jamk.fi

Langaton sensoriverkko lämpötilojen seurantaan

Jonne Vuorela

Opinnäytetyö
Toukokuu 2018
Tekniikan ja liikenteen ala
Insinööri (AMK), Tieto- ja viestintätekniikan tutkinto-ohjelma
Verkkotekniikka

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Vuorela, Jonne	Opinnäytetyö, AMK	Toukokuu 2018
	Sivumäärä 64	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Langaton sensoriverkko lämpötilojen seurantaan

Tutkinto-ohjelma
Tieto- ja viestintätekniikka, Tietoverkkotekniikka

Työn ohjaaja(t)
Rantonen Mika, Kokkonen Tero

Toimeksiantaja(t)
Vuorela Jonne

Tiivistelmä

Internet of Things (IoT) -laitteiden ja erilaisten ratkaisujen määrä on kasvanut kovaa vauhtia viimeisien vuosien aikana. Erilaisia toteutuksia kehitetään koko ajan lisää niin eri alojen yritysten, kuin yksityishenkilöidenkin toimesta. IoT mahdollistaa uudenlaisen, tarkemman ja helpomman seurannan ja hallinnan monessa erilaisessa tilanteessa. Laitteiden kasvava saatavuus ja laskevat hinnat ovat luultavasti suurimmat syyt kasvavalle kiinnostukselle toteuttaa erilaisia ratkaisuja IoT:n avulla.

Markkinoilta löytyy suuri määrä valmiita paketteja erilaisiin IoT-ratkaisuihin ja käyttötarkoituksiin. Opinnäytetyö toi yhden ratkaisun lisää. Tavoitteena oli rakentaa langaton sensoriverkko talviasuttavan mökin lämpötilojen seurantaan kolmesta eri sijainnista 50 metrin etäisyydellä toisistaan. Laitteistona haluttiin käyttää yleisesti käytössä olevia sensoreita, antureita ja muita laitteita. Alhaiset hankintakustannukset olivat yksi kriteeri työn toteuttamiselle.

Sensoriverkko rakennettiin käyttäen yhden piirilevyn tietokoneita, lämpötila-antureita ja radiolähettimiä. Lisäksi lämpötiladata varastoitiin Amazonin pilvipalveluun, josta se on seurattavissa reaaliaikaisesti. Sen avulla historian selaus on myös mahdollista. Huomioon otettiin myös mahdolliset tulevaisuudessa tehtävät sensorilisäykset ja niiden skaalautuvuus käyttöön.

Sensoriverkon avulla kerättyä dataa säilötään pidemmältä aikaväliltä. Datan avulla on mahdollista ennustaa jatkossa, miten eri vuodenajat ja säätilojen muutokset vaikuttivat mökin sisälämpötilojen muutoksiin. Tämä tieto mahdollistaa ennakoivan lämpötilojen säätämisen, jolloin hukkaenergiaa syntyy mahdollisimman vähän.

Avainsanat (asiasanat)
IoT, Raspberry Pi, Arduino, DHT11, NRF24L01, Amazon EC2

Muut tiedot

jamk.fi

Description

Author(s)	Type of publication	Date
Vuorela, Jonne	Bachelor's thesis	May 2018
		Language of publication: Finnish
	Number of pages: 64	Permission for web publication: x

Title of publication

Wireless sensor network for temperature measurement

Degree programme

Information and Communication Technology, Data Network Technology

Supervisor(s)

Rantonen Mika, Kokkonen Tero

Assigned by

Vuorela Jonne

Abstract

The number of IoT devices is increasing incredibly fast. There are more and more various IoT solutions available every day. Those solutions are mainly designed by companies nevertheless there is also a great amount of solutions from amateurs. IoT enables new kind of measuring, observation and automation. It is used in industrial companies mostly in their own production processes, a few projects in SMEs, home automation and learning. The growing number of devices and decreasing prices are certainly a good reason for interest in using IoT in versatile cases.

There are many different kinds of products on the market today. The thesis delivers up one more case. The goal was to make working wireless sensor network for measuring temperatures of a cottage. There were three different physical locations where sensors had to measure temperature data and send it to a gateway and Thingsboard. The distance between sensors was about 50 meters. One requirement for the devices was their low price.

The sensor network was built with single-board computers, temperature- and humidity sensors and radio transmitters. The data was stored in Amazon's cloud service and it was available everywhere via the Internet. Additionally any future additions to the sensor network were taken into account.

The measured data from the sensor network is to be stored for longer periods of time. With this information it is possible to see how outside temperature affects the indoor temperature. The information could be used to control and proactively adjust the heating in the cottage.

Keywords/tags (subjects)

IoT, Raspberry Pi, Arduino, DHT11, NRF24L01, Amazon EC2

Miscellaneous

Sisältö

Sanasto 4
1 Johdanto 6
2 Tutkimusmenetelmät 7
3 Tekniset osa-alueet 8
3.1 Ubuntu 8
3.2 Amazon EC2 8
3.3 Raspberry Pi 9
3.4 Thingsboard 10
3.5 DHT-anturit 11
3.5.1 DHT-11 11
3.5.2 DHT-22 12
3.6 Arduino 12
3.7 NRF24L01 13
3.8 MQTT 15
3.8.1 Message Queue Telemetry Transport 15
3.8.2 MQTT QoS 16
3.9 OpenWeatherMap 18
4 Sensoriverkon suunnitelma 19
4.1 Verkon rakenne 19
4.2 Laitteisto 20
4.3 Palvelualusta 22
4.4 Lämpötilan mittaus ja siirto 22
4.5 Paikallisen sääaseman lämpötiladatan lisääminen Thingsboardiin 23
5 Sensoriverkon toteutus 24
5.1 Thingsboard 24
5.1.1 Thingsboard Amazon EC2 alustalle 24
5.1.2 Thingsboardin asennus ja käyttöönotto 24

5.2 Rasperry Pi & DHT-11 27
5.3 Arduinot & DHT-11 31
5.3.1 Mittaus ja lähetys Arduinolla 31
5.3.2 Vastaanotto ja edelleenlähetys Raspberryllä 32
5.3.3 Arduino2 konfigurointi 36
5.4 Sääaseman tietojen liittäminen Thingsboardiin 37
5.5 Käyttöliittymän määritykset 39
5.5.1 Käyttäjien määrittely 39
5.5.2 Sähköpostihälytysten määritys 40
5.6 Internetyhteys 43
6 Pohdinta ja jatkotoimenpiteet 45
Lähteet 46
Liitteet 49
Liite 1. Amazon EC2 Instance types - Amazon Web Services (AWS) verkkosivuston taulukko 49
Liite 2. MQTT viestien selitykset 50
Liite 3. mqtt-dht22.py 50
Liite 4. Raspberrytbupload.py 52
Liite 5. openWeatherMap.py 53
Liite 6. Saadata.py 54
Liite 7. SaunatoMokki.ino 56
Liite 8. Sauna.py 59
Liite 9. OpenWeatherMap:n tarjoama säädata parsimattomana 61

Kuviot

Kuvio 1. Johdantotopologia 7
Kuvio 2. Raspberry Pi 3 Model B 9
Kuvio 3. Thingsboardin toimintatopologia 11
Kuvio 4. DHT-11-anturi 12
Kuvio 5. Arduino Uno R3 13
Kuvio 6. NRF24L01 14
Kuvio 7. MQTT-viestintä 16
Kuvio 8. MQTT QoS-tasojen viestit 17
Kuvio 9. OpenWeatherMapin tarjoamia sopimuksia 18
Kuvio 10. Verkon looginen topologia 20
Kuvio 11. Raspberryyn kytketyt laitteet 21
Kuvio 12. AWS t2.micro, Amazon EC2 Instance types 24
Kuvio 13. Thingsboardin kirjautumissivu 26
Kuvio 14. Thingsboardin pääsivu 27
Kuvio 15. Raspberry Pi 3 GPIO Pins (Alkuperäinen kuvio Dirkk 2017. Openclipart-verkkosivusto.) 28
Kuvio 16. Thingsboard Raspberryn lämpötila- ja kosteusanturin seurantaikkuna 29
Kuvio 17. Raspberrytbupload.py määritys käynnistyessä 30
Kuvio 18. Arduinon tuloste ja Raspberryn vastaanoton tuloste 34
Kuvio 19. Lämpötiladatan kulkeutumisen topologia Arduinolta Thingsboardiin 35
Kuvio 20. Raspberryn ja Arduinon lämpötiladatat allekkain 36
Kuvio 21. Arduino 2, NRF24L01 ja DHT-11 37
Kuvio 22. OpenWeatherMapin tarjoamat sääpalvelut 38
Kuvio 23. Mökin lämpötila ja sääaseman tiedot allekkain vertailukelpoisina 39
Kuvio 24. Kayttajan lämpötilaseuranta 39
Kuvio 25. Sähköpostihälytys-pluginin määritys 40
Kuvio 26. Korkean lämpötilan säännön määritykset 41
Kuvio 27. Sähköpostin generoinnin määritys 42
Kuvio 28. Sähköpostihälytys lähetetään lämpötilan noustessa yli 27 asteen 43
Kuvio 29. Huawei HiLink E353 43

Taulukot

Taulukko 1. Huawei E353 tekniset ominaisuudet (Alkuperäinen kuvio Elisan verkkosivusto, Huawei E353 3G - nettitikku) 44

Sanasto

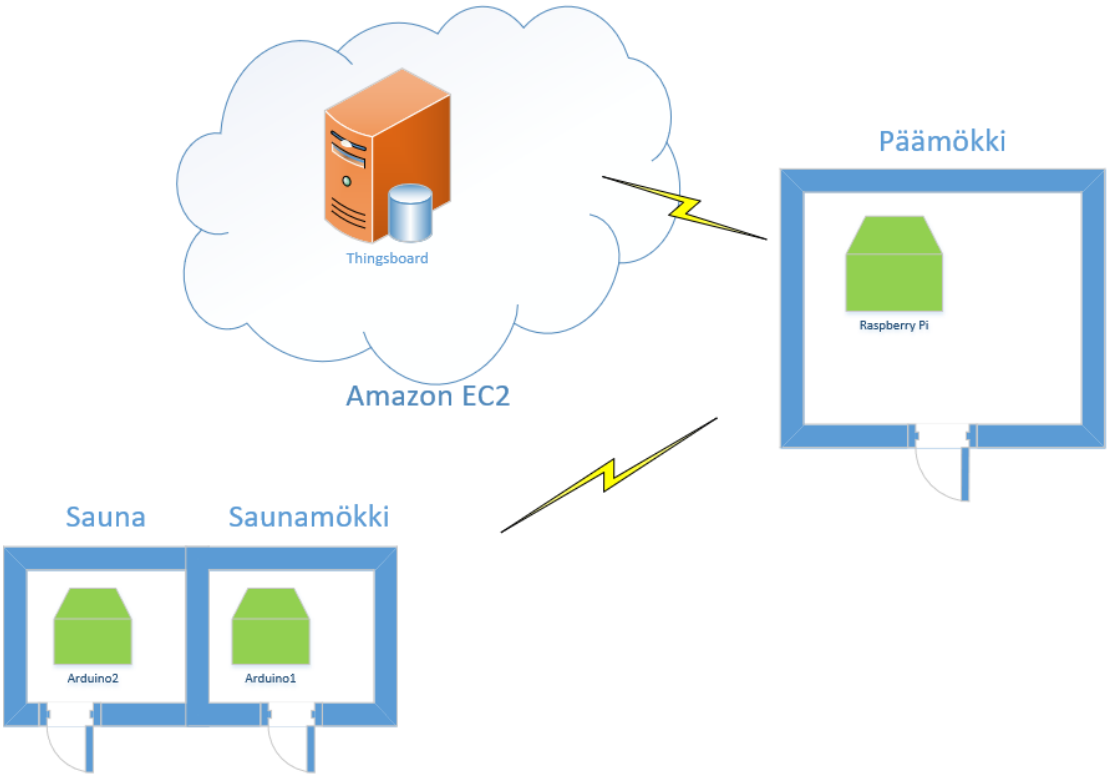
API	Application Programming Interface
AWS	Amazon Web Services
Bluetooth	Langattoman tiedonsiirron standardi
CoAP	Constrained Application Protocol
CPU	Central Processing Unit
EC2	Elastic Compute Cloud
Gb	Gigabitti, 1 000 000 000 bittiä
Ghz	Gigahertsi, 1 000 000 000 hertsiä
HDMI	High-Definition Multimedia Interface
HSQLDB	Hyper SQL Database
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IP	Internet Protocol
JSON	Javascript Object Notation
LAN	Local Area Network
m	Metri
M2M	Machine to Machine
mA	Milliampeeri
Mbps	Megabittiä sekunnissa
MHz	Megahertsi, 1 000 000 hertsiä
MQTT	Message Queuing Telemetry Transport
OWM	OpenWeatherMap
QoS	Quality of Service
RAM	Random-access memory

RHEL	Red Hat Enterprise Linux
SLES	SUSE Linux Enterprise Server
SMTP	Simple Mail Transfer Protocol
SPI	Serial Peripheral Interface
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
µA	Microampeeri
USB	Universal Serial Bus
V	Voltti
Wlan	Wireless Local Area Network
XML	Extensible Markup Language

1 Johdanto

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa langaton sensoriverkko talviasuttavan mökin lämpötilojen seurantaan. Sensoriverkon vaatimuksena oli, että se pystyisi mittaamaan kolmen fyysisen sijainnin lämpötiloja samaan aikaan. Lämpötila-antureiden etäisyys tulisi olemaan maksimissaan 50 metriä. Lisäksi kerättyjä lämpötilatietoja tulisi pystyä seuraamaan reaaliajassa, mutta myös historiatietojen tallentaminen olisi mahdollistettava. Tämä kaikki tieto tulisi olla saatavilla niin mökillä antureiden läheisyydessä kuin etänä mistä tahansa internetin välityksellä.

Tällaisen verkon rakentamiseen oli tarkoitus käyttää Raspberry Pi:tä, Arduinoja, NRF24L01 langattomia lähettimiä, DHT11- ja DHT22-lämpötila-antureita sekä Amazonin EC2 eli Amazon Elastic Compute Cloud-pilvipalvelualustaa. Tarkoituksena oli pystyttää Amazonin EC2 pilvipalvelualustalle Ubuntu Server, johon asennettaisiin Thingsboard. Thingsboard toimii lämpötilojen seurantarajapintana, varastoi lämpötiladatan ja hoitaa tarvittavat sähköpostihälytykset. Päämökkiin oli tarkoitus sijoittaa Raspberry Pi, joka mittaa päämökin lämpötiladataa sekä lähettää ja välittää lämpötiladatan Thingsboardiin. Raspberry Pi:hin liitetään NRF24L01-radiolähetin, sekä DHT-11-lämpötilasensori. Saunamökkiin ja itse sauna/suihkuhuoneeseen tuodaan Arduinot, jotka on varustettu myös NRF24L01-radiolähettimillä ja DHT-11-lämpötilasensoreilla. Etäisyys Rasperryn ja Arduinojen välillä on hieman vajaa 50m. Arduinot välittävät mitatun lämpötiladatan Raspberrylle ja Raspberry edelleen Thingsboardiin. Kuvio 1 mukaisesti voidaan nähdä, miten laitteet on sijoitettu.



Kuvio 1. Johdantotopologia

2 Tutkimusmenetelmät

Opinnäytetyön alkuperäisenä ongelmana selvitettiin, onko markkinoilla jo saatavilla tarvittavaan käyttöön sopivaa ratkaisua, jonka seurauksena lähdettiin pohtimaan, miten tällainen ratkaisu saataisiin itse rakennettua. Lisäksi täytyi ottaa huomioon minkälaisilla laitteistoilla ja konfiguraatioilla sensoriverkko voitaisiin saada toimimaan niin, että valmis ratkaisu palvelisi mahdollisimman hyvin käyttäjiä ja että sen toiminta ja huolto olisivat mahdollisimman yksinkertaisia. Täytyi myös pohtia mahdollisia tulevaisuudessa tehtäviä lisäyksiä sensoriverkkoon ja toteutettavan ympäristön skaalautuvuutta muutoksiin. Opinnäytetyö toteutettiin siis tapaustutkimuksena ja tutkimuskysymyksiä syntyi kolme:

1. Löytyykö markkinoilta jo valmista toimivaa ratkaisua tarvittavaan käyttöön?
2. Onko mahdollista toteuttaa tarvittava ratkaisu itse?
3. Millaisia komponentteja ja laitteita kuten Raspberry Pi, vaaditaan tarvittavan ratkaisun rakentamiseen?

3 Tekniset osa-alueet

3.1 Ubuntu

Ubuntu on avoimeen lähdekoodiin perustuva Linux-käyttöjärjestelmä. Ubuntun historia ulottuu aina vuoden 2004 ensimmäiseen versioon saakka. Ubuntusta on saatavilla niin graafinen kuin komentoriviinkin perustuva versio. Ubuntua päivitetään jatkuvasti ja tämän hetken viimeisin stabiili versio on 16.04.3 LTS (11.02.2018). (Desktop Features.)

Ubuntusta löytyy useita eri versioita erilaisiin käyttötarkoituksiin. Ubuntun perinteinen versio, Ubuntu Desktop on tarkoitettu normaaliin työpöytäkäyttöön ja esimerkiksi vanhaan tietokoneeseen Windowsin korvaajaksi sen keveyden vuoksi. Kyseiseen versioon on asennettu useita valmiita sovelluksia helpottamaan päivittäistä käyttöä. Toisena suosittuna versiona on Ubuntu Server. Tässä versiossa ei ole oletuksena graafista käyttöliittymää, vaan kaikki hallinta tapahtuu komentorivin kautta. Saatavilla on myös esimerkiksi Ubuntu Cloud, joka on tarkoitettu pilvipalveluiden käyttöön, Ubuntu Kylin Kiinan markkinoille sekä useita muita erilaisia versioita eri käyttötarkoituksiin. (Get Ubuntu n.d.; Ubuntu Kylin n.d.)

3.2 Amazon EC2

Amazon EC2 on Amazonin tarjoama pilvipalvelualusta niin yrityksille, kuin yksityishenkilöillekin. EC2 tarjoaa hyvin skaalautuvan alustan palvelimille. Amazonilta on mahdollista vuokrata virtuaalisia palvelimia omien tarpeiden ja käytön mukaan. Palvelun etu on sen helppokäyttöisyys ja skaalautuvuus muuttuviin tarpeisiin. Palvelimien laskentatehoa ja kapasiteettiä on mahdollista lisätä tai vähentää muutamassa minuutissa. (Amazon EC2 n.d.)

EC2 tarjoaa useita eri alustoja käyttäjälle, kuten Ubuntu, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES) ja Windows. Palvelusta on tarjolla vuoden ilmainen kokeilujakso ”Amazon Web Services Free Tier” (AWS) käyttäjälle tietyillä rajoituksilla. Ilmaisen kokeilujakson jälkeen palvelun käyttö jatkuu maksullisena. Hinta

määräytyy käytön ja valitun laskentatehon sekä lisäpalveluiden mukaan. Lisäpalveluina on mahdollista ostaa esimerkiksi Elastic IP, joka vastaa kiinteää julkista IP-osoitetta. Elastic IP hankitaan oman tilin käyttöön, jonka jälkeen sen voi sitoa mille tahansa virtuaaliselle tietokoneelle. (Amazon Elastic Compute Cloud 2018.)

3.3 Raspberry Pi

Raspberry Pi on Raspberry Pi Foundation-yrityksen suunnittelema yhden piirilevyn tietokone, joka kehitettiin käytettäväksi tietotekniikan opetuksessa ja erilaisissa pienissä projekteissa. Raspberrystä on kehitetty useita eri malleja. Tällä hetkellä uusin markkinoilla oleva malli on Raspberry Pi 3 Model B (ks. Kuvio 2). Kyseisestä mallista löytyy Broadcomin valmistama 1,2 gigahertsin (Ghz) taajuudella toimiva 64-bittinen 4-ydinsuoritin, 1 gigabitti (Gb) Random Access Memory –muistia (RAM), Wireless Local Area Network (Wlan) ja Bluetooth 4.1. Tietokone voidaan yhdistää High Definition Multimedia Interfacen (HDMI) avulla televisioon tai tietokoneen monitoriin. Virtansa tietokone saa Micro-Universal Serial Busin (USB) kautta esimerkiksi puhelimen laturin kautta. (Raspberry Pi 3 Model B Specifications n.d.)

Kuvio 2. Raspberry Pi 3 Model B

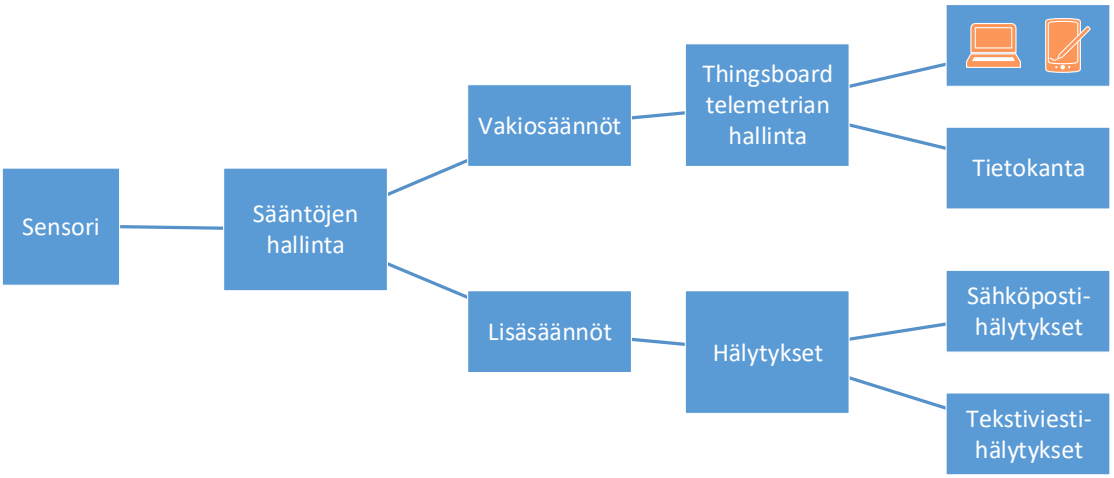
Raspberrylle on saatavilla useita eri käyttöjärjestelmiä. Yleisimmin käytetty on Debianiin perustuva Raspbian, joka on suunniteltu juuri Raspberry Pi:n eri versioille. Raspbian on ilmainen ja sitä päivitetään aktiivisesti. (Raspbian Documentation n.d.)

3.4 Thingsboard

Thingsboard on avoimeen lähdekoodiin perustuva Internet of Things (IoT) alusta. Sen avulla on mahdollista kerätä, säilöä, prosessoida ja visualisoida erilaista dataa. Thingsboardilla voidaan esimerkiksi kerätä lämpötiladataa erilaisilta antureilta, säilöä se ja tehdä säilötystä datasta graafisen käyttöliittymän kautta helposti luettavaa. Käyttöliittymän kautta voi myös ohjata erilaisia laitteita. (Thingsboard – Open-source IoT Platform)

Thingsboard on saatavilla useille eri alustoille. Se voidaan asentaa Windowsin, Ubuntun, CentOS:n tai Raspbianin päälle. Thingsboardilta löytyy myös tuki suoraan Dockeriin ja AWS EC2:lle asennukselle. Thingsboard käyttää datan siirtämiseen MQTT-, CoAP- tai HTTP-protokollaa. (Installation options n.d.)

Thingsboardin avulla on mahdollista rakentaa erilaisia ratkaisuja datan hallintaan ja seurantaan. Sensorista kerätty data siirretään asetettujen määritysten mukaisesti tietokantaan. Dataa on mahdollista seurata graafisen käyttöliittymän kautta selaimen avulla niin reaaliajassa kuin tietokannasta historiaa hakien. Järjestelmään on mahdollista määrittää lisäksi sääntöjä, jotka kerätyn datan täyttäessä lähettävät hälytyksen esimerkiksi sähköpostilla tai tekevät jonkin toimenpiteen järjestelmään (ks. Kuvio 3). (Working with telemetry data n.d.)



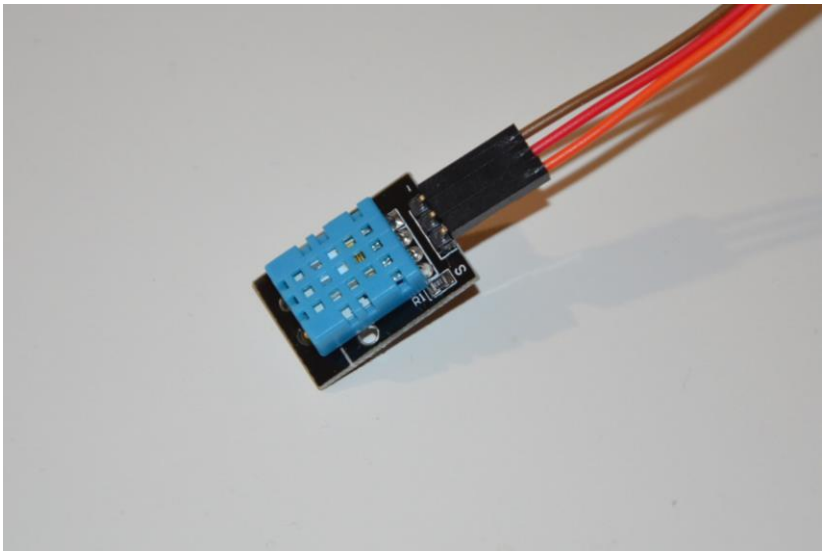
Kuvio 3. Thingsboardin toimintatopologia

3.5 DHT-anturit

DHT-anturit ovat yksinkertaisia ja edullisia lämpötila- ja kosteusantureita. Niitä ei ole tarkoitettu tarkkoihin vaativiin mittaukseen, sillä ne eivät ole kovin nopeita ja tarkkoja, mutta soveltuvat moneen yksinkertaiseen mittaukseen. Sensoreissa on siru, joka muuntaa analogisen signaalin digitaaliseksi, jolloin dataa on helppo lukea miltei millä vain yhteensopivalla laitteella. DHT-11- ja DHT-22-antureiden välillä ei ole fyysisesti suurta eroa. DHT-22:sta voisi sanoa DHT-11:n isoveljeksi. Se on tarkempi ja reagoi nopeammin muutoksiin. Lisäksi sen mittaama lämpötilahaitari on suurempi. Toiminnallisesti anturit ovat samanlaisia. (Lady, A 2018.)

3.5.1 DHT-11

DHT-11 on halvempi vaihtoehto DHT-sarjan antureista (ks. Kuvio 4). 04.02.2018 anturin sai tilattua halvimmillaan noin 0,8€ hintaan Ebaysta. Anturi toimii joko 3 tai 5 voltin (V) jännitteellä. DHT-11-ominaisuutta kosteuden mittaukseen markkinoidaan hyväksi 20-80:n % arvoilla 5:n % tarkkuudella. Lämpötilaa se pystyy mittaamaan ±2°C tarkkuudella 0-50°C välillä. (Aosong n.d; DHT-11 Digital Temperature and Humidity Sensor Temperature sensor Arduino 2018.)



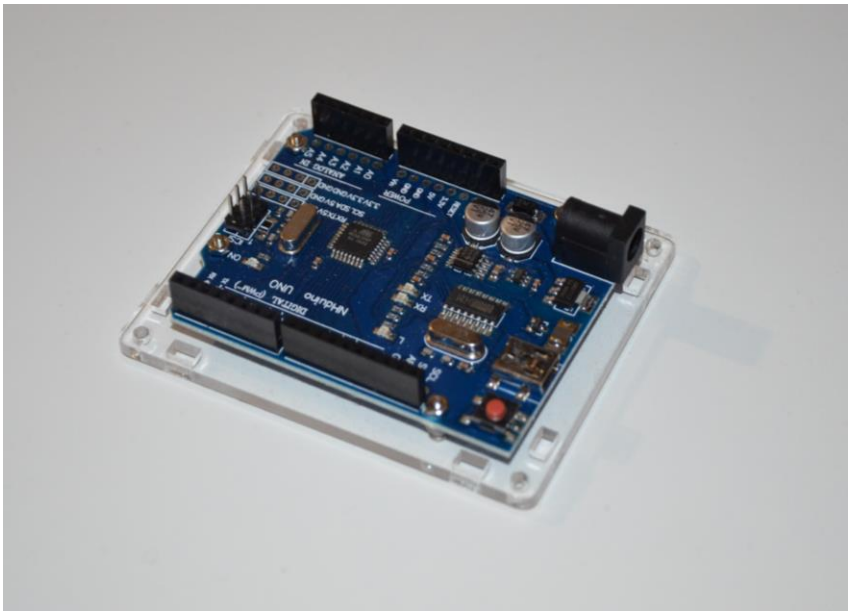
Kuvio 4. DHT-11-anturi

3.5.2 DHT-22

DHT-22 on hieman kalliimpi DHT-sarjan antureista. Sen hinnat alkavat noin 2,2 €:sta Ebayssa (04.02.2018). Myös DHT-22 toimii niin 3:n kuin 5:n voltin jännitteillä. Kosteuden mittauksen luvataan toimivan 2-5:n % tarkkuudella. Mittaus onnistuu kosteuden ollessa 0-100%. Lämpötilaa taas voidaan mitata ±0.5°C tarkkuudella -40 ja 80°C välillä. (Liu n.d.)

3.6 Arduino

Arduino on avoimeen lähdekoodiin perustuva alusta ja yhden piirilevyn mikrokontrolleri (ks. Kuvio 5). Arduinon historia alkaa Interaction Design Instituutista Ivreasta, Italiasta. Se kehitettiin alun perin opiskeluun, mutta laajemman tietoisuuden saavuttaessaan sitä on käytetty tuhansissa ja taas tuhansissa erilaisissa projekteissa. Siitä löytyy useita erilaisia hieman toisistaan poikkeavia malleja useilta eri valmistajilta. Pääosin mallit ovat samanlaisia liitännöiltään ja toiminnallisuudeltaan. Esimerkiksi USB-portti saattaa vaihdella mallin mukaan (What is Arduino? n.d).



Kuvio 5. Arduino Uno R3

Arduino Uno Rev3-versiota kaupataan Arduinon omilla verkkosivuilla 20.00 €:n hintaan ilman veroja. Kyseisestä mallista löytyy silti myös muiden valmistajien halvempia versioita. Esimerkiksi NHduino R3, joka on identtinen alkuperäiseen malliin, maksaa hieman yli 3 euroa. Arduinon käyttöjännite on 5 V, mutta syötetty jännite voi olla 7-12 V:n välillä. Kyseisessä mallissa on 14 digitaalista I/O-pinniä. Mikrokontrolleri on ATmega328P 32 KB flash-muistilla sekä 16 megahertsin (MHz) kellotaajuudella varustettuna. (Arduino Uno Rev3 n.d.; NEW UNO R3 ATmega328P CH340 Mini USB Board for Compatible-Arduino 2018.)

Arduinoon on mahdollista ohjelmoida erilaisia luku- ja tulostustoimintoja. Helpoiten näiden hallinnointi tapahtuu Arduino Desktop IDE-käyttöliittymän kautta, joka on saatavilla Windowsille, Mac OS X:lle ja Linuxille. Arduino IDE tukee C ja C++ ohjelmointikieliä. Käyttöliittymän kautta on mahdollista siirtää kirjoitettu ohjelma Arduinolle ja lukea Arduinon tulostetta. (SM 2017.)

3.7 NRF24L01

NRF24 on 2,4 Ghz:n taajuudella toimiva radiolähetin (ks. Kuvio 6). Tarkemmin radiolähettimen taajuus on 2.400 GHz - 2.525 GHz. Se soveltuu erityisen hyvin projekteihin

ja opiskeluun sen alhaisen hankintahinnan vuoksi. Lähetin on antennin kanssa saatavilla alle 3 $:n hintaan. NRF24L01-tuoteinfossa kerrotaan lähettimien soveltuvan yleiseen IoT käyttöön, kuten kodin automaatioon, hälytysjärjestelmään, tuotantoympäristöön ja vaikka leluksi. Lähettimen teoreettinen maksimi tiedonsiirtonopeus on 2 megabittiä sekunnissa (Mbps) langattomasti ja 8 Mbps Serial Peripheral Intefacen (SPI) kautta. Kanavia lähettimen käytössä on yhteensä 126. Ne jakautuvat 1 Mhz:n tarkkuudella, kun käytetään 1 Mbps tiedonsiirtonopeutta. Mikäli käytössä on 2 Mbps, kanavat jakautuvat 2 Mhz:n välein. (Single chip 2.4 GHz Transceiver 2006.)

NRF24 lähetin sisältää automaattisen virran säästön ja virran katkaisun, mikä tekee siitä oivallisen käytettävän esimerkiksi pattereiden avulla. Lähettimelle löytyy internetistä useita erilaisia mallikonfiguraatioita ja kirjastoja niin Arduinon kuin Raspberry Pi:n käyttöön. Erillisen antennin avulla saavutetaan jopa 100 metrin (m) lähetyskantama. (Single chip 2.4 GHz Transceiver 2006.)

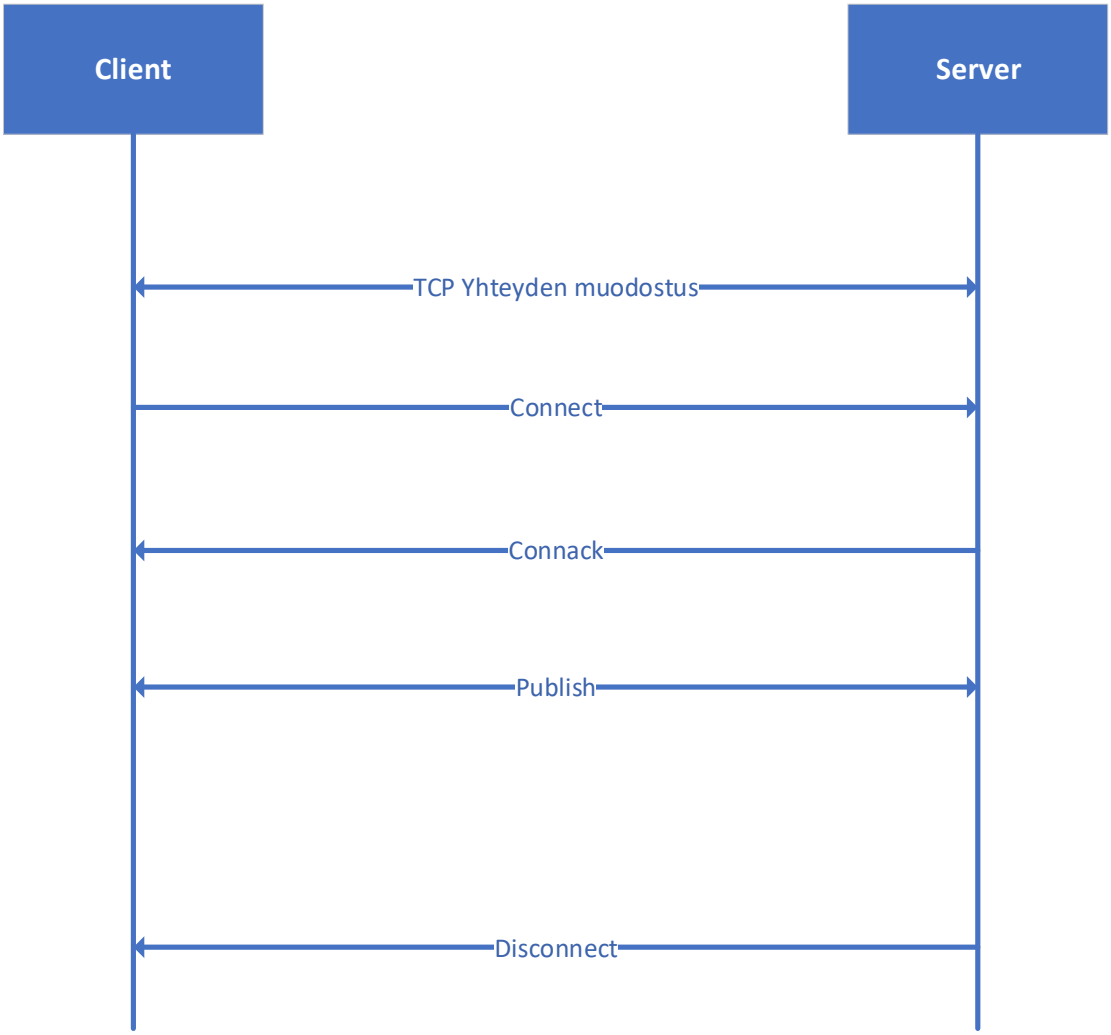
Kuvio 6. NRF24L01

3.8 MQTT

3.8.1 Message Queue Telemetry Transport

MQTT kehitettiin kevyeksi viestintäprotokollaksi. Sitä käytetään varsinkin Machine-to-Machine (M2M) tiedonsiirrossa sekä IoT-tyylisissä ratkaisuissa. MQTT mahdollistaa IoT-laitteiden lähettää tai jakaa niiden keräämää tai tuottamaa dataa. Data lähetetään palvelimelle, joka toimii Message Brokerina. Brokerina toimiva laite jakaa tiedon sen ennalta tilanneille kohteille. MQTT toimii Transmission Control Protocol/Internet Protocol (TCP/IP) protokollan päällä. Perinteisessä tapauksessa se käyttää 1883 porttia. Mikäli yhteys halutaan salata, käytetään Secure Sockets Layer/Transport Layer Security (SSL/TLS) ja 8883 porttia. (Rouse. 2018.)

MQTT-viestintä on yksinkertaisimmillaan vain muutaman viestin mittaista (ks. Kuvio 7). Aluksi muodostetaan TCP-yhteys. Kun yhteys on muodostettu, ”Asiakaslaite” pyytää yhteyden muodostusta CONNECT-viestillä. Palvelin hyväksyy oheisen pyynnön CONNACK-viestillä, jonka jälkeen PUBLISH-viestejä voidaan tämän jälkeen lähettää molempiin suuntiin. Näitä viestejä voidaan lähettää aina yhteyden katkaisemiseen asti. Viestinnässä voi olla muunkin tyylisiä viestejä, kuten Quality of Service (QoS) tasojen mukaan määräytyviä viestejä. Nämä eivät kuitenkaan ole pakollisia. (Piyush. 2017.)

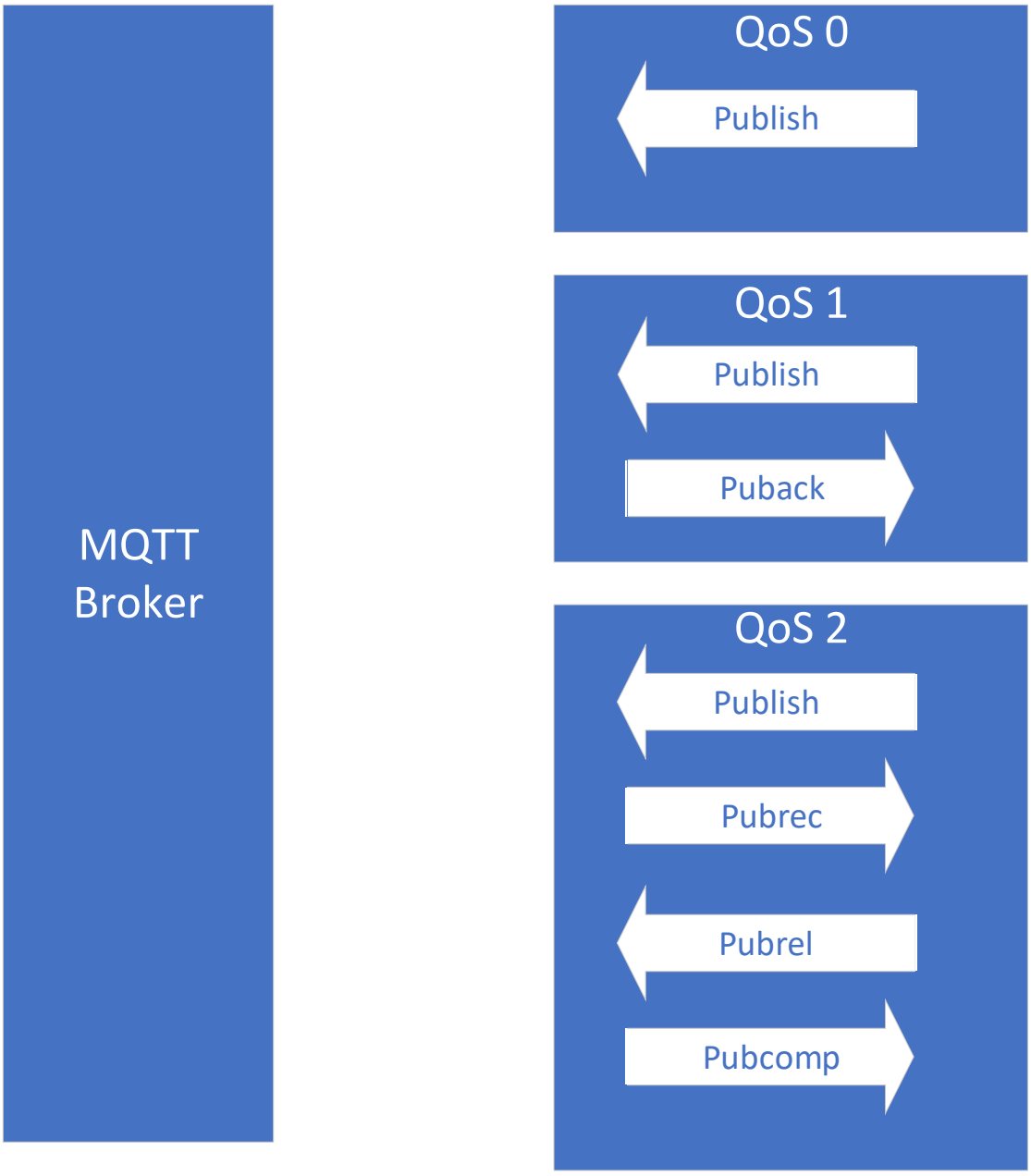


Kuvio 7. MQTT-viestintä

3.8.2 MQTT QoS

MQTT protokolla on hyvä valinta varsinkin langattomille verkoille. Näiden verkkojen viive ja kapasiteetti voivat vaihdella hyvinkin paljon riippuen välimatkasta ja muista olosuhteista. MQTT protokolla tukee kolmea eri QoS-tasoa. Tasot ovat nimetty yksinkertaisesti QoS 0, QoS 1 ja QoS 2. Yksinkertaisin taso, QoS 0 toimii "ammu ja unohda" tyylillä. Viesti lähetetään kerran ja poistetaan sen jälkeen. Tätä tasoa käytettäessä ei ole varmuutta, kulkeutuiko viesti ikinä perille. QoS 1-tasolla varmistetaan viestin menneen perille, jonka jälkeen viesti vasta poistetaan. Mikäli kuittausta viestistä ei siis saada tietyn aikajakson sisällä, sama viesti lähetetään uudelleen. Tämä voi aiheuttaa sen, että sama viesti saapuu MQTT Brokerille useamman kerran. QoS 2-tasolla lä-

hetys on kaksivaiheinen. Ensin lähetetään PUBLISH/PUBREC-viestit, joiden jälkeen lähetetään PUBREL/PUBCOMP-viestit. Tällä varmistetaan, että sama viesti ei saavu perille kahteen kertaan (Ks. Kuvio 8). MQTT viestien kuvaukset on selvennetty Liitteessä 2. (Rouse M. 2018.)



Kuvio 8. MQTT QoS-tasojen viestit

3.9 OpenWeatherMap

OpenWeatherMap (OWM) on erilaisia säätietoja tarjoava IT-alan yritys. Yritys on perustettu vuonna 2012, tarkoituksena tuoda kaikkien saataville ja käyttöön säätietoja. OWM tarjoaa säätietoja yli 200 000:n kaupunkiin ja sijaintiin. Dataa on saatavilla yli 40 000:lta sääasemalta ympäri maailman. Lisäksi saatavilla on satelliittidataa, jota voidaan hyväksikäyttää esimerkiksi laajojen tulipalojen havaitsemiseen.

OWM tarjoaa erilaisia Application Programming Interface (API)-rajapintoja datan käyttöön. Ilmaiseksi on tarjolla reaaliaikaiset säätiedot, sekä 5 päivän ajalta 3 tunnin tarkkuudella sääennusteet. OWN tarjoaa säätiedot Javascript Object Notation (JSON), Extensible Markup Language (XML) ja Hypertect Markup Language (HTML) formaateissa. Eri hintaisissa maksullisissa palveluissa on saatavilla tarkempaa säädataa, historiadataa säästä, ilmanlaadun muutokset ja paljon muuta säähän liittyvää. OWM tarjoaa monta erilaista ja eri hintaista sopimusta käyttäjilleen (ks. Kuvio 9). (About Company n.d; Weather API n.d)

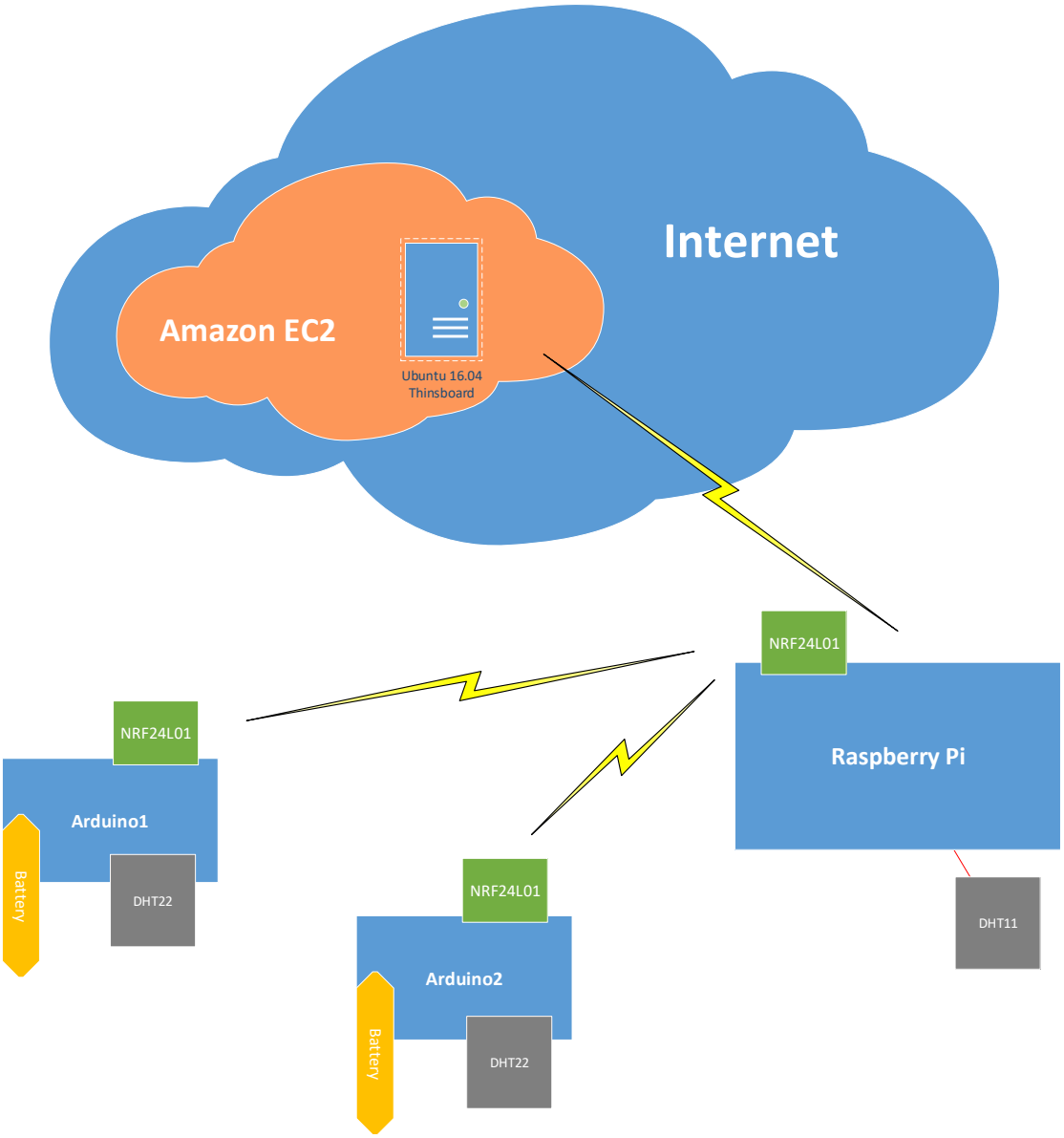
	Free	Startup	Developer	Professional	Enterprise
Price Price is fixed, no other hidden costs.	**Free**	**40 USD** / month	**180 USD** / month	**470 USD** / month	**2,000 USD** / month
Subscribe	Get API key and Start	Subscribe	Subscribe	Subscribe	Subscribe
Calls per minute (no more than)	60	600	3,000	30,000	200,000
Current weather API	✓	✓	✓	✓	✓
5 days/3 hour forecast API	✓	✓	✓	✓	✓
16 days/daily forecast API	-	✓	✓	✓	✓
Weather maps API	✓	✓	✓	✓	✓
Bulk download	-	-	-	✓	✓
UV index (beta)	✓	✓	✓	✓	✓
Air pollution (beta)	✓	✓	✓	✓	✓
Weather alerts (beta)	✓	✓	✓	✓	✓

Kuvio 9. OpenWeatherMapin tarjoamia sopimuksia

4 Sensoriverkon suunnitelma

4.1 Verkon rakenne

Toteutettavan verkon topologia oli melko yksinkertainen. Koska välimatkat antureiden välillä olivat melko pitkiä, yhteydet toteutetaan langattomasti. Raspberry Pi tulee toimimaan yhdyskäytävänä Arduinoihin liitettävien antureiden ja Amazonin pilvessä sijaitsevan Thingsboardin välillä (Ks. Kuvio 10).

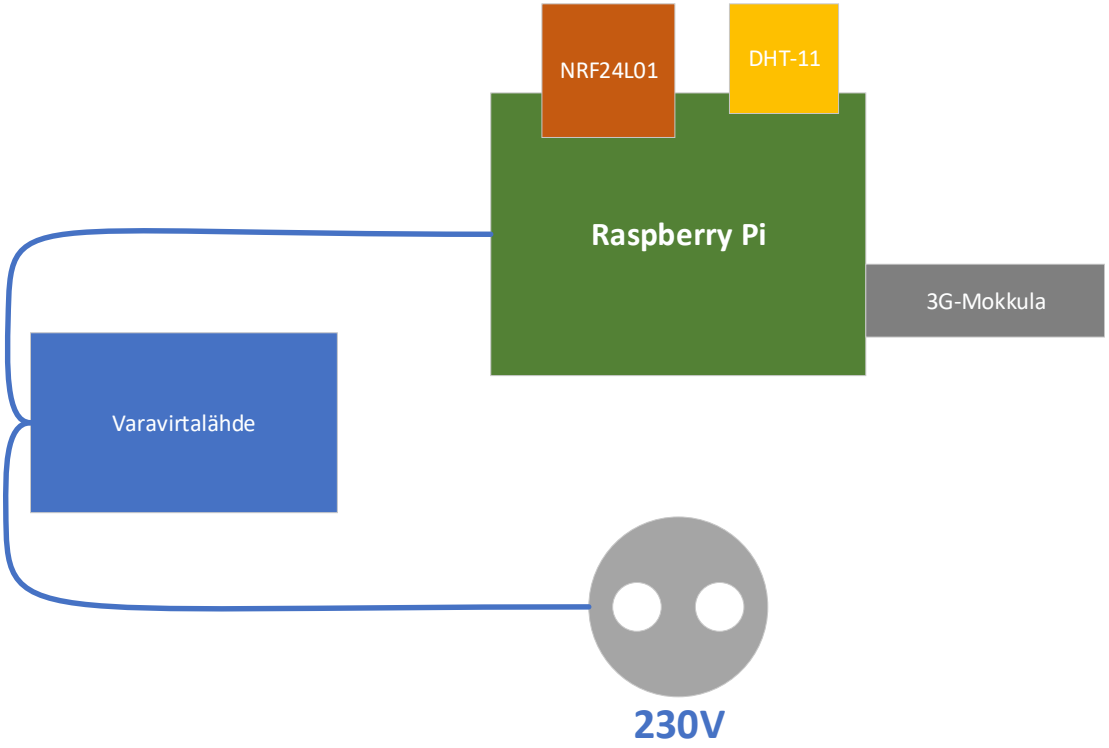


Kuvio 10. Verkon looginen topologia

4.2 Laitteisto

Opinnäytetyön suorittamiseen käytettiin useita erilaisia laitteita. Palvelin, jossa Thingsboard pyöri, on Amazonin pilvipalvelun päällä. Fyysisesti se siis sijaitsee jossakin Amazonin konesaleista.

Raspberry Pi:tä käytetään yhdyskäytävänä internetin välityksellä Amazonin palvelimelle. Se toimii myös itse yhden anturin datan kerääjänä. Raspberryyn liitetään DHT-11-lämpötila- ja kosteusanturi. Virtansa Raspberry Pi saa varavirtalähteestä, joka taas on kytketty puhelimen usb-laturilla verkkovirtaan (Ks. Kuvio 11). Näin sähkökatkon sattuessa Raspberry ei mene virroitta. Internetyhteys hoidetaan Raspberryyn liitettävällä 3G-mokkulalla. Lämpötilasensorien dataa vastaanottamaan liitetään myös NRF24L01 radiolähetin.



Kuvio 11. Raspberryyn kytketyt laitteet

Kaksi DHT-11-lämpötila- ja kosteusanturia liitetään kahteen erilliseen Arduinoon. Arduinoihin on myös liitetty NRF24L01-radiolähetin, jonka avulla ne lähettävät lämpötiladatansa Raspberrylle. Virtansa Arduinot saavat pattereista.

4.3 Palvelualusta

Alustana lämpötilojen seurannalle käytetään Thingsboardia. Se asennetaan Ubuntu Server 16.04 LTS:lle, joka pyörii Amazonin EC2 pilvipalvelualustalla. Tällöin Thingsboard on saatavilla helposti kaikkialta Internetin välityksellä ja lämpötilojen reaaliaikaista- ja historiadataa voidaan seurata millä tahansa laitteella. Palvelimelle määritetään kiinteä IP-osoite, joka Amazonin pilvipalvelussa kulkee nimellä Elastic IP.

Thingsboard räätälöidään käyttäjien tarpeiden mukaan mahdollisimman helppokäyttöiseksi. Alustalle määritetään jokaiselle käyttäjälle omat tunnukset, joka helpottaa entisestään määritysten räätälöintiä henkilökohtaisten mieltymysten mukaisiksi. Esimerkiksi toinen käyttäjä voi haluta tarkistaa vain nykyisen lämpötilan, kun taas toinen haluaa helposti nähdä myös historiatiedot viimeiseltä kuukaudelta.

4.4 Lämpötilan mittaus ja siirto

Lämpötilojen mittaus tapahtuu DHT-11- ja DHT-22 lämpötila- ja kosteusantureiden avulla. Kyseisiä antureita kytketään yksi suoraan Raspberry Pi:hin ja yksi kumpaankin Arduinoon. Anturit säädetään tekemään lämpötilamittaus kerran 30 minuutissa. Näin harvalla mittauksella on tarkoitus säästää Arduinojen kuluttamaa virtaa, sillä ne toimivat pattereiden varassa.

Kahden anturin datat lähetetään langattomasti Arduinoihin liitettyjen NRF24L01 radiolähettimien kautta Raspberry Pi:lle. Tämä toimii siis yhdyskäytävänä keräten kahden anturin lämpötiladatan, sekä itseensä liitetyn anturin datan ja välittää ne Amazonin pilveen Thingsboardille. Siirtoon käytetään MQTT-protokollaa. Protokolla käyttää 1883 porttia vakiona ja 8883 porttia salatun yhteyden kanssa.

Dataa tullaan keräämään jatkossa pidemmältä aikaväliltä. Tämän datan avulla voidaan tehdä arvioita ulkolämpötilan vaikutuksesta sisälämpötilaan. Kerättyjen lämpötilatietojen ja lämpötilamuutoksien perusteella tullaan tekemään data-analytiikan avulla ennakoivaa lämpötilojen säätöä, jotta hukkalämpö ja toisaalta taas putkistojen jäätymisen mahdollisuus saadaan minimoitua.

4.5 Paikallisen sääaseman lämpötiladatan lisääminen Thingsboardiin

Thingsboardiin lisätään paikallisen sääaseman lämpötiladata, jonka on tarkoitus olla vertailukelpoista dataa sisälämpötilan kanssa. Datan avulla voidaan seurata, kuinka paljon lämpötilanmuutokset ulkona vaikuttavat sisälämpötilan muutoksiin ja näin voidaan ennakoida esimerkiksi lattialämmityksen säätöä kovilla pakkasilla. Sääaseman keräämä lämpötiladata sijaitsee vain noin kahden kilometrin päässä mökin sijainnista, joten sen vertailukelpoisuus on melko hyvä. Internetissä on saatavilla OpenWeatherMap:n jakamaa säädataa ilmaiseksi, joten sitä käytetään projektissa.

5 Sensoriverkon toteutus

5.1 Thingsboard

5.1.1 Thingsboard Amazon EC2 alustalle

Thingsboardin konfigurointi aloitettiin luomalla tili Amazonin EC2 pilvipalvelualustalle. Tilin luomisen yhteydessä Amazon tarjoaa vuoden ilmaisen kokeilujakson uusille käyttäjille. Ilmainen kokeilujakso sisältää joitakin rajoitteita, mutta riittää täysin Thingsboardin pyörittämiseen. Amazonin EC2 palvelussa puhutaan virtuaalipalvelimista instansseina. Kyseisen käyttöönotetun instanssin tyypiksi valittiin t2.micro. Instanssin käytössä on 1 virtuaalinen Central Processing Unit (CPU), 1Gb RAM-muistia (Ks. Kuvio 12). T2.micro on tarkoitettu kevyiden palveluiden käyttöön.

Instance Type	vCPU	Memory (GiB)	Storage (GB)	Networking Performance	Physical Processor	Clock Speed (GHz)	Intel AVX†	Intel AVX2†	Intel Turbo	EBS OPT	Enhanced Networking†
t2.micro	1	1	EBS Only	Low to Moderate	Intel Xeon family	Up to 3.3	Yes	-	Yes	-	-

Kuvio 12. AWS t2.micro, Amazon EC2 Instance types

Palvelimen sijaintia ei voi valita kokeilujakson aikana. Thingsboardin käyttöön pystytetty palvelin sijaitsi Yhdysvalloissa, tarkemmin Ohiossa. Palvelimen käyttöjärjestelmäksi valittiin Ubuntu 16.04. Amazon on tehnyt pilvipalvelimen käyttöönoton mahdollisimman helpoksi. Käyttöjärjestelmän valinnan jälkeen palvelin asensi käyttöjärjestelmää muutaman minuutin, jonka jälkeen siihen pystyi ottamaan etäyhteyden SSH:n avulla.

5.1.2 Thingsboardin asennus ja käyttöönotto

Thingsboard käyttää alustanaan Javan kahdeksatta versiota. Myös yhdeksäs versio on jossakin määrin tuettu, mutta sen bugeista ja tietoturva-aukoista varoiteltiin, joten

käyttöön valittiin versio 8. Tämän jälkeen alkoi itse Thingsboardin asennus. Tietokantana toimii vakiona Hyper SQL DataBase (HSQLDB). Tämän vaihtoon ei nähty syytä. Thingsboardin sivuilta löytyi hyödylliset ohjeet asennukseen ja konfiguraatioiden määritykseen. Githubista oli saatavilla valmis paketti Thinsboardista. Kyseinen paketti ladattiin palvelimelle alla olevalla komennolla. (Installing ThingsBoard on Linux n.d.)

wget https://github.com/thingsboard/thingsboard/releases/download/v1.4/thingsboard-1.4.deb

Ladattu asennuspaketti suoritettiin alla olevalla komennolla.

sudo dpkg -i thingsboard-1.4.deb

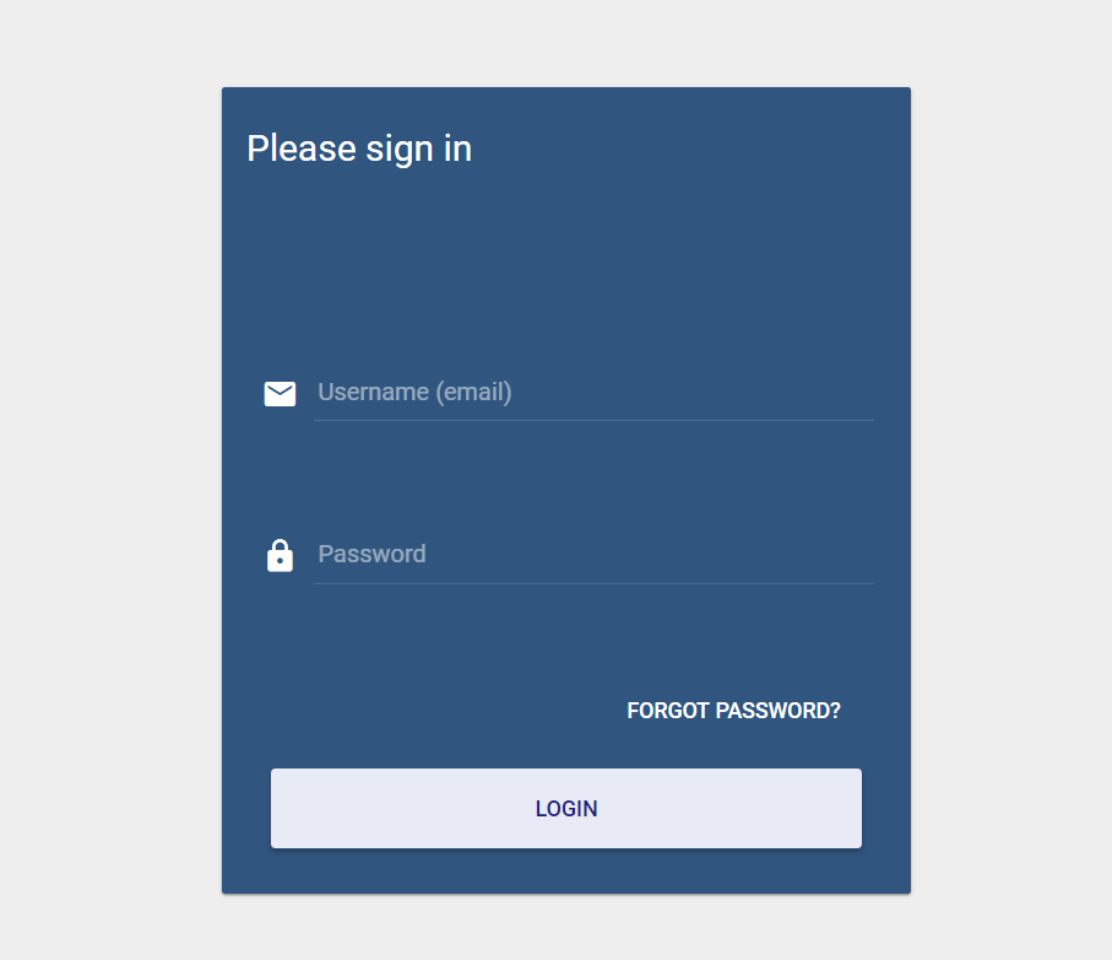
Koska Thingsboard asennettiin palvelimelle, jossa on vain 1GB RAM-muistia, rajattiin Thingsboardin käyttämän muistin määrää 256 megatavuun */etc/thingsboard/conf/thingsboard.conf* konfiguraatiotiedostossa alla olevalla konfiguraatiomuutoksella.

export JAVA_OPTS="$JAVA_OPTS -Xms256M -Xmx256M"

Oheisten komentojen jälkeen Thingsboard oli asennettu. Alla olevalla komennolla ladattiin niin sanottu demodata palvelun käyttöön. Tämän komennon jälkeen valmiiksi konfiguroidut käyttäjätunnukset toimivat palvelussa. Ilman komennon suorittamista palveluun ei siis pääse kirjautumaan millään tunnuksilla.

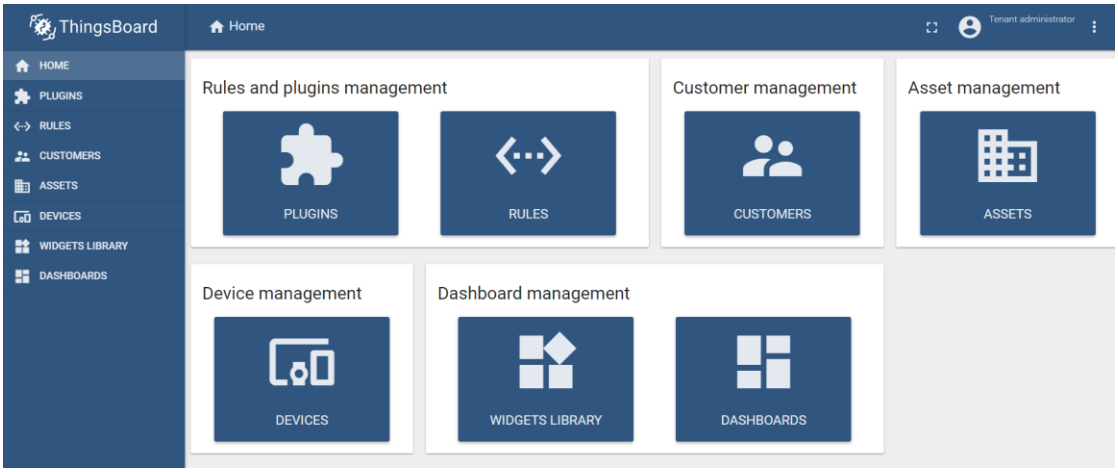
sudo /usr/share/thingsboard/bin/install/install.sh --loadDemo

Muutama minuutti Thingsboardin palvelun käynnistämisen jälkeen Thingsboardin webbikäyttöliittymään pääsi jo käsiksi selaimen kautta Amazonin palvelusta löytyvän palvelimelle asetetun IP-osoitteen avulla. Thingsboardin kirjautumissivun näkymä on hyvin yksinkertainen (Ks. Kuvio 13). Toimivat tunnukset löytyivät pienen etsimisen jälkeen Thingsboardin ohjeista.



Kuvio 13. Thingsboardin kirjautumissivu

Kirjautumisen jälkeen näkymä vaikuttaa todella selkeältä (Ks. Kuvio 14). Palvelusta löytyy valmiiksi konfiguroituja esimerkki widgettejä sekä seurantaikkunoita. Niiden muokkaus, tai kokonaan uusien luonti oli myös suhteellisen yksinkertaista. Things-boardin verkkosivuilta löytyi joitakin ohjeita muutosten tekemiseen.

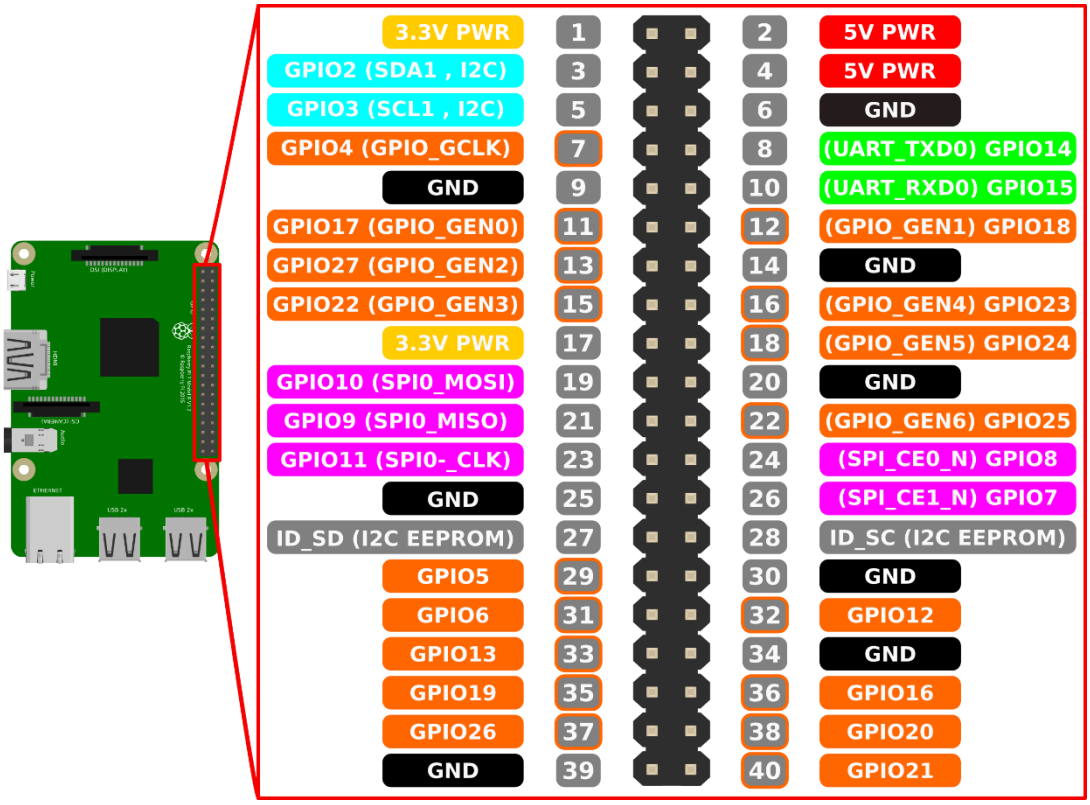


Kuvio 14. Thingsboardin pääsivu

5.2 Rasperry Pi & DHT-11

Suunnitelman mukaisesti Raspberry Pi:tä käytettiin yhdyskäytävänä Arduinojen ja Internetin, sekä Amazonin palvelimen välissä. Lisäksi Raspberryyn liitettiin yksi lämpötila-anturi. Käyttöjärjestelmäksi Raspberrylle valittiin Raspbian sen hyvän toimivuuden ja yhteensopivuuden vuoksi. Raspberryn sivuilta ladattiin viimeisin versio Raspbianin CLI-pohjaisesta käyttöjärjestelmästä, joka toteutushetkellä oli Raspbian Stretch Liten versio 4.9. Käyttöjärjestelmä siirrettiin 32Gb microsd-kortille ja liitettiin Raspberryyn. Raspbianin annettiin tehdä alkuasennukset, jonka jälkeen määriteltiin käyttäjätunnukset käyttöjärjestelmään, näppäimistöasetukset, verkkoyhteydet ja muut määritykset. Asennus- ja konfigurointihetkellä Raspberry liitettiin langattomaan lähiverkkoon.

Raspberryn oltua normaalisti toiminnassa, liitettiin siihen DHT-11 lämpötila-anturi. Anturi liitettiin Raspberryn GPIO pinneihin. Virta anturille saatiin pinnistä 17, maa pinnistä 9 ja data siirretään GPIO4 pinnistä numero 7 (Ks. Kuvio 15).



Kuvio 15. Raspberry Pi 3 GPIO Pins (Alkuperäinen kuvio Dirkk 2017. Openclipart-verkkosivusto.)

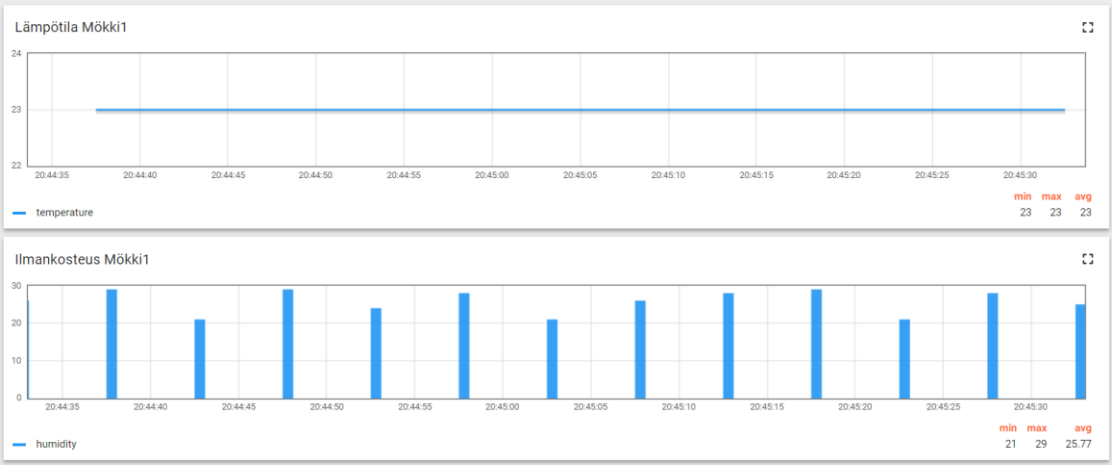
Jotta Thingsboard voisi tunnistaa juuri tietyn anturin datan, täytyi Thingsboardiin määritellä laitteelle oma Access Token. Laite nimettiin ja Access Token generoitiin. Jokaisella anturilla tai laitteella täytyy olla oma Access Token. Laitteiden data merkataan ja ohjataan oikeaan paikkaan sen avulla. Thingsboard siis tunnistaa tietyn datan Access Tokenin avulla. Raspberrylle asennettiin MQTT Python-kirjasto, sekä Adafruitin DHT-kirjasto alla olevien komentojen mukaan.

sudo apt-get install python-dev
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install

Thingsboardilta oli saatavilla malliskripti ”mqtt-dht22.py”. Kyseisestä skriptipohjasta siistittiin kaikki ylimääräinen pois ja määritettiin arvot vastaamaan laitteiden arvoja (Ks. Liite 3). Skriptiin määriteltiin Amazonin palvelimen IP-osoite, Anturille generoidun Access Tokenin ID, Raspberryn GPIO pinni, johon anturi liitettiin ja aikaväli,

kuinka usein lämpötilaa luetaan ja lähetetään. Skripti ajettiin ja jätettiin pyörimään taustalle.

Thingsboardin käyttöliittymään määritettiin Widgetit lämpötilalle ja ilmankosteudelle, jonka jälkeen niiden arvoja voitiin seurata livenä (Ks. Kuvio 16). Widgetit myös määritettiin laskemaan keskiarvo, sekä näyttämään minimi- ja maksimiarvot.



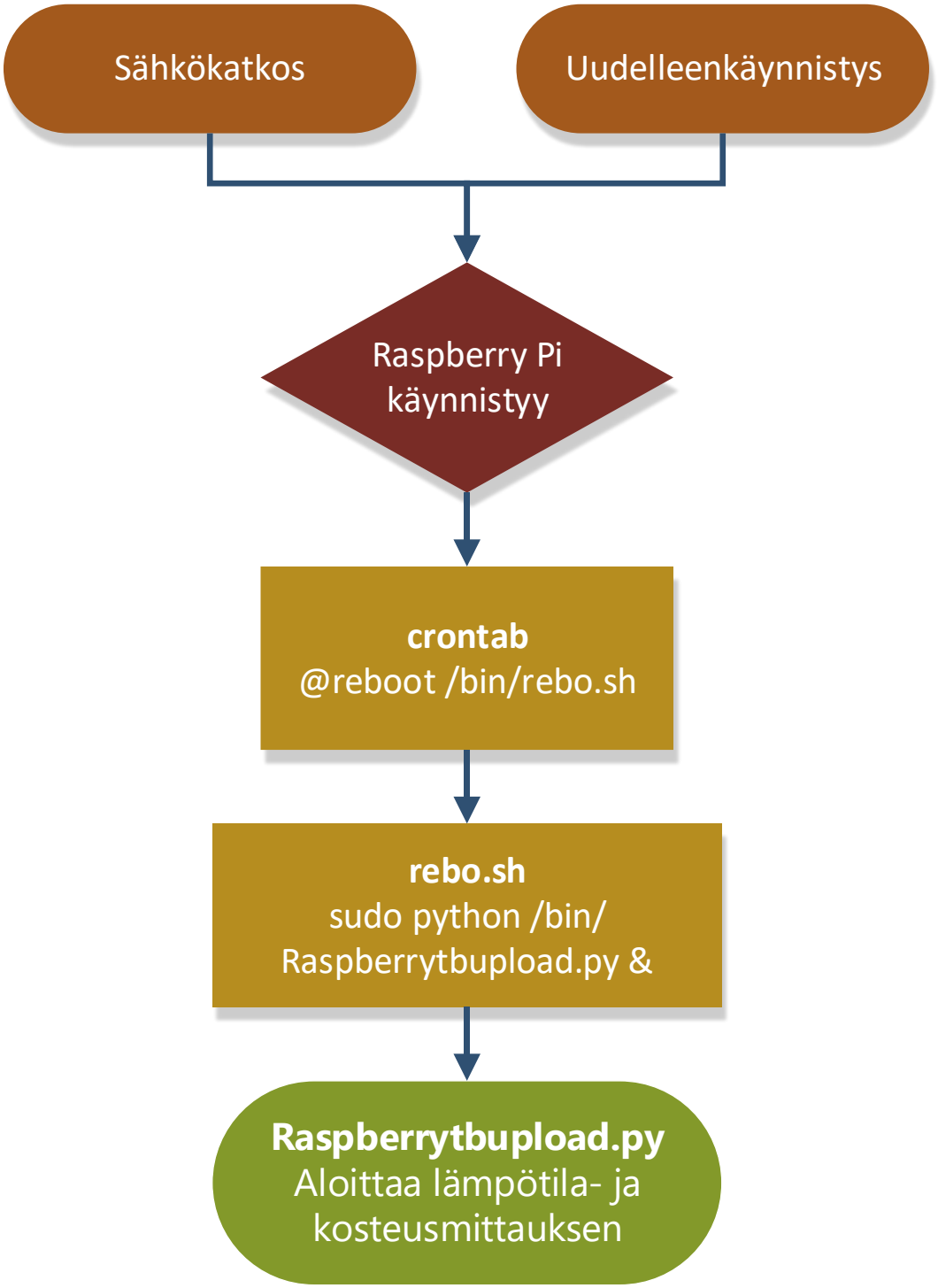
Kuvio 16. Thingsboard Raspberryn lämpötila- ja kosteusanturin seurantaikkuna

Jotta lämpötila- ja kosteusanturit käynnistävä mittaus käynnistyisi automaattisesti Raspberryn käynnistyessä, tehtiin /bin kansioon ”rebo.sh” niminen tiedosto johon määritettiin lämpötilamittauksen käynnistävä komento. Raspberrytbupload.py on muuten sama skripti, kuin *mqtt-dht22.py*, mutta turha lämpötilan ja ilmankosteuden tulostava rivi poistettiin (ks. Liite 3). Skriptin toimintatopologia on määritelty Kuvio 17:ssa.

sudo python /bin/Raspberrytbupload.py &

Rootin crontabbiin määriteltiin rivi, joka suorittaa rebo.sh skriptin käynnistyksen yhteydessä alla olevalla rivillä.

@reboot /bin/rebo.sh



Kuvio 17. Raspberrytbupload.py määritys käynnistyessä

Raspberryn virtalähteeksi liitettiin 10 000 mAh varavirtalähde. Tietyn mittaisen sähkökatkon ei siis pitäisi katkaista virtaa Raspberrystä ja näin nähdään myös, kuinka nopeasti sähköttömyys vaikuttaa sähkölämmitteiseen mökkiin. Virtapankin virran riittävyys testattiin lataamalla virtapankki täyteen ja irroittamalla se sitten verkkovirrasta. Virtapankkiin oli liitetty Raspberry joka mittasi lämpötilaa, sekä välitti sitä eteenpäin

Thingsboardille. Virtaa riitti noin 36 tunniksi, joka on riittävä oheista käyttöä ajatellen.

5.3 Arduinot & DHT-11

5.3.1 Mittaus ja lähetys Arduinolla

Noin 50m päähän Raspberrystä asetettiin Arduino mittaamaan toisen mökin lämpötilaa. Arduinon virtalähteeksi liitettiin 9 V:n paristo. Arduinoon liitettiin ensimmäiseksi DHT-11-lämpötila- ja kosteusanturi. Data luetaan Arduinon pinnistä 2. Virta anturille otetaan 3.3 V:n pinnistä ja maa anturille GND pinnistä.

Arduinoon liitettiin NRF24L01-radiolähetin. Lähetin toimii 3.3 V:n jännitteellä. Anturin muut pinnit liitettiin Arduinon 9-13 pinneihin. Konkreettiset pinnit, joita käytetään ja määritellään käytettävään skriptiin ovat lähettimen CE ja CSN pinnit, jotka liitettiin Arduinon 9 ja 10 pinneihin.

Lämpötilan mittaukseen ja sen lähettämiseen tehtiin skripti (ks. Liite 7), joka kirjoitettiin C++ kielellä. Skriptin muodostamiseen tarvittiin RF24-kirjasto, Adafruitin sensorikirjasto, DHT-antureille kirjasto, sekä SPI-kirjasto. Nämä kaikki asennettiin Arduino IDE-ohjelmaan ensin. Skriptissä määriteltiin lähettimen käytettävät pinnit alla olevan mukaisesti. Näiden pinnien liitoksia olisi voinut muuttaa mieleisekseen ja sopimaan kokoonpanoon, mikäli Arduinoon tulee muitakin laitteita kiinni.

```
// Lahettimen ce ja csn pinnit
RF24 radio(9, 10);
```

Arduinon skriptiin täytyi tehdä langattoman yhteyden luomiseen käytettävät määritykset alla olevan mukaisesti. Nämä määritykset täytyy olla samat niin Arduinolla, kuin Raspberrylläkin. Kanavaksi asetettiin 76, joka vastaa 2476 Mhz taajuutta.

```
// Maaritetaan kanava, jossa lahetetaan. Maaritykset oltava samat, kuin vastaanottavalla Raspberrylla
 radio.setChannel(0x76);
 radio.openWritingPipe(0xF0F0F0F0E1LL);
```

```
radio.enableDynamicPayloads();
```

Jokaisen lämpötilan mittauksen jälkeen lämpötila tulostetaan serial-portin kautta, eli tässä tilanteessa USB-portin kautta. Tämän avulla tulostetta on mahdollista seurata myös liittämällä Arduino tietokoneeseen. Mitattu lämpötila on tallennettu ”lampo”-muuttujaan, joka myöskin lähetetään NRF24L01-radiolähettimen kautta Raspberrylle.

```
// Tulostetaan Serialin kautta mitattu lampotila
   Serial.print("Temperature: ");
   Serial.print(event.temperature);
   Serial.println(" *C");
   const char lampo = (event.temperature);
// Lahetetaan mitattu lampo radiolahettimen kautta. Arvo tallennettu lampo-muuttujaan.
   radio.write(&lampo, sizeof(lampo));
```

5.3.2 Vastaanotto ja edelleenlähetys Raspberryllä

Kun Arduino oltiin saatu konfiguroitua mittaamaan ja lähettämään lämpötiladataa, oli vuorossa Raspberryn konfigurointi. Tarkoituksena oli vastaanottaa Arduinon mitattu data samanlaisella NRF24L01-radiolähettimellä kuin Arduinoonkin liitetty. Kun lämpötiladata saadaan vastaanotettua, parsitaan se oikeaan muotoon ja edelleen lähetetään Internetin yli Thingsboardiin.

Raspberryn skriptin kielenä käytettiin pythonia. Skriptiin rakennettiin ensin vastaanotto ja tulostus Arduinon mittaamalle datalle. Raspberryn lähettimen kanavat ja lähetysputket määriteltiin vastaamaan aiemmin määriteltyjä Arduinon arvoja alla olevan mukaisesti. Kanavaksi määritettiin 76, joka vastaa 2476 Mhz taajuutta.

```
# Maaritetaan kaytettava putki Arduinon ja Raspberryn lahettimien valille
pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE1], [0xF0, 0xF0, 0xF0, 0xF0, 0xE1]]
# Maaritetaan lahetyksen arvot, oltava samat kuin Arduinolla. Maksiminopeus
2MBPS. Lahetysteho maaritetty minimiin.
radio.setPayloadSize(32)
radio.setChannel(0x76)
```

Vastaanotettava data määriteltiin tallennettavaksi Arduinolampo nimellä alla olevan mukaisesti.

```
#Maaritetaan muuttujat ja luetaan saapuva data Arduinolampoon
    Arduinolampo = []
    radio.read(Arduinolampo, radio.getDynamicPayloadSize())
```

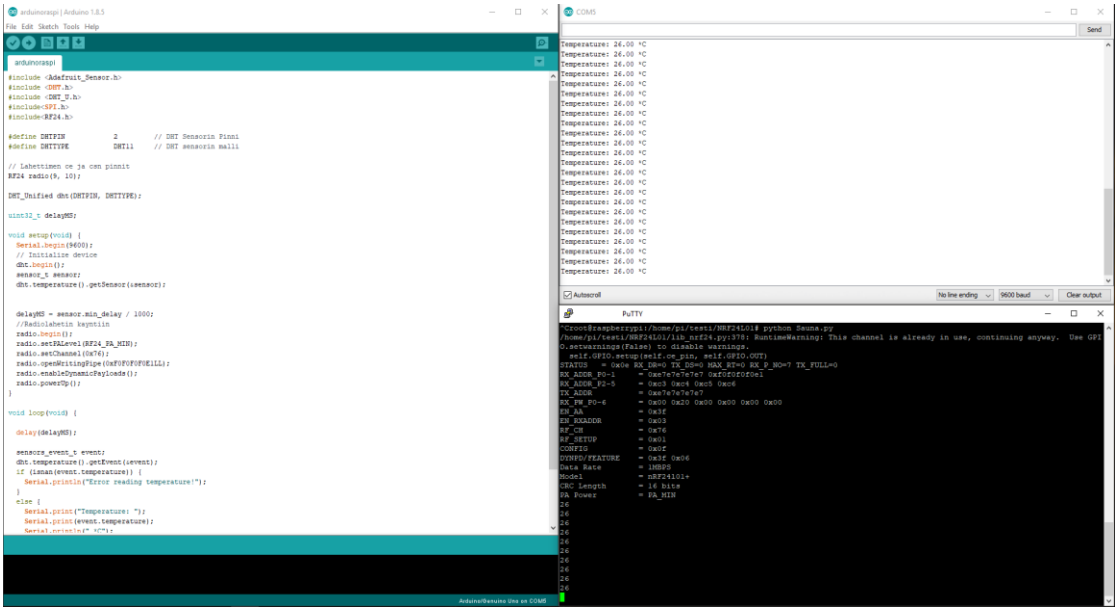
Kun Raspberry vastaanotti onnistuneesti mitattua dataa, konfiguroitiin skriptiin myös datan lähetykselle Thingsboardiin. Laitteelle generoitiin Thingsboardissa oma Access-Token datan yksilöimiseksi. Arvot määriteltiin skriptiin alla olevan mukaisesti.

```
#Thingsboardin IP-osoite
THINGSBOARD_HOST = 'xxx.xxx.xxx.xxx'
#Laitteen yksilollinen tunnus Thingsboardia varten
ACCESS_TOKEN = 'xxxxxxxxxxxxxxxxxx'
#Maaritellaan lahetysvali sekunteina
INTERVAL=1
```

Lämpötiladata oli tallennettuna ”Arduinolampo” muuttujaan, joka parsittiin oikeaan muotoon alla olevan mukaisesti. Tämän jälkeen lämpötiladata lähetettiin Thingsboardille.

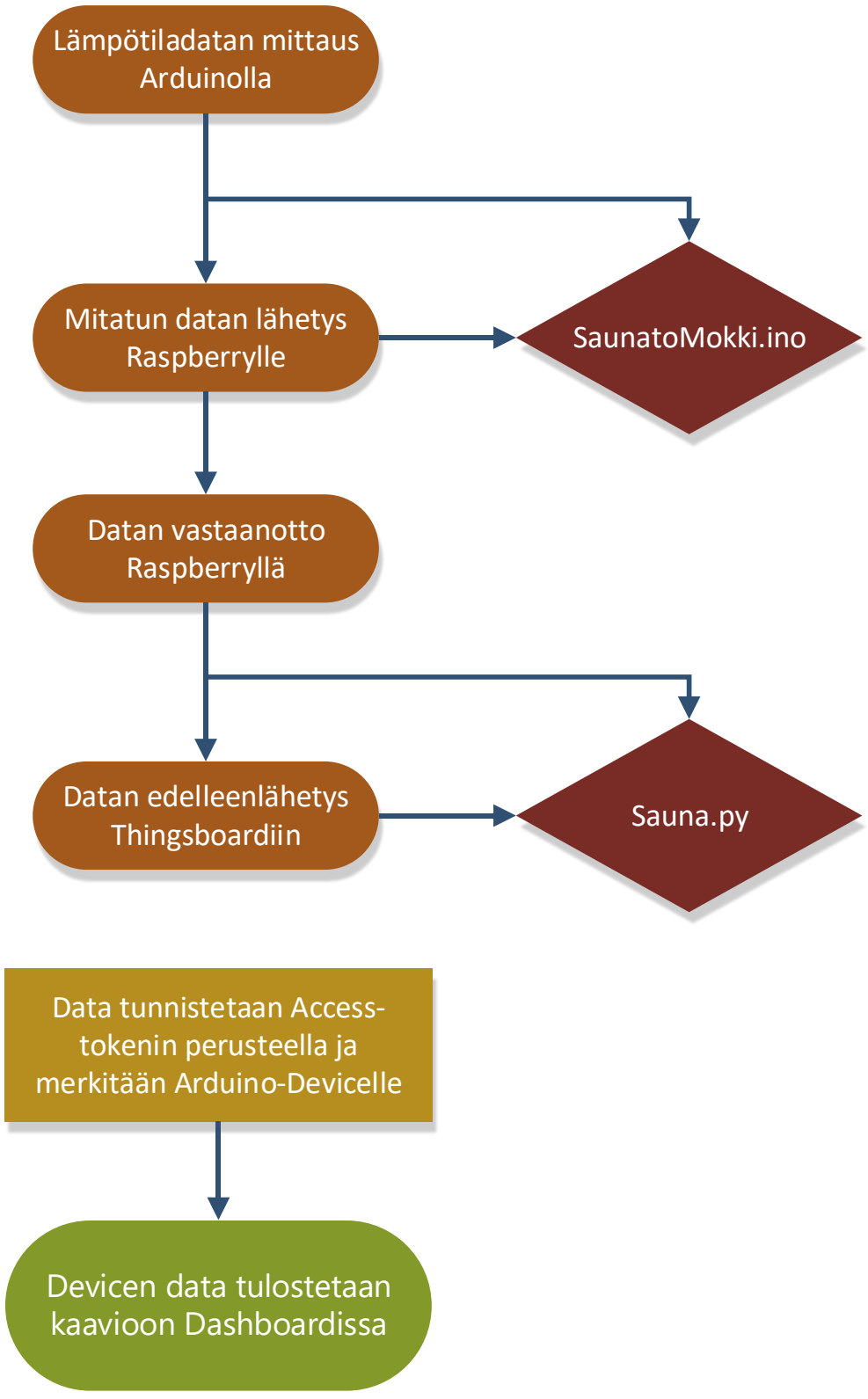
```
# Parsitaan luetusta datasta hakasulkeet pois, jolloin data voidaan lahettaa Thingsboardiin.
    lampo = 0
    lampo = str(Arduinolampo).strip('[]')
    sensor_data['temperature'] = lampo
#       print lampo # Mikali halutaan seurata cli:lta lampotilaa, poistetaan kommentointi
    client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)
```

Kuvion mukaisesti voitiin todeta, että Arduinon mittaamaa lämpötiladataa voitiin seurata Arduino IDE-ohjelmiston tulosteen, sekä Raspberryn Skriptin (ks. Liite 8) komentorivin tulosteen kautta (ks. Kuvio 18).



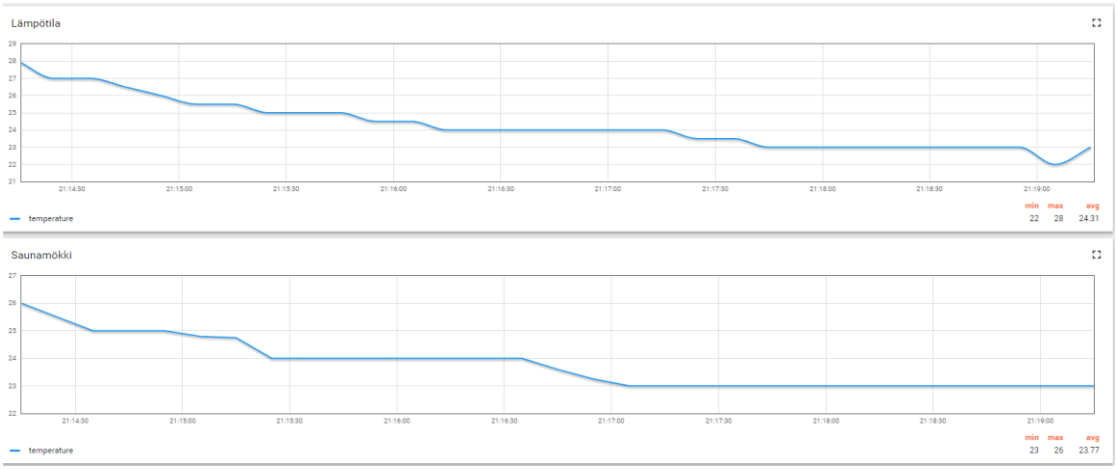
Kuvio 18. Arduinon tuloste ja Raspberryn vastaanoton tuloste

Arduinon lämpötiladatan mittaamiseen, lähettämiseen ja vastaanottoon tarvittiin siis kaksi skriptiä kahdelle eri laitteelle. Lisäksi Thingsboardille määriteltiin datan käytön määritelmät. Skriptien ja määritysten toimintaa selkeyttää Kuvio 19.



Kuvio 19. Lämpötiladatan kulkeutumisen topologia Arduinolta Thingsboardiin

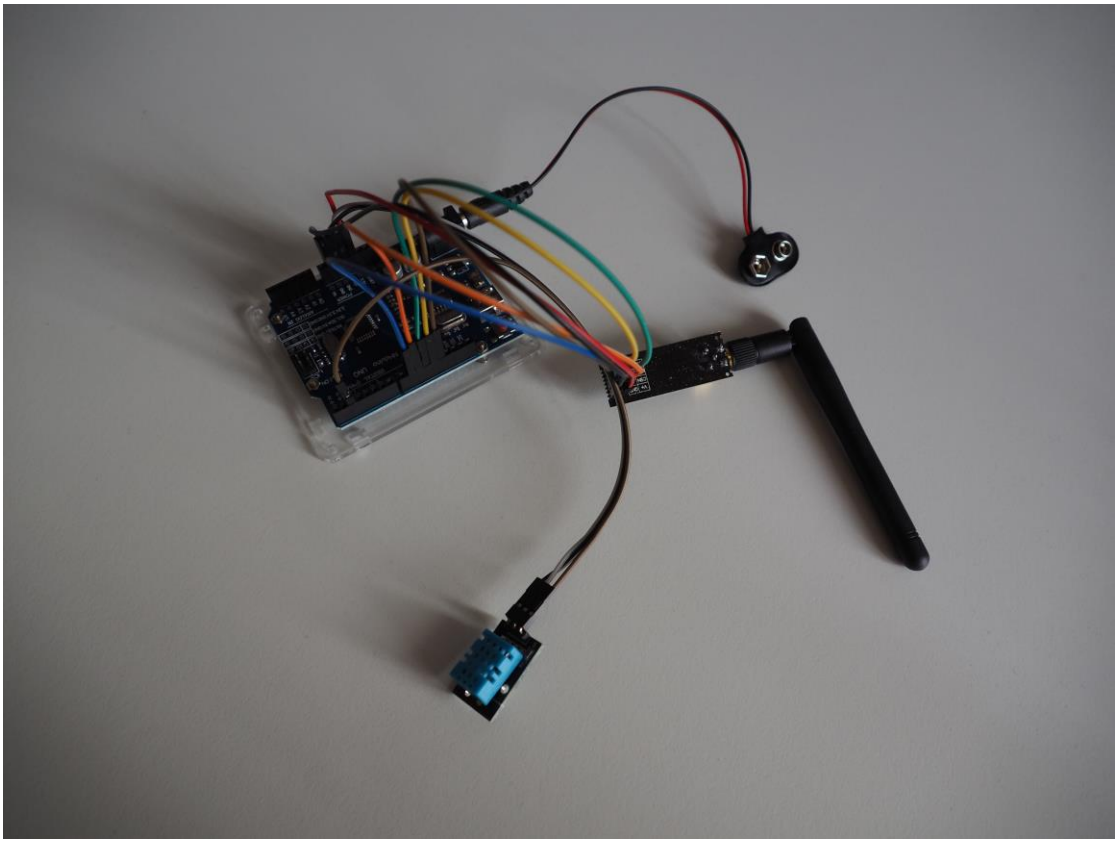
Arduinon lähettämää lämpötiladataa haluttiin myös seurata Thingsboardista, joten Dashboardiin määritettiin uusi graafinen kaavio. Tuohon kaavioon asetettiin datalähteeksi vain Arduinon Access-tokenille määritetty Device. Raspberryn ja Arduinon mittaamista datoista tehdyt kaaviot asetettiin allekkain. Kyseisellä hetkellä anturit olivat noin 30cm päässä toisistaan ja niihin puhallettiin lämmintä ilmaa, jonka jälkeen niiden annettiin olla huoneenlämmössä anturien reagoimisen todentamiseksi (ks. Kuvio 20).



Kuvio 20. Raspberryn ja Arduinon lämpötiladatat allekkain

5.3.3 Arduino2 konfigurointi

Sensoriverkkoon lisättiin myös toinen Arduino, joka on fyysisesti ja laitteistoltaan täysin identtinen ensimmäiseen Arduinoon nähden. Toiseen Arduinoon liitettiin NRF24L01-radiolähetin, sekä DHT-11-lämpötila-anturi (ks. Kuvio 21). Toisen Arduinon kohdalla jo tehtyjä skriptejä muokattiin sen verran, että data eroteltiin uudella Thingsboardissa generoidulla Access-tokenilla. Myöskin käytettävä dataputki vaihdettiin Arduinon ja Raspberryn lähettimien välillä.



Kuvio 21. Arduino 2, NRF24L01 ja DHT-11

5.4 Sääaseman tietojen liittäminen Thingsboardiin

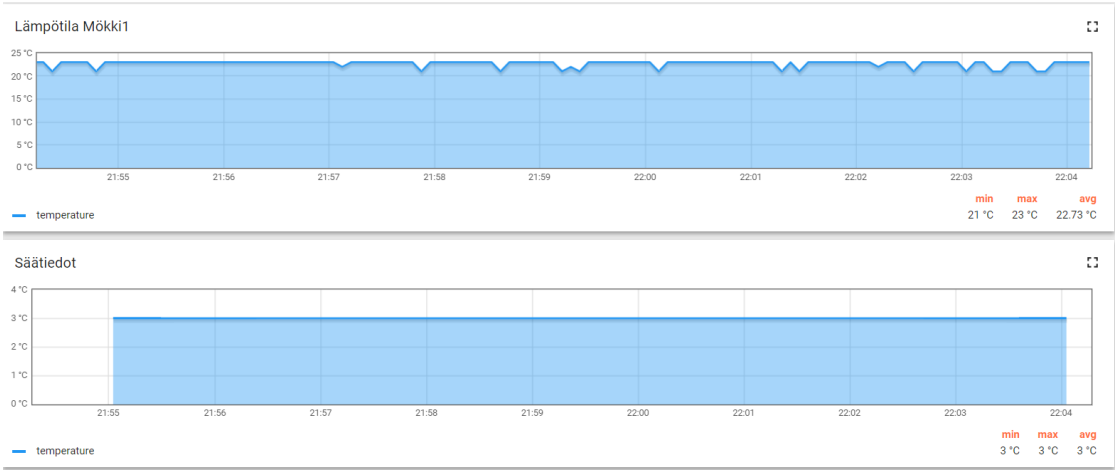
Jotta lämpötilan käyttäytymistä sisätiloissa ulkolämpötilan muuttuessa voitaisiin seurata, määritettiin Thingsboardiin ylimääräinen kuvaaja tätä varten. Sääaseman tiedot saatiin OpenWeatherMap palvelun kautta, joka tarjoaa erilaisia API-rajapintoja säätilan seurantaan. Ensimmäinen vaihtoehdoista on ilmainen ja sen ominaisuudet riittävät projektin tarkoituksiin (ks. Kuvio 22).

	Free	Startup	Developer	Professional	Enterprise
Price Price is fixed, no other hidden costs.	**Free**	**40 USD** / month	**180 USD** / month	**470 USD** / month	**2,000 USD** / month
Subscribe	Get API key and Start	Subscribe	Subscribe	Subscribe	Subscribe
Calls per minute (no more than)	60	600	3,000	30,000	200,000
Current weather API	✓	✓	✓	✓	✓
5 days/3 hour forecast API	✓	✓	✓	✓	✓
16 days/daily forecast API	-	✓	✓	✓	✓
Weather maps API	✓	✓	✓	✓	✓
Bulk download	-	-	-	✓	✓
UV index (beta)	✓	✓	✓	✓	✓
Air pollution (beta)	✓	✓	✓	✓	✓
Weather alerts (beta)	✓	✓	✓	✓	✓

Kuvio 22. OpenWeatherMapin tarjoamat sääpalvelut

Internetistä löytyy useita erilaisia esimerkkiskriptejä ja ohjeita säätietojen lataamiseen OpenWeatherMap:n palvelusta. Palveluun täytyi ensiksi rekisteröityä, jonka jälkeen käyttöön sai henkilökohtaisen API-KEYN. API-KEY on liitetty tilaukseen, joten projektissa sillä oli ilmaislisenssin oikeudet säätietoihin. Githubista löytyi Akora käyttäjän tekemä esimerkkiskripti, jota hyödynnettiin säätietojen latauksseen OpenWeatherMap:sta (ks. Liite 5). Toisena pohjana käytettiin Thingsboardin omaa mqtt-skriptiä (ks. Liite 3). Näiden kahden pohjalta ja joitakin rivejä lisäämällä saatiin toimiva skripti joka lataa säätiedot OpenWeatherMap:sta 60 sekunnin välein ja välittää ne AWS:ään Thingsboardiin (ks. Liite 6). Skripti nimettiin Saadata.py nimellä. OpenWeatherMap:n tarjoama data oli huomattavasti laajempaa, kuin projektin käyttöön tarvittiin (ks. Liite 9). Skriptin avulla siitä parsittiin vain lämpötiladata Thingsboardin käyttöön.

Saadata.py skripti lisättiin myös crontabbiin, kuten aiempi lämpötilamittauskin (ks. Kuvio 17). Tämän avulla säätietojen lataus käynnistyy myös automaattisesti taustalle, kun Raspberry Pi käynnistetään. Sääaseman lähettämästä datasta tehtiin graafinen kaavio, jotta ulkona muuttuvan säätilanteen arviointia sisälämpötilaan voitaisiin vertailla helposti. Kahdesta graafista nähdään nyt selkeästi ulkolämpötilan vaikutus sisälämpötilaan. Voidaan olettaa, että lämpötilan laskiessa ulkona on sen vaikutus sisälämpötilan muutokseen miltei suoraan verrannollinen (ks. Kuvio 23).

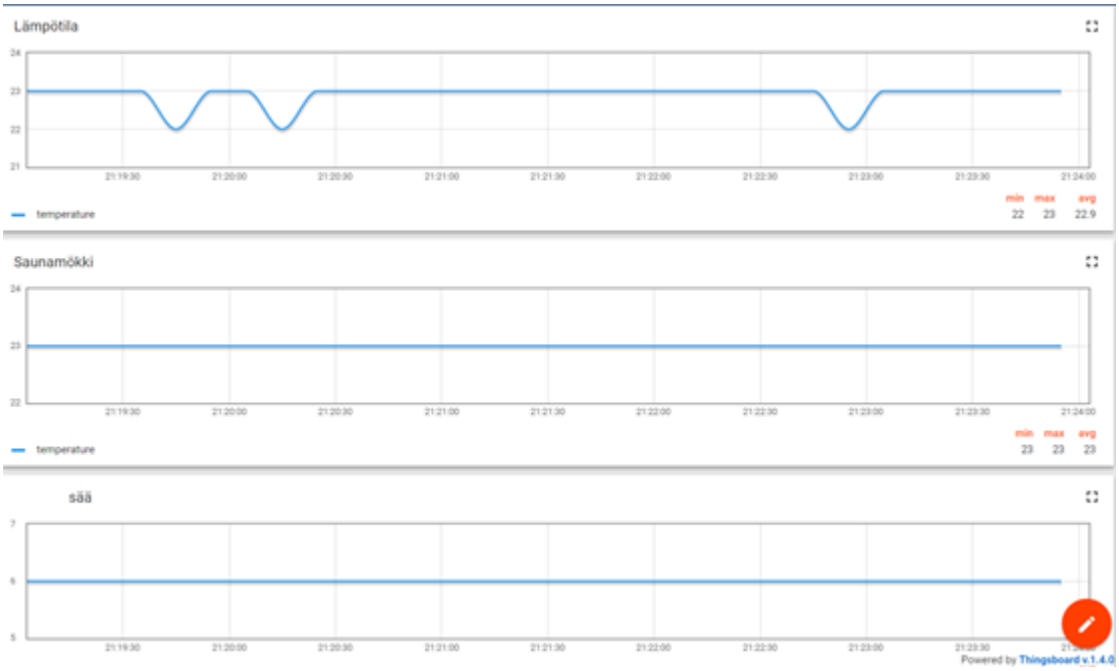


Kuvio 23. Mökin lämpötila ja sääaseman tiedot allekkain vertailukelpoisina

5.5 Käyttöliittymän määritykset

5.5.1 Käyttäjien määrittely

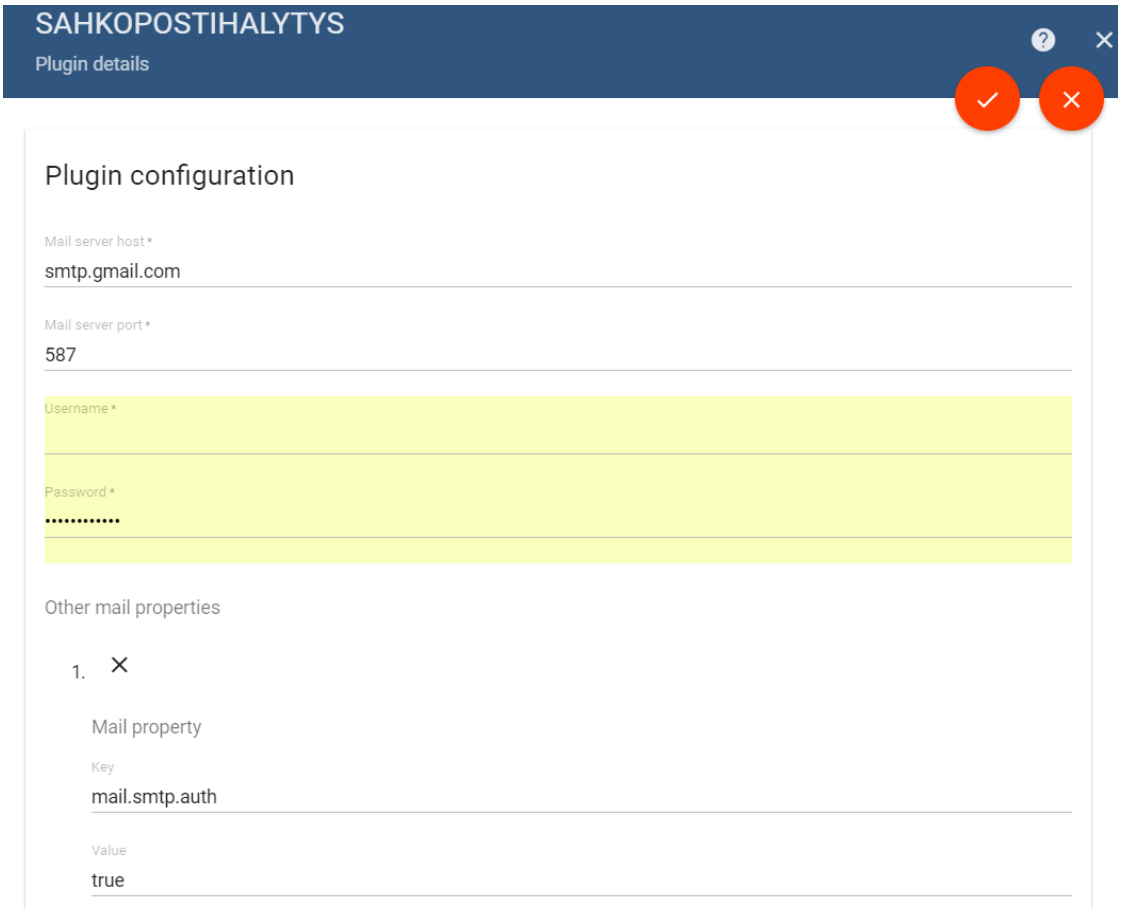
Thingsboardiin luotiin käyttäjä, jolla ei ole Admin-oikeuksia. Luonnollisesti tämän nimi oli Kayttaja. Käyttäjälle annettiin oikeus tarkkailla Arduinon, Raspberryn ja sääaseman dataa, jolloin käytössä oli kaikki data kaavioineen (ks. Kuvio 24).



Kuvio 24. Kayttajan lämpötilaseuranta

5.5.2 Sähköpostihälytysten määritys

Jotta mahdolliset suuret lämpötilamuutokset ja putkien jäätyminen kovilla pakkasilla voitaisiin estää, määritettiin Thingsboardiin sääntö, joka lähettää sähköpostin lämpötilan ylittäessä tietyn rajan. Tätä sääntöä varten tehtiin uusi gmail-tili vain kyseistä käyttötarkoitusta varten. Thingsboardiin tehtiin plugin, johon määritettiin gmailin SMTP-asetukset ja yhteys määritettiin käyttämään salattua yhteyttä (ks. Kuvio 25).



Kuvio 25. Sähköpostihälytys-pluginin määritys

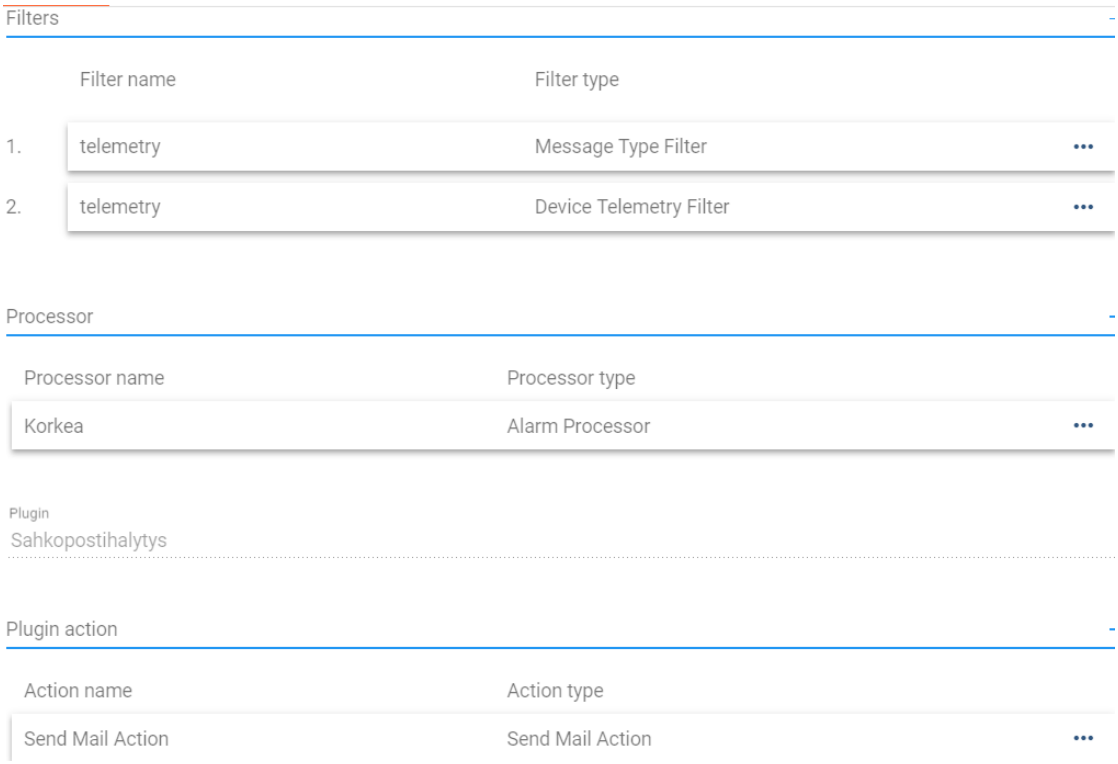
Thingsboardiin täytyi lisäksi luoda sääntö, jonka mukaan hälytyksiä tultaisiin lähettämään. Koska konfigurointihetkellä pakkasta ei ollut, tehtiin sääntö, joka ilmoittaa, mikäli mökin lämpötila nousee yli 27 asteen. Tämän avulla saatiin hälytysten toimivuus todennettua.

Sääntöön määriteltiin, että lähteenä käytetään mitattua lämpötilaa, jonka tyyppi on ”POST_TELEMETRY”, ja datana ”temperature” (ks. Kuvio 26). Tämän jälkeen yksinkertaisesti asetettiin hälytyksen generoinnille rajat. Hälytys luodaan, kun lämpötila nousee yli 27 asteen.

”temperature > 27”

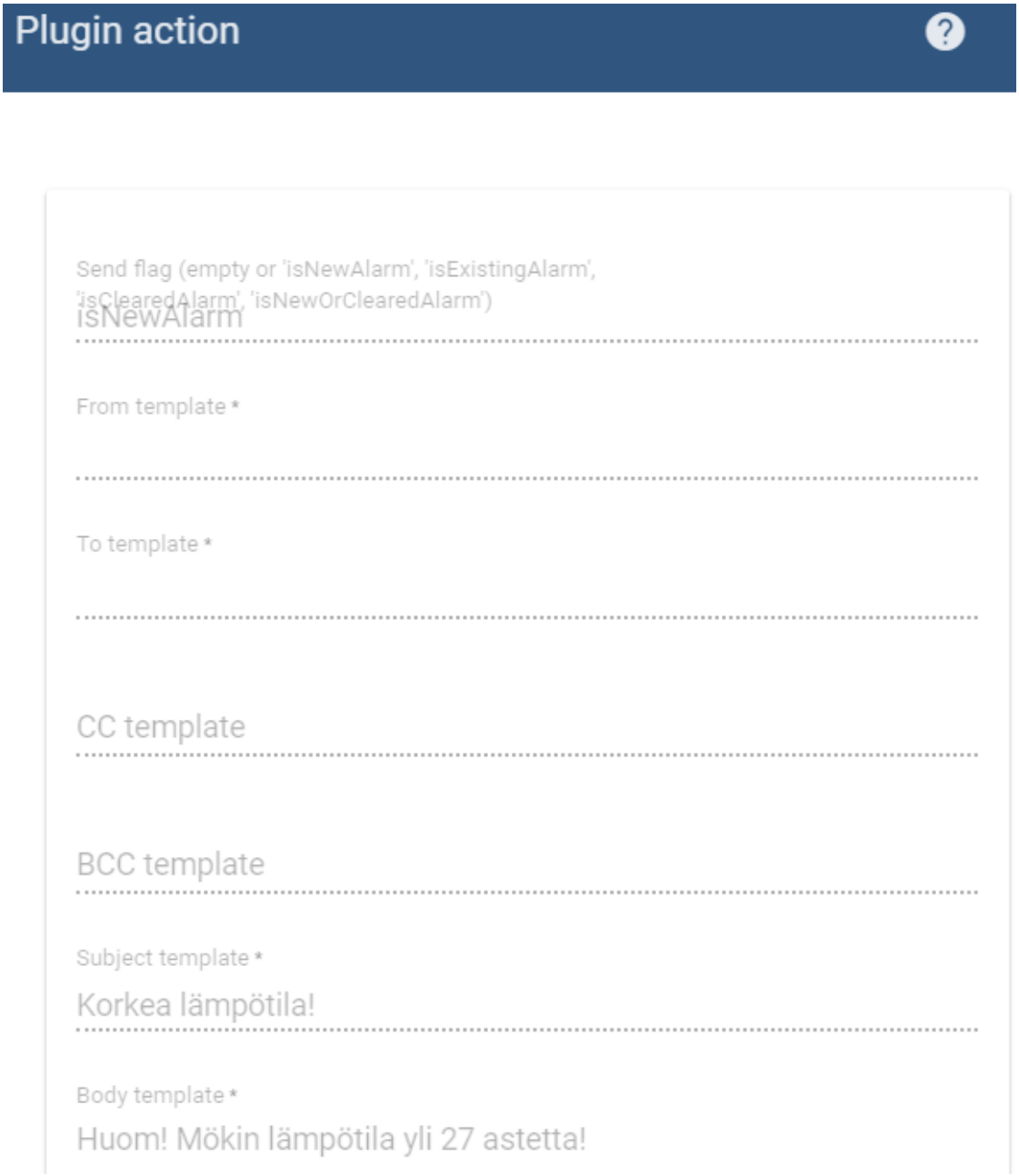
Ja hälytys nollataan, kun lämpötila laskee alle 27 asteen.

”temperature < 27”



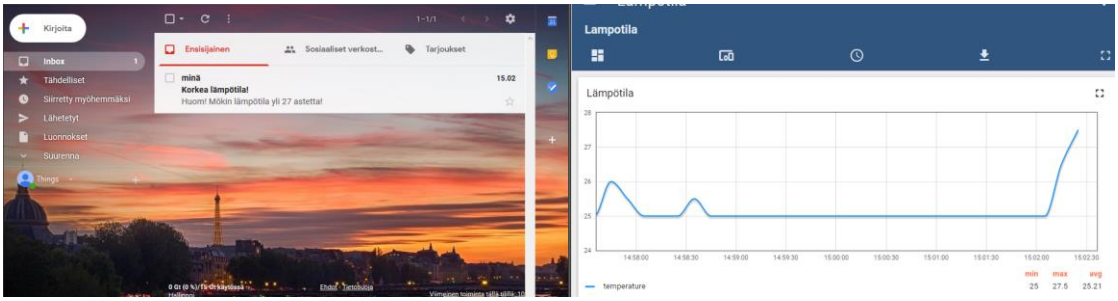
Kuvio 26. Korkean lämpötilan säännön määritykset

Varsinaisten hälytyksen arvojen jälkeen täytyi vielä määrittää itse sähköposti lähettämään hälytys. Pluginina käytettiin aiemmin luotua ”Sahkopostihalytys”:ta. Kuvio 27:n mukaisesti uuden hälytyksen generoitumisen yhteydessä lähetetään hälytys ”From template” sähköpostista, jona käytettiin aiemmin luotua uutta gmail-sähköpostia ja hälytys lähetetään ”To template”-sähköpostiin. Lisäksi määritettiin viestille Otsikko ja viestin sisältö.



Kuvio 27. Sähköpostin generoinnin määritys

Määritysten jälkeen hälytyksen toiminta testattiin lämmittämällä anturia, jotta lämpötila nousisi yli 27 asteen. Kuvio 28:sta voidaan huomata, että kun lämpötila nousee Thingsboardin kaaviossa yli 27 asteen, sähköpostiin lähetetään hälytys. Vastaavanlainen hälytys määritettiin, mikäli mökin lämpötila laskee alle 5 lämpöasteen. Tällöin tiedetään, että lämmitystä täytyisi käydä suurentamassa.



Kuvio 28. Sähköpostihälytys lähetetään lämpötilan noustessa yli 27 asteen

5.6 Internetyhteys

Raspberry oli yhdistettynä konfigurointivaiheessa langattomaan lähiverkkoon. Varsinaisessa sijoituspaikassa ei kuitenkaan ole aina saatavilla WLAN-verkkoa, joten laitteiston käyttöön hankittiin käytetty 3G-nettitikku (ks. Kuvio 29).

Kuvio 29. Huawei HiLink E353

Tarkemmin nettitikun malli oli Huawei HiLink E353. Se kykenee toimimaan 2G- ja 3G-verkossa, ja sen teoreettiset tiedonsiirtonopeudet ja tuetut taajuudet ilmenevät Taulukko 1:sta. Nettitikku konfiguroitiin Raspberrylle USB-Modeswitch paketin avulla. Nettitikkuun asetettiin Elisan SIM-kortti, jonka jälkeen yhteys toimi mobiilin kautta.

Taulukko 1. Huawei E353 tekniset ominaisuudet (Alkuperäinen kuvio Elisan verkkosivusto, Huawei E353 3G - nettitikku)

Tuetut taajuudet	2G: EDGE/GPRS/GSM 850/900/1800/1900MHz 3G: HSPA+/UMTS 2100/900MHz
Paras mahdollinen nopeus mobiiliverkossa	**3G** HSDPA: DL: 21 Mbit/s UL: 5,76 Mbit/s Verkon nopeuksien normaalit vaihteluvälit löydät täältä.
Tuki ulkoiselle antennille	Kyllä

6 Pohdinta ja jatkotoimenpiteet

Opinnäytetyön tavoitteena oli rakentaa langaton sensoriverkko, joka mahdollistaisi kolmen eri fyysisen sijainnen lämpötilan mittaamisen ja seuraamisen. Matkan varrella oli useita erilaisia hankaluuksia, mutta mielestäni ratkaisu palvelee käyttötarvetta varsin hyvin. Laitteisto asennetaan varsinaiseen sijaintiinsa kesän 2018 aikana, mutta sen toimivuus testattiin vastaavissa olosuhteissa.

Itse opinnäytetyön tekemisestä ja raportin kirjoittamisesta jäi iso määrä tietoa itselleni. Opinnäytetyötä lähdettiin tekemään hyvin niukoilla tiedoilla IoT-toteutuksista. Saman tyylistä sensoriverkkoa oli suunniteltu mielessä jo pitkään aiemmin, mutta ajan puutteen vuoksi se oli aina jäänyt toteuttamatta. Mikäli lähtisin tekemään uudelleen samanlaista projektia, en muuttaisi laitteistosta mitään, sillä ohjeita löytyi hyvin ja hankintakustannukset olivat suhteellisen pienet.

Alun perin sensoriverkon rakentamiseen oli hankittu RFM69 radiolähettimet. Kyseiset radiolähettimet osoittautuivat hyvin haastaviksi konfiguroida. Loppujen lopuksi konfiguraatiota ja toimivuutta yritettiin tehdä neljällä eri lähettimellä, joissa mukana oli kahta mallia, RFM69 ja RFM69H. Lähettimien kanssa vietettiin aikaa kymmeniä tunteja, jonka jälkeen hankittiin NRF24L01-lähettimet, joiden toimintaan saanti oli huomattavasti pienemmän vaivan takana. Ongelma saattoi olla juotoksissa piirilevyyn. Tämä toki on epätodennäköistä, sillä juotokset tehtiin neljään eri piirilevyyn ja yksikään lähettimistä ei toiminut. Internetistä löytyi hyvin heikosti tietoa lähettimien konfiguroinnista, joten virheiden etsiminenkin oli hieman hankalaa.

Kun laitteisto saadaan asennettua varsinaiseen toimintapaikkaansa mökkeihin, on mahdollista verkkoon liittää vielä lisää lämpötila-antureita. Esimerkiksi ulkolämpötilaa on mahdollista seurata suoraan mökin ulkoseinän luota lisäämällä yksi anturi verkkoon. Sen konfiguroiminen on hyvin helppoa edellisten mallin mukaisesti. Mikäli mökille halutaan jatkossa lisätä muita IoT-laitteita, kuten liiketunnistin, kamera tai mahdollisesti jopa lattialämmityksen säätö, voidaan kyseiset anturit ja säätimet integroida samaan sensoriverkkoon. Varsinkin lattialämmityksen säätö on mahdollisella toteutuslistalla, kun ensin saadaan kerättyä dataa ulkolämpötilojen vaikutuksesta mökin lämpötilaan. Lämpötila-antureita ja radiolähettimiä tilattiinkin opinnäytetyön viime metreillä lisää tulevaisuutta ajatellen.

Lähteet

About Company, Who we are. N.d. OpenWeatherMap:n verkkosivustolla. Viitattu 05.05.2018.
https://openweathermap.org/about

Aosong. Temperature and humidity module. N.d. DHT11 Product Manual. Viitattu 17.02.2018.
https://akizukidenshi.com/download/ds/aosong/DHT11.pdf

Amazon EC2. N.d. Amazonin verkkosivusto. 17.02.2018.
https://aws.amazon.com/ec2/

Amazon EC2 Instance types. N.d. Amazon Web Services (AWS) verkkosivuston taulukko. Viitattu 17.02.2018.
https://aws.amazon.com/ec2/instance-types/

Amazon Elastic Compute Cloud. 2018. Amazon User Guide for Linux Instances – verkkosivusto. Julkaistu 2018. Viitattu 17.02.2018.
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html

Arduino Uno Rev3. N.d. Arduino Store-verkkosivusto. Viitattu 18.02.2018
https://store.arduino.cc/arduino-uno-rev3

Desktop Features. N.d. Ubuntun verkkosivusto. Viitattu 17.02.2018.
https://www.ubuntu.com/desktop/features

DHT-11 Digital Temperature and Humidity Sensor Temperature sensor Arduino. N.d. Ebay verkkosivusto. Viitattu 04.02.2018.
https://www.ebay.com/itm/DHT11-DHT-11-Digital-Temperature-and-Humidity-Sensor-Temperature-sensor-Arduino/400489574221?hash=item5d3f09ef4d:g:t7gAAOSwRgJXiJFT

Dirkk. 2017. Raspberry Pi 3 GPIO Pin Chart with Pi. Openclipart-verkkosivusto. Julkaistu 06.05.2017. Viitattu 10.03.2018.
https://openclipart.org/detail/280972/raspberry-pi-3-gpio-pin-chart-with-pi

Get Ubuntu. N.d. Ubuntun verkkosivusto. Viitattu 18.02.2018.
https://www.ubuntu.com/download

Huawei E353 3G - nettitikku. N.d. Elisan verkkosivusto. Viitattu 15.05.2018.
https://elisa.fi/asiakaspalvelu/aihe/mobiililaajakaista/ohje/huawei-E353/#ominaisuudet

Installation options. N.d. Thingsboard dokumentaatio. Viitattu 17.02.2018.
https://thingsboard.io/docs/user-guide/install/installation-options/

Installing ThingsBoard on Linux. N.d. Thingsboard dokumentaatio. Viitattu 10.03.2018.
https://thingsboard.io/docs/user-guide/install/linux/

Kora, A. 2015. openWeathermap.py-skripti. Github-verkkosivusto. Julkaistu 28.01.2015. Viitattu 11.04.2018.

https://github.com/akora/openweathermap-python/blob/master/openWeatherMap.py

Lady, A. 2018. DHT11, DHT22 and AM2303 Sensors. Julkaistu 12.01.2018. Viitattu 17.02.2018.
https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf

Liu, T. N.d. Digital relative humidity & temperature sensor AM2302/DHT22, AM2302/DHT22 product manual. Viitattu 17.02.2018.
https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf

NEW UNO R3 ATmega328P CH340 Mini USB Board for Compatible-Arduino. N.d. Ebay verkkosivusto. Viitattu 18.02.2018.
https://www.ebay.com/itm/NEW-UNO-R3-ATmega328P-CH340-Mini-USB-Board-for-Compatible-Arduino/311155383820?epid=838665863&hash=item48724e5e0c:g:UswAAOSwNnRYfC0R

Piyush, R. 2017. Mqtt presentation - slideshare. Julkaistu 06.03.2017. Viitattu 10.03.2018.
http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

Raspberry Pi 3 Model B Specifications. N.d. Raspberryn verkkosivusto. Viitattu 17.02.2018.
https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Raspbian Documentation. N.d. Rasperry Pi:n verkkosivusto. Viitattu 17.02.2018.
https://www.raspberrypi.org/documentation/raspbian/

Rouse M. 2018. What is MQTT? Julkaistu 02.2018. Viitattu 10.03.2018.
https://www.slideshare.net/piyushrathi52/mqttmessage-queue-telemetry-protocol-presentation

Single chip 2.4 GHz Transceiver. 2006. Nordic Semiconductor ASA:n tuoteseloste. Julkaistu 03.2006. Viitattu 11.04.2018.
https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf

SM. 2017. Getting started with Arduino and Genuino products. Arduino – Getting started. Julkaistu 20.10.2017. Viitattu 18.02.2018
https://www.arduino.cc/en/Guide/HomePage

Thingsboard – Open-source IoT Platform. N.d. Thingsboardin verkkosivusto. Viitattu 17.02.2018.
https://thingsboard.io/

Ubuntu Kylin. N.d. Ubuntun verkkosivusto. Viitattu 17.02.2018.
http://www.ubuntukylin.com/index.php?lang=en

Weather API. N.d. OpenWeatherMap:n verkkosivusto. Viitattu 05.05.2018.
https://openweathermap.org/api

What is Arduino? Arduino – Introduction. N.d. Arduinon verkkosivusto. Viitattu 18.02.2018.
https://www.arduino.cc/en/Guide/Introduction

Working with telemetry data. N.d. Thingsboard Documentation. 17.02.2018.
https://thingsboard.io/docs/user-guide/telemetry/

Liitteet

Liite 1. Amazon EC2 Instance types - Amazon Web Services (AWS) verkkosivuston taulukko

Instance Type	vCPU	Memory (GiB)	Storage (GB)	Networking Performance	Physical Processor	Clock Speed (GHz)	Intel AVX†	Intel AVX2†	Intel Turbo	EBS OPT	Enhanced Networking†
t2.nano	1	0.5	EBS Only	Low	Intel Xeon family	up to 3.3	Yes	-	Yes	-	-
t2.micro	1	1	EBS Only	Low to Moderate	Intel Xeon family	Up to 3.3	Yes	-	Yes	-	-
t2.small	1	2	EBS Only	Low to Moderate	Intel Xeon family	Up to 3.3	Yes	-	Yes	-	-
t2.medium	2	4	EBS Only	Low to Moderate	Intel Xeon family	Up to 3.3	Yes	-	Yes	-	-
t2.large	2	8	EBS Only	Low to Moderate	Intel Xeon family	Up to 3.0	Yes	-	Yes	-	-
t2.xlarge	4	16	EBS Only	Moderate	Intel Xeon family	Up to 3.0	Yes	-	Yes	-	-
t2.2xlarge	8	32	EBS Only	Moderate	Intel Xeon family	Up to 3.0	Yes	-	Yes	-	-
m5. large	2	8	EBS Only	High	Intel Xeon Platinum	2.5	Yes	Yes	Yes	Yes	Yes
m5.xlarge	4	16	EBS Only	High	Intel Xeon Platinum	2.5	Yes	Yes	Yes	Yes	Yes
m5.2xlarge	8	32	EBS Only	High	Intel Xeon Platinum	2.5	Yes	Yes	Yes	Yes	Yes
m5.4xlarge	16	64	EBS Only	High	Intel Xeon Platinum	2.5	Yes	Yes	Yes	Yes	Yes
m5.12xlarge	48	192	EBS Only	10 Gigabit	Intel Xeon Platinum	2.5	Yes	Yes	Yes	Yes	Yes
m5.24xlarge	96	384	EBS Only	25 Gigabit	Intel Xeon Platinum	2.5	Yes	Yes	Yes	Yes	Yes
m4.large	2	8	EBS Only	Moderate	Intel Xeon E5-2676 v3*	2.4	Yes	Yes	Yes	Yes	Yes
m4.xlarge	4	16	EBS Only	High	Intel Xeon E5-2676 v3*	2.4	Yes	Yes	Yes	Yes	Yes
m4.2xlarge	8	32	EBS Only	High	Intel Xeon E5-2676 v3*	2.4	Yes	Yes	Yes	Yes	Yes
m4.4xlarge	16	64	EBS Only	High	Intel Xeon E5-2676 v3*	2.4	Yes	Yes	Yes	Yes	Yes
m4.10xlarge	40	160	EBS Only	10 Gigabit	Intel Xeon E5-2676 v3	2.4	Yes	Yes	Yes	Yes	Yes
m4.16xlarge	64	256	EBS Only	25 Gigabit	Intel Xeon E5-2686 v4	2.3	Yes	Yes	Yes	Yes	Yes

Liite 2. MQTT viestien selitykset

MQTT Message	Description
CONNECT	Client request to connect to server
CONNACK	Connect acknowledgement
PUBLISH	Publish message
PUBACK	Publish acknowledgement
PUBREC	Publish received
PUBREL	Publish release
PUBCOMP	Publish complete
SUBSCRIBE	Client subscribe request
SUBACK	Subscribe acknowledgement
UNSUBSCRIBE	Unsubscribe request
UNSUBACK	Unsubscribe acknowledgement
PINGREQ	PING request
PINGRESP	PING response
DISCONNECT	Client is disconnecting

Liite 3. mqtt-dht22.py

```
import os
import time
import sys
import Adafruit_DHT as dht
import paho.mqtt.client as mqtt
import json
```

```
THINGSBOARD_HOST = '<IP-Osoite>'
ACCESS_TOKEN = '<access_token>'

# Data capture and upload interval in seconds. Less interval will
# eventually hang the DHT22.

INTERVAL=5
sensor_data = {'temperature': 0, 'humidity': 0}
next_reading = time.time()
client = mqtt.Client()

# Set access token
client.username_pw_set(ACCESS_TOKEN)

# Connect to ThingsBoard using default MQTT port and 60 seconds keepalive interval
client.connect(THINGSBOARD_HOST, 1883, 60)
client.loop_start()
try:
   while True:
      humidity,temperature = dht.read_retry(dht.DHT11, 4)
      humidity = round(humidity, 2)
      temperature = round(temperature, 2)
      print(u"Temperature: {:g}\u00b0C, Humidity: {:g}%".format(temperature, hu-
midity))
      sensor_data['temperature'] = temperature
      sensor_data['humidity'] = humidity

      # Sending humidity and temperature data to ThingsBoard
      client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)
      next_reading += INTERVAL
      sleep_time = next_reading-time.time()
      if sleep_time > 0:
```

```
            time.sleep(sleep_time)
except KeyboardInterrupt:
    pass
client.loop_stop()
client.disconnect()
```

Liite 4. Raspberrytbupload.py

```
#!/usr/bin/env python
import os
import time
import sys
import Adafruit_DHT as dht
import paho.mqtt.client as mqtt
import json
#Thingsboardin IP-osoite
THINGSBOARD_HOST = '<IP-osoite>'
#Laitteen yksilollinen tunnus Thingsboardia varten
ACCESS_TOKEN = '<Access-token>'
# Datan luku ja lahetys ajanjakso sekunteina
INTERVAL=5
sensor_data = {'temperature': 0, 'humidity': 0}
next_reading = time.time()
client = mqtt.Client()
#  Kaytetaan Raspberryn DHT-anturin Access Tokenia
client.username_pw_set(ACCESS_TOKEN)
# Yhdistetaan Thingsboardiin MQTT:lla porttia 1883 kayttaen. 60 sekunnin keepalive
client.connect(THINGSBOARD_HOST, 1883, 60)
client.loop_start()
try:
    while True:
# Luetaan lampotiladataa DHT-anturilta, joka on liitettyna GPIO pinniin numero 4
```

```
        humidity,temperature = dht.read_retry(dht.DHT11, 4)
# Pyoristetaan muuttujien arvot 2 desimaalin tarkkuudella
        humidity = round(humidity, 2)
        temperature = round(temperature, 2)
# Maaritetaan mitatut arvot lahetettavaksi Thingsboardille
        sensor_data['temperature'] = temperature
        sensor_data['humidity'] = humidity
# Lahetetaan mitattu lampotila- ja kosteusdata Thingsboardille
        client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)
                next_reading += INTERVAL
        sleep_time = next_reading-time.time()
        if sleep_time > 0:
          time.sleep(sleep_time)
except KeyboardInterrupt:
   pass
client.loop_stop()
client.disconnect()
```

Liite 5. openWeatherMap.py

```
#!/usr/bin/python

# OpenWeatherMap service provides open weather data for more than 200,000
# cities and any geo location that is available on our website and through API.

import sys
import math
import requests

base_url = 'http://api.openweathermap.org/data/2.5/weather'
api_key = ''  # << Get your API key (APPID) here: http://openweathermap.org/appid
cities = ['Esztergom', 'London, UK']
```

```
def get_temperature(city):
  query = base_url + '?q=%s&units=metric&APPID=%s' % (city, api_key)
  try:
    response = requests.get(query)
    # print("[%s] %s" % (response.status_code, response.url))
    if response.status_code != 200:
      response = 'N/A'
      return response
    else:
      weather_data = response.json()
      return weather_data
  except requests.exceptions.RequestException as error:
    print error
    sys.exit(1)
def main():
  for city in cities:
    location = get_temperature(city)
    print location['name'] + ' ' + str(math.ceil(location['main']['temp'])) + ' C (' + str(lo-
cation['main']['temp']) + ')'
if __name__ == '__main__':
  main()
# TODO: fix TypeError: string indices must be integers, not str if base_url is incorrect
```

Liite 6. Saadata.py

```
#!/usr/bin/env python
#!/usr/bin/python
# OpenWeatherMap service provides open weather data for more than 200,000
# cities and any geo location that is available on our website and through API.
```

```
import time

import sys

import math

import requests

import paho.mqtt.client as mqtt

import json

# Maaritetaan Thingsboardin palvelimen IP-osoite, tunnus datalle, OpenWeather-
Map:n osoite, seka henkilokohtainen api_key. Lisaksi maaritellaan paikkakunta, josta
lampotiladataa halutaan.

THINGSBOARD_HOST = '<IP-osoite>'

ACCESS_TOKEN = '<Access-token>'

base_url = 'http://api.openweathermap.org/data/2.5/weather'

api_key = '<api-key>'

cities = ['<Kaupunki>, <Maakoodi>']

# Maaritetaan aikavali sekunteina, kuinka usein saadataa haetaan.

INTERVAL=60

sensor_data = {'temperature': 0}

next_reading = time.time()

client = mqtt.Client()

# Kaytetaan tunnusta, jolla yksiloidaan data

client.username_pw_set(ACCESS_TOKEN)

# Yhdistetaan Thingsboardiin MQTT:lla porttia 1883 kayttaen. 60 sekunnin keepalive

client.connect(THINGSBOARD_HOST, 1883, 60)

client.loop_start()


try:

   while True:

# Haetaan lampotiladataa kayttaen maariteltua paikkakuntaa ja henkilokohtaista
api_keyta.

      def get_temperature(city):

        query = base_url + '?q=%s&units=metric&APPID=%s' % (city, api_key)

        try:

          response = requests.get(query)
```

```
        # print("[%s] %s" % (response.status_code, response.url))
        if response.status_code != 200:
          response = 'N/A'
          return response
        else:
          weather_data = response.json()
          return weather_data
      except requests.exceptions.RequestException as error:
        print error
        sys.exit(1)
    def main():
      for city in cities:
# Tallennetaan lampotila lampo-muuttujaan.
        location = get_temperature(city)
        lampo = str(math.ceil(location['main']['temp']))
       # print lampo # Mikali skripti ajetaan manuaalisesti, voidaan kommentointi
poistamalla seurata lampotilan muutosta cli:lta.
# Maaritellaan lampo-muuttujan arvo lahetettavaksi Thingsboardille ja lahetetaan se.
        sensor_data['temperature'] = lampo
    if __name__ == '__main__':
      main()
    client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)
    next_reading += INTERVAL
    sleep_time = next_reading-time.time()
    if sleep_time > 0:
      time.sleep(sleep_time)
except KeyboardInterrupt:
  pass
client.loop_stop()
client.disconnect()
```

Liite 7. SaunatoMokki.ino

```
#include <Adafruit_Sensor.h>

#include <DHT.h>

#include <DHT_U.h>

#include<SPI.h>

#include<RF24.h>

#define DHTPIN          2        // DHT Sensorin Pinni

#define DHTTYPE         DHT11    // DHT sensorin malli

// Lahettimen ce ja csn pinnit

RF24 radio(9, 10);

DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;

void setup(void) {

 Serial.begin(9600);

// Maaritetaan lampotilan mittaus

 sensor_t sensor;

 dht.temperature().getSensor(&sensor);

// Maaritetaan viive mittausten valille

 delayMS = sensor.min_delay / 1000;

// Kaynnistetaan radiolahetin ja asetetaan lahetysteho minimille virran saastamiseksi

 radio.begin();

 radio.setPALevel(RF24_PA_MIN);

// Maaritetaan kanava, jossa lahetetaan. Maaritykset oltava samat, kuin vastaanotta-
valla Raspberrylla

 radio.setChannel(0x76);
```

```
  radio.openWritingPipe(0xF0F0F0F0E1LL);
  radio.enableDynamicPayloads();
  radio.powerUp();
}
void loop(void) {
// Kaytetaan aiemmin maariteltya viivetta lampotilojen mittauksessa
  delay(delayMS);
// Luetaan lampotila
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println("Error reading temperature!");
  }
  else {
// Tulostetaan Serialin kautta mitattu lampotila
    Serial.print("Temperature: ");
    Serial.print(event.temperature);
    Serial.println(" *C");
    const char lampo = (event.temperature);
// Lahetetaan mitattu lampo radiolahettimen kautta. Arvo tallennettu lampo-muut-
tujaan.
    radio.write(&lampo, sizeof(lampo));
  }
  delay(1000);
```

```
}
```

Liite 8. Sauna.py

```
#!/usr/bin/env python

import os

import time

import sys

import RPi.GPIO as GPIO

from lib_nrf24 import NRF24

import spidev

import paho.mqtt.client as mqtt

import json

#Thingsboardin IP-osoite

THINGSBOARD_HOST = '<IP-osoite>'

#Laitteen yksilollinen tunnus Thingsboardia varten

ACCESS_TOKEN = '<Access-token>'

#Maaritellaan lahetysvali sekunteina

INTERVAL=1

client = mqtt.Client()

next_reading = time.time()

sensor_data = {'temperature': 0}

# Arduinon DHT-anturin Access Token

client.username_pw_set(ACCESS_TOKEN)

# Yhdistetaan Thingsboardiin MQTT:lla porttia 1883 kayttaen. 60 sekunnin keepalive
```

```
client.connect(THINGSBOARD_HOST, 1883, 60)

client.loop_start()

GPIO.setmode (GPIO.BCM)

# Maaritetaan kaytettava putki Arduinon ja Raspberryn lahettimien valille

pipes = [[0xE8, 0xE8, 0xF0, 0xF0, 0xE1], [0xF0, 0xF0, 0xF0, 0xF0, 0xE1]]

radio = NRF24(GPIO, spidev.SpiDev())

radio.begin(0, 17)

# Maaritetaan lahetyksen arvot, oltava samat kuin Arduinolla. Maksiminopeus
2MBPS. Lahetysteho maaritetty minimiin.

radio.setPayloadSize(32)

radio.setChannel(0x76)

radio.setDataRate(NRF24.BR_1MBPS)

radio.setPALevel(NRF24.PA_MIN)

radio.setAutoAck (True)

radio.enableDynamicPayloads()

radio.enableAckPayload()

# Aloitetaan vastaanotto ja printataan lahettimen tiedot ja arvot, mikali skripti aje-
taan manuaalisesti.

radio.openReadingPipe(1, pipes[1])

radio.printDetails()

radio.startListening()

try:

   while True:

      while not radio.available(0):

         time.sleep(1/100)
```

```
#Maaritetaan muuttujat ja luetaan saapuva data Arduinolampoon
        Arduinolampo = []
        radio.read(Arduinolampo, radio.getDynamicPayloadSize())
        # Parsitaan luetusta datasta hakasulkeet pois, jolloin data voidaan lahettaa
Thingsboardiin.
        lampo = 0
        lampo = str(Arduinolampo).strip('[]')
        sensor_data['temperature'] = lampo
#       print lampo # Mikali halutaan seurata cli:lta lampotilaa, poistetaan kommen-
tointi
        client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)
        next_reading += INTERVAL
        sleep_time = next_reading-time.time()
        if sleep_time > 0:
           time.sleep(sleep_time)
except KeyboardInterrupt:
   pass
client.loop_stop()
client.disconnect()
```

Liite 9. OpenWeatherMap:n tarjoama säädata parsimattomana

```
cod             "200"
message         0.0023
cnt             3
list
0
```

dt	1523480400
main	
temp	271.33
temp_min	268.878
temp_max	271.33
pressure	1026.87
sea_level	1041.87
grnd_level	1026.87
humidity	78
temp_kf	2.45
weather	
0	
id	801
main	"Clouds"
description	"few clouds"
icon	"02n"
clouds	
all	12
wind	
speed	2.51
deg	247.001
sys	
pod	"n"
dt_txt	"2018-04-11 21:00:00"
1	
dt	1523491200
main	
temp	269.42
temp_min	267.584
temp_max	269.42
pressure	1027.28
sea_level	1042.29

grnd_level 1027.28

humidity 73

temp_kf 1.84

weather

0

id 800

main "Clear"

description "clear sky"

icon "01n"

clouds

all 0

wind

speed 2.18

deg 255.501

sys

pod "n"

dt_txt "2018-04-12 00:00:00"

2

dt 1523502000

main

temp 266.28

temp_min 265.049

temp_max 266.28

pressure 1027.48

sea_level 1042.56

grnd_level 1027.48

humidity 74

temp_kf 1.23

weather

0

id 800

main "Clear"

description	"clear sky"
icon	"01d"
clouds	
all	0
wind	
speed	1.78
deg	249.002
sys	
pod	"d"
dt_txt	"2018-04-12 03:00:00"
city	
id	
name	"<Kaupunki>"
coord	
lat	
lon	
country	"<Maakoodi>"
population	