

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia Teaduskond

Arvutisüsteemide instituut

Rainer Liis 164039IACB

**RUUMI SÜSTEEMIDE AUTOMATISEERIMINE**

**SENSORVÕRGU ABIL**

Bakalaureusetöö

**Juhendaja**

Jürgen Soom

MSc

Tallinn 2022

# **Autorideklaratsioon**

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

Autor: Rainer Liis

[03.01.2022]

# Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks oli arendada ruumi ventilatsiooni ning valgustuse automatiseerimist koos sensorvõrguga. Töö keskendub töötava prototüübi arendusele, mille käigus kõigepealt uuriti olemasolevaid lahendusi ning pandi paika parameetrid, mis keskkonnas süsteem tööle hakkab. Töö käigus loodi kaks andurit: inimese kohaloleku tuvastamiseks ning süsihappegaasi koguse õhus tuvastamiseks. Andurid on loodud arendusplaadi ESP32 põhjal, mis valiti välja heade omaduste, hinna ning tarkvara arendust toetavat SDK pärast. Lisaks kuulub süsteemi ka keskjaam, mille kaudu saab juhtida ja kontrollida kõiki alamsüsteeme ja andureid. Välja arendatud süsteem mõõdab süsihappegaasi taset ruumis ning vajadusel käivitab ventilaatori. Temperatuurianduriga tuvastatakse inimese kohalolu ruumis, mille põhjal juhitakse toa valgustust. Sensorite ja keskjaama vaheline suhtlus on korraldatud keskjaamaga üle WiFi kasutades MQTT protokollid.

Lõputöö on kirjutatud eest keeles ning sisaldab teksti 28 leheküljel, 7 peatükki, 13 joonist, 8 tabelit.

# **Abstract**

## **Room Systems Automation**

The aim of this thesis was to automate lighting and ventilation systems in a room. During the creation process, existing options were looked at and parameters were set in place for the environment of the system. Information for the automation is gathered through 2 sensors: a presence detector and a carbon dioxide sensor. The sensors were created based on the ESP32 development board, which was chosen for its good properties and support of an SDK, which aided with the development. In addition to the sensors a central hub exists, through which all the systems are managed and controlled. The developed system measures carbon dioxide levels in the air and if needed, turns on a ventilator. Human presence in the room is also detected and with this data the lamp is operated. Communications within a sensor is done via the I2C bus and communication between the sensors and central hub is done over WiFi using the MQTT protocol.

The thesis is in Estonian and contains 28 pages of text, 7 chapters, 13 figures, 8 tables.

# Lühendite ja mõistete sõnastik

ACK	<i>Acknowledge</i> , kinnitusbitt
Broker	Sõnumite vahendaja MQTT protokollis
CPU	<i>Central Processing Unit</i> , protsessor
Docker	Virtualiseerimistarkvara konteinerite jaoks
HTTP	<i>Hypertext Transfer Protocol</i> , hüpertexti edastamisprotokoll
GPU	<i>Graphics Processing Unit</i> , graafikaprotsessor
I2C	<i>Inter-Integrated Circuit</i> , kahejuhtmeliides
MCU	<i>Microcontroller unit</i> , mikrokontroller
MQTT	<i>Message Queuing Telemetry Transport</i> , sõnumiedastusprotokoll
NDIR	<i>Non-dispersive infrared sensor</i> , mittehajutav infrapunaseensor
P2P	<i>Peer-to-peer</i> , masinatevaheline suhtlus
PIR	<i>Pyroelectric(passive) infrared sensor</i> , püroelektriline(passiivne) infrapunaseensor
PPM	<i>Parts per million</i> , osakeste hulk miljoni kohta
PWM	<i>Pulse Width Modulation</i> , pulsilaiusmodulatsioon
SBC	<i>Single Board Computer</i> , ühe plaadi arvuti
SPI	<i>Serial peripheral interface</i> , järjestikune perifeeriasiin
SCADA	<i>Supervisory Control and Data Acquisition</i> , tehniliste protsesside jälgimise ja juhtimise süsteem
SCL	<i>Serial Clock</i> , taktsignaali
SDA	<i>Serial Data Signal</i> , andmesiin
RAM	<i>Random Access Memory</i> , muutmälu
UART	<i>Universal Asynchronous Receiver-Transmitter</i> , universaalne asünkroonne saatja-vastuvõtja
Wi-Fi	<i>Wireless Networking Technology</i> , traadita kohtvõrk

# Sisukord

<b>1</b>	<b>Sissejuhatus</b>	<b>1</b>
<b>2</b>	<b>Andurvõrgu olulisus</b>	<b>2</b>
2.1	CO2 olulisus . . . . .	2
2.2	Valgustuse olulisus . . . . .	3
2.3	Võimalikud lahendused . . . . .	3
2.3.1	Toa automaatiseerimise platvormide võrdlus . . . . .	4
2.4	Töötingimused . . . . .	5
2.5	Loodav süsteem . . . . .	6
<b>3</b>	<b>Elektroonika</b>	<b>7</b>
3.1	Arendusplaadi valik . . . . .	7
3.2	Temperatuuri sensor . . . . .	8
3.2.1	Anduri valik . . . . .	8
3.3	Co2 sensor . . . . .	10
3.3.1	Anduri valik . . . . .	10
3.3.2	Kalibreerimine . . . . .	11
3.4	Keskjaam . . . . .	11
<b>4</b>	<b>Tarkvara</b>	<b>14</b>
4.1	Süsteemi algoritm . . . . .	14
4.2	Võrdlus juhtmevabade andmeedastusprotokollide vahel . . . . .	16
4.3	MQTT . . . . .	17
4.3.1	Võrdlus riistvaraliste andmeedastusprotokollide vahel . . . . .	18
4.4	I2C . . . . .	19
4.4.1	Kohaloleku sensor . . . . .	20
4.4.2	Süsihappegaasi sensor . . . . .	21
4.5	Keskjaam . . . . .	21
4.5.1	Kasutajaliides . . . . .	22
4.5.2	Loodavad sensorid . . . . .	23
4.5.3	Automatiseerimine . . . . .	23
<b>5</b>	<b>Testimine</b>	<b>25</b>
5.1	I2C testimine . . . . .	25
5.2	MQTT testimine . . . . .	25
5.3	OpenHABiga suhtluse testimine . . . . .	26

5.4 Süsteemi demonstreerimine . . . . .	27
<b>6 Probleemid</b>	<b>28</b>
6.1 Inimese tuvastamine . . . . .	28
<b>7 Kokkuvõte</b>	<b>29</b>
<b>Kasutatud kirjandus</b>	<b>30</b>
<b>Lisa 1 - Koodirepositooriumid</b>	<b>33</b>
<b>Lisa 2 - Süsteemi demonstratsioon</b>	<b>34</b>

## Jooniste loetelu

1	<i>Riistvara plokk skeem</i> . . . . .	6
2	<i>Süsihappegaasi sensori algoritm</i> . . . . .	14
3	<i>Tuvastussensori algoritm</i> . . . . .	15
4	<i>Keskjaama algoritm</i> . . . . .	16
5	<i>i2c paketid[32]</i> . . . . .	19
6	<i>Süsihappegaasi sensori edastatavad baidid</i> . . . . .	21
7	<i>Semantiline mudel openHABis</i> . . . . .	22
8	<i>OpenHABi kasutajaliides</i> . . . . .	23
9	<i>Andmemaatriks otse kasutaja eest</i> . . . . .	26
10	<i>Andmemaatriks ülevalt toa lae alt</i> . . . . .	26
11	<i>MQTT kohalolekuandmed</i> . . . . .	26
12	<i>Kohaloleku sensori ekraan</i> . . . . .	27
13	<i>Süsihappegaasi ekraan</i> . . . . .	27



## Tabelite loetelu

1	<i>Süsihappegaasi kontsentratsioonide mõju inimesele . . . . .</i>	2
2	<i>Lahenduste võrdlus[4],[5],[6] . . . . .</i>	4
3	<i>Arendusplaatide võrdlus[8],[9],[10],[11],[12] . . . . .</i>	8
4	<i>Temperatuuriandurite võrdlus[14],[15] . . . . .</i>	9
5	<i>CO2 sensorite võrdlus[18],[19] . . . . .</i>	10
6	<i>Riistvaralahenduste võrdlus[21],[22],[23] . . . . .</i>	12
7	<i>MQTT ja sarnaste protokollide võrdlus[24],[25],[26] . . . . .</i>	17
8	<i>MQTT vahendajate võrdlus[29] . . . . .</i>	18

# 1. Sissejuhatus

Targa kodu süsteemid on muutumas aina populaarsemaks. Aastal 2020 hinnati globaalse turu väärtuseks 83 miljardit dollarit ning aastaks 2026 ennustatakse selle kasvu 140 miljardi dollarini[1]. Targa kodu süsteem koosneb peamiselt kolmest komponendist: sensorid, mis koguvad hetkeandmeid keskkonnast, erinevad seadmed, mille tööd juhitakse ja keskjaam, mis kogub andmeid ning juhib nende andmete põhjal seadmeid vastavalt kasutaja soovile. Sellised süsteemid aitavad muuta kodu mugavamaks, kuna vähendavad vajadust manuaalset seadmeid juhtida ning samas aitavad hoida raha kokku, juhtides süsteeme optimaalsemalt.

Käesolev pandeemiaolukord on järjest rohkem loonud vajadust töötada kodukontoris. Kui töökeskkonna sisekliima on ebameeldiv, vähendab see produktiivsust, kurnab töötajaid ning võib põhjustada ka lisaks peavaludele muid terviseprobleeme[2]. Kaks peamist põhilist aspekti keskkonna meeldivaks muutmisel on hea õhukvaliteet ja valgustus. Automaatne ventilatsiooni juhtimine tagab, et ruumis oleks piisavalt hea õhk töö tegemiseks ning inimesed ei väsiks liiga kiirelt ära. Lisaks pandeemiaolukorrale on ka energia hind kiirelt kerkimas. Seega oleks vajalik juhtida valgustust, nii et seda kasutataks vaid siis, kui seda on vaja, et hoida kokku kuludelt. Käesoleva projekti raames keskendutaksegi nende kahe süsteemi automatiseerimisele.

Ülesande teostus koosneb mitmest osast. Esimeses osas antakse teoreetiline ülevaade töös loodavate süsteemide vajadusest ning pannakse paika süsteemiparameetrid. Teises osas kirjeldatakse elektroonika komponentide valikut ning loodud süsteemi osi. Kolmandas osas antakse ülevaade kogu süsteemi tarkvaralisest poolest ning kasutatavatest protokollidest. Neljandas osas antakse ülevaade testimisprotsessist ning viiendas süsteemi puudujääkidest ja edasisest arendusest.

## 2. Andurvõrgu olulisus

### 2.1 CO2 olulisus

Kõrgendatud süsihappegaasi kontsentratsioon hingatavas õhus kontorites ja klassiruumides põhjustab inimestel töövõime ja informatsioonitöötlemise langust, suurendab peavalude hulka ning võib põhjustada tõsisemaid tervise kahjustusi. Kõrgeks süsihappegaasi tasemeks õhus loetakse kontsentratsiooni üle 1000ppm, ehk 0.1% õhust. Üldiselt on töövõime langemine 1000ppm kontsentratsiooni puhul statistiliselt märgatav ning alates 2500ppm on langus juba väga suur ja märgatav[3].

Väline õhk on süsihappegaasi keskmise kontsentratsiooniga 400ppm ja inimese hingeõhk umbes 30000ppm. Tabelis 1 on toodud välja erinevate süsihappegaasi tasemete mõjud inimese tervisele. Teades, et suur kontsentratsioon süsihappegaasi ruumis langetab töövõimet ja teeb uniseks, on märkimisväärne, et halvasti ventileeritud ruumides on selle taseme tõus ka üpris kiire. Kontoris mõõtmetega 3,5m x 4m ehk pindalaga 14m<sup>2</sup>, kus töötab üks inimene tõuseb süsihappegaasi tase 500ppm pealt 1000ppm tasemele kõigest 45 minutiga, kui ventilatsioon ei tööta. Eriti kiirelt kasvab süsihappegaasi kontsentratsioon paikades, kus koguneb palju inimesi, näiteks koosolekuruumid ja puhketoad[3].

Tabel 1. Süsihappegaasi kontsentratsioonide mõju inimesele

CO2 kontsentratsioon (ppm)	Mõju inimesele
>1000	väsimus ja töövõime langus
>2500	suur töövõime langus
>20000	hingeldamine
>100000	värinad ja minestamine
>250000	surm

## 2.2 Valgustuse olulisus

Ruumi valgustuse automaatse juhtimise peamised eesmärgid on mugavama keskkonna loomine ning energiasääst. Kui valgust juhtida automaatselt vastavalt inimese kohalolekule, saab hoida lampi töös vaid siis, kui seda reaalselt vaja on. Valgustuse automatiseerimine muudab ka keskkonda mugavamaks, vabastades inimese valguse sisse- ja välja lülitamisest. Kontorites kasutatakse peamiselt tööks arvuteid, mille ekraanid tihti peale kiirgavad tugevalt valgust. Kui ümbritsev keskkond on hämar, võivad ekraanid silmi eriti väsitada. Valgustuse juhtimine on seega väga kasulik sellises keskkonnas, et tekitada ühtlane valgustatuse tase.

## 2.3 Võimalikud lahendused

Erinevaid kodu süsteemide automatiseerimise lahendusi on turul palju. Neid on erinevate hinnaklasside juures ning igapähele on häid ja halbu omadusi. Erinevate süsteemide vahel valides on oluline kaaluda kõiki aspekte.

Kodu automatiseerimise täislahendusi pole eriti võimalik hankida, kuna alati on midagi rohkemat monitoorida ja juhtida. Küll aga on olemas üpris piiratud ja suletud süsteeme, kus sensorid ja juhtimissüsteem moodustavad suletud ökosüsteemi. Selliseid süsteeme antud projekti raames ei kaasatud valikutesse, kuna eesmärk on kasutada avatud standardeid ning kasutada erinevate tootjate pakutavaid lahendusi koos ühtse süsteemina. Selle suurim põhjus on tulevikukindlus, mida avatud süsteemid pakuvad.

Suletud ökosüsteemid on tootja meelevaldas, ning kui mingil hetkel ei soovita sellega enam jätkata, on võimalus, et automatiseerimisele investeeritud aeg ja raha kaob ilma tulemuseta. Avatud süsteemid on sellise asjade suhtes kindlamad, kuna ka arendus on jaotatud laiali ning erinevate seadmete tugi on lai. Selleks, et kõik süsteemid lakkaks töötamast, on vaja peaaegu kõikide tootjate otsust tootmine ja toetus lõpetada.

Mõned näited sellistest suletud süsteemidest on Apple HomeKit ja Samsung SmartThings. Sellised süsteemid tuginevad ainult tootja toetusel ning ei pruugi olla eriti tulevikukindlad, samas kui avatud lähtekoodiga lahendused on entusiastide poolt loodud ning juba kümnekond aastat turul olnud. Sellistel süsteemidel ei ole ka vaja, et sensorika kasutaks konkreetset protokollide suhtluseks jaamaga või ei pea see olema mingi kindla tootja oma. Avatud süsteemid lubavad siduda kokku igasuguseid erinevaid süsteeme ja neid juhtida erinevaid meetodeid kasutades.

### 2.3.1 Toa automaatiseerimise platvormide võrdlus

Tabelis 2 toodud võrdluses on 4 erinevat populaarset valikut kodu automatiseerimise lahenduse loomiseks. Peamiselt grupeeruvad need kaheks. Kinnised süsteemid konkreetse tootjaga: Samsungi SmartThings ja Apple HomeKit ning avatud lähtekoodiga: openHAB ja Home Assistant. Kinnistel süsteemidel on eelis ühtse süsteemina lihtsa ülesseadmise suhtes. Avatud platvormidel on omakorda eelis modulaarsuse näol ja rohkemate seadmete toetamise osas. Lisaks saab ka avatud platvormidega lihtsamini lisada süsteemi enda tehtud sensoreid. HomeKit seda ei toeta ja SmartThingsil sellise asja kasutamiseks on vaja lisada eraldi mooduleid.

Tabel 2. Lahenduste võrdlus[4],[5],[6]

	<b>openHAB</b>	<b>Home Assistant</b>	<b>HomeKit</b>	<b>SmartThings</b>
Teenusepakkuja	openHab	Home Assistant Core Team and Community	Apple	Samsung
Disain	modulaarne	modulaarne	kinnine	modulaarne (smart apps)
Automaatne seadmete tuvastus	jah	jah	poolautomaatne	poolautomaatne
Paigaldamise keerukus	suurem (vajab terminali kasutamise oskust)	kerge	kerge	kerge
Platvorm	kohalik riistvara	kohalik riistvara	pilveteenused	pilveteenused
Haldus	veebiliides	veebiliides	nutitelefoni rakendus	nutitelefoni rakendus
Keskjaama vajadus	jah	jah	mõni Apple seade	ei, piiratud funktsionaalsusega
Võimalus siduda ise loodud sensoreid	jah	jah	ei	ei
Lahenduse maksumus (EUR)	<100	<100	>400	>150

Mõlemad kinnised süsteemid eeldavad ka igal ajahetkel internetiühendust, kuna tegu on pilveteenusega. See aga toob sisse riskifaktori, et kui pilveteenus lakkab töötamast, nagu juhtus Amazoni pilveteenusega aastal 2020[7], siis seadmetega suhtlus ja rakendus ei tööta ja seega pole võimalik neid kasutada. Kuigi ka kohalikult töötaval riistvaral on võimalus katki minna ja töötamine lõpetada, on see ikkagi madalam tõenäosus. SmartThings töötab ka ilma keskjaamata, kuigi siis on funktsionaalsus ja suhtlusvõime seadmetega suhtlemiseks piiratud. Z-wave ja Zigbee protokollid kasutatavate seadmetega ei suuda SmartThings ilma keskjaamata suhelda. Apple HomeKit ka toetab väga piiratud hulka seadmeid ning kuigi see ei vaja ka eraldi keskseadet, eeldab see mõne Apple nutiseadme olemasolu. Nendel süsteemidel käib ka kogu süsteemihaldus mobiilirakenduse kaudu, samas kui avatud platvormidel on kasutusel lihtsalt veebiliides, mida saab muuta ja kasutada nii mobiilist kui ka arvutist. SmartThings on olemuselt modulaarne nagu ka vabavaralised variandid, kuid seal on ka osad teenused tasulised[5],[6].

Avatud platvormide openHAB ja Home Assistant on tööpõhimõttelt üpris sarnased ning erinevused esinevad pigem detailidesse laskudes. Home Assistant on kiiremini uuenev ning kiiremini uusi sensoreid toetav platvorm, samas kui openHAB on stabiilsem ning uuendused tulevad aeglasemalt. Ülessadmine on openHAB tarkvaral natuke keerulisem, võrreldes Home Assistantiga, muutes selle vähemkogenud inimestele hirmsamaks[4]. Autoril on openHABi tarkvaraga kokkupuude eelnevate projektide baasil, tänu millele on tarkvara kasutamine lihtsam.

Hinna poolest on avatud platvormid eelistatud, kuna ei eelda spetsiaalsete seadmete olemasolu. Tabelis toodud võrdluses on maksumuses arvestatud vaid keskseadme hinda, sensoreid sinna ei lisatud, kuna antud projekti raames luuakse need ise. Arvestades vaid keskjaama hindu, kerkivad suletud süsteemide maksumused mitmekordseks võrreldes avatud süsteemidega. Avatud platvormide hinnad on kehtivad eeldusel, et hangitakse eraldi seade selle tarkvara kasutamiseks, mitte ei tööta see kuskil olemasolevas masinas.

Projekti loomiseks valiti openHAB mitmel põhjusel. Esimene neist on, et tegu on avatud lähtekoodiga rakendus. Teiseks on see stabiilsem tarkvara võrreldes HomeAssistantiga ning ei vaja nii tihti uuendamist. Lisaks on ka selle süsteemi hind madal, sisaldades vaid hangitava riistvara hinda.

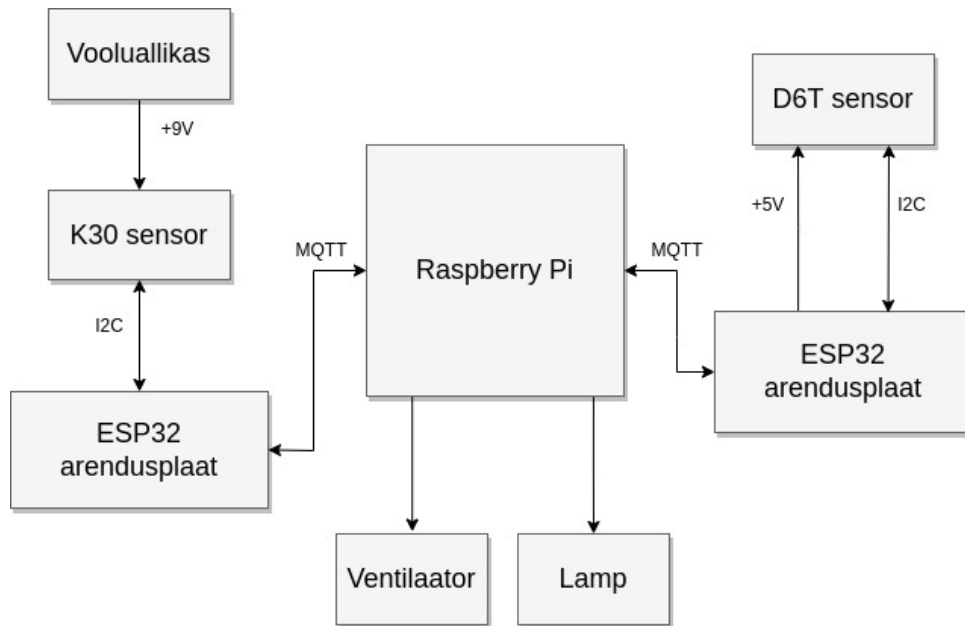
## **2.4 Töötingimused**

Loodav süsteem on kasutuses korteris, kus sensorid asuvad kodukontori ruumis ning keskjaam asub neist eraldi teises ruumis. Sensoritega juhitakse süsteeme samuti kontori-ruumis, kus asuvad valgustus ja ventilatsiooniseadmed. Kuna tegu on korteriga, ei ole seal

ekstreemseid töötingimusi nagu suured temperatuurikõikumised ning niiskustase on ka stabiilne. Seega ei pea arvesse võtma sensorite loomisel eriliste tingimustega. Ruum, kus süsteeme juhitakse on ka väike (<10m<sup>2</sup>).

## 2.5 Loodav süsteem

Joonisel 1 on kujutatud loodava süsteemi osade plokkskeemi. Sellel joonisel on näha erinevate osade omavaheline seotus.



Joonis 1. Riistvara plokkskeem

Kogu süsteem kujutab endast RaspberryPi platvormil töötavat keskjaama OpenHabi tarkvara, millega omakorda suhtlevad 2 sensorit, vastavalt inimese kohalolu ja süsihappegaasi jälgimiseks. Sensorid suhtlevad keskjaamaga üle Wi-Fi kasutades selleks MQTT protokoll. Keskjaamal on kasutusel sensoritega suhtlemiseks MQTT vahendaja Mosquitto, mis edastab sõnumeid keskjaamataravarale openHABile. OpenHAB kaudu kogutakse andmeid sensoritelt ning vastavalt muudatustele andmetes juhitakse ventilatsiooni ja valgustust.

## 3. Elektroonika

### 3.1 Arendusplaadi valik

Seoses limiteeritud ressursside ja autori kogemuste puudumise tõttu, ei toodeta antud lõputöö raames eraldi trükkplaati, vaid kasutatakse sensorite loomiseks arendusplaate. Süsteemi jaoks loodavad sensorid peavad suhtlema keskjaamaga ning samas tarbima võimalikult vähe voolu. Kuigi sensorid ja keskjaam võivad asuda üksteisele üpris lähestikku ühes ruumis, on antud töö raames kasutusjuht kus need asuvad erinevates ruumides. Sellest tulenevalt on vaja sensoritel juhtmevabalt suhelda keskjaamaga. Antud lahenduse jaoks valiti Wi-Fi, kuna selle kasutamiseks on keskjaamal vajalik riistvara olemas ning ka kohtvõrgu jaoks on ruuter olemas. See võimaldab vältida lisariistvara hankimise, mis muudaks kogu süsteemi kulukamaks.

Järgnevalt on toodud välja arendusplaatide parameetrid ning nendevaheline võrdlus. Tabel 3 on näha, et kõik valikus olevad plaadid, peale Arduino Uno sobiksid kohe kasutamiseks sensorite loomisel. Arduino plaadile tuleks veel juurde panna Wi-Fi võimekusega kiip, mis oleks aga keerulisem protsess. Teised lahendused pakuvad palju mugavamat ja kiiremat lähenemist. Kuigi sellel on ka arendustarkvara tugi, langeb Arduino Wi-Fi võimekuse puudumise tõttu konkurentsist välja. Parameetride poolest on kõige võimekam Raspberry Pi Zero W. Sellel plaadil on aga ka toimimiseks vaja paigaldada operatsioonisüsteem, läbi mille saab plaadil programme käivitada. Teistel plaatidel seda vaja pole ning saab programmi laadida otse plaadi püsimällu, kust seda käivitatakse. Lisaks on RaspberryPi'l kõige suurem volutarbimine aktiivse Wi-Fi ühenduse hoidmisel, mis annab teistele plaatidele tema ees eelise. ESP32 ja ESP8266 plaatidel on üpris sarnased omadused. Mõlemal plaadil on SDK tugi, mis teeb nende peal arenduse lihtsaks ja mugavaks. ESP32 ongi uuem versioon ESP8266 kiibist ning on võimekam. Sellel on suurem protsessori taktsagedus ning mäluhulk, mis teevad selle eelistatumaks ESP8266 ees. Hinnavahe nende kahe vahel ei ole ka eriti oluline, mistõttu see ei ole määrav punkt. Valituks osutus seega ESP32 arendusplaat selle Wi-Fi võimekuse ja piisavalt väikse volutarbimise tõttu. Lisaks mõjutas ka valikut reaalne arendusplaatide saadavus turul.



Tabel 3. Arendusplaatide võrdlus[8],[9],[10],[11],[12]

Parameetrid	ESP32	Raspberry Pi Zero W	ESP8266	Arduino Uno
Mikrokontroller	Tensilica Xtensa LX6	BCM2835	Tensilica Diamond Standard 106 Micro	ATmega328
Muutmälu	520KB	-	16KB	2KB
Välkmälu	Kuni 64MB	512MB	4KB	32KB
Wi-Fi	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	-
GPIO	34	40	17	23
Perifeeria	SPI, I2C, UART	SPI, I2C, UART	SPI, I2C, UART	SPI, I2C, UART
Aktiivse Wi-Fi ühenduse korral voolutarve (mA)	160...260	~250	80...200	-
Operatsioonisüsteemi vajadus	Ei	Jah	Ei	Ei
SDK tugi	Jah	Ei	Jah	Jah
Hind (EUR)	20	15	17	20

## 3.2 Temperatuuri sensor

### 3.2.1 Anduri valik

Projekti raames loodava lahenduse eesmärk on lülitada valgustust vastavalt vajadusele, kui inimene viibib ruumis. Seega peab loodav lahendus suutma tuvastada ka paigalolevaid inimesi, kuna ruum, milles juhitakse süsteeme, on kontor, kus küll liigutakse natuke, aga enamus aega ollakse paigal laua taga. Järgnevalt võrreldakse kahte erinevat tehnoloogiat inimese kohaloleku tuvastamiseks: PIR ja MEMS.

PIR sensor koosneb tavaliselt hermeetiliselt suletud metallkestast ning infrapuna läbilaskvast materjalist (tavaliselt ränilääts) loodud avast. Sensori igal alal on 2 materjali, mis mõlemad tunnetavad infrapuna kiirgust. Kui sensor on mitteaktiivne, mõõdavad mõlemad alad sama infrapuna taset, mis passiivselt keskkonnast tuleb. Kui selle eest liigub aga läbi soe keha, näiteks inimene, tuvastab esimene pool sensorist muutuse infrapuna koguses, tekitades positiivse diferentsiaalmuutuse. Kui soe keha kaob vaateväljast, tekib vastupidine

olukord ning tekitab sensoris negatiivse diferentsiaalmuutuse[13].

MEMS tehnoloogiaga sensori tööpõhimõte on järgnev: ränilääts koondab kiirgavat soojust(kaug-infrapuna) *thermopile* sensorile läätse all. *Thermopile* sensor tekitab elektromotoorjõudu vastavalt koondunud kiirguse hulga. Elektromotoorjõu ja anduri sisemise termosensori väärtused mõõdetakse, misjärel arvutab sensor välja mõõdetud väärtuse interpolatsiooniga, võrreldes mõõdetud väärtuseid sisemise väärtustabeliga[14].

Kasutades PIR tehnoloogiat, oleks vaja lisaks sensorile kasutada ka taimerit. Peamine puudujääk neil sensoritel on paigalseisvate soojusallikate tuvastamine, kuna paigal olles ei teki suuri muudatusi, mida sensor suudaks tuvastada. Inimene ruumis viibib enamuse ajast paigal ning seega tuleks leida optimaalne väärtus taimeriga, et ei peaks kogu aeg liikuma ja tuli ei jääks põlema liiga kauaks, kui inimene ruumist lahkub. MEMS sensorit kasutades saab aga reaajas jälgida, kas inimene asub ruumis ning seega saab inimene terve aja olla liikumatult. Tule saab kustutada koheselt, kui kaamera vaateväljast lahkutakse.

Tabel 4. *Temperatuuriandurite võrdlus*[14],[15]

<b>Parameetrid</b>	<b>D6T-1A-01</b>	<b>D6T-8L-09</b>	<b>D6T-44L-06</b>	<b>D6T-32L-01A</b>
Mõõdetavate punktide arv	1	8	16	1024
Mõõtepunktide paigutus	1x1	1x8	4x4	32x32
Protokoll	I2C	I2C	I2C	I2C
Toitepinge (V)	5	5	5	5
Voolutarve (mA)	3.5	5	5	5
Hind (EUR)	13.68	31.32	33.76	140.06

Tabelis 4 on toodud välja valik erinevatest D6T sensoritest, mis kõik kasutavad MEMS tehnoloogiat. Sensorid erinevad teineteisest mõõdetavate kanalite poolest. Kuna sensor on antud projektis kasutusel toas inimese tuvastamiseks, oleks vaja sensorit maksimaalse resolutsiooni ning vaateväljaga. Mida suurem on resolutsioon, seda lihtsam on saadud andmetest leida inimene. Suurima resolutsiooni tõttu valiti käesoleva projekti jaoks OMRONI 1024 kanaliga D6T sensor, mis suudab mõõta 32x32 ruudustikus temperatuuri. Lisaks sellele on sensoril üpris madal voolutarve ning ei nõua eraldi toiteallikat, mille tõttu saab sensorit toita otse läbi plaadi.

### 3.3 Co2 sensor

#### 3.3.1 Anduri valik

NDIR tehnoloogia põhineb põhimõttel, et gaasi molekulid neelavad infrapunakiirgust ja igal gaasil neeldub erineva lainepikkusega infrapunakiirgus. Tavaliselt on kasutuses *thermopile* sisseehitatud filtriga tuvastamiseks kindla gaasi kogust. Süsihappegaasil on tugev neeldumine infrapunakiirgusel lainepikkusega 4,26 mikromeetrit ja *band-pass* filter on kasutuses, et eemaldada valgus, mis ei asu selles lainepikkuses. Gaasi molekulid neelavad energiat lambist, millest tulenevalt tuvastatakse *thermopile* poolal valguse intensiivsuse muutuseid ja sellest arvutatakse, kui palju on seal gaasi. [16]

Fotoakustiline spektroskoopia on tehnika, mida saab kasutada nii gaaside, vedelike kui ka tahkete ainete tuvastamiseks ja koguse määramiseks[17]. Selle tööpõhimõte on fotoakustiline efekt, mis tähendab, et valguse neeldumisel tuvastatavas aines tekivad akustilised lained. Sellistes sensorites suunatakse valgusallikas (tihtipeale laser) materjali pihta teatud lainepikkusega, mistõttu kindlad molekulid neelavad energiat ning nende temperatuur tõuseb. Kui valgusallika intensiivsus muutub samal ajal, kui materjali kogus püsib sama, tekitab molekulide perioodiline soojenemine surve muutuseid, mida saab mikrofoniga tuvastada[17].

Tabel 5. CO2 sensorite võrdlus[18],[19]

Parameetrid	K30	XENSIV PAS CO2
Meetod	Mittehajutav infrapuna	Fotoakustiline spektroskoopia
Mõõtevahemik (ppm)	0...10000	0...10000
Täpne mõõtevahemik (ppm)	0...5000	400...5000
Maksimaalne mõõtesagedus (ms)	2000	5000
Täpsus (%)	+/- 3	+/- 3
Toitepinge (V)	5.5...14	12
Voolutarve (mA)	40	6.1
Ühenduvus	I2C, UART	I2C, UART, PWM
Hind (EUR)	88	92

Peamine erinevus võrdluses olnud sensorite vahel on, et need töötavad erinevate tehnoloogiatega. Suur hulk omadusi on neil võrdsed, nagu mõõtetäpsus kindlas vahemikus ja üleüldine mõõtevahemik. Erinevused tulevad sisse aga nii täpse mõõtevahemiku suuruses kui ka mõõtesagedusega. K30 sensor suudab mõõta tihedamini ning täpsemalt. Lisaks eelnevalt mainitud erinevustele, on ka sensori tööks vajaminev toitepinge madalam ning sensori lubatud eluiga poole pikem kui XENSIV PAS CO2 sensor. Võrdluse tulemusena valiti kasutuseks CO2Meter'i K30 10000ppm sensor.

### **3.3.2 Kalibreerimine**

Süsihappegaasi sensori kalibreerimine on vajalik, kuna sensor muutub ajas ebatäpsemaks nii valgusallika kui ka detektori kulumise tõttu. Selle lahendamiseks tehakse kalibreerimise ajal mitu mõõtmist ning võetakse neist keskmine. Seejärel vaadatakse, kui palju see erineb algsest näidust, mis tehases seadistati. See väärtus kirjutatakse mällu ning seejärel hakkab sensor mõõtmistel seda kasutama, et korrigeerida mõõdetud tulemusi[20].

Sensorit saab kalibreerida nii lämmastikuga kui ka lihtsalt värske õhu käes. Lämmastikuga on sensor ka tehases kalibreeritud ning seda on vaja kasutada, kui on vaja saada täpsust piirkonnas 0-400ppm. See meetod on aga kallim, kuna vajab suletud testkeskkonda, lämmastiku gaasi olemasolu ning kalibreerimistarkvara. Teine võimalus on värske õhu käes mõõta, kui täpsus ei ole nii kriitiline ja hind on tähtsam. Selleks kalibreeritakse sensorit õues 400ppm juures ning siis lahutatakse kalibreerimistulemusest 400[20].

Sellel sensoril on ka autokalibreerimisvõimekus, nii et peale esialgset ülesseadmist ei pea seda uuesti kalibreerima, vaid see saab sellega ise hakkama. See meetod toimib põhimõttel, et mingil ajal ruumis ei ole ühtegi inimest ning eeldatakse, et süsihappegaasi tase langeb 400ppm juurde. Sellega võetakse iga päev madalaim näit ja hoiustatakse mälus. Selline võimekus laseb sensoril lihtsalt töötada ilma, et seda peaks kuskilt eemaldama ja kalibreerima, kuid kui ruumis ei lange süsihappegaasi tase tagasi 400ppm juurde, siis hakkab sensor üle aja näitama valesid andmeid[20].

## **3.4 Keskjaam**

Peatükis 2 võrreldakse sobivaid tarkvarasid keskjaamana kasutamiseks. Välja valitud tarkvara on vaja käima panna riistvara peal. OpenHAB toetab kolme põhilist kasutusmeetodit. Seda saab kasutada lihtsalt rakendusena arvutis, Dockeri konteineri sees ning eraldi süsteemina mõne SBC peal.

Docker on mugav vahend haldamiseks masinal mitut eraldi jooksvat teenust, võimaldades neile pakkuda isoleeritud keskkonda, kus neil on olemas kõik vajalik oma tööks. Docker lisab aga ühe abstraktsioonikihi süsteemi juurde, mis ainsa rakenduse jaoks süsteemil ei ole optimaalne. Lisaks on konteinerite põhimõtte olla maksimaalselt minimaalne, eemaldades süsteemist kõik tööriistad ja rakendused, mis pole teenuse jaoks hädavajalikud. Kui teha konteiner ka täieliku operatsioonisüsteemi põhjal, kaob ka peamine argument konteinerite kasutamiseks ning tegu on justkui tavalise virtuaalmasinaga. Selline lahendus oleks mõistlik kui oleks server, kus jookseks lisaks keskjaama tarkvarale veel muud teenused.

Projekti raames on aga mõistlik valida mõni SBC keskjaamatarkvara tarbeks, kuna muid teenuseid lisaks keskjaamatarkvarale pole vaja. Riistvara valides oli põhiline kriteerium, et valitud lahenduse voolutarbimine oleks võimalikult madal ning et tarkvara haldamine seal oleks võimalikult mugav.

Tabel 6. Riistvaralahenduste võrdlus[21],[22],[23]

Parameeter	Raspberry Pi 2B	Raspberry Pi 4B	BeagleBone Black
Protsessor	ARM Cortex-A7	ARM Cortex-A72	ARM Cortex-A8
Graafikaprotsessor	VideoCore IV	VideoCore VI	SGX530
Muutmälu (GB)	1	2	0.5
Ethernet	100Mb	1Gb	100Mb
Wi-Fi	-	802.11ac	-
USB pesad	4xUSB 2.0	2xUSB 2.0 + 2xUSB 3.0	1x USB2.0
Hind (EUR)	44	56	48

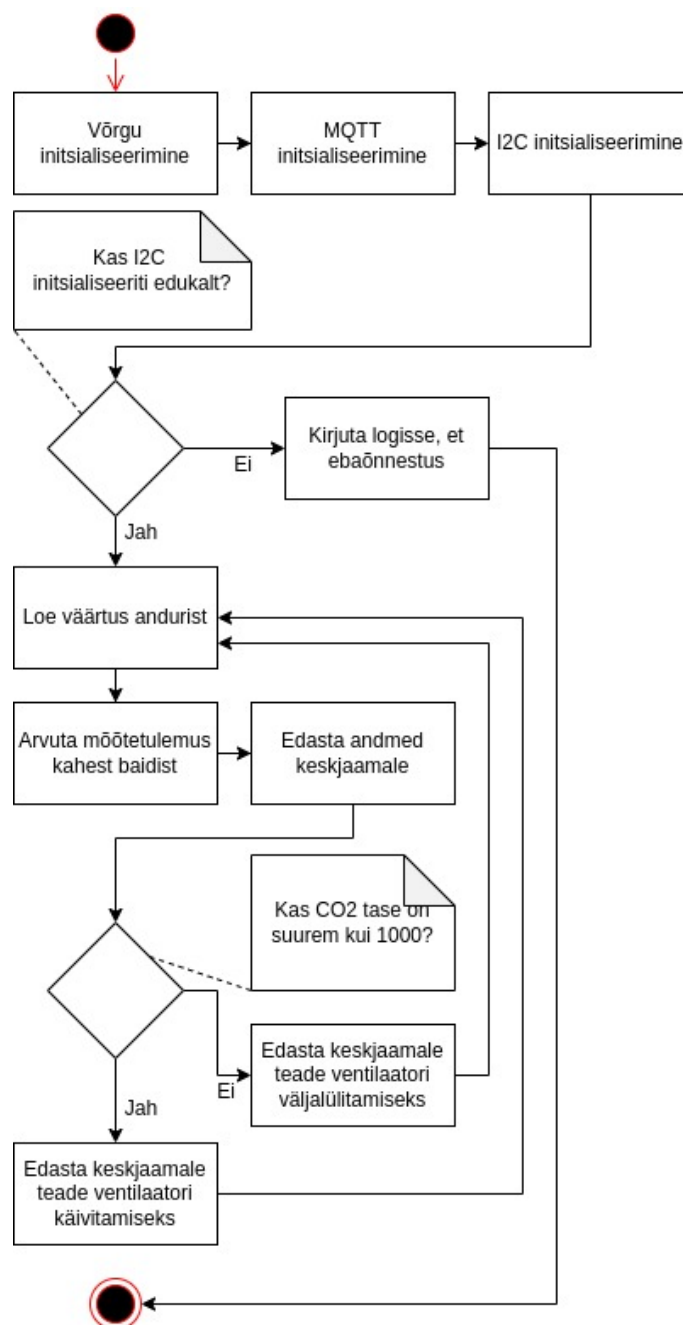
Võrdluses olevast kolmest arvutist on RaspberryPi 4 kõige võimekam. Sellel on kiireim protsessor ning kõige rohkem mälu. Sellel plaadil on teistega võrreldes ainsana olemas Wi-Fi võimekus. See on hea, kuid antud lahenduses pole see vajatud, kuna keskjaama saab ühendada korteri Wi-Fi ruuteri külge juhtmega ning see ei pea ringi liikuma. See arvuti on ka valikust kalleim. RaspberryPi 2 ja BeagleBone Black on üpris sarnase võimekuse ja hinnaga, kuigi RaspberryPi 2 on rohkema mäluga. Kõigil arvutitel on ka võimalus ühendada külge ekraan ning USB kaudu erinevaid sisendseadmeid, et hallata seadet otse.

Võrdluse tulemusel on näha, et tehniliste näitajate poolest on RaspberryPi 4 parim valik, kuid see on ka ühtlasi kulukaim valik. Autoril on varasemast projektist ka kasutada

RaspberryPi 2 arvuti ning kuna see on piisav, et openHAB saaks töötada, ei hakata uut riistvara ostma. Kui eelnevalt poleks seadet olemas, oleks mõistlik teha investeering ning hankida uuem RaspberryPi arvuti rohkema mäluga.

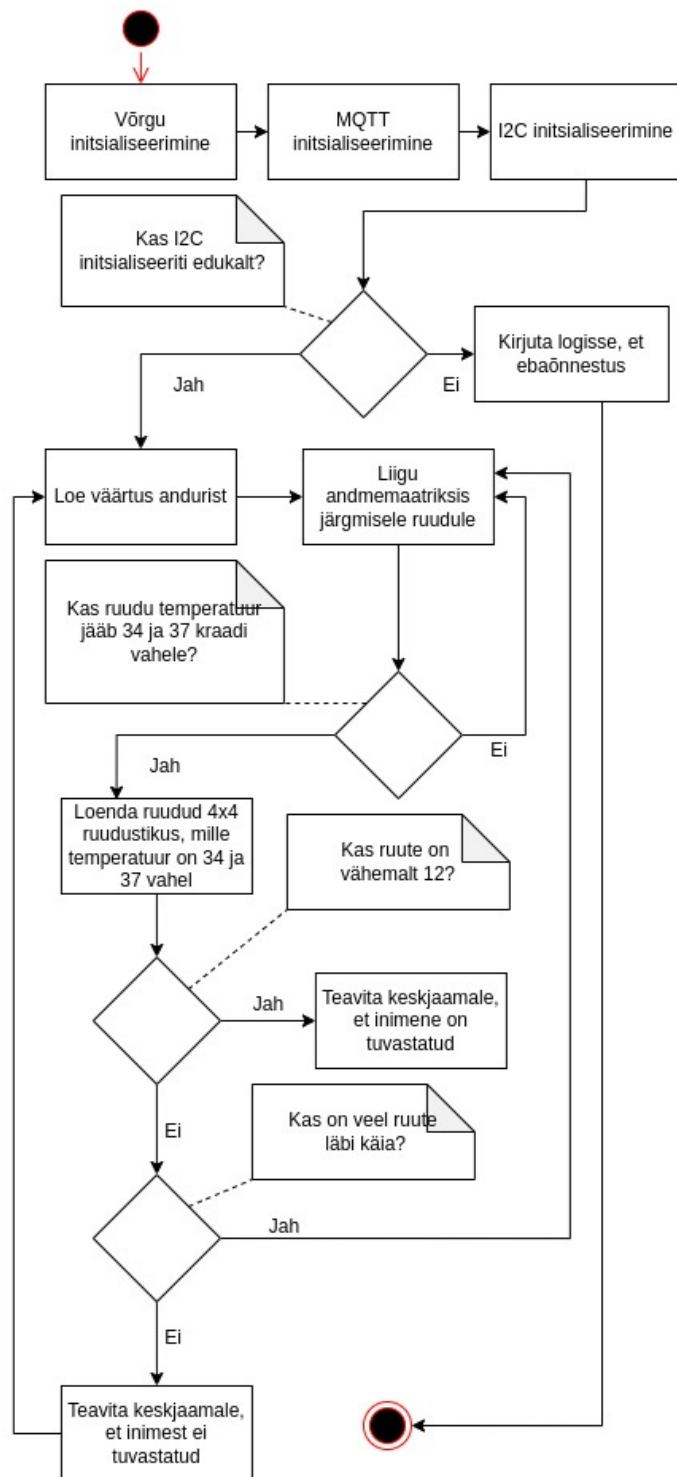
## 4. Tarkvara

### 4.1 Süsteemi algoritm



Joonis 2. Süsihappegaasi sensori algoritm

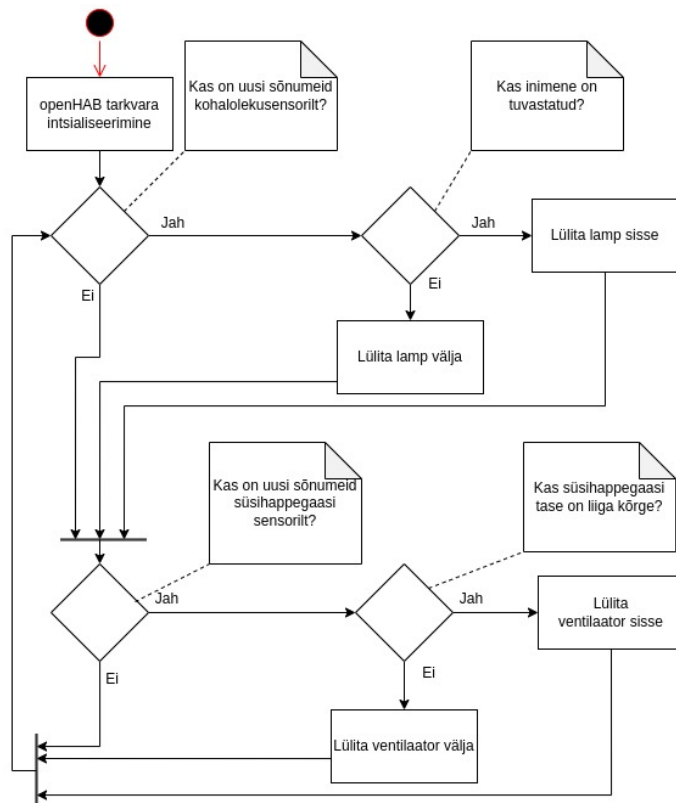
Järgnevalt antakse samm-sammuline ülevaade kogu süsteemi toimimisest. Joonisel 3



Joonis 3. Tuvastussensori algoritm

on toodud tuvastussensori algoritm ning joonisel 2 süsihappegaasi sensori oma. Peale tuvastussensori sisselülitamist initsialiseeritakse esimese asjana võrgunduse osa. Seejärel on viide, peale mida initsialiseeritakse MQTT protokolliga vajalik funktsionaalsus. Järgnevalt lülitatakse sisse I2C funktsionaalsus ning kontrollitakse, kas see õnnestus. Kui esineb viga, antakse nendest teada läbi *debug* konsooli. Peale seda alustatakse lõpmatult jätkuvat tsüklit, mille käigus tehakse mõõtmine, töödeldakse andmeid ning tuvastakse,





Joonis 4. Keskjaama algoritm

kas inimene on sensori vaateväljas. Tulemus edastatakse keskjaamale. Süsihappegaasi tööpõhimõte on sarnane tuvastussensorile. Esialgsed sammud on samasugused ning vahe tuleb sisse peamises töösüklis. Sensor teostab mõõtmise ning olles tulemise kätte saanud, edastab selle keskjaamale.

Keskjaama masinas töötab Mosquitto vahendaja, mis kuulab sissetulevaid sõnumeid ning jagab neid edasi. OpenHAB tarkvaras on loodud kanalid kuulamiseks erinevaid teemasid, millesse sõnumite saabumisel kuvatakse neid andmeid kasutajale. Lisaks on loodud reeglid, mis juhivad saabunud andmete põhjal seadmeid. Kui kohalolekusensorilt tulevad andmed, et inimene on tuvastatud, lülitatakse sisse lamp ning kui inimest enam ei tuvastata, lülitatakse see välja. Kui aga saadab süsihappegaasi sensor andmeid, et süsihappegaasi tase on liiga kõrge, pannakse puhur tööle. Keskjaama algoritm on nähtav joonisel 4

## 4.2 Võrdlus juhtmevabade andmeedastusprotokollide vahel

Projekti raames valmistatavad sensorid suhtlevad keskjaamaga üle Wi-Fi võrgu, mis tähendab, et neil on vaja kasutada suhtluseks mõnda protokollit, millega saaks andmeid saata keskjaamale. Järgnevas tabelis 7 võrreldakse erinevate protokollide omadusi ja valitakse välja sobiv protokoll sensoritele keskjaamaga suhtluseks.

Tabel 7. MQTT ja sarnaste protokollide võrdlus[24],[25],[26]

Protokollid	MQTT	HTTP	XMPP	AMQP
Arhitektuur	<i>pub/sub</i>	server/klient	server/klient (võimalik ka P2P)	<i>pub/sub</i>
Sõnumi suurus	väike	suur	suur	väike
Sõnumite tüüp	binaar	tekst	tekst	binaar
Range sõnumi struktuur	Ei	Jah	Jah	Ei
Keskse serveri vajadus	Jah	Jah	Jah	Jah
ESP32 SDK tugi	Jah	Jah	Ei	Ei

Kõik võrdluses toodud protokollid vajavad kasutamiseks keskset serverit. MQTT ja AMQP kasutavad *brokerit* ja *pub/sub* mudelit suhtluseks. See tähendab, et on üks keskne teenus, mille külge erinevad osad ühenduvad ning saadavad sinna ja saavad sealt sõnumeid. HTTP ja XMPP kasutavad server/klient mudelit. Suurim erinevus nende meetodite vahel on, et läbi vahendaja saavad liikmed andmeid nii kui tekib uusi sõnumeid teema alla, samas kui server/klient mudeliga peavad liikmed käima küsimas sõnumeid serverist. Võrdluses toodud protokollidest suurim andmemahut on HTTP protokollil. Teised protokollid on andmemahult väiksemad. Kõige pisem on andmemahut MQTT protokollil, mis annab sellele teiste ees eelise. Lisaks on ka MQTT protokoll toetatud ESP32 arendusplaadi SDK sees, mis annab eelise selle protokolliga suhtluse realiseerimiseks teiste ees. Kuigi ka HTTP on toetatud SDKs, ei ole selle muud omadused niivõrd head, mistõttu osutus valituks MQTT protokoll.

### 4.3 MQTT

MQTT on andmeedastusprotokoll, mis loodi seadmetevahelise suhtluse tegemiseks võimalikult efektiivselt ja kokkuhoidlikult. Protokoll loodi, et monitorida nafta *pipeline* SCADA süsteemides üle satelliidi lingi, mis oli sel ajal väga kallis. Protokoll toimib *pub/sub* süsteemil, mis tähendab, et on olemas keskne vahendaja (ingl. k. broker), mille külge ülejäänud seadmed ühenduvad. Vahendajale saavad seadmed andmeid saata teemade (ing.k. topic) all, mis teeb andmeedastuse selgeks ja tekitab struktuuri. Iga teemat saavad ka seadmed jälgida nii, et kui uus teade teema alla ilmub, saavad nad selle teate sisu. Selline struktuur

võimaldab luua konkreetset hierarhiat teemadesse ning igal seadmel avalikustada sõnumeid ainult sinna, kuhu vaja ning ka kuulata infot, mis on just talle vajalik[27].

MQTTI on unikaalseid omadusi, millest märkimisväärsamad on: [28]

- On loodud lihtsa ja ressursse vähenõudva protokollina, mis kasutab avalikustamise/jälgimise süsteemi vahendamaks informatsiooni seadmete ja serveri vahel.
- Ei ole vajalik, et iga seade oleks ühendatud serveri külge samal ajal, seade saab informatsiooni kätte siis, kui serveriga ühendub.
- Lubab informatsiooni edastamisel kasutada ranget hierarhiat ja teemade eraldatust, et iga seade saaks kätte ainult informatsiooni, mida vajab.

Andmed on kategoriseeritud skeemi /ruum/seade/andmed alusel. Selline hierarhia loodi lähtuvalt openHAB'i võimalusest organiseerida seadmed ruumide kaupa, mis teeb vigade avastamise protsessi lihtsamaks. Keskjaamas kasutatakse vahendajat, mis suhtleb omakorda openHAB'iga ja edastab sinna seadmetelt infot, mida see vajab.

Tabel 8. *MQTT vahendajate võrdlus*[29]

Vahendajad	Mosquitto	RabbitMQ	HiveMQ
Avatud lähtekood	Jah	Jah	Jah (hiljuti)
Suurus	Minimaalne	Suur	Suur
Seadistamise keerukus	Väike	Suur	Suur

Võrdluses välja toodud vahendajad on üpris sarnased. Peamised erinevused ilmnevad erinevate lisavõimekustega. Mosquitto on neist kõige minimaalsem, sellel on konfiguratsioonifail ning seda muutes saab rakendust juhtida. Teistel vahendajatel on ka käsurea liides, millega jooksvalt seadeid muuta. Antud projekti raames pole seda aga vaja, kuna vahendaja peab vaid suutma edastada sõnumeid keskjaama ning seadmete vahel. Peamine eesmärk on tarbida keskjaamal võimalikult vähe ressursse, et keskjaama tarkvara saaks töötada vabalt. Sellest võrdlusest osutus valituks Mosquitto.

### 4.3.1 Võrdlus riistvaraliste andmeedastusprotokollide vahel

Tabelis 5 on toodud kolme populaarse andmeedastusprotokolli võrdlus. Kuigi nad on olemuslikult erinevad, kasutatakse neid kõiki erinevatel põhjustel mikrokontrollerite ja nende lisaseadmete liidestamisel. Järgnevalt on tabelis toodud võrdlus ka lahti selgitatud.

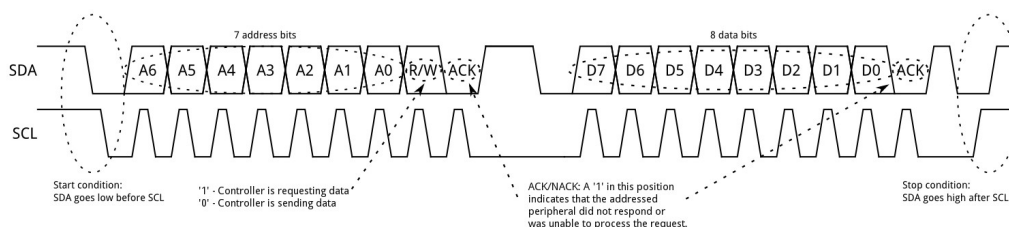
## 4.4 I2C

I2C protokoll on valitud kasutuseks antud projekti raames arendusplaatidele andurite lisamiseks. Protokoll valiti, kuna seda on füüsiliselt lihtne realiseerida, kasutades vaid kahte juhet. Lisaks eelnevale on protokoll kasutamiseks tugi olemas nii valitud arendusplaadil, kui ka enamikel sensoritel koos dokumentatsiooniga. Andmeedastuskiirus pole sellel suurim, kuid see pole ka määrav, kuna edastatavad andmehulgad pole suured.

I2C on aastal 1982 Philips'i poolt loodud jadasiin, mis loodi lihtsustama suhtlust komponentide vahel, mis asusid samal trükkplaadil. Algselt oli siini maksimaalne edastuskiirus 100kb/s, kuid hiljem lisandusid ka 400kb/s ja ka 3.4Mb/s kiirused. Sarnaselt andmeedastuskiiruse piiratusele oli algselt kasutusel vaid 7 bitine adresseerimine, millele lisandus hiljem 10 bitine. Protokoll on laialdaselt kasutusel ühendamiseks ka seadmeid, mis asuvad erinevatel plaatidel, võimaldades neil omavahel lihtsalt suhelda. Siini lihtsus ja paindlikkus on põhilised omadused, mis muudavad selle atraktiivseks lahenduseks paljudes kasutusjuhtudes. [30] Parimad omadused on:

- Vaja on vaid kahte juhet
- Pole kindlaid piiranguid edastuskiirusele, *master* kontrolleri määrab kiiruse
- Kõikide komponentide vahel on *master/slave* suhe
- Igat siinile ühendatavat seadet on tarkvaraliselt võimalik adresseerida
- Võimaldab kasutada mitut *master* seadet

Siin koosneb kahest juhtmest: SCL ja SDA. Neist esimene, SCL, on taktsignaali edastuseks ja teine, SDA, on andmete edastamiseks. Kella signaali genereerib alati kontrollseade. Selle eesmärk on lasta juhtseadmel seada andmeedastuskiirust ilma eelnevalt seadmetel vastava seadistuse tegemist[31].



Joonis 5. *i2c* paketid[32]

Joonisel 5 on näha tavalisi pakette I2C siinil. Lisaks tavalistele 1 ja 0 bittidele, on lubatud ka spetsiaalsed START ja STOP signaalid, mis käituvad sõnumite eraldajatena. Juhtmasin edastab START signaali, millele järgneb suhtluseks valitud seadme 7-bitine aadress. Nendele bittidele järgneb lugemise (0) või kirjutamise (1) bitt. Kui seade eksisteerib ja

tunneb enda aadressi ära, vastab see ACK bitiga, mille peale juhtmasin hakkab kas andmeid kuulama või liinile kirjutama. Seade vastavalt kas võtab andmeid vastu või edastab. Esimesena saadetakse suurim bit, millele järgnevad kõik väiksemad. Start signaaliks loetakse olukorda, kus SCL liin on kõrge ja SDA muutub kõrgest madalaks. Stop signaaliks loetakse olukorda, kus ka SCL on kõrge ja SDA muutub madalast kõrgeks. Kõik muud SDA liini muutused toimuvad SCL liini madalas olekus. Kui juhtmasin soovib alamasse kirjutada, saadab ta iga baidi eraldi, kuulates alamalt vastu tulevat ACK bitti. Kui juhtmasin soovib kuulata alamast andmeid, kuulab ta eraldi igat baiti, saates vastu ACK biti iga baidiga, välja arvatud viimasega. Viimane bait on kasutuses edastusvigade tuvastamiseks.

Igal alamseadmepool on oma unikaalne aadress. Alamseadmepool suhtlemiseks kirjutab ülem siinile alluva aadressi, kellega soovib suhelda. Siini kuulavad kõik seadmed ning kui seade tunneb ära enda aadressi, saadab siinile vastu kinnitusbiti. Peale seda saab hakata ülem saatma siinile käskude lugemiseks või kirjutamiseks ning siis kuulama, kas alam vastab talle ning saab andmed vastu võtta. Alamseadmed ei saa ise suhtlust alustada ega informatsiooni saatma hakata, mistõttu peab alati juhtmasin alustama andmeedastusprotsessi.

#### **4.4.1 Kohaloleku sensor**

Suhtlus D6T sensoriga käib arendusplaadil üle I2C liidese. Arendusplaadilt antakse käsk kirjutada 1 bait seadme aadressile ning sealt seejärel andmeid lugema hakata. Kuna projektis on kasutusel 1024 kanaliga sensor, tuleb sensorist lugeda kokku 2051 baiti andmeid. Sellel sensoril on ka omaette käsk andmete lugemiseks. Iga mõõtmistulemus saadetakse välja kahe eraldi baidina ja seetõttu tuleb need pärast juhtmasinas uuesti kokku pakkida. Edastatavate andmete seas on:

- PTAT - referentstemperatuur, mis on seadmes kasutusel temperatuuride määramiseks
- P0 - P1023 - andmepunktid, mis on kogutud mõõtmisega
- PEC - paketi veakontroll

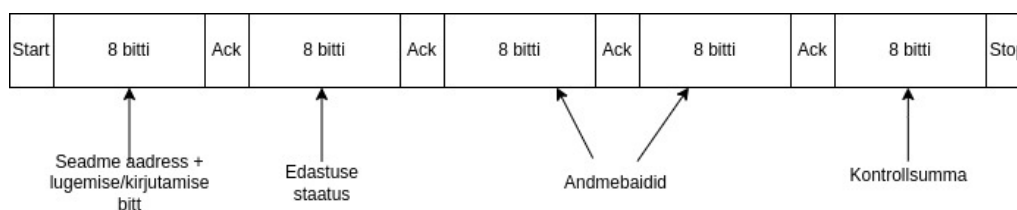
Esimesena edastatakse PTAT, millega saab kontrollida sensori enda sisemist referentsväärtust, et kontrollida ega sensoriga pole midagi juhtunud. Peale seda edastatakse andmepunktid ning lõpuks ka veakontroll, veendumaks et edastamisel ei tekkinud vigu.

Inimese tuvastamine on realiseeritud järgneva algoritmiga: sensor teeb mõõtmise ja hakkab järjest andmemaatriksit läbi käima. Kui leidub andmepunkt, mille temperatuur jääb 34 ja 37 kraadi vahele, otsitakse sellest andmepunktist algavat 4x4 ruudustikku. Kui sellest ruudustikust leitakse vähemalt 12 ruutu, mis jäävad ka samasse temperatuurivahemikku, annab sensor teate, et on leitud inimene. Selline algoritm on töö raames piisav, et näha toas

olevat inimest ja vältida ebavajalikku müra. Sensorile loodud programmikood on leitav lisast 1.

#### 4.4.2 Süsihappegaasi sensor

Suhtlus K30 sensoriga käib arendusplaadil üle I2C liidese. Arendusplaadilt antakse kõigepealt käsk kirjutada 1 bait seadme aadressile. Peale päringu edastamist peab ootama, et sensor suudaks päringut töödelda. Peale ootamist edastatakse käsklus andmeid lugema hakata. Edastatavate andmete sisu on toodud joonisel 6.



Joonis 6. Süsihappegaasi sensori edastatavad baidid

Andmebaidid pakitakse kokku ning edastatakse edasi keskjaamale. Sensorile loodud programmikood on leitav lisast 1.

#### 4.5 Keskjaam

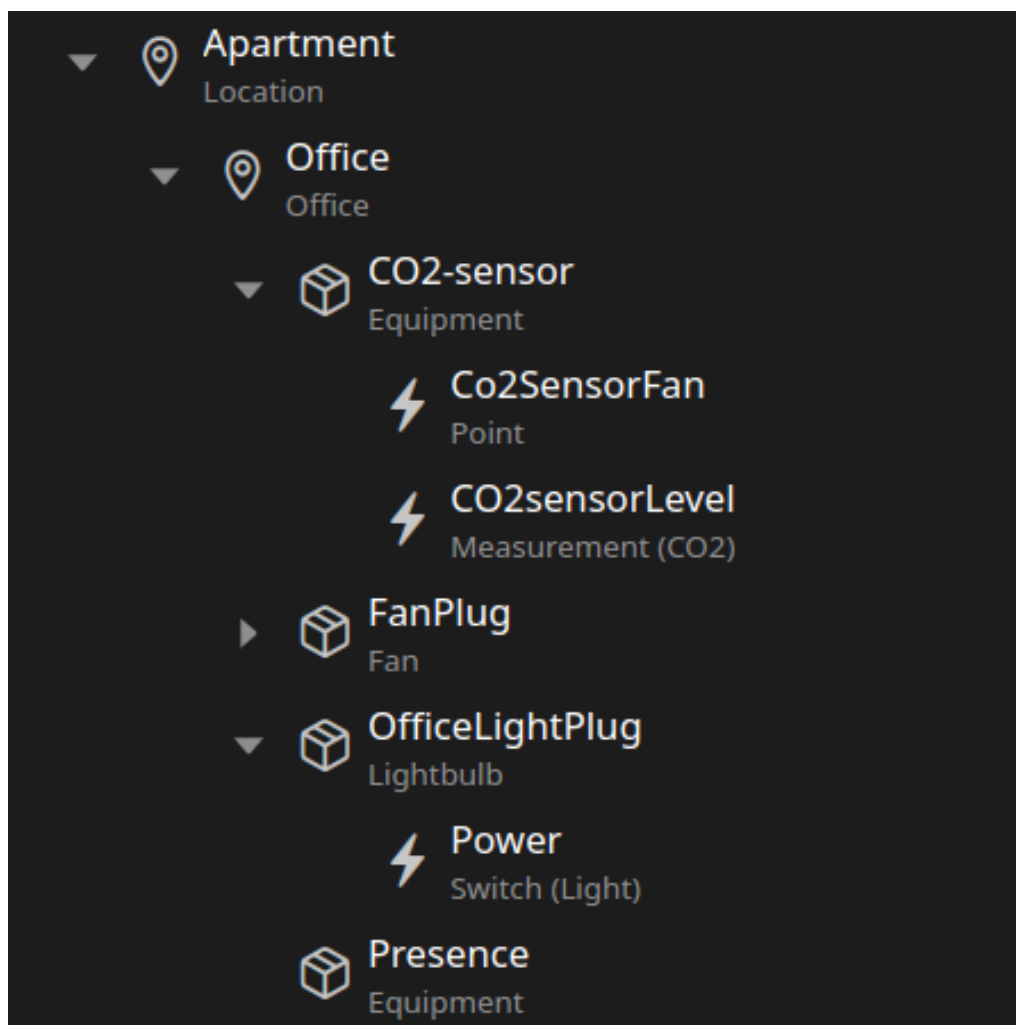
Keskjaamas kasutusel olev openHAB tarkvara jaotab suhtluse erinevateks kihtideks, mis on:

- *Binding* ehk tarkvara füüsilise kihiga liidestumiseks
- *Thing* ehk ese - tarkvaraline kujutus füüsilisest esemest
- *Channel* ehk kanal - *thing* ja *itemi* vaheline ühendus
- *Item* ehk asi - füüsilise seadme andmete kogum, abstraktsioon esemest
- *Rules* ehk reeglid, mille järgi käivad automaatsed tegevused
- *Sitemap* ehk kasutaja loodud kasutajaliides esemete juhtimiseks

Selline meetodika aitab eristada erinevaid tükke kontseptuaalselt ning ka praktiliselt. Võttes kasutusele *bindingu*, saab iga seade, mis kasutab seda protokollit, suhelda süsteemiga ning pole vaja luua erinevaid ühilduskihte. Füüsilistel süsteemis kasutatavatel seadmetel on tihti mitu võimekust. OpenHABis kasutusel oleva meetodikaga saab selliste seadmete iga võimekust käsitleda eraldi seadmena ning juhtida neid individuaalselt.

## 4.5.1 Kasutajaliides

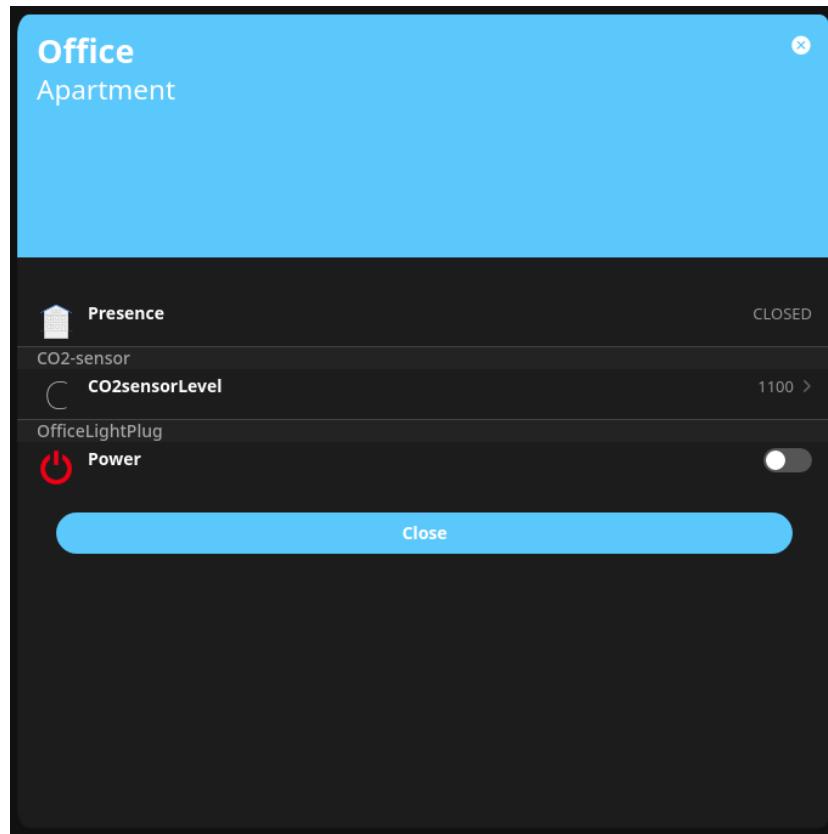
OpenHABis on esemete grupeerimiseks peamiselt kasutusel semantiline mudel. Selle mudeli põhiline eesmärk on luua abstraktsuskiht seadmete jaoks, mis meenutaks võimalikult palju füüsilise maailma seadmeid ilma, et kasutaja peaks iga seadme puhul teadma, mis on füüsiline seade selle taga. Kasutaja seisukohast ei ole vajalik teada, kas lüliti, mida ta lülitab, on taustal Philips Hue lambipirn, KNXi lüliti või mingi kolmas seade[33]. See mudel kasutab asju ehk *itemeid*, et seda abstraktsust luua. Selle mudeliga on ka osaliselt automatiseeritud erinevate lehtede loomine kasutajaliidesesse, mis teeb seadmete kontrollimise alade ja tubade kaupa mugavaks.



Joonis 7. Semantiline mudel openHABis

OpenHAB tarkvaras saab luua erinevaid kasutajaliideseid. Peamine neist on MainUI, mis lisati openHABi kolmanda versiooniga. See liides loodi eesmärgiga administreerimine ja ruumidest ülevaate saamine lihtsamaks teha. Lisaks pakub see kasutajaliides ka rollipõhist ligipääsu, et lasta seda kasutada kõigil ruumis viibijatel ilma administraatori õigusteta. Seda kasutajaliidest ei pea ka nullist üles ehitama nagu on olnud vajadus eelnevate versioo-

nide puhul. Kui on kasutuses semantiline mudel, genereeritakse see liides automaatselt olemasolevast informatsioonist. Selle lehe kujundust saab administraator ka vajadusel muuta. See on kasutuses projektis peamise kasutajaliidesena. Lisaks sellele on ka loodud paar lehte tahvelarvutist süsteemide juhtimiseks mõeldud HABPaneliga.



Joonis 8. *OpenHABi kasutajaliides*

## 4.5.2 Loodavad sensorid

Kohaloleku tuvastamise sensori jaoks on loodud kontakti tüüpi asi. See annab sensorile binaarse oleku, see on kas sees või väljas. Kuna sensor tuvastab inimese kohaloleku plaadi peal ja saadab edasi keskjaama ainult info, kas inimene on leitud või ei, on see hea sobivusega kasutuseks. Süsihappegaasi sensor aga on loodud tavalise numbrilise väärtusega. See annab võimaluse hästi joonistada graafikuid ning koguda ajalooliselt andmeid õhukvaliteedi kohta.

## 4.5.3 Automatiseerimine

Automatiseerimine on openHABis tehtud läbi reeglite. Reegleid saab luua nii läbi kasutajaliidesega kui ka koodi kirjutamisega. Antud juhul on loodud reeglid läbi kasutajaliidesega. Iga tegevuse jaoks on vaja luua eraldi reegel, seega lambi sisse- ja väljalülitamiseks on vaja kahte reeglit. Esimene jälgib kohaloleku sensori olekumuutust, et inimene on tuvastatud



ning lülitab lambi sisse, teine ootab, kuni inimest pole enam tuvastatud ning lülitab lambi välja. Sarnaselt on tehtud süsihappegaasi sensori ja ventilaatoriga: kui süsihappegaasi tase on liiga kõrge, lülitatakse ventilaator sisse ning kui tase on normaalne, lülitatakse ventilaator välja.

## 5. Testimine

Sensoreid testiti kokku kolmes staadiumis:

- I2C suhtluse kinnitamiseks
- MQTT suhtluse kinnitamiseks
- openHAB'iga suhtluse kinnitamiseks

Selline lähenemine garanteerib, et igasse järgnevasse staadiumisse liigutakse töötava süsteemiga ning see teeb vigade otsimise lihtsamaks. Kui eelnevalt on veendunud, et süsteem toimib, ei pea sealt enam vigu otsima ja saab keskenduda täpsemalt ühele alale. Järgnevalt on toodud ülevaade mõlema sensori testimise tulemustest ning väljunditest.

### 5.1 I2C testimine

I2C siinil suhtluse kinnitamiseks ühendati sensor arendusplaadi külge ning saadeti andme-päringuid. Suhtluse nägemiseks kasutati ESP32 arendusplaadi SDK sees olevat monitori. Vastu tulevad andmed väljastati konsoolile, mida sai näha monitorist.

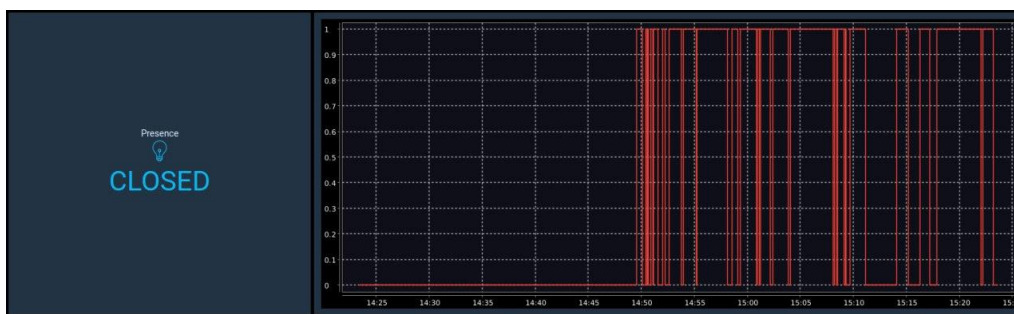
Süsihappegaasi sensoril on väljundiks vaid üks arv, mida kuvati iga päringuga ekraanile. Kohaloleku sensori puhul väljastati andmeid maatriksina ning seejärel visualiseeriti neid tabelarvutusprogrammiga. Joonisel 9 on kohaloleku sensori väljund, kui see on suunatud otse inimesele ning joonisel 10 ülevalt lae alt.

### 5.2 MQTT testimine

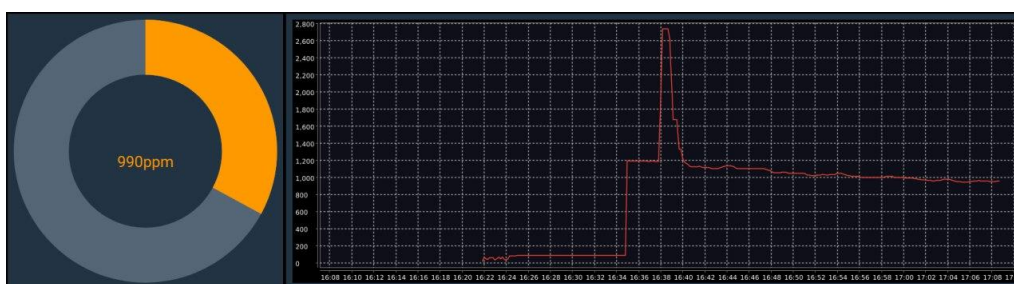
MQTT testimiseks kasutati vahendajaga kaasa tulevat tööriista `mosquitto_sub`, millega on võimalik terminalist kuulata kanalis liikuvaid andmeid. See toimib ka üle võrguühenduse, kuid kasutati vahendajaga samas masinas kohalikul. Mõlemad sensorid lülitati sisse, ning kui need olid WiFi võrku ühendunud, hakati vaatama, mis info keskjaamani jõudis. Järgneval joonisel 11 on toodud kohaloleku sensori andmekanali kuulamine, kus edastatakse binaarset väärtust, kas on tuvastatud inimene või ei.



sensorite testimisest.



Joonis 12. Kohaloleku sensori ekraan



Joonis 13. Süsihappegaasi ekraan

## 5.4 Süsteemi demonstreerimine

Süsteemi toimimise demonstreerimiseks on loodud 2 videot, mis on toodud all lisas 2. Esimesel on demonstreeritud lambi põlema panemist ning teisel süsihappegaasi süsteemi toimimist. Lampi ja ventilaatorit juhitakse juhtmevaba pistikuga, mis on ühendatud ka keskjaamaga. Esimesel juhul kui lamp tuvastab inimese, käivitub reegel, millega lülitatakse pistik sisse ning seega läheb lamp tööle. Kui inimene sensori vaateväljast kaob sulgub pistik ning lamp kustub. Süsihappegaasi sensoriga on loodud reegel, millega lülitatakse pistik sisse, kui süsihappegasi tase liiga kõrge on. Demonstratsiooni raames on lävendiks pandud 1000ppm.

## 6. Probleemid

### 6.1 Inimese tuvastamine

Antud töös on kasutatud inimese tuvastamiseks lihtsat algoritmi, mis otsib lähestikku asuvaid ruute, kus väärtus sobib inimkeha temperatuurivahemikku. Selline algoritm on aga algeline ning ei suuda vahet teha inimesel ja näiteks radiaatoril. Töö raames on see viga eemaldatud sensori paigaldamisega nii, et ükski soojusallikas peale inimese ei jää sensori vaatevälja.

Sensori töökindluse suurendamiseks muudes kasutusjuhtudes peaks aga muutma algoritmi, millega inimese kohalolu tuvastatakse. Esimene võimalus oleks dünaamiliselt vaadata, millistes ruutudes on muudatused ning millised on staatilised ning sellega seoses välistada statsionaarsed esemed tuvastusest. Teine võimalus on tuvastada inimese kuju, kuid see nõuaks sensori asetust väga spetsiifilisse kohta, kust oleks võimalik näha inimest konkreetse nurga alt.

Lisaks peaks lambi sisse- ja väljalülitamise efektiivsemaks juhtimiseks mõõtma ka valgustatuse taset ruumis. Seda saaks teha kas algelisemalt mõne fotoresistoriga samal arendusplaadil, või saab ka hankida mõne müügiloleva valgussensori, mis oleks täpsem. See aitaks vältida sisselülitamist piisavalt valgusajal, mis praegusel juhul jääb inimese kanda.

## 7. Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli automatiseerida ruumi kahte süsteemi. Nendeks olid õhuvahetus ning valgustus. Töö tulemusena valmis keskseade RaspberryPi arvutil töötava openHAB tarkvaraga ning kaks sensorit. Valgustuse juhtimiseks kasutatav kohaloleku sensor kasutas Omroni D6T infrapuna sensorit, et jälgida inimese kohalolekut toas ning vastavalt vajadusele lülitada sisse lampi. Õhukvaliteedi jälgimiseks valiti süsihappegaasi taseme jälgimine. Süsihappegaasi taseme jälgimine ruumis annab piisava ülevaate õhu kvaliteedist, ehk kui selle tase on kõrge, tuleks ruumi ventileerida ja kui tase on madal, võib eeldada et õhukvaliteet on normaalne. Selleks kasutati CO2monitors sensorit K30.

Mõlemad sensorid kasutavad ESP32 kiibil põhinevat arendusplaati, millega komponendid suhtlevad üle I2C siini. Arendusplaat suhtleb keskjaamaga üle WiFi saates MQTT protokolliga pakette. Keskjaamal on peal MQTT vahendaja tarkvara, mis võtab vastu sensoritelt sõnumeid ning keskjaama tarkvara kuulab sealt tulevaid uusi sõnumeid. Tarkvara kajastab saadud andmeid *dashboardil* ning saadab edasi sõnumeid vastavatele süsteemidele, mis juhivad valgust ja ventilatsiooni.

Antud projekti edasiarendusena on võimalik lisada süsteemi veelgi rohkem sensoreid, et edasi automatiseerida ruumi ja ka terve kodu süsteeme. Lisaks saab ka lisada erinevaid seadmeid mida juhtida. Keskjaam toetab ka palju erinevaid protokolle, nii et see ei piira uute seadmete lisamist. Saab juurde lisada ka erinevad kasutaja teatamise süsteemid, millega anda edasi kriitilist informatsiooni kui selleks on vajadus.

## Kasutatud kirjandus

- [1] *2021 Statistics: U.S. and Global Smart Home Systems Market Size Share Worth Over USD 140.2 Billion by 2026, Says Facts Factors.* [Kasutatud 2021-12-14]. URL: <https://finance.yahoo.com/news/2021-statistics-u-global-smart-113700538.html>.
- [2] *HEALTH EFFECTS OF BAD INDOOR AIR QUALITY.* [Kasutatud 2021-12-14]. URL: <https://sustainablelivingassociation.org/health-effects-of-bad-indoor-air-quality/>.
- [3] Michael Forster. *7 Reasons Why You Should Measure CO2 Inside Buildings.* [Kasutatud 2021-11-]. URL: <http://insight.gbig.org/7-reasons-why-you-should-measure-co2-inside-buildings/>.
- [4] Purdylounge.com. *OpenHAB Vs Home Assistant: Detailed Comparison By An Expert in 2021.* [Kasutatud 2021-11-26]. URL: <https://purdylounge.com/openhab-vs-home-assistant/>.
- [5] *Your home at your command.* [Kasutatud 2022-01-03]. URL: <https://www.apple.com/ios/home/>.
- [6] *From smart home to smarter living.* [Kasutatud 2022-01-03]. URL: <https://www.smartthings.com/getting-started>.
- [7] Jay Peters. *Prolonged AWS outage takes down a big chunk of the internet.* [Kasutatud 2021-12-09]. URL: <https://www.theverge.com/2020/11/25/21719396/amazon-web-services-aws-outage-down-internet>.
- [8] Espressif Systems Ltd. *ESP32 Series Datasheet.* [Kasutatud 2021-12-28]. URL: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf).
- [9] *Raspberry Pi Power Consumption Data.* [Kasutatud 2021-12-28]. URL: <https://www.raspberrypi-spy.co.uk/2018/11/raspberry-pi-power-consumption-data/>.
- [10] Giuseppe Cassibba. *RASPBERRY PI ZERO WH DATASHEET.* [Kasutatud 2021-11-27]. URL: <https://peppe80.com/raspberry-pi-zero-wh-datasheet/>.

- [11] *ESP8266: General Information Specifications*. [Kasutatud 2021-12-28]. URL: <https://esp8266-shop.com/esp8266-guide/esp8266-information/>.
- [12] *Arduino Uno Rev3*. [Kasutatud 2021-12-28]. URL: <https://store.arduino.cc/products/arduino-uno-rev3/>.
- [13] *How PIRs Work*. [Kasutatud 2021-12-16]. URL: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>.
- [14] OMRON Corporation. *MEMS Thermal Sensors D6T User Manual*. [Võrgumaterjal] [Kasutatud 2021-10-31]. URL: [https://omronfs.omron.com/ko\\_KR/ecb/products/pdf/en\\_D6T\\_users\\_manual.pdf](https://omronfs.omron.com/ko_KR/ecb/products/pdf/en_D6T_users_manual.pdf).
- [15] *D6T Datasheet*. [Kasutatud 2022-01-02]. URL: <https://www.digikey.com/en/datasheets/omron-electronics-inc-emc-div/omron-electronics-inc-emc-div-en-d6t>.
- [16] Texas Instruments Incorporated. *LMP91051 NDIR CO2 Gas Detection System*. [Kasutatud 2021-11-30]. URL: <https://www.ti.com/lit/an/snaa207/snaa207.pdf>.
- [17] *Photoacoustic Spectroscopy*. [Kasutatud 2021-12-01]. URL: <https://okumuragroup.sites.caltech.edu/research-interests/photoacoustic-spectroscopy>.
- [18] *Datasheet: K-30 Sensor*. [Kasutatud 2021-12-01]. URL: [http://co2meters.com/Documentation/Datasheets/DS\\_SE\\_0118\\_CM\\_0024\\_Revised9%20\(1\).pdf](http://co2meters.com/Documentation/Datasheets/DS_SE_0118_CM_0024_Revised9%20(1).pdf).
- [19] *Preliminary datasheet of XENSIV PAS CO2*. [Kasutatud 2021-12-01]. URL: [https://www.mouser.com/datasheet/2/196/Infineon\\_EVAL\\_PASCO2\\_SENSOR\\_DataSheet\\_v01\\_02\\_EN-2399950.pdf](https://www.mouser.com/datasheet/2/196/Infineon_EVAL_PASCO2_SENSOR_DataSheet_v01_02_EN-2399950.pdf).
- [20] *AN131 – CO2 Sensor Calibration: What You Need to Know*. [Kasutatud 2021-12-20]. URL: [http://co2meters.com/Documentation/Other/AN\\_SE\\_0020\\_Sensor\\_Calibration\\_Procedure.pdf](http://co2meters.com/Documentation/Other/AN_SE_0020_Sensor_Calibration_Procedure.pdf).
- [21] *Raspberry Pi 2 Model B*. [Kasutatud 2021-12-29]. URL: <https://www.raspberrypi.com/products/raspberry-pi-2-model-b/>.
- [22] *Raspberry Pi 4 Tech Specs*. [Kasutatud 2021-12-29]. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>.
- [23] *Beagleboard:BeagleBoneBlack*. [Kasutatud 2021-12-29]. URL: <https://elinux.org/Beagleboard:BeagleBoneBlack>.



- [24] IoTDunia.com. *Comparison between MQTT and HTTP protocols*. [Kasutatud 2021-11-25]. URL: <https://iotdunia.com/mqtt-and-http/>.
- [25] *Extensible Messaging and Presence Protocol (XMPP): Core*. [Kasutatud 2021-12-17]. URL: <https://datatracker.ietf.org/doc/html/rfc6120>.
- [26] *Difference Between AMQP vs MQTT*. [Kasutatud 2021-12-]. URL: <https://www.educba.com/amqp-vs-mqtt/>.
- [27] Steve Cope. *Beginners Guide To The MQTT Protocol*. [Kasutatud 2021-11-24]. URL: <http://www.steves-internet-guide.com/mqtt/>.
- [28] javatpoint. *MQTT protocol*. [Kasutatud 2021-11-24]. URL: <https://www.javatpoint.com/mqtt-protocol>.
- [29] *Comparison of MQTT Brokers*. [Kasutatud 2021-12-]. URL: <https://tewarid.github.io/2019/03/21/comparison-of-mqtt-brokers.html>.
- [30] telos Systementwicklung GmbH. *I2C-Bus: What's that?* [Kasutatud 2021-11-24]. URL: <https://www.i2c-bus.org/>.
- [31] SparkFun Electronics. *I2C*. [Kasutatud 2021-11-24]. URL: <https://learn.sparkfun.com/tutorials/i2c/all>.
- [32] Sparkfun. [Kasutatud 2021-11-26]. URL: [https://cdn.sparkfun.com/assets/learn\\_tutorials/8/2/I2C\\_Basic\\_Address\\_and\\_Data\\_Frames.jpg](https://cdn.sparkfun.com/assets/learn_tutorials/8/2/I2C_Basic_Address_and_Data_Frames.jpg).
- [33] *Semantic Model*. [Kasutatud 2021-12-30]. URL: <https://www.openhab.org/docs/tutorial/model.html>.

# **Lisa 1 - Koodirepositooriumid**

Kohaloleku sensor: <https://gitlab.cs.ttu.ee/raliis/k30>

Süsihappegaasi sensor: <https://gitlab.cs.ttu.ee/raliis/d6t>

## **Lisa 2 - Süsteemi demonstratsioon**

Valgustuse demonstratsioon:

<https://drive.google.com/file/d/1ISGyHBAIxDnI8o1yTrb0h42xSjwQhJh8/view?usp=sharing>

Õhukvaliteedi süsteemi demonstratsioon:

<https://drive.google.com/file/d/1PxP56pKwpymBiF-WH5MassPng2Tml68H/view?usp=sharing>