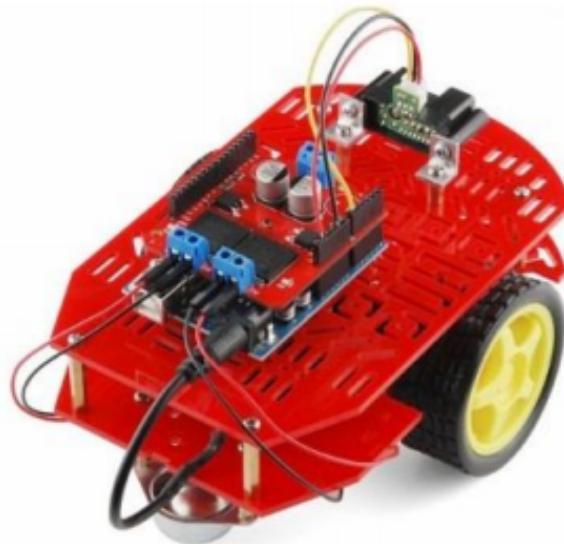


Forårs Semester 2016

Line following robot

Gruppe 4

2. Semester IT-Teknolog



Gruppe medlemmer: Anders Pedersen - Kasper Delfs - Kristian Porsborg

Vejledere: Jesper M. Kristensen og Steffen Vutborg

2. Semester
IT-teknolog
Sofiendalsvej 60
9200 Aalborg SV
<http://www.ucn.dk/>

Titel:

Line following automotive
robot

Projekt Periode:

2. Semester | Forårs semester 2016

Projectgruppe:

Gruppe 4

Medvirkende:

Anders Pedersen

Kasper Delfs

Kristian Porsborg

Vejleder:

Jesper M. Kristensen og

Steffen Vutborg

Sideantal: TBD ¹

Appendiks: TBD ²

Færdiggjort: 7/6-2016

Forord

I dette semester projekt skal en robot bygges til at følge en linje. Ud fra problem-formulering er noget hardware stillet til rådighed hvor gruppens formål vil være at implementere den nødvendige hard- og software, så robotten vil være i stand til at manøvre rundt på en oplagt linje. Produktet udvikles af it-teknologstuderende fra University College Nordjylland på 2. Semester på elektronik linjen. Produktet udvikles for at øge kompetencer og forståelser inden for allerede kendt eletorik- og programmerings viden med forbehold for at anvende det i praksis.

Anders Pedersen

Kasper Delfs

Kristian Porsborg

Indholdsfortegnelse

| | | |
|----------|---------------------------------|-----------|
| 1 | Foranalyse | 1 |
| 1.1 | Indledning | 1 |
| 1.2 | Line Track | 1 |
| 2 | Kravspecifikation | 4 |
| 3 | Hardware | 5 |
| 3.1 | Hardware Overblik | 5 |
| 3.2 | Sensor | 5 |
| 3.3 | Low-Pass filter | 6 |
| 3.4 | Motor | 7 |
| 3.5 | Test | 7 |
| 4 | Software | 8 |
| 4.1 | Software introduktion | 8 |
| 4.2 | Arduino | 9 |
| 4.3 | PID regulering | 10 |
| 4.4 | Test | 10 |
| 5 | Test | 13 |
| 6 | Konklusion | 14 |
| 7 | Perspektivering | 15 |
| | Bibliography | 16 |

Foranalyse 1

1.1 Indledning

Automatisering er en stor del af samfundet den dag i dag. Fabrikker anvender automatisering for at kunne øge produktion og salg.

En stigende automatisering ses især i masseproduktion, heriblandt eksempelvis produktionen af biler.

Sikkerhedsmæssige aspekter bliver udviklet i takt med automatiseringen hvor der i dag ses eksempler på biler som bremser selv hvis den forankørende kommer for tæt. Dette er med til at skabe fokus på blandt andet sikkerhed hvor computerstyring er stærkt tiltagende. Indenfor rammen af automatisering og sikkerhed kan om andre google nævnes med udvikling af deres fuldt automatiseret biler. Google's selvkørende bil (jf. kilde [\[googlebil\]](#))

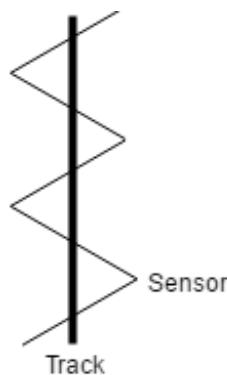
Dette projekt handler om at fremstille et automatiseret produkt i form af en selvkørende bil. Bilen skal injusteres med et line track system, som derved skal få den til at følge en opstillet bane, bestående af en hvid overflade og en sort streg af tape der markerer banens gang.

1.2 Line Track

Den linjefølgende robot følger en bestemt bane ved at anvende princippet om at modtage et konstant signal. Signalet sendes fra lyssensoren med en varierende værdi afhængig af farven på den kørete overflade.

Den forudbestemte bane er defineret ved en sort bane med en minimusbredde på 30 mm omgivet af hvid eller gråtonet omkringliggende farve.

Gruppen har valgt som fremgangsmåde først at anvende én sensor til at sende den værdi som læses fra den kørete overflade.



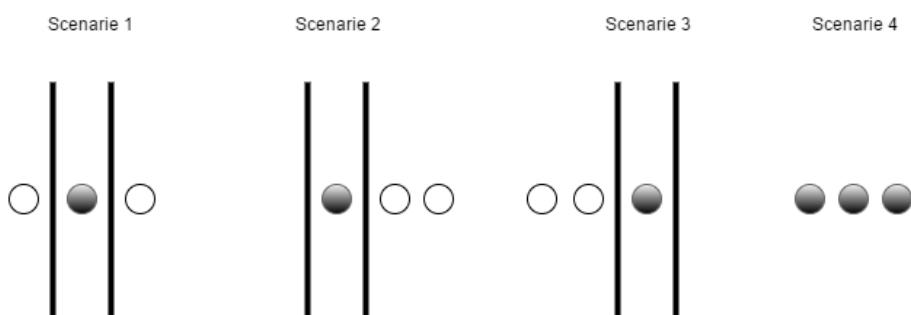
Figur 1.1: Sensor algoritme over én sensor

På ovenstående figur ses det hvordan sensoren registrerer den omkringliggende farve og drejer til højre. Herfra fortsætter bilen indtil den optegnede bane igen registreres og der igen læses en hvidnuanceret farve efter stregen. Da den hvidtonede overflade registreres på ny justerer bilen retning mod venstre for at kunne følge linjen.

Videreudvikling af Line Track

For at gøre bilen i stand til at kunne følge linjen efter bedste og hurtigst mulige evne har gruppen valgt at implementere yderligere sensore. Dette tillader bilen at kunne følge en lige linje samt være i stand til at registrere fremadkomne ændringer i banens forløb.

På nedenstående figur ses hvordan implementering af flere sensorer hjælper bilen i at bestemme banens forløb.



Figur 1.2: Sensoralgoritme over tre implementeret sensorer

Ved anvendelse af tre sensorer skabes fire mulige scenarier som bilen skal være i stand til at forholde sig til.

1. Scenarie

- Her ses det hvordan de tre sensorer tilgår banens forløb. Det antages at bilen følger en lige linje hvor den midterste sensor registrerer den sortnuanceret overflade. De to yderste sensorer registrerer en anden overflade som er ens på begge sider og bilen skal derfor ikke justere retning.

2. Scenarie

- Her registrere venstre sensor banens forløb hvilket indikerer at banens forløb har ændret sig. Den midterste og den højre sensor registrerer begge den hvidtonede overflade og bilen skal derfor justere ind til venstre for at den midterste sensor igen læser den mørke overflade så bilen igen kører en lige linje.

3. Scenarie

- Ligesom 2. scenarie registrerer her én af ydersensorne at banens forløb har ændret sig. Højre sensor registrerer den mørke overflade hvorfra de to andre sensorer registrerer den hvidtonede overflade. Herfra skal bilens retning ligeledes ændres så den midterste sensor læser den mørke overflade igen.

4. Scenarie

- I dette scenaie antages at en uforudset hindring eller hændelse har påvirket sensor eller banens forløb. Dette kan eksempelvis være fejlregistrering fra sensor eller snavs eller misfarvninger på den opstillede bane.
Ingen af sensorerne er i stand til at videresende en registreret måling af den pågældende overflade og bilen kan derved ikke registrere hvor den opstillede bane er. I dette scenarie står bilen derfor stille.

Kravspecifikation 2

I det følgende afsnit gives der indblik i de krav som er sat i problemanalysen samt de krav projektgruppen har sat for at imødegå produktet præsenteret i projektbeskrivelsen.

Generelle krav

1. Projektet skal konstrueres omkring Sparkfun's Magician Chassis hvor to dc motorer er inkluderet.
2. Til projektet skal et Arduino Chipkit UNO32 anvedes baseret på mikroprocessoren PIC32MX 32bit.
3. Produktet skal fremvises og demonstreres til projektevalueringen.
4. Projektet skal dokumenteres i form af en rapport.

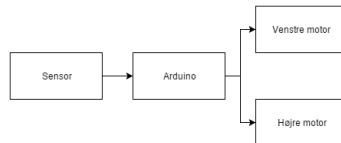
Krav til produktet

1. Produktet skal være i stand til køre på en linje med en minimumsbredde på 30 mm.
 - a) Styring skal foregå ved hjælp af feedback fra en eller flere lyssensorer.
 - b) Farven på linjen skal være sort eller gråtonet over 75%. Den omkring liggende farve skal være hvid eller gråtonet under 50%.
2. Softwaren til produktet skal skrives i MPLAP.

Hardware 3

3.1 Hardware Overblik

I dette afsnit bliver beskrevet hvilket hardware komponenter der blev valgt i forhold til produktet og hvorfor. Endvidere bliver funktionaliteten af de enkelte komponenter afviklet og der gives et overblik over det samlede produkt.



Det ovenstående flowchart viser hvordan produktet fungerer. Sensoren opfanger ved hjælp af linetrack et følsomhedssignal som enten er sort eller hvidt. Dette signal videresendes til arduinoen som konfigurerer signalet og justerer motorene efterfølgende.

3.2 Sensor

Sensoren der anvedes er en "Line Sensor Breakout - QRE1113" fra sparkfun.com. Det er en analog sensor som sidder på et breakout board i en spændingsdeling. Dette betyder at der blot skal aflæses spænding på en pin for at få en værdi der svarer til en lysstyrke fra sensoren.



Specifikationerne gældende for sensoren:

- 5VDC operating voltage
- 25mA supply current
- Optimal sensing distance: 0.125" (3mm)
- Dimension: 0.30 x 0.55 " (7.62 x 13.97 mm)

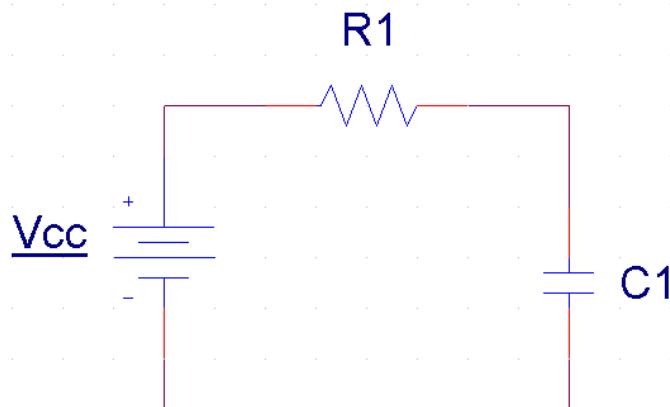
Kilde (Sparkfun.com)

For at kunne anvende lyssensoren med en arduino skal noget software skrives. Sensoren sidder i en spændingsdeling og outputtet fra lyssensoren tilkobles en pin på arduinoen. Herfra foretages en analog måling med ADC'en (analog til digital konverter) på arduinoen. Dette gøres ved at anvende analogRead() i softwaren.

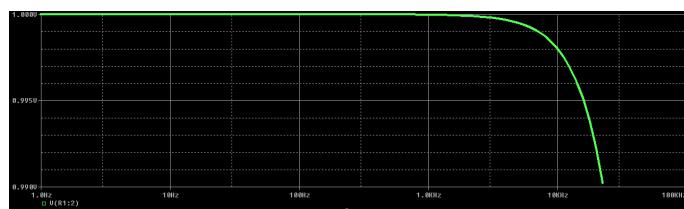
I sensorens spændingsdeling sidder en transistor som genererer høj frekvens støj. For at filtrere støjen væk anvendes et low-pass filter.

3.3 Low-Pass filter

Et low-pass filter er et filter som sorterer høje frekvenser fra men samtidig tillader DC (jævnstrøm) igennem.



Et low-pass filter kan konstrueres med en modstand og en kondensator i serie. Kondensatoren er forbundet til stel så den skaber en AC-kortslutning (vekselsstrøm) og derved filtrerer AC væk fra signalet. Lave frekvenser filtreres ikke væk, fordi kondensatoren bruger tid til at lade op og derved ikke længere fungerer som stel fra de frekvenser.



Figur 3.1: Eksempel på low-pass filter med 1k modstand og 1nF kondensator.

Som det ses på low-pass filteret ovenover¹ hvor frekvenser omkring 1k begynder at blive skåret fra.

¹FiXme Note: find ud af hvordan vi skal referer til figur

3.4 Motor

3.5 Test

Software 4

4.1 Software introduktion

Den indlende idé til en softwareløsning som kan følge en linje er udarbejdet, og ses på figur ??.

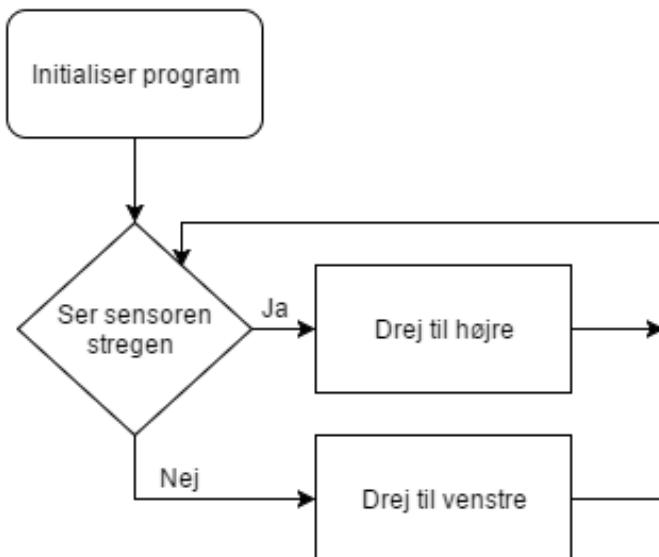
I dette afsnit ses der nærmere på den software som er skrevet i takt med udviklingen af projektet. Her introduceres om andre den indledende løsningsidé samt udviklingen deraf. Herfra ses der nærmere på virdereudviklingen ad denne, fra implementering af én til flere sensorer samt betydningen deraf for produktet.

Endvidere præsenteres test i henhold til virkningen af de forskellige stykker software som er konstrueret hertil.

Indledende problemløsning

For at gøre robotten i stand til at kunne følge den opstillede bane er en indledende problemløsning opsat. Her har gruppen valgt sætte fokus på implementering af én sensor som kan videreudvikles fremadrettet.

Softwareen skrevet har to funktioner baseret ud fra én registreret måling. På nedenstående figur ses hvorledes softwarens handling forløber.



Figur 4.1: Den inledende software løsning til linetracking med 1 sensor.

Som det ses på ovenstående figur har softwaren to funktioner. Ses stregen på den opstillede bane, hvis ja drej til højre, hvis nej drej til venstre. Dette resulterer i at robotten er i stand til at kunne følge kanten af stregen ved at registrere de forskellige overfladefarvemålinger.

4.2 Arduino

Den indledende løsning blev først implementeret på en Arduino Uno mikrocontroller hvorefter konceptet blev testet og verificeret. Koden er derfor i sin indledende fase også skrevet i Arduino.

```
void FollowLine()
{
    while (analogRead(sensor_one_pin) > mean_value) //sensor -> white paper
    {
        analogWrite(left_motor_pin, SLOW);      //turn right
        analogWrite(right_motor_pin, FAST);     //

        Serial.print("Turn right. Sensor data =");
        Serial.println(analogRead(sensor_one_pin));

    }
    while (analogRead(sensor_one_pin) < mean_value) //sensor -> black line
    {
        analogWrite(left_motor_pin, FAST);      //turn left
        analogWrite(right_motor_pin, SLOW);     //

        Serial.print("Turn left. Sensor data =");
        Serial.println(analogRead(sensor_one_pin));

    }
}
```

Figur 4.2: FollowLine() funktion.

På overstående figur ses funktionen follow_Line(). Koden fungerer sålades at når sensoren registrerer den hvide omkringliggende farve drejes der til højre og når den sorte linje registreres drejes der til venstre. Robotten kører derved kun på kanten af linjen.

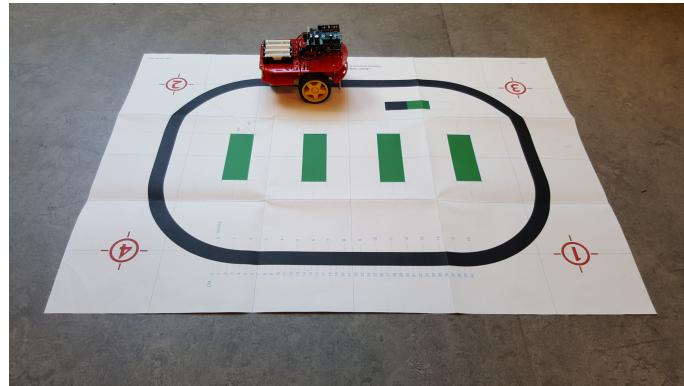
Test og delkonklusion

Ud fra det introducerede linetrack system blev konceptet testet ved brug af én sensor på den opstillede bane.

Ud fra denne test kan det konkluderes at softwaren virker i henhold til den indledende problemløsning. Robotten er i stand til at følge kanten af linjen ved hjælp af de forskellige overfladefarveværdier som sensoren registrerede gennem testen.

Implementering af flere sensore

For at robotten er i stand til at kunne følge linjen hurtigst og bedst muligt implementeres yderligere sensore. Dette gøres for at robotten kan navigere nøjagtigt og præcist. Sensorene er placeret ud fra en skabelon som gruppen har 3D printet for



Figur 4.3: Den inledende software løsning til linetracking med 1 sensor i et superloop.

at skabe de bedste omstændigheder muligt for at få så præcise måliner som muligt. Her anvendes konceptet vist i figur ?? i Line Track.¹. Herfra ændres softwaren fra den indledende problemløsning nu til at håndtere tre sensore i stedet for én.

Som det ses i overstående figur ses de fire mulige scenarier som er præsenteret i Line Track.² Som vist i flowchartet registrerer sensorene om stregen er visuel for sensoren eller ej. Såfremt sensoren registrerer en værdi som fortæller om den er på stregen, kører robotten ligeud indtil en anden værdi måles.

Hvis en anden sensor end den midterste afgiver en måling af den opstillede bane angiver dette at banens forløb har ændret sig, og robotten anvender derved de to funktioner anvist i flowchartet hvorvidt der skal justeres til venstre eller højre.

Såfremt ingen fejlmålinger fra sensoren eller svans stopper robotten fortsætter denne i et superloop.

Videreudvikling af Arduino kode

³

4.3 PID regulering

For at opnå en mindskning i fejl målingerne fra sensoren, implementeres en PID controller. En PID controller står for proportional-integral-derivative controller. Formålet med en PID er at man konstant mäter en fejl værdi som består af differensen mellem en ønsket værdi og en målt værdi. Controlleren skal forsøge at minimere fejl målingerne mest muligt ved at justerer fejl målingen over tid.

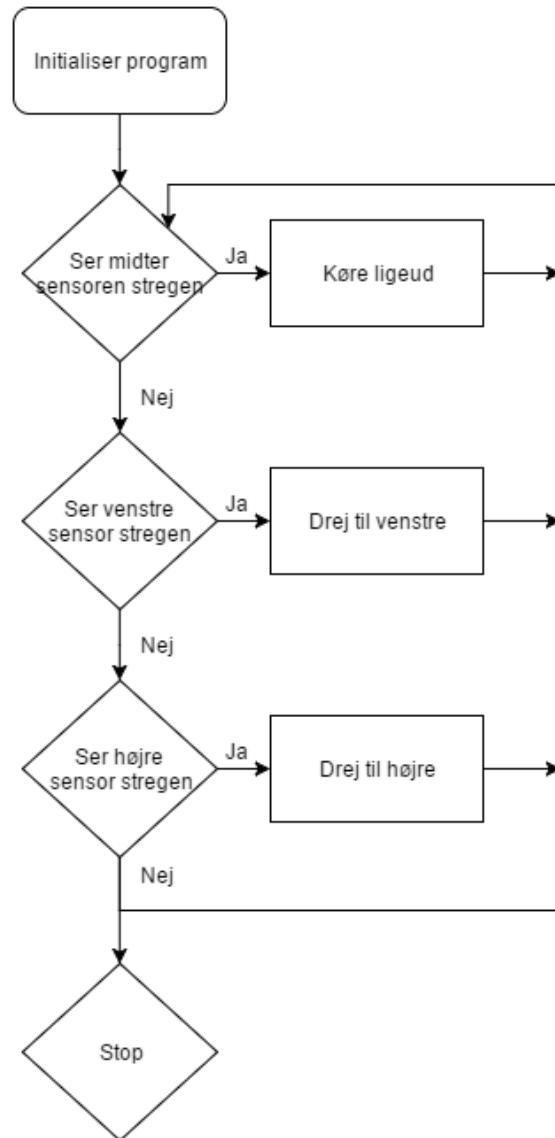
PID controlleren kan opskrives på følgende måde.

$$u(t) = K_p e(t) + K_i \int_0^t e(T) dT + K_d \frac{de(t)}{dt} \quad (4.1)$$

¹FiXme Note: lav reference til line track

²FiXme Note: reference til line track

³FiXme Note: indsæt her billede af koden med tre sensorer



Figur 4.4: Software flowchart over problemløsningen ved hjælp af tre sensore i et superloop.

hvor,

p , er proportional leddet som beskriver den nuværende værdi, hvilket er den ønskede værdi trukket fra den målte værdi.

i , er integrale leddet som beskriver den forhændsværende værdi, hvilket er
 $d =$

4.4 Test

```
void FollowLine()
{
    sensor_middle_read = analogRead(sensor_middle_pin);
    sensor_left_read = analogRead(sensor_left_pin);
    sensor_right_read = analogRead(sensor_right_pin);

    while (sensor_middle_read > mean_value &&
           sensor_left_read < mean_value &&
           sensor_right_read < mean_value)      //middle sensor sees the line
    {
        digitalWrite(left_motor_pin, HIGH);   //drive straight
        digitalWrite(right_motor_pin, HIGH);

        Serial.print("Drive straight. middle = "); //Prints the measured value to the consol
        Serial.print(sensor_middle_read);
        Serial.print(" left = ");
        Serial.print(sensor_left_read);
        Serial.print(" right = ");
        Serial.println(sensor_right_read);
    }
}
```

Figur 4.5: Software flowchart over problemløsningen ved hjælp af tre sensore i et superloop.

Test 5

Konklusion 6

I dette projekt havde gruppen til formål at designe og implementere den nødvendige hard- og software med henblik på at konstruere en automatisk linjefølgende robot.

Produktet er udviklet ud fra Magician chassis som blev leveret af Sparkfun med en dertil forud bestemt lyssensor. Produktet blev udviklet over flere stadier hvoraf den indledene fase var at implementere en sensor til en Arduino og få robotten til at følge linjen ud fra den feedback som kom fra sensoren.

Herfra implementerede gruppen flere sensore til produktet i form af videreudvikling af projektet. Dette resulterede i..... blah blah blah....

Perspektivering 7

Virker produktet eller viser projektet kun et kocpet bag udviklingen deraf?
Hvis produktet ikke virker, hvad kunne så være ideelt at se på både i form af konstruktion men også videreudviklingen af denne?
Hvad har det gjort for projektet at implementere tre sensorer frem for én?
Hvorfor har vi valgt at bruge en H-bro? -> (videreudvikling til baglæns kørsel)

Rettelser