

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl Paabut 206237IACB

Mõõteseadme kasutajaliidese disain ja toitelahendus videohilistuse mõõtmiseks

Bakalaureusetöö

Juhendaja: Priit Roosipuu
MSc

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Karl Paabut

13.05.2024

Annotatsioon

Antud lõputöö käsitleb videopildi viite mõõtmis süsteemi seadmete kasutajaliidese, toitelahenduse ja korpuse arendust. Ülesandeks on luua prototüüp seadmele kasutajaliides, toitelahendus ja korpus.

Uurisin turul olemasolevaid lahendusi ja nende sobivust lõputöö nõuetega. Turul olevad lahendused ei täitnud kõiki nõudeid. Otsustasin mudelleerida korpuse ise ja kasutada selle loomiseks 3D printereid. Korpus peab olema modulaarne, st. korpus peab ära mahutama 3 eri mikrokontrollerit. Aruandes toon välja ka korpuse ja seadme kasutajaliidese edasi arendusvõimalused. Korpust saab muuta kompaktsemaks vastavalt sellele, millist mikrokontrollerit kasutatakse.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 6 peatükki, 35 joonist, 1 tabelit.

Abstract

User Interface Design and Power Solution for a Video Latency Measurement Device

This thesis aims to create a user interface, a power solution, and a case for the prototype devices of a video latency measurement system. The video latency measurement system consists of two devices, that are identical both in appearance and in functionality. The measurement system works by having one device blink an LED in front of a videocamera and the second device is in front of a computer monitor or screen with a light sensor. Both devices are in sync with the same NTP server, so when the device with the LED makes the LED blink every even second, the device with the sensor can calculate the time difference between the LED blinking and the perception of the LED blinking on the computer monitor.

Since both of the devices need to be able to blink and use the sensor, the device has two states that the user can switch between. One state is called the „transmission state“, where the device blinks an LED and the other is the „receiving state“ where the device waits for the sensor to get a signal and calculates the latency. A user interface was created for three microcontrollers: the Raspberry Pi 4, the Raspberry Pi 5, and the ESP32-POE-ISO. All of the microcontrollers have an OLED screen connected to them that displays recent measuring data. Both of the Raspberries also have a graphical user interface that can be accessed through VNC. The graphical user interface displays a line graph and histogram of the measured data, giving the operator a first glance at the outcome of the measurement.

Regarding powering the devices, the author decided to go with PoE. PoE powers the device and transmits data through one cable eliminating the need for multiple input ports. All of the microcontrollers support PoE, with the ESP32-POE-ISO needing no extra hardware and both of the Raspberry Pi-s needing a PoE hat to be able to operate using PoE.

The suitability of already existing cases on the market was thoroughly investigated, but the examples were found lacking. The requirements of this project are quite specific, thus ruling out any prebuilt microcontroller case. It was decided that 3D printing was the

most cost-effective way to create a custom enclosure to house the microcontroller and other components.

Also outlined in this thesis are the future avenues of further development, including building a power switch that allows a device to accept multiple power sources (USB-C and PoE) and switch between them without itself. Another direction of further development that can be done is regarding the enclosure of both of the devices. When a definitive decision has been made regarding the choice of microcontroller, the size of the case can be significantly decreased making the enclosure cheaper and easier to produce. The case also needs to be made weatherproof, so applying an IP standard to the case is most definitely in order.

The thesis is in estonian and contains 37 pages of text, 7 chapters, 35 figures, 1 table.

Lühendite ja mõistete sõnastik

GPIO	<i>General-purpose Input Output.</i> Viik
I2C	<i>Inter-integrated circuit.</i> Kommunikatsiooni protokoll
LCD	<i>Liquid crystal display.</i> Ekraani tüüp
LED	<i>Light emitting diode.</i> Valgusdiod
NTP	<i>Network time protocol.</i> Protokoll
OLED	<i>Organic light emitting diode.</i> Ekraani tüüp
PD	<i>Powered device.</i> Toidetav seade
PLA	<i>Polylactic acid.</i> Filamendi tüüp
PoE	<i>Power over Ethernet.</i> Toitestandard
PSE	<i>Power sourcing equipment.</i> Toidet andev seade
RFB	<i>Remote frame buffer.</i> Protokoll
SPI	<i>Serial Peripheral Interface.</i> Kommunikatsiooni protokoll
VNC	<i>Virtual network computing.</i> Protokoll

Sisukord

1 Sissejuhatus	13
2 Ülevaade vaadeldavatest tehnoloogiate kasutusvaldkondadest.....	15
2.1 Power over Ethernet	15
2.1.1 Aktiivne PoE	16
2.1.2 Passiivne PoE	16
2.2 Virtual Network Computing	16
3 Kasutajaliides	18
3.1 Puutetundlik ekraan	18
3.2 SSD1306 OLED ekraan.....	19
3.3 Virtual Network Computing (VNC).....	19
3.4 Olekud	20
3.5 Kasutajaliidese koodid.....	20
3.5.1 Mikrokontrollerite Raspberry Pi 4 ja Raspberry Pi 5 koodid.....	20
3.5.2 Mikrokontrollerite Raspberry Pi 4 ja Pi 5 koodi erinevused.....	33
3.5.3 Mikrokontrolleri ESP32 kood	33
3.6 Kasutajaliidese testimise skeemid	35
4 Toitesüsteem.....	38
4.1 Seadmete toide.....	38
4.1.1 Mikrokontrolleri Raspberry Pi PoE <i>hat</i> -i tööpõhimõte	39
4.1.2 Mikrokontrolleri Raspberry Pi 5 PoE lahendus.....	39
5 Korpuse prototüüp	42
5.1 Modulaarsus.....	42
5.2 Turul olevad lahendused.....	42
5.2.1 Mikrokontrollerite individuaalsed korpused	42
5.3 3D printimine.....	44
5.3.1 3D printer Ender 3 V2	44
5.3.2 3D modelleerimine	44

5.3.3 Korpuse kokkupanek	44
5.3.4 Mikrokontrollerite hoidikud	46
6 Edasiarendusvõimalused	49
6.1.1 Toitelüliti	49
6.1.2 Korpuse kohandamine	49
6.1.3 Korpuse ilmastikukindlus	49
7 Kokkuvõte	50
8 Kasutatud kirjandus	51
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	55
Lisa 2 – Faili salvestamise algoritm	56
Lisa 3 – Graafiku kustutamise algoritm	57
Lisa 4 – Valgusdiodi vilgutamise algoritm	58
Lisa 5 – Raspberry Pi-de vastuvõtmisolek	59
Lisa 6 – ESP32 setup algoritm	60
Lisa 7 – Etherneti pistiku kinnitus	61
Lisa 8 – Kasutajaliideste koodid.....	61

Jooniste loetelu

Joonis 1. Mõõtesüsteemi kirjeldus.	13
Joonis 2. VNC server [3].	16
Joonis 3. SSD1306 LCD ekraan.	19
Joonis 4. Graafilise kasutajaliidese avavaade.	21
Joonis 5. Programmi käivitamise skeem.	21
Joonis 6. Saatmisolek graafilisel kasutajaliidesel.	22
Joonis 7. Saatmisolek OLED ekraanil.	22
Joonis 8. Saatmisoleku algoritm.	23
Joonis 9. Sünkronisatsiooni protsess.	24
Joonis 10. Vastuvõtmisolek OLED ekraanil.	25
Joonis 11. Vastuvõtmisolek graafilisel kasutajaliidesel.	26
Joonis 12. Vastuvõtmisoleku algoritm.	27
Joonis 13. Eraldi lõimel sensori funktsioon.	28
Joonis 14. Graafiku genereerimisfunktsioon.	29
Joonis 15. Simuleeritud tulemustega joongraafik.	30
Joonis 16. Simuleeritud tulemustega histogramm.	31
Joonis 17. 75 mõõtetulemusega joongraafik.	32
Joonis 18. 75 mõõtmistulemusega histogramm.	33
Joonis 19. ESP32-e olekute valik.	34
Joonis 20. ESP32-e saatmisolek.	35
Joonis 21. ESP32-e vastuvõtmisolek.	35
Joonis 22. Raspberry Pi-de kasutajaliideste skeem.	36
Joonis 23. ESP32 kasutajaliidese skeem.	37
Joonis 24. Raspberry Pi 4 koos PoE+ hat-iga.	38
Joonis 25. Raspberry Pi PoE toide [25], [26].	39
Joonis 26. Raspberry Pi-de PoE pinnide võrdlus [27], [29], [25].	40
Joonis 27. Raspberry Pi 5 PoE lahenduse skeem.	41
Joonis 28. Raspberry Pi 4b turul olev korpus [30].	43
Joonis 29. L-tähe kujuline seinatükk.	45

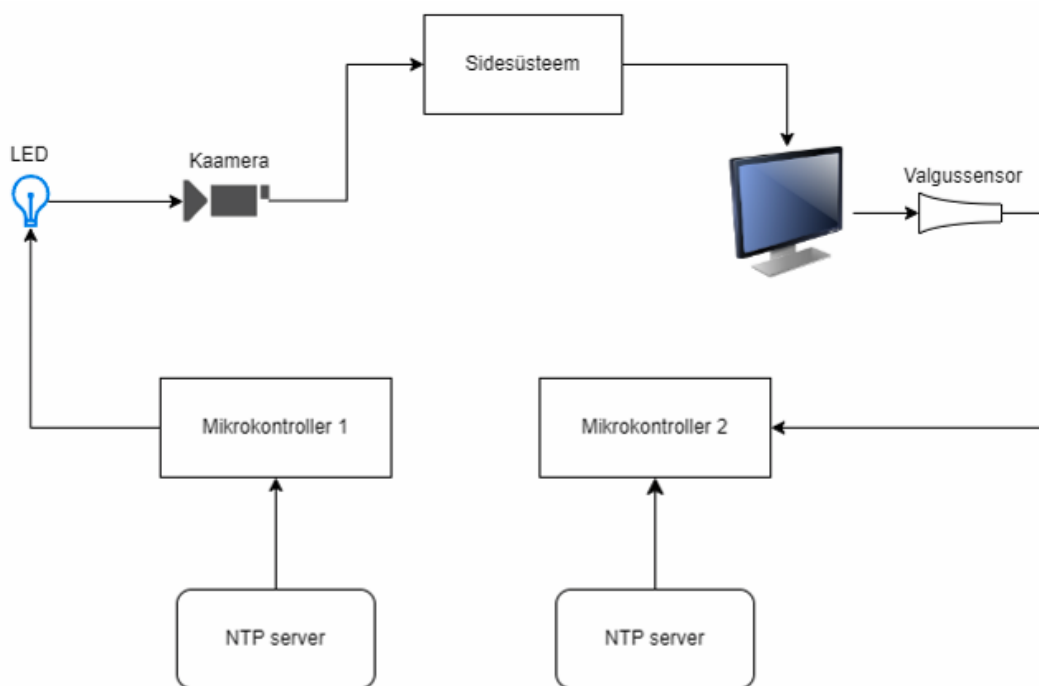
Joonis 30. Korpuse Etherneti pordi poolne vaade.	45
Joonis 31. Korpuse sensori ja valgussensori poolne vaade.	46
Joonis 32. Korpuse pealtvaade.	46
Joonis 33. ESP32-POE-ISO hoidik.	47
Joonis 34. RPi hoidik.....	47
Joonis 35. PoE <i>hat</i> -i hoidik.	48

Tabelite loetelu

Tabel 1. Aktiivse PoE tüübid.....	15
-----------------------------------	----

1 Sissejuhatus

Lõputöö on osa projektist, mis loob prototüüp videohilistuse mõõtmis süsteemi. Projekti eesmärk on luua süsteem, mis koosneb kahest mikrokontrollerist, LED-ist (*Light Emitting Diode*) ehk valgusdiodist ja sensorist [1]. Mikrokontroller 1 on mõõdetava kaamera ja mikrokontroller 2 on videopilti kuvava monitori juures. Mikrokontroller 1 vilgutab kaamera ees valgusdiodi ja Mikrokontroller 2 küljes on valgussensor, mis ootab monitoril valgusdiodi vilkumist. Selleks, et videopildilt saadud viidet arvestada, peavad mikrokontrollerite kellad olema sünkroonis. Mõlemad mikrokontrollerid küsivad aega samalt NTP (*Network Time Protocol*) serverilt. Valgusdiod vilgub ette määratud ajahetkel, kuna mikrokontrollerid on sünkroonis, saab Mikrokontroller 2 arvutada signaali saatmise ajatempli ja signaali vastuvõtmise ajatempli vahelise aja. Mõõdetava kaamera ja peakontoris oleva monitori vahel on kodeerimised ja muud protsessid, mis moodustavad eraldi sidesüsteemi. Sidesüsteem ei ole antud töö skoobis oluline ja seda antud lõputöö ei käsitle (Joonis 1).



Joonis 1. Mõõtesüsteemi kirjeldus.

Mikrokontrollerid, mida projekti käigus uuritakse ja võrreldakse on: Raspberry Pi 4, Raspberry Pi 5, ESP32-POE-ISO ja Arduino Mega. Arduino Megat projekti käigus uuriti, kuid ei osutunud valituks seega antud lõputöö raames seda mikrokontrollerit ei arvestata.

Käesoleva bakalaureuse lõputöö eesmärgiks on luua prototüüp süsteemile kasutajaliides, toitelahendus ja korpused. Mikrokontrollereid millega peab arvestama on 3, ehk lahendus peab olema modulaarne. Seadmeid süsteemis on kaks, seega korpuseid peab olema kaks ja nad peavad olema identsed nii välimuselt kui ka funktsionaalsuselt.

Algtingimused: Mikrokontrolleri jahutus ja soojendus ei ole kriitilised, sest seadet kasutatakse välistes tingimustes lühikest aega. Seadmeid on kaks ja nende korpused on identsed. Korpusele IP standardi määramist antud töö ei käsitle.

Uurimisküsimused, millele see lõputöö soovib vastata, on järgnevad:

- Kuidas luua väline kasutajaliides?
- Kuidas lahendada toitelahendus?
- Millised on turul pakutavad korpuse lahendused?
- Kuidas luua mitme mikrokontrolleriga sobiv korpuse lahendus?
- Kuidas sobitada kasutajaliidest ja korpust kolme erineva mikrokontrolleriga?

Luu prototüüp mõttesüsteemi seadmetele kasutajaliideseid ja korpused, et seadmeoperaator saaks mõõtmisi läbi viia ja esmaseid tulemusi hinnata. Lisaks peab tagama ka seadmetele stabiilse toite.

Süsteemi ülesanne on mõõta videopildi kaadri jäädvustamisest kuni monitorile jõudmise vahelist viidet. Kaamera ja monitori vahel on sidesüsteem, mida antud lõputöös ei uurita sest autoril puudub sellele ligipääs. [1]

2 Ülevaade vaadeldavatest tehnoloogiate kasutusvaldkondadest

Peamised kasutusel olevad tehnoloogiad on *Power over Ethernet (PoE)* ja *Virtual Network Computing (VNC)*. PoE on ideaalne toite ja infoedastamis lahendus ning VNC võimaldab mikrokontrollerile kaugelt ligi saada ilma, et mikrokontroller ja operaatori seade samas võrgus oleks [2], [3].

2.1 Power over Ethernet

Power over Ethernet, edaspidi PoE on standard, mis annab läbi *Ethernet*'i kaabli toidet. PoE jaguneb aktiivseks ja passiivseks. Aktiivne PoE jaguneb neljaks tüübiks, mille funktsionaalsust katavad kolm IEEE standardit. Kõige vanem PoE tüüp on tüüp 1, nimetatakse ka lihtsalt PoE-ks, mis kasutab IEEE 802.3af standardit. Tüüp 2, tuntud ka kui PoE+, kasutab IEEE 802.3at standardit. Tüübid 3 ja 4, tuntud ka kui PoE++ ja High-Power PoE, kasutavad mõlemad IEEE 802.3bt standardit (Tabel 1) . PoE kirjeldamiseks kasutatakse termineid *powered device*, edaspidi PD ja *power sourcing equipment*, edaspidi PSE [2].

Etherneti kaabel koosneb neljast juhtmepaarist. Tüübid 1 ja 2 kasutavad kahte paari toite edastamiseks ja kahte paari andmete edastamiseks. Tüübid 3 ja 4 kasutavad nii toite kui andmete edastamiseks kõiki nelja paari [4].

Tabel 1. Aktiivse PoE tüübid [2].

Standard	802.3af Tüüp 1 „PoE“	802.3at Tüüp 2 „PoE+“	802.3bt Tüüp 3 „PoE++“	802.3bt Tüüp 4 „High-Power PoE“
PD-ni jõudev võimsus	12,95 W	25,50 W	51 W	71 W
Maksimaalne PSE-i väljaandev võimsus	15,40 W	30,0 W	60 W	90 W

2.1.1 Aktiivne PoE

Aktiivse ehk *Active* PoE puhul toimub PSE ja PD-i e. klient seadme vahel infovahetus, mille tulemusel teeb PSE kindlaks, kas PD toetab PoE-d, seejärel tuvastab PSE, mis toite standardit seade toetab ning alles siis saadab seadmele toite peale [5]. Aktiivne PoE on enamlevinud PoE tüüp, antud lõputöös kasutab autor tüüp 1 ja tüüp 2 PoE-d ehk PoE-d ja PoE+-i.

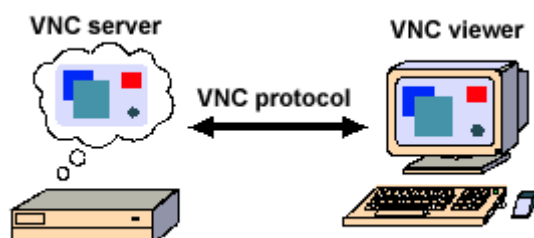
2.1.2 Passiivne PoE

Passiivne PoE ehk *Passive* PoE puhul ei toimu PSE ja PD vahel infovahetust, vaid PSE saadab PD-le kohe interneti ja toite peale. Kui ühendatud PD ei toeta antud pinget, siis seade võib läbi põleda. Passiivne PoE on enamasti kasutusel välitingimustes olevates seadmetes, nt turvakaamerad. Lõputöös antud PoE tüüpi ei kasutata, sest vaadeldavad mikrokontrollerid seda ei toeta [5], [6].

2.2 Virtual Network Computing

Virtual Network Computing, edaspidi VNC on lihtne graafilise töölaua jagamis protokoll, mis põhineb RFB (*remote frame buffer*) protokollil. *Frame buffer* on puhver või kogus mälu, millesse on salvestatud iga ekraani piksli värvi väärtus. RFB puhul edastatakse kliendile pidevalt andmeid, mis kirjeldavad ekraani pikslite väärtust. Videokaart viib ekraani RFB-ga vastavusse [7].

VNC abil on võimalik sisestada hiire ja klaviatuuri sisendeid teise arvutisse üle võrgu. VNC on lihtne protokoll, sellest tulenevalt saab seda kasutada peaaegu kõikidel operatsiooni süsteemidel [3].



Joonis 2. VNC server [3].

Antud lõputöös kasutatakse programmi nimega RealVNC, valisin selle programmi sest see on Raspberry enda poolt soovitatud. RealVNC lubab sätestada mikrokontrolleril VNC serveri (Joonis 2), mille abil saab seadme külge ühenduda ilma, et peaks olema samas kohtvõrgus. Ühe serveri küljes võib korraga olla mitu klienti, kes saavad kõik seadet mõjutada. Antud programmil on ka mitmeid kasutajamugavus hüvesid (nt. ekraani salvestus ja faili edastus), mis muutsid kasutajaliidese arenduse ja kasutajaliidese kasutamise mugavamaks [8].

3 Kasutajaliides

Lõputöö pea eesmärk oli luua prototüüp mõotesüsteemile kasutajaliides, mille abil seadme operaator saab veenduda, et seade on töökorras ning näha mõõtetulemusi. Korpuse küljes on 0.96 tolline OLED (*Organic Light Emitting Diode*) ekraan (SSD1306), mis näitab viimast mõõtmist, mõõtmiste keskväärtust ja standardhälvet (standardhälve ehk σ) [9]. Graafiku nägemiseks peab kasutama VNC ühendust, et seadmele kaugelt ligi pääseda.

Algselt oli kavas lahendada kasutajaliides ühe puutetundliku ekraaniga, kuid autor valis vale ekraani mis ei ühildunud kasutusel olevate seadmetega. Sobiliku ekraani omandamine oleks võtnud liiga kaua aega, seega otsustati kasutada VNC-d. Kasutajaliidesel peab olema kaks olekut, saatmisolek ja vastuvõtmisolek.

Saatmisolek on olek, kus seade vilgutab igal paarissekundil valgusdiodi. Vastuvõtmisolekus ootab seadme sensor signaali ning näitab nii OLED ekraanil kui ka graafilisel kasutajaliidesel mõõtmistulemustega seonduvat infot ja sünkronisatsiooni staatust. Mõõtmistulemustega seonduv info on viimane mõõtmistulemus, viimase 30 mõõtmistulemuse keskväärtus ja viimase 30 mõõtmistulemuse standardhälve.

Lisaks joonistatakse graafilisel kasutajaliidesel joongraafik ja histogramm. Joongraafikut uuendatakse jooksvalt, ühel graafikul on alati viimase 30 mõõtmise tulemus. Kui 30 mõõtmistulemust on käes, tekib joongraafiku kõrvale histogramm, mis näitab viimase 30 mõõtmistulemuse jaotust.

3.1 Puutetundlik ekraan

Ekraani valikul lähtuti LCD (*Liquid Crystal Display*) kontrolleri mudelist, taheti valida ekraan mis sobituks kõigi kasutusel olevate mikrokontrolleritega.

LCD kontrolleriiks valiti ILI9341, mis oli antud otstarbeks kohane. Ekraani valikul tegi autor vea, valides Waveshare 16498 ekraani [10]. Luges toote kirjeldust ja eeldati, et seadet on võimalik kasutada kas 8080 paralleel liidese või SPI-ga (*Serial Peripheral*

Interface). Ekraani tootja poolt avaldatud skeemi kohaselt ühildub ekraani LCD moodul STM mikrokontrolleri külge kasutades 8080 paralleel liidest ja puutetundlik moodul ühendub STM-i külge kasutades SPI-d [11]. Järeldati, et Waveshare 16498 on otstarbeks ehitatud ekraan, mis ühildub ainult STM mikrokontrolleriga.

Kirjutati Waveshare-ile meili, et saada lisainfot ekraani ja Raspberry Pi-de liidestamise kohta. Waveshare soovitas autoril omandada uus ekraan, mis on mõeldud Raspberry Pi-ga liidestamiseks.

Ekraan ei ühildu ESP32-ga, sest ESP32-l ei ole piisavalt GPIO (*General Purpose Input Output*) väljundeid, et ekraaniga ühenduda. Peamine põhjus, miks antud ekraani ei saa kasutada ei Raspberry Pi-dega ega ESP32-ga on vastava draiveri tarkvara puudumine.

3.2 SSD1306 OLED ekraan

SSD1306 on 0,96 tolline ühevärviline OLED ekraan, mis ühendub juhtseadmetega üle I2C liidese. Antud ekraan valiti tänu puutetundliku ekraani mitteühildumisele. SSD1306 (Joonis 3), on oma ehituselt väga robustne ja ekraan oli saadaval [9].



Joonis 3. SSD1306 LCD ekraan.

3.3 Virtual Network Computing (VNC)

Selleks, et näha mõõtmistulemuste graafikuid ja vahetada mõõteseadme olekut, peab seadme operaator kasutama VNC-d. Mikrokontrollerid millele on võimalik kaugelt ligi pääseda on Raspberry Pi 4 ja Raspberry Pi 5. Mõlemal mikrokontrolleril on seadistatud VNC server, mis käivitub automaatselt kui Pi sisse lülitada. Selleks, et VNC serveri külge ühendada, peab Pi-l olema ühendus Internetiga [12].

Kontrolleritele on võimalik ligipääseda kasutades seadet, millel on peale installitud RealVNC Viewer, edaspidi *viewer*. Lisaks kasutajaliidese kasutamisele on võimalik läbi

viewer'i pääseda ligi SD kaardile salvestatud logi failidele, ning neid mugavalt otse operaatori seadmesse eksportida [13].

RealVNC ainus puudujääk on kahe nädalane katseaeg, antud programmi saab tasuta kasutada kaks nädalat peale konto loomist, katseaja lõppedes tuleb osta tasuline pakett. Arenduse käigus tehti endale mitme meiliaadressiga mitu eri kontot.

3.4 Olekud

Kasutajaliidesel on kaks eri olekut, saatmisolek ja vastuvõtmisolek.. Seega saab sama seadet kasutada nii saatja kui ka vastuvõtjana. Mõlemas olekus on võimalik nii graafilisel kasutajaliidesel kui OLED ekraanil näha sünkronisatsiooni staatust ja seadme hetke olekut, ehk mida seade parasjagu teeb. Graafilisel kasutajaliidesel on näha joongraafikut ja histogrammi. ESP32-le puudub kaugelt ligipääsemis võimalus, seega ESP32-el puudub graafiline kasutajaliides. Kogu informatsiooni näidatakse OLED ekraanil.

3.5 Kasutajaliidese koodid

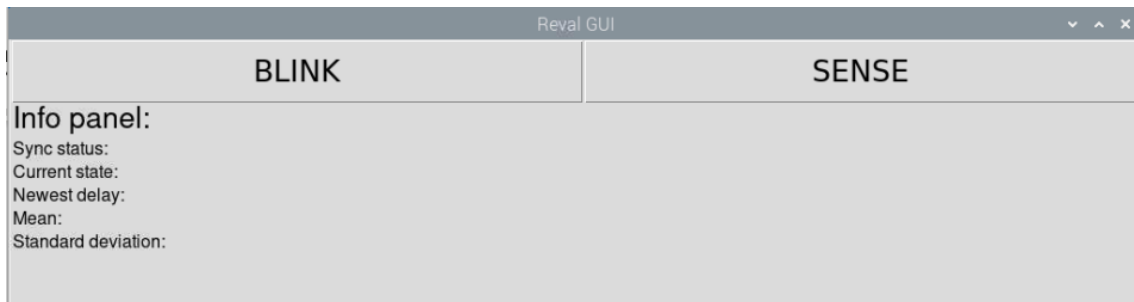
Kasutajaliidese koodid jagunevad mikrokontrollerite järgi, kuid nende eesmärk on sama. Peamine erinevus Raspberry Pi ja ESP32 koodide vahel on graafiku näitamine, ESP32-le ei ole võimalik VNC-ga ligi pääseda kuna puudub operatsioonisüsteem. ESP32 näitab informatsiooni ainult OLED ekraanil. Algoritmide skeemide (plokk skeemide) joonistamisel kasutas autor draw.io-d [14].

Kasutajaliidese koodide silumisel kasutati, vastavalt vajadusele, ChatGPT-d ja Copilot-it [15] [16].

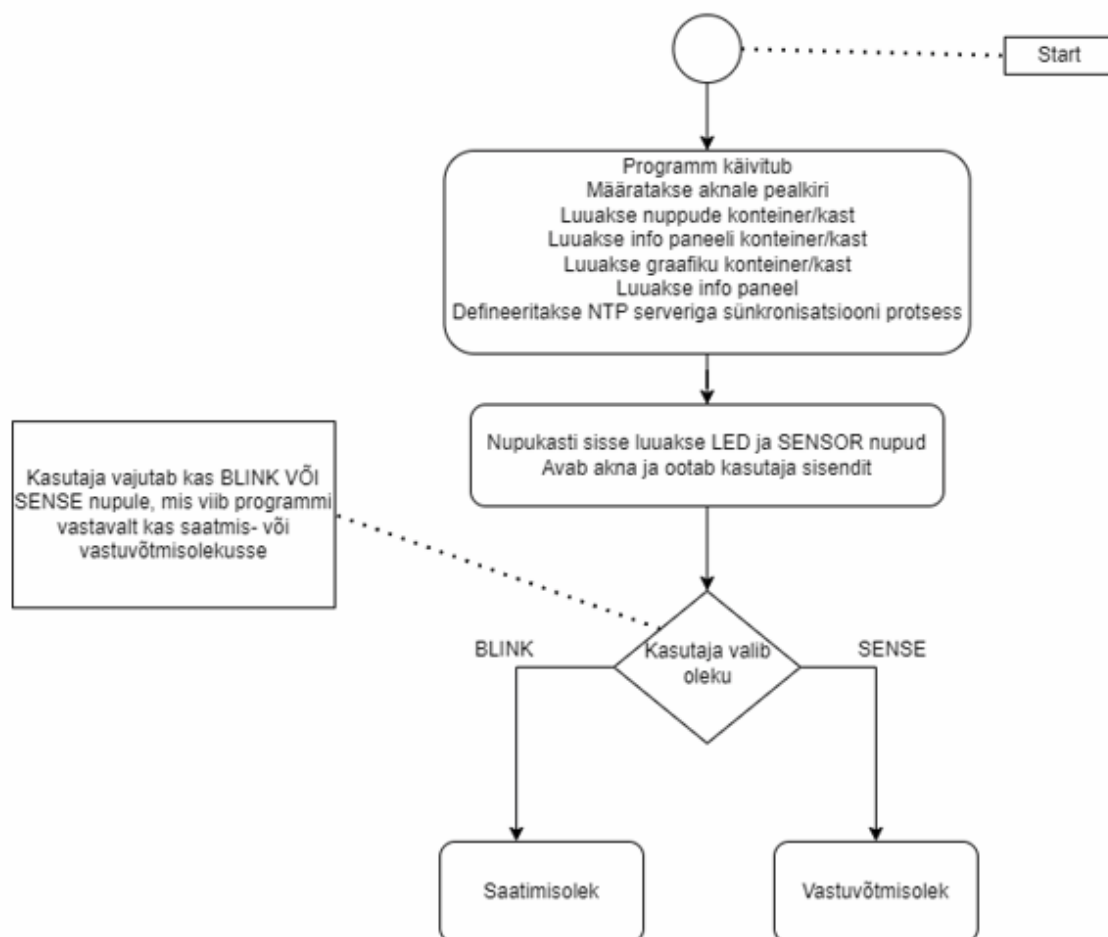
3.5.1 Mikrokontrollerite Raspberry Pi 4 ja Raspberry Pi 5 koodid

Programmi käivitumisel on esimese asjana operaatoril näha kahte nuppu „BLINK“ ja „SENSE“(Joonis 4) ning tühja info paneeli. „BLINK“ nupp viib seadme saatmisolekusse,

ehk hakkab vilgutama valgusdiodi ja „SENSE“ nupp viib seadme vastuvõtmisolekusse, kus sensor ootab signaali ja töötleb saadud infot (Joonis 5).



Joonis 4. Graafilise kasutajaliidese avavaade.

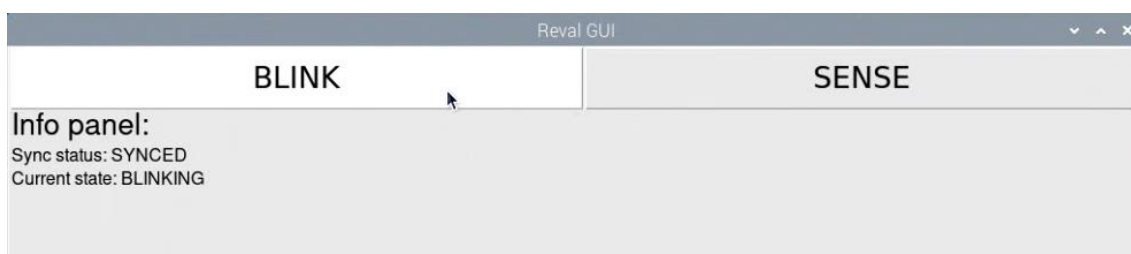


Joonis 5. Programmi käivitamise skeem.

3.5.1.1 Saatmisolek

Saatmisolekus (Joonis 8) kontrollitakse esimese asjana, kas sünkronisatsiooni protsess on eraldi lõimel juba käimas või on see vaja käivitada (Joonis 9). Seejärel peidetakse ebaoluline informatsioon ning kustutatakse vastuvõtmisolekus joonistatud graafik kui see eksisteerib (Lisa 3). Tasub mainida, et enne informatsiooni kustutamist, salvestatakse see logifaili (Lisa 2). Peale üleliigse informatsiooni eemaldamist kirjutatakse nii graafilisele kasutajaliidesele (Joonis 6) kui ka OLED-ile (Joonis 7) NTP sünkronisatsiooni tulemus ja praegune olek.

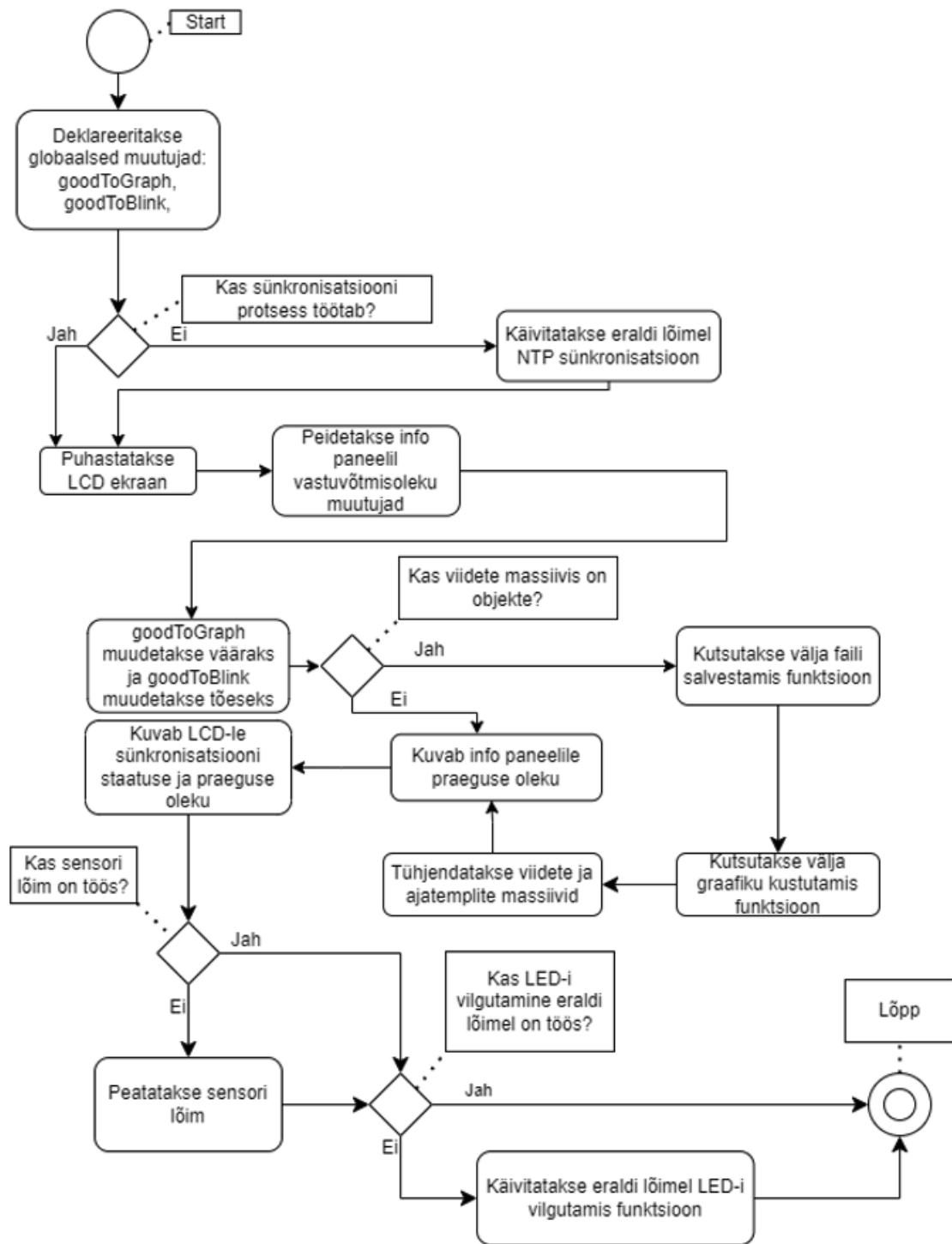
Viimasena käivitatakse eraldi lõimel valgusdiodi vilgutamise protsess (Lisa 4). valgusdiodi vilgutatakse igal paarissekundil, ehk vastuvõttev seade võtab vilkumise ajatempliks viimase paarissekundi ajatempli.



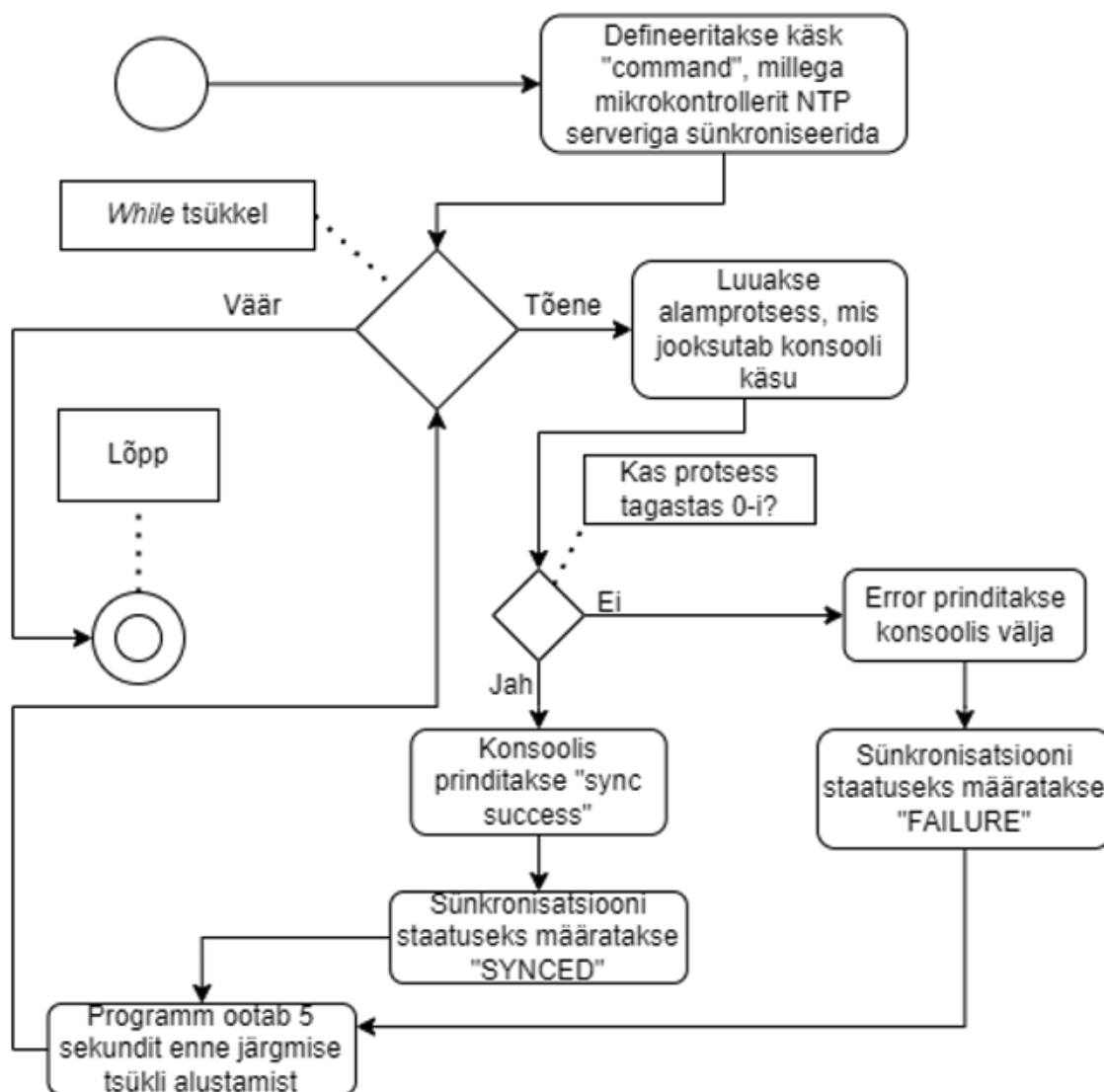
Joonis 6. Saatmisolek graafilisel kasutajaliidesele



Joonis 7. Saatmisolek OLED ekraanil



Joonis 8. Saatmisoleku algoritm.



Joonis 9. Sünkronisatsiooni protsess.

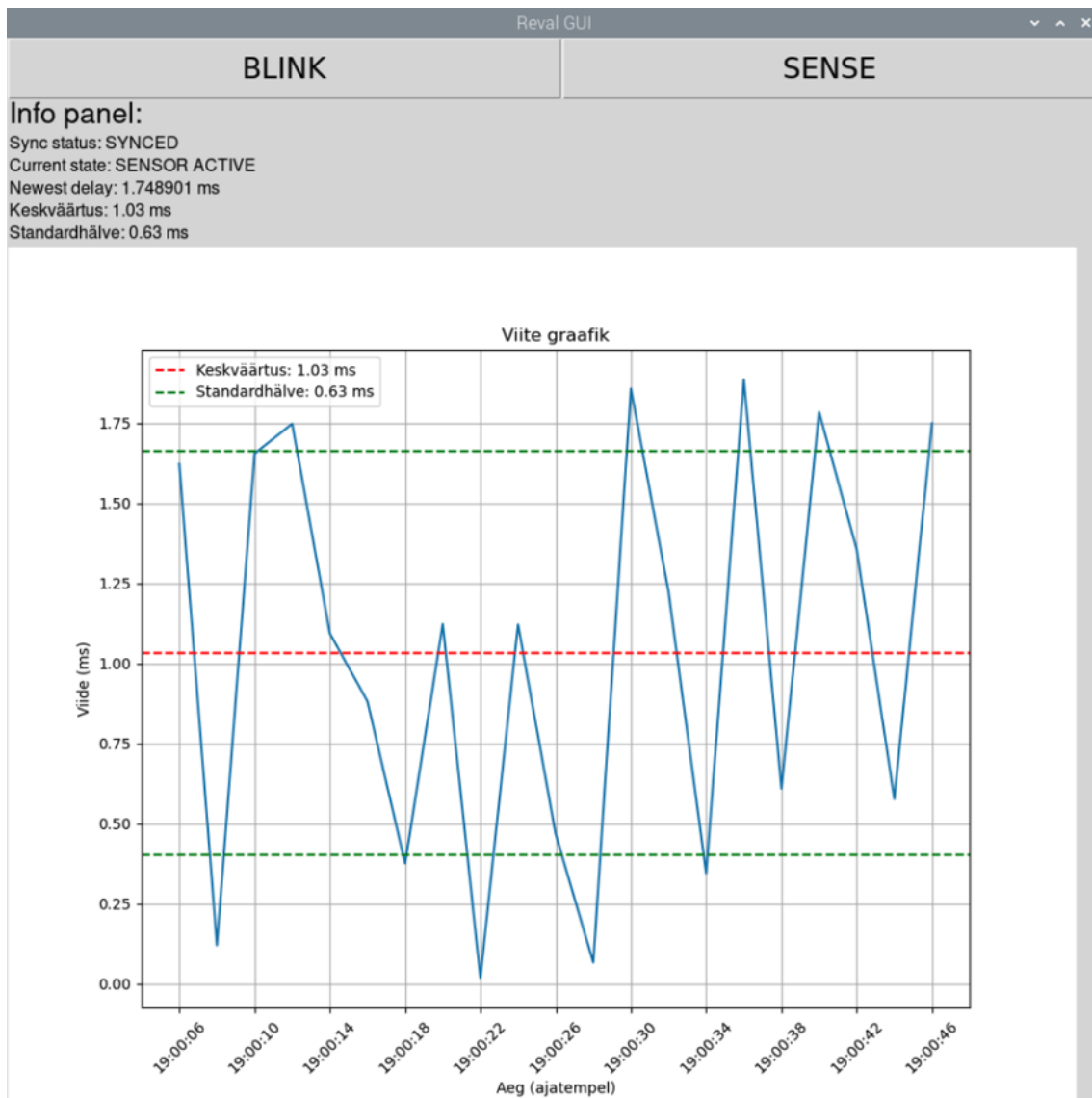
NTP sünkronisatsiooni funktsiooni kood on saanud tugevat inspiratsiooni varasemalt tehtud *proof-of-concept*'i (idee tõendamise) koodist [1]. Ajaküsimise funktsionaalsusega see lõputöö otseselt ei tegele, selle lõputöö raames peab olema NTP sünkronisatsiooni õnnestumine kuvatav. Funktsioon kutsutakse välja alles peale oleku valimist, sest kui funktsioon käivitada programmi käivitamisel, ei olnud võimalik info paneelil ja LCD ekraanil sünkronisatsiooni staatust kuvada. NTP serveriga uuendatakse aega iga viie sekundi tagant. Viie sekundilise intervalli põhjus on kasutajaliidese testimine ja Taltechi NTP serveri võimalikult vähene koormamine.

3.5.1.2 Vastuvõtmisolek

Vastuvõtmisolekus ootab mikrokontroller sensori signaali ehk seade ootab monitoril valgusdiodi vilkumist. Graafilisel kasutajaliidesel ja OLED ekraanil on võimalik näha viimast mõõdetud viidet, seni mõõdetud viidete keskväärtust ja standardhälvet (Joonis 10). Graafilisel kasutajaliidesel genereeritakse viimaste andmete alusel graafik (Joonis 11), ning jõudes 30 mõõtmiseni histogramm (Lisa 5).



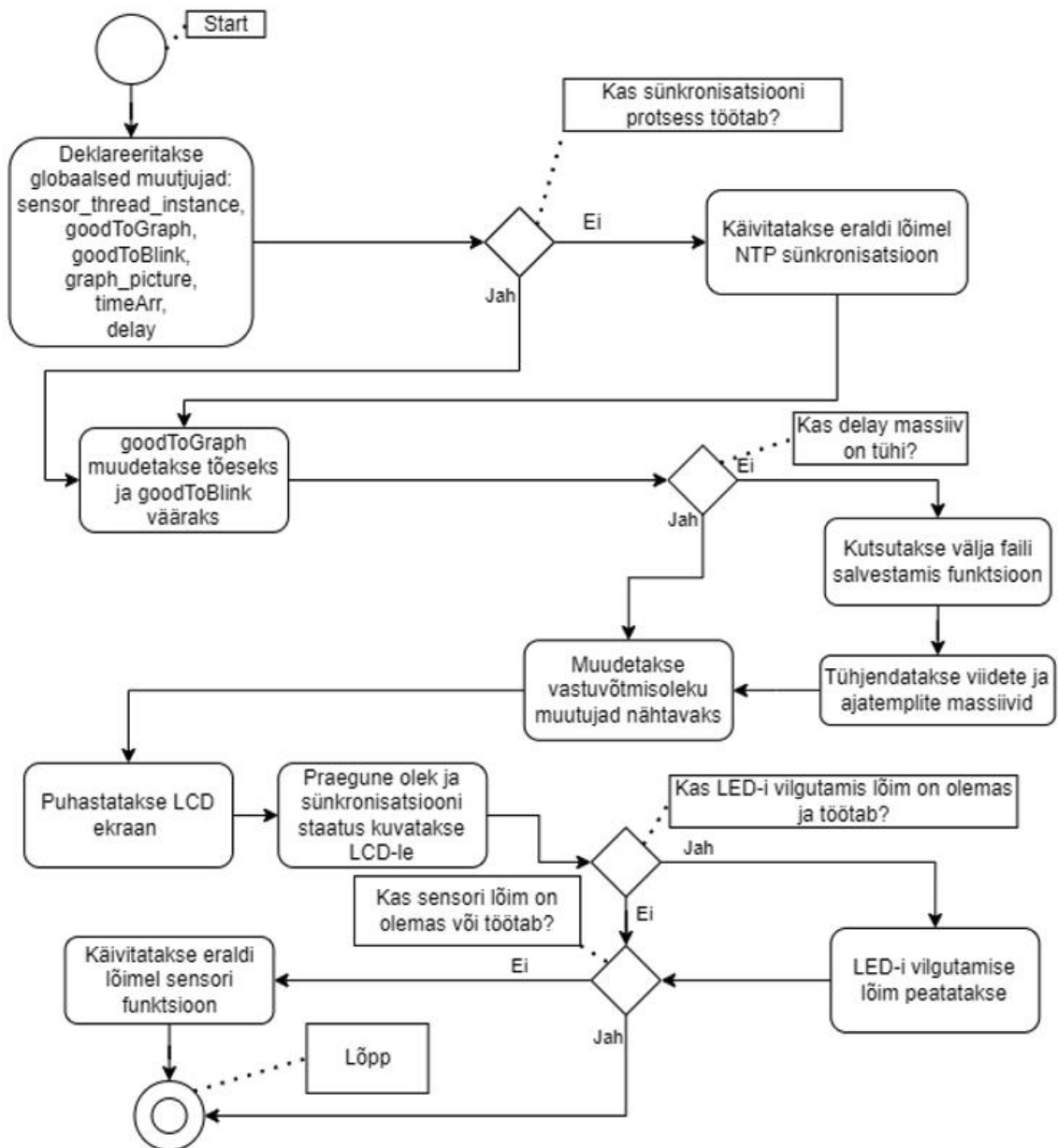
Joonis 10. Vastuvõtmisolek OLED ekraanil.



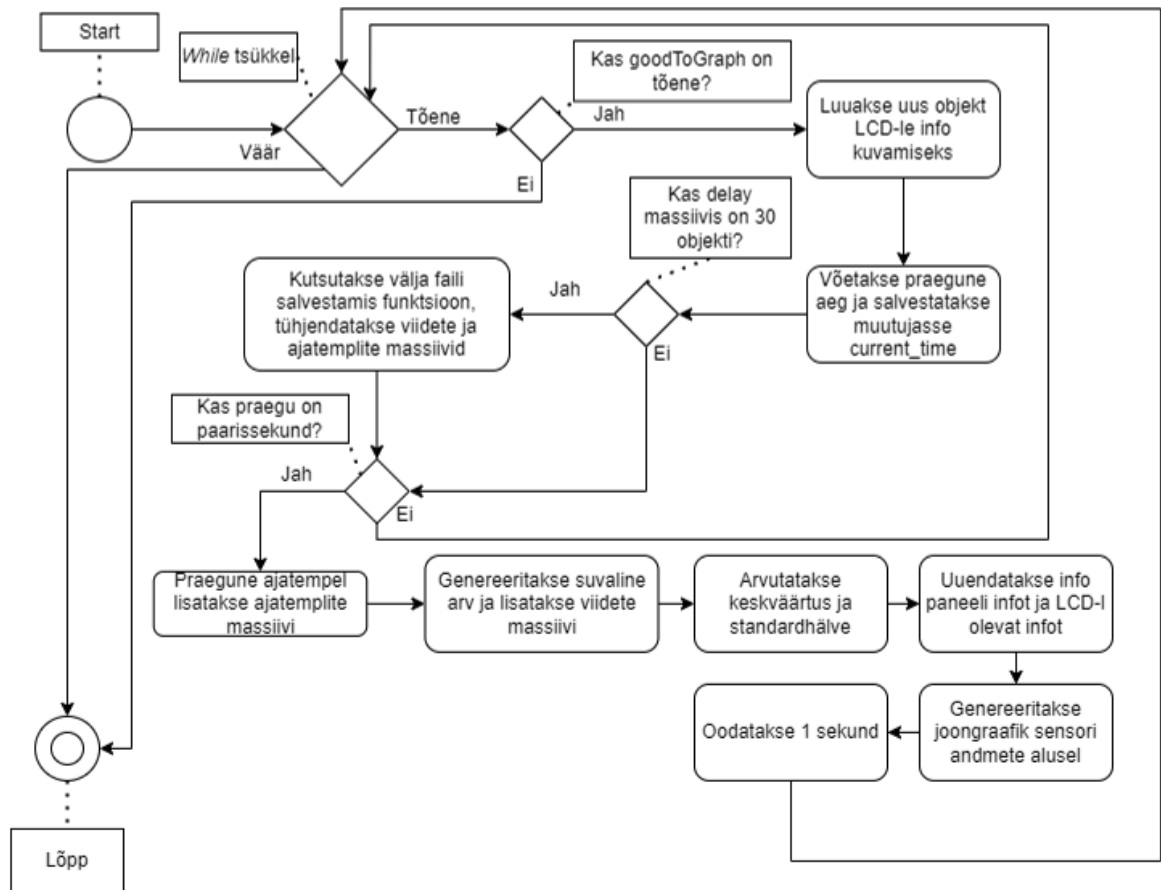
Joonis 11. Vastuvõtmisolek graafilisel kasutajaliidesel.

Vastuvõtmisolekus (Joonis 12) kontrollitakse samuti esimese asjana, kas sünkronisatsiooni protsess (Joonis 9) on eraldi lõimel töös või on see vaja käivitada. Seejärel kontrollitakse, kas viidete massiiv on tühi, kui massiiv ei ole tühi siis salvestatakse informatsioon eraldi logifaili. Peale faili salvestamist tühjendatakse viidete ja ajatemplite massiivid. Iga kord kui vajutatakse nuppu “SENSE” salvestatakse graafiku informatsioon faili ja kustutatakse vana informatsioon, ehk nupu uuesti vajutamisel värskendatakse graafikut ilma, et vana informatsioon täielikult kaoks.

Viimasena muudetakse graafilisel kasutajaliidesel nähtavaks mõõtetulemuste näitajad ning peatatakse valgusidiodi vilgutamise lõim, juhul kui see töötab. Viimasena käivitatakse eraldi sensori lõim, millel jookseb sensori kood (Joonis 13).



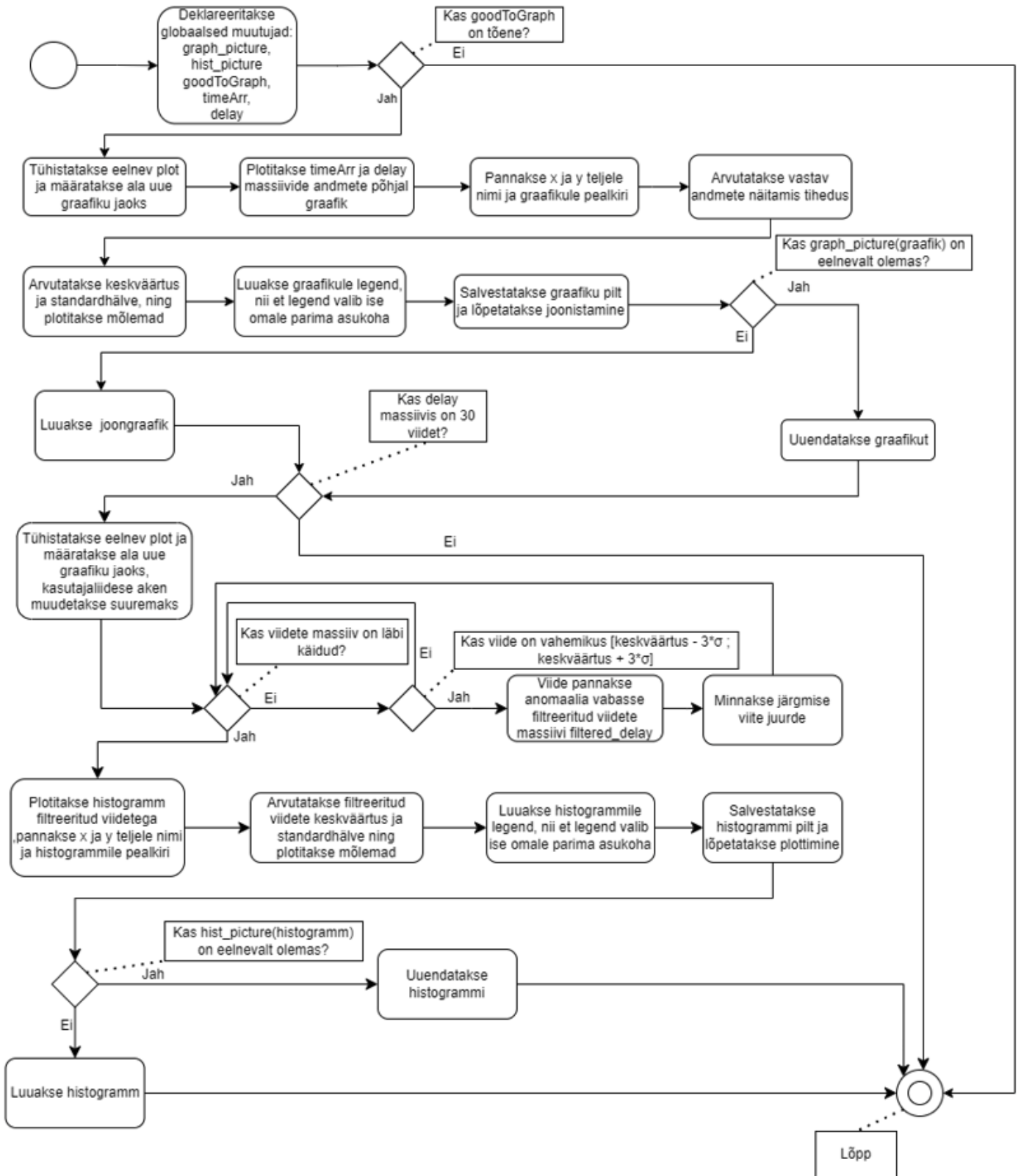
Joonis 12. Vastuvõtmisoleku algoritm.



Joonis 13. Eraldi lõimel sensori funktsioon.

Sensori funktsioon jookseb eraldi lõimel *while* tsüklis. Tsükli alates kontrollitakse, kas viiteid on massiivis 30. Kui viiteid on 30 siis kutsutakse välja faili salvestamise funktsioon (Lisa 2) ning kõik massiivid tühjendatakse. Funktsiooni peamine mõte on genereerida juhuslikke arve vahemikus 0 kuni 2 ms (nanosekundi täpsusega), mida saab kasutada mõõtmise simuleerimiseks. Juhuslike arvude vahemik on võetud *proof-of-concept*'ist saadud tulemuste alusel [1]. Iga paarissekund genereeritakse üks viide, mis lisatakse koos viidete massiivi. Peale viite genereerimist, kutsutakse välja graafiku genereerimisfunktsioon (Joonis 14).

3.5.1.2.1 Graafiku genereerimine



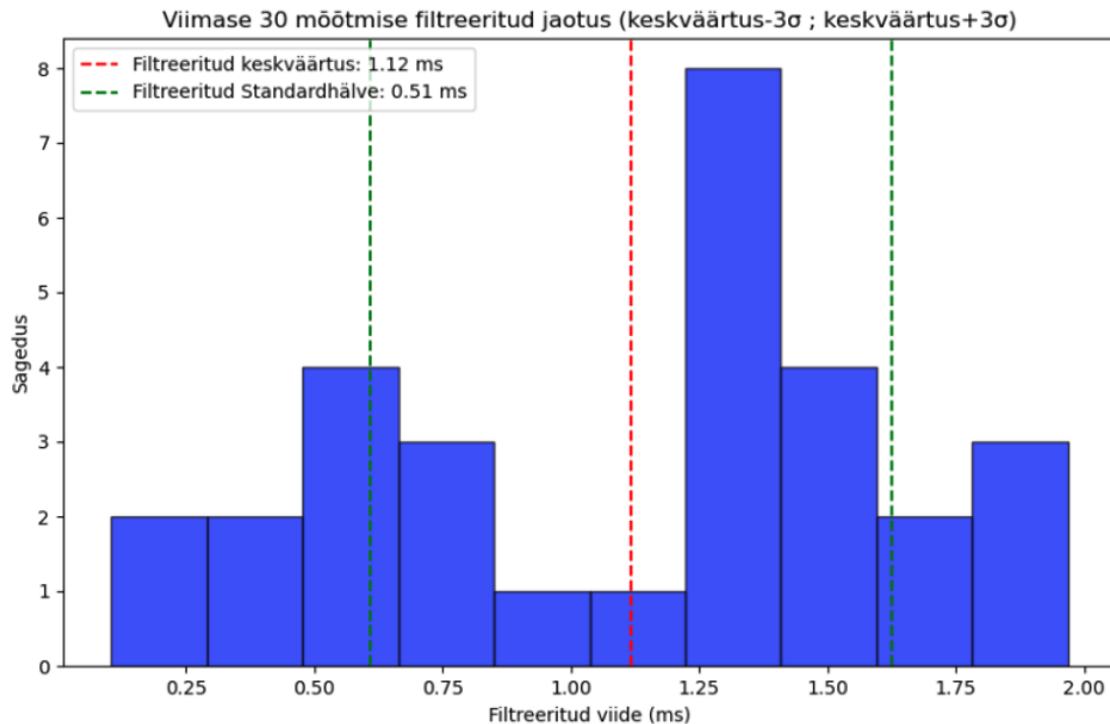
Joonis 14. Graafiku genereerimisfunktsioon.

Graafiku genereerimisfunktsioon joonestab esmalt joongraafiku viidete ja ajatemplite massiivi järgi (Joonis 15). Kõik graafikud on joonestatud kasutades Pythoni *matplotlib* teeki [17].



Joonis 15. Simuleeritud tulemustega joongraafik.

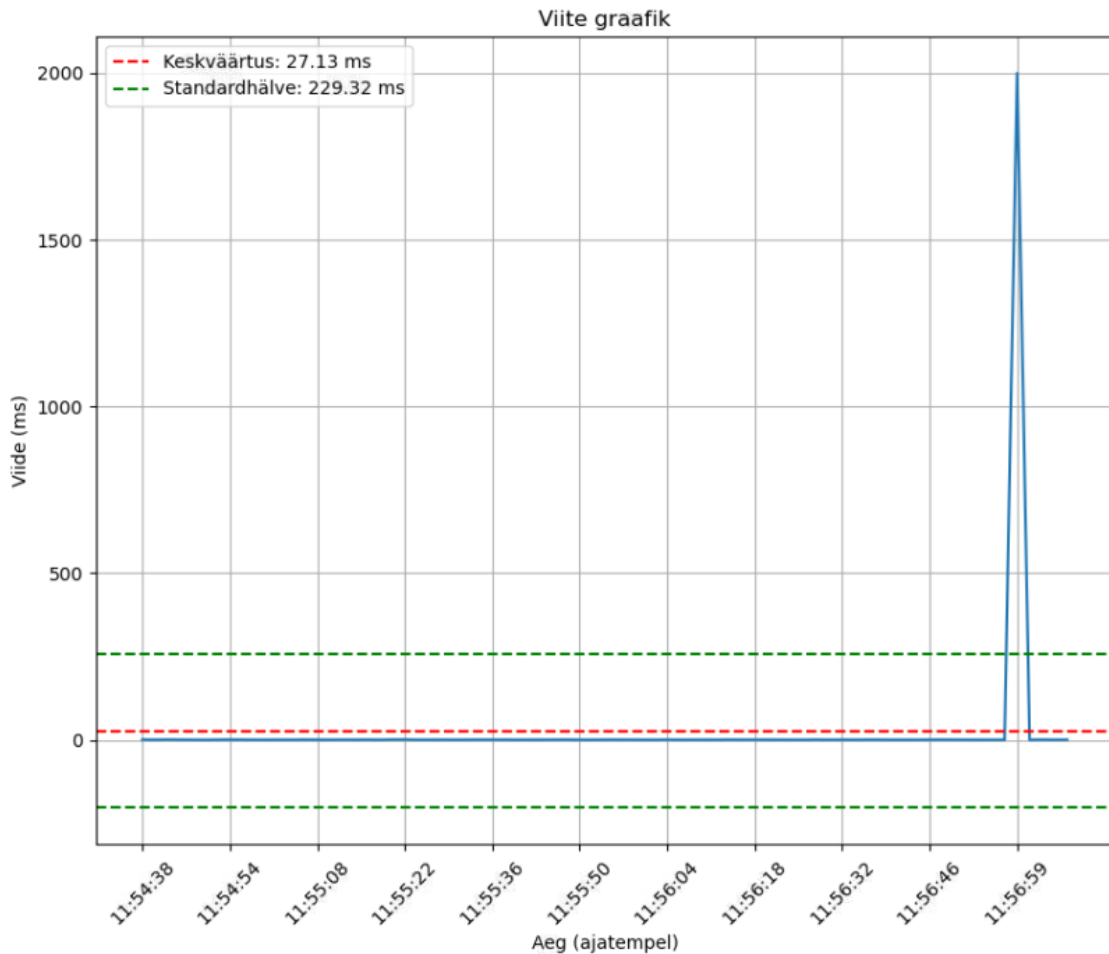
Kui 30 viidet on viidete massiivis, siis joonistab funktsioon joongraafiku kõrvale filtreeritud histogrammi (Joonis 16).



Joonis 16. Simuleeritud tulemustega histogramm.

Enne histogrammi joonistamist, käiakse läbi kogu viidete massiiv. Luuakse uus filtreeritud viidete massiiv, kuhu salvestatakse kõik viited, mis on vahemikus keskväärtus $\pm 3\sigma$. Kui joongraafikut uuendatakse pidevalt, siis histogramm jääb graafilisel kasutajaliidesel nähtavale kuni viidete arv jõuab uuesti 30-ni, seejärel histogrammi uuendatakse. Nõnda saab operaator esmase aimduse jaotusest, mis tuleks ilma histogrammita alles logifaile analüüsides välja.

Histogrammi algandmete filtreerimise vajadus selgus graafiku genereerimisfunktsiooni testimisel. Graafiku genereerimisfunktsiooni testimiseks, lasi autor funktsioonil koostada graafik varasemalt tehtud *proof-of-concept*'i projekti andmetel [1]. Andmeteks on 75 mõõtmist, mille alusel on koostatud joongraafik (Joonis 17) ja histogramm (Joonis 18). Reaalsete mõõtmiste puhul tuleb tihti ette anomaaliaid, kus mõni väärtus oli kordades suurem või väiksem kui kõik teised väärtused, selle tulemuseks oli joongraafik (Joonis 17), mis ei andnud edasi kasulikku infot. Sellest tulenevalt otsustas autor luua joongraafikule lisaks histogrammi, et näidata viidete jaotust.

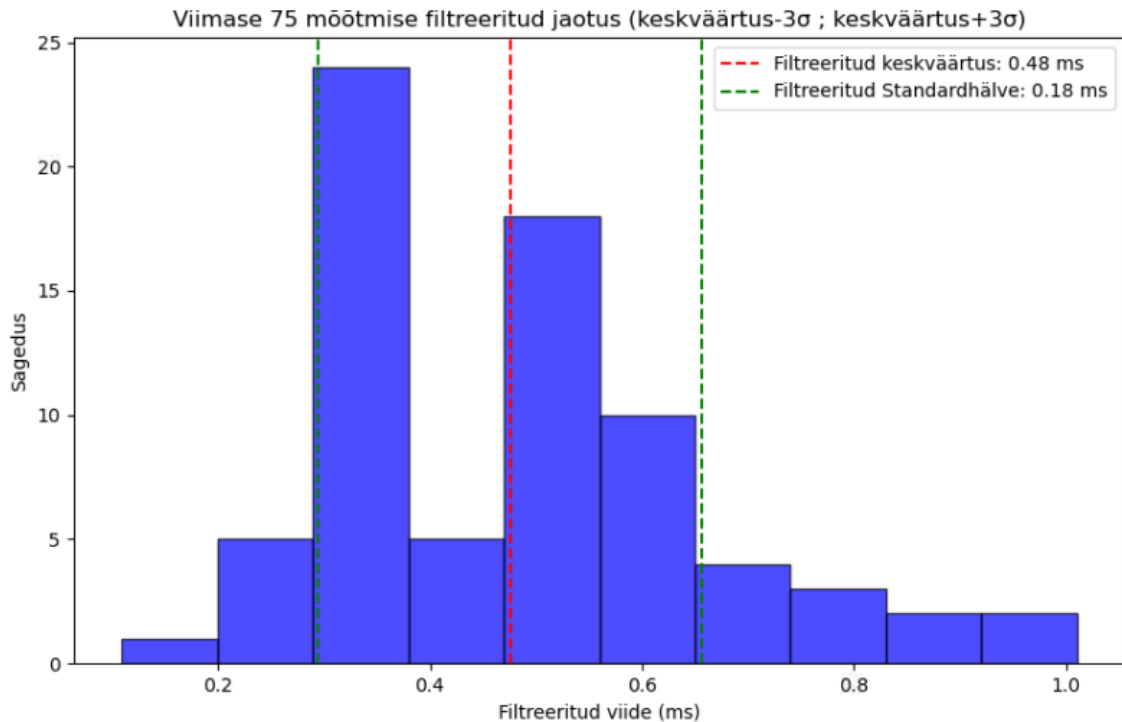


Joonis 17. 75 mõõtetulemusega joongraafik.

Viidete massiiv, antud juhul 75 mõõtmist, filtreeritakse välja anomaaliatest, mis jaotuse hindamist võivad segada. Histogramm koostatakse filtreeritud viidetega.

Filtreeritud viiteid faili ei salvestata, need on ainult jaotuse näitamiseks. Faili salvestatakse ainult need viited (filtreerimata), mille alusel tehti joongraafik.

Filtreeritud viidetega histogrammilt (Joonis 18) oli võimalik aru saada, milline oli tegelik viidete jaotus. Lisaks arvutati filtreeritud viidetega välja keskväärus ja standardhälve (Joonis 18), mis on võrreldes joongraafikuga (Joonis 17) märkimisväärselt erinevad.



Joonis 18. 75 mõõtmistulemusega histogramm.

3.5.2 Mikrokontrollerite Raspberry Pi 4 ja Pi 5 koodi erinevused

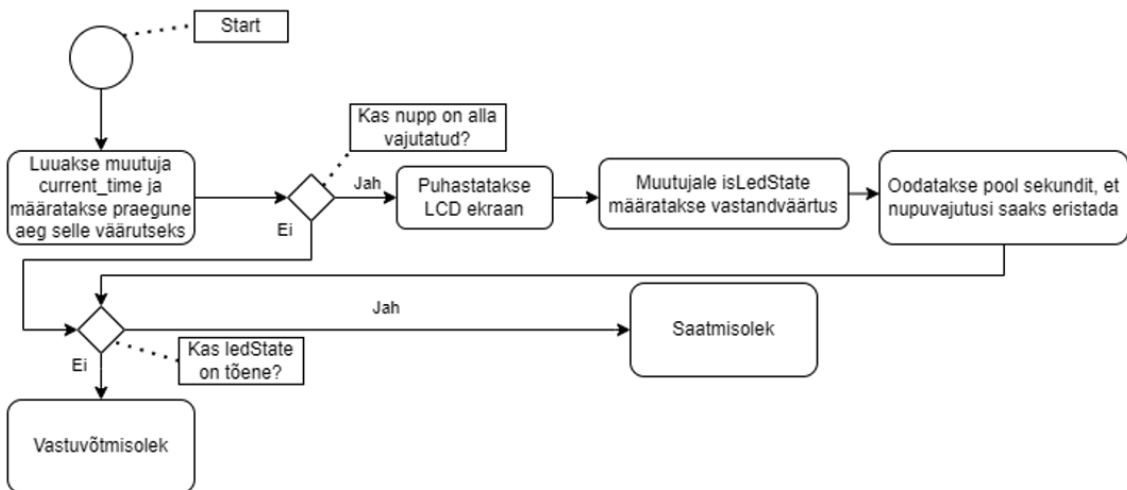
Pi-de puhul esinesid teatud erinevused teekide kasutamise puhul, loogika jäi samaks. OLED-i kasutamiseks kasutas Raspberry Pi 5 *luma.oled* teeki, Pi 4 kasutas aga *adafruit_ssd1306* teeki. valgusdiodi vilgutamiseks kasutas Pi 5 *gpiozero*'t ja Pi 4 kasutas *RPi.GPIO*'t.

3.5.3 Mikrokontrolleri ESP32 kood

ESP32-e kood on kirjutatud C keeles ja seadme programmeerimiseks kasutas autor Arduino IDE programmi.

ESP32-e kasutajaliides on tunduvalt lihtsam Raspberry Pi omadest, sest puudub graafiline kasutajaliides. Programmi initsialiseerimist iseloomustav kood asub lisa (Lisa 5). Programmi käivituses on seade määratud automaatselt saatmisolekusse, sest oleku vahetamiseks peab operaator vajutama BUT1 nuppu ESP32-e küljes. Nupu vajutamiseks, peab operaator eemaldama seadmelt kaane. Nupu vajutamine on antud lahenduses ebamugav, seega see ülesanne jääb operaatorile kes asub peakontoris monitori ees (Joonis 19).

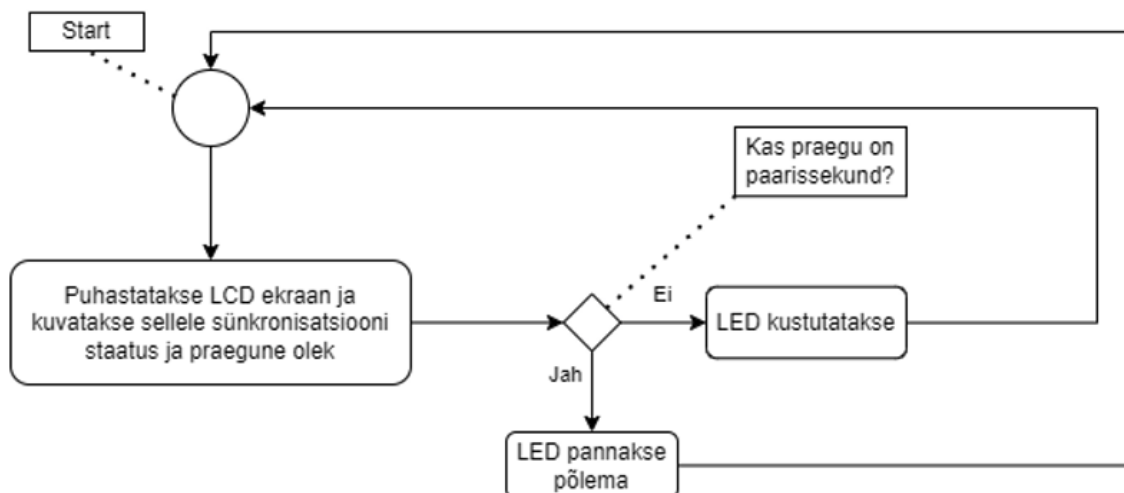
Peamine põhjus, miks autor ESP32-le välist nuppu ei paigaldatud on aja puudus.



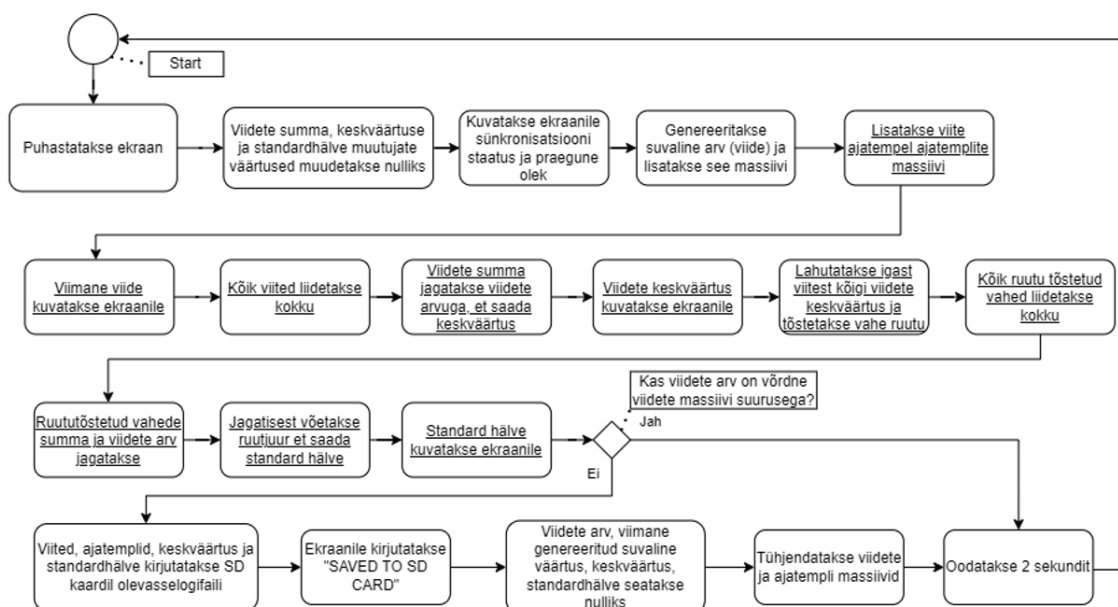
Joonis 19. ESP32-e olekute valik.

ESP32 vilgutab saatmisolekus igal paarissekundil valgusdiodi nagu Raspberry Pi-d (Joonis 20).

Vastuvõtmisolekus genereerib ESP32 viiteid, näitab OLED-il viimast mõõtetulemust, viimase 30 mõõtmise keskväärtust ja viimase 30 mõõtmise standardhälvet, ning salvestab viited ja arvutatud informatsiooni SD kaardile (Joonis 21).



Joonis 20. ESP32-e saatmisolek.

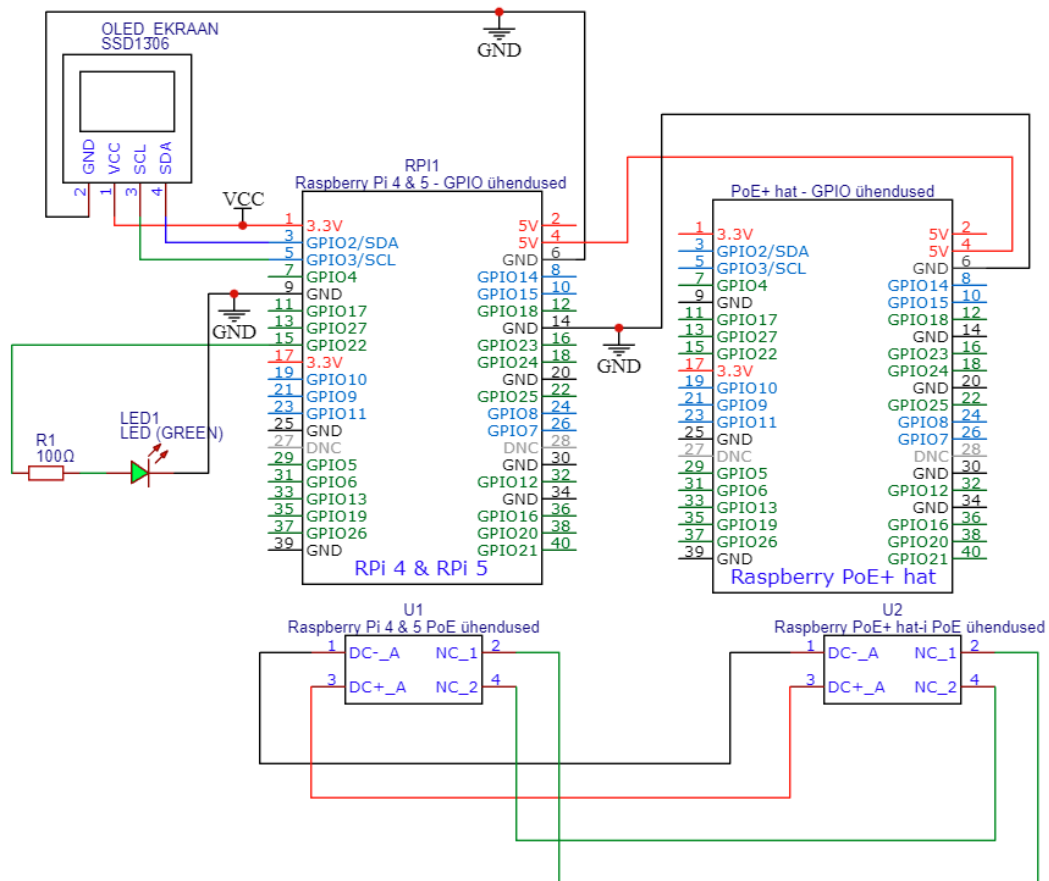


Joonis 21. ESP32-e vastuvõtmisolek.

3.6 Kasutajaliidese testimise skeemid

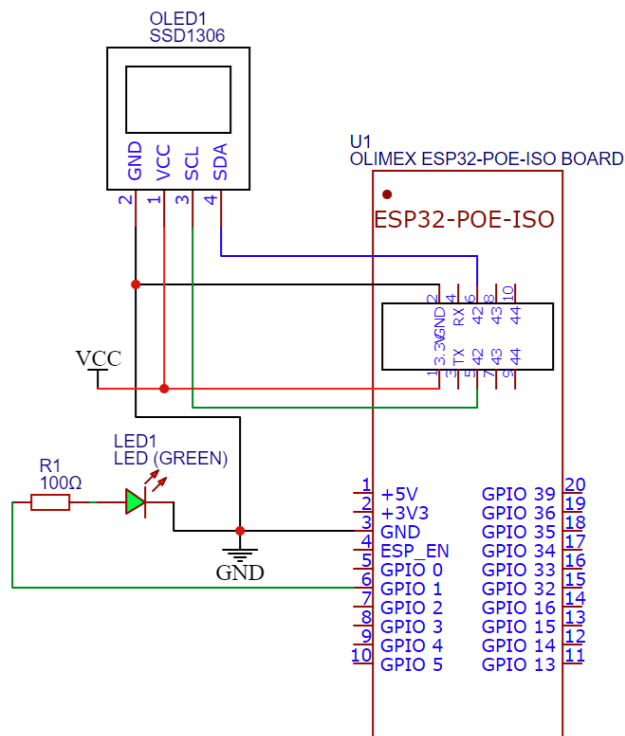
Kasutajaliidese arendamise käigus ühendati OLED ekraan ja valgusdiodid mikrokontrollerite külge (Joonis 22 ja Joonis 23). Mikrokontrolleriteks olid ESP32-POE-ISO, Raspberry Pi 4 ja Raspberry Pi 5 [18]. Raspberry Pi-de puhul olid ühendusskeemid identsed [19] [20].

Skeemide loomiseks kasutati EasyEDA-d [21].



Joonis 22. Raspberry Pi-de kasutajaliideste skeem.

OLED ühendati Pi-de külge vastavalt SSD1306-e andmelehele [22].



Joonis 23. ESP32 kasutajaliidese skeem.

4 Toitesüsteem

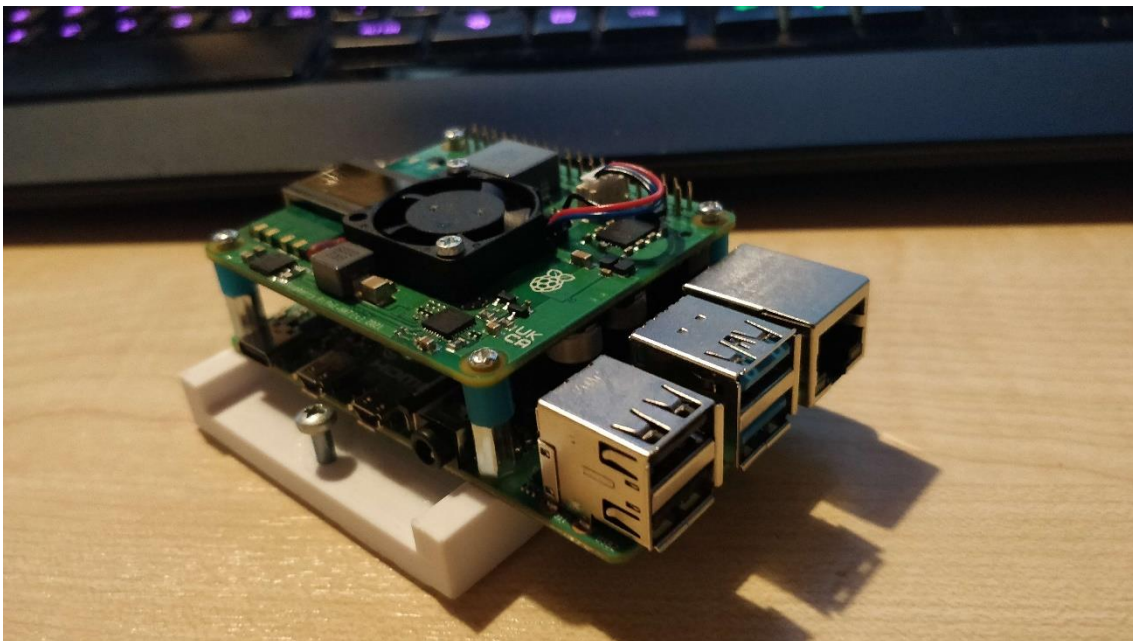
Toitesüsteemi arendust mõjutab eelkõige seadmete kasutus keskkond. Selleks, et mõõteseadet oleks võimalikult lihtne kasutada erinevates keskkondades, otsustati toitena kasutada PoE-d. PoE sisaldab endas nii andmeedastus- kui ka toitekaablit, mille tõttu on seadme kasutamiseks vajalik ainult üks juhe. Antud lõputöös otsustas autor kasutada PoE+ *hat*'i, sest PoE+ on tagurpidi ühilduv tavalise PoE-ga. PoE+ on ühtlasi enamlevinud lahendus ja selle tõttu toimus *hat*'i kättesaamine kiiremini.

Toitesüsteemi skeemide loomiseks kasutas autor EasyEDA-t ja Microsoft Paint-i [21], [23].

4.1 Seadmete toide

Raspberry Pi 4-le istub peale Raspberry Pi PoE *hat*, mis võimaldab Raspberry Pi 4-ja toita PoE-ga (Joonis 24). Pi 5-le aga antud *hat* ei sobi, sest Pi 5 PoE pinnid ei ole samas kohas mis Pi 4-ja omad (Joonis 26).

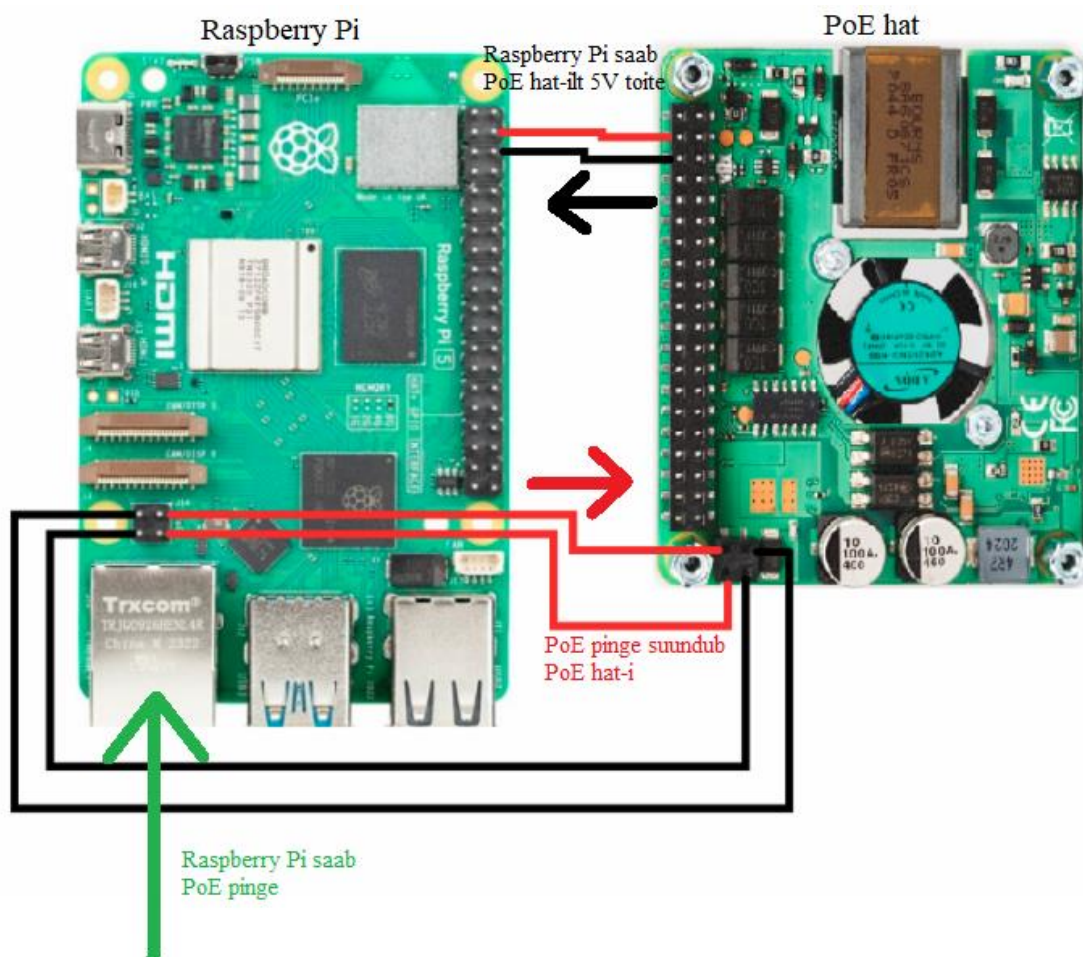
ESP32 lisa riistvara ei vaja, PoE pingemuundur on sisse ehitatud [24].



Joonis 24. Raspberry Pi 4 koos PoE+ hat-iga.

4.1.1 Mikrokontrolleri Raspberry Pi PoE hat-i tööpõhimõte

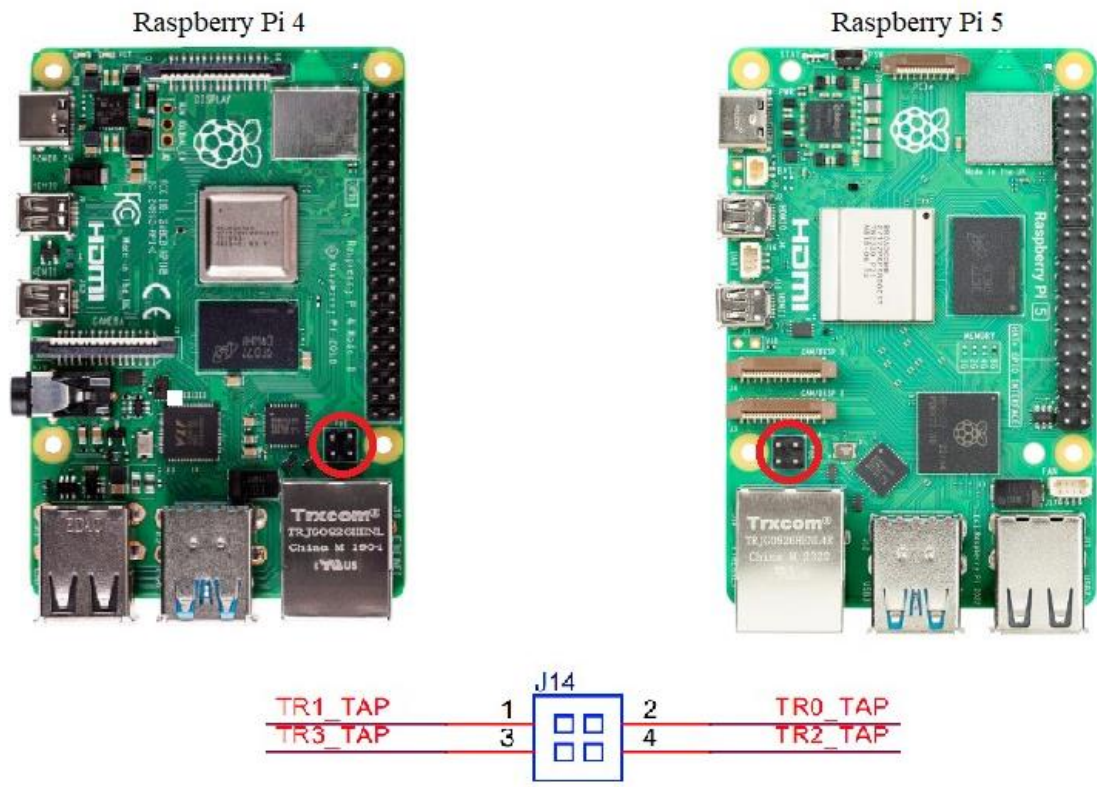
Raspberry Pi saab PoE sisse Etherneti pordist, PoE *hat* on ühendatud Pi PoE pinnide külge, läbi mille jõuab PoE pinge PoE *hat*'ti. PoE *hat* on DC-DC stepdown konverter, mis viib pinge 5V-le ja annab selle Pi toitepinnist sisse (Joonis 25).



Joonis 25. Raspberry Pi PoE toide [25], [26].

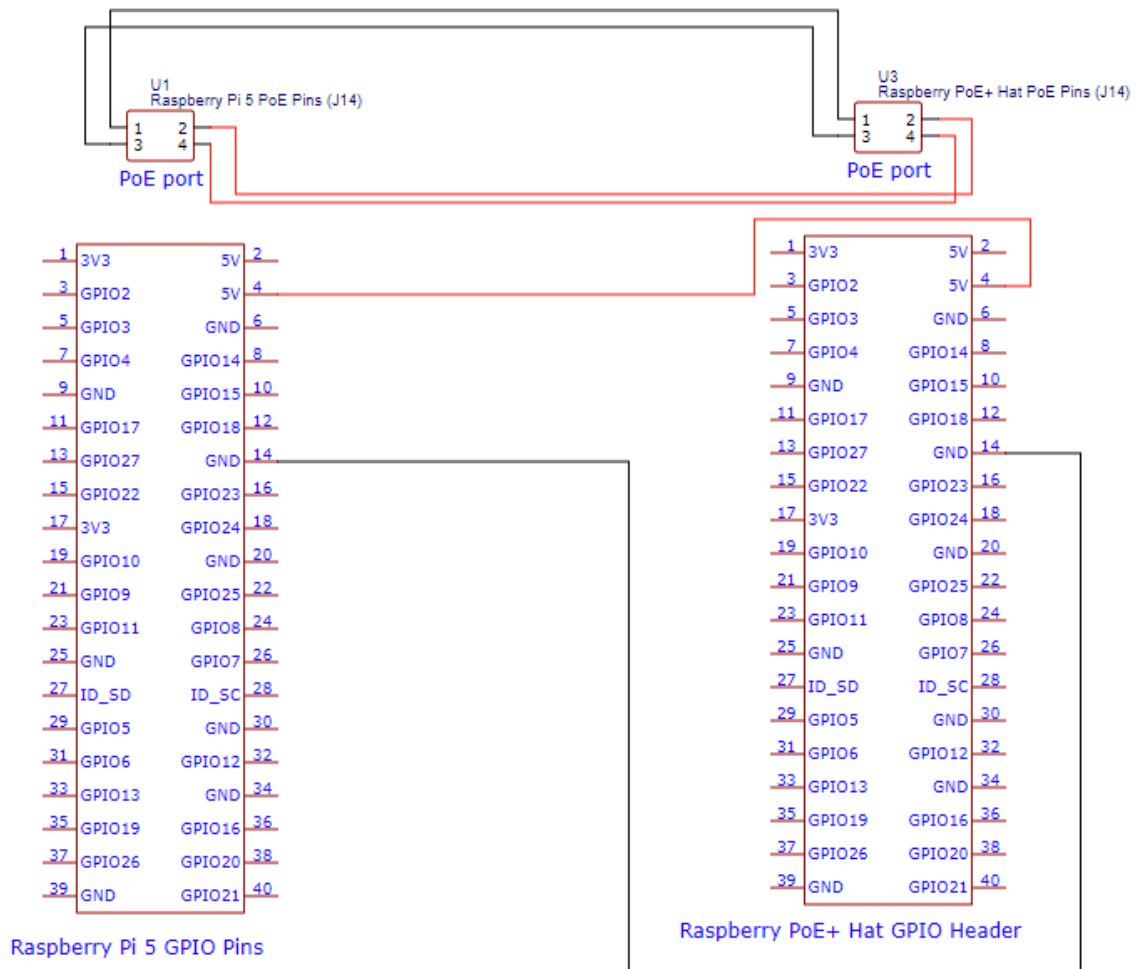
4.1.2 Mikrokontrolleri Raspberry Pi 5 PoE lahendus

PoE+ *hat*-i skeemi Raspberry avaldanud ei ole, kuid on olemas RPi 4 skeem, millest autor sai tuletada PoE+ *hat*-i pinnide tähendused [27]. Võrreldes RPi 4-a PoE pinne RPi 5-e PoE pinnidega, leidis autor et nende asukoht on muutunud aga järjekord on sama (Joonis 26) [28].



Joonis 26. Raspberry Pi-de PoE pinnide võrdlus [27], [29], [25].

Autor ühendas PoE+ *hat*-i J14 pistiku otse Raspberry Pi 5-e PoE pinnide (J14 ühenduse) külge ning PoE+ *hat*-il oleva GPIO headeri 5 V ja GND väljundpinnid otse Raspberry Pi 5-e 5 V ja GND sisendpinnide külge (Joonis 27).



Joonis 27. Raspberry Pi 5 PoE lahenduse skeem.

5 Korpuse prototüüp

Ühe seadme korpus koosneb kuuest tükist: alus, kaks L tähe kujulist seinatükki, kaks küljeplaati ja kaan. Lisaks valmistati mikrokontrollerite kinnitused, mida ma korpuse tükideks ei arvestata.

5.1 Modulaarsus

Korpus on prototüüp, seega mikrokontrollereid millega korpus peab ühildama on kolm. Sõltuvalt sellest peab korpus olema modulaarne, ehk säilitama kogu funktsionaalsuse sõltumata mikrokontrollerist, mida parasjagu kasutatakse.

5.2 Turul olevad lahendused

Lõputöö eesmärkide seas oli turul olevate lahenduste uurimine, mis antud ülesandega võiksid sobida. Sobiva lahenduse korral uurida nende omadusi ja vastavust toote eesmärkidele.

5.2.1 Mikrokontrollerite individuaalsed korpused

Nõudmised korpusele on liiga spetsiifilised, seega turul olevad tooted ei täida kõiki püstitatud nõudeid korraga. Turgu uurides leiti aga igale mikrokontrollerile eraldi korpuse, mida on võimalik lõpp lahendusse sisse töötada. Antud Raspberry Pi 4-ja korpusel on kruvi augud, mis tagaksid head kinnitusvõimalused (Joonis 28).



Joonis 28. Raspberry Pi 4b turul olev korpus [30].

Probleem, mis individuaal korpuste puhul kerkib on nende füüsiline kohmakus, kui ka majanduslik koormus. Kogu mõteseadme väliskeskkonna eest kaitsmiseks, peab igal juhul kogu süsteemile välise korpuse looma. Otsustati luua igale mikrokontrollerile lihtsa hoidiku/raami, mis hoiab mikrokontrollerit kindlalt paigal ja mille abil on võimalik mikrokontroller suurde korpusesse kinnitada (Joonis 33 ja Joonis 34).

5.3 3D printimine

Autor otsustas korpuse 3D printida kuna tänu Robotiklubi liikmelisusele, on olemas ligipääs ja õigus kasutada klubi vara isiklikel eesmärkidel. Lisaks on töölooja tutvusringkonnas inimene, kellel on isiklik 3D printer, mida on võimalik kasutada. Kõik printerid, mida antud lõputöö raames kasutati on Creality Ender 3 V2-ed ning kasutusel olevad filamendid on PLA ja PLA+. 3D mudelite loomiseks kasutati tarkvara Solidworks.

5.3.1 3D printer Ender 3 V2

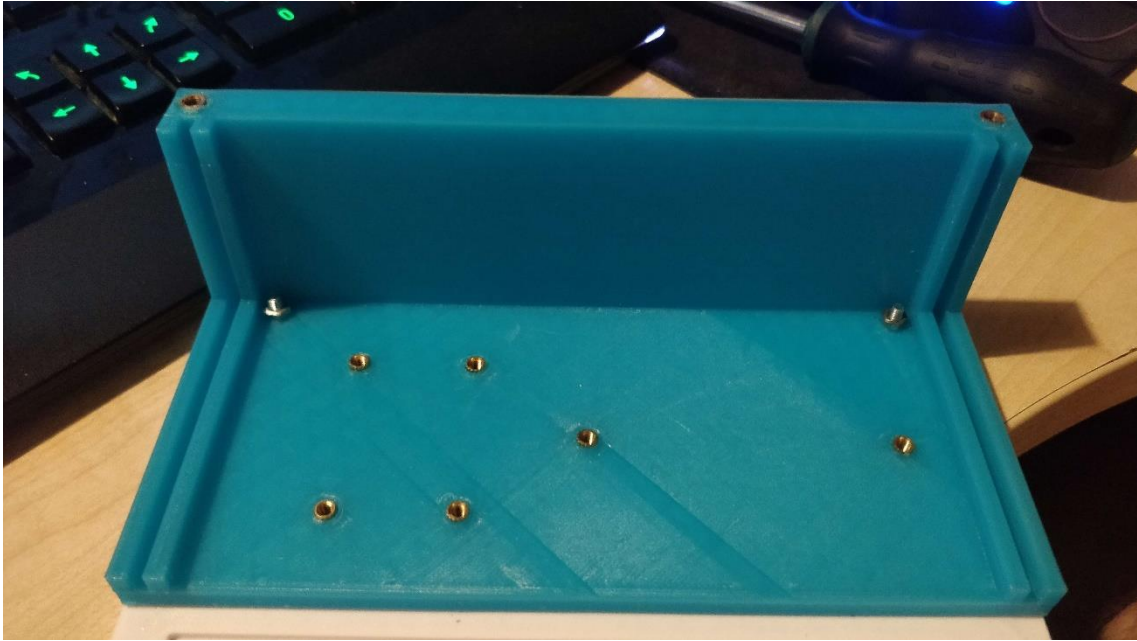
Ender 3 v2 on üks levinumaid Creality 3D printereid. Antud printer on võrdlemisi soodne algaja sõbralik 3D printer, millele leidub palju juhendeid ja õpetusi. Ender 3 v2-e ehitusmaht on 220x220x250 mm, mis on antud lõputöö jaoks piisav.

5.3.2 3D modelleerimine

Modelleerimiseks tuli esmalt joonistada 2D joonis, mida saab seejärel muuta kolme dimensiooniliseks kasutades “Extruded Boss/Base” tööriista. Kui tekkis vajadus loodud kolme dimensioonilisele objektile süvend või auk teha, kasutas autor „Extruded Cut“ tööriista.

5.3.3 Korpuse kokkupanek

Alustüki külge tuleb kruvida 2 L-tähe kujulist seinatükki (Joonis 29). Seinatükidesse on sulatatud M3 kruvide keermestatud sisendid. Keermesisenditega saab kinnitada korpuse külge korpuse kaane (Joonis 32), mikrokontrolleri hoidiku (Joonis 33 ja Joonis 34), Etherneti pistiku hoidiku (Lisa 6) ja PoE *hat*-i hoidja (Joonis 35). L-tähe kujulistel tükidel on sisse disainitud 3 mm sooned, mis on mõeldud korpuse küljeplaatide kinnitamiseks.



Joonis 29. L-tähe kujuline seinatükk.

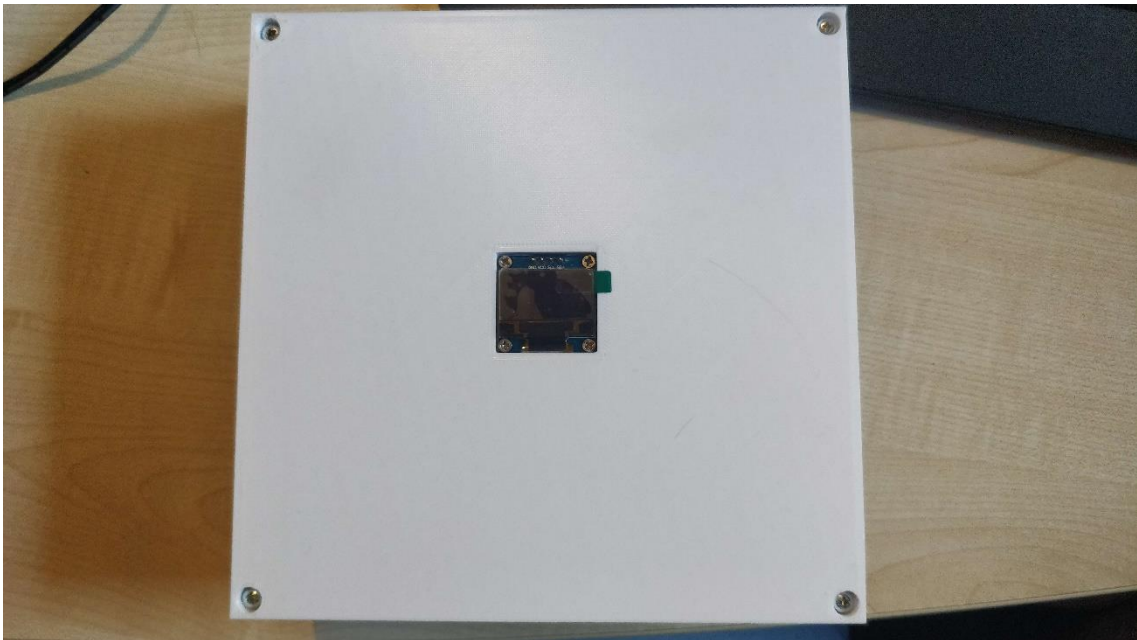
Ühe küljeplaadi külge on disainitud Etherneti pistiku auk (Joonis 30) ja teise küljeplaadi külge on disainitud sensori ja valgusdiodi ava (Joonis 31). Sensori ja valgusdiodi avad on jäetud lahtiseks, et oleks võimalik disainida spetsiifilised kinnitustükid.



Joonis 30. Korpuse Etherneti pordi poolne vaade.



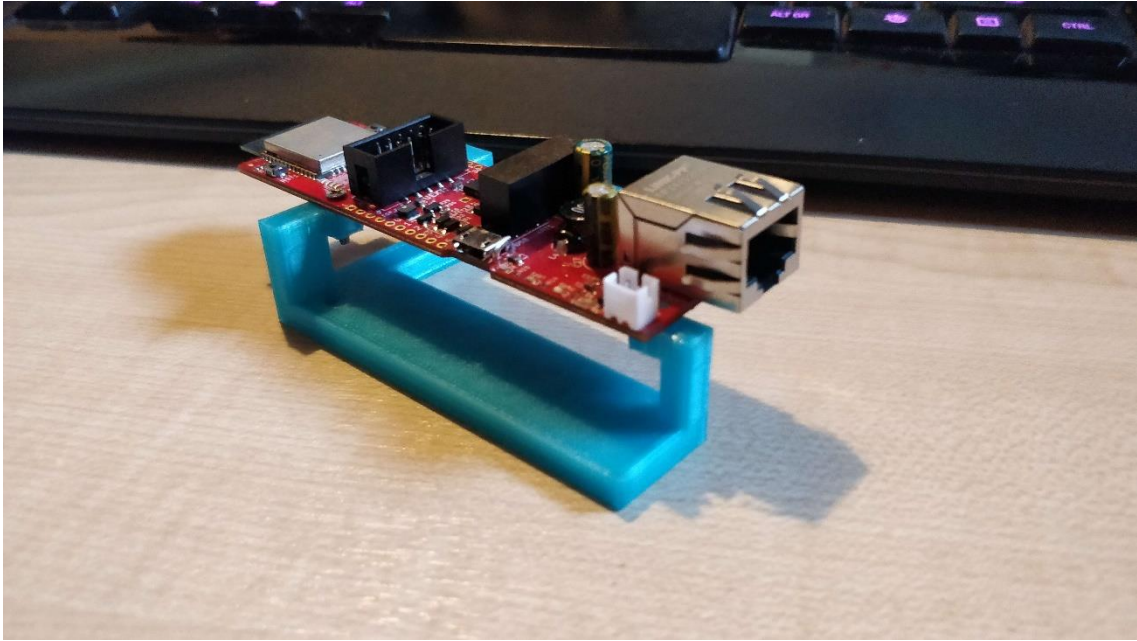
Joonis 31. Korpuse sensori ja valgussensori poolne vaade.



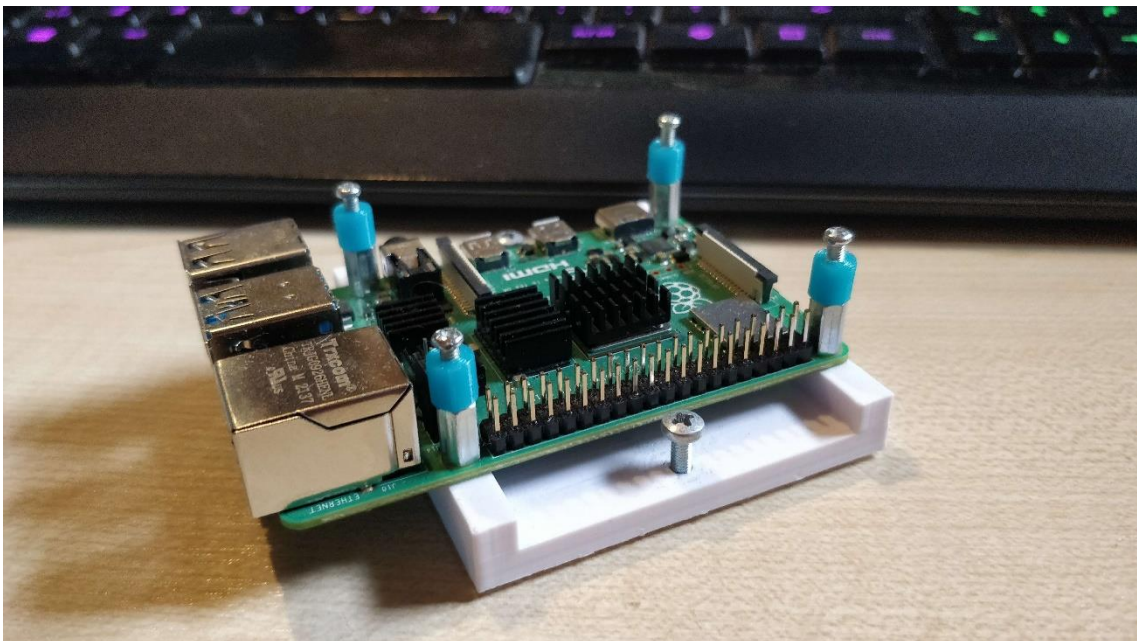
Joonis 32. Korpuse pealtvaade.

5.3.4 Mikrokontrollerite hoidikud

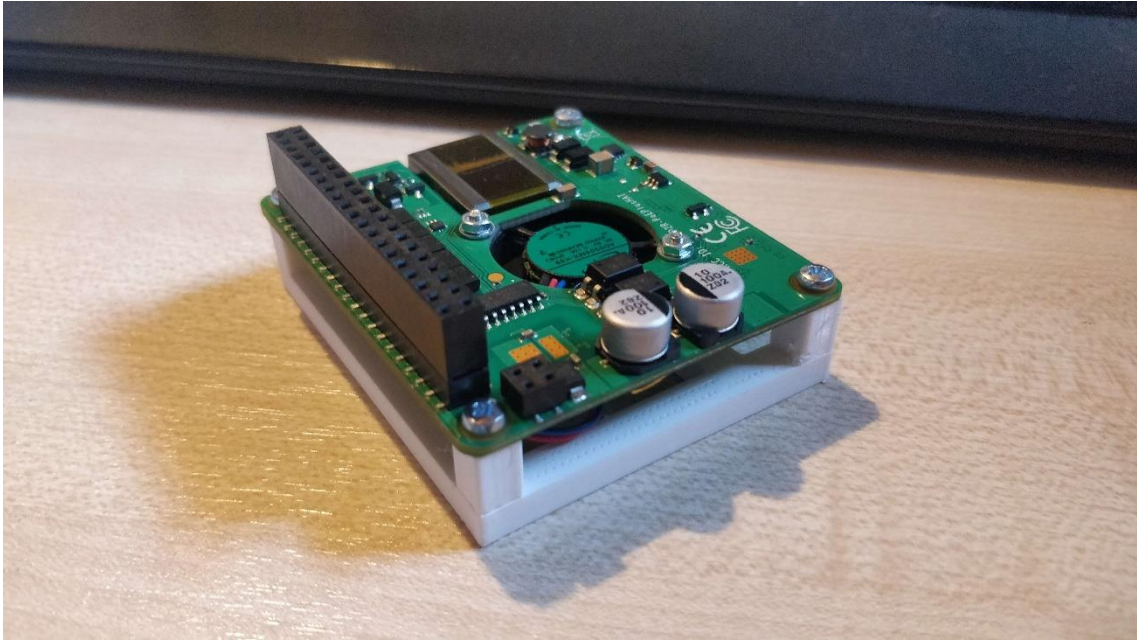
Mikrokontrolleri hoidikuid on kokku 3: ESP32-e hoidik (Joonis 33) ja Raspberry Pi-de hoidik (Joonis 34). Raspberry Pi-d on mõlemad sama augumustriga ning saavad seetõttu kasutada sama hoidikut.



Joonis 33. ESP32-POE-ISO hoidik.



Joonis 34. RPi hoidik.



Joonis 35. PoE *hat*-i hoidik.

6 Edasiarendusvõimalused

6.1.1 Toitelüliti

Lõpptootel võiks kindlasti olla toitelüliti. Toitelüliti mõte oleks tagada mikrokontrollerile mitu toitmiskiisi, ilma mikrokontrollerit maha põletamata. Kui lüliti külge on ühendatud tava toide (USB-C) ja PoE korruga, siis lüliti annaks mikrokontrollerile edasi ainult PoE toidet, nõnda vältimaks mikrokontrolleri kahjustamist. Üks edasiarendus punkte oleks uurida, kas säärane lüliti on PoE puhul võimalik ja kas see oleks ökonoomne.

6.1.2 Korpuse kohandamine

Selles lõputöös loodud korpus, vastab projekti raames paljudele nõudmistele mis lõpp tootes ei ole vajalikud. Antud korpuse ruumala, saab ühe mikrokontrolleri valides pooleks jagada, vältides ruumi raiskamist ja mass tootmisel materjali raiskamist. Lisaks võiks tulevikus korpusele sisse mudelleerida süvendid, mille abil saaks keermetatud sisendeid täpsemini sisse sulatada.

6.1.3 Korpuse ilmastikukindlus

Antud seadet hakatakse kasutama väli tingimustes, see tõttu tuleks korpusele määrata kohane IP standard. IP standard, tuntud ka kui IP kaitseklass või IP kood on arvuline väärtus mille järgi saab tooteid vastupidavuse järgi eristada. IP ehk *ingress protection* on standard, mis kirjeldab korpuse vastupidavust keskkonna teguritele. Vastupidavust kirjeldav kood koosneb kahest tähest ja kahest numbrist. Esimene number iseloomustab kaitset tahkete võõrkehade eest, teine number iseloomustab kaitset niiskuse eest [31].

Standard millele korpus võiks vastata on autori arvates IP33. IP33 tagaks piisava kaitse enamike tööriistade (nt kruvikeerajad) ja kaablite suuruste esemete eest. IP33 oleks ka piisav niiskuskaitse, et kaitsta seadet kerge vihma eest ning ei oleks liiga kulukas et toota.

7 Kokkuvõte

Käesoleva lõputöö üheks eesmärgiks oli luua mõõtesüsteemi seadmetele kasutajaliides. Valmis saadi kolme erineva mikrokontrolleri kasutajaliidestega. Kõigil mikrokontrolleritel on saavutatud saatmis- ja vastuvõtmisolek, mida saab kasutajaliidese abil valida. Mõlemal Raspberry Pi-l on võimalik olekut vahetada läbi graafilise kasutajaliidese, millele pääseb ligi kasutades VNC *viewer*'it. Graafilises kasutajaliidises on võimalik näha mõõtetulemuste keskväärtus, standardhälvet, joograafikut ja histogrammi. Mõlema Raspberry Pi ja ESP32-e puhul on näha OLED ekraanilt viimast mõõtetulemust, mõõtetulemuste keskväärtust ja standardhälvet.

Ainus tagasilöökk antud lõputöö tegemisel oli puuetundliku ekraani mitteühildumine kasutusel olevate mikrokontrolleritega. See viga võttis palju aega, et ümber teha ning sundis autorit minema algsest ajakavast kõrvale.

Toitelahendus lahendati PoE läbi, kõik mikrokontrollerid toetasid antud standardit. Toitelahenduse ainus lisa uurimist vajav koht oli Raspberry Pi 5-le PoE+ *hat*-i istutamine. Raspberry pole oma ametlikku PoE *hat*-i Raspberry Pi 5-le välja andnud, seega autor pidi Raspberry Pi 4-ja *hat*-iga hakkama saama. *Hat* suudeti uuema mikrokontrolleriga tööle saada, seega eesmärk on saavutatud.

Enne korpuse ehitamist, uuriti turul olemasolevaid korpuseid ja järeldas, et tema spetsiifilistele nõudmistele ükski neist ei vasta. Otsustati minna 3D printimise teed. 3D printimisel tekkis omajagu praak detaile, paljudel erinevatel põhjustel. Lõpptulemus on kaks funktsionaalsuselt identset korpust. Välimus on korpustel samuti identne, kui plastmassi värv välja arvata.

Saavutati seatud eesmärgid ning suhtlus juhendajaga oli pidev. Autor soovib tänada oma juhendajat Priit Roosipuud suurepärase juhendamise eest ja Kaiser Pärnametsa kasulike kommentaaride eest.

8 Kasutatud kirjandus

- [1] L. Kõrgmaa, A. Mändmets, K. Paabut ja A. Prääm, „Irdtorni videoühenduse viite mõõtmine,“ Taltech, Tallinn, 2023.
- [2] V. Technology, „What is Power Over Ethernet (PoE), and What is it Used for?,“ Versa Technology, [Võrgumaterjal]. Available: <https://versatek.com/what-is-power-over-ethernet/>. [Kasutatud 22 aprill 2024].
- [3] T. O. & O. R. Lab, „VNC - How it works,“ The Olivetti & Oracle Research Lab, [Võrgumaterjal]. Available: <https://web.mit.edu/cdsdev/src/howitworks.html>. [Kasutatud 22 aprill 2024].
- [4] „Power over Ethernet (PoE) Explained: PoE Standards, Types and Power Levels,“ Black Box, [Võrgumaterjal]. Available: <https://www.black-box.eu/en-int/page/23888/Resources/technical/Black-Box-Explains/Networking/poe-power-over-ethernet/>. [Kasutatud 4 mai 2024].
- [5] T. Networks, „Active PoE vs. Passive PoE: What's the Difference?,“ Teltonika Networks, 5 juuli 2023. [Võrgumaterjal]. Available: https://www.youtube.com/watch?v=mo2QOu8cK_I. [Kasutatud 22 aprill 2024].
- [6] X. L. Rillo, „Active or Passive PoE, That is the Question,“ Netgear, [Võrgumaterjal]. Available: <https://www.netgear.com/hub/business/network/active-or-passive/>. [Kasutatud 5 mai 2024].
- [7] U. I. t. C. G. -. C. 3490, „(Unit 0) Intro 4: Frame Buffer,“ 9 september 2020. [Võrgumaterjal]. Available: <https://www.youtube.com/watch?v=hEOvuwn2NdY>. [Kasutatud 2 mai 2024].
- [8] „RealVNC,“ RealVNC, [Võrgumaterjal]. Available: <https://www.realvnc.com/en/>. [Kasutatud 2 mai 2024].

- [9] Az-delivery, „0.96 inch OLED SSD1306 Display I2C 128 x 64 pixels Compatible with Arduino and Raspberry Pi,“ az-delivery, [Võrgumaterjal]. Available: <https://www.az-delivery.de/en/products/0-96zolldisplay>. [Kasutatud 22 aprill 2024].
- [10] „3.2inch 320x240 Touch LCD (D), With Touch Panel And Stand-Alone Controllers,“ Waveshare, [Võrgumaterjal]. Available: <https://www.waveshare.com/3.2inch-320x240-touch-lcd-d.htm>. [Kasutatud 5 mai 2024].
- [11] Waveshare, „SCHEMATIC1 : HY32E,“ [Võrgumaterjal]. Available: <https://files.waveshare.com/upload/9/93/3.2inch-320-240-Touch-LCD-SCH.pdf>. [Kasutatud 7 aprill 2024].
- [12] „Download RealVNC® Server,“ RealVNC, [Võrgumaterjal]. Available: <https://www.realvnc.com/en/connect/download/vnc/>. [Kasutatud 5 mai 2024].
- [13] „Download RealVNC® Viewer,“ RealVNC, [Võrgumaterjal]. Available: <https://www.realvnc.com/en/connect/download/viewer/>. [Kasutatud 5 mai 2024].
- [14] „diagrams.net,“ JGraph, [Võrgumaterjal]. Available: <https://app.diagrams.net>. [Kasutatud 9 mai 2024].
- [15] „ChatGPT,“ OpenAI, 14 märts 2023. [Võrgumaterjal]. Available: <https://chatgpt.com>. [Kasutatud 1 mai 2024].
- [16] „Copilot,“ Microsoft 365, 1 jaanuar 2024. [Võrgumaterjal]. Available: <https://copilot.microsoft.com>. [Kasutatud 1 mai 2024].
- [17] „Matplotlib: Visualization with Python,“ Matplotlib, [Võrgumaterjal]. Available: <https://matplotlib.org>. [Kasutatud 10 mai 2024].
- [18] DanKoloff, „ESP32-POE-user-manual.pdf,“ Olimex, 21 september 2023. [Võrgumaterjal]. Available: <https://github.com/OLIMEX/ESP32-POE/blob/master/DOCUMENTS/ESP32-POE-user-manual.pdf>. [Kasutatud 5 mai 2024].
- [19] „Raspberry Pi GPIO pinout,“ [Võrgumaterjal]. Available: <https://pinout.xyz>. [Kasutatud 5 mai 2024].
- [20] „Raspberry Pi 5 Pinout,“ Vilros, [Võrgumaterjal]. Available: <https://vilros.com/pages/raspberry-pi-5-pinout>. [Kasutatud 5 mai 2024].

- [21] „EasyEDA,“ EasyEDA, [Võrgumaterjal]. Available: <https://easyeda.com>. [Kasutatud 10 mai 2024].
- [22] „0,96 Zoll OLED Display Datenblatt,“ Az-Delivery, [Võrgumaterjal]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/0_96_Zoll_Display_Datenblatt_AZ-Delivery_Vertriebs_GmbH_241c4223-c03f-4530-a8c0-f9ef2575872f.pdf?v=1622442722. [Kasutatud 2 mai 2024].
- [23] „Paint,“ Microsoft, [Võrgumaterjal]. Available: <https://apps.microsoft.com/detail/9pcfs5b6t72h?hl=en-US&gl=US>. [Kasutatud 10 mai 2024].
- [24] DanKoloff, „ESP32-PoE-ISO_Rev_K.pdf,“ Olimex, 15 november 2022. [Võrgumaterjal]. Available: https://github.com/OLIMEX/ESP32-POE-ISO/blob/master/HARDWARE/ESP32-PoE-ISO-Rev.K/ESP32-PoE-ISO_Rev_K.pdf. [Kasutatud 29 aprill 2024].
- [25] „Raspberry Pi 5 - 4Gb version,“ Yadom, [Võrgumaterjal]. Available: <https://yadom.eu/raspberry-pi-5-version-4gb.html>. [Kasutatud 10 mai 2024].
- [26] „Raspberry Pi PoE+ HAT - Power over Ethernet for Raspberry Pi 4B/3B+ - SC0468,“ Botland, [Võrgumaterjal]. Available: <https://botland.store/raspberry-pi-hat-power-supply/19537-raspberry-pi-poe-hat-power-over-ethernet-for-raspberry-pi-4b-3b-sc0468-633696492820.html>. [Kasutatud 10 mai 2024].
- [27] „SCHEMATIC1 : REDUCED SCHEMATICS,“ Raspberry Pi Foundation, [Võrgumaterjal]. Available: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-reduced-schematics.pdf>. [Kasutatud 29 aprill 2024].
- [28] cstanton, „Raspberry Pi 5 Pinouts including GPIO for the 40 Pin Header,“ Element14, 17 oktoober 2023. [Võrgumaterjal]. Available: <https://community.element14.com/products/raspberry-pi/m/files/148385>. [Kasutatud 5 mai 2024].
- [29] „Dual-band WIFI BTH 5.0 Dual Micro Raspberry Pi 4 Computer Model B Pi4 4b 1GB 2GB 4GB 8GB,“ Alibaba, [Võrgumaterjal]. Available: https://www.alibaba.com/product-detail/1-2-4-8GB-Raspberry-Pi_62276292171.html. [Kasutatud 10 mai 2024].
- [30] „Case for Raspberry Pi 4B VESA v2 for monitor mounting - black,“ Botland, [Võrgumaterjal]. Available: <https://botland.store/raspberry-pi-4b-cases/14777->

case-for-raspberry-pi-4b-vesa-v2-for-monitor-5904422301378.html. [Kasutatud 7 mai 2024].

- [31] „IP koodi koostisosad ja nende tähendus,“ OÜ Dormikor, 25 aprill 2012. [Võrgumaterjal]. Available: https://www.oomipood.ee/article/26_ip_koodi_koostisosad_ja_nende_tahendus. [Kasutatud 16 märts 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

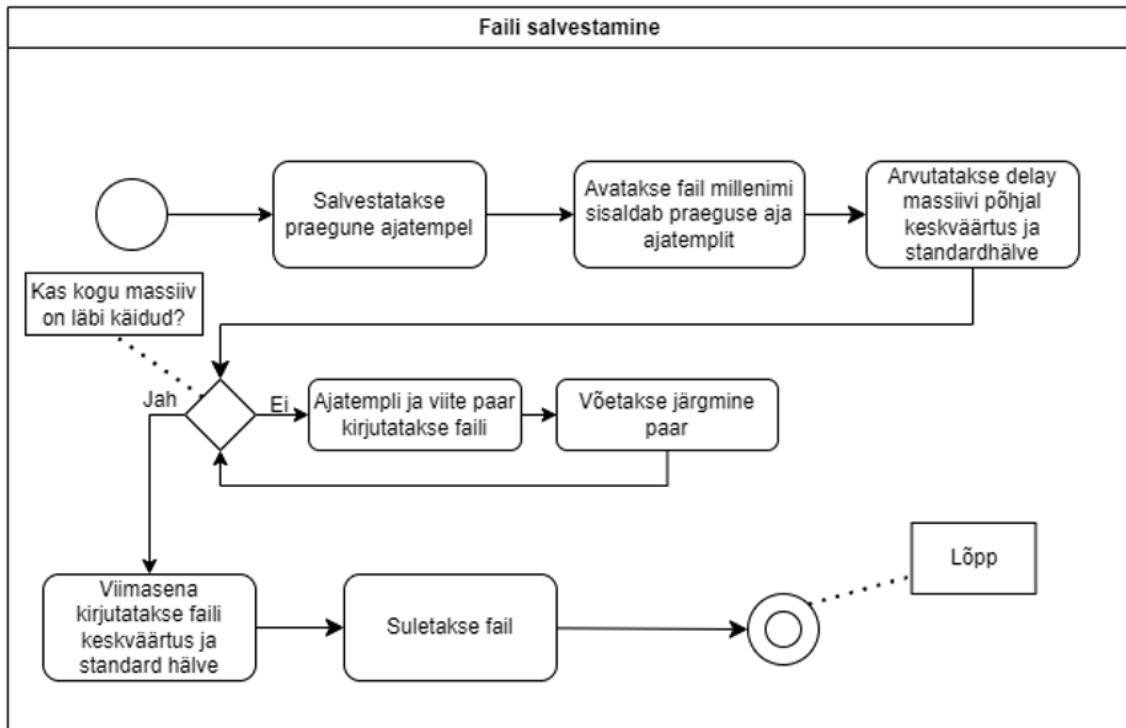
Mina, Karl Paabut

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Mõõteseadme kasutajaliidese disain ja toitelahendus videohilistuse mõõtmiseks“, mille juhendaja on Priit Roosipuu
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

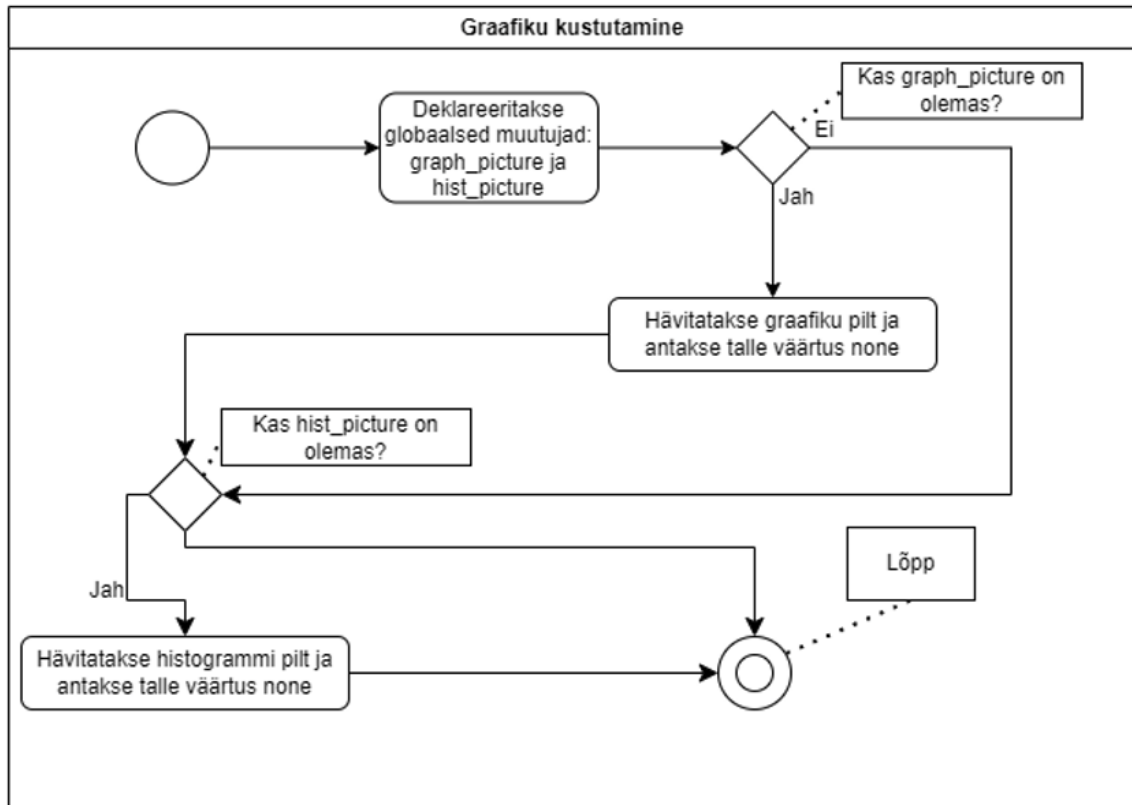
13.05.2024

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

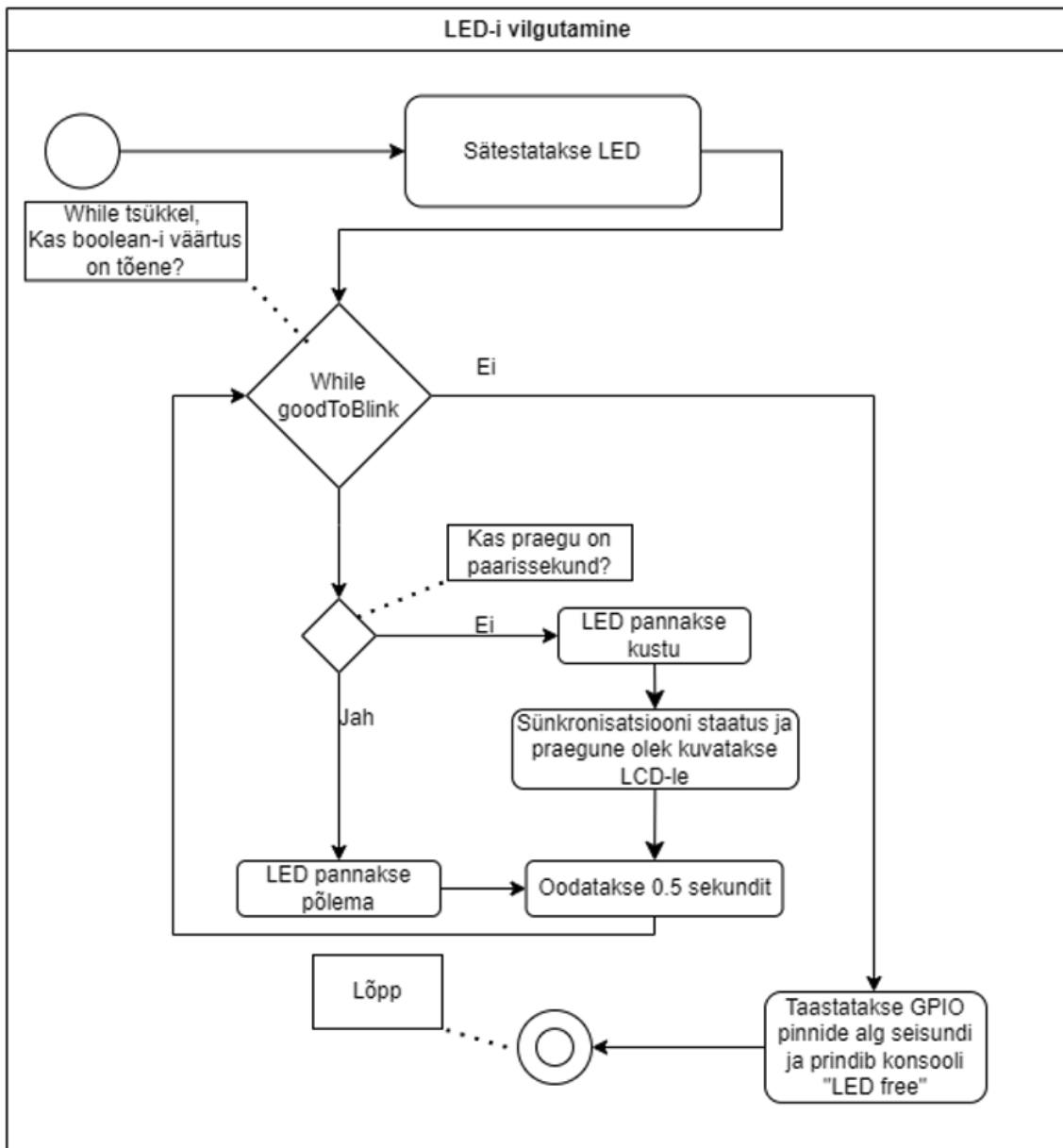
Lisa 2 – Faili salvestamise algoritm



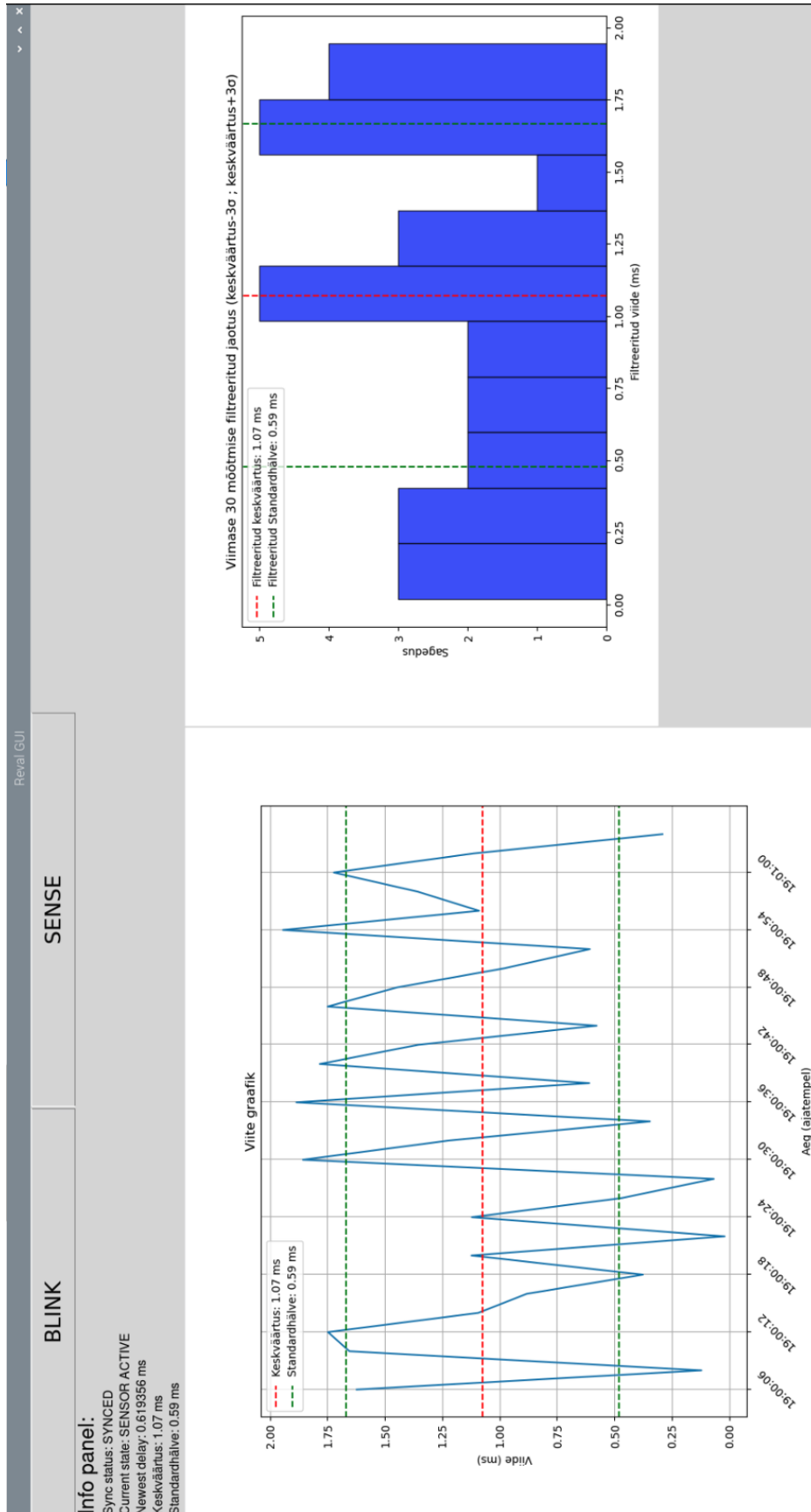
Lisa 3 – Graafiku kustutamise algoritm



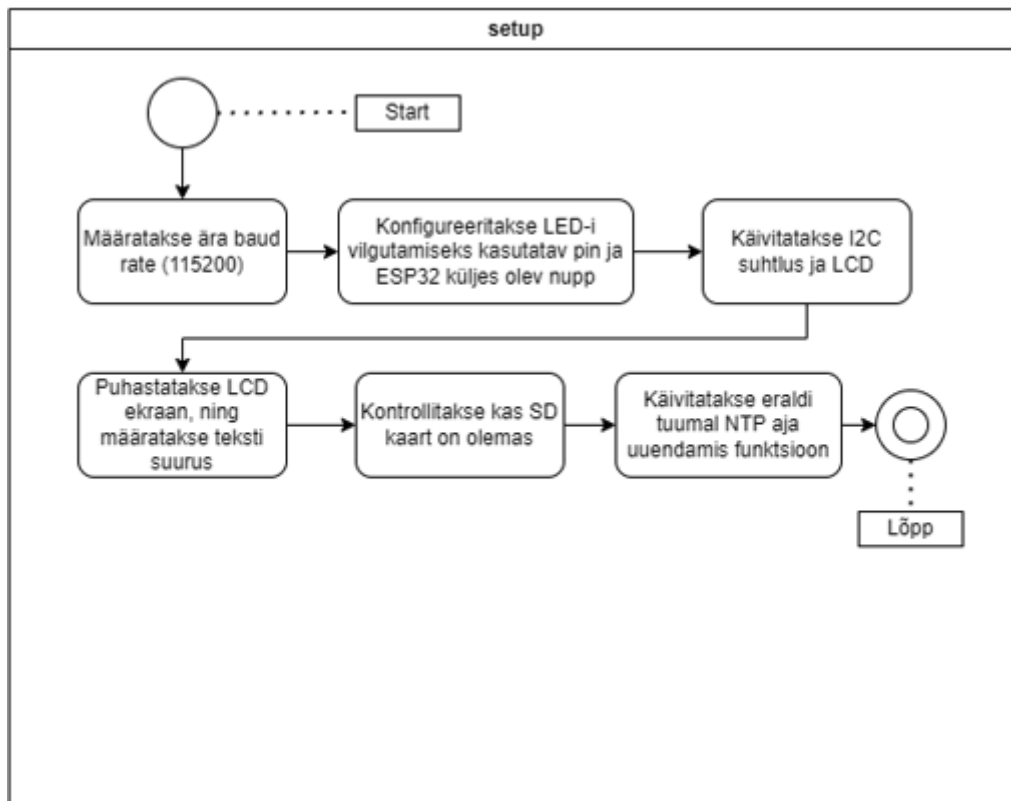
Lisa 4 – Valgusdiodi vilgutamise algoritm



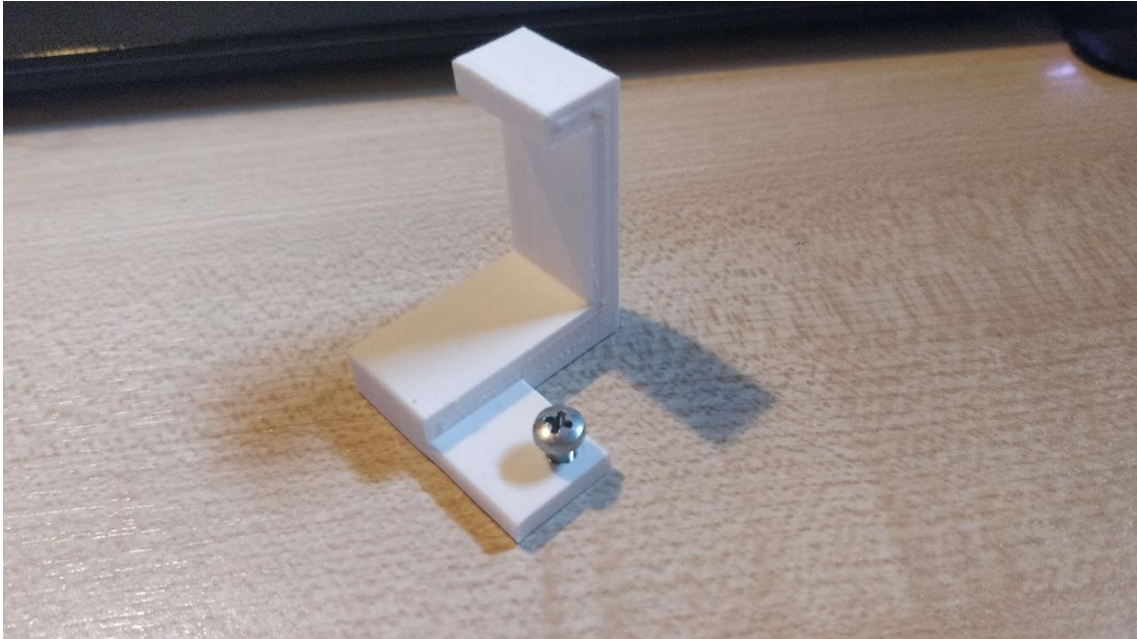
Lisa 5 – Raspberry Pi-de vastuvõtmissolek



Lisa 6 – ESP32 setup algoritm



Lisa 7 – Etherneti pistiku kinnitus



Lisa 8 – Kasutajaliideste koodid

Githubi link: <https://github.com/Klaabut/Video-latency-measurement-GUI>