

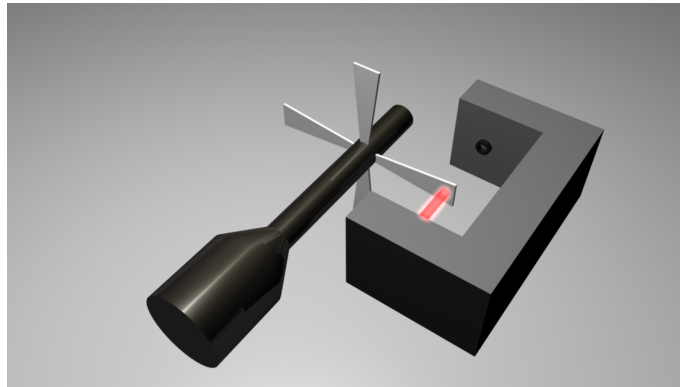
0.1 Måling af farten

I dette afsnit beskrives der hvordan en sensor udnyttes til at beregne farten bilen kører med. Der vil både blive beskrevet hardwaren til sensoren samt hvordan den softwaremæssigt bruger sensoren. Farten bliver brugt til at mappe banen samt kører med konstant hastighed. Disse funktioner vil blive beskrevet i afsnit .

indsæt
afsnit
for kon-
stant
hastig-
hed og
mapning

0.1.1 Fart Hardware - TCST1230

Vi benytter TCST1230, som er en optisk sensor, til at lave en fotogate. Sensoren består af en fotocelle og en vifte med 16 blade på. Hjulet med vifterne sidder på motorens akse og hver gang aksen drejer en omgang vil den bryde fotosensorens lys XX gange. Hver gang lyset brydes så fås en puls fra fotosensoren som sendes videre til micro-controlleren. Dette er illustreret i figur 1



Figur 1: 3D model af sensoren

Efter test er der sat flere blade på viften da for at præcist at kunne bruge målingerne fra sensoren krævede det en højere opløsning. Dette fås ved at tilføje flere blade til viften så vi får flere pulser pr. omdrejning af aksen. Der er således nu så 4 blade på viften hvilket giver os 4 pulser pr. akse omdrejning. Dette svarer til 16,8 pulser pr. hjul omdrejning.

Se afsnit ?? for udregningen af gearingen.

Dette giver en opløsning der er 4 gange bedre, end kun at have 1 vifte.

For bedre opløsning kunne der sættes flere vifter på, men kommer der for mange vifter på, vil det dog blive svært for sensoren at aflæse vifterne da deres mellemrum til sidst vil blive meget småt. Jo flere vifter der kommes på, jo sværere bliver det også at bygge ringen med vifterne.

Der bliver i dette tilfælde valgt at have 4 vifter da det kunne bygges forholdsvis let og det giver en opløsning der er høj nok til hvad vi ønskede. Ved blot 1 vifte kørte vi 2cm per omdrejning. Da vi ved hjulet kører 0,2381 omgange pr. motor omdrejning så kan længden den kører mellem hver puls udregnes ved at multiplicerer hjul omdrejninger med hjulets

omkreds:

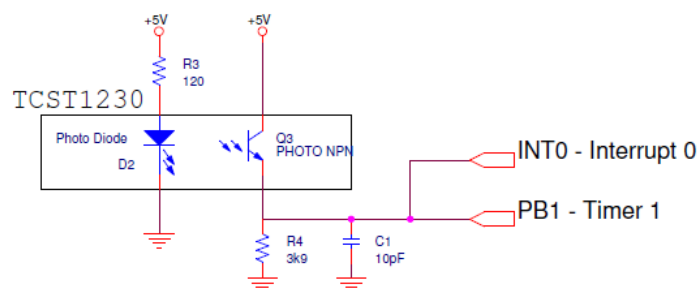
$$0,2381 * 8,5 = 2,02cm$$

Dette er ikke ret præcist da vi så kun kan aflæse længden/farten i intervaller af 2cm. Det blev så sat 4 vifter på og dette giver en præcision på:

$$(0,2381/4) * 8,5 = 0,51cm$$

Dette er markant bedre da den nu kun kører 0,51cm før den får en puls og udregner længde og hastighed. Så der fås flere mindre intervaller der kan tjekkes og reageres på.

Som det kan ses på tegningen af kredsløbet i figur 2 så er der sat nogle modstande og en kondensator på sensorens kredsløb.



Figur 2: Kredsløbet til TCST1230 Sensoren

Modstanden R3 i figur 2 er der for at sørge for at strømmen over dioden D2 ikke bliver højere end den kan tåle. Dioden kan nemlig kun klarer 30mA ifølge databladet.

Modstanden R4 virker som en pulldown modstand der sørger for at hive signalet langt nok ned, når fototransistoren ikke modtager lys, så der er logisk lavt signal.

Kondensatoren C1 benyttes til at sortere små signaler fra. Hvis signalet "hopper" frem og tilbage ved en overgang fra logisk lavt til logisk højt vil kondensatoren tage de små hurtige skift og sortere dem fra. På denne måde fjernes en del af støjen. Der blev gennem oscilloskop observeret pral på sensoren ved TTL signal behandling. Der var benyttet en forkert metode til at opbygge sensoren på. Der var anvendt en struktur som forårsagede at diode spændingen over fototransistoren ikke kunne komme under 0.6V. Der blev derfor udarbejdet en anden struktur som set i databadet s. 3. Modstanden benyttet til dioden blev udregnet vha. Ohms lov og ved at sætte strømmen lig 30mA. Dette resulterede i en modstand for R3 på 120 Ohm.

indsæt
data-
blad

TCST1230
s. 3

R4, som er pulldown modstanden, blev beregnet ved at kigge på figur 8 i databladet . Her blev modstanden udregnet til ca. 4K Ohm. Der anvendes her en 3.9K Ohm modstand

indsæt
data-
blad

0.1.2 Fart Software

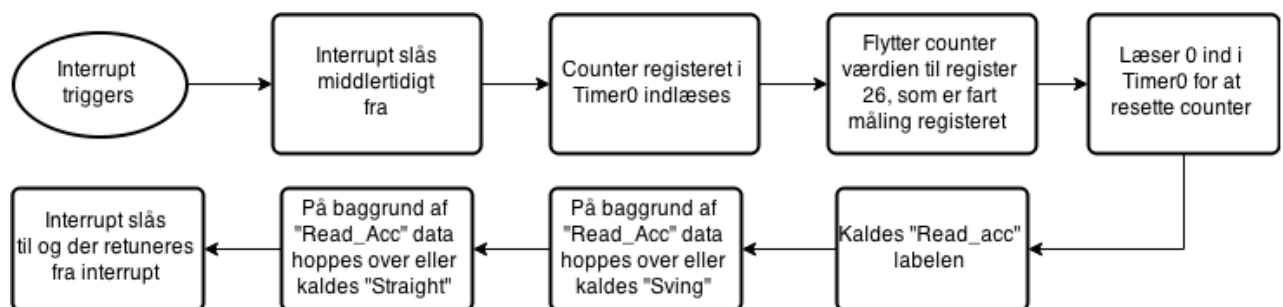
Sensorens output er sat på "PB1" og "PD3" som går til "Timer1" og "INT0" i micro-controlleren. Så hver gang den får en puls så bliver timeren og interruptet triggeret. Timeren bliver benyttet til at finde længden den har kørt. Dette er beskrevet i afsnit ??.

Interruptet benyttes sammen med Timer0 til at finde farten. Timer 0 er sat op så den inkremerer hvert 0.000064 sekund. Hvilket svarer til den inkremerer hvert 64. mikrosekund. Dette er sat ved at bruge en prescale på 1024. Da clock frekvensen på micro-processoren er 16MHz udregnes det således:

$$1/(16000000/1024) = 0,000064\text{sek} = 64\text{mikrosekunder}$$

Dette bruges for at finde tiden mellem pulses som kan benyttes som udtryk for farten. Timeren er ligeledes sat op så hvis der er overflow i Timer0 så kaldes et andet interrupt der blot nulstiller timeren og den fortsætter så.

Flowdiagrammet i figur 3 viser processen hver gang Interruptet bliver kaldt. Hvilket sker hver gang sensoren sender en puls.



Figur 3: Flowchart af softwaren til måling af farten

Der slås først interrupts fra globalt, for at dette interrupt ikke bliver afbrudt af et andet interrupt. Herefter indlæses den værdi tælleren er nået til. Denne værdi gemmes i registeret R26, som er vores "målt fart" register. Så ligges der 0 ind i Timer0, for at nulstille den. På den måde får vi hvor mange gange der er gået 64 mikrosekunder på 1 puls. F.eks. hvis der går 10 optællinger mellem timer puls, så har bilen farten: 7,9 m/s.

Dette kan ses i lookup table i afsnit

Inden interruptet sluttet kaldes nogle labels der benyttes til at bestemme hvor på banen

henvis
til look-
up table
og sørg
for tab-
le har
korrekte

den er. Disse er beskrevet i afsnit XX.

Til sidst så slås interrupts til igen og der returneres til der koden var nået til, indtil næste interrupt bliver kaldt.



indsæt
afsnit
label