet skal være tilstrækkeligt robust til at kunne tåle gentagne fald fra bordhøjde og genoptage kørsel derefter. Disse mål er ikke nået eller er bortfaldet. Robotten kan køre op af en hældning på omkring 16 grader, og det ønskes, fra kundens side, ikke at teste fald fra bordhøjde.

Produkt evaluering

Som det kan ses ud fra figur 8.1 er de fleste krav opfyldt, der er dog stadig nogle områder at arbejde videre med. Det vil være en god ide, at anskaffe nogle bælter, som er designede til formålet. Disse bælter skal have en overflade med bedre friktion, og som kan gribe fat i

trappen. Det kan være en idé at fremstille hele robotten i et andet materiale, der er lettere og tyndere. En anden implementering af beslaget til monteringen af tandhjulene kunne være en fordel, idet vibrationerne fra robotten kan få pinolskruen, der holder tandhjulet fast kan blive skruet op til trods for, at det har fået "lock tight". Det er en mulighed at få mere kraft ud af motorerne, ved at implementere gearing. Der kan fjernes materiale fra selve strukturen i robotten, som på denne måde kan mindske vægten uden at, fjerne dens strukturs stabilitet.

Oplagte ideer til fremtidig udvikling vil være et mere avanceret interface hvorfra robottens status kan udlæses. For eksempel kan en "lipo shutoff" funktion, der sørger for at beskytte lipobatteriet ved at lukke systemet ned, når batterispændingen bliver for lav. Herunder kunne tænkes et køsystem hvor en instruktør kunne tildele betjeningen af robotten eller ændre indstillinger.

Stepmotorene har den negative egenskab, at de bruger den samme mængde strøm hvad enten de holdes stille eller drives. En sleep funktionalitet til stepmotorer kunne implementeres, således der ikke leveres strøm igennem spolerne når robotten er inaktiv. Det kunne tænkes, denne funktionalitet udvides med gyrosensor eller andet således, robotten bevarer sit holdemoment hvis den ikke er placeret på en vandret flade.

Projektoplægget krævede brugen af en BeagleBone Black. Det ville være muligt at erstatte denne i projektet med en væsentlig billigere processor, da den implementerede funktionalitet er meget begrænset. Herunder bør der lægges overvejelser omkring hvordan softwaremodulerne kan fordeles mellem det webinterface, webserveren og robotten.

Requirement and test case reference	Design document reference	Implementation reference	Test acceptance date yyyy-mm-dd and initials	Comments
1	Timebox 8	Timebox 8	2018-04-11 TV	Udført og bestået. Ingen bemærkninger
1.1	Timebox 4	Timebox 4	2018-01-5 TV	Udført og bestået på testbench. Video i timebox 4 mappe.
1.2	Timebox 4	Timebox 4	2018-01-5 TV	Udført og bestået på testbench. Video i timebox 4 mappe.
1.3	Timebox 8	Timebox 8	2018-04-11 TV	Udført og bestået. Ingen bemærkninger
2	Timebox 8	Timebox 8	2018-04-11 MICH	Udført, og bestået med en hastighed på 40,5 cm/s.
3	Timebox 6	Timebox 6	2018-03-05 ARE	Udført og bestået
3.1	Timebox 1, Design	Timebox 1, Implementation	2017-11-08 Mich	Udført og bestået på iPhone, windows PC med mus og Android version >5.0.
3.2	Timebox 6	Timebox 6	2018-03-05 ARE	Udført og bestået
4	Timebox 8	Timebox 8	2018-04-18 JCT	Udført og bestået. Kræver nye plastik strips for hver udskiftning
5	Timebox 9	Timebox 9	2018-04-20 Mich	Udført og bestået.

6	Timebox 8	Timebox 8	2018-04-11 MICH	Bestået.
6.1	Timebox 1, Design	Timebox 1, Implementation	2017-11-08 Mich	Udført og bestået på iPhone, windows PC med mus og Android version >5.0.
6.2	Timebox 1, Design	Timebox 1, Implementation	2017-11-08 Mich	Udført og bestået på iPhone, windows PC med mus og Android version >5.0.
7	Timebox 8	Timebox 8	2018-04-11 MICH, TV	Bestået.
7.1	Timebox 1, Design	Timebox 1, Implementation	2017-11-08 Mich	Udført og bestået på iPhone, windows PC med mus og Android version >5.0.
7.2	Timebox 6	Timebox 6	2018-03-05 ARE	Udført og bestået
8	Timebox 8	Timebox 8	2018-04-11 MICH, TV	Udført, bestået, 84.5 cm
9	Timebox 9	Timebox 9	2018-04-18 JCT	Udført og bestået. Testet efter 40mins brug. Batterispænding 15.35V efter udførelse
10	Timebox 8	Timebox 8	2018-04-11 ARE	Udført og bestået uden bemærkninger.
10.1	Timebox 4	Timebox 4	2018-01-5 TV	Udført og bestået på testbench. Se video i timebox 4 mappe.
10.2	Timebox 8	Timebox 8	2018-04-11 ARE	Udført og bestået uden bemærkninger.
11	Timebox 8	Timebox 8	2018-04-11 JCT	Udført og bestået uden bemærkninger.
11,1	Timebox 1, Design	Timebox 1, Implementation	2017-11-08 Mich	Udført og bestået på iPhone, windows PC med mus og Android version >5.0.
12	Timebox 5	Timebox 5	2018-02-19 JCT	Udført og bestået
13	Timebox 8	Timebox 8	2018-04-18 JCT	Fejlet underkrav
13.1	Timebox 8	Timebox 8	2018-04-11 TV	Udført og bestået
13.2	Timebox 7	Timebox 7	2018-03-20 ARE	Udført og bestået.
13.3	Timebox 9	Timebox 9	2018-04-18 JCT	Fejlet. Robotten glider
13.4	Timebox 9	Timebox 9	2018-04-18 JCT	Fejlet. Robotten kan ikke løfte sig op på første trin
13.5	Timebox 9	Timebox 9	2018-04-18 JCT	Fejlet. Afhænger af krav 13.3 og 13.4
14	Timebox 9	Proces dagbog /timebox 9	2018-04-16 Mich, TV	Dispensation givet. Vi ønsker ikke at beskadige robotten under udførelse af denne test
15	Timebox 8	Timebox 8	2018-04-11 MICH, TV	Udført og bestået. forbrug: 32%
16	Timebox 9	Timebox 9	2018-04-18 JCT	Udført og bestået. 1.8Amp konstant forbrug.

17	Timebox 9	Timebox 9	2018-04-18 JCT	Fejlet. Afhængig af requirement 13 som ikke er opfyldt
----	-----------	-----------	----------------	--

Figur 8.1 - Requirement Verification Traceability Matrix

Procesevaluering

I den generelle strategy, der blev vedhæftet i hver timebox havde vi to artefakter vi løbende opdaterede. Den ene var "general requirement plan" og den anden var "Requirement, Verification, and traceability Matrix". Vi endte med blot at bruge RVTM og bruge en masse tid på at få den generelle plan til at passe efterfølgende. Det var meget tidskrævende idet der kom flere underkrav til løbende.

Det ville være en god idé, at vi fra starten havde delt vores krav op i flere underkrav, så der ikke er for mange krav, der skal løses til sidst. Vi havde mange krav, der var afhængige af andre krav. F. eks. starter mange af vores krav med "robotten skal". Hvilket betyder at kravet ikke kan løses før der er en robot.

Vi vil gerne droppe den store inddeling i "shall, will, must etc" og holde os til den prioriterede rækkefølge. Til gengæld vil vi have større fokus på den præcise ordlyd i hvert krav, så der står "skal", hvis det er essentielt, eller "må gerne", hvis det er mindre vigtigt.

Der er erfaret at elektronik kan købes for billigt, og det kunne nok have sparet os for noget tid at have anskaffet nogle ordentlige drivere med dokumentation fra starten, fra en pålidelig forhandler. Dette er også selv om det vil kræve, at vi venter på en længere levering, her bør en god planlægning sikre dette ikke påvirker projektet.

Det har været forsøgt at holde projektet så tæt som muligt på den oprindelig plan. Når dette ikke har været muligt grundet modgang fik stopklodserne ekstra opmærksomhed. Til tider har denne prioritering måske ikke været optimalt, da der blot er brugt flere ressourcer uden nødvendigvis en dybere vurdering af problemstillingen. Problemer bør tidligt bringes op i gruppen således der kan foretages en objektiv vurdering af mulige løsninger.

Efterhånden fik vi et større fokus på generel development plan. Hver timebox brugte vi lang tid på at gå den kritisk igennem, opdatere den og udspecificere opgaver. Derefter kunne der laves en grundig deployment beskrivelse, og det var en stor hjælp. Det skabte et godt overblik i starten af hver timebox, og det var tydeligt og præcis hvad der skulle laves, så det var nemmere at dele opgaverne op, og på den måde arbejde mere effektivt. Det skabte også klarhed over hvornår vi var færdige med en timebox, og hvad der skulle afleveres.

Vi lærte med tiden at begrænse mængden af procesarbejde i hver timebox. Konsekvensen har været, at vi ikke har nået igennem alle dele af EUDP-metoden, men har haft fokus på de processer og metoder, der ville støtte udviklingsarbejdet, og som ville kunne bruges i den tekniske dokumentation. På den måde fik vi mere overskud til at fokusere på at udvikle produktet og lave noget præcis dokumentation. Det blev også afspejlet i, at en stor del af den dokumentation, der blev udviklet i de første timeboxes ikke var særlig relevant senere.

Kundens afsluttende kommentarer

Nedenstående er en opsummering af deploymentmøde med kunden, se bilag [28].

“Jeg er imponeret over at i har nået så langt, trods alle de mekaniske problemer. Den kan flytte sig, og den kan de fleste ting. På nuværende tidspunkt er produktet et udtryk for en prototype, der kan arbejdes videre med. Det er en meget stor og synlig dims vi kan tage med på åben hus dage. Det er ikke så tit, at projekter når så vidt, at man næsten når målstregen. Deploymentbeskrivelsen og verifikation har fungeret rigtigt godt for jer.

Det var fint prioriteret at få lavet noget, der kunne vises i starten. Et synligt produkt. Det var fint at skabe luft for at give plads til alle de problemer, der nok skal komme. Det gjorde de også. Hvis i nu havde startet med mekanikken, så er det ikke sikkert, at i var kommet videre. “

9. Referencer

- [1] <http://eudp.dk/index.php/Documentation>
- [2] Bilag: E3PRO3 Projektforslag Bæltedrevet køretøj
- [3] Bilag: Kundemøde om krav med Per Lysgaard
- [4] Bilag: Launch Phase (System Architecture Design)
- [5] Bilag: Timebox 4 - Afsnit 4.8 - 4.10
- [6] Bilag: Timebox 7 - Afsnit 7.5
- [7] Bilag: Timebox 8 - Afsnit 8.2
- [8] Bilag: Timebox 9 - Afsnit 9.2
- [9] Bilag: Timebox 7 - Afsnit 7.3
- [10] <https://dk.rs-online.com/web/p/buck-convertere/5333743/>
- [11] Bilag: Timebox 3 - Afsnit 3.2 - 3.4
- [12] Bilag: Timebox 7 - Afsnit 7.3
- [13] https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf (side 73)
- [14] Bilag: Timebox 3 - Afsnit 3.5 - 3.8
- [15] <http://www.skottanselektronik.com/>
- [16] <https://github.com/PRO3-Team1/>
- [17] <https://github.com/PRO3-Team1/webinterface.git>
- [18] https://github.com/PRO3-Team1/ws_rclient.git
- [19] <https://github.com/PRO3-Team1/stepdriver.git>
- [20] Bilag: Timebox 5 - Afsnit 5.6 - 5.9
- [21] <http://socket.io>
- [22] <http://yoannmoinet.github.io/nipplejs/>
- [23] <https://pro3-robot.herokuapp.com/>
- [24] <https://pro3-robot.herokuapp.com/robot>
- [25] <https://github.com/derekmolloy/exploringBB/tree/master/library>
- [26] Angle calculator
- [27] Timebox 8 - Afsnit 8.7
- [28] PostProjekt - Konklusion
- [29] Requirements and Verification

10. Appendix

Angle calculator.pdf

E3PRO3 Projektforslag Bæltedrevet køretøj.pdf

General Strategy and Planning.pdf

Kundemøde om krav med Per Lysgaard.pdf

Launch Phase.pdf

PostProjekt.pdf

PreProject.pdf

Requirements and Verification.pdf

Timebox 1.pdf

Timebox 2.pdf

Timebox 3.pdf

Timebox 4.pdf

Timebox 5.pdf

Timebox 6.pdf

Timebox 7.pdf

Timebox 8.pdf

Timebox 9.pdf