



TALLINNA TEHNIKAÜLIKOOL

INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

3D-PRINTERITE NUTIKA KAITSEÜMBRISE JUHTSÜSTEEMI ARENDUS

DEVELOPMENT OF CONTROL SYSTEM FOR 3D PRINTER SMART ENCLOSURE
BAKALAUREUSETÖÖ

Üliõpilane: Sten Erik Aunpu

Üliõpilaskood: 179596EAAB

Juhendaja: Tarmo Korõtko, teadur

Tallinn, 2020

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“.....” 2020

Autor:

/ allkiri /

Töö vastab bakalaureusetööle esitatud nõuetele

“.....” 2020

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

“.....”2020

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Sten Erik Aunpu

Lõputöö liik: Bakalaureusetöö

Töö pealkiri: 3D-printerite nutika kaitseümbrise juhtsüsteemi arendus

Kuupäev: 20.05.2020

76 lk (lõputöö lehekülgede arv koos lisadega)

Ülikool: Tallinna Tehnikaülikool

Teaduskond: Inseneriteaduskond

Instituut: Elektroenergeetika ja mehhatroonika instituut

Töö juhendaja(d): teadur Tarmo Korõtko

Töö konsultant (konsultandid): Priit Norak

Sisu kirjeldus:

Antud bakalaureusetöös on kirjeldatud 3D-printerite nutika kaitseümbrise juhtsüsteemi arenduskäiku ja valmistamist. Juhtsüsteemi primaarseks ülesandeks on kaitseümbrise sees stabiilse temperatuuri hoidmine, et vähendada 3D-printitud detailide valmistamisel defekte nagu näiteks kaardumine. Seletatakse lahti sulatatud materjali sadestamise ehk FDM tehnoloogia olemus ning millised terviseriskid võivad kaasneda selle tehnoloogia kasutamisega. Tuuakse välja kaardumise tekkimise põhjused FDM tehnoloogia kasutamisel ning kuidas seda on võimalik vähendada. Autor projekteerib ja koostab seadme elektriskeemid, programmi ja elektroonika ning andurite korpused. Lõputöö tulemusena valminud seadme võimekust hoida stabiilset temperatuuri hinnatakse mõõtes kaitseümbrise sisetemperatuuri muutumist 3D-printimise ajal.

Märksõnad: 3D-printimine, FDM, ABS plastmass, kaitseümbris, ventilatsioon, kaardumine, VOC, mikroosakesed.

ABSTRACT

Author: Sten Erik Aunpu

Type of the work: Bachelor Thesis

Title: Development of control system for 3D printer smart enclosure

Date: 20.05.2020

76 pages (the number of thesis pages including appendices)

University: Tallinn University of Technology

School: School of Engineering

Department: Department of Electrical Power Engineering and Mechatronics

Supervisor(s) of the thesis: research scientist Tarmo Korõtko

Consultant(s): Priit Norak

Abstract:

The following Bachelor's thesis describes the development and construction of a control system for 3D printer smart enclosure. The primary function of the control system is to maintain a stable inside the enclosure to minimize the defect like warping that can arise when 3D-printing parts. Explains what is fused deposition modelling, FDM, based 3D-printing technology and what are its potential health risks to people. Reasons why warping happens and how it can be avoided are brought out. Author designs and constructs electrical schematic, program and electronic component housing for the control device. Manufactures devices ability to maintain a stable temperature was evaluated based on measuring temperature changes inside the enclosure during 3D-printing.

Keywords: 3D-printing, FDM, ABS plastic, enclosure, ventilation, warping, VOC, microparticles

LÕPUTÖÖ ÜLESANNE

Lõputöö teema:	3D printerite nutika kaitseümbrise juhtsüsteemi arendus
Lõputöö teema inglise keeles:	Development of control system for 3D printer smart enclosure
Üliõpilane:	Sten Erik Aunpu, 179596EAAB
Eriala:	Elektroenergeetika ja mehhatroonika
Lõputöö liik:	bakalaureusetöö
Lõputöö juhendaja:	Tarmo Korõtko
Lõputöö ülesande kehtivusaeg:	20.05.2020
Lõputöö esitamise tähtaeg:	20.05.2020

Üliõpilane (allkiri)

Juhendaja (allkiri)

Õppekava juht (allkiri)

1. Teema põhjendus

3D-printimisega tegelevad ettevõtted, kus on kasutusel korraga mitmed printerid, seisavad silmitsi erinevate probleemidega. 3D-printimisel kasutatakse enamjaolt sulatatud materjali sadestamise (ing. k. fused deposition modelling e. FDM) tehnoloogiat, mille kasutamisel paisatakse ümbritsevasse keskkonda mikro-osakesi, mis võivad sissehingatava õhu kaudu inimese organismi siseneda ning seal ladestuda. Lisaks paisatakse plastmassi sulatamise käigus õhku erinevaid mürgiseid osakesi. Ettevõttel on personali kaitsmiseks on tarvis eraldada 3D-printerid ülejäänud keskkonnast ning isoleerida ja filtreerida õhk printerite ümber. Lisaks on FDM tehnoloogiaga printimine temperatuuritundlik ja vajab kvaliteetsete detaili valmistamiseks temperatuuri ligikaudu 35 °C. Turul kättesaadavaid 3D-printrite nutikate kaitseümbriste lahendusi iseloomustab suhteliselt kõrge hind ning mahutavus ühele printerile.

Käesoleva töö raames luuakse ning katsetatakse universaalne juhtsüsteem eelnevalt arendatud 3D-printerite kaitseümbrisele. Arendatud juhtsüsteemi on erinevatesse 3D-printerite kaitseümbristesse lihtsasti integreeritav ning selle funktsionaalsus on lisaseadmete abil laiendatav.

2. Töö eesmärk

Tagada 3D printerite kvaliteetne töö nende töökeskkonna ühtlase temperatuuri hoidmise kaudu ning tagada inimesele ohutu töökeskkond vältides printimisel eralduvate mürgiste gaaside sattumist töökotta.

3. Lahendamisele kuuluvate küsimuste loetelu:

1. Kas nutika kaitseümbrise jaoks ehitatud juhtsüsteem suudab juhtida oma sisetemperatuuri ning õhuvahetust?
2. Kas nutika kaitseümbrise abil on võimalik tagada ühtlane temperatuur kogu printimise protsessi jooksul?
3. Kas nutika kaitseümbrise abil õhu filtreerimine printimise ajal ja peale printmist tagab inimesele ohutu töökeskkonna?

4. Lähteandmed

Lähteandmetena kasutan:

- Kasutatavate 3D printerite tehnilist dokumentatsiooni,
- Kaitseümbrise tehnilist dokumentatsiooni,
- Tehnoloogiliste seadmete andmelehti,
- Dokumenteeritud vaatlusi 3D printerite tööst.

5. Uurimismeetodid

Uurimismeetoditena kasutan:

- Katsetusi ja vaatlusi 3D printeritel,
- Keskkonna ja prinditava detaili ümbruses temperatuuri mõõtmist ning analüüsi,
- Keskkonna õhukvaliteedi parameetrite mõõtmist ning analüüsi.
- Vaatlusi prinditud detaili kvaliteedi hindamiseks.

Vaatlused dokumenteerin protokollide ja fotode abil. Mõõteandmete analüüsiks kasutan tabelarvutusprogramme.

6. Graafiline osa

Töö graafiline osa sisaldab nutika kaitseümbrise juhtsüsteemi elektriskeeme, tarkvara tööd iseloomustav voodiagramm ja kasutajaliideses navigeerimist selgitav joonis. Lisaks esitatakse graafiliselt katsete käigus kogutud mõõtetulemused ning fotod süsteemi ehitusest ja katsetustest. Juhtsüsteemi tarkvara väljatrükk ning muud olulised materjalid lisatakse töö lisade hulka.

7. Töö struktuur

Lõputöö struktuur:

1. Sissejuhatus – Probleemi kirjeldus.
2. Taustinfo kogumine ja tehnilise tingimuste täpsustamine.
3. Projekteerimine - Komponentide valik, elektri primaar- ja sekundaarosa projekteerimine ning tarkvara struktuuri ning juhtimisalgoritmide loomine.
4. Teostus – Ülevaade süsteemi realiseerimise kohta.
5. Kasutusele võtmine ja testimine - Esitatakse andmed, mis on kogutud eksperimentide tulemusel või käiku andmisel, mille abil hinnata püstitatud probleemide lahendamist.
6. Kokkuvõte - Võtab lühidalt kokku kogu eelneva, annab konkreetse hinnangu kas püstitatud probleem sai lahendatud ning pakub välja soovitusel parendusteks ning süsteemi edasi arendamiseks.

8. Kasutatud kirjanduse allikad

Kirjanduse allikatena kasutan:

- Teaduspublikatsioone,
- Tehniliste seadmete tootjate kodulehti

9. Lõputöö konsultandid

10. Töö etapid ja ajakava

- Teema teoreetilise taustaga tutvumine ja lähteandmete kogumine (14.02.2020)
- Lõputöö ülesanne esitatud (03.03.2020)
- Juhtsüsteemi riistvara komponendid välja valitud (13.03.2020)
- Juhtsüsteemi riistvara valmis (30.03.2020)
- Juhtsüsteemi tarkvara valmis (19.04.2020)
- Valmis süsteemi testimine ja mõõtmiste tegemine (20.04- 26.04.2020)
- Töö kirjaliku osa esimene versioon valmis ja selle juhendale läbilugemiseks saatmine (03.05.2020)
- Paranduste sisseviimine ja töö juhendajale teiseks läbilugemiseks saatmine (11.05.2020)
- Töö lõplik versioon valmis (18.05.2020)

SISUKORD

LÕPUTÖÖ LÜHIKOKKUVÕTE.....	3
ABSTRACT.....	4
LÕPUTÖÖ ÜLESANNE	5
EESSÕNA.....	10
LÜHENDITE JA TÄHISTE LOETELU	11
SISSEJUHATUS	12
1. TAUSTINFO KOGUMINE JA TEHNILISTE TINGIMUSTE TÄPSUSTAMINE.....	13
1.1 Autori vaatlused.....	13
1.2 3D-printimisel eralduvad osakesed	13
1.3 FDM.....	15
1.4 FDM 3D printimisel kasutatavad materjalid	16
1.5 Detailide kaardumine.....	17
1.6 Lähteandmed ja nõuded lahendusele	19
1.6.1 Kasutatavad 3D printerid ja vajalik õhuvoolu hulk	19
1.6.2 Kaitseümbris ja soovitud funktsionaalsus	20
1.7 Olemasolevad lahendused.....	21
2. PROJEKTEERIMINE.....	24
2.1 Tehnoloogilise lahenduse üldine kirjeldus.....	24
2.2 Komponentide valik	24
2.3 Elektriskeem.....	25
2.4 Elektroonika korpused	28
2.5 Tarkvara ja kasutajaliides.....	29
3. TEOSTUS	32
3.1 Elektroonika korpuste tootmine	34

3.2 Elektroonika ja korpuse sisese juhtmestuse valmistamine	35
3.2.1 Elektroonika ja juhtpaneel	35
3.2.2 Kaitseümbrise ja juhtpaneeli juhtmestus.....	36
4. KASUTUSELE VÕTMINE JA TESTIMINE	38
4.1 Temperatuuri mõõtmised.....	39
4.1.1 Mõõtmine 1.....	39
4.1.2 Mõõtmine 2.....	39
4.1.3 Mõõtmine 3.....	40
4.2 Mõõtmistulemuste analüüs.....	41
4.2.1 Temperatuuri tagamine kaitseümbrises	41
4.2.2 Ruumi õhukvaliteedi tagamine	43
4.2.3 Juhtsüsteemi täiendamine ja edasiarendamine	44
KOKKUVÕTE	45
SUMMARY	47
KASUTATUD KIRJANDUS.....	49
LISAD	52
Lisa 1 Pildid detailide kaardumisest	53
Lisa 2 Kasutatud komponentide nimekiri.....	55
Lisa 3 Juhtsüsteemi elektriskeem.....	57
Lisa 4 Juhtsüsteemi programmi voodiagramm	58
Lisa 5 Juhtsüsteemi programmi väljatrükk.....	59

EESSÕNA

Antud bakalaureusetöö sõnastati autori, Sten Erik Aunpu, algatusel autori tööandja, 3DKoda OÜ, poolt antud projekti põhjal. Ettevõtte poolseks konsultandiks oli Priit Norak, kes on ettevõttes üks asutajatest. 3DKoda OÜ käest saadi arendustöö lähteülesanne ja algandmed. Samuti andis tööandja autorile vajalikud ruumid, töövahendid ja majanduslikud ressursid bakalaureusetöö käigus läbi viidud arendustöö tegemiseks ning lahenduse realiseerimiseks. Autor soovib tänu avaldada Tarmo Korõtkole kelle juhendamisel lõputöö valmis.

LÜHENDITE JA TÄHISTE LOETELU

ABS	Akrüülnitriilbutadieenstüreen (ing k. <i>Acrylonitrile Butadiene Styrene</i> , ABS)
CAD	Arvuti abil projekteerimine (ingl k. <i>Computer-Aided Desing</i> , CAD)
FDM	Sulatatud materjali sadestamise (ing k. <i>Fused Deposition Modelling</i> , FDM)
FFF	Sulatatud materjali sadestamise (ing k. <i>Fused Filament Fabrication</i> , FFF)
HEPA	Kõrge efektiivsusega osakeste neeldumine(ing k. <i>High-Efficiency Particulate Absorbing</i> , HEPA)
HIPS	Löögikindel polüstüreen (ing k. <i>High Impact Polystyrene</i> , HIPS)
PC	Polükarbonaat (ing k. <i>Polycarbonate</i> , PC)
PID	Proportsionaal-integraal-diferentsiaal (ing k. <i>Proportional–Integral–Derivative</i> , PID)
PLA	Polüpiimhape (ing k. <i>Polylactic Acid</i> , PLA)
ppm	Osakest miljoni kohta (ing k. <i>particles per million</i> , ppm)
PWM	Pulsilaiusmodulatsioon (ing k. <i>Pulse-width modulation</i> , PWM)
VOC	Lenduvad orgaanilised ained (ing k. <i>Volatile Organic Components</i> , VOC)

SISSEJUHATUS

Antud lõputöö kirjeldab 3D-printerite kaitseümbrise juhtsüsteemi arendamise protsessi ning kasutusele võtmist. Arendatava juhtsüsteemi ülesandeks on hoida 3D-printerite ümber stabiilset temperatuuri ning tagada parem õhukvaliteet 3D-printerite kasutajale. Lõputöös tuuakse välja, miks on antud lahenduse arendamine vajalik ja oluline, kirjeldatakse mida kujutab endas sulatatud materjali sadestamise (ing. k. fused deposition modelling e. FDM) tehnoloogia, mis plastmasse peamiselt kasutatakse selle tehnoloogiaga, mis on nende materjalide 3D-printimise kontekstis olulised omadused, uuritakse olemasolevate lahenduse sobilikust ja kirjeldatakse probleemi lahenduseks valmistava prototüüp seadme projekteerimise ja valmis ehitamise protsessi.

Lõputöö teema valikul sai peamiselt lähtunud autorile talle tema tööandja, 3DKoda OÜ, poolt antud tööülesannetest. 3DKoda OÜ tegeleb igapäevaselt toodete kiirprototüüpimise teenuse pakkumisega, kasutades selleks erinevaid 3D-printimise tehnoloogiaid. Lõputöö käigus valminud seadme eesmärk on vähendada praagi tekkimise hulka tootmises ning õhukvaliteedi normide täitmise abil tagada töötajatele ohutu keskkond ning vältida hilisemate terviseprobleemide tekkimist, mis võivad 3D-printeritega regulaarsel kokkupuutel tekkida.

3DKoda OÜs tehakse peamiselt tööd FDM tehnoloogiat kasutavate 3D-printeritega. Selle tehnoloogia kasutamisel tekkivad mikroosakesed, mis paiskuvad ümbritsevasse õhku. Need osakesed võivad olla mürgised ning tekitada inimestel terviseprobleeme, sest sisse hingatava õhu kaudu sisenevad ning ladestuvad inimese organismi. Samuti on FDM tehnoloogiaga printimine temperatuuritundlik ja vajab kvaliteetsete detaili valmistamiseks stabiilset õhutemperatuuri ligikaudu 35 °C.

Turul kättesaadavaid 3D-printerite nutikate kaitseümbriste lahendusi iseloomustab suhteliselt kõrge hind ning mahutavus ühele printerile. Odavad lahendused on oma võimekuse ja funktsionaalsuse poolest limiteeritud. Enamik lahendusi on mõeldud ka ainult kindlatele 3D-printerite mudelitele.

Lõputöö on jaotatud neljaks peatükiks, mis kajastavad arendatud lahenduse jaoks olulist taustinfot, projekteerimist, teostust ning kasutusele võtmist. Juhtsüsteemi tarkvara väljatrükk, autori vaatluste dokumenteerimiseks tehtud pildid ja elektroonika komponentide nimekiri tuuakse välja lisade all.

Antud lõputöö tegemisel kasutati CAD tarkvarasid Autodesk Inventor Professional 2019 ja Zortrax Z-Suite. Programmi kirjutamiseks tarkvara Arduino IDE ning selle algoritmi koostamiseks ArgoUMLi. Elektriskeemid koostati Altium CircuitMakeriga. Mõõteandmete töötlemiseks kasutati Microsoft Excel tarkvara.

1. TAUSTINFO KOGUMINE JA TEHNILISTE TINGIMUSTE

TÄPSUSTAMINE

1.1 Autori vaatlused

Autori vaatlustel 3D-printerite töötamisele selgus, et 3D-printeritega samas ruumis viibides on masinate töötamise ajal tugev plastmass sulamise/põlemise lõhn ning viibides ruumis üle veerand tunni tekib inimesel hingamisraskused, kehv enesetunne ning ka tugev peavalu. Proovides neid probleeme likvideerida, näiteks avades tuulutamiseks ruumi ukse, halvenes jälle printimiste kvaliteet, sest ruumi sisenev õhk oli piisavalt jahe, et jahutada peale ruumi maha ka 3D-printerite komponente. Komponentide maha jahtumine soodustab prinditavate detailide kaardumist, mis teeb need kasutuskõlbmatuks. Detailide kaardumise dokumenteerimiseks tehtud pildid on lisas 1.

Autor ise ning teised ettevõtte töötajad on 3D-printimisel tekkiva tugeva lõhnaga ära harjunud ning ei pane seda enam tähele aga jätkuvalt esineb neil peavalusid ja halba enesetunnet. Eelnevalt autori poolt renditud lahenduse kasutusele võtmist köeti tuba, kus 3D-printeritd asuvad, umbes 30 °C juurde, et vähendada kaardumist. See tegi selles ruumis pikaajalise viibimise ebameeldivaks ning aitas võimendada 3D-printeritelt lenduvate mikroosakeste poolt põhjustatud halba enesetunnet.

1.2 3D-printimisel eralduvad osakesed

Vaatamata 3D-printerite populaarsuse kasvule ei ole paljud 3D-printerite kasutajad teadlikud sellest, et need eritavad õhku mikroosakesi ning kui kasutajad seda teavadki ei olda kursis, mida täpselt need aerosoolid endas kujutavad ning kas ja kuidas need inimeste tervist võivad mõjutada. 3D-printimisel eralduvate aerosoolidega arvestamine on muutumas tähtsaks, sest 3D-printerid on aina rohkem kasutusel haridusasutustes, avalikusele avatud ruumides või eraruumides, kus nende masinate ja nende potentsiaalselt mürgiste mikroosakestega võivad puutuda kokku lapsed. [1]

Illinois Tehnikaülikooli ja Texase Ülikooli Tsiivil-, arhitektuuri- ja keskkonnatehnika osakonda koostöös viidi 2016. aastal läbi uuring, mis keskendus FDM printimisel õhku lenduvate mikroosakeste ja VOC-de uurimisele. 16 erineva printeri ja filamendi kombinatsiooni, 5 erinevat jaemüügis kättesaadavat printerit ja 9 erinevat filamenti, testimisel leiti, et 3D-printimine tõstab suurel määral õhus olevate mikroosakeste arvu ning ka VOC-de kontsentratsiooni. Mikroosakeste eraldumine varjeerus 10^8 kuni 10^{11} osakest minuti jooksul sõltuvalt materjali ja 3D-printeri kombinatsioonist. Mikroosake eraldumine oli kõige suurem 3D-printimise esimesel 10-20 minutil ning siis langes tasemele, mis jäi stabiilseks printimise lõpuni. VOC-de mõõtmisel leiti, et ABS ja HIPS

materjali korral oli peamine VOC, mis eraldus, stüreen (10 kuni 110 $\mu\text{g}/\text{min}$), nailoni 3D-printimisel eraldus peamiselt kaprolaktaam (2 kuni 180 $\mu\text{g}/\text{min}$) ning PLA puhul laktiid (4 kuni 5 $\mu\text{g}/\text{min}$). Selles uuringus selgus, et PLA 3D-printimisel eraldab õhku kõige vähem mikroosakesi ning samas ka kõige vähem VOC-isid ja ABS materjali kasutamisel eraldub mõlemaid õhku kõige enam. [2]

Stüreen satub inimese organismi peamiselt hingamisteede kaudu ning vähesele määral imendub see ka läbi naha. Inimesel võib see regulaarsel kokkupuutel potentsiaalselt rasvkudedesse ladestuda. Stüreen tekitab inimestel silmade, kurgu ning hingamisteede ärritust alates kontsentratsioonist 43 mg/m^3 . Selle ärritusega kaasnevad peavalu, iiveldus ning hingamisraskused. Kokkupuude kontsentratsiooniga üle 426 mg/m^3 võib tekitada kopsupõletikku ning põhjustada inimestel astmat. Stüreeni peetakse ka kantserogeeniks, mis mõjutab inimese närvisüsteemi ja võib tekitada arenguhäireid. Suurendab immuunsüsteemiga seotud vähkide ja verevähi tekkimise tõenäosust. Eesti vabariigis kehtivate töökesekkonna keemiliste ohutegurite piirnormide järgi on stüreeni piirnorm 8 tunni jooksul 90 mg/m^3 . Maailma Terviseorganisatsiooni Euroopa haru aga soovib arvestades stüreeni potentsiaalseid pikaajalisi neurotoksilisi mõjusid, et regulaarsel kokkupuutel ei tohiks kontsentratsioon olla üle 0,26 mg/m^3 keskmiselt nädalas. [3] [4]

Kaprolaktaami sissehingamine võib põhjustada stüreeniga sarnaselt inimestel silmade ja hingamisteede ärritust. See ärritus võib omakorda tekitada inimestel peavalu ja halba enesetunnet. Erinevalt stüreenist ei peeta kaprolaktaami kantserogeenseks aineks. Eesti vabariigis kehtivate töökesekkonna keemiliste ohutegurite piirnormide järgi on kaprolaktaami piirnorm 8 tunni jooksul 5 mg/m^3 . [5] [4]

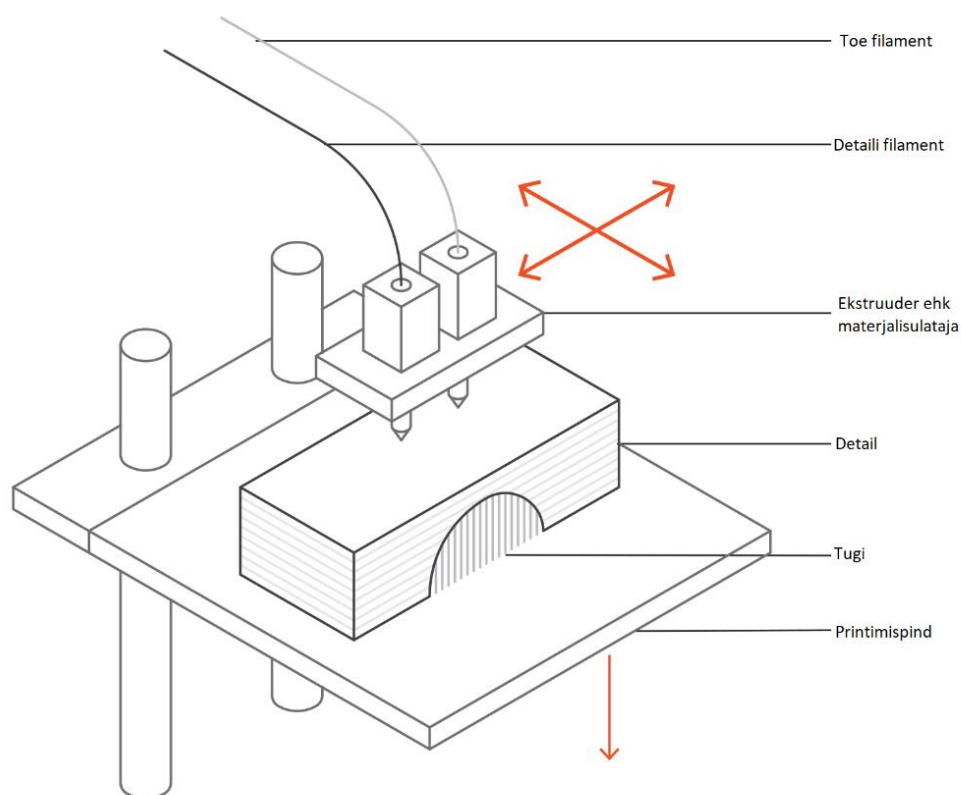
Laktiid on naturaalne looduslikult esinev ühend millel inimese tervisele negatiivseid mõjusid ei ole. Peale selle, et see on kasutusel PLA materjali tootmisel, mille käigus seda tagasi õhku paiskuv, on see kasutusel ka näiteks toidu säilitusainena. [6]

Kuigi on läbi viidud mitmeid uurimusi 3D-printerite töötamisel õhku lenduvate jääkide kohta, mis saavad sarnaseid lõpptulemusi, siis puudub nende vahel katsetuste ja andmete analüüsimise meetoodika ühtlus. Kõikides on ka kasutatud erinevaid mõõteseadmeid ning testkambrite ülesehitus ning parameetrid on erinevad. USA Atlanta Georgia Tehnika Instituudis viidi läbi uuring, kus uuriti kas Blue Angel meetodi, mis on kasutusel laserprinterite töö käigus eralduvate osakesete mõõtmiseks ning analüüsimiseks, oleks võimalik rakendada ka 3D-printerite puhul. Selle kasutamisel uuringu käigus kattusid mõõtetulemused teiste sarnaste uuringute tulemustega. Ainult osades PLA materjaliga seotud tulemustes esines märgatavaid erinevusi. Tulevikus 3D-printeritega seotud mõõtmiste teostamiseks soovivad teadustöö läbiviijad luua uue

standardiseeritud testimismeetodi, et oleks võimalik teha rohkem ja täpsemaid kvantitatiivseid võrdlusi erinevate selle teemaga seotud teadustööde vahel. Samuti tuuakse välja see, et kuna tahke materjali omadused ei anna head ülevaadet 3D-printerite heitmete eraldumise kohta oleks tulevikus mõistlik kõikide materjalide heitmeid esitada temperatuurist sõltuva funktsioonina ning oleks vaja luua uus testimine standard tööstuse jaoks, mis elimineeriks keskkonna tingimuste ja inimese poolt tekitatud tingimuste varieeruvuse. [1]

1.3 FDM

FDM ehk sulatatud materjali sadestamise tehnoloogia, kasutusel on ka lühend FFF (ing k. *Fused Filament Fabrication*, FFF), on tootmistehnoloogia, mis kujutab endas detaili loomist sulamaterjali kandmisel printimispinnale kihtkihi haaval mööda ettemääratud teekonda. Detail valmib, kui kõik vajalikud kihid on üksteise peale sadestatud, kihid omavahel kokku sulanud ja detail jahtudes kõvastunud. Sellise tehnoloogiaga detailide valmistamiseks kasutakse termoplastseid plastmasse, mida sisestatakse masinasse filamenti ehk ümmarguse niidi kujul. [7]



Joonis 1.1 Kahe ekstruuderiga FDM 3D-printeri ehitus [7].

FDM 3D-printimisega detailide tootmisel olulised seadme seadeparameetrid on ekstruuderi ehk materjalisulataja ja printimispinna temperatuur, printimise kiirus, jahutusventilaatori kiirus ning prinditavate kihtide paksus, mis jääb vahemikku 0,05 kuni 0,4 mm. [7]

Selle tehnoloogia plussideks on kuluefektiivsus, lihtne kättesaadavus ning lai valik materjale, mida on võimalik kasutada. Sellepärast on see hea tehnoloogia kiirprototüüpimise jaoks. FDM miinusteks on, et võrreldes teiste 3D-printimistehnoloogiatega on see ebatäpsem ning väga detailseid mudeleid ei ole võimalik kvaliteetselt realiseerida. Samuti ei ole enamasti võimalik ilma järeltöötluseta saavutada sujuvaid pindasid, sest kihid jäävad nähtavad. Kihiline ehitus teeb ka detailid anisotroopseteks ehk need ühes teljes tugevamad kui teised. [7]

1.4 FDM 3D printimisel kasutatavad materjalid

Materjalide valik FDM 3D-printimise puhul on lai. Kirjeldatud ja välja toodud on materjalid, mis leiavad kasutust autori töökohas.

Need materjalid on:

- **ABS** - Akrüülnitriilbutadienestüreen (ing k. *Acrylonitrile Butadiene Styrene*, ABS) on olnud 3D-printimiseks kasutusel pikka aega. See oli esimene materjal mida kasutati tööstuslikes 3D-printrites. See on jätkuvalt populaarne materjal oma odava hinna, heade mehaaniliste omaduste ning kõrge sulamistemperatuuri, umbes 220 °C, tõttu. Sellest tehtud detaile on materjali vastupidavuse tõttu võimalik väliskeskkonnas rakendada. ABS vajab printimisel temperatuuri stabiilsust, et vältida kaardumist, sest see tõmbub jahtumisel kokku. Miinuseks on ka see, et sellega printimisel tekib tugev ning ebameeldiv lõhn, mis teeb vajalikuks ventilatsiooni.[8]
- **HIPS** - Löögikindlat polüstüreeni (ing k. *High Impact Polystyrene*, HIPS) kasutatakse tihti selle kerguse, dimensioonilise stabiilsuse, see tõmbub jahtudes vähe kokku, ja ABS-iga sarnaste mehhaaniliste omaduste tõttu. HIPS-i kasutakse tihti ka tugimaterjaliks ABS-i 3D-printimisel, kui on kasutusel kahe eraldi ekstruuderiga 3D-printer. See sobib hästi ABS-i tugimaterjaliks, sest neil kahel materjalil on sarnased 3D-printimisomadused ning HIPS-i lahustumine limoneenis, teeb võimalikuks eemaldada toed ilma, et detaili pindadele jääks tugedest tingitud defekte. [9]
- **Nailon** – Polüamiid ehk nailon on plastitööstuses tuntud oma tugevuse ja painduvuse poolest. Selle materjali kasutamiseks 3D-printimiseks on vaja see kuumutada umbes 250 °C

juurde. Nailoni 3D-printimine on lõhnatu protsess. Nailoni kasutamisel 3D-printimisel on peamiseks probleemiks, et nailoni imab endasse niiskust ümbritsevast õhust. Niiske nailoniga 3D-printides prinditud detailide kvaliteet halveneb, sellepärast on tähtis nailoni korrektne hoiustamine. Sarnaselt ABS-iga võib see printimisel kaarduda. [10]

- **PC** - Polükarbonaat (ing k. *Polycarbonate*, PC) on suure tugevusega materjal, mis peab vastu ka rasketes keskkonna tingimustes. Seda on võimalik painutada ilma, et see murduks. Sarnaselt ABS-ile ja nailonile on sellel kõrge sulamistemperatuur ning soodumus 3D-printimisel kaarduda. PC nagu ka nailon imab endasse niiskust, mis alandab printimiste kvaliteeti ja toodetud detaili tugevust. [11]
- **PLA** - Polüpiimhape (ing k. *Polylactic Acid*, PLA) on üks populaarsemaid 3D-printimiseks kasutatavaid materjale ning see paljude 3D-printerite vaike printimismaterjaliks, sest seda saab printida madalatel temperatuuridel ning ei vaja sellepärast ka soojendatud printimispiinast. See teeb PLA heaks materjaliks millega õppida 3D-printimist kuna on odav ja lihtsasti prinditav. PLA on ka üks keskkonna sõbralikemaid turul olevaid 3D-printimismaterjale, sest seda on võimalik toota maisi ja suhkruroo saadustest. See teeb PLA taastuvaks materjaliks ja see on ka biolagunev. Sellega 3D-printimisel eraldub magus lõhn. PLA miinusteks on kehv vastupidavus kõrgetele temperatuuridele ning päikesevalgusele ja peale printimist jääb see rabadaks ehk võib kergelt katki minna. See kipub ka printimisel ilma jahutusventilaatorita laiali valguma, mis võib printimise kvaliteeti alandada. [12]

3DKoda OÜs täidetakse enamik tellimused kasutades Zortraxi poolt toodetud ABS-i põhise filamentiga nimega Z-ULTRAT. Selle koostisest on 90-100 % ABS, 0-3 % PC ja 0-10 % värvaineid või muud lisandid. Ettevõtte kogemused on näidanud, et see kaardub vähem kui puhas ABS. [13]

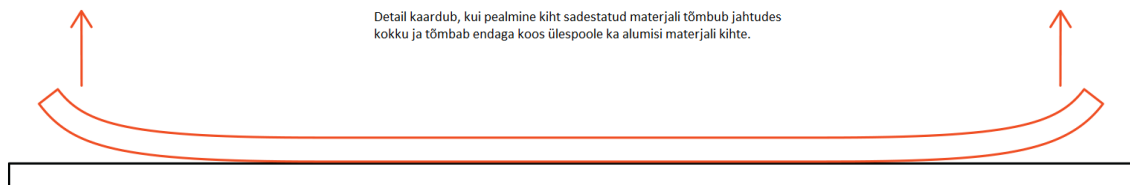
1.5 Detailide kaardumine

Kaardumine on kõige levinum defekt FDM tehnoloogiaga 3D-printimisel, mis tuleneb materjalide kokku tõmbumisest jahtumisel. Ebaühtlase jahtumise korral tekivad materjali sisepinged, mis tõmbavad materjali kas printimispiina või alumiste prinditud kihtide küljest lahti põhjustades nii detaili kaardumist. [7]

Kaardumise põhjusteks võivad olla:

- **Suured lamedad pinnad** – Ebaühtlane jahtumine [7]

- **Detailid õhukesed välja ulatuvad osad [7]**
- **Teravad nurgad – tuleks kasutada kõikjal raadiusi [7]**
- **Kasutatud materjalide omadused, mis seda soodustavad [7]**



Joonis 1.2 Detaili kaardumine FDM 3D-printimisel [7].

Kaardumise kui ühe peamise probleemi 3D-printimisel toovad välja ka printerite tootjad ise, nagu näiteks Prusa, Ultimaker ja Zortrax. Zortraxi M200, M200plus, M300 ja M300+ mudel 3D-printerid on autori tööandja poolt kasutusel. Esimese kaardumise põhjusena toovad eelmainitud 3D-printerite tootjad välja liiga madalate temperatuuridega printimise ehk kas printimispinna või materjali ekstruuderi temperatuur ei ole piisav. Liiga madal temperatuur tekitab detaili ebaühtlase või liiga kiire jahtumise, mis omakorda tekitab materjali sees kaardumist põhjustavaid sisepingeid. Temperatuuride ebasobivus võib olla nii tingitud välisest keskkonnast kui ka kasutajapoolsest 3D-printeri valesti seadistamisest printimise jaoks. [14] [15] [16]

Zortrax soovib ühe lahendusena näiteks ABS-iga 3D-printimisel teha printimispind enne printimise algust kokku atsetooni ja ABS plastmassi seguga, ABS plastmass lahustub atsetoonis, mille abil detail peaks printimispinna külge paremini kinni jääma. Prusa soovib ka lisada detailide pärast eemaldatavat lisamaterjali kohtadesse, kus detailil võib kaardumist tekkida. Ultimaker pakub välja ka eksperimentaalse võimaluse printida detaili ümber kaitsesein, mis ei lase jahedat õhku detailile ligi. Zortraxi programm, mis valmistab 3D-mudeli ette 3D-printimiseks, võimaldab kogu detaili alla lisada nii õelda lisamaterjalist parve, mille peale kogu detail printitakse ning see aitab detailil printimispinnale paremini kinni jääda ja samas tehes selle eemaldamise sealt pärast lihtsamaks. Kuigi ettevõtte kasutab ABS-i ja atsetooni segu ning kõik 3D-printitavad detailid printitakse parvedele on kaardumine pidev probleem. [14] [15] [16]

Ettevõttel on vaja lahendust, mis suudaks detailide kaardumist vähendada stabiilse keskkonna temperatuuri hoidmise abil. Samas peaks see seade olema võimeline tagama töötajatele ohutuma töökeskkonna terviseprobleemide tekkimise vältimiseks.

1.6 Lähteandmed ja nõuded lahendusele

Juhtsüsteemi arendamisel olulised aspektid millest lähtuda tuli olid kasutusel olevate 3D-printerite omadused, olemasolev kaitseümbris, mida hakatakse juhtsüsteemiga nutikaks muutma, ning ka 3DKoda OÜ nõuded ja soovid arendatavale lahendusele.

1.6.1 Kasutatavad 3D printerid ja vajalik õhuvoolu hulk

Ettevõttes on igapäevaselt kasutusel 5 Zortraxi FDM 3D-printerit. Kolm neist printeritest on Zortrax M200 mudelit, mis on tööalaga 200 x 200 x 200 mm, ülejäänud kaks on suurema tööalaga, 300 x 300 x 300 mm, M300 ja M300plus mudel 3D-printerid. Need on kõik pool suletud printerid ehk on pealt poolt avatud. Kaitseümbrises temperatuuri hoidmiseks vajaliku õhuvoolu arvutamiseks sai oletatud, et pool 3D-printerite poolt tarbitud elektrist muudetakse soojuseks. Aluseks võeti tootja poolt antud 3D-printerite maksimaalne elektri tarbimine. See oletus oli vajalik, sest teada on ainult materjali sulatava elemendi soojuslikvõimsus, milleks on 40 W ja vaja oli ka arvestada kuidagi muudelt komponentidelt eralduva soojusega. M200 mudel 3D-printerite maksimaalne elektrivõimsuse tarbimine on 200 W, M300 mudelil on 320 W ning M300plus mudelil 360 W. Õhuvoolu hulga arvutustel on toatemperatuuriks võetud $t_v = 22$ °C ning kaitseümbrise soovitud sisetemperatuuriks $t_s = 30$ °C. [17] [18]

Õhuvoolu hulga arvutamise valem [19]:

$$q = \frac{H}{\rho \cdot c \cdot (t_s - t_v)} \quad (1.1)$$

kus q – vajalik õhuvooluhulk, m³/s,

H – soojuslikvõimsus, mis eraldub printeritelt, W,

ρ – õhu tihendus, kg/m³,

c – õhu erisoojus, J/kg·K,

t_v – kaitseümbris väline temperatuur, °C,

t_s – soovitud kaitseümbrise sisene temperatuur, °C.

Kasutades ära valemit 1.1 ning võttes õhu tiheduse väärtuseks $\rho = 1,2 \text{ kg/ m}^3$ ning õhu erisoojuse väärtuseks $c = 1005 \text{ J/kg}\cdot\text{K}$ arvutati vajalik õhuvoolu hulk q . [19]

$$q = \frac{(3 \cdot 200 + 320 + 360) \cdot 0,5}{1,2 \cdot 1005 \cdot (30 - 22)} \approx 0,0663 \left[\frac{\text{m}^3}{\text{s}} \right] \approx 238,81 \left[\frac{\text{m}^3}{\text{h}} \right]$$

1.6.2 Kaitseümbris ja soovitud funktsionaalsus

Kaitseümbris, mida hakati juhtsüsteemiga nutikaks muutma, on eelnevalt valmistatud 3D Koda OÜ töötajate poolt. Puidust kaitseümbris on ehitatud metall raamiga kontoririivi sisse ning sellele seintesse olid valmis tehtud vajalikud avad ventilaatorite jaoks. Olid olemas ka filtrid mille abil kaitseümbrisest välja tõmmatavat õhku puhastada. Filtriteks on integreeritud tolmufiltriga 3M AXP3 gaasifiltrid. Samuti on kaitseümbrise siseste seadmete toitmiseks olemas kaheksa 230 VAC pistikut. Kaitseümbris ei ole õhukindel. Kaitseümbrise sisemised mõõtmed on 925x740x2777 mm.



Joonis 1.3 Kaitseümbris ilma usteta.

Kaitseümbrise juhtsüsteemi funktsioonid saab jaotada sõltuvalt nende olulisusest kaheks – primaarsed ja sekundaarsed.

Primaarsed funktsioonid:

- Kaitseümbrise sees ühtlase temperatuuri hoidmine
- Võimalus reguleerida hoitava temperatuuri väärtust
- Erinevate ventilatsiooni töörežiimide olemasolu
- Kasutajaliides juhtsüsteemi kontrollimiseks ja keskkonna tingimuste kuvamiseks

Sekundaarsed:

- Anduritest saadud väärtuste salvestamise võimalus
- Kaitseümbrise valgustuse reguleerimine
- Suitsu tuvastamine ja alarm, juhul kui see tuvastatakse

1.7 Olemasolevad lahendused

Võrreldud on kolme erinevat turul kättesaadavat lahendust. Probleemi ja lähteandmete põhjal valitud võrdlemisel arvesse võetavad parameetrid on hind, universaalsus, funktsionaalsus ning printerite mahutavus.

Esimene olemasoleva lahendus mille sobivust sai analüüsitud oli Zortrax HEPA Cover, mis mõeldud spetsiaalselt firmas kasutatavale M200 ja M300 seeria mudel 3D-printeritele. Antud seadme hind on tootja kodulehel väiksemate mudelite, M200 seeria, jaoks 200 € ning suuremate printerite, M300 seeria jaoks 250 33€. Seade kujutab endas plastmassist katet, mida on võimalik panna 3D-printeri otsa. Selle kate küljes on filtreerimismoodul, mis koosneb reguleeritava kiirusega ventilaatorist ning HEPA ehk kõrge efektiivsusega osakeste neeldumise (ing k. *High-Efficiency Particulate Absorbing*, HEPA) ja söe filtrist. Puuduseks sellel seadmel on, et kuigi see on mõeldud just kasutuseks olemasolevate printeritele siis mõnel printeril, mis on ettevõttes mitu aastat kasutusel olnud, kukuvad uksest eest ära kasutamisest tingitud kulumise tõttu ning ukseid ei püsi kinni ka osadel vähem kasutatud masinatel. Nende printerite puhul ei oleks sellest seadmest kasu. Nende seadmete kasutusele võtmine tuleks ka küllaltki kulukas, sest igale printerile on vaja osta eraldi seade ning neid ei oleks võimalik tulevikus taaskasutada kui peaks toimuma 3D-printerite masinapargi uuendus 3D-printerite vastu millega need kokku ei sobi. Selle toote funktsionaalsust ei ole ka võimalik lihtsalt laiendada. [20] [21]



Joonis 1.4 Zortrax M200 seeria HEPA Cover [20].

Eraettevõtte 3DPrintClean pakub kaitseümbrise terviklikku lahendust koos juhtsüsteemiga. Nende tootel 870 Safety Enclosure Pro on olemas kogu vajalik funktsionaalsus, mis on lahendusele määratud nõudmiste all välja toodud. Sellel seadmel on olemas temperatuuri ühtlaseks hoidmiseks ning õhu filtratsiooniks vajalik ventilatsiooni lahendus. Samuti on olemas valgustus ning võimalus üle WiFi saada infot kaitseümbrise sisekeskkonna kohta. Seadmesse mahuvad paljud 3D-printerite mudelid erinevatel tootjatelt, seal hulgas autori tööandja poolt hetkel kasutusel olevad. Tootja kodulehel on seadme hinnaks 2729 €, mis on ka selle seadme üheks peamiseks miinuseks, sest see mahutab korraga ainult ühe printeri. See kaitseümbris maksab sama palju või rohkem kui paljud 3D-printerid.[22]



Joonis 1.5 3DPrintClean 870 Safety Enclosure Pro [22].

Olemas on ka universaalne juhtimissüsteemi lahendus DIY Smart 3D Printer Enclosure KIT, mille saab ise integreerida olemas olevasse kaitseümbrisesse. See on välja arendatud Sloveenia 3D-printimise entusiasti Jure Korber poolt ning see on 40 € eest saadaval. See on funktsionaalsuse poolest küllaltki limiteeritud, ainult ventilaatori kiiruse juhtimine, tulekahju tuvastamine ja õhuniiskuse ning temperatuuri mõõtmine. Kirjelduse järgi on see ka pigem mõeldud kasutama kaitseümbrisega, kus on ainult üks 3D-printer. [23]



Joonis 1.6 DIY Smart 3D Printer Enclosure KIT [23].

Tabel 1.1.1 Olemasolevate lahenduste võrdlus.

Lahendus/Parameeter	Universaalsus	Funktsionaalsus	Mahutavus	Hind [€]
Zortrax HEPA Cover	Ainult Zortraxi 3D-printerid	Õhu filtreerimine	1 printer	200-250
3DPrintClean 870 Safety Enclosure Pro	Kõik 3D-printerid, mis mõõtudelt sobivad	Õhu filtreerimine, temperatuuri hoidmine, valgustus, läbi kasutajaliidese reguleeritav, õhu kvaliteedi mõõtmine	1 printer	2729
DIY Smart 3D Printer Enclosure KIT	Kõik 3D-printerid	Temperatuuri mõõtmine, ventilaatorite kiiruse reguleerimine	1 printer	40

Võttes arvesse, et olemasolevad lahendused pole majanduslikku mõistlikud või nende omadused on puudulikud, näeb autor, et on võimalik projekteerida ja valmis ehitada ettevõttesiseseks kasutamiseks universaalne juhtsüsteem, mis võimaldaks ära kasutada olemasolevat kaitseümbrist, materjale ja töövahendeid hoides nii rahaliselt kokku. Juhtsüsteem oleks võimalik projekteerida nii, et selle funktsionaalsust oleks tulevikus vajaduse korral lihtne lisaseadmete abil laiendada.

2. PROJEKTEERIMINE

2.1 Tehnoloogilise lahenduse üldine kirjeldus

Juhtsüsteemi primaarseks funktsiooniks on kaitseümbrise õhuvahetuse juhtimine ühtlase sisetemperatuuri tagamiseks. Õhuvahetuse teostamiseks kasutatakse nelja ventilaatorit mille kiirust kontroller reguleerib kasutades pulsilaiusmodulatsiooni (ing k. *Pulse-width modulation*, PWM). PWM väärtuse saab määrata kasutaja ise või arvutab selle välja tarkvaraline proportsionaal-integraal-diferentsiaal (ing k. *Proportional-Integral-Derivative*, PID) regulaator, mis töötab vastavalt kaitseümbrise sisetemperatuuri aritmeetilisele keskmisele väärtusele. PID regulaator on realiseeritud valmis PID regulaatori teegi abil. PID regulaatori komponentide võimendusteguriteks on määratud $K_p = 2$, $K_i = 5$ ja $K_d = 1$, lähtudes regulaatori teegi näidisprogrammidest. Muude PID regulaatori seadeparameetritena kasutati selle teegi vaikeväärtusi neile parameetritele. PID regulaator seadistati töötama režiimis, kus sisendi, keskmise temperatuuri, kasvamine põhjustab väljundi, ventilaatorite PWMi täiteteguri, kasvamist ehk väljundi suurendamisega proovitakse sisendi väärtust madaldada. Suurim lubatud erinevus soovitud seadepunktist on ± 2 °C. Temperatuuri mõõdavad kuus lae alla paigutatud temperatuuri andurit. Kasutaja saab määrata PID-kontrolleri seadepunkti, temperatuuri mida süsteem proovib hoida, vahemikus 20 kuni 40°C. Alumisest piirist madalam temperatuur põhjustaks kaardumist ning ülemisest piirist kõrgem temperatuur võib potentsiaalselt kahjustada 3D-printeri elektroonikat.

Lisaks temperatuuri anduritele on süsteemiga ühendatud ka kaks õhukvaliteediandurit. Üks anduritest mõõdab kaitseümbrise sees olevate VOC-de kontsentratsiooni õhus ja teine andur mõõdab mikroosakeste kogust õhus. Samuti on lisatud andur, mis on võimeline tuvastama suitsu olemasolu. Suitsu tuvastamisel antakse sellest helisignaali märku.

Kasutaja saab mõjutada juhtsüsteemi tööd 3,5-tollisele ekraanile kuvatud kasutajaliidese ja juhtkangi vahendusel. Samuti kuvatakse kasutajaliidese abil kasutajale temperatuuriandurite näidu ning näitude keskmine väärtus ja ka õhukvaliteedi andurite mõõtetulemused. Läbi kasutajaliidese on võimalik lisaks kontrollida kaitseümbrise sisese valgustuse tooni ja valgustugevust.

2.2 Komponentide valik

Juhtsüsteemi komponentide valikul lähtuti nende hinnast, tarneajast, kättesaadavust ning omavahelisest sobivusest. Komponenti telliti Eesti ja Euroopa e-poodidest, sest see tagas neile kõige lühemad tarneajad ning komponentide hind ja nende transporditasud olid piisavalt madalad,

et püsida projekti eelarve piirides. Juhtsüsteemi koostamiseks vajalike komponentide eelarveks oli ettevõtte poolt määratud 300 €. Kokku kulus vajalike asjade ostmiseks 243,16 €. Lisas 2 olevas tabelis on välja toodud juhtsüsteemi prototüübi valmistamiseks kulunud komponentide nimekiri koos hindade ning kogustega. See tabel ei sisalda juba ettevõttel olemas olnud materjale nagu näiteks, filament, jootetina, jootepasta, isoleerteip, takistid, lülitid, juhtmed jms. Enne lõplike valikute tegemist sai uuritud, kas komponendid sobivad omavahel kokku ning kas nende kohta on olemas internetis abiinfot, et probleemide esinemisel oleks need võimalik kiiresti lahendada.

2.3 Elektriskeem

Juhtsüsteemi kontrollseadmeks valiti Arduino MEGA 2560 rev3. Sellel seadmel on olemas lahenduse realiseerimiseks piisav arv sisendeid ja väljundeid, 54 digitaal sisendit/väljundit ning 16 analoog sisendit. 15 digitaalset sisendit/väljundit on võimelised välja andma PWM signaali, mis on vajalik ventilaatorite ja valgustuse juhtimiseks. Seade toetab ka erinevaid jada andmesideliideseid, mida on vaja andurite ühendamiseks. Valiti see kontrollseade, sest seda kasutavate lahenduste kohta on saadaval palju vabalt kättesaadavat infot ning autoril endal on selle kasutamisega eelnev kogemus. Süsteemi toitmiseks kasutatakse Mean Well LRS-75-12 toiteploki, mis muundab 230 V ühefaasilise vahelduvvoolu 12 V alalisvooluks. Antud toiteploki võimsus on 75 W, mis on piisav, et katta kogu juhtsüsteemi energiatarbimine. Ventilaatorite ja valgustuse maksimaalne energia tarbimine on kokku umbes 48 W ning ülejäänud elektroonika 8 W. Selle pingega peal töötavad süsteemis olevad ventilaatorid ja kaitseümbrise sisene valgustus. Juhtsüsteemi elektriskeem on nähtav lisas 3.

Süsteemiga ühendatud andurid, juhtkang ja muud lisaseadmed ei ole võimelised töötama toitepingega 12 V. Sellepärast on lisatud toiteahelasse reguleeritav alalisvoolu pinget madaldamise moodul, mille abil muudetakse 12 V neile seadmetele sobivaks 5 V toitepingeks. Need seadmed on eraldi toite peale, et mitte üle koormata Arduino 5 V toiteahelat, mida oleks saanud väiksema seadmete arvu korral kasutada. Samuti on lisatud toiteahelasse teine samasugune moodul Arduino MEGA toitmiseks, sest toites Arduino 12 V pingega ei ole sisse ehitatud lineaarne 5 V pingeregulaator suuteline ära toitma ekraani, mis saab toite otse Arduinolt. See tekitas ekraani pildi värelemist ning juhtsüsteem töötanud korrektselt. Selle probleemi lahendamiseks on reguleeritud Arduino toide 7 V peale. See tagas Arduino 5 V regulaatori efektiivsema töö, sest see pidi pinget vähem alandama.

Ekraaniks kasutatakse Arduino MEGA 2560 rev3 mõeldud 3,5-tollist TFT LCD moodulit, mille resolutsiooniks on 320x480 pikslit. See on ühendatud Arduino pesadesse D22 kuni D41. Ekraan ei

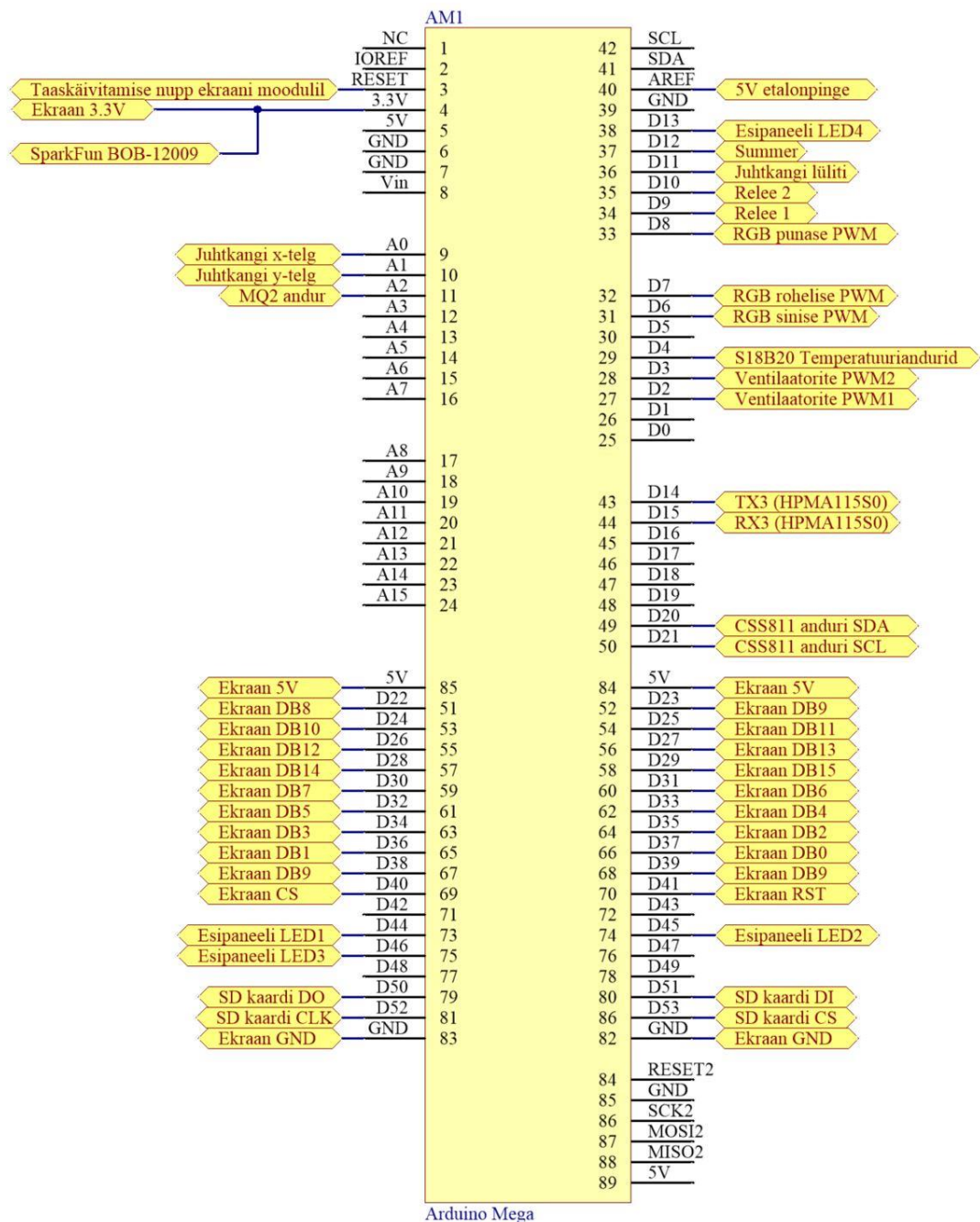
ole puuetundlik ning sellepärast kasutatakse 2-teljelist analoog juhtkangi süsteemile kasutajapoolsete sisendite andmiseks. Juhtkangi teljed on ühendatud analoog pesadesse A0 ja A1 ning selle lüliti pesasse D11.

Ventilaatorite toite sisse ja välja lülitamiseks kasutatakse kahte 5V releed, üks ventilaator lülitab kahe ventilaatori toidet. Arduino digitaalsete sisendite/väljundite annavad välja maksimaalselt 40mA voolu, mis ei ole piisab releede otse lülitamiseks. Sellepärast kasutatakse rele lülituste sooritamiseks npn-bipolaartransistor transistorid. Releede mähistega on rööbiti ühendatud kaitsedioidid, et kaitsta ülejäänud ahelat voolu järsu kasvamise eest, mis võib esineda releede lülitamisel. Releede lülitamist juhitakse Arduino pesade D9 ja D10 kaudu. [24]

Õhuvahetuse jaoks kasutatakse nelja Cryorig QF140 Performance, originaalset arvutite jaoks mõeldud, ventilaatorit. Maksimaalsel kiirusel on üks ventilaator võimeline liigutama tunnis 217 m³ õhku ehk juba üks ventilaator oleks peaaegu piisav, et hoida eelmises peatükis välja toodud näite tingimustel kaitseümbrises ühtlast temperatuuri. Suuremat õhuvoolu vajavate olukordade katmiseks on neid kasutusel neli, kaks sissetõmbe ja kaks väljatõmbe jaoks. Ventilaatorite kiirust saab reguleerida PWM signaali abil, mis saadetakse sagedusega 25 kHz läbi Arduino D2 ja D3 pesade.[25]

LED valgustuse värvitooni juhtimiseks kasutatakse kolme PWM signaale, igale RGB alatoonile erinev signaal. Selleks, et 5 V peal töötav Arduino suudaks juhtida 12 V peal töötavat valgustust kasutatakse Darlingtoni konfiguratsioonis npn-bipolaartransistor transistorid ehk transistorid, mis koosnevad kahest transistorist, mis on ühendatud nii, et teine transistor võimendab juba esimese poolt võimendatud signaali. See teeb suuremate koormuste juhtimise võimalikuks sama signaali juures. Pulsilaiusmodulatsiooni signaalid värvitooni muutmiseks tulevad pesadest D6, D7 ja D8.

Temperatuurianduritena on kasutusel kuus DS18B20 digitaalse termomeetri valmis moodulit. Nende andurite mõõtevahemik on -55 °C kuni 125 °C. Nende andurite täpsus on ± 0,5 °C vahemikus -10 °C kuni 85 °C. Need andurid kasutavad 1-Wire andmeedastus liidest, mis võimaldab kasutada mitme anduri andmete edastamiseks ning vastuvõtmiseks ainult ühte signaalikandjat. Kõik temperatuuriandurid on ühendatud pesasse D4. [26]



Joonis 2.1 Arduino MEGA ühendused.

Kasutusel on 2 erinevat õhukvaliteedi andurite. VOC-de taseme mõõtmiseks on kasutusel Adafruit CCS811 moodul, mis kasutab sünkroon andmeside liidest. See on ühendatud pesadesse D20 ja D21. Õhus olevate osakeste arvu mõõtmiseks kasutatakse Honeywell HPM115S0-XXX andurit. Selle anduri signaali loogiga pinge on 3,3 V, mis ei sobi kokku valitud kontroll seadmega, mis kasutab loogiga nivood 5V. Sellepärast lisati anduri kasutamiseks ahelasse loogika pingemuundur. Sobilikul pinge nivool olev signaal antakse edasi Arduino pesadesse D14 ja D15.

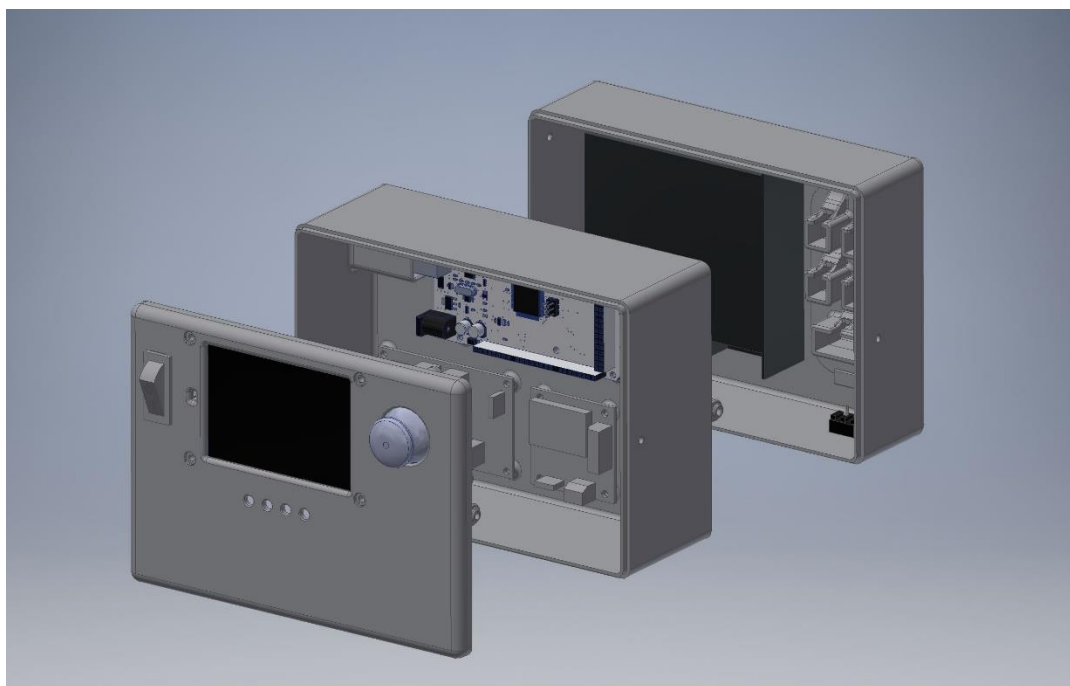
Suitsu tuvastamiseks on kasutusel MQ2 gaasi- ja suitsuandur, mille signaal on ühendatud pesasse A2. Juhul kui selle anduri pool tuvastatakse suitsu olemasolu antakse inimeste sellest teada helisignaali abil, mida tekitab summer. Summer on ühendatud pesasse D12.

Lisatud on ka Adafruiti SD-kaardi moodul, et oleks võimalik salvestada andurite mõõtetulemusi, mida saaks hiljem kasutaja analüüsida. See on ühendatud pesadesse D50 kuni D53.

2.4 Elektroonika korpused

Elektroonika korpuste modelleerimiseks kasutati tarkvara Autodesk Inventor Professional 2019. Korpuste modelleerimisel lähtuti ettevõttes olemasolevatest materjalidest ja tootmiseseadmetest. Nendeks olid 3D-printerid maksimaalse töötalaga kuni 300 x 300 x 300 mm ning ABSi põhine materjal Z-ULTRAT erinevates värvitoonides.

Modelleerida oli vaja korpused kõikidele juhtsüsteemiga ühendatud anduritele ning juhtimissüsteemi juhtpaneeli korpus mille esipaneeli kaudu kasutaja saab süsteemi tööd juhtida. Juhtpaneel on siseselt jaotatud kaheks sektsioonist. Ühes sektsioonis asub toiteplokk ja väljavahetatav pistikute kinnitus ning teises asub juhtsüsteemi elektroonika. Juhtpaneeli esipaneeli küljes asuvad 3,5-tolline ekraan, juhtkang, neli märguande LED-i ning lüliti 12 V toiteahela sisse ja välja lülitamiseks.



Joonis 2.2 Juhtpaneeli 3D-mudel koos elektroonikaga.

2.5 Tarkvara ja kasutajaliides

Juhtsüsteemi kontrolleri, Arduino MEGA 2560 rev3, programmeerimiseks kasutati Arduino IDE tarkvara. See on spetsiaalselt mõeldud Arduino tooteperekonda kuuluvate seadmete programmeerimiseks. Kasutati erinevaid valmis teeke, et lihtsustada ja kiirendada programmeerimise protsessi. Need lisati programmi Arduino IDE tarkvara vahendusel. Teekide kasutamisega oli võimalik põhiprogrammis välja kutsuda juba olemasolevaid funktsioone. 3D-printerite kaitseümbrise juhtsüsteemi programmi kirjutamisel kasutatud välised teegid olid:

- *LCDWIKI_KBV.h* – Ekraani riistvara haldamise eest vastutav teek. Vastutab registrite operatsiooni eest, andmete ja käskude edastamine eest, ekraani konfiguratsiooni eest, pikslite koordinaatide ja värvi seadete eest.
- *LCDWIKI_GUI.h* – Ekraanil kujundite ja tekstisümbolite näitamiseks vajalikke funktsioone sisaldav teek. Selleks teegi toimimiseks on vajalik riistvara teek *LCDWIKI_KBV.h*.
- *Adafruit_CCS811.h* – Adafruit CSS811 mooduli jaoks spetsiifiline teek. Selle teegi abil saab käivitada antud anduri ja küsida sellelt VOC-de kontsentratsiooni mõõtetulemust.
- *TimerThree.h* – Teek millega saab määrata ventilaatorite jaoks sobiliku PWMi sageduse. Sama teegi abil on võimalik muuta ka PWMi täitetegurit, et muuta ventilaatorite kiirust. See on seotud Arduino MEGA kolmanda taimeriga, mis ühendatud pesadesse D2, D3 ja D5.
- *OneWire.h* – 1-Wire protokolliga seadmete jaoks kasutatav teek. See teeb võimalikuks 1-Wire liini pealt lugeda lahenduses kasutatavate S18B20 moodulite poolt välja antavaid väärtusi.
- *DallasTemperature.h* – Teisendab 1-Wire liini pealt saadud väärtused reaalseteks temperatuuri väärtusteks. Annab kõikidele liinil olevatele anduritele indeksid mille järgi neid saab eristada ja hetkväärtusi neilt eraldi küsida.
- *PID_v1.h* – proportsionaal-integraal-diferentsiaalregulaatori ehk PID-kontrolleri teek, mis võimaldab lihtsalt ära määrata PID kontrolleri piirid ning võimendustegurite K_p , K_i ja K_d väärtused. On kasutatud selleks, et oleks võimalik luua ventilaatorite töörežiim, mis lähtub kaitseümbrise sisese temperatuuri keskmisest väärtusest ja kasutaja poolt määratud seadepunktist.

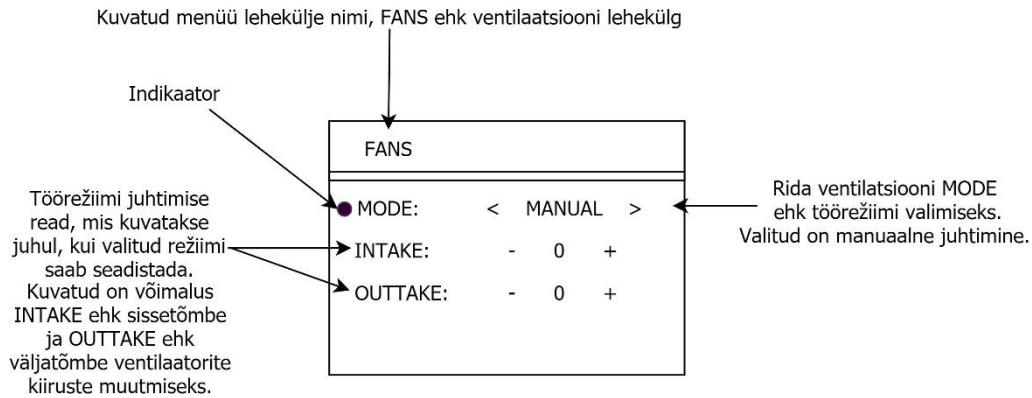
- *hpma115s0.h* – Honeywell HPM115S0-XXX mikroosakeste anduri töö juhtimiseks ning anduri poolt mõõdetavate väärtuste kättesaamiseks vajalik teek. Teek muudab ka mõõdetavad väärtused kohe loetavateks, et teegi kasutajal ei oleks vaja neid ümber teisendada.

Enne tervikliku programmi loomist testiti eraldi iga süsteemis kasutusel oleva anduri ja seadme toimimiseks vajalike teeke ja funktsioone, et olla kindel seadmete töökorras ning vähendada vajadust tegeleda probleemide lahendamise ja juhtsüsteemi tervikliku programmi koostamisega. Peale koodi osade eraldi testimist ja probleemide likvideerimist integreeriti need omavahel, et luua reaalne ja kasutatav juhtsüsteem. Juhtsüsteemi programmi voodiagramm on lisas 4 ja selle väljatrükk lisas 5.

Juhtsüsteemi programmi alguses lisatakse programmi kõik eelnevalt välja toodud välised teegid, defineeritakse kasutatavad muutujad ning määratakse neile väärtused. Järgmisena seadistatakse ära kontrolleri digitaalsed kontaktid sisenditeks ja väljunditeks. Määratakse ära need funktsioonid, mis kutsutakse esile ühekordselt juhtsüsteemi käivitamisel. Samuti määratakse ära, mis funktsioonid ning millises järjekorras tsükliiselt välja kutsutakse. Programmi tsükli alguses loetakse sisse juhtkangilt saadud sisendid ning andurite näidud, mille põhjal toimub ülejäänud funktsioonide esile kutsumine ja juhtsüsteemi talitlemine.

Juhtsüsteemi käivitamisel kuvatakse algselt kasutajaliidese ventilaatorite töö juhtimise lehekülge. Juhtsüsteemi kasutamist visualiseeriv indikaator on algselt kuvatud kõige ülemises asendis ehk lehekülge nimetuse kõrval. Liigutades juhtkangi x-teljes, kui indikaator on kõige ülemises asendis, muutub ekraanile kuvatud kasutajaliidese lehekülge. Kokku on neli erinevat lehekülge. Temperatuuriandurite ja õhukvaliteedi andurite näitude lehekülgedel olles ei ole kasutajal võimalik süsteemi tööd mõjutada. Need on mõeldud ainult väärtuste kuvamiseks.

Valgustuse leheküljel kuvatakse valgustuse juhtimiseks kasutatavate 8-bitiste PWM signaalide väärtused vahemikus 0 kuni 255. Nende signaalide väärtusi on kasutajal võimalik muuta. Liigutades juhtkangi y-teljes ülemisest asendist alla poole saab kasutaja x-teljes juhtkangi liigutades muuta valgustuse alatoonidele saadetud signaalide väärtusi sammuga 5. Muutub selle alatoonide PWM signaali väärtus, mille kuvatud nimetuse ette on indikaator kuvatud. Vajutades juhtkangi nupule, kui indikaator on alatoonide nimetuse ees, rakendub kõigile kolmele alatoonile ekraanile kuvatud PWM signaali väärtus. Enne signaalide rakendamist on võimalik kasutajal näha eelvaadet valitud valgustuse värvitoonist lehekülje alumisse serva kuvatavas ristkülikus.

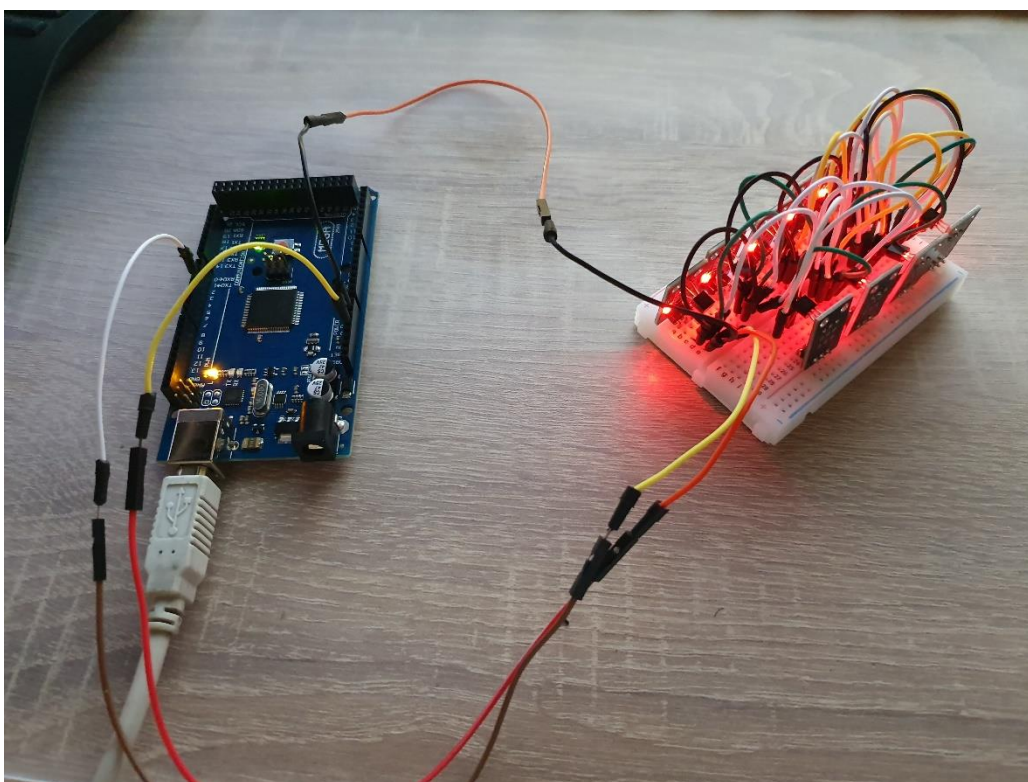


Joonis 2.3 Kasutajaliidese lehekülje ülesehitus.

Ventilaatorite juhtimise leheküljel liigutades indikaatorit ülevalt teise menüü ritta on võimalik muuta ventilaatorite töörežiimi. Juhtsüsteemi käivitamisel on ventilaatorid algselt välja lülitatud. Liigutades teises reas olles juhtkangi x-teljes muutub, mis töörežiimi nimetus kuvatakse. Kokku on neli erinevat töörežiimi õhuvahetuse juhtimiseks. Algne, kus kõik ventilaatorid on välja lülitatud. Manuaalne juhtimine, mis võimaldab kasutajal ise valida konstante kiirus eraldi sissetõmbe ja väljatõmbe ventilaatoritele. Seadepunkti režiim, mis üritab kaitseümbrises hoida kindlat temperatuuri väärtust mille juhtsüsteemi kasutaja määrab. Selle režiimi korral arvutab ventilaatorite kiirused välja PID-regulaator, mis lähtub kaitseümbrises olevate temperatuuri andurite mõõtetulemuste aritmeetilisest keskmisest väärtusest. Viimane režiim on mõeldud selleks, et filtreerida kaitseümbrises olev saastunud õhk VOC-dest ning mikroosakestest enne uste avamist.

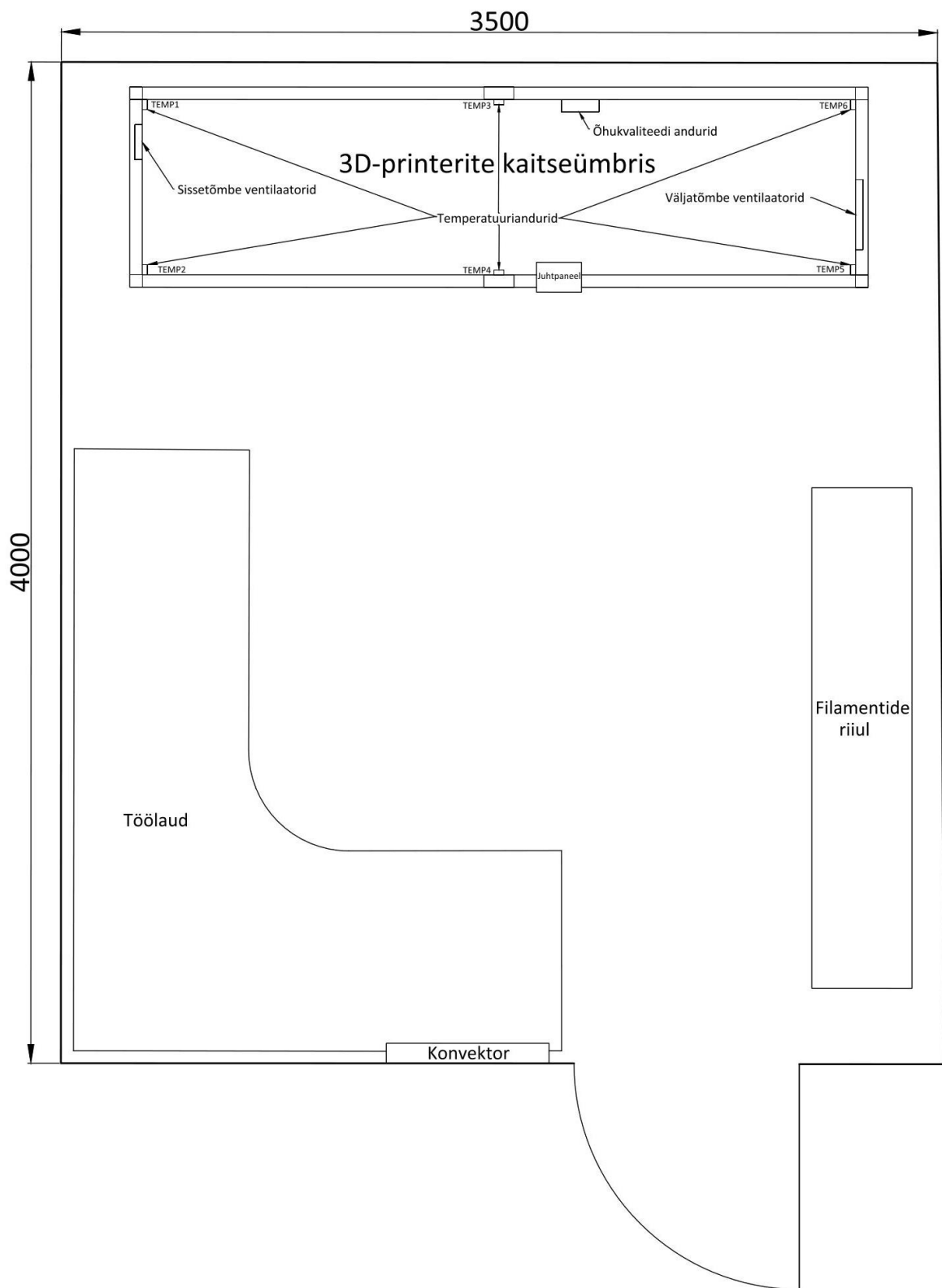
3. TEOSTUS

Projekteeritud juhtsüsteemi terviklikule realiseerimisele eelnes süsteemi osade eraldi testimine. Kontrolliti üle kõikide andurite toimimine, ventilaatorite juhtimine valitud kontrollseadmega, valgustuse juhtimise toimimine, juhtkangilt sisendite saamine ning kas makettplaatidele koostatud elektriskeemid toimivad korrektselt. Eeltestimine oli vajalik selleks, et kiirendada tervikliku süsteemi koostamist ning programmeerimist, sest selle käigus oli võimalik likvideerida riistvara ja tarkvaraga seotud probleeme.



Joonis 3.1 Temperatuuri andurite töö testimine.

3D-printerite kaitseümbris asub ettevõtte tootmises spetsiaalselt 3D-printerite jaoks mõeldud ruumis. Kaitseümbris asub ukse vastasseina ääres. Samal seinal, kus asub uks, on konvektor kütteseade. Lisaks kaitseümbrisele on toas veel riiul 3D-printimise filamentide hoiustamiseks ning töölaud erinevate tööde tegemiseks. Joonisel 3.2 on peale ruumiplaani välja toodu ka juhtsüsteemi osade asetus kaitseümbrise sees. Selle joonisel on näha ventilaatorite, temperatuuriandurite, õhukvaliteedi andurite ja juhtpaneeli asukoht.



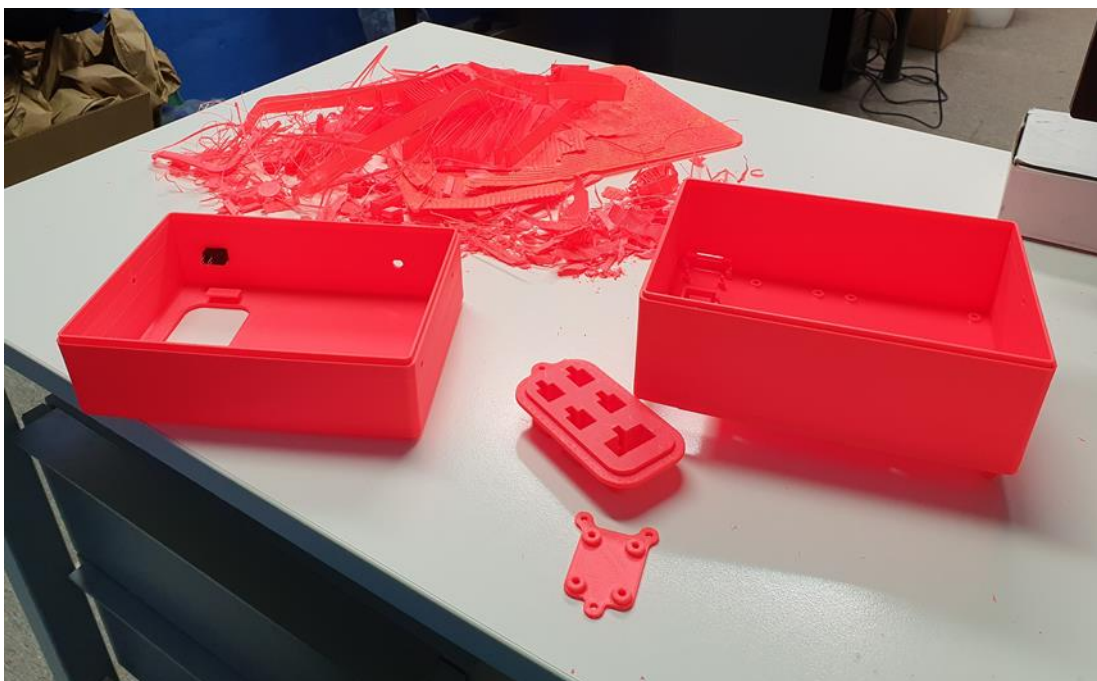
Joonis 3.2 Lihtsustatud ruumiplaan koos ruumi sisseadega. Mõõdud millimeetrites.

3.1 Elektroonika korpuste tootmine

Elektroonika korpuste tootmiseks kasutati ettevõttes olemasolevaid Zortraxi 3D-printereid. Korpuste detailide CAD-mudelid salvestati stl failidena ning muudeti 3D-printeritele sobilikeks failideks kasutades Z-Suite tarkvara. Selle tarkvara abil oli võimalik seadistada kõik 3D-printimise parameetrid nagu näiteks, kihtide paksus, printimispinnase ja ekstruuderi temperatuur jne. See jagab seadistuste põhjal detailid prinditavateks kihtideks ning lisab tugimaterjali. Selleks, et tagada detailide kinnijäämine printimispinnale ja vähendada kaardumise ohtu, kaeti printimispind ABSi ja atsetooni seguga.

Korpuste valmistamiseks kasutati materjali nimega Z-ULTRAT. See materjal oli juba ettevõttes olemas ning ettevõtte kogemuste põhjal vähem kaarduv kui tavaline ABS plastmass. Z-ULTRAT materjali kasutati kolmes erinevast värvitoonis, sõltuvalt sellest, mis oli sellel hetkel saadaval.

Kõik detailid tuli puhastada tugimaterjalist peale 3D-printimist. Avade mõõtude erinevused, mis tekkisid materjali kokku tõmbumisest jahtumisel, parandati eemaldades üleliigne materjal viilides või avad üle puurides.



Joonis 3.3 Juhtsüsteemi korpuse detailid koos eemaldatud tugi materjaliga.

Väiksemaid ja lihtsamaid korpuseid oli võimalik modifitseerida ja uuesti toota, kui need ei olnud sobilikud. Kuigi juhtpaneeli korpuse detailidel olid disainivigastid ja ühel detailil esines märgatavat

kaardumist võeti need kasutusele, sest COVID-19 epideemia tõttu oli 3D-printimise filamentide kättesaadavus piiratud ning detailide pikk printimisaeg oleks seganud tellimuste täitmist.

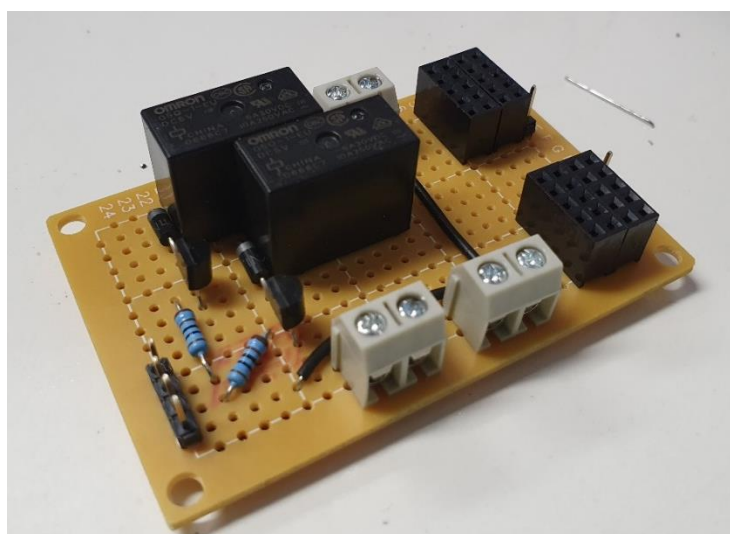


Joonis 3.4 Temperatuuri andurite korpuste arenduse käigus loodud variandid.

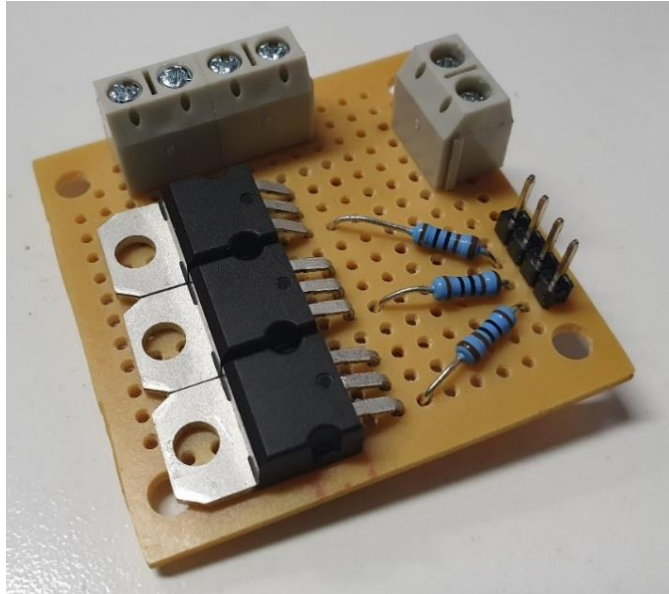
3.2 Elektroonika ja korpuse sisese juhtmestuse valmistamine

3.2.1 Elektroonika ja juhtpaneel

Juhtsüsteemi elektroonika koosneb peamiselt ostetud valmis trükkplaatidest. Lisaks neile trükkplaatidele loodi makettplaatide peale veel kaks elektroonika plaati. Esimesel makettplaadil on releed ja skeemid, mis on vajalikud nende lülitamiseks. Selle plaadi ühte otsa on lisatud võimalus toitepinget laiendada kasutades Dupont 2,54 mm pistikuid, mida kasutavad enamik süsteemis olevaid elektroonikaseadmeid. Teisele makettplaadil on komponendid ja elektriskeemid, mis on vajalikud, et Arduino abil oleks võimalik juhtida kaitseümbrise valgustust.



Joonis 3.5 Releede lülitamise plaat.



Joonis 3.6 RGB LED-riba kontrollimise plaat.



Joonis 3.7 Juhtsüsteemi korpus koos elektroonikaga.

3.2.2 Kaitseümbrise ja juhtpaneeli juhtmestus

Kaitseümbrise sisse veeti juhtmestus andurite, täiturite ja valgustuse jaoks. Kogu juhtmestuse koondumispunktiks on juhtpaneel. Juhtmetena kasutati 0,14 mm² ja 0,35 mm² ristlõikega isoleeritud kiulisi vaskjuhtmeid. Kogu juhtmestus on aetud läbi musta punutud nailonist kaitseümbrise, et vältida nende kahjustamist ning teha need vähem märgatavaks. Suurema

ristlõikega, $0,35 \text{ mm}^2$, juhtmed on kasutusel 12V toiteahelas, kus on kõige suurem voolutarbimine. $0,14 \text{ mm}^2$ ristlõikega juhe on kasutusel kõikide muude ühenduste tegemiseks.



Joonis 3.8 Osa andurite juhtmestusest koos andurite korpustega.

Juhtmestuses on kasutatud pistikute tootja Molex erinevate seeriade ja Dupont 2,54 mm pistikuid. Dupont pistikud on kasutusel juhtpaneeli sisestes ühenduste tegemiseks. Neid pistikuid kasutavad ka temperatuuriandurite, VOC-de ja suitsu tuvastamise andurite valmis moodulid. Honeywell HPMA115S0-XXX õhus olevate mikroosakeste arvu mõõtja kasutab Molex PicoBlade seeriapistikuid, mis on väiksed ja õrnad ning vältimaks vajadust nende lahti ühendamise järele on juhtmestuse tehtud Dupont pistikutega vaheühendus. Ventilaatorite ühendamiseks juhtmestusega kasutatakse Molex KK seeria pistikuid, mida kasutavad enamik arvuti ventilaatoreid. Vältimaks juhtpaneeli, mis on ukse küljes, külge ühenduvate juhtmete lahti tulemist kasutatakse lukustusmehhanismiga Molex Mini-Fit Jr. seeria pistikuid. Need on ka kasutusel 12 V ahela ühenduste jaoks juhtpaneeli sees.



Joonis 3.9 Juhtsüsteemi korpuse sisene elektroonika ja juhtmestus.

4. KASUTUSELE VÕTMINE JA TESTIMINE

Loodud ja kasutusel võetud lahenduse toimimise hindamiseks mõõdeti õhutemperatuuri stabiilsust 3D-printimise ajal. Võrreldi mõõtetulemusi, mis saadi juhtsüsteemi kasutamisel kaitseümbrise õhuvahetuse juhtimiseks, kasutamata toas olevat kütteseadet, ning mis on saadud, kui 3D-printerite jaoks üritatakse tagada sobilik temperatuur tervet tuba küttes. Sooritati kolmas mõõtmine, et näha kas ja kui stabiilsena suudab juhtsüsteem hoida kaitseümbrise sisest temperatuuri, kui tuba proovitakse ventileerimisega tahtlikult maha jahutada. Esimese ja teise temperatuuri mõõtmise ajal mõõdeti ka VOC-de ja mikroosakeste kontsentratsioon õhus, et hinnata kuidas kaitseümbrise kasutamine mõjutab ruumi õhukvaliteeti.

Tabel 4.1 Mõõtmiste andmed.

	Kuupäev	Katse kestus	t_v^0	Lisainfo
Mõõtmine 1	01.05.2020	16.00 - 23.00	24,5 °C	<ul style="list-style-type: none">• Termostaadiga küttekeha seadepunkt: 28 °C.• 3D-printerid paigutatud usteta kaitseümbrisesse.
Mõõtmine 2	02.05.2020	16.00 - 23.00	24,4 °C	<ul style="list-style-type: none">• Termostaadiga küttekeha pole kasutusel.• 3D-printerid paigutatud koos juhtsüsteemiga kaitseümbrisesse.• Kaitseümbrise sisetemperatuuri seadepunkt: 35 °C.
Mõõtmine 3	08.05.2020	13.00 – 23.00	26,3 °C	

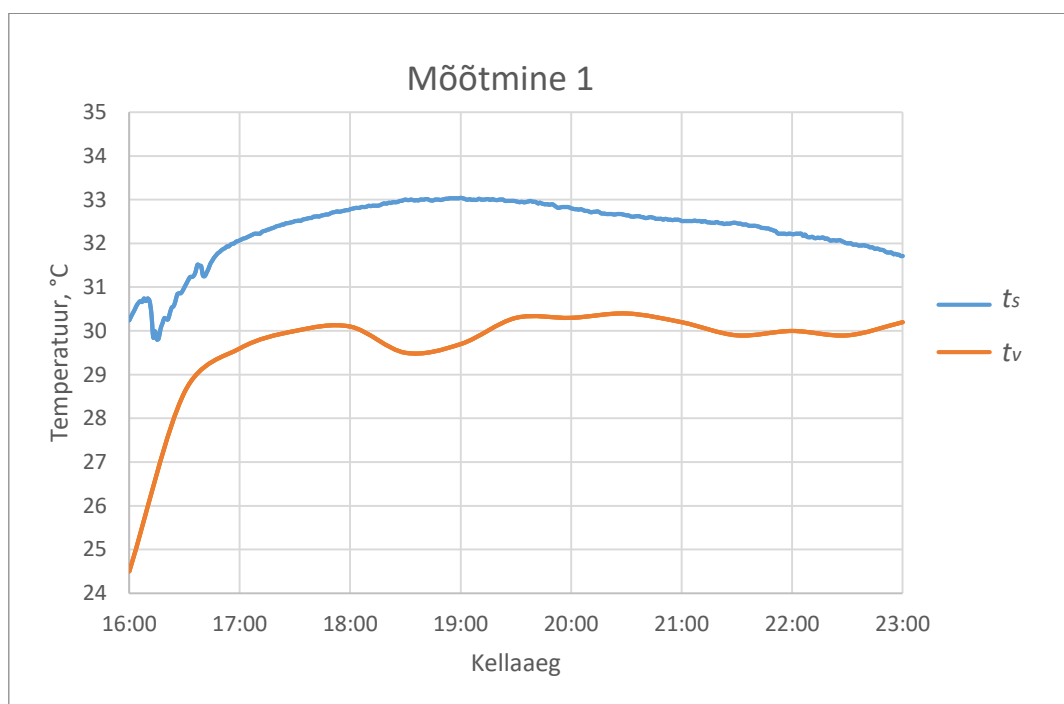
Mõõtetulemuste kogumiseks kasutati kahte PT100 tüüpi takistustermomeetrit. Üks neist asetati kaitseümbrise sisse joonisel 3.8 kujutatud õhukvaliteedi andurite juurde ning teine ruumis oleva töölaua peale. Kaitseümbrises olev temperatuuri andur ühendati andmete logeriga, mis seadistati mõõtma temperatuuri intervalliga üks minut. Laua peal olev andur ühendati temperatuurikontrolleriga mille ekraanile kuvatud väärtusi kirjutati üles iga poole tunni tagant.

Ruumi õhukvaliteedi parameetrite mõõtmiseks kasutati kahte komplekti andureid. Ühe komplektina kasutati juhtsüsteemi VOC-de ja õhu leiduvate mikroosakeste arvu mõõtjaid ning teise komplektina identseid andureid, mis olid asetatud ruumis oleva töölaua peale ja ühendatud läbi Arduino arvuti külge. Esimese komplekti näite vaadati juhtsüsteemi ekraanilt ning teise anduri komplekti näite vaadati läbi Arduino IDE tarkvara arvutist.

4.1 Temperatuuri mõõtmised

4.1.1 Mõõtmine 1

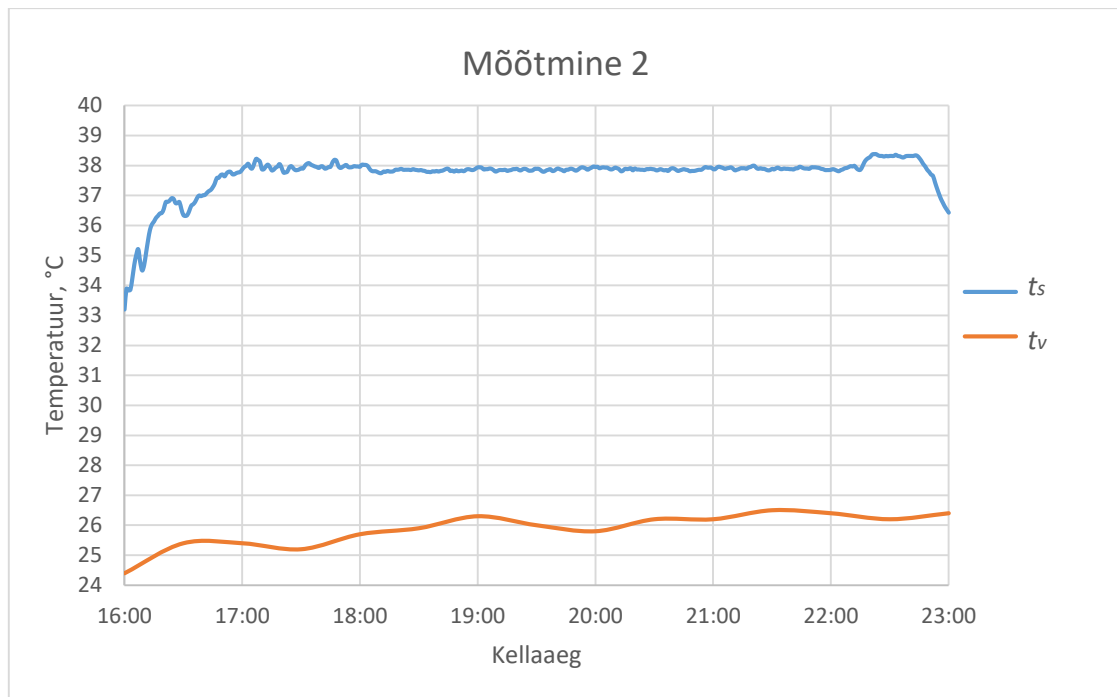
Mõõtmise 1, esitatud joonisel 4.1, korral stabiliseerus kaitseümbrise sisene temperatuur t_s tunni aja jooksul peale katse algust. Kaitseümbrise sisene temperatuur selle perioodi lõpuks oli 32 °C. Seejärel temperatuur tõusis kella 18.30-ks 33 °C juurde ja hakkas 19.30 sujuvalt langema. Katse lõpuks oli temperatuur kaitseümbrises 31,7 °C. Toatemperatuur t_v oli esimese katse korral stabiliseerumisperioodi lõpuks 29,6 °C ning varieerus siis kuni katselõpuni vahemikus $30 \pm 0,5$ °C.



Joonis 4.1 Kaitseümbrise sisene ja väline temperatuur esimese mõõtmise jooksul.

4.1.2 Mõõtmine 2

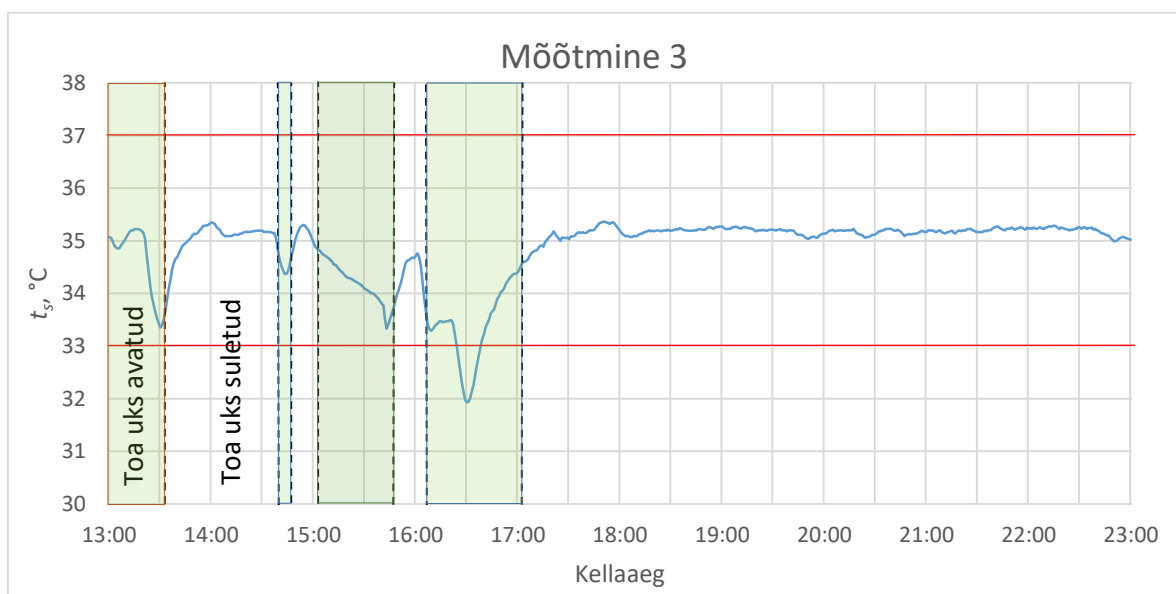
Teise katse, esitatud joonisel 4.2, stabiliseerus t_s samuti tund aega peale katse algust. Peale stabiliseerumisperioodi oli temperatuur $38 \pm 0,2$ °C kuni kella 22.15-ni. Peale seda tõusis temperatuur pooleks tunniks 38,4 °C juurde ning hakkas siis langema. See langus oli tingitud sellest, et 3D-printerid ei eraldanud enam soojust, sest 3D-prinditavad detailid olid valmis saanud. Kaitseümbrise sisene temperatuur oli stabiliseerumisest kuni languse alguseni lubatud piiridest, 35 ± 2 °C, väljas. Teisel katsel tõusis ruumitemperatuur stabiliseerumisperioodi lõpuks 25,4 °C juurde ning kõikus siis vahemikus $26 \pm 0,8$ °C mõõtmise lõpuni.



Joonis 4.2 Kaitseümbrise sisene ja väline temperatuur teise mõõtmise jooksul.

4.1.3 Mõõtmine 3

Kolmanda mõõtmise eesmärgiks oli tahtlikult mõjutada kaitseümbrise sisetemperatuuri stabiilsust ruumi maha jahutamiseks ventileerimisega. Ruumi maha jahutamiseks avati ja sulgeti toa uks ebaregulaarsete ajavahemike tagant, perioodil kell 13.00 kuni 17.00, et simuleerida reaalselt iga päevast toa kasutamist, kus toa uks jäätakse pidevalt kogemata lahti mingiks perioodiks. Ukse avamisega lasti tuppa suurest tootmisruumist õhk temperatuuriga $16 \pm 1 \text{ }^\circ\text{C}$.



Joonis 4.3 Kaitseümbrise sisene temperatuur mõõtmise 3 jooksul.

Perioodil 13.00 kuni 17.00 jäid temperatuuri kõikumised vahemikku 31,9 kuni 35,6 °C. Temperatuuri muutumine kaitseümbrises ja vahemikud millal ruumi uks oli korda mööda kas avatud või suletud on välja toodud joonisel 4.3. Samal joonisel on punaste joontega ära märgitud temperatuuri lubatud erinevused seadeväärtusest. Peale tööpäeva lõppu jäi juhtsüsteem hoidma temperatuuri $35,1 \pm 0,2$ °C.

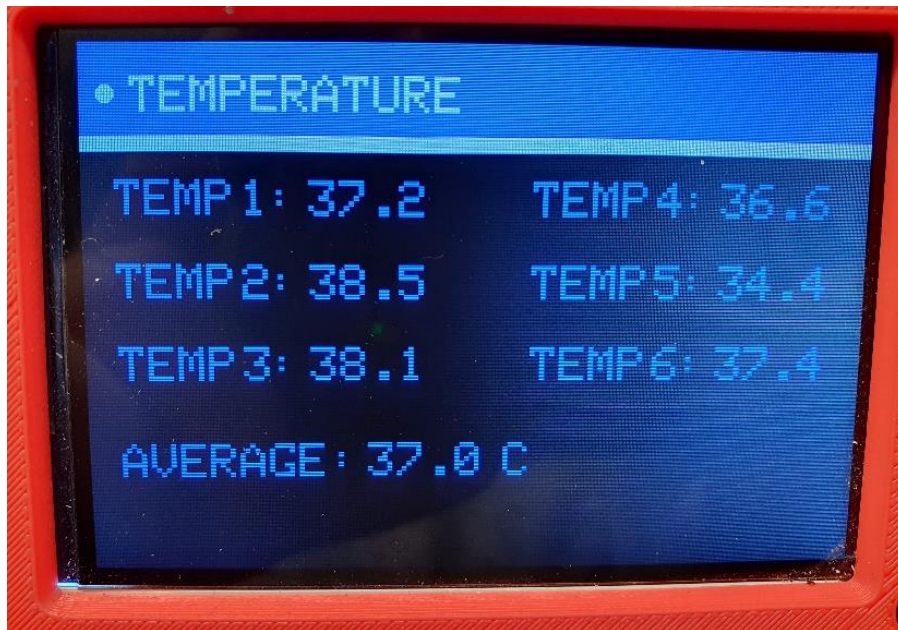
4.2 Mõõtmistulemuste analüüs

4.2.1 Temperatuuri tagamine kaitseümbrises

Mõlema mõõtmise korral oli näha tunni pikkust temperatuuri stabiliseerumisperioodi katse alguses. See periood 3D-printimisprotsessi ei mõjutanud, sest mõlema katse korral oli temperatuur alati vähemalt 30 °C ning enamik seda perioodi kulus 3D-printeritel töötemperatuurini soojenemiseks. Temperatuur püsis kaitseümbrises stabiilne peale stabiliseerumisperioodi mõlemal katsel. Esimesel katsel tõusis temperatuur kaitseümbrises kuni 33 °C ja hakkas siis aeglaselt tempos langema. Teisel katsel kõikus see kuni katse lõpus 3D-printerite töö lõppemist tingitud languseni 38 °C juures.

Esimese ja teise katse mõõtetulemuste võrdlemisel on näha, et esimesel katsel lisaks kaitseümbrise sisese õhu üles soojendamisele soojendati ka stabiliseerumisperioodi lõpuks üles terve tuba. Ruumitemperatuur oli esimesel mõõtmisel peale seda perioodi sarnaselt kaitseümbrise sisesele temperatuurile stabiilne mõõtmise lõpuni. Kõikudes selle aja jooksul 1 °C suuruses vahemikus. Teisel katsel oli ruumi soojenemine pikaajalisem protsess ning oli tingitud vaid 3D-printeritelt eralduvast soojusest. Toatemperatuur kasvas 2 °C võrra teise katse jooksul. Teise katse puhul oli näha, et kasutades nutikat kaitseümbrist ei ole vajadust tervet tuba soojaks kütta, et tagada 3D-printeritele sobilik temperatuur, vaid piisab ainult 3D-printeritelt eralduvast soojusest.

Teise katse tulemustel selgus, et kuigi temperatuur püsis 3D-printimise ajal kaitseümbrises stabiilne siis see ületas soovitud seadepunkti väärtust 3 °C võrra ning läks väljas poole lubatud piire, seadepunkt ± 2 °C. Arvestades seda, et temperatuur püsis mõõtmiste käigus stabiilsena saab oletada, et juhtimissüsteem kas ei reageerinud antud temperatuuri väärtusele või ei ole võimeline hoidma temperatuuri stabiilsena täpselt kasutaja poolt määratud seadepunktil. Seda miks temperatuur püsis küll stabiilne aga väljaspool lubatud piire aitab seletada juhtsüsteemi kasutajaliidese vahendusel ekraanile kuvatud kaitseümbrises olevate temperatuuriandurite mõõtetulemused, välja toodud joonisel 4.4.



Joonis 4.4 Kaitseümbrise siseste temperatuuriandurite mõõtetulemused teisel mõõtmisel.

Juhtsüsteem juhib ventilaatoreid vastavalt kaitseümbrise temperatuuriandurite keskmisel väärtusel, tähistatud joonisel 4.4 nimega AVERAGE. Mõõtmistulemuste kogumiseks kasutatud PT100 andur asus juhtsüsteemi temperatuurianduri, mille nimetus joonisel 4.4 on TEMP3, kõrval. Kuigi osad mõõdetud temperatuurid olid üle lubatud piiride siis juhtsüsteem neile ei reageerinud, sest keskmise temperatuuri järgi oli temperatuur kaitseümbrises täpselt lubatud ülemise piiri juures ning süsteemi jaoks ei olnud temperatuuri alandamine vajalik. Juhtsüsteem hoidis kogu 3D-printimise aja kaitseümbrise keskmise sisetemperatuuri lubatud ülemise piiri mitte seadepunkti väärtuse juures. See viitab sellele, et ventilaatoreid juhtiva PID regulaatori seadistus vajab optimeerimist.

Varieeruvus temperatuuri väärtustes tuleneb tõenäoliselt sellest missuguste 3D-printerite lähedal temperatuuriandurid asuvad. Ettevõttes kasutusel olevatel 3D-printeril on x- ja y-telg, mille küljes materjali ekstruuder liigub, fikseeritud kõrgusel masina ülemises osas ning printimispind liigub 3D-printimise ajal allapoole. Selline masinate ehitus tähendab, et osad temperatuuriandurid näitavad teistest kõrgemaid temperatuure, sest soojusallikas, ekstruuder, on neile lähemal. Kaitseümbrise ühes pooles asuvad Zortrax M300 seeria 3D-printerid on 150 mm kõrgemad kui M200 seeria 3D-printerid. Temperatuuri ebahütlus kaitseümbrises 3D-printimise kvaliteeti ei tohiks antud juhul mõjutada, sest kõik temperatuurid olid üle 30 °C.

Kolmanda mõõtmise tulemuste põhjal saab järeldada, et kaitseümbrise sisene temperatuur on mõjutatud sellest kas ruumi uks on lahti või mitte ehk väliskeskkonnast toimuvatest muutudest. Iga kord kui uks avati hakkas kaitse ümbrise temperatuur langema ja ukse sulgemisel see uuesti tõusis.

Perioodil 16.32 kuni 17.00 oli ainuke kord selle mõõtmise käigus kus temperatuur langes allapoole lubatud piiri ning ka ainuke kord, kui temperatuuri kaitseümbrises hakkas ukse avatud olemise ajal tõusma. See viitab sellele, et häiringute mõju oleks võimalik vähendada ventilaatoreid juhtiva PID regulaatori seadistuse optimeerimise või juhtimisalgoritmi täiendamisega

4.2.2 Ruumi õhukvaliteedi tagamine

Tabel 4.2 VOC-de ja mikroosakeste mõõtetulemused.

	Mikroosakeste kontsentratsioon väljas pool kaitseümbrist [ppm]	Mikroosakeste kontsentratsioon kaitseümbrise sees [ppm]	VOC-de kontsentratsioon väljas pool kaitseümbrist [ppm]	VOC-de kontsentratsioon kaitseümbrise sees [ppm]
Mõõtmine 1	60 - 80		26 - 117	
Mõõtmine 2	26 -30	250 - 260	0 - 10	39 - 392

Esimese temperatuuri mõõtmise ajal mõõdetud VOC-de ja mikroosakeste kontsentratsioonid ühikus osakest miljoni kohta (ing k. *particle per million*, ppm) kaitseümbrise sees ja väljas olid peaaegu identsed, näidud erinesid üksteisest 1 kuni 2 ppm võrra. See on tingitud sellest, et uste puudumise tõttu said osakesed ruumis vabalt laiali hajuda ning nende kontsentratsioon kogu ruumis ühtlustus. Kui selle mõõtmise ajal õhus olnud VOC-d kogus koosneks ainult stüreenist, mille 1 ppm = 4,26 mg/m³ [13], oleks ruumis õhus 111 kuni 498 mg/m³ stüreeni. See oleks üle töökeskkonna piirnormidest, mis on välja toodud peatükis 1.5.

Teise mõõtmise puhul vaadeldi mõlema anduri komplekti näite eraldi, sest kaitseümbrise takistas VOC-de ja mikroosakeste laiali levimist ruumi. Kaitseümbrise väliselt jäi kontsentratsioon vahemikku 0 kuni 10 ppm. Puhta stüreeni puhul vastaks see 0 kuni 42,6 mg/m³ -le, mis jääb töökeskkonnas lubatud piirnormidesse. Kaitseümbrise sees jäi VOC-de kontsentratsioon vahemikku 39 kuni 392 ppm, mis tähendaks, et juhul kui oleks tegemist oleks ainult stüreeniga oleks seda õhus 162 kuni 1667 mg/m³ ja see oleks üle kehtivate piirnormide.

Autori ja ta kolleegide subjektiivsete vaatlustel ei täheldatud juhtsüsteemi ja tervikliku kaitseümbrise kasutamisel VOC-de sattumist kaitseümbrise välisesse keskkonda. Ruumis ei olnud tunda enam ebameeldivat lõhna, mis tekib ABS plastmassi 3D-printimisel, ning toas oli lihtsam hingata. Inimestele tegi ruumis viibimise mugavamaks ka see, et ruumi temperatuur oli 4 kuni 5 °C madalam, kui siis kui 3D-printeritele tagati sobilik temperatuur kogu tuba küttes.

4.2.3 Juhtsüsteemi täiendamine ja edasiarendamine

Juhtsüsteemi täiendamist tuleks alustada ventilaatorite PID regulaatori parameetrite muutmisega. Tuleks seadistada see toimima nii, et tagaks võimalikult seadepunkti lähedase väärtuse ning ei jääks hoidma temperatuuri ülemise piiri juures. Selleks oleks vaja viia läbi uusi mõõtmisi, et näha kuidas PID regulaatori parameetrite muutmine mõjutab süsteemi tööd ja neid parameetreid vastavalt tulemustele korrigeerida.

Tulevikus tuleks kaaluda ka ventilaatorite juhtimisalgoritmi lisada peale temperatuuri keskmise väärtuse jälgimise ka iga temperatuurianduri näidu eraldi jälgimine. Selle abil oleks võimalik tagada, et juhul, kui ühe kaitseümbrise sisese piirkonna temperatuur läheb piirides välja piisavalt, et mõjutada printimiste kvaliteeti või rikkuda seadmeid, reageerib süsteem ka sellele. Hetkel on oht, et sellised olukorrad ei peegeldu alati keskmises temperatuuri väärtuses, mille järgi õhuvahetuse juhtimine toimib.

Süsteemi häirekindluse tõstmisele aitaks kaasa temperatuurianduri, mis mõõdab toatempratuuri, lisamine. Selle abil oleks juhtimissüsteemil võimalik kiiremini reageerida väliskeskkonnas toimuvatele muutustele nii, et kaitseümbrise sisetemperatuur ei oleks neist mõjutatud. Häiringuid oleks võimalik ka vähendada integreerides juhtsüsteemiga ruumi küttekeha juhtimine. See võimaldaks anda ruumi lisa soojust juhul, kui 3D-printrite enda soojusest ei piisa kaitseümbrise temperatuuri tagamiseks. Häiringute vähendamist tuleks silmas pidada ka PID regulaatori ja juhtimisalgoritmi täiendamisel.

KOKKUVÕTE

Käesolevas bakalaureusetöös on kirjeldatud 3D-printerite nutika kaitseümbrise juhtsüsteemi arenduskäik ning selle valmistamine.

Lõputöö esimeses osas on välja toodud, mida kujutab endas FDM 3D-printimistehnoloogia ning milliseid materjale selle tehnoloogiaga kasutada on võimalik. Tuuakse välja miks oleks 3D-printerite eraldamine muust keskkonnast ning ühtlase temperatuuri hoidmine FDM seadmete ümber vajalik nii inimeste heaolu huvides kui ka kvaliteetsete detailide tootmiseks. Samuti arvutatakse välja lahenduse projekteerimisele oluline õhuvooluhulk ning tuuakse välja funktsionaalsus, mis selle probleemi lahendusel peab olema. Taustinfo kogumise käigus jõuti järeldusele, et 3D-printerid on vaja kindlasti eraldada inimeste töökeskkonnast ning turul ei ole ettevõtte vajadustele ühtegi sobivat lahendust, mis tegi vajalikuks uue lahenduse arendamise.

Järgmises peatükis kirjeldatakse juhtimissüsteemi projekteerimisprotsessi. Põhjendatakse komponentide valikut, seadme CAD-mudeli disaini ja elektriskeemi koostamisel arvesse võetud tegureid. Antakse ülevaade juhtsüsteemi tarkvara ülesehitusest ning kirjeldatakse kasutajaliidese toimimist. Kasutades ära ettevõttes olemas vahendeid suudeti projekteerida lahendus, mis oli odavam ja funktsionaalsem kui turul leiduvad lahendused.

Kolmandas peatükis antakse edasi kuidas juhtsüsteem valmistati. Tuuakse välja, mis tootmiseseadmete abil toodeti elektroonika korpused, millest koosneb juhtpaneelis olev elektroonika ning kuidas on juhtsüsteemi juhtmestus koostatud. Kirjeldatakse kuidas kaitseümbris, juhtpaneel, andurid ja muu sisseseade üksteise suhtes toas asuvad. Valmib projekteeritud seade.

Viimases peatükis näidatakse juhtsüsteemi testimisel kogutud mõttetulemusi ning antakse nende põhjal hinnang juhtsüsteem võimekusele tagada kaitseümbrises sobiv ja stabiilne õhutemperatuur. Hinnatakse samuti kas antud lahendus aitab parandada ruumi õhukvaliteeti. Mõõtmistulemuste analüüsimisel selgus, et juhtsüsteemi abil suudetakse tagada stabiilne temperatuur printimise ajal juhul, kui ei ole välismõjusid. Samuti selgus, et nutika kaitseümbrise kasutamine tagab inimesele ohutuma töö keskkonna. Välja tuli ka kindel vajadus süsteemi juhtalgoritme tulevikus täiendada, et süsteem hoiaks temperatuuri täpselt kasutaja soovitud väärtuse juures.

Lõputöö tulemusena said lahendatud lõputöö ülesandes püstitatud küsimused ja valmistatud seade, mis vastab autori ja ta tööandja poolt sellele seadmele etteseadud primaarsetele nõutele. Seade on võimeline edukalt juhtima kaitseümbrise õhuvahetust ja sisetemperatuuri ning oli kasutaja poolt ümber seadistatav kasutajaliidese vahendusel. Nutikas kaitseümbris suutis tagada ühtlase temperatuuri 3D-printimise ajal peale seda, kui 3D-printerid olid üles soojenenud

kaitseümbrise sisese õhu. Näha oli aga seda, et seda selle juhtimisalgoritmid vajavad edasiarendamist ning stabiilsust on võimalik häirida mõjutades kaitseümbrise väliskeskkonda. Vaatamata sellele, et kaitseümbris ei ole hermeetiliselt suletav täheldati, et see suudab tagada inimesele ohutuma töökeskkonna.

Autor näeb, et seadet oleks võimalik tulevikus edasi arendada, optimeerida ja probleemsed kohad parandada. Prototüübi põhjal oleks võimalik luua lihtsustatud lahendus, mida saaks müüa näiteks 3D-printerite kodukasutajatele. Tulevikus tuleks ettevõttes juhtsüsteemi ja ruumi küttekeha juhtimine omavahel integreerida ning lisada veel toale ventilatsioon. See teeks võimalikuks toa ja kaitseümbrise sisekliima koos juhtimise, mis aitaks vältida toa ülekuumenemist või liigset jahtumist ning samal hoida kaitseümbrises võimalikult stabiilset temperatuuri.

SUMMARY

The following Bachelor's thesis describes the design process and construction of a control system for 3D printer smart enclosure.

In the first chapter of the thesis, the author describes what is FDM 3D printing and what materials can be used to print with it. Reason for why 3D printers should be separated from rest of the environment and why a stable temperature should be maintained around them are brought out. Also, airflow needed to keep a stable temperature is calculated and functionality that the system must have is brought out. Conclusion that 3D printer must be certainly removed from peoples work environment and there are no suitable solution available so a new one is needed were made while gathering background information about the topic.

The next chapter brings out the factors taken into account during component choice, CAD, electrical and software design. Authors gives an overview of the software structure and how the user interface is meant to function. A cheaper and more functional solution than what were available on the market could be designed by using the resources that the company has.

The third chapter gives an overview of how the control system was made. It is described how the housings for the electronics were made, how the electronics for the control system was put together and how the needed wiring and connections were made. The positioning of the enclosure, control panel, sensors and other installation is described. Designed device is assembled.

The final chapter shows results of testing conducted to evaluate the capability of the control system to maintain a stable and high enough temperature inside the enclosure. Also, how the system effects the air quality and well-being of the people inside the room is analyzed. Analyzing the measurement results showed that the control system can keep the internal temperature of the enclosure stable during 3D printing. Also, using the enclosure seemed to help improve the work environment. A clear need for improving the systems control algorithms in the future so it would maintain the exact temperature user wished for surfaced.

Questions raised in the thesis task were answered and a device fulfilling the primary needs set for it was manufactured as result of this Bachelor's thesis. The device was able to control the ventilation and internal temperature of the enclosure while also being configurable by the user through the user interface. Smart enclosure could maintain a stable temperature during 3D printing after the 3D printers had warmed up the air inside the enclosure. Need for the systems control algorithms to be developed further and the stability of that temperature could be affected by changing the

enclosures outer environment was also seen. Eventhough the enclosure was not airtight it seemed to make peoples work environment safer.

The author thinks that the device could be further developed, optimized and its problems fixed in the future. A simplified solution could be developed based on the device and sold to 3D printer home user. In the future for the companies use case the control system and heating management should be integrated and ventilation for the room added. It would make possible to control the indoor climate of the enclosure of the room together what would help avoid the room overheating or getting too cold while maintaining as stable as possible temperature inside the enclosure.

KASUTATUD KIRJANDUS

- [1] Qian Zhang, Jenny P. S. Wong, Aika Y. Davis, Marilyn S. Black, Rodney J. Weber, "Characterization of particle emissions from consumer fused deposition modeling 3D printers", School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA, School of Earth and Atmospheric Sciences, Georgia Institute of Technology, Atlanta, Georgia, USA, Underwriters Laboratories Inc., Marietta, Georgia, USA, 2017. [Võrgumaterjal]. Saadaval: <https://www.tandfonline.com/doi/full/10.1080/02786826.2017.1342029>.
- [2] Parham Azimi, Dan Zhao, Claire Pouzet, Neil E. Crain, Brent Stephens, "Emissions of Ultrafine Particles and Volatile Organic Compounds from Commercially Available Desktop Three-Dimensional Printers with Multiple Filaments", Department of Civil, Architectural and Environmental Engineering, Illinois Institute of Technology, Chicago, USA, Ecole des Ingenieurs de la Ville de Paris, 80 Rue Rebeval, Paris, France, Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin, USA, 2016. [Võrgumaterjal]. Saadaval: <https://pubs.acs.org/doi/full/10.1021/acs.est.5b04983>.
- [3] Maailma Terviseorganisatsioon Euroopa, "Styrene" , 2000. [Võrgumaterjal]. Saadaval: http://www.euro.who.int/__data/assets/pdf_file/0018/123066/AQG2ndEd_5_12Styrene.pdf?ua=1. Kasutatud: 28.04.2020.
- [4] Vabariigi Valitsus, (2019, oktoober 10), "Töökeskonna keemiliste ohutegurite piirnormid". [Võrgumaterjal]. Saadaval: https://www.riigiteataja.ee/akti/1171/0201/9002/VV_105m_lisa.pdf# . Kasutatud: 28.04.2020.
- [5] Ameerika Ühendriikide keskkonnaamet, "Caprolactam", 2000. [Võrgumaterjal]. Saadaval: <https://www.epa.gov/sites/production/files/2016-09/documents/caprolactam.pdf>. Kasutatud: 28.04.2020.
- [6] Futerro, "The lactide", [Võrgumaterjal]. Saadaval: http://www.futerro.com/products_lactide.html. Kasutatud: 28.04.2020
- [7] Alkaios Bournias Varotsis, 3D Hubs, "Introduction to FDM 3D printing". [Võrgumaterjal]. Saadaval: <https://www.3dhubs.com/knowledge-base/introduction-fdm-3d-printing/>. Kasutatud: 26.04.2020.
- [8] Simplify3D, "ABS". [Võrgumaterjal]. Saadaval: <https://www.simplify3d.com/support/materials-guide/abs/>. Kasutatud: 26.04.2020.

- [9] Simplify3D, "HIPS". [Võrgumaterjal]. Saadaval: <https://www.simplify3d.com/support/materials-guide/hips/>. Kasutatud: 26.04.2020.
- [10] Simplify3D, "Nylon". [Võrgumaterjal]. Saadaval: <https://www.simplify3d.com/support/materials-guide/nylon/>. Kasutatud: 26.04.2020.
- [11] Simplify3D, "Polycarbonate". [Võrgumaterjal]. Saadaval: <https://www.simplify3d.com/support/materials-guide/polycarbonate/>. Kasutatud: 26.04.2020.
- [12] Simplify3D, "PLA". [Võrgumaterjal]. Saadaval: <https://www.simplify3d.com/support/materials-guide/pla/>. Kasutatud: 26.04.2020.
- [13] Zortrax, "Safety data sheet: Z-ULTRAT", Z-ULTRAT andmeleht, 2018. [Võrgumaterjal]. Saadaval: https://cf.zortrax.com/wp-content/uploads/2018/06/Z-ULTRAT_Safety_Data_Sheet_eng-1.pdf. Kasutatud: 26.04.2020.
- [14] Marta Iwańczyk, Zortrax (2018, juuli 23), "The Complete Guide: How to Avoid Warping in Your 3D Prints". [Võrgumaterjal]. Saadaval: <https://zortrax.com/blog/reducing-warping-in-3d-prints/>. Kasutatud: 28.04.2020.
- [15] Ultimaker (2020, veebruar 25), "How to fix warping". [Võrgumaterjal]. Saadaval: <https://support.ultimaker.com/hc/en-us/articles/360012113239>. Kasutatud: 28.04.2020.
- [16] Prusa3D, "Warping". [Võrgumaterjal]. Saadaval: <https://www.prusa3d.com/warping/>. Kasutatud: 28.04.2020.
- [17] Zortrax, "Specifications and operational characteristics of Zortrax M Series 3D printers" [Võrgumaterjal]. Saadaval: <https://support.zortrax.com/m-series-specification/>. Kasutatud: 29.04.2020.
- [18] Zortrax, "Specifications and operational characteristics of the Zortrax M300 Plus" [Võrgumaterjal]. Saadaval: <https://support.zortrax.com/m300-plus-specification/>. Kasutatud: 29.04.2020.
- [19] The Engineering ToolBox, "Design of Ventilation Systems" 2003. [Võrgumaterjal]. Saadaval: https://www.engineeringtoolbox.com/design-ventilation-systems-d_121.html. Kasutatud: 29.04.2020.
- [20] Zortrax, "Zortrax HEPA Cover". [Võrgumaterjal]. Saadaval: <https://store.zortrax.com/parts-and-accessories/zortrax-hepa-cover-m200>. Kasutatud: 29.04.2020.

- [21] Zortrax, "Zortrax HEPA Cover". [Võrgumaterjal]. Saadaval: <https://store.zortrax.com/parts-and-accessories/zortrax-hepa-cover-m300>. Kasutatud: 29.04.2020.
- [22] 3DPrintClean, "870 Safety Enclosure Pro". [Võrgumaterjal]. Saadaval: <https://www.3dprintclean.com/product-page/model-870-safety-enclosure>. Kasutatud: 29.04.2020.
- [23] Jure Korber, MakerBOT Thingiverse. "Universal 3D Printer Smart Enclosure" . [Võrgumaterjal]. Saadaval: <https://www.thingiverse.com/thing:2792618> Kasutatud: 29.04.2020.
- [24] Arduino, "Digital pins". [Võrgumaterjal]. Saadaval: <https://www.arduino.cc/en/Tutorial/DigitalPins>. Kasutatud: 05.04.2020.
- [25] Cryorig, "QF140". [Võrgumaterjal]. Saadaval: <http://www.cryorig.com/qf140.php>. Kasutatud: 05.04.2020.
- [26] Dallas Semiconductor, " DS18B20 Programmable Resolution 1-Wire® Digital Thermometer", DS18B20 temperatuurianduri andmeleht. Saadaval: <http://data.oomipood.ee/kasutusjuhend/DS18B20.pdf>. Kasutatud: 05.04.2020.

LISAD

Lisa 1 Pildid detailide kaardumisest

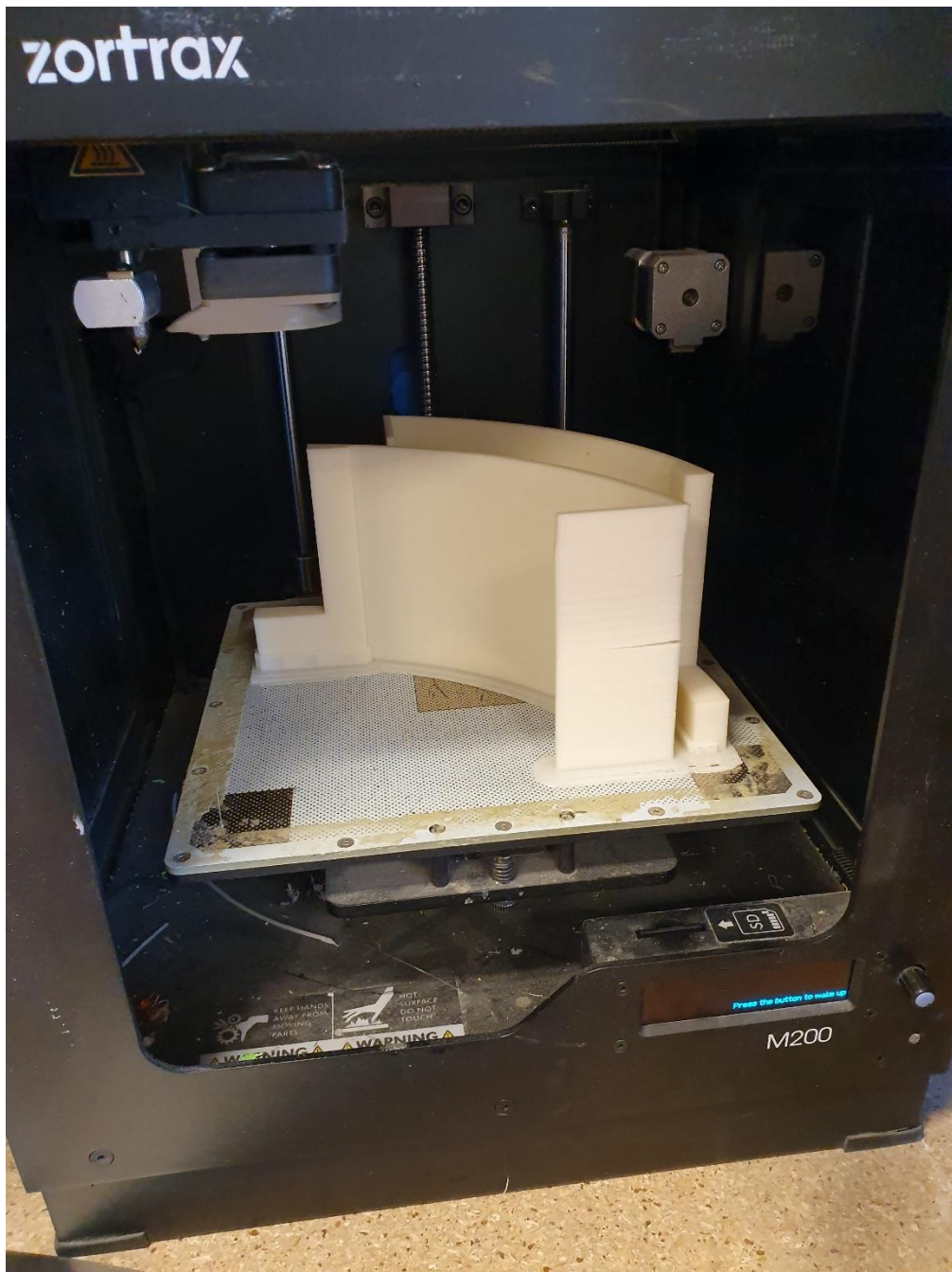
L1.1 Juhtsüsteemi korpuse kaardumine



L1.2 Detaili tugev kaardumine



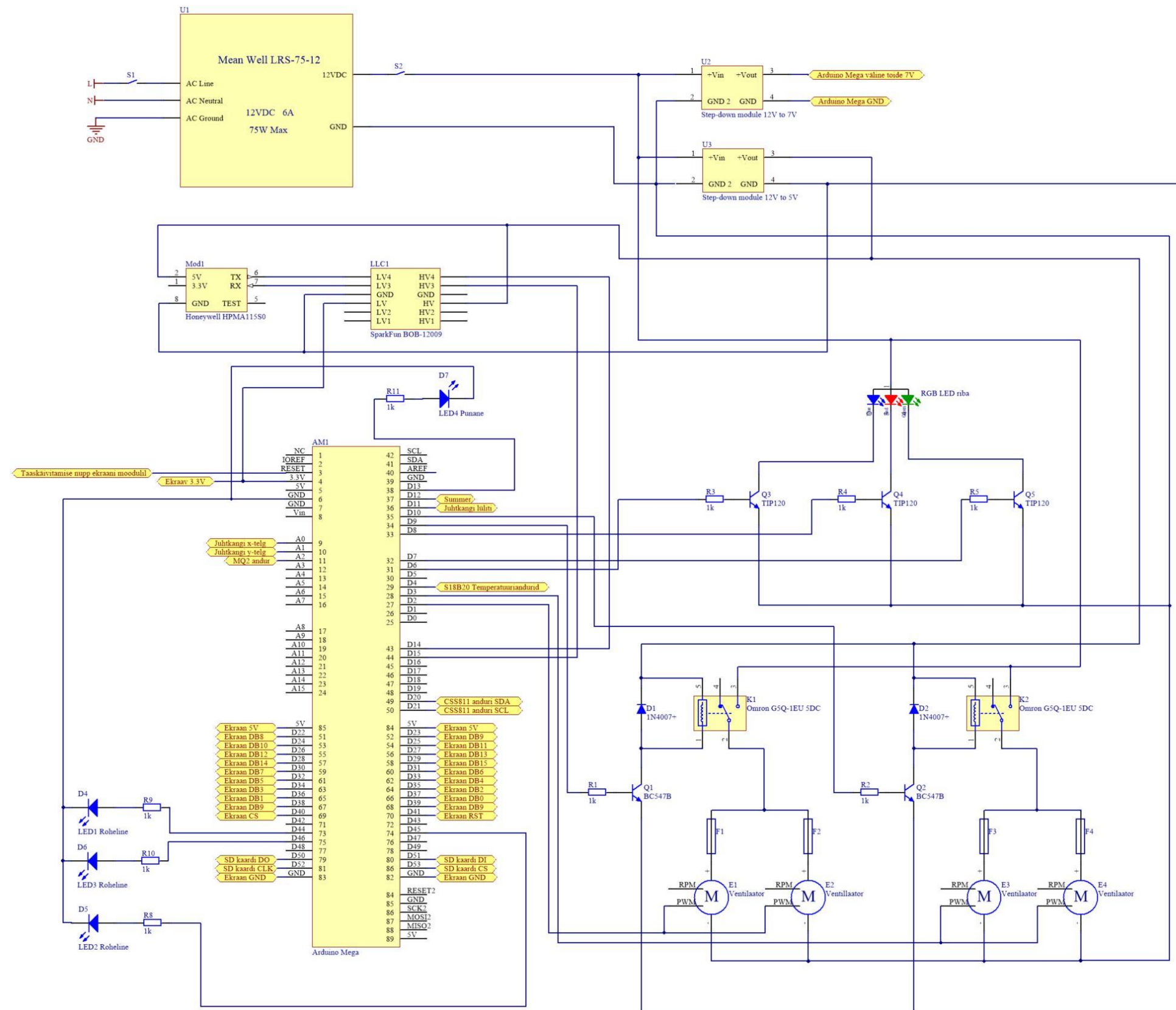
L1.3 Kaardumine keset detaili

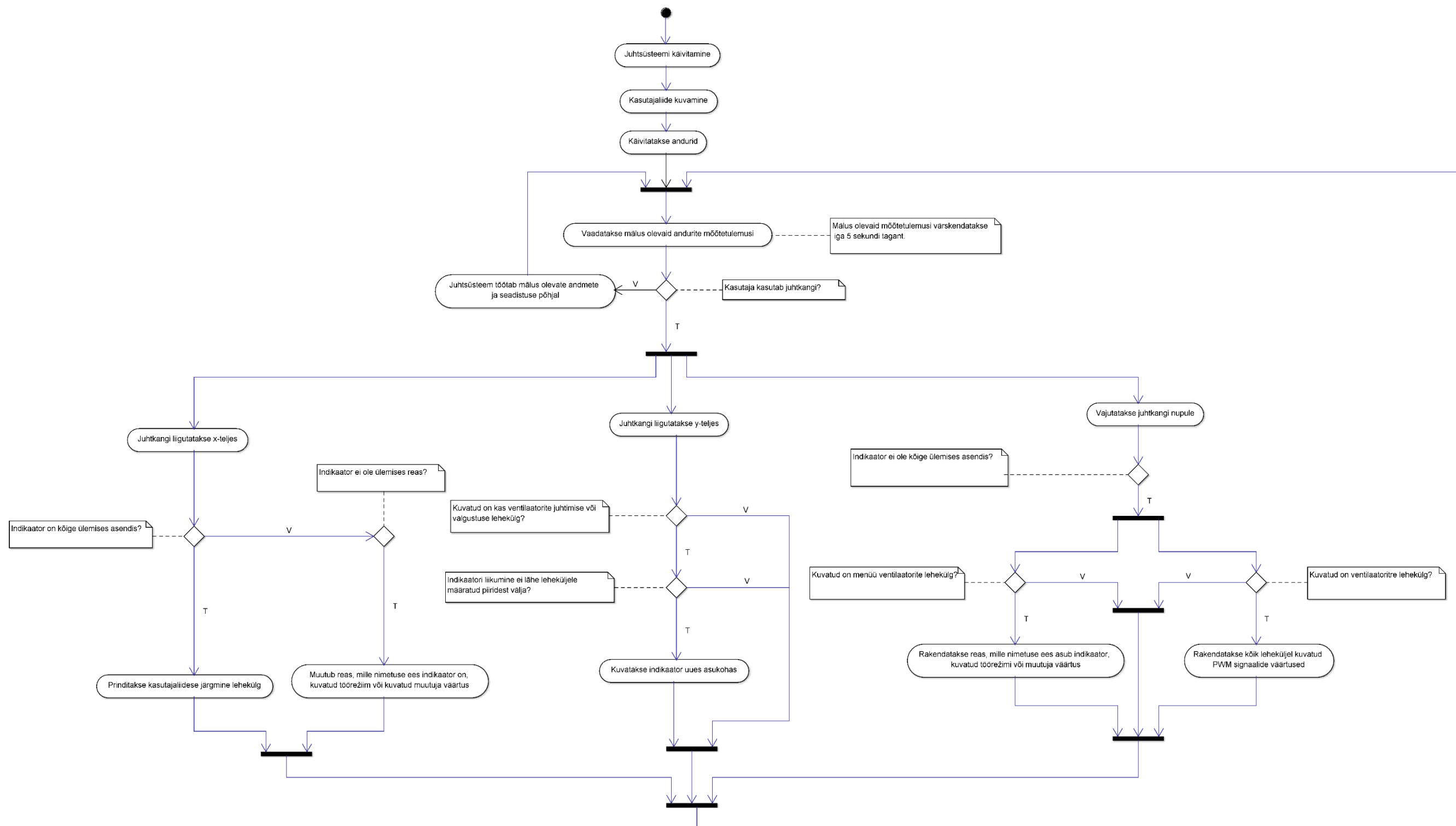


Lisa 2 Kasutatud komponentide nimekiri

Komponent	Kirjeldus/kommentaar	Hind	Kogus	Kokku
Meanwell LRS-75-12	12V toiteplokk	19	1	19
PS-MINI-DOWN-2596	DC/DC Step-Down moodul	5	1	5
Cryorig QF140 Performance	Ventilaator	10.41	4	41.64
Pro Power PPW00026	Arduino toite otsaga kaabel	1.71	1	1.71
Arduino Mega 2560 kloon		7.38	1	7.38
Arduino ekraan		7.79	1	7.79
18B20 temperatuuri anduri moodulid		2.72	6	16.32
Adafruit CCS811	VOC andur	18.09	1	18.09
Honeywell HPM115S0-XXX	Osakeste arvu mõõtja	36.22	1	36.22
MQ2	Suitsu andur	0.67	1	0.67
Omron G5Q-1EU 5DC	Relee	1.08	2	2.16
multicomp PRO 1N4007+	Diod	0.0624	2	0.1248
Vishay MBB02070C1001FCT00	1k Ω takisti	0.0391	9	0.3519
Cree LC503FPG1-30Q-A3-00001	LED roheline	0.257	3	0.771
Cree C503B-RCS-CW0Z0AA1	LED punane	0.154	1	0.154
multicomp PRO BC547B	Transistor releede lülitamiseks	0.118	2	0.236
ST MicroElectronics TIP120	Transistor LED riba jaoks	0.657	3	1.971
Klemmliist TB-001LC		0.2	6	1.2
Makettplaat (45x45mm)		0.8	1	0.8
Makettplaat (72*47mm)		1.6	1	1.6
Littlefuse 64700001003	Kaitsme hoidja	0.484	4	1.936
multicomp PRO MC000888	Kaitse 500mA	0.118	4	0.472
multicomp PRO MCMRA-1-KKFTC	Lüliti 1	0.795	1	0.795
Klahvlüliti ON-OFF 15A 250V	Lüliti 2	1.5	1	1.5
Sparkfun BOB-12009	Loogika pingemuundur	2.75	1	2.75

Adafruit MicroSD card breakout board	SD-kaardi moodul	7.56	1	7.56
Joystick moodul 2-axis XY		4	1	4
Kiuline montaaþijuhe Cu/PVC valge	Juhe AWG26 valge, hind meetri kohta	0.0675	30	2.025
Kõlarikaabel vask 2 x 0.35mm	Juhe AWG22 punane/must	0.38	20	7.6
Pro Power HS511-5M	Termokahanev isolatsioon	0.604	1.5	0.906
HellermannTyton 6TS3N BLACK	Juhtmete kaitseümbris	0.323	15	4.845
Molex 39-01-3043 - Mini-Fit Jr.	Molex 2x2 pistik emane	0.47	6	2.82
Molex 39-01-3103 - Mini-Fit Jr	Molex 2x5 pistik emane	0.35	1	0.35
Molex 39-01-2040 - Mini-Fit Jr.	Molex 2x2 pistik isane	0.1687	6	1.0122
Molex 39-01-2100 - Mini-Fit Jr.	Molex 2x5 pistik isane	0.2777	1	0.2777
Molex 39-00-0049	Molex isased crimpid	0.0615	25	1.5375
Molex 39-00-0047	Molex emased crimpid	0.066	25	1.65
Molex 47053-1000	Pistik ventilaatorite jaoks	0.3	4	1.2
Preciva Dupont Crimping Tool AWG26-18	Dupont pistikute komplekt koos tööriistaga	33.23	1	33.23
Juhtmekinnitus 5 - 7 mm	Karbis 100tk	3.5	1	3.5
Juhtmekinnitus 7 - 10 mm	Karbis 100tk	4.05	1	4.05
			Kokku:	243.16





Lisa 5 Juhtsüsteemi programmi väljatrukk

```
//Display librarys
#include <LCDWIKI_GUI.h> //Core graphics library
#include <LCDWIKI_KBV.h> //Hardware-specific library
// CCS811 sensor library (VOC eCo2 sensor)
#include <Adafruit_CCS811.h>
//Library for fan PWM controll
#include <TimerThree.h>
// Librarys for tempsensor
#include <OneWire.h>
#include <DallasTemperature.h>
Adafruit_CCS811 ccs;
//PID Library
#include <PID_v1.h>
//Library for HPMA115S0 particle sensor
#include <hpm115s0.h>

//the definiens of 16bit mode as follow:
//if the IC model is known or the modules is unreadable,you can use this
constructed function
LCDWIKI_KBV mylcd(ILI9486,40,38,39,-1,41); //model,cs,cd,wr,rd,reset

//define some colour values
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF
#define GREY 0x5AEB

#define MQ2pin (A2)
float MQ2sensorValue;
boolean smokeAlarm = LOW;

//Temperature sensor variables etc
#define ONE_WIRE_BUS 4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float SensorCount = 0;
float AverageTemp = 0;
float AverageTempSum = 0;
float TempReadings[6];
float sensor_value_refresh_delay = 5000;
float sensor_value_refresh = 0;

//HPMA115s0 variables
bool HPMA115s0status;
float particles25;
float particles10;
HPMA115S0 my_hpm(Serial3);

//CSS811 variables
int eCO2;
int TVOC;
```

```

// Joystick variables
int VRx = A0;
int VRy = A1;
int SW = 11;
int xPosition = 0;
int yPosition = 0;
int SW_state = 0;
int mapX = 0;
int mapY = 0;

//Front panel LED and buzzer variables

int LED1 = 44;
int LED2 = 45;
int LED3 = 46;
int LED4 = 13;
int Buzzer = 12;

// Variable for menu selection
int page_selector = 0;
int move_indicator = 0;
int indicator_position = 0;
int selected_ventilation_mode = 0;
int displayed_ventilation_mode = 0;
int y_max;
int mode_selected = 0;
int displayed_mode_change = 0;
int change_page = 0;
int mode_change = 0;
int mode_detail_change = 0;
int mode_value_change = 0;
int led_pwm_change1 = 0;
int led_pwm_change2 = 0;
int led_pwm_change3 = 0;

// Pins for controlling relays and fans
int FanRelay1 = 9;
int FanRelay2 = 10;
int IntakeFans = 2;
int OuttakeFans = 3;

// Setpoints
float displayed_TemperatureSetpoint = 20.0;
float TemperatureSetpoint = 20.0;
float displayed_fan_speed_setpoints[]={0,0};
float fan_speed_setpoints[]={0,0};
float fan_PWM_10bit_values[]={0,0};

//Parameters for fan PID control
double kp=2; //proportional parameter
double ki=5; //integral parameter
double kd=1; //derivative parameter
double maxTempDifference = 2;
double calculatedSetpointMin = 0;
double calculatedSetpointMax = 1023;
double tempDifference;
double calculatedSetpoint;
//initializing PID
PID myPID(&tempDifference, &calculatedSetpoint,
&maxTempDifference, kp, ki, kd, REVERSE);

```

```

//Pins for RGB LED strip PWMs
int Red = 8;
int Green = 7;
int Blue = 6;
//Variables for controlling led strip
int led_PWMs_displayed[3];
int led_PWMs_actual[3];

void setup() {
  // put your setup code here, to run once:

  Serial.begin (9600);
  //analogReference(DEFAULT);
  // Inputs and outputs

  //Display configuratio
  mylcd.Set_Rotation(3);
  mylcd.Init_LCD();
  mylcd.Set_Draw_color(random(255), random(255), random(255));
  mylcd.Fill_Screen(BLACK);

  // Joystick connections
  pinMode(VRx, INPUT);
  pinMode(VRy, INPUT);
  pinMode(SW, INPUT_PULLUP);

  // Timer for fan PWM
  Timer3.initialize(40); // Setting fan PWM to 40ms period (25kHz)

  // Relay pin configuration
  pinMode(FanRelay1, OUTPUT);
  pinMode(FanRelay2, OUTPUT);

  //Configuration for LED strip PWM pins
  pinMode(Red, OUTPUT);
  pinMode(Green, OUTPUT);
  pinMode(Blue, OUTPUT);

  // Temperature sensor setup
  sensors.begin();
  sensors.requestTemperatures();

  //PID setup
  myPID.SetMode(AUTOMATIC);
  myPID.SetOutputLimits(calculatedSetpointMin, calculatedSetpointMax);

  // HPM115s0 setup
  Serial3.begin(9600);
  my_hpm.stop_autosend();
  my_hpm.start_measurement();

  // CCS811 setup
  ccs.begin();
  ccs.available();

  pages();
  indicator();
  environment_sensors();
}

```

```

void loop(){
  // put your main code here, to run repeatedly:
  joystick_input();

  if( (millis() - sensor_value_refresh) > sensor_value_refresh_delay)
  {

    environment_sensors();
    sensor_value_refresh = millis();
  }

  if(millis() > 20000) // Non blocking delay to warm up sensor
  {
    MQ2_smokeAlarm();
  }

  menu();
  fan_control();
}

void joystick_input()
{
  xPosition = analogRead(VRx);
  yPosition = analogRead(VRy);
  SW_state = digitalRead(SW);
  delay(75);
}

void indicator_movement()
{
  delay(5);
  if(( 510 < xPosition < 530 ) && (page_selector == 0 || page_selector
== 3))
  {
    if(yPosition < 100)
    {
      indicator_positsion++;
      move_indicator = 1;
    }
    else if(yPosition > 1000)
    {
      indicator_positsion--;
      move_indicator = 1;
    }
  }
  else
  {
    move_indicator = 0;
  }

  if(indicator_positsion == (y_max)+1)
  {
    indicator_positsion = y_max;
  }
  else if ( indicator_positsion == -1)
  {
    indicator_positsion = 0 ;
  }
}

```

```

void indicator()
{
    if(indicator_positsion == 0)
    {
        mylcd.Set_Draw_color(WHITE);
        mylcd.Fill_Circle(15,30,5);
        mylcd.Set_Draw_color(BLACK);
        mylcd.Fill_Circle(15,100,5);
    }
    if(indicator_positsion == 1)
    {
        mylcd.Set_Draw_color(BLUE);
        mylcd.Fill_Circle(15,30,5);
        mylcd.Set_Draw_color(WHITE);
        mylcd.Fill_Circle(15,100,5);
        mylcd.Set_Draw_color(BLACK);
        mylcd.Fill_Circle(15,150,5);
    }
    if(indicator_positsion == 2)
    {
        mylcd.Set_Draw_color(WHITE);
        mylcd.Fill_Circle(15,150,5);
        mylcd.Set_Draw_color(BLACK);
        mylcd.Fill_Circle(15,100,5);
        mylcd.Set_Draw_color(BLACK);
        mylcd.Fill_Circle(15,200,5);
    }
    if(indicator_positsion== 3)
    {
        mylcd.Set_Draw_color(WHITE);
        mylcd.Fill_Circle(15,200,5);
        mylcd.Set_Draw_color(BLACK);
        mylcd.Fill_Circle(15,150,5);
    }
}

void FANS_PAGE()
{
    mylcd.Set_Text_Mode(0);
    mylcd.Set_Text_Back_colour(BLACK);
    mylcd.Set_Draw_color(BLUE);
    mylcd.Fill_Rectangle(0,0, 480, 55);
    mylcd.Set_Draw_color(WHITE);
    mylcd.Fill_Rectangle(0,65, 480,55);
    mylcd.Set_Text_colour(WHITE);
    mylcd.Set_Text_Back_colour(BLUE);
    mylcd.Set_Text_Size(3);
    mylcd.Print_String("FANS ", 30, 20);
    mylcd.Set_Text_colour(BLUE);
    int x = 30;
    int y = 90;
    mylcd.Print_String("MODE: ", x, y);
    mylcd.Print_String("<          >", 230, y);
    displayed_ventilation_mode = selected_ventilation_mode;
    displayed_TemperatureSetpoint = TemperatureSetpoint;
    for(int i=0; i < 2; i++)
    {
        displayed_fan_speed_setpoints[i] = fan_speed_setpoints[i];
    }
    displayed_mode();
}

```

```

}

void AIR_PAGE ()
{
    y_max = 0;
    mylcd.Set_Text_Mode(0);
    mylcd.Set_Text_Back_colour(BLACK);
    mylcd.Set_Draw_color(BLUE);
    mylcd.Fill_Rectangle(0,0, 480, 55); // mylcd.Fill_Rectangle(start X,
start Y, Start X-length X, Start Y- length Y);
    mylcd.Set_Draw_color(WHITE);
    mylcd.Fill_Rectangle(0,65, 480,55);
    mylcd.Set_Text_colour(WHITE);
    mylcd.Set_Text_Back_colour(BLUE);
    mylcd.Set_Text_Size(3);
    mylcd.Print_String("AIR QUALITY ", 30, 20);
    mylcd.Set_Text_colour(BLUE);
    int x = 30;
    int y = 90;
    mylcd.Print_String("PM 2.5: ", x, y);
    mylcd.Print_String("PM 10: ", x, y+50);
    mylcd.Print_String("eCO2: ", x, y+100);
    mylcd.Print_String("TVOC: ", x, y+150);
    mylcd.Print_String("PPM", 380, y);
    mylcd.Print_String("PPM", 380, y + 50);
    mylcd.Print_String("PPM", 380, y + 100);
    mylcd.Print_String("PPM", 380, y + 150);
    displaying_HPMA115s0_values();
    displaying_CSS811_values();
}

void TEMP_PAGE ()
{
    y_max = 0;
    mylcd.Set_Text_Mode(0);
    mylcd.Set_Text_Back_colour(BLACK);
    mylcd.Set_Draw_color(BLUE);
    mylcd.Fill_Rectangle(0,0, 480, 55); // mylcd.Fill_Rectangle(start X,
start Y, Start X-length X, Start Y- length Y);
    mylcd.Set_Draw_color(WHITE);
    mylcd.Fill_Rectangle(0,65, 480,55);
    mylcd.Set_Text_colour(WHITE);
    mylcd.Set_Text_Back_colour(BLUE);
    mylcd.Set_Text_Size(3);
    mylcd.Print_String("TEMPERATURE", 30, 20);
    mylcd.Set_Text_colour(BLUE);
    displaying_temp_menu();
    displaying_temp_values();
}

void RGB_PAGE ()
{
    y_max = 3;
    mylcd.Set_Text_Mode(0);
    mylcd.Set_Text_Back_colour(BLACK);
    mylcd.Set_Draw_color(BLUE);
    mylcd.Fill_Rectangle(0,0, 480, 55); // mylcd.Fill_Rectangle(start X,
start Y, Start X-length X, Start Y- length Y);
    mylcd.Set_Draw_color(WHITE);
    mylcd.Fill_Rectangle(0,65, 480,55);

```



```

mylcd.Set_Text_colour(WHITE);
mylcd.Set_Text_Back_colour(BLUE);
mylcd.Set_Text_Size(3);
mylcd.Print_String("RGB", 30, 20);
mylcd.Set_Text_colour(BLUE);
int x = 30;
int y = 90;
mylcd.Print_String("RED: ", x, y);
mylcd.Print_String("GREEN: ", x, y + 50);
mylcd.Print_String("BLUE: ", x, y + 100);
mylcd.Print_String("-", 280, y);
mylcd.Print_String("-", 280, y + 50);
mylcd.Print_String("-", 280, y + 100);
mylcd.Set_Draw_color(WHITE);
mylcd.Fill_Rectangle(0, 240, 480, 236);
for(int i=0; i < 3; i++)
{
    led_PWMs_displayed[i] = led_PWMs_actual[i];
}
mylcd.Set_Draw_color(led_PWMs_displayed[0], led_PWMs_displayed[1], led_PWM
Ms_displayed[2]);
mylcd.Fill_Rectangle(0, 320, 480, 240);
displaying_RGB_PWM_values();
}

void page_switching()
{

    if(xPosition > 1000  && (510 < yPosition < 530) && indicator_positsion
== 0 )
    {
        page_selectror++;
        change_page = 1;
    }
    else if(xPosition < 100 && (510 < yPosition < 530) &&
indicator_positsion == 0)
    {
        page_selectror--;
        change_page = 1;
    }
    else
    {
        change_page = 0;
    }

    if(page_selectror == 4)
    {
        page_selectror = 0;
    }
    else if ( page_selectror == -1)
    {
        page_selectror = 3 ;
    }

}

void pages ()
{
    if(page_selectror == 0)
    {
        FANS_PAGE();
    }
}

```

```

    }
    else if(page_selector == 1)
    {
        TEMP_PAGE();
    }
    else if(page_selector == 2)
    {
        AIR_PAGE();
    }
    else if(page_selector == 3)
    {
        RGB_PAGE();
    }
}

void displayed_mode()
{
    int x = 30;
    int y = 90;
    y_max = 1;
    mylcd.Set_Draw_color(BLUE);

    if(displayed_ventilation_mode == 0) // Ventilation is off
    {
        mylcd.Print_String("OFF", 320, y);
    }
    else if(displayed_ventilation_mode == 1) // Manual mode
    {
        mylcd.Print_String("MANUAL", 295, y);
    }
    else if(displayed_ventilation_mode == 2) // Temperature setpoint mode
    {
        mylcd.Print_String("SETPOINT", 275, y);
    }
    else if(displayed_ventilation_mode == 3) // Purge mode for venting the
    enclosure after 3D-printing is done
    {
        y_max = 1;
        mylcd.Print_String("CLEAR", 305, y);
    }
    displayed_mode_change = 0;
}

void mode_details()
{
    int x = 30;
    int y = 90;
    int y2 = 140;
    int x2 = 365;

    if(displayed_ventilation_mode == 1) // Manual mode
    {
        y_max = 3;
        mylcd.Print_String("INTAKE PWM: ", x, y + 50);
        mylcd.Print_String("OUTTAKE PWM: ", x, y + 100);
        mylcd.Print_String("-      +", 300, y + 50);
        mylcd.Print_String("-      +", 300, y + 100);

        for(int i = 0; i < 2; i++)

```

```

    {
        x2 = 365;
        if(displayed_fan_speed_setpoints[i] > 5)
        {
            x2 = 355;
        }
        if(displayed_fan_speed_setpoints[i] > 95)
        {
            x2 = 345;
        }
        mylcd.Print_Number_Int(displayed_fan_speed_setpoints[i],
x2, (y2+i*50), 0, ' ',10);
    }

}
else if(displayed_ventilation_mode == 2) // Temperature setpoint mode
{
    y_max = 2;
    y = y+50;
    mylcd.Print_String("TEMP.SETPOINT:", x, y);
    mylcd.Print_String("(-)      (+)", 280, y);
    mylcd.Print_Number_Float(displayed_TemperatureSetpoint, 1, 345, y,
'.', 0, ' ');
}
}

void mode_value_changes()
{
    int x = 30;
    int y = 90;
    int y2 = 140;
    int x2 = 365;

    if(displayed_ventilation_mode == 1) // Manual mode
    {
        for(int i = 0; i < 2; i++)
        {
            x2 = 365;
            if(displayed_fan_speed_setpoints[i] > 5)
            {
                x2 = 355;
            }
            if(displayed_fan_speed_setpoints[i] > 95)
            {
                x2 = 345;
            }
            mylcd.Print_Number_Int(displayed_fan_speed_setpoints[i],
x2, (y2+i*50), 0, ' ',10);
        }

    }
    else if(displayed_ventilation_mode == 2) // Temperature setpoint mode
    {
        y = y+50;
        mylcd.Print_Number_Float(displayed_TemperatureSetpoint, 1, 345, y,
'.', 0, ' ');
    }
}

```

```

}

void displaying_temp_menu()
{
    int x = 20;
    int y = 80;
    int x2 = 270;
    int y2 = 80;

    for(int i = 0; i < SensorCount; i++)
    {
        if(i < 3)
        {
            mylcd.Print_String("TEMP :    ", x, y);
            mylcd.Print_Number_Int(i+1, 95,y, 0, ' ',10);
            y = y + 50;
        }

        if(i > 2)
        {
            mylcd.Print_String("TEMP :    ", x2, y2);
            mylcd.Print_Number_Int(i+1, 345,y2, 0, ' ',10);
            y2 = y2 + 50;
        }
    }
    mylcd.Print_String("AVERAGE:    C", 20, 240);
}

void displaying_RGB_PWM_values()
{
    int x = 345;
    int y = 90;

    for(int i = 0; i<3; i++)
    {
        int x = 345;

        if(led_PWMs_displayed[i] > 5)
        {
            x = 335;
        }
        if(led_PWMs_displayed[i] > 95)
        {
            x = 325;
        }
        mylcd.Print_Number_Int(led_PWMs_displayed[i], x,y, 0, ' ',10);
        y = y+50;
    }
}

oid selecting_mode()
{
    if(indicator_positsion == 1 && page_selectror == 0)
    {
        if(xPosition > 1000 && (510 < yPosition < 530))
        {

```

```

displayed_ventilation_mode++;
displayed_mode_change = 1;
}
else if(xPosition < 100 && (510 < yPosition < 530) )
{
displayed_ventilation_mode--;
displayed_mode_change = 1;
}

if(SW_state == 0)
{
selected_ventilation_mode = displayed_ventilation_mode;
mylcd.Set_Draw_color(BLACK);
mylcd.Fill_Rectangle(0,120, 480, 320);
mode_detail_change = 1;
}
else
{
mode_detail_change = 0;
}
}
else
{
displayed_mode_change = 0;
}

if(displayed_ventilation_mode == 4)
{
displayed_ventilation_mode = 0;
}
else if (displayed_ventilation_mode == -1)
{
displayed_ventilation_mode = 3 ;
}
}

void temp_setpoint_value()
{
int x = 330;
int y = 140;

if(selected_ventilation_mode == 2 && page_selector == 0)
{
if(indicator_positsion == 2)
{
if(xPosition > 1000 && (510 < yPosition < 530))
{
if(displayed_TemperatureSetpoint < 40)
{
displayed_TemperatureSetpoint = displayed_TemperatureSetpoint
+ 0.5;

mylcd.Set_Draw_color(BLACK);
mylcd.Fill_Rectangle(x,y, x+90, y+40);
mode_value_change = 1;
}
}
else if(xPosition < 100 && (510 < yPosition < 530) )
{
if(displayed_TemperatureSetpoint > 20)

```

```

        {
            displayed_TemperatureSetpoint = displayed_TemperatureSetpoint
- 0.5;
            mylcd.Set_Draw_color(BLACK);
            mylcd.Fill_Rectangle(x,y, x+90, y+40);
            mode_value_change = 1;
        }

    }
    else
    {
        mode_value_change = 0;
    }

    if(SW_state == 0)
    {
        TemperatureSetpoint = displayed_TemperatureSetpoint;
    }
}
}
}

```

```

void Manual_fan_speed_setpoints()
{
    int x = 330;
    int y = 140;
    int x2 = x+90;
    int x3 = 365;

    if(selected_ventilation_mode == 1 && page_selectror == 0)
    {
        if(indicator_positsion > 1 )
        {
            int i = indicator_positsion - 2;

            if(xPosition > 1000 && (510 < yPosition < 530))
            {
                if(displayed_fan_speed_setpoints[i] < 100)
                {
                    displayed_fan_speed_setpoints[i] =
displayed_fan_speed_setpoints[i] + 5;
                    mylcd.Set_Draw_color(BLACK);
                    y= y + i*50;
                    int y2 = y+40;
                    int x2 = x+90;
                    mylcd.Fill_Rectangle(x,y, x2, y2);
                    mode_value_change = 1;
                }

            }
            else if(xPosition < 100 && (510 < yPosition < 530) )
            {
                if(displayed_fan_speed_setpoints[i] > 0)
                {
                    displayed_fan_speed_setpoints[i] =
displayed_fan_speed_setpoints[i] - 5;
                    mylcd.Set_Draw_color(BLACK);
                    y= y + i*50;
                    int y2 = y + 40;

```

```

        mylcd.Fill_Rectangle(x,y, x2, y2);
        mode_value_change = 1;
    }
}

if(SW_state == 0)
{
    fan_speed_setpoints[i] = displayed_fan_speed_setpoints[i];
}
}
else
{
    mode_value_change = 0;
}
}

void fan_control()
{
    if(selected_ventilation_mode == 0) // Ventilation is off
    {
        digitalWrite(FanRelay1, LOW);
        digitalWrite(FanRelay2, LOW);
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        Timer3.pwm(IntakeFans, 0);
        Timer3.pwm(OuttakeFans, 0);
    }
    else if(selected_ventilation_mode == 1) // Manual mode
    {
        manual_fan_speed();
    }
    else if(selected_ventilation_mode == 2) // Temperature setpoint mode
    {
        setpoint_fan_controll();
    }
    else if(selected_ventilation_mode == 3) // Purge mode for venting the
    enclosure after 3D-printing is done
    {

    }
}

void manual_fan_speed()
{
    for(int i = 0; i < 2; i++)
    {
        fan_PWM_10bit_values[i] = ((1023*fan_speed_setpoints[i])/100);
    }

    if(fan_speed_setpoints[0] == 0)
    {
        digitalWrite(FanRelay1, LOW);
        digitalWrite(LED1, LOW);
        Timer3.pwm(IntakeFans, 0);
    }
    else
    {

```

```

    digitalWrite(FanRelay1, HIGH);
    analogWrite(LED1, 10);
    Timer3.pwm(IntakeFans, fan_PWM_10bit_values[0]);
}

if(fan_speed_setpoints[1] == 0)
{
    digitalWrite(FanRelay2, LOW);
    digitalWrite(LED2, LOW);
    Timer3.pwm(OuttakeFans, 0);
}
else
{
    digitalWrite(FanRelay2, HIGH);
    analogWrite(LED2, 10);
    Timer3.pwm(OuttakeFans, fan_PWM_10bit_values[1]);
}
}

void setpoint_fan_controll()
{
    tempDifference = AverageTemp - TemperatureSetpoint;
    myPID.Compute();

    if(calculatedSetpoint == 0)
    {
        digitalWrite(FanRelay1, LOW);
        digitalWrite(FanRelay2, LOW);
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
    }
    else
    {
        digitalWrite(FanRelay1, HIGH);
        digitalWrite(FanRelay2, HIGH);
        analogWrite(LED1, 10);
        analogWrite(LED2, 10);
        Timer3.pwm(IntakeFans, calculatedSetpoint);
        Timer3.pwm(OuttakeFans, calculatedSetpoint);
    }
}

void selecting_RGB_PWMs()
{
    int x = 345;
    int y = 90;
    int x2 = 300;
    int y2 = 90;
    int x3 = 400;
    int y3 = 130;
    int i = 0;
    int j = 0;

    if(page_selectror == 3)
    {
        if (indicator_positsion > 0)
        {

```



```

i = indicator_positsion - 1;
y = y + 50*i;
y2 = y2 + 50*i;
y3 = y3 + 50*i;

if((510 < yPosition < 530))
{
    if(xPosition > 1000)
    {
        if(led_PWMs_displayed[i] < 255)
        {
            led_PWMs_displayed[i]= led_PWMs_displayed[i] + 5;
            mylcd.Set_Draw_color(BLACK);
            mylcd.Fill_Rectangle(x2,y2, x3, y3);

            if(led_PWMs_displayed[i] > 5)
            {
                x = 335;
            }
            if(led_PWMs_displayed[i] > 95)
            {
                x = 325;
            }

            mylcd.Print_Number_Int(led_PWMs_displayed[i], x, y, 0, '
',10);

            mylcd.Set_Draw_color(led_PWMs_displayed[0],led_PWMs_displayed
[1],led_PWMs_displayed[2]);
            mylcd.Fill_Rectangle(0,320, 480, 240);
        }
    }
    else if(xPosition < 100)
    {
        if(led_PWMs_displayed[i] > 0)
        {
            led_PWMs_displayed[i] = led_PWMs_displayed[i] - 5;
            mylcd.Set_Draw_color(BLACK);
            mylcd.Fill_Rectangle(x2,y2, x3, y3);

            if(led_PWMs_displayed[i] > 5)
            {
                x = 335;
            }
            if(led_PWMs_displayed[i] > 95)
            {
                x = 325;
            }

            mylcd.Print_Number_Int(led_PWMs_displayed[i], x, y, 0, '
',10);

            mylcd.Set_Draw_color(led_PWMs_displayed[0],led_PWMs_displayed
[1],led_PWMs_displayed[2]);
            mylcd.Fill_Rectangle(0,320, 480, 240);
        }
    }
}
}
}
}
}

```

```

    if(SW_state == 0)
    {
        for(int i=0; i<3; i++)
        {
            led_PWMs_actual[i] = led_PWMs_displayed[i];
        }
        Applying_RGB_PWM();
    }
}

void Applying_RGB_PWM()
{
    analogWrite(Red, led_PWMs_actual[0]);
    analogWrite(Green, led_PWMs_actual[1]);
    analogWrite(Blue, led_PWMs_actual[2]);
}

void environment_sensors()
{
    sensors.requestTemperatures();
    SensorCount = sensors.getDeviceCount();

    if(SensorCount > 0)
    {
        temperature_sensors();
    }

    if(page_selector == 1)
    {
        displaying_temp_values();
    }

    honeywell_particle_sensor();

    if(millis() > 20000) // Non blocking delay to warm up sensor
    {
        sensor_CCS811();
    }
}

void temperature_sensors()
{
    for(int i = 0; i < SensorCount; i++)
    {
        TempReadings[i]=sensors.getTempCByIndex(i);
        AverageTempSum =AverageTempSum + TempReadings[i];
    }

    AverageTemp = (AverageTempSum/SensorCount);
}

void honeywell_particle_sensor()

```

```

{
  HPMA115s0status = my_hpm.read(&particles25,&particles10);
  if(page_selector == 2)
  {
    if (HPMA115s0status == 1)
    {
      displaying_HPMA115s0_values();
    }
  }
}

void MQ2_smokeAlarm()
{
  MQ2sensorValue = analogRead(MQ2pin); // read analog input pin 0

  if(MQ2sensorValue > 300 && smokeAlarm == LOW)//Smoke detected
  {
    smokeAlarm = HIGH;
  }

  if(smokeAlarm == HIGH)
  {
    analogWrite(LED4,1023);
    tone(Buzzer, 3500, 100);
  }
  else if(SW_state == HIGH && smokeAlarm == HIGH)
  {
    digitalWrite(LED4, LOW);
    smokeAlarm = LOW;
  }
}

void sensor_CCS811()
{
  if(ccs.available())
  {
    if(!ccs.readData())
    {
      eCO2 = ccs.geteCO2();
      TVOC = ccs.getTVOC();
    }
    else
    {
      while(1);
    }

    if(page_selector == 2)
    {
      displaying_CCS811_values();
    }
  }
}

void displaying_temp_values()
{

```

```

int x = 135;
int y = 80;
int x2 = 385;
int y2 = 80;

for(int i = 0; i < SensorCount; i++)
{
    if(i < 3)
    {
        mylcd.Set_Draw_color(BLACK);
        mylcd.Fill_Rectangle(x,y,x+80, y+45);
        mylcd.Print_Number_Float(TempReadings[i], 1, x, y, '.', 0, '
');
        y = y + 50;
    }

    if(i > 2)
    {
        mylcd.Set_Draw_color(BLACK);
        mylcd.Fill_Rectangle(x2,y2,x2+80, y2+45);
        mylcd.Print_Number_Float(TempReadings[i], 1, x2, y2, '.', 0, '
');
        y2 = y2 + 50;
    }
}
mylcd.Set_Draw_color(BLACK);
mylcd.Fill_Rectangle(170,240,250,300);
mylcd.Print_Number_Float(AverageTemp, 1, 170, 240, '.', 0, ' ');
AverageTempSum = 0;
}

void displaying_HPMA115s0_values()
{
    int x = 290;
    int y = 90;
    mylcd.Set_Draw_color(BLACK);
    mylcd.Set_Text_Back_colour(BLACK);
    mylcd.Set_Text_colour(BLUE);
    mylcd.Fill_Rectangle(270,y,370, y+100);
    mylcd.Print_Number_Float(particles25, 1, x, y, '.', 0, ' ');
    mylcd.Print_Number_Float(particles10, 1, x, y+50, '.', 0, ' ');
}

void displaying_CSS811_values()
{
    int x = 290;
    int y = 190;
    mylcd.Set_Draw_color(BLACK);
    mylcd.Fill_Rectangle(290,y,370, y+100);
    mylcd.Print_Number_Int(eCO2, x,y, 0, ' ',10);
    mylcd.Print_Number_Int(TVOC, x,y+50, 0, ' ',10);
}

```