

Loppuraportti

Projekti #5 Pölynhallintalaitteisto automatisoidulla tehonsäädöllä ja etädiagnostiikalla



Date: 31.8.2017

Aleksi Ranta-aho
Antero Vaarnamo
Joel Tolonen
Miika Koivurinne
Pauli Paaso

Informaationsivu

Jäsenet

Aleksi Ranta-aho
Antero Vaarnamo
Joel Tolonen
Miika Koivurinne
Pauli Paaso

Projektipäällikkö

Aleksi Ranta-aho

Sponsoroiva yritys

Consair Oy

Aloituspäivämäärä

5.6.2017

Palautuspäivämäärä

31.8.2017

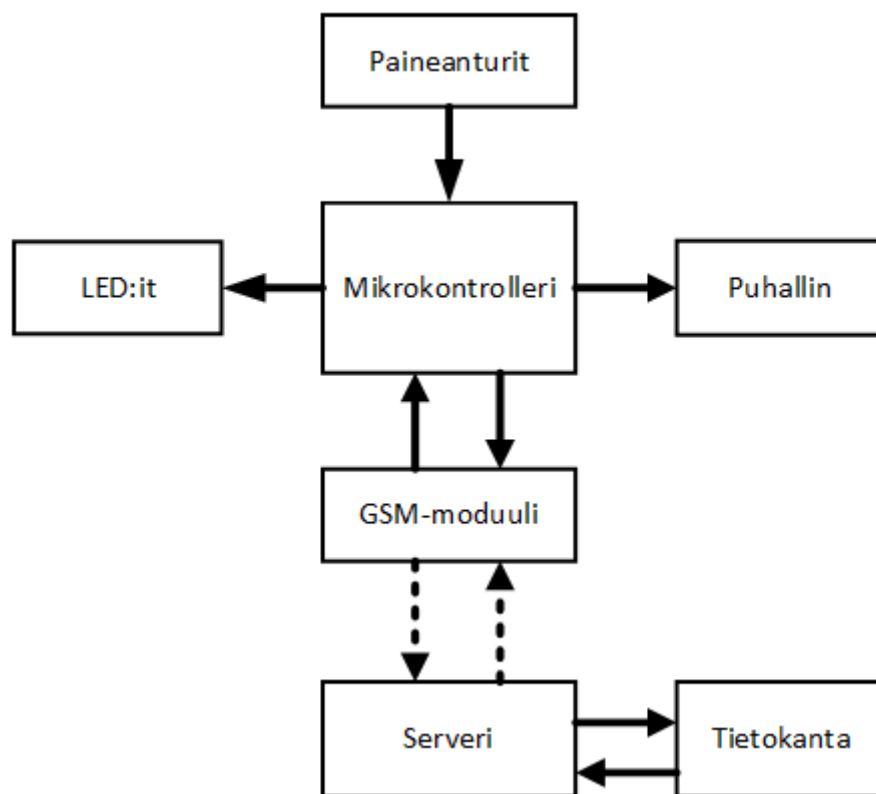
Tiivistelmä

Tämä dokumentti esittelee Aalto-yliopiston Protopaja-kurssilla kesällä 2017 oppilastyönä rakennetun kohdepoistolaitteen prototyypin. Laitteen perustoimintaperiaate on seuraava: laitteessa oleva puhallin imee ulkopuolelta ilmaa, josta pöly kerätään suodattimiin. Laitteen rakennuksessa ja suunnittelussa avusti työtä sponsoroiva yritys.

Suunnitelmat aloitettiin yrityksen toiveiden pohjalta. Laitteen tekniset vaatimukset määrittivät laitteen yleisen rakenteen ja tarvittavat osat. Laitteen rakennus aloitettiin projektisuunnitelman ja 3D-mallin laatimisen jälkeen.

Kohdepoistolaite yhdistää ohjelmistoja ja laitteistoja. Puhaltimen tehoa säädetään mikrokontrollerilla. Paineanturit mittaavat paine-eroa suodattimien yli ja näistä lasketaan säätöarvo puhaltimelle. Laite lähettää GSM-moduulin välityksellä tiedon laitteen käytön ajankohdasta palvelimelle, missä tieto tallennetaan tietokantaan. Tukkeutuneista suodattimista ilmoitetaan sähköpostilla laitteen vastuuhenkilölle.

Alla olevassa lohkokaaviossa (kuva 1) on kuvattu laitteen toimintaa yleisellä tasolla.



Kuva 1. Laitteen lohkokaavio

Abstract

This document presents the prototype of a local air cleaner built at the Aalto University Protopaja course in the summer 2017. The basic operating principle of the appliance is as follows: The fan in the device draws air from the outside and the dust is collected on filters. The sponsoring company assisted with the design and building of the device.

The plans were started based on the company's wishes. The technical specifications of the device determined the general structure and the necessary parts of the device. The construction of the device began after the project plan and the 3D model were drawn up.

The local air cleaner combines software and hardware. The fan power is adjusted by the microcontroller. A pressure sensor measures the air velocity and the microcontroller calculates a new adjustment value for the fan based on the measurement. Pressure sensors measure the pressure drop over the filters. This is used to determine if the filters are clogged. The device transmits information via GSM module to a server, where the data is stored in a database. The person in charge of the device will be notified of clogged filters by email.

Sisällysluettelo

Tiivistelmä	3
Abstract	4
Sisällysluettelo	5
1. Johdanto	6
2. Tavoite	6
3. Mekaniikka	7
3.1. Rakenteen suunnittelu	7
3.2. Rakentaminen	9
3.3. Letkut ja johdot	12
3.4. Maalaaminen	14
3.5. Ilmavirrat	14
4. Laitteisto	15
4.1. Puhallin	15
4.2. Virtalähde	15
4.3. Piirilevyt	15
4.4. Anturit	19
4.5. GSM-moduuli	21
5. Ohjelmistot	22
5.1. Mikrokontrollerin softa	22
5.1.1. Paineenmittaaminen	23
5.1.2. Moottorinohjaaminen	23
5.1.3. Datan lähetys	25
5.2. Virtuaalipalvelin	25
5.2.1. Datan vastaanotto ja tallennus Python-skriptillä	26
6. Nettisivut	27
7. Projektitoiminta	28
7.1. Tavoitteiden saavuttaminen	28
7.2. Aikataulu	29
7.3. Riskianalyysi	29
8. Yhteenveto ja johtopäätökset	29
Lähteet	30

1. Johdanto

Protopaja on Sähkötekniikan korkeakoulun järjestämä 10 opintopisteen kesäkurssi. Kurssilla opiskelijat suunnittelevat ja rakentavat kurssia sponsoroiville yrityksille prototyyppejä ryhmätöinä yritysten toiveiden mukaan.

Ryhmässämme on viisi jäsentä: kolme sähkötekniikan opiskelijaa, yksi automaatiotekniikan opiskelija sekä konetekniikan opiskelija.

Ryhmäämme tukeva yritys, Consair, valmistaa tällä hetkellä yhtä tuotetta, joka on pölynhallintatyöpiste Camu 1200. Laite suodattaa rakennustyömailla kuiva-aineiden sekoituksessa ilmaan vapautuvaa syövyttävää pölyä vähentäen keuhkosairauksia ja siivoamisen tarvetta, säästäten kustannuksia rakennustyömailla. Suodattimen vaihtoväliä haluttaisiin pidentää. Laitteen ilmavirta ei pysy vakiona, vaan laskee sitä mukaa, kun suodatin tukkeutuu. Tällä hetkellä käyttäjän ainoa tapa tarkastaa suodattimen kunto on avata koneen luukku ja visuaalisesti tarkastaa tila.

Projektin kesto on 12 viikkoa. Kaksi ensimmäistä viikkoa kului projektisuunnitelman laatimiseen. Seuraavien neljän viikon aikana suunnittelimme ja rakensimme puukotelon, opettelimme käyttämään puhallinta ja antureita ja aloimme suunnitella piirilevyä. Viimeisen kuuden viikon aikana rakensimme piirilevyn, laadimme koodit serverille, tietokannalle, GSM-moduulille ja moottorinohjaukselle ja viimeistelimme kotelon sekä nettisivut.

2. Tavoite

Yritys haluaa kehittää tuotettaan. Prototyyppiin toivottiin varoitusjärjestelmää, joka kertoo kumpi laitteen suodattimista on tukossa, rikki tai väärin asennettu sekä ilmoittaa tästä käyttäjälle esimerkiksi led-valoin. Tämän tarkoituksena on pidentää suodattimien vaihtoväliä ja helpottaa laitteen huoltoa rakennustyömailla. Laitteen on oltava mahdollisimman vaivaton käyttää, sillä rakennustyöntekijät eivät halua käyttää sitä, jos se hidastaa työntekoa. Laitteen tulee olla hiljainen, jottei kuulosuojaimia tarvitse käyttää. Kun suodatin tukkeutuu, laite ilmoittaa automaattisesti suodattimen vaihdon tarpeesta sähköpostilla tai tekstiviestillä, jotta asiantunteva huoltohenkilö saadaan ajoissa paikalle.

Puhaltimen tehoa halutaan säätää suodattimen painehäviön mukaan siten, että ilmavirran nopeus pysyy vakiona kaikissa tilanteissa. Näin säästetään moottoria ja melu saadaan pidettyä kurissa. Laitteeseen halutaan myös toiminto, joka säätää laitteen pyörimään hiljaisella teholla valmiustilassa, parantaen huoneen ilmanlaatua ja nopeuttaen käynnistystä.

Tavoitteemme oli rakentaa toimiva prototyyppi, joka täyttää edellä mainitut toiveet.

3. Mekaniikka

3.1. Rakenteen suunnittelu

Suurin osa laitteen toiminnasta oli päätetty yrityksen toimesta. Saimme kuitenkin vapauden muotoilla laitteen, sekä toteuttaa pieniä mekaniikan toimintaratkaisuja.

Ilmansuodattimet, puhaltimen pyörivä impelleri ja elektroniikka tarvitsevat suojaa ulkoisilta haitoilta. Ilmavirta täytyy ohjata suodattimien ja impellerin läpi, jotta ilma puhdistuu oikein. Ratkaisu näihin haasteisiin on sopivasti muotoiltu kotelo, jonka sisälle muut komponentit kiinnitetään.

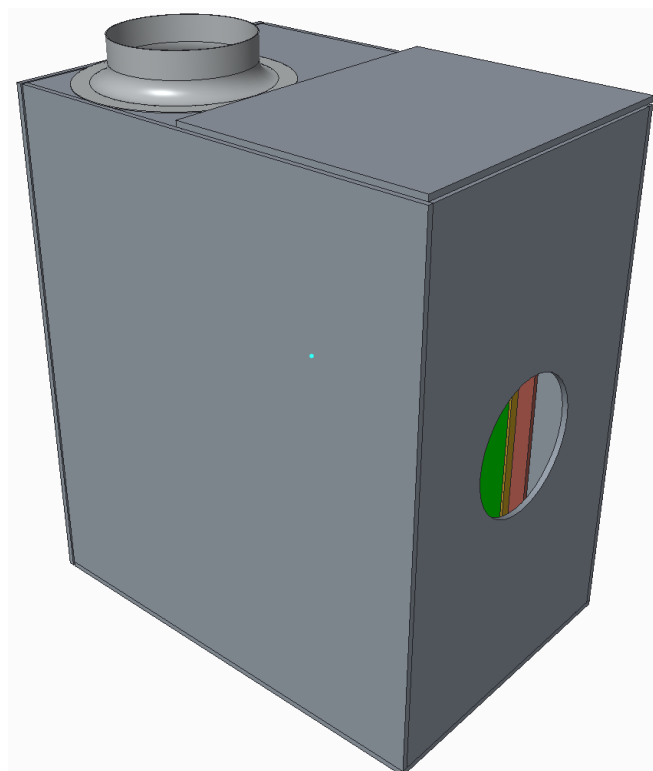
Ensimmäisissä suunnitelmissa suodattimet olivat vaakatasossa ja moottori laitteen yläosassa. Ajattelimme, että näin olisi säästetty arvokasta lattiatilaa rakennustyömailla. Ongelmaksi muodostuivat kuitenkin 90 asteen kulmat, jotka haittaavat ilmavirran kulkua. Jotta kulmilta oltaisiin välttytty, olisi tarvittu pyöreitä putkia, jotka vievät tilaa ja siten kumoavat alkuperäisen idean. Päädyimme asettamaan suodattimet pystyasentoon ja laittamaan moottorin suoraan niiden perään.

Suodattimien ilmatiiviiseen kiinnitykseen oli monta erilaista ideaa. Ensimmäiset ideat perustuivat kahteen paikoillaan olevaan kehykseen ja kahteen liikkuvaan kehykseen, joiden väliin suodattimet puristetaan ilmatiiviisti. Helppokäyttöisyyden takia usein vaihdettavan karkeasuodattimen tulee olla mahdollisimman yksinkertainen ja nopea vaihtaa. Päädyimme ratkaisuun, jossa liikkuvaan kehykseen kiinnitetään salvat, jotka voidaan kiristää kotelon ulkopuolelta. Hienosuodattimen ja karkeasuodattimen välisen tilan tulee olla ilmatiivis, joten ajattelimme kiristää hienosuodattimen liikkuvan kehyksen ruuveilla. Myöhemmin tajusimme, että tekemällä kahden suodattimen välisen kehyksen hieman pidemmäksi syvyysuunnassa sekä laittamalla tiivisteet molemmille puolille, voidaan molemmat suodattimet kiristää karkeasuodattimen edessä olevalla kehyksellä. Kehykset näkyvät ruskeina kuvassa 3.

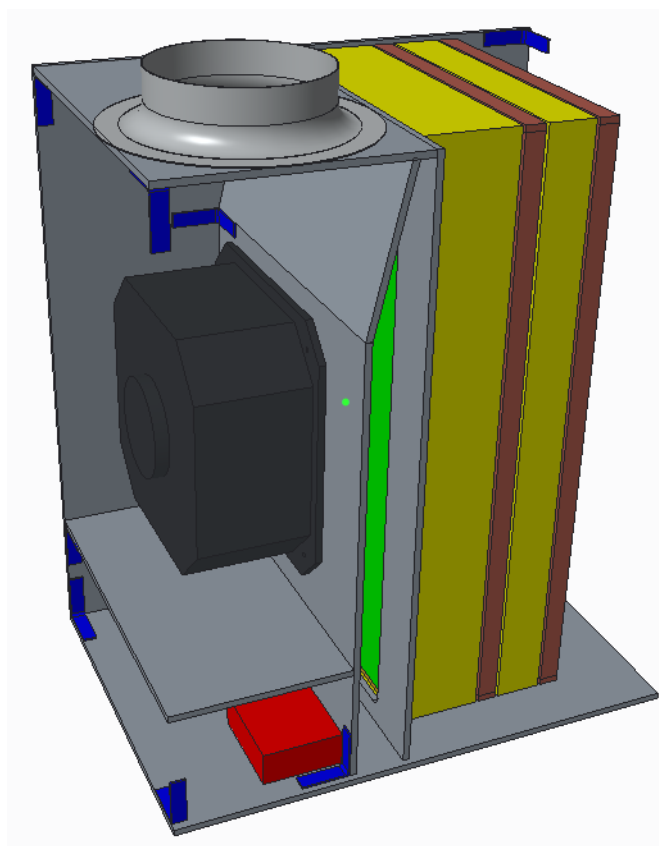
Harkitsimme kotelon materiaaliksi puuta, peltiä ja 3D-tulostettua muovia. Epäilimme, että 3D-tulostus olisi liian hidas tapa valmistaa kotelo ja osia tulisi liian monta, koska yliopiston 3D-tulostimet ovat pieniä, jolloin niillä ei voi tulostaa kokonaista seinää yhtenä kappaleena. Riittävän suurella 3D-tulostimella koko kotelo olisi kenties voitu tulostaa yhtenä kappaleena.

Päädyimme valmistamaan kotelon vanerista, sillä se on helpompaa työstää kuin pelti, mutta silti jäykempää suhteessa painoon paksuutensa johdosta. Vaneri ei myöskään johda sähköä, mikä lisää turvallisuutta, mutta toisaalta vaneri palaa peltiä helpommin. Vaneria on myös helppo liimata puuliimalla. Toisaalta peltiä voi taivuttaa kotelon kulmien kohdalta. Tilasimme 6.5 mm vaneria, koska se oli tarjolla olevista vaihtoehtoista sopivin. Materiaaliksi valittiin koivu, sillä havuvanerissa on paljon oksakohtia ja siten koloja vanerin pinnassa.

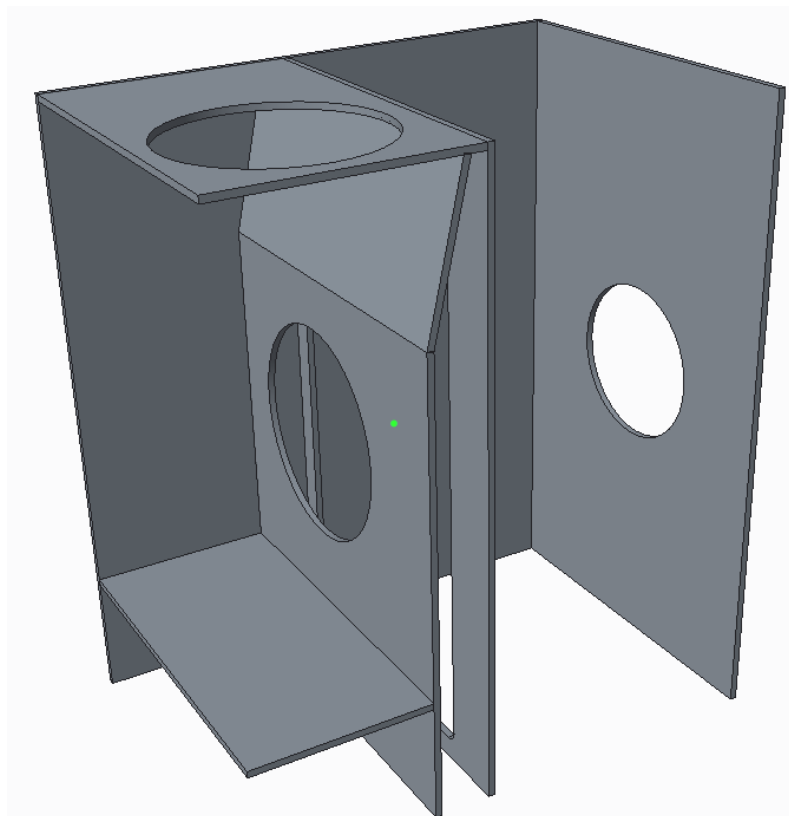
Teimme kotelosta mallin PTC Creo Parametric 3.0 -tietokoneohjelmalla varmistaaksemme komponenttien mahtumisen kotelon sisään sekä selvittääksemme leikattavien vaneripalojen mitat (kuvat 2-4).



Kuva 2. Kotelo ulkoa päin nähtynä



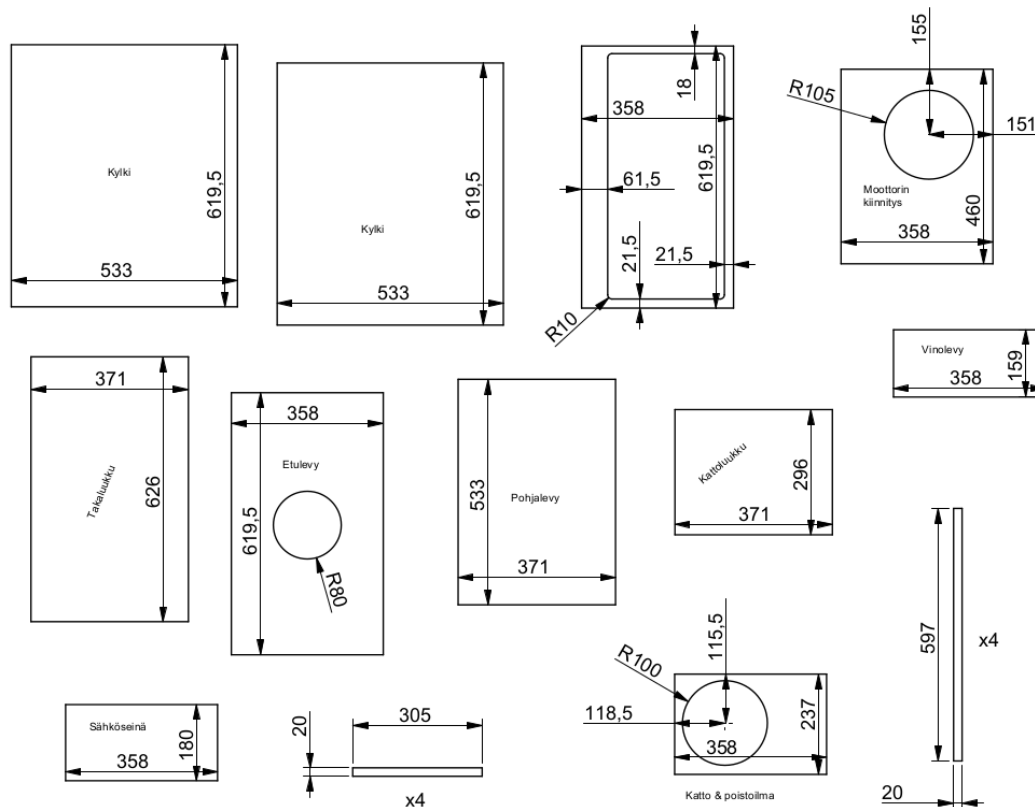
Kuva 3. Kotelo, jossa suodattimet, puhallin ja virtalähde paikallaan. Kolme seinää ja kattoluukku poistettuna.



Kuva 4. Kotelon toinen sivuseinä, etuseinä ja sisälle jäävät seinät näkyvissä. Muut osat piilotettuina.

3.2. Rakentaminen

Ajattelimme leikata kotelon seinät laserleikkurilla, mutta kuulimme, että niillä ei voi leikata kovin paksua materiaalia, joten harkitsimme jälleen 4 mm paksuisen vanerin käyttämistä. Kokeilimme 4 mm vanerin leikkaamista laserilla, mutta jälki oli palaneen näköinen ja epätasainen. Seinämme eivät myöskään olisi mahtuneet kokonaisina sähköpajan laserleikkuriin. Yliopiston Fablabilla olisi ollut suurempi laserleikkuri, mutta Fablab meni kiinni kesken kesäkurssin, joten päätimme leikata seinät sirkkelillä 3D-mallista saatujen mittojen mukaan (kuva 5).



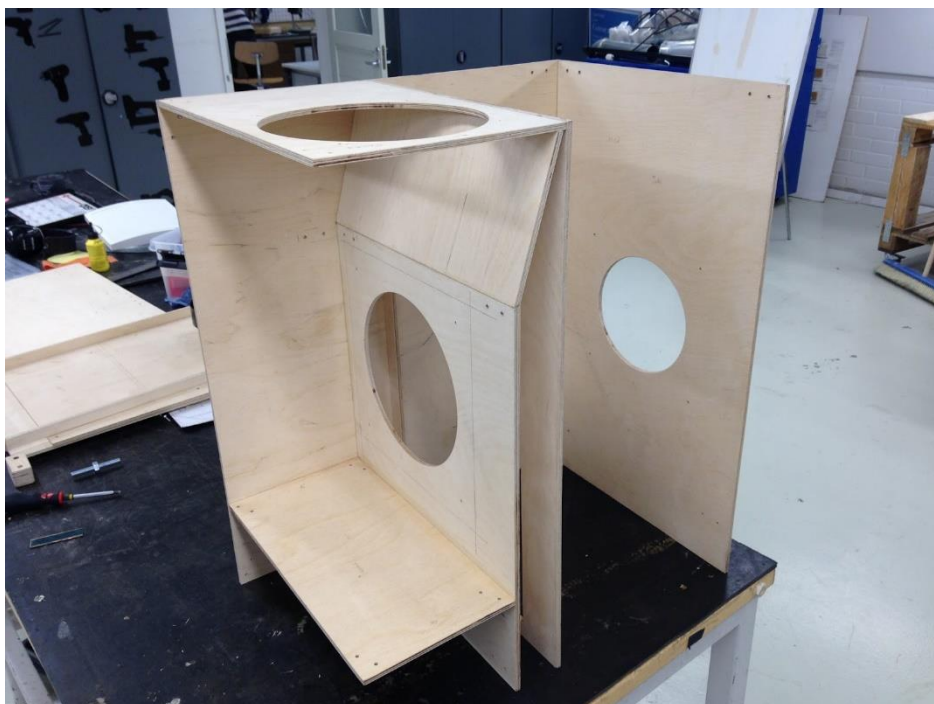
Kuva 5. Kotelon osien mitat.

Tämän jälkeen kotelon osiin tehtiin yläjyrsimellä pyöreät reiät, joiden kautta ilma pääsee kulkemaan. Tätä varten Design Factoryllä ei ollut sopivaa työkalua, joten päädyimme valmistamaan oman työkalun, jolla saa pyörätettyä yläjyrsintä tasaisessa ympyrässä (kuva 6).



Kuva 6. Yläjyrsin

Kun osat olivat valmiit, ryhdyimme poraamaan vaneriin reikiä kulmarautoja varten. Liimasimme vaneriosat toisiinsa kiinni Eri Keeperillä, jolloin levyt jäykistyivät toisiaan paremmin eri suunnissa. Lisäksi liima tekee liitoksista valmiiksi ilmatiiviimmät.



Kuva 7. Kotelon sisäseinät liimattu. Vertaa kuvaan 4.

Liimauksen jälkeen asensimme kulmaraudat ja kiinnitimme koteloon kaksi luukkua saranoilla. Suodattimien ja luukkujen reunoille liimasimme ikkunatiivistettä. Suodattimet voidaan vaihtaa kattoluukun kautta. Elektroniikkaan ja puhaltimeen pääsee käsiksi takaluukusta.



Kuva 8. Kotelo, jossa suodattimet asetettuna paikoilleen ja luukut avattuina.

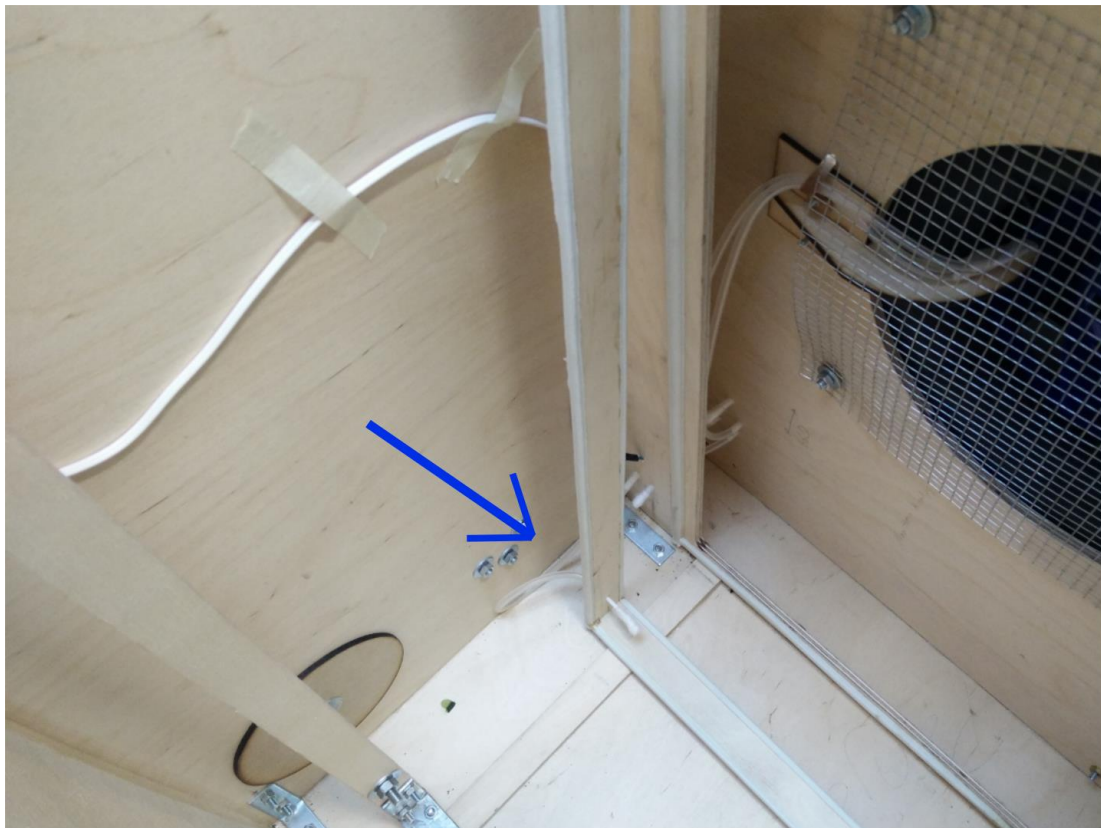
Työvaloja varten tehtiin laserleikkurilla vanerista erillinen kotelo. Työvalojen kotelon etummainen seinä on irrotettava, ja se kiinnittyy ruuveilla. Muut osat on liimattu toisiinsa, ja itse työvalokotelo kiinnittyy laitteeseen ruuveilla ja muttereilla. Työvalokotelon piirustusten tekemiseen käytettiin valmista ohjelmaa [1], joka luo valmiit sormiliitokset halutun kokoisen laatikon liitoskohtiin. Työvalokotelon piirustukseen tehtiin manuaalisesti reiät kiinnitysruuveille, sähköjohdolle ja työvaloille.

3.3. Letkut ja johdot

Paineen mittausta varten antureita ei ole sijoitettu suodattimien ympärille, vaan suoraan piirilevyyn. Paine saadaan antureihin ohuita, halkaisijaltaan noin 3 mm silikoniletkuja pitkin, joiden toinen pää on kiinni pisteessä, josta halutaan mitata painetta. Virtapiiri ja virtalähde on sijoitettu laitteen takaosaan. Painetta täytyy kuitenkin mitata suodattimien eri puolilta, ja sähköjohtoja tarvitaan ON/OFF-nappulaan, joka sijaitsee moottorin yläpuolella, sekä työvaloihin, jotka ovat kiinni sille tehdyssä työvalokotelossa laitteen etupuolella. Tätä varten edestä katsottuna suodattimien vasemmalla puolella on 4 cm leveä tila, jossa johdot kulkevat, sekä laitteen väliseiniin on tehty reikiä, joiden läpi johdot kulkevat. Paineenmittausta tasaamassa letkujen päässä on vanua. Johtojen kulku kotelossa näkyy kuvissa 9 ja 10.



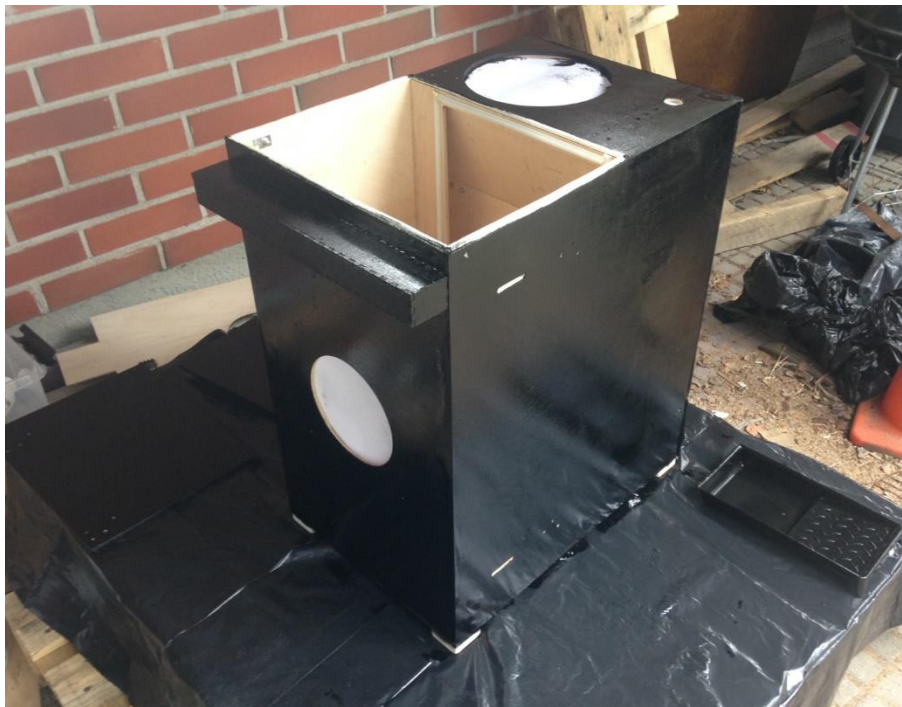
Kuva 9. Kotelo ylhäältä päin. Johdoille osoitettu tila on kuvassa vasemmalla.



Kuva 10. Letkuille varattu tila.

3.4. Maalaaminen

Laite päätettiin maalata mustaksi, sillä maaleja tilattaessa ei tiedetty, miltä eri värit tulisivat näyttämään laitteen päällä. Maaliksi valittiin iskunkestävä Miranol alkydimaali, joka ei lähde helposti kulumaan käytössä. Haittapuolena oli kuitenkin kuivumiseen vaadittu pitkä aika. Täydellinen kuivuminen kestää jopa kuukauden maalin hitaan hapettumisen vuoksi. Maalia ei ohennettu, sillä maali todettiin hyväksi sellaisenaan. Maalaamista ennen laitteen ruuvit peitettiin ja isoimmat kolhut paikattiin kitillä. Laite hiottiin rajallisen ajan vuoksi vain kahdella eri karkeusasteen hiomapaperilla, jolloin vanerin pintaan jäi epätasaisuuksia, jotka näkyvät maalipinnassa. Käytössä ei ollut maaliruiskua, vaan pensseleitä ja tela, mikä vaikutti maalipinnan tasaisuuteen haitallisesti. Laitteeseen maalattiin kaksi kerrosta maalia ja maalauskerrosten välissä oli yksi päivä. Vaikka laite oli kosketuskuiva jo päivän kuluttua, päätettiin odottaa 4 päivää, ettei maaliin tulisi kolhuja. Tämän neljän päivän aikana laitteen runkoa ei pystytty käyttämään. Virtapiiri, moottori ja kaikki ruuvit ja mutterit, jotka on tarkoitettu irrotettaviksi, poistettiin maalauksen ajaksi.



Kuva 11. Vastamaalattu kotelo.

3.5. Ilmavirrat

Kotelossa olevien ilman imu- ja poistoaukkojen halkaisija oli määritelty yrityksen toiveesta alan standardiksi 160 mm. Poistoaukkoa varten tilasimme metalliputken, joka pyöristää poistoaukon reunoja, vähentäen hukkatehoa. Putki toimii myös liittimenä, jos poistoaukkoa halutaan muokata mm. laittamalla ilmavirran nopeutta mittaava säätöpelti aukon päähän. Tilatun putken toinen pää oli leveämpi kuin tilatessa luulimme, joten jouduimme hieman leikkaamaan sitä. Radiaalipuhallin tarvitsee ympärilleen riittävästi tilaa ja jouduimme suunnittelemaan kotelon leveämmäksi kuin suodattimet, jotta puhaltimen ympärillä tapahtuisi vähemmän ilmavirran kuristusta. Imuaukon yläpuolella oleva työvalokotelo luo pyörteen imuaukolle, joka estää pölyä karkaamasta.

4. Laitteisto

4.1. Puhallin

Yritys oli jo perehtynyt erilaisiin puhallinratkaisuihin ja saimme heiltä käyttöömmme verkkovirralla toimivan radiaalisen puhaltimen. Puhaltimen pyörimisnopeutta säädetään PWM:llä. Pyörimisnopeus voidaan mitata puhaltimen moottorin lähettämistä sähköpulsseista. Puhallin kykenee saamaan aikaan 1500 m³/h tilavuusvirran tai 850 Pa staattisen paine-eron.

4.2. Virtalähde

Valitsimme virtalähteeksi TracoPowerin TXL-sarjan mallin, josta saamme suoraan tarvitsemamme 5 V ja 12 V jännitteet. 5 V jännite on virtalähteessä säädettävä ja säädimme sen 4,5 volttiin GSM-moduulin säästämiseksi.

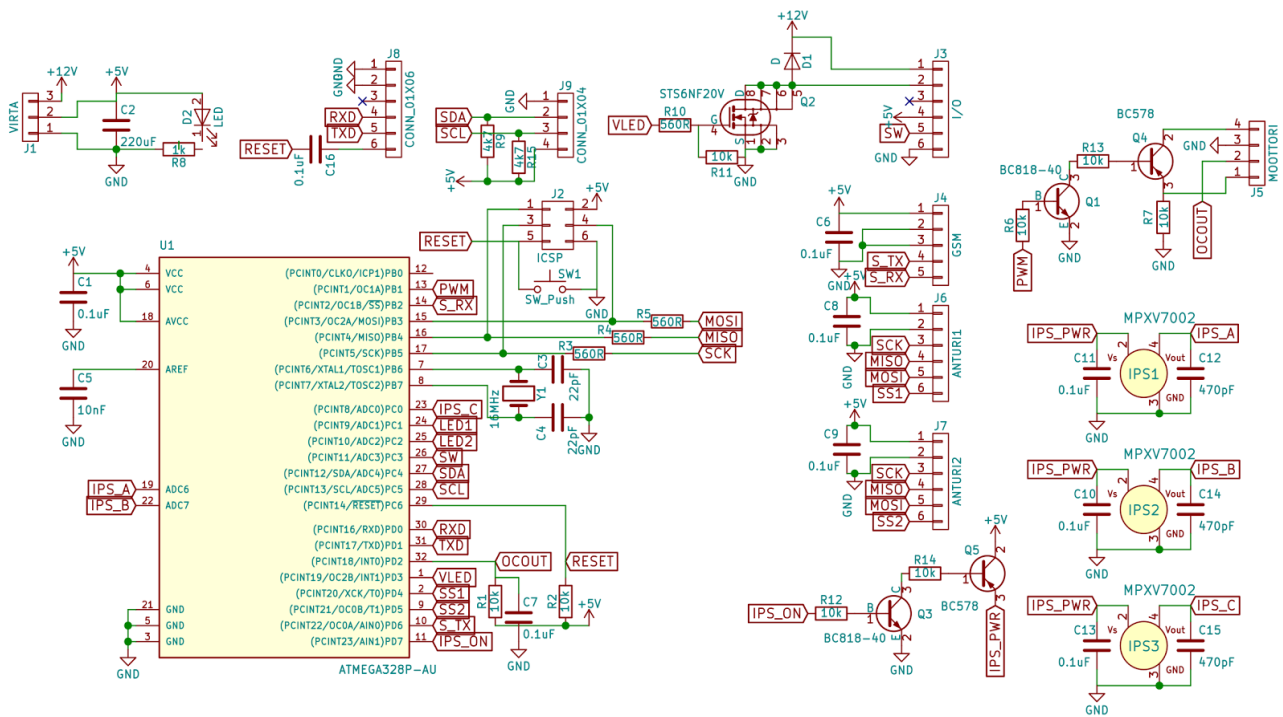


Kuva 12. Virtalähde

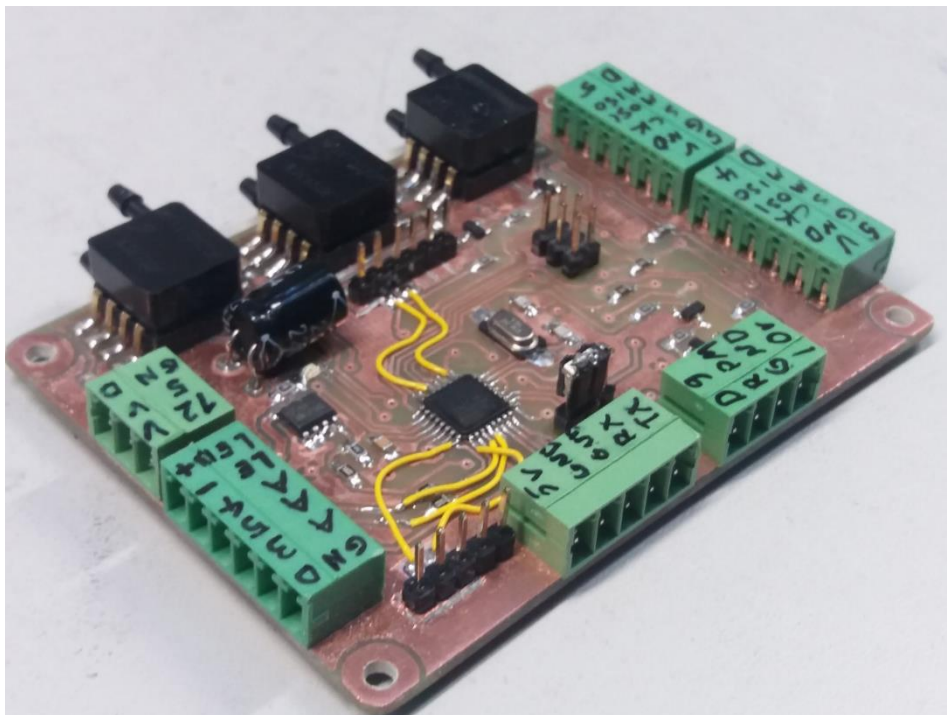
4.3. Piirilevyt

Laitteistoa ohjaa Atmelin 8-bittinen ATmega328P. Valitsimme mikrokontrollerin siitä löytyvien ADC ja PWM pinnien takia, joita tarvittiin analogisten paineantureiden lukemiseen ja moottorin hallintaan. Suosittu mikrokontrolleri löytyy mm. Arduino Unosta, mikä helpotti laitteiston testausta ennen varsinaisen piirin osien saapumista. Piirilevyille juotettiin ISP (In-System Programming) - ja sarjaportin kautta toimiva ohjelmointiliitin. Kontrollerin piirikytkennät näkyvät kuvassa 13 ja valmis piirilevy kuvassa 14.

Moottorin kierroslukua ohjataan syöttämällä sille jännitettä välillä 0 - 10 V. Mikrokontrollerista tulee ulos vain 5 V jännitettä ja jotta moottoria voitaisiin pyörittää täysteholla, jouduimme vahvistamaan 5 V jännitesignaalin 10 volttiin. Signaalin jännite nostetaan 10 volttiin käyttämällä kahta transistoria Sziklai-parina [2].

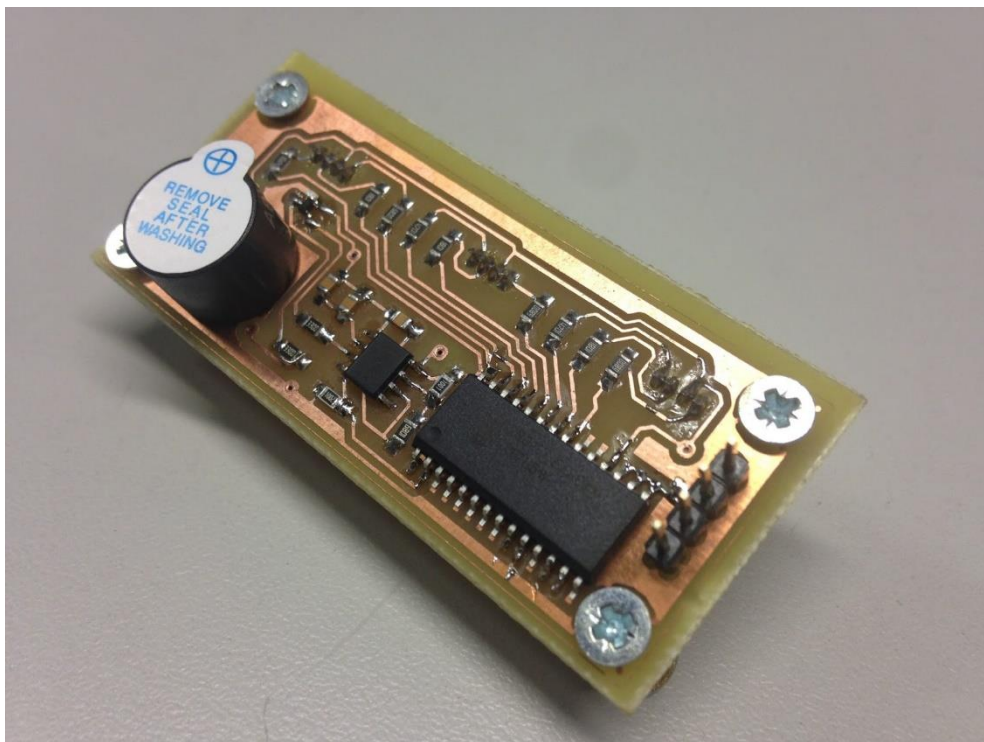


Kuva 13. Piirilevyn kytkentäkaavio

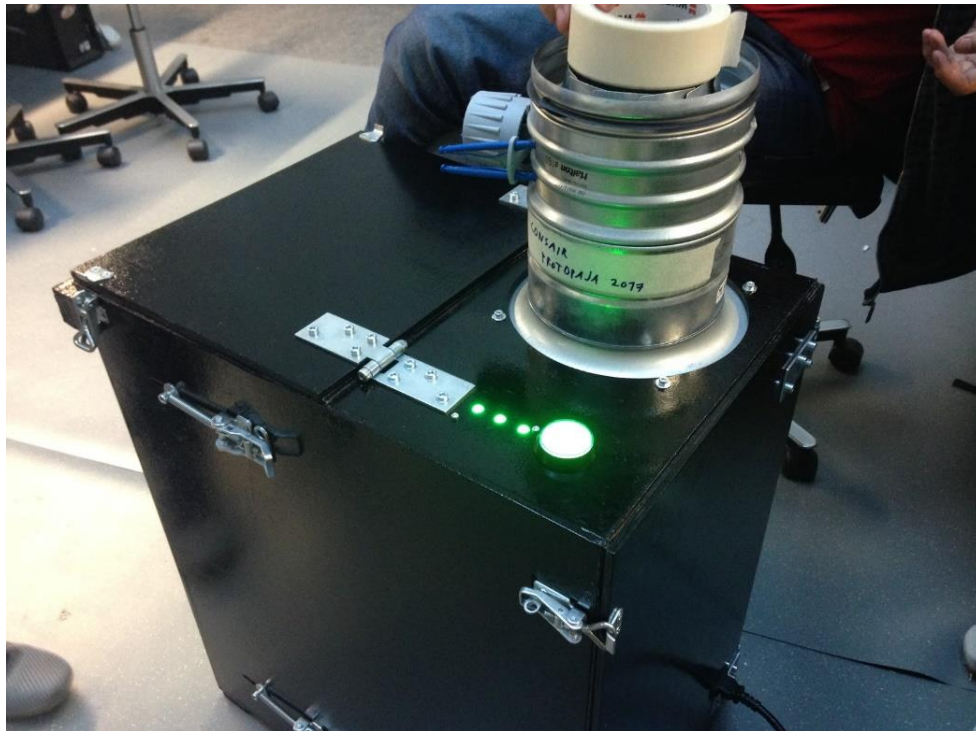


Kuva 14. Mikrokontrollerin piirilevy.

Prototyypiin lisättiin I/O-expander, koska päätimme käyttää RGB-LED:ejä yksiväristen LED:ien sijaan ja mikrokontrollerista loppuivat vapaat pinnit kesken. I/O-expander juotettiin RGB-LED:ien kanssa samalle piirilevylle. Samalle levylle juotettiin myös piezokaiutin ja 555-ajastinpiirillä toteutettu värähtelypiiri kaiuttimen värisyttämiseen noin 2,4 kHz taajuudella. Värähtelypiiri kytketään päälle yhdellä I/O-expanderin pinneistä, samalla tavalla kuin RGB-LED:itkin. LED:ien piirilevy on esitetty kuvassa 15.



Kuva 15. LED:ien piirilevyn pohjapuoli. LED:it ovat toisella puolella levyä.



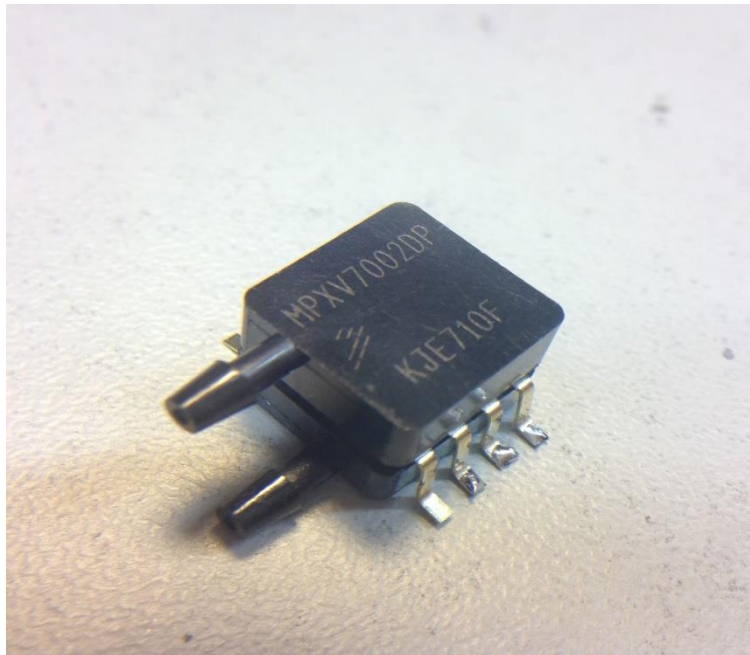
Kuva 16. Ledit päällä.



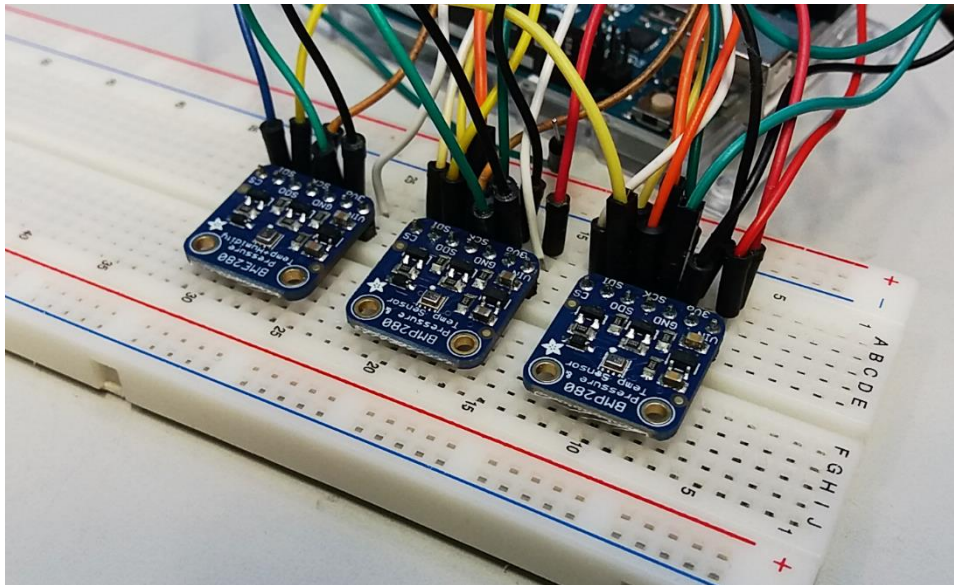
Kuva 17. Ledit ja opasteet.

4.4. Anturit

Käytämme paine-erojen mittaamiseen analogisia differentiaalisia MPXV7002DP paineantureita (kuva 18), koska ne on helppo juottaa samalle piirilevylle mikrokontrollerin kanssa ja paineletkut voidaan suoraan työntää kiinni anturien liittimiin. Käyttämiemme differentiaalisten antureiden resoluutio on parempi kuin esim. digitaalisten absoluuttista painetta mittaavien BMP280 antureiden resoluutio (kuva 19), muun muassa siksi, että differentiaalisten antureiden paineenmittausalue on kapeampi (± 2000 Pa) verrattuna absoluuttisten antureiden alueeseen (30 000 - 110 000 Pa). Mittaamamme paine-erot ovat suuruusluokkaa 100 Pa. Digitaalisissa antureissa olisi ollut sisäänrakennettu lämpötilakompensointi, mutta analogisissa antureissa ei ole sellaista, joten joudumme nollaamaan analogiset anturit muutaman minuutin välein.



Kuva 18. Käyttämämme analoginen differentiaalinen paineanturi.



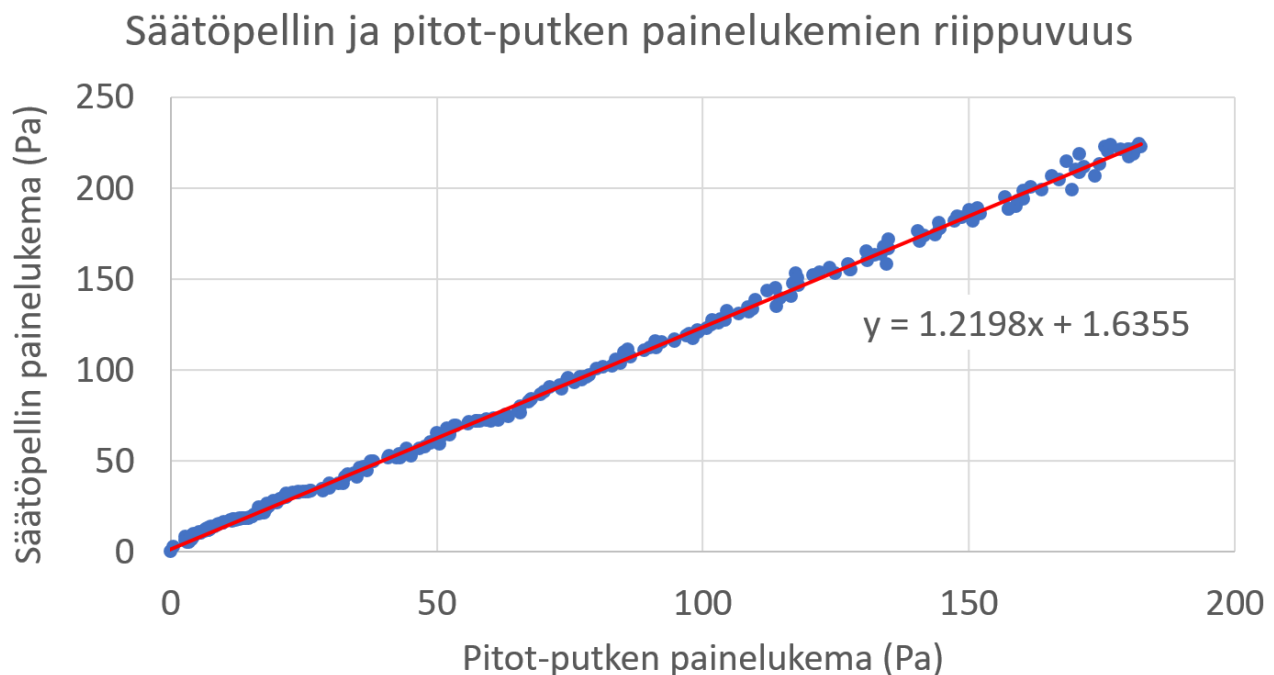
Kuva 19. Digitaaliset absoluuttiset BMP280- ja BME280-anturit.

Yksi antureista liitettiin letkuilla itse tehtyyn pitot-putkeen (kuva 20), jolla pystytään mittaamaan ilmavirran nopeutta laitteen sisällä. Kyseisen anturin painelukema annetaan suoraan PID-säätimelle, joka yrittää pitää kyseisen painelukeman, ja siten myös ilmavirran nopeuden, vakiona. Pitot putki sijoitettiin puhaltimen imuaukole, koska se on kapein kohta laitteen sisällä ja siten ilmavirran nopeus on suurimmillaan kyseisessä kohdassa. Puhaltimen imuaukko on molempien suodattimien takana, jolloin pitot putkelle virtaa puhdistettua ilmaa ja se on paremmin pölyltä suojassa.



Kuva 20. Pitot-putki puisen varren päässä puhaltimen imuaukolla.

Vertasimme pitot-putkesta mitattuja painelukemia kaupalliseen virtausnopeutta mittaavaan säätöpeltiin ja piirsimme molemmat painelukemat samaan kuvaajaan puhaltimen eri pyörimisnopeuksilla (kuva 21). Molempien mittaustapojen painelukemat asettuvat yhdelle suoralle puhaltimen pyörimisnopeudesta riippumatta, mikä osoittaa, että pitot-putkella saadaan mitattua järkeviä nopeuslukemia ilmapirrille laitteen sisällä. Kuvaajan oikeassa yläalaidassa, kun puhaltimen pyörimisnopeus on suuri, mittauspisteet alkavat hajota sovitetun suoran ympärille. Tämä hajonta johtuu luultavasti ilman pyörteilyn kasvusta laitteen sisällä suurilla virtausnopeuksilla.



Kuva 21. Itse tekemämme pitot-putken ja kaupallisen virtausta mittaavan säätöpellin vertailua puhaltimen eri pyörimisnopeuksilla.

4.5. GSM-moduuli

GSM-moduuliksi valikoitui Sain Smartin nelitaajuus SIM900-moduuli (kuva 22). Siinä on tarpeisiimme nähden riittävät ominaisuudet: GPRS ja tekstiviestin lähetyksen mahdollisuus. Emme tarvitse 3G-moduulia, koska siirrettävän tiedon määrä on projektissamme pieni eikä tiedonsiirron nopeus ole ratkaisevaa.



Kuva 22. Sainsmartin SIM900-moduuli

Mikrokontrolleri keskustele GSM-moduulin kanssa sarjayhteydellä. GSM-moduuli toimii modeemien ohjaukseen tarkoitetuilla AT-käskyillä, joihin moduuli vastaa ennalta määrätysti. Vastauksista voidaan päätellä, onnistuiko käskyn anto ja voidaanko seuraavaa käskyä suorittaa. Koska vastaukset tulevat viiveellä, voidaan ne lukea turvallisesti vasta jonkin ajan kuluttua käskyn annosta. Tämä hidastaa käskyjen antoa huomattavasti.

Tarkoituksena oli lähettää myös tieto laitteen GPS-koordinaateista, mihin olisimme tarvinneet GPS-moduulin. Tätä varten tilasimme Adafruitin FONA 808 piirilevyn SIM808-moduulilla, missä on sekä GSM- että GPS-ominaisuudet. Emme kuitenkaan saaneet moduulia toimimaan; moduuli ei hyväksynyt SIM:n pincodea, mitä ilman moduuli ei ottanut mitään komentoja vastaan. Rajoitetun ajan vuoksi emme käyttäneet resursseja SIM808-moduulin debuggaamiseen.

5. Ohjelmistot

5.1. Mikrokontrollerin softa

Ohjelmointikielinä käytämme C:tä ja C++ Arduinon kirjastoilla. Asensimme mikrokontrolleriin ISP-liittimen kautta Arduino Unostakin tutun *Optiboot*-bootloaderin koodin päivittämisen helpottamiseksi. Näin voimme sekä lukea debug-viestejä, että ohjelmoida mikrokontrolleria suoraan Arduino IDE:stä yhtä USB-serial -adapteria käyttäen. Käytimme bootloaderin asennuksessa ISP-ohjelmointilaitteena Arduino Unoa, koska meillä sattui olemaan sellainen ja se toimi Arduino IDE:n kanssa ilman laiteajureiden asentamista [3]. ISP-liittintä käyttäen koodi voidaan tulevaisuudessa asentaa mikrokontrollerille myös suoraan ilman bootloaderia.

5.1.1. Paineenmittaaminen

Laitteen käynnistuksen jälkeen paineantureilla mitataan useita arvoja ja näistä muodostetaan keskiarvo. Vähentämällä saatu keskiarvo antureiden lukemasta saadaan painelukemat nollattua samalle tasolle.

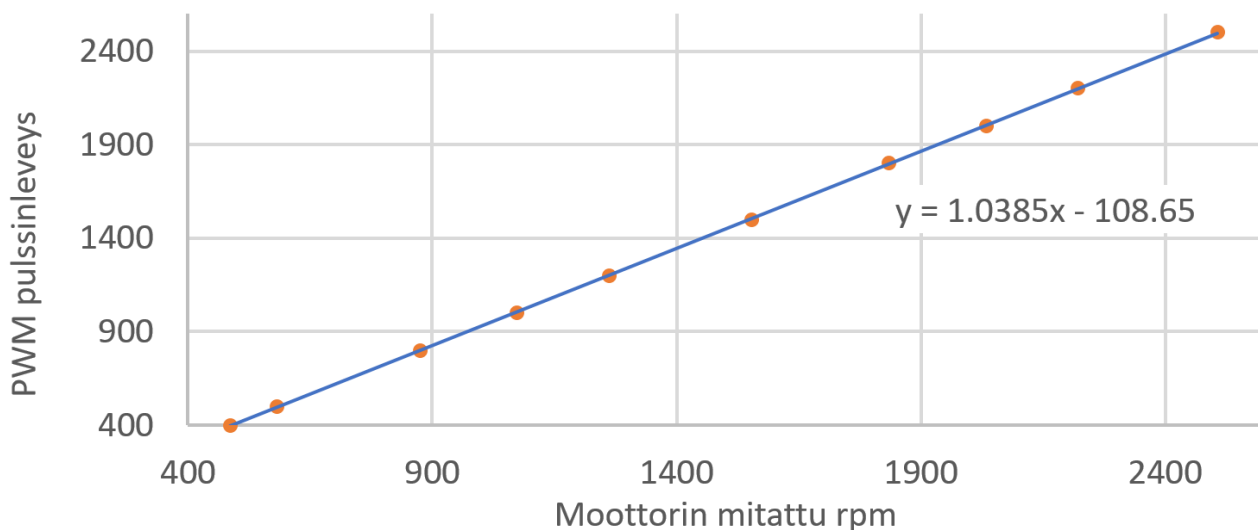
Anturit kannattaa nollata vain silloin, kun paine laitteen sisällä on kaikkialla sama, eikä ilma liiku. Jotta anturit voitaisiin nollata, täytyy moottori pysäyttää. Moottorin uudelleen käynnistäminen kuitenkin kestää noin 10 sekuntia, mikä viivästyttää laitteen käyttämistä, jos käyttäjä sattuu tarvitsemaan laitetta juuri, kun nollaus on meneillään. Laitteen käyttö viivästyisi vähemmän, jos puhallin ei pysähtyisi kokonaan nollauksen ajaksi, vaan pyörisi mahdollisimman hitaasti. Tämä kuitenkin aiheuttaisi nollaukseen muutaman pascalin virheen, joka ei välttämättä häittäisi liikaa ollessaan niin pieni.

Nollaus suoritetaan viiden minuutin välein, koska lämpötilan muutokset vaikuttavat anturien lukemiin. Tästä syystä anturien lukemat alkavat ajautua erilleen virtojen kytkemisen jälkeen kun elektroniikka lämpenee. Kun elektroniikka on lämmennyt, anturien lukemat pysyvät vakaampina. Jotta automaattinen nollaus ei häiritsisi laitteen käyttöä, nollataan anturit vasta 5 minuutin kuluttua laitteen viimeisestä käyttökerrasta. Tämä johtaa siihen, että jos laitetta ei käytetä lainkaan, anturit nollataan toistuvasti 5 minuutin välein. Toisaalta jos laitetta käytetään tauotta, antureita ei nollata kuin kerran virtajohtoa kytkettäessä.

5.1.2. Moottorinohjaaminen

Puhaltimen pyörimisnopeutta ohjataan pulssinleveysmodulaatiolla (PWM) jännitevälillä 0 - 10 V. Pölynhallintalaitteen ollessa päällä on tarkoituksena pitää ilmavirta laitteen läpi ennalta määritettynä vakioarvona. Kun suodattimet ovat puhtaita, riittää tähän pieni tuulettimen pyörimisnopeus. Laastipölyn kasaututtua suodattimiin, täytyy imutehoa kasvattaa, jotta vakioilmavirta säilyy. Suodattimien tukkeutumisaste muuttuu käytön aikana, kun suodattimen pintaan kerääntynyttä pölyä putoaa alas. Tällöin virtaus laitteen läpi kasvaa ja moottorin imutehoa tulee pienentää.

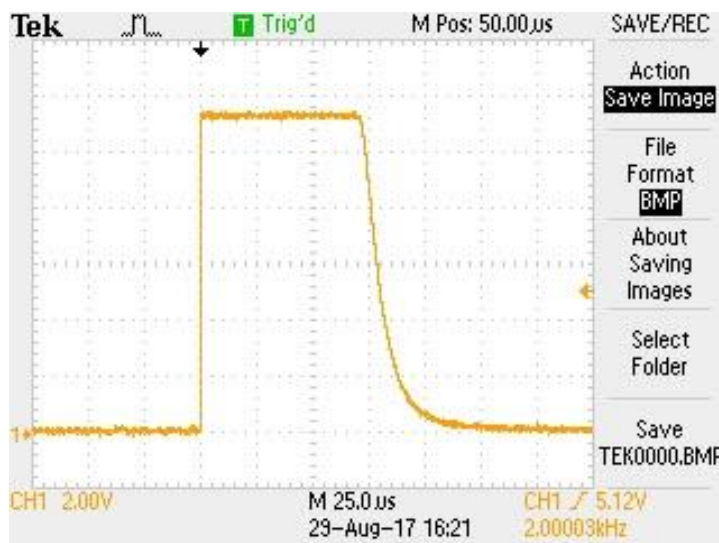
Pulssinleveysmodulaatio asetettiin mikrokontrollerissa siten, että pulssin leveydeksi voidaan antaa arvoja välillä 0 - 4000, mikä vastaa pulssisuhdetta 0 - 100 %. Moottori ohjelmoitiin valmistajan omalla ohjelmalla siten, että myös sen kierroslukurajat ovat välillä 0 - 4000 rpm. Näin moottorin pitäisi teoriassa pyöriä samaa kierroslukua kuin mitä sille syötetään pulssin leveytenä. Näin ei kuitenkaan käytännössä tapahtunut, sillä moottorilla on minimipyörimisnopeus 315 rpm. Kokeilimme ohjata moottoria eri pulssinleveyksillä ja mittasimme vastaavat moottorin kierrosluvut moottorin sisäiseltä takometrilta saatavia pulsseja mikrokontrollerilla lukien. Piirsimme näin saamamme mittaustulokset exceliin (kuva 23) ja sovitimme mittauspisteisiin suoran. Moottorille syötettävä PWM-signaali korjataan suoran yhtälöä käyttäen ja näin moottori saadaan pyörimään lähes samaa nopeutta kuin mitä sille käsketään. Kuvasta 23 nähdään, että suoran kulmakerroin ei ole 1,00 kuten voisi ajatella. Yksi mahdollinen selitys voi olla mikrokontrollerin ja moottorin välisen Sziklai-parin transistorien hidas sammuminen.



Kuva 23. PWM-signaalin pulssinleveyden ja moottorin kierrosluvun välinen riippuvuus.

Kun tarkastelimme moottorille syötettävää PWM-signaalia oskilloskoopilla, havaitsimme että PWM-signaalin nouseva reuna on jyrkkä, niin kuin pitääkin, mutta laskeva reuna on loiva (kuva 24). Laskevan reunan loivuus aiheuttaa sen, että moottorille syötettävän signaalin pulssisuhde on suurempi kuin on tarkoituksena. Suurilla kierrosluvuilla laskevan reunan loivuuden vaikutus on häviävän pieni, mutta pienillä kierrosluvuilla sen merkitys kasvaa. Tämä selittää kuvassa 23 sen, että 2400 kohdalla pulssinleveys ja moottorin mitattu rpm kohtaavat samalla suoralla, mutta 900 kohdalla moottori pyöriikin 900 rpm jo pienemmällä pulssinleveydellä kuin 900.

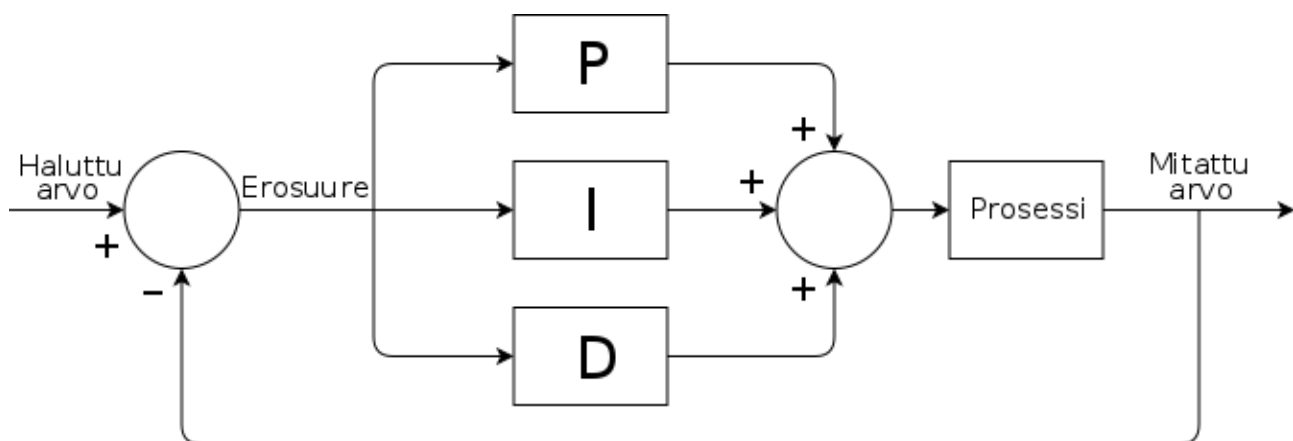
Ongelmaa voi yrittää korjata lisäämällä transistoreihin alavetovastukset, vaihtamalla transistorit nopeampiin MOSFET:hin tai käyttämällä transistorien sijaan level shifter -piiriä. Tällöin kuvan 23 yhtälöä ei ehkä tarvitsisi käyttää lainkaan PWM-signaalin korjaamiseen. Korjaamme signaalia yhtälöllä vain, kun haluamme asettaa puhaltimen vakiopyörimisnopeuteen. Signaalia ei tarvitse korjata silloin, kun ilmapirran nopeus halutaan pitää vakiona, sillä PID-säädin tekee korjauksen automaattisesti puolestamme.



Kuva 24. Moottorille syötettävää PWM signaalia oskilloskoopin näytöllä.

Moottorin kierroslukua säädetään PID-säätimellä (Proportional Integral Derivative), jonka toteutimme käyttämällä valmista Arduinolle tarkoitettua PID-kirjastoa [4]. Säädin toimii ohjaamalla moottorin kierroslukua kohti haluttua arvoa käyttämällä hyväksi näiden välistä erotusta ja säätimen PID-kertoimia. Aluksi säätimen kertoimet pyrittiin määrittämään Ziegler–Nichols-menetelmällä [5]. Näillä asetuksilla moottorin kierrosluku kuitenkin reagoi liian hitaasti optimaaliseen toimintaan verrattuna, joten lopulliset kertoimet määritettiin kokeilemalla erilaisia arvoja.

PID-säätimen toimintaperiaate on esitetty kuvassa 25. Kuvassa prosessi on puhallin, jolle syötetään PWM-signaalia vasemmalta. Mitattu arvo on pitot-putkesta mitattu dynaaminen paine, joka kuvastaa ilmavirran nopeutta. PID-säädin vertaa mitattua ilmavirran nopeutta haluttuun nopeuteen ja laskee P, I ja D -kertoimien avulla uuden arvon moottorille syötettävälle PWM:lle.



Kuva 25. PID-säätimen toimintaperiaate

5.1.3. Datan lähetys

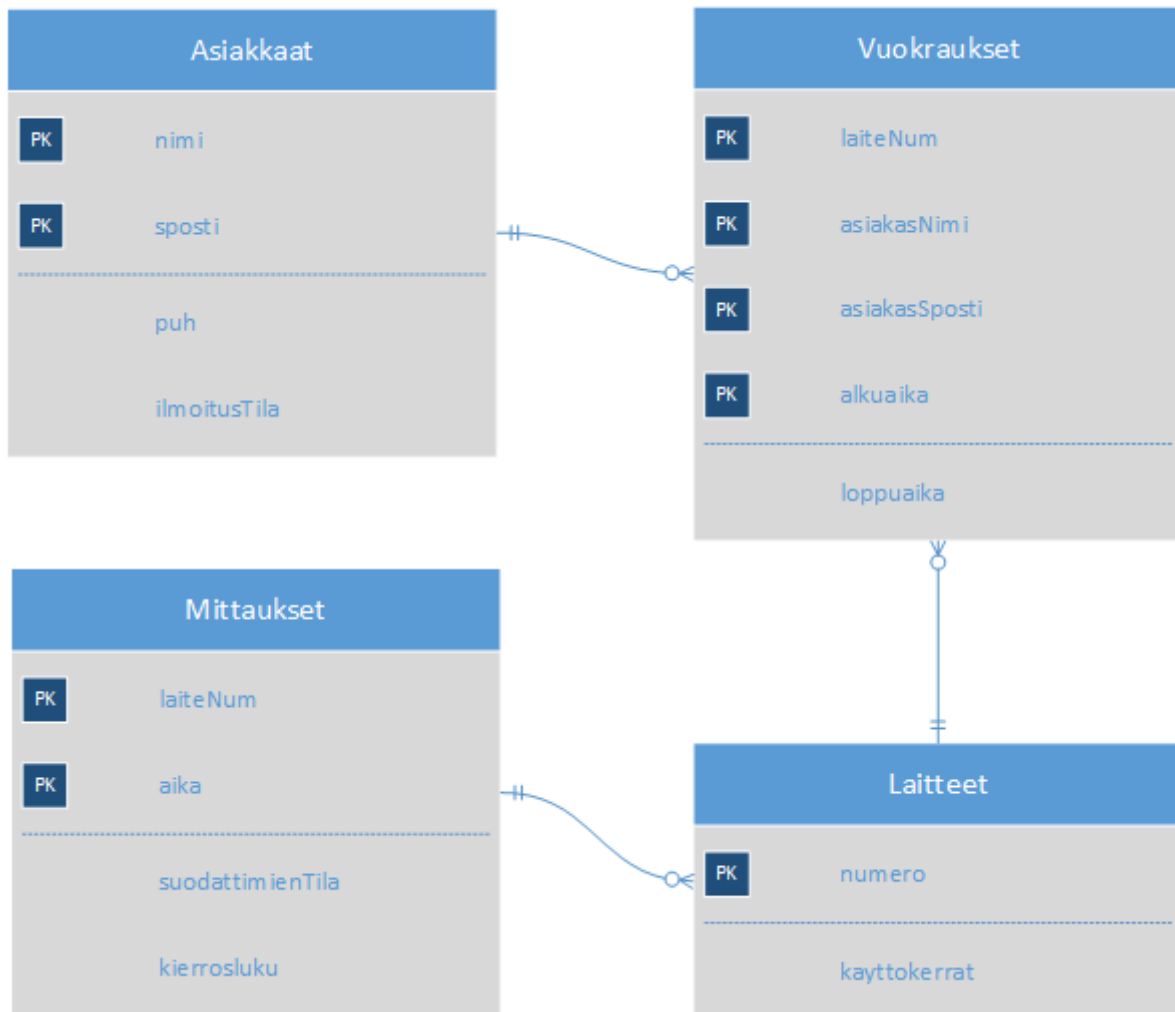
GSM-moduuli muodostaa TCP-yhteyden palvelimeen laitteen käynnistyksen jälkeen ja lähettää laitteen identifiointiin tarkoitetun numeron, tiedon suodattimen sen hetkisestä tukkoisuusasteesta ja moottorin siihen mennessä pyörähtämien kierrosten kokonaislukumäärän.

Palvelin vastaa GSM-moduulille laitteen vastuuhenkilön asetuksista riippuen lähettämällä numeron, johon lähetetään tekstiviesti. Tässä ilmeni ajoittain ongelmia. Välillä sarjayhteys ei toiminut eikä numeroa voinut tunnistaa, koska siirretty data oli lukukelvotonta. Välillä taas numero siirtyi onnistuneesti, mutta silti tekstiviestin lähetys epäonnistui tuntemattomasta syystä.

5.2. Virtuaalipalvelin

Saimme käyttöömmme virtuaalipalvelimen koululta. Virtuaalipalvelin pyörittää kirjoittamaamme python-skriptiä, joka vastaanottaa GSM-moduulin lähettämää dataa.

Python valittiin ohjelmointikieleksi, koska sen käytöstä oli jo kokemusta ja myös siksi, että virheiden käsittely on sillä yksinkertaisempaa kuin C:llä. Myös datan jatkokäsittely ja visualisointi onnistuu tarvittaessa kätevästi Pythonin lukuisilla, hyvin tuetuilla kirjastoilla (esim. numpy ja matplotlib). Python myös tukee SQLite-tietokantaa, mikä on yksinkertainen tapa toteuttaa pieni tietokanta, koska se ei vaadi erillistä tietokantapalvelinta. Tiedon lisääminen ja hakeminen tapahtuu SQLite-tietokannasta nopeasti ja turvallisesti toisin kuin tekstitiedostosta. Tietokanta koostuu neljästä taulukosta (kuva 26).



Kuva 26. Tietokannan ER-kaavio (rakennediagrammi)

5.2.1. Datavastaanotto ja tallennus Python-skriptillä

Skripti luo socketin ja alkaa kuuntelemaan sille annettua TCP-porttia. Mikäli vastaanotettu data on oikean muotoinen merkkijono, se pilkotaan listaksi. Lista sisältävät laitteen identifiointiin tarkoitettu laite numero, tieto suodattimien sen hetkisestä tukkoisuusasteesta ja moottorin siihen mennessä pyörähtämien kierrosten kokonaislukumäärä. Listan alkiot tallennetaan yhdessä lähetyksen ajankohdan kanssa serverillä SQLite-tietokantaan, Mittaukset-taulukkoon.

Mikäli vastaanotetusta merkkijonosta ilmenee se, että laitteen suodattimet pitäisi vaihtaa, niin skripti hakee tietokannan muista taulukoista tiedot laitteen sen hetkisestä vastuuhenkilöstä. Tämän jälkeen vastuuhenkilölle lähetetään sähköposti ja/tai tekstiviesti vastuuhenkilön toiveiden mukaan. Viestistä ilmenee mitkä suodattimet ovat tukossa. Lopuksi skripti sulkee yhteyden clienttiin ja jää odottamaan uutta yhteydenottoa.

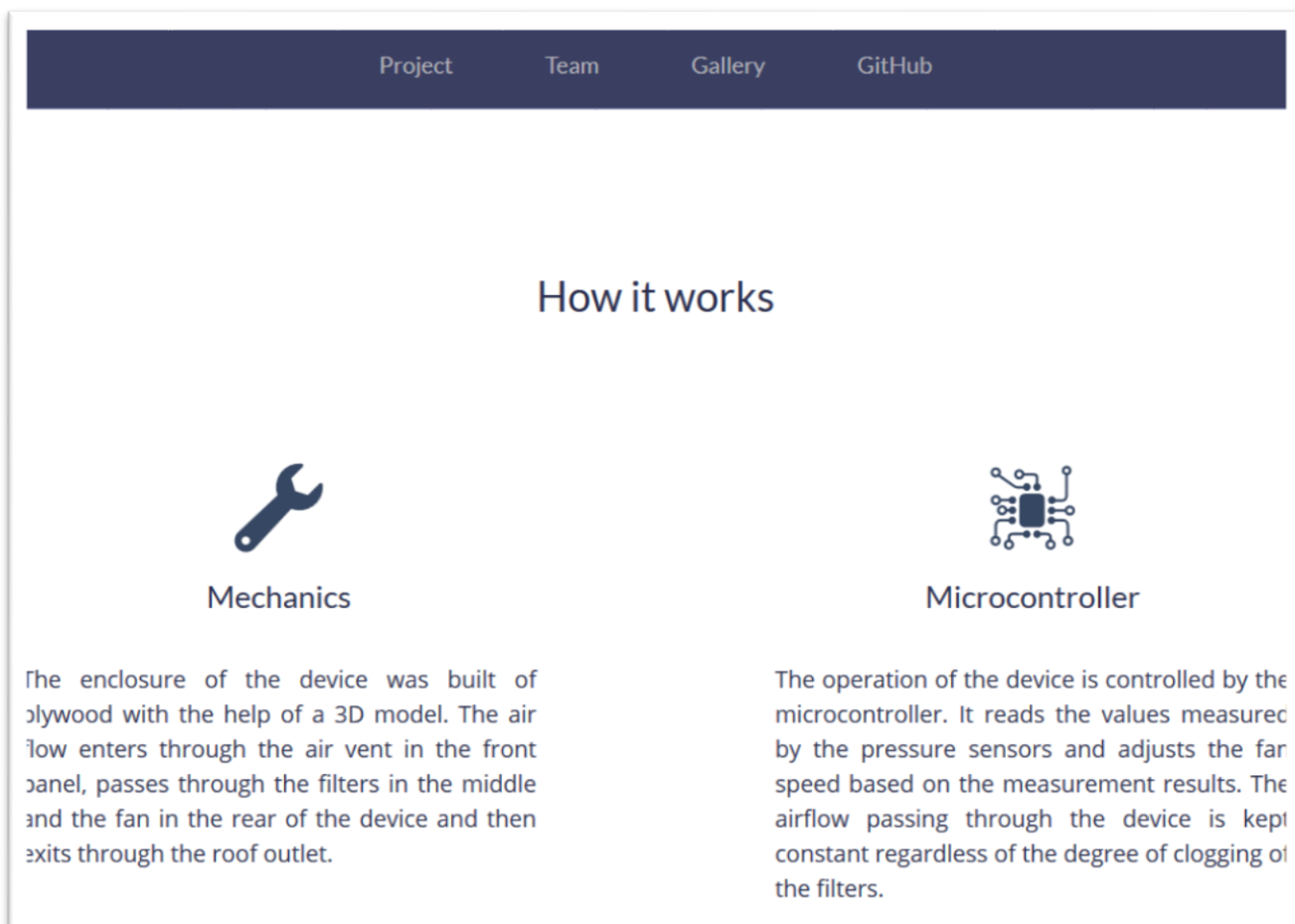
Skripti pitää myös kirjaa ilmenneistä virheistä kirjoittamalla ne lokitiedostoon myöhempää tarkastelua varten.

6. Nettisivut

Projektin kotisivut on jaettu kolmelle sivulle. Etusivu koostuu toisistaan erottuvista temaattisista osista: projektin lyhyt kuvaus videon kanssa, laitteen eri osa-alueiden esittely ja ryhmän jäsenten esittely ja päävastuualueet. Sivun ylälaidassa on horisontaalinen navigaatiopalkki, missä on linkit sivun eri osiin, GitHubiin ja Galleria-sivulle. Laitteen osa-alueiden tarkemmat kuvaukset löytyvät omalta sivultaan, minne on lisätty kuvia selitysten tueksi. Galleria-sivulla on muutama kuva kotelon ja piirilevyjen valmistuksesta. Sivut on toteutettu englanniksi.

Sivut mukautuvat käyttäjän ruudun kokoon eli soveltuvat sekä PC- että mobiilialustoille. Kun ruudun leveys alittaa 600 pikseliä, katoaa navigaatiopalkki. Sen tilalle ilmestyy vasempaan yläkulmaan valikkopainike, jota painamalla pystysuora navigaatiovalikko avautuu ja sulkeutuu.

Sivujen ohjelmointi on toteutettu HTML:llä ja CSS:llä eikä suunnittelussa ole käytetty valmiita sivupohjia tai muita koodimalleja.



Kuva 27. Ruutukaappaus projektin kotisivuilta

7. Projektitoiminta

Projekti eteni tehdyn projektisuunnitelman mukaisesti. Pahvista prototyyppiä emme rakentaneet, koska 3D-malli havainnollisti laitteen rakennetta riittävän hyvin. Kaikille ryhmän jäsenille muodostui mallin perusteella selkeä näkemys kokonaisuudesta. Ryhmän jäsenet keskittyivät tekemään niitä asioita, joita kukin osasi parhaiten.

7.1. Tavoitteiden saavuttaminen

Projekti onnistui omiin taitoihimme nähden erittäin hyvin. Lähes kaikki laitteelta toivotut ominaisuudet saatiin toteutettua, vaikka esim. GSM-moduulista, palvelimen käytöstä ja nettisivujen koodauksesta ei ollut aiempaa kokemusta. AVR C -koodia emme ehtineet tehdä ajan puutteen vuoksi, mutta Arduino-koodi riittänee havainnollistamaan prototyypin toimintaa.

Consair toivoi, että laite pystyisi tunnistamaan puuttuvan tai repeytyneen suodattimen, mutta asia jäi meiltä tutkimatta rajallisen ajan ja tärkeämpien tavoitteiden priorisoimisen johdosta. Suodattimen puuttumisen tarkistaminen olisi voinut onnistua ilman lisääntuneita paineantureiden avulla, tai vaihtoehtoisesti kotelon pohjalle lisättävillä rajakytkimillä. Repeytymisen tunnistaminen olisi luultavasti vaatinut lisää antureita, tai muuta tekniikkaa.

Tarkoitus oli lisätä BME280 mittaamaan ilmankosteutta ja lämpötilaa, jotka voitaisiin sitten lähettää palvelimelle tallennettavaksi. Kurssi loppui kuitenkin kesken, emmekä ehtineet lisätä tätä ominaisuutta.

7.2. Aikataulu

Projektisuunnitelman aikataulusta oltiin hieman jäljessä, lähinnä johtuen tilattujen tavaroiden toimitusongelmista, sekä tilauksien tekemiseen vaaditusta byrokratian määrästä. Tällä aikaa oli kuitenkin mahdollista tehdä muita asioita. Aikataulun viivästymisen vuoksi laitteen perusteelliseen testaukseen ei jäänyt riittävästi aikaa. Projekti oli kuitenkin testausta vaille valmis demopäivään mennessä ja sisälsi kaikki asiakkaan asettamat vaatimukset.

7.3. Riskianalyysi

Tilattujen osien toimituksessa kesti odotettua kauemmin. Fablabin ja Design factoryn kiinniolo heinäkuussa vaikeutti kotelon rakennusta. Emme myöskään pystyneet laserleikkaamaan suurimpaa osaa kotelon paloista, sillä sähköpajan laserleikkuri ei ole tarpeeksi hyvä ja iso. Päädyimme varasuunnitelmaan, eli leikkaamaan 6,5 mm vaneria sirkkelillä. Koska sirkkelillä ei saa tehtyä sormiliitoksia, kotelon kasaamisessa kului enemmän aikaa.

8. Yhteenveto ja johtopäätökset

Projektin aikana pääsimme soveltamaan opittua teoriaa käytännössä. Virtausmekaniikan, säätötekniikan ja elektroniikan ymmärtämisestä oli apua laitetta suunniteltaessa. Osalla ryhmän jäsenistä oli aiempaa kokemusta Eaglestä ja Creosta, mistä oli hyötyä, kun suunnittelimme piirilevyä ja kotelon 3D-mallia.

Kaikki projektiryhmän jäsenet oppivat uusia asioita. Opimme valmistamaan toimivan pölynhallintalaitteen IoT-ominaisuuksilla. Tämä edellytti erilaisten laitteistojen ja ohjelmistojen suunnittelua ja yhdistämistä sulavasti toisiinsa. Projektityöskentely opetti ryhmätyötaitoja, töiden roolitusta, projektisuunnitelman laatimista ja sen tärkeyttä. Näin suuressa projektissa työskentelystä ei ollut kenelläkään aiempaa kokemusta. Kireä aikataulu, viikottaiset ryhmätapaamiset ohjaajan kanssa ja yhteistyö yrityksen kanssa lisäsivät paineita toteuttaa projekti onnistuneesti. Joutuimme miettimään, mitkä asiat olivat projektin kannalta oleellisimpia ja priorisoimaan niiden toteutuksen. Ohjelmoidessamme huomasimme, että muidenkin ryhmän jäsenten tuli ymmärtää koodia, jotta pystyimme varmistamaan yhteensopivuuden.

Jälkeenpäin katsottuna laite olisi pitänyt olla valmiina jo kuukautta ennen demopäivää. Tällöin aikaa olisi jäänyt enemmän laitteen välttämättömien ominaisuuksien hiontaan, testaukseen ja huomaamattomien ongelmien löytämiseen ja korjaamiseen. Tästä huolimatta projekti oli tavoitteisiin nähden hyvin onnistunut.

Lähteet

1. MakerCase. Viitattu: 31.8.2017. <http://www.makercase.com/>
2. Sziklai pair. Viitattu: 28.8.2017. https://en.wikipedia.org/wiki/Sziklai_pair
3. Arduino as In-System Programmer. Viitattu: 28.8.2017.
<https://www.arduino.cc/en/Tutorial/ArduinoISP>
4. PID-kirjasto. Kirjaston versio: 1.2.1. Viitattu: 25.8.2017. <https://github.com/br3ttb/Arduino-PID-Library>
5. Ziegler-Nichols -menetelmä. Viitattu: 25.8.2017.
https://en.wikipedia.org/wiki/PID_controller#Ziegler.E2.80.93Nichols_method