



Vadim Zubchenko

# Työympäristön analysointiohjelma

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tutkinto-ohjelman nimi

Insinööriyö

28.1.2022

# Tiivistelmä

Tekijä: Vadim Zubchenko  
Otsikko: Työympäristön analysointiohjelma  
Sivumäärä: 57 sivua  
Aika: 10.1.2022

Tutkinto: Insinööri (AMK)  
Tutkinto-ohjelma: Tieto- ja viestintäteknikka  
Ammatillinen pääaine: Ohjelmistotuotanto  
Ohjaajat: Lehtori Jorma Rätty  
Founder / CEO Kim Janson

---

Insinöörityön aiheena oli sulautettu IoT-järjestelmä, joka tukee ilmanvaihdon järjestelmää säätämällä sen tehoa riippuen ihmisten määrästä. Työn tavoitteena oli kehittää ohjelmisto, joka seuraa ihmisten liikkumista ja kulkusuuntaa ja samalla laskee tilaan saapuvien ja sieltä poistuvien ihmisten lukumäärä ja mittaa tilan hiilidioksidipitoisuutta. Toisena tarkoituksena oli kehittää ohjelma, joka siirtää kerätyt tiedot ilmanvaihdon ohjausjärjestelmään langattoman verkon kautta.

Insinöörityössä tutkittiin lämpösensorin käytön kelpoisuutta ihmisten tunnistamiseen ja laskemiseen sulautetussa järjestelmässä. Kehitettiin ohjelma, jolla saatiin data kamerasta ulos analysoimista varten, sekä käytettiin koneoppimis- ja konenäkötekniikkaa datan analysointiin.

Työn suorittamiseen käytettiin LeViteZer Oy:n LVZ-R1-laitetta, joka sisältää muun muassa ohjelman kehittämiseen tarvittavan infrapuna- ja hiilidioksiditunnistimet. Projektiin kuului sekä tietopohjan tutkiminen että LVZ-R1-laitteeseen ja siihen rakennettujen laitteistoiden tutustuminen. Ohjelman kehittämisen jälkeen suoritettiin kokeita eri tilanteiden simuloimiseksi ja saatuja tuloksia analysoitiin. Projektin tulokseksi kehitettiin kaksi ohjelmaa, joista toinen kerää infrapunasensorilta dataa ja esittää sen värikuvana verkkosivulla ja toinen analysoi infrapunakameran kuvan ja laskee ihmisten määrän. Kehitetyt ohjelmat ja pidetyt kokeet havainnollistivat tietokonenäön käyttöä lämpökameratietojen analysointiin.

Insinöörityön aikana luotiin malli sulautetusta järjestelmästä, jota voidaan käyttää ilmanvaihtojärjestelmän kanssa yhdessä säätelemään ilmanlaatua riippuen ihmisten määrästä ja hiilidioksidipitoisuudesta tiloissa. Kun LVZ-R1 laite sisältää infrapuna-, ilmalämpö-, kosteus-, ilmanpaine ja äänisensorit, tässä työssä esitetyt tulokset, perustelut ja johtopäätökset, ovat ne hyvä pohja työympäristön kartoituksen ja ylläpitämisen projektin jatkokehittämiseen.

Avainsanat: lämpösensori, lämpökamera, ihmisten laskeminen, sulautettu järjestelmä, IoT-järjestelmä.

## Abstract

Author: Vadim Zubchenko  
Title: Work Environment Analysis Program  
Number of Pages: 57 pages  
Date: 10 January 2021

Degree: Bachelor of Engineering  
Degree Programme: Information and communications technology  
Professional Major: Software engineering  
Supervisors: Senior Lecturer Jorma Rätty  
Founder / CEO Kim Janson

---

The topic of the graduate study was an embedded IoT system, which could improve a ventilation system by adjusting proper ventilation rate depending on the number of people in the room. The aim of the study was to develop software that tracks the movement of people, the direction of their movement, counts the number of people and measures the concentration of carbon dioxide in the room. The second aim was to develop a program which transfers the gained data about the number of people to the ventilation control system via the wireless network.

In the study the suitability of using a thermal sensor for human identification and computing tasks in an embedded system were investigated. Programs extracting data from the camera and analyzing it using machine learning and machine vision methods were developed.

As an essential tool the LVZ-R1 device from the company LeViteZer was used. The device includes, among other sensors, an infrared, and a carbon dioxide sensor. The goals of the project were achieved by means of a thorough study of the theoretical part and the hardware part of the LVZ-R1 device. The project included experiments where various situations with thermal camera and people were simulated and where the results obtained were analyzed. The outcome was two programs, one of which receives the data from an infrared sensor and presents it as a color image on a web page, and the other one analyzes the image from an infrared camera and counts the number of people. The developed programs and conducted experiments illustrate the possibilities of using computer vision to analyze data from a thermal camera.

A model of an embedded system which can be used with a ventilation system to regulate air quality, depending on the number of people and the concentration of carbon dioxide in the room was created. The presented results and the 5 different sensors of the device provide a good basis for the further development of a project to maintain good conditions in a working environment.

Keywords: heat sensor, thermal camera, human recognition, embedded system, IoT system.

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Tietopohja	2
2.1	Ilman laatu	3
2.2	Hiilidioksidi ja henkilömäärä	4
2.3	Projektissa käytettyjen antureiden toimintaperiaate	5
2.3.1	Hiilidioksidipitoisuuden mittaaminen	5
2.3.2	Ihmismäärän tunnistaminen infrapunakameralla	8
2.4	Sulautettu järjestelmä.	12
2.4.1	Esineiden Internet (IoT) -konsepti sulautetussa järjestelmässä	14
2.4.2	Kehitysalustojen käyttö sulautetun järjestelmän rakentamiseen	16
2.4.3	Laitteiston tiedonsiirtoprotokollat	18
2.4.4	Ohjelmistotuki	20
3	Opinnäytetyön toteutus	21
3.1	Työn alkuvaihe	21
3.1.1	Laiteen asentaminen ja ohjelmointityökalujen valmistelu	21
3.1.2	Ensimmäinen testidata (äänianturi)	23
3.2	Infrapunakameran datan saaminen ja käsitteleminen	25
3.2.1	Web-käyttöliittymän luominen	31
3.2.2	Kuvan analysointi ja Ihmisten määrän laskeminen	33
3.2.3	Videon lukeminen VideoCapture-objektilla	35
3.2.4	Kohteen tunnistus	36
3.2.5	Kohteen seuraaminen	38
3.2.6	Ihmisten määrän laskeminen	39
4	Tulokset	40
4.1	Kokeiden olot, työkalut ja menetelmät	41
4.1.1	Lämpökameran virkistystaajuus	41

4.1.2	Historia- ja kynnysparametrien säätäminen	42
4.2	Kokeet kahteen eri korkeuteen asennetulla lämpökameralla	44
4.2.1	Kokeet 2.28 metrin korkeudella asennetulla lämpökameralla	44
4.2.2	Kokeet 2.58 metrin korkeudella asennetulla lämpökameralla	47
4.3	Tulosten analysointi ja käyttökelpoisuus	50
5	Yhteenveto	51
	Lähteet	54

## Lyhenteet ja käsitteet

- ADC: *Analog Digital Converter*. Muunnin, jonka avulla sisään tuleva analoginen signaali on mahdollista muuttaa digitaaliseen muotoon.
- API: *Application Program Interface*. Ohjelmointirajapinta on ohjelmistoliittymän tyyppi, jonka avulla eri ohjelmat voivat yhdistä ja vaihtaa tietoja keskenään, eli keskustella.
- DAC: *Digital Analog Converter*. Muunnin, jonka avulla on mahdollista muuttaa ulosmenevä digitaalinen signaali analogiseen muotoon.
- EEPROM: *Electrically Erasable Programmable Read-Only Memory*. Haihtumattoman muisti, jota käytetään asetustietojen, kalibrointivakioiden ja konfigurointiparametrien tallentamiseen mikroprosessorin tai mikrokontrollerin sisältävissä laitteissa.
- ESP-IDF: *Espressif IoT Development Framework*. Virallinen ESP32-kehitysalustoille suunniteltu kehitysympäristö.
- ESP32: Espressif Systemsin kehittämä mikrokontrolleri.
- FreeRTOS: *Real-time operating system*. Reaaliaikainen käyttöjärjestelmä sulautetuille laitteille, jonka avulla tietojen ja tapahtumien käsittely ohjelmassa määriteltä aikarajoituksilla.
- GPIO: *General Purpose Input Output*. Yleiskäyttöinen pinni mikrokontrollereissa tai mikroprosessoreissa, jonka avulla yhdistetään oheislaitteita kehitysalustaan. GPIO-pinnit on ryhmiteltä portteihin.
- I2C: *Inter IC or IIC*. Kaksisuuntainen tiedonsiirtoväylä laitteiden liittämiseen prosessoreihin ja mikrokontrollereihin kaksijohtimisen liitännän kautta.
- IO-port: *Input/Output Port* Tietokonelaitteen syöttö- tai tulostusportti, jonka pystyy lähettämään (lähtö) ja vastaanottamaan (syöttö) tietoja.
- IoT: *Internet of Things*. Esineiden internet. Internet-verkkoon kytketyt laitteet, joita voidaan ohjata tai valvoa verkkoyhteyden kautta.
- KNN: *K-Nearest Neighbors*. K Lähin naapuri -koneoppimisen perusalgoritmi, joka automaattisesti luokittelee syötteitä ja joka käytetään taustamallin luomiseen.
- Lua-RTOS: Lua-ohjelmointikieleen perustuva reaaliaikainen käyttöjärjestelmä, joka on suunniteltu toimimaan sulautetuissa järjestelmissä.

- LVZ-R1: *LeViteZer*. LeViteZer Oy-yrityksen kehittämä laite, joka sisältää erilaisia tunnistimia, sekä Ethernet-, WLAN- ja Bluetooth-verkkojen yhdistykseen työkalut.
- MOG: *Mixture-of-Gaussians*. Koneoppimisen algoritmi, joka käsittelee kuvan kehysten yhdistelmän ja käyttää kehysten staattisia taustapikseleitä taustamallin luomiseen.
- NDIR: *Nondispersive Infrared Sensor*. Hiilidioksidin mittaamiseen käytetyn anturin tyyppi, jonka mittausperiaate perustuu säteilyn kaasuabsorptioon tunnetulla aallonpituudella.
- NodeMCU: *Node and Micro-Controller Unit*. Avoimen lähdekoodin laiteohjelmisto, joka toimii Espressif Systemsin kehitysalustoilla ja käyttää Lua-ohjelmointikieltä.
- OpenCV: Avoimen lähdekoodin konenäkökirjasto konenäkö- ja koneoppimissovellusten kehitykseen.
- PoE: *Power over Ethernet*. Tekniikka, joka mahdollistaa tavallisten Ethernet-verkkokaapeleiden toimivan virtajohtona.
- SCL: *Serial Clock*. Sarjamuotoisen kellosignaaliaväylä tiedonsiirron synkronointiin oheislaitetta ja mikrokontrolleria kytkettäessä.
- SDA: *Serial Data*. Sarjamuotoisen tiedonsiirtoväylä, jonka kautta oheislaite yhdistetään mikrokontrolleriin.
- SPI: *Serial Peripheral Interface*. Tiedonsiirtoväylä laitteiden liittämiseen prosessoreihin ja mikrokontrollereihin nelijohtimisen liitännän kautta.
- Ta/To: *Ambient Temperature / Temperature of Object*. MLX90640-lämpöanturin ympäristön lämpötila / Laskettu kohteen lämpötila.
- UART: *Universal Asynchronous Receiver Transmitter*. Asynkronilinjasoitin, joka muuntaa rinnakkaismuotoista tietoa sarjamuotoiseksi ja päinvastoin oheislaiteen ja mikrokontrollerin väliseen kommunikointiin.
- VDD/VSS: *Voltage Drain Drain / Voltage Source Source*. Vdd- ja Vss-pinnit elektroniikkapiireissä viittaavat positiiviseen ja negatiiviseen jännitteeseen vastaavasti, joita käytetään virtalähteenä.
- Web API: *Web Application Programming Interface*. Rajapinta, johon voi lähettää tietoa ja hakea monelta laitteelta samanaikaisesti haku- vastausjärjestelmällä.

# 1 Johdanto

Hyvät työolot ja työympäristö ovat tärkeitä asioita työntekijän terveyden, hyvinvoinnin ja työsuorituksen kannalta. Toisaalta työturvallisuuslaki velvoittaa työnantajaa selvittämään, tunnistamaan ja ehkäisemään työympäristöstä aiheuttavat vaaratekijät ja tiedottaa niistä avoimesti ja oikea-aikaisesti (1).

Nykyaikaisissa toimistotiloissa, joiden käyttö ei ole jatkuvaa, ilmanvaihtoa käytetään joko jatkuvasti tai jaksottaisesti riittävän sisäilmaston tason varmistettavaksi (2). Insinööriyön ideana oli kehittää älykäs järjestelmä, joka reaaliajassa tulee tukemaan ilmanvaihtojärjestelmiä torjuntamaan sisäilmaongelmia hyvissä ajoin ja samalla energiataloudellisemmin.

Ilmanvaihdon mitoituksessa käytetään sisäilman hiilidioksidipitoisuutta perusmittauksena (3). Toiselta hiilidioksidin peruslähde työsisätiloissa on ihmisen hengitysilma (4). Nämä tekijät huomioon ottaen opinnäytetyön tavoitteena oli suunnitella sellainen järjestelmä, joka seuraa jatkuvasti samassa työtilassa oleskelevaa ihmismäärää ja hiilidioksidipitoisuutta, analysoi ilmanlaatua ja ohjaa ilmanvaihdon toimintaa todellisen tarpeen mukaan kyseisessä työtilassa.

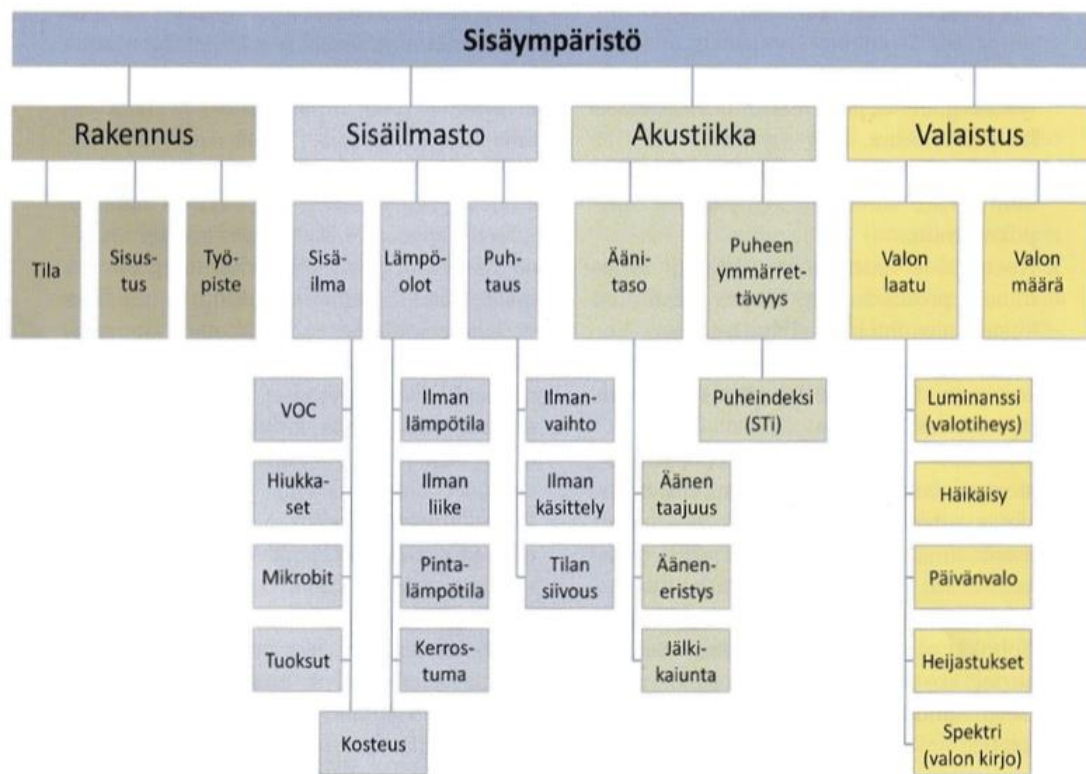
Isojen työtilojen nähden, kuten esimerkiksi luentosali tai tuotantohalli, tavoitteena oli keksiä tapa tunnistaa, missä työtilan kohdissa ja montako ihmistä on kokoontunut. Näin on mahdollista mitata tarkasti hiilidioksidipitoisuutta paikoissa, joissa ihmiset ovat tällä hetkellä.

Insinööriyön toteutettiin LeViteZer Oy yrityksen (5) yhteistyössä, joka tarjosi tutkimukseksi oma nimeltään LVZ-R1 laite. Sisältää kuusi eri sensoria LVZ-R1-laitteen avulla saatiin tarvittavat projektin kehityksessä ja analysoinnissa sisäilman mittaukset.



## 2 Tietopohja

Sisäilma on ilma, jota me hengitämme sisätiloissa ja joka paitsi ilman perusosien saattaa sisältää eri lähteistä peräisin kaasumaisia ja hiukkasmaisia epäpuhtauksia. Sisäilmassa olevat fyysiset, kemialliset ja mikrobiologiset tekijät, jotka tavalla tai toisella vaikuttavat ihmisen terveyteen kutsutaan sisäilmastoksi. Rakenteessa ihmisiin vaikuttavat myös muutkin tekijät, kuten akustiset olot, valaistus, tilasuunnittelu ja sisustus. Kaikki yllä mainitut käsitykset ja niiden tekijät kuvaavat sisäympäristöä. Ilmastoinnin kehityksessä tärkeimmät tekijät ovat ilman laatu, lämpöolosuhteet ja järjestelmien melu. (4; 6, s. 37.) Kuvassa 1 on havainnollistettu sisäympäristön laadun määrittelyyn vaikuttavat tekijät.



Kuva 1. Sisäympäristön laadun määrittelyyn vaikuttavat tekijät (6, s. 14).

## 2.1 Ilman laatu

Pystykö ihminen tunnistamaan puhdasta ilmaa aistiensa avulla? Monille on tuttu tilanne, kun sateen jälkeen, metsässä tai vaikka nurmikon ajamisen jälkeen aistitaan miellyttäviä tuoksuja ja viileyttä. Oman hengityksemme perusteella teemme helposti subjektiivisen johtopäätöksen, että ilma on raikasta ja puhdasta. Kuitenkin tieteellisen näkökulman nähden, jos ihminen haistaa ilmassa jotain hajua, se osoittaa epäpuhtauksien esiintymisen ilmassa.

Paitsi epäpuhtauksia ja hiukkasia, joita ihminen pystyy paljastamaan aisteilla, ilmassa on hajuttomia, mauttomia ja näkemättömiä aineita. Tieteelliseltä kannalta puhdas ilma on hajutonta, väritöntä ja mautonta kaasuseos ja sen kaasumaisien epäpuhtausien pitoisuuksien arvaamiseksi tarvitaan erikoistapoja ja laitetta.

Ilman laatu arvioinnissa otetaan huomioon lukuisten kemiallisten yhdisteiden, hiukkaisten ja mikrobien lisäksi on myös lämpötila ja kosteus. Kaikki nämä vaikuttajat tekevät sen määrittäminen mittaamalla monimutkaseksi tehtäväksi. Joidenkin epäpuhtauksien pitoisuus on mahdollista mitata suoraan paikan päällä laitella, mutta useimmiten tarvitaan laboratoriotutkimuksia. Tärkeä keino ihmisten terveyshaittojen vähentämisessä sisätilassa on ilmanvaihto. Ilmanvaihdolla saa vaikutettua epäpuhtauksien pitoisuuksiin hengitysilmassa. (6, s. 57.)

Hyvän sisäilma laadun varmistamiseksi on kehitetty erityyppisiä ilmastointijärjestelmiä, joiden keskinen yksikertainen tavoite on saada sopiva ilmanvaihto jokaista henkeä kohti oleskelutilassa. Ilmanvaihdon säädön toteuttamisessa tärkeä tekijä on henkilötiheys kyseisessä tilassa. Jos henkilömäärää ei ole mahdollista tunnistaa, käytetään vain lattiapinta-alan arvoa. (7, s. 38.)

Ilmastointijärjestelmät, jossa ilmanvaihtosäätö perustuu eri tunnistimien parametreihin, kutsutaan tarpeenmukaiseksi ilmanvaihtojärjestelmäksi. Sellaisia parametreja voivat olla lämpötila, hiilidioksidipitoisuus, kosteuspitoisuus ja henkilömäärä. Paitsi sisäilmaston parantamisesta, niiden tärkeänä etuna on alhainen energiakulutus, sillä ilmanvirran säätö tapahtuu reaaliaikaisen sisäilman parametrien ja henkilömäärän perustella.

Ilmanvaihdon tarpeellinen ilmavirta lasketaan erilaisten kuormitusten perustella. On olemassa epäpuhtaus-, lämpö- ja kosteuskuormitusten perustella tasapainotilanteen laskennan periaatteet. Tässä insinööriyössä käsiteltiin ihmisten tuottamien epäpuhtauskuormitusten pohjalla toimivaa ilmanvaihdon järjestelmää, jolle hyvänä indikaattorina on hiilidioksidi. Huonetilojen kuormitus vaihtelee erityisesti tiloissa, joissa on vaihteleva henkilökuormitus kuten luentosaleissa ja neuvottelutiloissa. (7, s. 64.) Hiilidioksidi ja henkilömäärä ovat hyviä työkaluja paitsi ilmavirran seurantaan myös sisäilman laadun arviointiin.

## 2.2 Hiilidioksidi ja henkilömäärä

Sisätilassa ihminen on tärkein hiilidioksidin lähde ja hiilidioksidipitoisuutta pidetään indikaattorina riittämättömästä ilmanvaihdosta (6, s. 76).

Koska ihmismäärän ja ihmishengityksillä tuottama hiilidioksidin välillä on suora yhteys, niitä kaksi parametria käytetään sisäilman laadun arviointiin ja ilmanvaihdon säätöön. Hiilidioksidipitoisuuden avulla lasketaan tarvittavan ilmavirran ja henkilömäärän perustella ennustetaan hiilidioksidipitoisuuden muuttumista ko. tilassa. (7, s. 21, s. 64–65.)

Hiilidioksidimittareilla ohjataan tilojen tarpeenmukaista ilmanvaihtoa ja havainnallisetään ilman laatua käyttäjille. Hiilidioksidipitoisuudesta ilmoitetaan ppm-yksikön avulla kun sen mittaamisen tulokseksi ilmoitetaan 500 ppm, tämä tarkoittaa, että 1 m<sup>3</sup> ilmaa sisältää 500 cm<sup>3</sup> hiilidioksidia. Taulukossa 1 on esitetty ilman laadun tavoitearvot huonetilan suunnitteluun hiilidioksidipitoisuuden nähden kolmen sisäilmastoluokkien suhteen. Kyseiset luokat kuvailevat ilman laadun eri tasoa. Laatuluokka S1 on yksilöllinen ilmalaatu, S2 kuvailee hyvää sisäilmastoa sekä kaksi muuta rakennus- ja terveydensuojelulain vähimmäisvaatimuksia vastaavasti. (8; 9, § 5; 10, § 8.)

Taulukko 1. Ilman laadun suunnittelu- ja tavoitearvoja.

	S1	S2	Ympäristöministeriön asetus	Sosiaali- ja terveysministeriön asetus
<b>Hiilidioksidipitoisuuslisä*[ppm]</b>	< 350	< 550	< 800	< 1150

\* suurempi kuin ulkoilman hiilidioksidin pitoisuuden oletusarvon 400 ppm

### 2.3 Projektissa käytettyjen antureiden toimintaperiaate

Anturit ovat esineiden Internet IoT (Internet of Things) -konseptin selkäranka. Ne yhdistävät reaali maailman ja virtuaali maailman. Anturit ovat laitteita, joilla voidaan mitata tiettyä luonnon ominaisuuksia. Yksinkertaistamiseksi otetaan lämpötila-anturi. Lämpötila-anturi käyttää yleensä termistoria. Termistori on elektroninen komponentti, jonka vastus muuttuu lämpötilan muuttuessa. Useimmissa termistoreissa vastus pienenee lämpötilan noustessa. Tämä vastus voidaan muuntaa jännitteeksi ja lukea ulkopuolisella mikrokontrollerilla, eli käyttää analogista mittaussuunnitelmaa. Näitä kutsutaan negatiivisten lämpötilakerrointen termistoreiksi NTC-termistori (Negative Temperature Coefficient). On olemassa myös anturit, joiden sisäänrakennettuna digitaalinen muunnin mahdollistaa tulokset suoraan digitaalisena. Edellinen selitys on esimerkki lämpötila-anturin toiminnasta. Projektissamme käytetyissä antureissa on erilaisia mekanismeja, joiden toimintoperiaatetta me käsittelemme tässä osassa. (26.)

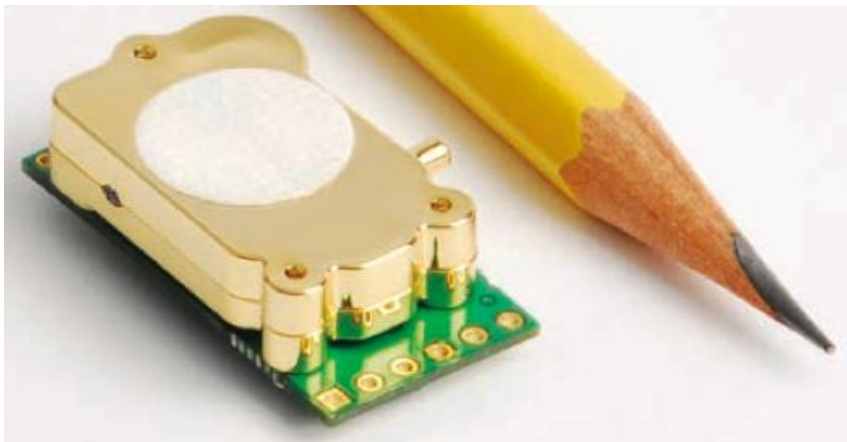
#### 2.3.1 Hiilidioksidipitoisuuden mittaaminen

Yleisemmin käytetyt hiilidioksidianturit jaetaan optisiksi infrapunakaasuantureiksi ja kemiallisiksi antureiksi niiden toimintaperiaatteiden mukaan. Lähes viime aikoihin asti lämpökemialliset CO<sub>2</sub>-anturit olivat houkuttelevimpina vaihtoehtoina sisäilmanvaihtojärjestelmän suunnittelussa niiden sähkönkulutuksen vaatimattomuuden ja pienen koon vuoksi. Kun taas optiset anturit ovat vähemmän alttiita ulkoisille

vaativille olosuhteille, ne ovat selektiivisimpiä mitattavalle kaasulle ja niillä on pidempi käyttöikä (11).

Taitettu optiikkaa ja heijastettu valoa kaarevan muodon metalloitun muovatun muovin, sekä infrapunavaloa lähettävien LED-valonlähteiden avulla mahdollistivat tehdä kovan kilpailijan kemialliselle hiilidioksidianturille.

Nykypäivän teollisuus tarjoaa monia erilaisia pienikokoisia kuluttavia, vain 3 mW:n tehoa optisia hiilidioksidianturia, jotka helposti integroivat sovelluksiin, joissa CO<sub>2</sub>-tasot on mitattava ja säädettävä.



Kuva 2. Telair T6713 CO<sub>2</sub> -anturi (12).

Kuvassa 2 on havainnollistettu käytetty projektissa hiilidioksidianturi Telaide T6713.

Projektissamme käytetty Telaide T6713 -anturi kuuluu optiseen infrapunasäteilyä käyttävien CO<sub>2</sub>-antureiden ryhmään. Tämän tyyppin antureilla on teollistermi NDIR-anturi (Nondispersive Infrared Sensor) eli hajautumaton infrapuna-anturi. NDIR-anturi rakennuksessaan on yksinkertainen spektroskooppinen anturi, jota käytetään usein kaasunilmaisimena. NDIR-nimensä se sai siitä, että kaasujen mitattaessa ei käytetä infrapunasädettä hajauttava laitetta, kuten prisma tai diffraktiohila spektrometreissä.

Infrapuna-anturin toimintaperiaatteen pohjalla on tärkeä hiilidioksidikaasun ominaisuus absorboida infrapunasäteilyä, eli kun sen molekyyliin kohdistuu infrapunasäteilyä, osa säteilystä ohittaa molekyylin ja osa imeytyy. Toisin sanoen mitä enemmän

hiilidioksidimolekyylejä on, sitä enemmän ne absorboivat infrapunasäteilyä ja sen intensiteetti pienenee, eli se on suoraan verrannollinen CO<sub>2</sub>-molekyylien määrään. Kun hiilidioksidi absorboi osan lämpösäteilystä lyhyillä infrapunaa aallonpituuksilla, NDIR-anturissa olevat hiilidioksidimolekyylit absorboivat vain tietyn aallonpituuden infrapunälähteestä, noin 4,3  $\mu\text{m}$ -pituudella. Tällaista ainulaatuista infrapunaspektriä erotetaan ja suodatetaan muista kaasujen aallonpituudesta optisen suodattimen avulla.

Saadun absorboitunut punasäteilyn intensiteetin ja alkuperäisen, ei CO<sub>2</sub>-absorpioitu punasäteilyn intensiteetin suhteen avulla, saadaan kuvan hiilidioksidin konsentraatiosta. (13.)

Käyttäen Beerin ja Lambertin lakia, jonka mukaan sähkömagneettisen säteilyn absorboituminen riippuu absorboivan aineen määrästä eksponentiaalisesti voidaan laskea vastaanotetun säteilyn voimakkuuden. (14.) Lambertin-Beerin lain yhtälöstä (kaava 1) nähdään, että vastaanotettu absorboituvalon voimakkuus on verrannollinen kammion sisällä olevien CO<sub>2</sub>-molekyylien määrään (15, s. 11241).

$$I = I_0 \times e^{-k \times l \times [CO_2]} \quad (1)$$

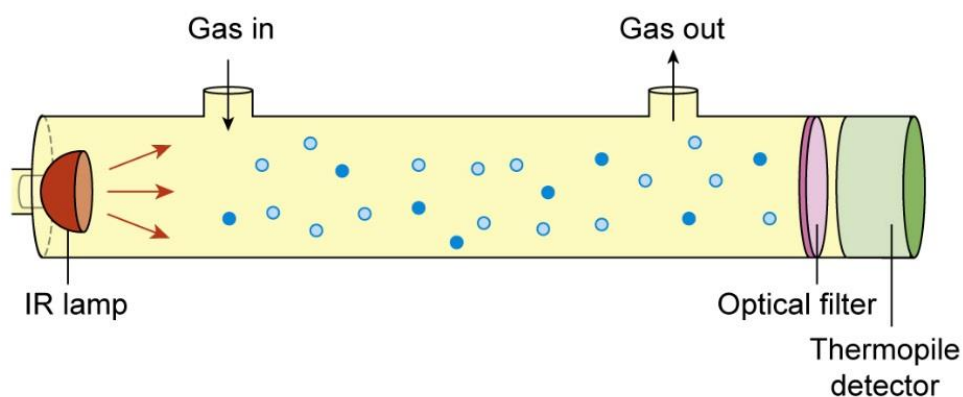
$I$  on säteilyn intensiteetti CO<sub>2</sub> absorboimisen jälkeen, ( $W \cdot m^{-2}$ )

$I_0$  on kammioon tulleen säteilyn intensiteetti, ( $W \cdot m^{-2}$ )

$k$  on hiilidioksidin absorptiokerroin 4,3  $\mu\text{m}$ -aallonpituudella, ( $cm^{-1}$ )

$l$  on absorptiopulun pituus, ( $cm$ )

$[CO_2]$  on mitattava hiilidioksidin konsentraatio, ( $mol \cdot dm^{-3}$ )



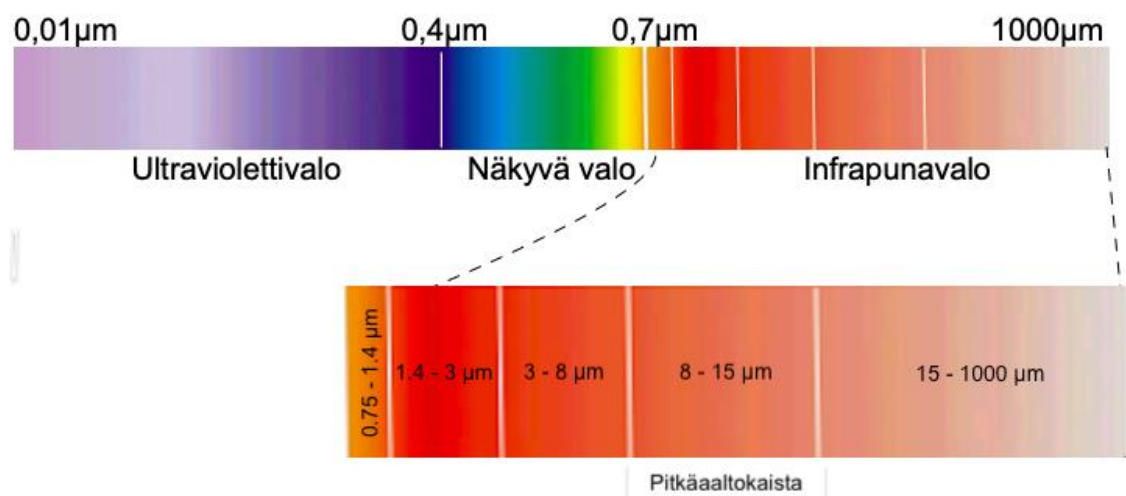
Kuva 3. Luonnos NDIR-anturin rakenteesta. (15, s. 11242).

Kuvassa 3 on esitelty Infrapuna-hiilidioksidianturin pääkomponentit. Infrapunalamppu (IR Lamp) lähettää infrapunasäteilyä ilmaa sisältävää näytekammiota pitkin, sitten optisen suodatin (Optical Filter) absorboi kaikki aallonpituudet paitsi hiilidioksidille ominaiset absorptioaallonpituudet, eli  $4,3 \mu\text{m}$  ja infrapunailmasin (Thermopile Detector) mittaa saapuvan säteilyn intensiteettiä.

Aluksi suunniteltu hiilidioksidianturin hyödyntäminen ohjelmassa rajattiin tästä työstä pois. Näin saatiin työn laajuutta pienennettyä huomattavasti. Hiilidioksidianturin käyttö voidaan käsitellä projektin jatkokehitystä varten.

### 2.3.2 Ihmismäärän tunnistaminen infrapunakameralla

Jokainen kappale, jonka lämpötila on yli absoluuttisen nollapisteen  $-273 \text{ }^\circ\text{C}$ , lähettää lämpösäteilyä eli infrapunasäteilyä. Tämä ominaisuus on taustalla jokaisen lämpökameran päätoimintaperiaatteessa.



Kuva 4. Infrapunasäteily sähkömagneettisessa spektrissä.

Kuva 4 havainnollistaa Infrapunasäteilyä sähkömagneettisessa spektrissä. Infrapunasäteily ulottuu näkyvän valon spektrin punaisesta reunasta  $0,7-1000 \mu\text{m}$ .

Infrapunakaista jaetaan myös viiteen kapeampaan kaistaan, kuten lähi-infrapunakaista 0,75—1,4  $\mu\text{m}$ , lyhyt-infrapunakaista 1,4—3  $\mu\text{m}$ , keski-infrapunakaista 3—8  $\mu\text{m}$ , pitkäaaltokaista 8—15  $\mu\text{m}$  ja ääri-infrapunakaista 15—1000  $\mu\text{m}$  (17).

Stefan-Boltzmannin lain mukaan mitä kuumemmaksi esine tulee, sitä enemmän se lähettää infrapunaenergiaa, eli infrapunasaateilyn voimakkuus riippuu kappaleen todellisesta lämpötilasta. Sitä otetaan hyväkseen saadakseen mitattua erittäin tarkat kosketuksettomat lämpötilamittaukset, liikkeen tunnistamiseen, läsnäolon havaitsemiseen tai lämpötila-anturiksi älykäsilmastoinnissa. Vaikka infrapunasaateily ei näy ihmissilmälle, voidaan visualisoida se näkyväksi mittauseriaatteita käsiteltäessä ja sovelluksia kehittäessä, koska se käyttäytyy samalla tavalla kuin näkyvä valo.

Kappale voi heijastaa, absorboitua tai siirtyä läpi muista lähteistä sisääntulevaa säteilyä. Osa absorboimasta energiasta päästetään sisäisesti ja osa heijastuu takaisin. Tämä ilmiö koskee myös silmälle läpinäkyviä materiaaleja, kuten lasia, kaasuja ja kirkkaita muoveja. Sen lisäksi osa IR-energiasta kulkee myös esineen läpi. (18). Kappaleen pintamateriaalirakenne ja sen ympäröivä taustasaateily, eli taustan lämpötila vaikuttavat yhdessä esineen tai materiaalin emissiivisyyteen (19).

Emissiivisyys kuvaa materiaalin kykyä lähettää omaa lämpösäteilyään verrattuna täydellisen mustan kappaleen säteilyyn. Se ilmoitetaan arvolla nolasta ykköseen. Teoreettisesti, materiaali, joka ei heijasta tai välitä IR-energiaa ja absorboi kaiken itseensä kohdistuvan säteilyn millä tahansa aallonpituudella kutsutaan täydelliseksi mustaksi kappaleeksi. Sen avulla emissiivisyys voidaan esittää matemaattisesti kaavalla 2 (20, s. 111).

$$\varepsilon_{\lambda} = \frac{W_{\lambda o}}{W_{\lambda b}} \quad (2)$$

$\varepsilon_{\lambda}$  on kappaleen emissiivisyys aallonpituudella  $\lambda$   
 $W_{\lambda o}$  on kappaleen säteilyteho aallonpituudella  $\lambda$   
 $W_{\lambda b}$  on mustan kappaleen säteilyteho aallonpituudella  $\lambda$

Täydellisen mustan kappaleen emissiivisyys on 1. Emissiivisyyden pieni arvo tarkoittaa, että suurin osa lämpösäteilyä heijastuu materiaalin pinnasta tai kulkee sen läpi. Materiaalit, joilla on suurempi emissiokyky, näyttävät olevan lähempänä todellista



lämpötilaa kuin materiaalit, jotka heijastavat enemmän eri lämpötiloja ympäristöönsä. Monien materiaalien emissiivisyyden arvoja tai emissiokertoimia eri aallonpituudella ja lämpötiloissa löytyy emissiivisyystaulukoista. Esimerkiksi mitattuna koko infrapunaspektrillä alumiinin karkean pinnan emissiokerroin on 0,06–0,07 lämpötiloissa 20–50 °C ja ihmisen ihon on 0,98 lämpötilassa 32 °C (21).

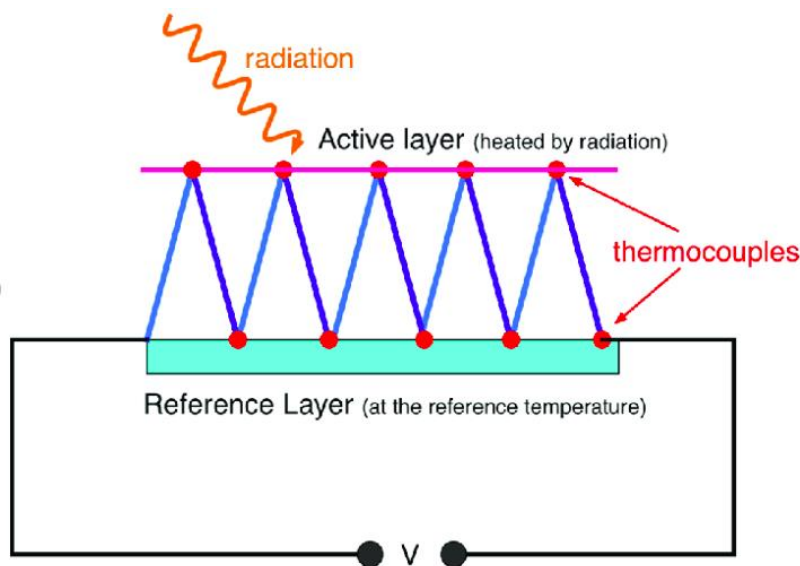
Koska lämpökamera ei tunnista materiaalin emissiivisyyttä automaattisesti, se on otettava huomioon, jotta kohteen lämpötila voidaan mitata oikein. Tämä parametri on syötettävä lämpökameran ohjelmistoon heijastavan vaikutuksen korjaamiseksi. Toiseksi, vaikka melkein aina on mahdollista määrittää mitattavan perusmateriaalin emissiivisyys, hankaluksia syntyy materiaaleista, joiden emissiivisyys muuttuu lämpötilan mukaan, kuten useimmat metallit. Tästä johtuen mitattavan materiaalin taustan lämpötilakin on tärkein IR-kameran parametri. Useimmiten sitä toteutetaan mittaamalla ympäristön lämpötilaa sisäänrakennetulla anturilla.

Riippuen tarkoitettuuksistaan IR-kameran toiminnossa hyödynnetään eri infrapunakaistoja. Esim. lähi-infrapunakaistaa 0,75—1,4 µm ei voida käyttää lämpökuvaukseen, sillä se ei havaitse kappaleen omaa lähettämää lämpösäteilyä alle n. 650°C. Lyhyt-infrapunan aallonpituuksilla 1,4—3 µm suurin osa lämpösäteilystä absorboi ilmassa olevalla vesihöyryllä ja hiilidioksidilla. Ääri-infrapunakaistaa infrapunakaista 15—1000 µm ei käytetä, sillä teollisuudella tällä hetkellä saatavilla olevat IR-kamerat eivät ole riittävän herkkiä havaitsemaan hyvin pieniä yli 15 µm aallonpituuksilla energiamääriä. Yleisemmin lämpökamerat toimivat tietyn spektrin infrapunakaistanleveyden sisällä, kuten keski-infrapunakaista ja pitkäaaltokaista. Keski-infrapunakaistaa 3–5 µm hyödynnetään silloin, kun ensisijaisena tavoitteena on saada korkealaatuisia kuvia sen sijaan, että keskityttäisiin lämpötilan mittaamiseen ja liikkuvuuteen. Pitkäaaltokaistalla 8–14 µm on optimaalinen vaihtoehto, kun kyseessä on hieman korkeampi kuin huoneen lämpötila kohteita, kuten ihmiskeho, sillä huoneen lämpötilassa olevat esineet säteilevät pääasiassa 8–25 µm infrapunakaistanleveyden sisällä (17). Materiaalien emissiokertoimia valitaan emissiivisyystaulukoista riippuen, mitä infrapunakaistaa käytetään lämpökamerassa.

Infrapunakamerat tai toisin kutsumanimellä infrapunasensorit ovat laitteet, jotka vastaanottavat lämpösäteilyä aallonpituudella 7,5–14 µm. Lämpösäteily iskee siihen

rakennettuun ilmaisimen materiaaliin, lämmittää sitä ja muuttaa sen sähköistä vastusta. Tämä vastusmuutos mitataan ja käsitellään lämpötiloiksi, joita voidaan käyttää kuvan luomiseen ja koko alueen lämpötilan mittaamiseen kosketuksettomaksi.

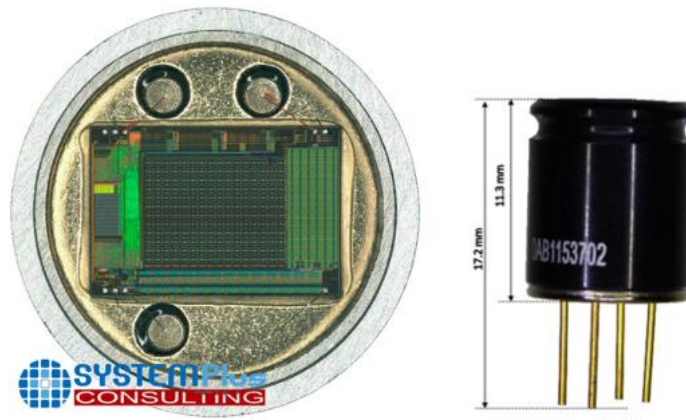
Kohteen lähettämä säteily kohdistuu sopivan lämpöanturin vastaanottavaan elementtiin. Tämä voi olla vastuselementti, joka on yleensä termopari tai lämpöpari. Lämpösensorin infrapunailmaisimen rakennukseltaan on termoparin sarja, jonne infrapunasäteily keskitetään piilinsseillä. Termoelementin sarjakytkentää kutustaan termopatsaaksi.



Kuva 5. Termopatsas-infrapunailmaisimen toimintaperiaatteen kaavio (22, s. 91).

Kuvassa 5 on havainnollistettu termopatsas-infrapunailmaisimen toimintaperiaatteen kaavio. Säteilytehon mittaamiseen käytetään sarjaa termoelementtejä, jotka on kytketty vuorottelen aktiiviselle puolelle ja referenssipuolelle. Aktiivinen puoli koostuu tyypillisesti hyvin ohuesta materiaalikerroksesta. Tämän sarjan lähtöjännite on verrannollinen aktiivisen- ja referenssikerrosten väliseen lämpötilaeroon. Absorboitu aktiiviseen kerrokseen kokonaissäteily, jossa lämpötilan nousu vaikuttaa absorboituun säteilytehon kasvamiseen, havaitaan ja muunnetaan jännitteeksi. (22, s. 91.)

Infrapunasensori rakennukseltaan on matriisi, jonka jokainen elementti on säteilyn vastaanottava termoparin sarja. Tämä mahdollistaa pienikokoisten lämpöanturien valmistuksen.



Kuva 6. MLX 90640 lämpöanturi ja sen matriisi, jossa on 768 termoelementtejä (23).

Kuvassa 6 on esitetty projektissamme käytetty infrapuna-anturi FIR-sensori (Far Infrared Sensor) nimeltään Melexis Sensor MLX90640, jonka matriisin koko on 32x24 pikseliä ja kamerakoko vain  $1 \text{ cm}^3$ . (23.)

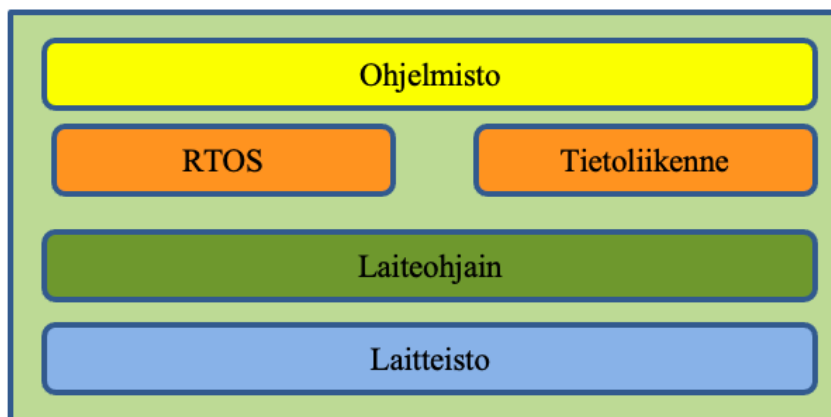
## 2.4 Sulautettu järjestelmä.

Henkilökohtaisella tietokoneella voi tehdä monenlaisia tietojenkäsittelytehtäviä, joten arkikielessä sitä kutsutaankin yleiskäyttöiseksi tietokoneeksi. Sen valmistaja ei tiedä, käytetäänkö sitä verkkotiedostopalvelimena, peleissä vai kirjoituskoneena. Koska yleiskäyttöistä tietokonetta yleisimmin käytetään vain rajattuihin tehtäviin, sen tehokkuuden, energiakulutuksen, koon ja painon ominaisuuksia tuskin voi pitää optimaalisina omaan tehtävään erikoistuneeseen sulautettuun järjestelmään.

Sulautettu järjestelmä on tietokoneen laitteiston, ohjelmiston ja tarvittavien mekaanisten tai elektronisten komponenttien yhdistelmä, joka on suunniteltu suorittamaan tietty toiminto. Useimmiten se on mikroprosessoripohjainen ohjaus-, hallinta- ja valvontajärjestelmä, joka on rakennettu suoraan ohjaamaansa laitteeseen. Tältä osin tämä asettaa tiettyjä vaatimuksia sulautetulle järjestelmälle, kuten minimaaliset virrankulutus, lämmitys, mitat, paino ja mahdollinen itsenäinen virtalähde. Noiden vaatimuksien toteuttamiseen kunkin sulautetun järjestelmän laitteisto on räätälöity nimenomaan tehtävänsä suorittavaan sovellukseen. Sen seurauksena tarpeettomat piirit

poistetaan ja laitteistoresurssit jaetaan aina kun mahdollista. Se mahdollistaa edullisemman järjestelmän rakentamisen, jolla on pienempi virrankulutus, vähemmän lämpövapautumista, pienempiä akkuja, vähemmän painoa, pienempää kokoa ja yksinkertaisempaa mekaanista rakennetta.

Sulautetut järjestelmät kehitetään niin, että ne toimivat usein ilman ihmisen välitöntä ohjausta eikä käyttäjälle tarvitse olla tietoinen sen komponentista ja toimintaperiaatteesta. Sulautettu järjestelmä voi olla kodinkoneita, lentokoneen lennonohjaus- tai sisäilmastointijärjestelmä sekä muut laitteet, joiden toiminnon ytimessä on prosessorin ja ohjelmiston yhdistelmä. Sen rakenne voi kuvata erillisinä lohkoina.



Kuva 7. Sulautetun järjestelmän ohjelmiston arkkitehtuuri (24, s. 6).

Kuvassa 7 on esitetty sulautetun järjestelmän ohjelmiston korkean tason kaavio. Vaikka sen erilliset lohkot ovat pinottu päällekkäin ja niillä on kiinteät reunat, ohjelmallisesti naapurilohkot on yhdistetty ja ne kykenevät keskustelemaan toistensa kanssa.

Ohjelmiston pinomaisien kerroksien pitäminen erillisinä ja hyvin määritellyn menetillä niiden välin kommunikointiin auttavat kirjoittamaan sulautetun järjestelmän hyvälaatuista ohjelmistoa. (24, s. 5.)

Laitteisto

Paitsi prosessoria ja ohjelmistoa, sulautettu järjestelmä sisältää ROM-lukumuistia (Read Only Memory), johon voidaan tallentaa suoritettava koodi ja väliaikainen keskusmuisti RAM (Random Access Memory) ajonaikaisten tietojen käsittelyä varten sekä tiedon lähetys- ja vastaanottoportit. Muistit voivat olla joko erillisinä mikrosiruina tai rakennettu samaan mikrosiruun prosessorin ja IO-porttien (Input/Output Port) kanssa. Toisessa tapauksessa kyseessä on mikrokontrolleri. (24, s. 5, s. 6.)

### Laiteohjain

Laiteajurit ovat sulautettuja ohjelmistomoduuleja, jotka sisältävät toimintoja yksittäisten laitteiden käyttämiseen. Riippuen tehtävästään, sulautettuun järjestelmään saattaa liittyä myös eri tunnistimia, kameraa, näppäimet, joilla on omat laiteohjaimet. Laiteajurien avulla ohjelmiston kehittäjän ei tarvitse tuntea miten laitteiston kanssa tapahtuu keskustelu. Tämä tekee tarpeettomaksi jokaisen laitteiston ohjaamisen varsinaisella koodilla ja helpottaa koodin ymmärtämiseen. (24, s. 5, s. 6.)

### RTOS (Real-Time Operating System)

Sovellusohjelmiston paremman organisointiin ja reagoivamman järjestelmän luomiseksi käytetään reaaliaikaista käyttöjärjestelmää, eli RTOS:aa. RTOS mahdollistaa järjestää jokaisen koodin suoritetta tehtävä niin että sen alka- ja lopputila ajaa tietyissä ajankohdissa ja vaiheissa. (24, s. 5, s. 6.)

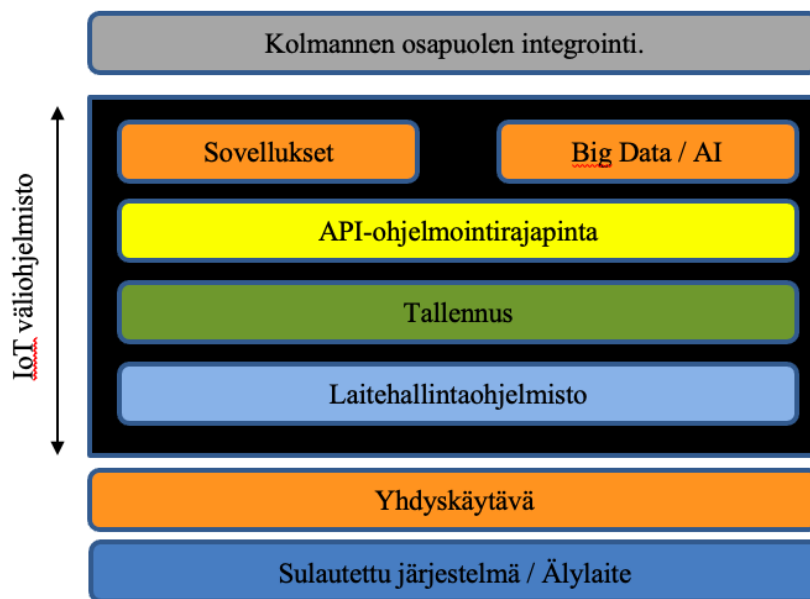
### Tietoliikenne

Internetin ja ulkopuolisten laitteiden yhdistämiseen ohjelmiston tietoliikenne/Internet-protokollilla on oma roolinsa Internet of Things -konseptia (IoT) toteuttamiseen. (24, s. 5, s. 6.)

#### 2.4.1 Esineiden Internet (IoT) -konsepti sulautetussa järjestelmässä

Internet-palvelujen ja mobiililaitteiden laajakäyttö sai alkunsa uuden konseptin nimeltään esineiden internet tai peräsin englanninkielisestä käsitteestä ”Internet of Things” lyhenne IoT, missä termi ”esine” pitää sisällään kaikki laitteet, joita voidaan

hallita ja valvoa verkon välityksellä. Tämän konseptin päätavoitteena on yhdistää nämä esineet Internetin avulla yhteiseen järjestelmään, joka kykenee automatisoimaan ihmisten päivittäiset tehtävät ottaen huomioon kaikki ihmisiä palvelevat laitteet ja palvelut, jotka parantavat ihmisten hyvinvointia. Näin ollen nykyään sulautetut järjestelmät saivat uudet ominaisuudet internetyhteyden ansiosta. Sellaisiin ominaisuuksiin kuuluu järjestelmän tai sen yksittäisen laitteen kaukohallinta, tiedon kerääminen ja analysointi, yhteys käyttäjän kanssa verkkopalvelimen tai mobiililaitteen kautta sekä muiden samankaltaisiin järjestelmiin ja laitteisiin kommunikointi, keskustelu ja yhteistyö. (25.)



Kuva 8. IoT-järjestelmän arkkitehtuuri (26, s.12).

Kuva 8 havainnollistaa sulautetun järjestelmän rakenne IoT-ratkaisun kannalta. Sen pino koostuu eri lohkoista, jotka ovat osa mitä tahansa tyypillistä IoT-järjestelmää. Kuvassa lohkoisten keskustelu tapahtuu sekä alhaalta ylös että ylhäältä alas.

Alimmaisena lohkona on tietoa keräävä ympäristöstään sulautettu järjestelmä tai älylaite, joka on yhteydessä laitehallintaohjelmistoon yhdyskäytävän kautta. Sitten tallennuksen avulla laitehallintaohjelmisto säilyttää yhteyden ja saadun tiedon sensorin ja API-ohjelmointirajapinnan (Application Program Interface) välissä. API-ohjelmointirajapinta on vastuussa käyttöliittymän luomiseksi sensorin, saadun tiedon ja

sovellusten välille. Sovellukseksi voi olla esimerkiksi mobiilisovellus, web-sivu tai mikä tahansa API-ohjelmointirajapinnalle yhteensopiva yksikkö, joka kykenee hallitsemaan ja valvomaan äylaitteita. Sovelluksen lisäksi saatuja tietoja saattaa käyttää analysointiin, koneoppimiseen tai tekoälyyn. Yllä mainittujen tasojen ylin lohko on mahdollinen kolmannen osapuolen integrointi. Se toimi sovittimena silloin, kun liitetty toinen IoT-järjestelmä tai -palvelu ei ole yhteensopiva olemassa olevalle API-ohjelmointirajapinnalle. (26, s.12.)

#### 2.4.2 Kehitysalustojen käyttö sulautetun järjestelmän rakentamiseen

Kaikkia yllä mainittuja sulautetun järjestelmän ja IoT-ratkaisuja ominaisuuksia ja vaatimuksia onnistuneesti toteutetaan kehitysalustoilla. Nykyään on saatavana laaja valikoima eri valmistajan kehitysalustoja. Kuten oli jo mainittu aikaisemmin, yhteen mikrosiruun rakennettu mikroprosessori, muistit ja IO-portit muodostavat mikrokontrollerin. Se antaa mahdollisuuden käyttää yhteistä rajapintaa kaikille mikrokontrollerin ominaisuuksille ja tekee helpommin sen liittymisen sulautettuun projektin piiriin. Tästä johtuen kehitysalustojen valmistajat pitävät mikrokontrolleria parempana vaihtoehtona tehokkaamman mikroprosessorin sijaan.

Mikrokontrolleriin pohjautuvat kehitysalustan korkea hyötysuhde, suuri muistikapasiteetti, mahdollisuus verkkokommunikaatioon. Samalla alhaiset virrankulutus ja hinta houkuttelevat sekä yksittäisiä ohjelmoijia että suuria IoT-laitteiden valmistajia. Markkinoilla on saatavilla suuri valikoima mikrokontrolleripohjaisia kehitysalustoja. Tämä projekti oli toteutettu nimeltään ESP32 (Espressif Systemsin kehittämä mikrokontrolleri) -mikrokontrollerin avulla, jota kutsutaan myös moduuliksi.

Kyseisen moduuli on ESP32-sarjan kehitysalusta, jonka on kehittänyt puolijohdevalmistaja Expressif Systems. Tähän mennessä yhtiö on tuonut markkinoille suosittuja ESP8266-, ESP32-, ESP32-S-, ESP32-C-sarjan siruja, moduuleja ja kehitysalustoja. ESP-mikropiirisarja on edullisten ja pienen virrankulutusten mikrokontrollerien perhe, joissa on integroidut Wi-Fi- ja Bluetooth-ohjaimet ja antennit. Ensimmäinen vuonna 2014 julkaistun 32-bittisen prosessorin ESP8266 on luonut nykyään laajasti käytetyn ESP32-sarjan. Sen mikrokontrollerien rakenne perustuu joko

Tensilica Xtensa LX6- tai Xtensa LX7 -mikroprosessoriin, jotka on varustettu yhdellä tai kahdella ytimellä kellotaajuudella joko 160 tai 240 MHz ja suorituskyvillä 600 DMIPS (Dhrystone millions of instructions per second.) Dhrystone-suorituskykytestin mukaan. Muistin kapasiteetti vaihtelee riippuen mallista 320Kb, 400Kb, 520 Kb RAM ja 128 Kb, 384 Kb, 448 Kb ROM (27). Espressif Systemin ESP32-sirua käyttävät muutkin valmistajat omien alustojen versioiden kehityksessä. Esimerkiksi, käytetty projektissa LVZ-R1-laite koostuu wESP32-kehitysalustasta ja kiinnitetty siihen 4 tunnistinta, kuten BME680 lämpötila-, kosteus-, paine-, kaasumittarien anturimoduulina, MLX90640-infrapunälämpöanturi, T6713-hiilidioksidianturi ja äänitunnistin.



Kuva 9. wESP32-kehitysalusta ja wESP32-prog USB-alimoduuli.

Kuvassa 9 on esitetty kehitysalusta wESP32 ja sen ohjelmointiin tarkoitettu wESP32-prog USB-alimoduuli. Molemmat ovat amerikkalaisen Silicognition LLC -yhtiön tuotteet. Yhtiö keskittyy sulautettuihin IoT-järjestelmiin kehittämiseen. Taulukossa 2 on esitetty wESP32-kehitysalustan ominaisuuksia. (27, 28.)



Taulukko 2. Silicognition kehitysalustan wESP32-ominaisuudet.

Ominaisuudet	wESP32
Moduuli	ESP32-WROOM-32
CPU	ESP32-D0WDQ6
Ytimien määrä	2
Mikrokontrolleri	Xtensa LX6 32-bit
Kellotaajuus, MHz	240
ROM, Kb	448
SRAM, Kb	520
FLASH, Mb	4
GPIO	15
SPI/I2C/I2S/UART	2/2/0/2
Ethernet	kyllä
Wi-Fi	kyllä
Bluetooth	kyllä
ADC	kyllä
DAC	kyllä

### 2.4.3 Laitteiston tiedonsiirtoprotokollat

On olemassa useita protokollia, joiden avulla tapahtuu kontrollerin ja laitteiston kommunikointi. Yleisimmin käytetyt protokollat ovat GPIO (General Purpose Input Output), Analoginen / jännitteen lukema, UART (Universal Asynchronous Receiver Transmitter), SPI (Serial Peripheral Interface) ja I2C (Inter IC, tiedonsiirtoväylä) (26).

## GPIO

GPIO on yksinkertainen kommunikaatiomekanismi, johon kuuluu yksi liitin ohjaimen ja anturin / toimilaitteen välillä. Tämä protokolla on melko hyvä muutaman bitin datan lähettämiseen (26).

## Analoginen lukeminen

Anturit toisinaan paljastavat mittarinsa jännitteen muodossa, ja mikrokontrolleri voi käyttää sisäistä analogisesta digitaalimuunninta (Analog Digital Converter, ADC) lukemaan tiedon jännitteen ja yhdistää sen listaan (26).

## UART

Tämä protokolla tukee valtavien tietomäärien siirtoa. Yleinen asynkroninen vastaanotin ja lähetin (UART) tarkoittaa tapaa, kun mikrokontrollerin lähetin kytketään anturivastaanottimeen ja vastaavasti anturilähetin kytketään mikrokontrollerin vastaanottimeen. Esimerkki UART-pohjaisesta anturista on sormenjälkitunnistin. Sormenjälkitunnistimen tiedot lähetetään ohjaimen UART:n kautta jatkokäsittelyä varten, kun käyttäjä on skannannut sormen. (26.)

## SPI

SPI-protokolla mahdollistaa anturin ja mikrokontrollerin vaihtaa tietoja kahden tietoliikennelinjan kautta. SPI on väylä, joka tukee useita antureita tai toimilaitteita, joita ohjataan yhdellä väylällä. Protokolla käyttää kolmea porttia, jotka on kytketty anturin ja ohjaimen väliin. Esimerkiksi Bosch BME280 -kosteuspaineanturi tukee SPI-protokollaa kommunikoimaan mikrokontrollerin ja anturin välillä. (26.)

## I2C

Inter-Integrated Circuit, lausutaan I-Squared-C tai I-two-C, joka käyttää yhtä datalinjaa ja kellolinjaa kommunikoimaan useiden ohjaimen kytkettyjen anturien välillä. Tämä protokolla on erittäin hyvä kehitysalustoilla kiinnitetyn tai lähellä toisista olevien laitteiden kommunikointiin. (26.)

## 2.4.4 Ohjelmistotuki

### Espressif ESP-IDF (Espressif IoT Development Framework)

Espressif Systems -yhtiö ei suunnittele ainoastaan IoT-siruja ja kehitysalustoja vaan myös käyttöjärjestelmiä ja sovelluskehysiä. ESP-IDF on ESP32:n virallinen Espressif-sovelluskehys. Koska ESP-IDF tarjoaa äärimmäisen joustavuuden, se on heti yhteensopiva wESP32:n kanssa. Sen ohjelmointi tapahtuu C- tai C++-ohjelmointikielillä.

Sulautetun järjestelmän kannalta tärkein tavoite oli varmistaa järjestelmän tehtävien käsittelynopeus, luotettavuus ja vakaus prosessorin alhaisesta tehokkuudesta ja mikrokontrollerin vaatimattomista resursseista huolimatta. Oli tarpeen selvittää, kuinka ajaa useiden antureiden ohjelmia samanaikaisesti jättämättä järjestelmää ylikuormitetuksi. Sitä varten ESP32 FreeRTOS (Real-time operating system) -käyttöjärjestelmän käyttäminen on ehdoton työkalu. Sitä on käytetty hyödyntämään paremmin kahta nopeaa prosessoria ja hallitsemaan lukuisia sisäänrakennettuja oheislaitteita FreeRTOS-käyttöjärjestelmä myös perustuu C- ja C++-ohjelmointikieleen.

ESP-IDF-projekteihin toteuttamista varten on olemassa suuri laaja työkaluja. Suosituimmat työkalut ovat Microsoft Visual Studio Code- ja Eclipse-ympäristöt sekä Arduino IDE -kehitysympäristö ESP32-tuella. MicroPython tarjoaa rikkaan Python 3 -ympäristön pienille mikrokontrollereille, joten ohjelmointi on mahdollista toteuttaa myös yleisellä Python-ohjelmointikielellä. Lua-RTOS-ESP32- ja Lua-ohjelmointikieleen perustuva NodeMCU (Node and Micro-Controller Unit) tarjoavat ESP32:ta tukevan ohjelmointiympäristön. Mongoose OS on IoT-laiteohjelmiston kehityskehys, joka myös tukee ESP32:ta. Sen mukana tulee paljon taustapuolen toimintoja laitteen hallintaan, kuten laitteen kehitysalustan ja OTA (Over-The-Air) -päivitykset.

### 3 Opinnäytetyön toteutus

#### 3.1 Työn alkuvaihe

Opinnäytetyön alussa ensisijaisena tehtävänä oli saada LVZ-R1-laite toimimaan, yhdistää se tietokoneeseen, perehtyä laiteen mikrokontrollerin ohjelmointiin ja saada ensimmäiset tulokset. Luvussa käsitellään n, miten saatiin laite toimimaan. Sen anturitiedot esitettiin päätteessä.

##### 3.1.1 Laitteen asentaminen ja ohjelmointityökalujen valmistelu

LVZ-R1 on PoE (Power over Ethernet) -teknologian yhteissopiva laite, mitä tarkoittaa, että se saa virran RJ45-portin kautta. Tässä tapauksessa kyseistä porttia ei käytetä vain virransyöttöön, mutta myös Ethernet-verkkoon kytkeytymiseen. Laitteelle aiemmin asennettu sovellus sisältää yksinkertaisen HTML (HyperText Markup Language) -sivun, joka visualisoi antureiden tulokset. Lisäksi sovelluksessa käytetty tukevaa verkkoprotokollaa DHCP (Dynamic Host Configuration Protocol) Espressif-ohjelmistokehys. DHCP-verkkoprotokollan tehtävänä on jakaa IP-osoitteita uusille lähiverkkoon kytkeytyville laitteille, mikä meidän tapauksessamme oli LVZ-R1. Näin ollen, kun käytetään kytkettyä paikalliseen lähiverkkoon reititintä sekä LVZ-R1-laitteen IP-osoitetta ja ethernet-kaapelia saimme laitteen toimimaan ja esittämään HTML-sivulle antureiden mittaamat tiedot. Kuvassa 10 on havainnollistettu LVZ-R1-laitteen kytkentäprosessi ja sen tuottama HTML-sivu.



Kuva 10. LVZ-R1 kytkettynä WLAN-reitittimeen PoE-virtalähteen kautta.

LVZ-R1-laitteen ohjelmointi tapahtui liittämällä se tietokoneeseen USB-portin kautta. Tätä varten laitteen kotelo avattiin siten, että wESP32-prog alimoduulin ja USB-kaapelin kautta voitiin liittää tietokone suoraan mikrokontrolleriin. Kuvassa 11 on esitetty laitteen liittämismenettely kokonaisuudessaan USB-kaapelin ollessa valmiina kytkettäväksi tietokoneeseen.



Kuva 11. LVZ-R1, PoE-virtalähde, WLAN-reititin, ja usb-kaapeli.

Mikrokontrollerin ohjelmointi toteutettiin käyttäen Visual Studio Codea, eli VS Code -kehitysympäristöä, mihin asennettiin PlatformIO IDE-laajennus-alusta. PlatformIO on monialustaistainen ja monia ohjelmistokehyksiä tukeva työkalu sulautettujen järjestelmien luomiseen C- ja C++ -kielellä. Se antoi mahdollisuuden käyttää ESP32-alustan virallista Espressif-sovelluskehystä ESP-IDF, sekä sen FreeRTOS-reaaliaikaista käyttöjärjestelmää. ESP-IDF FreeRTOS on muokattu versio vanilla FreeRTOS:sta, joka tukee käytettyä wEsp32-alustossa symmetrisiä moniprosessoria (SMP). Ohjelmoinnin lisäksi Visual Studio Code -ohjelmointiympäristö tarjosi koodin lataamisen mikrokontrolleriin, sen käynnistämisen ja koodin toiminnan seurannan konsolin kautta.

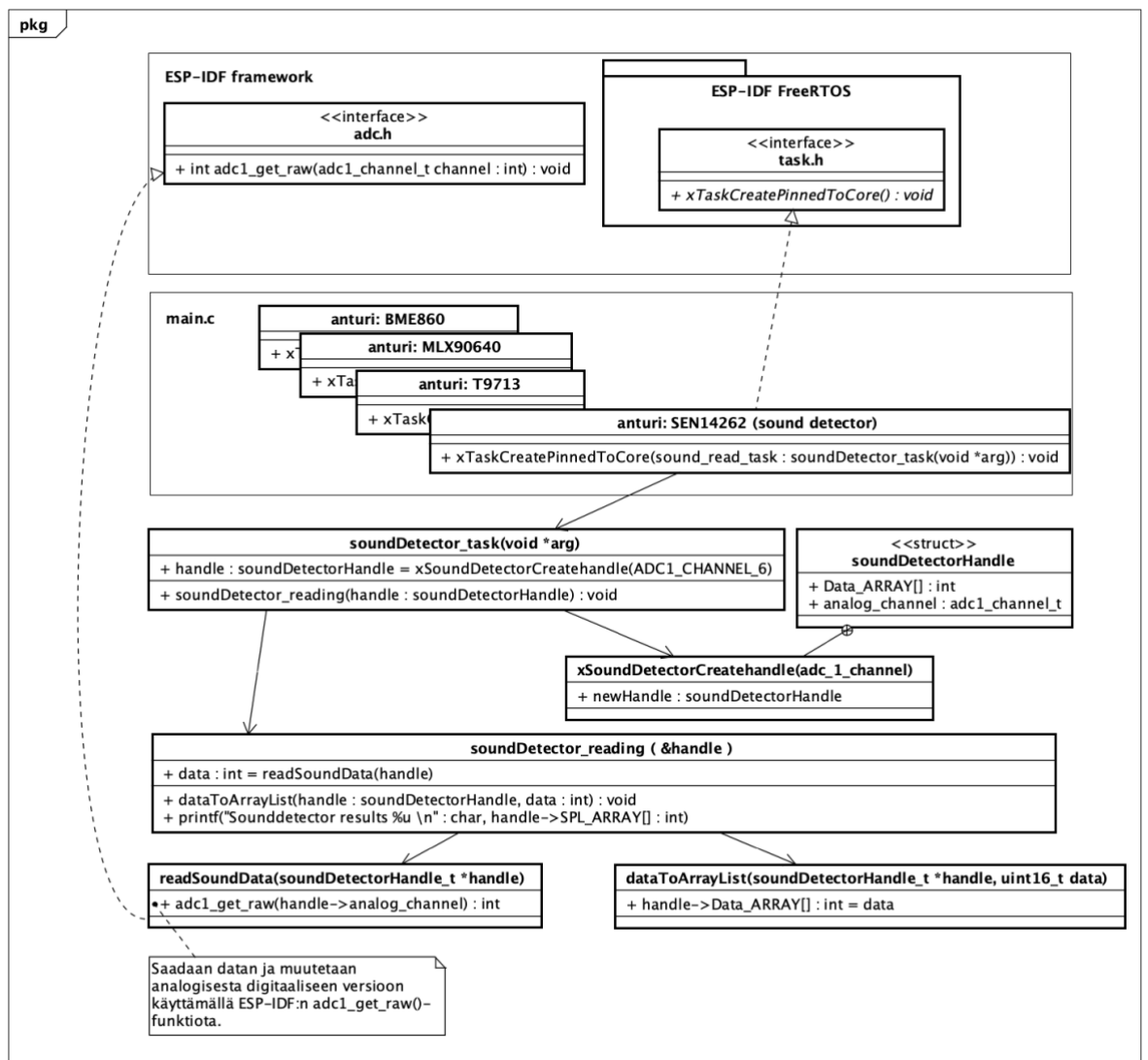
PlatformIO:n ESP-IDF-kehys vaatii Python-laajennuksen asentamisen sekä CMake- ja Ninja-työkaluja koodin kääntämiseksi. CMake on apuohjelma koodin konfigurointia varten, Ninja on vastaavasti käännösautomaatiojärjestelmä. Kaikki yllä mainitut laajennukset ja kirjastot asennettiin VS Code -ympäristön käyttäen.

### 3.1.2 Ensimmäinen testidata (äänianturi)

LVZ-R1-laite sisältää neljä eri tunnistinta, joista ensimmäinen BME680-anturi pystyy mittaamaan haihtuvien orgaanisten yhdisteitä VOC (Volatile organic compounds), lämpöä, kosteutta, painetta. Toinen T6713 antaa tietoa hiilidioksidipitoisuudesta. Kolmas SEN-14262 mittaa äänenvoimakkuutta, MLX90640 on infrapunalämpöanturi.

Ensimmäinen askel sen ymmärtämiseksi, kuinka antureiden tuottama data saadaan esille ohjelmallisesti, alettiin tutkia äänenvoimakkuutta mittaavasta anturista, sillä sen tuottama äänen voimakkuutta kuvaava analoginen data ei vaadi tai edellytä muokkaamista ja tulee luettavaksi suoraan Visual Studio -konsolista. Äänianturi paljastaa mittarinsa jännitteen muodossa, missä jännite seuraa äänen amplitudia, äänikortilla sen on verhokäyrä-nimeltään (engl. envelope) lähtösignaali (29). wEsp32-kehitysalustan käyttöohjelman mukaan äänianturin signaaleja vastaanotetaan GPIO34-portin kautta, joka on alustan analoginen A6-tulo (28, s. 4, s. 5). Koodin tavoitteena oli saada vaihtuvia jännitearvoja, muuttaa ne digitaaliseksi muodoksi sisäisen analogia-digitaalimuuntimen (ADC) avulla ja siirtää tiedot taulukkoon. Taulukon data tulostettiin päätteeseen. Ohjelma toteutettiin C-ohjelmointikielellä.

Ohjelmointikoodissa käytettiin FreeRTOS:n `xTaskCreatePinnedToCore()` API-funktiota luodakseen moniajaja sulautetuille reaaliaikaisille järjestelmille. Kyseisen funktion avulla jokaisen anturin toiminnolle järjestelmä jakaa prosessoriaikaa ja valita, mikä tehtävä tulisi suorittaa tietyllä aikavälillä ja tietyllä prosessorilla moniprosessorijärjestelmässä. Kuvassa 12 on esitetty, miten äänianturin toiminnon, eli tehtävän luottiin FreeRTOS-käyttöjärjestelmän käsittelyyn `xTaskCreatePinnedToCore()` API -funktiolla.



Kuva 12. Äänianturin datan saamisen toiminnon ja prosessien mallintaminen.

Kuvassa 12 olevasta mallista nähdään, että `xTaskCreatePinnedToCore()`-funktio kutsuu äänianturin datan saamisesta funktion nimeltään `soundDetector_task(void *arg)`. Seuraavaksi luottiin `soundDetectorHandle`--tietorakenne, jonka muuttujina ovat taulukko ja alustan analogisen sisäportin parametri. Sitten `soundDetector_reading()`-funktion sisällä olevilla `readSoundData()`-, `dataToArray()`- ja `printf()`-funktioilla vastaavasti saatiin data, alustettiin taulukko ja tulostettiin data konsoliin. Kuva 13 esittää äänisensorilta saadut tiedot tulostettuna päätteeseen.

```
Starting measurement in 1 seconds...
Starting measurement in 0 seconds...
Starting now.
-[0;32mI (6250) app_main: Created task: ADC_read
-[0m
-[0;32mI (6250) Sound detector new: Creating new detector handle=[0m
-[0;32mI (6250) Sound detector new: New detector handle created=[0m
Sounddetector results 0: 5600
Measurement number: 1
Sounddetector results 0: 8800
Measurement number: 2
Sounddetector results 0: 8800
Measurement number: 3
Sounddetector results 0: 8800
Measurement number: 4
Sounddetector results 0: 8800
Measurement number: 5
Sounddetector results 0: 10550
Measurement number: 6
Sounddetector results 0: 10550
Measurement number: 7
Sounddetector results 0: 10550
Measurement number: 8
Sounddetector results 0: 10550
Measurement number: 9
Sounddetector results 0: 10550
Measurement number: 10
Sounddetector results 0: 10550
Measurement number: 11
Sounddetector results 0: 10550
Measurement number: 12
```

Kuva 13. Äänianturin tulokset tulostettiin päätteeseen.

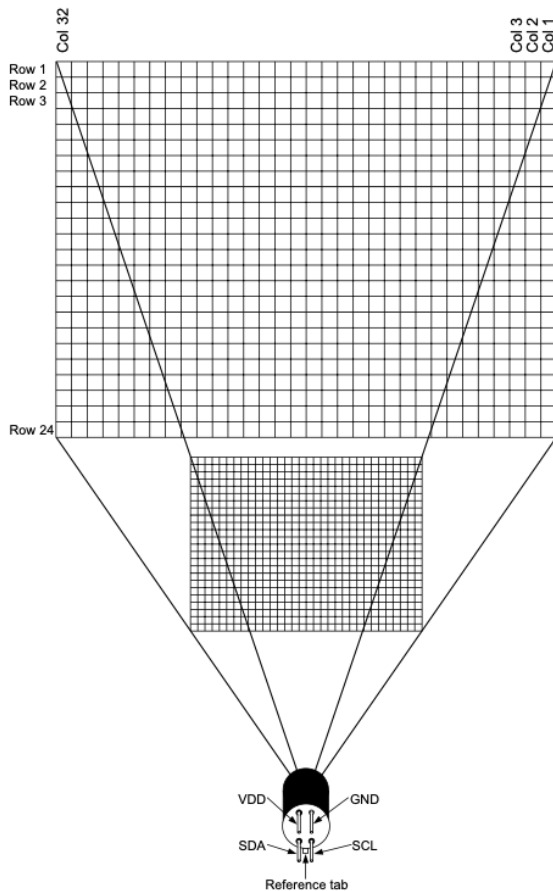
### 3.2 Infrapunakameran datan saaminen ja käsitteleminen

Tutustuminen käytetyn projektissa LMX90640-infrapunälämpökameran toimintaperiaatteeseen aloitettiin laitteen valmistajan kotisivultaan olevasta teknisistä dokumentaatioista. Laitteen sisärakennuksen ja teknisin tietojen lukuun ottamatta asiakirjat kuvailevat tiedonkeruuprosessia ja MLX90640 API eri funktioita ja mikrokontrolleriin yhdistämisajuria.

MLX90640-yksikkö on valmistettu standardin mukaisessa 4-kytkentäisessä TO39-kotelossa digitaalisella liitännällä. Sen lämpökamera on joukko 768 erittäin herkkää tehdaskalibroituja IR-elementtiä, eli pikseliä, jotka on järjestetty 24 riville ja 32 sarakelle. Jokainen IR-elementti mittaa kohteen lämpötilaa jopa 300 °C paikallisen



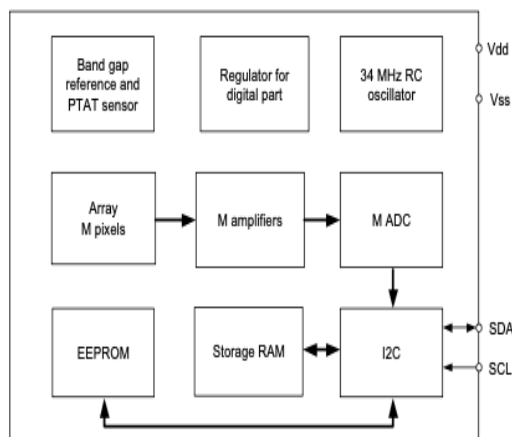
näkökentän FOV (Field Of View) sisällä. Kuvassa 14 on havainnollistettu lämpösensorien pikselien järjestystä, missä jokainen pikseli tunnustetaan sen rivi- ja sarakepaikalla Pix (i, j). (31.)



Kuva 14. Lämpökameran pikselien järjestys (31, s. 10).

Paitsi lämpökameraa, MLX90640-yksikko sisältää anturi sirun jännitettä VDD (Voltage Drain Drain) ja kotelossa ympäristön lämpötilaa  $T_a$  (Ambient Temperature) mittaavia anturit. Kuvassa 15 on esitetty MLX90640-yksikön tärkeimmät osat lohko-kaaviona, joka kuvailee laitteen toimintaa yleisellä tasolla. Lohkosta (Array M pixels) saatu lämpösensorista data, eli jännitearvo, menee vahvistuksen (lohko M amplifier) jälkeen sisäiseen analogia-digitaalimuuntimeen (lohko M ADC). ADC:ltä raakadata (IR data), joka on verrannollinen anturin vastaanottamaan infrapunaenergiaan, pääsee I2C (lohko I2C)-protokollan kautta RAM-muistiin. Lohko nimeltään PTAT (Proportional to Absolute Temperature)-sensori kuvailee yksikön kotelon ympäristön lämpötila  $T_a$  mittaava anturi. Näin ollen, kaikkien 786 termoelementtien raakadata (IR)-, ympäristön

lämpötilan Ta- ja jännitteen VDD-tulokset tallennetaan sisäiseen RAM-muistiin. Digitaalisen I2C-liittännän kautta muistissa olevaa raakadataa voidaan käsitellä wEsp32-kehitysalustalla, jolloin voi rakentaa lämpökuvan tai laskea lämpötilan katseltavan kohtauksen kussakin kohdassa. (31.)



Kuva 15. MLX90640-lämpökameran osien lohkokaavio (31, s. 1).

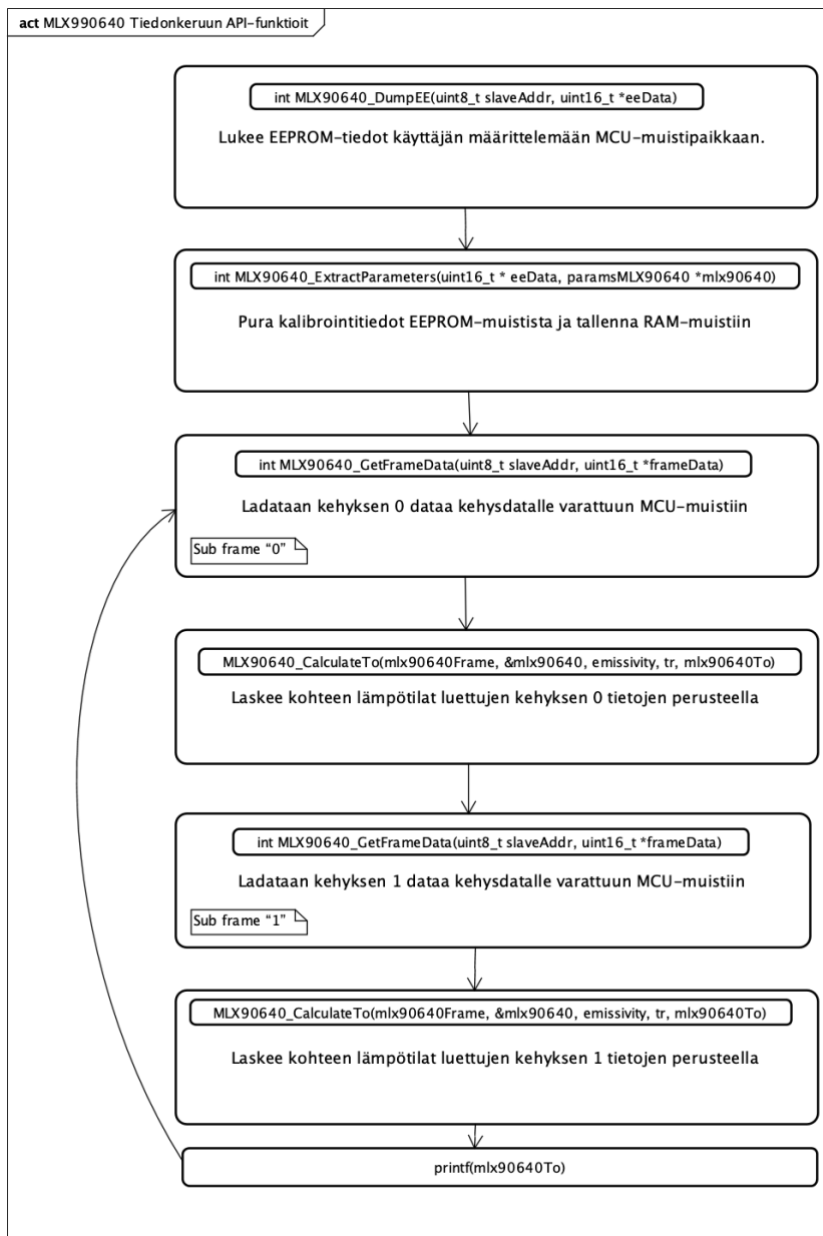
EEPROM (Electrically Erasable Programmable Read-Only Memory) -muisti (lohko EEPROM) sisältää yksikön kaikki kompensointi-, kalibrointi- ja konfigurointiparametrit jokaiselle IR-elementille, jotka ovat tarpeen tarkkojen lämpölaskelmien suorittamiseksi. Nämä laskelmat sisältävät ympäristön lämpötilan laskennan, pikselipoikkeama laskelman, pikselien välisen herkkyyseron kompensoinnin, kohteen emissiokyvyn kompensoinnin ja kohteen lämpötilalaskelman. Paitsi sitä EEPROM-alueella ovat muokattavissa olevia käyttäjäparametreja, kuten konfigurointirekisterit, I2C-osoite. Kaikki yllä mainitut parametrit on huomioitava ohjelmakoodin luomisen prosessissa. Tämän seurauksena varatussa muistissa tulee olla vähintään 832 solua, joista 768 antureille ja 64 kalibrointiparametreille. I2C-lohko kuvailee yksikön yhdistämisprotokollaa, jolla SDA (Serial Data)- ja SCL (Serial Clock) -porttien kautta yhdistetään mikrokontrolleriin. (31). Projektimme nähden yksikön SDA- ja SCL-porttien yhdistäminen wEsp32-alustaan toteutettiin GPIO13- ja GPIO4-porttien kautta.

Koko RAM-alue on jaettu kahdelle sivulle, joille pääsyä ohjaavat konfigurointirekisterit joko shakkikuviolla tai lomitetulla lukukuviolla. EEPROM-muistiin tallennetut

kompensointiparametrit on tehtaalla kalibroitu shakkikuvioiden käyttöä varten, mikä tuottaa tarkimmat tulokset käytettäessä tätä tilaa. Shakkikuvio-tila on valittu oletuksena. Konfigurointi- ja ohjausrekistereillä voidaan asettaa anturin toimintaparametrit, virkistystaajuus, ADC-resoluutio, mittaustila (jatkuva tai askeltila), lepotila, I2C-tilat FM tai FM+ (Fast Mode). FM ja FM+ ovat I2C:n tiedonsiirtoväylän tilat, jotka määrittävät tiedonsiirtonopeus 400 kbit/s tai 1 Mbit/s vastaavasti. Uudelleenkäynnistyksen yhteydessä tiedot näiden EEPROM-muistissa olevien rekisteripaikkojen vastaavista kopioista peilataan RAM-muistin toimintarekisteripaikkoihin, jolloin laite valmistetaan välittömästi käytettäväksi. Tämä mahdollistaa oletusarvojen muuttamisen, koska ne tallennetaan EEPROM-muistiin sen sijaan, että ne olisi koodattu laitteeseen. (31.)

MLX90640-yksikkö voi tuottaa 64 kuvaa sekunnissa FPS (Frame Per Second) ja sitä on mahdollista asettaa alkaen 0,5 kuvaa sekunnissa. Laitteen EEPROM-muistiin on mahdollista ohjelmoida haluttu virkistystaajuus, jolloin laitetta ei tarvitse asentaa uudelleen aina, kun se käynnistetään. Käytettävissä on kaksi toimintatilaa: laite voi jatkuvasti ottaa dataa IR-elementeistä ohjelmoidulla virkistystaajuudella tai se voi ottaa vain yhden kehyksen valitulta sivulta.

Ohjelmoinnin näkökulman nähden oli hyödyllistä tutustua valmistajan suositteluun mittauksen prosessin järjestykseen, jonka mukaan toteutettiin lämpökamerasta saadut tulokset näkyviin päätteessä. Kuva 16 kuvailee API-funktiot, jotka varmistavat oikean tiedonkeruun.

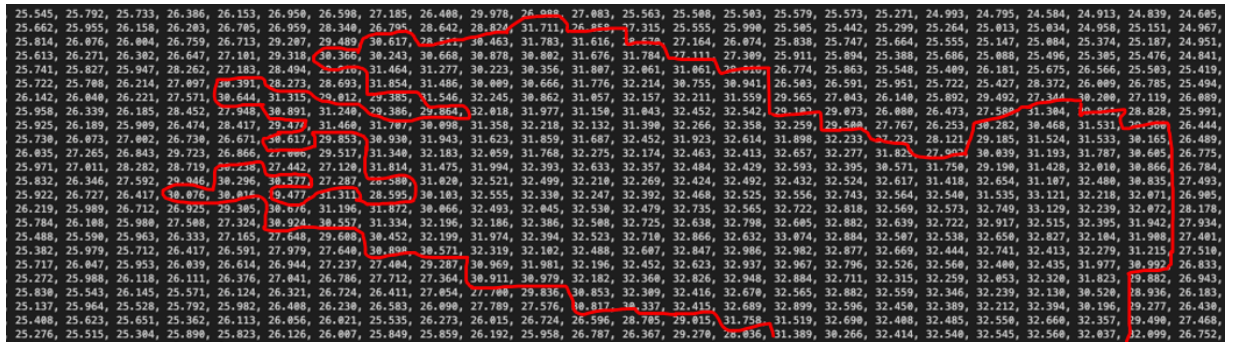


Kuva 16. MLX90640:n toiminnan prosessien järjestys datan saamiseen.

Kuten aiemmin mainittiin, lämpö kuvadatan kokoelma sisältää kaksi kehystä (sub frame 0 ja sub frame 1), joiden data vuorotellen täyttää koko lämpökuvalle varatun RAM-alueen shakin tai lomitetun järjestyksessä. Tämä takaa nopean ja luotettavan lämpökuvan tuloksen.

Kuvassa 17 on esitetty lämpökameran saadut tulokset, josta näkyvät alisivun 0 (Subpage 0)- ja alisivun 1 (Subpage 1) -kehysten arvot, sekä tulostettu esille



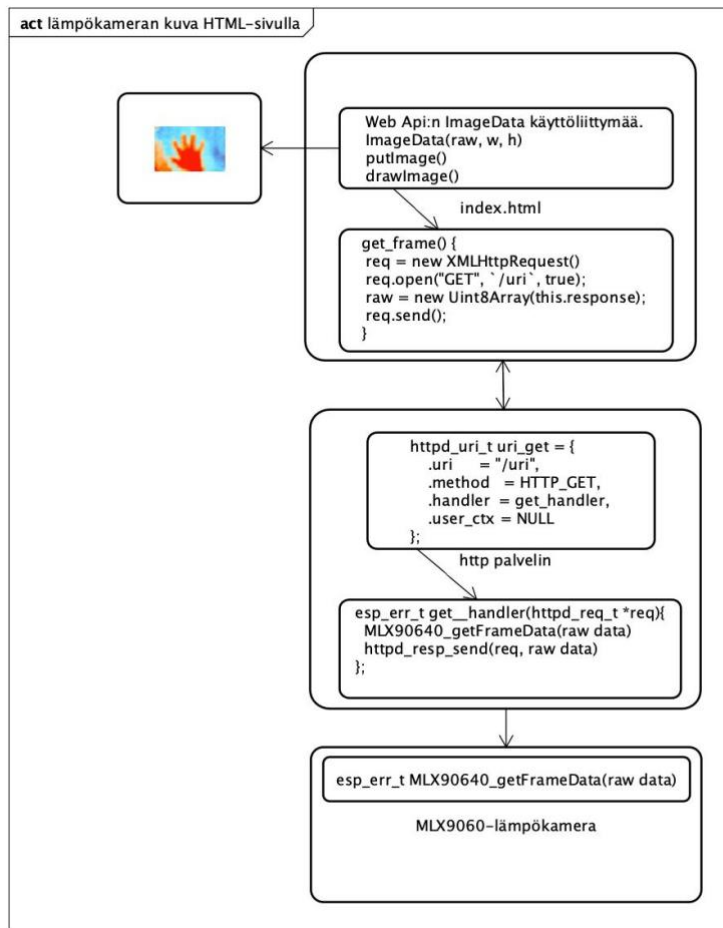


Kuva 18. Lämpökameran tuloksiin vaikuttaa sen edessä oleva käsi.

### 3.2.1 Web-käyttöliittymän luominen

ESP-IDF-kehitysympäristön HTTP (HyperText Transfer Protocol) -palvelinkomponentti tarjoaa mahdollisuuden käyttää verkkopalvelinta wESP32-kehitysalustalla. Sen funktioiden avulla toteutettiin asiakas- ja palvelinpuolen keskipälistä kommunikointia sekä luotiin Web-käyttöliittymän, jolla asiakas pystyy seuraamaan lämpökameran tuloksista reaaliaikaisesti. HTTP-palvelimen ja GET-metodin avulla siirrettiin lämpökameran datan asiakaspuolelle. Asiakaspuolella käyttäen HTML-keittä ja DOM (Document Object Model) -rajapintaa luotiin HTML-dokumentin, eli sivun. HTML-sivulle upotetuilla Javascript-funktioilla ja metodeilla rakennettiin tiedon käsittelymekanismi esittämään reaaliaikaisen videon asiakkaan käyttöliittymässä.

Kuvassa 19 on esitetty MLX90640-lämpökameran värillisen kuvan luomisen prosessi HTML-sivulle. Tätä varten palvelinpuolelle luotiin URI (Uniform Resource Identifier) -käsittelijän `uri_get{}`-rakenteen ja asiakkaan GET-pyyntöt hyväksyvä URI-käsittelijän `get_handler`-funktio. Se kutsuu lämpökamerasta vastaanotetun tietojen sisältävän `MLX90640_getFrameData()`-funktion ja lähettää sen asiakaspuolelle `httpd_resp_send()`-funktioilla.



Kuva 19. MLX90640-lämpökameran kuvan luomisen prosessi.

Asiakaspuolella `get_frame()`-funktio vastaanotti tiedot. Sitten käyttäen Web Api:n (Web Application Programming Interface) `ImageData`-käyttöliittymää ja sen metodia muutettiin dataa värilliseksi kuvaksi. Luotiin uuden `ImageData`-olion ja muunnettiin datan värilliseksi kuvaksi `ImageData()`-metodilla, jossa tiedot tallennetaan RGBA-arvojen muotoon. Asennettiin `putImageData()`-metodilla `ImageData`-olion kuvan tiedot HTML-sivun `canvas`-elementille. Lopussa `drawImage()`-metodin avulla piirrettiin kuvan esille. Lämpökameran muuttuvia tuloksia esittämiseen reaaliaikaisesti käytettiin JavaScriptin `setInterval()`-metodia, jolla `get_frame()`-funktiota kutsuttiin tietyin aikavälein. Esimerkkikoodi 1 on esimerkki reaaliaikaisen videon toteutuksesta HTML-sivulla.

```
function get_stream() {
  if ($("#stream").checked) {
    myInterval = setInterval(function () {
      get_frame();
    }, 250);
  } else {
    clearInterval(myInterval);
  }
}
```

**Esimerkkikoodi 1.** JavaScript:in `setInterval()`-metodi kutsuu `get_frame()`-funktion 250 millisekunnin välein, eli 4 Hz. Metodi `setInterval()` jatkaa funktion kutsumista, kunnes kutsutaan `clearInterval()` tai ikkuna suljetaan.

### 3.2.2 Kuvan analysointi ja Ihmisten määrän laskeminen

Kun yksi projektin tavoitteista oli sulautettu järjestelmä, seuraavana tehtävänä oli kehittää ja upottaa wEsp32-alustaan ohjelma, joka pystyisi analysoimaan infrapunakameran dataa ja laskemaan henkilömäärän. Tutustuttuaan nykyaikaisiin IoT-järjestelmiin autojen, ihmisten liikkeen ja objektitunnistukseen seurantaan, päätettiin kokeilla lähestymistapaa nimeltä tietokonenäkö. Sopivaksi vaihtoehdoksi valittiin vankka ja laajalti käytetty OpenCV-tietokonenäkö- ja koneoppimiskirjasto.

Ajatuksena oli asentaa tämä kirjasto wEsp32-alustalle ja toteuttaa tietokonenäkö- ja koneoppimisanalyysi alustaresurssin avulla. Tätä varten oli tarpeen asentaa OpenCV-kirjasto VS Code -ohjelmointiympäristöön. Asennusvaiheessa ilmeni vaikeuksia. LVZ-R1-laitteen ohjelma kirjoitettiin C-ohjelmointikielellä ja vaikka alkuperäisesti OpenCv-kirjasto kirjoitettiin alun perin C-kielellä, nykyinen versio oli saatavilla ainoastaan C++-kielellä. OpenCV:n ohjeiden mukaan OpenCV 2.4:n julkaisun jälkeen C-API on vanhentunut eikä sitä ole testattu ”C”-kääntäjällä. Asentaminen OpenCV 2.4 -version VS Code -ohjelmointiympäristöön epäonnistui koodin kääntämisvirheen vuoksi. Muun muassa saatiin selville, että OpenCV-kirjaston pohjalla kirjoitettu koodi on vaativa wEsp32-aluston suoritustehon kannalta, joten koodin rakentaminen LVZ-R1-laitteen sisään ei ollut järkevä.

Yllä mainitun syiden vuoksi päätettiin lämpökameran tuloksien analysointia suorittaa LVZ-R1-laitteen ulkopuolella käyttäen jo saatua http-käyttöliittymä. LVZ-R1-laitteen käyttöliittymä, joka http-rajapinnan kautta tuo html-sivulle infrapunakameran reaaliaikaisen kuvan, oli sopiva pohja kuvan analysointiin OpenCV-työkaluilla.



Tämäkin mahdollisti kuvan analysoinnin tietokonenäön avulla. Kyseinen menetelmä perustuu kuvankäsittelyohjelmaan, joka pystyy automaattisesti käsittelemään ja tulkitsemaan kamerasta saadun kuvan. Se voi esimerkiksi havaita infrakameran kuvassa olevia esineitä ja seurata niiden liikkeitään. Siten voidaan kehittää ohjelmaa tilan ihmismäärän laskemiseen, joka laskee huoneessa olevien ihmisten määrän sen mukaan, käveleekö henkilö ovesta sisään vai ulos.

Tutkittaessa OpenCV-kirjaston käyttöä jo olevissa Internetissä valmistuneissa projekteissa, valittiin sen toteutuksen Python-rajapinnalla nimeltään OpenCV-Python. Lähteiden esimerkit auttoivat tutustumaan sekä havainnollisesti että kokeellisesti OpenCV:n tietokonenäön periaatteisiin ja liittyviin algoritmiin toimintoon. Projektissamme esitetty ihmisten laskemisen ohjelma kehitettiin Adrian Rosebrockin ja Sergion Canun avoimien blogien pohjalla, jossa tehtiin tarvittavia lämpökameran videomateriaalimme analysointiin muutoksia. (32, 33.)

Lämpökameran videon käsittelyohjelma suoritettiin edelleen Visual Studio Code -ohjelmointiympäristössä. Tietokonenäön toteuttamiseen, Python-rajapinnan ja sen OpenCV-paketin lisäksi, täytyi asentaa useita laskemiseen tehoon tarkoitettuja python-paketteja. VS Code -ohjelmointiympäristö mahdollisti asentamaan sen Python-laajennuksen. Seuraavaksi asennettiin OpenCV-, NumPy-, Dlib- sekä Imutils-kirjstot. NumPy on sisältävä lukuisia funktioita taulukoiden käsittelyyn Python-ohjelmistokirjasto, jota käytetään tieteelliseen ja numeeriseen laskentaan. Dlib on yleiskäyttöinen alustanvälinen ohjelmistokirjasto, joka sisältää koneoppimisalgoritmeja ja työkaluja tietokonenäön ohjelmistojen rakentamiseen. Imutils on ohjelmistokirjasto, joka sisältää sarjan toimintoja peruskuvankäsittelytoimintojen tekemiseen, kuten käännös, kierto, koon muuttaminen, luuranko ja näyttö ja muut työkalut tietokonenäön ohjelmistojen rakentamiseen.

Kuvassa 20 on esitetty asennetut Python 3.9.5 ja sen kirjastojen versiot VS Code -pääätteessä.

```
(base) vladimzubchenko@Air-256MB Counter % python3
Python 3.9.5 (v3.9.5:0a7dcdb13, May 3 2021, 13:17:02)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.5.4-dev'
>>> import numpy
>>> numpy.__version__
'1.21.4'
>>> import imutils
>>> imutils.__version__
'0.5.4'
>>> █
```

Kuva 20. Python ja cv2(OpenCV)-, NumPy- ja Imutils-kirjastot asennettuna VS Code -pääteessä.

Lämpökameran videon käsittelyyn tehtäviksi oli tunnistaa ihmisiä ja erottaa heidät muista olevilla taustalla esineistä, seurata niiden liikkumisesta ja kulkusuuntaa ja samalla laskea tilaan saapuvien ja sieltä poistuvien ihmisten lukumäärä.

### 3.2.3 Videon lukeminen VideoCapture-objektilla

OpenCV tarjoaa videokaappaus-objekti, joka käsittelee kaiken videotiedoston avaamiseen ja sulkemiseen liittyvän tiedon. Objekti sisältää tietoja, joita käytetään kameran tai videotiedoston kehysten lukemiseen. Videon lähteeksi voidaan valita joko etukäteen tallennettu videomateriaali tai suoratoistovideo. Esimerkkikoodissa 2 on esitetty videon lukemisen toteuttaminen VideoCapture-objektilla ja sen tulos. Lämpökamera on asennettu katon alle ja suunnattu pystysuoraan alaspäin.

```

import cv2

cap = cv2.VideoCapture("videoLahde.avi")

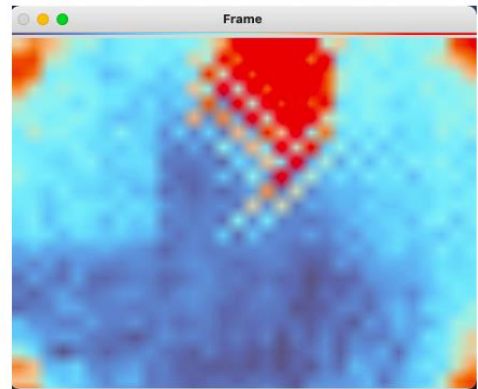
while True:
    ret, frame = cap.read()

    cv2.imshow('Frame', frame)

    c = cv2.waitKey(1)
    if c == 27:
        break

cap.release()
cv2.destroyAllWindows()

```



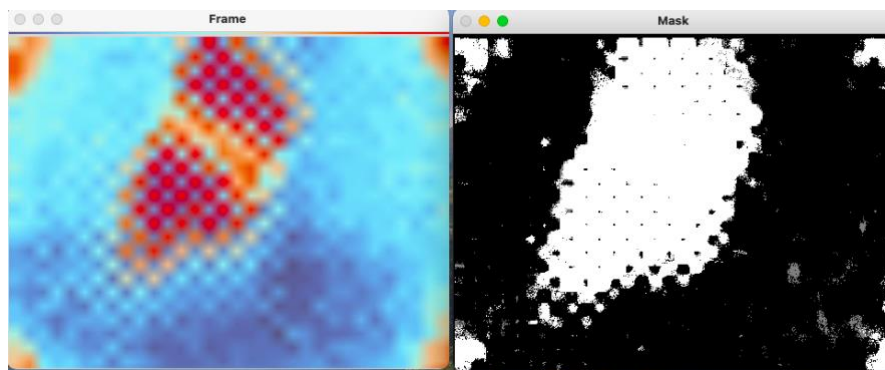
Esimerkkikoodi 2. OpenCV-funktio `cv2.VideoCapture("videoLahde.avi")` vastaanottaa video lähdteen ja luo videokaappauskohteen "cap", missä videolähteenä asennettiin videotiedoston nimen "videoLahde.avi". OpenCV avaa tiedoston ja valmistautuu lukemaan ja käsittelemään sen kehykset `cap.read()`-funktioilla While-toistorakennetta käyttäen. Tässä `ret` on `read()`-funktion palauttama Boolean arvo, ja se osoittaa, onnistuiko kehyksen sieppaus vai ei. Jos kehys kaapataan oikein, se tallennetaan `frame`-muuttujaksi. Ohjelman ajon aikana, `cv2.imshow()`-funktion avulla havainnollistetaan lämpökamerasta vastaanotetun ja OpenCV:n käsittelyssä olevan videon näytölle ja `waitKey()`-funktio mahdollistaa lopettaa ohjelman toiminnon painamalla `esc`-näppäintä.

### 3.2.4 Kohteen tunnistus

Lämpökameran kuvaan nähden OpenCV-kirjaston tehtävänä oli erottaa liikkuvia kohteita. Punaiset kohteet kuvaavat ihmiskehon lämpötilaa ja se eroaa siten videon paikallaan olevasta sinisestä taustasta. Vertailemalla jokaisen video kehyksen pikselien väriä ohjelma pystyy paljastamaan, onko kyseessä taustakuva vaiko liikkuva kohde.

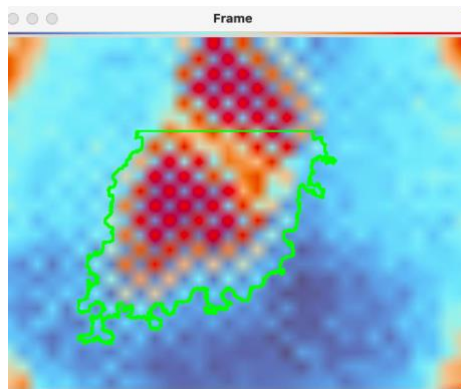
Tätä varten käytettiin taustamallin rakentamismenetelmä, eli kohtauksen staattiset osat tunnistettiin ja sisällytettiin sitten taustamalliin. Kaikki uudet kohteet, jotka ilmestyivät ja poikkesivat taustamallista, käytettiin sitten liikkuvien kohteiden havaitsemiseen. Luotettavampaan tulokseen saamiseen projektissamme käytettiin kahta erilaista metodia, kuten KNN (K-Nearest Neighbors) - ja MOG (Mixture-of-Gaussians) -metodia.

Ensin taustamallin rakentamiseen käytettiin KNN-metodin `createBackgroundSubtractorKNN` (`history`, `dist2Threshold`) -funktioita, jolla luotiin taustamalli. Funktion algoritmin pohjalla on koneoppimisessa käytetty k:n lähimmän naapurin menetelmää. Funktio sisältää kaksi parametria, josta `dist2Threshold`-parametri on kynnyusraja. Se määrittää, onko pikseli tarpeeksi lähellä kyseistä kohdetta ja kuuluuko se siihen. Mitä korkeampi on kynnyusraja, sitä vähemmän pikseleitä funktio määrittää nykyiseen kohteeseen. Tämä parametri ei vaikuta taustamallin päivitykseen. Toinen `history`-parametri kuvailee algoritmin oppimisnopeutta, jolla se oppi taustaa. Jos tietyssä kehyksien määrässä sama pikselin väri jää muuttumatta tai muuttuu heikosti, pikseli määritetään kuuluvan taustamalliin. Sen korkeampi arvo tarkoittaa hitaampaa algoritmin oppimisnopeutta ja tarkempaa taustamallia, koska se ei ota huomioon tilapäisesti ilmaantunutta paikallaan olevia kohteita. Kuvassa 21 on havainnollistettu alkuperäistä Frame-kuvaa ja sen pohjalta luotua Mask-taustamallia KNN-menetelmän avulla.



Kuva 21. Alkuperäisen kuva ja sen taustamalli.

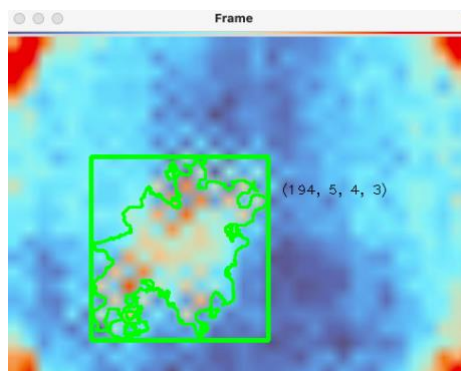
Jotta ohjelma käsittelee vain ihmismäärän laskemiseen tarvittavia kohteita, kuten ihmiskehon lämpötilan perusteella luotuja objekteja tai niiden erottamista toisistaan, niiden koon ja kuvan käsittelyalue oli rajoitettu. Objektien koko saadaan `counturArea()`-funktioilla, joka palauttaa nollasta poikkeavien pikselien määrän objektia kohden ja jota voidaan rajoittaa. Siten vain tietyn kokoiset ja tietyltä alueelta olevat kohteet valittiin käsittelyyn, kun taas toiset, esimerkiksi kuvakohina, poistettiin. Kuvassa 22 on esitetty käytetty ääriviivaa ja käsittelyalueen rajoituksia.



Kuva 22. Kohteen koon ja käsittelyalue.

### 3.2.5 Kohteen seuraaminen

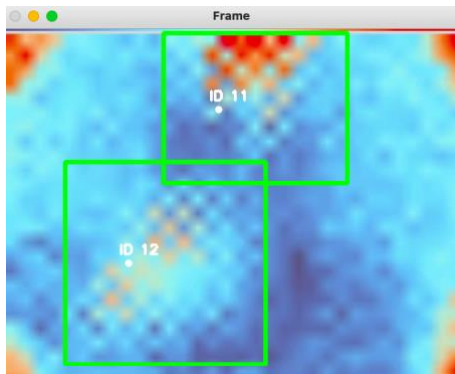
Kohteen seuraamiseksi kutakin esillä olevaa kohdetta ympäröivän ääriviivan koordinaatit saatiin käyttämällä `boundingRect(kohde)`-funktioita. Funktio laskee kohteen ääriviivan suorakulmion ja palauttaa x- ja y-koordinaatit sekä suorakulmion korkeuden ja leveyden. Näin ollen kohteen-koordinaatit ja koot voidaan kerätä taulukkoon ja käyttää tietojen seurantaan. Kuvassa 23 on esitetty kohteen ympäröivän suorakulmion koordinaatit sekä leveys- ja korkeusarvot.



Kuva 23. Kohteen koordinaatit ja koot.

Kohteiden liikkuaessa sekä vasta ilmestyneen, että jo havaitun kohteen muuttuvat koordinaatit pääsivät taulukkoon. Niiden erottamiseksi laskettiin kunkin kohteen keskipistettä, eli sentroidit ja laskettiin niiden välinen euklidinen etäisyys. Jos etäisyys ylittää tietyn kynnyksärajan, lisättiin kohde toiseen taulukkoon uutena kohteena

tunnusnumeroineen. Kuvassa 24 on esitetty kahden kohteen liike alhaalta ylös ja niille osoitetut ID-tunnisteet keskipisteiden välisen etäisyyden mukaan. Siten lämpökameran ohi kulkevia ihmisiä voidaan laskea ID-tunnisteita käyttäen.



Kuva 24. Kahden kohteen erottaminen keskipisteiden välisen etäisyyden perusteella.

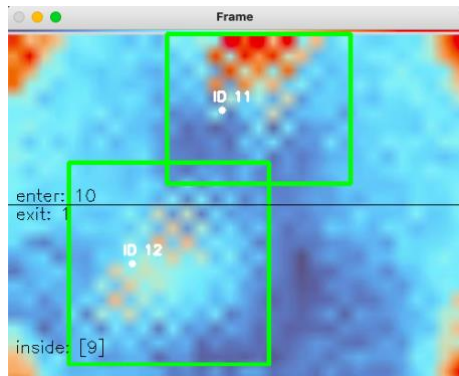
### 3.2.6 Ihmisten määrän laskeminen

Kuten aikaisemmin oli mainittu, huoneessa olevien ihmisten määrän laskemisessa oletettiin, että lämpökamera roikkuisi ulko-oven yllä. Sitten ratkaisuna olisi laskea tilan sisään astuvia henkilöitä ja ulos lähteviä henkilöitä. Lämpökamera riippui siis ovea kohti ihmiskulkusunnasta.

Kun jokaiselle kohdalle on kiinnitetty oma id-tunnus ja sen keskipisteen sijainti, voidaan kerätä taulukkoon kaikki muuttuvat keskipisteiden sijaintikoordinaatit. Niin ollen kohteen liikkumisen suunnan laskemiseen käytettiin sen muuttuvia keskipisteiden tietoja kohteen ensimmäisen paljastamisen jälkeen. Vertailemalla saman kohteen viimeisen keskipisteen koordinaattia sen aiempiin koordinaatteihin lasketaan liikkumisen suuntaa. Ylös ja alas liikkuva kohde tunnistetaan sen nykyään ja edellisten y-koordinaattien keskiarvon välisen eron perustella, esimerkiksi negatiivinen tulos tarkoittaa ylös- ja positiivinen alas-menevän.

Ihmisten määrän laskemiseen asennettiin tunnistusalueelle vaakasuora keskiviiva. Sen y-koordinaatti vertailtiin liikkuvan kohteen keskipisteiden y-koordinaatteihin ja paljastettiin, ylittääkö kohde keskiviivan ja mihin suuntaan. Siten laskettiin sisään

astuneiden ja ulos lähteneiden ihmisten määrä tiloissa. Kuva 25 näyttää käyttöliittymän, joka näyttää keskiviivan, sisään- ja poistuvien ihmisten määrän sekä huoneessa olevien ihmisten lukumäärän.



Kuva 25. Huoneessa olevien henkilön määrän laskeminen

Kuvassa 25 on esitetty tilanne, kun 10 henkilöä tulivat jo huoneeseen, jonka jälkeen huoneesta poistuu 2 henkilöä tunnisteilla ID11 ja ID12. Molempien kohtien liikkumissuunta on alhaalta ylös, joista toinen ID11:lla on ohittanut keskiviivan ja toinen ID12:lla lähestyy sitä.

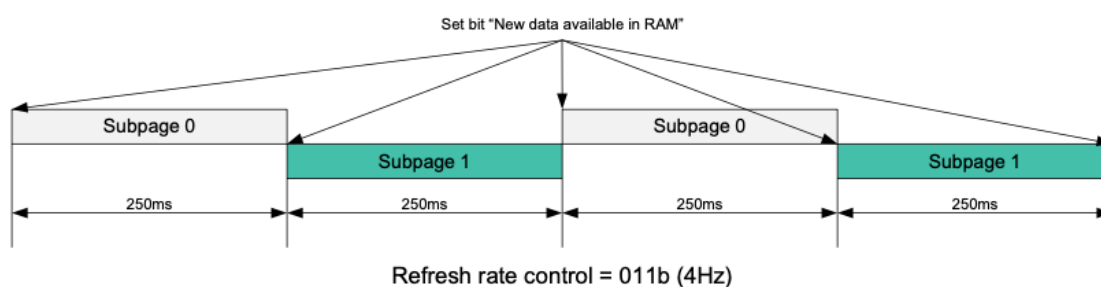
## 4 Tulokset

OpenCv-kirjaston työkaluilla voi käsitellä joko etukäteen ladattua ohjelmaa videota tai reaaliaikaista suoratoistovideota. Ohjelman toimintojen testauksiin pidettiin kotiloissa ja käytettiin yllä mainittua ensimmäistä tapaa. Lämpökamera oli ripustettu oven vieressä kattoon. Kokeissamme simuloitiin eri tilanteita, joissa tiloihin astuivat tai(ja) lähtevät eri määrä henkilöä yksittäin tai joukkona, kameran korkeutta sekä sen virkistystaajuutta muutettiin.

## 4.1 Kokeiden olot, työkalut ja menetelmät

### 4.1.1 Lämpökameran virkistystaajuus

Tulevien testien kannalta on mainittava, että ohjelman on luettava molemmat alisivut, sillä kohteen lämpötilan ( $T_o$ ) laskelmiin tarvittavat tiedot saadaan vain yhdistämällä molempien alisivujen tiedot. Tämän seurauksena  $T_o$ -arvon päivittäminen tapahtuu kaksi kertaa pienemmällä päivitysnopeudella kuin asetettu lämpökameraan virkistystaajuus. (31, s. 12). Kuvassa 24 on havainnollistettu lämpökameran virkistystaajuuden vaikutuksen sen datan saamiseen. Esimerkissä MLX90640-lämpökameran oletusvirkistystaajuus 2 Hz muutettiin 4 Hz:iin. Siten tulos saadaan kahdesti sekunnissa, eli taajuudella 2 Hz.



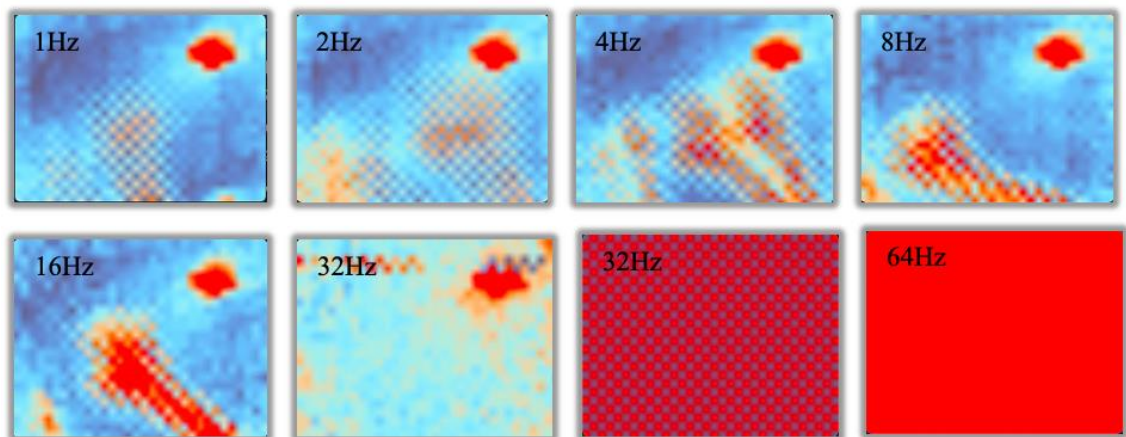
Kuva 26. Lämpökameran virkistystaajuuden ja sen luoman kuvan synkronointi (31, s. 12).

Lämpökameran virkistystaajuuden ja sen luoman kuvan synkronointiin ja samalla parempaan ihmisten liikkumisen havainnointiin sekä kuvan laadun saamiseksi ensimmäiset testit suoritettiin etsimällä optimaalista lämpökameran virkistystaajuutta ja sen kuvan päivittämisnopeutta HTML-sivulle. Kameran valmistaja tarjoaa ohjelmointikehittäjälle mahdollisuuden vaihtaa virkistystaajuutta 0,5–64 Hz, joista testattiin 1, 2, 4, 8, 16, 32 ja 64 Hz. Asetetun lämpökameran virkistystaajuuteen mukaan HTML:n kuvan päivittämisnopeus asetettiin kaksi kertaa vähemmäksi.

Koe suoritettiin liikuttamalla kättä kameran edessä samalla nopeudella ja tarkkailemalla, kuinka lämpökamera havaitsee käden liikkeen sekä paikallaan olevan lampun. Joten



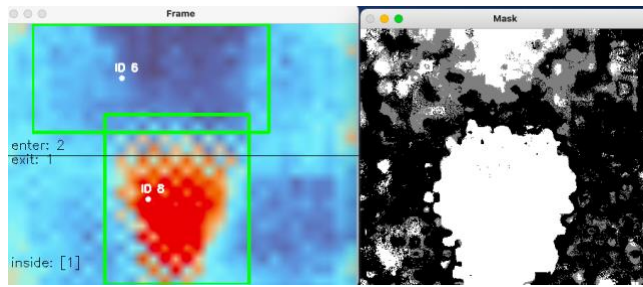
kokeen aikana havaittiin, että 32 ja 64 Hz eivät antaneet sopivaa kuvanlaatua kohinansa vuoksi. Taajuudella 1 ja 2 Hz kameran edessä liikkuvan käden jäljet eivät ole riittävästi havainnointiin. Toisaalta, 4 Hz:n jäljet näyttivät selkeämmältä, mutta liikkuva käsi antoi kuitenkin paljon vääriä jälkiä. Kuvasta 27 nähdään, että optimaalinen virkistystaajuus on joko 8 tai 16 Hz, missä muihin esimerkkeihin verrattuna, jäljet näkyvät selkeämmin, eikä ole vääriä jälkiä tai kohinaa. Siinä tapauksessa HTML-kuvan päivittämisenopeudeksi on asennettava vähintään 8 Hz, mutta ei enempää kuin 64 Hz. kuten lisäkokeet myöhemmin osoittivat. Samanaikaisesti on huomioitava nousevaa kohinatasoa virkistystaajuuksilla 32–64 Hz. Kuvassa on esitetty kaksi 32 Hz:n tapausta, joista oikeanpuoleinen ilmestyy vain hetkittäin.



Kuva 27. Liikkuvan käden havainto lämpökameran 1–64 Hz:n virkistystaajuudella.

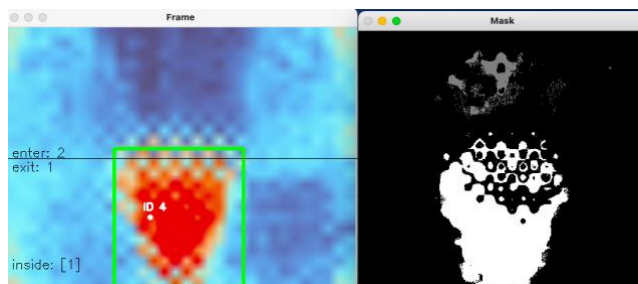
#### 4.1.2 Historia- ja kynnyksparametrien säätäminen

Kun optimaalinen virkistystaajuus oli selvitetty ja asennettu, lämpökamera on valmis havainnoimaan ihmisten liikettä. Ensin ihmisen liikkeen havainnointikoe suoritettiin vain yhdellä henkilöllä, joka astui sisään ja sitten lähti ulos. Testin tavoitteena oli säätää `createBackgroundSubtractorKNN(history, dist2Threshold)`-funktion parametreja sekä keskipisteiden välistä etäisyyden kynnyksrajaa, että saada mahdollisimman hyvät taustamallit ja sen pohjalta saada paremmat mahdollisuudet kohteiden havaitsemiseen ja seuraamiseen. Kuva 28 havainnollistaa kuvan Mask-taustamallin ja sen pohjalta väärin havaitun kohteen, vaikka testiin osallistui vain yksi henkilö.



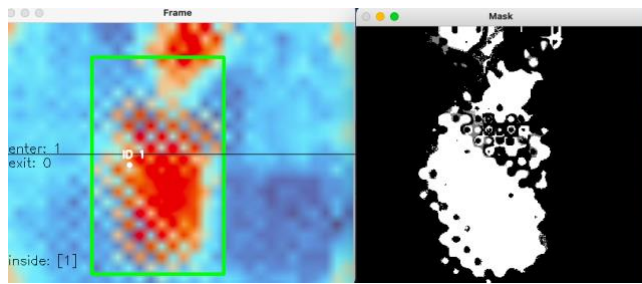
Kuva 28. Lämpökameran kuva ja sen taustamalli oletusparametreilla.

Kuvasta 28 havaitaan, että laitteen oletusparametreilla se ei tunnistanut luotettavasti edes yhtä henkilöä. Jouduimme aluksi säätämään laitteen parametreja, jotta edes tämä yksikertaisin koetilanne havaittiin oikein ja luotettavasti. Saimme kokeellisesti selvitettyä, että saimme parhaan tuloksen säätämällä history- ja dist2Threshold-parametreja. Kuvassa 29 on esitetty em. parametrien säätöjen jälkeen oikein havaittu kohde ja taustamalli.



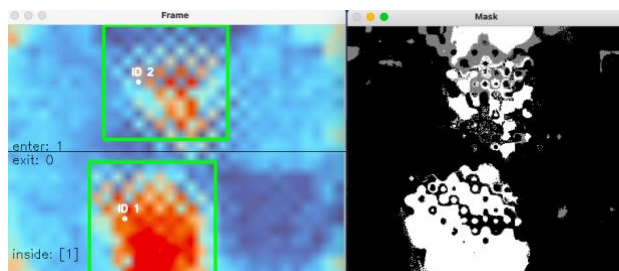
Kuva 29. Lämpökameran kuva ja sen taustamalli säädetyillä parametreilla.

Seuravaksi testattiin, kuinka lämpökamera havainnoi kahta ihmistä lähellä toisia. Kuvasta 30 näkyy, että lämpökamera ei havainnut tätäkään tilannetta aluksi oikein. Toinen henkilö jää huomaamatta, vaikka ihminen selvästi havaitsee kaksi eri kohdetta. Tässä tilanteessa ohjelma tunnisti kaksi henkilöä yhdeksi kohteeksi, jonka se on nimennyt tunnuksella ID1.



Kuva 30. Kaksi henkilöä vierekkäin lämpökameran kuvassa.

Tämä ongelma saatiin korjattua pienentämällä kohteiden keskipisteiden etäisyyden kynnyksrajaa siten, että ohjelma rekisteröi kohteen kahdeksi eri objektiksi. Kuva 31 havainnollistaa kynnyksrajan vaikutusta, kuvassa kynnyksraja on pienennetty arvoon 40. Kuvasta havaitaan, että lämpökamera tunnistaa nyt kaksi eri henkilöä.

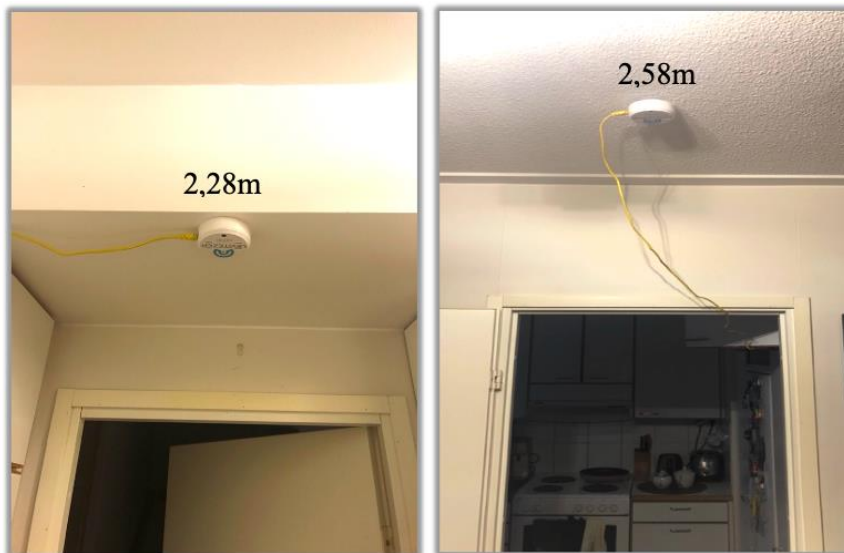


Kuva 31. Kaksi henkilöä vierekkäin keskipisteiden etäisyyden säätämisen jälkeen.

## 4.2 Kokeet kahteen eri korkeuteen asennetulla lämpökameralla

### 4.2.1 Kokeet 2,28 metrin korkeudella asennetulla lämpökameralla

Kameran korkeuden vaikutusta ohjelmatuloksiin testattiin ensin lämpökameralla, joka oli ripustettu 2,28 ja sitten 2,58 m:n korkeuteen. Kuvassa 32 on esitetty asennettu LVZ-R1-laite kattoon 2,28 ja 2,58 m:n korkeuteen.



Kuva 32. LVZ-R1 laite asennettuna 2,28 ja 2,58 metrin korkeuteen.

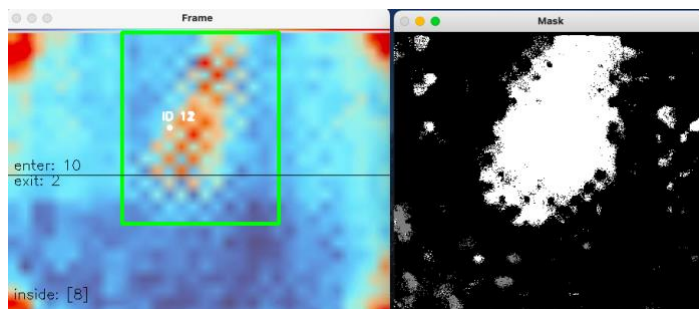
Tulevissa kaikissa kokeissa simuloidaan tilanne, kun 10 henkilöä tuli tilaan sisään ja sen jälkeen 2 poistuivat. Kokeen osallistujat olivat 1,84 m ja 1,64 m pitkiä. Ensimmäisessä kokeessa kaikki henkilöt kävelivät yksitellen. Toisessa kokeessa kaksi paria käveli vierekkäin. Kolmannessa ja neljännessä kokeissa tilanne toistui, mutta lämpökamera asennettiin korkeammalle. Tavoitteena oli seurata, pystyykö ohjelma havaitsemaan, seuraamaan ja laskemaan ihmisiä oikein.

Testien taulukoiden rastit osoittavat, että henkilö havaittiin ja jäljitettiin oikein, kun taas viivat osoittavat, että henkilö jäi havaitsematta. Mahdollisimman luotettavien tulosten saamiseksi jouduttiin taas säätämään tarvittavia asetuksia. Tärkeimmät asetukset olivat taustamallin historia, kohteiden keskipisteiden etäisyyden kynnyksäraja, sekä kohteen pienin nollasta poikkeavien pikselien määrä. Taulukossa 3 on esitetty saadut tulokset ensimmäisen kokeen jälkeen, jotka osoittavat ohjelman ihmisten määrän laskemisen oikein, eli yhteensä oli havaittu 12 henkilöä, joista 10 menivät sisään, 2 ulos ja tilaan jää 8.

Taulukko 3. Ripustetun 2,28 metrin korkeuteen lämpökameran kokeen 1 tulokset.

ID-tunnistus	1	2	3	4	5	6	7	8	9	10	11	12
Sisään	x	x	x	x	x	x	x	x	x	x		
Ulos											x	x
Sisällä laskettu / todellinen	8 / 8											
Yhteensä havaitut	12											

Ohjelma laski ihmisten määrän. Sisälle jää 8 henkilöä sen jälkeen, kun 10 tuli sisään ja 2 poistui yksitellen. Kuvasta 33 näkyy, että 10 henkilön laskeminen näyttää onnistuneen, koska sisään tulevien ja poistuvien sekä sisällä oleskelevien määrä on oikea



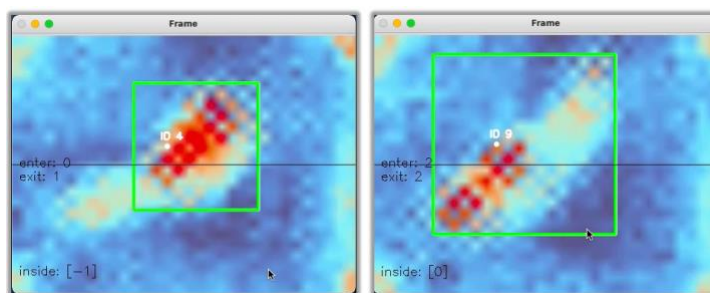
Kuva 33. Ohjelma laski ihmisten määrän. Sisälle jää 8 henkilöä sen jälkeen, kun 10 tuli sisään ja 2 poistui yksitellen.

Yllä olevassa testissä ihmiset pitivät noin puolen metrin etäisyyden toisistaan. Taulukossa 4 näkyy, miten ohjelma laski ihmisten määrä, kun 6 henkilöä meni sisään ja sitten 4 tuli ulos, joista 10:stä oli havaittu vain 8 henkilöä. Henkilöt ID3 ja ID4-tunnisteilla ja ID9 ja ID10-tunnisteilla kävelivät noin 40 cm:n etäisyydellä toisistaan.

Taulukko 4. Ripustetun 2,28 metrin korkeuteen lämpökameran kokeen 2 tulokset.

ID-tunnistus	1	2	3	4	5	6	7	8	9	10
Sisään	x	x	–	x	x	x				
Ulos							x	x	x	–
Sisällä laskettu / todellinen	1 / 2									
Yhteensä havaitut	8									

Taulukosta 4 ja kuvasta 34 nähdään, että objektien välinen pieni etäisyys johtaa siihen, että ohjelma lukee kaksi henkilöä yhdellä tunnuksella, kuten tapahtuu parien kanssa ID3-, ID4- ja ID9- ja ID10-tapauksissa. Kuten aikaisemmin oli mainittu, kohteiden keskipisteiden etäisyys määrittää, kuuluko havaittu kohde samaan kohteen vai pidetäänkö sitä uutena. Tästä johtuen voidaan olettaa, että säätämällä ohjelmallisesti keskipisteiden välistä etäisyyttä voidaan välttää tällaisia virheitä. Toisaalta lämpökameran matala asento voi vaikuttaa kohteiden näyttämiseen kuvassa siten, että ohjelma lukee yhdistyneet toisiaan suuret infrapunatäplät yhtenä kohteena.

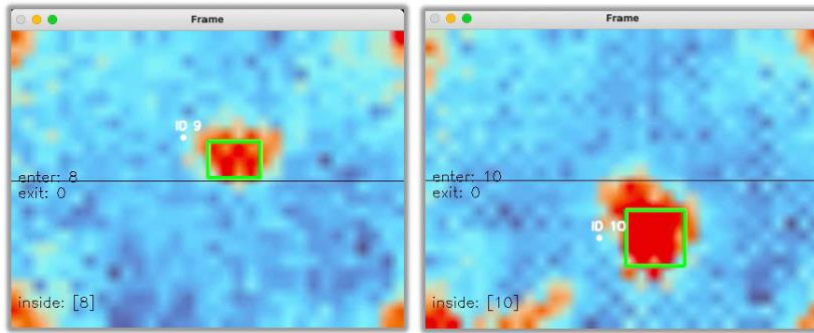


Kuva 34. Lämpökameran väärä käsitys kahdesta vierekkäisistä kohteista.

#### 4.2.2 Kokeet 2.58 metrin korkeudella asennetulla lämpökameralla

Kameran korkeuden vaikutusta ohjelmatuloksiin testattiin sitten lämpökameralla, joka oli ripustettu 2,58 metrin korkeuteen. Kuvasta 35 voidaan nähdä, että kohteista jääneet infrapunatäplien koot ovat pienempiä. Samalla kameran tunnistusalue on isompi.

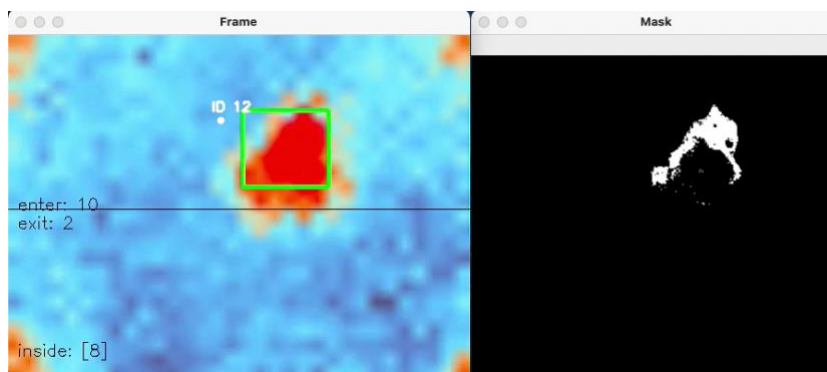
Vasemmassa kuvassa henkilön pituus on 1,64 m ja oikealla 1,84 m.



Kuva 35. Infrapunakuvat kamerasta ripustettuna 2,58 metrin korkeuteen.

Seuraavaksi pidettiin kolmas samantyyppinen koe, kuten matalan kameran sijainnin tapauksessa, eli simuloitiin 10 henkilön menemistä tilaan ja sitten 2 henkilöä tuli ulos yksi kerrallaan. Mutta tällä kertaa käytettiin taustamallin rakentamiseen toista metodia nimeltään MoG, sillä se toi paremman taustamallin ja tulokset. MoG:n algoritmissa jokainen taustapikseli sijoitetaan matriisiin ja sekoitetaan Gaussin jakauman avulla. Jokainen väri saa arvon, joka edustaa aikaa, jonka ne viipyvät kohtauksessa. Tällä tavalla staattisina pysyviä värejä käytetään taustan määrittämiseen. (34.)

Vastaavasti kuin KNN-funktiossa jouduttiin nyt säätämään MoG-algoritmin asetuksia. Tärkeimmät asetukset olivat taustamallin historia, kohteiden keskipisteiden etäisyyden kynnyksäraja sekä kohteen pienin nollasta poikkeavien pikselien määrä. Kuva 36 esittää kokeen tuloksen sekä Mask-taustamallin, joka tällä kertaa mallinnettiin käyttäen MoG-menetelmää.



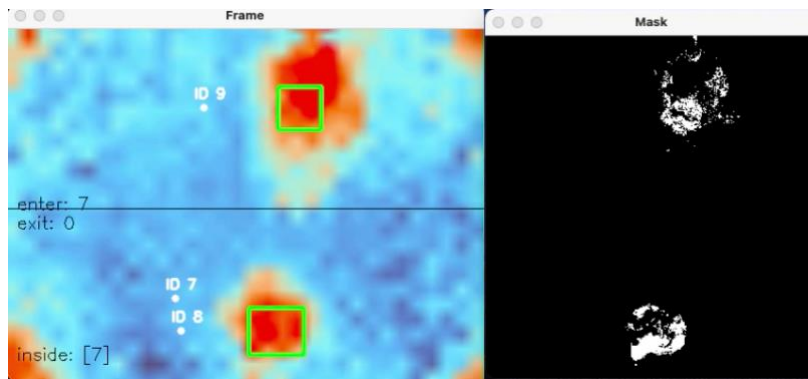
Kuva 36. Ohjelma laski ihmisten määrän. Sisällä jaa 8 henkilöä, kun 10 tuli sisään ja 2 poistui yksi kerrallaan.

Taulukosta 5 nähdään, että samoin kun kameran matalan sijainnin tapauksessa, koe meni onnistuneesti ja kaikki henkilöt havaittiin sekä laskettiin oikein.

Taulukko 5. Ripustetun 2,58 metrin korkeuteen lämpökameran kokeen 3 tulokset.

ID-tunnistus	1	2	3	4	5	6	7	8	9	10	11	12
Sisään	x	x	x	x	x	x	x	x	x	x		
Ulos											x	x
Sisällä laskettu / todellinen	8 / 8											
Yhteensä havaitut	12											

Neljännessä kokeessa simuloitiin tilannetta, jossa 10 ihmistä tuli tilaan ja sitten 2 poistui, mutta kaksi paria käveli vierekkäin. Kuva 37 kuvastaa vierekkäin käveleviä henkilöitä lämpökameran alla.



Kuva 37. Käveleviä vierekkäin henkilöitä lämpökameran alla.

Taulukosta 6 nähdään, että 2,59 metrin ripustettu kamera antoi paremmat tulokset kuin matalan kameran sijainti, mutta kuitenkin yhdessä tapauksessa ID8-kohdetta ei ollut laskettu oikein. Kaikki mahdolliset ohjelman parametrien säätäminen sekä taustamallin muodostumistavan muuttuminen eivät parantaneet tilannetta. Esimerkiksi kuvassa 35 on esitetty, kuinka ID8-objekti liikkuessa alaspäin, ylitti keskiviivan, mutta objektin



selkeästä muodosta kuvassa huomaamatta, ohjelma jätti sen huomioimatta. Kyseisessä esimerkissä nähdään, että henkilöitä käsitellään yhtenä esineenä. Ohjelma sekoitti sen olemattomaan ID7-esineeseen lämpökameran ympärillään syntyneen kohinan vaikutuksen vuoksi. Tämän seurauksena kuvan alla oleva olematon ID7-objekti ei ylittänyt keskiviivaa, eikä ohjelma siten voinut laskea sitä. Tilannetta voi korjata rajaamalla lämpökameran tunnistusalueetta ohjelmallisesti, mutta samalla se saattaa jättää joitain esineitä huomaamatta.

Taulukko 6. Ripustetun 2,58 metrin korkeuteen lämpökameran kokeen 4 tulokset.

ID-tunnistus	1	2	3	4	5	6	7	8	9	10	11	12
Sisään	x	x	x	x	x	x	x	–	x	x		
Ulos											x	x
Sisällä laskettu / todellinen	7 / 8											
Yhteensä havaitut	11											

#### 4.3 Tulosten analysointi ja käyttökelpoisuus

Tehtyjen lämpökameran tulosten analysoinnin jälkeen todettiin, että tärkeimmät tulosten laatuun vaikuttavat tekijät ovat kamerasijaintikorkeus ja sen mukaan ohjelmassa säädetyt asetukset. Näitä asetuksia ovat taustamallin luontitapa, kohteiden keskipisteiden välisen etäisyyden säätäminen, havaintoalueen määrittäminen ja MLX90640-lämpökameran aiheuttaman kohinan vaikutus kuvan alueella, erityisesti kuvan reunoilla. Projektin testeissä käytetty KNN-luontimenetelmä loi selkeän objektien tunnistusmallin, mutta aiheutti paljon kohinaa taustamallissa. Toisena tekniikkana käyttämämme MoG-menetelmä antoi selkeän taustamallin, mutta kohteiden tunnistaminen toimi huonommin.

Tavalliseen kameraan verrattuna lämpökamera on välttämätön kalu ihmisten havaitsemiseksi huonoissa olosuhteissa, kun näkyvyys tai valaistus ei ole riittävän hyvää.

Huoneen oven ja lämpökameran sijainnista riippuen ohjelman hienosäätö on tarpeen. Suoritetut kokeet osoittivat, että lämpökameralla ohjelma pystyi seuraamaan ihmisten liikkumista kohtalaisen hyvin, varsinkin kun ihmiset tulevat tilaan yksi kerrallaan. Ongelmat syntyivät parina kulkevien ihmisten tunnistamisessa. OpenCV-kirjaston työkaluilla tehdyt muutokset ja säädöt lisäsivät tulosten luotettavuutta, mutta eivät onnistuneet poistamaan kaikkia syntyneitä virheitä. Tämä saattaa johtua 32x24-kameran alhaisesta resoluutiosta, jolla kehitetty ohjelma ei pystynyt erottamaan toistensa lähellä käveleviä ihmisiä riittävän tarkasti. Testeissä käytettiin myös 40 cm etäisyyttä, mikä on todella pieni arvo. Tällöin laitteisto ei tunnista täysin luotettavasti kaikkia henkilöitä. Käytännössä ihmiset harvoin kävelevät työympäristössä näin lähellä toisiaan. Jos ihmiset kävelevät tyypillisellä etäisyydellä toisistaan, voidaan todeta lämpökameraan havaitsevan kohteet melko luotettavasti.

## 5 Yhteenveto

Alun perin projektin tavoitteena oli kehittää sulautettu järjestelmä LVZ-R1 laitteen pohjalla. Tarkoituksena oli luoda sulautettu järjestelmä, joka pystyisi seuraamaan huoneessa olevien ihmisten määrää sekä hiilidioksidipitoisuutta. Koska laitteeseen sisäänrakennettu infrapunakamera pystyy havaitsemaan ihmisten liikkeitä, tavoitteena oli analysoida saatua dataa ja laskea ihmisten määrää. Näiden tietojen perusteella voisimme ennustaa hiilidioksiditason nousua ja tietyn kynnyksen ylittyessä ohjata huoneen sisäistä ilmanvaihtojärjestelmää lisäämään tai vähentämään sen tehoa.

Työn kehityksen aikana havaittiin, että työmäärä tulee todella laajaksi. Joten päätettiin jättää hiilidioksidin mittauksen tämän työn ulkopuolelle. Tämän ominaisuus voidaan toteuttaa jatkokehityksen yhteydessä.

Työssä tutkittiin lämpökameran MLX90640 toimintaperiaatetta ja kehitettiin ohjelmisto, joka esittää lämpökamerasta saadun tiedon värikuvana Web-rajapinnan kautta. Lämpökameran kuva loi pohjan toisen ohjelman kehittämiseksi, joka pystyy

havaitsemaan ihmisiä, seuraamaan heidän liikettä ja havaitsemaan tuleeko ihminen tilaan vai lähtekö hän pois. Kehitetty järjestelmä toimi IoT-laitteena, joka tuottaa dataa palvelimeen ja sieltä asiakkaan puolelle, jossa sitä analysoidaan. Kehitetty järjestelmä antaa asiakkaalle melko luotettavasti tietoa ihmisten lukumäärästä huoneessa.

Alun perin tavoitteena oli kehittää ohjelmisto, joka toimisi LVZ-R1-laitteen sisällä. Päätettiin projektin aikana, että tässä työssä kehitetään järjestelmä, jossa data analysoidaan tietokoneella. Tämä johtui siitä, että resursseja vaativa OpenCV-kirjasto valittiin varsinaiseksi lämpökuvaa-analyysin työkaluksi ja kaikki kirjaston asennusyritykset wEsp32-alustan ohjelmointiympäristöön olivat epäonnistuneet. Jatkokehityksessä tavoitteena on toteuttaa datan analysointi LVZ-R1 laiteen sisällä. Lämpökameran etujen ohella, jatkokehityksessä on tarpeen ottaa huomioon ne lämpökameran puutteet, jotka havaittiin tämän työn aikana. Esimerkiksi vakavia ongelmia olivat kamerassa syntynyt kohina sekä kameran matala resoluutio, mikä sai ohjelman havaitsemaan kaksi ihmistä yhdeksi esineeksi heidän ollessaan lähellä toisiaan.

Kehittämämme lämpökameran tulosten analysointi mahdollistaa paitsi ihmisten jäljittämisen ja laskemisen, myös muiden lämpöä säteilevien esineiden ja laitteiden havaitsemisen huoneissa ja teollisuuden tiloissa. Näin ollen samaa lähestymistapaa voidaan käyttää moniin muihin tarkoituksiin, kuten työpaikan tulipalon havaitsemiseen, laitteiden lämpötilojen tarkkailuun tai havaitsemaan ruuhkatilanteita. Taloudellisen merkityksen kannalta, alhaisen virrankulutuksen, laajojen teknisten ominaisuuksien, pienen kokoisena ja samalla pienen hinnan vuoksi MLX90640-lämpökamera on houkutteleva työkalu sulautetun järjestelmän rakentamiseen.

Insinööriyöprosessin aikana vahvistimme taitoa lukea koodia, sekä ymmärtää sen rakennetta ja tutustua nopeasti ennestään tuntemattomaan ohjelmistokehykseen. Muiden ohjelmointikielten rakenteiden ja metodien osaaminen helpottivat projektimme aloittamista nopeasti uudessa ympäristössä, järjestelmällä ja laitteistolla. Projektin aikana tutustuttiin markkinoilla oleviin erilaisiin ilmanlaatumittareihin, jotka sisälsivät hiilidioksidimittarin ja infrapunasensorin. Selvitimme niiden vahvat ja heikot puolet. Tutkittiin myös, mikä on hyvä ilmanlaatu työympäristössä ja miten sitä säädetään lailla. Seuraavana tutkittiin LVZ-R1:n rakennetta ja sen sisäänrakennettuja komponentteja

sekä niiden ohjelmistoja ja toimintaperiaatteita. Laitteiden tutkimukseen käytettiin aktiivisesti komponenttivalmistajien verkkosivustoja ja muiden kehittäjien esimerkkejä. Kun laitetta ohjaava ohjelmisto on alun perin kehitetty C-kielellä, opiskeltiin C-ohjelmointikieltä ja mikrokontrollerin FreeRtos-käyttöjärjestelmän toimintaperiaatetta. Näin saimme selvitettyä LVZ-R1 laitteen toiminnan periaatteet. Tutustuimme, kuinka ohjelmoidaan mikrokontrolleria tietokoneen avulla ja mitä ohjelmointiympäristöä käytetään. Tutkimuksessani käytettiin suomen- ja englanninkielisiä lähteitä eri verkkosivustoilta ja tieteellisistä kirjoista. Työssä tutustuimme erilaisiin tietokonenäkö- ja koneoppimisteknikoihin käyttäen sekä C-, että Python-ohjelmointikieltä.

## Lähteet

- 1 Työturvallisuuslaki 23.8.2002/738. 2 luku Työnantajan yleiset velvollisuudet. 2002. Verkkoaineisto. Finlex Internet-palvelu. <[www.finlex.fi/fi/laki/ajantasa/2002/20020738#a738-2002](http://www.finlex.fi/fi/laki/ajantasa/2002/20020738#a738-2002)>. Luettu 14.2.2021.
- 2 Ilmanvaihdon perusteet. 2008. Verkkoaineisto. Sisäilmayhdistys ry. <[www.sisailmayhdistys.fi/Perustietoa-sisailmasta/Ilmanvaihdon-perusteet](http://www.sisailmayhdistys.fi/Perustietoa-sisailmasta/Ilmanvaihdon-perusteet)>. Luettu 25.8.2021
- 3 HTP-arvot 2002. Työsuojelusäädöksiä 3. Sosiaali- ja terveysministeriö, Työsuojeluosasto, Kemian työsuojeluneuvottelukunta. Tampere 2002.
- 4 Perustietoa sisäilmasta. Verkkoaineisto. Sisäilmayhdistys ry. <[www.sisailmayhdistys.fi/Terveelliset-tilat/Sisailmasto/Kemialliset-epapuhautudet](http://www.sisailmayhdistys.fi/Terveelliset-tilat/Sisailmasto/Kemialliset-epapuhautudet)>. Luettu 25.08.2021
- 5 LeViteZer. 2021. Verkkoaineisto. LeViteZer Oy. <<http://www.levi-tezer.com>>. Luettu 25.8.2021.
- 6 Sandberg, Esa. 2014. Sisäilmasto ja ilmastointijärjestelmät. Ilmastointiteknikka, Osa 2. Helsinki: Talotekniikka-Julkaisut Oy.
- 7 Seppänen, Olli. 2004. Ilmastoinnin suunnittelu. Helsinki: Talotekniikka-Julkaisut Oy.
- 8 RT 07-11299, Sisäilmastoluokitus 2018. Sisäympäristön tavoitearvot, suunnitteluohjeet ja tuotevaatimukset. 2018. Helsinki: Rakennustieto Oy
- 9 Ympäristöministeriön asetus uuden rakennuksen sisäilmastosta ja ilmanvaihdosta. 2017. 1009/2017
- 10 Sosiaali- ja terveysministeriön asetus asunnon ja muun oleskelutilan terveydellisistä olosuhteista sekä ulkopuolisten asiantuntijoiden pätevyysvaatimuksista. 2015. 545/2015.

- 11 Infrared Sensor Technology and its Impact on HVAC CO2 Measurement Accuracy. 2013. Verkkoaineisto. Vaisala.  
<[www.vaisala.com/sites/default/files/documents/CEN-TIA-HVAC-CO2-Measurement-Accuracy-B211311EN-A.pdf](http://www.vaisala.com/sites/default/files/documents/CEN-TIA-HVAC-CO2-Measurement-Accuracy-B211311EN-A.pdf)>. Luettu 7.9.2021
- 12 Telaire T6713 | Miniature CO2 Sensor Module - Datasheet. 2021. Verkkoaineisto. Amphenol Advanced Sensors. <[www.amphenol-sensors.com/en/telaire/co2/525-co2-sensor-modules/3399-t6713](http://www.amphenol-sensors.com/en/telaire/co2/525-co2-sensor-modules/3399-t6713)>. Luettu 9.9.2021
- 13 Miten NDIR CO2 toimii? 2021. Verkkoaineisto. CO2Meter.com.  
<[www.co2meter.com/blogs/news/6010192-how-does-an-ndir-co2-sensor-work](http://www.co2meter.com/blogs/news/6010192-how-does-an-ndir-co2-sensor-work)>. Luettu 8.9.2021.
- 14 Beerin ja Lambertin laki. 2016. Verkkoaineisto. Wikipedia.  
<[https://fi.wikipedia.org/wiki/Beerin\\_ja\\_Lambertin\\_laki#Lahteet](https://fi.wikipedia.org/wiki/Beerin_ja_Lambertin_laki#Lahteet)>. Paivitetty 3.4.2016. Luettu 9.9.2021.
- 15 Luciana B. Mendes ym. 2015. NDIR Gas Sensor for Spatial Monitoring of Carbon Dioxide Concentrations in Naturally Ventilated Livestock Buildings. Verkkoaineisto. <[www.mdpi.com/journal/sensors](http://www.mdpi.com/journal/sensors)>. Luettu 9.9.2021
- 16 Valo ja Spektri. 2003. Verkkoaineisto. Ilmatieteen laitos. <<https://space.fmi.fi/oppimateriaali/envisat/valonsade/spektri.html>>. Luettu 14.9.2021.
- 17 Infrared. 2021. Verkkoaineisto. Wikipedia.  
<<https://en.wikipedia.org/wiki/Infrared>>. Luettu 14.9.2021.
- 18 Infrared Temperature Measurement Theory and Application. 2018. Verkkoaineisto. Omega Engineering Inc. <<https://www.omega.com/en-us/resources/infrared-temperature-measurement-theory-application>>. Luettu 15.9.2021.
- 19 Lamposateily & Infrapuna. 2019. Verkkoaineisto. Infradex Oy.  
<[www.infradex.com/lamposateily-ja-lampokamera/](http://www.infradex.com/lamposateily-ja-lampokamera/)>. Luettu 13.9.2021.

- 20 16 Lämpökuvauksen teoria. Publ. No. 1 557 713 Rev. A – FINNISH (FI). 2003. Verkkoaineisto. Infradex Oy. <<https://www.infradex.com/wp-content/uploads/2016/08/teoria.pdf>>. Luettu 13.9.2021.
- 21 18 Emissiivisyystaulukot. Publ. No. 1 557 713 Rev. A – FINNISH (FI). 2003. Verkkoaineisto. Infradex Oy. <<https://www.infradex.com/wp-content/uploads/2016/08/emissiivisyys.pdf>>. Luettu 13.9.2021.
- 22 Toshiyo Tamura ym. 2018. Current Developments in Wearable Thermometers. 2018. Verkkoaineisto. Article in Advanced Biomedical Engineering. <[www.researchgate.net/publication/324463908\\_Current\\_Developments\\_in\\_Wearable\\_Thermometers](http://www.researchgate.net/publication/324463908_Current_Developments_in_Wearable_Thermometers)>. Luettu. 20.9.2021.
- 23 Melexis Far Infrared Thermal Sensor MLX90640. 2019. Verkkoaineisto. SYSTEM PLUS CONSULTING. <[www.systemplus.fr/reverse-costing-reports/melexis-far-infrared-thermal-sensor-mlx90640/](http://www.systemplus.fr/reverse-costing-reports/melexis-far-infrared-thermal-sensor-mlx90640/)>. Luettu 20.9.2021.
- 24 Barr, Michael. Massa Anthony. 2006. Programming Embedded Systems, 2nd Edition. Sebastopol: O'Reilly Media, Inc.
- 25 Tietotekniikan termitalkoot. 2017. Verkkoaineisto. Sanastokeskus ry. <<http://www.tsk.fi/tsk/termitalkoot/hakemistot-267.html>>. Luettu 3.10.2021
- 26 Ravulavaru, Arvind. 2018. Enterprise Internet of Things Handbook. Birmingham: Packt Publishing Ltd.
- 27 ESP Product Selector. Product Comparison. 2020. Verkkoaineisto. Espressif Inc.<<https://products.espressif.com/#/product-comparison>>. Luettu 8.10.2021
- 28 wEsp32 Product Brief. 2021. Verkkoaineisto. Silicognition LLC. <<https://wesp32.com>>. Luettu 9.10.2021.
- 29 Sound Detector Hookup Guide.2014. Verkkoaineisto. Digi-Key Electronics. <<https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/SoundDetectorHookupGuide.pdf>>. Luettu 24.5.2021.

- 30 FreeRTOS Documentation. 2016. Verkkoaineisto. FreeRTOS Real-time operating system for microcontrollers. [https://www.freertos.org/fr-content-src/uploads/2018/07/161204\\_Mastering\\_the\\_FreeRTOS\\_Real\\_Time\\_Kernel-A\\_Hands-On\\_Tutorial\\_Guide.pdf](https://www.freertos.org/fr-content-src/uploads/2018/07/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf). Luettu 15.6.2021.
- 31 Datasheet for MLX90640. 2021. Verkkoaineisto. Melexis. <https://www.melexis.com/en/product/MLX90640/Far-Infrared-Thermal-Sensor-Array>. Luettu 11.7.2021.
- 32 Canu, Sergio. 2021. Object Tracking with Opencv and Python. Verkkoaineisto. Pysource LTD. < <https://pysource.com/2021/01/28/object-tracking-with-opencv-and-python/>>. Luettu 14.10.2021.
- 33 Rosebrock, Adrian. 2018. OpenCV People Counter. Verkkoaineisto. PyimageSearch. < <https://www.pyimagesearch.com/2018/08/13/opencv-people-counter/>>. Luettu 14.10.2021.
- 34 Gabriel, Garrido; Prateek, Joshi. 2018. OpenCV 3.x with Python By Example - Second Edition. E-kirja. Packt Publishing.