



Abdirauf Osman

Toimipisteen etäohjaus ja -seuranta Raspberry Pin avulla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikan tutkinto-ohjelma

Insinöörityö

03.11.2021

Tiivistelmä

Tekijä:	Abdirauf Osman
Otsikko:	Toimipisteen etäohjaus ja -seuranta Raspberry Pin avulla
Sivumäärä:	37 sivua
Aika:	03.11.2021
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine:	Automaatiotekniikka
Ohjaajat:	Timo Tuominen, Lehtori

Insinööri työ oli luonteeltaan kehitys- ja opetustyö. Työssä pyritään luomaan asiakkaan toiveiden mukainen ratkaisu ja opettaa samalla, miten kyseinen ratkaisu on tehty. Projekti on tehty osana Digi-salama-hanketta ja heidän asiakkaansa toiveesta.

Työn teoriaosuudessa käytiin läpi projektin systeemissä käytettyä tekniikkaa, kilpailevia vaihtoehtoja ja niiden historiaa.

Insinööri työn käytännön osuuden tarkoituksena oli luoda toteutus, jossa etsitään IoT-väylä ratkaisu asiakkaan määrittämän kohteen etäohjaukseen ja valvontaan. Työn käytännön osuudesta ei voida avata salassapitosopimuksen vuoksi hankkeen asiakkaan yritystä eikä kohdetta, johon se oli tehty.

Lopputuloksena syntyi IoT-järjestelmä Raspberylle, jossa pystytään varastoimaan saatua dataa verkkoon ja ohjaamaan Raspberyyyn liitettyjä komponentteja sisäistä verkkoa käyttäen. Lopuksi luotiin ohjekirja asiakkaalle tukemaan, miten kyseinen Raspberyn IoT-toteutus luodaan ja vaihtoehtoisia ratkaisua toteutuksen rinnalle, mikäli niistä olisi hyötyä palvelemaan paremmin asiakkaan tarpeita. Ohjekirjassa käytiin läpi perusteellisesti systeemin vaiheita sitä, kuinka se toimii.

Avainsanat: Mikrotietokone, Raspberry, Python, Pilvipalvelut, Node-RED, MongoDB, Visual Studio, Arduino, IoT, NodeJS.

Abstract

Author:	Abdirauf Osman
Title:	Site Control and Monitoring of the Office with Raspberry Pi
Number of Pages:	37 pages
Date:	03 November 2021
Degree:	Bachelor of Engineering
Degree Programme:	Electric and Automation engineering
Professional Major:	Automation Technology
Supervisors:	Timo Tuominen, Senior Lecturer

The engineering work was developmental and teaching work in nature. The aim of the work was to create a solution that meets the customer's wishes and at the same time teaches how that solution is made. The project was made as part of the Digi-Salama project and at the request of their client.

The theoretical part of the work covered the technology used in the project system, competing terms and conditions and their history.

The purpose of the practical part of the engineering work was to create an implementation of an IoT solution for remote control and monitoring of a customer-specified site and objects. For the practical part of the work, due to the non-disclosure agreement, it is not possible to subscribe the project client's company, nor the site for which it was made.

The result is an IoT system by Raspberry Pi, which can store the received data on the network and control the components connected to Raspberry Pi using the internal network. Finally, a manual was created for the customer to explain how that Raspberry Pi IoT implementation is created and giving alternative solutions alongside the implementation if these could be useful to better serve the customer's needs. The manual thoroughly reviews the steps of the system and how it works.

Keywords: Microcomputer, Raspberry, Python, Cloud service, Node-RED, MongoDB, Visual Studio, Arduino, IoT, NodeJS.

Sisällys

1	Johdanto	1
2	Systeemissä käytetyt valinnat, tekniikan taustaa ja ratkaisuja	2
2.1	Mikrotietokone	2
2.1.1	Raspberry PI	3
2.1.2	Arduino	5
2.1.3	Raspberry vs. Arduino	7
2.2	Ohjelmoinnin kehitystyökalut	8
2.2.1	Node-Red	8
2.2.2	Visual Studio	8
2.3	Pilvipalvelut	9
3	Järjestelmän toteutus	12
3.1	Lähtökohta	12
3.2	Järjestelmän suunnittelu	12
3.3	Komponenttien alustus ja käynnistys	13
3.3.1	Komponenttien kytkeminen	13
3.3.2	Komponenttien toiminnan määrittäminen	17
3.3.3	Yhteys	21
3.4	IoT-osuus	31
4	Yhteenveto	33
4.1	Lopputuotoksen läpikäynti	33
4.2	Haasteet	34
4.3	Jatkokehitys ja tulevaisuuden näkymät	34
	Lähteet	35

Lyhenteet

Adafruit: Tuo kirjastoja ohjelmointikielille.

FBD: Flow based Programming, vuopohjainen ohjelmointi.

IOT: Internet of things, esineiden internet.

IBM: Teknologiayritys.

Node-Red: Ohjelmointityökalu.

MQTT: Message Queue Telemetry Transport, Viesti jonon telemetrian siirto protokolla.

ORM: Object-relational mapping. Ohjelmiston oliomallin mukaisen esityksen kuvaaminen relaatiotietokantamallin mukaiseksi esitykseksi ja kääntäen.

RAM: Random Access Memory, hajasaantimuisti.

TKHJ: Tietokannan hallintajärjestelmä. Ohjelmisto tiedon tehokkaan hakemisen, säilyttämisen ja päivittämisen toteuttamiseksi.

1 Johdanto

Insinööriyön tavoitteena on luoda IoT-järjestelmä, jossa voi tarkkailla ja ohjata etäältä toimipisteen ympäristöä, tilannetta ja komponentteja. Lisäksi asiakkaalle luotiin ohjekirja, jota seuraamalla pystytään toteuttamaan, miten kyseinen järjestelmän pystytään itse luomaan. Insinööriyö koostuu käytännön osuudesta ja teoriaosuudesta. Työn teoriaosuudessa käytiin läpi projektin systeemissä käytettyä tekniikkaa, kilpailevia vaihtoehtoja ja niiden historiaa. Insinööriyön käytännön osuuden tarkoituksena oli luoda toteutus, jossa etsitään IoT-väyläratkaisu asiakkaan määrittämän kohteen etäohjaukseen ja valvontaan.

Projektin ytimenä on selvittää keinoja, miten toteuttaa Raspberry Pi -ohjaus ja seurata datan liikkumista etäisesti. Projektissa käytetään mikrotietokoneena Raspberry Piä, jolle luodaan IoT-väylä etäohjausta ja -valvontaa varten. IoT on lyhenne sanoista "Internet of Things". Sillä tarkoitetaan laitteiden automaattista tiedonsiirtoa, ohjausta ja valvontaa internetverkon kautta. Kyseinen tekniikka on lisääntynyt viime vuosien varrella ja tulee lisääntymään teollisuudessa ja kotitalouksissa huomasti tulevaisuudessa. Piirilevytietokoneidenkin käyttö ja tarjonta on kasvanut viime vuosien saatossa ja tulevaisuudessa. Piirilevytietokoneita käytetään nykypäivänä yritysten, opiskelijoiden, harrastelijoiden ja yritysten käytössä.

Insinööriyöllä pyritään tuomaan jo olemassa olevaa tekniikkaa nykyaikaan liittämällä sitä internetin maailmaan (IoT). Insinööriyö on luonteeltaan kehitys- ja opetustyö. Työssä pyritään luomaan asiakkaan toiveiden mukainen ratkaisu ja opettamaan samalla miten kyseinen ratkaisu on tehty.

Projekti on tehty osana Digi-salama-hanketta ja heidän asiakkaansa toiveesta projekti on salassapitosopimuksen alainen. Siksi ei voida mainita yrityksen nimeä eikä kohdetta, mihin projektia on suunniteltu. Digi-Salama-hankkeessa toteutetaan Salamaprojekteja pääkaupunkiseudun yrityksille. Projekteja toteutetaan viidessä digitaalisessa industry 4.0-teemassa: Autonomiset liikkuvat robotit, Cobotiikka, koneoppiminen automaatioissa, VR/Holdeck/AR sekä Digital Twin-teknologia. Kyseisiä projekteja toteutetaan yhteensä 25 kappaletta. Toimialat Salamaprojekteissa ovat elintarvike, logistiikka, kiinteistö, kiertotalous ja korkean osaamisen tuotanto. Digi-salama-hanketta toteuttavat Vantaan kaupunki ja Metropolian Ammattikorkeakoulu yhteistyössä Tampereen yliopis-

ton TRINITY-hankkeen, VTT:n DIH2 Suomen Robotiikkayhdistys ry:n ja Suomen Automaatioseuran kanssa. [24.]

Projektin vaatimuksena ja tavoitteena oli luoda IoT-väyläratkaisu asiakkaan määrittämän kohteen etäohjaukseen ja valvontaan. Asiakkaan toive projektin alussa oli saada ensituntuma ja ratkaisu Raspberry IoT -toteutukselle, oppia sen toiminta ja löytää uusia keinoja toteuttaa haluttu tehtävä.

Insinööriyössä päämääränä oli tehdä järjestelmä, joka tuo jo olemassa olevat komponentit internetin maailmaan, joissa niiden toimintaa voidaan ohjata ja tarkkailla. Työssä käytettiin keskuksena mikrotietokonetta, jonka ympärille liitettiin halutut komponentit ja sille luotiin IoT-väylä. Projekti työn tarkoituksena oli soveltaa pilvipalvelin ympäristöä hyödyntäen valmiiksi olevia komponentteja systeemiin, jossa luodaan sopiva ratkaisu asiakkaan työympäristön tarpeeseen.

2 Systemissä käytetyt valinnat, tekniikan taustaa ja ratkaisuja

Systeemi on joukko elimiä, jotka toimivat yhdessä osana isompaa kokonaisuutta. Elimillä tarkoitetaan kyseisen systeemin kohdalla mikrotietokonetta, ohjelmoinnin kehitystyökalua ja pilvipalvelua. Tässä luvussa käydään läpi edellä mainittujen elimien taustaa eri vaihtoehtoisista valintoja ja valintojen kilpailijoita.

2.1 Mikrotietokone

Mikrotietokone on tietokone, jossa on mikroprosessori ja käyttää keskusmuistina RAM-muistia (Random Access Memory). Neljännen sukupolven tietokoneita kutsutaan mikrotietokoneiksi. Mikrotietokonejärjestelmistä käytetään myös termiä mikrokontrolleri. Mikrotietokoneessa on vähintään mikroprosessori, ohjelmointimuisti, data muisti ja lähtö/tulo portit (I/O).

Datan lukemisen lisäksi RAM-muistiin voidaan tallentaa mikrotietokoneista muistia. Datan käsittelyyn tarvittavat ohjelmat ja tiedostot on tallennettu RAM-muistiin. RAM-muistista katoaa data välittömästi laitteen käyttöjännitteen katkeamisen jälkeen. Yleensä mikrotietokoneissa on apumuistilaitteita, kuten esimerkiksi SD-kortti tai kovalevy, jolle käsiteltävä data ja ohjelmat pysytyttään pysyvästi tallentamaan. Mikrotietokoneita

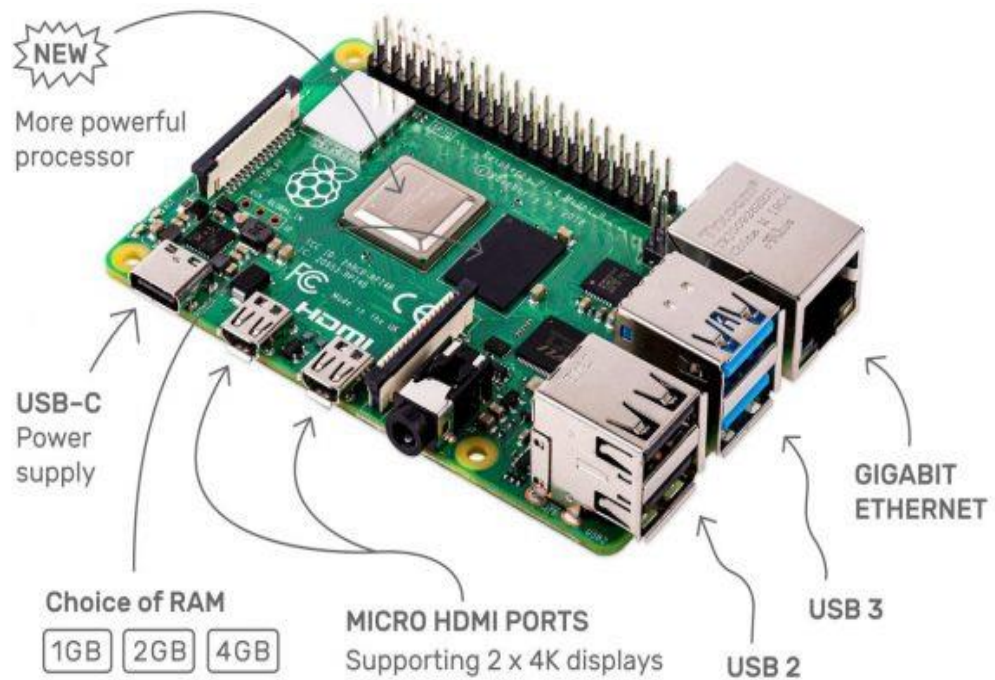
käytetään useassa joka päivä käytetyssä laitteessa, kuten Hi-Fi-laitteissa, TV-
kaukosäätimissä ja useissa muissa kohteissa. [2 ja 3.]

2.1.1 Raspberry Pi

Raspberry Pi on mikrotietokone, joka on kooltaan kirjastokortin kokoinen. Sen suorituskyky vastaa vanhemman sukupolven tietokoneita. Laite on mikrotietokone, joka ei sisällä kovalevyä, jonka vuoksi siihen täytyy käyttäjän itse ladata käyttöjärjestelmä, joka tallennetaan Raspberryy-n-asennetulle SD-muistikortille. Rasbian on Linux-pohjainen käyttöjärjestelmä, joka on yksinkertaistettu minimalistista käyttötarkoitusta varten. Sen käyttö on helppo oppia ja se on helposti saatavilla.

Raspberryn käyttö on yleistynyt nuorten sekä digiharrastelijoiden keskuudessa sen helppokäyttöisyyden vuoksi. [8.] Raspberry Pi -mikrotietokone sai alkunsa Englannissa Cambridgen yliopistossa vuonna 2006 silloisen tietotekniikan opiskelijoiden ryhmän ansiosta. Mutta varsinainen projekti alkoi vasta vuonna 2008 yliopiston tietoteknisessä laboratoriossa. Raspberry Pi -laitteen tarkoitus on edistää opiskelijoiden tietojenkäsittelyosaamista, sillä projektiryhmä oli havainnut lukuvuosien aikana puutteita opiskelijoiden tietoteknisessä osaamisessa, jolloin he päättivät työstää projektia ja kehittää ratkaisua ongelmaan. Sen johdosta syntyi idea tehdä helppo käyttöinen pankkikortinkokoinen tietokone tukemaan keskeisiä taitoja informaation aikakaudella. Raspberry-nimi tulee viittauksesta hedelmien nimeämisperinteeseen mikrotietokoneiden vanhana aikana. [7.]

Raspberry Pi -laitetta on myyty maailman laajuisesti yli 12,5 miljoonaa yksikköä, mikä teki Raspberry Pi:stä kolmanneksi myydyimmän yleiskäyttöisen mikrotietokoneen. Kyseinen myyntiluku on nyt yli 19 miljoonaa myytyä tietokonetta. [6.]



Kuva 1. Raspberry Pi 4 -piirilevy kokonaisuudessaan komponenttien sijaintikohdat selitetty. [14.]

Taulukko 1. Raspberry Pi malliluettelo ominaisuuksittain. [1.]

Tuote	Ethernet	Muisti	USB	Bluetooth	WLAN
Raspberry Pi Model 1 B+	Kyllä	256 mt	4	Ei	Ei
Raspberry Pi Zero	Ei	512mt	0	Ei	Ei
Raspberry Pi 3 Model B+	Kyllä	1000 mt	4	Kyllä	Kyllä

Raspberry Pi 3 Model A+	Ei	512mt	1	Kyllä	Kyllä
Raspberry Pi Model 4 B	Kyllä	2000, 4000, 8000 mt	4	Kyllä	Kyllä

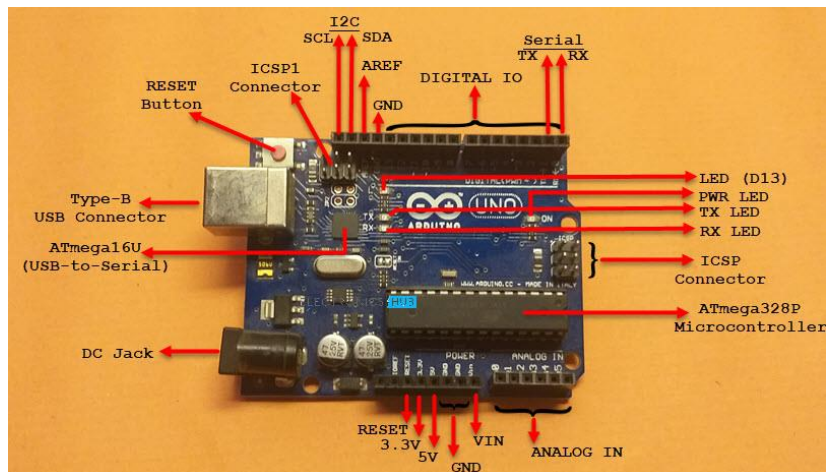
2.1.2 Arduino

Arduino on avoimen lähdekoodin elektroniikka-alusta. Se perustettiin vuonna 2005 opetuskäyttöön opiskelijoille. [13.] Arduino koostuu mikrokontrollerista sekä ohjelmistosta. Laitteiston pinniliitäntäkohtiin voidaan kytkeä erilaisia antureita, moottoreita, valoja sekä muita komponentteja helposti. Arduinon ohjelmistokielenä toimii C:hen ja C++:aan perustuvaan Arduino-ohjelmointikieleen.

Boards	Microcontroller	Operating Voltage/s (V)	Digital I/O Pins	PWM Enabled Pins	Analog I/O Pins	DC per I/O (mA)	Flash Memory (KB)	SRAM (KB)	EEPROM (KB)	Clock (MHz)	Length (mm)	Width (mm)	Cable	Native Network Support
Uno	ATmega328	5	14	6	6	20	32	2	1	16	68.6	53.4	USB A-B	None
Leonardo	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	micro-USB	None
Micro	ATmega32u4	5	20	7	12	40	32	2.5	1	16	48	18	micro-USB	None
Nano	ATmega328	5	22	6	8	40	32	2	0.51	16	45	18	mini-B USB	None
Mini	ATmega328	5	14		6	20	32	2	1	16	30	18	USB-Serial	None
Due	Atmel SAM3X8E ARM Cortex-M3 CPU	3.3	54	12	12	800	512	96	X	84	102	53.3	micro-USB	None
Mega	ATmega2560	5	54	15	16	20	256	8	4	16	102	53.3	USB A-B	None
M0	Atmel SAMD21	3.3	20	12	6	7	256	32	X	48	68.6	53.3	micro-USB	None
Yun Mini	ATmega32u4	3.3	20	7	12	40	32	2.5	1	400	71.1	23	micro-USB	Ethernet/Wifi
Uno Ethernet	ATmega328p	5	20	4	6	20	32	2	1	16	68.6	53.4	Ethernet	Ethernet
Tian	Atmel SAMD21	5	20	12	0	7	16000	64000	X	560	68.5	53	micro-USB	Ethernet/Wifi
Mega ADK	ATmega2560	5	54	15	16	40	256	8	4	16	102	53.3	USB A-B	None
M0 Pro	Atmel SAMD21	3.3	20	12	6	7	256	32	X	48	68.6	53.3	micro-USB	None
Industrial 101	ATmega32u4	5	7	2	4	40	16000	64000	1	400	51	42	micro-USB	Ethernet/Wifi
Uno Wifi	ATmega328	5	20	6	6	20	32	2	1	16	68.6	53.4	USB A-B	Wifi
Leonardo Ethernet	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	USB A-B	Ethernet
MKR1000	Atmel SAMD21	3.3	8	12	7	7	256	32	X	48	64.6	25	micro-USB	Wifi

Kuva 2. Arduinon evoluutio. [18.]

Arduinoa voidaan käyttää erilaisissa projekteissa ja sovelluksissa. Arduino-ohjelmisto on helppokäyttöinen aloittelijoille sekä monipuolisen soveltuva myös kokeneille käyttäjille. Akateemisissa ympäristöissä Arduino on saanut kehuja opettajien ja opiskelijoiden keskuudessa, jotka käyttävät sitä halpojen tieteellisten instrumenttien rakentamiseen, ohjelmoinnin ja robotiikan aloittamiseen. Arduino on yhteen seuraaville käyttöjärjestelmille: Macille, Windowsille ja Linuxille. [11.]



Kuva 3. Arduino-piirilevyn rakenne. [17.]



Kuva 4 Raspberry vs. Arduino. [13.]

2.1.3 Raspberry vs. Arduino

Vertailussa Raspberry Pi ja Arduino saattavat näyttää hyvin samanlaisilta, mutta molemmilla on ainutlaatuiset ominaisuuskyvyt sekä heikkoudet, mikä erottaa heidät toisistaan, ja ne ovat tästä syystä hyvin erilaisia laitteita. Suuri ero on laitteiden välillä. Arduinossa on mikrokontrolleri ja Raspberry Pi:ssä on mikroprosessori. Tässä työssä päädyttiin käyttämään Raspberry Pi:tä sen IoT-toimivuuden takia. [13.]

2.2 Ohjelmoinnin kehitystyökalut

Ohjelmoinnintyökalu on ohjelmistokehittäjien tietokoneohjelma, jota he käyttävät sovellusten ja ohjelmien luomiseen, ylläpitoon ja vianetsintään. Insinööriyössä käytettiin Node-Rediä sen monipuolisuuden ja helposti ymmärrettävän ohjelmointityökalun, jolla on helppo luoda IoT-väylä, sen monipuolisuuden johdosta. [10.]

2.2.1 Node-Red

Node-RED aloitti toimintansa vuoden 2013 alussa Nick O'Learyn ja Dave Conway-Jonesin sivuprojektina työn ohella ja alkoi konseptina soveltamaan MQTT-verkkoprotokollaa. Node-Red on vuopohjainen ohjelmointityökalu. Vuopohjaisen ohjelmointityökalun on alun perin kehittänyt IBM:n Emerging, joka on avoimen lähteen koodialusta. Node-Redin jokaisella solmulla (node, eli noodi) on hyvin määritelty tehtävä ja tarkoitus, jolle välitetään datatietoa solmulohkoja hyödyntäen. Tiedon välittämisen jälkeen siitä siirtyy datatietoa solmuihin, jotka välittävät dataa eteenpäin. Verkko on vastuussa datavirrasta solmujen välillä. [5.]

2.2.2 Visual Studio

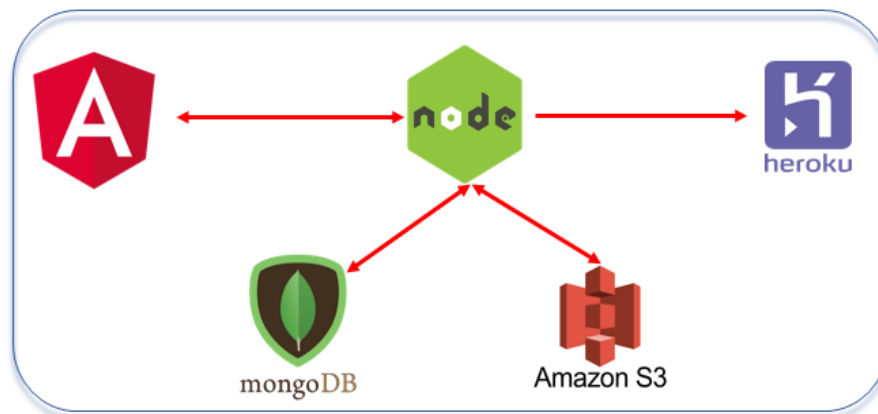
Visual Studio kehitettiin noin 20 vuotta sitten. Sen ensimmäinen versio oli Visual Studio 97. Tämän jälkeen Visual Studioista on julkaistu monta versiota. Viimeisimpänä Microsoft Visual Studio 2019 on tullut markkinoille. Visual Studio on yhteensopiva käyttöjärjestelmiin Windowsiin, Linuxiin ja MacOsiin. Visual Studio on integroitu kehitysympäristöohjelmisto, jonka avulla voidaan kirjoittaa ja muokata koodia. Sen käyttöliittymää käytetään ohjelmistokehitykseen, koodin muokkaamiseen, virheidenkorjaukseen ja koodin rakentamiseen.

Visual Studio sisältää koodieditorin, joka tukee IntelliSense-tekniikkaa (koodin täydennyskomponentti) ja koodin muokkausta. Integroitu debuggeri (virheiden etsijä ja poistaja) toimii sekä lähde- että konetason virheidenkorjaajana. [4.]

2.3 Pilvipalvelut

Pilvipalvelu on nimike palveluille internetissä. Niitä käytetään muun muassa varastoi-
maan ja siirtämään dataa. Eli pilvipalveluilla hankitaan tietokapasiteettiä internetin väli-
tyksellä. Pilvipalveluiden merkittävä ominaisuus on palveluiden hankkimisen helppous.
Käyttäjä maksaa palvelusta haluamansa ajan, jota käyttäjä haluaa hyödyntää palvelua.
Eli käyttäjän ei tarvitse omistaa lisenssiä palveluun käyttääkseen, vaan maksaa vaan
käytetystä palvelusta. Esimerkiksi tuntimäärällisesti sen verran kuin tarvitsee palvelua
tai siirrettävän datan kapasiteetin pohjin. [22.]

MongoDB Atlas on MongoDB:n hallinnoima pilvipalvelutietokanta. Se on suunniteltu
kehittäjille yrityksille tietokantaratkaisuksi. [20.]



Kuva 5. Kuvassa eri pilvipalveluiden yhteensopivuus. [19.]

MongoDB Atlaksen sisään rakennetun suojaustoiminnon ansiosta voivat yritykset sala-
ta arkaluonteisia tietoja hyödyntäen Atlas-tietokantaa. Turvaohjaussuojaustoiminnot
noudattavat vaatimusten mukaisia turvaprotokollia ja täyttävät turvastandardit sekä
tietoturvariskianalyysit.

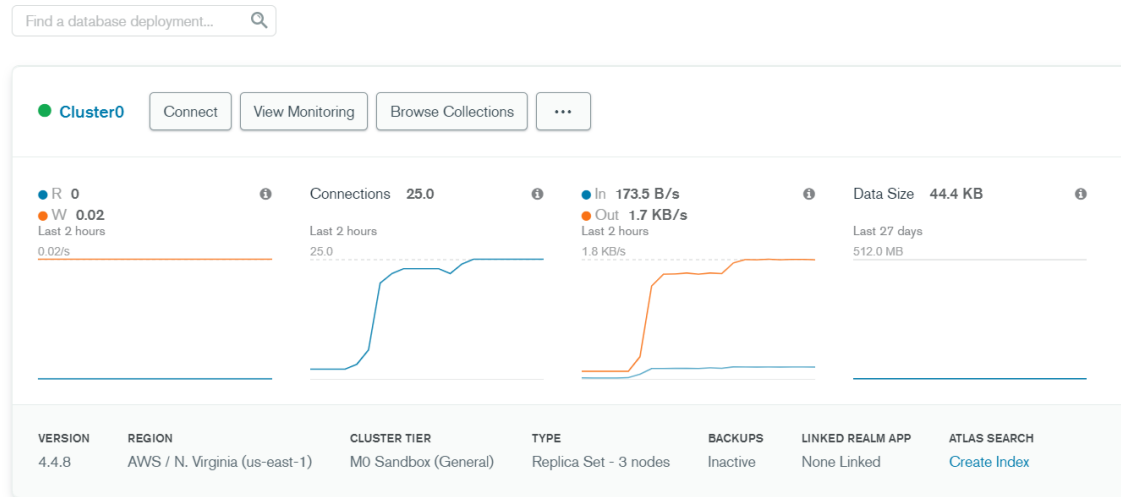


Kuva 6. Mongodb Atlaksen toimintarakenne. [21.]

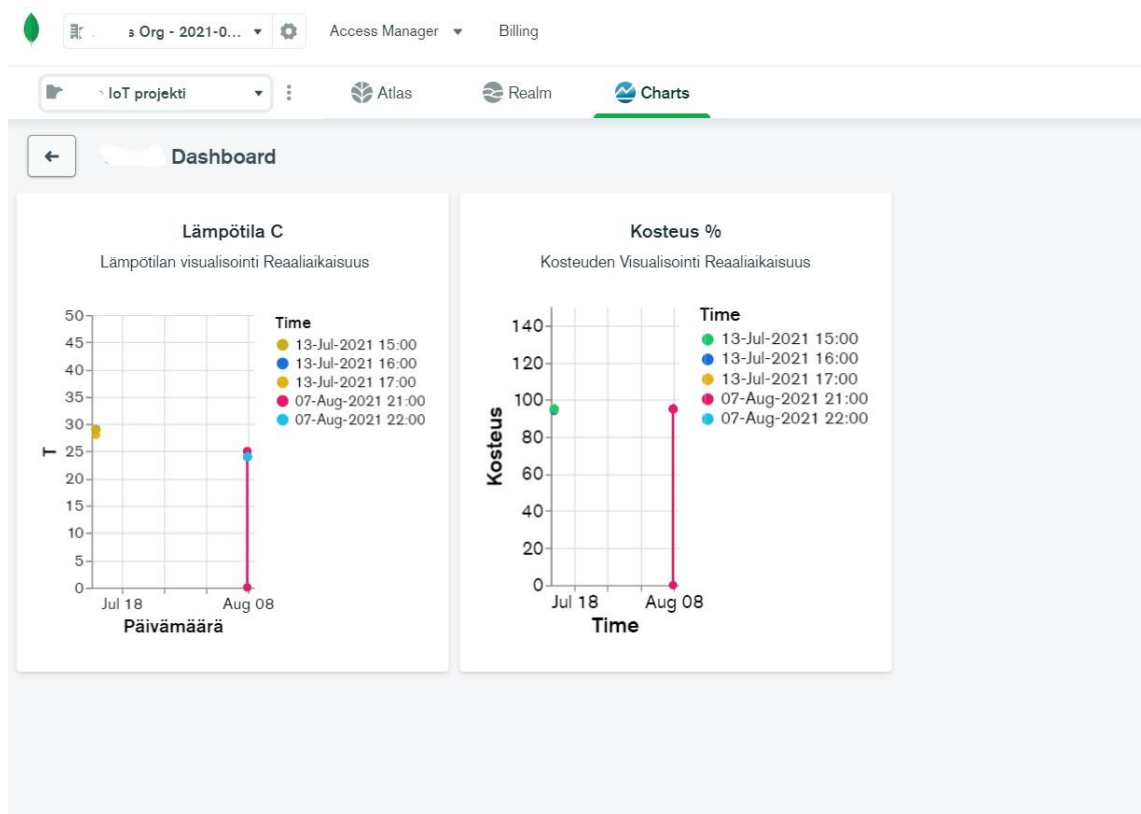


Kuva 7. Kilpailevia pilvipalvelujen tarjoajia. [21.]

Database Deployments



Kuva 8. MongoDB Atlakseen tulleen datan visualisointi.

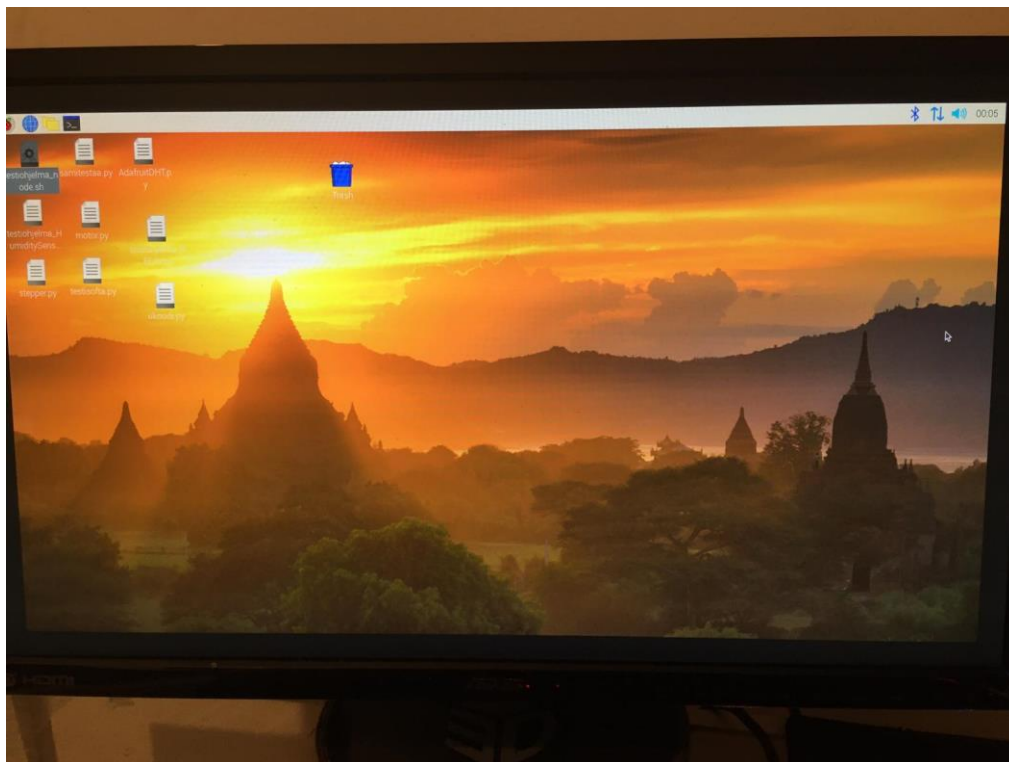


Kuva 9. MongoDB Atlakseen liitettyjen anturidatan seuranta.

3 Järjestelmän toteutus

3.1 Lähtökohta

Työn toteutus vaatii osaamista kyseisistä osaamisen osa-alueista: elektroniikasta ja automaation tietotekniikka. Projektissa luodaan systeemi, jossa etsitään IoT-väyläratkaisu asiakkaan määrittämän kohteen etäohjaukseen ja valvontaan.



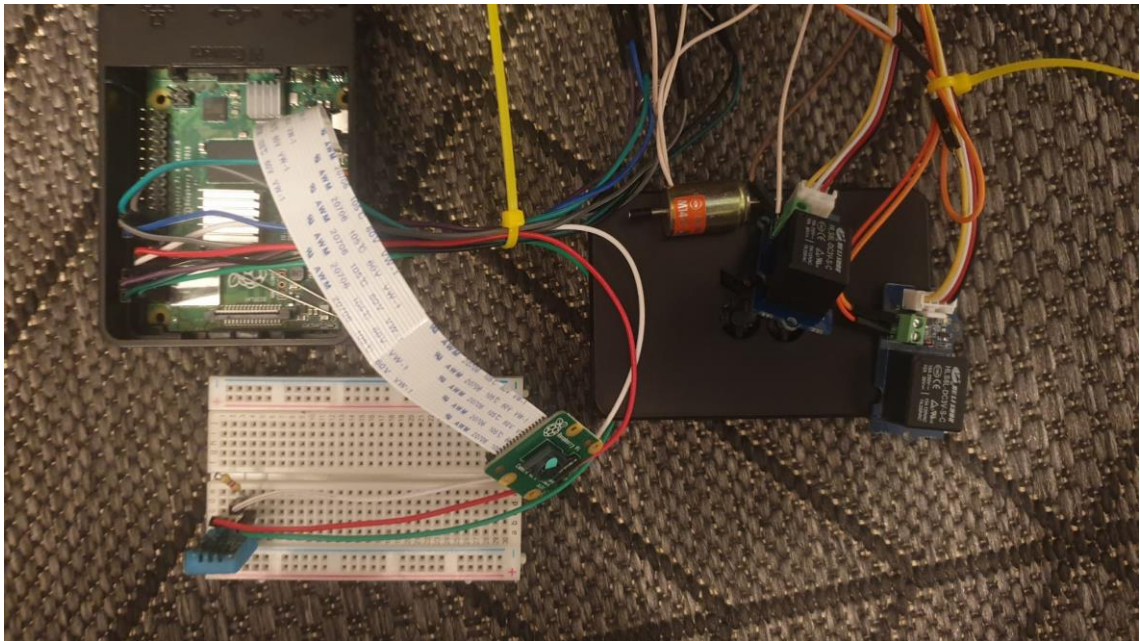
Kuva 10. Raspberry Pi 4 B -malli, Raspberry Pi:n Raspbian-käyttöjärjestelmän aloitusnäyttö.

3.2 Järjestelmän suunnittelu

Järjestelmässä käytettiin mikrotietokoneena Raspberry Piä. Se on helppokäyttöinen mikrotietokone ja halpa lisäosa arkiseen soveltamiseen tai yrityksen tarpeisiin.

IoT-osuuden suunnitellut oli useita vaihtoehtoja. Lopputuloksena parhaana vaihtoehtona selvisi Node-Red sen helppokäyttöisyyden, monimuotoisuuden ja monimuotoisuuden johdosta. Sillä on helppo toteuttaa erilaisia tiedonsiirtoprotokollia.

Raspberry Pi 4 IoT-järjestelmä sisältää 6 komponenttia: Raspberry Pi 4, 2x rele, solenoidi, lämpötila-anturi, kosteusanturi, kamera ja testauslautta.



Kuva 11. Raspberryn komponenttien kytkentä.

3.3 Komponenttien alustus ja käynnistys

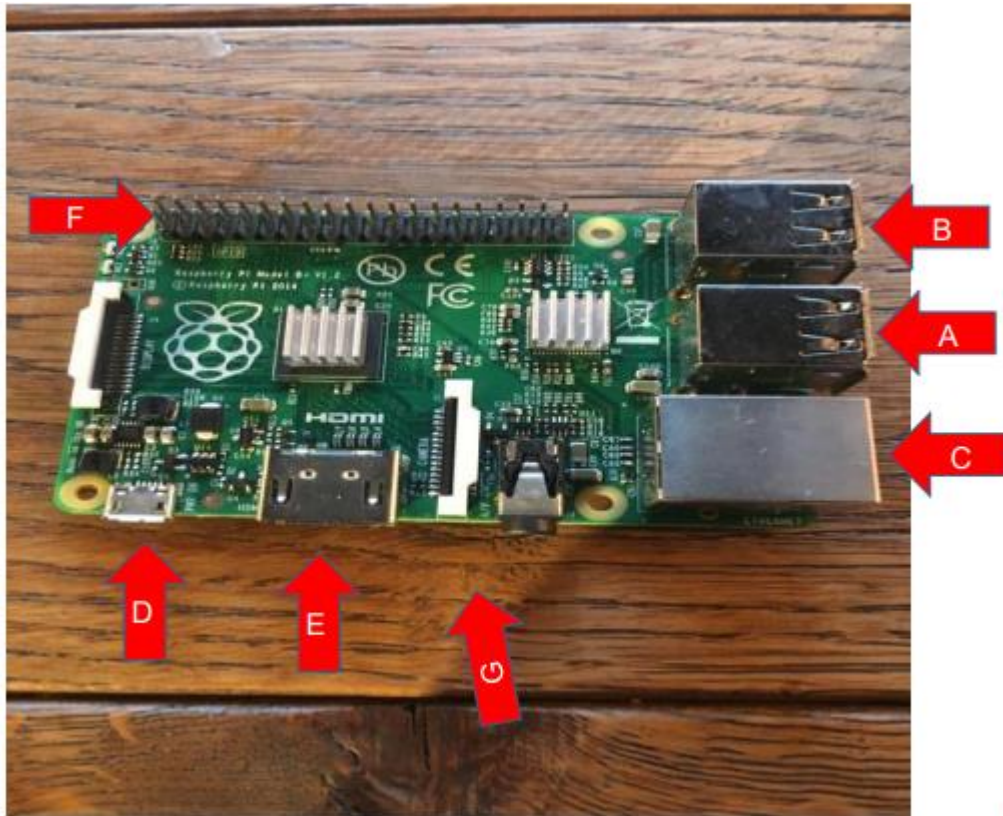
Raspberry Pi vaatii käynnistyäkseen SD-kortin sekä virtalaturin. USB-C-laturi sopii hyvin virtälähteeksi. Raspberry Piin pystyy liittämään lukuisia komponentteja. Seuraavaksi on näytetty, miten projektissa käytetyt komponentit on kytketty.

Raspberry Pi on alussa ilman käyttöjärjestelmää, joten siihen täytyy asentaa Rasbian - käyttöjärjestelmä. Rasbian asennetaan Raspberryn SD-kortille. Rasbian on Linux - pohjainen käyttöjärjestelmä, joka on optimoitu toimimaan yhdessä Raspberryn kanssa.

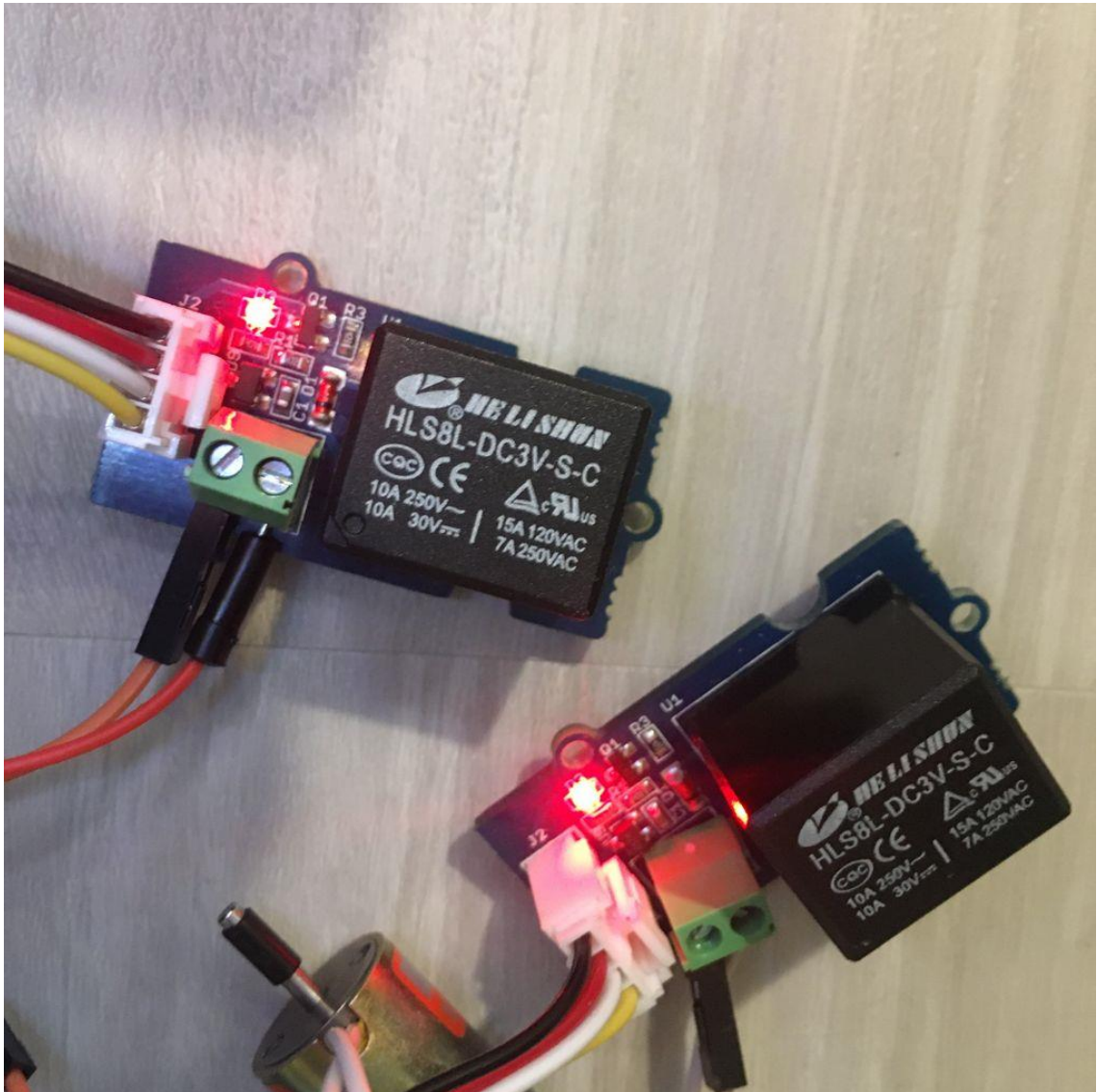
3.3.1 Komponenttien kytkeminen

Raspberryyyn liitettyjen komponenttien kytkentä asianmukaisesti on tärkeää toiminnan kannalta. Komponenttien kytkentävaiheessa määritetään, mitkä komponentit kytketään mihinkin pinneihin (Kuva 15). Mihin pinniin mikäkin komponentti on kytketty määrittää, mistä mitäkin komponenttia ohjataan. Komponentit saavat virtansa pinneistä, jotka on

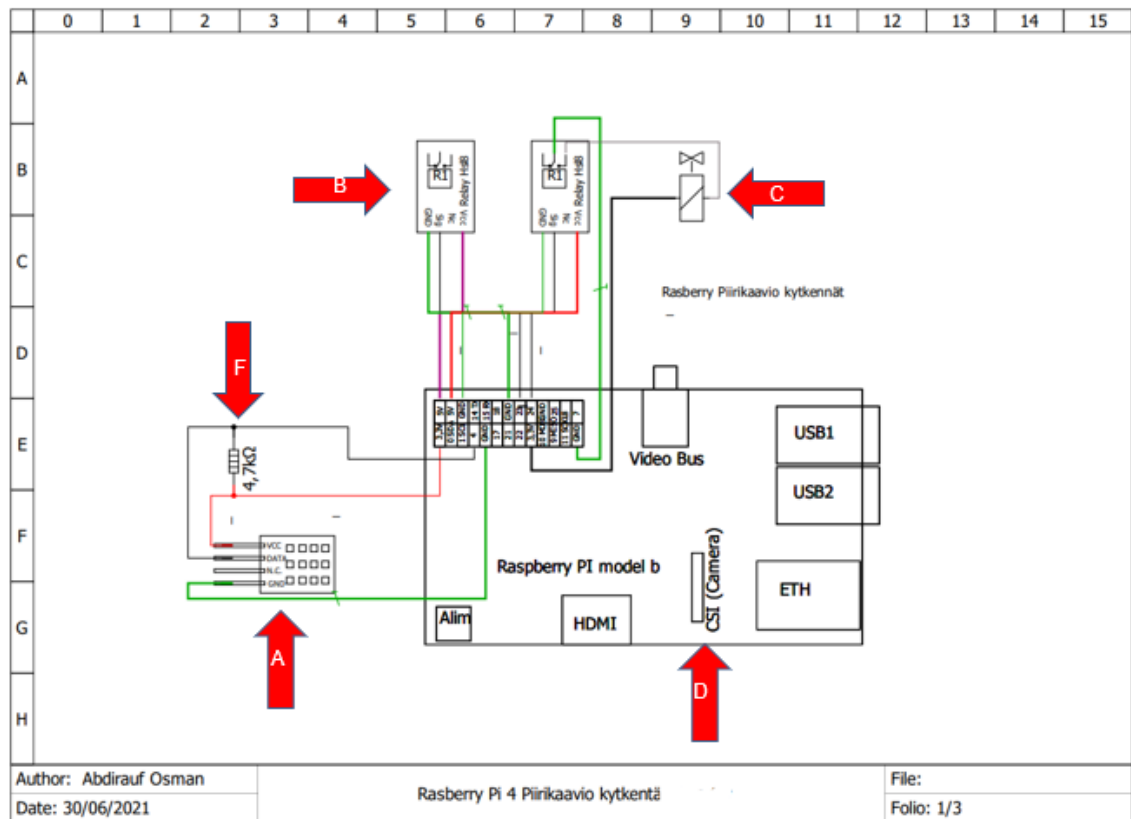
merkitty antavan virtaa ja maadoituksen samalla kaavalla (Kuva 15). Lämpötila- ja kosteusanturilla mitataan ja hallinnoidaan toimipisteen olosuhteita. Antureista tuleva tilatiedon tarkoitus on seurata olosuhteiden ja prosessin etenemistä.



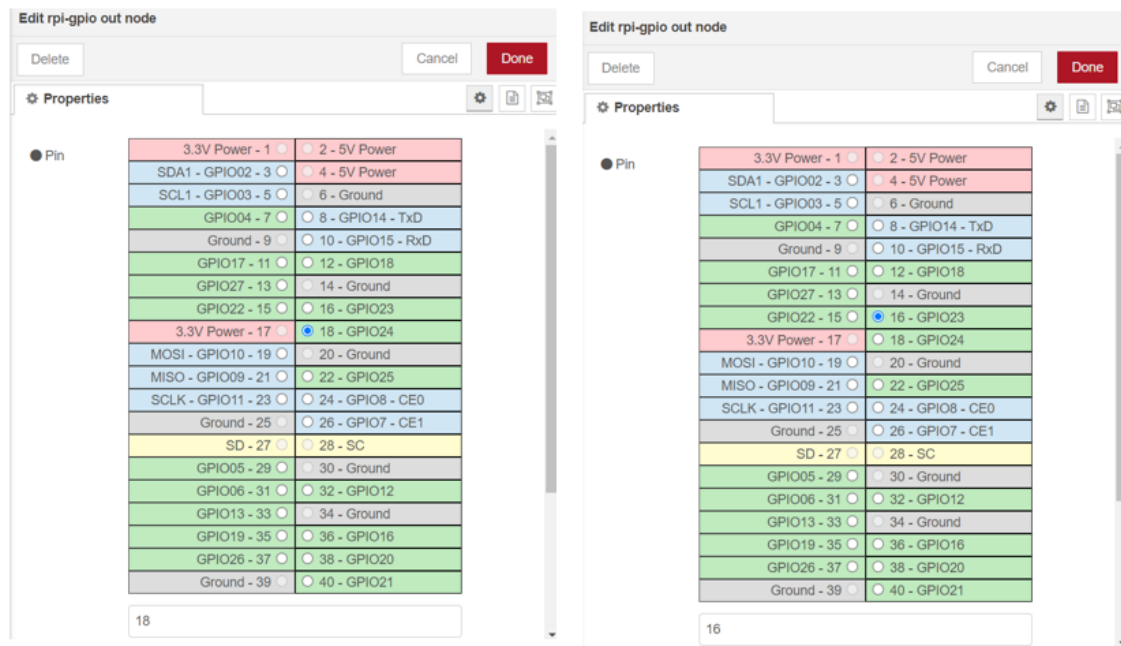
Kuva 12. Raspberry Pi B-malli. (A, B) USB, (C) Ethernet, (D) micro-USB virta aukko, (E) HDMI, (G) kamera, (F) pin-kytkentäkohdat.



Kuva 13. HLS8L-releet ja solenoidi.



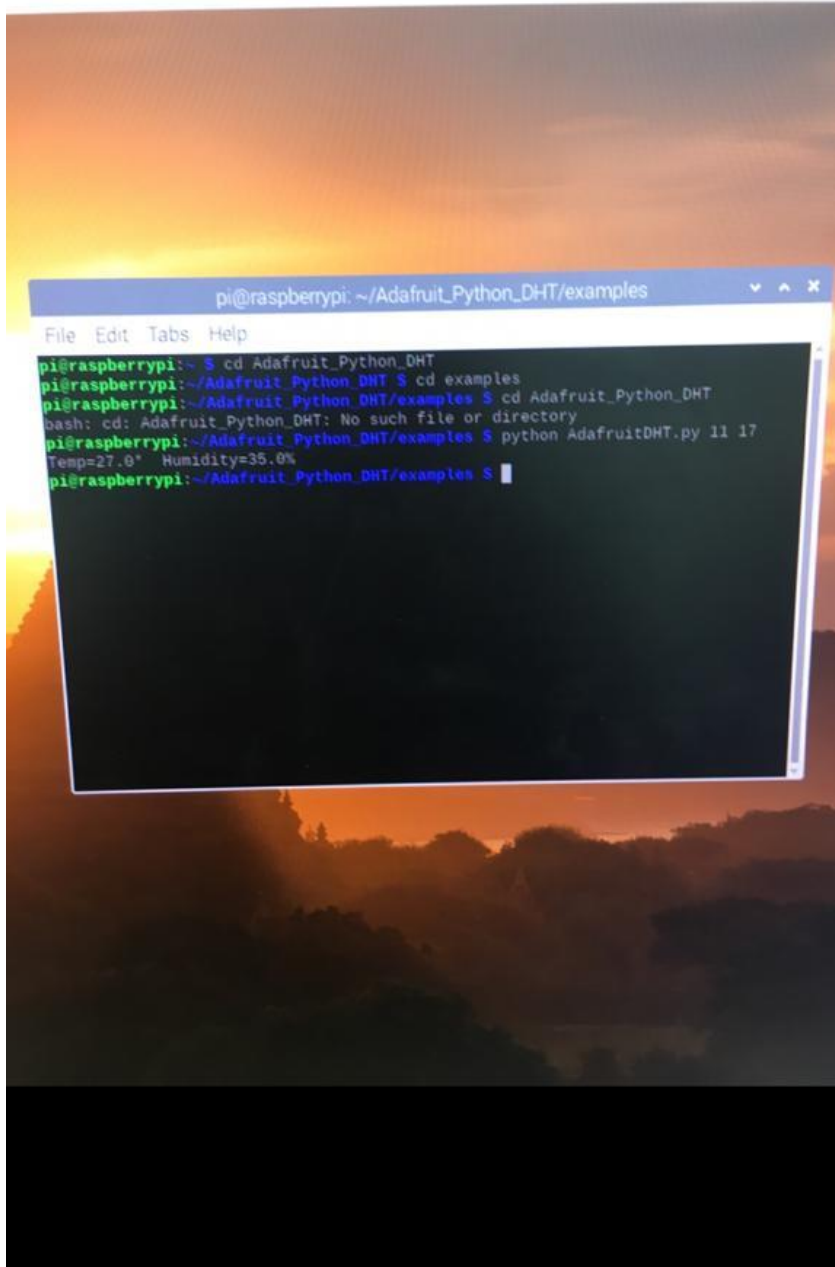
Kuva 14. Piirikaavio Raspberry 4 Pi kytkentäkaavio. DHT11-Lämpötila- ja kosteusanturi (A), releet (B), solenoidi (C), kamera (D), vastus (F).



Kuva 15. Raspberry Pi:n pinnikartta. Vasemmalla solenoidin rele ohjaus päälle ja oikealla rele 2 kytkettynä lähtöön.

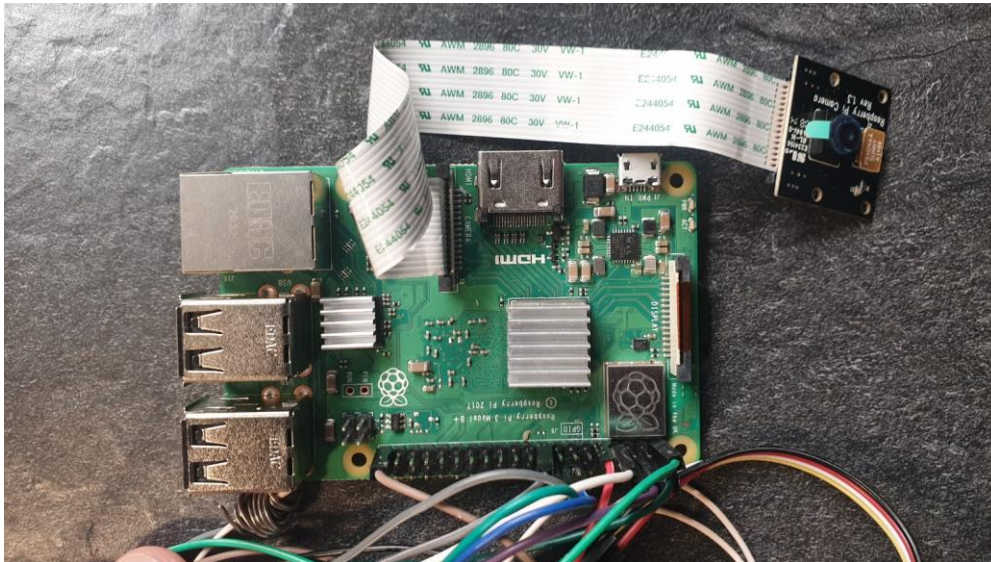
3.3.2 Komponenttien toiminnan määrittäminen

Anturien tarkoitus on seurata olosuhteiden ja prosessin etenemistä. Lämpötila- ja kosteusanturilla mitataan ja hallinnoidaan olosuhdetilaa. Kaikkien komponenttien toiminta määritetään Raspberryn Python-ohjelmointiohjelmalla ja Python-kielellä (Kuva 18 ja Kuva 19).

A screenshot of a terminal window on a Raspberry Pi. The window title is 'pi@raspberrypi: ~/Adafruit_Python_DHT/examples'. The terminal shows a series of commands: 'cd Adafruit_Python_DHT', 'cd examples', and another 'cd Adafruit_Python_DHT' which results in an error 'bash: cd: Adafruit_Python_DHT: No such file or directory'. Then, the command 'python AdafruitDHT.py 11 17' is executed, outputting 'Temp=27.0° Humidity=35.0%'.

```
pi@raspberrypi:~$ cd Adafruit_Python_DHT
pi@raspberrypi:~/Adafruit_Python_DHT$ cd examples
pi@raspberrypi:~/Adafruit_Python_DHT/examples$ cd Adafruit_Python_DHT
bash: cd: Adafruit_Python_DHT: No such file or directory
pi@raspberrypi:~/Adafruit_Python_DHT/examples$ python AdafruitDHT.py 11 17
Temp=27.0° Humidity=35.0%
pi@raspberrypi:~/Adafruit_Python_DHT/examples$
```

Kuva 16. Raspberry Pi:hin Adafruit -asennus DHT11-anturille.



Kuva 17. Raspbery Pi -kameralla.

```
import Adafruit_DHT

sensor=Adafruit_DHT.DHT11
gpio=4

humidity, temperature = Adafruit_DHT.read_retry(sensor, gpio)

if humidity is not None and temperature is not None:
    print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature, humidity))
```

Kuva 18. DHT11-lämpötila- ja kosteusanturin toiminnan määrittäminen.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)
RELAIS_1_GPIO = 23
GPIO.setup(RELAIS_1_GPIO, GPIO.OUT)
GPIO.output(RELAIS_1_GPIO, GPIO.HIGH)
#GPIO.output(RELAIS_1_GPIO, GPIO.LOW)

state = "ON"
|
```

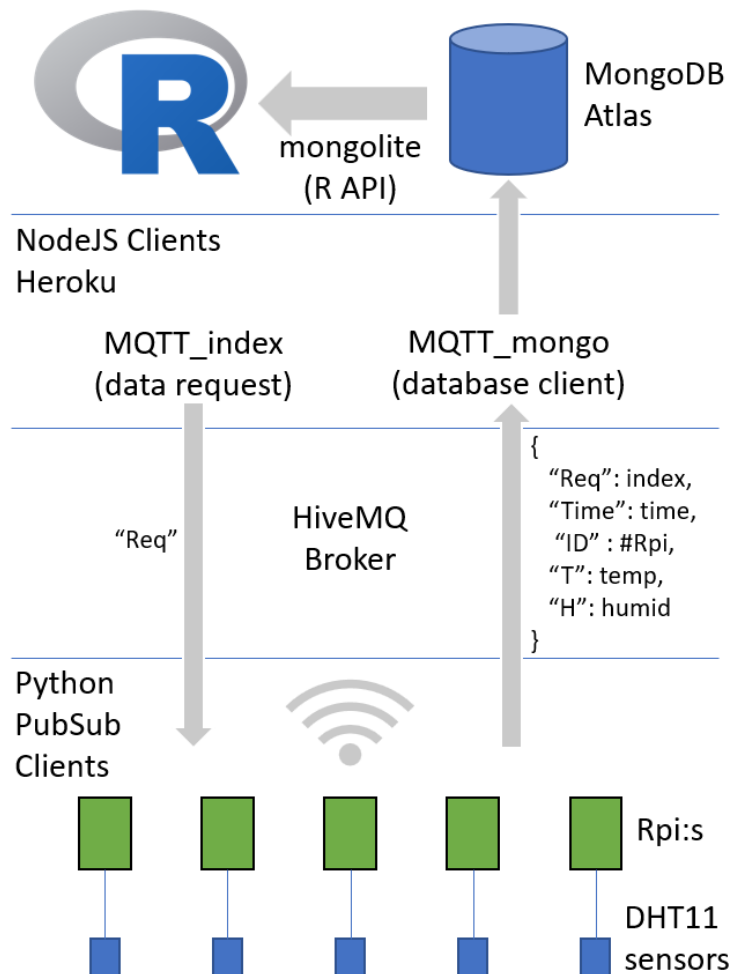
Kuva 19. Releen toiminnan määrittäminen Pythonilla.

3.3.3 Yhteys

Yhteyden luomiseen on käytetty kiinteätä yhteyttä yhdistämällä Raspberry Pi4 -modeemiin Ethernet-piuhaa käyttäen (Kuva 20). IoT-väylän luomisen myötä mahdollistetaan etäohjaus prosessin käynnistykseen.

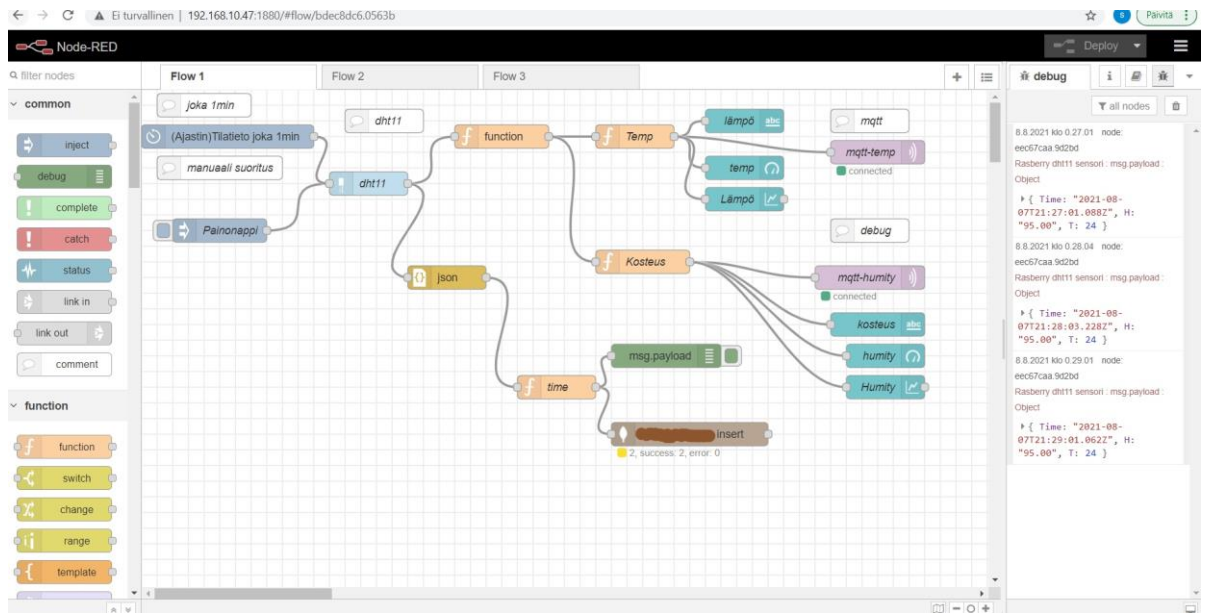


Kuva 20. Raspberryn Pi:n yhteyden liittäminen Ethernet-piuhalla reitittimeen.



Kuva 21. IoT-karttakaavoitus antureille, ohjelmalle ja pilvipalvelimelle. [23.]

Node-Red on ohjelmointityökalu, jolla toteutetaan erilaisia ohjelmointialustoja. Työtilassa tehdään FBD-nimisiä (FLoW Based Programming) laatikkoja ja määritetään niille omat toiminnot, joita kutsutaan noodeiksi. Noodien tarkoitus on yksinkertaistaa vaikeita ohjelmointivirtauksia työtilassa. Node-Red-yhteys määritetään erikseen Rasberista, jolloin Node-Red-konsolin välille yhteys. Tämän jälkeen Node-Red antaa oman IP-osoitteen käyttöjärjestelmän hallinnointiohjelmistosta. Aluksi ladataan Node-JS ohjelma, jotta voidaan pyrittä Node-Red-konsolia. Node-JS:llä luodaan yhteys Node-Rediin tai tiedonsiirto sovellukseen, kuten Heroku.

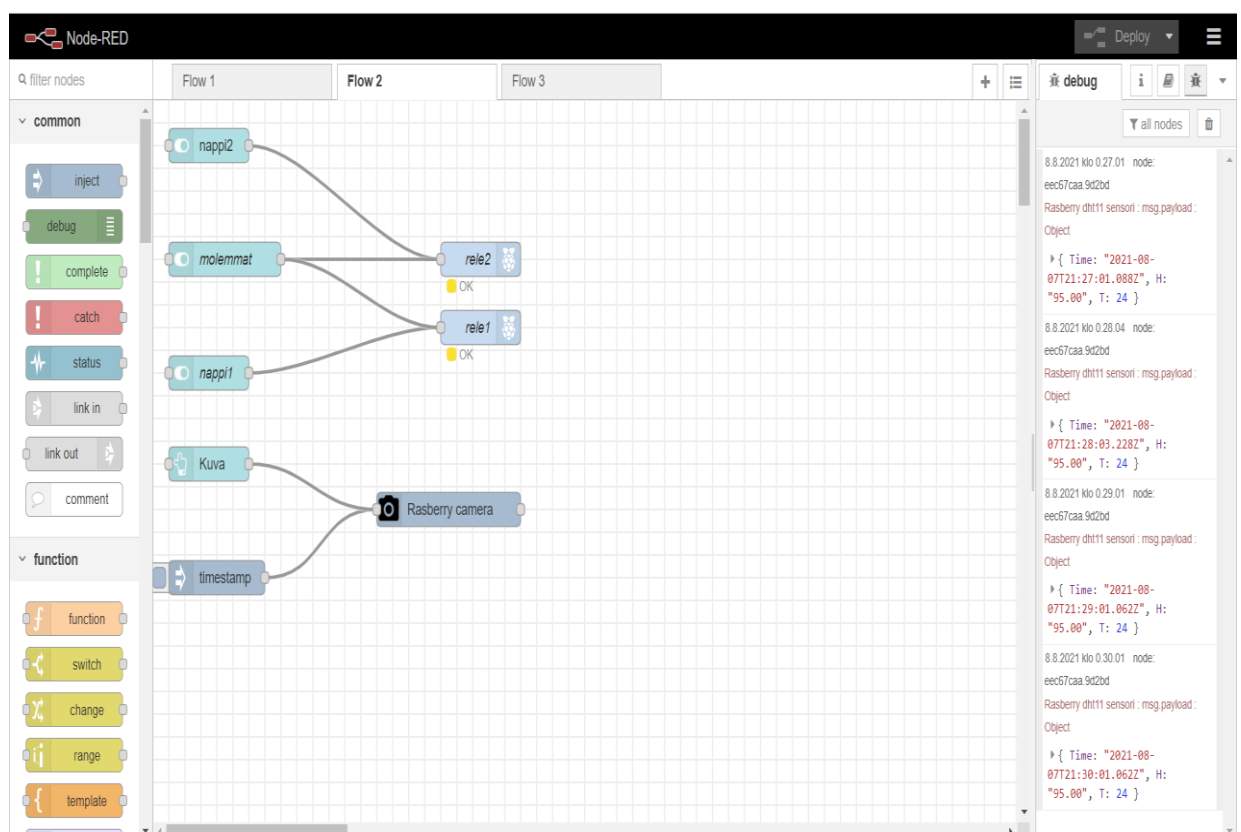


Kuva 22. Node-RED-ohjelmistotyökalu

Taulukko 2. Selityksiä Kuva 22 käytetyille nodeille.

Nimi	Toiminto
<i>Deploy</i>	Lataa tehdyn/muutetun ohjelman Node-Redin palvelimeen.
<i>Painonappi</i>	Nappi toimii kytkimenä antaa käskyn suorittaa anturilta datan lukemisen. sillä voidaan käynnistää virta manuaalisesti napsauttamalla painiketta editorissa ja käynnistämään virtaukset automaattisesti säännöllisin väliajoin.
<i>Function/Time</i>	Toiminta noodi sallii JavaScript -koodin suorittamisen sen läpi välitettyjä viestejä vastaan.
<i>Json</i>	Data muunnos JSON formaattiin.
<i>Kosteus anturi Dht11</i>	Kosteus anturi tieto tulee Raspberrystä siihen määritetyille GPIO
<i>Tempetarture</i>	Lämpötila anturi
<i>msg.payload</i>	noodi näyttää JavaScript -koodin suorittamisen sen läpi välitettyjä viestejä vastaan. Viesti välitetään objektina nimeltä msg.payload hyötykuorma -ominaisuus, joka sisältää viestin rungon. Muu kohdat voidaan liittää viestiin omana ominaisuutena.

<i>Debug</i>	Debug -näkylässä voidaan käyttää viestien näyttämiseen editorin Debug -sivupalkissa. Palkissa on jäsennelty näkymä lähetetyistä viesteistä, mikä helpottaa viestin tutkimista
<i>Mqtt Humidity/ Mqtt temp</i>	Lyhenne : Message Queue Telemetry MQTT on julkaisu- ja tilausprotokolla, joka tarkoittaa, että palvelimen kanssa kommunikoinnin sijaan asiakaslaitteet tässä tapauksessa (Raspberry PI 4) ja -sovellukset julkaisevat ja tilaavat välittäjän käsittelemiä aiheita. MQTT käyttää tyypillisesti IP: tä (Internet Protocol) joka ohjaa dataliikennettä tilaavien ja julkaisevien clienttien välillä. Broker tallentaa viestit vain väliaikaisesti.
<i>Insert Mongodb</i>	Tietokannan hallintajärjestelmä ja käyttää asiakirjakeskeistä tietokantamallia, joka tukee erilaisia datamuotoja. Se sisältää joukon kokoelmia noodina ja tallentaa tiedot rasberista antureista tai muista sille määrätystä noodeista . Mongodb Pystyy käsittelemään erilaisia tietoja ja valtavia määriä dataa MongoDB – tietokantaan.

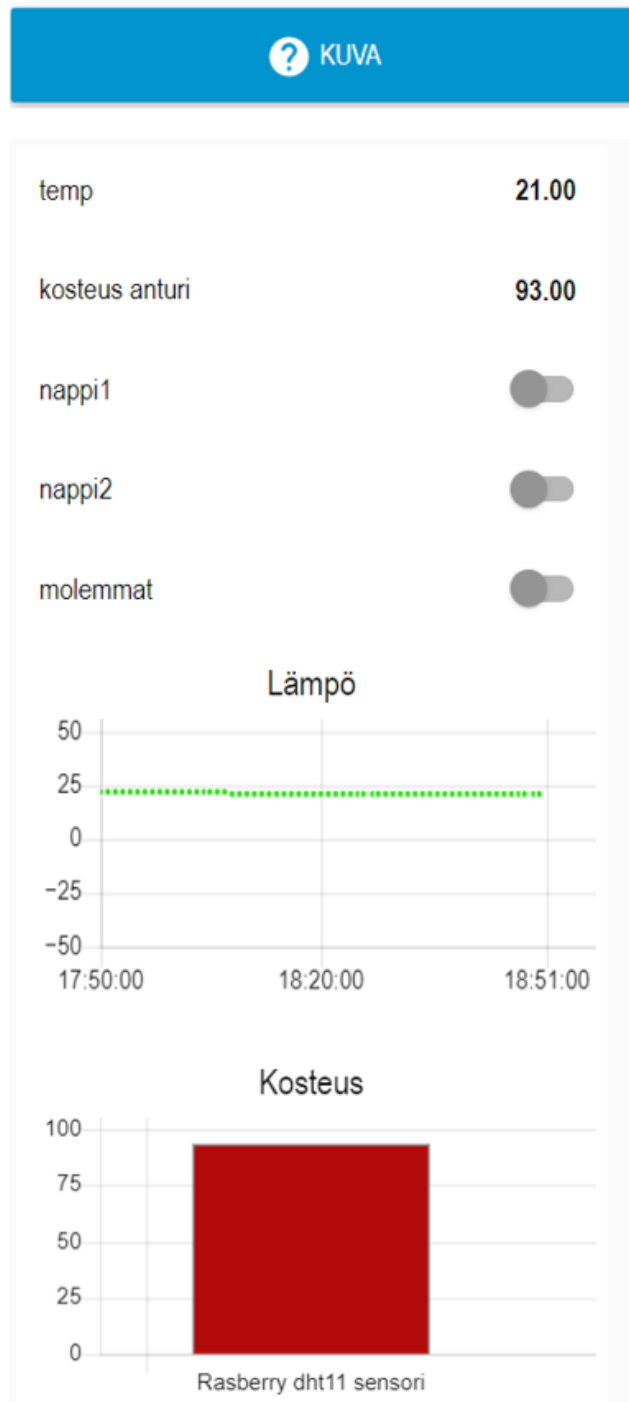


Kuva 23. Noodit yhdistettyinä rele1, kamera, painonappi, rele.

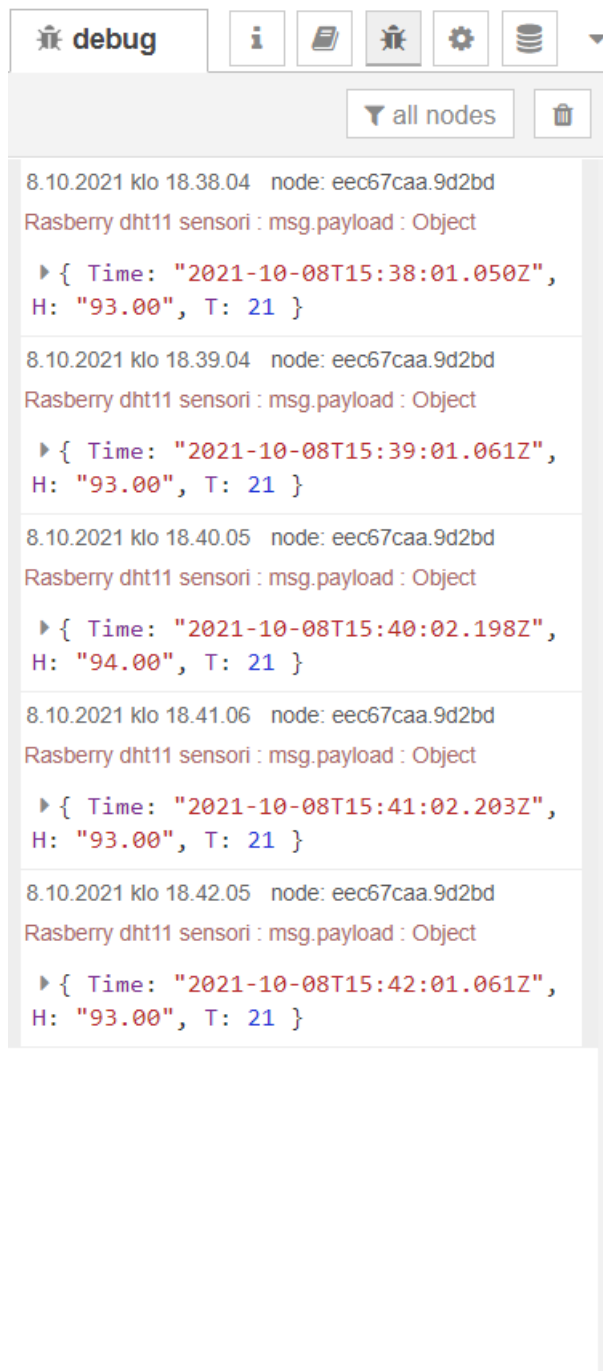
Taulukko 3. Selityksiä Kuva 23 käytetyille nodeille.

Nimi	Toiminto
<i>Nappi 1</i>	Nappi toimii kytkimenä antaa käskyn suorittaa anturilta datan lukemisen .
<i>Molemmat</i>	Nappi toimii kytkimenä molemmille releelle saman aikaisesti sekä käskyn suorittaa anturilta datan lukemisen.
<i>Rele 1</i>	Rele komponentti noodi 1
<i>Rele 2</i>	Rele komponentti noodi 2
<i>Raspberry camera</i>	Kamera noodi
<i>Timestamp</i>	Toimii kytkimenä manuaalisesti antamaan käskyn kameralle ottaamaan kuvan (Kuva 28).
Debug viesti	ilmoittaa viestit noodi (Kuva 25).

Verkonsisäisiä komentoja voidaan ohjata ja seurata käyttämällä Node-REDin-käyttöliittymää (Kuva 24).



Kuva 24. Node-RED-käyttöliittymä.



Kuva 25. Debug-valikkonäkymä. DHT11-anturitiedot.

Edit rpi-dht22 node

Delete

Cancel

Done

 **Properties**



 Topic

Raspberry dht11 sensori

 Sensor model

DHT11 

 Pin numbering

BCM GPIO 

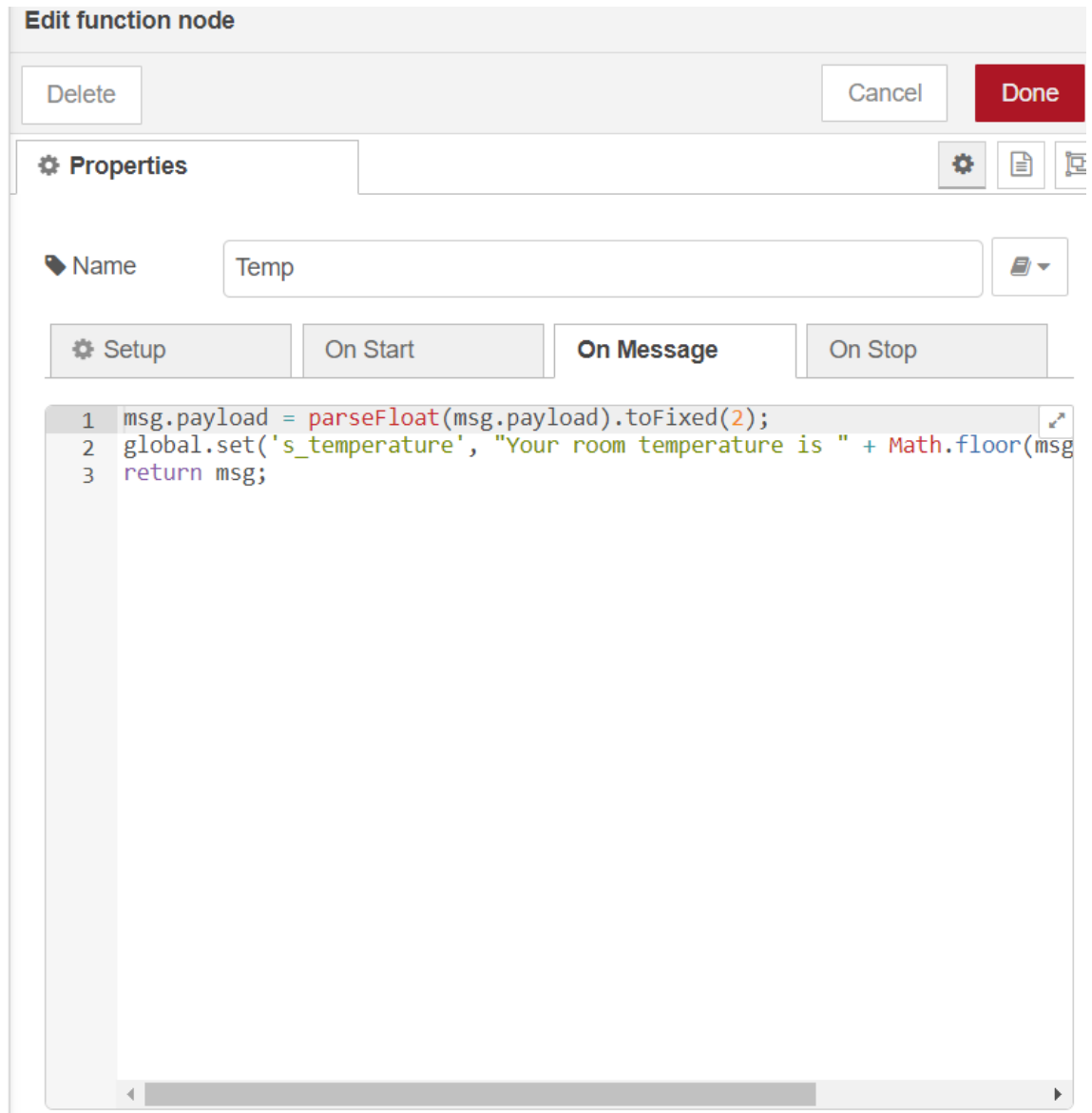
 Pin number

4 

Name

dht11

Kuva 26. DHT11-anturin Node-RED-alustus.



Kuva 27. Lämpötila-anturin alustus Node-Rediin.

Edit cron node

Delete

Cancel

Done

Properties

Name

(Ajastin)Tilatieto joka 1min

Topic

Topic

CRON
Expression

1 * * * * *

Kuva 28. Node-RED:iin asetutun ajastimen alustus.

Edit json node

Delete Cancel Done

Properties

⚙️ Action Convert between JSON String & Object ▼

⋮ Property msg. payload

📁 Name Name

Object to JSON options

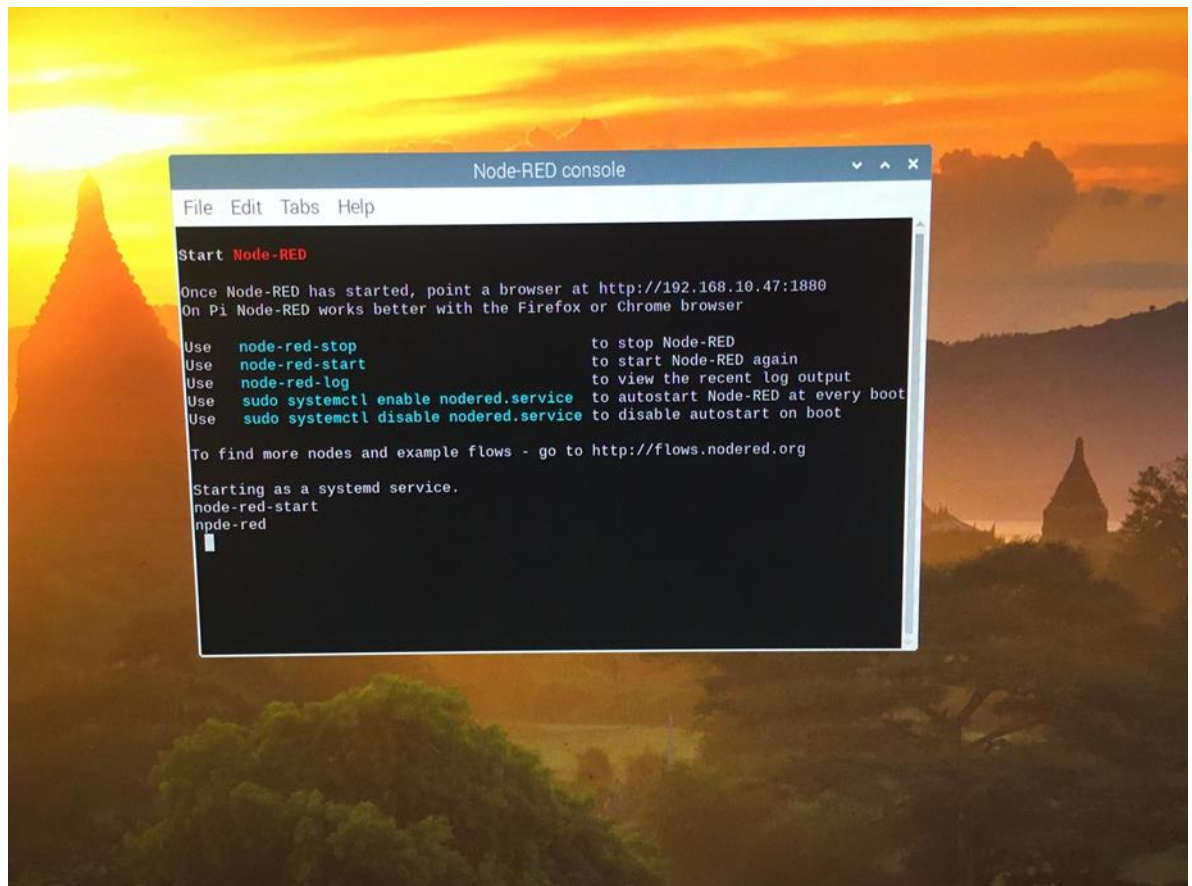
☐ Format JSON string

Kuva 29. JSON-noodi, joka muuntaa tulevan datan JSON-muotoon.

3.4 IoT-osuus

Verkkoon pääsy mahdollistaa IoT-osuuden soveltamisen ja pilvipalveluiden käytön esimerkiksi datan varastointiin. Tässä projektissa päädyttiin käyttämään MongoDB Atlas -datanvarastointi pilvipalvelua. Kyseinen tietokanta ei seuraa mitään kiinteästi määrät-

tyä taulukkoskeemaa, minkä ansiosta se skaalautuu hyvin horisontaalisesti. MongoDB Atlas on kasvanut JSON tietovarastojasta yhdeksi suosituimmaksi NoSQL-tietokantaratkaisuksi, jolla on tehokkaat tietojenkäsittely- ja hallintatoiminnot.



Kuva 30. Rasperryn ja Node-Redin välisen yhteyden luominen.

Raspberry PI:n IoT-kokonaisuus vaiheittain

1. Rasperryn-komponenttien asennus.
2. Rasperryn-komponenttien toiminnan määrittäminen, eli ohjelmointia Pythonilla
3. Node-Red-ohjelmointityökalu: Sisäisen käyttöliittymän luominen.
4. Datavarastointi alustan tilin luominen ja alustaminen tulevan datan vastaanottamiseen.

5. Node-Redin yhdistäminen datanvarastointipalstalle. Node-Redissä IoT-väylän yhdistäminen halutulle Nettipalsta-alustalle.

4 Yhteenveto

Insinööriyön aiheena oli Raspberryn IoT-toteutus, jonka tavoitteena oli toteuttaa Digisalama-hankkeen asiakkaan tarpeiden mukainen Raspberryn etäohjaus ja -seurantatoteutus. Työn teoriaosuudessa käytiin läpi projektin systeemissä käytettyä tekniikkaa, kilpailevia vaihtoehtoja ja niiden historiaa. Insinööriyön käytännön osuuden tarkoituksena oli luoda toteutus, jossa etsitään IoT-väyläratkaisu asiakkaan määrittämän kohteen etäohjaukseen ja valvontaan. Työn käytännön osuudesta ei voida avata salassapitosopimuksen vuoksi hankkeen asiakkaan yritystä eikä kohdetta, johon se oli tehty.

4.1 Lopputuotoksen läpikäynti

Työ oli moniulotteinen, ja päämäärä selkeytyi projektin edetessä. Alussa määritetty tavoite oli luoda IoT-väyläratkaisu asiakkaan määrittämän kohteen etäohjaukseen ja valvontaan. Asiakkaan toive projektin alussa oli saada ensituntuma ja ratkaisu Raspberry IoT -toteutukselle, oppia sen toiminta ja löytää uusia keinoja toteuttaa haluttu tehtävä. Projektin laajuuden vuoksi jouduttiin keksiä täsmällisempiä ratkaisuja ja muuttaa alkuperäisiä asetuksia, jotta tavoitteeseen päästiin. Komponentit päivittyivät projektin edetessä.

Alkuperäinen suunnitelma oli luoda IoT-väylä Visual Studiolla, mutta päädyttiin käyttämään ohjelmointityökaluna Node-Rediä. Node-Red osoittautui olemaan sopivampi ratkaisu Raspberryn IoT-väylän luomiselle. Lopputuloksena syntyi IoT-järjestelmä Raspberrille, jossa pystytään varastoimaan saatua dataa verkkoon ja ohjaamaan Raspberryn liitettyjä komponentteja sisäistä verkkoa käyttäen. Lopuksi luotiin ohjekirja tukemaan asiakasta, miten kyseinen Raspberryn IoT-toteutus luodaan ja vaihtoehtoisia ratkaisua toteutuksen rinnalle, mikäli niistä olisi hyötyä palvelemaan paremmin asiakkaan tarpeita. Ohjekirjassa käytiin läpi perusteellisesti systeemin vaiheita sitä, kuinka se toimii.

Aikataulu projektin toteutuksessa venyi uusien osaamisalueiden oppimisen takia ja projektin vaatimusten muuttuessa työn edetessä. Erityisesti työssä syvennyttiin elektroniikkaan ja automaation tietotekniikalla sen liittämistä IoT-maailmaan.

4.2 Haasteet

Lähtökohta insinöörityön päämäärästä oli hieman epäselvä, koska vaatimukset muutuivat projektin edetessä. Työn toteutus vaati paljon uuden oppimista elektroniikasta ja tietotekniikasta, jotka poikkesivat omasta suuntautumisalastani.

Haasteena IoT-toteutuksessa oli tiettyjen tiedonsiirtoprotokollien liittäminen Node-Rediin, kuten REST-API:iin. Insinöörityön toteutus osui kriittiseen aikaan, kun koronapandemia alkoi uudestaan leviämään, mikä toi mukanaan uusia haasteita, jotka vaativat erityisjärjestelyjä. Erityisjärjestelyillä tässä tapauksessa tarkoitetaan esimerkiksi projektille suunnattuun kohteeseen pääsyä turvallisuussyistä, mikä puolestaan hankaloitti kokonaisuuden hahmottamista. Kommunikaatio työhön suunnatun avustavan konsultin kanssa ja asiakkaan kanssa vaikeutui kesälomien ajoittuessa samanaikaisesti työn toteutuksen kanssa.

4.3 Jatkokehitys ja tulevaisuuden näkymät

Projektin jatkokehityksen näkymät ovat rajattomat. Asiakkaalle annetun ohjekirjan avulla voidaan lisätä uusia ulottuvuuksia systeemille. Uusilla ulottuvuuksilla tarkoitetaan IoT-ominaisuuksien ja kohteiden lisäämistä kohteeseen, sekä uusien komponenttien liittämistä systeemiin. Tulevaisuudessa vanhan tekniikan liittäminen IoT-maailmaan lisääntyy huomasti, sekä uusissa tekniikan laitteissa on paneuduttu integroimaan laitteet yhteensopiviksi jo olemassa oleviin IoT-systeemeihin.

Lähteet

1. Buy Raspberry PI. 2021. Verkkoaineisto. Raspberry PI.
<<https://www.raspberrypi.com/products/>>. Luettu.13.9.2021.
2. Koskinen, J. 2006. Mikrotietokonetekniikka. Sulautetut järjestelmät. Otava Oy, Finland, Keuruu.
3. Dogan, Ibrahim. 2015. Microcomputer Systems. Verkkoaineisto. ScienceDirect.
<<https://www.sciencedirect.com/topics/engineering/microcomputer-system>>. Luettu.16.9.2021.
4. Visual Studio. 2021. Verkkoaineisto. Incredibuild.
<<https://www.incredibuild.com/integrations/visual-studio>>. Luettu.18.9.2021.
5. About. 2021. Verkkoaineisto. Node-RED. <<https://nodered.org/about/>>. Luettu.19.9.2021.
6. History of the Raspberry PI. 2021. Verkkoaineisto. Hitaltech.
<<https://hitaltech.co.uk/the-history-of-the-raspberry-pi/>>. Luettu.21.9.2021.
7. Vilches, Jose. 2012. Interview whit Raspberry's founder Eben Upton. Verkkoaineisto. Techspot. <<https://www.techspot.com/article/531-eben-upton-interview/>>. Luettu.22.9.2021.
8. Gale, J. (2019) Raspberry Pi, Tips, Tricks&Fixes. BDM's Ultimate series
9. Mullins. Roberts. 2012. Department of Computing Science and Technology. Verkkoaineisto. University of Cambridge.
<<https://www.cl.cam.ac.uk/projects/raspberrypi/>>. Luettu.23.9.2021.
10. Programming tools. 2021. Verkkoaineisto. Wikipedia.
<https://en.wikipedia.org/wiki/Programming_tool>. Luettu.27.9.2021.
11. What is Arduino. 2018. Verkkoaineisto. Arduino.
<<https://www.arduino.cc/en/Guide/Introduction>>. Luettu.30.9.2021.

12. Compare Raspberry Pi and Arduino. 2019. Verkkoaineisto. TipsMaker.
<<https://tipsmake.com/compare-raspberry-pi-and-arduino>>. Luettu.1.30.2021.
13. Singh, Rajbir. 2020. Raspberry Pi vs Arduino. Verkkoaineisto. Hackster.io.
<<https://www.hackster.io/rjrajbir/raspberry-pi-vs-arduino-05f69b>>. Luettu.3.10.2021.
14. Whitwam, Ryan. 2019. The Raspberry Pi 4 has a flawed USB-C Port. Verkkoaineisto. ExtremeTech. <<https://www.extremetech.com/electronics/294665-the-raspberry-pi-4-has-a-flawed-usb-c-port>>. Luettu.5.10.2021.
15. MongoDB Evolved – Version History. 2021. Verkkoaineisto.
<<https://www.mongodb.com/evolved>>. Luettu.7.10.2021.
16. Teja, Ravi. 2021. Arduino UNO Pinout, Specification, Board Layout, Pin Description. Verkkoaineisto. Electronics Hub.
<<https://www.electronicshub.org/arduino-uno-pinout/>>. Luettu. 10.10.2021.
17. Aidan. 2018. Arduino Boards, Compared. Verkkoaineisto. Core-Electronics.
<<https://core-electronics.com.au/tutorials/compare-arduino-boards.html>>. Luettu.13.10.2021.
18. Bharadawaz, Kari. 2018. Angular 5 Gallery App using AWS S3 & Heroku-Part1: Approach & Client-Side Development. Verkkoaineisto. Codeburst.IO.
<<https://codeburst.io/angular-5-gallery-app-using-aws-s3-heroku-part-1-approach-client-side-development-82ef56d009e7>>. Luettu.14.10.2021.
19. JavaTpoint. 2021. MongoDB Atlas. Verkkoaineisto. JavaTpoint.
<<https://www.javatpoint.com/mongodb-atlas>>. Luettu. 25.10.2021.
20. MongoDB. 2021. What is MongoDB Atlas. Verkkoaineisto. MongoDB.
<<https://docs.atlas.mongodb.com/>>. Luettu.27.10.2021.
21. Scribd. 2021. MongoDB.local Chigago 2019: MongoDB Atlas Security 101 for Developers. Verkkoaineisto. Slideshare.
<<https://pt.slideshare.net/mongodb/mongodb-local-chicago-2019-mongodb-atlas-security-101-for-developers>>. Luettu.30.10.2021.

22. Heino, P. 2010. Pilvipalvelut. Hämeenlinna: Kariston Kirjapaino Oy.
23. rerkki, Eritec. 2019. Verkkoaineisto. Github.
<<https://github.com/rerkki/Ecitec/issues/1>>. Luettu.10.11.2021
24. Digi-Salama. 2020. Uudellamaalla Digi-Salamoi. Verkkoaineisto.
<<https://digisalama.metropolia.fi/hanke/>>. Luettu. 17.9.2021.