



Høgskulen
på Vestlandet

BACHELOROPPGAVE:
BO21E-24 SENSORNETTVERK FOR
INNSAMLING AV NEDBØRSDATA

<Espen Leiknes Larsen>
<Jostein Ring>

31. mai. 2021

Dokumentkontroll

<i>Rapportens tittel:</i> BO21E-24 Sensornettverk for innsamling av nedbørsdata	<i>Dato/Versjon</i> 31. mai. 2021/1.0
	<i>Rapportnummer:</i> B021E-24
<i>Forfatter(e):</i> Espen Leiknes Larsen Jostein Ring	<i>Studieretning:</i> KOM18
	<i>Antall sider m/vedlegg</i> 29
<i>Høgskolens veileder:</i> Anne-Lena Kampen, anne-lena.kampen@hvl.no	<i>Gradering:</i> Åpen
<i>Eventuelle Merknader:</i> Vi tillater at oppgaven kan publiseres.	

<i>Oppdragsgiver:</i> HVL/Småkraft AS	<i>Oppdragsgivers referanse:</i>
<i>Oppdragsgivers kontaktperson(er) (inkludert kontaktinformasjon):</i> Trond Tollefsen, trond.tollefsen@smaakraft.no	

Revisjon	Dato	Status	Utført av
0.1	19.01.2021	Opprettet dokument fra mal	Jostein Ring
0.2	05.02.2021	Omgjort dokument fra forstudie til bachelor	Jostein Ring
0.3	18.05.2021	Dokumentet er sendt inn til veileder for vurdering	Jostein Ring
0.4	24.05.2021	Dokumentet er revidert basert på veileders tilbakemelding.	Jostein Ring
1.0	31.05.2021	Dokumentet ferdigstilt for innlevering	Espen Larsen og Jostein Ring

Forord

Denne bachelorrapporten er skrevet av Jostein Ring og Espen Leiknes Larsen, studenter under linjen Kommunikasjonsingeniør / Cyberingeniør ved Høgskulen på Vestlandet. Rapporten dokumenterer vårt arbeid og bakgrunnen for oppgaven BO21E-24: Sensornettverk for innsamling av nedbørsdata.

Vi ønsker å takke Høgskulen på Vestlandet for en flott oppgave og vår interne veileder Anne-Lena Kampen for gode tilbakemeldinger og solid veiledning gjennom hele prosjektet.

Vi vil også å takke Trond Tollefsen fra Småkraft som har trådd inn som ekstern veileder da vi var foruten. Han har gitt oss et innblikk inn i verdenen av kraftverk og har hjulpet oss inn på en spennende og lærerik oppgave.

Sammendrag

Bacheloroppgaven vår gikk ut på å lage et sensornettverk for å kunne bedre estimere mengden energi et vannkraftverk kan produsere basert på værdata. Differansen mellom estimert energi solgt på kraftbørsen og faktisk energi produsert må kompenseres i form av gebyrer.

Sensornettverket består av to nedbørmålere, som mellomlagrer dataene til et SD-kort og sender disse over WiFi til en mottaker. Mottakerenheten overfører deretter dataene over internett til Småkrafts database for å sammenflettes med andre data målt. Nedbørmålerne plasseres i nedbørsfelt til småkraftverk som da krever at strømforsyning er levert av et batteri og at de er tilstrekkelig innkapslet for å tåle norsk natur.

Gjennom rapporten opplyses det om hvilke komponenter vi har vurdert som best til å måle nedbør, samt komponenter til å måle andre verdier som temperatur og vannføring. Det blir diskutert en del rundt kompatible og ikke kompatible komponenter når det gjelder å bygge et sensornettverk med Arduino-enheter. Det blir vist hvor stor forskjell det kan bli på et sensorsystem ut ifra hvilke hovedkomponenter som systemet bygges på. Grunnet behovet for batteri er det også lagt vekt på at nedbørsensorene skal være strømsparsomme.

Testing av systemet har gitt oss gode resultater og etterkommer de fleste av kravene vi har satt. Den største faktoren som vi mener må sees på videre og forbedres er lading av batteri og strømforbruk. Systemet vil teoretisk kreve en batterikapasitet på 33Ah for å ha en batterilevetid på én måned, mest sannsynlig mer i praksis, og vi mener den mest effektive måten å redusere dette på er å lade batteriet med solcellepanel.

Innhold

Dokumentkontroll	2
Forord	3
Sammendrag	3
1 Innledning	6
1.1 Oppdragsgiver	6
1.2 Problemstilling	6
1.3 Hovedidé for løsningsforslag	6
2 Kravspesifikasjon	7
3 Analyse av problemet	8
3.1 Utforming av mulige løsninger	8
3.1.1 Løsningsalternativ 1	8
3.1.2 Løsningsalternativ 2	8
3.1.3 Vurderinger i forhold til verktøy og Hardware/Software komponenter	8
3.2 Konklusjon	10
4 Realisering av valgt løsning	11
4.1 Senderenhet	11
4.1.1 Design	11
4.1.2 Kode	13
4.2 Mottakerenhet	14
4.2.1 Design	14
4.2.2 Kode	15
4.3 Influx	16
4.3.1 Oppsett	16
4.3.2 Krav	16
4.3.3 Realisering	16
4.3.4 Konklusjon	16
5 Testing	17
5.1 Kalibrering av nedbørmåler	17
5.1.1 Bakgrunn	17
5.1.2 Konklusjon	18
5.2 Strømforbruk	19
5.2.1 Bakgrunn	19
5.2.2 Konklusjon	20

5.3	Kommunikasjonsrekkevidde	21
5.3.1	Bakgrunn.....	21
5.3.2	Konklusjon	21
6	Diskusjon	22
6.1	Tidsbruk.....	22
6.2	Komponenter	22
6.2.1	Forstudiet	22
6.2.2	Midtveis.....	23
6.2.3	Ettertanke.....	23
6.3	Veien videre.....	24
6.3.1	Utbedre kode.....	24
6.3.2	Komponenter.....	24
7	Konklusjon.....	25
	Referanser	26
Appendiks A	Figurliste	27
Appendiks B	Tabelliste	27
Appendiks C	Forkortelser og ordforklaringer.....	27
Appendiks D	Prosjektledelse og styring.....	28
D.1	Prosjektorganisasjon	28
D.2	Fremdriftsplan	28
D.3	Risikoliste.....	29

1 Innledning

1.1 Oppdragsgiver

Småkraft AS er et selskap som produserer strøm ved hjelp av småkraftverk. Siden etableringen i 2002 har de bygget over 100 småkraftverk spredt rundt i hele Norge. Disse kraftverkene styres via nettet fra hovedkontoret i Bergen og fra distriktskontorene i Flatanger, Harstad, Sandnes og Oslo. Småkraft har rundt 20 ansatte og opererer med over 100 driftsoperatører. De produserer ca. 1450 GWh fornybar energi årlig som dekker ca. 90.000 husholdninger. Målet til Småkraft er å utnytte små elver og bekker til produksjon av fornybar energi. Grunneiere leier ut elver og bekker til Småkraft mens lokale driftsarbeidere og entreprenører bygger ut og vedlikeholder småkraftverkene. [1]

1.2 Problemstilling

Småkraft forutser en større endringstakt på kraftbørsen i forbindelse med at flere mindre lokale energiprodusenter kommer inn på markedet. Salg av energi på kraftbørsen krever at energitilbydere leverer rett energimengde til rett tid, noe som er utfordrende for fornybare energikilder som sol, vind og vannkraft uten vannreservoarer. Differansen mellom estimert og produsert energi må kompenseres i form av gebyrer. Småkraft ønsker å finne en løsning for vannkraft for å bedre forutse hvor mye energi kraftverket kan produsere til hvilke tider. Så langt vi vet finnes det per i dag ingen løsning for denne problemstillingen. Såfremt plassering av eventuelt utstyr for å løse problemstillingen havner innenfor grunneiers eiendom så anser vi at det ikke er noen utenforliggende faktorer som vil kunne ekskludere eventuelle løsningsforslag.

1.3 Hovedidé for løsningsforslag

Hovedmålet med oppgaven er å estimere/beregne hvor mye energi kraftverket kan produsere innenfor et bestemt tidsintervall. Dette styres av vannføringen (volumet vann som beveger seg) i elven en produserer energi fra innenfor tidsintervallet. En kan generelt forutse endringer i vannføring basert på mengden nedbør i nedslagsfeltet til elven. Småkrafts løsningsforslag er å måle nedbørsmengde i nedslagsfeltet og måle vannføring i tilførselselver og bekker, og forutse kraftproduksjonen basert på disse dataene.

2 Kravspesifikasjon

Det endelige systemet kan beskrives som en sort boks som mottar data for deretter å sende dataene videre til en server hvor dataene behandles. Informasjonen skal distribueres ofte, historikken skal lagres, og det skal varsles når den sorte boksen ikke fungerer som den skal. Boksen skal kunne fungere over lengre tid uten vedlikehold, og den skal være innkapslet for å tåle norsk vær.

Kravspesifikasjon:

Løsningen skal:

- oppdatere informasjon senest hvert 5. minutt ved endringer.
- lagre historikk av input data.
- ha lagringskapasitet på mikrokontrolleren
- Sende "keep-alive" meldinger hver 30 min
- sende informasjon til en database.
- ha ekstern strømforsyning med batteri.
- ha en strømsparingsmodus på mikrokontrolleren (for å redusere størrelse på batteri)
- ha *IP-grad* for å tåle norsk natur og vær.
- tåle temperaturer ned til -20°C

Løsningen skal ikke:

- påvirke vannføringen i elven
- negativt påvirke naturen

I tillegg til kravene som omfatter systemet er det flere mål som det er ønskelig å oppnå, men som ikke er kritiske for at et løsningsalternativ skal være gjennomførbart:

- Løsningen bør ikke kreve vedlikehold oftere enn 6 måneder.
- Løsningen skal implementere en metode for å lade batteri (for å redusere størrelsen på batteri).

3 Analyse av problemet

Problemstillingen baserer seg på hypotesen om at en kan forutsi energiproduksjonen til et kraftverk ved å måle mengden nedbør i nedslagsfeltet og vannføring i tilførsels- elver og bekker. Det finnes flere andre potensielle faktorer som kan påvirke energiproduksjon, f.eks. temperatur, fuktighet i jordsmonn og terreng rundt. For å begrense omfanget av oppgaven vil utarbeidet løsningsforslag i utgangspunktet kun basere seg på bruk av nedbør-, temperatur- og vannførings-måler. Løsningsalternativene skal bidra til å øke Småkrafts datagrunnlag for å forutse energiproduksjon.

Den sorte boksen beskrevet i kravspesifikasjon vil måtte bestå av en eller flere sensorer og en mikrokontroller som mottar og behandler data fra sensorene. I tillegg skal dataene sendes til en mottaker.

For å beskytte utstyret, og for å ikke påvirke naturen rundt negativt, må utstyr som er plassert ute i felt kapsles inn. IP44 bør tilfredsstille kravet for tilstrekkelig innkapsling, men bør testes før bruk. Utstyret må også være laget for å tåle vintertemperaturer. Enheten skal bruke et batteri som energikilde. Det må innføres tiltak for strømsparing og batteriet må dimensjoneres for strømforbruk, slik at vedlikeholdsintervallet ikke blir for hyppig. Det finnes strategier for strømsparing både i software og hardware som kan implementeres på mikrokontrolleren. Nærmere vurdering av faktisk utstyr vil kunne fastslå om strømforbruket er tilstrekkelig lavt slik at batteriet ikke blir for stort. Hvis dette ikke er oppnåelig må det vurderes alternativer for å lade batteri, f.eks. solcellepanel. Utstyret vil også kreve en form for antenne for å sende informasjonen trådløst til mottaker.

3.1 Utforming av mulige løsninger

Småkrafts forslag til løsning på problemstillingen består i å måle nedbørsmengde i nedslagsfeltet og måle vannføring i tilførsels- elver og bekker og forutse mengden energi som kan produseres. På bakgrunn av dette forslaget har vi kommet frem til følgende løsningsalternativ.

3.1.1 Løsningsalternativ 1

Løsningsalternativ 1 er en videreutvikling av Småkrafts forslag om å bruke et trådløst sensornettverk til å måle nedbør og vannføring. Systemet vil bestå av en basestasjon i form av en mikrokontroller, med batteri, som mottar data fra en vannføringssensor og en værstasjon som inneholder blant annet en automatisk nedbørmåler. Mikrokontrolleren og nedbørmåleren vil bli plassert på land mens vannføringssensoren monteres undervanns i elv. Mikrokontrolleren vil motta data fra sensorene, lagre database lokalt og sende dem trådløst til kraftverket før de lokale dataene fjernes igjen. Flere basestasjoner kan implementeres i systemet. Disse vil kommunisere trådløst med kraftverket.

3.1.2 Løsningsalternativ 2

Løsningsalternativ 2 vil basere seg på bruk av kun nedbørmålere og temperatursensor. Systemet vil bestå av én eller flere stasjoner med hver sin mikrokontroller, automatisk nedbørmåler, temperatursensor og batteri som til sammen utfører målingene og sender dataene trådløst til en mottakende mikrokontroller ved kraftverket. Mottakerenheten vil lagre alle dataene i en database og derifra kan de brukes for kalkulasjon av energiproduksjon.

3.1.3 Vurderinger i forhold til verktøy og Hardware/Software komponenter

Vi har vurdert en del forskjellige instrumenter som er aktuelle for prosjektet og vi har valgt å dele inn utstysbehovet inn i tre kategorier; Måling av vannføring, måling av nedbør og systemet som helhet.

3.1.3.1 Vannføring

Vannføring er hastigheten på en mengde vann som renner i elv/bekk, og regnes ut ved å ta volumet av en «skive» av elven og måle hastigheten på vannet i den skiven. For å måle vannføring i elven så finnes det tre alternativer; måling med flygel, fortynningsmetoden og dopplerinstrument.

Flygel er et mekanisk instrument med propell som settes ned i vannet. Propellen roterer proporsjonalt med hastigheten vannet beveger seg i.

Ved bruk av fortynningsmetoden helles en kjent mengde salt lengre oppe i en elv. Dette vil blande seg jevnt i vannet. Lengre ned i elven blir det plassert ut sensorer som måler ledningsevnen til vannet i elven. Vannstrømningen beregnes ut fra ledningsevnen som er proporsjonalt med saltnivået.

Et akustisk dopplerinstrument monteres under vann enten på bunnen i elven eller på en av sidene. Dopplerinstrumentet måler hastigheten til suspenderte partikler i vannet og vil ut ifra det kunne måle hastigheten i vannet. I tillegg kan det kartlegge geometrien og observere vannstand i elven og kan ut ifra alt dette kalkulere vannføringen. Det finnes dopplerinstrument designet for å monteres permanent for å ta målinger over lengre tid. Ulempen er at et slikt instrument er svært kostbart, rundt 1000 USD, og at det er et meget komplisert og omfattende apparat. [2]

Grunnet dopplerinstrumentets høye kostnad er det uaktuelt å bruke den i prosjektet. Da står alternativene på flygel eller fortynningsmetoden. Både måling med flygel og fortynningsmetoden anses som «manuelle» målinger (det er behov for at personell utfører målingene fysisk). Ettersom et av kravene til systemet er at det skal være mest mulig automatisk og at det skal utføres flere målinger hver eneste dag, egner begge disse metodene seg dårlig som permanent løsning.

3.1.3.2 Nedbørmåling

Nedbørmålingene krever kontinuerlig monitorering, minimalt vedlikehold og en løsning som kan kobles direkte til en mikrokontroller. Vi har vurdert to typer automatiske nedbørmålere som er aktuelle; «tipping bucket» og «weighing precipitation gauge».

Prinsippet til en «tipping bucket» er at regn renner ned gjennom en trakt og i et av to små kar som fylles, når det ene karet er fylt tipper den over og tømmes mens det andre fylles. Det er kjent hvor mye vann som er i karet når den tipper. Det kan regnes hvor mye nedbør det er ut ifra hvor ofte karet tipper. Metoden er noe mindre nøyaktig fordi det kan slutte og regne med karet delvis fylt, noe som kan gi et misvisende signal ved neste måling. Avanserte modeller kan måle snø, sludd og hagl, men krever da et større apparat og varmeelement. Denne typen er mye dyrere, mer komplisert å implementere, og vil øke strømforbruk drastisk. [3] [4]

En «weighing precipitation gauge» er en nedbørmåler som måler massen til nedbøren i en beholder, og kan da fastslå hvor mye nedbør som har falt ved å måle økning i masse over tid. Måleren er mer komplisert å implementere enn en «tipping bucket», men den er mer nøyaktig. Denne typen kan monteres med enten frostvæske eller varmeelement som gjør det mulig å måle snø, sludd og hagl. [5]

Vi har valgt å gå for en «tipping bucket»-løsning, av den enklere varianten. Det er den minst kompliserte måleren å implementere i løsningen og det er et budsjettvennlig alternativ. Ulempen er at vi ikke vil ikke kunne måle nedbør annet enn regn.

3.1.3.3 System og Software

Vi har valgt å styre systemet med mikrokontrollere, og for valg av mikrokontroller og programvare har vi valgt å bruke Arduino. Dette på grunnlag av at vi har brukserfaring og god kunnskap om Arduino fra tidligere i studiet, det eksisterer store mengder dokumentasjon, ressurser og kompatible komponenter for å implementere begge løsningsalternativene, og kostnadene for utstyret er relativt lavt.

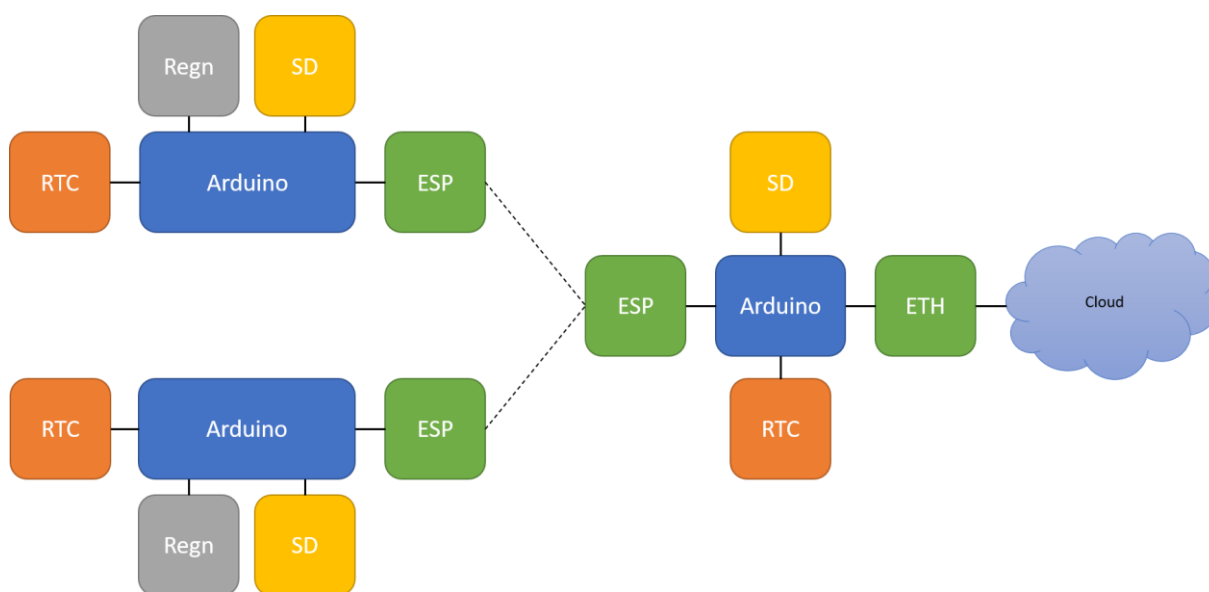
Av temperatursensorer har vi sett på en DHT22 – temperatur og luftfuktighetsmåler. De er relativt billige, enkle å implementere i løsningen, og kan operere i temperaturer mellom -40°C til 80°C. Den kan også måle relativ *RF* 0-100%. [6]

3.2 Konklusjon

Vi har vurdert begge løsningsforslag og kommet frem til at valget faller på Løsningsalternativ 2. Løsningsalternativ 2 er et sensornettverk av automatiske nedbørmålere som samler inn data og sender disse dataene til en mottakerenhet ved kraftverket. Løsningsalternativ 1 har, ifølge oss, størst potensial for å løse problemstillingen, men er ikke finansielt gjennomførbart som et bachelorprosjekt. Vi har heller ikke nok forkunnskap om utstyr for måling av vannføring. Løsningsalternativ 2 er derfor den beste løsningen fordi vi har tilstrekkelig kunnskap til å gjennomføre prosjektet og er mindre finansielt krevende å realisere.

4 Realisering av valgt løsning

Etter å ha gjennomført forstudien, og gått for et nedbørsensornettverk som valgt løsning, fortsatte vi med research av hardwarekomponenter for å realisere systemet. Etter samtaler med ekstern veileder fra Småkraft ble vi informert om en del begrensninger som setter ytterligere krav til hvordan systemet må utformes. Plasseringen av utstyret vil være langt fra selve kraftverket sammen med annet måleutstyr som Småkraft allerede har plassert ut. Derfor trengtes det en mottakerenhet som mottar dataene fra målerne, og sender disse videre til Småkraft over internett via kabel. Mottakerenheten vil ha mulighet for å kobles til strøm, og hensyn til batterilevetid er derfor ikke relevant for denne enheten. Figur 1 viser generelt hvordan hele systemet skal settes sammen med de ytterligere kravene fra Småkraft.



Figur 1: Blokkfigur system. Systemet illustrer to Arduino senderenheter (venstre) og en Arduino mottakerenhet (høyre).

En del av komponentene som var nødvendig for å bygge enhetene hadde vi allerede slik at arbeidet kunne begynnes raskt. Vi eide allerede 2 stk. Arduino Uno, RTC, WiFi-moduler, *Breadboard* til å koble komponentene sammen, samt kabler. Resten av komponentene ble kjøpt inn umiddelbart slik at prosjektet ikke ble forsinket fordi essensielle deler ikke var kommet frem. Deretter begynte bygging og koding av systemet; først senderenheten deretter mottakerenheten.

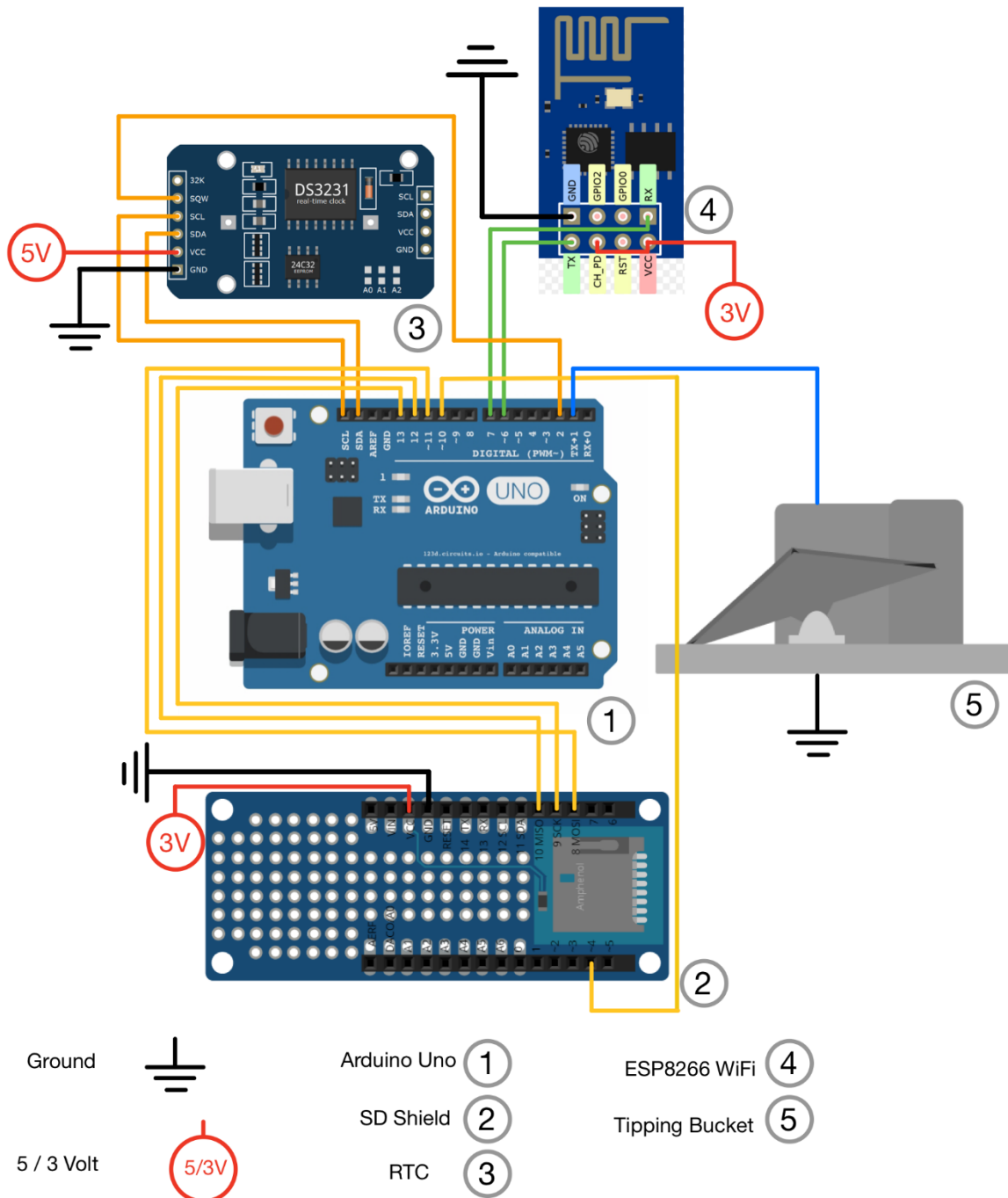
4.1 Senderenhet

I tillegg til selve mikrokontrolleren, en Arduino Uno, og nedbørsensoren består senderenheten av en SD-kortleser, en real-time clock (RTC) og en WiFi-modul av typen ESP-01 [7], koblet sammen på et breadboard, som vist i Figur 1. Systemet er drevet på strøm via USB under utvikling og testing. Mikrokontrolleren har også en spenningsregulator med en ordinær DC-plugg for bruk av strømadapter eller batteripakke.

4.1.1 Design

SD-kortleseren ble implementert for å kunne mellomlagre dataene før den sendes til mottakerenheten, RTC for å gir dataene nøyaktig tidsstempling og WiFi-modulen for å motta og sende dataene. Gjennom research av RTC, under implementering, ble det oppdaget at den har en innebygd temperatursensor med hovedoppgave å kalibrere klokken basert på omgivelsestemperaturen.

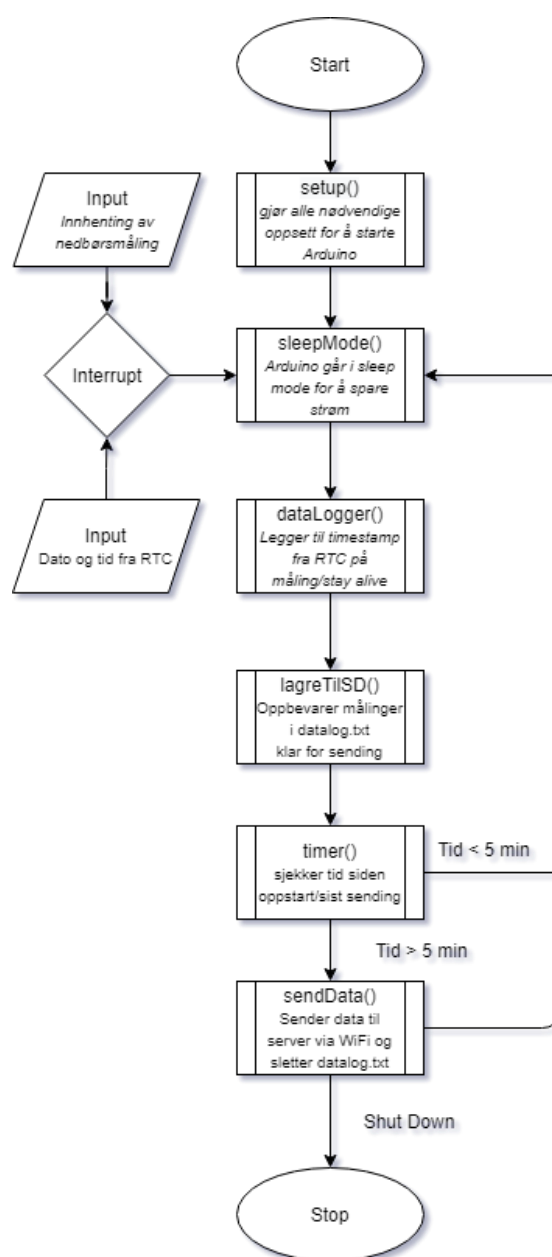
Dataene fra den kan hentes ut og presenteres. Temperaturavlesningene ble vurdert som nøyaktig nok slik at vi ikke trengte å implementere en ekstern temperatursensor. Det er imidlertid fortsatt mulig å legge til en ekstern sensor på et senere tidspunkt om ønskelig. På «Figur 2: Skisse senderenhet» så ser en hvordan hele systemet er satt sammen. SD-kortleser og WiFi-modul drives på 3V fra Arduino, mens RTC drives på 5V. Nedbørmåleren er koblet til interrupt-pinne 2 på Arduino Uno. I tillegg til vanlig tilkobling for kommunikasjon så er RTC også koblet til interrupt-pinne 3 for å kunne vekke Arduino fra hvilemodus ved hjelp av en alarm.



Figur 2: Skisse senderenhet

4.1.2 Kode

Før selve logikken ble implementert i systemet ble hver modul koblet til og funksjonstestet individuelt, før de ble ført sammen i den helhetlige koden. I Figur 3 så vises hvordan hele koden fungerer prinsipielt. Når enheten skrues på går den inn i en setup-fase hvor RTC, SD-kortleser og WiFi-modul verifiseres at fungerer korrekt. Deretter går den i hvilemodus som den vekkes ut av når den får et signal fra nedbørmåleren om at det er registrert nedbør. Deretter logger den dataene ved å putte en dato- og tidsstempling på målingen, legger til temperatur og lagrer dette som en tekststreng på SD-kortet. På et 5 minutters intervall sender den all data på SD-kortet til mottakerenheten. Dette intervallet holdes av RTC. Arduino-en blir også vekket på et 30 minutters intervall hvor det blir registrert en måling uten nedbørdata, med datostempling og temperatur, som blir sendt til mottakerenheten. Dette for å gi beskjed om at den fortsatt har kommunikasjon med mottakerenheten. Det er ingen automatiserte prosesser i koden, hverken på sender- eller mottakerenheten for dette, men mangel på mottatte meldinger vil kunne oppdages manuelt.



Figur 3: Flytskjema kode senderenhet

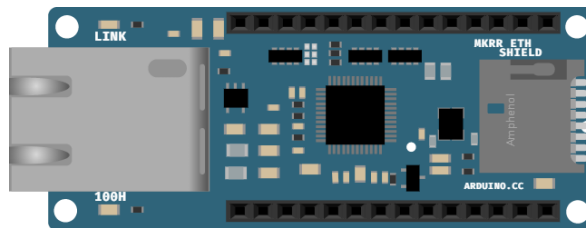
4.2 Mottakerenhet

Underveis i arbeidet med senderenheten ble arbeidet med mottakerenheten startet. Dette for å kunne teste kommunikasjon mellom enhetene før senderenheten kunne bli ansett som ferdig. Først ble kode for å teste mottak og lagring av informasjon lastet opp på hardwaren for en senderenhet, deretter ble selve enheten satt sammen og skrevet dedikert kode til.

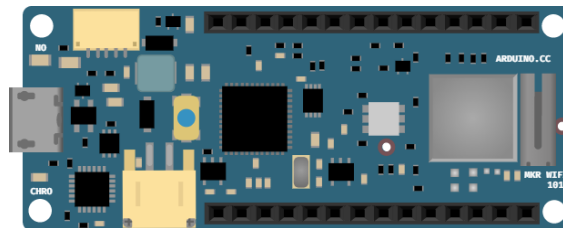
4.2.1 Design

I utgangspunktet skulle mottakerenheten bestå av en Arduino Uno, lik som senderenhetene, og ettersom vi i utgangspunktet skulle ha to senderenheter ble det tidlig bestilt inn en ekstra for mottakerenheten. Underveis i arbeidet med mottakerenheten innså vi at Uno-en ikke hadde tilstrekkelig minne, og det ble kjøpt inn en Arduino MKR WiFi. Dette er utdypet videre i 4.3 Influx. Denne hadde flere komponenter integrert i kretskortet sammen med mikrokontrolleren, bla. en RTC og WiFi-modul, og trengte derfor kun mottak for SD-kort og Ethernetport. Dette ble gjort enkelt ved å benytte et ekspansjonskort for Arduino MKR som gir kortet nettopp en Ethernetport og en port for SD-kort, og plasseres oppå selve mikrokontrolleren. Disse vises i Figur 5: Ethernet Shield

og Figur 4: Arduino MKR.



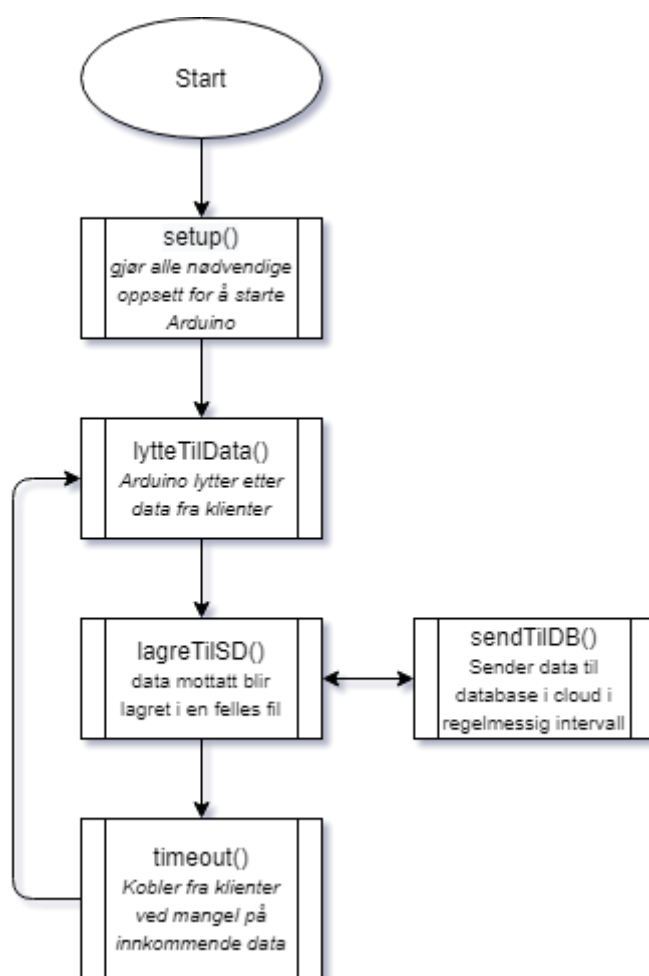
Figur 5: Ethernet Shield



Figur 4: Arduino MKR

4.2.2 Kode

Figur 6 viser hvordan koden fungerer prinsipielt. Ved oppstart oppretter mottakerenheten et aksesspunkt og en server som senderenhetene kan koble seg opp mot. Deretter sjekker den kontinuerlig etter klienter som kobler seg til og lagrer all data som kommer inn på SD-kortet. Når senderenheten kobler seg fra serveren, som den skal gjøre før den går i strømsparingsmodus, så går mottakeren tilbake etter å vente på klient. Hvis, mot formodning, senderenheten ikke kobler seg fra serveren, har mottakeren en timeout-funksjon som kobler fra klienter som er koblet til over lengre tid uten at det kommer inn data. Etter planen skulle dataene eksporteres videre til Småkraft via internett, men uforutsette problemer gjorde at dette målet ikke ble oppnådd. Dette er videre beskrevet i 4.3 Influx.



Figur 6: Flytskjema kode mottakerenhet

4.3 Influx

Utover i prosjektet ble vi oppfordret til å benytte InfluxDB for lagring av dataene våre. InfluxDB er en database som kan håndtere store nettverk med IoT-sensorer og samler inn data for analyse og overvåking av data. Data kan bli sendt inn enten via Influx sin eget toolkit eller via deres skyløsning. All data blir tidsstempelt for lagring og en kan enkelt se på tidligere data eller data i sanntid med god nøyaktighet. InfluxDB har et bredt klientbibliotek med mange kjente løsninger som for eksempel C#, Python, JavaScript m.m. Heldigvis for oss har InfluxDB til og med et klientbibliotek for Arduino, og det var med dette vi skulle implementere databaselagring for systemet vårt. Småkraft har egen InfluxDB-profil som de benytter til å overvåke sine egne sensornettverk. Vi fikk tildelt en bruker og tilgang til et prosjekt opprettet for oss og vårt arbeid.

4.3.1 Oppsett

Vi benyttet InfluxDB Cloud via nettleser til å sette opp for innsamling av data. Her logget vi inn, med brukeren vi hadde fått tildelt, i "organisasjonen" til Småkraft. Det ble laget en "bucket" som er hvor den spesifikke dataen vår blir samlet. Når bucket-en blir laget generer InfluxDB skreddersydd kode til de forskjellige klientbibliotekene for å nå ønsket bucket. Her er en URL inkludert som bestemmer hvor dataen sendes over nettet. Med gratis bruker kan en velge en av Influx sine servere, men Småkraft har sin egen og dermed sin egen URL også. Det siste som trengs på InfluxDB er et "token". Tokens kan genereres og kreves for tilknytning til de forskjellige bucket-ene.

4.3.2 Krav

For å realisere InfluxDB for mottakerenheten, måtte vi først installere Arduino sitt bibliotek for InfluxDB, ESP8266 InfluxDB. Dette biblioteket kommer med en samling av flere andre biblioteker og bruker andre som er standard i Arduino *IDE* Dessverre oppdaget vi at dette biblioteket ikke støtter alle Arduino-kort, det støtter bare de to kortene ESP8266 og ESP32.

Etter at oppdragsgiver var blitt informert om problemene med kompatibilitet ble vi oppfordret til å benytte SSL tilknytning mellom InfluxDB og Arduino-enheten vår for sikker kommunikasjon og heller sende meldinger over REST-API. Det viste seg at enheten som vi brukte, Arduino Uno, hadde for lite internt minne for å bruke SSL biblioteket, og det ble kjøpt inn en Arduino MKR.

4.3.3 Realisering

Vi testet flere metoder for å realisere REST-meldinger fra Arduino-en til InfluxDB, men dette viste seg å bli for utfordrende. Vi var ikke i stand til å finne en metode for å sende data til Influx uten å bruke det dedikerte biblioteket for ESP. Før vi startet på bachelorprosjektet så hadde vi en ESP-32 liggende, og vi forsøkte å laste inn biblioteket og eksempelkode for å se om vi klarte å opprette kommunikasjon. Etter mye prøving, feiling og testing klarte vi å få sendt data til InfluxDB ved å bruke det spesifikke kortet.

4.3.4 Konklusjon

Det var en utfordring å implementere Influx i løsningen, og vi brukte betydelig mye tid på å få det til å fungere. Dessverre var dette såpass seint i prosjektet at vi begynte å bli presset for tid. Selv om vi klarte å få kommunikasjon mellom ESP32 og InfluxDB, var det alt for seint til å implementere det kortet inn i resten av prosjektet og vi var nødt til å kutte ut Influx som en del av den endelige løsningen.

5 Testing

Vi har valgt å gjennomføre tre tester av systemet som er viktige for å måle om systemet etterkommer kravspesifikasjonen, og generelt vurdere ytelsen til systemet; kalibrering av nedbørmåler, måling av strømforbruk og rekkevidden på den trådløse overføringen mellom sender og mottaker ble målt.

5.1 Kalibrering av nedbørmåler

Enheten benytter en «tipping bucket» for å måle mengden nedbør. Måleren har en trakt som regnet renner ned i til en vippearms som har to beholdere. Hvordan måleren fungerer er beskrevet i kapittel 3.1.3.2. Produsenten opplyser at en puls tilsvarer 0.2794mm nedbør [8], og vi ønsket å teste dette for å kunne kalibrere systemet.



Figur 7: Kalibrering av nedbørmåler

5.1.1 Bakgrunn

Millimeter nedbør regnes frem ved å ta arealet av beholderen som vannet kan komme inn i, måle mengden vann i beholderen, og dele disse verdiene slik:

$$\frac{\text{kubikkmillimeter vann [mm}^3\text{]}}{\text{areal av trakt [mm}^2\text{]}} = \text{millimeter nedbør [mm]}$$

For å nøyaktig vite hvor mye vann som puttes inn i måleren benyttet vi en liten 1 ml sprøyte slik at mengden vann ved hver puls kunne måles så nøyaktig som mulig og skrives ned. Vi registrerte 10 pulser og tok et gjennomsnitt av disse. Dette er ikke nok for å gi et helt korrekt resultat på hvor unøyaktig måleren er, dette ville krevd betydelig flere målinger, men vil gi oss en bedre kalibrering enn å kun gå ut ifra produsentens verdier.

Gjennom forstudiet ble det avklart at en «tipping bucket» er en mindre nøyaktig form for måling av nedbør, og det viste også resultatene, med en klar differanse mellom oppgitt verdi fra produsent og faktiske målinger.

Variabel	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10
Verdi (ml)	1,35	1,88	1,64	1,95	1,43	1,80	1,40	1,53	1,41	1,93
Resultat (mm)	0,2455	0,3418	0,2982	0,3545	0,2600	0,3273	0,2545	0,2782	0,2564	0,3509

Oppgitt verdi [mm]	0,2794
Areal (mm ²)	5500
Gjennomsnitt [mm]	0,30
Gjennomsnitt [%]	6,2016

Tabell 1: Resultat kalibrering av nedbørmåler

Som vist i Tabell 1 fikk vi både verdier over og under oppgitt verdi fra produsent, med et gjennomsnittlig avvik på 6,2 prosent.

5.1.2 Konklusjon

Måleren viste seg å gjennomsnittlig rapportere mindre nedbør enn faktisk målte verdien basert på produsentens verdier. Dette er viktig kunnskap slik at systemet blir kalibrert til den nye verdien. Kalibrering av målere på generell basis er noe som alltid må gjøres for å forsikre korrekt data, vist gjennom denne testen.

5.2 Strømforbruk

Vi har gjennomført en strømforbrukstest av senderenheten. Dette for å kunne gjøre vurderinger på hvor stor kapasitet batteriet bør ha for å oppnå ønsket batterilevetid. Testen ble gjennomført med multimeter koblet inn mellom spenningsforsyningen og spenningsregulatoren til mikrokontrolleren for å overvåke det totale strømforbruket til kretsen.

5.2.1 Bakgrunn

Ifølge kravspesifikasjonen skal ikke systemet kreve vedlikehold oftere enn hver 6. måned. Dette kravet kom som et resultat av at det endelige produktet vil bli implementert i et stort omfang hvis det blir tatt i bruk. Det vil resultere i et stort antall arbeidstimer hvis vedlikeholdsintervallet på systemet er for hyppig. Resultatet av strømforbrukstesten vil derfor brukes til å vurdere dimensjon på batteriet som gir en batterilevetid så nært 6 måneder som mulig, men faktorer som batteriets fysiske størrelse og pris vil også bli tatt med i denne vurderingen. Systemet ble testet i to forskjellige program; blank kode og vanlig kode. Blank kode ble testet for å se hvor mye enheten med alle komponentene i utgangspunktet brukte av strøm, for deretter å se hvor stort strømforbruk systemet hadde med vanlig kode. Det ble tatt tre målinger over tid fra hvert program.



Figur 8: Bilde av testoppsett strømforbruk

Resultatet er vist i tabell 2. Det er en markant forbedring av strømforbruk mellom blank kode, og kode slik den er implementert med strømsparing. Systemet har økt strømforbruk under sending, men sendetiden er liten nok til at vi vurderer det som ikke signifikant sammenliknet med perioden den vil være i strømsparingsmodus.

Program	Test 1	Test 2	Test 3
Blank kode [mA]	117,7	119	117,8
Vanlig kode [mA] (sending)	44 (100)	45(100)	43,8 (100)

Tabell 2: Resultat strømforbrukstest

Ved å ta resultatet av strømforbruket med vanlig kode og gange det opp med antall dager vi ønsker at systemet skal kjøre uten batteriskifte får vi disse resultatene:

$$45mA * 24 \text{ timer} * 30 \text{ dager} = 32400 \text{ mAh}$$

$$45mA * 24 \text{ timer} * 180 \text{ dager} = 194400 \text{ mAh}$$

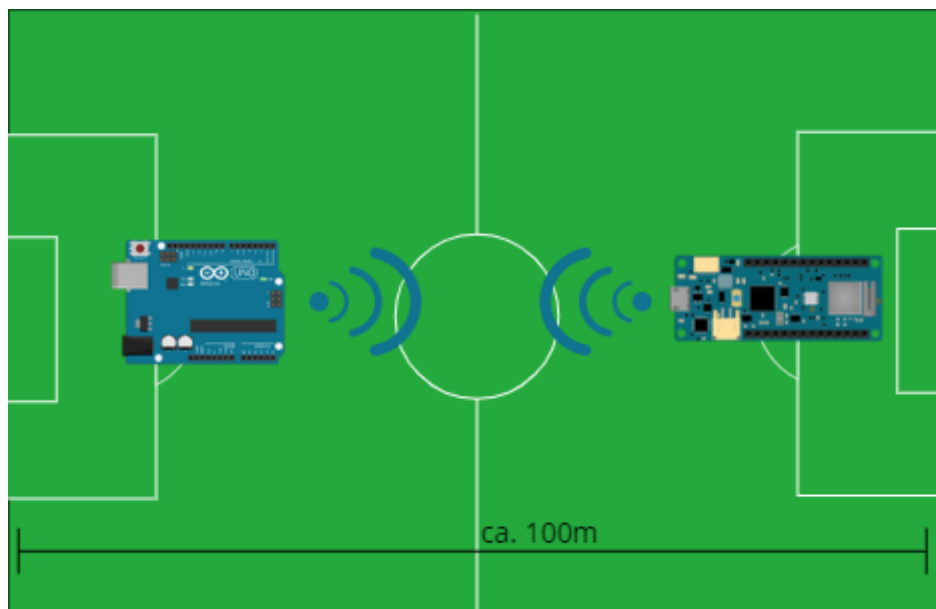
Resultatene viser en meget høy batterikapazität for å etterkomme de krav vi har satt til batterilevetid for systemet.

5.2.2 Konklusjon

Grunnet systemets relativt høye strømforbruk kreves det en batterikapazität som er urealistisk høy for å etterkomme krav til batterilevetid. Det eksisterer batteriprodukter på markedet som kan levere en måneds batteritid, men det vil være kostbart. For at et eventuelt ferdig produkt basert på dette prosjektet skal tas i bruk kreves det store forbedringer både gjennom lavere forbruk, og vurdering i forhold til lading av batteriet f.eks. gjennom bruk av solcellepanel.

5.3 Kommunikasjonsrekkevidde

For å vurdere rekkevidden til systemet ble det utført en rekkeviddetest. Hensikten med testen er å vurdere hvor stor avstand det kan være mellom sender- og mottakerenheten. Vi estimerer en rekkevidde på ca. 100 meter basert på anekdotiske bevis fra brukertester på forum.



Figur 9: Rekkeviddetest

5.3.1 Bakgrunn

Rekkeviddetesten ble utført på en fotballbane på ca. 100 meters avstand, i klart vær, for å fastslå maksimal rekkevidde for systemet under ideelle forhold, vist i Figur 9: Rekkeviddetest. Enhetene ble holdt ca. 1 meter opp fra bakken. Mottakerenheten opprettet et WiFi-aksesspunkt og ble holdt i den ene enden av banen, og den andre enheten prøvde å koble til aksesspunktet, hvorpå det ble lest av signalstyrke. Resultatene fra tabell 3 viser en støytterskel på -89 dBm, og ble oppnådd på ca. 80 meters hold.

Program	Test 1	Test 2	Test 3
Maks avstand (80m) [dBm]	-89	-89	-89
Kloss hold (1-2m) [dBm]	-69	-69	-69

Tabell 3: Resultat rekkeviddetest

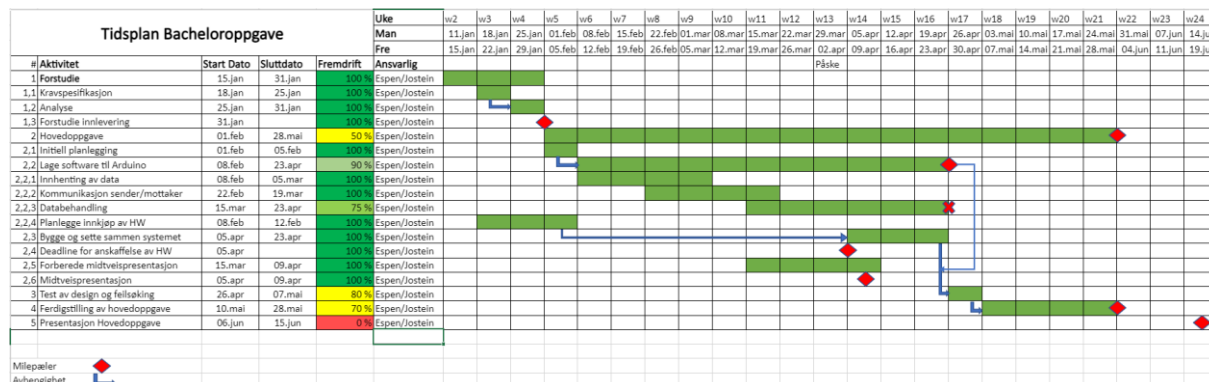
5.3.2 Konklusjon

Rekkeviddetesten ble vurdert som en suksess basert på våre estimater og resultatene av testen. En forbedring av rekkevidde vil kreve et mer kostbart nettverkskort med mulighet for en ekstern antenne for enten én eller begge enhetene.

6 Diskusjon

6.1 Tidsbruk

Vi har tidlig i prosjektet laget en fremdriftsplan for hvordan tidsbruket til de forskjellige oppgavene skulle fordeles.



Figur 10 Fremdriftsplan

Vi jobbet godt og holdt fremdriftsplanen gjennom nesten hele prosjektet. I starten var vi effektive og var foran skema. Det var rundt seint «Kommunikasjon sender/mottaker» og «Databehandling at vi begynte å problemer med tidspress. Under mesteparten av prosjektet har vi jobbet i lag side om side og holdt på med de samme eller litt forskjellige oppgavene. Ved enkelte oppgaver som ikke krevde mye kommunikasjon har vi jobbet hjemme hver for oss. Fremdriftsplanen er ytterligere beskrevet i 7Appendiks D .2.

6.2 Komponenter

Gjennom prosjektet har vi planlagt hvilke komponenter som skulle brukes, men noen av dem har underveis blitt byttet ut eller blitt fjernet, eller noen helt nye komponenter har blitt satt inn i prosjektet. Vi gjør rede for hvilke komponenter vi initialt tenkte å bruke, hvilke endringer som skjedde underveis og litt om hva vi føler vi kunne gjort annerledes.

6.2.1 Forstudiet

I den opprinnelige ideen kom vi fram til i forstudiet. Fra den skulle systemet vårt basere seg på Arduino plattformen ved bruk av utviklerkortet Arduino Uno. Det skulle da være Uno-enheter som tar alle målinger og sender dem inn til en Uno-mottaker som prosesserer alle dataene. I det første designet skulle senderen inneholde en del ekstra moduler, men her er en liste med alle komponentene (med unntak av ledninger):

- Arduino Uno – brukes som hovedkort
- Tipping bucket, med RJ-11 tilkobling - for å måle nedbør og tilkobling til hovedkort
- RTC – for tidsstempling og alarmer
- ESP8266 – For WiFi signal og kommunikasjon
- DHT22 – For å måle temperatur

Mottakeren skulle også inneholde ESP8266 og RTC, men i tillegg skulle den ha SD-kortleser for lagring av dataene. Vi tenkte det ikke ville være nødvendig med SD-kort på senderen ettersom dataene ikke skulle lagres der, men heller sendes og slettes.

6.2.2 Midtveis

Da vi begynte med å programmere komponentene våre fant vi fort ut av feilene i systemet. Et av kravene til senderen var at den skulle gå på batteri, og dermed var en av måtene vi tenkte å spare strøm på å ikke sende data konstant, men heller i intervaller med en viss tid mellom. Dette medførte at vi måtte lagre data på senderen i en liten periode og da innså i fort at det interne minnet på Arduino Uno-kortet var så lite at det ikke var egnet til noe form for lagring. Derfor måtte vi anskaffe SD-kort til senderne også. Vi oppdaget samtidig at RTC-komponenten som vi brukte hadde innebygget temperaturmåler som vi valgte å benytte oss av og dermed gikk vekk ifra DHT-22 temperaturmåleren som vi tenkte i starten.

Vi hadde nå kommet opp med et nytt oppsett med komponenter:

- Arduino Uno – brukes som hovedkort
- Tipping bucket, med RJ-11 tilkobling - for å måle nedbør og tilkobling til hovedkort
- RTC – for tidsstempling og alarmer
- ESP8266 – For WiFi signal og kommunikasjon
- Arduino SD Shield TSX0004 – for lokal lagring av data

Oppdragsgiver ønsket SSL tilknytning for kommunikasjon med database. Mottakeren som også skulle være en Arduino Uno og som skulle sende data via SSL hadde for lite internt minne for å i det hele tatt benytte seg av SSL biblioteket som var tilgjengelig på Arduino-plattformen. Løsningen vi kom opp med var da å erstatte kortet med et annet som hadde mer minne tilgjengelig. Vi anskaffet dermed et annet kort: Arduino MKR WiFi 1010. Dette var et fysisk mindre kort, men med mer minne, innebygget WiFi enhet og innebygget RTC. På kraftverkene til Småkraft ville det være mulighet for kablet nettverkstilkobling. Vi fant derfor en utvidelsesmodul til MKR-en med både ethernetport og SD-lagring på. Mottakersystemet vil da bestå av:

- Arduino MKR WiFi 1010
- Arduino MKR Ethernet Shield

De to komponentene til mottakeren kan settes over hverandre og behovet for ledninger er redusert til kun strøm- og ethernetkabel.

6.2.3 Etertanke

Vi brukte mye tid på å bestemme oss for hvilke komponenter vi skulle bruke i prosjektet og generelt så viste de fleste valgene seg å være nyttige. Den største tabben vi henholdsvis gjorde var å velge Arduino Uno-kortet som kontroller uten å gjøre omfattende research på kortet og dens egenskaper. Vi hadde tidligere brukt Uno i studiet og til de små oppgavene som ble utført i undervisningen var kortet velegnet. For også å spare penger, siden vi hadde kortet fra før, tenkte vi det var en god ide. Vi tok dermed Uno-en for “god fisk” og konsentrerte oss heller om alle de andre komponentene og gikk ut ifra at systemet skulle bygges rundt Uno som hovedkort. Dette viste seg å være en stor tabbe da det største hinderet til systemet viste seg å være den Uno-ens meget begrensede minne. Hadde vi fått begynt på nytt hadde vi heller valgt å gå for løsningen som ble til for mottakeren for senderen. Den var litt mer kostbar, men selve kortet er mindre og har alle komponentene innebygd i seg eller i like store utvidelseskort som modulært kan utvides med som også gjorde systemet mye “renere” og uten ledninger over alt. Deretter ville vil valgt en ESP-32 for mottakerenheten ettersom det er det eneste kortet som er kompatibelt med Influx sitt bibliotek for Arduino.

6.3 Veien videre

Ut ifra kravspesifikasjonene og hvordan vi så for oss at systemet skulle ende opp er det en del mangler som vi ikke fikk implementert.

6.3.1 Utbedre kode

Vi har skrevet en del kode som fungerer godt til oppsettet vårt. Vi har løst problemer som sending og dyp søvnmodus m.m. Oppsettet vi endte på var et server / klient nettverk hvor mottaker fungerer som server og sendere fungerer som klienter. For dette har vi brukt Arduinos biblioteker som kan brukes sammen med ESP8266 enheten. Slik koden vår er bygd klarer vi fint å få tilknytting til flere klient og å motta meldinger fra disse, men har ingen måte å skille dem annet enn å føre inn i datastrengen hvilken klient dataene kom fra og vil kreve ytterligere koding og testing.

Systemet skal kunne måle nedbør. Så langt vi har kommet, registrerer systemet en puls hver gang tipping bucket-en tipper. Noe så bør løses videre er dermed å sikre en metode som utfører selve beregningen av hvor mye nedbør alle pulsene utgjør. Enten dette blir gjort i koden til sender eller mottaket, eller om dette blir gjort på InfluxDB.

6.3.2 Komponenter

Systemet i prosjektet har utviklet seg i løpet av prosjektperioden. Mottakeren har blitt byttet fra en Arduino Uno til en Arduino MKR WiFi 1010. Dette hadde vært gunstig å gjøre på sendersiden også, og bytte ut mottakeren med en ESP-32. Ved å gjøre det erstattes fem uavhengige deler, som må kobles sammen med ledninger (Arduino Uno, SD-kortleser, RTC, ESP8266, nedbørmåler), med to deler som kan modulært festet oppå hverandre (Arduino MKR WiFi 1010, SD-kortleser) og et breakout board for å koble til en RJ-11 kontakt (Nedbørmåler). På denne måten blir systemet mindre begrenset av mengde minne, men det blir også et «renere» system med mindre løs kabler og ulike komponenter. ESP-32 vil kunne kommunisere med Influx, dog det ville krevd en annen løsning enn Ethernet Shield får å gi den tilgang til internett.

Ifølge kravspesifikasjonene skal systemet tåle kuldegrader og ha en viss vanntett innkapsling. Denne innkapslingen var noe vi ikke hadde tilstrekkelig tid til å løse. Forslag til videre arbeid vil være å finne en god måte å pakke inn systemet i en boks som hindrer/begrenser at nedbør og fukt påvirker systemet negativt.

Vi har hele veien planlagt at systemet skal gå på batteri ettersom meningen var å installere det i felt uten strøm tilgjengelig. Vi har testet en del hvor mye strøm systemet bruker for å finne passende størrelse på batteri. Vi derimot ikke planlagt nøyaktig hvilken løsning som passer best som batteripakke og har heller ikke gått til anskaffelse av noe batteri. Å finne en løsning for lading med f.eks. solcellepanel mener vi vil være mest effektivt, men forbedring og optimalisering av strømforbruk og anskaffelse av batteri er også anbefalt som videre arbeid.

7 Konklusjon

Vi oppnådde de viktigste punktene for denne oppgaven som var å registrere nedbørsmålinger og opprette kommunikasjon mellom sender- og mottakerenhet. Dessverre rakk vi ikke å implementere sending av data til Influx i oppgaven, men oppnådde noen resultater ved bruk av en annen type mikrokontroller. Testing viser at det finnes stort forbedringspotensial når det gjelder strømforbruk og vil være en viktig faktor for at systemet skal kunne bli implementert som en løsning for Småkraft.

Disse er klare punkter som vi mener bør fokuseres på i fremtiden:

- Innkapsling – som en del av kravspesifikasjonen skulle løsningen ha en innkapsling for å tåle vær, vind og kaldere temperaturer. Dette var noe ikke vi rakk å gjennomføre i prosjektperioden, og er noe som må tas tak i videre.
- Batteri og lading – fremtidige oppgaver bør fokusere på å realisere en løsning med lading f.eks. bruk av solcellepanel. Dette vil kunne gjøre at systemet kan fungere mye lengre uten behov for å bytte batteri. Dessuten er det få batterier på markedet i dag ment for et slikt system med nok kapasitet til å gi en teoretisk levetid lengre enn 30 dager.
- Influx – vi gjorde ikke nok research på Influx i forkant, og oppdaget for seint at vi ikke hadde valgt et kompatibelt kort for å bruke Influx sitt bibliotek for Arduino. Om Influx skal benyttes som database så må løsningen benytte seg av en ESP-32 som mottakerenhet.
- Valg av mikrokontroller – vi mener vi begikk en feil i å benytte oss av Arduino Uno som vi allerede eide, og ikke gjorde skikkelig research på bruk av andre aktuelle mikrokontrollere. MKR-en var et bedre alternativ ettersom den både hadde en WiFi-modul og RTC på kortet i tillegg til betydelig mer minne. Totalt sett er Arduino MKR et mye bedre kort, uten noen bakdeler sammenliknet med Uno, og prisforskjellen er marginal. Selve løsningen ville også vært mye mer elegant ved å redusere mengden kabler nødvendig og bruk av breadboard.

Samlet sett anser vi prosjektet som vellykket med mange gode resultater underveis, men som krever mer arbeid for å utvikle en komplett løsning.

Referanser

- [1 «Småkraft,» [Internett]. Available: <https://smaakraft.no/om-smaakraft/> . [Funnet 20 Januar] 2021].
- [2 Noregs vassdrags- og energidirektorat, «NVE, Vannstand og vannføring, Målinger og metoder,»] 24 April 2020. [Internett]. Available: <https://www.nve.no/hydrologi/vannstand-og-vannforing/malinger-og-metoder/?ref=mainmenu>. [Funnet 26 Januar 2021].
- [3 Maximum Weather Instruments, [Internett]. Available: <https://www.maximum-inc.com/>. [Funnet] 26 Januar 2021].
- [4 G. Kielland, «Store Norske Leksikon, pluviometer,» 13 Januar 2017. [Internett]. Available:] <https://snl.no/pluviometer>. [Funnet 26 Januar 2021].
- [5 H. Nygård, «Store Norske Leksikon, automatisk nedbørmåler,» 6 September 2017. [Internett].] Available: https://snl.no/automatisk_nedb%C3%B8rm%C3%A5ler. [Funnet 26 Januar 2021].
- [6 Adafruit, «Adafruit - DHT22,» [Internett]. Available: <https://www.adafruit.com/product/385>.] [Funnet 24 Mai 2021].
- [7 Kjell & Company, «Kjell & Company - ESP01,» [Internett]. Available:] https://www.kjell.com//no/produkter/elektro-og-verktoy/arduino/moduler/wifi-modul-for-arduino-esp8266-p87947?gclid=Cj0KCQjwna2FBhDPArisACAEC_UpNybzCmFc-0tpCOWPGAx98XstatmK7kM_cRddIPjb_pKT-WuWjCsaAj5_EALw_wcB&gclid=aw.ds. [Funnet 24 Mai 2021].
- [8 SparkFun Electronics, «Datasheet Weather Sensor Assembly,» [Internett]. Available:] <https://www.sparkfun.com/datasheets/Sensors/Weather/Weather%20Sensor%20Assembly..pdf> . [Funnet 30 April 2021].

Appendiks A Figurliste

Figur 1: Blokkfigur system. Systemet illustrer to Arduino senderenheter (venstre) og en Arduino mottakerenhet (høyre).....	11
Figur 2: Skisse senderenhet.....	12
Figur 3: Flytskjema kode senderenhet	13
Figur 4: Arduino MKR	14
Figur 5: Ethernet Shield.....	14
Figur 6: Flytskjema kode mottakerenhet	15
Figur 7: Kalibrering av nedbørmåler.....	17
Figur 8: Bilde av testoppsett strømforbruk	19
Figur 9: Rekkeviddetest.....	21
Figur 10 Fremdriftsplan	22

Appendiks B Tabelliste

Tabell 1: Resultat kalibrering av nedbørmåler	18
Tabell 2: Resultat strømforbrukstest.....	19
Tabell 3: Resultat rekkeviddetest.....	21
Tabell 4: Risikoliste	29

Appendiks C Forkortelser og ordforklaringer

IP-grad	System for å angi kapslingsgrad til elektrisk utstyr. IP-44 tilsvarer sprutsikkert.
RF	Relativ luftfuktighet. Forholdet mellom vanddampmengden i luften og maksimal vanddampmengde luften kan inneholde. Oppgis i prosent.
Støyterskel	Minimum signalstyrke for fungerende dataoverføring. Oppgis i dBm.
RTC	Real-time Clock. En integrert krets som måler tid.
Interrupt	Et avbrudd i kodens normale forløp for å betjene en annen mer tidskritisk hendelse. Arduino har spesifikke pinner på kortet for å motta signaler som trigger interrupts kalt en interrupt-pinne.
IDE	Integrated Development Environment, software for å lage, kompilere, laste opp og/eller utføre kode. Eks. Arduino IDE, Visual Studio, NetBeans.
Breadboard	Koblingsbrett for bruk i svakstrømskretser. Gjør det mulig å sette sammen og teste elektroniske kretser uten bruk av et kretskort.
Breakout board	Kretskort for å dele (break out) pinner/ledere fra hverandre for å lettere kunne koble til hver enkelt pinne/leder.

Appendiks D Prosjektledelse og styring

Prosjektgruppen består av to personer, Espen Leiknes Larsen og Jostein Ring. Intern veileder for prosjektet er Anne-Lena Kampen, ekstern veileder fra Småkraft er Trond Tollefsen.

D.1 Prosjektorganisasjon

Grunnet liten prosjektgruppe med to personer har det ikke blitt utpekt noen prosjektleder, og arbeidet har blitt jevnt fordelt.

D.2 Fremdriftsplan

Dette er en tekstbeskrivelse av fremdriftsplanen for prosjektet. Fremdriftsplanen er tilgjengelig som et eget dokument.

- 1) Forstudie – Fremdriftsplanen begynner med denne forstudien, som ble ferdigstilt og levert 31. januar 2021.
- 2) Hovedoppgave – Hovedoppgaven er delt inn i flere underpunkt som beskrives under. Frist for innlevering av hovedoppgaven er 28. mai 2021.
 - a) Initiell planlegging – Her inngår videre planlegging som bygger på forstudien, og legger veien videre for hvordan prosjektet skal organiseres og gjennomføres. Forventet å ta 1 uke.
 - b) Lage software til Arduino – Løsningsforslaget benytter seg av flere mikrokontrollere av typen Arduino. Disse trenger software for å kunne operere. Underpunktene er hovedfunksjonene som en eller flere av mikrokontrollerne skal inneha i prioritert rekkefølge. Tidbolkene satt av til de forskjellige hovedfunksjonene er flytende, men endelig frist for ferdig software er satt til 23. april 2021. Dette for å ha tilstrekkelig tid til å teste systemet før rapporten skal skrives.
 - c) Planlegge innkjøp av HW – Initiell vurdering av utstyr som trengs for å gjennomføre løsningen. Frist satt til 5. februar 2021, dog uforutsette behov kan komme i løpet av prosjektet.
 - d) Bygge og sette sammen systemet – Frist satt til 23. april 2021 for å gi tilstrekkelig tid for testing av løsningen.
 - e) Deadline for anskaffelse av HW – Frist satt til 5. april 2021 for å gi nok tid til leveranse av komponenter slik at bygging av systemet kan fullføres innen frist.
- 3) Forberede midtveispresentasjon – Vi skal holde en muntlig midtveispresentasjon av prosjektet som skal holdes mellom 22. mars og 9. april 2021. Det er satt av to uker til dette basert på når vi skal presentere.
- 4) Midtveispresentasjon – Holdes mellom 22. mars og 9. april 2021.
- 5) Test av design og feilsøking – Der er satt av en uke til test av ferdig løsning og feilsøking av dette. Siste frist er satt til 7. mai 2021 for å gi tilstrekkelig tid til å ferdigstille hovedoppgaven.
- 6) Ferdigstilling av hovedoppgave – 4 uker er satt av til å ferdigstille hovedoppgave. Vi skal begge gjennomføre eksamen i andre fag i denne perioden.
- 7) Presentasjon av hovedoppgave – Muntlig presentasjon av hovedoppgaven vil gjennomføres mellom 6. og 15. juni 2021. Arbeid etter frist for innlevering vil bli brukt til å forberede denne.

D.3 Risikoliste

Risikolisten tar for seg ulike scenarioer som kan være til ulempe for prosjektet. Listen inneholder forklaring på hendelser, sannsynligheten for at de skjer, konsekvensen av hendelsen, risikofaktoren hendelsen medfører og hvordan hendelsen kan unngås eller forebygges. Sannsynlighet og risiko vurderes ved et tall mellom 1 og 10 hvor 1 er lav og 10 er høy. Risikofaktor vurderes ved sannsynlighet x risiko og er et tall fra 1 til 100 hvor 1 er liten og 100 er stor risiko.

Hva kan skje	Sannsynlighet	Konsekvens	Risikofaktor	Tiltak
Systemet kan gå ned på grunn av softwareproblem	6	2	12	Nøye testing og utbedring av kode underveis i utviklingen.
Komponenter blir ødelagt/forsvinner på grunn av ytre påkjenninger	6	4	24	Teste innkapsling før montering. Finne årsak til problemet og lage nytt design på montering
Utstyr/verktøy er ikke mulig å anskaffe	5	10	50	Finne alternativer til utstyr/verktøy tidlig
Utstyr/verktøy har for lang leveringstid	7	9	63	Bestille utstyr/verktøy i god tid før implementering
Manglende finansiering	5	8	40	Finne billigere utstyr, simulere deler av løsningen

Tabell 4: Risikoliste