

Pokaljagten

TNT

Gruppe 5 med vejleder Henning Hargaard

Afleveret 23. Juni 2014

1. Semesterprojekt

Elektro, IKT, Stærkstrøm

Ingeniørhøjskolen i Aarhus



Youssef Hussein

201270047

Mohammad Rasuli

11339

Kasper Kjær

201206225

Philip Krogh-Pedersen

201311473

Henrik Bagger Jensen

201304157

Morten Nørgaard

201371179

Indhold

1. Problemformulering	5
1.1. Aktør og kontekst diagram	6
2. Kravspecifikation	7
2.1. Funktionelle krav	7
2.2. Ikke funktionelle krav	9
2.3. Godkendelsesformular	11
3. Systemarkitektur	12
3.1. Hardware	12
3.1.1. Definitionsliste	12
3.1.2. IBD-diagram	12
3.2. Software	15
4. Design	20
4.1. Timer	20
4.1.1. Software	20
4.2. Baglys	24
4.2.1. Hardware	24
4.2.2. Software	27
4.3. Detektorer	28
4.3.1. Hardware	28
4.4. Forlys	29
4.4.1. Hardware	29
4.4.2. Software	32
4.5. Lyd	34
4.5.1. Software	34
4.6. Motorstyring	35
4.6.1. Hardware	35
4.6.2. Software	37
4.7. Strømstyring	39
4.7.1. Hardware	39
4.8. Main	40
4.8.1. Software	40
5. Accepttest	41
5.1. Accepttestspecifikation for funktionelle krav (Use Cases)	41

5.2.	Accepttestspecifikation for ikke-funktionelle krav (Use Cases)	44
6.	Diskussion	47
6.1.	Generelt.....	47
6.2.	Software	47
6.3.	Hardware	47
7.	Konklusion	47
7.1.	Konklusion – Gruppe	47
7.2.	Konklusion – Philip.....	47
7.3.	Konklusion – Henrik.....	48
7.4.	Konklusion – Rasuli	48
7.5.	Konklusion – Youssef	48
7.6.	Konklusion – Kasper	48
7.7.	Konklusion – Morten	49

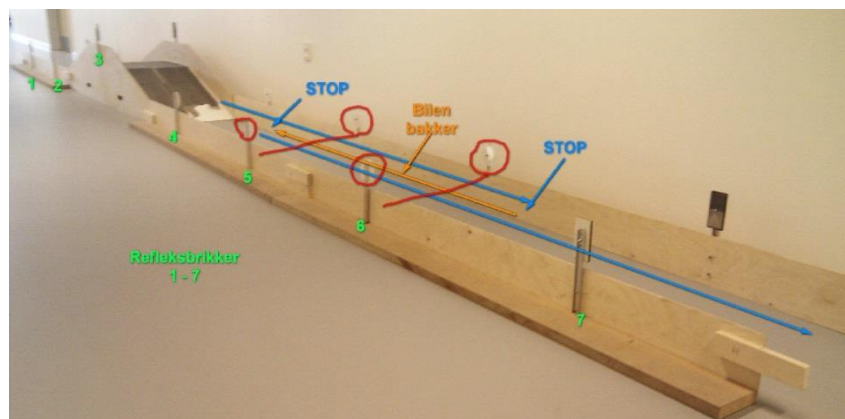
1. Problemformulering

Det system, der skal udvikles, er en elektrisk bil som vist nedenunder.



Figur 1 - Elektrisk bil med printplader og accelerationsmåler

Bilens formål er at gennemkøre denne bane med færrest mulige point.



Figur 2 - Bane med reflektorer

Bilen skal forsynes med fremadrettet kørellys (hvidt) og baglys (rødt), samt bremselys (kraftigt rødt). Bremselys og baglys udsendes fra samme kilde og kan reguleres i styrke.

Hvert enkelt for-, bag- eller bremselys kan realiseres som et antal lysdioder, der opbygges som et sæt. Hvert sæt kan opbygges som en passende kombination af serie- og/eller parallel-forbindelser.

I projektet anvendes microcontrolleren Mega32 monteret på et STK500 kit.
Denne anvendes til:

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

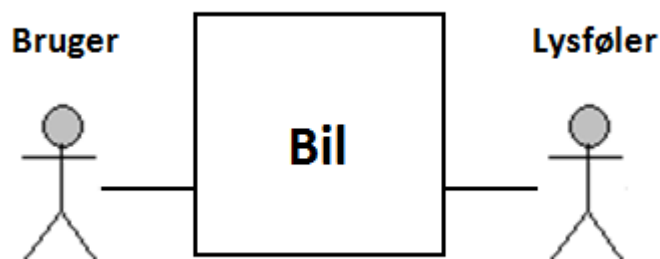
Bilen skal køre på en bane med bander, hvor banderne vil sikre at bilen bliver på banen. Banderne udstyres med 7 par refleksbrikker, der kan detekteres af optiske proximity sensorer placeret på bilen.

De 7 par refleksbrikker er placeret parvist overfor hinanden på hver sin side af banen i positionerne:

- 1) 0,5 m fra start
- 2) 0,5 m før bakke
- 3) På toppen af bakken
- 4) 0,5 m efter bakken
- 5) 1,5 m efter bakken
- 6) 2,5 m efter bakken
- 7) 0,5 m før mål

Refleksbrikkernes vertikale center placeres 30 mm højere end bilens monteringsplade.

1.1. Aktør og kontekst diagram



Figur 3 - Aktør-kontekst diagram

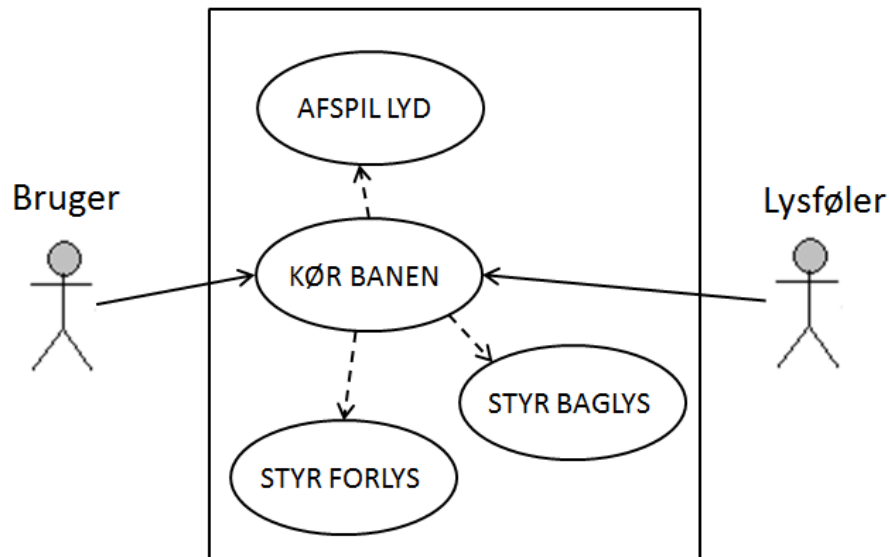
Det system, som skal udvikles, er selve den elektriske bil.

Aktøren "Bruger" kan få bilen til **at starte et gennemløb af banen**. Denne aktør er primær.

Aktøren "Lysføler" registrer, hver gang en refleksbrik passeres. Denne aktør er sekundær.

2. Kravspecifikation

2.1. Funktionelle krav



Figur 4 - Use case diagram

Definitionsliste

- **Belastning af motor:** En middelspænding hen over bilens motor på over 0 volt.
- **Konstant belastning:** Middelspændingen hen over motoren er konstant og over 0 volt. Det er tilladt for middelspændingen, at stige eller falde op til 0,5 volt inden for et sekund.
- **Belastning øges:** Middelspændingen hen over motoren vokser mere end 0,5 volt inden for et sekund.
- **Belastning mindskes:** Middelspændingen hen over motoren falder med mere end 0,5 volt inden for et sekund.
- **Almindeligt baglys:** Et konstant lys. (Se punkt 4.3 under ikke funktionelle krav).
- **Bremselys:** Et konstant lys, der er 4-5 gange kraftigere end almindeligt baglys. (Se punkt 4.4 under ikke funktionelle krav).

1. Use case 1: "Kør banen"

Mål

Denne Use case beskriver gennemkørsel af en bane.

Initieres af: Bruger.

Normalt scenarie

- 1.1. Brugeren tænder for bilen og placerer den, så den kan køre forlæns ind på banen. En specifik "startmelodi" afspilles, hvorefter bilen begynder at køre (Use case 2).
- 1.2. Når bilen har passeret refleksbrik nummer 6, standser bilen inden refleksbrik nummer 7 nås.
- 1.3. Bilen bakker, indtil refleksbrik nummer 5 er passeret, og standser inden refleksbrik nummer 4 nås.
- 1.4. Bilen kører forlæns, indtil refleksbrik nummer 7 nås.
- 1.5. Bilens bringes til standsning i målområdet, der er defineret som 0,5-2,0 meter efter refleksbrik nummer 7.

2. Use case 2: "Afspil lyd"

Definitionsliste

- **Startmelodi:** Udsnit af AC/DC med TNT (lead guitar). Demolyd 1¹ er vedlagt.
- **Slutmelodi:** Udsnit af omkvæd fra TNT af AC/DC (lead guitar). Demolyd 2² er vedlagt.
- **Refleksbriklyd:** En ca. et sekunds lyd. Demolyd 3³ er vedlagt.

Mål

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

Initieres af: Use case 1 "Kør banen".

Normalt scenarie

- 2.1. Når bilen aktiveres afspilles "startmelodi".
- 2.2. Hver gang en refleksbrik passeres, afspilles "refleksbriklyd".
- 2.3. Når refleksbrik nummer 7 er passeret og bilen holder stille, afspilles "slutmelodi".

¹ <https://dl.dropboxusercontent.com/u/10305972/Startmelodi.wav>

² <https://dl.dropboxusercontent.com/u/10305972/Slutmelodi.wav>

³ <https://dl.dropboxusercontent.com/u/10305972/Refleksbriklyd.wav>

3. 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

3.1. Forlyset skal være tændt, så længe bilen er i bevægelse, ellers skal forlyset være slukket.

4. Use case 4: "Styr baglys"

Mål

Denne Use case beskriver styringen af bilens indbyggede baglys.

Baglyset kan være slukket, lyse med almindeligt baglys eller med et kraftigere bremselys.

Initieres af: Use case 1 "Kør banen".

Normalt scenarie

4.1. Baglys skal være slukket når motoren ikke er belastet.

4.2. Så længe bilens motor belastes konstant eller belastningen øges, tændes bilens almindelige baglys.

4.3. Hvis belastningen på bilens motor mindskes tændes bremselys.

2.2. Ikke funktionelle krav

1. Generelle krav

1.1. Middelværdien af forsyningsspændingen til bilens motor skal styres via pulsbreddemodulation (PWM) fra 0 til maksimal batterispænding (7.2V +/- 0.5V) i 100 trin.

1.2. På bilens højre og venstre side skal placeres detektorer, der som minimum kan registrere en R80 refleksbrik i afstanden fra 2 cm til 25 cm.

1.3. Print til detektorer skal have en bredde på 100 mm +/- 1 mm.

2. Bilens lyd

2.1. Frekvensen af tonerne til afspilning gennem bilens højttaler skal styres med et firkantsignal.

Startmelodi består af: E3 - G3 - A3.

Refleksbriklyd består af: E4 - G4 - A4.

Slutmelodi består af: E3 - G3 - A3 - B3 - D4 - E4 - G4 - A4 - B4.

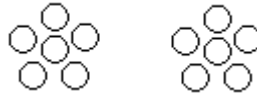
Frekvensen for disse toner er givet i Tabel 1 herunder:

Tone	E3	G3	A3	B3	D4	E4	G4	A4	B4
Frekvens Hz s ⁻¹	164,81 (±5 Hz)	196,00 (±5 Hz)	220,00 (±5 Hz)	246,94 (±5 Hz)	293,66 (±5 Hz)	329,63 (±5 Hz)	392,00 (±5 Hz)	440,00 (±5 Hz)	493,88 (±5 Hz)

Tabel 1 - Toners Frekvens

3. Bilens forly

3.1. Forlyset skal bestå af to sæt af seks stk. hvide (farvekode: InGaN⁴, White) LED dioder (Figur 5).



Figur 5 - Forlys på bilen, bestående af 2 sæt af 6 stk. hvide LED dioder

3.2. Implementeres med to hvide LED-sæt, der monteres med et sæt i henholdsvis højre og venstre side forrest på bilen.

3.3. Når forlyset er tændt, skal hver LED-diode lede en middelstrøm på $30 \text{ mA} \pm 1 \text{ mA}$.

4. Bilens baglys

4.1. Baglyset skal bestå af to sæt af seks stk. røde (farvekode: GaP⁵, Red) LED dioder (Figur 6).



Figur 6 - Baglys på bilen, bestående af 2 sæt af 6 stk. røde LED dioder

4.2. Implementeres med to røde LED-sæt, der monteres med et sæt i henholdsvis højre og venstre side forrest på bilen.

4.3. Når "almindeligt baglys" er aktiveret, skal hver LED-diode, der indgår i de førømtalte LED-sæt, lyse svarende til en middelstrøm på $3 \text{ mA} \pm 1 \text{ mA}$.

4.4. Når "bremselyset" er aktiveret, skal hver LED-diode, der indgår i de førømtalte LED-sæt, lyse svarende til en middelstrøm på $15 \text{ mA} \pm 1 \text{ mA}$.

Accepttestspecifikation findes under 5. Accepttest.

⁴ http://en.wikipedia.org/wiki/Indium_gallium_nitride

⁵ http://en.wikipedia.org/wiki/Gallium_phosphide

2.3. Godkendelsesformular

Forfatter(e):	Gruppe 5 (Ansvarlig: Philip Krogh-Pedersen)
Godkendes af:	Philip Krogh-Pedersen(intern), Gruppe 1(ekstern)
Projektnummer:	E1PRJ1
Kunde:	Henning Hargaard

Ved underskrivelse af dette dokument accepteres det af begge parter, som værende kravene til udviklingen af det ønskede system.

Sted og dato:

Henning Hargaard

Philip Krogh-Pedersen

3. Systemarkitektur

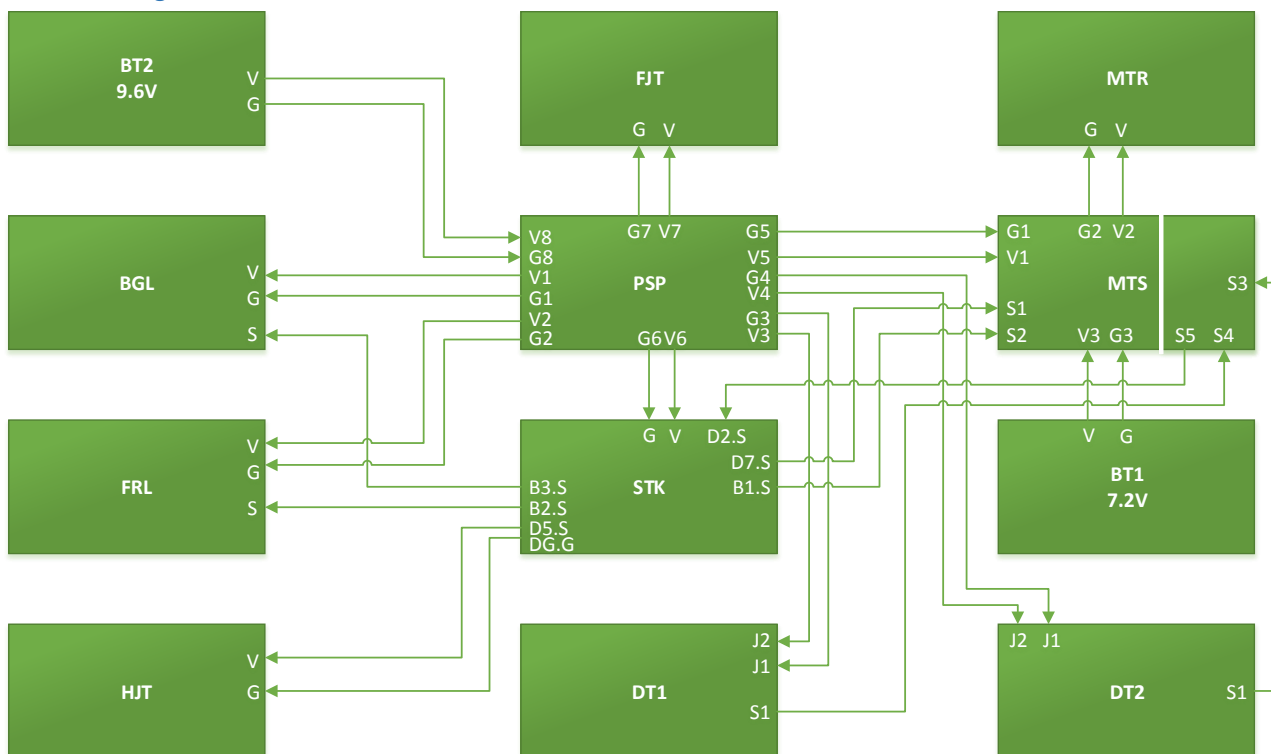
3.1. Hardware

Herunder vil systemarkituren for hardware systemerne, der skal indgå i bilen, uddybes. De eksterne komponenter er færdiglavede komponenter der bliver udnyttet i dette projekt. Disse komponenter vil derfor ikke nødvendigvis blive beskrevet yderligere under denne rapport.

3.1.1. Definitionsliste

- PSP - Strømforsyning
- BT1 - 7,2 V Batteri (Ekstern komponent)
- BT2 - 9,6 V Batteri (Ekstern komponent)
- BGL - Baglys
- DT1 - Detektor 1 (Ekstern komponent, se bilag 051/052)
- DT2 - Detektor 2 (Ekstern komponent, se bilag 051/052)
- FJT - Fejltæller
- FRL - Forly
- HJT - Højttaler
- MTS - Motorstyring
- MTR - Motor
- STK - STK-500

3.1.2. IBD-diagram



Figur 7 – IBD diagram over HW – Porte med "G" til sidst i navnet er stel. Porte med "V" til sidst i navnet er en form for strømforsyning. Porte med "S" til sidst i navnet, er en form for signal. De to detektorer er pre-definerede, derfor er portene navngivet anderledes. Se Tabel 2 og Tabel 3 for flere detaljer.

Bloknavn	Funktionsbeskrivelse	Signaler	Kommentar
Baglys	Lyse på 2 styrker, samt slukket	0,0 V 5,0 V Signal (5 V)	Reference Strømforsyning PWM indgangssignal
Batteri 1	Forsyne kredsløb med 7,2 V spænding	0,0 V 7,2 V	Reference Strømforsyning
Batteri 2	Forsyne kredsløb med 9,6 V spænding	0,0 V 9,6 V	Reference Strømforsyning
Detektor 1	Detektere refleksbrikker	0,0 V 5,0 V Signal	Reference Strømforsyning On/Off udgangssignal
Detektor 2	Detektere refleksbrikker	0,0 V 5,0 V Signal	Reference Strømforsyning On/Off udgangssignal
Fejltæller	Skal tælle og anvise fejl	0,0 V 9,6 V	Reference Strømforsyning
Forlys	Lyse og slukket	0,0 V 5,0 V Signal (5 V)	Reference Strømforsyning On/Off indgangssignal
Højtaler	Afspille melodi	0,0 V Signal (5 V)	Reference PWM indgangssignal
Motor	Skabe fremdrift	0,0 V Signal (7,2 V)	Reference PWM indgangssignal
Motorstyring	Sende PWM signaler til bilens motor, således retning og hastighed styres.	0,0 V 7,2 V Signal (5 V) Signal (7,2 V) Signal (5 V)	Reference Batteri 1 (indgang) PWM indgangssignal PWM udgangssignal On/Off indgangssignal
Strømforsyning	Leverer strøm til kredsløbet	0,0 V 5,0 V 9,6 V 9,6 V	Reference (stel) Strømforsyning (udgang) Batteri 2 (indgang) Strømforsyning (udgang)
STK-500	Styre kredsløbene	0,0 V 9,6 V Signal Signal Signal Signal Signal Signal Signal Signal	Reference (stel) Strømforsyning On/Off udgangssignal (FRL) PWM udgangssignal (BGL) PWM udgangssignal (MTR) PWM udgangssignal (HJT) Reference (HJT) On/Off indgangssignal (DT1) On/Off indgangssignal (DT2) On/Off udgangssignal (MTR)

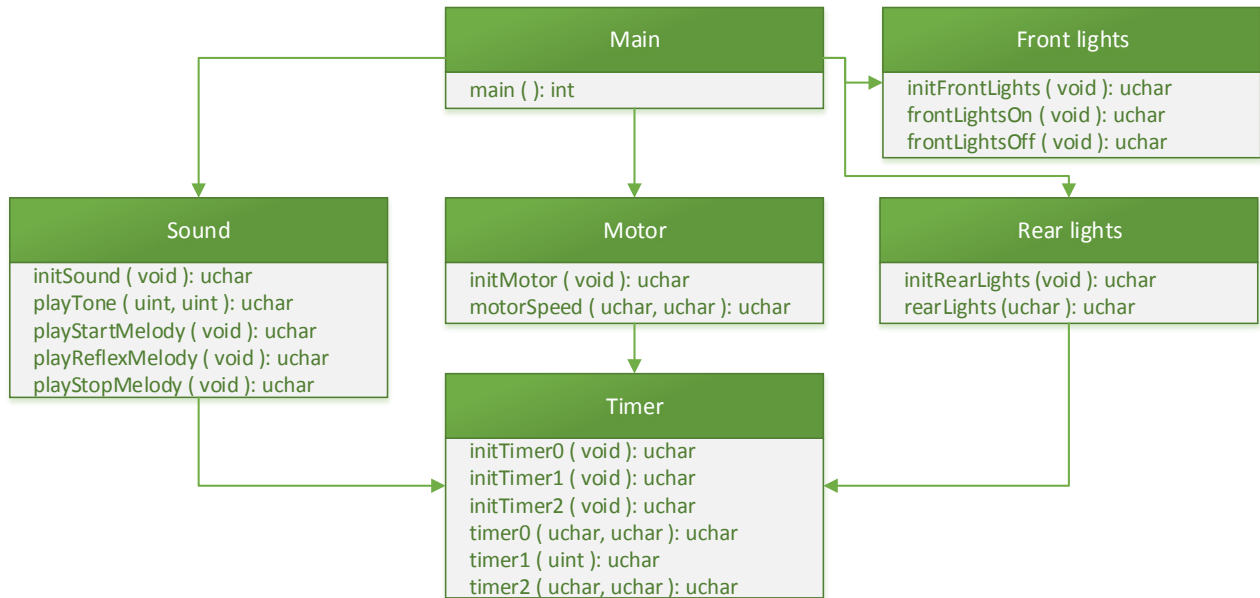
Tabel 2 – Block beskrivelse tilhørende Figur 7 IBD diagram over HW

Signal-navn	Funktion	Område	Port 1	Port 2	Kommentar
0 V	Stel	0,0 V	PSP.G1 PSP.G2 PSP.G3 PSP.G4 PSP.G5 PSP.G6 PSP.G7 BT2.G BT1.G MTS.G2 STK-DG.G	BGL.G FRL.G DT1.J1 DT2.J1 MTS.G1 STK.G FJT.G PSP.G8 MTS.G3 MTR.G HJT.G	
5 V	Strømforsyning	4,9-5,1 V	PSP.V1 PSP.V2 PSP.V3 PSP.V4 PSP.V5	BGL.V FRL.V DT1.J2 DT2.J2 MTS.V1	
7,2 V	Strømforsyning	7,1-7,3 V	BT1.V MTS.V2	MTS.V3 MTR.V	PWM
9,6 V	Strømforsyning	9,5-9,7 V	BT2.V PSP.V6 PSP.V7	PSP.V8 STK.V FJT.V	
Signal	Signaler til styring af hardware enheder.	0-5 V	DT1.S1 DT2.S1 MTS.S5 STK-B3.S STK-B2.S STK-D5.S STK-D7.S STK-B1.S	MTS.S4 MTS.S3 STK-D2.S BGL.S FRL.S HJT.S MTS.S1 MTS.S2	On/Off On/Off On/Off PWM On/Off PWM PWM On/Off

Tabel 3 – Signalbeskrivelse af Figur 7 IBD diagram over HW

3.2. Software

Herunder vil systemarkitekturen for software systemerne, der skal indgå i bilen, uddybes.



Figur 8 – Blokdiagram over SW (uchar er en forkortelse for unsigned char, uint er en forkortelse for unsigned int)

Blok: Rear lights**Ansvar:** Driverens ansvar er at tænde baglysende og styre bremselyset.**Funktioner:**

unsigned char initRearLights (void)

Parametre: IngenReturværdi: True (1) hvis den ønskede initialisering er kørt.
False (0) hvis der skete en fejl under initialiseringen.Beskrivelse: Initialiserer baglys.

unsigned char rearLights (unsigned char)

Parametre: Den ønskede duty cycle(0-100).Returværdi: True (1) hvis den ønskede funktion er kørt.
False (0) hvis der skete en fejl under kørslen.Beskrivelse: Styrer baglysets lysstyrke.**Blok:** Front lights**Ansvar:** Driveren styrer forlysets tilstand, der kan være hhv. tændt eller slukket.**Funktioner:**

unsigned char initFrontLights (void)

Parametre: IngenReturværdi: True (1) hvis den ønskede initialisering er kørt.Beskrivelse: Initialiserer forlys.

unsigned char frontLightsOn (void)

Parametre: Ingen.Returværdi: True (1) hvis den ønskede funktion er kørt.Beskrivelse: Funktionen tænder forlys.

unsigned char frontLightsOff (void)

Parametre: Ingen.Returværdi: True (1) hvis den ønskede funktion er kørt.Beskrivelse: Funktionen slukker forlys.

Blok: Motor

Ansvar: Driveren skal styre motorens hastighed og retning.

Funktioner:

unsigned char initMotor (void)

Parametre: Ingen

Returværdi: True (1) hvis den ønskede initialisering er kørt.

False (0) hvis der skete en fejl under initialiseringen.

Beskrivelse: Initialiserer motor.

unsigned char motorSpeed (unsigned char, unsigned char)

Parametre: 1) Den ønskede duty cycle(0-100), som spændingen over motoren skal have.

2) Hvilken retning bilen skal køre (0=frem, 1=tilbage).

Returværdi: True (1) hvis den ønskede funktion er kørt.

False (0) hvis der skete en fejl under kørsel.

Beskrivelse: Styrer bilens retning og hastighed.

Blok: Sound**Ansvar:** Driveren skal styre bilens højttaler.**Funktioner:**

unsigned char initSound (void)

Parametre: IngenReturværdi: True (1) hvis den ønskede initialisering er kørt.
False (0) hvis der skete en fejl under initialiseringen.Beskrivelse: Initialiserer lyd.

unsigned char playTone (unsigned int, unsigned int)

Parametre: 1) Den ønskede længde af tone i millisekunder.
2) Den ønskede frekvens for tone (antal puls per sekund).Returværdi: True (1) hvis den ønskede funktion er kørt.
False (0) hvis der skete en fejl under kørsel.Beskrivelse: Tester bilens højttaler.

unsigned char playStartMelody (void)

Parametre: IngenReturværdi: True (1) hvis den ønskede funktion er kørt.
False (0) hvis der skete en fejl under kørsel.Beskrivelse: Afspiller startmelodi over bilens højttaler.

unsigned char playReflexMelody (void)

Parametre: IngenReturværdi: True (1) hvis den ønskede funktion er kørt.
False (0) hvis der skete en fejl under kørsel.Beskrivelse: Afspiller refleksbriklyd over bilens højttaler.

unsigned char playStopMelody (void)

Parametre: IngenReturværdi: True (1) hvis den ønskede funktion er kørt.
False (0) hvis der skete en fejl under kørsel.Beskrivelse: Afspiller slutmelodi over bilens højttaler.

Blok: Timer

Ansvar: Driverens ansvar er at initialisere ATmega32'ens indbyggede timere. Samt kunne sætte disse timere til at køre et ønsket PWM-signal.

Funktioner:

unsigned char initTimer0 (void)

Parametre: Ingen

Returværdi: True (1) hvis den ønskede initialisering er kørt.

Beskrivelse: Funktionen skal initialisere Timer0 til fast PWM.

unsigned char initTimer1 (void)

Parametre: Ingen

Returværdi: True (1) hvis den ønskede initialisering er kørt.

Beskrivelse: Funktionen skal initialisere Timer1 til CTC.

unsigned char initTimer2 (void)

Parametre: Ingen

Returværdi: True (1) hvis den ønskede initialisering er kørt.

Beskrivelse: Funktionen skal initialisere Timer2 til fast PWM.

unsigned char timer0 (unsigned char, unsigned char)

Parametre: 1) Den ønskede duty cycle (0-100).

2) Den ønskede pre-definerede frekvens (antal puls per sekund).

Returværdi: True (1) hvis den ønskede kode er kørt.

False (0) hvis der er givet en forkert parameter.

Beskrivelse: Funktionen skal sætte Timer0 til et PWM signal.

unsigned char timer1 (unsigned int)

Parametre: Den ønskede frekvens (antal puls per sekund).

Returværdi: True (1) hvis den ønskede kode er kørt.

False (0) hvis der er givet en forkert parameter.

Beskrivelse: Funktionen skal sætte Timer1 til et CTC signal.

unsigned char timer2 (unsigned char, unsigned char)

Parametre: 1) Den ønskede duty cycle (0-100).

2) Den ønskede pre-definerede frekvens (antal puls per sekund).

Returværdi: True (1) hvis den ønskede kode er kørt.

False (0) hvis der er givet en forkert parameter.

Beskrivelse: Funktionen skal sætte Timer2 til et PWM signal.

4. Design

4.1. Timer

4.1.1. Software

Timer er en driver der bliver brugt af de tre andre drivere, Sound, Motor og Rear Lights, som set på Figur 8 under systemarkitektur for software. Formålet af driveren er at initialisere og styre de tre timere som ATmega32 chippen kan udnytte i dette projekt.

De tre timere skal bruges individuelt til forskellige hardwarekredsløb. Herunder er de seks funktioner delt op således at deres design kan forklares individuelt.

Timer
<pre>initTimer0 (void): uchar initTimer1 (void): uchar initTimer2 (void): uchar timer0 (uchar, uchar): uchar timer1 (uint): uchar timer2 (uchar, uchar): uchar</pre>

Figur 9 - Timer blok uddrag fra software blokdiagram (Figur 8)

initTimer0

Denne funktion skal initierer timer0 på en ATmega32 chip til et 8-bit fast PWM signal med udgang på Port B ben 3 (PB3), via STK500-kittet. Signalet skal clear (0V) OC0 på compare match, og set (5V) OC0 på TOP. Clockfrekvensen skal sættes til standard uden pre-scaler. Dette gøres gennem opslagsværker på Atmel Mega32 chippen.

initTimer1

Denne funktion skal initierer timer1 på en ATmega32 chip til et 16-bit CTC signal med udgang på Port D ben 5 (PD5), via STK500-kittet. Signalet skal toggle (0V/5V) OC1A på compare match. Clockfrekvensen skal sættes til standard uden pre-scaler. Dette gøres gennem opslagsværker på Atmel Mega32 chippen.

initTimer2

Denne funktion skal initierer timer2 på en ATmega32 chip til et 8-bit fast PWM signal med udgang på Port D ben 7 (PD7), via STK500-kittet. Signalet skal clear (0V) OC2 på compare match, og set (5V) OC2 på TOP. Clockfrekvensen skal sættes til standard uden pre-scaler. Dette gøres gennem opslagsværker på Atmel Mega32 chippen.

timer0

Denne funktion skal kunne variere frekvensen samt duty cyclen af timer 0. Timer 0 har fire forskellige pre-scalere da den er initialiseret til et fast PWM signal. Den kan selvfølgelig også sættes til ikke at bruge en scaler.

For at give brugeren af funktionen flest mulige muligheder for indstilling af både duty cycle og frekvens, kan koden opstilles som angivet på aktivitetsdiagrammet (Figur 10).

Udregninger af forskellige frekvenser kan ses herunder.

o - default ingen pre-scaler:

$$\frac{3686400}{1 \cdot 255} \text{ Hz} \approx 14456 \text{ Hz}$$

1 - pre-scaler 8:

$$\frac{3686400}{8 \cdot 255} \text{ Hz} \approx 1807 \text{ Hz}$$

2 - pre-scaler 64:

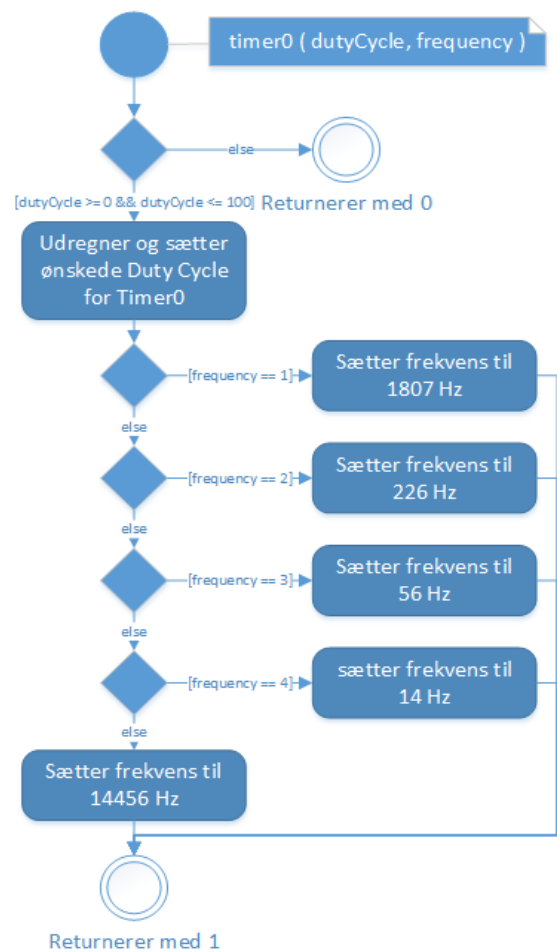
$$\frac{3686400}{64 \cdot 255} \text{ Hz} \approx 226 \text{ Hz}$$

3 - pre-scaler 256:

$$\frac{3686400}{256 \cdot 255} \text{ Hz} \approx 56 \text{ Hz}$$

4 - pre-scaler 1024:

$$\frac{3686400}{1024 \cdot 255} \text{ Hz} \approx 14 \text{ Hz}$$



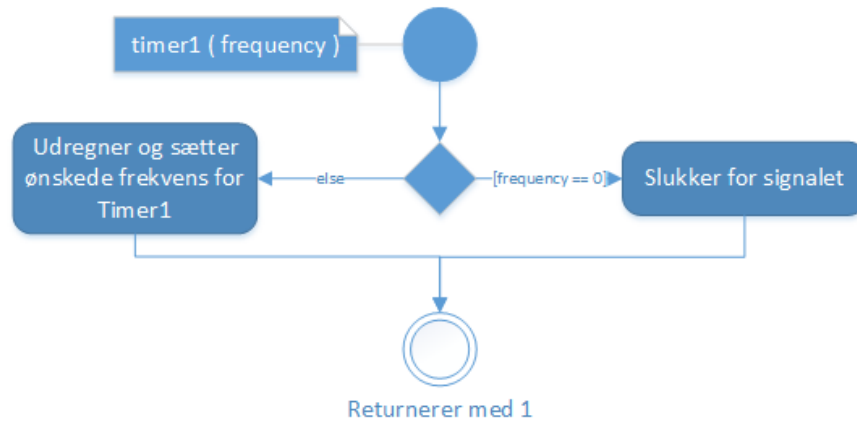
Figur 10 - Aktivitetsdiagram for timer0 funktionen

Beregningen af PWM signalets duty cycle har 8-bit registeret OCR0 indflydelse på. Funktionen for denne ser således ud:

$$OCR0 = \frac{(duty\ cycle \cdot 100)}{255}$$

timer1

Denne funktion skal kunne variere frekvensen for timer 1. Timer 1 er under initTimer1 sat til maksimal clockfrekvens der ikke skal røres ved da timer 1's opgave er at skabe et signal til at afspille lyd. Jo højere clockfrekvensen des større præcision på tonerne. Dog skal der tages højde for at den skal kunne slukke for lyden. Herunder er der et aktivitetsdiagram for timer1 funktionen (Figur 11).



Figur 11 - Aktivitetsdiagram over timer1 funktionen

Sound-driveren der skal udnytte timer1 har brug for at kunne beregne den ønskede frekvens af CTC signalet. Dette har 16-bit registeret ICR1 indflydelse på. Funktionen for denne beregning ser således ud:

$$ICR1 = \left(\frac{3686400}{2 \cdot \text{frekvens}} \right) - 1$$

timer2

Denne funktion skal kunne variere frekvensen samt duty cyclen af timer 2. Timer 2 har seks forskellige pre-scalere da den er initialiseret til et fast PWM signal. Den kan selvfølgelig også sættes til ikke at bruge en pre-scaler.

For at give brugeren af funktionen flest mulige muligheder for indstilling af både duty cycle og frekvens, kan koden opstilles som angivet på aktivitetsdiagrammet (Figur 12).

Udregninger af forskellige frekvenser kan ses herunder.

o - default ingen pre-scaler:

$$\frac{3686400}{1 \cdot 255} \text{ Hz} \approx 14456 \text{ Hz}$$

1 - pre-scaler 8:

$$\frac{3686400}{8 \cdot 255} \text{ Hz} \approx 1807 \text{ Hz}$$

2 - pre-scaler 32:

$$\frac{3686400}{32 \cdot 255} \text{ Hz} \approx 452 \text{ Hz}$$

3 - pre-scaler 64:

$$\frac{3686400}{64 \cdot 255} \text{ Hz} \approx 226 \text{ Hz}$$

4 - pre-scaler 128:

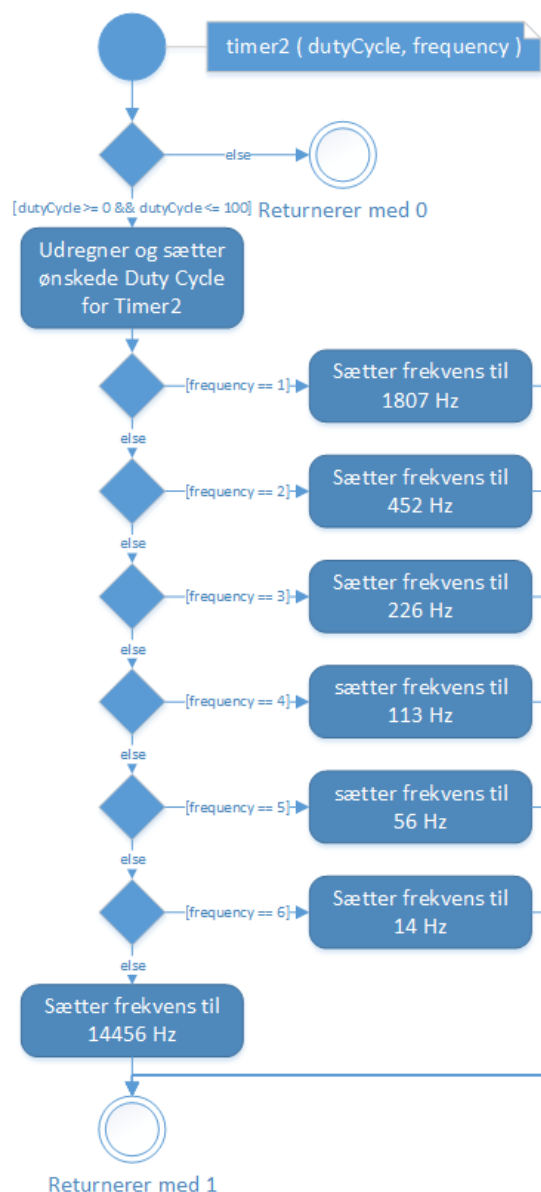
$$\frac{3686400}{128 \cdot 255} \text{ Hz} \approx 113 \text{ Hz}$$

5 - pre-scaler 256:

$$\frac{3686400}{256 \cdot 255} \text{ Hz} \approx 56 \text{ Hz}$$

6 - pre-scaler 1024:

$$\frac{3686400}{1024 \cdot 255} \text{ Hz} \approx 14 \text{ Hz}$$



Figur 12 - Aktivitetsdiagram over timer2 funktionen

Beregningen af PWM signalets duty cycle har 8-bit registeret OCR2 indflydelse på. Funktionen for denne ser således ud:

$$OCR2 = \frac{(duty\ cycle \cdot 100)}{255}$$

4.2. Baglys

4.2.1. Hardware

For at man kan styre LEDerne, anvendes en BJT (Bipolar Junction Transistor). I en BJT kan en meget lille strøm styre en stor strømændring gennem en transistor.

Med hensyn til kravspecifikation, skal almindeligt baglys lyse svarende til en middelstrøm på 36 mA.

$$12LED \cdot 3mA = 36mA$$

Bremselyset skal lyse svarende til en middelstrøm på 180 mA.

$$12LED \cdot 15mA = 180mA$$

Beregning

Ifølge databladet for LEDerne er spændingsfaldet over LEDerne på 1.85 V og tilsvarende for BD139 er spændingsfaldet for kollektor-emitter (V_{CE}) på 0.5 V. Når PWM duty cycle er på 100 % afgives en 5 V spænding. Derfor kan spændingen over modstanden (V_{R_LED}) beregnes til:

$$V_{R_LED} = 5V - 0.5V - 1.85V = 2.65V$$

Ifølge kravspecifikation, skal bremselyset igennem hver LED, lyse svarende til en middelstrøm på 15 mA. Derfor kan størrelse på LED modstanden (R_{LED}) beregnes til:

$$R_{LED} = \frac{2.65V}{0.015A} = 176.67\Omega$$

For at beregne modstanden, som er tilkoblet mellem PWM signal og Base. Skal Base-benet (I_B) beregnes.

Når BJT befinder sig i dets aktive område, har den en β i intervallet 50 - 400 og vi har derfor målt den valgte transistors β til at være 125. Derfor kan I_B beregnes til:

$$I_K = 12LED \cdot 15mA = 180mA = 0.18A$$

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

$$I_B = \frac{0.18A}{125} = 0.00144A = 1.44mA$$

I samme område er spændingen over V_{BE} tæt på 0.7 V.

Derfor kan spændingen over modstanden mellem PWM signal og Base (V_{R_B}) beregnes til:

$$V_{R_B} = 5V - 0.7V = 4.3V$$

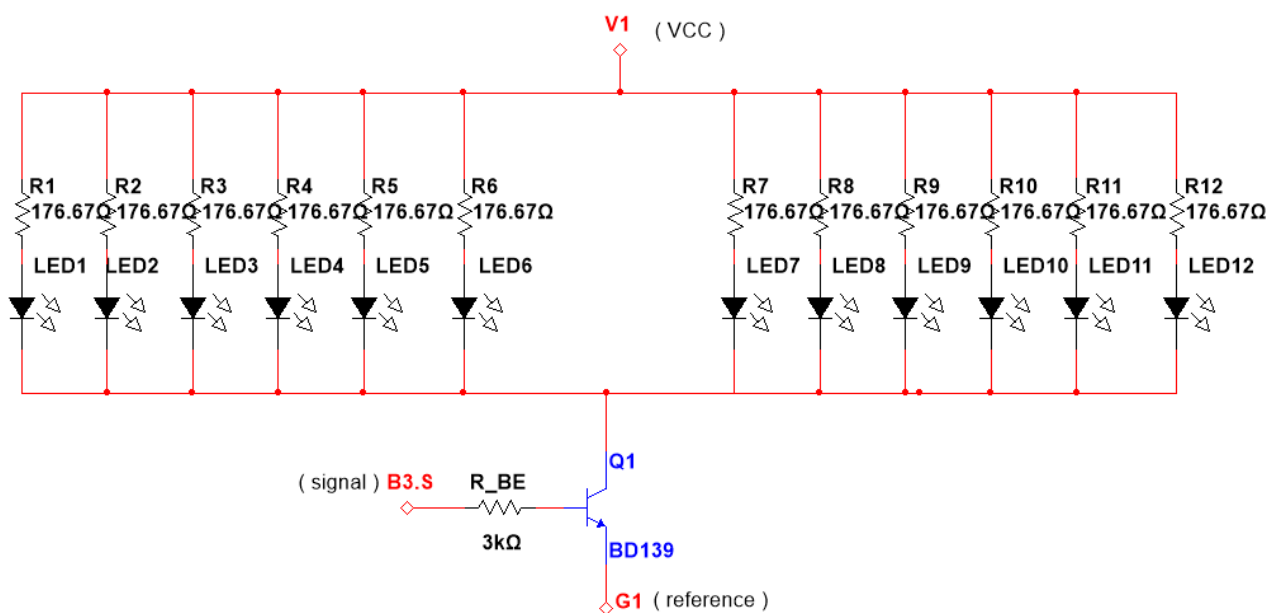
Derfor modstanden mellem PWM signal og Base beregnes til:

$$R_B = \frac{V_{R_B}}{I_B}$$

$$R_B = \frac{4.3V}{0.00144A} = 2986.1\Omega = 3.0k\Omega$$

Opstilling

På følgende opstilling (Figur 13) ses det teoretiske diagram for baglyset.

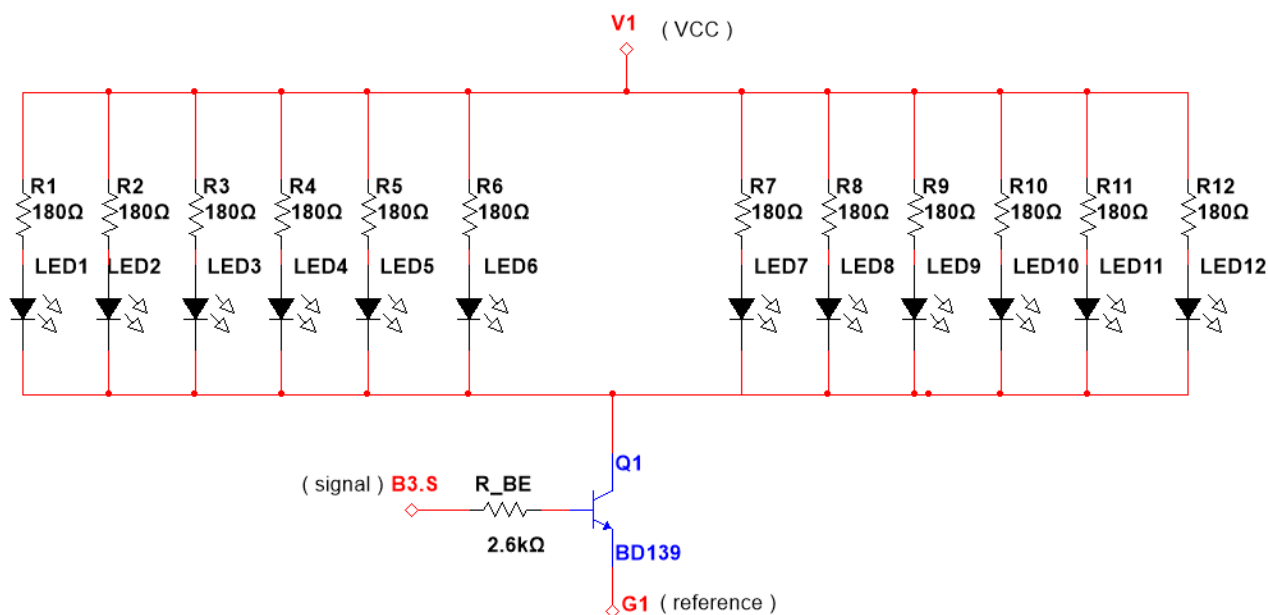


Figur 13 - Teoretisk diagram for baglys

På diagrammet er VCC koblet til LEDerne. LEDerne er koblet til transistoren på kollektor-benet. PWM signalet er koblet til Base-benet, for at kunne regulere lysstyrken på LEDerne. Desuden tilkobles reference til Emitter-benet.

I opbygningen af kredsløbet er der anvendt 180 ohm i stedet for 176.67 ohm, da vi, grundet det begrænsede udvalg af modstande, har været nødsaget til at anvende de tilgængelige modstande, der ligger nærmest den beregnede værdi.

Samtidig har vi anvendt 2.6k ohm i stedet for 3.0k ohm. Da vi ved den beregnede værdi får en for lav strøm over LEDerne. På følgende opstilling (Figur 14) ses det reelle diagram for baglyset.



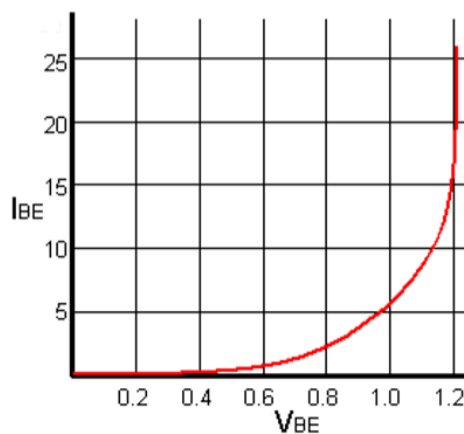
Figur 14 - Reelt diagram for baglys

Bipolar Junction Transistor (BJT)

Når BJT er i sit inaktive område (cut-off) er spændingen over base-emitter (V_{BE}) mindre end 0.5 V og strømmen til base tæt på 0. Derfor er strømmen igennem kollektor og base lig med 0. Dette er også afbilledet på Figur 15.

Dvs. når PWM signalet er højt (5 V), løber strømmen igennem LEDerne og de aktiveres.

Når PWM signalet er lavt (0 V), løber der ikke noget strøm igennem LEDerne og derfor slukkes de.



Figur 15 - Kurve for strøm og spænding i BJT (se bilag 005)

4.2.2. Software

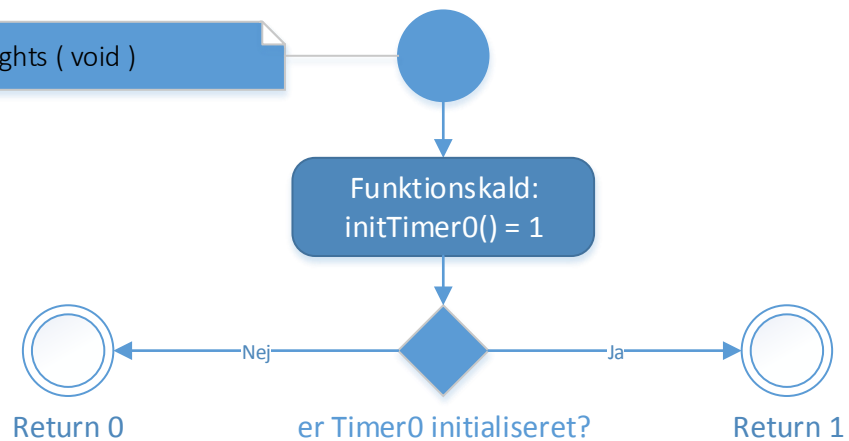
initRearLights (void): uchar

Funktionen initialiserer baglys og har en prototype, der ser således ud:

```
unsigned char initRearLights (void);
```

Funktionen anvender en hjælpefunktion; `initTimer0(void)` fra driveren "Timer.h" til at klargøre relevante udgange og timer 0. Derudover klargøres udgangen til det signal (STK-B3.S), der skal styre bilens baglys. Se Figur 16.

Hvis en fejl opstår og initialiseringen ikke gennemføres, returnerer funktionen 0, og hvis den gennemføres returneres 1.



Figur 16 – UML aktivitetsdiagram over `initRearLights`

rearLights (uchar dutyCycle): uchar

Funktionen styrer, vha. et PWM signal (STK-B3.S), bilens to baglys-sæt og lysstyrke af disse. Prototypen ser således ud:

```
unsigned char rearLights (unsigned char dutyCycle);
```

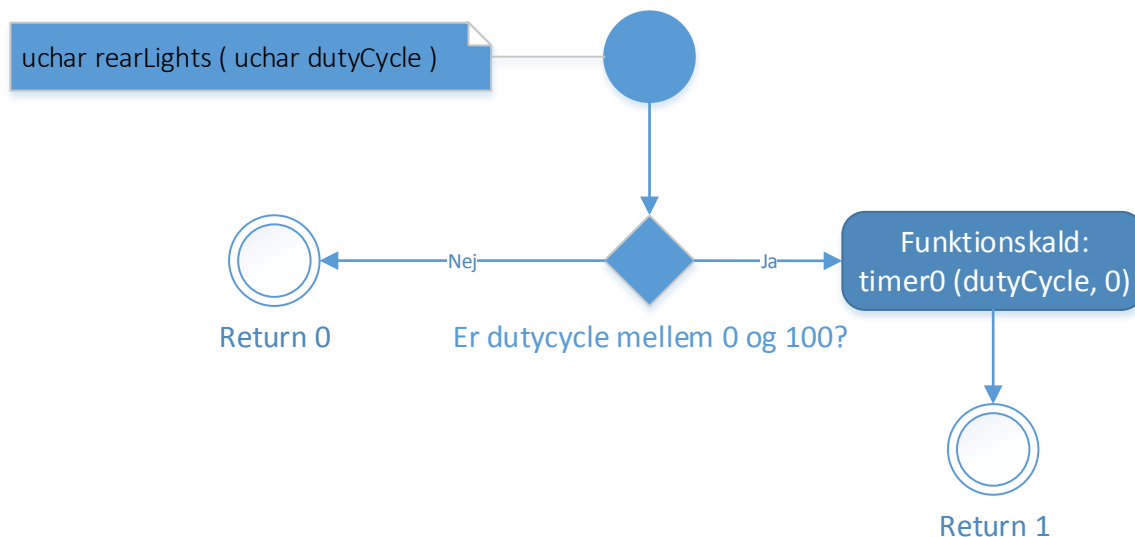
Parameteren kontrolleres for dets gyldighed. Hvis den er ugyldig, gøres ingenting og der returneres 0. Hvis den er gyldig, gennemføres resten af funktionen og der returneres 1. Se Figur 17.

dutyCycle skal have en værdi i intervallet [0:100]. Værdien angiver den duty cycle, som signalet skal have, angivet i procent.

Funktionen anvender timer 0 på STK 500 kittet til at generere et PWM signal (STK-B3.S). Timer 0, som findes i driveren "Timer.h", anvendes via funktionen;

Timer0 (dutyCycle, frequency)

"frequency" sættes til 0, hvilket svarer til vores default frekvens på 14.456 Hz.



Figur 17 – UML aktivitetsdiagram over rearLights

4.3. Detektorer

4.3.1. Hardware

Detektorerne er hardware dele der har til formål at kunne fortælle hvor langt bilen er nået på banen, ved hjælp af refleksbrikker. Disse detektorer er lys-detektorer der ved afsendelse af lys har mulighed for at registrere når dette afsendte lys bliver reflekteret tilbage. Når dette sker afgiver detektorerne et 5V signal på deres ben. Disse detektorer er eksterne komponenter til dette projekt.

Grænsefladen mellem detektorerne og STK-500 kittet er konstrueret af en OR-gate til at samle de to detektorers signal til et enkelt, i serie med et lille kredsløb af en modstand samt kondensator der fjerner prel.

4.4. Forlys

4.4.1. Hardware

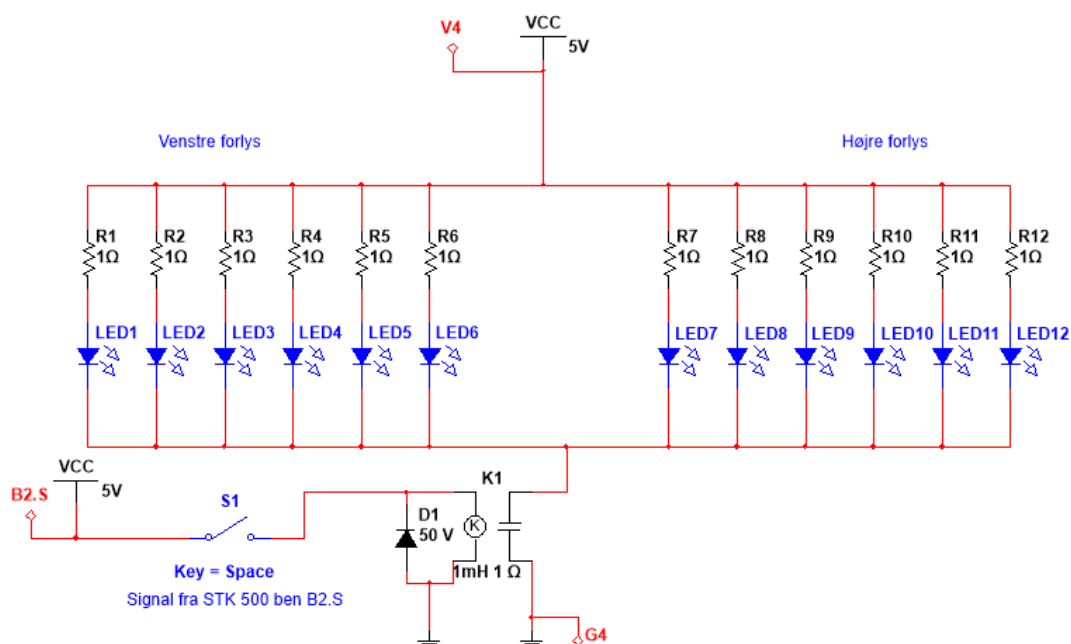
Forlys modulet består af følgende komponenter:

6 stk. hvid LED'er i henholdsvis på venstre og højre foran bilen i alt 12 LED'er.

12 stk. modstande som for modstande til de 12 LED'er.

1 stk. Hamlin HE721A0500 relæ til at styre on/off mode på forlysende, som bliver styret af et signal fra STK-B2.S benet (). LED'erne er slukket når relæet er i off mode.

På Figur 18 ses diagrammet for forlys hardware.

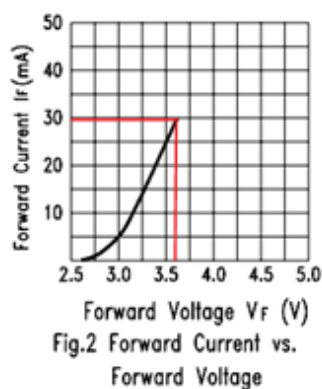


Figur 18 - NI Multisim simulering af forlys designet. VCC 5V spændingskilde, R1-R2 modstande, LED1-LED12 LED lysdioder og K1 er enkelt relæ som on/off switch som styres af S1 står for STK-B2.S benet.

Vi har beregnet for modstandene R1-R12 på følgende måde:

Strømmen gennem LED lysdioderne LED1-LED12 er besluttet til at være 30 mA +/- 1 mA.

Ifølge databladet aflæses spændingsfaldet over LED lysdioden til 3,6 V ved 30 mA se Figur 19.



Figur 19 - Graf over spændingsfald over LED lysdioden

Forsyningsspændingen VCC er 5V, og spændingsfaldet over modstandene R1-R12 er ifølge KVL det samme på 3.6 V. Dette udregnedes således:

$$V_R = 5V - V_{S.F.LED} = 5V - 3.6V = 1.4V$$

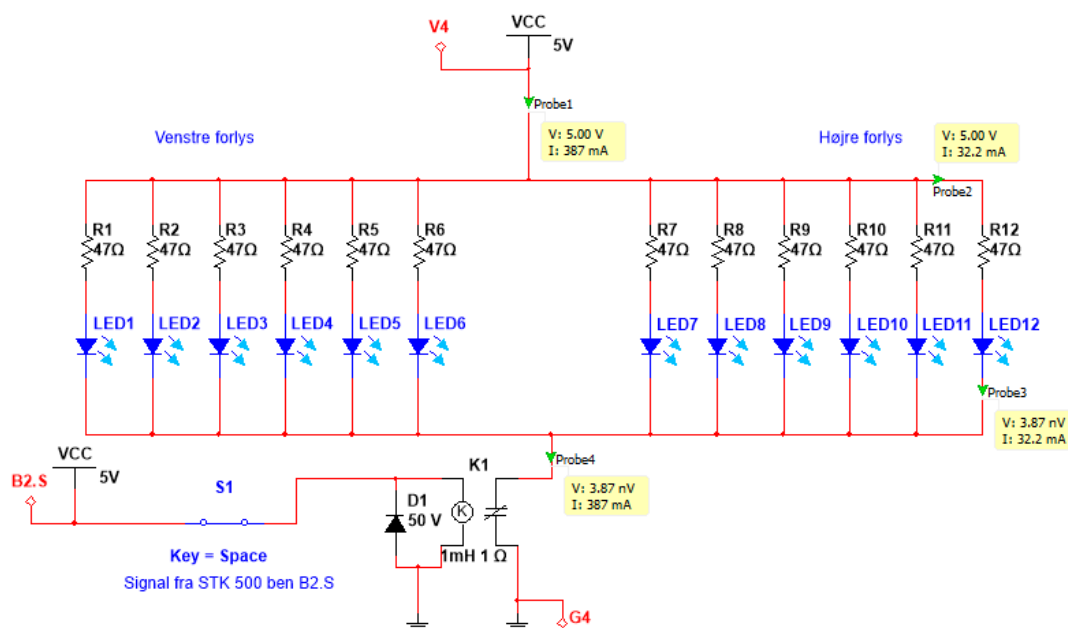
$$V_{S.F.LED} = 3.6V$$

Nu kan vi udregne modstandene for LED'erne således:

$$R_{LED} = \frac{V_R}{I_{LED}} = \frac{1.4V}{30mA} = 46,67\Omega \approx \underline{\underline{47\Omega}}$$

$$I_{LED} = 30mA$$

Nu kan vi simulere forlyset designet i multisim, for at afprøve vores analysedelen se Figur 20.

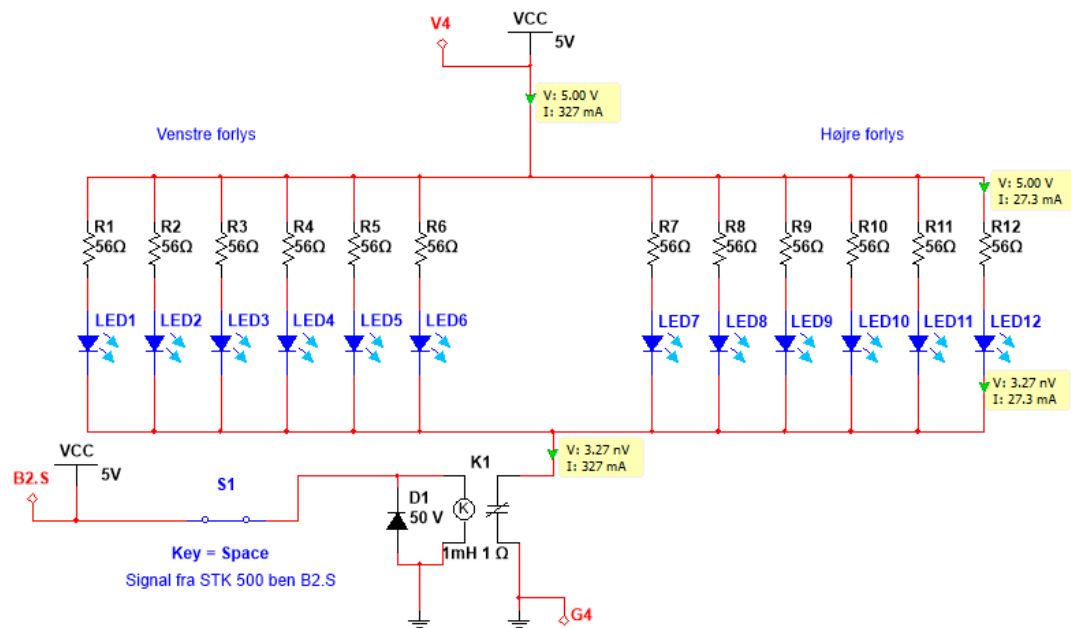


Figur 20 - NI Multisim simulering af forlyset designet. VCC 5V spændingskilde, R1-R2 modstande, LED1-LED12 LED lysdioder og K1 er enkelt relæ som on/off switch som styres af S1 står for STK-B2.S benet.

Som det kan ses, stemmer simuleringen ikke helt overens med udregningerne. For at eliminere usikkerheden af modstands beregningen, har vi bygget et kredsløb med 1 stk. LED og 1 stk. modstand på fumlebræt for nærmere undersøgelse. Der kan være afvigelser på komponenter i multisim i forhold til det reelle komponent, samt at multisim ikke indeholder alle specifikke komponenter.

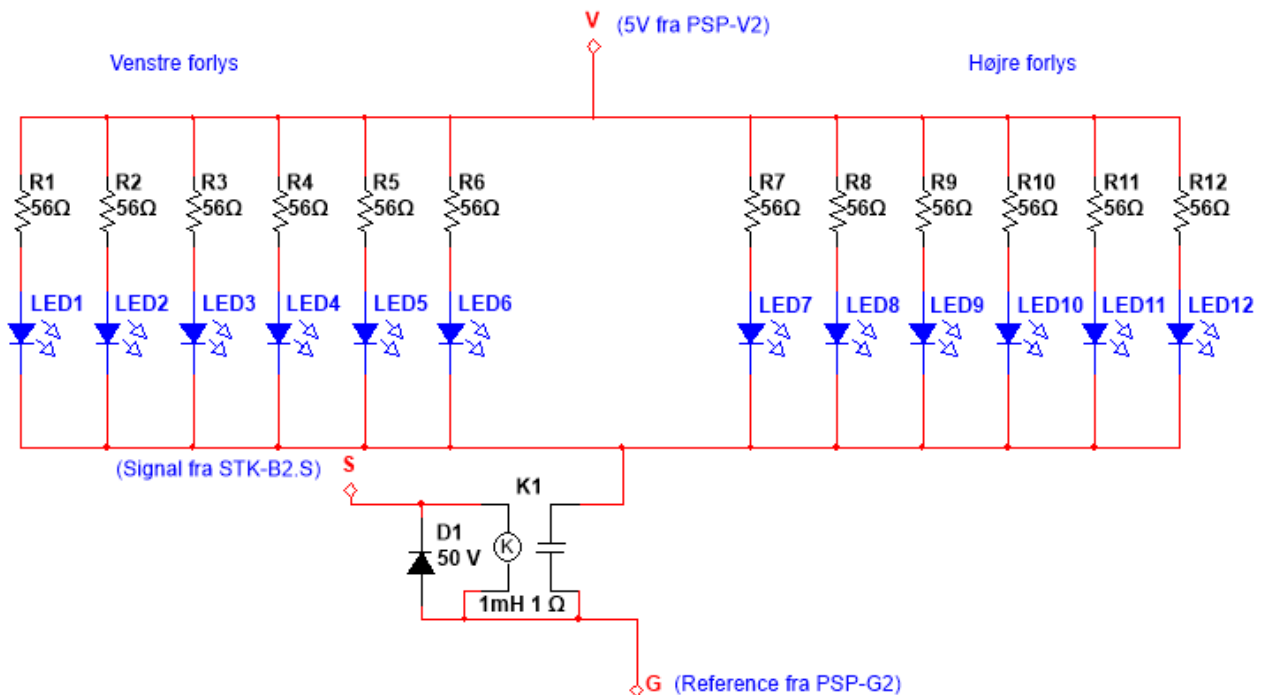
Vi har prøvet med følgende forskellige modstande:

1 stk. 47Ω, 1 stk. 51Ω, 1 stk. 56Ω samt 1 stk. hvid LED. Vores målinger har vist os at 47Ω gav en strøm på ca. 34 mA, mens 56Ω gav en strøm på ca. 30,21 mA som er inden for grænsen på +/- 1 mA. På grund af disse observationer har vi valgt at bruge 12 stk. af 56Ω i vores implementering af forlyset hardware modulet (Figur 21). I følgende figur har vi målt strømmen til 27.3 mA, men i realisering er den på ca. 30,21 mA.



Figur 21 - NI Multisim simulering af forlys designet. VCC 5V spændingskilde, R1-R2 modstande, LED1-LED12 LED lysdioder og K1 er enkelt relæ som on/off switch som styres af S1 står for STK-B2.S benet.

Herunder vises diagrammet for det endelige design af forlys hardware (Figur 22).



Figur 22 - Diagrammet af forlys designet.

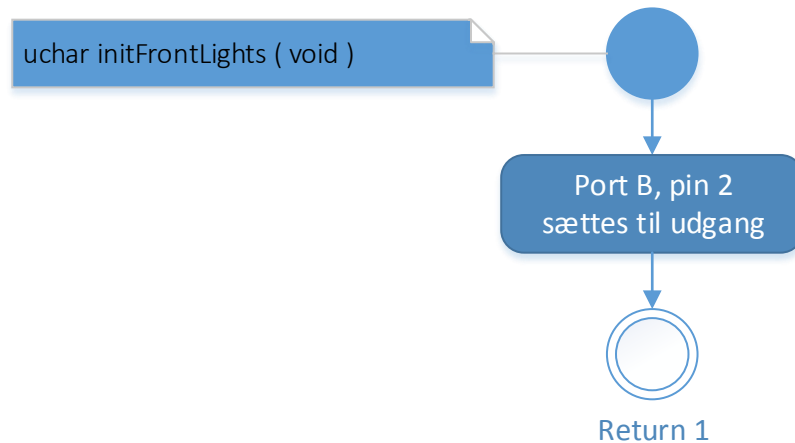
4.4.2. Software

initFrontLights(void): uchar

Funktionen initialiserer forlys og har en prototype, der ser således ud:

```
unsigned char initFrontLights (void);
```

Funktionen sætter den ønskede pin (STK-B2.S) til udgang ved at ændre register DDRB, således at port B ben 2 sættes til udgang. Hvis dette gennemføres returneres 1. Se følgende aktivitetsdiagram (Figur 23).



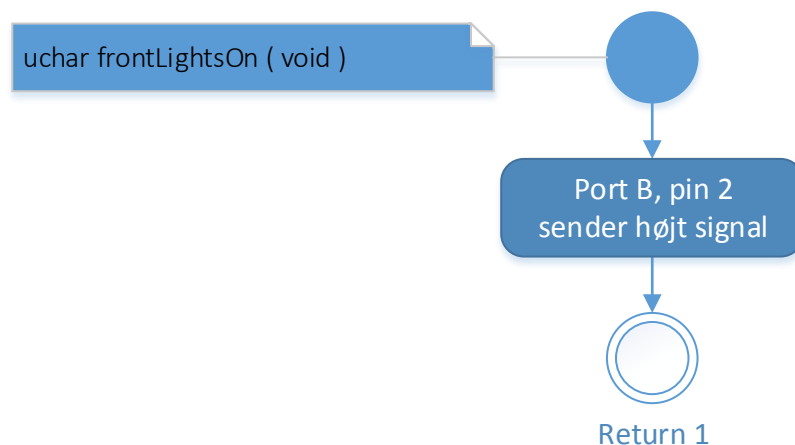
Figur 23 - UML aktivitetsdiagram over `initFrontLights`

frontLightsOn(void): uchar

Funktionen tænder forlys og har en prototype, der ser således ud:

```
unsigned char frontLightsOn (void);
```

Funktionen sætter udgang (STK-B2.S) til at sende et højt signal og dermed tænde forlyset. Hvis dette gennemføres returneres 1. Se følgende aktivitetsdiagram (Figur 24).



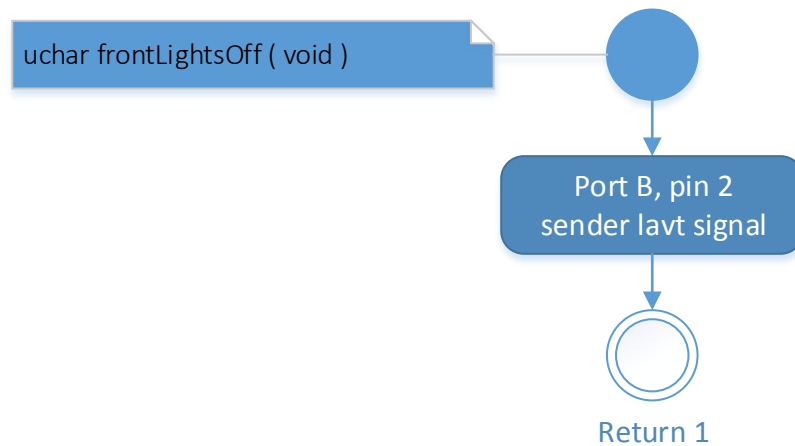
Figur 24 - UML aktivitetsdiagram over `frontLightsOn`

frontLightsOff(void): uchar

Funktionen slukker forlys og har en prototype, der ser således ud:

```
unsigned char frontLightsOff (void);
```

Funktionen sætter udgang (STK-B2.S) til at sende et lav signal og dermed slukke forlyset. Hvis dette gennemføres returneres 1. Se følgende aktivitetsdiagram (Figur 25).



Figur 25 - UML aktivitetsdiagram over `frontLightsOff`

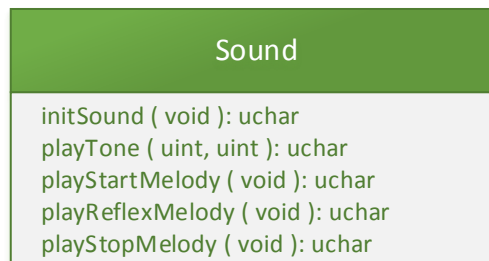
4.5. Lyd

4.5.1. Software

Sound er en driver der bruges af main og udnytter timer driveren, som set på Figur 8 under systemarkitekturen. Sound driveren har til formål at bruge timer driveren til initialisering og afspilling af lyd over en højttaler.

Sound driveren skal have funktioner der kan afspille startmelodi, refleksbriklyd og slutmelodi som beskrevet under

kravspecifikationen, use case 2. Derudover skal den kunne spille en tone i en given frekvens og tid for at kunne teste tonernes frekvens i forhold til det menneskelige øre.



Figur 26 - Sound blok uddrag fra software blokdiagram (Figur 8)

initSound

Denne funktion har til formål at initialisere timer 1, gennem timer driveren, til brug af følgende funktioner.

playTone

Denne funktion har til formål at afspille en lyd med en given frekvens i en given tid. Dette skal gøres gennem funktionskald til timer 1, samt delays. Denne funktion er hovedsageligt til at kunne teste timer 1's styrke som lydafspiller samt den inkluderede højttaler til STK-500 kittet. Et aktivitetsdiagram over denne funktion er angivet til højre (Figur 27).

playStartMelody

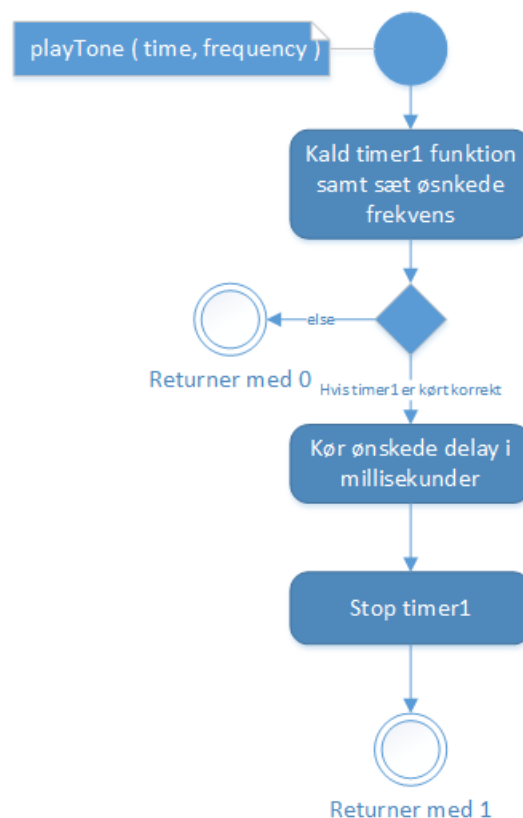
Denne funktion skal kunne afspille startmelodi som beskrevet under kravspecifikationen, use case 2. Dette skal gøres gennem en lang række funktionskald til timer 1, samt delays imellem disse.

playReflexMelody

Denne funktion skal kunne afspille refleksbriklyd som beskrevet under kravspecifikationen, use case 2. Dette skal gøres gennem funktionskald til timer 1, samt delays imellem disse.

playStopMelody

Denne funktion skal kunne afspille slutmelodi som beskrevet under kravspecifikationen, use case 2. Dette skal gøres gennem en lang række funktionskald til timer 1, samt delays imellem disse.



Figur 27 - Aktivitetsdiagram over playTone

4.6. Motorstyring

4.6.1. Hardware

Motorstyringsmodulet består af følgende komponenter:

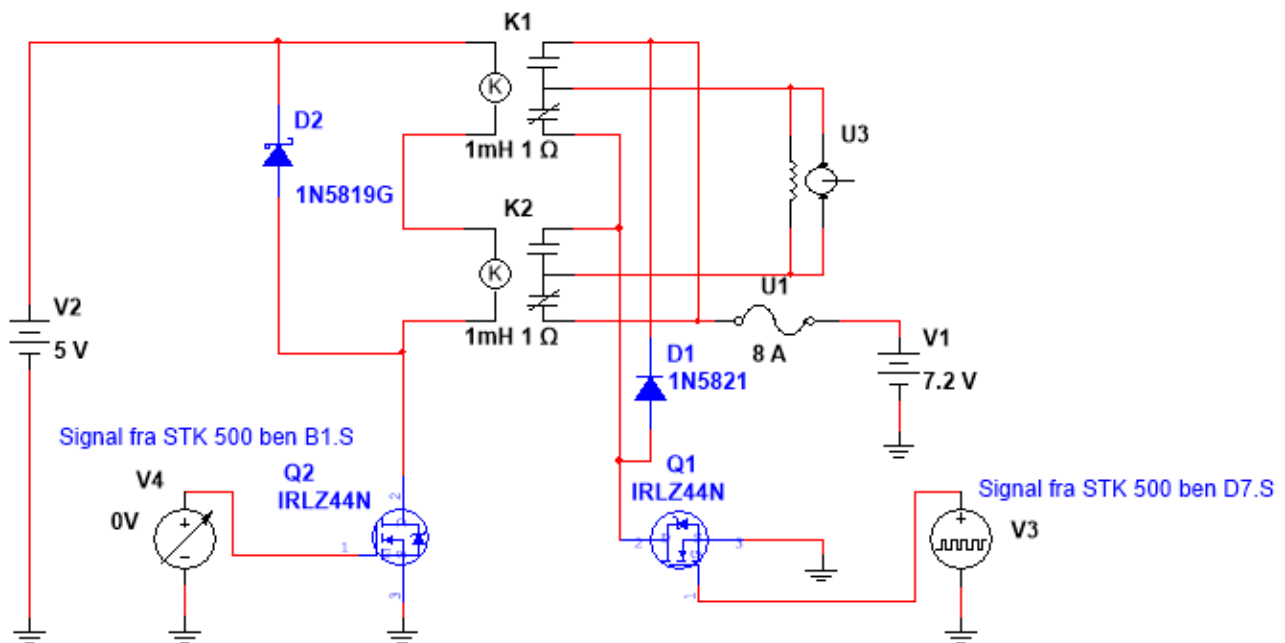
1 stk. dobbelt relæ (se bilag 007) til at styr bilens retning, dvs. når relæet er i off mode køre bilen forlæns, og mens den er i on mode køre bilen baglæns (Figur 28). I figuren blev der brugt 2 stk. enkelt relæer, da vi kunne ikke finde double relæer i NI Multisim.

1 stk. IRLZ44 Mosfet transistor (se bilag 011), samt 1 stk. 1N5819 diode til styring af on/off modes på dobbelt relæet via et signal fra STK-B1.S benet se Figur 28.

1 stk. IRLZ44 Mosfet transistor, samt 1 stk. 1N5821 diode til styring af motorens hastighed via STK-D7.S benet som sende PWM signal se Figur 28.

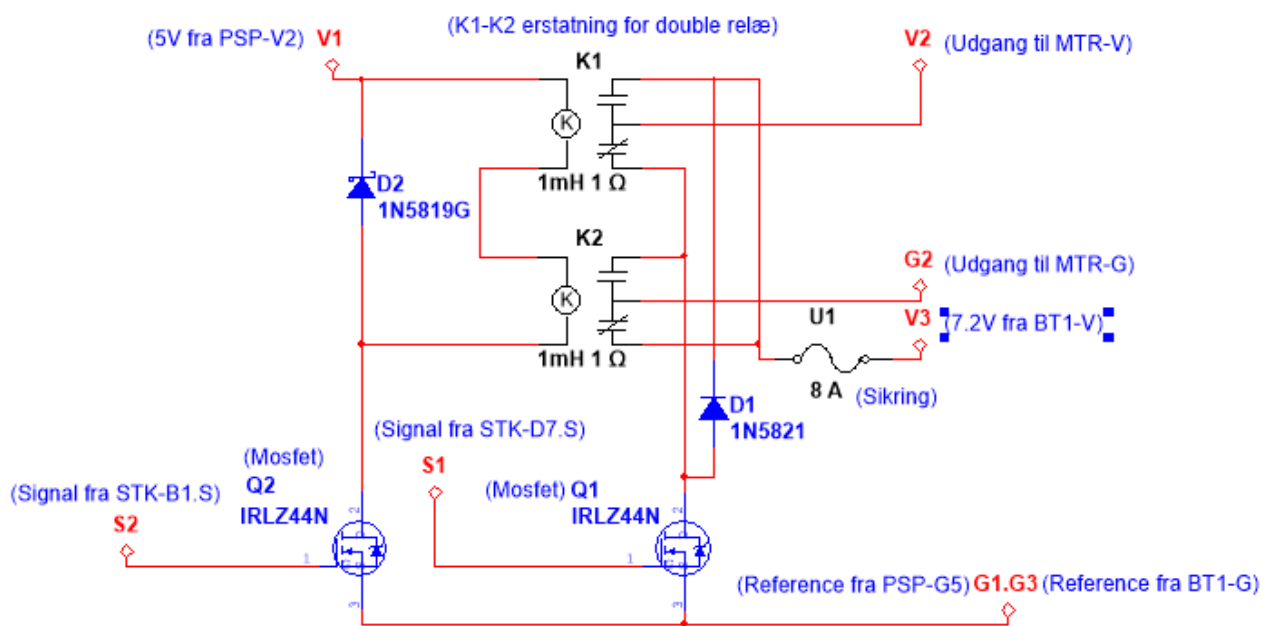
Dioderne kunne vi ikke regne ud, da vi har ikke haft undervisning i motor elektronik. Vi har talt med vejlederen om emnet, og han har rådet os til at bruge dioderne som nævnt ovenover.

Herunder er diagrammet for kredsløbet opbygget i multisim (Figur 28).



Figur 28 - NI Multisim opstilling af motorstyrings designet. K2 og K3 er enkelte relæer som erstatning for dobbelt relæ, D1 og D2 er dioder, M1 er motoren, V4 er signalen fra STK-B1.S til on/off, V3 er PWM signalen fra STK-D7.S, Q2 mosfet transistor IRLZ44 til styring af on/off på relæet og Q1 mosfet transistor IRLZ44 til styring af belastningen på motoren via STK-D7.S PWM signal.

Herunder diagrammet for det endelige design af motorstyringsmodulet er følgende (Figur 29):



Figur 29 - Diagrammet af motorstyrings designet.

4.6.2. Software

motorSpeed (uchar, uchar): uchar

Funktionen styrer, vha. et PWM signal (STK-D7.S) og et almindeligt ON/OFF signal (STK-B1.S), bilens retning og hastighed. Se aktivitetsdiagram (Figur 30). Prototypen ser således ud:

```
unsigned char motorSpeed(unsigned char dutyCycle, unsigned char direction);
```

De to parametre kontrolleres for deres gyldighed. Hvis blot én af dem er ugyldig, gøres ingenting og der returneres 0.

dutyCycle skal have en værdi i intervallet [0:100]. Værdien angiver den duty cycle, som signalet skal have, angivet i procent.

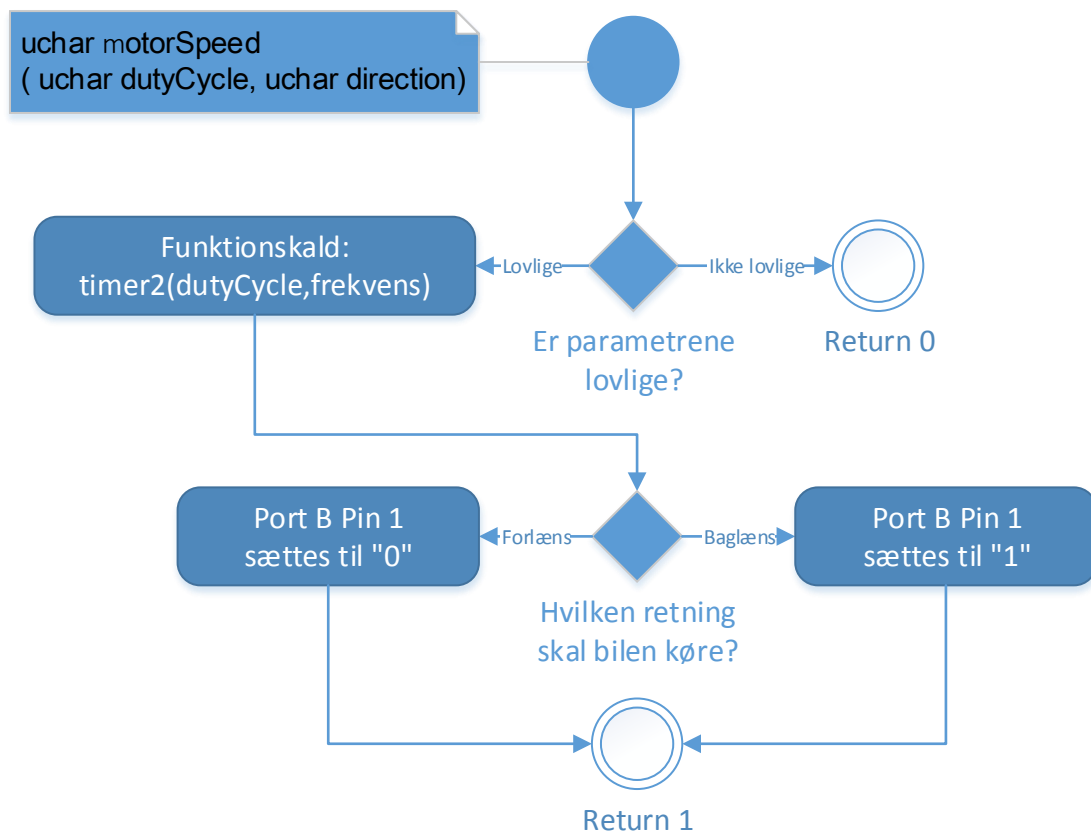
direction skal være 0 eller 1, hvor 0 betyder fremad og 1 betyder baglæns.

Funktionen anvender timer 2 på STK 500 kitted til at generere et PWM signal (STK-D7.S). Timer 2 anvendes via funktionen;

```
timer2 (dutyCycle, frequency)
```

som findes i driveren "Timer.h".

"frequency" sættes til 4, hvilket svarer til 113 Hz, da frekvensen ikke må være for høj. Dette skyldes at den interne modstand i spolen, der er i motoren, er større jo større frekvensen er.



Figur 30 - UML aktivitetsdiagram over motorSpeed

initMotor (void): uchar

Funktionen initialiserer motorstyringen og har en prototype, der ser således ud:

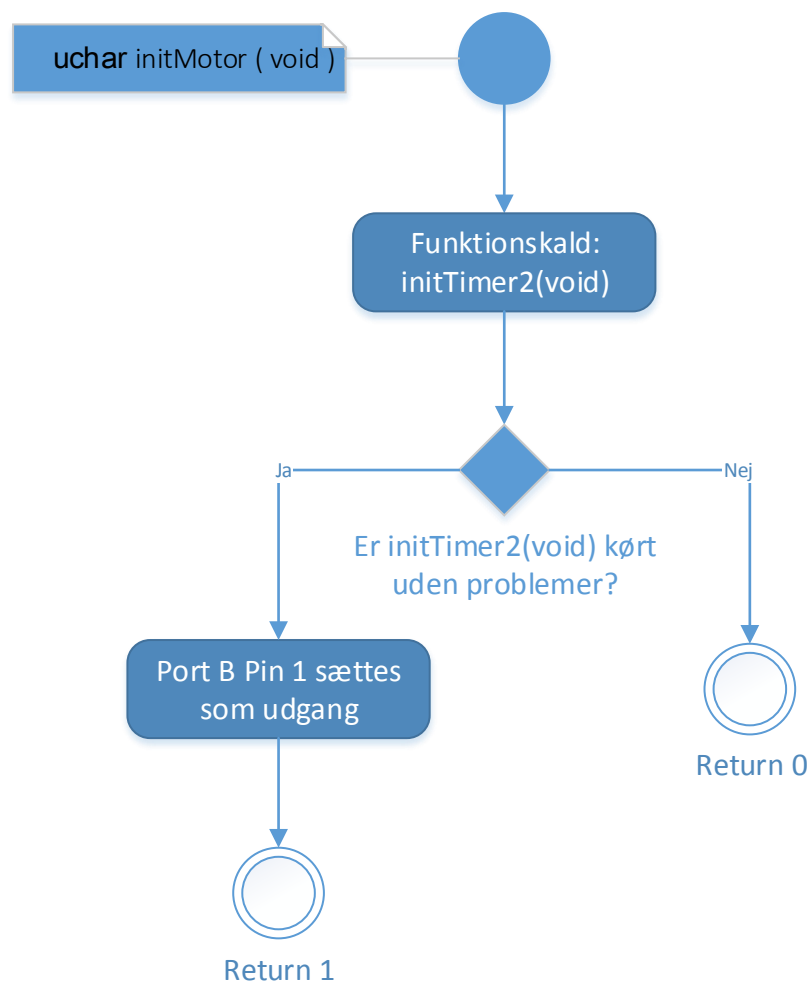
```
unsigned char initMotor(void);
```

Funktionen anvender en hjælpefunktion; `initTimer2(void)` fra driveren "Timer.h" til at klargøre relevante udgange og timer 2.

Derudover klargøres udgangen til det signal (STK-B1.S), der skal styre bilen retning.

Hvis en fejl opstår, og initialiseringen ikke gennemføres, returnerer funktionen 0, og hvis den gennemføres returneres 1.

Aktivitetsdiagram er som følgende (Figur 31).



Figur 31 – UML aktivitetsdiagram over `initMotor`

4.7. Strømstyring

4.7.1. Hardware

9,6V batteriet er tilsluttet til

- STK 500 kit (ekstern elektronik)
- Fejltælleren (ekstern elektronik)
- Spændingsregulator (7805) med et output på 5,0V.

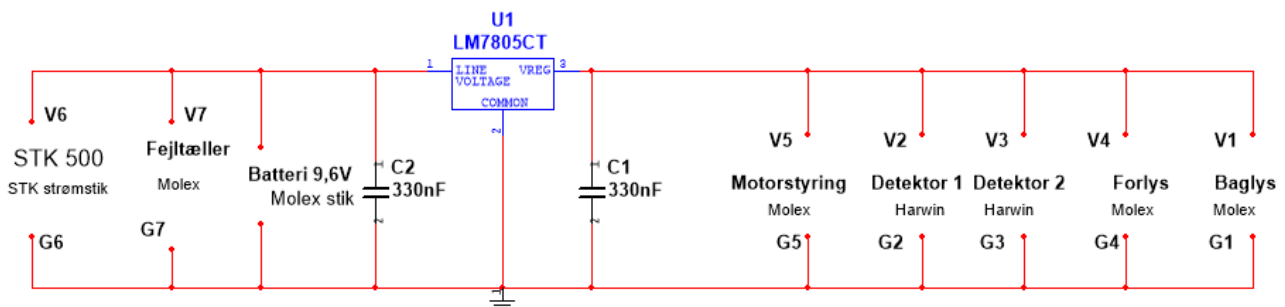
Komponentliste: (Hentet fra IDE øvelse 3, da samme kredsløb var brugt i opgave 2, bilag 058)

2 stk. 330nF Kondensator.

1 stk. LM7805 Spændingsregulator

Spændingsregulatorens output på 5,0V leverer strøm til bilens lygter, reflektions-detektorerne, samt til strømstyringen.

For at forhindre at forbindelserne mellem de forskellige elementer falder fra hinanden har vi valgt at bruge Molex-Stik som bindeled, med undtagelse af reflektionsdetektorerne, hvor vi har brugt harwin stik.



4.8. Main

4.8.1. Software

main (void): int

Denne funktion skal, ved hjælp af de drivers, der er udarbejdet, kunne styre bilens gennemkørsel af banen. Programmet tager udgangspunkt i de use cases der er beskrevet i kravspecifikationens funktionelle krav.

Meget essentielt for programmet, er den globale variabel, der hedder "counter". Indholdet i denne variabel er antallet af refleksbrikker, som bilen har detekteret. Ved hjælp af denne værdi kan vi bestemme hvordan bilen skal opføre sig, alt efter hvor langt på banen den er kommet.

Eksempel:

Når bilen nærmer sig bakken, skal motoren yde mere, hvis bilen skal nå toppen. Vi ved at bakken er placeret lige efter refleksbrik nummer 2. Den information bruger vi til at programmere bilen, således at den øger spændingen over motoren, når "counter" får værdien 2.

På samme måde tilpasser vi hele tiden bilens hastighed og retning mm, baseret på hvor langt bilen er på banen. Dette gøres for alle værdier af "counter" i intervallet [0:11], da bilen skal passere 11 refleksbrikker i alt.

Indholdet af "counter" bliver optalt ved hjælp af STK-500 kittets eksterne interrupt (INT0). Et interrupt sker som følge af et højt signal(5V) på (STK-D2.S), der bliver sendt fra detektorerne. INT0 er initialiseret således at et rising edge signal trigger. Når et interrupt fremkommer afspilles refleksbriklyden og der ligges én til counter.

Et detaljeret aktivitetsdiagram kan findes under Bilag 016.

Præcist hvad der sker under de forskellige faser, altså efter hver refleksbrik passeres, bliver bestemt ved adskillige tests under implementeringsfasen. Det endelige resultat kan ses under Bilag 053 (Main.c)

5. Accepttest

5.1. Accepttestspecifikation for funktionelle krav (Use Cases)

Use case 1: "Kør banen"

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/ kommentar
1.1	Brugeren tænder for bilen og placerer den, så den kan køre forlæns ind på banen. Bilen aktiveres og en specifik "startmelodi" afspilles, hvorefter bilen begynder at køre (Use case 2).	Brugeren tænder for bilen og placerer den, så den kan køre forlæns ind på banen ved at passere startlinjen. Bilen aktiveres.	Visuel test: Bilen afspiller en specifik "startmelodi" og kører derefter forlæns ind på banen.	Resultat er som forventet.	OK
1.2	Når bilen har passeret refleksbrik nummer 6, standser bilen inden refleksbrik nummer 7 nås.	Brugeren tænder for bilen og placerer den, så den kan køre forlæns ind på banen ved at passere startlinjen. Bilen aktiveres.	Visuel test: Bilen standser mellem refleksbrikkerne 6 og 7.	Resultat er som forventet.	OK
1.3	Bilen bakker, indtil refleksbrik nummer 5 er passeret, og standser inden refleksbrik nummer 4 nås.	Brugeren tænder for bilen og placerer den, så den kan køre forlæns ind på banen ved at passere startlinjen. Bilen aktiveres.	Visuel test: Bilen bakker og standser mellem refleksbrikkerne 4 og 5.	Resultat er som forventet.	OK
1.4	Bilen kører forlæns, indtil refleksbrik nummer 7 nås.	Brugeren tænder for bilen og placerer den, så den kan køre forlæns ind på banen ved at passere startlinjen. Bilen aktiveres.	Visuel test: Bilen kører forlæns og passerer refleksbrik 7.	Resultat er som forventet.	OK
1.5	Bilens bringes til standsning i målområdet, der er defineret som 0,5-2,0 meter efter refleksbrik nummer 7.	Brugeren tænder for bilen og placerer den, så den kan køre forlæns ind på banen ved at passere startlinjen. Bilen aktiveres. Når bilen holder stille, måles med et målebånd fra refleksbrik nummer 7 til bilens forreste punkt.	Måling: Bilen standser mellem 0,5 og 2 meter efter refleksbrik nummer 7.	Resultat er som forventet.	OK

Use case 2: "Afspil lyd"

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/kommentar
2.1	Når bilen aktiveres afspilles "startmelodi".	Use Case 1 ("Kør banen") udføres.	Akustisk test: Når bilen aktiveres for at køre ind på banen afspilles "startmelodi". Denne skal stemme overens med hastighed og tone efter bilag 059.	Resultat er som forventet.	OK
2.2	Hver gang en refleksbrik passeres, afspilles "refleksbriklyd".	Use Case 1 ("Kør banen") udføres.	Akustisk test: Hver gang en refleksbrik passeres, afspilles "refleksbriklyd". Denne skal stemme overens med hastighed og tone efter bilag 060.	Resultat er som forventet.	OK
2.3	Når refleksbrik nummer 7 er passeret og bilen holder stille, afspilles "slutmelodi".	Use Case 1 ("Kør banen") udføres.	Akustisk test: Når bilen holder stille efter at have kørt banen afspilles "slutmelodi". Denne skal stemme overens med hastighed og tone efter bilag 061.	Resultat er som forventet.	OK

Use case 3: "Styr forlyset"

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/kommentar
3.1	Forlyset skal være tændt, så længe bilen er i bevægelse, ellers skal forlyset være slukket.	Use Case 1 ("Kør banen") udføres.	Visuel test: Mens bilen holder stille, er forlyset slukket. Lige så snart bilen er i bevægelse tænder forlyset. Når bilen holder stille i målområdet, slukkes forlyset igen.	Resultat er som forventet.	OK

Use case 4: "Styr baglyset"

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/kommentar
4.1	Baglyset skal være slukket når motoren ikke er belastet.	Use Case 1 ("Kør banen") udføres med et multimeter (voltmeter) monteret parallelt på bilens motor. Der aflæses på voltmeteret samtidig med at baglyset betragtes.	Visuel test: Når motoren er ubelastet er baglyset slukket.	Resultat er som forventet.	OK
4.2	Så længe bilens motor belastes konstant eller belastningen øges, tændes bilens almindelige baglyset.	Use Case 1 ("Kør banen") udføres, med et multimeter (voltmeter) monteret parallelt på bilens motor. Der aflæses på voltmeteret samtidig med at baglyset betragtes.	Visuel test: Når motoren belastes konstant eller øges, vises "almindeligt baglyset".	Resultat er som forventet.	OK
4.3	Hvis belastningen på bilens motor mindskes tændes bremselys.	Use Case 1 ("Kør banen") udføres, med et multimeter (voltmeter) monteret parallelt på bilens motor. Der aflæses på voltmeteret samtidig med at baglyset betragtes.	Visuel test: Når motorens belastning mindskes, vises "bremselys".	Resultat er som forventet.	OK

5.2. Accepttestspecifikation for ikke-funktionelle krav (Use Cases)

Bemærkninger:

- Som multimeter (amperemeter/voltmeter/ohmmeter) anvendes "TENMA Palm Size Digital Multimeter"
- Som Oscilloskop anvendes " Analog Discovery, USB Oscilloscope"

Generelle krav

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/ kommentar
1.1	Middelværdien af forsyningsspændingen til bilens motor skal styres via pulsbreddemodulation (PWM) fra 0 til maksimal batterispænding (7.2V +/- 0.5V) i 100 trin.	Der laves et testprogram til microcontrolleren (ATMega 32), der cyklisk genererer et PWM-signal med dutycycle fra 0 – 100% i 100 trin med en periodetid på mellem 50 og 100 millisekunder. Signalet aflæses med et oscilloskop.	Ved observation af oscilloscopet konstateres det at PWM-signalet ændrer sig monotont fra 0 – 100% dutycycle inden for 10 sekunder.	Resultat er som forventet. Se evt. Bilag 015.	OK
1.2	På bilens højre og venstre side skal placeres detektorer, der som minimum kan registrere en R80 refleksbrik i afstanden fra 2 cm til 25 cm.	Refleksbrikken fjernes	Detektorerne registrerer ingen refleksbrik.	Resultat er som forventet.	OK
		Refleksbrikken føres foran højre/venstre detektor i afstanden fra 2 til 25 cm. Detektorerne testes én af gangen.	Højre/venstre detektor registrerer refleksbrikken.		
1.3	Print til detektorer skal have en bredde på 100 mm +/- 1 mm.	Printets bredde måles med skydelære.	Printets bredde er 100 mm +/- 1 mm.	Printes bredde er målt til 100 mm.	OK

Bilens lyd

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/ kommentar
2.1	Frekvensen af tonerne til afspilning gennem bilens højttaler skal styres med et firkantsignal.	Der laves et testprogram til microcontrolleren, der cyklisk genererer et firkant-signal med en dutycycle der ender ud i en frekvens på den ønskede tone i minimum 10 sekunder. Signalet sendes ud på bilens højttaler, hvorefter det akustiske signalet aflæses med en Cherub WST-520c instrument tuner ⁶ .	Akustisk test: Den ønskede tone må maksimalt ligge +/- 5 (enheder på Cherub WST-520c) fra tonens teoretiske frekvens. Dette gentages for alle toner der bruges til startmelodi, refleksbriklyd og slutmelodi.	Resultat er som forventet.	OK

Bilens forlyd

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/ kommentar
3.1	Forlyset skal bestå af to sæt af seks stk. hvide (farvekode: InGaN ⁴ , White) LED dioder.	Der tælles antal dioder, og det kontrolleres at dioderne er en model, hvor fabrikanten angiver farven til at være InGaN ⁴ , White i databladet.	Det forventes at der er to sæt af seks stk. dioder og at der er anvendt en model, med en farve opgivet til at være InGaN ⁴ , White.	Resultat er som forventet.	OK
3.2	Implementeres med to hvide LED-sæt, der monteres med et sæt i henholdsvis højre og venstre side forrest på bilen.	Visuel test: Det kontrolleres at LED'erne er opdelt i to sæt jf. Figur 5. De to sæt er placeret i hhv. højre og venstre side, forrest på bilen.	At LED-dioderne sidder som beskrevet jf. kravet.	Resultat er som forventet.	OK
3.3	Når forlyset er tændt, skal hver LED-diode lede en middelstrøm på 30 mA ± 1 mA.	Strømmen igennem én af dioderne måles med et amperemeter (multimeter).	Forventet middelstrøm på 30 mA +/- 1 mA.	Målt til 30,2 mA. Resultat er som forventet.	OK

⁶ <http://store.musicbasics.com/t-wst-520c.html>

Bilens baglys

Krav nr.	Krav	Test	Forventet resultat	Resultat	Godkendt/ kommentar
4.1	Baglyset skal bestå af to sæt af seks stk. røde (farvekode: GaP ⁵ , Red) LED dioder.	Der tælles antal dioder, og det kontrolleres at dioderne er en model, hvor fabrikanten angiver farven til at være GaP ⁵ , Red i databladet.	Det forventes at der er to sæt af seks stk. dioder og at der er anvendt en model, med en farve opgivet til at være GaP ⁵ , Red.	Resultat er som forventet.	OK
4.2	Implementeres med to røde LED-sæt, der monteres med et sæt i henholdsvis højre og venstre side forrest på bilen.	Visuel test: Det kontrolleres at LED'erne er opdelt i to sæt jf. Figur 6. De to sæt er placeret i hhv. højre og venstre side, forrest på bilen.	At LED-dioderne sidder som beskrevet jf. kravet.	Resultat er som forventet.	OK
4.3	Når "almindeligt baglys" er aktiveret, skal hver LED-diode, der indgår i de førømtalte LED-sæt, lyse svarende til en middelstrøm på 3 mA \pm 1 mA.	Strømmen igennem én af dioderne måles med et amperemeter (multimeter).	Forventet middelstrøm på 3 mA \pm 1 mA.	Målt til 3,31 mA. Resultat er som forventet.	OK
4.4	Når "bremselyset" er aktiveret, skal hver LED-diode, der indgår i de førømtalte LED-sæt, lyse svarende til en middelstrøm på 15 mA \pm 1 mA.	Strømmen igennem én af dioderne måles med et amperemeter (multimeter).	Forventet middelstrøm på 15 mA \pm 1 mA.	Målt til 14,52 mA. Resultat er som forventet.	OK

6. Diskussion

6.1. Generelt

Selvom delene er blevet konstrueret hver for sig og af forskellige personer, har det vist sig, at ved sammensætning har kommunikationen imellem disse virket. Dette skyldes en stærk kommunikation og strukturering i gruppen. De indledende faser i projektet (kravspecifikation og systemarkitektur) har altså fungeret efter hensigten.

Selvom det ovenstående er tilfældet, har der været enkelte situationer, hvor der er blevet opdaget fejl i en tidligere fase. Dette kan skyldes måske mangel på grundighed og anvendelse af sparringspartner fra gruppen.

6.2. Software

Der har ingen bemærkelsesværdige problemer været ved sammensætning af softwaren.

6.3. Hardware

Det er først på et sent tidspunkt i forløbet, at der er blevet taget højde for signalbehandlingen fra detektorerne. Dette har bevirket, at der har været behov for at ændre en smule i systemarkitekturen efterfølgende.

Der er også blevet lavet om i valget af både forlys og baglys hardware komponenterne efter, at de var blevet loddet, da der er blevet opdaget fejl i beregninger og/eller opslag i datablade. Dette har bevirket unødvendigt arbejde.

7. Konklusion

7.1. Konklusion – Gruppe

Bilen kan gennemføre banen som ønsket, med mellem 10 og 30 fejl. Accepttesten er ligeledes markeret med OK, hele vejen igennem. Bilen er forsynet med de i problemformuleringen beskrevne enheder og funktioner, som er nærmere defineret i kravspecifikationen.

7.2. Konklusion – Philip

Det har været rigtig spændende at arbejde med dette semesterprojekt, og det har givet et godt indblik i, hvad jeg tror rigtigt ingeniør arbejder indbefatter. Jeg har lært meget om selve projekt-processen, på trods af den begrænsede mængde undervisning der har været i alle de værktøjer vi har fået tildelt.

Indledningsvis virkede kravspecifikationen en smule uoverskuelig og indviklet, men efterhånden som jeg fik lov at arbejde med den, har jeg fået en rigtig god forståelse for, hvordan sådan en skal se ud. Dette gør sig også gældende med mange andre faser i projektet.

Under design og implementerings faserne, har mine ansvarsområder næsten udelukkende været software, hvilket har være rigtig godt at arbejde med, da der har været styr på systemarkitekturen på det område. Dog skal jeg næste gang lave min design fase færdig, før jeg begynder på implementeringen. Det har ikke været et problem for mig at skrive koden uden at være færdig med designet, da jeg havde strukturen i hovedet. Jeg følte det nemmere at udtrykke den, ved at skrive koden først, end ved at skrive designfasen. Derfor skal jeg arbejde på at lave designet ved hjælp af pseudokode og aktionsdiagrammer før jeg kaster mig over koden.

Jeg har i gruppen påtaget mig en lederrolle, hvor jeg har været ordstyrer og tovholder i det omfang det har været nødvendigt. Der har været behov for én, der tog initiativ og som kunne holde folk til ilden. Det har

jeg forsøgt at gøre, uden at være en sur idiot. Jeg har oplevet at gruppen har accepteret og anerkendt min måde at gøre dette på.

Alt i alt har det været et godt, spændende, sjovt og lærerigt projekt.

7.3. Konklusion – Henrik

Projekt 1 har været belærende på mange områder. Først og fremmest hvordan man som ingeniør bearbejder et projekt fra start til slut. Selv hvis dette ikke har været den helt præcis samme fremgangsmetode som ude i en virksomhed, har det givet et godt indblik i proceduren. Det har lige såvel givet et godt indblik i grænsefladerne mellem individuelle enheder, software såvel som hardware.

Personlighedsmæssigt har det også været en øjenåbner. Jeg har oplevet at arbejde med personligheder der fungerer godt med min, men også personligheder jeg ikke fungerer med. Jeg har lært at jeg skal være bedre til at takle problemer i grupper og at jeg skal være bedre til at sætte tilstrækkelige forventninger.

Alt i alt føler jeg at projektet er gået tilstrækkeligt efter mine tidlige forventninger.

7.4. Konklusion – Rasuli

Projektet giver mig en god forståelse, om hvordan der er sammenhæng mellem det teoretiske og det praktiske. Med hensyn til projektet, har jeg lært hvordan man kan samarbejde med forskellige folk og hvordan man skal respektere andres meninger, selvom man ikke er enige. I forbindelse med ingeniørfaglige styrker og svagheder, synes jeg, at jeg var god til finde fejl, holde mine aftaler og lave min opgaver til tiden. Desuden jeg var god til hardwaredel, for eksempel beregninger for at finde de rigtige komponenter, designe diagrammer, teste dem på fumlebræt og opbygge dem på printplade. Jeg havde problemer med softwaredelen og vil herefter sørge for at blive bedre til programmeringen. Projektet har bekræftet mig i, at den uddannelse som jeg har valgt, leder mod det arbejdsområde, som jeg har lyst til at beskæftige mig med i fremtiden.

7.5. Konklusion – Youssef

Jeg synes godt om at man afslutter første semester med et projekt, som bygger på den viden man har tilegnet sig på semestret. Dette afspejles på måden, jeg har løst mine opgaver på, som jeg havde ansvar for i projektet. Jeg har trygt kunne stole på viden og hjælp fra både mine medstuderende i gruppen og mine lærere. Jeg har lært meget af at søge hjælp og rådgivning. Jeg har også taget nogle beslutninger omkring udførelsen af mine opgaver, hvor jeg har undervurderet tiden til at udføre og undersøge for eventuelt fejl, som gav mig stress. Under denne proces har jeg lært at bruge kontrollere og lette min stress ved at søge hjælp hos mine gruppemedlemmer og mine lærere. På den måde har jeg kommet ud af min stress på en effektivt og lærerig måde. Min gruppe led under, at to af vores gruppemedlemmer har valgt at stoppe med semestret sent i projektprocessen, men igen her har vores resterende gruppemedlemmer i fællesskab løst problemet effektivt og stressfrit og dette har været gruppens styrke. Derfor har projektet været et godt projektforløb for mig, da man hele tiden lærer noget nyt om sine grænser og kompetencer.

7.6. Konklusion – Kasper

Dette semesterprojekt har, for mig, samlet trådene mellem de forskellige 1. semesterfag på bedste vis. Min forståelse for grænseflader og sammenspil mellem hardware og software har tidligere været ubekræftede formodninger, men dette projektforløb har afklaret mange af netop disse formodninger. Samtidigt har fordybelsen i arbejdet vækket min interesse for og givet indblik i før ukendte områder.

Jeg mener, at vores gruppe har fungeret godt. Der har været gensidig respekt blandt gruppens medlemmer og for det stykke arbejde de hver især har udført.

Jeg har igennem forløbet haft følelsen af, at jeg har beholdt overblikket og ikke ladet mig unødigt stresse. Udover at udføre mine opgaver, har jeg også kunne bidrage med hjælp til hvem i gruppen der måtte have haft brug for det og dette har personligt bekræftet mig i, at jeg også fremover ønsker at beskæftige mig med denne uddannelsesretning og arbejdsområde.

7.7. Konklusion – Morten

Fagligt har arbejdet i gruppe 5, betydet meget for min indsigt i hvordan kode KAN skrives, der har nemlig været stor forskel på hvordan jeg ville have skrevet koden i forhold til hvordan den er blevet udarbejdet af vores software-team. Jeg har så til gengæld kunnet komme med nogle hardwaremæssige ting de andre i gruppen ikke havde tænkt over, eksempelvis den tidsmæssige besparelse der kan være ved at benytte et Vero-board med rækker, frem for ét med øer.

Arbejdsomt har gruppen fungeret fint, især fordi der har været meget få prioritetsmæssige faktorer der spillede ind da arbejdsopgaverne skulle fordeles (hvilket gjorde fordelingen nem) og fordi indstillingen i gruppen, hele tiden har været at man er startet på uddannelsen for at lære noget. Folk har været meget arbejdsivrige og der derfor har været flere tilfælde af at der var for mange om buddet når en opgave skulle besættes.

Der har ikke været behov for den helt store faglige sparring, men for at sikre at ingen strandede med en opgave de ikke magtede, så har vi lavet et sikkerhedsnet der fungerer på den måde at der til hver enkelt opgave er (mindst) 1 assistent tilkøbt. Denne arbejdsstruktur har gjort at hver enkelt gruppemedlem får (indgående) indsigt i mere end blot sin egen arbejdsopgave, hvilket er med til at skabe større faglig kunnen. Der opstod så desværre den uheldige situation at både den ansvarlige og assistenten for design af forlyset valgte at forlade gruppen. Men da begge har valgt at starte på 1. semester igen, kan valget af forlyset som arbejdsopgave formodentlig tilskrives en faglig usikkerhed, og det er en situation der næppe genopstår på de senere semestre.

Strukturomt har gruppen haft meget faste linjer, der har fra starten været den samme ordstyrer, den samme mødereferent og de samme tovholdere osv. gennem hele projektet. Vi startede tidligt med at holde møder (vi havde holdt vores 5. møde inden andre grupper havde deres første), vores planlægning samt interne sammenhold i gruppen har været på højt niveau som følge af dette. Det skal nok lige tilføjes at vi havde en del "spildtid" under 'afstemning af vores forventninger'-fasen, da der gik noget tid med at finde ud af at folk udtrykte ting forskelligt men faktisk mente det samme.

Alt i alt har det været en fornøjelse at være i gruppen, og der er ingen af gruppens medlemmer jeg ville være ked af at arbejde sammen med på et senere semester.