

# Linux-based Command and Control for Off-grid Research Facility

BO16-G16

Niklas Kalmár, Morten R. Linstad, Martin Todorov og Rune W. Rokne

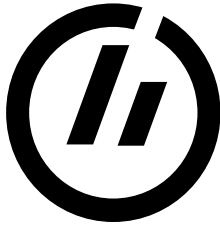


Bacheloroppgaven 2016 ved Høgskolen i Østfold, IT-avdelingen

Remmen, Halden

12.05.16





# HØGSKOLEN I ØSTFOLD

Avdeling for Informasjonsteknologi  
Remmen  
1757 Halden  
Telefon: 69 21 50 00  
URL: [www.hiof.no](http://www.hiof.no)

## BACHELOROPPGAVE

Prosjektkategori: Utvikling	<input checked="" type="checkbox"/>	Fritt tilgjengelig
Omfang i studiepoeng: 20	<input type="checkbox"/>	Fritt tilgjengelig etter
Fagområde: Dataingeniør	<input type="checkbox"/>	Tilgjengelig etter avtale med oppdragsgiver

Tittel: Linux-based Command and Control for Off-grid Research Facility	Dato: 12. mai 2016
Forfattere: Niklas Kalmar, Morten R. Linstad, Martin Todorov & Rune W. Rokne	Veileder: Gunnar Misund
Avdeling / Program: Avdeling for Informasjonsteknologi	Gruppenummer: BO16-G16
Oppdragsgiver: HiØ/IT	Kontaktperson hos oppdragsgiver: Erling Strand og Geir Strøm

### Ekstrakt:

En vitenskapelig målestasjon skal plasseres et sted uten tilgang på strøm. Energibehovet dekkes av egenprodusert strøm og batterier, men energisystemet må overvåkes og de enkelte komponentene må styres. Vi ser for oss at dette best kan løses med en Linux-basert løsning med internetttilgang. Systemet rapporterer til en database på HiØ og eventuelle avvik må generere alarmer. Det skal logges data om energisystemene samt vær/vind-data. Databasen må ha et webgrensesnitt som gir god oversikt. Fra dag til dag må systemet forvalte energien fra egen produksjon, batterier og om nødvendig tilskudd fra en brenselcelle (begrenset med drivstoff). Feil må oppdages og varsles. I første omgang vil systemet i sin helhet prøves ut på Remmen vinteren 2015/2016, men etter hvert skal målestasjonen plasseres i Hessdalen, nær Rørørs.

3 emneord:

Energiproduksjon
Linux
Forskningsstasjon





# Sammendrag

Prosjektet gikk ut på å bygge en autonom målestasjon som produserte sin egen energi. Prosjektets oppdragsgiver var *Project Hessdalen*. Formålet med prosjektet har vært å gi et grunnlag for å undersøke i hvor stor grad en målestasjon kan være selvforsynt med energi.

Systemet som ble bygd er delt i fire undersystemer; Målesystemet, Energisystemet, Styrings-systemet og det Eksterne Lagringssystemet.

Energisystemet som ble utviklet består av en brenselcelle, solcellepaneler, en vindmølle og en tilhørende batteribank. Målesystemet som ble utviklet utfører klimatiske målinger og det brukes sensorer som er egnet for forholdene de skal operere under. Styringssystemet utgjøres av en Raspberry Pi som kjører en Linux distribusjon. Det Eksterne Lagringssystemet består av en Linux server installert med en Apache webserver og en MySQL database.

Ved implementasjon var det nødvendig å bygge flere kretskort som gjorde det mulig å opprette en kobling og kommunikasjon mellom de ulike systemene. For kommunikasjon ble det benyttet flere forskjellige standarder, både digitale og analoge. Det var også nødvendig å gjøre tilpasninger for å sørge for at forskjellige enheter i systemet fikk riktig driftspenning, og at målesignalene fra sensorene var innenfor riktig spenningsrange. Det måtte også programmeres applikasjoner for både autonom og manuell styring og overvåking av systemene. Det ble installert en database både lokalt på Styringssystemet og på Det Eksterne Lagringssystemet. Styringssystemet ble koblet til en mobil Internett ruter, deretter ble det opprettet en kobling mellom de to databasene. Et webgrensesnitt ble laget for fremvisning av de historiske dataene til både energi- og målesystemet.

Det ble gjennomført en rekke tester av de ulike delene i systemet. Testingen ble gjennomført som en iterativ prosess over to runder og det ble avdekket flere svakheter blant annet i webgrensesnittet og styringssystemet. Disse feilene ble rettet opp i testfasen.

Systemet som er bygd gir et godt grunnlag for å undersøke i hvor stor grad en målestasjon kan være selvforsynt med energi. Styringssystemet overvåker og registrerer data om målestasjonens energisystem, og forbruksdata og produksjonsdata er direkte sammenlignbart. Hovedformålet anses derfor som nådd.



# Takk Til

Prosjektgruppen vil rette en stor takk til gruppens veileder Gunnar Misund, for god veiledning gjennom hele prosjektet. Det vil også rettes en stor takk til gruppens oppdragsgivere Erling P. Strand og Geir K. Strøm for et spennende prosjekt og for gode råd og hjelp.



# Innhold

<b>Sammendrag</b>	<b>i</b>
<b>Takk Til</b>	<b>iii</b>
<b>Figurliste</b>	<b>x</b>
<b>Tabelliste</b>	<b>xi</b>
<b>1 Introduksjon</b>	<b>1</b>
1.1 Prosjektgruppen . . . . .	1
1.2 Oppdragsgiver . . . . .	1
1.3 Bakgrunn for Oppdraget . . . . .	1
1.3.1 Bachelorprosjektet . . . . .	3
1.4 Formål og hovedmål . . . . .	4
1.4.1 Delmål 1 . . . . .	4
1.4.2 Delmål 2 . . . . .	4
1.5 Leveranser . . . . .	4
1.6 Metode . . . . .	4
1.7 Rapportstruktur . . . . .	5
<b>2 Bakgrunnsanalyse</b>	<b>7</b>
2.1 Energisystemet . . . . .	7
2.1.1 Solenergi . . . . .	8
2.1.2 Vindenergi . . . . .	9
Vindturbinteknologi . . . . .	9
2.1.3 Ikke-fornybar energi . . . . .	9
Stasjonen . . . . .	10
En brenselcelle . . . . .	10
2.1.4 Batteribank . . . . .	10
Hvorfor? . . . . .	10
2.2 Målesystem . . . . .	10
2.2.1 Temperatur . . . . .	11
2.2.1.1 Måling av temperatur . . . . .	11
Det finnes flere metoder . . . . .	11
2.2.2 Vind . . . . .	12
2.2.2.1 Måling av vind . . . . .	12
To metoder . . . . .	12
2.2.3 Elektrisk strøm . . . . .	13

2.2.3.1	Måling av strøm . . . . .	13
2.2.4	Luftfuktighet . . . . .	14
2.2.4.1	Måling av fuktighet . . . . .	14
	Generelle metoder . . . . .	14
2.2.5	Krav til sensorer . . . . .	14
2.3	Styringssystemet . . . . .	15
2.3.1	Hva skal styres? . . . . .	16
2.3.2	Hvordan kan et styringssystem overstyres? . . . . .	16
2.3.3	Styringsenhet . . . . .	16
2.3.4	Krav til styringsenhet for dette prosjektet . . . . .	16
2.3.5	Alternativer . . . . .	17
2.4	Det Eksterne Lagringssystemet . . . . .	17
2.4.1	Mulige løsninger . . . . .	17
2.5	Lignende prosjekter . . . . .	18
2.5.1	Prosjekt Hessdalen - Værstasjoner . . . . .	18
2.5.2	Complete DIY Raspberry Pi Weather Station with Software . . . . .	19
2.5.3	Low cost Arduino/Android-based Energy-Efficient Home Automation System with Smart Task Scheduling . . . . .	19
<b>3</b>	<b>Planlegging</b>	<b>21</b>
3.1	Energisystemet . . . . .	21
3.1.1	Chargekontrollere . . . . .	21
3.1.1.1	TriStar MPPT-60 . . . . .	21
3.1.1.2	TriStar-45 . . . . .	22
3.1.1.3	Styrker og svakheter ved chargekontrollerne . . . . .	23
3.1.2	Brenselcelle . . . . .	23
3.1.2.1	EFOY Pro 2400 Duo . . . . .	24
3.1.2.2	Ekstraustyr for brenselcellen . . . . .	25
	Interface Adapter - A1 . . . . .	25
	EFOY DuoCartSwitch . . . . .	25
	Zarges isolasjonkasse . . . . .	25
3.1.2.3	Styrker og svakheter . . . . .	26
3.2	Målesystemet . . . . .	27
3.2.1	Vindsensorer . . . . .	27
3.2.1.1	Wind Speed Transmitter - Compact . . . . .	27
3.2.1.2	Wind Direction Transmitter - Compact . . . . .	28
3.2.1.3	Ekstraustyr for vindsensorene . . . . .	29
3.2.1.4	Styrker og svakheter . . . . .	29
3.2.2	Temperatur- og fuktighetssensorer . . . . .	30
3.2.3	EE210 Outdoor . . . . .	30
3.2.4	Styrker og svakheter - EE210 Outdoor . . . . .	30
3.2.5	HTU21D . . . . .	31
3.2.5.1	Styrker og svakheter - HTU21D . . . . .	31
3.3	Styringssystemet . . . . .	32
	Raspberry Pi . . . . .	32
	Styrker og svakheter . . . . .	33
	Et linux basert system . . . . .	33

	Tilleggsutstyr til Raspberry Pi . . . . .	33
3.3.1	Hvordan skal den styre? . . . . .	34
3.3.2	Planen for overstyring og fjernstyring . . . . .	35
3.4	Plan for det eksterne lagringssystemet . . . . .	35
<b>4</b>	<b>Implementasjon</b>	<b>37</b>
4.1	Energisystemet . . . . .	37
4.2	Målesystemet . . . . .	39
4.2.1	Elektronikk sensordel . . . . .	40
4.3	Styringssystemet . . . . .	42
4.3.1	Elektronikk styringsdel . . . . .	43
4.3.2	Klargjøring av RPi . . . . .	45
4.3.3	Hovedprogram . . . . .	45
	Tanken bak . . . . .	45
	Kommunikasjon . . . . .	46
	Modulen for ADCene . . . . .	47
	Kommunikasjonen med charge controller . . . . .	47
	HTU21D . . . . .	47
	For å opprette kommunikasjon mellom RPi og brenselcellen . . . . .	47
	For styring av enheter . . . . .	48
	Alarm og sammenligningssystemet . . . . .	48
	Database logging . . . . .	48
	Configfil . . . . .	49
4.3.4	Overstyring . . . . .	49
4.3.5	CRON-jobber . . . . .	50
	MySQL-dump . . . . .	50
	Ruteren . . . . .	50
	En del av loggfilene . . . . .	50
	Fordi loggingen . . . . .	50
	Pigpiod . . . . .	51
	Hovedprogrammet . . . . .	51
4.3.6	Watchdog . . . . .	51
4.3.6.1	Ruter . . . . .	51
4.4	Det eksterne lagringssystemet . . . . .	51
4.4.1	Lokal database på Raspberry Pi . . . . .	51
	Vaer-tabellen . . . . .	51
	Produksjon-tabellen . . . . .	52
	Forbruk-tabellen . . . . .	52
	Drivstoff-tabellen . . . . .	52
4.4.2	Ekstern database ved HiØ . . . . .	52
4.5	Webgrensesnitt . . . . .	53
<b>5</b>	<b>Testing</b>	<b>57</b>
5.1	Førsterunde teknisk testing . . . . .	57
5.2	Andrerunde teknisk testing . . . . .	62
5.3	Testing som ikke har blitt utført i dette prosjektet . . . . .	63

<b>6</b>	<b>Diskusjon</b>	<b>65</b>
6.1	Formål og hovedmål . . . . .	65
6.2	Leveranser . . . . .	66
6.3	Metode . . . . .	66
6.4	Oppgaven . . . . .	66
6.4.1	Det positive . . . . .	67
6.4.2	Problemer underveis . . . . .	67
6.4.3	Hva kunne ha blitt gjort annerledes? . . . . .	68
<b>7</b>	<b>Konklusjon</b>	<b>69</b>
7.1	Videreføring . . . . .	69
	<b>Bibliografi</b>	<b>71</b>
	<b>Vedlegg</b>	<b>73</b>
<b>A</b>		<b>73</b>



# Figurer

1.1	AMS som står der nå . . . . .	2
1.2	AMS sin nåværende posisjon . . . . .	3
2.1	Et oversiktsbilde av systemet som skal bygges . . . . .	8
2.2	Et oversiktsbilde av dataoverføring . . . . .	18
3.1	Detaljert system modell . . . . .	22
3.2	Figur av chargekontroller . . . . .	23
3.3	EFOY Pro 2400 Duo . . . . .	24
3.4	Interface manual . . . . .	25
3.5	DuoCartSwitch . . . . .	26
3.6	Zarges isolasjonskasse plassert i vognen . . . . .	26
3.7	Vindhastighetsensor . . . . .	27
3.8	Vindretningssensor . . . . .	28
3.9	Monteringsstang for vindsensorene . . . . .	29
3.10	EE210 Outdoor . . . . .	30
3.11	HTU21D . . . . .	31
3.12	Raspberry Pi 2 . . . . .	32
3.13	ADC Pi Plus . . . . .	34
4.1	Bildet viser boden med solcellepanel montert på den ene siden . . . . .	38
4.2	Spenningsregulator kretsen utgave 1 . . . . .	38
4.3	Eksempeldata som kan hentes ut fra brenselcellen . . . . .	39
4.4	Batterier som er utenfor kassen til brenselcellen . . . . .	39
4.5	Knutepunktet for sammenslåing av batteriene og strømuttak for enheter . . . . .	40
4.6	Masten med sensorer for vindretning- og hastighet montert på toppen . . . . .	41
4.7	En tradisjonell spenningsdeler på inngangssignalet blir galt. . . . .	42
4.8	Inngangsimpedansen fungerer som en spenningsdeler med en ytre motstand. . . . .	42
4.9	Optkoblerens utgangssignal kobles som jordsignal på styringselement. . . . .	44
4.10	Kommunikasjonen mellom enheter og hovedprogrammet. . . . .	46
4.11	Et bilde som illustrerer menyen for overstyringsprogrammet. . . . .	49
4.12	Et bilde som illustrerer databasen for overstyringsprogrammet. . . . .	50
4.13	De fire tabellene. . . . .	52
4.14	Tabell som viser temperaturmålinger . . . . .	53
4.15	Graf som viser temperaturmålinger . . . . .	54
4.16	Stolpediagram for visning av drivstoffniåv . . . . .	54
4.17	Forsiden av webgrensesnittet . . . . .	55

5.1	Temperatur og vind måling fra systemet . . . . .	59
5.2	Tabell fra yr . . . . .	59

# Tabeller

5.1	Testtabell for webgrensesnitt. . . . .	57
5.2	Testtabell for database på ekstern server. . . . .	58
5.3	Testtabell for database. . . . .	58
5.4	Testtabell for loggfiler. . . . .	59
5.5	Testtabell for alarmsystemet test 1. . . . .	60
5.6	Testtabell for styring test 1. . . . .	60
5.7	Testtabell for overstyring test 1. . . . .	61
5.8	Testtabell for kommunikasjon test 1. . . . .	61
5.9	Testtabell for alarmsystemet test 2. . . . .	62
5.10	Testtabell 2 for webgrensesnitt. . . . .	62
5.11	Testtabell for styring test 2. . . . .	62
5.12	Testtabell for overstyring test 2. . . . .	63



# Kapittel 1

## Introduksjon

Dette kapitlet vil ta for seg introduksjon av gruppemedlemmene, oppdragsgiver og gir en gjennomgang av bachelorprosjektet. Det vil også bli presentert hovedmål og delmål samt hvilken metode som skal benyttes i prosjektet.

### 1.1 Prosjektgruppen

Morten R. Linstad er 3. års dataingeniørstudent. Har tidligere gått Elektro og Elektronikk på videregående skole. Har en variert arbeidsbakgrunn innen helse og service. Har jobbet som studentassistent i Matematikk 1, Matematikk for IT og jobber nå som studentassistent i Statistikk og økonomi.

Martin Solberg Todorov er 3. års dataingeniørstudent. Har arbeidserfaring innen industri og service. Jobber nå som studentassistent i faget Databaser.

Rune Weisæth Rokne er 3. års dataingeniørstudent. Har tidligere gått IKT på videregående skole. Har jobbet med ulike ting og har et fagbrev som IT-servicemedarbeider.

Alf Niklas Håkonsen Kalmár er 3.års dataingeniørstudent. Har jobbet som studentassistent i fagene Datakommunikasjon, Matematikk 1 og Fysikk og kjemi.

### 1.2 Oppdragsgiver

Oppdragsgiver er *Project Hessdalen* som er et langtidsprosjekt ved Høgskolen i Østfold. Kontaktpersoner er høgskolelektor Erling P. Strand og overingeniør Geir Kramviken Strøm, begge ansatt ved Høgskolen i Østfold.

### 1.3 Bakgrunn for Oppdraget

*Project Hessdalen* er et prosjekt som ble startet i 1983. Idag er dette et samarbeidsprosjekt mellom avdeling for informasjonsteknologi og avdeling for ingeniørfag ved Høgskolen i Østfold. Prosjektet ble etablert etter at det hadde blitt observert en rekke ukjente lys-fenomener i Hessdalen like utenfor Røros i Norge, disse fenomenene går under samlebetegnelsen Hessdalsfenomenet. *Project Hessdalen* søker å finne ut mer om disse lysfenomenene.

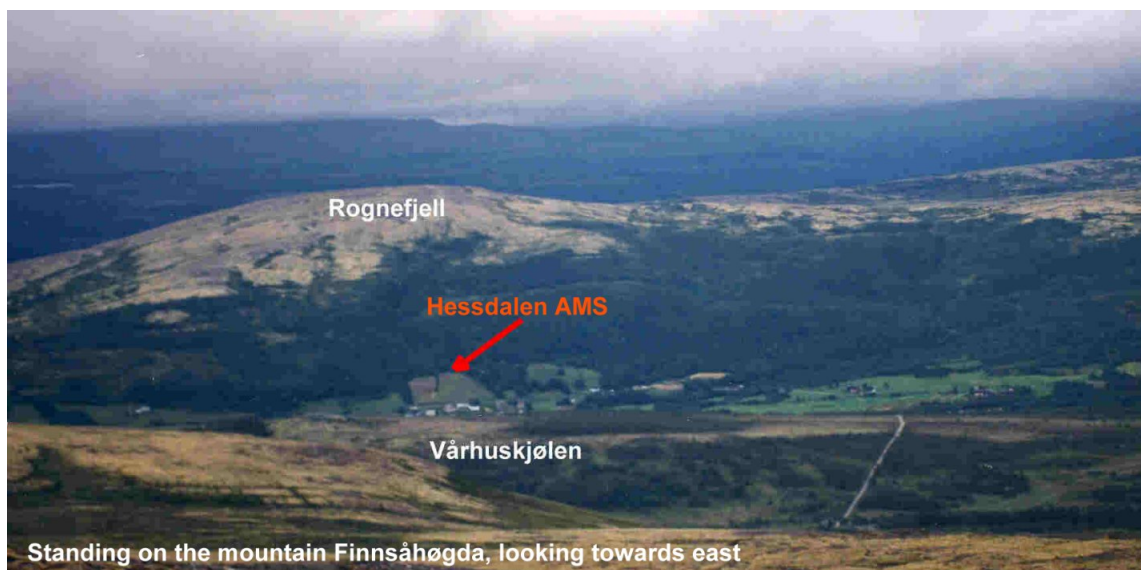
I dag har *Project Hessdalen* en automatisk målestasjon (AMS) plassert i Hessdalen vist i figur 1.1. Denne målestasjonen er tilkoblet strømmettet, og har ikke et eget system for energiproduksjon. Stasjonen har tre forskjellige kamerasystemer tilkoblet. I tillegg gjør den værmålinger, målinger

for elektromagnetisk stråling, og den har et magnetometer og en såkalt ”reg-sensor” koblet til. Denne målestasjonen står plassert litt lenger ned i dalen.

Nå ønsker *Project Hessdalen* å teste ut i hvor stor grad en felt-forskningsstasjon kan dekke sitt eget energibehov. Hvis en målestasjon kan klare å produsere nok energi til å drifte seg selv, kan en fremtidig målestasjon plasseres fritt uavhengig av hvordan strømmettet er lagt opp i Hessdalen. En ønsket lokasjon er på toppen av Rognefjell vist i figur 1.2 [P.Strand, 2015].



Figur 1.1: Bildet viser den nåværende automatiske målestasjonen (AMS) i Hessdalen som er tilkoblet strømmettet.<sup>2</sup>



Figur 1.2: Bildet viser Rognefjell som ruver over den nåværende målestasjonen Hessdalen AMS <sup>3</sup>

### 1.3.1 Bachelorprosjektet

Det skal lages et styring- og overvåkningssystem for en målestasjon. Dette skal gjøres i et prosjekt som har fått navnet "Linux-based Command and Control for Off-grid Research Facility". Dette systemet skal utvikles ved Høgskolen i Østfold, avdeling Remmen. Målestasjonen skal i størst mulig grad være autonom, men skal kunne fjernstyres via internettilkobling ved behov. For å bygge en slik målestasjon, er det kjøpt inn en enkel mobil brakke-container. Energisystemet som skal benyttes skal bestå av solcellepaneler, en vindmølle og en brenselcelle. Den skal også ha en tilhørende batteribank, hvor produsert energi lagres. Det er kjøpt inn batteribank, solcellepaneler og charge controller, men brenselcelle og vindmølle er foreløpig ikke bestilt.

Systemet skal også logge og sende vær/vind-data og data om energisystemet til en database i en ekstern server lokalisert ved Høgskolen i Østfold. Det er derfor tenkt å bruke en liten datamaskin som en sentral intelligent enhet i systemet. Denne skal kjøre et Linux operativsystem. Databasen i den eksterne serveren som mottar data skal ha et web-grensesnitt hvor dataene presenteres. I hvilket tidsintervall de forskjellige dataene skal logges og sendes vil bestemmes senere i prosjektet i samarbeid med oppdragsgiver. Systemet må ha flere forskjellige sensorer som måler innetemperatur, utetemperatur og vind. Det kan også være aktuelt med fuktighetssensorer inne i stasjonen. Det skal også plasseres et kamera i nærheten av stasjonen. Dette kameraet må ha et varmeelement som fjerner snø og is fra å hindre sikten. Dette varmeelementet vil være den mest energikrevende komponenten i systemet. Det er også viktig at forskningstasjonen har et alarmsystem som kan varsle dersom uønskede tilstander oppstår i systemet. Et passende system for dette må utvikles i tett dialog med oppdragsgiver. Det er foreløpig ukjent hvilke tilstander som skal gi alarmer. Det kan også være aktuelt å benytte redundans for enheter som er kritiske for at systemet ikke skal feile, eller i seg selv kan være spesielt utsatt for å svikte. Det er foreløpig ikke

<sup>2</sup>AMS som står der nå: <http://www.hessdalen.org/pict/station/ams-new1-big.jpg>

<sup>3</sup>AMS sin nåværende posisjon: <http://www.hessdalen.org/pict/nature/ams-finnsa-big.jpg>

klart hvor og om det i det hele tatt skal benyttes redundans. Her vil også pris og energiforbruk til enhetene ha en betydning for hvilke valg som fattes.

## 1.4 Formål og hovedmål

Prosjektet vil kunne gi grunnlag for å undersøke i hvor stor grad en målestasjon drevet av dette energisystemet kan være selvforsynt på energi. Dette vil igjen kunne gi svar på begrensningene målestasjonen har på bruk av komponenter som krever energi.

### 1.4.1 Delmål 1

*Project Hessdalen* får et egenutviklet styring- og overvåkningssystem som kan benyttes til målestasjoner med et eget energisystem.

### 1.4.2 Delmål 2

Et styring- og overvåkningssystem vil kunne benyttes av *Project Hessdalen* til å vurdere hvor mye utstyr energisystemet har kapasitet til å drifte.

## 1.5 Leveranser

Gruppen skal levere et styring- og overvåkningssystem til et energisystem i henhold til oppdragbeskrivelsen. Basert på data fra systemet skal det lages et webgrensesnitt for å presentere disse på en oversiktlig måte.

I tillegg skal gruppen levere hovedrapport, kontrakter, hjemmeside, forprosjektrapport, prosjektplakat og refleksjonsnotat.

## 1.6 Metode

Ved prosjektets startfase kartlegges andre prosjekter og teknologiske løsninger som er relevante for denne oppgaven. Denne informasjonen kan bli brukt til å minske tidsbruken på prøving og feiling.

Gruppen vil lage en oversikt over hvilke enheter som skal inngå i systemet, samt en modell som beskriver hvordan det skal settes opp. Modellen skal beskrive alle fysiske enheter som skal inngå i systemet og hvordan disse skal kommunisere. Når modellen er ferdigstilt vil det lages en bestillingsliste som avklarer hvilke enheter som skal kjøpes inn.

Montering vil foregå i tett samarbeid med oppdragsgiver for å sikre at dette foregår etter oppdragsgivers ønske. Ved usikkerhet rundt montering av utstyr vil aktuell leverandør kontaktes for å forhindre skade som kan oppstå ved feilkobling.

Utvikling og testing vil bli gjort simultant eller uavhengig av montering da det er usikkerhet tilknyttet leveringstiden på de ulike enhetene. Avhengig av hvilke enheter som er tilgjengelig kan det tenkes at gruppen vil benytte seg av simulering for å kunne ferdigstille deler av prosjektet.

Etter å ha utviklet de ulike delene i modellen, må løsningene testes for å se om kravet som blir stilt i modellen er oppfylt. Dette er en iterativ prosess hvor prosjektgruppen til enhver tid vil vurdere om kravene er oppfylt eller om endringer må gjøres. Ved å gjennomføre prosjektet på denne måten vil prosjektet hele tiden bli testet og vurdert opp mot oppdragsgiver sitt ønske.



## **1.7 Rapportstruktur**

Rapporten er bestående av ulike kapitler som har ulike formål.

I kapittel 2 blir det tatt for seg en grundig analyse og utdypning av oppgaven til prosjektet samt en analyse av relevant teknologi og andre relevante prosjekter.

Videre i kapittel 3 er det planlegging og design for prosjektet. Her vil de ulike enhetene som skal bli tatt i bruk bestemmes.

Kapittel 4 tar for seg implementeringen av prosjektet hvor løsninger dokumenteres og i kapittel 5 blir det en evaluering av det implementerte arbeidet. I tillegg vil testingen som blir utført dokumentert i dette kapitlet.

De to siste kapitlene vil være en vurdering av arbeidet, hvor kapittel 6 tar for seg om målene er blitt nådd og i hvilken grad prosjektet har blitt utført. I kapittel 7 vil det være en konklusjon for prosjektet og forslag til fremtidige prosjekter.



## Kapittel 2

# Bakgrunnsanalyse

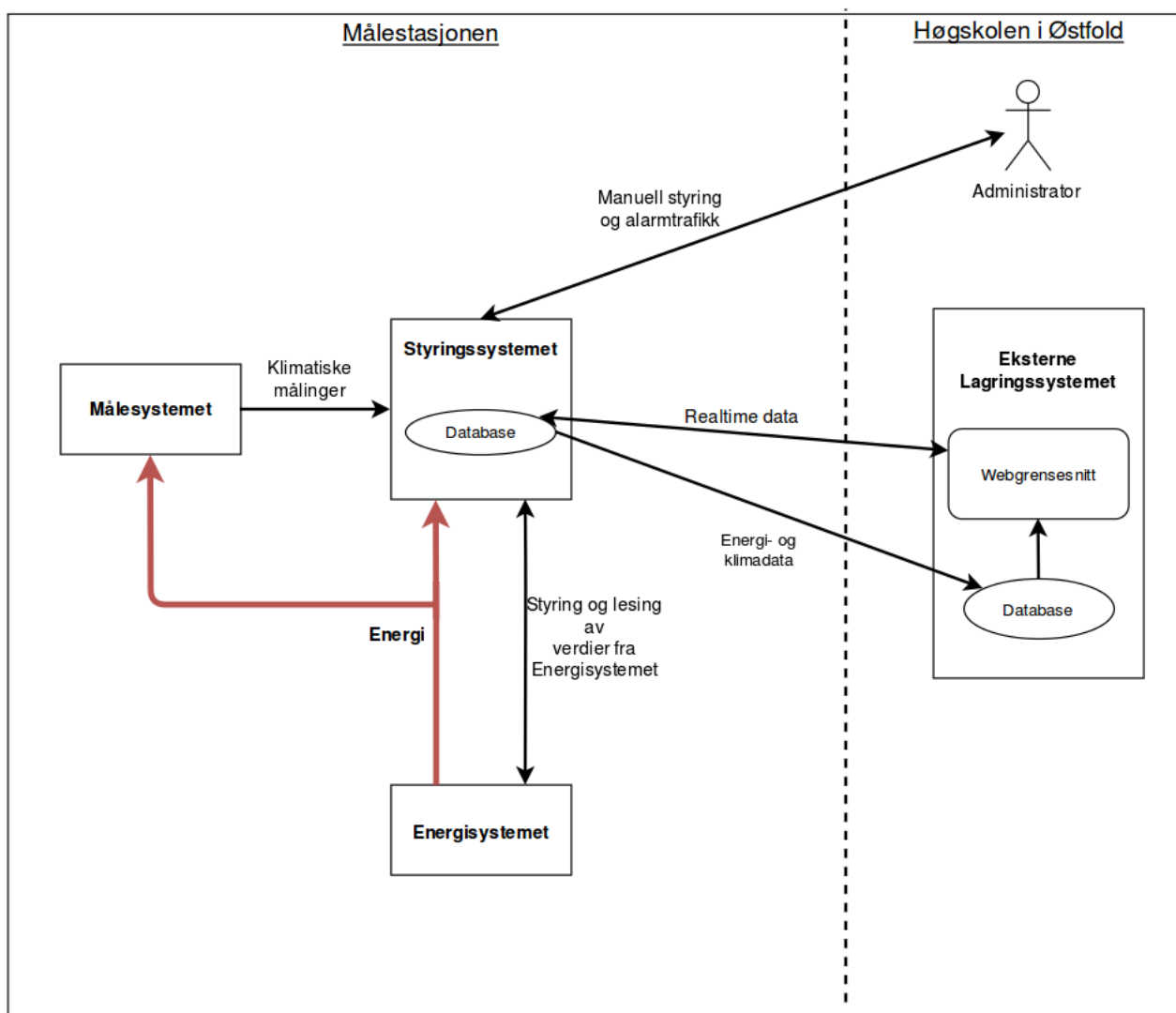
Det skal bygges et styring- og overvåkningssystem av en off-grid, autonom målestasjon med egen energiproduksjon. Målestasjonen består i hovedsak av fire separate undersystemer. Det eksterne lagringssystemet, energisystemet, målesystemet og styringssystemet. Styringssystemet som utgjør kjernen i målestasjonen skal kjøre et Linux operativsystem. Styringssystemet må også ha tilstrekkelig med fysiske tilkoblingsmuligheter som muliggjør styring og lesing av verdier fra enheter i de andre systemene i målestasjonen. Styringssystemet må kunne kobles opp mot Internett og må ha en form for sikker lagring av verdier som måles fra de andre systemene. I tillegg skal Styringssystemet utløse alarmer hvis det oppstår uønskede hendelser eller unormale målinger.

Styringssystemet må også ha en mulighet for manuell styring og konfigurasjon. Verdiene som måles av Styringssystemet skal lagres lokalt, så sendes over Internett til det eksterne lagringssystemet i forhåndsdefinerte tidsintervaller. Det Eksterne Lagringssystemet vil være en server plassert på Høgskolen i Østfold. Slik vil målingene sikres og lagres. Det Eksterne Lagringssystemet vil ha et web-grensesnitt for presentasjon av historiske målinger. Energisystemet skal produsere energi til å dekke behovet til de andre systemene. Energisystemet må overvåkes og styres av Styringssystemet. Det siste systemet, Målesystemet, vil måle vær- og klimaverdier. Styringssystemet lagrer verdier fra Målesystemet og Energisystemet, og sørger for at disse dataene blir overført til det Eksterne Lagringssystemet. Se figur 2.1 for en systemoversikt.

### 2.1 Energisystemet

Energisystemets oppgave er å sørge for at de andre systemene i målestasjonen får tilstrekkelig med energi for å gjennomføre sine prosesser. Oppdragsgiver har planlagt at Energisystemet skal bestå av en vindmølle, solcelle-paneler, brenselcelle og en batteribank. Selv om energisystemet kun eksisterer i målestasjonen for å understøtte målestasjonens hovedprosess, måling av vær og klima, så er utviklingen og driften av energisystemet interessant for oppdragsgiver i seg selv. Prosjektet faller under ”grønn-IT paraplyen” og oppdragsgiver ønsker å teste ut ny miljøvennlig energiteknologi. I tillegg er det ønskelig å teste ut hvor mye energi man klarer å hente ut av et miljøvennlig energisystem. Denne testingen er planlagt å gjøre etter at styring- og overvåkningssystemet er ferdig utviklet og er utenfor dette prosjektets rammer. Det er likevel viktig for den senere testingen at Styringssystemet som bygges får tilstrekkelig med data fra Energisystemet. Det må anskaffes og lagres data som kan fortelle hvor mye energi som produseres, hvordan den produseres, hvor mye som forbrukes totalt og hvor mye energi som forbrukes av de enkelte enhetene i målestasjonen.

### Systemoversikt



Figur 2.1: Et oversiktsbilde av systemet som skal bygges

#### 2.1.1 Solenergi

”Solenergi, strålingsenergi fra Solen. Mengden av solenergi som treffer Jorden i løpet av ett år, er om lag 15 000 ganger større enn hele verdens årlige energiforbruk”.<sup>1</sup> Solenergi kan bli utnyttet på flere måter. Den enkleste formen er direkte oppvarming, dette kan bli brukt til oppvarming av boliger og vann, og for energi til matlaging og tørking. Varmekraftstasjoner kan benytte seg av solenergi for å varme opp væskefylte rør som føres videre til dampturbiner. Solceller konverterer strålinger fra solen til elektrisitet<sup>1</sup>

”Solcelle, lysfølsom halvlederdiode som omdanner lys (sollys) til elektrisk strøm. Solceller benyttes som kraftforsyning til satellitter, romfartøy, fotoutstyr og i stigende utstrekning til generell

<sup>1</sup>Solenergi: <https://snl.no/solenergi>

kraftforsyning basert på solenergi”<sup>2</sup>.

”Fotoelektrisk emisjon eller foto-emisjon oppstår når de enkelte elektroner treffes av lyskvant med tilstrekkelig energi til å frigjøre elektronet”<sup>3</sup>. Solceller baserer seg nettopp på fotovoltaisk effekt.

Mellom to elektroder blir det plassert halvleder materiale. Det vanligste halvleder materialet er silisium, i teorien skal effektiviteten til silisium være på 28%, men den ligger mellom 15% og 24% i praksis. Bølgelengden på det virksomme lyset er kortere enn ca. 1.0 µm. For at effekten ut av solcellene skal bli stor nok må de kombineres og monteres som store paneler<sup>2</sup>.

### 2.1.2 Vindenergi

Vindkraft har blitt utnyttet i minst 3000 år. Frem til det 20.århundre ble vindkraft brukt som mekanisk kraft til vannpumpe eller å male korn. Vindkraften ble etter hvert erstattet med forbrenning av fossilt brensel, da det er en mer stabil kraftkilde. I nyere tider har fokuset på vindkraft økt igjen, men nå er tankene rettet mer mot elektrisk kraft istedenfor mekanisk[Ackermann i inni, 2005, s. 7]. Dane Poul LaCour var den første som bygde en vindturbin som produserte elektrisitet<sup>4</sup>.

**Vindturbineteknologi** Vindenergisystem kan bli delt inn i de som er avhengig av aerodynamisk drag og de som er avhengig av aerodynamisk løft. De med drag har en veldig liten kraft koeffisient. I moderne tider er aerodynamisk løft dominant i vindturbiner. Løftesystemer bruker blader som samhandler med innkommende vind. Løftekraften er et resultat av dragkraften i vindretningen fra bladene som avskjærer luftstrømmen og en kraftkomponent som står vinkelrett på luftmotstanden. Løftekraften er et multiplum av dragkraften og er derfor den relevante drivkraften til rotoren[Ackermann i inni, 2005, s. 21].

Vindturbiner som bruker aerodynamisk løft kan videre bli delt i henhold til retningen til spinnaksen, horisontalaksen og vertikalaksen. Vertikaleakser har den fordelen at de er uavhengig av vindretningen og at girkassen og generatormaskinen kan bli plassert på bakken. En horisontalaksevindturbin består av et tårn og en nacelle som er montert på toppen av tårnet. Nacellen består av girkassen, generatormaskinen og rotoren. Forskjellige mekanismer kan bli brukt for å plassere nacellen i vindretningen, eller ut av retningen ved tilfeller hvor vindstyrken er for stor. På små vindturbiner er det et hale ”ror” som styrer den i riktig retning. På større vindturbiner blir den plassert i riktig posisjon ved hjelp av elektrisk motor som får beskjed fra hale ”roret” hvilken posisjon den skal ha [Ackermann i inni, 2005, s. 21].

### 2.1.3 Ikke-fornybar energi

Ikke-fornybare energikilder er energikilder som er basert på ressurser som finnes i naturen. Disse ressursene er ikke fornybare i et menneskelig perspektiv og kan betraktes som en lagerressurs som kan gå tom<sup>5</sup>. I dag er de viktigste ikke-fornybare energikildene i verden olje, naturgass og kull. Disse energikildene kalles fossile energikilder og er opprinnelig organisk materiale som er fanget i sedimentære bergarter og omdannet i jordskorpa under høyt trykk og høy temperatur. Som

<sup>2</sup>Solcelle: <https://snl.no/solcelle>

<sup>3</sup>Emisjon: <https://snl.no/emisjon/fysikk>

<sup>4</sup>Dane Poul LaCour: <http://windowstoworldhistory.weebly.com/poul-la-cour-pioneered-wind-power-in-denmark.html>

<sup>5</sup>Store norske leksikon: [https://snl.no/ikke-fornybare\\_energikilder](https://snl.no/ikke-fornybare_energikilder)

et sluttprodukt av denne prosessen finner vi i dag råolje, naturgass, oljesand, oljeskifer og kull i jordskorpa<sup>6</sup>.

**Stasjonen** skal stå ute i naturen og oppdragsgiver ønsker derfor at energiproduksjon skal foregå miljøvennlig og støyfritt. Blant ikke-fornybare energikilder som egner seg for dette prosjektet, er strømaggregat den vanligste. Strømaggregat lager imidlertid støy og er ofte ikke miljøvennlige, det er derfor ikke en god løsning i dette prosjektet.

Oppdragsgiver ønsker mer kunnskap om teknologien bak brenselceller. En brenselcelle produsere energi miljøvennlig og uten å lage støy, det er derfor besluttet at en brenselcelle skal benyttes som ikke-fornybar energikilde i prosjektet.

**En brenselcelle** produserer elektrisk energi gjennom en kjemisk reaksjon uten forbrenning<sup>7</sup>. Det finnes mange typer brenselceller, de klassifiseres primært etter hvilke type elektrolytt de bruker. Typen brenselcelle bestemmer blant annet hvilken elektrokjemisk reaksjon som foregår i cella, hvilken katalysator som blir brukt, drifts-temperatur og hva slags drivstoff som brukes<sup>8</sup>.

Brenselceller kan brukes til å drive praktisk talt alle enheter og maskiner som går på batteri. I motsetning til et vanlig batteri som etter hvert vil gå tom, vil en brenselcelle kontinuerlig produsere strøm så lenge den har tilgang på drivstoff. Bærbare datamaskiner, mobiltelefoner og kameraer er eksempel på elektroniske enheter som kan drives av brenselceller<sup>9</sup>.

#### 2.1.4 Batteribank

I prosjektet skal det benyttes en batteribank på 24 volt. Den består av åtte 12 volt batterier av merket Suntek. Batteriene er blybatterier av typen AGM. AGM står for "absorbent glass mat", som betyr at syren absorberes av tynne glassfibermatter. Fordelen med AGM batterier er at de er tette og vedlikeholdsfrie. De har også en lav indre motstand og har evnen til å gi ut høye strømninger når det er nødvendig [Jens Petter Grimstad Andersen, 2015, s.9].

**Hvorfor?** Når systemet produserer mer energi enn det forbruker blir det et energioverskudd, det er ønskelig å lagre dette overskuddet i batteribanken. Det er hensiktsmessig å lagre energioverskuddet til perioder med lav energiproduksjon.

## 2.2 Målesystem

Målesystemet i målestasjonen består av sensorer som måler vær- og klimaverdier. Oppdragsgiver ønsker at det skal måles verdier for vindhastighet, vindretning, temperatur ute, relativ luftfuktighet ute, temperatur inne og relativ luftfuktighet inne. Styringssystemet tolker disse måleverdiene fra Målesystemet over i kjente størrelser innen meteorologi. Styringssystemet skal lagre verdiene lokalt og sende dataene videre til det Eksterne Lagringssystemet i planlagte tidsintervaller. I det Eksterne Lagringssystemet kan dataene brukes til å regne ut nye interessante verdier. Dette kan for eksempel være gjennomsnittsverdier, maksimum- og minimumsverdier. I tillegg til å tolke, lagre og sende verdiene fra Målesystemet, kan det også være interessant å benytte noen av disse

<sup>6</sup>NDLA: <http://ndla.no/nb/node/23025>

<sup>7</sup>CHFCA: <http://www.chfca.ca/education-centre/what-is-a-fuel-cell/>

<sup>8</sup>Energy.gov: <http://energy.gov/eere/fuelcells/types-fuel-cells>

<sup>9</sup>National Geographic: <http://environment.nationalgeographic.com/environment/global-warming/fuel-cell-profile/>

verdiene til å avgjøre hvordan styringen skal foregå. For eksempel kan en temperaturmåling fra Målesystemet benyttes til å avgjøre om varmelementer i utvendige sensorer og annet utstyr skal være på eller av.

Generelt deles målinger opp i to hovedtyper, passive og aktive målinger. Hensikten med passive målinger er å observere naturlig stråling fra jorden. Slike instrumenter blir ofte brukt på værsatellitter, og oversiktsbildene man ser på værmeldingene er ofte et resultat av det. Aktive instrumenter blir brukt der hvor frekvensområdet til den naturlige strålingen er så liten at den ikke kan bli observert. Instrumentet sender ut elektromagnetisk signal, signalet reflekteres og spres av jord og atmosfære. Styrken på signalet som blir sendt tilbake gir informasjon om forholdene<sup>10</sup>.

### 2.2.1 Temperatur

Temperatur er et objektivt mål på hvor varmt eller kaldt et objekt er. Temperatur og varme er ikke det samme, men det er en sammenheng mellom de. Temperatur er et mål på den indre energien til et objekt, og varme er mål på hvordan energi overføres fra et objekt til et annet. Atomene i et materiale beveger seg fortere desto mer varme som blir absorbert, og derfor øker temperaturen til materialet. Ved absolutt null(-273,15) vil det også være null bevegelse i atomene. Temperatur kan bli målt i flere ulike skalaer, i Amerika blir det brukt Fahrenheit, og Celsius blir brukt stort sett i resten av verden. I fysikkens verden er det derimot mer vanlig å bruke Kelvin som benevnelse, den er justert slik at verdien er 0 ved det absolutte nullpunktet<sup>11</sup>.

#### 2.2.1.1 Måling av temperatur

Temperatur måles vanligvis i små trehytter. Det er flere retningslinjer for å få målinger med samme forhold slik at det kan bli mulig å sammenligne målingene. Fargen på huset som er omringet temperatursensoren skal helst være hvit for å reflektere mest mulig solstråling. Posisjonen til huset bør helst være på et åpent område slik at ingen gjenstander kan gi skygge for huset. Det skal også være plassert på kortklippet gress, da plassering på for eksempel asfalt eller myr vil gi helt forskjellige temperaturer da det er mye sol. Hvis huset skal bli plassert på bakken skal sensoren være 2 meter over. Veggene til huset skal ikke være helt tette, slik at huset er godt ventilert samtidig som det er beskyttet mot nedbør<sup>12</sup>.

**Det finnes flere metoder** for måling av temperatur. Det benyttes vanligvis enten et termoelement, en resistance temperature detector (RTD) eller en termistor. Temperatur er den fysiske parameteren som blir målt mest og konvertert til elektronisk form. Flere typer av temperatur transducere responderer til temperatur og produserer korresponderende indikasjoner på endringer i en fysisk karakteristisk som kan bli oppdaget av en elektrisk krets [Thomas L. Floyd, 2014, s. 969].

- **Termoelement:** Termoelement blir laget ved å koble to ulike metaller. En liten spenning, kalt seebeck spenning, blir produsert over kryssningen mellom metallene når det blir oppvarmet. Mengden med spenning som blir produsert er avhengig av hvilket type metall som blir brukt, og er direkte proporsjonal med temperaturen til kryssningen. Termokupler blir brukt i bestemte industrier siden de kan bli brukt til å måle veldig høye temperaturer. [Thomas L. Floyd, 2014, s. 969-970]

<sup>10</sup>Målinger: [http://met.no/Forskning/Fjernmaling/Passive\\_og\\_aktive\\_malinger/](http://met.no/Forskning/Fjernmaling/Passive_og_aktive_malinger/)

<sup>11</sup>Temperatur: <http://physics.about.com/od/glossary/g/temperature.htm>

<sup>12</sup>Måling av temperatur: <http://www.yr.no/artikkel/slik-males-temperaturen-1.8296562>

- **RTD:** Er en motstands-enhet hvor motstanden endres direkte med temperaturen. En RTD er mer lineær enn en termokupel. Generelt blir RTDer brukt til å oppdage temperaturer på to måter. Den første måten er at den blir drevet av en strømkilde, og siden strømmen er konstant så er endringene i spenningene over den, proporsjonal med endringene i motstand med temperatur. Den andre måten er at den er koblet i en 3-lednings brokrets, og broens output-spenning blir brukt til å oppdage endringene i RTDens motstand som er proporsjonal med temperaturen. Temperatur rekkevidden til en RTD er mer begrenset enn termokupel [Thomas L. Floyd, 2014, s. 973-975].
- **Termistor:** Er også en motstands-enhet, laget av semikonduktormateriale som nikkeloksid og kobolt oksid. De fleste motstander i termistorer endrer seg omvendt med temperaturen. Termistorens karakteristikk er mer ulineær enn termokupel og RTD, den er egentlig logaritmisk. Likt som RTD har termistoren ganske begrenset temperatur skala, men den er mer sensitiv enn både RTD og termokupler og er generelt billigere [Thomas L. Floyd, 2014, s. 976-977].

## 2.2.2 Vind

”Vind er luft i horisontal bevegelse og karakteriseres ved vindhastighet (for eksempel knop, m/s eller beaufort) og vindretning”<sup>13</sup>. Vindretning vil si hvilken retning det blåser fra. Det er to måter å angi denne retningen på, kompassretning og i grader. Kompassretninger blir ofte brukt i værvarsler og grader blir brukt i metrologiske tjenester. På vindfart blir benevningene m/s, knop eller km/h. Begrepet vindhastighet er i streng forstand både fart og retning, men blir ofte bare brukt om fart<sup>14</sup>.

### 2.2.2.1 Måling av vind

Vind kan variere mye i både retning og fart på kort tid, så for å kunne si noe fornuftig av målingene blir det kun brukt gjennomsnittsverdier. I følge Meteorologiske institutt er det vanlig å bruke gjennomsnittsverdien i løpet av 10 minutter ved observasjoner av vind. De har også noen retningslinjer som bør følges for plassering av en vindmåler. Den skal plasseres 10 meter over bakken, det skal bli tatt hensyn til vegetasjon og terreng rundt, da dette kan påvirke resultatet. Det er bevegelsene i ”den frie atmosfæren” som skal bli målt<sup>15</sup>.

**To metoder** for å måle vind. Den første og mest tradisjonelle er å måle retning og fart med to instrumenter. Den andre metoden er å måle både retning og fart ved hjelp av ultralyd<sup>15</sup>.

- **Skålkorsanemometeret og vindfløy:** Fart bli målt av et Skålkorsanemometeret som roterer etter hvert som det fanger opp luftas bevegelsesenergi. Instrumentet har skåler som er montert på en aksel. Denne akselen roterer etter hvert som vinden blir fanget opp av skålene. Akselens rotasjonshastighet er proporsjonal med vindhastigheten. Det er et telleverk som

<sup>13</sup>Hva er vind: <https://metlex.met.no/wiki/Vind>

<sup>14</sup>vind: <https://snl.no/vind>

<sup>15</sup>Vindmålere: <https://metlex.met.no/wiki/Vindm\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\accent23a\egroup\spacefactor\accent@spacefactorlere>



mottar en elektrisk puls for hver rotasjon til akselen. Denne telleren gjør disse pulsene om til hastighet<sup>16</sup>.

Retning blir målt av en vindfløy som er montert på en aksling. Posisjonen til akslingen påvises av flere kontaktpunkter. Avlesningsutstyret som registrerer farten er det samme avlesningsutstyret som registrerer hvilken posisjon akslingen har, altså hvilken retning vinden blåser<sup>16</sup>.

Ulempen med dette utstyret er at det blir brukt bevegelige deler som blir slitt og må byttes ut etterhvert. Ved bruk av bevegelige deler reduseres evnen til å oppfatte raske måleendringer, som for eksempel turbulens. Da egner det seg bedre med ultralyd målinger<sup>16</sup>.

- Ultralyd: Det å benytte seg av ultralyd for å måle vinden er den mest moderne metoden. Instrumentet har aksene nord-sør og øst-vest, hvor pulser av ultralyd måles mellom disse. Hastigheten på disse pulsene blir lagret i vektorer, og en vektorsumming gir et resultat for retning og fart, dette gjøres internt i instrumentet. Fordelen med slike instrumenter er at det ikke finnes bevegelige deler, og har derfor lang levetid<sup>17</sup>.

### 2.2.3 Elektrisk strøm

Elektrisk strøm oppstår når elektriske krefter utfører arbeid ved å flytte ladninger. Strøm er også en klart definert størrelse. Enheten for å angi strøm er C/s, Coulomb per sekund, denne enheten har fått navnet ampere(A)[Grimenes, 2010, s. 307,312]. ”Hvis det strømmer en ladning  $q$  igjennom et tverrsnitt i løpet av tida  $t$ , er strømmen  $I$  lik ladning per tid  $I = \frac{q}{t}$ ”.[Grimenes, 2010, s. 312].

#### 2.2.3.1 Måling av strøm

Det er ikke mulig å se strøm eller spenning, derfor brukes det instrumenter for å måle strøm. Det finnes både analoge og digitale måleinstrumenter som kan brukes, i dag er det mest brukt digitale, men ved noen situasjoner kan analoge gi et bedre bilde av hendelsesforløpet<sup>18</sup>.

**Amperemeter** er vanlig å bruke for å måle strøm. Et amperemeter skal kobles i serie, dette fordi amperemeteret har veldig liten indre resistans. Siden strøm velger ”minste motstands vei” kan det fort oppstå kortslutning hvis amperemeteret ikke kobles i serie med en forbruker<sup>18</sup>. Denne indre motstanden er en shuntmotstand. En shuntmotstand har en veldig liten, men nøyaktig motstandsverdi. Siden denne verdien er kjent og nøyaktig er det mulig å finne strømmen i kretsen ved å finne spenningsfallet over shuntmotstanden. Ohms lov sier at  $U = R * I$ , deles spenningsfallet med shuntmotstandens verdi er strømmen funnet<sup>19</sup>.

<sup>16</sup>Skålkorsanemometer: <https://metlex.met.no/wiki/Sk\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\accent23a\egroup\spacefactor\accent@spacefactor\lkoranemometer>

<sup>17</sup>Vindmåler med ultralyd: [https://metlex.met.no/wiki/Vindm\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\accent23a\egroup\spacefactor\accent@spacefactor\lere\\_med\\_ultralyd](https://metlex.met.no/wiki/Vindm\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\accent23a\egroup\spacefactor\accent@spacefactor\lere_med_ultralyd)

<sup>18</sup> Måling av strøm: <http://ndla.no/nb/node/123272>

<sup>19</sup>Beregne shunt: <http://www.datamaskin.biz/Hardware/computer-drives-storage/49366.html>

## 2.2.4 Luftfuktighet

”Luftfuktighet er et mål på hvor mye vann det er i atmosfæren”<sup>20</sup>. Denne målingen er en av flere parametre som benyttes for å forutse været.

### 2.2.4.1 Måling av fuktighet

For å måle luftfuktigheten benyttes et hygrometer. Det finnes en rekke typer hygrometer, hvor hårhygrometer og duggpunkthygrometer er mye benyttet. En mer nøyaktig, men krevende metode er bruken av psykrometer. Det er flere typer enn nevnt, men her vil det begrenses til å kun handle om den generelle funksjonaliteten bak de overnevnte.

**Generelle metoder** for måling av luftfuktighet.

- **Hårhygrometer:** Hygrometer som tar i bruk egenskaper til hår ved ulik fuktighet. Det blir ofte brukt en bunt hestehår<sup>21</sup>. Hårene vil ved tørt vær trekke seg sammen, og ved vått vær strekke seg ut. Hårene tilknyttes en visermekanisme og en skala for å indikere hva luftfuktigheten er<sup>20</sup>.
- **Psykrometer:** Hygrometermetode som utnytter varmetapet som oppstår ved fordampning og at fordampningen øker når relativ fuktighet blir mindre. Hygrometeret er bestående av to termometre, et som er tørt og et som holdes fuktig med destillert vann. Luftfuktigheten bestemmes ut i fra en tabell<sup>21</sup>.
- **Duggpunkthygrometer:** ”Duggpunkthygrometeret benyttes særlig ved lave temperaturer. Man bestemmer den temperatur som luften må kjøles til for at vanddampen skal fortettes; ved å slå opp i tabell sammenholdt med den ordinære lufttemperatur kan man så bestemme luftfuktigheten”<sup>21</sup>.

”I tekniske måleinstrumenter utnyttes endring i stoffers kjemiske og elektriske egenskaper for å kunne måle luftfuktigheten. Det er også mulig å ta nytte av diffusjonen til vanddamp og dens absorpsjonsspektrum for å bestemme luftfuktigheten”<sup>21</sup>.

### 2.2.5 Krav til sensorer

Det er viktig at de sensorene som skal benyttes ikke trenger vedlikehold og at de er operative til enhver tid. Av dette stilles det krav til en viss robusthet avhengig av plassering til sensoren. Eksempelvis vil en sensor plassert innendørs ikke ha de samme krav som en sensor som skal plasseres utendørs.

I Hessdalen kan det tidvis være ekstreme værforhold, spesielt med tanke på kulde. På grunn av dette har det blitt satt en grense for hvor lave temperaturer de ulike sensorene skal kunne håndtere. Under drift kreves det at sensorene er funksjonelle ned til  $-40^{\circ}\text{C}$ . Dette er av stor betydning for valg av sensorer som skal benyttes. For sensorer med bevegelige deler vil også være behov for å ha et varmeelement som kan beskytte mot is. Når varmeelementer skal inkluderes i systemet økes energibehovet og systemet må dimensjoneres deretter. For sensorer med varmeelement vil det være viktig at disse kan styres, i en form for av- og på-styring. Dette for å konservere energi ved høyere temperaturer hvor varme er overflødig. Alle sensorer må være kompatible med et 24V

<sup>20</sup>Luftfuktighet: <http://met.no/Hygrometer.9UFRvS0R.ips>

<sup>21</sup>Hygrometer: <https://snl.no/hygrometer>

system. Det vil si at sensorer som krever høyere spenning ikke vil bli vurdert. Det er også viktig at output fra sensor er mulig å benytte i systemet, eksempelvis et analogt output signal på en gitt spenningsrange.

Det er et krav at sensorer skal være stabile. Med stabil menes at de er funksjonelle til enhver tid med tilnærmet null vedlikehold, i tillegg til at måledata er korrekte og mest mulig nøyaktig. Nøyaktigheten på sensorene vil variere basert på hva som skal måles, men det er satt noen grenser som det skal tas hensyn til ved innkjøp. Det tas utgangspunkt i Meteorologisk institutt sine målestandarder <sup>22</sup>.

- For vindhastighet er det m/s, meter per sekund, som er måleenheten. Her er minimums måleområde fra 0m/s til 50m/s med en nøyaktigheten tilnærmet  $\pm 0,5$  m/s.
- For vindretning så er det °(grader) som skal benyttes. Her er måleområdet 0° til 360° med en nøyaktighet på  $\pm 5^\circ$ .
- For temperatur er det grader Celsius (°C) som skal måles i minimums måleområde -40°C til +35°C. Nøyaktigheten vil variere avhengig av temperaturspranget, men fra oppdragsgivers synspunkt er det nok med en nøyaktighet på  $\pm 0,5^\circ\text{C}$ . Dette gjelder for både innendørs og utendørs sensor.
- For relativ fuktighet (RH) er det måling i prosent (%), fra 0 til 100%. Nøyaktigheten skal være innenfor  $\pm 3\%$  innendørs og utendørs.

Hvis det skulle vise seg å være vanskelig å anskaffe sensorer som oppfyller de ovennevnte kravene kan det i noen tilfeller være aktuelt med andre typer eller modeller. Hvis dette skulle være tilfellet vil det bli avklart med oppdragsgiver for å sikre at ønsker og krav er tilfredsstilt fra alle hold.

Utover dette vil pris også bli vurdert, selv om det spiller en mindre rolle er det en faktor som må tas hensyn til. Oppdragsgiver vil være den som har siste ord, og vil kunne velge mellom flere ulike sensorer i forskjellige prisklasser, som vil bli presentert som aktuelle.

## 2.3 Styringsystemet

Et styringssystem er et system med enheter eller et sett med enheter som behandler, kommanderer, dirigerer eller regulerer oppførelsen til andre enheter eller systemer for å få det ønskede utfallet. Med andre ord kan et styringssystem blir definert slik, et styringssystem er et system, som kontrollerer andre systemer<sup>23</sup>.

På engelsk blir begrepet "control" benyttet når det er snakk om å styre eller regulere et system. På norsk er det vanlig å bruke begrepet "styring" ved slike prosesser. En mer eksakt beskrivelse av styring er å påvirke en prosess uten aktiv bruk av tilbakekobling. Typisk for slike systemer er at det er mange digitale signaler, av/på signaler. Dette kan være systemer som lysstyring, produksjonslinjer og lignede systemer.

Begrepet regulering benytter seg aktivt av tilbakekobling for å styre prosessen. Dette er nødvendig for å få et bedre resultat og eliminere feil i et system. Systemer som er avhengig av å måle tilstander benytter seg av dette, for eksempel temperaturregulering, invertert pendel og mange flere.

<sup>22</sup>Standarder fra met.no:[http://met.no/KRAV+VED+ETABLERING+AV+AUTOMATISKE+V\OT1\AERSTASJONER+I+NORGE.b7C\\_w7zQ1P.ips#FFDD](http://met.no/KRAV+VED+ETABLERING+AV+AUTOMATISKE+V\OT1\AERSTASJONER+I+NORGE.b7C_w7zQ1P.ips#FFDD)

<sup>23</sup>Styringssystem: <http://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/>

### 2.3.1 Hva skal styres?

Dette systemet består av flere reguleringsoppgaver. Varmeelementene til kameraene og vindsensorene skal slås på når det nærmer seg kuldegrader ute, og elementene skal slås av når temperaturen stiger igjen. For at dette skal fungere automatisk må det benyttes aktiv tilbakekobling for å at systemet skal kunne ta korrekte avgjørelser. Brenselcellen i dette systemet skal fungere som en "sikring" for energisystemet. Ved noen anledninger kan det hende systemet ikke klarer å produsere nok energi og spenningen på batteribanken nærmer seg et kritisk lavt nivå. Da må brenselcellen slå seg på for å forsyne systemet med energi. For at dette skal kunne fungere må det bli benyttet aktiv tilbakekobling for å vite spenningen på batteribanken til enhver tid. Data skal bli sendt over Internett til det eksterne lagringssystemet, stasjonen er derfor avhengig av å kunne kontakte omverden via en ruter. Denne ruter kan det være greit å ha kontroll over, slik at systemet selv sjekker om den fungerer slik den skal, og eventuelt restarter den ved behov.

Styringsenheten er hjernen til hele systemet, dersom den slutter å fungere vil store deler av systemet falle sammen. Det må derfor være mulig å kontrollere at den er oppe og går som den skal. Oppdragsgiver hadde et ønske om å benytte seg av en form for "Watchdog timer" til å håndtere denne oppgaven. En Watchdog timer er et stykke hardware som blir brukt for å automatisk detektere programvareuregelmessigheter og resette prosessoren hvis det oppstår<sup>24</sup>.

### 2.3.2 Hvordan kan et styringssystem overstyres?

Noen ganger er det behov for å overstyre innstillingene som er satt i et styringssystem. Det er flere måter dette kan bli utført på, den enkleste måten er å fysisk skru av og på enheter, men dette skal være en autonom stasjon, så det er ikke aktuelt for dette prosjektet. Det finnes mer elegante løsninger som en web-applikasjon eller en mobil-applikasjon, men det var ikke et ønske fra oppdragsgiver. De ville ha en enkel og mer sikker løsning. Det som gjenstår da er fjerntilkobling til systemet. Secure Shell (SSH), også kjent som Secure Socket Shell, er både en applikasjon og en protokoll som benyttes for å opprette en sikker tilkobling til en fjernstyrt datamaskin. Fordelen med SSH er at kommunikasjonen mellom klienten og serveren er kryptert og passordbeskyttet. Det er også mulig å autentiserer koblingen via digitale sertifikater for å unngå uønskede pålogginger<sup>25</sup>.

### 2.3.3 Styringsenhet

Er intelligensen i systemet, det er her beregninger blir utført for å bestemme pådraget eller påvirkningen til prosessene. En styringsenhet må minst ha muligheten for én analog inngang og én analog utgang og den må ha muligheten til å endre referansepunktet. En slik styringsenhet kan bestå av elektronikk, mekanikk eller være et dataprogram[Roppestad, 2015, S. 1].

### 2.3.4 Krav til styringsenhet for dette prosjektet

Oppdragsgiver har et ønske om et Linux basert system, så en elektronisk eller mekanisk løsning faller bort og da står vi igjen med et dataprogram. Videre blir det stilt krav om at stasjonen skal kunne måle flere verdier, så det er et behov for flere digitale og analoge innganger. Systemet skal også kunne påvirke flere prosesser, for det meste av/på-påvirkning, så det blir stilt krav til at enheten må ha digitale utganger. Enheten skal kommunisere med en ruter, så den må ha et

<sup>24</sup>Watchdogtimer: <http://www.embedded.com/electronics-blogs/beginner-s-corner/4023849/Introduction-to-Watchdog-Timers>

<sup>25</sup>Overstyring av styringssystem: <http://searchsecurity.techtarget.com/definition/Secure-Shell>

nettverkskort eller mulighet for adapter. Det er også oppdragsgivers ønske at den kan bruke en "Watchdog timer". Siden dette er en off-grid stasjon skal også strømforbruket til enheten tas med i betraktning.

### 2.3.5 Alternativer

Mikrokontrollere er et mulig valg til bruk som en styringsenhet, men siden oppdragsgiver ønsker et Linux basert system vil ikke en mikrokontroller ha tilstrekkelig ressurser til å kjøre et Linux operativssystem<sup>26</sup>. En stasjonær eller bærbar PC vil ha ressurser nok til å kunne kjøre Linux. Den vil også støtte alle de andre kravene, bortsett fra at den må ha en tilleggsmodul for å kunne behandle analoge signaler. En annen ulempe er den vil ha et ganske høyt strømforbruk. Teknologien er i stadig utvikling og datamaskiner blir stadig billigere, kraftigere og mindre arealkrevende. Dette har åpnet opp for en ny type datamaskiner som har ressurser nok til å kjøre et komplett operativssystem, men som ikke krever særlig større areal enn et kredittkort. Eksempel på slike datamaskiner er Raspberry Pi, CHIP og Odroid C2 <sup>27</sup> <sup>28</sup>. Slike datamaskiner støtter nesten alle kravene, med unntak av kravet om en analog/digital konverter. Dette er likevel ikke noe problem da slike konvertere kan kjøpes som tilleggsmoduler og monteres på maskinene.

## 2.4 Det Eksterne Lagringssystemet

Det eksterne Lagringssystemet skal være lokalisert på Høgskolen i Østfold, og skal motta data fra målestasjonen i planlagte tidsintervaller. Det eksterne lagringssystemet skal lagre all historisk data som mottas, mens lagringen av data i målestasjonen vil slettes med jevne mellomrom. Det er derfor viktig at løsningen som bygges sikrer at dataene faktisk er overført og mottatt før de eventuelt slettes internt. I det eksterne lagringssystemet skal det bygges et webgrensesnitt som presenterer de historiske dataene. Det er et ønske fra oppdragsgiver at selve overføringen skjer med en times tidsintervall. Derfor vil dataene som er lagret og som presenteres i webgrensesnittet ideelt sett ikke være eldre enn én time. Det kan være aktuelt å benytte kortere tidsintervall. Det er også ytret et ønske fra oppdragsgiver at webgrensesnittet har en egen underside som henter data direkte fra målestasjonen. Denne undersiden kan benyttes for å se om målestasjonen er oppe og kjører.

### 2.4.1 Mulige løsninger

Det er fra Høgskolen i Østfold levert en ekstern server som skal benyttes til å bygge det eksterne lagringssystemet. Dette er en virtuell instans som kjører på Høgskolen sin serverpark. Maskinen kjører Debian GNU/Linux og ble levert fra Høgskolen ferdig installert med MySQL og Apache. Apache var i november 2015 verdens mest brukte webserver<sup>29</sup>. MySQL er et databasehåndteringssysteme for relasjonsdatabaser<sup>30</sup>. En mulig løsning for overføring av data

<sup>26</sup>Sammenligning mikrokontroller: <http://www.embedded.com/design/operating-systems/4425751/Comparing-microcontroller-real-time-operating-systems>

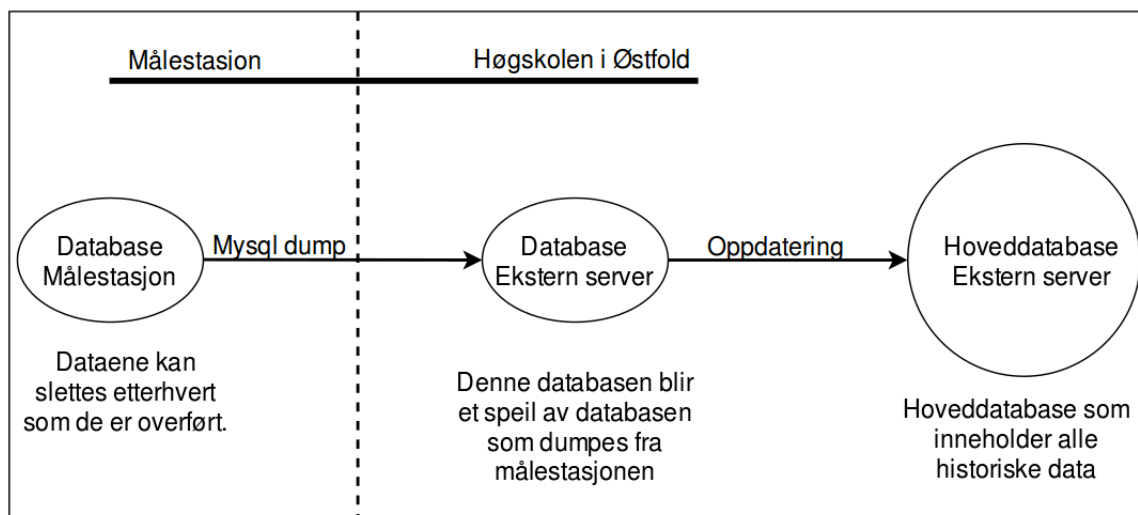
<sup>27</sup>CHIP: <http://www.tek.no/artikler/hva-med-en-knottliten-datamaskin-til-under-70-kroner/186452>

<sup>28</sup>Odroid C2: <http://www.tek.no/artikler/her-er-en-ny-og-kraftig-raspberry-pi-konkurrent/276690>

<sup>29</sup>Web survey: <http://news.netcraft.com/archives/2015/11/16/november-2015-web-server-survey.html>

<sup>30</sup>What is mysql: <http://searchenterpriselinix.techtarget.com/definition/MySQL>

kunne vært å sende TCP-pakker fra målestasjonen til den eksterne serveren. Siden både målestasjonen og den eksterne serveren skal ha en databaseløsning, er det muligens enklere og sikrere å sørge for at målingene lagres internt i målestasjonens database først. Så kan disse dataene med jevne mellom dumpes direkte fra målestasjonens database og over i en database på den eksterne serveren. Database på den eksterne serveren er da en kopi av databasen i målestasjonen. Den eksterne serveren kan deretter ta dataene over fra kopi-databasen og oppdatere hoveddatabasen. Figur 2.2 viser hvordan dette kan gjøres.



Figur 2.2: Et oversiktsbilde av dataoverføring

Webgrensesnittet kan bygges direkte mot hoveddatabasen i den eksterne serveren. Her kan for eksempel programmeringsspråket PHP benyttes for å levere data fra databasen til webgrensesnittet. I tillegg til HTML og CSS, kan også SVG og D3.js benyttes for å fremstille dataene i grafiske animasjoner i nettleseren. SVG, scalable vector graphics, er et XML-basert bildeformat.<sup>31</sup> D3.js er et javascript-bibliotek som benytter HTML, CSS og SVG slik at datasett kan fremvises med grafiske animasjoner<sup>32</sup>.

## 2.5 Lignende prosjekter

For å få et bedre grunnlag til å kunne utføre dette prosjektet er det blitt gjort research på å finne andre prosjekter som er lignende. Skulle det være informasjon og erfaringer gjort tidligere som er relevant for dette prosjektet, vil dette være av interesse og lignende løsninger kan kanskje bli benyttet.

### 2.5.1 Prosjekt Hessdalen - Værstasjoner

Dette prosjektet er en tidligere bacheloroppgave gjennomført våren 2014 ved Høgskolen i Østfold. Kort oppsummert handlet prosjektet om å lage to værstasjoner for måling av temperatur, luftfuktighet, lufttrykk, vindhastighet og vindretning[Mikael Johansen Grimstad, 2014].

<sup>31</sup>SVG: [https://no.wikibooks.org/wiki/Grafiske\\_filformat/SVG](https://no.wikibooks.org/wiki/Grafiske_filformat/SVG)

<sup>32</sup>D3.js: <https://d3js.org>

I prosjektet ble følgende utstyr benyttet for målinger og kontroll: [Mikael Johansen Grimstad, 2014, side 30-33]

- Ethernet 2.1 - Mikrokontroller
- SHT10 - Temperatur og luftfuktighetssensor
- BMP180 - Lufttrykksensor
- WG2/O50 - Vindhastighetsensor
- WRG2/O50 - Vindretning sensor

Den største forskjellen fra dette prosjektet og det ovennevnte er at nå vil ikke målestasjonen være tilkoblet strømmettet, men driftes av egenprodusert energi. Dette byr på ekstra utfordringer da forbruk av hver enhet må tas hensyn til. I tillegg vil det bli benyttet en liten datamaskin i stedet for en mikrokontroller.

Når det kommer til målingene ligner dette systemet, og det eneste som uteblir i dette prosjektet kontra det andre er måling av lufttrykk. Derimot så skal det måles energiproduksjon og forbruk.

Ved valg av sensorer vil det være naturlig å ta med både oppdragsgivers erfaringer samt erfaringen til studentene fra værstasjonsprosjektet inn i vurderingen.

### 2.5.2 Complete DIY Raspberry Pi Weather Station with Software

Er et hobbyprosjekt som går ut på å bruke Raspberry Pi som en intelligent enhet i en værstasjon. Hovedkonseptet i det prosjektet ligner på oppgaven til dette bachelorprosjektet. De skal lese av flere ulike sensorer og representere dette på en webside. Forskjellen er at det ikke blir utført noen styring av enheter, og at mye av utstyret de benytter seg av er "hobby" preget, og kan derfor ikke bli benyttet i dette prosjektet. Det er derimot mye inspirasjon å hente når det kommer til kommunikasjon med enheter og at noe slikt er mulig med en mikro datamaskin<sup>33</sup>.

### 2.5.3 Low cost Arduino/Android-based Energy-Efficient Home Automation System with Smart Task Scheduling

Er en artikkel som tar for seg hvordan en Arduino kan bli brukt til å automatisere hjemmet. Her vil Arduino fungere som en styringsenhet, den vil utføre forskjellige handlinger etter hvordan eier påvirker den. Dette viser hvordan en slik enhet kan bli brukt til å påvirke andre enheter i et system samtidig som det viser at slike små mikrokontrollere kan bli brukt til veldig mye og hvor kraftfulle slike enheter er [Baraka i inni, 2013].

---

<sup>33</sup>RPI værstasjon: <http://www.instructables.com/id/Complete-DIY-Raspberry-Pi-Weather-Station-with-Software-ALLSTEPS>





## Kapittel 3

# Planlegging

Dette kapitlet vil ta for seg hvilke valg som har blitt gjort og hvilke enheter som har blitt brukt i de ulike delene av systemet. Det som blir utført og hvorfor akkurat et spesifikt element ble valgt vil bli begrunnet, samt en detaljert beskrivelse av hver enkelt enhet. Fordeler og ulemper vil bli diskutert der det er mulig. Valgene og beslutningene vi gjør i dette kapitlet vil bli utgangspunktet for implementeringen og videre arbeid. Figur 3.1 er en detaljert modell av systemet som viser alle enheter og hvilken type kommunikasjon som skal benyttes.

### 3.1 Energisystemet

I seksjonen om Energisystemet 2.1 i analysekapitlet blir oppgavene til systemet og oppdragsgivers ønske om at energisystemet skal bestå av vindmølle, solcellepanel, brenselcelle og en batteribank forklart. På bakgrunn av dette har prosjektet valgt å gå for enheter som oppfyller oppdragsgivers ønske og som har en viss form for intelligens slik at det er mulig å hente ut informasjon som kan brukes i styringssystemet eller logges i en database. Videre i denne seksjonen vil de valgte enheten bli grundig beskrevet. Det vil også bli utredet hvordan det vil være mulig å kommunisere med de ulike enhetene.

#### 3.1.1 Chargekontrollere

En chargekontroller er en enhet som er tilknyttet batteriladning. Dette er enheten som styrer hvor mye et batteri kan lades og hvor mye strøm som kan trekkes fra batteriet. Som kontroller skal enheten også forhindre at batteriet overlades, samtidig som de ikke skal gå tomme. Dette bidrar til å øke levetiden på batteriene som er i bruk.

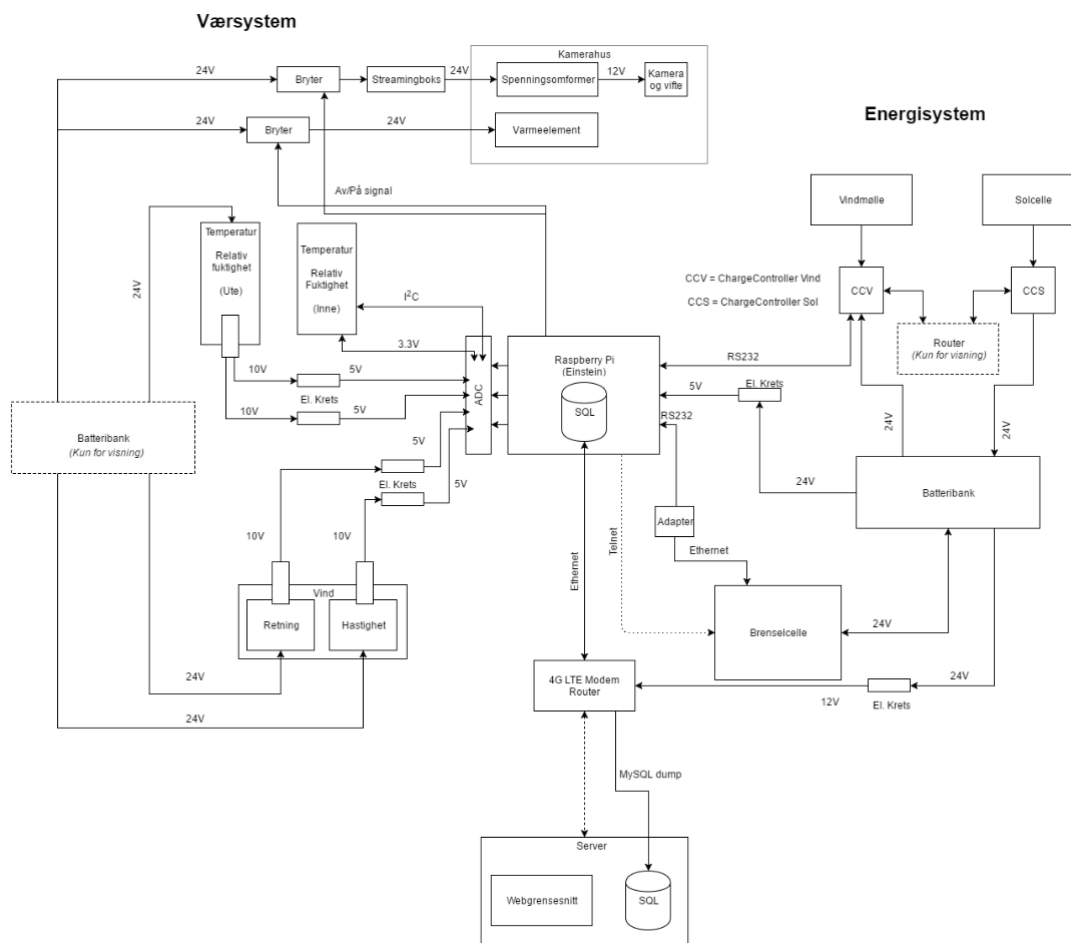
I dette prosjektet vil det bli benyttet to chargekontrollere levert av det amerikanske selskapet Morningstar<sup>1</sup>, hvor en benyttes for solcellepanel og en for vindkraft. Disse er valgt av oppdragsgiver, så andre alternativer er utelukket.

##### 3.1.1.1 TriStar MPPT-60

Dette er en solenergi-basert chargekontroller, og prosessen for batteriladning er optimalisert for å forlenge batterilevetiden og ha god systemytelse. Selvdiagnostikk og feilhåndtering på det elektriske forhindrer eventuelle skader som kan oppstå under installasjon eller ved systemsvikt.

---

<sup>1</sup>Morningstar: <http://www.morningstarcorp.com/company/>



Figur 3.1: Detaljert system modell

Kontrolleren har i tillegg åtte brytere for endring av innstillinger, flere kommunikasjons porter og terminal for å kunne måle batteritemperatur og spenning<sup>2</sup>.

Det er to andre modeller av MPPT-serien, MPPT-30 og MPPT-45. I forhold til hverandre så ligger ulikhetene i de tekniske spesifikasjonene, hvor muligheter for tilkoblinger og hvilke kommunikasjonsprotokoller som kan bli benyttet er noen av forskjellene. Modellen som blir benyttet i dette prosjektet, MPPT-60, er den modellen med flest alternativer. Den håndterer spenninger på 12, 24, 36 eller 48 Volt DC. Alternativene for kommunikasjonsporter er RS-232, EIA-485, MeterBus<sup>TM</sup> og Ethernet<sup>2</sup>.

### 3.1.1.2 TriStar-45

For vindproduksjonen i dette prosjektet blir TriStar-45 brukt for å styre batteriladningen. TriStar-45 er en tre-funksjonell kontrollert som forsørger pålitelig batteriladning, load control og diversion

<sup>2</sup> TriStar MPPT Manual, s.58-59: <http://www.morningstarcorp.com/wp-content/uploads/2014/02/150V-TS-MPPT-Operators-Manual.pdf>

<sup>3</sup>Figur av chargekontroller: [www.morningstarcorp.com/wp-content/uploads/2013/11/TS-MPPT\\_Angled-60-F\\_retouch\\_feature.png](http://www.morningstarcorp.com/wp-content/uploads/2013/11/TS-MPPT_Angled-60-F_retouch_feature.png)



Figur 3.2: Chargekontroller for sol: TriStar MPPT-60 <sup>3</sup>

regulation. Akkurat som TriStar MPPT-60, vist i figur 3.2, er den optimalisert for å forlenge batterilevetiden og ha god systemytelse, men det er en lavere modell og den har bare muligheten for kommunikasjon over RS-232. Det er også satt opp med diversion regulation for å beskytte batteriet mot overladning. Overskuddsenergi blir avledet fra batteriet og til en alternativ DC resistance load, som brenner av overskuddsenergien.

### 3.1.1.3 Styrker og svakheter ved chargekontrollerne

Den største fordelen er at begge enhetene lader batteriet samtidig som de beskytter batteriet mot overspenning. En annen stor fordel er at det har blitt utviklet modbus register for begge enhetene. Dette forenkler kommunikasjonen til enhetene. Det er ikke mange ulemper med disse enhetene da de er enkle og robuste, men det at den ene bruker seriell kommunikasjon er et lite minus. Det hadde vært mye bedre med modbus over TCP/IP på grunn av hastighet.

## 3.1.2 Brenselcelle

En brenselcelle er en enhet som produserer energi ved å forbrenne et drivstoff. Den vil være nyttig når det ikke produseres tilstrekkelig med sol- og vindenergi. Brenselcellen måler spenningen i batteriet og lader batteriet når spenningen er ved en gitt nedre grense.

I seksjonen om energisystemet 2.1 i analysekapittelet, er det nevnt at oppdragsgiver ønsker å benytte en brenselcelle. Siden dette er teknologi hverken medlemmene av prosjektgruppen eller oppdragsgiver har brukt før har det vært viktig å gjøre en del research på temaet. Under søkeprosessen var det kun en produsent som dukket opp som aktuell, EFOY.

EFOY klassifiserer sine brenselceller innenfor to kategorier; privat og profesjonell anvendelse <sup>4</sup>. Etter noe informasjonsinnhenting på de ulike kategoriene ble det fort klart at det kun var

<sup>4</sup>EFOY: <http://www.efoy.com/>

de profesjonelle modellene som kunne være aktuell. Dette skyldtes at de som falt innenfor privatkategorien ikke ville være robust nok for dette prosjektet.



Figur 3.3: EFOY Pro 2400 Duo <sup>5</sup>

Under den andre kategorien derimot var det 4 modeller å velge mellom; EFOY Pro 800/800 Duo og EFOY Pro 2400/2400 Duo. Duo-modellene har mulighet til å koble til flere kanner med drivstoff. De ulike alternativene ble presentert ovenfor oppdragsgiver, og av disse ble EFOY Pro 2400 Duo som vist i figur3.3 valgt. Mye på grunn av at 2400 versjonen<sup>6</sup> kunne produsere over det dobbelte av energi i motsetning til 800 versjonen<sup>7</sup>. I tillegg var det mulighet for å koble til fire stykk 28 liters kanner med metanol på PRO utgaven.

Leverandør og distributør av brenselcellen var Power Controls AS, lokalisert i Moss. De tilbød god service under bestillingsprosessen og har vært veldig hjelpsomme underveis ellers i prosessen.

### 3.1.2.1 EFOY Pro 2400 Duo

Denne brenselcellen bruker metanol som drivstoff og har mulighet for fjernstyring og overvåkning over RS-232 protokollen. Den gir ut 110W og kan brukes til 12V eller 24V batteri. Brenselcellen er robust og kan operere under krevende klimatiske forhold hvis den er godt isolert. Det kan tilkobles 4 metanolkanner på 28 liter hver, det gjør at brenselcellen kan operere over lengre perioder uten tilsyn<sup>8</sup>.

<sup>5</sup>EFOY Pro 2400 Duo: <http://www.powercontrols.no/systemer>

<sup>6</sup>EFOY versjoner: <http://www.efoy-pro.com/page/efoy-pro-24002400-duo#page-tabs>

<sup>7</sup>EFOY versjoner: <http://www.efoy-pro.com/page/efoy-pro-800800-duo#page-tabs>

<sup>8</sup>EFOY manual: [http://www.efoy-pro.com/sites/default/files/download/sfc\\_user\\_manual\\_efoy\\_pro\\_800\\_2400\\_en\\_v2\\_web.pdf](http://www.efoy-pro.com/sites/default/files/download/sfc_user_manual_efoy_pro_800_2400_en_v2_web.pdf)

### 3.1.2.2 Ekstraustyr for Brenselcellen

For å få mest mulig nytte av brenselcellen er det noe tilhørende utstyr som er blitt kjøpt inn. Dette inkluderer utstyr som benyttes for tilkobling mot en datamaskin, bruk av drivstoff og en isolasjonkasse for å kunne håndtere de lave temperaturene.

**Interface Adapter - A1** Denne adapteren, vist i figur 3.4, brukes til å koble brenselcella til en datamaskin. I den ene enden har den en RJ45-inngang og i den andre enden har den en RS-232 plugg. RS-232 pluggen kobles til datamaskinen, så kobles en Cat 5 kabel mellom brenselcellen og adapteren.



Figur 3.4: Interface Adapter <sup>9</sup>

**EFOY DuoCartSwitch** DuoCartSwitch, vist i figur 3.5 blir brukt til å koble seg på 2 drivstoffkanner fra 1 drivstoffinngang på brenselcellen. Den bytter drivstoffkanne automatisk når kannen den bruker går tom. Det benyttes 2 av DuoCartSwitch i dette prosjektet, de kobles på hver sin drivstoffinngang på brenselcellen. De er nødvendig for å kunne koble til 4 metanolkanner av gangen, uten dem er det bare mulig å koble til 2 drivstoffkanner.

DuoCartSwitch må kobles til cella med ethernetkabel slik at den kan kommunisere og få beskjed når det skal byttes mellom drivstoffkannene. Det er bare 2 Ethernet-innganger på brenselcellen, dette er ikke nok til to DuoCartSwitch og et interface. Derfor er en "port-doppler" inkludert med hver DuoCartSwitch.

**Zarges isolasjonkasse** Dette er en isolasjonkasse som brenselcellen og drivstoffkannene skal monteres i. Den inneholder to 12V transportbatterier koblet i serie og én vifte for ventilasjon. Størrelsen på kassen er 170x80x70cm og den har plass til fire metanolkanner. For at alle kannene skal kunne tømmes automatisk kreves det to av DuoCartSwitch som nevnt ovenfor. På denne måten vil brenselcellen være mest mulig autonom.

<sup>9</sup>Interface manual: [http://www.efoy-pro.com/sites/default/files/web-um-ia1-151901091-en-de-130610\\_0.pdf](http://www.efoy-pro.com/sites/default/files/web-um-ia1-151901091-en-de-130610_0.pdf)

<sup>10</sup>DuoCartSwitch: <http://www.fta.be/ftaupload/DuocartSwitchCutL.jpg>

Figur 3.5: DuoCartSwitch <sup>10</sup>

Figur 3.6: Zarges isolasjonskasse plassert i vognen

### 3.1.2.3 Styrker og svakheter

En åpenbar fordel med denne brenselcellen er muligheten for kommunikasjon. Den kan enkelt kobles til via RS-232 og styres via en datamaskin. Den er også i stor grad autonom og benytter batterispenningen som grunnlag for å ta avgjørelser for styring. Den har også innebygde temperatursensorer som den bruker til å avgjøre når den skal slå på "anti freeze" funksjonen. Den kan derfor operere lenge uten menneskelig interaksjon. I tillegg er det et miljøvennlig alternativ i forhold til mange andre løsninger. Fordi stasjonen skal stå i naturen er det også bra at den avgir

lite støy.

Den største svakheten med brenselcellen er prisen, det er mange alternative strømgeneratorer som opererer for lavere kostnad. Den har også ganske få driftstimer i forhold til pris. Klimatiske forhold er også i større grad en utfordring fordi brenselcellen er væskebasert og ikke operer under frysepunktet. Kuldeproblemet medfører at det må brukes mer penger på ekstrautstyr for isolasjon og ventilasjon.

## 3.2 Målesystemet

Dette systemet vil bestå av en rekke ulike sensorer som det blir utdypet i analysekapittelet i seksjonen om målesystemet 2.2. Der blir det også konkludert med hvordan dette systemet skal fungere ved at det logger ved gitte tidspunkter, og sender noen verdier videre til styringssystemet. Ved valg av sensorer er det viktig at de ulike kravene som blir stilt i seksjonen om sensorkrav 2.2.5 overholdes.

Videre i denne seksjonen kommer det en beskrivelse av hvilke enheter som blir brukt samt en begrunnelse, med bakgrunn av analysekapittelet, hvorfor de ulike sensorene ble valgt.

### 3.2.1 Vindsensorer

Det ble vurdert flere alternativer til vindsensorer, hvor blant annet noen benyttet ultralyd slik det blir drøftet i analysekapittelet 2.2.2.1. Det ble fort klart at selv om disse benytter den mest moderne metoden og trolig har lengst levetid, så ble det for dyrt.

Etter å ha lest rapporten til gruppen, nevnt i seksjon 2.5.1 og snakket med oppdragsgiver, så ble det tatt en beslutning om at vindsensorer av samme type skulle benyttes. Dette blant annet pga. oppdragsgivers erfaring med disse fra før, samtidig så oppfylte de kravene som blir nevnt i underkapittelet 2.2.5 for vindmålinger. Sensorene hadde også innebygde varmeelementer som gjør at de kan benyttes i kaldt klima uten å bli ødelagt.

#### 3.2.1.1 Wind Speed Transmitter - Compact



Figur 3.7: Wind Speed Transmitter <sup>11</sup>

Dette er en sensor for å måle vindhastighet. Sensoren fungerer slik at toppen, sett i figur 3.7 roteres av vinden og en opto-elektronisk hastighetskanner produserer en frekvens som deretter transformeres til et analogt signal av en integrert omformer for målingene<sup>12</sup>.

Det er flere tilgjengelige modeller av denne vindsensoren; WG2/O-10, WG2/O-20 og opp til WG2/O-50. De eneste forskjellene på disse er den elektriske outputen. Her var det ønskelig med O-20 modellen, men da leverandør kun hadde O-10 og O-50 var valget enkelt. Sistnevnte hadde 4-20mA output, og av den grunn ble den uaktuell.

Modellen valgt for dette prosjektet er av typen WG2/O-10. Dette på grunn av at den har elektriske output er på 0-10V, som er lettere å forholde seg til enn de andre. Selv om det er ønskelig med 0-5V, som O-20 modellen tilbød, kan det i dette tilfellet benyttes en spenningsdeler. Sensoren har en driftsspenning på 24Volt og har en elektrisk krets som styrer varmeelementet<sup>12</sup>. Det er viktig med varmeelementet for å hindre skade på de bevegelige delene sensoren har.

### 3.2.1.2 Wind Direction Transmitter - Compact



Figur 3.8: Wind Direction Transmitter<sup>13</sup>

Dette er en sensor som måler vindretningen. Basert på hvordan vinden blåser så roteres haleroret 3.8 automatisk og vindretning kan leses av i form av grader, 0-360°.

Det samme gjelder her som for sensor for vindhastighet, det finnes flere modeller. Leverandøren hadde kun to alternativer tilgjengelig, WRG2/O-10 og WRG2/O-50. Forskjellene ligger også her kun i den elektriske outputen, og med det så ble modell O-10 valgt. Denne har en elektrisk output på 0-10Volt og må også ha en spenningsdeler. Denne sensoren har i likhet med sensor for vindhastighet 3.2.1.1 et innebygd varmeelement.





ZM/O

Figur 3.9: Illustrasjon på montering av stag og sensor <sup>14</sup>

### 3.2.1.3 Ekstraustyr for vindsensorene

For å kunne montere vindsensorene på en fornuftig måte er det blitt kjøpt inn et tverrstag. På dette staget monteres vindsensorene på hver side, vist i figur 3.9. Det vil også monteres en lynavleder på staget.

### 3.2.1.4 Styrker og svakheter

Begge vindsensorene har lignende virkemåter og spesifikasjoner, forskjellen ligger hovedsaklig i de roterende delene. Av den grunn slås denne seksjonen sammen for disse to.

Et av kravene for denne type sensor, slikt som med alt annet av utstyr er håndteringen av det utfordrende klimaet. En av årsakene til at akkurat denne typen ble valgt skyldes at den har et innebygd varmeelement. Dette gjør at minimumstemperatur for drift er langt lavere.

Selv om varmeelementet er nødvendig i vinterhalvåret, har det også en negativ effekt. For at det skal være operativt kreves det en driftsspenning på 24V og dette vil trekke mer strøm fra anlegget. Hvis ikke det er nok tilgjengelig energi til enhver tid risikeres det skade på enhetene.

Disse vindsensorene baserer seg på metoden med Skålkorsanemometeret og vindfløy nevnt i bakgrunnsanalysen om måling av vind 2.2.2.1. Selv om den andre metoden som benytter Ultralyd har lengre levetid, så er den også dyrere. I tillegg hadde oppdragsgiver fra tidligere erfaring med valgte vindsensorer og syntes de fungerte godt til sitt formål.

<sup>11</sup>Vindhastighetsensor: [http://www.fuehlersysteme.de/media/catalog/product/cache/2/image/9df78eab33525d08d6e5fb8d27136e95/W/G/WG2\\_O.jpg\\_13.jpg](http://www.fuehlersysteme.de/media/catalog/product/cache/2/image/9df78eab33525d08d6e5fb8d27136e95/W/G/WG2_O.jpg_13.jpg)

<sup>12</sup>Manual for Wind Speed Transmitter, s.11-12, 16: <http://www.fuehlersysteme.de/downloads/d1/file/id/141/manual.pdf>

<sup>13</sup>Vindretningssensor: [http://www.fuehlersysteme.de/media/catalog/product/cache/2/image/9df78eab33525d08d6e5fb8d27136e95/W/R/WRG2\\_O.jpg\\_13.jpg](http://www.fuehlersysteme.de/media/catalog/product/cache/2/image/9df78eab33525d08d6e5fb8d27136e95/W/R/WRG2_O.jpg_13.jpg)

<sup>14</sup>Monteringsstag for vindsensorene: [http://www.fuehlersysteme.de/media/catalog/product/cache/2/image/9df78eab33525d08d6e5fb8d27136e95/z/m/zm\\_o-30.jpg](http://www.fuehlersysteme.de/media/catalog/product/cache/2/image/9df78eab33525d08d6e5fb8d27136e95/z/m/zm_o-30.jpg)

### 3.2.2 Temperatur- og fuktighetssensorer

Det er ønskelig å måle temperatur både innendørs og utendørs, for dette kreves det flere sensorer. Det var flere alternativer ved valg av temperatur- og fuktighetssensorer, hvor blant annet den tidligere gruppens alternativ ble vurdert 2.5.1. Problemet med denne var at den ordinært var laget for å måle temperatur og fuktighet i jorden. Selv om den håndterte kravene til kuldegrader, var det ønskelig fra oppdragsgiver med noe litt mer robust.

### 3.2.3 EE210 Outdoor

For å måle temperatur og fuktighet utendørs er det blitt kjøpt en "hybridsensor", dette innebærer at den måler flere parametre samtidig. I dette tilfellet er det fuktighetssensor og temperatursensor i én. Sensoren har navnet EE210 Outdoor, og for å sikre mest nøyaktige målinger er det et strålingsskjold av type HA010501 montert som tilleggsutstyr, vist i figur 3.10.

Sensoren har en driftsspenning på 24Volt og gir en elektrisk analog output på 0-10Volt. Den gir målinger på relativ fuktighet på 0-100% og temperaturer fra -40°C til 60°C.



Figur 3.10: Bildet viser sensor EE210 med strålingsskjold <sup>15</sup>

### 3.2.4 Styrker og svakheter - EE210 Outdoor

Sensoren oppfyller de målenøyaktighetskrav som benyttes av meteorologisk institutt 2.2.5. En ulempe med en hybridsensor kan være at dersom temperaturmåleren blir ødelagt, er det knyttet usikkerhet rundt hvorvidt fuktighetssensoren fortsatt vil fungere.

Hvis kun sensoren skulle bli montert, ville den bli utsatt for direkte sollys og vind. Dette ville medført unøyaktige måledata. Det er for øvrig ikke et problem her da sensoren monteres i

<sup>15</sup>EE210 Outdoor:[http://www.epluse.com/uploads/tx\\_eeproducts/EE210\\_Outdoor\\_web\\_01.jpg](http://www.epluse.com/uploads/tx_eeproducts/EE210_Outdoor_web_01.jpg)

strålingsskjoldet. Strålingsskjoldet bidrar til mer presise målinger av både fuktighet og temperatur, og gir i tillegg sensoren en naturlig ventilasjon som gir kort responstid<sup>16</sup>.

### 3.2.5 HTU21D

Det er interessant å kunne måle temperatur og fuktighet innendørs da flere av systemets komponenter er innendørs. De fleste er montert i kassen til brenselcellen vist i figur 3.6, og det er også her sensoren vil bli plassert. Denne plasseringen gir muligheten for å se at temperaturen holder seg innenfor de rammene som utstyret håndterer. I tillegg til å se at fuktighetsnivået holder en akseptabel verdi.

Her ble det i realiteten ikke tatt noe valg. Først ble det utført en test som gikk ut på å lese av data fra sensoren ved bruk av styringsenheten. For å utføre testen ble det lånt en sensor av type HTU21D fra Makerspace ved HiØ. HTU21D er en temperatur og fuktighetssensor. Erfaringene fra denne testingen var at sensoren fungerte veldig bra, og fordi sensoren skal være innendørs trenger den ikke å være spesielt robust. En rimelig sensor vil derfor være tilstrekkelig. Derfor ble en HTU21D kjøpt inn og brukt i dette prosjektet.

HTU21D har en varierende driftsspenning fra 1.5-3.6Volt, men den er typisk på 3.0V<sup>17</sup>. Den kommuniserer digitalt med I2C protokollen og måler relativ fuktighet fra 0-100 % og temperaturer fra -40°C til 125°C.

#### 3.2.5.1 Styker og svakheter - HTU21D



Figur 3.11: Figuren viser hvordan sensoren HTU21D ser ut <sup>18</sup>

Sensoren er en typisk sensor brukt i amatørprosjekter med en pris på drøye 130 kroner. Den er veldig liten av størrelse, se figur 3.11, noe som gjør at den i prinsippet kan monteres hvor som helst. For å bruke den trengs det kun fire tilkoblede pins. HTU21D's strømforbruk er meget lavt

<sup>16</sup>EE210 Outdoor datablad: [http://downloads.epluse.com/fileadmin/data/product/ee210/BA\\_EE210\\_OUTDOOR\\_e.pdf](http://downloads.epluse.com/fileadmin/data/product/ee210/BA_EE210_OUTDOOR_e.pdf)

<sup>17</sup>HTU21D manual: <http://cdn.sparkfun.com/datasheets/BreakoutBoards/HTU21D.pdf>

<sup>18</sup>HTU21D: <https://cdn.sparkfun.com/assets/parts/8/6/6/7/12064-01.jpg>

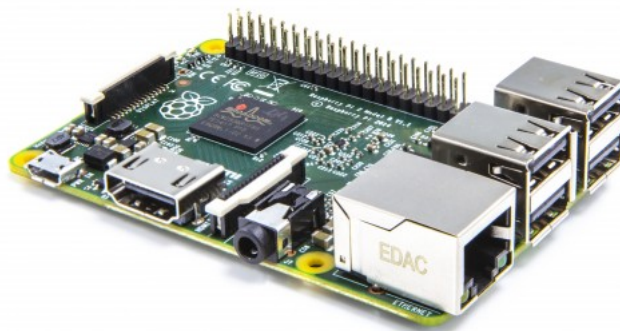
sammenlignet med andre enheter, noe som er veldig positivt. Sensoren har en rask responstid, som bidrar til å sikre nøyaktige og tidsriktige målinger.

Selv om sensoren håndterer lave temperaturer, så tåler den ikke vann. Uten en form for vanntett beholder må den med andre ord benyttes innendørs.

### 3.3 Styringssystemet

I analysekapittelet under seksjonen om styringssystemet 2.3 blir alle krav som blir stilt til styringssystemet og styringsenheten drøftet og utredet. Den mest kritiske enheten i systemet er styringsenheten, i avsnittet 2.3.5 blir ulike alternativer drøftet. Konklusjonen falt på at en datamaskin som får plass på et lite kretskort vil være den beste løsningen å bruke som styringsenhet. På bakgrunn av dette falt valget på Raspberry Pi, heretter referert til som RPi. Grunnen til at valget falt på denne og ikke CHIP, Odroid C2 eller andre lignende datamaskiner var fordi RPi har mye større brukerstøtte og community enn andre typer med lignende spesifikasjoner.

**Raspberry Pi** er en billig og liten datamaskin på størrelsen med et kredittkort. Det britiske selskapet Raspberry Pi Foundation har utviklet produktet med tanke på å oppmuntre læring om databehandling. Denne tanken har vist seg å være god da stadig flere benytter RPi i undervisningssammenhenger<sup>19</sup>.



Figur 3.12: Raspberry Pi 2<sup>20</sup>

RPi har mange ulike modeller på markedet. Forskjellene ligger i prosessorhastighet, størrelsen på minnet, antall USB-porter, antall GPIO-pinner og strømforbruk. Siden alle de ulike variantene er relativt billige er ikke dette en stor faktor for hvilken versjon som skal bli kjøpt inn.

Dette prosjektet skal utforske hvor mye utstyr en målestasjon kan ha med et eget energisystem, det er da lurt å velge en av de versjonene med flest GPIO-pinner. Det er to versjoner som har 40 GPIO-pinner og de er RPi model B+ og RPi 2 modell B, vist i figur 3.12. Forskjellen på disse to variantene ligger i prosessorhastighet, størrelsen på minnet og strømforbruk. Modell B har blant annet dobbelt så mye RAM, og prosessoren kjører på litt høyere frekvens.

RPi 2 Modell B er den av de to aktuelle modellene som vil bli brukt i dette prosjektet.

<sup>19</sup>Hva er en Raspberry Pi: <https://www.raspberrypi.org/help/what-is-a-raspberrypi/>

<sup>20</sup>Raspberry Pi 2: <https://www.raspberrypi.org/blog/raspberrypi-2-on-sale/>

Modell B har følgende spesifikasjoner <sup>21</sup>:

- Prosessor: 900MHz quad-core ARM Cortex-A7 CPU
- RAM: 1GB
- 40 GPIO pins
- VideoCore IV 3D grafikk kjerne
- Porter: 4 USB, HDMI, Ethernet, 3.5mm audio jack
- Interface: Kamera (CSI) og Display (DSI)
- Micro SD-kort inngang

**Styrker og svakheter** for RPi avhenger av hva som er den gitte oppgaven den skal utføre. Den største fordelen ved denne modellen er at den har bedre prosessorhastighet og mer minne enn de andre alternativene. Dette bidrar til en hurtigere dataprosessering og den vil kunne håndtere flere oppgaver samtidig på en bedre og mer effektiv måte, noe som dette prosjektet er avhengig av.

Forbruk av strøm er en kritisk faktor i dette prosjektet. Selv om valgt modell er den som bruker mest strøm har ikke summen av dette stor nok betydning til at det skal vektlegges i dette tilfellet. Det kan argumenteres for at det er negativt at den koster mer, men prisforskjellene er så små at de ikke blir tatt hensyn til<sup>22</sup>. Isteden prioriteres optimal ytelse og funksjonalitet.

GPIO pinnene er godt egnet for digital styring og avlesning, men skal RPi kommunisere med analoge signaler må det være en A/D konverter mellom enhetene. Dette krever en utvidelse av RPi<sup>23</sup>.

Kommunikasjonsmulighetene til RPi er begrenset, den kan blant annet ikke kommunisere med RS-232 protokollen. Dette kan by på problemer når den skal kommunisere med brenselcellen. Det finnes flere ulike måter å løse dette på. En av løsningene vil være å benytte en mikrokontroller som kan bruke RS-232 protokollen. Mikrokontrolleren vil da håndtere styringen av brenselcellen, deretter vil RPi styre mikrokontrolleren. En annen løsning vil være å gå til innkjøp av en utvidelse som gjør RPi istand til å kommunisere via RS-232 protokollen.

**Et linux basert system** var et av oppdragsgivers ønsker for dette prosjektet. En rekke ulike operativsystemer og distribusjoner kan kjøres på RPi. Raspbian er stiftelsens offisielle støttede operativsystem og baserer seg på Linux Debian<sup>24</sup>. Dette er operativsystemet som vil bli benyttet på RPi i prosjektet.

**Tilleggsutstyr til Raspberry Pi** er nødvendig fordi, som det er skrevet i avsnittet om styrker og svakheter 3.3 til RPi, så har den ikke muligheten for å kommuniserer med analoge signaler uten en utvidelse. Det er da behov for en A/D konverter mellom enhetene. Av denne grunn er det blitt kjøpt inn to A/D konvertere.

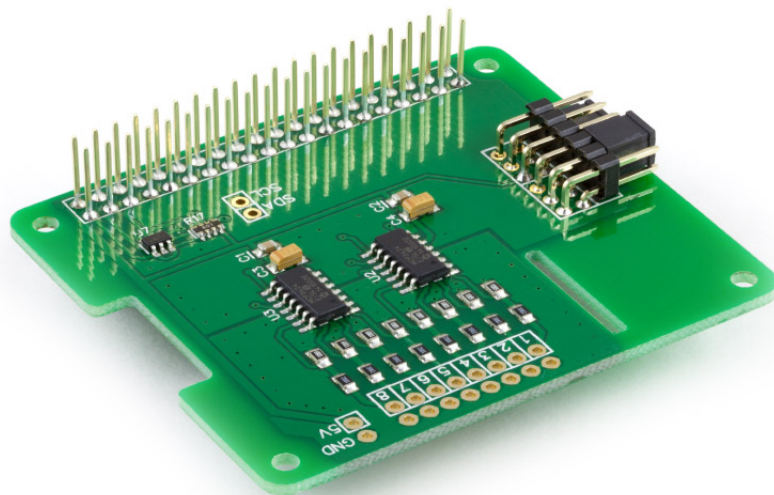
ADC Pi Plus vist i figur 3.13, er en utvidelse for RPi som konverterer analoge signaler til digitale, og det er denne utvidelsen som blir brukt i dette prosjektet. Det er en 17 bit

<sup>21</sup>Spesifikasjoner: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md>

<sup>22</sup>Priser: <https://www.raspberrypi.org/help/faqs/#buyingCost>

<sup>23</sup>A/D konverter: <https://www.abelectronics.co.uk/p/56/ADC-Pi-Plus---Raspberry-Pi-Analogue-to-Digital->

<sup>24</sup>Raspbian: <https://www.raspberrypi.org/downloads/raspbian/>

Figur 3.13: ADC Pi Plus<sup>25</sup>

ADC med 8 kanaler. Kortet er basert på to mikro-chiper MCP3424 A/D, som er deltasigma A/D konvertere, med lav støy differensial input. Mikrochippene kommuniserer med RPi via kommunikasjonsprotokollen I2C. Utvidelsen får sin strømtilførsel fra RPi og utvider GPIO pinnene som gjør det mulig å montere opp til 4 ADC oppå hverandre på toppen av selve RPi<sup>25</sup>.

### 3.3.1 Hvordan skal den styre?

Det vil være et hovedprogram som vil kjøre kontinuerlig, for å kontrollere at alle deler av systemet fungerer slik det skal. Dette er et veldig tregt system, fordi det omhandler systemer og hendelser som bruker tid på å endre seg. Nettopp av denne grunnen vil ikke programmet kontrollere verdier til enhver tid, men den vil typisk kjøre hvert minutt for så å "sove" i mellom kontrollene. Hovedprogrammet vil bli delt opp i flere ulike deler hvor hver del tar for seg de ulike systemene i målestasjonen. På denne måten vil hovedprogrammet bli oversiktlig og lett forståelig. Under er

<sup>25</sup>ADC Pi Plus: <https://www.abelectronics.co.uk/p/56/ADC-Pi-Plus-Raspberry-Pi-Analogue-to-Digital>

det en pseudokode som illustrerer planen for hvordan hovedprogrammet vil fungere.

```

while True do
  i = 0
  Målesystemet:
  if i % 10 == 0 then
    | Hent alle verdier fra målesystemet og logg
  else
    | Hent interessante verdier for energisystemet
  end
  Vurder verdierne, og utløser eventuelle hendelser eller alarmer ved feil eller avvik

  Energisystemet:
  Har nødvendige verdier fra målesystemet, leser så av aktuelle verdier fra
  energisystemet.
  Vil så vurdere disse verdiene og utløse hendelser eller alarmer ved eventuelle feil eller
  avvik.
  if i % 10 == 0 then
    | Logg verdier fra energisystemet
  end
  i++
end

```

### 3.3.2 Planen for overstyring og fjernstyring

Slik det blir nevnt i analysekapittelet i seksjonen om overstyring 2.3.2 er bruk av SSH den beste tilnærmingen for å fjernstyre og overstyre dette systemet. Tanken er da at en person kan sitte på en ekstern lokasjon og logge seg inn på systemet for å gjøre ønskede endringer. Det vil derfor bli laget et eget program som kan startes via SSH. Her vil brukeren få flere ulike valgmuligheter som vil gjøre det enkelt å gjøre endringer på systemet. Et eksempel kan være hvis oppdragsgiver finner ut av at hovedprogrammet gjør en feilvurdering og setter varmeelementene til kameraene på når det ikke er behov for det. Da kan oppdragsgiver benytte dette programmet via SSH for å overstyre det autonome programmet.

## 3.4 Plan for det eksterne lagringssystemet

I prosjektet vil en server ved Høgskolen i Østfold bli brukt som eksternt lagringssystem. Det skal samles inn store mengder viktig data, og eventuelle tap av data vil være kritisk. Det er derfor viktig at serveren har gode prosedyrer for feilhåndtering. Serveren er en virtuell instans som kjøres i VMware. VMware har innebygde funksjoner for backup og sikring av data.





## Kapittel 4

# Implementasjon

Dette kapitlet tar for seg hvordan systemet er blitt satt opp og hvilke endringer som eventuelt er blitt gjort i forhold til planleggingskapitlet 3. Systemets komponenter er montert på eller i boden vist i figur 4.1. Kode for webgrensesnitt og hovedprogram for styring legges på en minnepennen, men noen deler av koden vil bli forklart i detalj.

### 4.1 Energisystemet

Energisystemet er som tidligere nevnt i kapittel 2.1 bestående av tre energigenererende kilder; Solceller, vindmølle og brenselcelle. Produsert energi skal lade opp en batteribank som systemet benytter som strømkilde.

Det er fire solcellepanel i systemet fra produsent Solartek, og disse er montert på veggen til boden vist i figur 4.1. Valget av type solcellepanel og monteringen av disse er det oppdragsgiver som har tatt ansvar for. Panelene er tilkoblet en chargekontroller 3.1.1.1 som tar seg av ladeprosessen til batteriene for solenergi. Chargekontrolleren er tilkoblet ruter 4.3.6.1 via Ethernet og benytter Modbus TCP/IP for kommunikasjon. De data som kan bli avlest er i form av Wh og sier noe om hvor mye energi som solcellene genererer, akkurat nå og totalt. Det er i tillegg mulig å se hvor mye batteriene blir ladet.

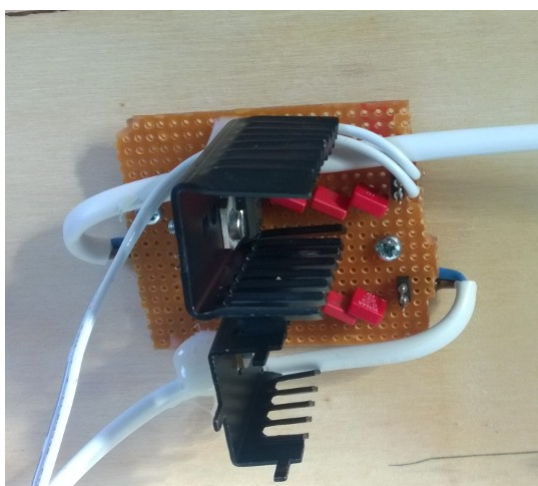
Vindmøllen som benyttes er av type LE600 og er blitt valgt av oppdragsgiver. Monteringen av denne er det oppdragsgiver som har tatt ansvaret for. Vindmøllen er i likhet med solcellepanelene også tilkoblet en chargekontroller 3.1.1.2. Oppgaven dens er lik som for sol men den benytter en annen form for kommunikasjon, Modbus seriell. En av utfordringene med denne typen chargekontroller kontra den andre er at data som kan hentes ut er begrenset. Eksempelvis hvor det på chargekontrolleren til solcellene er mulig å se strømmen som genereres og hvor mye batteriene lades, så er det for denne chargekontrolleren begrenset til å kun se hvor mye som er blitt produsert. En løsning på dette kunne vært og hatt en shuntmotstand mellom batteribank og chargekontroller, men på grunn av dårlig tid og sen levering av nødvendige komponenter er det ikke mulig å gjennomføre i dette prosjektet.

Brenselcellen i systemet er av type EFOY Pro 2400 vist i figur 3.3. I og med at dette er en veldig kostbar komponent i systemet er den blitt valgt i tett samarbeid med oppdragsgiver. Selve brenselcellen ble levert ferdig oppsatt og konfigurert av Power Controls AS. Ved levering var den tilkoblet to batterier som senere ble benyttet for å teste den første utgaven av spenningsregulatoren 4.2. Som et resultat av dette ble det brukt mer metanol enn hva en kunne forvente hvis batteriene allerede var tilkoblet batteribanken. Nå er batteriene i parallell med resten av banken og forbruket av metanol ved plussgrader er tilnærmet null. Dette skyldes at anti-freeze-mode kun trenger å



Figur 4.1: Bildet viser boden med solcellepanel montert på den ene siden

kjøre dersom temperaturen ute nærmer seg frysepunktet, og i tillegg genereres det mer solenergi enn hva batteriene klarer å lade opp. Derimot hvis batterispenningen synker under nedre grense vil brenselcellen kjøre til batterienes øvre grense nås. Grensene er konfigurert på forhånd, og det er valgt å beholde disse innstillingene. Nedre grense er satt til 24.0V og øvre grense er 28.0V.



Figur 4.2: Spenningsregulator kretsen utgave 1

Brenselcellen benytter RS-232 for kommunikasjon og er koblet til en RPi for å ekstrahere data. Batterispenning, operasjonstid, status og approksimert drivstofforbruk er eksempler på data som er tilgjengelig, vist i figur 4.3.

Alle enhetene er tilkoblet en felles batteribank som resten av systemet tar strømmen fra. Denne

```
sfc
battery voltage 25.87V
output current -0.05A
operation time (charge mode) 63.2h
operating state: auto off
operating mode: auto
cumulative output energy 6546.3Wh
no error
cartridge level above sensor or no sensor (cartridge 1.1 consumed 5.7041)
no warning
```

Figur 4.3: Eksempeldata som kan hentes ut fra Brenselcellen



Figur 4.4: Batterier som er utenfor kassen til Brenselcellen

batteribanken er bestående av ti 12 volt batterier, og ikke åtte som det i utgangspunktet var tiltenkt i analysekapittelet 2.1.4. Årsaken til dette er at med Brenselcellen fulgte det med to batterier som også ble koblet til. For systemets drift spiller dette ingen rolle bortsett fra at det er mulig å lagre unna mer energi. Batteriene er koblet i serie to og to, for så å kobles i parallell med resten, som vist i figur 4.4. Resultatet er en batteribank på 24 volt som tiltenkt, men med to ekstra batterier. Figur 4.5 viser hvor batteribanken er koblet til og hvor systemet henter strømmen fra. Av figuren kan en også se at det er lagt inn en sikring, denne er på 25 ampere.

## 4.2 Målesystemet

Målesystemet er bestående av sensorer for vær- og klimaverdier, slik det står beskrevet i kapittel 2, bakgrunns analyse 2.2. Det er tre sensorer montert utendørs; En for vindretning, en for vindhastighet og en for både temperatur og fuktighet. Innendørs er det én sensor som måler temperatur og luftfuktighet, denne er plassert i kassen som Brenselcellen befinner seg i.

Vindsensorene er påmontert et stag slik som det er vist i figur 3.9. Staget er satt på toppen av en mast som ruver omtrent 6m over bakken. Bilde av masten ses i figur 4.6. I manualen for vindsensorene kommer det frem at de bør være plassert omtrentlig 10m over bakken med ingen



Figur 4.5: Knutepunktet for sammenslåing av batteriene og strømuttak for enheter

forstyrrende element i rundt<sup>1</sup>. Vognen slik den er plassert nå er på en liten bakketopp, og hvis høyden på bakketoppen medregnes så oppfylles kravet til plasseringen av sensorene. Selv om det ikke skulle være helt korrekt for å få mest mulig riktig data, så er det funksjonaliteten som er i fokus i dette prosjektet. Sensorene skal plasseres et annet og mer permanent sted senere og da vil det være viktigere å ta hensyn til kravene for montering.

Temperatur- og fuktighetssensoren er plassert i et strålingsskjold som gir beskyttelse mot sol og nedbør, men sørger for å opprettholde tilstrekkelig ventilasjon 3.2.3. Skjoldet er fastmontert på den samme veggen på boden som kabelgjennomføringen og masten. Sensoren selv er heftet fast på skjoldet med sonden pekende opp.

#### 4.2.1 Elektronikk sensordel

Alle sensorene som benyttes i prosjektet, bortsett fra sensoren som måler temperatur og fuktighet inne, leverer et analogt målesignal på intervallet 0 til 10 volt. ADCen som benyttes som et shield på RPi måler derimot signaler på et spenningsintervall fra 0 til 5 volt. Det var derfor nødvendig å sette opp en spenningsdeler før signalet fra sensorene blir målt. Det ble bestemt å benytte vindretningssensoren til å teste spenningsdeler og ADC. Dette fordi vindretningssensoren enkelt kan manipuleres til å gi ut signaler på hele spennings-spekteret. Litt overraskende ble det oppdaget at signalet inn til ADC aldri nådde 5 volt, til tross for at motstandene i spenningsdeleren ble målt til å være like ( $1k\Omega$ ), og signalet ut fra sensor ble målt til å være 10 volt. Spenningen over inngangen på sensoren stoppet opp ved ca 4,85 volt. Databladet til ADCen avslørte at inngangene på ADCen har en inngangsimpedans på  $16,8k\Omega^2$ . Dermed førte denne oppkoblingen til at den ene motstanden i spenningsdeleren kom i en parallellkobling med inngangsimpedansen, som vist i figur 4.7.

Følgende utregning viser da at motstanden i parallellkoblingen mellom den ene motstanden i spenningsdeleren og inngangsimpedansen blir:  $\frac{1}{1000} + \frac{1}{16800} = \frac{1}{R_{\text{parallell}}}$ .  $R_{\text{parallell}} = 944\Omega$ . Hvis det da kobles på 10 volt på spenningsdeleren, vil det i følge teorien være:  $\frac{10}{1000+944} * 944 = 4,85$  volt på inngangen, altså over parallellkoblingen. Dette stemte med det vi målte. Løsningen på dette ble å benytte inngangsimpedansen som et ledd i spenningsdeleren, se figur 4.8. En

<sup>1</sup>Manual vindretningssensor <http://www.fuehlersysteme.de/downloads/dl/file/id/143/manual.pdf>

<sup>2</sup>adc-pi-plus schematic: <https://www.abelectronics.co.uk/docs/stock/raspberrypi/adcpilus/adc-pi-plus-schematic.pdf>



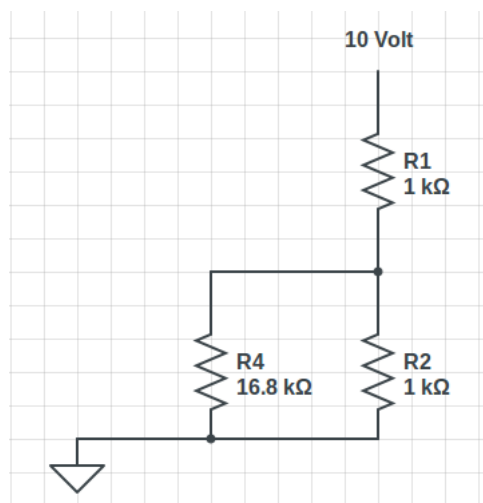
Figur 4.6: Masten med sensorer for vindretning- og hastighet montert på toppen

motstand på  $16,8k\Omega$  burde dele signalet slik at ved 10 volt ut fra sensor, burde det legges seg 5 volt over inngangen. I praksis viste det seg at denne inngangsimpedansen varierte noe fra inngang til inngang. Derfor ble det nødvendig å måle opp og bestemme motstander til hver enkelt inngang som skulle benyttes. Det henvises til tillegg A for å se hvilke størrelser på motstandene som ga korrekt måleintervall på hvilke innganger.

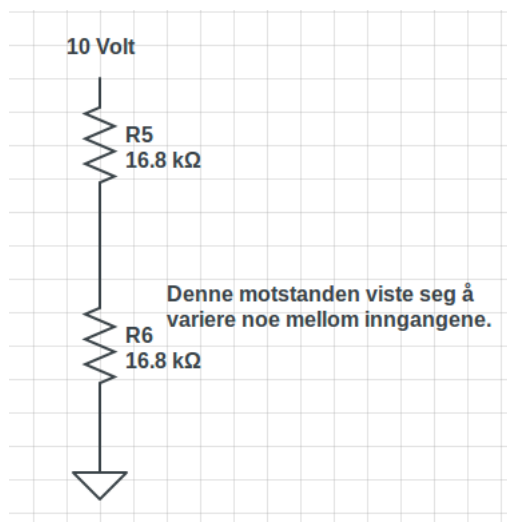
Det ble vurdert om det skulle benyttes et lavpassfilter for å filtrere ut støy på målesignalene. Beslutningen falt ned på at støyfiltreringen kan gjennomføres i tilstrekkelig grad i programvaren. Dette gjøres ved at en rekke samplinger blir utført. Deretter blir samplingene sortert etter verdi. Et midtsegment av den sorterte listen plukkes ut og utgjør grunnlaget for en gjennomsnittsverdi som aksepteres som målt verdi. Tanken er derfor at målinger preget av støy vil havne utenfor dette midtsegmentet.

Som en sikring fra sensor mot ADC ble det benyttet 5,6 volts zenerdioder. Disse står da koblet i sperreretning mellom målesignal og jord. Hvis spenningen skulle overstige 5,6 volt vil disse diodene begynne å lede i sperreretning mot jord, og dermed hindre at store strømmer ledes inn mot ADCen.





Figur 4.7: En tradisjonell spenningsdeler på inngangssignalet blir galt.



Figur 4.8: Inngangsimpedansen fungerer som en spenningsdeler med en ytre motstand.

Detaljerte koblingsbeskrivelser og kretsskjemaer er å finne i tillegg A.

### 4.3 Styringssystemet

En viktig del av styringssystemet er muligheten til å kunne slå av og på enheter som brukes i systemet. I dette systemet er det de energikrevende varmelementene som skal ha en av/på-styring, samt routeren og kameran systemet.

### 4.3.1 Elektronikk styringsdel

For å kunne benytte GPIO-pinnene på RPi til å slå av og på elementer i systemet ble det benyttet optokoblere. GPIO-pinnene kan settes til å operere binært, altså en av/på-logikk. Hvor ”av” er 0 volt, og ”på” er 3,3 volt. Optokoblerne som ble brukt i prosjektet er av typen HCPL-3120. Disse er på 2.5 amper og har en driftspenning fra 15-30 volt. 2.5 amper ble regnet for å være tilstrekkelig for systemet. Ingen av kretsene optokobleren skal inngå i overgår 2 amper. For eksempel vil et varmelement på 20 watt og 24 volt tilsvare rundt 0,8 amper. En optokobler fungerer slik at når den får et styringssignal inn på inngangen, tennes det en lysdiode inne i optokobleren. Optokobleren er koblet til en driftsspenning, og når en fototransistor inne i optokobleren reagerer på lyset fra lysdioden begynner den å lede strøm<sup>3</sup>.

Spenningen mellom jord og utgangen på optokobleren blir da tilnærmet lik driftspenningen. Bruken av lysdiode og lystransistor gjør at det blir et galvanisk skille mellom enheten som sender inn styringssignalet og driftspenningen. Når et høyt signal sendes ut fra RPi, vil optokobleren sende driftsspenning ut på utgangen. Når et lavt signal sendes inn på inngangen, vil optokoblerens utgang gå mot 0 volt. I alt ble det benyttet fem optokoblere i systemet. En til router, to til hvert sitt varmelement i vindsensorene, en til kamera, kameravifte og kamera-streamingsboks og en til varmelementet i kamerahuset.

Vindsensorene kom levert med innebygde varmelementer med tilhørende elektronisk krets som styrer om varmelementet skal slås av eller på. Begge vindsensorene ble levert med kabel montert fast i sensor. Kabelen inneholdt i alt 7 ledninger, 3 par og 1 jordingsleder. Et par er for sensorens driftsspenning, et par er målesignalet og det siste paret er varmeelementets driftsspenning. Det at varmeelementet har egne ledninger for spenningskilde er nødvendig for å muliggjøre av/på-styring. Det er altså mulighet for å overstyre den innebygde kretslogikken til varmeelementet ved at strømmen kuttes selv om sensorene i utgangspunktet ønsker å slå varmeelementet på. Det er derimot ikke mulig å kjøre varmeelementene hvis den innebygde logikken i sensorene slår av varmeelementet. Det er noe uklart hvordan denne logikken er bygd opp i sensoren, databladet til sensoren avslører ingenting om dette. Leverandør opplyste i en telefonsamtale at kretsen var bygd slik at sensorene slo seg av når temperaturen nådde ca 5 grader celsius, og på når temperaturen falt til ca 2-3 grader celsius<sup>4</sup>.

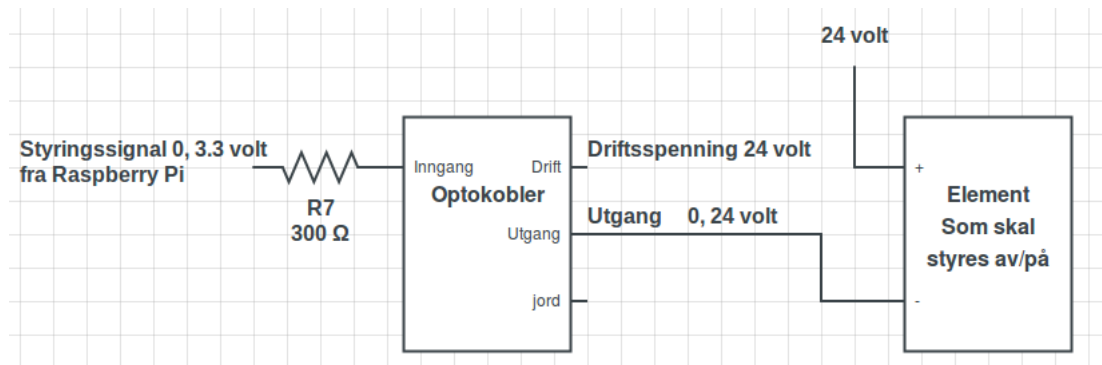
Siden styringen av varmelementene handler om å kunne spare energi er det neppe så interessant å kjøre varmelementene når utetemperaturen stiger over 5 grader. Det er mest interessant i et energiperspektiv at man har mulighet til å kutte ut varmeelementene for å spare energi. I tillegg fungerer den innebygde kretsen som en sikring for at systemet ved en feil ikke blir stående å kjøre varmeelementet når det ikke er behov for det, og dermed brenne av unødvendig energi. Også varmeelementet i kamerahuset hadde en termostat koblet til. Denne ble beholdt for å sikre at varmen ikke skulle overstige de grensene som produsenten har satt, men det er altså mulighet for å koble ut varmeelementet selv om termostaten ikke gjør det.

Optokoblerene som ble brukt i av/på-styringen ble koblet slik at den positive lederen til enheten som skal slås av eller på, blir koblet direkte på driftsspenningen i stasjonen, mens den negative lederen blir koblet til optokoblerens utgang. Dette er vist i figur 4.9 Når optokobleren mottar et høyt signal fra RPi, vil spenningen mellom optokoblerens utgang og jord bli tilnærmet lik driftsspenningen. Da blir spenningsforskjellen mellom polene til enheten 0 volt, og enheten er av. Når optokobleren mottar et lavt signal fra RPi, faller spenningen mellom jord og optokoblerens utgang ned til 0, og spenningsforskjellen mellom positiv og negativ pol på elementet som skal

<sup>3</sup>Optokobler: <https://snl.no/optokobler>

<sup>4</sup>Fuehlersysteme: <http://www.fuehlersysteme.de/>

styres blir tilnærmet lik driftspenningen, og elementet er nå på. Dermed vil altså et høyt signal ut på styringspinnen på RPi slå av elementet, og et lavt signal ut fra styringspinnen vil slå på elementet. I henhold til optokoblerens datablad ble det også koblet en kondensator på 1 mikroFrahad mellom jord og driftspenning på optokoblerne.



Figur 4.9: Optokoblerens utgangssignal kobles som jordsignal på styringselement.

I tillegg til de tre varmeelementene er det også koblet opp mulighet til å slå av eller på både router og kameranettet. Med kameranettet menes vifta i kamerahuset, streamingboksen til kamera og kameraet selv. Alle disse komponentene kjøres på 12 volt, bortsett fra streamingboksen som kjøres på 24 volt. Behovet for å kunne slå av og på routeren er for å resette den hvis den skulle låse seg og dermed nekte systemet adgang til Internett. Måten dette gjøres på er ved at RPi med jevne mellomrom forsøker å pinge en rekke servere. Hvis systemet oppdager at den ikke har kobling ut mot Internett kan det være at routeren har gått i lås. Dermed vil RPi forsøke å kutte strømmen til routeren for og deretter slå den på igjen.

Den mobile routeren som benyttes i prosjektet krever 12 volt spenningskilde. Batterispenningen i stasjonen er som kjent på 24 volt. Det var derfor nødvendig å benytte en spenningsregulator. Valget falt på regulatoren L7812 som tar spenningen ned fra 24 til 12 volt. Ulempen med spenningsregulatoren er at den brenner av endel energi som varme, men siden spenningsregulatoren er plassert inne i isolasjonskassen til brenselcellen vil denne varmenergien kunne benyttes til oppvarming av miljøet inne i kassen på kalde vinterdager. Dermed reduseres behovet for at brenselcellen selv må starte opp for å holde innertemperaturen i kassa høy nok til at brenselcellen kan fungere. Spenningsregulatoren har tre pinner. Den ene pinnen er til driftspenningen, her 24 volt. Den andre pinnen er jord. Tredje pinnen er da utspenningen på 12 volt. Optokobleren kobles slik at utgangen på optokobleren løfter jord på spenningsregulatoren til 24 volt. Dermed blir spenningsforskjellen mellom driftspenning og jord 0 volt når optokobleren får positivt innsignal. Dermed blir den tredje pinnen på spenningsregulatoren, spenningen til routeren, 0 volt. Når optokobleren kobler ut, faller spenningen på optokoblerens utgang til 0, og spenningsforskjellen på driftspenning og jord på spenningsregulatoren blir 24 volt. Dermed blir den tredje pinnen 12 volt, og routeren kjører.

I kamerahuset ble det montert en DC til DC konverter. Denne ble levert med kamerahuset, og er et komplett kretskort som har positiv og negativ inngang og utgang. Kabelen som ble trekt opp til kamerahuset måtte da inneholde i alt tre ledere. En leder for 24 volt. Denne ble splittet ut til positiv tilkobling på varmeelementet og inngangen på DC til DC konverteren. De to andre lederne var utgangene fra de to optokoblerne. Den ene optokobleren som skulle styre varmeelementet, og den andre optokobleren som skal styre kameranettet. Disse ble det koblet inn på henholdsvis



varmeelementets negative inngang, og spenningsregulatorens negative inngang. Verdt å merke seg at streamingboksen ikke står i kamerahuset, men fortsatt styres av samme optokobler som slår av og på konverteren i kamerahuset. Denne koblingen ble gjort ved å splitte utgangen fra optokobleren i to signaler. Hvorav det ene går opp til kamerahuset og inn i konverteren, mens den andre går til streamingboksens negative inngang.

RPi krever en driftsspenning på 5 volt. Dermed var det behov for enda en spenningsregulator. Her falt valget på L7805. Denne tar spenningen ned fra 24 til 5 volt. De to spenningsregulatorene som ble benyttet til router og RPi krevde at det ble montert kjøleribber som hindrer at de overopphetes. I tillegg var det også krav i databladet om at det skulle kobles opp kondensatorer mellom pinnene. Det henvises til tillegg A for nærmere detaljer om hvordan dette ble gjort.

En viktig del av systemet er å overvåke hvor mye de forskjellige komponentene forbruker av energi. For å løse dette ble det brukt shuntmotstander. I alt tre shuntmotstander ble montert. Alle tre har egenskapene at de kan måle opp til seks amper. Spenningen som måles over shunten i intervallet 0 til 6 amper, er 0 til 150 mV. En shunt ble montert i kretsen sammen med RPi og routeren. Den andre ble montert i kretsen til vindsensorene, herunder også varmeelementene til vindsensorene. Den tredje ble montert i kretsen med kamerasystemet, herunder også varmeelementet i kamerahuset. For å måle spenningen over shuntmotstandene benyttes det en differensiell ADC. Det føres da to signaler til hver inngang på den differensielle kanalen. En fra hver side av shuntmotstanden. Differansen mellom disse to målingene er da spenningsfallet over shuntmotstanden. I programmet regnes det ut hvor mye watt som forbrukes i hver krets. Det gjøres ved å først beregne den konstante motstanden i shunten.  $R_{shunt} = \frac{150mV}{6Amper}$ . Deretter kan strømmen som går i kretsen beregnes ved følgende formel:  $I = \frac{U_{maaling}}{R_{shunt}}$ . Effekten kan da regnes ut ved  $P = U_{batt} * I$ .

### 4.3.2 Klargjøring av RPi

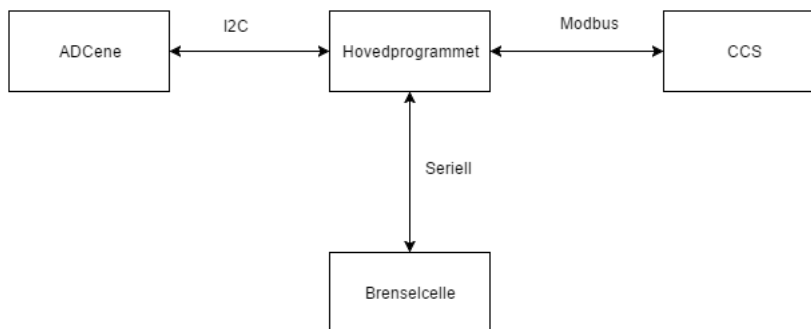
Underveis i arbeidet ble det gjort en rekke installasjoner og konfigurasjoner på RPien. En komplett liste over alle installasjoner som ble utført er å finne på minnepennen med navn "installasjonsliste". Operativsystemet Raspbian kommer ferdig installert med et grafisk brukergrensesnitt. Det anses som unødvendig å boote inn i det grafiske brukergrensesnittet, derfor er "boot option" satt til å boote rett inn i terminal for å minske oppstartstid, energiforbruk og maskinressurser. Det har også blitt gjort et par konfigurasjoner som sørger for at RPi kan kommunisere over seriell og I2C. Disse konfigurasjonsalternativene, med flere, er tilgjengelige i et konfigurasjonsprogram som kan startes fra terminal ved å utføre kommandoen "sudo raspi-config".

### 4.3.3 Hovedprogram

Hovedprogrammet ble i sin helhet skrevet i programmeringsspråket python 3. Valget falt på python 3 fordi det er et enkelt og kraftfullt programmeringsspråk. Det er også det språket som Raspberry Pi foundation anbefaler<sup>5</sup> til nybegynnere. Dette medfører at python 3 er benyttet i mange bøker, guider og eksempler som er å finne på Internett. Det fremstår også som lettere å finne nyttige biblioteker og moduler til prosjekter for RPi hvis python 3 benyttes.

**Tanken bak** hovedprogrammet baserer seg på pseudokoden 1 i planleggingskapittelet. Programmet er delt opp i to deler, målesystemet og energisystemet. Den evige while-løkken vil kun utføre logging til databasen hver tiende iterasjon. I alle de andre iterasjonene vil programmet

<sup>5</sup>RPI help <https://www.raspberrypi.org/help/faqs/>



Figur 4.10: Kommunikasjonen mellom enheter og hovedprogrammet.

kun hente ut og tolke verdier som er viktig for styringen av systemet. Verdier som utetemperatur og batterispenning vil avgjøre om ulike enheter skal bli skrudd av eller på. Før målingene blir logget til databasen blir de vurdert opp mot forrige verdi og eventuelle feil eller avvik utløser en alarm.

Det er snakk om to typer vurderinger som gjøres. Den første handler om å vurdere verdiene mot forrige måling. Hvis forskjellene mellom målingene svinger for mye over tid kan det tenkes at det er noe feil på sensorene eller i systemet. Slike hendelser må inntreffe ti ganger på rad før systemets eier blir varslet om hendelsen. Skjer det derimot én gang blir det logget i en intern loggfil. Den andre vurderingen som gjøres er basert på hvordan omgivelsene rundt stasjonen er. Er det en måleverdi som krysser en øvre eller nedre grense, for eksempel at temperaturen faller ned mot frysepunktet, vil hovedprogrammet vurdere om en enhet skal bli skrudd av eller på. Hovedprogrammet tar disse avgjørelsene ved å sammenligne verdiene som blir målt med forhånds konfigurerte grenser som er satt i en database. Disse grensene kan derfor endres av systemets administrator.

I python benyttes såkalte moduler. Dette er filer som inneholder variabler og metoder, som kan bli brukt i andre filer. Hovedprogrammet består av en fil som importerer mange slike moduler som løser ulike oppgaver som avlesning av verdier fra ADCene, Charger Controllene, brenselcellen og databasen. Det vil også være slike moduler som benytter seg av GPIO-pinnene for å styre de ulike enhetene i systemet. På denne måten blir hovedprogrammet mer oversiktlig, enklere å forstå og tolke.

**Kommunikasjon** for hovedprogrammet består av avlesning av sensorer, shunter, charge controllere og brenselcelle. Figur 4.10 viser kommunikasjons-protokollene mellom hovedprogrammet og de forskjellige enhetene i systemet. Sensorene og shunterne sender analoge signaler inn på ADCene hvor de blir konvertert til digitale signaler. For å lese av disse verdiene må RPi derfor kommunisere med ADCene. Denne kommunikasjonen foregår over I2C. "I2C is a serial protocol for two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems"<sup>6</sup>. Kommunikasjonen til charge controlleren foregår over modbus. Det er en åpen Master/slave applikasjonsprotokoll, som på grunn av sin enkelhet og robusthet har blitt en industristandard for kommunikasjon mellom enheter <sup>7</sup>. Figuren 4.10 viser bare charge controlleren for solcellene. Dessverre viste det seg at charge controlleren for vind,

<sup>6</sup>Info om I2C <http://i2c.info/>

<sup>7</sup>Modbus TCP <http://goo.gl/DJd2Z6>

som ble kjøpt inn av oppdragsgiver, ikke leverte verdier som var interessante for dette prosjektet. Dette er nærmere beskrevet i diskusjonskapittelet 6, men det har blitt utviklet en modul som viser hvordan kommunikasjonen med denne enheten kan foregå. Denne modulen finnes på minnepennen under navnet "lesVindCC.py". Kommunikasjon mellom brenselcella og RPi er en seriell kommunikasjon over en RS-232 kobling. Seriell kommunikasjon vil si at det sendes én og én bit av gangen.

**Modulen for ADCene** ble hentet fra ADC produsenten sin GitHub side, hvor det var et bibliotek som kan benyttes for å kommunisere med ADCene<sup>8</sup>. Noen endringer har blitt utført på koden i dette biblioteket. Biblioteket har en metode for avlesning av spenningen som blir målt. Denne metoden har blitt benyttet for å lese av faktiske verdier sensoren og shuntene måler. Eksempelvis avlesning av relativ fuktighet blir formelen for å konvertere fra avlest voltnivå seende slik ut:  $\frac{\text{lestverdi}}{5} * 100$ . Dette vil gi ut et tall fra 0 – 100% som tilsvarer den relative fuktigheten. Støyfiltreringen utføres i software, og ikke i hardwaren. Dette medfører at én sampling av data ikke vil være nok, men en mengde samplinger må bli utført. Støyfiltreringen blir da gjort ved at en rekke samplinger blir utført og alle de avleste verdiene blir lagt i en liste. Deretter sorteres listen slik at de avvikende verdiene havner i hver sin ende av listen. Et segment av verdiene som ligger midt i den sorterte listen blir plukket ut og gjennomsnittet av dette segmentet blir verdien som vil bli returnert til programmet som avlest verdi. Kildekoden ligger på minnepennen under navn "ABE\_ADCpi.py" og "ABE\_ADCCifferential.py", metodene som har blitt lagt til heter read\_winddirection(), read\_windspeed(), read\_temperature(), read\_rh(), read\_forbruk() og avg().

**Kommunikasjonen med charge controller** baserer seg på et eksempel fra GitHub<sup>9</sup>. Dette er modbus-kommunikasjon, og siden modbusadressene ligger åpent tilgjengelig på Tristars sine hjemmesider var det enkelt å få scriptet til å returnere de ønskede verdiene. Dette er direkte avlesning av minneadresser i charge controlleren, så her var det ikke behov for støyfiltrering. For mer detaljer ligger kildekoden på minnepennen under navnet read\_register.py".

**HTU21D** er den digitale sensoren som blir brukt for å måle temperatur og fuktighet innendørs. Denne sensoren benytter seg av I2C for å kommunisere med RPi. Det blir benyttet et bibliotek kalt "pigpio" som tillater kontroll over GPIO pinnene. For å benytte dette biblioteket må daemonen "pigpiod" kjøre i bakgrunnen. For å kunne kommunisere med daemonen må det bli opprettet et objekt av "pigpio". Objektet leser så på I2C adressen til sensoren, for så å sende en kommando ut på I2C bussen for å si til sensoren at den ønsker enten temperatur eller fuktighets data. Når RPi mottar dataene, er det bare å følge formlene i databladet for å få resultatet. Mer detaljer om kildekoden til modulen ligger på minnepennen under navnet "HTU21D.py".

**For å opprette kommunikasjon mellom RPi og brenselcellen** ble det kjøpt inn en RS-232 interface adapter. Brenselcellen er satt opp slik at den vil svare hvis den får tilsendt kommandoer via seriell kommunikasjon. Kommandoene den vil svare på, og hva den vil svare, er nøye beskrevet i manualen som følger med RS-232 adapteren. For å programmere denne prosessen ble det valgt å benytte den ferdige modulen pyserial. Her opprettes det enkelt og greit et seriell-objekt som får de korrekte konfigurasjonene som cellen krever, herunder baudrate, stopbit, port m.m. Deretter

---

<sup>8</sup>ADC python bibliotek [https://github.com/abelectronicsuk/ABElectronics\\_Python3\\_Libraries](https://github.com/abelectronicsuk/ABElectronics_Python3_Libraries)

<sup>9</sup>Bibliotek for charge controller <https://github.com/brocktice/tristar-python-modbus>

benyttes et såkalt `textIOWrapper`-objekt til å sende og lese verdier. Brenselcellen kan også styres og konfigureres over seriell kommunikasjon. I hovedprogrammet leses det kun av verdier som batterispenning og drivstoffnivåer. Mer detaljer om kildekoden finnes på minnepennen under navnet "lesCelle.py".

Det er likevel mulig for systemadministrator å styre brenselcellen ved å benytte styringsprogrammet. Her er det programmert en terminal-modul som sender verdiene systemadministrator ønsker, og skriver ut svarene cellen gir. Igjen henvises det til adapterens manual <sup>9</sup> for å se hvilke kommandoer som er tilgjengelige. Dette systemet finnes også på minnepennen under navnet "Hessdalen.py".

**For styring av enheter** er det en modul som heter "on\_off.py" som håndterer logikken for av/påstyring av fysiske enheter i systemet. Slik det er beskrevet i avsnittet om oppkobling styringssystem 4.3.1 kan noen av enhetene slås av og på ved hjelp av optokoblere koblet mellom GPIO-pinnene på RPi'en og enhetene selv. For at pinnene skal kunne sende ut et høyt eller lavt signal må de bli konfigurert for dette. For at en pinne skal kunne konfigureres må det være mulig å referere til de ulike pinnene. Det er to måter for å finne ut hvilken "ID" en GPIO-pinne har. Den ene metoden er å bruke "GPIO NUMBERING", men det gir lite mening for andre enn datamaskinen. Den andre metoden er såkalt "PHYSICAL NUMBERING", denne metoden gjør det lettere å identifisere pinnene fysisk da det bare er å telle pinnene for å finne ut hvilken "ID" de har<sup>10</sup>. Når det er mulig å referere til en pinne, må man sette opp at det skal være en output pinne og ikke input. Da dette har blitt gjort er det bare å sende true eller false ut på pinnene som kan styre optokoblerne. For mer detaljer ligger kildekoden til modulen på minnepennen.

**Alarm og sammenligningssystemet** er en modul som sammenligner de målte verdiene med de forrige målingene. Måten dette blir utført på er at absoluttverdien av differansen mellom nåverdien og den forrige blir sammenlignet med en fornuftig grense. Er absoluttverdien større enn grensen vil en alarm-teller øke med 1. Er den innenfor grensen vil telleren nullstilles. Hvis alarm-telleren når en fastsatt verdi, vil en alarm utløses og varsle eieren av systemet med en fornuftig melding. Det er en alarmteller for hver variabel som holder på informasjon som kan nå kritiske grenser. Dermed vil systemet varsle systemadministrator hvis unormale verdier blir målt. I denne modulen ligger det også en funksjon som tar i mot en melding, for så å sende denne meldingen til systemadministratoren. Denne funksjonen benytter seg av "sendmail", som er en mangesidig e-post server som støtter flere overførings- og leveringsmetoder for e-post. I dette eksempelet blir det benyttet "SMTP". For mer detaljer ligger kildekoden til modulen på minnepennen under navnet "alarmOgSammenligne.py".

**Database logging** er en modul som tar seg av databasetilkoblingen og innsetting av data i databasen. For å kommunisere med MySQL-databasen, blir python biblioteket `pymysql` brukt. Her blir det først opprettet en kobling mot databasen, så blir det utført en spørring eller innsetting i databasen og tilslutt må koblingen avsluttes. Siden både det autonome hovedprogrammet og styringsprogrammet som er gjort tilgjengelig for systemadministrator kan kommunisere med databasen, er det en viss fare for at begge programmene forsøker å lese av eller endre verdier samtidig. Derfor utføres all kommunikasjon med databasen i transaksjoner slik at databasen tar seg av rekkefølgen ulike operasjoner blir utført i, og unngår kollisjoner og bevarer databasens

---

<sup>10</sup>RPI GPIO <https://www.raspberrypi.org/documentation/usage/gpio/>

konsistens. For mer detaljer ligger kildekoden for denne modulen på minnepennen under navnet ”loggDB.py”.

**Configfil** var et ønske fra oppdragsgiver. En configfil gjør det enkelt å endre på innganger, utganger, porter og lignede som systemet benytter i koden. Dermed vil det være enklere å konfigurere systemet uten å måtte sette seg inn i koden som er skrevet. Dette ble løst ved å bruke en modul i python som heter ”config parser”. Denne modulen gjør det mulig å lese en configfil, for så å hente ut seksjoner med nøkler og verdier. Konfigurasjonsfilen finnes på minnepennen under navnet ”konfigurasjon.ini”.

#### 4.3.4 Overstyring


```
#####
#
#Velkommen til Linux Based Off Grid Reasearch Facility eller LBOGRF som vi #
#liker å kalle det. Dette er brukerprogrammet for stasjonen og her kan du #
#se hvilke grenseverdier som er satt for systemet og flere forskjellige #
#stater på enheter. Det er også mulig å endre på disse verdiene og endre #
#om programmet skal få lov til å skru av eller på ulike elementer. Dette er #
#er et enkelt program som forteller deg akkurat hva du skal gjøre. #
#OBS! Blir noe overstyrt vil den sette auto_mode=False #
#
#####

Velg 0 for å avslutte programmet
Velg 1 for å sette verdier på brenselcellenil
Velg 2 for å sette grenseverdier for vindmålesystemet
Velg 3 for å sette grenseverdier kamerasytemet
Velg 4 for å overstyre vindsystemet
Velg 5 for å overstyre kamerasytemet
Velg 6 for å se verdiene som er satt
Velg 7 for å sette alle verdier i auto mode

Valget ditt: █
```

Figur 4.11: Et bilde som illustrerer menyen for overstyringsprogrammet.

Det kan være at det oppstår en situasjon hvor systemadministrator ønsker å overstyre det autonome programmet og sette verdier manuelt. Derfor har det blitt laget et manuelt styringsprogram som kan startes fra terminalen. Dette krever at systemadministrator logger seg inn via en ssh-tilkobling. Hver parameter som kan styres har en egen unik boolsk attributt i databasen kalt `auto_mode`. Når det manuelle styringsprogrammet benyttes til å sette en parameter vil denne `auto_mode` parameteren settes til `false`. Det autonome hovedprogrammet sjekker alltid om en parameter den ønsker å endre står i `auto_mode = true` før den gjør endringer. Dermed vil manuelt satte parametere være utilgjengelig for det autonome programmet. Det vil ikke være mulig for det autonome programmet å endre parameteren før systemadministrator har satt parameterens ”`auto_mode`” tilbake til `true` via styringsprogrammet. Viktig å merke seg er at selve endringen av `auto_mode` statusen endres automatisk til `false` når en ”on\_off” parameter settes manuelt. Terminalprogrammet som har blitt utviklet har også en egen funksjon for å sette alle parametre i `auto_mode = true`. Dermed er det forholdsvis enkelt å sette systemet tilbake i full auto. Figur 4.11 viser hvilke valg som er mulig i det manuelle styringsprogrammet, det har et valg for brenselcellen selv om dette valget ikke vil utføre noen handling. Det har blitt tatt med fordi det kan tenkes at en gang i fremtiden skal systemet også styre brenselcellen. Figur 4.12 viser hvordan control databasen ser ut, og hvilke parametere som tilgjengelige.



Enhet	MaxVerdi	MinVerdi	Av/paa	Auto_mode
brenselcelle	29.0	24.0	1	1
varme_kamera	3.0	1.0	0	1
varme_vind	3.0	1.0	0	1

Vil du gå tilbake til menyen? [J/n]

Figur 4.12: Et bilde som illustrerer databasen for overstyringsprogrammet.

### 4.3.5 CRON-jobber

Cron er navnet på det programmet som lar unixbrukeren kjøre kommandoer eller script automatisk på gitte tidspunkter/datoer. Vanlige bruksområder er systemadministrative oppgaver som backup<sup>11</sup>. I dette prosjektet blir det brukt for planlagte operasjoner hvor RPi skal kommunisere med en annen server, utføre en kontrollsjekk eller utføre filbehandling.

For å kommunisere med eksterne servere må ssh benyttes. For å unngå å sende brukernavn og passord i klartekst ble det opprettet ssh-nøkler ved å benytte kommandoen «ssh-keygen» i terminalen på RPien. Det ble ikke satt noe passord da dette vil vanskeliggjøre den automatiske kontakten som gjøres via cronjobber. For å laste opp nøkkelen til serveren ble kommandoen «ssh-copy-id brukernavn@ip» benyttet. Ved en ssh-tilkobling behøves derfor ikke brukernavn og passord etter at nøkkelen har blitt overført.

**MySQL-dump** er en funksjon som sender en kopi av databasen på RPien til databasen på serveren ved HIØ. Denne handlingen er lagret i en CRON-jobb som utføres hvert 30. minutt.

**Ruteren** har mulighet for å bli restartet digitalt, og siden det er en mysql dump som går hvert 30. minutt er det greit å kjøre en sjekk på om systemet har tilgang på internett eller ikke. Det kan godt tenkes at ruterens låser seg og da vil ikke systemet klare å dumpe data til det eksterne lagringssystemet. Det har derfor blitt utviklet et script som sjekker ruterens. Dette scriptet blir lagt inn i crontab og kjører også hvert 30. minutt. Dette scriptet benytter seg av et verktøy i Linux-shellet som heter fping. Dette er et verktøy som pinger en IP adresse 8 ganger og returnerer True eller False. Scriptet har blitt bygd opp slik at det først pinger ruterens for å se om RPi har kontakt med ruterens, og så pinger den 8.8.8.8 som er DNS serveren til Google. Hvis scriptet returnerer False på en av disse testene vil den benytte en GPIO-pinne for å omstarte ruterens.

**En del av loggfilene** på Raspberry pi skal vises frem i webgrensesnittet, for å gjøre dette må filene flyttes til serveren ved HIØ. For å overføre rapporter til serveren blir en Linux-funksjon kalt "rsync" benyttet. Det er totalt fire CRON-jobber for overføring av loggfiler.

**Fordi loggingen** som blir utført i hovedprogrammet legger til hendelser nederst i loggfilene blir de eldste hendelsene lagres øverst, derfor må innholdet reverseres før sending. For å reversere innholdet benyttes "tac" hvor utskriften skrives til en reversert fil som så sendes.

<sup>11</sup>Definisjon CRON <http://www.unixgeeks.org/security/newbie/unix/cron-1.html>

**Pigpiod** er en daemon som må kjøres i bakgrunnen for at hovedprogrammet skal kunne kommunisere med den digitale sensoren. For å sørge for at dette programmet kjøres legges det inn en cronjob som starter denne daemonen ved oppstart.

**Hovedprogrammet** må startes som en egen prosess slik at det kjøres uavhengig av hvilken bruker som er innlogget. I Rasbian så er "huponexit" satt, noe som vil si at hvis en bruker logger ut vil alle prosesser som denne brukeren har startet avsluttes<sup>12</sup>. Derfor må hovedprogrammet starte ved oppstart slik at det er root brukeren som eier prosessen.

### 4.3.6 Watchdog

En "watchdog" er en funksjon som gjør at RPi kjører en restart dersom operativsystemet krasjer. "Watchdog"-programmet sender et signal til /dev/watchdog hvert tiende sekund, får den ikke svar blir det gjort en omstart. "Watchdog"-funksjonen har blitt testet med en såkalt "fork bomb". Det er en kodelinje som krasjer systemet og trigger "watchdoggen".

#### 4.3.6.1 Ruter

Ruteren som benyttes i dette prosjektet er ASUS 4G-N12 Wireless N300 LTE Router. Den benytter det mobile nettverket til Telenor og simkort for dette er blitt tildelt av oppdragsgiver. Nettverket har en hastighet på opp til 300 Mbps, men er begrenset til 10GB hver måned. Når denne grensen er nådd senkes datahastigheten. Basert på erfaring av hvor mye data som genereres og hvor mye som skal sendes vil ikke dette bli et problem.

## 4.4 Det eksterne lagringssystemet

Nevnt i seksjonen om den Overordnede systembeskrivelsen 2 er det en server som skal være på Høgskolen. Som beskrevet i mulige løsninger 2.4.1 er dette en Linux maskin med database av type MySQL og Webserver av type Apache installert. Utover dette er phpMyAdmin installert. Dette er en gratis programvare skrevet i PHP for å håndtere administrasjon av en MySQL database installasjon<sup>13</sup>.

### 4.4.1 Lokal database på Raspberry Pi

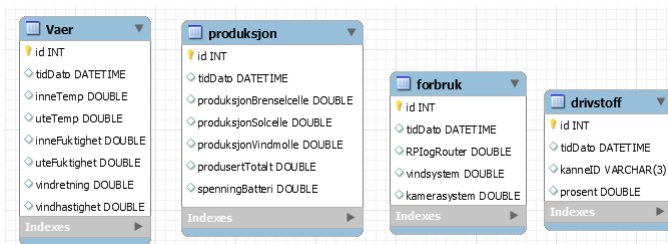
Databasen er bestående av fire tabeller: vaer, produksjon, forbruk og drivstoff vist i figur 4.13.

På grunn av begrenset lagringskapasitet er det ønskelig å utføre rutinemessig sletting i databasen på RPi. Dette blir gjort ved bruk av en scheduled event som kjører en gang i året og sletter den første halvdel av den lagrede dataen. Det ble vurdert å gjøre det oftere, men siden dataen tar så liten plass er det ikke hensiktsmessig.

**Vaer-tabellen** skal holde alle værrelaterte måleverdier. Den vil motta verdier for temperatur ute og inne, luftfuktighet ute og inne samt vindhastighet og retning. Disse værparameterne skal hentes inn og lagres hvert 10. minutt. Hver gang verdier blir satt inn i tabellen vil de få en tilhørende heltalls-ID og et tidsstempel som forteller når målingen ble gjort. ID-kolonnen holder på datatypen

<sup>12</sup>Prossesser i linux <https://askubuntu.com/questions/685281/why-my-process-is-still-running-after-i-log->

<sup>13</sup>phpMyAdmin: <https://www.phpmyadmin.net/>



Figur 4.13: De fire tabellene.

integer og er lagret med funksjonen ”auto increment”. Ettersom tidsstempel går ned på sekundnivå er det hensiktsmessig å bruke en heltalls-id for lettere å kunne hente ut bestemte rader.

**Produksjon-tabellen** skal holde verdier for energiproduksjon. Hvert 10. minutt blir det lagret data om hvor mye brenselcellen, solcellene og vindmølla produserer, i tillegg til batterispenningen. ID og datoTid-kolonnene fungerer på samme måte som i værtabellen.

**Forbruk-tabellen** skal holde alle verdier for hvor mye de ulike enhetene bruker. Tabellen vil holde på forbruket til RPi, router, oppvarming av kamerahus, kamera, sensor for vindretning, sensor for vindhastighet. Det vil bli gjort en måling hvert 10. minutt, ID og tidDato-kolonnene vil fungere likt som i værtabellen.

**Drivstoff-tabellen** lagrer informasjon om hvor mye drivstoff som er igjen i de fire drivstoffkannene i systemet. Det lagres en heltalls-ID og et tidsstempel sammen med kannaID og hvor mye drivstoff som er igjen i kanna målt i prosent.

Alle scheduled events og create table setninger for databasen på RPi ligger lagret i filen Hessdalen.sql på vedlagt minnepenn.

#### 4.4.2 Ekstern database ved HiØ

Denne databasen skal motta data fra databasen på RPi. Dataen som mottas skal lagres i nye tabeller som skal inneholde historisk data.

For å flytte data fra databasen på RPi til databasen ved HIØ brukes funksjonen ”mysqldump”. Med denne kommandoen kan man enkelt overføre alle tabeller fra en database til en annen over Internett på en sikker måte. Det er planlagt å gjøre denne prosedyren en gang hver halvtime.

For hver gang det blir gjort en ”mysqldump” blir det lagret fire nye versjoner av tabellene ”vaer”, ”produksjon”, ”forbruk” og ”drivstoff” i databasen ved HIØ. Det blir ikke lagret duplikater og eldre versjoner av tabellene vil bli overskrevet av de nye tabellene.

Fordi databasen på RPi blir rutinemessig ryddet, vil den ikke holde på historisk data fra lang tid tilbake. For å lagre historisk data blir det opprettet tre nye tabeller i databasen ved HIØ, ”h\_vaer”, ”h\_prod”, ”h\_forbruk” og ”h\_drivstoff”. Det blir brukt ”scheduled events” til å lagre historisk data i disse tabellene. Disse ”eventene” vil flytte ny data inn i de historiske tabellene hver gang det blir gjort en ”mysqldump”.

Alle scheduled events og create table setninger for databasen på RPi ligger lagret i filen HIOF.sql på vedlagt minnepenn.



## 4.5 Webgrensesnitt

Webgrensesnittet er laget for å kunne hente ut og vise data fra databasen på den eksterne serveren. Dataene som blir vist er delt opp i fire deler; Vær, energiproduksjon, energiforbruk og statusrapporter. Vær innebærer målinger fra sensorer både ute og inne. Værdelen av webgrensesnittet består av to filer, en for grafer og en for tabeller. Filene er vedlagt på minnepenn med navnene vaergraf.php og vaertab.php. Som vist i figur 4.14 viser tabellene dataene grafene baserer seg på, og kan benyttes til å se måleverdier ved spesifikke tidspunkt.

### Værdata

#### Temperaturmålinger

Dato og tid	Temperatur inne	Temperatur ute
2016-04-19 08:38:50	17	8
2016-04-19 09:10:00	18	9
2016-04-20 12:39:57	19	12
2016-04-20 20:47:12	23	6
2016-04-20 21:00:27	22	5
2016-04-21 09:39:18	20	8
2016-04-21 10:16:49	20	8
2016-04-21 11:00:02	20	9
2016-04-21 12:02:47	21	9
2016-04-21 13:41:42	21	9
2016-04-21 14:03:18	21	10
2016-04-21 16:06:56	23	10
2016-04-21 18:02:46	24	12
2016-04-21 19:00:14	24	12
2016-04-21 21:08:05	23	8
2016-04-21 22:03:12	21	7
2016-04-21 23:19:55	22	6

Temperaturtabell  
 Fuktighettabell  
 Vindtabell  
 Nå data

Figur 4.14: Tabell som viser temperaturmålinger

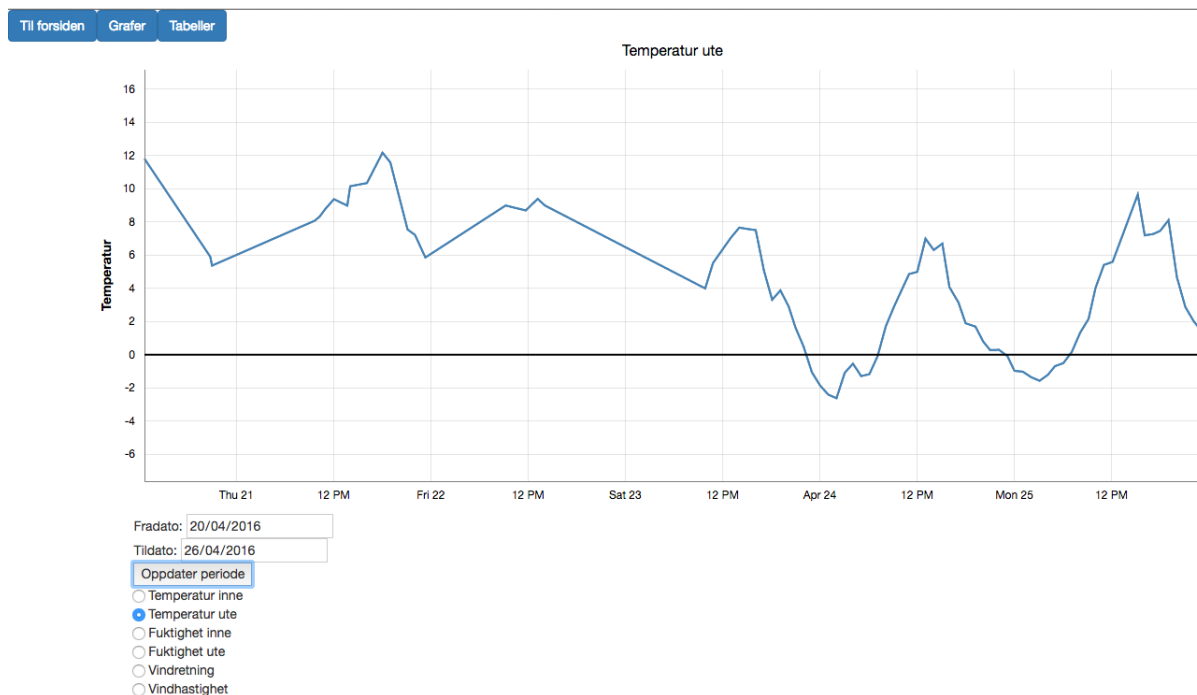
Grafene for vær ligger lagret i en felles fil hvor man kan se på alle de ulike værmålingene. Grafene er skalerbare og har et rutenett i bakgrunnen som gjør det enklere å lese dataene.

Delen om energiproduksjon viser det som blir produsert fra solcelle, vindmølle og brenselcelle. Produksjonsdelen har samme oppbygning som værdelen og består av en side for grafer og en for tabeller. Disse filene er vedlagt på minnepenn med navnene prodgraf.php og prodtab.php. I denne delen kan man se hvor mye energi de forskjellige enhetene i systemet produserer. I tillegg til dette er det lagt inn beregninger som kalkulerer og presenterer hvor mye systemet produserer totalt. Et eksempel på en graf vises i figur 4.15.

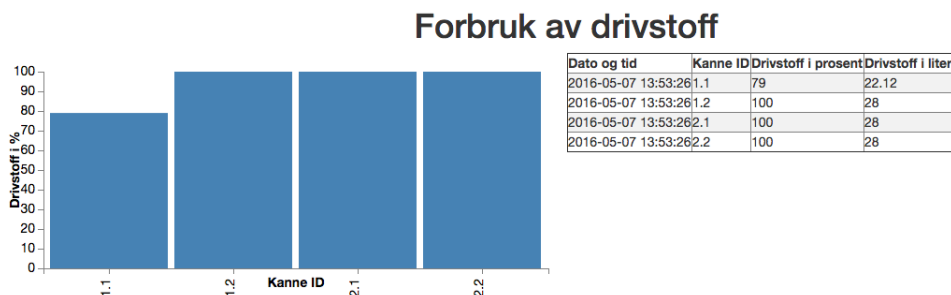
Energiforbruk viser forbruket for alle enheter som er tilkoblet systemet. Forbrukdelen består også av en side for grafer og en side for tabeller, men har i tillegg en side som presenterer drivstoffstatusen til brenselcellen. Filene for energiforbruk ligger vedlagt på minnepenn med navnene forbrukgraf.php, forbruktab.php og drivstoff.php. Grafene og tabellene tar for seg hvor mye strøm som blir brukt av de forskjellige enhetene.

Siden for drivstoffstatus inneholder et stolpediagram med fire stolper vist i figur 4.16, hvor hver av stolpene representerer en drivstoffkanne som er koblet til brenselcellen. Y-aksen i dette diagrammet viser hvor mange prosent av drivstoffet som er igjen på hver av kannene. Ved siden av stolpediagrammet er det lagt inn en tabell som viser tidspunktet de brukte målingene ble tatt, samt drivstoffstatus i prosent og liter.

Rapportdelen består av fire loggfiler som inneholder informasjon om alarmer, GPIO, modbus og brenselcellen. På alarmsiden ser man en logg over uønskede hendelser i systemet. GPIO-siden viser om enheter blir skrudd av eller på ved hjelp av GPIO-pinnene. Modbus-siden viser spesielle



Figur 4.15: Graf som viser temperaturmålinger

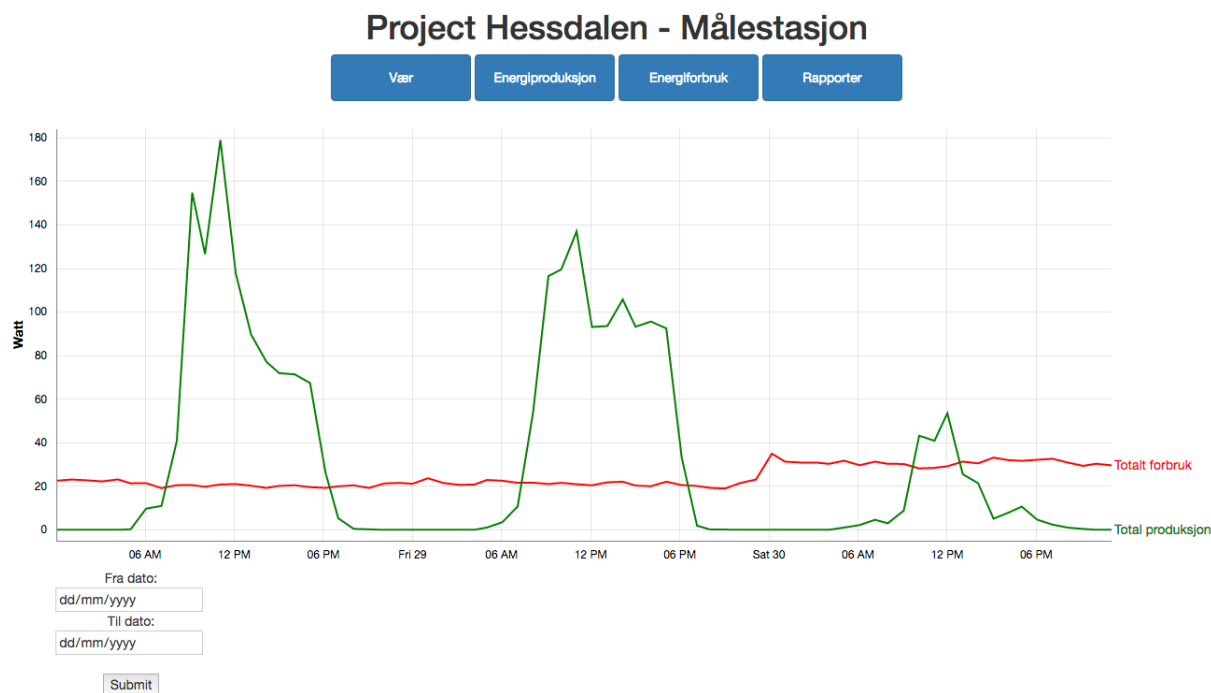


Figur 4.16: Stolpediagram for visning av drivstoffniåv

hendelser i forbindelse med modbus. Brenselcelle-siden viser en logg over uvanlige hendelser i brenselcella.

Det er forsøkt å holde grensesnittet så simpelt som mulig, samtidig som dataene blir vist på en oversiktlig og god måte. Figur 4.17 viser forsiden til webgrensesnittet og er det første brukeren ser. Graf tar for seg forholdet mellom det totale forbruk og produksjon.

For uthenting av data tas det utgangspunkt i databasen lokalisert ved HiØ, altså de data som blir dumpet fra databasen på RPi hvert 30. minutt. Selv om det er utgangspunktet, så er det lagt inn en direkte kobling mot RPi sin database. Denne koblingen samler siste målinger i en felles tabell kalt "Nå data". Ved å ha denne ekstra tilkoblingen kan en få de siste dataene hvis det skulle være et behov for det. I tillegg medfører det en form for test mot RPi. Hvis ingen data blir vist er tilkoblingen brutt. Da kan det være enten feil med selve RPi, RPi sin tilkobling til ruter eller ruter mot Internett.



Figur 4.17: Forsiden av webgrensesnittet

I utviklingen av webgrensesnittet har det blitt benyttet HTML, PHP og javascript. PHP har blitt brukt for å koble seg til databasene, hente ut data som skal brukes og lagre denne i et format som kan brukes i javascript. Javascript er brukt til å presentere dataene i grafer og tabeller. For å lage grafene har det blitt benyttet et javascript-bibliotek kalt D3.js<sup>14</sup>. D3.js er et verktøy som er spesiallaget for å visualisere data på nettsider. Koden for webgrensesnittet er lagt inn på minnepennen under filene som slutter '.php'.

<sup>14</sup>D3 bibliotek: <https://d3js.org/>



## Kapittel 5

# Testing

I dette prosjektet har det blitt utført systematiske tester. Testingen har startet med "toppen" av hele systemet, altså webgrensesnittet. Systemet blir så testet gradvis nedover helt til fysiske koblinger mellom enhetene i målestasjonen. Testresultatene ble vurdert og eventuelle endringer eller forbedringer ble utført etter vurderingen. Dette har da vært en iterativ prosess slik at systemet blir optimalisert best mulig. Det har blitt simulert ulike scenarier eller feil som kan inntreffe i systemet. Simuleringer av disse hendelsene har blitt utført for å teste hvordan systemet håndterer disse hendelsene. Det har under hele utviklingsprosessen foregått uformelle tester av enheter, men dette blir ikke utredet her med mindre det var en spesiell hendelse.

### 5.1 Førsterunde teknisk testing

Dette avsnittet vil være bygd opp slik at det først kommer en tabell som tilhører en del av systemet. I tabellen vil kolonnen til venstre forklare testen eller situasjonen, og kolonnen til høyre vil vise resultatet. Resultatet kan bli svart på to måter. Den ene metoden er enten ved JA eller NEI, og den andre er ved enten BRA, GREIT eller DÅRLIG. Resultatsvaret kommer an på hvordan feilen eller scenarieret er formulert. Under tabellen vil det komme et avsnitt som forteller på hvilken måte testene har blitt utført og hvordan resultatene ble funnet. Dersom en mulig løsning blir funnet vil løsningen bli beskrevet og eventuelt utført.

**Testing webgrensesnitt** (tabell 5.1):

Test	Resultat
Fungere webgrensesnittet med rpi nede?	Greit
Nå data	Dårlig

Tabell 5.1: Testtabell for webgrensesnitt.

I brukertesten har en gruppe med mennesker fått tilgang til nettsiden og trykket rundt og testet den. De har gitt prosjektgruppen tilbakemelding på brukervennligheten til webgrensesnittet. Tilbakemeldingene var stort sett gode, men det kom frem noen svakheter i løsningen. For å manøvrere mellom ulike tabeller er det brukt radio buttons, noen brukere opplevde at tabellene forsvant dersom de klikket flere ganger på samme knapp. Grunnen til dette er at når brukeren klikket på knappen, ble en if-test som sjekker synligheten på det valgte elementet aktivert. Det viste seg at denne if-testen var unødvendig, den fungerte som en toggle som ikke er hensiktsmessig i dette tilfellet. Løsningen på dette problemet er å fjerne denne if-testen. Dette er nå i orden.

Et annet problem noen brukere hadde, var om visse datoer ble valgt så fungerte ikke grafene. Grunnen til at dette problemet oppstod er fordi det ikke blir utført noen validering av input. Dette er nå i orden.

Webgrensesnittet støtter funksjonalitet for å hente ut data direkte fra målestasjonen. Et problem var at dersom RPi var nede ville ikke sidene for fremvisning av tabeller fungere, slik det kommer frem i tabell 5.1. Dette fordi et PHP-script prøvde å opprette tilkobling til RPi og aldri gikk videre. Dette problemet kan løses ved å legge til en if-test som sjekker om tilkobling kan opprettes, dersom det ikke er mulig så vil ikke scriptet koble seg til. Det skal nå være i orden.

**Database på ekstern server (tabell 5.2):**

Test	Resultat
Håndterer den at internettforbindelse er nede?	Ja

Tabell 5.2: Testtabell for database på ekstern server.

Overføring av data fra databasen på RPi til databasen på den eksterne serveren blir gjort med en funksjon som kalles mysqldump. Denne operasjonen blir utført hver halvtime ved hjelp av en cron-jobb på RPi. Rad 1 i tabellen 5.2 tar for seg situasjoner der denne overføringen ikke skjer som planlagt. Testen er gjort ved å kutte internettforbindelsen i en gitt periode og sjekke om all data blir overført riktig etter at forbindelsen er gjenopprettet.

I databasen på den eksterne serveren er det lagret scheduled events som fører over dataene som mottas fra RPi til tabeller for historisk data hver time. Disse jobbene baserer seg på tidsstempel og vil bare overføre data med tidsstempel som er nyere enn det som finnes i de historiske tabellene fra før. Det vil si at dersom internettforbindelsen mellom RPi og ekstern server kuttes i en gitt tidsperiode, vil de lagrede jobbene finne og overføre all data som ikke har blitt mottatt i denne tidsperioden når forbindelsen er oppe og går igjen. Systemet har håndtert testingen på en god måte, konklusjonen er derfor at systemet håndterer slike feil godt.

**Database logging på RPi (tabell 5.3):**

Test	Resultat
Håndterer databasens NULL verdier fra hovedprogrammet? (Ikke kontakt med enheter)	Nei
Er dataene fornuftige i forhold til fysiske forholdene?	Ja

Tabell 5.3: Testtabell for database.

Rad 1 i tabellen 5.3 ble testet ved å sette en målingsvariabel til å være None, som er Null-objektet i python. Grunnen til at dette ikke fungerte var fordi sql-insetningen brukte en form for formatering som ikke forstod at None var det samme som Null. Dette ble løst med å formatere det slik at python tolker inputen i metoden som en tekststreng slik at None får samme betydning som Null. Rad 2 ble testet slik at observasjoner stemte overens med målingene. Dette ble bare gjennomført for vindhastighet og vindretning. Temperatur og vindhastighet ble testet og kontrollert opp mot værvarsel fra yr.no. Figuren 5.1 viser maks, min og gjennomsnittsverdien som systemet har målt siden 28 april til 2 mai, og figur 5.2 viser verdiene yr.no har målet i den samme tidsperioden. Dette viser samsvar mellom målingene og yr med et slingringsmonn på  $\pm 2^{\circ}\text{C}$  og

Dato	MaxTemp	MiddleTemp	MinTemp	MaxVindhastighet	MiddleVindhastighet
2016-05-02	12.7	7.5	1.7	7.1	1.8
2016-05-01	8.3	5.1	3.0	4.6	1.0
2016-04-30	7.4	4.1	2.3	6.1	0.4
2016-04-29	11.5	6.1	2.2	7.8	2.2
2016-04-28	10.8	4.6	0.3	4.2	0.9

Figur 5.1: Temperatur og vind måling fra systemet

$\pm 3m/s$ . Feilen i rad 1 var enkel å fikse og en ny test ble gjennomført umiddelbart med vellykket resultat.

Dato	Temperatur °C				Nedbør			Vind i m/s	
	Maks	Min	Middel	Normal	Akkumulert nedbør i mm (målt kl 7 for siste 24 timer)	Nedbørttype	Snødybde cm	Maks	Middel
2. mai 2016	12,6°	5,6°	8,9°	7,8°	0,0 mm	---	0 cm	5,9 m/s	3,6 m/s
1. mai 2016	10,3°	4,4°	7,1°	7,5°	19,5 mm	---	0 cm	5,1 m/s	3,2 m/s
30. april 2016	8,5°	3,1°	4,6°	7,3°	2,6 mm	---	0 cm	7,9 m/s	2,3 m/s
29. april 2016	10,3°	3,2°	6,7°	7,1°	3,5 mm	---	0 cm	7,7 m/s	4,9 m/s
28. april 2016	9,9°	1,9°	5,4°	6,9°	0,5 mm	---	0 cm	6,7 m/s	3,6 m/s

Figur 5.2: Temperatur og vind målingene fra yr.no <sup>1</sup>

#### Logging av feilmelding (tabell 5.4):

Test	Resultat
Blir hendelsene som inntreffer logget i riktig fil?	Ja
Gir loggfilene mening eller er de kryptiske?	Ja

Tabell 5.4: Testtabell for loggfiler.

Testen i rad 1 har blitt testet kontinuerlig i utviklingsprosessen. Det har blitt laget selvstendige loggfiler for de ulike delene av hovedprogrammet. Kontrollen har vært slik at de ulike delene sender logg til sin logg-fil. Det eneste problemet var at hovedprogrammet ble opprettet først og da ble den en form for root logg. Problemet med det var at den mottok loggene fra alle delene av systemet. Dette kom også ofte dobbelt, mye feilsøking har blitt utført her. Det ble derimot ikke funnet en god løsning, så det endte med at den ble værende, men ikke aktivt i bruk. Rad 2 ble kontrollert ved at loggfilene ble grundig lest og analysert. De fleste loggene var meningsfulle, men det var noen av exception-meldingene som ikke var intuitive. Et Google-søk var til hjelp for å tolke feilmeldingene. Disse er derimot veldig viktig å logge ved eventuelle feil på systemet.

#### Alarmer (tabell 5.5):

Testen i rad 1 ble utført ved å simulere flere forskjellige feil. Først ble det testet om alarmer gikk av ved en brå økning eller senkning av en måling. Dette ble utført slik at den etter 10 iterasjoner satt de ulike målingene til 10 mindre enn hva de faktisk måler. Dette ga umiddelbar effekt og alle alarmene gikk av. Det er verdt å merke seg at en slik test vil kun være fornuftig for temperatur, fuktighet, batterispenning og forbruk. De andre målingene kan variere ganske mye på kort tid, så de blir ikke vurdert her.

Test	Resultat
Går alarmene av når det faktisk er feil? (ikke falsk alarm)	Ja
Blir alarmen varslet til oppdragsgiver?	Nei

Tabell 5.5: Testtabell for alarmsystemet test 1.

Rad 2 ble enkelt testet ved å se om varselmailen ble sendt. Dette skjedde aldri fordi alarmtelleren aldri ble større enn 1. Grunnen til at telleren aldri ble større enn 1 var fordi slik alarmsystemet hadde blitt utviklet ville den for hver iterasjon kopiere over forrige måling til en sammenlignings dictionary. På denne måten var det veldig sjeldent at noen målinger hadde en større absolutt differanse enn 5 fra forrige måling 10 ganger på rad. Det ville ha vært dumt å ikke kopiere over den forrige målingen for hver test, siden det kan tenkes at det var et vendepunkt og da ville aldri alarmsystemet fungere etter det punktet. En mulig løsning derfor å sjekke om hver måling befinner seg i ytterkanten av sine grenser. Det vil være ekstremt sjelden at en måling er helt eksakt sin minimumsmåling eller maksimumsmåling 10 ganger på rad med et 10 minutters intervall mellom hver test. Dette vil derfor være en god indikasjon på om en enhet er nede eller ikke. Grunnen til at en None-verdi ikke vil fungere er fordi ADCene vil returnere 0 selv om det ikke er kontakt med sensoren, som vil si at programmet alltid får svar.

**Styring** (tabell 5.6):

Test	Resultat
Utfører den styring av enheter?	Ja
Er informasjon som hentes inn nok til å fornuftige valg?	Ja
Blir det tatt avgjørende styringsvalg dersom nødvendige data ikke er tilgjengelig?	Ja

Tabell 5.6: Testtabell for styring test 1.

Rad 1 ble testet på flere måter. Først ble det satt grenser som var overdrevne slik at systemet måtte skru de av og på. Dette ble logget i GPIO og alarmloggene. Dette i seg selv er ikke godt nok for å kvalitetssikre at systemet faktisk styrer enhetene. Derfor har det også blitt utført fysiske målinger med multimeter for å sjekke at spenningen i kretsene har falt og økt ved styring av de ulike enhetene. Dette har også kommet tydelig fram i forbruket i databasen.

Rad 2 har blitt testet opp mot det oppdragsgiver ønsker å styre. Dette henger mye sammen med testen i rad 1. Her ble også testen utført ved å sette grenser som fremprovoserer styring. Alt det som oppdragsgiver ønsker at hovedprogrammet skal styre ble styrt. Informasjonen for å ta avgjørelsene baserer seg på batterispenning og utetemperatur.

Testen i rad 3 er også sterkt knyttet til dette med informasjonen hovedprogrammet bruker. Her ble det simulert at utetempersensoren og målingen av batterispenningen feilet. Mangelfull måling av batterispenningen taklet systemet godt, siden den returnerte None, og dette kunne ikke bli brukt til å ta avgjørelser, noe som er et helt greit utfall ved en slik hendelse. For utetemperatur vil ADC alltid returnere 0 selv om den ikke får kontakt med sensor. Dette betyr at avlesningen av sensoren vil returnere  $-40\text{ }^{\circ}\text{C}$ . Hovedprogrammet godtar dette og skrur varmen på i alle enhetene. Dette er ikke alltid fornuftig, så her bør utetemperaturen settes til å være None hvis alarmtelleren



når 10. Da bør også alle optokoblere settes lave slik at den innebygde mekanismen i de ulike enhetene styrer om varmen skal være på eller av.

**Overstyring** (tabell 5.7):

Test	Resultat
Funger overstyringen slik at den låser hovedprogrammet fra å gjøre endringer?	Ja
Hva skjer når flere brukere prøver å endre en verdi i databasen samtidig?	Bra
Brukertest	Greit

Tabell 5.7: Testtabell for overstyring test 1.

Rad 1 ble testet ved at `auto_mode` først ble satt til å være `False` for å hindre programmet fra å endre noe. For å virkelig teste at hovedprogrammet ikke gjennomførte en styring ble helt urimelige grenseverdier satt på enheter som skal styres. Programmet gjorde ingenting, men med engang `auto_mode` ble satt til `True` slo den av enhetene.

Rad 2 ble testet ved at flere brukere var innlogget og prøvde å endre på samme ting samtidig. Dette foregikk samtidig som hovedprogrammet kjørte i bakgrunnen. Det var ingen problemer med dette, den siste endringen ble stående som den gjeldene. Det fungerer siden database innsettingen er en transaksjon i dette systemet, så transaksjoner blir ordnet etter tur og orden. Brukertesten ble utført av flere utenforstående og oppdragsgiver. De utenforstående klarte enkelt å bruke den, men forstod ikke så mye hva de gjorde. Oppdragsgiver skjønnte mer siden han var mer satt inn i systemet. Han så bare et problem med systemet. Problemet var at hvis han gjør en feil i parameterendringene må han endre alt på nytt. Dette er rettet opp slik at systemet gir en mye mer dynamisk kontroll av parameterne.

**Kommunikasjon** (tabell 5.8):

Test	Resultat
Kommunikasjon med CC	Bra
Kommunikasjon med brenselcelle	Greit
Når begge enhetene er nede	Greit

Tabell 5.8: Testtabell for kommunikasjon test 1.

Her har det blitt utført fysiske tester hvor koblingene mellom enhetene og RPi har blitt dratt ut. Først ble det testet hvordan hovedsystemet taklet at den mistet kontakten til charger controlleren. Hovedprogrammet stoppet ikke opp, og feilmeldinger ble logget til modbus filen. Databasen logget `Null` for de verdiene som ble hentet ut fra enheten, så dette var vellykket.

Testen for brenselcellen fikk en feilmelding med engang den ble dratt ut. Dette fordi det var en test som hadde lurt seg på utsiden av `try`-blokken i koden. Etter at denne ble lagt inn i `try`-blokken gikk det bra. Programmet stoppet ikke opp og databasen logget bare `Null` for de verdiene som skal hentes ut fra brenselcellen.

Den siste testen var å dra ut begge koblingene samtidig. Hovedprogrammet stoppet ikke opp, men det var noen uforutsette målinger som gikk tapt. Dette var forbrukmålingene. Grunnen til at disse ble `Null` var fordi at programmet leser av batterispenningen for å få en best mulig tilnærming

på forbruket til de forskjellige enhetene, men siden både brenselcellen og charger controlleren var nede fikk den ikke tak i den verdien. En mulig løsning på dette er å utføre en test om den får tak i verdien eller ikke, og hvis den ikke gjør det, så setter vi den til å være 24 volt. Dette er den verdien systemet skal ligge rundt og vil derfor kunne gi akseptable verdier.

## 5.2 Andrerunde teknisk testing

Avsnittene i denne seksjonen vil ha samme oppbygging som i førsterunde med teknisk testing som vist i tabell 5.1. Avsnittene vil kun bestå av de delene som ikke oppfylte kravene sine i førsterunde med teknisk testing. Her vil det bli gjort videre testing for å se om endringene ga det resultatet som var ønskelig eller tester for å få en grundigere analyse av problemet.

**Alarmer** (tabell 5.9):

Test	Resultat
Blir alarmen varslet til oppdragsgiver?	Ja

Tabell 5.9: Testtabell for alarmsystemet test 2.

Etter de foreslåtte endringene i evalueringen i seksjonen til alarm 5.1 fungerte alarmsystemet slik at alarmtelleren også oppdaget at en sensor var nede. Skjedde dette 10 ganger på rad sendte det en mail til oppdragsgiveren. Denne grensen kan oppdragsgiver justere etter eget ønske i configfilen.

**Testing webgrensesnitt** (tabell 5.10):

Test	Resultat
Input test	Greit

Tabell 5.10: Testtabell 2 for webgrensesnitt.

For å teste dette har det blitt gjort forskjellige kombinasjoner og størrelser av dato input. Når begge datoene var innenfor tidsperioden hvor det har blitt gjort målinger fungerte det som det skulle. Når fra-datoen var før tidsperioden med målte verdier viser grafen data fra første måling til og med til-dato. Hvis fra-dato er innenfor tidsperioden med målte verdier og til-dato er etter viser grafen data fra og med fra-datoen til og med siste måling. Når fra-datoen er før måleperioden og til-datoen er etter viser grafen data for alle målinger. Hvis fra-dato er større enn til-dato vil det ikke vises noen graf. Hvis hele intervallet mellom fra og til-dato er utenfor intervallet med målinger vil ikke grafen vises. Grafene fungerer så lenge en av datoene er innenfor intervallet med målt data og fra-dato er mindre enn til-dato. Problemet med at fra-dato må være mindre enn til-dato kan sikres ved bruk av en if-test som returnerer en feilmelding dersom dette ikke stemmer. I tillegg kan det brukes en if-test som sjekker at fra-dato er innenfor intervallet med måldata. Dette har blitt utført og nå får brukeren en feilmelding som sier at valget deres var utenfor et område med målte verdier.

**Styring** (tabell 5.11):

Test	Resultat
Blir det tatt avgjørende styringsvalg dersom nødvendige data ikke er tilgjengelig?	Nei

Tabell 5.11: Testtabell for styring test 2.

Etter de foreslåtte endringene i evalueringen i seksjonen til styring 5.1, vil programmet oppdage at en avgjørende enhet er nede og sette den målte verdien til den aktuelle enheten til å være None. På denne måten blir ikke noen valg tatt i hovedprogrammet siden den ikke klarer å sammenligne None med en grenseverdi. Hovedprogrammet vil logge hendelsen og si hvilken enhet som er nede.

**Overstyring** (tabell 5.12):

Test	Resultat
Brukertest	Bra

Tabell 5.12: Testtabell for overstyring test 2.

De mulige løsningene til systemet som ble nevnt i seksjonen til overstyring 5.1 har blitt utført. Det har også blitt utført nye brukertester for å teste om systemet fungerte slik oppdragsgiver ønsket det. Det har blitt et mye mer fleksibelt og brukervennlig system etter gjennomførte endringer.

### 5.3 Testing som ikke har blitt utført i dette prosjektet

Det var planer om å utføre en form for stresstest i dette prosjektet. En effektmotstand vil kunne simulere forbruket for flere enheter, og på den måten kan det testes hvor mye oppdragsgiver kan belaste systemet uten at det svikter. Det må også bli utført en lang og strukturert test om vinteren hvor sol-produksjonen ikke er like bra, da vil systemet være mer avhengig av brenselcellen. Det er da viktig å teste hvordan systemet vil oppføre seg på vinterstid. Det kan godt tenkes at noen endringer i prioritetslisten eller i alarmsystemet kan gjøre at den vil være enda mer energisparende. Kanskje programmet trenger flere operasjonsmoduser, som for eksempel sommer\_mode og vinter\_mode. Dette er tester og muligheter som kan videreføres til andre prosjekter.



# Kapittel 6

## Diskusjon

I dette kapitlet vil det bli drøftet i hvor stor grad målene for prosjektet har blitt oppnådd. Prosjektets leveranser vil bli gjennomgått og evaluert. Metoden som har blitt tatt i bruk blir analysert og eventuelle forbedringer vil reflekteres over. Kapitlet vil ta for seg selve oppgaven, samt de positive og negative sidene ved prosjektet. Til slutt vil det være en oppsummering som tar for seg hva som kunne blitt gjort annerledes.

### 6.1 Formål og hovedmål

I underkapittel 1.4 sier hovedmålet at prosjektet skal kunne gi grunnlag for å undersøke i hvor stor grad en målestasjon drevet av dette energisystemet kan være selvforsynt på energi. Dette målet er i stor grad nådd da systemet er ferdigstilt og klart for å testes i henhold til målet.

Videre sier også hovedmålet at testingen skal kunne gi svar på begrensningene målestasjonen har på bruk av komponenter som krever energi. Slik systemet er laget, så vil energiforbruket og produksjonen bli logget til en database. Dataene i denne databasen blir sendt til en ekstern server som fremstiller disse dataene i et webgrensesnitt. I dette webgrensesnittet blir det tydelig i hvilken grad systemet dekker sitt eget energibehov, og ved videre testing vil dette kunne gi en indikasjon på hvor mange enheter som kan bli benyttet på systemet. Systemet slik det er bygget i dag er i stor grad skalerbart. Blir det utført endringer eller utvidelser av systemet må systemet tilpasses slik at endringene blir en del av systemet. Dette kan være tilpasninger som utvidelse av nye kretskort, sende signaler inn på de ledige ADC inngangene, stappe opp nye ADCer, legge til nye moduler i hovedprogrammet og oppdatere databasen både på lokal og ekstern server. Systemet har et godt grunnlag for utvidelse og alle delene av systemet vil fungere som en ”oppskrift” på eventuelle endringer.

I underkapittel 1.4.1 sier delmål 1 at det skal utvikles et styring- og overvåkningssystem til målestasjoner som er selvforsynt på energi. Dette målet er i stor grad oppnådd, da et fungerende system er ferdigstilt. Systemet som er utviklet er skreddersydd for de enheten som blir tatt i bruk, men stammen i systemet kan brukes for lignende systemer med andre enheter. I slike tilfeller må endringer utføres som det ble nevnt i diskusjonen om hovedmålet ovenfor.

Delmål 2 i underkapittel 1.4.2 sier at det endelige systemet vil kunne benyttes til å teste hvor stor kapasitet energiproduksjonssystemet har. Det blir lagt spesiell vekt på hvor mye energi konsumerende utstyr som systemet kan utvides med. Det er gode muligheter for dette i det endelige systemet som leveres. Derfor anses dette målet som oppnådd.

I dette prosjektet har det ikke blitt utført en stresstest for å finne ut kapasiteten til systemet. Det har vært et omfattende og tidskrevende prosjekt, og dette har medført til at en stresstest ikke har

latt seg gjennomføre i dette prosjektet. Det har derimot blitt observert at energiproduksjonen har dekket mer enn nok av energiforbruket til nå. En videreføring av prosjektet kan utføre en stresstest for å finne begrensningene til systemet.

## 6.2 Leveranser

I underkapittel 1.5 er det definert at gruppen skal levere et styring- og overvåkningssystem til et energisystem i henhold til oppdragbeskrivelsen. Systemet som blir levert er autonomt, og har mulighet for fjernstyring og overstyring. Det er også utviklet et system som leser av verdier fra sensorer, brenselcelle, charger controller og shunter etter tidsbestemte intervaller definert av oppdragsgiver. Disse dataene vil bli lagret i en lokal database, og sendes regelmessig til en database på den eksterne serveren. Webgrensesnittet baserer seg på databasen på den eksterne serveren. Her blir dataene presentert på en intuitiv og brukervennlig måte. Styring- og overvåkningssystemet kjører på operativsystemet Raspbian, som er en distribusjon av Debian Linux. Systemet benytter GPIO-pinnene til å ta kontrollen over flere enheter i systemet, og kan derfor skru de av eller på etter behov. Det er et innebygget alarmsystem som sjekker og kontrollerer om hendelser og målinger er korrekte. Alarmsystemet vil også varsle oppdragsgiver via e-post ved spesielle hendelser. På bakgrunn av disse argumentene ble de forventede leveransene levert i dette prosjektet.

## 6.3 Metode

Metoden beskrevet i underkapittel 1.6 er forsøkt fulgt i så stor grad det lot seg gjøre. Som det er forklart skal det kartlegges andre prosjekter og teknologiske løsninger som kan være relevante. Av de prosjektene som kunne ha noen relevans ble det sett nærmere på en bacheloroppgave utført ved Høgskolen i Østfold i 2014, og en sammenligning med denne beskrives i underkapittel 2.5.1.

Tidlig i prosjektet ble det laget en oversiktsmodell 3.1 for det komplette systemet med alle enheter som skal inngå samt koblinger og kommunikasjon. Denne modellen ble tatt utgangspunkt i ved bestillinger av de ulike enhetene. Monteringen ble utført i tett samarbeid med oppdragsgiver. Grunnen til dette er fordi ingen av gruppe medlemmene hadde riktig faglig bakgrunn til å gjøre oppkoblingen på en sikker og forsvarlig måte. Det bidro til at monteringen av systemet tok lenger tid enn forventet. På grunn av dette ble ikke prosessen for prosjektet iterativ hele veien slikt det var planlagt. I første fase ble det mer en waterfall-metode som ble fulgt. Ved usikkerhet på tekniske spesifikasjoner for de forskjellige enhetene ble de forskjellige leverandørene kontaktet hyppig.

Under testfasen ble det en mer iterativ prosess, hvor enheter og programkode systematisk ble testet som beskrevet i kapittel 5. Testingen har bidratt til å rette opp feil, og kartlegge noe av det som kunne blitt gjort bedre.

## 6.4 Oppgaven

Oppdraget er spesifisert på en slik måte at gruppen har stått veldig fritt i hvordan den endelige løsningen skal se ut. Krav og ønsker fra oppdragsgiver finnes i underkapittelet 1.3.1. Utover disse så har komponent- og enhetsvalg vært fritt, men i tett dialog med oppdragsgiver. Oppdraget i seg selv har vært uten grenser og det har vært opp til prosjektgruppen å sette begrensninger. Det har vært veldig varierte arbeidsoppgaver, alt fra å montere utstyr, arbeid med elektronikk, programmering og utvikling av web.

Det er gjort mye research for å finne kompatible og akseptable enheter. Ved bestilling har tidligere erfaringer gjort av oppdragsgiver og en tidligere bachelorgruppe har blitt tatt i betraktning. Utover dette er det benyttet ny og spennende teknologi i form av en brenselcelle. Dette var et ønske fra oppdragsgiver, men informasjonsinnhenting og bestilling er håndtert av gruppen.

Med gruppens begrensede erfaring innenfor sensorteknologi og fornybar energiproduksjon har det vært et utfordrende men spennende prosjekt. Det mest usikre har vært bruken av brenselcelle da verken gruppen eller oppdragsgiver hadde erfaring med denne type teknologi. Med god hjelp fra leverandør nevnt i seksjon 3.1.2 har prosessen med brenselcellen vært enklere enn forventet.

#### 6.4.1 Det positive

Under prosjektets gang har gruppen tilegnet seg mye ny kunnskap innenfor web, Python, Linux, servere og elektronikk. Til tross for begrensede forkunnskaper om dette ble det levert et velfungerende og solid system som gruppen kan si seg fornøyd med. Det har selvfølgelig forbedringspotensial, men med begrensning på tid og forkunnskaper er det ikke å forvente annet.

Andre positive ting er den erfaringen gruppen har fått med prosjektarbeidet. Det har blitt dannet et klarere bilde på hvor mye arbeid som kan utføres på drøye 500 timer for hver den enkelte person. I tillegg har det vært interessant å kunne jobbe med ny teknologi.

#### 6.4.2 Problemer underveis

Dette prosjektet krevde mye elektronikkunnskaper for å komme i mål. De fleste enhetene som skulle kobles på systemet trengte et egenutviklet kretskort. Dette krevde mye lesing og opparbeidelse av kunnskaper. En hendelse hvor jord ble koblet på inngangsspenningen førte til at kretskortet kortslo og flere baner ble brent av. En mulig årsak til dette kan ha vært stress i en periode med stort arbeidspress. Dette førte til en forbedring av kretskortet da hvert kort fikk sin egen inngangsspenning som var koblet direkte til batteripolene. Hele kortet ble kablet om med flertrådede kabler. Dette ble utført fordi det var flere ledninger som hadde brukket av i tilkoblingene.

I prosjektet er det to ulike typer charge controllere. Den til sol kan kommunisere over ethernet, og den for vind er en lavere modell og kan bare kommunisere over seriell. Problemet var at koden for ethernet kommunikasjonen måtte skrives om slik at den fungerte for seriell kommunikasjon. Det viste seg at «pymodbus» som var biblioteket som ble brukt for kommunikasjonen til charge controlleren for sol, også hadde funksjoner for seriell kommunikasjon. Det var bare noen justeringer i oppsettet, så skulle det fungere fint. Problemet var at dette biblioteket ikke hadde utviklet denne seriell kommunikasjonen for python 3 enda. Løsningen ble å bruke et annet bibliotek som heter "minimalmodbus", som fungerer med python 3.

Etter at kontakten med charge controlleren var opprettet oppstod det ett nytt problem. Dessverre viste det seg at charge controlleren til vindmølla ikke hadde mulighet til å skille mellom hvor mye energi som blir produsert og levert til batteriene, og hvor mye energi som blir brent av hvis batteriene er fulladet. Dette medfører at det ikke er mulig å lese av energiproduksjonen til vindmøllen direkte fra charge controlleren. En mulig måte å løse dette problemet på er å benytte en ny shuntmotstand til måling av strømmen som går mellom charge controlleren og batteribanken.

Et annet problem var at for å lage et webgrensesnitt etter oppdragsgivers ønske har det vært nødvendig å bruke HTML, PHP og javascript. Ingen av gruppemedlemmene har erfaring med webprogrammering og det har derfor vært mye nytt å sette seg inn i under utviklingen. På grunn av dette har prosessen vært tidkrevende og den endelige løsningen kunne blitt bedre.

### **6.4.3 Hva kunne ha blitt gjort annerledes?**

Prosjektet har hatt en grei flyt igjennom hele prosessen, men det er noen ting som kunne og kanskje burde ha blitt gjort annerledes. Informasjonsuthenting fra oppdragsgiver kunne ha blitt gjort mer effektivt slik at andre deler av prosjektet kunne ha startet tidligere. Dette gjelder spesielt informasjon om hvilke fysiske enheter som skulle måles. Dette kunne ha medført at det tekniske arbeidet kunne ha startet mye tidligere.

Det har vært stor tillit mellom gruppemedlemmene og ingen klare grenser har vært satt til hva som må bli gjort, utenom det som blir krevd fra Høgskolen. Dette har ført til at gruppemedlemmer har hatt ulike arbeidsrutiner og aktivitet. Prosjektarbeidet kunne vært mer strukturert slik at arbeidsmengden hadde vært jevnere spredt gjennom prosjekts gang.



# Kapittel 7

## Konklusjon

Prosjektets hovedformål var å gi et grunnlag for å undersøke hvor mye energikrevende utstyr en målestasjon drevet av eget energisystem kan inneholde. Delmålene var å utvikle et styring- og overvåkningssystem som kan benyttes med et eget energisystem. Det har blitt utviklet et overvåkning- og styringssystem som henter ut interessante data om både energiproduksjon og energiforbruk i målestasjonen. Systemet lagrer disse dataene og presenterer de i grafer og tabeller. Dette gir et godt grunnlag for å undersøke forholdet mellom forbruk og produksjon.

Skal det plasseres ytterligere utstyr i stasjonen må det likevel gjøres utvidelser i styring- og overvåkningssystemet. Det må gjøres nye tilkoblinger og programkoden må muligens utvides. Med grunnlag av de beskrivelsen som er gjort i rapporten burde det være forholdsvis greit for utenforstående personer med rett kompetanse å utføre dette arbeidet.

Det er utviklet mulighet for manuell overstyring av det autonome programmet. Det er også opprettet en konfigurasjonsfil for å sette opp hvilke verdier man ønsker at programmet skal operere etter. I tillegg er det utviklet et system som sender alarm til systemadministrator hvis systemet oppdager alvorlige feil. Systemet logger også hendelser, og disse loggene er tilgjengelig for systemadministrator via Internett. Dessverre viste det seg at charge controlleren til vindmølla ikke hadde mulighet til å skille mellom hvor mye energi som blir produsert og levert til batteriene, og hvor mye energi som blir brent av hvis batteriene er fulladet. Dermed er det i dag ikke mulig å måle energiproduksjonen til vindmøllen. Valg av charge controller var oppdragsgiver selv ansvarlig for, og denne ble ikke levert før prosjektet nærmet seg slutfasen. En mulig måte å løse dette problemet på er å benytte en ny shuntmotstand til måling av strømmen som går mellom charge controlleren og batteribanken. Det er ledige kanaler på begge ADCene som er koblet til RPi. Dessverre ble det ikke tid til dette i dette prosjektet.

Systemet som er bygd er best å anse som en grunnmur på hvordan et slik system kan bygges og driftes. Mye av arbeidet kan videreføres direkte til en ferdigstillelse av en autonom målestasjon som produserer sin egen energi og kommuniserer sine data til et webgrensesnitt. Hovedformålet om å bygge et system som gir et grunnlag for å undersøke hvor mye energikrevende utstyr en målestasjon drevet av eget energisystem kan inneholde, anses som nådd. Dette til tross for at det pr dags dato ikke er mulig å måle energiproduksjonen fra vindmølla. Delformålet om å utvikle et styrings- og overvåkningssystem som kan benyttes med eget energisystem anses som nådd.

### 7.1 Videreføring

Systemet har bare blitt testet i løpet av våren 2016, og det er vanskelig å konkludere i hvor stor grad systemet klarer å dekke sitt eget energibehov før det har vært igjennom en vintersesong.

Systemet burde også være igjennom en stresstest slik det blir omtalt i testing kapittelet 5. Dette kan bli videreført til et selvstendig prosjekt hvor begrensingene for systemet blir utforsket.

Webløsningen bærer preg av at gruppen har lite erfaring med webprogrammering, men databasen som ligger i bunnen av løsningen er solid. Denne databasen kunne ha blitt tatt med videre til et prosjekt for webutviklere som kunne bygd et mer funksjonelt webgrensesnitt.

Det kan tenkes at oppdragsgiver ønsker en enklere måte å overstyre systemet på. Hovedprogrammet er laget slik at det benytter en database for å hente ut grenseverdiene som det benytter i valgene sine. For å endre disse innstillingene er det bare en databasetilkobling til databasen på RPi som skal til. En elegant løsning på dette hadde vært å lage en applikasjon for mobile enheter.

# Bibliografi

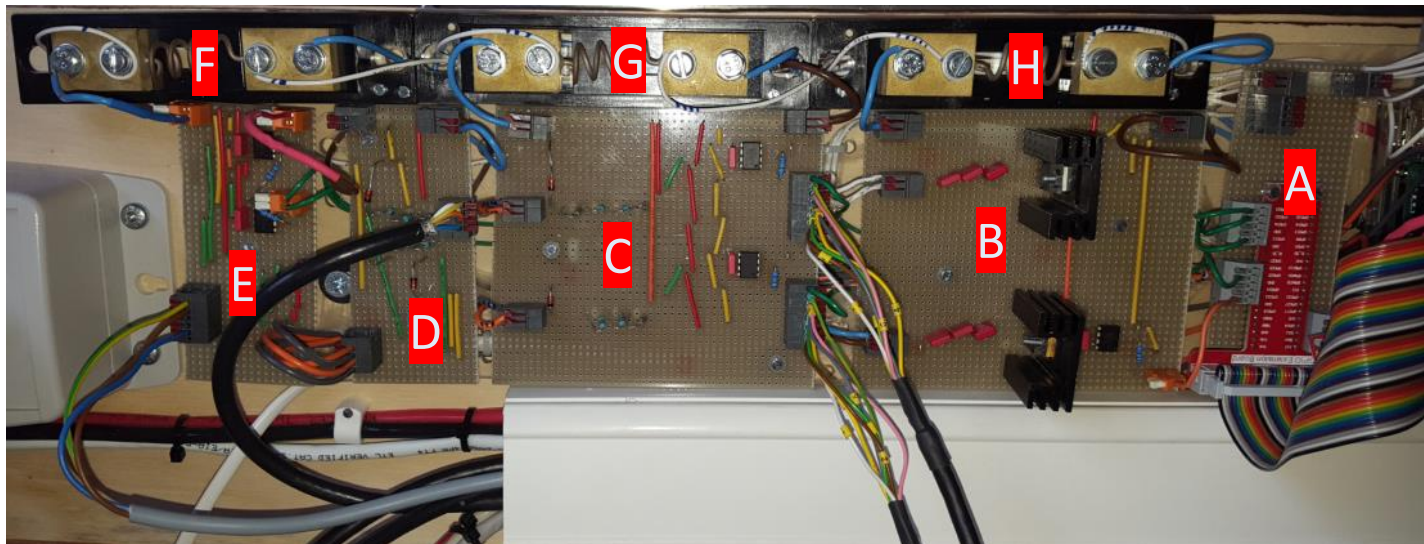
- [Ackermann i inni, 2005] Ackermann, T. i inni (2005). *Wind power in power systems*, volume 140. Hentet 2016-02-23, fra, publisher=Wiley Online Library.
- [Baraka i inni, 2013] Baraka, K., Ghobril, M., Malek, S., Kanj, R., i Kayssi, A. (2013). Low cost arduino/android-based energy-efficient home automation system with smart task scheduling. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*, pages 296–301. IEEE.
- [Grimenes, 2010] Grimenes, Arne Auen og Jerstad, P. o. S. B. (2010). *Rom Stoff Tid*, volume 3. Cappelen Damm AS. Hentet 2016-03-5, fra,.
- [Jens Petter Grimstad Andersen, 2015] Jens Petter Grimstad Andersen, Michael Glomnes, H. J. o. E. A. K. (2015). Styring av solcellesystem og implementering i undervisning. Hentet 2016-05-02, fra, <http://frigg.hiof.no/bo15-g28>.
- [Mikael Johansen Grimstad, 2014] Mikael Johansen Grimstad, Kristoffer Jensen, K. N. K. o. M. L. (2014). Prosjekt hessdalen - værstasjoner. Hentet 2016-02-22, fra, <http://frigg.hiof.no/bo14-g15>.
- [P.Strand, 2015] P.Strand, E. (2015). Project hessdalen. Hentet 2016-01-17, fra, [http://www.hessdalen.org/index\\_n.shtml](http://www.hessdalen.org/index_n.shtml).
- [Roppestad, 2015] Roppestad, R. (2015). *Del 2. Styring og regulering*, volume 1. Hentet 2016-04-03, fra, publisher=[http://www.it.hiof.no/rr/itd30005/pdf\\_dok/f\\_notater\\_2015/](http://www.it.hiof.no/rr/itd30005/pdf_dok/f_notater_2015/).
- [Thomas L. Floyd, 2014] Thomas L. Floyd, D. L. B. (2014). *Electronic Fundamentals - Circuits, Devices and Applications*, volume 1060. Hentet 2016-03-02, fra, publisher=Pearson.



## **Tillegg A**

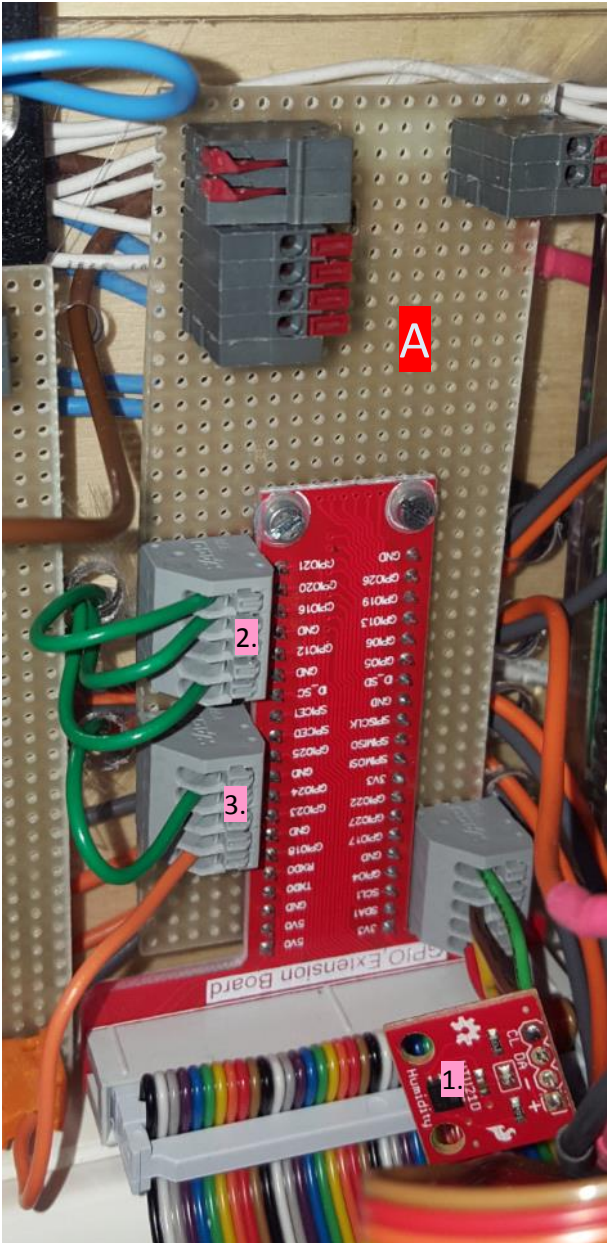
# Oppkobling elektronikk - Målestasjon

Kortene er skrudd fast på en polykarbonatplate. Platen er skrudd fast på innsiden av kassen til brenselcella. Dermed befinner løsningen seg i en isolert kasse. Varmen som avgis elektronikken, spesielt spenningsregulatoren, er med å bidra til at temperaturen i kassa ikke faller under de grensene som brenselcella krever. Mellom polykarbonatplaten og veggen er det bygd opp med gummiskiver. Dette gjorde det mulig trekke ledningene mellom plata og veggen i kassa.



A	GPIO-buss med sensor for innvendig temperatur og fuktighet
B	Raspberry Pi og router. (Strømforsyning og av/på krets for router)
C	Vindretning og vindhastighet. Strømforsyning, av/på krets for varmeelementer og spenningsdelere og målesignaler.
D	Temperatur og fuktighet ute. Strømforsyning, spenningsdelere og målesignaler.'
E	Kamerasystem. Av/på-krets for varmeelement i kamerahus og av/på krets for kamera, vifte og streamingboks.
F	Shuntmåler for kamerasystemet.
G	Shuntmotstand for vindsensorer
H	Shuntmotsand for rasperry pi og router

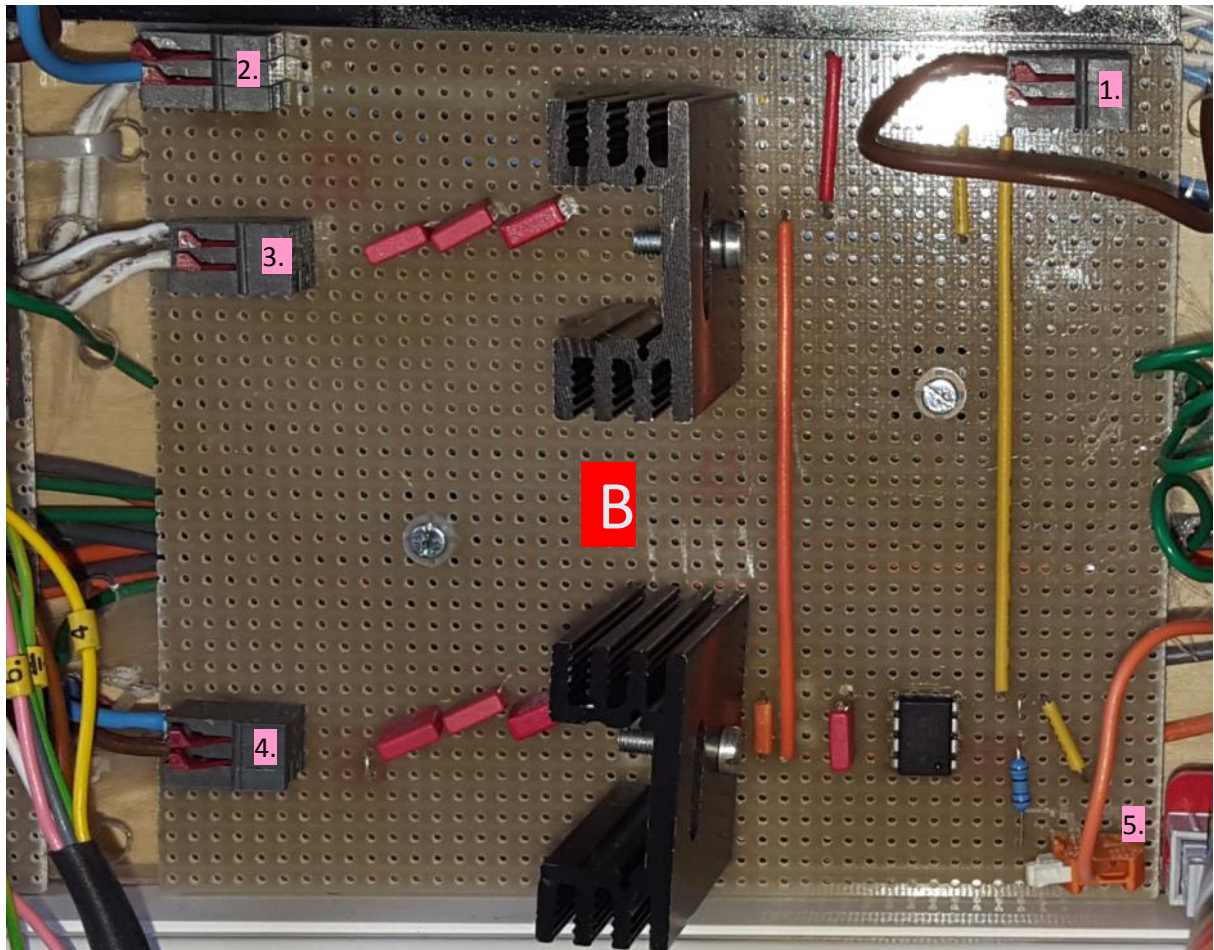
## A - Raspberry Pi buskort og sensor inne



1. Sensor varme og fuktighet inne  
Fra topp, kun de som er koblet til:
  - Jord
  - CL
  - DA
  - 3,3 volt positiv leder
2. GPIO  
Fra topp, kun de som er koblet til:
  - Kamera med vifte og streaming (pinne 40, GPIO 21)
  - Varmeelement kamerahus (pinne 38, GPIO 20)
  - Vindretning varmeelement (pinne 32, GPIO 12)
3. Vindhastighet varmeelement (pinne 22, GPIO 25)
  - Router (pinne 16, GPIO23)



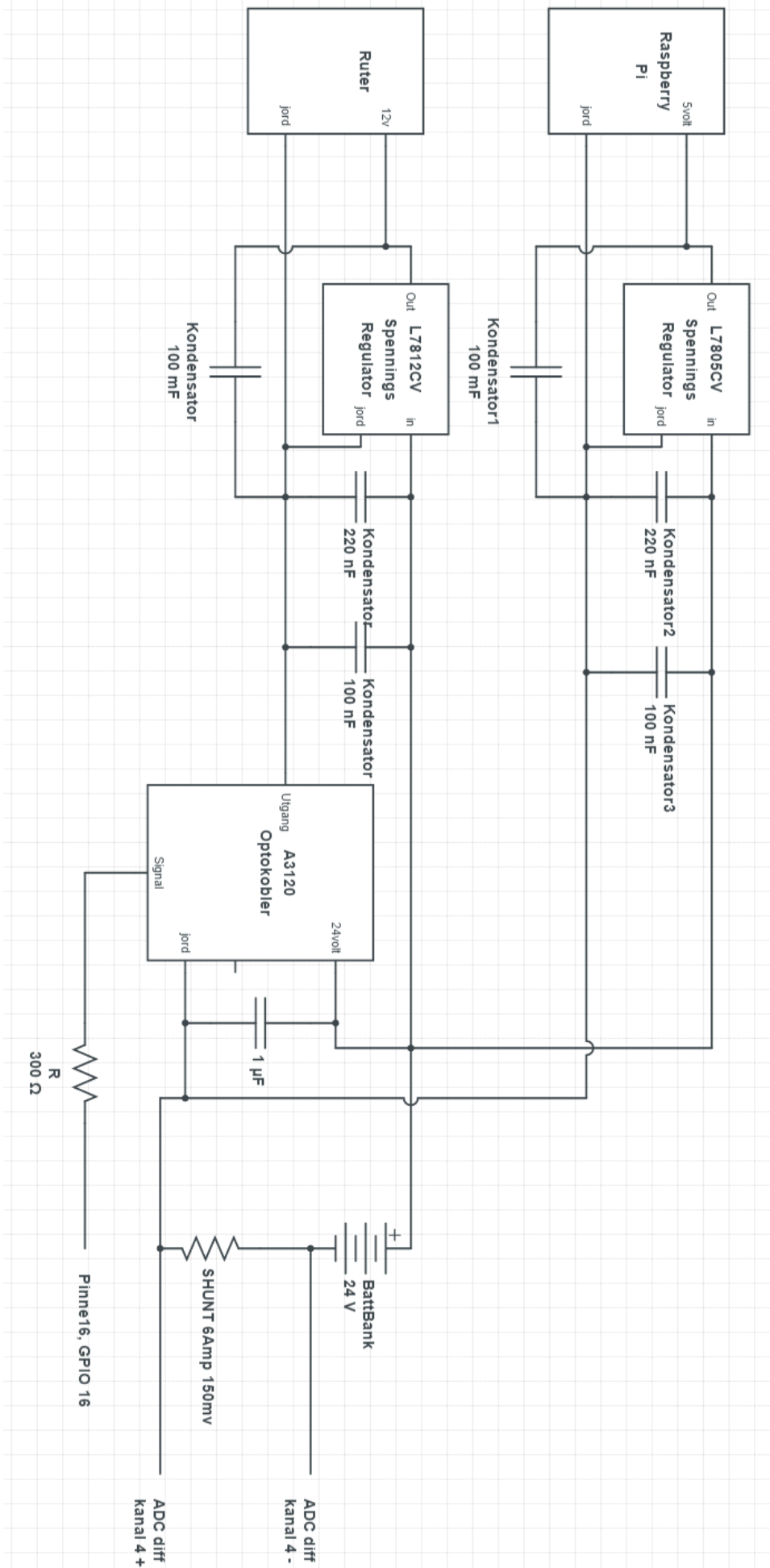
## B - Raspberry Pi og Router Bilde



1. 24 volt fra batteri kobles til i øverste port.
2. Jord kobles til i nederste port - går videre gjennom tilhørende shunt.
3. Rasperry Pi strømforsyning. Jord kobles øverst, positiv leder nederst.
4. Router strømforsyning. Jord kobles øverst, positiv leder nederst.
5. GPIO-signal som slår av/på router

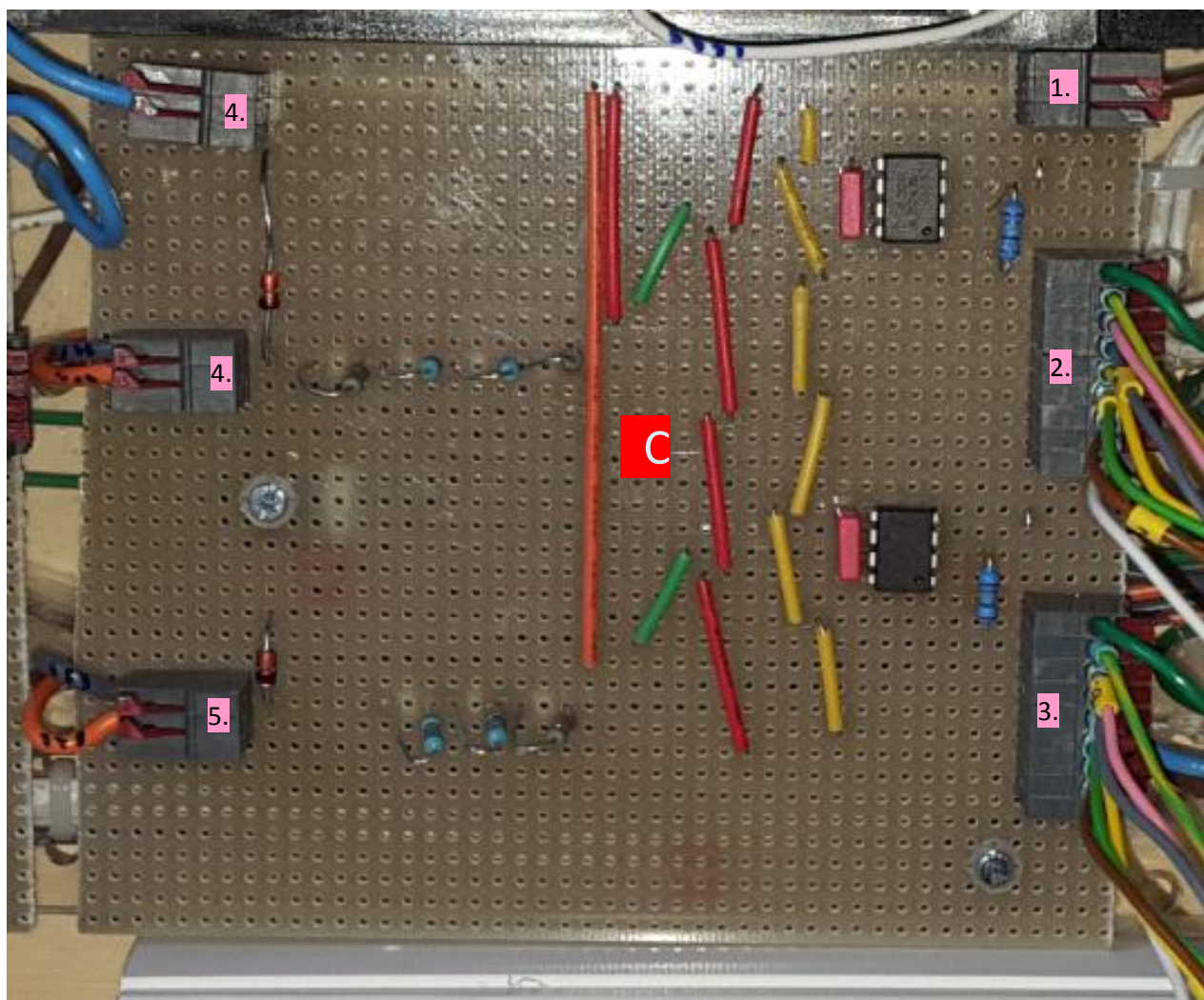


## B - Raspberry pi og router kobling



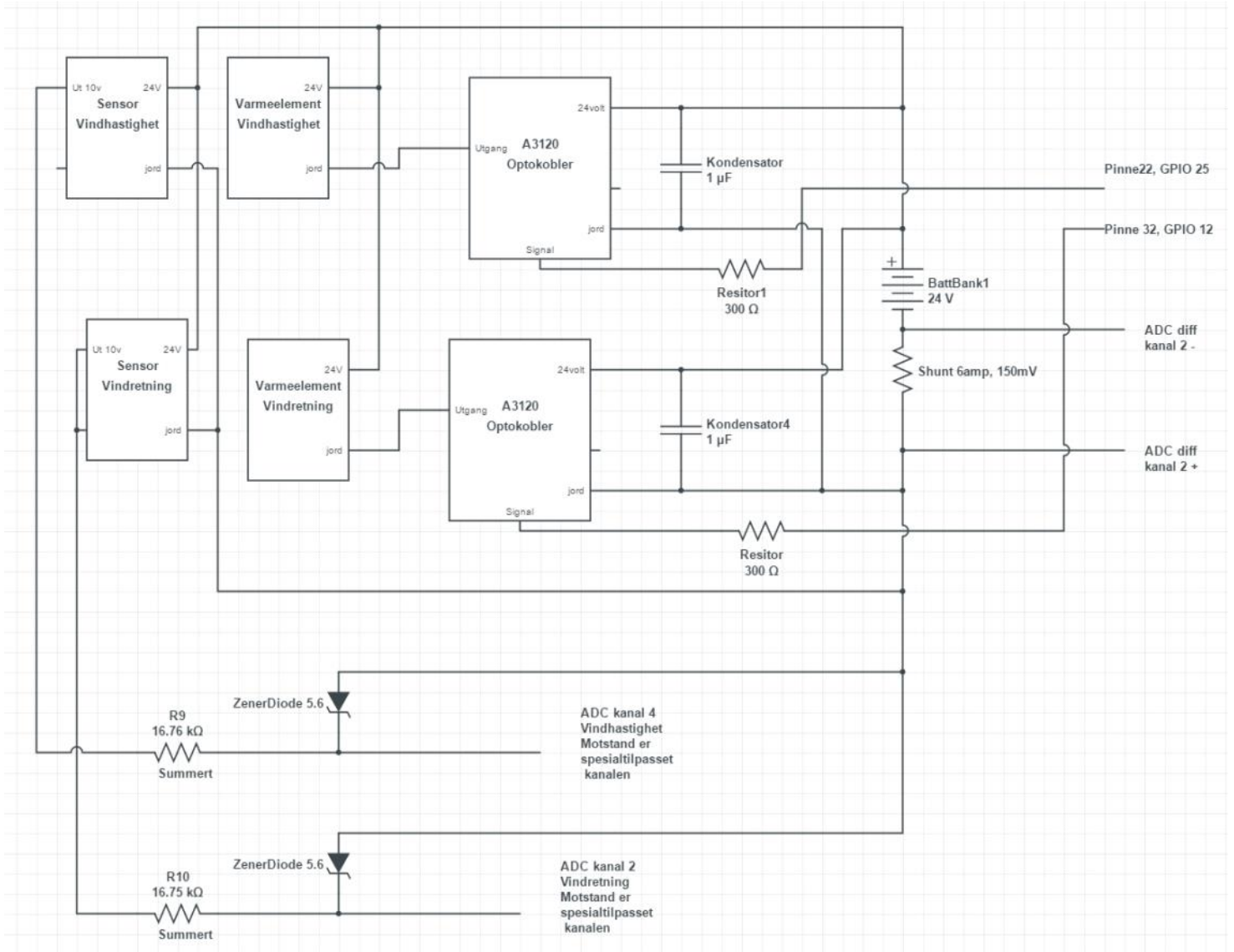
# C - Vindsensorer bilde

30. april 2016 14:09



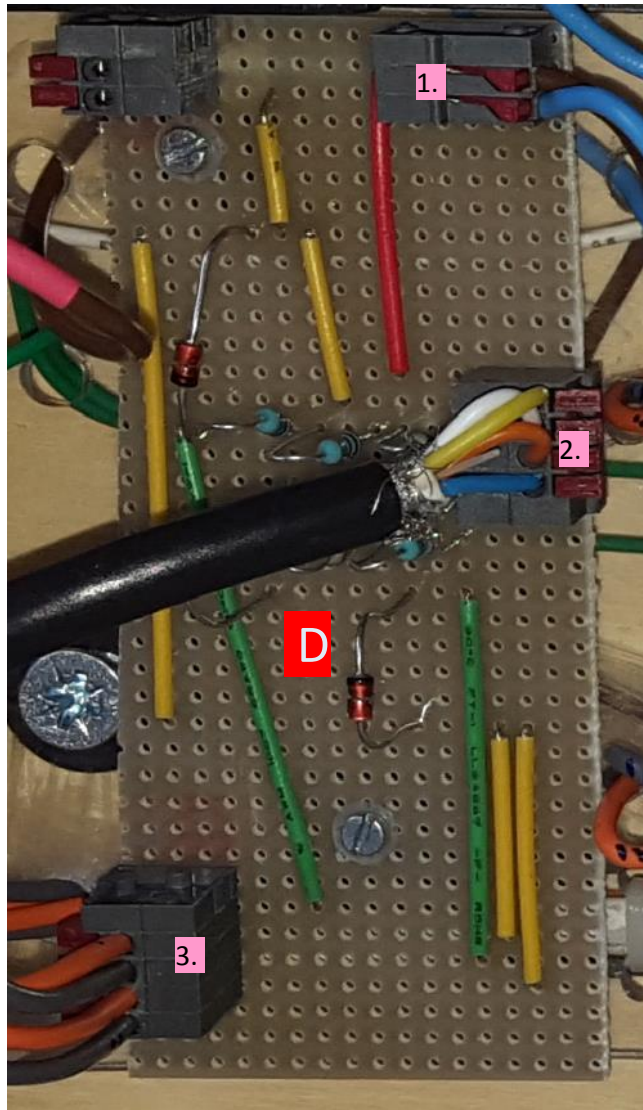
1. Strømforsyning til kortet fra batteri, kobles i **øverste port**
2. Tilkobling **vindretningsensor**  
Fra øverst:
  - GPIO-signal som slår av/på optokobler til varmeelement
  - Jordingsledning, sensor
  - negativ leder varmeelement
  - positiv leder varmeelement
  - negativ leder målesignal
  - positiv leder målesignal
  - negativ leder driftspenning sensor
  - positiv leder driftspenning sensor
3. Tilkobling **Vindhastighet**  
Fra øverst:
  - GPIO-signal som slår av/på optokobler til varmeelement
  - Jordingsledning, sensor
  - negativ leder varmeelement
  - positiv leder varmeelement
  - negativ leder målesignal
  - positiv leder målesignal
  - negativ leder driftspenning sensor
  - positiv leder driftspenning sensor
4. Jordledning som går til tilhørende shunt, **nederste port**
5. Målesignal vindretning til ADC positivt signal nederst
6. Målesignal vindhastighet til ADC positivt signal neders

# C - Vindsensorer koblingskjema



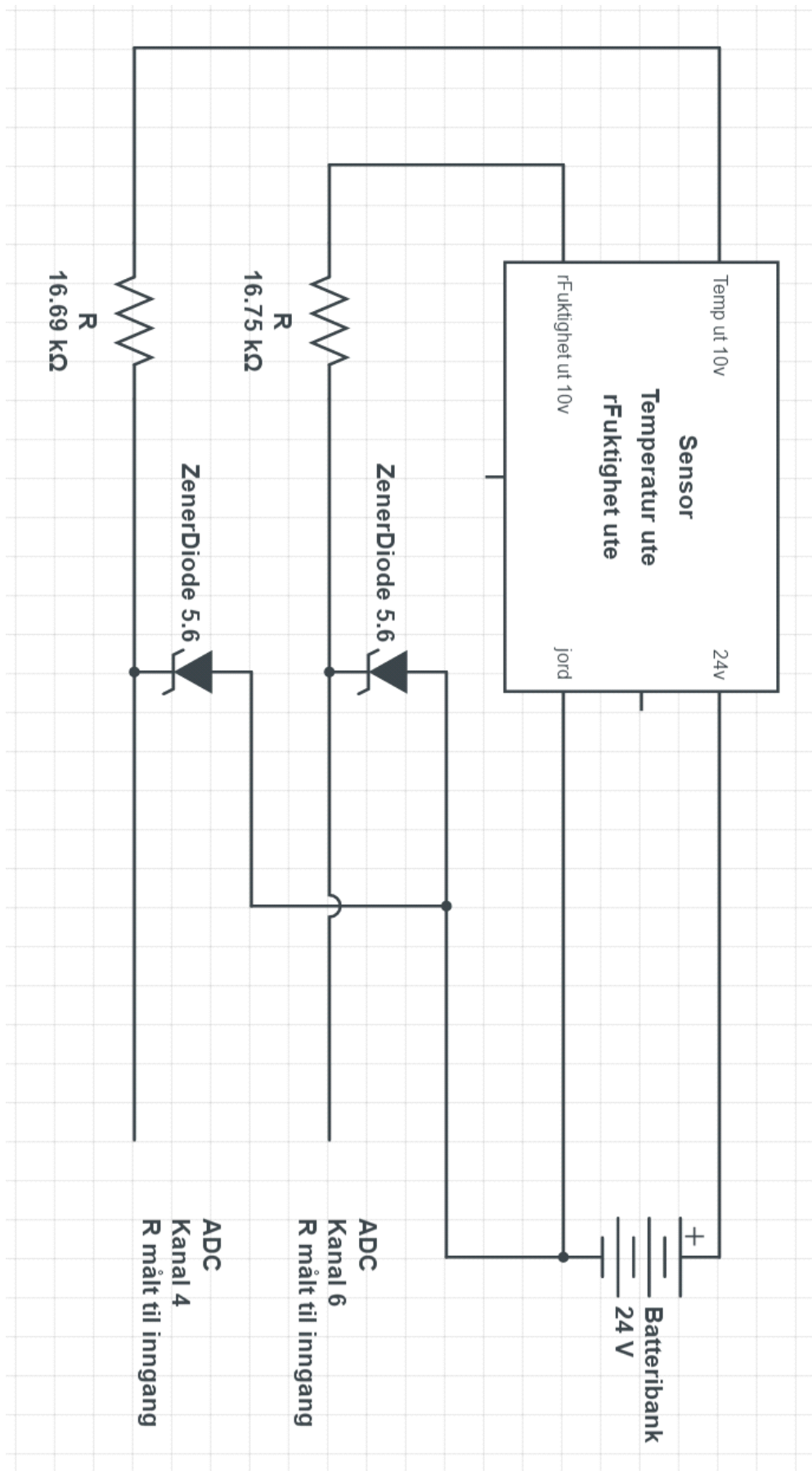
# D - Temperatur bilde

30. april 2016 14:30



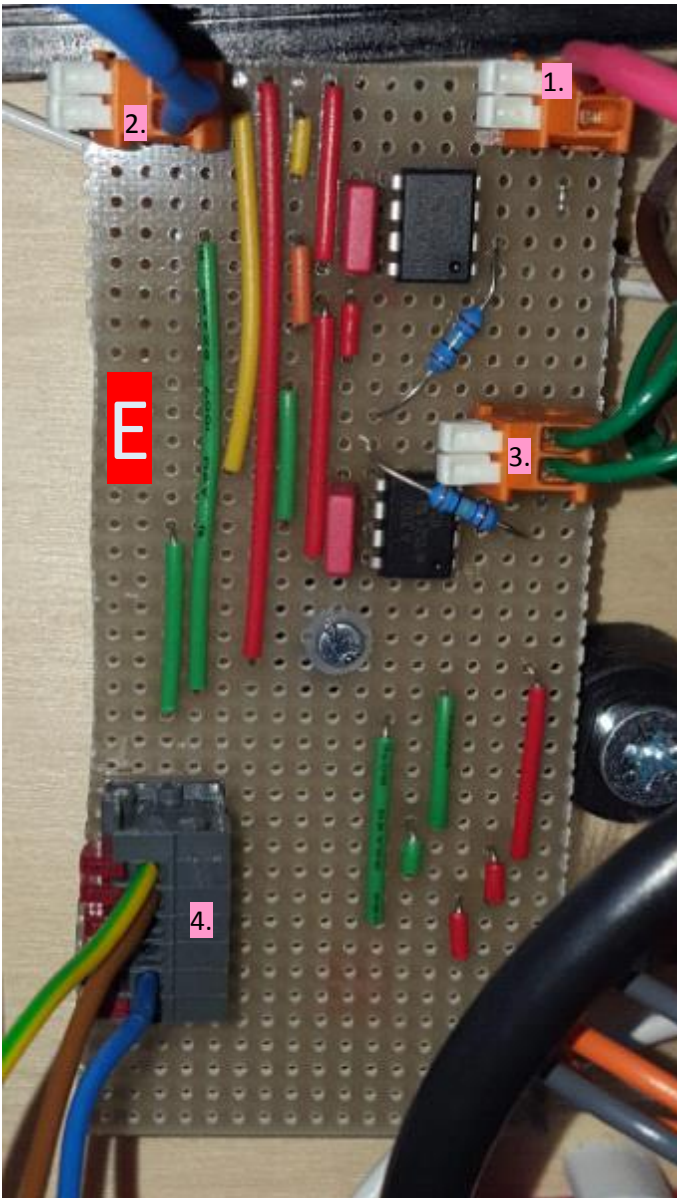
1. Strømforsyning fra batteri til kortet. Her skal både positiv og negativ leder kobles til. **Positiv leder øverst.**
2. Tilkobling utvendig temperatur og fuktighetsensor.  
Tilkobling fra toppen:
  - Driftspenning positiv leder sensor
  - Driftspenning Jord sensor
  - Relativ luftfuktighet ute
  - Temperatur ute
3. Målesignaler til ADC, temperatur og fuktighet.  
Fra topp:
  - Positiv målesignal relativ fuktighet
  - jord målesignal relativ fuktighet
  - Positiv målesignal temperatur ute
  - jord målesignal temperatur ute

# D - Temperatur koblingskjema



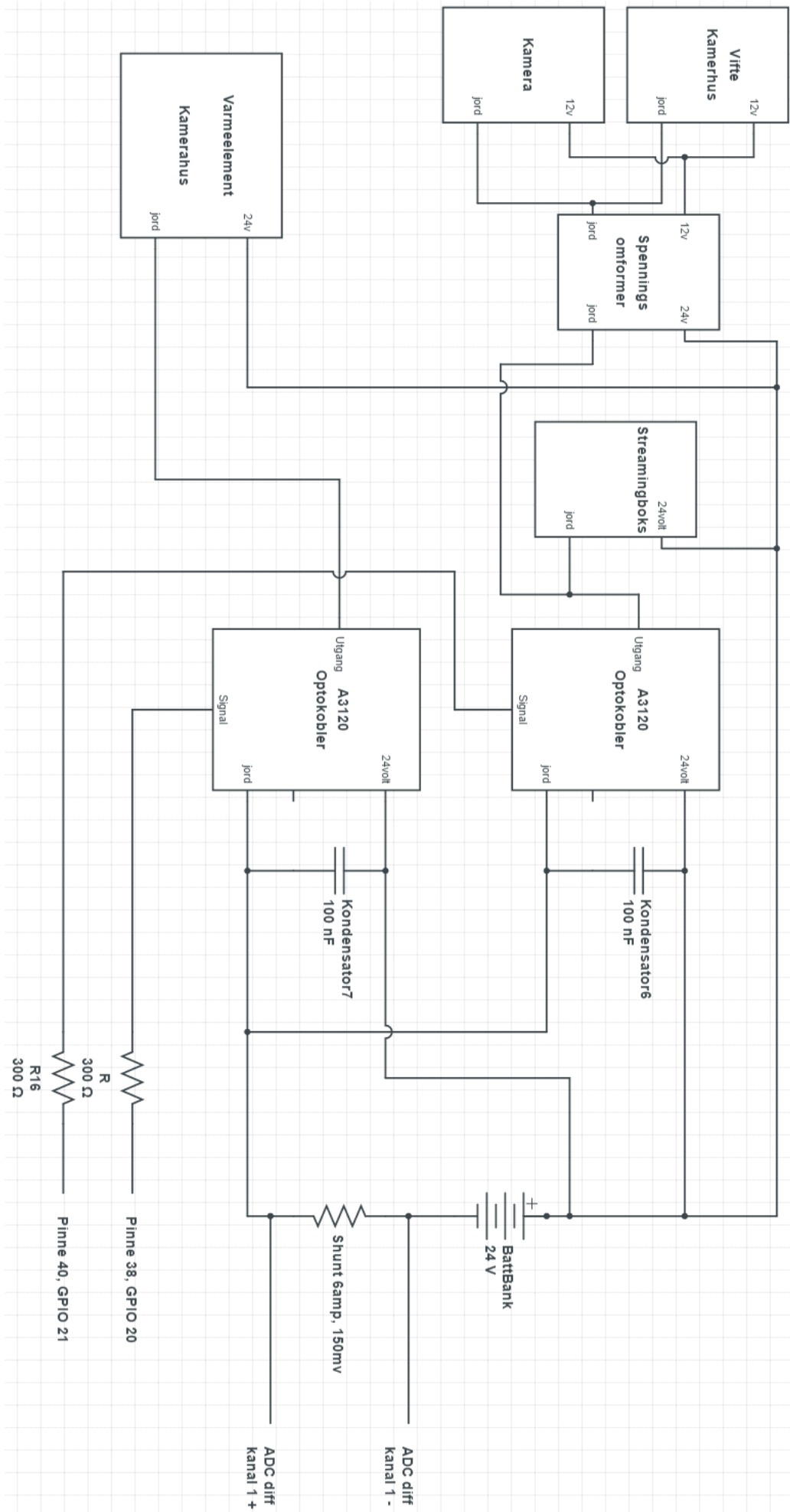


## E - Kamerasystem bilde



1. Driftspenning fra batteri i øverste port
2. Jord, kobles til tilhørende shunt i nederste port
3. GPIO signaler for av/på-logikk
  - Øverste GPIO til optokobler tilhørende kamera, vifte og streamingboks
  - Nederste GPIO til optokobler tilhørende kamerahusets varmeelement
4. Kamerasystem
  - Fra topp:
  - Jord til kamera i kamerahuset (fra optokobler)
  - Driftspenning til kamerahus
  - Jord til streamingboks (ikke tilkoblet på bilde - fra optokobler)
  - Driftspenning streamingboks (ikke tilkoblet på bilde)
  - Jord til varmeelement i kamerahus (fra egen optokobler)
  - Ledig port for 24 volts driftspenning.

# E - Kameran system koblingskjema



# F, G, H

30. april 2016 14:48

Shuntene er koblet som siste ledd i hver sin krets før minuspoleen på batteriet. Disse kunne stått rett etter den positive polen. I utgangspunktet brukte vi en ikke-differensiell ADC til måling av shunt-verdier. Ved bruk av en ikke-differensiell ADC må shuntene stå sist i kretsen. Ved bytte til en differensiell ADC ble derfor shuntene stående slik de opprinnelig gjorde i kretsen.

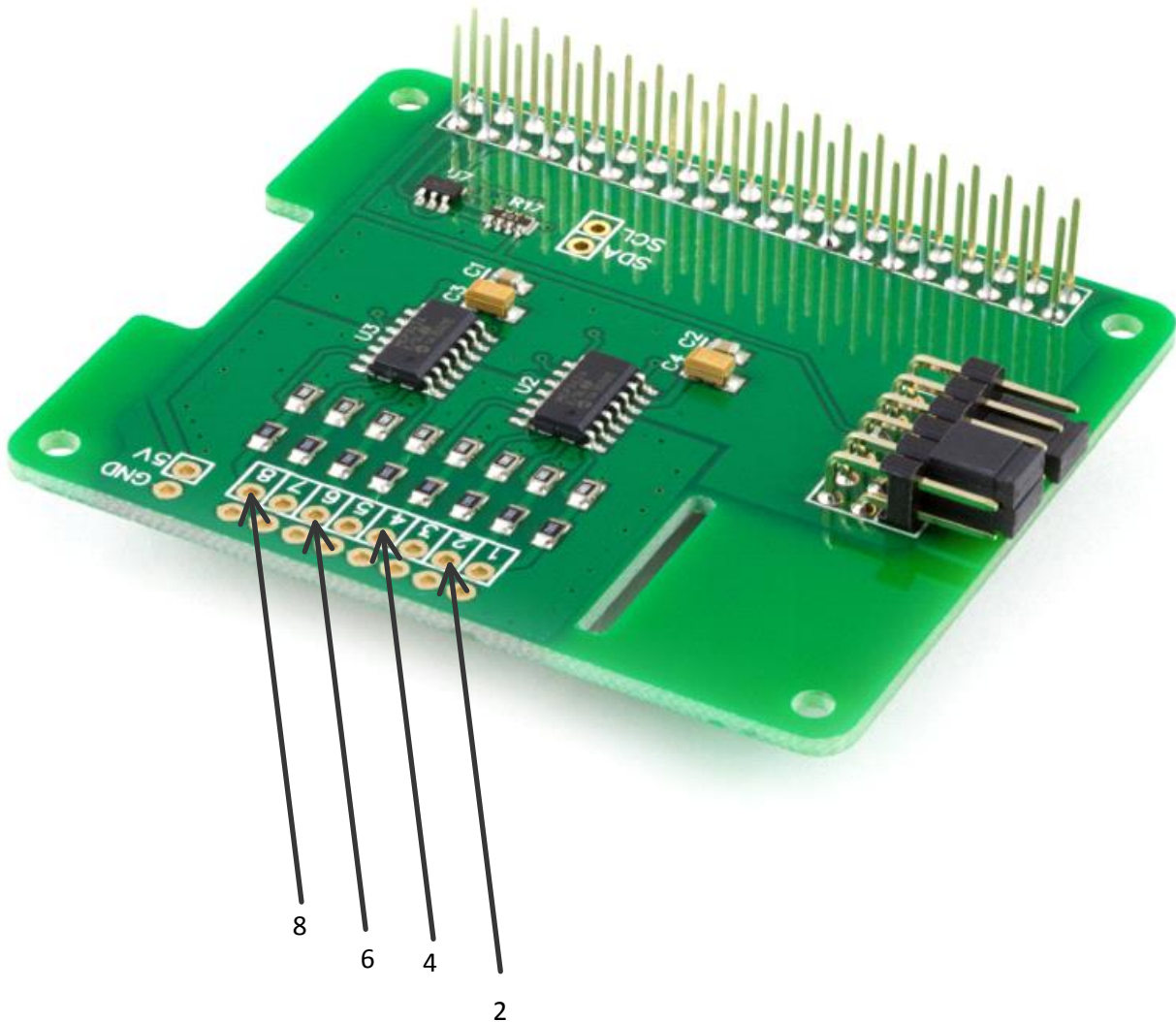


1. Jordledning fra kort E (kammersystem)
2. Til kanal 1+ ADC differensiell
3. Til kanal 1- ADC differensiell
4. Jordledning til batteri
5. Jordledning fra kort C (Vindsensorer)
6. kanal 2+ ADC differensiell
7. kanal 2- ADC differensiell
8. Jordledning til batteri
9. Jordledning fra kort B (raspberry og router)
10. Til kanal 4+ ADC differensiell
11. Til kanal 4- ADC differensiell
12. jordledning til batteri



# ADC PI Plus

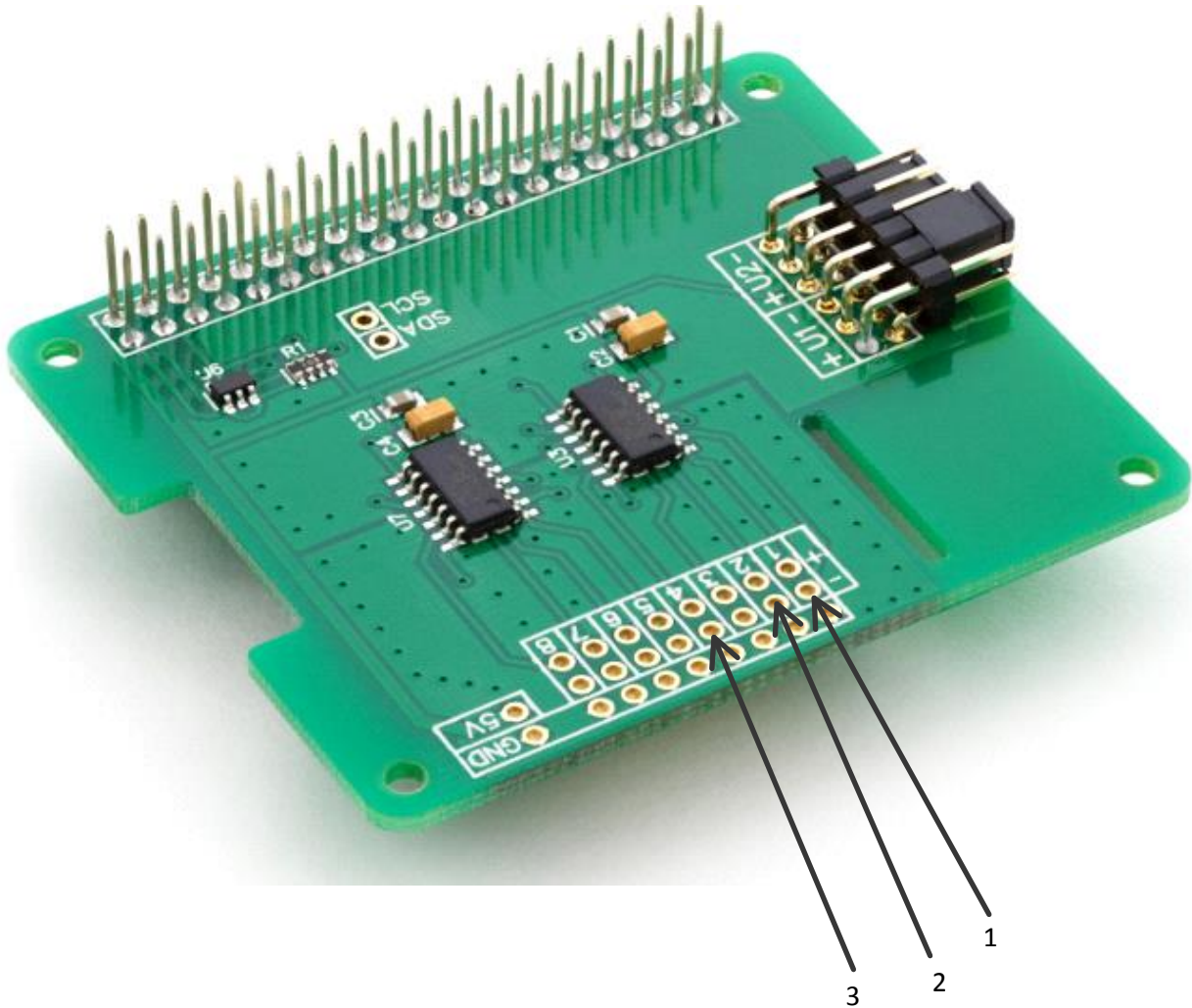
ADC-pi plus er stacket øverst i på raspberry pi en.



- Kanal2: Målesignal Vindretning
- Kanal4: Målesignal vindhastighet
- Kanal 6: Målesignalk Relativ fuktighet ute
- Kanal 8: Målesignal Temperatur ute

# ADC Pi differential

ADC-differential er stacket mellom ADC pi plus og raspberry pi.  
Tilkoblingene er loddet fast.



- Kanal 1: Shunt fra kamerasystem. Ledning før shunt til +, og ledning etter shunt til -
- Kanal 2: Shunt fra vindsensorene. Ledning før shunt til +, og ledning etter shunt til -
- Kanal 4: Shunt fra rasperry pi og router. Ledning før shunt til +, og ledning etter shunt til -



