

KARELIA-AMMATTIKORKEAKOULU
Tietotekniikan koulutus

Ville Mutanen

OLOSUHDESEURANTA SENSORIVERKOLLA

Opinnäytetyö
Toukokuu 2020



OPINNÄYTETYÖ
Toukokuu 2020
Tietotekniikan koulutus

Karjalankatu 3
80200 JOENSUU
(013) 260 600

Tekijä
Ville Mutanen

Nimeke
Olosuhdeseuranta sensoriverkolla

Tiivistelmä

Tässä opinnäytetyössä toteutettiin olosuhdeseurantalaitteisto sisätilan lämpötilan ja valoisuuden mittaamiseen. Työn tarkoitus oli selvittää, onko kokoonpano mahdollinen ja mitä toteutukseen vaaditaan. Työssä selitetään kokonaisuudessa käytetyt teknologiat, mitä konsepti tarkoittaa, miten käytännön toteutus on hoidettu, työn tulokset, ja lopuksi on pohdinta.

Keskeisimpiä fyysisiä komponentteja toteutuksessa olivat ZigBee-protokollaa käyttävät XBee-radiot ja niihin kytketyt anturit sekä Raspberry Pi, joka toimi kokonaisuuden yhdyskäytävälaitteena. Laitteiden väliseen viestintään käytettiin MQTT-protokollaa ja tiedonkäsittely tapahtui Node-RED:iä käyttäen.

Opinnäytetyössä saatiin toteutettua laitteisto, joka mittasi lämpötilaa ja valoisuutta. Toteutuksen lopputulos jäi kuitenkin puutteelliseksi, sillä suunniteltua pilvipalvelukomponenttia ei toteutettu. Opinnäytetyö antaa pohjan mahdolliselle jatkokehitykselle, jos laitteiston haluaisi kaupallistaa.

Kieli

suomi

Asiasanat

IoT, Node-RED, XBee, ZigBee

Sivuja 15

Liitteet 1

Liitesivumäärä 1



THESIS
May 2020
**Degree Programme in Information
Technology**

Karjalankatu 3
80200 JOENSUU
FINLAND
(013) 260 600

Author
Ville Mutanen

Title
Environmental monitoring with sensor network

Abstract

In this thesis an environmental monitoring system for measuring temperature and brightness of interior was implemented. The goal of the thesis was to find out if it's possible to build an environmental monitoring system and if possible what the requirements for such a system are. Thesis explains used technologies and the concept. It also tells how the practical implementation was done and what the results are. The last part is the conclusion.

Essential physical components in the product were XBee radios using ZigBee protocol and the sensors attached to the radios. Also Raspberry Pi working as a gateway. Communication between devices was handled with MQTT protocol and data processing was done with Node-RED.

Thesis managed to deliver a system to monitor temperature and brightness. The final product lacked properties as a planned component for cloud connection was not implemented. The thesis provides foundation for future development for marketising the product.

Language

Finnish

Pages 15
Appendices 1
Pages of Appendices 1

Keywords

IoT, Node-RED, XBee, ZigBee

Sisältö

1	Johdanto	6
2	Käytetyt teknologiat.....	6
	2.1 XBee-radio ja ZigBee-protokolla	6
	2.2 Raspberry Pi	7
	2.3 MQTT	7
	2.4 Node-RED.....	7
3	IoT-salkku	8
	3.1 Konsepti	8
	3.2 Fyysinen sisältö.....	8
	3.2.1 XBee.....	9
	3.2.2 Raspberry Pi.....	10
	3.3 Looginen rakenne	10
4	Olosuhdeseurantalaitteiston asennus ja toiminta.....	11
5	Tulokset	13
6	Pohdinta.....	13
	Lähteet.....	15

Liitteet

Liite 1 Osaluettelo fyysisistä komponenteista

Lyhenteet

IEEE Institute of Electrical and Electronics Engineers

IoT Internet of Things, asioiden/esineiden internet

GPIO General Purpose Input/Output

MQTT MQ Telemetry Transport

1 Johdanto

Olosuhdeseuranta on tärkeää esimerkiksi sisäilman laadun ja viihtyvyyden kannalta. Yksityiskäytössä liian kuuma sisäilma ei ole käyttäjälle mukavaa ja se ei ole energiatehokasta. Yritykselle olosuhdeseuranta on tärkeää yrityksen tuotteiden säilyvyyden ja laadun kannalta. Esimerkiksi kylmätilojen pitää pysyä kylmänä ja valolle arkojen tuotteiden on pysyttävä tarpeeksi pimeässä.

Opinnäytetyön tavoitteena on luoda sisätilan olosuhdeseurantaan laitteisto langattomia XBee-radioita ja Raspberry Pi:tä käyttäen. Mittauskohteina ovat lämpötila ja valoisuus. Mittausdata kerätään radioihin kiinnitetyillä antureilla, tämän jälkeen data lähetetään Raspberry Pi:hin ja data käsitellään Raspberry Pi:ssä Node-RED:iä käyttäen.

Opinnäytetyön toisessa luvussa avataan laitteistossa käytetyt teknologiat ja kerrotaan niiden taustoja. Kolmannessa osiossa avataan IoT-salkku konsepti ja kerrotaan mitä siihen vaaditaan. Neljännessä luvussa kerrotaan, mitä osia ja ohjelmistoja laitteiston kasaukseen vaaditaan. Viidennessä luvussa esitetään tulokset. Lopuksi on pohdinta.

2 Käytetyt teknologiat

2.1 XBee-radio ja ZigBee-protokolla

ZigBee-protokolla on ZigBee Alliance:n kehittämä langattoman kommunikoinnin standardi, joka on rakennettu IEEE 802.15.4 -standardin päälle. ZigBee laajentaa standardia lisäämällä siinä jo olevien fyysisen kerroksen ja MAC-alikerroksen päälle verkkokerroksen ja sovelluskerroksen kehikon. Sovelluskerroksen kehikko koostuu sovellustuki-alikerroksesta ja ZigBee Device Objects -osasta. [1, 1.]

Protokolla on yksi suosituimmista älykotiratkaisuissa. Suosion ansiosta sille on tekninen tuki monissa isoissa älykotituotteissa, joita ovat esimerkiksi Amazon Echo Plus, Samsung SmartThings ja Philips Hue. [2.]

XBee on Digi International -yhtiön tavaramerkki pienille radiomoduuleille, jotka käyttävät ZigBee-protokollaa. Radiot voivat toimia yksinäisinä yksikköinä tai ne voidaan liittää esimerkiksi Arduino-mikrokontrolleriin. Radiot sisältävät I/O-pinnejä, joihin voi liittää sensoreita.

2.2 Raspberry Pi

Raspberry Pi on edullinen ja pienikokoinen yhden piirilevyn tietokone, jonka voi kytkeä näyttöpäätteeseen tai televisioon. Lisäksi siihen voi kytkeä näppäimistön ja hiiren. Raspberry Pi kykenee samaan kuin perinteinen pöytätietokone, tosin pienemmällä teholla: sillä voi selata internetiä, tehdä taulukoita, käsitellä tekstiä tai vaikka pelata. [3.]

Raspberry Pi soveltuu erityisen hyvin IoT-ratkaisuihin, sillä siinä on yleiskäyttöisiä pinnejä (engl. *GPIO*). Nämä pinnit voi ohjelmoida joko lukemaan tai lähettämään signaaleja. [4.]

2.3 MQTT

MQTT on asiakasohjelman ja palvelimen välisen julkaisu/tilaus viestinnän siirtoon tehty protokolla. Protokolla on kevyt, avoin, yksinkertainen ja helppo toteuttaa. Protokolla on hyvä erityisesti laitteiden välisessä viestinnässä sekä IoT-ratkaisuissa sen matalan energiankulutuksen vuoksi. [5.]

MQTT-protokollan käyttö vaatii välittäjäpalvelimen (engl. *broker*), joka hoitaa viestinnän ja mahdollisen salauksen. Opinnäytetyössä käytettäväksi ohjelmaksi valikoitui avoimen lähdekoodin Eclipse Mosquitto.

2.4 Node-RED

Node-RED on flow-pohjainen ohjelmointityökalu, joka perustuu Node.js -ajoympäristöön ja graafiseen selainpohjaiseen editoriin. Selaimessa ajettavassa editorissa tarvittavia koodipalikoita (engl. *node*) vedetään työtilaan, ja niitä voidaan yhdistää toisiinsa. Alun perin MQTT-viestien käsittelyä varten tehty työkalu laajeni myöhemmin yleiskäyttöiseksi. [6.]

3 IoT-salkku

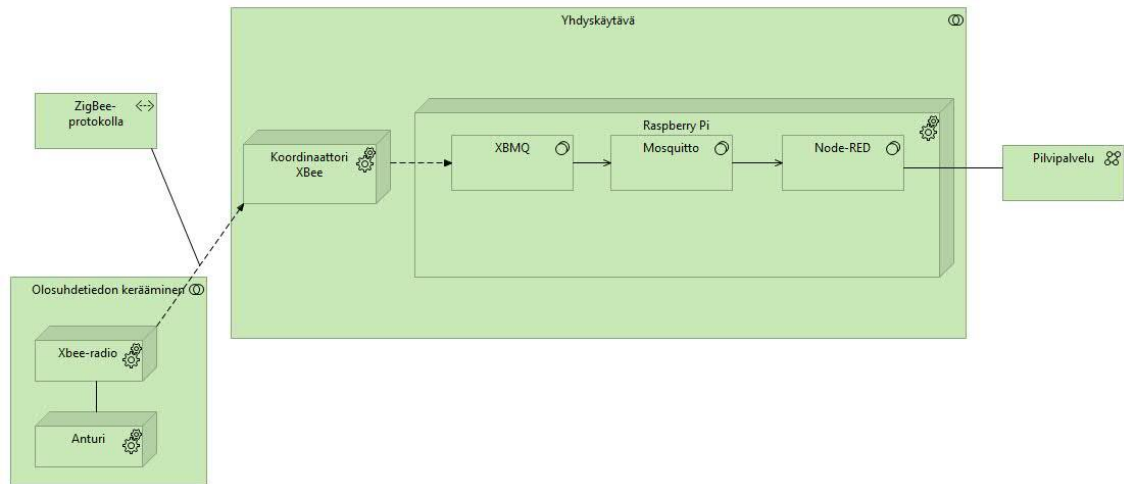
3.1 Konsepti

IoT-salkkukonsepti tarkoittaa kokonaisuutta, jossa fyysiset komponentit ovat valmiiksi kasattuja tai pientä kasausta vaativia. Lisäksi tarvittavat ohjelmistot ovat käyttöä varten valmiiksi asennettuja, ja niiden asetukset noudattavat vaadittuja spesifikaatioita.

Opinnäytetyössä toteutettu kokonaisuus on yksinkertaistettu esimerkki, jolla voi seurata lämpötilaa ja valoisuutta kahdesta kohteesta. Kaikkia ominaisuuksia ei ole toteutettu.

3.2 Fyysinen sisältö

Salkun fyysiseen sisältöön kuuluu Raspberry Pi -yhdyskäytävälaite, jossa on kiinni XBee-radio, joka toimii koordinaattorina. Lisäksi salkkuun sisältyy tässä tapauksessa kaksi muuta XBee-radiota, joissa on kiinni antureita. Antureina toimivat TMP36 lämpötila-anturi ja 10k ohmin LDR-valoisuusanturi. Radioissa kiinni olevat anturit keräävät dataa, ja radiot lähettävät datan koordinaattoriradioon. Yhdyskäytävälaite, jossa koordinaattori on kiinni, käsittelee datan ja tilanteesta riippuen lähettää sen eteenpäin. Kuvassa 1 näkyy konseptin fyysinen rakenne.



Kuva 1. Fyysinen rakenne, käytetty Archimate-mallinnuskieltä.

3.2.1 XBee

Opinnäytetyössä käytettävät XBee-radiot ovat DIGI International -yhtiön kehittämiä ja ne käyttävät ZigBee-protokollaa. Yhtiöltä on saatavissa myös muita XBee-laitteita, jotka käyttävät Euroopassa 868 MHz ja Amerikassa 900 MHz taajuutta. Lisäksi on saatavilla radioita, jotka käyttävät matkapuhelinverkkoa. [7.]

XBee-radiot ovat siitä käteviä sensoriverkkoa rakentaessa, että niiden lisäksi ei välttämättä tarvitse erillistä mikrotietokonetta, sillä niissä on analogisia ja digitaalisia IO-pinnejä. Huomioitavaa on, että digitaaliset IO-pinnit lukevat ja kirjoittavat vain HIGH- ja LOW-arvoja, joten sensorit, jotka antavat ulos digitaalista dataa, eivät toimi ilman välimuuntimia. [8, 103-104.]

XBee Series 2 -radiot, joilla opinnäytetyö tehtiin, tunnetaan nimellä ZNet 2.5, ja ne ovat edellisen sukupolven radioita, jotka käyttävät ZigBee-protokollaa. Suositeltavaa onkin jatkossa tehdä uudella kannalla esimerkiksi Digi XBee 3 ZigBee 3 -radioilla. [7.]

XBee-verkkoa asennettaessa jokaiseen laitteeseen tulee asentaa oikea firmware riippuen, missä roolissa laite tulee olemaan verkossa. Koordinaattorille (Coordinator), reitittimelle (router) ja päätelaitteelle (end device) on olemassa omat firmwarensa. [8, 35.]

XBee mesh -verkko tarvitsee aina yhden koordinaattorin, joka luo verkon ja mahdollistaa muiden XBee-laitteiden liittymisen verkkoon. Koordinaattoria ei voi pistää virransäästötilaan ja sen on hyvä saada virtaa vakaasta lähteestä. [8, 35.]

Riippuen verkon koosta siinä voi olla myös reitittäjiä, jotka toimivat koordinaattorin tavoin, mutta eivät itse luo verkkoa vaan liittyvät siihen ja mahdollistavat muiden reitittimien ja loppulaitteiden liittymisen [8, 35]. Reitittimen firmwaresta riippuu, voiko se itse kerätä sensoridataa vai toimiiko se vain datan reitittimenä.

Kolmas laiteluokka mesh-verkossa on päätelaitteet, jotka lähettävät ja vastaanottavat tietoa reitittimen tai koordinaattorin kautta, johon ne ovat yhteydessä. Nämä loppulaitteet voivat mennä virransäästötilaan, ja niiden ei tarvitse olla verkkovirrassa. [8, 35.]

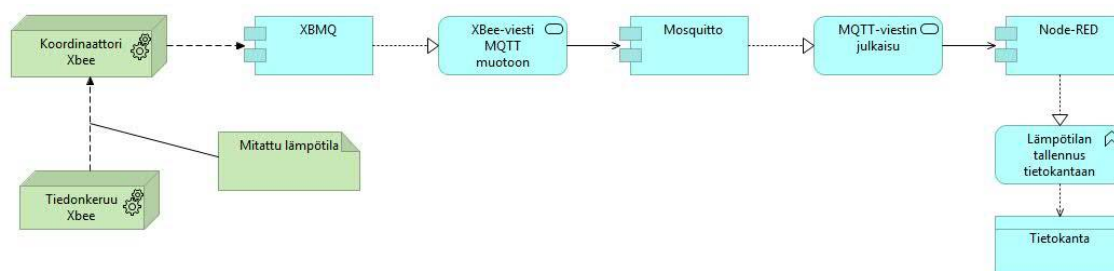
3.2.2 Raspberry Pi

Raspberry Pi toimii kokonaisuuden yhdyskäytävälaitteena, jossa sensoreilta saatavaa tietoa voidaan käsitellä ja lähettää eteenpäin. Raspberry Pi:ssä on kiinni koordinaattori XBee, joka vastaanottaa tiedonkeruun XBee-radioiden mitaaman ja lähettämän sensoritiedon, ja tässä ratkaisussa julkaisee tiedot MQTT-protokollaa ja Node-RED:iä käyttäen.

Ideaalitilanteessa kerätty tieto lähetetään eteenpäin pilvipalveluun, jossa se on helposti käyttäjän nähtävissä ja käytettävissä. Pilvipalvelu mahdollistaa tietojen käytön mistä tahansa ja ei näin ollen sido tietoja paikalliseen laitteeseen.

3.3 Looginen rakenne

Looginen rakenne koostuu tiedonkeruun XBee-radioiden keräämästä ja lähettämästä olosuhdetiedosta, ja tiedonkäsittelyprosessista Raspberry Pi:ssä. Tiedon siirtymiseen ja käsittelyyn tarvitaan useita ohjelmia: 1) XBeeMQ, 2) Eclipse Mosquitto ja 3) Node-RED. Kuvassa 2 näkyy looginen rakenne komponenttimuodossa.



Kuva 2. Looginen komponenttirakenne, käytetty Archimate-mallinnuskieltä.

XBMQ on ohjelma, joka mahdollistaa Xbee-radioiden välisen kommunikoinnin MQTT-protokollaa käyttäen. Radioiden keräämä anturidata voidaan julkaista radion omaan otsikkoon, ja tarvittaessa radioille voidaan myös lähettää viestejä ohjelmaa käyttäen. XBMQ on saatavissa Java- tai Node.js-versiona [9]. Tässä tapauksessa käytettiin Java-versiota.

Eclipse Mosquitto on kevyt, avoimen lähdekoodin viestinvälityspalvelin, joka käyttää MQTT-protokollaa [10]. Mosquitto toimii taustalla kaikkien MQTT-viestien käsittelijänä. Se vastaanottaa asiakkaiden (engl. *client*) julkaisemat viestit, tarkastaa kuka on tilannut kyseisen viestin ja välittää viestin oikealle vastaanottajalle. Lisäksi ohjelma säilöo istuntotiedot ja käsittelee asiakkaiden todentamisen ja valtuuttamisen. [11.]

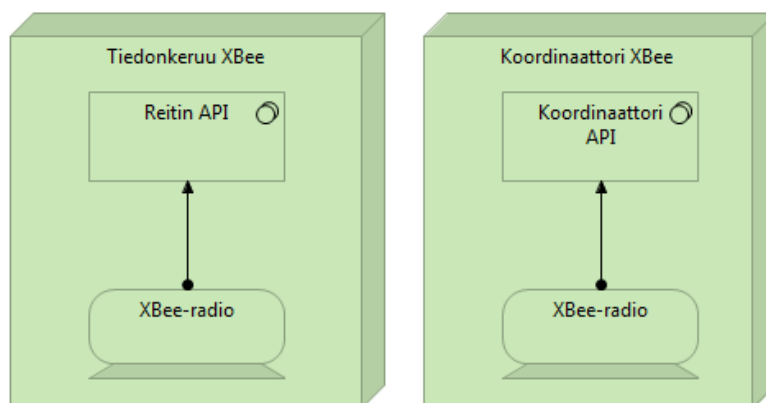
Node-RED on kokonaisuuden aivot, sillä sitä käyttämällä kaikki toiminnallisuus toteutetaan. Node-RED ohjelmoidaan tilaamaan sensoriradioiden MQTT-viestit. Tarvittaessa raakadata muunnetaan luettavaan muotoon. Sen jälkeen radioiden lähettämä tilatieto tallennetaan tietokantaan. Haluttaessa esimerkiksi lämpötilalle voidaan ohjelmoida funktio, joka lähettää varoituksen lämpötilan noustessa liian korkeaksi.

4 Olosuhdeseurantalaitteiston asennus ja toiminta

Demokokoonpano olosuhdeseurantaan rakennettiin vuonna 2017 ohjelmointilaboratoriossa. Komponentit, joilla kokoonpano tehtiin, löytyvät liitteestä 1. Rakennetulla kokonaisuudella seurattiin ohjelmointilaboratorion lämpötilaa ja valoisuutta. Lämpötilaa seurattiin noin vuorokauden ajan, ja näytteenottotiheys oli näyte sekunnissa.

Työn yhdyskäytävälaitteena käytettyyn Raspberry Pi:hin asennettiin Raspbian-käyttöjärjestelmä. Sen jälkeen asennettiin Eclipse Mosquitto, joka toimi MQTT-välittäjänä. Mosquitton toiminta testattiin Chrome-selaimeen asennetulla MQTTLens-nimisellä laajenuksella. Mosquitton asentamisen ja testauksen jälkeen ladattiin ja asennettiin XBMQ. XBMQ-ohjelman latauslähteenä käytettiin GitHub:ia [9], josta löytyi myös asennus- ja konfiguraatio-ohjeet. Kolmantena tarvittavana ohjelmana ladattiin ja asennettiin Node-RED, jonka ohjeet löytyivät lähteestä [12].

Työssä tarvittiin kolme XBee-radiota, jotka piti ohjelmoida ennen käyttöä. Ohjelmointiin käytettiin Digi-yhtiön XCTU-ohjelmaa. Lisäksi tarvittiin XBee Explorer Dongle, jolla radiomoduuli liitettiin tietokoneeseen ohjelmointia varten. Radioihin asennettiin oikeat firmwaret, yksi koordinaattori API-firmware ja kaksi reititin API-firmwarea. Radioille piti antaa sama PAN ID. Lisäksi radioiden uniikit MAC-osoitteet piti ottaa talteen, sillä niistä muodostui MQTT-otsikot. Kahteen reititin-radioon liitettiin mittausanturit. Ensimmäiseen radioon liitettiin lämpötila-anturi TMP36 ja toiseen valoisuusanturi. Kuvassa 3 näkyy radioiden firmwaret.



Kuva 3. XBee-radioiden firmwaret.

Koordinaattoriradio liitettiin Raspberry Pi:hin edellä mainitulla Donglella ja se sai virtansa Raspberry Pi:stä. Kaksi muuta radiota toimivat olosuhdeseuraajina ja ne mittasivat tilan valoisuutta ja lämpötilaa. Virtansa ne saivat paristoista ja olivat näin liikuteltavissa. Radiot sijoitettiin tilan hyllyyn.

Kun radiot olivat valmiit, ohjelmoitiin Node-RED käsittelemään radioiden lähettämä data. Lämpötilaa mittaavan radion data tallennettiin tekstitiedostoon ja sille

tehtiin myös funktio, joka tarkkaili lämpötilaa. Lämpötilan noustessa liian korkeaksi, tässä tapauksessa 30 °C, syttyi LED-valo demonstroiden varoitusta.

5 Tulokset

Opinnäytetyön lähtökohtana oli rakentaa laitteisto, jolla pystyttiin toteuttamaan olosuhdeseurantaa. Fyysisessä laitteiston rakentamisessa käytettiin saatavilla olevia XBee-radioita ja Raspberry Pi -mikrotietokonetta.

Olosuhdeseurannassa onnistuttiin mittaamaan lämpötilaa ja valoisuutta. Lämpötila tallentui onnistuneesti paikalliseen tekstitiedostoon; oikeassa toteutuksessa tieto olisi hyvä tallentaa joko paikalliseen tietokantaan tai pilveen. Valoisuutta pystyttiin lukemaan, mutta sitä ei tallennettu. Myös varoitusfunktio eli LED-valon syttyminen toimi lämpötilan noustessa yli 30 °C:seen.

6 Pohdinta

Työn tavoite eli olosuhdeseuranta lämpötilan ja valoisuuden suhteen onnistui, mutta työn skaala oli liian pieni. Seurattuja kohteita ja mitattuja olosuhdetekijöitä olisi voinut olla enemmän. Esimerkiksi kosteutta olisi voitu mitata, mutta se olisi vaatinut pelkän radion lisäksi myös välimuuntimen, kuten Arduinon, lukemaan sensorin antaman datan ja muuttamaan sen radion tukemaan muotoon. Lisäksi mittausaikaväli oli liian lyhyt ja näytetiheys liian suuri. Sopiva mittausväli olisi voinut olla esimerkiksi viikko ja näytetiheys näyte/tunti. Tässä työssä tulokset tallennettiin tietokannan sijaan tekstimuodossa, mikä on epäkäytännöllistä. Työn vaiheiden tarkka dokumentointi ja kuvien ottaminen olisi myös ollut hyvä idea.

Jatkokehityksenä voitaisiin luoda käyttöliittymä, jossa kerätty tieto olisi helpommin luettavissa, esimerkiksi kuvaajina. Mitattavasta kohteesta riippuen erilaisten mittausantureiden lisääminen ei myöskään olisi huono idea. Mikäli olosuhdeseurantaa tehdään isommassa mittasuhteessa, pitää ottaa huomioon radioiden kantama ja tarvittaessa lisätä reititinradioita ylläpitämään verkkoa.

Jos työ haluttaisiin tuotteistaa kaupalliseen käyttöön, tulisi selvittää onko tuotteelle kysyntää ja onko sillä kilpailijoita. Myös tietoturvaan tulisi kiinnittää erityistä huomiota. Lisäksi tuotteen ulkonäkö on tärkeä osa kaupallista käyttöä ja näin koteloinnin suunnittelu ja muutenkin tuotteen näytävyyden suunnitteluun tulisi kiinnittää huomiota.

Opinnäytetyön kirjoittamiseen meni pitkä aika, sillä koeasetelma jäi liian pieneksi ja näin ollen tulokset jäivät vähänlaisiksi. Tämä laski motivaation kirjoittamiseen ja siitä oli vaikea saada kiinni. Koen kuitenkin kasvaneeni työn tekemisestä ja vaikeasta kirjoitusprosessista.

Lähteet

1. ZigBee Alliance. ZigBee Specification. 2015. [Viitattu 27.4.2020.] Saatavissa: <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf>
2. Zigbee Alliance. Smart Home. 2020. [Viitattu 31.3.2020.] Saatavissa: <https://zigbeealliance.org/market-uses/smart-home/>
3. Raspberry Pi. 2019. [Viitattu 13.5.2019.] Saatavissa: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
4. Raspberry Pi. 2019. [Viitattu 13.5.2019.] Saatavissa: <https://www.raspberrypi.org/documentation/usage/gpio/>
5. Andrew Banks, Ed Briggs, Ken Borgendale, Rahul Gupta. MQTT Version 5.0. 2019. [Viitattu 13.5.2019.] Saatavissa: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
6. Node-Red. 2019. [Viitattu 14.5.2019.] Saatavissa: <https://nodered.org/about/>
7. Digi.com. Tuotteet. 2020. [Viitattu 27.4.2020.] Saatavissa: <https://www.digi.com/products/embedded-systems/digi-xbee>
8. Digi.com. ZigBee RF Modules, User Guide. 2016. Päivitetty toukokuu 2018. [Viitattu 19.5.2017.] Saatavissa: <https://www.digi.com/resources/documentation/digidocs/pdfs/90000976.pdf>
9. Bythell, A. XBMQ-java. 2015. Päivitetty 7.5.2019. [Viitattu 15.4.2020.] Saatavissa: <https://github.com/angryelectron/xbmq-java>
10. R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," The Journal of Open Source Software, vol. 2, no. 13, May 2017. [Viitattu 16.4.2020.] Saatavissa: <https://doi.org/10.21105/joss.00265>
11. HiveMQ tiimi. Client, Broker / Server and Connection Establishment - MQTT Essentials: Part 3. Julkaistu 17.7.2019. [Viitattu 17.4.2020.] Saatavissa: <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>
12. Node-RED. 2020. [Viitattu 21.4.2020.] Saatavissa: <https://nodered.org/docs/getting-started/raspberrypi>

Osaluettelo fyysisistä komponenteista

1. Raspberry Pi 3 Model B
2. Muistikortti, Kingston 8GB
3. 3 kpl XBee Series 2 -radioita
4. Lämpötila-anturi TMP36
5. 0.1 μ F kondensaattori lämpötila-anturin häiriönpoistoon
6. Valoisuusanturi, 10k ohm LDR
7. Vastus, 20k ohm
8. Usb Explorer Dongle
9. Led, punainen
10. Koekytkentälevy, 3 kpl
11. Koekytkentäjohtoja, noin 30kpl
12. Virtalähde Raspberry Pi
13. Paristokotelo kahdelle AA-paristolle x2
14. 4 AA-paristoa