Pole Position
AUR2: Gruppe 2

Uddannelse og semester:

Robotteknologi - 2. semester

Dato for aflevering:

27. Maj 2015

Vejleder:

Preben Hagh Strunge Holm

Gruppe medlemmer:

Joakim Grøn, Anders Fredensborg Rasmussen,
Daniel Holst Hviid, Jonas Alexander Lundberg Andersen,
Kristian Hansen & William Bergmann Børresen

Teknisk Fakultet
Syddansk Universitet

Indholdsfortegnelse

1 Indledning

1.1 Kravspecifikation

1.2 Problemformulering

1.3 Projektafgrænsning

1.4 Tidsplan

2 Hardware

2.1 Accellerometer

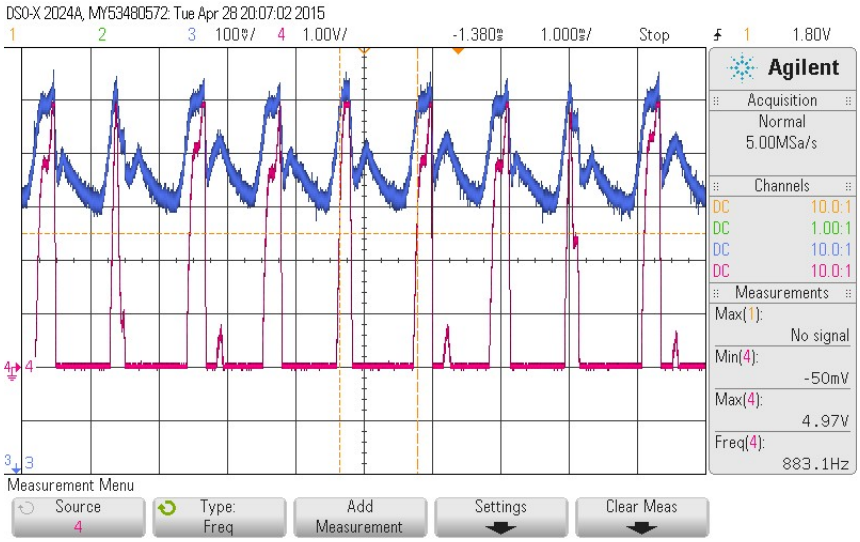
2.2 Lap sensor

Til at holde øje med når bilen krydsede målstregen, faldt valget på en CNY70 optisk sensor. Den består af en infrarød lysdiode og en phototransistor bygget ind i samme hus. Dioden sender hele tiden infrarødt lys ud gennem et lille vindue, og hvis der er en flade tæt på til at reflektere lyset, vil det blive opfanget af phototransistoren, der sidder ved siden af dioden bag et lysfilter. Mængden af lys der bliver reflekteret tilbage afhænger af afstanden til fladen og typen af materiale der er på overfladen - en hvidt malet streg vil for eksempel reflektere mere lys tilbage end en sort materet overflade.

For at gøre signalet fra sensoren brugbart for microcontrolleren, ville vi bruge en komparator til at digitalisere outputtet. Oprindeligt var planen at bruge en LM311 comparator, men det viste sig at den interne komparator i microcontrolleren lige så nemt kunne bruges til det formål, og så ville der også skæres ned på antallet af eksterne komponenter og derved mindske pladsforbruget på vores print. En anden fordel ved at bruge den interne komparator, er at det bliver muligt at bruge den interne spændingsreference på 2.56 V som input til komparatoren. Så er det bare et spørgsmål om at dimensionere outputtet fra CNY70 sensoren med en modstand, så spændingen der kommer fra den sorte overflade på banen ligger under 2.56V og spændingen fra den hvide målstreg ligger over. Microcontrolleren aflæser komparatoren ved at se på outputtet fra den, og det kan sættes op så den sender et interrupt, når der kommer en rising eller falling edge eller når outputtet skifter tilstand. Med denne løsning bruges der kun tre eksterne komponenter - en modstand hver til henholdsvis lysdioden og phototransistoren samt CNY70 sensoren. Modstanden til lysdioden blev sat til 150 ohm, så der løber godt 30 mA i det kredsløb. For at opfylde spændingskravene til komparatoren blev der brugt en 22 kilo-ohms modstand - det gav en spænding på banen omkring 0.8 V og en spænding på målstregen på ca 4.5 V.

2.3 Tachometer

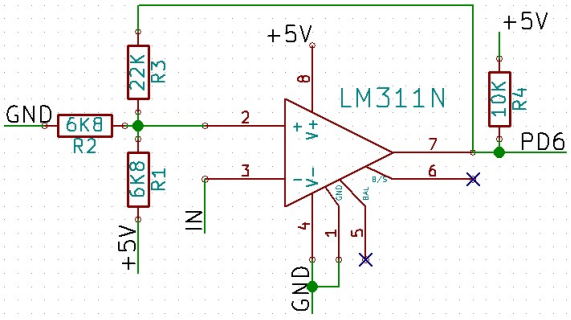
Til at måle omdrejningshastigheden på bilen, samt at opfylde kravet om en elektrofysisk sensor/aktuator, blev der valgt en analog hall sensor som monteredes på motoren. Sensoren har tre ben - et til 5 V, et til stel og et output. Outputtet ligger og svinger omkring 2.5 V og ændrer sig i positiv eller negativ retning alt afhængigt af, hvordan den magnetiske flux ændrer sig i nærheden af sensoren. Når motoren kører, vil man på hall sensorens output så kunne måle, når polerne passerer tæt forbi sensoren. Det giver et svagt og noget støjet signal, der skal behandles for at microcontrolleren kan bruge det til noget. Det rå signal havde en peak-to-peak spænding på 100-200 mV, så det var ret vigtigt først at få det forstærket op. Det havde også den tilføjede bonus at skære noget af den mere højfrekvente støj fra, fordi den forstærker vi valgte, en instrumenteringsforstærker AD623, havde en knækfrekvens på godt og vel 90 kHz under de forhold vi arbejdede under.



Figur 1: Her ses signalet fra hall sensoren før (blå) og efter (lyserød) det er blevet behandlet af forstærkeren.

Ud fra de første målinger, viste det sig at den lave del af signalet var meget støjfyldt og svær at behandle, så vi valgte at skære den del af signalet fra med et offset på forstærkeren se figur 1. Offsettet blev styret af et potentiometer, der blev brugt som en slags justerbar referencespænding.

For at digitalisere signalet, så microcontrolleren kunne behandle det, blev der brugt en komparator - denne gang som en schmitt trigger for at formindske fejl. Det var ikke



Figur 2: Diagram over schmitt trigger kredsløbet.

nødvendigt at designe spændingstærsklerne på schmitt triggeren så præcist. De skulle bare ligge relativt langt fra hinanden, så eventuel støj ikke kunne få den til at skifte tilstand utilsigtet. Ud fra målinger med oscilloskop viste det sig, at et spænd på over 0.5 V var rigeligt til at forhindre støj i at få schmitt triggeren til at skifte tilstand.

$$V_{TL} = \frac{R_2||R_3}{R_2||R_3 + R_1} * 5V \tag{1}$$

$$V_{TH} = \frac{R_2}{R_2 + R_1||(R_3 + R_4)} * 5V \tag{2}$$

Til at beregne spændingstærsklerne skal man finde spændingen på det ikke-inverterende ben, når komparatoren er i mætning og når den er åben. Så ender man med to ligninger med to ubekendte, hvis man fastsætter to af modstandene og spændingstærsklerne på forhånd. Modstandsværdierne blev dikteret af udvalget på komponentlageret, og ved hjælp af ligning (1) og (2) designede vi spændingstærskler på henholdsvis 2.2 V og 2.8 V.

2.4 Bremse

For at bremse bilen kortsluttes motorterminalerne. En kortslutning bremser motoren ved at lade et magnetfelt der modsætter sig motorens bevægelse blive induceret i spolerne. Hvis kortslutningen ikke er til stede kan der ikke løbe nogen strøm og derfor heller ikke dannes noget modsatrettet magnetfelt. Selve kortslutningen bliver udført af 5 volts relæ der bliver styret af en MOSFET forbundet til et IO-ben på microcontrolleren. Se figur..

2.5 Elektromagnet

3 Software

3.1 Generel struktur

3.2 Protokol

3.3 Eksempler(sensor)

3.3.1 Tachometer

Til at behandle signalet fra hall sensoren, benyttes input capture funktionen til Timer 1. Input capture funktionen har sit eget interrupt og fungerer ved automatisk at læse værdien af Timer 1's timer/counter registre (TCNT1A/TCNT1B) over i to input capture registre (ICR1L/ICR1H) når der kommer et interrupt. Input capture er den sekundære funktion til PD6, og sættes op i Timer 1's kontrol registre (TCCR1A/TCCR1B). Vi valgte at sætte den op med en prescaler på 1/8 og en falling edge interrupt trigger. Det giver en opdateringsfrekvens på 0.5 μs og en maksimal tid på 32.5 ms inden timer/counter registrene overflower. Det er fint nok til at kunne rumme tiden imellem flere interrupts. I interruptrutinen beregnes tiden imellem 4 interrupts for at finde ud af hvor lang tid motoren er om at dreje en omgang (Der sidder tre poler på motoren, så 4 interrupts vil svare til en hel omgang for motoren). Denne tid kan så bruges som et udtryk for hastigheden, da de to ting er omvendt proportionale. Bilens reelle hastighed kan også beregnes, men på grund af proportionaliteten er det ligegyldigt, fra microcontrolleren's synspunkt, om man bruger det ene eller det andet - man skal bare huske at en høj hastighed vil give en lav tid og omvendt, når man skriver programmet, der skal bearbejde hastigheden. Jo færre cycles man bruger på at få noget brugbart ud i den anden ende jo bedre, i næsten alle tilfælde, når der også skal laves andre ting sideløbende med input capture interruptet.
For at beregne tiden skal der holdes styr på pulserne - der skal bruges en tid fra den første puls og en tid fra den sidste puls. Det letteste er at tælle en variabel op, hver gang der modtages et interrupt således at, når variablen er 1 så læser man den første tid, og når variablen er 4 så læses den sidste tid. Derefter trækkes de to tider fra hinanden for at finde forskellen. Alle værdier og variable gemmes i SRAM-hukommelsen på tildelte addresser indtil de skal bruges igen.
Det viste sig at hall sensoren gav mange udslag når bilen holdt stille og en af polerne var

lige på grænsen til at passere sensoren. Vi antager at det er fordi feltet er meget ustabilt i grænseområdet.

3.3.2 Lap sensor

For at initialisere microcontrolleren's komparator skal den sættes op i dens eget status register (ACSR) og i Special Function IO Registret (SFIOR). Man kan vælge flere forskellige inputs til komparatoren - PB2(AIN1) og PB3(AIN0) er henholdsvis ikke-inverterende og inverterende inputs som standard. Alternativt kan hele PORTA bruges til det inverterende og den interne bandgap reference på 2.56 V til det ikke-inverterende. ACME bit'en i SFIOR enabler ADC multiplexeren, så den styrer hvilket ben på PORTA der bliver brugt. ACBG bit'en enabler bandgap referencen.

Table 21-1. Analog Comparator Multiplexed Input

ACME	ADEN	MUX2:0	Analog Comparator Negative Input
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

Figur 3: I tabellen kan man se at ACME bit'en lader multiplexeren vælge hvilket ben på PORTA, der er input til komparatorens inverterende ben. Medmindre ADC'en er slået til - så bliver inputtet taget fra PB2(AIN1)

Man har også mulighed for at vælge, hvornår komparatoren skal sende et interrupt. Interruptet bliver sendt som en reaktion på hvad der sker på komparatorens output-ben. Vi har sat den op til at sende et interrupt, når outputtet toggler tilstand, men der er også mulighed for at køre på enten falling eller rising edge. Det betyder i vores tilfælde ikke noget hvilken metode man vælger på grund af måden koden bearbejder interruptet, men mere om det senere.
I interruptrutinen til lap sensoren beregnes omgangstiden. Hver gang målstregen krydses læses tiden fra tre SRAM addresser, hvor tiden microcontrolleren har været tændt millisekunder er gemt i 24 bits. Interruptrutinen sørger for at gemme et 24 bits time stamp i SRAM'en hver gang den køres. Med den totale tid og et time stamp fra det forrige interrupt, er det bare et spørgsmål om at trække de to fra hinanden for at finde omgangstiden. For at forhindre ugyldige omgangstider, hvis der kommer mere end et

interrupt på samme målstreg, kontrolleres det om omgangstiden er længere end 512 millisekunder se figur 4.

```
sub     R0, R3
sbc     R1, R4                  ; Difference between current time and last time stamp
sbc     R2, R5

cpi R1, 2
brlo Lap_Time_End
```

Figur 4: Her er den nuværende tid gemt i R0, R1, R2 og time stampet i R3, R4, R5. Ved at kontrollere om den mellemste byte i R1 er under 2 kan det afgøres, om der er gået nok tid siden sidste time stamp til at anse interruptet som gyldigt.

Det er også efter denne kontrol at man kan ændre på alle de parametre der kun skal ændres en gang når målstregen krydses. Inden man forlader interruptrutinen er det vigtigt at resette comparator interrupt flaget i ACSR. Hvis flaget er sat, når det globale interrupt bliver enablet efter rutinen, anser microcontrolleren det som om den har modtaget et nyt interrupt, og så vil den køre hele rutinen igen.

3.4 Hastighedskontrol

3.5 AI

3.6 GUI

4 Test/Resultater

5 Diskussion

5.1 Fejlkilder

6 Konklusion

7 Litteraturliste

Figure ??

- `www.playground.arduino.cc/uploads/Main/mpu-6050.jpg`
- Dato: 25/05/2015

Figure 3

- `URL:SKAL_FINDES_KRSITIAN!!!!!!!`
- Dato: xx/xx/2015

Electrical Engineering

- Principles and Applications
- 6th Edition
- Allan R. Hambley
- ISBN-13 978-0-273-79325-0

Atmega32A-PU Datasheet

- www.atmel.com/images/doc2503.pdf
- Dato: xx/xx/2015

AVR 8bit Instruction set

- www.atmel.com/images/doc0856.pdf
- Dato: xx/xx/2015

8 Bilag