

## Loppuraportti

# Savox - Wearable Thermal Camera (OPERAY Pro)



Päivämäärä: 31.8.2023

Henrik Toikka

Iiro Karppanen

# Informaatio sivu

## Ryhmä

Henrik Toikka  
Iiro Karppanen

## Yhteyshenkilö

Iiro Karppanen

## Sponsoroiva yritys

Savox Communications Oy

## Aloituspäivämäärä

1.6.2023

## Palautuspäivämäärä

31.8.2023

# Sisällysluettelo

Informaatio sivu.....	2
Sisällysluettelo.....	3
<b>Tiivistelmä.....</b>	<b>4</b>
<b>1. Vaatimukset ja tavoitteet.....</b>	<b>5</b>
1.1. Käyttäjakohtaiset vaatimukset.....	5
1.2. Kameroiden vaatimukset.....	5
1.3. Alustan valinta.....	5
1.2. Akunkesto ja kestävyys.....	6
<b>2. Suunnittelu.....</b>	<b>6</b>
2.1. Osaluettelo ja hinnasto.....	6
2.2. Elektroniikka ja laitteisto.....	6
2.3. Ohjelmisto ja koodit.....	7
<b>3. Piirilevy.....</b>	<b>7</b>
3.1. Kameraliitännät.....	7
3.2. GPIO pinnit.....	8
3.3. Latauspiiri.....	8
3.4. Testaus.....	9
3.5. Lopputulos.....	10
<b>4. Palvelin.....</b>	<b>11</b>
<b>5. Yhteydet (WebSocket ja HTTP API pyyntö).....</b>	<b>11</b>
<b>6. Laitteen ohjelmisto.....</b>	<b>12</b>
<b>7. Käyttöliittymä ja nettisivu.....</b>	<b>14</b>
<b>8. 3D mallinnus.....</b>	<b>14</b>
8.1. Kotelointi.....	14
8.2. Kiinnitys.....	16
<b>9. Projektitoiminta.....</b>	<b>16</b>
9.1 Tavoitteiden saavuttaminen.....	16
9.2. Aikataulu.....	17
9.3. Riskianalyysi.....	17
<b>10. Kehityskohteet.....</b>	<b>18</b>
10.1. Elektroniikan kehityskohteet.....	18
10.2. Mallinnuksen kehityskohteet.....	18
10.3. Ohjelmiston kehityskohteet.....	19
<b>Liitteet.....</b>	<b>20</b>
1. Kytkenäkaavion kuvat.....	20
2. Tulostetut ja leikatut mallit.....	22
3. Laitteen ohjelmiston asennus.....	22
4. Käytetyt ohjelmistot.....	23
5. Juliste.....	24
6. Linkit.....	25
<b>Lisenssi.....</b>	<b>25</b>
<b>Lähteet.....</b>	<b>25</b>

# Tiivistelmä

Tavoitteena on kehittää kannettava kameralaitte, jossa on integroituna lämpökameramoduuli sekä HD värikamera. Laitetta voidaan käyttää työkaluna etsintöihin, pelastus- ja poliisioperaatioihin. Laitte kykenee tunnistamaan kamerasyötteistä ihmisen lämpömassan ja laajalinssisellä kameralla alueen. Helppokäyttöinen kameralaitte voidaan helposti kiinnittää varustukseen erilaisilla liittimillä.

# 1. Vaatimukset ja tavoitteet

## 1.1. Käyttäjakohtaiset vaatimukset

Käyttäjän kannalta oleellista on laitteen helppokäyttöisyys. Laite kiinnitetään klipsillä varustukseen ja virtaa saatuaan menee automaattisesti päälle. Videon lähetys alkaa käyttöjärjestelmän avautuessa, joten laitteen käyttöönotto on nopeaa. Laitetta ei tarvitse erikseen operoida vaan se toimii autonomisesti taustalla. Laitetta pystyy ladata, joten sen kanssa ei tarvitse käyttää ulkoista akkua.

Käyttöliittymän on oltava helppokäyttöinen, joka myös näyttää hyvältä.

Etäoperaattorin ja laitteen haltijan on kyettävä nopeasti näkemään kameroiden syötteet riittävän nopealla resoluutiolla, taajuudella ja viiveellä.

## 1.2. Kameroiden vaatimukset

Lämpökameran on kyettävä tunnistamaan ihmisen lämpömassa selvästi. HD kamera kuvaa mahdollisimman laajalta alueelta. Tarpeeksi hyvän resoluution lisäksi lämpökameran toisena kriteerinä on sopiva virkistystaajuus, jotta sen kuvaamaa videota on miellyttävää seurata ja etäoperaattorilta ei jää hetkeäkään huomaamatta. Osassa markkinoilta löytyvistä kameramoduuleista on vain 9hz virkistystaajuus, joten ne karsittiin pois.

Valitsimme lämpökameraksi Infray P2 Pro:n. Kyseisestä moduulista löytyy kaikki tarvittavat ominaisuudet ja se on USB C liittimen ansiosta helppo yhdistää muuhun laitteistoon. Arvioimme, että P2 Pro:ssa on riittävä resoluutio ihmismuodon tunnistamiseen.

## 1.3. Alustan valinta

Projektin alussa harkitsimme ESP32:a kameroiden kuvien käsittelyyn ja niiden lähettämiseen eteenpäin. Kaista-analyysi laskelmien jälkeen oli selvää, että tarvitsemme enemmän prosesointitehoa. ESP32 ei pysty lähettämään samaan aikaan molempien kameroiden kuvaa tarvittavalla nopeudella.

Viikkopalaverissa käytyjen keskustelujen pohjalta Raspberry Pi Compute Module 4 osoittautui sopivaksi vaihtoehdoksi. Compute Moduuli täyttää ja ylittää tarvittu muistivaatimukset, mutta on silti kompakti. Siinä ei myöskään ole mitään USB portteja eikä muitakaan liitäntöjä valmiina, joten sille suunnitellaan oma piirilevy. Piirilevystä tulee samankokoinen kuin itse Compute Module:sta ja ne yhdistetään toisiinsa.

## 1.2. Akunkesto ja kestävyys

Laitetta käytettäisiin pitkissä operaatioissa, jotka saattaisivat kestää useampia tunteja. Pyrimme siis projektissa saavuttamaan mahdollisimman pitkän akunkeston ja teemme akun lataamisesta helppoa. Akunkestotavoitteena on yksi kokonainen työpäivä eli noin 8 tuntia.

Laitteen on oltava kohtuullisten iskujen kestävä, jolloin laitteen käyttäjän ei tarvitse aktiivisesti varoa tuhoamasta laitetta. Vedenkestävyys olisi käytön kannalta kriittistä, mutta tähtäämme pieneen vedenkestävyyteen tässä prototyypissä.

## 2. Suunnittelu

### 2.1. Osaluettelo ja hinnasto

Infiray P2 Pro (262.92€)

- 252x192 resoluutio

Raspberry Pi Camera module 3 Wide (32.24€)

- 640x480 - 4608x2592 resoluutio

Raspberry Pi Compute Module 4 (64,40€)

Raspberry Pi Compute Module 4 IO Board (Testausta varten)

USB-C female to USB-A male adapteri (Testausta varten)

LiPO akku 1600 mAh (noin 15€)

Tilattu PCB (noin 38€)

Yhden laitteen kokonaishinta on ~ 348€

### 2.2. Elektroniikka ja laitteisto

Raspberry Pi Compute Module 4 yhdistetään muuhun piiriin sille suunnitellulla piirilevyllä, jossa on liitännät kameramoduuleille. Levy sisältää virtaportin, jolla voi

ladata akkua. PCB ja kamerat ovat sisäistetty 3D-printattuun PLA koteloon. Kotelon takana on helposti liitettävä liitoskomponentti, joka on helposti irroitettavissa. Liitoskomponentin voi yhdistää Molle-liitoksella esimerkiksi liiviin.

## 2.3. Ohjelmisto ja koodit

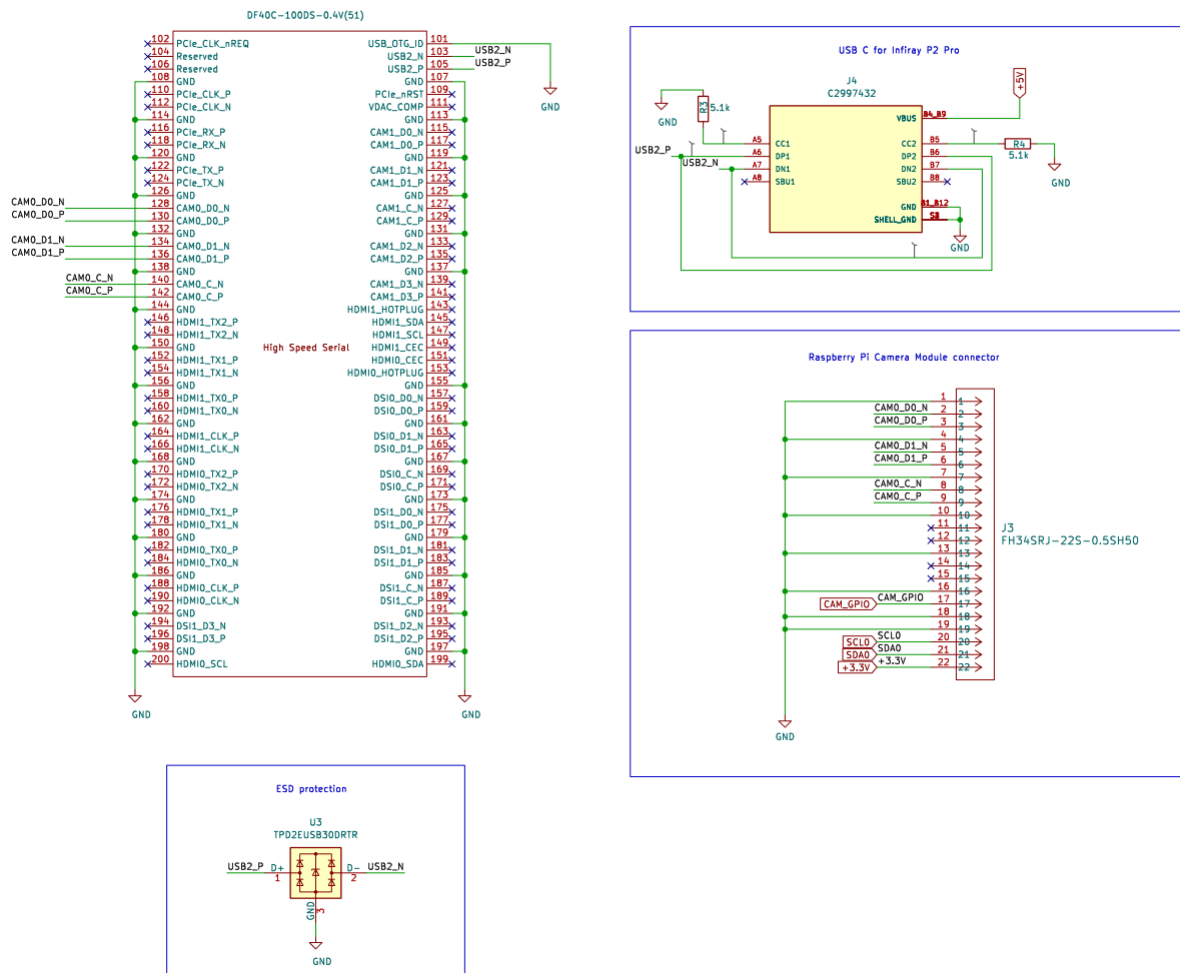
Compute Module:en asennetaan RaspOS Lite. Laitteeseen voi yhdistää SSH-yhteyden kautta palvelimen puolelta. Laite avaa datalähetykseen liittyvät ohjelmakoodit automaattisesti crontab-työkalulla, kun käyttöjärjestelmä käynnistyy. Palvelin kykenee ottamaan vastaan syötteitä useammasta laitteesta samanaikaisesti, ja lähettämään ne loppukäyttäjille käyttäen TLS-suojattua WebSocket-protokolla yhteyttä.

# 3. Piirilevy

## 3.1. Kameraliitännät

Lämpökamera yhdistetään USB C portin kautta piirilevyyn. USB C portista on vedetty Compute Modulen liittimiin differentiaalipari, jolle on laskettu oikea impedanssi. USB datalinjojen välisen differentiaaliimpedanssin tulee olla  $90 \Omega \pm 15\%$ , jotta signaalit kulkevat oikein. Lämpökameran USB portin eteen on laitettu ESD suojauskomponentti, joka suojaa kallista kameramoduulia sähköstaattisilta purkauksilta.

HD kamera on yhdistetty 22 pinnisellä fpc liittimellä, joka on tyypillinen ratkaisu Raspberry Pi Camera Moduulien kanssa. Samanlaisia liittimiä löytyy monista Raspberry Pi:n omista piirilevyistä.



(Kuva 1. Skematikka liittimistä)

### 3.2. GPIO pinnit

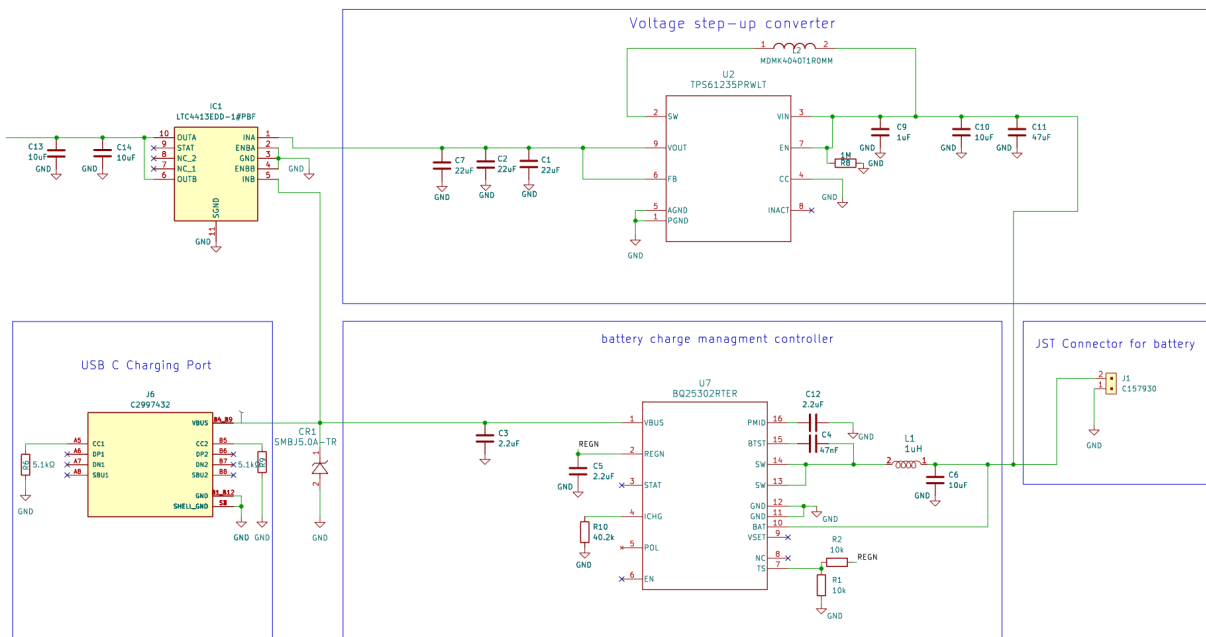
Halusimme lisätä laitteeseen nappuloita, LED valoja ja värisijän. Näitä kaikki voi hallita GPIO pinnien avulla. Compute Modulesta kiinnitettiin tarpeen mukaan linjat GPIO headerin käyttöön.

### 3.3. Latauspiiri

Koska laitteesta piti tehdä kannettava, halusimme siihen sisäisen akun, jota on helppo ladata. Piirilevyyn on jätetty paikka JST liittimelle, johon voi kiinnittää 3.7v Li-Po tai Li-Ion akun. Piirissä on akun lataamista valvova hallintasiru, joka syöttää akulle latausvirtaa oikealla profiililla. Jänniteregulaattori nostaa akun jännitteen 5 volttiin, jota Compute Module tarvitsee. Akkua lataava hallintasiru ja



jänniteregulaattori tarvitsivat ympärilleen paljon kondenssattoreita, vastuksia ja käämejä, joka monimutkaisti piiriä.



(Kuva 2. Skematiikka latauspiiristä)

### 3.4. Testaus

Tilatuissa piirilevyissä suurin osa komponenteista oli valmiiksi ladottuna kiinni, mutta osa piti kiinnittää itse. Tllasimme piirilevyjä kolmesti kurssin aikana.

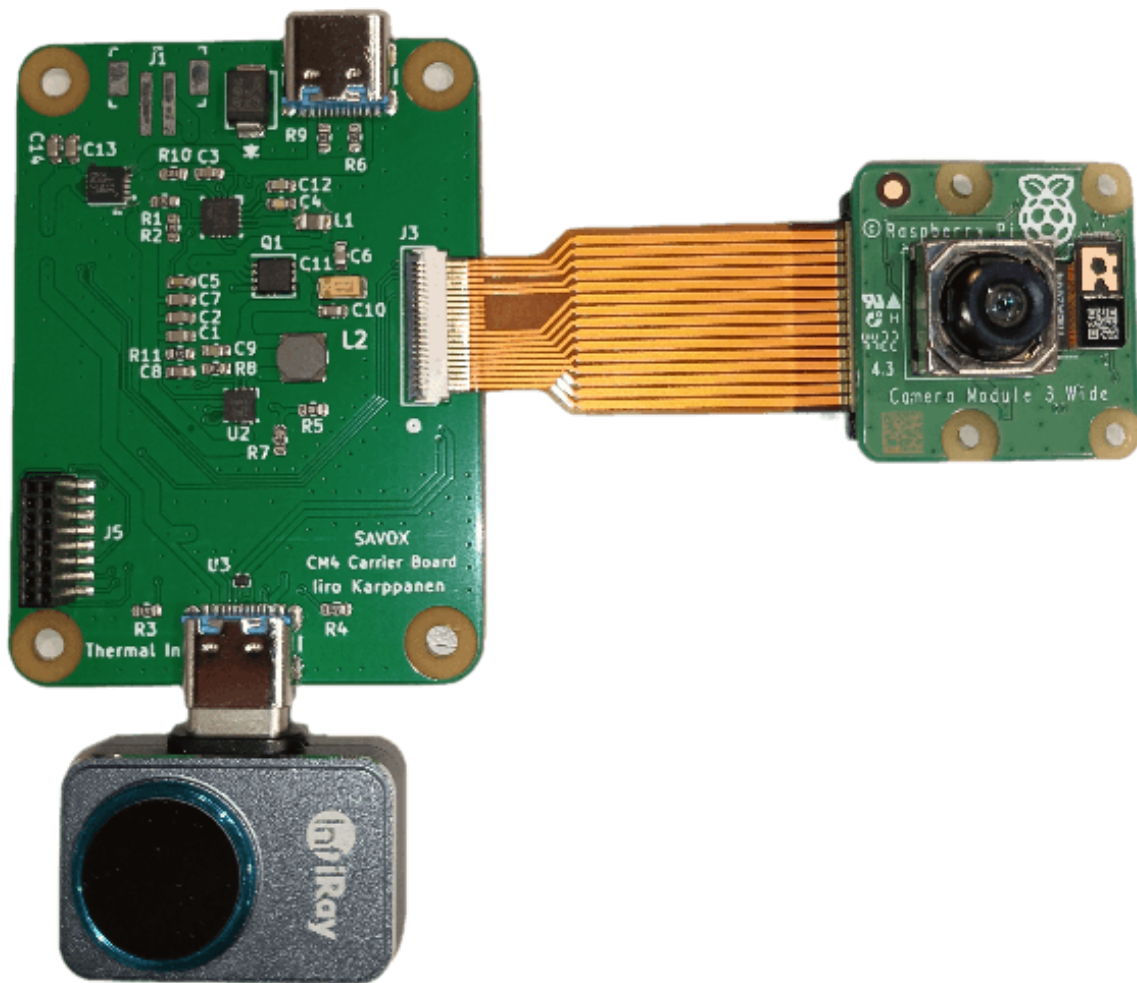
Ensimmäinen versio tilattiin 07.07.2023. Tässä versiossa meillä oli vaikeuksia kiinnittää Compute Moduulin liittimiä kiinni, joten emme päässeet testaamaan käynnistyykö laite vai ei. Levystä löytyi myöhemmin myös useita suunnitteluvirheitä latauspiirissä.

Toinen versio tilattiin 21.07.2023 ja siihen saatiin tarvittavat liittimet valmiiksi ladottuna. Tässä piirissä oli huolimattomuusvirhe ja laite ei käynnistynyt. Compute Moduulin liittimien gnd pinnit eivät olleet yhteydessä maatasoon, mikä esti käynnistymisen.

Kolmas versio tilattiin 11.09.2023 ja se saatiin toimimaan kuten pitääkin. Tässäkin versiossa oli tehty joitakin huonoja valintoja suunnittelun aikana, mutta ne ratkaistiin

ohjelmisto puolella. Piiriä suunnitellessa HD kameraportti oli yhdistetty Compute Modulen cam0 pinniin, joka on toissijainen kameraliitäntä. Kamera olisi kannattanut yhdistää cam1 pinniin. Compute Module:ssa on valmiiksi tuki cam0 pinnissä I2C1 väylälle, mutta suunnitellessa oli laitettu se käyttämään I2C0 väylää. Näiden ratkaisujen takia Compute Module ei heti tunnistanut HD kameraa. Laite suorittaa käynnistyessä config.txt tiedoston, jossa sille voidaan antaa erilaisia komentoja. Tähän tiedostoon piti lisätä komento joka pakottaa Compute Modulen käyttämään oikeaa I2C väylää. Lisäksi piti laittaa toinen komento, joka aktivoi cam0 liitännän.

### 3.5. Lopputulos



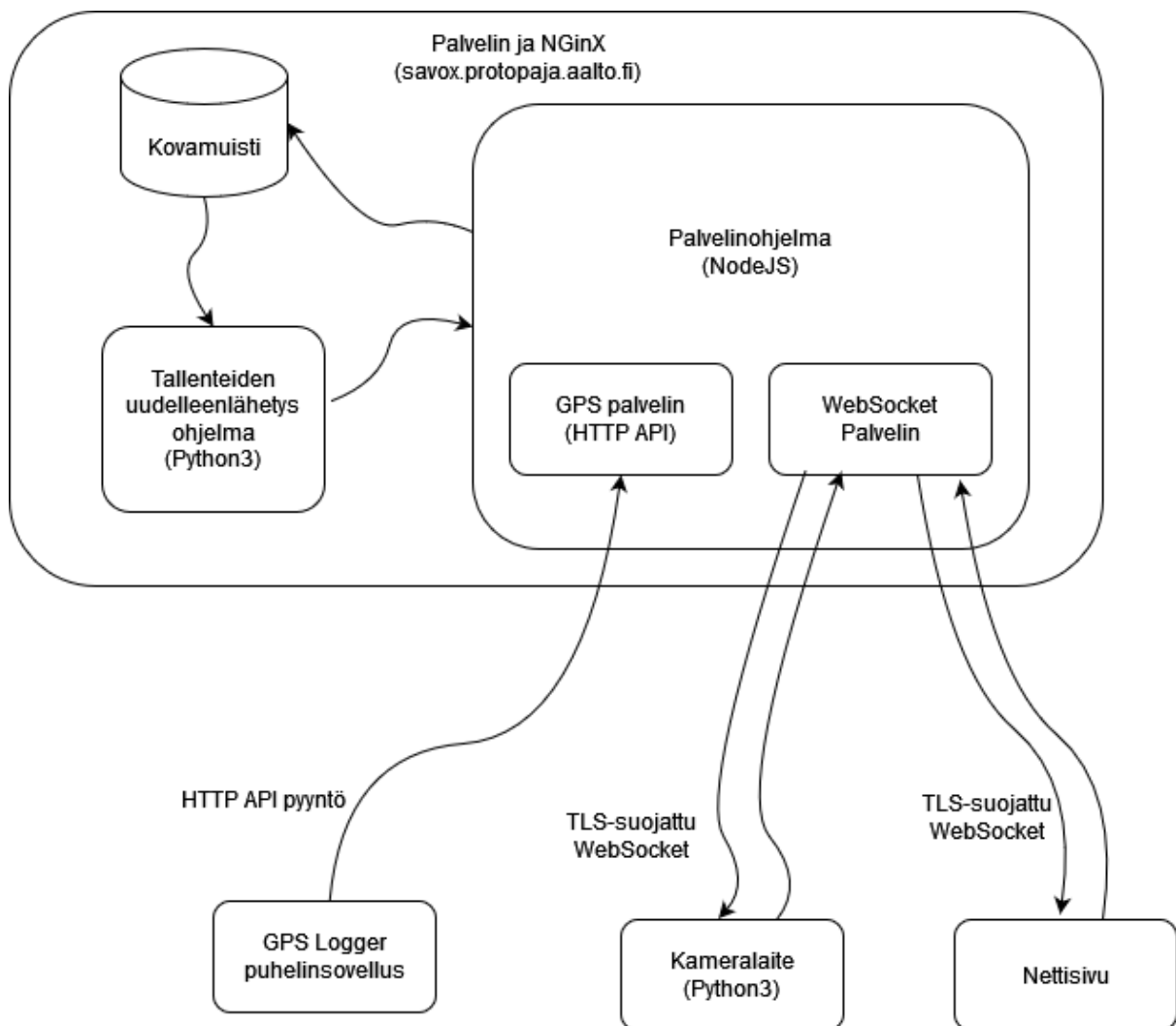
(Kuva 3. Kuva koteloimattomasta omasta piirilevystä)

## 4. Palvelin

Käytimme Protopaja-kurssin tarjoamaa protopaja.aalto.fi -sivua. Palvelintarjoajaksi asennettiin NGINX, joka lähettää HTML/CSS/JS tiedostot käyttäjille. Palvelinohjelma avaa WebSocket palvelimen, ja vastaanottaa laitteilta dataa. Palvelinohjelma välittää datan kaikille nettisivun käyttäjille. Palvelimella voi tallentaa kuvadataa laitteista ja erillisellä Python ohjelmalla tallenteita voidana lähettää nettisivun käyttäjille. Palvelinkoodi on kirjoitettu NodeJS frameworkilla.

Sijaintitieto saadaan GPS Logger -puhelinsovelluksesta, joka tekee noin sekunnin välein HTTP API pyynnön palvelimeen.

## 5. Yhteydet (WebSocket ja HTTP API pyyntö)



(Kuva 4. Datansiirto diagrammi ohjelmien ja laitteiden välillä)

Suurin datansiirto tapahtuu WebSocket protokollan välityksellä, joka on TLS-salattu ja binäärimuotoinen. Haluttiin WebSocket binaarimuodossa jolloin datansiirto voidaan minimoida. Tämä vaatii että jokainen muuttuja (kuvadata, kuvankoko, laitteen nimi, ym.) lähetetään tietyssä muodossa.

Yhden muuttujan muoto koostuu neljästä osasta:

- Muuttujan nimen tavupituus (32 unsigned int, Big Endian)
- Muuttujan nimi (utf-8 string)
- Datan tavupituus (32 unsigned int, Big Endian)
- Data (tavulista)

Jokainen muuttuja voidaan yhdistää peränjälkeen yhdistettyyn binääriseen muotoon, joka voidaan lähettää yhtenä pakettina palvelimelle.

Laitteen API token laitetaan WebSocket yhteyden header osioon, johon myös laitetaan laitteen nimi. Header osiosta palvelin voi myös lukea esimerkiksi mistä URL-osoitteesta laite tai nettikäyttäjä on yhdistänyt.

Sillä kameralaitte ei sisällä GPS anturia, niin käytettiin matkapuhelinta GPS sijainnin keräämiseen. Käytettiin GPS Logger -sovellusta, jolla sijainnin voi lähettää palvelimelle on HTTP API pyynnöllä, johon yhdistetään pituus- ja leveysasteet HTTP parametrina. Lisäksi tämän header osioon laitetaan laitteen nimi, jolloin matkapuhelin voi toimia tietyn laitteen GPS anturina.

## 6. Laitteen ohjelmisto

Laitteelle asennettiin RaspOS Lite. Laite yhdistää automaattisesti laitteelle konfiguroituun lähiverkkoon, jonka jälkeen se avaa palvelimeen automaattisesti SSH yhteyden. Laite avaa Python kielellä kirjoitetut kameraluku ja -lähetysohjelman, joka yhdistää WebSocket palvelimeen. Jos ohjelmat kaatuvat, niin Crontab työkalu avaa ohjelmat uudelleen minuutin välein. Kameroiden lukemiseen käytetään

python3-picamera2 ja python3-opencv kirjastoja, ja lähetykseen python3-websockets kirjastoa.

Laitteilla on laitenimi (devicename), jonka avulla palvelin ja nettikäyttäjä voi erottaa eri kameralaitteet. Palvelimella on myös salainen API token (esim. "Secret\_token 123"), jonka avulla palvelin voi päästää ja estää laitteita lähettämästä dataa eteenpäin.

Yhdellä laitteen silmukan aikana tapahtuu seuraavat asiat:

- Lukee kameroiden syötteet
- Laittaa tarvittavat muuttujat yhdistettyyn binääriseen muotoon (Katso osio 6.)
- Lähettää muuttujat palvelimelle

Laitteelta voidaan lähettää käytännössä minkälaista dataa halutaan, mutta datansiirto halutaan minimoida. Yksi paketti sisältää seuraavat parametrit:

- Laitenimi (devicename)
- Lämpökameran data (therm\_data)
- Värikameran data (hd\_data)
- Lämpökameran resoluutio (therm\_w ja therm\_h)
- Värikameran resoluutio (hd\_w ja hd\_h)

## 7. Käyttöliittymä ja nettisivu



(Kuva 5. Nettisivun karttanäkymä)

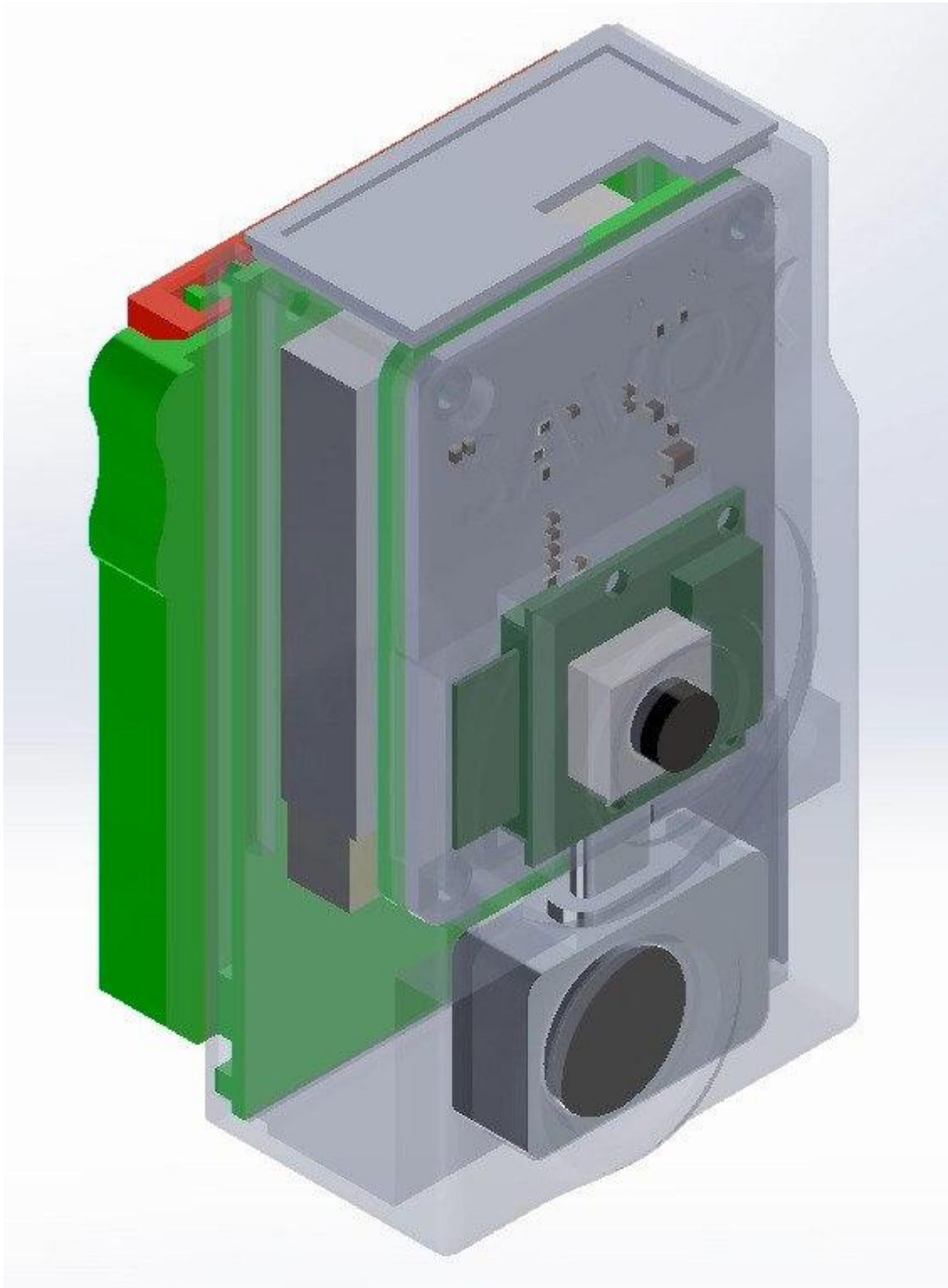
Nettisivulla on lista kameroista joista voi yksitellen valita kameran nimensä mukaan. Kustomisoitavalla karttaominaisuudella voi katsoa eri kameroiden syötteitä, kun kameroiden pallojen päälle siirtää hiiren. Kameroita voi myös seurata kokonäyttötilassa.

Nettisivu vastaanottaa JPEG-tiedoston, jonka se päivittää HTML canvas -elementille. Nettisivulla kameroiden syötteet näkyvät mittausten mukaan noin 25fps.

## 8. 3D mallinnus

### 8.1. Kotelointi

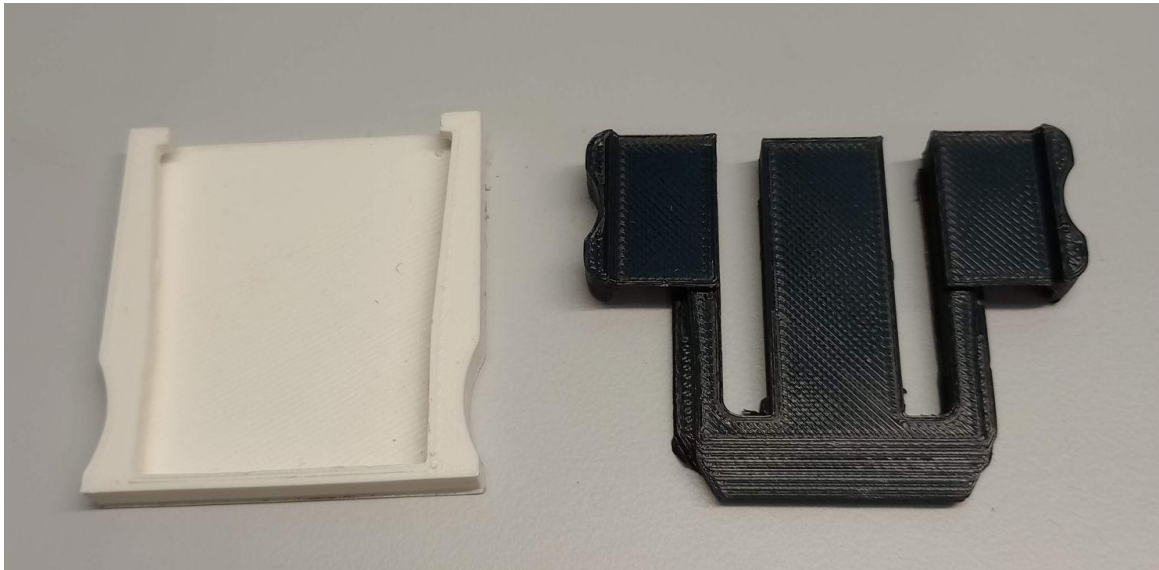
Ulkokuori mallinnettiin SOLIDWORKS:illa ja printattiin PLA-materiaalilla Ultimaker S3 ja S5 3D-printtereillä. Karheaan ulkopintaan käytettiin Ultimaker Slicer ohjelman "Fuzzy skin" asetusta. Kotelon koko on 47x35x85 mm. Koteloinnin seinät ovat paksuudeltaan 1.5mm, ja tekstin syvyys on 0.6mm. Lopulliseen mallinnukseen ei lisätty linssisuojava tai vesitiivisteitä.



(Kuva 6. Lopullinen mallinnus SOLIDWORKS näkymässä)

## 8.2. Kiinnitys

Laitteen kiinnityspala (valkoinen) voi yhdistää liiviin Molle kiinnityksellä. Laitteen puolella oleva liitoskohta mallinnetaan suoraan kotelon takakanteen. Laitteen voi irroittaa liivipalasta kiinnitysosan “korvia” painamalla. Liitoksen kokonaissyvyys on noin 12mm.



(Kuva 7. Mukavasti klikkautuva kiinnityskomponentti. Valkoinen osa on kiinni liivissä ja musta kiinni laitteen takakannessa)

## 9. Projektitoiminta

### 9.1 Tavoitteiden saavuttaminen

Suurin osa tavoitteista tuli saavutettua, kuitenkin projektin alkuperäinen visio muuttui jonkin verran. PCB saatiin lämpökamera ja normaali kamera toimimaan, mutta akkupiiri ei toimi odotetulla tavalla. Täten täysi integrointi ei ole mahdollista. Älytoimintoja ei ehditty toteuttamaan. Laite toimii tavoitteen mukaan täysin langattomasti ja videot lähettyvät reaaliaikaisesti palvelimelle pienellä viiveellä.



## 9.2. Aikataulu

Ohjelmistopuolella saimme kaiken valmiiksi demopäivään mennessä. Kameroiden kuvien lähetys serverillä ja sieltä nettisivulle onnistuu hyvin ja kartalla näkyvät laitteiden sijainnit.

Piirilevyn kanssa oli useita haasteita ja emme kerenneet integroimaan sitä laitteeseen demopäivään mennessä. Demopäivän jälkeen kamerat saatiin toimimaan. Aikataulu venyi siis tässä jonkin verran.

## 9.3. Riskianalyysi

Projektisuunnitelmaan listatuista riskeistä osa toteutui. Yksi ryhmäläinen lopetti projektin alussa kurssin ja aikaraja on ylittynyt joissakin projektin osa-alueissa. Olimme osanneet ennakoida mahdolliset riskit melko hyvin suunnitelmassa, ja improvisoitiin demopäivää varten testi laitteella.

Budjetista käytettiin kaikki ja se ylittyikin hieman, eli riski budjetin loppumisesta toteutui. Saimme kuitenkin kaikki tarvittavat komponentit ja piirit tilattua, joten tämän riskin toteutumisesta ei aiheutunut mitään seuraamuksia.

Suunnitelmaan listatuista riskeistä puuttuu ainoastaan tilanne, jossa projektiin suunniteltu piirilevy ei toimi, joka on hyvinkin mahdollista. Tässä tilanteessa joudutaan käyttämään aikaa ongelmien korjaamiseen ja uuten tilaukseen, josta voi tulla 1-3 viikon viivästys.

# 10. Kehityskohteet

## 10.1. Elektroniikan kehityskohteet

Laitteesta haluttiin mahdollisimman pieni, joten kameramoduulien kannattaisi olla pienempiä ja suoraan piirilevyssä kiinni. Meidän budjettiin ei löytynyt sopivaa smd tyyppistä lämpökameramoduulia. Valitsimme siis isomman moduulin, joka on tarkoitettu kiinnitettäväksi suoraan älypuhelimien usb porttiin kiinni. Moduulissa on paksu ulkokuori, joka vie paljon turhaa tilaa.

Laitteessa käytetyssä Compute Module 4:ssa on prosessointitehoa enemmän mitä tarvitaan ja Compute Modulessa itsessään on paljon meidän projektille turhia toimintoja. Olisimme voineet valita jonkin irrallisen prosessorin ja suunnitella sen ympärille Compute Modulea huomattavasti pienemmän piirin. Esim. STM32-sarjasta olisi voinut löytyä molempien videoiden käsittelyyn tarpeeksi tehokas, mutta pieni prosessori. Tässä tapauksessa piirilevyn suunnitelu olisi kuitenkin ollut huomattavasti haasteellisempaa ja etenkin kurssin alussa meillä ei ollut tarpeeksi osaamista tälläisen valinnan tekemiseen.

## 10.2. Mallinnuksen kehityskohteet

Lopullisessa mallinnuksessa ei käytetty linssisuojaaja tai vesitiivistettä.

Testimallinnuksissa tehtiin valmis tiiviste TPU-95a filamentista, ja linssisuoja.

Haluttiin pitää lopullisen mallinnuksen koko absoluuttisessa minimissä niin näitä ei lisätty.

Alunperin tarkoituksena oli tehdä matalampi mallinnus, jossa kamerat ovat symmetrisesti mallinnuksen etukannessa päällekkäin. Tämä vaatisi erillisen USB-johdon, joka pitää ottaa huomioon PCB-mallinnuksessa. Ehdittiin tilata piiri missä tämä mahdollisuus jää pois, niin tämä jää kehitysideaksi.



(Kuva 8. Mahdollinen pienempi kotelointi, joka sisältää akryylisen värikamerasuojan ja TPU-95a materiaalista tulostetun vesitiivisteen. Tyylikkäämpi ja symmetrinen ulkonäkö)

### 10.3. Ohjelmiston kehityskohteet

Ei ehditty tähän hätään tekemään ihmisen tunnistamista. Tämä toteutettaisiin palvelimen puolella käyttämällä OpenCV mediapipe tai YOLO malleja. Ideana on, että kun lämpökamerassa havaitaan ihminen, mutta ihminen ei näy värikamerassa, niin palvelin lähettää laitteelle siitä ilmoituksen automaattisesti.

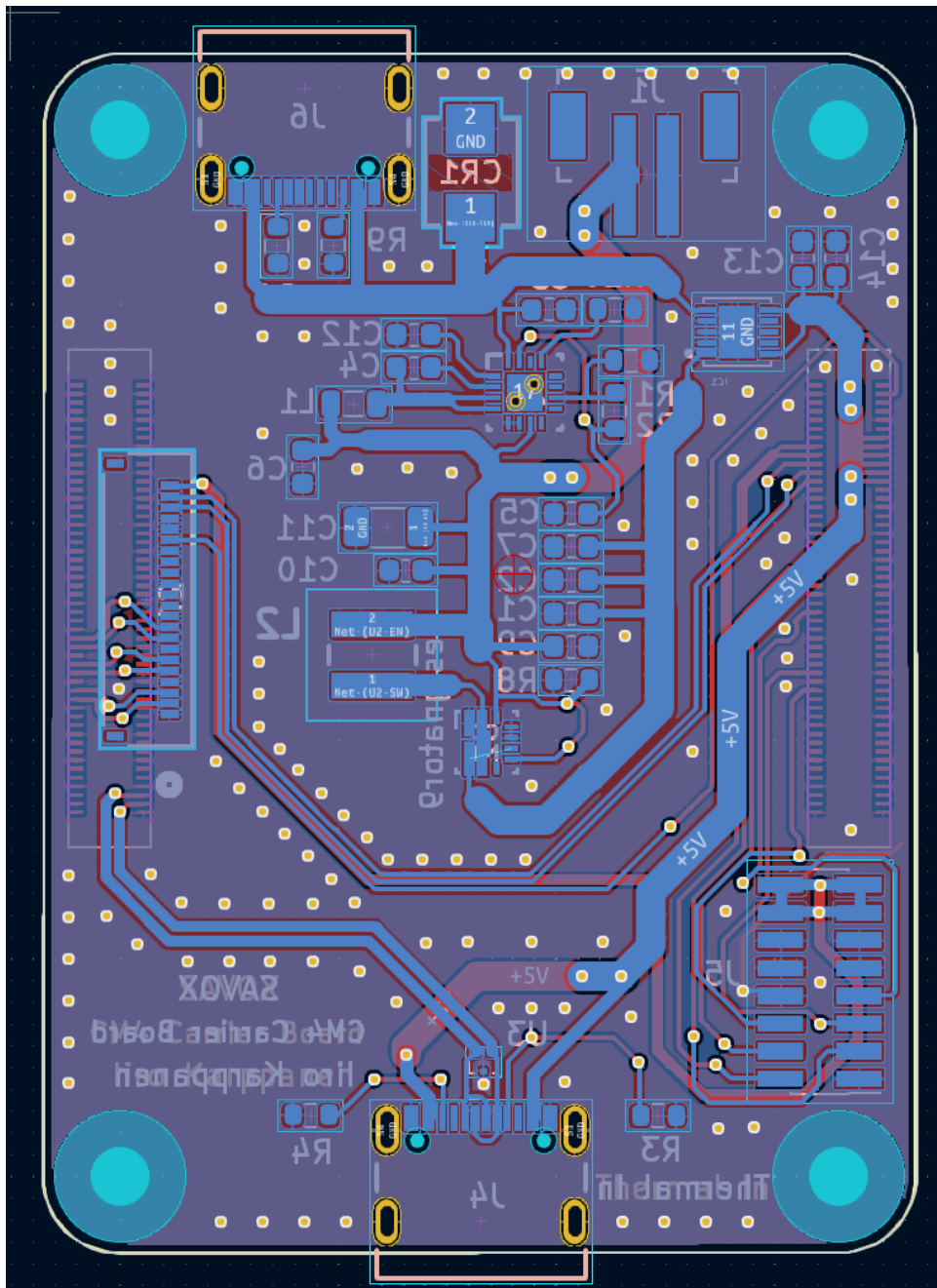
Värikamera- ja termokamera syötteet olisi myös hyvä sovittaa yhteen. Esimerkiksi jos havaitaan ihminen tietyllä alueella, niin värikameran syötteessä piirretään esimerkiksi ulkoraja ihmiselle, tai jollain muulla tavalla yhdistetään kuvat.

Tämänhetkinen laitteen virran saamisesta syötteiden lähetykseen kuluu noin 80 sekuntia, joka voisi olla paljon nopeampi. Lisäksi WebSocket yhteys välillä pätkii. Parempi olisi, että kameroiden syötteet lähetettäisiin esimerkiksi MJPEG streamilla, eikä yksittäisillä JPEG kuvina.

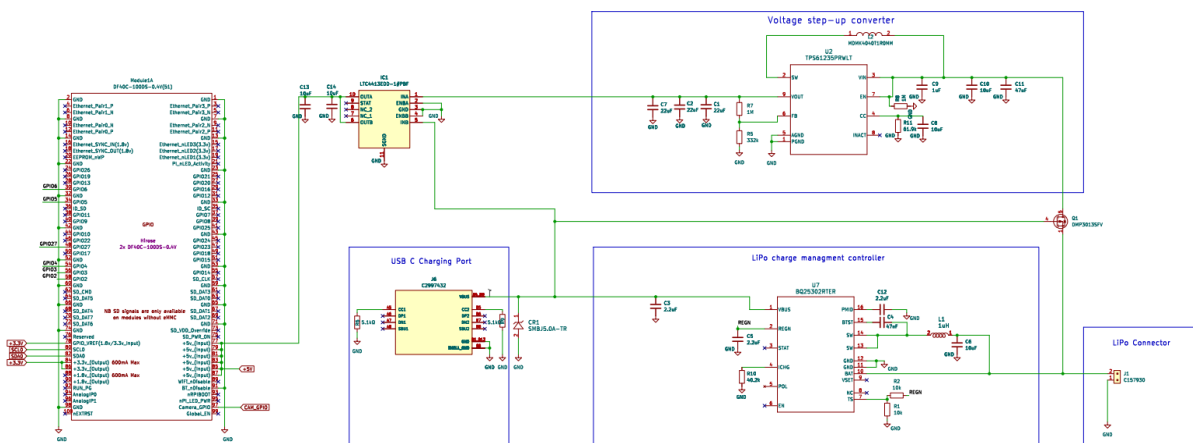
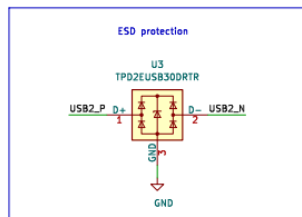
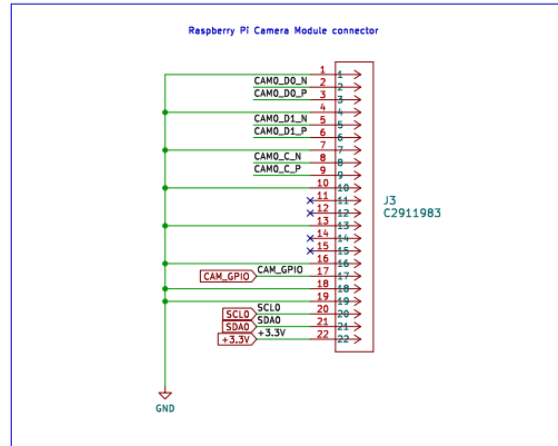
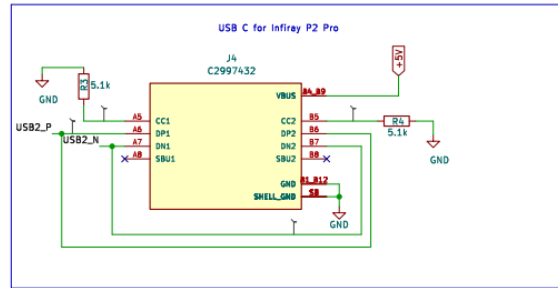
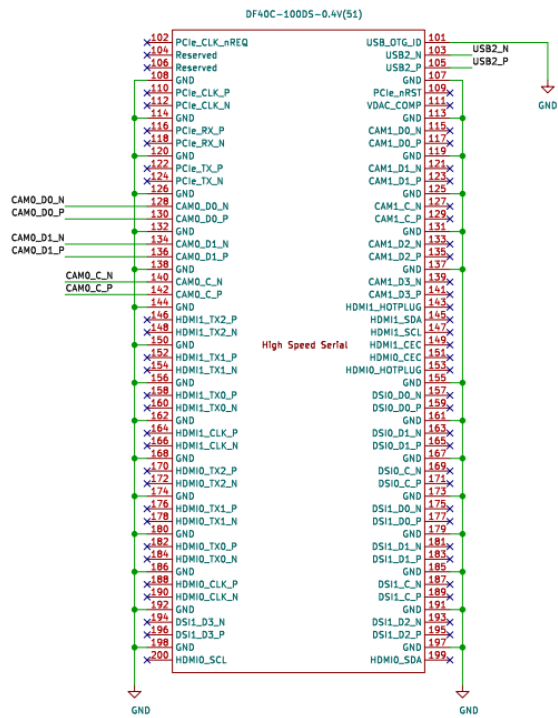
Palvelin lähettää jokaisen kamerasyötteen jokaisesta kamerasta jokaiselle nettikäyttäjälle. Tämä tietenkään ei ole hyvä, sillä nettikäyttäjä katsoo ainoastaan yhden kamerasyötteen kerrallaan.

# Liitteet

## 1. Kytentäkaavion kuvat

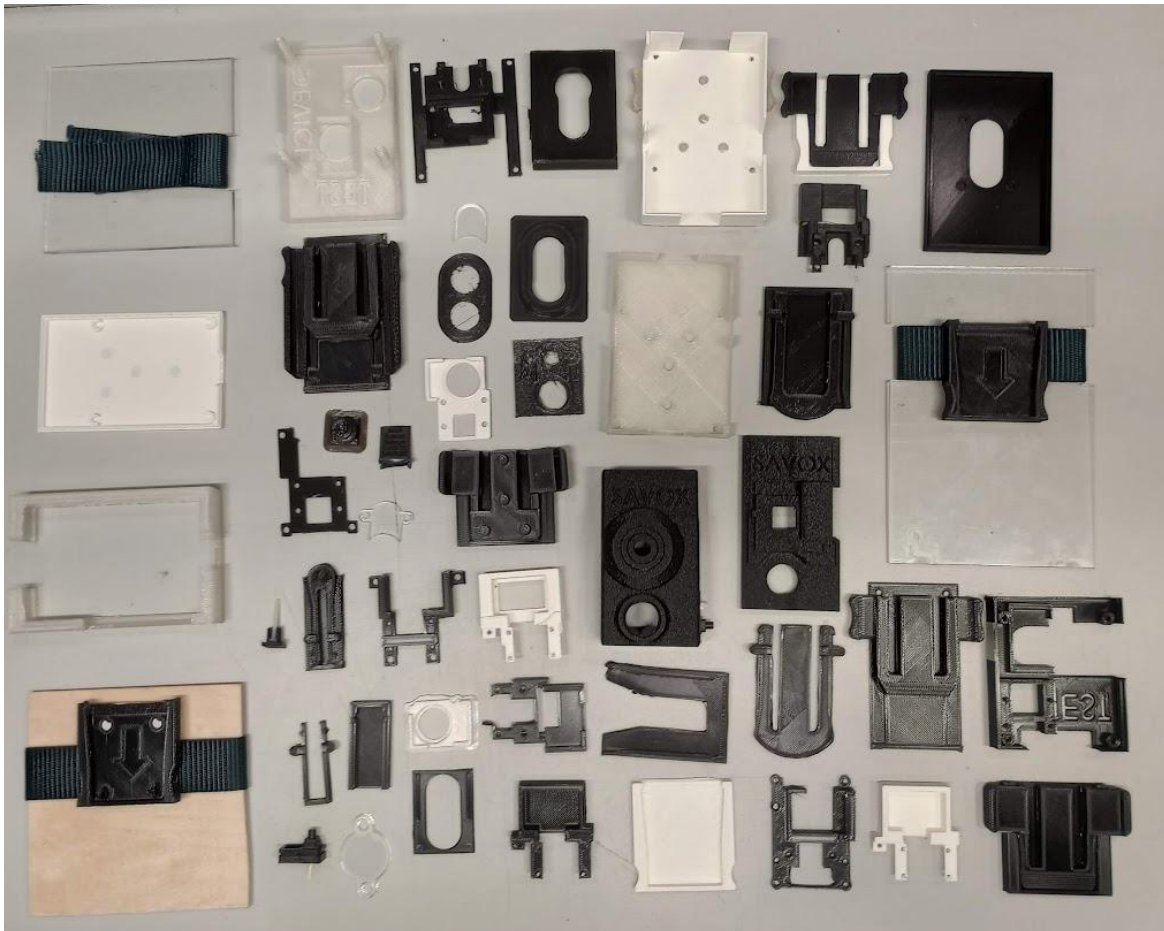


(Kuva 9. KiCAD näkymä piirilevyn footprint osiosta)



(Kuva 10. KiCAD näkymä piirilevyn skematiikka osiosta)

## 2. Tulostetut ja leikatut mallit



(Kuva 11. Mallinnuksen iteraatioita: kipua, tuskaa ja epätoivoa)

## 3. Laitteen ohjelmiston asennus

### 1. CM4 alkuasennus IO Boardin kanssa

- Laita IO Boardissa johto "disable EMMC boot" kohdalle
- Yhdistä MicroUSB ja aja rpiboot ohjelma tietokoneella
- Asenna EMMC muistiin RaspOS Lite (64 bit version)
- Yhdistä HDMI + näppäimistö (muista irroittaa Micro-USB ennen tätä)
- Päivitä kaikki ohjelmat
  - `sudo apt-get update -y && sudo apt-get upgrade -y`
- Asenna NetworkManager (nmcli)
  - `sudo apt-get install network-manager`
  - `systemctl start NetworkManager.service`
  - `systemctl start NetworkManager.service`
  - Yhdistä lähiverkkoon "nmcli dev connect wifi ..."
- Generoi SSH avaimet "ssh-keygen"
- Muokkaa sshd\_config tiedostoa ja asenna haluamasi yhteys asetukset. Tärkeimmät ovat:
  - `PublicKey Authentication yes`

- Password no
  - Gateway yes
  - Kopioi SSH avain palvelimelle sh\_copy\_id (anna hetkellisesti serverillä lupa päästä sisään salasanan avulla, tai kopioi avaimet jollakin tietoturvalisemmalla tavalla)
  - Luo käyttäjä serverillä (nimi esimerkiksi "rpi")
  - Kopioi "open\_ssh.bash" skripti esimerkiksi scp-työkalulla laitteelle. (skripti mukana laitteen koodi gitlabissa)
  - Muokkaa /boot/config.txt tiedostoon oikeat dtoverlay ja muut haluamasi asetukset. Nämä asetukset riippuvat siitä mitä kameraa tai laitetta käyttää. Tärkeimmät
    - dtoverlay=imx708
    - **i2c swap** (ainoastaan, kun CM4 on kiinnitettynä meidän piirilevyyn)
  - Asenna crontab ohjelma, ja muokkaa sen konfiguraatio haluamallasi tavalla
    - Aja "sudo crontab -e"
    - Muokkaa niin että skriptit ajautuvat minuutin välein ja rebootin aikana (ohjeet netistä, tai laite koodi repossa "crontab\_example.txt")
  - Kopioi serverin julkinen SSH avain (public key) laitteelle
  - Irroita HDMI ja näppäimistö
2. Ohjelmien asennus laitteelle
- Yhdistä palvelimelta laitteelle SSH yhteyden kautta (skripti myös tätä varten palvelin repossa)
  - Asenna tarvittavat kirjastot
    - sudo apt-get install git python3 python3-websockets python3-opencv python3-picamera2 -y
  - Lataa ohjelmakoodit laitteelle
    - git clone <https://version.aalto.fi/gitlab/toikkah2/savox-device-client>
  - Avaa palvelinohjelma ja testaa, että ohjelmakoodi toimii.
  - Asenna open\_cameras.bash tiedosto samalla tavalla kuin open\_ssh.bash aiemmin crontabiin
3. Testaaminen
- Reboottaa laite
  - Avaa palvelin ohjelma
  - Laitteen pitäisi alkaa lähettämään dataa, jos ei, niin avaa ohjelma manuaalisesti ja yritä selvittää

## 4. Käytetyt ohjelmistot

SOLIDWORKS (3D-mallinnus)

KiCAD (Piirisuunnittelu)

VSCoDe (Ohjelmointi)

GIMP (Juliste)

5. Juliste

**OPERAY Pro**  
*One device, two worlds*

SAVOX

Record to cloud

Wireless

Dual Camera

AI  
Human detection

Web interface

*"Enhance your operations with an integrated dual camera system"*

Made by **Henrik Toikka & Iiro Karppanen** at Protocamp 2023

**A!**

(Kuva 12. Juliste)



## 6. Linkit

Laitekoodi (Device software)

- <https://version.aalto.fi/gitlab/toikkah2/savox-device-client>

PCB mallinnukset ja skematiikka (PCB models and schematics)

- <https://version.aalto.fi/gitlab/karppai1/savox-pcb>

Palvelin ja käyttäjäpuoli (server and client)

- <https://version.aalto.fi/gitlab/toikkah2/savox-server>

Mallinnukset (Models)

- <https://version.aalto.fi/gitlab/toikkah2/savox-models>

## Lisenssi

Tämä projekti käyttää MIT lisenssiä.

## Lähteet

Digikey: Creating a Raspberry Pi Compute Module 4 (CM4) Carrier Board in KiCad  
<https://www.digikey.fi/fi/maker/projects/creating-a-raspberry-pi-compute-module-4-cm4-carrier-board-in-kicad/7812da347e5e409aa28d59ea2aaea490>

Raspberry Pi .2019. Compute Module 4 datasheet  
<https://datasheets.raspberrypi.com/cm4/cm4-datasheet.pdf>

Raspberry Pi .2019. Compute Module 4 IO Board datasheet  
<https://datasheets.raspberrypi.com/cm4io/cm4io-datasheet.pdf>

Hakolahti Esa: LÄMPÖKAMERAN KEHITYS JA KÄYTTÖ. Opinnäytetyö, ohjelmistotuotanto, Turun Ammattikorkeakoulu 2011  
[https://www.theseus.fi/bitstream/handle/10024/36505/Lampokameran\\_kehitys\\_ja\\_kaytto.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/36505/Lampokameran_kehitys_ja_kaytto.pdf?sequence=1)

GPS Logger sovellus:  
<https://f-droid.org/packages/com.mendhak.gpslogger/>

MapBox kirjaston dokumentaatio:  
<https://docs.mapbox.com/mapbox-gl-js/api/>