

SYDDANSK UNIVERSITET

SEMESTERPROJEKT I AUTONOME ROBOTTER, (F18) - RB-PRO2

CIV. ROBOTTEKNOLOGI 2. SEMESTER - GRUPPE 6

DEADLINE FOR AFLEVERING: 25/05-2018

Pole Position

Studerende:

August Ravn Mader
aumad17@student.sdu.dk
16/04-1997

August R.M.

Studerende:

Daniel Rathleff Johansen
dajoh17@student.sdu.dk
16/09-1997

Daniel Dj

Studerende:

Mads Ingemann Lorentzen
mlore17@student.sdu.dk
20/02-1998

Mads Ingemann

Studerende:

Markus Dahl Lauritzen
mlaur17@student.sdu.dk
07/02-1997

Markus Dahl Lauritzen

Studerende:

Mirzet Traljesic
mtral17@student.sdu.dk
29/05-1996

Mirzet Traljesic

Studerende:

Troels Zink Kristensen
tkris17@student.sdu.dk
30/04-1997

Troels Zink Kristensen

Vejleder:

Dorthe Sølvason
dorthe@mmti.sdu.dk

Abstract

The purpose of this project is to create an autonomous Scalextric car, which is able to map its path on an unknown Scalextric track with the help of constructed sensors. This, in turn, is utilized to achieve the fastest possible lap time.

Serial communication is set up, which allows communication with the car from an external device. It is possible to send and receive data with a Bluetooth module and a program written in C++.

For mapping to be possible, three sensors are constructed. The three sensors are: A sensor to detect the goal line, a sensor to detect wheel revolutions, and a sensor to detect turns. It is possible to map an unknown track with data from these sensors.

The communication protocol and the three sensors are utilized in a program written in Assembly. To achieve the fastest possible lap time, PWM, ADC and interrupts are used to control the velocity of the car; the data from an accelerometer; and the data from two reflective sensors respectively.

To optimize the lap time, it is important to identify the forces that affect the car. It is possible to identify an optimal bodywork for the car by testing factors such as aerodynamic forces.

The project concludes with a test run achieving 5,6% faster lap times, compared to the original car.

Indholdsfortegnelse

1 Forord	1
2 Indledning	2
3 Problemformulering	3
4 Kommunikation	4
4.1 Kommunikationsprotokol	4
4.2 Baudrate	5
4.3 Kondensator	6
4.4 Flowdiagram	7
5 Sensorer	8
5.1 Kortlægning af bane	8
5.2 Målstregsensor	8
5.3 Afstandssensor	11
5.4 Svingsensor	12
5.5 Delkonklusion	15
6 Assembler	16
6.1 PWM	16
6.2 ADC	17
6.3 Interrupts	17
6.4 Flowdiagram	20
6.5 Hastighed i sving	22
6.6 Hurtig runde	22
6.7 Omgangstid	23
6.8 Delkonklusion	23
7 Design af karrosseri	24
7.1 Acceleration	24
7.2 Svinghastighed	26
7.3 Delkonklusion	30
8 Valg af karrosseri	31
8.1 Delkonklusion	33
9 Konklusion	34
10 Perspektivering	35
11 Litteraturliste	36
12 Bilag	37
12.1 Oversigt over elektronisk bilag	37

1 Forord

Rapporten er udarbejdet af August Ravn Mader, Daniel Rathleff Johansen, Mads Inge-mann Lorentzen, Markus Dahl Lauritsen, Mirzet Traljesic og Troels Zink Kristensen.

Projektet (RB-PRO2) tilhører uddannelsen, Civilingeniør i Robotteknologi, på 2. semester 2018 v. Syddansk Universitet i Odense. Projektet dækker 10 ECTS point, hvor de to kurser, RB-KMI2 og RB-PSW2, anvendes som supplement til at udføre projektet.

Projektet blev udleveret d. 26. februar 2018 og afsluttes med aflevering af rapport d. 25. maj 2018. Der er givet en arbejdsperiode på 12 uger og fire dage. Ud fra denne arbejdsperiode er en tidsplan udarbejdet, hvilket ses under *Tidsplan* i det elektroniske bilag (Zip-fil).

Vi vil gerne takke vores vejleder, Dorthe Sølvason, for at have vejledt os.

Læsevejledning

Der skal gøres opmærksom på, at kilder refereres i rapporten med tal-eksponenter placeret i firkantede parenteser, således: [2] med eventuelt sidenummer: [2] (side 20). Disse referencer kan ses under *Litteraturliste*. Datablade refereres blot med det tilhørende *navn*. Datablade findes under *Datablade* i det elektroniske bilag (Zip-fil). Herudover refereres der også til journaler til specifikke forsøg. Disse journaler er tilgængelige i det elektroniske bilag. PDF-filen er interaktiv, hvilket vil sige, at man kan klikke på litteraturhenvisningerne og kapitlerne i indholdsfortegnelsen for at komme direkte til kapitlet eller litteraturliste.

2 Indledning

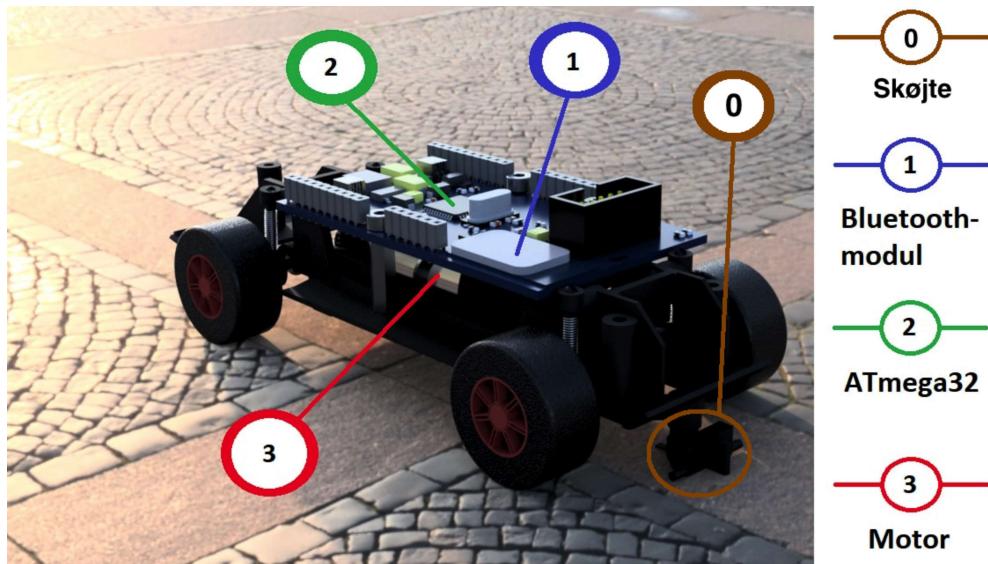
Formålet med projektet, er, at designe/konstruere hardware og software til en racerbil baseret på en mikrocontroller. Racerbilen skal gennemføre en omgang på en ukendtbane så hurtigt som muligt. Banen vil bestå af langsider og sving, hvor racerbilen vil drage fordel af ændringer i hastighed, i løbet af en omgang.

En mikrocontroller består af en CPU, ROM, RAM og I/O-enheder.^[1] Mikrocontrolleren som anvendes i dette projekt er en ATmega32, hvor "32" betyder, at der er 32KB Program Memory (ROM), hvilket er den mængde data, som permanent kan blive gemt på mikrocontrolleren.^[2] Mikrocontrollerens tilhørende CPU har en clockfrekvens på 16 MHz. Racerbilens hastighed skal automatisk ændres når en langside eller et sving forekommer. Det udføres vha. Assembler-kode. Assemblersproget er et lavniveau programmeringssprog, som automatisk omdannes til maskinkode vha. en compiler.

Ydermere skal bilen være i stand til at starte og stoppe vha. en Bluetooth-forbindelse mellem en PC og racerbilen. Denne funktion skal udføres igennem et program, der er skrevet i C++. Dette program skal kunne hente data fra bilen, fra eksempelvis sensorer, men også sætte en bestemt hastighed.

Fremover vil racerbilen blive refereret til som bilen eller racerbilen.

Figur 1 viser en 3D-model af undervognen på bilen. På undervognen sidder et board, hvorpå mikrocontrolleren, Bluetooth-modulet mm. er monteret.



Figur 1: Undervognen med de essentielle dele beskrevet.

3 Problemformulering

Projektgruppen skal foretage en ombygning af bilen, således, at den bliver autonom. Dette opnås vha. diverse sensorer, hvortil kredsløb konstrueres; dertil udvikles et Assembler-program, hvilket behandler sensorernes data. Programmet skal sørge for, at bilen kan sætte den hurtigste omgangstid på en ukendt Scalextric bane. For at opnå denne omgangstid kortlægges banen ved brug af sensorer. Udover dette skal et C++ program skrives, som kan kommunikere med bilen over Bluetooth. Programmet skal følge en kommunikationsprotokol (se *Projektoplæg* i elektroniskbilag) der kan starte, stoppe og hente data fra bilen; for eksempel data fra sensorer.

Følgende problemstillinger undersøges for at opnå den hurtigste omgangstid:

- Banen har sving, hvilke skal detekteres. Sensoren skal placeres under karosseriet samt kunne detektere sving ved høj hastighed. Med en høj nøjagtighed kan forskellen på et ydre- eller indre-sving detekteres.
- For at afgøre, hvornår bilen har passeret den hvide målstreg, skal der konstrueres en sensor til at detektere målstregen. Sensoren skal være i stand til at detektere målstregen ved høj hastighed.
- En sensor skal anvendes til at opmåle banens længde. Sensoren skal opmåle banen med minimal usikkerhed på trods af, at spændingen over banen er varierende.
- Data fra sensorer bearbejdes i Assembler-programmet. Dette Assembler-program skal være i stand til at anvende data fra sensorerne til at opmåle banen og derved sætte en bestemt hastighed ift. bilens placering på banen; således bilen ikke falder af banen.
- Der skal etableres forbindelse mellem PC og bil, hvilket udføres vha. Bluetooth. Kommunikationen programmeres i C++ og Assembler.

Projektet udarbejdes ud fra problemformuleringen.

4 Kommunikation

Det er et krav til projektet, at der udarbejdes kommunikation mellem bil og PC. Dette krav ses under Problemformulering. Kommunikationen skal realiseres vha. Bluetooth, hvilket er baseret på seriel kommunikation. Bluetooth-kommunikationen skal etableres i både C++ og i Assembler-programmet.

4.1 Kommunikationsprotokol

Seriel kommunikation mellem PC'en og bilen skal udføres vha. en bestemt kommunikationsprotokol. Protokollen fremgår af projektoplægget, som skal overholdes, med undtagelse af ekstra tilføjelser. Protokollen er opbygget på følgende måde (se figur 2):

TYPE	COMMAND	DATA [7.0]
------	---------	------------

Figur 2: Kommunikationsprotokollen opdelt i tre.

TYPE, COMMAND og DATA fylder hver især et byte, hvilket er tre bytes totalt.

TYPE angiver, hvorvidt der skal sættes en værdi i bilen (SET) eller hentes en værdi fra bilen (GET).

COMMAND angiver, hvorvidt bilen skal starte (Start) eller stoppe (Stop).

DATA sætter en specifik hastighed på bilen ud fra en brugerdefineret procentdel af den maksimale hastighed, dvs. en angivelse fra 0-100%.

De forskellige kommandoer repræsenteres med følgende hexadecimale værdier:

- | | | |
|-------------|---------------|-------------|
| • TYPE | • COMMAND | • DATA |
| – SET: 0x55 | – Start: 0x10 | |
| – GET: 0xAA | – Stop: 0x11 | – 0x00-0x64 |

Med følgende data vælges kommandoen start og hastigheden sættes til 100% (se figur 3):

TYPE	COMMAND	DATA
0x55 (SET) 1 byte	0x10 (Start) 1 byte	0x64 (100%) 1 byte

Figur 3: Eksempel med TYPE = SET, COMMAND = Start og DATA = 100%.

4.2 Baudrate

Dataene, som overføres mellem PC og racerbil, transmitteres med en baudrate på 9600 baud (også kaldet for bits per second, bps), dvs. 9600 bits pr. sekund.

For at mikrocontrolleren opererer med en baudrate på 9600bps, skal prescaleren på UBRR-registret have en bestemt værdi. Værdien til prescaleren findes ved følgende formel^{[4] (side 405)}:

$$UBRR = \frac{f_{OSC}}{(16 \cdot BAUD)} - 1$$

f_{OSC} er clock-frekvensen, som er 16MHz. Divisoren er sat til 16, da mikrocontrolleren er sat til at operere synkront, og BAUD er den ønskede baudrate på 9600bps:

$$UBRR = \frac{f_{OSC}}{(16 \cdot BAUD)} - 1 = \frac{16 \cdot 10^6 Hz}{16 \cdot 9600 bps} - 1 = 103,17$$

UBRR-registret består af både UBRRH og UBRL, hvor 103,17 gemmes i UBRL, da det kun kræver ét byte at repræsentere binært. Kun heltal kan indsættes, og derfor gemmes 103 i UBRL. For at bestemme størrelsen på fejlen, bestemmes den reelle baudrate:

$$BAUD = \frac{f_{OSC}}{(16 \cdot (UBBR + 1))} = \frac{16 MHz}{(16 \cdot (103 + 1))} = 9615,4 bps$$

Fejlens størrelse er:

$$Fejlprocent = \left(\frac{9600 bps}{9615,4 bps} - 1 \right) \cdot 100\% = 0,16\%$$

En fejlprocent på 0,16% er acceptabel.

4.3 Kondensator

Der er i dette afsnit benyttet formler til udregning af kondensator fra bogen "Electrical Engineering Principles and Applications" [3].

Ustabil kontakt forekommer mellem banen og bilens flettede kobbertråde, samt mangel på konstant spænding over skinnerne på hele banen. Det resulterer i problemer med Bluetooth kommunikationen. Dette løses med en kondensator. Ved at anvende en kondensator med et spændingspotentiale på 15V eller højere, vil boardet altid forsynes med den påkrævede spænding på 5V; dette skyldes at kondensatoren er sat på før spændingsregulatoren der omdanner de 15V fra skinnerne til boardets nødvendige 5V.

Kondensatorens størrelse bestemmes ved at måle strømmen boardet trækker. Denne måles ved at placere bilen på banen, uden at motoren er operativ. Boardet aflæses ved hjælp af et multimeter til at trække en strøm på ca. 0,1A.

Kondensatorens størrelse C , hvis enhed er farad, bestemmes med følgende ligning:

$$C = \frac{Q}{v}$$

Hvor Q er elektrisk ladning, hvis enhed er C , og v er spænding, hvis enhed er V .

Spændingen kendes allerede, og Q bestemmes:

$$Q = I \cdot t$$

Hvor I er strøm og t er tid i sekunder.

Det vælges, at kondensatoren skal være i stand til at holde en spænding over boardet på 5V efter omformeren (15V til 5V) og leve en strøm på 0,1A i minimum 0,1 sekund.

$$Q = I \cdot t = 0,1A \cdot 0,1s = 0,01C$$

Kondensatorens størrelse C kan bestemmes:

$$C = \frac{Q}{v} = \frac{0,01C}{15V} = 0,000667F = 667\mu F$$

En kondensator med en størrelse på $667\mu F$ skal anvendes, for at boardet kan trække en strøm på 0,1A og have en spænding over boardet på 5V i 0,1s.

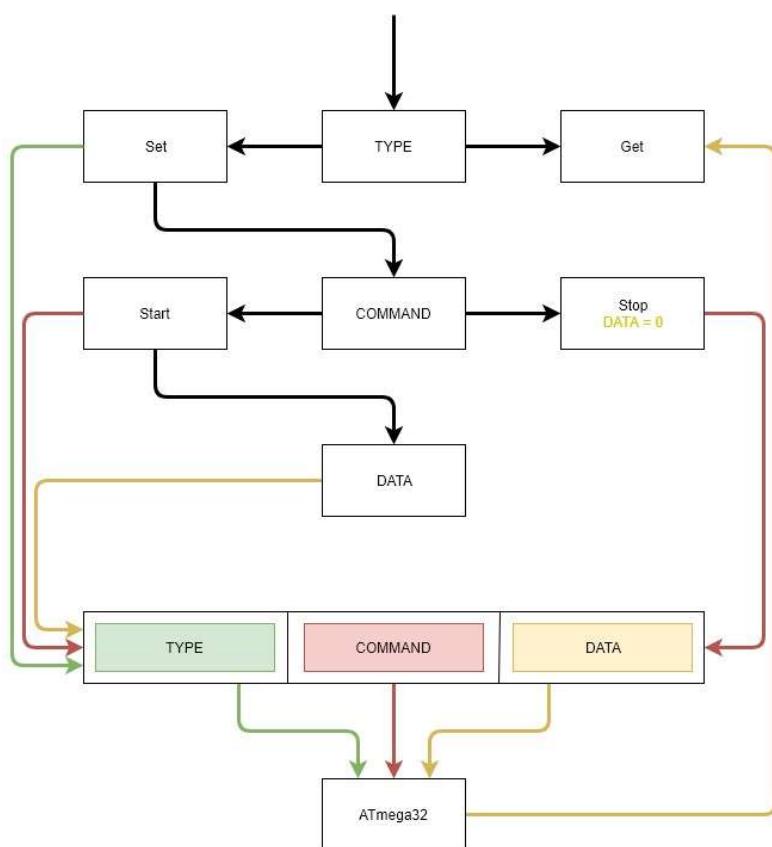
En LHK general purpose kondensator på $1000\mu F$ (se databladet i elektronisk bilag) opfylder minimumskravene og vil leve den nødvendige spænding og strøm i 0,15s.

4.4 Flowdiagram

Figur 4 viser flowdiagrammet for kommunikationen mellem en PC og racerbilen.

Kommunikationen startes; herefter vælges *TYPE*. Hvis der ønskes transmission af data til mikrocontrolleren anvendes *Set*. Herefter videresendes brugeren til *COMMAND*-feltet, hvor man enten starter eller stopper bilen. Hvis bilen skal køre, vælges *Start*, og derefter vælges en specifik *DATA*, som angiver bilens hastighed i procent. *Stop* anvendes til at stoppe bilen, hvilket sætter *DATA* = 0. Der sendes i alt tre bytes, hvor *TYPE*, *COMMAND* samt *DATA* fylder en byte hver. De tre bytes transmitteres én byte af gangen.

Hvis der i stedet ønskes at hente data fra racerbilen, anvendes *Get*. Denne type læser den seneste værdi, som mikrocontrolleren har sendt. Værdien udskrives i C++ programmet eller et tilsvarende program, som f.eks. det udleverede terminalprogram, MTTTY.



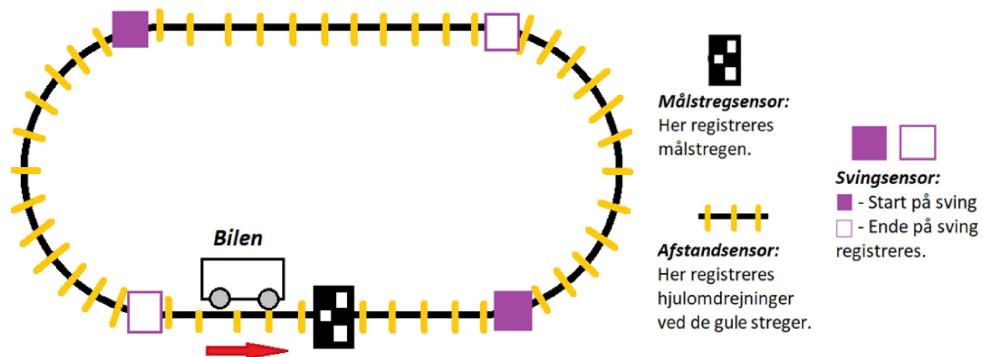
Figur 4: Flowdiagram for kommunikation mellem racerbil og PC

5 Sensorer

Kommunikationen mellem en PC og bilen er nu etableret. Der skal konstrueres sensorer ud fra de tre første krav i problemformuleringen, hvilket ses under afsnittet: 3 Problemformulering.

5.1 Kortlægning af bane

Tre sensorer vil opfylde kravene i problemformuleringen. De skal henholdsvis bestemme, hvornår bilen har kørt en omgang på banen; hvor lange de enkelte langsider og swing er; og hvornår et sving forekommer. Disse tre sensorer vil resultere i en komplet kortlægning af en ukendt bane. Figur 5 illustrerer, hvornår sensorerne skal operere.



Figur 5: Illustration af, hvordan sensorerne skal være i stand til at operere.

Med de følgende sensorer kan kortlægningen foregå på følgende måde:

Målstregsensoren registrerer målstregen og starten på en omgang. Efter målstregen vil antallet af hjulomdrehninger registreres og optælles indtil et sving forekommer. Dette efterfølges af starten på et sving, hvor antallet af hjulomdrehninger optælles i svinget indtil en ny langside forekommer. Dette fortsætter indtil bilen registrerer målstregen igen.

5.2 Målstregsensor

Bilen kører på en ukendt bane. For at bestemme længden af banen, skal bilen være i stand til at detektere, hvornår den har kørt en omgang. Målstregen er angivet med hvid maling (1,5 cm i bredden), hvilket medfører, at der reflekteres mere lys end ved den sorte bane. Der kan drages fordel af en IR-diode, som emitterer infrarød lys, hvilket reflekteres og modtages af en fototransistor.

De sensorer som opfylder kravene, og som er på lager på SDU, beskrives herunder:

CNY70:

CNY70 er en reflektiv sensor, med en indbygget IR-diode og fototransistor. Disse to komponenter er lokaliseret ved siden af hinanden i samme boks. Sensoren er lille, da den har dimensionerne: 7 x 7 x 6 mm (se databladet i elektronisk bilag). Den fungerer ved, at IR-dioden emitterer infrarød lys. Dette lys reflekteres tilbage til fototransistoren, og mængden af reflekteret lys, afhænger af materialet, der blyses.

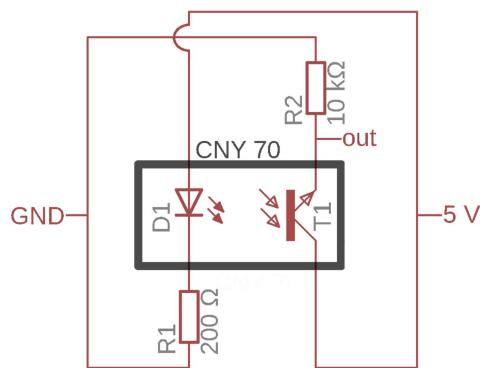
OP593 og OP298:

OP593 er en fototransistor og OP298 er en IR-diode. Disse to er separate komponenter, hvor IR-dioden emitterer lys på en overflade og tilbage til fototransistoren. Disse to komponenter kan vendes som ønsket, og giver derfor flere muligheder.

Valg af komponent:

Ud fra vurdering, vælges CNY70-sensoren fremfor de to separate komponenter, OP593 og OP298. CNY70-sensoren er lille og kompakt, hvilket er en fordel for dette projekt med en bil med sparsom plads. Da både IR-dioden og fototransistoren skal pege nedad for at kunne detektere de to farver, sort og hvid, er CNY70 den ideelle sensor for dette projekt.

Figur 6 viser kredsløbet for målstregensoren:



Figur 6: Kredsløbet for målstregensoren. Denne sensor registrerer når bilen passerer målstregen.

- D1: IR-diode
- R1: Resistor med 200 Ω
- T1: Fototransistor
- R2: Resistor med 10 k Ω

Udover CNY70-komponenten er der også to resistorer, hvor R1 vælges ud fra databladet for CNY70.

Resistor R1 sørger for, at der en bestemt spænding over IR-dioden, hvor resistor R2 sørger for mindre støj for de værdier, som sendes til fototransistoren.

Ud fra databladet for CNY70, aflæses det at strømmen i lederetningen for IR-dioden er 50 mA (I_{max}) og spændingen over dioden er minimum 1,25V ($v_{d_{min}}$). Dette giver en modstand på:

$$R_d = \frac{v_{d_{min}}}{I_{max}} = \frac{1,25V}{0,05A} = 25\Omega$$

En resistor skal sættes i serie med dioden. Størrelsen på resistoren bestemmes ud fra en forsyningsspænding på 5V og den maksimale collector-strøm:

$$v_{fs} = I_{max} \cdot (R1 + R_d)$$

$$5V = 0,05A \cdot (R1 + 25\Omega)$$

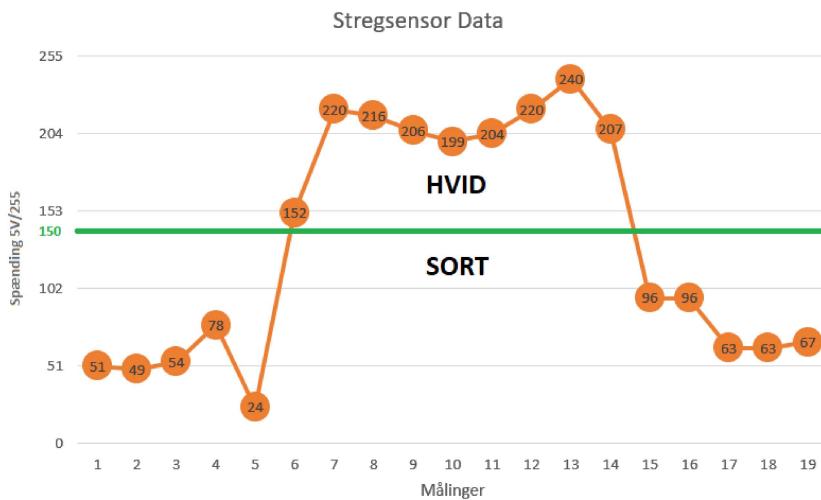
$$R1 = 75\Omega$$

Resistoren R1 bestemmes til 75 Ω. Resistoren der anvendes skal have en modstand større end 75 Ω. Derfor vælges en resistor på 200 Ω, som giver en strøm igennem dioden på ≈ 20mA. Tilmed tilføjes en resistor, R2, på 10kΩ på udgangen (se figur 6) for at få outputtet tæt på 5V når der er hvid foran sensoren.

I det elektroniske bilag under *Journal til Målstregsensor og Afstandssensor*, som viser værdier som modtages fra sensoren.

Figur 7 viser en graf over værdier fra den sorte Scalextric bane inden og efter værdier for den hvide målstreg. Værdierne mellem 152 og 240 angiver en registrering af den hvide målstreg.

På figuren aflæses en lav værdi (her: 96) efter 207, hvilket indikerer et skift fra hvid til sort. En separation af "hvid" og "sort" sættes til 150, hvilket ses på figuren.



Figur 7: Resultater fra CNY70 ved passering af målstreg.

5.3 Afstandssensor

Afstanden på banens langsider og sving kan ikke bestemmes vha. målstregssensoren. Målstregssensoren kan bestemme, *hvornår* bilen har kørt en omgang på banen. Banen kortlægges ved at bestemme længden af langsiderne og svingene på den ukendte bane vha. en Afstandssensor. Dette realiseres ved brug af optælling af hjulomdrehninger på langsider og i sving.

Der er to oplagte sensorer, der opfylder formålet. Den ene sensor indeholder komponenten CNY70, som anvendes i Målstregssensoren. Den anden sensor er en mekanisk switch.

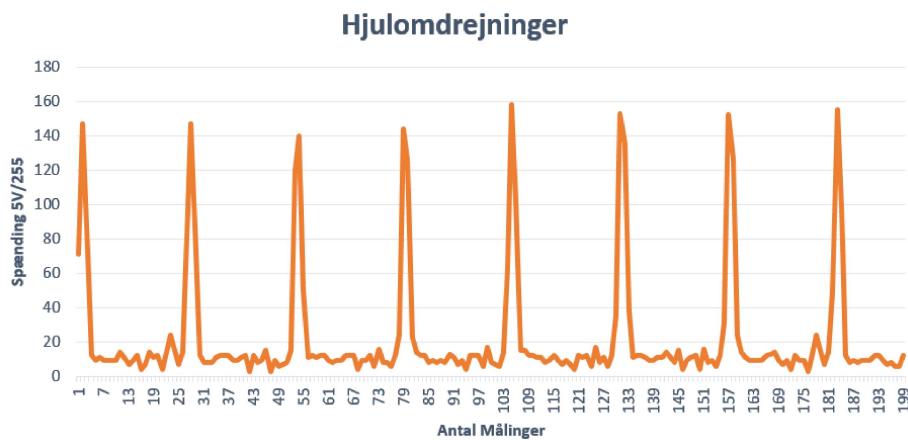
Mekanisk switch:

En mulighed for måling af hjulomdrehninger kan realiseres med en mekanisk switch. Switchen kan monteres på siden af bilen. For at aktivere switchen, kan en udløser i form af en lille metalstang monteres på periferien af en hjulaksel. Hver gang bilen har gennemført en hjulomdrehning, vil metalstangen påføre en kraft til switchen, som herved vil aktiveres og registrere en hjulomdrehning.

Valg af komponent:

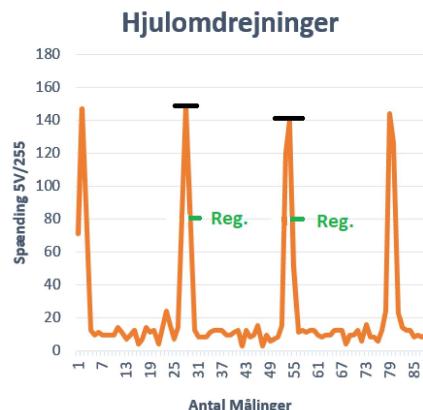
CNY70 vælges som komponent til at konstruere afstandssensoren. Den er kompakt og funktionel, hvilket gør den ideel til formålet. Funktionen er allerede testet for Målstregssensoren. For at sensoren kan opfange hjulomdrehninger, anvendes samme teknik som ved sensoren til detektion af målstregen. Hvidt tape anvendes til detektion af hjulomdrehninger, hvilket resulterer i tilnærmelsesvis samme spænding for den hvide maling, som detekteres af Målstregssensoren. Hver gang sensoren detekterer det hvide tape, registreres én hjulomdrehning. Afstandssensoren er opbygget på samme måde, som ved Målstregssensoren.

Figur 8 viser grafen over måleresultaterne for sensoren:



Figur 8: Målinger fra afstandssensoren. Én hjulomdrehning = toppunkt til fald under 80.

En hjulomdrejning registreres hver gang grafen går fra et toppunkt til et fald på y-aksen. Når værdien går fra at være høj og krydser ca. 80 på y-aksen, registreres en hjulomdrejning. Dette kan ses på figur 9, hvor en registrering er markeret med grønt.



Figur 9: Registrering af hjulomdrejning

5.4 Svingsensor

Målstregnsensoren og Afstandsensoren gør det muligt henholdsvis at bestemme, hvornår bilen har kørt en omgang, og hvor mange hjulomdrejninger bilen kører pr. langsiden og pr. sving. For at bestemme hjulomdrejninger på langsider og sving separat, skal sving bestemmes vha. en tredje sensor. Følgende sensorer kan løse opgaven:

- Kamera
- Gyroskop
- Accelerometer

Kamera:

Fordele: Det ville være muligt at se sving inden de kommer, dvs. første omgang kan køres i høj fart, og de følgende kan optimeres.

Det er muligt at bestemme, hvor mange grader et sving er og derved justere, hvilken hastighed, der kan køres i det sving (se journal *hastigheder i sving*, i elektronisk bilag).

Ulemper: Banen har 2 spor, hvilket kan gøre det svært at bestemme med et kamera, hvilken bane den skal fokusere på.

Billedbehandling i Assembler-programmering er kompliceret og samtidig har en Atmega32 ikke kraftig nok hardware til at kunne behandle billeder.

Gyroskop:

Fordele: Kan ligge fladt ned og måle. Dette simplificerer monteringsprocessen på bilen, samt det optager ikke plads på siderne hvor de to CNY70-sensorer skal placeres.

Ulemper: Det var ikke muligt at anskaffe et gyroskop, og derfor heller ikke teste det.

Accelerometer:

Fordele: Let at få fat på, eftersom det kunne anskaffes hos Jan Petersen, lektor på SDU.

Målinger fra accelerometeret giver værdier med acceptabel præcision.

Ulemper: Der er ingen betydelige ulemper ved accelerometeret.

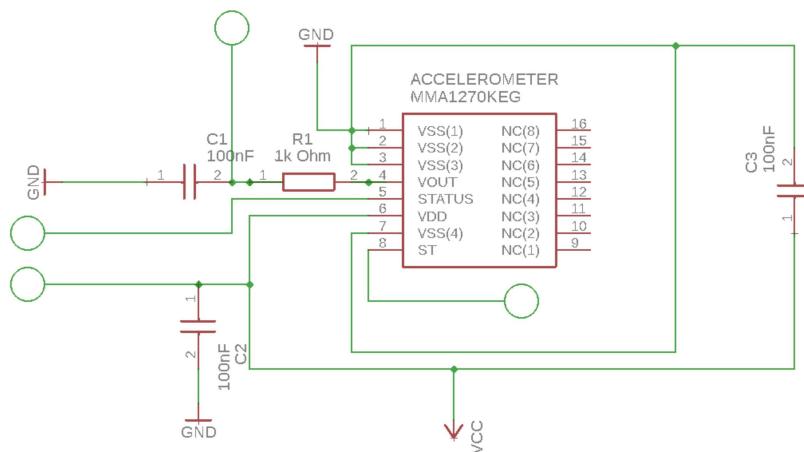
Accelerometeret vælges frem for de andre sensorer, da billedbehandling ikke kan lade sig gøre grundet utilstrækkelig processorkraft samt opbevaringsplads på mikrocontrolleren. Da det ikke var muligt at teste et gyroskop vælges det fra.

Analog sensorer:

De valgte sensorer er alle analoge sensorer. En analog sensor afgiver en spænding, som kan omdannes til et digitalt signal der kan bruges til at beregne størrelsen af svinget. Det input der eksempelvis kommer fra accelerometeret gør det muligt at bestemme om den kører på en langside eller et sving, ud fra det tal, som sensoren giver.

Kredsløb:

Accelerometeret der anvendes er et MMA1270KEG, hvilket er et et-akset analogt accelerometer. Et kredsløb designes efter databladets anvisning (se datablad i elektronisk bilag); omend portene STATUS og Self-Test (ST) ikke vil anvendes, grundet disse ikke er nødvendige for brugen af accelerometeret. Figur 10 viser kredsløbet:



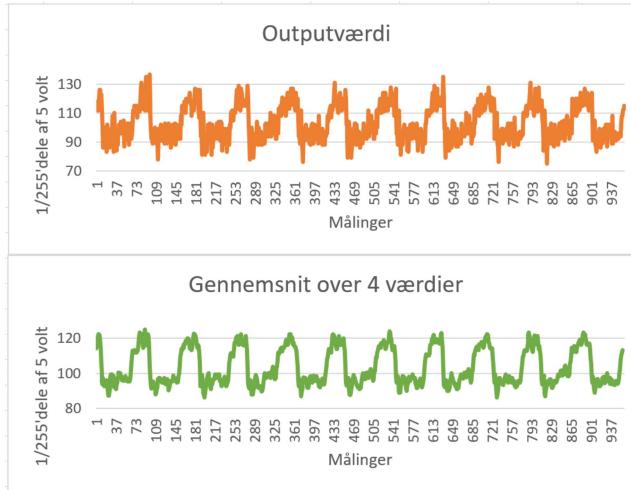
Figur 10: Kredsløb for accelerometeret MMA1270KEG.

Støjbekæmpelse: Med accelerometeret monteret på bilen og forbundet til mikrocontrolleren gennem en Analog/Digital omformer køres bilen på en bane med to 180° venstresving ift. kørselsretningen, mens data fra accelerometeret observeres.

Under prøverunderne er accelerometeret monteret på bilen med tape. Støjens størrelse besværliggører vurderingen af, hvornår et sving starter. Af den årsag tages middelværdien af de næste fire værdier, til at plotte den grønne graf i figur 11.

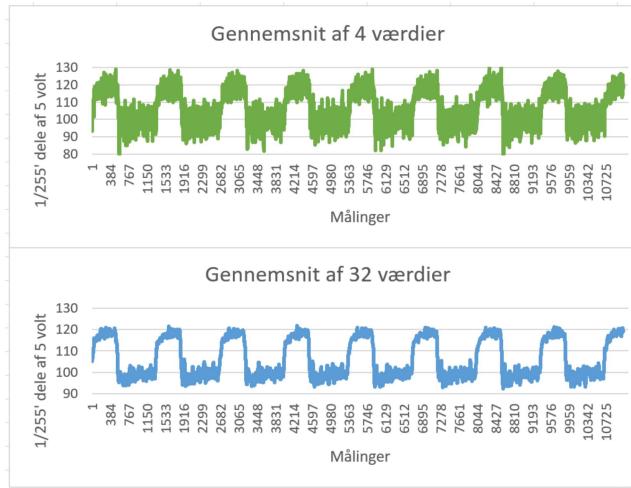
Her er det muligt at se forskel på strækninger og sving. Ved lige strækninger måler accelero-

meteret ingen kraftpåvirkning, hvilket resulterer i værdier omkring 120. I sving registreres værdier på omkring 90.



Figur 11: Data fra accelerometeret, observeret med 70 værdier pr. sekund. Ved den direkte outputværdi (orange) overlapper sving og lige strækninger hinanden.

Bluetoothmodulet kan sende data hurtigere end 70 bytes pr. sekund. Mikrokontrolleren sættes til at sende 1400 bytes i sekundet i stedet, og dermed observeres nedenstående data (se figur 12).



Figur 12: Data fra accelerometeret, observeret med 1400 værdier pr. sekund. Gennemsnit over fire værdier er ikke tilstrækkelig til at undgå overlapning.

Her bliver gennemsnittet af fire værdier relativt mindre, således, at langsiden og sving igen har mange overlappende værdier. Når der i stedet tages gennemsnittet over 32 værdier er støjlen igen bagateliseret.

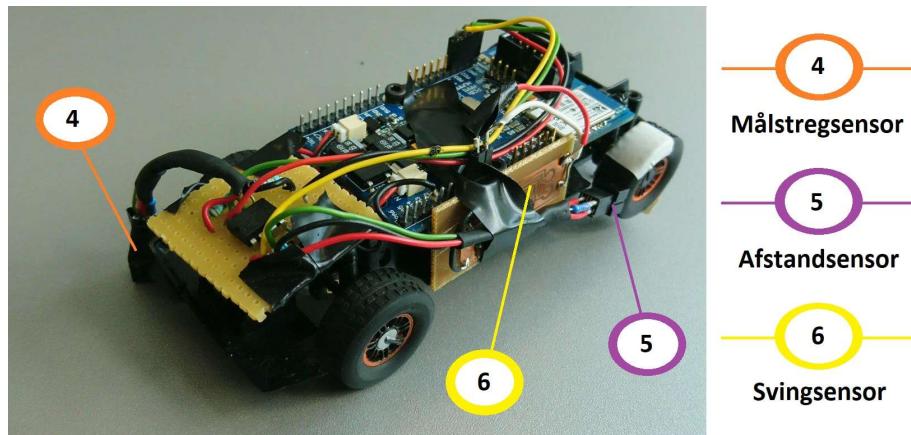
Disse data afviger fra, hvad der reelt vil ske i mikrocontrolleren:

Mikrocontrollerens ADC kører ikke med en frekvens på 1,4 kHz, men rettere ≈ 125 kHz.

Derfor er gennemsnittet over 32 værdier sandsynligvis ikke tilstrækkeligt for at fjerne støj'en. Der vælges istedet at tage gennemsnittet af 128 målinger, da det er den største eksponent af to der kan divideres med på én byte.

5.5 Delkonklusion

Figur 13 viser undervognen med de tre tilhørende sensorer monteret på bilen, henholdsvis Målstregnsensor, Afstandssensor og Svingsensor.



Figur 13: Undervogn med de tre tilhørende sensorer

4. **Målstregensoren** er konstrueret med en reflektiv sensor, som kan detektere målstregen hver gang denne passerer.
5. **Afstandssensoren** er konstrueret med den samme reflektive sensor som ved Målstregensoren. Denne kan detektere rotation af det højre forhjul.
6. **Svingsensoren** er konstrueret med et accelerometer. Ved udregning af gennemsnit af flere værdier fra accelerometeret, registreres sving på en bane.

6 Assembler

Bluetooth kommunikation og de konstruerede sensorer skal indgå i en kode der kan flashes til mikrocontrolleren, som gør bilen i stand til at køre selv, uden brug af hjælp udefra.

Assembler-koden skal kunne aflæse de forskellige sensorer, og ud fra data bestemme hastighederne på de forskellige delstrækninger af en vilkårlig bane. For ikke at vælte af skal hastigheden nedsættes kort inden et sving kommer, hvilket koden også skal kompensere for. For at opnå alt dette anvendes tre overordnede Assembler-funktioner.

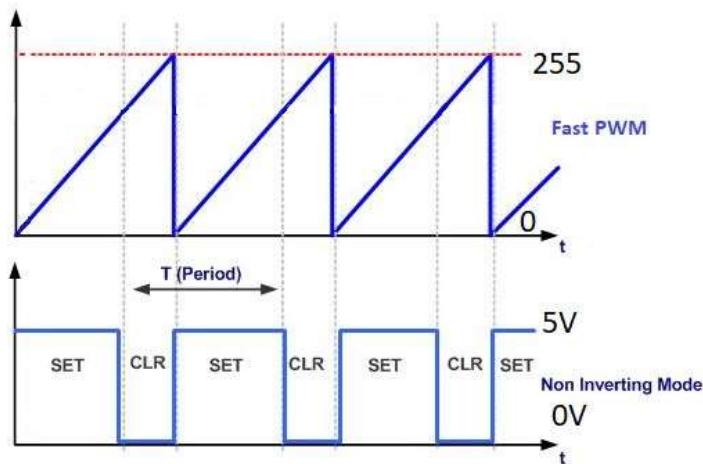
Formålet med koden er at kunne forbedre en omgang kørt med konstant hastighed rundt på en bestemt bane med både højre- og venstresving.

I følgende under afsnit anvendes databladene: "ToyRaceCar_V2_REV3" og "ATmega32A", samt bogen: "The AVR Microcontroller and Embedded Systems"^[4].

6.1 PWM

I dette afsnit er internetsiden "Pulse Width Modulation"^[5] brugt.

Pulse width modulation (PWM) virker ved at regulere tiden, for hvornår en spænding er tændt/slukket. Den gennemsnitlige spænding vil derved ændre sig afhængigt af forholdet mellem tiden der er tændt og slukket. Mikrocontrolleren har flere indbyggede PWM udgange, men til dette projekt benyttes OCR2, da denne er forbundet til transistoren, som leverer strømmen til bilens motor.



Figur 14: Illustration af PWM-signal.

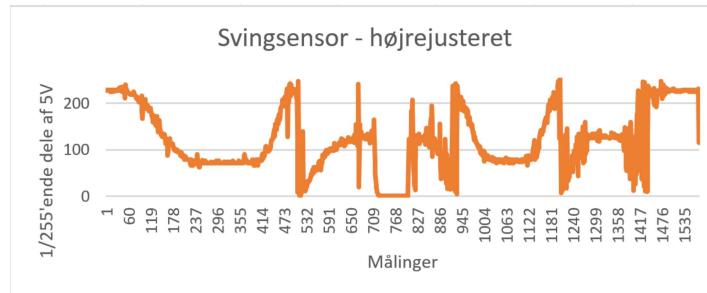
PWM'en opsættes som ikke-inverteret fast-PWM uden prescaling således, at den følger boardets clockfrekvens. Figur 14 er en illustration af, hvordan et ikke-inverterende fast-

PWM signal kan se ud. Denne PWM muliggør en regulering af spændingen over motoren fra 0-15 volt ved at sende en værdi mellem 0-255 ud på OCR2.

Hastigheden på banen kan på denne måde optimeres ved at sætte OCR2 til forskellige værdier afhængigt af, hvor på banen bilen befinder sig.

6.2 ADC

Analog til digital konvertering virker ved at omdanne en analog spænding til en binær værdi mellem 0-1023. Mikrocontrolleren har en indbygget omformer med intern referencespænding. Til opsætningen af omformeren vælges en prescaler på 128 for at holde ADC'ens frekvens på under 200 kHz [4] (side 475). I omformeren udregnes en 10-bit værdi. Det er dog nødvendigt med en 8-bit værdi for at kunne sammenligne og branche. Derfor venstrejusteres resultatet, og der læses kun fra det høje register. Dermed opnås mindre præcise resultater da konvertering foregår i grovere trin. Samtidig er det også muligt at udnytte omformerens fulde rækkevidde med udsving fra 0 til 5 V. Grafen nedenfor viser hvorfor venstrejustering er nødvendigt (se figur 15):



Figur 15: Data fra accelerometeret fra en omgang på en bane. Det kan ses at store udsving ikke kan repræsenteres, da der opstår overflow. Dertil er der også en hel del støj.

6.3 Interrupts

For at undgå at pinge hjulsensoren og målstregssensor konstant, opsaettes de som eksterne interrupts. Dette muliggører, at der kan testes på inputtene parallelt med den resterende kode. Når der registreres et input sættes koden på pause og den tilhørende underroutine eksekveres.

Opsætning af trigger-tilstand:

Interrupts har fire forskellige trigger-tilstade: Low Level, Any logic change, Falling edge og Rising edge. Til vores formål kan der benyttes to af de fire trigger-tilstade: Falling edge eller Rising edge, da disse vil aktivere interrupt'en en gang pr. detekteret streg. Falling edge anvendes således, at målstregen registreres *efter* den passerer.

Opsætning af op-amps:

Målstregssensoren og Afstandssensoren fungerer vha. interrupts. Hver gang registrering af målstregen og hjulomdrejninger forekommer, vil koden udføre et interrupt på den specifikke sensor.

I sektion 5.2, Målstregssensoren, og 5.3, Afstandssensoren, beskrives, at der udsendes stor spænding ved den hvide målstreg og det hvide tape på hjulet, samt lav spænding ved banen og hjulet. Værdierne blev sat til henholdsvis 150 og 80 ud af 255 (se *Journal Målstregssensor og Afstandssensor*), hvor 255 svarer til den maksimale spænding på 5V. Hvis værdien for Målstregssensoren overstiger 150, registreres en målstreg, og på samme måde ved Afstandssensoren ved en overstigning af 80, registreres en halv hjulomdrejning.

Ved hjælp af ADC kan det programmeres, således, at der kun registreres værdier over 150 eller 80. Problemet med denne metode, er, at der konstant skal pinges på sensoren for at se om den er steget over 150 eller 80. Dette problem kan løses ved at opsætte sensorerne som interrupts ved hjælp af en operationsforstærker, der anvendes som en komparator, hvilket omdanner det analoge output fra sensorene til et logisk output.

Værdierne omdannes til en referencespænding således (ift. den maksimale spænding på 5V):

$$\frac{5V \cdot 150}{255} \approx 3V$$

$$\frac{5V \cdot 80}{255} \approx 1,6V$$

Der skal anvendes spændingsdeling for at få den ønskede spænding.

Forholdet mellem de to resistorer bestemmes således for:

... Målstregssensor:

$$\frac{5V - 3V}{5V} \approx \frac{2}{5}$$

... Afstandssensor:

$$\frac{5V - 1,6V}{5V} \approx \frac{3,4}{5}$$

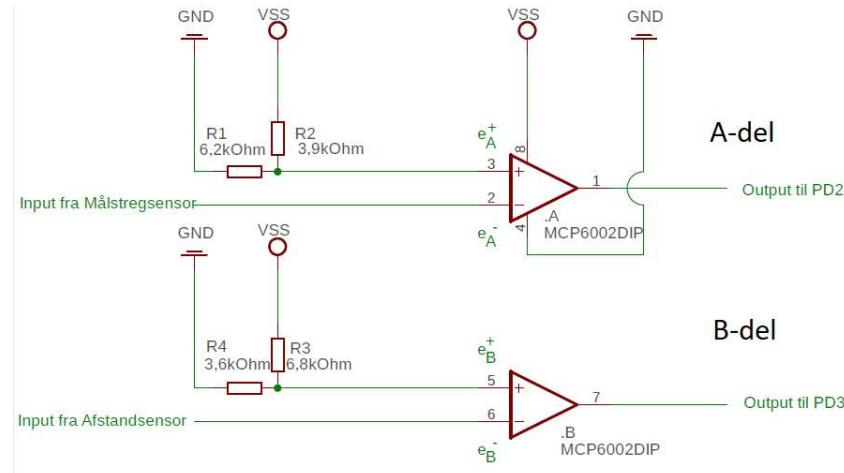
Ud fra disse beregninger skal Målstregssensoren have to resistorer. Disse resistorer skal henholdsvis have 40% og 60% af spændingen over dem. En resistor på $3,9k\Omega$ og en resistor på $6,2k\Omega$ anvendes, hvor det reelle forhold bliver:

$$\frac{3,9k\Omega}{3,9k\Omega + 6,2k\Omega} \approx 0,39 \approx \frac{2}{5}$$

Og Afstandssensoren skal have to resistorer: Disse resistorer skal hændholdsvis have 32% og 68% af spændingen over dem. En resistor på $6,8k\Omega$ og en resistor på $3,6k\Omega$ anvendes, hvor det reelle forhold bliver:

$$\frac{6,8k\Omega}{3,6k\Omega + 6,8k\Omega} \approx 0,65 \approx \frac{3,25}{5}$$

Kredsløbet konstrueres således:



Figur 16: Kredsløbet for de to op-amps til Målstregssensoren og Afstandssensoren.

MCP6002DIP (se databladet under elektronisk bilag) er en dual Rail-to-Rail operationsforstærker, hvor Målstregssensoren forbindes til A-delen, og Afstandssensoren forbindes til B-delen.

Når inputtet fra Målstregssensoren og Afstandssensoren er under den ønskede spændingsdel vil operationsforstærker A og/eller operationsforstærker B udsende høj spænding til hændholdsvis PD2 og PD3, som er porte (external interrupts) på mikrocontrolleren.

Når høj spænding udsendes til PD2 eller PD3 vil der ikke udføres et interrupt, da interrupts er aktivt lave. Hvis spændingen derimod er lav, aktiveres et interrupt.

Referencespændingen over R1 er:

$$e_A^+ = \frac{R1}{R1 + R2} \cdot 5V = \frac{6,2k\Omega}{6,2k\Omega + 3,9k\Omega} \cdot 5V = 3,07V$$

... hvilket er den spænding, som inputtet skal overstige for at aktivere et interrupt.

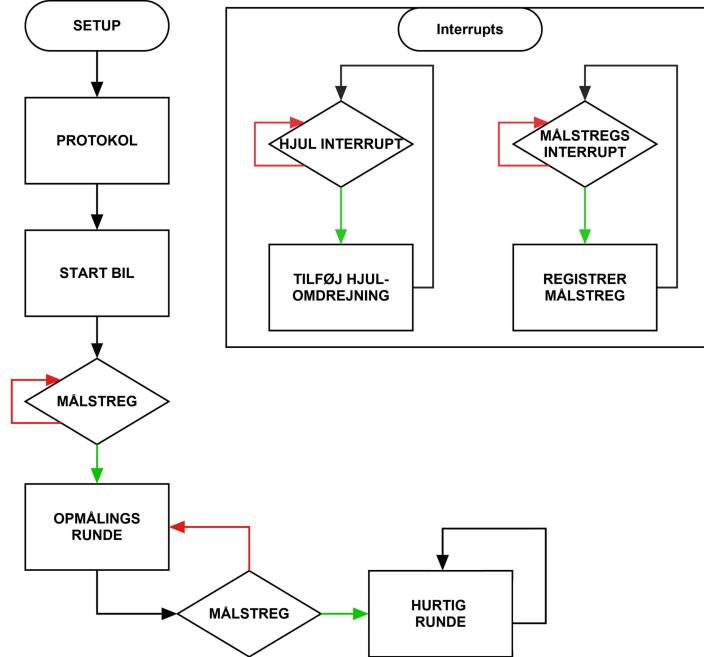
Det samme for R4:

$$e_B^+ = \frac{R4}{R4 + R3} \cdot 5V = \frac{3,6k\Omega}{3,6k\Omega + 6,8k\Omega} \cdot 5V = 1,73V$$

Når disse referencespændinger overstiges, vil det specifikke interrupt aktiveres.

6.4 Flowdiagram

Koden opererer ud fra nedenstående simplificerede flowdiagram. Et mere uddybet flowdiagram er udarbejdet, og kan ses under det elektroniske bilag (*Flowdiagram*).



Figur 17: Simplificeret flowdiagram for Assembler-koden. Rektangler symboliserer handlinger, f.eks. eksekvér ”hurtig runde”. Rhomber symboliserer sammenligninger, eller branches, hvor rød er false, og grøn er true.

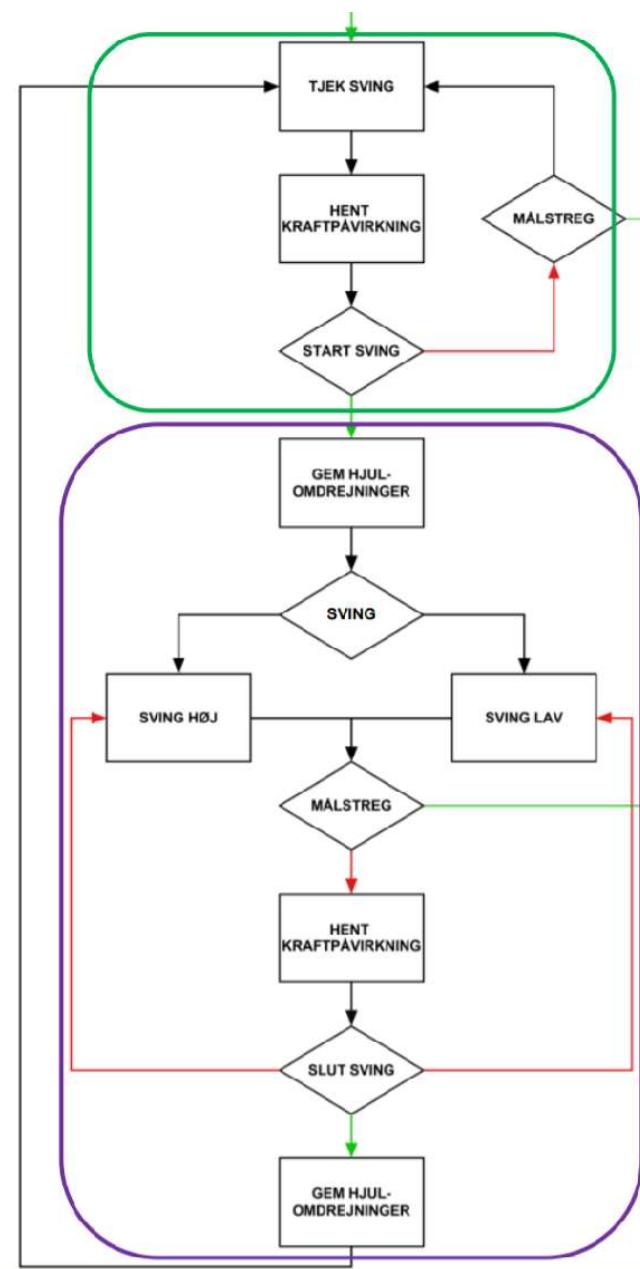
Som det fremgår af flowdiagrammet, startes der med at lave en setup af alle nødvendige pins og porte på bilen. Dernæst ventes der kommandoen START fra C++ programmet. Når denne kommer startes bilen til en vilkårlig hastighed, og interrupts'ene bliver aktive. Bilen tjekker derefter hele tiden om målstregen er nået. Når målstregen opfanges begynder bilen at kortlægge banen. Dette uddybes i afsnittet: *Kortlægning af bane*. Når målstregen opfanges, nu for anden gang, begynder bilen på en hurtig runde, hvor kodelementet ”HURTIG RUNDE” eksekveres og gentages hver gang målstregen detekteres. ”HJUL INTERRUPT” eksekveres hver gang Afstandsensoren registrerer farven hvid på hjulet, og antallet inkrementeres. Denne værdi gemmes i et 8-bits register. Hjulet har to hvide områder for øget præcision. Med en hjulomkreds på ca. 8 cm, kan mikrocontrolleren gemme en distance op til:

$$\frac{255 \cdot 8\text{cm}}{2} = 1020\text{cm} \approx 10m$$

Denne grænse kunne hæves betydeligt, hvis de halve hjulomdrejninger i stedet gemmes i to registre (16 bit), men dette vurderes ikke at være nødvendigt.

Kortlægning af bane

Kortlægningen af banen kan opdeles i to loop: et for lige strækninger markeret med grønt, og et for sving markeret med lilla.



Lige Strækning

”TJEK SVING”:

En gennemsnitsværdi hentes fra accelerometeret, og hvis denne ikke er indeholdt i [110..150[fortsættes der til lilla sektion. Hvis værdien fra accelerometeret er indeholdt i det fornævnte interval tjekkes der om målstregen er registreret, og hvis det er tilfældet, forlades looptet (se *Flowdiagram* i elektronisk bilag). Ellers returneres der til ”TJEK SVING”.

I to tilfælde vil ”Lige Strækning” eksekveres:

1. Hvis bilen lige er startet, og målstregen er blev opfanget (Grøn pil oppefra).
2. Et sving er færdigt (pil fra venstre).

Denne kodedel vil terminere, hvis enten et sving begynder, eller hvis målstregen opfanges på strækningen.

Sving

Hvis et sving er begyndt, gemmes hjulomdrejningerne tilhørende strækningen, der lige er foretaget.

Dernæst tjekkes om det er et højre- eller venstresving; venstresving kaldes "SVING LAV" og højresving, "SVING HØJ". "SVING HØJ" og "SVING LAV" sætter den værdi der senere sammenlignes med accelerometer gennemsnittet i "SLUT SVING".

Derefter tjekkes om målstregen er registreret, hvilket ikke burde forkomme i et sving. Dette fungerer som fejsikring i forbindelse med forstyrrelse på accelerometeret, omkring målstregen.

Sidst hentes igen data fra accelerometeret for at tjekke om svinget er afsluttet. Hvis svinget er afsluttet gemmes hjulomdrejningerne for svinget som en delstrækning. Ellers startes loopet forfra. "SVING" vil kun startes når et sving er påbegyndt.

6.5 Hastighed i sving

Den maksimale hastighed i sving er forskellig alt efter længden af svinget. Igennem test er følgende resultater for hastigheder i forskellige størrelser sving fundet:

Sving størrelse	Hastighedsprocent
45°	39,22%
90°	37,65%
135°	37,65%
180°	34,51%

Tabel 1: Resultater fra *Journal hastighedsforskelse i sving*

Disse værdier er dog ikke de anvendte værdier. Den lange bremselængde som er beregnet (se *journal til bremselængde* i elektroniske bilag) forårsager, at hastigheden sat i koden er nødt til at være lavere end den beregnede. Dette medfører, at bremselængden skal medregnes i den endelige hastighed, for at bilen kan nå at retardere inden den når ind i svinget. Derfor er hastigheden for sving sat lavere i koden (de endelige værdier er tilpasset ud fra "trial and error").

6.6 Hurtig runde

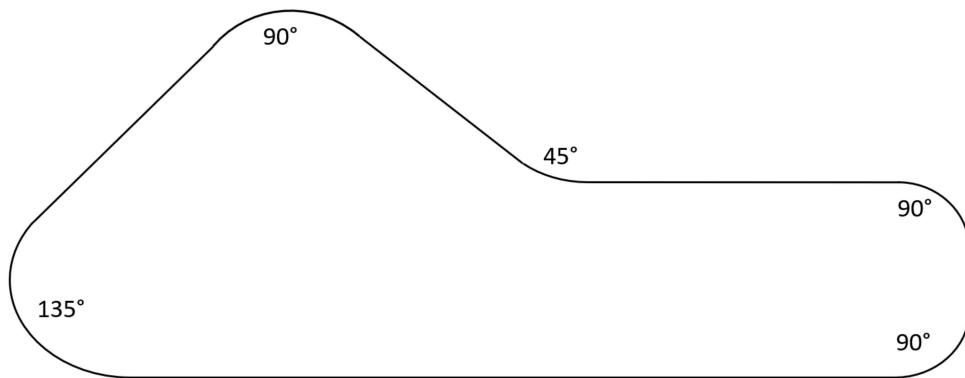
Bilen kører hurtigst uden det udleverede karrosseri, og derfor er test udført uden dette (se afsnit 8, Valg af karosseri).

Efter banen er opmålt påbegyndes den hurtige omgang, hvilket realiseres ved at indlæse den første gemte strækning og fratrække bremselængden. Herefter sættes hastigheden til 160 (63% af 255). Bilen kører ved denne hastighed indtil den lige strækningen er gennemført. Når strækningen er gennemført indlæses næste strækning (sving) og bremselæng-

den ligges til for at længden af banen forbliver den samme. Hastigheden sættes nu til 90 (35% af 255) og bilen kører ved denne hastighed indtil svinget er gennemført. Koden vil på denne måde oscillere mellem en lige strækning og et sving indtil bilen registrerer målstregen igen.

6.7 Omgangstid

For at bekræfte, at den nu autonome bil er hurtigere, end før den blev styret af programmet, køres der to runder på en opstillet bane med både højre- og venstresving (se figur 18).



Figur 18: Skitse af banen der er testet på, med indikeret vinkel på sving.

Ved en maksimal konstant hastighed på ca. 43% (se *Journal til hastighedsforskelse i sving*) rundt på banen, blev gennemsnittet af omgangstiden over tre omgange 4,87 sekunder. Med koden blev gennemsnittet af tre omgange 4,61 sekunder, resulterende i en forbedring på 5,58%.

6.8 Delkonklusion

Mikrocontrolleren vil konstant få data ind fra de tre sensorer, og behandle disse som nødvendigt. Den vil jævnligt tjekke om der kommer nye kommandoer udefra via kommunikationsprogrammet. Når bilen bliver bedt om at starte, vil den først køre en runde for at kortlægge banen. Dernæst kører den så hurtigt som muligt rundt på banen ud fra kortlægning, indtil den bliver bedt om at stoppe igen.

Efter autonomisering af bilen er omgangstiden forbedret med 5,58%.

Optimering af omgangstid afhænger udeover koden også af bilens fysiske form og egenskaber. Dette undersøges i næste afsnit.

7 Design af karrosseri

Der er i afsnittet anvendt formler og teori fra bøgerne, "Race Car Aerodynamics"^[6] og "Vejen til Fysik A2"^[7].

For at optimere designet af karrosseriet inden for to hovedpunkter, acceleration og svinghastighed, vil de forskellige aspekter undersøges i de følgende afsnit.

Som følge af beslutningerne taget i sensorafsnittet (se afsnit 5, Sensorer) og det resulterende udseende af bilen efter monteringen af sensorerne (se sektion 5.5, Delkonklusion), skal karrosseriet kunne rumme sensorerne.

7.1 Acceleration

For at opnå størst mulig acceleration vil to fysiske fænomener, som har indflydelse behandles: karrosseriets masse og luftmodstand.

Masse

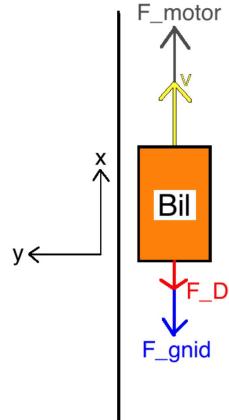
For at bilen kan opnå den højeste acceleration, er det vigtigt at forstå Newtons 2. lov:

$$\begin{aligned}F_{res} &= m \cdot a \Leftrightarrow \\a &= \frac{F_{res}}{m}\end{aligned}\tag{1}$$

Acceleration afhænger af den resulterende kraft og bilens masse, og hvis den højeste acceleration skal opnås, skal massen være så lille som mulig.

Luftmodstand

For at forklare luftmodstanden, beskrives den resulterende kraft i x-retningen (kørselsretningen).



Figur 19: Illustrering af bilen set oppe fra, hvor motorkraften F_{motor} , gnidningskraften F_{gnid} , luftmodstanden F_D og hastigheden v er indtegnet.

Ud fra illustrationen kan den resulterende kraft i x-retningen opskrives:

$$F_{res_x} = F_{motor} - F_D - F_{gnid} \quad (2)$$

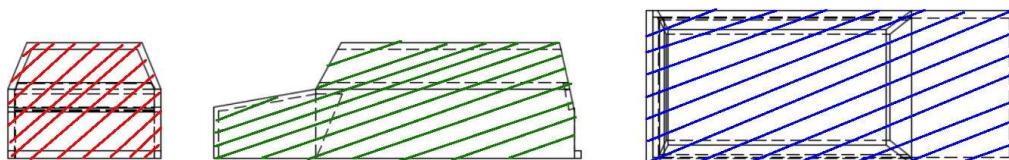
Hvor F_{motor} er kraften fra motoren, F_D er luftmodstanden og F_{gnid} er gnidning mellem bilen og vejbanen. Den højeste resulterende kraft kan opnås, hvis luftmodstanden formindskes, eftersom man vil opnå den højeste resulterende kraft, se ligning (1).

Luftmodstand er defineret på følgende måde:

$$F_D = \frac{1}{2} \cdot \rho \cdot C_D \cdot A_X \cdot v^2$$

Hvor ρ er densiteten for luft, C_D er formfaktoren, A_X (se figur 20) er tværsnitsarealet og v er bilens hastighed.

Det betyder, at den højeste acceleration opnås ved at reducere bilens tværsnitsareal og formfaktor. Formfaktoren afhænger af bilens form; desto lettere bilen kan passere gennem luften desto mindre formfaktor.



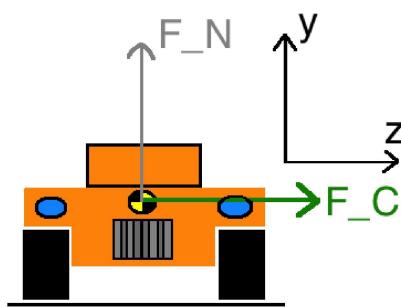
Figur 20: Illustrering af bilens forskellige tværsnitsarealer. A_X , A_Y og A_Z

7.2 Svinghastighed

Det næste optimeringsmål er at køre gennem sving med højst mulig hastighed, dermed forkortes omgangstiden.

Massemidtpunkt

Bilen har en skøjte (se figur 1) nede i banen for at være i stand til at følge banen rundt. Så længe skøjten er nede i sit spor, vil bilen ikke kunne blive afsporet. Gnidningskraften mellem bilen og banen har derfor ingen indflydelse på svinghastigheden. Bilen vil først blive afsporet når bilen tipper af banen; derfor skal bilens massemidtpunkt undersøges. I et sving bliver bilen påvirket af normalkraften samt centrifugalkraften. Disse kræfter visualiseres på figur 21:



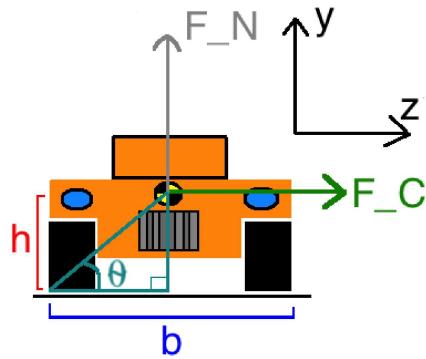
Figur 21: Illustrering af bilen set forfra hvor bilen befinner sig i et sving, indtegnet med masse midtpunkt og centrifugalkraften (F_C) og normalkraften (F_N) i et koordinatsystem.

Den resulterende kraft i y-retningen og z-retningen opskrives ud fra illustreringen:

$$F_{res_y} = F_C$$

$$F_{res_z} = F_N$$

Punktet lige før bilen vælter, kan bestemmes via trigonometri med en retvinklet trekant.



Figur 22: Illustrering af bilen set forfra, hvor bilen befinder sig i et sving, indtegnet med massemidtpunkt, normalkraften (F_N) og centrifugalkraften (F_C) i et koordinatsystem, samt højden af bilens massemidtpunkt (h) og bilens afstand mellem hjul (b). Derudover er vinklen θ indtegnet ved bilens ydrepunkt på hjulet til massemidtpunktet.

Ud fra illustreringen bestemmes punktet lige før bilen vælter på følgende måde:

$$F_C \cdot \tan(\theta) = F_N$$

Bilen vil derfor vælte når følgende er opfyldt:

$$F_C \cdot \tan(\theta) > F_N$$

Derfor skal vinklen θ være så lille som muligt for at mindske risikoen for at bilen vælter. Vinkel θ kan bestemmes på følgende måde med bilens højde og bredde (se figur 22):

$$\begin{aligned} \theta &= \tan^{-1} \left(\frac{h}{\frac{1}{2} \cdot b} \right) \Leftrightarrow \\ &= \tan^{-1} \left(\frac{2 \cdot h}{b} \right) \end{aligned}$$

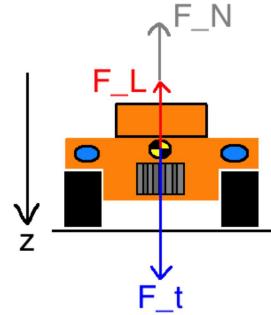
Derefter opstilles følgende ligning, som beskriver, hvornår bilen vælter afhængig af bilens bredde og højden på massemidtpunktet:

$$\begin{aligned} F_C \cdot \tan(\theta) &= F_N \\ F_C \cdot \tan \left(\tan^{-1} \left(\frac{2 \cdot h}{b} \right) \right) &= F_N \Leftrightarrow \\ F_C \cdot \frac{2 \cdot h}{b} &= F_N \end{aligned} \tag{3}$$

For at minimere risikoen for at bilen tipper i et sving, skal bilens hjulafstand forøges og massemidtpunktets højde skal ligges så lavt som muligt.

Normalkraft

F_N er bilens normalkraft og afhænger af den resulterende kraft i z-retningen.



Figur 23: Illustrering af bilen set forfra med normalkraft (F_N), tyngdekraft (F_t) og liftkraften (F_L) og massecentrum indtegnet i et koordinatsystem.

Ud fra illustreringen kan den resulterende kraft i z-retningen opskrives:

$$\begin{aligned} F_N &= F_{res_z} \\ &= F_t - F_L \end{aligned}$$

Hvor F_t er tyngdekraften og F_L er liftkraften. Tyngdekraften er defineret på følgende måde:

$$F_t = m \cdot g$$

Hvor m er bilens masse og g er tyngdeaccelerationen.

Liftkraften er den kraft, der løfter bilen op fra vejbanen, og er defineret på følgende måde:

$$F_L = \frac{1}{2} \cdot \rho \cdot C_L \cdot A_Z \cdot v^2$$

Hvor ρ er densiteten for luft, C_L er koefficienten for lift, A_Z (se figur 20) er tværsnitsarealet i z-retningen og v er bilens hastighed. Normalkraften F_N kan nu opskrives:

$$\begin{aligned} F_N &= F_t - F_L \\ &= (m \cdot g) - \left(\frac{1}{2} \cdot \rho \cdot C_L \cdot A_Z \cdot v^2 \right) \end{aligned} \tag{4}$$

Til sidst opskrives udtrykket for centrifugalkraften, F_C , som er defineret på følgende måde:

$$F_C = \frac{m \cdot v^2}{r} \tag{5}$$

Hvor m er bilens masse, v er hastigheden og r er svingets radius.

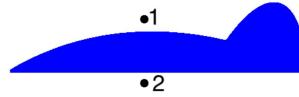
Maksimal svinghastighed

Der kan nu bestemmes et samlet udtryk for den maksimale hastighed i et sving, ved at bruge formel (3), (4) og (5):

$$\begin{aligned}
 F_C \cdot \frac{2 \cdot h}{b} &= F_N \\
 F_C \cdot \frac{2 \cdot h}{b} &= F_t - F_L \\
 \left(\frac{m \cdot v^2}{r} \right) \cdot \frac{2 \cdot h}{b} &= (m \cdot g) - \left(\frac{1}{2} \cdot \rho \cdot C_L \cdot A_Z \cdot v^2 \right) \Leftrightarrow \\
 v &= \sqrt{\frac{2m \cdot g \cdot r \cdot b}{4 \cdot m \cdot h + \rho \cdot C_L \cdot A_Z \cdot r \cdot b}}
 \end{aligned} \tag{6}$$

For at forøge bilens svinghastighed, skal bilens hjulafstand (se b i figur 22) forøges og massemidtpunktet ligge så lavt som muligt. Da afstanden mellem bilens hjul ikke kan ændres, er det kun bilens massemidtpunkt, der kan justeres. Liftkoefficienten, C_L , tilnærmes 0, eller hvis muligt, negativ. Dette opnås ved at skabe højt tryk foroven og lavt tryk forneden. Bernoullis princip forklarer, hvordan højt og lavt tryk skabes:

$$\frac{p_1}{\rho} + \frac{(v_1)^2}{2} = \frac{p_2}{\rho} + \frac{(v_2)^2}{2}$$



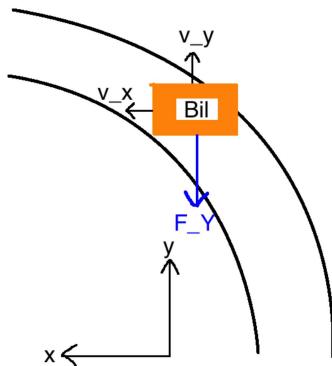
Figur 24: Illustrering af Bernoullis Princip.

Hvor p_1 og p_2 er tryk, ρ er densitet og v_1 og v_2 er hastigheder. Ved at benytte Bernoullis princip kan trykket styres ved at ændre hastighederne. Derfor skal hastigheden for luften foroven være så lav som mulig. Omvendt forneden, skal luften bevæge sig så hurtigt som muligt for at opnå en negativ liftkoefficient, C_L .

Udover at opnå den højeste svinghastighed og acceleration, vil det også være en fordel at forhindre bilen i at skride ud i sving. Dette kan modvirkes af sidekraften; derfor undersøges denne.

Sidekraften

Hvis bilen laver en udskridning, har bilen en hastighed i både x-retningen, v_x , og y-retningen, v_y (grundet udskridning); derved er der også en luftmodstand i begge retninger. Luftmodstanden i y-retningen kaldes for sidekraften.



Figur 25: Illustrering af bilen set oppe fra i et sving med sidekræften F_Y samt hastigheden i både x- og y-retningen indtegnet i et koordinatsystem.

Det kan ses på illustreringen, at sidekraften har retning ind mod svingets centrum. Sidekraften presser derfor bilen tilbage ind i svinget. For at øge chancen for at bilen kan rette op igen, skal bilens sidekraft være så stor som mulig. Sidekraften er defineret på følgende måde:

$$F_Y = \frac{1}{2} \cdot \rho \cdot C_Y \cdot A_Y \cdot v_y^2$$

Hvor ρ er densiteten for luft, C_Y er formfaktoren for bilen i y-retningen, A_Y er tværsnitsarealet (se figur 20) i y-retningen og v_y er bilens hastighed i y-retningen. For at forøge sidekraften skal formfaktoren og tværsnitsarealet forstørres.

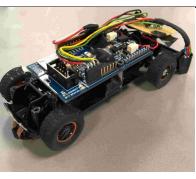
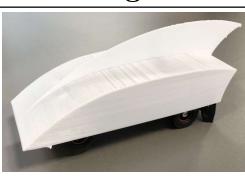
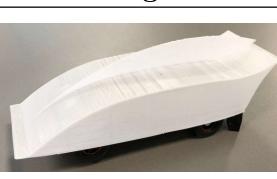
7.3 Delkonklusion

For at designe det bedste karrosseri fokuseres på følgende punkter:

- Mindst mulig vægt for at maksimere accelerationen (se ligning (1)) samt lille luftmodstand (F_D) (se ligning (2)).
- For at opnå højeste svinghastighed skal massemidtpunktet ligge så lavt som muligt (se ligning (3)) samt lav eller negativ liftkoefficient, C_L (se ligning (6)).
- Høj sidekraft (F_Y) for at hindre udskridning.

8 Valg af karrosseri

Ud fra delkonklusionen (se sektion 7.3) designes to forskellige karrosserier:

Uden karrosseri	Udleveret karrosseri	Design 1	Design 2
			

Tabel 2: Billede af karrosserierne som vil blive sammenlignet.

Karrosserierne sammenlignes på følgende punkter.

- Masse
- Luftmodstand
- Liftkraft
- Sidekraft

Masse

Resultat fra vejning af karrosserierne:

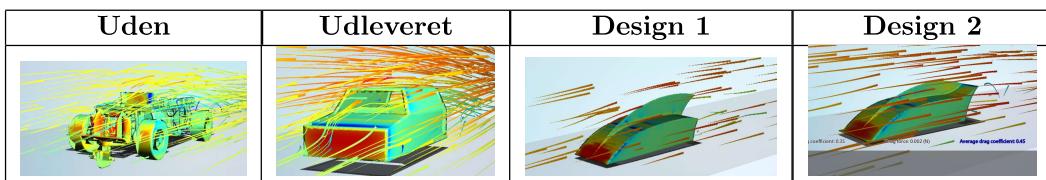
Karrosseri	Masse [g]
Uden	0
Udleveret	22,6
Design 1	34,9
Design 2	35,7

Tabel 3: Resultat af vejningen.

Design 1 og 2 har størst masse, hvilket medfører, at bilen med disse karrosserier opnår en lavere acceleration, eftersom motorkraften er konstant, og der er en større masse, der skal accelereres (se ligning (1)). Bilen har også et højere massemidtpunkt (se ligning (3)).

Luftmodstand

Formfaktoren C_D af de fire modeller i x-retningen (kørselsretningen) bestemmes i CFD-programmet Autodesk Flow Design.



Tabel 4: Simulering af luftmodstand

Luftmodstandsområdet bliver bestemt på følgende måde: Luftmodstandsområde = $C_D \cdot A_X$

Design	Formfaktor C_D	Tværsnitsareal A_X [cm ²]	Luftmodstandsområde [cm ²]
Uden	0,29	17,98	5,21
Udleveret	0,65	27,60	17,94
Design 1	0,29	24,50	7,10
Design 2	0,35	30,10	10,53

Tabel 5: Resultat af simuleringen af luftmodstand

Ud fra simuleringen konkluderes, at luftmodstandsområdet er størst ved det udleverede karrosseri. Af de to designede karrosserier har Design 2 det største luftmodstandsområde, eftersom Design 2 har en front splitter og en bagvinge, som øger luftmodstanden.

Liftkraft

I dette afsnit arbejdes der med lift, hvortil der ses på toppen af karrosseriet. Dertil menes der med tværsnitsarealet A_Z , at det er arealet af toppen af karrosseriet (se figur 20). Det antages at massemidtpunktets placering er lig med følgende:

$$\frac{2 \cdot h}{b} = 1$$

For at bestemme koefficienten for lift, C_L , benyttes følgende ligning, som er udledt ud fra ligning (6):

$$C_L = -\frac{2(2 \cdot v^2 \cdot m - g \cdot m \cdot r)}{v^2 \cdot \rho \cdot A_Z \cdot r}$$

Hvor hastigheden v er den maksimale hastighed i et sving. Et forsøg gennemføres ved at opsætte en bane i en cirkulær form og bilens hastighed inkrementeres indtil bilen falder af banen, og hastigheden noteres. Liftkraftsområdet bliver bestemt på følgende måde: Liftkraftsområde = $C_L \cdot A_Z$

Karrosseri	Liftkoefficient	Tværsnitsareal A_Z [cm ²]	Liftkraftsområde [cm ²]
Uden	-1,00	60,28	-60,28
Udleveret	3,59	86,01	308,74
Design 1	12,92	115,73	1495,57
Design 2	44,72	127,80	5715,30

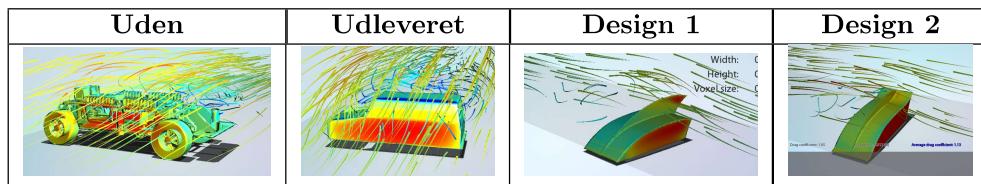
Tabel 6: Resultat fra *Journal til bestemmelse af koefficienten for lift CL for karrosseri*.

Forsøget viser, at liftkraften trykker ned på bilen uden karrosseri. Derfor kan bilen køre hurtigere rundt i et sving uden karrosseri. Årsagen til at liftkoefficienten (C_L) er positiv for karrosserierne, skyldes luften trænger ind under karrosseriet, og løfter op i bilen. De to designede karrosserier er ikke aflukket i bunden; derfor kommer der luft ind fra bunden og trækker op i karrosseriet, hvilket gør koefficienten højst ved disse. Design 2 har

det højeste liftkraftsområde, da den har det største tværsnitsareal. Hvis de to designede karrosserier havde været sluttet tæt om undervognen kunne de give bedre resultater. Resultaterne er ikke 100% præcise, eftersom det antages, at massemidtpunktets placering ikke har indflydelse på den maksimale svinghastighed. Forsøget giver en indikation om liftkoefficienten (C_L) er negativ eller positiv. For at se, hvordan testene er udført (se *Journal til bestemmelse af koefficienten for lift CL for karrosseri* i elektronisk bilag).

Sidekraft

Formfaktoren af de fire modeller i y-retningen bestemmes i CFD-Programmet Autodesk Flow.



Tabel 7: Simuleringen af sidekraft

Sidekraftsområde bliver bestemt på følgende måde: Sidekraftsområde = $C_Y \cdot A_Y$

Karrosseri	Formfaktor C_Y	Tværsnitsareal $A_Y [cm^2]$	Sidekraftsområde $[cm^2]$
Uden	0,82	84,94	69,65
Udleveret	1,08	84,60	91,37
Design 1	1,24	97,80	121,27
Design 2	1,05	90,00	94,50

Tabel 8: Resultat af simuleringen af sidekraft

Design 1 har det største sidekraftsområde, pga. finnen på karrosseriet er den største, og har derfor størst mulighed for at rette op, hvis bilen laver en udskridning.

8.1 Delkonklusion

På baggrund af undersøgelserne af de forskellige modeller, viser resultaterne, at bilen uden karrosseri vil give den hurtigste omgangstid, da det ingen masse tilfører, og derved også har det laveste massemidtpunkt. Hvis et karrosseri skal vælges, vil det være Design 1, da det leverede de bedste resultater ift. luftmodstandsområde, liftkraftsområde og sidekraftsområde end Design 2. Design 1 leverede også bedre resultater end det udleverede karrosseri ift. luftmodstandsområde og sidekraftsområde. Selvom det udleverede karrosseri har en lavere vægt, hvilket resulterer i større acceleration, er der ikke plads til sensorerne. For Design 2 gælder det, at den er dårligere i alle kategorier.

9 Konklusion

Dette afsnit omfatter konklusion af problemformuleringen (se sektion 3).

Det er muligt at oprette en Bluetooth forbindelse, og derved sende og modtage data til/fra mikrocontrolleren. Målstregsenoren, Afstandsensoren og Svingsensoren giver en samlet funktion til at kortlægge en ukendt bane. I en hurtig runde kører bilen ca. 63% af den maksimale hastighed på langsider og i sving ca. 35%. For ikke at påbegynde svinget med for høj hastighed, skiftes der til svinghastighed inden svinget er nået. Derfor sættes langsiden-hastigheden ikke højere, da det vil kræve en større bremselængde. Med kodden implementeret kører bilen ca. 5,6% hurtigere på en bestemt bane (se figur 18 i sektion 6.7) end hvis bilen kørte med en maksimal konstant hastighed på 43% rundt på hele banen uden at bilen tipper af.

- Der er valgt at benytte et analogt accelerometer til at detektere sving. Accelerometeret står på højkant, hvilket betyder, at det er monteret på siden af bilen i stedet for under karosseriet. Det er muligt at kende forskel på højre- og venstresving, men der tages ikke højde for dette, da samme hastighed anvendes. Der tages ikke forbehold for ydre- og indre-sving, da accelerometeret ikke har en tilstrækkelig præcision.
- Der benyttes en reflektiv sensor til at detektere målstregen. Datahastigheden fra sensoren er 1 MHz, hvilket gør, at målstregen aldrig vil overses. Falling edge benyttes for registrering af målstregen efter den passeres.
- For at måle banens længde optælles antallet af hjulomdrehninger. Dette sker ved at pege en reflektiv sensor mod hjulet og klippe to stykker hvid tape overfor hinanden på dette hjul. Mikrocontrolleren vil derfor kende længden af hver strækning i antal halve hjulomdrehninger.
- De to reflektive sensorer aktiverer hvert sit interrupt, der kører hver sin underroutine. Data fra accelerometeret konverteres til et digitalt signal via ADC'en. PWM anvendes til at sætte en bestemt hastighed ud fra kortlægningen af banen.
- Etablering af Bluetooth-forbindelse mellem en PC og bilen er realiseret vha. af et program i C++. Kommunikationsprotokollen opfyldes til et minimum; den indeholder ikke REPLY-funktionen. Data fra sensorerne hentes via et tredjeparts terminalprogram.

10 Perspektivering

Dette afsnit omfatter idéer til ændringer og forbedringer af projektet. Forbedringen til højre hænger sammen med resultatet fra projektet til venstre.

Resultat

1. Der kan kommunikeres med bilen fra pc med den ønskede kommunikations protokol.
2. Bremsning af bilen sker ved at ændre hastigheden til en lavere hastighed.
3. Svingsensoren og Assembler-kodens behandling af forskellige størrelser sving.

Forbedring

1. Tilføje funktionalitet til protokol for at hente specifik sensor data.
2. .a Tilføje elektromagnet.
.b Tilføje bremsekredsløb.
3. Tilføje speciel hastighed for størrelsen af sving.

1. Forbedring

Til bilen er der knyttet flere forskellige analoge sensorer (se afsnit 5); alle sammen nogen der afgiver data. Denne data bruges til at bestemme, hvordan bilen opfanger banen, og hvordan den skal køre ud fra det. For at få fat i denne data anvendes kommunikationsprotokollen. Som forbedring til denne kommunikationsprotokol vil tilføjelsen af en funktion, der kan trække data ud fra en specifik sensor, fremfor alle sensorene, være en simplificering for forståelsen af, hvilke data der arbejdes med.

2.a Forbedring

For at maksimere perioden, hvor bilen kan køre hurtigst muligt, kan tilføjelsen af en elektromagnet være behjælpelig. På langsiderne skal bilen køre maksimal hastighed hele tiden. Når den skal ind i et sving, skal bilen have en lavere hastighed for ikke at skride af banen. En elektromagnet vil forkorte den nødvendige bremselængde, som skal til for at bilen kan nå at nedsætte hastigheden før et sving. Den nuværende bremselængde er lang og desto større hastigheden bliver, desto længere bliver bremselængden (se *Journal til bremselængde* i elektronisk bilag).

Elektromagneten har nogle problematikker. Elektromagneten består af en jernkerne samt kobbertråd viklet om kernen. Ved tilføjelse af elektromagneten vil der tilføjes vægt til bilens samlede masse. Tilmed vil elektromagneten trække meget strøm. Det kan resultere i store spændingsfald på banen.

Disse problematikker vil tilsammen resultere i en lavere tophastighed og acceleration.

2.b Forbedring

Et bremsekredsløb kan konstrueres på to måder: Bremselængden kan forkortes ved at kortslutte motoren. Ved at kortslutte motoren vil resistensen i motoren medføre en opbremsning.

Bremselængden kan også forkortes ved at vende spændingen. Ved at vende spændingen over motoren, vil der påføres en kraft til motoren i den modsatte retning, hvilket vil medføre en opbremsning.

3. Forbedring

En optimering af bilens hastighed, ville forekomme ved tilføjelse af specielle funktioner for sving alt efter størrelsen af svinget. Ifølge (se journal, *Hastighedsforskel i sving* i elektronisk bilag) vil der kunne opnås en hastighedsoptimering i sving på 12,01 procent ved at tage højde for forskellen i et 180 graders sving og et 45 graders sving. Dataene fra Afstandsensoren gør dette muligt ved at sortere sving efter længden af sving.

11 Litteraturliste

- [1] Mikrocontroller. Udgivet af Wikipedia. Internetadresse:
<https://da.wikipedia.org/wiki/Mikrocontroller> - Besøgt d. 27.03.2018
- [2] ATmega32. Udgivet af Microchip. Internetadresse:
<https://www.microchip.com/wwwproducts/en/ATmega32> - Besøgt d. 27.03.2018
- [3] Allan R. Hambley, Electrical Engineering Principles and Applications 6. udgave, 2013
- [4] Ali Mazidi, Muhammad m.fl.: The AVR Microtroller and Embedded System, 1. udgave, Pearson, 2011.
- [5] Pulse Width Modulation. Udgivet af Sparkfun. Internetadresse:
<https://learn.sparkfun.com/tutorials/pulse-width-modulation> - Besøgt d. 14.05.2018
- [6] Joseph Katz, Race Car Aerodynamics Designing for Speed, 2. udgave, 1995
- [7] Knud Erik Nielsen og Esper Fogh, Vejen til Fysik A2, 2. udgave, 2011

12 Bilag

12.1 Oversigt over elektronisk bilag

Dette bilag omhandler beskrivelse af indholdet på det elektroniske bilag.

Det beskrives her i alfabetisk rækkefølge:

Assembler-kode:

- main.asm

C++ kode:

- LinuxSerialPort.cppx
- main læsbart i Word.docx
- main.cpp
- SerialPort.cpp
- SerialPort.hpp
- WinSerialPort.cppx

Datablade:

- Accelerometer - MMA1270KEG.pdf
- ATmega32A.pdf
- IR-reflektiv sensor - CNY70.pdf
- Fototransistor - OP593.pdf
- IR-diode - OP298.pdf
- Kondensator - LHK Series General Purpose.pdf
- Operationsforstærker - MCP6002.pdf
- ToyRaceCar_V2_REV3.pdf

Flowchart:

- Flowdiagram med kommentering.pdf

- Flowdiagram.png

Journaler:

- Journal til bestemmelse af koefficienten for lift CL for karosseri.pdf
- Journal til bestemmelse af koefficienten for lift CL for karosseri.xlsx
- Journal til bremselængde.pdf
- Journal til hastighedforskelsving.pdf
- Journal til måling af hastighed.pdf
- Journal til Målstregsensor og Afstandssensor.pdf
- Liftkoefficienten.nb

Model af racerbil:

- Model af racerbil.html

Projektoplæg:

- Pole Position f2018.pdf

Tidsplan:

- Tidsplan 1. version.pdf
- Tidsplan 2. version.pdf