



Mikael Bjong

Dronen säädätin mittaus- ja lähetyjärjestelmä sekä jäätäminen ilmiönä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

17.8.2021

Tiivistelmä

Tekijä:	Mikael Bjong
Otsikko:	Dronen säädatan mittaus- ja lähetysjärjestelmä sekä jäätäminen ilmiönä
Sivumäärä:	31 sivua
Aika:	17.8.2021
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine:	Automaatiotekniikka
Ohjaaja:	Yli-opettaja Erkki Räsänen

Insinööritöön tarkoituksena oli tuottaa dronen mittausdatan keräyksen tiedonsiirron toteutus. Lisäksi tutkittiin jäätämisen vaikutuksia ja tunnistettavuutta dronen telemetriasta. Mittausdatan keruun ja siirron tavoitteena oli saada mitattua säädataa ympäristöstä ja lähettää sitä Ilmatieteen laitoksen kokeelliseen rajapintaan. Tavoitteena oli myös tutkia, onko dronen jäätäminen tunnistettavissa telemetriasta ja mahdollisesti kehittää koneoppimismalli jäätämisen itsenäiseen tunnistamiseen.

Työssä kaiken pohjana käytettiin Particle Electron -IoT-kehitysalustaa, johon liitettiin BME280-barometri ympäristön mittaamiseen sekä IR-moduuli jäätämisen havainnoimiseen. Ympäristön säädatan lisäksi Electron keräsi paikkatietoa käyttäen Googlen Geolocator-integraatiota. Mittaukset koottiin rajapinnan käyttämään JSON-muotoon ja lähetettiin onnistuneesti. Electron oli varustettu 3G-yhteydellä sekä Li-Po-akulla, joka mahdollisti täyden riippumattomuuden ympäristöstä tai aluksesta.

Jäätämisen vaikutus telemetriaan toteutettiin ainoastaan teorian tasolla dronen vikaantumisen vuoksi. Lähteiden mukaan olennaisimmat vaikutukset telemetriaan ovat nosteen aleneminen sekä virrankulutuksen kasvu. Aikaisemmasta testilennon lokista havaittiin mahdollista värinää sekä hankaluuksia päästä dronella tavoitekorkeuteen. Jäätämisen havaitsemiseen kehitettiin optinen ratkaisu. IR-moduulia hyödyntäen tarkkailtiin IR-LEDin heijastavuutta, johon jäätävät olosuhteet vaikuttavat.

Avainsanat: drone, JSON, REST, API, jäätäminen

Abstract

Author:	Mikael Bjong
Title:	Drone Weather Data Measurement and Transmission System and Icing as a Phenomenon
Number of Pages:	31 pages
Date:	17 August 2021
Degree:	Bachelor of Engineering
Degree Programme:	Electrical and Automation Engineering
Professional Major:	Automation Engineering
Instructor:	Erkki Räsänen, Principal Lecturer

The purpose of this thesis work was to produce a data measurement and transfer implementation for drones, and to study how icing affects the drone's telemetry and whether it is reliably recognizable. Another objective was also to potentially develop a machine learning application for independent recognition of icing. The implementation's goal was to measure weather data and send it to FMI's experimental API.

The basis of this thesis work was Particle Electron, which is an IoT-development kit. Electron was connected to a BME280 barometer to measure the weather data. Also, an IR-module was connected to detect icing. Electron also tracked its location using Google Geolocator integration. This location data was combined with weather data and embedded into JSON-format. This was sent to the API successfully. The Electron included 3G cellular connection and Li-Po battery, which made it totally self-sufficient unit and not dependent on the environment, nor on the vehicle using it.

The effects of icing to telemetry were addressed only on a theoretical level. This is due to the drone's technical failure. According to the sources, the most significant effects of icing are decrease in lift and increase in power consumption. An earlier test flight log was analyzed. There was possible vibration detected and difference between altitude and desired altitude, which can be a result of icing. An optical application was made to detect icing. The IR-module was utilized to detect icing conditions by monitoring IR-LED's reflection intensity.

Keywords: Drone, JSON, REST, API, Icing.

Sisällys

Lyhenteet

1	Johdanto	1
2	Älykäs teknologia	2
2.1	Esineiden internet (IoT)	3
2.2	5G-teknologia	4
3	Verkkotekniikat	5
3.1	API	5
3.2	REST	5
3.3	JSON	6
4	Dronet	7
5	Jäätäminen ilmiönä	8
5.1	Vaikuttavat tekijät	8
5.2	Ympäristöolosuhteet	8
5.3	Vaikutus telemetriaan	9
6	Mittaus- ja lähetysyksikkö	9
6.1	Anturointi - Seeed Grove BME280 -barometri	10
6.2	Particle Electron	11
6.3	IR-moduuli	12
7	Datan lähetys	14
7.1	Publish-Subscribe	14
7.2	Google Geolocator API	14
7.3	Particle ja Geolocator-integraatio	15
7.4	Rajapinta	16
7.5	Particle ja FMI-integraatio	16
8	Particle-koodi	18
8.1	Määrittelyt	18
8.2	Setup	19

8.3	Pääohjelma	20
9	Jäätämisen havainnointi	22
9.1	Telemetry	23
9.2	Käytännön sovellus	24
9.3	Jatkokehitysidea	26
10	Yhteenveto	27
10.1	Tavoitteet ja tulokset	27
10.2	Kehitysideat ja pohdinta	28
	Lähteet	30

Lyhenteet

- API: Application Programming Interface. Ohjelmointirajapinta, jonka tehtävä on saada eri arkkitehtuureilla tai ohjelmointikielillä rakennetut järjestelmät kommunikoimaan keskenään.
- Firmware: Laitteen perustoimintoja ohjaava laiteohjelmisto.
- IoT: Internet of Things. Esineiden Internet, jossa ihmisten lisäksi esineet ovat verkon avulla yhteydessä toisiinsa.
- IR-LED: Infrapunavaloa säteilevä diodi.
- JSON: JavaScript Object Notation. JavaScriptiin pohjautuva tiedonsiirtomuoto.
- Pub-Sub: Publish-Subscribe. Viestinnän malli, jossa serveri toimii välikätenä julkaisijoille ja tilaajille.
- REST: Representational State Transfer. Tietoverkkojen kautta toisiinsa yhdistettäville sovelluksille tarkoitettu arkkitehtuurimalli
- UAV: Unmanned Aerial Vehicle. Miehittämätön ilma-alus.
- Webhook: Tapahtumalla aktivoitu HTTP-verkkopyyntö.

1 Johdanto

Tämän insinöörityön tarkoituksena oli tutkia ja toteuttaa UAV:n (Unmanned Aerial Vehicle, miehittämättömän ilma-aluksen) lento-olosuhteiden mittausdatan siirtoa sekä analysoida, että voiko ilma-alus havaita telemetriasta laitteen jäätämistä. Työn pohjana käytetty UAV oli Holybro x500 -drone. Aluksen liikkeitä ohjaava lennonohjain oli Pixhawk 4, joka perustuu avoimeen lähdekoodiin.

Projekti on aloitettu alun perin syksyllä 2020 osana DigiSalama-hanketta. Hankkeessa oli mukana kaksi ryhmää. IT-ryhmä hoiti tietoliikenteeseen ja datan käsittelyyn liittyvät asiat ja koneautomaatioryhmä dronen suunnittelun, kokoamisen, anturien asennuksen ja testilennot. Insinöörityö on tuotettu toimijoille Insta ja Ilmatieteen laitos.

Ilmatieteen laitos on liikenne- ja viestintäministeriön tutkimuslaitos. Ilmatieteen laitos on perustettu 1838 ja toiminut nykyisellä nimellä vuodesta 1968 lähtien. Sen tehtävänä on tuottaa havainto- ja tutkimustietoa ilmakehästä, lähiavaruudesta ja meristä. Tieto tuotetaan yleisen turvallisuuden, liikenteen, elinkeinoelämän ja kansalaisten tarpeisiin. [1; 2.]

Insta on vuonna 1960 perustettu konserni, joka tarjoaa palveluita teknologian haastavilla sektoreilla. Insta on jaettu viiteen toimialaan:

Sähköautomaatio (Insta Automation Oy).

Tietojärjestelmät ja kyberturvallisuus (Insta DefSec Oy).

Avioniikka, miehittämätön ilmailu ja korkeateknologia (Insta ILS Oy).

Data-analytiikka, liikkuvat laitteet ja koneiden digitalisaatio (Insta Digital Oy).

Kyber-, tietoturva- ja tietosuojakonsultointi (Insta Secrays Oy). [3; 4.]

Ilmatieteen laitos oli kiinnostunut mahdollisuuksista lähettää mittausdataa kokeelliseen rajapintaansa. Tarkoitukseni oli viedä projekti loppuun kehittämällä tiedonsiirron malli. Insinööriyössä tutkittiin datansiirron mahdollisuuksia niiden tarjoamaan rajapintaan joko lentojen jälkeen offline-tilassa tai reaaliaikaisesti lennossa. Insta puolestaan oli kiinnostunut tavoista, joilla pystyttäisiin telemetriasta päättämään jäätämisen ilmiötä. Tavoitteena oli tutkia, onko dronen mahdollista itsenäisesti tiedostaa ja varoittaa jäätämisestä saatuaan riittävästi dataa.

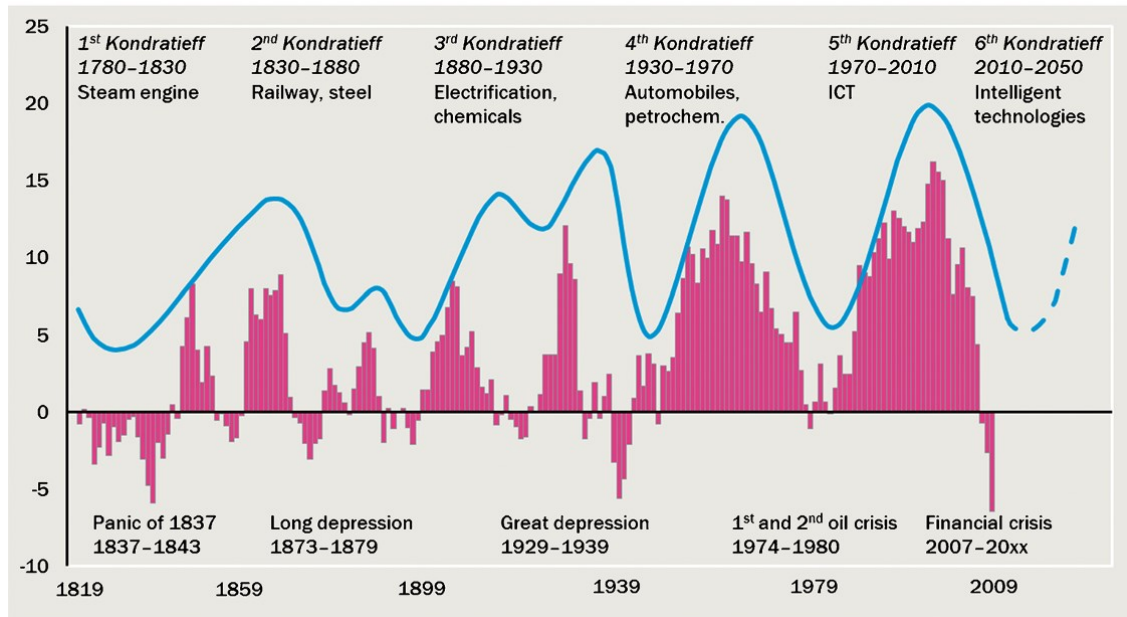
Insinööriyön aihe on tärkeä ja ajankohtainen. Teknologian kehittyessä UAV:n tarjoamia mahdollisuuksia tutkitaan ja hyödynnetään entistä enemmän. Alusten käytön laajentuminen entisestään tuo esille käytännön kysymyksiä ja haasteita, joihin toivon tämän työn vastaavan.

2 Älykäs teknologia

Venäläinen taloustieteilijä Nikolai Kondratjev esitti aaltoteorian talouden kehittymisestä 1930-luvulla. Kondratjevin mukaan moderni talous kehittyy 40–60 vuoden sykleissä. Näitä syklejä kutsutaan Kondratjevin aalloiksi. Jokainen aalto alkaa teknologisilla innovaatioilla, jotka vaikuttavat läpi talouden ja yhteiskuntajärjestelmän. (Kuva 1.) Teorian mukaan elämme nyt 5. ja 6. aallon murroksessa. Jokaiseen aaltoon sisältyy seuraavat asiat:

- Uudet teollisuudet muodostuvat, korvaten vanhat.
- Uusi laajentunut talouskasvu osakemarkkinoiden kasvaessa.
- Uudet arvojärjestelmät alkavat vallita, hallitsemaan julkista keskustelua ja suunnittelua.
- Uudet ammatit ja taitojen standardit muodostuvat.
- Uudet yhtiökulttuurit alkavat hallita. [5, s. 1–3.]

Modern economies fluctuate in a cycle of 40–50 years
 Rolling 10-year yields of the Standard & Poors 500 equity index and
 the Kondratieff's waves



Kuva 1. Talouden tuottavuuden vaihtelu Kondratjevin aaltojen mukaisesti 40–50 vuoden sykleissä [5].

Älykäs teknologia pitää sisällään laajan skaalan ilmenemismuotoja.

Seurauksena on älykkään teknologian entistä syvempi integraatio arkielämään ja käyttämiimme palveluihin. Ilmenemismuodot ovat sulautuneet toisiinsa muodostaen älykästä teknologiaa. Esimerkiksi dronen täyden potentiaalin hyödyntäminen tarvitsee propellien lisäksi suuren määrän sensoreita ja verkkotekniikkaa. Oleellisena asiana itsensä ja ympäristön mittaus ja niiden muodostaman datan siirto sekä käsittely.

2.1 Esineiden internet (IoT)

Esineiden internetille eli IoT:lle ei ole virallista määritelmää. Se on monimuotoinen ja laaja ilmiö, jonka kuvaus riippuu hieman näkökulmasta. Yleinen määritelmä käsittää ajatuksen, että maailmamme on jatkuvasti enemmän yhteydessä ja verkossa. Olemme ajassa, jossa eivät vain ihmiset ole yhteydessä toisiinsa vaan myös esineet, luoden esineiden internetin. [6, s. 29.]

IoT nähdään perinteisen internetin loogisena seuraajana ja sen laajentumisena fyysiseen maailmaan. Tekemämme asiat määrittäisivät enemmän datan perusteella kuin tapojen tai kokeilun seurauksena. Lähes kaikissa esineissä olisi erilaisia sensoreita, joiden valtava määrä dataa analysoitaisiin, mikä johtaisi älykkäämpiin ratkaisuihin. [7, s. 4 ja 9]

Tietoturva on yksi suurista haasteita IoT:n laajentuessa. Verkkorikolliset hyökkäävät entistä enemmän serverien sijaan päätelaitteisiin. IoT-laitteet ovat todennäköisesti huomattavasti heikommin suojattuja kuin serverit. Laitteet sijaitsevat kuluttajaa fyysisesti lähellä aiheuttaen suuren riskin arvokkaan tiedon vuotamiselle. Tietoturvan ohella suuri haaste on yksityisyys. Laitteet on varustettu lukuisilla sensoreilla, jotka tarkkailevat käyttäjien henkilökohtaisia parametreja. [8, 1.8 ja 1.10]

2.2 5G-teknologia

Edellisen sukupolven 4G-verkkotekniikka alkaa olla elinkaarensa päässä älykkään teknologian näkökulmasta. 4G:n rajoitteet tulevat ilmi laitteiden määrän ja suurten datasiirron tarpeiden myötä. Älykkään teknologian uudet muodot, kuten IoT, verkkopohjaiset tekoälysovellukset ja autonomiset ajoneuvot, luovat väistämättä tarpeen 5G-teknologialle. Uuden sukupolven tiedonsiirto tarjoaa useita parannuksia verrattuna 4G-teknologiaan, tärkeimpinä mm.

- korkeataajuista kaistaa käyttämällä 10 kertaa nopeampi tiedonsiirron huippunopeus
- 4–5 kertaa matalampi viive
- 90 % parempi energiatehokkuus. [9.]

5G:n mahdollisuuksia on hyödynnetty muun muassa seuraavassa drone-kokeilussa. Nokia, Nordkapp, Suomen ympäristökeskus, Telia ja Vaisala kehittivät kokeilun, jossa tutkittiin sinilevän levinneisyyttä Itämeressä hyödyntämällä konenäköä. Aikaisemmin sinilevätilannetta on kartoitettu useista

lähteistä. Kokeilussa drone lensi laajalla alueella näkökentän ulottumattomissa ja data siirrettiin reaaliajassa konenäölle. Hyvissä olosuhteissa konenäkö havaitsi sinilevän jopa yli 90 %:n tarkkuudella. 5G:n mahdollistama korkealaatuinen videokuva, nopeat yhteydet ja reaaliaikainen tekoälyn hyödyntäminen tarjoaa uusia mahdollisuuksia ympäristön tutkimiseen. Muun muassa ympäristötuhojen paikannus on toteutettavissa tehokkaasti. [10]

3 Verkkotekniikat

3.1 API

API (Application Programming Interface) tarkoittaa suomeksi ohjelmointirajapintaa. Sen tehtävä on saada eri arkkitehtuureilla tai ohjelmointikielillä rakennetut järjestelmät kykenemään kommunikoidaan keskenään. Käytännön esimerkki olisi käyttöjärjestelmän rajapinta, jota hyödyntämällä ohjelmat kykenevät käyttämään keskusmuistia ja tiedostoja. [11]

3.2 REST

REST (Representational State Transfer) on tietoverkkojen kautta toisiinsa yhdistettäville sovelluksille tarkoitettu arkkitehtuurimalli. Ideana on web-protokollien hallitsemiseen tarkoitettu arkkitehtuurimalli. REST-sovellusten välillä käytetään yleensä HTTP-protokollaa. [11]

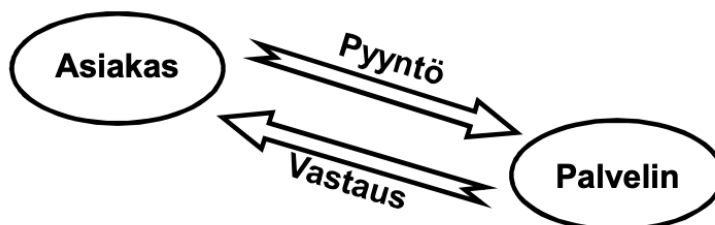
RESTin tavoitteena on muun muassa

- vähentää latenssia ja verkkoviivettä
- tehostaa skaalautuvuutta ja komponenttien itsenäistä kehittyvyyttä
- parantaa rajapintojen yksinkertaisuutta, siirrettävyyttä ja luotettavuutta.

[11;12]

RESTin rakenne koostuu asiakkaasta ja palvelimesta. Kuvan 2 mukaisesti palvelin vastaa asiakassovelluksen lähettämien viestien vastaanottamisesta ja

käsittelystä. Asiakkaan vastuulle jää viestien lähettäminen oikeassa muodossa ja vastausten käsittely tarpeiden vaatimalla tavalla. [11]



Kuva 2. RESTin perusrakenne. Asiakas saa aina vastauksen palvelimelta pyynnöstä huolimatta. [11]

3.3 JSON

JavaScript Object Notation (JSON) on kevyt, JavaScriptiin pohjautuva tiedostomuoto. JSON (kuva 3) on tavallaan tiedostona kuin sanakirja, joka muodostaa avain-arvopareja, jossa arvo voi myös olla JSON-dokumentti. Sanakirjasta poiketen JSON sallii myös taulukoiden, kokonaislukujen ja merkkijonojen käytön. [13.]

```

{
  "nimi": {
    "etunimi": "Mikael",
    "sukunimi": "Bjong"
  },
  "ikä": 24,
  "harrastukset": ["painonnosto", "kutominen"]
}
  
```

Kuva 3. Yksinkertainen JSON-rakenne havainnollistettuna.

JSON on helposti luettavissa ihmisen tai koneen toimesta. Yksinkertaisen rakenteensa ansiosta siitä on kehittynyt yksi suosituimmista formaateista datanvaihtoon verkkopalveluissa. Verkkopalvelut kommunikoivat käyttäjiensä

kanssa API-pyyntöjä ja vastauksia JSONia hyödyntäen HTTP-protokollan yli. [13.]

4 Dronet

UAV (Unmanned Aerial Vehicle) tarkoittaa miehittämätöntä ilma-alusta. Esimerkiksi, drone on lentävä robotti, jota ei ole miehitetty lennon aikana. Dronet lentävät etäohjauksella tai itsenäisesti ja kantavat yleensä kuormaa. Dronejen kehitys navigaatiossa, etäohjauksessa ja energian varastoinnissa on mahdollistanut laajan skaalan erilaisia droneja eri käyttökohteisiin. Tarpeen näille sovelluksille on luonut ihmisen paikallaolon vaikeus, mahdottomuus, vaarallisuus tai tarpeettomuus. Dronejen käyttö sotilasoperaatioissa, planeettojen tutkimuksissa ja pelastustehtävissä ovat lähivuosina herättäneet eniten huomiota. [14.]

Alun perin dronet olivat lähinnä lentoharrastajien ja tiettyjen sotilasvoimien käytössä. Ajan kuluessa uudet yritysideal, tutkimukset ja käyttökohteet ovat laajentuneet monelle sektorille. Esimerkiksi maatalous-, rakennus-, elokuva-, infrastruktuuri-, kuljetus- ja kaukokartoitusalat hyödyntävät jo drone-teknologiaa. Myös UAV:n mahdollisuuksia urbaaniin liikkumiseen tutkitaan. Tavaroiden kuljetussovellukset ovat herättäneet paljon huomiota ja mielenkiintoa. Tämä todennäköisesti tulee laajentamaan dronet ensimmäisenä laajaan kuluttajakäyttöön. [15, s. 1.]

Suurimmat kuluttajien käyttöön suunnatut projektit ovat muun muassa Googlen Wing- ja Amazon Air -palvelut. Arvioidaan, että matalalla lentävät miehittämättömien alusten lentojen määrä syrjäyttää miehitetyt lennot vuoteen 2035 mennessä. [16, s. 1.]

5 Jäätäminen ilmiönä

5.1 Vaikuttavat tekijät

Jään muodostuminen on monimutkainen ilmiö, johon vaikuttavat monet tekijät. Opinnäytetyöni ei keskity jäätämiseen ilmiönä, joten vain ilmiön olennaisimmat tekijät ovat otettu huomioon.

Jään syntymiseen on tiettyjä vaatimuksia, joita ilman jäätämistä ei voi tapahtua, esimerkkinä lämpötila: mikäli ilman lämpötila on jäätymispisteen yläpuolella, muilla vaikuttavilla jäätämisen tekijöillä ei ole merkitystä. Tämä helpottaa päättelyä.

5.2 Ympäristöolosuhteet

Normaali ilmapaineen vallitessa puhdas vesi jäätyy 0 asteessa. Tästä voidaan poiketa, mikäli tapahtuu alijäähtymistä. Jäätymispisteen tavoittaminen on suurin tekijä UAV:n jäätämisessä. Alijäähtyneessä tilassa oleva vesi voi pysyä nestemäisenä, vaikka jäätymispiste olisi alitettu. [17.]

Ilmanpaine vaikuttaa jäätymispisteeseen. Normaali ilmanpaine maan pinnalla on 1013 hPa. Noustessa ilmakehässä ylöspäin vain 8 metrin nousu korkeussuunnassa merkitsee noin 1 hPa:n vähennystä ilmanpaineessa. Mitä suurempi ilmanpaine, sitä pienempi jäätymispiste. Opinnäytetyön kohteena oleva ilma-alus nousee vain EU-standardin määrittelemiін rajakorkeuksiin (avoin-toimintakategoriassa 120 m), joten ilmanpaineen muutos on marginaalinen. [18; 19 artikla 42.]

Anturin mittaama suhteellinen kosteus tarkoittaa prosenttia siitä vesimäärästä, joka kyseisessä lämpötilassa voi enimmillään olla [20].

5.3 Vaikutus telemetriaan

Telemetria on aluksen sisäistä dataa, joka koostuu moottorin, ohjauksen, asennon ja rungon parametreista. Jäätämällä on negatiivinen vaikutus lentämisen suorituskykyyn. Vaikutus vaihtelee lentoaluksen koon ja massan mukaan. Pienemmässä aluksessa jään määrä on suhteessa painoon ja suorituskykyyn isompi tekijä. Tiedetään, että seuraavat ilmiöt ovat avaintekijöitä jään vaikutusta tutkittaessa telemetriasta:

- Noste voi laskea jopa 70 prosenttia ja virrankulutus kasvaa 250 prosenttia.
- Jään muodostama epätasainen kerros häiritsee lavan aerodynamiikkaa aiheuttaen värähtelyä alukseen.
- Nosteen laskettua alus ei kykene nousemaan tavoitekorkeuteen.
- Aluksella voi olla vaikeuksia leijua samassa pisteessä, mikä on nykyaikaisilla droneilla yksi normaaleista toimenpiteistä. [15, s. 4.]

6 Mittaus- ja lähetysyksikkö

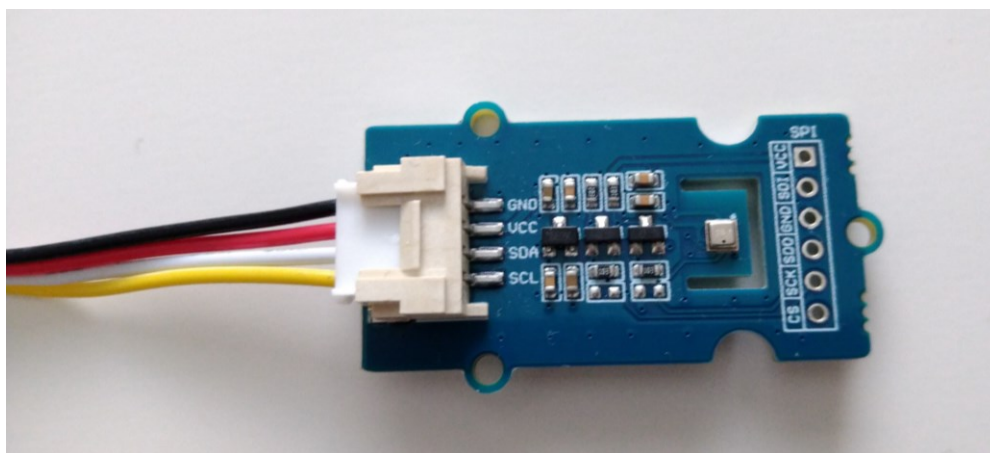
Ilmatieteen laitos oli kiinnostunut mittausdatan siirtämisestä ilma-aluksesta itselleen. Parhaimmassa tapauksessa alus mittaisi dataa ympärillä olevista olosuhteista ja lähettäisi niitä reaaliajassa.

Mittausdata koostui seuraavista parametreista:

- lämpötila
- ilmankosteus
- ilmanpaine
- latitudi
- longitudi
- korkeus merenpinnasta.

6.1 Anturointi - Seeed Grove BME280 -barometri

Parametrien mittaukseen valittiin Groven valmistama barometri (kuva 4). Kyseinen barometri pohjautuu Boschin BME280-sensoriin. Tämä mahdollistaa luotettavan ja tarkan mittausdatan vähäisellä virrankulutuksella. Barometri tarvitsee vähintään 3,3 voltin sisääntulojännitteen, jonka sai kätevästi liitettävästä kehitysalustasta.



Kuva 4. Seeed Grove BME280 -barometri.

Barometri hyödyntää Boschin sensoria parametrien mittaukseen. Barometri toimii Adafruitin BME280 -kirjaston kanssa pienillä muutoksilla. Kirjasto muuttaa sähköisen signaalin numeeriseksi arvoksi, joka jälleen sijoitetaan niitä vastaaviin muuttujiin: t,h,p ja a. Muuttujien arvot formatoidaan julkaistavaan muuttujaan merkkijonoksi, johon lisätään myös aikaleima. Sijainnin tilauksen jälkeen sijoitettiin parametrit "lat"- ja "lon"-muuttujiin. Tästä lisää 8. luvussa, jossa kerrotaan, kuinka barometri toimii ohjelman tasolla. Taulukossa 1 esitetty teknisiä tietoja barometrasta.

Taulukko 1. BME280:n tekniset tiedot.

Parametri	Arvo
Käyttöjännite	3,3 V
I/O-jännite	3,3 V
Virta käytössä	0,4 mA
Käyttölämpötila	-40–85 celsiusastetta
Ilmanpaineen mittausalue	300–1100 hPa (1 hPa = 100 Pa) ±1,0 hPa:n tarkkuudella
Ilmankosteuden mittausalue	Suhteellinen kosteus 0–100 % ±3 %:n tarkkuudella
Väylät	SPI, I2C (vakio)

6.2 Particle Electron

Reaaliaikainen datan lähetys vaatii luotettavan verkkoyhteyden. Aluksen on pystyttävä olemaan verkossa sijainnistaan riippumatta, joten Wi-Fi-pohjainen ratkaisu ei tullut kysymykseen. Mittausanturin on myös oltava liitettävissä lähettävään komponenttiin. Vaatimusten perusteella datan lähetykseen ja projektin keskeisimmäksi ympäristöksi valittiin Particle Electron.

Particle Electron (kuva 5) on kehitysalusta IoT-sovelluksiin. Particlen valmistama Electronin mukana toimitetaan SIM-kortti, joka tukee 3G-yhteyksiä. Hintaan kuuluu ilmainen datan käyttö kulukatolla.



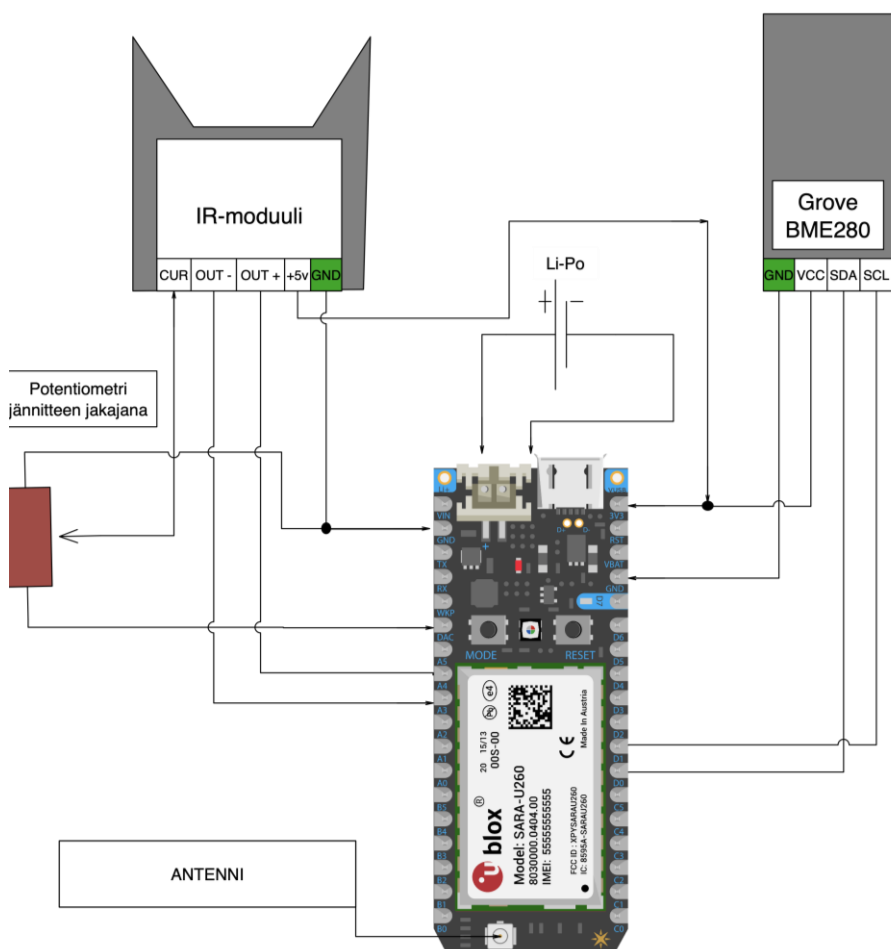
Kuva 5. Electron alkuperäisessä pakkauksessaan [21].

Particle tarjoaa selainpohjaisen WebIDE-ympäristön laitteen ohjelmointiin. Selainympäristössä on myös yhteisön ylläpitämä kirjasto, josta voi sisällyttää ohjelmia tai osioita omaan sovellukseen. Valmis sovellus voidaan päivittää laitteeseen suoraan selaimesta. Datat säästämiseksi Particlen tuotteita on mahdollista operoida suoraan käyttöjärjestelmän komentorivin kautta. Esimerkiksi varmiin sovelluksen firmware-tiedoston voi ladata selaimesta ja päivittää Electron USB-yhteyden avulla. Komentorivistä voi myös käyttää muita selainversion toimintoja; tosin käyttömukavuus kärsii.

6.3 IR-moduuli

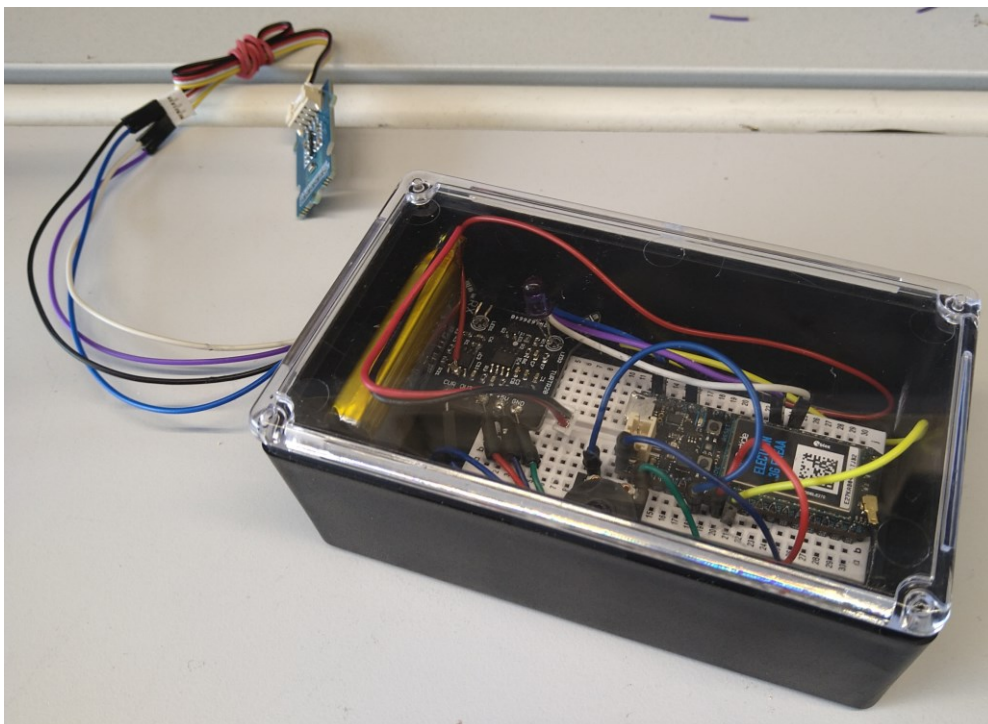
Jäätämisen visuaaliseen havainnointiin hyödynnettiin IR-moduulia. Moduuliin on kytketty IR-LED, jonka valoa vastaanotin havaitsee. Kuvassa 6 esitellään järjestelmän kytkentäkaavio. Käyttöjännite kytkettiin Particle Electronin 3,3 voltin ulostulojännitteeseen. LEDin valon voimakkuutta säädetään syöttämällä haluttu eksitaatiojännite CUR-nastaan. Eksitaatiojännite säädetään kytketyllä 10 kilo-

ohmin potentiometrillä, joka on kytketty jännitteen jakajaksi. Ulostulosignaali OUT- ja OUT+ kytketään Electronin analogisiin sisääntuloihin A3 ja A4.



Kuva 6. Järjestelmän kytkentäkaavio.

Kuvassa 7 valmis mittaus- ja lähetysyksikkö on liitettävissä mihin vain liikkuvaan alukseen.



Kuva 7. Valmis yksikkö.

7 Datan lähetys

Ilmatieteen laitos oli kiinnostunut datan lähetyksestä tietokantaansa. Haluttiin tutkia mahdollisuuksia liikkuvasta datan kerääjästä ja lähettäjistä. Olennaista lähetetyssä datassa olisi paikkatieto ja olosuhdeanturien mittaamat parametrit.

7.1 Publish-Subscribe

Publish-Subscribe on viestinnän malli, jossa data ei siirry suoraan lähettäjältä vastaanottajalle. Näistä toimijoista puhutaan termillä julkaisija ja tilaaja. Julkaisija lähettää datan serverille, tietämättä vastaanottajia. Datan toimitus tilataan halutessa serveriltä. Yleisesti käytetään lyhennettä "Pub-Sub".

7.2 Google Geolocator API

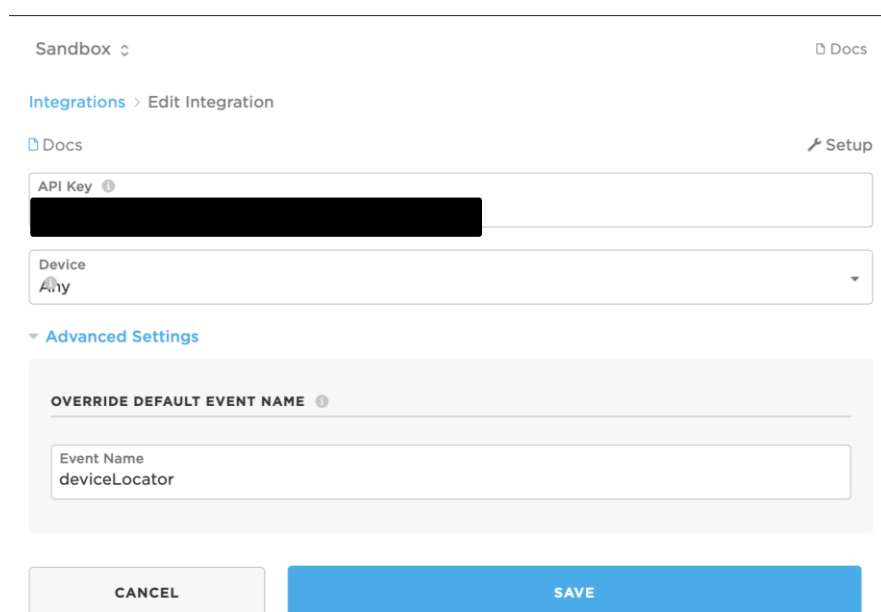
Sijainnin määrittämiseen käytettiin Googlen Geolocatoria Publish-Subscribe-menetelmällä. Google Geolocator API on Googlen tarjoama paikannukseen

käytettävä rajapinta. Se paikantaa tilaajan käyttäen liikkuvan laitteen löytäviä puhelinmastoja sekä Wi-Fi-solmuja. Jos laite ei kykene paikantamaan kumpaakaan edellistä, käytetään tilaajan IP-osoitteen paikannusta, joka vaikuttaa radikaalisti paikannustarkkuuteen. Kommunikaatio kulkee tilaajan ja julkaisijan välillä HTTPS-prokollan mukaisesti.

7.3 Particle ja Geolocator-integraatio

Googlen paikkannuksen tilaus tarvitsee toimiakseen Particlen Webhookin (kuva 8). Webhook on Particlen tarjoama tiedonvälitysmenetelmä, joka hyödyntää luvussa 7.1 mainittua Pub-Sub-mallia yhdistääkseen fyysiset laitteet digitaalisen verkon palveluihin.

Rajapinnan käyttö vaatii Google-tilin. Google Geolocator on otettava käyttöön Google-tilin asetuksista. Käyttöön saa henkilökohtaisen API-avaimen, joka syötetään integraatiota luodessa Particlen kotisivuilla.



Sandbox ▾ Docs

Integrations > Edit Integration

Docs Setup

API Key ⓘ

Device
Any ▾

Advanced Settings

Override Default Event Name ⓘ

Event Name
deviceLocator

CANCEL SAVE

Kuva 8. Google Geolocator-integraation lisääminen Particle Consolessa.

Integraation viimeinen vaihe oli liittää lähetyskoodiin Geolocator-kirjasto. Koodiin täytyy asettaa Subscribe-metodilla pyyntö paikannuksen saamiseksi.

Näin integraatio aktivoituu ja lähettää Googlelle paikannuspyynnön. Tämä vaihe viimeistellään loppuun koodin kirjoitusosiossa.

7.4 Rajapinta

Ilmatieteen laitos tarjosi kokeellisen REST-rajapintansa käyttöön. Lähetetyn datan täytyi olla JSON-muodossa ja sisältää tiettyjä ennalta määriteltyjä parametreja, kuten "station-id", "wgs84 -lat sekä lon" ja aikaleima. Näiden jälkeen "data"-taulukkoon voitiin sijoittaa halutut parametrit ja arvot, kunhan JSON-rakennetta ei rikota.

Mikäli lähetetty datapaketti rekisteröidään onnistuneesti, saadaan paluukoodi 200, joka tarkoittaa onnistunutta vastaanottamista. Mikäli rakenteessa oli pienikin virhe, saatiin paluukoodi "502". Tämä aiheutti pitkään ongelmia, kun joutui "sokkona" lähettämään dataa testivaiheessa olevaan rajapintaan.

7.5 Particle ja FMI-integraatio

Integraationäkymästä luodaan uusi Webhook. Kenttiin laitetaan kuvan 9 mukaisesti pyydetyt asiat:

- Event Name, laitetaan sen julkaisun nimi, joka aktivoi Webhookin ja lähettää JSONin rajapintaan.
- URL, API-avain syötetään rajapinnan URL-osoitteen perään. Rakenne on muotoa: URL + "?apikey=" + API-avain.
- Request Type osioon lisätään halutun metodin muoto, valitsemme POST.
- Request Format, valitaan pyynnön muoto.
- Device, kohdassa valitaan mitkä tilin laitteet saavat Webhookin käyttöönsä.
- Status, tästä voi Webhookin kytkeä pois käytöstä väliaikaisesti.

Sandbox 

[Docs](#) | [Contact Sales](#)

[Integrations](#) > Edit Integration

WEBHOOK BUILDER CUSTOM TEMPLATE

[Read the Particle webhook guide](#)

Event Name ⓘ
saa

URL ⓘ

Request Type
POST

Request Format
JSON

Device
Any

Status
Enabled

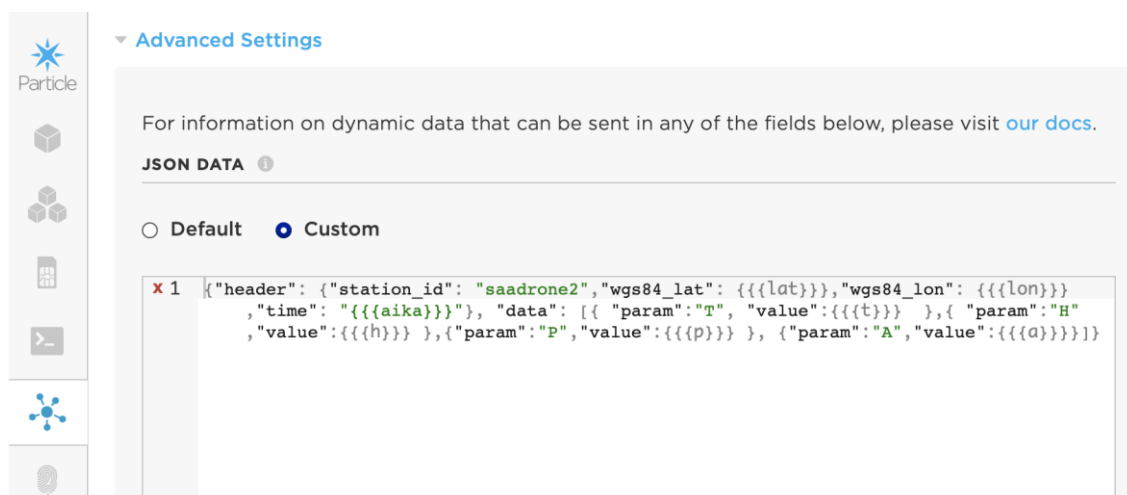
▶ [Advanced Settings](#)

CANCEL

SAVE

Kuva 9. Webhookin luominen Ilmatieteen laitoksen kokeelliseen rajapintaan.

Kuvan 10 mukaisesti "Advanced Settings" -osioon lisätään JSON-rakenne. Integraation oma POST-metodi aktivoituu, kun se havaitsee määrittelemäsi tapahtuman julkaisun Electronilta.



Kuva 10. JSON-datan luominen. Avainten arvoina käytetty muuttujia pääohjelmasta.

Webhook tarjoaa mahdollisuuden sijoittaa koodissa olevia muuttujia avainten arvoksi laittamalla muuttuja kolmen siipisulkeen sisään. Kyseiset muuttujat on löydyttävä samalla nimellä sovelluksen koodista. Tallennuksen jälkeen integraatio on valmis käytettäväksi. Webhook aktivoituu, kun alussa määritelty Event julkaistaan.

8 Particle-koodi

8.1 Määritykset

Virallisesti Particle ei ole tuettuna Grove-barometrin yhteensopivana alustana. Sisällyttämällä Adafruitin valmistaman barometrin kirjastot, Groven barometri saadaan toimimaan. Particlen ja barometrin välillä käytetään I2C-sarjaliikennettä.

Koodin alustuksessa tehdään kuvan 11 mukaiset määritykset:

- Sisällytetään "particle.h"-kirjasto.
- Sisällytetään Google Device Locator -kirjasto.
- Sisällytetään Adafruit-kirjastot.

- Määritetään MOSI (datalinja) ja SCK (kellolinja) nastat barometrille.
- Määritetään merenpinnan korkeuden ilmanpaine.

```
1
2 #include "Particle.h"
3 #include <Adafruit_BME280.h>
4 #include <Adafruit_Sensor.h>
5 #include <google-maps-device-locator.h>
6 #define BME_MOSI D0
7
8
9
10
11 #define BME_SCK D1
12 #define SEALEVELPRESSURE_HPA (1013.25)
13
14
15 GoogleMapsDeviceLocator locator;
16
17 Adafruit_BME280 bme;
18
```

Kuva 11. Ohjelmakoodin määrytykset sekä eri kirjastojen sisällytys.

8.2 Setup

Setup (kuva 12) suoritetaan vain kerran ohjelmaa käynnistäessä. Määritetään DAC-nastasta ulostuloportti ja tehdään ulostulo jännitteettömäksi. Google Geolocator -tilauksen yhteyteen määritettiin pyyntöjen frekvenssi sulkujen sisään. Mikäli väyläyhteys sensorilta Particleen toimii, ohjelmakoodi julkaisee: "BME 280 mittaus ok!".

```

24 void setup() {
25
26
27   pinMode(DAC, OUTPUT);
28   analogWrite(DAC, 0);
29
30   Serial.begin(9600);
31
32
33   locator.withSubscribe(locationCallback).withLocatePeriodic(15);
34
35   if (bme.begin(0x76)) {
36     Particle.publish("BME280 mittaus ok!",PRIVATE);
37   }
38
39
40
41
42
43 }
44

```

Kuva 12. IR-moduulin LEDin eksitaatiojännitteen alustus, sijainnin tilaus Google Geolocatorilta ja väyläyhteyden tarkistus.

8.3 Pääohjelma

Setupin jälkeen koodissa siirrytään Pääohjelmaan (kuva 13). Google Geolocatorin palauttavat "lat" ja "lon" sekä olosuhdeanturilta saadut arvot sijoitettiin muuttujiin. Olosuhdeanturin muuttujille määritetään jokin epäolennainen arvo alustuksen vuoksi.

Parametreista ja paikkatiedosta luodaan muuttuja "data". Kyseinen muuttuja formatoidaan merkkijonoksi. Muuttujaan lisätään vielä aikaleima oikeassa muodossa. Publish-metodilla saadaan julkaistua muuttuja, joka näkyy Event-osiossa. Julkaisulle annetaan nimi "saa", joka aktivoi Webhookin. Kuvassa 16 rivillä 83 Loop-osioon lisätään Geolocatorin silmukkarakenne.

```

45
46 void locationCallback( float lat, float lon, float accuracy) {
47
48
49     float t,h,p,a = -99;
50
51
52
53     t = bme.readTemperature(); // *C
54     h = bme.readHumidity(); // %
55     p = bme.readPressure() / 100.0F; // hPa
56     a = bme.readAltitude(SEALEVELPRESSURE_HPA);
57
58
59
60
61
62
63     String data = String::format(
64     "{ \"t\": %f, \"h\": %f, \"p\": %f, \"a\": %f, \"lat\":%f, \"lon\":%f, \"aika\":",
65     t,h,p,a,lat,lon);
66     data = data + "\"" + Time.format(Time.now(), "%Y-%m-%dT%H:%M:%SZ") + "\"}";
67
68
69     Particle.publish("saa", data, PRIVATE);
70
71
72
73 }
74
75
76

```

Kuva 13. Mittausarvojen tallennus muuttujiin ja "data"-muuttujan julkaisu.

Kuvan 14 alaosiosta nähdään paluukoodi "200:OK", joka tarkoittaa onnistunutta vastaanottamista rajapinnalta. Lähetetty JSON-rakenne näkyy oikealla, jossa kaikki mitattavat parametrit.

📄 🔍 🔍 Search for events ADVANCED			
NAME	DATA	DEVICE	PUBLISHED AT
heijastavuus	473.000000	Electron_-_Mikael	6/28/21 at 1:56:25 pm
hook-response/deviceLo...	61.0030611,25.6655256,905	particle-internal	6/28/21 at 1:56:25 pm
hook-sent/deviceLocator		particle-internal	6/28/21 at 1:56:24 pm
deviceLocator	{"c":{"o":"DNA","a":{"i":"7111...	Electron_-_Mikael	6/28/21 at 1:56:24 pm
heijastavuus	476.000000	Electron_-_Mikael	6/28/21 at 1:56:23 pm
heijastavuus	463.000000	Electron_-_Mikael	6/28/21 at 1:56:22 pm
heijastavuus	471.000000	Electron_-_Mikael	6/28/21 at 1:56:20 pm
heijastavuus	467.000000	Electron_-_Mikael	6/28/21 at 1:56:19 pm
heijastavuus	467.000000	Electron_-_Mikael	6/28/21 at 1:56:18 pm
heijastavuus	468.000000	Electron_-_Mikael	6/28/21 at 1:56:16 pm
heijastavuus	476.000000	Electron_-_Mikael	6/28/21 at 1:56:15 pm
heijastavuus	472.000000	Electron_-_Mikael	6/28/21 at 1:56:13 pm
heijastavuus	474.000000	Electron_-_Mikael	6/28/21 at 1:56:12 pm
hook-response/saa/0	("200": "OK")	particle-internal	6/28/21 at 1:56:10 pm

saa

Published by [REDACTED] on 6/28/21 at 1:56:10 pm

PRETTY

RAW

```
{
  "t": 26.24
  "h": 39.96582
  "p": 1001.626404
  "a": 97.22673
  "lat": 61.003059
  "lon": 25.665525
  "aika": "2021-06-28T10:56:09Z"
}
```

Kuva 14. Event-näkymässä rajapintaan lähetetyt arvot.

9 Jäätämisen havainnointi

Työn toinen tilaaja Insta oli kiinnostunut UAV:n jäätämisestä lennon aikana.

Insta oli kiinnostunut saamaan tuloksia seuraaviin aiheisiin:

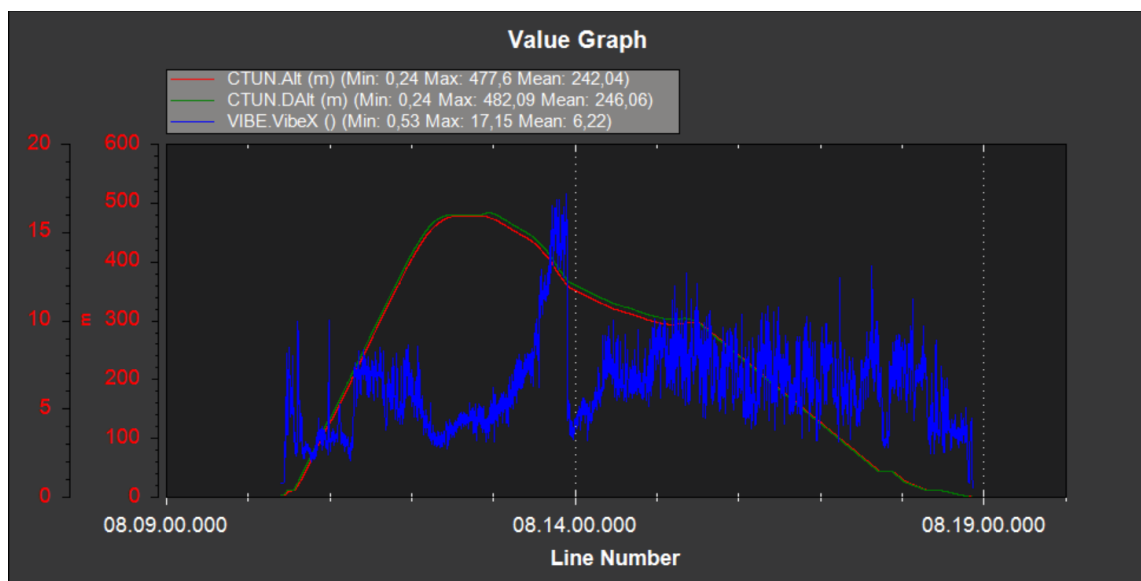
- Voiko dronen telemetriasta luotettavasti havaita lentokelpoisuutta?
- Mihin telemetrian parametreihin jäätäminen selkeästi vaikuttaa?
- Pyritään kehittämään algoritmi, joka havaitsee jäätämisen ja antaa siitä ilmoituksen.

Dronen olisi täten pystyttävä lennossa päättelemään, onko jäätä rakenteissa ja millä todennäköisyydellä. Jäätämisen havainnointiin pystyy käyttämään ja täytyy käyttää muitakin keinoja kuin moottorin telemetria. Jäätämisen havainnointiin sain hyvin vapaat kädet, kuinka haluan sen suorittaa.

Alun perin tavoitteeni oli käyttää telemetriaa reaaliaikaisesti algoritmin kanssa todennäköisyyden tunnistamiseen jäätämisen tilasta. Insinööriyön muut vaiheet veivät aikaa oletettua enemmän sekä lukuisat takaiskut hidastivat tekemistä. Täten reaaliaikainen ennustus jäätämisestä algoritmin avulla jäi tekemättä. Drone rikkoutui projektin aikana oikosulun seurauksena. Varaosia tilattiin Saksasta ilmailuharrastukseen keskittyneestä liikkeestä, mutta toimitusaika aiheutti ongelmia pandemian aikana. Projektin takaraja tuli vastaan, ja täten testilennot jäätävissä olosuhteissa jäivät puuttumaan.

9.1 Telemetria

Ilmatieteen laitos toimitti lentodataa eräältä lennoltaan, jolloin dronen propellit olivat keränneet jäätä. Luvussa 5 mainittiin jäätämisen mahdollisista vaikutuksista ilma-alukseen. Siinä kerrottiin jään voivan vaikuttaa aluksen kykyyn päästä tavoitekorkeuteen sekä värähtelyn lisääntymisestä. Tutkittuani lennon telemetriaa tehtiin kuvan 15 esittämiä havaintoja.



Kuva 15. Tavoitekorkeuden ja mitatun korkeuden eroavaisuus sekä X-akselin mukaista värähtelyä.

Kuvasta voidaan nähdä, että tavoitekorkeuden ja todellisen korkeuden välillä on eroa. Myös X-akselin näkökulmasta havaittavissa on värähtelyä. Nämä

havainnot voivat todistaa teorian todeksi. Mahdolliseen jäätämisen koneoppimisessa olisi järkevää huomioida nämä arvot.

9.2 Käytännön sovellus

Projektissa tutkittiin myös konenäön mahdollisuuksia jäätämisen havaitsemiseen telemetrian lisäksi. Vaihtoehtona olisi ollut toteuttaa jäätämisen havaitseminen Arduino Portentan avulla. Jälleen toimitusvaikeudet osoittautuivat ongelmaksi.

Päätettiin käyttää vaihtoehtoista suunnitelmaa jäätämisen havaitsemiseen. Tarkoituksena oli todeta anturien tuottamasta datasta sekä visuaalisesti optisella anturilla, onko ilmiö todennäköinen lennon aikana. Rakennettu järjestelmä varustettiin läpinäkyvällä muovikannella. Kanteen osoitettiin IR-LED, joka heijastaa infrapunavaloa. Kansi heijastaa valon takaisin mittauslaatikon sisään, missä on vastaanotin. Testien aikana huomattiin kuitenkin eroavaisuuksia mittauksissa. Auringon valo säteilee IR-valoa, joka aiheutti vääriä mittauksia. Tämä suodatettiin pois Particlen koodissa jättäen pelkkä IR-LEDin heijastama valo.

```

83 locator.loop();
84
85
86 delay(50); //Viive ennen mittauksia
87
88 double out_neg = analogRead(A3); //Mittaa out-
89
90
91 delay(50);
92 double out_pos = analogRead(A4); //Mittaa out+
93
94 double sunlight = out_pos - out_neg; // Auringon valo on mitattujen signaalien erotus
95
96 analogWrite(DAC, 1024); // Kytetään päälle eksitaatiojännite omalle IR-LEDille
97
98 delay(50);
99
00 double out_neg2 = analogRead(A3); //Mitataan signaali uusiin muuttujiin
01
02
03 delay(50);
04 double out_pos2 = analogRead(A4); //Mitataan signaali uusiin muuttujiin
05
06 double diff2 = out_pos2 - out_neg2; // diff2 = IR-LED + Aurinko infrapuna säteily
07
08 double ir = diff2 - sunlight; // Vähennetään mitattu auringon vaikutus, jolloin jäljelle jää IR-LEDin heijastavuus
09
10
11
12 Particle.publish("heijastavuus", String(ir), PRIVATE); // Julkaistaan arvo, joka näkyy Event-näkymässä
13
14
15 analogWrite(DAC, 0);
16
17
18 delay(1000);
19
20
21 }

```

Kuva 16. Electronin koodin silmukkaosio, jossa IR-moduulin ohjelmakoodi. Koodissa mitataan IR-moduulin signaalia ja suodatetaan haluamaton infrapunavalon pois.

Ensin mitattiin sensorin havaitsema auringonvalo LEDin ollessa pois päältä. Tämä tallennetaan muuttujaan "auringonvalo", minkä jälkeen välittömästi nostetaan eksitaatiojännite LEDille arvoon 1024. Suoritetaan mittaus uudestaan, jossa $\text{diff2} = \text{LEDin valo} + \text{auringonvalo}$, ja tämäkin tallennetaan muuttujaan. Kolmannessa muuttujassa vähennetään toisesta mittauksesta auringonvalo, jolloin jää jäljelle pelkkä IR-LEDin todellinen heijastus. Tämä viimeinen arvo julkaistaan, joka näkyy Event-osiossa.

heijastavuus	476.000000	Electron_-_Mikael	6/28/21 at 1:56:23 pm
heijastavuus	463.000000	Electron_-_Mikael	6/28/21 at 1:56:22 pm
heijastavuus	471.000000	Electron_-_Mikael	6/28/21 at 1:56:20 pm
heijastavuus	467.000000	Electron_-_Mikael	6/28/21 at 1:56:19 pm
heijastavuus	467.000000	Electron_-_Mikael	6/28/21 at 1:56:18 pm
heijastavuus	468.000000	Electron_-_Mikael	6/28/21 at 1:56:16 pm
heijastavuus	476.000000	Electron_-_Mikael	6/28/21 at 1:56:15 pm
heijastavuus	472.000000	Electron_-_Mikael	6/28/21 at 1:56:13 pm
heijastavuus	474.000000	Electron_-_Mikael	6/28/21 at 1:56:12 pm

Kuva 16. Event-osio. Suodatuksen jälkeen heijastavuuden arvot olivat tasaisia.

Kannen jäätyessä heijastavuus pienenee. Kansi kasteltiin ja jäädytettiin, jolloin heijastavuus putosi jopa kymmenesosaan alkuperäisestä.

9.3 Jatkokehitysidea

Dronekit on avoin ja ilmainen Python-kirjasto dronen ja käyttäjän väliseen kommunikointiin. Se tarjoaa pääsyn yhdistetyn laitteen telemetriaan ja muihin parametreihin. Dronekitin avulla voisi olla mahdollista saada telemetria vietyä eteenpäin reaaliaikaista käsittelyä varten. Tätä ei ehditty kehittämään tai testaamaan pitemmälle, vaan esitellään projektin jatkokehitystä varten.

Lisätietoa työkalusta (kuva 17) löytyy osoitteesta

<https://github.com/dronekit/dronekit-python>.

 peterbarker examples: set_attitude_target: avoid use of None in to_quaternion	
..	
channel_overrides	Modernize Python 2 code to get ready for Python 3
create_attribute	revise URL in comment.
drone_delivery	Python 3: define basestring, use bytes.decode() (#705)
flight_replay	Modernize Python 2 code to get ready for Python 3
follow_me	examples/follow_me: set timeout on connect to 300
gcs	Modernize Python 2 code to get ready for Python 3
guided_set_speed_yaw	Modernize Python 2 code to get ready for Python 3
mission_basic	Modernize Python 2 code to get ready for Python 3
mission_import_export	mission import / export example :
performance_test	Modernize Python 2 code to get ready for Python 3
play_tune	Add play_tune() method on vehicle, example for same
reboot	examples: add a reboot example
set_attitude_target	examples: set_attitude_target: avoid use of None in to_quaternion
simple_goto	Python 3: define basestring, use bytes.decode() (#705)
vehicle_state	Modernize Python 2 code to get ready for Python 3
README.md	s/diydrones/dronekit-python/dronekit/dronekit-python/

Kuva 17. Dronekitin useat kirjastot Githubissa.

Telemetria on mahdollista tulostaa reaaliaikaisesti. Käyttäjä Tiziano Fiorenzani on tehnyt esimerkkikoodin telemetrian lukuun. Valitut parametrit saadaan tulostettua joko simuloidussa tilassa tai yhdistämällä droneen.

10 Yhteenveto

10.1 Tavoitteet ja tulokset

Insinööriyön yhtenä tavoitteena oli kerätä dataa ja kehittää tiedonsiirron malli, jolla data saataisiin siirrettyä Ilmatieteen laitoksen rajapintaan. Tuloksena kehitettiin lähetyslaatikko, joka toteutti tämän. Laatikko on liitettävissä melkein mihin vain liikkuvaan laitteeseen ilman mitään riippuvuuksia. Dataa kerättiin

onnistuneesti anturilla ja tarkka paikkatieto saatiin myös mukaan. Pitkän hiomisen jälkeen rajapinta saatiin onnistuneesti vastaanottamaan dataa JSON-muodossa. Tämä onnistui myös olematta riippuvainen langattomista verkoista, koska lähetyslaatikolla on SIM-kortillinen 3G-valmius. Mielestäni datan siirtoon ja mittaukseen liittyvät vaatimukset täytettiin.

Toinen insinööriyön tavoite oli jäätämisen tutkiminen ilmiönä. Tavoitteeni oli tutkia, voiko dronen telemetriasta luotettavasti havaita jäätämistä, sekä toteuttaa mahdollisesti koneoppimismalli. Dronella ei päästy suorittamaan testilentoja luonnollisissa jäätämisolosuhteissa. Tähän vaikutti projektin lyhyehkö kesto sekä laitteiston vikaantuminen. Työssä käytettävän drone oli todella tee-se-itse-mallinen, mikä tarkoitti todellista perehtymistä aihealueeseen ja laitteistoon. Telemetrian reaaliaikainen hyödyntäminen osoittautui hankalaksi, eivätkä jälkikäteen SD-kortille tallennetut tiedostot olleet valmiiksi käsiteltävässä muodossa. Jäätämisen tutkimiseen liittyy olennaisesti myös aluksella lentäminen. Lentäminen oli kaukana käyttäjäystävällisestä ja vaati liian pitkän tutustumisen. Harmillisesti laitteiston vikaantumisen sekä aihealueen vaikea oppimiskäyrä aiheutti lentodatan puuttumisen.

Referenssidatana pystyttiin käyttämään Ilmatieteen laitokselta saatua dataa sekä edellisenä vuonna tuotettua dataa. Datasta oli havaittavissa lento- ja tavoitekorkeuden eroavaisuutta. Tämä voi mahdollisesti johtua jäätävistä olosuhteista. Jäätämistä käsittelevässä teoriaosuudessa esittelin jäätämisen vaikutuksia erilaisiin tekijöihin, jotka voivat vaikuttaa dronen telemetriaan. Yksi vaikuttavista tekijöistä oli lento- ja tavoitekorkeuden eroavaisuus. Infrapunavälillä kuitenkin saatiin muodostettua jäätämisen havainnoimiseen optinen sovellus. Mielestäni tavoite saatiin ainoastaan osittain saavutettua. Tässä aihealueessa olisi kehityksen varaa.

10.2 Kehitysideat ja pohdinta

Ilmatieteen laitoksen rajapinnan lisäksi mittausdata voitaisiin lähettää muuhunkin tietokantaan. Täten voitaisiin muodostaa visuaalisesti miellyttävä

hallintapaneeli, jossa näkyisi mitatut tiedot. Jäätämisen tila voisi myös näkyä paneelissa, jolloin paneeli varoittaisi tietyn raja-arvon ylittyessä. Tähän hyviä työkaluja olisivat esimerkiksi MongoDB ja Node-red.

Jäätämisen tutkimiseen pitäisi keskittää kehitystyötä. Ensimmäinen tehtävä olisi varmistaa telemetrian saanti laitteesta reaaliaikaisesti eikä vain jälkikäteen SD-kortilta. Tästä esimerkkinä aikaisemmin mainittu Dronekit-kirjasto.

Koneoppimisalgoritmina suosittelen käyttämään monimuuttujaregressioanalyysia (MVR) tai päättelypuuta (DT) jäätämisen todennäköisyyden arviointiin.

Työn pohjana voisi olla hyvä jatkossa käyttää käyttäjäystävällisempää dronea. Aikaa tuhlautui todella paljon dronen tutustumiseen ja sen käyttöön. Malli voisi olla sopiva drone-harrastajalle, mutta aiheeseen tutustumattomalle oppimiskäyrä oli todella haastava. Testilennoissa oli suuri riski laitteen rikkoutumiselle laitteen haastavan ohjauksen vuoksi. Suosittelen tutustumaan enemmän kaupallisten mallien tarjontaan. Tämä olisi tehtävä telemetrian saamisen näkökulmasta.

Lähteet

- 1 Ilmatieteen laitos – turvaa ja sujuvuutta yhteiskunnalle. Verkkoaineisto. Ilmatieteen laitos. <<https://www.ilmatieteenlaitos.fi/ilmatieteen-laitos>>. Luettu 20.3.2021.
- 2 Laki Ilmatieteen laitoksesta. 2018. 212/6.8.2018.
- 3 Lupa luottaa jo 60 vuoden ajan. Verkkoaineisto. Insta. <<https://www.insta.fi/instan-tarina>>. Luettu 20.3.2021.
- 4 Insta on turvallisen ja kilpailukykyisen yhteiskunnan rakentaja ja kehittäjä. Verkkoaineisto. Insta. <<https://www.insta.fi/tietoa-meista>>. Luettu 20.3.2021.
- 5 Wilenius, Markku. 2014. Leadership in the sixth wave—excursions into the new paradigm of the Kondratieff cycle 2010-2050. Verkkoaineisto. <<https://link.springer.com/content/pdf/10.1007/s40309-014-0036-7.pdf>>. 6.3.2014. Luettu 25.3.2021.
- 6 Chebudie, Abiy Biru; Minerva, Roberto & Rotondi, Domenico. 2014. Towards a definition of the Internet of Things (IoT). Politecnico di Torino. Verkkoaineisto. <https://www.researchgate.net/publication/317588072_Towards_a_definition_of_the_Internet_of_Things_IoT>. 27.5.2015. Luettu 1.4.2021.
- 7 Firouzi, Farshad; Chakrabarty, Krishnendu & Nassif, Sani. 2020. Intelligent Internet of Things. Springer, Cham. Verkkoaineisto. <<https://link.springer.com/book/10.1007/978-3-030-30367-9>>. 2020. Luettu 12.4.2021.
- 8 Buyya, Rajkumar & Dastjerdi, Amir Vahid. 2016. Internet of Things. E-kirja. Morgan Kaufmann. 2016. Luettu 18.4.2021.
- 9 What is 5G? Verkkoaineisto. Cisco. <<https://www.cisco.com/c/en/us/solutions/what-is-5g/5g-vs-4g.html#~benefits-of-5g>>. Luettu 25.4.2021.
- 10 5G used for monitoring blue-green algae in the Baltic Sea. 2019. Verkkoaineisto. Telia. <<https://www.teliacompany.com/en/news/news-articles/2019/5g-used-for-monitoring-blue-green-algae-in-the-baltic-sea>>. Luettu 10.5.2021.
- 11 Leinonen, Reijo. 2020. Lehtori, Automaatiotekniikka, Metropolia ammattikorkeakoulu. Verkkoluento – Integroitimenetelmät. 27.10.2020.
- 12 Representational State Transfer (REST ja Web-suuntautunut arkkitehtuuri (WOA) arkkitehtuurityylinä. 2009. Verkkoaineisto. Helsingin Yliopisto.

- <https://www.cs.helsinki.fi/group/cinco/teaching/2009/soc-seminaari/papers/kotilainen_paper.pdf>. Luettu 20.5.2021.
- 13 Bourhis, Pierre; Reutter, Juan L.; Suárez, Fernando & Vrgoč, Domagoj. 2017. JSON: data model, query languages and schema specification. Cornell University. Verkkoaineisto. <<https://arxiv.org/abs/1701.02221v1>>. Luettu 23.5.2021.
 - 14 Hassanalian, Mostafa & Abdelkefi, Abdessattar. 2017. Classifications, applications, and design challenges of drones: A review. Verkkoaineisto. <https://www.researchgate.net/publication/316673697_Classifications_applications_and_design_challenges_of_drones_A_review>. New Mexico State University. 2017. Luettu 30.5.2021.
 - 15 Hann, Richard. 2020. Atmospheric Ice Accretions, Aerodynamic Icing Penalties, and Ice Protection Systems on Unmanned Aerial Vehicles. Norwegian University of Science and Technology. Verkkoaineisto. <<https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2657638>>. 2020. Luettu 7.6.2021.
 - 16 Liu, Dianxiong; Xu, Yuhua; Wang, Jinlong; Chen, Jin; Yao, Kailing; Wu, Qihui & Anpalagan, Alagan. 2020. Opportunistic UAV Utilization in Wireless Networks: Motivations, Applications, and Challenges. IEEE Communications Magazine. Verkkoaineisto. <<https://ieeexplore.ieee.org/abstract/document/9112744>>. 2020. Luettu 12.6.2021.
 - 17 Ilmakehä ABC, jäätyminen. Verkkoaineisto. Ilmatieteen laitos. <<https://www.ilmatieteenlaitos.fi/ilmakeha-abc?term=J%C3%A4%C3%A4tyminen>>. Luettu 18.6.2021.
 - 18 Ilmanpaine. Verkkoaineisto. Ilmatieteen laitos. <<https://www.ilmatieteenlaitos.fi/ilmanpaine>>. Luettu 18.6.2021.
 - 19 Comission Delegated Regulation (EU) 2019/945. 2019. Verkkoaineisto. Official Journal of the European Union. <https://eur-lex.europa.eu/eli/reg_del/2019/945/oj>. Luettu 24.6.2021.
 - 20 Ilmakehä ABC, Suhteellinen kosteus. Verkkoaineisto. Ilmatieteen laitos. <<https://www.ilmatieteenlaitos.fi/ilmakeha-abc?term=Suhteellinen%20kosteus>>. Luettu 24.6.2021.
 - 21 Quickstart Electron. Verkkojulkaisu. Particle. <<https://docs.particle.io/quickstart/electron/>>. Luettu 1.7.2021.