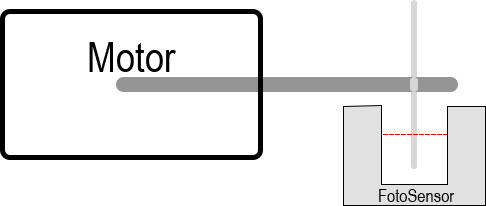
# Måling af Fart

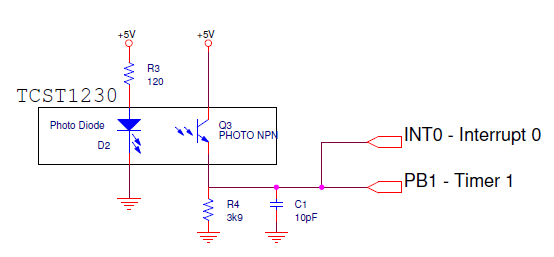
I dette afsnit beskrives der hvordan en sensor udnyttes til at beregne farten bilen kører med. Der vil både blive beskrevet hardwaren til sensoren samt hvordan den softwaremæssigt bruger sensoren. Farten bliver brugt til at mappe banen samt kører med konstant hastighed. Disse funktioner vil blive beskrevet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

## Sensor 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 XX \todo{indsæt figur, evt 3d figur i stedet}



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 pulses pr. omdrejning af aksen. Der er således nu så 4 blade på viften hvilket giver os 4 pulse pr. akse omdrejning. Dette svarer til XX pulses pr. hjul omdrejning. Dette giver en præcision på XXX \todo{indsæt værdier i XX}

Som det kan ses på tegningen af kredsløbet \todo{indsæt figur kredsløb} så er der sat nogle modstande og en kondensator på sensorens kredsløb.

Modstanden R3 i figur XX \todo{indsæt figur} 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. \todo{indsæt datablad}

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 sorterer 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 sorterer 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. \todo{TCST1230 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.

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

## Sensor Software

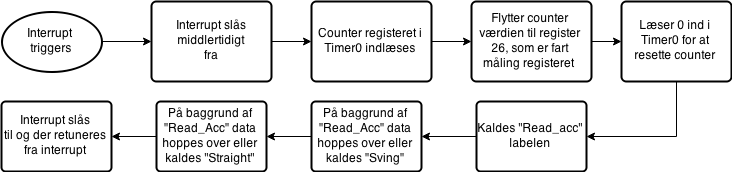
Sensorens output er sat på ”PB1” og ”PD3” som går til ”Timer1” og ”INT0” i micro-controlleren. Så hver gang den får en puls så bliver timeren og interruptet triggeret. Timeren bliver benyttet til at finde længden den har kørt. Dette er beskrevet i afsnit XX \todo{korrekt afsnit}.

Interruptet benyttes sammen med Timer0 til at finde farten. Timer 0 er sat op så den inkrementer hvert 0.000064 sekund. Hvilket svarer til den inkrementerer 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/(16 000 000/1024) = 0.000064 Sekund.

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 XX \todo{indsæt figur} viser processen hver gang Interruptet bliver kaldt. Hvilket sker hver gang sensoren sender en puls.



Der slåes 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 er gået 128 mikrosekunder på 1 puls så har bilen farten:

128 mikrosekunder / puls = XX / XX \todo{indsæt korrekt udregning til en fart der kan bruges}

Da der går XX pulses per hjul omdrejning.

Inden interruptet sluttes kaldes nogle labels der benyttes til at bestemme hvor på banen den er. Disse er beskrevet i afsnit XX. \todo{indsæt afsnit label}

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

# Måling af afstand kørt

I dette afsnit beskrives hvordan bilen måler hvor langt den er kørt. Her benyttes den samme sensor som bruges til udregning af farten der køres med.

Afstanden bilen har kørt bruges til at mappe banen som der kan læses nærmere om i afsnit XX \todo{indsæt afnist}

## Hardware – TCST1230

Da sensoren er den samme som benyttes til måling af fart vil dens hardware ikke blive beskrevet her. Der henvises i stedet til afsnit XX \todo{indsæt afsnit hardware}

Den eneste ændring på hardwaren er at sensores udgangssignal til micro-controlleren går på indgang ”PB1.”

## Software – TCST1230

Hver gang sensoren sender en puls til ”PB1” på micro-controlleren så vil Timer1 inkrementer med en. Timer1 er valgt da det er en 16bit timer og derfor kan der køres XX meter før timeren giver overflow. Dette er rigeligt til formålet, da efter hver lige strækning vil tælleren nulstilles. \todo{indsæt tal}

Da der vides hvor mange pulses der går på en hjul omdrejning, nemlig 4.2, så kan der ved at tælle pulses udregnes hvor langt den har kørt. \todo{indsæt tal og henvis til journal}

Timer1 er opsat med en prescale der tager udgangspunkt i et ekstern input. Dette betyder at det eksterne signal benyttes som clock-frekvens med rising edge. Rising edge betyder at en periode tager udgangspunkt i signalet på vej op i stedet for på vej ned.