



Satakunnan ammattikorkeakoulu  
Satakunta University of Applied Sciences

MIKA PALOMÄKI

# **3D-TULOSTIMEN POISTOILMAN AUTOMATISOINTI**

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN  
TUTKINTO-OHJELMA  
2023

## TIIVISTELMÄ

Palomäki, Mika: 3D-tulostimen poistoilman automatisointi

Opinnäytetyö, AMK

Tutkinto-ohjelma: Sähkö- ja automaatiotekniikka

Kesäkuu, 2023

Sivumäärä: 88

Projektin tavoite oli tehdä 3D-tulostimen ympärille kammio, jossa voitaisiin kontrolloida vaaralliseksi luokiteltuja nanohiukkasia ja muita kaasuja, joita vapautuu, kun sulatetaan muovia. Tavoitteena oli myös saada lämpötila pysymään tasaisena kammion sisäpuolella. Kammioon sisä- ja ulkopuolelle asennettiin lämpö-, kosteus- ja ilmanlaatuanturit, joita kontrolloitiin Arduino Uno -mikro-ohjaimella. Poistoilmatuulettimena käytettiin 140 mm PC-tuuletinta, jota projektin edetessä modifioitiin imukykyisemmäksi HEPA-suodattimen läpi, jotta lämpötilan kontrolloiminen olisi onnistunut.

Tuulettimen modifiointi imuriksi paransi huomattavasti imutehoa HEPA-suodattimen läpi, mutta se ei riittänyt pitämään kammion lämpötilaa tasaisena. Projektin loppuvaiheissa toinen ilmanlaatuantureista oli mennyt rikki, joten ilmanlaatuarvot kammion ulkopuolelta jäivät ottamatta. Tuloksissa kuitenkin ilmeni, että tuuletus on hyvinkin tärkeää, kun 3D-tulostetaan ABS-muovia. Huonosti tuuletetussa tilassa nanohiukkaset kerääntyvät 3D-tulostimen ympärille, mikä kasvattaa terveydellisiä riskejä.

Avainsanat: 3D-tulostin, Nanohiukkanen, Arduino, Poistoilmatuuletus

## Abstract

Palomäki, Mika: 3D printer exhaust air automation

Bachelor's thesis

Degree programme: Electrical and automation engineering

June, 2023

Number of pages: 88

The goal of the project was to make a chamber around a 3D printer, where it would be possible to control nanoparticles classified as dangerous and other gases that are released when plastic is melted. The goal was also to keep the temperature stable inside the chamber. Temperature, humidity and air quality sensors were installed inside and outside of the chamber, where they were controlled with an Arduino Uno microcontroller.

A 140 mm PC fan was used as an exhaust air fan, which was modified to be more absorbent through a HEPA filter in order to control the temperature successfully as the project progressed.

Modifying the fan into a vacuum greatly improved the suction power through the HEPA filter, but it was not enough to keep the chamber temperature stable. In the final stages of the project, one of the air quality sensors had broken, so the air quality values from outside of the chamber were not recorded. However, the results showed that ventilation is very important when 3D printing ABS plastic. In a poorly ventilated space, nanoparticles accumulate around the 3D printer, which increases health risks.

Keywords: 3D-printer, Nanoparticles, Arduino, Exhaust ventilation

# SISÄLLYS

1	JOHDANTO	4
2	3D-TULOSTUKSEN HISTORIA	5
3	3D-TULOSTUSMATERIAALIT JA HAITAT	7
	3.1 Savonia-ammattikorkeakoulun päästötutkimus	8
	3.2 WASP ja Alveo-3D päästötutkimukset	13
4	3D-TULOSTUS TURVALLISESTI	14
	4.1 Materiaalit ja kierrätys	15
	4.2 Turvallisuus	16
5	ARDUINO UNO	16
	5.1 I2C-liitäntä	18
	5.2 Ilmanlaadun mittaaminen	22
6	TUTUSTUMINEN ARDUINOON	23
	6.1 Haastavammat harjoitukset	24
	6.2 Toteutus arduinolla	26
7	TARVITTAVAT KOMPONENTIT	29
8	TULOSTIN, KAMMIO JA POISTOILMA	36
	8.1 Kammio	40
	8.2 Kammion toteutus	45
	8.3 Ensimmäinen suunnitelma ja 3D-tulostus.	49
9	KÄYTTÖLIITTYMÄ JA OHJELMOINTI	58
	9.1 Ohjelman suunnittelu	60
	9.2 Nappikoodin ohjelmointi	64
	9.3 Ulkoasu arduinolle	68
10	TULOKSET	71
	10.1 Testitulokset ABS	72
	10.2 Testi tulokset PLA	73
11	POHDINTA	75
	LIITE 1	79

## 1 JOHDANTO

3D-tulostimet ovat yleistyneet kodeissa. Koska vietämme yli 50 % päivästä kodeissamme sisällä, olisi tärkeää huolehtia sisäilman puhtaudesta. ABS-muovia tulostettaessa vapautuu kemikaaleja, nanohiukkasia ja mahdollisesti myös hääkää, mitä ei sopisi päästää leviämään huonetiloihin tai hengitysilmaan. Sopivin paikka 3D-tulostimelle olisi tila missä ei oleskeltaisi jatkuvasti ja ilma vaihtuisi mahdollisimman hyvin. Kaikilla kotitalouksilla ei kuitenkaan ole tällaisia tiloja. (Väisänen, 2017, s. 14.)

3D-tulostimissa ei tule mukana kammiota, jossa saadaan eristettyä tulostuksessa syntyviä haitallisia kemikaaleja, mutta on mahdollista ostaa koteloitu 3D-tulostin, mikä nostattaa 3D-tulostimen hintaa huomattavasti. ABS-muovin tulostaminen ilman kammiota on terveystriski.

## 2 3D-TULOSTUKSEN HISTORIA

3D-tulostimet nykypäivänä ovat suhteellisen edullisia. Ne eivät vie paljoa tilaa ja miltei mielikuvitus on rajana projekteissa, joita niillä pystyy tekemään. Tällä hetkellä 3D-tulostimet mahtuvat helposti työpöydälle, mutta 30 vuotta sitten 3D-tulostimet olivat työpöydän kokoisia, maksoivat omakotitalon verran ja projektit olivat hyvin rajoitettuja. 1980-luvun puolivälistä lähtien 3D-tulostus on kehittynyt. (3Dsourced, 2021.)

Toukokuussa 1981 Hideo Kodama julkaisi yksityiskohtia nopeasta prototyypitekniikasta (engl. Rapid prototyping). Tämä tutkimus oli ensimmäinen kirjallisuustuotos kerros kerrokselta tapahtuvasta valmistuksesta. Kodama yritti saada patenttia fotopolymerteekniikkaan, jossa astiassa tasaisesti levitettynä oleva hartsi altistetaan ultravioletivalolle, minkä seurauksena valolle altistettu kohta kovettuu. Patenttia tähän tekniikkaan Kodamalle ei koskaan myönnetty. (3Dsource, 2021.)

Chuck Hull oli ensimmäinen, joka suunnitteli ja rakensi prototyypin 3D-tulostimesta vuonna 1983. Chuck Hull haki patenttia 1984 ja se hyväksyttiin vuonna 1986. (3Dsource, 2021.)

Chuck Hull perusti maailman ensimmäisen 3D-tulostinyrityksen, 3D Systems Corporationin. Yritys on edelleenkin suurin ja menestynein 3D-tulostusalalla oleva yritys. Ensimmäinen 3D-tulostin julkaistiin markkinoille vuonna 1987 (Kuva 1). Tulostinta kutsuttiin nimellä SLA-1 Stereolithography. (Redaktion, D. 2020.)



*Kuva 1. SLA-1 Sterolithography ensimmäinen markkinoille ilmestynyt 3D-tulostin. (3Dprint.com, 2015)*

Muutaman vuoden jälkeen Hullin keksinnöistä, kehitettiin uusi 3D-tulostusmenetelmä, SLS (Selective Laser Sintering). Tekniikan kehittäjä oli Carl Deckard vuonna 1987. SLS-menetelmässä esilämmitettyä pulveria levitetään tasaisia ohuita kerroksia ja jokainen kerros sulatetaan haluttuun muotoon yhteen kerroksina käyttäen suurteholasaria. (3Dsource, 2021.)

Vuonna 1988 Scott Crump patentoi FDM-tulostusteknologian (Fused Deposition Modeling), joka on tällä hetkellä yleisin koti- ja harrastekäytössä oleva 3D-tulostustekniikka. FDM-prosessissa filamenttilankaa syötetään kuumen suuttimen läpi. Sulanut filamentti pursotetaan alustalle, jossa filamentti kovettuu kerros kerrokselta haluttuun muotoon. Stratasys on Scott Crumpin perustama yritys, joka on nykypäivänäkin tunnetuimpia 3D-tulostusalan yrityksiä. (3Dsource, 2021.)

### 3 3D-TULOSTUSMATERIAALIT JA HAITAT

3D-tulostamisessa on tärkeää tietää mitä materiaaleja kannattaa käyttää, jotta saavutettaisiin haluttu lopputulos. Materiaaleja on saatavilla monia ja jokaisella materiaalilla on erilaiset ominaisuudet. Ruoka-astian 3D-tulostamiseen kannattaa käyttää elintarvikehyväksytyjä materiaaleja. (Ubel, M. 2021.)

PLA (Polylaktidi) on biohajoava ja ympäristöystävällinen 3D-tulostusmateriaali. PLA:sta valmistettuja pakkauksia voi nähdä kaupan vihannes- ja hedelmähyillyillä. PLA tulostaminen FDM-periaatteella on käyttäjän kannalta helppoa. Riittää, että tulostuslämpötila on oikea ja filamenttiin ei ole päässyt kosteutta. PLA:n sulamisesta aiheutuvat hajut ovat hyvin mietoja, myös partikkeleita erittyy huomattavan vähän tulostettaessa. (Filamentti.com)

ABS:sta (Akrylibutadieenistyreeni) on valmistettu mm. legopalikoita, laatikoita ja lastenleluja. Tavarointa, jotka vaativat iskunkestävyyttä, muotonsa pitävyyttä ja kovuutta. Sulaessaan ABS:stä erittyy myrkyllisiä kemikaaleja ja mikropartikkeleita, joten ilmanvaihdon täytyisi olla kunnossa, kun ABS-materiaalilla tulostetaan. (Filametti.com)

TPU:n (Termoplastinen polyuretaani) soveltuvuuskohteita ovat mm. sähkötyökalut, urheiluvälineet ja suojakuoret matkapuhelimiin. Koska TPU on kumimaista ja erittäin joustavaa materiaalia, soveltuu se hyvin esineisiin, jotka vaativat joustavuutta ja kimmoisuutta. (Filametti.com)

Nylon (Nylon Polyamide) on 3D-tulostusmateriaalina vahva, kestävä ja monipuolinen. Se on sopiva materiaali teknisten osien tulostamiseen. Sillä on erittäin pieni kitkakerroin, mikä tekee siitä ihanteellisen tulostusmateriaalin osille, jotka vaativat hyvää veto- ja mekaanista lujuutta. (Ubel, M. 2021.)



### 3.1 Savonia-ammattikorkeakoulun päästötutkimus

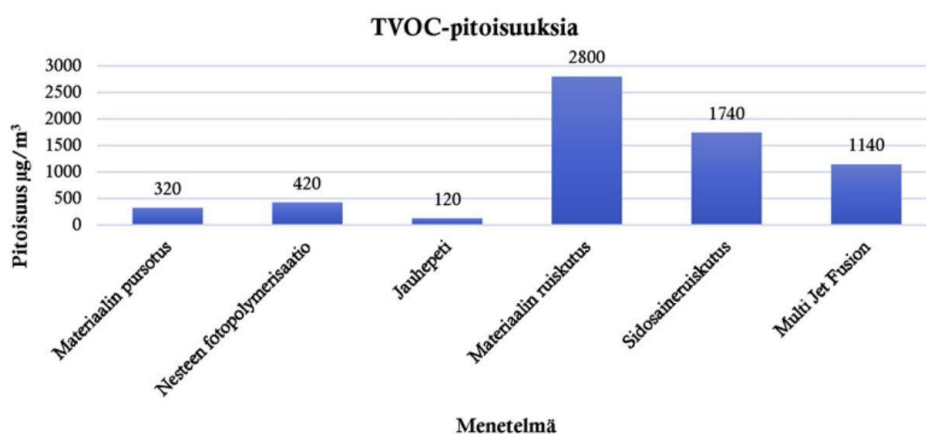
Terveydelle syntyviä haitallisia altisteita 3D-tulostusprosessista on tutkittu jo vuosikymmeniä. Savonia-ammattikorkeakoulun 3D-laboratoriossa sekä yrityksissä, jotka hyödyntävät 3D-tulostustekniikka, on tehty tutkimus terveydelle haitallisten altisteiden määrästä. Materiaaleista haihtuvista orgaanisista yhdisteistä käytetään lyhennettä VOC (Volatile Organic Compound) ja niiden kokonaispitoisuudesta TVOC (Total Volatile Organic Compound). Tutkimuksen mukaan 3D-tulostuksessa syntyy terveydelle haitallisia altisteita, mutta terveydelle haitallisten altisteiden esiintyvyys on melko matalaa. Altisteet ovat kemiallisia yhdisteitä ja hiukkasia ja terveystvaikutteet ilmenevät vasta pitkäaikaisesta altistumisesta. (Väisänen, A. 2017, s. 5.)

3D-tulostuslaitteita hankittaessa täytyy ottaa huomioon niiden sijainti. (Kuva 2). Tulostettaessa vapautuu päästöjä, jotka tulee huomioida. Päästöjen vapautumiseen vaikuttaa monet tekijät, kuten tulostuslämpötila, menetelmä, laite ja käytettävä tulostusmateriaali. Päästöjen pitoisuuteen vaikuttaa ilmavaihto ja kohdepoistojen tehokkuus. (Väisänen, A. 2017, s. 6.)

Taso	Pitoisuusalue ( $\mu\text{g}/\text{m}^3$ )	Tulkinta/toimenpiteet
1	< 300	Tilanne hyvä
2	> 300–1000	Tilanne kohtuullinen (mikäli yksittäisten yhdisteiden raja-arvot ei ylity)
3	> 1000–3000	Pitoisuus koholla
4	> 3000–10 000	Pitoisuus liiallinen
5	> 10 000	Pitoisuus ei hyväksyttävä

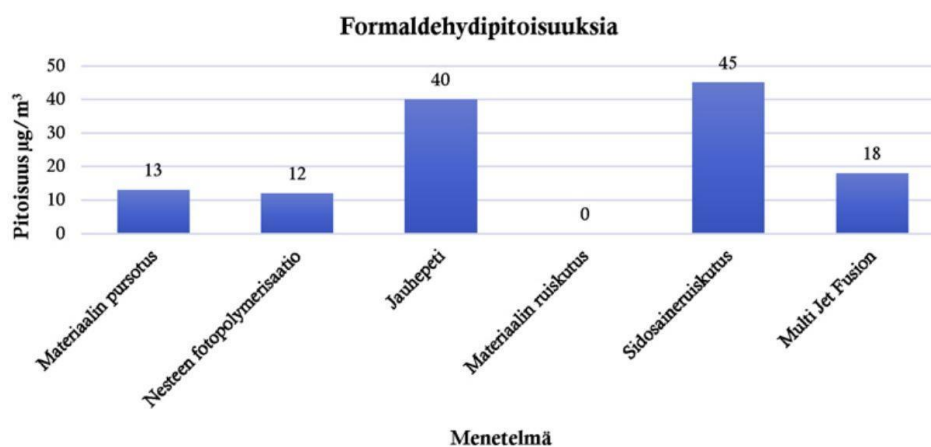
Kuva 2. TVOC-tasot ja tulkinta (Väisänen, A. 2017, s. 6.)

Haihtuvat orgaaniset yhdisteet ovat yleisimpiä muovista peräisin olevia altisteita ja voivat aiheuttaa allergioita, ärsytysoireita, hengitystiesairauksia ja jopa syöpää. Lämpöhajotessa 3D-tulostettaessa muoviyhdisteistä vapautuu yhdisteitä. (VOC-yhdisteiden yhteenlaskettua pitoisuutta) tarkastellaan TVOC-pitoisuuden avulla. Tyypillisesti yhdisteitä esiintyy matalina pitoisuuksina, mutta yksittäisiä yhdisteitä on useita kymmeniä. (Kuva 3). LIVA-hankkeessa mitattuja TVOC-pitoisuuksia eri 3D-tulostusmenetelmillä. (Väisänen, A. 2017, s. 6.)



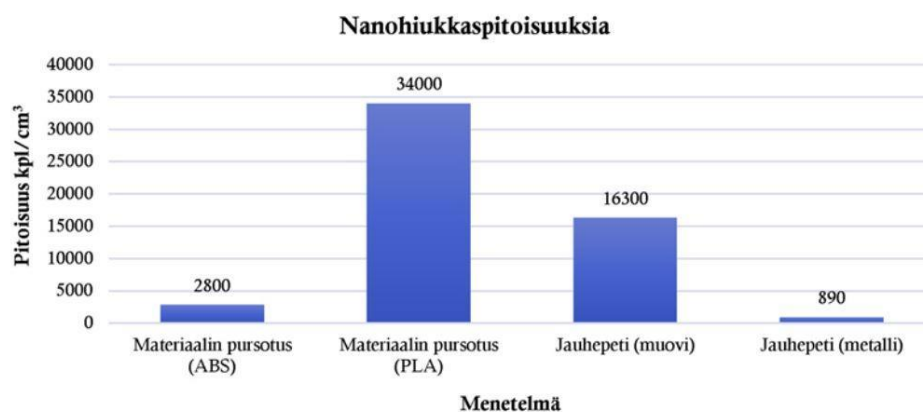
Kuva 3. TVOC-pitoisuuksia eri 3D-tulostusmenetelmistä. (Väisänen, A. 2017, s. 7.)

Formaldehydi on luokiteltu 1B-luokan karsinogeeniksi, minkä oletetaan aiheuttavan syöpää ja 2 luokan mutageeniksi, minkä taas oletetaan vaurioittavan perimää. Formaldehydiä syntyy erilaisten muovien lämpöhajoamistuotteina. Kuvassa 4 on LIVA-hankkeessa mitattuja formaldehydipitoisuuksia eri tulostusmenetelmissä. Pitoisuudet ovat kylläkin pieniä, joten riski syövän kehittymiselle on pieni, vaikka altistumista olisi pidempikin jakso. Haitalliseksi luokiteltu Formaldehydipitoisuus on  $370 \mu\text{g}/\text{m}^3$ , joten riski saada terveydellisiä haittoja ei ole merkittävä. (Väisänen, 2017, s. 7.)

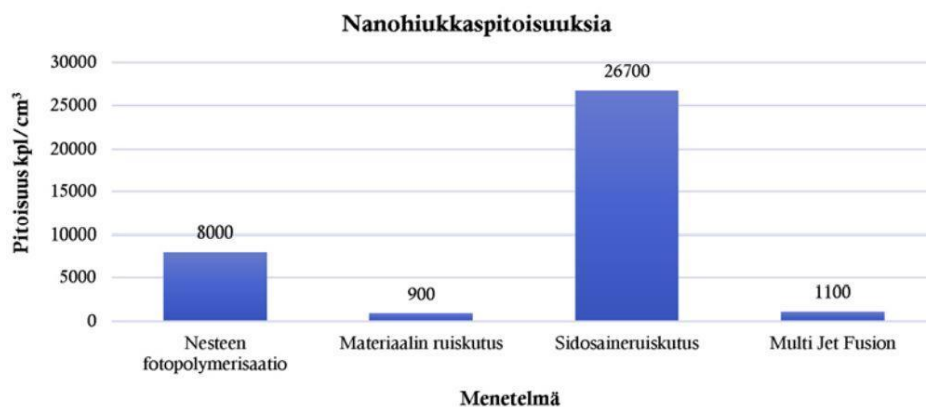


Kuva 4. Formaldehydipitoisuus eri 3D-tulostusmenetelmissä (Väisänen, A. 2017, s. 7.)

Muovin lämpöhajoamisessa ja nestemäisten tulostusmateriaalien haihtuessa ja näiden kemikaalien kiinnittyttyä toisiinsa muodostavat ne nanohiukkasia. Nanohiukkaset on yhdistetty hengitys- ja sydänsairauksiin, koska nanohiukkaset ovat tarpeeksi pieniä kulkeutumaan hengitysteitse suoraan verenkiertoon. Nanohiukkasten terveysvaikutuksista ei ole vielä vankkaa tietoa (Kuva 5 ja 6). Terveyslaitoksen tavoitetasoksi on ehdotettu muoviperäisille nanohiukkasille alle 40 000 hiukkasta ja metalliperäisille alle 20 000 hiukkasta. (Väisänen, A. 2017, s. 8.)



Kuva 5. LIVA-hankeen mitattuja nanohiukkaspitoisuuksia eri materiaaleista FDM- ja DLP menetelmällä. (Väisänen, A. 2017, s. 8.). Luultavammin kuvaajassa on mennyt PLA ja ABS sekaisin.



Kuva 6. LIVA-hankkeen mitattuja nanohiukkaspitoisuuksia eri tulostusmenetelmillä. (Väisänen, A. 2017, s. 8.)

### 3.2 WASP ja Alveo-3D päästötutkimukset

3DSafety.org:in ja italialaisen 3D-tulostinvalmistajan WASP:n tekemässä uudessa tutkimuksessa analysoitiin FDM-menetelmässä vapautuvien myrkyllisten VOC-yhdisteiden ja nanohiukkasten tarkkoja määriä. Laboratoriotutkimukset osoittivat, että ABS on huomattavasti myrkyllisempää, kuin PLA. PLA päästää myös vaarallisia VOC-yhdisteitä, jos sitä pursotetaan yli 200 asteen lämpötilassa. Materiaalit eri jälleenmyyjiltä vapauttavat hyvin erilaisia määriä VOC-yhdisteitä, vaikka niitä käytettiin samassa 3D-tulostimessa ja tulostettiin samoilla lämpö- ja nopeusarvoparametreilla. Toinen kriittinen näkökulma liittyy nanohiukkasiin, koska ABS-muovia 3D-tulostettaessa muodostui 3–30 kertaa enemmän nanohiukkasia, kuin PLA-muovilla tulostettaessa. Laboratoriotutkimuksissa osoitettiin myös, että kesti 10–30 minuuttia, että nanohiukkaspitoisuus palasi takaisin lähtötasolle, kun 3D-tulostin, oli lopettanut tulostuksen. (Sher, D. 2015.)

Alvaeo3D:n tekemässä tutkimuksessa käytettiin PLA- ja ABS-muoveja. Filamentit tilattiin samalta valmistajalta, jotta ominaisuudet olisivat mahdollisimman samanlaiset. Eri valmistajien filamenttien välillä on mahdollista löytää suuria eroja koostumuksessa. Kukin valmistaja voi lisätä lisäaineita, joilla saadaan erityisiä ominaisuuksia, esimerkiksi väriä, mekaanista-, lämmön-, rakennekestävyyttä jne. Tutkimuksessa PLA-langan pursotuslämpötila oli 215°C ja tulostuspedin +60°C. ABS-langan pursotuslämpötila oli 250°C ja lämpötason 110°C. Testi tehtiin 3D-tulostimelle suunnitellussa kammiossa aktivoimatta ilmansuodatusjärjestelmää. Ennen kumpaakin testiä mitattiin nanohiukkasten määrä ilmasta. Määrä oli kummassakin testissä 4000kpl/cm<sup>3</sup>. PLA:n tulostuksessa saatiin maksimipitoisuudeksi 66000kpl/cm<sup>3</sup> keskimäärin noin 15000 kpl/cm<sup>3</sup>. ABS:n tulostuksessa saatiin maksimipitoisuudeksi 260000 kpl/cm<sup>3</sup> keskimäärin noin 150000 kpl/cm<sup>3</sup>. (Martini, L. 2023.)

## 4 3D-TULOSTUS TURVALLISESTI

3D-tulostaminen kasvattaa suosiotansa. Sen myötä tulee huolenaiheita, jotka liittyvät 3D-tulostettaessa vapautuviin höyryihin sekä niiden aiheuttamiin mahdollisiin terveysvaaroihin. Monet tutkimukset ovat osoittaneet, että vapautuvissa höyryissä piilee riski ihmisten terveydelle. Sen takia on hyvä tietää, kuinka pystyä välttämään tämä riski. Näiden riskien välttämiseksi on monia mahdollisuuksia koti- tai työympäristössä. (Ekaran, S. 2022.)

Yksi keinoista on varmistaa, että ilmanvaihto 3D-tulostimen ympärillä on riittävä, jotta höyryt ja hiukkaset eivät pääse kertymään samaan paikkaan. Tämän voi tehdä joko ikkunoita avaamalla, kierrättämällä ilmaa tuulettimella tai asentamalla poistoilman lähelle 3D-tulostinta. Turvallisin tapa on asentaa 3D-tulostimen ympärille kammio, jotta terveydelliset riskitekijät ja hajut saadaan pysymään kammion sisällä ja siepattua tuulettimella suodattimeen tai vaihtoehtoisesti ohjattua ulos ilmanvaihtokanavia pitkin. Tässä järjestelmässä täytyy huomioida paloturvallisuus. (Ekaran, S. 2022.)

### 4.1 Materiaalit ja kierrätys

Käytä turvallisimmiksi luokiteltuja filamenttimateriaaleja. ABS on yleisempiä käytettyjä filamentteja sen iskunkestävyyden ja kovuuden takia, mutta siitä erittyy paljon myrkyllisiä höyryjä. Myös haju, jota erittyy muovin sulaessa tulostuspäässä, on epämiellyttävä. Hajun takia ABS-muovia olisi hyvä 3D-tulostaa suljetussa ympäristössä. Hyvä vaihtoehto samankaltaisten ominaisuuksien osalta oli PETG. Myös filamenttien säilytykseen on (poista sana: myös) syytä kiinnittää huomiota. Filamentti on syytä pitää erossa lämpimistä paikoista, kuten suoralta auringonvalolta, näin välttyä filamentin turhalta kaasuuntumiselta. (Ekaran, S. 2022.)

Jokainen meistä yrittää tehdä kaikkien elämästä vihreämpää tavalla tai toisella. 3D-tulostaminen on riippuvainen energiankäytöstä ja hiilijalanjäljen pienentäminen on tänä päivänä todella tärkeää. On hyvä tapa oppia mitkä filamentit ovat kierrätettäviä ja mitkä eivät ole. PLA on pääasiassa valmistettu kasvitärkkelyksestä, joten se on

100 % kompostoitava filamentti. PETG ei ole biohajoava mutta sen voi kierrättää. Jos olet ympäristöystävällinen ja haluat pienentää hiilijalanjälkeä, käytä vain PLA filamentteja, muut filamentit eivät ole biohajoavia tai helposti kierrätettäviä. On myös mahdollista uudelleen käyttää PLA filamenttia siihen löytyvillä silppureilla ja puristimilla. (Hamill, S. 2022.)

#### 4.2 Turvallisuus

Hajuisten materiaalien, kuten ABS muovin työstämisessä voi olla hyödyllistä käyttää hengityssuojaimia. Suojaimiksi kannattaa valita siihen tarkoitukseen suunniteltuja suojaimia, jotka suojaavat höyryiltä ja nanohiukkasilta.

Suosittelavaa on myös käyttää suojalaseja suojautuakseen tulostuksessa vapautuvilta hiukkasilta. Kappaleiden jälkikäsittelyssä on syytä käyttää hanskoja, koska kappaleiden tulostetut reunat voivat olla teräviä.

3D-tulostimen säännöllinen puhdistaminen on myös suositeltavaa. Sillä saadaan estettyä pölyn ja hiukkasten pakkautuminen, mikä voi aiheuttaa pölyn leviämisen tulostuksien aikana takaisin ilmaan. (Ekaran, S. 2022.)

Koska 3D-tulostimen pursotin ja tulostuspeti voivat kuumentua jopa 250/110 asteeseen tai jopa korkeammaksi, on siinä samat vaarat kuin hellassa ja missä tahansa muussa laitteessa, joka kuumenee. On suositeltavaa ostaa ylimääräinen palovaroitin ja asentaa se suoraan 3D-tulostimen yläpuolelle, myös palosammutin tai palopeite ja laittaa ne 3D-tulostimen lähetyville. Myös markkinoilla on 3D-tulostimien kammioihin asennettavia lähettämiä, joilla saadaan yhteys suoraan älypuhelimiin tai kammioon sisälle asennettava palosammutinputki. Putken toisessa päässä on lämpötilalämpöisin, joka reagoi räjäyttämällä ja levittäen palonestoainetta, kun lämpötila nousee yli 105°C kammion sisäpuolella. (Czeiszperger, M. 2020.)

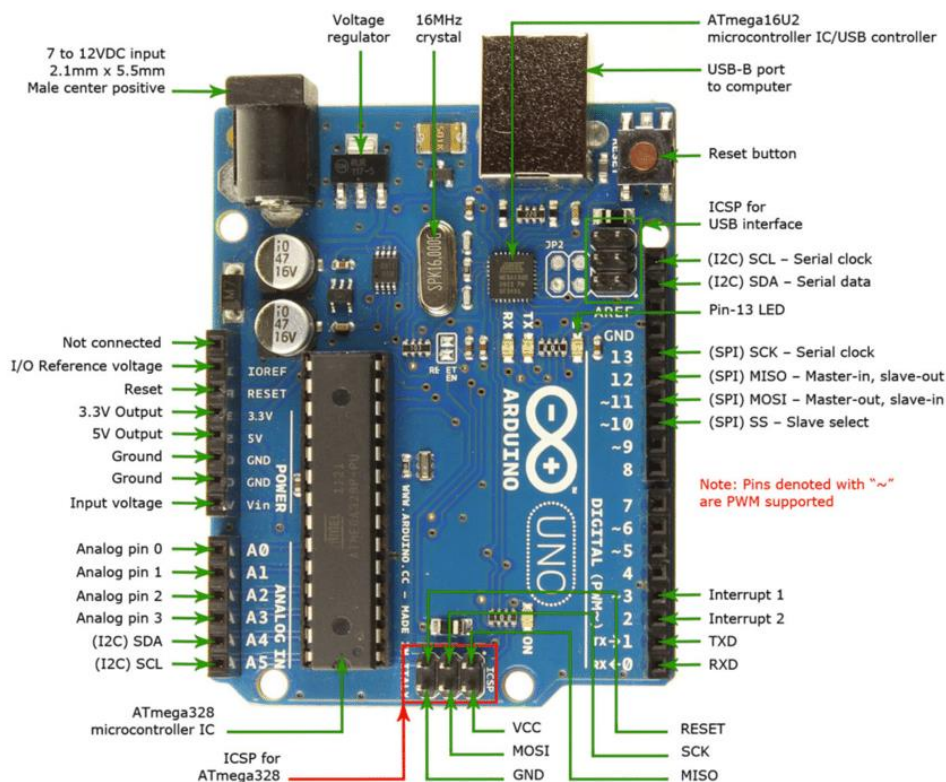
## 5 ARDUINO UNO

Arduino on avoimen lähdekoodin elektroniikka-alusta, joka perustuu helppokäyttöisiin laitteistoihin ja ohjelmointiin. Arduinolla pystyy lukemaan tuloja antureista, painikkeista tai älypuhelimien viesteistä ja ne pystytään muuttamaan lähdeiksi, joilla voidaan kytkeä ledi päälle tai vaikka moottori. (Arduino, 2023)

On kolme tapaa, millä Arduino voidaan kytkeä virtoihin (Kuva 7). Joko USB-kaapelin kautta, jossa maksimi jännite saa olla 5 voltia, verkkovirralla toimivalla virtalähteellä tai paristolla. Jälkimmäisissä suositeltava jännite on 7–12 voltia. Arduinoa voidaan myös kytkeä jopa 20 volttiin, mutta valmistaja ei suosittele käyttämään yli 12 voltissa pitkäaikaisia jaksoja, koska komponentit voivat ylikuumentua.

Paristolla kytkentä tapahtuu Vin-pinnin kautta hyppyjohtimilla. 5V ja 3.3V syöttää ulkoisille komponenteille virran. Arduinosta löytyy viisi kappaletta (GND) maapinniä, jotka ovat toisiinsa yhteydessä. Arduinon voi joko ohjelmallisesti resetoida tai manuaalisesti napista. (Zait, 2018.)





Kuva 7. Arduino Uno ja sen liitäntäpinnit. (Jameco.com, 2015.)

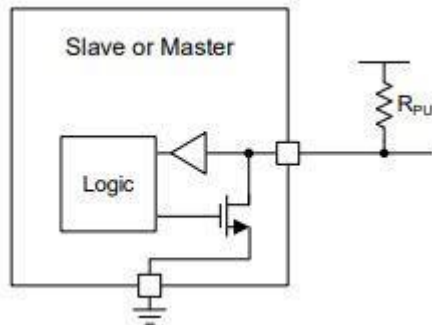
Arduinossa on kuusi analogista pinniä, jotka hyödyntävät ADC:tä engl. (Analog to Digital Converter). ADC on elektroninen piiri mitä käytetään muuntamaan analogista signaalia digitaaliseksi. ADC:lla on 10-bitin resoluutio, se voi esittää saadun analogisen jännitteen 1024 eri digitaalisella tasolla. ADC muuntaa jännitteen biteiksi, jotta mikroprosessori ymmärtäisi sen. Pinnit 0–13 toimivat digitaalisina tulo- ja lähtöpinneinä. 3, 5, 6 ja 9 on PWM-ominaisuus. (Zait, A. 2018.)

## 5.1 I2C-liitäntä

I2C (Serial communication protocol) on yleinen protokolla. Sillä mahdollistetaan sarjaviestintä kahden johdon avulla monien laitteiden kanssa. (Murray, M.)

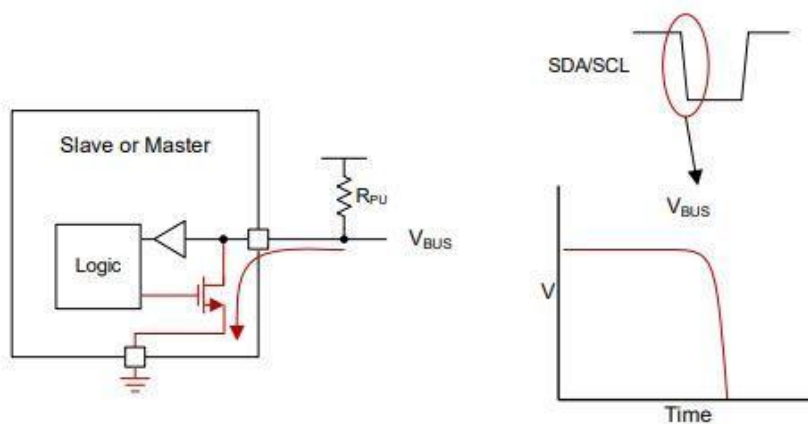
I2C:n ymmärtämiseksi on tärkeää tietää, mistä sisäinen rakenne koostuu (Kuva 8). Puskurista syöttötietojen lukemiseen ja avattavasta FET transistorista tietojen lähettämiseen. Laite kykenee vain vetämään väylälinjan matalalle tai vapauttamaan

väylälinjan ja antamaan vetovastuksen nostaa jännitettä. Ominaisuus antaa mahdollisuuden kaksisuuntaiseen viestittämiseen. (Valdez, J & Becker, J. 2015, s.2.)



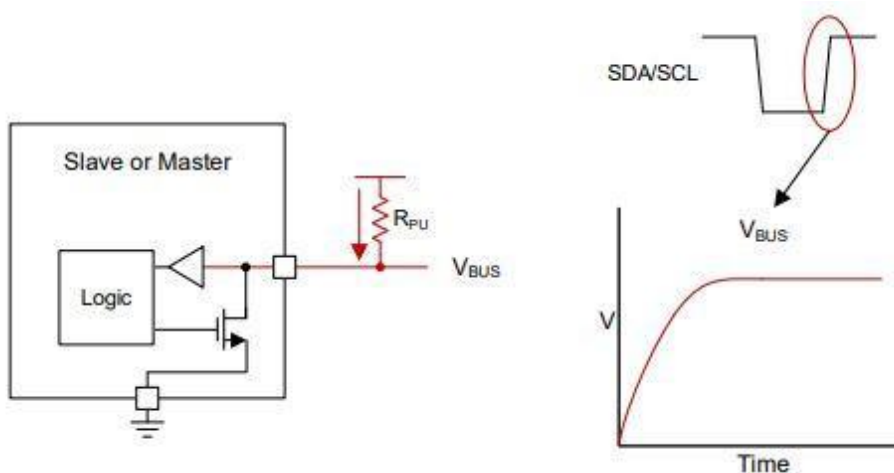
Kuva 8. SDA/SCL-linjan sisäinen perusrakenne. (Valdez, J & Becker, J. 2015, s.2.)

Virran virtaus väylän vetäminen matalaksi. (Kuva 9). Logiikka, joka haluaa lähettää matalan signaalin aktivoi alas vedettävän FET transistorin, mikä tarjoaa lyhyen maadoituksen ja vetää linjan matalalle. (Valdez, J & Becker, J. 2015, s.2.)



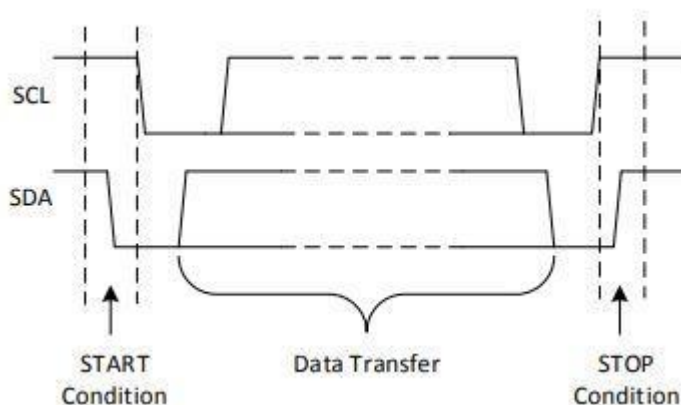
Kuva 9. Väylälinjan vetäminen matalalle. (Valdez, J & Becker, J. 2015, s.2.)

Kuvassa 10. mikäli slave tai master haluaa lähettää logiikan korkealle, voi se vapauttaa väylän vain sammuttamalla FET-transistorin. Tämä jättää väylän kellumaan, ja vetovastus vetää jännitteen, joka tulkitaan korkeaksi. (Valdez, J & Becker, J. 2015, s.3.)



Kuva 10. Väylälinjan vapauttaminen korkeaksi (Valdez, J & Becker, J. 2015, s.3.)

Kuvassa 11. Kommunikointi I2C liitännän kautta käynnistyy, kun isäntä lähettää ALOITUS-ehdon ja päättyy, kun isäntä lähettää LOPETUS-ehdon. Korkeasta matalaan SDA-linjalla siirtyminen, kun SCL on korkea määrittää se ALOITUS-ehdon ja kun SCL on korkea määrittää se LOPETUS-ehdon. (Valdez, J & Becker, J. 2015, s.3.)



Kuva 11. ALOITUS- ja LOPETUS-ehdot. (Valdez, J & Becker, J. 2015, s.4.)

## 5.2 Ilmanlaadun mittaaminen

Ilmanlaatua mitattaessa tutkitaan, kuinka puhdasta tai saastunutta ilma on. Saastunut ilma voi olla haitallista terveydelle tai ympäristölle. Ilmanlaatua mitataan ilmanlaatuindeksillä AQI (Air Quality Index). AQI arvot kulkevat 0–500. Ilmakehässä on kahta kaasua; typpeä ja happea, mitkä ovat välttämättömiä elämälle. Ilmassa on kuitenkin monia muita kaasuja ja hiukkasia. AQI:lla seurataan 5 suurinta ilmansaastetta: alailmakehän otsoni, hiilimonoksidi, rikkidioksidi, typpioksidi ja ilmassa olevia hiukkasia tai aerosolia. Alailmakehän otsoni ja hiukkaset ovat epäpuhtauksia, jotka aiheuttavat suurimmat riskit ihmisten terveydelle. (SciJinks, 2023.)

AQI mittapuue kulkee 0–500 välillä. (Kuva 12). Mittapuussa alle 50 edustaa hyvää ilmanlaatua ja yli 300 edustaa epäterveellistä ilmanlaatua. Mittapuussa alle 100 pidetään yleensä tyydyttävänä kansanterveyden suojelemiseksi sovellettavan lyhyen aikavälin kansallisen ilmanlaatuunormien tasoa. Tulos, joka ylittää 100 pidetään epäterveellisinä herkille ihmisille ja 150 ylittävät ovat epäterveellisiä kaikille. (AirNow, 2023.)

Air Quality Index Levels of Health Concern	Numerical Value	Meaning
Good	0 to 50	Air quality is considered satisfactory, and air pollution poses little or no risk.
Moderate	51 to 100	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is not likely to be affected.
Unhealthy	151 to 200	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201 to 300	Health alert: everyone may experience more serious health effects.
Hazardous	301 to 500	Health warnings of emergency conditions. The entire population is more likely to be affected.

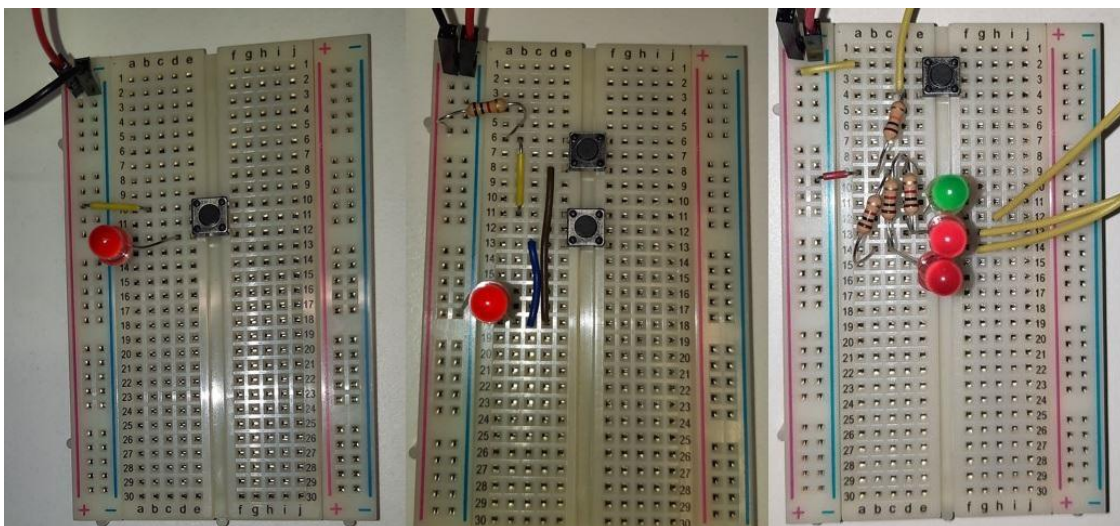
Kuva 12. Ilmanlaatuindeksi arvot ja selitys. (SciJinks, 2023).

Ilmanlaatuanturit ovat laitteita, joilla pystytään valvomaan ilmassa olevia haitallisia kaasua ja hiukkasia, jotka voivat isoissa määrin olla terveydelle haitallisia. Yleisimmät ilmanlaatuanturin kaasut ja epäpuhtaudet ovat hiukkaset, otsoni, rikkidioksidi, typpioksidi ja hiilimonoksidi. Ilmanlaatuanturin toiminta periaate on, että se laskee tietyn tilan läpi kulkevien hiukkasten koon ja määrän, joista lasketaan arvio, kuinka paljon kaasua ja hiukkasia kulkeutuu tietyn tilan läpi. Yleisesti käytetyt ilmanlaatuanturit käyttävät lasereita hiukkasten seurantaan. (Long, M. 2021.)

## 6 TUTUSTUMINEN ARDUINOON

COVID-19 takia oli tutustuminen Arduinoon jäänyt lähes mitättömäksi, koska kaikki ammattikorkeakoulut olivat etäopetuksessa, joten tutustuminen projektiin alkoi Arduinon aloituspakkauksella. Pakkauksessa oli hyvä valikoima erilaisia komponentteja ja myös ohjelmoinnin alkeet menivät askeleittain haastavammaksi. Harjoittelu alkoi aivan perusasioista ja ajan mittaan haastavammiksi.

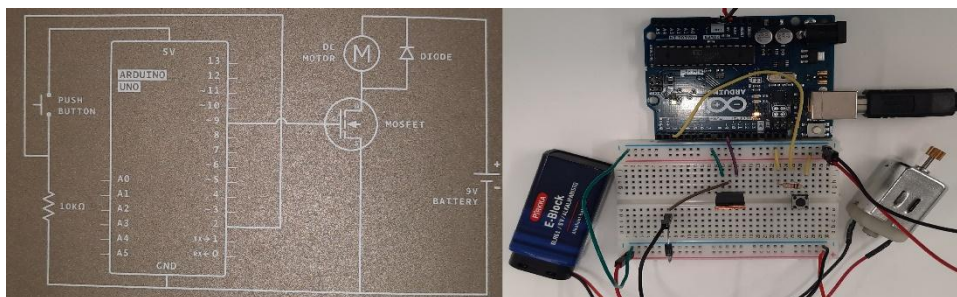
Kuvassa 13. vasemmalla on hyvin yksinkertainen virtapiiri, missä nappia painamalla pohjaan saadaan ledi syttymään (vastus vaihdettu johtimeen, ledi paloi todella himmeänä vastuksen kanssa). Kuvan 13 keskellä painamalla jompaakumpaa nappia saadaan ledi syttymään. Kuvassa 13 oikealla, tässä harjoituksessa on jo lyhyehkö koodi kirjoitettu, vihreä ledi syttyy, kun nappia ei ole painettu. Kun taas nappia painetaan, vihreä ledi sammuu ja yksi punainen ledi syttyy. Punainen ledi sammuu  $\frac{1}{4}$  sekunnin päästä ja sytyttää toisen punaisen ledin, kun tästä on kulunut  $\frac{1}{4}$  sekuntia punainen ledi sammuu ja ohjelma alkaa alusta. (Fitzgerald, S. & Shiloh, M. 2015 s.24–40).



Kuva 13. Arduinolla ensimmäiset harjoitukset.

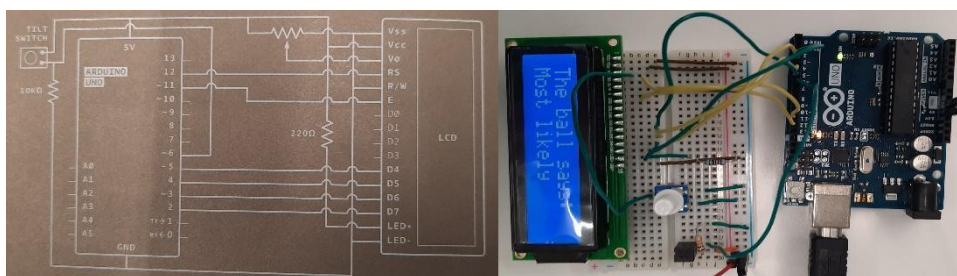
## 6.1 Haastavammat harjoitukset

Harjoituksessa (Kuva 14) 9 voltin pariston positiivinen puoli on yhdistetty DC moottorin positiiviseen napaan ja pariston negatiivinen napa tuodaan leipälaudan kautta hyppyjohtimilla Arduinin maapinniin. Napilla annetaan signaali Arduinin 2 pinniin, millä aktivoidaan 9 pinnin ja täten annetaan signaali transistorin vasempaan pinniin. Tätä pinniä transistorissa kutsutaan hilaksi. Jännitteen muutos hilaan muodostaa yhteyden kahden muun pinnin välille. Moottorin negatiivinen johdin kytketään transistorin keskimmäiseen pinniin, mitä kutsutaan nelukseksi. Transistorin oikeanpuoleinen pinni (lähde) kytketään Arduinin maapinniin. Diodilla estetään moottorin tuottaman takajännitteen paluun. (Fitzgerald, S. & Shiloh, M. 2015 s.95–101).



Kuva 14. Harjoituksessa nappia painamalla pohjassa moottori käynnistyy ja sammuu, kun napista irrotetaan.

Harjoituksessa (Kuva 15) kallistusanturin vasemmasta pinnistä kytketään johto 5 volttiin ja oikeanpuoleisesta pinnistä kytketään maahan 10 kilo-ohmin vastuksen kautta ja samasta kallistusanturin pinnistä kytketään johto Arduinon pinniin 6. LCD näytössä (RS) pinni ohjaa missä merkit näkyvät näytöllä (R/W) pinni asettaa näytön luku- tai kirjoitustilaan. (EN) pinni kertoo LCD-näytölle, että se vastaanottaa komennon. (D0-D7) on merkkitietojen lähettämistä varten näytölle. (D4-D7) vain harjoituksessa käytössä. (Vo) Pinni kytketään potentiometrin keskimmäiseen pinniin, vasen kytketään maahan ja oikea 5 volttiin. (Vss, LED- ja R/W) kytketään maahan. (Vcc) on LCD-virtalähde, joka yhdistetään 5 volttiin. Myös (LED+) kytketään 5 volttiin, mutta se kulkee 220 ohmi vastuksen kautta. Arduinon pinni 2 on kytketty LCD D7, pinni 3 LCD D6, pinni 4 LCD D5 ja pinni 5 LCD D4. (D4-D7) kertovat näytölle, mikä merkki näytetään. LCD-näytön (E) pinni on yhdistetty Arduinon pinni 11 ja LCD-näytön (RS) pinni yhdistämällä vielä Arduinon pinni 12 pystytään kirjoittamaan LCD näytölle. (Fitzgerald, S. & Shiloh, M. 2015 s.115–122).



Kuva 15. Näpäyttämällä kallistusanturia päältä päin LCD näytölle tulee jokin 8 valmiiksi kirjoitetusta tekstistä.



## 6.2 Toteutus arduinolla

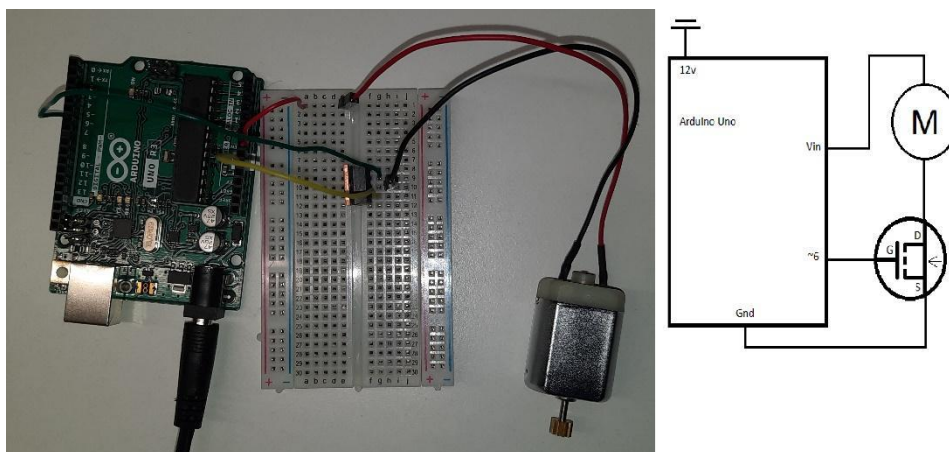
Arduino on kytketty 12 voltin adapteriin. Kuvassa 16. Vasemmassa on yleismittarilla katsottu Arduinosta ulos saatavan jännitteen määrä (3.3V) pinnistä on 3.3131 voltia. Keskimmaisessä kuvassa on katsottu (5V) pinnistä saatava jännitteen määrä mikä oli 5.0482 voltia ja oikeassa kuvassa on (Vin) pinnin ulos saatava jännitemäärä 11.377 voltia.



Kuva 16. Arduinosta ulostulo jännitteiden testaus yleismittarilla.

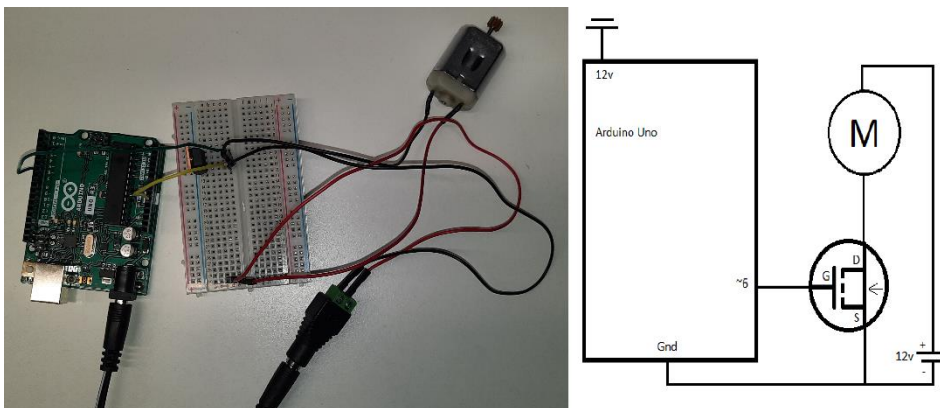
Pulssileveysmodulaatiossa, englanniksi Pulse-Width Modulation, (PWM) idea on katkoa virran määrää vaihtelevan levyisiksi pulsseiksi. Asian voi myös ajatella kytkemällä valonkatkaisijan pois päältä ja takaisin päälle, mutta on-, off-jaksoja täytyisi olla tuhansia kertoja sekunnissa, jopa kymmeniätuhansia kertoja riippuen sovelluksesta, johon sitä käytetään. (Lunkka, K. 2011, s.2.)

Kuvassa 17. Arduinossa on paikka 12 voltin adapterille ja siitä on myös mahdollista saada eri volttimääriä ulos, vaikka Arduino olisi kytketty 12 voltin adapteriin. Käyttämällä (Vin) pystyy ottamaan 12 voltia, (5V) saa 5 voltia ja (3.3V) tietysti 3.3 voltia. Koska kyseessä on 12 voltin tuuletin, käytettiin (Vin) pinniä. (Vin) Pinni yhdistetään tuulettimen plus nappaan. Tuulettimen negatiivinen napa yhdistetään transistorin nieluun (D). Transistorin Hila (G) yhdistetään Arduinon Pinni ~6 millä pystytään ohjaamaan tuulettimen nopeutta PWM signaalilla. (S) eli transistorin lähde yhdistetään Arduinon maanapaan. Kävi ilmi, että PWM ohjaus tällä kytkennällä aiheutti ikävän piipitysäänen aina, kun virtaa katkottiin, eli tuulettimen nopeutta hidastettiin. Mutta kun virtaa ei enää katkottu, piipittävää ääntä ei enää kuulunut. Tämä johtui siitä, että PWM on kuultavalla matalalla taajuudella. (Joutsenvirta, A.)



Kuva 17. Tuulettimen käyttö PWM-ohjauksella. Kuvassa DC-moottori esittää 12 voltin PC-tuuletinta.

Kuvassa 18. Adapteriin on lisätty haaroitin, josta yksi kytkeytyy Arduinoon. Haaroittimella saadaan virta tuotua Arduinoon ja siihen kiinnitettäviin komponentteihin ja toinen puoli haaroittimesta kytketään tuulettimen positiiviseen napaan ja negatiivinen Arduinon maahan. Kuvassa DC moottori esittää 12 voltin PC tuuletinta. Tällä kokoonpanolla ei tullut enää piipittävää ääntä.

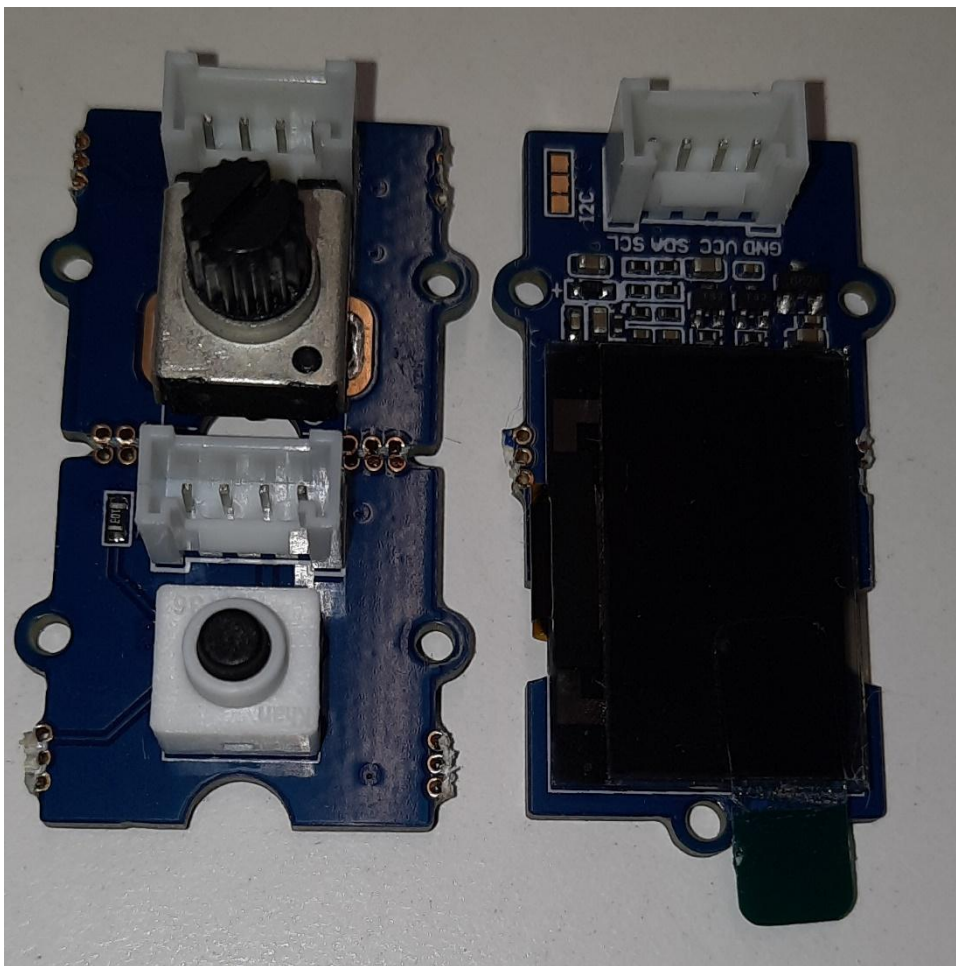


Kuva 18. Tuulettimen käyttö PWM-ohjauksella ilman piipittävää ääntä. Kuvassa DC-moottori esittää 12 voltin PC-tuuletinta.

## 7 TARVITTAVAT KOMPONENTIT

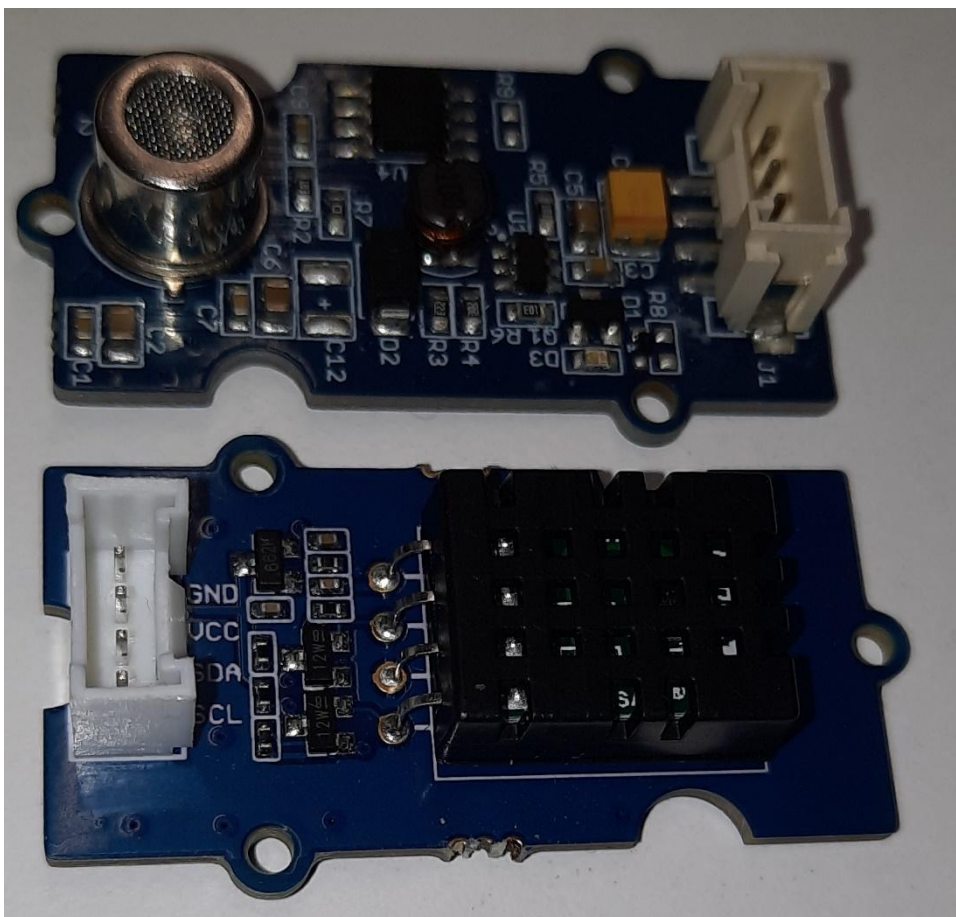
Ennen komponenttien tilausta piirrettiin paperille hahmotelma siitä, mitä tarvitaan, jotta tilaukset tulisivat yhdellä kertaa, eikä uutta tilausta tarvitsisi tehdä. Itse Arduinolla oli valmis anturipaketti, jota oli helppo hyödyntää tilausta tehdessä.

Käyttöliittymään tarvitaan näyttö mistä voidaan seurata lämpötila-, kosteus- ja ilmanlaatuarvoja. (Kuva 19). Tähän soveltui hyvin OLED näyttö 0.96" mikä on I2C liitännällä. Myös käyttöliittymässä täytyisi pystyä myös vaihtamaan asetuksia nappi digitaalisella liitännällä, ja potentiometri analogisella liitännällä.



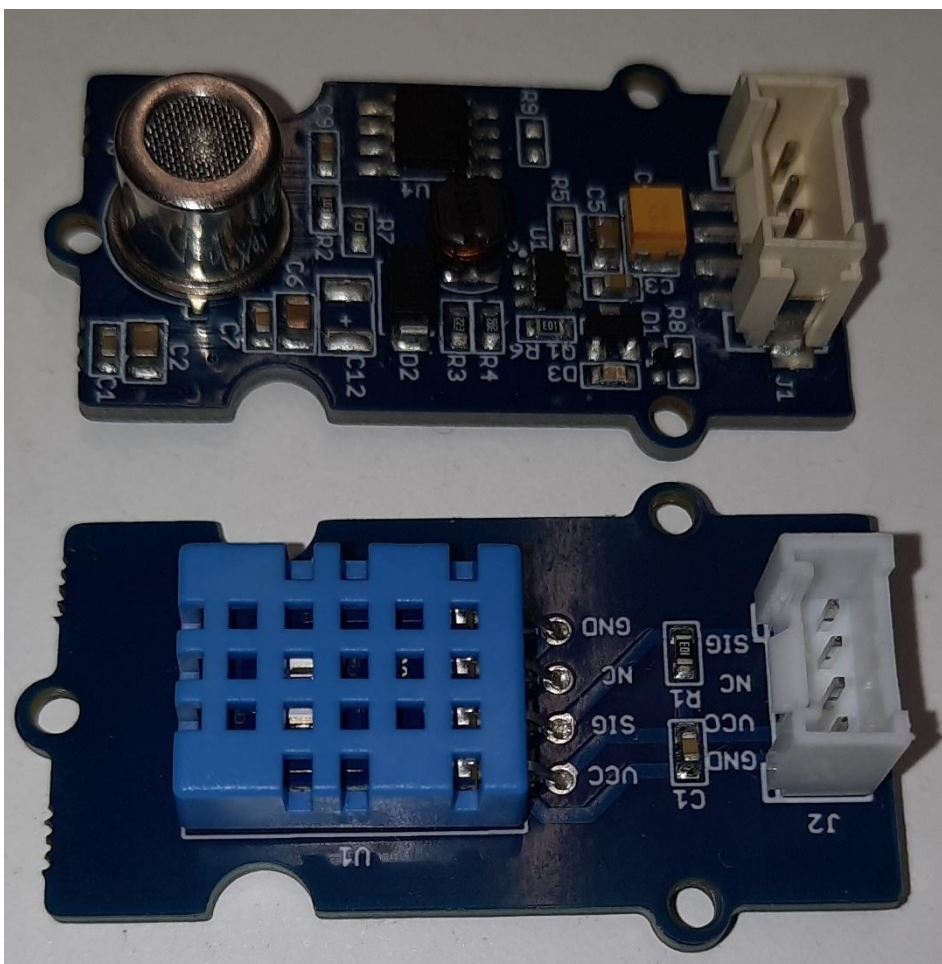
Kuva 19. Käyttöliittymän ohjaus.

3D-tulostimen ympärille rakennettavaan kammioon sisälle tarvitsisi lämpötila- ja kosteusanturin sekä ilmanlaatuanturin, mikä ei kuulunut anturipakettiin. (Kuva 20). Ylempi komponentti on ilmanlaatuanturi v1.3 analogisella liitännällä ja alempi komponentti lämpötila- ja kosteusanturi DHT 20, I2C liitännällä.



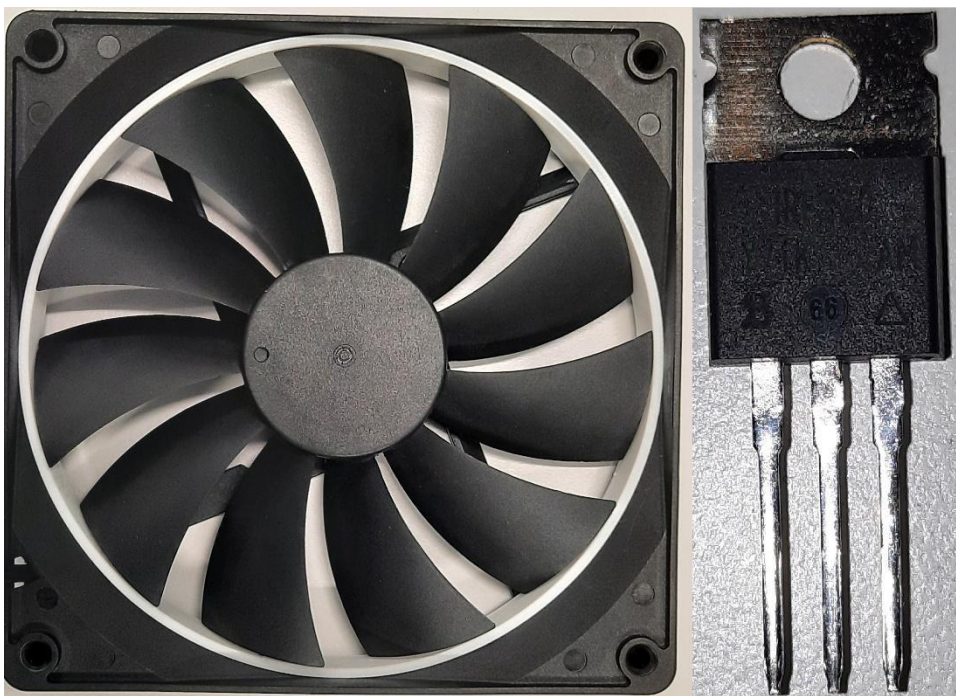
Kuva 20. Kammion sisäanturit.

Myös kammion ulkopuolelle tarvittiin samat anturit, kuin sisäpuolelle. (Kuva 21). Ainoana erona oli, että ulkopuolen lämpötila- ja kosteusanturi on DHT11 v1.2 digitaalisella liitännällä. DHT11 on DHT 20:tä hieman hitaammin lukeva sekä lämpötilan ja kosteuden tarkkuus on hieman epätarkempi. (Ong, D. 2021.) Kumpikaan näistä komponenteista ei kuulunut anturipakettiin.



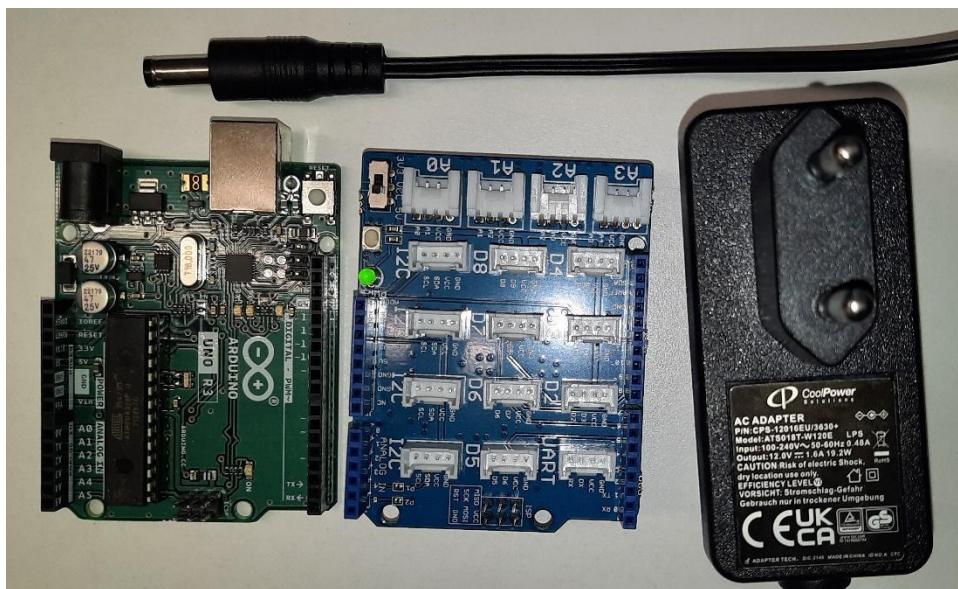
Kuva 21. Kammion ulkoanturit

Tulostuksessa kammion lämpötila nousee, joten valituksi tuli kuvan 22. 140 mm 12V Corsair PC-tuuletin, koska ne ovat edullisia ja luotettavia. Toinen syy, minkä takia valittu tuuletin tuli käyttöön, oli se, että se löytyi jo valmiiksi. Arduino aloituspakkauksesta saatiin kanavatransistori IRF-520, millä pystyisi kontrolloimaan tuulettimen pyörimisnopeutta.



*Kuva 22. Kammion lämpötilan kontrollointi.*

Arduinon anturipaketin mukana tuli (Kuva 23). Arduino Uno sekä Sensorshield, jonka pystyy liittämään Arduinon päälle ja kaikki anturit saa siististi liitettyä itse shieldiin 4-pinnisillä liitäntäjohdoilla. Myös tilaukseen laitettiin 12V, 1.6A virta-adapteri. Tilauksessa oli myös haaroitin adapteriin, mitä kuvassa ei näy.



Kuva 23. Arduino, siihen liitettävä anturikilpi ja virtalähde.

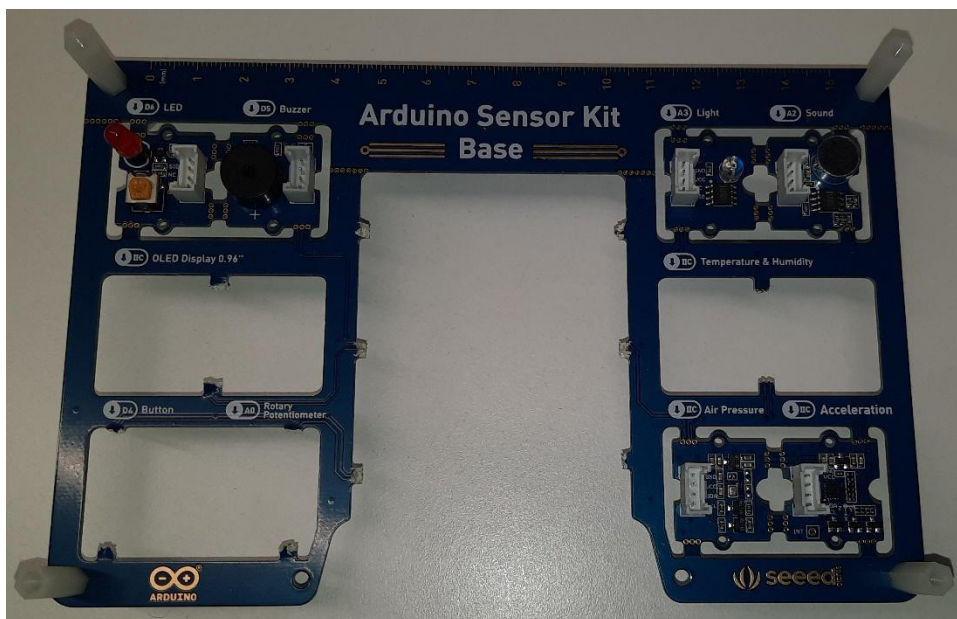


Suodattimena toimi (Kuva 24). Philips/Electrolux-pölynimureihin tarkoitettu HEPA-suodatin. Imureiden HEPA-suodattimet ovat tunnetusti erittäin hyviä ja tehokkaita.



*Kuva 24. HEPA-suodatin.*

Kuvassa 25. Sensoripaketista jäi jäljelle vielä LED-valo, pietsokaiutin millä pystyy toistamaan binäärisiä ääniä. Fotoresistori, jolla voidaan lukea valon voimakkuuksia. Mikrofoni, jolla voidaan mitata äänen värähtelyä. Ilmanpaineen mittaamiseen vaadittu anturi ja liikkeen havaitsemiseen käytettävä anturi, millä pystytään mittaamaan liikettä X-, Y- ja Z- akseleilla. (Arduino, 2021.)



Kuva 25. Sensoripakkauksesta ylimääräiseksi jääneet anturit.

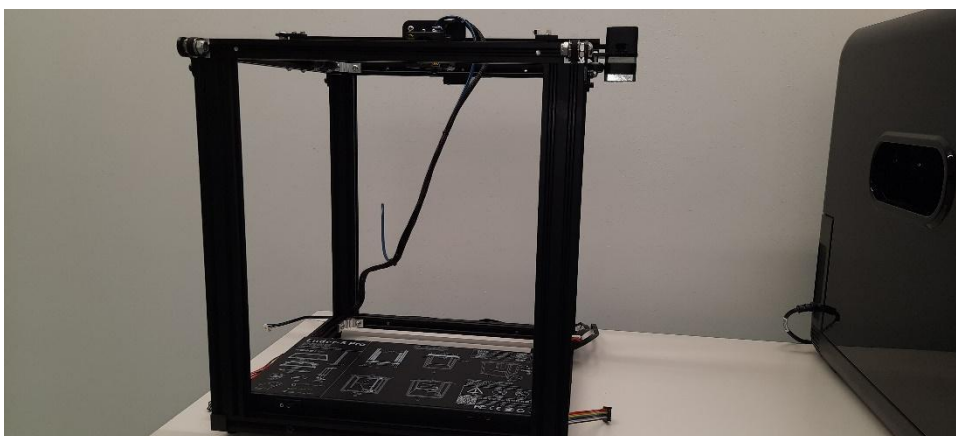
## 8 TULOSTIN, KAMMIO JA POISTOILMA

Ender-5 Pro 3D-tulostimen kokoamiseen oli laadittu hyvät ja selkeät ohjeet. Ohjeita seuraamalla ei voinut epäonnistua tässä tehtävässä. Kuvassa 26 3D-tulostimen alakehyksen päälle asennettiin 4 kappaletta alumiiniprofiilia, jotka kiinnitettiin kahdeksalla M5X25 pultilla alakehikon alapuolella olevista rei'istä.



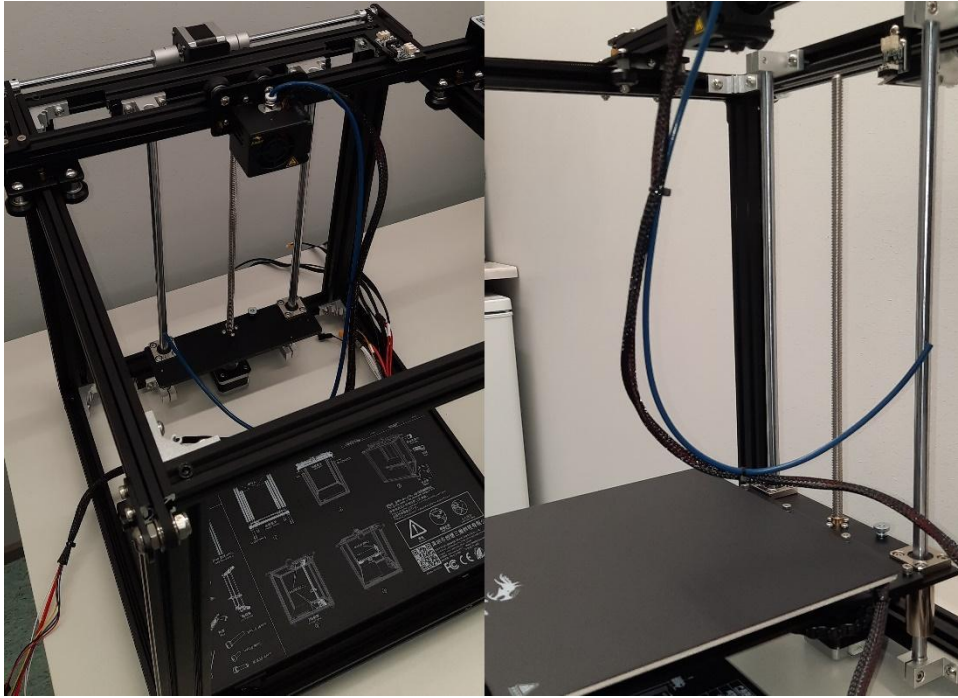
*Kuva 26. 3D-tulostimen alakehys ja alumiiniprofiilit.*

Alumiiniprofiilien päälle asetettiin 3D-tulostimen yläkehys, jossa ovat X- ja Y-askelmootorit (Kuvan 27). Yläkehys kiinnitettiin jokaisesta kulmasta kahdella M5X25 kokoisella pultilla.



*Kuva 27. Yläkehysen asentaminen, missä ovat X/Y-askelmootorit.*

Vasemmalla kuvassa 28. tulostimen Z-askelmoottori kehys, joka asennettiin ylä- ja alakehyksistä kahdella M5x30 pultilla. Z-askelmoottorikehyksessä oli pieni alusta, jossa oli valmiit reiät lämmitettävälle alustalle kiinnitykseen. Kiinnittämiseen käytettiin kuutta M4x10 koneruuvia.



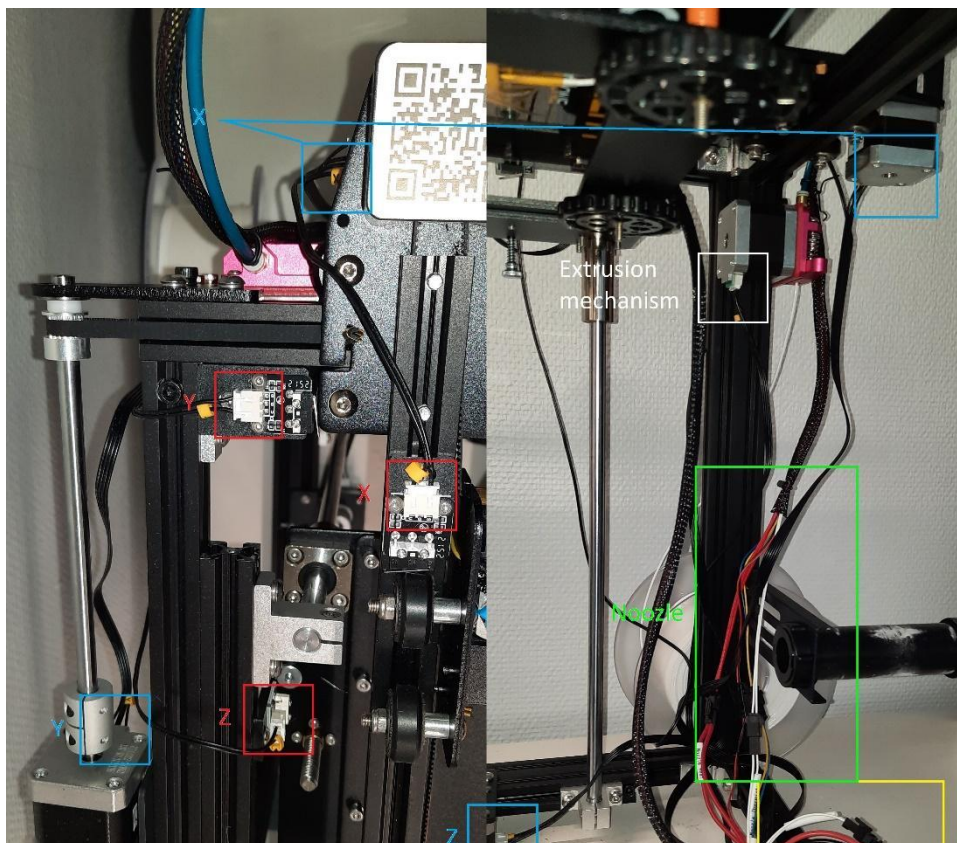
Kuva 28. Z-askelmoottori ja tulostusalusta.

Näyttö kiinnitettiin oikeaan etummaiseen alumiiniprofiiliin, sillä kiinnitys muualle olisi tarvinnut pidemmän kaapelin (Kuva 29). Kaapelin pituus antoi näytön nousta vain n. 8 cm. Langansyöttömoottori oli mahdollista asentaa joko oikean taka- tai oikeaan etualumiiniprofiiliin yläosaan. Vasemmalla puolella oleviin alumiiniprofiileihin täytyisi huomioida asennuksessa tulostuspedin laskeutuminen, mikä voisi laskiessaan repiä rikki langansyöttömoottorille menevän kaapelin. Filamenttipidikkeen asennus ei ollut välttämätön, mutta myös sille oli runsaasti eri kiinnityskohtia. Kaikkien näiden kiinnittämiseen oli varattu kaksi M4X8 pulttia ja vastakappaleena toimi M4 T-nut mikä mahdollisti kiinnityksen jokaiseen alumiiniprofiiliin.



*Kuva 29. Näyttö, langansyöttömoottori ja filamentti pidike.*

Kuvassa 30. on punaisilla merkityt X, Y ja Z ovat rajakytkimiä. Sinisellä merkityt kohdat ovat X, Y, Z askelmoottoreille menevät kaapelit. Valkoisella merkitty kohta on langansyöttömoottori, millä syötetään tulostuslankaa suuttimelle, joka sitten pursottaa sitä tulostuspedille. Vihreällä ovat suorakulmion sisälle merkityt kaapelit, jotka menevät kaapelikeräimeen, millä saadaan siististi vietyä kaapelit pursottimen tuulettimeen. Keltaisella merkityt kaapelit ovat tulostusalustaan menevät kaapelit.



Kuva 30. Kaapelien paikat.

## 8.1 Kammio

Suunnitelma oli, että kammio integroitaisiin 3D-tulostimeen kiinni, mutta 3D-tulostimen alaosassa oleva näyttö ja yläosassa X- ja Y-akselin askelmoottorit olisivat tehneet kammion tekemisestä aikaa vievän, joten kammiota ei integroitaisi kiinni tulostimeen.

Kammion leveydessä piti huomioida näyttö ja tulostimen päällä oleva X-akselin moottori (Kuva 31). Syvyydessä täytyi ottaa huomioon virtakytkin, micro-USB-liitin ja myös muistikortille oli paikka 3D-tulostimen etuosassa. Korkeudessa otettiin huomioon suuttimelle tulevat kaapelit ja samassa kaapeliryppäessä oleva lankaputki, missä lanka kulkee, kun lankamoottori syöttää lankaa pursottimeen. Pursotin liikkui X-, Y-akseleilla ja siinä oli vaarana, että jos korkeudesta tekee liian matalan, johdot voivat takertua kannessa olevista pulteista, joita suunniteltiin käytettävän tuulettimen ja käyttöliittymän asennukseen, tai että lanka ei pääse liikkumaan putkessa, koska se pääsee litistymään. Vaihtoilma tulisi alapuolelta kammioon, koska kammio lepäisi n. 5 mm korkuisien tassujen päällä. Kammion seinät ja katto tehtäisiin 3 millimetrin paksuisesta polykarbonaatista (PC). Seinät kiinnitettäisiin toisiinsa 20X20 millimetrin muovikulmallalistalla.



*Kuva 31. Kammion 3D-suunnitelma.*

Kannen kiinnitystä suunniteltiin aluksi kahdeksalla kulmamuoviosalla. Ne kiinnitettäisiin kammion sivusta 5 millimetrin pulteilla ja kansi myös olisi kiinni 5 millimetrin pulteilla kulmamuoviossa, mutta niissä olisi korvamutterit, jotta kannen avauksen ja kiinnityksen voisi tehdä käsin ilman työkaluja. Ideaa kuitenkin muutettiin, koska kannen avaukseen ja kiinnitykseen täytyisi olla helpompikin tapa. (Kuva 32). Kulmamuoviosilla kiinnitys unohdettiin ja suunniteltiin kannen sivulle kaksi saranaa, joissa kammion sivuun kiinnitettävän osan saranatappin reikä on ovaalin muotoinen. Tämä auttaa siinä, että kansi voi laskeutua kiinni-asennossa vielä muutaman millin alemmas, eikä jäisi tiivistämättömiä kohtia kannen ja alustan väliin. Kannen päälle saranan vastakappale kiinnitettäisiin myös 5 millimetrin pulteilla. Saranareikä siinä olisi aivan pyöreä. Saranatappina toimisi 5 millimetrin pultti.



*Kuva 32. Kulmamuoviosien uudelleen suunnittelu.*



Ovensaranaan suunniteltiin, että siihen laitetaan tiiviste väliin. (Kuva 33), joten kammioon kiinnitettävän saranakappaleen alle tulisi 6 millimetrin korokepala polykarbonaatista. Saranan vastakappaleeseen kiinnitettäisiin 3 millimetrin polykarbonaattiovi. Oven ja kammion väliin tulisi tiivistenauha, jonka tiivistysväli olisi 2,5–3,5 millimetriä. Saratappina toimisi 5 millimetrin pultti. Saranareikä ei ole läpi asti, vaan viimeisessä saranareiässä on 2 millimetrin pohja.



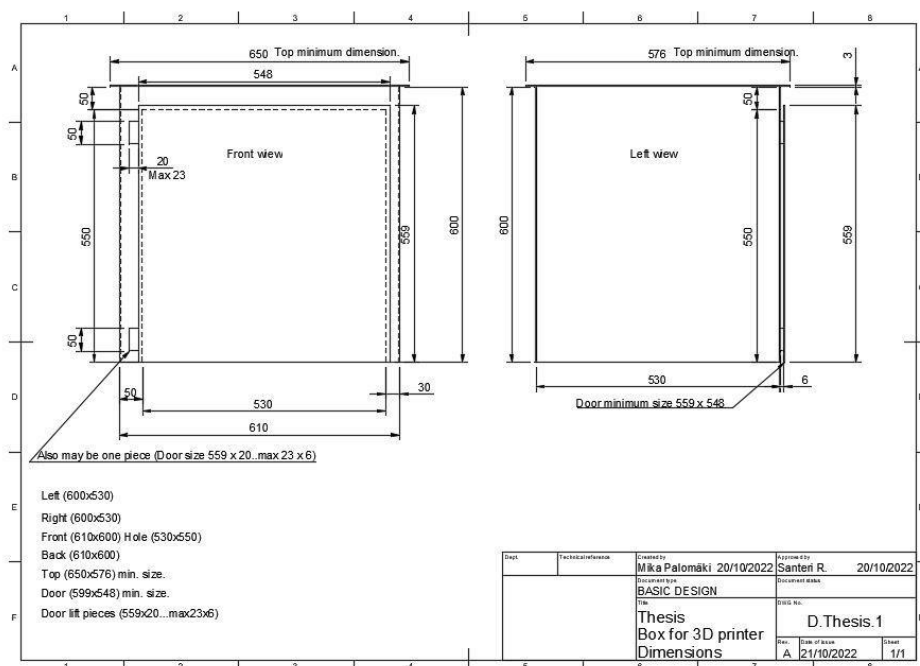
*Kuva 33. Oven saranoiden suunnittelu.*

Ovi suunniteltiin pysymään kiinni kolmella magneetilla. (Kuva 34). Magneeteille tulostettaisiin omat osat, jotka kiinnitettäisiin kammion sisäpintaan ja tulostusosiin tehtäisiin magneeteille oma paikka, johon ne liimattaisiin kiinni. Vastakappaleena toimisi millin vahvuinen metallipala, joka liimattaisiin kiinni oven takapuolelle magneettien kohdille. Kahvaksi suunniteltiin pyöreänuppinen kahva, joka kiinnitetään kahdella 5 mm pultilla.



*Kuva 34. Oven lukituksen ja kahvan suunnittelu.*

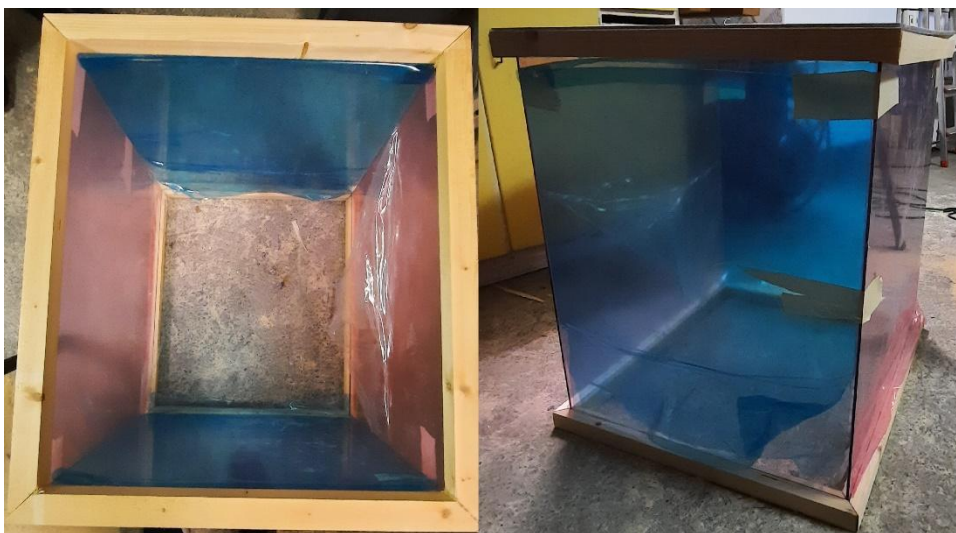
Kammion polykarbonaattilevyjen sauhukseen tehtiin (Kuva 35) CAD-piirustukset, koska levyihin hyödynnettäisiin ensisijaisesti ylijäämälevyjä.



Kuva 35. CAD-piirustukset.

## 8.2 Kammion toteutus

PC-levyä ei tilattu vaan sitä löytyi omasta ylijäämistöstä. Levyä löytyi 3 millimetrin ja 4 millimetrin paksuista. (Kuva 36). Etu- ja takaosat (vaaleanpunainen) tehtiin 3 millimetrin paksuudesta, sivut 4 millimetrin (sininen) paksuudesta. Myös kanteen pystyi hyötykäyttämään 4 millimetrin paksuista levyä. Levyt eivät olisi pysyneet kasassa pelkillä 20X20 mm muovikulmalistoilla. Levyille piti suunnitella kehikko, joka olisi tukeva. Ylijäämistöstä löytyi myös 30X30 mm paksuista listaa mistä saisi tehtyä tukevan ylä- ja alakehikon. Listoihin sahattiin 15 mm ura, mikä alkoi 10 mm listan reunasta ja ura olisi hieman leveämpi, kuin itse PC-levyn paksuus. Tämän jälkeen räätäilyönä katkottiin jiirisahalla päät jiiriin juuri oikean pituisiksi. Yhteen laitettuna jiirit muodostivat siistin 90 asteen kulman. PC-levyt teipattiin kulmista kiinni toisiinsa, mikä helpotti listojen uran sovittamista PC-levyyn. Oviaukko tehtiin yhteen 3 mm (vaaleanpunainen) levyyn. Levyyn mitattiin ja merkattiin ovenreikä (530X535) mm, alareunaa ei sahattaisi kokonaan pois niin, kuin CAD-piirustuksissa on, vaan siihen jäisi 15 mm soiro millä saisi kiinnitettyä PC-levy kehikkoihin. Oviareiän merkattuun kohdan kulmiin porattiin reiät, mistä mahtui kuviosahan terä. Sahaus piti tehdä hyvin hitaasti, koska reunat olivat ohuet, varsinkin 15 mm kiinnitysoiro. Listat kiinnitettiin toisiinsa päistä ensiksi poraamalla alkureikä, johon ruuvattiin ruuvi. Alkureiällä välttyttiin puun halkeamiselta.



Kuva 36. Tukevan ylä- ja alakehikon teko.

Ala- ja yläkehikko maalattiin valkoisella maalilla kahteen kertaan. Maalauksen kuivuttua levyt laitettiin ensiksi alakehikkoon 15 mm uraan yksitellen, minkä jälkeen yläkehikko laskettiin levyjen päälle ja haettiin urakohta, minne levyt pienellä naputuksella sopivat. (Kuva 37). Levyt kiinnitettiin kehikoihin mittaamalla sisäpuolelta uraan uponneesta 15 mm puoliväli ja poraamalla 2 mm kokoinen reikä, mikä ylettyi vain PC-levyn läpi. Reikään ruuvattiin reiän paksuinen ruuvi, jotta PC-levy ei halkeaisi. Kulmamuovilistat leikattiin oikean kokoisiksi, minkä jälkeen niihin levitettiin liima ja asennettiin kulmaan. Teipillä sai hyvin kiristettyä listan tiukasti kulmaan, jotta liimaus onnistuisi kunnolla. Kannen tiivisteenä käytettiin ikkunoihin tarkoitettua tiivistenauhaa, mikä ajoi asiansa vallan mainiosti.



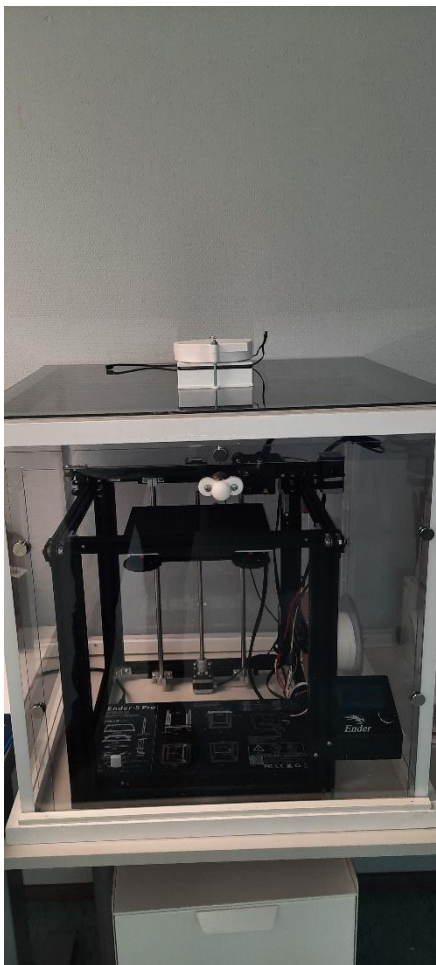
*Kuva 37. Kehikoiden ja muovisenkulmalista kiinnitys.*

Ylijäämistössä ei ollut enää tarpeeksi isoa 3–4 mm palaa PC-levyä, mutta 2 mm paksuista levyä löytyi. Se olisi kuitenkin vähän liian taipuisaa verrattuna 3–4 mm PC-levyyn. Ovesta unohdettaisiin saranat ja käytettäisiin sen sijaan magneetteja. (Kuva 38). Ovesta tehtiin 560X560, magneeteille haettiin mahdollisimman hyvät kohdat, jossa niiden sijainti tiivistäisi oven silmämääräisesti mahdollisimman hyvin. Kohtien löydyttyä liimattiin kammion sisäpuolelle ja oven etupuolelle kaksipuoliteipillä magneetit samoihin kohtiin toisiinsa nähden. Oven alapuolelle naulattiin lista, joka tiivistäisi alaosan ovesta.



*Kuva 38. Kammion ovi ilman saranoita.*

Kammioista tuli kaikin puolin siisti. (Kuva 39). Isommat tulostukset on mahdollista ottaa yläkautta nopeasti pois tulostuksen jälkeen, nostamalla kansi vain pois. Langan voi sijoittaa joko sisä- tai ulkopuolelle kammiota. Ulkopuolelta lanka tuodaan kammion alapuolelta. Kuvassa ei ole vielä asennettu tassuja kammioon. Myös kammion nostaminen pois kokonaan 3D-tulostimen ympäriltä on mahdollista ilman mitään ylimääräisiä toimenpiteitä.



*Kuva 39. Valmis kammio 3D-tulostimen ympärillä.*

### 8.3 Ensimmäinen suunnitelma ja 3D-tulostus.

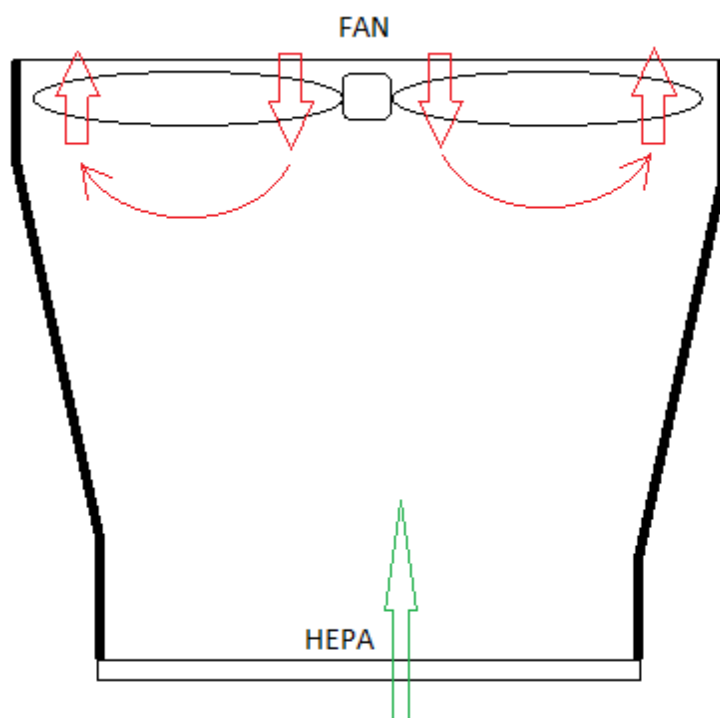
140 mm PC tuuletin kiinnitetään neljällä pultilla kappaleen ylhäällä oleviin reikiin. HEPA-suodatin asetetaan kappaleiden väliin. Kuvan 40 ylemmässä kappaleessa on 4 ulokekohtaa, joilla saadaan alimmaisen kappaleen 4 lukituskynttä taipumaan, kun kappaleita painetaan yhteen. Tietyissä kohtaa kynnet napsahtavat takaisin alkuperäiseen asentoon ja lukittautuvat ulokkeiden taakse. Irrottaminen tapahtuisi taivuttamalla lukituskyntsiä ja vetämällä ylempi osa irti. Tarkoituksena oli, että suodatin saataisiin nopeasti ja ilman työkaluja vaihdettua uuteen. Suodattimen vaihtokokeiluissa lukituskyntset tahtoivat menettää jäykkyytensä tai katkesivat lukituksen avauksessa.



*Kuva 40. Ensimmäinen 3D-suunnitelma tuulettimelle ja suodattimelle.*



Tarkoituksena oli, että lämpötilaa pystyisi kontrolloimaan tuulettimella. Ilmeni, että tuuletin ei pystynyt imemään koko sen teholla ilmaa suodattimen läpi, vaan vain hyvin pieni osa ilmasta meni suodattimen läpi (Kuva 41). Suurin osa ilmasta kiersi tuulettimen siivekkeiden välissä siten, että keskustan kautta tuuletin imi ja sivuilta puhalsi. Tämä selvisi, kun asetti A4-paperin tuulettimen päälle, paperi imeytyi tuulettimen keskustaan ja sivut hieman väpättivät, kun tuuletin pyrki saamaan ilmaa pois sivujen kautta ja kun HEPA-suodattimen otti pois, tuuletin puhalsi paperin kokonaisuudessaan pois.



Kuva 41. Kokoonpanossa ilmenevä haaste.

Ulkoseinän ja siivekkeiden välissä oli noin 2 millimetriä tyhjää tilaa. Välikohtaan tulostettiin kappale, millä siivekkeet ja ulkoseinän saisi mahdollisimman lähelle toisiaan ja siten voisi pienentää ilman kiertoa siivekkeiden välissä. (Kuva 42). Myös tekemällä keskustaan esteen saisi estettyä, että tuuletin ei imisi ilma takaisin vaan koko imu tapahtuisi HEPA-suodattimen kautta. Kokeilulla ei imuteho parantunut millään lailla.



*Kuva 42. Tuulettimen muokkaus kappaleet ovat valkoisia.*

Idea tähän kokeiluun syntyi pienestä kämmenrikkaimurista. Muuttaminen PC-tuuletin imuriksi voisi auttaa imutehossa. PC-tuuletinta piti hieman vain muokata. (Kuva 43). Kaikki tuulettimen siivekkeet piti katkoa ja keskiöstä saada hiottua mahdollisemman pyöreäksi, että pyöräminen pysyisi vakaana. Sivuleikkureilla siivekkeet menivät helposti poikki. Hiominen tapahtui kuumaliimalla ruuvi mahdollisimman lähelle tuulettimen keskustaa. Akkuporakone kiinnitettiin ruuviin millä saatiin keskusta pyörimään tasaisesti ja nauhahiomakoneen hiomapaperi kiedottiin ympäri millä saatiin ylimääräiset siivekkeiden jämät hiottua pois siististi ja tasaisesti.



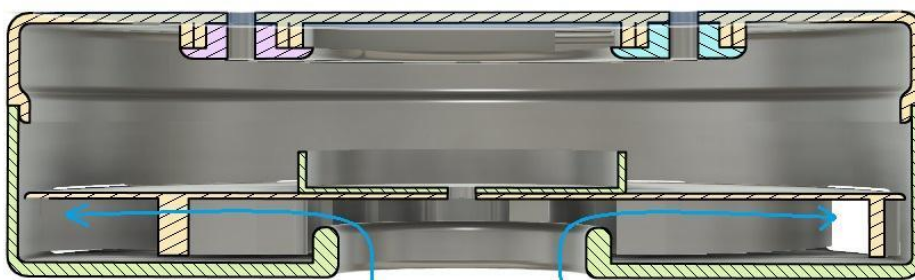
*Kuva 43. Kuvassa vasemmalla alkuperäinen tuuletin ja kuvassa oikealla muokattu tuuletin.*

Kuvassa 44. Ylimmäiseen kappaleeseen on tehty paikka, johon modifioitu tuulettimen alaosa saa asetettua. Modifioitun tuulettimen päälle tulee kaksi painoalustaa, joilla saadaan tuuletin pysymään paikoillaan. Kiristämiseen tarvitaan neljä kappaletta 5 mm koneruuvia. Hiottu PC tuulettimen keskusta liitetään kuvassa näkyvään sille räätälöityyn pidikkeeseen. Reikä keskustassa on siivekkeiden ja pidikkeen kohdistamista varten. Siivekkeet ja pidike tulostettiin erikseen, kohdistamalla ne oli helppoa liimata yhteen tarkasti. Kohdistamiseen käytettiin 5 mm koneruuvia.



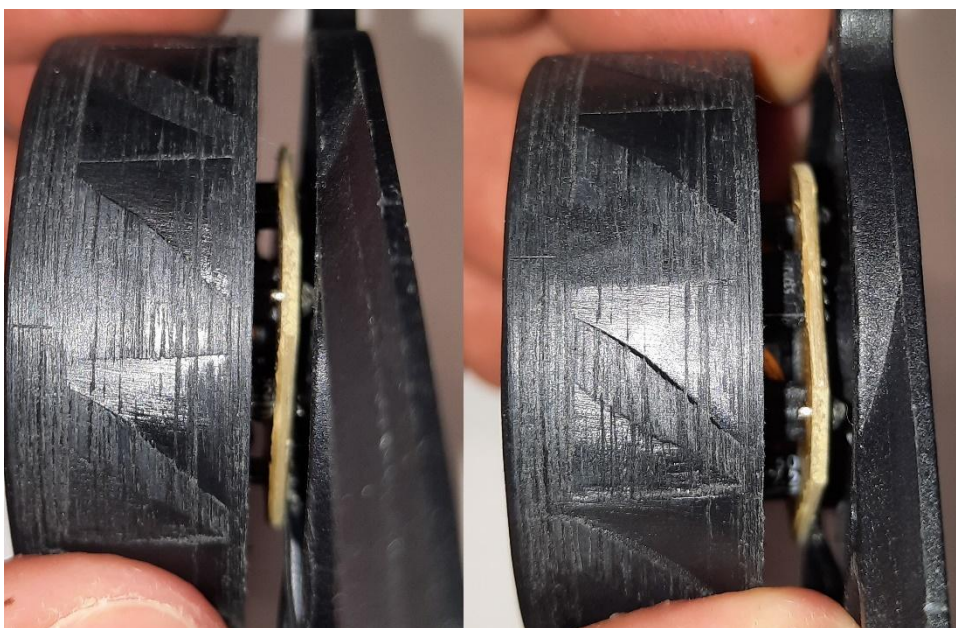
*Kuva 44. Räjätyskuva turbotuuletin VI.0.*

Alimmainen osa on ilmanvirtauksen ohjain. Ilma imeytyy keskustasta sisään ja sivuilta se puhalletaan ulos. Keskustan pyörimisnopeus on nopeampaa kuin sivuilla. Tällöin alipaine muodostuu keskustaan (Kuva 45). Imuteho suodattimen läpi tällä kokoonpanolla oli hyvinkin toimiva. Teho oli sen verran hyvä, että kun laittoi HEPA-suodattimen alipainepuolelle, sai jopa A4 paperin pysymään kiinni HEPA-suodattimessa PC-tuulettimen ollessa täydellä teholla. Aiemmissä yrityksissä ei pystynyt edes huomaamaan kynttilän savullakaan minkäänlaista ilmavirtausta HEPA-suodattimen läpi.



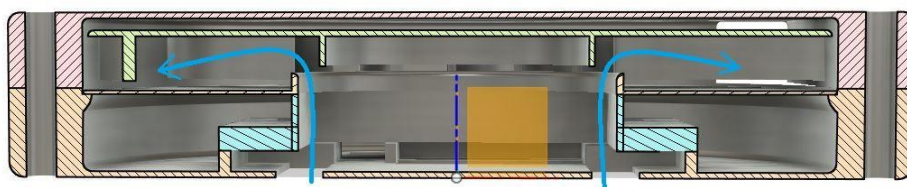
Kuva 45. Leikkauskuva turbotuuletin V1.0

Ratkaisu oli toimiva, mutta epäluotettava. PC Tuuletin toimi harjattomasti, missä magneetikentät vaihtavat magneettisuus suuntaansa ja saavat tuulettimen pyörimään. (Kuva 46). Tuulettimen keskustaa sai vedettyä, jolloin magneetit eivät saaneet kontaktia toisiinsa. Ensimmäisistä uusista siivekkeistä tulostuksesta tuli pomminkestävä, eli aineenvahvuus oli lähempänä 3 millimetriä, mikä antoi painoa siivekkeille vetää magneetit erilleen toisistaan. Pientämällä vahvuudet mahdollisimman pieniksi oli huomattavissa siltikin pientä katkoa pyörimisessä.



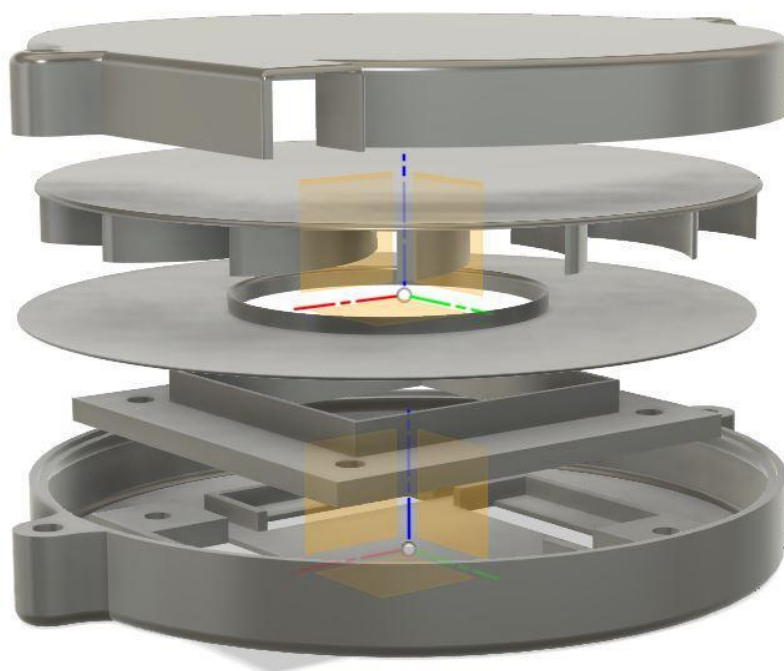
Kuva 46. Vasemmassa kuvassa tuuletin normaalissa asennossa. Oikeassa tuuletinta vedetään kevyesti, silloin magneetit ei saa kontaktia.

Ratkaisu painolle oli kääntää tuuletin toisinpäin. (Kuva 47). Tässä kokoonpanossa ei painolla ollut merkitystä magneettien kontaktiin, koska siivekkeet lepäsivät tuulettimen päällä.



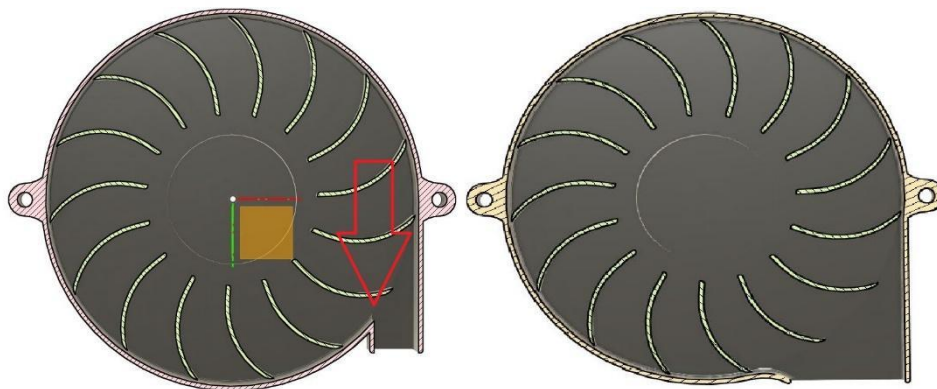
Kuva 47. Leikkauskuva turbotuuletin V 1.1

Kuvassa 48. Näkyvään alimpaan osaan asennettiin modifioitu tuuletin. Osassa on myös suorakulman muotoisia reikiä, joista tuloilma pääsee sisälle. Modifioitu tuuletin kiinnitettiin alimpaaseen osaan sen päälle laskettavalla osalla, mikä toimii myös tuloilmaohjaimena, kiinnitykseen käytetään 4 kappaletta M5 koneruuvia. Keskimmäisellä osalla ohjataan ilma siivekkeisiin. Siivekkeet pyörivät vastapäivään, mikä aiheuttaa alipaineen keskustaan ja poistoilma puhalletaan ylimmäisen osan neliskanttisesta putkesta.



*Kuva 48. Turbotuuletin VI.1 räjäytyskuva.*

Kammio vaimensi kohtalaisesti 3D-tulostimesta tulevaa ääntä, mutta tuuletin piti pyöriessään ikävän turbomaisen vinkuvan äänen, mistä turbotuuletin sai nimensä. (Kuva 49). Ääni johtui ilmanvirtauksen osuessa terävään kulmaan puhallinkotelon sisässä. Vinkuna vaimeni, kun kulmaa leikkasi hieman pois ja pyörästi sitä teipillä, tämä auttoi puhallinkotelon uudelleensuunnittelua.



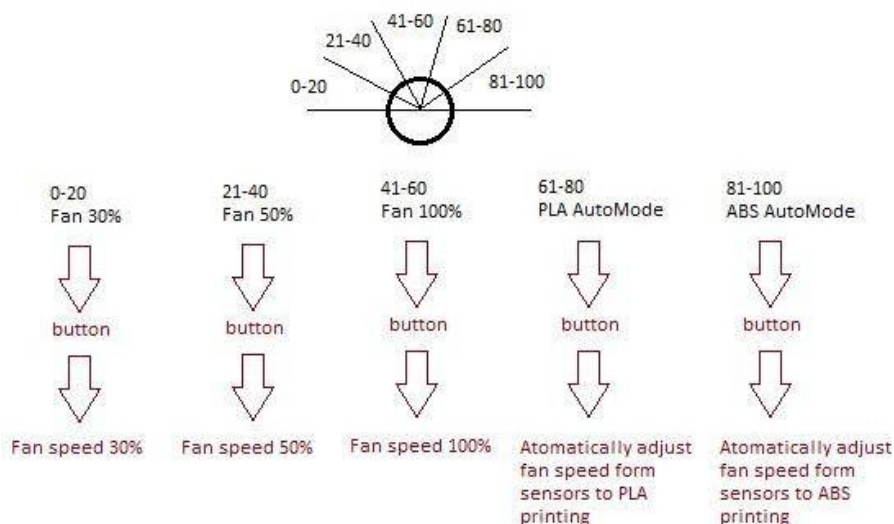
*Kuva 49. Puhallinkotelon äänenvaimennus.*



## 9 KÄYTTÖLIITTYMÄ JA OHJELMOINTI

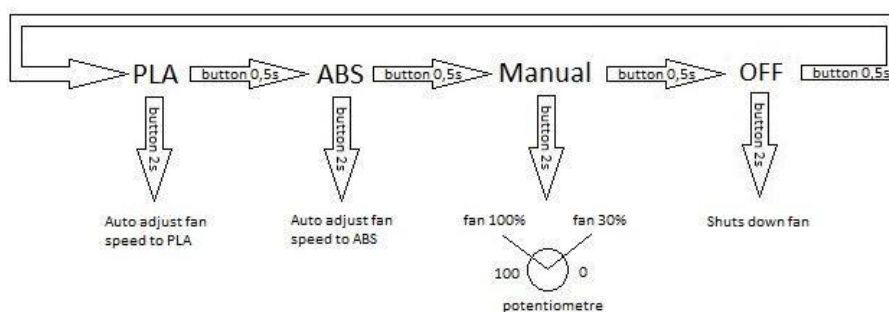
Projektissa ei ollut kuin nappi ja potentiometri, joilla pystyi ohjaamaan käyttöliittymää. Molemmista tehtiin oma suunnitelma ja ne esitettiin asiakkaalle. Asiakkaalta tiedusteltiin kummalla olisi helpompaa ja mielekkäämpää käyttää käyttöliittymää, vai tulisiko sitä ohjata jollain ihan toisenlaisella ohjaimella.

Potentiometrin alueet jaoteltaisiin viiteen eri alueeseen. (Kuva 50). Alueelta siirryttäessä seuraavalle alueelle vaihtuisi vain teksti, mutta mitään ei vielä hyväksyttäisi. 0–20, 21–40, 41–60 olisi manuaalisen ohjauksen alueet tuulettimelle, 61–80 Tuuletin toimisi automaattisesti analysoiden sensoreita PLA:n tulostamisessa ja 81–100 Olisi ABS:n tulostamiseen tarkoitettu alue mikä tietyn ajan välein automaattisesti tarkistaisi lämpötilan ja ilmanlaadun ja säätäisi tuulettimen nopeutta. Vasta nappia painamalla hyväksyttäisiin sillä alueella olevat toiminnot.



Kuva 50. Potentiometrin käyttöliittymä.

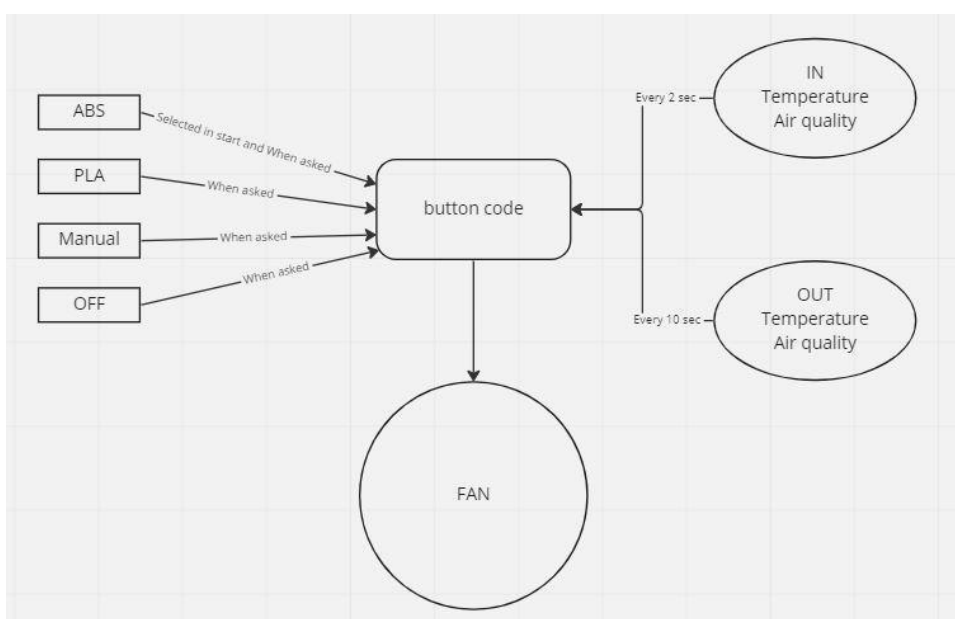
Kuvassa 51. kun nappia painettaisiin alle kaksi sekuntia, vaihtuisi vain teksti. PLA ja ABS toiminnot olisivat automaatti toimintoja ja toimisivat samalla periaatteella, kuin potentiometri käyttöliittymässä, manuaalitoiminnolla pystyisi ohjaamaan tuuletinta potentiometrillä manuaalisesti 30 % - 100 % nopeudella ja OFF-toiminto sammuttaisi tuulettimen. Toimintojen hyväksyntä tapahtuisi painamalla nappia pohjassa yli kaksi sekuntia. Napilla käyttöliittymän ohjaaminen oli helpompaa ja nopeampaa, kuin potentiometrillä, joten käyttöliittymää alettiin toteuttaa napilla.



Kuva 51. Napin käyttöliittymä.

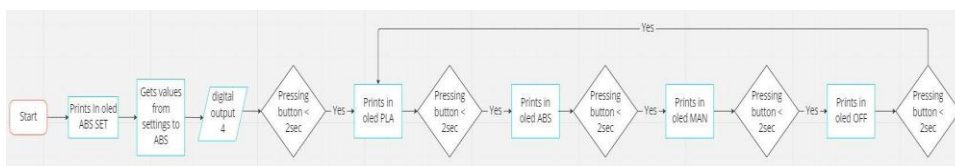
## 9.1 Ohjelman suunnittelu

Suunnittelu alkoi isosta kokonaisuudesta ja sitä pilkkomalla pienempiin osiin. (Kuva 52). Kaiken ohjaus tapahtuu nappikoodilla. PLA, Manuaali ja OFF -toiminnot tulevat vain, jos niitä kutsutaan. ABS-tila otetaan käyttöön aina, kun järjestelmä käynnistyy. Sen jälkeen tilaa täytyy kutsua manuaalisesti. Anturit, jotka ovat sisällä kammiossa päivittävät tiedot joka kahden sekunnin välein ja kammion ulkopuolella olevat anturit päivittävät tiedot 10 sekunnin välein. Koska anturien lukeminen hidastaa koodia, laitettiin ulko- ja sisäensensoreille eri päivittymisajat.



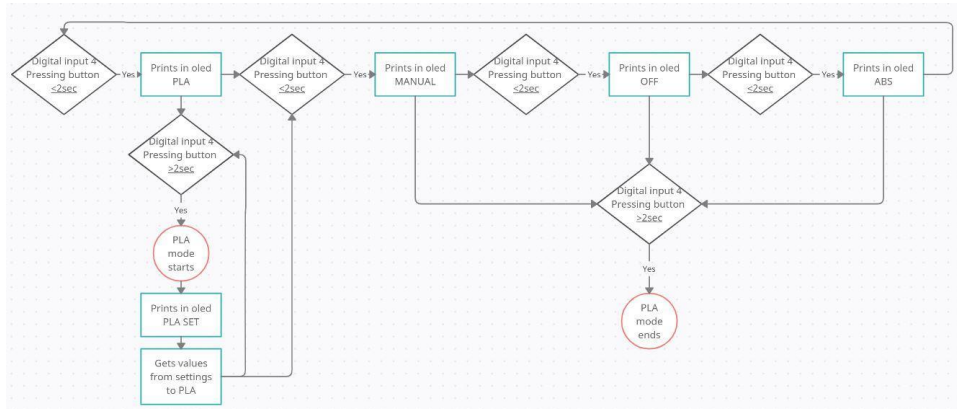
Kuva 52. Vuokaavio ohjelman kokonaisuudesta

Käynnistyksessä kuvassa 53. tuulettimelle tuodaan suoraan ABS-arvot. Pinniin 4 on liitetty painonappi, mitä painamalla pääsee liikkumaan käyttöliittymässä. Alle kahden sekunnin painonapin painamisella vaihtuu vain teksti PLA, ABS, MAN ja OFF. OFF-tilan jälkeen painettaessa painonappia alle kaksi sekuntia tekstin vaihtuminen alkaa alusta.



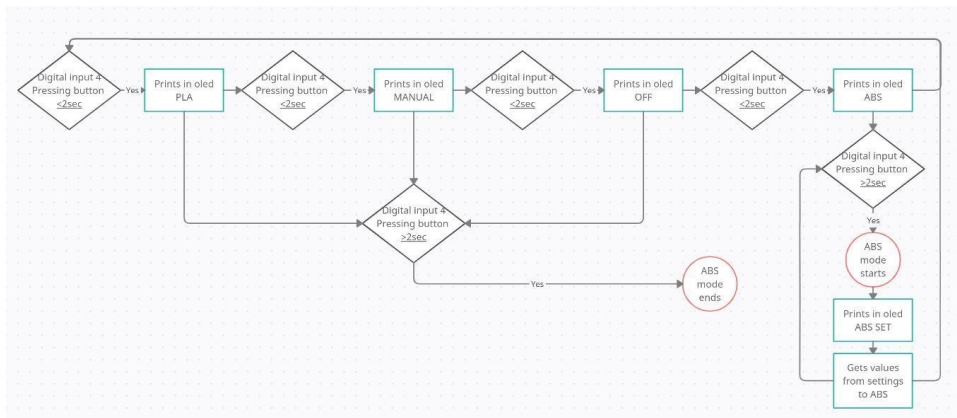
Kuva 53. Napin vuokaavio

Kuvassa 54. Kun käyttöliittymässä on vaihdettu hyväksyttäväksi tekstiksi PLA. Painamalla painonappia 2 sekuntia pohjassa, kutsutaan PLA-tila. PLA tilassa hyväksyttävän tekstin selaaminen on mahdollista ilman, että se häiritsee kutsuttua tilaa. Vasta kun hyväksyttävä tila on vaihdettu joksikin muuksi, kuin PLA, PLA tila päättyy.



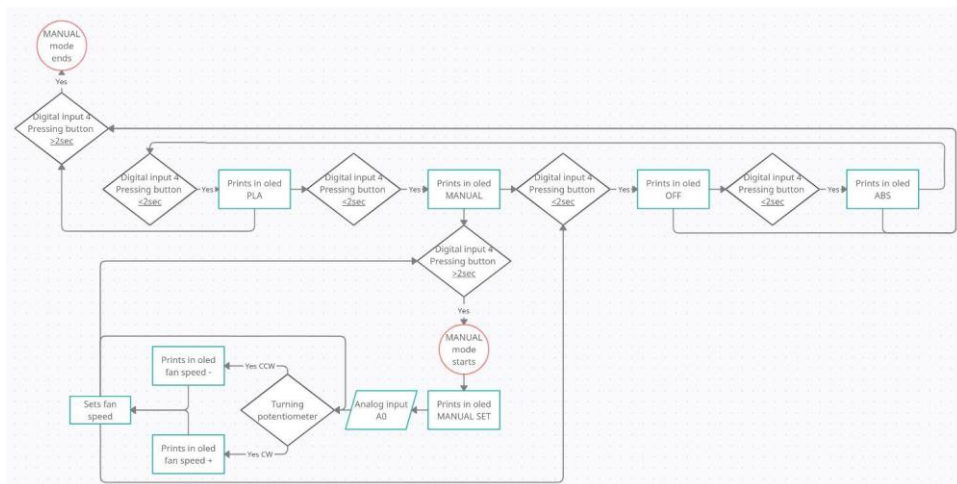
Kuva 54. PLA tilan vuokaavio

Kuvassa 55. ABS tila on käyttöliittymältään samanlainen, kuin PLA tila. Tilan lopettaminen tapahtuu valitsemalla PLA, MANUAL tai OFF tila.



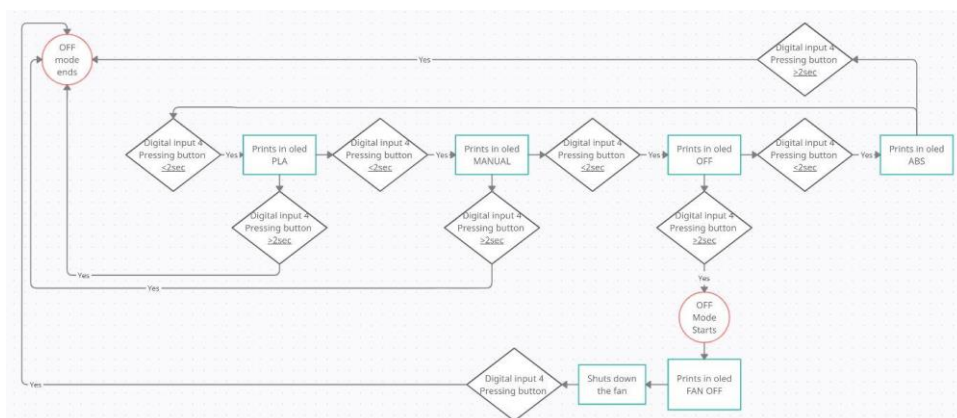
Kuva 55. ABS tilan vuokaavio

Manuaalitilan kutsunta tapahtuu painamalla painonappia pohjassa kaksi sekuntia hyväksyttävän tekstin ollessa MANUAL. (Kuva 56). A0 pinniin liitettyä potentiometriä kääntämällä myötäpäivään tuulettimen nopeus kasvaa ja vastapäivään tuulettimen nopeus hidastuu. Manuaalitilan päättämiseksi painetaan painonappia pohjassa kaksi sekuntia hyväksyttävän tekstin ollessa jokin muu, kuin MANUAL.



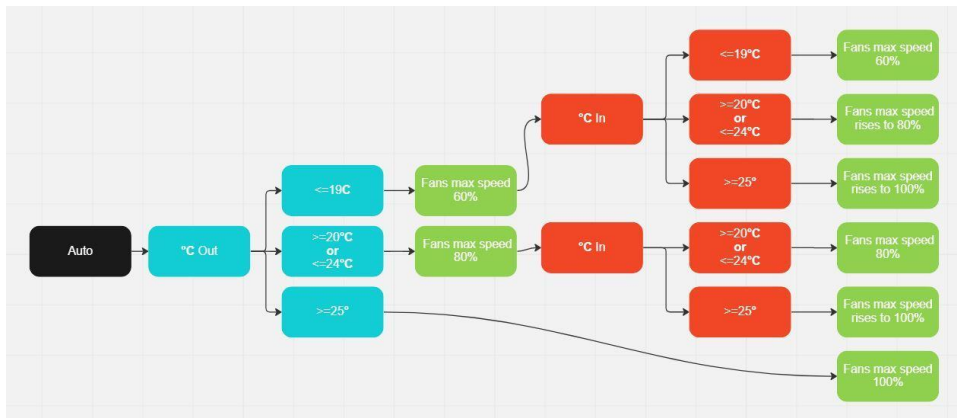
Kuva 56. Manuaali tilan vuokaavio

Tuulettimen ja OLED-näytön sammuttaminen ja tyhjentäminen suoritetaan kutsumalla OFF tilaa. (Kuva 57). OLED-näyttö tyhjentyy ja tuulettimelle menevät PWM arvot tippuvat noltaan. Off tilassa painonappia painettaessa pitkään tai nopeasti kutsuu automaattisesti se ABS tilan.



Kuva 57. OFF-tilan vuokaavio

PLA ja ABS tila. Sinisissä kammion ulkolämpötilan, punaisissa kammion sisälämpötila ja vihreät ovat tuulettimen nopeudet. (Kuva 58). ABS tilassa erona PLA tilaan on, että OLED näytölle tulee lämpötilan ja kosteuden lisäksi ilmanlaatu.

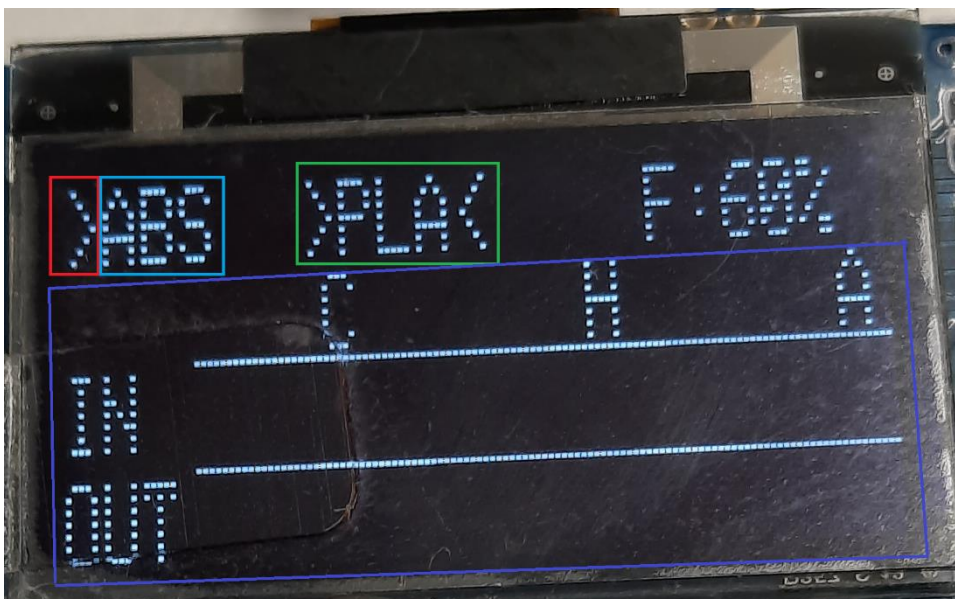


Kuva 58. Automaatti tilassa lämpötilan kontrolloiminen.

## 9.2 Nappikoodin ohjelmointi

Nappikoodissa yksi haastavimmista osuuksista oli saada tapahtumaan eri asia nappia painamalla lyhyesti tai pitkään. Nappikoodi kirjoitettiin kolme kertaa uudestaan, koska jossain vaiheessa koodin kirjoitusta tuli tilanne missä ei päässyt etenemään. Aloittaminen uudestaan puhtaalta pöydältä ja tekemällä asioita hieman eri tavalla auttoi nopeastikin tilanteissa, joissa oli umpikuja tullut vastaan. Oli myös tärkeätä, että ei kerralla kirjoittanut tai poistanut liikaa koodia ilman, että kokeili sen toimivuutta. Varmin tapa oli kirjoittaa tai poistaa yksittäisiä rivejä ja kokeilla sen jälkeen toimivuus.

Kuvassa 59 vasemmassa yläkulmassa punaisella merkattu kohta vaihtuu joko < tai > riippuen siitä, kuinka kauan nappia painetaan. Merkin ollessa < ja napista irrotettaessa sinisellä oleva nameValue vaihtaa arvoa 0, 1, 2 ja 3, missä 0 = ABS, 1 = PLA, 2 = MAN ja 3 = OFF. Jos arvon vaihto tapahtuu 3:sen aikana, alkaa se uudestaan 0:sta. Jos merkki vaihtuu > ja sinisellä nameValue kohdassa on arvo 2 = MAN ja napista irrottaa, vaihtuu vihreällä accepted\_value arvo samaksi, kuin nameValue. accepted\_value arvot ovat 0 =>ABS<, 1 = >PLA<, 2 => MANUAL< ja 3 = >OFF<.



```

SHORT_PRESS_BUTTON_PRESSING <
LONG_PRESS_BUTTON_PRESSING >
nameValue
accepted_value
PRINT

```

Kuva 59. Koodin kulku nappikoodissa.

Liitteessä 1 kohdassa 13. Koodin alussa luetaan napintila `buttonState`, mitkä voivat olla joko painettuna = 1 tai ei painettu = 0. Jos nappia ei ole painettuna ja `accepted_value` on erisuuri kuin 3 = OFF. Kutsutaan `TEMP_HUMI_OUT` ja `TEMP_HUMI_IN` mistä luetaan sisä- ja ulkosensoreiden lämpö- ja kosteusarvot, eli anturit luetaan kaikissa muissa arvoissa paitsi OFF arvossa. `lastButtonState` on määritelty arvoksi 0. Jos nappia painetaan (`buttonState`) muuttuu arvosta 0 arvoon 1, jolloin painoajastin `pressTime` arvo lähtee kasvamaan 0–∞. Napista irrotettaessa luetaan `pressTime` -arvo. `Accepted_value` ollessa 3 = OFF hypätään suoraan else komenttoon. Napin ollessa painettuna `pressTime` arvon ollessa 0–4. Kutsutaan SPBP `Short_Press_Button_Pressing` missä Oled-näytölle kirjoitetaan < merkki. Tällä näytetään käyttäjälle `pressTime` tila. Napin ollessa painettuna `pressTime` arvon ollessa 5–∞. Kutsutaan LPBR `Long_Press_Button_Release`, missä Oled-näytölle kirjoitetaan > merkki. Myös tämä on käyttäjälle tarkoitettu näkymä `pressTime` tilasta. Else if komennoissa kutsutaan `accepted_value` arvoon määritelty kutsu. Else komento toteutuu vain `accepted_value` arvon ollessa 3.

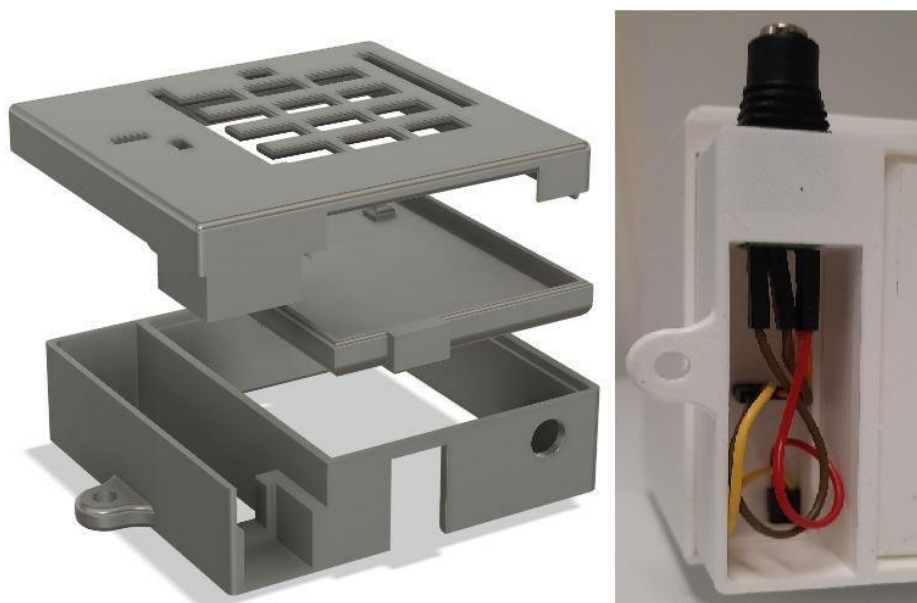


Liitteessä 1 kohdassa 16. Valitaan valkoinen väri tekstille. Seuraavalla kerralla, kun SPBR kutsutaan poistetaan mustalla teksti, jotta ei tule päällekkäin kirjoitusta. Valitaan tekstin sijainti (X, Y) akseleilla. Missä (0, 0) on OLED-näytön vasen yläkulma. Asetetaan sen hetkinen arvo, joka on nameValue. Tuodaan komennot OLED-näytölle. PressTime -arvo määritellään takaisin 0. nameValue arvo kasvaa yhdellä aina, kun SPBR kutsutaan. Koska nameValue kasvaa joka SPBR kutsusta yhdellä voisi se kasvaa äärettömyyksiin asti. nameValue ollessa arvossa 4 tuodaan se takaisin arvoon 0, jolloin se saadaan pysymään arvoissa 0, 1, 2, 3.

Liitteessä 1 kohdassa 17 LPBR kutsun alussa accepted\_value arvo määritellään samaksi arvoksi, kuin nameValue, mutta accepted\_value kasvoi aina yhdellä arvolla isommaksi kuin nameValue, jotenka vähentämällä accepted\_value arvosta yhden, saatiin accepted\_value arvo pysymään samana, kuin nameValue. Jos accepted\_value arvo menee -1 muuttuu se arvoksi 3. Tiedot tuodaan OLED-näytölle ja presstime nollataan.

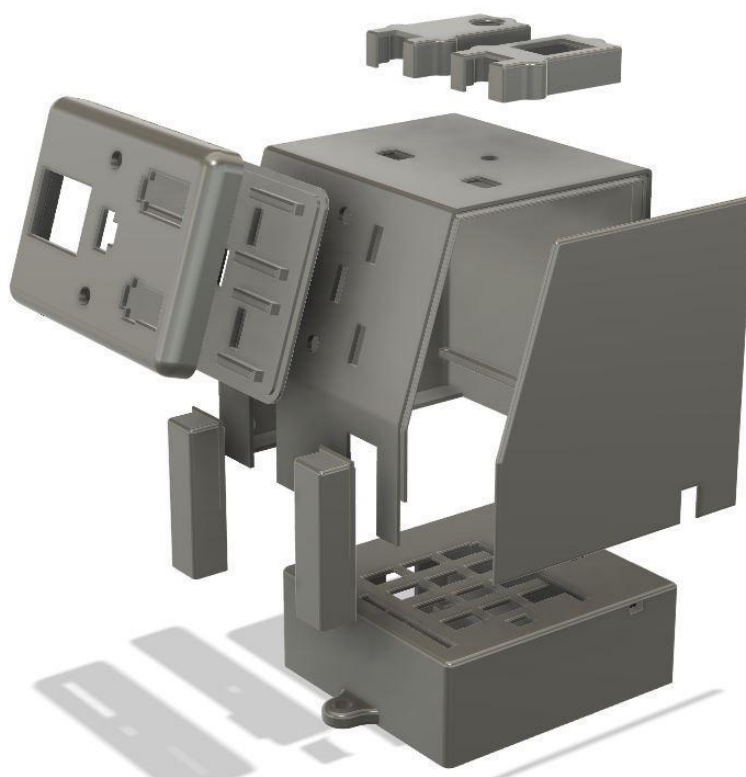
### 9.3 Ulkoasu arduinolle

Arduinolle 3D-suunniteltiin vielä kuoret, joilla saataisiin piilotettua johdot siististi, etteivät ne olisi sekaisin. (Kuva 60). Suunnittelussa pyrittiin ottamaan huomioon miltei kaikki kaapeleiden kytkentäpaikat, niin Arduinossa, kuin sen päälle liitettävässä sensorshieldissä. Toinen suunnittelun tavoite oli, että mitään osaa ei tarvitsisi liimata.



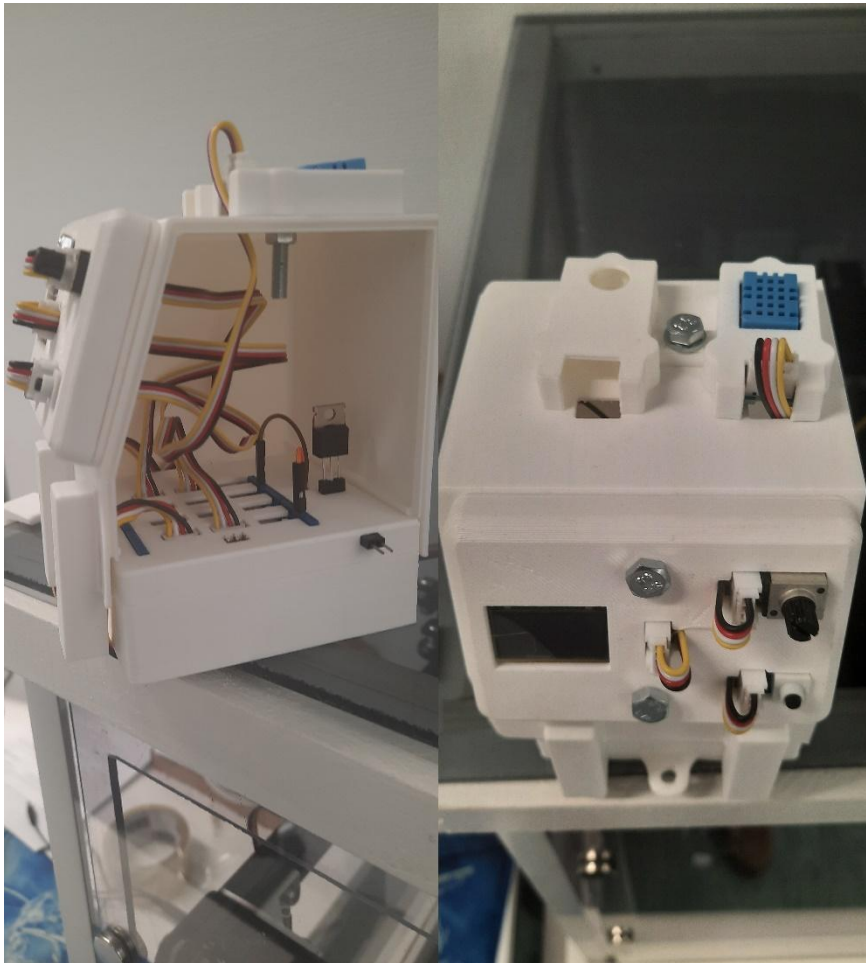
*Kuva 60. 3D-suunnitelma Arduinolle ja tuulettimelle menevien kaapeleiden piilottaminen.*

Myös kotelo suunniteltiin siten, että liimaa ei tarvitse käyttää, vaan kaikki osat olisivat helposti irrotettavissa (Kuva 61). Kotelon päällä olevaan osaan tulee ulkopuolelle tarkoitettut anturit. Oled-näyttölle, napille ja potentiometrille on kotelon etuosassa erillinen kappale, johon ne saa siististi kiinnitettyä. Ainoat liimattavat osat ovat kotelon etuosassa olevat kaksi suorakulman muotoista putkea. Putkien tarkoitus on naamioda kammioon menevien sisäensoreiden johdot. Kotelon sivussa oleva kansi on kiinnitettävissä press-fit tyylillä.



*Kuva 61. Kotelo komponenttien siistille esillepanolle.*

Kuvassa 62 Ainoat palaset, jotka on liimattu, ovat kammion sisälle menevien johtojen naamioputket.



*Kuva 62. Arduino kotelo 3D-tulostettuna.*

## 10 TULOKSET

Epäpuhtauksien ja lämpötilan testauksessa joko poistoilmatuuletin oli päällä tai ei ollut päällä. Kaikki testit tehtiin ABS-asetuksilla, koska siinä näkyi myös AQI-arvot. Kammion ulkopuolinen ilmanlaatuanturi oli mennyt rikki, joten kammion ulkopuolelta mitattavat AQI-arvot jäivät saamatta.

Kuvassa 63. Sisäpuolen ilmanlaatuanturi sijaitsee poistoilmatuulettimen vieressä kammion yläosassa ja lämpötila-anturi lämpötason korkeudella kammion oven vasemmassa reunassa.



Kuva 63. Anturien sijainti kammiossa.

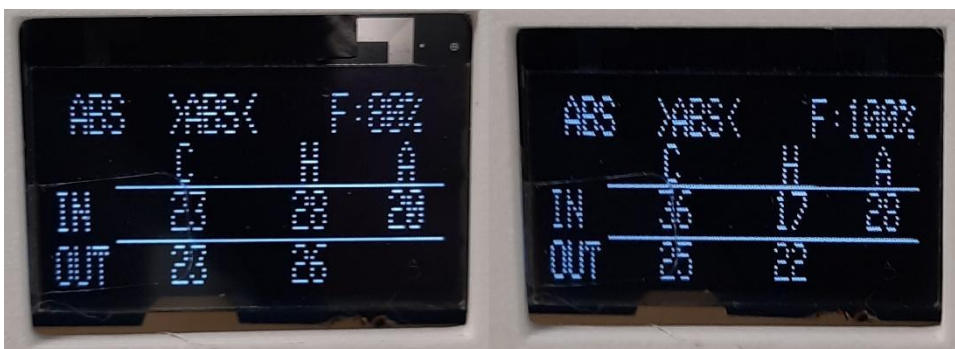
## 10.1 Testitulokset ABS

Tulostuksessa kammion poistoilmapuhallin oli kytketty pois päältä. Tason lämpötila oli 100 asteessa ja pursotus lämpötila 250 astetta. 3D-tulostimen pursotintuuletin oli 100 %. (Kuva 64). Lämpötila kammion sisällä 20 minuutin tulostuksen jälkeen kasvoi 10 astetta ja ilmanlaatu arvo AQI kasvoi hiljalleen korkeammaksi. Tulostuksen päätyttyä ero ilmanlaadussa oli 13 AQI arvoa.



Kuva 64. ABS 20minuutin tulostus, poistoilma kytketty pois. Arvot ennen ja jälkeen.

Tulostuksessa kammion poistoilmapuhallin oli kytketty päälle. Tason lämpötila oli 100 asteessa ja pursotuslämpötila 250 astetta. 3D-tulostimen pursotintuuletin oli 20 %. (Kuva 65). Lämpötila kammion sisällä 20 minuutin tulostuksen jälkeen kasvoi 13 astetta ja ilmanlaatu arvo AQI kasvoi hiljalleen korkeammaksi. Tulostuksen päätyttyä ero ilmanlaadussa oli 8 AQI arvoa.



Kuva 65. ABS 20 minuutin tulostus, Poistoilma päällä. Arvot ennen ja jälkeen.

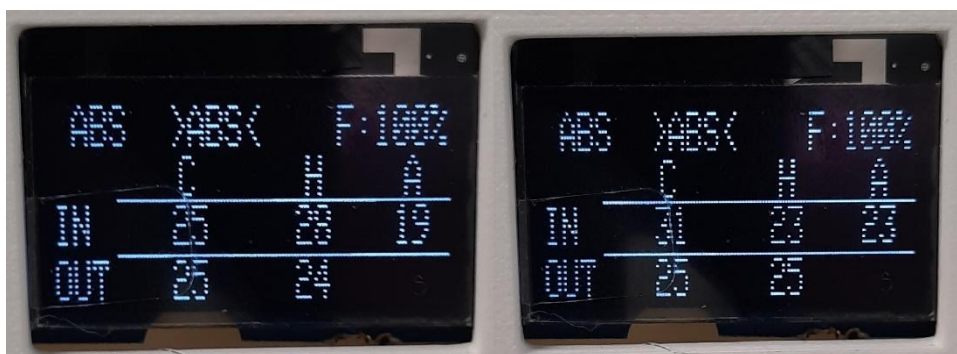
## 10.2 Testitulokset PLA

Tulostuksessa kammion poistoilmapuhallin oli kytketty pois päältä. Tason lämpötila oli 80 asteessa ja pursotuslämpötila 200 astetta. 3D-tulostimen pursotintuuletin oli 100 %. (Kuva 66). Lämpötila kammion sisällä 20 minuutin tulostuksen jälkeen kasvoi 6 astetta ja ilmanlaatu arvo AQI kasvoi hiljalleen korkeammaksi. Tulostuksen päätyttyä ero ilmanlaadussa oli 8 AQI arvoa.



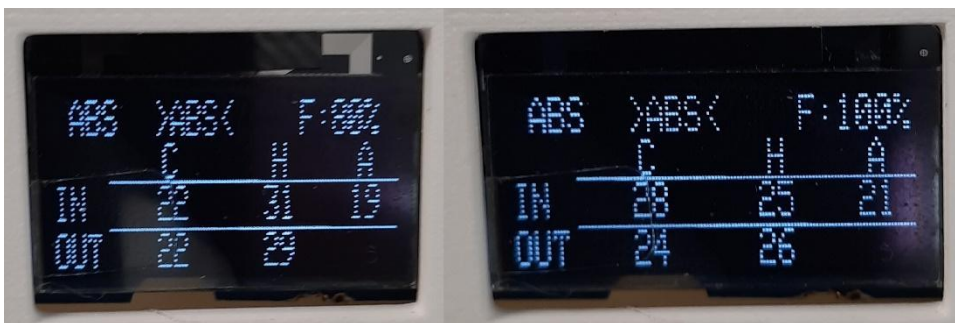
Kuva 66. PLA 20 minuutin tulostus, poistoilma kytketty pois. Arvot ennen ja jälkeen.

Tulostuksessa kammion poistoilmapuhallin oli kytketty päälle. Tason lämpötila oli 80 asteessa ja pursotuslämpötila 200 astetta. 3D-tulostimen pursotintuuletin oli 100 %. (Kuva 67). Lämpötila kammion sisällä 20 minuutin tulostuksen jälkeen kasvoi 6 astetta ja ilmanlaatu arvo AQI kasvoi hiljalleen korkeammaksi. Tulostuksen päätyttyä ero ilmanlaadussa oli 4 AQI arvoa.



Kuva 67. PLA 20 minuutin tulostus, poistoilma kytketty päälle. Arvot ennen ja jälkeen.

Tulostuksessa kammion poistoilmapuhallin oli kytketty päälle, myös HEPA-suodatin oli tulostuksen ajaksi otettu pois, millä saisi ilmavirtausta parannettua tuulettimelle. Tason lämpötila oli 80 asteessa ja pursotus lämpötila 200 astetta. 3D-tulostimen pursotintuuletin oli 100 %. (Kuva 68). Lämpötila kammion sisällä 20 minuutin tulostuksen jälkeen kasvoi 6 astetta ja ilmanlaatu arvo AQI kasvoi hieman. Tulostuksen päätyttyä ero ilmanlaadussa oli 2 AQI arvoa.



Kuva 68. PLA 20 minuutin tulostus, poistoilma kytketty päälle ja suodatin poistettu. Arvot en-nen ja jälkeen.



## 11 POHDINTA

Poistoilmatuuletuksen ollessa pois päältä pystyi arvioimaan, kuinka kammion sisällä hiukkaset ja muut epäpuhtaudet pääsivät vapaasti leijailemaan ilmanlaatuanturin lävitse, näin ollen anturin lukemat kasvoivat korkeammaksi. Kun taas poistoilmatuuletus oli päällä, muutti se ilmavirtausta sen verran, että hiukkaset ja epäpuhtaudet eivät menneet ilmanlaatuanturin lävitse, vaan suuntasivat ne suoraan virtauksen lähdeä kohti. Testeissä osoitettiin, että ilman minkäänlaista poistoilmavirtausta 3D-tulostamalla nanohiukkaset ja muut terveydelle haitalliset kaasut kerääntyvät 3D-tulostimen ympärille ja voivat pitkällä aikavälillä aiheuttaa terveysongelmia. Siksi on syytä huomioida hyvä tuuletus 3D-tulostimen ympärille. PLA-tulostustesteissä ilman suodatinta ja poistoilmatuuletuksen ollessa päällä kammion lämpötila nousi saman verran jokaisessa testissä. Projektissa ei onnistuttu kontrolloimaan lämpötilaa.

Lämpötila- ja AQI-arvojen seuranta olisi ollut helpompaa, jos ohjelmaan olisi lisännyt, että kammion ulko- sekä sisäpuolelta alin ja ylin arvo lämpötilasta ja ilmanlaadusta olisi tallennettu muistiin Arduinoon, mistä niitä olisi ollut mahdollista seurata OLED-näytöllä. Olisi hyvä pystyä muuttamaan raja-arvoja suoraan napin tai potentiometrin kautta ilman, että koodiin tarvitsee kajota. Lämpötilan kontrolloiminen projektin kannalta ei ollut lupaava edes Turbotuulettimen avulla, vaan tulisi se tarvitsemaan joko tehokkaamman, useamman tai jopa suoraan pursottimen ympärille kohdistetun poistoilmatuuletuksen.

## LÄHTEET

3Dsource. (10.8.2021). The complete history of 3D printing: From 1980 to 2022. Haettu 1.4.2023 osoitteesta <https://www.3dsourced.com/guides/history-of-3d-printing/>

AirNow. (n.d.) Haettu 24.3.2023 Air quality index (AQI) Basics <https://www.airnow.gov/aqi/aqi-basics/>

Arduino. (9.3.2023). What is Arduino? Haettu 4.3.2023 osoitteesta <https://docs.arduino.cc/learn/starting-guide/whats-arduino>

Arduino. (n.d.). Haettu 3.2.2023 Arduino sensor kit overview. <https://store.arduino.cc/products/arduino-sensor-kit-base?queryID=undefined>

Czeiszperger, M. (20.11.2020). Let's talk about 3D printer fire safety: "Will this thing burn down my house?" Haettu 1.3.2023 osoitteesta <https://www.3dupfitters.com/blogs/news/lets-talk-about-3d-printer-fire-safety-will-this-thing-burn-down-my-house>

Ekaran, S. (6.12.2022). 3D printer fumes: Are they dangerous to your health? Haettu 21.3.2023 osoitteesta <https://www.makeuseof.com/3d-printer-fumes-dangerous-your-health/>

Filamentti.com, (n.d.). Haettu 20.3.2023 Filamenttien tekniset ominaisuudet. <https://www.filamentti.com/tekniset-tiedot/>

Fitzgerald, S. & Shiloh, M. (2015). The Arduino project book. Haettu 2.3.2023 osoitteesta [https://www.academia.edu/41764595/ARDUINO\\_PROJECTS\\_BOOK](https://www.academia.edu/41764595/ARDUINO_PROJECTS_BOOK)

Hamill, S. (10.7.2022). Recycling 3D printer filament: How & Which filament can be recycled? Haettu 17.3.2023 osoitteesta <https://www.3dsourced.com/guides/recyclable-3d-printer-filaments/>

Joutsenvirta, A. (n.d.) Haettu 26.2.2023 <http://web.uniarts.fi/akustiikka/index7f5e.html?id=13&la=fi>

Long, M. (15.12.2021). Complete guide to air quality sensors. Haettu 12.4.2023 osoitteesta <https://www.electromaker.io/blog/article/air-quality-sensors#:~:text=An%20air%20quality%20sensor%20measures%20the%20size%2C%20and,is%20calculated.%20How%20exactly%20this%20is%20accomplished%20varies.>

Lunkka, K. (2011). Pulssileveysmodulaation käyttö moottorihjauksessa [AMK-opinnäytetyö, Lahden Ammattikorkeakoulu]. Theseus. [https://www.theseus.fi/bitstream/handle/10024/32920/Lunkka\\_Kalle.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/32920/Lunkka_Kalle.pdf?sequence=1)

Martini, L. (1.3.2023). What is the difference between PLA and ABS filaments in terms of nanoparticle emission? Haettu 27.3.2023 osoitteesta

<https://www.alveo3d.com/en/what-is-the-difference-between-pla-and-abs-filaments-in-terms-of-nanoparticle-emissions/>

Murray, M. (n.d.) How I2C works (I2C explained simply). Haettu 27.2.2023 osoitteesta <https://www.thegeekpub.com/18351/how-i2c-works-i2c-explained-simply/>

Ong, D. (2021). Introducing DHT20 temperature and humidity sensor and comparing it with DHT11!. Haettu 22.2.2023 osoitteesta <https://www.seeedstudio.com/blog/2021/05/25/introducing-dht20-temperature-and-humidity-sensor-and-comparing-it-with-dht11/>

Redaktion, D. (24.3.2020). Pioneers of printing: Chuck Hull and the beginning of 3D printing. Haettu 1.2.2023 osoitteesta <https://blog.drupa.com/en/chuck-hull-3d-printing-en/>

SciJinks. (2023). How is air quality measured?. Haettu 15.3.2023 osoitteesta <https://scijinks.gov/air-quality/>

Sher, D. (25.10.2015). How toxic are ABS & PLA Fumes? Haettu 1.4.2023 osoitteesta <https://3dprintingindustry.com/news/toxic-abs-pla-fumes-3dsafety-org-inquires-vocs-60796/~:text=A%20second%20critical%20aspect%20is%20that%20relating%20to,times%20those%20that%20occur%20when%20using%20PLA%20filament.>

Ubel, M. (4.11.2021). 3D printing materials – The ultimate guide. Haettu 2.3.2023 osoitteesta <https://all3dp.com/1/3d-printing-materials-guide-3d-printer-material/>

Väisänen, A. (2017). 3D-tulostuksen terveydelle haitalliset altisteet ja työhygieniä [AMK-opinnäytetyö, Savonia-ammattikorkeakoulu]. Theseus. <https://www.theseus.fi/handle/10024/140283>

Valdez, J & Becker, J. (2015). Understanding the I2C Bus. Texas instruments. <https://www.ti.com/lit/an/slva704/slva704.pdf>

Zait, A. (22.4.2018). An introduction to Arduino uno pinout. Haettu 27.3.2023 osoitteesta <https://www.circuito.io/blog/arduino-uno-pinout/>

## LIITE 1

```
1/ ---Library---//
#include "Air_Quality_Sensor.h"
#include "Arduino.h"
#include "Arduino_SensorKit.h"
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "DHT.h"

2/ ---Do one time---//
int Clear_air_Job_Done = 0;

3/ ---OLED---//
#define OLED_RESET -1
Adafruit_SSD1306 display(OLED_RESET);

4/ ---Air Quality IN---//
AirQualitySensor sensorIN(A1); // IN
int AIR_State_IN = 0;
unsigned long AIR_previousMillis_IN = 0;
const long AIR_interval_IN = 2000; //1000 = 1 second

5/ ---Air Quality OUT---//
AirQualitySensor sensorOUT(A2); //OUT
int AIR_State_OUT = 0;
unsigned long AIR_previousMillis_OUT = 0;
const long AIR_interval_OUT = 10000; //1000 = 1 second

7/ ---Temperature & Humidity IN---//
#define DHTPIN_IN (0x38);
#define DHTTYPE_IN DHT20 //OUT PIN IC2 DHT20
DHT dhtIN(DHTTYPE_IN);
int TH_State_IN = 0;
unsigned long TH_previousMillis_IN = 0;
const long TH_interval_IN = 2000; //1000 = 1 second

8/ ---Temperature & Humidity OUT---//
#define DHTPIN_OUT 5
```

```

#define DHTTYPE_OUT DHT11 //IN PIN 2 DHT 11
DHT dhtOUT(DHTPIN_OUT, DHTTYPE_OUT);

#if defined(ARDUINO_ARCH_AVR)
#define SERIAL Serial
#elif defined(ARDUINO_ARCH_SAMD) || defined(ARDUINO_ARCH_SAM)
#define SERIAL SerialUSB
#else
#define SERIAL Serial
#endif

int TH_State_OUT = 0;
unsigned long TH_previousMillis_OUT = 0;
const long TH_interval_OUT = 10000; //1000 = 1 second

9/ ---ABS & PLA---//
unsigned int Temperature_IN;
unsigned int Humidity_IN;
unsigned int AirValue_IN;
unsigned int Temperature_OUT;
unsigned int Humidity_OUT;
unsigned int AirValue_OUT;
int WAIT_State = 0;
unsigned long WAIT_previousMillis = 0;
const long WAIT_interval = 1000; //1000 = 1 second
int Fan60 = 153; //Fan speed 60% = 153pwm
int Fan80 = 204; //Fan speed 80% = 204pwm
int Fan100 = 255; //Fan speed 100% = 255pwm

10/ ---MANUAL---//
int potentiometer = A0; //Assign to pin A0
int fanpin = 3; //Assign to pin ~3
int pot_value = 0;
int potvalue = 0;
int prosentvalue;

int MANUAL_State = 0;
unsigned long MANUAL_previousMillis = 0;
const long MANUAL_interval = 200; //1000 = 1 second

11/ ---BUTTON---//
#define button 8 //connect button to D4
int buttonState = 0; //current state of the button
int pressTime = 0; //current state of the timer
char *names[] = { " ABS", " PLA", " MAN", " OFF" };

```

```

int nameValue;
int lastnamevalue;
char *Accepted[] = { ">ABS<      ", ">PLA<      ", ">MANUAL<", ">OFF<      "
};
int accepted_value;
int lastButtonState = 0;

12/ ---SETUP---//
void setup() {

    pinMode(button, INPUT);          //initialize the pushbutton pin as an
input:
    pinMode(potentiometer, INPUT); //Sets the pinmode to input
    pinMode(fanpin, OUTPUT);
    pinMode(DHTPIN_OUT, OUTPUT);
    Serial.begin(9600);
    sensorIN.init();
    sensorOUT.init();
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.setRotation(90);
    display.clearDisplay();
    dhtIN.begin();
    dhtOUT.begin();
    Environment.begin();
    display.setFont();
    display.setTextColors(WHITE);

    while (!Serial)
        ;

    display.setCursor(0, 0);
    display.println("Waiting sensors");
    display.setCursor(0, 8);
    display.println("to init.");
    display.display();
    delay(20000);

    if (sensorIN.init()) {

        display.setCursor(0, 16);
        display.println("AIRIN ready");
        display.display();
    }
    delay(1000);
    if (sensorOUT.init()) {

```

```

    display.setCursor(0, 24);
    display.println("AIROUT ready");
    display.display();
}

delay(1000);
display.clearDisplay();
TEMP_HUMI_IN();
TEMP_HUMI_OUT();
ABS_SET();
SHORT_PRESS_BUTTON_RELEASE();
PRINT();
}

```

13/ ///////////////////////////////////BUTTON ///////////////////////////////////

```

void loop() {

    buttonState = digitalRead(button); //reading buttonState, which can
    be either pressed = 1 or not pressed = 0.

    if (pressTime == 0 && accepted_value != 3) { // Read temperature and
    humidity sensors when button is not pressed and accepted value is 0, 1,
    2

        TEMP_HUMI_IN(); //Read every 2s temperature and humidity sensors
        INSIDE
        TEMP_HUMI_OUT(); //Read every 10s temperature and humidity sensors
        OUTSIDE
    }
    if (buttonState != lastButtonState) { // If button is pressed
    (buttonstate = 1 and lastButtonState = 0) If button is not pressed
    (buttonstate = 0 and lastButtonState = 0)
        pressTime++; // Presstime starts when button is pressed and ends
        when button is released
    }
    if (accepted_value != 3) { // If acceptedvalue is 3 => (OFF) Skip to
    else command

        if ((buttonState != lastButtonState) && (pressTime <= 4)) {

            SHORT_PRESS_BUTTON_PRESSING(); // Showing when namevalue and
            accepted_value will change
        }
        if ((buttonState == lastButtonState) && (pressTime <= 5) &&
        (pressTime >= 1)) {

```

```

        SHORT_PRESS_BUTTON_RELEASE(); // Releasing button for short
press time (Changes the namevalue)
    }
    if ((buttonState != lastButtonState) && (presTime >= 5)) {

        LONG_PRESS_BUTTON_PRESSING(); // Pressing button long time Shows
when accepted_value will change
    }
    if ((buttonState == lastButtonState) && (presTime >= 5)) {

        LONG_PRESS_BUTTON_RELEASE(); // Releasing button for long press
time (Change the accepted_value)

    } else if (accepted_value == 0) {
        ABS(); // Releasing button on long press time and acceptedvalue
is 0 => (ABS)

    } else if (accepted_value == 1) {
        PLA(); // Releasing button on long press time and acceptedvalue
is 1 => (PLA)

    } else if (accepted_value == 2) {
        MANUAL(); // Releasing button on long press time and
acceptedvalue is 2 => (MANUAL)
    }

    } else {
        OFF(); // Releasing button on long press time and accepted value
is 3 => (OFF) Skips the PRESS_BUTTON
    }
}

```

14/ //////////////////////////////////// MANUAL ////////////////////////////////////

```

static MANUAL() {

    unsigned    long    MANUAL_currentMillis    =    millis();

    if (MANUAL_currentMillis - MANUAL_previousMillis >= MANUAL_interval)
    {
        //save the last time you used the sensor
        MANUAL_previousMillis = MANUAL_currentMillis;

        //if the sensor is off turn it on and vice versa:
        if (MANUAL_State == LOW) {
            MANUAL_State = HIGH;
        } else {

```



```

    MANUAL_State = LOW;
}

    pot_value = analogRead(potentiometer);    // Read the value from
the potentiometer connected to the A0 pin
    potvalue = map(pot_value, 0, 1022, 0, 255); // Map the value from
0, 1023 to 0, 255 (77 pwm is 30% and 255 pwm is 100%)

    {
        analogWrite(fanpin, pot_value); // writes the potvalue value
0... 255 to fanpin
    }

    presentvalue = map(pot_value, 0, 1022, 30, 100); // printed
present value to u8x8 0, 255 to 30, 100 (0 pwm is 30% and 255 pwm is
100%)
    display.setCursor(91, 0);
    display.print("F:");
    display.setCursor(102, 0);
    display.print(presentvalue); // prints the presentvalue 30% ...
100% to OLED
    display.print("% ");
    display.display();
}
}

```

15/ ////////////////////////////////// LONG\_PRESS\_BUTTON\_PRESSING //////////////////////////////////

```

static LONG_PRESS_BUTTON_PRESSING() {
    display.setTextColor(WHITE, BLACK);
    display.setCursor(0, 0);
    display.print(">"); //Shows when button is pressed enough to change
the accepted_value
    display.display();
}

```

16/ ////////////////////////////////// SHORT\_PRESS\_BUTTON\_RELEASE //////////////////////////////////

```

static SHORT_PRESS_BUTTON_RELEASE() {
    display.setTextColor(WHITE, BLACK);
    display.setCursor(0, 0);
    display.print(names[nameValue]);
    display.display();
    pressTime = 0; //Reset presstime
    nameValue++; //NameValue +1

    if (nameValue == 4) {
        // If nameValue is equal to 4 then nameValue is 0
    }
}

```

```

    nameValue = 0;
}
}

```

17/ ////////////////////////////////// LONG\_PRESS\_BUTTON\_RELEASE //////////////////////////////////

```

static LONG_PRESS_BUTTON_RELEASE() {
    accepted_value = nameValue; // Accepted_value changes same than
nameValue
    accepted_value -= 1;        // When turning accepted_value to same as
nameValue. accepted_value is one too much so minus 1 at accepted_value
    if (accepted_value == -1)  // if accepted_values is equal to -1 then
it turns to 3
        accepted_value = 3;    // accepted_value is 3
    display.setTextColor(WHITE, BLACK);
    display.setCursor(37, 0);
    display.print(Accepted[accepted_value]);
    display.display();
    pressTime = 0;            // Reset presstime
}

```

18/ ////////////////////////////////// SHORT\_PRESS\_BUTTON\_PRESSING //////////////////////////////////

```

static SHORT_PRESS_BUTTON_PRESSING() {
    display.setCursor(0, 0);
    display.println("<"); //Shows when button is pressed enough to
change the NameValue
    display.display();
}

```

19/ ////////////////////////////////// OFF //////////////////////////////////

```

static OFF() {
    display.clearDisplay(); // Clears OLED screen
    digitalWrite(fanpin, 0); // change the PWM signal to 0
    display.display();

    buttonState = digitalRead(button);

    if (buttonState != lastButtonState) { //When button is pressed
        ABS_SET();
        SHORT_PRESS_BUTTON_RELEASE();
        PRINT(); //Prints C H A and lines
    }
}

```

20/ ////////////////////////////////// Print //////////////////////////////////

```

static PRINT() {
    display.setCursor(0, 16);
    display.print("IN");
    display.setCursor(0, 25);
    display.print("OUT");
    display.setCursor(40, 8);
    display.print("C");
    display.setCursor(81, 8);
    display.print("H");
    display.setCursor(113, 8);
    display.print("A");
    display.drawLine(20, 24, 128, 24, WHITE);
    display.drawLine(20, 15, 128, 15, WHITE);
    display.display();
}

```

21/ ////////////////////////////////// Temperature & Humidity IN //////////////////////////////////

```

static TEMP_HUMI_IN() {
    unsigned long TH_currentMillis_IN = millis();

    if (TH_currentMillis_IN - TH_previousMillis_IN >= TH_interval_IN) {
        //save the last time you used the sensor
        TH_previousMillis_IN = TH_currentMillis_IN;

        //if the sensor is off turn it on and vice-versa:
        if (TH_State_IN == LOW) {
            TH_State_IN = HIGH;
        } else {
            TH_State_IN = LOW;
        }

        Temperature_IN = dhtIN.readTemperature(); // Reads sensors
temperature
        Humidity_IN = dhtIN.readHumidity(); // Reads sensors humidity

        display.setTextColor(WHITE, BLACK); //Text color is showed first in
white then it erased whit black
        display.setCursor(38, 16); // Where is the text starting
(X,Y)
        display.println(Temperature_IN); //Shows temperature in OLED
        display.setCursor(78, 16); // Where is the text starting
(X,Y)
        display.println(Humidity_IN); //Shows humidity in OLED
        display.display(); // displays everything what is
before this
    }
}

```

```
}
```

```
22/ ////////////////////////////////// Air Quality IN //////////////////////////////////
```

```
void AirQuality_IN(){
```

```
    unsigned long AIR_currentMillis_IN = millis();
```

```
    if (AIR_currentMillis_IN - AIR_previousMillis_IN >= AIR_interval_IN)
    {
```

```
        //save the last time you blinked the LED
```

```
        AIR_previousMillis_IN = AIR_currentMillis_IN;
```

```
        //if the LED is off turn it on and vice-versa:
```

```
        if (AIR_State_IN == LOW) {
```

```
            AIR_State_IN = HIGH;
```

```
        } else {
```

```
            AIR_State_IN = LOW;
```

```
        }
```

```
        int qualityIN = sensorIN.slope();
```

```
        AirValue_IN = sensorIN.slope();
```

```
        Serial.print("Sensor IN value: ");
```

```
        Serial.println(sensorIN.getValue());
```

```
        Serial.println(sensorIN.slope());
```

```
        display.setTextColor(WHITE, BLACK);
```

```
        display.setCursor(110, 16);
```

```
        display.display();
```

```
        if (qualityIN == AirQualitySensor::FORCE_SIGNAL) { // 201- = Don't breathe the air
```

```
            display.println(sensorIN.getValue());
```

```
        } else if (qualityIN == AirQualitySensor::HIGH_POLLUTION) { // 150-200 = High pollution
```

```
            display.println(sensorIN.getValue());
```

```
        } else if (qualityIN == AirQualitySensor::LOW_POLLUTION) { // 51-149 = Low pollution
```

```
            display.println(sensorIN.getValue());
```

```
            if (sensorIN.getValue() < 100){
```

```
                display.fillRect(122, 16, 5, 7, BLACK);
```

```
                display.display();
```

```
            }
```

```
        } else if (qualityIN == AirQualitySensor::FRESH_AIR) { // 0-50 = Fresh air
```

```
            display.println(sensorIN.getValue());
```

```
    }  
  }  
}
```

23/ ////////////////////////////////// Temperature & Humidity OUT //////////////////////////////////

```
static TEMP_HUMI_OUT() {  
  unsigned long TH_currentMillis_OUT = millis();  
  
  if (TH_currentMillis_OUT - TH_previousMillis_OUT >= TH_interval_OUT)  
  {  
    //save the last time you used the sensor  
    TH_previousMillis_OUT = TH_currentMillis_OUT;  
  
    //if the sensor is off turn it on and vice-versa:  
    if (TH_State_OUT == LOW) {  
      TH_State_OUT = HIGH;  
    } else {  
      TH_State_OUT = LOW;  
    }  
  
    Temperature_OUT = dhtOUT.readTemperature();  
    Humidity_OUT = dhtOUT.readHumidity();  
  
    display.setTextColor(WHITE, BLACK);  
    display.setCursor(38, 25);  
    display.println(Temperature_OUT);  
    display.setCursor(78, 25);  
    display.println(Humidity_OUT);  
    display.display();  
  }  
}
```

24/ ////////////////////////////////// Air Quality OUT //////////////////////////////////

```
void AirQuality_OUT() {  
  
  unsigned long AIR_currentMillis_OUT = millis();  
  
  if (AIR_currentMillis_OUT - AIR_previousMillis_OUT >=  
AIR_interval_OUT) {  
    //save the last time you blinked the LED  
    AIR_previousMillis_OUT = AIR_currentMillis_OUT;  
  
    //if the LED is off turn it on and vice-versa:  
    if (AIR_State_OUT == LOW) {  
      AIR_State_OUT = HIGH;  
    }  
  }  
}
```

```

    } else {
        AIR_State_OUT = LOW;
    }

    int qualityOUT = sensorOUT.slope();
    AirValue_OUT = sensorOUT.slope();
    Serial.print("Sensor OUT value: ");
    Serial.println(sensorOUT.getValue());
    Serial.println(sensorOUT.slope());
    display.setTextColor(WHITE, BLACK);
    display.setCursor(110, 25);
    display.display();

    if (qualityOUT == AirQualitySensor::FORCE_SIGNAL) { // 201- =
Don't breathe the air
        display.println(sensorOUT.getValue());
    } else if (qualityOUT == AirQualitySensor::HIGH_POLLUTION) { //
150-200 = High pollution
        display.println(sensorOUT.getValue());
    } else if (qualityOUT == AirQualitySensor::LOW_POLLUTION) { // 51-
149 = Low pollution
        display.println(sensorOUT.getValue());
        if (sensorOUT.getValue() < 100){
            display.fillRect(122, 25, 5, 7, BLACK);
            display.display();
        }
    } else if (qualityOUT == AirQualitySensor::FRESH_AIR) { // 0-50 =
Fresh air
        display.println(sensorOUT.getValue());
    }
}
}
}

```

25/ ////////////////////////////////// SETS ABS BACK //////////////////////////////////

```

static ABS_SET() {
    accepted_value = 0;
    nameValue = 0;
    display.setTextColor(WHITE, BLACK);
    display.setCursor(37, 0);
    display.print(Accepted[accepted_value]); //u8x8 prints to line ("")
    display.setCursor(0, 0); //Choosing line to print
    display.println(names[nameValue]); //u8x8 prints to line ("")
    display.display();
}

```

26/ ////////////////////////////////// FAN100%////////////////////////////////////

```

static FAN100() {
    analogWrite(fanpin, Fan100);
    display.setTextColor(WHITE, BLACK);
    display.setCursor(91, 0);
    display.print("F:100% ");
    display.display();
}

```

27/ ////////////////////////////////// FAN80%////////////////////////////////

```

static FAN80() {
    analogWrite(fanpin, Fan80);
    display.setTextColor(WHITE, BLACK);
    display.setCursor(91, 0);
    display.print("F:80% ");
    display.display();
}

```

28/ ////////////////////////////////// FAN60%////////////////////////////////

```

static FAN60() {
    analogWrite(fanpin, Fan60);
    display.setTextColor(WHITE, BLACK);
    display.setCursor(91, 0);
    display.print("F:60% ");
    display.display();
}

```

29/ ////////////////////////////////// ABS //////////////////////////////////

```

static ABS() {
    Clear_air_Job_Done = 0; // When Clear_air_Job_Done is 0 Airquality
sensor will be readed
    AirQuality_IN(); //Read every 2sec
    // AirQuality_OUT(); //Read every 10sec

```

```

unsigned long WAIT_currentMillis = millis();

```

```

if (WAIT_currentMillis - WAIT_previousMillis >= WAIT_interval) {
    //save the last time you blinked the LED
    WAIT_previousMillis = WAIT_currentMillis;

```

```

//if the LED is off turn it on and vice-versa:

```

```

if (WAIT_State == LOW) {

```

```

    WAIT_State = HIGH;
} else {
    WAIT_State = LOW;
}
//Temperature out 19 / in 19
if ((Temperature_OUT <= 19) && (Temperature_IN <= 19) &&
(accepted_value == 0)) {
    FAN60();
    //Temperature out 19 / in 19-24
} else if ((Temperature_OUT <= 19) && (Temperature_IN >= 20 &&
Temperature_IN <= 24) && (accepted_value == 0)) {
    FAN80();
    //Temperature out 19 / in 25
} else if ((Temperature_OUT <= 19) && (Temperature_IN >= 25) &&
(accepted_value == 0)) {
    FAN100();
    //Temperature out 20-24 / in 20-24
} else if ((Temperature_OUT >= 20 && Temperature_OUT <= 24) &&
(Temperature_IN >= 20 && Temperature_IN <= 24) && (accepted_value ==
0)) {
    FAN80();
    //Temperature out 20-24 / in 25
} else if ((Temperature_OUT >= 20 && Temperature_OUT <= 24) &&
(Temperature_IN >= 25) && (accepted_value == 0)) {
    FAN100();
    //Temperature out 25 / in 25
} else if ((Temperature_OUT >= 25) && (accepted_value == 0)) {
    FAN100();
}
}
}
}

```

30/ ////////////////////////////////// PLA //////////////////////////////////

```

static PLA() {
    if (Clear_air_Job_Done == 0) {
        ClearAIR();
        Clear_air_Job_Done = 1; // When Clear_air_Job_Done is 1 Airquality
sensor will be not readed
    }
}

```

```

unsigned long WAIT_currentMillis = millis();

```

```

if (WAIT_currentMillis - WAIT_previousMillis >= WAIT_interval) {
    //save the last time you blinked the LED
    WAIT_previousMillis = WAIT_currentMillis;
}

```



```

//if the LED is off turn it on and vice-versa:
if (WAIT_State == LOW) {
    WAIT_State = HIGH;
} else {
    WAIT_State = LOW;
}
//Temperature out 19 / in 19
if ((Temperature_OUT <= 19) && (Temperature_IN <= 19) &&
(accepted_value == 1)) {
    FAN60();
    //Temperature out 19 / in 19-24
} else if ((Temperature_OUT <= 19) && (Temperature_IN >= 20 &&
Temperature_IN <= 24) && (accepted_value == 1)) {
    FAN80();
    //Temperature out 19 / in 25
} else if ((Temperature_OUT <= 19) && (Temperature_IN >= 25) &&
(accepted_value == 1)) {
    FAN100();
    //Temperature out 20-24 / in 20-24
} else if ((Temperature_OUT >= 20 && Temperature_OUT <= 24) &&
(Temperature_IN >= 20 && Temperature_IN <= 24) && (accepted_value ==
1)) {
    FAN80();
    //Temperature out 20-24 / in 25
} else if ((Temperature_OUT >= 20 && Temperature_OUT <= 24) &&
(Temperature_IN >= 25) && (accepted_value == 1)) {
    FAN100();
    //Temperature out 25 / in 25
} else if ((Temperature_OUT >= 25) && (accepted_value == 1)) {
    FAN100();
}
}
}
}

```

31/ ////////////////////////////////// Do one time //////////////////////////////////

```

static ClearAIR() {
    display.setTextColor(WHITE, BLACK);
    display.setCursor(120, 16);
    display.println(" ");
    display.setCursor(120, 25);
    display.println(" ");
    display.display();
}

```