

Prøvetrykk 3.17

Nils Kr. Rossing

Jøran Grande

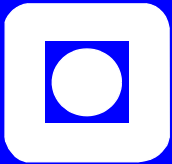
Thomas Gansmoe

Jens F. Dalsgaard Nielsen

Miljø- og Romteknologi (CanSat) – m/sensorteknologi



NTNU



Trondheim

Program for
lærerutdanning

Skolelaboratoriet
for matematikk, naturfag
og teknologi

August 2016

Miljø- og Romteknologi (CanSat) – m/sensorteknologi

Miljø- og Romteknologi (CanSat) – m/sensorteknologi

Trondheim 2016

Bidragstere:

Nils Kr. Rossing, (nils.rossing@plu.ntnu.no) Skolelaboratoriet ved NTNU

Jøran Grande (joran@rocketrange.no), NAROM

Thomas Gansmoe (thomasg@rocketrange.no), NAROM

Jens Fredrik Dalsgaard Nielsen (jdn@es.aau.dk), Aalborg universitet

Layout og redigering: Nils Kr. Rossing, Skolelaboratoriet ved NTNU

Tekst og bilder Nils Kr. Rossing, Skolelaboratoriet ved NTNU

Faglige spørsmål rettes til:

Skolelaboratoriet for matematikk naturfag og teknologi, NTNU

v/Nils Kr. Rossing, 73 55 11 91

nils.rossing@plu.ntnu.no

Realfagbygget, Høgskoleringen 5
7491 Trondheim

Skolelaboratoriet

Telefon: 73 55 11 43

Telefaks: 73 55 11 40

<http://www.ntnu.no/skolelab/>

Prøvetrykk 3.17, Rev 3.17 – 29.08.16

Siste utgave av boka kan dessuten lastes ned fra:

<http://www.ntnu.no/skolelab/sl-bla-bokserie>

Miljø- og Romteknologi, CanSat – m/sensorteknologi

Nils Kr. Rossing

Jøran Grande, Thomas Gansmoe, Jens F. Dalsgaard Nielsen

Skolelaboratoriet for matematikk, naturfag og teknologi, NTNU





Forord

Hensikten med dette heftet er å samle erfaringer fra utprøving av CanSat-kittet benyttet ved kurset Miljø- og Romteknologi (CanSat) – høsten 2012, samt å beskrive ulike sensorer som kan vurderes anvendt sammen med CanSat'en.

Beskrivelsen av sensorer strekker seg imidlertid ut over anvendelser i CanSat og faget Teknologi og forskningslære. Vi har valgt en noe bredere tilnærming til feltet siden sensorer brukes i mange ulike sammenhenger og ikke bare innen romteknologi. Heftet kan derfor også brukes som tilleggsstoff til Fysikk 1 i videregående skole. Høsten 2012 er alle ungdomsskoler pålagt å tilby valgfag på åttende trinn. To av de åtte beskrevne valgfagene er Teknologi i Praksis og Forskning i Praksis. Kurset Miljø- og Romteknologi tilbys derfor til lærere i ungdomsskolen som skal tilby ett eller begge de nevnte valgfagene.

Det er forøvrig ikke meningen at heftet skal erstatte eksisterende CanSat håndbok eller Jens Dalsgaard Niensens nettsted: <http://www.control.aau.dk/~jdn/edu/cansat-kit/index.html> ved Universitetet i Aalborg, men være et supplement.

Vi vil videre rette en takk til Jens F. Dalsgaard Nielsen (jdn@es.aau.dk) som tok initiativet til å utvikle et nytt CanSat og Simon Jensen (simon@es.aau.dk) som har lagt ut kretskortet (Arduino shield).

7. mai 2013 ble det i Trondheim arrangert et slipp fra ballong med elever fra Heimdal og Strinda videregående skoler. Skolelaboratoriet ved NTNU sto for arrangementet, med støtte av **Frode Øren**, **Ole Petter Håkegaard** og **Einar Solstad**. Dette var en nyttig erfaring til tross for at ikke alt gikk som planlagt, det var litt for mye vind og litt for lite helium i ballongen. Likevel ble det gjennomført 10 slipp. Slippet ble gjentatt våren 2014 med elever fra Strinda videregående skole. Denne gangen gikk det noe bedre da vinden ikke var så sterk og vi hadde nok helium. Imidlertid slet ballongen seg etter tredje slipp slik at vi mistet ballongen med slippmekanismen. Ny slippmekanisme ble snekret samme kveld og returnert til NAROM dagen etter. I 2015 var hovedansvaret for slippet fra ballong gitt til Heimdal vgs ved **Carina Tiday**, som forvaltet ansvaret på en meget god måte. Været sto oss også be slik at det på alle måter ble et vellykket slipp. I tillegg til Heimdal stilte Strinda med elevgrupper med CanSat til slippet.

Utgave 3.17 er supplert med informasjon om innsamling og behandling av GPS-data. dette stoffet er dels bygget på det **Ole Petter Håkegaard** tidligere har presentert.

Nils Kr. Rossing

Skolelaboratoriet ved NTNU
August 2016





Innhold

1 Innledning	17
2 Montering og installasjon	19
2.1 Kort beskrivelse av systemets deler	19
2.1.1 Databehandlingsenhet	19
2.1.2 Datainnsamlingsenhet	22
2.1.3 Kommunikasjonsenhet	28
2.1.4 Datalagringsenhet	33
2.1.5 Kraftforsyningsenhet	34
2.2 Montering	35
2.2.1 Montering av datainnsamlingsenheten	35
2.2.2 Montering i brusboks (Can)	43
2.3 Antennen	45
2.4 Uttesting av hårdvaren	46
2.4.1 Innledende uttesting av sensorene på shield-kortet	46
2.4.2 Kalibrering av sensorene	47
2.4.3 Test av seriell datalinje	50
2.4.4 Test av radiokommunikasjon	50
2.4.5 Test av datalagringsenheten	51
2.5 Lagring av data på file (terminalprogrammer)	51
2.5.1 Terminal	52
2.5.2 Tera Term VT	53
3 Oppskyting	57
3.1 Ballongslipp	57
3.1.1 Utstyr for ballongslipp	57
3.1.2 Slippmekanisme	57
3.1.3 Reglement for oppsending av forankret ballong	60
3.1.4 Termineringsmekanisme for forankrede ballonger	61
3.2 Bygg en egen slippmekanisme	64
3.3 Beregning av fallhastighet	68
3.4 Simulering av CanSat dropp for å finne fallhastighet	71
4 Programmering	77
4.1 Installasjon av programvare	77
4.1.1 Arduino programeditor	77



4.2	Programstruktur	80
4.3	Viktige kommandoer	80
4.3.1	Generelle kommandoer	80
4.3.2	Avlesning av sensorer	83
4.3.3	Programmering av kommunikasjonsenheden APC220	85
4.3.4	Skriv data til Openlog	85
4.3.5	Lagring av data fra GPS-modul GY-NEO6MV2	89
5	Analyse av data	95
5.1	Skrive data til file	95
5.1.1	Lagre rådata	95
5.1.2	Signatur	95
5.1.3	Tidsangivelse	95
5.1.4	Skilletegn	96
5.1.5	Den endelige datafilen	96
5.2	Importer data til Excel	96
5.3	Klargjøring av data	99
5.4	Legg inn formler	100
5.4.1	Trykk	100
5.4.2	Høyde	101
5.4.3	Temperatur	101
5.4.4	CO ₂	102
5.4.5	Akselerasjon	102
5.5	Lage grafer	102
5.6	Visualisering av GPS-data i Google Earth	104
5.6.1	Installasjon av Google Earth	104
5.6.2	Plotting av en trase i Googel Earth	105
5.6.3	Editering av kml-fila	108
6	Noen utvalgte systemkomponenter	111
6.1	Operasjonsforsterkeren	111
6.1.1	Prinsipiell virkemåte og sentrale parametere for operasjonsforsterkeren	113
6.1.2	Praktiske oppkoblinger	115
6.1.3	Forsterker med enkel spenningsforskyning	117
6.1.4	Komparatoren	118
6.1.5	Differensialforsterkeren	118



6.1.6	Subtraksjons- og summasjonsforsterkere	120
6.2	AD-konverteren - fra analoge til digitale signaler	120
6.2.1	Sampling	120
6.2.2	AD-konverteren	122
7	Sensorer	125
7.1	Kontrollkortets og sensorkortets grensesnitt (CanSat)	125
7.1.1	Analogt grensesnitt:	125
7.2	Innledende betraktninger om sensorer	127
7.2.1	Hva er en sensor?	127
7.2.2	Egenskaper ved sensorer	129
7.3	Temperaturfølsomme sensorer	131
7.3.1	Temperaturfølsom motstand (NTC og PTC -motstander)	131
7.3.2	Integrert temperatursensor LM35DZ	134
7.3.3	IR-sensor	136
7.4	Lysfølsomme sensorer	138
7.4.1	Fotomotstand (LDR - Light Dependent Resistor)	138
7.4.2	Fototransistorer	140
7.4.3	Kameramoduler	144
7.5	Lufttrykkfølsomme sensorer	145
7.5.1	Måling av lufttrykk	146
7.5.2	Trykksensorene MPX4100A og MPX4115 (Motorola)	146
7.5.3	Måling av høyde basert på trykkmålinger	148
7.5.4	Barometeret BMP085 (Bosch)	150
7.6	Fuktighetsfølsomme sensorer	151
7.6.1	Måling av fuktighet	151
7.6.2	Fuktighetssensoren H25K5A	154
7.6.3	Fuktighetssensoren HiH4000 Honeywell	156
7.7	Bevegelsesensorer	157
7.7.1	Akselerometer	157
7.7.2	Bevegelsessensor	160
7.8	Avstandsensorer	161
7.8.1	Ultrasonisk avstandsmåler - SRF005	161
7.9	Magnetfeltfølsomme sensorer	162
7.9.1	Hovedinndeling av magnetfeltsensorer.	163
7.9.2	Hallsensorer	165



7.9.3	Magneto-resistivitet	166
7.9.4	Elektronisk kompass – CMPS03	167
7.9.5	Anvendelser av magnetometer	168
7.10	Gass-sensorer	168
7.10.1	Pellistoren	169
7.10.2	Gassensor LPG, Metan – HS-133	172
7.10.3	CO ₂ innholdet i atmosfæren	172
7.10.4	Gass-sensor CO ₂ – MQ-135	173
7.10.5	Gassensor CO ₂ – MG811	173
7.10.6	Måling av CO ₂ ved absorpsjon av IR-stråling (C100)	175
7.10.7	Måling av ozon-konsentrasjon	177
7.11	Partikkelsensorer	179
7.11.1	Røykdetektor (røyk, varme)	179
7.11.2	Støvdetektorer (elektrodynamisk teknikk)	182
7.12	GPS-modul	182
7.12.1	EM-411 (GPS-modul)	183
7.12.2	GY-NEO6MV2 Flight (GPS-modul)	184
8	Laboratorieoppgaver	187
8.1	Laboratorium 1 – Datainnsamling med CanSat	187
8.1.1	Del 1 - Måling av trykk og temperatur som funksjon av høyden – «Primary mission»	187
8.1.2	Del 2 - Måling med selvvalgt sensor som funksjon av høyden – «Secondary mission»	188
8.2	Laboratorium 2 – Fallhastighet	193
8.3	Oppkobling av ekstra sensor for uttesting	195
8.4	Karakterisering av lyssensor.	196
8.5	Nødvendig laboratorieutstyr	198
8.6	Generelle retningslinjer for skriving av laboratorierapport	200
9	Referanser	203
Vedlegg A	Eksempler på kode	205
A.1	Oppretting og skriving av data til Openlog	205
A.2	Konfigureringsprogram APC220	207
A.3	Program for styring av terminering (BallongSquid.ino) Skrevet av Thomas Gansmoe, NAROM)	214
A.4	Kode for lesing fra GPS-modul	216
Vedlegg B	Regler for Forankrede ballonger	223



Vedlegg C	Leverandører	224
C.1	Oversikt over komponenter til CanSat kit'et	224
C.2	Leverandører av ulike sensorer og andre komponenter	226
Vedlegg D	Produktdata	228
D.1	APC220	228
Vedlegg E	Fargekoding av motstander	231
Vedlegg F	Læreplaner	232
F.1	Teknologi og Forskningslære 1	232
F.2	Teknologi og forskningslære 2	233
F.3	Teknologi i Praksis (TiP) - ungdomsskolen	234
F.4	Forskning i Praksis (FiP) - Ungdomsskolen	235





Revisjonsrapport

Revisjonsrapporten gir oversikt over revisjoner i de kapitlene som er pensum. Forsøker å holde dette på et minimum. De øvrige kapitlene suppleres etter behov slik at heftet skal bli mest mulig anvendelig for bruk dersom CanSat skal gjennomføres som prosjekt i klasserommet:

Rev. 3.5 Avsnitt 7.7.1 på side 157

Avsnittet er supplert med utdypende forklaring til "roll", "pitch" og "yaw" knyttet til CanSat og akselerometerets spesifikasjoner

Rev. 3.7 Avsnitt 7.10.6 på side 175

Avsnittet er supplert med diagrammer og informasjon om bruk av CO₂ sensoren C100.

Rev. 3.8 Avsnitt 7.7.1 på side 157

*Rettet feil vedrørende g-verdi ved 1,65 V (denne er 0 g og ikke 1 g)
Ligning 7.19 er lagt inn.*

Rev. 3.8 Avsnitt 7.6.3 på side 156

*Lagt til omregningsformler for relativ fuktighet for HiH4000
Ligningene 7.16 til 7.18 er lagt inn.*

Rev. 3.8 Avsnitt 2.1.1 på side 19

Oppdatert pris på Arduino UNO R3.

Rev. 3.9 Avsnitt 2.1.3 på side 28

*Ny versjon av programmet for konfigurering av APC220 (Alternativ 1)
Ny programlisting i Avsnitt A.2 på side 207.*

Rev 3.10 Avsnitt 3.1 på side 57

Omtale av ballongslipp våren 2013 i Trondheim og tilføyelse av vedlegg B som omtaler regler for forankrede ballonger.

Rev 3.11 Noen mindre feil er rettet opp

Rev. 3.12 Vedlegg på side 229

Oversikt over omregningsformler for sensorer som er standard på shield kortet.

Rev. 3.13 Avsnitt 2.4.1 på side 46 og Avsnitt 2.4.2 på side 47

Det er tilføyd et avsnitt om innledende målinger på de analoge inngangene (se avsnitt 2.4.1) Vedlegg E er flyttet til avsnitt 2.4.2 og kalt: Kalibrering av sensorer

Rev. 3.14 Avsnitt 7.5.4 på side 150, Avsnitt 7.8.1 på side 161 og Avsnitt 2.5.1 på side 52

Det lagt inn omtale noen nye sensorer og heftet er korrekturelest mht. ortografiske feil. Et nytt terminalprogram (Terminal) er lagt til.

Rev. 3.15 Avsnitt 2.5.1 på side 52

Et nytt terminalprogram (Terminal) er lagt til.



Rev. 3.16 avsnitt 2.4.2 på side 45 og 46,

Gjort noen rettinger i kalibreringsrutinene for akselerometeret.

Beskrevet bygging av slippmekanisme

Skrevet en introduksjon for analyse av data.

Rev. 3.17 Avsnitt 4.3.5 på side 89 – Hent data fra GPS-modul og skriv til SD-kort

Avsnitt 5.6 på side 104 – Presentasjon av innsamlede GPS-data i Google Earth

Avsnitt 7.12 på side 182 – Spesifikasjoner og oppkobling av GPS-modul



1 Innledning

Heftet er en samling og systematisering av grunnleggende erfaringer gjort under arbeidet med å prøve ut grunnmodulen av CanSat byggesettet utviklet av Jens Dalsgaard Nielsen og miljøet han er tilsluttet ved Aalborg Universitet. Byggesettet er lagt opp ved NAROM, Andøya for bruk på lærerkurs og til undervisningsformål i ungdomsskolen og videregående skole. Slik byggesettet leveres i dag inneholder det:

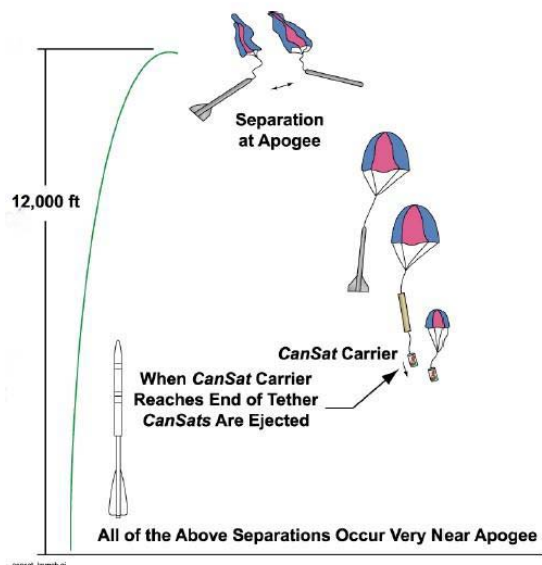
1. Byggesettet:
 - Monteringsbraketter med skruer, muttere, øyeskrue
 - Arduino UNO
 - Sensor Shield kort
 - Akselerator 3D
 - Temperatursensor med $1\mu\text{F}$ kondensator og $75\ \Omega$ motstand
 - Trykksensor
 - Strapper, kontaktlister, sokler for kontaktlister
 - 2 stk Sender/mottaker m/antenner
 - 1 stk USB-adapter
2. Utstyr for programmering
 - USB-kabel
 - Programvare fra nettet
3. Fallskjerm

Ved hjelp av byggesettet skal det bygges en liten sonde som skal skytes ut fra en rakett eller slippes fra ballong, fra 300 – 1000 meters høyde. Sonden skal være på størrelse med en Cola-boks og ha en maksimal vekt på 350 g. Sonden utstyres med sensor kort (shield) som leser av sensorer og overfører måledata til en radiosender som telemetrerer dataene til en mottaker på bakken (bakkestasjon). Sendingen foregår i ISM-båndet dvs. omkring 434 MHz (ca. 70 cm bølglengde). Idet sonden skytes ut/slippes, faller sonden kontrollert mot bakken i fallskjerm. Sensorer samler inn måledata under fallet som kontinuerlig overføres til bakkestasjonen.

Minimumsspesifikasjoner

Det stilles følgende krav til sonden:

1. Sonden skal minimum måle trykk og temperatur hvert 3. sekund.
2. Sonden skal overføre data til bakkestasjonen under fallet.





3. Sonden skal bygges slik at den får plass i en 330 ml standard Cola-boks, eller slik at ingen deler av sonden stikker ut over omfanget til en slik boks.
4. Boksen skal veie mindre enn hva en full Cola-boks veier, dvs. mindre enn 350 g.
5. Sonde skal drives med batteri eller solcellepaneler.
6. Antennen skal være fleksibel og ikke stikke ut mer enn ca. 10 cm når den er stuvet sammen ved oppskyting.
7. En fallskjerm skal være forsvarlig festet til den ene enden av boksen.

Krav til bakkestasjon

Bakkestasjonen skal...

1. ... kunne motta på senderfrekvensen til senderen i sonden (ca. 434 MHz)...
2. ... og kan være utstyrt med en rettningsantenne som kan følge sonden i fallet.
3. ... motta alle data fra sonden under fallet og lagre i PCen for senere analyse og presentasjon

Krav til bæreraket

Bæreraketten skal...

1. ... løfte sonden opp til toppunktet for ferden hvor en eller flere (to) sonder skytes ut
2. ... en forsinket ladning inne i raketten skyter ut sondene slik at den kan falle fritt mot bakken
3. ... minimum løfte sondene til ca. 800 meter

Ballongslipp

Som et alternativ til oppskyting kan en slippe sonden fra en værballong. Enten en fritt svevende ballong eller en som er festet til en snor til bakken. En slik løsning vil være billigere og lar seg lettere gjennomføre på egen skole enn en rakettoppskyting. Ulempen er at slippet blir mer avhengig av vind- og værforholdene og vil sannsynligvis måtte skje fra noe lavere høyde.

NB! Både oppskyting og slipp fra ballong må avtales med de lokale myndigheter, ev. nærliggende flyplasser.

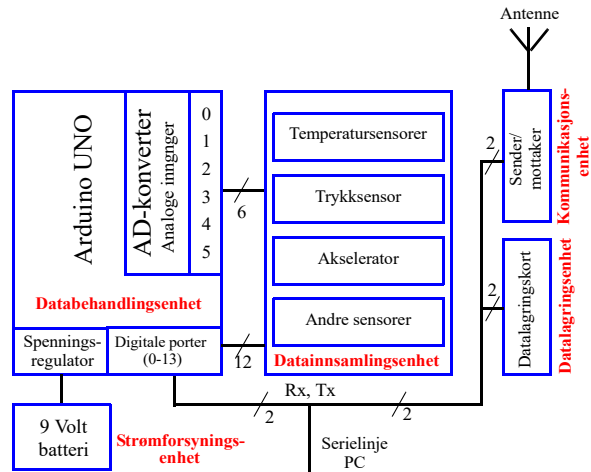


2 Montering og installasjon

2.1 Kort beskrivelse av systemets deler

Sonden består hovedsakelig av seks deler:

1. Databehandlingsenhet
(plattform - Arduino UNO)
2. Datainnsamlingsenhet
(nyttelast med til hørende sensorer)
3. Kommunikasjonsenhet
(sender/mottaker for overføring av data til bakkestasjon)
4. Datalagringsenhet
(datalogger for lagring av data)
5. Kraftforskyningsenhet
(batteri med spenningsregulator)
6. Fallsjerm
(system for kontrollert fall)



Blokkdiagrammet over til høyre viser hvordan de ulike enhetene er forbundet.

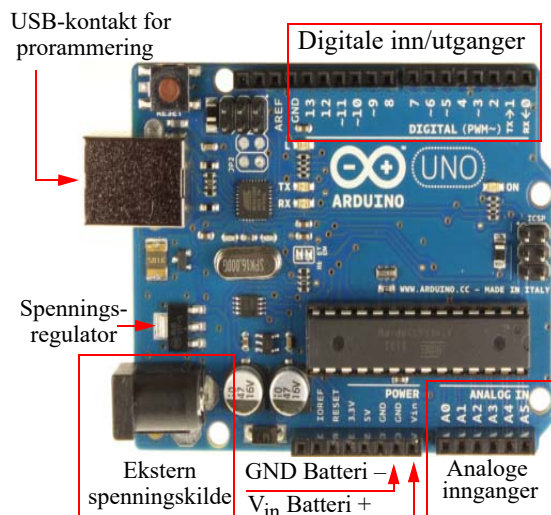
2.1.1 Databehandlingsenhet

Databehandlingsenheten består av en tradisjonell Arduino UNO R3, som er en av de mest populære i mikrokontroller-kortene i Arduino-familien, og dermed leveres til en overkommelig (Robonors pris kr. 249,- inkl. MVA + frakt).

Kortet er bygget opp omkring Atmel prosessoren ATmega328P med en klokkefrekvens på 16 MHz og et flash lager på 32 kbyte, SRAM 2 kbyte og EEPROM 1 kbyte har følgende:

- **Digitale I/O-porter**

Kortet har 14 digitale inn/utporter (I/O-porter) som kan programmeres til enten å være en inn- eller en utgang. Seks av disse (3, 5, 6, 9, 10 og 11) kan pulsbreddemoduleres (pwm), disse er merket med ~ på kortet. Maksimal strøm på I/O portene er 40 mA.





- **Analoge innganger**

Kortet har 6 analoge innganger

- **USB-kontakt** for direkte tilkobling av PC, for programmering av kortet. Under programmeringen tilføres kortet spenning fra USB-kontakten. Dersom dennebelastes med mer enn 500 mA vil strømforsyningen bli brutt inntil strømtrekket reduseres under denne grensen.

- **Strømtilførsel**

Tilkoblingspunkter for batterieliminatør anbefalt spenning 7 – 12 V (grenseverdier 6 – 20 V). Batteri kan enten tilkobles eliminatorpluggen (2.1 mm + senter) eller via V_{in} (+) og GND (-). 5V utgangen lever spenning til f.eks. datainnsamlingskortet. Enkelte komponenter trenger lavere spenning som ev. kan leveres fra 3.3 V utgangen.

- **Reset**

Kortet inneholder en RESET-knapp som resetter programmet som

Kantkontaktene er montert slik at tilleggskort kan monteres rett ned på Arduino-kortet.

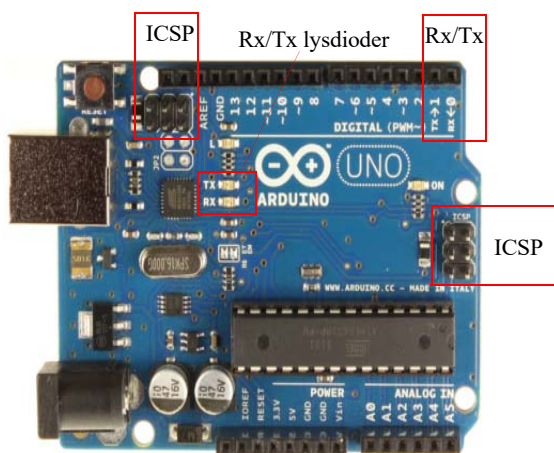
Kortet støtter ulik data kommunikasjon med omverdenen.

- **Rx/Tx**

Kortet støtter UART seriell datakommunikasjon via Rx og Tx portene (I/O-port 0 og 1). Det er også disse som benyttes for programmering av kortet. Disse er også tilkoblet to dioder som vil blinke når kortet kommuniserer med USB/PC. Tilsvarende vil skje når data overføres til programeditoren for monitorering av data på PC-skjermen.

- **I²C-databus**

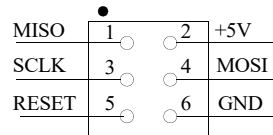
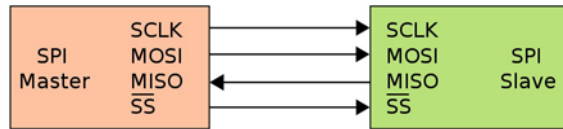
I²C står for Inter IC-bus, og er ment å være akkurat det, da den ble utviklet av *Philips Semiconductor* tidlig på 80-tallet. Bussen er svært enkel med sine to linjer (klokke (SCL) og datalinje (SDA)). Videre er hver krets langs bussen adresserbar. Bussen er dessuten utstyrt med kollisjonsdeteksjon¹. I starten var den definert med en hastighet på 100 kb/s. Senere, etter som en trengte raskere dataoverføring, er *Fast mode* - 400 kb/s og *High speed* - 3,4 Mb/s definert. En slik buss finnes også på Arduino UNO'en og deler linjer med A4 – SDA og A5 – SCL. To av analoge inngangene.



1. For mer informasjon se: <http://www.i2c-bus.org/>

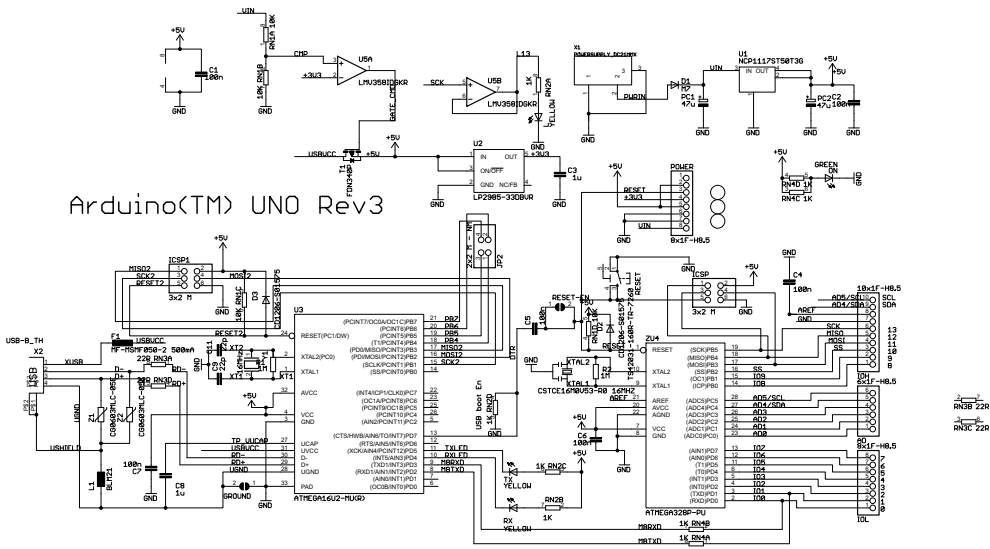
SPI-databus (eller ICSP)

Serial Peripheral Interface buss (SPI-buss). Er en firelinjes databuss brukt for å overføre data til utstyr utenfor mikroprosessoren (periferutstyr). Databussen ble opprinnelig utviklet av Motorola. I en slik dataoverføring fungerer den ene modulen som master (sjefen), og den andre som slave (tjener). På figuren over er enheten til venstre master, og enheten til høyre slave. Master tar initiativet til dataoverføringen ved hjelp av SS-signallinjen, og sender sine data på linjen MOSI (Master Output Slave Input) linjen, mens slaven mottar på sin MOSI linje. Master mottar så svar på sin forespørsel til slaven som sender data tilbake på linjen MISO (Master Input Slave Output) som mottas av master på linjen MISO (Master Input Slave Output). SCLK er klokkesignalet som bestemmer takten til dataoverføringen. SS signalet (Slave select) varsler slaven om at nå ønsker master kontakt. Dersom det er flere slaver i systemet trengs flere SS-linjer, en til hver slave². Arduino UNO har to slike busser, en ICSP og en ICSP1 som også er koblet til D11 - D13.



Figuren under viser kortets kretsskjema hentet fra:

http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf.



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

2. http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus



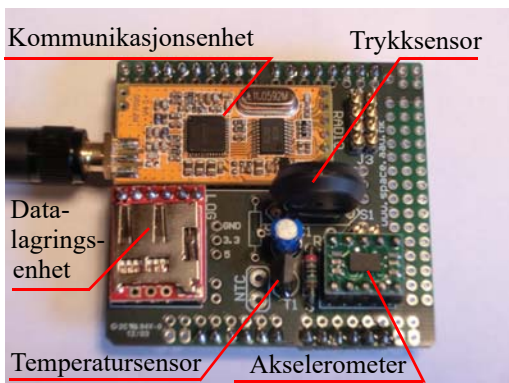
For mer informasjon om Arduino UNO R3 se: <http://arduino.cc/en/Main/ArduinoBoardUno/>

2.1.2 Datainnsamlingsenhet

Denne enheten består av et kort utviklet av miljøet rundt Jens Dalsgaard Nielsen ved Aalborg Universitet.

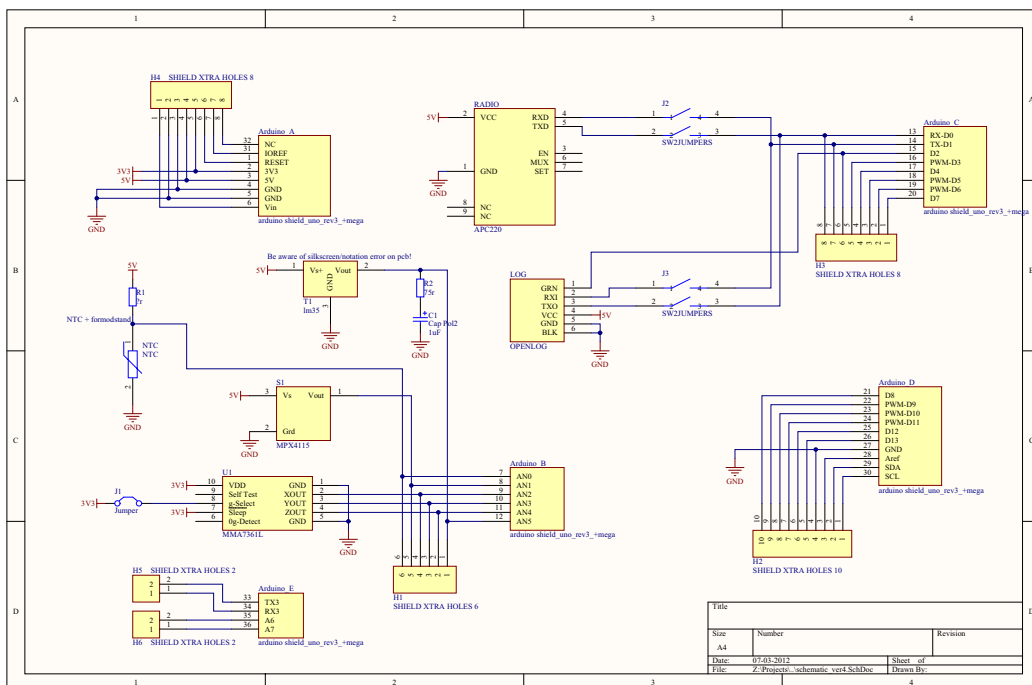
Kortet har følgende funksjoner:

- Kommunikasjonsenhet (APC220 m/antenne)
- Trykksensor (MPX4115)
- Temperatursensor (LM35DZ)
- Temperatursensor (NTCLE100E3103JB)
- Akselerometer 3-akse (MMA7361L)
- Datalagringsenhet (Openlog)
- I tillegg er det satt av plass til en NTC-sensor.



Kretskortskjema

Figuren under viser koblingsskjemaet til datainnsamlingskortet (versjon 3).

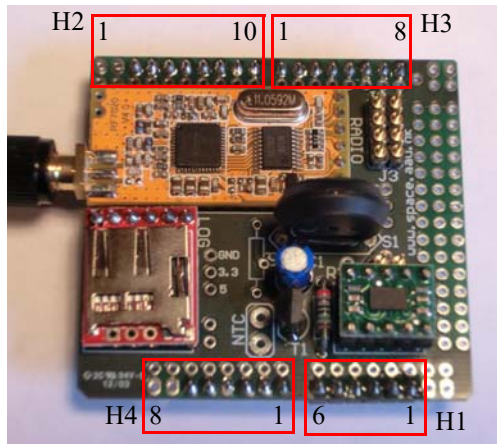




Følgende viser hvor stiftlistene er plassert på kortet og hvordan de er nummerert:

Det er i alt fire kontakter:

- H1 - Analoge porter, H1-1-6
- H2 - Digital porter (D8-D13), H2-1-10
- H3 - Digitale porter (D0-D7), H3-1-8
- H4 - Power supply, Reset, H4-1-8



Analoge kanaler – H1

De seks analoge kanalene er alle i bruk og er forbundet til Arduino UNO på følgende måte:

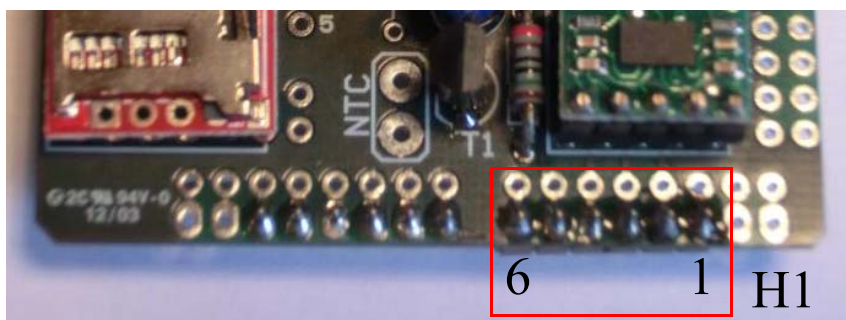
Kontakt	Måling	Sensor	Område	Analog kanal
H1-1	Temperatur (LM35)	LM35DZ	0 - 0,5 V	5
H1-2	Akselerometer z-akse	MMA7361L	0 - 3,3 V	4
H1-3	Akselerometer y-akse	MMA7361L	0 - 3,3 V	3
H1-4	Akselerometer x-akse	MMA7361L	0 - 3,3 V	2
H1-5	Trykk	MPX4115	3,7 - 4,2 V	1
H1-6	Temperatur (NTC)	NTCLE100E3103JB	1,0 - 3,5 V	0

Alle analoge innganger kan håndtere spenninger fra 0 – 5 V. Omregning fra spenning til verdi gjøres ved hjelp av følgende formel:

$$\text{Digital verdi} = \frac{\text{Målt spenning} \cdot 1023}{5V} \quad (2.1)$$



Bildet under viser plasseringen til stiftlista *H1*.



Forøvrig se på side 229 for omregning fra tallverdier til temperatur, trykk og akselerasjon.

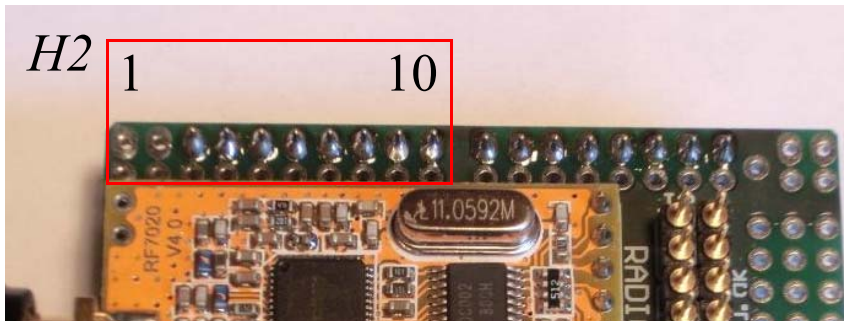
Digitale I/O-porter (8 –13) – *H2*

Tabellen under viser pinningen til stiftlista *H2*.

Kontakt	Betegnelse	Funksjon	Digital I/O-port
H2-1	SCL	Ubrukt	
H2-2	SDA	Ubrukt	
H2-3	Aref	Ubrukt	
H2-4	GND	Ubrukt	
H2-5	D13	Ubrukt	13
H2-6	D12	Ubrukt	12
H2-7	PWM-D11	Ubrukt	11
H2-8	PWM-D10	Ubrukt	10
H2-9	PWM-D9	Ubrukt	9
H2-10	D8	Ubrukt	8



Bildet under viser plasseringen av *H2*.

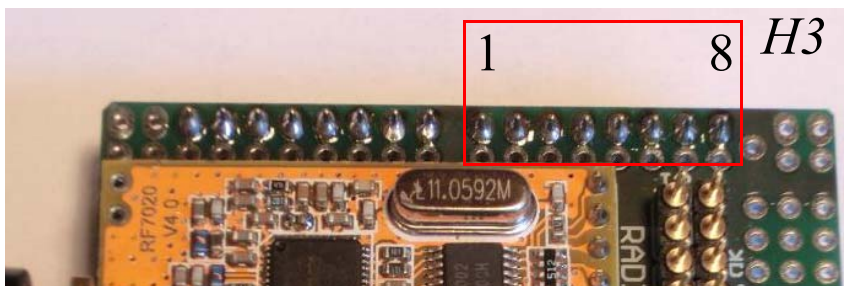


Digitale I/O-porter (0-7) – *H3*

Tabellen under viser pinningen til stiftlista *H3*.

Kontakt	Betegnelse	Radio	Datalogger	Digital I/O-port
H3-1	D7	Ubrukt		7
H3-2	PBM-D6	Ubrukt		6
H3-3	PBM-D5	Ubrukt		5
H3-4	D4	Ubrukt		4
H3-5	PBM-D3	Ubrukt		3
H3-6	D2		Velg Datalogger (1)	2
H3-7	Tx	Send data til radio	Skriv til datalogger	1
H3-8	Rx	Mottatt data fra radio	Les fra datalogger	0

Bildet under viser plasseringen av *H3*.



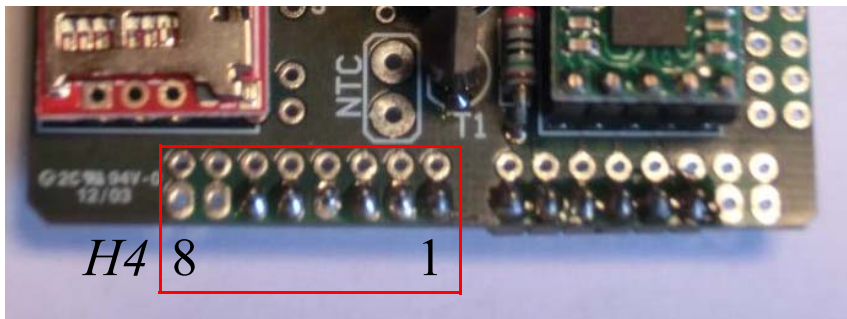


Strømforskyning og Reset – H4

Tabellen under viser pinningen til stiftlista H4.

Kontakt	Betegnelse	Funksjon
H4-1	Vin	Batterikontakt 7 – 12 V
H4-2	GND	Jord
H4-3	GND	Jord
H4-4	5 V	5 V spenningskilde
H4-5	3,3 V	Spennning til akselerator
H4-6	RESET	Reset programmet
H4-7	IOREF	
H4-8	NC	Ikke tilkoblet

Bildet under viser plasseringen av H4.





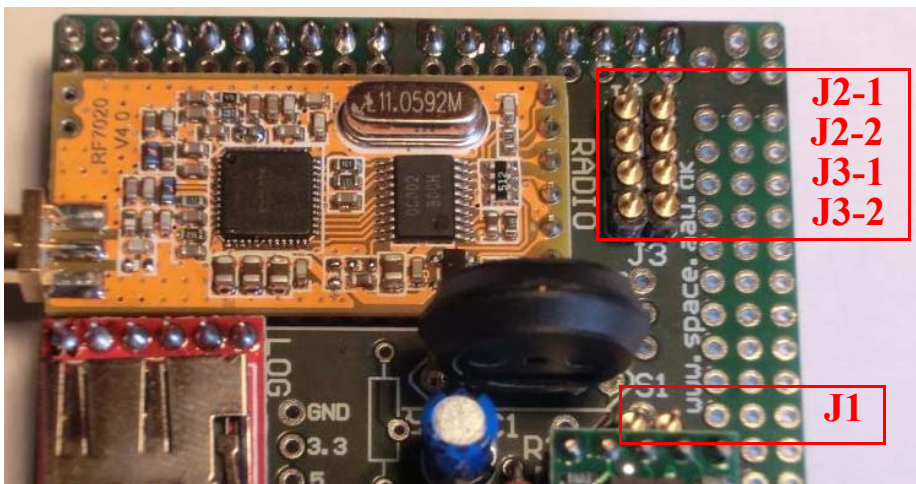
Jumpere

Kortet har 5 jumpere (strapper). Disse har følgende funksjoner. Tabellen under viser betydningen av disse. J2 og J3 kan være innkoblet samtidig.

Jumper	Tilkoblet	Fjernet
J1	± 6 g	$\pm 1,5$ g
J2-1	Ved bruk av radiosender	Fjernes ved programmering av Arduino'en
J2-2	Ved bruk av radiomottaker	
J3-1	Ved skrivning til SD-kort	Fjernes ved programmering av Arduinoen
J3-2	Ved lesing fra SD-kort	

Akseleratoren har to følsomhetsområder, uten jumper (J1) er kretsen mer følsom, men med et mindre måleområde ($\pm 1,5$ g). Med jumper (J1) økes måleområdet, men følsomheten reduseres (± 6 g).

Bildet under viser plasseringen av jumperne. Legg merke til at J2 og J3 alltid opptrer i par:



Programmering (NB)

Når Arduino'en skal programmeres må J2 og J3 alltid fjernes eller at hele datainnsamlingskortet fjernes. Ellers oppnås ikke kontakt med kortet.

Montering av datainnsamlingsenheten

Denne enheten er den eneste som må monteres før CanSat'en er operativ.



2.1.3 Kommunikasjonsenhet

APC220 er en radio som er billig om meget enkel å bruke.

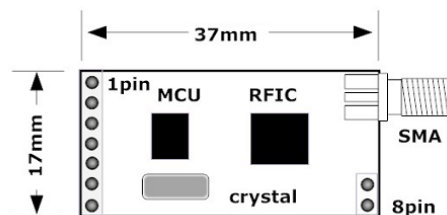
Noen sentrale data³:

- **Senderfrekvens:** 418–455 MHz
- **Forskyningsspennning** 3,3 – 5,5 V
- **Strømforbruk mottaking:** 28 mA
- **Strømforbruk sending:** 42 mA ved 20 mW
- **Sendereffekten** er inntil 20 mW (+13dBm)
- **Datarate til kretsen:** 1200 - 57600 bps
- **Datarate radiokanal:** 1200 - 19200 bps



APC220

Pin 1 GND
in 2 VCC
in 3 EN
in 4 RXD
in 5 TXD
in 6 AUX
Pin 7 SET



I vårt tilfelle vil det være aktuelt å sende med en datarate på maksimalt 9600 bps. En kan imidlertid regne med en effektiv datarate på under det halve, siden det legges inn ekstra bit (redundans) for å korrigere feil i overføringen.

Kortet mates med data og styresignaler via 7 terminaler. For at kortet skal være aktivt må EN enten være større enn 1,6 V eller frakoblet. Gjennom TXD tilføres kortet data på seriell form for utsending, mens mottatte data kan hentes ut fra RXD.

En USB–TTL overgang mellom sender/mottaker kortet og USB inngang på PCen følger med. Denne kan brukes både for å programmere sender/mottaker modulene og under vanlig bruk for å ta data inn i PCen. For mer data se vedlegg D.1.

Det neste som må gjøres er å sette opp APC220 slik at den kommuniserer på riktig frekvens og med riktig baud rate. Her finnes det to alternative måter:



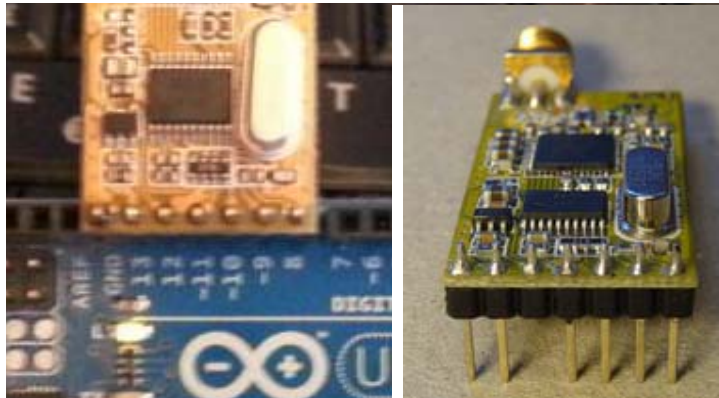
USB-TTL overgang

- **Alternativ 1:** Konfigurering av APC220 ved bruk av Arduino-kortet
- **Alternativ 2:** Konfigurering av sender/mottaker kortene med *RF-magic*

3. http://www.prathobbies.com/info_pages/cansat_tx_datasheet.pdf

Alternativ 1: Konfigurering av APC220 ved bruk av Arduino-kortet⁴

Den enkleste måten å konfigurere radioen APC220 på, er ved å bruke de digitale I/O-portene på Arduino UNO'en. Radioen plugges ned i portene 8 – GND, pekende ut fra Arduino-kortet, som vist til venstre på figuren til høyre. Husk også klipp av tredje bein fra venstre da enkelte PCer har problemer med dette som vist til høyre på figuren til høyre (Foto: Jens F. Dalsgaard Nielsen).



Gjør følgende:

1. Overfør konfigureringsprogrammet “apc220cansat” over til Arduino'en på vanlig måte⁵.
2. Åpne monitorvinduet. Følgende melding skal komme opp:
APC config program -vrs 102 - Nov 06 2012
3. Det er nå mulig å konfigurere APC220 ved å skrive kommandoer i innboksen øverst i monitorvinduet som vist under. Kommandoen overføres til Arduino'en ved å trykke sendknappen lengst til høyre.:



4. Plasser markøren i innboksen og trykk *Return* og følgende meny kommer opp:
APC config program -vrs 102 - Sep 29 2012
pingOFF echoOFF
commands:

```
r: Read apc220 radio config
e: go into echo mode: receive char, add 1 and return
n: no more echo - back to normal
p: print how to attach radio to Arduino
w: Write apc radio config...
   'w' FFFFFFF R P B C - number of letters indicates precise number of digits
   FFFFFFF: frequency: 434000 (434 MHz) range 418000-455000
   R:      Rf data rate - 1/2/3/4 equals 2400/4800/9600/19200bps
```

4. Utviklet av Jens Dalsgård Nielsen og Thomas Gansmoe

5. En listing av programmet finnes i vedlegg A.3 og på It's learning.



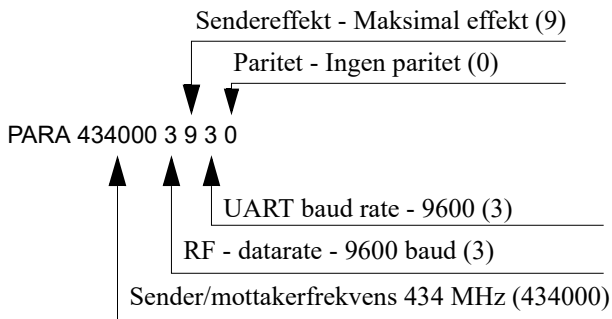
- P: Radio output power - 0.. 9 equals 13dBm (20mW).
B: UART baudrate - 0/1/2/3/4/5/6 equals 1200/2400/4800/9600/19200/
38400/57600bps
C: Byte Check Parity - 0/1/2 equals NoCheck(8N1)/EvenParity(8E1)/
OddParity(8O1)

Write example: w 434000 3 9 3 0 is...
434,000 MHz 9600 baud in air, 20mW, 9600baud on UART, No Parity
(8N1)

After 10 seconds with no keyboard input we will emit a char every two
second going into ping mode - emit a char every two second on apc220!

5. Les eksisterende konfigurasjon av APC:

Skriv "r" i innboksen og trykk Return. Arduino'en returnerer APC220's konfigurasjon f.eks.:



6. Echo On

- Ecko slås på med kommandoen "e"

7. Echo Off

- Echo slås med kommandoen "n"

8. Konfigurasjon

- Oppkoblingen av APC220 vises ved å trykke "p"

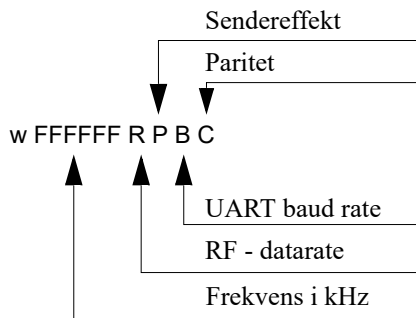
Attach radio directly to Arduino on pin 8 to gnd

```
-----  
Aaref  
RGND-----  
D13|=antenna connector ->  
IOREF U12|  
RESET I11|  
3.3V N10 |  
5V O 09 |  
GND 08 -----  
GND P |
```



Vin	C 07
	B ~ 06
A0	~ 05
A1	04
A2	~03
A3	02
A4	TX 01
A5	RX 00

9. Skriv ny konfigurasjon til APC:
Skriv "w" i innboksen fulgt av etterfølgende parametere:



Se pkt. 4 for alternativer

Baudrate (symboler pr. sek):

En bør være klar over at når man benytter radiolinken er den reelle baudraten (her bitraten) betydelig lavere enn den en ber om. Det skyldes at radiolinken tilfører ekstra bit til feilsjekk og feilretting. I praksis betyr det at dersom man ber om en rate på 9600 baud, så kan man regne med byterate på typisk 800 byte/sek.

Det sikreste er imidlertid å prøve seg fram.

Alternativ 2: Konfigurering av sender/mottaker kortene med RF-magic

For å programmere sender/mottaker kortene trengs programmet: APC22X_V12A (RF-magic). Dette kan hentes ned fra nettet på følgende nettsted:

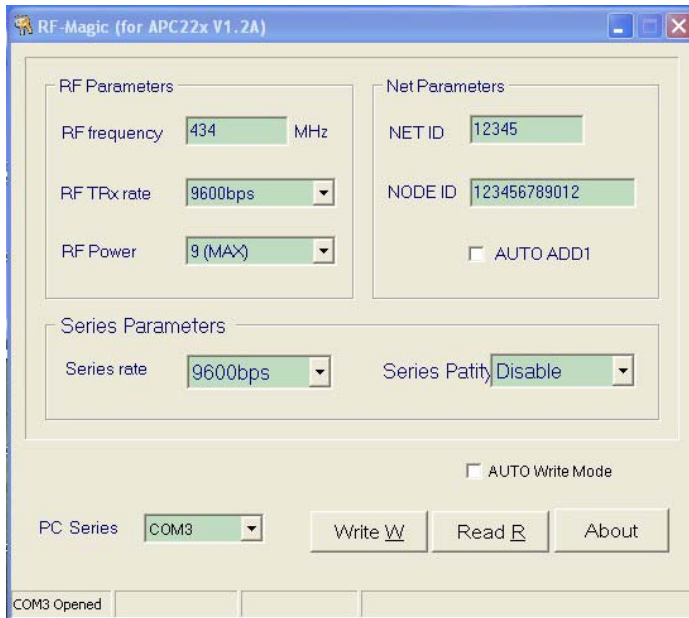
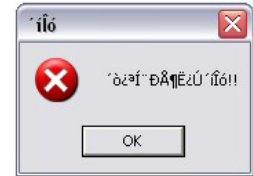
<http://wiki.openpilot.org/display/Doc/APC220+Transceiver+Telemetry>

Programmet **RF-magic** finnes et stykke ned på siden under avsnittet *Module PC connection* sammen med drivere for USB-TTL overgangen.

For å programmere sender/mottakerkortene går en fram på følgende måte:



1. Sett sender/mottakeren inn i USB-TTL overgangen og sett det inn i en ledig USB plugg på PCen. Forhåpentligvis vil Windows kjenne igjen modulen og installere de nødvendige driverne.
2. Start programmet APC22X_V12A
3. Dersom feilmeldingen vist til høyre kommer opp skyldes det sannsynligvis at programmet ikke finner USB-TTL overgangen på forventet COM-port. I så fall må com-port adressen endres. Dette er forklart på den nevnte nettsiden⁶.
4. Når programmet har fått kontakt med sender/mottaker-kortet kommer følgende dialogboks opp.



5. Her kan følgende parametere settes opp og overføres til sender/mottaker-kortet:
Frekvens f.eks. 434 MHz
Datarate f.eks. 9600 bps
Sendereffekt f.eks. 9 som tilsvarer 20 mW (maks)
PCens serieport (COM)
Nettverksparametere, NETT ID og NODE ID
Paritet f.eks. Disable
6. Skriv de valgte parametrene til sender/mottaker kretsen ved å trykke *Write W*

6. <http://wiki.openpilot.org/display/Doc/APC220+Transceiver+Telemetry>



Problemer med kommunikasjon med com-porten

Dersom man har problemer med kommunikasjonen mot com-porten anbefaler **Thomas Gansmoe**⁷ (NAROM) følgende framgangsmåte:

USB-TTL converteren (det lille kortet mellom radio og PC) trenger en egen driver for å lage en virtuell COM-port. Denne kan hentes fra: <http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>

Driveren fra dette nettstedet http://www.silabs.com/Support%20Documents/Software/CP210x_VCP_Windows.zip burde fungere for de fleste som benytter operativsystemene WinXP, Vista, Win7 eller Win8.

Installer driveren. Når dette er gjort burde du se «Silicon Labs CP210x» under Porter i Enhetsbehandlingen når den er pluggert inn i PCen (Kontrollpanel -> Maskinvare og Lyd -> Enhetsbehandling -> Porter (Com og LPT)).

Her vil du også se hvilket COM-port nummer den har. Hvis dette er et høyt tall (over 10) kan du forsøke å endre dette til en lavere COM-port (Egenskaper-> Portinnstillinger).

Selve programvaren (RF-magic) trenger administratorrettigheter. Da er det ikke nok at du har en bruker med administratorrettigheter. Høyreklikk på programikonet og velg «Kjør som administrator». Uten å gjøre dette vil programmet starte, men du vil ikke få kontakt med radioen.

ISM-båndet 433.050 – 434.790 MHz (LPD433)

Det finnes en rekke ISM-bånd (Industrial, Scientific and Medical) som er ment for ulike kommunikasjonsformål og annen bruk hvor det ikke kreves beskyttelse for gjensidig interferens. En må derfor sørge for at bruken kan tolerere å bli forstyrret.

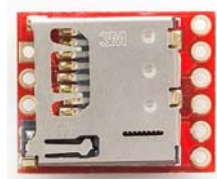
Båndet 433.050 – 434.790 MHz er et slikt bånd som gjerne benyttes for kommunikasjon eller styring over korte avstander. Båndet er delt inn i 69 kanaler hver med en båndbredde på 25 kHz og maksimal sendereffekt er 10 mW.

For mer informasjon se: <http://en.wikipedia.org/wiki/LPD433>

2.1.4 Datalagringsenhet

Openlog DEV-09530⁸ er en “open source” datalogger som kan lagre data på en minnebrikke av typen microSDHC med inntil 32 HGbyte minne kan brukes (pris kr. 89,- for 2 Gbyte, 99 for 4 Gbyte hos Expert).

På undersiden sitter en integrert krets som vist på figuren til høyre. Denne tar imot data på seriell form (2 400 – 115 000 bps) og skiver og leser til minnekortet.



7. Anbefalt i e-post 30. april 2013

8. Se mer informasjon om dataloggeren: <https://www.sparkfun.com/products/9530>

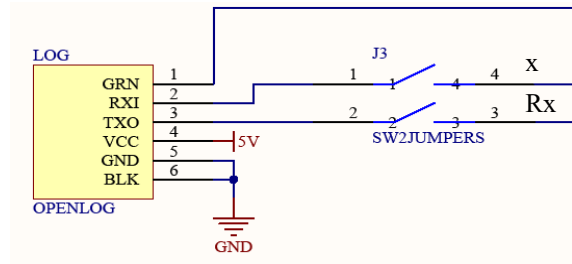


Kortet håndterer supplyspenninger fra 3.3 – 12 V og trekker 2 mA i stand by og maksimalt 6 mA under skrivning og lesing av data. Kretsen er beskyttet mot ombytting av supply-spenningen. En LED på kortet lyser når data skrives. Loggeren kan frakobles ved å fjerne jumperne J3, se figuren under.

Kortet tillater filnavn med 8.3 karakterer. Dvs. åtte karakterer før punktet og tre etter. En god regel er å bruke store og små bokstaver og tall, men unngå spesialtegn.

Loggeren har seks terminaler:

GRN	Loggeren resettes når denne legges lav
RXI	Serielle data skrives til Openlog
TXO	Serielle data leses fra Openlog
VCC	Supply spenning 3,3 - 12 V
GND	Jord, 0 V
BLK	Jordet



Dersom Tx på Arduino UNO er tilknyttet RXI og Rx på Arduino UNO er tilkoblet TXO kan en skrive til Openlog ved hjelp av `print()` og `println()` kommandoene.

Mer informasjon om kretsen kan finnes i databladet:

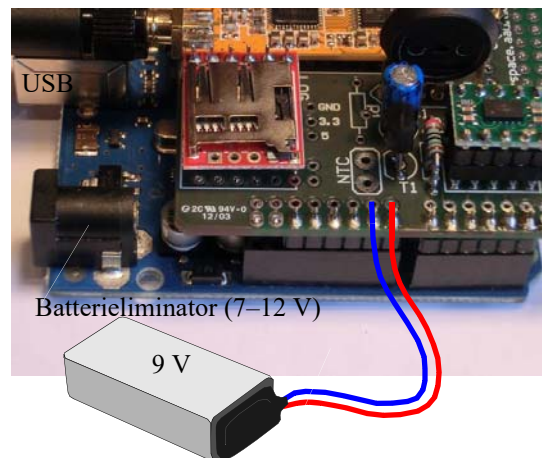
<https://github.com/sparkfun/OpenLog/wiki/Datasheet>

For skrivning og lesing av data til og fra loggeren se Avsnitt 4.3.4 på side 85.

2.1.5 Kraftforsyningsenhet

Kretsen kan i prinsippet ha tre typer spenningskilder.

1. Ved programmering vil forsynings-spenningen tilføres via USB kontakten
2. Under normal bruk vil spenningen tilføres via et 9 V batteri av typen 6LF22/6LR61
3. Dersom det under eksperimentfasen er behov for mer strøm enn en USB kan drive, kan en koble inn en batteri-eliminator på Arduino-kortet. Denne må selvfølgelig erstattes av et batteri med tilstrekkelig kapasitet ved slipp eller rakettoppskyting.

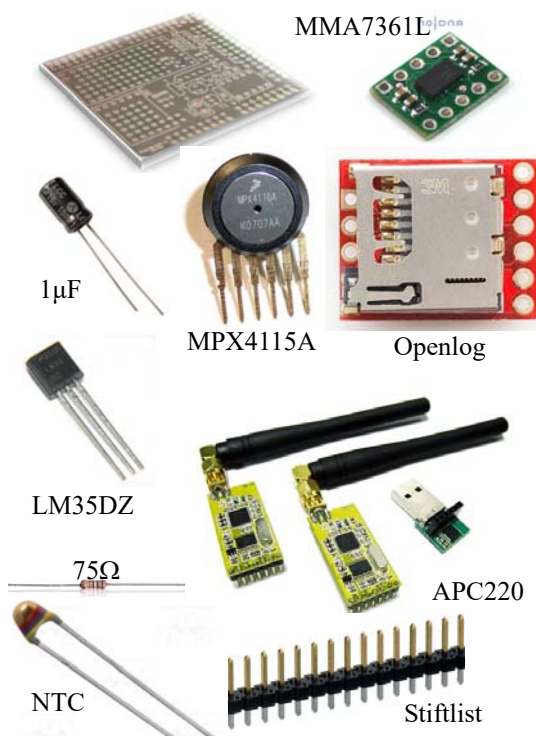


2.2 Montering

2.2.1 Montering av datainnsamlingsenheten

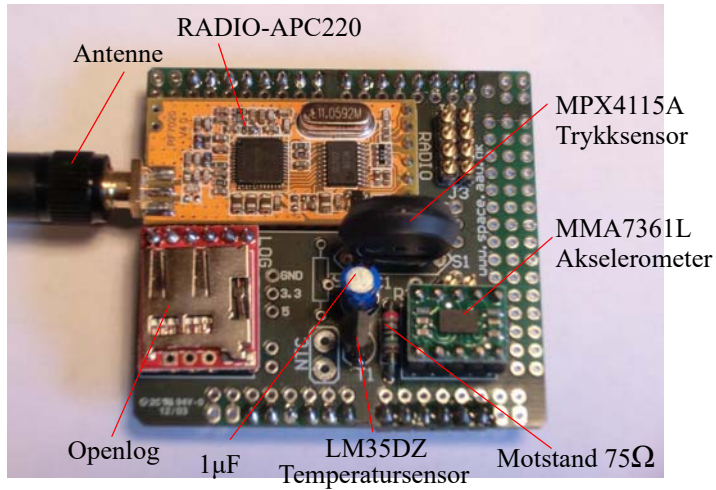
Det eneste som skal monteres innledningsvis er datainnsamlingskortet. Dette består av følgende deler:

- Kretskort, utviklet og i salg fra Aalborg Universitet.
- Stiftlist kan kjøpes fra ELFA og monteres langs kanten av kretskortet for å forbinde det med Arduino UNO kontrollkortet.
- Akselerometer MMA7361L
Dette er montert på et lite kretskort med ti bein (DIL - Dual In Line) (se Avsnitt 7.7.1 på side 157).
- Trykksensor MPX4115A
Denne har seks bein (SIL - Single In Line) (se Avsnitt 7.5.1 på side 146)
- Datalogger, Openlog
Denne har seks bein (SIL) (se Avsnitt 2.1.4 på side 33)
- Radiomodul APC220
Denne har sju bein (SIL) (se Avsnitt 2.1.3 på side 28)
- Motstand 75Ω brun - sort, fiolett, grønn, sort (1%)
- Elektrolytt kondensatoren $1\mu\text{F}$, legg merke til polariteten
- Temperatursensor LM35DZ
Denne skal plasseres med den flate siden inn mot motstanden. Dette er omvendt av hva som er markert på kortet (se Avsnitt 7.3.2 på side 134)
- NTC-motstand – NTCLE203E3103FB0
Nominellverdi $10\text{k}\Omega$ 5% (brun, sort, orange - gull (topp))

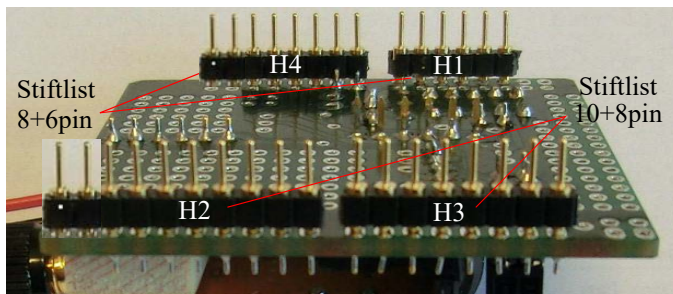




- Stiftlister for strapping
 - 2x4 DIL
 - 2 SIL

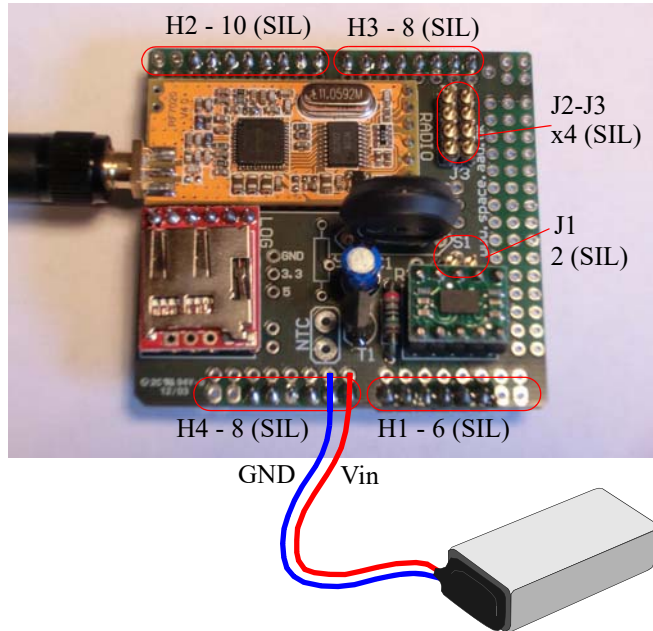


- Kontaktlister langs kanten av kortet
 - H1 - 6 (SIL)
 - H2 - 10 (SIL)
 - H3 - 8 (SIL)
 - H4 - 8 (SIL)



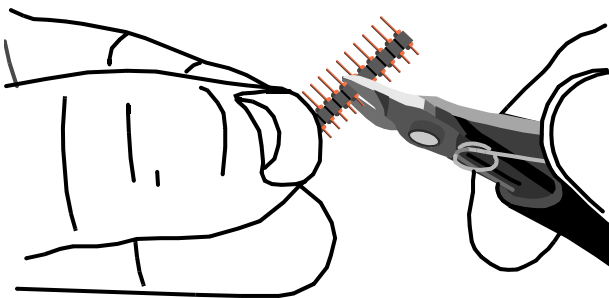


- Batterikontakt for 9 V batteri



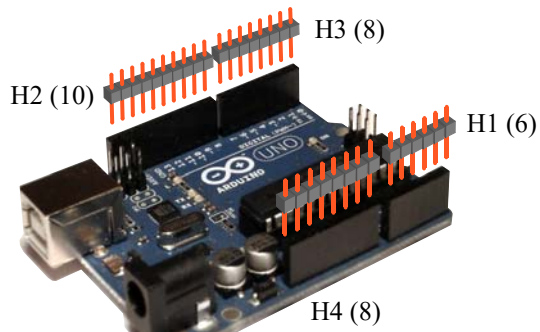
Detaljert byggebeskrivelse

Følgende beskrivelse viser byggeprosessen trinn for trinn:



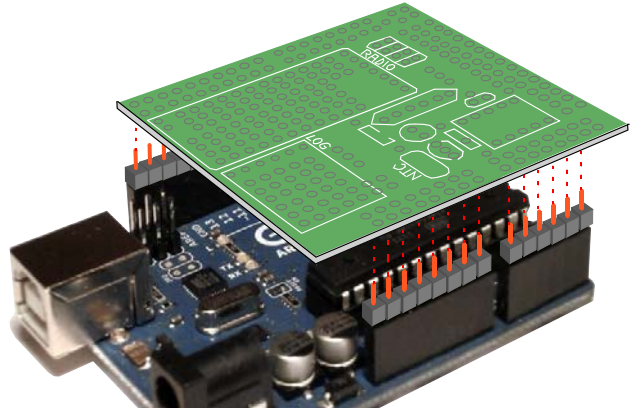
1. [] Klipp opp stiftlistene med en sideavbiter i følgende lengder:
 - 6 stifter (H1)
 - 10 stifter (H2)
 - 8 stifter (H3)
 - 8 stifter (H4)

2. [] Sett stiftlistene ned i kontaktene i Arduino-kortet som støtte under loddingen.

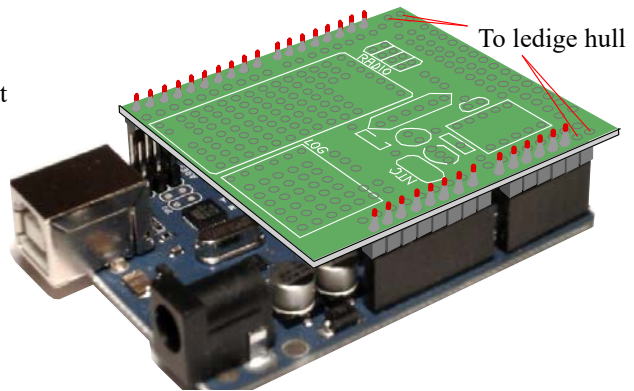




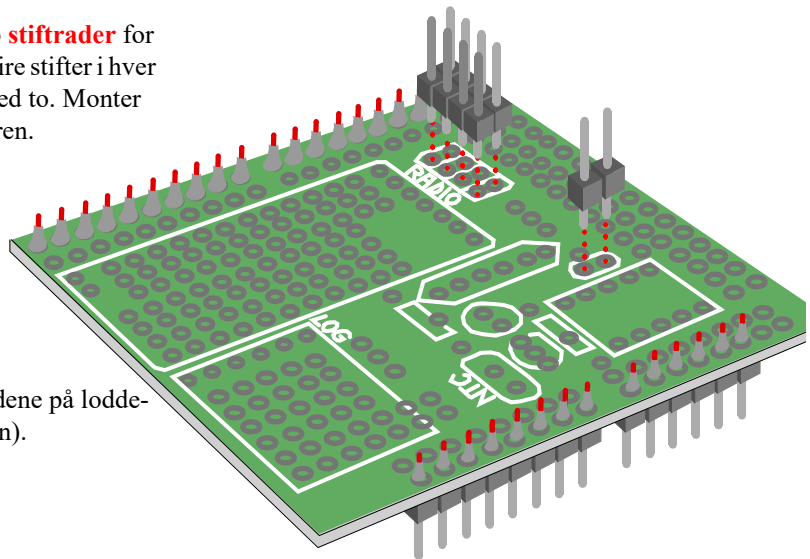
3. Plasser datainnsamlingskortet på stiftlistene.



4. Plasser **datainnsamlingskortet** på stiftlistene. Det skal være to ledige hullrader bakerst på kortet.
5. Lodd fast stiftradene på komponentsiden (oversiden) av kortet



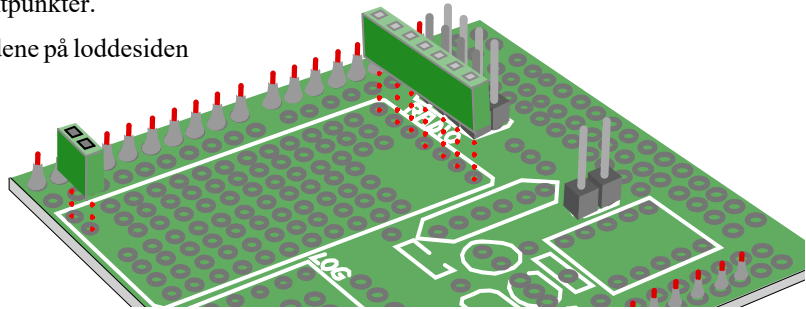
6. Klipp opp to **stiftrader** for "jumper" med fire stifter i hver rad, og en rad med to. Monter som vist på figuren.



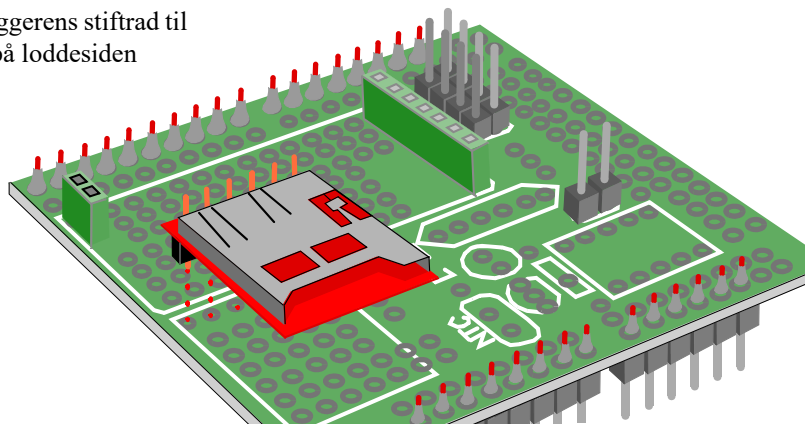
7. Lodd stiftradene på loddesiden (undersiden).



8. [] Monter **soklene** for radiokortet, 7 + 2 kontaktpunkter.
9. [] Lodd stifradene på loddesiden (undersiden).



10. [] Monter **dataloggeren**.
11. [] Lodd dataloggerens stifrad til loddepunktene på loddesiden (undersiden).

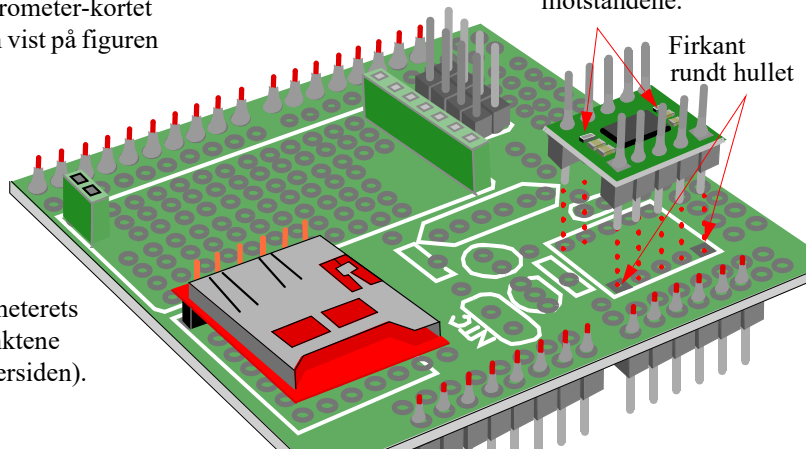


12. [] Monter stifrekket på begge sider av **akselerometer**-kortet.
Vend den korte siden av stiftene ned. Monter akselerometer-kortet på sensorkortet som vist på figuren
Pass på at de to *små svarte motstandene* er plassert inn mot midten av kortet.

NB! De to svarte motstandene.

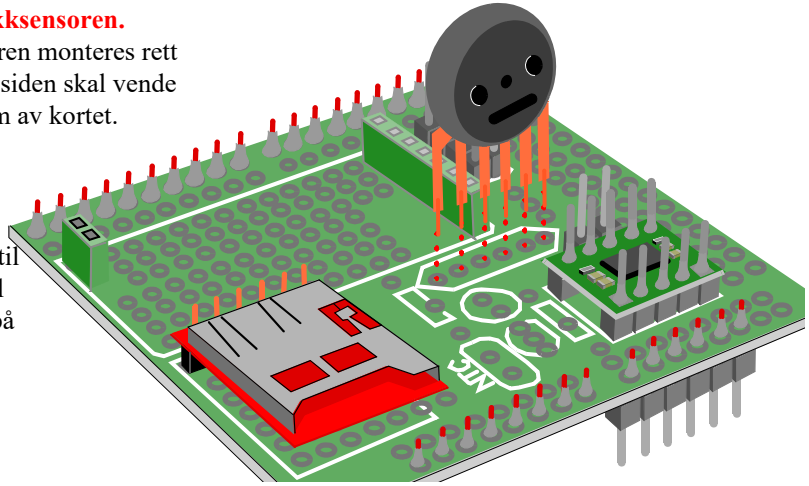
Firkant rundt hullet

13. [] Lodd akselerometerets stifrad til loddepunktene på loddesiden (undersiden).





14. [] Monter **trykksensoren**.
Pass på at sensoren monteres rett vei. Den blanke siden skal vende inn mot senstrum av kortet.

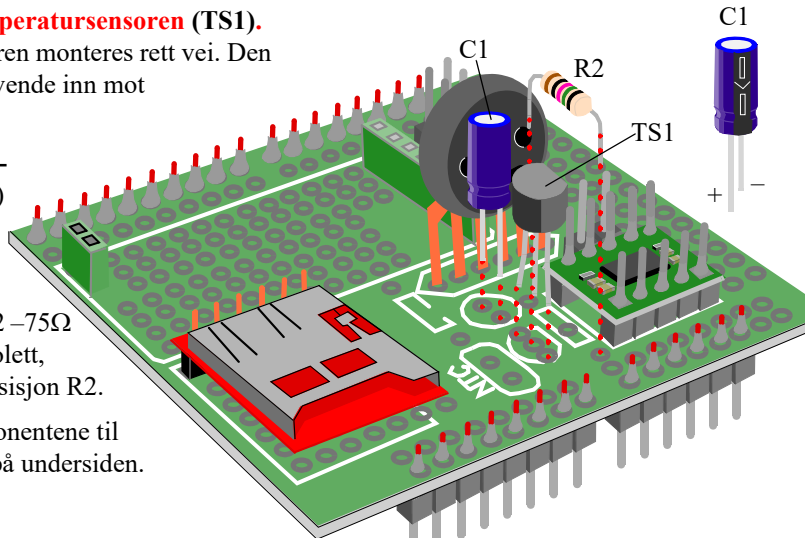


15. [] Lodd beina til trykksensoren til loddepunktene på loddessiden (undersiden).

16. [] Monter **temperatursensoren (TS1)**.
Pass på at sensoren monteres rett vei. Den flate siden skal vende inn mot akselerometeret.

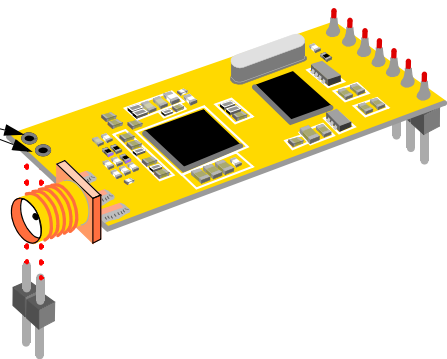
17. [] Monter **kon-**
densatoren (C1)
med (-) inn mot sentrum av kortet. Monter **motstanden (R2 –75Ω**
– brun – sort, fiolett, grønn, sort) i posisjon R2.

18. [] Lodd komponentene til loddepunktene på undersiden.



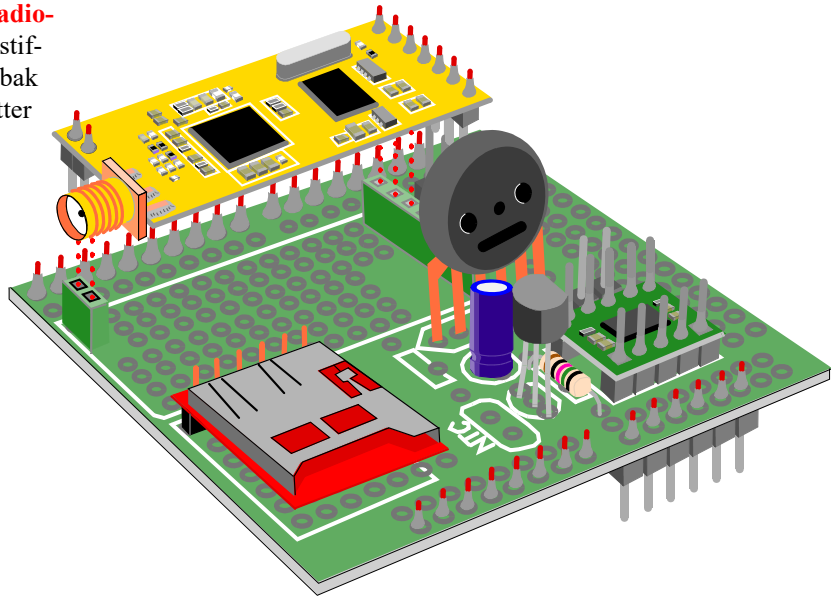
19. [] Monter **stiflisten** med to stifter foran på radiokortet. De kort stiftene skal monteres i kortet. Loddes på oversiden

20. [] Lodd **stiflisten** på komponentsiden (oversiden) av radiokortet

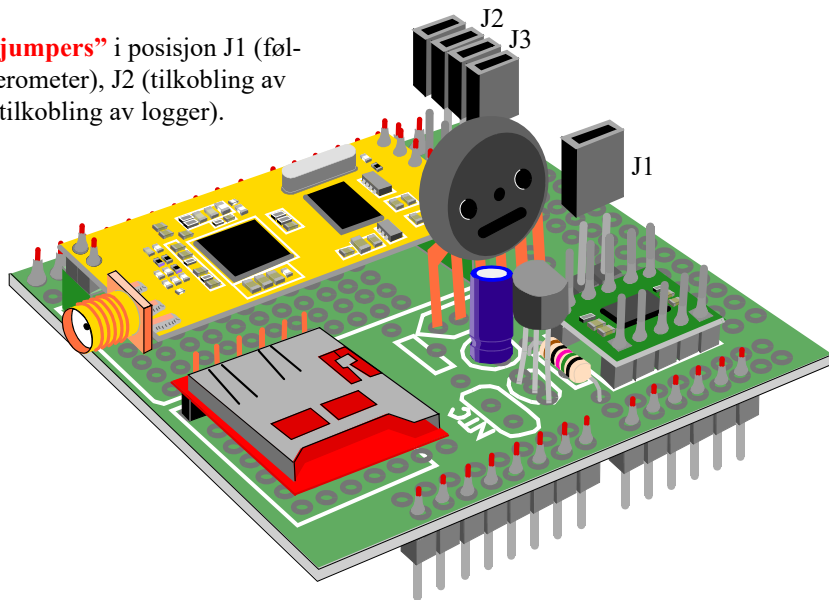




21. [] Monter **Radio-kortet** slik at stiftene foran og bak på kortet smetter ned i soklene.

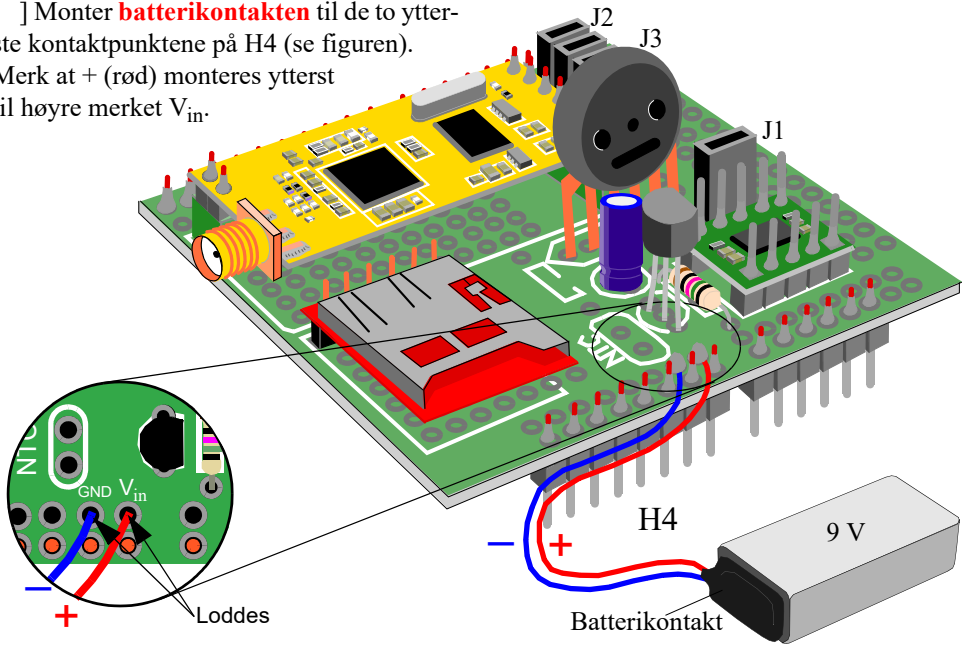


22. [] Monter **“jumpers”** i posisjon J1 (følsomhet akselerometer), J2 (tilkobling av radio) og J3 (tilkobling av logger).

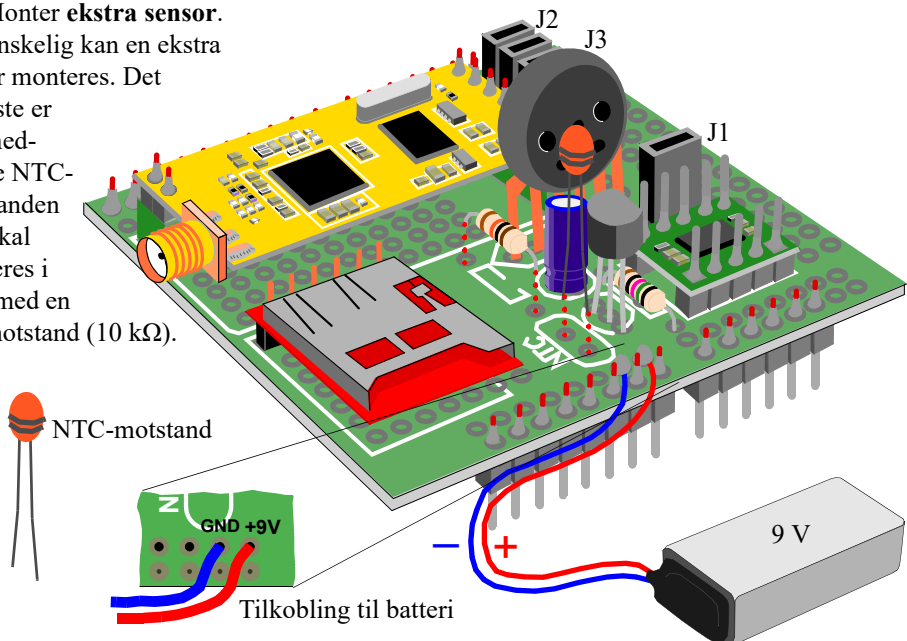




23. [] Monter **batterikontakten** til de to ytterste kontaktpunktene på H4 (se figuren). Merk at + (rød) monteres ytterst til høyre merket V_{in} .



24. [] Monter **ekstra sensor**. Om ønskelig kan en ekstra sensor monteres. Det enkleste er den medsendte NTC-motstanden som skal monteres i serie med en fast motstand (10 k Ω).

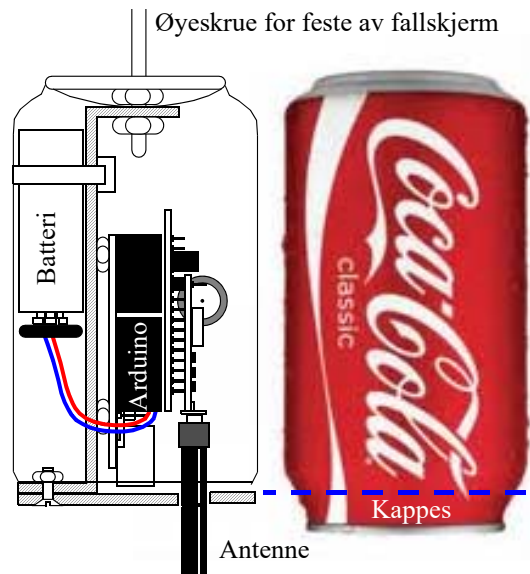


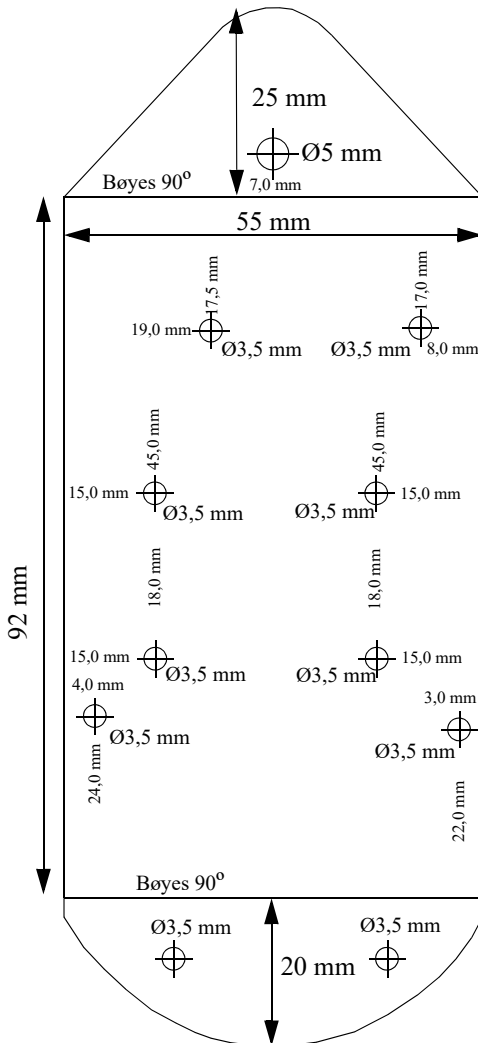
Komponentlister med leverandører og prisantydninger finnes i vedlegg Avsnitt C.1 på side 224.

2.2.2 Montering i brusboks (Can)

Hele konseptet baserer seg på at hverken vekt eller volum skal være større enn en standard brusboks. Elektronikken må monteres på en robust brakett som tåler en oppskyting og frigjøring fra en rakett. Videre festes fallskjermen direkte til braketten og ikke til boksen. Brusboksen blir derfor kun et utvendig hylster. Betydningen av en robust konstruksjon for å oppnå et vellykket resultat, kan ikke understrekes sterkt nok.

Tegningen til høyre viser hvordan elektronikken kan monteres i brusboksen. I dette tilfellet har vi valgt å la antennen stikke ut på undersiden. Å la en stiv antenne stikke ut på undersiden, er ingen lur konstruksjon med tanke på utskyting fra rakett eller mht. å ta i mot støtet ved landing, men er praktisk under utprøving. Et bedre alternativ er å benytte en trådanntenne som loddes direkte til kretskortet eller monteres på en SMA kontakt (se avsnitt Avsnitt 2.3 på side 45).



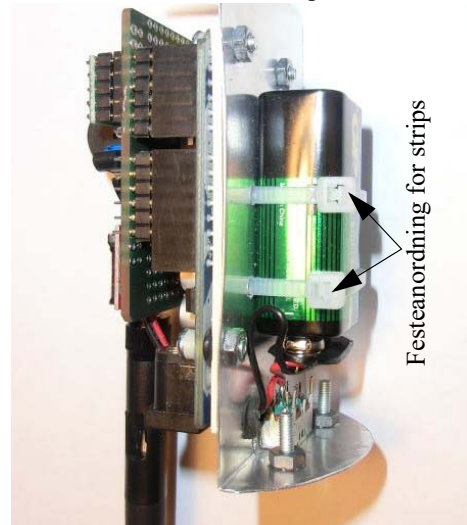


Tegningen til venstre viser braketten med mål og plassering av hull før den er knekt (bøyd). Det kan benyttes 0,5 – 1,0 mm aluminiumsplate.

Plata knekkes henholdsvis øverst 90° inn i papirplanet og nederst 90° ut av papirplanet. Knekklinjene er markert øverst og nederst på malen til venstre.

En mutter tilpasset øyeskruen for feste av fall-skjerm, limes fast på innsiden. Vi har brukt tokomponents epoksy-lim.

Husk å sette av hull for å stripse fast batteriet som monteres på baksiden av braketten. Der-som batteriet monteres rett bak mikrokontrollerkortet, må en sørge for at feste-anordningen til stripsene ikke kommer på undersiden av kortet som vist på bildet under.

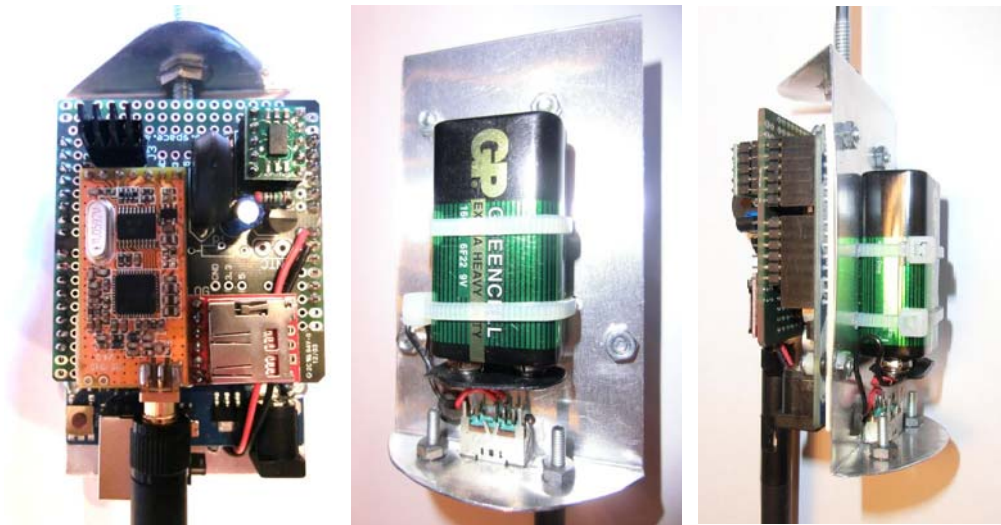


Om ønskelig kan en av/på-bryter monteres på braketten slik at en slipper å måtte demontere son-den for å slå av og på strømmen. En må imidlertid være klar over at en bryter vil være et sårbart punkt under oppskyting og frigjøring fra raketten.

Et sirkulært bunnlokk kan skrues fast til den halvsirkelformede nederste delen av braketten. Her må en enten lime mutrene på baksiden eller bruke små plateskruer. Bunnplate må ha hull til antennen.



På bildene under ser vi et eksempel på hvordan sonden kan monteres.



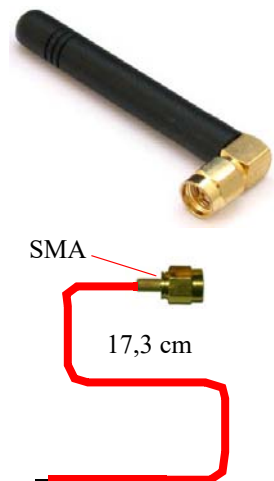
2.3 Antennen

Antennen som normalt følger med APC220, er en forkortet “pisk-antenne”. Det er rimelig å tro at en liten forkortningsinduktans ligger nederst i foten til antenna.

Et godt alternativ er en enkel trådentenne som enten loddes direkte til senderens utgang eller monteres til en SMA-kontakt. Normalt vil en slik antenne være $1/4$ -bølgelengde og laget av isolert ledning. Tykkelsen er relativt ukritisk, det viktigste er at den er robust og er fleksibel. Antennelengden kan lett beregnes fra ligning (2.2) når frekvensen, f , og lyshastigheten, c , er kjent:

$$L = \frac{c}{4f} = \frac{3 \times 10^8}{4 \cdot 434 \times 10^6} [\text{m}] = 0,173 \text{ m} \quad (2.2)$$

Frekvensen 434 MHz gir dermed en antennelengde på 17,3 cm.

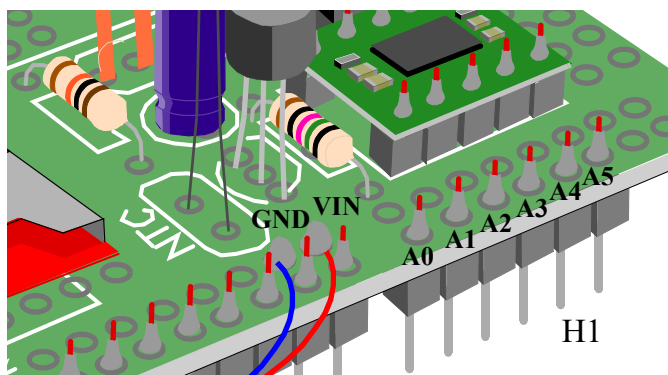




2.4 Uttesting av hårdvaren

2.4.1 Innledende uttesting av sensorene på shield-kortet

Etter at man har kontrollert lodninger og at alle komponenter er montert på riktig sted, kan det være lurt å måle spenningene fra de ulike sensorene før noen programmer er lagt inn. Man måler da spenningen mellom jord (GND) og hver av de seks analoge inngangene som man finner på kontakt H1, som vist på figuren under.



Tabellen under gir en oversikt over omtrentlige spenningsverdier målt på de ulike analoge inngangene under normale forhold i laboratoriet. Det antas at kortet ligger rettvendt og flatt på bordet.

Kontakt	Analog kanal	Måling	Sensor	Område	Forventet spenning
H1-1	A5	Temperatur (LM35)	LM35DZ	0 - 5,0 V	ca. 0,2 V
H1-2	A4	Akselerometer z-akse	MMA7361L	0 - 3,3 V	ca. 2,45 V (-1 g)
H1-3	A3	Akselerometer y-akse	MMA7361L	0 - 3,3 V	ca. 1,65 V (0 g)
H1-4	A2	Akselerometer x-akse	MMA7361L	0 - 3,3 V	ca. 1,65 V (0 g)
H1-5	A1	Trykk	MPX4115	0 - 5,0 V	ca. 4,2 V
H1-6	A0	Temperatur (NTC)	NTCLE100E3103JB	0 - 5,0 V	ca. 2,7 V

Dersom det er betydelige avvik mellom målt og forventet verdi bør man slå av strømmen og gå over lodninger og se etter om komponentene er plassert riktig. Dermed om noen av sensorene er ødelagt.

Før du går videre med uttesting av hårdvaren må programeditoren installeres slik at test-programmer kan lastes ned på Arduino-kortet. Dette er beskrevet i avsnitt 4.1 på side 77.



2.4.2 Kalibrering av sensorene

Før CanSat'en kan benyttes for et virkelig måleoppdrag, må verdiene fra de ulike sensorene regnes om til ønsket form samt kalibreres. I dette avsnittet skal vi oppsummere de viktigste omregningsformlene for de faste sensorene på CanSat shield-kortet og vise hvordan de kan kalibreres.

Omregning fra mottatt tallverdi til spenning:

$$\text{Spenning} = (\text{Tallverdi}/1024) * 5,0 \text{ V} \quad (2.3)$$

H1-1 (A0) Omregning fra spenning til temperatur (LM35DZ)

Spenning (V) fra sensor er omregnet fra tallverdien som hentes inn fra AD-konverteren

$$\text{Temperatur (Celsius)} = \text{spenning (V) fra sensor} * 100 (+ \Delta T) \quad (2.4)$$

Korreksjonen $+\Delta T$ legges inn på bakgrunn av en kontrollmåling i det aktuelle temperaturområdet. Denne korreksjonen korrigerer kun for avvik i y-retningen. Det antas at kurven er lineær.

For mer informasjon om halvleder temperatursensorer, se avsnitt 7.3.2 på side 134.

H1-2, 3, 4 (A1, 2, 3) Omregning av akselerometerverdier fra spenning til g (MMA7361L)

Når akselerometeret befinner seg i fritt fall (uten rotasjon), vil det ikke påvirkes av g-krefter i noen retning. Dvs, at spenningene som kommer fra akselerometeret i x-, y- og z-retning, skal alle ideelt sett være 1,65 V. Følsomheten for måleområdet $\pm 1,5 \text{ g}$, er $0,8 \text{ V/g}$. Vi kan dermed sette opp:

$$\text{Akselerasjonen (g)} = (\text{Målt spenning (V)} - 1,65 \text{ V} + \Delta V)/0,8 \text{ V/g} \quad (2.5)$$

Korreksjonsverdien $+\Delta V/0,8$ legges inn på bakgrunn av måling.

Kalibrering gjøres ved å legge CanSat'en i ulike posisjoner og notere verdiene som vist på figuren. Derne st kan man korrigere ved å legge til eller trekke fra verdier for å forskyve måleområdet. Eller man kan multiplisere med en faktor for å ekspandere eller krympe måleområdet.

La oss se på et eksempel:

+g Calibration of the accelerometer
+/- 1,5 g (option)

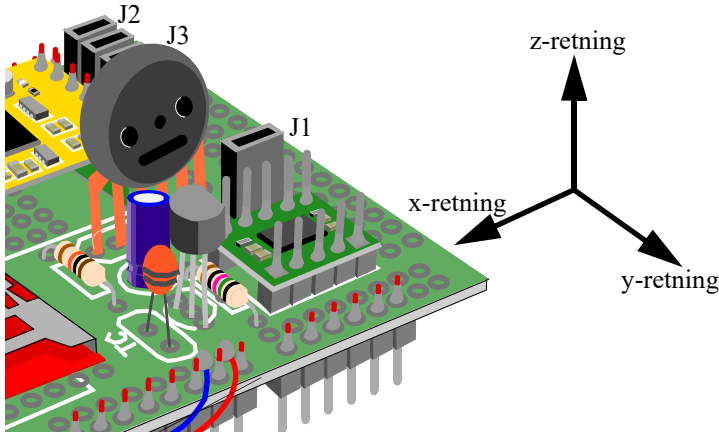
 $V_x = 1,65\text{V}$ $V_y = 1,65\text{V}$ $V_z = 2,45\text{V}$	 $V_x = 1,65\text{V}$ $V_y = 0,85\text{V}$ $V_z = 1,65\text{V}$	 $V_x = 2,45\text{V}$ $V_y = 1,65\text{V}$ $V_z = 1,65\text{V}$
 $V_x = 1,65\text{V}$ $V_y = 2,45\text{V}$ $V_z = 1,65\text{V}$	 $V_x = 0,85\text{V}$ $V_y = 1,65\text{V}$ $V_z = 1,65\text{V}$	

1. Legg CanSat'en mest mulig horisontalt på bordet. Spenning V_z skal være 2,45 V, dvs. den skal vise en tyngdeakselerasjon lik $+1g$.



2. La oss si at vi i stedet måler 2,35 V. Med andre ord vi har et avvik som vi må korrigere for. Det gjør vi ved å legge til korreksjonen lik $\Delta V = +0,1$ V til alle verdiene i z-retning. Vi antar da at avviket er det samme over hele måleområdet, dvs. at måleverdien kun er forskjøvet (*offset*).
3. Vi setter korreksjonsverdiene for spenningen inn i formelen og beregner den korrigerte g-verdien i z-retning.

$$\text{Akselerasjonen (g)} = (\text{Målt spenning (V)} - 1,65 \text{ V} + 0,1 \text{ V}) / 0,8 \text{ V/g} \quad (2.6)$$



Når CanSat holdes i den retningen som antydnet ved hjelp av aksekorset til høyre på figuren, skal resultatet i z-retning være +1,0 g. At fortegnet til den målte akselerasjonen blir positivt, skyldes at tyndeakselerasjonen, som virker nedover, vil presse arduinoen ned mot bordet. En lignende effekt ville vi ha fått dersom vi satte Arduinoen i en akselerert bevegelse oppover, dvs. en akselerasjon på +1g.

Tilsvarende gjøres for x- og y-retningen.

For mer informasjon om akselerometeret se avsnitt 7.7.1 på side 157.

H1-5 (A4) Omregning fra spenning til trykk (MPX4115)

Omregning fra spenning målt fra AD-konverteren (V_p) og til trykk (P), er oppgitt av fabrikanten til å være [7]:

$$P = 22,222 * V_p + 10,556 \text{ [kPa]} \quad (2.7)$$

For omregning fra lufttrykk (Pa) til høyde (meter), kan følgende omregning benyttes:

$$h = \frac{T_1}{a} \left(\left(\frac{p}{p_1} \right)^{\frac{aR}{g_0}} - 1 \right) + h_1 \quad (2.8)$$



Hvor:

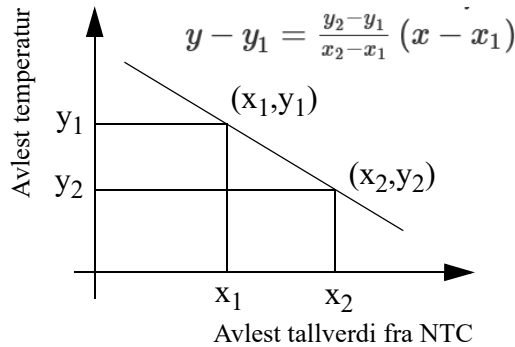
h	Beregnet høyde i meter
h_1	Starthøyde i meter
T	Temperatur i Kelvin
T_1	Starttemperatur i høyden h_1
a	Temperaturgradient, foreslått verdi $-0,0065$ K/m
p	Trykk i Pa
p_1	Trykk i Pa ved starthøyden
g_0	Tyngdeakselerasjonen $9,81$ m/s ²
R	Den spesifikke gasskonstant $287,06$ J/kg K

For mer informasjon om trykksensorer se avsnitt 7.5 på side 145.

H1-6 (A5) Omregning fra spenning til temperatur (C) (NTCLE100E3103JB)

Denne beregningen antar at det er en omtrent lineær sammenheng mellom måleverdi og temperaturer i det aktuelle området. Dette er ikke langt fra sannheten i et avgrenset område (f.eks. 0 – 50 C). Under denne betingelsen kan følgende metode benyttes med tilstrekkelig nøyaktighet:

1. Bruk et termometer å mål “virkelig” temperatur f.eks. innendørs (y_1), les av tallverdien som kommer fra NTC-motstanden (x_1).
2. Ta så Arduino’en med ut samtidig som du logger data (vi antar at det er kaldere ute). Mål “virkelig” temperatur med et termometer (y_2) og les av tallverdien fra CanSat (x_2).



Figuren over viser hvordan topunktsformelen kan brukes for å finne sammenhengen mellom målte verdier og temperatur.

Topunktsformelen for lineære ligninger kan også skrives slik:

$$y = ((y_2 - y_1)/(x_2 - x_1)) * (x - x_1) + y_1 \quad (2.9)$$

Hvis $k = (y_2 - y_1)/(x_2 - x_1)$ (stigningskoeffisienten), kan vi skrive ligningen slik:



$$y = k(x - x_1) + y_1 \quad (2.10)$$

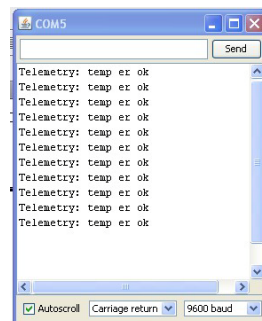
Sett verdiene inn i formelen over og finn et uttrykk for y (temperatur i C) som funksjon av tallverdien hentet fra AD-konverteren som måler på NTC-motstanden (x).

For mer informasjon om NTC-resistorer se avsnitt 7.3.1 på side 131.

2.4.3 Test av seriell datalinje

Det kan være lurt å begynne med Arduino UNO kortet alene:

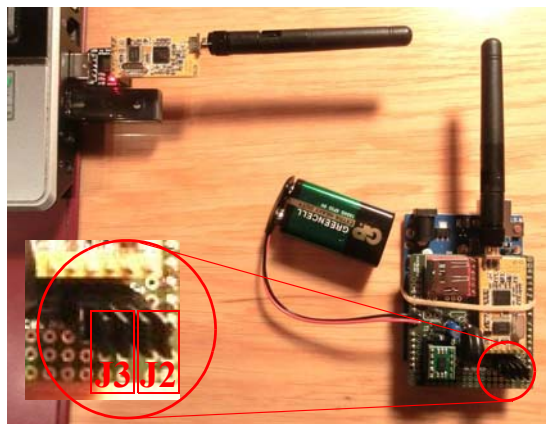
- Last opp programeksempel: **cansat kode serial (01)**
Fra nettstedet: http://www.control.aau.dk/~jdn/edu/cansat-kit/cansat_01_seriel.html **Husk strappene J2 og J3 må være ut ved programmering av Arduino-kortet.**
Start programmet og registrer at lysdioden på kortet blinker med jevne blink.
- Åpne monitorvinduet ved å klikke på ikonet til høyre på menylinja i editoren.
- Monitoren vil da skrive ut det Arduino sender tilbake på PCens serieinngang (USB).



2.4.4 Test av radiokommunikasjon

Det samme programmet som vist i eksempelet foran kan også brukes til å teste radiokanalen.

- Last opp programeksempel: **cansat kode serial (01)** som beskrevet over. **Pass på at strappene J2 og J3 er tatt ut ved nedlasting av programmet.**
- Sett inn strappene J2-1 og J2-2 (innfelt bilde til høyre).
- Sett opp begge APC220-modulene ved hjelp av programmet *RF-magic* som beskrevet i Avsnitt 2.1.3 på side 28. Vær nøye med at begge APC220-modulene er konfigureres likt og arbeider på samme frekvens.
- Plugg en av modulene inn i USB-porten på PCen, ved hjelp av USB adapteren, og legg den andre, *mobile enheten*, i nærheten. Husk at den mobile enheten må tilkobles egen strømforsyning, batteri eller batterieliminatort, siden den ikke får strøm fra USB-porten.





- Åpne Arduino editoren, velg riktig USB-port ved hjelp av *Tool* og *Serial Port* på menylinja og åpne monitorvinduet. Du skal da se de utsendte dataene som om CanSat-modulen var koblet til PCen med USB-kabelen.

2.4.5 Test av datalagringsenheten

Det samme programmet som vist i eksemplene foran kan også brukes til å teste datalagringsenheten, Openlog.

- Last opp programeksempel: *cansat kode serial (01)* som beskrevet over.
- Sett inn strappene J3-1 og J3-2.
- Koble batteriet til mobilenheten som automatisk vil starte programmet og skrive data inn i datalagringsenheten (datalogger/Openlog).
- Slå av mobilenheten, ta mikro-kortet ut av loggeren og sett det inn i kortleseren på PCen, eller en kortadapter som vist på figuren over.
- Kontroller innholdet i filene ved hjelp av en teksteditor.



2.5 Lagring av data på file (terminalprogrammer)

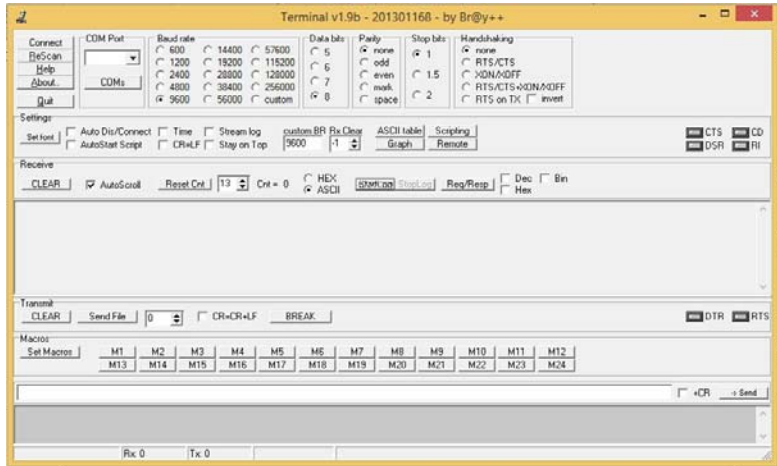
Når dataene sendes på seriell form til USB-inngangen på PCen kan de vises i Arduino'ens monitorvindu. Imidlertid kan en ikke lagre datastrømmen som vises i monitorvinduet. Alternativt kan en benytte et terminalprogram for visning og lagring av data.



2.5.1 Terminal

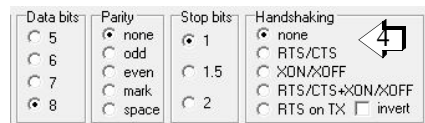
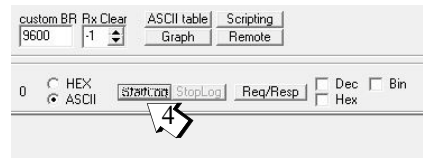
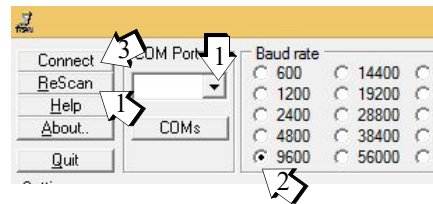
Programmet *Terminal* trengs ikke å installeres, men kan kjøres direkte fra filen som er hentet ned fra nettet: <https://sites.google.com/site/terminalbpp/> (eller fra programfilen utlevert på kurset).

Når du trykker på programikonet starter programmet og man får opp vinduet vist på figuren til høyre.



Oppsett av programmet:

1. *Velg COM-port.*
Velg COM-porten til radioen
Om du ikke finner den riktige porten kan du prøve å trykke *ReScan*. Da vil alle tilgjengelige porter leses inn på nytt.
2. *Velg datahastighet*
Velg 9600 Baud fra listen
3. *Opprett kontakt*
Trykk *Connect* for å koble opp forbindelsen mot COM-porten.
4. *Oppsett av kommunikasjon*
Oppsett av kommunikasjonen gjøres ved å velge antall *Data bits*, *Parity*, *Stop bit* og *Handshaking*.



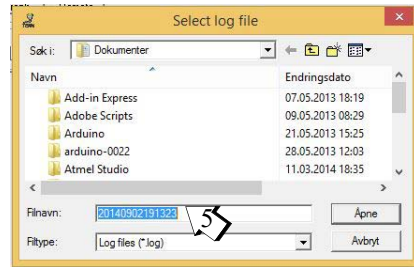


5. Lagring av data

Trykk *StartLog* for å starte logging av data. Du vil da få spørsmål om å oppgi et filnavn. Oppgi filnavn og katalog og trykk *Åpne*.

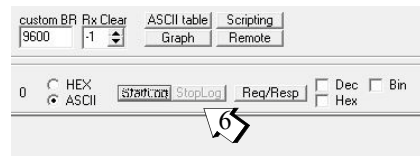
6. Avslutt lagring

Trykk *StopLog* for å avslutte logging av data.



7. Filformat

Dersom dataene er hensiktsmessig organisert, kan filen leses rett inn i Excel.



2.5.2 Tera Term VT

Programmet viser data som kommer på en av serieportene (USB). Programmet er gratis og kan hentes fra:

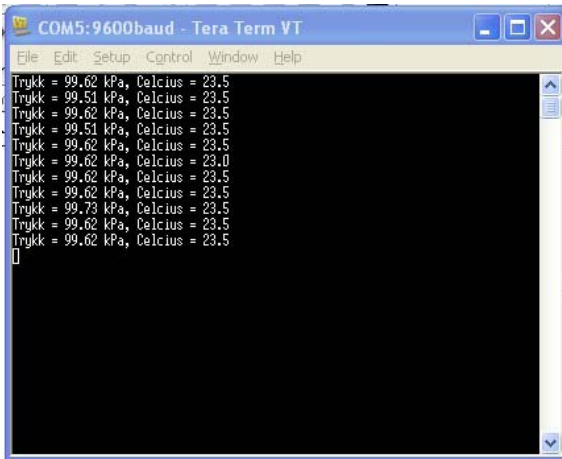
<http://logmett.com/index.php?/download/tera-term-475-freeware.html>

Versjon 4.75 er den siste pr. 31. aug. 2012.

Programmet installeres og en ikon legges på skrivebordet. Når programmet starter kommer følgende vindu opp (figuren til høyre):

Velg seriell port og finn den riktige com-porten og trykk OK.





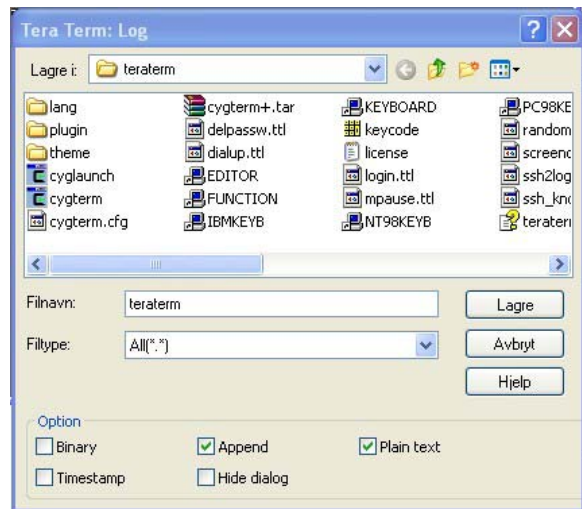
Deretter kommer terminalvinduet som vist til venstre.

Dersom en ønsker å lagre data gjøres dette ved å velge *File* fra menylinjen og deretter velg *Log*.

Velg katalog og skriv inn filnavnet. Dersom en velger et filnavn som finnes og merker av i boksen *Append*, vil programmet fylle inn nye data på slutten av filen. Dersom en ikke krysser av *Append* vil filen skrives over.

Kryss av *Plain text* dersom dataene skal tolkes som ASCII-tegn.

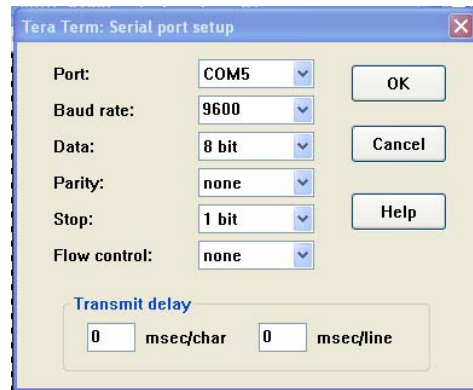
Om dataene er rent tallmateriale med komme mellom hvert tall på linjen, vil disse lett kunne kopieres inn i Excel.



Samtidig med terminalvinduet vil det når en velger å skrive data til fil komme opp et *Log* menyvindu. Dette forteller hvilket filnavn og hvor dataene ligger. Her kan en også stoppe datalagringen for en tid (*Pause*) eller stoppe lagringen (*Close*).



Dersom en har behov for å sette datahastighet (Baud rate) eller velge en ny com-port. Kan det gjøres ved å velge *Setup* fra menylinjen og *Serial port*. Se figur til høyre.







3 Oppskyting

3.1 Ballongslipp

Et alternativ til rakettoppskyting er ballongslipp. En ballong med helium ca. 150 cm i diameter før slipp, er mer enn tilstrekkelig for å løfte to CanSat'er til 1000 meter om man har tilstrekkelig med tråd. En fjernstyrt utløsermekanisme kan benyttes for å utløse CanSat'ene når man når ønsket høyde. Med en fallhastighet på ca. 8–12 m/s vil falltiden ved vertikalt fall ta ca. 80–130 sek. i vindstille vær. En kan oppnå rikelig med resultater selv ved 300–400 meters høyde. Ev. kan man redusere fallhastigheten.

3.1.1 Utstyr for ballongslipp



Snora kan f.eks. vindes opp på et fiskehjul for linefiske eller pilk som vist på figuren til venstre. Men enklere utgaver kan brukes. F.eks. en gammel bilfelg lagret opp og påmontert sveiv.

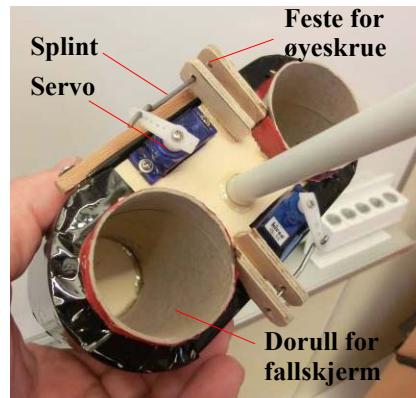
Ballongen er en helium ballong for innsamling av metrologiske data. (f.eks. Totex TX350). Skoler som ønsker å arrangere sitt eget ballongslipp kan henvende seg til NAROM for kjøp av ballong. Pris på forespørsel.

I tillegg trengs helium, som kan kjøpes hos en forhandler av gass, f.eks. AGA eller lignende. Disse selges ofte på stålflasker som returneres etter bruk. En bør kjøpe et munnstykke for fylling av ballonger. Dette kan gjenbrukes hver gang en trenger helium. Slike koster noen hundre kroner. En 50 liters flaske med 200 bars trykk (10 000 liter), holder til 5 - 6 fyllinger.

3.1.2 Slippmekanisme

Her er det mange muligheter. Det enkleste er å la utstyret følge ballongen opp og ned. En kan dermed til en hver tid ha god kontroll på hvor CanSat'en befinner seg.

Dersom man ønsker å gjennomføre selve slippet, kan man bruke en ekstra snor for å dra ut en splint slik at CanSat'en slippes. Dette krever at det lages en passende utløsermekanisme til å feste CanSat'en til ballongen.





Eller man kan lage en fjernstyrt utløsermekanisme. En slik kan bygges med deler for fjernstyring av modellfly. Bildet over til høyre viser utløsermekanismen som ble benyttet under lærerкурset ved NAROM august 2011. Denne kan slippe to CanSat'er uavhengig av hverandre. De to servo-ene betjener hver sin splint som holder CanSat'ene oppe. Fallskjermen pakkes inn i et rør (dorull) slik at festesnorene til fallskjermen ikke floker seg. Det er særdeles viktig at den pakkes slik at den folder seg ut idet den slippes.



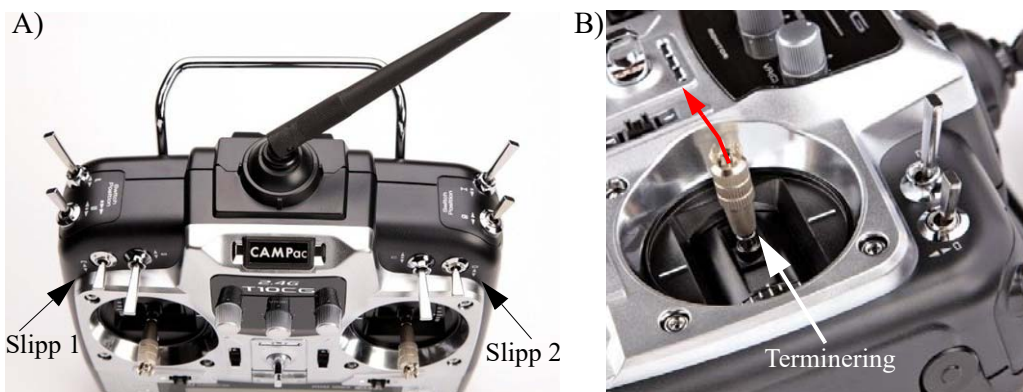
Figur 3.1 Fjernstyrt slippmekanisme for to CanSat'er

Fjernstyringsenheten som ble benyttet i dette tilfellet var en FutabaT10CAG 10 kanals sender og en F6106 HFC 6 kanals mottaker fra samme firma⁹.



Figur 3.2 Fjernstyringsenheten FutabaT10CAG 10 kanals sender og en F6106 HFC 6 kanals mottaker fra samme firma.

Styringen av slipp 1 (kanal 5 og kanal 6) og slipp 2 var tilkoblet de to bryterne som antydnet på figur 3.3A. Ved å skyve den høyre spaken mot venstre og holde den der i 5 sek. utløses termineringen, figur 3.2B.



Figur 3.3 A) Slipp 1 og slipp 2, hendelen presses forover ved slipp og trekkes bakover ved festing av CanSat. B) Ved å skyve den høyre spaken mot venstre og holde den der i 5 sek. utløses termineringen.

Utstyret er i salg hos ELEFUN (www.elefun.no). Her finnes også billigere varianter av fjernstyringsutstyr.

9. Thomas Gansmoe ved NAROM har utviklet utstyret og valgt fjernstyringsenhet.



3.1.3 Reglement for oppsending av forankret ballong

Regler for oppsending av forankrede ballonger finnes i **BSL F 1-1 Lufttrafikkregler, Kapittel IX. Forankrede ballonger**. Dette regelverket finnes på følgende nettsted: <http://www.lovdata.no/cgi-wift/ldles?doc=/sf/sf/sf-20030207-0252.html>. Regelverket finnes i vedlegg B. Det viktigste er at nærmeste enhet av Luftfartstilsynet eller NOTAM-kontor (ofte tårnet ved nærmeste flyplass) må **varsles senest 14 dager før oppsendingen**. Strengere regler gjelder dersom oppsendingen skal skje innen en radius på 10 km fra flyplass.

Meldingen skal inneholde:

- a) navn, adresse og eventuelt telefonnummer til den som har ansvaret for oppsendingen,
- b) sted/posisjon for forankringen,
- c) ballongens maksimale høyde,
- d) dato, tid og varighet for oppsendingen,
- e) opplysninger om hvordan ballongen med forankringsutstyr er varselmerket.

Dessuten skal ballongen utstyres med en termineringsmekanisme dersom fortøyningen skulle ryke. Ballongen skal dessuten merkes slik at den er lett synlig. Vi brukte oljebasert rød spraymaling.

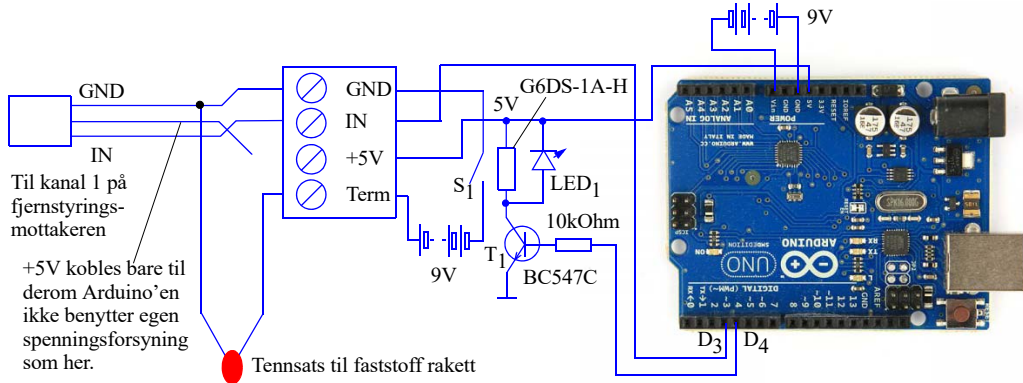
Ved slipp skal tårnet på nærmeste flyplass varsles, likeså når slippet avsluttes. Det er primært helikoptertrafikk som kan bli hindret av forankrede ballonger.

Jeg varslet Svein O. Aasen (Svein.Aasen@caa.no) ved Luftfartstilsynet pr. e-post og fikk et skriftlig tillatelse tilbake med pålegg om å følge reglene i lovverket.

3.1.4 Termineringsmekanisme for forankrede ballonger

Dersom forankringen løsner og ballongen stiger til værs, skal den utstyres med en automatisk eller fjernstyrt termineringsmekanisme.

Thomas Gansmoe (NAROM) har utviklet et termineringssystem basert på en Arduino UNO koblet til en av kanalene (kanal 1) til fjernstyringsmottakeren omtalt foran. På bakgrunn av erfaringene fra denne løsningen har **Nils Kr. Rossing** bygget opp følgende utstyr for montasje på toppen av utløsermekanismen:



Figur 3.4 Koblingskjema for terminering av ballong.

Termineringsanordningen benytter som tidligere en Arduino UNO som dekoder signalet fra fjernstyringsmottakeren via digital inngang D_3 . Når programvaren har detektert signal på inngangen i 5 sekunder, går den digitale utgangen D_4 høy slik at transistoren T_1 åpner og reléet S_1 kobler inn og tennsatsen får strøm. En LED er koblet over spolen i reléet for å kortslutte “flyback” spenninger, som ellers ville kunne ødelegge transistoren. I vår løsning har vi valgt å benytte et eget 9V batteri til Arduino'en. Dette kan unngås ved at man benytter 5V fra fjernstyringsmottakeren og kobler den til 5V utgangen på Arduino'en. En slik løsning er noe ureglementert da spenningen går uregulert inn på Arduino-kortet og kan dermed skade kortet om spenningen blir for høy. Vi benytter en egen spenningskilde til tennsatsen da denne belaster batteriet temmelig hardt. På den måten unngår vi at et kraftig spenningsfall i forbindelse med terminering påvirker funksjonen til den øvrige elektronikken.

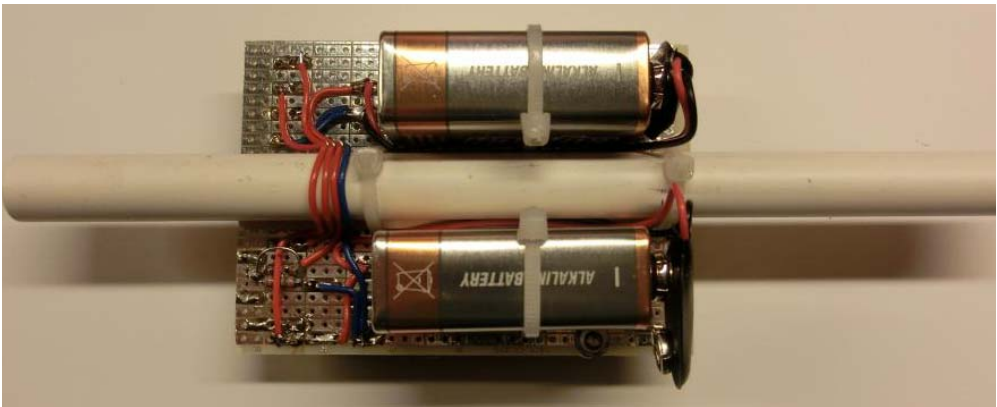
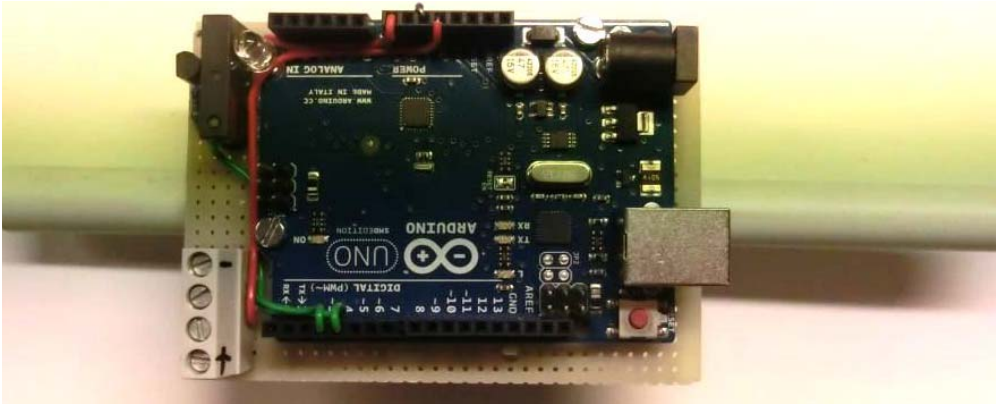


Figur 3.5 Oppkobling av termineringsmekanismen.

Arduino'en programmeres med programmet: *BallongSquid.ino*, se vedlegg A.3¹⁰.

Termineringskretsen ble festet til et elektriskrør slik at den kunne smettes inn på snøret til ballongen på oversiden av slippmekanismen som vist på de neste bildene.

10. BallongSquid.ino er skrevet av Thomas Gansmoe NAROM. Programlisten kan fås ved henvendelse til Thomas Gansmoe eller Nils Kr. Rossing



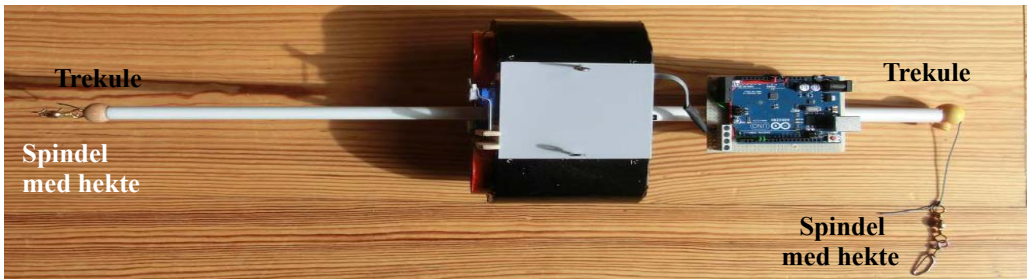
Figur 3.6 Termineringselektronikken montert på et elektriskerrør slik at den kan smettes inn på snora til ballongen.

Det hele monteres sammen med slippmekanismen som vist på figur 3.7.

For å lette sammenkoblingen av de ulike enhetene, kan man benytte en kort nylonstrå og kraftige spindler med hekter på over- og undersiden av elektronikken slik at de lett kan klipses inn på snora som forankrer ballongen.



Figur 3.7 Termineringselektronikken montert sammen med slippmekanismen.



Figur 3.8 Termineringselektronikken monteres på toppen av utløsningsmekanismen. En snor med trekuler og spindler i begge ender føres gjennom rørene som vist.

Som termineringsanordning ble benyttet tennsatsen til en rakettmotor. Denne ble tapet inntil ballonggummien ved innblåsningsrøret til ballongen, figur 3.9. I tillegg kan en benytte en mindre rakettmotor koblet til tennsatsen og legge det hele inne i ballongen før den fylles med gass. Dette vil sikre at ballongen blir terminert om nødvendig.



Figur 3.9 Tennsatsen tapes inntil ballongen.

Ballongen fylles med helium til den har et løft på mellom 1,2 – 1,5 kg. Dersom det er litt vind er det lurt å øke løfteevnen. Med større løft blir ballongen mindre følsom for avdrift Ballongen merkes med rød farge for lettere å bli sett.



Figur 3.10 CanSat'en er montert til slippmekanismen. En Cola-boks beskytter termineringsenheten.



En nylonsnor spunnet opp på et fiskehjul ble montert på en stor trekloss og benyttet som forankring.



Figur 3.11 Ballongen ble forankret i en nylonsnor spunnet opp på et fiskehjul (Hitra). Ballongen ble merket med rød oljebasert spraymaling.

Føringen for nylonsnoren bør monteres tett ved spolen slik at en unngår at snora faller av spolen og kilder seg fast i opplagringspunktet.

3.2 Bygg en egen slippmekanisme

Slippmekanismen består av en spesiallaget boks med plass til to CanSat'er med fallskjerm. Fallskjermene plasseres i to papprør, en på hver side av boksen som vist på figuren under. Øyeskruen hvor fallskjermen er festet, festes i mellomrommet mellom to lister som stikker ut over kanten av boksen.



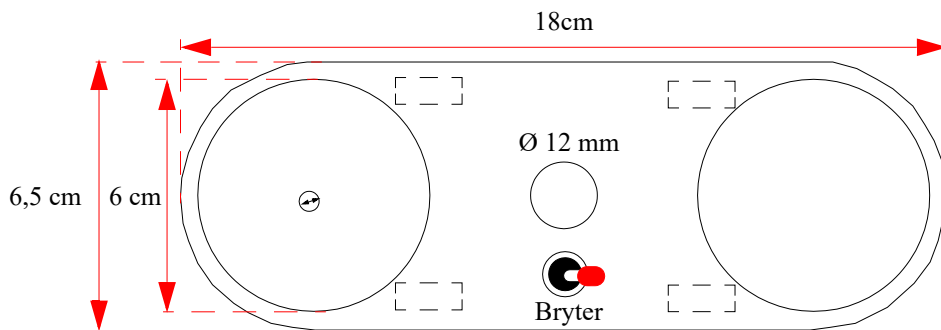
En servo skyver et stag gjennom to hull gjennom listene som vist til høyre på figuren over. Et elektrisk rør for snøret som holder ballongen går gjennom boksen.



Elektronikken, radiomottakeren og batteriet er plassert i rommet mellom de to papprørene.



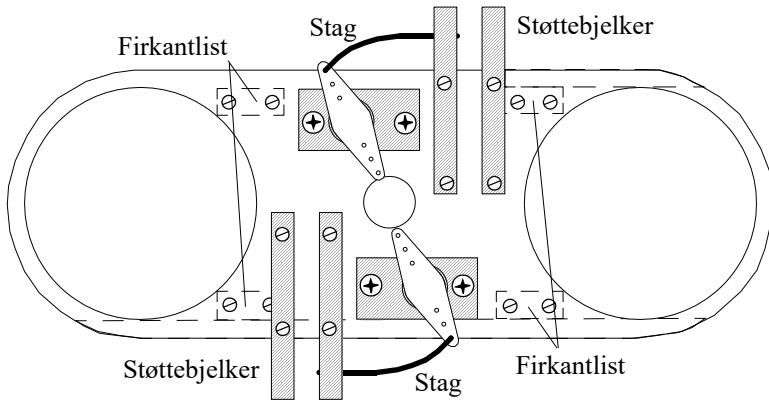
Figuren under viser en tegning av *topplata*. Topplata er laget av 3 mm kryssfiner. De to hullene dekkes til slutt med folie slik at de blir tette.



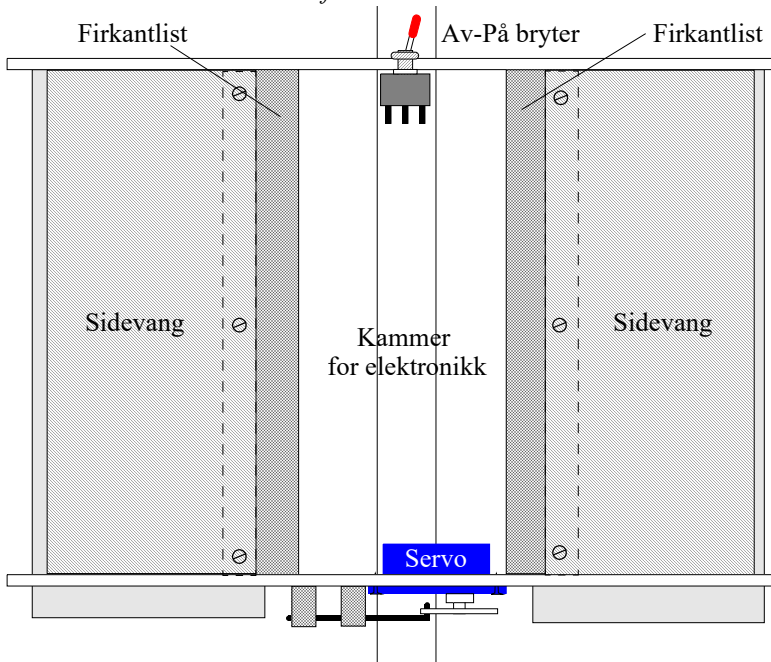
En på-av bryter monteres i topplata. Alternativt kandenne monteres i bunnplata, slik at den ikke er så utsatt for fuktighet.



Figuren under viser bunnplata som er laget på tilsvarende måte. I plata er det montert to *servoer* som drar i to *stag* laget av 1,5 mm ståltråd. Stagene går inn gjennom hull i to *støttebjelker* som er skrudd fast i bunnplata.



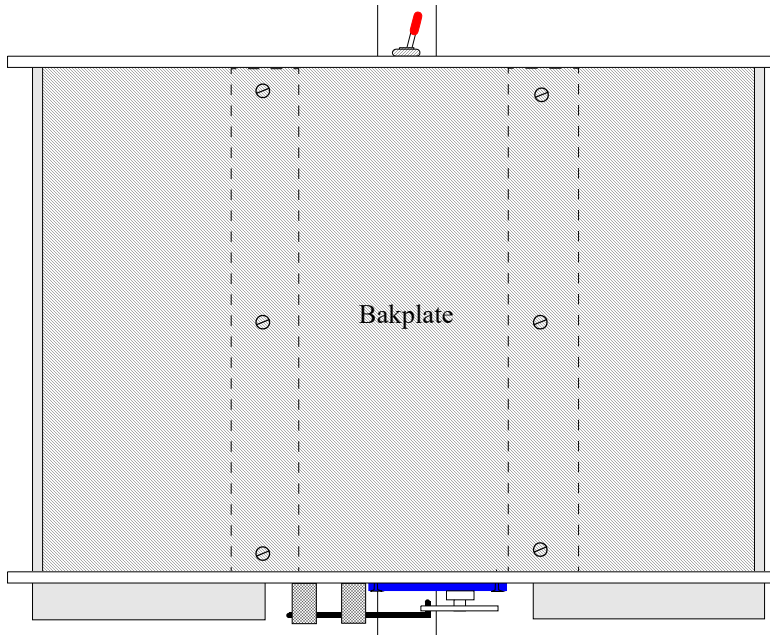
Papprørene går gjennom topp- og bunnplata. De kan også gjerne stikke 1 cm ut under bunnplata. De to platene er skrudd sammen med fire *firkantlister*.



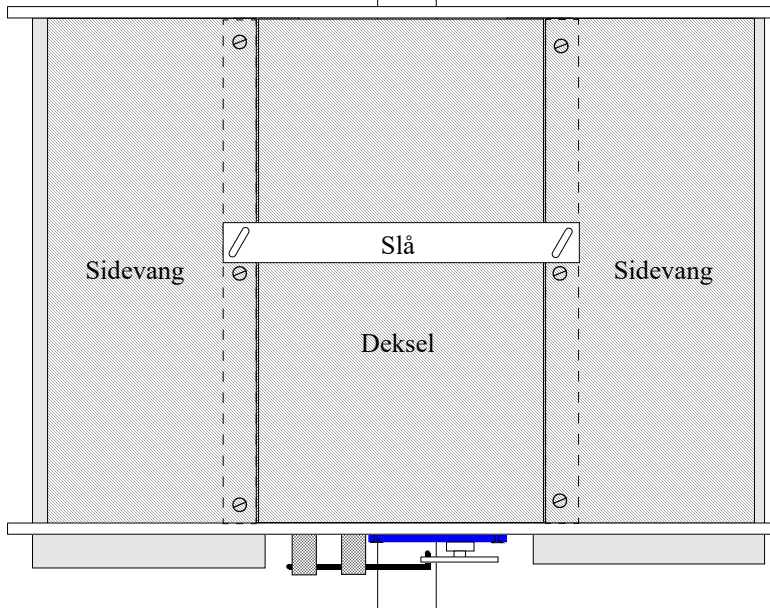
De to *sidevangene* er 3 mm kryssfiner som monteres en på hver side av åpningen inn til kammeret for elektronikken.



På baksiden er det montert et heldekkene *bakplate* som går ut til sidene der kromningen begynner, se figuren under.

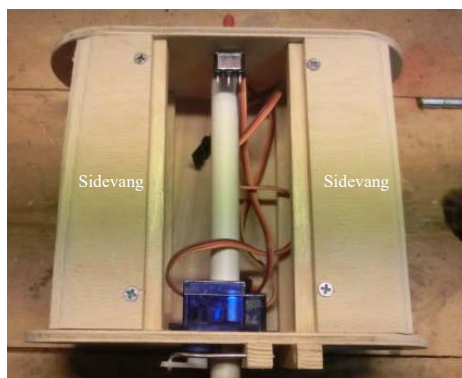
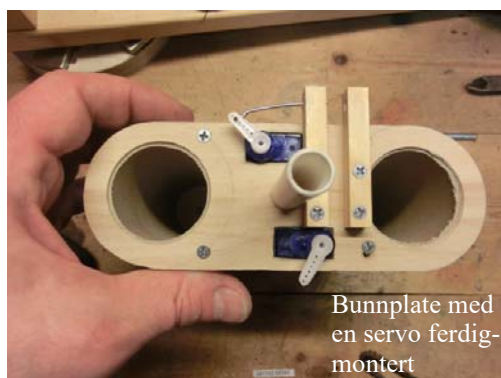


Et deksel i 3 mm finer dekker åpningen inn til kammeret med elektronikken. Dekselet kan enten festes med to eller fire øyeskruer, eller med en slå, se figuren under.





Bildene under viser slippmekanismen framstilt i tre.



Dekselet monter foran kammer for elektronikk



Tilslutt kles boksen utvendig med plastfolie eller laminert papir som stiftes fast i topp og bunnplatene.

Ellers brukes en standard fjernstyringsenhet med mottaker, mottakeren legges inn i kammeret og tilkobles batteriet. Servoene kobles til to av kanalene til mottakeren.

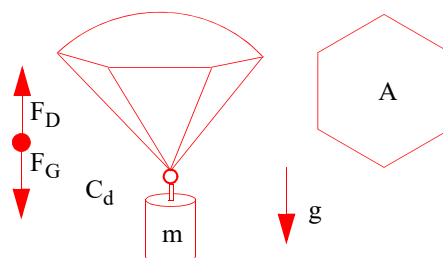
3.3 Beregning av fallhastighet

Fallhastigheten bestemmes av en rekke ulike parametere.

Vi antar at sonden inntar terminalhastigheten etter svært kort tid, slik at den faller med konstant fart omtrent fra slippstidspunktet. Når dette inntreffer vil *drag*'et, F_D , på grunn av luftmotstanden, være lik tyngdekrafta, F_G :

$$F_D = F_G \quad (3.1)$$

$$F_G = m \cdot g \quad (3.2)$$





Hvor:

$g =$ Tyngde akselerasjonen – 9,81 m/s²
 $m =$ Massen av CanSat – maks. 0,350 kg

En kan også finne en formel for *drag*'et, F_D :

$$F_D = 1/2 \cdot C_d \cdot \rho \cdot v_t^2 \cdot A \quad (3.3)$$

Hvor:

$C_d =$ *Drag*-faktor varierer blant annet med formen på fallskjermen (kule-kalott $C_d = 1,5$)

$\rho =$ Tettheten til luft (typisk $\rho = 1,22$ kg/m³) varierer med lufttrykket

$v_t =$ Terminalhastigheten

$A =$ Tverrsnittsarealet av fallskjermen projisert ned på horisontalplanet.

Setter ligning (3.2) og (3.3) inn i ligning (3.1) og løser denne mht. terminalhastigheten, får vi:

$$v = \sqrt{\frac{2mg}{C_d \rho A}} \quad (3.4)$$

Drag-faktoren er avhengig av formen på fallskjermen. Dersom den kan settes tilnærmet lik en kuleflate vil den i teorien ha en *drag*-faktor nær $C_d = 1,5$. Dersom fallskjermen tilnærmet er en sirkulær flate med samme tverrsnitt som kula, vil en C_d nær 0,75 være nærmere sannheten. En fallskjerm som vist på figuren over, vil ha en *drag*-faktor som ligge et sted mellom disse verdiene¹¹.

Bestem *drag*-faktoren:

Drag-faktoren for en gitt fallskjerm kan måles ved følgende metode:

1. Mål og beregn arealet, A , til fallskjermen (projisert areal)
2. CanSat'en henges under fallskjermen og finn massen, m
3. Slipp CanSat'en i fallskjerm fra en avsats fra 5–10 meter over bakken
4. Mål tida til fallet over en gitt fallhøyde. Eller film slippet med et kamera med et kjent antall bilder i sekundet. Plasser en målestav i bildet for å bestemme dimensjonene.
5. Løs ligningen (3.4) mht C_d . Sett inn måleverdier og beregn C_d .

11. <http://my.execpc.com/~culp/rockets/descent.html>



Beregning av fallskjermens areal:

For å finne det totale arealet for en fallskjerm, kan vi bruke illustrasjonen til høyre som viser en 6-kantet fallskjerm innskrevet i en sirkel.

Av figuren ser vi at fallskjermen består av 6 like trekkanter. Det betyr at det totale arealet for denne fallskjermen er $A = 6 \cdot A_T$ hvor A_T er arealet til hver av trekantene.

For å finne arealet for den enkelte trekanten, A_T , kan vi benytte formelen:

$$A_T = \frac{sh}{2} \quad (3.5)$$

Vi kan måle lengdene s og h direkte på fallskjermen. Det totale arealet av den 6-kantede fallskjermen blir dermed:

$$A = \frac{6sh}{2} \quad (3.6)$$

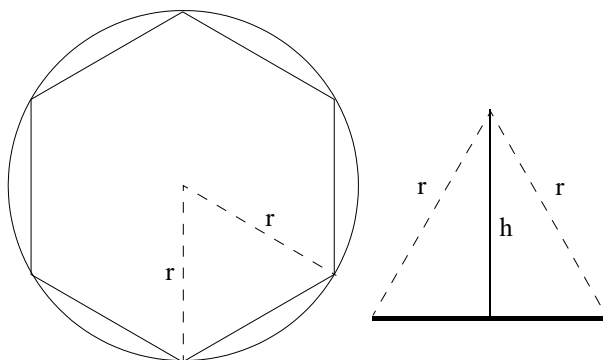
Det er mulig å finne et mer generelt uttrykk for beregning av arealet for en fallskjerm formet som en regulær mangekant med et vilkårlig antall kanter. For mer informasjon om beregning av arealet til fallskjerner, se:

<http://www.sunward1.com/imagespara/The%20Mathematics%20of%20Parachutes%28Rev2%29.pdf>

Når fallskjermens totale areal A er bestemt, kan fallhastigheten bestemmes av ligning: (3.4)

På kurset vil det bli gjennomført slipp av CanSat med fallskjerm uten modifikasjoner. Fallhastigheten kan dermed bestemmes dersom slipp høyden er kjent (for eksempel ved hjelp av data fra trykksensoren) og tiden CanSat'en bruker på fallet til bakken (hvordan kan falltiden bestemmes?). Ved å anta konstant fart, kan formelen $v = s/t$ brukes for å beregne fallhastigheten.

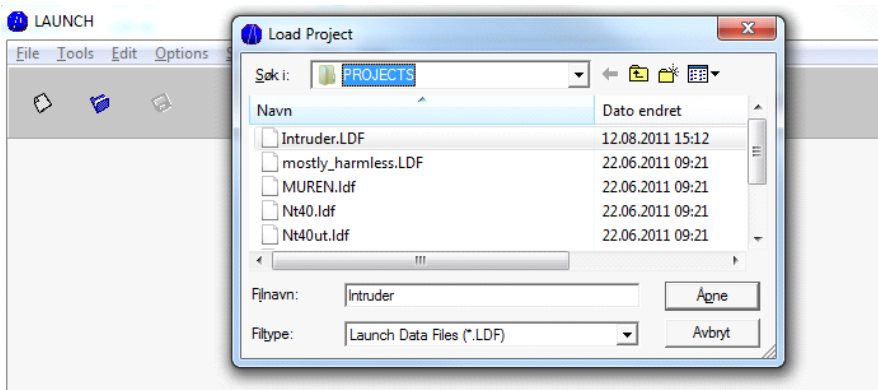
Sammenlign målingene av fallhastigheten med beregningene.





3.4 Simulering av CanSat dropp for å finne fallhastighet

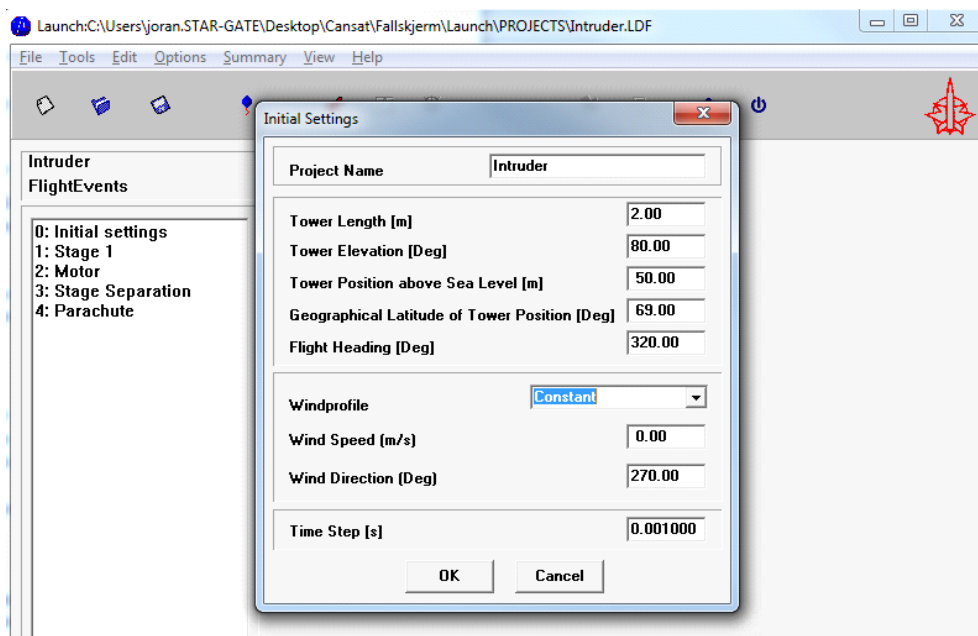
Her vil vi gi en kort introduksjon til simuleringsprogrammet *launch* som kan simulere slipp av en CanSat fra en rakett. Programvaren kan hentes på It's learning. Start programmet *Launch* og velg *Open*, som ligger under *File* menyen. Naviger til folderen *PROJECTS* og velg filen *Intruder.LDF*:



Figur 3.12 Åpne filen *Intruder.LDF*.



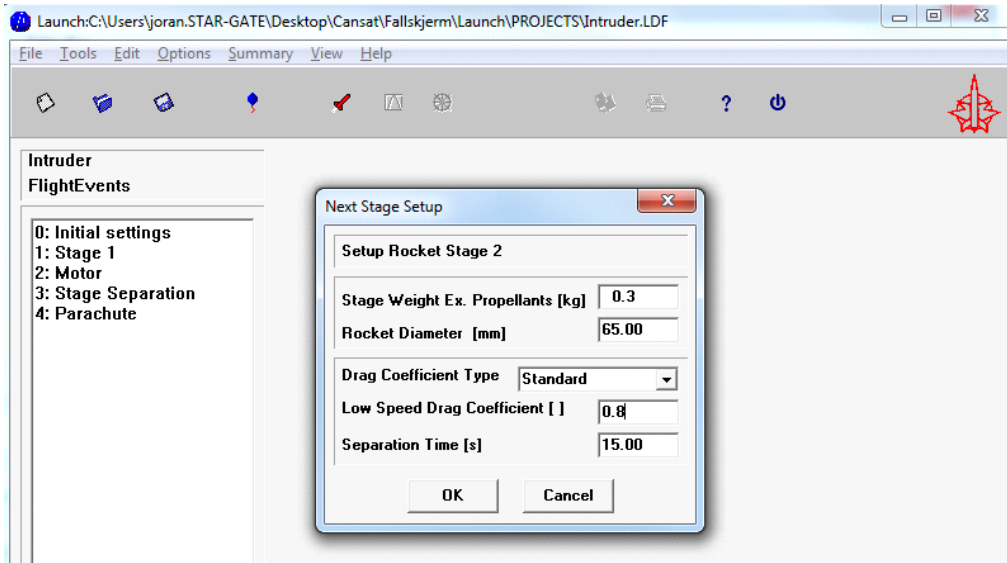
Et skjermbilde viser ulike *flights events* for *Intruder* raketten som NAROM bruker for å skyte opp CanSat'ene med. For denne laboratorieøvelsen er det av interesse å justere parametrene under *Initial settings*. Ved å klikke på *Initial settings* fås følgende skjermbilde opp:



Figur 3.13 “Initial settings”

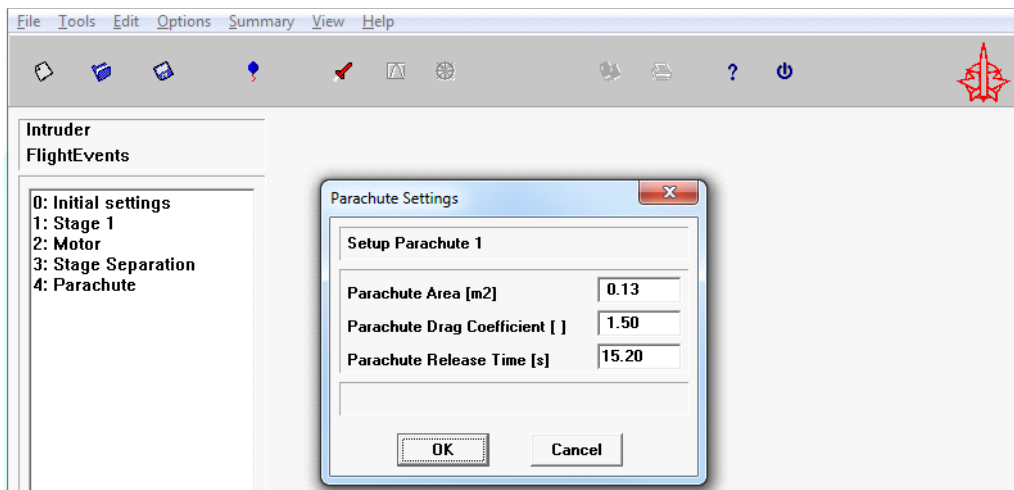
Under *Initial settings* er det mulig å justere plasseringen av utskytningsrampen. I vårt tilfelle gjør vi ingen endringer her. I tillegg er det mulig å endre vindprofilen ved å legge inn vindstyrke og vindretning. I denne omgang gjør vi ingen endringer her, men om ønskelig så kan man senere gå tilbake og endre på vindhastigheten og se hvilken innvirkning dette kan ha på CanSat'en. Nederst er angitt $Time Step = 0.001s$ som angir tidsintervallene for hver beregning i simuleringsprogrammet. Heller ikke denne trengs endres. Trykk OK til slutt.

Det neste som står oppført på listen er *Stage 1* og *Motor*. Disse to *FlightEvents* er satt av NAROM for å simulere selve *Intruder* raketten og skal ikke endres. Gå derfor videre og klikk på *Stage Separation*.



Figur 3.14 "Stage Separation"

Under *Stage Separation* skal parametere for selve CanSat'en legges inn. Her angis vekten og diameteren til CanSat'en. I dette tilfellet er det lagt inn en vekt på 0,3 kg og en diameter på 65 mm. Under *Low Speed Drag Coefficient* angis *drag*-koeffisienten som i dette tilfellet er 0,8. Separasjonstiden angir når CanSat'en skal komme ut av Intruder raketten. Separasjonstiden er ca. 15 sekunder. Klikk deretter OK.



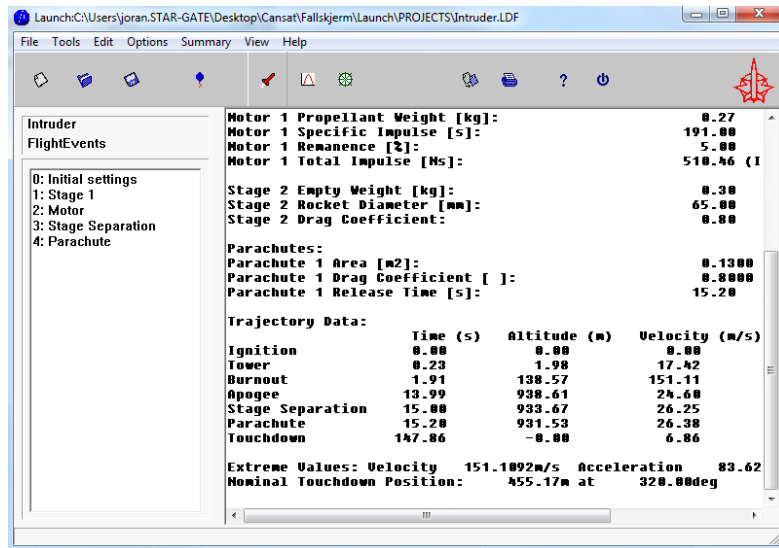
Figur 3.15 Parachute setting



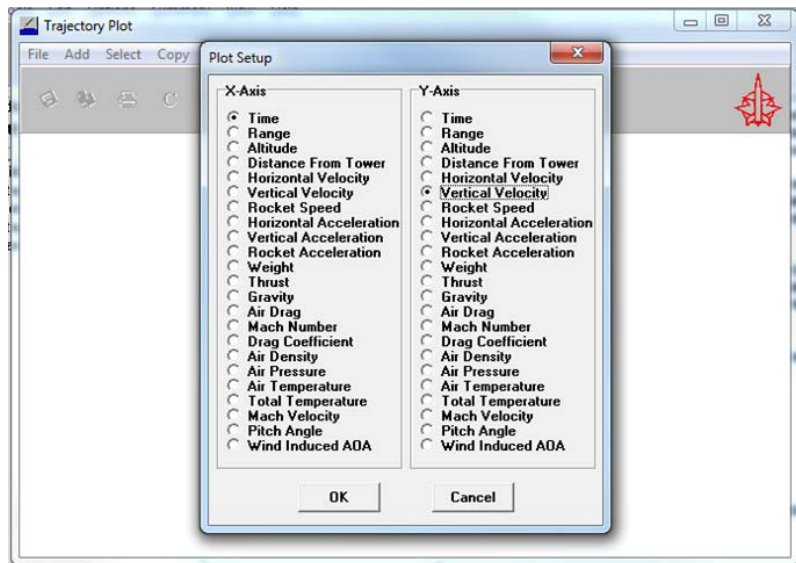
Under *Parachute settings* velges det totale arealet til fallskjermen, som i dette tilfellet antas å være $0,13 \text{ m}^2$. I tillegg legges drag-koeffisienten til fallskjermen inn på nytt lik 0,8. I simuleringsprogrammet antar vi at fallskjermen folder seg ut rett etter at CanSat'en er kommet ut av raketten (eller er sluppet fra ballongen). Vi setter derfor utløsningen av fallskjermen til å være 15,2 s. Velg deretter OK.

Vi er nå klare til å starte simuleringen med de parameterne som er matet inn. Klikk på det røde symbolet som er formet som en rakett. Simuleringen skal gå raskt og vi vil få opp et skjermbilde som vist på figuren under til høyre.

I figur 3.16 ser vi en del parametere, blant annet *Touchen* som i dette tilfellet viser en hastighet på $6,86 \text{ m/s}$. Dette vil være terminalhastigheten for CanSat'en. Denne verdien kan sammenlignes med tidligere beregninger. Vi kan få mer informasjon ved å klikke på *Trajectory plot* på siden av *Run calculation*.



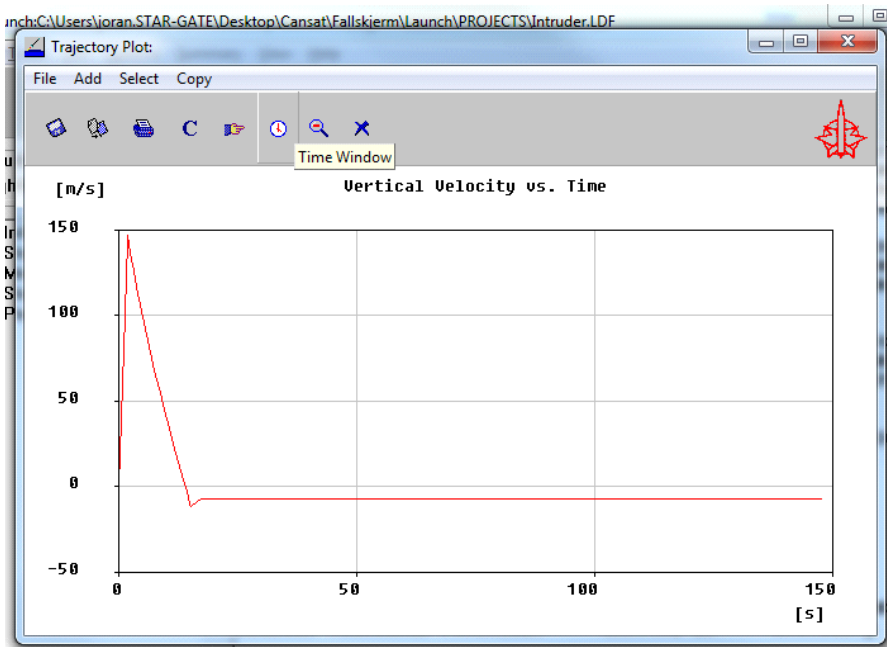
Figur 3.16 "Run Calculation"



Figur 3.17 "Trajectory plot"



Her er det mulig å hente frem mange forskjellige plot (se figur 3.17). Vi er imidlertid mest interessert i fallhastigheten, dvs. vertikal hastighet. Velg *Time* for x-aksen og *Vertical Velocity* for y-aksen og klikk OK, og følgende plott vil vises (se figur 3.18):



Figur 3.18 Vertikal hastighet som funksjon av tiden

De første 15 sekundene viser hastigheten til raketten, mens alt etter dette viser hastigheten til Can-Sat'en som faller i fallskjerm. Vi kan zoomme inn på deler av plottet ved å klikke på *Time Window*.

Merk at det også er mulig å eksportere data fra dette programmet ved å velge *Export data* fra *File* menyen. Da lagres resultatet som en tekstfil som for eksempel kan importeres i Excel for å plote resultatene.

Til slutt kan det nevnes at dersom vi verken kjenner fallhastigheten eller dragkoeffisienten til fallskjermen, så er det nødvendig å gjennomføre dropp-tester for å bestemme terminalhastigheten. Dette gjelder spesielt dersom man skal lage en egen fallskjerm eller gjøre modifikasjoner på den fallskjermen som følger med kit'et. En måte å gjøre dette på er ved å gjennomføre flere dropp fra for eksempel en bygning med en kjent høyde over bakken. Noter falltiden over et gitt antall dropp og beregn en gjennomsnittlig fallhastighet med formelen vist tidligere, $v = s/t$. Merk at vi antar at fallhastigheten er konstant. Men vil den egentlig være det under et virkelig slipp?





4 Programmering

CanSat programmeres i C. I dette kapitlet skal vi først installere program-editoren og vise hvordan den brukes. Dernest skal vi ganske kort foreslå en måte å strukturere CanSat-programmet på, for deretter å gi en oversikt over noen viktige og anvendelige kommandoer.

4.1 Installasjon av programvare

Det er en programpakke som skal installeres:

- Arduino programeditor

4.1.1 Arduino programeditor

Litt historikk

Arduino er et mikrokontrollerkonsept utviklet i den vesle italienske byen *Ivrea* på begynnelsen av dette årtusen. Hensikten var å lage et kontrollerkort som skulle gjøre det enklere og billigere for studenter å lære seg bruk av mikrokontrollere. Det første Arduino-produktet ble utviklet av grunnleggerne *Massimo Banzi* og *David Cuartielles*. De oppkalte prosjektet etter *Arduin of Ivrea* som var den vesle byens historiske heltefigur. Navnet betyr *sterk venn* og burde passe godt for et kraftig kontrollerkort. Det som startet som et lokalt prosjekt for studenter i 2005, hadde i 2010 spredd seg til hele verden. I februar 2010 hadde de solgt 120 000 eksemplarer. Den tilhørende programvaren ble utviklet av studenten *Hernando Barragán* ved det lokale universitetet i Ivrea, som en “open source” kode.

Kortet var bygget opp omkring *AVR mikrokontrollere* fra Atmel (hovedsakelig ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560). Kontrollkortet i CanSat anvender ATmega168. Dette er en serie kontrollere som anvender RISK-arkitektur, en svært enkel, men meget effektiv arkitektur. Det er moro å vite at den første kontrolleren i denne serien ble utviklet av studentene *Alf-Egil Bogen* og *Vegard Wollan* ved NTH på begynnelsen av 1990-tallet. Etter endt studium tok de med seg konseptet inn i firmaet Nordic VLSI (nå NORDIC Semiconductor), hvor det ble videreutviklet. I 1995 gikk de ut av Nordic VLSI og ble snart kjøpt opp av Atmel og driver i dag firmaet Atmel Norge. De sier selv at AVR ikke har noen spesiell betydning, men det er allment akseptert at det opprinnelig sto for *Alf* (Egil Bogen) and *Vegard* (Wollan) 's Risc processor.

Nedlasting av programvare

Arduino-editoren og kompilatoren kort sagt det som trengs hentes fra:

<http://arduino.cc/hu/Main/Software>

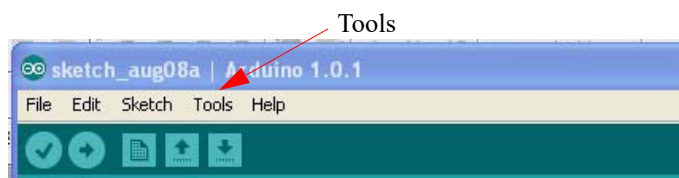
Aktuell versjon av Arduino er Arduino 1.0.1. Filen som har navnet *arduino-1.0.1-windows* er pakket som en zip-fil og er på ca. 86 Mbyte. En tilsvarende fil er tilgjengelig for Mac fra samme nettsted.



Installasjon av programvaren:


1. Klikk på fila **arduino-1.0.1.zip**
2. For å pakke ut fila trenger du programmet WinRAR som kan hentes fra: <http://www.rar-lab.com/download.htm> eller Winzip som kan hentes fra: www.winzip.com/downwz.htm
3. Velg *Extract to* fra menylinjen øverst og velg f.eks. C:/Programfiler og trykk OK. Programfilene legges da i en egen underkatalog (arduino-1.0.1) i katalogen *Programfiler*.
4. Programmet startes ved å klikke på programikonet:  .
5. Koble til USB-kabelen til ønsket port.

6. Klikk på *Tool* på menylinjen og velg *Board*. Her velges hvilken variant i Arduino familien du skal jobbe med. I dette tilfellet velg: *Arduion UNO...*









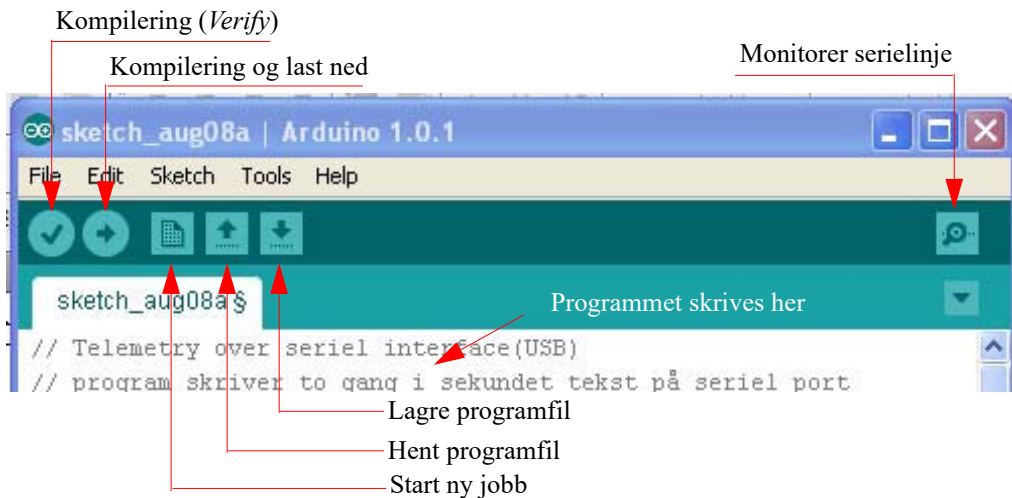
7. Klikk på *Tool* på menylinjen og velg *Serial Port*. Sjekk at riktig port (Com?) er valgt.

Programmet er klart til bruk og du kan skrive inn programlinjene. Når programmet er ferdig skrevet, skal det *kompileres*, dvs. overføres til en binærkode som mikrokontrolleren forstår. Dersom programmet inneholder ulovlige kommandoer eller skrivefeil, så vil kompilatoren varsle om det og vise på hvilken linje feilen er avslørt. Det er ikke nødvendigvis alltid der feilen er gjort.

Dernest skal programmet lastes ned til kontrollenhetens minne (Arduino-kortet). Dette gjøres ved å trykke på knappen  .

Kort oversikt over Arduino-editoren

-  Verifiser at koden er riktig
-  Kompiler og last ned programmet til kontrollenheten
-  Hent nytt "arbeidsark"
-  Hent en eksisterende programfil
-  Lagre programfil
-  Monitorer data sendt tilbake på serielinjen



Manglende kontakt med kortet

Det hender at en ikke oppnår umiddelbar kontakt med Arduino-kortet når en forsøker å laste ned et program. Feilmeldingen: **avrduede: stk500_getsync(): not in sync: resp=0x00** i meldingsvinduet betyr at det ikke oppnås kontakt med kortet. Dette kan skyldes flere ting:

- Kabelen ikke tilkoblet, eller ødelagt
- Feil port er valgt av programeditoren
Endres ved å velge: *Tool* og *Serial Port* fra menylinjen i editoren
- Feil type kontroller-kort valgt
Endres ved å velge: *Tool* og *Board* fra menylinjen for så å velge rett kort, i vårt tilfelle *Arduino UNO*.
- Rx/Tx linjene har en tilleggsfunksjon som ikke er frakoblet under programmeringen. F.eks. ved at datainnsamlingsenheten ikke er frakoblet, eller at strappene J2 og J3 ikke er fjernet under programmeringen.
- Manglende driver

Ved installasjon av drivere gjøres følgende (Windows 7 - tilsvarende kan gjøres for XP):

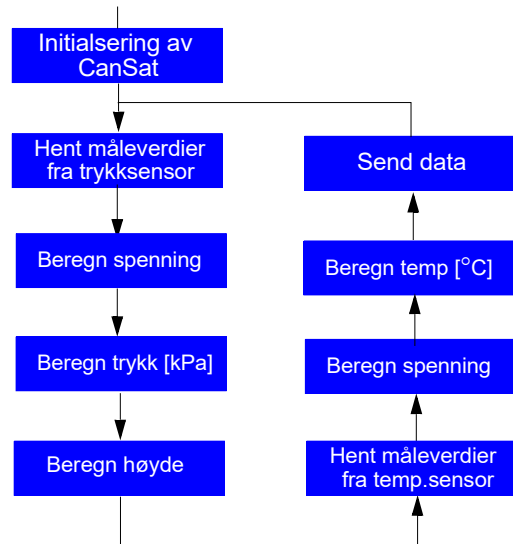
- Åpne kontrollpanelet
 - Velg: *System og sikkerhet*
 - Velg: *System*
 - Velg: *Enhetsbehandling* (venstre meny)
 - Velg: *Andre enheter* eller *Porter* og finn den porten som Arduino er tilkoblet
 - Høyreklikk på porten og velg *Oppdater driverprogramvare* fra menyen
 - Velg: *Søk på datamaskin etter driverprogramvare*
 - Bla gjennom og finn katalogen *Arduino 1.0.1/Drivere* - pek på den og start installasjon



Om du er heldig installeres driveren og du får kontakt med Arduino-kortet.

4.2 Programstruktur

Databehandlingsenheten (Arduino UNO) styrer hele prosessen i CanSat'en. Den henter inn data fra datainnsamlingsenheten (*shield* kortet), omregner fra tallverdi til trykk, temperatur og høyde, tilrettelegger og sørger for å sende data til bakkestasjonen via kommunikasjonsenheten, eller til datalagringsenheten (Openlog). Dette gjøres i en gjentakende sløyfe som vist på figuren til høyre. En slik oversikt over programmet kalles et *flydiagram*.



4.3 Viktige kommandoer

Referansemanualen til C for bruk ved programmering av Arduino-prosessorer finnes på følgende nettside:

<http://arduino.cc/en/Reference/HomePage>

Referanse manualen ligger også under *Help* på menylinjen i programeditoren som vist på figuren til høyre.



4.3.1 Generelle kommandoer

Programstruktur

Programmet består av en rekke mindre rutiner omsluttet av klammeparenteser. I `void setup()` rutinen initieres mikrokontrolleren, mens selve programmet legges under `void loop()` rutinen.

```
void setup()
{
  <initiering>
}
void loop()
{
  <programkode>
}
```




```
}
```

Alle kommandoer må avsluttes med; (semikolon)

Initiering av dataoverføring til PC

Under uttestingen kan det være praktisk at data leses tilbake til terminalen. Datahastigheten settes opp i setup-rutinen med kommandoen: `Serial.begin(9600)`; her satt til 9 600 baud:

```
void setup()
{
  Serial.begin(9600);
}
```

Kommentarer:

Kommentarer kan skrives hvor som helst og begynner med

```
// Dette er en kommentar
```

Disse blir fjernet under kompilering og overføres ikke til mikrokontrolleren.

Deklarasjon av variable:

I C må alle variable deklarerer før de kan brukes og gjerne i starten av programmet. Deklarasjonene må inneholde *type* og *navn* på variabelen:

Deklarering kan også gjøres innenfor hver subrutine. Slike variable gjelder da bare innenfor den rutinen de er definert:

```
void loop()
{
  int a;           // deklarasjon av 16 bit heltall (word)
  char b;         // deklarasjon av 8 bit karakter (byte)
  char c, d;      // deklarasjon av to 8 bits karakterer (byte)
  float e;        // deklarasjon av variabelen e som et desimaltall f.eks. 1,65 (32 bit, dobbel word)
  unsigned long f; // deklarasjon av 4 byts heltallsvariabel f (32 bit) uten fortegn
  boolean g;      // deklarasjon av en boolsk variabel g som kan ha verdiene 0 og 1
  <programkode>
}
```

Skriv tilbake til PC skjerm:

Følgende kommandoer skriver en variabel eller en tekst tilbake på terminalvinduet i programeditoren.

```
Serial.print(a); // Skriver variabelen a til en linje på skjermen,
                // neste skrivekommando skriver på samme linje
Serial.println(a); // Skriver variabelen a til en linje på skjermen,
                // neste skrivekommando skriver på ny linje
Serial.println("Hallo"); // Skriver teksten Hallo til en linje på skjermen,
                // neste skrivekommando skriver på ny linje
```



Det er også mulig å kombinere tekst og variable i samme printkommando:

```
Serial.println("Trykk:", a);    // Skriver teksten Trykk: til en linje på Arduino monitoren,  
                               // etterfulgt av innholdet i variabelen a, skifter deretter til ny linje  
  
Serial.println(f, 2);          // Skriver desimalvariabelen f til terminal på PC med to desimaler  
  
Dersomman ønsker å skrive rådata til monitoren, tilen file på SD-kortet eller til radioen kan det være greit å  
skille de ulike variable fra hverandre. I tillegg kan det være lurt å legge inn en signatur i starten og et tall  
som angir tid fra oppstart. Som for eksempel:  
  
Serial.print("Signatur: ");  
Serial.print(millis());        // Skriver ut antall millisekunder fra at Arduinoen ble startet  
Serial.print(" ");  
Serial.print(temperatur_LM35_raadata);  
Serial.print(" ");  
Serial.print(temperatur_NTC_raadata);  
Serial.print(" ");  
Serial.print(trykk_raadata);  
Serial.print(" ");  
Serial.print(akselerator_x_raadata);  
Serial.print(" ");  
Serial.print(akselerator_y_raadata);  
Serial.print(" ");  
Serial.print(akselerator_z_raadata);
```

Definer digitale porter som inngang eller utgang:

Kontrolleren ATmega 328 har en rekke porter, digitale og analoge. De digitale portene må defineres som inn- eller utgang. Dette gjøres i setup-rutinen:

```
void setup()  
{  
    pinMode(8,OUTPUT);          // Definerer pinne 8 som utgang, dette gjøres under setup  
    pinMode(7,INPUT);          // Definerer pinne 7 som inngang, dette gjøres under setup  
    pinMode(6,INPUT_PULLUP);   // Definerer pinne 6 som inngang med pullup motstand, dette er nyttig  
                               // for at inngangen ikke skal henge fritt (sveve)  
                               // når den ikke er tilkoblet noe  
}
```

Lese og skriv til en digital port:

Digitale porter kan enten settes til høy eller lav spenning. Dette gjøres med følgende kommandoer:

```
boolean bolsk;                // Definerer den boolske variabelen bolsk kan ha verdien 0 eller 1  
  
void loop()  
{  
    digitalWrite(8, HIGH);     //Setter port 8 høy (5 V)  
    digitalWrite(8, LOW);      //Setter port 8 lav (0 V)  
    bolsk = digitalRead(7);     // Leser den digitale verdien på port 7 og setter i variabelen bolsk  
}
```



Vent-kommando:

Dersom vi ønsker at programmet skal ta en pause kan vi skrive følgende:

```
delay(1000);           //Stopper programmet i 1000 msek (1 sek)
```

Aritmetiske operasjoner

```
sum = a + b;           // Summen av  $a + b$  settes i variabelen sum  
diff = a - b;         // Differansen av  $a - b$  settes i variabelen diff  
prod = a * b;         // Produktet av  $a * b$  settes i variabelen prod  
kvo = a / b;          // Koeffisienten av  $a / b$  settes i variabelen kvo
```

4.3.2 Avlesning av sensorer

Dette avsnittet oppsummerer noen viktige kommandoer for innhenting av data fra sensorene.

Tabellen under viser hvilke sensorer som er tilkoblet de ulike analoge portene på Arduino-kortet:

Kontakt	Måling	Sensor	Område	Analog kanal
H1-1	Temperatur	LM35DZ	0 - 5,0 V	5
H1-2	Akselerometer z-akse	MMA7361L	0 - 3,3 V	4
H1-3	Akselerometer y-akse	MMA7361L	0 - 3,3 V	3
H1-4	Akselerometer x-akse	MMA7361L	0 - 3,3 V	2
H1-5	Trykk	MPX4115	0 - 5,0 V	1
H1-6	Temperatur (NTC)	NTCLE100E3103JB	0 - 5,0 V	0

Les verdi fra AD-konverter

Syntaksen for lesing fra en AD-konverter inngang kan skrives som:

```
<variabel> = analogRead(<kanal>); //Kanal kan ha verdiene 0 til 5
```

Eksempel 1:

```
int VERDI;  
VERDI = analogRead(0); //Digitale verdien fra AD-kanal 0 leses inn i verdi
```

Eksempel 2:

```
void loop()  
{  
    int pressure; //Deklarerer pressure som en heltalls-variabel  
    pressure = analogRead(1); //Leser av trykksensoren på AD-kanal 1  
    Serial.println(pressure); //Skriv resultatet tilbake til terminalen (PC)  
}
```

Eksempel 3:



```
void loop()
{
  int temperature;           //Deklarerer temperatur som heltallsvariabel
  temperature = analogRead(5); //Leser av temperatursensoren på AD-kanal 5
  Serial.println(temperature,2); //Skriv resultatet tilbake til Arduino monitoren (PC) med 2 desimaler
}
```

Forandring av referansen til AD-konverteren

Under vanlige betingelser vil temperaturen variere mellom 5 – 40 °C. Vi legger merke til at innen dette temperaturområdet vil temperatursensoren LM35 kun bruke en liten del av området til AD-konverteren. Med en følsomhet på 10mV/K og en oppløsning hos AD-konverteren på 4,89 mV er det vanskelig å oppnå en nøyaktighet bedre enn ca. 0,5 °C. Siden temperaturkurven fra LM35 i teorien skal gå gjennom null (0V ved 0°C) så vil variasjonen i det aktuelle området være fra typisk 50mV til 400 mV. Dvs. at vi kun får utnyttet under 10 % av AD-konverterens totale område (dynamiske område).

```
analogReference (INTERNAL);
```

Denne kommandoen setter AD-konverterens referansespenning til 1,1 V i stedet for 5 V. Dermed fordeles de 1024 nivåene over spenningen 0–1,1 V og oppløsningen i spenning blir ca. 1,07 mV hvilket gjør at temperaturopløsningen hos LM35 blir ca. 0,11 °C.

En må imidlertid være klar over at man kan ha behov for å endre referansespenningen i programet dersom man har andre sensorer som trenger 5V. Det er også noe uklart om en AD-konverter som har en referansespenning på 1,1 V tåler spenninger opp til 5 V.

Omregning fra digital verdi til desimal spenning og trykk

```
void loop() {
  int pressure;           //Deklarerer pressure som en heltalls-variabel
  float voltage;         //Deklarerer voltage som en desimal-variabel
  float kpa;             //Deklarerer trykk i kPa som en desimalverdi
  pressure = analogRead(1); //Leser av AD-kanal 1 hvor trykksensoren leverer signalet sitt
  voltage = (float)pressure/1023.0 * 5.0; //Omregning fra digital til desimal spenningsverdi
  // (float) kommandoen konverterer utregningen til desimaltall
  kpa = 22.222 * voltage + 10.556; //Omregning fra desimal spenning til desimalt trykk
  Serial.println(kpa,2); //Skriv resultatet tilbake til terminalen (PC) med 2 desimaler
}
```

Omregning fra digital verdi til desimal spenning og temperatur i Celsius

```
void loop()
{
  int temperature;           //Deklarerer temperatur som heltallsvariabel
  float voltage;           //Deklarerer voltage som desimalverdi
  float celcius;           //Deklarerer voltage som desimalverdi
  temperature = analogRead(5); //Leser av temperatursensoren på AD-kanal 5
}
```



```
voltage = (float)temperature/1023.0 * 5.0; //Omregning fra digital til desimal spenningsverdi,
//(float) kommandoen konverterer utregningen til desimaltall
celcius = voltage * 100.0; //Omregning fra spenning til Celsius
Serial.println(celcius,2); //Skriv resultatet tilbake til terminalen (PC) med 2 desimaler
}
```

4.3.3 Programmering av kommunikasjonsenheten APC220

Siden DRx og DTx på kommunikasjonsenheten APC220 er koblet til seriekommunikasjonslinjen, vil alt som skrives til serieporten både sendes til USB-inngangen på PC også sendes til kommunikasjonsenheten. Denne er helt transparent slik at den kan behandles likt med serielinjen som går i kabel.

```
void setup
{
  Serial.begin(9600); //9600bps is default for OpenLog
}
void loop
{
  Serial.println("123");
  delay(500);
}
```

Programmet over vil sende *123* i ASCII-kode til senderen. Dersom en tilsvarende mottakeren er koblet opp på USB inngangen på PCen og Arduino monitoren er knyttet opp mot denne USB-porten, vil det skrives *123* i monitorvinduet på skjermen.

Konfigurering av APC220 fra Arduino'en

Som vi har sett så kan kommunikasjonsenheten konfigureres ved hjelp av spesiell programvare (se side 29).

4.3.4 Skriv data til Openlog

Forenklet datalagring

Dersom Tx på Arduino UNO er tilknyttet RXI på Openlog, og Rx på Arduino UNO er tilkoblet TXO på Openlog, kan en skrive til loggeren ved hjelp av kommandoene *print()* og *println()* som vist i eksempelet under. Pass på at dataene skrives på ASCII-form og ikke binært. Dermed kan dataene lett leses med en vanlig teksteditor.

```
void setup
{
  Serial.begin(9600); // Setter dataratene til 9600 bps (er også default)
}
void loop
{
  Serial.println("123");
  delay(500);
}
```



Dette enkle programmet skriver tallene 123 inn i en fil. *Ny fil opprettes automatisk hver gang Openlog slås på.* Filene får navn ala *LOG0000n.txt*, hvor n er et løpende nummer som går fra 0000.

Ved større datamengder og stor hastighet kan det være nødvendig å legge inn et lite delay etter `Serial.print` kommandoen:

```
void setup
{
  Serial.begin(115200);    //Sett dataraten til 115 200 bps
}
void loop
{
  for(int i = 1 ; i < 10 ; i++) {
    Serial.print(i, DEC);
    Serial.println(":abcdefghijklmnopqrstuvwxyz-!#"); // Skriv 10 ASCII strenger til Open log
    delay(15);
  }
}
```

Følgende fil skriver verdier for variablene spenning, trykk og temperatur til serieporten. Dersom Openlog og/eller radiolinken er tilkoblet, skrives det samme til disse.

```
char buff[50];
int ledPin = 13;
float Voltage = 3.5134;
float Trykk = 100.650;
int Temp = 22;
void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  Serial.print("Spenning = ");
  Serial.print(Voltage,2);
  Serial.print(" V, ");
  Serial.print("Trykk = ");
  Serial.print(Trykk,3);
  Serial.print(" kPa, ");
  sprintf(buff,"Temperatur = %01d", Temp); //Skriver streng inn i buff - alternativ måte
  Serial.println(buff); //Skriver strengen buff til serieport/Openlog

  digitalWrite(ledPin, HIGH); //Slår av og på lysdiode
  delay(1000);
  digitalWrite(ledPin, LOW);
}
```



```
delay(1000);  
}
```

Avansert datalagring

Noen ganger ønsker en å utføre mer avanserte funksjoner som f.eks. å opprette filer med spesifiserte navn, ev. slette andre filer i datalagringsenheten (Openlog). For å utføre slike kommandoer må man sette openlog i kommandomodus. Men før vi ser på hvordan vi gjør det, la oss se på noen kommandoer som det kan være nyttig å kjenne til:

new *filnavn*

Oppretter en ny fil med navnet *filnavn*. Filnavnet skal være av formen 8.3 (f.eks. **new** *filename.txt*). Etterfølgende data skrives inn i filen.

append *filnavn*

Legger inn nye data på slutten av en tidligere opprettet fil, gjerne etter at programmet har vært stoppet og startet på nytt (f.eks. **append** *filename.txt*). Dersom en ønsker å avslutte denne operasjonen kan en sende kommandoen ctrl+z (ASCII 26).

write *filnavn offset*

Denne kommandoen gir mulighet til å skrive inn i filen *filename* fra en gitt posisjon, *offset*. F.eks. vil kommandoen **write** *filename.txt 516*, starte skrivingen ved byte nr. 516 ved neste *serial.print* kommando.

rm *filnavn*

Denne kommandoen fjerner filen med navnet *filnavn* fra Openlog.

I tillegg finnes en rekke andre kontrollkommandoer. Disse er omtalt bl.a. på følgende nettside:

<https://github.com/sparkfun/OpenLog/wiki/Command-Set>



For å anvende slike kommandoer må vi sette Openlog i kommandomodus. Flytdiagrammet til høyre viser hvordan dette gjøres:

Reset Openlog:

Openlog resettes ved å legge pinne GRN lav i 100 ms. GRN er koblet til D2 (digital port 2) på Arduino-kortet. *Reset* gjøres med følgende kommando:

```
digitalWrite(2, LOW);  
delay(100);  
digitalWrite(2, HIGH);
```

Vent til Openlog svarer:

Ved *reset* eller overføring av en hvilken som helst annen kommando, må en vente på at Openlog svarer med "<". Denne ventefunksjonen kan utføres på følgende måte:

```
while(1)  
{  
  if(Serial.available())  
    if(Serial.read() == '<') break;  
}
```

while(1) rutinen vil forsette i det uendelige. Den eneste måten å komme ut av *while*-rutinen, er ved å nå fram til et *break*. Dette oppnås i dette tilfellet dersom begge *if*-betingelsene er oppfylt. *Serial.available()* angir hvor mange data som ligger klart i utbufferet hos Openlog, dersom dette er forskjellig fra 0 leses innholdet i bufferet med kommandoen *Serial.read()*. Om den leste karakteren er "<" er betingelsen gyldig og programmet bryter ut av *while*-rutinen.

Sett Openlog i kommandomodus:

For å sette Openlog i kommandomodus sendes tre ganger *ctrl-z* etter hverandre. *Ctrl-z* har ASCII-verdien 26.

```
Serial.write(26);  
Serial.write(26);  
Serial.write(26);
```

Deretter må en vente på at Openlog svarer med "<".

Send kommandoer til Openlog:

Derneft sendes den ønskede kommandoen. I sin enkleste form kan den se slik ut dersom vi ønsker å opprette en ny fil med navnet *CanSat.txt*":

```
Serial.println("new CanSat.txt");
```

Deretter må en som vanlig vente på at Openlog svarer med "<".





Normalt vil en ikke bare åpne en ny fil, en vil også skrive til den, ikke bare en gang, men fortløpende legge til data etter som disse kommer inn. Da kan en f.eks. benytte append kommandoen:

```
Serial.println("append CanSat.txt");
```

for så å vente på at Openlog svarer med "<".

Skriv data til Openlog:

Skriving av data gjøres på vanlig måte med følgende kommando:

```
Serial.println("...data");
```

Dataene skrives på ASCII-form, dvs. i hermetegn.

En komplett rutine for skriving er vist i vedlegg A.1 på side 205 som er hentet fra nettstedet:

https://github.com/sparkfun/OpenLog/blob/master/OpenLog_CommandTest/OpenLog_CommandTest.ino

og kode for skriving til Openlog finnes på:

https://github.com/sparkfun/OpenLog/blob/master/OpenLog_Test_Sketch/OpenLog_Test_Sketch.ino

og for å skrive og lese fra fil i Openlog:

https://github.com/sparkfun/OpenLog/blob/master/OpenLog_ReadExample/OpenLog_ReadExample.ino

4.3.5 Lagring av data fra GPS-modul GY-NEO6MV2

For omtale av GPS-kretsen og oppkobling se avsnitt 7.12.2 på side 184.

I dette avsnittet skal vi hente inn data fra GPS-enheten GY-NEO6MV2 og lagre disse på SD-kortet som er på CanSat shield-kortet.

Installasjon av bibliotek

Biblioteket *TinyGPS* må installeres for å kunne hente ut data fra GPS-kontrollkortet. I tillegg må vi installere *SoftwareSerial* for å kunne kommunisere på to valgte datalinjer (Tx/Rx) i vårt tilfelle velges disse som dataportene D3 (Tx) og D4 (Rx) på Arduino-kortet.

Biblioteket kan hentes fra:

<https://github.com/mikalhart/TinyGPS/releases/tag/v13>

Hent ned zip-fila som heter: *Source code (zip)*. Ikke pakk ut fila, men lagre den på et sted du kan finne den igjen. Biblioteket installeres ved å åpne velge Skisse/Include Library/Add .zip library i Arduino editoren (IDE) og hent opp fila¹².

12.Ev. se <https://github.com/mikalhart/TinyGPS/releases/tag/v13>



Enkleste kode

Fra avsnitt 5.6 på side 104 så ser vi at foretrukket format på koordinat- og høydedataene er:

lengdegrad,breddegrad,høyde

Alle verdier oppgitt som desimaltall og uten mellomrom etter komma.

Koden under viser den enkleste utgaven av programvaren for å hente inn koordinatdata og skrive til både monitoren i Arduino (IDE) og til SD-kortet:

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
/* This sample code demonstrates the simplified use of a TinyGPS object.
   It requires the use of SoftwareSerial, and assumes that you have a
   9600-baud serial GPS device hooked up on pins 4(Rx) and 3(Tx).
*/
TinyGPS gps;                               // Definerer en TinyGPS type kalt gps
SoftwareSerial ss(4,3);                     // Definerer en SoftwareSerial
type kalt ss
static void smartdelay(unsigned long ms);

void setup()
{
  Serial.begin(115200);                     // Sett datahastighet til monitor
  ss.begin(9600);                           // Sett datahastighet til GPS-
kontrollkort
}

void loop()
{
  float flat, flon;                          // Definer variable for lengde- og
breddegrad
  int year;                                  // Definer variable for år, dato og tid
  byte month, day, hour, minute, second, hundredths;
  unsigned long age;                         // Definer variabel for age

  gps.f_get_position(&flat, &flon, &age); // Henter breddegrader og lengdegrader
  Serial.print(flon,6);                      // Skriver ut lengdegrader i grader
desimalt
  Serial.print(", ");                        // Sett inn komma som skilletegn
  Serial.print(flat,6);                      // Skriver ut breddegrader i
grader desimalt
  Serial.print(", ");                        // Sett inn komma som skilletegn
```



```
Serial.print(gps.f_altitude(), 2);           // Skriver ut høyde over havet
i meter
Serial.println();

    smartdelay(1000);
}

static void smartdelay(unsigned long ms)
{
    unsigned long start = millis();           // Hent antall millisekunder siden
    reset av Arduino-kortet
    do                                       // Testen gjøres etter loopen er
    kjørt
    {
        while (ss.available())gps.encode(ss.read());
    }
    while (millis() - start < ms); // Testen gjøres før loopen er kjørt
}
```

Etter innsamling av data kan disse lastes opp i f.eks. Notepad++ eller i Excel, se avsnitt 5.2 på side 96. Dersom man ønsker å visualisere traseen i Google Earth så klippes dataene inn i kml-koden vist i avsnitt 5.6 på side 104.

Kode med feilsjekk og flere data

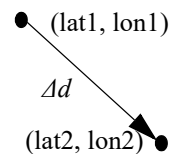
Den enkle programkoden omtalt i forrige avsnitt inneholder ingen sjekk mht. om dataoverføringen mellom GPS-enheten og Arduino er korrekt. En mer omfattende kode med feilsjekking finnes i vedlegg A.4.2 på side 217.

Kode for beregning av akkumulert distanse

Det finnes i dag en rekke App'er for pader og smarttelefoner som beregner gangavstand etter som man beveger seg i gater eller i terrenget, disse er basert på smarttelefonens eller padens innebygde GPS mottaker. Et eksempel på en slik er Runkeeper¹³.

Har vi først koblet en GPS-mottaker til vår Arduino, så burde det også være mulig å beregne tilbakelagt distanse. Siden høydemålingene er relativt usikre så nøyer vi oss med å bruke lengde- og breddegradene.

Som vist på figuren over så har vi to sett med koordinater (lat_1, lon_1 og lat_2, lon_2) og ønsker å beregne avstanden, Δd , mellom disse to settene av koordinater. Siden jorda er krummet som en kule så er ikke dette en triviell beregning. Under er gjengitt formelapparatet for beregningen¹⁴:



13. <https://runkeeper.com/>

14. <http://www.movable-type.co.uk/scripts/latlong.html>



$$\varphi_1 = \frac{\pi \cdot lat_1}{180} \quad (4.1)$$

$$\varphi_2 = \frac{\pi \cdot lat_2}{180} \quad (4.2)$$

$$\Delta\varphi = \varphi_2 - \varphi_1 \quad (4.3)$$

$$\Delta\lambda = \pi \cdot \frac{(lon_2 - lon_1)}{180} \quad (4.4)$$

$$a = \sin\left(\frac{\Delta\varphi}{2}\right)^2 + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin\left(\frac{\Delta\lambda}{2}\right)^2 \quad (4.5)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (4.6)$$

$$\Delta d = R \cdot c \quad (4.7)$$

Hvor:

φ_1	= Breddegrad posisjon 1 (lat_1) målt i radianer
φ_2	= Breddegrad posisjon 2 (lat_2) målt i radianer
$\operatorname{atan2}$	= Arkus tangens med to argumenter for å beholde informasjon om kvadrant
Δd	= Distansen i meter
R	= Jordradien i m ($6,371 \cdot 10^6$ m)

Dette er en relativt nøyaktig beregning, men kan være litt plundrete da standardbiblioteket til Arduino f.eks. ikke tilbyr $\operatorname{atan2}()$. Imidlertid finnes det en forenklet versjon som kan være nyttig når man ikke trenger stor nøyaktighet eller når avstandene mellom målepunktene er liten.

Ekvirektangulær tilnærming

I vårt tilfelle er avstandene noen meter slik at det skulle være helt uproblematisk å bruke den forenklete beregningsmetoden:

$$\varphi_1 = \frac{\pi \cdot lat_1}{180} \quad (4.8)$$

$$\varphi_2 = \frac{\pi \cdot lat_2}{180} \quad (4.9)$$

$$\Delta\lambda = \pi \cdot \frac{(lon_2 - lon_1)}{180} \quad (4.10)$$

$$x = \Delta\lambda \cdot \cos\left(\frac{(\varphi_1 + \varphi_2)}{2}\right) \quad (4.11)$$



$$y = \varphi_2 - \varphi_1 \quad (4.12)$$

$$\Delta d = R \cdot \sqrt{x^2 + y^2} \quad (4.13)$$

Hvor:

- φ_1 = *Breddegrad posisjon 1 (lat₁) målt i radianer*
- φ_2 = *Breddegrad posisjon 2 (lat₂) målt i radianer*
- Δd = *Distansen mellom punktene i meter*
- R = *Jordradien i m (6,371·10⁶ m)*

Akkumulert distanse

Den akkumulerte distansen finner vi da enkelt ved å summer opp alle de små distansene mellom hvert målepunkt. En må passe på at målepunktene ikke blir for sjeldene slik at en mister de fine detaljene i registreringen.

$$d = \sum_1^n \Delta d \quad (4.14)$$

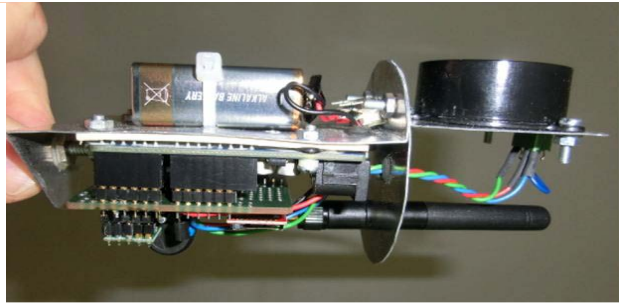
Hvor

- d = *Lengden av traseen, dvs. summen av de mange små avstandene*
- n = *Antallet målepunkter*



5 Analyse av data

Et praktisk og vanlig verktøy for analyse av data er Excel. I dette kapitlet skal vi se på de vanligste funksjonene for å importere og tegne ut grafer for innsamlede data fra CanSat. Til dette eksempelet bruker vi en CanSat med temperatur, trykk, 3-akse akselerometer og en CO₂-sensor som vist på bildet til høyre. Det må bemerkes at denne CanSat'en ikke tilfredsstillt krav til normert lengde.



5.1 Skrive data til file

5.1.1 Lagre rådata

Dersom man ønsker å redusere sannsynligheten for å miste data pga. feil i programmeringen, vil det være fornuftig å sende over, eller lagre rådata. Behandlingen av dataene kan likevel lett gjøres i etterbehandlingen i Excel. Overfører man beregnede data og man er så uheldig å regne feil i CanSat'en før data lagres eller overføres til bakkestasjonen, så er dataene tapt for alltid.

For å kalibrere og teste ut CanSat'en kan det være greit å la Arduino'en regne om til gjenkjennbare verdier. Dette er ikke til hinder for at man overfører eller lagrer rådata.

Følgende gir noen anbefalinger for organisering av data:

5.1.2 Signatur

Skriv inn en enkel signatur på hver linje slik at man er sikker på at de dataene man samler inn kommer fra den ønskede CanSat'en. Dersom data skal overføres på radio og det er flere samtidige brukere, kan man risikere at man tar inn data fra naboens CanSat i stedet for sin egen. Dette vil man lett oppdage når man har satt sin signatur på dataene.

5.1.3 Tidsangivelse

Inkluder en teller som angir tiden når dataene samles inn. En teller kan være grei dersom man har god kontroll på tidspunktet for hver gang telleren skrives til filen. En bedre måte er å skrive inn tiden fra Arduino'en ble startet (ev. restartet). Funksjonen:

```
millis();
```

returnerer en verdi som er lik antallet millisekunder siden start/restart. Denne funksjonen vil gå ut over sitt område først etter ca. 50 dager, hvilket skulle holde for de fleste formål.



5.1.4 Skilletegn

Verdiene som skrives til fil skilles med et skilletegn. Her kan man i prinsippet bruke ulike tegn f.eks.: <> ; tab. Det kan imidlertid være fornuftig å unngå “.” og “,” da disse gjerne brukes som desimalpunkt. Semikoloen kan derfor være et greit tegn å bruke.

5.1.5 Den endelige datafilen

I filen under har vi kun brukt en teller og “,” som skilletegn. I dette tilfellet går dette greit fordi vi vet at det tas én punktprøve hvert 4. sekund og vi overfører rådata uten komma.

```
Serial.print(teller);           // Teller tidsintervaller/målinger
Serial.print(", ");
Serial.print(digValuePres);    // Trykk digital rådataverdi
Serial.print(", ");
Serial.print(digValueTemp);    // Temperatur digital rådataverdi
Serial.print(", ");
Serial.print(digValueCO2);     // CO2 innhold digital rådataverdi
Serial.print(", ");
Serial.print(digValueXAksl);   // Akselerometer digital rådataverdi x
Serial.print(", ");
Serial.print(digValueYAksl);   // Akselerometer digital rådataverdi y
Serial.print(", ");
Serial.println(digValueZAksl); // Akselerometer digital rådataverdi z
```

Figuren under viser den endelige filen:

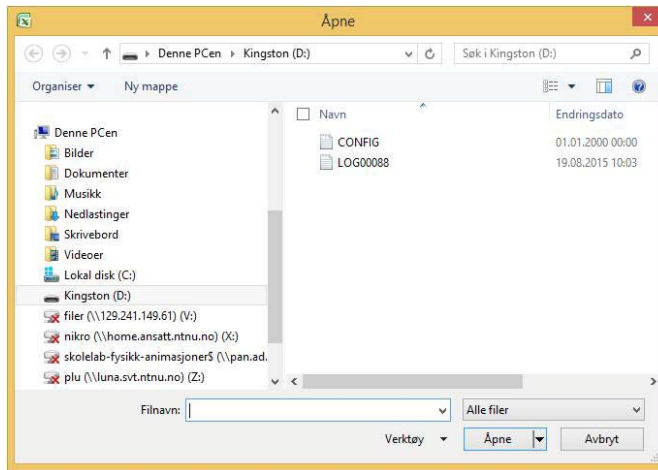
```
2, 837, 56, 566, 346, 408, 408
3, 837, 56, 566, 349, 408, 409
4, 837, 56, 562, 347, 407, 407
5, 837, 56, 564, 346, 407, 407
6, 837, 56, 566, 347, 408, 408
7, 838, 56, 575, 345, 409, 409
8, 837, 56, 570, 346, 408, 408
9, 837, 56, 567, 347, 408, 408
10, 837, 56, 567, 347, 408, 408
11, 838, 56, 570, 344, 409, 409
12, 838, 56, 586, 347, 408, 408
13, 838, 56, 592, 347, 407, 407
14, 837, 56, 587, 346, 409, 408
```

5.2 Importer data til Excel

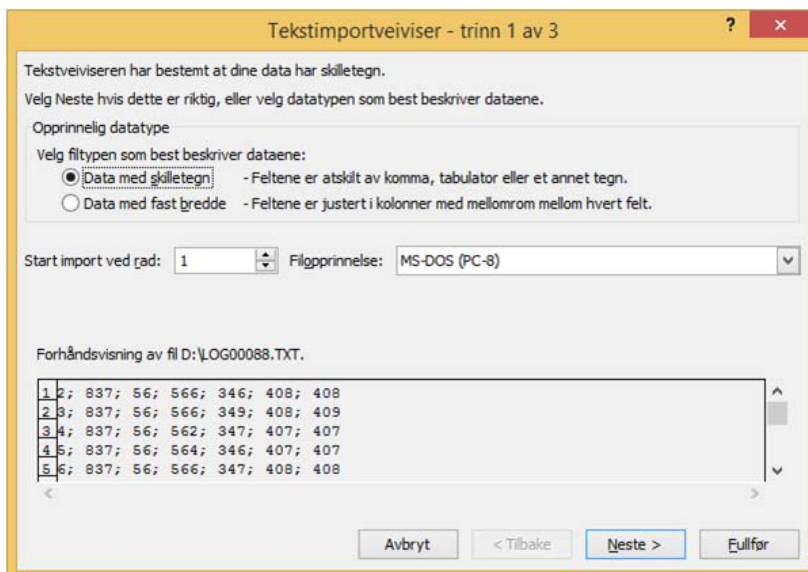
Importer av data i Excel kan gjøres på ulike måter. En måte som fungerer er å hente inn filen med “Open”. Da vil man ledes gjennom følgende vinduer for at dataene skal bli formatert på ønsket måte.



Velg ønsket fil:

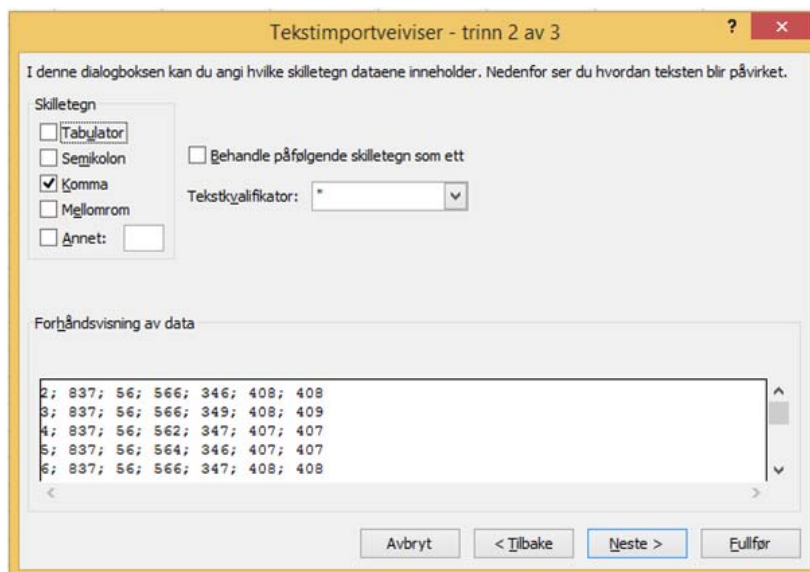


Velg “Data med skille tegn” og bestem fra hvilken rad du ønsker å starte å importere data fra. Trykk så “Neste”:

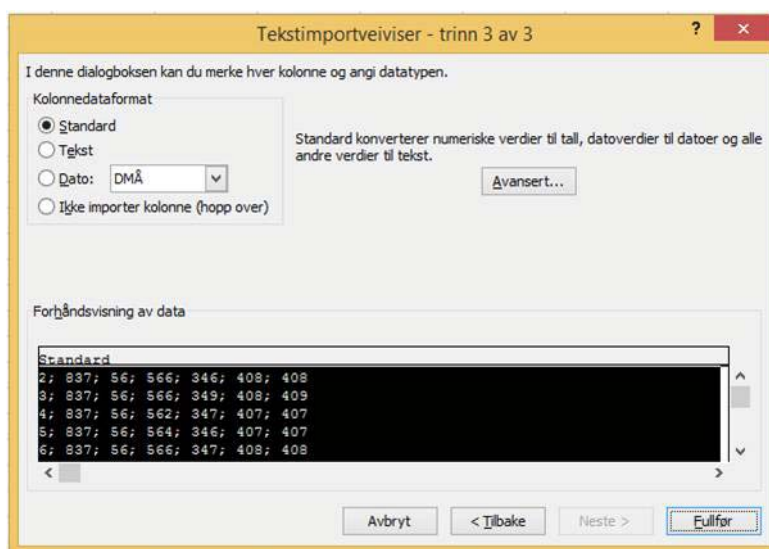




Velg hvilken type skilletegn som er benyttet. I vårt tilfelle har vi benyttet “komma”.



Tilslutt velges hvilken type data det gjelder, i vårt tilfelle er det standard tallformat. Dersom vi skal importere flyttall (med komma), velges desimalpunktet, “,” eller “.” ved å velge “Avansert”.



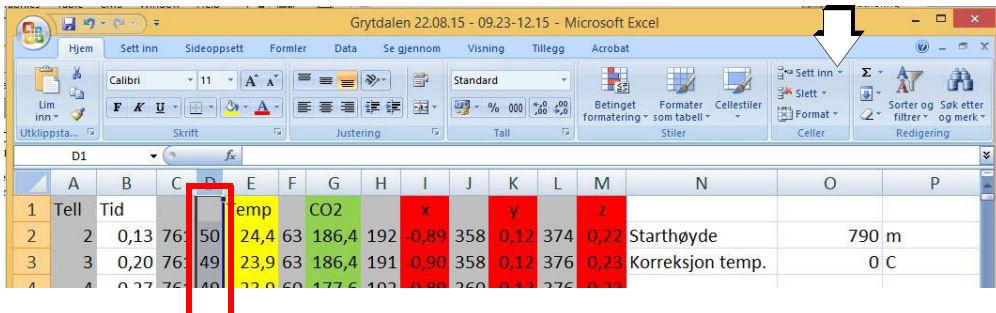


Dermed skal dataene være importert i regnearket.

	A	B	C	D	E	F	G
1	2	761	50	63	192	358	374
2	3	761	49	63	191	358	376
3	4	761	49	60	192	360	376
4	5	761	49	58	190	359	375
5	6	761	49	63	192	359	375
6	7	760	49	66	191	359	375
7	8	761	50	63	189	359	374
8	9	761	51	63	191	358	374
9	10	761	51	65	191	360	376
10	11	761	51	68	191	358	374
11	12	761	52	66	192	359	376
12	13	761	52	68	193	358	375
13	14	761	52	71	181	343	362
14	15	761	51	71	107	336	292
15	16	760	50	66	155	360	334
16	17	761	50	63	161	344	296
17	18	761	49	55	156	326	324
18	19	760	49	55	156	328	318
19	20	760	49	60	165	324	330
20	21	761	49	60	168	328	315
21	22	760	49	65	181	346	315
22	23	758	49	61	169	340	308
23	24	754	48	65	145	309	320

5.3 Klargjøring av data

Det neste som må gjøres er å konvertere data fra digitale verdier via spenning til fysisk størrelse. Det er nødvendig å opprette en ny kolumne eller benytte en ledig for hver ny størrelse som skal beregnes. Vi har valgt å opprette en ny kolonne for trykk til høyre for de digitale trykkverdiene. I så fall merkes kolonnene ved å trykke på bokstavmarkeringen over kolonnen til høyre for der vi ønsker å opprette en ny kolonne (merket med rødt i figuren under). Velg så pilen ved siden av “Sett inn” (se pilen på figuren under) og velg “Sett inn arkkolonner”.





Dermed opprettes en ny kolonne.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Tell	Tid			Temp	CO2				x		y		z			
2	2	0,13	761		50	24,4	63	186,4	192	-0,89	358	0,12	374	0,22	Starthøyde		790 m
3	3	0,20	761		49	23,9	63	186,4	191	-0,90	358	0,12	376	0,23	Korreksjon temp.		0 C
4	4	0,27	761		49	23,9	60	177,6	192	-0,89	360	0,13	376	0,23			

5.4 Legg inn formler

5.4.1 Trykk

Vi skal nå legge inn omregningsformelen fra den digitale verdien for trykket i kolonne C og plassere verdien i den nye kolonne D. Først regner vi om fra digital verdi til spenning (U_p):

$$U_p = 5V * C2 / 1023 \quad (5.1)$$

Deretter fra spenning til trykk (p) i henhold til formelen i databladet:

$$p = 22,222 * U_p + 10,556 \text{ [kPa]} \quad (5.2)$$

Dersom vi kombinerer de to formlene får vi:

$$p = 22,222 * (C2 / 1023) * 5 + 10,556 \quad (5.3)$$

Som legges inn i funksjonsfeltet i regnearket.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Tid		Trykk	Høyde		Temp	CO2			x		y		z		
2	0,13	761	10,556	790,00	50	24,4	63	186,4	192	-0,89	358	0,12	374	0,22	Starthøyde	790 m
3	0,20	761	93,21	790,00	49	23,9	63	186,4	191	-0,90	358	0,12	376	0,23	Korreksjon temp.	0 C

Deretter kopieres feltet D2 til de resterende feltene i kolonne D.

Slik gjør man for alle formler.



5.4.2 Høyde

Høyden (h) baserer seg på kolonnen med verdier for trykk, temperaturen ved start (T_1) og høyden ved start (h_1):

$$h = \frac{T_1}{a} \left(\left(\frac{p}{p_1} \right)^{\frac{aR}{g_0}} - 1 \right) + h_1$$

- h Beregnet høyde i meter
- h_1 Starthøyde i meter (Q2)
- T Temperatur i Kelvin
- T_1 Starttemperatur i høyden h_1 (G2)
- a Temperaturgradient, foreslått verdi $-0,0065$ K/m
- p Trykk i Pa (D2)
- p_1 Trykk i Pa ved starthøyden (D2)
- g_0 Tyngdeakselerasjonen $9,81$ m/s²
- R Den spesifikke gasskonstant $287,06$ J/kg K

Dette kan i vårt tilfelle skrives på følgende måte:

$$h = (271 + G2) / (-0,0065) * ((D2 / D2)^{(-0,0065 * 287,06) / 9,81} - 1) + Q2 \quad (5.4)$$

hvor parameterne er skrevet inn i lista over.

5.4.3 Temperatur

Etter å ha regnet om den digitale verdien for temperaturen til spenning, så settes den inn i formelen for omregning fra spenning til temperatur:

$$T = V_T * 100 (+ \Delta T) \quad (5.5)$$

hvor:

- T Er temperaturen i °C
- V_T Er avlest spenning som beregnes ut fra den digitale verdien (F2)
- ΔT Er ev. korreksjon i temperaturen (Q3)

Satt inn i formelfeltet blir denne i vårt tilfelle:

$$= ((F2 / 1023) * 5) * 100 + Q3 \quad (5.6)$$

Legg merke til formen Q3, som betyr at denne cellen skal holdes fast lik Q3 selv om man kopierer formelen ned over hele kolonnen.



5.4.4 CO₂

I vår CanSat har vi en CO₂-måler som erstatter NTC-motstandens plass. Omregningen fra spenning til CO₂ i ppm kan i følge databladet uttrykkes som:

$$\text{CO}_{2\text{ppm}} = \text{FS } V_{\text{out}}/V_{\text{supply}} \quad (5.7)$$

hvor:

- FS Fullskala konsentrasjon (2000 ppm)
- V_{out} Målt spenning på utgangen (H2 er den digitale spenningsverdien)
- V_{supply} Forskyningspenning som tilføres sensoren (3.3 V)

$$\text{CO}_{2\text{ppm}} = 2000 * ((\text{H2}/1023) * 5) / 3,3 \quad (5.8)$$

5.4.5 Akselerasjon

Akselerometeret måler akselerasjon i x-, y- og z-retning. Dataene fra hver av retningene beregnes og legges i hver sin kolonne. Akselerasjonene (A_X , A_Y , A_Z) beregnes ut fra de tre målte spennin- gene V_{AX} , V_{AY} og V_{AZ} . Vi kan sette opp følgende sammenhenger:

$$A_X = (V_{AX} - 1,65 + \Delta V_X) / S_g \text{ [m/s}^2\text{]} \quad (5.9)$$

$$A_Y = (V_{AY} - 1,65 + \Delta V_Y) / S_g \text{ [m/s}^2\text{]} \quad (5.10)$$

$$A_Z = (V_{AZ} - 1,65 + \Delta V_Z) / S_g \text{ [m/s}^2\text{]} \quad (5.11)$$

hvor

- A_X , A_Y , A_Z er den beregnede akselerasjonen i x-, y- og z-retning
- V_{AX} , V_{AY} , V_{AZ} er den målte spenningen fra hver av akselerometerne i x-, y- og z-retningen (de digitale verdiene konverteres til spenning).
- ΔV_X , ΔV_Y og ΔV_Z er korreksjonsspenningen for hver av retningene
- S_g er følsomheten som er 800mV/g i området +/- 1,5 g

Dersom vi setter inn omregningsformler fra digitale verdier til spenning kan vi skrive:

$$\text{Akselerasjon x-retning} = (((J2/1023) * 5) - 1,65) / 0,8 \quad (5.12)$$

I vårt tilfelle inneholder kolonnen J de digitale verdiene for spenningen i x-retningen. Det er ikke lagt inn noen korreksjonsverdi i ligning (5.12).

Når alle verdiene er beregnet og lagt i en kolonne kan vi lage grafer.

5.5 Lage grafer

Slik kan man lage grafer:



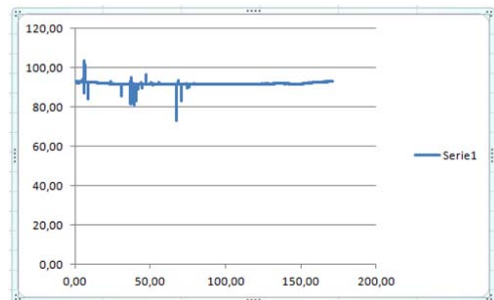
1. Merk de to kolonnene som skal representeres av en graf.
Det kan være lurt å lage en overskrift på kolonnene. Overskriften vil automatisk tilordnes seriene i grafen.

	A	B	C	D	E	F	G	H	I	J	K
1	2	0,13	761	93,12896	790,00000	50	24,4	63	186,4	192	-0,8
2	3	0,20	761	93,12896		49	23,9	63	186,4	191	-0,9
3	4	0,27	761	93,12896		49	23,9	60	177,6	192	-0,8
4	5	0,33	761	93,12896		49	23,9	58	171,6	190	-0,9
5	6	0,40	761	93,12896		49	23,9	63	186,4	192	-0,8
6	7	0,47	760	93,02045		49	23,9	66	195,3	191	-0,9
7	8	0,53	761	93,12896		50	24,4	63	186,4	189	-0,9
8	9	0,60	761	93,12896		51	24,9	63	186,4	191	-0,9

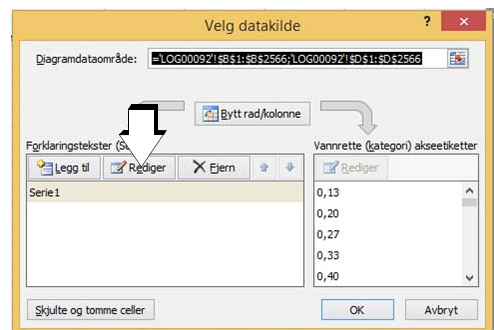
2. Velg type representasjon fra menyen “Sett inn” og velg f.eks. punktdiagram med linjer mellom de ulike punktene i diagrammet:



3. Når dette er gjort vil diagrammet tegnes ut som vist på figuren til høyre.

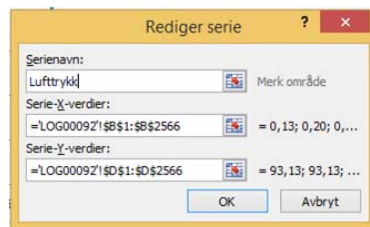


4. Dersom vi skal legge inn et navn på grafen i figuren gjør vi det ved å høyre-klikke på rammen av figuren og velge “Merk data” i menyen som framkommer.

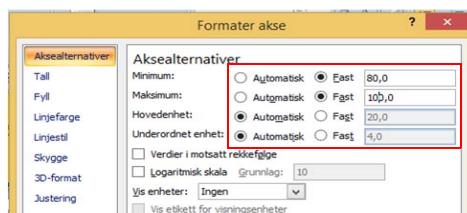




5. Merk "Serie 1" og trykk rediger. Skriv inn nytt "Serie-navn".

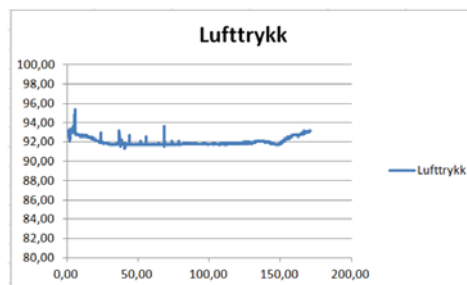


6. Dersom man ønsker å endre på akseverdiene, høyre-klikker man på akse og får opp boksen vist til høyre. F.eks. kan det være ønskelig å sette faste min. og maks. verdier på den vertikale akse som vist på figuren til høyre.



7. Det endelige diagrammet kan dermed bli som vist i figuren til høyre.

Slik kan man gjøre for alle sammenhenger man ønsker å lage grafer av.



5.6 Visualisering av GPS-data i Google Earth

Data fra GPS-moduler krever gjerne en litt annen behandling enn ved hjelp av regneark. Slike data kan f.eks. plottes i Google Earth. Avsnitt 7.12.2 på side 184 omtaler oppkobling av GPS-modulen, mens innhenting og skrivning av posisjonsdata er beskrevet i avsnitt 4.3.5 på side 89.

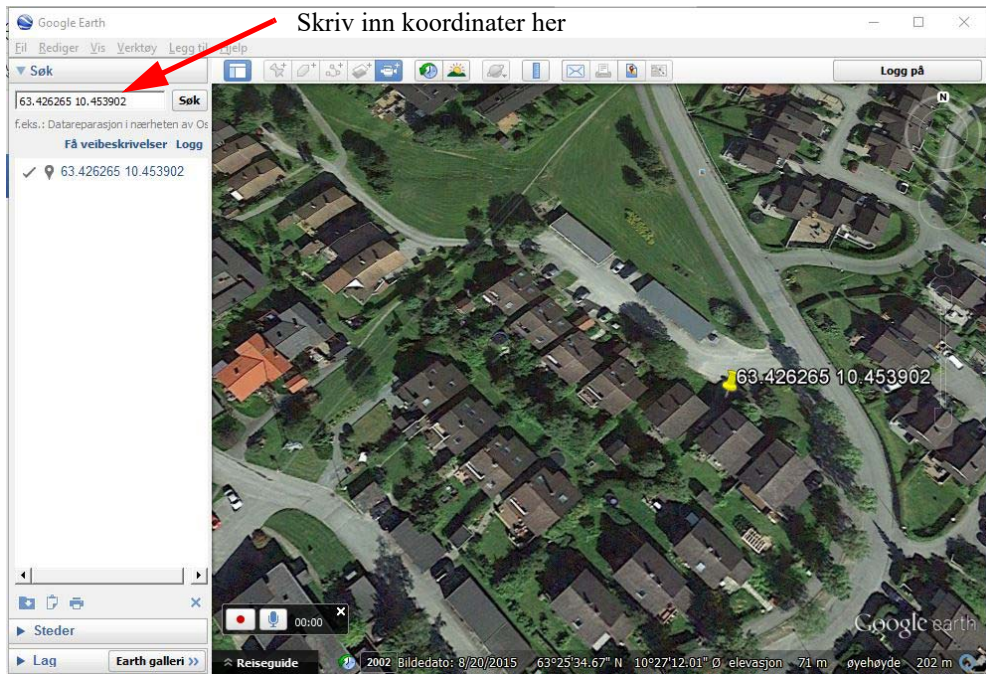
I dette avsnittet skal vi se hvordan vi kan hente inn de lagrede dataene og plote dem i Google Earth.

5.6.1 Installasjon av Google Earth

Google Earth kan lastes ned og installeres fra følgende adresse: <https://www.google.com/earth/>



Skjermbildet under viser brukergrensesnittet for Google Earth.



I øverste venstre hjørne finnes et søkefelt. Her kan man enten skrive en adresse, eller man kan skrive inn koordinater (med desimaler). I dette tilfellet har jeg skrevet inn:

Breddegrad: 63.426265

Lengdegrad: 10.453902

Da vil Google Earth zoome inn akkurat til disse koordinatene som i dette tilfellet var rett utenfor veggen der jeg sitter. Dvs. det kan se ut som den bommer med et par meter (ringen angir plasseringen av mottakeren).

Høydeangivelsen for det angitte stedet varierer mellom 60 – 80 meter over havet (Google Earth angir en høyde som varierer mellom 65 – 75 meter).



5.6.2 Plotting av en trase i Google Earth

Dersom man ønsker å plote en trase i Google Earth må man benytte et programmeringsspråk kalt “Keyhole Markup Language” (KML). Dette er et “markup language” utviklet for visualisering av to- og tredimensjonale strukturer knyttet til kartdata. Språket ble utviklet i forbindelse med etableringen av Google Earth som ble lansert i 2004. I 2008 ble KML godkjent som en internasjonal standard for denne type geografisk visualisering.



Siden språket er temmelig omfattende og vi kun trenger å bruke en beskjeden del av det, så benytter vi en ferdige programkode og klipper inn våre data for lengde-, breddegrad og høyde. Disse legges inn som en liste med data i kml-koden (se under), før kml-fila lagres med et ønsket navn.

Koordinater og høyde data legges inn som vist under:

-112.2550785337791,36.07954952145647,2357

Lengdegrader [°], Breddegrader [°], Høyde [m]

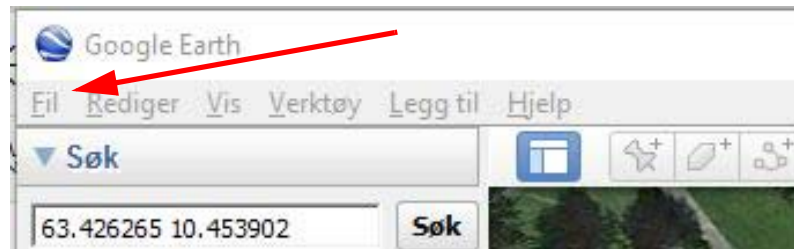
Programkode skrevet i kml (legg merke til at et sett med eksempelkoordinater og høyde er klippet inn).

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Paths</name>
    <description>Examples of paths. Note that the tessellate tag is by default
      set to 0. If you want to create tessellated lines, they must be authored
      (or edited) directly in KML.</description>
    <Style id="yellowLineGreenPoly">
      <LineStyle>
        <color>7f00ffff</color>
        <width>4</width>
      </LineStyle>
      <PolyStyle>
        <color>7f00ff00</color>
      </PolyStyle>
    </Style>
    <Placemark>
      <name>Absolute Extruded</name>
      <description>Transparent green wall with yellow outlines</description>
      <styleUrl>#yellowLineGreenPoly</styleUrl>
      <LineString>
        <extrude>0</extrude>
        <tessellate>0</tessellate>
        <altitudeMode>absolute</altitudeMode>
        <coordinates>
          -112.2550785337791,36.07954952145647,2357
          -112.2549277039738,36.08117083492122,2357
          -112.2552505069063,36.08260761307279,2357
          -112.2564540158376,36.08395660588506,2357
          -112.2580238976449,36.08511401044813,2357
          -112.2595218489022,36.08584355239394,2357
          -112.2608216347552,36.08612634548589,2357
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```



```
-112.262073428656,36.08626019085147,2357  
-112.2633204928495,36.08621519860091,2357  
-112.2644963846444,36.08627897945274,2357  
-112.2656969554589,36.08649599090644,2357
```

```
</coordinates>  
</LineString>  
</Placemark>  
</Document>  
</kml>
```

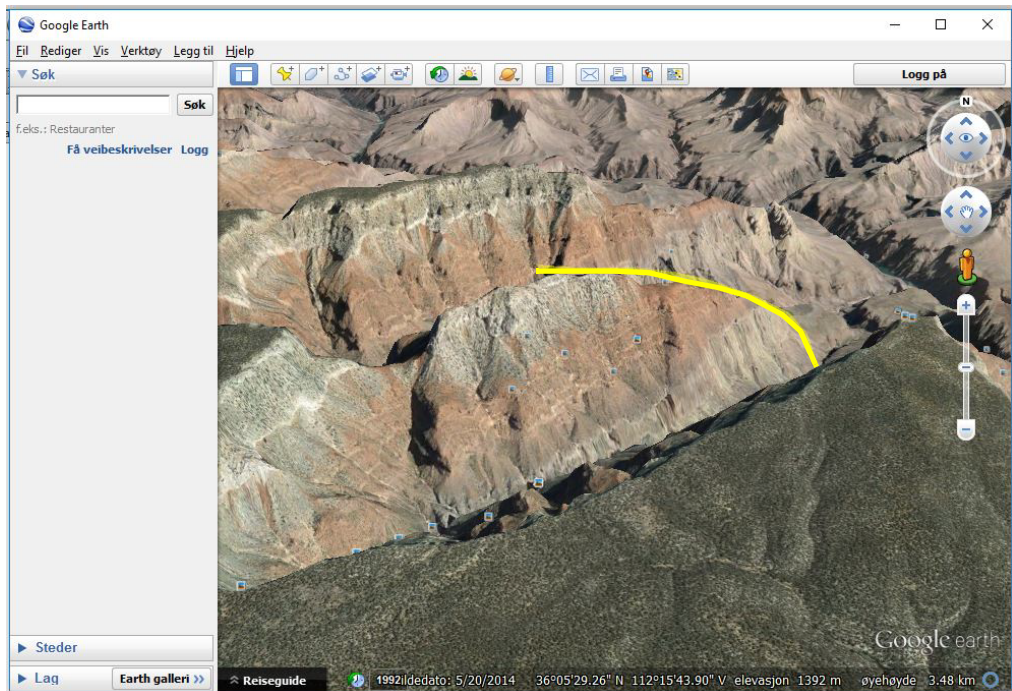


De koordinatene som pr. i dag ligger i eksempelet angir en helikopterflyvning

over Grand Canyon i Colorado, USA. Eksempeldataene byttes ut med de aktuelle dataene og kodefila lagres under et ønsket filnavn som ender med .kml.

Filen hentes opp i Google Earth ved å velge *Fil* og *Åpne* for å laste opp den filen hvor koden og dataene ligger.

En vil da få tegnet inn traseen som angitt av lista med koordinater og vist på riktig sted. Eksempelet under viser traseen til helikopteret over Grand Canyon (gul linje).





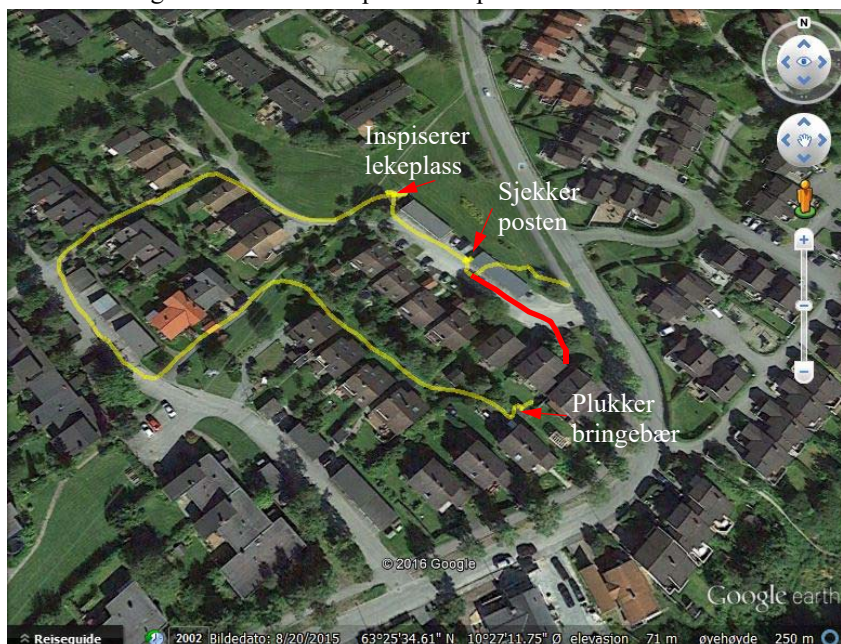
Dersom man ønsker å vise hvor man har godt en tur så kan det være upraktisk å måtte vise absolutt høyde. Høydemålinger kan ha store avvik slik at en kan oppleve at traseen går mange meter over bakken eller forsvinner under bakken til tross for at man hadde begge beina på jorda. I slike situasjoner kan det være greit å bytte ut kommandoen:

```
<altitudeMode>absolute</altitudeMode>
```

med kommandoen:

```
<altitudeMode>clampToGround</altitudeMode>
```

Dermed vil kurven følge bakken som vist på traseen på bildet under.



Vi legger imidlertid merke til at mottakeren har problemer i starten. I dette området er avviket stort før den tar seg inn. Den røde kurven angir den riktige ruta. Deretter er den svært nøyaktig. Posisjonsdata samples hvert 3. sekund.

Ved bruk av GPS-data ved oppsending av CanSat kan man selvfølgelig ikke neglisjere høydeinformasjonen. Man bør imidlertid være klar over risikoen for relativt store avvik. En bør vurdere å legge inn høydedata fra høydemåleren framfor GPS-data.

5.6.3 Editering av kml-fila

Hvilket program skal man så bruke for å legge inn koordinater og høyde. Et nyttig editeringsverktøy til dette formålet er Notepad++. Dette programmet er en litt avansert teksteditor som ikke gjør noen endringer med filformatet så fremt man ikke ønsker det. Notepad++ kan lastes ned fra:

<https://notepad-plus-plus.org/download/v6.9.2.html>



Figuren under viser et utsnitt av brukergrensesnittet til Notepad++.

```
C:\Arduino\CanSat\kml-files\Path.kml - Notepad++
Fil Rediger Søk Vis Format Språk Oppsett Makro Utfør Tillegg Vindu ?
Path.kml [3]
1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://www.opengis.net/kml/2.2">
3 <Document>
4 <name>Paths</name>
5 <description>Examples of paths. Note that the tessellate tag is by default
6 set to 0. If you want to create tessellated lines, they must be authored
7 (or edited) directly in KML.</description>
8 <Style id="yellowLineGreenPoly">
9 <LineStyle>
10 <color>7f00ffff</color>
11 <width>4</width>
12 </LineStyle>
13 <PolyStyle>
14 <color>7f00ff00</color>
15 </PolyStyle>
16 </Style>
17 <Placemark>
18 <name>Absolute Extruded</name>
19 <description>Transparent green wall with yellow outlines</description>
20 <styleUrl>#yellowLineGreenPoly</styleUrl>
21 <LineString>
22 <extrude>0</extrude>
23 <tessellate>0</tessellate>
24 <altitudeMode>absolute</altitudeMode>
25 <coordinates>
26 -112.2550785337791,36.07954952145647,2357
```

For å forenkle prosessen med å klippe inn data i kml-koden, er det praktisk å skrive koordinat- og høydedata i det formatet som programmet ønsker: Lengdegrad, breddegrad, høyde. Husk å bruke punktum som desimalpunkt og komma mellom hver verdi. **NB! Det skal ikke være mellomrom etter kommaene.** For å se hvordan dette kan gjøres se avsnitt 4.3.5 på side 89.



6 Noen utvalgte systemkomponenter

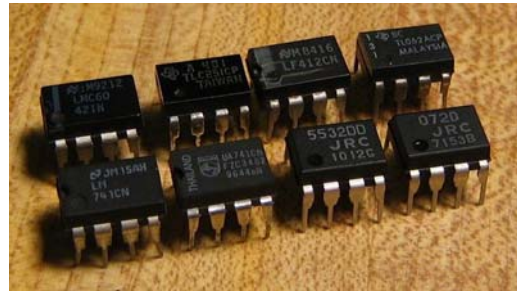
I moderne elektronikkdesign brukes transistoren som enkeltkomponent i stadig mindre grad. Som oftest møter vi transistorer som del av et større system, integrert på en silisumbrikke. Dette gjelder både komponenter som behandler analoge så vel som digitale signaler. Ved å integrere mange komponenter på samme silisumbrikke (chip), oppnås en betydelig komprimering og prisreduksjon, i tillegg til at en ofte oppnår bedre elektriske egenskaper, som for eksempel høyere hastighet (båndbredde) og lavere effektforbruk. En av de største utfordringene med den stadig økende tettpakkingen av transistorer (integrasjonen), er å bli kvitt varmen som oppstår når det går strøm i kretsene. Dette gjør man i dag ved å montere store kjøleflater og vifter på kretsene. Eller aller helst, reduserer effektforbruket til kretsen ved kreative teknologiske løsninger.

I dette avsnittet skal vi se på noen sentrale byggeblokker som er viktig i forbindelse med måle- og sensorteknologi.

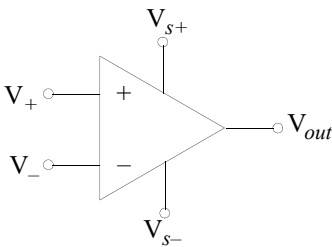
6.1 Operasjonsforsterkeren

Operasjonsforsterkeren ble utviklet på 1960-tallet primært som en generell byggeblokk for å utføre analoge matematiske regneoperasjoner som addisjon, subtraksjon, derivasjon og integrasjon. Senere ble operasjonsforsterkeren videreutviklet til mange forskjellige formål, ikke minst som et generelt forsterkerelement.

Moderne operasjonsforsterkere leveres som integrerte kretser med fra 6 til 16 bein. På bildet til høyre¹⁵ ser vi noen eksempler på operasjonsforsterkere montert i plastpakning med 8 bein plassert i to rader. Denne typen pakning betegnes ofte *DIL* – *Dual In Line* eller *DIP-8* – *Dual in Parallell* med 8 bein.



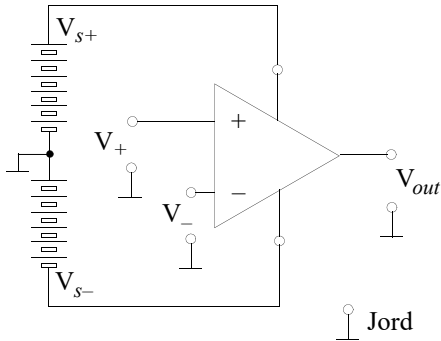
Figur 6.1 Eksempler på integrerte operasjonsforsterkere.



Kretssymbolet for en operasjonsforsterker er som vist i figuren til venstre. Som vi ser har forsterkeren to innganger (V_+ og V_-) og en utgang (V_{out}). I tillegg er det vanlig at den bruker to forskyningsspenninger f.eks. $V_{s+} = +9\text{ V}$ og $V_{s-} = -9\text{ V}$. Men det finnes også flere varianter som kun trenger en spenning.

I tillegg kan det være tilkoblingsterminaler for justering av kretsens egenskaper. Mange vil kanskje savne jord som referansepotensial for spenningene på inn- og utgang. Denne er ofte underforstått i forenklede koblingsskjemaer som på figuren til venstre. I den neste figuren har vi inkludert jord.

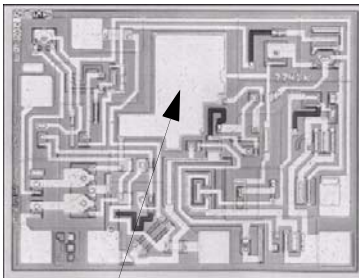
15.Hentet fra <http://en.wikipedia.org/wiki/Image:Op-amps.jpg>



Figur 6.2 Operasjonsforsterker med spenningsstilkobling og jording.

logi. Spesielt var det eksepsjonelt at man hadde klart å inkludere en kondensator på 30 pF på selve brikken. Kondensatorer er generelt vanskelige komponenter å integrere da de tar stor plass (til venstre på figur 6.3). Kondensatoren er viktig for å gjøre kretsen stabil, dvs. at den ikke begynner å lage signaler på egen hånd (oscillere).

Forstørret bilde av chip for $\mu A741$



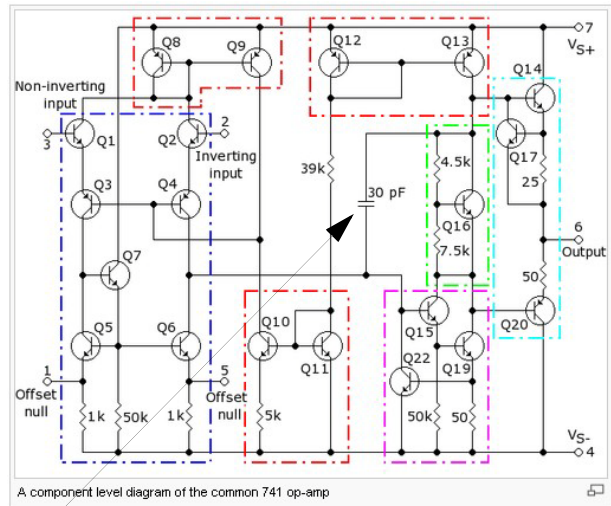
Kondensator

Vi legger også merke til at den i tillegg till signalinngangene og utgangen, har to tilkoblingspunkter kalt *Offset null*. Disse to tilkoblingene kan brukes til å gjøre kretsen mer nøyaktig, dvs. sørge for at 0 V på inngangen virkelig blir 0 V på utgangen.

Figur 6.2 viser hvordan jordpunktene er koblet til midtpunktet mellom de to spenningskildene. Vi har ikke brydd oss om å markere at alle jordpunktene er koblet sammen, hvilket er tilfelle.

En av de første integrerte operasjonsforsterkerne som ble laget gikk under betegnelsen $\mu A741$ og ble utviklet av **Bob Widlar** i 1965 for det amerikanske firmaet Fairchild. Denne er senere blitt en klassiker som fortsatt produseres og brukes. Riktignok er designet forbedret, men i grove trekk er det den samme konstruksjonen. Prisen for en $\mu A741$ hos ELFA er kr. 3,20 når man kjøper 10 stk.

Figur 6.3 viser koblingsskjemaet for operasjonsforsterkeren $\mu A741$. Etter dagens standard er dette en meget enkel krets, men da den ble lansert på midten av 60-tallet, var den meget avansert tekno-



Kondensator

Figur 6.3 Koblingsskjema for den opprinnelige operasjonsforsterkeren $\mu A741$



6.1.1 Prinsipiell virkemåte og sentrale parametere for operasjonsforsterkeren

Som nevnt har operasjonsforsterkeren to innganger, en inverterende og en ikke-inverterende inngang. Ved bruk av den inverterende inngangen vil et signal bli snudd opp ned på utgangen, vi sier at det blir *invertert* (figur 6.4 A).

Tilsvarende vil et signal påtrykt den ikke-inverterende inngangen, ikke bli invertert. Dette er forsøkt illustrert i figur 6.4 B. Vi legger merke til at signalet på utgangen også er forsterket.

Egentlig er det differansesignalet mellom de to inngangene som forsterkes. Dersom vi kobler sammen de to inngangene V_+ og V_- og sender samme signal inn på begge, så vil vi ideelt sett få null signal eller spenning på utgangen av kretsen. Forsterkningen, A , til operasjonsforsterkeren alene kan vi definere som:

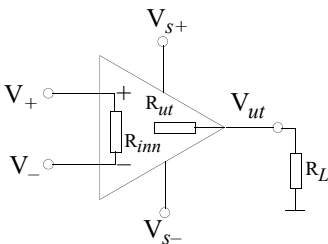
$$A = \frac{V_{out}}{V_+ - V_-} \quad (6.1)$$

Som vi ser så er forsterkningen definert som forholdet mellom signalet på utgangen (A_{out}) og spenningen mellom de to inngangene ($V_+ - V_-$). Denne forsterkningen kalles ofte *Open loop* forsterkningen.

Før vi går videre skal vi se på egenskapene til en ideell operasjonsforsterker, slik vi skulle ønske at den var.

Inngangsmotstand (R_{inn}):

For at kretsen skal belaste signalkilden (f.eks. en mikrofon) minst mulig, ønsker vi at verdien til inngangsmotstanden (R_{inn}) til operasjonsforsterkeren skal være størst mulig, ideelt sett uendelig høy. For $\mu A741$ er denne mellom 300 k Ω og 2 M Ω .



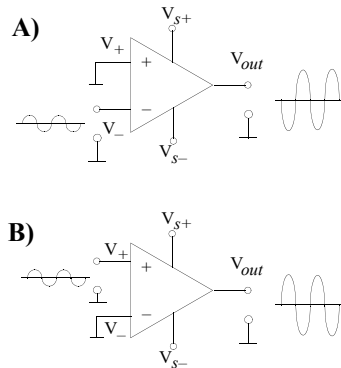
Figur 6.5 Inn- og utgangsmotstand til en operasjonsforsterker.

Det finnes dessuten andre operasjonsforsterker som har vesentlig høyere inngangsmotstand.

Verdien til inngangsmotstanden for en virkelig krets er altså ikke uendelig, men ofte tilstrekkelig høy til at vi kan betrakte den som uendelig høy.

Utgangsmotstand (R_{ut}):

Utgangsmotstanden, R_{ut} , til operasjonsforsterkeren er en ekvivalent motstand koblet i serie med kilden inne i kretsen som leverer spenningen V_{ut} på utgangen. Utgangen belastes med *lastmotstanden* eller bare *lasten*, R_L . Lasten kan f.eks.



Figur 6.4 Inverterende og ikke-inverterende inngang.



være en høyttaler, en ørepropp eller noe annet. Vi får dermed en spenningsdeling mellom den indre utgangsmotstanden, R_{ut} og den ytre lasten, R_L . Dermed vil en del av effekten som skulle ha vært levert til lastmotstanden, bli til varme inne i kretsen og ikke komme til nytte. Dersom verdien til den indre motstanden er lav i forhold til den ytre lastmotstanden, vil lite gå til spille. Er den derimot ganske stor i forhold til lastmotstanden, så vil mye av den leverte effekten bli til varme inne i kretsen. Vi ønsker derfor at den indre motstanden på utgangen (R_{ut}) skal ha så liten verdi som mulig og helst 0Ω . For en $\mu A741$ er R_{ut} typisk 75Ω , det vil si at den relativt høy, Konsekvensen er at utgangen av $\mu A741$ må belastes forsiktig, dvs. relativt høye belastningsmotstand.

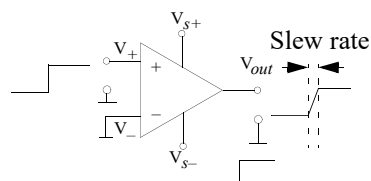
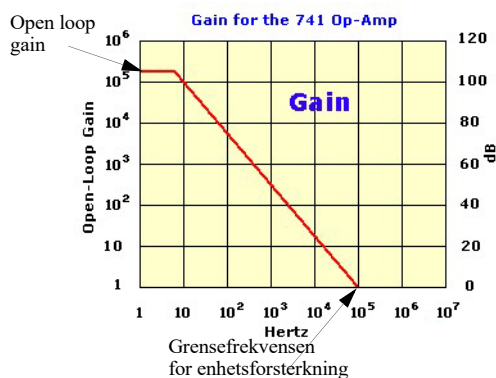
Båndbredde og forsterkning

Båndbredden til en operasjonsforsterker forteller oss noe om hvor høye signalfrekvenser den kan forsterke. Etter som frekvensen til signalet øker, vil forsterkningen avta. Dette er omtrent som for øret vårt. Båndbredden for øret til et nyfødt barn kan være opp mot 20 kHz, dvs. øret kan oppfatte lyder med frekvenser fra ca 16–20 000 Hz. Etter som vi blir eldre vil den øvre grensen avta. Dvs. at ørets båndbredde reduseres.

En måte å måle båndbredden til en operasjonsforsterker på, er finne den frekvensen hvor forsterkningen, A , er lik 1. Ved meget lave frekvenser, eller ved likespenning, så kan forsterkningen for en $\mu A741$ være på typisk 200 000 ganger. Dette er en enorm forsterkning. Men alt ved 100 Hz er forsterkningen falt til ca. 8 000. Etter som frekvensen øker vil forsterkningen avta raskt til signalnivået på inngangen er like sterkt som signalet på utgangen. Da er forsterkningen lik 1. Frekvensen da dette inntreffer, kalles *grensefrekvensen for enhetsforsterkning*. Det ideelle ville ha vært at forsterkning og båndbredde var uendelig.

Slew rate er en annen måte å angi båndbredden til en operasjonsforsterker på. Vi påtrykker da en sprangfunksjon, dvs. vi lar spenningen stige fra 0 V til en eller annen spenning (f.eks. 1 V) i løpet av meget kort tid. På utgangen vil vi da oppdage at spranget ikke er uendelig bratt, men trenger tid til å forandre nivået fra ett spenningsnivå til et annet. Hvor rask denne endringen er, sier noe om båndbredden til kretsen. Evnen til rask endring kalles kretsens *slew rate* og angis som Volt/ μs (dvs. hvor mange volt den klarer å stige fra 0 Volt i løpet av ett μs). Målingen er ofte gjort når kretsen er koblet opp med en forsterkning lik 1 (enhetsforsterkning).

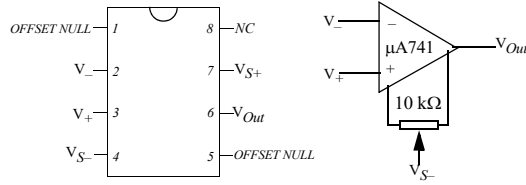
Ideelt sett ønsker vi at *slew rate* er uendelig stor, for en $\mu A741$ er den imidlertid $0,5 \text{ V}/\mu s$, som ikke er spesielt raskt, men tilstrekkelig i mange sammenhenger.





Offset:

Offset er en parameter som skyldes ubalanse i operasjonsforsterkerens interne komponenter på grunn av produksjons-spredning, men kan også skyldes forandring i oppførsel pga. av endringer i temperaturen under vanlig bruk. Dersom vi kobler sammen de to inngangene V_+ og V_- slik at spenningen mellom dem er 0 V, så ville vi forvente at vi på utgangen måler 0 V. Slik vil det imidlertid ikke alltid være. Et avvik fra 0 V vil derfor være et avvik fra det ideelle. Dette avviket kalles *offset*.



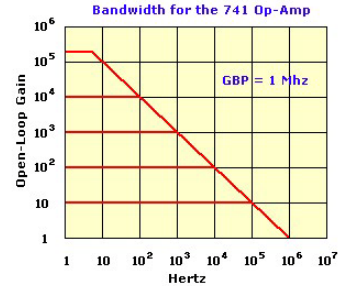
Figur 6.6 Justering av offset.

Til venstre på figur 6.6 er vist hva som er tilsluttet de ulike beina på kretsen (*pinningen*). Her er det to terminaler som er betegnet *Offset null*. Disse kobles til hver side på et potensiometer som vist til høyre på figur 6.6. Midtuttaket på potensiometeret tilkobles V_{S-} (– på spenningsforskyning). Ved å justere potensiometeret kan en få 0 V på utgangen.

6.1.2 Praktiske oppkoblinger

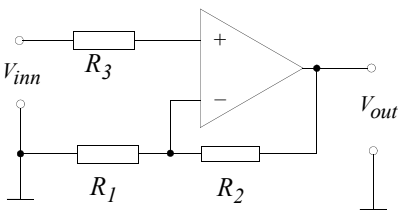
Det sjelden vi trenger en forsterkning på 200 000 eller en båndbredde på mange MHz. Vi vil derfor som oftest redusere forsterkningen til ønsket nivå ved å *tilbakekoble forsterkeren*.

Ved å ta noe av signalet på utgangen og koble det tilbake til den inverterende inngangen, vil forsterkningen reduseres betydelig. Det er som om signalet på utgangen motarbeider signalet på inngangen. Ved bruk av et nettverk av motstander kan vi bestemme akkurat hvor mye av signalet på utgangen som skal kobles tilbake. Graden av tilbakekobling bestemmer hvilken forsterkning kretsen får.



Figur 6.7 Sammenhengen mellom båndbredde og forsterkning

Dessuten øker båndbredden dersom vi reduserer forsterkningen ved hjelp av tilbakekobling. I figur 6.7 ser vi at en forsterkning på 10 gir en båndbredde på ca. 100 kHz.



Figur 6.8 Tilbakekoblet ikke-inverterende forsterker.

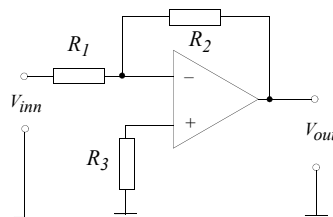
Figur 6.8 viser en enkel måte å koble signalet tilbake til inngangen. Forsterkningen er bestemt av forholdet mellom verdiene til motstandene R_1 og R_2 . Siden vi fører signalet inn på +inngangen (via R_3), vil ikke signalet bli invertert. R_3 beskytter inngangen og er relativt ukritisk. Det er imidlertid ikke uvanlig å velge den lik parallellkoblingen av R_1 og R_2 . Den er uten betydning for forsterkningen. For å forenkle figuren har vi tillatt oss å sløyfe batteriene og jordpunktet mellom de to spenningskildene.



Forsterkningen, A , til en slik krets er bestemt av følgende ligning:

$$A = 1 + \frac{R_2}{R_1} \quad (6.2)$$

Spenningsforsterkningen A er en tilnærming under forutsetning av at *Open loop* forsterkningen er mye større enn den ønskede forsterkningen A , hvilket den er for lave frekvenser. Etter som frekvensen øker, vil tilnærmingen i lign. (7.1) bli dårligere og dårligere. Dette gir seg utslag i at forsterkningen faller. Vi ser også at verdien til R_1 må gjøres svært stor dersom vi skal få en forsterkning nær 1, hvilket kan gjøre at kretsen blir ustabil.



Figur 6.9 Tilbakekoblet inverterende forsterker.

Ønsker vi en spenningsforsterkning ned mot 1 eller under 1, og vi kan tillate at signalet blir invertert, så er det ofte mer hensiktsmessig å bruke oppkoblingen vist på figur 6.9. Her vil forsterkningen være bestemt direkte av forholdet mellom R_2 og R_1 og gitt av lign. (6.3):

$$A = -\frac{R_2}{R_1} \quad (6.3)$$

Legg merke til fortegnet som indikerer at forsterkeren inverterer signalet.

Vi legger merke til at ingen av de to forsterkerkoblingene er følsomme for *Open loop* forsterkningen. Om denne er 200 000 eller 300 000 så betyr dette lite for forsterkningen for en tilbakekoblet krets, så lenge spenningsforsterkningen holdes på et akseptabelt lavt nivå. Kretsen er derfor relativt ufølsom for variasjoner i dataene fra krets til krets.

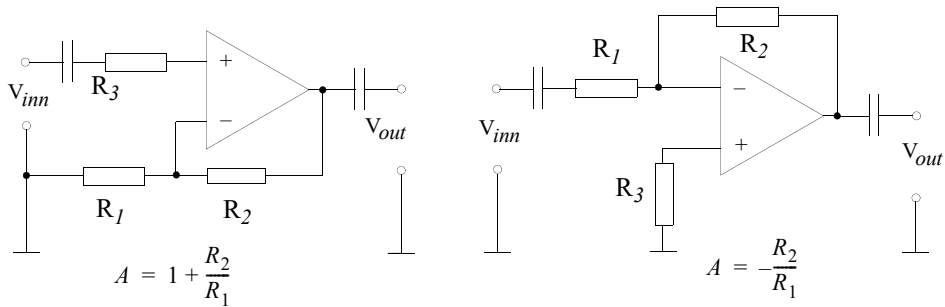
R_3 har ingen betydning for forsterkningen og pleier å velges i området 1 k Ω til 10 k Ω .

Dersom vi ønsker å bygge en inverterende forsterker med en forsterkning på 10. Kan vi typisk velge $R_2 = 10$ k Ω , hvilket skulle gi $R_1 = 1$ k Ω ($A = -R_2/R_1 = -10$).

Begge disse forsterkerne er likespenningsforsterkere og vil derfor i teorien fungere fra 0 Hz og oppover. Likespenningsforsterkere har imidlertid en tendens til å endre seg med temperaturen (*temperaturdrift*). Dersom en ikke har behov for å forsterke likespenning, kan det være greit å



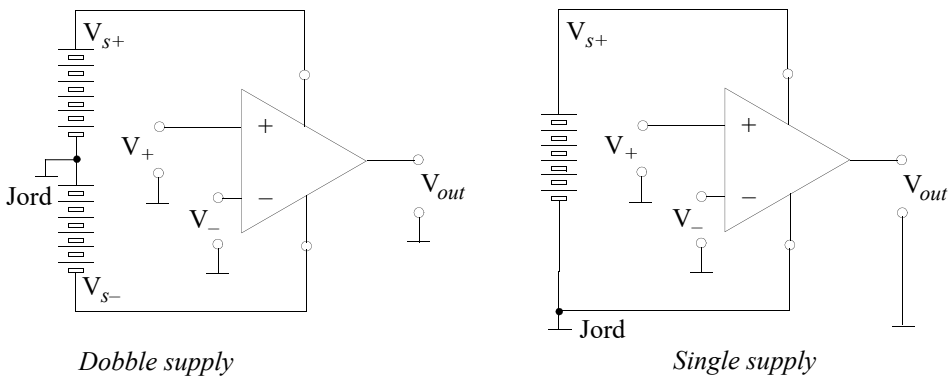
sette en kondensator i serie med inngangen. Da unngår en temperaturdrift, men kretsen vil ikke fungere for likespenning og for lave frekvenser.



Figur 6.10 Ikke inverterende og inverterende forsterkerkobling med kondensator på inn- og utgang.

6.1.3 Forsterker med enkel spenningsforskyning

Så langt har vi sett på forsterkere med dobbel spenningsforskyning. Noen ganger kan dette være praktisk dersom en ønsker å levere en meget lavfrekvent spenning som varierer omkring 0 V. Andre ganger vil en foretrekke å bruke bare en spenningskilde og kan til gjengjeld tillate at utgangsspenningen varierer omkring en spenning forskjellig fra 0 V. Alt det vi har sagt foran vil gjelde også for en operasjonsforsterker som er beregnet på en enkelt spenningsforskyning, eller *single supply*. Figuren under viser de to variantene.



Eksempler på single supply operasjonsforsterkere er:

- CA3130
- TS920 (dobbel)



6.1.4 Komparatoren

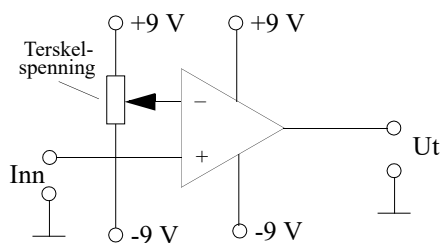
En komparator er en elektronisk krets som sammenligner en spenning på inngangen med en på forhånd innstilt terskelspenning. Når spenningen på inngangen overstiger terskelen, vil utgangen f.eks. gå fra 0 V til maksimal spenning (9 V).

La oss se på et konkret eksempel.

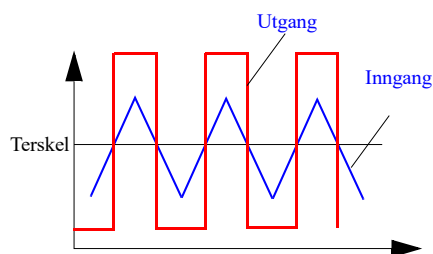
Figur 6.11 viser en komparator hvor minus-inngangen er koblet til et potensiometer. Ved hjelp av potensiometeret kan vi sette komparatorens *terskelspenning*. Pluss-inngangen tilkobles spenningen vi ønsker å sammenligne med terskelen. Spenningsnivået på utgangen vil fortelle oss om spenningen på inngangen er over eller under terskelspenningen.

I vårt eksempel vil spenningen på utgangen være f.eks. +9 V dersom inngangsspenningen passerer terskelspenningen. Om inngangsspenningen er under terskelspenningen vil utgangsspenningen være nær -9 V.

viser hvordan utgangsspenningen mellom + og -9V idet trekantspenningen på inngangen passerer terskelspenningen.



Figur 6.11 Oppkobling av komparator

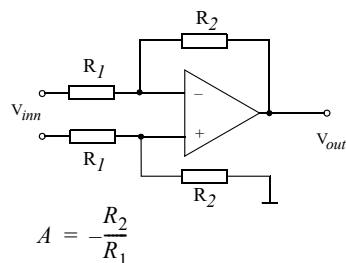


Figur 6.12 Inn- og utgangsspenning for komparatoren med terskelspenning som vist.

6.1.5 Differensialforsterkeren

Mens de to koblingene vi studerte i forrige avsnitt forsterker et signal referert til jord, så vil en differensialforsterker forsterke et signal eller en likespenning mellom to punkter adskilt fra jord. Dette vil være spesielt aktuelt når vi skal bruke forsterkeren til målinger som vi skal se i forbindelse med bygging av en målebro. Siden operasjonsforsterkeren i utgangspunktet forsterker spenningen som ligger mellom de to inngangene V_+ og V_- , så skulle en tro at denne forsterkeren var godt egnet som differensialforsterker. Oppkobling vist på figur 6.13 vil da også fungere, men vil ha en del ulemper.

Dersom vi ønsker å endre forsterkningen så må to av motstandene endres. Enten begge R_1 eller begge R_2 . Videre vil kretsen ha en relativt lav inngangsmotstand, som i noen sammenhenger kan være et problem.



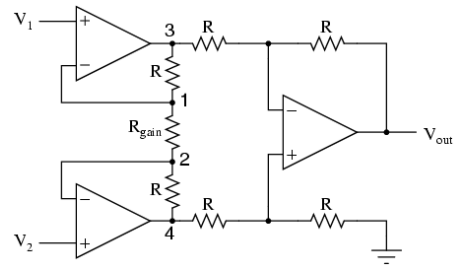
$$A = -\frac{R_2}{R_1}$$

Figur 6.13 Differensialforsterker med én operasjonsforsterker.



Begge disse problemene kan lett løses ved å bruke tre forsterkere koblet opp som vist på figur 6.14¹⁶. Denne koblingen kalles en *instrumenteringsforsterker*

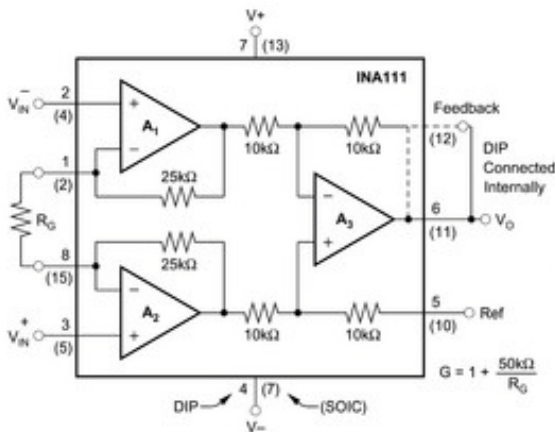
Forsterkningen i kretsen kan endres kun ved å endre R_{gain} . Dessuten er hver av de to inngangene til differensialforsterkeren buffret slik at verdien til inngangsmotstanden på instrumenteringsforsterkeren blir høy.



$$A_v = \left(1 + \frac{2R}{R_{gain}}\right)$$

Figur 6.14 Differensialforsterker med tre operasjonsforsterkere.

Det er slett ikke nødvendig å koble opp tre slike forsterkere for å få en instrumentforsterker. De finnes ferdig oppkoblet som integrerte løsninger på én brikke. En-stykkprisen for en slik krets hos ELFA er ca. kr. 100,00.



INA111 (Burr Brown)



Figur 6.15 Integrert differensialforsterker fra Burr-Brown.

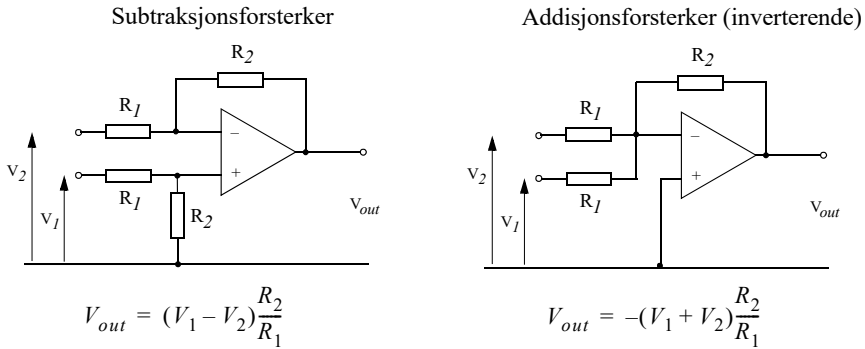
16. Stoffet til dette avsnittet er hentet fra

http://www.opamp-electronics.com/tutorials/the_instrumentation_amplifier_3_08_10.htm



6.1.6 Subtraksjons- og summasjonsforsterkere

Operasjonsforsterkeren egner seg dessuten godt både for å subtrahere to signaler fra hverandre, eller å legge sammen to (eller flere signaler) signaler. Figuren under gjengir to koblinger som fungerer slik.



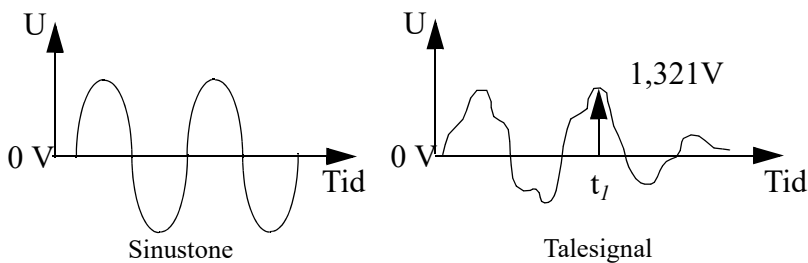
6.2 AD-konverteren - fra analoge til digitale signaler

Analog til digitalkonvertere (AD-konvertere) er viktige komponenter for å omdanne analoge signaler til digitale tallverdier. La oss først se litt på *sampling* og hva *digital representasjon* av et signal er.

6.2.1 Sampling

Vi tenker oss at vi ønsker å måle spenningen på et batteri. Vi tar da et voltmeter som f.eks. gir oss en verdi lik 4,32 V. Om vi kunne måle nøyaktig nok, kunne vi tatt med flere siffer etter komma.

I stedet for å måle spenningen på et batteri, kan vi tenke oss å måle spenningen til et lydsignal eller en sinustone. Disse vil være kontinuerlige og varierende signaler som i prinsippet kan inneholde alle mulige signalnivåer, eller om vi omdanner dem til elektriske signaler, kan inneholde alle mulige spenningsnivåer.

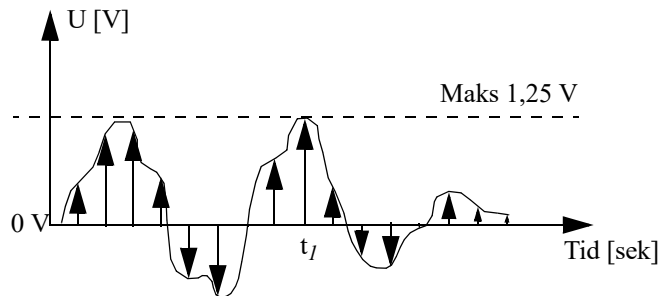


Figur 6.16 Eksempler på kontinuerlige og varierende signaler.



Dersom vi på et gitt tidspunkt måler amplitudeverdien til signalet (*øyeblikksverdien*), kan vi f.eks. tenke oss at vi måler verdien 1,321 V. Vi kan til og med tenke oss at vi måler kontinuerlig. Da vil vi få en strøm av tall som endrer seg hele tiden i takt med signalnivået (*øyeblikksverdiene*).

Normalt har vi ikke behov for å måle hele tiden, men i enkelte punkter. Dette kalles å *punktprøve* eller "*sample*" signalet. på denne måten gjør vi om et kontinuerlig varierende signal (analogt signal) til en rekke enkelt verdier eller tall (digitalt signal).



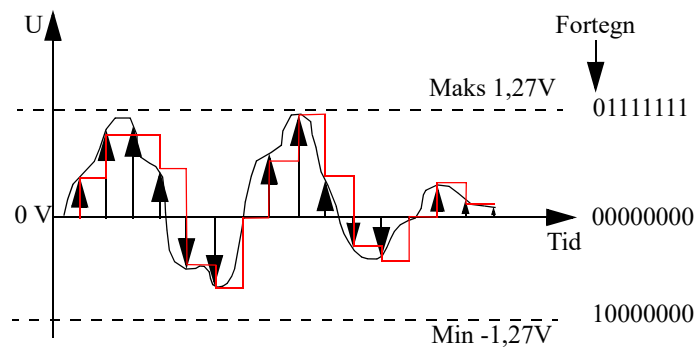
Figur 6.17 Punktprøving av et kontinuerlig signal.

En kan nå spørre seg hvor ofte må vi punktprøve et signal, for at vi senere skal være istand til å gjenskape det analoge signalet mest mulig nøyaktig.

Uregelmessige signaler, som f.eks. tale inneholder mange forskjellige frekvenser. Dersom vi ønsker å gjenskape signalet "eksakt" må vi minst punktprøve signalet dobbelte fort som den høyeste frekvensen signalet inneholder. Skal vi f.eks. punktprøve en sinus med en frekvens på 1000 Hz, må vi sample denne med en frekvens på minst 2000 sampler i sekundet. Skal vi punktprøve et talesignal som inneholder frekvenser fra 150 - 2 500 Hz, må vi minst bruke en punktprøvingshastighet på 5 000 punktprøver i sekundet.

Skal vi gjenskape et analogt signalet som er punktprøvd med en punktprøvingshastighet som er det dobbelte av den høyeste frekvensen i signalet, trenger vi et **ideelt** lavpassfilter med en båndbredde på 1000 Hz, som er nesten umulig å lage. En pleier derfor å *oversample* signalet, slik at kravet til filteret reduseres og lar seg realisere. En vil derfor punktprøve et signal på f.eks. 1000 Hz med en punktprøvingshastighet på 3000 Hz.

I eksemplet over har vi tenkt oss at vi opererer med mange siffer etter komma for å få avlesningen så nøyaktig som mulig. Dette er ofte ikke nødvendig. Det kan f.eks. være tilstrekkelig å ta med to siffer etter komma, i tillegg til at vi begrenser signalnivået på inngangen til maksimalt +/- 1,25 V. Dermed trenger vi ikke å angi signalet med mer enn 250 ulike nivåer.



Figur 6.18 Velger nærmeste nivå med en nøyaktighet på to siffer etter komma



Vi ønsker dessuten å angi signalnivået i det binære tallsystemet. Vi vet da at vi må operere med 8 bit for at vi skal kunne angi minst 250 nivåer. 8 bit klarer å angi $2^8 = 256$ ulike nivåer. Noen ganger ønsker vi dessuten at høyeste digitale siffer angir fortegnet. Et ett tall i høyeste siffer angir negative verdier og en null, positive verdier.

Vi ønsker dermed at signalverdien 0 V skal

	A)	B)
angis med tallet	+ 1,27 V -0111 1111	+ 1,27 V -0111 1111
00000000. Signalver-	+ 0,03 V -0000 0011	+ 0,03 V -0000 0011
dien 10 mV med tallet	+ 0,02 V -0000 0010	+ 0,02 V -0000 0010
00000001, 20 mV med	+ 0,01 V -0000 0001	+ 0,01 V -0000 0001
tallet 00000010 osv. Til-	0,00 V - 0000 0000	0,00 V - 0000 0000
svarende vil	- 0,01 V -1000 0001	- 0,01 V -1111 1111
signalverdien -0.10 mV	- 0,02 V -1000 0010	- 0,02 V -1111 1110
angis med tallet	- 0,03 V -1000 0011	- 0,03 V -1111 1011
10000001, verdien -20	- 1,27 V -1111 1111	- 1,27 V -1000 0000
mV med tallet		
10000010 osv. som vist		
i figur 6.19. A.		

Toer's komplement

Det er imidlertid vanligere å bruke såkalt

toer's komplement (two's complement). Også i dette tilfellet angir det mest signifikante siffer (MSB - lengst til vestre) fortegnet, 0 positivt heltall, 1 negativt heltall. Det minst positive tallet større enn 0 er i vårt tilfelle 0000 0001. Det minst negative tallet vil som vi så tidligere være 1000 0001. I toer's komplement vil imidlertid det minst negative tallet bli 1111 1111 (figur 6.19 B). Årsaken til at vi ofte bruker denne notasjonen for binære negative heltall, er at vi slipper å tenke på tallets fortegn når vi adderer eller subtraherer binære tall. Det vil gå av seg selv.

Toer's komplement for et negativt heltall lages ved at vi tar det tilsvarende positive binære heltallet og inverterer alle sifrene ($0 \Rightarrow 1$ og $1 \Rightarrow 0$). Dernest legges 1 til tallet vi da får og vi har fått den negative verdien av tallet representert ved toer's komplement.

La oss se et eksempel på hvordan AD-konverteringen kan gjøres i praksis.

6.2.2 AD-konverteren

Det finnes mange måter dette kan gjøres på. Vi skal her se på en som er den enkleste og raskeste, men også den dyreste: *Flash konverteren*.

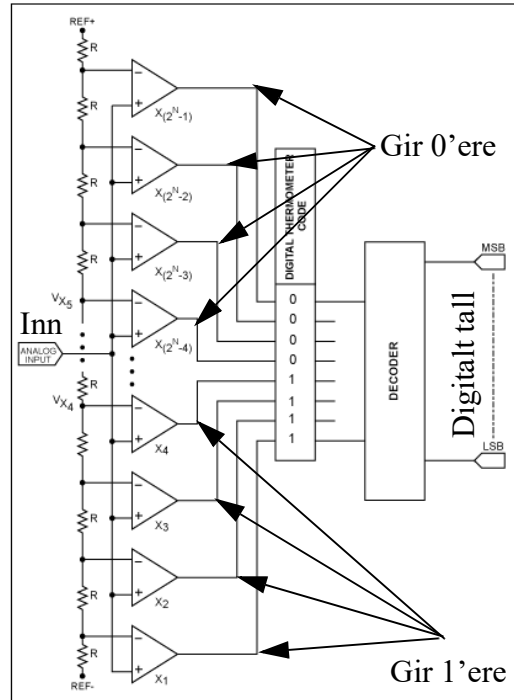
Signalet føres parallelt inn til en rekke komparatorer. Tersklinngangen til den enkelte komparatoren er koblet til en rekke av seriekoblede motstander som setter terskelverdiene til hver av komparatorene til riktig nivå. Tar vi utgangspunkt i eksempelet vårt hvor spenningsområdet -1,27–+1,27 deles opp i 255 ulike nivåer, så må forskjellen mellom hver terskelverdi være 0,01 V.



Dersom signalet inn på en komparator overskrider komparatorens terskelverdi, settes utgangen til 1 (dvs. høy spenning f.eks. 5 V). Om den er under terskelen, settes utgangen til 0 (dvs. lav spenning f.eks. 0 V). Alle komparatorer som har et referansenivå som er lavere enn det innkommende signalet, gir 1'ere (nederst), og alle komparatorer som har et referansenivå høyere enn det innkommende signalet gir 0'ere (øverst). Deretter følger en dekode som omdanner rekken av 0 og 1 til et digitalt tall som antydnet i figur 6.19.

En AD-konverter av denne typen som skal kunne skille mellom 256 forskjellige nivåer, må ha 256 komparatorer. Dekoderen gjør om de 256 0'erne og 1'erne til 8 bit som er tallet som angir spenningsnivået på inngangen. Om 8 bit gir for dårlig nøyaktighet, så finnes det AD-konvertere med både 10, 12, 14 og 16 bit.

AD-konverteren i CanSat er av en litt annen type, som er enklere og billigere, men ikke på langt nær så rask som denne.



Figur 6.20 Flash A/D-konverter





7 Sensorer

I denne delen skal vi se på ulike typer sensorer for bruk i mange sammenhenger. Noen av dem vil egne seg for å inkludere i CanSat. Beskrivelsen er langt fra komplett.

Det finnes en rekke leverandører av sensorer. Følgende kan være vært å oppsøke:

- **ELFA:** <http://www.elfa.se>
ELFA er et svensk komponentfirma som leverer raskt, men som ikke nødvendigvis er billigst. Har en meget oversiktlig og fyldig nettbutikk.
- **RS Norway:** <http://no.rs-online.com/web/home.html>
RS er et internasjonalt firma med en norsk filial. Firmaet er primært et komponentleverandør, men leverer raskt og greit. Har en fyldig nettbutikk.
- **Robot Electronics:** <http://www.robot-electronics.co.uk/index.html>
Er et engelsk firma som har spesialisert seg på robotkomponenter for hobbybruk og Arduino-komponenter.
- **Sparkfun Electronics:** <http://www.sparkfun.com/categories/23?page=1>
Dette er et amerikansk firma som leverer alle mulige sensorer, ofte leverte på små kort med digitalt grensesnitt som lett lar seg integrere med mikrokontrollere.
- **Tech supplys:** <http://www.rev-ed.co.uk/picaxe/>
Er et engelsk firma som leverer billige komponenter spesielt rettet mot PICAXE-moduler. Utvalget er relativt beskjedent, men prisene er lave.

La oss kort oppsummere prosessorens grensesnitt:

7.1 Kontrollkortets og sensorkortets grensesnitt (CanSat)

7.1.1 Analogt grensesnitt:

Tre av de åtte analog til digital-konverterne (ADC) (10 bit) er gjort lett tilgjengelig på det medfølgende sensorkortet, hvor to er benyttet til temperatur- og trykksensoren. ADC har følgende spesifikasjon:

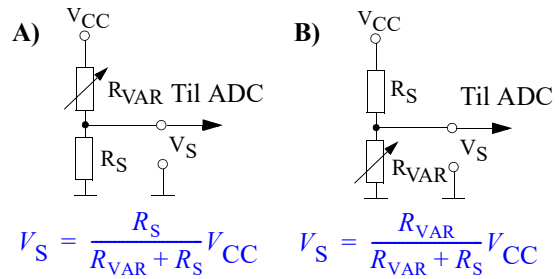
- 8 *multipleksede* innganger (dvs. at inngangene i tur og orden bytter på å bruke ADC'en)
- Spenningsområde: 0 – 5V (når $V_{cc} = 5\text{ V}$)
- Oppløsning: 10 bit, dvs. $\approx \pm 10\text{ mV}$ ($\pm 2\text{ LSB}$)
- *Konverteringstid:* $< 230\ \mu\text{s}$ (13 – 230 μs) dvs. maks 76,9 ksps (kilo samples pr. sekund) med begrenset oppløsning.

Spenningen inn på ADC'en måles mellom inngangen og jord. Enkelte sensorer vil også ha behov for supplyspenning ($V_{cc} = 5\text{ V}$) som er gjort tilgjengelig på sensorkortet. For nærmere diskusjon se omtalen av de enkelte sensorene og databladet for mikrokontrolleren [4] side 245.



Spenningsdeleren:

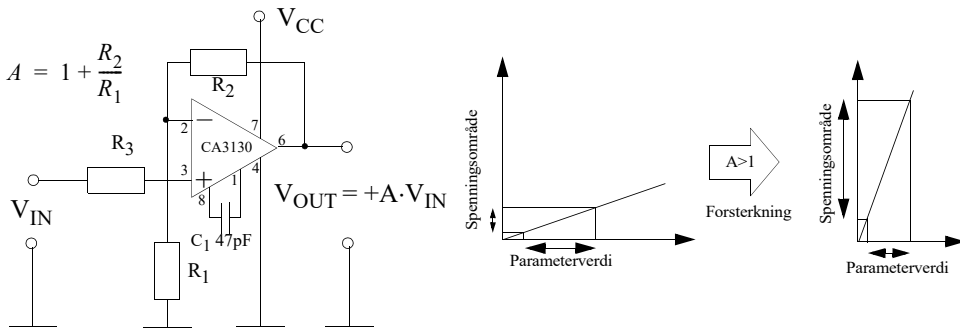
Mange sensorer er rent resistive. Dvs. at det aktuelle fenomenet (temperatur, fuktighet, trykk osv.) endrer sensorens indre resistans eller ledningsevne. Noen sensorer har inkludert elektronikk som konverterer endring i ledningsevne til en endring i spenning. Enklere sensorer krever ofte bruk av en spenningsdeler til å utføre konverteringen fra endring i resistivitet til endring i spenning. En slik spenningsdeler er vist på figuren til høyre. R_{VAR} er motstanden som avhenger av den aktuelle parameteren, mens R_S er en valgt fast seriemotstand. V_{CC} er batterispenningen. Ved hjelp av ligningene kan en beregne spenningen ut av spenningsdeleren når motstandsverdiene er kjent.



Økt dynamisk område (forsterkning):

Enkelte sensorer kan ha relativt liten variasjon innenfor parameterens aktuelle variasjonsområde. F.eks. kan vi være interessert i å måle temperaturer mellom 0 og 30°C, hvilket igjen kan gi en spenningsvariasjon på f.eks. 300 mV. Siden det maksimale utslaget til ADC'en er hele 5 V vil vi dermed bare utnytte en liten del av ADC'ens totale område. Vi sier at vi ikke får utnyttet ADC'ens *dynamiske område* (0–5V). Dette kan vi bøte på ved å forsterke signalet. Med en forsterkning på 10x vil en variasjon på 300 mV bli til 3 V. Dermed får vi utnyttet ADC'en bedre.

I denne sammenhengen er det praktisk å benytte integrerte *operasjonsforsterkere* som leveres i mange utgaver. Dette er små allsidige forsterkerelementer som kan brukes på mange ulike måter. Les mer om operasjonsforsterkere i vedlegg 6.1



Forsterkningen, A , bestemmes av motstandene R_1 og R_2 ut fra formelen:

$$A = 1 + \frac{R_2}{R_1} \quad (7.1)$$

R_3 er relativt ukritisk, men det kan være gunstig å velge denne lik parallellkoblingen av R_1 og R_2 .

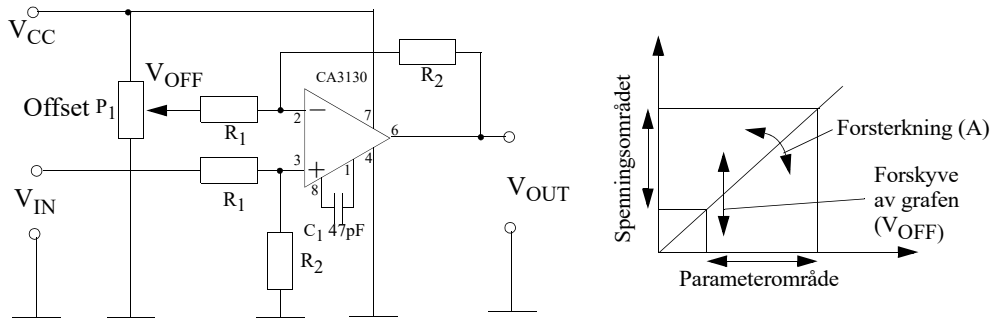


$$R_3 = \frac{R_1 \cdot R_2}{R_1 + R_2} \quad (7.2)$$

I eksempelet over er vist en *ikke inverterende forsterker*. Noen ganger kan det være enklere å benytte en inverterende forsterker (se vedlegg 6.1). I dette eksempelet er benyttet operasjonsforsterkeren CA3130 en vel så egnet forsterker er TS912¹⁷ som inneholder to like forsterkere.

Forskyve nullpunktet (offset)

I tillegg til å øke forsterkningen kan vi ha behov for å forskyve nullpunktet til spenningen fra sensoren. Dette kan gjøres ved å trekke fra eller legge til en spenning. På figuren til høyre trekker vi spenningen V_{OFF} fra V_{IN} i tillegg til at vi forsterker signalet.



Forsterkningen, A , er gitt av ligning (7.3):

$$A = -\frac{R_1}{R_2} \quad (7.3)$$

Dermed får vi:

$$V_{OUT} = A \cdot (V_{IN} - V_{OFF}) \quad (7.4)$$

I dette eksempelet er V_{OFF} justerbar med et potensiometer P_1 .

7.2 Innledende betraktninger om sensorer

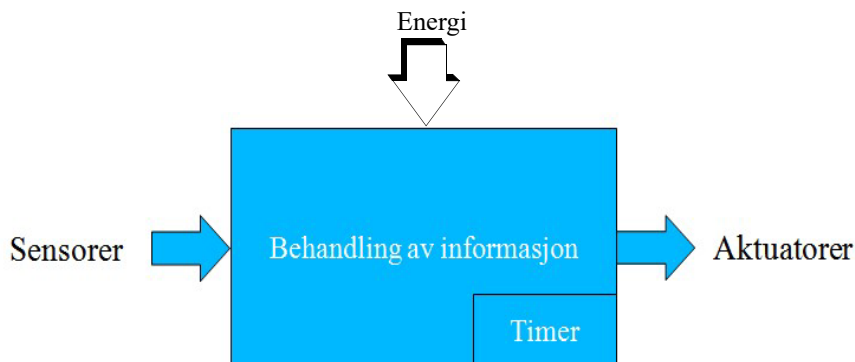
7.2.1 Hva er en sensor?

La oss først sette sensoren inn i en sammenheng. På figuren under ser vi i midten en enhet for *behandling av informasjon*. Til venstre *sensorer* og til høyre *aktuatorer*. En sensor er en enhet som registrerer kvantiteten til en fysisk størrelse i omgivelsene (f.eks. temperatur, lysstyrke, trykk o.l.), mens en aktuator er i denne sammenheng, en innretning som utfører en aksjon (f.eks. tenner

¹⁷.Benyttet ved NAROM.



lys, åpner en dør, lager en lyd, viser bilde o.l.). I noen tilfeller finnes en klokke (timer) som f.eks. gjør at aksjoner utsettes i tid. I tillegg må systemet tilføres energi (f.eks. fra batter, solceller, strømnettet e.l.). *Databehandlingsenheten bestemmer hvilken aksjon aktuatorene skal utføre og til hvilken tid på bakgrunn av informasjon innhentet fra sensorene.*



Nesten uansett hvilken mekanisk, elektrisk eller elektronisk hjelpemiddel vi tar for oss så kan det puttes inn i denne modellen. La oss se på noen eksempler som drar definisjonen på sensor langt.

- **Strykejernet** “sensor” kan være av/på bryteren og termostaten. Aktuatoren er varmelementet som gjør at jernet blir varmt.
- **Radioens** “sensorene” er antennen som fanger opp radiosignalene. Det kan også være alle knappene som styres av lytteren. På bakgrunn av informasjonen fra antenne og knapper, gir den ut et hørbart signal, som i denne sammenhengen er lydsignaler i høyttaleren (aktuatoren).
- **Den automatiske døråpneren** har gjerne en bevegelsessensor som registrerer at noen nærmer seg døra. Denne informasjonen behandles og systemet bestemmer at motorer eller hydraulikk (aktuator) skal åpne dørene.
- **Trafikklys** kan være et eksempel på et system hvor sensordelen kan være noe uklare dersom lyset kun slavisk gjennomgår en sekvens. Dette blir enklere dersom det er knapper for å be om grønt lys for fotgjengere eller magnetfeltsensorer i gatelegemet som registrerer kjøretøyer. Aktuatoren er lysene og ev. lydsignaler for fotgjengere.

Finn gjerne flere eksempler og diskuter hvor grensen bør gå for hva vi kan kalle en sensor.

Definisjon på sensor og aktuator

Følgende er et forsøk på en forenklet definisjon på en sensor og en aktuator:

Definisjon av sensor:

En sensor er en transduser som omdanner én fysisk størrelse til en annen fysisk størrelse som gjerne er elektrisk målbar.

Definisjon av aktuator:

En aktuator er en transduser som omdanner én fysisk størrelse (ofte elektrisk) til en annen fysisk størrelse som kan utføre en oppgave eller et mekanisk arbeid.



Veien fra en fysisk størrelse kan noen ganger være kronglete. Lista under viser eksempler på mulige veier fra den fysiske størrelsen til et målbart signal.

En fysisk størrelse, det være seg lys, temperatur, lyd, partikler eller noe annet, vil i sensoren omdannes til en av størrelsene til venstre i lista under:

Fra fysisk størrelse → endring i:

- resistans → spenning/strøm
- induktans → resonanskrets → frekvens/fase (telling eller tid)
- kapasitans → resonanskrets → frekvens/fase (telling eller tid)
- piezo-resistivitet → spenning
- piezo-elektrisk (spenning)
- utvidelse (elastisitet) → resistivitet → spenning
- temperatur (termisk) → resistivitet/ledningsevne → spenning/strøm
- bevegelse → telling eller tid

Selv om vi den fysiske størrelsen skaper en endring i resistivitet eller kapasitet eller lignende, så er ikke dette noe vi uten videre kan måle. Vi må ofte gå veien om måling av *strøm*, en *telling* eller måling av *tid*. Disse vil så kunne representere den opprinnelige størrelsen.

7.2.2 Egenskaper ved sensorer

Sensorer kan karakteriseres med ulike parametere, her er de viktigste:

Måleområde:

Måleområdet til en sensor er området mellom minste og største verdi av den målte størrelsen som ikke gir større avvik enn det vi kan tillate. Øker vi tillatt avvik mellom målt verdi og virkelig verdi, kan som oftest måleområdet utvides. *F. eks. kan -25 °C til $+125\text{ °C}$ være måleområdet for en temperatursensor. Innen dette området skal ikke avviket fra den virkelige verdien være større enn $\pm 2\text{ °C}$.*

Oppløslighet:

Dette er den minste målbare verdi av den fysiske størrelsen som sensoren er i stand til å registrere, og uttrykkes gjerne som en prosentandel av sensorens totale måleområde. *F. eks. kan $0,15\text{ °C}$ være en minste forskjell som en temperatursensor kan måle. Av et måleområde på 150 °C vil dette gi en oppløselighet på 1 ‰ .*

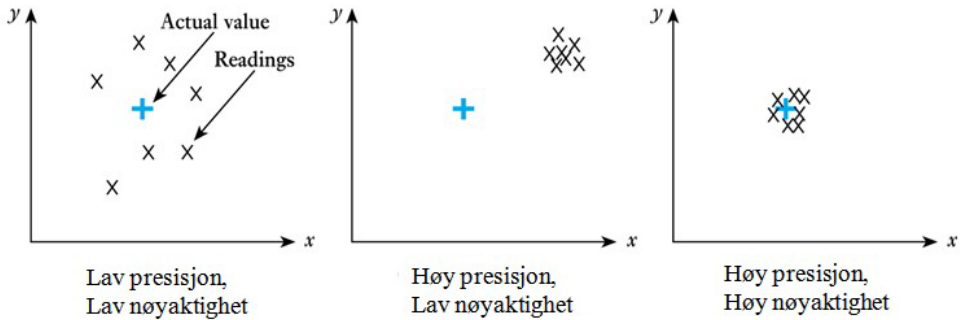
Målefeil:

Målefeilen er forskjellen mellom målt verdi og virkelig verdi. En målefeil kan deles inn i *tilfeldige* og *systematiske* feil. Tilfeldige feil er ofte spredt omkring den riktige verdien og hyppigheten vil avta når avstanden til den riktige verdien øker. Tilfeldige feil vil ha en middelvei nær den riktige verdien. Systematiske feil er spredt omkring et målepunkt som ligger tilside for den riktige verdien. Midling av flere målinger vil derfor ikke gi riktigere verdier.



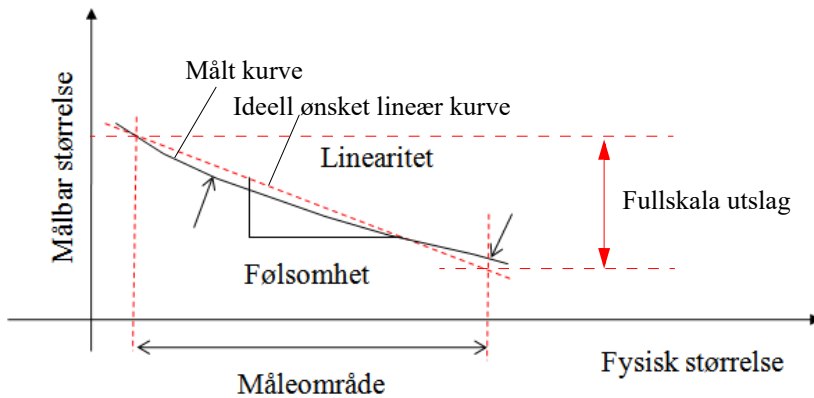
Presisjon og nøyaktighet:

Figuren under viser forskjellen på en måleseries *presisjon* og *nøyaktighet*.



Linearitet:

Ofte er det praktisk at det er en *lineær* sammenheng mellom verdien til den fysiske størrelsen og signalet ut av sensoren. Dette er ikke alltid tilfelle. Men heller ikke en sammenheng som i utgangspunktet skal være det vil være i praksis være helt lineær. Figuren under viser hvordan en kurve avviker fra den lineære ønskede verdien.



Vi ser at den målte kurven avviker fra den ideelle. Dette avviket måles mellom maksimale avvik på begge sider av den lineære kurven. Avviket angis i prosent av *fullskala verdien*.

Følsomhet:

Følsomheten til en sensor angis som endring av signalet ut av sensoren for en gitt forandring av den fysiske størrelsen inn på sensoren (se figuren over). *F.eks. kan en temperatursensor ha en følsomhet på 10 mV/°C.*



7.3 Temperaturfølsomme sensorer

I dette avsnittet skal vi se hvordan vi kan måle temperatur. Dette kan enten gjøres ved å benytte en resistiv sensor hvor motstandsmaterialet i resistoren endrer seg med temperaturen. Eller hvor det aktive materialet er en halvleder som endrer ledningsevne med temperaturen. Mens noen temperatursensorer kun består av en motstand/halvleder, inkluderer andre elektronikk og kan levere en spenning som varierer med temperaturen.

7.3.1 Temperaturfølsom motstand (NTC og PTC -motstander)

Metaller vil normalt ha økende resistans med økende temperatur. I et halvledermateriale vil flere ladningsbærere løftes opp i ledningsbåndet slik at ledningsevnen går opp, dvs. at resistansen blir mindre.

De fleste motstandsmaterialer endrer resistans som funksjon av temperaturen. Som regel er dette uønsket, men i noen spesielle tilfeller ønsker man nettopp en slik variasjon og utformer komponenten og materialet deretter. Slike motstander brukes også i forbindelse med måling eller deteksjon av temperaturendringer, eller til å motvirke uønsket temperaturdrift i elektronisk utstyr.

- NTC - Negative Temperatur Coefficient, dvs. at resistansen avtar med økende temperatur.
- PTC - Positive Temperatur Coefficient, dvs. at resistansen øker med økende temperatur.

NTC-motstanden

NTC-motstander er laget av et materiale hvis resistivitet varierer sterkt med temperaturen. Som navnet sier (Negative Temperature Coefficient - NTC) så avtar resistansen med økende temperatur.

NTC-motstander er derfor vanligvis bygget opp som en polykrytalinsk *halvleder* som kan bestå av en blanding av krom, mangan, jern, kobolt og nikkel, som sintres¹⁸ sammen med et plastisk bindemiddel.

En forenklet sammenheng mellom resistansen (R) og temperaturen (T) kan uttrykkes som:

$$R = Ae^{B/T} \quad (7.5)$$

hvor A og B er "konstanter" bestemt av materialet og temperaturen. Konstantene kan betraktes som *tilnærmet konstante* innen begrensede temperaturområder.

I datablader for NTC-motstander oppgis gjerne resistansen (R_T) for en referansetemperatur (T_T). I et temperaturområde rundt denne referansetemperaturen antas B -verdien å være tilnærmet konstant ($B_{25/85}$ – B -verdien er tilnærmet konstant innen området 25°C til 85°C).

Vi kan da sette opp følgende:

$$R = Ae^{\frac{B_{25/85}}{T}} \quad (7.6)$$

18. Sintring betyr at metallpulver knyttes sammen ved hjelp av oppvarming, men uten å smelte.



$$R_r = A e^{\frac{B_{25/85}}{T_r}} \quad (7.7)$$

Ved å eliminere A fra disse uttrykkene, kommer vi fram til følgende sammenheng, løst med hensyn til resistansen R :

$$R = R_r \cdot e^{\left(\frac{B_{25/85}}{T} - \frac{B_{25/85}}{T_r}\right)} \quad (7.8)$$

Dette uttrykket går under betegnelsen *Beta-formelen*.

Når vi skal beregne verdien for en NTC-motstand ved en gitt temperatur, slår vi opp B -verdien, R_r og T_r i databladet, sørger for at de aktuelle temperatuere ligger innenfor området til B -verdien, og beregner R ved å sette inn ønsket temperatur T . Temperaturen angis i grader Kelvin.

RH16

Fra databladet¹⁹ for RH16 finner vi følgende: R_{25} er referansemotstand (R_r) ved 25 °C ($T_r = 298$ K):

	R_{25}				$B_{25/50}$	$B_{25/85}$	τ [sek]	
RH16	3G202 * *	2k Ω	○	○	○	3,450K \pm 1%	3,488K	6
	6D502 * *	5k Ω	○	○	○	3,930K \pm 1%	3,941K	6
	3H103 * *	10k Ω	○	○	○	3,450K \pm 1%	3,486K	6
	6E103 * *		—	○	○	3,950K \pm 1%	4,001K	6
	3U503 * *	50k Ω	○	○	○	3,950K \pm 1%	4,025K	6
	3U803 * *	80k Ω	○	○	○	3,950K \pm 1%	4,025K	6
	4A104 * *	100k Ω	—	○	○	4,020K \pm 1%	4,099K	6
	4L304 * *	300k Ω	—	○	○	4,550K \pm 1%	4,629K	6

Figur 7.1 Datablad for NTC-motstand RH16, 2–300 k Ω

Med disse dataene kan vi skrive:

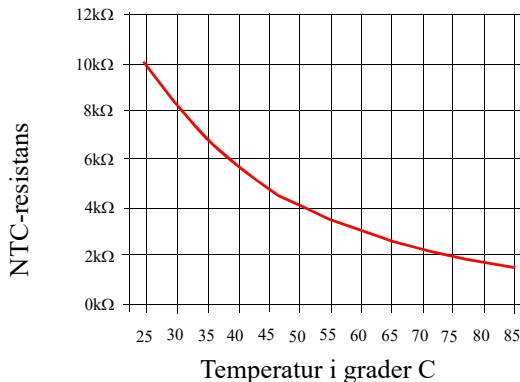
$$R = 10k \cdot e^{\left(\frac{3486}{T} - \frac{3486}{298}\right)} \quad (7.9)$$

hvor $B_{25/85} = 3486$ (RH16 – 10 k Ω) og referansetemperaturen $T_r = 298$ K.

19. Databladet er hentet fra: <http://www.elfa.se/pdf/60/06027916.pdf>



Dersom vi beregner verdier for R i temperaturområdet 25°–85°C, får vi følgende graf:



Figur 7.2 NTC resistans som funksjon av temperaturen RH16 10 kΩ

En annen viktig parameter for NTC-motstander, er hvor raskt resistansen endrer seg med temperaturen. Denne parameteren betegnes *NTC-motstandens tidskonstant* (τ), og angir den tiden det tar for resistansen og endre seg til 63,2% av den nye resistansen etter at temperaturen har endret seg 1 K (Kelvin) over omgivelsestemperaturen. En antar at temperaturendringen ikke er forårsaket av indre oppvarming på grunn av elektrisk strøm som flyter gjennom motstanden.

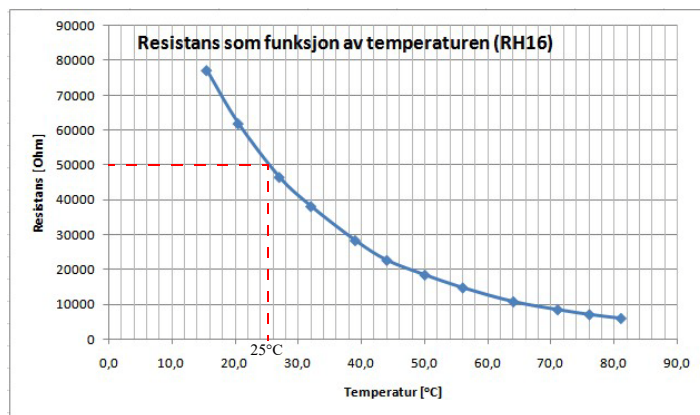
I vårt eksempel er $\tau = 6$ sek. (se tabellen i figur 7.1). Dette er kanskje en av de største ulempene med denne teknologien, den har relativ lang tidskonstant.

Kalibrering

Vi har gjort målinger på RH16 – 50 kΩ og fått følgende kurve.

Målingen ble gjort med et Ohm-meter. NTC-motstanden ble senket ned i et glass med ca. 2 dl springvann oppvarmet med en vannkoker før det fikk lov til å kjøle seg i romtemperatur. Vi ser at ved 25°C er verdien omtrent 50 kΩ som spesifisert i databladet.

En datalogger vil også kunne benyttes.



Figur 7.3 Resistans som funksjon av temperatur for RH16 – 50 kΩ

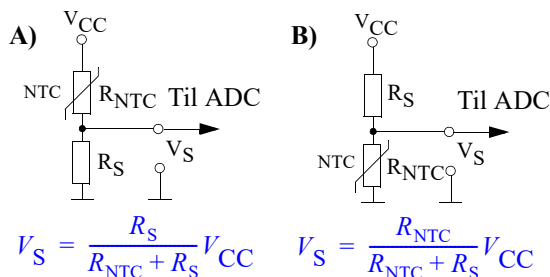


En kan gjøre et lignende regnestykke for NTCLE100E3103JB0 termistoren (VISHAY)²⁰. Dette er NTC-motstanden har 10 kΩ ved 25°C og en toleranse (nøyaktighet) på 5% (J). De strategiske parametrene for denne typen er vist i tabellen under.

ELECTRICAL DATA AND ORDERING INFORMATION								
R ₂₅ [Ω]	B _{25/85} - VALUE		UL APPROVED [Y/N]	12NC ORDERING CODE 2381 640 6.... (1)	SAP MATERIAL NO. NTCLE100E3..... (2)	COLOR CODE (3)		
	[K]	[± %]				I	II	III
680	3560	1.5	N	*681	681*B0	Blue	Grey	Brown
1000	3528	0.5	N	*102	102*B0	Brown	Black	Red
1500	3528	0.5	N	*152	152*B0	Brown	Green	Red
2000	3528	0.5	N	*202	202*B0	Red	Black	Red
2200	3977	0.75	Y	*222	222*B0	Red	Red	Red
2700	3977	0.75	Y	*272	272*B0	Red	violet	Red
3300	3977	0.75	Y	*332	332*B0	Orange	Orange	Red
4700	3977	0.75	Y	*472	472*B0	Yellow	Violet	Red
5000	3977	0.75	Y	*502	502*B0	Green	Black	Red
6800	3977	0.75	Y	*682	682*B0	Blue	Grey	Red
10 000	3977	0.75	Y	*103	103*B0	Brown	Black	Orange
12 000	3740	2	Y	*123	123*B0	Brown	Red	Orange
15 000	3740	2	Y	*153	153*B0	Brown	Green	Orange
22 000	3740	2	Y	*223	223*B0	Red	Red	Orange
33 000	4090	1.5	N	*333	333*B0	Orange	Orange	Orange
47 000	4090	1.5	N	*473	473*B0	Yellow	Violet	Orange
50 000	4190	1.5	N	*503	503*B0	Green	Black	Orange
68 000	4190	1.5	N	*683	683*B0	Blue	Grey	Orange
100 000	4190	1.5	N	*104	104*B0	Brown	Black	Yellow
150 000	4370	2.5	Y	*154	154*B0	Brown	Green	Yellow
220 000	4370	2.5	Y	*224	224*B0	Red	Red	Yellow
330 000	4570	1.5	N	*334	334*B0	Orange	Orange	Yellow
470 000	4570	1.5	N	*474	474*B0	Yellow	Violet	Yellow

Oppkobling mot ADC

Siden grensnett til kontrolleren krever en spenning, kobles NTC-motstanden i serie med en motstand som vist i figuren under. Velg verdien på seriemotstanden lik den nominelle verdien på NTC-motstanden (R₂₅). Spennings nivået V_S beregnes fra formlene som antydnet på figuren under. Legg merke til at oppkoblingen på tegning A gir økende spenning V_S med økende temperatur, mens oppkoblingen i tegning B gir fallende spenning med økende temperatur.



På bakgrunn av ligningene foran kan en utvikle et uttrykk for temperaturen som funksjon av spenningen som kan legges inn i prosessoren.

7.3.2 Integrrert temperatursensor LM35DZ

Dette er en integrert temperatursensor med følgende nøkkelparametere:

- Kalibrert i henhold til Celsius skalaen

20. Datablad: <http://www.vishay.com/docs/29046/29046.pdf>



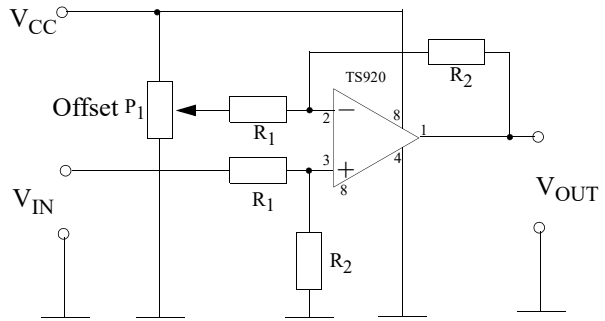
For å optimalisere sensoren til ADC'en innenfor det aktuelle temperaturområdet må vi gjøre følgende:

1. Bestemme aktuelt temperaturområde
2. Legge inn forsterkning slik at temperaturområdet dekker mest mulig av ADC'ens dynamiske område 0–5 V.
3. Forskyve skalaen (*offset*) slik at den faller innenfor ønsket spenningsområde.

Se nærmere i *Sensor Circuits for CanSat use* [1].

Følgende kretsløsning kan ev. brukes til å justere forsterkning og offset: Forsterkningen er gitt av følgende uttrykk:

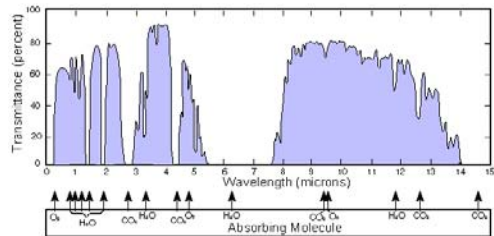
$$\text{Forsterkning} = \frac{R_2}{R_1} \quad (7.12)$$



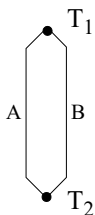
7.3.3 IR-sensor

Varmestråling er som kjent elektromagnetisk stråling i den infrarøde delen av spekteret. Et relativt bredt område av spekteret som strekker seg fra 740 nm på kanten av den røde delen av det synlige delen av spekteret, og til ca. 300 μm , eller fra 1 til 400 THz.

Figuren til høyre viser i hvilke deler av IR-spekteret, fra synlig lys (<1 μm) og opp til 15 μm , som er transparent for stråling. Legg merke til de dype hullene i spekteret, som blokkerer for innstråling og som skyldes absorpsjon av H_2O , CO_2 og O_2 gass.



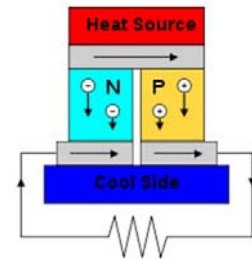
Alle gjenstander og mennesker sender ut IR-stråling (varme). Denne strålingen kan registreres av IR-sensorer, som f.eks. kan benyttes til å avsløre varmelekkasje fra bygninger, varmgang av elektrisk utstyr, måling av kroppstemperatur og som bevegelsessensorer mm.



ELFA og andre leverer en rekke ulike IR-sensorer som baserer seg på en såkalte *Seebeck-effekt*, som er en termoelektriske effekt. Effekten kalles også *Peltier-Seebeck-effekten* da den ble oppdaget samtidig av flere. **Thomas Johann Seebeck** (1770–1831) oppdaget effekten i 1821. Han

oppdaget at en kompassnål dreide seg når den nærmet seg en lukket krets hvor to forskjellige metaller, A og B, var koblet sammen i to punkter med forskjellig temperatur (vist på figuren til venstre). Det gikk altså en elektrisk strøm i kretsen.

Den samme effekten utnyttes i moderne IR-sensorer. I stedet for å bruke ulike metaller, benyttes p-dopet og n-dopet halvledermateriale, som vist på figuren til høyre.



Figuren viser to elementer, ett av hver type, som er seriekoblet av det grå metallet på topp og i bunn. Vi vet at p- og n-materialene i utgangspunktet er elektrisk nøytrale, men at n-dopet materiale har et *overskudd* med frie negative ladninger og p-dopet materiale har et *underskudd* av negative ladningsbærere. Det siste kan også betraktes som et overskudd av positive ladningsbærere (hull).

Dersom det skapes en temperaturforskjell i halvledermaterialet, vil det i den delen av materialet som blir oppvarmet, skapes flere frie ladningsbærere. Dvs. at den tilførte varmeenergien “løfter” elektroner ut av valensbåndet og inn i ledningsbåndet. For å jevne ut forskjellene begynner ladningene å diffundere fra den varme til den kalde delen av halvlederen. På denne måten oppstår et elektrisk felt inne i materialet. Når n- og p-materialet er seriekoblet, som vist i figuren, vil det totale elektriske spenningen øke. I et praktisk termoelektrisk element er en rekke slike enkelt-elementer seriekoblet, slik at en oppnår en tilstrekkelig høy spenning.

IR-sensorer som er i salg f.eks. hos bl.a. ELFA, utnytter denne effekten. Inne i en metallkapsel ligger et termoelektrisk elementet.

Varmestrålingen slippes inn i kapselen gjennom et vindu av silisium som er transparent for IR-stråling. Framsiden av termoelementet er dekket av et mørkt materiale som lett absorberer strålingen og blir oppvarmet. Baksiden av elementet kjøles av kapselen. Det aktive arealet kan typisk være $0,5 \text{ mm}^2$. Som vi har sett skapes en spenning når det oppstår en temperaturforskjell mellom for- og baksiden av elementet.

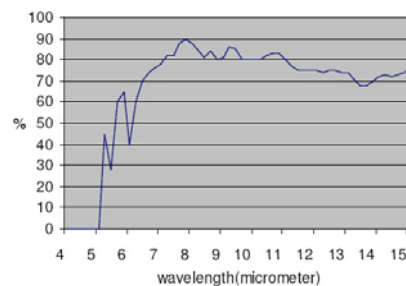


Denne spenningsforskjellen tas ut på IR-sensorens bein, og forsterkes opp og behandles av omkringliggende elektronikk.

IR-sensoren SMTIR 9901 er en slik komponent. Her er noen av spesifikasjonene til sensoren:

- Følsomhet: 110 V/W
- Responstid: 40 msek.
- Temperaturområde: -20 til +100 °C
- Bølgelengder > 5,5 μm (lavpassfilter)

Vi legger merke til at følsomheten synes svært høy. En må imidlertid huske på at effektforskjellene inne i kapselen er svært små.





Temporal temperaturskanning

Temporal temperaturskanning brukes til, på en rask måte, å måle kroppstemperaturen ved f.eks. feber. Det finnes både profesjonelt og kommersielt utstyr som måler kroppstemperaturen på denne måten. Målingen baserer seg på infrarød skanning av huden rett over *temporal arterien*, som kommer opp på framsiden av øret og ligger ca. 1 mm under huden i panna. Denne åra har en stabil blodstrøm uavhengig av ytre press. Den infrarøde strålingen måler energitapet fra huden. Siden temperaturtapet avhenger av romtemperaturen, må også den samtidig måles.



Modell HV-T36

Det har vært en del diskusjon på nettet om denne metoden er tilstrekkelig reproducerbar og gir god nok overensstemmelse med kjernetemperaturen i kroppen²¹.

7.4 Lysfølsomme sensorer

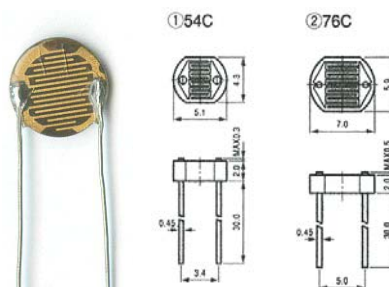
Deteksjon av lys kan gjøres på mange ulike måter. I dette avsnittet skal vi se hvordan vi kan bruke *LDR* (Light Dependent Resistor) og fototransistorer som lysfølsomme komponenter.

7.4.1 Fotomotstand (LDR - Light Dependent Resistor)

Grunnen til at vi velger å omtale en fotomotstand er at den gjennom tidene har vært en gjenganger i mange elektronikkprosjekter. Aktuelle leverandører av fotomotstander finnes i vedlegg B.

Fotomotstander har tradisjonelt vært laget av Cadmium-Sulfid (CdS) belagt med fingerelektroder som vist på figuren til høyre. I mørket vil stoffet CdS være omtrent isolerende og kan gi en motstand på over 1 M Ω . Belyser vi derimot stoffet, kan resistansen falle til under 1 k Ω .

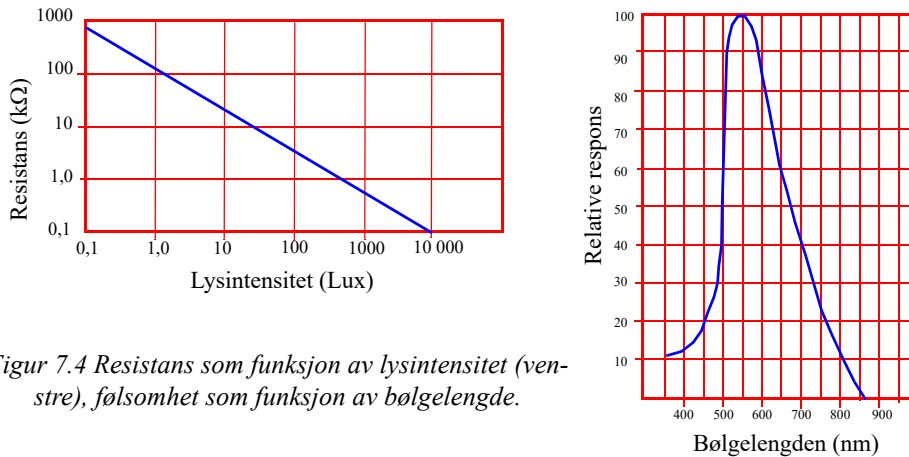
Årsaken er at fotoner (lys) med tilstrekkelig energi, eksisterer elektroner fra valensbåndet til ledningsbåndet, hvor de kan bevege seg fritt og bidra til ladningstransporten. Effekten er imidlertid ikke like framtrede for alle frekvenser. Til høyre på figur 7.4 ser vi at materialet er spesielt føl-



21. http://journals.lww.com/journaloftraumanursing/Abstract/2009/01000/Temporal_Artery_Scanning_Falls_Short_as_a.10.aspx



somt for lys i det synlige området av spekteret nær 540 nm (nanometer, 10^{-9} m). Vi ser også (til venstre på figuren) at det er en omtrent lineær sammenheng mellom lysstyrken målt i lux og resistansen (begge skalaer er logaritmiske).



Figur 7.4 Resistans som funksjon av lysintensitet (venstre), følsomhet som funksjon av bølgelengde.

Ut fra følsomhetskurven til høyre på figur 7.4, ser vi at fotomotstanden er spesielt effektiv i ett område nær 450 nm (grønn-blå).

Lysfølsomme motstander er imidlertid relativt langsomme. En endring i lysstyrken på noen μ sek, kan gi en responstid på opp til 100 msek. hos fotomotstanden. Men i mange tilfeller er dette mer enn godt nok.

For å konvertere endring i resistans til spenning, kan vi bruke en enkel spenningsdeler (se figuren under). Her trengs normalt ingen målebro eller forsterker for å registrere endring i resistans siden endringen er så stor.

Lysintensitet måles i lux. 1 lux er 1 lumen pr. m^2 .

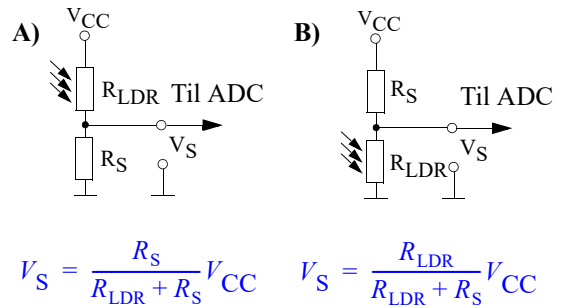
Dette tilsvarer:

- Fullt sollys 11 000 lux (eller ca. 1000 W/m^2)
- Sollyset en tidlig morgen 6 000 lux
- Belysningen i et TV-studio 1 000 lux
- Et godt opplyst kontor 400 lux
- Lyset fra en fullmåne 1 lux



Oppkobling mot ADC

Siden grensnittet til kontrolleren krever en spenning, kobles LDR-motstanden i serie med en motstand som vist i figuren til høyre. Velg verdien på seriemotstanden lik den nominelle verdien på LDR-motstanden. Spenningsnivået V_S beregnes fra formlene som antydnet på figuren. Legg merke til at oppkoblingen på tegning A gir økende spenning V_S med økende lysstyrke, mens oppkoblingen i tegning B gir fallende spenning med økende lysstyrke.



Kalibrering:

Utfordringen blir å finne en omregningsformel fra lysstyrke til spenning:

1. Mål spenning som funksjon av lysstyrke (krever lysmåler)
2. Bruk regresjon for å finne et best tilpasset funksjonsuttrykk
3. Lagg omregningsformelen inn i prosessoren

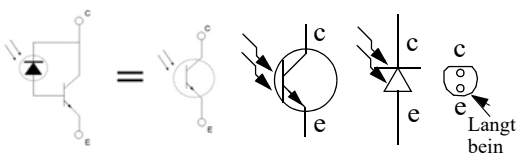
7.4.2 Fototransistorer


Denne type lysfølsomme halvlederkomponenter kan operere i to ulike måter (modus):

- Fotoledende modus, forspent i sperreretning
- Fotovoltaisk modus, forspent i lederetning (som hos solceller)

Fototransistor koblet i fotoledende modus m/ekstra transistor (PT204-6C)

En diode (PN-overgang) vil normalt lede strøm når den forspennes i lederetning og sperre når den forspennes i sperreretning. Likevel vil det alltid flyte en liten lekkasje- eller *reversstrøm* når den er forspent i sperreretningen. Dersom PN-overgangen eksponeres for lys, vil reversstrømmen øke med økende lysstyrke ved at energien i lyset øker antallet ladningsbærere. PN-overgangen er vanligvis følsom både for synlig og infrarød stråling.

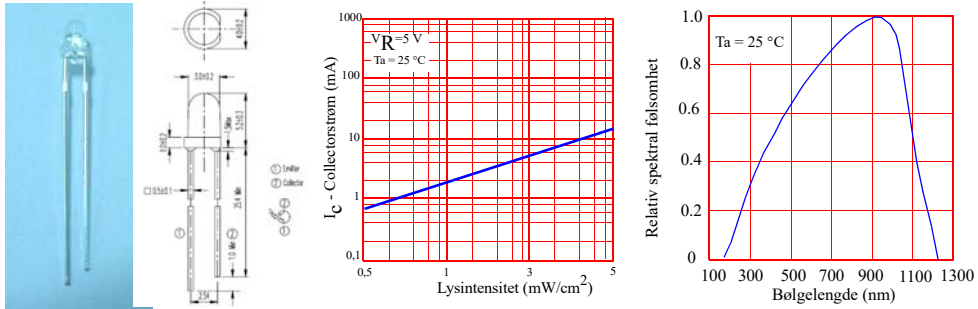


En fototransistor av denne typen bygges ofte opp av en PN-overgang (fotodiode) koblet i serie med basen på en transistor som vist i figuren lengst til venstre. Vi legger merke til at PN-overgangen () er forspent i sperre-

retning. Det er reversstrømmen til PN-overgangen som ledes inn på basen til transistoren og som forsterkes opp. På denne måten styres basestrømmen i transistoren av styrken til det innfallende lyset. Med andre ord vi bytter ut en strømførende baseledning med en varierende lysstyrke.



Som et eksempel har vi valgt å se på fototransistoren **PT204-6C** som leveres fra *Everlight* og forhandles bl.a. av ELFA til pris av et par kroner [12]. Et bilde av PT204-6C er vist til venstre på figuren under.



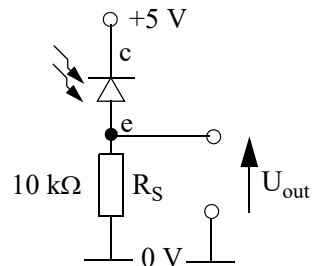
Reversstrømmen kan være meget lav. Transistoren forsterker reversstrømmen slik at strømmen gjennom transistoren blir betydelig større. Av diagrammet til venstre på figuren over, ser vi at strømmen i fototransistoren er ca. 0,7 mA ved en lysstyrke på ca. $0,5 \text{ mW/cm}^2$ (5 W/m^2) og 16 mA ved 5 mW/cm^2 (50 W/m^2)²².

Fototransistorens følsomhet er også avhengig av bølgelengden til lyset. Av figuren over til høyre ser vi at PT204-6C er spesielt følsom for bølgelengder på grensen av synlig og infrarødt lys. Synlig lys går fra ca. 400–700 nm. Toppen for PT204-6C ligger på ca. 900 nm. Vi ser imidlertid at den fungerer godt også i store deler av den synlige delen av spekteret.

Fototransistorer kan lages av ulike materialer og vil dermed bli følsomme for ulike bølgelengder. Tabellen under viser hvordan ulike materialer gir fototransistorer med ulike følsomhetsområder²³.

Silisium:	190–1100 nm
Germanium:	400–1 700 nm
Indium Gallium Arsenid:	800–2 600 nm
Blysvulfid	<1000–3600 nm

Dersom vi ønsker å bruke fototransistoren som lysmåler eller lysdetektor, kobler vi den opp som en spenningsdeler som vist på figuren til høyre. Vi har sett at jo mer lys fototransistoren utsettes for, jo større reversstrøm går det i dioden og gjennom seriemotstanden R_S , og jo høyere blir spenning U_{out} . Skal den brukes som lysmåler kan U_{out} kobles til en AD-konverter eller en forsterker om nødvendig. Brukt som lysdetektor kan den kobles til en komparator eller basen på en transistor.



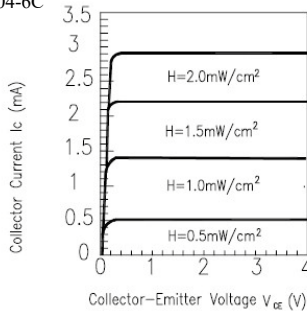
For å karakterisere en fototransistor, kan vi koble en variabel collector–emitterspenning over transistoren samtidig som vi måler collector strømmen (I_C) for ulike nivåer av innfallende lysmengde.

22. Til sammenligning kan nevnes at typisk lysstyrke i et rom kan være $2 - 10 \text{ W/m}^2$, mens lysstyrken utendørs en strålende sommerdag kan være typisk 1000 W/m^2 .

23. Se <http://en.wikipedia.org/wiki/Photodiode>



PT204-6C



Med en gitt mengde lys, vil strømmen i transistoren øke opp til et visst nivå hvor strømstyrken er uforandret selv om collector-emitter-spenningen økes. Vi har nådd fototransistorens metningsstrøm ved denne lysstyrken. Økes lysstyrken, vil vi få fram en lignende kurve, men med større metningsstrøm. Sammenhengen mellom metningsstrøm, lysstyrke og collector-emitterspenning for PT204-6C er vist på figuren til venstre.

Det er viktig å merke seg at en fototransistor av typen PT204-6C ikke vil levere noen spenning eller strøm med mindre den forspennes. Den er derfor en komponent med

en ledningsevne som endres med lysstyrken og vil ikke fungere i fotovoltaisk modus.

Fototransistorene SFH300 og SFH 203

Disse komponentene består av en ren PN-overgang (uten en ekstra transistor) og vil derfor kunne fungere både i fotovoltaisk og i fotoledende modus.

Brukt i fotovoltaisk modus vil disse komponentene *være små solceller og fungere som en strøm eller spenningskilde.*

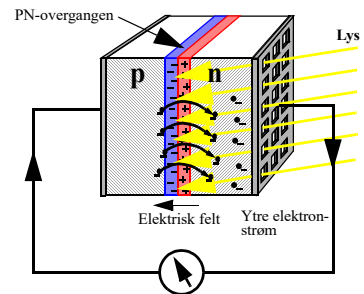
I fotovoltaisk modus vil disse fungere på følgende måte:

Fototransistorer (og solceller) er som oftest laget av grunnstoffet *silisium*. Vi tenker oss at vi lager to tynne skiver av silisium. Den ene skiven forurenses vi med *fosfor* som gjør at denne skiven *gjærne avgir elektroner*, vi sier at skiven er *n-dopet*. Den andre skiven forurenses vi med *bor* som medfører at denne skiven *lett tar til seg elektroner*, vi sier at skiven er *p-dopet*.

Når vi legger en n-dopet og en p-dopet skive inntil hverandre, får vi en *pn-overgang* (en *diode*). Når de to skivene berører hverandre vil det flyte en strøm av elektroner fra den n-dopede skiven over til den "elektronhungrige" p-dopede skiven. På grunn av ladningstransporten, vil det bygge seg opp en spenning over pn-overgangen. Etter hvert som spenningen øker, vil strømmen ladninger gå stadig tregere. Til slutt vil den stoppe helt opp. Foreløpig har vi ikke koblet til noen ytre krets.

I et smalt område nær pn-overgangen vil det nå være svært få fri ladningsbærere (elektroner). Området vil derfor virke som en sperre for strømmen gjennom overgangen. Dersom vi imidlertid belyser pn-overgangen, vil energien i lyset frigjøre nye elektroner i overgangsområdet. Disse vil strømme over til n-området som blir negativt ladet samtidig som p-området blir positivt ladet.

Om vi nå kobler en ledning mellom n-området og p-området, vil det gå en elektrisk strøm i ledningen som vist på figuren over. Denne *ytre* strømmen kan så benyttes til å drive en motor eller få en lyspære til å lyse. Vi har fått en solcelle.





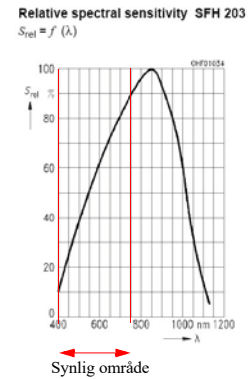
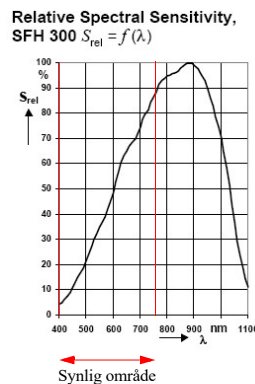
Forenklet kan vi si at pn-overgangen representerer et hinder for elektronene. Lys vil imidlertid tilføre elektronene energi slik at de kan passere over hinderet. Når de først er kommet over, klarer de ikke å komme tilbake, men må gå utenom gjennom den ytre ledningen hvor de gjør et arbeid på veien.

En fototransistor i fotovoltaisk modus vil fungere som en liten sol-celle, og vil, når den belyses, levere en spenning²⁴ på ca. 0,54 V. Kobler vi den inn i en ytre krets, vil den også gi en liten strøm.



Bildet til høyre viser en typisk fototransistor av denne typen, f.eks. SFH300 eller SFH203. Som vi ser så ligner den på en lysdiode.

Figurene til venstre viser følsomhetsdiagrammet som funksjon av frekvensen, for henholdsvis SFH300 (venstre) og SFH203 (høyre). Vi ser at de to er ganske like mht til båndbredde. Begge har de sitt mest følsomme område inne i den infrarøde delen av spekteret.

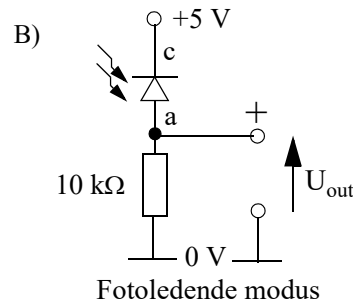
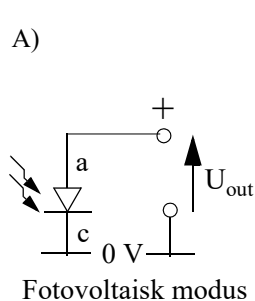


Oppkobling for karakterisering:

Denne typen fototransistor kan i prinsippet brukes på begge måtene:

- I *fotovoltaisk modus* koblet som en selvstendig strøm- eller spenningskilde.
- I *fotoledende modus* koblet som en lysstyrt transistor (fototransistor) koblet i sperreretning.

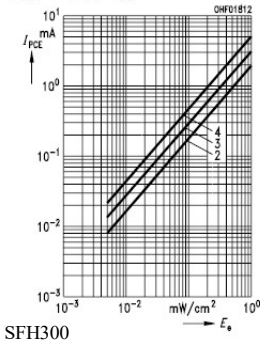
Figurene under viser de to alternativene:



24. Tomgangsspenning er den spenningen en måler over fotodioden når den ikke belastes

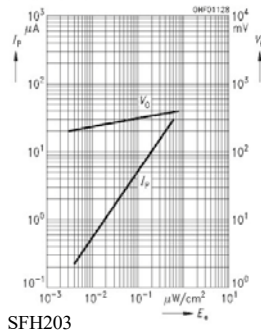


Photocurrent
 $I_{PCE} = f(E_e), V_{CE} = 5 \text{ V}$



SFH300

Photocurrent $I_p = f(E_e), V_R = 5 \text{ V}$
Open-circuit-voltage $V_L = f(E_e)$
SFH 203 FA



SFH203

Begge koblingene kan brukes som kilde til en AD-konverter eller forsterker.

Likevel anbefales alternativ B da denne gir større spenningsving, samtidig som den ikke går i metning ved kraftig belysning slik alternativ A gjør (den fotovoltaiske spenning vil flate ut ved ca. 0,54 V. Strømmen vil derimot øke lineært med lysstyrken over et større område). Til vårt formål kan fototransistoren karakteriseres ved at vi måler U_{out} som funksjon av lysstyrken

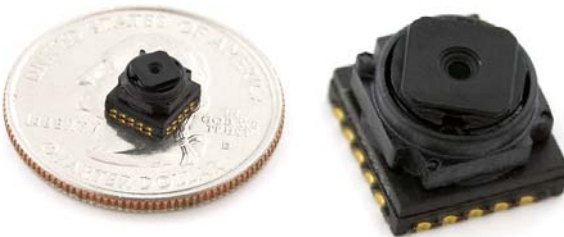
Diagrammene på figuren til venstre

viser reversstrømmen i fotoledende modus som funksjon av lyseffekten i mW/cm^2 for fototransistorene SFH300²⁵ og SFH203.

7.4.3 Kameramoduler

Digitale kameraer er en stor mengde lyssensorer som er ordnet i en matrise. Matrisene kan ha forskjellig størrelse. Moderne digitalkameraer har vanligvis over 10 mill lyssensorer. Vi skal her peke på noen få varianter som kan være kandidater som bør vurderes for bruk i forbindelse med f.eks. CanSat.

TCM8230MD, TOSHIBA (Sparkfun - 10\$)



Som det framgår av figuren til venstre er dette et miniatyrkamera (6 x 6 x 4,5 mm) til en pris av 10 \$²⁶ fra firmaet Sparkfun. Kameraet leverer bilder i VGA format (640 x 480 piksler) og har en øvre bildehastighet på 30 fps²⁷ med et optisk format på 1/6". En prosessor chip er inkludert i kameramodulen og leverer bildedata på standard seriell

form eller I²C buss. Kameramodulen har innebygget automatisk forsterkningskontroll og hvitbalanse.

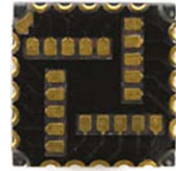
25. De ulike kurvene (2–4) angir ulike varianter av fototransistoren SFH300.

26. <http://www.sparkfun.com/products/8667>

27. fps - frame pr. sec.



Kameraet krever to spenningskilder, en på $2,8 \pm 0,2$ V (sensor (fotodiode) I/O) og $1,5 \pm 0,1$ V (sensor (ADC), digitalt). Med andre ord det kan være litt plundrete og framskaffe de rette forskyningsspenningene. En annen utfordring er tilkoblingen som krever en svært nennsom loddebolt eller spesialutstyr for montasje av overflate komponenter. En må regne med å lage et eget kretskortlegg for denne modulen dersom en ønsker å bruke kameraet



Skal kameraet overføre bilder fra en CanSat må en enten overføre enkeltbilder, eller ha en egen radiokanal for bildeoverføring.

Ønsker man bedre oppløsning kan man velge TCM8240MD (1300 x 1040 piksler). Prisen er den samme (10 \$), men vil selvfølgelig kreve mer av overføringskanalen. Dette kameraet synes imidlertid å ha fått noe blandet mottagelse.²⁸ For mer informasjon se også databladet²⁹.

7.5 Luftrykkfølsomme sensorer

Måling av luftrykk er spesielt viktig i forbindelse med værvarsling. I denne sammenheng brukes værballonger for å måle luftrykket som funksjon av høyden over bakken. Å overvåke trykket i en flykabin er også viktig. Dersom trykket faller under et visst nivå, utløses oksygenmaskene slik at passasjerene ikke skal komme til skade.

En annen viktig anvendelse har vært som høydemåler for ballonger, fly og lavbane-raketter. Dette er mulig siden trykkfallet opp gjennom atmosfæren er relativt forutsigbart. Med en trykkmåler som er godt kalibrert, kan en komme ned i en nøyaktighet på noen få meter. Vi skal senere se på hvordan vi regner om fra trykk til høydemeter over havet.

I forbindelse med CanSat er vi spesielt interessert i å bruke trykkmålinger for å anslå høyden.

28. <http://www.sparkfun.com/products/8668>

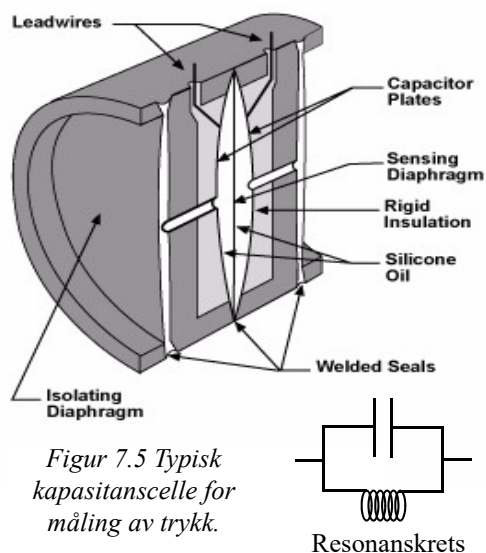
29. <http://www.sparkfun.com/datasheets/Sensors/Imaging/TCM8230MD.pdf>



7.5.1 Måling av lufttrykk

Måling av lufttrykk ved endring i kapasitans

Figuren til høyre viser en av flere måter å lage en trykksensor på. En membran er spent opp i et kammer. Kammerets vegger er plater i en kondensator. Membranen består av et materiale som gjør at kapasitansen mellom platene i kammeret endrer seg når membranen beveger seg som følge av trykkendringer. Rommet foran membranen er fylt med silikonolje og står i forbindelse med et isolerende membran foran på trykksensoren. Kondensatorplatene i veggen i det indre rommet inngår i en resonanskrets. Når membranen endrer form som funksjon av trykkforandringer, endres også resonansfrekvensen til resonatoren. Dersom en kjenner sammenhengen mellom trykk og frekvens, kan en bestemme trykket på bakgrunn av frekvensmålinger.



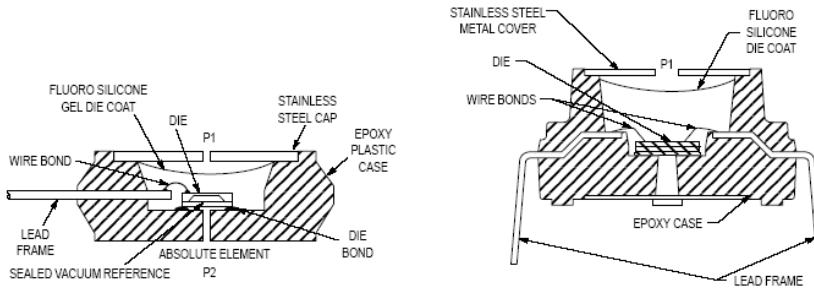
Måling av lufttrykk ved endring i resistans (piezo-resistivitet)

Den piezo-resistive effekten er forskjellig fra den piezo-elektriske effekten. Den piezo-resistive effekten ble oppdaget av **Lord Kelvin** i 1856. Først i 1954 oppdaget C.G. Smith at germanium- og silisiumkrystaller hadde spesielt store variasjoner i ledningsevnen når de ble utsatt for mekanisk stress. Ledningsevnen til materialer er avhengig av mengden ladningsbærere i ledningsbåndet og hvor lett elektroner kan frigjøres fra valensbåndet. Dette er igjen avhengig av størrelsen på *båndgapet* mellom lednings- og valensbåndet i materialet. Når de nevnte materialene utsettes for stress, vil båndgapet endre seg og dermed også ledningsevnen.

7.5.2 Trykksensorene MPX4100A og MPX4115 (Motorola)

Motorolas *MPX4100A* og *MPX4115* er sensorer som bruker piezo-resistive elementer som trykkfølere. Dvs. at lufttrykket presser på framsiden av en piezo-resistiv skive. Baksiden av skiven har et veldefinert trykk (f.eks. vakuum). Skiva er utformet slik at de resistive elementene i skiva kan

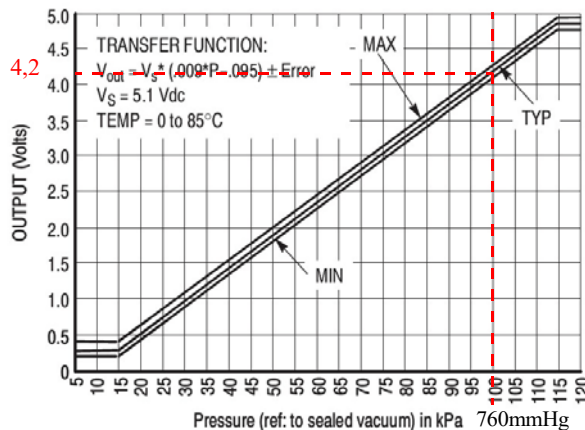
inngå i en målebro slik at endring i resistivitet kan omgjøres til en spenning. Denne forsterkes til ønsket nivå. De nevnte sensorene er utstyrt med en integrert forsterker slik at sensorene leverer en passende spenning.



Figur 7.6 Utforming av trykksensor med piezo-elektriske elementer.

Sammenhengen mellom trykk og spenning på utgangen av sensoren MPX4100A er vist i figur 7.7

Figuren viser også komponentens målesikkerhet. Dette er produksjonsusikkerhet mellom de ulike komponentene, samt temperaturspredning. Vi legger også merke til at normalt lufttrykk ligger helt i toppen på skalaen, hvilket betyr at denne sensoren egner seg best til måling av normalt og lavere lufttrykk, hvilket er det vi har ved en rakettoppskytning eller et ballongslipp. Vi ser at spenningsnivået ut av sensoren passer godt til et slikt formål.



Figur 7.7 Spenning på utgangen som funksjon av trykk i kPa.

Siden vi måler spenning (V_P i Volt), men ønsker lufttrykk (P i kPa), må vi regne om fra spenning til lufttrykk. Fabrikanten har oppgitt følgende sammenheng [7]:

$$V_P = 5,0 (0,009 P - 0,095) \quad (7.13)$$

Dersom vi setter inn 100 kPa får vi en spenning 4,025 V hvilket ikke er så fjernt fra det vi kan lese av på diagrammet over ($\approx 4,2$ V).

Snur vi om på formelen vil vi kunne skrive:

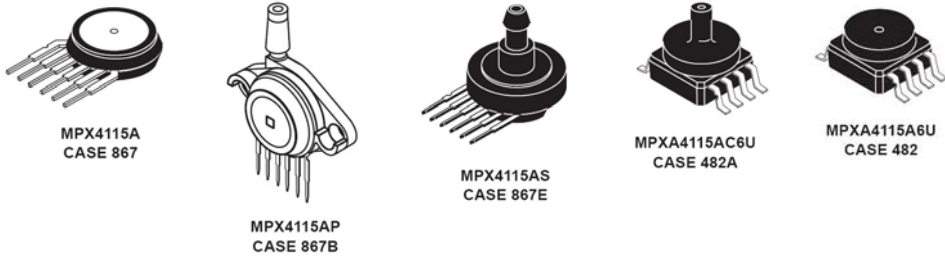
$$P = 22,222 V_P + 10,556 \quad (7.14)$$



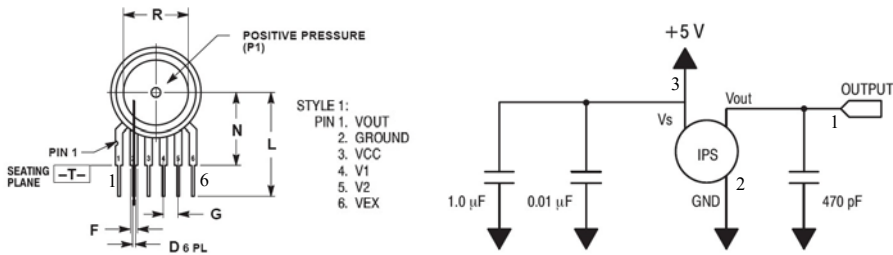
Derne ønsker man ofte å regne om fra trykk til høyde hvilket er noe mer utfordrende, se ligning 7.15 på side 150.

Mekanisk utforming av MPX415XX

MPX4115 leveres i ulike pakninger, se figuren under. CanSat leveres med MPX4115A mens MPX4115AP eller AS kan egne seg bedre til å karakterisere sensoren i og med at en kan koble på en slange slik at det er lettere å utføre målinger under kontrollert trykk.



Sensoren kobles til spenningskilden (+5 V) og vil da gi en signalspenning på utgangen proporsjonal med trykket i kPa, som vist på figur 7.7. Figuren under viser til venstre sensorens pinning og til høyre anbefalt avkobling. Avkobling med kondensatorer brukes for å unngå at støy på forskyningsspenningen skal redusere kvaliteten på målingene.



Informasjonen er hentet fra databladet [7].

7.5.3 Måling av høyde basert på trykkmålinger

Trykk måles normalt i Pascal hvor $1 \text{ Pa} = 1 \text{ N/m}^2$.

Tidligere ble trykk målt i atmosfærer (atm), mmHg eller Bar.

En normalverdi for lufttrykket er:

$$1 \text{ atm} = 760 \text{ mmHg} = 1.01325 \text{ Bar} = 1013.25 \text{ mBar} = 101325 \text{ Pa} = 1013.25 \text{ hPa}$$

Vi legger merke til at h(ekto)Pa er det samme som m(illi)bar.

Lufttrykket er bestemt av tyngden til det “havet” av luft som vi befinner oss på bunnen av. Lufttrykket er derfor avhengig av den mengden av luft som til en hver tid befinner seg over hodet på oss. Vekta av luftmengden er avhengig av tyngdekraften, tykkelsen på luftlaget og tettheten, som



igjen er avhengig av hvordan lufta forflytter seg og av temperaturen, dvs. værforholdene. Som vi ser er det mange faktorer å ta hensyn til. Likevel finnes det gode håndregler som gjør at en kan gjøre tilstrekkelig nøyaktige høydemålinger på bakgrunn av trykkmålinger.

En regner normalt at trykket faller med 1 millibar pr. 8 meter, eller ca 12.5 millibar pr. 100 meter. Dette stemmer ikke så verst for de første 2000 meter, deretter minker trykket mindre for hver 1000 meter.

Normalt refereres alle trykkmålinger til havnivået. En meteorologisk stasjon som oppgir barometerstand ved stasjonen, har vanligvis regnet om verdiene til havnivået.

Tabellen under viser typiske verdier for sammenhengen mellom trykk, lufttetthet, temperatur og høyde over havet.

HoH	Temperatur	Lufttrykk	Tetthet	
(m)	(C)	(hPa)	(kg/m ³)	
0000	15.0	1013	1.2	
1000	8.5	900	1.1	
2000	2.0	800	1.0	(Galdhøpiggen)
3000	-4.5	700	0.91	
4000	-11.0	620	0.82	
5000	-17.5	540	0.74	
6000	-24.0	470	0.66	
7000	-30.5	410	0.59	
8000	-37.0	360	0.53	
9000	-43.5	310	0.47	(Mount Everest)
10000	-50.0	260	0.41	(Marsjhøyde rutefly)
11000	-56.5	230	0.36	
12000	-56.5	190	0.31	
13000	-56.5	170	0.27	
14000	-56.5	140	0.23	
15000	-56.5	120	0.19	
16000	-56.5	100	0.17	
17000	-56.5	90	0.14	
18000	-56.5	75	0.12	
19000	-56.5	65	0.10	
20000	-56.5	55	0.088	
21000	-55.5	47	0.075	
22000	-54.5	40	0.064	
23000	-53.5	34	0.054	
24000	-52.5	29	0.046	
25000	-51.5	25	0.039	
26000	-50.5	22	0.034	
27000	-49.5	18	0.029	
28000	-48.5	16	0.025	
29000	-47.5	14	0.021	
30000	-46.5	12	0.018	
31000	-45.5	10	0.015	
32000	-44.5	8.7	0.013	
33000	-41.7	7.5	0.011	
34000	-38.9	6.5	0.0096	
35000	-36.1	5.6	0.0082	



Omregningen fra trykk til høyde må også ta hensyn til temperaturen. Temperaturen vil dessuten forandre seg med høyden.

Det er normalt lettere å forholde seg til en omregningsformel enn en tabell. Ulempen med en formel er at de mange parametrene kan gi stor usikkerhet i beregningen. I *The CanSat book* er følgende sammenheng utledet/referert:

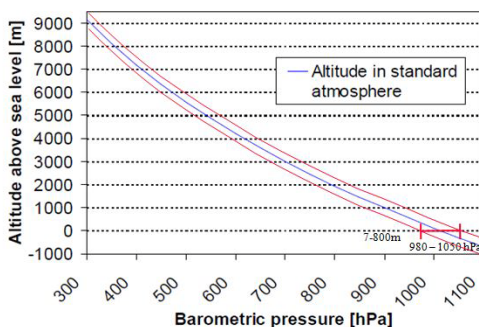
$$h = \frac{T_1}{a} \left(\left(\frac{p}{p_1} \right)^{\frac{-aR}{g_0}} - 1 \right) + h_1 \quad (7.15)$$

Hvor:

h	Beregnet høyde i meter
h_1	Starthøyde i meter
T	Temperatur i Kelvin
T_1	Starttemperatur i høyden h_1
a	Temperaturgradient, foreslått verdi $-0,0065$ K/m
p	Trykk i Pa
p_1	Trykk i Pa ved starthøyden
g_0	Tyngdeakselerasjonen $9,81$ m/s ²
R	Den spesifikke gasskonstant $287,06$ J/kg K

Denne formelen kan enten legges inn i datainnsamlingsenheten i CanSat, men bedre i programvaren som behandler data. Har man rådataene fra sonden, har en større mulighet til etterbehandling enn om man bare har de omregnede dataene.

Diagrammet til høyre viser sammenhengen mellom trykk og høyde med økende høyde over havnivået. Normale variasjoner i lufttrykket ved bakken kan være fra 980 hPa til 1050 hPa (millibar). Denne naturlige variasjonen kan derfor gi en absolutt endring i høydeberegningen på 7 – 800 meter dersom man ikke kalibrerer på bakkenivå.



7.5.4 Barometeret BMP085 (Bosch)

BMP085 er en trykksensor som går under betegnelsen *barometer*. Selve sensoren er basert på piezo-resistivitet teknologi og levert av firmaet Robert Bosch. Kretsen leverer trykkdata via en digital I²C buss. I tillegg til trykksensoren inneholder kretsen en temperatursensor. her er noen nøkkeldata:

- Måleområde: 300 – 1100 hPa (millibar) – som typisk tilsvarer 9 000 – 500 meter

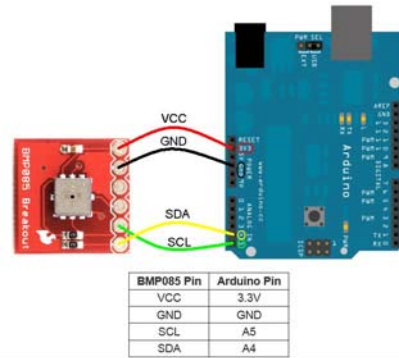


- Supplyspenning: 1,8 – 3,6 V
- Lavt effektforbruk: 5 μ A ved 1 måling pr. sek.
- Responstid: 7,5 ms (maks)
- *Standby* strøm: 0,1 μ A
- Langtidsstabilitet: \pm 1 hPa pr. 12 måneder.

Kretsen leveres fra Sparkfun monteret på et kretskort for lettere å kunne kobles til f.eks. en Arduino (“*breakout board*”).

For å lese dataene via I²C bussen brukes et spesielt bibliotek: `#include <wire.h>`

For mer informasjon se: <https://www.sparkfun.com/tutorials/253> her finner du også databladet og programvare for Arduino.



7.6 Fuktighetsfølsomme sensorer

7.6.1 Måling av fuktighet

I forbindelse med ballongslipp vil det være interessant å måle den relative fuktigheten. Dette sier noe om vanninnholdet i atmosfæren, noe bl.a. meteorologene er interessert i. I dette avsnittet skal vi se på noen teknikker for å måle relativ fuktighet.

Hva er relativ fuktighet?

Luft har evnen til å oppta vanddamp. Men mengden som kan tas opp er ikke ubegrenset. Når lufta ikke klarer å ta opp mer, sier vi at den er *mettet*. Luft som er mettet med vanddamp defineres å ha en relativ fuktighet på 100%. Dersom lufta er helt fri for vanddamp er den tørr og vil ha en relativ fuktighet på 0%.

100% fuktighet tilsvarer et visst antall gram vann pr. m³. Hvor mange gram dette tilsvarer er avhengig av både lufttrykket og ikke minst av temperaturen. Jo høyere temperatur jo mer vanddamp kan lufta oppta før den er mettet. Når skyer driver inn mot land, blir de gjerne presset opp i kaldere luftlag. Når mettet luft blir kaldere vil den gi fra seg noe av fuktigheten som regn.

Det er derfor vanlig at vi måler trykk og temperatur sammen med fuktighet. En slik kombinert måler kalles ofte en PTU-sonde (**P**ressure - **T**emperatur - **hU**midity).

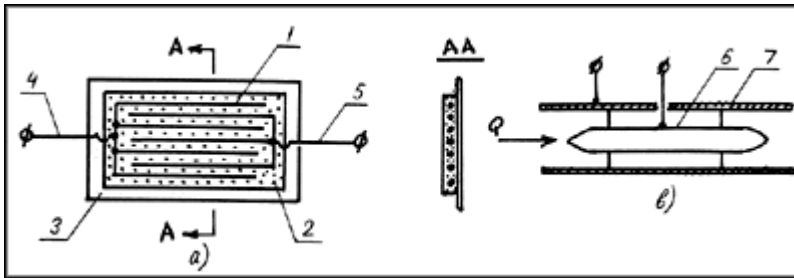


Temperaturen til det tørre termometeret er avmerket langs den horisontale aksens, mens temperaturen til det våte termometeret er avmerket langs den buede delen av diagrammet. De buede linjene i diagrammet angir den relative luftfuktigheten.

Relativ luftfuktighet bestemmes ved å lese av verdien på den buede linjen der linjene for temperaturen målt på tørt og vått termometer møtes. Vi legger merke til at ved 100% relativ fuktighet, er temperaturen til de to termometrene like.

Resistivt eller kapasitivt hygrometer [11]a

Resistive hygrometer baserer seg på at et materiale endrer resistivitet (elektrisk motstand) når det trekker til seg fuktighet. Til høyre på figuren under er vist metallfingre som går om hverandre i et materiale med en viss elektrisk motstand (resistivitet). Dette materialet har evnen til å oppta og avgi fuktighet i takt med fuktigheten i lufta omkring. Den målte resistiviteten er derfor et mål for fuktigheten i lufta. Materialer som brukes er bl.a. Litiumklorid og Aluminiumoksid.



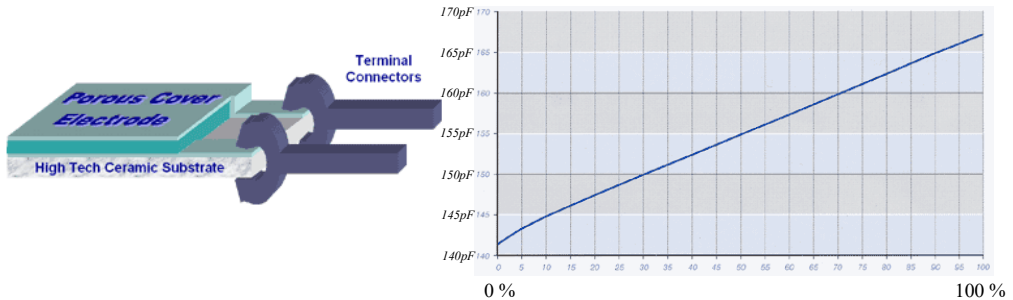
Figur 7.9 a) Resistivt hygrometer og b) kapasitivt hygrometer.

1. elektrode, 2. hygroskopisk resistivt materiale, 3. substrat (holder), 4. og 5. tilførselsledninger, Q - luftstrøm m/fuktighet, 6. og 7. indre og ytre elektrode.

Det tradisjonelle *kapasitive hygrometeret* består av to elektroder som er plassert henholdsvis inne i og langs sidene i et rør som fører den fuktige luftstrømmen. De dielektriske egenskapene til lufta vil endre seg avhengig av mengden fuktighet i luftstrømmen. Denne endringen forandrer kapasiteten mellom de to elektrodene. Endringen i kapasitans kan måles, og gir et mål for den relative fuktigheten i luftstrømmen.



Moderne kapasitive fuktighetssensorer er bygget opp som en platekondensator. Den ene elektroden er porøs slik at den kan slippe fuktighet inn til det dielektriske materialet mellom de to elektrodene. Fuktigheten endrer dermed kapasitansen som kan avleses av en tilhørende elektronikk.



Figur 7.10 Moderne kapasitiv fuktighetssensor³⁰.

7.6.2 Fuktighetssensoren H25K5A

Dette er en relativt rimelig sensor for måling av luftfuktighet. Sensoren er resistiv, med andre ord den kan oppfattes som en motstand som endrer resistans som funksjon av luftens relative fuktighet (RH). I tillegg vil den også endre resistans for ulike temperaturer.

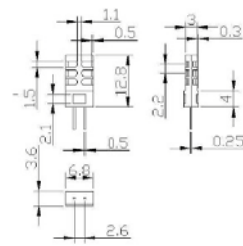
Sensor resistance table (unit:K Ω) At 1 KHz, Voltage 1Vrms.

	0°C	5°C	10°C	15°C	20°C	25°C	30°C	35°C	40°C	45°C	50°C
20%RH				21000	13500	9800	8000	6300	4600	3800	3200
25%RH		19800	16000	10500	6700	4803	3900	3100	2300	1850	1550
30%RH	12000	9800	7200	5100	3300	2500	2000	1500	1100	900	750
35%RH	5200	4700	3200	2350	1800	1300	980	750	575	430	350
40%RH	2800	2000	1400	1050	840	630	470	385	282	210	170
45%RH	720	510	386	287	216	166	131	104	80	66	51
50%RH	384	271	211	159	123	95	77	63	52	45	38
55%RH	200	149	118	91	70	55	44	38	32	30	24
60%RH	108	82	64	51	40	31	25	21	17	14	12
65%RH	64	48	38	31	25	20	17	13	11	9	8
70%RH	38	29	24	19	16	13	10.5	9	8.2	7.1	6.0
75%RH	23	18	15	12	10	8.5	7.2	6.4	5.8	5.0	4.1
80%RH	16	12	10.2	8.1	7.2	5.7	5.0	4.4	4.0	3.3	2.9
85%RH	10.2	8.2	6.9	5.5	4.7	4.0	3.6	3.2	2.9	2.4	2.0
90%RH	6.9	5.4	4.7	4.1	3.2	2.8	2.5	2.3	2.1	1.8	1.5

30. Informasjonen er hentet fra <http://www.ist-ag.com/english/products/humid/p14TR.html>

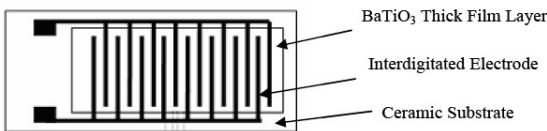


Av figuren over ser vi at resistansen ved 25°C faller fra ca. 9 MΩ til 3 kΩ når den relative fuktigheten øker fra 20% til 90%. Dvs. en svært markert endring. Nøyaktigheten er oppgitt til +/- 5% RH ved 60% RH. Dvs. den er relativ unøyaktig.



Databladet anbefaler at sensoren tilføres en vekselspanning på 1 kHz i stedet for en likespenning, for å unngå polarisering av sensoren, noe som kompliserer bruken. Målinger synes å bekrefte dette³¹.

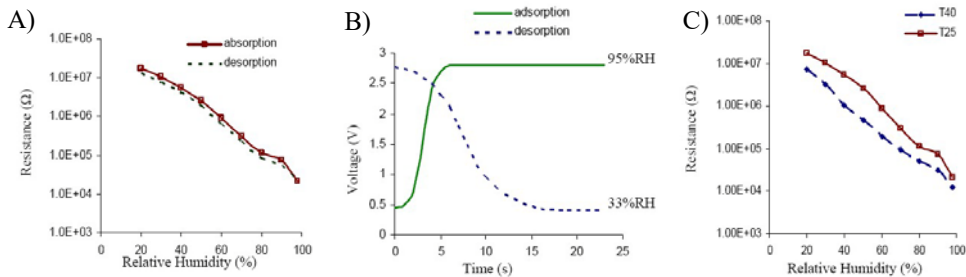
Komponenten er dessuten liten og ikke særlig dyr. Hos ELFA koster denne ca. 30,- i stykkpris. Figuren til høyre viser dimensjonene i millimeter. Videre tåler den spenninger opp til 10 V³².



Selv om det er vanskelig å finne detaljerte beskrivelser av hvordan sensoren er bygget opp, er det rimelig å anta at den består av et isolerende keramisk materiale med fingerelektroder som er dekket med et

kjemisk stoff som endrer resistivitet som funksjon av fuktigheten i omgivelsene. Dette materialet kan for eksempel være BaTiO₃. Figuren over viser hvordan sensoren kan være konstruert³³. Sensoren er montert inn i et perforert plastdeksel.

Kurvene i figuren under viser resultatet av målinger utført på en slik sensor.



Figur 7.11 Målinger på BaTiO₃ materiale. A) Resistans som funksjon av relativ fuktighet ved 25 °C, B) Tidsrespons ved endring av fuktighet C) Resistans som funksjon av relativ fuktighet ved 25 og 40 °C.

Figur 7.11 A) viser resistansen til sensoren som funksjon av relativ fuktighet for 25 °C. Den heltrukne linjen er målinger gjort ved stigende fuktighet, stiplet linje er verdier målt ved fallende fuktighet. Vi ser at de to måleseriene ikke er helt sammenfallende, dvs. at sensoren har *hysteres*³⁴. Figur 7.11 B) viser hvordan sensoren gir respons ved meget brå endringer i fuktighet. Heltrukken linje viser responsen ved en rask økning i fuktighet. Vi ser at sensoren trenger ca. 2–3 sekunder for å gå fra ca. 30% til 95% relativ fuktighet. Vi legger også merke til at responsen er vesentlig langsommere ved fallende fuktighet (12–13 sek.) enn ved stigende. Figur 7.11 C) viser

31. Målinger utført av Knut Nygård ved Frydenlund skole synes å bekrefte dette

32. Se forøvrig databladet: <http://www.sensorelement.com/humidity/H25K5A%20spec.pdf>

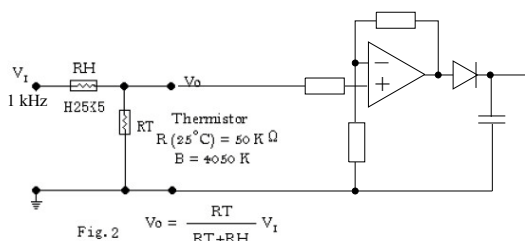
33. Figuren er hentet fra [11]c.

34. Hysteres betyr at kurvene ikke følger samme forløp ved stigende som fallende temperatur



relativt stor variasjon i resistans ved varierende temperatur. Vi ser at resistansen faller med økende temperatur. Dette er ikke urimelig da vi vet at relativ fuktighet nettopp varierer med temperaturen. Når temperaturen øker kan lufta oppta mer damp. Ved f.eks. 40% relativ fuktighet så vil lufta inneholde mer vanddamp ved en høy temperatur enn ved en lav temperatur. Det ideelle hadde vært at sensoren hadde tatt hensyn til dette og kompensert for variasjonen over temperaturområdet. Dvs. vi kunne avlese riktig verdi for relativ fuktighet uavhengig av temperaturen.

Dette er mulig ved å benytte en NTC-motstand (termistor) som endrer verdi med temperaturen på en slik måte at den kompensere for avvikene. Databladet for H25K5A anbefaler en termistor (NTC-motstand) på 50 kΩ ved 25 °C.



Ideelt sett skulle spenningen på utgangen av spenningsdeleren til venstre være konstant som funksjon av temperaturen dersom relativ fuktighet holdes konstant. Det vil i praksis si at lufta omkring sensoren måtte tilføres mer vanddamp når temperaturen økte slik at den relative fuktigheten ble uforandret. Dette er vanskelig å få til uten avansert måleutrustning.

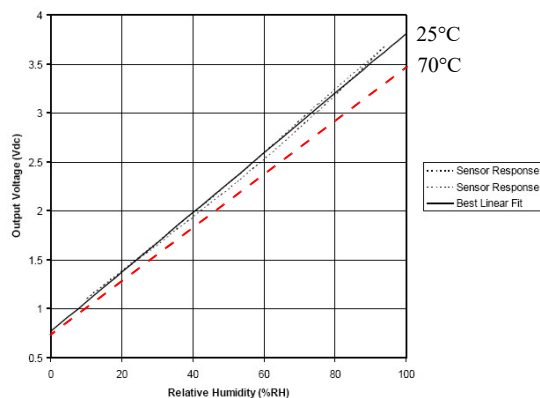
Dersom vi tilfører kretsen en vekselspanning på 1 kHz må spenningen på utgangen likerettes og filtreres før den føres til AD-konverteren som vist på figuren over.

7.6.3 Fuktighetssensoren HiH4000 Honeywell

HiH4000 er en kapasitiv luftfuktighetssensor med innebygget elektronikk for omforming av kapasitet til elektrisk spenning. Kretsen er vesentlig dyrere enn H25K5A (stk. pris kr. 218,- hos ELFA).



Den har en nær lineær sammenheng mellom relativ fuktighet og spenning som vist på figuren til høyre (ved 25°C). Variasjonen er imidlertid betydelig dersom temperaturen avviker vesentlig fra 25°C. Vi ser videre at spenningen på utgangen er nær ideell for bruk i CanSat idet verdiene spenner fra ca. 0,75 V ved 0% til 4,75 V ved 100% relativ fuktighet. Den trekker også særdeles lite strøm hvilket selvfølgelig er gunstig ved batteridrift.



En ulempe med de fleste fuktighetssensorene er at de reagerer relativt langsomt på endringer i luftfuktighet. Uten luftsirkulasjon antydes en omslagstid på typisk 15 sek. Dersom denne skal brukes i forbindelse med CanSat bør den plasseres slik at den utsettes for luftstrømmen som passerer forbi sensoren.

Med utgangspunkt i grafen på figuren over til høyre kan en estimere en sammenheng mellom målt spenning og relativ fuktighet. En kan også utnytte kjennskapet til temperaturvariasjonen.

Følgende oppgis i databladet:

$$V_{out} = V_s(0,0062 \cdot \text{Sensor RH}) + 0,16), \text{ typisk ved } 25 \text{ }^\circ\text{C} \quad (7.16)$$

$$\text{Sensor RH} = (V_{out} - 0,16 \cdot V_s)/(V_s \cdot 0,0062) \quad (7.17)$$

Hvor:

V_{out} er målt spenning på utgangen av sensoren

V_s er forskyningsspenningen (batterispenningen)

Sensor RH er ukorrigert fuktighetsmåling ved sensoren

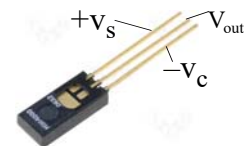
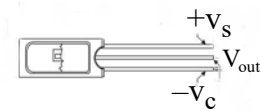
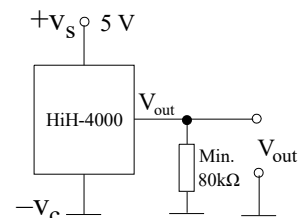
Siden vi ønsker å måle relativ fuktighet så må vi også ta hensyn til temperaturen

$$\text{True RH} = (\text{Sensor RH})/(1,0546 - 0,00216 \cdot T), T \text{ i } ^\circ\text{C} \quad (7.18)$$

True RH er relativ fuktighet i %

T er temperatur i $^\circ\text{C}$

Figuren til venstre viser en typisk oppkobling og nederst sensoren pinning sett forfra (hvor fuktigheten trenger inn i sensorhuset).



7.7 Bevegelsesensorer

I denne gruppen har vi valgt å inkludere *akselerasjonssensorer* og *bevegelsessensorer*. Mens akselerometeret befinner seg på gjenstanden som er i bevegelse, vil bevegelsessensoren normalt være fast montert og detektere eller måle bevegelsen til en person eller gjenstand som kommer innen sensorens aksjonsområde.

7.7.1 Akselerometer

Siden CanSat sonden faller med en relativt konstant fart gjennom atmosfæren skulle en tro at måling av akselerasjon var relativt uinteressant. En kan imidlertid registrere, spinn, g-krefter ved oppskyting og ved returnering.

Følgende akselerometer er montert på datainnsamlingsenheten i CanSat'en.



Akselerometeret MMA7361L

Her er noen nyttige data:

- Velg mellom to måleområde fra $\pm 1,5g$ eller $\pm 6g$.
Velges ved å legge *g-select* inngangen til 0 ($\pm 1,5g$) eller 1 ($\pm 6g$).
- Supplyspenning $V_{CC} = 2,2 - 3,4 V$
- Følsomhet 800 mV/g ved $\pm 1,5g$ ($S_{g1,5}$)
- Følsomhet 206 mV/g ved $\pm 6g$ ($S_{g6,0}$)
- 0 g har en typisk utspenning på $1,65 V \pm 0,165$ (ved 3,3 V supply)

Omregning fra målt spenning (V_{out}) til tyngdeakselerasjon (a_g) blir dermed:

$$a_g = V_{out} \cdot S_g - 1,65V \quad (7.19)$$

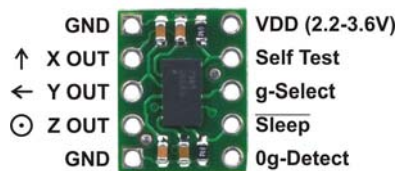
Hvor følsomheten er S_g (enten er 800 mV/g ved $\pm 1,5g$ ($S_{g1,5}$) eller 206 mV/g ved $\pm 6g$ ($S_{g6,0}$))

Vanlige anvendelser:

- Tilt og bevegelsessensor i 3D spill
- Deteksjon av fritt fall hos HDD MP3 spillere, Laptop'er og PCer
- Bildestabilisering, elektronisk kompass, oppringing ved bevegelse hos mobiltelefoner
- Bevegelsessensor i pedometre
- Bevegelsessensor hos roboter

Bildet høyre viser layout og pinning.

- XOUT/YOUT/ZOUT signalutgangen fra de tre retningene



- *Self Test* - Tester riktig funksjon etter installasjon
- *g-select* - Legges den til 0 er måleområdet $\pm 1,5g$, legges den til 1 er området $\pm 6g$
- *Sleep* - Lav på denne inngangen setter akselerometeret i *sleep-mode* for å spare strøm.
- *0g-Detect* - Detekterer fritt fall (utgang)
- GND kobles til jord
- VDD supply spenning (2,2 til 3,6 V)

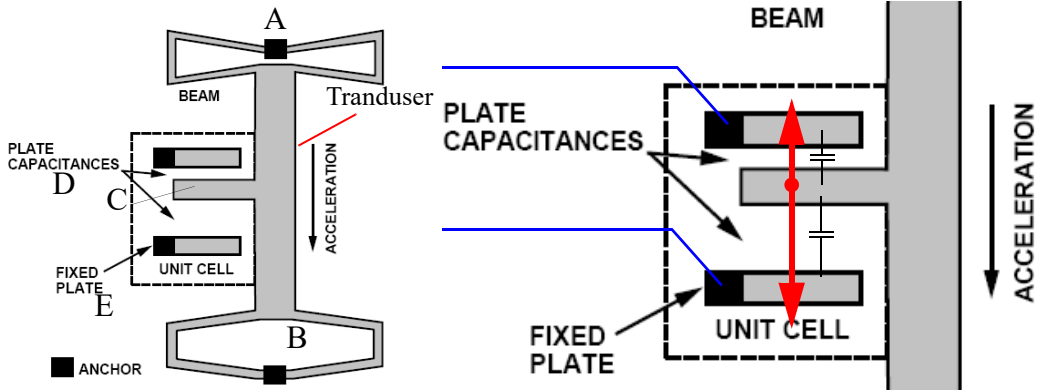
Databladet anbefaler 3,3 nF mellom signalutgangene XOUT/YOUT/ZOUT og jord for å redusere innslag av klokkestøy.

Prinsippet bak målingen av akselerasjonen baserer seg på massens treghet. I akselererte systemer vil massen påvirkes av krefter som kan måles.

Ved hjelp av en etseteknikk lages det *transducere* som henger fritt mellom to punkter A og B (se figuren under). Transduseren er laget ved å legge et lag med polysilisiumstruktur utformet med ønsket form (se figuren under) på toppen av et lag med silisiumoksid. Oksidlaget på undersiden



ettes bort slik at transduseren blir hengende fritt. En flik C av transduseren kan bevege seg mellom to faste kondensatorplater (D og E). Når akselerometeret utsettes for en akselerasjon i lengderetningen til transduseren, vil den påvirkes av krefter siden den har masse (treghet). Fliken C vil dermed forskyves i forhold til de to faste metallplatene D. Flikene C og D vil ikke berøre hverandre men danne kondensatorer som vist på figuren til høyre.

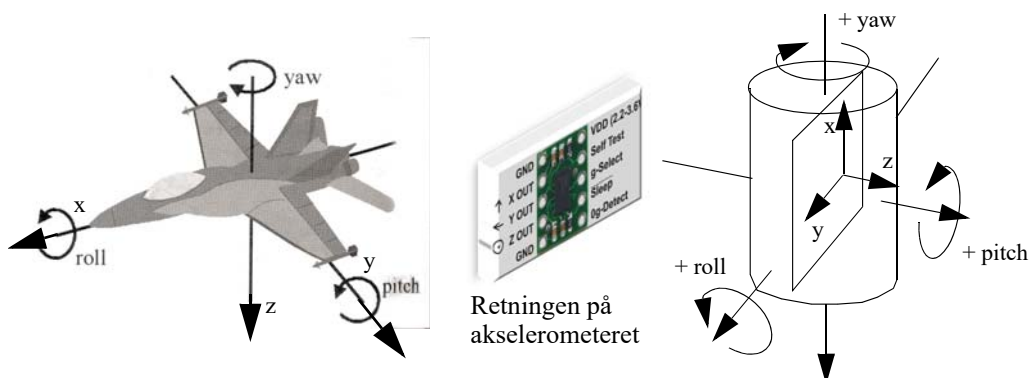


Figur 7.12 Prinsippskisse av akselerometerets kapasitive sensor.

Når fliken beveger seg endres de kapasitive egenskapene. Denne endringen genererer et signal som kan relateres til akselerasjonen.

Kretsen måler statisk akselerasjon og vil dermed også måle tyngdeakselerasjonen g , og vil kunne fungere som tilt-sensor (måler helningsvinkelen). Dvs. den kan registrere hvordan de ulike sensorene heller i forhold til gravitasjonens retning. Det kan i enkelte tilfeller være nyttig å bruke informasjon fra en slik sensor for å studere hvordan en CanSat er orientert når den faller gjennom atmosfæren.

Vi definere “yaw” som positiv dreining om z-aksen, “pitch” som positiv dreining om y-aksen og “roll” som positiv dreining om x-aksen. I vår CanSat er akselerometerkortet montert vertikalt slik at x-aksen peker oppover og y- og z-aksen peker ut til sidene når CanSat’en står vertikalt som vist på figuren under til høyre. Øyeskruen og fallskjermen er da på oversiden sammen





“Roll” og “pitch” kan da beregnes på følgende måte:

$$Pitch = \text{asin}(a_y/1g) \quad (7.20)$$

$$Roll = \text{asin}(a_z/1g) \quad (7.21)$$

Hvor a_y og a_z er akselerasjonen i henholdsvis y- og z-retning slik akselerometeret er plassert i CanSat'en. g er tyngdeakselerasjonen.

“Pitch og “roll” beskriver dermed helningen i de to retningene i forhold til retningen til tyngdeakselerasjonen og akselerometerets plassering.

Problemstilling:

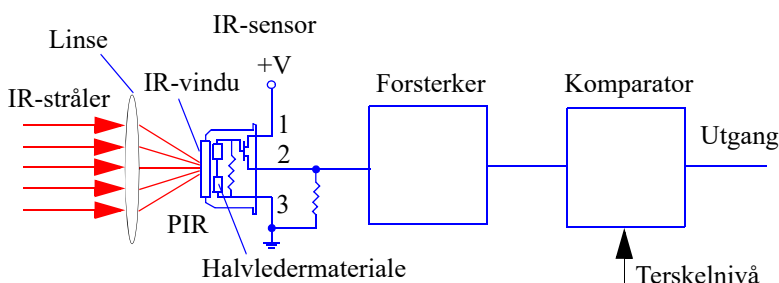
Er det mulig å måle rotasjonshastigheten med denne kretsen? Ev. finn ut hvordan kretsen må plasseres dersom en ønsker å måle rotasjonshastigheten.

7.7.2 Bevegelsessensor

Mange forskjellige metoder brukes for å detektere menneskelig tilstedeværelse i et rom. I bolig- hus finner vi vanligvis “Passiv InfraRed detector’er” (PIR) eller “Pyroelectric sensor”. Alt levende sender ut varmestråling (infrarød eller IR-stråling). Det samme gjelder egentlig alle gjenstander. Forskjellen på gjenstander og mennesker (ev. dyr) er at gjenstandene stort sett holder seg i ro. IR-detektorer registrerer derfor gjenstander (eller mennesker) som har en viss varmeutstråling og som er i bevegelse. En varm ovn vil derfor ikke detekteres, siden den står i ro.

En reflektor er montert bak et deksel av plast. Reflektoren samler IR-strålene inn mot selve detektoren som er montert i en liten metallkapsel med et lite vindu laget av silisium. Silisium har den egenskapen at det er gjennomsiktig for IR-stråler, slik glass er gjennomsiktig for lys.

Det er viktig at sensoren er tilpasset strålingsspekteret fra levende vesener. Mens bølgelengdene for lys ligger i området 400–750 nm (nanometer, 10^{-9} m), så finner vi IR-strålingen fra et menneske i området 9 000–10 000 nm. Følsomhetsområdet for slike sensorer er derfor i området 8 000 - 12 000 nm (eller 8–12 μ m).



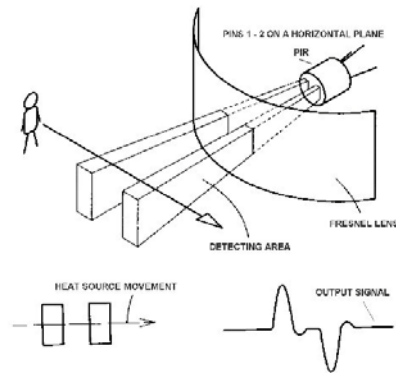
Figur 7.13 Prinsippkisse for en IR-detektor.

Selve sensoren ligner på en lyssensor. IR-stråler er elektromagnetisk stråling, eller *fotoner*, akkurat som lys. Når fotonene treffer halvledermaterialet slår de løs elektroner. De frie elektronene vil så øke den elektriske ledningsevnen i materialet. Sender vi strøm gjennom halvledermaterialet,

vil strømmen øke med økende ledningsevne. Økningen i strømstyrke registreres av tilhørende elektronikk. Siden en kun er interessert i de *raske* endringene i mottatt IR-stråling, fjernes de langsomme endringene av elektriske signalfiltre.

Enkelte detektorer mottar stråling i to retninger som ligger tett ved hverandre som vist på figur 7.14. Ved generell oppvarming av rommet vil det registreres økt IR-stråling i begge de to retningene. Dersom kilden beveger seg forbi detektoren, vil IR-strålingen registreres først i den ene, og deretter i den andre retningen. I så fall vil detektoren registrere bevegelse og alarmen utløses.

En komparator sørger dessuten for at nivået på IR-strålingen må over et visst nivå for at det skal registreres en bevegelse. En komparator sammenligner nivået til signalet med et referansenivå. Dersom signalet overskrider referansenivået (terskelnivået) gis et signal på utgangen, viss ikke, skjer ingen ting. Ved å endre referansenivået kan en bestemme IR-detektorens følsomhet.



Figur 7.14 Tostråle IR-detektor.

7.8 Avstandssensorer

Dette er sensorer som bestemmer avstanden til en vegg eller en gjenstand og som gjerne baserer seg fra en refleks av et utsendt signal. Det kan være ultralyd, en IR-puls eller lignende.

La oss først studere en sensor som baserer seg på refleksjon av lyd.

7.8.1 Ultrasonisk avstandsmåler - SRF005

Som navnet sier så baseres denne sensoren seg på å sende ut en ultrasonisk lydimpuls, som så reflekteres fra et mål. Sensoren er så følsom at den kan detektere et mål med en diameter på 3 cm i en avstand på 3 meter. Følgende data er oppgitt for sensoren:

- Supply spenning – 5 V
- Strømtrekk – 30 mA Typ. 50 mA Max.
- Frekvens – 40 kHz
- Maks. rekkevidde – 3 m
- Min. rekkevidde – 3 cm
- Følsomhet – Detekterer et mål med 3cm diameter kosteskaft i en avstand på > 3 m
- Input trigger – 10 μ S min. TTL signal





- Ekkopulse – Positivt TTL signal, bredden er proporsjonal med avstanden.

Sensoren kan brukes på to ulike måter. Enten via *en pinne*, hvor trigger-signalet sendes ut på samme pinne som ekkot returnerer på, eller via *to pinner*, hvor trigger-signalet og ekkosignalet bruker hver sin pinne.

Oppkobling

Ved bruk av 5 polt kontakt (+ 5 V og 0 V er merket på kretskortet):

- + 5 V kobles til 5 V
- Er ikke i bruk
- Kobles direkte til pinnen på mikrokontrolleren
- Mode – kobles til 0 V
- 0 V kobles til 0 V

Ved bruk av 3 polt kontakt (SIG og 0 V er merket på kretskortet):

- Signal (SIG) kobles direkte til pinnen på mikrokontrolleren
- + 5 V Connect to 5 V
- 0 V Connect to 0 V

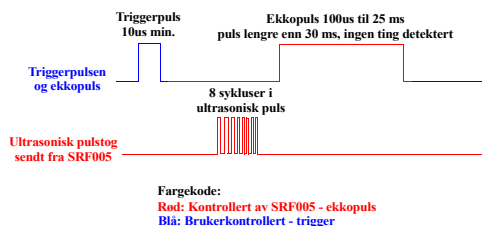
Når man bruker den 3-polte kontakten må man lodde en *strap* mellom mode og 0 V.

Timing

Ved bruk i en-pinne modus, så vil den ene pinnen først være en digital utgang (blå) for å sende trigger-pulsen, for deretter å bli en digital inngang (rød) som “lytter” etter ekkopulsen.

Lengden av ekkopulsen er proporsjonal med avstanden til det reflekterende objektet. Når lengden av ekkopulsen passerer 30 ms, så ansees det ikke å være noe i veien. ???

SRF005 timing diagram, en-pinne modus



7.9 Magnetfeltfølsomme sensorer³⁵

Magnetometer, instrument for måling av magnetisk feltstyrke og flukstetthet eller magnetiske egenskaper i et materiale. Finnes i en rekke utforminger og virkemåten kan bygge på svært forskjellige fysiske prinsipper. I tillegg finnes magnetsensorer for bruk i forbindelse med styring og overvåking.

35. Størstedelen av dette avsnittet er hentet fra oppgave besvarelsen til Nic Furu, Anita Bordevik og Eivind Tjensvoll under Romteknologikurset 2005. Gjengitt med tillatelse fra forfatterne.



Det magnetiske feltet betegnes ofte *B-feltet* med en magnetisk feltstyrke som måles i enheten **Gauss** (G) i cgs-systemet eller **Tesla** [N·s/C·m] i SI-systemet. Sammenhengen mellom Gauss og Tesla kan uttrykkes som: **1 Tesla = 10 000 Gauss**.³⁶

Magnetisk feltstyrke assosieres ofte med de to størrelsene *B-felt* og *H-felt*. Disse angir begge magnetisk feltstyrke, men brukes i forskjellige sammenhenger og har ofte ulike måleenheter. B-feltet assosieres gjerne med det feltet som dannes av ladninger i bevegelse og måles i Tesla. B-feltet kalles derfor også for *magnetisk induksjon* eller *magnetisk flukstetthet*. Når magnetiske felter (B-felt) passerer gjennom materialer av ulike slag, vil feltet i større eller mindre grad påvirkes av materialets magnetiske egenskaper. I denne sammenheng kan det være uklart hvor stor del av det kombinerte feltet som skyldes det ytre feltet og hva som skyldes materialets bidrag. H-feltet assosieres gjerne med det kombinerte feltet i materialet og måles også i Tesla. Sammenhengen mellom B- og H-feltet kan uttrykkes som:

$$B = \mu_m \cdot H \quad (7.22)$$

Hvor μ_m er materialets relative permeabilitet eller *materialets magnetiske gjennomtrengelighet*. For luft er $\mu_m = 1$.

For at vi skal få et visst begrep om hvor sterk én Gauss er, så kan følgende være til hjelp:

- $10^{-9} - 10^{-8}$ G Magnetiske felter som skyldes elektriske strømmer i hjernen
- 0,31 – 0,58 G Det jordmagnetiske feltet ved overflata
- 50 G En typisk kjøleskapsmagnet
- 100 G En liten jernmagnet
- 2000 G En liten neodym-magnet
- 15 000 - 30 000 G Magneter i MRI-utstyr (Magnetic Resonance Imaging)

7.9.1 Hovedinndeling av magnetfeltsensorer.

Magnetfeltsensorer deles gjerne inn i *magnetometre* for $B < 1\text{mT}$ og *gaussmetre* for $B > 1\text{mT}$ ³⁷. Det er vanlig at Gauss-metre bruker Hall-effekten, magnetoresistivitet, magnetodioder eller magnetotransistorer som det aktive elementet.

Magnetometre deles gjerne inn i vektor- og skalar-typer.

Vektormagnetometre måler magnetfeltets komponent i en gitt retning. De bruker f. eks. induksjonsspoler, fluxgateprinsippet, SQUID (strøm gjennom et isolerende lag mellom to superledere), magnetoresistivitet og fiberoptikk.

Skalare magnetometre måler magnetfeltets absoluttverdi. To typer dominerer – begge baserer seg på atomkjerners og atomers egenskaper i stoffer – *protonpresesjon* (slingringen i spinnets til atomkjernen) og "*optically pumped manometer*" (bruker Zeeman-effekten, – magnetfeltets virkning på eksitasjon og absorpsjon i alkalisk damp).

36. [http://en.wikipedia.org/wiki/Gauss_\(unit\)](http://en.wikipedia.org/wiki/Gauss_(unit)) og http://en.wikipedia.org/wiki/Magnetic_field

37. mT står for milliTesla som er en måleenhet for magnetisk feltstyrke.



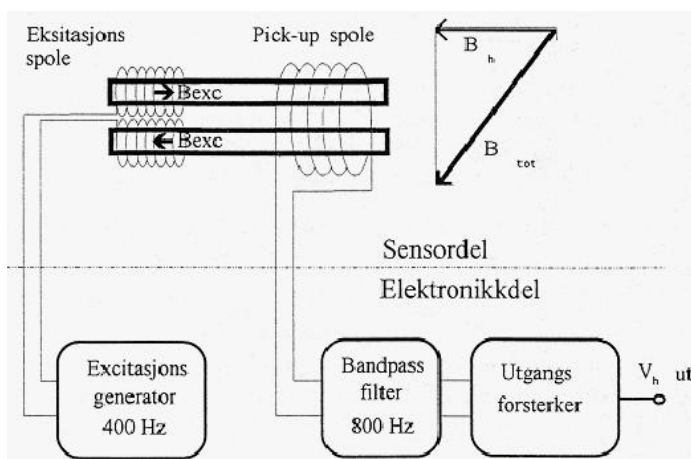
Metningskjernemagnetometeret eller Fluxgate-magnetometeret

Fluxgate-magnetometeret blir mest brukt i dag. Virkemåten har vært kjent fra omkring 1920 og de første sensorene ble utviklet i 1936. Fra 1940–50-åra har fluxgatesensorer vært "arbeidshestene", både i rommet og for diverse målinger fra fly m.m. En sensor alene måler feltstyrkens komponent i sensorens definerte akseretning. Ved å montere tre sensorer montert med 90 grader på hverandre (triaxial fluxgate magnetometer), kan magnetfeltets vektorielle størrelse måles.

I dette tilfellet består sensorelementet av to parallelle staver med relativt høy permeabilitet. Hver stav har en spole/viklinger. Viklingene på de to stavene seriekobles slik at vekselstrømmen på 400 Hz som tilføres, setter opp et like stort, men motsatt rettet magnetfelt i de to stavene. Strømstyrken settes slik at de to kjernestavene går i magnetisk metning i deler av hver halvperiode av vekselstrømmen. En felles pick-up-spole for de to stavene fanger opp et indusert signal

fra kjernen. Om kjernen ikke påvirkes av et eksternt magnetfelt, vil feltet fra de to stavene utligne hverandre, og signalet på pickup-spolen vil bli null. Dersom det finnes et ytre magnetfelt, vil dette forskyve magnetiseringsnivået, og man får asymmetri i de to halvperiodene som endrer nivået til den 2. harmoniske komponenten på 800 Hz. Dette registreres og brukes til å bestemme styrken på det ytre feltet. Slike magnetometre vil måle feltkomponenter i en bestemt retning. De måler ikke det absolutte felt, slik som f.eks. et protonmagnetometer, og må derfor kalibreres mot et absolutt instrument. Når magnetfeltet endres, får vi induert strøm i pickup-spolen. Den induerte strømmen digitaliseres til en prosessor. Strømmen i hvert viklingspar blir kontrollert av en 12-bit digital til analog konverter (DAC). Sensorsignal og feedback-signal blir kontinuerlig overført til prosessorenheten, som kalkulerer verdien på magnetfeltet ved å skalere og summere de mottatte data.

Andre ganger har en behov for mindre magnetfeltsensorer for overvåking, styring og kontroll.

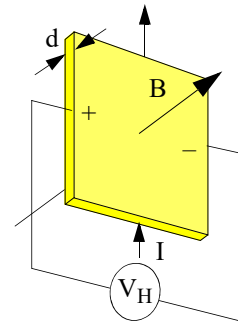


Figur 7.15 Prinsippkisse for et Fluxgate-magnetometer.

7.9.2 Hallsensorer

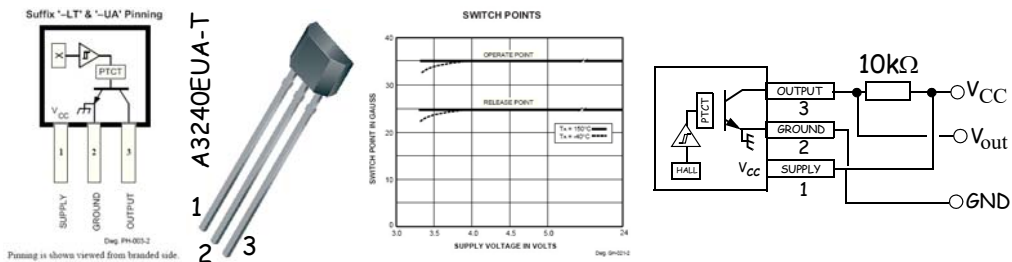
Halleffekten ble oppdaget i 1879 av **Edwin Hall** (1855–1938). Han observerte at det oppsto en elektrisk spenning i et strømførende materiale som ble utsatt for et magnetfelt (se figuren til høyre). Hall oppdaget effekten da han arbeidet med sitt doktorarbeid ved *Johns Hopkins university* i Baltimore i Maryland, USA.

Effekten er en naturlig følge av Lorentz-kraften på ladninger som beveger seg i et magnetfelt. Et magnetfeltet som står loddrett på strømretningen, vil påvirke ladningen med en kraft som står normalt på strømretningen og på magnetfeltet. Dermed vil det oppstå en liten, men målbar ladningsforskjell mellom sidekantene til den elektriske lederen. Dette er vist på figuren til høyre.³⁸



A3240EUA (magnetisk bryter)

Det finnes en rekke billige magnetsensorer i salg som anvender på Hall effekten. Figuren under viser sensoren A3240EUA som produseres av firmaet Allegro. Denne fungerer som en magnetisk styrt bryter. Bryteren vil åpne så snart et tilstrekkelig sterkt magnetfelt nærmer seg komponenten. Slik komponenten er koblet til høyre på figuren under, så vil den være *aktiv lav*, dvs. at spenningen på utgangen legges nær jord når den påvirkes av et magnetfelt. Bryteren slår inn når feltet overstiger ca. 35 Gauss. Har magnetbryteren først slått inn, må nivået under ca. 25 Gauss for at den igjen skal bryte strømmen. Vi sier at komponenten har *hysteresis*. Kretsen fungerer med batterispenninger fra $V_{CC} = 4,2$ V til 24 V, og kan slå av og på strømmer på inntil 25 mA.



En variant av denne kretsen inneholder en såkalt *latch*. Dvs. at når den først er utsatt for et tilstrekkelig sterkt magnetfelt, så holder den bryteren lukket helt til en magnet av motsatt polaritet, åpner bryteren. ELFA har tidligere solgt disse for ca. 15 kr/stk, men de er tatt ut av sortimentet.

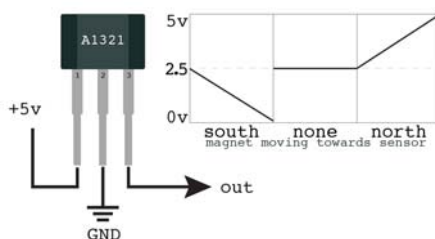
38. http://en.wikipedia.org/wiki/Hall_effect



A1321/2/3 (Lineær Hall-sensor)

Det finnes også lineære Hall-sensorer hvor spenningen på utgangen er lineært varierende avhengig av styrken på magnetfeltet innenfor et visst område. En slik er **A1321** (ELFA 73-339-25, stykkpris ca. kr. 26,-) som produseres av firmaet *Allegro*.

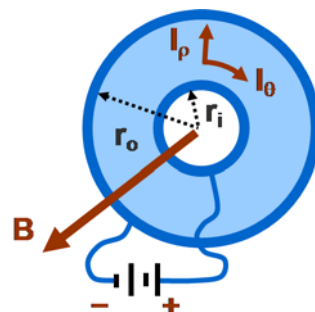
Dersom ingen magnet er i nærheten, vil spenningen ligge midt mellom forskyningsspenningen og jord. Med et økende felt fra en sørpole, vil spenningen synke mot 0V. Med et økende felt fra en nordpol, vil spenningen øke mot forskyningsspenningen (se figuren til høyre).



Forskyningsspenningen skal ligge mellom 4,5 V–5,5 V. Sensoren leveres i tre varianter med forskjellig følsomhet (2,5 mV/G (A1323), 3,75 mV/G (A1322) og 5,0 mV/G (A1321)). Dersom vi for enkelthetskyld sier at området for hver magnetisk polaritet er på ca. 2,2 V og vi velger en sensor med en følsomhet på 5mV/G, vil det være mulig å registrere en variasjon i magnetfeltet for hver av polaritetene fra 0–440 G. Vi registrerer også at det vanskelig lar seg gjøre å måle jordmagnetfeltet (<1G) med en slik sensor uten at man forsterker signalet.

7.9.3 Magneto-resistivitet

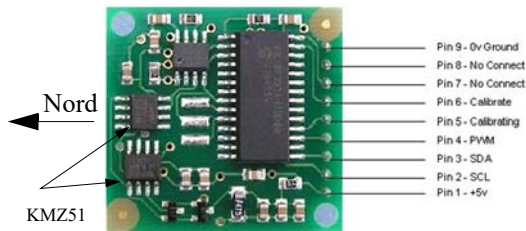
I 1856 oppdaget den skotske fysikeren **Lord Kelvin** (1824–1907) et materiale som endret resistans når det ble utsatt for et magnetfelt. Han benyttet jern- og nikkelholdige materialer i sine eksperimenter, men lyktes ikke å endre resistiviteten med mer enn ca. 5%. En måte og forstå denne effekten på kan knyttes til magnetfeltets evne til å avbøye ladninger i bevegelse, slik at strømveien blir lengre. Dette demonstreres f.eks. ved den såkalte *Corbino disken* (se figuren til høyre). Disken har en ledende ring innerst og en tilsvarende ring ytterst. Det magneto-resistive materialet befinner seg mellom de to ringene. En strøm ledes fra den innerste til den ytterste ringen. Påtrykkes disken et magnetfelt B langs aksen, vil Lorentz-krefter avbøye ladningene slik at de går en lengre vei i materialet, hvilket medfører høyere resistivitet.



I dag benyttes halvledermaterialer (f.eks. indium antimonid) hvor mobiliteten (ledningsevnen) til ladningsbærerne i halvlederen påvirkes av magnetfeltet og dermed endres resistiviteten. På denne måten har en lyktes med å endre resistiviteten opp til 100%.³⁹ Denne teknologien benyttes blant annet i enkelte elektroniske kompasser.

39. <http://en.wikipedia.org/wiki/Magneto-resistance>

7.9.4 Elektronisk kompass – CMPS03



KMZ51 er en integrert krets produsert av Philips som inneholder to magnetoresistive elementer koblet inn i en *Wheatstones målebro*. To slike kretser danner grunnlaget for en kompassmodul som går under betegnelsen CMPS03 (se figuren til venstre). Legg spesielt merke til at de to KMZ51 kretsene er *plassert normalt på hverandre*. CMPS03

måler dermed projeksjonen av magnetfeltet i det planet kretsen er orientert. Kretsen er i utgangspunktet kalibrert for ca. 67° nordlig bredde. Skal den brukes i områder ved andre breddegrader bør den kalibreres på nytt. Dette gjøres ved å legge kretsen horisontalt, for deretter i tur og orden å legge den i de fire himmelretningene, mens kalibreringen utføres. Kalibreringen startes med en kommando.

Kretsen leverer data om feltretningen på to ulike måter. Enten som en avlest digital verdi fra et register via en I^2C buss (SDA og SCL), eller som et pulsbreddemodulert (PVM) signal. Denne modulen er foreslått brukt i forbindelse med Arduino- eller PIC-prosessorer og ble utviklet spesielt i forbindelse med orientering av roboter. Kretsen krever en forskyningsspenning på 5 V (25 mA).

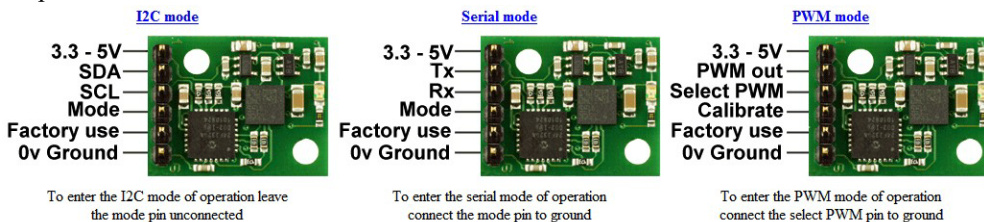
Pris på CMPS03 er i størrelsesorden 22 £ fra Robot Electronics⁴⁰

Elektronisk kompass CMPS10 - med tilt-kompensasjon



CMPS10 fungerer omtrent som CMPS03, men utfører en feltmåling i tre retninger. Dermed kan feltvektoren bestemmes i rommet relativt kortets orientering. Kretsen leverer en heltallsverdi i område 0–3599 i retningene x, y og z. Dette representerer $0\text{--}359,9^\circ$ med oppløsning på $0,1^\circ$. Kretsen kan avleses i ulike formater enten via I^2C -buss, PWM signal eller i seriell form som vist på figuren under. Feltretningen kan i teorien avleses hvert 13,3 ms. Kretsen opererer med en forskyningsspenning på fra 3,3 V til 5 V og har et typisk strømtrekk 25 mA⁴¹.

Pris på CMPS10 er i størrelsesorden 20 £ fra Robot Electronics.



40. <http://www.robot-electronics.co.uk/acatalog/Compass.html>

41. <http://www.robot-electronics.co.uk/hm/cms10doc.htm>



Flere magnetfeltsensorer (moduler) fra *Honeywell* og *Ocean Server Technology Inc.* er vist på Sparkfun's hjemmeside: <http://www.sparkfun.com/categories/23>

7.9.5 Anvendelser av magnetometer

Magnetometre og magnetsensorer har mange forskjellige bruksområder. En av de viktigste anvendelser er måling av magnetiske felt på bakken og i verdensrommet. De nyttes i geofysiske målinger, i borehull og i kartlegging av mineralforekomster fra fly, fartøy og på bakken. Magnetometre er også i bruk som metalldetektorer, i sikkerhetskontroll på flyplasser og til lokalisering av metallgjenstander under jorden og i vann, f.eks. i minedetektorer og i utstyr for oppsporing av undervannsbåter.

I skolen kunne man ha laget og brukt en metalldetektor i forbindelse med geologiundervisningen på 8. trinn, eller i forbindelse med metaller på 9. trinn.

Flere anvendelsesområder:

- Magnetfeltet til sola, planetene, måner, asteroider, kometer ved romferder.
- Jordmagnetismen målt fra satellitter og fly, på overflata, fra skip, i borehull og gruver.
- Elektroniske kompass.
- Geologers søk etter jernholdige mineraler.
- Paleomagnetiske målinger av remanent magnetisme i havbunnen.
- Arkeologiske søk, minesøk og andre metallsøk i jordsmonnet.
- Søk etter ubåter.
- Felt fra kraftlinjer, jordkabler, varmekabler osv.
- UFO-søk i Hessdalen.
- Biologisk neuromagnetisme.
- Lesehoder for magnetbånd og disketter
- Kompasser.
- Mekaniske målinger som f.eks. toleranser for aksler (kamaksel-kontroll), turtallsmålinger.
- Kjøretøydeteksjon i lyskryss og ved trafikktegnringer.
- Sikkerhetskontroll på flyplasser.

Mer stoff finnes i referansene i [13].

7.10 Gass-sensorer

Mangfoldet av gass-sensorer er svært stort og metodene som anvendes for deteksjon er like variert og tilpasset den enkelte gass. I dette avsnittet skal vi se på virkemåten til noen typer gass-sensorer.

7.10.1 Pellistoren⁴²

Selv om de tidligste gass-sensorene så dagens lys i 1920 årene, var det først på slutten av 1950-tallet at utviklingen skjøt fart. Det var primært i Japan at gass-sensorer ble utviklet. Dette skyldtes dels et økende antall gassulykker i japanske gruver og dels den hyppige bruken av gass til koking og oppvarming i japanske hjem.

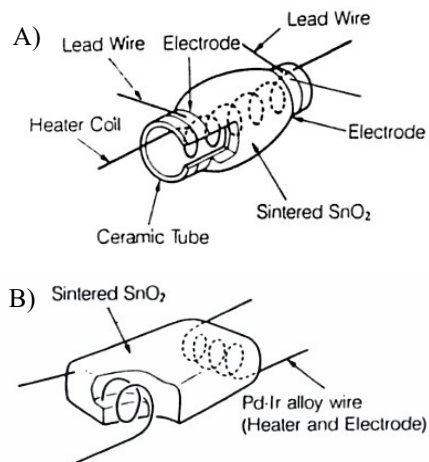
Det finnes mange ulike teknikker for å detektere gass. Her har vi valgt å beskrive en teknologi som benyttes i sensorer som er tilgjengelig fra ELFA til en overkommelig pris, og på den måten kan brukes i prosjekter i skolen og ev. i forbindelse med CanSat.

Man oppdaget tidlig at temperaturen i enkelte stoffer, oppvarmet til noen hundre grader, økte når de ble utsatt for brennbare gasser slik som metan (CH_4), isobutan, butan og propan m.fl. Stoffene var også elektrisk ledende med en gitt resistans som endret seg med temperaturen (vanligvis økende resistans med økende temperatur). Dermed ble en istand til å registrere endringen ved å måle strømmen gjennom materialet.

Et slikt sensorelement går ofte under navnet *pellistor*. Figur 7.16 viser to ulike prinsipper. I figur A er materialet som reagerer med gassen lagt omkring et keramisk rør, materialet er i denne sammenheng sintret tinnoksid (SnO_2). Gjennom røret går en glødetråd som varmer opp tinnoksiden. Denne metoden kalles indirekte siden varmetråden ikke er bakt inn i motstandsmaterialet.

Når tinnoksiden kommer i kontakt med gassen, vil temperaturen øke og resistansen i stoffet øker tilsvarende. Økningen i motstanden registreres i en ytre krets, som igjen f.eks. kan utløse en alarm, eller måle gasskonsentrasjonen.

I figur B er varmetrådene innbakt i motstandsmaterialet. Varmetrådene virker også som elektrode som registrerer endringen i materialets resistans.



Figur 7.16 Katalytisk sensor

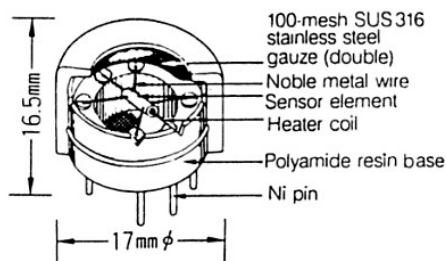
42.LPG - Liquefied Petroleum Gas



Hele sensoren, som er svært liten, monteres i et lite metallhus med et vindu hvor gassen kan slippe inn.

Det finnes i dag en rekke sensorer på markedet som detekterer ulike typer gasser. Over omtalte vi tinnoksid som reagerer på ulike brennbare gasser. Det er imidlertid utviklet en rekke ulike materialer som reagerer på forskjellige gasser.

Tabellen under viser hvordan forskjellige stoffer reagerer optimalt på ulike gasstyper.



Figur 7.17 Sensor montert i metallhus.

TABLE 1. Key Solid-State Gas Sensor R & D Papers in the First Two Decades

Year	Material	Signal	Detecting gas	Researcher	Ref.
1962	ZnO (Thin film)	E.C.	H ² , Alcohol, etc.	Seiyama, et. al.	4
	SnO ₂	E.C.	Combustible gas	Taguchi.	6
1963	SnO ₂ +Pd, Pt, Ag.	E.C.	Combustible gas	Taguchi.	7
1966	ZnO, SnO ₂ , etc.	E.C.	Reducing gas	Seiyama, et. al.	8
	SnO ₂ +Al ₂ O ₃	E.C.	Combustible gas	Taguchi.	9
1967	WO ₃ +Pt	E.C.	H ₂ , N ₂ H ₄ , WH ₃ , H ₂ S	Shaver.	10
	In ₂ O ₃ +Pt	E.C.	H ₂ , Hydrocarbon	Loh.	11
1969	SnO ₂ +SiO ₂	E.C.	Combustible gas	Taguchi.	12
1971	ZnO+Pt+Ga ₂ O ₃	E.C.	CH ₄ , NH ₃	Bott, et. al.	13
1972	SnO ₂ +Pd	E.C.	Propane	Seiyama, et. al.	14



1975	La _{1-x} Sr _x CoO ₃ , etc.	E.C.	Alcohol	Sakurai, et. al.	15
	V ₂ O ₅ +Ag (Thin film)	E.C.	NO ₂	Sakai, et. al.	16
	ZnO+Ga ₂ O ₃ +Pd, Pt.	E.C.	H ₂ , CO, Hydrocarbon	Ichinose, et. al.	17
	TiO ₂	E.C.	O ₂	Tien, et. al.	18
	CoO	E.C.	O ₂	Logothetis, et. al.	19
	Pd MOS FET	T.V.	H ₂	Lundström.	20
1976	Pd/CdS	R.	H ₂	Steele, et. al.	21
1977	SnO ₂ +Pd+ThO ₂	E.C.	CO	Nitta, et. al.	22
1978	γ-Fe ₂ O ₃	E.C.	Propane	Matsuoka, et. al.	23
	Co ₃ O ₄	E.C.	CO	Stetter.	24
	Ag ₂ O	S.P.	Mercaptan	Tsubomura, et. al.	25
	Pd/TiO ₂	R.	H ₂	Tsubomura, et. al.	26
	Metal-phthalocyanine	E.C.	NO ₂	Sadaoka, et. al.	27
	Anthracene	E.C.	Amine, Carboxylic acid	Suzuki, et. al.	28
1979	ZnO (Thin film)	E.C.	Alcohol	Heiland, et. al.	29
1980	SnO ₂ -Ultra fine particle	E.C.	Combustible gas	Abe, et. al.	30
1981	α-Fe ₂ O ₃	E.C.	CH ₄ , H ₂ , etc.	Nakatani, et. al.	31
	ZnO+V ₂ O ₅ +MoO ₃	E.C.	Freon	Shiratori, et. al.	32
1982	ZnO (Single crystal)	E.C.	CO	Jones, et. al.	33
	SnO ₂ (Thin film)	E.C.	Combustible gas	Chang.	34
	SnO ₂ (Thin film)	E.C.	Combustible gas	Sotomura, et al.	35

E.C.: Electric conductivity. T.V.: Threshold voltage. R.: Rectification. S.P.: Surface potential.

ELFA leverer flere gass-sensorer fra firmaet Sencera Co Ltd. Alle av nevnte type har forvarming og er optimalisert for ulike typer gass. Gassene butan, isobutan, propan m.fl. betegnes ofte som LP-gasser (LPG-Liquified Petroleum Gas):

HS-129 - Hydrogen, LPG, Alkohol

HS-131 - Metan, LPG

HS-133 - LPG, Metan,

HS-134 - Karbonoksid, Hydrogen,

HS-135 - Isobutan, alkohol, svoveldioksid, røyk, CO₂.

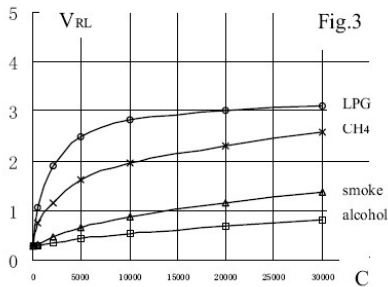


Figur 7.18 Gassensor fra Sencera Co Ltd

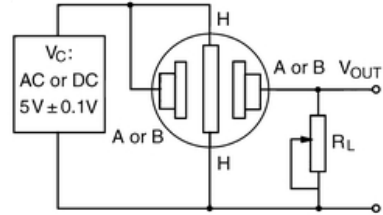


7.10.2 Gassensor LPG, Metan – HS-133

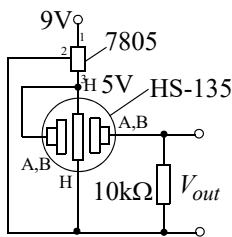
Figur 7.19 viser følsomhetskarakteristikker for ulike gasser for sensoren HS-133. Dvs. spenning som funksjon av konsentrasjonen i ppm (parts pr. million). Spenningen framkommer ved at sensoren kobles inn i en liten krets som vist til høyre på figur 7.19.



Figur 7.19 Følsomhetskurve for ulike gasser.



Figur 6.19 Oppkobling av sensorelement.

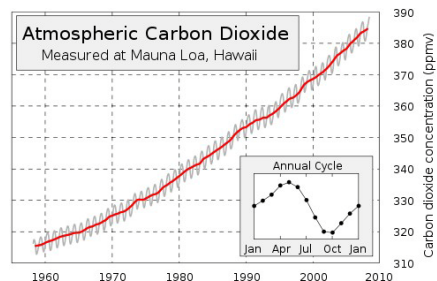


Varmetråden som skal varme opp sensoren til riktig temperatur, må tilføres en nøyaktig spenning på 5 V +/- 0,1 V, derfor benyttes en spenningsregulator som sørger for at spenningen holdes nær 5 V. Spenningen kobles til terminalene H-H. Motstandsmaterialet er koblet inn mellom A-A eller B-B. Den interne motstanden kobles i serie med spenningskilden og en ytre regulerbar motstand. Utgangsspenningen V_{RL} (V_{out}) måles over R_L . R_L justeres til ønsket spenningsnivå.

En R_L på 10 kΩ ga etter ca. 15 min. en hvilespenning V_{out} på ca. 2 V. Det så ut til at ved første gangs bruk så falt spenningen V_{out} fra 5 V ned til 2 V. Oppvarmingen av sensoren trekker ca. 800 mW. Ved å blåse litt butan på sensoren spratt V_{out} umiddelbart opp i ca. 5 V. Ved avbrudd og senere påslag av kretsen steg spenningen opp fra 0 for den stabiliserte seg på ca. 2 V.

7.10.3 CO₂ innholdet i atmosfæren

I forbindelse med CanSat og romvirksomhet kan det være aktuelt å måle ikke brennbare gasser som f.eks. CO₂. Figuren til høyre viser hvordan CO₂-innholdet i atmosfæren har endret seg de siste 50 åra. Som vi ser så har det vært en jevn sesongavhengig økning fra ca. 310 ppm⁴³ til 380 ppm i 2008⁴⁴. Vi ser også en liten tendens til at økningen går stadig raskere.

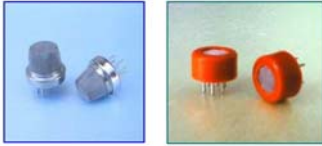


43. ppm - parts pr. million

44. http://en.wikipedia.org/wiki/Carbon_dioxide_in_Earth's_atmosphere

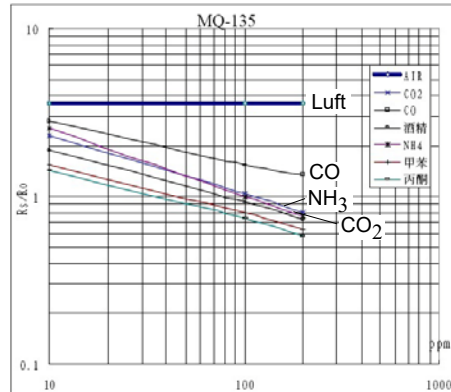
Det er ikke enkelt å finne billige CO₂ sensorer med tilstrekkelig følsomhet. MG-135 kan synes å være en aktuell sensor.

7.10.4 Gass-sensor CO₂ – MQ-135



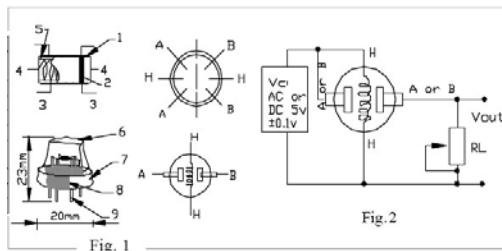
Denne sensoren fungerer etter samme prinsipp som beskrevet foran og er primært beregnet på registrering av ammoniakk (NH₃) og alkohol, men burde la seg bruke til CO₂ så lenge gassen ikke forstyrres av de andre gassene sensoren er følsom for.

Figuren til høyre viser følsomhetskurven for ulike gasser. Vi ser at NH₄ ligger svært nær CO₂. Grafen angir relativ endring i resistans i sensoren som funksjon av konsentrasjonen. Vi ser at kurvene er tegnet for verdier fra 10–200 ppm, men databladet synes å antyde at de kan strekkes noe høyere opp. Dette må imidlertid kontrolleres.



Figuren under viser hvordan sensoren kobles opp med en lastmotstand RL (til høyre) i serie med det aktive elementet i sensoren som er et tynt lag med tinndioksid (SnO₂). Varmeelementet skal forsynes med 5 ± 0,1 V og trekker ca. 800 mW.

Parts	Materials
1 Gas sensing layer	SnO ₂
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceramic	Al ₂ O ₃
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni



MQ-135 selges for 9 € hos Cooking Haacks⁴⁵.

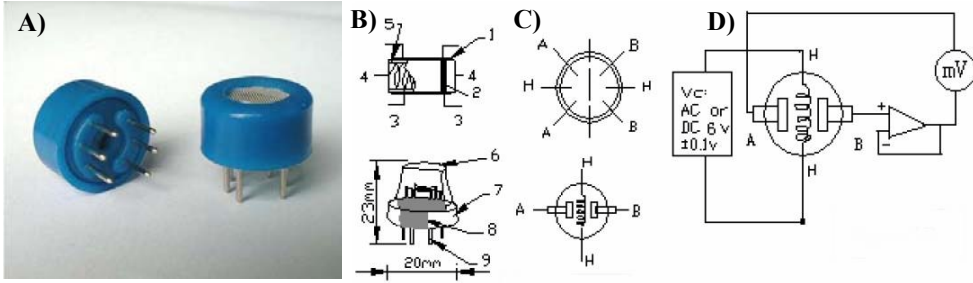
7.10.5 Gassensor CO₂ – MG811

Dersom vi trenger å måle større verdier av CO₂ så er kanskje sensoren MG811 et alternativ. MG811 er basert på et litt annet prinsippet enn beskrevet foran. Komponenten leveres i to varianter (se A og B på figuren under). Oppbyggingen kan kort beskrives slik: Et porselensrør (5) varmes opp av et varmeelement (4). Røret er dekket med et tynt lag med elektrolytt (1) som er tilsluttet elektroder av gull (2) og platina (3). Det hele er bygget inn i et kammer. En metallnetting

45. <http://www.cooking-hacks.com/index.php/shop/sensors/mq135-air-quality.html>



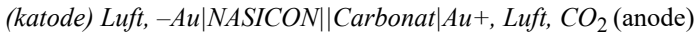
(6) sørger for å slippe gass inn i kammeret. Det hele er montert på en bakelitt-sokkel (8) med forniklede kobberpinner for tilkobling.



Varmeelementet skal tilføres en spenning på $6 \pm 0,1$ V (AC eller DC) (pinne H–H) og krever et strømtrekk på typisk 200 mA (1,2 W). Sensoren kan registrere CO_2 -nivåer fra 350 til 10 000 ppm, hvilket er i overkant av det som kreves for å registrere verdier i atmosfæren, som ligger i området rundt 380 ppm.

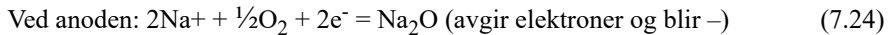
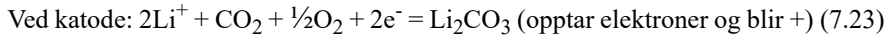
Det aktive elementet er koblet mellom terminal A-B. Tegning D i figuren over viser testoppkoblingen, men kan også benyttes for oppkobling av kretsen i drift. Som vist er det benyttet en operasjonsforsterker med en forsterkning lik 1.

Reaksjonen mellom gass og materiale kan uttrykkes på følgende måte:

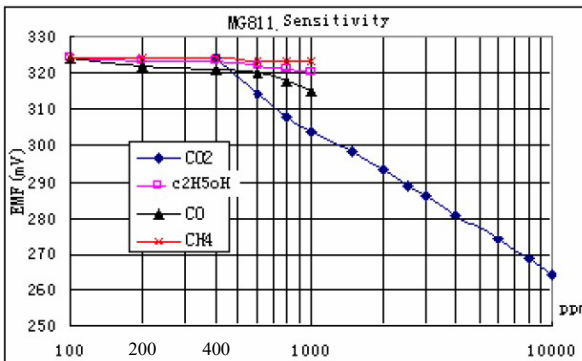


Hvor NASICON sannsynligvis er et halvleder materiale.

Ved hver av elektrodene vil det skje en kjemisk reaksjon som beskrevet av ligningene under:



$$EMF = \frac{E_c - (R \cdot T)}{2F \ln p(\text{CO}_2)} \quad (7.25)$$



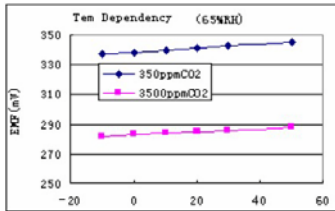
Hvor EMF er spenningen (elektromotorisk kraft) på utgangen, E_c – er en konstant og R – er gasskonstanten, begge relatert til volumet til det aktive materialet i sensoren, T – er absolutt temperatur, F – Faraday's konstant, $p(\text{CO}_2)$ er konsentrasjonen til CO_2 .

Figuren til venstre viser sammenhengen mellom spenningen målt på utgangen og CO_2 konsentrasjonen.

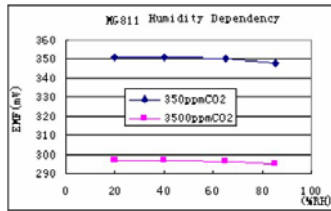


En utfordring med denne type sensor er å hindre at målingene også påvirkes av temperatur og luftfuktighet, parametere en ofte ikke har kontroll over.

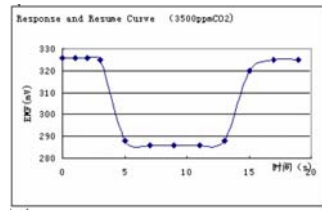
Figuren under viser hvordan målingene påvirkes av temperatur (A) og relativ fuktighet (B). Figuren under til høyre viser responstiden til sensoren, dvs. hvor fort den reagerer på endringer i gasskonsentrasjon.



Avhengighet av temperatur

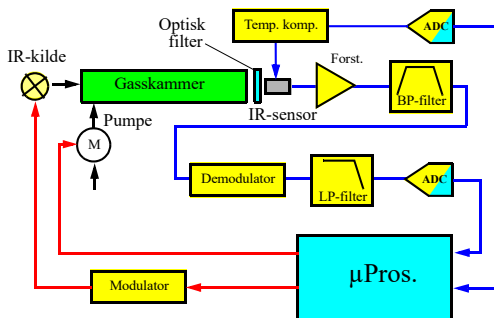


Avhengighet av luftfuktighet



Tidsrespons

7.10.6 Måling av CO₂ ved absorpsjon av IR-stråling (C100)



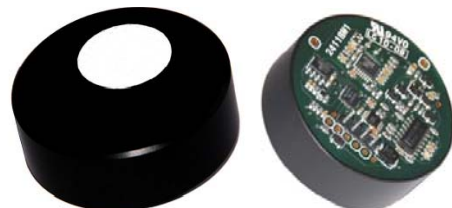
Den skotske firmaet GSS (*Gas Sensing Solution*) baserer sin C100 CO₂-sensor på CO₂-gassens evne til å absorbere IR-stråling. Teknologien går ofte under betegnelsen NDIR (*Non Dispersive Infra-Red sensor*). Lufta pumpes inn i et lite kammer. En IR-stråle sendes fra en lyskilde i den ene enden av kammeret og detekteres av en IR-sensor i den andre enden. Et optisk filter er plassert foran detektoren for å fjerne alt lys unntatt den bølgelengden hvor IR-strålingen absorberes av CO₂ gassen.

Dyrere instrumenter har ofte et parallelt referansekammer med nitrogen. Ved å sammenligne absorpsjonen i de to kamrene, kan man fjerne effekten av uønskede forstyrrelser.

Selv med referansekammer og optiske filter, kan man være plaget av andre gasser som f.eks. H₂O og SO₂ som har delvis overlappende absorpsjonsspektre med CO₂.

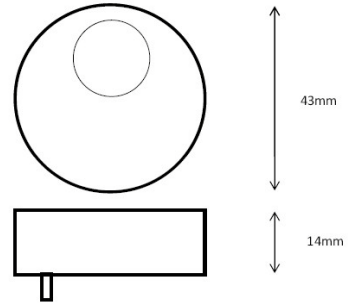
C100 sensoren leverer normalt et mål for CO₂-konsentrasjonen på digital form, men leveres også med analog utgang. Under er listet opp noen nøkkelparametere for C100:

- Måler konsentrasjoner fra 0–2 000 ppm
- Nøyaktighet $\pm 40 \text{ ppm} \pm 3\%$ ved avlesning
- Linearitet: $< 1\%$ av full skala



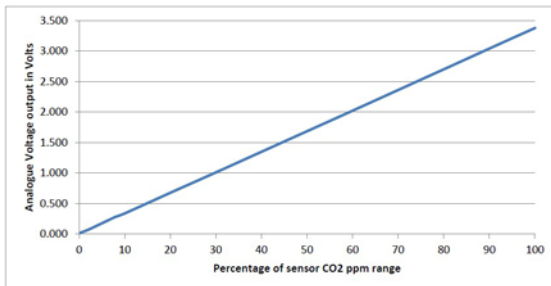


- Temperaturområde: $-25^{\circ}\text{C} - 55^{\circ}\text{C}$
Fuktighetsområde: 0 – 95% relativ fuktighet (ikke kondensert)
Gasstrykk (lufttrykk): 950 mbar – 1050 mbar
- Responstid: 9 sek. – 2 min. (avhengig av krav til nøyaktighet)
- Oppvarmingstid: < 3 min.
- Effektforbruk: 100 mW
- Forsyningsspennning: 3,3 V – 5,5V
- God immunitet til andre gasser og H_2O
- Kan leveres med RS232-grensesnitt og analogt grensesnitt.
- Mer informasjon kan finnes på <http://www.gassensing.co.uk/technical-downloads.asp>

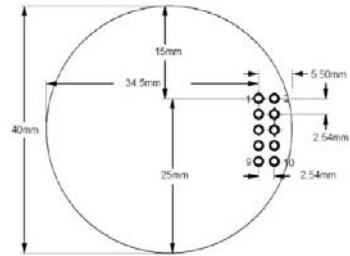


Oppkobling og bruk av C100

CO_2 målinger kan enten leses ut som en spenning (Analogue O/P) eller som et digitalt signal (Sensor Rx (In) / Sensor Tx (out)). Her vil vi kun beskrive avlesning som en spenning.



Output Linearity. (This graph assumes a supply voltage of 3.3V)



Function	Pin #	Pin #	Function
GND	1	2	N/C
+3.3V	3	4	N/C
Sensor Rx (In)	5	6	N/C
Sensor Tx (out)	7	8	Nitrogen Zero
Analogue O/P	9	10	Fresh Air Zero

Figuren over til venstre viser sammenhengen mellom spenningen (Analogue O/P - V_{out}) og CO_2 innholdet som prosent av fullt utslag som i vårt tilfelle er 2000 ppm.

Omregningen fra spenningen ut av sensoren til CO_2 konsentrasjon i ppm kan uttrykkes slik:

$$C_{\text{CO}_2}[\text{ppm}] = \text{FS} \cdot V_{\text{out}} / V_{\text{supply}} \quad (7.26)$$

Hvor:

$C_{\text{CO}_2}[\text{ppm}]$ = Målt CO_2 konsentrasjonen i ppm

FS = Fullskala konsentrasjon – f.eks. 2000 ppm

V_{out} = Målt spenning på utgangen – Analogue O/P

V_{supply} = Forsyningsspennningen som tilføres sensoren – f.eks. 3,3 V



Alle spenninger refereres til jord (GND).

Kalibrering av sensoren kan gjøres på flere måter:

1. *Nitrogen zero*

Pinne 8 ligger normalt høy (3,3 V). Dersom denne legges til jord i minimum 1,5 sek. vil sensoren anta at den målte verdien inneholder null CO₂. For at dette skal fungere som kalibrering, må sensoren befinne seg i CO₂ frie omgivelser, f.eks. i ren nitrogengass.

2. *Fresh air zero*

Pinne 10 ligger normalt høy (3.3 V). Dersom denne legges til jord i minimum 1,5 sek. vil sensoren anta at den målte CO₂-verdien er 450 ppm, som i frisk luft. For at dette skal fungere som kalibrering, må sensoren befinnes seg utendørs i frisk luft.

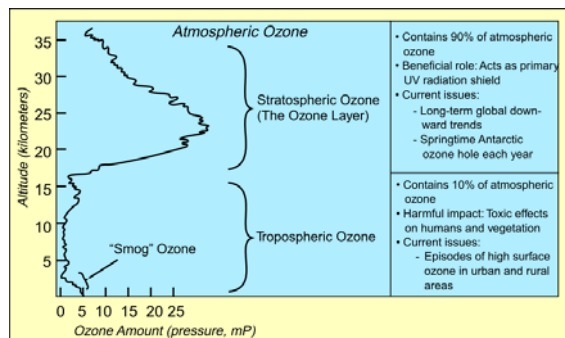
Sensoren kan bestilles fra GSS i Skottland til en-stykkpris på ca. 145£ pluss toll.

For mer informasjon se referanse [8], [9] og [10].

7.10.7 Måling av ozon-konsentrasjon

Ozon i atmosfæren

Gassen ble første gang karakterisert i 1840 av den tysk-sveitsiske kjemikeren **Christian Friedrich Schönbein** (1799–1868). Han ga den navnet Ozon av gresk *ozein* som betyr lukt. Hvilket er relevant siden gassen har en karakteristisk lukt som kan kjennes etter tordenvær. Ozon har de seneste årene vært i fokus etter at en oppdaget at ozon-laget som omgir jorda ble svekket av freongasser o.l. Laget befinner seg hovedsakelig i stratosfæren i en høyde fra 10–50 km (se figuren til høyre). Siden ozon absorberer stråling i området 270–400 nm (ultrafiolett stråling <320 nm), vil dette laget redusere mengden av UV som når ned til bakken. Ozon er giftig, hvilket gjør at en forsøker å redusere mengden ozon i den luften en puster inn. Ozon er også en kraftig drivhus gass ca. 1000 mer effektivt enn CO₂, men bidrar likevel ikke så mye siden mengdene er så små. Totalt sett utgjør ozon bare ca. 0,00006% eller 600 ppb (parts pr. milliard). Siden mengdene er så små vil det være en krevende oppgave å måle konsentrasjoner med enkelt utstyr.

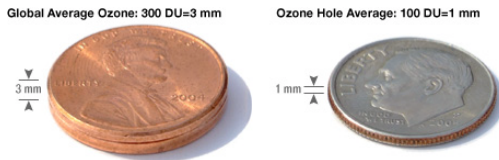


<http://en.wikipedia.org/wiki/Ozone>

Mengden ozon kan måles enten i $\mu\text{mol/mol}$ eller nmol/mol , hvor $1 \mu\text{mol/mol} = 1 \text{ ppm}$ (parts pr. million) og $1 \text{ nmol/mol} = 1 \text{ ppb}$ (ppb – part pr. billion (milliard)). En mer vanlig enhet for måling av ozon konsentrasjon i atmosfæren er *Dobson*. 1 Dobson er det antallet ozon-molekyler som skal til for å lage et 0,01mm lag med ren ozon ved en temperatur og et trykk på 1 atm. I praksis betyr det at dersom en har en atmosfærisk konsentrasjon på 1 Dobson, så vil en søyle med et tverrsnitt på 1 cm^2 inneholde $2,69 \cdot 10^{16}$ molekyler. Atmosfæren vil normalt inneholde 300 ganger så mye

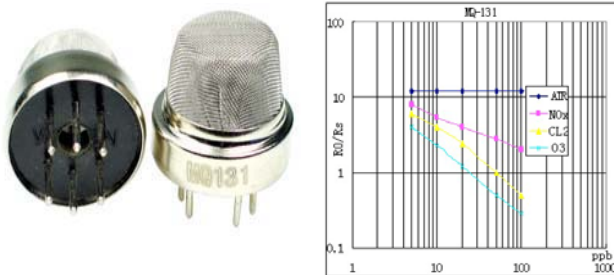


dvs. 300 Dobson. Gjøres denne mengden om til et lag med med konsentrert ozon vil det utgjøre 3 mm. Innenfor ozon-hullet har det vært målt mengder på ned til 100 Dobson. Hvilket utgjør et lag på 1 mm. NASA har illustrert dette forholdet på denne måten⁴⁶.



Ozon-sensorer bygges ved hjelp av flere ulike teknikker:

Ozon (O₃) måling ved hjelp av pellistor-element



Denne sensoren anvender en lignende metode som beskrevet foran og som brukes for en rekke gasser. En variant som egner seg for O₃ er MQ131⁴⁷. Det gassfølsomme materialet er SnO₂. Av grafen til høyre på figuren ser vi at sensoren er følsom for verdier helt ned til 5 ppb (parts pr. billion). Sensoren kobles på samme måte

som omtalt tidligere. Den kan derfor være aktuell for måling av nivåer slik vi finner dem i atmosfæren.

H-MOS (*Heated Metal Oxide Semiconductor*)

Denne typen sensor fungerer ved at en varmer opp et lite stykke substrat av platina til ca. 300°F. Ved denne temperaturen blir substratet svært følsomt for ozon. Ved hjelp av halvlederteknologi genereres en spenning proporsjonal med ozon-konsentrasjonen. Denne teknologien gir en relativt langsom sensor som trenger lang oppvarmingstid (timer) og gir langsom respons (1–2 min), til gjengjeld er den svært lineær. Røyk, klorider og andre organiske stoffer vil gi målefeil [14]d.

Elektrokjemisk sensor

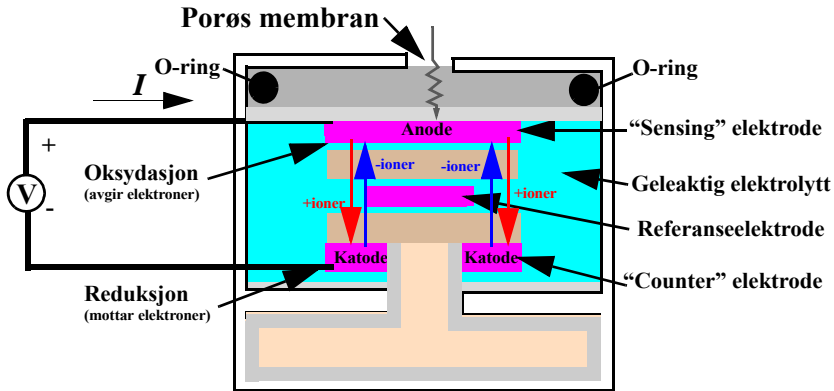
Denne typen sensorer gir langt raskere respons (1–2 sek). Den elektrokjemiske sensoren består av en porøs membran som tillater gasser å trenge inn i en celle som består av en geleaktig elektrolytt med tilhørende elektroder. Anoden ligger nærmest membranen og katoden lengst borte, som vist på figuren til høyre. Når gassen trenger inn i elektrolytten vil den forandre det elektrokjemiske potensialet mellom elektrodene idet det skjer en oksidasjon (avgir elektroner) ved den øverste elektroden og en reduksjon (opptar elektroner) ved den nederste. Det skapes dermed en strøm i

46. <http://ozonewatch.gsfc.nasa.gov/facts/dobson.html>

47. MQ131 selges av firmaet: Futurlec http://www.futurlec.com/Ozone_Gas_Sensor.shtml



elektrolytten idet positive ioner flyter mot katoden og negative ioner mot anoden. Gasser som lett reduseres (opptar elektroner), som oksygen, nitrogen, oksyder og ozon, reduseres ved katoden, mens gasser som lett oksideres (avgir elektroner), som CO, NO₂ og H₂S, oksideres ved anoden.



Figur 7.20 Elektrokjemisk sensor for måling av bl.a. ozon-konsentrasjon.

Denne potensialforskjellen forsterkes og registreres av tilhørende elektronikk. Signalet vil være proporsjonalt med konsentrasjonen av f.eks. ozon dvs. partialtrykket til ozonen i lufta som omgir sensoren. Målingen er svært nøyaktig og angir verdier i området ppb (parts pr. billion). Teknologien gir god reproducerbarhet og en nøyaktighet ned mot 0.01 ppm (10 ppb), men målingene ødelegges lett av klorider og elektromagnetisk støy (EMF/RFI). Dessuten er denne typen sensor følsom for temperaturvariasjoner og har en relativt kort levetid [14]b og d.

Denne type sensorer egner seg derfor for måling av ozon i stratosfæren og blir ofte brukt ved ballongslipp som kan komme opp i ca. 30 000 meters høyde.

7.11 Partikkelsensorer

Slike detektorer anvendes bl.a. til å måle støvmengde, røyk og annen forurensning. Her vil vi primært se på røykdetektorer.

7.11.1 Røykdetektor (røyk, varme)

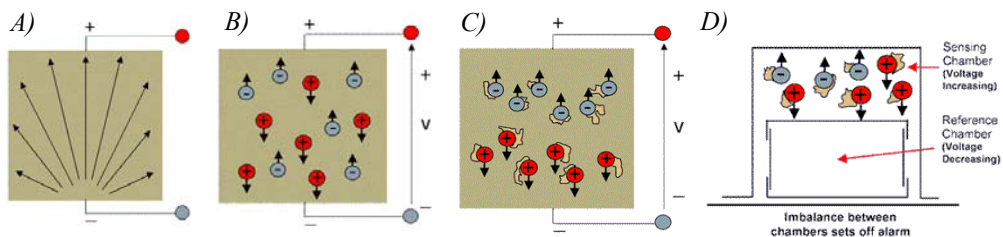
Det finnes hovedsakelig tre typer røykdetektorer:

1. Ioniserende røykdetektorer
2. Optiske/fotoelektriske røykdetektorer
3. Infrarøde røykdetektorer



Ioniserende røykdetektorer [15]

I et ioniseringskammer inne i detektoren ligger en liten radioaktiv kilde (americium 241) som stråler ut alfa-partikler, dvs. heliumkjerner (figur 7.21 A). Denne ioniserer lufta inne i kammeret slik at den blir svakt elektrisk ledende. Ledningsevnen bestemmes ved hjelp av en elektrisk probe som måler den svake strømmen gjennom den ioniserte lufta i kammeret (figur 7.21 B).



Figur 7.21 Ioniserende røykdetektor.

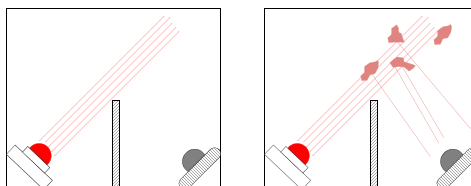
Kammeret er åpent slik at røykpartikler kan komme inn. Røykpartiklene som siver inn i kammeret vil feste seg til ionene og endre den svake elektriske strømmen gjennom kammeret (figur 7.21 C). Den tilhørende elektronikken registrerer denne endringen og alarmen går. Det er heller ikke uvanlig at ioniseringskammeret er delt i to, hvor den ene delen er lukket og fungerer som et referansekammer (figur 7.21 D).

Energien i alfa-partiklene er ikke kraftigere enn at de kan stoppes av noen få cm luft eller et stykke papir. Selv om strålingskilden er svært svak og ikke utgjør noen helsefare, så bør denne typen røykvarslere behandles om spesialavfall.

Ioniserende røykdetektorer egner seg for deteksjon av **små røykpartikler** (ned til 0,01 mikrometer), som oppstår i branner som utvikler seg raskt. Slike partikler er ikke synlige med det blotte øye. Slike detektorer klarer imidlertid ikke å skille små støvpartikler fra damp og stekos, og er dermed utsatt for falske alarmer. Dessuten er de ikke så følsomme for store partikler som gjerne oppstår ved ulmebranner.

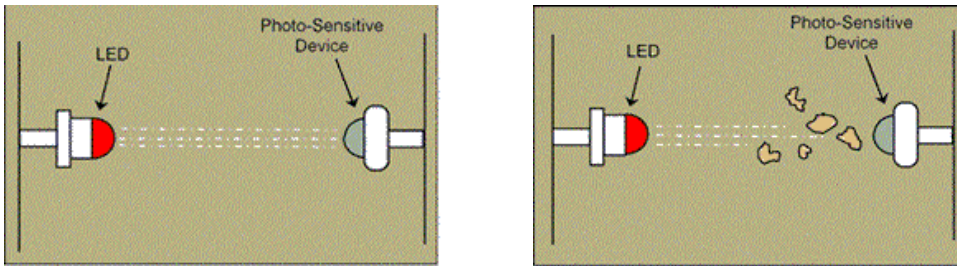
Optiske og fotoelektriske røykdetektorer [15]

Også i denne typen røykvarslere finner vi et røykkammer. En lyskilde er plassert i den ene enden av et lystett kammer. En følsom fotoelektrisk sensor er plassert et annet sted i kammeret. Normalt vil ikke lyskilden belyse sensoren. Kommer det derimot røykpartikler inn i kammeret, vil partiklene reflektere lys som treffer detektoren slik at alarmen går.



Figur 7.22 Prinsippet for en fotoelektrisk røykdetektor, "refleksjonstype".

En annen variant baserer seg på at røykpartikler kommer i veien for lysstrålene fra lyskilden slik at lyset som treffer sensoren dempes.



Figur 7.23 Prinsippet for en fotoelektrisk røykdetektor, "transmisjonstype" [15].
LED er en lysdiode (LED - Light Emitting Diode).

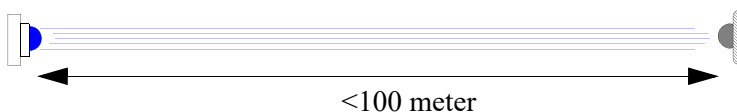
Fotoelektriske røykvarslerer er vanligvis mer følsomme for **større røykpartikler** enn den ioniserende røykdetektoren. Slike partikler oppstår gjerne ved ulmebranner og branner som utvikler tykk røyk. Sensoren er tilsvarende mindre følsom for små røykpartikler som oppstår ved branner som utvikler seg raskt med åpen flamme.

Etter som bruken av flammedempende materialer er blitt vanligere, har den optiske detektoren tatt i bruk mer og mer og er etter hvert blitt den vanligste typen røykvarsler. Dessuten unngår en avfallsproblemene med ioniserende kilder.

Infrarød røykdetektor

Denne typen røykdetektorer består av to deler. En del som sender ut infrarød stråling (IR-sender), og en del som mottar den utsendte strålingen (IR-mottaker). Strålen passerer det området som skal overvåkes med hensyn til røyk. Dersom strålen dempes på grunn av at røykpartikler skygger for strålegangen, vil mottakeren motta et litt svakere lyssignal som utløser alarmen.

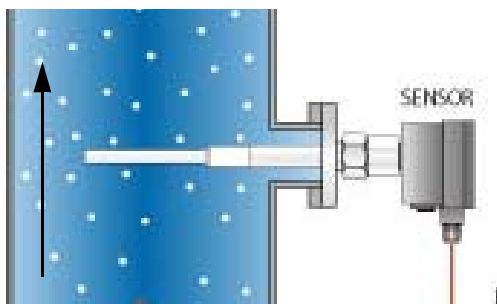
Slike detektorer kan dekke områder på opptil 100 meter og egner seg godt for industrihaller og større lagerrom. Utstyret må imidlertid monteres slik at ikke annen trafikk kommer inn i strålegangen. Det er ikke vanskelig å tenke seg at lignende løsninger kan brukes til å avsløre inntrengere dersom utstyret monteres i passende høyde.



Figur 7.24 IR-røykdetektor.



7.11.2 Støvdetektorer (elektrodynamisk teknikk)



For deteksjon av partikkel- eller støvkonsentrasjon brukes ofte andre teknikker enn optiske. En vanlig teknikk er å benytte en såkalt elektrodynamisk teknikk som baserer seg på at ladde partikler treffer en måleprobe installert i en luft- eller gasstrøm hvor partikkelkonsentrasjonen skal måles. Slike prober går ofte under betegnelsen PCME-prober. De ladde partiklene som treffer proben vil skape en strøm av ladingspulser som

danner grunnlaget for målingen. Jo høyere frekvens av pulser, jo høyere konsentrasjon. Slik proben er utformet vil det være en lineær sammenheng mellom pulsfrekvens og konsentrasjon.

Partikler som legger seg på proben vil gi et konstant signal som kan filteres bort. Dersom en kun tar hensyn til de stadige endringene i ladning (pulsfrekvensen) vil en få et mål for øyeblikksverdien til partikkelkonsentrasjonen. Dersom en ikke har en partikkelstrøm, kan dette skapes ved at lufta eller gassen pumpes forbi proben med en konstant hastighet [16]a.

Med denne teknikken kan en måle konsentrasjoner fra $0,01 \text{ mg/m}^3$ til $10\,000 \text{ mg/m}^3$. Metoden regnes for å være stabile og lite påvirkelige av opphoping av støv på proben eller av farten til gassstrømmen. DX800, DX810 og DX820 m.fl. er varianter av slike prober som er i bruk [16]b. Ingen av disse synes spesielt egnet for CanSat-bruk på grunn av pris og kompleksitet.

7.12 GPS-modul

GPS eller *Global Positioning System* består av 24 satellitter som kretser omkring jorda med en omløpstid på 11 t 58 min. i en høyde av ca. 20200 km over jordoverflata. Normalt vil dette antallet være tilstrekkelig for, til enhver tid, å ha fri sikt til 8–10 satellitter i åpent terreng. Hver satellitt sender ut et kodet tidssignal som mottas av mottakerne. I tillegg til å inneholde informasjon om nøyaktig tid, inneholder signalet tidspunkt for utsendelse og en lang kode som mottakerne bruker for nøyaktig å bestemme tidspunktet for mottatt signal. På denne måten kan mottakersystemet bestemme tiden hvert av signalene bruker fra hver av satellittene til mottakeren. De målte tidsforsinkelsene brukes så til å beregne posisjonen til mottakeren [3].

En **GPS-modul** er et sett av GPS mottakere som gjør det mulig å følge flere GPS-satellitter samtidig. Seks er ikke uvanlig. 4 er et minimum for å kunne beregne koordinater pluss høyde.



7.12.1 EM-411 (GPS-modul)

Dette er en meget kompakt og billig GPS-modul som er i salg hos ELFA (78-400-02, kr. 413,75,- inkl. MVA⁴⁸). Modulen har følgende data:



- Bygget opp rundt chip-settet SiRF Star III
- Mottakerfrekvens: 1575,42 MHz
- Antall kanaler: 20
- Følsomhet: -159 dBm
- Nøyaktighet: 10 m, 2D rms⁴⁹
5 m, 2D rms WAAS⁵⁰
- Farten: 0,1 m/s (farten som er benyttet med den nevnte nøyaktighet).
- Tid: 1 μ s (største avvik mellom klokka i mottaker og klokka i satellitt)
- Dato: WGS-84⁵¹ (World Geodetic System – 1984)
- Låsetid: Lås på nytt 0,1 sek. i gjennomsnitt
Varm start 1 sek. i gjennomsnitt
Lunken start: 38 sek. i gjennomsnitt
Kald start 42 sek. i gjennomsnitt
Dette er viktig å merke seg siden den aktive tiden til CanSat er så kort
- Maks verdier: Maks. målehøyde: 18 000 meter
Maks. hastighet 515 m/s
Maks. aks. 4g
Maks. spinn 20 m/s
- Effekt: Spenning: 4,5 – 6,5 V DC
Strømforbruk: 60 mA
- Dig. utgang: Spenningsnivå: TTL nivå (0 – 2,85 V)
RS-232
Baud rate: 4 800 baud (symboler/sek.)
Format: NMEA 0183 GGA, GSA, GSV, RMC, VTG, GLL
- Størrelse: 30 x 30 x 10,5 mm
- Temperaturområde: -40°C til 85°C

48.Pr. 20.08.16

49.rms - root - mean - square, som gir en slags midlere verdi for feilen over tid

50. WAAS - Wide Area Augmentation System. Satellitt-bakkesystem som utfører korreksjoner. Dette kan gi en betydelig forbedring av nøyaktigheten (ned til 3 m). <http://www8.garmin.com/aboutGPS/waas.html>

51. World Geodetic System 1984 - er det koordinat frammeverket som er vanlig å bruke i dag. Systemet ble etablert i 1984 og revidert i 2004. http://en.wikipedia.org/wiki/World_Geodetic_System

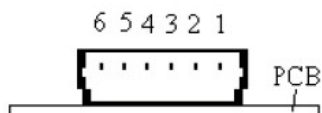


Pinning

Figuren til høyre viser kretsens pinning.

1. GND – Jord (kobles til minus på batteriet)
2. VIN – DC power + 4,5 til 6,5 V DC
3. TX – Utgang som leverer posisjonsdata til prosessor
4. RX – Inngang for overføring av kommandoer til mottakersystemet
5. GND – Jord (kobles til minus på batteriet)

- 1 : GND
- 2 : VIN
- 3 : TX
- 4 : RX
- 5 : GND
- 6 :

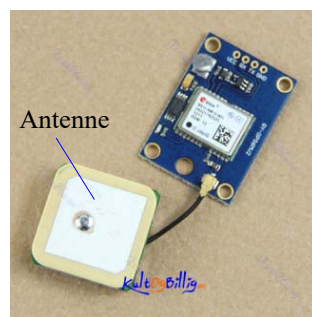


7.12.2 GY-NEO6MV2⁵² Flight (GPS-modul)

Teknisk - spesifikasjon

Dette er en annen vanlig brukt GPS-modul. Den kan bl.a. kjøpes fra www.kultogbillig.no for kr. 198,- inkl. MVA, eller fra Deal-extreme for kr. 128,83. Den består av to separate enheter forbundet med en koaksialkabel. Det ene er antennemodulen, den andre analyse- og datamodulen.

- Enheten er bygget opp rundt chip-settet: ARM
- Mottakerfrekvens: 1575,42 MHz
- Antall kanaler: 50 kanaler
- Følsomhet: -161dBm ("Tracking & Navigation)
- Nøyaktighet: - Horisontal: 2,5 m
- Hastighet: 0,1 m/s
- Kurs retning: 0,5°
- Tid: 30 ns (tidspulssignalets nøyaktighet)
- Dato: Leverer data og tid ned til sekunder
- Låsetid: Varm (Hot) start: 1 sek. i gjennomsnitt
Kald start: 27 sek. i gjennomsnitt
Dette er viktig å merke seg siden den aktive tiden til CanSat er så kort
- Maks verdier: Maks. målehøyde: 50 000 meter
Maks. hastighet 500 m/s
Maks. akselerasjon 4 g
Maks. oppdatering 5 Hz



⁵². Datablad: <https://www.openimpulse.com/blog/wp-content/uploads/wp-content/uploads/wpsc/downloadables/GY-NEO6MV2-GPS-Module-Datasheet.pdf>



- Forsyning: Spenning: 2,7 – 3,6 V DC
Strømforbruk: Max. 47 mA
- Dig. utgang: Spenningsnivå: TTL nivå (0 – 2,85 V)
UART, USB (12 Mbit/sek), SPI (100 kbit/sek)
Baud rate: 9 600 baud (symboler/sek.)
Format: NMEA GSV, RMC, GSA, GGA, GLL, VTG, TXT
- Størrelse: 25 x 25 x 8 mm (antennemodul) +
25 x 35 x 5 mm (kontrollmodul)
- Temperaturområde: –40°C til 85°C
- Backup batteri: For å holde data ved “Power down”

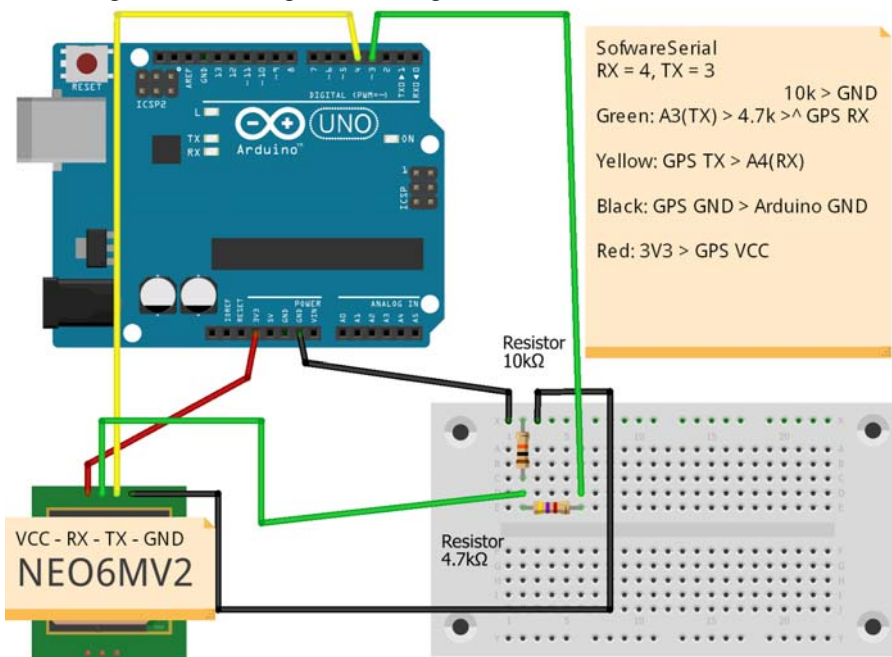
Oppkobling mot Arduino UNO

Kontroll-modulen kobles til Arduino UNO via fire terminaler:

V _{CC}	3,3 V
Rx (GPS)	Pin 3 (fungerer som Tx – UNO via spenningsdeler)
Tx (GPS)	Pin 4 (fungerer som Rx – UNO)
GND	GND



Ved å lodde en stiftlist til terminalene kan en lett koble opp modulen på et koblingsbrett for testing som vist i figuren under⁵³.



53. <http://arduinostuff.blogspot.no/2014/05/neo6mv2-gps-module-with-arduino-uno-how.html>



Legg spesielt merke til at det benyttes to motstander. Dette er for å senke signalspenningen fra UNO'en som har 5 V spenningsforsyning og GPS-modulen som har en spenningsforsyning på 3,3 V.

Behandling av GPS-data

Dette er omtalt i avsnittet 5.6 på side 104.



8 Laboratorieoppgaver

Her beskrives tre laboratorieoppgaver gitt under kurset Romteknologi CanSat.

8.1 Laboratorium 1 – Datainnsamling med CanSat

8.1.1 Del 1 - Måling av trykk og temperatur som funksjon av høyden - «Primary mission»

Forutsetninger

- Grunnversjonen av CanSat er bygget opp og testet
- Trykk og temperatur kan avleses
- Ev. innsamlede data kan legges på fil

Tilgjengelig utstyr

- I tillegg til vanlig labutstyr og PC vil følgende være tilgjengelig
- Termometer (et til hver gruppe)
- Hårføner (to på deling)
- En lufttrykksmåler (en på deling)

Problemstilling:

- Vurder kvaliteten til temperatur, trykk- og høydemålinger gjort under ballongslippet

Oppgave:

Oppgaven består av flere deler:

- Kontroller at avlest temperatur fra sensorkortet ved romtemperatur og en temperatur til, er i overensstemmelse med referanseinstrument (termometer)
- Legg ev. inn en korleksjon i CanSat-programvaren eller i etterbehandlingen
- Kontroller avlest trykk fra sensorkortet og sammenlign med dagens lufttrykk målt med referanseinstrument (barometer)
- Legg ev. inn en korleksjon i CanSat-programvaren eller i etterbehandlingen
- Legg inn omregningsformler fra trykk til høyde
- Begrunn de valgene som er gjort under veis og diskuter forventet nøyaktighet og avvik på målinger gjort under ballongslippet
- Gjennomfør slippet og samle data
- Analyser og diskuter resultatene



Vedlegg - Dokumentasjon oppgave a:

- *The CanSat book* (generelt tilgjengelig)
- Dokumentasjon av CanSat hardware (sensorkortet)
- Dokumentasjon av aktuelle sensorer og datablader

8.1.2 Del 2 - Måling med selvvalgt sensor som funksjon av høyden - «Secondary mission»

Tilgjengelig utstyr:

I tillegg til nødvendig verktøy og PC vil følgende være tilgjengelig:

- Lysmåler (en til hver gruppe som jobber med denne, inntil fem)
- Lyskilder (en til hver gruppe som jobber med denne, inntil fem)
- Luftfuktighetsmåler (en på deling)
- Strømforskyninger, koblingskort, nødvendig kabler, verktøy (et sett til hver gruppe)
- Multimetere (en til hver gruppe)
- Luftfuktighetssensor HiH4000 (en til hver gruppe som jobber med denne)
- Lyssensorer SFH300 (en til hver gruppe som jobber med denne)
- Lysfølsom motstand (LDR)
- Temperatursensor (LM35/NTC) med tilhørende operasjonsforsterker (TS920)
- Akselerometer (MMA7361L)
- Motstandsett (tilstrekkelig utvalg og mengde)
- Koblingsbrett og diverse kabler (en til hver gruppe)
- Veroboard, kontakter og kabler for oppkobling

Problemstilling:

- Vurder kvaliteten til tilleggsmålingen (“Secondary mission”) dere har gjort under ballongslippet (Lysstyrke, luftfuktighet, temperatur...)

Oppgave:

Hver gruppe skal arbeide med én sensor (ev. flere om de rekker det). Oppgaven består av flere deler:

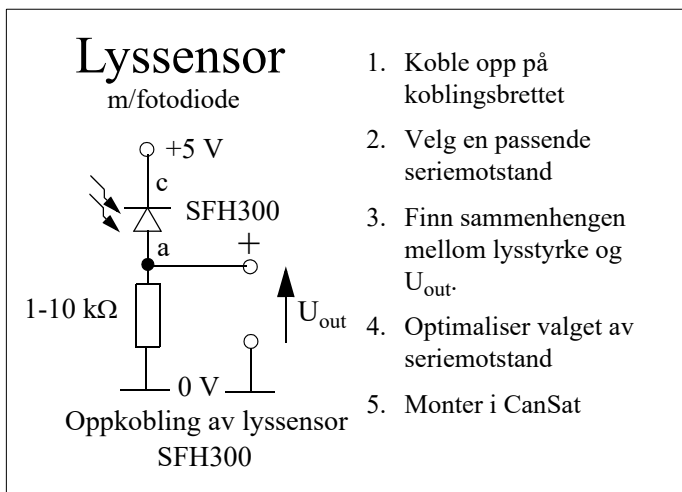
- Velg en sensor (ikke helt fritt valg). Studer databladet til den aktuelle sensoren
- Koble opp sensoren på koblingsbrettet og finn en sammenheng mellom
- den aktuelle størrelsen (Lysstyrke, luftfuktighet, temperatur, akselerometer...) og spenningen på utgangen av sensoren.



- Koble sensoren opp mot CanSat ved hjelp av lab
- . kabler og registrer data. Legg inn nødvendig omregning for optimal registrering av den aktuelle størrelsen
- Begrunn de valgene som er gjort under veis og diskuter forventet nøyaktighet og avvik på målinger gjort under ballongslippet
- Monter den aktuelle sensoren i CanSat på et lite testkort, koble den til sensorkortet/kontrollkortet. Begrunn valg av plassering og orientering.
- Diskuter mulige anvendelser av målingene gjort av den aktuelle sensoren
- Gjennomfør slippet og samle måledata fra den aktuelle sensoren
- Analyser og diskuter resultatene

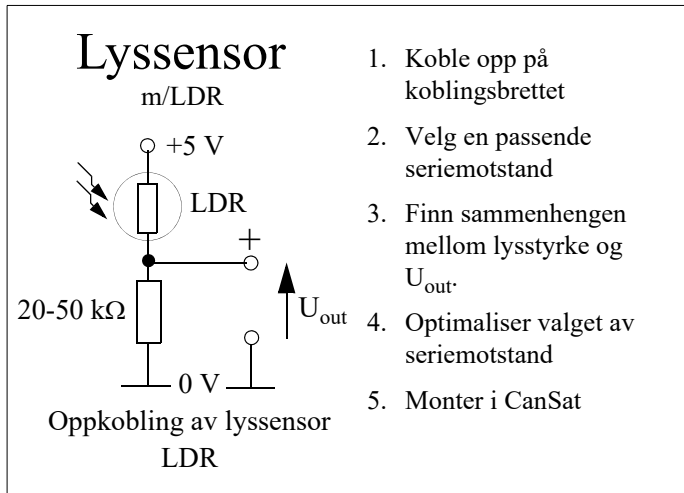
Vedlegg A – Forslag til testoppkoblinger

- Forslag til oppkobling for karakterisering av lyssensorer ved bruk av fotodiode:

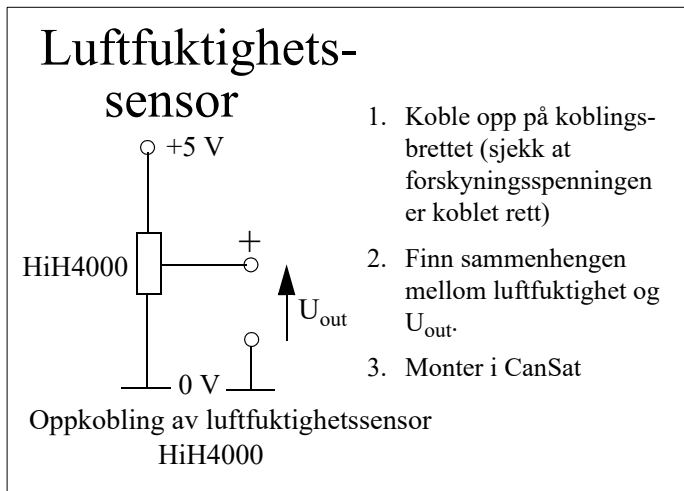




- Forslag til oppkobling for karakterisering av lyssensorer ved bruk av LDR:

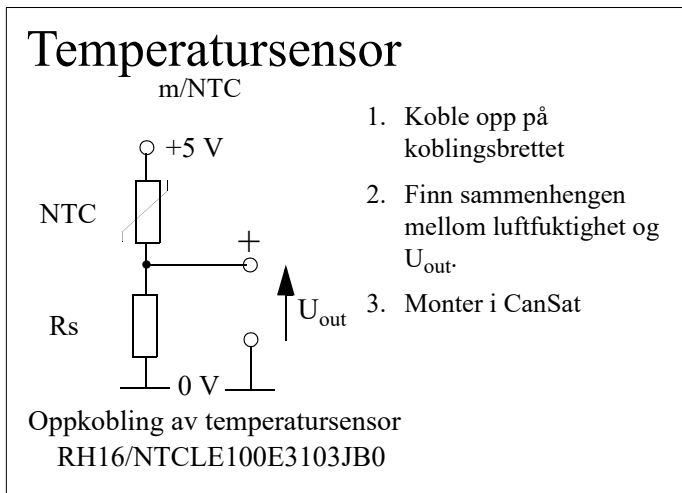


- Forslag til oppkobling for karakterisering av luftfuktighetssensorer:

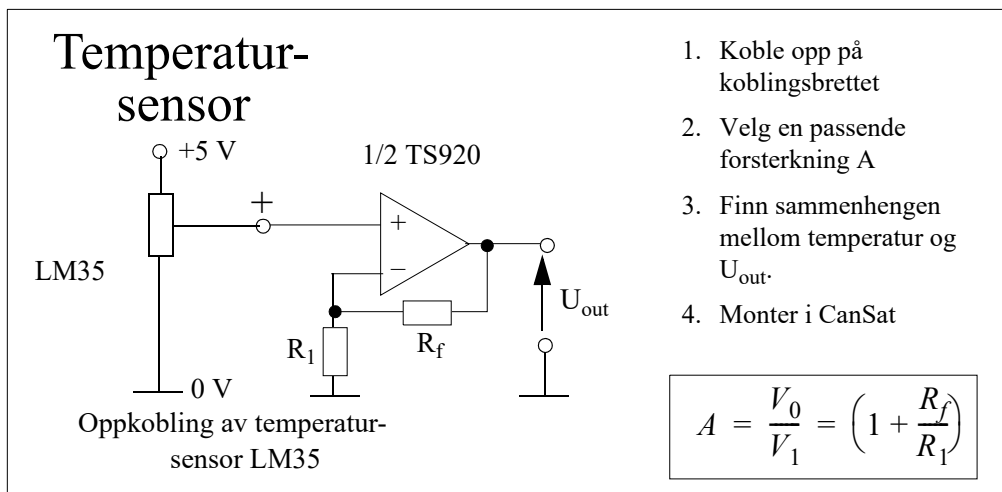




- Forslag til oppkobling for karakterisering av NTC motstand:



- Forslag til oppkobling av temperatursensor med operasjonsforsterker



- Oppkobling av akselerometeret – se egen beskrivelse

Vedlegg B – Dokumentasjon oppgave:

- Wang, *Sensor Circuit for CanSat use*
- Wang, Mathisen, *The CanSat book* (generelt tilgjengelig)
- Rossing, *Grunnleggende elektronikk og sensorteknikk*
- Dokumentasjon av CanSat hardware (kontrollkort og sensorkortet)



- Dokumentasjon av aktuelle sensorer og datablader
 - SFH300 (Lyssensor)
 - LDR (Lysfølsom motstand)
 - HiH4000 (Luftfuktighetssensor)
 - MMA7361L (Akselerometer)
 - NTCLE100E3103JB (NTC – temperatursensor)
 - RH16 (NTC - temperatursensor)
 - TS920 (Operasjonsforsterker)

Vedlegg C Tips for rapportskrivning

Laboratorierapporten skal skrives i henhold til vanlige retningslinjer for rapportering (se eget skriv). I tillegg til at rapporten besvarer problemstillingene, bør den inneholde følgende:

- Beskrive CanSat i korte trekk med vekt på sensordelen og datainnsamlingen
- Beskrive den valgte sensoren i henhold til datablader
- Dokumenter ev. utled de matematiske sammenhengene brukt i forbindelse med oppgavene
- Vurdere ellers hva som bør være med for å gi en helhetlig framstilling
- Sammenlign om mulig, målinger gjort av flere CanSat-sonder gjort under omtrent like betingelser



8.2 Laboratorium 2 – Fallhastighet

Forutsetninger

- Grunnversjonen av CanSat er bygget opp og testet
- Trykk og høyde kan avleses
- Innsamlede data kan legges på fil og analyseres

Tilgjengelig utstyr:

I tillegg til nødvendig verktøy og PC vil følgende være tilgjengelig:

- Programvare for simulering av bane ved slippet
- Ev. materialer for framstilling av fallskjerm

Problemstilling:

- Beskriv en framgangsmåte for beregning av fallhastighet for en CanSat-sonde i fallskjerm og bedøm kvaliteten til metoden ved hjelp av målinger ved ballongslipp

Oppgave:

Oppgaven består av flere deler:

- Gjør beregninger og lag en fallskjerm som tilfredsstiller kravene til fallhastighet
- På bakgrunn av tilgjengelige data, utfør en simulering med anbefalt programvare
- Begrunn de valgene som er gjort under veis og diskuter forventet nøyaktighet og avvik på målinger gjort under ballongslippet
- Gjennomfør slippet og samle data som beskriver fallhastigheten
- Analyser og diskuter resultatene

Laboratorierapport:

Laboratorierapporten skal skrives i henhold til vanlige retningslinjer for rapportering (se eget skriv). I tillegg til at den besvarer problemstillingen og beskriver framgangsmåten, bør den inneholde følgende:

- Beskrive i korte trekk simuleringsprogrammet
- Dokumentere ev.
- å utlede de matematiske sammenhengene brukt i oppgaven.
- Ellers å vurdere hva som bør være med for å gi en helhetlig framstilling av oppgaven

Vedlegg – Dokumentasjon:

- The CanSat book (generelt tilgjengelig)
- Dokumentasjon av simuleringsprogram

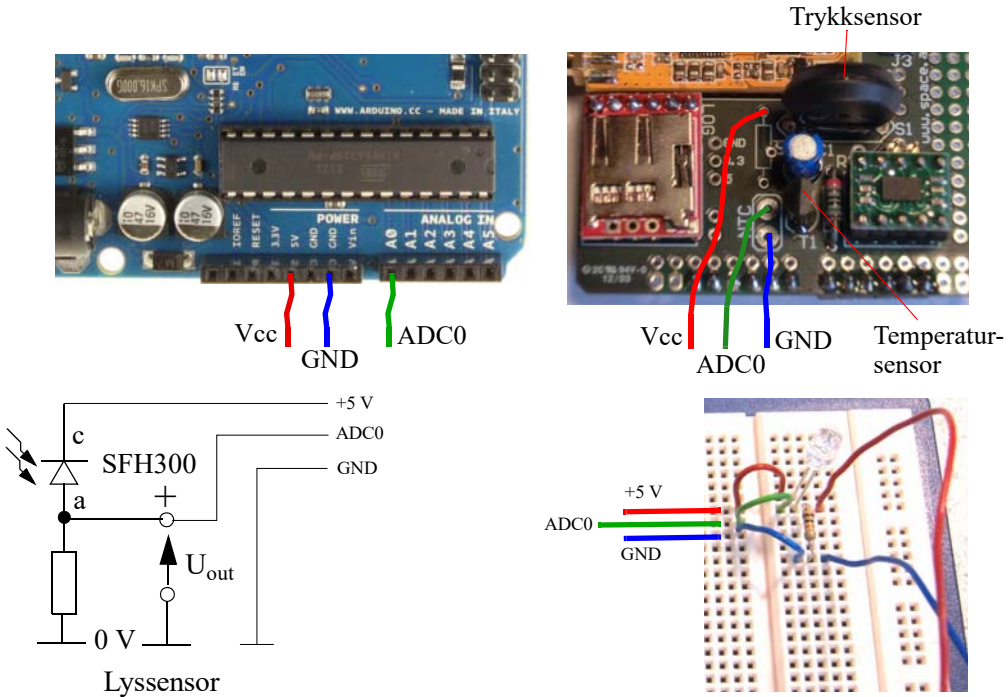




8.3 Oppkobling av ekstra sensor for uttesting

Sensorkortet har en ekstra ledig posisjon der NTC-motstanden er tiltenkt plassert. Her kan alternative sensorer plasseres. Se figuren under. I en eksperimentfase kan det være praktisk å koble seg fra kontaktene på kortet til et koblingsbrett, både for å karakterisere og for å overføre målingen til CanSat'en. Dette er vist på figuren til høyre på figuren under.

Alternativt kan en fjerne *shield* kortet og koble seg direkte til kantkontakten på Arduino UNO-kortet som vist til venstre på figuren under.

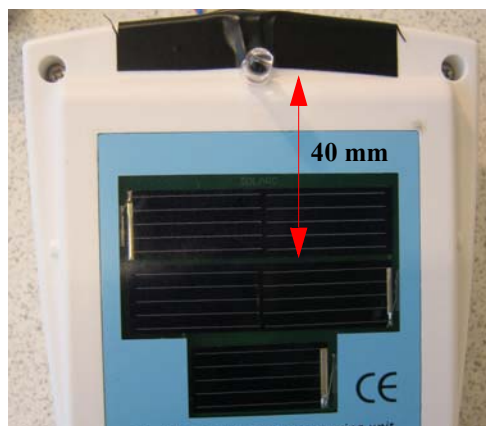




8.4 Karakterisering av lyssensor.

En slik måling er helt avhengig av i hvilken grad en er i stand til å måle lysstyrken på samme sted som der lyssensoren sitter. Dette er ikke alltid så lett dersom lyskilden er relativt liten. Bruk av en større lyskilde kan derfor vær en fordel. I dette tilfelle har montert lyssensoren ca. 40 mm til side for midtpunktet av lysmåleren som er en solcelle. Se bildet til høyre.

Videre benyttet vi en liten bordlampe. Dermed ble usikkerheten relativt stor ved høye verdier av lysstyrken. Små avvik i plassering mellom lyssensor og lysmåler ga relativt store utslag. Målingene ble derfor indikative.



Målerresultater

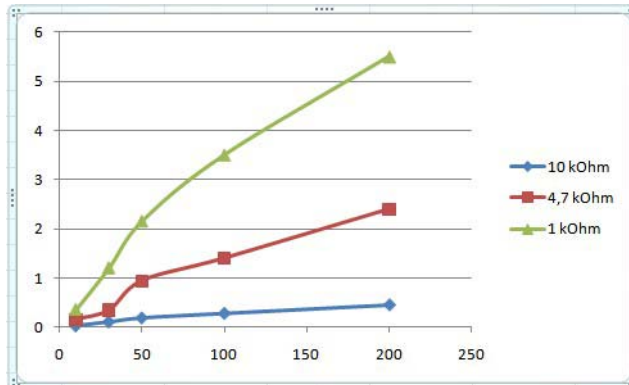
Det ble gjort målinger av to lyssensorer (fototransistorer), hver med tre ulike seriemotstander

Lysstyrke (W/m^2)	100 Ω	1 k Ω	4,7 k Ω	10 k Ω
SFH203				
10		0,032	0,150	0,35
30		0,110	0,340	1,20
50		0,190	0,940	2,15
100		0,280	1,40	3,5
200		0,450	2,40	5,5
SFH300				
10	0,19	2,15	-	-
30	1,07	4,70	-	-
50	2,05	-	-	-
100	2,80	-	-	-
200	3,15	-	-	-





Kurvene er vist i figuren under:





8.5 Nødvendig laboratorieutstyr

<u>Utstyr</u>	<u>Antall</u>	<u>Laboratorium</u>	<u>Ansvarlig</u>
CanSat kit m/tilbehør	10	1, 2	NAROM
Bærbare PCer	10 min.	1, 2	Deltagerne
Verktøy: - Skrutrekkere, tenger, - Loddeutstyr, tinnsugere - Linjal, saks	10 sett	1, 2	NAROM
Multimeter	10	1, 2	NAROM
CanSat grunnpakke	10	1, 2	NAROM
Termometre (glass eller elektroniske)	10	1	NTNU
Luftrykkmåler/Barometer	1	1	NAROM
Hårføner?	5	1	NAROM
Simuleringsprogram Rakettbane	(10)	2	NAROM
Materialer for framstilling av fallskjermer	(10)	2	NAROM
Lysmålere (MacSolar)	5	2	NTNU
Lyskilder	5	2	NTNU
Luftfuktighetsmåler LM8010	1	2	NTNU
Strømforskyninger (Min. 5V)	10	2	NAROM
Sensor HiH4000	10	2	NTNU
Lyssensor SFH300	10	2	NAROM
LDR	10	2	NTNU
Temperatursensor LM35DZ	10	2	NAROM
Operasjonsforsterker	10	2	NAROM
Motstandsett E12/24 rekka	nok	2	NAROM
Dokumentasjon GPS - EM-411 (2 dok.)	22	2	NAROM
Datablader LM35DZ	22	2	NAROM
Datablader SFH300	22	2	NAROM
Datablader HiH4000	22	2	NAROM
Datablader NTC	22	2	NAROM
Datablader MPX4115	22	2	NAROM
Koblingsbrett	10	2	NTNU
Koblingsbrett	?	2	NAROM
Div. kabler (flatkabel (5p) koblingsbrett og CanSat)	12 sett	2	NTNU



Annet, for demo/ekstra

Lyssensorer SFH203	10	2	NTNU
Fuktighetssensor H25K5A	10	2	NTNU
LM35DZ	5	2	NTNU
Operasjonsforsterker CA3130 + div	10	2	NTNU
Diverse sensorkort	sett	2	NTNU



8.6 Generelle retningslinjer for skriving av laboratorierapport

Layout

- Skrifttype: Times New Roman
- Skriftstørrelse: 12
- Linjeavstand 1 (enkel)
- Margstørrelser 2,5 cm topp, bunn og venstre marg, høyre marg 1,5 cm
- Høyst opp på første siden: Navn på student, navn på øvelse/rapport, emne (Romteknologi - (CanSat)), år (2011) og NTNU
- Tekst: Figurtekst under figur, tabelltekst over tabell

Form

Størrelse: 5–6 A4-sider pr. rapport

- Innledning*
- Teoretisk bakgrunnsstoff*
- Utstyr og måleoppstilling (skisse)*
- Gjennomføring og resultater*
- Diskusjon*
- Konklusjon*
- Referanser*

Innhold

Labbrapportene skal være kortfattet, men ha et sammenhengende språk! Dere kan ta utgangspunkt i forelesninger og utdelt materiell tilknyttet laboratoriene, **men skriv med egne ord**. Beskrivelsen under danner grunnlaget for vurdering og karaktersetning av rapporten

- Innledning:** Beskrivelse av problemstillingen. Laboratorieoppgaven/problemstillingen skal være så godt beskrevet at en som er ukjent med oppgaven skal forstå problemstillingen. Den som leser innledningen sammen med konklusjonen skal få en god oversikt over problemstillingen og resultatene av arbeidet.
- Teoretisk bakgrunnsstoff:** Teoretisk bakgrunn og grunnlag for laboratorieoppgaven. Teorien legger grunnlaget for hvilke forventninger man har (hypotesedanning). Alle formler som benyttes, og all teori knyttet klargjøring av begreper og basisteori legges til denne delen.
- Utstyr og måleoppstilling:** Denne delen skal inneholde beskrivelse av metoder og utstyr som ble brukt, og skisser av måleoppsettet.
- Gjennomføring og resultater:** Beskrivelse av resultatene, gjerne med grafer, figurer og/eller tabeller. Nevn spesielle ting knyttet til gjennomføringen. **Ikke** inkluder teoretisk bakgrunnsstoff her, referer til punkt 2.



5. **Diskusjon:** Forklaring av hva resultatene kan bety, dvs. tolking av resultat sett i forhold til forventninger/hypoteser satt opp på bakgrunn av teorien. Diskuter spesielt det som er uventet.
6. **Konklusjon:** Gi en kort oppsummering av hovedresultatene og diskusjonen. Konklusjonen skal ikke inneholde nye opplysninger, men kun referere hva som er kommet fram tidligere i rapporten. Sammen med innledningen skal konklusjonen gi en god oversikt over problemstilling og resultater.
7. **Referanser:** Faktaopplysninger skal refereres med mindre det er allmenn kjente ting. Referansene skal ha med forfatter, tittel, årstall for utgivelse og forlag, ev. tidsskift og volum, ev. sidetall.

Dersom resultatene uteble eller eksperimentet ble mislykket, beskriv så godt dere kan det dere observerer og drøft årsakene til det resultatet dere fikk. Forsøk også å dra lærdom av det som skjedde. Et “mislykket” eller et uventet resultat er også et resultat og forteller en historie.

NB! Hver laboratorierapport får bestått/ikke bestått. Det vil gis korte tilbakemeldinger på innleveringene. En kan søke veiledning under arbeidet med oppgavene. Når de er levert, er det også den samme vurdering som gjelder for vurdering i mappeinnleveringen ved eksamen dersom rapporten blir brukt i den endelige vurderingen. For å få godkjent eksamen må alle rapporter være innlevert og godkjent.





9 Referanser

- [1] Torstein Wang, *Sensor Circuit for CanSat use*, NAROM manual 2011
- [2] T. Wang, S.V. Mathisen, *The CanSat book*, NAROM manual 8. juni 2011
- [3] Gunnar Stette, *Romteknologi - Del av faget Teknologi og Forskningslære (CanSat)*, NTNU juli 2011.
- [4] Datablad ATmega168A
http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf
- [5] Datablad LM35
<http://www.national.com/ds/LM/LM35.pdf>
- [6] Datablad SFH300
https://www1.elfa.se/data1/wwwroot/assets/datasheets/ebSFH300_630448_dat_en_de.pdf
- [7] Datablad MPX4115A
<http://www.datasheetcatalog.org/datasheet/motorola/MPX4115A.pdf>
- [8] Datablad C100 (CO₂) – Ambient, data sheet
COZIR Ultra Low Power Carbon Dioxide Sensor
- [9] Datablad C100 (CO₂)
COZIR Voltage Output Option
- [10] Datablad C100 (CO₂)
Software User's Guide
- [11] **Hygrometer:**
a) <http://zone.ni.com/devzone/devzone.nsf/webcategories/7A2D35CB9262326E8625684A004EEF4D>
b) *H25K5A*
<http://www.sensorelement.com/humidity/H25K5A%20spec.pdf>
c) R. Wagiran, W. S. Wan Zaki, S. B. Mohd Noor, A. H. Shaari and I. Ahmad
Characterization of screen printed BaTiO₃ thick film humidity sensor,
International Journal of Engineering and Technology, Vol. 2, No. 1, 2005, pp. 22-26
<http://eng.upm.edu.my/~ijet/04-v2n1.pdf>
d) *HiH4000*
https://www.elfaelektronikk.no/elfa3~no_no/elfa/init.do?item=73-056-75&toc=0
- [12] **Fototransistor:**
a) <http://www.elfa.se/pdf/75/07520323.pdf> (datablad PT204-6C)
- [13] **Magnetometre:**
a) Aschehoug og Gyldendals store norske leksikon 1988
b) <http://storenorskeleksikon.no>
c) http://www.afl.hitos.no/mfysikk/atm/polatm2.htm#tth_sEc2E
d) <http://sci.esa.int/science-e/www/object/index.cfm?fobjectid=33964&fbody->



longid=1443

e) E. M. Purcell, *Electricity and Magnetism*, McGraw-Hill Book Company, 1963.

f) Steven A. Macintyre, *Magnetic Field Measurements*, CRC Press LLC, 1999

g) Caruso M J m.fl., *A new perspective on Magnetic Field Sensing*

<http://www.sensormag.com/articles/1298/mag1298/main.shtml>

i) André Balogh, *Space Instrumentation*, <http://www.sp.ic.ac.uk/~balogh/spinstr.htm>

[14] **Ozon sensorer:**

a) http://www.ozonesolutions.com/HMOS_vs_Electrochemical_Ozone_Sensors.html

b) <http://www.delphian.com/sensor-tech.htm#ELECTROCHEMICAL%20SENSORS>

c) Futurlec: http://www.futurlec.com/Gas_Sensors.shtml

d) http://www.ozonesolutions.com/HMOS_vs_Electrochemical_Ozone_Sensors.html

[15] **Røykvarslere**

a) <http://www.kidde.com.au/How+Smoke+Alarms+Work.shtml>

[16] **PCME partikkelprober:**

a) <http://www.groupinstrumentation.com.au/media/7245/electrodynamic%20technology%20for%20particulate%20monitoring.pdf>

b) http://www.pcme.co.uk/product_pdf/DX800%20series.pdf

[17] **NDIR CO2-sensor:**

a) <http://cdn.shopify.com/s/files/1/0019/5952/files/C100-Datasheet.pdf?1286094077>

b) http://en.wikipedia.org/wiki/Nondispersive_infrared_sensor

c) http://o.b5z.net/i/u/10004484/i/brochures/gas%20monitoring/CO2_Carbon_Dioxide_Sensor_ETS.pdf



Vedlegg A Eksempler på kode

A.1 Oppretting og skriving av data til Openlog

Koden viser hvordan en kan opprette en ny fil med et selvvalgt navn i Openlog for deretter å skrive data fortløpende inn i filen. For å unngå at samme file forsøkes opprettet ved hver oppstart (power on), er det benyttet en tilfeldighetsfunksjon (*random*) som ved hver oppstart tilordner filnavnet et tilfeldig nummer.

```
/*  
6-1-2011  
SparkFun Electronics 2011  
Nathan Seidle
```

Controlling OpenLog command line from an Arduino

Connect the following OpenLog to Arduino:

```
TXO of OpenLog to RX of the Arduino  
RXI to TX  
GRN to 2  
VCC to 5V  
GND to GND
```

NOTE: When uploading this example code you must temporarily disconnect TX and RX while uploading the new code to the Arduino. Otherwise you will get a "avrduide: stk500_getsync(): not in sync" error.

This example code assumes the OpenLog is set to operate at 9600bps in NewLog mode, meaning OpenLog should power up and output '12<'. This code then sends the three escape characters and then sends the commands to create a new random file called nate####.txt where #### is a random number from 0 to 999.

This code assume OpenLog is in the default state of 9600bps with ASCII-26 as the escape character.

Be careful when sending commands to OpenLog. println() sends extra newline characters that cause problems with the command parser. The new v2.51 ignores \n commands so it should be easier to talk to on the command prompt level. This example code works with all OpenLog v2 and higher.

```
*/  
char buff[50];  
int fileNumber;  
  
int statLED = 13;  
int resetOpenLog = 2;
```



```
void setup() {
  pinMode(statLED, OUTPUT);
  pinMode(resetOpenLog, OUTPUT);

  randomSeed(analogRead(0));
  Serial.begin(9600);

  //Reset OpenLog
  digitalWrite(resetOpenLog, LOW);
  delay(100);
  digitalWrite(resetOpenLog, HIGH);

  //Wait for OpenLog to respond with '<' to indicate it is alive and recording to a file
  while(1) {
    if(Serial.available())
      if(Serial.read() == '<') break;
  }

  //Send three control z to enter OpenLog command mode
  //This is how Arduino v0022 used to do it. Doesn't work with v1.0
  //Serial.print(byte(26));
  //Serial.print(byte(26));
  //Serial.print(byte(26));

  //Works with Arduino v1.0
  Serial.write(26);
  Serial.write(26);
  Serial.write(26);

  //Wait for OpenLog to respond with '>' to indicate we are in command mode
  while(1) {
    if(Serial.available())
      if(Serial.read() == '>') break;
  }

  fileNumber = random(999); //Select a random file #, 0 to 999

  //Send new (random from 0 to 999) file name

  //Old way
  sprintf(buff, "new nate%03d.txt\r", fileNumber);
```



```
Serial.print(buff); //\r in string + regular print works with older v2.5 Openlogs
```

```
//New way
//sprintf(buff, "new nate%03d.txt", fileNumber);
//Serial.println(buff); //regular println works with v2.51 and above
```

```
//Wait for OpenLog to return to waiting for a command
while(1) {
  if(Serial.available())
    if(Serial.read() == '>') break;
}
sprintf(buff, "append nate%03d.txt\r", fileNumber);
Serial.print(buff);
```

```
//Wait for OpenLog to indicate file is open and ready for writing
while(1) {
  if(Serial.available())
    if(Serial.read() == '<') break;
}
}
```

```
void loop() {
  Serial.println("Yay!");
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

A.2 Konfigureringsprogram APC220

Skrevet av Thomas Gansmoe og tilrette lagt av Jens F. Dalsgaard Nielsen (Rev. 06.11.12)

```
#include <SoftwareSerial.h>
```

```
/* plug in apc220 at pin 8-GND with apc pcb pointing away from Arduino
 * Jens(AAU), Thomas(ARS)
 */
```

```
const int vrs = 102;
const int rxPin = 10; //rx pin is tx pin on apc
const int txPin = 11; //tx pin is rx pin on apc
const int setPin = 8; //set pin
```



```
const int enPin = 12; // no use
const int fiveV = 13; // 5V to the APC220

int echo_mode = 0;

#define TIMEOUT 10000L
#define PING_INTERVAL 2000L

unsigned long timeout,lastPing;

#define PARM_LGT 10
#define CMD_BUF_LGT 50

char var1[PARM_LGT], var2[PARM_LGT], var3[PARM_LGT], var4[PARM_LGT], var5[PARM_LGT],
var6[PARM_LGT];

SoftwareSerial apc220(rxPin, txPin); // Crt softserial port and bind tx/rx to appropriate PINS

void drawRadio()
{

Serial.println("Attach radio directly to Arduino on pin 8 to gnd");
Serial.println(F(" "));
Serial.println(F("-----"));
Serial.println(F("  A  aref"));
Serial.println(F("  R  GND-----"));
Serial.println(F("  D  13      |=antenna connector ->"));
Serial.println(F("IOREF U  12      |"));
Serial.println(F("RESET I  11      |"));
Serial.println(F("3.3V N  10      |"));
Serial.println(F("5V O  09      |"));
Serial.println(F("GND  08-----"));
Serial.println(F("GND P  |"));
Serial.println(F("Vin C  07"));
Serial.println(F("| B ~ 06"));
Serial.println(F("A0 ~ 05"));
Serial.println(F("A1  04"));
Serial.println(F("A2 ~ 03"));
Serial.println(F("A3  02"));
```




```
Serial.println(F("A4 TX 01"));
Serial.println(F("A5 RX 00"));
Serial.println(F("-----"));
}

void set_para(char hz[], char rf_rate[], char pwr[], char uart_rate[], char sc[]) {
// sanity chk
if (strlen(hz) != 6) {
  Serial.println("Freq parm not 6 digits... - legal is 418000 - 455000");
  return;
}
if (strlen(rf_rate) != 1) {
  Serial.println("RF parm is not 1 digit: legal values is 1/2/3/4");
  return;
}
if (strlen(pwr) != 1) {
  Serial.println("Power parm is not 1 digit: legal values is 1..9");
  return;
}

if (strlen(uart_rate) != 1) {
  Serial.println("Uart baudrate parm is not 1 digit: legal values is 0..6");
  return;
}
if (strlen(sc) != 1) {
  Serial.println("Parity parm is not 1 digit: legal values is 0/1/2");
  return;
}

digitalWrite(setPin, LOW); // set radio in config mode
delay(50);
apc220.print("WR");
apc220.print(" ");

apc220.print(hz);
apc220.print(" ");
```



```
apc220.print(rf_rate);
apc220.print(" ");

apc220.print(pwr);
apc220.print(" ");

apc220.print(uart_rate);
apc220.print(" ");

apc220.print(sc);

apc220.write(0x0D);
apc220.write(0x0A);
delay(10);

// read feedback from radio

while (apc220.available()) {
  Serial.write(apc220.read());
}

digitalWrite(setPin, HIGH); // set radio back in normal rx/tx mode
}

void get_para(void) {
  int i = 0;
  char buff[CMD_BUF_LGT];

  digitalWrite(setPin, LOW); // set radio in config mode

  delay(10); // stabilize please

  apc220.println("RD"); // ask for data
  delay(10);

  while (apc220.available()) {
    Serial.write(apc220.read());
  }
}
```



```
digitalWrite(setPin, HIGH); // set radio back in normal tx/rx mode
}

void resetPing(void) {
  echo_mode = 0;
  timeout = millis();
}

void helpMenu(void) {
  Serial.println("");
  Serial.println(F("commands:"));
  Serial.println(F(" r : Read apc220 radio config"));
  Serial.println(F(" e : go into echo mode: receive char, add 1 and return"));
  Serial.println(F(" n : no more echo - back to normal"));
  Serial.println(F(" p: print how to attach radio to Arduino"));
  Serial.println(F(" w : Write apc radio config ..."));
  Serial.println(F(" 'W' FFFFFFF R P B C - number of letters indicates precise number of digits" ));
  Serial.println(F(" FFFFFFF: frequency: 434000 (434 MHz) range 418000-455000 "));
  Serial.println(F(" R: Rf data rate - 1/2/3/4 equals 2400/4800/9600/19200bps"));
  Serial.println(F(" P: Radio output power - 0 .. 9 9 equals 13dBm(20mW)."));
  Serial.println(F(" B: UART baudrate - 0/1/2/3/4/5/6 equals 1200/2400/4800/9600/19200/38400/57600bps"));
  Serial.println(F(" C: Byte Chek Parity - 0/1/2 equals NoCheck(8N1)/EvenParity(8E1)/OddParity(8O1)"));
  Serial.println(F(""));
  Serial.println(F("Write example: w 434000 3 9 3 0 is..."));
  Serial.println(F(" 434,000 MHz 9600 baud in air, 20mW, 9600baud on UART, No Parity(8N1)"));
  Serial.println(F("After 10 seconds with no keyboard input we will emit a char every two second"));
}

void get_cmd(void) {

  if((echo_mode == 0) && (TIMEOUT < (millis() - timeout))) { // time and state for beacon mode ?
    apc220.write(!);
    Serial.println("going into ping mode - emit a char every two second on apc220 !");
    lastPing = millis();
    echo_mode = 2;
    goto xxx;
  }
}
```



```
if (echo_mode == 2) { // beacon mode
  if (PING_INTERVAL < (millis() - lastPing)) {
    apc220.write('x');
    Serial.println(lastPing/1000);
    lastPing = millis();
  }
  goto xxx;
}

if (echo_mode == 1) { // echo mode
  char c;

  if (apc220.available()) {
    c = apc220.read();
    Serial.write(c);
    apc220.write(c+1);
  }
  goto xxx;
}

xxx:

if (Serial.available()) {
  int i=0;
  char buff[CMD_BUF_LGT];

  delay(100);

  while (Serial.available() && (i < CMD_BUF_LGT-1)) {
    buff[i++] = Serial.read();
  }
  buff[i] = '\0';
  Serial.println(buff); //echo input

  resetPing();

  var1[0] = 0x00; // reset
```



```
    sscanf(buff, "%s %s %s %s %s %s", var1, var2, var3, var4, var5, var6);
    switch (var1[0]) { // one letter commands :-)
    case 'r': {
        get_para();
        break;
    }
    case 'w': {
        set_para(var2, var3, var4, var5, var6);
        break;
    }
    case 'e' :{
        echo_mode = 1;
        break;
    }
    case 'n' :{
        echo_mode = 0;
        break;
    }
    case 'p':{
        drawRadio();
        delay(2000);
        break;
    }
    default:
        echo_mode = 0;
        helpMenu();
    }
}
```

```
void setupSoftAPC(void)
{
    pinMode(setPin,OUTPUT);
    digitalWrite(setPin,HIGH);
    pinMode(fiveV,OUTPUT); // 5V
    digitalWrite(fiveV,HIGH); // turn on 5V
    delay(50);
    pinMode(enPin,OUTPUT); // ENABLE
}
```



```
digitalWrite(enPin,HIGH); //
delay(100);
apc220.begin(9600);
}

void setup() {
  Serial.begin(9600);
  Serial.print("APC version: ");
  Serial.println(vrs);
  Serial.println(__DATE__);
  setupSoftAPC();
  timeout = millis(); // just to start
}

void loop() {
  get_cmd();
}
```

A.3 Program for styring av terminering (BallongSquid.ino)

Skrevet av Thomas Gansmoe, NAROM)

```
int pwmInn = 3;
int ledPin = 13;
int outPin = 4;
int outTestPin = 7;
int pulsewidth;
long int timer1, blinkTime=0;
boolean trigger = LOW, flag1=LOW;
void setup()
{
  Serial.begin(9600); // Starts serial port
  pinMode(ledPin, OUTPUT);
  pinMode(pwmInn, INPUT); // sets the pin as input
  pinMode(outPin, OUTPUT);
  pinMode(outTestPin, OUTPUT);
  digitalWrite(outPin,LOW);
  digitalWrite(outTestPin,LOW);
}
void loop()
{
```



```
trigger=LOW;
flag1=LOW;
pulsewidth=pulseIn(pwmInn,HIGH);
while (pulsewidth>1700&&pulsewidth<2000){
  if (flag1==LOW){
    flag1=HIGH;
    timer1=millis();
  }
  if (millis()-timer1>5000){
    trigger=HIGH;
    digitalWrite(outPin,trigger);
  }
  Serial.print(pulsewidth);
  Serial.print(" - Trigger: ");
  Serial.print(trigger);
  Serial.print(" - Flag: ");
  Serial.print(flag1);
  Serial.print(" - Timer: ");
  Serial.print(millis()-timer1);
  Serial.println(" --InLoop");
  if ((millis()-blinkTime)>100){
    if (digitalRead(ledPin)==LOW){digitalWrite(ledPin,HIGH);}
    else {digitalWrite(ledPin,LOW);}
    blinkTime=millis();
  }
  if (trigger==HIGH) digitalWrite(ledPin,LOW);
  pulsewidth=pulseIn(pwmInn,HIGH);
}
Serial.print(pulsewidth);
Serial.print(" - Trigger: ");
Serial.print(trigger);
Serial.print(" - Flag: ");
Serial.print(flag1);
Serial.print(" - Timer: ");
Serial.print(millis()-blinkTime);
Serial.print(" - ");
Serial.print(digitalRead(7));
Serial.print(" - ");
```



```
Serial.println(millis());

if ((millis()-blinkTime)>500){
  if (digitalRead(ledPin)==LOW){digitalWrite(ledPin,HIGH);}
  else {digitalWrite(ledPin,LOW);}
  blinkTime=millis();
}
digitalWrite(outPin,trigger);
if (millis(>60000) digitalWrite(outTestPin,HIGH);
}
```

A.4 Kode for lesing fra GPS-modul

A.4.1 Enklest mulig kode:

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>

/* This sample code demonstrates the simplified use of a TinyGPS object.
   It requires the use of SoftwareSerial, and assumes that you have a
   9600-baud serial GPS device hooked up on pins 4(rx) and 3(tx).
*/

TinyGPS gps; // Definerer en TinyGPS type kalt gps
SoftwareSerial ss(4,3); // Definerer en SoftwareSerial type kalt ss
static void smartdelay(unsigned long ms);

void setup()
{
  Serial.begin(115200); // Sett datahastighet til monitor
  ss.begin(9600); // Sett datahastighet til GPS-kontrollkort
}

void loop()
{
  float flat, flon; // Definer variable for lengde- og breddegrad
  int year; // Definer variable for år, dato og tid
```




```
byte month, day, hour, minute, second, hundredths;
unsigned long age; // Definer variabel for age

gps.f_get_position(&flat, &flon, &age); // Henter breddegrader og lengdegrader
if (flon < 90.0 && flat < 180.0) // Hindrer skriving før GPS er låst
{
  Serial.print(flon,6); // Skriver ut lengdegrader i grader desimalt
  Serial.print(","); // Sett inn komma som skilletegn
  Serial.print(flat,6); // Skriver ut breddegrader i grader desimalt
  Serial.print(","); // Sett inn komma som skilletegn
  Serial.print(gps.f_altitude(), 2); // Skriver ut høyde over havet i meter
  Serial.println();
}
smartdelay(1000);
}

static void smartdelay(unsigned long ms)
{
  unsigned long start = millis(); // Hent antall millisekunder siden reset av Arduino-kortet
  do // Testen gjøres etter loopen er kjørt
  {
    while (ss.available())gps.encode(ss.read());
  }
  while (millis() - start < ms); // Testen gjøres før loopen er kjørt
}
```

A.4.2 Kode med fyldige data og feilmeldinger

```
#include <SoftwareSerial.h>
```

```
#include <TinyGPS.h>
```

```
/* This sample code demonstrates the normal use of a TinyGPS object.
```



It requires the use of SoftwareSerial, and assumes that you have a 4800-baud serial GPS device hooked up on pins 4(rx) and 3(tx).

```
*/
```

```
TinyGPS gps;
```

```
SoftwareSerial ss(4, 3);
```

```
static void smartdelay(unsigned long ms);
```

```
static void print_float(float val, float invalid, int len, int prec);
```

```
static void print_int(unsigned long val, unsigned long invalid, int len);
```

```
static void print_date(TinyGPS &gps);
```

```
static void print_str(const char *str, int len);
```

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

```
  Serial.print("Testing TinyGPS library v. "); Serial.println(TinyGPS::library_version());
```

```
  Serial.println("by Mikal Hart");
```

```
  Serial.println();
```

```
  Serial.println("Sats HDOP Latitude Longitude Fix Date    Time    Date Alt   Course Speed  
Card Distance Course Card Chars Sentences Checksum");
```

```
  Serial.println("      (deg) (deg)   Age          Age (m)  --- from GPS ---- -- to  
Trondheim --- RX  RX    Fail");
```

```
  Serial.println("-----  
-----");
```

```
  ss.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```



```
float flat, flon;
unsigned long age, date, time, chars = 0;
unsigned short sentences = 0, failed = 0;
static const double LONDON_LAT = 51.508131, LONDON_LON = -0.128002;
static const double TRONDHEIM_LAT = 63.436668, TRONDHEIM_LON = 10.383333;

print_int(gps.satellites(), TinyGPS::GPS_INVALID_SATELLITES, 5);
print_int(gps.hdop(), TinyGPS::GPS_INVALID_HDOP, 5);
gps.f_get_position(&flat, &flon, &age);
print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 10, 6);
print_float(flou, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6);
print_int(age, TinyGPS::GPS_INVALID_AGE, 5);
print_date(gps);
print_float(gps.f_altitude(), TinyGPS::GPS_INVALID_F_ALTITUDE, 7, 2);
print_float(gps.f_course(), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);
print_float(gps.f_speed_kmph(), TinyGPS::GPS_INVALID_F_SPEED, 6, 2);
print_str(gps.f_course() == TinyGPS::GPS_INVALID_F_ANGLE ? "*** " : TinyGPS::cardi-
nal(gps.f_course()), 6);
print_int(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0xFFFFFFFF : (unsigned
long)TinyGPS::distance_between(flat, flon, LONDON_LAT, LONDON_LON) / 1000,
0xFFFFFFFF, 9);
print_float(flat == TinyGPS::GPS_INVALID_F_ANGLE ? TinyGPS::GPS_INVALID_F_AN-
GLE : TinyGPS::course_to(flat, flon, LONDON_LAT, LONDON_LON),
TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);
print_str(flat == TinyGPS::GPS_INVALID_F_ANGLE ? "*** " : TinyGPS::cardi-
nal(TinyGPS::course_to(flat, flon, TRONDHEIM_LAT, TRONDHEIM_LON)), 6);

gps.stats(&chars, &sentences, &failed);
print_int(chars, 0xFFFFFFFF, 6);
print_int(sentences, 0xFFFFFFFF, 10);
print_int(failed, 0xFFFFFFFF, 9);
Serial.println();
```



```
smartdelay(1000);
}

static void smartdelay(unsigned long ms)
{
    unsigned long start = millis();
    do
    {
        while (ss.available())
            gps.encode(ss.read());
    } while (millis() - start < ms);
}

static void print_float(float val, float invalid, int len, int prec)
{
    if (val == invalid)
    {
        while (len-- > 1)
            Serial.print('*');
        Serial.print(' ');
    }
    else
    {
        Serial.print(val, prec);
        int vi = abs((int)val);
        int flen = prec + (val < 0.0 ? 2 : 1); // . and -
        flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
        for (int i=flen; i<len; ++i)
            Serial.print(' ');
    }
    smartdelay(0);
}
```



```
static void print_int(unsigned long val, unsigned long invalid, int len)
{
    char sz[32];
    if (val == invalid)
        strcpy(sz, "*****");
    else
        sprintf(sz, "%ld", val);
    sz[len] = 0;
    for (int i=strlen(sz); i<len; ++i)
        sz[i] = ' ';
    if (len > 0)
        sz[len-1] = ' ';
    Serial.print(sz);
    smartdelay(0);
}
```

```
static void print_date(TinyGPS &gps)
{
    int year;
    byte month, day, hour, minute, second, hundredths;
    unsigned long age;
    gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);
    if (age == TinyGPS::GPS_INVALID_AGE)
        Serial.print("***** ");
    else
    {
        char sz[32];
        sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ",
            month, day, year, hour, minute, second);
        Serial.print(sz);
    }
}
```



```
print_int(age, TinyGPS::GPS_INVALID_AGE, 5);
smartdelay(0);
}
```

```
static void print_str(const char *str, int len)
{
    int slen = strlen(str);
    for (int i=0; i<len; ++i)
        Serial.print(i<slen ? str[i] : ' ');
    smartdelay(0);
}
```



Vedlegg B Regler for Forankrede ballonger

Kapittel IX. Forankrede ballonger

§ 9-1. Kapittelets virkeområde

Dette kapittel gjelder forankrede ballonger når de på grunn av størrelse, brennbarhet, høyde over bakken mv. kan medføre fare eller ulempe for luftfarten.

§ 9-2. Krav i forbindelse med oppsending av forankrede ballonger

1. Forankrede ballonger må bare sendes opp, og holdes oppe, når bakkesikten er minst 8 km. De tillates heller ikke holdt oppe når de er nærmere skyer enn 300 m vertikalt og/eller 1.500 m horisontalt.
2. Oppsending av forankrede ballonger i kontrollert luftrom krever tillatelse fra Luftfartstilsynet. Søknad om tillatelse må være Luftfartstilsynet i hende minst 14 dager før oppsendingen skal finne sted.
3. Utenfor kontrollert luftrom gjelder bestemmelsen i annet ledd tilsvarende når oppsendingen skjer nærmere en flyplass enn 10 km eller skjer innenfor et av Luftforsvarets lavflygingsområder.
4. Hvis avstanden fra en flyplass er 10 km eller mer, kan oppsendingen til høyder over 45 m bare finne sted såfremt melding er gitt minst 14 dager forut til nærmeste enhet av lufttrafikk tjenesten eller til NOTAM-kontoret.
5. Melding etter fjerde ledd skal inneholde:
 - a) navn, adresse og eventuelt telefonnummer til den som har ansvaret for oppsendingen,
 - b) sted/posisjon for forankringen,
 - c) ballongens maksimale høyde,
 - d) dato, tid og varighet for oppsendingen,
 - e) opplysninger om hvordan ballongen med forankringsutstyr er varselmerket.
6. Forankrede ballonger skal være utstyrt med en anordning for punktering av ballongen. Anordningen må kunne utløses automatisk dersom ballongen kommer i drift. Hvis eier eller bruker av en forankret ballong blir kjent med at ballongen er kommet i drift uten at punktering har skjedd, plikter han straks å underrette nærmeste enhet av lufttrafikk tjenesten eller NOTAM-kontoret.



Vedlegg C Leverandører

C.1 Oversikt over komponenter til CanSat kit'et

Prisene er rene prisantydninger uten frakt, men med MVA kjøpt i små antall.

Komponent	Type	Forhandler	Pris
Plattform	Arduino UNO R3	www.KultogBillig.no	139,00
http://kultogbillig.no/Arduino-Uno-R3-Rev3-kompatibelt-kort?search=Arduino UNO			
Trykksensor	MPX4115a	Farnell	104,96,-
http://no.farnell.com/freescale-semiconductor/mpx4115a/sensor-abs-press-16-7-psi-867-6/dp/1457150			
Barometer	BMP180/Bosch	KultOgBillig	69,00
http://kultogbillig.no/BMP180-BOSCH-Barometrisk-Digital-Digital-Luft-Trykk-Sensor-Modul-For-Arduino?keyword=BMP180			
Barometer, Akselerometer, Gyroskop, Magnetisk sensor	GY80	dealExtreme	101,99
http://www.dx.com/p/gy-80-bmp085-9-axis-magnetic-acceleration-gyroscope-module-for-arduino-145912			
Temperatursensor (halvleder)	LM35DZ	ELFA	13,18
https://www.elfaelektronikk.no/elfa3~no_no/elfa/init.do?item=73-276-79&toc=19595			
Temperatursensor (NTC)	NTCLE100E3103JB0	Farnell	6,33
http://no.farnell.com/jsp/search/productdetail.jsp?CMP=i-ddd7-00001003&sku=1187031			
Akselerometer	MMA7361L	Pololu (\$11,95)	80,00
http://www.pololu.com/catalog/product/1246			
Radio	APC220	Drobotics (£33,00)	330,00
http://www.droboticonline.com/index.phpapc220-radion-communication-module.html			
Stiflist (kobling) 10 stk. x 40	Break Away Headers	KultOgBillig	29,-
http://kultogbillig.no/Elektronikk/elektronikk-kabler-og-kontakter/40-pins-2-54mm-Stiflist-Header			
Datalogger	OpenLog DEV-09530	Sparkfun	\$24,95
https://www.sparkfun.com/products/9530			
Datalogger	OpenLog DEV-09530	Hobby Electronics UK	£17,26
http://www.hobbytronics.co.uk/openlog-datalogger			



Komponent	Type	Forhandler	Pris
Jumpere (6 stk)	Amp Modu w/handle 0,8	Farnell	15,00
http://no.farnell.com/te-connectivity-amp/0-0881545-2/jumper-amp-modu-with-handle-0-8gold/dp/4218176			
Female Headers 7p		ELFA	9,48
https://www.elfaelektronikk.no/elfa3~no_no/elfa/init.do?item=43-783-03&toc=19754			
Female Headers 2p		ELFA	5,56
https://www.elfaelektronikk.no/elfa3~no_no/elfa/init.do?item=43-783-03&toc=19754			
Kondensator	El.lytt 1 uF	Farnell	1,00
http://no.farnell.com/vishay-bc-components/mal203858108e3/capacitor-radial-63v-1uf/dp/1834164			
Motstand	75 Ohm	Farnell	0,50
Batterikontakt	9V Snap connector	ELFA	11,23
https://www.elfaelektronikk.no/elfa3~no_no/elfa/init.do?item=42-043-01&toc=19108&q=batterih%C3%A5llare+9v			
Strømplugg	2,1 mm	Robonor	8,75
http://www.robonor.no/mage/default/2-1mm-plugg-med-kabelstotte.html			
Antenne (laget av koaksialkabel m/plugg) Må kjøpe to (kabel med to kontakter)	AMPHENOL CONNEX - 135101-01-24.00 - COAXIAL CABLE ASSY, RG-316, 24IN	Farnell	92,84 pr. stk
Antenne (laget av koaksialkabel m/plugg) Må kjøpe to (kabel med to kontakter)	SMA LEAD ST/ST RG316 0.5M	RS-Online	124,24 pr. stk
http://no.rs-online.com/web/p/coaxial-cable-assemblies/4097828/			
http://no.farnell.com/jsp/search/productdetail.jsp?CMP=i-ddd7-00001003&sku=1776766			
Fallskjerm	18" nylon fallskjerm	Apogee component.	43,00
http://www.apogeerockets.com/Building_Supplies/Parachutes_Recovery_Equipment/Parachutes/Low_Power/18_Nylon_Parachute			
USB2.0 Cable A/B		ELFA	30,25
https://www.elfaelektronikk.no/elfa3~no_no/elfa/init.do?init=2&shop=ELFA_NO-NO&item=25-112-42			
MicroSD Card	2GB	ELFA	28,50
https://www.elfaelektronikk.no/elfa3~no_no/elfa/init.do?item=73-019-80&toc=0&q=sandisk			



Komponent	Type	Forhandler	Pris
SanDisk micro SDHC	8GB	KultOgBillig	89,00
http://kultogbillig.no/Elektronikk/Minnekort			
Kretskort		Aalborg universitet	50,-
Ta kontakt med: jdn@es.aau.dk			
Totalt:			

C.2 Leverandører av ulike sensorer og andre komponenter

- **ELFA:** <http://www.elfa.se>

ELFA er et svensk komponentfirma som leverer raskt, men som ikke nødvendigvis er billigst.

Har en meget oversiktlig og fyldig nettbutikk (stk priser ex. MVA pr. aug. 2016).

- NTC-motstand, RH16 (10 kOhm) Pris: Ca. kr. 9,27 ELFA - 160-27-932
- Fototransistor, PT204 Pris: Ca. kr. 6,75 ELFA - 110-44-313
- Luftfuktighetssensor, H25K5A Pris: Ca. kr. 33,20 ELFA - 173-02-665
- Luftfuktighetssensor, HiH4000-004 Pris: Ca. kr. 141,00 ELFA - 173-21-491
- Gass-sensor, HS-129 Pris: Ca. kr. 105,00 ELFA - 173-02-672
- Gass-sensor, HS-131 Pris: Ca. kr. 99,80 ELFA - 173-02-706
- Gass-sensor, HS-133 Pris: Ca. kr. 128,00 ELFA - 173-02-714
- Gass-sensor, HS-134 Pris: Ca. kr. 153,00 ELFA - 173-02-722
- Gass-sensor, HS-135 Pris: Ca. kr. 106,00 ELFA - 173-02-730

- **Futurlec:** <http://www.futurlec.com/index.shtml>

Internasjonalt firma som selger elektronikk og sensorer. Har avdeling i England.

- **RS Norway:** <http://no.rs-online.com/web/home.html>

RS er et internasjonalt firma med en norsk filial. Firmaet er primært et komponentleverandør, men leverer raskt og greit. Har en fyldig nettbutikk.

- FM Radiomottaker RRFQ2-433 Pris: kr. 149,50
- Trykksensor, MPX4115A Pris: kr. 148,26

- **KultOgBillig:** <http://kultogbillig.no>

Dette firmaet har spesialisert seg på det norske hobbymarkedet med vekt på Arduino produkter.

- GPS-modul GY-NEO6MV2 Pris: kr. 198,00 (Aug. 2016)

- **Pololu:** <http://pololu.com/catalog/product/1247>

Pololu Robotics and Electronics er et amerikansk firma

- Akselerometer 3D (MMA 7361LC) Pris: Ca. 11,95 \$ (ut av salg) (1,5/6g)
- Akselerometer 3D (MMA7341LC) Pris: Ca. 9,95 \$ (3/9g) (Aug. 2016)



- **Drobotics:** <http://www.droboticonline.com/index.php/apc220-radion-communication-module.html> Dette er et engelsk firma som bl.a. leverer utstyr til Arduino.
- APC220, kommunikasjonsenhet Pris: Ca. 33£ + MVA
- **Sparkfun Electronics:** <http://www.sparkfun.com/categories/23?page=1>
Dette er et amerikansk firma som leverer alle mulige sensorer, ofte leverte på små kort med digitalt grensesnitt som lett lar seg integrere med mikrokontrollere.
- Miniatyrkamera, TCM8230MD Pris: 10\$
- **Tech supplys:** <http://www.rev-ed.co.uk/picaxe/>
Er et engelsk firma som leverer billige komponenter spesielt rettet mot PICAXE-moduler. Utvalget er relativt beskjedent, men prisene er lave.
- LDR Pris: 0,24£
- **Pratt hobbies:** <http://www.pratthobbies.com/proddetail.asp?prod=CANSAT-1>
Leverer CanSat byggesettet og annet utstyr til rakett oppskyting (priser pr. aug. 2011):
- CanSat kit (gammel versjon) Pris: 259 \$



Vedlegg D Produktdata

D.1 APC220

Wireless RF *transceiver* (Transmitter/Receiver)

1. Dimensions 39 mm x 19 mm x 2.8 mm
2. Receive current $\leq 30\text{mA}$
3. Sleep Current $\leq 5\mu\text{A}$

APC220 wireless RF transceiver (including antenna)

Description:

APC220 module is highly integrated micro-power half-duplex wireless data transmission module, the embedded high-speed and high-performance single-chip radio frequency chip, TTL level serial interface.

Technical Specifications:

1. Frequency range 431MHz to 478MHz (1KHz steps)
2. GFSK modulation (Gaussian Frequency Shift Keying)
3. Channel spacing 200KHz frequency interval
4. Transmission power 20mW (13 dBm) (10 level adjustable)
5. Receiver sensitivity -117dBm @ 1200 bps
6. Air transmission rate of 1200 – 19200 bps
7. Interface rate 1200 – 57600 bps
8. Interface efficacy approach 8E1/8N1/8O1
9. Interface buffer space 512bytes
10. Humidity 10% ~ 90% (no condensation)
11. Operating Temperature -20 °C – 70 °C
12. Power Supply 3.3 - 5.5V ($\pm 50\text{mV}$ ripple)
13. Emission current $\leq 35\text{mA}$ @ 10mW
14. Receive current $\leq 30\text{mA}$
15. Sleep Current $\leq 5\mu\text{A}$
16. Transmission distance 1000 m distance (open to viewing distance)
17. Dimensions 39 mm x 19 mm x 2.8 mm



Distribution list:

Shipping included APC220 a antenna pair, and the USB-TO-TTL converter (Price approximately \$40 or £30.

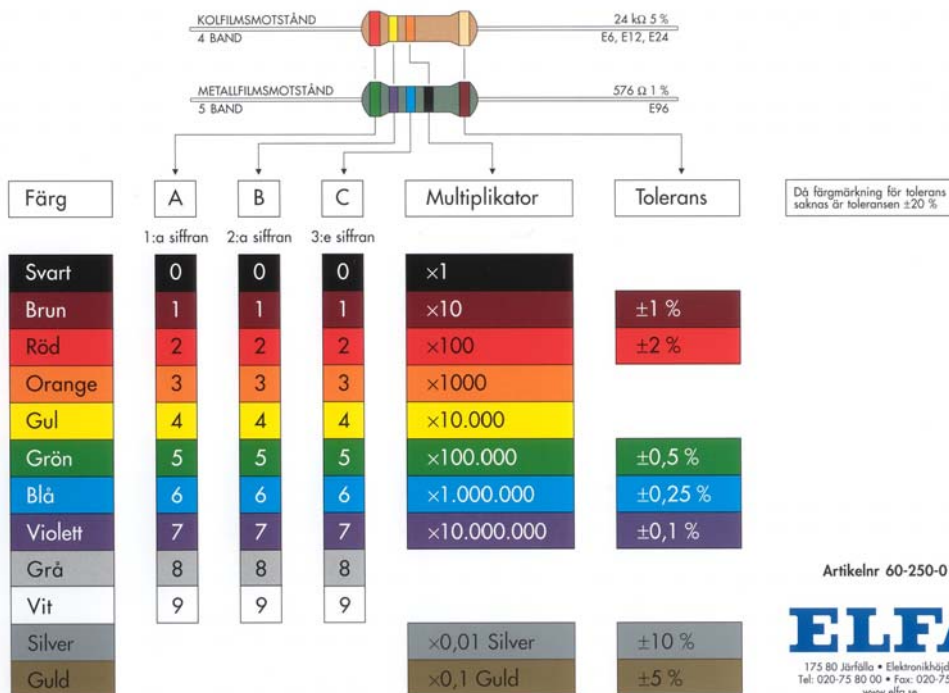




Vedlegg E Fargekoding av motstander

Motstander er ofte merket med fargede ringer som angir resistansen eller verdien til motstanden. Antallet ringer er avhengig av hvor nøyaktig resistansen er. Nøyaktigheten angis som en toleranse i %. Dvs. at verdien skal garantert være innenfor et angitt prosentvist avvik. Ofte angis toleransen ved hjelp av en gull- eller sølvfarget ring. En sølvfarget ring antyder at toleransen er innen +/-10%, en gullfarget innen +/-5%. Også andre farger kan brukes for mer nøyaktige motstander med mindre toleranse. Når vi leser verdien til motstanden skal gull- eller sølvringen være til høyre. De resterende ringene leses deretter fra venstre mot høyre.

FÄRGGODSSCHEMA FÖR MOTSTÅND



Plansjen over viser sammenhengen mellom farge og tall.



Vedlegg F Læreplaner

F.1 Teknologi og Forskningslære 1

Fargekodene forsøker å antyde hva som **lett kan** knyttes til CanSat og romteknologi (rødt/kursiv), hva som er litt på siden (blått) og det som vanskelig lar seg oppfylle innen prosjektet (sort).

Den unge ingeniøren - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne:

- *planlegge og bygge en konstruksjon som er fast eller bevegelig, og som har en definert funksjon*
- bruke tredimensjonale tegninger eller skisser i utvikling av konstruksjoner
- bruke forskjellige materialer og former for sammenføyninger og begrunne valg av materialer og byggemåte ut fra materialenes egenskaper og konstruksjonens funksjon
- *bruke sensorer og styringssystemer i forbindelse med forsøk og konstruksjoner*
- *dokumentere og vurdere konstruksjoners fysiske egenskaper og funksjonalitet ved hjelp av målinger og enkle beregninger*

Den unge forskeren - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne:

- *gjøre rede for hvordan et naturvitenskapelig prosjekt planlegges, gjennomføres og etterarbeides før det blir publisert (planlegge målinger utført i løpet av droppet)*
- *planlegge, gjennomføre, analysere og dokumentere systematiske målinger*
- om støy, *luftforurensning*, inneklima og vannkvalitet, og drøfte virkninger på helse og miljø

Teknologi, naturvitenskap og samfunn - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne

- drøfte etiske, miljømessige, kulturelle og politiske sider ved teknologisk utvikling
- *beskrive den historiske utviklingen av en teknologisk innretning, forklare virkemåten og drøfte anvendelser i samfunnet*
- gjøre rede for utvikling og produksjon av et teknologisk produkt og vurdere produktets brukervennlighet, utviklingsmuligheter og miljøpåvirkning
- *beskrive prinsipper og virkemåte for noen moderne instrumenter i industri, helsevesen eller forskning, og gjøre rede for nytten og eventuelle skadevirkninger*
- kartlegge og presentere praktisk bruk av realfag i en lokal bedrift eller institusjon

Design og produktutvikling - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne



- *gjøre rede for funksjonen til vanlige komponenter i elektroniske kretser, og gjenkjenne komponentene i en krets*
- *lage elektroniske kretser ved å lodde komponenter og simulere og teste kretsene*
- *forme og utvikle produkter som har en definert funksjon og inneholder elektronikk*
- *dokumentere og presentere designprosesser fra idé til ferdig produkt*
- *begrunne valg av materialer i produkter og vurdere produktenes form og funksjon, miljømessige konsekvenser, estetikk og forbedringsmuligheter*
- *utføre målinger med eller teste et eget produkt, og vurdere kvaliteten på produktet med tanke på funksjonalitet*

F.2 Teknologi og forskningslære 2

Den unge forskeren - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne

- *gjøre rede for et forskningsprosjekt i en bedrift eller institusjon, og beskrive problemstillinger, organisering, målestyr, resultater og finansiering*
- *planlegge og gjennomføre naturvitenskapelige undersøkelser basert på egne ideer, og presentere arbeidet i en vitenskapelig form*
- *drøfte resultater fra egne undersøkelser i forhold til relevant kunnskap på området, og vurdere hvordan kontroll av variabler og reproduserbarhet er ivarettatt*

Naturvitenskapelige arbeidsmetoder - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne

- *forklare hva som menes med modell, teori og hypotese, og gjøre rede for hvordan de brukes og utvikles i forskning*
- *drøfte ved å bruke eksempler hvordan empiriske data kan styrke eller forkaste en hypotese*
- *gjøre rede for hvordan forskning utvikles og kvalitetssikres gjennom samarbeid, kritisk vurdering og argumentasjon*
- *gjøre rede for strukturen i en vitenskapelig publikasjon eller presentasjon*

Forskning, teknologi og samfunn - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne

- *beskrive kjennetegn ved grunnforskning, anvendt forskning og utviklingsarbeid og gjøre rede for hovedtrekk ved finansiering og styring*
- *gjøre rede for betydningen av naturvitenskapelig forskning og teknologiutvikling for næringsliv og samfunn*



- drøfte økonomiske, miljømessige og etiske spørsmål i forbindelse med naturvitenskapelig forskning og teknologiutvikling
- *drøfte og gi eksempler på hvordan forskningsresultater og ny teknologi formidles og brukes av forskningsinstitusjoner, medier, bedrifter, interessegrupper og myndigheter*

Vitenskapsfilosofi og vitenskapsteori - Beskrivelse av hovedområde

Mål for opplæringen er at eleven skal kunne

- beskrive hovedtrekk i den historiske utviklingen av vitenskapelige tenkemåter og drøfte teknologiens rolle i denne utviklingen
- gjøre rede for hovedideene til noen sentrale vitenskapsteoretikere og vitenskapsfilosofer vurdere hvordan argumentasjon i aktuelle naturvitenskapelige debatter bygger på empiriske resultater, teoretisk kunnskap og ideologisk ståsted

F.3 Teknologi i Praksis (TiP) - ungdomsskolen

Formål

Valfaga skal medverke til at elevane, kvar for seg og i fellesskap, styrker lysta til å lære og opplever meistring gjennom praktisk og variert arbeid. Valfaga er tverrfaglege og skal medverke til heilskap og samanheng i opplæringa.

Teknologi handlar om den menneskeskapte verda og om innretningar og system som kan gjere kvardagen betre. Opp gjennom tidene har menneska brukt kreativitet og skaparevner til å utvikle reiskapar, maskinar og andre teknologiske produkt og løysingar. Teknologien grip inn på mange område, og har gjeve og kan gje både moglegheiter og utfordringar, både for den einskilde og for samfunnet. Innanfor teknologien finn vi dei enklaste verktøy og produkt og den mest avanserte elektronikken. Erfaring med og innsikt i teknologi kan fremje personleg utvikling, demokratisk deltaking og medverke til eit aktivt forhold til ein teknologisk kvardag.

Valfaget teknologi i praksis skal motivere elevane til å utvikle teknologiske produkt med utgangspunkt i lokale behov og problemstillingar. Prosessen frå idé til eit ferdig produkt kan medverke til skaparglede og meistringsoppleving. Gjennom eige arbeid og i samarbeid med andre kan elevane utvikle ferdigheiter og innsikt. Det inneber å prøve ut eigne talent og moglegheiter på ulike steg i prosessen, vurdere prosessar og produkt og få tilbakemeldingar frå andre.

Valfaget handlar om å planleggje, konstruere og framstille gjenstandar og produkt med varierte materiale og teknologiske løysingar. Kunnskap om teknologiske produkt som blir brukte i dagleglivet, gjev eit godt grunnlag for å forbetre produkt og utvikle nye produkt.

Valfaget hentar hovudelement frå matematikk, naturfag og kunst og handverk/duodji. Element frå norsk/samisk, RLE og samfunnsfag kan også inngå.

Hovedområder

Undersøkingar



Hovedområdet handlar om korleis teknologiske produkt er konstruerte og verkar, kva for prosessar som inngår i utvikling og bruk, og kva for behov produkta dekkjer. Utvikling, konstruksjon og produksjon av teknologi inngår i hovudområdet, i tillegg til helse, miljø og sikkerheit (HMS). Kunnskap om korleis teknologien byggjer på nokre grunnleggjande prinsipp, og korleis ny teknologi byggjer på tidlegare erfaringar, høyrer også med til hovudområdet.

Idéutvikling og produksjon

Hovudområdet omfattar planlegging, framstilling og utprøving av eigne produkt og konstruksjonar. Planar for framstilling og utprøving av eigne produkt og konstruksjonar byggjer på kravspesifikasjon.

I utviklingsfasen er kjennskap til design og verkemåte til andre produkt viktig. Diskusjon omkring ulike sider ved produkta er viktig i alle fasar av produktutviklinga og kan også medverke til å forbetre prosessar og produkt.

Kompetansemål

Undersøkingar

- undersøkje teknologiske produkt og dei vala som er gjorde med omsyn til bruk, tekniske løysingar, funksjonalitet og design
- *demonstrere riktig bruk av utvalde verktøy*
- vurdere teknologiske produkt ut frå brukartilpassing, HMS-krav og miljøtilpassing

Idéutvikling og produksjon

- *utvikle ein realistisk kravspesifikasjon for eit teknologisk produkt og beskrive kva behov produktet skal dekkje*
- *framstille produktet med eigna materiale, komponentar, og funksjonelle teknologiske løysingar*
- *bruke kunnskap om andre produkt i arbeidet med eige produkt*
- *teste eigne produkt og foreslå moglege forbetringar*

F.4 Forskning i Praksis (FiP) - Ungdomsskolen

Formål

Valgfagene skal bidra til at elevene, hver for seg og i fellesskap, styrker lysten til å lære og opplever mestring gjennom praktisk og variert arbeid. Valgfagene er tverrfaglige og skal bidra til helhet og sammenheng i opplæringen.

Forskning handlar om å utvikle ny kunnskap og innsikt. Formuleringar som ”forskning viser at” eller ”vitenskapelige undersøkingar har vist at” brukes ofte og viser at tilliten til forskning er stor. Forskning skal også være kritisk, prøve ut om forskningsresultater er riktige og bidra til debatt. Erfaring med utforskning kan derfor danne grunnlag for egne meninger og kritisk tenkning, og gi elevene bedre mulighet til å forholde seg til debatt om forskning på en hensiktsmessig måte.



Opplæringen i valgfaget forskning i praksis skal bidra til at elevene får erfaring med vitenskapelige metoder og arbeidsmåter. Valgfaget skal stimulere til undring, aktiv handling for å teste ut løsninger og utvikle evnen til å stille nye spørsmål. Opplæringen skal bidra til å finne forklaringer på det som er observert, og gjennom kildegransking, eksperiment og observasjon kontrollere om forklaringene holder.

I valgfaget forskning i praksis blir elevene utfordret til å undersøke aktuelle spørsmål relatert til natur, miljø og klima, samt kultur-, samfunns- og arbeidsliv. Dette innebærer at de skal finne problemstillinger de ønsker å undersøke, vurdere mulige forklaringer, planlegge og gjennomføre undersøkelser, bruke utstyr og teknikker for datainnsamling, bearbeide data, og vurdere og formidle resultatene. Slik bidrar valgfaget med erfaringer og ferdigheter om de praktiske sidene ved forskning. Valgfaget stimulerer til nysgjerrighet og bruk av fantasi for å finne forklaringer. Det bidrar også til innsikt i hvordan etablert kunnskap og andres forskning kan trekkes inn i egen utforskning. Opplæringen skal legge til rette for at elevene får erfare hvordan det å gi og få tilbakemeldinger underveis i en utforskende prosess fremmer kvaliteten på resultatene.

Valgfaget henter hovedelementer fra naturfag, matematikk og samfunnsfag, men tema kan også hentes fra ungdomstrinnets øvrige fag.

Hovedområder

Idéutvikling

Hovedområdet omfatter de kreative sidene som inngår i alle stadier av forskningsprosessen. Utgangspunktet er en problemstilling som man lurer på og vil finne svar på. Det omfatter videre å lage gode forskningsspørsmål, få innspill fra andre, reformulere i lys av innspillene og, planlegge undersøkelser. I dette hovedområdet utvikler elevene mulige hypoteser, henter inn andre relevante data og sammenlikner og diskuterer ideer med andre elever.

Praktisk utforskning

Hovedområdet omfatter arbeid med ulike prosesser ved praktisk gjennomføring av de planlagte undersøkelsene. Det er viktig at elevene prøver ut og bruker forskjellige metoder, gjør seg kjent med og tar i bruk ulike typer utstyr og samler inn og systematiserer data. I tillegg skal elevene prøve ut tolkninger og sammenlikne egne funn med andres samt formidle resultatet.

Kompetansemål

Idéutvikling

- *finne problemstillinger, formulere forskbare spørsmål og forslag til hypoteser*
- *planlegge undersøkelser basert på egne eller gruppas forskningsspørsmål og hypoteser*
- *delta i samtaler om egen og andres utforskning*
- *argumentere for egne hypoteser i lys av funn og andre relevante undersøkelser*

Praktisk utforskning

- *gjennomføre planlagte undersøkelser og foreta relevante justeringer underveis*
- *bruke relevante metoder og utstyr for innsamling og analyse av data*



-
- *demonstrere og forklare metoder, virkemåten til utstyr og prosedyrer for datainnsamling i gjennomførte forskningsprosjekter*
 - *systematisere data slik at mønstre kommer tydelig fram og vurdere usikkerheter*
 - *formidle resultater fra egne prosjekter*





Hensikten med dette heftet er å samle erfaringer fra utprøving av CanSat-kittet benyttet ved kurset Miljø og Romteknologi (CanSat) - høsten 2012 – 15, samt å beskrives ulike sensorer som kan vurderes anvendt sammen med CanSat.

Beskrivelsen av sensorer strekker seg ut over anvendelser i CanSat og teknologi og forskningslære. Vi har valgt en noe bredere tilnærming til feltet siden sensorer brukes i mange ulike sammenhenger og ikke bare innen romteknologi. Heftet kan derfor brukes som tilleggsstoff til Teknologi og Forskningslære 1 og Fysikk 1 i videregående skole, men også av lærere som underviser de nye valgfagene i ungdomsskolen Teknologi i Praksis og Forskning i Praksis.

Nils Kr. Rossing

Førstelektor ved Skolelaboratoriet og
prosjektleder ved Vitensenteret
E-post: nils.rossing@plu.ntnu.no

Thomas Gansmoe

NAROM
E-post: thomas@narom.no

Jøran Grande

NAROM
E-post: joran@narom.no

Jens Dalsgaard Nielsen

Professor, Aalborg universitet, Danmark
E-post: jdn@es.aau.dk



Trondheim

**Program for
lærerutdanning**

Skolelaboratoriet

for matematikk, naturfag
og teknologi

Tlf. 73 55 11 43

Faks 73 55 11 40

<http://www.ntnu.no/skolelab>