



Kesämökin ilmalämpöpumpun etäohjaus ja -seuranta Node-RED- ohjelmointityökalun avulla

Ammattikorkeakoulututkinnon opinnäytetyö
Tieto- ja viestintätekniikka, insinööri (AMK)
Kevät, 2024
Hanna Sinkkilä

Tieto- ja viestintäteknikka

Tekijä Hanna Sinkkilä

Työn nimi Kesämökin ilmalämpöpumpun etäohjaus ja -seuranta Node-RED-ohjelmointityökalun avulla

Ohjaaja Timo Karppinen

Tiivistelmä

Vuosi 2024

Tämän toiminnallisen opinnäytetyön tarkoituksena oli perehtyä ilmalämpöpumpun etäohjausmahdollisuuksiin ja toteuttaa Node-RED-ohjelmointityökalun avulla kesämökkiin helposti laajennettavissa oleva kotiautomaatiojärjestelmä, jolla voidaan etäohjata ilmalämpöpumppua ja seurata rakennuksen lämpötilaa. Tavoitteena oli helpottaa yksityiskäytössä olevan kesämökin sisälämpötilan ohjausta ja seurantaa, sekä mahdollistaa tämän avulla säästöjä.

Opinnäytetyössä perehdyttiin ilmalämpöpumpun etäohjaimiin, vertailtiin niitä keskenään ja valittiin niistä tähän kohteeseen sopivin. Etäohjaimia löytyi useita erilaisia, ja näistä valittiin käytettäväksi laitteeksi Broadlink RM PRO. Tietoperustassa esiteltiin toteutukseen tarvittavat ohjelmistot, laitteet ja viestintäprotokollat. Käytettäväksi alustaksi valittiin Raspberry Pi Zero 2 W -minitietokone, johon asennettiin Node-RED-ohjelmointityökalu. Sisäilman lämpötilaa mittaavina laitteina käytettiin RuuviTag-antureita. Näistä lähetettiin data Node-RED:lle MQTT-sanomilla.

Toteutusosiossa kuvattiin Raspberry Pi:n, Node-RED-ohjelmointityökalun, RuuviTag-antureiden ja Broadlink RM Pro -infrapunaohjaimen asennus. Lisäksi kuvattiin Node-RED:n Dashboard-, Remote-RED- ja WhatsApp-lisäosien asennukset. Dashboard-lisäosan avulla luotiin helppokäyttöiset näkymät, joilla käyttäjä voi tarkastella antureiden dataa ja ohjata ilmalämpöpumppua. Remote-RED-lisäosa mahdollisti dashboard-näkymään pääsyn etänä ja WhatsApp-lisäosan avulla luotiin puhelimeen tulevat hälytykset rakennuksen liian alhaisista lämpötiloista.

Opinnäytetyön tuotoksena valmistui tavoitteiden mukainen järjestelmä. Node-RED todettiin toimivaksi ohjelmointityökaluksi ilmalämpöpumpun etäohjaukseen varsinkin tilanteessa, jossa etäohjauksella säädetään vain ilmalämpöpumpun toimintotilaa ja lämpötiloja.

Avainsanat Etäohjaus, ilmalämpöpumppu, kotiautomaatio, Node-RED

Sivut 38 sivua ja liitteitä 10 sivua

The purpose of this functional thesis was to explore the remote control capabilities of an air source heat pump and implement a home automation system for a summer cottage using the Node-RED programming tool. This system, which can be easily expanded, allows for remote control of the air source heat pump and monitoring of the building's temperature. The objective was to simplify the control and monitoring of indoor temperature in a privately used summer cottage, ultimately leading to cost savings.

This thesis involved researching and comparing of the various remote control options for air heat pumps and selecting a most suitable one for this project. Among several options, the Broadlink RM Pro was chosen. The necessary software, devices and communication protocols were outlined. The Raspberry Pi Zero 2 W mini-computer was selected as the platform, with Node-RED installed for programming. RuuviTag sensors were utilized for indoor air temperature measurements, with data transmitted to Node-RED via MQTT messages.

The implantation section detailed the setup process for the Raspberry Pi, Node-RED, RuuviTag sensors and Broadlink RM PRO infrared controller. Additionally, installations of Node-RED's Dashboard, Remote-RED and WhatsApp add-ons were explained. The Dashboard add-on facilitated user-friendly views for sensor data visualization and air source heat pump control. Remote-RED enabled remote access to the dashboard, while the WhatsApp add-on was employed for sending temperature alerts to mobile phones.

The thesis resulted in the completion of a system aligned with the defined objectives. Node-RED proved to be an effective programming tool for remote controlling of the air heat pump, particularly in scenarios where the remote-control functionality is limited to adjusting mode and temperature settings.

Keywords Air source heat pump, home automation, Node-RED, remote control

Pages 38 pages and appendices 10 pages

Sisällys

1	Johdanto	1
2	Ilmalämpöpumpun etäohjaus	2
2.1	Ilmalämpöpumppu.....	2
2.2	Ilmalämpöpumpun vikatilanteet	2
2.3	Kohteen ilmalämpöpumppu	4
2.4	Ilmalämpöpumpun etäohjaimet.....	4
2.4.1	Panasonic CZ-TACG1	4
2.4.2	AirPatrol WiFi V5 E	5
2.4.3	Sensibo Sky	6
2.4.4	Nedis SmartLife yleiskaukosäädin	7
2.4.5	Broadlink RM Pro	7
2.5	Etäohjainten vertailu	8
3	Toteutukseen tarvittavat ohjelmistot, laitteet ja viestintäprotokollat	10
3.1	Node-RED	10
3.2	Raspberry Pi	12
3.3	RuuviTag	12
3.4	MQTT	13
4	Etäohjauksen ja -seurannan toteutus	15
4.1	Suunnitelma	15
4.2	Raspberry Pi:n asennus.....	15
4.3	Node-RED-ohjelmointityökalun asennus.....	17
4.4	RuuviTag-anturien asennus.....	19
4.5	Broadlink RM PRO -infrapunaohjaimen asennus	22
4.5.1	e-Control-sovelluksen asennus.....	22
4.5.2	Etäohjaimen asennus Node-RED:iin	22
4.5.3	Kaukosäätimen komentojen opettelu ja tallentaminen.....	24
4.5.4	Komentojen lähettäminen ilmalämpöpumpulle.....	26
4.6	Dashboard-lisäosan asennus	26
4.7	Remote-RED-lisäosan asennus	31
4.8	Hälytysten asetus.....	32
4.9	Valmis järjestelmä	33
5	Pohdinta.....	36
	Lähteet.....	39

Kuvat ja taulukot

Kuva 1. Panasonic CZ-TACG1 (Panasonic, n.d.).....	5
Kuva 2. AirPatrol WiFi V5 E (AirPatrol, n.d.).....	6
Kuva 3. Sensibo Sky (Sensibo, n.d.)	6
Kuva 4. Nedis SmartLife yleiskaukosäädin (Nedis, n.d.).....	7
Kuva 5. BroadLink RM Pro (BroadLink Finland, n.d.).....	8
Kuva 6. Node-RED-ohjelmointityökalun työtila (OpenJS Foundation, n.d.-b).	10
Kuva 7. Raspberry Pi Zero 2 W (Raspberry Pi Foundation, n.d.)	12
Kuva 8. RuuviTag anturi (Ruuvi Innovations Oy, n.d.-a)	13
Kuva 9. MQTT-protokollan arkkitehtuuri (HiveMQ, 2024).	14
Kuva 10. Raspberry Pi Imager-sovellus.....	16
Kuva 11. Node-RED:n asetusten asettaminen.....	17
Kuva 12. Node-RED:n käyttöliittymän päänäköymä.....	18
Kuva 13. RuuviTagien tuloste.	19
Kuva 14. Node-RED ensimmäisten nodejen lisäyksen jälkeen.	21
Kuva 15. e-Control sovelluksen päänäköymä.	22
Kuva 16. Node-RED:n päävalikko.....	23
Kuva 17. Flow-kaavio Broadlink-laitteen tietojen hakemiseksi.....	23
Kuva 18. Kaukosäätimen komennon opettelusta saatu tuloste.....	24

Kuva 19. Kaukosäätimen komentoja tallentava flow-kaavio.	25
Kuva 20. Raspberry Pi:n levyille tallennetut komennot.	26
Kuva 21. Ilmalämpöpumpulle komentoja lähettävä flow-kaavio.....	26
Kuva 22. Ilmalämpöpumpulle lämmityskomentoja lähettävä flow-kaavio.	28
Kuva 23. Lämmityksen dashboard-näkymä.....	28
Kuva 24. Keittiön RuuviTag-anturin mittaustulokset dashboardilla esittävä flow-kaavio.	30
Kuva 25. Node-RED:n sivupalkin valikko.....	31
Kuva 26. Keittiön RuuviTag-anturin mittaustulokset dashboardilla esittävä flow-kaavio, johon on lisätty hälytystoiminto.	33
Kuva 27. Hälytystoiminnon testauksessa saadut WhatsApp-viestit.....	33
Kuva 28. Lämmitys- ja jäähdytys-näkymät Remote-RED-sovelluksessa.	34
Kuva 29. Mittaukset-näkymä Remote-RED sovelluksessa.	35
Taulukko 1. Vikatilanteiden laadullinen arviointi.	3
Taulukko 2. Ilmalämpöpumpun etäohjainten vertailu.	9

Liitteet

- Liite 1. Tiedoston getAndSendRuuviDatas.py sisältö.
- Liite 2. JavaScript ohjelmakoodi etäohjaimelle lähetettävän sanoman luomiseen.
- Liite 3. JavaScript ohjelmakoodi tiedoston polun luomiseen.
- Liite 4. JavaScript ohjelmakoodi aikaleiman luomiseen.
- Liite 5. JavaScript ohjelmakoodi JavaScript-objektina saatujen datojen jakamiseen omiin ulostuloihin.

- Liite 6. JavaScript ohjelmakoodi RuuviTag-anturin pariston varaustason laskemiseen.
- Liite 7. JavaScript ohjelmakoodi aikaleiman sekä lämpötila-arvon ja aikaleiman luomiseen.
- Liite 8. JavaScript ohjelmakoodi aikaleiman, lämpötila-arvon ja aikaleiman sekä hälytysten luomiseen.
- Liite 9. Kuvat valmiista Node-RED:n flow-kaavioista.

1 Johdanto

Tämän toiminnallisen opinnäytetyön tarkoituksena on perehtyä ilmalämpöpumpun etäohjausmahdollisuuksiin ja toteuttaa Node-RED-ohjelmointityökalun avulla kesämökkiin helposti laajennettavissa oleva kotiautomaatiojärjestelmä, jolla voidaan etäohjata ilmalämpöpumppua ja seurata rakennuksen lämpötilaa. Tavoitteena on helpottaa yksityiskäytössä olevan kesämökin sisälämpötilan ohjausta ja seurantaa sekä mahdollistaa tämän avulla säästöjä.

Idea opinnäytetyön aiheesta tuli omasta tarpeesta. Kohteena oleva kesämökki on 60-luvulla rakennettu hirsimökki, jota on aiemmin pidetty talvisin kylmillään. Keväisin ja syksyisin lämmityksen apumuotona on takan lisäksi käytetty sähköpattereita. Rakennukseen tehtiin muutama vuosi sitten mittava remontti, jonka yhteydessä uusittiin kaikki kalusteet ja asennettiin ilmalämpöpumppu sekä veden läpivirtauslämmitin. Tästä lähtien mökissä on pidetty talvisin ylläpitolämpöä.

Ilmalämpöpumpussa ei ole aiemmin ollut etäohjausta eikä mökin sisälämpötilaa ole myöskään ollut mahdollista seurata etänä. Edellistalvena oli poikkeuksellisen pitkiä kylmiä pakkasjaksoja, jolloin mökin sisälämpötila pääsi ilmalämpöpumpusta huolimatta laskemaan pakkasen puolelle aiheuttaen vikaantumisia kodinkoneisiin. Tämän seurauksena kohteeseen päätettiin hankkia järjestelmä etäohjausta ja -seurantaa varten. Järjestelmän haluttiin olevan luotettava, edullinen, helppokäyttöinen sekä helposti laajennettavissa oleva esimerkiksi etäohjattavilla lampuilla, pistorasioilla ja kameroilla.

Tutkimuskysymykset, joihin opinnäytetyössä haetaan vastauksia ovat:

1. Millaisia etäohjaimia ilmalämpöpumpulle on?
2. Miten Node-RED soveltuu ilmalämpöpumpun etäohjaukseen?

Opinnäytetyössä tutkitaan erilaisia ilmalämpöpumpun etäohjaimia ja valitaan näistä sopivin. Opinnäytetyön tuotoksena toteutetaan yksityiskohtaisesti kuvattuna etäohjaimen ja Node-RED:n avulla vapaa-ajan asuntoon kotiautomaatiojärjestelmä, jolla voidaan etäohjata ilmalämpöpumppua. Lisäksi automaatiojärjestelmään toteutetaan lämpötilan seuranta käyttäen RuuviTag-antureita.

2 Ilmalämpöpumpun etäohjaus

2.1 Ilmalämpöpumppu

Ilmalämpöpumppu on kaikkiin talotyyppeihin sopiva lämmityslaitte, joka koostuu yhdestä ulkoyksiköstä ja vähintään yhdestä sisäyksiköstä. Lämmityskäytössä laitteen toiminta perustuu ulkoyksikön talteen ottamaan lämpöön, jota saadaan kierrättämällä ulkoilmaa ja jäähdyttämällä se. Tämä talteen otettu lämpö siirretään sisäyksikköön ja luovutetaan huoneilmaan. Yksi sisäyksikkö kykenee levittämään lämpöä tavallisesti noin 30–100 neliömetrin alueelle. (Motiva, 2022)

Pääsääntöisesti ilmalämpöpumppua ei suositella rakennuksen ainoaksi lämmitysmuodoksi, sillä sen tehontuotto ja hyötysuhde heikkenevät ulkolämpötilan laskiessa. Kovimmat pakkaset saattavat laskea ilmalämpöpumpun hyötysuhteen lähes sähkölämmityksen tasolle. Leutona talvena yksittäinen ilmalämpöpumppu saattaa riittää tuottamaan rakennuksen tarvitseman lämmön, mutta sen rinnalle tulisi hankkia erillinen varalämmitysmuoto, joka kytkeytyy päälle termostaatin ohjaamana. (Motiva, 2022)

2.2 Ilmalämpöpumpun vikatilanteet

Ilmalämpöpumppu on mekaaninen laite, jonka toimintaan voivat vaikuttaa useat erilaiset vikatilanteet. Yleisimpiä ongelmia ilmalämpöpumpun toiminnassa ovat huono lämmitysteho, jäänyt höyrytimen patteri, ilmalämpöpumpun kova ääni, sähköongelmat sekä ilmalämpöpumpun huono ilmavirtaus. (Energy Center Finland Oy, n.d.-b)

Ilmalämpöpumpun lämmitystehon heikkenemisen mahdollisia syitä ovat likainen ilmansuodatin, kylmäainevuoto ja höyrytimen patterin jäätyminen. Ilmansuodattimet on mahdollista vaihtaa itse, mutta kylmäaineen lisäys vaatii ammattilaisen. Jäänyt höyrytin on mahdollista saada sulatettua sammuttamalla ilmalämpöpumppu ja antamalla sen sulaa ennen uudelleenkäynnistystä. (Energy Center Finland Oy, n.d.-b)

Ulkoyksikköön saattaa kertyä irtonaisia osia tai roskia, jotka voivat vaikuttaa laitteen pitämään ääneen tai suorituskykyyn. Ongelma voidaan korjata sammuttamalla laite ja poistamalla mahdolliset roskat tai kiristämällä löysät osat. Suorituskykyyn voivat vaikuttaa myös viat sähkökomponenteissa, kuten termostaatissa tai katkaisijassa. Nämä ongelmat vaativat ammattilaisen korjausta. (Energy Center Finland Oy, n.d.-b)

Ilmalämpöpumppu muodostaa kondenssivettä sisäyksikköön viilentäessä ja ulkoyksikköön lämmittäessä. Nämä ovat normaaleja toimintoja, jotka liittyvät ilmalämpöpumpun toimintaan. Kondenssivesi voi aiheuttaa vuoto-ongelmia ja vesivahingon, mikäli sisäyksikkö on asennettu väärin ja vesi ei pääse vapaasti virtaamaan ulos. Lisäksi ulkoyksikön alle voi kertyä pakkasella kondenssivettä, joka jäätyessään voi vaurioittaa laitetta. (Energy Center Finland Oy, n.d.-a)

Taulukossa 1 on arvioitu mahdollisia kohteen ilmalämpöpumppuun kohdistuvia vikatilanteita. Tilanteet on arvioitu todennäköisyyden ja vaikuttavuuden perusteella. Taulukon värit kuvaavat kokonaisvaikuttavuutta, jossa punainen väri kuvaa vahvinta kokonaisvaikutusta ja vihreä väri heikointa.

Taulukko 1. Vikatilanteiden laadullinen arviointi.

		Todennäköisyys		
		Pieni	Keskinkertainen	Suuri
Vaikutus	Pieni			
	Keskinkertainen			
	Suuri	<ul style="list-style-type: none"> - Kylmäainevuoto - Vialliset sähkökomponentit - Laitteen jäätyminen - Sähkönsyöttöongelma - Roskat ulkoyksikössä - Kaukosäätimen vanhat patterit - Väärä asetus (lämmitys/viilennys/auto) - Viallinen vikavirtasuojia - Ulkoyksikön jäätyminen - Kondenssiveden kertyminen järjestelmään 	- Likainen ilmansuodatin	

2.3 Kohteen ilmalämpöpumppu

Tämän opinnäytetyön kohteena on yksityinen pyöröhirrestä rakennettu 40 neliömetrin kokoinen kesämökki, joka sijaitsee Etelä-Suomessa. Kohde on sähköistetty ja siellä pidetään talvisin ylläpitolämpöä. Asunnossa oleskellaan pääsääntöisesti loppukevästä myöhäiseen syksyyn, joten kesällä on tarve viilennykselle ja muina aikoina lämmitykselle. Kohteeseen on hankittu ilmalämpöpumppu, joka on malliltaan Panasonic CS-QZ9SKE. Ilmalämpöpumppu on noin viisi vuotta vanha, ja siinä ei ole ennestään etäohjausmahdollisuutta.

Ilmalämpöpumppu on kohteen ainoa lämmönlähde takan lisäksi. Kohteesta löytyy langaton verkkoyhteys.

Ilmalämpöpumppua ohjataan kaukosäätimellä, joka lähettää käskyt ilmalämpöpumpulle infrapunasäteilyä hyödyntäen. Kaukosäätimestä voidaan säätää esimerkiksi pumpun toimintotilaa, lämpötilaa, tuulettimen nopeutta sekä pystysuuntaisen ilmavirtauksen kulmaa. Tässä kohteessa ilmalämpöpumpun etäohjauksella on tarpeellista säätää vain toimintotilaa lämmityksen ja viilennyksen välillä sekä lämpötilaa 8 ja 25 asteen välillä. Lisäksi laite tulisi voida sammuttaa.

2.4 Ilmalämpöpumpun etäohjaimet

Verkkohaku hakusanoilla ”ilmalämpöpumpun etäohjaus” tuottaa runsaasti tuloksia, mistä nähdään, että tarpeeseen on saatavilla paljon erilaisia kaupallisia laitteita. Tähän vertailuun päätettiin valita viisi erilaista langattomassa verkossa toimivaa ohjainta, jotka olivat erilaisista hintaluokista ja helposti saatavilla olevia.

2.4.1 Panasonic CZ-TACG1

Panasonic CZ-TACG1 on Panasonicin uusimpien ilmalämpöpumppujen sisäyksikköön asennettavissa oleva lisäosa. Ilmalämpöpumpusta tulee löytyä CN-CNT-liitin, johon moduuli liitetään. Lisäksi kohteesta tulee löytyä langaton verkkoyhteys. Ilmalämpöpumpun ohjaus toteutetaan älypuhelimella tai tabletilla Panasonic Comfort Cloud -sovelluksen kautta.

Laitteen hinta on noin 200 euroa. (Mr.LVI, n.d.)

Laitteen ja sovelluksen avulla voidaan etäohjata ilmalämpöpumppua kaikilla ilmalämpöpumpun kaukosäätimestä löytyvillä toiminnoilla. Ohjaimen ja ilmalämpöpumpun välillä kommunikointi tapahtuu molempiin suuntiin, ja tämän ansiosta ilmalämpöpumpun

antamat vikahälytykset on mahdollista saada sovellukseen. (Mr.LVI, n.d.) Node-RED:sta löytyy lisäosa node-red-contrib-panasonic-comfort-cloud, jonka avulla voidaan yhdistää laite Panasonic Comfort Cloud -pilven kautta Node-RED:iin (OpenJS Foundation, n.d.-d). Laite on esitetty kuvassa 1.

Kuva 1. Panasonic CZ-TACG1 (Panasonic, n.d.).



2.4.2 AirPatrol WiFi V5 E

AirPatrol WiFi on langattomaan verkkoon yhdistettävä ilmastoinnin ja ilmalämpöpumpun etäohjain. Laitteen toiminta on testattu useiden laitevalmistajien kanssa ja siihen on määritelty asetukset valmiiksi siten, ettei laitetta tarvitse erikseen ohjelmoida eri pyynnöille. Käyttöön tarvitaan mobiililaitte ja siihen ladattu AirPatrol-sovellus. (AirPatrol, n.d.)

Etäohjauksen lisäksi laitteeseen on mahdollista säätää ilmoituksia huoneen lämpötilan tai ilmankosteuden mukaan. Lisäksi ilmalämpöpumpun ohjaamiseksi on mahdollista säätää kalenteriin perustuvia toimintoja. Laitteen hinta on noin 120 euroa, ja se on yhteensopiva Amazon Alexan ja Google Homen kanssa. (AirPatrol, n.d.) Laite on esitetty kuvassa 2.

Kuva 2. AirPatrol WiFi V5 E (AirPatrol, n.d.).



2.4.3 Sensibo Sky

Sensibo Sky on pienikokoinen etäohjain, joka toimii kaikkien ilmastointilaitteiden ja ilmalämpöpumppujen kanssa, joita ohjataan kaukosäätimillä. Etäohjauksen lisäksi laitteesta löytyy lämpötila- ja kosteusanturit. Sensibo Sky tarvitsee toimiakseen langattoman verkon, ja sen avulla ilmalämpöpumppua voidaan etäohjata joko mobiilisovelluksella tai selaimella. (Sensibo, n.d.)

Laite tukee Amazon Alexaa ja Google Homea (Sensibo, n.d.) ja sen hinta on noin 100 euroa (Power, n.d.). Sensibo Sky on yhdistettävissä Sensibon pilvipalvelun Sensibo Cloudin kautta Node-RED:iin lisäosalla nimeltä node-red-contrib-sensibo (OpenJS Foundation, n.d.-e). Laite on esitetty kuvassa 3.

Kuva 3. Sensibo Sky (Sensibo, n.d.).



2.4.4 Nedis SmartLife yleiskaukosäädin

Nedis SmartLife yleiskaukosäädin on infrapunakaukosäädin, jolla voidaan etäohjata kauko-ohjattavia laitteita. Se tarvitsee toimiakseen WiFi-verkon ja mobiililaitteen, johon tulee ladata Nedis SmartLife -sovellus. Sovellus on ladattavissa GooglePlay- ja AppStore-kaupoista. (Nedis, n.d.)

SmartLife-sovelluksessa on käytössä suuri tietokanta, joka tukee yli 80 000 laitetta. Tämän lisäksi laitteelle on mahdollista opettaa tuntemattomien infrapunalaitteiden tietoja manuaalisesti. Sovellukseen voidaan ohjelmoida automaatioita esimerkiksi aikataulujen ja olosuhteiden mukaan. Laitteen hinta on noin 25 euroa, ja se on yhteensopiva Amazon Alexan ja Google Homen kanssa. (Nedis, n.d.) Laite on esitetty kuvassa 4.

Kuva 4. Nedis SmartLife yleiskaukosäädin (Nedis, n.d.).



2.4.5 Broadlink RM Pro

Broadlink RM Pro on WiFi-verkossa toimiva yleiskauko-ohjain. Sen avulla voidaan etäohjata infrapuna- tai radiokauko-ohjattavia laitteita, kuten televisiota tai ilmalämpöpumppua. Laitetta ohjataan Broadlink e-Control-mobiilisovelluksella, joka on ladattavissa AppStoresta tai GooglePlay-kaupasta. (BroadLink Finland, n.d.)

Broadlink RM-laitteita varten on kehitetty Node-RED:iin lisäosa node-red-contrib-broadlink-control, jonka avulla laitetta voidaan ohjata tämän kautta (OpenJS Foundation, n.d.-c). Laite maksaa noin 50 euroa (Kärkkäinen, n.d.). Laite on esitetty kuvassa 5.

Kuva 5. BroadLink RM Pro (BroadLink Finland, n.d.).



2.5 Etäohjainten vertailu

Kohteen etäohjausta varten haluttiin etäohjain, joka olisi mahdollisimman edullinen, helposti saatavilla oleva ja yhteensopiva käytettävän Node-RED-ohjelmointityökalun kanssa. Lisäksi haluttiin, että laitteen ja Node-RED:n voisi yhdistää lokaalisti ilman laitteen omaa pilvipalvelua. Vertailu on esitetty taulukossa 2. Vertailussa pisteytettiin kukin osa-alue numerolla yhdestä viiteen siten, että yksi oli heikoin ja viisi parhain tulos. Lopuksi laskettiin pisteet yhteen. Vertailun perusteella ohjaimeksi valittiin Broadlink RM Pro, joka oli ainoa lokaalisti Node-RED:iin yhdistettävissä oleva laite.

Taulukko 2. Ilmalämpöpumpun etäohjainten vertailu.

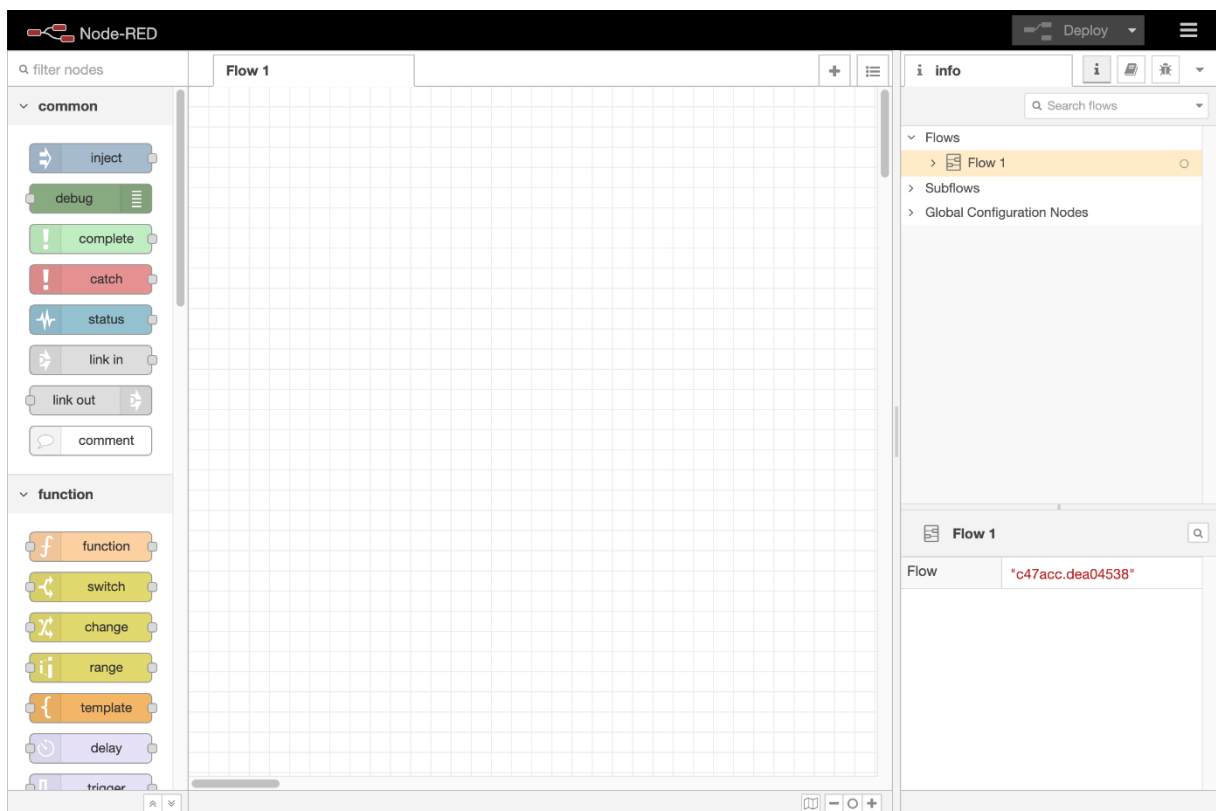
	Panasonic CZ-TACG1	AirPatrol WiFi V5 EU	Sensibo Sky	Nedis SmartLife yleiskaukosäädin	Broadlink RM Pro
Hinta	1	2	2	5	4
Node-RED yhteensopivuus	5	1	5	1	5
Yhdistettävissä Node-RED:iin lokaalisti	1	1	1	1	5
Saatavuus	5	5	5	5	3
Yhteensä	12	9	13	12	17

3 Toteutukseen tarvittavat ohjelmistot, laitteet ja viestintäprotokollat

3.1 Node-RED

Node-RED on graafinen flow-pohjainen ohjelmointityökalu, joka on alun perin IBM:n kehittämä vuonna 2013 ja kuuluu nykyään osaksi OpenJS-säätiötä. Se on Node.js-pohjainen ja sen ohjaamiseen käytetään verkkoselainta. Node-RED:ia voidaan pyörittää esimerkiksi Raspberry Pi -minitietokoneella, paikallisesti tietokoneella, Dockerissa tai Microsoft Azuren pilvessä (OpenJS Foundation, n.d.-a). Node-RED:n päänäköymä on kuvassa 6.

Kuva 6. Node-RED-ohjelmointityökalun työtila (OpenJS Foundation, n.d.-b).



Node-RED:n toiminta perustuu flow-kaavioihin, jotka koostuvat nodeista (OpenJS Foundation, n.d.-a). Tässä opinnäytetyössä käytetään näitä englanninkielisiä termejä flow ja node. Flow-kaaviossa jokaisella nodella on sille tarkkaan määritelty tarkoitus. Jokainen node ottaa vastaan sille annettua dataa, tekee jotain tällä datalla ja lopuksi välittää dataa eteenpäin. Node-RED:n etuna on se, että sen toimintaa on mahdollista ymmärtää, vaikka käyttäjä ei ymmärtäisi jokaisen noden yksittäisiä koodirivejä. (OpenJS Foundation, n.d.-a)

Nodet yhdistetään toisiinsa vetämällä niiden väliin johdot (engl. wires). Jokaisella nodella voi olla enintään yksi sisääntuloportti ja useita ulostuloportteja. Yksittäisen noden sisääntuloporttiin voidaan kuitenkin yhdistää johto useamman eri noden ulostuloista. Ladatut nodet löytyvät työtilan vasemmasta reunasta, ja niitä on mahdollista ladata lisää lataamalla Node-RED:iin erilaisia lisäosia. (OpenJS Foundation, n.d.-b)

Node-RED Dashboard (node-red-dashboard) on Node-RED:in lisäosa, joka tarjoaa mahdollisuuden luoda reaaliaikaisen mittaristonäkymän (dashboard) datasta (OpenJS Foundation, n.d.-f). Tässä opinnäytetyössä käytetään englanninkielistä termiä dashboard. Dataa on mahdollista esittää esimerkiksi tekstimuodossa, graafina tai erilaisina graafisina mittareina. Lisäksi dashboardin avulla voidaan saada käyttäjältä syötteitä esimerkiksi liukusäätimellä tai painikkeella. (OpenJS Foundation, n.d.-f)

Node-RED Broadlink Control (node-red-contrib-broadlink-control) on Node-RED:iin ladattavissa oleva lisäosa. Lisäosan avulla voidaan ohjata Broadlink-merkkisiä infrapunaohjaimia ja siten kontrolloida erilaisia infrapunaohjattavia laitteita, kuten ilmastointilaitteita ja televisioita. Lisäosa mahdollistaa Broadlink-laitteiden etsimisen verkosta, kaukosäätimen komentojen opettelun sekä opeteltujen komentojen lähettämisen. (OpenJS Foundation, n.d.-c)

Remote-RED (node-red-contrib-remote) on Node-RED:iin asennettavissa oleva lisäosa, jonka avulla voidaan saada etänä mobiililaitteella pääsy Node-RED:n dashboard-näkymään. Sovelluksen käyttö on maksullista, ja siihen on saatavissa erihintaisia tilauksia sen mukaan, mitä toimintoja sovelluksen kautta halutaan käyttää. Mobiililaitteeseen tulee ladata sovelluskaupasta Remote-RED-sovellus. (Heilman, n.d.-a) Pelkkä dashboardin etäkäyttö maksaa sovelluksen kautta 6,49 euroa vuodessa.

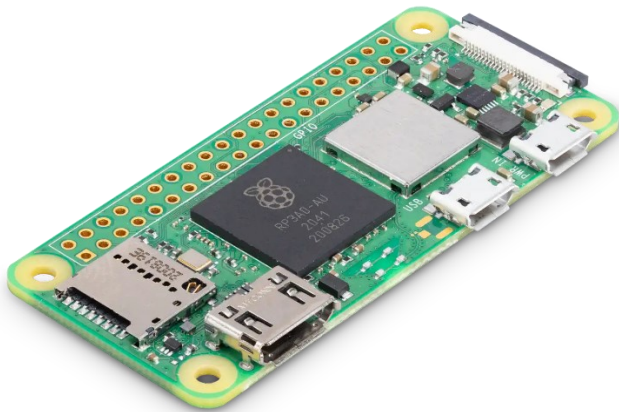
Remote-RED toimii siten, että se muodostaa yhden tai useamman SSH-tunnelin Remote-RED:n palvelimien ja oman Node-RED-palvelimen välille. Remote-RED-sovelluksen kautta suoritettu kommunikointi Remote-RED:n palvelinten kanssa näiden SSH-tunnelien käyttämiseksi on kryptattu ja suojattu asiakassertifikaatilla. Palvelu pyrkii korkeaan tietoturvasoon, mutta tästä huolimatta palvelu muodostaa verkkoon yhden tai useamman tunnelin, mikä aiheuttaa tietoturvariskin. Tästä syystä suositellaan suojaamaan palvelut salasanoilla ja käyttämään kriittisiä järjestelmiä omalla vastuulla. (Heilman, n.d.-b)

3.2 Raspberry Pi

Raspberry Pi on edullinen ja pienikokoinen yhden piirilevyn Linux-pohjainen tietokone. Sen on julkaissut alun perin vuonna 2012 Raspberry Pi-säätiö, jonka tavoitteena on ollut tehdä tietotekniikan opiskelusta helpommin lähestyttävää. Laitteita käytetään yleisesti esimerkiksi ohjelmoinnin opiskeluun, kotiautomaatioon sekä teollisiin sovelluksiin. (Opensource, n.d.)

Opinnäytetyötä varten valittiin Raspberry Pi Zero 2 W (kuva 7), joka maksaa suomalaisessa verkkokaupassa noin 30 euroa (Verkkokauppa.com Oyj, n.d.). Laitteessa on monipuoliset liitännät, kuten langaton verkkoyhteys sekä Bluetooth (Raspberry Pi Foundation, 2021), jota tarvitaan RuuviTag-antureiden yhdistämiseen. Raspberry Pi Zero 2 W -laitteessa on 512 megatavua SDRAM-muistia ja yhden gigahertsin neliytiminen suoritin (Raspberry Pi Foundation, n.d.).

Kuva 7. Raspberry Pi Zero 2 W (Raspberry Pi Foundation, n.d.).



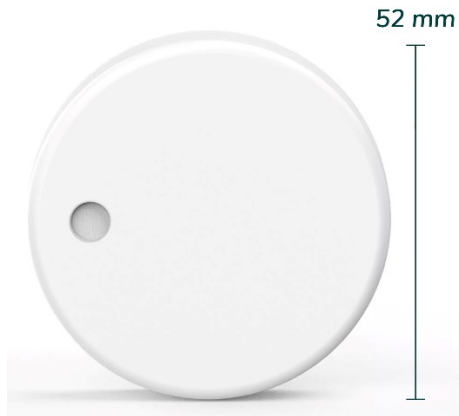
3.3 RuuviTag

RuuviTag on pienikokoinen langaton anturi, joka mittaa lämpötilaa, ilmankosteutta, ilmanpainetta ja liikettä. Virtalähteenä siinä on paristo, jonka käyttöikä on kahdesta kolmeen vuotta. Laitteeseen hankitun erikoispariston avulla käyttölämpötila on -40 asteesta +85 asteeseen. Anturin hinta yksittäin ostettuna on noin 40 euroa. (Ruuvi Innovations Oy, n.d.-a)

RuuviTag (kuva 8) käyttää kommunikointiin oletusarvoisesti protokollanaan Bluetoothia. Laitteen ohjelmisto, elektroniikka ja mobiilisovellukset ovat avointa lähdekoodia, mikä tekee siitä helposti integroitavan. (Ruuvi Innovations Oy, n.d.-a) RuuviTag-anturien kanssa

kommunikointiin on kehitetty Python-kirjasto nimeltä ”ruuvitag-sensor”. Tässä opinnäytetyössä käytetään kahta RuuviTag-anturia. Anturien valintaan vaikutti niiden omistaminen jo ennen projektia.

Kuva 8. RuuviTag anturi (Ruuvi Innovations Oy, n.d.-a)

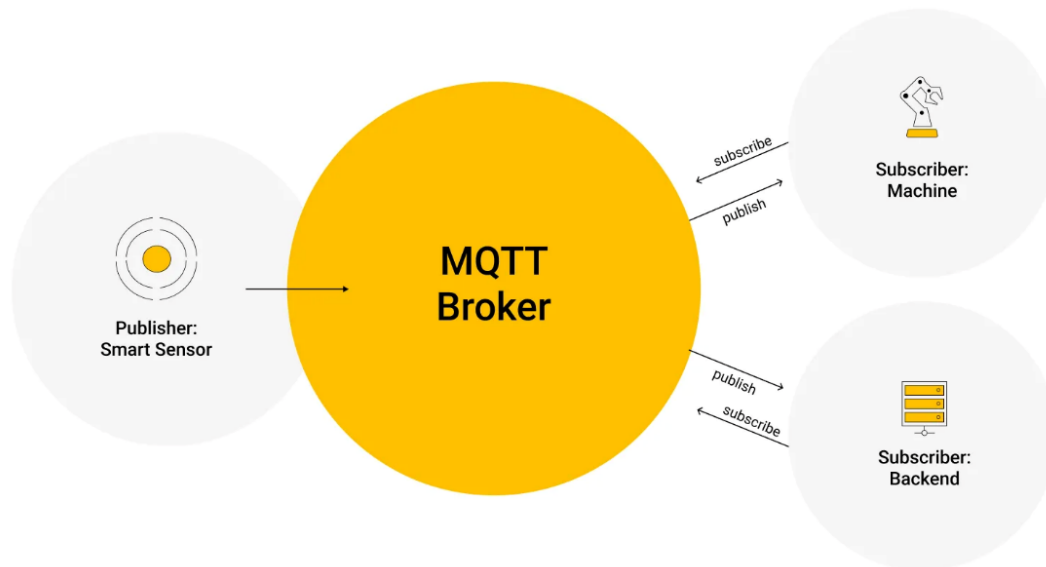


3.4 MQTT

MQTT on yksinkertainen ja kevyt viestintäprotokolla, joka on kehitetty alkujaan vuonna 1999. Sitä käytetään laajasti erilaisiin esineiden internetin (engl. Internet of Things, IoT) sovelluksiin. MQTT sopii erinomaisesti etälaitteiden yhdistämiseen sen käyttämän minimaalisen verkon kaistanleveyden ansiosta. (MQTT.org, n.d.)

MQTT:n toiminta perustuu julkaisuihin (engl. publish) ja tilauksiin (engl. subscribe). Lisäksi keskiössä ovat välittäjä (engl. broker) ja asiakkaat (engl. client). Viestin kulkemiseksi tarvitaan asiakas, joka julkaisee sanomia välittäjälle. Viestin mukana määritetään aihe (engl. topic), joka voi olla esimerkiksi ”mökki/lämpötila”. Sanoman vastaanottavat välittäjän kautta kaikki ne asiakkaat, jotka ovat tilanneet tämän saman aiheen. MQTT:n toiminta yksinkertaistettuna on esitetty kuvassa 9. (HiveMQ, 2024)

Kuva 9. MQTT-protokollan arkkitehtuuri (HiveMQ, 2024).



Tässä opinnäytetyössä käytetään lähettävän pään asiakkaana Eclipse Paho, joka tarjoaa valmiin kirjaston Python-ohjelmointikielelle. Eclipse-säätiön Paho-projekti on avoimen lähdekoodin toteutus, joka tarjoaa erilaisia toteutuksia sekä avoimille että vakioviestiprotokollille (Eclipse Foundation, n.d.-b). Vastaanottavan pään asiakkaana toimii Node-RED.

Välittäjänä tässä opinnäytetyössä käytetään Eclipse-säätiön Mosquittoa. Se on avoimen lähdekoodin kevyt ja vähäisesti virtaa kuluttava viestinvälittäjä. Mosquittoa voidaan pyörittää omalla palvelimella tai vaihtoehtoisesti käyttää projektin tarjoamaa testipalvelinta. (Eclipse Foundation, n.d.-a) Tässä opinnäytetyössä pyritetään Mosquittoa paikallisesti Raspberry Pi:illä.

4 Etäohjauksen ja -seurannan toteutus

4.1 Suunnitelma

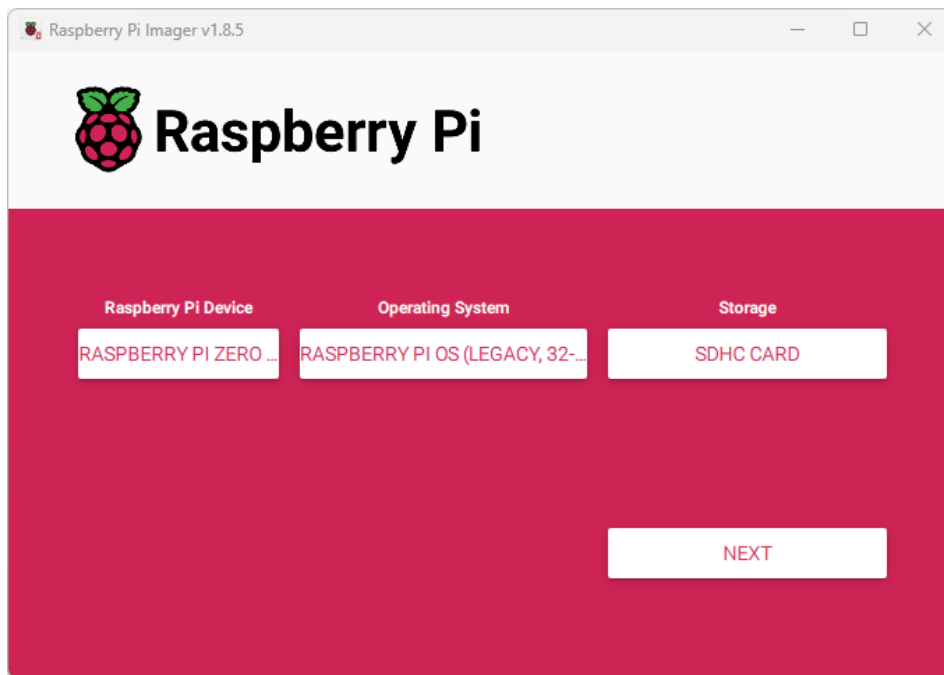
Suunnitelmana oli toteuttaa lämpötilan seuranta ja ilmalämpöpumpun etäohjausjärjestelmä, jota voidaan ohjata etänä mobiililaitteella. Automaatio toteutetaan käyttäen Node-RED:iä, jota ajetaan Raspberry Pi Zero 2 W:ssä. Lämpötilan seurantaan käytetään RuuviTag-antureita, joista haetaan data Python-koodilla ja toimitetaan Node-RED:iin MQTT-sanomilla. Lisäksi liian matalista sisäilman lämpötiloista lähetetään hälytykset haluttuihin puhelinnumeroihin WhatsApp-viesteillä.

Etäohjaukseen käytetään Node-RED:n UI-lisäosaa, jonka dashboard-näkymään päästään Node-RED:n Remote-RED-lisäosan avulla. Ilmalämpöpumpun ohjauksen tulisi toimia siten, että käyttäjän pitää voida muuttaa ilmalämpöpumpun toimintoa lämmityksen ja jäähdytyksen välillä sekä muuttaa lämpötilaa ja tarvittaessa sammuttaa laite.

4.2 Raspberry Pi:n asennus

Raspberry Pi:n käyttö aloitettiin käyttöjärjestelmän asennuksella microSDHC-muistikortille. Ensin avattiin Windows-tietokoneelta Raspberry Pi Imager -sovellus ja syötettiin käytetty 32 gigatavun microSDHC-muistikortti Windows-tietokoneen muistikortinlukijaan. Sovelluksesta valittiin käytettäväksi laitteeksi Raspberry Pi Zero 2 W ja käyttöjärjestelmäksi sovelluksen suosittama Raspberry Pi OS 32-bit (kuva 10).

Kuva 10. Raspberry Pi Imager-sovellus.



Seuraavasta valikosta valittiin laitteen asetusten kustomointi. Avautuvan valikon "general"-osiossa asetettiin palvelimelle ja käyttäjälle halutut nimet sekä käyttäjän salasana. Lisäksi syötettiin langattoman verkon tiedot, jotta laite osaa yhdistää siihen automaattisesti. "Services"-välilehdeltä sallittiin SSH-yhteyden ottaminen laitteeseen. Tämän jälkeen käyttöjärjestelmä ajettiin muistikortille ja se asetettiin Raspberry Pi:hin sekä käynnistettiin laite.

Raspberry Pi:lle ladattiin käyttöjärjestelmä, jossa on graafinen käyttöliittymä. Tämä helpottaa laitteen konfigurointia esimerkiksi silloin, jos langaton verkkoyhteys vaihtuu ja täytyy päästä muuttamaan verkkoyhteyden tietoja. Graafisen käyttöliittymän käyttö onnistuu miniHDMI-liitännän kautta. Tässä opinnäytetyössä Raspberry Pi:tä käytetään Windows-tietokoneen terminaalista SSH-yhteyden kautta. Laitteeseen muodostettiin SSH-yhteys komennolla:

```
ssh käyttäjätunnus@palvelin.local
```

Tämän jälkeen syötettiin salasana, jonka jälkeen laitetta voitiin käyttää komentoriviltä. Laitteeseen haettiin vielä viimeisimmät päivitykset komennoilla:

```
sudo apt update  
sudo apt upgrade -y
```

4.3 Node-RED-ohjelmointityökalun asennus

Node-RED asennettiin Raspberry Pi:lle yksinkertaisella komennolla, joka latsi ja ajoi automaattisesti asennukseen tarvittavat komentosarjat. Käytetty komento oli:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-
installers/master/deb/update-nodejs-and-nodered)
```

Asennuskomentosarja latsi Node-RED:n, Node.js:n ja npm:n. Asennukseen meni aikaa noin kuusi minuuttia ja sen jälkeen asennuskomentosarja loi settings.js-tiedoston annettujen syötteiden perusteella (kuva 11).

Kuva 11. Node-RED:n asetusten asettaminen.

```
Node-RED Settings File initialisation
=====
This tool will help you create a Node-RED settings file.
✓ Settings file · /home/pi/.node-red/settings.js

User Security
=====
✓ Do you want to setup user security? · Yes
✓ Username ·
✓ Password · *****
✓ User permissions · full access
✓ Add another user? · No

Projects
=====
The Projects feature allows you to version control your flow using a local git repository.
✓ Do you want to enable the Projects feature? · Yes
✓ What project workflow do you want to use? · manual - you must manually commit changes

Editor settings
=====
✓ Select a theme for the editor. To use any theme other than "default", you will need to install @node-red-contrib-theme
s/theme-collection in your Node-RED user directory. · default
✓ Select the text editor component to use in the Node-RED Editor · monaco (default)

Node settings
=====
✓ Allow Function nodes to load external modules? (functionExternalModules) · Yes

Settings file written to /home/pi/.node-red/settings.js
```

Seuraavaksi käynnistettiin Node-RED:n prosessi komennolla:

```
node-red-start
```

Tämä komento pitää prosessin käynnissä, vaikka palvelu pysäytettäisiin komennolla Ctrl-C tai ikkuna suljettaisiin. Tämä ei kuitenkaan käynnistä Node-RED:iä Raspberry Pi:n uudelleenkäynnistyksen jälkeen. Koska Node-RED:n haluttiin käynnistyvän automaattisesti

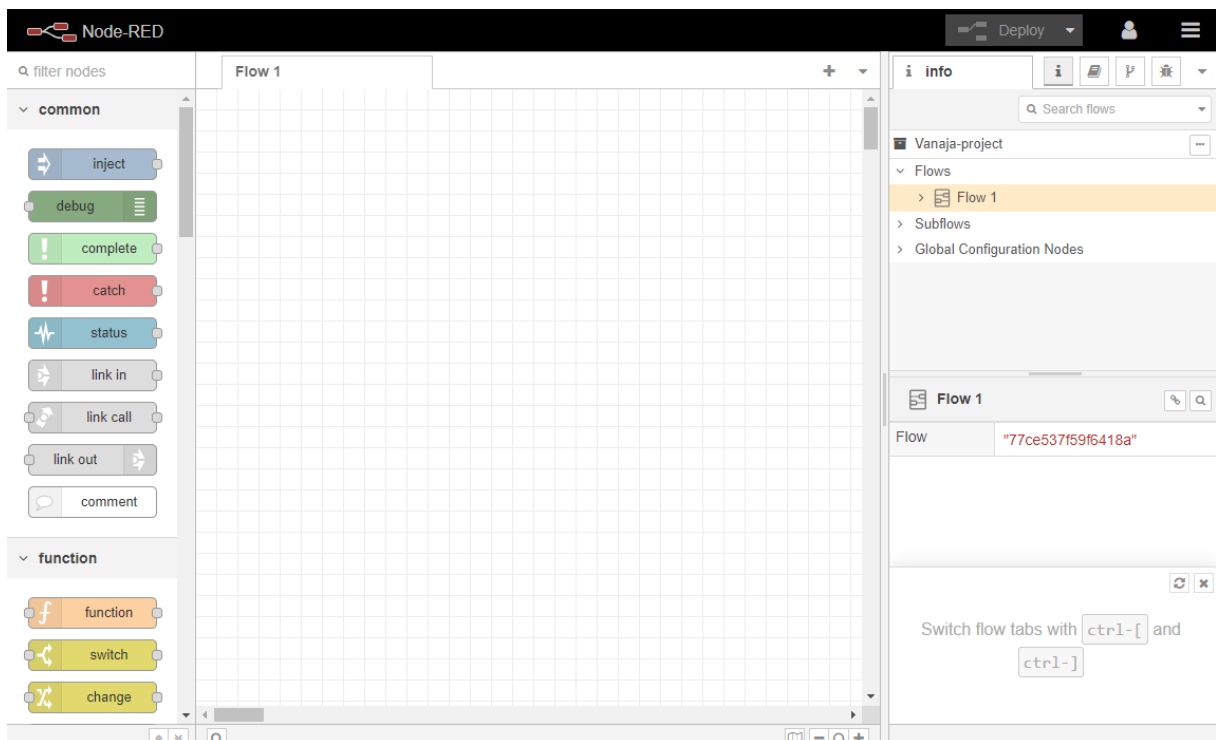
aina Raspberry Pi:n käynnistyessä, pysäytettiin aiemman komennon suoritus näppäimillä Ctrl-C ja annettiin komento:

```
sudo systemctl enable nodered.service
```

Nyt Node-RED oli valmiina käyttöön. Palvelu pyörii portissa 1880, jonne päästiin Windows-tietokoneen selaimella osoitteessa <http://<palvelimen nimi>:1880>. Palvelu pyytää aiemmin luotua käyttäjätunnusta ja salasanaa, joiden antamisen jälkeen seuraa ohjattu esittely.

Koska Node-RED:n asennuksen yhteydessä otettiin versionhallinta käyttöön, pyytää palvelu tätä varten asettamaan käyttäjätunnuksen ja sähköpostiosoitteen. Lisäksi projektille ja Flow-tiedostolle tulee antaa nimet ja päättää, sallitaanko credentials-tietojen salaus. Credentials-tietojen salaus sallittiin ja annettiin valinnainen salausavain. Tämän jälkeen Node-RED:in graafinen käyttöliittymä on valmiina käytettäväksi. Kaavioihin muutoksia tehdessä tulee aina ajaa ohjelma palvelimen kautta valitsemalla oikeasta yläkulmasta "deploy." Kuvassa 12 on esitetty Node-RED:n käyttöliittymän päänäkymä.

Kuva 12. Node-RED:n käyttöliittymän päänäkymä.



4.4 RuuviTag-anturien asennus

RuuviTag-anturien asennus aloitettiin lataamalla komentoriviltä Pythonin ”ruuvi-tag-sensor”-kirjasto komennolla:

```
python -m pip install ruuvi-tag-sensor
```

RuuviTag-kirjasto tarvitsee lisäksi toimiakseen ”bluez-hcidump” -paketin, joka sallii Raspberry Pi:n ja Bluetooth-laitteiden välisen kommunikoinnin. Tämä asennettiin komennolla:

```
sudo apt-get install bluez-hcidump && echo +++ install successful +++
```

Tämän jälkeen oltiin valmiita testaamaan käytössä olevien kahden RuuviTagin tietojen hakemista komennolla:

```
python3 ~/.local/lib/python3.9/site-packages/ruuvi-tag_sensor -f
```

Saatu tuloste on esitetty kuvassa 13. Kirjaston avulla löydettiin kaksi RuuviTagia, ja tulostettiin näiden lähettämä data. MAC-koodit on sensuroitu kuvasta.

Kuva 13. RuuviTagien tuloste.

```
Finding RuuviTags. Stop with Ctrl+C.
Start receiving broadcasts (device hci0)
FYI: Calling a process with sudo: hciconfig hci0 reset
FYI: Spawning process with sudo: hcidump -i hci0 --raw
FYI: Spawning process with sudo: hcitool -i hci0 lscan2 --duplicates --passive
FD: [REDACTED]
{'data_format': 5, 'humidity': 48.33, 'temperature': 6.99, 'pressure': 1005.36,
 'acceleration': 1004.462045076866, 'acceleration_x': 12, 'acceleration_y': 28,
 'acceleration_z': 1004, 'tx_power': 4, 'battery': 2897, 'movement_counter': 5,
 'measurement_sequence_number': 62842, 'mac': 'fd43120b8361', 'rssi': -84}
F5: [REDACTED]
{'data_format': 5, 'humidity': 40.73, 'temperature': 19.16, 'pressure': 1005.39,
 'acceleration': 990.1434239543279, 'acceleration_x': 64, 'acceleration_y': -12,
 'acceleration_z': 988, 'tx_power': 4, 'battery': 2929, 'movement_counter': 36,
 'measurement_sequence_number': 2414, 'mac': 'f5e5e581dc6f', 'rssi': -79}
```

Seuraavaksi asennettiin paikallinen Mosquitto broker MQTT-sanomien lähetyksistä varten. Lisäksi annettiin tälle palvelulle lupa käynnistyä palvelimen uudelleenkäynnistyksen yhteydessä ja käynnistettiin palvelu. Näissä käytettiin komentoja:

```
sudo apt install -y mosquitto mosquitto-clients
sudo systemctl enable mosquitto.service
sudo systemctl start mosquitto.service
```

Tietoturvan vuoksi haluttiin, että MQTT-sanomia on mahdollista lähettää ainoastaan tunnusta ja salasanaa käyttäen. Tämän toteuttaminen vaati Mosquitto-palvelun konfigurointia siten, ettei palvelua voi käyttää anonyymisti. Mosquitton konfiguraatiotiedosto löytyy polusta `"/etc/mosquitto/mosquitto.conf"`. Tiedoston loppuun lisättiin seuraavat rivit:

```
listener 1883
allow_anonymous false
password_file /etc/mosquitto/pwdfile
```

Seuraavaksi luotiin käyttäjätunnus ja salasana, joita käyttäen voidaan lähettää paikalliselle Mosquittole MQTT-sanomia. Komentona oli:

```
sudo mosquitto_passwd -c /etc/mosquitto/pwdfile <käyttäjätunnus>
```

Tämän jälkeen komentoriville tuli syöttää haluttu salasana. Salasanan luonnin jälkeen palvelu käynnistettiin uudestaan komennolla:

```
sudo systemctl restart mosquitto.service
```

Seuraava vaihe oli Python-ohjelman luominen. Tässä käytettävää MQTT-protokollaa varten ladattiin Paho client Python-kirjasto komennolla:

```
pip3 install paho-mqtt
```

Python-ohjelman tarkoitus oli hakea kahden RuuviTagin datat ja lähettää ne valittuihin MQTT-aiheisiin Node-RED:lle noudettavaksi. Ohjelmakoodi on esitetty liitteessä 1. Ohjelmakoodissa on käytetty aiemmin ladattuja Python-kirjastoja `paho.mqtt.client` ja `ruuvitag_sensor.ruuvi`. MQTT-brokerille yhdistettäessä täytyy antaa parametrina `"client.username_pw_set()"`-funktiolle aiemmin luomamme käyttäjätunnus ja salasana.

RuuviTagien datat on tarkoitus lähettää Node-RED:iin viiden minuutin välein. Tämän toteuttamiseksi käytettiin crontab:ia. Crontab:n muokkaus tehtiin komennolla:

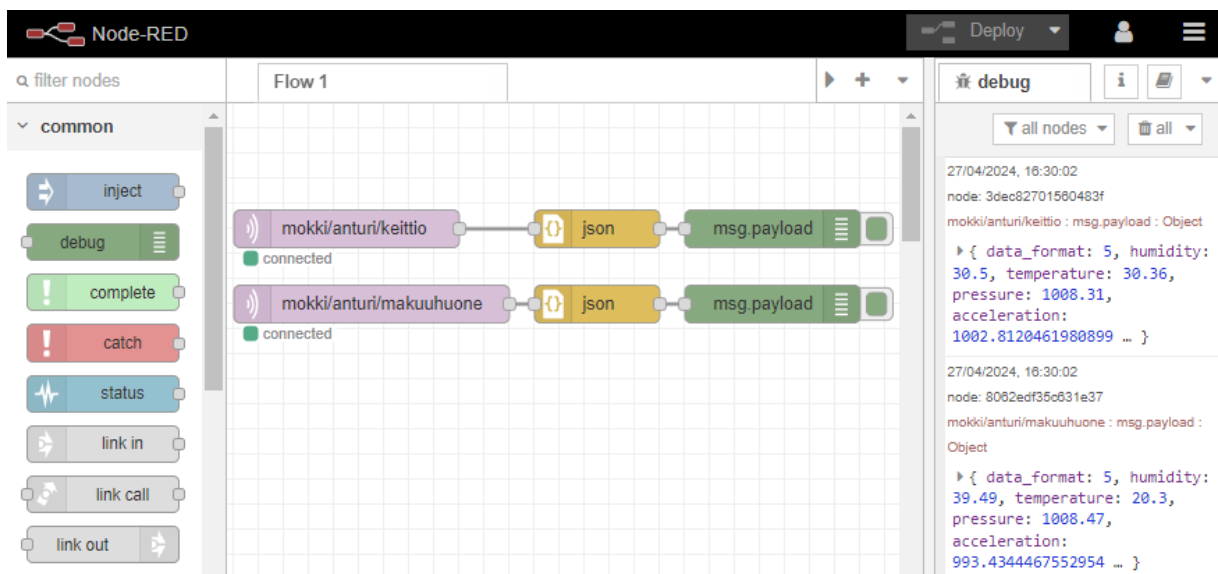
```
crontab -e
```

Tiedostoon lisättiin alimmaisiksi rivit `*/5 * * * * /usr/bin/python3 /home/pi/ruuvit/getAndSendRuuviDatas.py`, joka määritteli Linux-järjestelmään uuden aikataulutetun tehtävän. Tehtävä käynnistää viiden minuutin välein Pythonin suorittamaan juuri luodun tiedoston `getAndSendRuuviDatas.py`.

Viimeisenä vaiheena RuuviTagien asennuksessa konfiguroitiin Node-RED kuuntelemaan MQTT-topic-kanavia, joihin Python-ohjelma lähettää sanomia. Node-RED:sta löytyy valmiiksi MQTT-sanomien käsittelyyn tarkoitettuja nodeja `network`-kategorian alta. Valittiin käytettäväksi kaksi `mqtt-in`-nodea, jotka vedettiin flow-kaavioon. Nodet muokattiin kuuntelemaan localhost-serveriä ja porttia 1883, joka on MQTT:n oletusportti. Serverien asetuksissa tuli security-välilehden alle syöttää aiemmin konfiguroitu käyttäjätunnus ja salasana, jotta Node-RED pystyi yhdistämään MQTT-brokerille. Ensimmäiselle nodelle määriteltiin seurattavaksi topic-kanavaksi `mokki/anturi/keittio` ja toiselle `mokki/anturi/makuuhuone`.

RuuviTagien lähettämä data on JSON-muotoista, ja se haluttiin muuttaa datan tulevaa jatkokäsittelyä varten JavaScript-objektiksi. Molempien MQTT:ta kuuntelevien nodejen perään raahattiin `json`-nodet. Viimeiseksi valittiin kaavioiden perään `debug`-nodet, joiden avulla saadaan tuloste debug-konsoliin kaavioissa liikkuvasta datasta. Tämän vaiheen flow-kaavio on esitetty kuvassa 14.

Kuva 14. Node-RED ensimmäisten nodejen lisäyksen jälkeen.

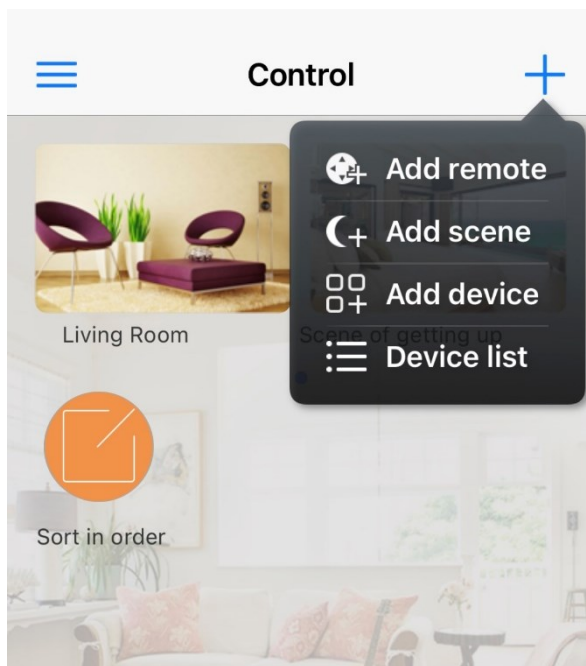


4.5 Broadlink RM PRO -infrapunaohjaimen asennus

4.5.1 e-Control-sovelluksen asennus

Broadlink RM PRO-infrapunaohjaimen asennus aloitettiin sijoittamalla laite ilmalämpöpumpun lähelle ja kytkemällä se verkkovirtaan. Seuraavaksi ladattiin iPhone-mobiililaitteeseen AppStore-sovelluskaupasta Broadlinkin sovellus nimeltä e-Control. Sovelluksessa valittiin päänäkömstä oikealta yläkulmasta plus-painikkeen alta ”Add device” (kuva 15), ja tämän jälkeen syötettiin käytössä olevan langattoman verkon tiedot. Näin saatiin laite yhdistettyä langattomaan verkkoon.

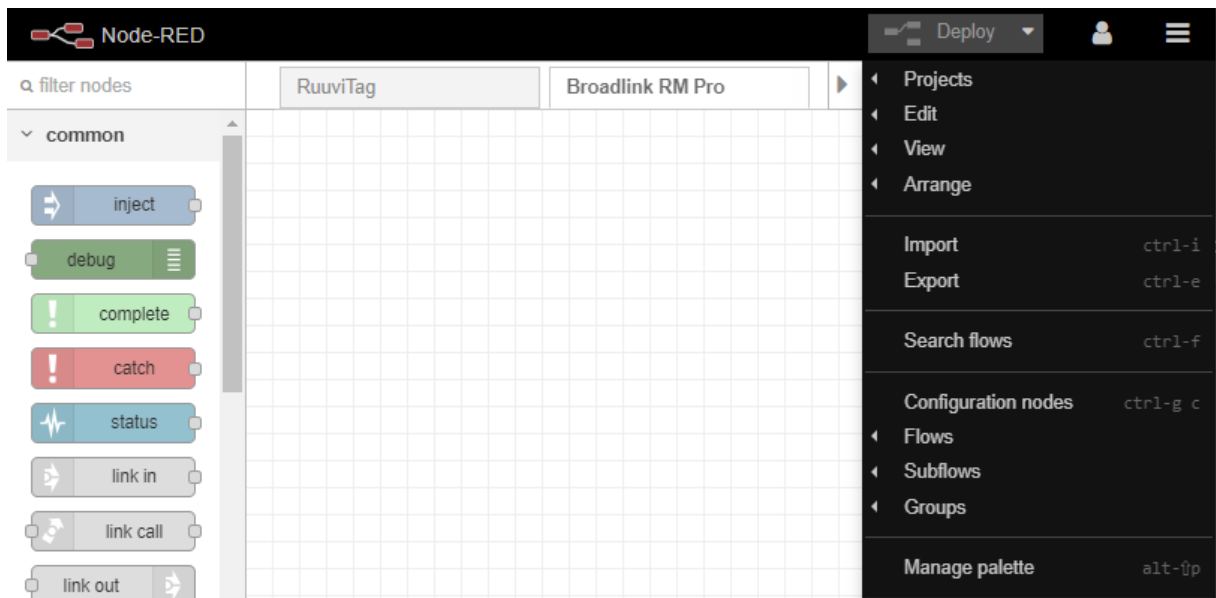
Kuva 15. e-Control sovelluksen päänäkömä.



4.5.2 Etäohjaimen asennus Node-RED:iin

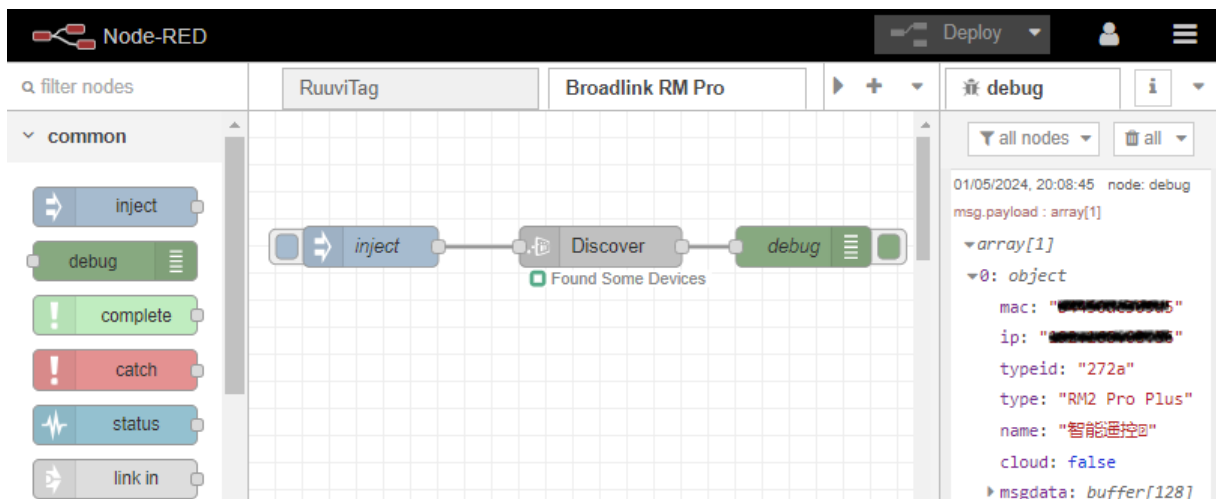
Seuraava vaihe oli Broadlink RM PRO:n yhdistäminen Node-RED-ohjelmointityökaluun. Node-RED:iin asennettiin lisäosa, jonka avulla se voidaan yhdistää paikallisesti Broadlink-laitteeseen. Lisäosien asennus onnistuu Node-RED:n päävalikosta valitsemalla ”manage palette” (kuva 16). Avautuneesta valikosta valittiin ”install”-välilehti, jossa haettiin ”node-red-contrib-broadlink-control” ja asennettiin tämä. Broadlinkin lisäosan asennus lisäsi päänäkömän vasemmassa reunassa olevaan node-listaukseen kohdan ”broadlink”.

Kuva 16. Node-RED:n päävalikko.



Seuraavaksi luotiin flow-kaavio, jolla voidaan hakea verkossa olevien Broadlink-laitteiden tiedot. Node-listauksen "common"-osiosta raahattiin ensimmäiseksi nodeksi "inject". Tämän perään haettiin "broadlink"-osiosta node nimeltä "discover". Viimeiseksi lisättiin "common"-osiosta "debug"-node, jolloin nähdään tehdyn flow-kaavion tulostus. "Deploy"-painikkeen painamisen jälkeen painettiin "inject"-noden vasemmassa reunassa olevaa painiketta, jonka seurauksena saatiin debug-konsoliin tulosteena laitteen tiedot (kuva 17).

Kuva 17. Flow-kaavio Broadlink-laitteen tietojen hakemiseksi.



Jotta voidaan opettaa Node-RED:lle ilmalämpöpumpun kaukosäätimen komentoja ja lähettää näitä, tarvitaan laitteen MAC-osoite, IP-osoite sekä laitteen tyyppi-id. Lisäksi tarvitaan "SharedData"-hakemiston polku.

4.5.3 Kaukosäätimen komentojen opettelu ja tallentaminen

Seuraavaksi luotiin flow-kaavio, jonka avulla voidaan opetella ilmalämpöpumpun kaukosäätimen komentoja. Ensimmäiseksi nodeksi valittiin edellisen vaiheen tapaan "inject". Tämän perään otettiin "broadlink"-osiosta "RM"-niminen node. Tätä nodea tuplaklikkaamalla saatiin auki sen konfiguraatioasetukset. "Device"-kohdasta valittiin "add new rmdevice" ja painettiin viereistä kynä-ikonia. Tästä aukesi valikko, johon syötettiin aiemmassa vaiheessa saadut MAC-osoite, IP-osoite ja tyyppinumero. Lisäksi tarvittiin "catalog"-kohtaan hakemiston polku, johon laitettiin tätä varten luotu "/home/pi/catalog".

Tämän jälkeen painettiin "add", valittiin "Action"-kohdasta "learn" ja hyväksyttiin noden asetukset valitsemalla "done". Viimeiseksi nodeksi valittiin "debug". Nyt "inject"-noden painiketta painaessa debug-konsoliin tulostui teksti, joka pyysi painamaan kaukosäätimen opeteltavasta painikkeesta 30 sekunnin kuluessa. Painiketta painettua saatiin debug-konsoliin tulosteena objekti, joka sisältää listamuotoisen "data":n sekä "base64data":n, jonka muoto on määrittelemätön. Näistä ensimmäinen on kaukosäätimen ilmalämpöpumpulle lähettämä komento ja tätä tullaan käyttämään Node-RED:n ilmalämpöpumpulle lähetettävänä komentona. Debug-tulosteet ovat esitetty kuvassa 18.

Kuva 18. Kaukosäätimen komennon opettelusta saatu tuloste.

```
02/05/2024, 18:51:27 node: ac0302cc.1943b
msg : string[68]
"Broadlink: IR Scan - Please tap the
remote button within 30 seconds."

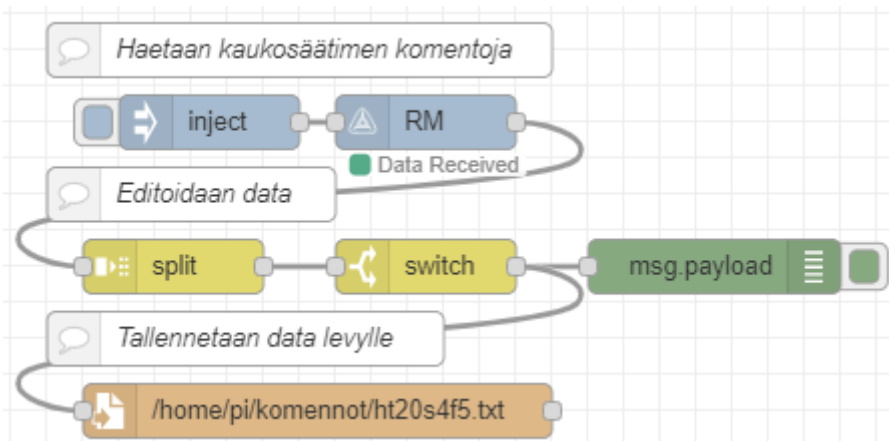
02/05/2024, 18:51:29 node: debug 2
msg.payload : Object
  ▾ object
    ▸ data: array[460]
      base64data: undefined
```

Seuraavaksi muokattiin komentoja hakevaa flow-kaaviota siten, että "debug"-node irrotettiin ja "RM"-noden perään lisättiin "split"-node. Tämä node jakaa payloadin kahteen erilliseen osaan. Koska haluttiin käyttää vain ensimmäistä osaa, valittiin seuraavaksi nodeksi "switch".

Tämän asetuksista valittiin säännöksi "head" ja arvoksi 1, jolloin node poimii saapuvan listan alusta ensimmäisen indeksin sisällön. Nyt kaavion lopullisena arvona on pelkkä ilmalämpöpumpulle lähetettävä komento.

Opetellut komennot tallennettiin Raspberry Pi:n levyille, jotta ne jäävät muistiin ja niiden käyttö olisi jatkossa sujuvampaa. Tämä tehtiin lisäämällä kaavion perään vielä "write file" -node. Noden asetuksissa sijoitettiin tiedoston poluksi `"/home/pi/komennot/"` ja tiedostonimeksi `"<tiedostonimi>.txt"`. Koska ilmalämpöpumpun kaukosäädin lähettää jokaisella painikkeen painalluksella kaikki kaukosäätimessä olevat asetukset, määritettiin tiedostonimi asetusten mukaisesti. Esimerkiksi `"ht20s4f5.txt"` sisältää komennon, joka määrittää toimintotilaksi lämmityksen (heat), lämpötilaksi 20 (temp), puhalluksen kulmaksi 4 (swing) ja puhalluksen voimakkuudeksi 5 (fan). Kaukosäätimen komentoja tallentava flow-kaavio on esitetty kuvassa 19.

Kuva 19. Kaukosäätimen komentoja tallentava flow-kaavio.



Opetuskaavion avulla tallennettiin Raspberry Pi:n levyille kaikki komennot, joita ilmalämpöpumpulle oli määrä voida lähettää. Koska etäohjauksella oli tarpeen muuttaa ainoastaan ilmalämpöpumpun toimintoa lämmityksen ja viilennyksen välillä sekä säätää lämpötilaa asteen tarkkuudella, ei komentoja tarvinnut opettaa suurta määrää. Komentojen opetusprosessissa muutettiin tallennettavan tiedoston nimeä, koottiin Node-RED "deploy"-painikkeella ja painettiin tämän jälkeen "inject"-painiketta. Tämän jälkeen painettiin ilmalämpöpumpun kaukosäätimestä haluttu tila. Esimerkiksi tallennettaessa lämmitystoiminnon lämpötilaa 18 asetettiin kaukosäätimen ohjaus ennen injektointia 18,5 asteen asentoon. "Inject"-noden napin painalluksen jälkeen painettiin kaukosäätimestä lämpöä yhden alaspäin, jolloin saatiin komento 18 asteelle. Kuvassa 20 on esitetty hakemiston `"/home/pi/komennot/"` sisältö eli kaikki tallennetut komennot.

Kuva 20. Raspberry Pi:n levyille tallennetut komennot.

```

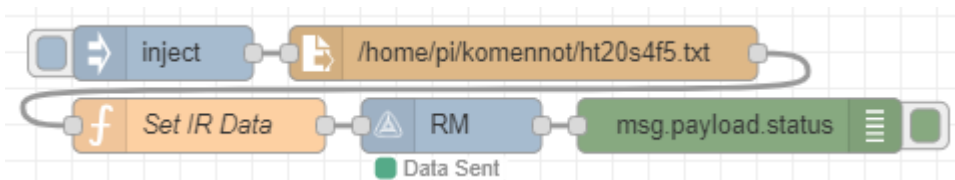
ct16s4f5.txt ct21s4f5.txt ht10s4f5.txt ht15s4f5.txt ht20s4f5.txt ht25s4f5.txt
ct17s4f5.txt ct22s4f5.txt ht11s4f5.txt ht16s4f5.txt ht21s4f5.txt ht8s4f5.txt
ct18s4f5.txt ct23s4f5.txt ht12s4f5.txt ht17s4f5.txt ht22s4f5.txt ht9s4f5.txt
ct19s4f5.txt ct24s4f5.txt ht13s4f5.txt ht18s4f5.txt ht23s4f5.txt off.txt
ct20s4f5.txt ct25s4f5.txt ht14s4f5.txt ht19s4f5.txt ht24s4f5.txt

```

4.5.4 Komentojen lähettäminen ilmalämpöpumpulle

Seuraavaksi luotiin kaavio (kuva 21), jonka avulla ilmalämpöpumpulle lähetettiin komentoja. Ensimmäiseksi nodeksi valittiin "inject" ja seuraavaksi "read file", jonka avulla voidaan lukea aiemmassa vaiheessa tallennettujen tiedostojen sisältöjä. Tiedoston nimeksi määriteltiin aiempaa vaihetta vastaava "/home/pi/komennot/ht20s5f5.txt". Seuraavaksi valittiin "function"-node, jonka avulla voidaan lisätä kaavioon oma JavaScript-funktio. Funktion tarkoituksena on käyttää tiedostosta luettua ilmalämpöpumpulle lähetettävää komentoa ja yhdistää se sanomaan, jonka seuraavaksi lisättävä "RM"-node käsittelee. Funktion JavaScript-koodi on liitteessä 2.

Kuva 21. Ilmalämpöpumpulle komentoja lähettävä flow-kaavio.



RM-noden asetuksista valittiin laitteeksi käytetty ohjain ja toiminnoksi valittiin "set from msg.payload", joka muodostaa lähetettävän komennon saamansa sanoman perusteella. Viimeiseksi nodeksi sijoitettiin "debug", jotta voitiin todeta sanoman lähetyksen onnistuneen. Inject-painiketta painamalla nähtiin, että etäohjaimen syttyi oranssi valo, debug-konsoliin tulostui teksti "OK" ja ilmalämpöpumppu antoi merkkiäänän komennon vastaanottamisen merkiksi, sekä muutti puhalluksen suuntaa pystysuunnassa. Näin voitiin todeta flow-kaavion toimivan oikein.

4.6 Dashboard-lisäosan asennus

Seuraava vaihe oli tehdä ilmalämpöpumpulle komentojen lähettämisestä ja RuuviTag-anturien mittaustuloksista käyttäjäystävällisempiä. Tätä varten ladattiin Node-RED:iin lisäosa

node-red-dashboard. Lisäosan myötä node-valikkoon ilmestyi uusi osio "dashboard" ja sen alle erilaisia vaihtoehtoja käyttäjäsyötteen saamiseksi ja datan esittämiseksi. Dashboard-näkymään pääsee oletuksena selaimella syöttämällä osoitteeksi "http://<palvelimen nimi>:1880/ui".

Ilmalämpöpumpun ohjaus toteutettiin siten, että lämmitykselle ja jäähdytykselle luotiin omat valikot. Ensin luotiin lämmitystoiminnon valikko. Käyttäjäsyötteen saamiseksi lisättiin "slider"-node, jonka asetuksista lisättiin uusi dashboard-ryhmä nimeltä "lämmitys". Ryhmään lisättiin uusi välilehti eli "tab", joka nimettiin myös nimellä "lämmitys". Kaikki tähän ryhmään jatkossa lisättävät nodet tulevat näkyviin samalle välilehdelle. Lopuksi määriteltiin vielä asetuksista liikusäätimen minimiarvoksi 8 ja maksimiarvoksi 25 sekä annettiin nodelle vielä nimi "lämmitys." Tämä node palauttaa numeerisen arvon 8 ja 25 välillä.

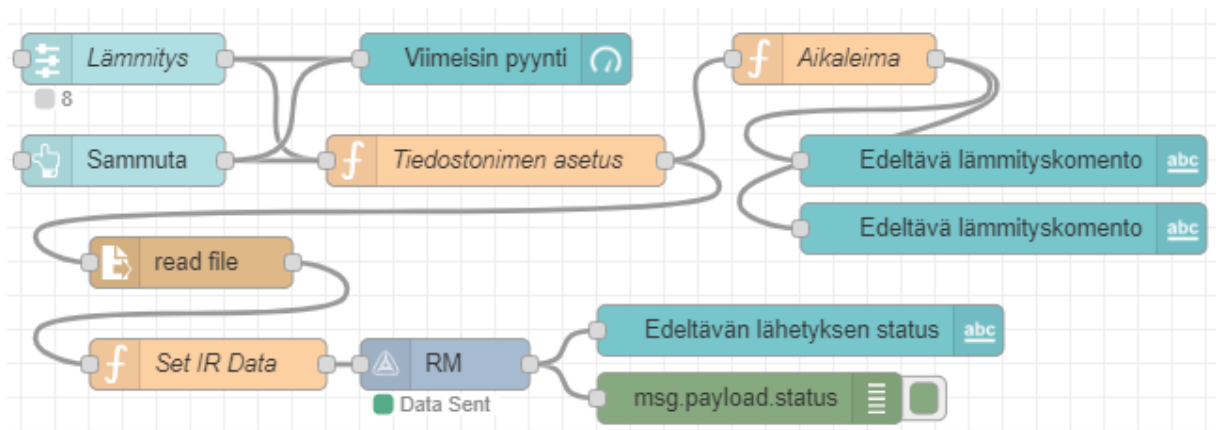
Ilmalämpöpumppu tulee voida myös sammuttaa, joten tätä toimintoa varten lisättiin "button"-node. Tämän asetuksista valittiin ryhmäksi aiemmin luotu "lämmitys" ja syötettiin "label"-kohtaan painikkeeseen tulevaksi tekstiksi "sammuta". Lisäksi määriteltiin "payload"-arvoksi "off", jolloin painiketta painaessa sen palauttamaksi arvoksi tulee "off".

Seuraavaksi yhdistettiin nämä luodut nodet aiemmin luotuun flow-kaavioon (kuva 21), joka luki tiedoston ja lähetti sen sisällön perusteella ilmalämpöpumpulle komennon. "Inject"-nodea ei enää tarvittu, koska syöte tulee käyttäjältä. Tiedostoa lukevan noden asetuksia muutettiin siten, että tiedostonimen kohdasta vaihdettiin "path" pois ja laitettiin tilalle "msg.payload". Tällöin tiedosto tullaan hakemaan siitä polusta, jonka arvon node saa edeltävältä nodelta.

Käyttäjäsyötteitä vastaanottavien nodejen ja tiedoston sisältöä lukevan noden väliin luotiin "function"-node, joka ottaa vastaan "lämmitys"- tai "sammutus"-nodelta tulevan arvon ja sijoittaa sen payload-arvoksi. Näin tiedostoa lukeva node saa luettavakseen annettua käyttäjäsyötettä vastaavan tiedostonimen. Tämän "function"-noden JavaScript-ohjelmakoodi on esitetty liitteessä 3.

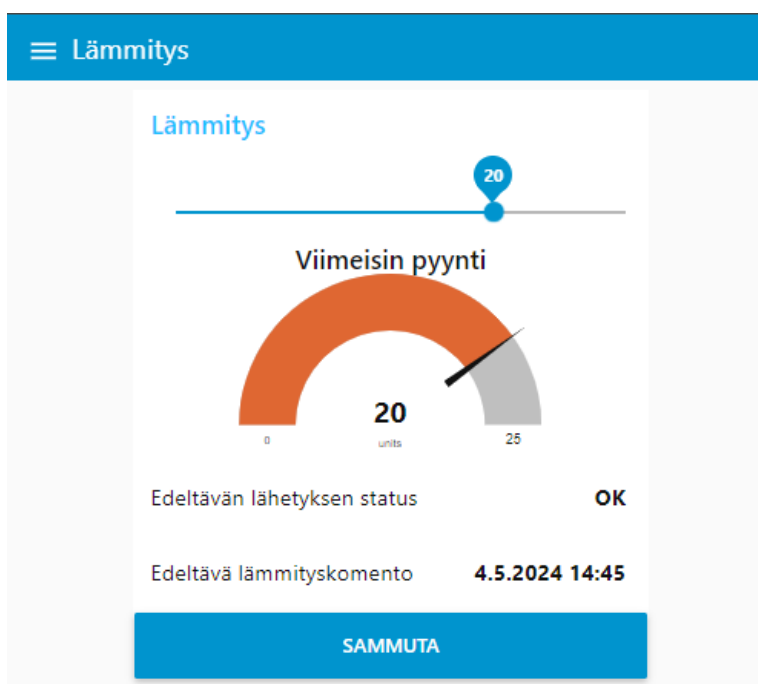
Dashboard-näkymää varten lisättiin tiedot siitä, mitä lämpötilaa on viimeksi pyydetty ja millä aikaleimalla, sekä pyynnön status. Lisäykset tehtiin kuvan 22 mukaisesti. Viimeisin pyynti toteutettiin "gauge"-nodella, johon yhdistettiin sekä lämmitysnode että sammutusnode. Noden asetuksista määriteltiin node käyttämään arvoa "{{value}}", jonka se saa lämmitys- tai sammutusnodelta. Mittarin alueen minimiksi määriteltiin 0 ja maksimiksi 25. Arvo 0 tulee silloin, kun ilmalämpöpumppu sammutetaan. Alueeseen valittiin myös värit siten, että minimiarvossa alueen väri on vaalean oranssi ja maksimiarvossa tumman punainen.

Kuva 22. Ilmalämpöpumpulle lämmityskomentoja lähettävä flow-kaavio.



Aikaleimaa varten luotiin "function"-node, johon lisättiin aikaleiman palauttava ohjelmakoodi (liite 4). Aikaleiman esitystä varten luotiin kaksi "text"-nodea, joiden arvoiksi annettiin "{{msg.payload}}", eli aikaleima-noden palauttama arvo. Toisen noden ryhmäksi määriteltiin lämmityksen sijasta "jäähdytys", jotta myös tässä näkymässä voidaan nähdä, milloin viimeksi on annettu lämmityskomento. Viimeiseksi lisättiin "RM"-noden perään "text"-node, joka esittää arvon "{{msg.payload.status}}"-sisällön. Tällöin voidaan nähdä dashboardilta edeltävän lähetyksen status. Tämänhetkinen lämmityksen dashboard-näkymä on esitetty kuvassa 23.

Kuva 23. Lämmityksen dashboard-näkymä.



Jäähdytysnäköymän toteutusta varten kopioitiin kaikki lämmitys-nodet ja muokattiin niitä vastaamaan jäähdytys-osion tarvetta. Kaikkien "dashboard"-nodejen ryhmät vaihdettiin lämmityksestä jäähdytykseen, jolloin ne sijoituivat oikeaan näköymään. "Viimeisin pyynti"-noden värimaailmaa vaihdettiin niin, että minimiarvossa väri on vaaleansininen ja maksimiarvossa tummansininen. "Tiedostonimen asetus"-nodessa tiedostonimen ensimmäinen h-kirjain korvattiin c-kirjaimella viillennyksen merkiksi. Nyt jäähdytys-näköymän dashboard oli myös valmis testattavaksi.

Lämmityksen ja jäähdytyksen dashboard-näköymiin haluttiin vielä lisätä RuuviTag-antureilta saadut keittiön ja makuuhuoneen lämpötilat. Lisäksi haluttiin luoda oma näköymä "mittaukset", johon sijoitetaan graafiset kuvaajat lämpötila-, ilmankosteus- ja ilmanpaine- arvoista. Tähän näköymään lisätään myös graafit, joista näkee RuuviTag-anturien paristojen varaustason.

RuuviTagien datan hakeminen Node-RED:iin jäi aiemmassa vaiheessa kuvan 14 tilanteeseen. Seuraavaksi valittiin keittiön mittaustuloksia tuova flow-kaavio ja lisättiin "json"-noden perään "function"-node. Tähän nodeen kirjoitettiin ohjelmakoodi, jossa jaetaan JavaScript-objektina saadut datat omiin ulostuloihin. Ohjelmakoodi on esitetty liitteessä 5. Noden asetusten "setup"-välilehden alta säädettiin myös ulostulojen määrää neljään. Tämän jälkeen ensimmäisestä ulostulosta saatiin lämpötilan arvo, toisesta ilmankosteus, kolmannesta ilmanpaine ja neljännestä pariston varaustaso.

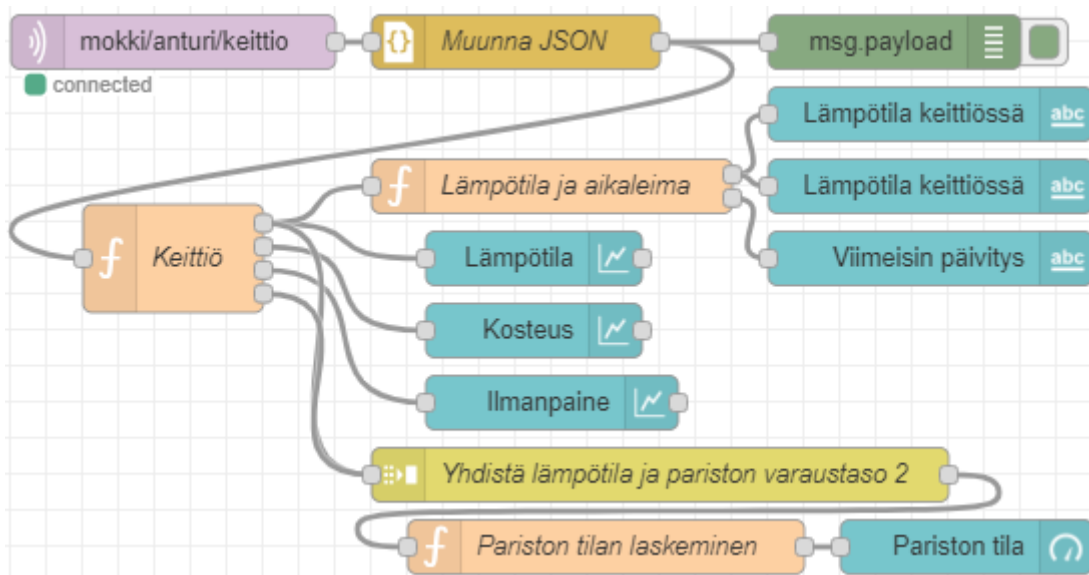
Seuraavaksi luotiin "function"-noden ensimmäiseen, toiseen ja kolmanteen ulostuloon "chart"-nodet. Ensimmäisen nimeksi annettiin "label"-kohtaan lämpötila, toisen kosteus ja kolmannen ilmanpaine. Kaikkiin määriteltiin X-akseli näyttämään graafia yhden viikon ajalta ja valittiin päivämäärän esitysmuodoksi "Day HH:mm". Lisäksi luotiin uusi ryhmä "keittiö", jonka välilehdeksi luotiin "mittaukset". Kaikki kolme luotua nodea määriteltiin tähän "keittiö"-ryhmään.

RuuviTag lähettää pariston varaustason numeerisena arvona, joka kuvaa pariston varaustasoa voltteina. Esimerkiksi lukema 3000 tarkoittaa lukemaa 3.000 volttia (V). Ruuvin omassa RuuviStation-sovelluksessa paristoindikaattori toimii siten, että alle -20 asteen lämpötilassa vähäisen paristojännitteen raja on 2 V, -20–0 asteen välillä vähäisen paristojännitteen raja on 2,3 V ja yli 0 asteessa vähäisen paristojännitteen raja on 2,5 V. (Ruuvi Innovations, n.d.-b) Tässä toteutuksessa päätettiin käyttää tätä samaa lämpötilalogiikkaa.

Seuraavaksi lisättiin "join"-node, johon vedettiin linkit lämpötilan ja pariston varaustason ulostuloista. Tarkoituksensa oli yhdistää nämä arvot yhdeksi listaksi. Noden asetuksista määriteltiin tilaksi manuaali ja valittiin yhdistettäväksi kaikki "msg.payload"-arvot. Näistä määriteltiin muodostettavaksi lista. Määriteltiin myös, että sanoma lähetetään eteenpäin aina, kun on yhdistetty kaksi sanoman osaa. Tähän perään lisättiin "function"-node, joka laskee pariston varaustason aiemmin mainitun kaavan perusteella käyttäen lämpötilaa ja pariston varaustaso. Tämä ohjelmakoodi on esitetty liitteessä 6. Pariston varaustaso palautetaan numeerisena arvona nollan ja sadan väliltä kuvaten prosentteja. Arvossa nolla pariston varaustaso on vähäinen ja viimeistään tällöin paristo on vaihdettava. Viimeiseksi lisättiin vielä "gauge"-node, joka näyttää dashboardilla pariston varauksen arvon.

Viimeiseksi lisättiin kolmas node lämpötilan ulostuloon. Tähän lisättiin "function"-node, jonka tehtävänä oli lisätä aikaleima "mittaukset"-näkömään ja lisätä lämpötila-arvo aikaleimoineen sekä "lämmitys"-näkömään että "jäähdytys"-näkömään. JavaScript-ohjelmakoodi on liitteessä 7. Noden asetuksista määritettiin ulostulojen määräksi kaksi. Ensimmäiseen ulostuloon liitettiin kaksi "text"-nodea, joista toisen ryhmäksi määriteltiin "lämmitys" ja toisen "jäähdytys". Toiseen ulostuloon lisättiin "text"-node, jonka ryhmäksi määriteltiin "mittaukset". Keittiön sijoitetun anturin datan vastaanottava ja esittävä kaavio on esitetty kuvassa 24.

Kuva 24. Keittiön RuuviTag-anturin mittaustulokset dashboardilla esittävä flow-kaavio.



Makuuhuoneeseen sijoitettavan anturin kaavio rakennettiin kopioimalla keittiön kaavio "json"-nodesta eteenpäin. Koska kaikkien mittauksien haluttiin näkyvän samalla "mittaukset" välilehdellä, luotiin uusi ryhmä nimeltä "makkari" ja lisättiin tämän välilehdeksi "mittaukset".

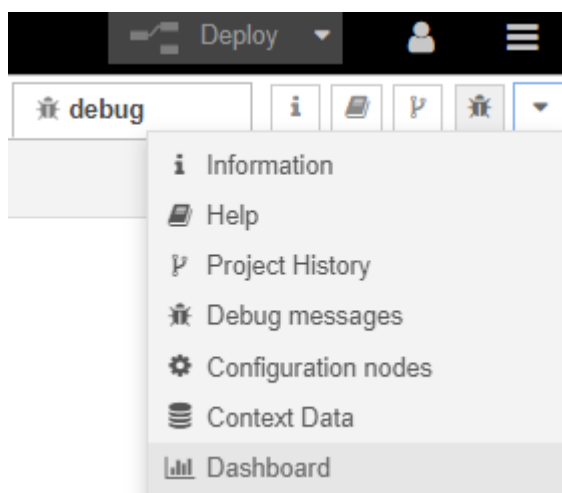
”Lämpötila keittiössä”-nodejen nimeksi vaihdettiin ”lämpötila makkarissa”. Muiden dashboard-nodejen ryhmät vaihdettiin tähän juuri luotuun uuteen ryhmään ja lisäksi ”keittiö”-funktion nimeksi vaihdettiin ”makkari”.

4.7 Remote-RED-lisäosan asennus

Seuraava vaihe oli saada Node-RED:n dashboard-näkymä mobiililaitteeseen. Tähän käytettiin Node-RED:n lisäosaa nimeltä Remote-RED, jonka käyttöönotto Node-RED:ssä aloitettiin lataamalla ”node-red-contrib-remote”-lisäosa. Lisäksi ladattiin mobiililaitteeseen sovelluskaupasta ”Remote-RED”-sovellus. Lisäosan latauksen jälkeen vedettiin flow-kaavioon ”remote access” -node, ja avattiin sen asetukset. Asetuksista painettiin ”connect Remote-RED App” -painiketta, joka avasi näkymään QR-koodin. Tämä luettiin mobiililaitteen Remote-RED sovelluksella, jonka jälkeen yhdistäminen oli valmis. Viimeiseksi tallennettiin vielä Node-RED:n puolelta asetukset, jonka jälkeen dashboard oli nähtävissä Remote-RED sovelluksella.

Dashboard-näkymä tuli vielä optimoida kännykän näyttöön sopivaksi. Muokkaus oli mahdollista tehdä valitsemalla kuvan 25 mukaisesti Node-RED:n sivupalkin valikosta ”dashboard”. Avautuneesta valikoista voitiin muokata luotuja välilehtiä ja ryhmiä. ”Lämmitys”-välilehden alta muokattiin ”lämmitys”-ryhmän asetuksista ryhmän leveydeksi 7, ja testattiin tämä kännykkää apuna käyttäen sopivaksi. Lisäksi muutettiin ”lämmitys”-ryhmän ”viimeisin pyynti” -noden kooksi 7 x 3. Muiden nodejen koot saivat olla asennossa ”auto”. ”Jäähdytys”-ryhmälle tehtiin vastaavat toimenpiteet.

Kuva 25. Node-RED:n sivupalkin valikko.



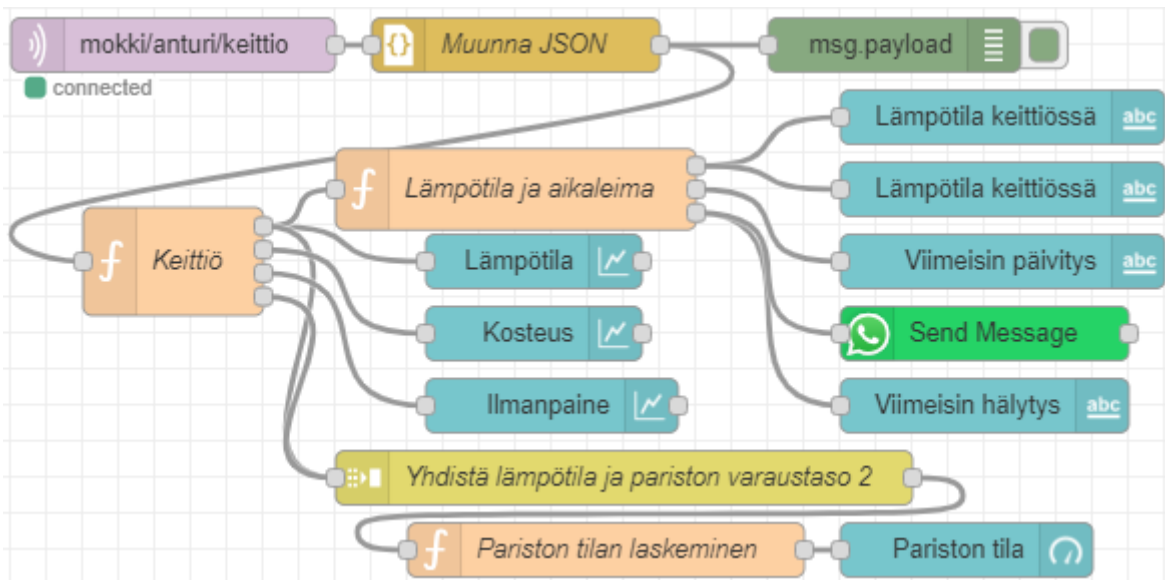
"Mittaukset"-ryhmästä muutettiin vielä sekä "keittiö"- että "makkari"-ryhmien leveyksien arvoksi 12 ja tarkistettiin toimivuus kännykällä. Tätä näkymää katsotaan mobiililaitte vaakatasossa, jotta graafeista saatiin isommat ja yksityiskohtaisemmat. "Lämpötila"-, "ilmankosteus"-, "ilmanpaine"- ja "pariston tila"-nodejen kooksi määriteltiin 12 x 5. Muiden nodejen arvot saivat olla asennossa "auto". Lisäksi muutettiin asetuksista vielä välilehdet ja ryhmät haluttuihin järjestyksiin.

4.8 Hälytysten asetus

Hälytyksiä varten ladattiin Node-RED:n lisäosa "node-red-contrib-whatsapp-cmb". Hälytysten asettaminen aloitettiin keittiön puolen flow-kaaviosta. Ensimmäiseksi muokattiin "lämpötilat ja aikaleima"-nodea, johon lisättiin kolmas ulostulo ja jonka ohjelmakoodia muokattiin lisäämällä sinne kolmas ulos lähtevä payload "messageOut". Tämän arvoksi tuli teksti, jossa määriteltiin lähetettävän WhatsApp-viestin sisältö. Ohjelmakoodissa määriteltiin hälytysten lähtevän, jos lämpötila-arvo on pienempi kuin viisi. Testatessa aloitettiin kokeilemalla arvolla 30, jotta voitiin varmistua viestien oikein lähtemisestä. Tämän vaiheen ohjelmakoodi on liitteessä 8.

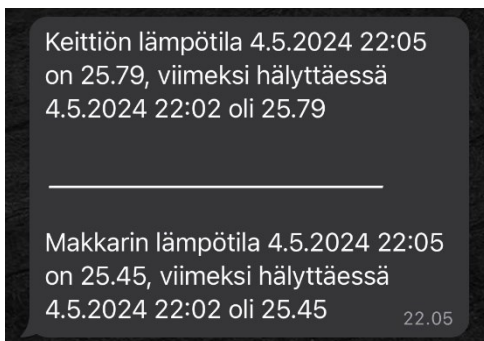
Kolmanteen ulostuloon lisättiin "send message"-node. Tämän asetuksista luotiin uusi käyttäjätunnus, johon tuli syöttää käyttäjän nimi, puhelinnumero ja "API-key". Tämän avaimen luomisen ohjeisiin löytyi linkki asetuksista. Tämä vaati ohjeistetun mallisen WhatsApp-viestin lähettämistä annettuun puhelinnumeroon, josta vastauksena saatiin API-avain. Tämä sijoitettiin paikoilleen ja muokattiin tämän jälkeen vielä "message"-kohtaan lähetettäväksi arvoksi "msg.payload". Kolmanteen ulostuloon lisättiin vielä "text"-node, jonka "label"-kohtaan annettiin tekstiksi "Viimeisin hälytys", ja arvoksi määriteltiin "{{msg.payload}}". Lisäksi ryhmäksi määriteltiin "[mittaukset] keittiö". Valmis kaavio on esitetty kuvassa 26.

Kuva 26. Keittiön RuuviTag-anturin mittaustulokset dashboardilla esittävä flow-kaavio, johon on lisätty hälytystoiminto.



Lopuksi tehtiin vastaavat muutokset ja lisäykset myös makuuhuoneen puolen flow-kaavioon, ja muokattiin ne sopimaan makuuhuoneen puolelle. Näin saatiin hälytykset tulemaan myös makuuhuoneen anturin mukaan. Testatessa saatiin kuvan 27 mukainen WhatsApp-viesti.

Kuva 27. Hälytystoiminnon testauksessa saadut WhatsApp-viestit.

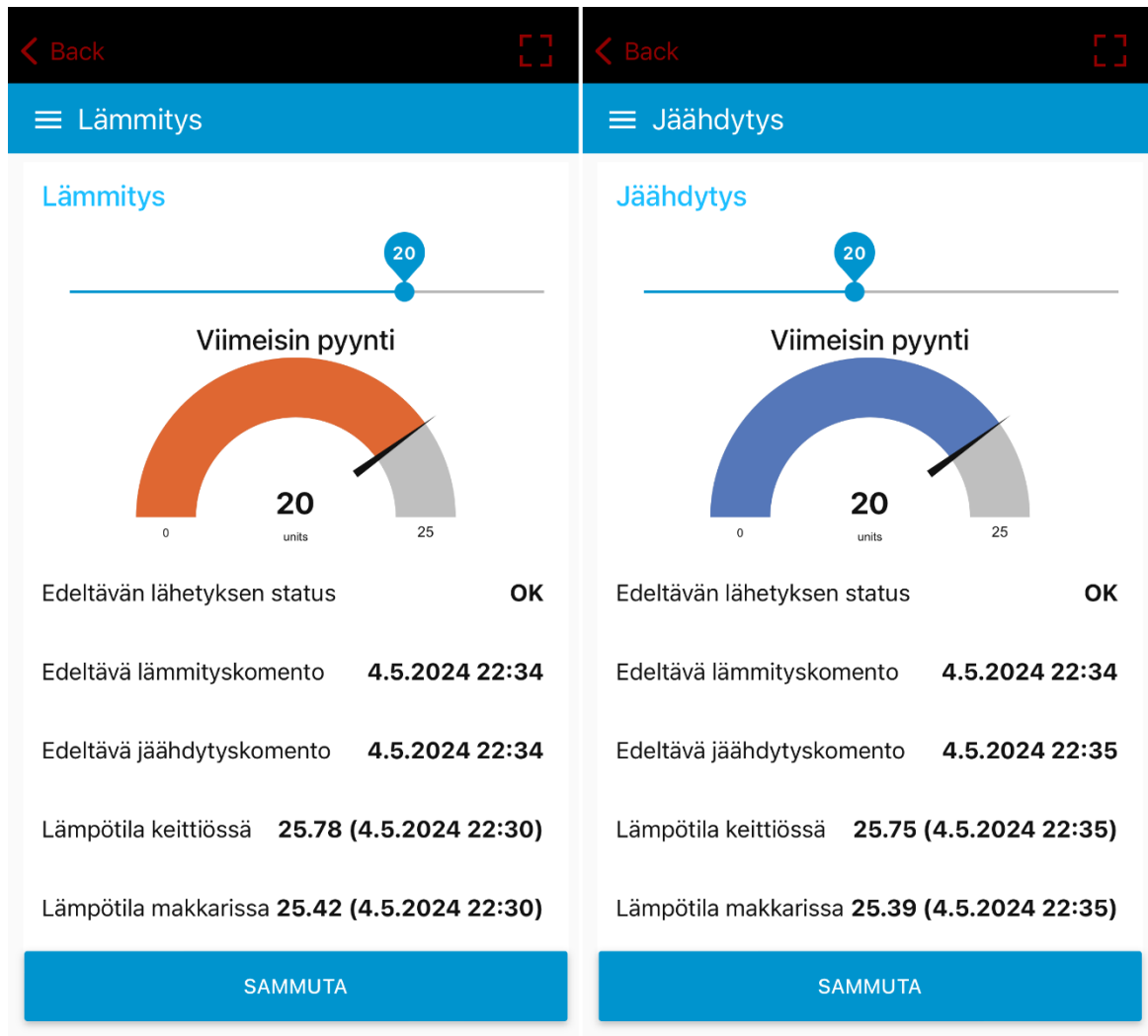


4.9 Valmis järjestelmä

Valmiissa Node-RED-kokonaisuudessa flow-kaaviot on jaettu kahdelle eri välilehdelle, jotka on nimetty "RuuviTag" ja "Broadlink RM PRO". Ensimmäisessä välilehdessä sijaitsee flow-kaaviot, jotka liittyvät antureiden lukuun ja näiden esittämiseen. Lisäksi sinne on sijoitettu Remote-RED-lisäosan tarvitsema node. Toisen välilehden flow-kaaviot liittyvät ilmalämpöpumpun etäohjaukseen. Kaavioiden kuvat ovat liitteessä 9.

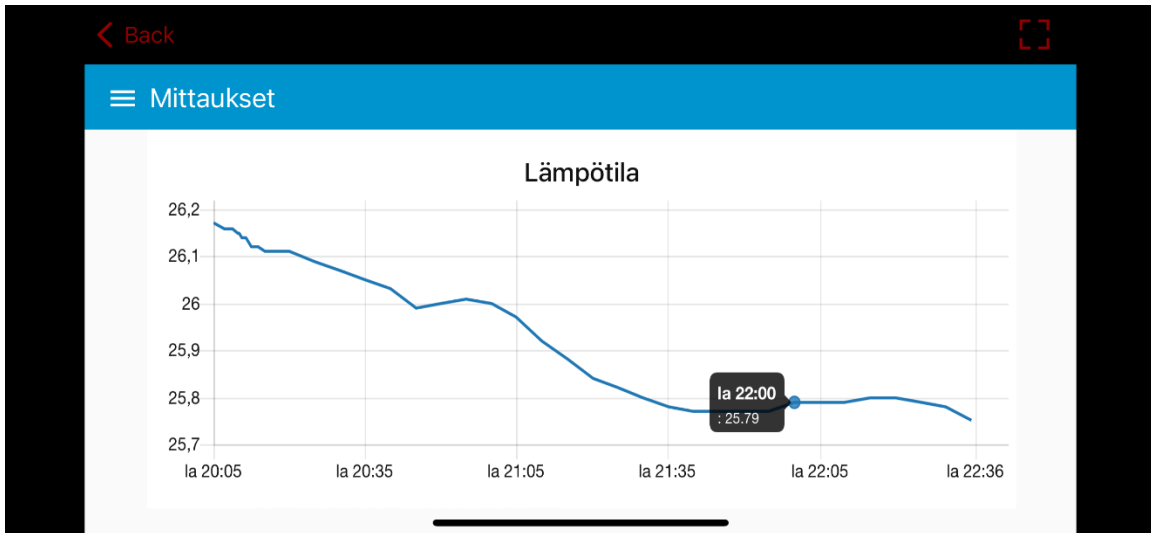
Mobiililaitteen Remote-Red-sovelluksella päästään tarkastelemaan valmiita ohjaus- ja seurantanäkymiä. Kuvassa 28 on esitetty sekä lämmitys- että jäähdytysvälilehdet. Näistä näkymistä voidaan säätää ilmalämpöpumpun lämpötilaa ja sammuttaa laite. Lisäksi voidaan tarkastella viimeisimpiä komentoja ja lämpötilojen mittaustuloksia.

Kuva 28. Lämmitys- ja jäähdytys-näkymät Remote-RED-sovelluksessa.



Kuvassa 29 on esitetty mittaukset-välilehteä. Näkymässä on kuvaajia, jotka kuvaavat sekä keittiöön että makuuhuoneeseen sijoitettujen RuuviTag-antureiden mittaustuloksia. Näkymään on sijoitettu myös paristojen varaustason näyttävä graafi ja lisäksi teksti, joka ilmoittaa, milloin viimeisin päivitys mittaustuloksiin on tehty. Eri välilehtien välillä voidaan liikkua vasemmassa yläkulmassa sijaitsevan valikon kautta.

Kuva 29. Mittaukset-näkymä Remote-RED sovelluksessa.



5 Pohdinta

Tämän toiminnallisen opinnäytetyön tarkoituksena oli perehtyä ilmalämpöpumpun etäohjausmahdollisuuksiin ja toteuttaa Node-RED-ohjelmointityökalun avulla kesämökkiin helposti laajennettavissa oleva kotiautomaatiojärjestelmä, jolla voidaan etäohjata ilmalämpöpumppua ja seurata rakennuksen lämpötilaa. Tavoitteena oli helpottaa yksityiskäytössä olevan kesämökin sisälämpötilan ohjausta ja seurantaa sekä mahdollistaa tämän avulla säästöjä. Tutkimuskysymyksinä oli selvittää, millaisia etäohjaimia ilmalämpöpumpulle on ja miten Node-RED soveltuu ilmalämpöpumpun etäohjaukseen.

Opinnäytetyöprosessin alussa perehdyttiin ilmalämpöpumpun etäohjausmahdollisuuksiin ja valittiin projektiin sopiva etäohjain. Erilaisia etäohjaimia löytyi odotettua enemmän, joten näiden markkinat olivat odotettua suuremmat. Etäohjaimia löytyi sekä opinnäytetyöhön valittuja WiFi-verkossa toimivia että GSM-ohjauksella toimivia. Broadlinkin versio valikoitui sen helpon Node-RED-integroitavuuden ja edullisen hinnan vuoksi. Koska Node-RED kommunikoi laitteen kanssa paikallisesti ilman pilvettä välissä, jää yksi iso komponentti epävarmuustekijöineen pois välistä. Broadlinkin etäohjaimen heikkoutena voidaan pitää sen epävarmaa saatavuutta kotimaisista verkkokaupoista.

Opinnäytetyön tuotoksena luotiin Node-RED-pohjainen järjestelmä, jonka avulla voidaan etäohjata ilmalämpöpumppua ja seurata rakennuksen sisälämpötiloja. Node-RED oli melko uusi työkalu opinnäytetyön tekijälle, ja sen käyttö hieman monimutkaisemmassa projektissa kiinnosti. Oletus ennen opinnäytetyön aloitusta oli, että ilmalämpöpumppua ohjaavasta kaaviosta tulisi monimutkainen. Lopullisista kaavioista tuli kuitenkin melko yksinkertaisia.

Node-RED:n osuutta oli hankala suunnitella kovin tarkasti etukäteen, koska sen soveltuvuudesta tämänkaltaiseen projektiin ei ollut tietoa. Suurimmaksi kysymykseksi ohjelmoinnin edetessä tuli se, miten käyttäjältä kannattaisi ottaa vastaan syötteitä ilmalämpöpumpun komentamiseksi. Projektia tehdessä opittiin myös se, että ilmalämpöpumpun kauko-ohjain lähettää jokaisella napin painalluksella kaikki näytöllä näkyvät komennot ilmalämpöpumpulle. Alkuperäinen oletus oli, että esimerkiksi pelkälle lämmön nostamiselle olisi oma komento. Mikäli haluttaisiin jatkokehityksenä laajentaa etäohjausmahdollisuutta myös esimerkiksi puhalluksen nopeuden suhteen, tulisi tämän toteuttamiseksi suunnitella uusi logiikka.

Käyttäjäsyytöiden vastaanotto päätettiin jakaa omille välilehdilleen siten, että lämmitys- ja jäähdytystoiminnoilla on omat välilehtensä. Tällöin käyttäjältä ei tarvita muuta syötettä kuin

haluttu lämpötila. Riskinä tässä tunnistettiin se, että käyttäjä saattaisi epähuomiossa käyttää lämmitystä ja jäähdytystä ristiin. Tämän minimoimiseksi valittiin lämmitysnäkymään punertavia sävyjä ja jäähdytysnäkömään sinisiä sävyjä. Lisäksi laitettiin molempien näkymien edeltävien komentojen aikaleimat näkömään sekä lämmitysnäkymässä että jäähdytysnäkömässä.

Hälytystoiminto oli jo suunnitteluvaiheessa päätetty toteuttaa Node-RED:n WhatsApp-lisäosan avulla. Tästä oli jo ennestään kokemusta ja lisäosa oli todettu hyvin toimivaksi. Tämä toimi myös testatessa hyvin. Opinnäytetyön edetessä ja Remote-RED-lisäosaan tutustuessa kävi ilmi, että myös siinä on olemassa maksullinen hälytystoiminto. Tämä toiminto kustantaa noin kolme euroa vuodessa, joten tämän voisi ottaa käyttöön, mikäli WhatsApp-lisäosan kanssa tulisi ongelmia. Myös tätä testattiin projektin aikana, sillä tätä ominaisuutta oli mahdollista kokeilla muutaman kerran maksuttomasti.

Node-RED:n avulla tehdyn toteutuksen etuja ovat ehdottomasti sen helppo laajennettavuus ja halvat kustannukset. Raspberry Pi Zero 2 W vaikuttaisi jaksavan hyvin pyörittää Node-RED:ia, ja sen ostohinta on vain noin 30 euroa. Tämän projektin hintaa nostavat RuuviTag-anturit, joista on maksettu yhteensä noin 60 euroa. Nämä anturit olisi varmasti mahdollista korvata halvemminkin laitteilla; esimerkiksi toisen voisi helposti korvata halvalla suoraan Raspberry Pi:hin kytketyllä anturilla. Tämä voisi olla hyvä jatkokehitysidea. Lisäksi projektin kustannuksiin kuului Broadlink RM PRO-etäohjain, jolla oli hintaa noin 50 euroa. Tällöin koko projektin tarvikkeiden hinnaksi tuli noin 140 euroa.

Ilmalämpöpumppeihin tutustuessa kävi ilmi, ettei laitetta kannattaisi pitää rakennuksen ainoana lämmityslaitteena. Tämä koettiin myös käytännössä kohteessa, kun viime talvena luotettiin yksin ilmalämpöpumppuun ja kohteen astianpesukone ja veden läpivirtauslämmitin jäätyivät rikki. Edeltävän talven ilmalämpöpumpun häiriöille ei löytynyt selkeää syytä, mutta jatkossa hälytysjärjestelmän ansiosta kesämökille pystytään lähtemään välittömästi selvittämään ongelmaa ilmalämpöpumpun toiminnassa.

Kehitysideana varalämmönlähteeksi voitaisiin hankkia sähkökäyttöinen öljytäytteinen lämmityspatteri. Tämä pitäisi kesämökin lämpötilan nollan yläpuolella, vaikka ilmalämpöpumpun toiminta lakkaisi tai heikkenisi. Patteri voitaisiin kytkeä WiFi-pistorasiaan, joka voitaisiin yhdistää rakennettuun Node-RED-ympäristöön. Tällöin automaatio voisi kytkeä patterin päälle aina silloin, kun järjestelmä lähettää hälytysviestin WhatsAppiin. Vaihtoehtoisesti tai lisäksi dashboard-näkymään voitaisiin tehdä painike patterin käynnistämiseksi ja sammuttamiseksi.

Projektissa kiinnitettiin huomiota tietoturvaan perustasolla. Raspberry Pi:ssä käytettiin dokumentaatiota varten oletustunnusta ”pi”, mutta tämä vaihdettiin lopuksi, sillä oletustunnuksia ja -salasanoja ei tule koskaan käyttää. Node-RED:iin lisättiin tietoturvan takia perinteinen autentikointi, joten päästäkseen palveluun käyttäjän tulee syöttää tunnus ja salasana. MQTT-sanomien lähetyksen mukana pitää syöttää kirjautumistiedot ja lisäksi kohteen langaton verkkoyhteys on salasanasuojattu.

Tietoturvan kannalta suurimmaksi uhaksi muodostui Remote-RED-lisäosa, joka avaa yhteyden internettiin. Remote-RED:n sivuilla mainitaan palvelun pyrkivän saavuttamaan korkean tietoturvatason, mutta samalla varoitettiin asettamasta saavutettavaksi kriittisiä järjestelmiä, joita ovat esimerkiksi oven avaaminen, kamera tai mikrofoni. Aina, kun jokin järjestelmä on yhteydessä internettiin, muodostuu tietoturvariski. Mikäli tähän järjestelmään haluttaisiin tuoda esimerkiksi valvontakamera, tulisi tietoturvaa miettiä erittäin kriittisesti.

Opinnäytetyöprosessin lopussa järjestelmä siirrettiin kohdekesämökille. Järjestelmän kasaaminen ja konfigurointi oli suoritettu jo valmiiksi käyttäen kesämökin mobiiliverkkoa, joten tämän suhteen siirto ei tuonut muutoksia. Siirron jälkeen huomattiin kuitenkin etäohjaimen lakanneen toimimasta. Broadlinkin laitteita verkosta hakevan flow-kaavion avulla saatiin selville, että laitteen IP-osoite oli vaihtunut. Uusi osoite korjattiin RM-noden asetuksista ja tämän jälkeen laite alkoi taas toimia. Mahdollinen sähkökatkos voi aiheuttaa IP-osoitteen vaihtumisen, joten jatkokehityksenä tulisi selvittää ratkaisu tähän ongelmaan.

Ilmalämpöpumpun komennot oli opetettu ja kehitysprosessin aikaiset testaukset oli tehty täysin kohteen ilmalämpöpumppua vastaavalla laitteella. Siirron jälkeen huomattiin, että komennot eivät kuitenkaan vastanneet toisiaan. Sammutus-komento toimi oikein ja ilmalämpöpumppu antoi merkkiäänän myös muita komentoja antaessa, mutta näiden oikeellisuudesta ei voitu olla varmoja. Tästä syystä komennot opetettiin uudestaan. Tämä onnistui varsin nopeasti opetus-flow-kaavion avulla. Komentojen uudelleenopetteluun jälkeen järjestelmä vaikutti toimivan oikein.

Node-RED-ohjelmointityökalulla luotiin ilmalämpöpumppua etäohjaava järjestelmä. Opinnäytetyöprosessin aikaisen kokemuksen perusteella Node-RED vaikuttaisi soveltuvan hyvin varsinkin yksinkertaiseen ilmalämpöpumpun etäohjaukseen. Komentojen opettelu ja lähettäminen onnistuu helposti, mutta käyttäjäsyötteiden vastaanottaminen vaatii kehittämistä, mikäli ilmalämpöpumpun puhallusnopeutta ja -kulmaa halutaan ohjata.

Lähteet

- AirPatrol. (n.d.). *AIRPATROL WIFI V5 EU*. <https://airpatrol.com/fi/products/airpatrol-wifi-smart-and-universal-air-conditioner-controller>
- BroadLink Finland. (n.d.). *Hallitse laitteitasi läheltä ja kaukaa*. <http://www.broadlink.fi/page7/>
- Chege, J. (19.2.2024). *How to Install and Run Node-Red on Raspberry Pi StepByStep*. <https://thrivead.com/install-and-run-node-red-on-raspberry-pi>
- Eclipse Foundation. (n.d.-a). *Eclipse Mosquitto™*. <https://mosquitto.org/>
- Eclipse Foundation. (n.d.-b). *Paho*. <https://wiki.eclipse.org/Paho>
- Energy Center Finland Oy. (n.d.-a). *Ilmalämpöpumpun kondenssivesi – Näin estät vuotovahingot tehokkaasti*. <https://energycenter.fi/ilmalampopumppu/ilmalampopumpun-kondenssivesi/>
- Energy Center Finland Oy. (n.d.-b). *Ilmalämpöpumpun yleisimmät viat ja niiden korjaaminen*. <https://energycenter.fi/vinkit/yleisimmat-vikatilat-ilmalampopumpussa-ja-vinkeja-niiden-korjaamiseen/>
- Heilman, T. (n.d.-a). *REMOTE-RED*. <https://www.remote-red.com>
- Heilman, T. (n.d.-b). *Terms of use of Remote-RED*. <https://www.remote-red.com/en/terms/>
- HiveMQ. (15.2.2024). *MQTT Tutorial: An Easy Guide to Getting Started with MQTT*. <https://www.hivemq.com/blog/how-to-get-started-with-mqtt/#heading-mqtt-tutorial-an-easy-guide-to-getting-started-with-mqtt>
- Kärkkäinen. (n.d.). *Broadlink RM Pro Wi-Fi yleiskauko-ohjain*. Haettu 18.4.2024 osoitteesta <https://www.karkkainen.com/verkkokauppa/broadlink-rm-pro-wi-fi-yleiskauko-ohjain>
- Motiva. (21.12.2022). *Ilmalämpöpumppu (ILP)*. https://www.motiva.fi/ratkaisut/uusiutuva_energia/lampopumput/lampopumpputeknologia/ilmalampopumppu
- MQTT.org. (n.d.) *MQTT: The Standard for IoT Messaging*. <https://mqtt.org/>
- Mr.LVI. (n.d.) *PANASONIC CZ-TACG1 ILMALÄMPÖPUMPUN WIFI ETÄOHJAIN*. Haettu 18.4.2024 osoitteesta <https://www.xn--lmpumppuhuolto-0kb22a.com/index.php/ilmalaempoepumput/lisatarvikkeet-ilmalampopumput/etaohjaimet/panasonic-smart-cloud-cz-tacg1-wifi-ohjain-detail>
- Nedis. (n.d.). *SmartLife IR Kaukosäädin*. Haettu 24.4.2024 osoitteesta <https://nedis.fi/fi/product/alykoti/ohjauslaitteet/kaukosaatimet/550695156/smartlife-ir-kaukosaaedin-wi-fi-yleismalli-signaalin-kantama-5-m-38-khz-usb-virta-android-ios-musta>
- OpenJS Foundation. (n.d.-a). *About*. <https://nodered.org/about/>
- OpenJS Foundation. (n.d.-b). *Editor Guide*. <https://nodered.org/docs/user-guide/editor/>
- OpenJS Foundation. (n.d.-c). *node-red-contrib-broadlink-control 2.1.6*. <https://flows.nodered.org/node/node-red-contrib-broadlink-control>

- OpenJS Foundation. (n.d.-d). *node-red-contrib-panasonic-comfort-cloud 1.5.1*.
<https://flows.nodered.org/node/node-red-contrib-panasonic-comfort-cloud>
- OpenJS Foundation. (n.d.-e). *node-red-contrib-sensibo 0.6.1*.
<https://flows.nodered.org/node/node-red-contrib-sensibo>
- OpenJS Foundation. (n.d.-f). *node-red-dashboard 3.6.5*.
<https://flows.nodered.org/node/node-red-dashboard>
- Opensource. (n.d.). *What is a Raspberry Pi?* Haettu 16.4.2024 osoitteesta
<https://opensource.com/resources/raspberry-pi>
- Panasonic. (n.d.) CZ-TACG1 [kuva] <https://www.panasonic.com/my/air-solutions/products/control-system/accessory/cz-tacg1.html>
- Power. (n.d.) *SENSIBO SKY ILMASTOINTILAITTEEN KAUKO-OHJAIN, VALKOINEN SEN-SKY-01-WH*. Haettu 20.4.2024 osoitteesta <https://www.power.fi/smart-home-alykoti/alykodin-sisailma/alytermostaatit/sensibo-sky-ilmastointilaitteen-kauko-ohjain-valkoinen-sen-sky-01-wh/p-1065872>
- Raspberry Pi Foundation. (n.d.). *Raspberry Pi Zero 2 W*.
<https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>
- Raspberry Pi Foundation. (2021). *Raspberry Pi Zero 2 W*.
<https://datasheets.raspberrypi.com/rpizero2/raspberry-pi-zero-2-w-product-brief.pdf>
- Ruuvi Innovations Oy. (n.d.-a). *Langaton lämpötila-, kosteus-, ilmanpaine- ja liikeanturi*.
<https://ruuvi.com/fi/ruuvitag/>
- Ruuvi Innovations Oy. (n.d.-b). *Tietoja paristoista ja RuuviTagin pariston vaihdosta*.
<https://ruuvi.com/fi/ruuvitag-paristo-vaihto/>
- Sensibo. (n.d.). *Sensibo Sky*. Haettu 20.4.2024 osoitteesta
<https://sensibo.com/products/sensibo-sky>
- Verkkokauppa.com Oyj. (n.d.) *Raspberry Pi Zero 2W - yhden piirilevyn tietokone*. Haettu 20.4.2024 osoitteesta <https://www.verkkokauppa.com/fi/product/760951/Raspberry-Pi-Zero-2W-yhden-piirilevyn-tietokone>

Liite 1. Tiedoston getAndSendRuuviDatas.py sisältö

```
import paho.mqtt.client as mqtt
from ruuvitag_sensor.ruuvi import RuuviTagSensor, RunFlag

# Yhdistetään paikalliselle MQTT-brokerille
mqttBroker = "localhost"
client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2, clean_session=True)
client.username_pw_set(username="XXX", password="XXX")
client.connect(mqttBroker, port=1883)

# Lähetetään anturilta saatu data julkaistavaksi
def handle_data(found_data):
    julkaise(found_data[1])
    # Pysäytetään handle_data-funktio
    run_flag.running = False

# Formatoidaan anturilta saatu data ja julkaistaan se MQTT:llä haluttuun topiciin
def julkaise(sensorValue):
    strSensorValue = str(sensorValue)
    sensorValueFormatted = strSensorValue.replace("\\", "\\")
    client.publish(topicName, sensorValueFormatted)

# Määritellään RunFlag suorituksen pysäyttämistä varten
run_flag = RunFlag()

# Haetaan ja lähetetään ensimmäisen anturin data
mac = ["FD:XX:XX:XX:XX:XX"]
topicName = "mokki/anturi/keittio"
RuuviTagSensor.get_data(handle_data, mac, run_flag)

# Haetaan ja lähetetään toisen anturin data
run_flag.running = True
mac = ["F5:XX:XX:XX:XX:XX"]
topicName = "mokki/anturi/makuuhuone"
RuuviTagSensor.get_data(handle_data, mac, run_flag)

# Kirjaututaan ulos paikalliselta MQTT-brokerilta
client.disconnect()
```

Liite 2. JavaScript ohjelmakoodi etäohjaimelle lähetettävän sanoman luomiseen.

```
var data = msg.payload;
```

```
msg.payload = {  
  "action": "send",  
  "data": data,  
};
```

```
return msg;
```

Liite 3. JavaScript ohjelmakoodi tiedoston polun luomiseen.

```
var value = msg.payload;

if (value == "off") {
  msg.payload = "/home/pi/komennot/off.txt"
  return msg;
}

msg.payload = `/home/pi/komennot/ht${value}s4f5.txt`

return msg;
```

Liite 4. JavaScript ohjelmakoodi aikaleiman luomiseen.

```
var today = new Date();
var date = today.getDate() + '.' + (today.getMonth() + 1) + '.' + today.getFullYear();
var minutes = today.getMinutes();
if (minutes < 10) { minutes = "0" + minutes };
var time = today.getHours() + ":" + minutes;
var dateTime = date + ' ' + time;
var msgOut = {payload:dateTime};

return msgOut;
```

Liite 5. JavaScript ohjelmakoodi JavaScript-objektina saatujen datojen jakamiseen omiin ulostuloihin.

```
var temperature = msg.payload.temperature;  
var humidity = msg.payload.humidity;  
var pressure = msg.payload.pressure;  
var battery = msg.payload.battery;
```

```
var msgOut = [ "", "", "", "" ];
```

```
msgOut[0] = {payload:temperature};  
msgOut[1] = {payload:humidity};  
msgOut[2] = {payload:pressure};  
msgOut[3] = {payload:battery};
```

```
return msgOut;
```

Liite 6. JavaScript ohjelmakoodi RuuviTag-anturin pariston varaustason laskemiseen.

```
var temp = msg.payload[0];
var batteryValue = msg.payload[1];
var minValue = 2500

if (temp < -20) {
  minValue = 2000;
}
else if (temp <= 0) {
  minValue = 2300;
}

var maxValue = 3000;

if (batteryValue > 3000) {
  batteryValue = 3000
}

var combineValue = maxValue - minValue;

var batteryPerc = Math.round(((batteryValue-minValue)/combineValue)*100);

msg.payload = batteryPerc;

return msg;
```

Liite 7. JavaScript ohjelmakoodi aikaleiman sekä lämpötila-arvon ja aikaleiman luomiseen.

```
// Aikaleiman luonti
var today = new Date();
var date = today.getDate() + '.' + (today.getMonth() + 1) + '.' + today.getFullYear();
var minutes = today.getMinutes();
if (minutes < 10) { minutes = "0" + minutes };
var time = today.getHours() + ":" + minutes;
var dateTime = date + ' ' + time;

// Lämpötilan asetus
var temp = msg.payload;

// Lämpötila + aikaleima
var tempDateOut = `${temp} (${dateTime})`;
// Aikaleima
var dateOut = dateTime;

var msgOut = ["", ""];

msgOut[0] = {payload:tempDateOut};
msgOut[1] = {payload:dateOut};

return msgOut;
```

Liite 8. JavaScript ohjelmakoodi aikaleiman, lämpötila-arvon ja aikaleiman sekä hälytysten luomiseen.

```
// Aikaleiman luonti
var today = new Date();
var date = today.getDate() + '.' + (today.getMonth() + 1) + '.' + today.getFullYear();
var minutes = today.getMinutes();
if (minutes < 10) { minutes = "0" + minutes };
var time = today.getHours() + ":" + minutes;
var dateTime = date + ' ' + time;

// Lämpötilan asetus
var temp = msg.payload;

// Hälytyksen asetus
var messageOut;

var counter = context.get('keittioTemp');
var timestamp = context.get('keittioTimestamp');
if (typeof counter == "undefined") {
    counter = 0;
}
if (typeof timestamp == "undefined") {
    timestamp = "(ei edeltävää aikaleimaa)";
}

if (temp < 5 ) {
    context.set("keittioTemp", temp);
    context.set("keittioTimestamp", dateTime);
    messageOut = `Keittiön lämpötila ${dateTime} on ${temp}, viimeksi hälyttäessä
    ${timestamp} oli ${counter}`;
}

// Lämpötila + aikaleima
var tempDateOut = `${temp} (${dateTime})`;
// Aikaleima
var dateOut = dateTime;

var msgOut = [ "", "", "" ];

msgOut[0] = {payload:tempDateOut};
msgOut[1] = {payload:dateOut};
msgOut[2] = {payload:messageOut}

return msgOut;
```

Liite 9. Kuvat valmiista Node-RED:n flow-kaavioista.

