

Juho Ahoranta

OTA-KAMMIOIDEN TURVALLISUUS- JA INFORMAATIOJÄRJESTELMÄ

OTA-KAMMIOIDEN TURVALLISUUS- JA INFORMAATIOJÄRJESTELMÄ

Juho Ahoranta
Opinnäytetyö
Kevät 2021
Sähkö- ja automaatiotekniikan tutkinto-
ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma, sähkötekniikka

Tekijä: Juho Ahoranta

Opinnäytetyön nimi: OTA-kammioiden turva- ja informaatiojärjestelmä

Työn ohjaaja: Ensio Sieppi (Oamk), Pasi Kalliokoski (Nokia)

Työn valmistumislukukausi ja -vuosi: Kevät 2021 Sivumäärä: 26

Tämä opinnäytetyö tehtiin Nokia Oyj:lle. Työn tavoitteena oli valmistaa ratkaisu, jonka avulla OTA-kammion tila on helposti nähtävissä sekä etäkäyttäjille että laboratoriotilassa työskenteleville. OTA-kammiot ovat Orbis systemsin merikonteista muuntamalla valmistamia radiotaajuisten laitteiden testitiloja. Tällaisia testitiloja käytetään esimerkiksi 5G-radioiden testaukseen.

Opinnäytetyön tarkoituksena oli parantaa työturvallisuutta ja testattavien laitteiden käyttöturvallisuutta. OTA-kammiossa, josta työssä käytetään nimitystä kontti tai kammio, on valmiiksi turvalaitteisto, joka katkaisee testilaitteelta virran, jos ovi aukeaa. Turvalaitteen tarkoituksena on pitää kaikki radiotaajuinen säteily kontin sisällä sekä estää testilaitteiden vaurioituminen.

Ennen työn aloitusta ainoastaan kontin aktiivinen käyttäjä pystyi tietämään kontin sen hetkisen tilan. Työn tuloksena kontin tila on helposti nähtävillä kontin oven vieressä olevasta valokyltistä ja etänä verkkosivun kautta.

Asiasanat: Arduino IDE, OTA-kammio, Python, Radiotaajuinen säteily

ABSTRACT

Oulu University of Applied Sciences
Degree programme in electrical and automation engineering, electrical engineering

Author: Juho Ahoranta

Title of thesis: OTA chamber safety and information system

Supervisor(s): Ensio Sieppi (Oamk), Pasi Kalliokoski (Nokia)

Term and year when the thesis were submitted: Spring 2021

Number of pages: 26

The employer of this bachelor thesis was Nokia Oyj. There was a need for a safety system for OTA-chambers. The goal of this safety system was to make it easy for laboratory workers and remote workers to check the state of an OTA-chamber. OTA-chambers are test environments made by Orbis systems. These test environments are used for testing 5G-radios.

The device introduced in this thesis was made to improve the safety of laboratory workers and devices under test in OTA chambers. These OTA chambers are made from modified sea containers and are usually called chamber, sea container or SeaC. The chambers are already fitted with safety measures that should cut the power from device under test when the door is opened. The combination of these safety devices should keep all radio frequency radiation inside the chamber.

Before this thesis only the current user of OTA chamber knew the state of the device under test inside the chamber. Also, remote workers had no way of knowing if a door of the chamber was open and the only way to find out was to ask someone in the laboratory to check if a door is open.

Keywords: Device under test, OTA chamber, Radio frequency radiation

LYHENTEET

BTS	Base Station, kontissa sijaitseva tukiasema
EMC	Electromagnetic Compatibility, sähkömagneettinen yhteensopivuus
IDE	Integrated Development Environment, ohjelmointiympäristö, jolla suunnitellaan ja toteutetaan ohjelmia
IoT	Internet of Things, asioiden internet
LED	Light Emitting Diode, valoa lähettävä diodi
MDF	Medium Density Fibreboard, puolikova puusta valmistettu kuitulevy
OTA	Over The Air, ilmateitse tapahtuva tiedonsiirto
OTAVA	Over The Air Validation Area, Ruskossa sijaitseva Nokian kehitys- ja tutkimuskeskus
PDU	Power Distribution Unit, etäohjattava 16-paikkainen virtalähde testilaitteille
RF-säteily	Radiotaajuinen säteily
Seac	Sea container, muokattu merikontti, jonka sisällä testilaitteet sijaitsevat.
Skripti	Ohjelma, jota ei tarvitse kääntää ennen suorittamista
TCP	Transmission Control Protocol, tiedonsiirtoprotokolla
WiFi	Langaton tiedonsiirto-protokolla, joka perustuu IEEE 802.11 standardeihin

SISÄLLYS

1	JOHDANTO	7
2	TOIMEKSIANTAJAN ESITTELY	9
3	YMPÄRISTÖ JA LAITTEET	10
3.1	OTA RF -kammio	10
3.2	NodeMCU.....	11
3.3	Raspberry pi.....	11
3.4	Arduino IDE.....	12
3.5	REST.....	13
3.6	Node.js	14
3.7	Python	14
3.8	Telnet	15
4	TOIMINTA	16
4.1	Kokonaiskuva	16
4.2	Infobox	17
4.3	Control server.....	19
4.4	Tukiaseman etäohjattava virtalähde	21
5	JATKOKEHITYS.....	23
6	YHTEENVETO	24
	LÄHTEET.....	25

1 JOHDANTO

Opinnäytetyön tarkoituksena oli tuottaa, testata ja asentaa OTA-kammioille turvallisuus- ja informaatiojärjestelmä. OTA-kammiot ovat pääasiassa 5G-radioiden testaukseen valmistettuja testitiloja. Testitilat on eristetty radiotaajuista säteilyä vaimentavalla materiaalilla, joka mahdollistaa useiden tällaisten testitilojen olemisen lähekkäin. Järjestelmän tehtävä on kertoa kammion sen hetkinen tila laboratoriotilassa oleville ja etäkäyttäjille. Tällä hetkellä ainoastaan kammiota käyttävä testaaja tietää, mitä kammion sisällä tapahtuu, mutta etäkäyttäjällä ei ole mahdollisuutta tietää, onko kammion etu- tai takaovi auki. Informaatiojärjestelmän avulla sekä laboratoriotilassa olevat että etäkäyttäjät tietävät kammion sen hetkisen tilan helposti.

Opinnäytetyö tehtiin Nokian tarpeesta tuoda kaikkien OTA-kammioiden käyttöturvallisuus samalle tasolle. Kuuden metrin yksiovisissa kammioissa on Orbis systemsin testilaitteen suuntausjärjestelmä, jonka ohjauslaitteeseen kuuluu ”liikennevalo”, jonka avulla kontin tila on helposti nähtävissä. Tämä ratkaisu on kuitenkin hyvin kallis ja sen avulla ei ole nähtävissä, onko kammion ovi auki. Valmiin ratkaisun hinta on noin 30 000 euroa. Sen hankkiminen useisiin kammioihin tulisi hyvin kalliiksi. Opinnäytetyössä esitelty ratkaisu on huomattavasti kustannustehokkaampi ja sen avulla saadaan seitsemän metrin kaksiovisiin kammioihin lisäksi toiminto, jolla näkee myös, ovatko kammion molemmat ovet kiinni.

Tässä opinnäytetyössä esitetyssä ratkaisussa kammion tila esitettäisiin liikennevalojen tapaan värikoodein. Valon ollessa vihreä kammion virta on katkaistu ja oven avaaminen on turvallista. Punaisen valon palaessa virta on päällä ja ovi tulee pitää kiinni. Punaisen valon vilkkuessa joko testitila on aktiivinen tai testattavalle laitteistolle ollaan tekemässä ohjelmistopäivitystä. Punaisen valon vilkkuessa on erityisen tärkeää, että ovia ei avata. Jos virta katkeaa kesken testauksen, joutuu testaaja suorittamaan testauksen kokonaan uudelleen ja pahimmassa tapauksessa testilaitteisto vaurioituu. Sama pätee myös ohjelmistopäivitykseen.

Ensimmäisen kerran keskustelut esimiestasolla tämän tyyppisestä järjestelmästä heräsivät vuoden 2020 alkupuolella OTA-kammioiden turvallisuuskatsauksessa. Siinä huomautettiin 7 metrin kammioiden ”liikennevalon” puutteesta. Järjestelmän tarpeellisuus kuitenkin kyseenalaistettiin OTA-kammion automaattisen sammutusjärjestelmän olemassaolon vuoksi.

Aluksi oli kartoitettava, mitä laitteistoa ja ohjelmia kyseisten ominaisuuksien toteuttamiseen tarvittiin. Työprosessin aikana oli tutustuttava näiden laitteiden ja ohjelmien toimintaan. Suurin osa käytetyistä lähteistä on englanniksi. Työssä käytettyjä lähteitä ovat esimerkiksi Arduino IDE käyttöohje, Enics PDU power breaker communication spec ja OTA RF chamber technical specification.

Sähköturvallisuuden perusvaatimukset on annettu sähköturvallisuuslaissa ja se koskee yleistä sähköturvallisuutta ja EMC:tä. Sähköturvallisuuslain 6. § koskee näitä perusvaatimuksia.

Sähkölaitteet ja -laitteistot on suunniteltava, rakennettava, valmistettava ja korjattava niin sekä niitä on huollettava ja käytettävä käyttötarkoituksensa mukaisesti niin, että:

- 1) niistä ei aiheudu kenenkään hengelle, terveydelle tai omaisuudelle vaaraa;*
- 2) niistä ei sähköisesti tai sähkömagneettisesti aiheudu kohtuutonta häiriötä;*
- 3) niiden toiminta ei häiriinny helposti sähköisesti tai sähkömagneettisesti.*

Jos sähkölaite tai -laitteisto ei täytä 1 momentissa säädettyjä edellytyksiä, sitä ei saa saattaa markkinoille, luovuttaa toiselle eikä ottaa käyttöön. (1.)

2 TOIMEKSIANTAJAN ESITTELY

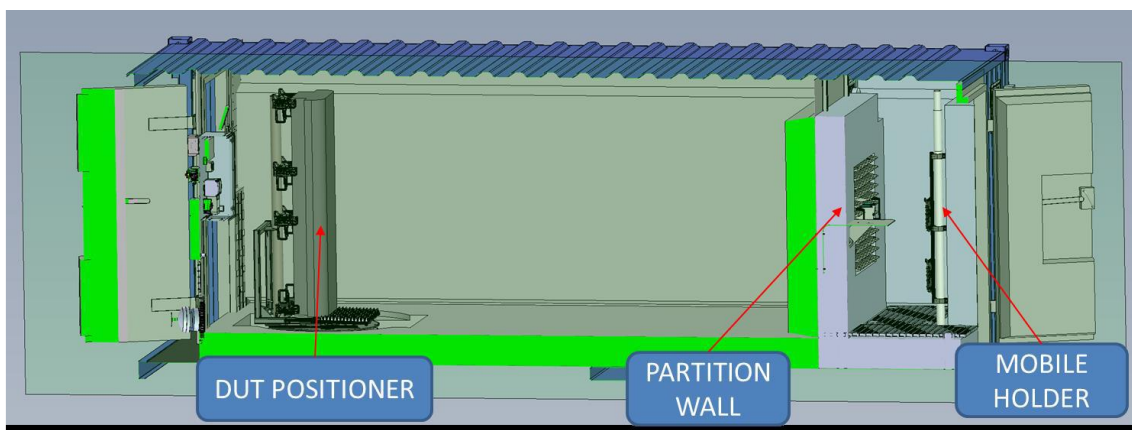
Nokia Oyj on suomalainen telekommunikaatio- ja informaatioteknologian yritys, jonka pääkonttori sijaitsee Espoossa. Nokia työllistää noin sata tuhatta työntekijää yli sadassa eri maassa ja sen liikevaihto vuonna 2019 oli 23 miljardia euroa. Nokia tunnetaan parhaiten sen 2000-luvun alun matkapuhelimien valmistuksesta ja myynnistä. Vuonna 2014 matkapuhelinbisnes myytiin Microsoftille ja sen jälkeen Nokia on keskittynyt patenttien lisensointiin, IoT-tekniologioihin ja telekommunikaatioinfrastruktuuriin. Ruskossa sijaitseva OTAVA on Nokian tutkimus- ja kehitysyksikkö, jossa testataan ja integroidaan 4G- ja 5G-tekniologiaa. (2.)

3 YMPÄRISTÖ JA LAITTEET

Tässä luvussa on esitelty pääpiirteittäin työssä käytetyt tärkeimmät ympäristöt ja laitteet. Luvun tarkoituksena on auttaa lukijaa ymmärtämään, minkälaisia teknologioita ratkaisun tuottamiseen on tarvittu.

3.1 OTA RF -kammio

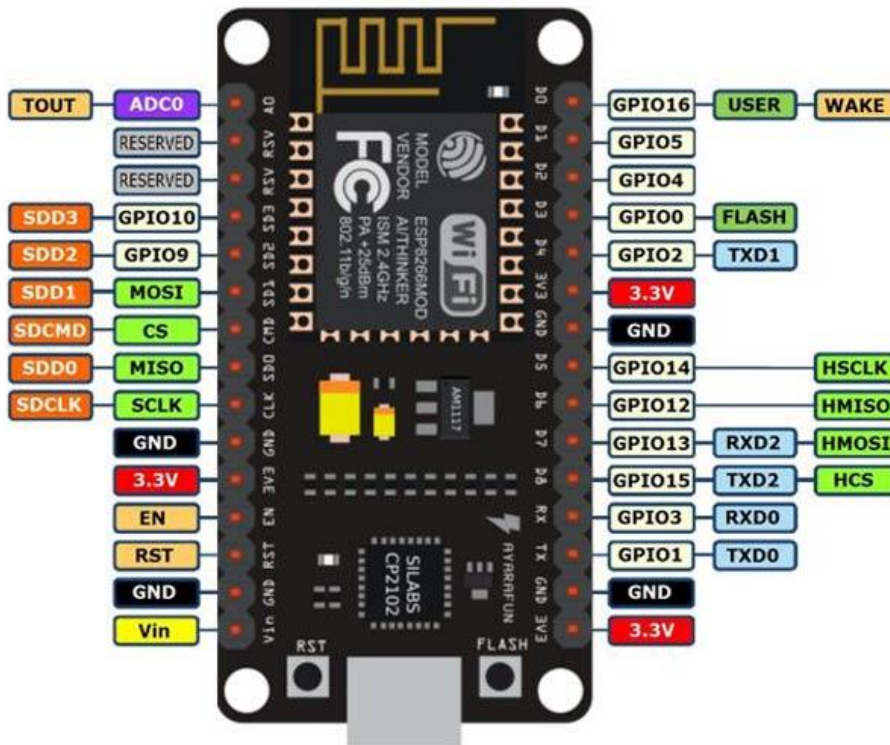
Ympäristö, johon järjestelmä tuotetaan ja asennetaan, on Orbis systemsin valmistama OTA RF kammio 7m. Kammio on valmistettu merikontista muuntamalla. Kontteja käytetään 4G- ja 5G-radioiden testaamiseen. Kontin sisäseinät on vuorattu radiotaajuista säteilyä absorboivalla materiaalilla. Materiaalit on suunniteltu siten, että ne poistavat kaikua eikä radiotaajuinen säteily pääse vuotamaan tilan ulkopuolelle. Tämän ansiosta kontit voivat olla vierekkäin ilman, että ne tuottavat häiriötä toisilleen. OTA-kammion ovi avautuu kääntämällä ovesta oleva lukitusvarsi yläasentoon. Lukitusapit ovat hahloissa ja laskeutuessaan alaspäin ne avaavat ovea hieman. Kontin oveen on liitetty katkaisija, joka katkaisee virran kontin sisällä olevasta testilaitteesta, kun ovi avataan. Kuvassa 1 on esitetty merikontin läpileikkaus. Ennen työn aloittamista konttien oville puuttui sähkönsyöttö. Tässä opinnäytetyössä esitettyä ratkaisua varten kammion ulkopuolelle etu- ja takaovien viereen asennettiin pistorasiat, joista laitteet saavat virtaa. Pistorasioiden asentamiseksi piti tehdä Jiraan pyyntö lab support -tiimille, jonka avulla laitteelle saatiin pysyvä virransyöttö. (3, s. 4–8.)



KUVA 1. OTA RF kammio (3, s.1)

3.2 NodeMCU

NodeMCU on avoimen lähdekoodin IoT alusta, joka on rakennettu Espressif systemsin valmistaman ESP8266 12F-moduulin ympärille. ESP8266 sisältää tietokoneen peruselementit eli prosessorin, muistin ja langattoman verkkoyhteyden. NodeMCU on lähes samanlainen laite kuin arduino. Uudet kirjastot päivittämällä se on yhteensopiva Arduino IDE:n kanssa. Arduino IDE:tä käytetään ohjelmien kirjoittamiseen C++ kielellä ja lataamiseen laitteelle. ESP8266 ja Arduino IDE ovat hyvin suosittuja, joten niille on myös helppo löytää ohjeita verkosta. Kuvassa 2 on esitetty NodeMCU:n liitinkaavio. Kun ohjelma on ladattu laitteelle, laite toteuttaa sitä niin kauan kuin laitteeseen syötetään virtaa. (4.)



KUVA 2. NodeMCU liitinkaavio (5.)

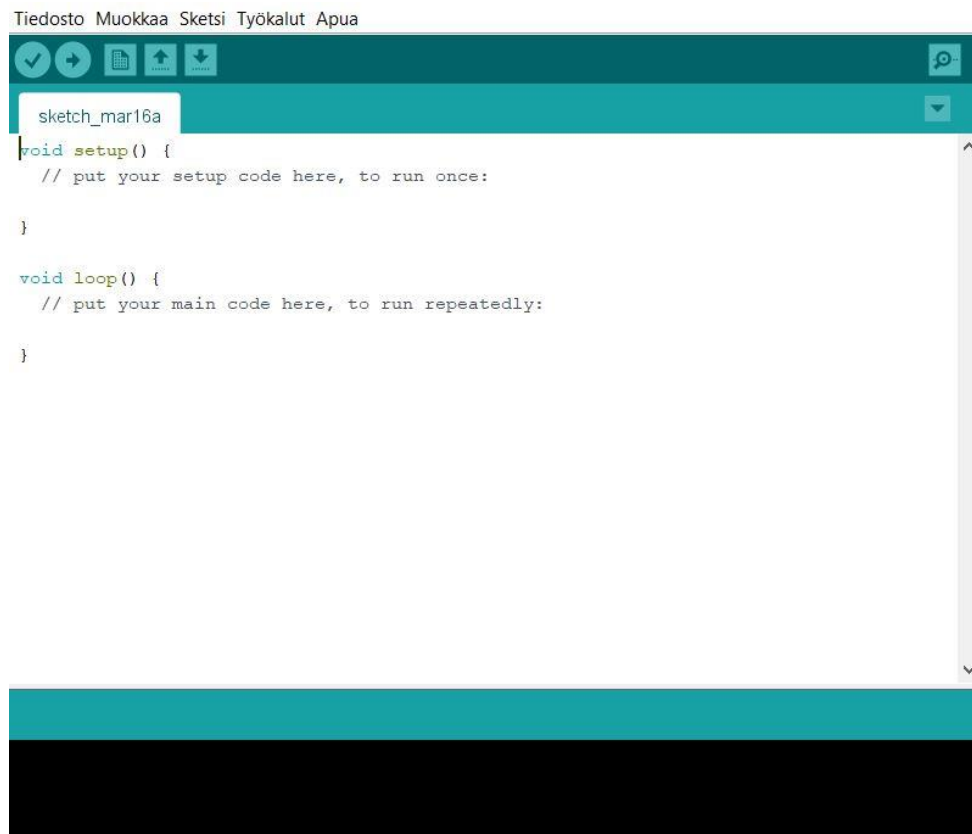
3.3 Raspberry pi

Raspberry pi on brittiläisen Raspberry pi foundationin ja Broadcomin yhteistyössä kehittämä yhden piirilevyn tietokone. Käytännössä se sisältää kaikki samat komponentit kuin normaali tietokone mutta taskukoossa. Alun perin raspberry pi:t oli tarkoitettu kouluhin ohjelmoinnin opetukseen, mutta ne nousivat yllättävään suosioon ja nyt niitä käytetään kaikkialla robotiikasta säämittauksiin.

Työssä käytettävä raspberry pi on Raspberry pi 4 model B, joka on viimeisin julkaistu versio laitteesta. Raspin mukana tulee Raspbian linux-pohjainen käyttöjärjestelmä esiasennettuna SD muistikortille. Tässä työssä raspi toimii vain välittäjänä infoboxin ja laboratorioverkon välillä. (6.)

3.4 Arduino IDE

Arduino IDE, eli integrated development environment, on ohjelma, jolla voidaan ohjelmoida arduino käyttämällä C++ ohjelmointikieltä muistuttavaa ohjelmointikieltä. Ohjelmaa voidaan käyttää myös minkä tahansa yhteensopivan mikrokontrollerin kanssa kuten työssä käytetyn NodeMCU:n. Arduino IDE on saatavilla Mac-, Windows- ja Linux-käyttöjärjestelmille ilmaiseksi Arduinon omalta verkkosivulta. Kuvan 3 kaltainen näkymä aukeaa, kun Arduino IDE käynnistetään.



KUVA 3. Uuden sketsin näkymä

Ohjelma sisältää tekstieditorin koodin kirjoittamiselle, viestikentän vikailmoituksille ja työkalurivin, josta löytyy painikkeita yleisille toiminnoille ja valikoille. Arduino IDE:llä kirjoitettuja koodeja kutsutaan sketsiksi ja tallennetun sketsin päätte on .ino. Tekstieditori sisältää perustoimintoja kuten teks-

tin kopioinnin ja liittämisen sekä tekstihaun. Oikeassa alalaidassa myös näkyy kytketty laite ja kytketyn laitteen sarjaportin numero. Työkalurivillä ovat painikkeet koodin varmistamiseen, lähettämiseen, avaamiseen, uuden luomiseen ja sarjaportin monitorin avaamiseen. Varmistuspainikkeesta painamalla ohjelma tarkistaa koodin virheiden varalta ja ilmoittaa niistä viestikentässä. Myös lähetä-painike varmistaa koodin, mutta se myös lähettää koodin valitulle laitteelle. Tästäkin tulee ilmoitus viestikenttään. Työssä käytetyn NodeMCU:n sarjaportin nimi on yleensä COM3. (7.)

3.5 REST

Web-rajapinnan toteuttamiseen on olemassa monia eri web-teknologioita. Tässä luvussa esitellään työssä käytetty REST-arkkitehtuuri. REST on lyhenne ja tulee sanoista representational state transfer. Restillä on kuusi pääpiirrettä, joilla se määritetään. Nämä kuusi pääpiirrettä ovat asiakas ja palvelin, tilattomuus, välimuistin käyttö, yhtenäinen rajapinta, kerroksittainen järjestelmä ja ladattava koodi. Näiden rajoitteiden tarkoituksena on taata toimivuus eri resurssien välillä.

REST kokonaisuus voidaan siis jakaa kuuteen osaan. Ensimmäinen osa on asiakas ja palvelin. Näiden kahden selkeä jako mahdollistaa molempien komponenttien itsenäisen kehittymisen. Palvelimen tehtävä on viestien käsittely ja asiakkaan tehtävä on viestien lähettäminen oikeassa muodossa ja vastausten käsittely.

Toinen osa on tilattomuus. Tilattomuudella tarkoitetaan sitä, että palvelimelle ei jää tietoa suoritetusta kyselystä. Kaikkien asiakkaalta palvelimelle suoritettujen kyselyiden tulee sisältää kaikki tarvittavat tiedot operaation suorittamiseen.

Kolmas kohta on välimuistin käyttö. Välimuistin käytön tarkoituksena on parantaa verkon tehokkuutta. Kyselyille voidaan antaa lupa tallentua asiakkaan välimuistiin ja samaa dataa voidaan käyttää myöhempisiin kyselyihin.

Neljäs kohta on yhtenäinen rajapinta. Se mahdollistaa useiden eri asiakkaiden saman resurssin käytön. Resurssien erottaminen muista palveluista mahdollistaa yksittäisten komponenttien itsenäisen kehittymisen. Yhtenäisen rajapinnan huonona puolena on tehokkuuden laskeminen, kun data liikkuu standardoidussa muodossa, joka ei ole optimaalinen tietyn käyttäjän tarpeisiin. Yhtenäisen

rajapinnan saavuttamiseksi tarvitaan neljä lisärajoitetta, joilla ohjataan komponenttien toimintaa. Neljä lisärajoitetta ovat resurssien tunnistaminen, resurssien manipulointi esitysten kautta, itseselitteiset viestit ja hypermedia tilakoneena.

Viidentenä kohtana on kerroksittainen järjestelmä. Sen avulla luodaan kerroksittainen järjestelmä, jossa yksittäinen komponentti ei näe muuta kuin seuraavan kerroksen. Kerroksia voivat olla esimerkiksi palomuurit ja välityspalvelimet. Näitä kerroksia voidaan käyttää myös järjestelmän skaalamiseen tasapainottamalla kuormitusta useiden verkkojen ja prosessorien välille. Dataa voidaan myös muokata jokaisessa kerroksessa erikseen ja sitten lähettää se seuraavaan kerrokseen.

Kuudentena on ladattava koodi. Sen avulla voidaan ladata ja suorittaa koodia skriptien avulla. Tämä yksinkertaistaa asiakasohjelmia vähentämällä ennalta määrättyjä ominaisuuksia. Ladattava koodi on kuitenkin vain valinnainen rajoite. (8.)

3.6 Node.js

Node.js on asynkroninen tapahtumapohjainen järjestelmä, joka on tarkoitettu skaalattavien verkkoapplikaatioiden tekemiseen. Tämä tarkoittaa sitä, että useita käskyjä voidaan suorittaa samanaikaisesti eivätkä ne estä toisten käskyjen suorittamista. Kun käskyt eivät vaikuta toisiinsa, ei ohjelma myöskään voi jäädä jumiin. Node.js käyttää event loop-järjestelmää, joka perustuu Rubyn event machineen ja Pythonin twistediin. (9.)

Npm eli node package manager on ohjelma, jolla on kaksi käyttötarkoitusta. Se on säilytyspaikka avoimen lähdekoodin Node.js projekteille ja se on myös komentorivityökalu, jota käytetään paketien asentamiseen ja versiohallintaan. Yksi tärkeä osa npm:ää on se, että paketteja asennettaessa npm pitää automaattisesti riippuvuussuhteen kunnossa. (10.)

3.7 Python

Python-kieli on oliopohjainen korkean tason ohjelmointikieli. Se on hyvin yksinkertainen oppia ja siitä syystä myös hyvin käytetty kieli. Python tukee moduuleita ja paketteja, jonka vuoksi ohjelmat

voidaan suunnitella modulaarisiksi ja koodin osia voidaan käyttää monissa projekteissa. Kun moduuli on tehty, sen voi helposti skaalata tuotavaksi muihin projekteihin. Python-tulkki ja peruskirjastot ovat ilmaisia ja saatavilla kaikille alustoille, mikä tekee siitä hyvin houkuttelevan vaihtoehdon.

Python on hyvin käytännöllinen ohjelmointikieli ja sen avulla voidaan ohjelmoida monenlaisia käytännöllisiä ohjelmia. Python on tulkattava kieli, joka tarkoittaa sitä, että sitä ei käännetä tietokoneelle ymmärrettävään muotoon käynnistyksen yhteydessä. Tällaisia kieliä kutsutaan yleisesti skriptauskieliksi. Pythonia voidaan käyttää prosessoimaan tekstiä, näyttämään numeroita tai kuvia tai laskemaan laskuja ja tallentamaan dataa. Sitä käytetään siis prosessoimaan dataa taustalla. (11.)

3.8 Telnet

Telnet on verkkoprotokolla, jolla voidaan olla yhteydessä muihin laitteisiin komentoikkunan kautta. Telnetiä käytetään yleensä laitteiden etäkäyttöön ja monitorointiin. Alun perin telnetiä käytettiin terminaaleissa. Näiden koneiden käyttöön riitti pelkkä näppäimistö, sillä kaikki näytöllä oli pelkkää tekstiä. Nykyään telnetiä käytetään virtuaaliterminaalissa, joka on nykyaikaisen tietokoneen tapa kommunikoida käyttäen telnet-protokollaa. Kaikissa tietokoneissa telnet ei ole oletuksena käytössä ja se pitää silloin käydä erikseen ottamassa käyttöön asetuksista.

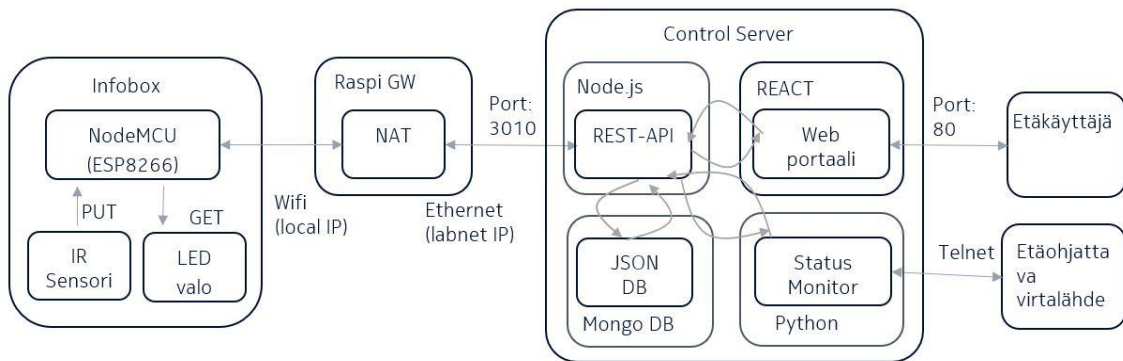
Telnet ei ole kuin muut Transmission Control Protocol (TCP) -yhteyttä käyttävät protokollat kuten HTTP, joka siirtää tiedostoja serverin ja pääteen välillä. Sen sijaan asiakaspuolella käytetään telnet-ohjelmaa, jolla otetaan yhteys isäntäkoneeseen. Kun yhteys on luotu, voidaan asiakkaan koneelta lähettää komentorivi komentoja suoritettavaksi isäntäkoneelle. Telnet-yhteydellä saadaan siis suora kontrolli isäntäkoneesta. (12.)

4 TOIMINTA

Tämän luvun tarkoituksena on antaa lukijalle hyvä yleiskuva ratkaisun eri osien toiminnasta. Luku on jaettu neljään alalukuun, jotka ovat kokonaiskuva, infobox, control server ja tukiaseman etäohjattava virtalähde. Jokaisessa alaluvussa käydään läpi kyseisen osion toiminta.

4.1 Kokonaiskuva

Laitteen toiminnan kokonaiskuva on kuvattuna kuvassa 4.



KUVA 4. Laitteen toimintakaavio

Laitteen toiminta voidaan jakaa neljään osaan. Osat ovat infobox, raspberry pi, control server ja BTS power breaker eli tukiaseman etäohjattava virtalähde. Infoboxit sisältävät NodeMCU:n, infrapunasensorin ja LED-valon ja ne ovat WiFi-yhteyden avulla yhteydessä Raspberry pi:hin. Raspberry pi:n avulla infobox saa yhteyden laboratorioverkkoon. Control server on järjestelmän aivot, joka kerää dataa tietokantaan, jonka perusteella LED-valon väri vaihtuu.

Infobox tarvitsee WiFi-yhteyden Raspberry pi:hin, joka on yhteydessä laboratorioverkossa olevaan serveriin, saadakseen serveriltä tilatiedon. Tilatiedon saatuaan infoboxissa oleva LED-valo joko vaihtuu tai pysyy samana riippuen kontin tilasta. Control server sisältää 4 funktiota, jotka ovat REST-API, JSON-tietokanta, status monitor python skripti ja webportaali. Status monitor monitoroi BTS power breakeria eli tukiaseman etäohjattavaa virtalähdettä ja lähettää saamansa datan tietokantaan.

4.2 Infobox

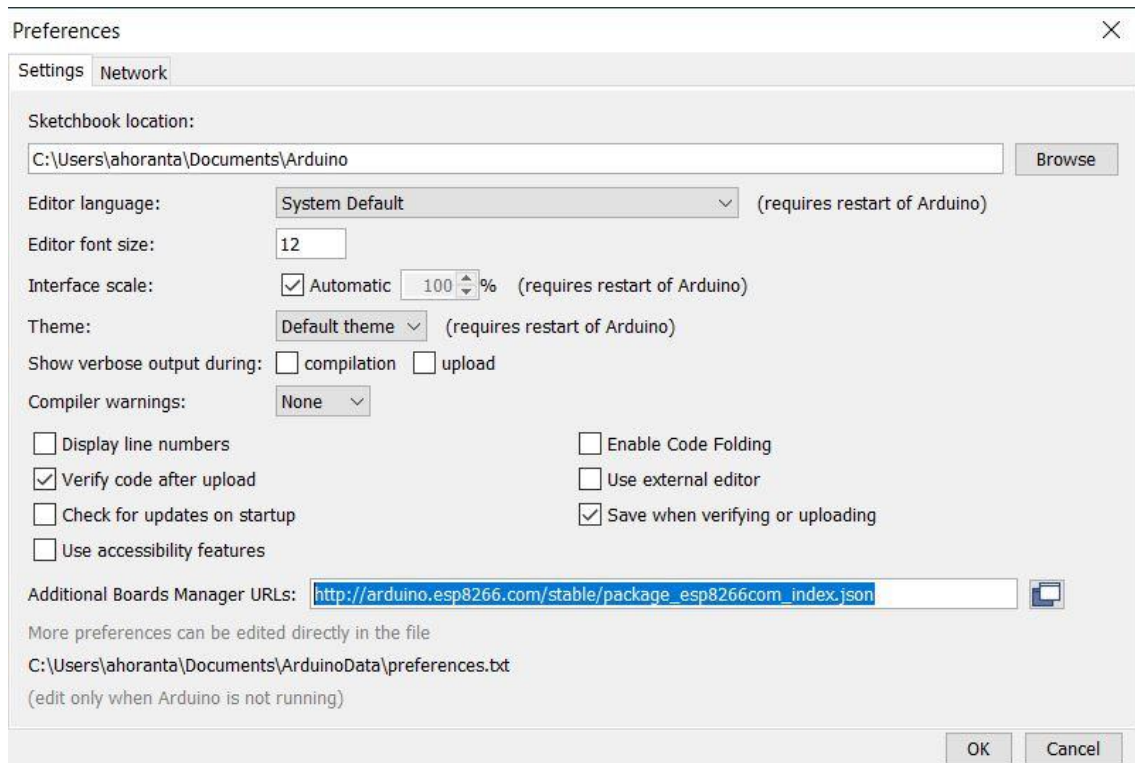
Järjestelmän fyysinen osa on Infobox, jonka tehtävä on antaa visuaalinen representaatio kontin sen hetkisestä tilasta. Tämä tuotetaan LED-valolla, jonka värit vastaavat control serveriltä saatua tietoa. Itse laatikko, jonka sisälle osat tulevat, on tehty laser-leikatusta MDF-levystä. Leikatuista palasista koottiin palapelin tavoin laatikko. Laatikon alareunassa on lovi, josta tulee läpi akryylilevy. LED-valo on suunnattu akryylilevyn siten, että se kulkee ylhäältä alaspäin levyn läpi, jolloin se valaisee akryylilevyssä olevat kaiverrukset. Myös infrapunasensorin kotelo on valmistettu laser-leikatuista mdf-levyn palasista. Kuvassa 5 on nähtävillä järjestelmän fyysinen laite kokonaisuudessaan.



KUVA 5. Infobox ja infrapunasensori kytkettynä virtalähteeseen

Ennen kuin NodeMCU oli ohjelmoitavissa Arduino IDE:llä piti siihen päivittää uudet kirjastot ja Arduino IDE:hen piti lisätä uusi piirilevy. Uusien piirilevyjen lisääminen Arduino IDE:lle on hyvin yksinkertainen prosessi, varsinkin kun kyseessä on tunnettu alusta. NodeMCU:n lisääminen Arduino IDE:lle tapahtuu siten, että Arduino IDE:n ylävalikosta valitaan kohta "File" ja siitä avautuvasta valikosta valitaan kohta "Preferences", se voidaan myös tehdä painamalla näppäinyhdistelmää control ja pilkku. Kuvassa 6 on esitetty preferences-näkymä. Näkymän alalaidassa on kohta "Addition boards manager URLs", jonne lisätään json linkki, joka sisältää monien ESP8266 piirilevyjen tiedot. Tämän jälkeen Arduino IDE:n ylävalikosta valitaan kohta "Tools" ja siitä avautuvasta valikosta kohta "Manage libraries", se voidaan myös tehdä näppäinyhdistelmällä control, shift ja I. Näin saadaan

avattua näkymä, josta voi hakea hakusanoilla projektiin tarvittavia kirjastoja. Täältä haetaan "esp8266 by ESP8266 community"-niminen kirjasto. Kirjasto asennetaan yksinkertaisesti painamalla ruutuun ilmestyvää install-painiketta. Näiden vaiheiden jälkeen Arduino IDE tunnistaa NodeMCU:n ja asettaa sille automaattisesti sarjaportin COM3. Kun Arduino IDE:llä ladataan ohjelma NodeMCU:lle, tulee Arduino IDE:n viestikenttään punaisella teksti "hard resetting via RTS pin". Tämä on ihan normaalia ja tarkoittaa, että ohjelma on onnistuneesti ladattu NodeMCU:lle.



KUVA 6. Arduino IDE preferences-näkymä

Työssä käytettiin 3–5 voltin tasavirralla toimivaa säädettävää infrapunasensoria. Sensori on liitetty NodeMCU:hun kolmella johtimella. Sensorissa on kolme liitintä, jotka ovat out, gnd ja vcc. Sensorin out-liitin on liitetty mikrokontrollerin D1-liittimeen, gnd-liitin on liitetty gnd-liittimeen ja vcc on liitetty 3.3V-liittimeen (kuva 2).

Infrapunasensorin tehtävänä on monitoroida OTA-kammion oven lukitustapin asentoa. Oven lukitustapin ollessa yläasennossa sensorin havaitsee sen ja laite tulkitsee oven olevan lukossa. Jos taka- tai etuoven lukitustappi ei ole yläasennossa, tulkitsee laite oven avonaiseksi, jolloin molempien ovien LED-valot vaihtuvat keltaisiksi. Keltainen valo on merkki ainakin yhdestä avonaisesta ovesta. Oven ollessa auki virransyöttö OTA-kammiossa olevalle testilaitteelle katkeaa eikä testilinjalla voi silloin suorittaa testausta.

Kun infobox liitetään virtalähteeseen, se alkaa toteuttamaan NodeMCU:hun ladattua ohjelmaa. Ensimmäisenä ohjelma testaa LED-valon toimivuuden asettamalla punaisen, vihreän ja sinisen värin maksimiin tuottaen valkoisen valon. Valotestin jälkeen laite yrittää luoda WiFi-yhteyden valittuun IP-osoitteeseen. Tässä tapauksessa kyseinen IP-osoite on välittäjänä toimivan raspberry pi:n IP-osoite. Kun yhteys on luotu onnistuneesti, alkaa laite kuuntelemaan REST-API:lta tulevaa tietoa ja samalla lähettämään infrapunasensorilta saatua tietoa oven tilasta. REST-API:lta tuleva tieto tulee infoboxille JSON-formaatissa, jonka ohjelma kääntää laitteen ymmärtämään muotoon. Kuvassa 7 on esitetty pieni osa LED-valoja ohjaavaa ohjelmaa.

```
//LED-Neopixel setup
strip.begin();           // INITIALIZE NeoPixel strip object (REQUIRED)
strip.show();           // Turn OFF all pixels ASAP
strip.setBrightness(50); // Set BRIGHTNESS to about 1/5 (max = 255)

//Showing "PowerON" by White LED color
colorWipe(strip.Color( 255, 255, 255), 100); //Swipe LED WHITE
```

KUVA 7. Käynnistyksen yhteydessä tapahtuva valkoisen valon syttyminen

4.3 Control server

Control serveriksi kutsuttu osio sisältää JSON-tietokannan, REST-API:n, status monitor python skriptin ja web portaalin. Sen tärkeimmät osat ovat REST-API ja python skripti. Control server siis sisältää suurimman osan laitteen niin sanotuista liikkuvista osista.

Status monitor on python skripti, joka tarkastelee tukiaseman etäohjattavalta virtalähteeltä tulevaa virtatietoa. Tätä virtatietoa verrataan ennalta määrättyyn lukuarvoon ja sen perusteella status monitor päättää, kulkeeko tukiasemalle virtaa vai ei. Tässä tapauksessa lukuarvoksi on asetettu 1 ampeeri. Kun tukiasemalle kulkevan virran määrä on 1 ampeeri tai enemmän, skripti tulkitsee virran olevan päällä ja infoboxin LED-valon väri muuttuu punaiseksi. Yksi skripti ohjaa molempia infoboxeja samanaikaisesti (kuva 8).

```
Power breaker: #4 of 172.31.1.104
PowerBreaker Status: red
Current value: 6.0(A)
Infobox #: ['3', '4']
Door Status: ['locked', 'locked']
LED status of boxes: ['red', 'red']
LED status of OTA: red
```

KUVA 8. Status monitor skriptin tuloste

Status monitor skripti avaa telnet-yhteyden tukiaseman etäohjattavalle virtalähteelle. Telnet-yhteyden kautta voi hakea virtalähteeltä monia eri tietoja, mutta tässä työssä ainoa relevantti informaatio oli kanavan numero 4 virtatieto. Skriptiä käynnistettäessä annetaan halutun etäohjattavan virtalähteen IP-osoite, halutun virtakanavan numero ja sitä vastaavan OTA-kammion numero.

Skripti monitoroi myös infoboxien infrapunasensoreita. Yhdistämällä virtatiedon ja infrapunasensoreiden tiedon skripti päättää, minkä värinen valo pitää olla päällä. Jos molemmat ovet ovat kiinni ja virta on alle yhden ampeerin, muuttuu "LED status of OTA: red" muotoon "LED status of OTA : green". Tämän jälkeen skripti lähettää "changesignal" komennon REST-API:lle, joka vaihtaa JSON datan "red" kohdan kohdaksi "green". Tämä viesti kulkee infoboxille, joka vaihtaa LED-valon värin punaisesta vihreäksi.

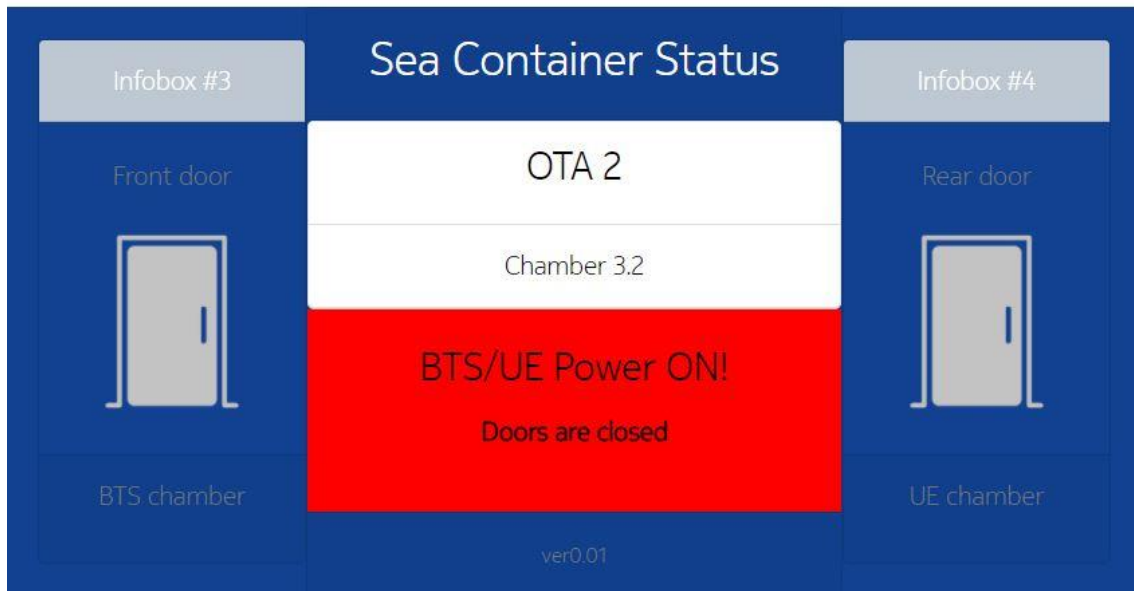
REST-API toimii rajapintana serveripuolen ja fyysisen laitteen välillä sekä muiden serveripuolen osien välillä. Sen tehtävä on kuljettaa tietoa järjestelmän eri osien välillä. Eri osien välillä kulkeva tieto on JSON-muodossa. Käytettävät metodit ovat POST, GET, PUT ja DELETE, jotka on esitetty kuvassa 9.

1. **[POST] (server IP):3010/api/(InfoBox ID#)/**
Create new JSON data set for Infobox ID#
2. **[GET] (server IP):3010/api/(InfoBox ID#)/**
Response JSON data stream of infobox ID#
3. **[PUT] (server IP):3010/api/(InfoBox ID#)/**
Update JSON data stream of infobox ID#
4. **[Delete] (server IP):3010/api/(InfoBox ID#)/**
Delete JSON data stream of infobox ID#
5. **[GET] (server IP):3010/ api/(InfoBox ID#)/changesignal/(green/yellow/red)**
Change "signal:" status of Infobox ID# from green/yellow/red
6. **[GET] (server IP):3010/ api/(InfoBox ID#)/changedoor/(unlock/locked)**
Change "door:" status of Infobox ID# from unlock or locked

KUVA 9. REST-API:n käyttämät metodit

REACT web portaalin kautta voi tarkastella kontin tilaa etänä. Web portaalista on helposti nähtävissä etuoven ja takaoven tila erikseen ja testilaitteen virtatieto. Kuvassa 10 on esitetty web portaalin näkymä, kun ovet ovat kiinni ja testilaite on päällä.

OTAVA Sea Container infobox system



KUVA 10. Web portaalin näkymä ovet kiinni ja virta päällä

Vasemmalla puolella on etuovi ja etuovelle asennetun infoboxin indeksinumero. Keskellä on OTA-kammion numero, testilaitteen virtatieto ja sitä vastaavan LED-valon väri. Jos toinen kontin ovista aukeaisi, näkyisi kyseinen ovi web portaalissa avonaisena ja punainen väri vaihtuisi keltaiseksi.

4.4 Tukiaseman etäohjattava virtalähde

Tukiaseman etäohjattava virtalähde on Enics PDU power breaker 48-16. Se on 16-paikkainen ethernetin yli ohjattava 48VDC virtalähde. Kaikki 16 lähtöä ovat johdonsuojälähtöjä ja jokainen lähtö on erikseen vaihdettavissa ja ohjattavissa. Virtalähdettä voi ohjata myös web käyttöliittymän kautta, mutta se on rajoitettu vain laboratorioverkon sisäpuolelle. Tässä työssä etäohjattavan virtalähteen monitorointi tapahtui telnet-yhteyden kautta. Yleisimmin etäohjattavaa virtalähdettä käytetään testilaitteiden sammuttamiseen ja uudelleen käynnistämiseen esimerkiksi vikatilanteissa. (13.)

Virtalähteen etäohjaukseen ja monitorointiin ethernet-yhteyden ylitse käytetään IEEE448-standardin mukaisia komentoja. Jos komennossa yksikin merkki on väärin, ei laite palauta mitään viestiä.

Komennoilla voidaan vaihtaa jokaisen kuudentoista eri lähdön nimi tai tila. Etäohjaukselle ei tässä työssä ollut mitään tarvetta, joten pelkkä monitorointi riitti. Etäohjattavasta virtalähteestä olisi voinut monitoroida monia asioita kuten jännitettä tai lämpötilaa, mutta työn kannalta ainoa tärkeä tieto oli virtatieto. Kuvassa 11 on esitetty esimerkkikomento virtatiedon hankkimiselle, selitys komennolle ja esimerkkivastaus.

Command	Description and Parameters	Response
CURRENT:<channel>?	Current measurement for selected output channel.	CURRENT:<channel>:<value>

KUVA 11. Virtatiedon hankkimiseen käytettävä komento (14, s. 6)

5 JATKOKEHITYS

Työssä esitetyllä ratkaisulla voidaan tarkastella yhden kontin tilaa. Tulevaisuudessa tällainen laite pitäisi saada asennettua yhteensä kahdenkymmenen OTA-kammion etuoveen ja takaoveen. Useiden tällaisten yksiköiden valmistaminen ja asentaminen on aika helppoa, mutta serveripuolelle on tehtävä joitain muutoksia järjestelmän toimimiseksi. Vastuu serveripuolen jatkokehityksestä tukemaan ainakin 40 infobox-yksikköä on annettu yrityksen sisällä henkilöille, joilla on aiheesta enemmän kokemusta.

Opinnäytetyön tekemisen aikana rakennettiin yhteensä 5 toimivaa yksikköä. Loppujen 35 yksikön kustannusarviot on esitetty taulukossa 1. Osat loppuihin yksiköihin on jo tilattu ja ne kootaan kunhan osat saapuvat.

TAULUKKO 1. Tulevien yksiköiden kustannukset

Arduino ESP8266 v2 Amica	6,49 €	35	227,15 €
IR Sensor 10pcs	11,90 €	4	47,60 €
GPIO Cable	7,99 €	2	15,98 €
GPIO Connector assembly	10,98 €	2	21,96 €
LED strip (5M300LED, 10cm for 1pc)	29,98 €	1	29,98 €
Capacitor (10V 1000uF, 20pcs)	8,25 €	2	16,50 €
Connector LED (10pcs dual end)	11,99 €	2	23,98 €
Connector capacitor (10pcs dual end)	7,59 €	2	15,18 €
USB cable (USB-A, USB-micro, 3m)	10,99 €	14	153,86 €
USB charger (USB-A port, 4pcs)	20,47 €	10	204,70 €
Raspberry Pi 4 kit	109,98 €	2	219,96 €
Total			992,83 €

6 YHTEENVETO

Työn tarkoituksena oli löytää ratkaisu, jonka avulla OTA-kammion sisällä olevan testilaitteen tila-tieto on helposti nähtävillä sekä laboratoriotilassa työskenteleville ja etätyöntekijöille. Opinnäyte-työn tuloksena syntyi fyysinen laite sekä web portaali, joiden avulla laboratoriossa olevat työntekijät ja etätöitä tekevät voivat nähdä kontin sen hetkisen tilan helposti.

Työ alkoi fyysisen laitteen suunnittelulla ja kokoamisella. Ensimmäinen malli oli todella vaikea koota ja liian terävät kulmat aiheuttivat sen, että osat menivät helposti rikki. Kuvassa 6 näkyvä laite on toinen ja viimeisin versio laatikosta, mutta siinä on ensimmäinen versio akryylilevystä. Akryylilevyn toisessa ja nykyään käytetyssä versiossa on tehty raidat ulkoreunoille, jotta valon väri näkyisi pa-remmin eikä niin suuri osa valosta pakenisi levyn alalaidasta.

Vaikein osa työstä oli serveripuolen toiminnot ja virtatiedon saaminen etäohjattavalta virtalähteeltä. Yrityksestä kuitenkin löytyi paljon henkilöitä, jotka pystyivät auttamaan kaikissa ongelmissa. Kaiken kaikkiaan laitteesta tuli toimiva ja se täyttää sille asetetut vaatimukset.

LÄHTEET

1. Sähköturvallisuuslaki 1135/2016. Hakupäivä 4.5.2021.
[Sähköturvallisuuslaki 1135/2016 - Säädökset alkuperäisinä - FINLEX®](#)
2. Nokia Oyj:stä tietoa. Hakupäivä 9.12.2020.
[Nokia \(yritys\) – Wikipedia](#)
3. OTA chamber tech spec 0.4.0 Sisäinen lähde
4. NodeMCU ESP8266 Detailed review. Hakupäivä 3.2.2021.
[NodeMCU ESP8266 Specifications, Overview and Setting Up \(make-it.ca\)](#)
5. NodeMCU liitinkaavio. Hakupäivä 31.3.2021.
[NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet \(components101.com\)](#)
6. Raspberry pi 4. Tuotekuvaus. Hakupäivä 14.1.2021.
<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
7. Aduino IDE. Käyttöohje. Hakupäivä 9.11.2020
[Arduino Software \(IDE\) | Arduino](#)
8. Fielding, Roy Thomas 2000. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine. Hakupäivä 19.4.2021.
https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
9. About Node.js. Hakupäivä 20.4.2021.
[About | Node.js \(nodejs.org\)](#)
10. What is npm? Hakupäivä 20.4.2021.
[What is npm? | Node.js \(nodejs.org\)](#)

11. What is Python? Hakupäivä 22.4.2021.
[What is Python? - PythonForBeginners.com](https://pythonforbeginners.com/what-is-python/)
12. Fisher, Tim 2020. How to use telnet client in windows. Hakupäivä 26.4.2021.
[How to Use the Telnet Client in Windows \(lifewire.com\)](https://lifelwire.com/how-to-use-the-telnet-client-in-windows/)
13. PDU Power Breaker 48-16 Datasheet. Hakupäivä 1.12.2020.
[PDU Power Breaker 48-16 2017.pub \(enics.com\)](https://enics.com/PDU-Power-Breaker-48-16-2017.pub)
14. Enics Communication Spec PDU Power Breaker 48-16