

Forår 2023

Semesterprojekt 1

SW9Vejleder: Henrik Olsen / Henning Hargaard**Dato: 16/06/2023**

#1	Stud.nr.: 202209875	Navn: David Tam Minh Nguyen
#2	Stud.nr.: 202210047	Navn: August Thorbjørn Duelund Gravesen
#3	Stud.nr.: 202209869	Navn: Mads Dittmann Villadsen
#4	Stud.nr.: 202209876	Navn: Anders Roman Knudsen
#5	Stud.nr.: 201808281	Navn: Nahom Melakehail Tadesse

Indholdsfortegnelse

Arbejdsfordeling	1
Indledning	1
Problemformulering	1
Kravspecifikation	5
Funktionelle krav	6
Use case 1: "Kør banen"	6
Use case 2: "Afspil lyd"	6
Use case 3: "Styr forlys"	7
Use case 4: "Styr baglys"	7
Ikke-funktionelle krav	7
Acceptspecifikation	8
Accepttestsspecifikation for funktionelle krav (use cases)	8
Accepttestsspecifikation for ikke-funktionelle krav	11
Systemarkitektur	14
Hardwarearkitektur	14
Block Definition Diagram (BDD)	14
Softwarearkitektur	16
Internal Block Diagram (IBD) (Anders)	16
Forlys (Nahom)	18
Baglys (Mads)	18
Sensor/Interrupts (Mads)	18
Motor (Anders og David)	19
Lyd (August)	20
Hardwaredesign	22
Hardware for lys (Nahom og Mads)	22
Forlys (Nahom)	22
Baglys (Mads)	25
Hardware for motor (Anders og David)	30
Komponenterne i kredsløbet	30
Hardware for lyd (August)	33
Indledning	33
Skitsering/Diagram	33
Fysisk konstruktion på veroboard	33
Hardware for power supply(Anders)	36
Softwaredesign	36
Lys (Nahom og Mads)	36

Sensor/Interrupts (Mads og August).....	37
Motor (Anders og David)	38
Lyd (August).....	42
Main (Anders og David)	46
Implementering og modultest	47
Lys (Nahom og Mads)	47
Motor (Anders og David)	49
Lyd (August).....	50
Integrationstest (David og Anders)	50
Accepttest (David).....	55
Funktionelle krav (use cases).....	55
Ikke-funktionelle krav	58
Konklusion (Mads)	61
Konklusion for hardwaredelen	61
Konklusion for softwaredelen	61
Konklusion for projektet	61
Individuelle konklusioner	62
Mads	62
David	62
Nahom	62
August	63
Anders	63
Bilagoversigt.....	64
Bilag 1. Konkurrencen	64
Bilag 2: Fysiske mål for karrosseriet.....	64
Bilag 3: Tidsplan	64
Bilag 4: Samarbejdsaftale.....	64
Bilag 5: Mødeindkalder + referater	64
Bilag 6: Main.c	64
Bilag 7: Lysdiode-øvelse (øvelse 3) - PRJ1	64
Bibliografi	64

Arbejdsfordeling

Opgaver / Navne	Mads	August	David	Nahom	Anders
Motor			✓		✓
Sensor	✓				✓
Lys	✓			✓	
Lyd		✓			

Tabel 1: Arbejdsfordeling

Indledning

I 1. semesterprojektet er formålet at få en selvkonstrueret bil til at fuldføre en gennemkørsel af en opsatbane med forskellige forhindringer og sensorer, som bilen skal kunne reagere på. Der er 4 komponenter, som skal implementeres på bilen, som gruppen har fået sat til rådighed af studiet. Disse 4 komponenter er motor, sensorer, lys og lyd. Når bilen er konstrueret, skal der implementeres software, som skal være med til at få bilen til at køre frem og tilbage ved forskellige hastigheder, lyse på forskellige niveauer, samt få den til at reagere på forskellige refleksler ved hjælp af sensorerne, som aktiverer at en lyd/melodi afspilles. Der er til gennemførsel af projektet lavet både HW- og SW-arkitektur, som er med til at give et bedre overblik over kompleksiteten ved bilen, og hvordan de forskellige komponenter er koblet sammen. Der er lavet forskellige kravspecifikationer til bilen, samt en accept-test, som er med til at teste, om bilen lever op til de forskellige satte krav.

Problemformulering

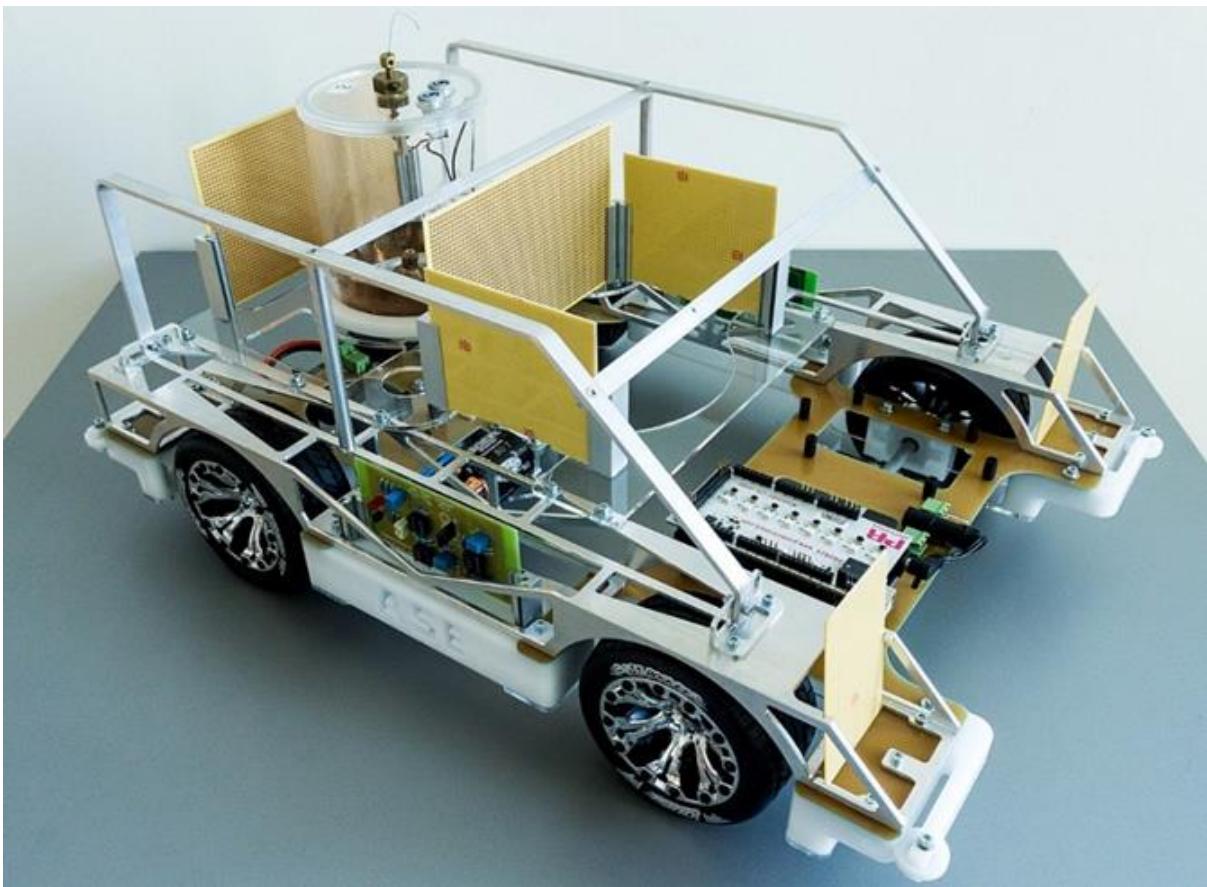
Det tværfaglige projekt PRJ1, der gennemføres på 1. semester for E- og SW-studierne, drejer sig om udvikling af hardware og software til styring af en elektrisk bil.

Som en indledning til projektet gennemføres 4 øvelser i E-LAB. Disse kan betragtes som en forundersøgelse eller analyse af de teknikker, der tænkes anvendt under projektet.

Projektet skal udarbejdes efter retningslinjerne i dokumentet "Vejledning til gennemførelse af projekt 1".

Under projektets forskellige faser gives der enkelte forelæsninger, der uddyber indholdet af disse faser.

Et bilkarrosseri med påmonteret motor bliver udleveret til hver enkelt projektgruppe. Se Figur 1.



Figur 1: Karrosseri med printplader, pendul og Arduino Mega2560

Den færdige bils formål er, at den skal kunne gennemkøre en konkurrencebane efter regler, der er nærmere beskrevet i bilag 1, ”Konkurrencen”.

Bilen skal udstyres med:

- Mikrocontroller.
- Elektronik til regulering af motor.
- Sensorer til detektering af refleksbrikker.
- For- og baglygter.
- Elektronik til styring af for- og baglygter.
- Elektronik og højttaler til afspilning af lyd.

Nogle af de nødvendige hardwareenheder vil helt eller delvis blive udviklet i andre kurser på 1. semester. Arbejdet med disse opfattes derfor ikke som en del af semesterprojektet. Det drejer sig om:

- Print med kredsløb til detektering af refleksbrikker. Dette print skal forbindes til bilens mikrocontroller. Hvis der foretages modifikationer af printet for at tilpasse det projektet, skal disse modifikationer beskrives i projektdokumentationen.
- ”Fejltæller”: Print med kredsløb til tælling og visning af pulser fra et pendul. Pendulet anvendes kun som et påmonteret måleudstyr under konkurrencen, og den er ikke en del af selve projektet.

Bilen skal forsynes med fremadrettet kørelys (hvidt) og baglys (rødt), samt bremseleys (kraftigt rødt). Se figur 2 og figur 3.

Bremseleys og baglys udsendes fra samme kilde og skal kunne reguleres i to forskellige styrker.

Hvert enkelt for-, bag- eller bremseleys kan realiseres som et antal lysdioder, der opbygges som et LED-sæt.

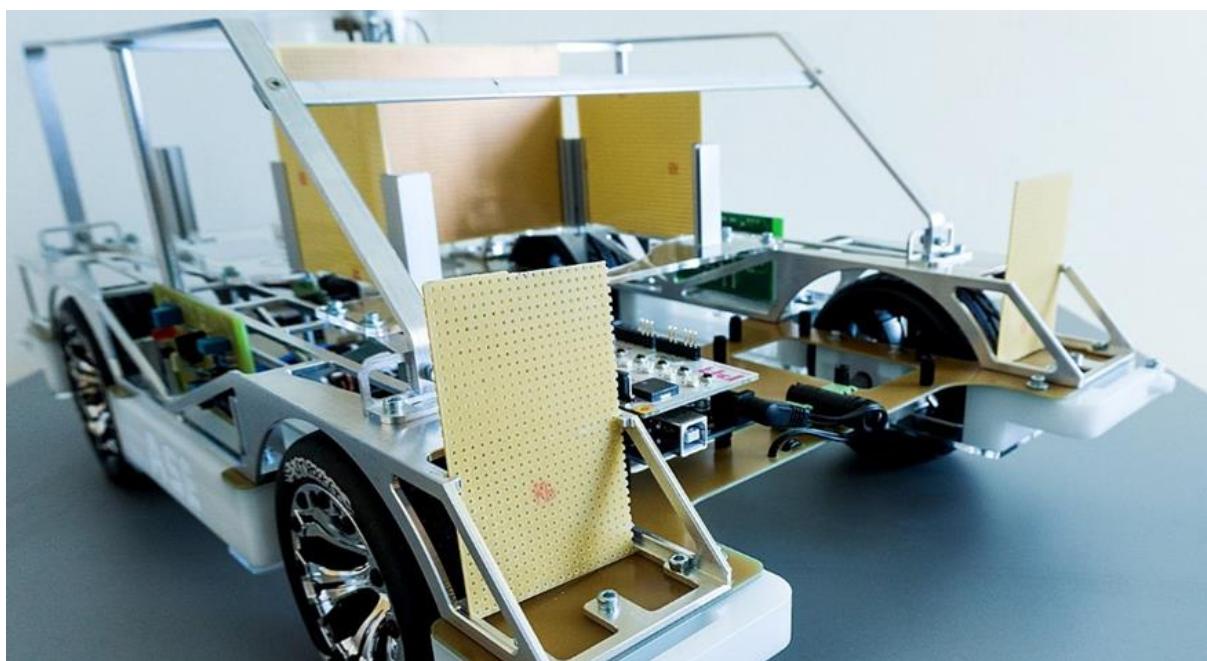
Hvert sæt kan opbygges som en kombination af serie- og/eller parallel-forbundne lysdioder.

I projektet anvendes mikrocontrolleren MEGA2560 på et "Arduino Mega2560" kit. Denne anvendes til:

- Styring af bilens fart og retning.
- Styring af for- og baglygternes lysstyrke.
- Detektering af banens refleksbrikker.
- Afspilning af lyde og/eller melodier.

Bilen skal under konkurrencen køre på en bane med to bander, hvor banderne vil sikre, at bilen bliver på banen. Banderne er udstyret med 7 par refleksbrikker, der kan detekteres af optiske sensorer placeret på bilen. Ved start skal bilen afspille en specifik startlyd eller startmelodi, og efter passage af mållinjen skal afspilles en specifik slutlyd eller slutmelodi. Ved passage af hver enkelt af refleksbrikkerne skal bilen afgive en specifik "refleksbriklyd".

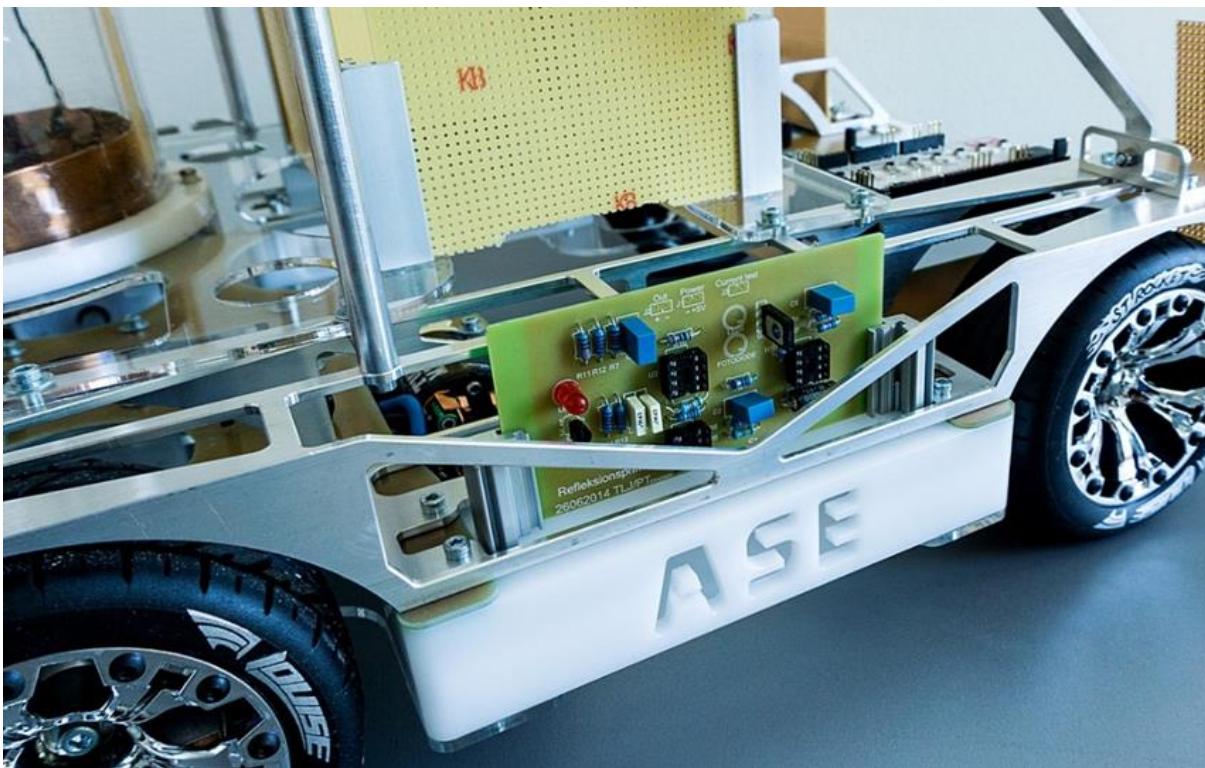
På figur 4 er vist placering af et af printene til detektering af refleksbrikker (der skal være et på hver side). Figur 5 viser en mulig placering af print med anden elektronik, som skal konstrueres for at kunne styre bilen. De præcise fysiske mål for bilkarrosseriet kan ses i bilag 1.



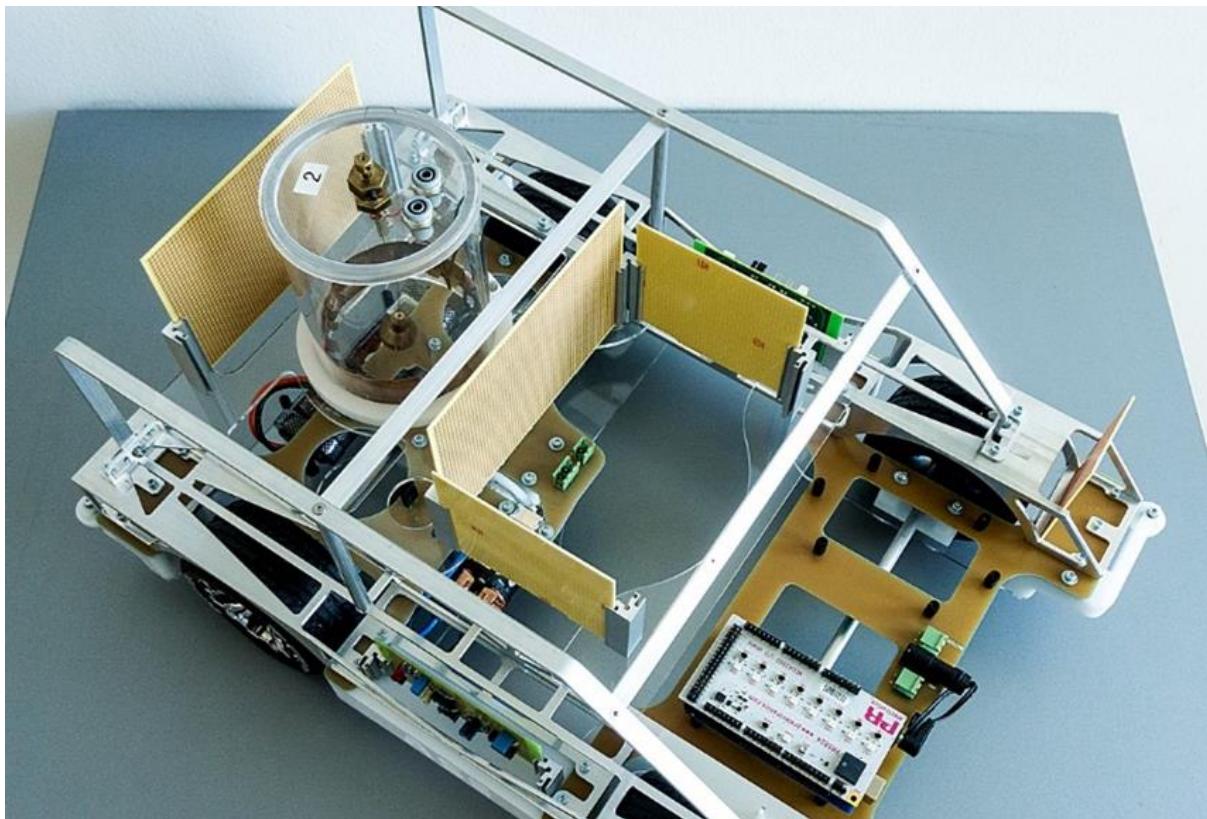
Figur 2: Placering af to print til forlys, højre og venstre side



Figur 3: Placering af print til bag- og bremselflys



Figur 4: Placering af print til detektering af refleksbrikker



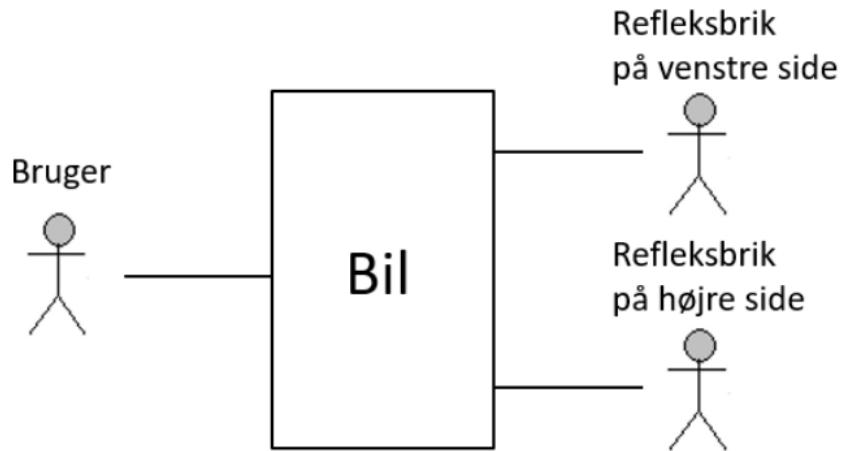
Figur 5: Print med øvrig elektronik placeret midt i bilen

Kravspecifikation

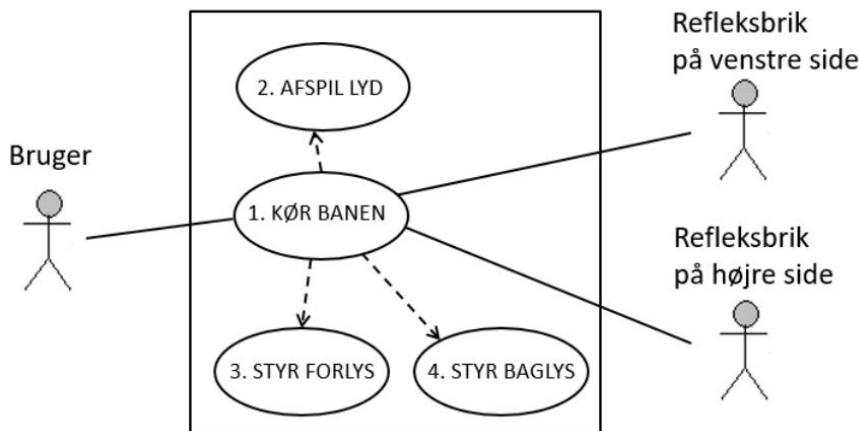
Det system, som skal udvikles, er selve den elektriske bil.

Figur 6 viser systemet ”Bil” med de aktører, som interagerer med den. Aktøren ”Bruger” kan få bilen til at starte et gennemløb af banen. Brugeren er en primær aktør.

Aktøren ”Refleksbrik på venstre side” kan detekteres af bilen, hver gang en sådan passerer. Tilsvarende gælder for aktøren ”Refleksbrik på højre side”. Refleksbrikkerne er sekundære aktører.



Figur 6: Aktør-kontekst diagram



Figur 7: Use case diagram

Funktionelle krav

Use case 1: "Kør banen"

Mål

Denne use case beskriver, hvordan banen skal gennemkøres.

Initieres af: Bruger.

Normalt scenarie

1. Brugeren tænder for bilen og placerer den, så den kører forlæns ind på banen ved at passere startlinjen.
2. Når bilen har passeret refleksbrik nummer 6, bringes bilen til standsning, inden refleksbrik nummer 7 nås.
3. Bilen bakker, indtil refleksbrik nummer 5 er passeret, og bringes til standsning, inden refleksbrik nummer 4 nås.
4. Bilen kører forlæns, indtil refleksbrik nummer 7 nås.
5. Bilens bringes til standsning i målområdet, der er defineret som området mellem banens slut-ende og en meter efter banens slut-ende.

Generelt gælder:

- Lyd afspilles som beskrevet i use case 2 "Afspil lyd".
- Forlyset styres som beskrevet i use case 3 "Styr forlys".
- Baglyset styres som beskrevet i use case 4 "Styr baglys".

Use case 2: "Afspil lyd"

Mål

Denne use case beskriver styring af en i systemet indbygget lydgiver.

Initieres af: Use case 1 "Kør banen".

Normalt scenarie

1. Når bilen tændes for at køre ind på banen, afspilles "trommeslageren"

2. Hver gang en refleksbrik passeres, afspilles ”Gas”.
3. Når refleksbrik nummer 7 passeres, afspilles ”trommeslageren”

Use case 3: ”Styr forlys”

Mål

Denne use case beskriver styringen af bilens indbyggede forlys. Forlyset kan være slukket eller tændt.

Initieres af: Use case 1 ”Kør banen”.

Normalt scenarie

Forlyset skal være tændt, når bilens motor påtrykkes en spænding, for at bringe den til at køre. Ellers skal forlyset være slukket.

Use case 4: ”Styr baglys”

Mål

Denne use case beskriver styringen af bilens indbyggede baglys. Baglyset kan være slukket, lyse med mellemstyrke (”almindeligt baglys”) eller lyse med kraftig styrke (”bremselys”).

Initieres af: Use case 1 ”Kør banen”.

Normalt scenarie

- Baglyset skal lyse med mellemstyrke (”almindeligt baglys”), når bilens motor påtrykkes en spænding, for at bringe den til at køre. Ellers skal baglyset være slukket.
- Baglyset skal lyse med kraftig styrke (”bremselys”), mens spændingen til bilens motor mindskes. Bremselyset skal herefter forblive tændt i 0,5 sekund (+/- 0,1 sekund).

Ikke-funktionelle krav

1. Generelle krav

- 1.1. Bilen skal på en enkelt opladning af dennes batterier kunne gennemføre mindst 5 gennemkørsler af banen.
- 1.2. På bilens højre og venstre side skal placeres detektorer, der kan registrere en R80 refleksbrik i afstanden 2 cm til 25 cm. Se figur 5 og figur 6.
- 1.3. Bilen monteret med al udstyr må maksimalt veje 5,1 kg.
- 1.4. Bilens maksimale højde skal være 41 cm.

2. Bilens forlys

- 2.1. Implementeres med 2 hvide LED-sæt af 4 hvide dioder, der monteres med et sæt i henholdsvis højre og venstre side på vandret led ved siden af hinanden.
- 2.2. Når forlyset er tændt, skal middelstrømmen gennem hvert LED-sæt være 50 mA +/- 5 mA.

3. Bilens bag- og bremselys

- 3.1. Implementeres med 2 LED-sæt af 5 røde dioder, der monteres med et sæt i henholdsvis højre og venstre side, og sidder i en X-opstilling.
- 3.2. Ved ”bremselys” skal middelstrømmen gennem hvert LED-sæt være 50 mA +/- 5 mA.

3.3. Ved ”almindeligt baglys” skal middelstrømmen gennem hvert LED-sæt være 10 mA +/- 2 mA.

4. Bilens lyd

4.1. Startlyd / startmelodi: Der bliver afspillet ”Trommeslageren.mp3” - 4 sekunder.

4.2. Refleksbriklyd: Der bliver afspillet ”Gas.mp3” - 2 sekunder.

4.3. Slutlyd / slutmelodi: Der bliver afspillet ”Trommeslageren.mp3” 4 sekunder.

Acceptspecifikation

Accepttestsspecifikation for funktionelle krav (use cases)

Use case 1 Kør banen	Test	Forventet resultat	Opnået resultat	Godkendt/Kom- mentar
Punkt 1 + Punkt 2	<p>Brugeren tænder for bilen ved hjælp af tændknap, og placerer den, så den kører forlæns ind på banen ved at passere startlinjen.</p> <p>Når bilen har passeret refleksbrik nummer 6, bringes bilen til standsning, inden refleksbrik nummer 7 nås.</p>	Visuel test: Bilen standser mellem refleksbrikkerne 6 og 7.		
Punkt 3	Bilen bakker, indtil refleksbrik nummer 5 er passeret, og bringes til standsning inden refleksbrik nummer 4 nås.	Visuel test: Bilen bakker og standser mellem refleksbrikkerne 4 og 5.		
Punkt 4	Bilen kører forlæns, indtil refleksbrik nummer 7 nås.	Visuel test: Bilen kører fra mellem refleksbrikkerne 4 og 5 og standser ved refleksbrik 7.		
Punkt 5	Bilens bringes til standsning i målområdet, der er defineret som området mellem banens slut ende, hvor sidste refleksbrik også	Visuel test: Bilen kører fra refleksbrik 7, og standser mellem banens slut ende, hvor sidste refleksbrik også		

	meter efter banens slutende.	befinder sig, og en meter efter banens Slutende.		
--	------------------------------	--	--	--

Tabel 2: Use case 1

Use case 2 Afspil lyd	Test	Forventet resultat	Opnået resultat	Godkendt/kommentar
Punkt 1	Use Case 1 ("Kør banen") udføres.	Akustisk test: Når bilen tændes for at køre ind på banen: Der afspilles 4 sekunders trommeslageren, og gruppemedlemmerne lytter for at bekræfte.		
Punkt 2	Hver gang en refleksbrik passereres, afspilles en lydeffekt, som er en bil, der gasser op.	Akustisk og visuel test: Bilen placeres før en hvilken som helst refleksbrik og tændes for at forbi passere den. Når den forbi passeres, så skal en lydeffekt afspilles, som er en bil, der gasser op.		
Punkt 3	Når refleksbrik nummer 7 passerer, afspilles en lyd, hvor en 4 sekunders trommeslageren spilles.	Akustisk test: Vi får bilen til at passerer den 7. refleks og gruppemedlemmerne hører om, der bliver afspillet en lyd, som er en 4 sekunders trommeslageren.		

Tabel 3: Use case 2

Use case 3 Styr forlys	Test	Forventet resultat	Opnået resultat	Godkendt/kommentar
Punkt 1	Forlyset skal være tændt, når bilens motor påtrykkes en spænding, for at bringe den til at køre fremad. Når bilen bakker, skal forlyset også være tændt. Når	Visuel test: Vi tænder for bilen, og får den til at køre frem, hvilket betyder, at der påtrykkes en spænding på motoren. Derefter ser		

	bilen ikke køre, så skal forlyset være slukket.	vi, om forlygterne tænder.		
--	---	----------------------------	--	--

Tabel 4: Use case 3

Use case 4 Styr baglys	Test	Forventet resultat	Opnået resultat	Godkendt/kom- mentar
Punkt 1	Baglyset skal lyse med mellemstyrke ("almindeligt baglys"), når bilens motor påtrykkes en spænding, for at bringe den til at køre. Ellers skal baglyset være slukket.	Visuel test: Vi placerer bilen i hvilken som helst position, og tænder den, for at få den til at bevæge sig/køre. Når den bevæger sig/kører, så kigges der på, om baglyset lyser med mellemstyrke. Når bilen holder op med at køre, så kigges der på, om baglyset slukker, da der ikke længere vil være spænding på bilens motor.		
Punkt 2	Baglyset skal lyse med kraftig styrke ("bremselys"), mens spændingen til bilens motor mindskes. Bremselflyset skal herefter forblive tændt i 0,5 sekund +/- 0,1 sekund.	Visuel test: Bilen placeres mellem 2 refleksbrikker, hvor hældningen er flad. Derefter tændes bilen, og der observeres, om baglysende lyser kraftigere end, der bilen startede, proportionalt med at hastigheden på bilen falder. Når lyset begynder at lyse kraftigt, startes et stopur på en digital enhed, og der måles fra, at lyset begynder at lyse kraftigere til, at styrken igen mindskes, hvilket skal være 0,5 +/- 0,1 sekunder.		

Tabel 5: Use case 4

Accepttestsspecifikation for ikke-funktionelle krav

Krav nr.	Krav	Test	Forventet resultat	Opnået resultat	Godkendt/Kommentar
1.1	Bilen skal på en enkelt opladning at dens batterier kunne gennemføre mindst 5 gennemkørsler af banen.	Vi kører lader bilen fuldfører bane, hvorefter gruppen placerer den ved start imens, at vi tæller hvor mange gange den kører frem og tilbage. Inden gruppen sætter den til at køre skal den have fuldt opladt batteri.	Vi forventer at vores bil vil kunne komme til at køre frem og tilbage på den bane, som er bygget minimum 5 gange.		
1.2	På bilens højre og venstre side skal placeres detektorer, der kan registrere en R80 refleksbrik afstanden 2 cm til 25 cm.	Vi får bilen til at køre forbi hvilke som helst R80-refleksbrikker på både højre og venstre side i forskellige afstande mellem 1 26cm for at se om de bliver registreret af bilens detektorer.	Vi forventer at detektorerne på hver side af bilen registrerer de forbipasserende R80-refleksbrikker, når bilen bevæger sig forbi dem.		
1.3	Bilen monteret med al udstyr må maksimalt veje 5,2 kg	Bilen monteret med al udstyr vejes på en vægt med en målenøjagtighed bedre end 100 gram.	Vægten viser en vægt mindre end eller lig med 5,1 kg.		
1.4	Bilens maksimale højde skal være 41 cm.	Vi mäter bilen fra dens laveste punkt, op til bilens højeste med en tommestok. Tommestokken skal have en målenøjagtighed på 5 mm	Det forventes at vi mäter en højde som er mindre end eller lig med 41 cm.		

2.1	Forlyse implementeres med 3 hvide LED-sæt, der monteres med et sæt i henholdsvis højre og venstre side på vandret led ved siden af hinanden.	Visuel: Testet om der sidder 3 hvide LED sæt på vandret led på hhv. højre og venstre side af bilen.	Bilen opfylder kravet om LED-monteringen.		
2.2	Når forlyset er tændt, skal middelstrømmen gennem hvert LED sæt være 50 mA +/- 5 mA.	Vi sætter et multimeter i serie med LED-sæt-kredslobet, og mäter herefter, om strømmen er 50 mA +/- 5mA.	LED-sættet opfylde kravet om en middelstrøm på 50 mA.		
3.1	Bag- og bremselflys implementeres med 2 LED-sæt af 5 røde dioder, der monteres med et sæt i henholdsvis højre og venstre side, og sidder i en X-opstilling.	Visuel: Tester om der sidder 2 LED-sæt med 5 røde dioder i en X-opstilling på hhv. højre og venstre side af bilen.	Bilen opfylder kravet om LED-monteringen på bagsiden af bilen.		
3.2	Ved ”bremselflys” skal middelstrømmen gennem hvert LED sæt være 50 mA +/- 5 mA.	Vi sætter properne, fra et multimeter med en nøjagtighed på +/- 0,8%, på anoden og katoden for hvert LED-sæt, og mäter herefter, om strømmen er 50 mA.	Bilen opfylder kravet om strømstyrke.		
3.3	Ved ”almindeligt baglys” skal middelstrømmen gennem hvert LED sæt være 10 mA +/- 2 mA.	Ved hjælp af multimeter med en nøjagtighed på +/- 0,8%, måles middelstrømmen gennem hver LED-sæt til at være 10mA.	Multimeteret med en nøjagtighed på +/- 0,8% viser en værdi på 10mA.		
4.1	Startlyd / start melodi	Når bilen tændes/bilens kode bliver kørt, så starter den med at afspille en 4 sekunders	Når der tændes for bilen/bilen starter, afspilles en 4 sekunders trommeslageren, og		

		trommeslageren, Når denne trommeslageren er færdig, begynder bilens hjul at køre i om-drejninger.	derefter begynder bilen at køre.		
4.2	Refleksbrik Lyd	Når bilen kører forbi de forskellige R80-reflekser i løbet af banen, så afspilles en lyd af en bil, som gasser op (gas). Denne lyd afspilles hver gang bilen forbipasserer reflekserne ved hjælp af bilens monteret detektører.	Når bilen kører forbi de forskellige reflekser på banen, så afspilles en lyd af en bil, som gasser op, som gruppen har valgt, og dette udføres ved alle R80- reflekserne igennem banen, på nær ved slutningen, hvor en slutlyd, som siger “What are you doing stepbro?” afspilles i stedet.		
4.3	Slutlyd / slutmelodi	Når bilen når målstregen, og kommer til den sidste R80-refleks på banen, så afspilles en slutlyd, som er en 4 sekunders trommeslageren. Dette gøres ved at bilen ved, hvor mange reflekser der er kommet forbi forinden, og derefter skifter over til en slags ”slut-løkke,” som udfører noget kode KUN ved slutningen, som er med til at afspille den lyd.	Når bilen når målstregen, samt den sidste R80-refleks på banen, så afspilles en slutlyd, som er en 4 sekunders trommeslageren, ved hjælp af et stykke kode, som KUN udføres ved dette stadie af banen.		
5.1	Kør op ad en bakke på 1 meter med	Bilen placeres ved bunden af en bakke	Bilen starter ved bunden ad bakken		

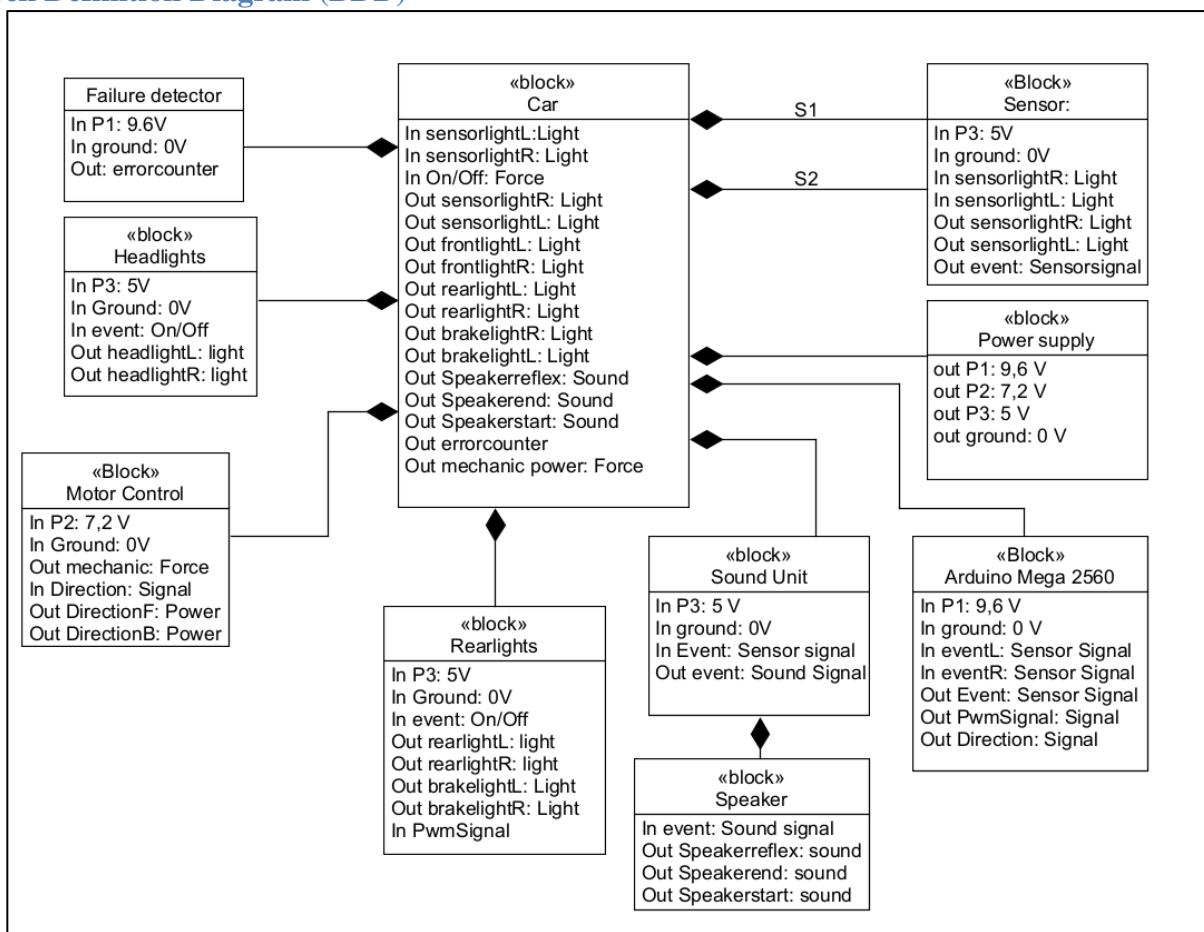
	hældning 30 grader.	på 1 meter, som har en hældning på 30 grader. Derefter startes bilen, og der observeres om bilen har nok spænding til at kunne køre op ad bakken og nå toppen.	på 1 meter med en hældning på 30 grader, og kører derefter op ad bakken indtil at toppen nås.		
--	---------------------	--	---	--	--

Tabel 6: Accepttestspezifikation for ikke-funktionelle krav

Systemarkitektur

Hardwarearkitektur

Block Definition Diagram (BDD)



Figur 8: Block Definition Diagram (Tegnet af Anders)

Blokbeskrivelse af BDD (Mads)

Blok	Funktionalitet
Car	Alle komponenter er bygget på denne.

Headlights	2 LED-sæt af 4 hvide LED-pærer, som sidder på vandret led ved siden af hinanden foran på både højre og venstre side af bilen, og lyser.
Rear Lights	2 LED-sæt af 5 røde LED-pærer, som sidder i et kryds-mønster ved siden af hinanden bagved på både højre og venstre side af bilen og lyser.
Sensor	Detekterer lys, som reflekteres fra refleksbrikkerne på banen, som bilen skal igennem.
Arduino Mega 2560	Processerer input fra Sensor.
Failure Detector	Detekterer fejl.
Speaker	Afspiller lyde og melodier: Startmelodi, Slutlyd og Reflekslyde.
Power supply	Forsyner alt elektronik på bilen med strøm. Bl.a. Arduino Mega 2560, Sensor, Headlights, m.m.

Tabel 7: Blokbeskrivelse af BDD

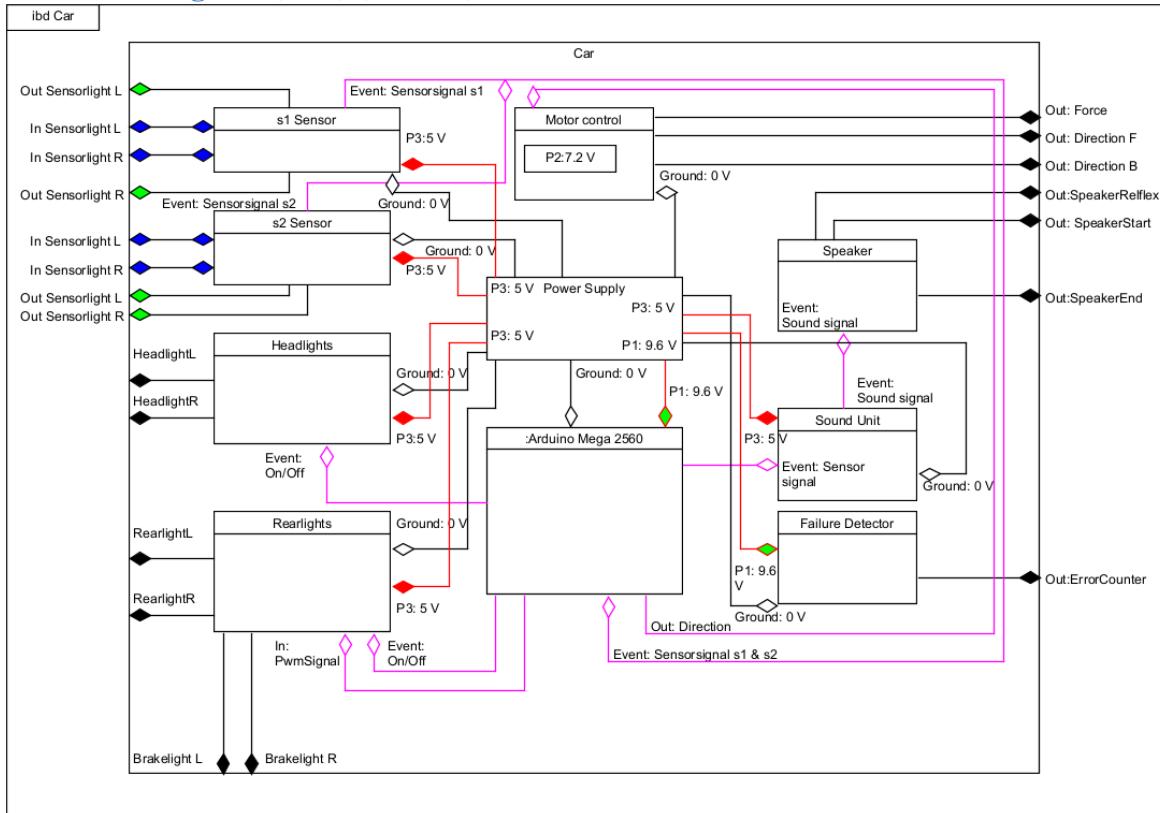
Signal-Navn	Funktion	Om-råde	Port 1 (source)	Port 2 (Destination)	Kommentar
Ground	Reference til analog spænding	0,0V	Power Supply, ground	Sensor 1, Sensor 2, Arduino Mega2560, Motor, Headlights, Rearlights, Failure detector, Sound unit,	Stel
P1:	Forsyningsspænding	9,6 V	Power Supply, 9,6 V	Failure detector, Arduino Mega 2560,	
P2:	Forsyningsspænding	7,2 V	Power Supply, 7,2 V	Motor Control,	
P3:	Forsyningsspænding	5,0 V	Power Supply, 5,0 V	Headlights, Rearlights, Soundunit. Sensor 1, Sensor 2,	Kommer fra vores Spænding regulator.
OutBrake-Light	Fysisk lys		Rearlight		Lys ud af Bremselygter
OutRear-Lights	Fysisk lys		Rearlight		Lys ud af Bremselygter

Pwm Signal	Styre bremselflys	5,0 V	Arduino Mega 2560	RearLights,	Styrer vores bremselflys
OutHead-lights	Fysisk lys		Headlight		Skal lyse ved InEvent: On
Error-Counter	Skal detektere når pendulet slår cylinder		Failure Detector		Skal tælle fejl

Tabel 8: Tabel over signaler (Anders)

Softwarearkitektur

Internal Block Diagram (IBD) (Anders)



Figur 9: Internal Block Diagram (IBD)

Signalnavn	Funktion	Om-råde	Port 1 (source)	Port 2 (destination)	Kommentar
Ground	Reference til analog spænding	0,0V	Power Supply, ground	Rearlights, Sound unit, Speaker, Failure detector, Arduino Mega 2560, Headlights, Sensors, Motorcontrol.	
P1	Forsyningsspænding til flere komponenter	5.0V	Power Supply	Rearlights, Headlights, Sound unit, Sensors,	
P2	Forsyningsspænding til vores motor	7.2V	Power Supply(Batteri)	Motor Control	

P3	Forsyningsspænding til flere komponenter	9.6V	Power Supply(Batteri)	Arduino Mega 2560, Failure Detector	Der er senere lavet ændringer, så Failure detector tager 5V, Pga en ombygning af den.
Event On/Off	Bruges til at styre om lyset er tændt eller slukket	0.0-5.0V	Arduino Mega 2560	Rearlights, Headlights	
Event Sound Signal	Signal kommer fra vores Somo enhed og sender det ud til vores højtalere		Somo Sound Unit	Speaker	
In PwnSignal	Bruges til at styre lysstyrken af vores baglys.	5.0V	Arduino Mega 2560	Rearlights	
Out: Direction	Bruges til at styre den retning vores motor kører	5.0V	Arduino Mega 2560	Motor Control	
Event Sensors signal s1/s2	Bruges til at kører bilen igennem banen. Samt styring af motor, lys og lyd.	5.0V	Sensor s1/s2	Arduino Mega 2560	
Out: ErrorCounter	Tæller antal fejl	5.0V	Failure Detector	7-Segments Display	
Out Sensorlight (L/R)	Vores Sensor sender et lys ud som kan blive reflekteret af refleksbrikkerne.		Sensor s1/s2	"Refleksbrik"	
In Sensor light(L/R)	Det lys som bliver reflekteret tilbage til refleksion detektoren		"Refleksbrik"	Sensor s1/s2	
Brakelight(L/R)	Vores baglys lyser mere når bilen den bremser	5.0V	Baglys		
RearLight(L/R)	Vores baglys lyser med en konstant styrke som placeringsslys.	5.0V	Baglys		
Headlight(L/R)	Vores forlys lyser med en konstant styrke som placeringlys.		Forlys		
Out: Speaker/End/Start/Reflex	Somo enheden, afspiller lyd ved start/reflexbrik og slut.		Sound unit	Speaker	
Out: Direction (F/B)	Vores bils retning bliver styret af et Pwn signal samt en H-bro.	5.0V	Motor Control	Bil	

Tabel 9: Signalbeskrivelse for IBD-forbindelserne

Forlys (Nahom)

Forlys:

+ initForlys() : void

Anvar:

Modulets/klassens ansvar er at kontrollere bilens forlys, så det tænder, efter vi har tændt for bilen.

Funktion:

void initForlys();

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Initialiserer softwaren for forlyset ift. PWM, valg af output ben og clock prescaler.

Baglys (Mads)

Baglys:

+ initBaglys() : void

+ styrkebaglys(double lysstyrke) : void

Anvar:

Modulets/Klassens ansvar er at kontrollere bilens baglys, så den kan lyse med forskellige lysstyrker ud fra, om bilen kører fremad, bakker eller er stoppet.

Funktioner:

void initBaglys();

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Initialiserer softwaren for baglyset ift. PWM, valg af output ben og clock prescaler.

void styrkebaglys(double lysstyrke);

Parametre: Den ønskede lysstyrke (0-100), som er den procentvise angivelse af lysets maximal styrke.

Returværdi: Ingen

Beskrivelse: Omsætter parameteren lysstyrke til en værdi, som er med til at bestemme baglysets lysstyrke.

Sensor/Interrupts (Mads)

Interrupts:

+ initINTE() : void

Ansvar:

Modulet/Klassens ansvar er at få interrupts aktiveret for bilen, hvilket gør det muligt at indsætte forskellige events i main.c-filen ud fra forskellige sensordetektioner på banen, som bilen skal køre igennem.

Funktioner:

void initINTE()

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Tænder globale interrupts ved hjælp af sei();, tænder ekstern interrupts for INT0 og INT1 og sætter INT0 og INT1 til RISING EDGE INTERRUPT.

Motor (Anders og David)

Motor control:

```
+speed (in speed:double) void  
+directionchange(double) void  
+stop ()void
```

Ansvar:

Modulets/klassens ansvar er at kontrollere bilens motor, så den kan køre fremad eller baglæns med variabel hastighed. Modulets/klassens ansvar er også at standse bilens motor.

Retningen styres af global variable, der tæller hvilken refleksbrik den er nået til.

```
void speed_ (double)
```

Parametre: Den ønskede speed (0-100), som er den procentuelle angivelse af motorens maksimale ydelse

Returværdi: Ingen

Beskrivelse: Omsætter parameteren speed til en PWM duty factor, som styrer motoren.

```
void directionchange (double)
```

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Indholder en ”if”-sætning der styrer om relæet er hhv. tændt eller slukket ved at sætte en vilkårlig pin på ArduinoMega2560 til et logisk HIGH eller LOW.

```
void stop()
```

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Får motoren, og dermed bilen, til at standse, ved at sætte ORC3A=0

```
void inittimer()
```

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Skal bruges til at vælge hvilken timer, samt mode den skal være i. Samtidig også bruges til at sætte de nødvendige pins på Mega2560'eren til outputs.

Lyd (August)

UART:

```
+initUART(unsigned long baudRate, unsigned char DataBit) void
+SendChar(char Tegn) void
```

Ansvar:

At kunne initialisere UART med rigtig BAUDrate og Databit. Herved også sende en char over UART.

Void initUart(unsigned long baudRate, unsigned char DataBit)

Parametre: indstilling af ønsket baudrate og antallet af databit's

Returværdi: Ingen

Beskrivelse: Initialiserer MEGA2560's UART til at kunne sende og modtage tegn.

void SendChar(char Tegn)

Parametre: char'en der ønskes sendt

Returværdi: Ingen

Beskrivelse: Venter på at MEGA2560 er klar til at sende tegn, sender derefter ønsket tegn over UART.

SOMO-II

```
+SOMOplay() void
+SOMOpause() void
+SOMOvolumeIN(int level) void
+SOMOvolumeDEC(int level) void
+SOMOstop() void
+SOMOtrack(int track)
```

Ansvar:

Skal kunne spille, vælge track, pause, stoppe, skrue op, skrue ned og stoppe afspilningen af musik via UART.

Void SOMOplay()

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Starter afspilning af lyd

void SOMOpause()

Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Pause den musik der netop spiller

void SOMOvolumeIN(int level)

Parametre: Antallet af niveauer som lyden ønskes hævet

Returværdi: Ingen

Beskrivelse: Hæver lyden på SOMO-II til ønsket niveau

void SOMOvolumeDEC(int level)

Parametre: Antallet af niveauer som lyden ønskes nedsat

Returværdi: Ingen

Beskrivelse: Nedsætter lyden på SOMO-II til ønsket niveau

Void SOMOstop()



Parametre: Ingen

Returværdi: Ingen

Beskrivelse: Stopper helt afspilning af lyd

Void SOMOtrack(int track)

Parametre: den ønskede track på SD-kortet

Returværdi: Ingen

Beskrivelse: Starter afspilning af den ønskede nummereret MP3 fil på SD-kortet.

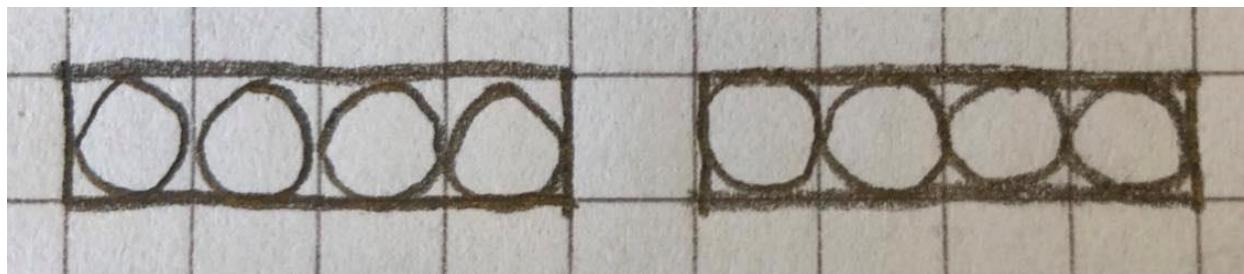
Hardwaredesign

Hardware for lys (Nahom og Mads)

Bilen i 1. semesterprojekt skal udstyres med både frontlys og baglys. Både frontlys og baglys skal lyse, når bilen startes, og baglyset skal kunne regulere ud fra om bilen bremser eller bakker. For at kunne gøre dette skal der gøres brug af PWM-signaler, som sendes via Arduinoen. Bilen får monteret 2 forlygter og baglygter, som henholdsvis er placeret i både højre og venstre side, foran og bagpå bilen. Dette gøres ved hjælp af hvide LED-sæt til fronten, og røde LED-sæt til bagenden.

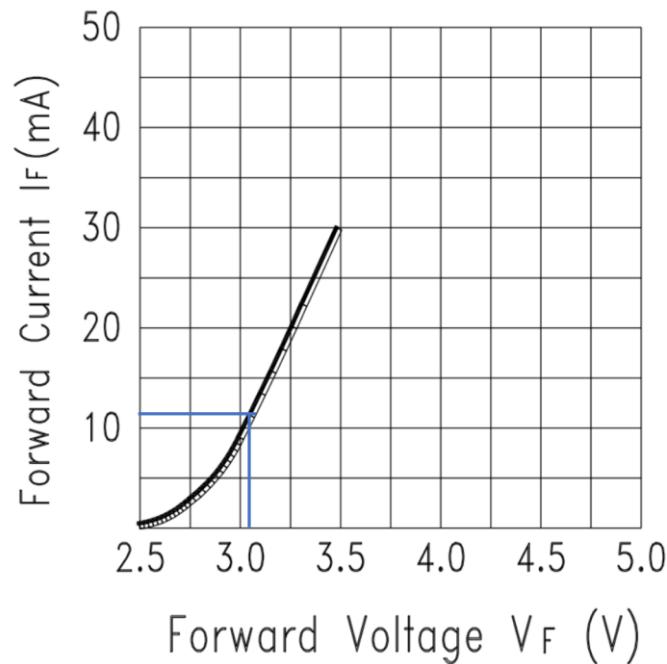
Forlys (Nahom)

Til forlyset anvendes LTW-2S3D8(Only, n.d.). Bilens forlygter består af 2 gange 4 dioder:



Figur 10: Skitse af opstilling af LED-sæt for forlys

Forlysets LED-sæts middelstrøm skal være på $50\text{mA} \pm 5\text{mA}$. Derfor skal vi have en middelstrøm på $12,5\text{ mA}$ i hver lysdiode, eftersom $50\text{mA}/4 = 12,5\text{mA}$. Vi gør brug af databladet for en LTW-2S3D8 LED, til at finde spændingsfaldet over LED'en.



Figur 11: LTW-2S3D8 spændingsfald diagram

Vi kan aflæse spændingsfaldet for vores LED til at være ca. 3,10V. Sammenlagt med vores forsyningsspænding på 5V, kan vi nu bruge dette til at finde spændingsfaldet over modstanden:

$$V_{forsyning} = 5V$$

$$V_{LED-spændningsfald} = 3,10V$$

$$I_{LED-sæt} = 50mA$$

$$V_{modstand-spændningsfald} = 5V - 3,1V = 1,9V$$

Ligning 1: Bestemmelse af spændingsfaldet over modstanden

Formodstand er dermed:

$$R_{LED} = \frac{V_{modstand-spændningsfald}}{\frac{I_{LED-sæt}}{4}}$$

Ligning 2: Beregning af formodstanden

$$R_{LED} = \frac{1,9V}{12,5mA} = 152 \Omega$$

Ligning 3: Beregning af formodstanden

Formodstanden til lysdioderne for forlys er beregnet til at være på 152Ω til hver diode. Men da der ikke er en modstand på 152Ω tilgængelig i værkstedet, vælger vi i stedet at anvende en formodstand på 150Ω . Årsagen til, at der bruges en modstand på 150Ω er, at større strøm kan justeres ned til den ønskede værdi ved hjælp af software. Kredsløbet kan nu opbygges på multisim.

Multisim diagram for opstilling af forlys

Til at starte med opbygger vi et kredsløb i multisim, hvor vi vil lave de forbindelser som senere skal laves i den virkelige verden. Vi sætter formodstande på alle LED lys. Disse LED lys er alle forbundet til ben 2 (collector) på BD139. Ben 1 (emitter) forbindes til Ground, mens ben 3 (base) er forbundet til vores arduino. Fra tidligere opgave ved vi dette om BD139 transistoren¹:

$$\beta = 20$$

$$V_{CE} = 0,2V$$

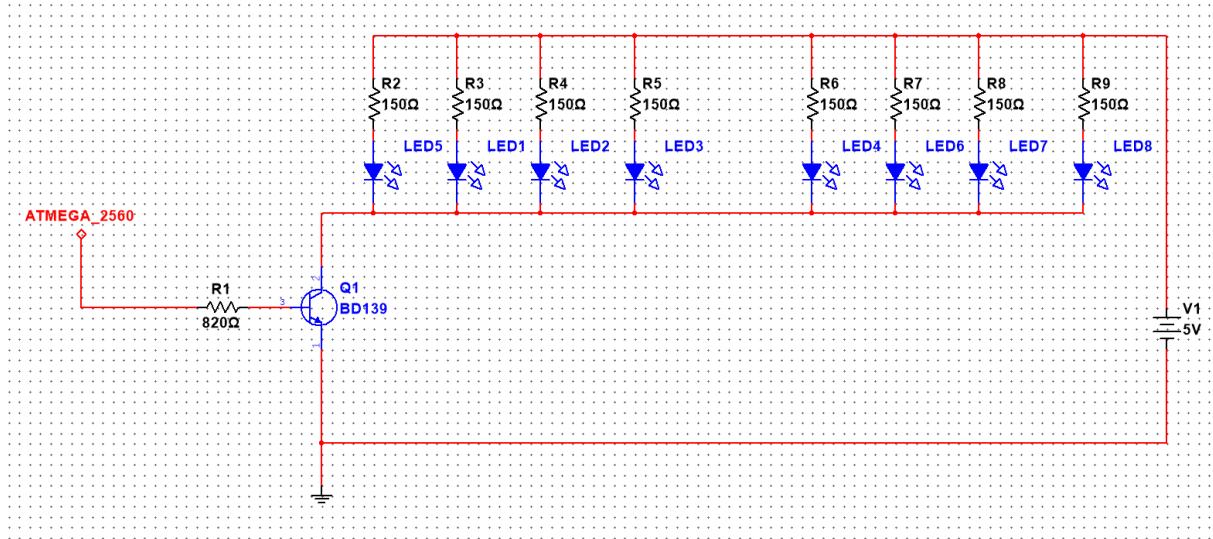
$$V_{BE} = 0,7V$$

¹ Bilag 7: Lysdiode

I den nævnte opgave har vi også regnet på modstanden der skal forbindes til resistoren BD139. Dette gøres på følgende formel:

$$-5 + \frac{0,3}{20} * R_1 + 0,7$$

Vi opbygger dermed følgende kredsløb i Multisim som ses på Figur 12:

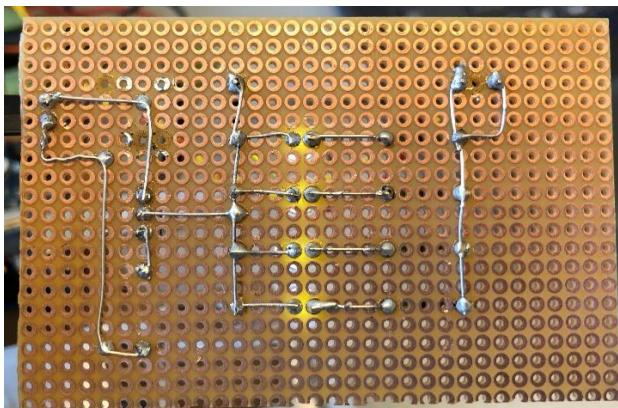


Figur 12: Multisim simulering

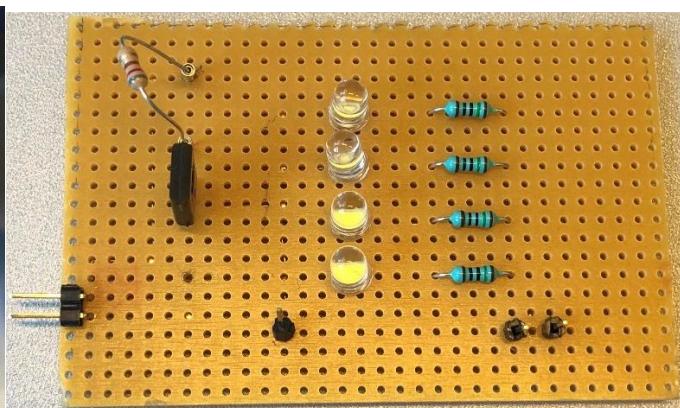
Vi har gjort brug af 8 LED'er, samt 8 formodstande på 150Ω på hver af modstandene. Derudover er der blevet gjort brug af en BD139 transistor. Forsyningsspændingen på 5V kommer ned igennem modstandene og hen til LED'erne. LED'erne lyser først når transistoren får nok spændingsforskæl mellem emitter og collecter. Arduino'en giver et firkantsignal mellem 0 og 5 V. Kredsløbet opstilles hermed på et fumlebræt, hvorefter der testes om det lever på til kravene.

Fysisk konstruktion på veroboard

Efter at have opstillet kredsløbet i Multisim kan det herefter viderekonstrueres på et fumlebræt for at kunne teste om det reelt fungerer. Eftersom vi kan vurdere kredsløbet til at fungere korrekt, bygger vi det igen på et veroboard. Dette veroboard vil være den endelige konstruktion for forlyset, hvilket efterfølgende vil sættes på bilen. Det er de præcist samme forbindelse som der blev lavet på fumlebrættet. I denne omgang har vi dog valgt at lodde forbindelserne sammen. På den måde kan vi være sikre på, at vi ikke har nogle løse forbindelser. Forsiden og bagsiden af det færdiglavede veroboard kan ses på figurerne nedenfor:



Figur 13: Bagseite af konstrueret veroboard

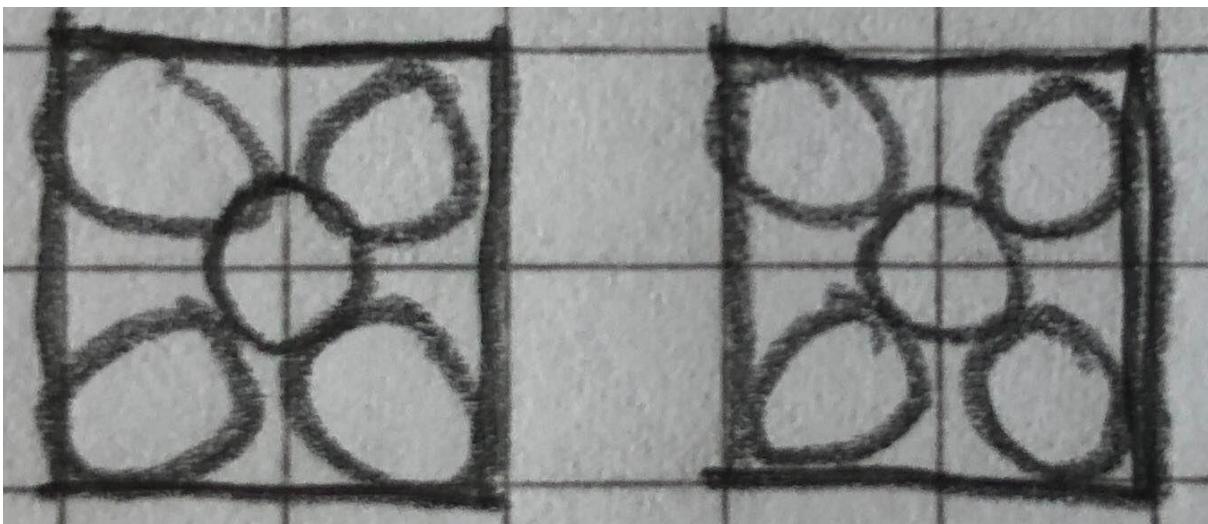


Figur 14: Forside af konstrueret veroboard

Opgaven har sat krav for middelstrømmen til at være på 50 mA +/- 5 mA per LED-sæt. For at sikre dette er tilfældet har vi, ved hjælp af et multimeter målt middelstrømmen ved at sætte en probe på ledningen for VCC og en anden probe på en hardwin pin. På den måde bliver amperemeteret en del af forbindelsen. Vi får målt amperen til at være på 55 mA, hvilket vi godtager eftersom vi havde sat en grænse på +/- 5 mA.

Baglys (Mads)

Til baglys anvendes røde dioder af modelnummer FYL-5013HD1C. Bilens baglys består af 2 LED-sæt af 5 røde dioder, opsat i en X-opstilling, som set på nedenstående figur:



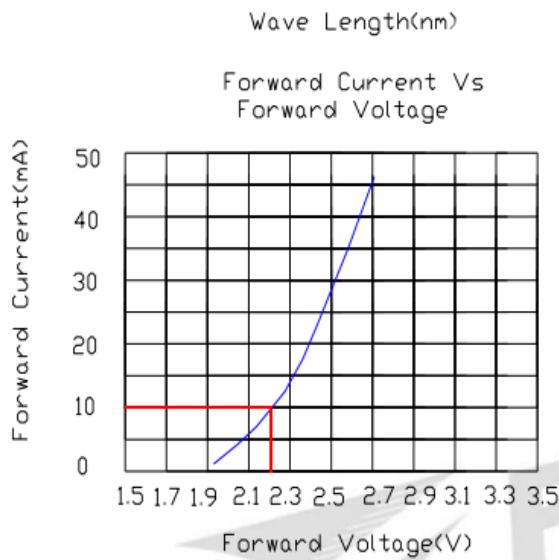
Figur 15: Skitse af LED-Sættenes opstilling

Bestemmelse af formodstand

Til baglyset skal middelstrømmen gennem hvert LED-sæt være 50 mA +/- 5 mA, når bilen bremser, og ved bilens almindelige baglys skal middelstrømmen gennem hvert LED-sæt være 10 mA +/- 2 mA. Da gruppen ved at bremselflysets middelstrøm har den største værdi af de to på 50 Ma +/- 5 mA, tager gruppen udgangspunkt i dette, da man senere kan regulere strømmen ned vha. PWM-signaler.

Bremselflysets LED-sæt skal have en middelstrøm på 50 mA ligesom frontflyset på bilen, hvilket betyder, at hver lysdiode ud af vores i alt 5 i hver LED-sæt skal have en strøm på $50\text{mA} / 5 = 10\text{mA}$. For at bestemme

modstanden bestemmer gruppen værdien af spændingsfaldet over dioderne ved hjælp af databladet for FYL-5013HD1C:



Figur 16: Datablad for FYL-5013HD1C²

Ifølge databladet for den røde diode af modelnummeret FYL-5013HD1C, skal spændingsfaldet over LED'en være ca. 2,2V ved 10 mA. Forsyningsspændingen er på 5V, og derfor bliver spændingsfaldet over modstanden:

$$V_{forsyning} := 5 \text{ V}$$

$$V_{LEDfald} := 2.2 \text{ V}$$

$$I_{LEDsæt} := 50 \text{ mA}$$

$$V_{MODSTANDfald} := 5 \text{ V} - 2.2 \text{ V} = 2.8 \text{ V}$$

Ligning 4: Bestemmelse af spændingsfaldet over modstanden

Formodstanden kan derfor bestemmes på følgende vis:

$$R_{LED} := \frac{V_{MODSTANDfald}}{\frac{I_{LEDsæt}}{5}} = \frac{2.8 \text{ V}}{\frac{50 \text{ mA}}{5}} = 280 \Omega$$

Ligning 5: Beregning af formodstanden

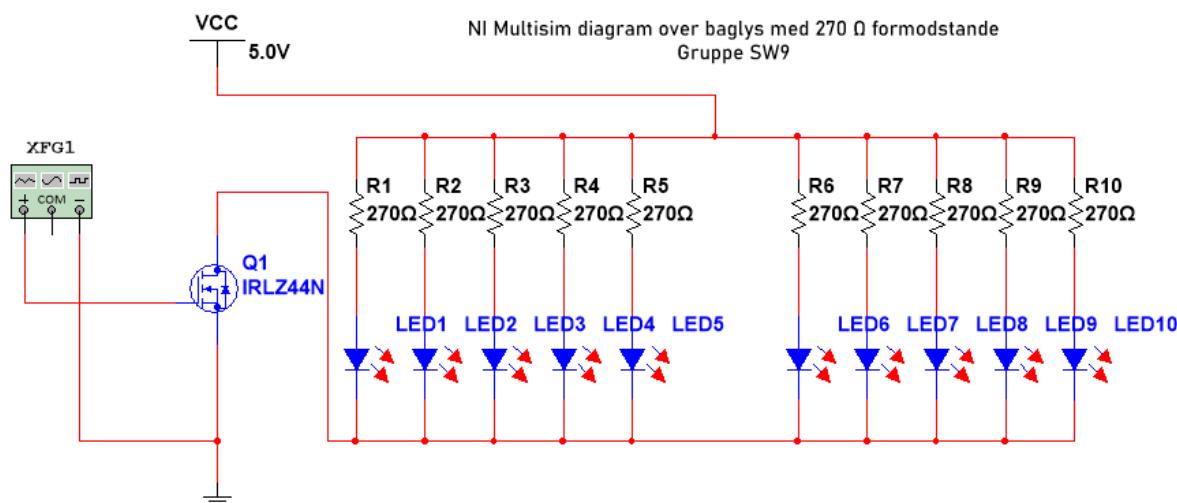
² (FORYARD)

Formodstanden til lysdioderne for bilens baglys er hermed beregnet til at være på 280Ω , så der anvendes en modstand fra værkstedet på 270Ω . Grunden til at der bruges en mindre modstand, er for at få nok strøm igenem. Kredsløbet opstilles nu i programmet NI Multisim.

NI Multisim diagram for opstilling

Til opstillingen af kredsløbet i NI Multisim for projektgruppens baglys benyttes 10 LED-komponenter, 10 formodstande på 270Ω og en MOSFET IRLZ44n transistor som har tre ben: Gate, Drain og Source. Grunden til at transistoren MOSFET IRLZ44n anvendes er, at der skal sendes signaler til kredsløbet via projektgruppens Arduino. Signalet sendes ind via Gate, Drain forbinderes til de røde lysdioder og Source går ned til Ground.

Kredsløbet for hele baglyset opstilles derfor sådan, at der anvendes en transistor der deler begge LED-sæt med formodstandene. Opbygningen kan ses på nedenstående figur:

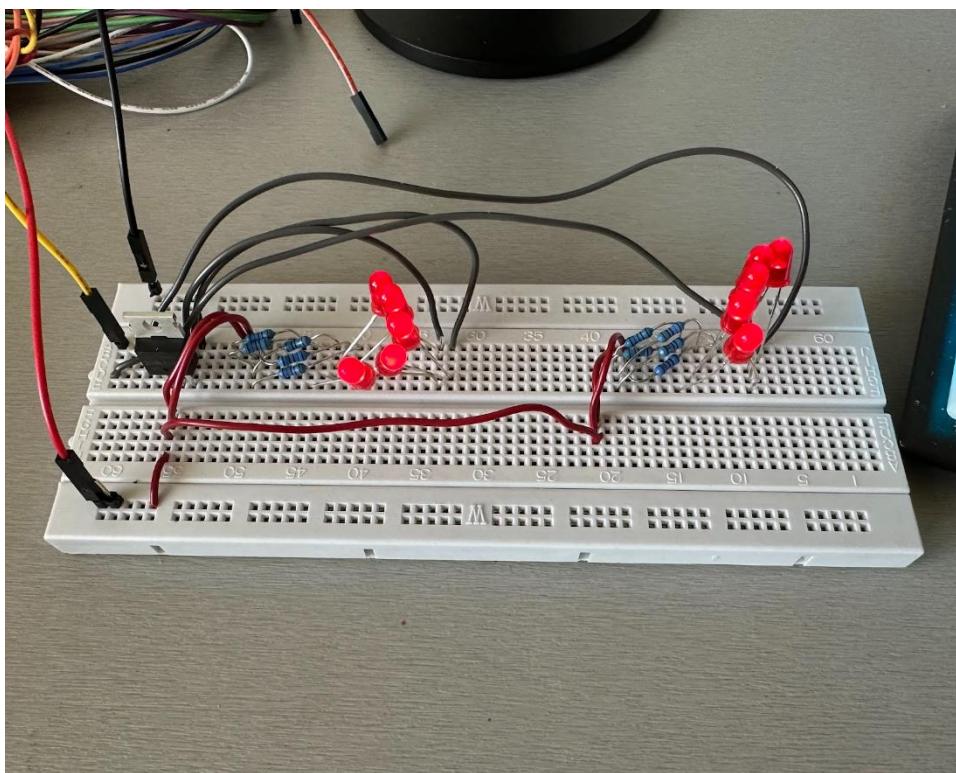


Figur 17: Prototype af det fysiske kredsløb i Ni Multisim

På diagramoversigten i NI Multisim over baglyset ses der, at forsyningsspændingen (VCC) på 5V kommer igennem formodstandene, og derefter hen til de 10 røde lysdioder. De røde lysdioder lyser først, når transistoren får nok spændingsforskæl mellem Gate og Source. Den benyttede MOSFET transistor har en Gate Threshold spænding på minimum 1V, og Arduinoen giver et firkantsignal mellem 0 V og 5 V. Når Gate Threshold spændingen opnås, er der forbindelse mellem lysdioderne og Ground. Da gruppen nu har dannet sig et overblik over, hvordan opstillingen af kredsløbet ser ud, kan kredsløbet nu opstilles på et fumlebræt, og dernæst testes ift. kravene.

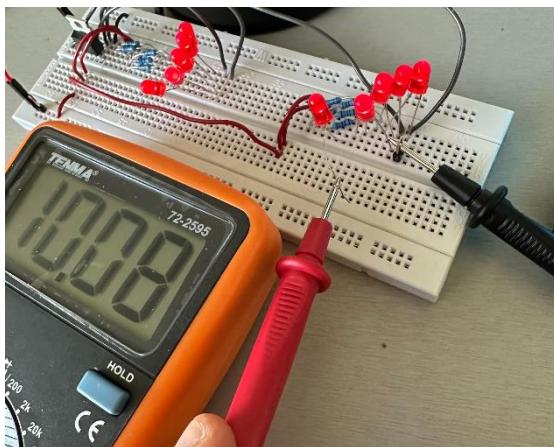
Fysisk opstilling på fumlebræt

Efter at have opstillet kredsløbet i NI Multisim kan det herefter viderekonstrueres på et fumlebræt for at kunne teste om det reelt fungerer. På Figur 18 nedenfor ses den fysiske opstilling på fumlebræt af diagrammet fra tidligere afsnits Figur 17, som var konstrueret i NI Multisim:

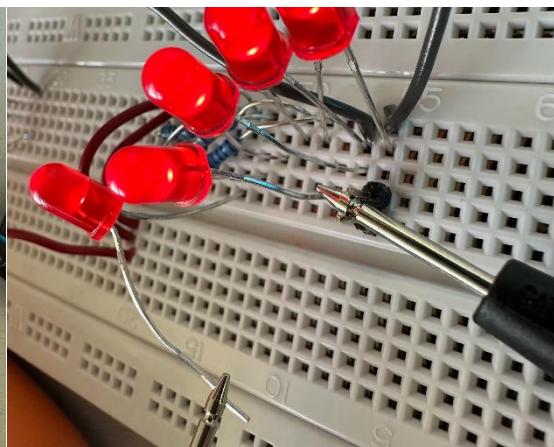


Figur 18: Konstruktion af prototypekredsløbet på fumlebræt fra NI Multisim

Efter at have konstrueret kredsløbet på fumlebrættet kunne middelstrømmen testes, da gruppen nu havde de reelle lysdioder, som passer med formodstandene, hvilket NI Multisim ikke havde. For at gøre dette blev der indsatt et amperemeter i serie med en tilfældig lysdiode i kredsløbet, hvor der blev målt en middelstrøm på 10.08 mA:



Figur 19: Måling af middelstrøm

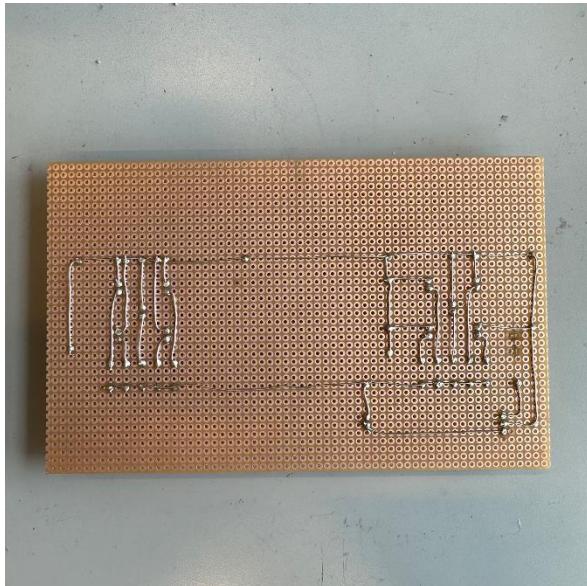


Figur 20: Nærbillede af måling

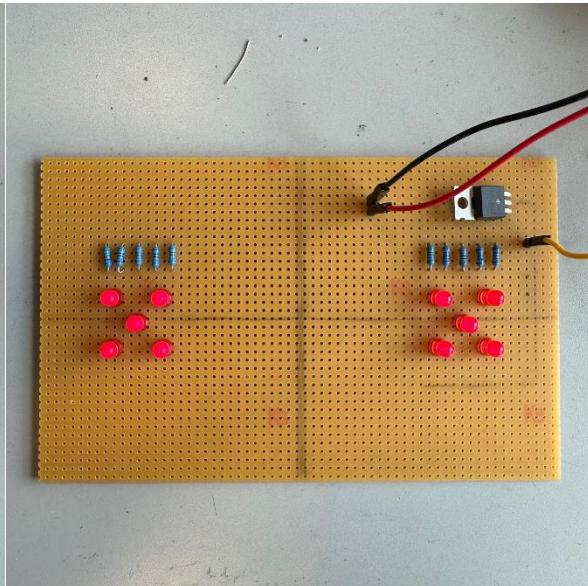
I projektkravene står der, at middelstrømmen ved ”bremselys” gennem hvert LED-sæt skal være 50 mA +/- 5 mA. Da der her blev målt en middelstrøm på 10.08 mA i 1 af lysdioderne ud af de i alt 5 i det ene LED-sæt, giver det en middelstrøm gennem LED-sættet på $10.08 \text{ mA} * 5 = 50.4 \text{ mA}$. Da dette er indenfor kravene for projektet, kan gruppen hermed viderekonstruere kredsløbet på et veroboard.

Fysisk konstruktion på veroboard

Efter at opstillingen var blevet konstrueret på et fumlebræt, og var blevet testet, så kunne opstillingen tages et skridt videre, og hermed implementeres på et veroboard. Da opstillingen skulle matche med opstillingen fra fumlebrættet, så var det let for gruppen at forbinde de præcis samme forbindelser ved brug af loddetin på vero-boardets bagside. Hele konstruktionen kan ses på figurerne nedenfor:

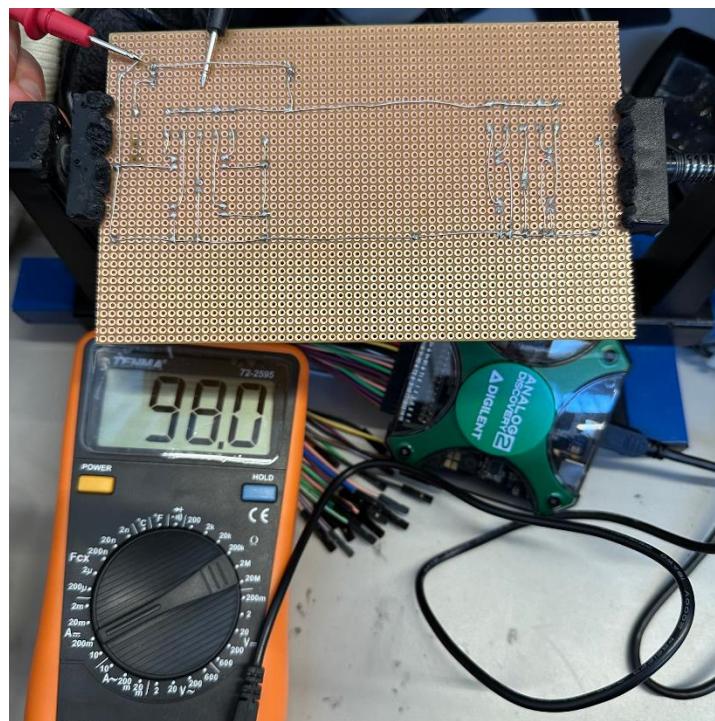


Figur 21: Bagside af konstrueret veroboard



Figur 22: Forside af konstrueret veroboard

Efter at opstillingen var blevet konstrueret og implementeret på veroboardet, blev det testet for at sikre, at middelstrømmen stadig var indenfor opgavens satte krav på $50 \text{ mA} +/- 5 \text{ mA}$ per LED-sæt. Denne test er dokumenteret på figuren nedenfor:



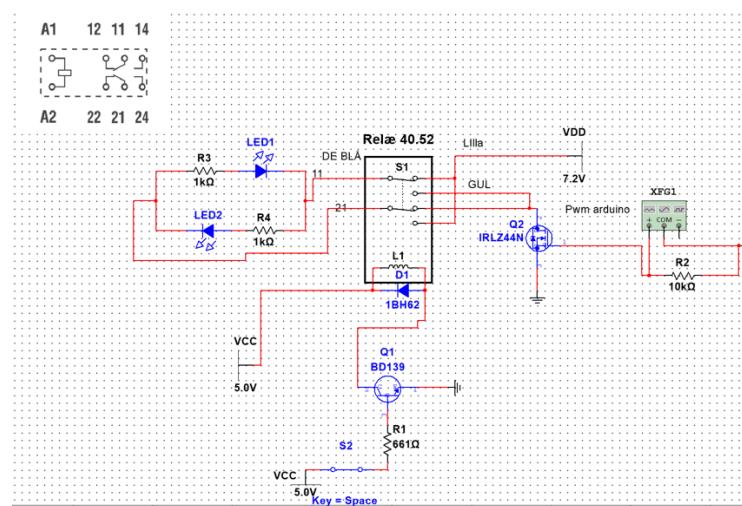
Figur 23: Test af middelstrøm for hele kredsløbet gennem begge LED-sæt sammenlagt

I testen blev et amperemeter indsatt som en del af kredsløbet. Den røde probe fra amperemeteret blev sat til udgangen af alle lysdioder, altså de 2 LED-sæt samlet, og den sorte probe fra amperemeteret blev sat til Ground. Herefter målte amperemeteret en middelstrøm på 98.0 mA. Da dette er for begge af projektgruppens 2 LED-sæts samlede middelstrøm, overholder det, at kravet om middelstrømmen max må være $50 * 2 = 100 \text{ mA} \pm 5 * 2 = 10 \text{ mA}$. Derfor er konstruktionen af projektgruppens baglys på veroboardet nu fuldendt.

Hardware for motor (Anders og David)

I motordelens hardware er der lavet et kredsløb der kan tage et firkantsignal, samt et HIGH/LOW signal fra en mikrocontroller til at styre bilens retning og hastighed.

Her er der blevet gjort brug af et relæ, diverse transistorer, zenerdioder og strømforsyninger.



Figur 24: Motor-kredsløb

Komponenterne i kredsløbet

Relæet der er blevet brugt i kredsløbet, er relæ 40.52, da den fungerer som en H-bro, hvilket kan styre strømmens retning og kan derfor også styre motorenens retning alt efter om den er tændt eller slukket.

Nominal voltage U_N V	Coil code 9.005	Operating range		Resistance R Ω	Rated coil consumption I at U_N mA
		U_{min} V	U_{max} V		
5		3.65	7.5	38	130

Tabel 25: DC coil specifikationer over 40.52 ved 5V.³

Relæet er et 0,65W relæ der ud fra tabellen overfor kan fungere ved en spænding mellem 3,65-7,5V. Samtidig har relæet også en inde modstand på 38Ω og der kører en strøm på $130mA$ hen over relæet.

³ (finder) - PRODUCT SPECIFICATION 40.52 - 2 POLE 8 A (5 MM PIN PITCH)

For at kunne styre om relæet er tændt eller slukket, er der sat et led mellem relæet og ArduinoMega2560'eren i form af transistoren BD139(Q1). Transistoren har også sin egen formodstand på 661Ω (R1). Transistoren bruges i denne sammenhæng som en form for elektrisk kontakt.

Den resistor-værdi skal bruges, da der fra en Multi-sim simulering er blevet målt et spændingsfald over collector til base på 0,7V.

$$V_{CB} = 5V - 0,7V = 4,3V$$

Strømmen I_B bliver herefter fundet:

$$I_B = \frac{I_C}{\beta}$$

I_C kan man fra tabel 21 observeres til at være 130mA og $\beta = 20$.

$$I_B = \frac{130mA}{20} = 6,5mA$$

Til sidst findes modstanden ved hjælp af Ohms lov.

$$R_B = \frac{4,3V}{6,5mA} = 661,5385\Omega$$

Fra BD139 bliver spændingen ført gennem ”relæ-spolen” (L1) der styrer strømmens retning og en zenerdiode (5,1V) der er til for at stabilisere spændingen i kredsløbet.

Transistoren BD139 består af tre ben: Emitter, base og collector. I dette kredsløb sidder basen til Mega2560'eren der tager imod signalet der bliver sendt ind i transistoren.

Herfra bliver signalet ført videre til emitteren, som sidder i forlængelse af kredsløbet og fører signalet videre ind i relæet.

Til sidst er collectoren, som er forbundet til stel.

Udover signalet fra Mega2560'eren og BD139, får relæet endnu et signal fra Mega2560'eren. Dette signal i forlængelse med MOSFET-transistoren IRLZ44N styre motorens hastighed. Dette signal er også udstyret med en pull-down modstand på $10k\Omega$.

IRLZ44N transistoren har ligesom også tre ben dog med andre funktioner: Gate, Drain og Source.

Transistorens ”Gate” (ben nr. 1) tager imod firkantsignalet fra Mega2560'eren og sender det videre ud i kredsløbet gennem transistoren.

”Drain” (ben nr. 2) er både forbundet til strømforsyningen V_{DD} (7,2V) og resten af kredsløbet. Fra dette ben til resten af kredsløbet sendes det firkantsignal videre der stammer fra Mega2560'eren, gennem relæet og indtil motoren.

Mellem ”Drain” og V_{DD} er der også blevet gjort brug af en zenerdiode for at kunne regulere spændingen i kredsløbet.

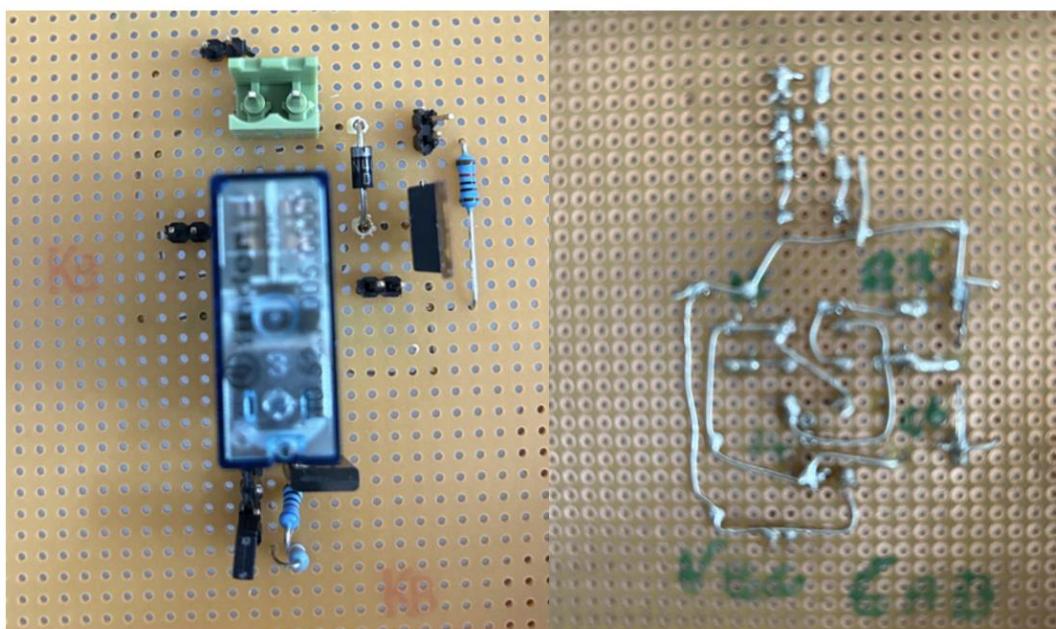
Sidste ben ”Source” (Ben nr. 3) er stelforbindelsen.

Herunder ses en stykliste af komponenterne fra kredsløbet:

Antal	Symbol på diagram	Komponent	Fabrikat/type
1	R1	Modstand	661 Ohm
1	R2	Modstand	10000 Ohm
1	R3	Modstand	1000 Ohm
1	R4	Modstand	1000 Ohm
1	Q1	Transistor	Part No.: IRLZ44N
1	Q2	Transistor	Part No.: BD139
1	40.52	Relæ	Part No.: 40.52
1	M	Motor	Motor
1	D1	Zenerdiode	5.1V
1	D2	Zenerdiode	5.1V
1	S2	Switch	Switch
1	V_cc	Microcontroller	5V/Arduino Mega2560
1	V_dd	Batteri	7.2V Enrich Power

Tabel 22: Stykliste over motor-modul

Det er til sidst blevet loddet på et veroboard:



Figur 26: For-og bagside af veroboard.

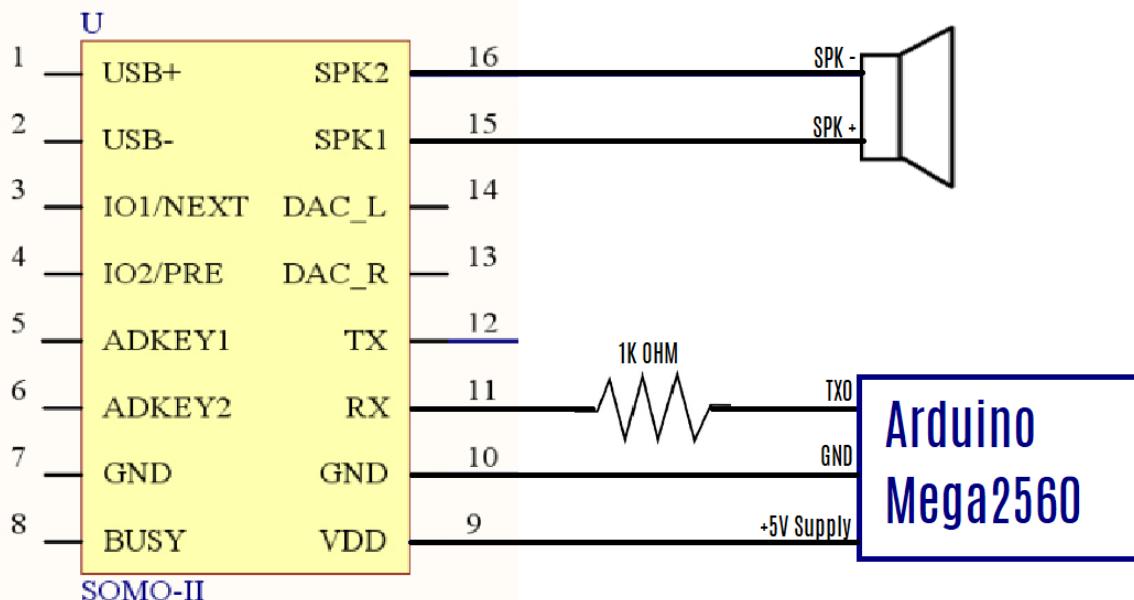
Hardware for lyd (August)

Indledning

En af de valgfrie dele af projektet er lyden. Denne skal, ifølge vores kravspecifikation, kunne spille lyd ved start, slut og refleks-detektion. Til dette udnyttes en SOMO-II MP3 afspiller, som kan spille lyd direkte fra et SD-kort til en tilsluttet højttaler. Denne kan tage kommandoer ved UART, hvilket betyder at det vil være muligt at kommunikere med den fra vores MEGA2560. SOMO-II spilleren kræver desuden også 5V og GND hvilket vi trækker fra vores MEGA2560. Dette får vi fra SOMO-II databladet som er vedlagt i litteraturlisten. SOMO-II'en kan fungere ved blot 3.3V ifølge det vedlagte datablad, men dette påvirker højttalerens lydstyrke. Derfor vælger vi 5V. En vigtig detalje er, at vores MEGA2560 har et TXd-output med en spænding som vores SOMO-II ikke kan holde til. Derfor udnyttes der imellem disse to en $1\text{k}\Omega$ modstand.

Skitsering/Diagram

Vi kan skitsere vores opsætning til lydafspilningen:

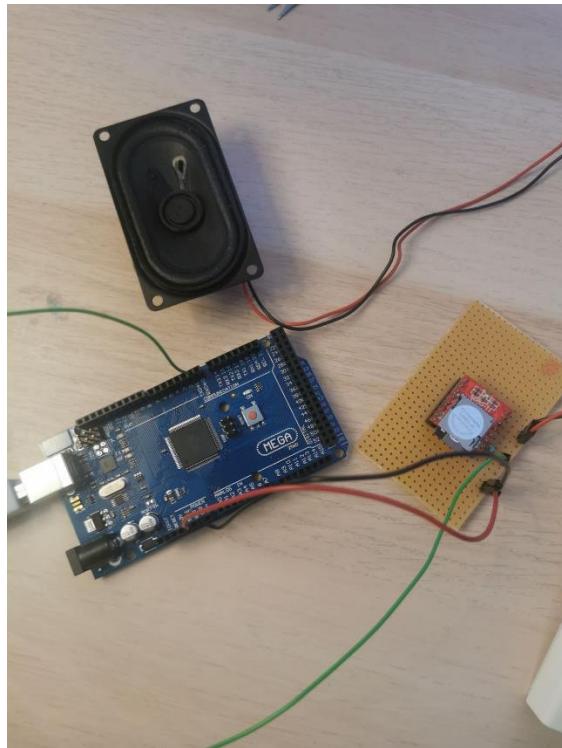


Figur 27: Diagram af lydsystemet

Bemærk her at kun UART'ens TX0 port udnyttes på MEGA2560. SOMO-II og MEGA 2560 understøtter "Full duplex", de kan altså kommunikere begge veje uafhængigt, men vi vælger kun at udnytte TX fra MEGA2560 til RX på SOMO-II da vi kun ønsker at sende signaler til afspilleren, ikke modtage. Højttaleren er på 8Ω og forbinderes med SPK- til SPK2 og SPK+ til SPK1.

Fysisk konstruktion på veroboard

Vi kan nu konstruere denne lyd-afspiller på et veroboard og teste med vores MEGA2560. Da SOMO-II ikke må loddes på (og det gør det nemmere at skifte den ud), sætter vi den i en sokkel. Denne loddes på, og SOMO-II sættes i.



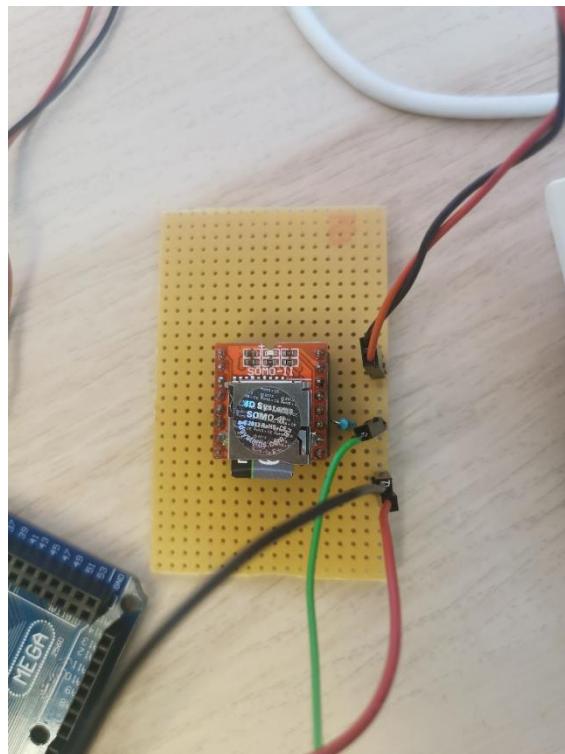
Figur 28: Lydsystemets opsætning

GND/SPK- = SORT

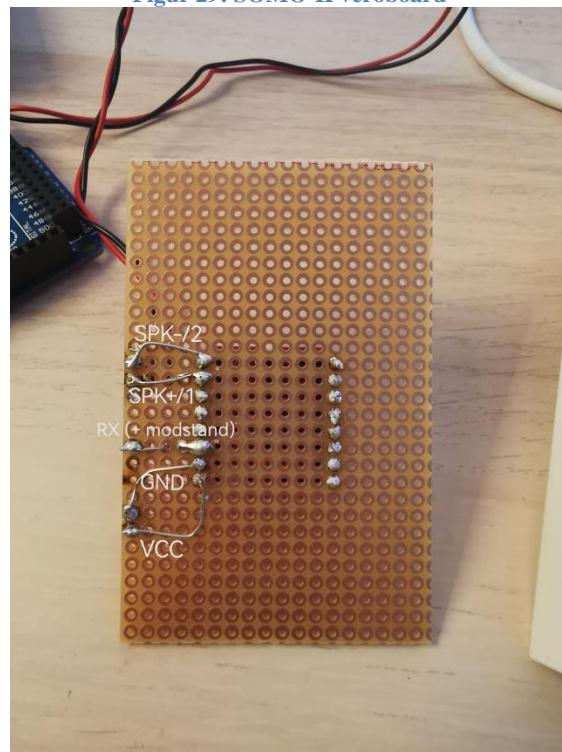
VCC/SPK+ = RØD

UART = GRØN

Bemærk at shield'et på MEGA2560 er fjernet for at kunne tilgå portnavne. Vi har nu et fuldt funktionelt lydsystem som kan kontrolleres med MEGA2560 over UART. C-koden der styrer lyden beskrives i detalje under "Softwaredesign". Nedenunder ses billeder af VERO-boardet i bedre detaljer samt lodninger.



Figur 29: SOMO-II veroboard

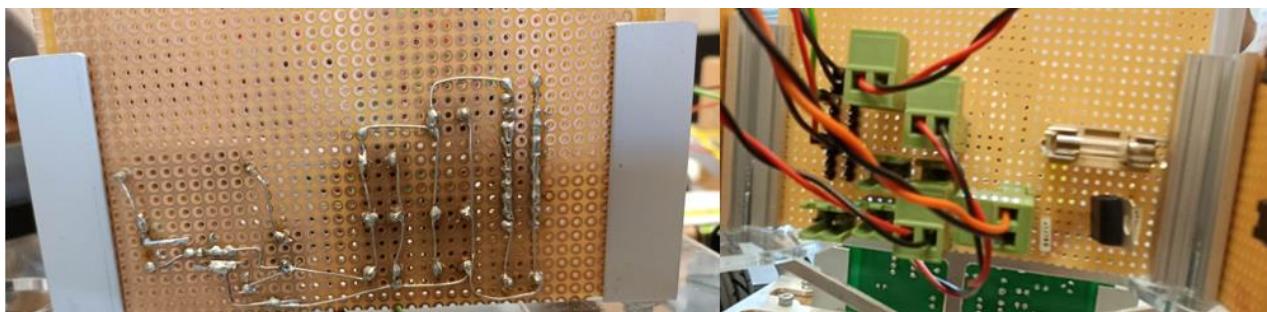


Figur 30: SOMO-II veroboard lodninger

Hardware for power supply(Anders)

Fordi vi skulle bygge vores fejl detektor om, så derfor valgte vi i anden omgang ikke at inkludere en power supply på den og lave en individuel en.

Til vores bil har vi også konstrueret en power supply, det var vi nødt til fordi at vores Arduino ikke kan levere nok strøm. Til at alt vores andet hardware ville kunne køre. Så derfor har vi taget designet af den powersupply, som vi lavede til vores fejl detektor og loddet den på et veroboard.



Figur 31 Forside og bagside af PSU.

Samt har vi også haft brug for ekstra ground, da der ikke er nok i vores Arduino, så det har vi også implementeret på den.

Softwaredesign

Lys (Nahom og Mads)

Forlys (Nahom)

initForlys er skrevet til at initialisere forlysene på bilen. Det er blevet gjort ved at vælge PB ben 6 som output for PWN. DDRH sættes til ben 6 ved at ændre på dets værdi. Efterfølgende bestemmer vi mode for PWM og sætter clock prescaleren til 1. Til sidst sætter vi duty cycle på 875. Koden ser sådan ud:

```
void initForlys(){  
    DDRH = 0b00001000;  
    TCCR4A = 0b10000011;  
    TCCR4B = 0b00000001;  
    OCR4A=875;  
}
```

Baglys (Mads)

For at få baglyset til at fungere, samt alt andet hardware i denne verden, så skal der noget bestemt software til. Denne software skal kunne læses af ATMEGA2560 mikrokontrolleren, og til dette har gruppen anvendt programmeringssproget C.

Void initBaglys()

Funktionen "initBaglys" skal initialisere baglysene ift. PWM, valgt output ben og clock prescaler. Den tager først og vælger ben nummer 11 ved hjælp af sætte DDRB's værdi i det binære talsystem. Derefter vælges den

mode, som PWM skal have samt clock prescaleren, som ved begge også gøres ved hjælp af det binære talsystem:

```
void initBaglys()
{
//PB BEN 11 SOM OUTPUT FOR GRUPPENS PWM
    DDRB = 0b00100000;
    // MODE = 3 (PWM, PHASE CORRECT, 10 BIT)
    // SET OC1A ON MATCH DOWN COUNTING / CLEAR OC1A ON MATCH UP COUNTING
    TCCR1A = 0b10000011;
    // CLOCK PRESCALER = 1
    TCCR1B = 0b00000001;
}
```

Figur 32: void initBaglys()

Efter dette er baglysene blevet initialiseret.

Void styrkeBaglys(double lysstyrke)

Funktionen ”styrkeBaglys” skal styre baglysenes lysstyrke ved hjælp af at ændre på duty cyclen for PWM-signalen. Dette gøres ved hjælp af at tage funktionens parameter, og derefter ændre værdien for OCR1A. OCR1A er ”TOP” på TIMER 1 (TCR1), hvilket er timeren, som er blevet brugt til at styre baglyset på bilen:

```
void styrkeBaglys(double lysstyrke)
{
    // STYRER DUTY CYCLE
    OCR1A = 1023/100*lysstyrke;
}
```

Figur 33: void styrkebaglys(double lysstyrke)

Sensor/Interrupts (Mads og August)

Interrupts er en genial måde for mikrokontrolleren at udføre noget ekstern kode, hvorefter den returnerer tilbage til koden den var i gang med at foretage. For at få disse interrupts implementeret skal der noget bestemt software til. Denne software skal ligesom med forlyst kunne læses af ATMEGA2560 mikrokontrolleren, og til dette har gruppen anvendt programmeringssproget C.

void initINTE()

Funktionen ”initINTE” skal tillade globale interrupts samt tænde for ekstern interrupts for interrupt0 (INT0) og interrupt1 (INT1). Globale interrupts tillades ved hjælp af at bruge funktionen sei(); og afslåes ved hjælp af at bruge funktionen cli();. Det er meget vigtigt at deklære, da hvis dette ikke gøres, så virker interrupts ikke selvom, at man tænder og indstiller sine valgte interrupts ved hjælp af at ændre værdierne for EIMSK og EICRA. Her aktiveres der for INT0 og INT1, da det er dem vi bruger her. Disse sættes til rising edge, sådan at de aktiveres så snart at sensoren registrer en refleks, da dette vil forsage spændingen at stige. Vi vælger ikke falling edge, da det er vigtigt at interrupts bliver slukket hurtigst muligt efter en refleks, sådan at 2 counts ikke registrer.

```
void initINTE(){
    sei(); // TILLADER INTERRUPTS
    EIMSK=EIMSK|0b00000011; // TÆNDER EKSTERN INTERRUPTS FOR INT0 OG INT1
    EICRA=EICRA|0b00001111; // SÆTTER INT0 OG INT1 TIL RISING EDGE INTERRUPT
}
```

Figur 34: void initINTE()

ISR(INT0_vect) + ISR(INT1_vect)

Efter at have tilladt globale interrupts og tændt for INT0 og INT1 kan interrupts opsættes inde i main.c-filen. Først starter interruptet med at øge variablen ”count”, som er en global optæller for, hvor mange reflekser på banen bilen er kommet forbi. Efter dette slukkes interrupts, et delay indsættes for at undgå 2 countere på samme tid, flag nulstilles og der tændes igen for interrupts som set oppe i tidligere afsnit med initINTE() funktionen. Interrupts er tilsluttet INT0 og INT1 hvor vores lyssensorer er sat til. Da disse øger spændingen ved refleksion detekteret, vil dette aktivere en interrupt-routine da vi har initialiseret interrupts til at igangsætte på rising edge. på Koden for begge interrupts kan ses nedenfor på Figur 35:

```
// INTERRUPTS:
ISR(INT0_vect) {
    count++; // GLOBAL OPTÆLLER
    EIMSK=EIMSK|0b00000000; // SLUK FOR INTERRUPTS
    _delay_ms(500); // DELAY FOR AT UNDGÅ 2 COUNTERE
    EIFR=EIFR|0b00000011; // NULSTILLER FLAG
    EIMSK=EIMSK|0b00000011; // TÆNDER FOR INTERRUPTS
}

ISR(INT1_vect) {
    count++; // GLOBAL OPTÆLLER
    EIMSK=EIMSK|0b00000000; // SLUK FOR INTERRUPTS
    _delay_ms(500); // DELAY FOR AT UNDGÅ 2 COUNTERE
    EIFR=EIFR|0b00000011; // NULSTILLER FLAG
    EIMSK=EIMSK|0b00000011; // TÆNDER FOR INTERRUPTS
}
```

Figur 35: ISR(INT0_vect) + ISR(INT1_vect)

Motor (Anders og David)

Der er på baggrund af SW-arkitekturen over motoren blevet opstillet tre funktioner samt en variabel ”dutycycle”.

Funktionen ”directionchange” skal styre bilens retning. Den tager variablen ”direction” som enten har en værdi på 1 eller 0. Først er bit 1 på DDRC gjort til output. Herefter er der blevet opstillet en ”if-sætning” der siger at hvis direction ==1, så vil bit 1 på ”PORTC” være 1. Ellers if (else if) direction!=1 altså hvis direction ikke er lig med 1 vil alle bits på PORTC være slukket.

```
void directionchange (double direction) {
    DDRC=0xFF;
    if ((direction == 1)) {
        PORTC=0b00000010;
```

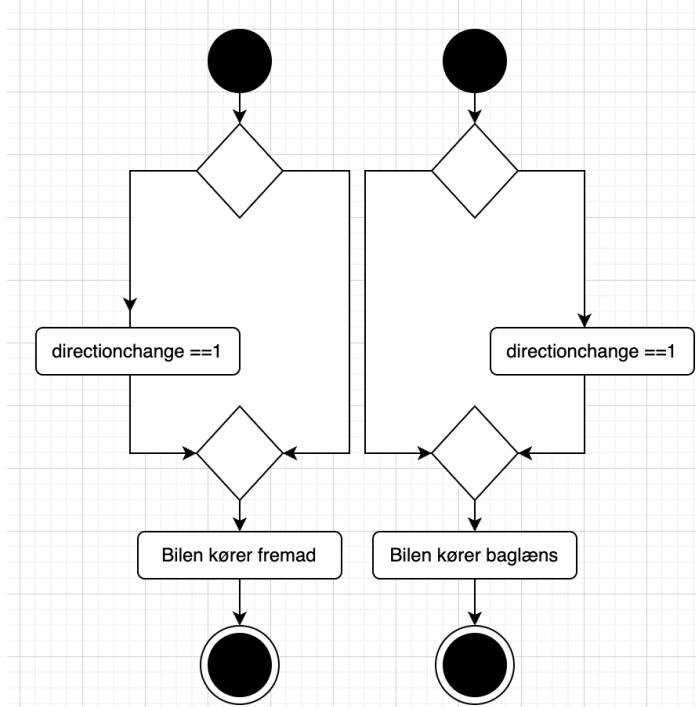
```
}
```

```
{
```

```
Else if ((direction!=1))
```

```
    PORTC=0b00000000;
```

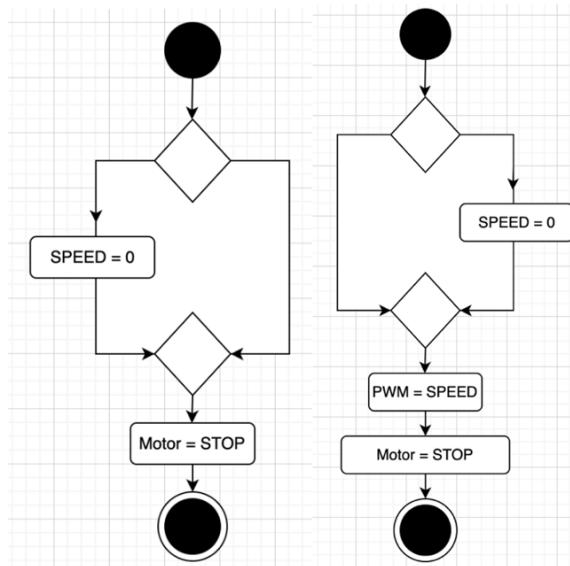
```
}
```



Figur 36: UML-diagram over funktionen `directionchange`.

Funktionen "stop" skal sætte bilen til standsning. Det gøres ved at sætte ORC3A = 0. ORC3A er "TOP" på timer3, hvilket er timeren der er blevet brugt for at styre motoren på bilen.

```
void stop() {
    ORC3A=0;
}
```

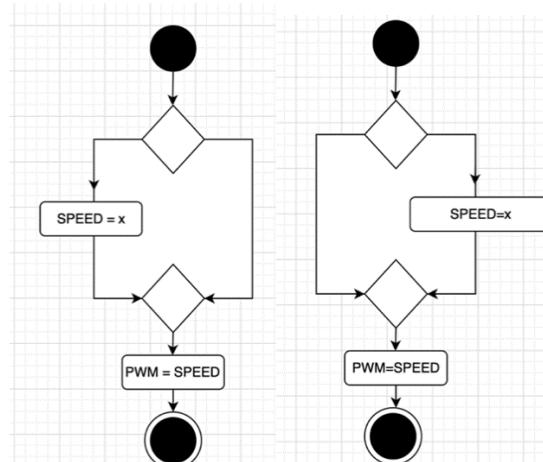


Figur 37: UML-diagram over funktionen "stop"

Funktionen "speed" tager variablen speed til at sætte hastigheden på bilen. Der er i funktionen blevet opstillet en ligning, hvor "speed" som er en varibel fra 0-100, divideres med 100 og ganges med en varibel dutycycle som er defineret til at have en værdi på 1023, hvilket er den maksimale "TOP-værdi" på timer3.

```
double dutycycle_=1023;

void speed_(double speed) {
    OCR3A=(dutycycle/100) *speed;
}
```



Figur 38: UML-diagram over funktionen "speed"

For at funktionerne kan fungere er der opstillet en funktion mere til opstilling af timeren.

```
void inittimer() {  
TCCR3A=0b10000011;  
TCCR3B=0b00000011;  
DDRE=0b00001000;  
}
```

Funktionen "inittimer" bruges til at initiere timer3.

Først indstilles registeret TCCR3A til at sætte den i PWM phase correct 10 bit, sætte den til clear ved upcounting og set ved downcount.

Herefter TCCR3B til at indstille prescaleren til 64.

Derefter sættes topværdien til det maksimale hvilket også er den maksimale topværdi for en 10-bit timer.

Til sidst bliver bit 3 på PINE til output.

Koden er på den måde bygget op, at funktionerne derfor kan kaldes alt efter hvilken refleksbrik bilen er nået til. Herfra kan variablen "speed" ændres til den hastighed bilen skal køre.

Lyd (August)

Indledning

I softwaren skal der laves en driver for SOMO-II. Heri skal den kunne spille, pause, stoppe, afspille en bestemt track men en int som input, skrue op med en int som input og skrue ned med en int som input. Men for at kunne gøre dette, skal vi også kunne initialisere UART'en på Mega 2560 og kunne sende char's over den. Disse UART-funktioner kommer fra LAB i MSYS hvor vi lavede vores egne UART driver. Alt kode skrives i C i atmel studio.

UART-driver

Fra UART-driveren i MSYS er vi interesseret i to funktioner. InitUART() og SendChar(). Disse er relevante, da vi altid skal bruge initUART() til at kunne initialisere UART'en med den rigtige BAUD-rate og data-bit. Disse skal sættes til en BAUD-rate på 9600 og 8 databits ifølge databladet for SOMO-II. SendChar() er vigtig, da vores SOMO-II skal modtage en række hex-tal for at kunne udføre specifikke funktioner. Her er det bl.a. vigtigt at en ny char kun sendes når transmitter registeret er tomt, sådan at vi ved at et nyt tegn kan sendes. Dette sørger SendChar() for. Der laves en headerfil (UART.h) med skabelonen for disse funktioner, og en C-fil (UART.c) med implementeringen:

```
/*  
 * UART.h  
 *  
 * Created: 10-05-2023 11:51:55  
 * Author: August  
 */  
  
//OBS!!! DISSE FUNKTIONER KOMMER FRA UART-DRIVEREN SOM VI LAVEDE I MSYS - SKRIV I RAPPORT!!  
void InitUART(unsigned long BaudRate, unsigned char DataBit);  
void SendChar(char Tegn);
```

Figur 39: UART.h

```
//OBS!!! DISSE FUNKTIONER KOMMER FRA UART-DRIVEREN SOM VI LAVEDE I MSYS
#include "UART.h"
#include <avr/io.h>
#include <stdlib.h>

#define XTAL 16000000

void InitUART(unsigned long BaudRate, unsigned char DataBit)
{
    if ((BaudRate >= 300) && (BaudRate <= 115200) && (DataBit >=5) && (DataBit <= 8))
    {
        UCSR0A = 0b00100000;
        UCSR0B = 0b00011000;
        UCSR0C = (DataBit-5)<<1;
        UBRR0 = (XTAL+(8*BaudRate))/(16*BaudRate) - 1;
    }
}

void SendChar(char Tegn)
{
    while ( (UCSR0A & (1<<5)) == 0 )
    {}
    UDR0 = Tegn;
}
```

Figur40: UART.c

SOMO-II-driver

Der skal laves 6 funktioner til SOMO'en:

- void SOMOplay(); - Skal starte afspilning
- void SOMOpause(); - skal midlertidigt pause afspilning
- void SOMOvolumeIN(int level); - Skal øge lydstyrken med antal int's
- void SOMOvolumeDC(int level); - Skal mindske lydstyrken med antal int's
- void SOMOstop(); - Skal stoppe afspilning helt
- void SOMOtrack(int track); - skal starte med at spille int-track fra SD-kortet

Den ønskede funktionalitet opnås ved at bruge SendChar() fra UART-driveren. I Datasheet'et til SOMO-II står der en række hex-tal for hver funktion den har. Vi kalder derfor SendChar() til hvert hex for at sende hele kommandoen. Dog skal der gøres lidt ekstra når vi har muligheden for at give en int som argument til funktionen. Dette er fordi SOMO-II har en checksum , som udregnes for hver kommando. Denne afhænger af de tal der sendes, og hvis vi ændrer dem, vil checksum også ændre sig. Det observeres for SOMO-funktionen "SPE-CIFY TRACK #" at den rigtige checksum kan opnås ved at 7'ende hex-tal sættes til: 253-int track. Ingen driveren delt op i to dele, en header fil (SOMO-II.h) og en c-fil (SOMO-II.c). Lydstyrkefunktionerne er blot while-loops der kører int-antal gange.

```
void SOMOplay();
void SOMOpause();

//Volume preset is 20. MAX is 30, MIN is 5. Volume is saved until power is cut.
void SOMOvolumeIN(int level);
void SOMOvolumeDC(int level);

void SOMOstop();
void SOMOtrack(int track);
```

Figur41: SOMO-II.h

```

#include <avr/io.h>
#include <stdlib.h>
#include "SOMO-II.h"

// Target CPU frequency
#define XTAL 16000000

void SOMOplay()
{
    SendChar(126);
    SendChar(0x0D);
    SendChar(0x00);
    SendChar(0x00);
    SendChar(0x00);
    SendChar(0xFF);
    SendChar(0xF3);
    SendChar(0xEF);
}

void SOMOpause()
{
    SendChar(0x7E);
    SendChar(0x0E);
    SendChar(0x00);
    SendChar(0x00);
    SendChar(0x00);
    SendChar(0xFF);
    SendChar(0xF2);
    SendChar(0xEF);
}

void SOMOvolumeIN(int level)
{
    while(level>0){
        SendChar(0x7E);
        SendChar(0x04);
        SendChar(0x00);
        SendChar(0x00);
        SendChar(0x00);
        SendChar(0xFF);
        SendChar(0xFC);
        SendChar(0xEF);
        level--;
    }
}

void SOMOvolumeDC(int level)
{
    while(level>0){
        SendChar(0x7E);
        SendChar(0x05);
        SendChar(0x00);
        SendChar(0x00);
        SendChar(0x00);
        SendChar(0xFF);
        SendChar(0xFB);
        SendChar(0xEF);
        level--;
    }
}

void SOMOstop()
{
    SendChar(0x7E);
    SendChar(0x16);
    SendChar(0x00);
    SendChar(0x00);
    SendChar(0x00);
    SendChar(0xFF);
    SendChar(0xEA);
    SendChar(0xEF);
}

void SOMOtrack(int track){
    SendChar(0x7E);
    SendChar(0x03);
    SendChar(0x00);
    SendChar(0x00);
    SendChar(track);
    SendChar(0xFF);
    SendChar(253-track);
    SendChar(0xEF);
}

```

Figur 42: UART.c

Main (Anders og David)

Start: Vores bil begynder at kører fremad ved at kalde speed_, samt at den afspiller vores start lyd.

1: Speed_ forsætter samtidigt med vores reflex lyd bliver afspillet.
Sætter os styrke til baglys.

2: Speed_ forsætter samtidigt med vores reflex lyd bliver afspillet

3: Refleks lyd bliver afspillet igen, samt sætter vi vores hastighed ned fordi vi er kommet op ad bakken.

4: Refleks lyd bliver afspillet igen, samt sætter vi speed_lidt opad for at komme hurtigere afsted.

5: Refleks lyd bliver afspillet igen, mens speed_ ligger på sammen værdi

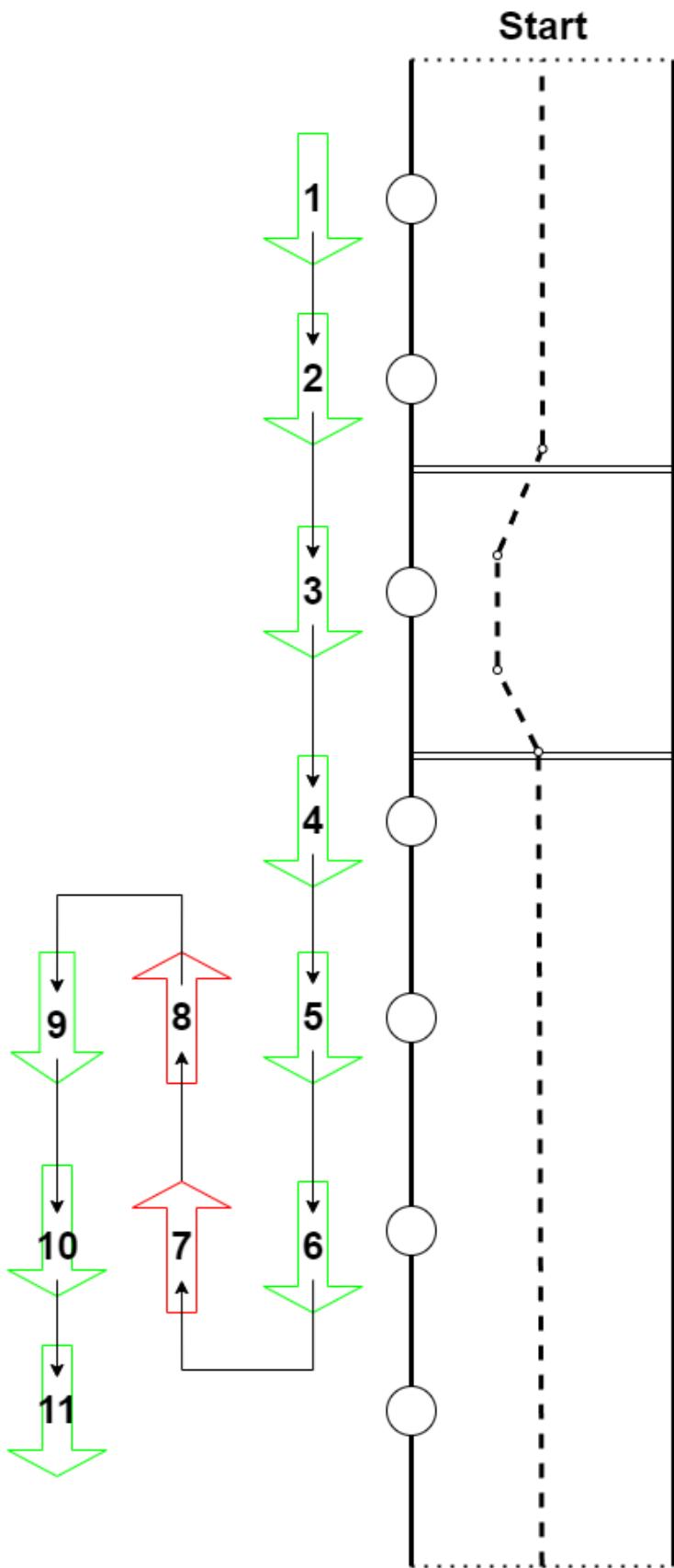
6: Her kalder vi for første gang Direction Change_ fordi at vi skal til at køre baglæns. Vi sætter også lysstyrken på vores baglys op, fordi bilen bremser.

7: Vi kommer forbi den sammen refleks igen, så derfor kalder vi funktionen speed igen.

8: Her kalder vi for anden gang Direction Change_ fordi at vi skal til at køre forlæns. Vi sætter også lysstyrken på vores baglys op, fordi bilen bremser.

9/10: Her bliver vores funktion speed_kaldt til cirka det sammen, samt at når den krydser 9. bliver der afspillet en lyd.

11: Som slut spiller vi vores "Start/slut" igen, Stopper bilen og kalder et delay inden vi slukker for baglyset. For at signalere at bilen holder stille.



Slut

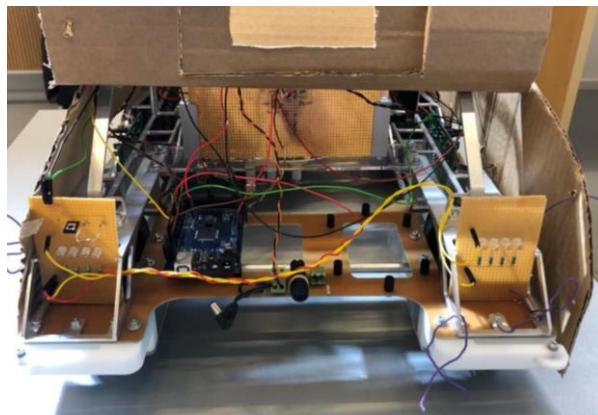
Figur 43 Banen set oppe fra med pile for at beskrive koden.

Implementering og modultest

Lys (Nahom og Mads)

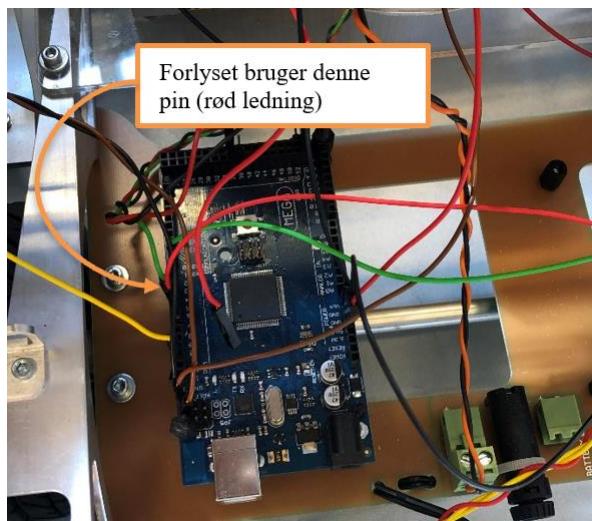
Forlys (Nahom)

Forlyset består af to LED-sæt som er loddet fast på hver deres veroboard. Disse 2 veroboard placeres foran på både højre og venstre side af bilen, hvor de vil fungere som forlys. På figuren nedenfor ses, hvordan det ser ud:



Figur 39: Forlys placeret på bilen

Forlyset på bilen er forbundet til en ATMEGA2560 mikrocontroller ved hjælp af 3 forbindelser. Vi har en ledning til VCC, en ledning til Ground og en ledning til PWM-signalet. PWM-signalet styrer hvornår forlyset er slukket og tændt.

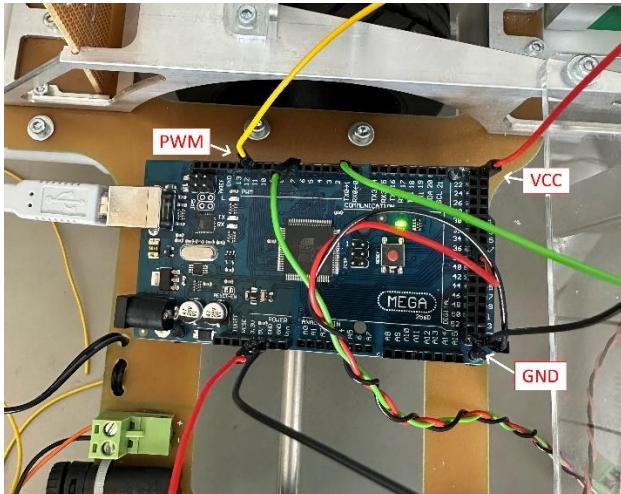


Figur 40: ATMEGA2560 tilkoblinger

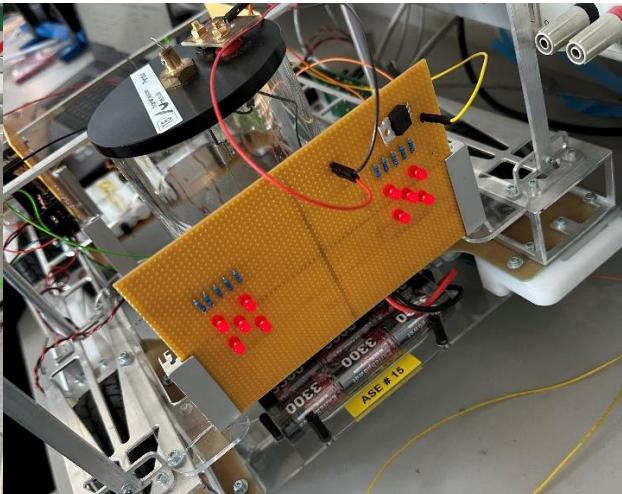
I specifikationerne for ATMEGA2560 mikrocontrolleren fremgår det, at den valgte pin er tildelt funktionssignalet OCR4A. Derfor anvender vi Timer 4 i vores software til at generere et PWM-signal. Vi foretager de nødvendige initialiseringer af TCCR4A og TCCR4B og definerer derefter duty cycle-værdien på OCR4A til at være 875.

Baglys (Mads)

Baglysets 2 LED-sæt er, som tidligere beskrevet, blevet loddet på et veroboard, som påsættes bilen. Bilkarrosseriet består allerede af nogle konstruerede placeringsmuligheder, som bliver benyttet til påsætningen. Dette kan ses på nedenstående figurer:



Figur 44: Tilkoblinger på ATMEGA2560

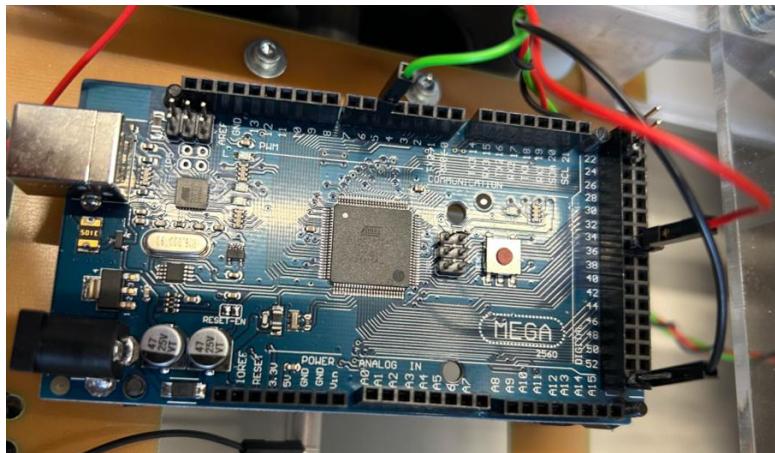


Figur 44: Veroboard monteret på bilkarrosseriet

På Figur 44 og Figur 45 ses de forskellige ledninger, som går fra gruppens ATMEGA2560 mikrocontroller, placeret på forenden af bilen, til veroboardet, som er placeret på bagenden af bilen. Gruppen har en ledning til PWM-signalet, som er tilsluttet den nummereret pin 11 på mikrokontrolleren. Pins fra 0-13 kan lave PWM-signaler, og derfor benyttes den valgte pin. Det kan på databladet for ATMEGA2560 mikrokontrolleren ses, at den valgte pin har funktionssignalet OC1A. Derfor benyttes TIMER 1 til at lave PWM-signalet ved hjælp af at i softwaren at initialisere TCCR1A og TCCR1B, og derefter deklære duty cyclen på OCR1A, som er med til at ændre på lysstyrken for baglyset.

Motor (Anders og David)

Kredsløbet fra figur 20 (billede af multi-sim) er herefter blevet loddet fast på et veroboard, hvorfra det herefter bliver forbundet til ATMEGA2560, som skal køre en testkode, der er blevet skrevet til micro-controlleren for at teste motorens forskellige funktioner.



Figur 45: Rød = V_{cc} / Sort = Stel / Grøn = PWM

Veroboardet over motor-hardwaren bliver forbundet til ATMEGA2560's hhv. pin 5 på E-registeret hvor den skal modtage PWM-signalet der skal styre bilens hastighed og om den skal stoppe op.

Her er pin 5 på E-registeret blevet gjort til output, da det også er der Timer3 kan sende PWM-signal ud fra på den specifikke micro-controller. Det er blevet gjort således:

```
DDRE=0b00001000;
```

Bilens anden funktion er at kunne styre retningen den skal køre, ved at tænde og slukke relæet. Det er blevet gjort ved at gøre bit 1 på C-registeret og ground, som enten skal være IN eller OUTPUT, som beskrevet i afsnittet "SW-design".

Test-softwaren der er blevet skrevet dertil har til formål at teste: ændringer i hastighed, at få bilen til at stoppe og ændre bilens køreretning:

```
int main (void)
{
inittimer_();
    while (1) {
}
PORTC=0b00000000;
speed_(95);
_delay_ms(5000);
_delay_ms(5000);
stop();
speed_(90);
directionchange(1);
_delay_ms(5000);
directionchange(0);
_delay_ms(5000);
```

```
stop();  
}
```

Der kan ud fra test-softwaren konstateres at bilens hard-og softwaren virker.

Lyd (August)

For at teste lyden er der skrevet et simpelt program der tester de forskellige funktioner som er blevet lavet. Opstillingen på bilen matcher den beskrevet i "Hardwaredesign - lyd). Der bruges altså TX0, 5V og GND fra Mega2560. Et rigtigt output i lydformat kan ikke dokumenteres, dog kan det konstateres hvis det virker. Følgende program bruges:

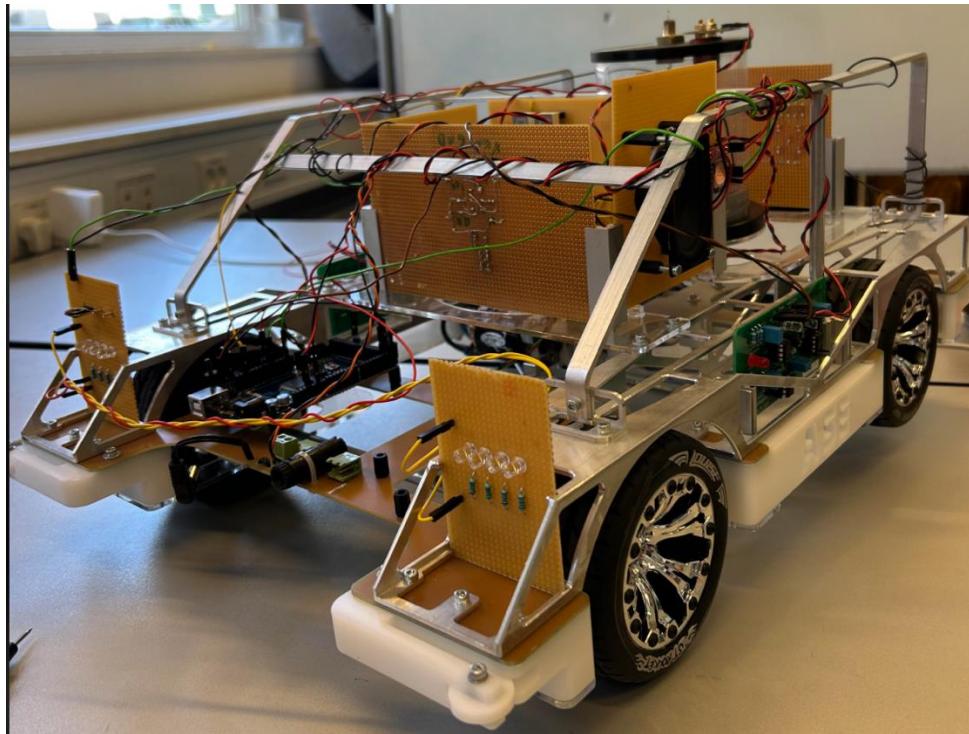
```
#include "UART.h"  
#include "SOMO-II.h"  
#include <util/delay.h>  
  
// Target CPU frequency  
#define XTAL 16000000  
  
  
int main(){  
    InitUART(9600,8); //Initialiserer UART  
    SOMOvolumeIN(7); //Skruer lydstyrken op med 7  
    SOMOtrack(1); //Starter afspilning af første fil på SD-kortet  
    _delay_ms(4000); //Venter 4 sekunder  
    SOMOpause(); //Pauser musikken  
    _delay_ms(100); //Venter 0,1 sekunder  
    SOMOplay(); //Starter afspilning efter pause  
    _delay_ms(2000); //Venter 2 sekunder  
    SOMOvolumeDC(10); //Skruer ned for lyden med 10  
    SOMOtrack(2); //Starter afspilning af anden fil på SD-kortet  
    _delay_ms(8000); //Venter 8 sekunder  
    SOMOstop(); //Stopper al afspilning  
  
    return 0;
```

Figur 46: Test-program til lyd

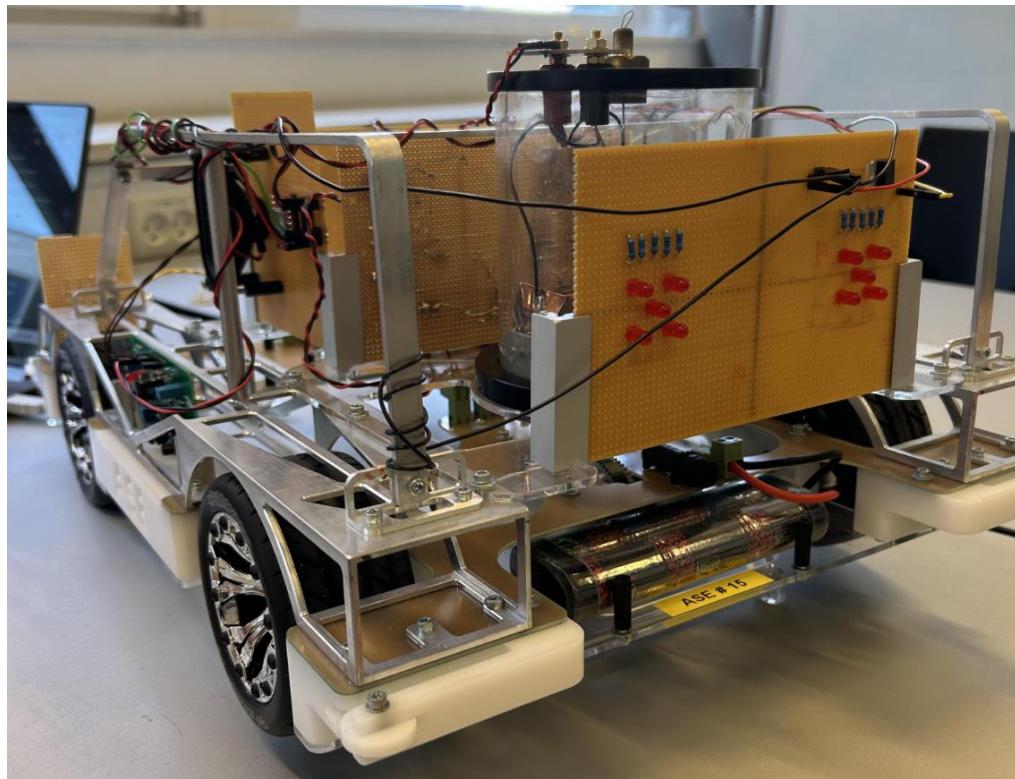
Efter kørsel af programmet, kan det afgøres at lyden opfører sig som beskrevet.

Integrationstest (David og Anders)

Ved integrationstest-fasen er alle moduler/hardwaremæssige komponenter blevet påsat bilkarrosseriet, og der er herefter blevet testet om alle komponenter spiller sammen samt overholder de forskellige krav sat for projektet.



Figur 47: Bilen med alle hardware moduler sat på og tilsluttet set forfra



Figur 48: Bilen med alle hardware moduler sat på og tilsluttet set bagfra

På figur 36 og 37 ses billeder af det endelige resultat af bilen påsat alle hardware-moduler tilsluttet batterier og ATMEGA2560.

Til tilslutningen af de forskellige moduler både mellem modulerne og til ATMEGA2560, er der blevet gjort

brug af forskellig farvede ledninger til hhv. V_{cc} (rød), Stel (sort), PWM fra motor og lyd (grøn), PWM fra lys (gul) og signal til interrupt fra refleks detektorer (brun). Her er der også taget højde for vibrationer fra bilen under kørsel og elektrisk støj ved at sno ledningerne om hinanden.

Udover tilslutningen af alle hardware-moduler er alle funktionerne fra de forskellige SW-moduler samlet i forskellige drivers og implementeret i en samlet main.c fil der skal bruges til at kunne udføre de forskellige test til accepttest-fasen i næste hoved afsnit.

Main.c-filen kan ses nedenfor på figur 49,50,51 og 52:

```
/*
 * MAIN.C
 *
 * CREATED: 01-06-2023 23:38:17
 * AUTHOR: GRUPPE SW9
 */

// DEFINITION AF CPU FREKVENS SAMT INKLUDERING AF BIBLIOTEKER OG HEADERFILER:
#define XTAL 16000000
#define F_CPU 16000000
#include <avr/io.h>
#include <avr/io.h>
#include <stdlib.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include "LYD/UART.h"
#include "LYD/SOMO-II.h"
#include "MOTOR/MOTOR.h"
#include "INTERRUPT/SENSOR.h"
#include "FORLYS/FORLYS.h"
#include "BAGLYS/BAGLYS.h"
int count=0;

// INTERRUPTS:
ISR(INT0_vect) {
    count++; // GLOBAL TÆLLER OPTÆLLER
    EIMSK=EIMSK|0b00000000; // SLUK FOR INTERRUPTS
    _delay_ms(500); // DELAY FOR AT UNDGÅ 2 COUNTERE
    EIFR=EIFR|0b00000011; // NULSTILLER FLAG
    EIMSK=EIMSK|0b00000011; // TÆNDER FOR INTERRUPTS
}

ISR(INT1_vect) {
    count++; // GLOBAL TÆLLER OPTÆLLER
    EIMSK=EIMSK|0b00000000; // SLUK FOR INTERRUPTS
    _delay_ms(500); // DELAY FOR AT UNDGÅ 2 COUNTERE
    EIFR=EIFR|0b00000011; // NULSTILLER FLAG
    EIMSK=EIMSK|0b00000
'011; // TÆNDER FOR INTERRUPTS
}

int main() {

// INITIALISERING AF BIL:
    initINT(); // INITIALISER INTERRUPTS TIL SENSOR
    InitUART(9600,8); // INITIALISER LYD
```

```

initBaglys(); // INITAILISER BAGLYS
initForlys(); // INITAILISER FORLYS
SOMOvolumeIN(5); // INDSTILLER LYDSTYRKE
SOMOtrack(1); // STARTER LYD 1
_delay_ms(5000); // VENTER TIL LYD ER AFSPILLETT (5 SEK.)
styrkebaglys(80); // INDSTILLER BAGLYSETS LYSSTYRKE
count=0; // SÆTTER COUNT TIL 0, SÅ DEN ER LIG MED 0 FRA START
intittimer_(); // INITIALISER MOTOR

// IMENS DER ER 0/INGEN SENSORER REGISTRERET:
while(count==0) {
}

if(count==1) {
    // UDFØR HANDLING VED FØRSTE REFLEKS:
    SOMOtrack(2); // LYD NUMMER 2 (FARTLYD) AFSPILLES
    speed_(90); // FART 90%
    styrkebaglys(30); // STYRKE FOR BAGLYS
}

// IMENS DER ER 1 SENSOR REGISTRERET:
while(count==1) {
    styrkebaglys(30); // STYRKE FOR BAGLYS
    speed_(90); // FART 90%
}

if(count==2) {
    // UDFØR HANDLING VED ANDEN REFLEKS:
    SOMOtrack(2); // LYD NUMMER 2 (FARTLYD) AFSPILLES
    speed_(90); // FART 90%, DA BILEN SKAL OPAD BAKKEN
}

// IMENS DER ER 2 SENSORER REGISTRERET:
while (count==2) {
    speed_(90); // FART 90%, DA BILEN SKAL OPAD BAKKEN
}

if (count==3) {
    // UDFØR HANDLING VED TREDJE REFLEKS:
    SOMOtrack(2); // LYD NUMMER 2 (FARTLYD) AFSPILLES
    speed_(50); // FART 50%, DA BILEN SKAL FORSIGTIGT NEDAD BAKKEN
}

// IMENS DER ER 3 SENSORER REGISTRERET:
while (count==3) {
    speed_(50); // FART 50%, DA BILEN SKAL FORSIGTIGT NEDAD BAKKEN
}

if (count==4) {
    // UDFØR HANDLING VED FJERDE REFLEKS:
    SOMOtrack(2); // LYD NUMMER 2 (FARTLYD) AFSPILLES
    speed_(60); // FART 60%
}

// IMENS DER ER 4 SENSORER REGISTRERET:
while(count==4) {
    speed_(60); // FART 60%
}

if (count==5) {
    // UDFØR HANDLING VED FEMTE REFLEKS:
    SOMOtrack(2); // LYD NUMMER 2 (FARTLYD) AFSPILLES
    speed_(50); // FART 50%
}

```

```
// IMENS DER ER 5 SENSORER REGISTRERET:  
    while(count==5) {  
        speed_(50); // FART 50%  
    }  
  
    if (count==6) {  
        // UDFØR HANDLING VED SJETTE REFLEKS:  
        directionchange(1); // 1 = BIL KØRER FREMAD / 0 = BIL KØRER  
BAGUD  
        styrkebaglys(90); // STYRKE FOR BAGLYS  
        speed_(60); // FART 60%  
    }  
  
// IMENS DER ER 6 SENSORER REGISTRERET:  
    while(count==6) {  
        speed_(60); // FART 60%  
    }  
  
    if (count==7) {  
        // UDFØR HANDLING VED SYVENE REFLEKS:  
        speed_(50); // FART 50%  
        styrkebaglys(30); // STYRKE FOR BAGLYS  
    }  
  
// IMENS DER ER 7 SENSORER REGISTRERET:  
    while(count==7) {  
        speed_(50); // FART 50%  
    }  
  
    if (count==8) {  
        // UDFØR HANDLING VED OTTENE REFLEKS:  
        _delay_ms(200); // DELAY SÅ BILEN KOMMER TÆTTERE PÅ "RIGTIGT  
                      STOPPESTED"  
        directionchange(0); // 1 = BIL KØRER FREMAD / 0 = BIL KØRER  
BAGUD  
        styrkebaglys(30); // STYRKE FOR BAGLYS  
        speed_(60); // FART 60%  
    }  
  
// IMENS DER ER 8 SENSORER REGISTRERET:  
    while (count==8) {  
        directionchange(0); // SKIFTER RETNING  
        speed_(60); // FART 60%  
    }  
  
    if (count==9) {  
        // UDFØR HANDLING VED NIENE REFLEKS:  
        SOMOTrack(2); // LYD NUMMER 2 (FARTLYD) AFSPILLES  
        speed_(50); // FART 50%  
    }  
// IMENS DER ER 9 SENSORER REGISTRERET:  
    while(count==9) {  
        speed_(50); // FART 50%  
    }  
  
    if (count==10) {  
        // UDFØR HANDLING VED TIENDE REFLEKS:  
        speed_(60); // FART 60%  
    }  
  
// IMENS DER ER 10 SENSORER REGISTRERET:  
    while (count==10) {  
        speed_(60); // FART 60%
```

```

        }

    if (count==11) {
        // UDFØR HANDLING VED ELFTE REFLEKS:
        SOMOTrack(1); // SPILLER START/SLUT LYD/MELODI
        stop(); // STOPPER BILEN
        _delay_ms(1000); // LAVER ET DELAY (1 SEK), SÅ BAGLYSET SLUK-
        KER,
                                         NÅR BILEN HOLDER STILLE
        styrkebaglys(0); // SLUKKER BAGLYS
    }

// IMENS DER ER 11 SENSORER REGISTRERET:
while (count==11) {
    stop(); // STOPPER BILEN
}

return 0;
}

```

Ud fra den endelige main.c-fil, og sammen med de færdige hardwaremoduler, kan der udføres forskellige tests taget fra accepttestsspecifikationen.

Accepttest (David)

Funktionelle krav (use cases)

Use case 1 Kør banen	Test	Forventet resultat	Opnået resultat	Godkendt/Kom- mentar
Punkt 1 + Punkt 2	Brugeren tænder for bilen ved hjælp af tænd-knap, og placerer den, så den kører forlæns ind på banen ved at passere startlinjen. Når bilen har passeret refleksbrik nummer 6, bringes bilen til standsning, inden refleksbrik nummer 7 nås.	Visuel test: Bilen standser mellem refleksbrikkerne 6 og 7.	Efter tryk på tænd knap kører bilen fremad og bringes til standsning mellem refleksbrik 6 og 7.	
Punkt 3	Bilen bakker, indtil refleksbrik nummer 5 er passeret, og bringes til standsning inden refleksbrik nummer 4 nås.	Visuel test: Bilen bakker og standser mellem refleksbrikkerne 4 og 5.	Mellem refleksbrik 4 og 5 bringes bilen til standsning.	

Punkt 4	Bilen kører forlæns, indtil refleksbrik nummer 7 nås.	Visuel test: Bilen kører fra mellem refleksbrikkerne 4 og 5 og standser ved refleksbrik 7.	Efter standsning mellem refleksbrik 4 og 5 fortsætter bilen med at køre forlæns frem til refleksbrik 7.	
Punkt 5	Bilens bringes til standsning i målområdet, der er defineret som området mellem banens slut ende og en meter efter banens slut ende.	Visuel test: Bilen kører fra refleksbrik 7, og standser mellem banens slut ende, hvor sidste refleksbrik også befinder sig, og en meter efter banens slut ende.	Efter passeirng af refleksbrik 7 standser bilen op og slutter inden for en meter efter banens slut.	

Tabel 10: Use case 1

Use case 2 Afspil lyd	Test	Forventet resultat	Opnået resultat	Godkendt/kommentar
Punkt 1	Use Case 1 ("Kør banen") udføres.	Akustisk test: Når bilen tændes for at køre ind på banen: Der afspilles 4 sekunders trommeslageren, og gruppemedlemmerne lytter for at bekræfte.	Der bliver afspillet trommeslageren ved tænding.	
Punkt 2	Hver gang en refleksbrik passeres, afspilles en lydeffekt, som er en bil, der gasser op.	Akustisk og visuel test: Bilen placeres før en hvilken som helst refleksbrik og tændes for at forbi passere den. Når den forbi passeres, så skal en lydeffekt afspilles, som er en bil, der gasser op.	Ved passering af hver refleksbrik afspilles lyden af en bil der gasser op.	
Punkt 3	Når refleksbrik nummer 7 passeres, afspilles en lyd hvor 4 sekunders trommeslageren spilles.	Akustisk test: Vi får bilen til at passerer den 7. refleks og gruppemedlemmerne hører om, der afspilles en lyd der er 4 sekunders trommeslageren.	Der bliver ved passerig af refleksbrik 7 afspillet 4 sekunders trommeslageren.	

Tabel 11: Use case 2

Use case 3 Styr forlys	Test	Forventet resultat	Opnået resultat	Godkendt/kom- mentar
Punkt 1	Forlyset skal være tændt, når bilens motor påtrykkes en spænding, for at bringe den til at køre fremad. Når bilen bakker, skal forlyset også være tændt. Når bilen ikke køre, så skal forlyset være slukket.	Visuel test: Vi tænder for bilen, og får den til at køre frem, hvilket betyder, at der påtrykkes en spænding på motoren. Derefter ser vi, om forlygterne tænder.	Ved tænding af bilen, kan der bekræftes at forlygterne også tænder.	

Tabel 12: Use case 3

Use case 4 Styr baglys	Test	Forventet resultat	Opnået resultat	Godkendt/kom- mentar
Punkt 1	Baglyset skal lyse med mellemstyrke ("almindeligt baglys"), når bilens motor påtrykkes en spænding, for at bringe den til at køre. Ellers skal baglyset være slukket.	Visuel test: Vi placerer bilen i hvilken som helst position, og tænder den, for at få den til at bevæge sig/køre. Når den bevæger sig/kører, så kigges der på, om baglyset lyser med mellemstyrke. Når bilen holder op med at køre, så kigges der på, om baglyset slukker, da der ikke længere vil være spænding på bilens motor.	Bilens baglys lyser med mellemstyrke når den bevæger sig i hvilken som helst retning, og stopper med at lyse når bilen har udført banen og er stoppet helt op.	
Punkt 2	Baglyset skal lyse med kraftig styrke ("bremselys"), mens spændingen til bilens motor mindskes. Bremselflyset skal herefter forblive tændt i 0,5 sekund +/- 0,1 sekund.	Visuel test: Bilen placeres mellem 2 refleksbrikker, hvor hældningen er flad. Derefter tændes bilen, og der observeres, om baglysende lyser kraftigere end, der bilen startede, proportionalt med at hastigheden på bilen falder.	Der kan under visuel test ses at baglyset lyser kraftigere proportionalt med at bilens spænding mindskes, og det indenfor 0,5-1 sekund.	

		Når lyset begynder at lyse kraftigt, startes et stopur på en digital enhed, og der måles fra, at lyset begynder at lyse kraftigere til, at styrken igen mindskes, hvilket skal være 0,5 +/-0,1 sekunder.		
--	--	--	--	--

Tabel 13: Use case 4

Ikke-funktionelle krav

Krav nr.	Krav	Test	Forventet resultat	Opnået resultat	Godkendt/Kommentar
1.1	Bilen skal på en enkelt opladning at dens batterier kunne gennemføre mindst 5 gennemkørsler af banen.	Vi kører lader bilen fuldfører bane, hvorefter gruppen placerer den ved start imens, at vi tæller hvor mange gange den kører frem og tilbage. Inden gruppen sætter den til at køre skal den have fuldt opladt batteri.	Vi forventer at vores bil vil kunne komme til at køre frem og tilbage på den bane, som er bygget minimum 5 gange.	Bilen kan udføre banen 6 gange på en opladning.	
1.2	På bilens højre og venstre side skal placeres detektorer, der kan registrere en R80 refleksbrik afstanden 2 cm til 25 cm.	Vi får bilen til at køre forbi hvilke som helst R80-refleksbrikker på både højre og venstre side i forskellige afstande mellem 1-26cm for at se om de bliver registreret af bilens detektorer.	Vi forventer at detektorerne på hver side af bilen registrerer de forbipasserende R80-refleksbrikker, når bilen bevæger sig forbi dem.	Bilen kan opfange refleksbrikkerne på en afstand mellem 2 cm og 25 cm.	
1.3	Bilen monteret med al udstyr må	Bilen monteret med al udstyr vejes på en vægt med en	Vægten viser en vægt mindre end	Bilen vejer 4,565kg.	

	maksimalt veje 5,2 kg	målenøjagtighed bedre end 100 gram.	eller lig med 5,1 kg.		
1.4	Bilens maksimale højde skal være 41 cm.	Vi mäter bilen fra dens laveste punkt, op til bilens højeste med en tommestok. Tommestokken skal have en målenøjagtighed på 5 mm	Det forventes at vi mäter en højde som er mindre end eller lig med 41 cm.	Bilens højde er på 26,4cm	
2.1	Forlys implementeres med 4 hvide LED-sæt, der monteres med et sæt i henholdsvis højre og venstre side på vandret led ved siden af hinanden.	Visuel: Testet om der sidder 4 hvide LED sæt på vandret led på hhv. højre og venstre side af bilen.	Bilen opfylder kravet om LED-monteringen.	Der er monteret 1 sæt hvide LED lygter på hhv. højre og venstre side på forsiden af bilen.	
2.2	Når forlyset er tændt, skal middelstrømmen gennem hvert LED sæt være 50 mA +/- 5 mA.	Vi sætter et multimeter i serie med LED-sætkredsløbet, og mäter herefter, om strømmen er 50 mA +/- 5mA.	LED-sættet opfylde kravet om en middelstrøm på 50 mA.	Forlyset opnår en middelstrøm på 52 mA mens det er tændt.	
3.1	Bag- og bremselys implementeres med 2 LED-sæt af 5 røde dioder, der monteres med et sæt i henholdsvis højre og venstre side, og sidder i en X-opstilling.	Visuel: Tester om der sidder 2 LED-sæt med 5 røde dioder i en X-opstilling på hhv. højre og venstre side af bilen.	Bilen opfylder kravet om LED-monteringen på bagsiden af bilen.	Der er på bagsiden af bilen monteret et sæt baglygter på hhv. højre og venstre side af bilen med 5 røde LED lys der skal forestille sig et X.	
3.2	Ved "bremselys" skal middelstrømmen gennem hvert LED sæt være 50 mA +/- 5 mA.	Vi sætter properne, fra et multimeter med en nøjagtighed på +/- 0,8%, på anoden og katoden for hvert LED-sæt, og mäter herefter, om strømmen er 50 mA.	Bilen opfylder kravet om strømstyrke.	Ved standsning opnår bremselyset en middelstrøm på 48 mA.	

3.3	Ved ”almindeligt baglys” skal middelstrømmen gennem hvert LED sæt være 10 mA +/- 2 mA.	Ved hjælp af multimeter med en nøjagtighed på +/- 0,8%, måles middelstrømmen gennem hver LED-sæt til at være 10mA.	Multimeteret med en nøjagtighed på +/- 0,8% viser en værdi på 10mA.	Ved almindeligt baglys opnår baglyset en middelstrøm på 10,08 mA.	Se eventuelt afsnit med hardwaredesign af baglys
4.1	Startlyd / start metodi	Når bilen tændes/bilens kode bliver kørt, så starter den med at afspille en 4 sekunder trommeslageren, Når denne trommeslageren er færdig, begynder bilens hjul at køre i om-drejninger.	Når der tændes for bilen/bilen starter, afspilles en 4 sekunders trommeslageren, og derefter begynder bilen at køre.	Der bliver afspillet startlyden trommeslageren ved tænding af bilen.	
4.2	Refleksbrik Lyd	Når bilen kører forbi de forskellige R80-reflekser i løbet af banen, så afspilles en lyd af en bil, som gasser op (gas). Denne lyd afspilles hver gang bilen forbipasserer reflekserne ved hjælp af bilens monteret detektorer.	Når bilen kører forbi de forskellige reflekser på banen, så afspilles en lyd af en bil, som gasser op, som gruppen har valgt, og dette udføres ved alle R80- reflekserne igennem banen, på nær ved slutningen, hvor slutlyden trommeslageren afspilles.	Der bliver ved passering af hver refleksbrik afspillet lyden af en bil som gasser op.	
4.3	Slutlyd / slutmetodi	Når bilen når målstregen, og kommer til den sidste R80-refleks på banen, så afspilles en slutlyd trommeslageren. Dette gøres ved at bilen ved, hvor mange reflekser der er kommet forbi forinden, og derefter skifter	Når bilen når målstregen, samt den sidste R80-refleks på banen, så afspilles en slutlyd trommeslageren, ved hjælp af et stykke kode, som KUN udføres ved dette stade af banen.	Ved passering af sidste refleksbrik udføres der en delay hvorefter der efter et sekund bliver afspillet slutlyden.	

		over til en slags “slut-løkke,” som udfører noget kode KUN ved slutningen, som er med til at afspille den lyd.			
5.1	Kør op ad en bakke på 1 meter med hældning 30 grader.	Bilen placeres ved bunden af en bakke på 1 meter, som har en hældning på 30 grader. Derefter startes bilen, og der observeres om bilen har nok spænding til at kunne køre op ad bakken og nå toppen.	Bilen starter ved bunden ad bakken på 1 meter med en hældning på 30 grader, og kører derefter op ad bakken indtil at toppen nås.	Bilen kan uden problemer kører op ad en bakke med en hældning på mindst 30 grader.	

Tabel 14: Accepttestspezifikation for ikke-funktionelle krav

Konklusion (Mads)

Konklusion for hardwaredelen

Vi kan ud fra hardwaredelen konkludere, at alt hardwaren, som er monteret på bilkarrosseriet, opfylder kravene, og virker. I gruppen er de forskellige stykker hardware blevet fordelt ud på de forskellige gruppemedlemmer, hvor Anders og David har stået for motoren, August for lyden/SOMO, Nahom for forlys, Mads for baglys og Anders og Mads for sensor. Hvert gruppemedlem har haft til ansvar, at deres hardware har overholdt kravene, og har været fuldt funktionel for, at bilen i sidste ende har kunne fungere og fuldende banen.

Konklusion for softwaredelen

Vi kan ud fra hardwaredelen konkludere, at alt softwaren, som skal styre hardwaren monteret på bilkarrosseriet, fungerer. Softwaren består af en main.c fil, som er den fil, som bliver læst af mikrokontrolleren, så snart den tændes, og hvor den derefter udfører sine forskellige handlinger på de forskellige hardwarekomponenter ud fra, hvor den befinner sig på banen. Denne main.c fil indeholder adskillige header- og funktionsfiler for de forskellige hardwarekomponenter, som gør, at de kan styres inde i main-filen. Disse header- og funktionsfiler er blevet udarbejdet af hvert gruppemedlem ud fra viden fra kurserne/fagene Mikrocontroller Systemer og Objektorienteret Programmering, som er kurser/fag på første semester.

Konklusion for projektet

Vi kan ud fra projektet konkludere, at der med viden fra de forskellige kurser/fag på 1. semester kan udvikles hardware og software, som sammen ender ud i en velfungerende elektronisk bil. Udeover pensum på de forskellige kurser er der også blevet udført 4 forskellige øvelser i E-lab, der har været med til at hjælpe med

forarbejdet på projektet, og har givet en bedre forståelse for bla. PWM-signaler sammenspil med en DC-motor. Alle stillede krav i projektet opfyldes for både hardware og software. Alle gruppemedlemmer har udarbejdet og fuldendt deres tildelt del af projektet, og tidsplanen er blevet overholdt.

Individuelle konklusioner

Mads

Jeg kan ud fra projektet individuelt konkludere, at jeg har fået et stort udbytte fra alle dele, som projektet har indeholdt.

Jeg har lært, hvordan man arbejder sammen med andre, hvor hver person har haft sin egen individuelle opgave ift. at fuldende et endeligt fysisk produkt samt rapport. Jeg har også lært, hvordan man håndtere forskellige gruppemedlemmers personligheder og arbejdsmoraler ift. at nå forskellige deadlines, og hvor vigtigt det er, at have en klar aftale om, hvornår noget specifikt arbejde er færdiggjort.

Både hardware og softwaremæssigt har jeg lært virkelig meget, og jeg har også lært, hvordan man håndterer, når noget ikke virker, og hvordan man ved hjælp af fejlfindingstests kan komme frem til endelig at få noget til at virke.

Semesterprojektet på 1. semester har været bygget op omkring alle kurserne/fagene på 1. semester, hvor man har fået lov til at benytte sig af den viden, som man har fået fra de forskellige individuelle kurser/fag. Personlig synes jeg, at det har været en vildt god læringsmetode, da jeg altid har været sådan en person, som lærer mere ved at få det i hænderne sammenlignet med bare at have det skrevet på papir.

David

Jeg har ud fra dette projekt fået en del udbytte fra alle aspekter i projektet. Jeg har især lært en masse om hvordan det er at arbejde i en gruppe og hvordan man skal agere med de forskellige typer mennesker vi har i gruppen. Også hvordan der gøres brug af hinandens styrker og svagheder. Dette er noget jeg ikke har haft meget erfaring med fra gymnasiet.

Samtidig har jeg lært en masse fra hardware-delen og har under projektet stødt på en masse småfejl under softwaren, som jeg håber ikke støder på i fremtidige opgaver.

Projektet har efter min mening haft en god rød tråd sideløbende med kurserne på studiet og sammen med vejleder, der har været med til at give en følelse af personligudvikling og fremskridt.

Nahom

Som studerende har jeg haft en givende oplevelse med dette projekt. Gennem gruppearbejdet har vi kunnet styrke vores samarbejde og opdage hinandens individuelle evner på en bedre måde.

Projektet har haft en betydelig faglig indvirkning ved at få os til at dykke dybt ned i et specifikt emne både inden for software og hardware. Dette har skabt en mere omfattende forståelse af, hvordan software og hardware arbejder sammen i et gruppeprojekt, når man udvikler et produkt.

Vejledningsmødrene har også været meget gode til at besvare de spørgsmål, der er opstået undervejs og hjulpet os med at finde en struktur. Alt i alt har dette projekt været en yderst lærerig samt spændende oplevelse, der har fungeret som et godt supplement til den undervisning vi har fået hen over semesteret.

August

Overordnet set har projektet givet god forståelse for analyse af elektriske kredsløb, udarbejdelse af kredsløb efter diagram, microcontrollers og programmering af hardware i C. Projektet var delt op i dele, sådan at man nemt kunne have fokus på sin egen del, men med den overordnede sammenhæng med det færdige produkt, gav det også forståelse for effektivt gruppearbejde. Problemer opstod dog hurtigt ved fejlfinding, da alle skulle have en hånd med i spillet for at selv den mindste fejl kunne udbedres.

At arbejde frit i en større gruppe har altså også givet sine egne udfordringer. Alle arbejder ikke ens, og da man altid vil opnå optimal arbejdskraft fra alle, kræver det en del planlægning. Her har Insight-profilerne været en hjælp. Det har både hjulpet at kende andre, men personligt har det hjulpet mest at kende sine egne svagheder. At vide at jeg har en tendens til at være meget indadvendt har givet mig fokus på at bidrage mere til møder. Jeg har været god til at bidrage med at få mine dele lavet godt og hurtigt med god forståelse for processen. Dog vil jeg gerne se mig selv blive bedre til at kommunikere, samarbejde og møde mere velforberedt op.

Anders

Jeg har som studerende følt at jeg har fået et godt udbyttet fra projektet. Meget af den erfaring kommer af at der meget sjældent var succes ved første forsøg. Nok nærmere allersidste forsøgt, så selvom det har været surt har jeg fået en masse viden og ekspertiser.

Gruppe arbejde aspektet, har været meget nyt for mig på den her måde. Hvor man har en leder og roller, syntes det har fungeret markant bedre. Kontra hvad jeg har oplevet på mit gymnasium. Jeg har os fået brugt insight profiler til at, finde ud af hvordan man samarbejde bedste med dem fra gruppen. Når jeg undere mig over ting.

Så helt konkret har det været en meget god oplevelse, og glæder mig til at lærer mere de andre år. Også selvom der er blevet brugt utallige timer på fejltæller og bil som bare lever deres eget liv og gør hvad de har lyst til selvom det bare er et stykke hardware, kunstig intelligens??

Bilagoversigt

Bilag 1: Konkurrencen

Bilag 2: Fysiske mål for karrosseriet

Bilag 3: Tidsplan

Bilag 4: Samarbejdsaftale

Bilag 5: Mødeindkaldelser + referater

Bilag 6: Main.c

Bilag 7: Lysdiode-øvelse (øvelse 3) - PRJ1

Bilag 7: Lysdiode

Bibliografi

finder. (u.d.). *PRODUCT SPECIFICATION 40.52 - 2 POLE 8 A (5 MM PIN PITCH)*. Hentet fra

<https://brightspace.au.dk/content/enforced/90159->

LR19026/09.012.0005.pdf?_d2lSessionVal=NH26AgZGxuASWL4iuv1Qkeb8m&_d2lSessionVal=NH26AgZGxuASWL4iuv1Qkeb8m&_d2lSessionVal=OUVx4Yw84IOxAv7pd2benIWW3&_d2lSessionVal=n8gqRiKpv2IrD8S0Ix7HBYeN

FORYARD. (u.d.). *PRODUCT SPECIFICATION Model No. FYL-5013HD1C*. Hentet fra

<https://www.tme.eu/Document/c1c1a1e6a3d907fd779ee5ed0172e6ba/FYL->

5013HD1C.pdf?fbclid=IwAR3QAImlHpqOz4xZJsgv3ErzzX7-

DiPlKZnRMRAuGMAegAqJ8aWFBAhogKo

SOMO-II Datablad:

https://brightspace.au.dk/content/enforced/90159-LR19026/SOMO-II_datasheet_R_1_4.pdf