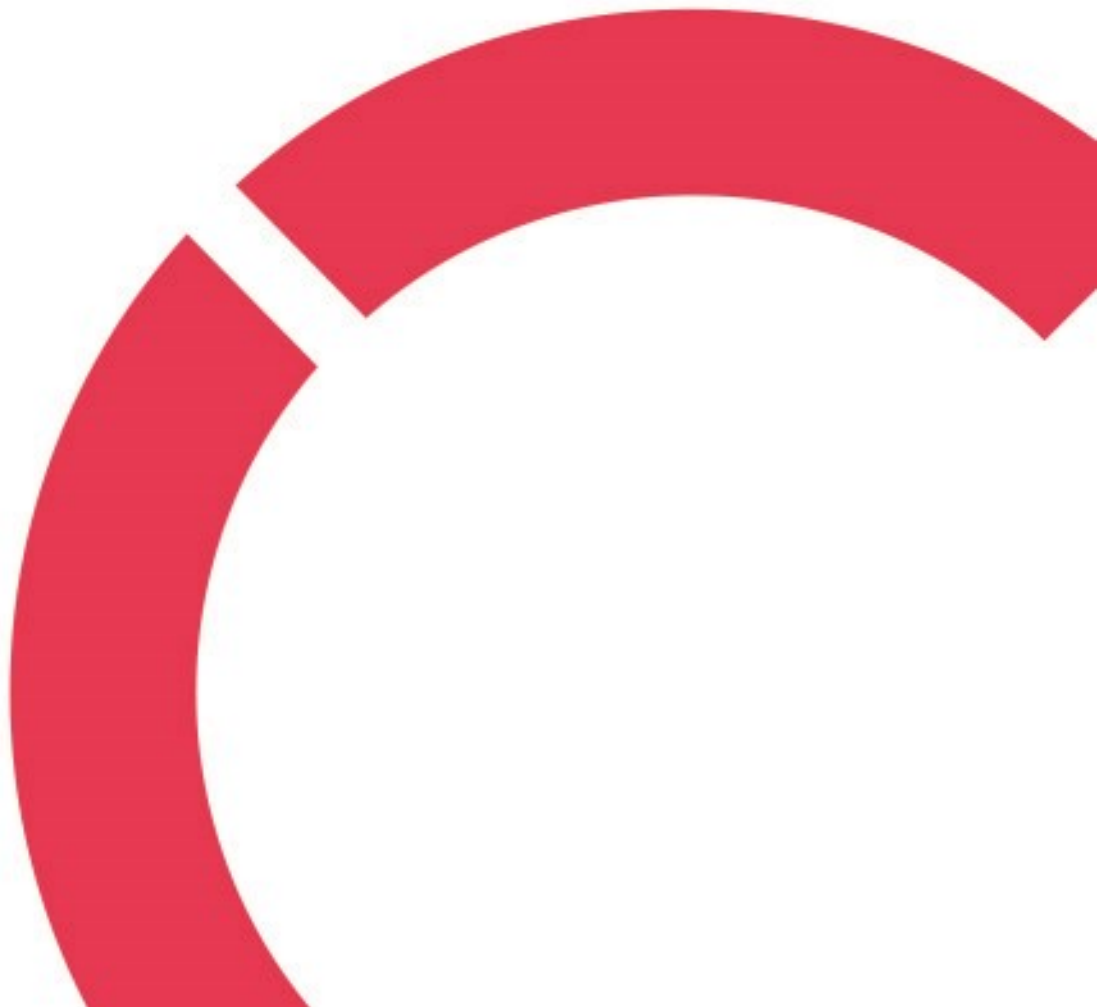


Riku Rampanen

IOT-YMPÄRISTÖN KÄYTTÖNOTTO

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tieto- ja viestintäteknikan koulutus
Joulukuu 2023**



TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	Aika Joulukuu 2023	Tekijä/tekijät Riku Rampanen
Koulutus Tieto-, ja viestintäteknikka		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
Työn nimi IoT-ympäristön käyttöönotto		
Työn ohjaaja Sari Lipsanen		Sivumäärä 22
<p>Erään yrityksen toimeksiannosta toteutettiin opinnäytetyö, jossa keskityttiin avoimen lähdekoodin IoT-alustan käyttöönottoon ja sen soveltamiseen erilaisiin käyttötarkoituksiin. Työn tavoitteena oli luoda monipuolinen IoT-ympäristö, joka integroi fyysisiä laitteita avoimen lähdekoodin ratkaisuihin ja mahdollistaa niiden hallinnan eri tekniikoiden avulla. Tutkimuksessa esiteltiin käytännön esimerkkejä, kuten säätietojen haku ja Raspberry Pi:n ohjaus IoT-alustan välityksellä, havainnollistaen avoimen lähdekoodin IoT-ratkaisujen laajoja sovellusmahdollisuuksia. Menetelminä hyödynnettiin sekä käytännön toteutuksia että kirjallista aineistoa. Työn tuloksena korostui avoimen lähdekoodin IoT-ratkaisujen monipuolisuus erilaisten käyttötapauksien toteuttamisessa ja niiden potentiaali monipuolisten sovellusten kehittämisessä. Johtopäätöksenä voidaan todeta, että avoimen lähdekoodin IoT-alustat tarjoavat vahvan pohjan innovatiivisille ja skaalautuville ratkaisuille eri toimialoilla.</p>		
Asiasanat Avoin lähdekoodi, IoT, Raspberry Pi		

ABSTRACT

Centria University of Applied Sciences	Date December 2023	Author Riku Rampanen
Degree programme Information Technology		
Name of thesis MAIN HEADING. Sub-heading if used		
Centria supervisor Sari Lipsanen		Pages 22
Instructor representing commissioning institution or company		
<p>The thesis, commissioned by a certain company, focused on the implementation of an open-source IoT platform and its application across various use cases. The aim was to establish a diverse IoT environment that integrates physical devices into open-source solutions, allowing their management through various techniques. The study showcased practical examples, such as fetching weather data and controlling Raspberry Pi via the IoT platform, illustrating the extensive application possibilities of open-source IoT solutions. Both practical implementations and literature were utilized as methodologies. The results emphasized the versatility of open-source IoT solutions in realizing diverse use cases and their potential in developing a wide range of applications. In conclusion, open-source IoT platforms serve as a robust foundation for innovative and scalable solutions across different industries.</p>		
Key words IoT, Raspberry Pi, Open Source		

KÄSITTEIDEN MÄÄRITTELY

Avoin lähdekoodi

Ohjelmisto, joka on julkaistu sellaisella lisenssillä, että sen lähdekoodi on julkisesti saatavilla. Tämä antaa mahdollisuuden tarkastella, muuttaa ja levittää ohjelmistoa kenelle tahansa.

Botti

Automaattinen tietokoneohjelma, joka suorittaa tiettyjä tehtäviä ilman ihmisen väliintuloa.

Docker

Alusta sovellusten kehittämiseen ja ajamiseen konttiympäristössä, joka mahdollistaa sovellusten paketoiminnan ja toimittamisen yhtenäisellä tavalla.

IoT

(Internet of Things) Viittaa laitteiden, kuten kodinkoneiden, sensoreiden ja muiden älykkäiden laitteiden, yhdistämiseen internetiin, jolloin niitä voidaan seurata, hallita ja kerätä dataa automaattisesti.

IoT-alusta

Ohjelmistotyökalu tai palvelu, joka mahdollistaa IoT-laitteiden yhdistämisen, hallinnan, datan keruun ja analysoinnin.

JavaScript

Ohjelmointikieli, jota käytetään pääasiassa verkkosivujen ja -sovellusten interaktiivisuuden lisäämiseen.

Python

Yleiskäyttöinen ohjelmointikieli, joka tunnetaan helppoudestaan ja monipuolisuudestaan.

Raspberry Pi

Kompakti ja edullinen tietokone, joka on suunniteltu opetuskäyttöön ja harrastusprojekteihin.

Telegram

Pikaviestisovellus, joka mahdollistaa viestien, äänen, videoiden ja tiedostojen lähettämisen.

UI

(User Interface) Käyttöliittymä, joka on rajapinta ihmisen ja tietokoneen tai muun laitteen välillä ja jon-ka avulla käyttäjä voi vuorovaikuttaa laitteen tai ohjelmiston kanssa.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

IOT-YMPÄRISTÖN KÄYTTÖÖNOTTO.....	1
1 JOHDANTO.....	1
2 MITÄ IOT TARKOITTAÄ	2
2.1 IoT historiaa	2
2.2 IoT teollisuudessa	3
2.3 IoT kuluttajilla	3
3 IOT-RAKENTEEN SUUNNITTELU.....	4
4 IOT-KOKONAISUUDEN OSAT.....	5
4.1 Raspberry Pi.....	5
4.2 Docker	5
4.3 Node-RED	5
4.4 Python ohjelma.....	5
4.5 Telegram	6
5 RASPBERRY PI.....	7
5.1 Käyttöjärjestelmä.....	7
5.2 Raspberry Pi käyttöönnotto	7
6 MIKÄ DOCKER ON.....	10
6.1 Docker asennus.....	10
6.2 Valinnaiset osat.....	11
6.2.1 Portainer	11
6.2.2 Portainer asennus.....	11
7 NODE RED.....	13
7.1 Node-RED asennus.....	13
7.2 Moduulit ja niiden asentaminen	14
8 PYTHON	15
9 TELEGRAM	17
9.1 Telegram-botti	17
9.2 Botin luominen ja käyttöönnotto	17
10 TOTEUTUS.....	19
11 POHDINTA.....	22
LÄHTEET	1
LÄHTEET	19

KUVAT

KUVA 1. IoT-rakenne suunnitteluvaiheessa	4
KUVA 2. Raspberry Pi Imager	8
KUVA 3. Docker info	10
KUVA 4. Portainer hallintapaneeli	12
KUVA 5. Node-RED asennus	13
KUVA 6. Node-RED hallintapaneeli.....	14
KUVA 7. loki.log tiedoston sisältö.....	16
KUVA 8. BotFather keskustelu	18
KUVA 9. Esimerkin 1 Node-RED näkymä.....	19
KUVA 10. Esimerkin 2 Node-RED näkymä.....	20

1 JOHDANTO

Tässä opinnäytetyössä käsitellään ja tutkitaan avoimen lähdekoodin IoT-alustan käyttämistä ja käyttöönottoa sekä sen erilaisten käyttötarkoitusten ja mahdollisuuksien hyödyntämistä. Tarkoituksena on suunnitella IoT-ympäristö, jossa yhdistyvät erilaiset fyysiset laitteet, IoT-alusta, erilaiset ulkoiset ohjelmat ja kaiken ohjaaminen käyttämällä erilaisia tekniikoita.

Työn keskeisenä kysymyksenä on, mitä IoT-alusta tarkoittaa ja kuinka se voidaan toteuttaa käyttämällä avoimen lähdekoodin työkaluja.

Työn tutkimustarve syntyi siitä, että Centria-ammattikorkeakoululle haluttiin tutkia mahdollisuutta toteuttaa ruokalassa lounaan maksaminen käyttämällä esimerkiksi kasvojentunnistusta perinteisen korttimaksamisen sijaan. Tutkimustarpeen yksi reunaehto oli, onko kyseistä suunnitelmaa mahdollista toteuttaa järkevästi, edullisesti ja tarkoitukseen räätälöitynä. Tämän pohjalta tuli selvittää, olisiko kyseinen suunnitelma mahdollista toteuttaa käyttämällä avoimen lähdekoodin ratkaisuja, jotka olisivat ilmaisia ja helposti muokattavissa haluttuun käyttötarkoitukseen.

Työssä selvitetään, mitä IoT ja IoT-alusta tarkoittavat. Kokonaisuuden on tarkoitus toimia konseptitodistuksena, eli esitellä IoT-ympäristön monipuolisuutta ja mahdollisuuksia mahdollisimman hyvin käyttämällä ilmaisia ja vapaasti käytettävissä olevia ohjelmistoja. Työssä esitellään yksinkertainen sää-tietojen haku sekä Raspberry Pi:n ohjaaminen ja valvonta käyttäen IoT-alustaa.

Tutkimuksen ensimmäisessä vaiheessa aihealuetta tutkitaan kirjallisuuden kautta ja laaditaan suunnitelma kokonaisuudesta, joka halutaan toteuttaa. Suunnitelman valmistuttua tutkitaan eri vaihtoehtoja toteuttaa haluttu lopputulos, jonka jälkeen valitaan työkalut ja aletaan rakentamaan kokonaisuutta suunnitelman pohjalta. Lopputulos saattaa työn edetessä muuttua alkuperäisestä suunnitelmasta, koska työn suunnitelma toimii työn karttana.

2 MITÄ IOT TARKOITTAÄ

IoT (Internet of Things), eli esineiden internet, tarkoittaa sitä, että useat eri laitteet, jotka ovat liitettynä tietoverkkoon, pystyvät kommunikoimaan keskenään verkkoyhteyden välityksellä automaattisesti ja niitä voidaan ohjailla ja tarkkailla etänä. Toisin kuin perinteisissä laitteiden välisissä yhteyksissä, IoT-ympäristössä eri laitteet voivat kommunikoida laajemmin keskinäisesti internet-yhteyden välityksellä, ja kyseessä voi olla hyvinkin erilaisia laitteita kuten antureita, palvelimia, ja arkipäiväisempiä kodin laitteita kuten jääkaappeja tai äänentoistolaitteita.

2.1 IoT historiaa

IoT, eli Internet of Things -konsepti, nimettiin virallisesti vasta vuonna 1999, vaikka aiempia kuvausta vastaavia laitteita oli jo aiemmin käytössä. Yksi aikaisempia esimerkkejä on 1980-luvulta, kun eräässä yhdysvaltalaisessa yliopistossa opiskelijat yhdistivät limsa-automaatin verkkoon, jotta he voisivat verkon yli tarkistaa, onko laitteessa tölkkejä ja ovatko ne kylmiä, ilman että heidän tarvitsisi mennä paikalle katsomaan tilannetta itse. (Foote 2022.)

2000-luvun alussa IoT tuli teollisuuden käyttöön käytännössä, kun erilaisia tuotteita ja laitteita alettiin merkitsemään RFID-siruilla, jolloin tietokoneiden oli helppo pitää niistä lukua ja hallita niitä. Ensimmäisiä vastaavanlaisia käyttökohteita olivat kauppaketju Walmart ja Yhdysvaltain puolustusvirasto. (Foote 2022.)

Vuonna 2011 IoT tavoitti tavalliset kuluttajat Ring-ovikellon muodossa. Kyseiseen ovikelloon on mahdollista ottaa yhteys internetin välityksellä, jolloin voidaan katsoa videokuvaa oveelta. (Foote 2022.)

Vuonna 2012 Sveitsissä aloitettiin pilottiohjelma ensimmäisen älykaupungin suunnittelussa, jonka tavoitteena oli vähentää jätteiden määrää ja maksimoida energian käyttö. (Foote 2022.)

Vuonna 2013 IoT alkoi olla osa tavallisten ihmisten jokapäiväistä elämää. Huomiota herättävimpinä esimerkkinä ovat sydämentahdistimet. (Foote 2022.)

2.2 IoT teollisuudessa

IoT mahdollistaa esimerkiksi teollisuuden alalla monimutkaisten työprosessien simuloinnin, eri työprosessien datan keräämisen tehokkaammin ja tehtaiden keskinäisen kommunikoinnin. Teollisuuden internetin kehittyessä prosessien hajauttaminen eri tehtaisiin tai laitoksiin helpottuu, eikä ole tarvetta tehdä kaikkea “saman katon alla”. Näin ollen tehtaiden ja prosessien muuttaminen ja laajentaminenkin helpottuu, koska muutoksia voidaan tehdä paikallisesti ja välittää tieto muutoksista reaaliaikaisesti tehtaan muille osille.

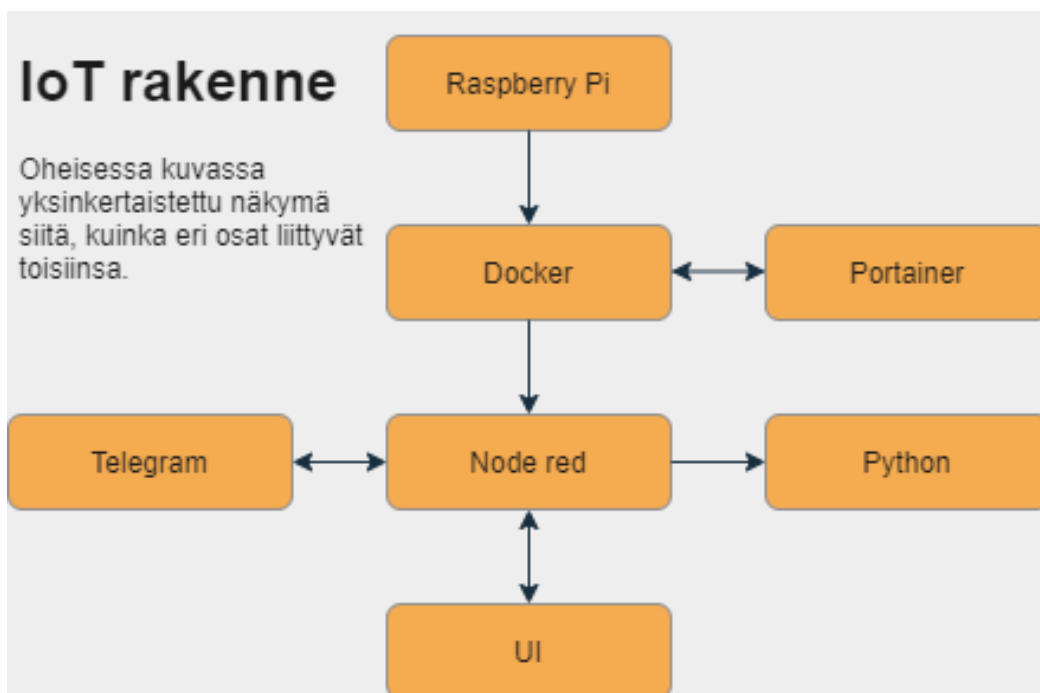
2.3 IoT kuluttajilla

Tavallisille kuluttajille IoT-käsite voi tuntua vieraalta, vaikka todellisuudessa yhä useampi nykyajan digitaalinen laite voidaan luokitella kuuluvaksi IoT:n piiriin. Yksinkertaisena esimerkkinä voisi olla älykello, joka kerää tietoa kuntoilun aikana ja siirtää tiedot erilaisiin pilvipalveluihin ja sitä kautta esimerkiksi kännykän sovellukseen.

3 IOT-RAKENTEEN SUUNNITTELU

IoT-ympäristön tulisi olla suunniteltu siten, että siihen on kytketty erilaisia antureita haluttua dataa varten, ja tätä dataa pystytään tarkastelemaan web-käyttöliittymän avulla. Ympäristön tulisi myös olla mahdollista automatisoida kerätyn datan perusteella. Esimerkkinä, jos lämpötila nousee tai laskee yli tietyn rajan, se lähettää ilmoituksen käyttäjälle sekä suorittaa haluttuja toimenpiteitä automaattisesti. Muita vaatimuksia, joita ympäristöön tulisi sisältyä, on mahdollisuus lähettää kerätty data myös ulkoiselle Python-ohjelmalle.

Suunnittelun ensimmäisessä vaiheessa määritellään koko ympäristön rakenne, joka helpottaa toteutuksen jokaista vaihetta toimimalla eräänlaisena karttana. Rakenteesta tulee selkeästi näkyä, kuinka eri laitteet ja rajapinnat kommunikoivat keskenään ja kuinka ne liittyvät toisiinsa. Kuvassa 1 on määritelty IoT-rakenne suunnitteluvaiheessa. (Rampanen 2021)



KUVA 1. IoT-rakenne suunnitteluvaiheessa. (Rampanen 2021)

4 IOT-KOKONAISUUDEN OSAT

Tässä osassa käsitellään lyhyesti työssä käytettyjä ohjelmistoja ja laitteistoja, niiden tehtävät sekä kuinka ne ovat osa rakennettua IoT-ympäristöä. Työssä on pyritty hyödyntämään useita erilaisia alustoja, joilla voidaan näyttää erilaiset IoT-mahdollisuudet.

4.1 Raspberry Pi

Raspberry Pi on erittäin pienikokoinen ja helppo ottaa käyttöön, joten se sopii mainiosti tämänkaltaiseen projektiin. Se toimii opinnäytetyössä IoT-ympäristön keskuksena eli hubina sekä palvelimena, jonka kautta kaikki eri osat kommunikoivat keskenään. (Opensource)

4.2 Docker

Opinnäytetyössä Dockeria käytetään Node-RED-ympäristön sekä sen liitännäisten konttisovellusten ajamiseen. Dockerin avulla eri osien asennusprosesseja voidaan selkeyttää huomattavasti. Lisäksi, jos on tarpeen lisätä tai poistaa eri osia, se onnistuu Dockerin avulla erittäin yksinkertaisesti. (Docker)

4.3 Node-RED

Vaikka tarjolla on useita avoimen lähdekoodin alustoja, kuten openremote, Node-RED valikoitui käytettäväksi alustaksi sen selkeyden ja käyttäjäystävällisyyden ansiosta. Node-RED ohjailee ja välittää tietoa eri IoT-osien välillä. (Node-RED)

4.4 Python ohjelma

IoT-ympäristöön on liitetty yksinkertainen Python-ohjelma, joka saa halutut argumentit ja datan Node-RED-alustan kautta. Tällä halutaan havainnollistaa, kuinka IoT-ympäristöä voi helposti laajentaa omiin tarpeisiin.

4.5 Telegram

IoT-alustan etäkäyttöä varten luodaan Telegram-pikaviestisovellukseen botti. Tämä botti pystyy ohjaimaan IoT-alustan osia vastaanottamalla komentoja sekä palauttamaan haluttuja tietoja käyttäjälle. Telegramia voidaan käyttää esimerkiksi ilmoitusten vastaanottoon, mikäli IoT-ympäristössä tapahtuu muutoksia tai jokin anturi saavuttaa tietyn arvon. (Telegram)

5 RASPBERRY PI

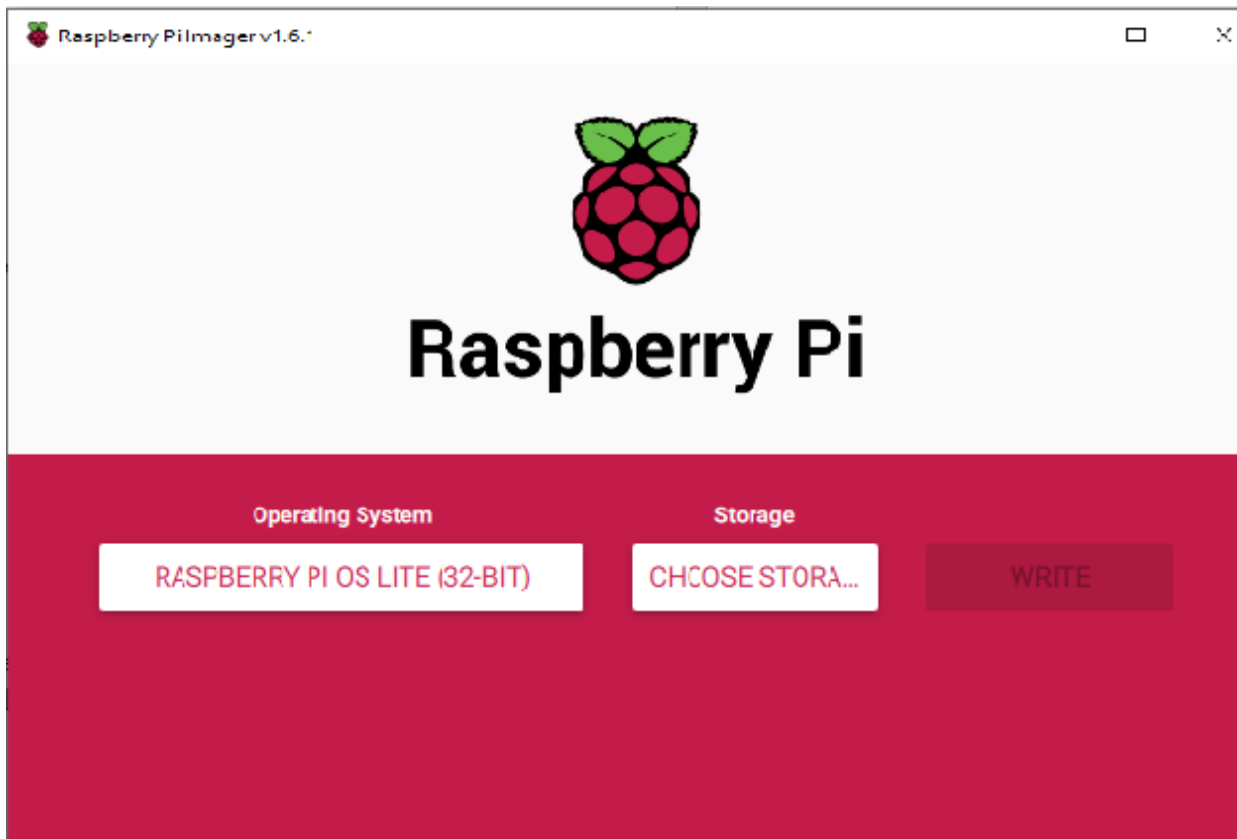
Raspberry Pi on erittäin pieni, noin luottokortin kokoinen tietokone, jossa kaikki komponentit mahtuvat yhdelle piirilevyllä. Pienen kokonsa ja edullisuutensa ansiosta se on erittäin suosittu erilaisissa harrasteprojekteissa sekä opetuskäytössä. (Opensource)

5.1 Käyttöjärjestelmä

Koska Raspberry Pi toimii tässä työssä ilman näyttöä, käyttöjärjestelmäksi valikoitui Raspberry Pi OS Lite. Tämä versio toimii ilman graafista työpöytäkymää. Valitsemalla tämän käyttöjärjestelmän asennuksen koko pysyy mahdollisimman pienenä ja vältetään tarpeettomia ominaisuuksia.

5.2 Raspberry Pi käyttöönotto

Käyttöjärjestelmän asennukseen käytetään Raspberry Pi Imager -ohjelmaa, joka kirjoittaa halutun käyttöjärjestelmän ulkoiselle MicroSD-kortille. Ensiasennukseen tarvitaan myös näyttö ja näppäimistö, jotka voidaan asennuksen jälkeen irrottaa. Jatkossa konfigurointi voidaan suorittaa käyttämällä SSH-yhteyttä. Kuvassa 2 on määritelty Raspberry Pi Imager käyttö. (Rampanen 2021)



KUVA 2. Raspberry Pi Imager. (Rampanen 2021)

Kun muistikortti on valmisteltu, laitteeseen voidaan kytkeä näppäimistö, näyttö, verkkokaapeli ja virtalähde. Ensimmäisen käynnistyksen yhteydessä käyttöjärjestelmä suorittaa vielä viimeisiä asennuksia, jonka jälkeen laitetta voi suoraan alkaa käyttämään kirjautumalla sisään oletustunnuksilla:

- **username:** pi
- **password:** raspberry

Ensimmäisiin suositeltuihin toimiin kuuluu varmistaa, että järjestelmä on ajan tasalla päivittämällä se, vaihtaa oletussalasana - mikä on huomattava tietoturvariski, kun laite liitetään verkkoon - sekä sallia SSH-yhteys etäkäyttöä varten. Kaikki operaatiot ovat helposti suoritettavissa muutamalla komennolla, jotka voidaan kirjoittaa suoraan avoimna olevalle komentoriville.

Järjestelmän ajantasaisuuden varmistaminen ja päivittäminen komentoriviltä:

- **sudo apt-get update**
Tarkistaa onko järjestelmään tarjolla uusia päivityksiä.
- **sudo apt-get upgrade**
Asentaa mahdolliset päivitykset.

Salasanan muuttaminen:

- Kirjoita komentoriville: **passwd** ja paina enter.
- Järjestelmä kysyy nykyisen salasanan(**raspberry**)
- Järjestelmä pyytää antamaan uuden salasanan.
- Järjestelmä pyytää uuden salasanan uudestaan.

Mikäli salasana vaihdettiin onnistuneesti, ilmoittaa järjestelmä siitä viestillä:

- **passwd: password updated succesfully.**

SSH-yhteyden salliminen on niin ikään erittäin helppo suorittaa antamalla seuraavat komennot:

- **sudo systemctl enable ssh**
- **sudo systemctl start ssh**

Tämän jälkeen voit hakea Raspberry Pi:n IP-osoitteen SSH-yhteyttä varten komennolla:

- **hostname -I**

Kun edellä mainitut toiminnot on suoritettu, voidaan irrottaa Raspberry Pi:stä näyttö, hiiri ja näppäimistö ja jatkaa käyttöä SSH-yhteyden välityksellä.

6 MIKÄ DOCKER ON

Docker pakkaa halutun sovelluksen ja sen liitännäiset osat virtuaaliseen säiliöön (konttiin), jota voidaan ajaa millä tahansa käyttöjärjestelmällä, esimerkiksi MacOS, Linux tai Windows. Verrattuna perinteiseen virtuaaliympäristöön, kuten Oraclen VM VirtualBoxiin, Docker on huomattavasti vähemmän tehoa vaativa, käyttäjäystävällisempi, ja sen avulla voidaan ajaa useita virtuaaliympäristöjä samanaikaisesti. (Docker)

6.1 Docker asennus

Dockerin asennus Raspberry Pi:lle suoritetaan käyttämällä komentoriviä seuraavasti:

- **curl -fsSL https://get.docker.com -o get-docker.sh**

Komento lataa Convenience-skriptin, jonka tarkoituksena on yksinkertaistaa Dockerin asennus.

- **sudo sh get-docker.sh**

Komento suorittaa skriptin, joka asentaa Dockerin ja sen tarvittavat paketit. Kun asennus on valmis, terminaali näyttää asennetun Dockerin version sekä muita tietoja asennuksesta. Kuvassa 3 on määritelty Docker asennuksen varmistus. (Rampanen 2021)

```

pi@raspberrypi:~$ docker version
Client: Docker Engine - Community
Version: 20.10.6
API version: 1.41
Go version: go1.13.15
Git commit: 370c289
Built: Fri Apr 9 22:46:18 2021
OS/Arch: linux/arm
Context: default
Experimental: true

Server: Docker Engine - Community
Engine:
Version: 20.10.6
API version: 1.41 (minimum version 1.12)
Go version: go1.13.15
Git commit: 8728dd2
Built: Fri Apr 9 22:44:17 2021
OS/Arch: linux/arm
Experimental: false
containerd:
Version: 1.4.4
GitCommit: 05f951a3781f4f2c1911b05e61c160e9c30eaa8e
runc:
Version: 1.0.0-rc93
GitCommit: 12644e614e25b05da6fd08a38ffa0cfe1903fdec
docker-init:
Version: 0.19.0
GitCommit: de40ad0
pi@raspberrypi:~$

```

KUVA 3. Docker info. (Rampanen 2021)

6.2 Valinnaiset osat

Vaikka Dockeria voidaan käyttää SSH-yhteyden välityksellä ja komentorivin kautta, monille on mielekkäämpää ja yksinkertaisempaa käyttää graafista käyttöliittymää. Tästä syystä seuraavaksi asennetaan Dockeriin Portainer-niminen säiliö (kontti), joka tarjoaa Dockerille selainpohjaisen käyttöliittymän.

6.2.1 Portainer

Portainer on avoimen lähdekoodin työkalu konttisovellusten hallintaan. Se yksinkertaistaa erilaisten konttisovellusten hallintaa tarjoamalla käyttöliittymän, joka on graafinen. Tämä tukee ajatusta siitä, että IoT-ympäristön tulisi olla mahdollisimman helppokäyttöinen ja havainnollinen. (Portainer n.d)

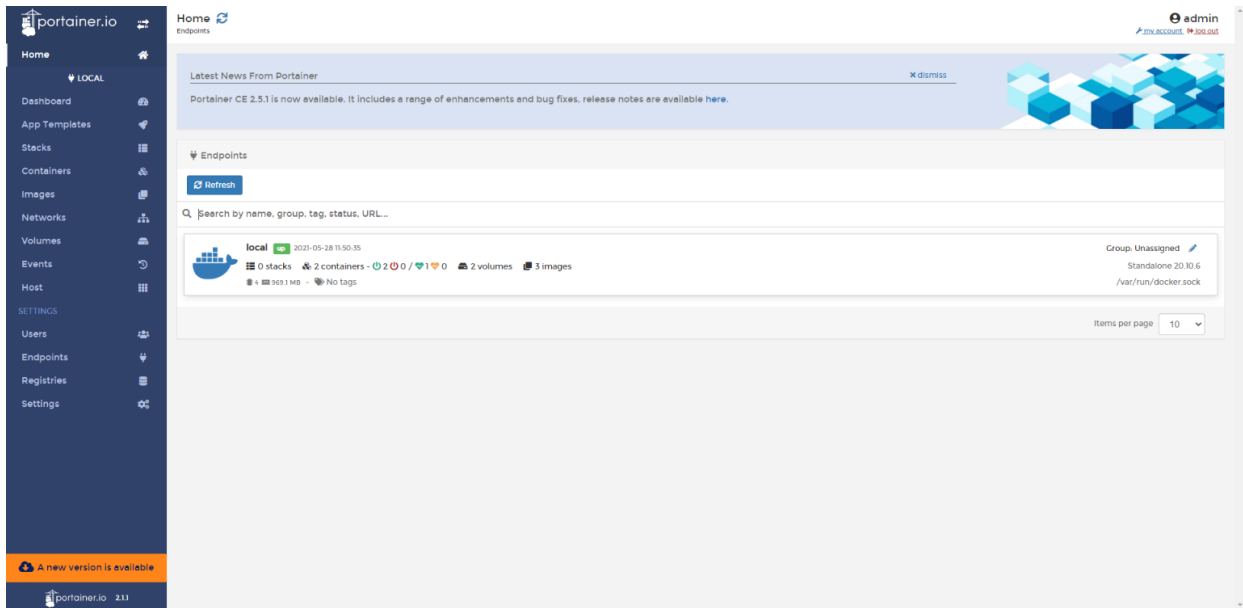
6.2.2 Portainer asennus

Portainerin asennus Dockeriin tapahtuu seuraavasti:

- **sudo docker pull portainer/portainer-ce:linux-arm**
Tämä komento lataa Portainerin imagen Raspberry Pi:lle.
- **sudo docker run -d -p 9000:9000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:linux-arm**

Komento asentaa Portainerin, avaa portin 9000 käyttöä varten, nimeää kontin "portainer" ja määrittelee Portainerin käynnistymään aina, kun Docker käynnistetään.

Kun asennus on valmis, Portaineria voi käyttää avaamalla selaimessa osoitteen **http://[Raspberry Pi:n IP-osoite]:9000**. Kuvassa 4 on kuvattu Portainer sovelluksen käyttöliittymä. (Rampanen 2021)



KUVA 4. Portainer hallintapaneeli. (Rampanen 2021)

7 NODE RED

Node-RED on avoimen lähdekoodin ohjelmointityökalu, joka mahdollistaa erilaisten fyysisten laitteiden, kuten lämpötila-antureiden, ohjelmointirajapintojen ja online-palveluiden yhdistämisen IoT-ympäristöön.

Vaikka Node-RED käyttää ohjelmointikielenään JavaScriptiä, se ei vaadi paljoa perinteistä ohjelmointia. Sen sijaan se keskittyy visuaaliseen ohjelmointiin, jossa suurin osa logiikasta toteutetaan virtakaa-vion tyyliin liitettävillä moduuleilla. Tämä tekee yhteyksien ja logiikan hahmottamisesta intuitiivista. (Node-RED)

7.1 Node-RED asennus

Node-RED:n asennus Docker-ympäristöön tapahtuu seuraavalla komennolla:

- **docker run -it -p 1880:1880 -v node_red_data:/data --name mynodered nodered/node-red**
Komento käynnistää Node-RED kontin tai luo sellaisen, jos sitä ei ole olemassa. Se myös avaa portin 1880 ja antaa kontille nimen "mynodered". Kuvassa 5 on esitelty Node-RED ympäristön käynnistäminen. (Rampanen 2021)

```

Welcome to Node-RED
=====
12 May 09:03:11 - [info] Node-RED version: v1.3.4
12 May 09:03:11 - [info] Node.js version: v10.24.1
12 May 09:03:11 - [info] Linux 5.4.72-microsoft-standard-WSL2 x64 LE
12 May 09:03:11 - [info] Loading palette nodes
12 May 09:03:12 - [info] Dashboard version 2.29.0 started at /ui
12 May 09:03:12 - [info] Settings file : /data/settings.js
12 May 09:03:12 - [info] Context store : 'default' [module=memory]
12 May 09:03:12 - [info] User directory : /data
12 May 09:03:12 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 May 09:03:12 - [info] Flows file : /data/flows.json
12 May 09:03:12 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

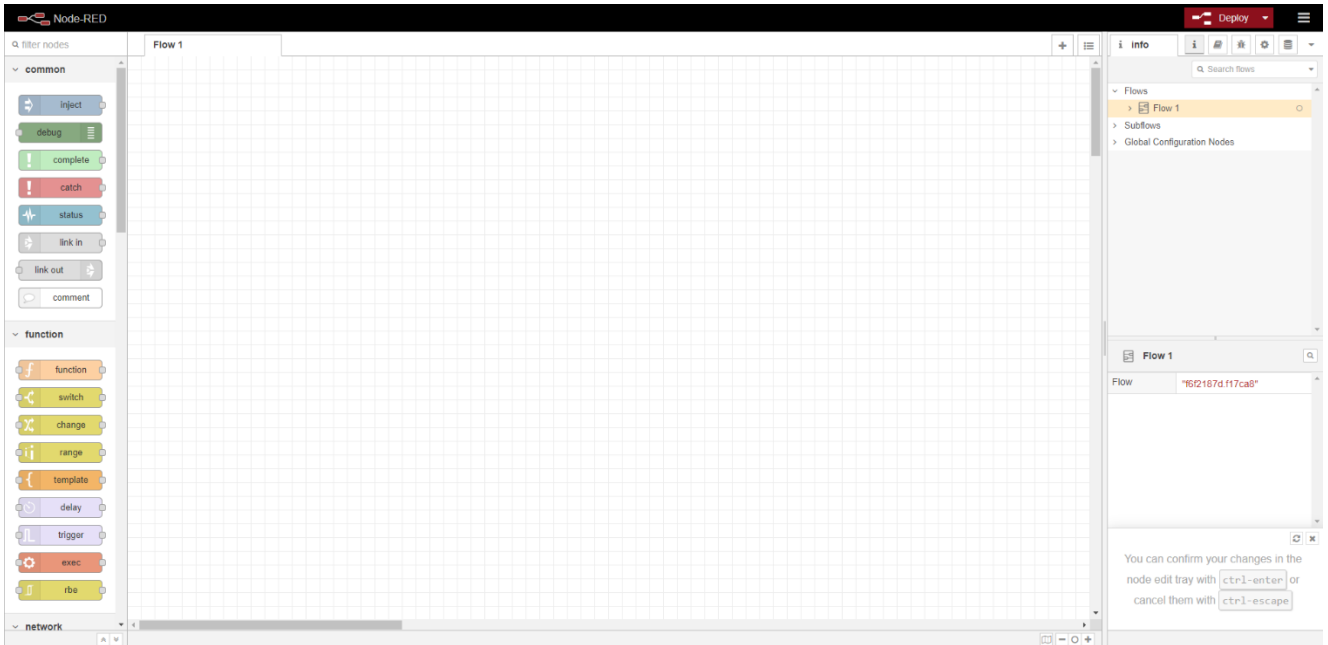
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
12 May 09:03:12 - [info] Server now running at http://127.0.0.1:1880/
12 May 09:03:12 - [info] Starting flows

```

KUVA 5. Node-RED asennus. (Rampanen 2021)

Kun Node-RED:n asennus on valmis ja ohjelma on käynnissä, pääset sen graafiseen käyttöliittymään avaamalla selaimessa seuraavan osoitteen:

- **[http://\[Raspberry Pi:n osoite\]:1880](http://[Raspberry Pi:n osoite]:1880)**. Kuvassa 6 on määritelty Node-RED käyttöliittymä. (Rampanen 2021)



KUVA 6. Node-RED hallintapaneeli. (Rampanen 2021)

7.2 Moduulit ja niiden asentaminen

Node-RED:iin on saatavilla paljon avoimen lähdekoodin moduuleja, joilla voidaan lisätä ominaisuuksia. Moduuleja voi myös itse kirjoittaa. Asentaminen tapahtuu joko komentoriviltä tai hallintapaneelin kautta. Työssä käytettyjä moduuleja ja niiden käyttötarkoitus osana tätä työtä:

- **node-red-contrib-pythonshell**
Mahdollistaa Python-skriptien suorittamisen Node-RED:stä.
- **node-red-contrib-telegrambot**
Node, jonka avulla voidaan lähettää ja vastaanottaa viestejä Telegram-botille.
- **node-red-node-openweathermap**
Node, jolla voidaan hakea tietoa OpenWeatherMap:sta.

8 PYTHON

Python on yksinkertainen ja helposti opittava ohjelmointikieli, joka taipuu moniin käyttötarkoituksiin sen monipuolisuuden ja korkean tason tietorakenteiden vuoksi. Python valikoitui osaksi tätä opinnäytetyötä sen helppokäyttöisyyden ja monipuolisuuden takia. Pythonilla voidaan helposti kirjoittaa erilaisia pieniä ohjelmia, jotka helpottavat automaatiota, tai sillä voidaan luoda web-sovelluksen back-end.

Työssä varsinaisen python-ohjelman osuus on erittäin pieni ja ohjelma on yksinkertainen, koska tarkoituksena on tutkia, kuinka python-ohjelmia voidaan liittää osaksi kokonaisuutta ja kuinka niille voidaan ohjata tietoa esimerkiksi antureista.

Alla oleva koodi suoritetaan aina kun paikallista lämpötilaa haetaan käyttämällä web-liittymästä löytyvää nappia tai lähettämällä komento Telegramin kautta. Ohjelma vastaanottaa argumenttina lämpötilan arvon ja kirjoittaa sen talteen loki.log tiedostoon.

```
#coding: utf-8
import sys
from datetime import datetime

c = sys.argv[1]
d = datetime.now()

with open('loki.log', 'a') as file:
    file.write("Lämpötila(C): '{0}' - Päivämäärä:
' {1}'\n".format(c, d))
```

Tässä lyhyessä koodissa Node-RED:n lähettämä lämpötilan arvo tallennetaan muuttujaan c ja sen hetkinen kellonaika muuttujaan d. Tämän jälkeen ohjelma avaa tiedoston loki.log, tai luo sellaisen, jos sitä ei ole, ja kirjoittaa saadun lämpötilan ja kellonajan määritetyllä tavalla tiedostoon uudelle riville. Kuvassa 7 on esitelty python skriptin suorittama logi tiedosto. (Rampanen 2021)

```

GNU nano 4.6                               loki.log
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:06.855088'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:07.588786'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:08.349880'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:09.024454'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:09.921008'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:11.366250'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:13.082138'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:13.910338'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:14.604134'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:15.380536'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:16.107742'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:16.791183'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:25.873996'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:26.772991'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:27.508968'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:28.149000'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:28.767852'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:29.326254'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:29.933790'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:30.453960'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:30.725178'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:49:30.965309'
Lämpötila (C): '22' -- Päivämäärä: '2021-06-17 10:53:24.963703'
Lämpötila (C): '14.2' -- Päivämäärä: '2021-06-23 05:39:52.458454'
Lämpötila (C): '14.6' -- Päivämäärä: '2021-06-23 07:07:10.148939'
Lämpötila (C): '15.5' -- Päivämäärä: '2021-06-28 05:29:25.717225'

^G Get Help      ^C Write Out    ^R Where Is     ^X Cut Text      ^J Justify      ^_ Cur Pos      M-U Undo        M-A Mark Text   M-] To Bracket
^O Exit          ^R Read File    ^N Replace      ^P Paste Text   ^Z To Spell    ^_ Go To Line   M-E Redo        M-C Copy Text   ^C Where Was

```

KUVA 7. loki.log tiedoston sisältö. (Rampanen 2021)

9 TELEGRAM

Telegram on pikaviestintäsovellus, joka mahdollistaa viestien, kuvien, videoiden ja tiedostojen lähettämisen yksityisesti tai ryhmissä. Se tarjoaa salatun viestinnän ja monia ominaisuuksia, kuten salaiset keskustelut, teemat, bottien käytön ja kanavat. Telegram on tunnettu myös avoimesta API:staan, mikä mahdollistaa kehittäjille sovellusten rakentamisen sen päälle. Lisäksi se on käytettävissä useilla eri alustoilla, kuten puhelimilla, tietokoneilla ja selaimilla. (Telegram)

9.1 Telegram-botti

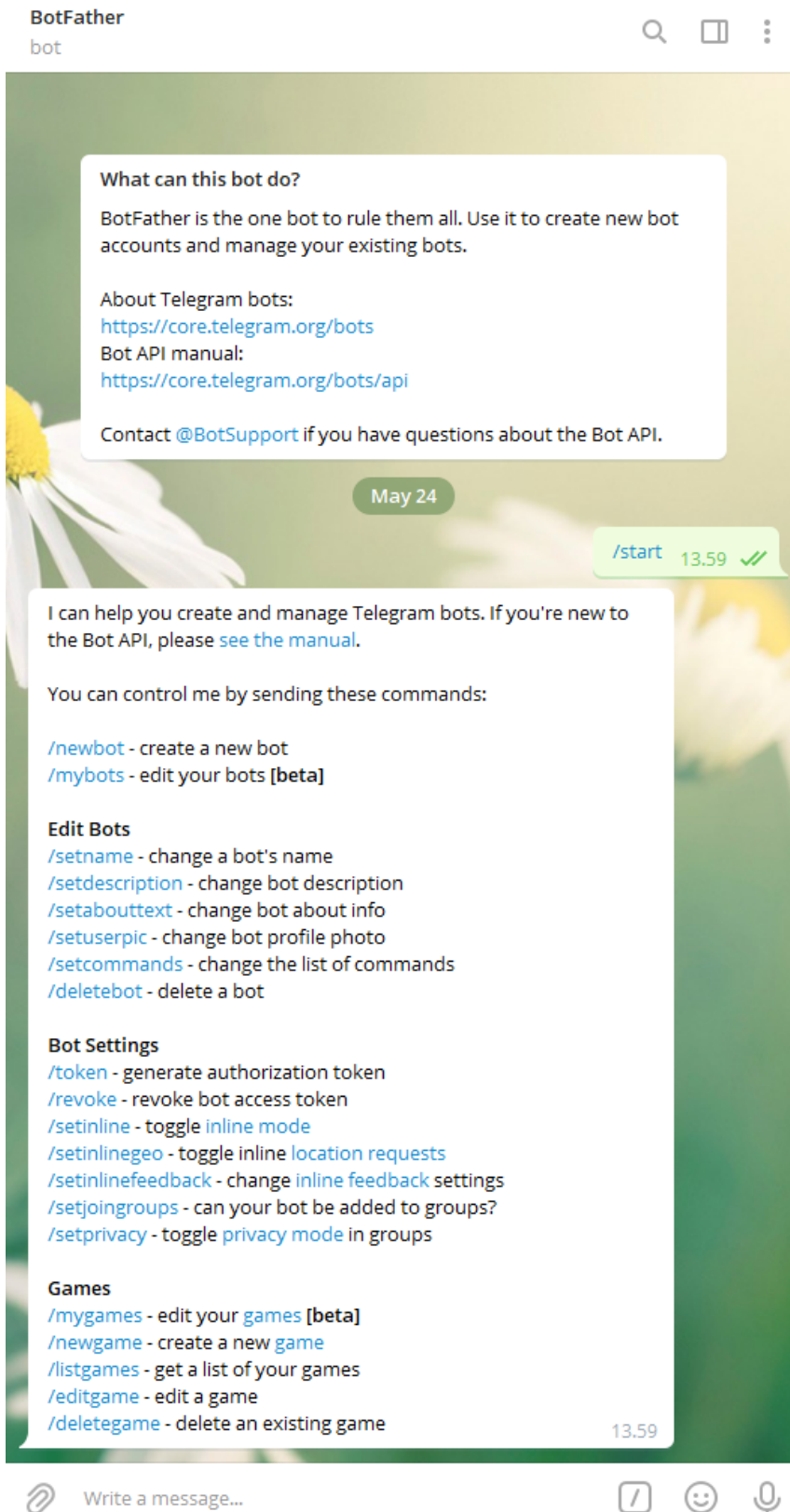
Telegram-botti on kolmannen osapuolen sovellus, joka toimii Telegram-sovelluksen sisällä. Bottia voidaan ohjata lähettämällä sille komentoja chatin avulla. Osana tätä opinnäytetyötä luodaan oma botti, jolla voidaan ohjata Node-RED-alustaa ja vastaanottaa sieltä takaisin viestejä.

9.2 Botin luominen ja käyttöönotto

Botin luominen on erittäin suoraviivaista ja helppoa, sillä Telegramissa on valmiina botti, joka auttaa uusien bottien luomisessa. Botin luomiseksi ja hallitsemiseksi tarvitsee vain aloittaa uusi keskustelu Telegramin BotFatherin kanssa ja käyttää komentoa /newbot.

Tämän jälkeen annetaan botille nimi ja käyttäjätunnus lähettämällä keskustelussa viesti halutulla nimellä ja käyttäjätunnuksella. Kun botin luonti on valmis, lähettää BotFather sinulle viestin, joka sisältää access tokenin. Tätä tokenia tarvitset käyttääksesi botin HTTP API:a.

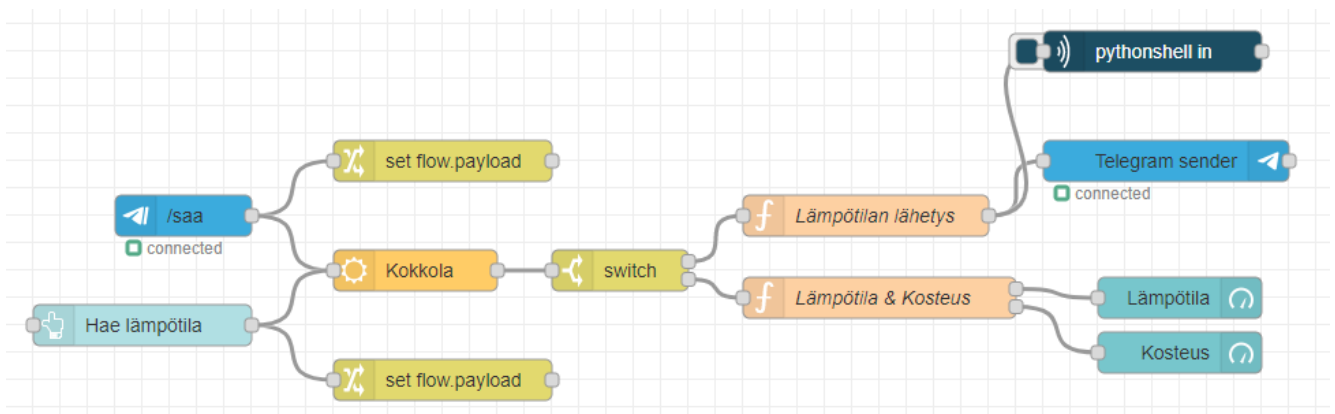
Botin muokkaaminen myöhemmin onnistuu lähettämällä viestejä samaan keskusteluun, jossa botti alun perin luotiin. Kuvassa 8 on esitelty Telegram-botin käyttöönotto. (Rampanen 2021)



KUVA 8. BotFather keskustelu. (Rampanen 2021)

10 TOTEUTUS

Työssä on esimerkkinä tehty kaksi erilaista IoT-sovellutusta, joiden avulla esitetään muutamia tapoja yhdistää eri laitteita ja ympäristöjä toisiinsa. Ensimmäisessä esimerkissä esitetään, kuinka voidaan hakea säätiedot ja esittää ne web-liittymässä, tai lähettää ne käyttäjälle viestinä käyttäen Telegram-sovellusta, sekä tallentaa haetut tiedot erilliseen lokitiedostoon lyhyen python-ohjelman avulla. Kuvassa 9 on määritelty Node-RED flow kaavio esimerkille 1. (Rampanen 2021)



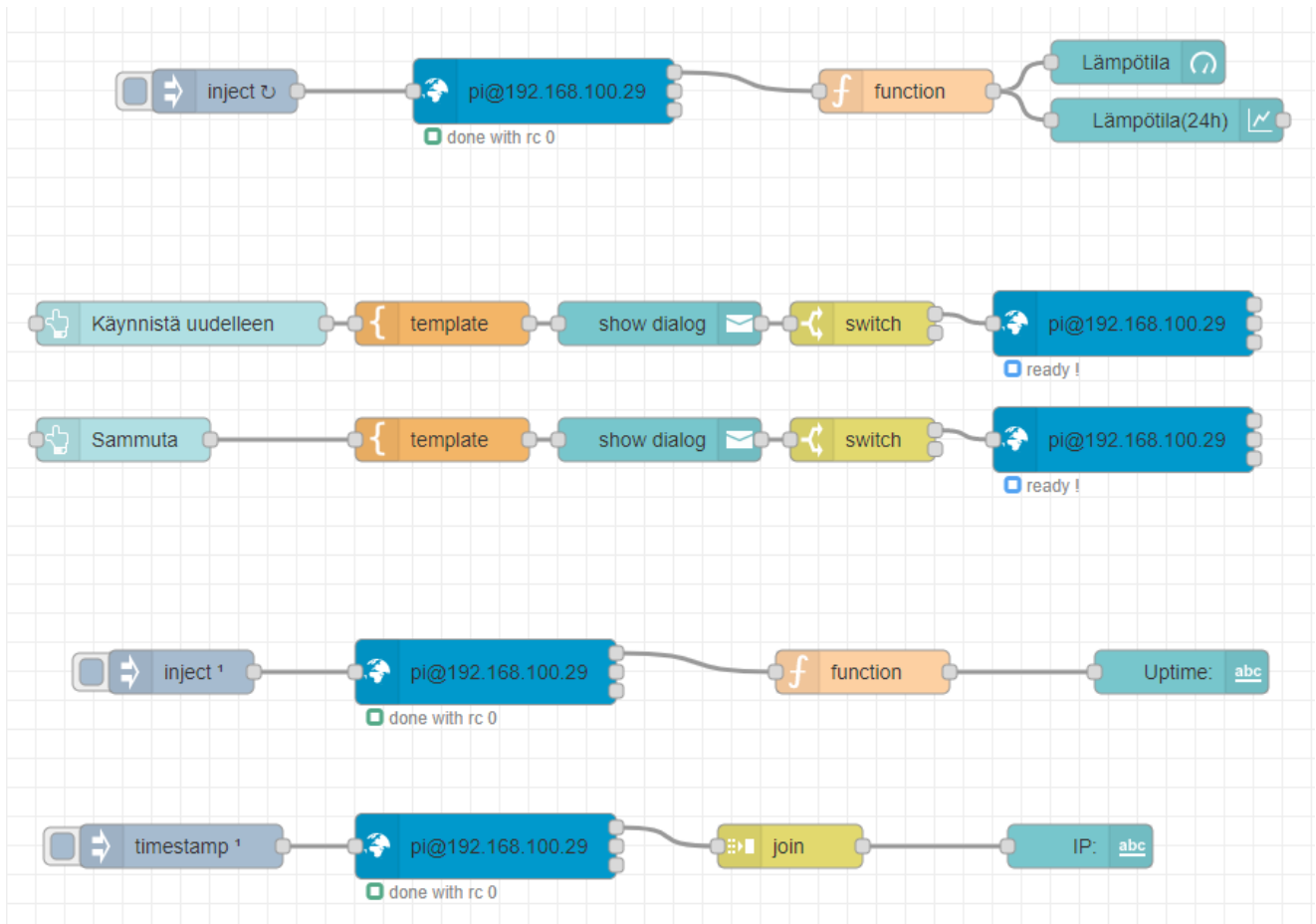
KUVA 9. Esimerkin 1 Node-RED näkymä. (Rampanen 2021)

Yllä olevassa kuvassa on kuvattu ensimmäisen esimerkin toiminta Node-RED:n flow-ohjelmointia käyttäen. Alussa käyttäjä joko painaa web-käyttöliittymästä löytyvää nappia tai lähettää Telegram-bottille viestin /saa, joka saa ohjelman hakemaan säätiedot. Riippuen siitä, painettiinkö nappia vai lähetettiinkö viesti, ohjelma näyttää halutut tiedot käyttäjälle.

Jos käyttäjä päättää lähettää viestin, ohjelma hakee paikallisen lämpötilan ja ohjaa ulostulon Telegram-nodelle, joka vastaa käyttäjän viestiin halutulla sisällöllä. Tässä tapauksessa ohjelma lähettää käyttäjälle viestin, jossa kerrotaan Kokkolan tämänhetkinen lämpötila. Käyttäjän painaessa web-käyttöliittymästä löytyvää nappia, ohjelma hakee paikallisen lämpötilan ja suhteellisen ilmankosteuden, esittäen ne graafisesti samalla sivulla.

Ensimmäistä esimerkkiä olisi helppo laajentaa esimerkiksi niin, että palvelimeen liitetään erilaisia antureita, jotka mittaavat eri huoneiden lämpötiloja ja kosteuksia.

Toisessa esimerkissä on tehty yksinkertainen käyttöliittymä, jolla voidaan seurata ja hallita työssä käytettyä Raspberry Pi:tä web-liittymän kautta. Alla olevassa kuvassa on esitetty toisen esimerkin toiminta Node-RED:n flow-ohjelmointia käyttäen. Kuvassa 10 on määritelty Node-RED flow kaavio esimerkille 2.



KUVA 10. Esimerkin 2 Node-RED näkymä. (Rampanen 2021)

Ohjelma näyttää web-käyttöliittymässä palvelimen, tässä tapauksessa Raspberry Pi:n, tämänhetkisen lämpötilan, lämpötilan vaihtelut viimeisen vuorokauden ajalta, yhtäjaksoisen käynnissä oloajan sekä paikallisen ja ulkoisen IP-osoitteen. Käyttöliittymässä on myös painikkeet, joilla palvelin voidaan käynnistää uudelleen tai sammuttaa. Koska Node-RED-alustaa ajetaan konttisovelluksena, sillä ei ole suoraa pääsyä palvelimeen. Tämän vuoksi työssä on hyödynnetty SSH-yhteyttä, ja uudelleenkäynnistys- sekä sammutuskomennot lähetetään tämän yhteyden välityksellä palvelimelle.

Toista esimerkkiä voisi laajentaa esimerkiksi niin, että kun palvelimen lämpötila ylittää tietyn rajan, se lähettää varoituksen käyttäjälle. Jos lämpötila jatkaa nousuaan, palvelin sammuttaisi itsensä automaattisesti vahinkojen välttämiseksi. Käyttöliittymässä olisi myös mahdollista esittää monia muita palvelimeen liittyviä tietoja.

11 POHDINTA

Saatuani työn valmiiksi olin osittain tyytyväinen lopputulokseen. Aiheesta oli mielestäni sopivasti tietoa tarjolla, mutta koska aihepiiri oli erittäin laaja ja minulla ei varsinaisesti ollut konkreettista käyttökohdetta, työn toteutuksessa oli vaiheittain haasteita pitää se helposti ymmärrettävänä ja selkeänä kokonaisuutena.

Koen, että jos minulla olisi ollut alussa selkeämpi suunta ja todellinen tarve, työn rajausta ja toteutusta olisivat olleet selkeämpiä. Kaiken kaikkiaan olen kuitenkin tyytyväinen lopputulokseen, joka mielestäni palvelee alussa asetettua tavoitetta. Tavoitteenani oli osoittaa IoT-alustan monipuoliset käyttökohteet ja sen suhteellinen helppous käyttämällä ilmaisia ohjelmistoja.

Koska opinnäytetyö on ollut pitkäkestoinen projekti, IoT-sovellukset ovat myös kehittyneet valtavasti, ja niistä olisi voinut rakentaa monenlaisia mielenkiintoisia kokonaisuuksia.

LÄHTEET

Docker. Dockerin asennus Raspberry Pi OS:lle. Saatavilla:

<https://docs.docker.com/engine/install/raspberry-pi-os/>

Foote, K.D. 2022. A Brief History of the Internet of Things. Dataversity. Saatavissa:

<https://www.dataversity.net/brief-history-internet-things/>. Viitattu 29.11.2023.

Node-RED. Running under Docker: Node-RED. Saatavilla: [https://nodered.org/docs/getting-](https://nodered.org/docs/getting-started/docker)

[started/docker](https://nodered.org/docs/getting-started/docker)

Opensource. What is Raspberry Pi? Saatavilla: [https://opensource.com/resources/raspberry-](https://opensource.com/resources/raspberry-pi)
[pi](https://opensource.com/resources/raspberry-pi)

Portainer. Portainer Knowledge Base. Saatavilla: <https://portal.portainer.io/knowledge>

Rampanen, R. 2021. Erilaisia tapoja yhdistää IoT-laitteita ja -ympäristöjä toisiinsa. Opinnäytetyön toiminnallinen osuus. 2021.

Telegram. Bots: An introduction for developers. Saatavilla: <https://core.telegram.org/bots>