

Loppuraportti

Projekti #3: Consair IoT: Rakennustyömaan olosuhteiden seuranta



Date: XX.8.2018

Jere Miely
Sampo Lappalainen
Lauri Westerholm
Mikko Lähtenmäki
Vikke Tiirola

Information page

Students

Jere Mielty

Sampo Lappalainen

Lauri Westerholm

Mikko Lähteenmäki

Vikke Tirola

Project manager

Jere Mielty

Sponsoring Company

Consair Oy

Starting date

4.6.2018

Submitted date

31.8.2018

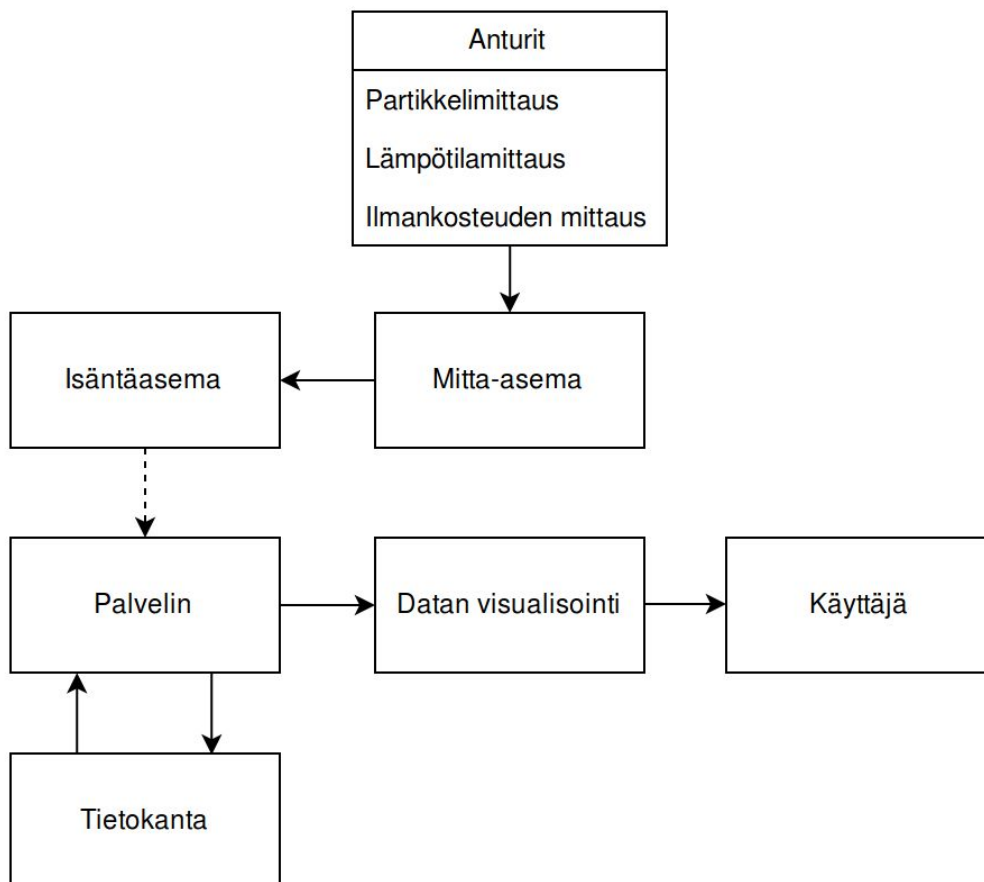
Tiivistelmä

Tämä dokumentti esittelee Aalto-yliopiston Protopaja-kurssilla kesällä 2018 oppilastyönä rakennetun rakennustyömaan olosuhteiden seurantalaitteen prototyypin. Projektin tavoitteena oli suunnitella IoT-toteutus, joka mittaa työmaan olosuhteita, käsittelee mittadatan helposti luettavaan muotoon ja lähettää datan internetin kautta käyttäjälle.

Järjestelmä koostuu useammasta laitteesta. Mitta-asetat mittaavat ilmankosteutta, lämpötilaa ja pölyisyyttä ja lähettävät datan radioteitse isäntäasemalle, joka lähettää sen eteenpäin palvelimelle. Palvelin muokkaa datan esitettävään muotoon loppukäyttäjälle.

Projektin aikana on testauksen myötä saatu valittua käyttötarkoitukseen soveltuvat anturit, jotka on liitetty osaksi mitta-asettaa. Mitta-asetasta on saatu onnistuneesti kerättyä dataa isäntäasemalle ja jatkolähetettyä dataa isetetoteutetulle palvelimelle.

Lohkokaaviossa on kuvattu järjestelmää yleisellä tasolla.



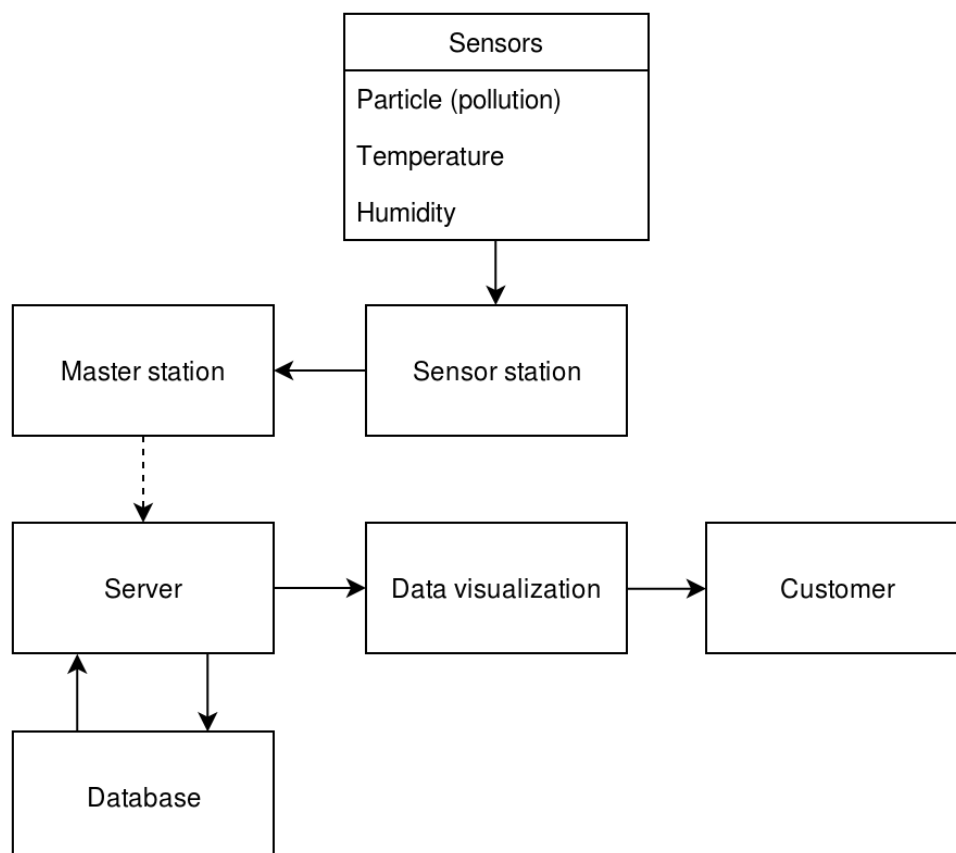
Abstract

This document presents the prototype of a construction site condition tracker device built at the Aalto University's Protopaja course in the summer 2018. The aim of the project was to design an IoT device which measures construction site conditions and sends the collected data over the Internet to the user in a readable format.

The system is made of several devices. Sensor stations measure air humidity, temperature and pollution. Sensor stations transmit the data over radio to a master station that sends it to the server. The server modifies the data in the form presented to the end user.

During the project, testing has resulted in the selection of suitable sensors for the sensor station. Data from the prototype version of the sensor station has successfully been collected to the master station and been forwarded to a self-build server.

Block diagram of the system.



Sisällysluettelo

Tiivistelmä	3
Abstract	4
Sisällysluettelo	5
Johdanto	7
Tavoite	7
Komponentit	8
Mikro-ohjain	8
Muut komponentit	8
Mitta-asema	8
Anturit	9
Prototyypissä tarvittavat anturit	9
Valitut anturit	9
Perusteet anturivalinnoille	11
Anturien kehitysehdotukset	11
Tiedonvälitys	12
Tiedonsiirron mahdolliset toteutustavat	12
Tiedonsiirto mitta-asemien ja isäntäaseman välillä	13
Toteutustapa	13
Tekniikan vahvuudet	14
Tekniikan rajoitteet	14
Kehitysmahdollisuudet	15
Isäntäaseman ja palvelimen välillä	16
3D-mallinnus kuoren suunnittelussa ja kuoren valmistaminen	17
PCB:n suunnittelu ja valmistus	23
Web	24
Palvelin	24
Nettisivut	26
Aikaansaannokset	27
Järjestelmän käyttöohjeet	27
Mitta-asetat	27
Isäntäasema	29
Palvelimen ulkoasu	30
Projektitoiminta	38
Tavoitteiden saavuttaminen	38

Aikataulu	39
Riskianalyysi	40
Lopullinen tehtäväjako	42
Yhteenveto ja johtopäätökset	44
Henkilökohtaiset oppimiskokemukset	44
Johtopäätökset	45
Liitteet	45
Lähteet	45

1. Johdanto

Pölynhallinta ja ilmanlaatu ovat työterveyden kannalta keskeisiä osa-alueita työmailla. Koska pölynhallinnan tehokkaaseen toteutukseen tarvitaan mittaustietoa ilmanlaadusta, ei tällaisen pölynhallinnan toteuttaminen ole useimmiten rakennustyömailla mahdollista ilman ulkoista ratkaisua. Rakennustyömailla olosuhteet ovat muutenkin haastavat vaihtuvan henkilöstön ja mekaanisen rasituksen takia.

Tässä tapauksessa ilmanlaadun sekä muun ympäristömittausdatan tehokas mittaaminen vaatii käytännössä kestävään kuoreen suljettua anturiyksikköä. Tällä tavalla kerätyn tiedon voi lähettää lähes reaaliajassa eteenpäin pilveen, jossa tieto prosessoidaan helpommin ymmärrettävään muotoon. Tämän projektin kimmokkeena on tarve saada tietoa rakennustyömaan ilmanlaadusta. Tuotteelle voidaan visioida myös muitakin sovelluksia, kuten ympäristön seuraaminen betonin kuivumisen ja työntekijöiden terveyden kannalta. IoT -ratkaisuille on tyypillistä, että tuotteille löytyy myös ennalta arvaamattomia käyttötarkoituksia.

2. Tavoite

Tavoitteena on suunnitella ja toteuttaa rakennustyömaalle prototyypilaitte, joka mittaa työmaan olosuhteita, lähettää datan erillisen verkkoyksikön (isäntäaseman) kautta pilveen ja raportoi olosuhteista käyttäjälle (työmaamestarille). Käyttäjälle suunnitellaan graafinen selainpohjainen käyttöliittymä. Käyttöliittymästä tulee nähdä reaaliaikaista sekä myös aiemmin kerättyä dataa työmaan olosuhteista ja käyttäjä saa ilmoituksia mikäli mittausarvot ylittävät viitearvot. Tällaisia mitta-aseimia on tarkoitus rakentaa useampia, jolloin saadaan dataa useammasta kohteesta. Laitte mittaa pölyisyyttä, ilmankosteutta ja lämpötilaa. Laitteen toiminnallisuutta demonstroidaan mahdollisesti kenttätestaamalla laitetta oikealla työmaalla.

3. Komponentit

3.1. Mikro-ohjain

Laitteen kehityksen alussa suunnitelmassa oli käyttää Pycomin Wipy 3.0 -mikro-ohjainta keräämään mittausdataa siihen liitetystä antureista ja lähettämään se wifi-ominaisuuksiensa avulla pilvipalveluun nähtäväksi, joko itsenäisenä wifiin kytkettynä mitta-asemana tai muiden mitta-asemien dataa välittävänä isäntäasemana. Tämä toteutus toimi välittämällä anturimoduulin dataa Pycomin omaan selainpalveluun. Suureksi ongelmaksi kuitenkin ilmeni kirjastojen ja muun koodaustuen vähyys python-pohjaisella Wipyllä, minkä vuoksi partikkelianturin ja radiomoduulin käyttöönotosta arvioitiin tulevan todella työlästä. Vaikka isäntäaseman antureista luovuttiin lopulta, oli isännän kyky radioyhteyteen välttämättömyys valitsemamme tiedonsiirtotopologian kannalta.

Mitta-aseman kohdalla päätimme siirtyä Arduinoon ryhmänjäsenen omakohtaisen käyttökokemuksen sekä Arduinon laajan kirjasto- ja koodaustuen perusteella. Leipälautatesteissä on riittänyt Arduino UNO ja lopullisessa, PCB:n sisältävässä prototyypissä tarvitsemme pelkästään ATmega328p:n. ATmega328p:ssä (siis myös Arduino UNOssa) on juuri sopivasti väyliä ja toiminnallisuutta kokonaista mitta-asemaa varten.

Isäntäasema vaihdettiin Wipystä NodeMCU:hun (ESP32:sta ESP8266:een). NodeMCU:n pystyy yhdistämään wifiin samalla lailla kuin Wipyn. Toisin kuin Wipy, NodeMCU on täysin yhteensopiva Arduinon koodin ja koodausympäristön kanssa, mikä tekee siirtymän Arduino UNOjen välisestä testauksesta käyttöönottoon NodeMCU:lla nopeaa.

3.2. Muut komponentit

Mitta-asema

- 128x32 pikselinen oled-näyttö
- nRF24L01 radiomoduuli (käytössä myös isäntäasemassa)
- Tuulettimet
- Transistorit (NPN BC337)
- 3,8 V Lipo-akku

Mitta-asemissa päädyttiin käyttämään pienikokoista (128x32) oled-näyttöä aseman statuksen ilmoittamiseen. Käyttäjä siis näkee näytöstä, onko asema kytketty onnistuneesti isäntäasemaan. Näyttö sopii myös ongelmatilanteista ilmoittamiseen.

Tuulettimia käytetään tehostamaan ilmankiertoa kotelossa ja siten parantamaan pölymittauksen luotettavuutta. Tarkoituksena oli asettaa laite lähetysten välillä lepotilaan ja kytkeä npn-transistoreilla anturit pois päältä, jotta laitteen virrankulutusta saataisiin optimoitua. Anturien poiskytkeminen aiheutti kuitenkin ongelmia anturien luotettavuuden kanssa, joten siitä luovuttiin. Viimeisessä piirilevyversiossa ainoastaan tuuletin kytketään pois päältä transistorin avulla.

Lopullinen prototyypiasema saa käyttöjännitteensä yksikennoisesta Lipo-akusta. Akku pystytään lataamaan usb-liittimen avulla piirilevyyn kytketyn latauspiirin kautta.

4. Anturit

4.1. *Prototyypissä tarvittavat anturit*

Prototyypissä tarvittavat anturit riippuvat suoraan laitteen käyttötarkoituksesta. Toteutetun prototyypin on tarkoitus mitata sisätyömaan ympäristöolosuhteita tiloissa työskentelevien ihmisten terveyden kannalta. Näin ollen pölymittaus (pienhiukkasmittaus) on laitteen keskeisimpiä tehtäviä. Laitteeseen tarvitaan siis pölyanturi, joka kykenee ilmoittamaan sisäilmanlaadun kvalitatiivisella tarkkuudella: vaarallinen, huono, kohtalainen, hyvä.

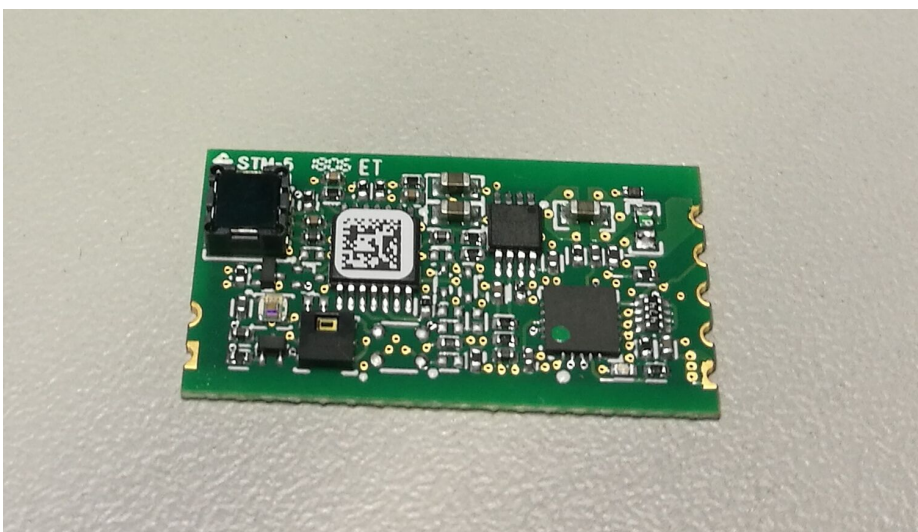
Pölymittauksen lisäksi työntekijöiden turvallisuuden sekä viihtyvyyden kannalta keskeistä on mitata huoneen lämpötilaa. Tähän tarvitaan lämpötila-anturi. Rakennustyön laadun osalta on puolestaan olennaista mitata ilmankosteutta, koska kosteus vaikuttaa esimerkiksi laastin, betonin, sekä maalien kuivumiseen.

Välttämättömiä prototyyppiin ovat siis pöly-, lämpötila- ja ilmankosteusanturit. Näitä antureita tarvitaan kaikille mitta-asemille, isäntäasema ei tarvitse lainkaan antureita.

4.2. *Valitut anturit*

-AmbiMate MS4 Sensor Module (Kuva 1):

Sisältää liikkeentunnistus-, valo-, lämpötila- ja ilmankosteusanturit. Toimii I2C-väylällä osoitteessa 0x2A. Kykenee mittaamaan valoisuutta, lämpötilaa ja ilmankosteutta sekunnin välein [Lähteet 3].



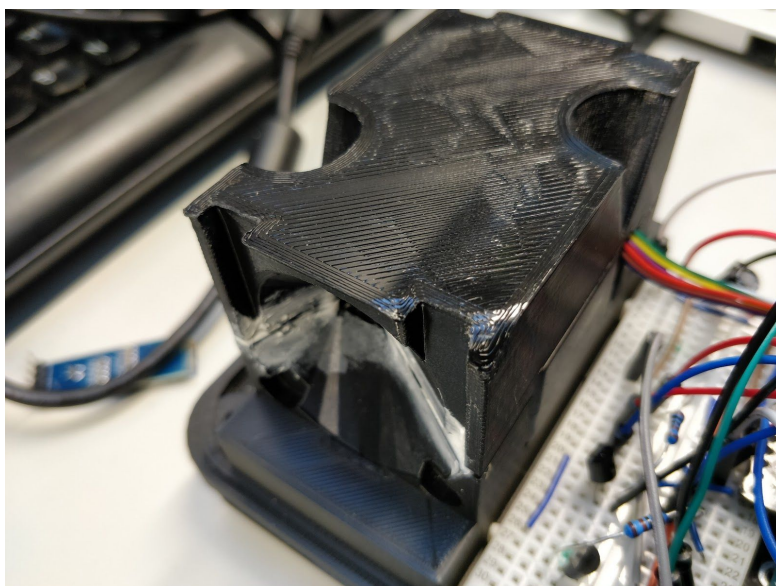
Kuva 1: Ambimate MS4 Sensor Module

-DFROBOT SEN0177 pölyanturi (Kuva 2):

Digitaalinen pölyanturi, joka käyttää UART-sarjaväylää kommunikointiin. Pystyy mittaamaan eri kokoisia pölyhiukkasia: 0.3~1.0, 1.0~2.5 ja 2.5~10 mikrometrin halkaisijan partikkeleita. Anturin vasteajan on ilmoitettu olevan alle 10 s [Lähteet 2].



Kuva 2: Partikkelianturi SEN0177



Kuva 3: Partikkelianturi SEN0177 supilokuorensa sisällä

4.3. *Perusteet anturivalinnoille*

Päädyimme Ambimate anturimoduuliin, sillä se sisältää pölyanturia lukuunottamatta kaikki anturit, joita laite tarvitsee. Valmis moduuli helpottaa huomattavasti piirisuunnittelua, josta kenelläkään ryhmän jäsenistä ei ole paljoa kokemusta. Anturimoduulin käyttöönotto oli suhteellisen yksinkertaista, joten aikaa ei mennyt turhaan säätäessä useampaa erillistä anturia.

Testasimme kahta erilaista partikkelianturia, joista yksi oli digitaalinen ja toinen analoginen. Analogisen anturin, Optical dust sensor GP2Y1010AU0, kanssa oli paljon ongelmia, emmekä saaneet siitä pölymittauksen kannalta riittävän hyödyllistä dataa. Vaikutti, että anturi pystyi vain mittaamaan tarkkuudella: ei ollenkaan pölyä, todella paljon pölyä. Analogisen anturin ongelmat saattoivat olla seurausta häiriöherkistä kytkennöistä tai esimerkiksi virheellisestä ADC-muunnoksesta. Analoginen anturi olisi mahdollisesti saatu toimimaan kunnolla, jos sen testaamisen olisi ollut käytettävissä enemmän aikaa.

Digitaalinen anturi, SEN0177, on huomattavasti kalliimpi komponentti, mutta siitä saamme luotettavaa dataa, kunhan anturia pidetään tarpeeksi kauan päällä ennen mittaamista. SEN0177 kykenee mittaamaan 0,3 - 10 mikrometrin hiukkasia ja erottelemaan hiukkaset toisistaan. Partikkelilukumäärän ohella anturi pystyy myös ilmoittamaan mittauksen konsentraationa. Kun anturin kytkee päälle, kestää jonkin aikaa ennen kuin anturin mittaus asettuu. Jos anturia kytkee liian nopeassa syklissä päälle ja pois, sen mittadatan luotettavuus laskee huomattavasti. Digitaalinen anturi toimi kuitenkin testeissämme huomattavasti luotettavammin kuin analoginen vastine. Päädyttiin siis käyttämään digitaalista anturia.

4.4. *Anturien kehitysehdotukset*

Lopputuotetta ajatellen anturivalinnat eivät ole parhaat mahdolliset. Ambimate anturimoduuli on kalliimpi kuin erilliset lämpötila- ja ilmankosteusanturit. Erilliset anturit olisi myös mahdollista sijoittaa paremmin koteloon. Nyt esimerkiksi Ambimate valoanturi on turha, koska se sijaitsee kotelon sisällä piilossa. Ambimate-anturimoduulin havaittiin myös olevan hyvin herkkä sekoittamaan koko laitteen toiminnan, jos I2C-väylän kontaktit eivät ole hyvät. Jo liian pitkät johdot vaikuttavat aiheuttavan ongelmia kyseisen anturimoduulin kanssa.

Pölyanturi SEN0177 on myös kallis anturi, mikä pitkälti selittyy sillä, että se käyttää tiedonsiirtoon sarjaväylää ja mittaa kohtalaisen tarkasti. Kyseisen pölyanturin mitta-alue on kuitenkin rajallinen ja hyvin suuret pölymäärät aiheuttavat ongelmia. Kun pölyä on paljon partikkelimittaus ei enää toimi, mikä näkyy puskurin ylivuotona. Toisaalta anturin konsentraatiomittaus ei toimi hyvin silloin, kun pölyä on vähän.

Projektin aikana testasimme myös laajalti virransäästöä: tavoitteena oli kytkeä paljon virtaa kuluttava pölyanturi (jatkuva virrankulutus n. 80 mA) suureksi osaksi ajasta pois päältä. Anturin oma sleep-toiminto ei testeissämme toiminut kunnolla, anturilta kesti useampi minuutti palautua toimintakuntoon lepotilaan asettamisen jälkeen. Kun puolestaan yritimme kytkeä anturia pois päältä npn-transistorin avulla, anturi toimi useimmiten takaisinkytkemisen jälkeen kohtalaisen hyvin, mutta välillä se antoi selvän virhemittauksen. Anturi vaikuttaa siis toimivan luotettavasti ainoastaan, kun se on koko ajan päällä ja mittauksia suoritetaan lähes jatkuvasti.

Toteutimme prototyyppissämme lopulta pölymittauksen siten, että mittaamme noin minuutin. Sen jälkeen keskiarvoistamme datan, lähetämme sen eteenpäin ja jatkamme mittaamista. Tämä menettely maksimoi SEN0177 luotettavuuden, mutta kuluttaa myös paljon virtaa. Mahdollisessa lopputuotteessa olisi luultavasti järkevintä käyttää jotain muuta pölyanturia, joka ei vaatisi jatkuvaa päälläoloa, kuten SEN0177.

5. Tiedonvälitys

5.1. *Tiedonsiirron mahdolliset toteutustavat*

Projektin ensimmäisten viikkojen aikana pohdittiin erilaisia tiedonsiirron topologioita prototyyppille. Lähestymistapoja vaikutti olevan periaatteeltaan kaksi erilaista. Näillä toteutustavoilla on omat hyvät ja huonot puolensa.

Ensimmäinen vaihtoehto olisi toteuttaa itsenäinen mitta-asema, joka olisi suoraan yhteydessä palvelimelle ja lähettäisi omat mittatietonsa. Kun haluttaisiin kattavampi rakennustyömaan mittausta, tällaisia yksiköitä pitäisi olla useampi ja kaikki niistä tarvitsisivat verkkoyhteyden tiedon välittämiseen. Tällaisen verkkoyhteyden muodostaminen voisi rakennustyömaalla osoittautua ongelmalliseksi puuttuvan talotekniikan vuoksi. Vahvuutena tässä toteutustavassa on sen tietynlainen yksinkertaisuus: täytyy suunnitella yksi laite ja sen jälkeen valmistaa useampia näitä identtisiä yksiköitä.

Toinen vaihtoehto on toteuttaa erilliset mitta-asetat ja sen jälkeen liittää ne yksittäiseen isäntäasemaan, joka kokoaa mittatiedot sekä välittää ne palvelimelle. Tässä toteutustavassa vahvuutena on se, että ainoastaan isäntäasema tarvitsee verkkoyhteyden. Haasteena puolestaan on mahdollisesti järjestelmän laajennettavuus ja hieman monimutkaisempi toimintaperiaate. Toisin sanoen tämä toteutustapa vaatii kahden erilaisen laitteen suunnittelua, joista molempien tulee pystyä mukautumaan toimintaympäristöön eli siihen, kuinka monta mitta-asemaa järjestelmään on kytketty.

Prototyypin topologian toteutustavan lisäksi myös tiedonsiirtoon palvelimelle oli olemassa kaksi vaihtoehtoa: tavallinen Internet-pohjainen kommunikaatio ja IoT-verkko, esimerkiksi LoRaWan kautta. Työmaiden oman IoT-verkon käyttäminen antaisi luotettavan vaihtoehdon epävarmalle mahdollisuudelle saada wifi-yhteys, mutta sen suurimman ongelman muodostavat tarkat rajoitukset dataliikenteen määrässä. Tämän verkon käyttäminen vaatisi siis erittäin tehokasta tiedon pakkaamista. Kuitenkaan hyvälläkään pakkaamisella lähetyksiä ei voida tätä verkkoa käyttäen tehdä useita kertoja tunnissa ennen kuin datarajoitukset tulevat vastaan. Toinen IoT-verkon merkittävä haaste on siihen liittyvän dokumentaation vähyys, mikä luultavasti on seurausta siitä, että palvelu on suhteellisen uusi.

Ajatus IoT-verkon käyttämisestä hyllytettiin ja tiedonsiirtoon palvelimelle päädyttiin käyttämään normaalia http-request -pohjaista kommunikointia. Pääsyyinä tähän valintaan oli IoT-verkon tiukat datarajoitukset sekä IoT-verkkoihin liittyvän informaation puute.

Projekti päätettiin toteuttaa topologian osalta siten, että on erikseen mitta-asemia ja isäntäasema (toteutustapa 2 edellä). Valinnan pääperuste oli työmaaympäristön huomionti ja

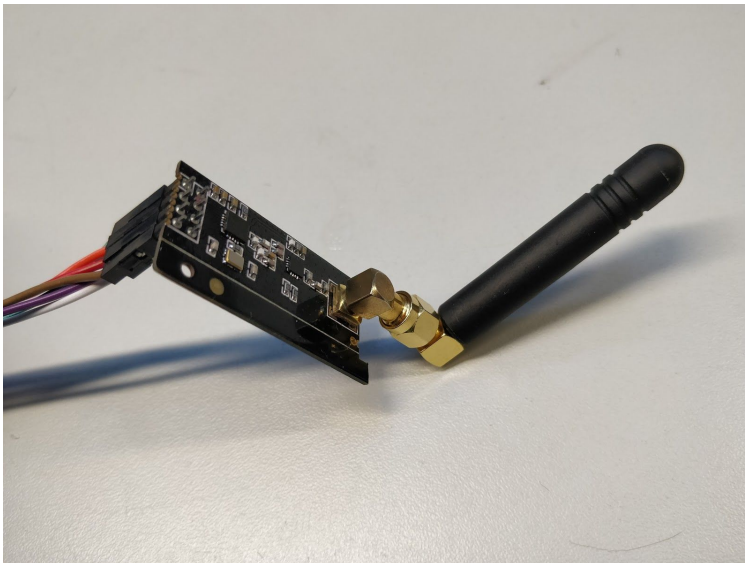
sen sisältämät verkkoyhteyden muodostamiseen liittyvät ongelmat. Haluttiin siis valita toteutusperiaate, jolla olisi edellytykset toimia käytännössäkin ilman, että pitäisi rakentaa esimerkiksi työmaan kattavaa wifi-verkkoa. Tässä toteutustavassa todellakin riittää, että vain isäntäasemalla on verkkoyhteys. Kommunikointi isännän ja mitta-asemien välillä puolestaan tapahtuu radiopiirien välityksellä.

5.2. Tiedonsiirto mitta-asemien ja isäntäaseman välillä

5.2.1. Toteutustapa

Yhteys mittausdataa lähettävän mitta-aseman ja sitä vastaanottavan isäntäaseman välillä on toteutettu nRF24L01 radiomoduulien avulla, jotka voivat sekä lähettää että vastaanottaa viestejä 2,4 GHz taajuudella. Tarkemmin ottaen käytetään radiomoduulin versiota, jossa on vahvistin ja kohinanvaimennin: NRF24L01 + PA/LNA (Kuva 4). Tässä versiossa on vahvistimen myötä huomattavasti parempi kantama kuin halvemmassa perusversiossa sekä moduuli on kohinanvaimentimen ansiosta varmatoimisempi.

NRF24L01 piireihin päädyttiin projektissa pääasiassa siksi, että niitä löytyi suoraan sähköpajalta ilman, että olisi tarvinnut tilata Kiinasta. Valitsemalla nämä piirit päästiin aloittamaan tiedonsiirron toteuttaminen varhaisessa vaiheessa ilman ylimääräistä tilausten odottelua. Lisäksi piireihin löytyi kattava valmis Arduino-kirjasto (RF24), jonka itse ohjelmoimiseen olisi mennyt hyvin paljon aikaa jo valmiiksi tiukasta projektiaikataulusta.



Kuva 4: nRF24L01 Transceiver

Kommunikoinnissa on lähdetty oletuksesta, että isäntäasema on koko ajan päällä (kytketty verkkovirtaan) ja kuuntelee jatkuvasti omaa osoitettaan. Kun uusi mitta-asema liittyy verkkoon, se lähettää isäntäasemalle id-pyyynnön, jolla se saa isännältä oman

uniikin tunnisteiden. Näin mitta-asetat voidaan erottaa toisistaan, kun tietoja lähetetään isäntäasemalta eteenpäin palvelimelle. Uniikit tunnisteet ovat juoksevia numeroita alkaen arvosta 1 arvoon 7 asti. Isäntä jakaa arvot järjestyksessä ja arvoja ei pysyvästi tallenneta mihinkään vaan aina, kun laitteet käynnistetään uudelleen, myös id:t jaetaan uudelleen.

Normaalitoiminnan aikana mitta-asetat lähettävät isäntäasemalle päivitetyn anturidatan sisältävän viestin (alla Datapaketti) kerran minuutissa. Ne odottavat isännältä kuittausta ja tarvittaessa lähettävät viestin uudelleen, kunnes isäntä kuittaa saaneensa viestin. Kuittausviesti on ennalta määritetty viesti, joka riippuu mitta-asetan id-tunnuksesta.

Dataviestit lähetetään Ascii-muodossa ja erottimena eri anturiarvojen välillä on käytetty puolipistettä. Näin isännän on helppo purkaa paketti, koska anturiarvoilla on vakiosijainnit paketissa.

Tavuu	1	2,5,8,11,13	3 - 4	6 - 7	9 - 10	12	14 - 15
Sisältö	Id	Erotin	Pölyisyys	Lämpöt.	Ilman kost.	Valoisuus	Akun jänn.
Arvo	0 - 9	;	0 - 99	0 - 99	0 - 99	0 - 9	0 - 99

Datapaketti: 16 tavua, sisältö tavuittain, toteutettu Ascii-muodossa

5.2.2. Tekniikan vahvuudet

Toteutettu tiedonsiirtotapa tukee mitta-asettien lisäämistä niin sanotusti lennosta. On siis mahdollista aloittaa yhdellä mitta-asetalla ja jälkikäteen laajentaa asetien verkkoa ilman, että aiempiin aseisiin tulee tehdä muutoksia. Pääperusteena toteutettuun ratkaisuun olikin se, että käyttäjältä ei tällä toteutustavalla vaadita monimutkaisia laitteiden konfigurointeja.

Tekniikka tarjoaa avoimessa hyvän kantaman isäntä asetien ja mitta-asettien välille. Testeissä saavutettiin luotettava noin 100 m kantama. Tämä siis mahdollistaa avoimessa hyvän laajennettavuuden järjestelmään.

5.2.3. Tekniikan rajoitteet

Radiokommunikoinnin toteutuksessa on käytetty jokaiselle mitta-asetalle omaa lähetys- ja vastaanotto-osoitetta. Tämän toteutuksen puolesta tällä hetkellä koodi tukee seitsemää mitta-asettaa yhtä isäntäasemaa kohden. Mitta-asettien määrää on kuitenkin mahdollista lisätä, mutta niiden määrän lisääntyessä myös radiokommunikointiin liittyvien virhetilanteiden todennäköisyys kasvaa.

Kommunikointi on tehty sillä oletuksella, että isäntäasema on jatkuvasti päällä. Jos isäntäasema kytkeytyy pois päältä, mitta-asetat jäävät odottamaan isännän vastausta ja kuluttavat paljon virtaa. Kun isäntäasema kytketään takaisin, tilanteen pitäisi normalisoitua aiemmin isäntään yhteydessä olleiden mitta-asettien osalta. Uusien mitta-asettien lisääminen kuitenkin sekoittaa järjestelmän ainakin nykyisessä toteutuksessa. Kun muodostetaan toimiva verkko isäntäasetan ja mitta-asettien välille, ensin siis käynnistetään

isäntäasema, ja isännän ollessa käynnissä mitta-asetat lisätään yksittellen radioyhteyteen isännän kanssa (Käyttöohjeet liitteenä).

5.2.4. Kehitysmahdollisuudet

Radiomoduulien kantaman on havaittu olevan suhteellisen huono sisätiloissa ilman näköyhteyttä asemien välillä. Seinät ja muut esteet heikentävät huomattavasti 2.4 GHz radion kantamaa. Testeissä pystyttiin välillä saavuttamaan isäntäaseman ja mitta-asetan välille ainoastaan luotettava yhden huoneen etäisyyden kantama, eli noin 10 - 15 m.

Tällaisessa toimintaympäristössä matalamman taajuuden piiri toimisi paremmin, johtuen radioaaltojen fysikaalista ominaisuuksista. Yleisestikin ottaen laadukkaampaan piiriin investointi olisi keino parantaa kantamaa. Tosin samalla myös virrankulutus luultavasti kasvaisi.

Lopputuotetta ajatellen nykyisestä rajoitteesta, seitsemän mitta-asetmaa isäntäasemaa kohden, olisi hyvä päästä eroon. Rajoitus pohjautuu siihen, että koodissa on kaikille asemille ennaltamääritetty omat osoitteet id:n suhteen. Lisäksi kuittaukseen käytetään id riippuvaisen merkkijonon välittämistä. Nämä käytännöt tekevät järjestelmästä hieman varmatoimisemman. Kuittaukset pystyisi toteuttamaan kuitenkin ilman, että ne vaatisivat ennaltamääritettyä merkkijonoa. Id riippuvaisen osoitteiden käyttäminen ei puolestaan edes lisää järjestelmän varmuutta merkittävästi verrattuna yhteen lähetys- ja kuunteluosoitteeseen.

Mitta-asetmien liittämistä isäntäasemaan pohdittiin projektin aikana ja siihen olisi olemassa useita eri toteutuksia. Nykyinen toteutus, jossa tunnisteet ja laitenumerot muuttuvat aina uudellenkäynnistyksen yhteydessä, ei sellaisenaan sovellu tuotantolaitteeseen.

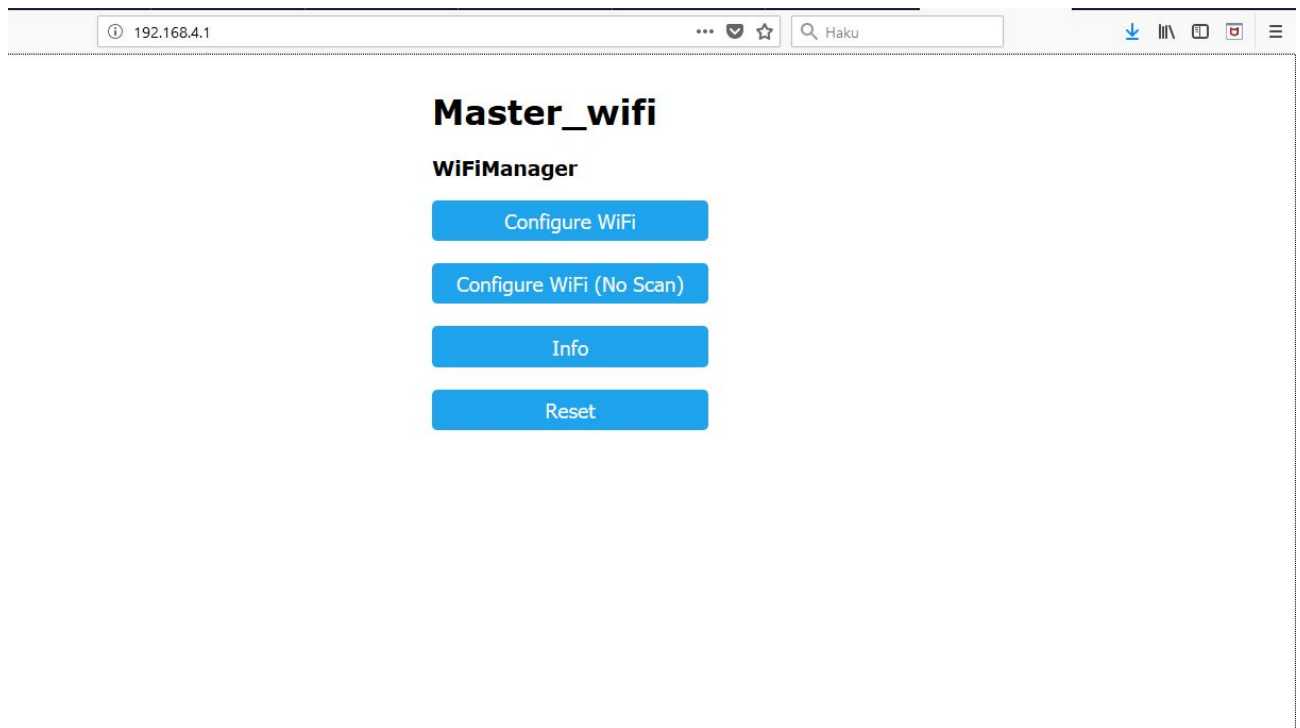
Yksi vaihtoehto olisi käyttää nykyisen tyyppistä toteutusta, mutta tallentaa ensimmäisen kerran isäntään yhdistettäessä id-tunnisteet mikrokontrollerin pysyväälle muistille (EEPROM). Tätä toteutusta varten mitta-asetmilla tulisi myös olla painike / toiminnallisuus, jolla pystyy tyhjentämään EEPROM:lta id:n sisältävän muistiosoitteen. Tällöin asemien sammuttaminen toimisi paljon järkevämmän, mutta nopeaa testausta ajatellen järjestelmä olisi monimutkaisempi.

Toinen vaihtoehto olisi asettaa id manuaalisesti käyttäjän toimesta. Mittalaitteessa voisi olla esimerkiksi konfigurointipainikkeet, joilla id valittaisiin tietystä joukosta. Tässä toteutuksessa hyödynnettäisiin lisäksi mitta-asetman näyttöä. Ongelmana toteutuksessa on tietysti se, että tällöin mitta-asetman käyttöönotto hieman monimutkaistuu käyttäjän näkökulmasta.

Kolmas vaihtoehto olisi yksinkertaisesti kovakoodata uniikki id jokaiselle laitteelle erikseen. Tällöin palvelimelle tulisi lisätä nykyistä enemmän mahdollisuuksia asettaa selitteet laitteille ja näin mahdollistaa mitta-asetmien erottaminen. Kovakoodattu id ei siis tulisi ollenkaan näkyville käyttäjälle, vaan vain laitteisiin liitetyt selitteet näytettäisiin.

5.3. Isäntäaseman ja palvelimen välillä

Isäntäasema käyttää wifi-yhteyden muodostamiseen WiFiManager-kirjastoa. NodeMcu siis pystyttää oman tukiaseman, johon yhdistämällä käyttäjä voi määrittää, mitä wifiä isäntäasema käyttää (Kuva 5) . Isäntäasema sen jälkeen tallentaa wifin ssid:n ja salasanan EEPROM-muistiin. Tarkemmat ohjeet liitteenä, Käyttöohjeet.



Kuva 5: Wifin määrittäminen isäntäasemalle

Tiedonsiirto isäntäaseman ja palvelimen välillä tapahtuu HTTP-POST -requestien avulla. Isäntä siis kokoaa mitta-asemalta saadun datapaketin merkkijonoksi, jolla on vakioformaatti. Tämän jälkeen isäntä käyttää HTTPClient-kirjastoa lähettämään merkkijonon json-headerin mukana palvelimelle.

Isäntäasema on suunniteltu siten, että se kykenee mukautumaan siihen kytketyn piirilevyn versioon. Toisin sanoen piirilevyn ensimmäisestä versiosta puuttuu jännitemittaus, joten isäntäasema ei välitä jännitetietoa palvelimelle ollenkaan. Jos jännitemittaus on puolestaan toiminnassa, isäntä välittää myös sen palvelimelle.

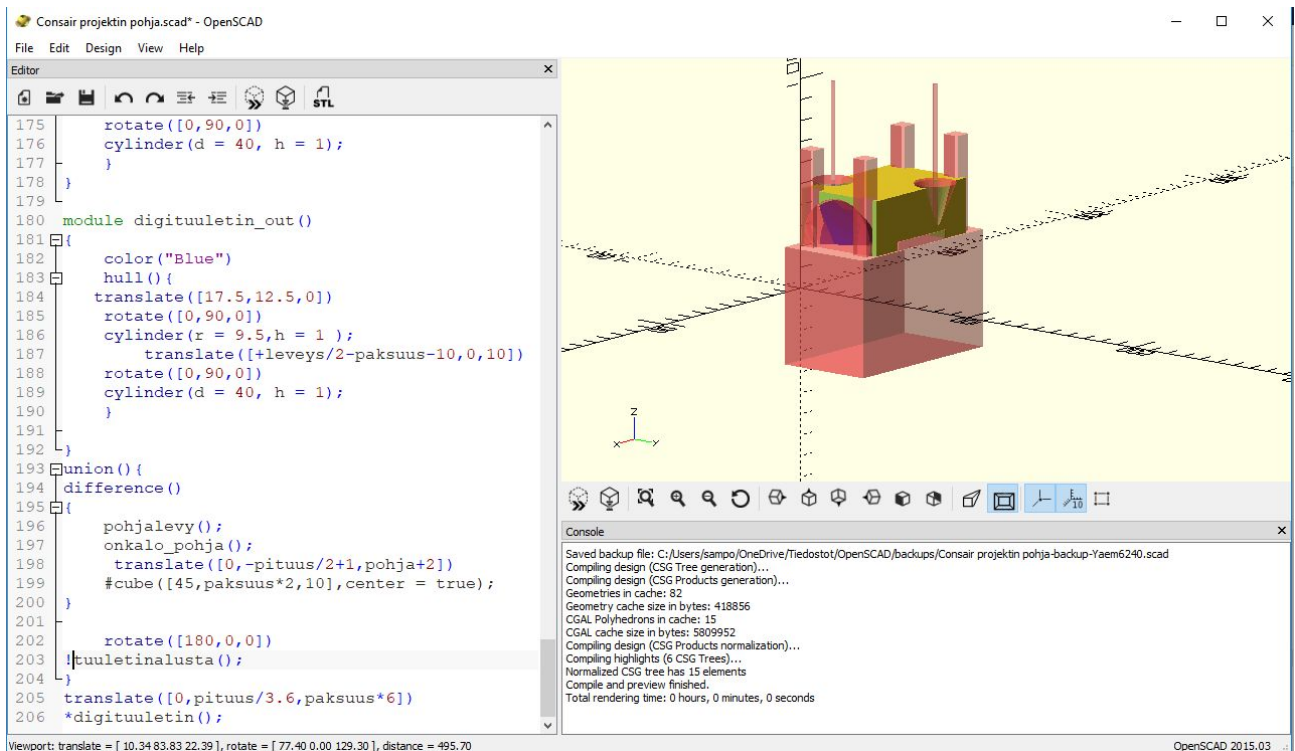
Isäntäasema pääpiirteissään toimii siten, että se koko ajan kuuntelee ja odottaa mitta-asemilta yhteydenottoja. Kun isäntä vastaanottaa viestin mitta-asemalta, se tarkastaa viestin ja välittää sen välittömästi palvelimelle. Tämän jälkeen isäntäasema jatkaa kuuntelua ja yhteydenottojen odottamista.

Toteutetulla tiedonsiirrolla päästään muutaman sekunnin vasteeseen. Mitta-aseman lähettämät mitta-arvot voidaan siis muutaman sekunnin kuluessa nähdä palvelimelle. Vasteaika on vähintäänkin riittävä järjestelmän käyttötarkoitukseen nähden.

6. 3D-mallinnus kuoren suunnittelussa ja kuoren valmistaminen

Laitteiden kuoret päätettiin jo varhaisessa vaiheessa toteuttaa pääasiassa mallintamalla ne 3D-ympäristössä ja tulostamalla 3D-printterillä. Vaikka kuoren olisi voinut toteuttaa myös laserleikkaamalla ne akryylista tai massalevystä, olisi tällainen toteutus ollut vähemmän joustava muotoilun suhteen ja hankalampi saada pölyn kestäväksi.

OpenSCAD osoittautui mallintamiseen sopivaksi työkaluksi. Kyseisellä ilmaisella ohjelmalla kappale kuvataan koodinomaisella syntaksilla. Ohjelmalla on puutteensa, kuten vuorovaikutteisuuden puuttuminen, mutta oikein tekemällä sillä tehdyt kappaleet on helppo parametrisoida.



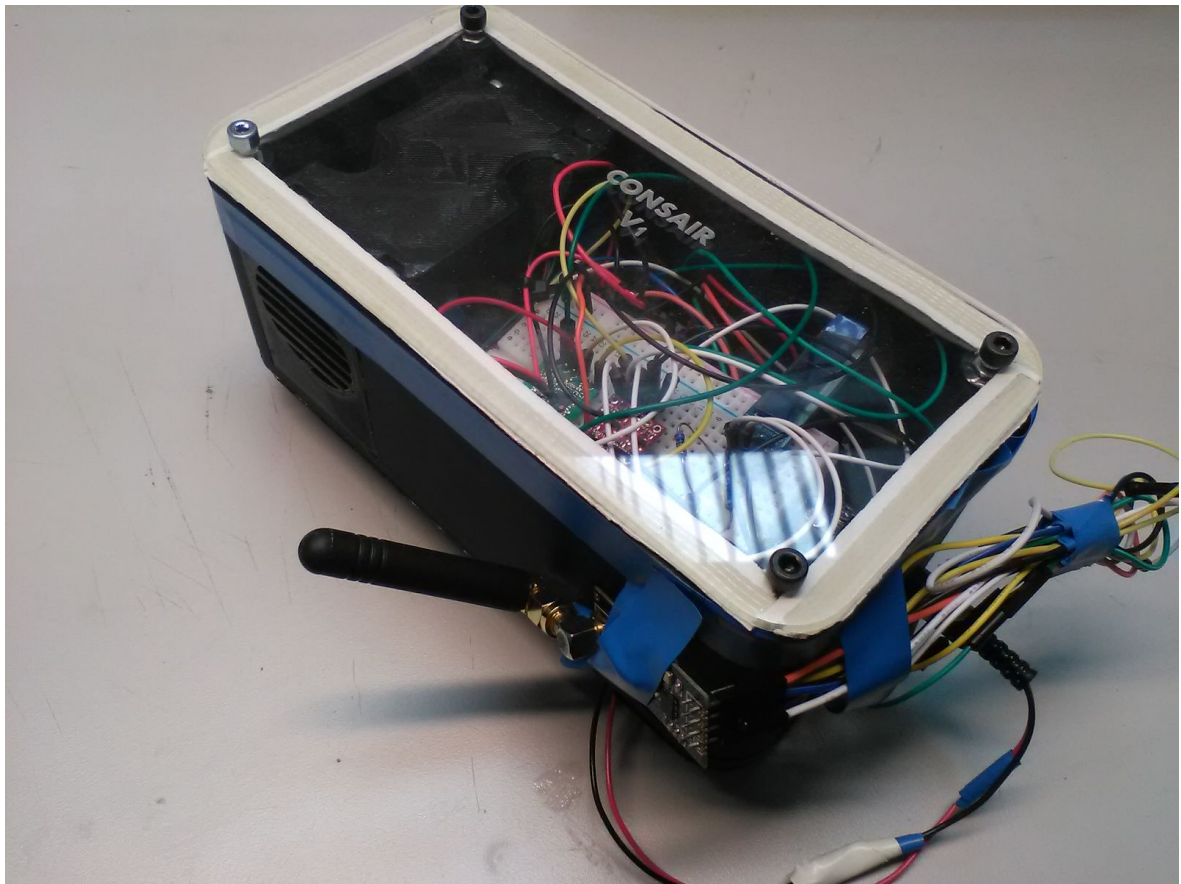
Kuva 6: Esimerkinäkymä OpenSCAD-ohjelmasta

OpenSCADilla tehdyt kappaleet muunnettiin tämän jälkeen stl-tiedostoiksi ja tulostettiin Ultimaker-tuotepöydän tulostimilla käyttäen Cura-ohjelmaa. 3D-tulostuksen rajoitteet täytyy pitää mielessä. Tulostin ei voi tulostaa tyhjän päälle, mikä rajoittaa ilman tukimateriaalia tehtyä tulostusta, minkä lisäksi tulostuksessa syntyvät epätarkkuudet pitää huomioida toleransseissa.

Suunnittelussa logiikka ja muut häiriöherkät komponentit pitää suojata ulkoilmalta. Siksi antureille tuleva ilma kuljetetaan tuulettimen ohjaamana laitteen läpi eristettyä käytävää pitkin. Toinen vaihtoehto olisi ollut sijoittaa anturit laitteen ulkopuolelle, mutta tällainen ratkaisu ei olisi

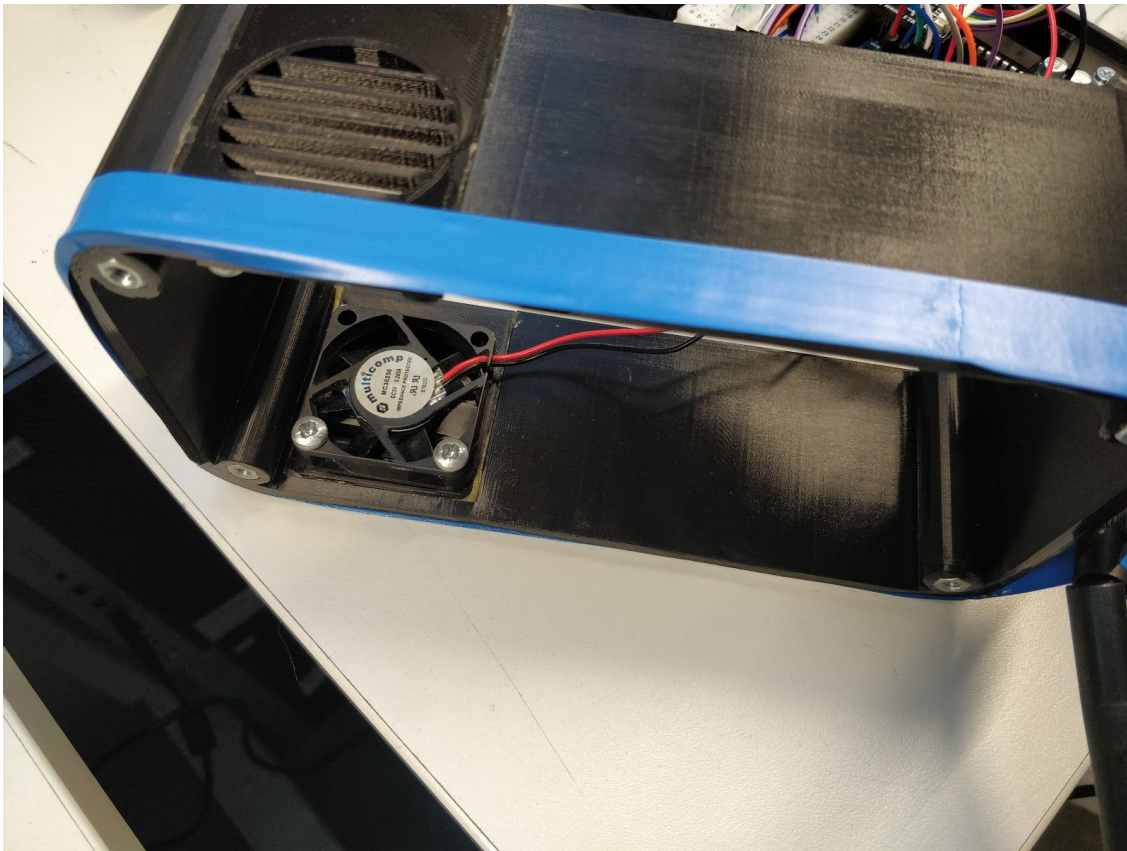
ollut mekaanisesti kestävä. Kasauksessa käytettiin pääasiassa ruuveja ja kuoreen liimattuja muttereita, sekä vähemmässä määrin kuumaliimaa ja teippiä.

Kurssin aikana mitta-aseman kuori suunniteltiin käytännössä alusta asti kolme kertaa (nimetty V1, V2 ja V3, piirilevyn numerointi ei noudata samaa logiikkaa, V1 virtapiiri päätyi lopulta V2 kuoreen jne.). Ensimmäinen, väliaikaiseksi suunniteltu kuoren versio, V1, oli suunniteltu pitämään sisällään Arduinon ja anturien lisäksi lähinnä pienen leipälaudan muita komponentteja varten, sekä kuoren ulkopuolisen, pohjaan kiinnittyvän paristokotelon. Tämä osoittautui ajan kuluessa riittämättömäksi tilan suhteen näytön ja muiden lisäyksien jälkeen ja radiopiiri antennineen sijoittui lopulta kuoren ulkopuolelle. Tämä oli myös ainoa kuori, missä kokeiltiin pölyn viemistä pölyanturille suppilonmallisella ratkaisulla. Käytännössä tilaa ei jäänyt muille antureille. Myöhemmissä versioissa päädyttiin lieriömallisiin tunneleihin.



Kuva 7: V1 -kuori lopullisessa ulkoasussaan

V1 -kuoressa myös hyödynnettiin laserleikattua akryyliä enemmän kuin muissa kuorissa. OpenSCAD -ohjelmassa sai helposti tehtyä kuorta mukailevan 2D -mallin, jonka pystyi viemään SVG-tiedostona Inkscape ohjelmaan.



Kuva 8: Kuoren ulkoseinä tuulettimien kanssa

V2 -kuoressa päädyttiin kokeilemaan kerrosmaista rakennetta, jossa anturit on sijoitettu “laudan” toiselle puolelle ja muut komponentit toiselle. Koko komeus voidaan sitten ripustaa laudassa olevasta renkaasta. Kuoren suunnittelussa pyrittiin huomioimaan kestävyys ja käytettävyys paremmin kuin aiemmin. Printtimateriaalina kokeiltiin tavallisen PLA-tulostinmateriaalin lisäksi vähemmän haurasta tough-PLA:ta. Näyttö, akku ja virtakytkin huomioitiin alusta lähtien suunnittelussa ja radiopiirin kiinnitystä tutkittiin lukuisilla koeprinteilla. Silti printtaamiseen kuluvan ajan ja aikatauluongelmien takia osia kuoresta jouduttiin suunnittelemaan ja tulostamaan vailla tarkkaa tietoa piirilevyn ja latauspiirin dimensioista.. Lisäksi anturien puolella kotelosta oli paljon tyhjää tilaa, lähinnä tuulettimien koon takia. Lisäksi akun leveyden takia kuoresta piti tehdä leveä. Näistä syistä kuori on melko kookas ja kestävä mutta osittain tyhjä. Kuoren USB-symbolin tekeminen vaati dxf-tiedoston tekemistä inkscapeissa.



Kuva 9: V2 -kuori. Koostuu valkoisesta keskilaudasta ja mustista kuoren puolikkaista. Ripustusrenkas sijaitsee laitteen toisella puolella

V3 -kuori tehtiin kompaktimmaksi kuin V2 ja ulkonäköön kiinnitettiin enemmän huomiota. Kuori suunniteltiin verrattain nopeasti ja koodi on tästä johtuen heikommin parametrisoitu ja siten vaikeaselkoisempaa kuin aiemmilla kuorilla. Pyöreä muotoilu aiheutti omat ongelmansa ja suunnitelmissa olleista jaloista piti luopua tilanpuutteen vuoksi. Lisäksi virtakytkimen ja radiopiirin kiinnityksissä tapahtui virhearvioita ja sekaannuksia teknisistä syistä, minkä vuoksi antenni ja virtakytkin piti vaihtaa pienemmäksi ja sovittaa kuoreen. Laitteen rengasmaisen keskiosa piti tulostaa käyttämällä apuna veteen liukenevaa PVA-tukimateriaalia. Jo V2 -kuoren tunnelimainen alaosa osoittautui työlääksi tulostaa tavallisilla tukiratkaisuilla materiaalin poistamisen hankaluuden vuoksi ja vielä kapeampi V3:n tunneli olisi ollut kapeampana ja ahtaampana käytännössä mahdoton tulostaa samanlaisella metodilla. Lisäksi keskiosa tulostettiin kahdessa osassa, osaksi ajan säästämiseksi, osaksi oppimistarkoituksessa. Laitteeseen tarkoitettua piirilevyn hajottua kuorta ei koskaan päästy testaamaan tositarkoituksella ja lopulliset testit piti suorittaa V2 -kuoren laitteella.



Kuva 10: V3 -kuori

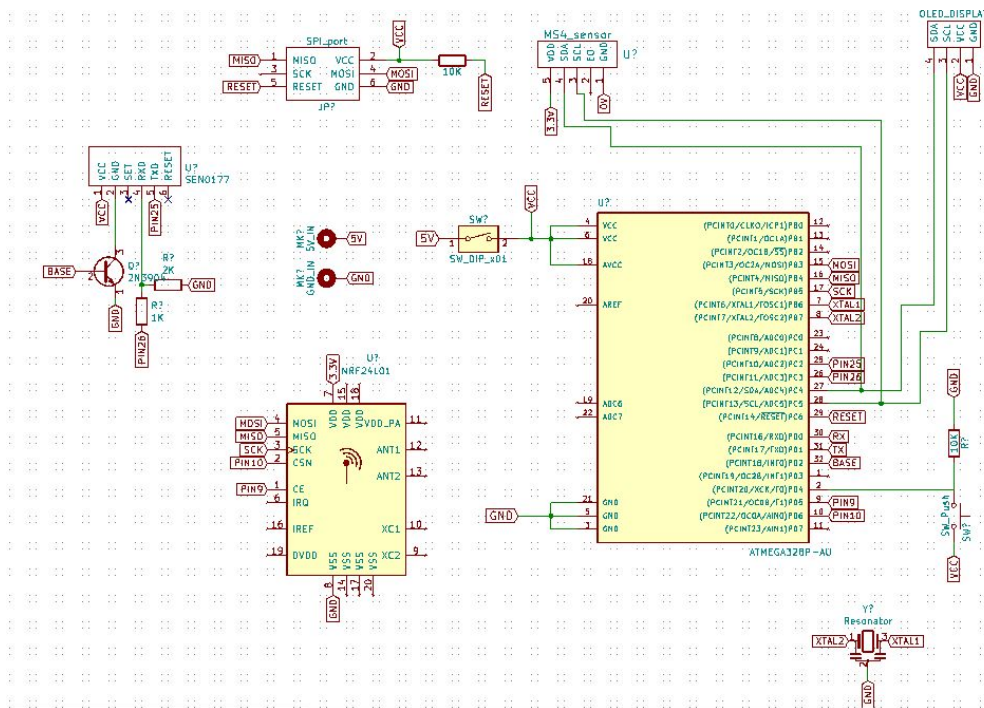
Loppuvaiheessa myös isäntäasemalle päätettiin tehdä kuori. Tämä on luonteeltaan lähes kosmeettinen laitteen luonteen vuoksi. Laitte koostuu mikrokontrollerin lisäksi käytännössä radiopiiristä ja antennista. Silti huomiota herättävä ulkonäkö ja kestävyys olisi eduksi ja yksinkertainen pyramidi -mallinen kuori syntyi verrattain nopeasti.



Kuva 11: Isäntälaitteen kuori

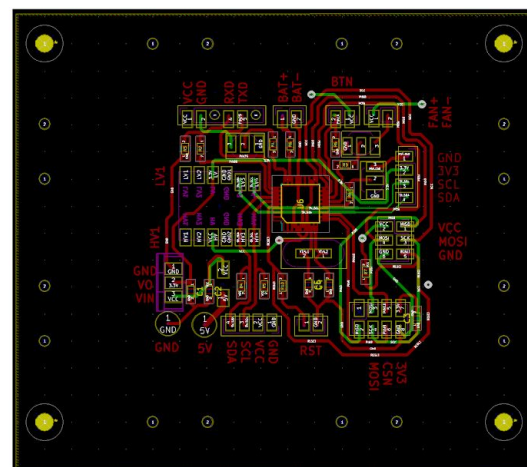
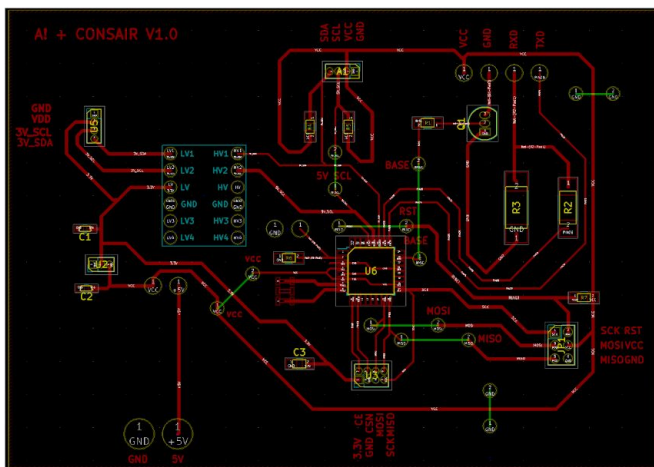
Jälleen kerran kuoressa oli joitain lastentauteja ja radiopiiri piti kiinnittää teipillä. Kotelo myös käytännössä estää mikrokontrollerin virtaledien havainnoinnin, mikä haittaa käyttöä.

7. PCB:n suunnittelu ja valmistus



Kuva 12: Mitta-aseman piirikaavio

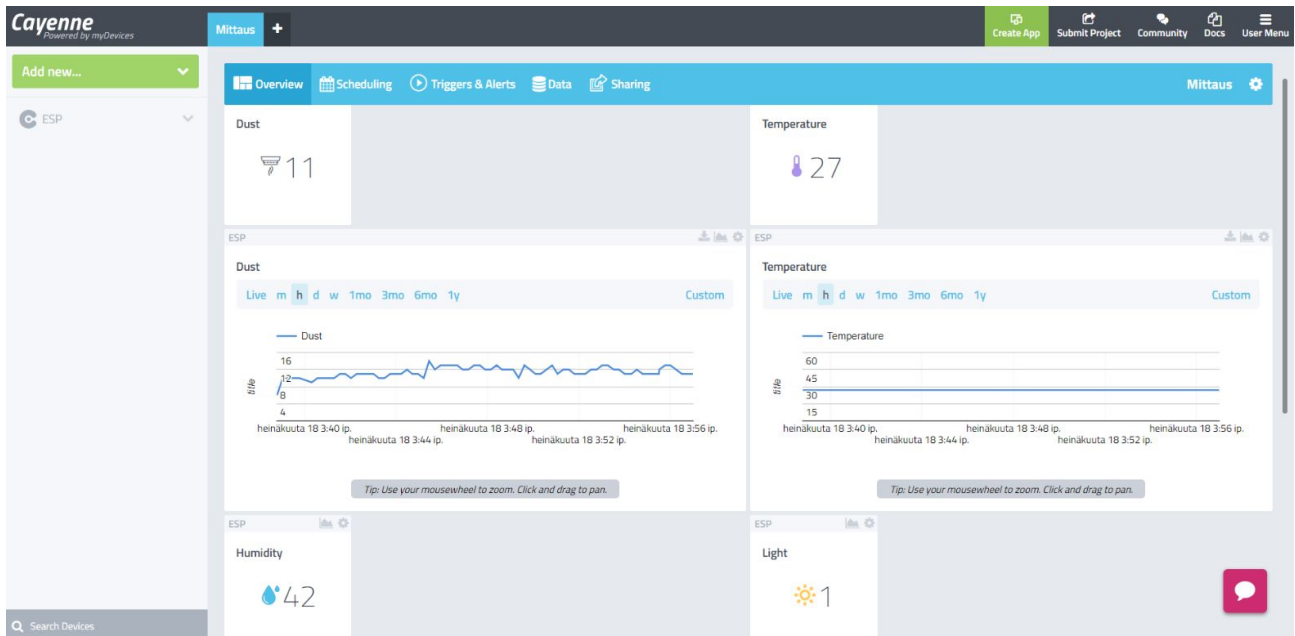
PCB suunniteltiin leipälautaversion pohjalta kun kaikki komponentit oli saatu toimimaan yhdessä. Mikroprosessorina käytettiin Atmega328p-Au:ta jota myös Arduino käyttää. Tämän avulla päästiin mahdollisimman pienillä muutoksilla koodin kanssa eteenpäin. Ensimmäinen versio piirilevystä oli yksipuoleinen ja siinä ilmeni ongelmaksi 20MHz oskillaattori, joka oli sensoreille liian nopea. Kun oskillaattori vaihdettiin 16MHz kiteeseen, piiri toimi toivotulla tavalla. Piirilevystä suunniteltiin vielä toinen, kaksipuolinen versio, jolloin komponentit saatiin pienempään tilaan. Tämän version kanssa ilmeni paljon ongelmia, joiden selvittämiseen ja korjaamiseen meni odotettua enemmän aikaa. Ongelmia syntyi ensin jo suunnitteluvaiheessa, kun kicad kaatui ja pcb-tiedosto korruptoitui, jolloin 15h työ täytyi aloittaa lähes alusta. Syövytyksessä muutamat data- ja sähkölinjat syöpyivät pois ja muutama linja katkesi todennäköisesti huonon levyn takia. Pcb saatiin lopulta toimimaan. Se tosin myös hajosi juuri päivää ennen demotilaisuutta.



Kuva 13: piirilevyn ensimmäinen versio (vasemmalla) ja toinen versio (oikealla)

8. Web

Alussa päädyimme käyttämään kolmannen osapuolen (cayenne.mydevices.com) tarjoamaa palvelua datan esittämiseen verkossa. Palvelun käyttö oli helppoa ja se myös tarjosi riittävästi ominaisuuksia alkuvaiheen testaamista silmälläpitäen. Halusimme kuitenkin ottaa haasteen oman palvelun suunnittelusta ja toteutuksesta.



Kuva 14: Mitta-aseman data Cayenne-palvelussa visualisoituna

8.1. Palvelin

Päädyimme käyttämään GitHubin korkeakouluopiskelijoille tarjoamaa virtuaalipalvelinta DigitalOceanilta. Virtuaalipalvelin on käytännössä Debian-pohjainen tietokone, jolla on kiinteä IP-osoite ja jonka palomuurista on aukaistu muutama portti http-pyyntöjä varten. Debianin päälle asensimme Apache2 HTTP-palvelinohjelman.

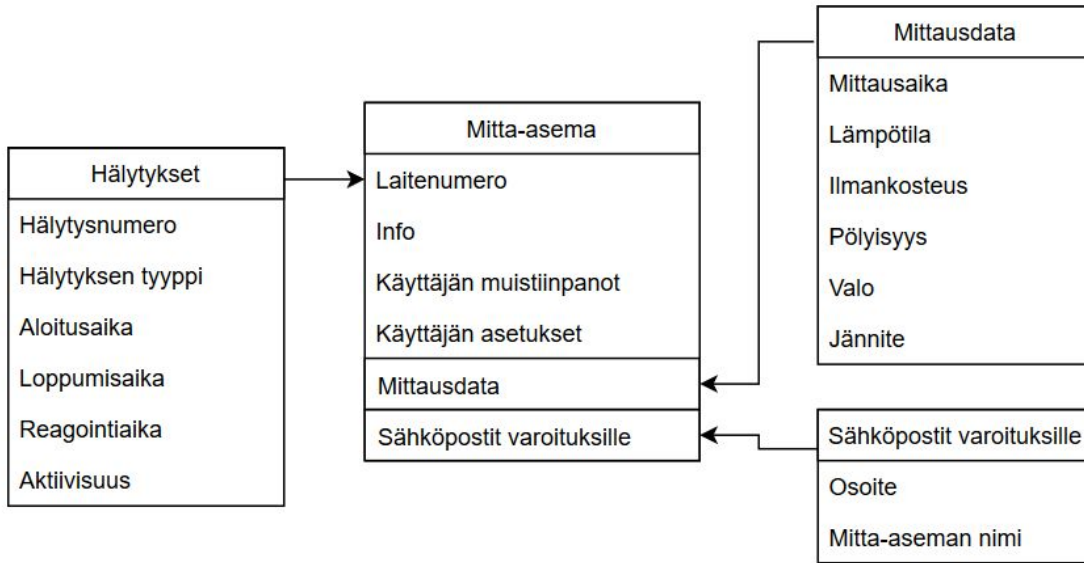
Ohjelmistokehykseksi valitsimme Django, koska kaikilla ryhmän jäsenillä on kokemusta Python-ohjelmoinnista ja Django on tällä hetkellä suurin Python-pohjainen webbikehys. Django käyttää oletusarvoisesti SQLite-tietokantaohjelmaa, joka oli meidän dataliikenteellemme riittävän kattava.

Apachen ja Django yhteistoiminta on melko yksinkertaista. Apache vastaanottaa HTTP-pyyntöjä clientiltä (esimerkiksi asiakkaan selain) ja ohjaa pyynnön Djangoille. Django lukee pyynnön ja erottelee siitä muun muassa osoitteen, kuten `www.domain/charts/`, jonka jälkeen se tekee osoitteelle määritellyt toimenpiteet ja palauttaa lopulta HTML-sivun clientille.

Varoitusjärjestelmän luonnissa käytimme hyväksi Django (sekä Apachen) ominaisuutta ohjata palvelimelle saapuvia pyyntöjä muualle työstettäväksi. Varoitusten käsittelyyn käytimme Celery-ohjelmaa. Celery mahdollistaa useiden varoitusten yhtäaikaisen prosessoinnin taustalla (daemon-prosessi) ilman suurta lisäkuormaa Djangoille.

Palvelimen toiminnan peruseriaate on yksinkertainen: Palvelin vastaanottaa isäntäaseman lähettämää dataa HTTP-pyyntönä. Data puretaan JSON-muotoisesta paketista ja tallennetaan tietokantaan. Tietokannan päivittyessä myös nettisivuilla olevat mittausarvot sekä niistä

piirrettävät graafit päivittyvät ilman että käyttäjän tarvitsee manuaalisesti päivittää sivua. Käyttäjä pystyy selaamaan eri mitta-asemien mittausdataa joko taulukkomuodossa tai graafeina. Käyttäjä voi hallinnoida mitta-asemien lisätietoja, hälytyksiä, sekä niiden raja-arvoja ja niihin liitettyjä sähköposteja.



Tietokannan rakenne

8.2. *Nettisivut*

Sivusto koostuu kolmesta pääosiosta: kirjautumisesta, datan visualisoinnista ja hälytysjärjestelmästä. Kirjautuminen tapahtuu pääkäyttäjän luomilla tunnuksilla. Ilman kirjautumista sivusto ei näytä mitään dataa. Sisäänkirjautumisen toteutuksessa on käytetty Django:n omaa kirjautumismallia.

Dataa visualisoidaan taulukkona ja diagrammeina, mitkä päivittyvät aina uuden datalähetyksen saapuessa automaattisesti. Taulukkomuodossa voi tarkastella joko yksittäisen sensoriaseman dataa, tai kaikkien asemien viimeisimpiä mittauksia. Mittausarvot ovat värikoodattuja: vihreä on normaali arvo, keltainen on varoitus ja punainen kertoo hälytyksestä. Diagrammit käyttävät myös varoitusarvoja. Charts-sivulta voi valita sensoriaseman, jonka diagrammeja halutaan tarkastella. Grafiikan piirtämiseen käytimme Chart.js kirjastoa.

Warnings-sivulla käyttäjä voi määrittellä mitattaville arvoille hälytysrajat, sekä sähköpostiosoitteet joihin hälytykset lähetetään. Sivulta näkee myös voimassa olevat hälytykset ja niiden lisätiedot. Kaikki hälytykset pystytään myös tallentamaan CSV-tiedostomuodossa tarkempaa tarkastelua varten.

9. Aikaansaannokset

9.1. Järjestelmän käyttöohjeet

Järjestelmän käyttöönotosta on olemassa kattavat käyttöohjeet, jotka löytyvät liitteenä, Käyttöohjeet. Käyttöohjeista löytyy myös lisää tietoa esimerkiksi tiedonsiirtoon liittyvien id-tunnisteiden jakamisesta, ja siitä miten se vaikuttaa järjestelmän konfigurointiin. Käyttöohjeista löytyy myös ohjeita yleisimpien virhetilanteiden selvittämiseen.

Myös palvelimen lyhyet käyttöohjeet ja eri sivujen kuvaukset löytyvät Käyttöohjeista.

9.2. Mitta-asetat

9.2.1. Komponentit

- (1) 3D-printattu kuori
- (2) ATmega328p mikro-ohjain (Ohjelmoitu Arduino IDE:llä SPI kautta)
 - (a) Ensimmäisessä prototyypissä Uno
- (3) DFROBOT SEN0177 pölysensori
- (4) AmbiMate MS4 -anturimoduuli
- (5) nRF24L01-radiomoduuli
- (6) 128×32 oled-näyttö
- (7) Pienikokoinen tuuletin
- (8) NPN transistoreja
- (9) TP4056 lataus- ja boosteripiiri (piirilevyllisissä versioissa)
- (10) 3.8 V Lipo-akku (piirilevyllisissä versioissa)

9.2.2. Ohjelmistot

- (1) Mitta-asettien koodit löytyvät Githubista: https://github.com/lwesterl/Protopaja_p3
- (2) Koodit MIT-lisenssin alla
- (3) Kaikki koodit, joiden nimessä *slave* ovat mitta-asetille

9.2.3. Ensimmäinen prototyyppi (V1)

- (1) Toiminut hyvin, ei havaittu ongelmia viime aikoina
 - (a) Tätä käytettiin demossa, kun muut mitta-asetat eivät toimineet
- (2) Vaatii toimiakseen paristot/akkuparistot alapuolella olevaan koteloon
- (3) Tuuletinta ei ole kytketty
- (4) Jatkuva virrankulutus n. 100 mA

- (5) Pölyanturin virtaus huono johtuen kotelosta

9.2.4. Toinen prototyyppi, valko-musta -kotelo (V2)

- (1) Korjattiin demon jälkeen
 - (a) Ambimate-anturimoduulin kontaktit olivat huonot
 - (b) Vaihdettiin koko anturimoduuli toiseen
- (2) Tällä tehtiin verrokkimittaukset 27.8, joissa pölymittausta verrattiin Trotec:iin (9.5)
 - (a) Toimi ongelmitta
- (3) Tuulettimeen ei ohjausta, vaan jatkuvasti päällä
- (4) Jatkuva virrankulutus n. 200 mA (tuuletin + boosteri lisäävät)
- (5) Boosterin epätasaista ja korkeaa syöttöjännitettä ei korjattu
 - (a) 100 μ F kondensaattori tasoittaisi paljon syöttöä
 - (b) Syöttöjännitettä saisi alennettua lisäämällä vastuksia, regulaattorilla tai vaihtamalla boosterin parempaan
- (6) Sisällä 3.8 V Lipo, akun lataus pitäisi toimia (Kuva 15)
 - (a) Laitetta voi pitää päällä tai pois päältä, kun akkua lataa
 - (b) Liittimenä Micro-USB
 - (c) Testattu ennen demoa (kun vielä silloin toimi)
 - (d) Tuolloin hajoaminen ei hyvin suurella todennäköisyydellä johtunut akun lataamisesta



Kuva 15: Micro-USB -latausliittimet

9.2.5. Kolmas prototyyppi, keltainen kotelo (V3)

- (1) Tämän prototyypin piti sisältää piirilevyn v2, jossa olisi ollut myös akun jännitteenmittaus
- (2) Piirilevy kuitenkin hajosi, kun yritettiin parannella boosteria
 - (a) AtMega328p vaihdettu
 - (b) Resonaattori vaihdettu
 - (c) Kondensaattorit tarkastettu
 - (d) Vastukset tarkastettu
 - (e) Piiri yritetty tarkastaa oikosulkujen osalta
 - (f) Silti AtMegan pinnien jännitteet eivät ole kohdallaan
 - (g) Kellosignaali ei oskilloskoopilla tarkastettuna ole kunnollinen (ei erotettavaa taajuutta)
- (3) Kotelon sisällä näyttö, latauspiiri ja kaapeleita
 - (a) Muut komponentit irrallaan

9.3. Isäntäasema

9.3.1. Komponentit

- (1) 3D-printattu kuori
- (2) NodeMCU mikro-ohjain (Ohjelmoitu Arduino IDEllä)
- (3) nRF24L01-radiomoduuli

9.3.2. Ohjelmistot

- (1) Isäntäaseman koodit löytyvät Githubista: https://github.com/lwesterl/Protopaja_p3
- (2) Koodit MIT-lisenssin alla
- (3) Isäntäasemalle suunnatut koodit alakansiossa *master_koodit*

9.3.3. Laitteen toiminta

- (1) Vaatii toimiakseen jatkuvan verkkovirran 5V adapterin avulla
- (2) Päälle laitettaessa vaatii joka kerta Wifin konfiguroinnin
 - (a) Ohjeet liitteenä, Käyttöohjeet
 - (b) Eroaa tältä osin Githubista löytyvästä koodista (erot tarkemmin Käyttöohjeissa)
- (3) Koodissa puutteelliset suodatukset mitta-aseman virhelähetyksille
 - (a) Githubista löytyvällä koodilla kykenee kuitenkin erottamaan, lähettääkö mitta-asema jännitetietoa vai ei. Osaa tilanteen mukaan välittää jännitteen tai olla välittämättä

9.4. Palvelimen ulkoasu

9.4.1. Kirjautumissivu

Protopaja Consair

Please sign in

Username
Password
Sign in

© 2018

9.4.2. Graafit

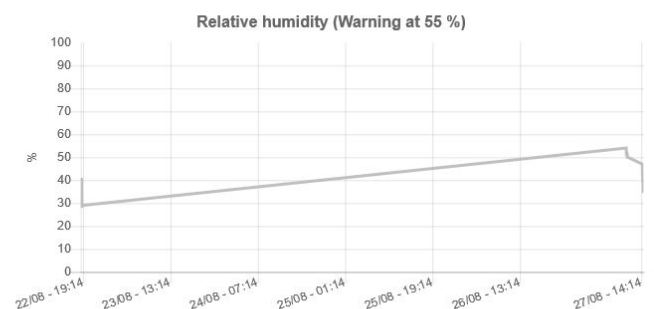
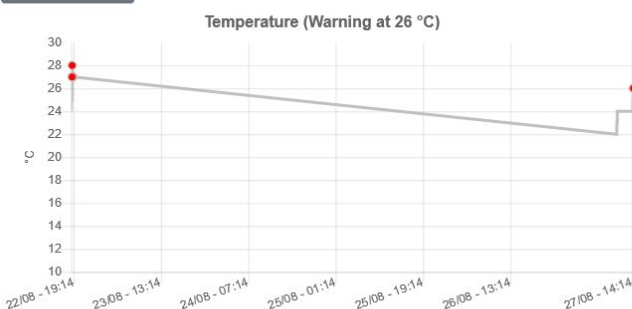
- (1) Pöly (kpl/cl)
- (2) Lämpötila (°C)
- (3) Ilmankosteus (%)
- (4) Valoisuus (suhteellinen arvo, 0 - 9)

[Home](#) [Charts](#) [Devices](#) [Incoming Data](#) [Alarms](#)

[Log out](#)

Charts

Select device ▾



9.4.3. Kaikki mitta-asetat

All devices

Only the latest data points for each device

Get further device information by clicking

Sensor station 1	
Description	Testattiin eilen
Collection time	Aug. 27, 2018, 2:41 p.m.
Temperature	25 °C
Dust particles	1 pcs. / cl
Humidity	35 %
Light	0

Sensor station 2	
Collection time	Aug. 28, 2018, 1:51 p.m.
Temperature	24 °C
Dust particles	0 pcs. / cl
Humidity	48 %
Light	0

Protopaja Consair © 2018

9.4.4. Mitta-aseman datanäkymä

Device: Sensor station 1

Time	Temperature	Dust	Humidity	Light	Battery
Aug. 27, 2018, 2:41 p.m.	25 °C	1 pcs. / cl	35 %	0	
Aug. 27, 2018, 2:40 p.m.	25 °C	1 pcs. / cl	35 %	0	
Aug. 27, 2018, 2:39 p.m.	25 °C	1 pcs. / cl	36 %	0	
Aug. 27, 2018, 2:38 p.m.	26 °C	1 pcs. / cl	38 %	0	
Aug. 27, 2018, 2:37 p.m.	26 °C	2 pcs. / cl	39 %	0	

Device Description: **Testattiin eilen**

Set description

Submit

Data points

Choose...

Remember my preference

Submit

Protopaja Consair © 2018

9.4.5. Saapuva data

- (1) Ilmoittaa saapuvan datan siitä asti, kun ikkuna avattu
- (2) Tässä tapauksessa tyhjä, koska mikään laite ei lähettänyt kuvaushetkellä

Home Charts Devices Incoming Data Alarms Log out

Device	Device info	Time	Temp (°C)	Dust (pcs. / cl)	Humidity (%)	Light
Protopaja Consair © 2018						

9.4.6. Hälytysjärjestelmä, etusivu

Home Charts Devices Incoming Data Alarms Log out

Configure alarms ▾

Current alarms

Aug. 27, 2018, 12:28 p.m.	Sensor station 2: too many dust particles!	Inactivate alarm
Aug. 22, 2018, 6:22 p.m.	Sensor station 2: temperature too high!	Inactivate alarm

Protopaja Consair © 2018

9.4.7. Hälytysten konfigurointi

- (1) Pöly (kuvassa)
- (2) Lämpötila
- (3) Ilmankosteus
- (4) Valoisuus

[Home](#) [Charts](#) [Devices](#) [Incoming Data](#) [Alarms](#)

[Log out](#)

Alarm settings

Caution

Modifying settings will affect to whole alarm system including all alarm emails

Dust Temp Humd Light

Warning: 8 pcs./d
Alarm: 20 pcs./d
Crossings: 1

Sensor station 2

All the devices your modification affects

All	▼
All	
Sensor station 1	
Sensor station 2	

Set the value, which will trigger warnings. The value must be smaller than alarm treshold, otherwise no warnings are shown

9.5. Pölymittauksen vertailu

Projektin tavoitteiden saavuttamisen analysointia varten toteutettiin pölymittauksen vertailu, jossa prototyypin toimintaa verrattiin kaupalliseen Trotec-pölymittariin. Mittaukset toteutettiin niin sanotussa pölylaatikossa, jossa hienojakoista vehnä jauhoa kierrätettiin 12V tuulettimen avulla. Laatikossa oli paljon pölyä, minkä seurauksena laitteet olivat kauttaaltaan pölyssä, kun ne poistettiin laatikosta.

Huomioitavaa prototyypin pölymittauksesta on se, että pölyä mitataan minuutin ajan ja sen jälkeen keskiarvoistetaan. Koska pölymittausdata lähetetään Ascii-koodattuna kahdella tavulla saadaan lähetettyä arvoja vain väliltä 0 - 99. Eli jos pölymittausarvo on suurempi kuin 99, lähetetään silti 99. Testeissä on havaittu, että pölyanturi kykenee valitulla skaalauksella ilmoittamaan lukumääriä ainakin arvoon 200 asti ennen puskurin ylivuotoa. Toisin sanoen tällä hetkellä ei välitetä isäntäasemalle eikä palvelimelle arvoja käyttäen koko anturin mittausskaalaa. Anturiaseman pölymittaus on lisäksi toteutettu siten, että myös ylivuoto tilanteessa lähetetään arvo 99.

Trotec:in mittarin osalta on huomioitava, että se mittaa noin 20 sekunnin jaksoissa, joten minuutin mittausdatan saamiseksi on summattu yhteen kolmen mittausjakson arvot.

Verrokkimittauksen eteneminen pääpiirteittäin:

12.26:

Aloitettiin mittaus, verrokkina Trotec

12.28:

Pölyä niin paljon, että prototyypin mitta-alue ylittyy (luultavasti skaalaus koodissa, ei välttämättä anturin ylivuotoa)

Hälytys generoitui, asetuksiin laitettu, että hälytysraja ylitettävä kahdesti ennen hälytystä

Hälytys myös sähköpostiin

12.29 - 12.32:

Tuuletin pyörii tasaisella kierrosnopeudella. Pölyn pitäisi siis kiertää laatikossa suhteellisen tasaisesti

12.33:

Pölyä kierrättävää tuuletinta alettu säätämään portaittain hiljemmalle. Pölyn määrän pitäisi alkaa vähentyä

12.40:

Tuuletin pysäytetty, laatikossa kuitenkin todella paljon pölyä, jota mitta-aseman oma tuuletin kierrättää

12.41:

Kotelon sisälle puhallettiin, aiheutti pölypiikin anturiin. Ilmankosteus nousi samalla

12.43:

Laite poistettu pölylaatikosta, asetettu pöydälle ja laitteen kotelosta ylimääräinen pöly putsattu. Tilanne normalisoituu pölyn suhteen. Hälytykseen kirjataan normalisoitumisaika

Testin jälkeen palvelimella (Kuva 16) mittadata oli nähtävissä. Siitä nähtiin selvästi, että pölyhälytysraja ylitettiin ensimmäisen kerran 12:27. Lopulta puolestaan tilanne normalisoitui 12:43. Voidaan myös huomata 12:41 piikki sekä pölyssä että ilmankosteudessa. Tällöin koteloon puhallettiin ylimääräisen pölyn puhdistamiseksi. Palvelimelta nähdään myös selvästi, että suuret pölymäärät aiheuttivat arvon 99 lähettämisen mitta-asemalta.

Time	Temperature	Dust	Humidity	Light	Battery
Aug. 27, 2018, 12:45 p.m.	27 °C	4 pcs. / cl	47 %	0	
Aug. 27, 2018, 12:44 p.m.	27 °C	4 pcs. / cl	48 %	0	
Aug. 27, 2018, 12:43 p.m.	27 °C	4 pcs. / cl	51 %	0	
Aug. 27, 2018, 12:42 p.m.	27 °C	22 pcs. / cl	54 %	0	
Aug. 27, 2018, 12:41 p.m.	27 °C	99 pcs. / cl	56 %	0	
Aug. 27, 2018, 12:40 p.m.	27 °C	45 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:39 p.m.	27 °C	47 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:38 p.m.	27 °C	42 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:37 p.m.	27 °C	44 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:36 p.m.	27 °C	49 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:35 p.m.	27 °C	51 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:34 p.m.	27 °C	63 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:33 p.m.	26 °C	87 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:32 p.m.	26 °C	99 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:31 p.m.	26 °C	99 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:30 p.m.	26 °C	99 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:29 p.m.	26 °C	99 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:28 p.m.	26 °C	99 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:27 p.m.	26 °C	69 pcs. / cl	45 %	0	
Aug. 27, 2018, 12:26 p.m.	26 °C	4 pcs. / cl	45 %	0	

Device Description: **Aloitetaan testimittaukset 12.26**

Kuva 16: Pölytestauksen jälkeinen näkymä palvelimella

Pölytestaus generoi 12:28 hälytyksen testilaitteen pölystä (Kuva 17). Hälytys generoitui toisen pölyhälytysrajan ylityksen jälkeen, kuten palvelimelle on oletuksena asetettu. Tämän takia hälytystä ei siis tullut vielä 12:27, jolloin pöly ensimmäisen kerran ylitti rajan.

Home Charts Devices Incoming Data Alarms Log out

Alarm

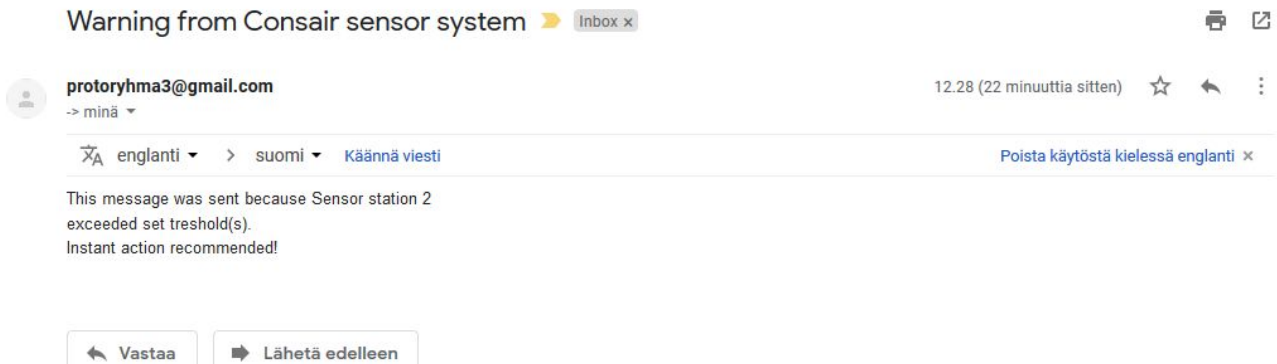
- **Sensor station 2: too many dust particles!**
- ID: 3209ead5-9aee-4572-b4e6-d2a5e9ea8a4c
- Started: Aug. 27, 2018, 12:28 p.m.
- Situation normalized: Aug. 27, 2018, 12:43 p.m.
- [For extra information, get all alarms as csv](#)

[Inactivate alarm](#)

Protopaja Consair © 2018

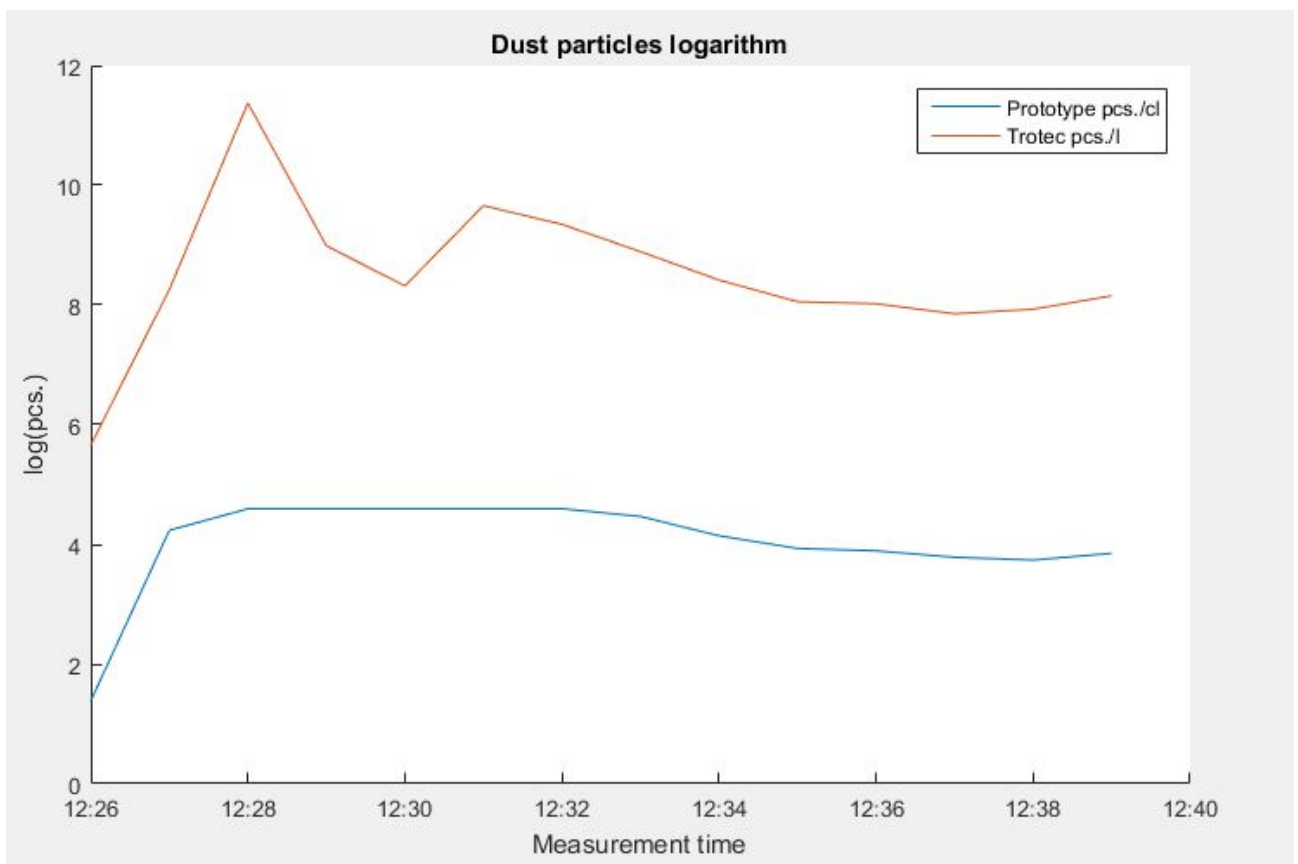
Kuva 17: Pölytestauksesta generoitunut varoitus

Samaan aikaan, kun palvelimelle generoitiin hälytysosioon hälytys, lähetettiin sähköpostiin varoitus tilanteesta (Kuva 18). Sähköpostista siis ilmenee, mikä mitta-asema hälytyksen aiheutti. Hälytyksen syytä ei tosin erikseen spesifioida.



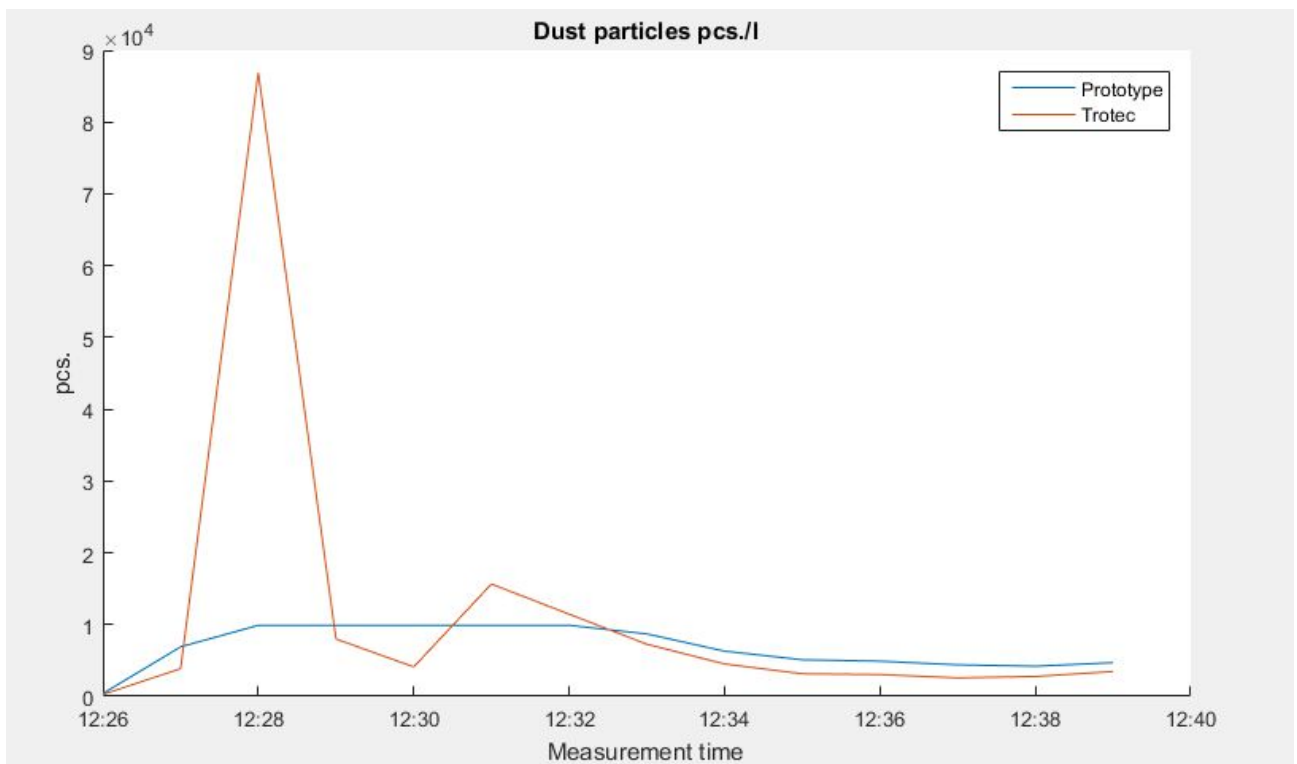
Kuva 18: Sähköpostiin saapunut hälytys

Varsinaista mittalaitteiden (prototyypä vs. Trotec) vertailua varten mittadata koottiin vektoreihin Matlabilla ja piirrettiin. Ensimmäisessä (Kuva 19) mittadata ilmaistuna logaritmisella asteikolla. Huomioitavaa se, että molemmat laitteet mittaavat yli yhden mikrometrin hiukkasten lukumäärää, mutta Trotec mittaa litraa ja prototyyppi senttilitraa kohden. Tätä eroa on yritetty balansoida logaritmisella asteikolla.



Kuva 19: Pölymittauksen logaritmi

Toinen pölymittaus kuvaaja (Kuva 20) esittää laitteiden pölymittauksen samalla skaalalla eli hiukkasten lukumääränä litrassa. Huomioitavaa on se, että prototyyppi ei kykene mittaamaan hiukkasia näin tarkasti vaan, mittadata-arvot on kerrottu 100, jotta päästään samaan yksikköön. Tämä puolestaan tiputtaa mittatarkkuutta Trotec:iin verrattuna. Kuvaajassa näkyy nyt hyvin, että prototyypin maksimiarvo on 9900.



Kuva 20: Pölymittauksen absoluuttiarvot skaalattuina, huomio prototyypin maksimiarvo, 9900

Yhteenvedon vertailusta voidaan todeta se, että prototyypin ja Trotecin mittakäyrät ovat muodoiltaan vastaavat, jos ei huomioida prototyypin skaalatun mitta-alueen rajallisuutta, eikä Trotecin liikuttamisesta johtuvaa häiriötä 12:30 kohdalla. Pienillä tai kohtalaisilla pölymäärillä prototyypin mittaus vaikuttaa melko luotettavalta, se on tasaisesti hieman suurempi kuin Trotecin. Mittauksen aikana Trotec oli tosin laatikon reunassa, joten se luultavasti altistui hieman pienemmällä pölymäärälle. Prototyypin mittatarkkuus on hyvä huomioden, että Trotec imee sisäänsä litran verran ilmaa ja mittaa siitä, kun taas prototyypin mittatilavuus on vain murto-osa Trotecista. Trotec on myös laitteena samaa kokoluokkaa kuin koko mitta-aseman prototyyppi. Pölymittauksessa on siis onnistuttu saavuttamaan sille asetut tavoitteet suhteellisen hyvin. Lopputuotetta ajatellen mitta-alueen ylärajaa pitäisi kuitenkin saada nostettua merkittävästi.

Pölymittauksen kuitenkin vaikeimmin ratkaistavissa oleva ongelma on se, että pölyn vaarallisuuden vaikuttaa pitkälti se, mitä partikkeleita pöly sisältää. Tämä voidaan nähdään hyvin siitä, että haitalliseksi tunnetuissa viitearvoissa on paljon eroavaisuuksia eri aineiden välillä [Lähteet 1]. Eli esimerkiksi asbestipöly on paljon vaarallisempaa kuin kovapuupöly. Näitä molempia on kuitenkin mahdollisesti rakennustyömaalla ilmassa. Tähän pölyntunnistusongelmaan prototyyppi ei tarjoa mitään ratkaisua.

10. Projektitoiminta

10.1. Tavoitteiden saavuttaminen

Suunnitelmassa määrittelimme projektin tavoitteet oheisella tavalla:

Tavoitteena on suunnitella ja toteuttaa rakennustyömaalle prototyyppilaitte, joka mittaa työmaan olosuhteita, lähettää datan pilveen ja raportoi olosuhteista käyttäjälle (työmaamestarille). Käyttäjälle suunnitellaan graafinen käyttöliittymä, joko mobiiliapplikaatio tai selainpohjainen. Käyttöliittymästä tulee nähdä reaaliaikaista sekä myös aiemmin kerättyä dataa työmaan olosuhteista ja käyttäjä saa ilmoituksia mikäli mittausarvot ylittävät viitearvot. Laitteen toiminnallisuutta demonstroidaan mahdollisesti kenttätestaamalla laitetta oikealla työmaalla.

Projektin lopussa voidaan todeta, että käytännössä kaikki tavoitteet saavutettiin. Saimme valmistettua kaksi mitta-asemaa, Arduino Uno-pohjaisen ja omalla piirilevyllä olevan, sekä yhden isäntäaseman. Kaikki kolme näistä asemista toimivat lopussa sillä tavalla, että pystyimme demonstroimaan toimivan konseptin. Saimme siis välitettyä mitta-asemilta dataa isäntäaseman kautta palvelimelle, jossa data esitettiin useammassakin muodossa.

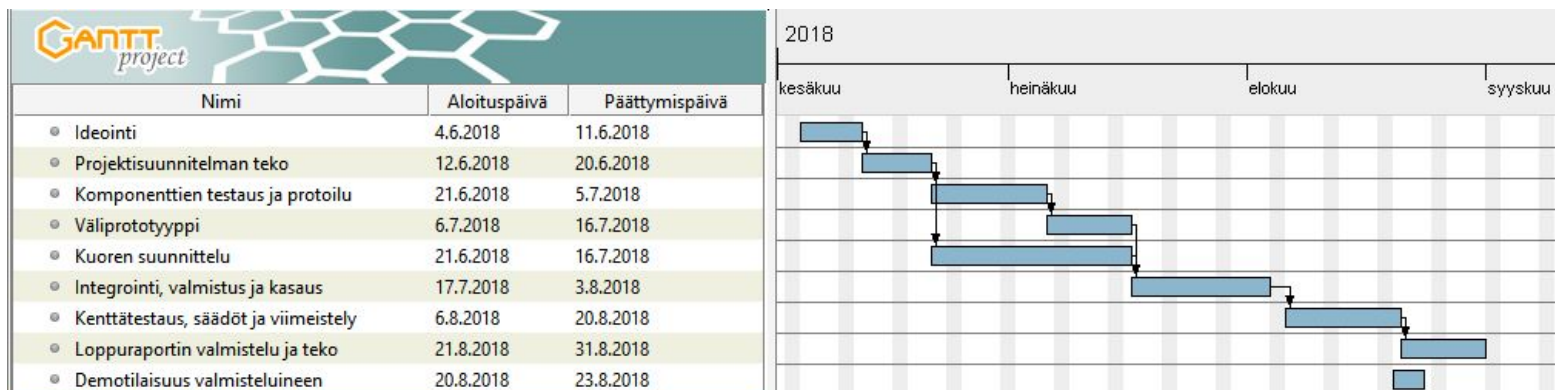
Palvelimella mittadata saadaan näkyviin sekä graafeina että taulukkomuodossa, joka helpottaa pidemmän aikavälin tarkastelua. Eri näkymät päivittyvät kaikki reaaliaikaisesti (sekunnin välein) ilman, että käyttäjän tarvitsee ladata sivua uudelleen. Graafinen käyttöliittymä toteutettiin siis selainpohjaisena. Palvelin sisältää myös hälytysjärjestelmän.

Tavoitteista poiketen hälytysjärjestelmän hälytysarvoja ei ole lyöty lukkoon, vaan järjestelmä antaa loppukäyttäjälle mahdollisuuden kustomoida asetusarvoja. Liittämällä sähköpostin palvelimen hälytysjärjestelmään, käyttäjä saa ilmoitukset viitearvojen ylittämistä, kuten tavoitteisiin oli asetettu. Tavoitteet on siis palvelimen toiminnan osalta ylitetty siinä mielessä, että käyttäjällä on enemmän mahdollisuuksia vaikuttaa palvelun toimintaan kuin tavoitteisiin oli kirjattu.

Järjestelmälle ei ehditty suorittaa kenttätestausta rakennustyömaalla kuten suunnitelmassa oli ei-pakolliseksi tavoitteeksi asetettu. Tähän kenttätestauksen puutteeseen pääsyyinä oli omalla piirilevyllä valmistuneen version myöhästyminen ja sen kanssa kohdatut tekniset ongelmat. Pystyimme kuitenkin testaamaan pölymittauksen luotettavuutta ja vertaamaan sitä kaupalliseen tuotteeseen. Lopputulokset pölymittauksesta olivat kokonaisuudessaan hyviä, jos huomio prototyypin rajallisen mitta-alueen.

10.2. Aikataulu

Projektisuunnitelmassa määrittelimme karkeasti aikataulun oheisella tavalla:



Lyhyesti voidaan todeta, että projektin alkupuolella etenimme paremmin aikataulun ja suunnitellun tehtäväjärjestyksen (kts. Projektisuunnitelma) mukaisesti. Projektin puolivälin jälkeen aikataulusta alettiin hiljalleen jäädä, ja lopputuloksena projektin lopussa oltiin esimerkiksi piirisuunnittelun osalta monta viikkoa aikataulua perässä.

Aikataulun osalta ratkaisevana käännekohtana voidaan pitää väliprototyyppiä, joka saatiin kyllä ajallaan tiedonsiirron ja anturien integroinnin osalta valmiiksi. Tämän prototyypin valmistumisen jälkeen siihen lisättiin kuitenkin vielä toiminnallisuuksia, esimerkiksi oled-näyttö, joiden toimintakuntoon korjaamiseen meni oletettua enemmän aikaa. Tästä alkoi viivästymskierre piirisuunnittelun osalta: lopullista piiriä ei voinut suunnitella kokonaan valmiiksi ennen, kuin väliprototyyppi saataisiin täysin valmiiksi sekä toimivaksi. Sitten, kun väliprototyyppi oli täysin valmis, voitiin toteuttaa lopullisen piirin ensimmäinen versio, jossa puolestaan ilmeni uusia ongelmia ja puutteita.

Piiriversio 2 valmistuminen lykkääntyi edellä mainittujen ongelmien lisäksi ilmenneiden teknisten ongelmia takia viimeiselle viikolle ennen prototyypin demoamista. Piiriversio 2 sisälsi puolestaan odottamattoman paljon erilaisia vikoja muun muassa piirilevyn syövytysprosessista johtuen. Näitä vikoja yritettiin korjata kurssin demoviikon aikana, mutta lopulta piiriä ei saatu ollenkaan toimimaan. Aikataulun mukaisesti lopullisen piirin eli tässä tapauksessa version 2 olisi pitänyt valmistua elokuun ensimmäisellä viikolla.

Myöhästymiset johtivat myös siihen, että suunnitelmassa ollut kenttätestaus rakennustyömaalla jouduttiin jättämään väliin. Sen sijaan projektin viimeisellä viikolla tehtiin pölymittauksen vertailutestit (9.2). Suunnitelmasta poiketen myös projektin loppuraportin koostaminen jäi kokonaisuudessaan viimeiselle viikolle.

Projektin työmäärä oli kokonaisuutena arvioitu suunnitelmassa likimäärin oikein. Suunnitelmassa oli kuitenkin jätetty palvelimen toteutus kokonaan auki. Aikaa oli varattu lähinnä palvelimen suunnitteluun, mutta toteuttamiseen ei melkein ollenkaan. Voi siis sanoa, että palvelimen toteuttamiseen meni esimerkiksi osa alun perin lopputestaamiseen varatusta ajasta. Toisaalta piirilevyn ja yleisesti ottaen rautapuolen debuggaamiseen ei ollut varattu tarpeeksi aikaa, mikä näkyi aikataulusta jäämisinä. Kotelon suunnitteluun menevä aika ja erityisesti 3D-tulostamiseen menevä aika oli hieman aliarvioitu. Lisäksi dokumentointiin menevä aika oli arvioitu pahasti alakanttiin.

Tiedonsiirron ja anturien osalta suunnitelma ajan käytöstä puolestaan piti hyvin paikkansa. Myös demonstraation valmisteluun menevä aika oli arvioitu hyvin kohdalleen. Arviot pitivät

luultavasti hyvin paikkansa siksi, että näitä toteutuksia pohdittiin suunnitteluvaiheessa muita enemmän.

10.3. Riskianalyysi

Projektisuunnitelmassa riskejä luokiteltiin oheisella tavalla:

Riski	Vakavuus (vaarallisuus * yleisyys), 1 - 10	Riskin huomiointi
Sairastuminen	3	Työn tasainen jakaminen läpi projektin, jotta yksittäisen henkilön sairastuminen ei aiheuta vakavia viiveitä
Komponenttien saatavuusongelmat	4	Suunnittelu siten, että käytetään vain komponentteja, joiden saatavuus on hyvä
Laiterikko (työstölaitteen hajoaminen)	4	Yleensä voidaan korvata rikkoutunut laite ja toteuttaa tarvittava tuote jollakin muulla tavalla. Lisäksi redundanssi esimerkiksi 3D-printterien suhteen on hyvä
Integrointiongelma (toteutetut kokonaisuudet eivät toimi yhdessä)	8	Yhteistyö ryhmän sisällä rajapintoja toteuttaessa. Huolellinen suunnittelu
Ryhmän osaamisen puute	8	Kysytään apua assistenteilta ja yrityksestä. Tarvittaessa muutetaan toteutustapaa helpommaksi eli alennetaan lopputuotteen kriteerejä
Budjetin ylittyminen (vakava ylitys)	2	Tiedetään, että sponsoroivaa yritystä ei haittaa pieni budjetin ylitys (eli alle 200 €). Jotta ylittäisimme budjetin vakavasti, pitäisi rahankäytön epäonnistua kokonaan, mikä estetään rahankäytön suunnittelulla
Keskeisen komponentin hajoaminen	7	Yritetään hankkia useampia komponentteja, jotta meillä olisi varaosia. Kuitenkin komponentin hajoamisesta aiheutuva selvittely vie mahdollisesti paljon

		ylimääräistä aikaa
Ryhmän kommunikaatio-ongelmat	5	Sovitaan yhteisistä kommunikaatioväylistä. Keskustellaan viikkopalaverissa kommunikaation onnistumisesta
Dokumentaatio puutteellinen	3	Dokumentoidaan kattavasti jo projektin aikana, vähintäänkin aina, kun saadaan merkittävä osakokonaisuus valmiiksi
Tavoitteista myöhästyminen	5	Realistinen suunnitelma aikataulun suhteen. Eri osa-alueista vastuussa olevat henkilöt dokumentoivat ja tarkkailevat vastuualueen edistymistä

Kokonaisuutena riskinarviointi oli onnistunut hyvin. Oikeastaan kaikki projektin aikana kohdatut ongelmat ja viivästystä aiheuttaneet tekijät oli listattu jo projektisuunnitelmassa.

Komponenttien saatavuusongelmat oli arvioitu riskinä alakanttiin siltä osin, että virheellinen Amazon-tilaus, akun latauspiiri, aiheutti jopa merkittävää viivästystä oman piirin suunnitteluun. Tämä johtui siitä, että ongelman ilmettyä ei tiedetty, mistä saadaan korvaava komponentti nopeasti, ja jouduttiin siksi käyttämään ylimääräistä aikaa vaihtoehtojen arviointiin.

Suunnitelmassa integrointiongelmat oli arvioitu täysin oikein, koska ne olivat eniten viivästystä ja ylimääräistä työtä aiheuttanut osa-alue. Piirilevyn ongelmat voisi myös luokitella tähän kategoriaan. Myös välillä parempi ryhmän sisäinen kommunikaatio olisi vähentänyt ylimääräisen työn määrää, joten se oli riskinä arvioitu hyvinkin onnistuneesti keskikastin yläpuolelle.

Suunnitelman mukaisesti budjetissa pysyminen ei aiheuttanut käytännössä mitään ongelmia. Myöskään pajan työstölaitteiden kanssa ei ollut merkittäviä ongelmia eivätkä vähäiset sairastumiset haitanneet työn etenemistä lähes ollenkaan.

10.4. Lopullinen tehtäväjako

Projektisuunnitelmassa tehtävien vastualueet oli määritelty seuraavasti:

Vastuumatriisi (X tarkoittaa päävastuuta ja O osallistumista)

	Jere Miely	Lauri Westerholm	Mikko Lähteenmäki	Sampo Lappalainen	Vikke Tirola
Dokumentointi	O	O	O	O	X
Anturit	X		O	O	O
Testaus	O	X	O	O	O
Piirisuunnittelu	X	O	O		O
Kuoren suunnittelu				X	O
Tiedonsiirto	O	O	X		
Käyttöliittymä		X	O	O	
Laitteen kasaus	O	O		O	X

Projektin aikana vastualueet muuttuivat kuitenkin hyvin merkittävästi. Muutoksen taustalla oli pääsyyinä suunnitelman jälkeiset muutokset kunkin ryhmän jäsenen projektiin käytettävissä olevassa ajassa. Projektin aikana on myös ollut tehtäviä, joissa ei ole selvästikään ollut vastuuhenkilöä, esimerkiksi dokumentointi.

Toisaalta suunnitelmassa mainittu käyttöliittymä oli epätarkka ilmaus, jolla viitattiin lähinnä palvelimen käyttöliittymään. Myös tiedonsiirto on nyt jälkikäteen järkevämpää jaoitella kahteen osaan, mitta-aseman ja isännän sekä isännän ja palvelimen välille. Ohessa vastuumatriisi, joka kuvastaa lopullista toteutunutta vastualueiden jakautumista:

	Jere Miely	Lauri Westerholm	Mikko Lähteenmäki	Sampo Lappalainen	Vikke Tirola
Dokumentointi	O	O	O	O	O
Anturit		X	O	O	
Testaus	O	X	O	O	O
Piirisuunnittelu	X	O			
Kuoren suunnittelu				X	
Tiedonsiirto		X	O		

mitta-asemalta isännälle					
Tiedonsiirto isännältä palvelimelle		O	X		O
Käyttöliittymä palvelimella		O			X
Laitteen kasaus	O	O	O	X	

11. Yhteenveto ja johtopäätökset

11.1. Henkilökohtaiset oppimiskokemukset

Jere Mielty:

Kurssi oli kaikin puolin opettavainen ja monipuolinen. Itselleni täysin uusi asia oli projektin vetäminen, josta opin paljon. Samoin piirisuunnittelu ja kicad -suunnitteluohjelma olivat täysin uusia. Sen opetteluun meni paljon aikaa, mutta opitut taidot tulevat varmasti käyttöön tulevaisuudessa. Pintaliitoskomponentteja en ollut aiemmin juottanut ja kehityin myös tällä osa-alueella huomasti.

Sampo Lappalainen:

Kokonaisuutena kurssi oli varsin mielenkiintoinen. Monet kurssilla käytetyistä työkaluista olivat joko ennestään vähän käytettyjä tai kokonaan uusia tuttavuuksia. Oma panostukseni kurssin edessä painottui hardwareen ja 3D-mallintamiseen, mutta muiden työskentelyn seuraaminen avasi hieman myös softapuolen tekemistä.

Lauri Westerholm:

Kurssilla pääsi tekemään montaa erilaista asiaa, mikä oli hyvä oppimisen kannalta. Olen erityisesti tyytyväinen siihen, että kurssin aikana opettelini web-ohjelmoinnin perusteita tekemällä osan palvelintoteutuksesta. Rautapuolen osaamisen suhteen kurssille tultaessa taitoni olivat lähes olemattomat, mutta nyt kurssin jälkeen ymmärrän selvästi paremmin komponenttien, anturien ja väylien toimintaa.

Vikke Tirola:

Keskityin kurssilla suurimmaksi osaksi palvelinpuoleen. Nettisivuprojekteissa olen ollut ennenkin mukana, mutta olen tehnyt ainoastaan frontend-puolta, joten nyt jouduin opettelemaan täysin uuden puolen funktionaalisten sivujen toteuttamisesta. Myös yhteisen koodausprojektin hallinta tuli projektin aikana tutuksi.

Mikko Lähtenmäki:

Tämän kurssin aikana on tullut puuhailtua eniten Arduino-pohjaisen ohjelmoinnin kanssa, mikä ei ole minulle sinänsä uutta eikä siten itsessään kehittänyt kykyjäni. Projektina tämä on kuitenkin ollut todella monipuolinen ja sen ansiosta olen voinut perehtyä moneen itselleni ennalta melko tuntemattomaan aiheeseen vaikka vain sivusta seuraamalla. Lisäksi tämä on ensimmäinen kunnon IoT-sovellus, jota olen ollut toteuttamassa ja tunnen erityisen hyödylliseksi osata nyt hahmottaa sellaisen rakenteen ja toteutustavat.

11.2. Johtopäätökset

Projekti oli alkuperäiset tavoitteet huomioon ottaen onnistunut. Alkuperäiseen suunnitelmaan tuli isojakin muutoksia ryhmän yhteisen tietämyksen kasvaessa, mutta muutokset olivat kuitenkin tarpeellisia, ja niiden toteuttamiseen käytetty aika näkyy myös lopputuloksessa.

12. Liitteet

Liite 1: Käyttöohjeet (pdf)

Liite 2: Projektisuunnitelma (pdf)

13. Lähteet

[1] Sosiaali- ja terveysministeriö, 23.12.2016, HTP-arvot 2016. Haitallisiksi tunnetut pitoisuudet, viitattu 30.8.2018:

http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/79109/08_2016_HTP-arvot_suomi_22122016_netti_kansilla.pdf?sequence=1&isAllowed=y

[2] DFRobot, SEN0177, viitattu 30.8.2018:

https://www.dfrobot.com/wiki/index.php/PM2.5_laser_dust_sensor_SKU:SEN0177

[3] TE connectivity, Ambimate MS4, viitattu 30.8.2018:

http://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Specification+Or+Standard%7F114-133092%7FE%7Fpdf%7FEnglish%7FENG_SS_114-133092_E.pdf%7F2314277-1

Ohjelmistolähteet

Yleiset Arduino kirjastot ja kuvaukset: <https://www.arduino.cc/>

NodeMcu:n ESP8622: <https://github.com/esp8266/Arduino>

NRF24L01-radiomoduurien kirjasto: <https://github.com/nRF24/RF24>

Oled-näytön kirjasto: <https://github.com/olikraus/u8g2>

Isäntäaseman WifiManager: <https://github.com/tzapu/WiFiManager>

Django framework-kuvaus: <https://www.djangoproject.com/>

Palvelimen käyttöönotto: <https://www.digitalocean.com/community/tutorials/>

Apache2 HTTP-palvelin: <https://httpd.apache.org/>

Celery: <http://www.celeryproject.org/>

OpenSCAD: www.openscad.org/

Inkscape: <https://inkscape.org/en>