

Robotkonstruktion

Rasmus Friis Kjeldsen
razzle@diku.dk

17. oktober 2003

Indhold

1	Indledning	1
1.1	Motivation og målgruppe	1
1.2	Indhold	1
1.3	Praktiske bemærkninger	1
2	Microchip PIC indlejrede processorer	2
2.1	Arkitektur	2
2.2	I/O og styring af ydre enheder	3
2.3	Overførsel af program til PIC'en	4
2.3.1	Minimalt kredsløb	4
2.3.2	Programmeringskredsløb	5
2.3.3	PC-software og overførsel af programmer	7
2.4	Eksempelprogram	7
2.5	Bootloader	8
2.6	Programmeringsteknikker	8
2.6.1	Fastkommaaritmetik	8
2.6.2	Tabelopslag og lineær interpolation	9
2.7	Andre indlejrede processorer	9
3	Strømforsyning	9
4	Digitalelektronik	12
4.1	TTL logik og integrerede kredse	12
4.2	Symboler	12
4.3	Pull up/pull down modstand	12
4.4	Schmitt-trigger	12
4.5	Praktisk implementation	13
5	Sensorer	13
5.1	Digi-tast	14
5.2	Måling af spænding	14
5.3	Måling af batterispænding	15
5.3.1	Tilslutning til processor	15
5.3.2	Udregning af batterispænding	16
5.4	Lysfølsom modstand	16
5.4.1	Tilslutning til processor	16
5.4.2	Eksempel på anvendelse	17
5.4.3	Andre lyssensorer	21
5.5	Tachometer	21
5.6	Dreje- og skydepotentiometer	22
5.7	Sharp IR-afstandssensor	23
5.7.1	Virkemåde	23
5.7.2	Kalibrering	25
5.7.3	Tilslutning til processor	27
5.7.4	Kommunikation med processor	29
5.7.5	Montering på robot	30
5.7.6	Miljøets indflydelse på sensoren	35
5.7.7	Eksempler på anvendelse	38

5.7.8 Andre afstandssensorer	39
6 Aktuatorer	40
6.1 Lysdiode	40
6.2 7-segment LED-display	40
6.3 DC-motor	41
6.3.1 Tilslutning	41
6.3.2 Pulsbreddemodulering	42
6.4 Servomotor	43
7 Kommunikation	43
7.1 I ² C	43
7.2 RS-232	44
8 Regulering	44
8.1 Hastighedsregulering	45
9 Fordele og ulemper ved selvbyggede kredsløb	46
9.1 RCX	46
9.2 Handy Board	46
9.3 SV-203	46
10 Konklusion	47
A Kildekode	48
B Indkøbssteder	49
B.1 Acroname Inc.	49
B.2 Brinck Elektronik	49
B.3 Koz Electronics	49
B.4 LEGO	49
B.5 Microchip	50
B.6 RS-components	50
Litteratur	51

1 Indledning

1.1 Motivation og målgruppe

Gennem arbejde med robotter, både ved kurser på DIKU, privat interesse og senest ved deltagelse i DTU-konkurrencen RoboCup (2002 og 2003) [Dan(2003)] har jeg opbygget en større erfaring med konstruktion af robotter. Herved menes konstruktion af *controller* (styrende elektronikkredsløb inkl. cpu) til styring af robotter, konstruktion af *platform* (robotens fysiske udformning), samt udvikling af programmel til styring af controllere - Det sidste både på lavniveau (driverniveau) og på højniveau (overordnet problemløsende logik).

Denne erfaring ønsker jeg her at videregive i form af en rapport, som giver en pædagogisk fremstilling af problematikken ved robotbygning. Rapporten kan anvendes som introduktion til robotbygning, en slags *how-to*. Målgruppen tænkes at være datalogistuderende, som har et minimum af elektronikerfaring, såsom håndtering af loddekolbe og kendskab til udseende og benplacering af basale elektronikkomponenter som f.eks. modstande, kondensatorer, dioder og transistorer. Anvendes denne rapport af en gruppe, vil det være tilstrækkeligt at bare en enkelt i gruppen har den nødvendige elektronikerfaring. Elektronikfagsproget er holdt på et niveau, så en gennemsnits-datalogistuderende kan følge med.

Jeg skal gøre opmærksom på, at min egen faglige elektronikbaggrund begrænser sig til elektronik som tilvalg i folkeskolen, samt interesse for specielt digitalelektronik på hobbyplan.

1.2 Indhold

Indholdet af rapporten er koncentreret om elektroniske komponenter og kredsløb, driverprogrammel til dette samt mere overordnet "mellemniveau"-styring, såsom hastighedskontrol. Rapporten vil ikke gennemgå bygning af robotens platform. Et forslag ville dog være at den opbygges i LEGO, eller at en fjernstyret bil ombygges.

Rapporten er skrevet ud fra en praktisk indgangsvinkel: Hvorledes bygges en controller til en robot, hvilke problemer kan man støde ind i undervejs, og hvordan løser man disse problemer.

I rapporten præsenteres en række muligheder for at opbygge en robotcontroller, men ikke en facitliste. Af pladshensyn beskriver rapporten ikke alle mulighederne til bunds, men henviser i stedet til, hvor man kan finde mere information.

Én sensor vil dog blive beskrevet ekstra grundigt, nemlig afstandssensoren. Denne har jeg erfaring med fra flere robotter, og den dokumentation, jeg har kunnet finde om den er ukomplet og ikke videnskabeligt begrundet. Derfor har jeg selv gennemmålt og beskrevet den i detaljer.

1.3 Praktiske bemærkninger

Figurer med grafer og diagrammer er placeret inde i opgaven, i modsætning til i et appendiks, for at lette læserens tilgang til dem.

Når nye termer introduceres i opgaven, er de markeret med *kursiv*, og termer, der refererer til kommandonavne, mappenavne, programregistre etc., er sat i *courier*.

Kildekode, til de programeksempler der refereres til, er kun placeret på den vedlagte cd-rom, da selve koden hovedsagligt er “bitfedteri”, og er kun interessant, hvis læseren selv skal bruge den til et projekt. I et sjældent tilfælde er det lettere bare at klippe og klistre fra kildefilerne på cd-rommen.

I mappen `src/controller/` ligger driverprogrammel til de gennemgåede sensorer og aktuatorer. I filen `src/controller/robocupController.c` ligger kildekoden til det program, som styrede den robot, jeg deltog med i Robocup 2003 [Kjeldsen(2003b)]. I mappen `web/` ligger kopier af hjemmesider og manualer, som refereres i litteraturlisten. I mappen `matlab/` ligger matlabfiler med de måledata som ligger til grund for graferne i rapporten. I mappen `rapport/robotkonstruktion.pdf` ligger PDF-filen, som denne tekst er en udskrift af.

Appendiks A indeholder et eksempelprogram til PIC-processoren beskrevet i kapitel 2.

Appendiks B indeholder en liste over steder, hvor man kan købe de elektronikkomponenter, som er beskrevet i rapporten.

2 Microchip PIC indlejrede processorer

De indlejrede PIC-processorer fra Microchip [Mic(2003)] er velegnede som basis for styreelektronik til mindre robotter. De kan anvendes enten alene, som hovedprocessor, eller som bindeled mellem en større computer (stationær, bærbar eller pda) og robotens styreelektronik. I det følgende vil PIC’ens arkitektur og dens indbyggede kredsløb til diverse former for I/O blive gennemgået.

Den PIC-model, jeg beskriver, er en PIC 16F876, men de andre modeller i 16Fxxx-serien har tilsvarende egenskaber, med lidt variationer af indbygget hardware, antal I/O-ben og størrelse af hukommelse.

2.1 Arkitektur

Jeg vil indlede med at henvise til [Mic(1997)], som er PIC16Fxxx-seriens referencemanual og til [Mic(2001)], som er manualen til PIC16F876 (og resten af PIC16F87x-serien). Herefter vil jeg kun opridse de mest interessante eller specielle egenskaber ved PIC’ens arkitektur.

PIC’en er temmelig speciel, idet den har en Havardarkitektur. Langt størstedelen af almindelige processorer er opbygget efter Von Neumann-arkitekturen, hvor data- og programlager er fælles. En Havardarkitektur har separat datalager og programlager. Datalageret er almindeligt ram, som bliver slettet, når strømmen til processoren afbrydes, mens programlageret er flashram, hvis indhold bevares selv uden strøm. Både data- og programlager er indlejret på chippen sammen med en del ydre enheder, så som *USART* (seriel kommunikationsenhed) og

A/D-konverter.

PIC'en er en RISC-processor med kun 35 maskinkodeinstruktioner. Hvert instruktionsord er netop 14 bit, og hver datalagercelle er 1 byte, 8 bit. Alle instruktioner tager en instruktionscyklus (= 4 klokeykler) at udføre, med undtagelse af hop-instruktioner, som tager to instruktionscykler at udføre. Som en kuriøsitet kan nævnes, at hop-instruktionerne kun kan udføre ubetingede hop. Har man brug for et betinget hop, må en instruktion med betinget udførelse af næste instruktion bruges før hop-instruktionen.

Alt I/O og selv statusflag (såsom nulflag og overløbsflag) er memory mapped. Processoren har kun eet arbejdsregister, kaldet W. For mere om I/O, se afsnit 2.2.

På grund af instruktionsordets korte længde er der ikke et tilstrækkeligt antal bit til at adressere hele datalageret, på trods af at det kun er 512 bytes stort¹. Derfor er datalageret opdelt i 4 banker, så en instruktion, der har brug for at adressere datalageret, kun skal indeholde 8 bit, de 2 sidste bit sættes i den specielle STATUS-lagercelle, som er spejlet til samme adresse i alle 4 banker.

Tilsvarende har hopinstruktioner heller ikke tilstrækkeligt med adressebit, og bruger en mekanisme svarende til den ovennævnte.

PIC'en understøtter også interrupts. Prioriterede interrupts er dog ikke understøttet, selvom mange forskellige kilder til interrupts er det. PIC'en bruger ikke en interruptvektor, men kun en enkelt fælles interrupthandler. Denne må så håndtere alle interruptkilderne.

PIC'ens har en stak, som anvendes ved kald til subrutiner og ved interrupts til at lagre returadressen på. Den er begrænset til kun 8 niveauer, og den kan kun tilgås indirekte ved hop- og returinstruktioner. Til gengæld bruger den ikke af det sparsomme datalager, men har sit eget separate lager.

Til lagring af data når PIC'en er slukket, har den et lille flash datalager på 256 bytes.

2.2 I/O og styring af ydre enheder

Hvert af PIC'ens I/O-ben er memory mapped til en bit i PIC'ens datalager, som er samlet i databytes, kaldet porte, af 8 I/O-bit. PIC16F876 har 3 I/O-porte kaldet PORTA, PORTB og PORTC, og har i alt 22 I/O-ben — PORTA er ikke fuldt udnyttet. I/O-benets retning (input eller output) kan styres i software, med de til hver port svarende dataretningsregistre TRISA, TRISB og TRISC.

Ud over at være forbundet til en generisk I/O-port er mange af PIC'ens ben multiplexet med en eller to andre specielle funktioner. Disse inkluderer bla.:

- Analog indgang, hvor benspændingen ($0 - V_{cc}$ volt) vha. en A/D-konverter konverteres til et 8 eller 10 bit tal, se afsnit 5.2.
- Interrupt ved ændret værdi, hvor et interrupt genereres ved ændret logisk

¹Kun 368 bytes er frit lager, da resten bliver brugt til memory mapped I/O, statusregistre og spejling af lagerceller.

niveau, enten på den stigende eller faldende flanke eller på begge.

- Hardware-tæller, som tæller ændringer af logisk niveau enten på den stigende eller faldende flanke.
- Hardware USART, hvor to I/O-ben bliver brugt til tovejs RS-232-kommunikation, se afsnit 7.2.
- Hardwaregenerering af pulsbreddemoduleringsignal, se afsnit 6.3.

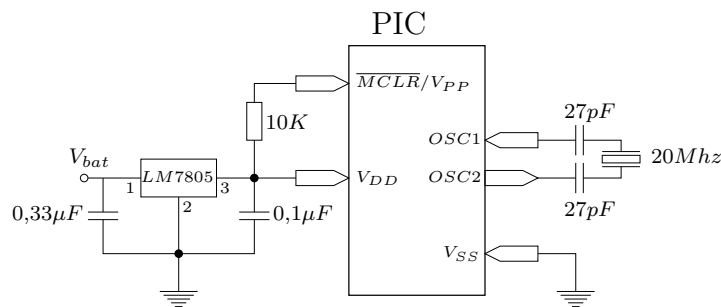
Da hvert ben har flere forskellige funktioner, og da ikke alle ben kan udføre alle funktioner, kræver det grundig planlægning af kredsløbets layout i de tilfælde, hvor man har brug for mange både generelle og specielle funktioner.

2.3 Overførsel af program til PIC'en

I dette afsnit vil jeg først præsentere det minimale kredsløb, der behøves, for at få PIC'en til at fungere. Herefter et kredsløb, som kan bruges til at overføre programmer fra en almindelig PC til PIC'ens flash-programlager — Et sådant kredsløb kaldes et programmerings- eller "brænder"-kredsløb. Sidst vil jeg beskrive det nødvendige programmel til krydsoversættelse af C- og assemblerprogrammer, samt overførsel af programmer til PIC'en.

2.3.1 Minimalt kredsløb

Antallet af nødvendige ydre komponenter, som er krævet for at få en PIC-processor til at fungere, er minimalt. På figur 1 ses et sådant kredsløb. 7 ydre komponenter er alt, hvad der er nødvendigt, og dette inkluderer endda en spændingsregulator, som giver en stabil 5V-spænding, som også kan anvendes til andet logikkredsløb.



Figur 1: Minimalt PIC-kredsløb

Spændingsregulatoren LM7805 og dens to hjælpekondensatorer (på $0,33\mu F$ og $0,1\mu F$) er ikke strengt nødvendige: PIC'en accepterer en forsyningspænding på mellem $4,75V$ og $5,25V$, som strengt taget godt kunne komme direkte fra et batteri, eller en anden ekstern stabil strømforsyning. Da jeg i denne opgave antager, at målet med kredsløbet er at styre en kørende robot, vil jeg gå ud fra,

at batterier er den anvendte strømforsyning, og derfor er spændingsregulatoren berettiget. Se også afsnit 3. Microchip kalder forsyningsspændingsbenet for V_{DD} , men jeg kalder forsyningsspændingen for V_{cc} , hvilket er normal konvention. I resten af denne tekst antages det implicit, at $V_{cc} = 5V$.

Modstanden på $10K\Omega$ er en *pull-up* modstand (se afsnit 4.3), som holder \overline{MCLR} høj. \overline{MCLR} resetter processoren, hvis den bliver sat til logisk lav.

$20Mzh$ -krystallet og de to hjælpe-kondensatorer genererer PIC'ens klokfrekvens.

Kredsløbet på figur 1 kræver dog, at der allerede er overført et program til PIC'ens hukommelse. Et sådant programmeringskredsløb præsenteres herunder.

2.3.2 Programmeringskredsløb

Kredsløbet på figur 2 forbinder en PIC via et serielkabel til en PCs RS-232-serielport. Kredsløbet er bygget analogt til Brinck Elektroniks [Bri(2003b)] Br 873-byggesæt ([Bri(2003a)]), som igen er kompatibelt med Jens Dyekjær Madsens "PIC-Programmer 2 for PIC16C84 etc.", også kendt som "JDM-programmer", se [Madsen(2000)]. Br 873-byggesættet er et godt sted at starte, hvis man ikke har mod på at bygge sin egen brænder efter figur 2.

V_{cc} på figur 2 skal forbindes til en $5V$ strømforsyning, f.eks. spændingsregulatoren fra figur 1.

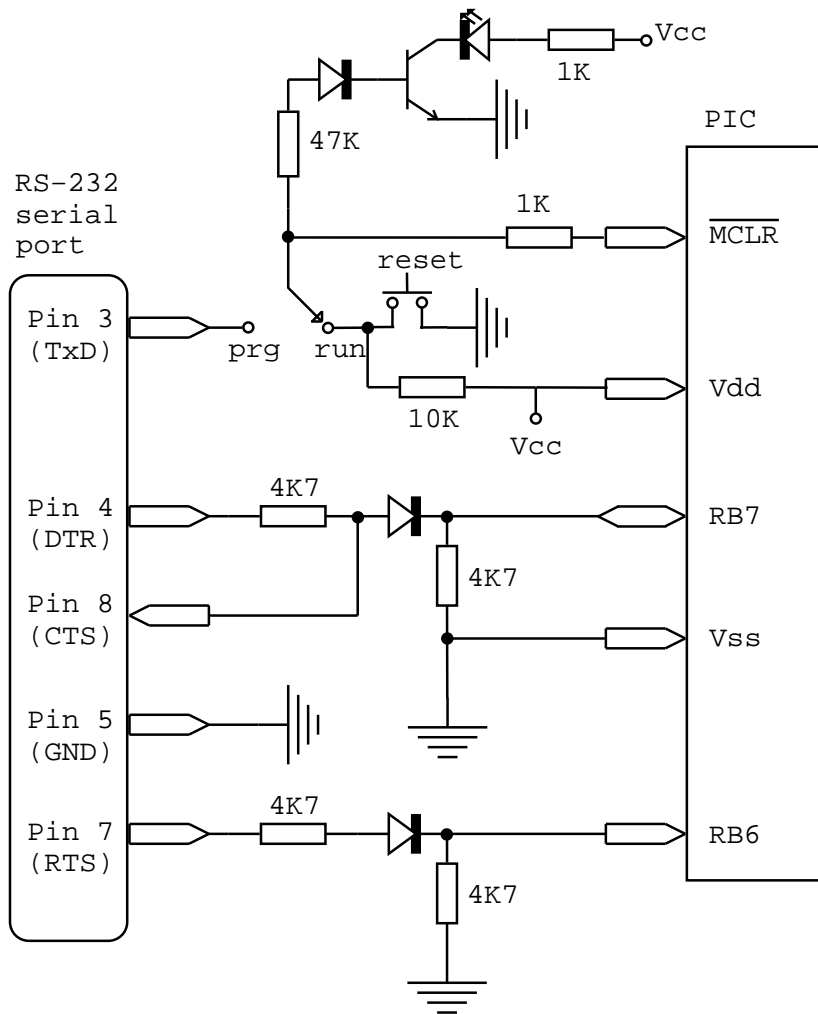
Diodetyper er ikke kritiske for virkemåden, og kan f.eks. være af typen 1N4148. Transistortypen er heller ikke kritisk, og kan f.eks. være en BC547.

Hvis PIC'en skal kunne køre det overførte program, mens den sidder i kredsløbet, er det nødvendigt at tilføje et krystal og to hjælpe-kondensatorer, f.eks. som på figur 1. "prg/run"-skydeomskifteren skal så sættes i "run"-stilling for at køre det overførte program.

Det, kredsløbet gør, er blot at justere RS-232-portens logiske niveauer på $\pm 13V$ til PIC'ens logiske niveauer på $0V$ og $5V$, se også afsnit 7.2. Dette gøres vha. to spændingsdelere hver bestående af to modstande og en diode. Desuden bruges RS-232-portens $+13V$ på \overline{MCLR}/V_{PP} -benet, hvilket sætter PIC'en i programmeringsmode. Se også referencemanualen [Mic(1997)] under "In-Circuit Serial Programming", referencemanualen [Mic(2002)] og afsnit 7.2. Det er dog ikke nødvendigt at vide, hvordan programmeringsmodet rent teknisk virker, for at overføre programmer til PIC'en. Det kan overlades til overførselsprogrammet, se afsnit 2.3.3.

Endeligt indeholder kredsløbet en resetknap, som kan resette processoren, samt en status-lysdiode, som lyser, både når der overføres program til processoren, og når den udfører et program.

Kredsløbet antager, at serielporten på PC'en afgiver signaler på ca. $\pm 13V$. Det er dog ikke altid tilfældet — Især bærbare serielporte fungerer ved væsentligt lavere spændinger, f.eks. $\pm 8V$. $8V$ er ikke nok til at aktivere PIC'ens programmeringsmode, jf. [Mic(1997)]. Kredsløbet kan godt ændres til at fungere ved $\pm 8V$, men denne problemstilling ligger uden for rammerne af denne opgave.



Figur 2: Programmeringskredsløb.

2.3.3 PC-software og overførsel af programmer

Programmer til PIC'en skrives på en almindelig PC. Assembleren `gpasm` (til Linux), som er en del af `gputils` [Fre(2003b)] er fri software og kan bruges til at skrive assemblerprogrammer til PIC'en med. Desværre findes der ikke en fri C-krydsoversætter til PIC'en, men C-krydsoversættere kan købes fra CCS [CCS(2003)] og Hi-Tech [Hi-(2003)].

Fælles for assemblere og krydsoversættere er, at de oversætter til PIC-maskinsprog gemt i Intel IHX-format, se [Wieman(2003)] og [Int(1988)]. Konventionen for programmer i Intel IHX-format er at gemme dem i filer med suffix `.HEX`.

For at overføre det oversatte program i IHX-format til PIC'en, skal man benyttes et program, som understøtter det anvendte programmeringskredsløb. Programmerne `picprog` [Hyvätti(2003)] (til Linux) og `IC-Prog` [Gijzen(2003)] (til Windows) understøtter kredsløbet i afsnit 2.3.2, der som nævnt er kompatibelt med "JDM-programmer" (skal vælges som det anvendte programmeringskredsløb i `IC-Prog`).

Programmer kan også overføres med en bootloader, se afsnit 2.5.

2.4 Eksempelprogram

Der præsenteres her et minimalt eksempelprogram i assembler for at give læseren en ide om, hvad der som minimum behøves for at skrive et program til PIC'en. Eksempelprogrammet kan ses i appendix A, og på den vedlagte cd-rom i mappen `src/minimum.example/`.

Eksempelprogrammet får en lysdiode forbundet til RA4 til at blinke (afsnit 6.1 forklarer, hvordan man forbinder en lysdiode til en PIC). At få en lysdiode til at blinke er de indlejrede processorers svar på et "Hello world"-program — Man tester, at en minimal version af hardware og software fungerer.

Det, der er specielt interessant ved programmet og ved programmer til PIC'en i almindelighed, er programlinjen:

```
__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_OFF & _HS_OSC & _LVP_OFF
```

Her defineres PIC'ens *device configuration word* (da: konfigurationsord), også kaldet PIC'ens *fuses* (da: sikringer). Konfigurationsordet overføres sammen med programmet til PIC'en, og kan *ikke* ændres af et kørende program. Konfigurationsordet definerer bla. det anvendte krystal (jf. afsnit 2.3.1), om det kørende program kan skrive i programlageret, og om *Watchdog Timer*'en skal benyttes. Her er det især vigtigt at definere det rigtige krystal og at slå *Watchdog Timer*'en fra, hvis den ikke specifikt ønskes anvendt. Den ovenstående konfigurationslinje passer til kredsløbet i figur 1 og selvsagt også til eksempelprogrammet. Se også referencemanualen [Mic(2001)] under "Configuration Bits" for mere information om PIC'ens konfigurationsord.

2.5 Bootloader

Når PIC'en programmeres med programmeringskredsløbet i afsnit 2.3.2, tager overførslen forholdsvis lang tid. Dette vil specielt vise sig ved lange programmer, hvor overførselstiden kan komme op på 5 minutter. Det, der tager tid, er den specielle "In-Circuit Serial Programming"-protokol, som er væsentligt langsommere end f.eks. den almindelige RS-232-protokol.

Da PIC'en har indbygget USART, og da den kan skrive i sit eget programlager, kan programmer også overføres over RS-232 (for tilslutning af PIC til RS-232, se afsnit 7.2). Det kræver blot et program på PIC'en, som kan modtage programdata over serielforbindelsen og skrive dem det rette sted i programlageret. I nogle tilfælde kræves desuden program på PC'en, som kan sende programdata. Et sådant PIC-program kaldes en *bootloader* og kan mindske overførselstiden til ca. 30 sekunder for et program, som fylder hele hukommelsen.

Jeg har skrevet en bootloader til PIC16F876, som let kan modificeres til at kunne anvendes til andre PIC16F8xx-modeller. Kildekoden, samt dokumentation til bootladeren, kan ses på den vedlagte cd-rom i mappen `src/bootloader/`, nyeste version af koden findes på websiden [Kjeldsen(2003a)].

Bootladeren kræver, at PIC'ens *RX*- og *TX*-ben er forbundet gennem en *level-shifter* til *TX* og *RX* på PC'ens serielport. Dette vil blive uddybet i afsnit 7.2.

For at anvende bootladeren, som ligger i programfilen `bootloader.hex`, skal denne overføres på normal vis til PIC'en. Hver gang PIC'en derefter resettes, lytter bootladeren i et halvt sekund på serielporten. Hvis overføringsprogrammet `boottransfer` er startet på PC'en med en PIC-programfil i Intel IHX-format som argument, overføres den til PIC'ens hukommelse, og programmet startes herefter. Da et kørende PIC-program ikke kan skrive konfigurationsordet, kan bootladeren ikke skrive dette, og det ignoreres, hvis det indgår i det overførte program. Bootladeren tjekker, om det overførte program vil overskrive bootladeren selv eller resetvektoren og giver en fejlmeddelelse, hvis det er tilfældet.

2.6 Programmeringsteknikker

Da indlejrede processorer sjældent har en FPU, og da de som regel kun har små mængder hukommelse til rådighed, må der anvendes programmeringsteknikker, som kan omgå disse hændringer.

2.6.1 Fastkommaaritmetik

PIC'en har ikke en FPU, ligesom den hverken har divisions- eller multiplikationsmaskinkodeoperationer. Anvendes flydendekommaaritmetik, kommer programmerne derfor let til at fylde meget, samtidig med at de kører langsomt. Dette kan afhjælpes ved at anvende *fastkommaaritmetik*, hvor et decimaltal repræsenteres som et heltal ved at være multipliceret med en konstant.

Ved udregninger kan en sådan heltalsrepræsentation direkte anvendes uden problemer. Dog skal man ved multiplikation eller division huske, at resultatet er

multipliseret med konstanten to gange:

$$(a \cdot k) + (b \cdot k) = (a + b) \cdot k \quad (1)$$

$$(a \cdot k) \cdot (b \cdot k) = (a \cdot b) \cdot k^2 \quad (2)$$

Ligeledes skal man passe på, at mellemregninger ikke giver over- eller underflow — Med fastkommatal kan man *ikke* bare gå ud fra, at de almindelige algebraiske regneregler, som f.eks. den associative regel gælder. F.eks. gælder:

$$\frac{(a \cdot b)}{c} = \left(\frac{a}{c}\right) \cdot b \quad (3)$$

ikke altid ved brug af fastkommatal².

2.6.2 Tabelopslag og lineær interpolation

Tabelopslag i stedet for brug af tidskrævende matematiske funktioner som f.eks. de trigonometriske funktioner eller kvadratrodsfunktionen kan reducere både køretid og hukommelsesforbrug.

Lineær interpolation mellem tabelværdier kan reducere behovet for store tabeller og kan også bruges til at reducere kalibreringstabellers størrelse. Dette kræver dog, at funktionen, som tabellens værdier repræsenterer, er tilstrækkeligt tæt på at være lineær.

2.7 Andre indlejrede processorer

PIC'en er ikke den eneste indlejrede processor, som er let at anvende til specialbygget robotstyretelektronik. Atmels [Atm(2003)] AVR-serie af indlejrede processorer har stort set den samme funktionalitet som Microchips PIC-serie, og AVR-serien har endda den store fordel, at GCC C-oversætteren [Fre(2003a)] kan oversætte til AVR-arkitekturen.

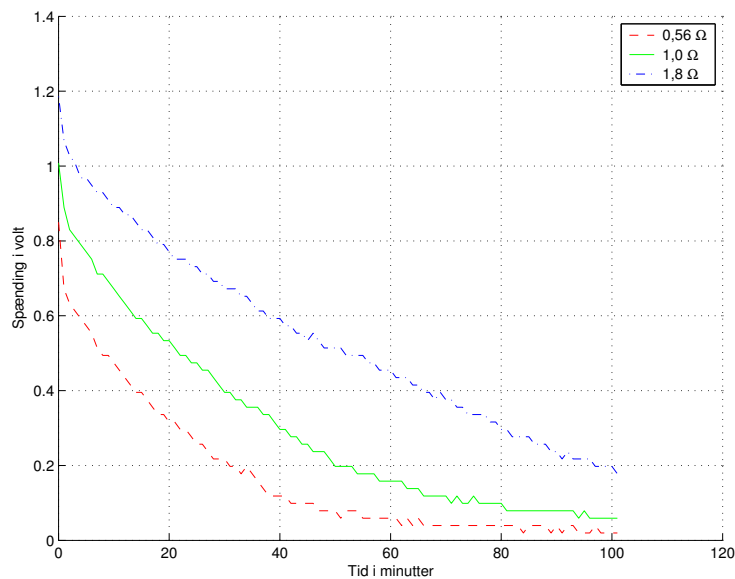
Artiklen [Lin(2002)] giver en god introduktion til den hardware og den software (til Linux), som er nødvendig for at komme i gang med at bygge AVR-baseret styreelektronik. Programmeringskredsløbet i artiklen er endda enklere end det i afsnit 2.3.2 præsenterede kredsløb. Grunden til at jeg i denne tekst har valgt at beskrive PIC'en er, at jeg først for nylig har stiftet bekendskab med AVR-serien. Den ser meget lovende ud.

3 Strømforsyning

Som nævnt i afsnit 2.3.1 er batterier et oplagt valg som strømforsyning. Det er dog en dårlig ide at forsyne sit styrekredsløb direkte fra batterier. Batterier giver

²De algebraiske regler gælder heller ikke altid ved flydendekommaaritmetik. Faktisk kan man sagtens komme ud for, at udtrykket: $a + 1 = a$ er sandt!

nemlig ikke præcis den spænding, som er trykt på dem, f.eks. $1,5V$ på almindelige brunstensbatterier, eller $1,2V$ på genopladelige batterier. Det trykte tal er kun et udtryk for den “gennemsnitlige” batterispænding. Den faktiske spænding afhænger af batteriets opladningsgrad, som falder gradvist ved strømforbrug, og af den aktuelle strømstyrke, som trækkes fra batteriet. På figur 3, 4 og 5 ses testmålinger af to typer brunstensbatterier og en type af NiMH (nikkel metalhydrid) genopladelige batterier. Hver batteritype er testet ved at tilslutte tre batterier af den aktuelle type til tre forskellige små modstande, for at teste deres egenskaber ved forskellige effekttræk.

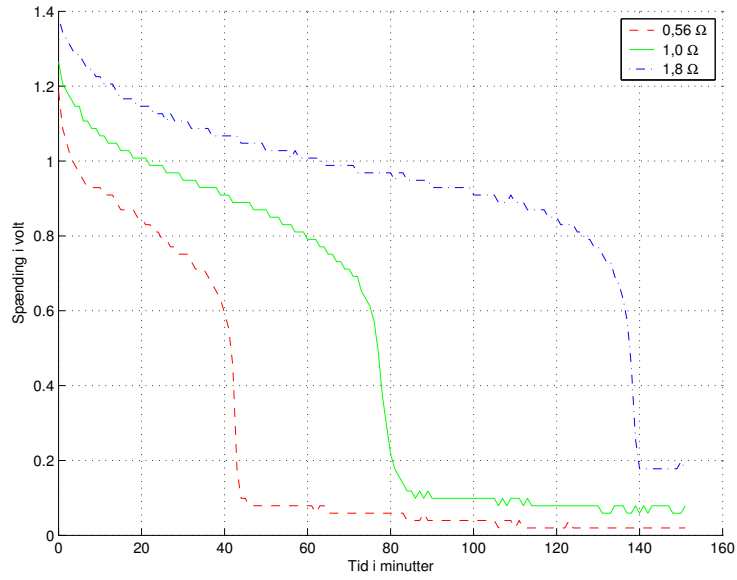


Figur 3: Testmåling af “Philips Super” AA brunstensbatterier.

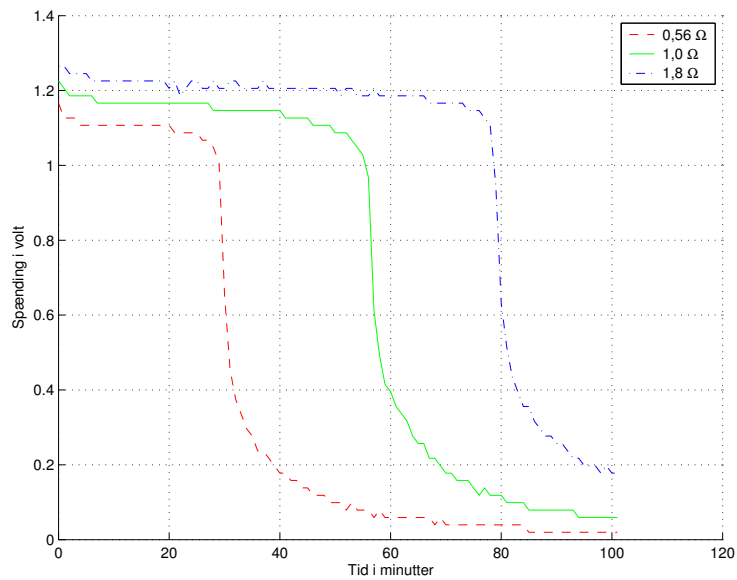
Som det ses af graferne, giver brunstensbatterierne en faldende spænding over batteriernes levetid, værst med Philipsbatterierne. NiMH-batterierne giver en næsten konstant spænding i det meste af deres levetid, hvorefter spændingen hurtigt falder. Selv om Duracellbatterierne holder længere end NiMH-batterierne, kan dette ikke opveje deres faldende spænding.

Som nævnt i afsnit 2.3.1 kan man afhjælpe batteriernes ustabile spænding ved at anvende en spændingsregulator, som f.eks. LM7805. Ulempen ved at anvende en spændingsregulator er, at den, for LM7805’s vedkommende, skal have en forsyningspænding på mindst $7V$, og at noget af effekten afsættes i den som varme. Se dataarket [Fai(2001)], og se også afsnit 6.4 for løsning af problemer ved store strømtræk.

Til små mobile robotter kan anbefales genopladelige batterier af typen NiMH frem for ikkegenopladelige brunstensbatterier, da de som vist giver en mere stabil batterispænding, og da man hurtigt vil få brugt endog rigtig mange brunstensbatterier. Derfor er genopladelige batterier mere praktiske, mere økonomiske — og mere miljørigtige — også selv om de let koster 35 kroner stykket. Blyakkumulatorer kan også anvendes, de er billigere, men de er store og tunge, og derfor



Figur 4: Testmåling af “Duracell Plus” AA brunstensbatterier.



Figur 5: Testmåling af “Uniross” 1500mAh AA genopladelige NiMH-batterier.

ikke så anvendelige på små robotter.

4 Digitalelektronik

I dette afsnit gennemgås kort nogle enkelte teknikker og faldgruber, som er praktiske at kende til, når man har med digitalelektronik at gøre. Det er ikke ment som en introduktion til digitalelektronik. For en sådan se f.eks. [Jones(1993)].

4.1 TTL logik og integrerede kredse

I modsætning til elektronik i almindelighed er digitalelektronik ganske let at gå til, hvis man har en datalogisk baggrund. Logiske gates (porte) fra datalogien er i digitalelektronikken implementeret i hardware, og de logiske værdier 0 og 1 bliver i TTL-elektronik repræsenteret som hhv. 0V og 5V. Integrerede kredse, som implementerer diverse gates og andre logiske funktioner, fås som standardmodeller f.eks. i HEF4000-serien [Phi(2003b)].

4.2 Symboler

I diagrammerne i denne tekst betyder V_{cc} forsyningsspændingen, som tidligere nævnt implicit antages at være 5V. GND (ground) betyder jord eller 0V (relativt til V_{cc}). V_{bat} betegner batterispændingen.

Symbolerne (for modstande, kondensatorer etc.) er de normalt anvendte europæiske symboler. Betydningen af disse kan findes i introducerende elektronikbøger, f.eks. [Jones(1993)].

4.3 Pull up/pull down modstand

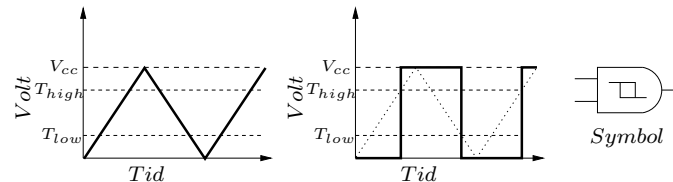
Efterlades et ben på en logik-IC uforbundet skal man være opmærksom på, at det *ikke* er det samme, som at benet er logisk 0, hvilket man måske godt kunne tro. Udelades f.eks. modstanden på venstre kredsløb på figur 7, vil benets tilstand være udefineret, når kontakten ikke er sluttet, og blot ganske små mængder støj kan få benet til at skifte logisk tilstand. En sådan modstand kaldes en *pull-down*-modstand, da den holder benets logiske niveau nede. En *pull-up*-modstand holder derimod det logiske niveau højt ved at være forbundet til V_{cc} .

Nogle IC'er har indbygget pull-up/pull-down-modstande, og f.eks. på PIC'ens port B kan interne pull-up-modstande tilvælges i software.

4.4 Schmitt-trigger

Støjfyldte signaler, som ikke hurtigt og præcist møder TTL-logikkredses input-specifikationer, kan få en gate til at skifte logisk tilstand utilsigtet mange gange hurtigt efter hinanden. Dette er meget upraktisk, hvis man f.eks. har en tæller tilsluttet porten. En løsning herpå kan være at anvende en *Schmitt-trigger*. Den

kan omforme et støjfyldt signal til et pænt og skarpt signal. Den fungerer ved først at skifte fra 0 til 1, når indgangsspændingen kommer over et vist niveau V_{high} , og fra 1 til 0, når spændingen kommer under et bestemt niveau V_{low} , se figur 6.



Figur 6: Til venstre: Eksempel på indgangsspænding til en Schmitt-trigger som funktion af tid. I midten: Den tilsvarende udgangsspænding fra Schmitt-triggeren. Til højre: Symbolet for en and-gate med Schmitt-trigger på indgangene.

Visse af PIC'ens I/O-ben har en indbygget Schmitt-trigger.

4.5 Praktisk implementation

Når et kredsløb skal opbygges, kan man anvende forskellige “platforme” til at tilslutte de anvendte komponenter på. Et *fumlebræt* er et plastic-“bræt” med en masse huller med stik i, som er elektrisk forbundet rækkevist. Det er praktisk at bruge, når der skal eksperimenteres med nye kredsløb. Men komponenterne kan ikke loddes fast, og fumlebrættet er derfor upraktisk at anvende som en endelig platform for et kredsløb, som skal monteres på en robot.

Et *veroboard* er et stykke printplade med en matrix af forborede huller samt tilhørende loddeører på bagsiden. Komponenter kan loddes på, og sidder derfor godt fast. Veroboardet er fleksibelt, men kredsløbet kan let blive noget rodet, da man sjældent kan nøjes med loddeøerne på bagsiden til at lave forbindelse mellem komponenterne, men i stedet må ty til ledningsforbindelser. Veroboard kan anbefales til mindre komplicerede kredsløb og til eksperimenterende kredsløb.

En *PCB*-printplade (Printed Circuit Board) er en printplade, hvor kredsløbets forbindelser er lavet i kobber på pladens bagside. Komponenterne monteres i huller man selv borer. PCB er derfor mere besværligt at bruge end et veroboard, men det er ideelt, hvis man skal bruge det samme kredsløb mange gange, da man let kan kopiere et en gang udtænkt kredsløb.

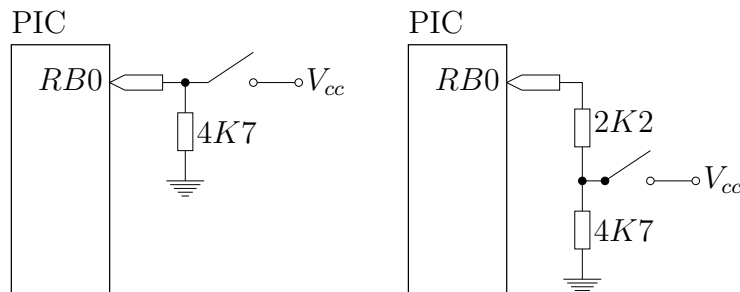
5 Sensorer

I dette kapitel gennemgås en række sensorer, som kan anvendes sammen med PIC-processorer og lignende processorer. Sensorer, som f.eks. kameraer og mikrofoner, der kræver stor båndbredde, vil ikke blive gennemgået, da PIC-processorer er dårlige til at håndtere store datamængder.

Særlig interesse skal rettes mod nogle afstandssensorer fra Sharp sidst i kapitlet. Grunden hertil er, at de er meget anvendelige til små kørende robotter, og at de er lette at anvende, hvis man kender til de forhold, som påvirker deres virkemåde. Samtidig har jeg opbygget en erfaring med dem, som jeg her gerne vil samle og videreformidle.

5.1 Digi-tast

En digi-tast (kontakt) er nok den enkleste form for sensor. Tilslutningen til processoren er også simpel, den sluttet direkte til et digitalt I/O-ben på processoren, se venstre kredsløb på figur 7. Dog skal man, jf. afsnit 4.3, huske en pull-down modstand (eller pull-up, hvis kontakten forbindes til GND), så I/O-benet ikke “flyder”, når tasten ikke er sluttet.



Figur 7: Tilslutning af digitast til processor. Venstre: Enkel tilslutning. Højre: Tilslutning sikret mod kortslutning.

Hvis I/O-benet ved en fejl bliver programmeret til at være output og logisk lav, vil det trække mere strøm, end det kan tåle. Dette kan forhindres med en enkelt ekstra modstand ($2K2$), som vist i højre kredsløb på figur 7.

Når digitasten bliver trykket ned, sluttet forbindelsen mellem V_{cc} og I/O-benet. Dog kan et enkelt tryk på kontakten blive opfattet af I/O-benet som flere, hvis der er lidt dårlig kontakt, hvilket er meget almindeligt. Dette kaldes *prelling*. Prelling kan forhindres i hardware, f.eks. med en Schmitt-trigger, men kan også let håndteres i software. En løsning er at definere en overgang fra høj til lav (hhv. lav til høj), som værende reel, når den nye tilstand har været stabil i f.eks. 100 ms.

Digitaster kan anvendes til flere formål, som brugerinput og kollisionsdetektor (f.eks. ved montering på en kofanger). På min Robocup-robot, se [Kjeldsen(2003b)], anvendte jeg en digitast til at detektere de bomme, robotten fik point for at passere under.

5.2 Måling af spænding

Mange af modellerne i Microchips PIC serie har indbygget A/D (analog til digital) konverter, hvilket gør måling af elektrisk spænding meget let. Et analogt I/O-ben tilsluttes den spænding, som ønskes målt, og A/D-konverteringen

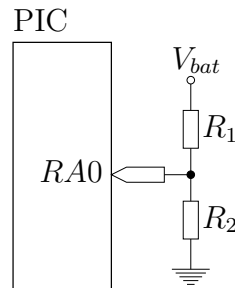
giver 8 bit eller 10 bit tal som uddata. Ved 10 bit uddata, svarer 0 til GND og 1023 svarer til V_{cc} , jf. [Mic(1997)]. PIC'ens specifikationer kræver, at spændingen ligger mellem GND og V_{cc} , og at indgangsimpedansen er højest $10K\Omega$. A/D-konverterens opsætning er omfangsrig, men problemfri når den først er implementeret, se [Mic(1997)] for specifikationerne. For et program-eksempel på, hvordan opsætningen kan implementeres, se cd-rommen under `src/controller/ad_driver.c`.

5.3 Måling af batterispænding

Det er praktisk, hvis en robot kan måle batterispændingen. Herved kan robotten tage forholdsregler i tilfælde af, at batterierne er ved at være opbrugte. Den kan f.eks. advare operatøren om den uheldige situationen vha. af en lysdiode, et display eller en advarselslyd.

5.3.1 Tilslutning til processor

Er batterispændingen V_{bat} større end V_{cc} (hvilket den nødvendigvis er, hvis PIC'en forsynes gennem en spændingsregulator), er det nødvendigt at mindske den spænding, der skal måles. Dette kan gøres vha. en *spændingsdeler* (to serielt forbundne modstande), som vist på figur 8.



Figur 8: Måling af batterispænding

Modstandene R_1 og R_2 kan nu findes ud fra Ohms lov og den højeste forventede V_{bat} . Det antages, at strømmen ind i I/O-benet er ubetydelig, altså $0A$. Ud fra Ohms lov haves så:

$$U_{R1} = R_1 I \quad \text{og} \quad U_{R2} = R_2 I \quad (4)$$

Hvis V_{bat} højst er $15V$, kan vi udregne R_1 og R_2 for $V_{bat} = 15V$, så spændingen på I/O-benet $V_{I/O}$ højst er $5V$, og så vil $V_{I/O}$ også være mindre end $5V$ for $V_{bat} < 15V$.

Antag altså $V_{bat} = 15V$. Hvis $V_{I/O}$ skal være $5V$, er der således et spændingsfald over R_2 på $U_{R2} = 5V$, hvilket medfører $U_{R1} = 10V$. De to ligninger (4) har så tre ubekendte, men hvis enten R_1 eller R_2 sættes til at være fast, kan den anden

udregnes:

$$R_2 = \frac{U_{R2}R_1}{U_{R1}} \quad (5)$$

Vælges $R_1 = 4700\Omega$ gives altså:

$$R_2 = \frac{5V \cdot 4700\Omega}{10V} = 2350\Omega \quad (6)$$

Kredsløbet kan naturligvis også bruges til at måle andre spændinger end V_{bat} . Hvis spændingen, der ønskes målt, er mindre end eller lig V_{cc} , er det enklere at udelade spændingsdeleeren, da den så er unødvendig.

5.3.2 Udregning af batterispænding

Hvis PIC'ens A/D-konverter sættes op til at give et 8 bit tal som uddata, svarer uddata $O_{A/D} = 0$ til indgangsspænding $V_{I/O} = 0V$ og $O_{A/D} = 255$ til $V_{I/O} = 5V$. Da sammenhængen er lineær, kan $V_{I/O}$ udregnes fra $O_{A/D}$ som:

$$V_{I/O} = \frac{5O_{A/D}}{255} \quad (7)$$

Batterispændingen kan nu udregnes fra formel (7) og Ohms lov, som (mellemregninger udeladt):

$$V_{bat} = \frac{5(R_1 + R_2)O_{A/D}}{255R_2} \quad (8)$$

5.4 Lysfølsom modstand

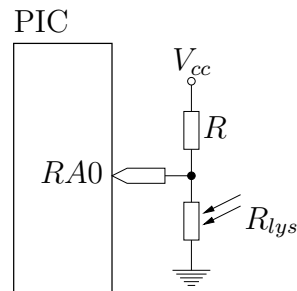
En lysfølsom modstand er en modstand, som ændrer resistans afhængig af intensiteten af det lys, som falder på den. Forhold mellem lysintensitet og resistansen afhænger af typen af lysfølsom modstand, men resistansen kan f.eks. ligge mellem 100Ω (direkte sollys) - $1M\Omega$ (helt mørke). Et løst gæt er, at sammenhængen er logaritmisk.

5.4.1 Tilslutning til processor

Tilslutningen foretages med den lysfølsomme modstand som den ene af de to modstande i en spændingsdeleer, som vist på figur 9. Den faste modstands resistans kan vælges, så den passer til den lysfølsomme modstands arbejdsområde. $R = 1K\Omega$ er passende for det førnævnte interval, men valget er ikke meget kritisk.

For en givet R , og ud fra Ohms lov, kan spændingen $V_{I/O}$ på I/O-benet udregnes for resistanser R_{lys} af den lysfølsomme modstand (mellemregninger udeladt):

$$V_{I/O} = V_{cc} - R \frac{5}{R + R_{lys}} \quad (9)$$



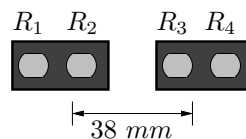
Figur 9: Tilslutning af lysfølsom modstand til processor.

Ud fra formel (9) kan udregnes, at direkte sollys ($R_{lys} = 100\Omega$) giver $V_{I/O} = 0.45V$, og at mørke ($R_{lys} = 1M\Omega$) giver $V_{I/O} = 5V$. Altså haves et passende arbejdsområde. Den lysfølsomme modstands arbejdsområde måles med et ohmmeter. Dette bør gøres for hvert ekstremum af det ønskede lysintensitetsinterval, så en passende værdi af R kan vælges ud fra formel (9).

5.4.2 Eksempel på anvendelse

Lysfølsomme modstande kan eksempelvis anvendes til en *stregsensor*, som kan anvendes til at få en robot til at følge en streg. Jeg vil her beskrive den stregsensor, som jeg brugte på den robot, som deltog i Robocup 2003.

Opbygning Stregsensoren består af fire ens lysfølsomme modstande, positioneret som vist på figur 10. At jeg valgte netop fire modstande på sensoren grunder i, at det var det antal analoge I/O-ben jeg havde til rådighed på robotten. De lysfølsomme modstandes modelnummer kender jeg desværre ikke. At der ikke er lige langt mellem modstandene, skyldes at robotten typisk befinder sig midt på strengen, og at det er specielt interessant at måle lysintensiteter omkring kanten af strengen. Strengen er 38 mm bred og består af hvid lærredstape påklæbet en overflade bestående af grå fliser.



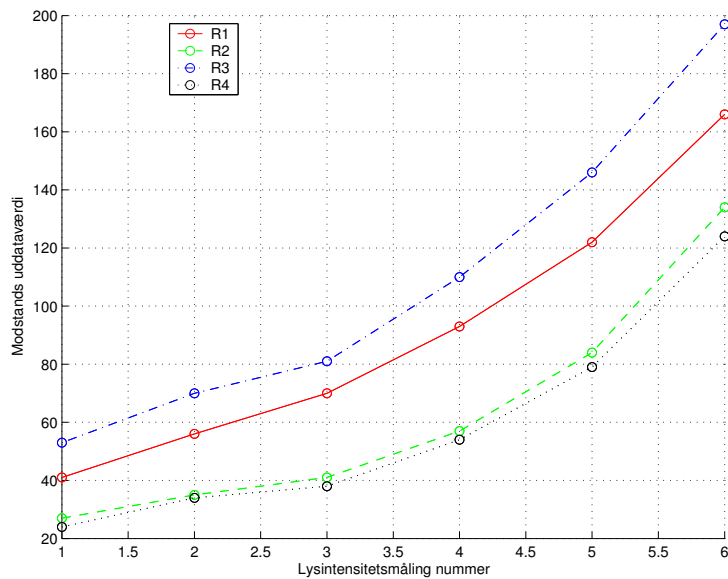
Figur 10: Stregsensoren består af fire lysfølsomme modstande (lysegrå).

På robotten er stregsensoren monteret umiddelbart foran forhjulenes aksler (robotten har forhjulsstyring) og vinklet ca. 30° , så den peger lidt ind under robotten. Dette mindsker følsomheden over for udefrakommende lys. Derudover oplyser fem kraftige røde lysdioder det område, sensoren peger ned mod.

Det, jeg ønsker af få som uddata fra sensoren, er to ting: Er der en streg under sensoren eller ej, og hvis der er, hvor langt fra midten af strengen befinder sensoren

(og dermed robotten) sig? Kald denne afstand d_s (den regnes med fortegn: positivt, hvis strengen befinder sig til højre for robotten).

Kalibrering af modstande Da modstandene, selv om de er af samme type, desværre har lidt forskellige karakteristika, giver de forskellige værdier for samme lysintensitet. Dette kan der rettes op på ved en lille kalibreringstabel. Ved at måle uddata fra modstandene i samme lysintensitet, kan sammenhængen mellem deres uddata findes. Se figur 11, hvor dette er gjort for 6 forskellige lysintensiteter.



Figur 11: De lysfølsomme modstandes uddata afhængig af lysintensitet.

R_1 regnes som værende *ground truth*, så værdierne for R_2 , R_3 og R_4 omregnes ved opslag i tabellen til den tilsvarende R_1 -værdi. Da der kun er få værdier i tabellen, bruges lineær interpolering for at udregne de manglende værdier. Dette reducerer antallet af nødvendige målinger og sparer plads i PIC'ens datalager. Ved den her beskrevne kalibrering giver alle de lysfølsomme modstande nu *samme* værdi, ved samme intensitet.

Detektion af streg Det kan nu let afgøres, om der er en streg under stregsensoren: Hvis forskellen mellem den mindste og den største af de kalibrerede modstandsværdier er større end en fast tærskelværdi, antages det at der findes en streg under sensoren. I praksis fungerer dette godt.

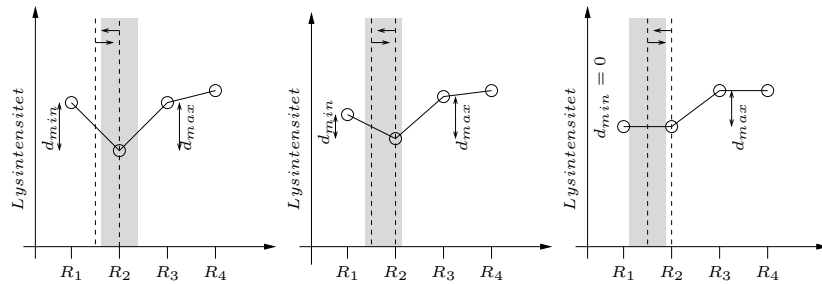
Beregning af afstand For at finde afstanden fra stregens midte til sensorens (og dermed robotten) midte, d_s , findes nu først en funktion δ_s , som afhænger af d_s , men ikke nødvendigvis lineært.

I det følgende betegner R_i af praktiske grunde både den i 'te lysfølsomme modstand, og *værdien* af den i 'te lysfølsomme modstand. Hvad der menes fremgår af sammenhængen.

Ideen er, at strengen må befinde sig i nærheden af den lysfølsomme modstand, som giver den mindste værdi, R_{min} , hvilket svarer til den højeste lysintensitet. I det sjældne tilfælde, hvor der er to mindste værdier, vælges blot en tilfældig af dem. Den enkleste løsning er nu at sætte d_s til at være afstanden fra midten af sensoren ud til modstanden R_{min} . Med 4 modstande giver dette en meget grov diskretisering på netop 4 værdier. Skal robotten styres herefter, giver det nødvendigvis en meget ustabil kørsel.

En bedre ide er, at udnytte at modstandene giver en værdi, som er afhængig af intensiteten: Jo længere fra strengen, jo højere værdi.

Eksempel på udregning af δ_s På figur 12 ses tre eksempler på uddata fra de 4 lysfølsomme modstande. Den underliggende streg starter med at befinde sig under R_2 og slutter med at befinde sig netop midt mellem R_1 og R_2 . R_2 antages i alle tre tilfælde at have den mindste værdi, svarende til den største lysintensitet, og R_1 antages at være mindre end R_3 . Da R_2 har mindste værdi, må strengen være tættere på R_2 end på de tre andre modstande. Det ønskes nu at finde et mål for, hvor langt fra R_2 stregens midte befinder sig.



Figur 12: Tre situationer, hvor strengen ligger mellem R_1 og R_2 . Stregens position er symboliseret ved den grå kasse, og stregens midte må nødvendigvis ligge mellem eller på de to stiplede linjer.

Defineres d_{min} og d_{max} som:

$$d_{min} = R_1 - R_2 \quad \text{og} \quad d_{max} = R_3 - R_2 \quad (10)$$

så giver forholdet $\Delta = \frac{d_{min}}{d_{max}}$ et mål for, hvor langt fra R_2 mod R_1 stregens midte befinder sig. Som det også ses af figuren, gælder at $0 \leq \Delta \leq 1$.

Dette forhold kan bruges til at foretage en interpolation af stregens position. Hvis p_i betegner den i 'te modstands afstand fra midten af robotten, regnet med positivt fortegn til højre for midten, haves:

$$\delta_s = p_2 - (p_2 - p_1)\left(1 - \frac{d_{min}}{d_{max}}\right) \quad (11)$$

Generaliseret udregning af δ_s Fra ovenstående eksempel kan en generaliseret udgave af formel (11) udledes. Hvis i betegner indeks for den modstand, som har mindste værdi af alle de 4 modstande, $imin$ betegner indeks for den af de to naboer til R_i som har den mindste værdi, og $imax$ betegner indeks for den af de to naboer til R_i , som har den største værdi, kan formel (11) generaliseres til formel (12):

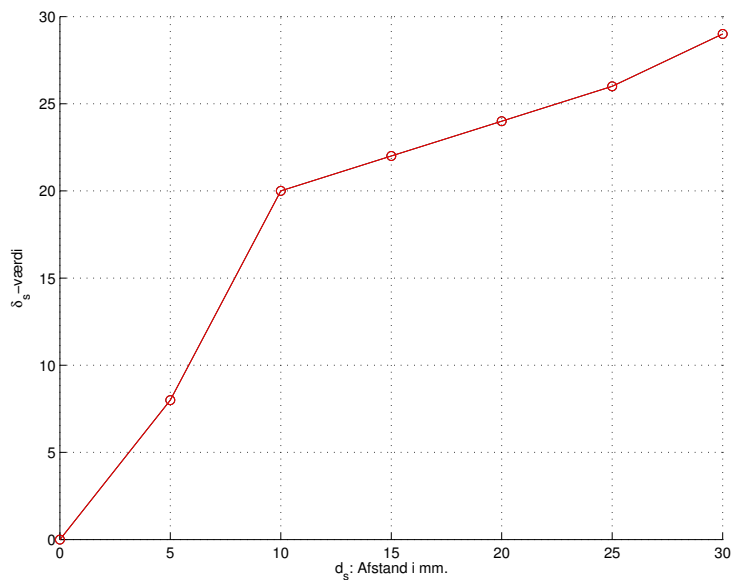
$$\delta_s = \frac{(1 + \Delta)p_i + (1 - \Delta)p_{imin}}{2} \quad (12)$$

Her er Δ givet ved:

$$\Delta = \frac{R_{imin} - R_i}{R_{imax} - R_i} \quad (13)$$

Denne generalisering har et randproblem, nemlig hvor R_1 eller R_4 har den mindste værdi. De to modstande har nemlig kun een nabo. Her kan $imin$ vælges som indeks for den nabo, som eksisterer, og $imax$ vælges til indeks for en tilfældig af de to modstande, som endnu ikke er valgt. Hvorfor dette virker, vil jeg ikke komme nærmere ind på her, men hovedsagen er, at det virker i praksis.

d_s kan nu findes ud fra δ_s , ved opslag i en tabel af δ_s for en række kendte værdier af d_s . Lineær interpolering kan også her anvendes for de værdier af δ_s , som ikke findes i tabellen, for at spare lagerplads. Figur 13 viser, hvordan denne tabel ser ud for min streksensor.



Figur 13: Kalibreringstabel til udregning af d_s ud fra δ_s . For hver 5 mm er δ_s noteret.

Styring efter d_s Når robotten skal følge strengen ud fra d_s , haves et reguleringsproblem: Robottens forhjul skal sættes i en vinkel θ_h beregnet ud fra nuværende og eventuelt tidligere d_s , robottens vinkel i forhold til strengen og robottens nuværende hastighed. Dette kan synes som et temmeligt komplekst problem, især da robottens vinkel i forhold til strengen ikke er kendt. I praksis fungerer den meget enkle formel (14) rigtigt godt, forbeholdt at robotten ikke kører alt for hurtigt:

$$\theta_h = cd_s \quad (14)$$

Her er c en dæmpningskonstant, som kan findes empirisk, den afhænger af den enkelte robots konstruktion. Sættes c for højt, vil robotten slingre, og sættes c for lavt, vil robotten styre "trægt" og derfor være dårlig til at følge sving på strengen.

5.4.3 Andre lyssensorer

Ud over lysfølsomme modstande findes en række andre lysfølsomme elektroniske komponenter, som f.eks. lysfølsomme dioder og lysfølsomme transistorer (fotodioder og fototransistorer). En nærmere beskrivelse af disse vil jeg ikke komme ind på her, men de er heller ikke svære at forbinde til en microprocessor. Forbindelse af en fototransistor til PIC'en beskrives kort i gennemgangen af tachometeret i afsnit 5.5.

5.5 Tachometer

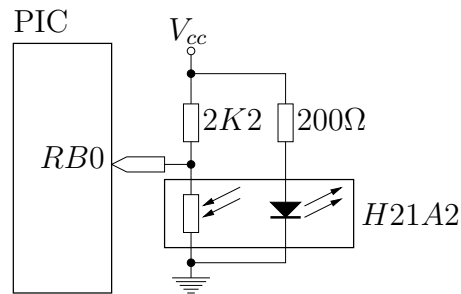
Et tachometer er en hastighedsmåler ("speedometer"). Et tachometer er essentielt på en kørende robot for at kunne regulere robottens hastighed. Det kan også bruges til at måle kørt afstand med og til beregning af robottens acceleration.

En enkel metode at konstruere et tachometer på er at montere en skive med huller i på en af robottens hjulaksler og at anvende en *optogaffel* til at registrere skivens rotationshastighed med. Det er denne metode en almindelig computer-mus anvender. En optogaffel er blot et plastichus, hvori der er monteret en lyskilde og en lyssensor.

Et eksempel på en optogaffel er Fairchild's H21A2 [Fai(2003)], som består af en infrarød lysdiode som lyskilde og en fototransistor, der er følsom over for infrarødt lys, som lyssensor. Man kan også bygge sin egen optogaffel af en lysdiode og en fototransistor eller lysfølsom modstand, hvis man ikke kan finde en til robotten passende optogaffel.

Tilslutning foregår analogt til tilslutning af en lysfølsom modstand, se figur (14). Anvendes en anden fototransistor, skal modstanden på 2,2 $K\Omega$ på figuren tilpasses den anvendte fototransistor.

Det er praktisk at tilslutte tachometeret til et I/O-ben med Schmitt-trigger (se afsnit 4.4), da det forhindrer prelling. Det er også praktisk at vælge et I/O-ben med interrupt- eller tællerfunktion, så programmet ikke konstant skal aflæse I/O-benet for at undersøge, om der er sket en ændring i dets tilstand.



Figur 14: Tilslutning af optogaffel (Fairchild H21A2) til processor.

Afstanden kan nu måles som antallet af impulser fra tachometeret, eventuelt kan der omregnes til en passende SI-enhed, f.eks centimeter, ved kalibrering.

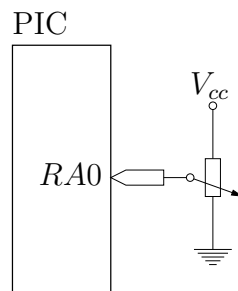
Hastigheden kan udregnes ved at tælle antallet af impulser i løbet af en fast periode eller ud fra tiden mellem to på hinanden følgende impulser.

For at øge præcisionen kan der eventuelt tælles både på impulsens stigende og faldende flanke.

For et mere avanceret eksempel på anvendelse af et tachometer, se afsnit 8.

5.6 Dreje- og skydepotentiometer

Et potentiometer fungerer som en spændingsdeler, man kan indstille. Det kan indstilles, så det kan give spændinger mellem 0 og V_{cc} , hvis det tilsluttes som på figur 15.



Figur 15: Tilslutning af potentiometer til I/O-ben med A/D-konverter.

Et drejepotentiometer kan f.eks anvendes til at måle en akse vinkel (men kan normalt ikke dreje hele vejen rundt), og et skydepotentiometer kan bruges til at måle translation.

5.7 Sharp IR-afstandssensor

En serie af forholdsvis billige (ca. \$20), præcise og letanvendelige afstandssensorer fremstilles af Sharp. Serien hedder GP2xxx. Denne sensor er let at forbinde til en microprocessor og kan give endog meget præcise afstandsmålinger i intervallet 10 til 100 centimeter afhængigt af, hvilken model der anvendes. Se [Sha(2003c)] for en liste af afstandssensorer fra Sharp.

Præcisionen af sensoren afhænger dog i høj grad af, at den bliver anvendt rigtigt. Dette afsnit vil forklare, hvordan sensoren virker og tilsluttes, samt vise og forklare en række af de forhold, som influerer på sensorens målinger.

5.7.1 Virkemåde

Den sensormodel, der her skal gennemgås, er Sharp GP2D02. Resultaterne formodes at kunne generaliseres til de andre modeller i serien, da de er opbygget på en tilsvarende måde.

Fælles for alle modellerne er, at de er baseret på et par bestående af en infrarød (IR) lysdiode (IR-LED) og en infrarød lyssensor. IR-lysdioden fungerer som en almindelig lysdiode, men sender lys i det infrarøde spektrum. IR-lyssensoren er et lille CCD-array, svarende til et lille eenskanlinjes ccd-kamera, som er følsomt i det infrarøde spektrum. Lysdioden og CCD-arrayet er begge fastmonteret i et plastickabinet.

Sensoren måler afstanden til et objekt ved triangulering, se figur 16. De forskellige typer sensorer giver som uddata enten en spænding i intervallet 0 til 5V eller et 8 bit tal (0–255). GP2D02 giver et 8 bit tal.

CCD-array'et kan modeleres som et hulcamera (eng: pinhole camera), jf. [Sha(2003a)]. Dette er gjort på figur 16, og af figuren kan der, ved betragtning af trekanter, ses:

$$\frac{f}{d} = \frac{D}{s} \Leftrightarrow D = \frac{fs}{d} \quad (15)$$

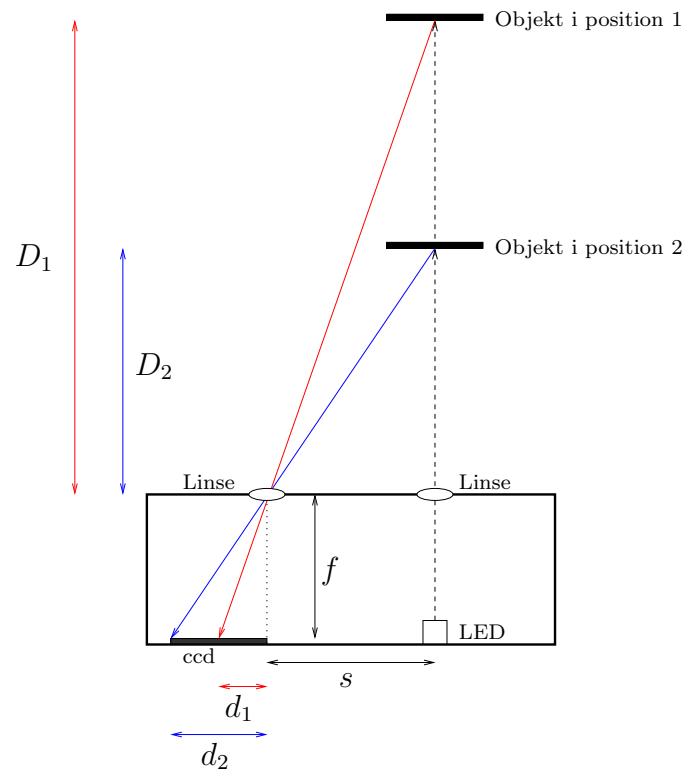
Afhængigt af hvor koordinatsystemerne har deres origo, kan det være nødvendigt at indføre translatoriske konstanter d_0 , D_0 :

$$D = \frac{fs}{d - d_0} + D_0 \quad (16)$$

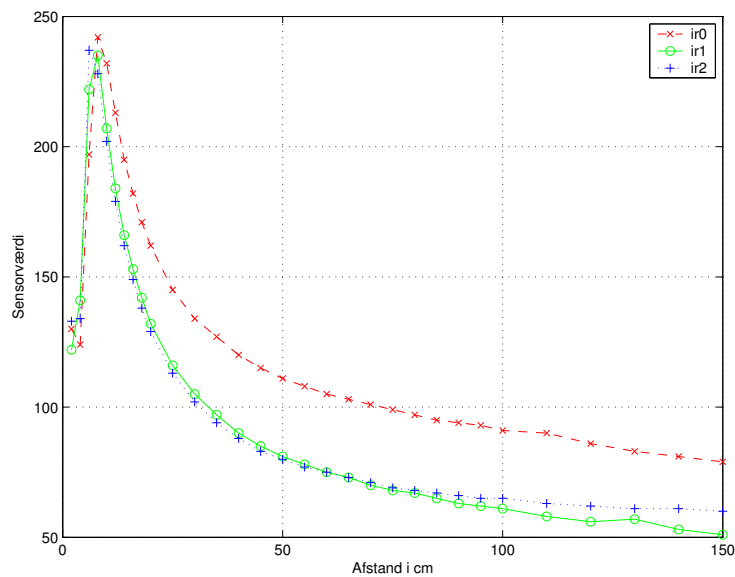
Jeg har tre GP2D02-sensorer til rådighed, i denne tekst navngivet ir0, ir1 og ir2. Foretages en testmåling af dem, giver det graferne på figur 17. GP2D02 er specificeret til at måle korrekte afstande i intervallet 10 til 80 cm, jf. [Acr(2003b)], og som det ses, giver testmålingerne en graf, som minder om formel (16) i dette interval. Dette uddybes senere i dette afsnit.

Hvis intet andet er nævnt, er alle testmålinger i dette afsnit foretaget under følgende forhold:

- Belysning: Rummet er dunkelt: Indendørs i dagslys, med gardiner for vindue.



Figur 16: Hulkameramodel af GP2D02: Lysdioden (LED) udsender IR-lys i en tynd lysstribe (stiplet linje). Afhængig af objektets position reflekterer det lysstrålen, så den rammer forskellige steder (fuldt optrukne linjer) på CCD-array'et.



Figur 17: Testmåling af tre Sharp GP2D02 afstandssensorer: Sensorværdi som funktion af afstand for de tre sensorer.

- Objekt: Mat hvidt “printerpapir” på plan overflade.
- Orientering: Vinkelret på objektet, vertikalt monteret sensor (dvs. IR-LED’en er oppe, og CCD-array’et er nede), se figur 25.
- Objektet står stille under målingen, og rumtemperaturen er ca. 20°.

Samtlige testmålinger findes som matlab-filer på den vedlagte cd-rom i mappen `matlab/`, navngivet `irmaaling*.m`.

5.7.2 Kalibrering

Formel (16) kan bruges til at kalibrere sensoren, så afstanden i centimeter kan beregnes ud fra sensorens uddata. Først omskrives formel (16). $f \cdot s$ kan omskrives til en enkelt konstant, fs , og ud fra målinger har jeg erfaret, at D_0 kan sættes til 0:

$$D = \frac{fs}{d - d_0} \quad (17)$$

Formel (17) er også den, som [Barello(2003)] er kommet frem til. Han giver dog ikke en forklaring på, hvorfor den virker. fs og d_0 kan nu findes, enten ved mindste kvadraters metode, eller mere enkelt ved at løse to ligninger med to ubekendte, nemlig formel (17) med hhv. (d_1, D_1) og (d_2, D_2) . Sidstnævnte metode har den fordel, at med bare to målinger kan en sensor kalibreres, og på trods af dette spinkle grundlag bliver resultatet meget præcist.

fs og d_0 findes således ved at løse to ligninger med to ubekendte (mellemregninger udeladt):

$$fs = \frac{D_1 D_2 (d_2 - d_1)}{D_1 - D_2} \quad (18)$$

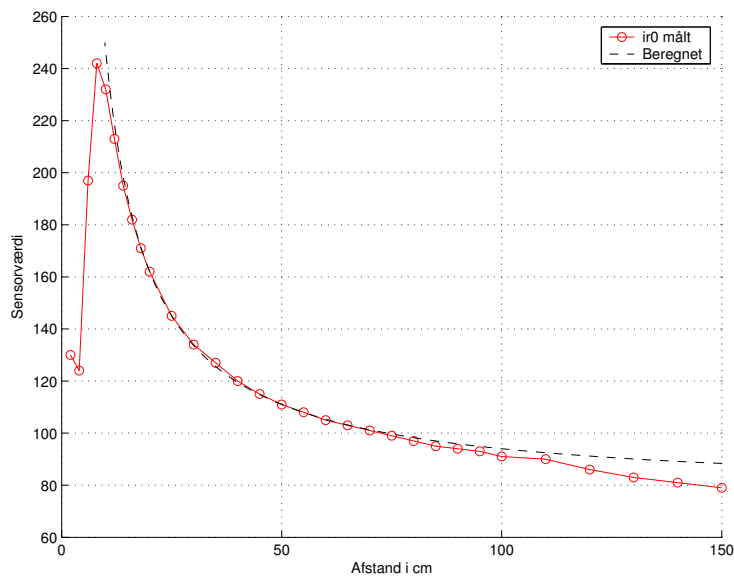
$$d_0 = \frac{D_1 d_1 - D_2 d_2}{D_1 - D_2} \quad (19)$$

Findes nu fs og d_0 for sensor ir0 i testmålingen fra figur 17, for sensorværdierne ved hhv. $(D_1, d_1) = (20 \text{ cm}, 162)$ og $(D_2, d_2) = (50 \text{ cm}, 111)$:

$$fs = \frac{20 \cdot 50 \cdot (111 - 162)}{20 - 50} = 1700$$

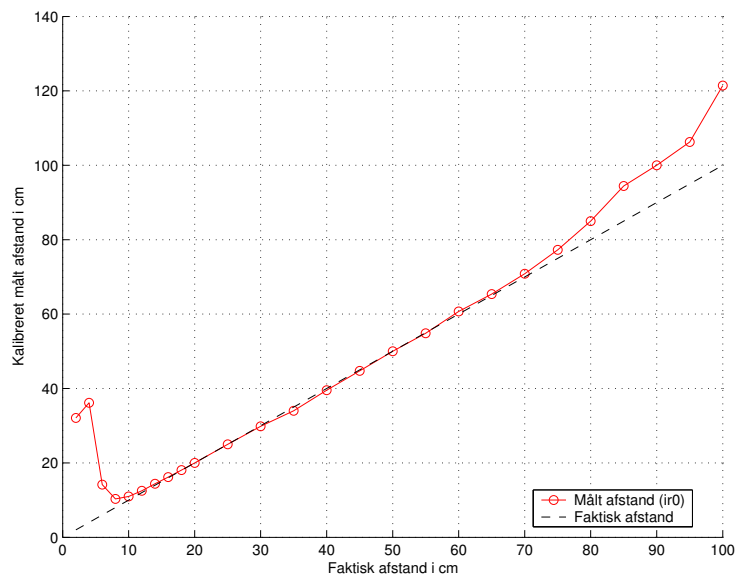
$$d_0 = \frac{20 \cdot 162 - 50 \cdot 111}{20 - 50} = 77$$

Bruges formel (17) sammen med ovenstående fs og d_0 , kan sensorens output approksimeres temmeligt nøjagtigt, se figur 18. Ligeledes kan uddata fra sensoren vha. formlen omregnes til afstande i centimeter, og også her fås rigtig gode resultater, se figur 19.



Figur 18: Sensorværdi som funktion af afstand. Afstanden er hhv. målt og beregnet ud fra formel (17) med $fs = 1700$ og $d_0 = 77$.

En forudsætning, for at ovenstående kalibrering virker, er, at afstanden til objektet er over ca. 10 cm. Er afstanden mindre, vil den målte afstand blive længere end den faktiske afstand, jf. sensorens uddata for afstande under 10 cm, se figur 17. Ved at montere sensoren på en praktisk måde, kan denne situation helt



Figur 19: Den med sensoren målte og derefter med formel (17) kalibrerede afstand som funktion af den faktiske afstand ($f_s = 1700$ og $d_0 = 77$), samt den faktiske afstand som funktion af sig selv til sammenligning.

undgå, se afsnit 5.7.5. Jeg har ingen forklaring for de tvetydige sensorværdier for afstande under 10 cm, men de er reproducerbare og afhænger deterministisk af afstanden, og er derfor ikke bare et udslag af tilfældige værdier.

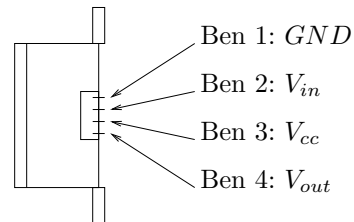
En anden metode til at kalibrere sensoren på, er ganske simpelt at lave en tabel, som for hver sensorværdi i intervallet 0–255 indeholder den tilsvarende faktiske afstand. Denne metode anvender Eyebot’erne i deres Robios operativsystem, se [Eye(2003)]. Metoden er dog meget tidskrævende og giver generelt ikke bedre resultater end kalibrering vha. formel (17).

5.7.3 Tilslutning til processor

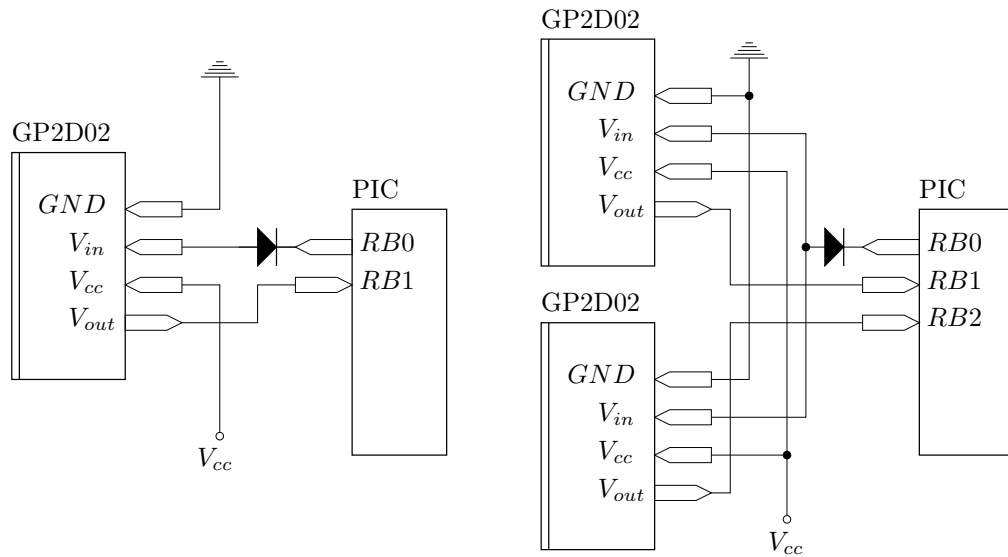
Tilslutningen af GP2D02 til en mikroprocessor er forholdsvis enkel. På figur 20 er placeringen af sensorens ben indtegnet. Sensoren har fire ben, hvoraf de to er til strømforsyning hhv. $V_{cc} = 5V$ og GND . De to andre, V_{in} og V_{out} , anvendes til seriel kommunikation med processoren, jf. [Acr(1999)]. Se også afsnit 5.7.4.

V_{out} kan forbindes direkte til et I/O-ben på processoren, men V_{in} bruger *open-drain input*, hvilket vil sige, at logisk 1 svarer til at der ingen strøm løber fra benet, og at logisk 0 svarer til at der løber en strøm fra benet. Spændingen på dette ben må samtidig ikke overstige 3V, jf. [Sha(2003b)]. Løsningen er at forbinde benet til processoren gennem en diode, hvilket ses i venstre kredsløb på figur 21.

Haves mere end en sensor, behøves kun yderligere et processor-I/O-ben pr. sensor, idet alle sensorernes V_{in} kan deles om det samme I/O-ben på processoren,



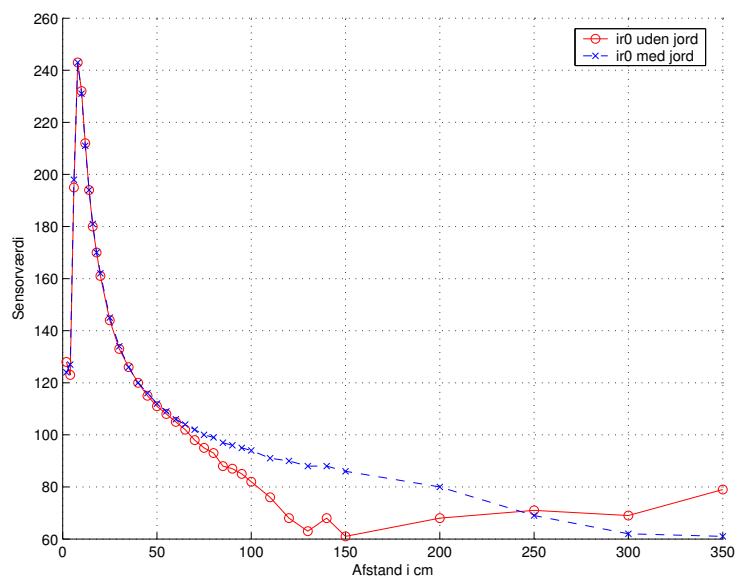
Figur 20: GP2D02-afstandssensor set fra siden med bennumre og benanvendelse.



Figur 21: GP2D02-afstandssensors tilslutning til processor.

hvilket højre kredsløb på figur 21 viser. Dette bevirker dog at hver sensor ikke kan styres individuelt, hvilket kan være en ulempe, hvis man har et begrænset strømbudget. Ligeledes kan sensorerne også interferere med hinanden ved samtidige målinger, hvilket vil blive uddybet i afsnit 5.7.5.

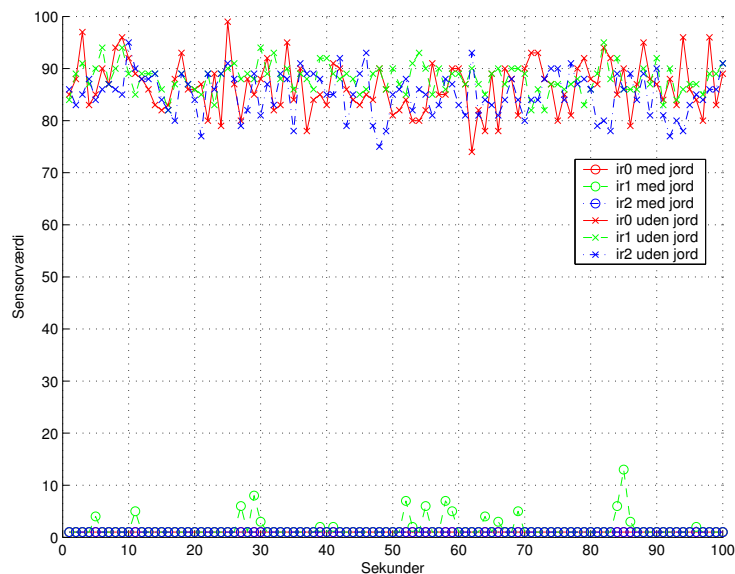
En pudsig egenskab ved sensoren er, at det plastichus, sensorens elektronik, IR-LED og CCD er monteret i, er fremstillet af elektrisk ledende plastic (kan let afprøves med et ohmmeter), jf. [Barello(2003)]. Dette viser sig at være meget vigtigt ved afstandsmålinger på lange afstande, hvilket vil sige over 80 cm. Forbindes sensorens hus til *GND*, giver den meget mere stabile og korrekte målinger ved lange afstande, som det ses på figur 22 og 23. Forbindelsen til jord kan let etableres vha. en jordet kabelsko, som skrues fast på sensoren.



Figur 22: Afstandssensors målinger hhv. med og uden jordet hus. Med jordet hus giver sensoren jævnt faldende værdier, når afstanden stiger, selv ved store afstande, modsat uden jordet hus.

5.7.4 Kommunikation med processor

Selve kommunikationen mellem processoren og sensoren foregår med en master/slave baseret seriel protokol. Se [Sha(2003b)] eller [Acr(1999)] for timingdiagram og protokol. Aflæsningen af en eller flere afstandssensorer kan med fordel implementeres ved et timer-interrupt, som driver timingen af kommunikationen. Jeg har implementeret et sådant program, se cd-rommen under `src/controller/ir_driver.c`.



Figur 23: Afstandssensorernes målinger ved “uendelig” afstand hhv. med og uden jordet hus. Den “uendelige” afstand er simuleret ved at afskærme sensorernes IR-LED’er. Den reelle afstand til nærmeste objekt er over 2 meter, men burde være irrelevant. Jordet hus giver meget stabile og mere korrekte målinger.

5.7.5 Montering på robot

Sensorerne skal monteres på robotten, så de peger i den retning, hvor objekterne, hvortil der ønskes målt afstand, befinder sig. Der findes flere faldgruber ved monteringen af sensorerne, som ikke alle er lige oplagte. Disse vil jeg gennemgå her.

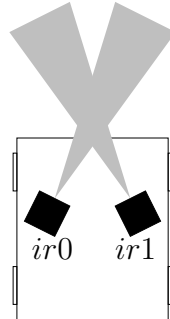
Da afstandssensorerne, jf. afsnit 5.7.2, giver tvetydige resultater ved afstande under ca. 10 cm, er det en fordel at montere dem, så de rent fysisk aldrig kommer ud for en sådan situation, se figur 24.

På grund af sensorernes virkemåde, har det også betydning for afstandsmålingerne, om sensorerne placeres vertikalt eller horisontalt på robotten, se figur 25, og hvor langt fra jorden de er placeret, så IR-LED’ens lyskegle ikke rammer jorden.

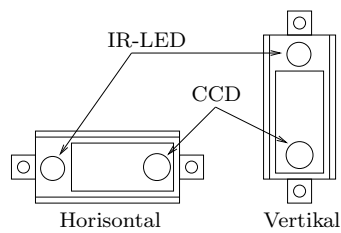
Robotten tænkes som et køretøj, der kører på en forholdsvis plan overflade, “jorden”. En typisk situation er, at der på kørefladen findes forhindringer, som skal undgås, eller at der er en væg, som skal følges. Et specialtilfælde opstår, når IR-LED’ens lyskegle rammer et hjørne af et objekt, så kun en del af lyskeglen reflekteres, som på figur 26.

Umiddelbart kunne man tro, at denne situation blot ville give en afstandsmåling, som er noget længere, end hvis hele lyskeglen ramte objektet. Dette er dog ikke altid tilfældet, jf. [Barello(2003)]. Ved at anvende opstillingen på figur 27, hvor sensoren er orienteret horisontalt, fås måledata på grafen på figur 28.

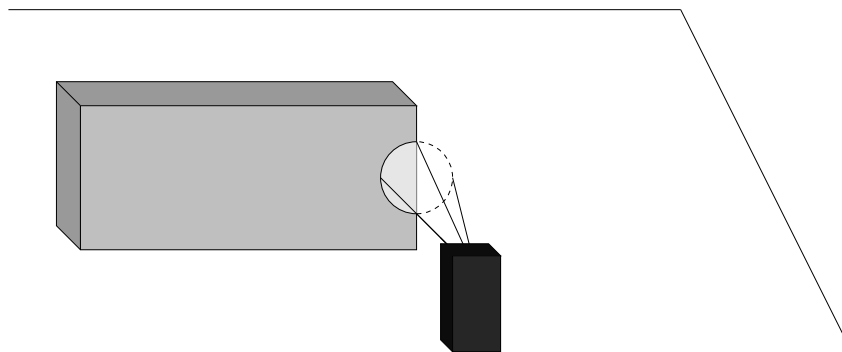
Orienteres sensoren derimod vertikalt, fås måledata på grafen på figur 29.



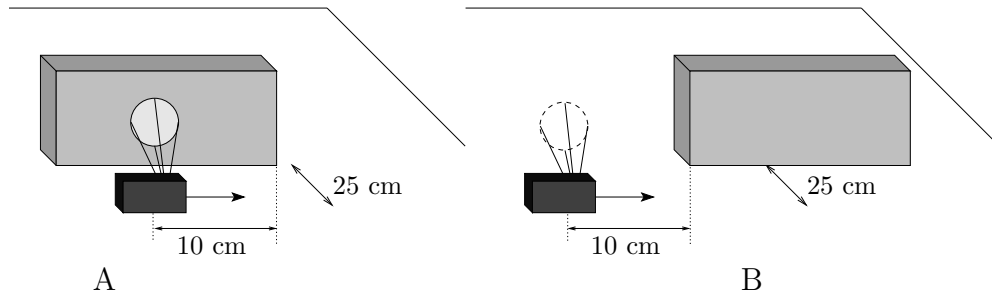
Figur 24: Ved at placere afstandssensorerne mindst 10 cm fra kanten af robotten, kommer objekter aldrig tættere på sensorerne end 10 cm.



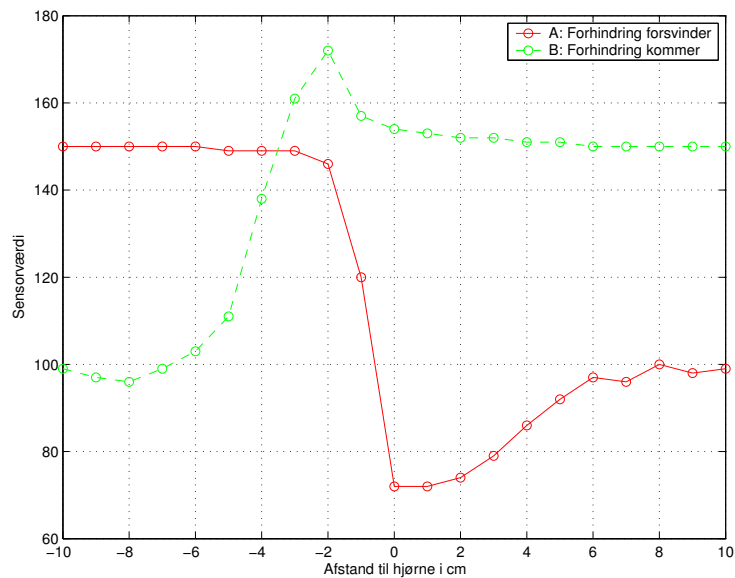
Figur 25: Illustration af, hvad der menes med horisontal og vertikal orientering af afstandssensor, samt placeringen af IR-LED og CCD på sensoren.



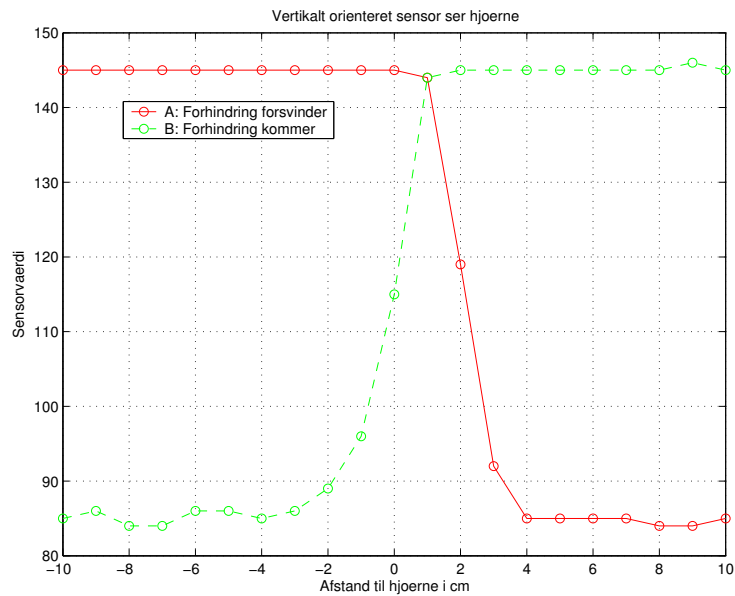
Figur 26: IR-LED'ens lyskegle rammer et hjørne og bliver derfor kun delvist tilbagekastet. Lyskeglediameteren er overdrevet for forståelsens skyld.



Figur 27: Horisontalt monteret sensor ser mod plant objekt 25 cm væk. Objektets hjørne er 10 cm fra sensorens midtpunkt, og sensoren er orienteret, så IR-LED'en er foran CCD'en (dvs. IR-LED'en er til højre for CCD'en på figuren). Der er mere end 150 cm til nærmeste næste objekt. I A starter sensoren foran objektet, og flyttes forbi hjørnet på objektet. I B starter sensoren ved siden af objektet, og flyttes forbi hjørnet på objektet. Lyskeglediameteren er overdrevet for forståelsens skyld.



Figur 28: Graf af målinger fra opstilling som på figur 27. Store værdier svarer til korte afstande og små værdier svarer til lange afstande.



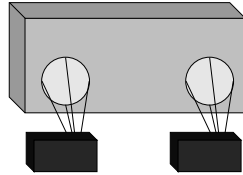
Figur 29: Graf af målinger fra opstilling som på figur 27, men med sensoren *vertikalt* orienteret. Store værdier svarer til korte afstande og små værdier svarer til lange afstande.

Som det ses af figur 29, giver den vertikale orientering de resultater, man umiddelbart ville forvente, mens den horisontale orientering giver *både* kortere afstandsmålinger end hvis objektet havde været helt inde foran sensoren, og længere afstandsmålinger end hvis objektet slet ikke havde været i nærheden af sensoren, som det ses af figur 28.

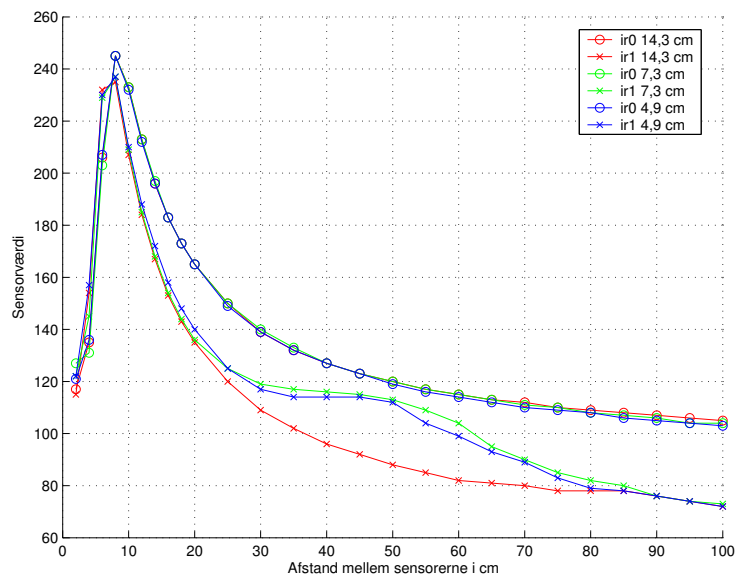
Hvorfor det forholder sig sådan kan jeg ikke forklare, men resultaterne taget i betragtning må det anbefales at montere ir-sensorerne *vertikalt*. Man kan med fordel vende dem, så IR-LED'en er øverst (og CCD'en nederst), da sandsynligheden for at IR-keglen rammer "jorden" så mindskes.

Et andet argument, for at montere sensorerne vertikalt, er at undgå interferens mellem sensorerne. Monteres to sensorer parallelt som vist på figur 30, f.eks. for at finde robotens vinkel i forhold til objektet, vil en situation, hvor hver sensor kan se refleksionen af både sin egen ir-led og den anden sensors IR-LED, let opstå. Dette sker selvfølgelig kun, hvis begge sensorer måler samtidigt.

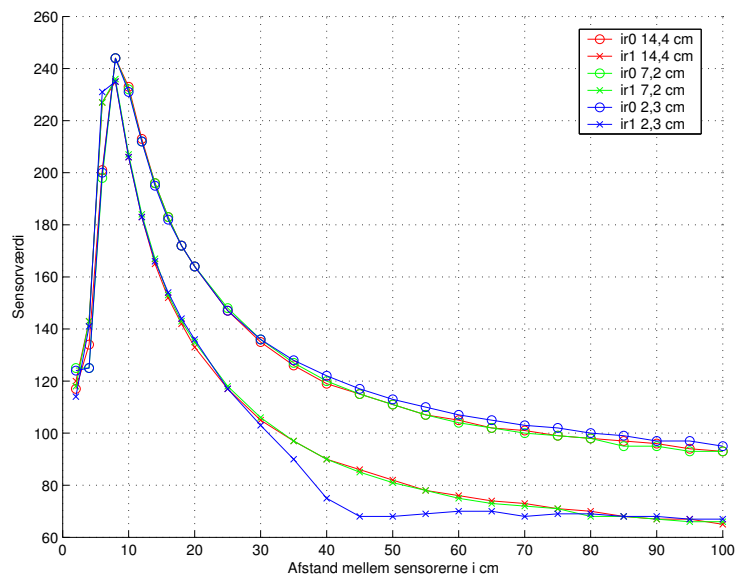
Som forsøgsmålingerne på figur 31 og 32 viser, er interferensen væsentligt større ved horisontal montering end ved vertikal montering. Dette kan forklares ud fra modellen på figur 16. Med horisontalt monterede sensorer kan hver CCD både se sin egen sensors IR-LED og den anden sensors IR-LED. Som det også ses af figur 31, er interferensen størst for den sensor, hvis CCD er tættest på den anden sensors IR-LED.



Figur 30: To horisontale sensorer monteret parallelt. Lyskeglediameteren er overdrevet for forståelsens skyld.



Figur 31: To horisontalt orienterede sensorer som måler i samme retning på samme tid. ir1's CCD er tættest på ir0's IR-LED.



Figur 32: To vertikalt orienterede sensorer som måler i samme retning på samme tid.

5.7.6 Miljøets indflydelse på sensoren

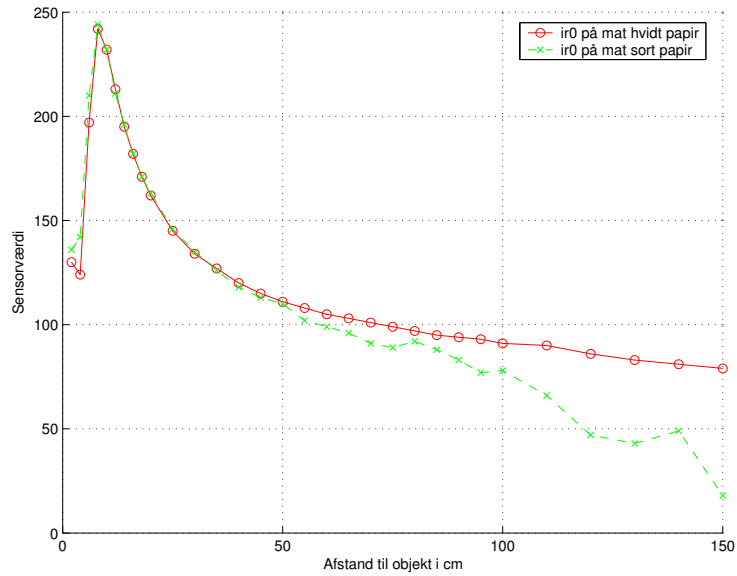
Farve og refleksion Afstandssensoren GP2D02's datablad indledes med at sige, at sensoren er: "*Impervious to color and reflectivity of reflective object*" (Upåvirkelig af farve og refleksivitet af det reflekterende objekt), jf. [Sha(2003b)]. Dette må siges at være lidt af en tilsnigelse. Da sensoren virker ved at måle refleksionen af det IR-lys den udsender, kan den i sagens natur ikke være helt upåvirkelig af objektets farve og refleksivitet, i hvert fald ikke i det infrarøde område af spektret. Målinger på hhv. mat hvidt og mat sort papir, giver graferne på figur 33.

Heraf ses, at mindre refleksion giver mindre værdier end ellers ved store afstande svarende til, at afstandssensoren måler afstanden til at være længere, end den er.

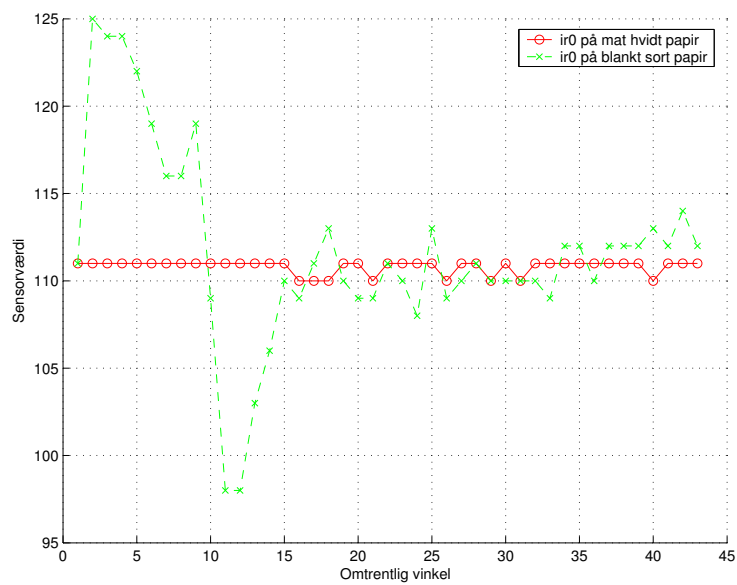
Målinger ved forskellige vinkler på hhv. mat hvidt og blankt sort papir, giver graferne på figur 34.

Læg mærke til, at vinklen stort set ingen betydning har, når objektet er mat og derfor giver en diffus refleksion. Til gengæld har den endog meget stor betydning for afstandsmålingen, når objektet er blankt og derfor giver en højlysrefleksion. Når vinklen er lille, har højlysrefleksionen stor betydning, men betydningen bliver mindre, når vinklen bliver større og sensoren ikke længere kan se højlysrefleksionen. Måles næsten vinkelret på et blankt objekt, giver sensorerne på grund af højlysrefleksionen også værdier med langt større spredning end ved måling på et mat objekt.

Kommer man ud for at skulle anvende afstandssensoren til at måle afstande



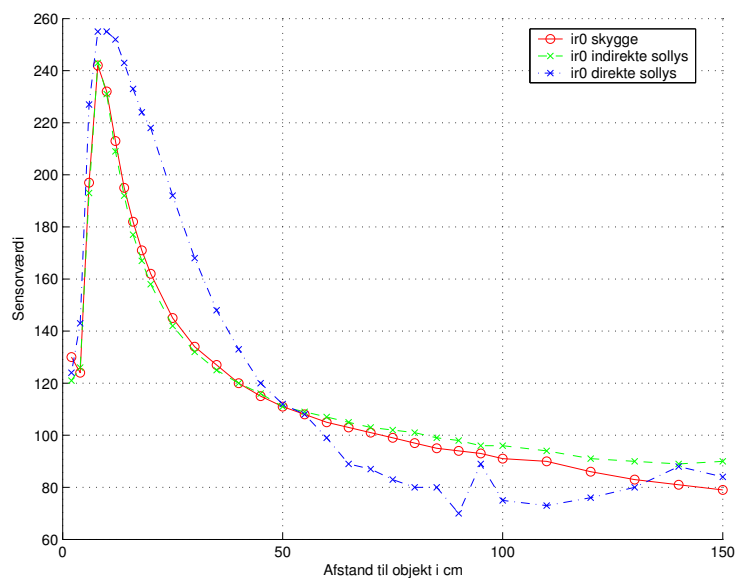
Figur 33: Afstandssensoren ir0's afstandsværdier ved måling på hhv. mat hvidt papir og mat sort papir.



Figur 34: Afstandssensoren ir0's afstandsværdier ved stigende vinkel på hhv. mat hvidt papir og blankt sort papir. Vinklen stiger fra ca. 0 til ca. 40 grader.

til blanke objekter, er det derfor hensigtsmæssigt, hvis det er muligt, at vinkle afstandssensoren, så den ikke peger vinkelret mod objektet. Som det ses af 34, behøver vinklen ikke at være større end ca. 15° .

Belysning Belysningen af objektet, der måles afstand til har ingen betydning for målingerne, så længe den ikke indeholder infrarødt lys. Direkte sollys er en kraftig infrarød lyskilde, og som det ses på figur 35, har direkte sollys også væsentlig betydning for måleværdierne. Jeg har desværre ikke en (infrarød) lysmåler til rådighed, derfor er lysforholdene, som målingerne på figuren er foretaget under, beskrevet i stedet for målt nøjagtigt.

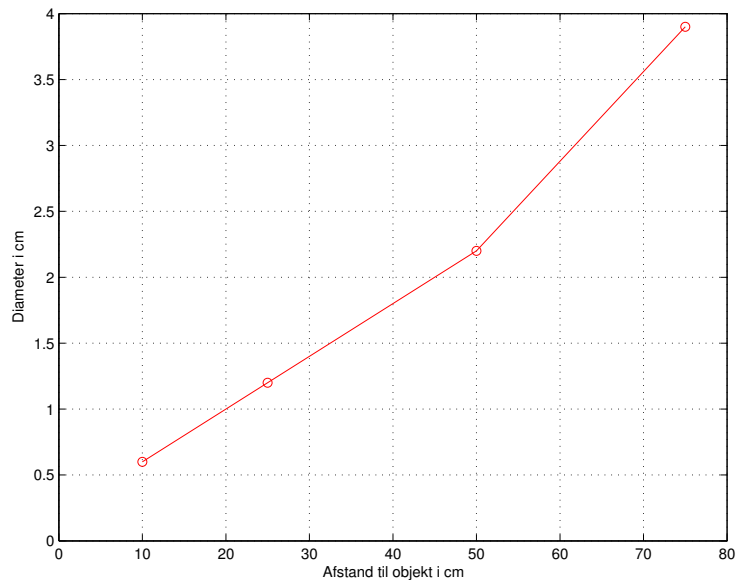


Figur 35: Sensoren måler afstanden til mat hvidt papir i skygge (gardiner for nordvendt vindue i dagslys), indirekte sollys (rummet er oplyst af sollys) og direkte sollys (dog gennem vindue).

En anden kraftig infrarød lyskilde er en almindelig (IR-baseret) tv- eller stereofjernbetjening. En sådan vil typisk give en lysplet meget lig den afstandssensoren selv udsender, og derfor kan interferens fra en fjernbetjening få afstandssensoren til at give en stort set vilkårlig værdi!

Temperatur Ifølge [Sha(2003a)], er sensorerne en smule temperaturfølsomme, men under “normale” forhold, f.eks. under almindelige rumtemperatursvingninger, kan man se bort herfra.

Objektstørrelse Da IR-LED’ens lyskegle bliver større, jo længere afstanden er mellem sensor og objekt, sætter det en begrænsning på, hvor små objekter sensorerne kan måle afstand til. På figur 36 kan ses en måling af IR-lyskeglens størrelse som funktion af afstanden.



Figur 36: IR-lyskeglens diameter som funktion af afstanden.

Målingerne er foretaget ved i mørke at filme et hvidt stykke papir med et videokamera, mens en afstandssensor lyser på papiret. Da de fleste videokameraer er bare lidt følsomme over for infrarødt lys, kan lyskeglen ses på kameraets billede. Fremgangsmåden giver dog en temmelig stor usikkerhed i målingerne.

5.7.7 Eksempler på anvendelse

Typiske eksempler på anvendelse af afstandssensorerne på kørende robotter er kollisionsdetektion og den klassiske “følg væggen”-opgave, som jeg kort vil gennemgå herunder.

“Følg væggen”-opgaven går ud på at få en robot til at køre langs en væg i en passende, gerne konstant, afstand. Jeg vil her kort vise, hvordan “følg væggen”-opgaven let kan klares med en robot med to afstandssensorer og forhjulsstyring.

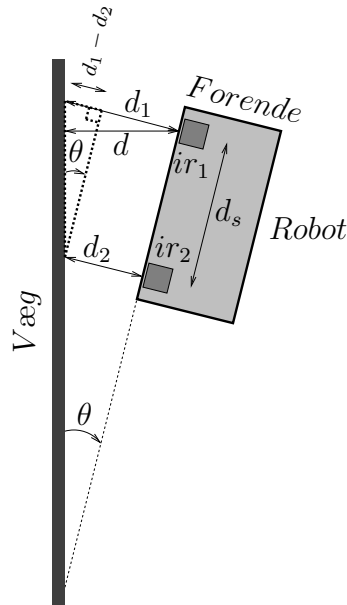
Situationen er opsummeret på figur 37.

Afstanden mellem midten af sensorerne, d_s ³, er kendt, da den er fast og kan måles direkte på robotten. Robottens vinkel i forhold til væggen θ kan udregnes ud fra de to afstandssensors målinger d_1 og d_2 og en trekantsbetragtning:

$$\tan \theta = \frac{d_1 - d_2}{d_s} \Leftrightarrow \theta = \tan^{-1}\left(\frac{d_1 - d_2}{d_s}\right) \quad (20)$$

Måles θ kontinuerligt, og sættes robotens hjulvinkel θ_h ligeledes kontinuerligt til $\theta_h = -\theta$, vil robotten rette sig op parallelt med væggen, når den kører

³Denne d_s har intet at gøre med den d_s som blev brugt i afsnit 5.4.2



Figur 37: Robotten skal følge væggen. Den har monteret to afstandssensorer til den side, hvor væggen den skal følge, befinder sig. Vha. trekantsbetragtning, kan den ønskede hjulvinkel udregnes ud fra afstandssensorernes målinger.

(positiv hjulvinkel θ_h svarer til at robotten drejer til højre). Skal robotten også holde en konstant afstand til væggen, kan θ_h korrigeres, f.eks. proportionalt med differensen mellem den ønskede afstand og den aktuelle afstand d mellem væg og fronten af robotten. d kan findes ved endnu en trekantsbetragtning:

$$\cos \theta = \frac{d}{d_1} \Leftrightarrow d = d_1 \cos \theta \quad (21)$$

Hvis den ønskede afstand mellem robot og væg er d_d , kan hjulvinklen θ_h nu, ved at kombinere formel (20) og (21), kontinuert under kørslen sættes til:

$$\theta_h = c(d_d - d_1 \cos \theta) - \tan^{-1}\left(\frac{d_1 - d_2}{d_s}\right) \quad (22)$$

c er en justeringskonstant, som kan findes empirisk.

5.7.8 Andre afstandssensorer

Ikke alle afstandssensorer fungerer som Sharps. [Acr(2003a)] sælger også afstandssensorer baseret på ultralyd. Den væsentligste forskel er, at de er væsentlig mindre retningsbestemte, og at det er problematisk at anvende flere af dem samtidigt pga. interferens. De er dog formodentlig gode til kollisionsdetektion, da de giver afstanden til det nærmeste objekt inden for en bred vinkel. Andre sensorer er baseret på laser. De er meget retningsbestemte, og kan måle meget

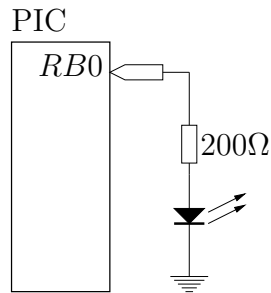
lange afstande. Jeg har dog ingen erfaring med at anvende dem i robotsammenhæng.

6 Aktuatorer

I dette kapitel gennemgås en række aktuatorer, som er lette at anvende sammen med PIC-processorer, og som er oplagte at anvende på kørende robotter.

6.1 Lysdiode

En lysdiode er meget enkel at slutte til en PIC. Den kan tilsluttes direkte til et I/O-ben, da hvert I/O-ben kan trække op til 20mA, hvilket er nok til at drive en lysdiode. Lysdioden forbindes gennem en strømbegrænsende modstand til I/O-benet, som på figur 38.



Figur 38: Tilslutning af lysdiode til processor.

Lysdioder er praktiske at anvende som statusindikatorer, men kan f.eks også anvendes i et tachometer, som nævnt i afsnit 5.5, og til oplysning af objekter som f.eks Sharps IR-afstandssensor.

Lysdioder kan fås i mange forskellige farver, og som før nævnt kan de også udsende infrarødt lys.

6.2 7-segment LED-display

Lysdioder fås indbygget i et hus, hvor 7 rektangulære lysdioder er arrangeret, så de udgør delene af et digitalt tal (som i et digitalur), eventuelt inklusiv et decimalpunktum, som på figur 39. Et eksempel på et sådant *7-segment LED-display* er Vishay TDSR 5160 [Vis(2003)].



Figur 39: Et 7-segment LED-display med decimalpunktum.

Huset kan have 7 eller 8 ind-ben afhængigt af, om det har decimalpunktum eller ej, og et fælles nul-ben. Hvert ind-ben forbindes gennem en strømbegrænsende modstand til et I/O-ben på processoren, som beskrevet i afsnit 6.1.

Vil man spare på antallet af anvendte I/O-ben, kan man eventuelt anvende en eller flere skiftetekredse som 74HCT164 [Phi(2003a)] eller HEF4094B [Phi(1995)] til at drive 7-segment LED-display'et.

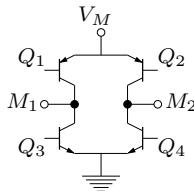
Et 7-segment LED-display, eller flere af sådanne monteret ved siden af hinanden, kan anvendes som display, som kan vise decimal- eller hexadecimaltal.

6.3 DC-motor

En *DC-motor* er en jævnstrømsmotor (DC står for Direct Current). DC-motorer har som regel en høj rotationshastighed og lav kraft (vridningsmoment). Hvis et større vridningsmoment er nødvendigt, nødvendiggør det en nedgearing. En nedgearet DC-motor er et oplagt valg til at drive en kørende robot.

6.3.1 Tilslutning

Når en DC-motor skal tilsluttes en processor, kan det ikke gøres direkte til et I/O-ben, da en DC-motor trækker mere strøm end PIC'ens I/O-ben kan levere — i størrelsesordenen 100mA til 40A, afhængig af motormodel. Processorens signal må forstærkes, og det kan man bruge transistorer til. Transistorerne skal være i stand til at kunne trække den store strømstyrke, og det kan f.eks. *FET* (Field Effect Transistor) transistorer. En smart konstruktion er at bruge 4 FET'er kombineret til en såkaldt *H-bro*, som skitseret på figur 40.



Figur 40: En skitse af en H-bro, bestående af 4 transistorer. En DC-motor kobles til M_1 og M_2

Er H-broens transistorer Q_1, Q_4 åbne og Q_2, Q_3 lukkede, vil strømmen løbe fra M_1 gennem motoren og ud gennem M_2 . Er Q_2, Q_3 derimod åbne, og Q_1, Q_4 lukkede, vil strømmen løbe den modsatte vej fra M_2 til M_1 , og altså vende motorens omløbsretning. Motoren kan desuden kortsluttes ved at åbne Q_1, Q_2 og lukke Q_3, Q_4 (eller omvendt), hvorved motoren virker som en bremse. Lukkes alle transistorerne, kører motoren ikke, men bremser heller ikke. For mere om H-broer, se [Brown(2002)].

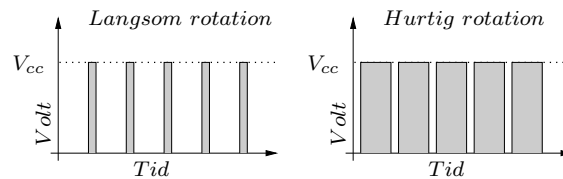
H-broen på figur 40 er kun en skitse, idet DC-motorer har nogle beklagelige egenskaber, som ville kunne brænde transistorerne på figuren af. Både enklere, mere kompakt og mere driftssikkert er det at anvende en integreret H-bro,

som er en H-bro integreret i en IC-kreds, f.eks den integrerede H-bro BA6286 [Roh(2003)]. Kredsen skal vælges, så den er i stand til at klare det strømforbrug, den anvendte motor trækker.

En BA6286 kan trække en $9V$ LEGO-motor (modelnr. 71427). Motoren har indbygget nedgearing (360 rpm ubelastet), og trækker kun max. 360mA, selvom den er meget kraftig, strømforbruget taget i betragtning. En LEGO-motor er meget anvendelig til at drive små robotter, naturligvis specielt hvis robotten er konstrueret i LEGO. For mere information om LEGO-motorens karakteristik, se [Hurbain(2003)].

6.3.2 Pulsbreddemodulering

De fleste integrerede H-broer har to logiske indgange svarende de 4 kombinationer “forlæns”, “baglæns”, “brems” og “frijul”, samt en indgang til *pulsbreddemodulering*. Med pulsbreddemodulering kan motorens omdrejningshastighed tillige styres. Ved høj frekvens, 10 khz eller mere, sendes pulser af varierende bredde fra et I/O-ben på processoren til H-broen, som illustreret på figur 41. H-broen forsyner motoren med strøm under pulsen, og i mellemrummet mellem pulserne lukkes for strømmen til motoren. På grund af den høje moduleringsfrekvens og en DC-motors funktionsmåde, svarer dette til at H-broen regulerer motorens forsyningsspænding.



Figur 41: Pulsbreddemodulering. Korte pulser giver langsom rotation, lange pulser giver hurtig rotation.

Det kan undre, at H-broen ikke direkte regulerer spændingen over motoren, i stedet for at bruge den tilsyneladende mere indviklede pulsbreddemodulering. Det har dog sin gode grund, da det reducerer effekttabet i H-broens transistorer kraftigt. Transistorer har nemlig den egenskab, at der afsættes mindst effekt i dem, når de enten er helt lukkede eller helt åbne, og det bruger ikke en, i denne sammenhæng, væsentlig effekt at skifte tilstand.

Mange af modellerne i PIC-serien har indbygget hardware til pulsbreddemodulering. I software kan moduleringsfrekvensen og pulslængden vælges, så PIC’ens hardware genererer moduleringssignalet på et I/O-ben, se [Mic(1997)] under “Capture/Compare/PWM modules”.

For et mere avanceret eksempel på anvendelse af en DC-motor, se afsnit 8.

6.4 Servomotor

Betegnelsen en *servomotor* dækker over en motor med indbygget regulering og kan dække over alt fra små servomotorer i fjernstyrede biler og modelfly til servomotorer på store industrirobotter. I dette afsnit bruges begrebet om de små servomotorer til fjernstyrede biler.

En servomotor er et firkantet hus, hvorfra der stikker en aksel og en tre-leder ledning (+5V, GND og kontrol) ud. Akslen kan ikke rotere hele vejen rundt, rotationen er begrænset til ca. 180° lidt afhængig af modellen. Inde i huset sidder en DC-motor (se afsnit 6.3), en gearing, et drejepotentiometer (se afsnit 5.6), som måler akslens vinkel, og noget styreelektronik. Positioneringen af akslen styres af servoen: Ud fra signalet på kontrolledningen sørger den indbyggede elektronik for at dreje akslen til den ønskede position, ligesom elektronikken også sørger for at akslen holder positionen, selv hvis en ydre påvirkning forsøger at dreje akslen.

Signalet, som sendes over kontrolledningen, er en puls (5V) med længde proportional med den ønskede vinkel, efterfulgt af en pause (GND). En pulslængde på 1,5 ms svarer til neutral position, 0°. En puls på mindre end 1,5 ms, giver en positiv vinkel (mod uret), en længere puls giver en negativ vinkel (med uret). Den præcise timing afhænger af servomodellen. Pulsen skal gentages hver 20–30 ms. Hvis der ikke sendes en puls til servoen, vil den lade motoren gå i frihjul og ikke aktivt forsøge at holde den nuværende position. For en introduktion til servoers virkemåde, se [McManis(2003)].

Kontrolledningen kan sluttes direkte til et I/O-ben på processoren. Servoens strømforsyning på 5V kan man slutte til samme spændingsregulering, som forsyner processoren, men det kan give problemer, når servoen trækker meget strøm, og batteriet ikke er fuldt opladet. Dette kan afhjælpes med en stor kondensator mellem batteri og spændingsregulator (gerne over 1000 μ F). Bedre er det både at have en separat spændingsregulator til at forsyne servomotoren og en stor kondensator mellem batteriet og spændingsregulatorerne). Denne model fungerede godt på min Robocup-robot [Kjeldsen(2003b)]. Allerbedste løsning er at have separate batterier, så strømforsyningen til processor og logik er helt uafhængig af strømforsyning til servomotorer og også fremdriftsmotorer.

7 Kommunikation

I dette afsnit vil I²C-protokollen kort blive introduceret, og herefter beskrives, hvorledes man bruger PIC'ens indbyggede hardware til kommunikation over RS-232-protokollen.

7.1 I²C

I²C er en seriel synkron master/slave bus protokol, som anvendes til inter-IC kommunikation. PIC'en har indbygget hardware til I²C kommunikation. Protokollen vil ikke blive beskrevet nærmere her, men det skal blot nævnes, at den er oplagt at anvende f.eks. sammen med en ekstern EEPROM IC.

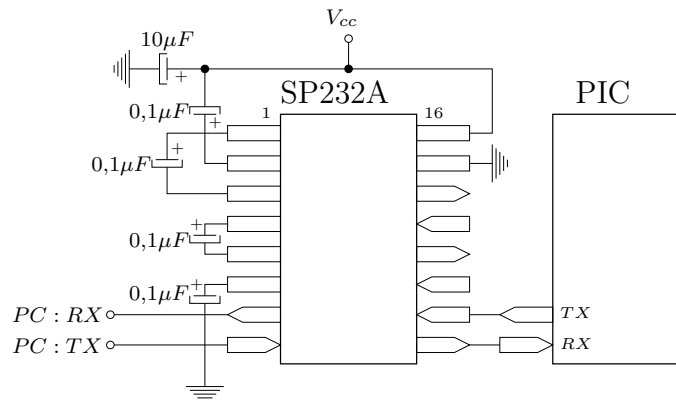
De fleste PC motherboards benytter I²C til kommunikation af f.eks. CPU-temperaturmåling og blæserhastigheder.

For mere information om PIC'ens I²C hardware, se [Mic(2001)].

7.2 RS-232

Kommunikation over RS-232 er meget simpelt, da PIC'en har indbygget USART i hardware. Dog skal PIC'en tilsluttes gennem en *level-shifter*, da PIC'ens logiksignaler er hhv. 5V og 0V, hvor RS-232 protokollens signaler er ca. $\pm 13V$. Protokollen har stor tolerance, fra $\pm 3V$ til $\pm 25V$, jf. [www(2003)].

Den integrerede kreds SP232A fra Sipex [Sip(2000)] er en level-shifter, som kan anvendes sammen med PIC'ens logik. Den tilsluttes som på figur 42.



Figur 42: Tilslutning af RS-232-serielport gennem en level-shifter til PIC'en.

PIC'ens USART behøver ikke nødvendigvis at blive forbundet gennem et *kabel* til pc'ens serielport ej heller til andre PIC'er. Forbindelsen kan også foregå over IR med en infrarød lysdiode og en infrarød modtager, som det gøres på LEGO Mindstorms RCX-klodser, se [LEG(2003b)] og [LEG(2003a)]. Forbindelsen kan også foregå over radio f.eks. via to RadioMetrix BiM-418-F-moduler [van der Veen(2003)] og [Rad(2003)], som det også gøres på Eyebot-robotterne [Eye(2003)].

Har man tænkt sig kun at anvende sit PIC-kredsløb som et mellemlid mellem sensorer/aktuatorer og en større computer som en pda, bærbar eller pc, er det oplagt at lade kommunikationen mellem PIC-kredsløb og computer foregå over RS-232.

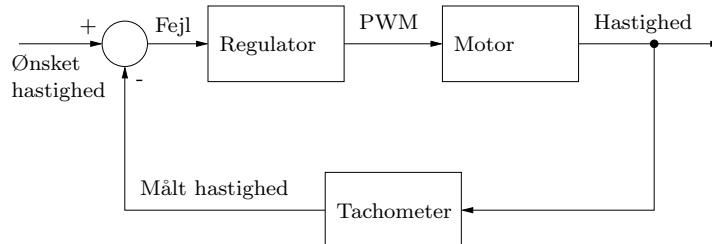
8 Regulering

Reguleringsteknik er en fællesbetegnelse for en række metoder og teknikker, som kan bruges til at regulere fysiske størrelser [Ole Jannerup(2000)].

Da emnet er omfangsrigt, skal her kun præsenteres et enkelt konkret eksempel på anvendelse af reguleringsteknik. For en teoretisk introduktion til reguleringsteknik, se [Ole Jannerup(2000)].

8.1 Hastighedsregulering

Hastigheden af en kørende robot ønskes reguleret, så robotten hurtigst muligt opnår en ønsket hastighed v_d . Robotens fremdriftssystemet består af en regulator (PIC) som pulsbreddemodulerer en motor, der driver robotten. Et tachometer måler den aktuelle hastighed og sender målingen til PIC'en. På figur 43 ses et blokdiagram af systemet.



Figur 43: Blokdiagram over hastighedsreguleringssystem. PWM står for pulsbreddemodulering.

Reguleringsproblemet består nu i at finde en funktion $u(e)$, som ud fra fejlen, dvs. forskellen mellem den ønskede og den målte hastighed $e = v_d - v_m$ giver en pulsbreddemodulering, så robotten opnår den ønskede hastighed.

Her præsenteres uden yderligere teoretisk udredning en god og enkel løsning på reguleringsproblemet, nemlig PID-regulatoren (PID: Proportional, Integral og Differential). u defineres som den vægtede sum af fejlen, integralet af fejlen og den afledede af fejlen:

$$u(e) = k_p e + k_i \int e + k_d \dot{e} \quad (23)$$

Formel (23) kan diskretiseres, så den kan implementeres i en løkke, som jævnligt aflæser tachometerets værdi og opdaterer pulsbreddemoduleringen. j betegner i den diskretiserede formel herunder iterationsnummeret, og Δt betegner længden af tidsrummet mellem hver iteration:

$$u(e(j)) = k_p e(j) + k_i \sum_{x=0}^j e(x) \Delta t + k_d \frac{e(j) - e(j-1)}{\Delta t} \quad (24)$$

Summen af fejlen kan udregnes som en løbende opdateret sum, og behøver ikke udregnes forfra i hver iteration.

Vægtningkonstanterne k_p , k_i og k_d kan findes empirisk. For en forklaring af vægtningkonstanternes betydning for systemets opførsel, se [Uni(1997)], som

også giver en kort introduktion til PID-regulering, ligesom [Mac(1997)], hvorfra formel (23) og (24) stammer.

9 Fordele og ulemper ved selvbyggede kredsløb

Selvbyggede kredsløb har klare oplagte fordele og ulemper. Den helt store fordel er, at de er fleksible: Et kredsløb kan opbygges præcist med den nødvendige funktionalitet, hverken mere eller mindre. Ulemperne er flere: Det tager lang tid at konstruere kredsløbet, og driversoftware skal implementeres til det specifikke kredsløbs ydre enheder. Et håndbygget kredsløb kan sjældent bygges så småt og strømbesparende som et tilsvarende industrifremstillet, og det vil sjældent være lige så robust rent fysisk. Et håndbygget kredsløb og dets software kræver desuden lang tids afprøvning for at man kan sikre sig, at det er stabilt.

Her skal nævnes nogle færdigbyggede robotcontrollere, som man eventuelt kan anvende i stedet for et håndbygget, samt fordele og ulemper ved disse.

9.1 RCX

RCX-klodsen fra LEGO Mindstorms [LEG(2003b)] har et Hitachi-baseret kredsløb, med 64KB RAM, tre sensorindgange, tre aktuatorudgange, IR-kommunikation (bl.a. til overførsel af program), et lille LCD-display, et par knapper og plads til batterier i klodsen. Med LEGO Mindstorms følger både to motorer og tryk- og lyssensorer.

Fordele: *RCX*'en kan programmeres i C [Fre(2003c)], den er ekstremt robust (da den er udviklet til børn), og platform bygges let i LEGO. Ulemper: Antallet af ind- og udgange er stærkt begrænset. Især antallet af indgange er et problem. Indgangene bruger tillige en besværlig protokol, se [Gasperi(1998)], hvilket gør det kompliceret at bygge sine egne sensorer, og der findes ikke så mange typer sensorer, der passer til den.

9.2 Handy Board

Handy Board'et [han(2003)] er en 68HC11-baseret controller, som fås enten samlet eller som byggesæt. Den har 64KB ram og indbygget LCD-display.

Fordele: Man skal ikke selv designe hele sit kredsløb, men kan nøjes med sensor/aktuator-delen. Ulemper: Man skal kunne en del elektronik for at anvende det.

9.3 SV-203

SV203 [Pon(2003)] er en Microchip PIC-baseret servo motor controller. Kommunikation med controlleren foregår over RS-232.

Fordele: Den kan anvendes uden elektronikkunnen, hvis man ikke har brug for digital I/O og analog input. Ulemper: Den kan kun styre servomotorer (eller

kompatible komponenter) og digital output, og kun med de i controlleren indbyggede kommandoer, hvilket giver visse begrænsninger især mht. hastighed og fleksibilitet.

10 Konklusion

I denne rapport har jeg præsenteret, hvordan man anvender en PIC-processor som basis for styreelektronikken til en robot, og nogle muligheder for at tilslutte sensorer og aktuatorer til processoren. Sharps GP2D02 IR-afstandssensor er blevet grundigt gennemgået, jeg har vist anvendeligheden af denne, samt beskrevet de fundne faldgruber ved anvendelsen af den.

Før man går i gang bør man overveje, om man i stedet for at bygge sin egen controller, skal købe en færdigbygget, som passer til formålet. Det kan spare meget tid, og en færdigbygget controller er formodentligt ikke dyrere end en selvkonstrueret. En færdigbygget controller er dog begrænset i sin fleksibilitet, hvor en selvkonstrueret kan bygges til at passe nøjagtigt til den opgave, som skal løses. Som det fremgår af rapporten, består robotbygning af mange små skridt på vejen til en færdig og virkende robot. Robotbygning er måske ikke "rocket science", men tager trods alt rigtigt lang tid. Jeg vil her opfordre til, at man går i gang i god tid og er grundig med at gennemgå sin konstruktion og afprøve det skrevne programmel, da det tager længere tid, end man forestiller sig.

A Kildekode

Kildekoden til de udviklede programmer ligger på den vedlagte cd-rom, i mappen `src`. I dette appendiks er kildekoden til `minimum_example.asm` medtaget, da der i afsnittet 2.4 refereres direkte til en af linierne i dette program:

```

;      Definition af PIC-type:
LIST   P=PIC16F876

;      Definition af konfigurationsord ("fuses"):
        __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_OFF & _HS_OSC & _LVP_OFF

;      Saet radix til decimal (ellers vil tal som default blive opfattet som hexadecimalt)
        RADIX dec

;      Inkluder PIC registerdefinitioner:
#include p16f876.inc

        org 0x20      ; Memory start i bank 0
;      Reserver hukommelse for variable her:
status      res 1
timer1      res 1
timer2      res 1
timer3      res 1

;      Resetvector:
        org 0x0000
        goto      programstart

;      Interruptvector (bruges ikke af dette program)
        org 0x0004

programstart
        bcf      PORTA, 4      ; RA4=0

        bsf      STATUS,RP0      ; Register bank 1
        bcf      TRISA^80h,4      ; RA4: output.
        bcf      STATUS,RP0      ; Register bank 0

        movlw   0xf0
        movwf   status      ; status=0xf0

hop
        swapf   status,F      ; byt om paa mest- og mindstbetydende nibble i status
        btfsc   status,0      ; if(! (status&0x01)) {
        goto    on            ;     goto on;
        bcf     PORTA, 4      ; } else { RA4=1;
        goto    if_end        ;     goto if_end; }
on
        bsf     PORTA, 4      ; RA4=1;
if_end

; Busy-wait i ca. 500ms (ved 14Mhz):
        clrf    timer1      ;timer1=0
        clrf    timer2      ;timer2=0
        movlw   8
        movwf   timer3      ;timer3=8

busywait_loop
        decfsz  timer1,F      ;if(--timer1!=0)
        goto    busywait_loop ;     goto busywait_loop;
        decfsz  timer2,F      ;if(--timer2!=0)
        goto    busywait_loop ;     goto busywait_loop;
        decfsz  timer3,F      ;if(--timer3!=0)
        goto    busywait_loop ;     goto busywait_loop;

        goto   hop

        end                ; End of program

```

B Indkøbssteder

I dette appendiks er listet en række indkøbssteder, hvor man kan købe robot-hardware og elektronik.

B.1 Acroname Inc.

Adresse: www.acroname.com

- Sharp GP2D02-afstandssensor. Ca. \$20 pr. stk. + forsendelse.
- Mange robot-relaterede komponenter til hobbybrug.

B.2 Brinck Elektronik

Adresse: www.brinck.dk

- LM7805 spændingsregulator.
- NiMH batterier.
- SP232A level shifter.
- 74HCT164 skifteregister.
- HEF4094B skifteregister.
- 7-segment LED-display.
- Servomotor.
- Diverse elektronik.

B.3 Koz Electronics

Adresse: www.koz.dk

- Atmel AVR Atmega16

B.4 LEGO

Adresse: www.lego.dk

- LEGO.

B.5 Microchip

Adresse: www.microchip.com

- PIC indlejrede processorer.

Microchip har en “samples”-service (vareprøveservice), hvor man *gratis* kan få tilsendt op til fem produkter. Adresse: sample.microchip.com.

B.6 RS-components

Adresse: www.rs-components.dk

- Sharp GP2D02-afstandssensor.
- Diverse elektronik.

NB! RS-components sælger ikke til privatpersoner.

Litteratur

- [Acr(1999)] *The Sharp GP2D02 and GP2D05 Infrared Object Detectors*. Acroname Inc., 1999. Lille hæfte, som følger med GP2D02, når den købes hos Acroname.
- [Acr(2003a)] *Acroname Inc for Easier Robotics*. Acroname Inc., 2003a. URL <http://www.acroname.com>.
- [Acr(2003b)] *Sharp GP2D02 Infrared Ranger*. Acroname Inc., 2003b. URL <http://www.acroname.com/robotics/parts/R19-IR02.html>. CDROM: web/R19-IR02.html.
- [Atm(2003)] *Atmel Corporation*. Atmel Corporation, 2003. URL <http://www.atmel.com/>.
- [Barello(2003)] L. Barello. *Interfacing with GP2D02 sensors*, 2003. URL <http://www.barello.net/Papers/GP2D02/>. CDROM: web/GP2D02.html.
- [Bri(2003a)] *Br 873 - PIC-demonstration board*. Brinck Elektronik, 2003a. URL <http://www.brinck.dk/byggesaet/br873.asp>.
- [Bri(2003b)] *Brinck Elektronik*. Brinck Elektronik, 2003b. URL <http://www.brinck.dk/>.
- [Brown(2002)] J. Brown. *Brief H-Bridge Theory of Operation*, 2002. URL <http://www.dprg.org/tutorials/1998-04a/>.
- [CCS(2003)] *CCS, INC*. CCS, 2003. URL <http://www.ccsinfo.com/>.
- [Dan(2003)] *DTU RoboCup: Roboternes tumleplads*. Danmarks Tekniske Universitet, 2003. URL <http://www.robocup.dtu.dk/>.
- [Eye(2003)] *Eyebot*. Eyebot, 2003. URL <http://www.ee.uwa.edu.au/~braunl/eyebot/>.
- [Fai(2001)] *Fairchild MC78XX/LM78XX/MC78XXA 3-Terminal 1A Positive Voltage Regulator*. Fairchild, 2001. URL <http://www.fairchildsemi.com/pf/LM/LM7805.html>. CDROM: web/LM7805.pdf.
- [Fai(2003)] *Fairchild H21A2*. Fairchild, 2003. URL <http://www.fairchildsemi.com/pf/H2/H21A2.html>. CDROM: web/H21A2.pdf.
- [Fre(2003a)] *GCC Home Page*. Free Software Foundation, 2003a. URL <http://gcc.gnu.org/>.
- [Fre(2003b)] *GNU PIC Utilities*. Free Software Foundation, 2003b. URL <http://gputils.sourceforge.net/>.
- [Fre(2003c)] *Welcome to the brickOS home page*. Free Software Foundation, 2003c. URL <http://brickos.sourceforge.net/>.
- [Gasperi(1998)] M. Gasperi. *MindStorms RCX Sensor Input Page*, 1998. URL <http://www.plazaeearth.com/usr/gasperi/lego.htm>.

- [Gijzen(2003)] B. Gijzen. *IC-Prog Prototype Programmer*, 2003. URL <http://www.ic-prog.com/index1.htm>.
- [han(2003)] *The Handy Board*. handyboard.com, 2003. URL <http://handyboard.com/>.
- [Hi-(2003)] *Hi-Tech Software*. Hi-Tech, 2003. URL <http://www.htsoft.com/>.
- [Hurbain(2003)] P. Hurbain. *Lego 9V Technic Motors compared characteristics*, 2003. URL <http://www.philohome.com/motors/motorcomp.htm>.
CDROM: web/motorcomp.htm.
- [Hyvätti(2003)] J. Hyvätti. *Picprog 1.3 documentation*, 2003. URL <http://hyvatti.iki.fi/~jaakko/pic/picprog.html>.
- [Int(1988)] *Intel Hexadecimal Object File Format Specification*. Intel, 1988. URL http://www.cse.unsw.edu.au/~pleb/data_sheets/Hexfrmt.pdf.
CDROM: web/Hexfrmt.pdf.
- [Jones(1993)] M. H. Jones. *A practical introduction to electronic circuits*, second edition, 1993.
- [Kjeldsen(2003a)] R. F. Kjeldsen. *Bootloader*, 2003a. URL <http://reblag.dk/robot/bootloader/>.
CDROM: src/bootloader/.
- [Kjeldsen(2003b)] R. F. Kjeldsen. *A PIC-based LEGO controller*, 2003b. URL <http://www.reblag.dk/robot>.
- [LEG(2003a)] *LEGO - Play on*. LEGO, 2003a. URL <http://www.lego.com>.
- [LEG(2003b)] *LEGO Mindstorms*. LEGO, 2003b. URL <http://mindstorms.lego.com/eng/default.asp>.
- [Lin(2002)] *Programming the AVR Microcontroller with GCC*. LinuxFocus.org, 2002. URL <http://www.linuxfocus.org/English/March2002/article231.shtml>.
CDROM: web/article231.shtml.
- [Mac(1997)] *Digital Control Systems*. Macquarie University, 1997. URL <http://www.comp.mq.edu.au/courses/comp324/cn4/cn4.html>.
- [Madsen(2000)] J. D. Madsen. *PIC-Programmer 2 for PIC16C84 etc.*, 2000. URL <http://www.jdm.homepage.dk/newpic.htm>.
- [McManis(2003)] C. McManis. *Servo-Motor 101*, 2003. URL http://www.repairfaq.org/filipg/RC/F_Servo101.html.
CDROM: web/F_Servo101.html.
- [Mic(1997)] *PICmicro Mid-Range MCU Family Reference Manual*. Microchip, 1997. URL <http://www.microchip.com/1010/suppdoc/refernce/midrange/786/index.htm>.
CDROM: web/33023a.pdf.
- [Mic(2001)] *Microchip PIC16F87X Data Sheet*. Microchip, 2001. URL <http://www.microchip.com/download/lit/pline/picmicro/families/16f87x/30292c.pdf>.
CDROM: web/30292c.pdf.

- [Mic(2002)] *Microchip PIC16F87X EEPROM Memory Programming Specification*. Microchip, 2002. URL <http://www.microchip.com/download/lit/suppdoc/specs/39025f.pdf>. CDROM: web/39025f.pdf.
- [Mic(2003)] *Microchip website*. Microchip, 2003. URL <http://www.microchip.com/>.
- [Ole Jannerup(2000)] P. H. S. Ole Jannerup. *Introduktion til reguleringsteknik*, 2000.
- [Phi(1995)] *8-Stage Shift-and-Store Bus Register*. Philips, 1995. URL <http://www.philipslogic.com/products/shiftregisters/4094/>. CDROM: web/hef4094b.pdf.
- [Phi(2003a)] *8-bit serial-in/parallel-out shift register*. Philips, 2003a. URL <http://www.semiconductors.philips.com/pip/74HCT164.html>. CDROM: web/74HC_HCT164.CNV_2.pdf.
- [Phi(2003b)] *HEF4000 series*. Philips, 2003b. URL <http://www.philipslogic.com/products/hef/>.
- [Pon(2003)] *SV203 Servo Motor Controller*. Pontech, 2003. URL <http://www.pontech.com/products/sv200/>.
- [Rad(2003)] *Engineers preferred choice for UHF & VHF Low Power Radio modules*. Radiometrix, 2003. URL <http://www.radiometrix.co.uk/>.
- [Roh(2003)] *BA6286*. Rohm, 2003. URL <http://www.rohm.com/products/databook/motor/pdf/ba6286.pdf>. CDROM: web/ba6286.pdf.
- [Sha(2003a)] *Application Note: Distance Measuring Sensors*. Sharp, 2003a. URL http://www.sharpsma.com/sma/Products/Opto/Osd/General_DMS_AN.pdf. CDROM: web/General_DMS_AN.pdf.
- [Sha(2003b)] *Sharp GP2D02*. Sharp, 2003b. URL http://www.sharp.co.jp/products/device/ctlg/jsite23/table/pdf/osd/optical_sd/gp2d02_j.pdf. CDROM: web/gp2d02-j.pdf.
- [Sha(2003c)] *Sharp MSA*. Sharp, 2003c. URL http://www.sharpsma.com/sma/Products/Prod-frame/opto.htm?main=http://www.sharpsma.com/sma/products/opto/optical_systems.htm.
- [Sip(2000)] *Sipex SP231A/232A/233A/310A/312A Enhanced RS-232 Line Drivers/Receivers*. Sipex, 2000. URL http://www.sipex.com/products/pdf/SP232_312A.pdf. CDROM: web/SP232_312A.pdf.
- [Uni(1997)] *PID Tutorial*. University of Michigan, 1997. URL <http://www.engin.umich.edu/group/ctm/PID/PID.html>. CDROM: web/PID.html.
- [van der Veen(2003)] J. S. van der Veen. *ELMO autonomous robot with radio communication*, 2003. URL <http://www.connectionlab.com/elmo/>.
- [Vis(2003)] *Standard 7-Segment Display 13 mm*. Vishay Semiconductors, 2003. URL <http://www.vishay.com/document/83126/83126.pdf>. CDROM: 83126.pdf.

- [Wieman(2003)] S. Wieman. *Intel Hex Format*, 2003. URL http://margo.student.utwente.nl/el/misc/intl_hex.htm. CDROM:
web/intl_hex.htm.
- [www(2003)] *How Serial Ports Work*. www.howstuffworks.com, 2003. URL
<http://computer.howstuffworks.com/serial-port.htm>.