



Utvikling og testing av alternativ metode for henting og lagring av værdata fra målestasjoner uten å benytte en lokal logger

*Forfattere: Ian Philip Eglin, Bjørn Mathias Singh Helseth og Hemund Ulvar Engmark Øyulvstad
Kuben videregående skole, Oslo*

I samarbeid med Meteorologisk Institutt ble det i over en periode på to måneder utført et forsøk for å se om det er mulig å fjerne loggerne som er plassert ved værstasjoner rundt i landet. Loggerne gjør at om noe skjer med tilkoblingen til værstasjonene, vil alle data likevel bli lagret på stedet, og man mister ingen målinger. Målet var å finne ut om man kan fjerne loggeren ved sensoren, overføre verdiene trådløst og likevel opprettholde høy oppetid. Ved å benytte forskjellige programmeringspråk som blant annet Python, ble det laget et program som hentet inn data direkte til en server. Forsøket viser at det er fullt mulig å erstatte loggermetoden som blir brukt i dag til en loggermetode, som logger eksternt på en database. Med programmet som ble brukt i forsøket, ble resultatet en total nedetid på 11,2 %, men en forventet nedetid på 4,3 %. Dette er innenfor det vi regner som akseptabelt.

INNLEDNING

Når Meteorologisk Institutt foretar målinger for å logge temperaturer rundt i Norge, benytter de seg av en temperatursensitiv motstand som registrerer hvor stor resistans, R [Ω], det er i en krets. Denne resistansen blir så konvertert til en temperaturverdi i en logger som er fysisk koblet opp mot sensoren ved hver målestasjon. Dette viser seg å være en effektiv, men nokså kostbar løsning ettersom hver enkelt logger koster i overkant av 30.000 kroner per enhet. Årsaken til at denne metoden har blitt benyttet, er fordi den er pålitelig og har en relativt lav nedetid (nedetid – at det ikke er kontakt mellom programmet og ruterer) slik at det er svært sjelden at en måling ikke blir hentet.

I dette forskningsprosjektet undersøkes det om det er mulig at den nåværende metoden for temperaturlogging kan erstattes med et program som henter informasjonen fra en måler ved hjelp av en servertilkobling mellom datamaskin og sensor. Derfor er forskningsspørsmålet vårt: «Kan et system som sender rådata til en ekstern datamaskin som konverterer verdiene, ha like nøyaktige målinger og ha lite nedetid

i forhold til det systemet som blir benyttet i dag?» I dette forskningsprosjektet blir det tatt utgangspunkt i en temperaturmåler som er satt opp ved Meteorologisk Institutt ved Blindern i Oslo. Sensoren er koblet opp mot en 3G ruter som gjør det mulig å kommunisere med systemet uten å være koblet til systemet. På denne måten kan en sende data over nett slik at disse mottas av en ekstern datamaskin, slik vi har gjort i dette forsøket. For at løsningen skal regnes som pålitelig må det være en total opptid (opptid – at det er kontakt mellom programmet og ruter) på minimum 85 %. For at målingene kan bli kalt nøyaktige nok må 95 % av målingene være ± 1 °C i forhold til de målinger hentet fra *yr.no* (Meteorologisk Institutt, 2017). Målingene fra *yr.no* brukes som referanse i forsøket.

TEORI

PT100 sensor (resistans termometer) blir brukt for å måle temperaturer (Meteorologisk Institutt). Sensoren bruker metallet platina for å måle resistans. Ved 0 °C er resistansen i sensoren 100 ohm Platina har en motstandsfaktor som gjør at resistansen øker ved temperaturøkning (SKS Sensors, u.d.). En av de største forskjellene på denne sensoren fra andre, er at den er veldig stabil over lengere perioder. Sensoren er en «Klasse B 1/10 DIN» og har et temperatursprang på -196°C → +600°C og en usikkerhet på 0°C = $\pm 0.3/10$ °C (SKS Sensors, u.d.). Ohm-verdi fra ruter regnes om med *Formel 1* for å få resultatene i grader C.

Da bruker man formelen:

$$t = \frac{-R_0 \cdot a + \sqrt{R_0^2 \cdot a^2 - 4 \cdot R_0 \cdot b \cdot (R_0 - R_t)}}{2 \cdot R_0 \cdot b}$$

Formel 1: Konverterer en resistans R til en temperatur (Köhler, 2016)

hvor konstantene:

$$a = 3,9083 \cdot 10^{-3}$$

$$B = -5,7750 \cdot 10^{-7}$$

R_0 = Resistansen til sensoren ved 0°C

R_t = Resistansen til sensoren ved et gitt tidspunkt

METODE

Det ble benyttet tre ulike programmeringsspråk, tre markeringsspråk og socket-teknologi, til å utvikle en server og en nettside. Hovedprogrammet åpnes fra en batch-fil. Denne filen håndterer også eventuelle feilmeldinger og feil i målinger som gjør at selve loggingsprogrammet stopper. Programmet som startes av batch-filen er et Python script som åpner og kobler seg opp til ruter som står i målerboksen med en socket-tilkobling. Et signal "#04" blir sendt til ruter via programmet og åpner for at vi får verdien av PT100 sensoren (en motstandsverdi).

Denne verdien blir tatt imot av samme program som sendte "#04" signalet. I programmet blir også verdien tatt inn i en funksjon som tar motstandsverdien og regner ut verdien i grader celsius ved hjelp av *Formel 1*. Dette blir videre brukt til å regne ut et minuttmiddel.

Programmet sender så minuttmiddelet som en desimalverdi til en ekstern database. Databasen lagrer en id for målingen og legger ved et tidsstempel.

Nå som dataene er lagret i denne eksterne databasen, kan programmet starte på nytt for å hente neste verdi. Noen ganger kan man oppleve feilmeldinger som gjør at programmet stanser. Derfor er det lagt inn en batch fil som en start-fil som sjekker om scriptet kjører. Dersom scriptet ikke kjører, vil den starte programmet på nytt.

For å hente de data som er lagret i databasen til en *nettside*, legges det ved et PHP script med en linje SQL som henter verdiene (Codespot.no, 2017). Disse verdiene blir formatert inn i et HTML dokument som gjør at man ryddig kan studere og analysere dataene uten å måtte logge seg inn på selve databasen.

Målet er å ikke ha for mye nedetid, slik at data kan sammenlignes med den løsningen som allerede finnes hos meteorologisk institutt. Måten vi regner nedetiden på med de verdiene vi får inn, er ved å bruke ID'en til hver enkelt måling i løpet av en dag og addere disse sammen. I løpet av en dag vet man at det er 1440 minutter. Ved å telle antall mottatte målinger finner vi prosentandel nedetid.

RESULTATER

Det ble samlet inn data over en periode på syv dager. Før programmet ble oppdatert var det opp imot 70 % nedetid, etter oppdateringen er det er total nedetid på 11,2 %. Av de 11,2 % er 6,9 prosentpoeng i det området på *Figur 1* som er indikert med en pil. (Se forklaring på hvorfor vi ignorerer verdiene fra yr.no og avbruddet i diskusjonen). Det betyr at man kan forvente en nedetid på 4,3 % med oppdatert program. Dette er langt under målet som vi satt til 15 % før forsøket.



Figur 1: Viser sammenheng mellom temperaturresultatene i forsøket sammenliknet med verdiene fra yr.no

Figur 1 viser resultatene fra temperaturene vi samlet inn over perioden på syv dager. Den røde pilen peker på et område hvor det var nedetid fra t.o.m. 17.12.2017 kl.03:00 til t.o.m. 17.12.2017 kl.15:00. Dermed er også målingene fra yr.no fjernet for denne perioden i forsøket.

Tabell 1: Viser sammenligning av temperaturmålinger fra yr.no og våre egne målinger per time kl 09:00 – kl 15:00, 13. desember 2017 (Codespot.no, 2017), (Meteorologisk Institutt, 2017)

Klokkeslett	Målt temperatur	Temperatur fra yr.no
09:00	-7,39 °C	-7,3 °C
10:00	-7,09 °C	-7,1 °C
11:00	-6,67 °C	-6,8 °C
12:00	-5,93 °C	-6,0 °C
13:00	-5,25 °C	-5,5 °C
14:00	-3,91 °C	-4,1 °C
15:00	-2,78 °C	-2,8 °C

DISKUSJON

Det største avbruddet vi hadde, er det som er avmerket på *Figur 1* med en pil. Her hadde vi et avbrudd på omtrent 12 timer. En mulig forklaring på dette avbruddet kan komme av at dette skjedde før vi oppdaterte programmet. Da måtte programmet startes på nytt manuelt hver gang noe gikk galt. Dette var ikke mulig siden det ikke befant seg personell i nærheten. Vi har valgt å se bort ifra dette avbruddet, på grunn av at med oppdatert program får vi ikke denne typen avbrudd. Programmet vil starte på nytt av seg selv hver gang noe går galt.

Grunnet at vi mistet data tilsvarende 12 timer, valgte vi å fjerne tilsvarende målinger fra yr.no. Dette gjør at grafene kan sammenliknes. Se *Figur 1*.

Nedetiden vi registrerte i forsøksperioden var på 11,2 %. Vi kom frem til at av de 11,2 % er 6,9 % unødvendig, fordi etter oppdatert program vil det ikke forekomme slike avbrudd. Vi kan dermed anta 4,3 % nedetid. Derfor vil vi anbefale for meteorologisk institutt å bytte ut loggeren på mindre værstasjoner, med en videreutvikling av vårt program. På de større stasjonene vil vi anbefale at de fremdeles har en logger, ettersom med vår metode må man regne med nedetid.

Metoden som ble brukt i forsøket har virket godt. Det eneste som ikke gikk så godt som vi hadde håpet, var at det ikke ble samlet inn så mye data som ønsket. Om vi hadde hatt lenger tid på oss, hadde nok ikke dette vært ett problem. Temperaturene fra forsøket er ikke helt like med yr.no sine. Dette kan være grunnet at sensorene sto et stykke fra hverandre. En mulig måte å forhindre dette på, er å plassere sensorene nærmere hverandre.

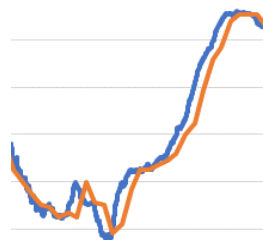
Det at programmet ikke samlet inn data med et intervall på nøyaktig 60 s hver gang, kan ha ført til forskjellene i temperatur. En annen faktor som også påvirket resultatene i *Figur 1*, er at man ikke fant ut hvordan yr.no regnet ut resultatene sine. Dermed kan de ha blitt regnet ut på en annen måte i forsøket, noe som kan ha ført til andre resultater.

Programmet ble utviklet over kun et par uker og er dermed ikke helt feilfritt. Dette medfører at det blir nedetid, og temperaturene blir kanskje ikke målt til riktig tid. Under all nedetiden vil man ikke motta noe data i det hele tatt. Dermed er kanskje ikke programmet som ble utviklet i dette forsøket, den beste løsningen for

å bli brukt kommersielt. Man kan eventuelt gå inn for å sette et budsjett for utvikling av et program over en tidsperiode på ett til to år. Da vil man kunne lage et mye mer sofistikert program.

Forsøket viser at det absolutt er mulig å lage et system som gjør det mulig å lagre dataene i en database uten bruk av logger. Konsekvensene av å ikke ha loggeren kan bli veldig store, om noe som helst skjer med tilkoblingen. Så man burde nok programmere inn koder som gjør hele systemet sikrere.

Figur 1 viser at temperaturene vi har hentet inn, stemmer godt med yr.no sine verdier. Forskyvingen av temperaturen i Figur 2 kan skyldes at det er nedetid som ikke har blitt registrert og tidfestet, og dermed er det for få målinger. Dette gjør at grafen blir kortere og dermed ikke passer med yr.no sin graf. Det at den retter seg opp igjen kan være grunnet at det ble tatt for mange målinger en stund. En annen mulig forklaring på forskyvningen kan være at sensorene stod et stykke unna hverandre, og dermed kan for eksempel et vindkast ha påvirket bare en sensor.



Figur 2: Viser et utsnitt av Figur 1 hvor temperaturene ikke samsvarer med hverandre

Ruteren vi brukte i forsøket, er en 3G-ruter. 3G-rutere er treige, dermed kan dette være grunnen til mange av problemene våre. Dataen bruker for lang tid på å bli sendt, dette forvirrer programmet. Det kan gå så langt som å få programmet til å krasje.

Grunnet at koden styres av en datamaskin, er det ikke mange mulige usikkerheter. Den ene usikkerheten vi kan tenke oss, er usikkerheten i sensoren, som kan ha gitt oss feil verdier. Det er ikke mulig å regne ut en gitt usikkerhet, ettersom sensoren har forskjellig usikkerhet ved forskjellige temperaturer. Vi kan si at usikkerheten er omtrent $0,03\text{ }^{\circ}\text{C}$, fordi vi vet at ved $0\text{ }^{\circ}\text{C}$ har usikkerheten dette verdien.

I tabell 1 ser vi sammenheng mellom temperatur som vi målte, og temperaturen som ble målt av Meteorologisk institutt. Vi ser at det ikke er betydelig forskjell mellom våre og deres resultater. Derfor har vi konkludert med at resultatene er brukbare.

Usikkerhet på oppe- og nedetid er noe vi ikke regner som relevant. Dette kommer av at målingene ble gjort av et program som regner ut nøyaktig hvor mange prosent av forventede målinger som ble registrert. Det er ikke en usikkerhet hvor mange målinger som mangler, da dette er et bestemt antall målinger i systemet.

KONKLUSJON

Våre forsøk viser at temperaturen vi målte stemmer godt overens med de offentlige verdiene til Meteorologisk Institutt. Nedetiden vi registrerte i forsøksperioden var på 11,2 %. Vi kom frem til at av de 11,2 % kan man se bort fra 6,9 prosentpoeng om man tar i bruk sist oppdatert program. Dette fordi etter oppdatert program vil det ikke forekomme slike avbrudd. Vi kan dermed konkludere med totalt 11,2 % nedetid, men forvente en nedetid på 4,3 % med oppdatert program. Derfor vil vi anbefale for meteorologisk institutt å bytte ut loggeren på mindre værstasjoner med en videreutvikling av vårt program. På de større stasjonene vil vi anbefale fremdeles å ha en logger, ettersom med vår metode må man regne med nedetid.

BIBLIOGRAFI

Codespot.no. (2017, desember). Codespot.no. Hentet fra Tabeller: <http://codespot.no/met/tabeller.php>

- Harrison. (2014, August 23). *YouTube.com*. Hentet fra Python 3 Programming Tutorial - Sockets: client server system: <https://www.youtube.com/watch?v=WrtEbUkUssc&list=PLQVvvaaoQuDe8XSftW-RAxdo6OmaeL85M&index=59>
- Hellmann, D. (2017, April 30). *PyMOTW.com*. Hentet fra TCP/IP Client and Server: <https://pymotw.com/2/socket/tcp.html>
- Köhler, U. (2016, Januar 02). *techoverflow.net*. Hentet fra Accurate calculation of PT100/PT1000 temperature from resistance: <https://techoverflow.net/2016/01/02/accurate-calculation-of-pt100pt1000-temperature-from-resistance/>
- Meteorologisk Institutt. (2017, November 01). *Yr.no*. Hentet fra Værvarsel for Blindern, Oslo, Detaljert varsel: https://www.yr.no/sted/Norge/Oslo/Oslo/Blindern/time_for_time_detaljert.html
- Meteorologisk Institutt. (2017). *Yr.no*. Hentet fra Datosøk, Blindern, Oslo: <https://www.yr.no/sted/Norge/Oslo/Oslo/Blindern/almanakk.html?>
- Meteorologisk Institutt. (u.d.). *Værstasjoner i norske kommuner*. Meteorologisk Institutt. Hentet fra https://cms.met.no/site/2/klimaservicesenteret/dimensjonerende-nedb%C3%B8r/hjelp-til-%C3%A5-etablere-v%C3%A6rstasjon/_attachment/7107?_ts=150a498c4cc
- SKS Sensors. (u.d.). *skssensors.com*. Hentet fra How does a Pt100 sensor work?: <http://www.skssensors.com/en/faq/how-do-pt100-sensors-work/>