

Maalämpöpumpun monitorointi ja etäohjaus

Janne Latomäki

Opinnäytetyö
Joulukuu 2019
Insinööri (AMK), tieto- ja viestintätekniikka
Tietoverkkoturvallisuus ja teollinen internet

Tekijä(t) Latomäki, Janne	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Joulukuu 2019
	Sivumäärä 47	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Maalämpöpumpun monitorointi ja etäohjaus Mahdollinen alanimi		
Tutkinto-ohjelma Insinööri (AMK), tieto- ja viestintätekniikka		
Työn ohjaaja(t) Olli Väänänen		
Toimeksiantaja(t) Janne Latomäki		
<p>Tiivistelmä</p> <p>Opinnäytetyössä tutkittiin etämonitoroinnin ja mahdollisen etäohjauksen toteutusmahdollisuutta omakotitalon lämmityslähteenä käytettyyn maalämpöpumppuun. Kyseinen pumppu oli ikäluokaltaan sellainen, ettei siinä ollut valmiina mitään rajapintaa pumpun ulkopuoliseen maailmaan, mutta siinä oli valmiudet lisätä erityyppisiä sähköisiä rajapintoja ulkopuolisiin järjestelmiin.</p> <p>Tutkimustyössä tietoturvanäkökulma oli tärkeässä roolissa. Tietoturva piti varmistaa sellaiselle tasolle, että mahdollisten tietomurtojen tai muiden vihamielisten hyökkäysten tapahtuessa, lämpöpumpulle tai kiinteistölle ei tapahdu fyysistä vahinkoa.</p> <p>Tutkimuksen alussa hieman kartoitettiin olemassa olevia valmiita kaupallisia vaihtoehtoja sekä ilmaisia open source -ratkaisuja.</p> <p>Monitorointi toteutettiin asentamalla maalämpöpumpun ohjainlogiikkaan Modbus-liityntä, jonka kautta voitiin lukea mittaustietoja maalämpöpumpusta. Tietojen lukeminen toteutettiin virtualisoidulla Linux-palvelimella, joka sijaitsi lämpöpumpun kanssa samassa tilassa. Tietojen lukeminen tapahtui Python-ohjelmointikielellä toteutetulla ohjelmistolla. Ohjelmisto luki määrätyin aikavälein mittaus-, status- ja hälytystiedot maalämpöpumpusta ja lähetti ne salattua WebSocket-yhteyttä hyväksikäyttäen pilvipalvelimelle. Pilvipalvelimella ajossa ollut Python-palvelin vastaanotti tiedot ja talletti ne SQL-tietokantaan. Lisäksi tiedot lähetettiin reaaliajassa mahdollisille web-asiakkaille, jotka olivat tietojen vastaanottohetkellä palvelimella.</p> <p>Lopputuloksena oli yksinkertainen, kevyt ja käytettävä monitorointijärjestelmä, jonka web-käyttöliittymästä näki maalämpöpumpun tilan.</p>		
Avainsanat (asiasanat) Maalämpöpumppu, Modbus, Python, JSON, WebSocket, HTML5, Javascript, PHP, etämonitorointi		
Muut tiedot		

Author(s) Latomäki, Janne	Type of publication Bachelor's thesis	Date December 2019 Language of publication: Finnish
	Number of pages 47	Permission for web publication: x
Title of publication Monitoring system for geothermal heat pump		
Degree programme Information and Communication Technology		
Supervisor(s) Olli Väänänen		
Assigned by Janne Latomäki		
Abstract <p>The thesis examined possibilities to implement a remote monitoring system for a geo-thermal heat pump located in a detached house. The heat pump of the house was obsolete and did not include any interfaces to external systems.</p> <p>Information security had an important role in the project. The level of cyber security had to be so high that there were no possibilities to cause any physical damage to the house or the geothermal heat pump.</p> <p>Some commercial and open source monitoring implementations were investigated at the beginning of the study. The base of the monitoring systems was implemented by installing a Modbus interface card on the control logic of the geothermal heat pump, which was needed when the data was read from the heat pump. The software for reading data was implemented with a virtualized Linux server located in the same room as the geo-thermal heat pump. Data was read with software implemented with Python programming language. The software read the data concerning measuring, status and alarms from the heat pump and sent the data to the cloud server via an encrypted WebSocket connection. Python software on the cloud server received the data and saved them to the SQL database. The receiver software of the cloud server also sent the data to the connected web clients in real time.</p> <p>The thesis resulted in a quite simple, light and usable monitoring system in which the user can see the status of the geothermal heat pump on the web user interface.</p>		
Keywords/tags (subjects) Geothermal heat pump, Modbus, JSON, WebSocket, Java script, PHP, remote monitoring		
Miscellaneous (Confidential information)		

Sisältö

Lyhenteet ja termit	4
1 Työn lähtökohdat	5
1.1 Tavoitteet	5
1.2 Kohde ja nykytila	5
1.3 Tavoitetila	7
1.3.1 Monitorointi	7
1.3.2 Etäohjaus	8
1.4 Valmiit ratkaisut	9
2 Maalämpö	10
2.1 Yleistä	10
2.2 Maalämpöpumppu	11
3 Käytettävät teknologiat	13
3.1 Toteutuksen vaihtoehdot	13
3.2 Lämpöpumpun ulkoinen liityntä	14
3.2.1 EH-net	14
3.2.2 GSM-control	14
3.2.3 Modbus	14
3.2.4 LON	16
3.3 Käyttöjärjestelmä	16
3.4 Ohjelmistokomponentit	17
3.4.1 Back end -ohjelmisto	17
3.4.2 Front end -ohjelmisto	18
3.4.3 Tietokanta	18
4 Järjestelmän arkkitehtuuri	19
4.1 Järjestelmän kokonaiskuvaus	19
4.2 Tiedonkeruuosion komponentit	20
4.2.1 Lämpöpumppu	21
4.2.2 Palvelin	23
4.2.3 Ohjelmisto	23

	2
4.3 Pilviosion komponentit.....	24
4.3.1 Palvelin.....	24
4.3.2 Ohjelmisto	24
5 Ohjelmisto.....	24
5.1 Tiedonkeruu	24
5.2 Pilvipalvelu.....	29
5.3 Verkkosivusto	32
6 Testaus ja vikatilanteista toipuminen.....	35
6.1 Vikatilanteet	35
6.2 Palvelinpää	35
6.2.1 Pilvipalvelimen uudelleenkäynnistys.....	36
6.2.2 Tiedonkeruupalvelimen uudelleenkäynnistys.....	36
6.2.3 Tietoliikennekatkos.....	36
6.2.4 Tietokanta.....	36
6.2.5 Tietomurtoyritykset.....	36
6.3 Tiedonkeruupää	37
6.3.1 Tiedonkeruupalvelimen uudelleenkäynnistys.....	37
6.3.2 Pilvipalvelimen uudelleenkäynnistys.....	37
6.3.3 Lämpöpumpun vikatilanne.....	37
6.4 Web-käyttöliittymä	38
6.4.1 Toimivuus	38
6.4.2 Useat sessiot.....	38
7 Yhteenveto.....	38
7.1 Tavoitteet ja tulokset	38
7.2 Jatkokehitysideoita.....	39
Lähteet	41
Liitteet.....	43
Liite 1. Ouman 203GT yleiskytkentäkaavio	43
Liite 2. Tietokannan SQL-malli.....	44

Kuviot

Kuvio 1. Lämpöässä V13	6
Kuvio 2. Ouman 203GT	6
Kuvio 3. Lämpöpumppumäärät Suomessa	11
Kuvio 4. Lämpöässä V13 prosessikaavio.....	12
Kuvio 5. Modbus RTU, RS485 väylä	15
Kuvio 6. Modbus kehysrakenne.....	15
Kuvio 7. Järjestelmän arkkitehtuuri.....	19
Kuvio 8. Järjestelmän verkkokuva	20
Kuvio 9. Ouman EH-200 Modbus-kortti	21
Kuvio 10. Modbus -kortin DIP-kytkin.....	22
Kuvio 11. USB-RS485 -muunnin.....	22
Kuvio 12. HP ProLiant Microserver G8	23
Kuvio 13. Tiedonkeruuosion toimintaperiaate.....	25
Kuvio 14. ReaderV13.....	26
Kuvio 15. Lukufunktio	27
Kuvio 16. Mittauksien tietorakenne	28
Kuvio 17. JSON paketti sanomajonoon.....	28
Kuvio 18. Sanomajonojen hallinta	29
Kuvio 19. Pilvipalveluosion toimintaperiaate	30
Kuvio 20. WebSocket kuuntelijoitten alustus.....	31
Kuvio 21. on_message-funktio	31
Kuvio 22. Mittausdatapaketin kulku.....	32
Kuvio 23. Lämpömittarin luonti.....	33
Kuvio 24. Verkkosivuston mittaussivu.....	34
Kuvio 25. Verkkosivuston asetussivu.....	35

Lyhenteet ja termit

CAT6	Tietoliikennekaapelityyppi
COP	Coefficient of performance. Lämpöpumpun lämpökerroin
CPU	Central processing unit, suoritin
DIP-kytkin	Dual in line. Kytkintyyppi heikkovirtasovelluksissa
Ethernet	Lähiverkko
HTTPS	Hypertext Transfer Protocol Secure. WWW-sivujen siirto-protokolla
IoT	Internet of things
IP	Internet protocol
JS	Javascript -ohjelmointikieli
JSON	Javascript object notation, tietoformaatti
LON	Local Operating Network
MLP	Maalämpöpumppu
Modbus	Tiedonsiirtoprotokolla
Modbus RTU	Modbus -protokollan sarjaliikenneversio
MQTT	Kevyt tiedonsiirtoprotokolla
Python	Ohjelmointikieli
RS485	Sarjaliikenneväylä
SMS	Lyhytsanoma, tekstiviesti
SQL	Kyselykieli tietokannoille
SULPU	Suomen Lämpöpumppuyhdistys ry
VPN	Virtual private network
WebSocket	Tiedonsiirtoprotokolla
WS	Lyhenne, WebSocket
WSS	Lyhenne, Secure WebSocket

1 Työn lähtökohdat

1.1 Tavoitteet

Työssä tutkittiin mahdollisuuksia toteuttaa nykyaikainen, pilvipalveluun pohjautuva etämonitorointi ja -ohjaus maalämpöpumppuun, jossa normaalisti on pelkästään lämpöpumpun paikallinen ohjauspaneeli.

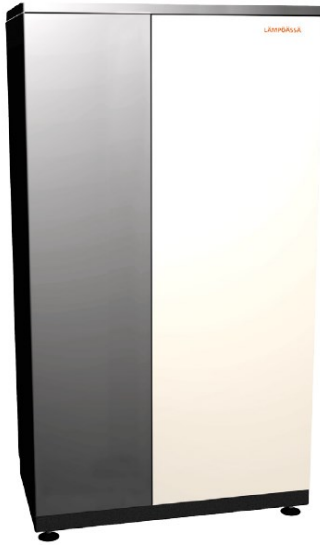
Tavoitteena oli toteuttaa etämonitorointiratkaisu tai sen runko, jota on helppo laajentaa myöhemmin uusilla ominaisuuksilla tai lisäämällä siihen kokonaan uusia laitteita kuten ilmanvaihtokone tai erillisiä mittaustietoja muista järjestelmistä.

Tietoturva tuli olemaan olennaisena osana toteutusta, ja mahdollisen tietomurron tapahtuessa järjestelmällä ei voitaisi aiheuttaa fyysistä vahinkoa kiinteistölle tai oheislaitteille.

1.2 Kohde ja nykytila

Kohteena oli Pirkanmaalla oleva, vuonna 2008 rakennettu puurunkoinen, tiiliverhottu omakotitalo, jossa on vesikiertoinen lattialämmitys. Kiinteistöön kuuluu erillinen talousrakennus, jonka lämmitysmuotona on vesikiertoinen lattialämmitys. Talousrakennuksen ja omakotitalon välissä on lämpökanaali.

Kohteen lämmityslähteenä on Lämpöässä V13 maalämpöpumppu (ks. Kuvio 1).



Kuvio 1. Lämpöässä V13 (Lämpöässä V7.0-V30.0 käyttöohje)

Lämpöässä V13 on Suomessa valmistettu maalämpöpumppu, kyseisessä mallissa käytetään suomalaisen Ouman Oy:n ohjauslogiikkaa EH-203GT, joka on Lämpöässälle räätälöity malli Ouman EH-203 -mallista (ks. Kuvio 2).



Kuvio 2. Ouman 203GT (203GT Lämmönsäädin, käsikirja)

Ohjainlogiikka sisältää pienen näppäimistön ja LCD-näytön. Käyttöliittymän rakenne on puumainen valikko, jossa liikutaan näppäimistön nuolipainikkeilla. Ohjaimen voidaan vakiona liittää muutamia ulkoisia antureita tai ohjata digitaalilähtöjä. Ohjaimessa on liitännävalmius GSM-modeemille ja erilaisille väyläkorteille. Logiikan yleiskytkentäkaavio on esitetty liitteessä 1.

Maalämpöpumppu on sijoitettuna erilliseen tekniseen tilaan, jonne on kulku talon ulkopuolelta erillisestä ovesta. Tämänkaltainen ratkaisu on nykytaloissa varsin yleinen, mutta siinä on haittapuolena teknisten laitteiden ja järjestelmien valvonnan hankaluus ja lämpöpumpun tilan seuraaminen ilman etävalvontaa on vaivalloista. Ainoat mahdollisuudet huomata mahdollinen vikatilanne on se, että lämpimän veden tuotto tuntuu huonolta tai sitten seinän läpi kuuluu pieni vaimea piippaus jonkin hälytyksen merkiksi.

1.3 Tavoitetila

Työn ensisijaisena tavoitteena oli toteuttaa järjestelmä, jolla lämpöpumpun mittaus- ja hälytysdata tallennetaan tietokantaan ja sitä voidaan monitoroida internetissä olevasta pilvipalvelusta verkkoselainta apuna käyttäen. Kyseisessä kohteessa käytännön toteutus valvontajärjestelmästä tulisi olla sellainen, että valvonta- ja mittausdatan tulisi nähdä talon pukuhuonetilassa olevasta Tablet-tietokoneesta tai matkapuhelimesta. Toissijainen tavoite oli lämpöpumpun säätöparametrien muuttaminen pilvipalvelusta. Mittausdatan historiatietojen visualisointiin ei tässä työssä keskitytty.

Järjestelmän työnimeksi tuli varhaisessa vaiheessa Homecontrol.

1.3.1 Monitorointi

Maalämpölämmitysjärjestelmän tilasta saa hyvän kokonaiskuvan seuraamalla muutamia lämpötiloja ja tilatietoja, joita maalämpöpumpussa on. Tärkeimmät seurattavat tiedot ovat

- Ulkolämpötila
- Lämminvesivaraajan yläosan lämpötila

- Lämminvesivaraajan alaosan lämpötila
- Kuumakaasun lämpötila
- Lattialämmityspiirin (L1) menoveden lämpötila
- Lattialämmityspiirin (L2) menoveden lämpötila
- Kompressorin käyntitieto
- Sähkövastuksen päälläolotieto
- Hälytykset

Edellä mainitut tiedot saadaan luettua suoraan lämpöpumpun ohjauslogiikalta sarjaliikennettä, LON-rajapintaa tai Modbus-rajapintaa apuna käyttäen.

Kaikki lämpöpumpusta luettu tieto tallennetaan myöhempää käyttöä varten. Tietojen jakaminen erityyppisiin palveluihin tai tietokantoihin, esimerkiksi MQTT-broker, tulisi olla helposti järjestettävissä. Järjestelmän tulee tuottaa tarvittaessa tarkkaa lokitietoa.

1.3.2 Etäohjaus

Työn suunnitteluvaiheessa oli tavoitteena tehdä lämpöpumppuun kaiken kattava pilvipalvelusta tehtävä etäohjaus. Tästä kuitenkin luovuttiin ensisijaisesti tietoturvasyistä mutta myös siksi, että lämpöpumpun säätöparametreihin ei juurikaan tarvitse puuttua. Lämpöpumppuun tarvitsee koskea käytännössä vain kaksi kertaa vuodessa, keväällä ja syksyllä, ja tuolloinkin muutetaan vain lattialämmityksen kiertopumppujen nopeutta, se tehdään mekaanista kytkintä kääntämällä eikä ole toteutettavissa etäohjauksella käytettävissä olevilla kiertopumppumalleilla. Kaiken kattavan etäohjauksen sijaan tutkittiin mahdollisuutta tehdä pilvipalvelusta ohjattava tehostustoiminto, jolla saadaan maksimoitua lämpimän käyttöveden riittävyys esimerkiksi saunapäivinä. Tämänkaltainen yksinkertainen tehostustoiminto ei vaaranna kiinteistön tai lämpöpumpun vaurioitumisriskiä mahdollisen tietoturvahyökkäyksen tapahtuessa.

1.4 Valmiit ratkaisut

Yleinen tietotekniikan kehittyminen ja halpeneminen on näkynyt myös uusissa maalämpöpumpuissa. Nykyisin uusissa maalämpöpumpuissa on kehittyneemmät ohjainlogiikat ja niihin on jo valmiiksi integroitu erilaisia etäyhteystoimintoja. Useimmat yhdistyvät suoraan lähiverkon kautta valmistajan pilvipalveluun, josta loppukäyttäjä voi verkkoselaimella seurata pumpun tilaa tai antaa uusia ohjauskomentoja. Tämänkaltaiset valmistajakohtaiset pilvipalvelut voivat olla lisämaksullisia palveluita tai sitten ne voivat kuulua laitteen vakiovarustukseen. Lämpöpumpussa voi olla optiona lisähallintapaneeli, joka voidaan sijoittaa asuintiloihin.

Kohteen lämpöpumppu on vuodelta 2008, sen ohjausjärjestelmä on pelkästään paikallinen ja siinä ei ole vakiovarusteena mitään pilvitoiminteita tai muutakaan etäohjausvalmiutta. Ohjausyksikön siirtäminen asuintiloihin olisi periaatteessa mahdollista, mutta käytännössä ei järkevää.

Tutkimuksen alussa selvitettiin mahdollisesti jo olemassa olevia ratkaisuita. Kohteen maalämpöpumpussa oleva Ouman-lämmönsäädin on yleinen ja suhteellisen iäkäs laite. Sille on olemassa joitakin kaupallisia, Ouman Oy:n omia ratkaisuita, mutta mitään kolmannen osapuolen kaupallista valvontalaitteistoa ei löytynyt. Oumanin omista kaupallisista vaihtoehtoista on kerrottu luvussa 3.2.

Ei-kaupallisia ratkaisuita datan lukemiseen sarjaportin kautta on olemassa pienehköjä harrastusprojektityyppisiä sekä massiivisempia usean henkilön open source -projekteja kuten openHAB. openHAB oli yhtenä mahdollisuutena toteuttaa projekti, mutta se hylättiin, koska openHAB tuntui siinä vaiheessa liian massiiviselta ratkaisulta, vaikka olisi varmasti ollut toimiva. openHAB on iso open source -kotiautomaatiojärjestelmä, joka tukee lukuisia eri laitteita ja protokollia (openHAB addons 2019). Suoraan Ouman-tukea ei ole mainittu, mutta tuki löytyy esimerkiksi Modbus-protokollalle, jonka avulla Oumania voi ohjata. openHAB on toteutettu Java-ohjelmointikielillä ja projektissa on aktiivinen käyttäjäyhteisö.

Hieman tavallisesta poikkeavana vaihtoehtona maalämpöpumpun valvontaan oli käyttää jotakin valmista IT-ympäristöjen valvontaohjelmistoa kuten Zabbixia. Zabbix on yleinen IT-järjestelmien valvontajärjestelmä, jolla voidaan valvoa palvelimia ja verkkolaitteita. Zabbix mahdollistaa monipuolista valvontaa ja jonkin verran myös järjestelmien ohjausta. Koska Zabbix valvoo IP-laitteita, tulisi maalämpöpumpun ja IP-verkon välille rakentaa jokin yhdyskäytävä. Zabbix oli mielenkiintoisena ja erilaisena vaihtoehtona pitkään mukana, mutta lopullisena hylkäysperusteena oli se, että käyttöliittymää olisi pitänyt muuttaa hyvinkin paljon, että siitä olisi tullut yksinkertainen ja kotikäyttäjätystävällinen. Laajan konfiguroitavuutensa takia Zabbix olisi ollut varteenotettava vaihtoehto.

2 Maalämpö

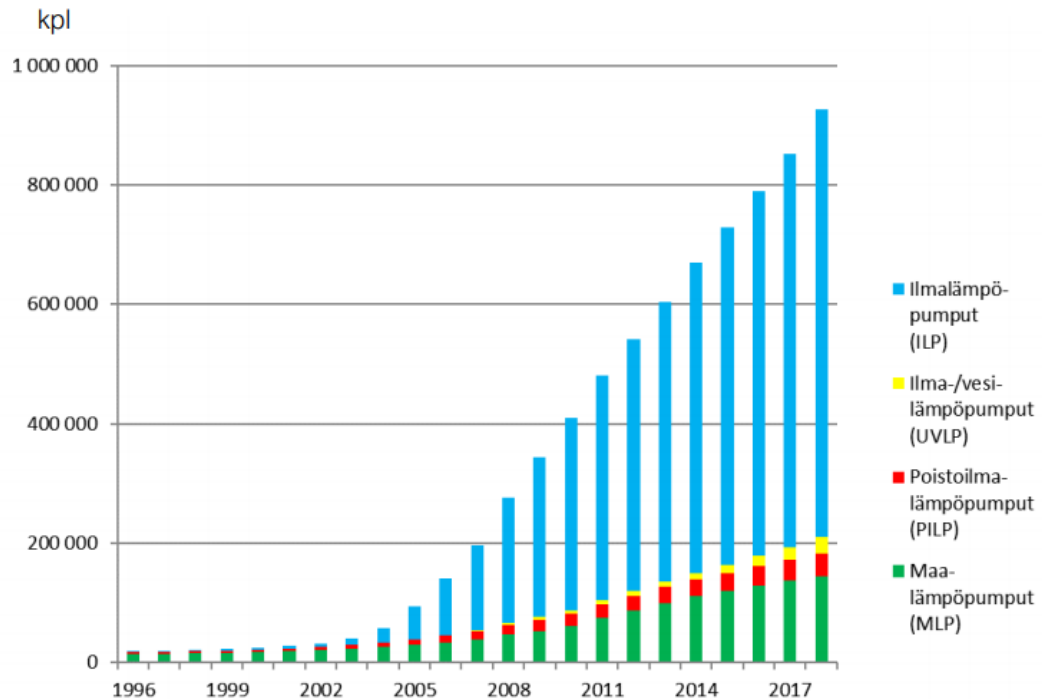
2.1 Yleistä

Maalämpö on lämpöä, joka on varastoitunut maaperään, kallioon tai veteen. Maalämpöä on myös syvemmillä oleva lämpöenergia, joka on vapautunut pohjaveden virtauksista tai maapallon ytimen fissionenergiasta. Maalämpöpumppu kerää tätä energiaa talteen ja luovuttaa esimerkiksi asuinkiinteistöjen lämmitykseen (Maalämpöpumppu, Motiva Oy).

Lämpöpumpun idea on kehitetty vuonna 1853 ja ensimmäinen maalämpöpumppu on rakennettu 1940-luvun lopulla. Ensimmäinen kaupallinen maalämpöasennus on tehty vuonna 1948 Portlandissa, Oregonissa (Maalämpö, Wikipedia).

Suomessa oli 10000 lämpöpumppua Suomen Lämpöpumppuyhdistyksen perustamisvuonna 1999 kun vertailun vuoksi Ruotsin vastaava luku oli 300000 kappaletta (Katsaus lämpöpumppuhistoriaan ja markkinaan, 2019). Nykyisin Suomen lämpöpumppumäärät ovat kasvaneet paljon ja vuonna 2018 Suomen lämpöpumppumäärä oli lähes miljoona lämpöpumppua (Lämpöpumppumarkkinat, 2019). Kuviosta 3 näkyy Suomen lämpöpumppujen määrät. Ilmalämpöpumput ovat yleistyneet rajusti 2000-

luvulla ja maalämpöpumppujenkin määrän kasvu on ollut merkittävää ja esimerkiksi jo vuonna 2011 lähes puolet uusista omakotitaloista lämmitettiin maalämmöllä (Lämpöä omasta maasta 2012).

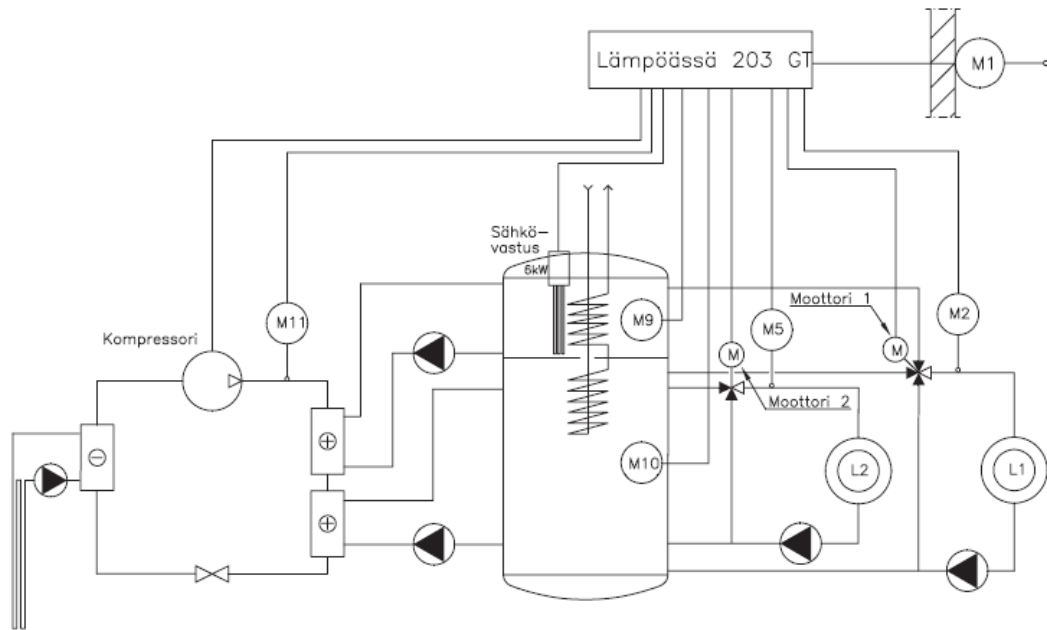


Kuvio 3. Lämpöpumppumäärät Suomessa (Lämpöpumppumarkkinat, 2019)

2.2 Maalämpöpumppu

Keruuputkistossa kiertää jäätymätön neste, joka lämpenee muutaman asteen matkansa aikana. Kerupiirin nesteestä saatava lämpö höyrystää lämpöpumpussa kiertävän kylmäaineen. Höyrystyneen kylmäaineen painetta nostetaan kompressorilla, jolloin myös sen lämpötila nousee. Kylmäaine lauhtuu lämpöpumpun lauhduttimessa jälleen nesteeksi, jolloin se luovuttaa lämpöä lämmönjakoverkkoon ja lämpimään käyttöveteen. (Maalämpöpumppu 2019)

Lämpöpumpun prosessikaavio on esitetty kuvassa (ks. kuvio 4).



Kuvio 4. Lämpöässä V13 prosessikaavio (203GT Lämmönsäädin, käsikirja, s.25)

Kuvasta ilmenee lämmityksen ja pumpun toiminnan kannalta olennaisia mittauskohteita M1 (ulkolämpötila), M2 (L1 -piirin menovesi), M5 (L2 -piirin menovesi), M9 (varaajan yläosan lämpötila), M10 (varaajan alaosan lämpötila) ja M11 (kuumakaasun lämpötila) (203GT Lämmönsäädin, käsikirja, s.25)

Pumpussa on kaksi lämmityspiiriä, joista toinen on kytketty märkiin tiloihin ja toinen normaaleihin huonetiloihin. Kyseisestä järjestelystä on se etu, että kesäaikana suihkutilojen lattioissa voi kiertää lämmin vesi ilman, että muuta kiinteistöä lämmitetään. Lämpöpumpun teoreettinen ottoteho valmistajan ilmoituksen mukaan on noin 4kW ja antoteho 13kW (Lämpöässä V7.0-V30.0 käyttöohje). Edellä mainituista teholuokista voidaan laskea kyseiselle pumppumallille lämpöpumppujen yhteydessä esiintyvä COP-arvo

$$COP = \frac{P_{ANTO}}{P_{OTTO}} = \frac{13kW}{4kW} = 3,25 \quad (1)$$

missä P_{ANTO} = lämpöpumpulle ilmoitettu lämpöenergian antoteho

P_{OTTO} = lämpöpumpun ottama sähköteho

COP arvo 3,25 on tyypillinen lämpöpumppuvalmistajien antama arvo, joka tarkoittaa yksinkertaistettuna sitä, että kyseessä oleva maalämpöpumppu tekee yhdellä kilowatilla sähköä noin 3,25 kW lämpöä.

Tutkimuskohteen maalämpöpumppu kerää lämmön 207m syvästä lämpökaivosta, jonka halkaisija on 5,5". Keruupiirissä on yhteensä noin 3x220m eli noin 660m. Keruuputkisto on kytketty siten, että kaivoon menee keruuliuos kahta putkea pitkin ja liuos tulee ylöspäin yhtä putkea pitkin. Keruuliuos on ympäristöystävällistä alkoholi-pohjaista liuosta

3 Käytettävät teknologiat

3.1 Toteutuksen vaihtoehdot

Suunniteltaessa monitoroinnin toteutusta, on tutkittava etukäteen joitakin avainkysymyksiä

- Millä ja missä muodossa lämpöpumpusta saadaan tietoa ulos?
- Minkälaista ohjelmistoa tarvitaan?
- Missä ohjelmistoa ajetaan?
- Mitä tiedolle tehdään?
- Onko tietoturva kunnossa?

Lämpöpumpun liittämiseen ulkoisiin järjestelmiin on muutamia vaihtoehtoja, joista tulee valita tarkoitukseen sopivin. Lisäksi ohjelmisto voidaan toteuttaa hyvin monilla eri ohjelmointikielillä, tekniikoilla ja osin myös jo olemassa olevia ohjelmistoja apuna käyttäen. Ohjelmiston ajoympäristöjä on valittavana rajallinen määrä ja niistä tulee valita sopivin ja kerätyn datan tallennustavalle ja -paikalle on lukuisia vaihtoehtoja.

3.2 Lämpöpumpun ulkoinen liityntä

Lämpöpumpun Ouman-ohjaimeen on saatavilla valmistajalta sovitinkortit LON- ja Modbus-väyliin sekä liityntä ulkoiselle GSM-modeemille RS232 -väylän kautta. Lisäksi on mahdollista käyttää Oumanin omia erillisiä tuotteita säätimen liittämiseksi esimerkiksi lähiverkkoon. Luvuissa 3.2.1-3.2.4 on lyhyesti esiteltyinä ne liityntätavat, jotka on mainittu tuetuiksi Ouman EH-203 -säätimen käsikirjassa. (Eh-203 Lämmönsäädin, käsikirja, v2.18b)

3.2.1 EH-net

EH-net on Oumanin tarjoama laite, jonka avulla Oumanin laitteita voidaan liittää internettiin. EH-net -laite on kytkettynä Modbus -väylällä Oumanin ohjauslogiikkaan ja lisäksi EH-net sisältää web-palvelimen, jonka kautta loppukäyttäjä pääsee hallitsemaan EH-net:iin liitetyjä laitteita. Palveluun kuuluu ”OUMAN nimipalvelu ja tietoturva”. (EH-net esite). EH-netin elinkaari on päättynyt ja se on poistunut myynnistä. Tarkempi tutustuminen olisi vaatinut investointeja, jotka eivät olisi olleet perusteltuja.

3.2.2 GSM-control

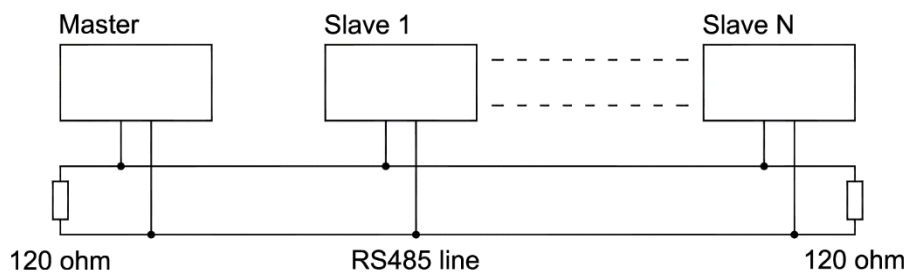
Ouman EH-203 -säätimen RS-232 -liityntään on mahdollista liittää GSM-modeemi. GSM-modeemin avulla laitetta voidaan etäohjata ja valvoa SMS-viesteillä. (Eh-203 Lämmönsäädin, käsikirja, v2.18b). SMS-viesteillä tapahtuva prosessin tai laitteen valvonta ja säätö puolustavat nykyaikana paikkaansa sellaisissa ympäristöissä, joissa nykyaikaisia datayhteyksiä ei ole saatavilla.

3.2.3 Modbus

Modbus-protokolla on viestinvälitysprotokolla, jonka kehitti Modicon, nykyisin Schneider Electric:iin kuuluva yhtiö vuonna 1979 (Modbus FAQ 2019). Modbus on laajalti levinnyt protokolla, jota käytetään paljon esimerkiksi kiinteistöautomaatiossa.

Modbus-protokollasta on kolme eri päätyyppiä, Modbus RTU, Modbus ASCII ja Modbus TCP/IP. RTU-versiossa fyysisenä siirtotienä käytetään RS485-sarjaliikenneväylää, ASCII-versiossa RS232 tai RS485-sarjaliikenneväylää ja TCP/IP-versiossa nimensä mukaisesti normaalia TCP/IP-verkkoa.

Modbus RTU-väylä koostuu Modbus-laitteista, joilla jokaisella on yksillöllinen osoite. Modbus-väylässä on master- ja slave-laitteita, joista master lähettää kyselyn tietyille slave-laitteille ja slave laitteet lähettävät kyselyyn vastauksen. Kuviossa 5 on Modbus RTU RS485-väylän periaatteellinen rakenne.



Kuvio 5. Modbus RTU, RS485 väylä (Protocollo Modbus su RS485 2017)

Laitteiden lukumäärä väylällä rajoittuu protokollan kehysformaatin mukaisesti 247 laitteeseen. Kuviossa 6 on kuvattuna Modbus RTU -protokollan kehysrakenne.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4*	8 BITS	8 BITS	$n \times 8$ BITS	16 BITS	T1-T2-T3-T4*

*For T1-T2-T3-T4, 3.5 character times at no communication.

Modbus Frame Structure-RTU Mode

Kuvio 6. Modbus kehysrakenne (Modbus tutorial. N.d)

Ouman EH-203 -säätimeen on mahdollista liittää Modbus-laajennuskortti, jolla säätimeen saadaan Modbus RTU -rajapinta (Eh-203 Lämmönsäädin, käsikirja, v2.18b).

Liityntävalinta valikoitu Modbus-väyläksi. Modbus-väylän valinta on perusteltua monesta eri syystä.

- Modbus-protokollalle on hyvä ohjelmistotuki monissa eri ohjelmoinkielissä
- Modbus-liityntäkortti on suhteellisen edullinen ja niitä on saatavilla
- Kohteen ilmanvaihtokone tukee Modbus-protokollaa
- Modbus-protokolla on standardi
- Modbus on laajalti käytetty

3.2.4 LON

LON-verkko on kaksisuuntainen sarjaväylä, jota käytetään yhdistämään taloteknisiä järjestelmiä kuten valaistusta, ilmastointia tai kulunvalvontaa. LON-verkko koostuu pääverkosta, aliverkoista, kanavista, segmenteistä ja solmuista (Vahala 2015). LON-verkko on suunniteltu huomattavasti isompiin järjestelmiin kuin mitä tässä tutkimustyössä tarvitaan, siksi sen valinta ei ollut perusteltua.

3.3 Käyttöjärjestelmä

Monitorointiohjelmiston tuli toimia normaalilla pc-laitteistolla, joten realististen käyttöjärjestelmien määrä oli pieni: Windows tai Linux.

Monitorointiohjelmiston käyttöjärjestelmäksi valikoitui Linux. Linuxin valinnan perusteena on sen ilmaisuus, erinomainen soveltuvuus palvelinjärjestelmiin, hyvä tietoliikennepuolen konfiguroitavuus ja hyvä tietoturvan taso. Linuxiin löytyy kaikki tässä järjestelmäkokonaisuudessa tarvittavat palvelinkomponentit ja niiden etähallinta on helppoa.

3.4 Ohjelmistokomponentit

Monitorointiohjelmistokokonaisuus voidaan karkeasti jakaa kahteen eri osaan: taustajärjestelmään (back end) ja käyttöliittymään (front end). Taustajärjestelmä hoitaa tiedon lukemisen, tietoliikenteen ja datan tallentamisen kolmansiin järjestelmiin, esimerkiksi tietokantaan tai julkiseen IoT -pilvipalveluun.

Toinen pääosa eli käyttöliittymä kommunikoi taustajärjestelmän kanssa ja hoitaa mitausdatan näyttämisen ihmiselle sopivassa, miellyttävässä visuaalisessa muodossa.

3.4.1 Back end -ohjelmisto

Lähtökohtaisesti ohjelmiston ydin oli tarkoitus toteuttaa ohjelmointikielellä, joka on tunnettu, laajalti käytetty, hyvin dokumentoitu ja johon löytyy tarvittaessa riittävästi esimerkkikoodia. Lisäksi ohjelmointikieleen tuli ehdottomasti löytyä valmiina tarvittavat kirjastot Modbus RTU -protokollalle ja WebSocket-tiedonsiirtoon.

Edellämainitut vaatimukset huomioiden tutkittiin kahta erityyppistä vaihtoehtoa, joko C-kielellä toteutettavaa tai jollakin skriptikielellä toteutettavaa ratkaisua. C-kielelle löytyy tarvittavat kirjastot ja Linuxiin hyvät C-kielen kehitystyökalut. Toinen vaihtoehto oli Python-ohjelmointikieli. C-kielinen toteutus hylättiin, koska sen kehitystyö olisi ollut hitaampaa ja hankalampaa kuin Python-kehitys. C-kielen etuna olisi ollut nopeus, mutta se ei ollut tässä projektissa olennaista. Python-ohjelmointi oli jo aikaisemmin tuttua ja kieli itsessään hyväksi todettu, joten ainoat tarkasti selvitettävät asiat olivat Modbus-tuki ja websocket-tuki.

Pythonille löytyy Modbus-kirjasto nimeltään PyModbus ja sen ominaisuuksista löytyy tuki kaikille Modbus-protokollille mukaan lukien Modbus RTU (PyModbus – A Python Modbus Stack 2017). Hyvän dokumentaation ja ominaisuuksien takia valinta kohdistui PyModbus-kirjastoon.

WebSocket kirjastoja Pythonille löytyy useita ja valinta kohdistui Tornado web framework -kirjastoon, jossa on paljon ominaisuuksia, hyvä dokumentaatio ja jolla

voi WebSocket-liikenteen lisäksi toteuttaa projektissa tarvittavan web-palvelimen (Tornado Web Server. N.d.).

3.4.2 Front end -ohjelmisto

Monitorointiohjelmiston käyttöliittymän toteutukselle oli käytännössä kaksi vaihtoehtoisia isompaa linjaa: joko natiivi Android-ohjelmisto tai www-sivu. Näistä kahdesta valikoitui www-sivu siksi, että se ei ole laiteriippuvainen ja sen toteutus on huomattavasti yksinkertaisempaa ja nopeampaa kuin natiivin mobiiliohjelmiston.

Front endin www-toteutusteknologiaan oli myös vaihtoehtoja kuten jonkin julkaisualustan käyttö, Java-pohjainen sovellus tai sitten puhdas HTML-sivu. Näistä lyhyen tarkastelun jälkeen valikoitui HTML-sivu, koska se on yksinkertainen, helppo muokata, erinomaisesti yhteensopiva ja kevyt. Lisäksi www-sivustoa tarjoava palvelin ei vaadi kohtuuttomia palvelinresursseja kuten muistia tai CPU-tehoa.

Koska valvontasivu pitää olla dynaaminen ja automaattisesti päivittyvä, tulee front end sisältämään Javascriptiä. Vaikkakin käyttöliittymä on kokonaisuuden näkyvin osa, siihen ei haluttu kehittää itse komponentteja vaan lopulliseen toteutukseen pyrittiin käyttämään mahdollisimman paljon valmiita kirjastoja ja valmista grafiikkaa.

3.4.3 Tietokanta

Tietojen tallentamiseen on monia teknologisia vaihtoehtoja, esimerkiksi tietokanta, pilvipalvelu tai tiedosto. Kuten aiemmin todettiin, ohjelmiston rakenteen tuli olla sellainen, että mikään valittu tietovarasto ei ole liiaksi ”kovakoodattu”, vaan tietovaraston sijaintia ja tyyppiä tulee voida pienellä vaivalla muuttaa ja mahdollisuuksien mukaan tietovarastoja tulisi voida lisätä useampia.

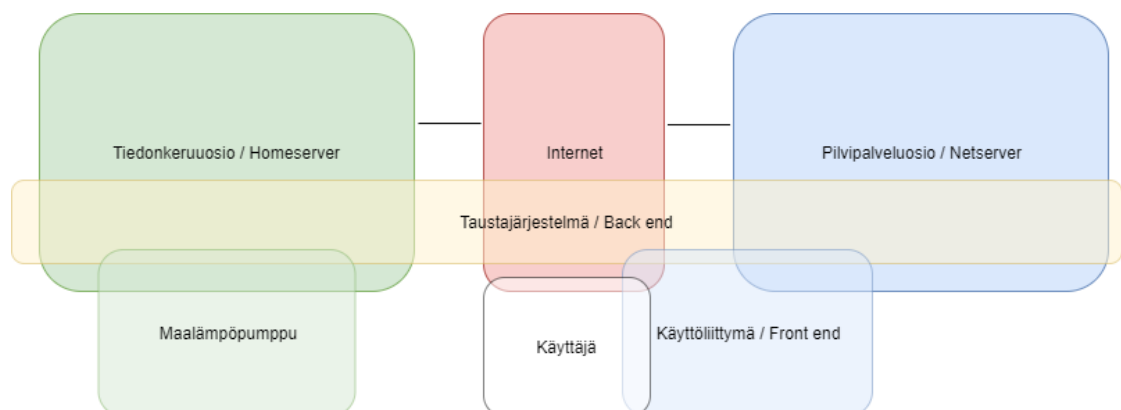
Tietokannaksi valikoitui SQL-pohjainen MariaDB. Valintaan vaikuttivat aikaisemmat hyvät kokemukset, helppo konfigurointi ja saatavuus Linux-alustoille. Kevyempi vaihtoehto SQLite olisi myös ollut toimiva ratkaisu, mutta odotettavissa oli, että kantaan

tulee suhteellisen paljon rivejä, joten ns ”oikean” SQL-ratkaisun käyttäminen oli perusteltua. MariaDB oli myös valintana siksi, että tietokannan perusrakenteena voitiin käyttää aikaisemmassa Advanced Databases-opintojakson harjoitustyön tuloksena syntynyttä tietokantarunkoa (Latomäki, J & Vulli, M, AnturiTietokanta 2016). Tietokannan SQL-malli on liitteessä 2.

4 Järjestelmän arkkitehtuuri

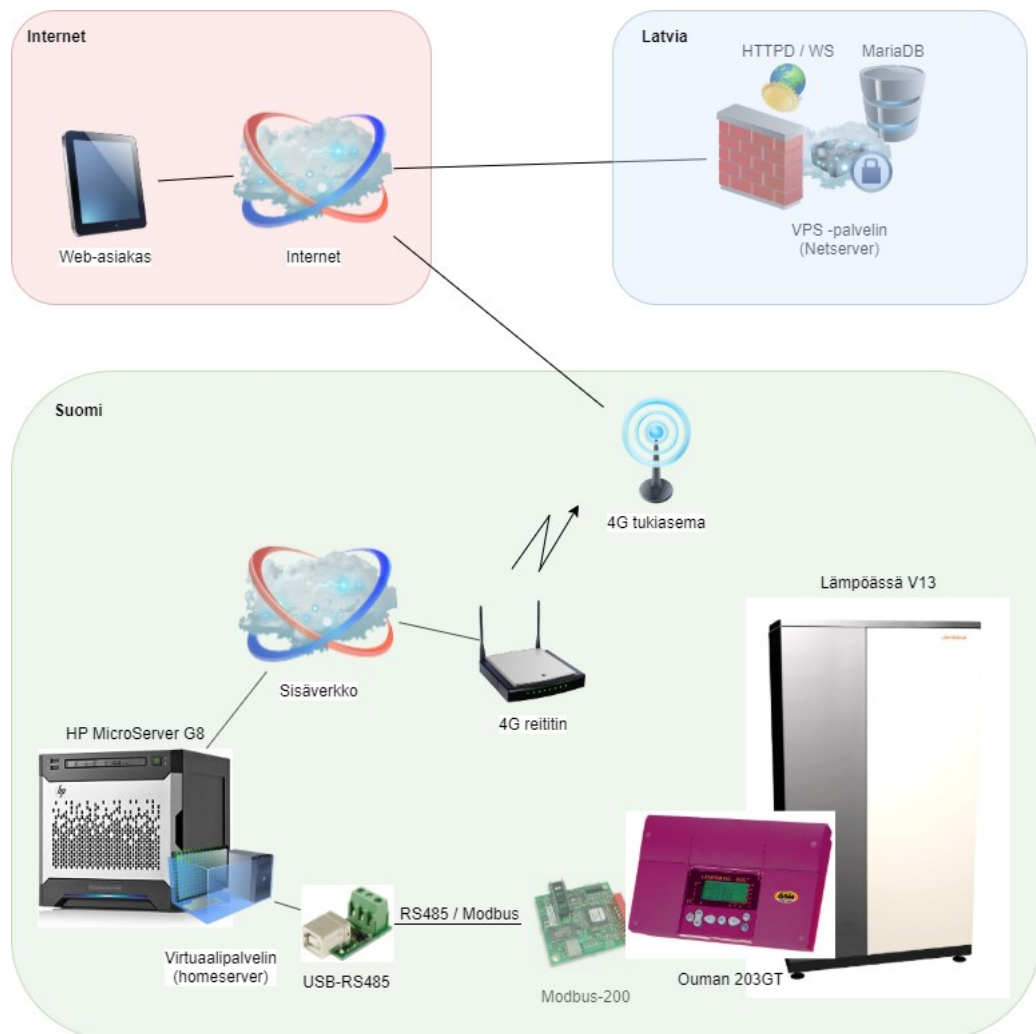
4.1 Järjestelmän kokonaiskuvaus

Monitorointijärjestelmä voidaan karkeasti jakaa front end ja back end -jaon lisäksi myös tiedonkeruuosioon (homeserver) ja pilvipalveluosioon (netserver) (ks. kuvio 7). Homeserver-ohjelmistoa ajetaan lämpöpumpun läheisyydessä olevalla laitteistolla ja netserver-ohjelmistoa ajetaan erillisellä palvelimella, jonka sijainnille ei ole muita vaatimuksia kuin se, että sinne voidaan muodostaa TCP/IP-yhteys homeserver-palvelimelta.



Kuvio 7. Järjestelmän arkkitehtuuri

Järjestelmän verkkokuvassa (ks. kuvio 8) on kuvattu yhteydet eri komponenttien välillä sekä niiden sijainti.



Kuvio 8. Järjestelmän verkkokuva

Tiedonkeruusion komponentit sijaitsevat fyysisesti lähellä monitoroitavaa lämpöpumppua. Kerätty data siirtyy kiinteistön sisäisen LAN-verkon kautta 4G-reitittimelle, josta edelleen VPS-palvelimelle. VPS -palvelin on kolmannelta osapuolelta hankittu virtuaalinen Linux CentOS 7 -käyttöjärjestelmällä varustettu palvelin. VPS-palvelimen fyysinen sijainti on Latvia.

4.2 Tiedonkeruusion komponentit

Tiedonkeruuhjelmisto sisältää varsinaisen ohjelmakoodin lisäksi myös palvelimen ja sen fyysisen liitynnän maalämpöpumppuun.

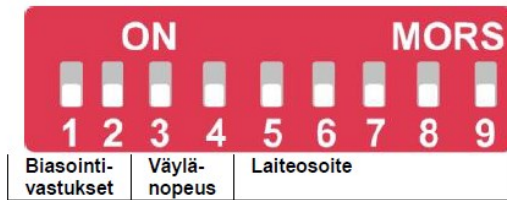
4.2.1 Lämpöpumppu

Maalämpöpumpun Ouman-ohjaimessa on valmius Modbus RTU -protokollan käyttöön. Modbus-protokollan toteutus ei ole laitteessa vakiovarusteena, vaan se täytyy erikseen ottaa käyttöön asentamalla fyysinen Modbus-moduuli ohjainlaitteen sisälle. Modbus-moduuli on pieni piirikortti, joka liitetään piikkirimaliitännällä ohjainlaitteeseen kiinni. Modbus-moduulin kuva on kuviossa 9. Kuvio 9. Ouman EH-200 Modbus-kortti



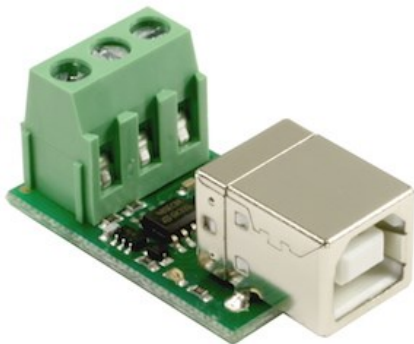
Kuvio 9. Ouman EH-200 Modbus-kortti (EH-200 säätimen Modbus-rajapinta)

Asennuksen yhteydessä moduulille konfiguroidaan DIP-kytkimillä (ks. kuvio 10) väylän biasointivastus, nopeus- ja osoite parametrit. Biasointivastus otetaan käyttöön DIP-kytkimillä 1-2, väylän nopeus DIP-kytkimillä 3-4 ja osoite konfiguroidaan DIP-kytkimillä 5-12. Osoite voi olla väliltä 1-247 (Ouman. EH-200 säätimen Modbus-rajapinta, 3)



Kuvio 10. Modbus -kortin DIP-kytkin (EH-200 säätimen Modbus-rajapinta, 3)

Modbus-kortti tarjoaa Modbus RTU -rajapinnan käytettäväksi fyysisellä RS485 sarjavylyllä. RS485 on galvaanisesti erotettu väylä, jossa vain yksi laite voi lähettää kerrallaan ja muut laitteet kuuntelevat. (Ouman. EH-200 säätimen Modbus-rajapinta, 3). RS485-väylä on teollisuudessa yleinen kenttäväylä, mutta vaatii oman sovittimensa toimiakseen tavallisissa pc-laitteissa, ja tämän vuoksi palvelimeen tuli saada järjestettyä RS485 -yhteensopiva liityntä. RS485-liitynnän toteuttamiseen oli käytännössä kaksi vaihtoehtoa, palvelimen sisäinen kortti tai ulkoinen USB-liityntään kytkettävä kortti. Näistä vaihtoehdoista valittiin helpommin saatavilla oleva ja edullisempi USB-RS485-sovitinmoduuli (ks. Kuvio 11).



Kuvio 11. USB-RS485 -muunnin (USB-RS485 Communications Module, Technical specification)

Fyysinen kytkentä lämpöpumpun Modbus-kortin ja palvelimessa olevan USB-RS485 -muuntimen välissä on normaali CAT6 ethernet -kaapeli, josta on käytössä 1 kierretty pari. Kaapelin pituus on 2 m.

4.2.2 Palvelin

Palvelin on laite, joka on fyysisesti kytketty sekä internetiin että maalämpöpumpuun. Palvelimessa ajetaan tiedonkeruusion ohjelmistoa, joka toimii yhdyskäytävänä lämpöpumpun ja internetin välillä.

Palvelinrautana toimii HP ProLiant Microserver Gen8 -palvelin (ks. kuvio 12), joka on pientoimistoille tarkoitettu pienikokoinen palvelin. Palvelimen käyttöjärjestelmänä on VMware ESXi -virtualisointialusta, jossa yhtenä virtuaalikoneena on CentOS 7 -käyttöjärjestelmällä varustettu virtuaalikone.



Kuvio 12. HP ProLiant Microserver G8 (www.techradar.com, HP ProLiant MicroServer Gen8 review)

4.2.3 Ohjelmisto

Tiedonkeruuhjelmisto on toteutettu kokonaan Python -kielellä. Käytetty Python versio oli aluksi versio 2, mutta vaihtui sittemmin versioon 3. Ohjelmistoa ajetaan kokonaisuudessaan virtuaalipalvelimella.

4.3 Pilviosion komponentit

Pilvipalvelussa ainoana komponenttina on virtuaalinen palvelin, jossa tapahtuu datan vastaanotto tiedonkeruusta, datan tallentaminen tietokantaan sekä tarjotaan web-palvelua loppukäyttäjälle.

4.3.1 Palvelin

VPS -palvelin, johon pääsy SSH-yhteydellä ja jossa on root-tason oikeudet. Palvelin normaali Linux-palvelin jossa on CentOS 7 käyttöjärjestelmä ja on johon asennettu ja konfiguroitu Apache web -palvelu ja MariaDB -tietokantapalvelu. Tietoturvasoa nostettu iptables-palomuurisäännöillä sekä https-varmenteella.

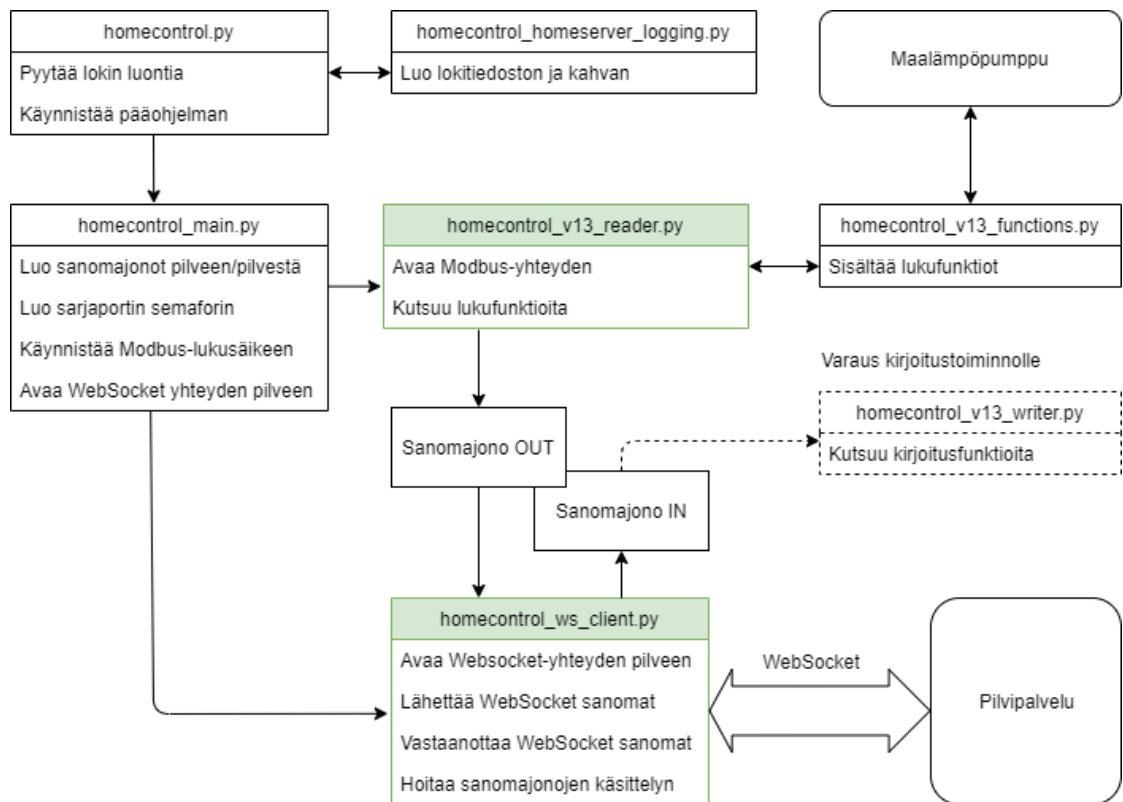
4.3.2 Ohjelmisto

Ohjelmisto toteutettu käyttäen Python 3-versiota, Javascriptiä sekä HTML5-kieltä.

5 Ohjelmisto

5.1 Tiedonkeruu

Tiedonkeruuohjelmiston toimintaperiaate on esitetty kuviossa 13.



Kuvio 13. Tiedonkeruusion toimintaperiaate

Tiedonkeruu ohjelmisto käynnistetään Linux-konsolista komennolla

```
python3.4 homecontrol.py
```

Käynnistyksen jälkeen alustetaan lokitiedosto ja lokiasetukset, jonka jälkeen käynnistetään `Homecontrol_main`-säie. `Homecontrol_main` on käynnistysluokka, joka alustaa sanomajonot, sarjaportin semaforin, avaa WebSocket-yhteyden pilvipalveluun sekä käynnistää säikeen, jossa luetaan tietoja lämpöpumpulta. Kuviossa 13 on merkitty vihreällä värillä tiedostot, joiden sisältämät säikeet jäävät ikuisen silmukkaan.

Ikuiset silmukat `ReaderV13` ja `Ws_client` hoitavat mittaustietojen lukemista lämpöpumpulta sekä tietojen lähetystä pilvipalveluun. Kuviossa 14 on esitetty tietojen lukusäikeen alku.

```

13 class ReaderV13(threading.Thread):
14     def __init__(self, queue_in, queue_out, serial_semaphore, running):
15         threading.Thread.__init__(self)
16         self.queue_in = queue_in
17         self.queue_out = queue_out
18         self.serial_semaphore = serial_semaphore
19         self.running = running
20
21     try:
22         self.client = ModbusClient(method='rtu', port='/dev/ttyUSB0', timeout=1, baudrate=38400, data=8, stopbits=2)
23     except:
24         logging.error("Error creating modbus client: " + str(sys.exc_info()[0]))
25         logging.debug("Modbus client: " + str(self.client))
26
27     self.start()
28
29     def init_client(self):
30     try:
31         self.client = ModbusClient(method='rtu', port='/dev/ttyUSB1', timeout=1, baudrate=38400, data=8, stopbits=2)
32     except:
33         logging.error("Error creating modbus client: " + str(sys.exc_info()[0]))
34         logging.debug("Modbus client: " + str(self.client))
35
36

```

Kuvio 14. ReaderV13

Reader13 -luokka periytyy Pythonin Thread -luokasta, jolloin lukurutiinit saadaan tausta-ajoon. Luokalle annetaan kutsuttaessa parametreina kaksi sanomajonoa, sarjaportin semafori sekä *running* parametri. Sanomajonoista vain toinen on tässä toteutuksessa käytössä ja sitä käytetään pilvipalveluun lähetettävän tiedon jonona. Toiseen sanomajonoon tulee data pilvipalvelusta, mutta tätä toimintoa ei ole tietoturvasyistä käytössä tässä toteutuksessa. *Running* -parametria ei tässä toteutuksessa käytetä ja se on mukana tulevaisuuden kehitystä varten ja sillä voidaan hoitaa säikeen hallittu sammutus.

Rivillä 22 (ks. Kuvio 14. ReaderV13) alustetaan Modbus-yhteys maalämpöpumppuun. Funktio ModbusClient sisältyy PyModbus-kirjastoon. Alustuksessa annettavilla parametreilla määritetään Modbus-yhteyden tyyppi, sarjaportti, aikakatkaisun pituus, sarjayhteyden nopeus, databittien lukumäärä ja stop-bittien lukumäärä. Maalämpöpumpulla oleva Modbus-liityntä on Modbus RTU-tyyppinen, joten parametrina annetaan *method='rtu'*. Muut asetukset ovat vapaasti valittavia, mutta ne pitää olla vastaavat, mitä lämpöpumpun Modbus-korttiin on konfiguroitu.

Ohjelmiston funktiokutsut ja komennot ovat mahdollisuuksien mukaan try-except -rakenteen sisällä, jolloin saadaan mahdollisista virhetilanteista lokitalenne.

Kuviossa 15 on esimerkki funktiosta, jolla luetaan kuumakaasun lämpötila maalämpöpumpulta.

```
141 def read_hotgas_temp(client):
142     logging.debug("Function: read_hotgas_temp ")
143     logging.debug("Modbus client: " + str(client))
144     try:
145         data = client.read_holding_registers(HOTGAS_TEMP, 1, unit=1)
146         logging.debug("Hotgas temperature " + str(data.registers[0] / 100))
147     except:
148         print(sys.exc_info())
149     return data.registers[0] / 100
150
151
```

Kuvio 15. Lukufunktio

Lukufunktiolle annetaan parametrina Modbus-asiakkaan ilmentymä (ks Kuvio 14). Funktiota kutsuttaessa sarjaporttiyhteys on auki ja tiedon lukeminen tapahtuu PyModbus-kirjaston `read_holding_registers`-funktiolla. Funktiolle annetaan parametrina aloitusrekisterin numero, luettavien rekisterien lukumäärä ja kohdelaitteen osoite. Esimerkitapauksessa rekisterin numero on määritelty aikaisemmin ja funktio kutsussa käytetään muuttujaa `HOTGAS_TEMP`, rekistereitä luetaan 1 kpl ja lämpöpumpun Modbus-osoite on 1. Funktio palauttaa luetun rekisterin arvon ja tässä esimerkissä palautusarvo tulee jakaa luvulla 100, että saadaan todellinen kuumakaasun lämpötila. Lukufunktion lopuksi luettu tieto palautetaan sitä kutsuneelle funktiolle.

Kun kaikki määritellyt tiedot on luettu pumpulta, niistä koostetaan Python dictionary-tietorakenne (ks. kuvio 16).

```

53 self.measurements["measurements"].append({
54     "v13_measurement_time": timestamp,
55     "v13_outside_temp": read_outside_temp(self.client),
56     "v13_tank_up_temp": read_up_temp(self.client),
57     "v13_tank_low_temp": read_low_temp(self.client),
58     "v13_hotgas_temp": read_hotgas_temp(self.client),
59     "v13_l1_temp": read_l1_temp(self.client),
60     "v13_l2_temp": read_l2_temp(self.client),
61     "v13_compressor_status": read_compressor_status(self.client),
62     "v13_resistor_status": read_resistor_status(self.client),
63     "v13_setting_l1_minus20": read_setting_L1_minus20(self.client),
64     "v13_setting_l1_0": read_setting_L1_0(self.client),
65     "v13_setting_l1_plus20": read_setting_L1_plus20(self.client),
66     "v13_setting_l2_minus20": read_setting_L2_minus20(self.client),
67     "v13_setting_l2_0": read_setting_L2_0(self.client),
68     "v13_setting_l2_plus20": read_setting_L2_plus20(self.client),
69     "v13_setting_tank_up_temp": read_setting_up_temp(self.client),
70     "v13_setting_tank_low_temp": read_setting_low_temp(self.client),
71     "v13_resistor_time": read_resistor_time(self.client),
72     "v13_compressor_time": read_compressor_time(self.client),
73     "v13_alarm_sum_a": read_alarm_sum_a(self.client),
74     "v13_alarm_sum_b": read_alarm_sum_b(self.client)
75 })
76

```

Kuvio 16. Mittauksien tietorakenne

Tietorakenteen muodostamisen jälkeen Modbus-yhteys suljetaan ja vapautetaan sarjaportin semafori. Yhteyden sulkeminen ja semaforin vapautus ei olisi pakollista, mutta testausvaiheessa on tullut ilmi, että kun yhteys on kauan auki, saattaa dataliikenteeseen tulla virheitä.

Yhteyden sulkemisen jälkeen kuvion 16 tietorakenteesta muodostetaan JSON-paketti ja se laitetaan sanomajonoon (ks. kuvio 17).

```

96 # Json & put measurements to ques
97 data = json.dumps(self.measurements)
98 self.queue_out.put(data)

```

Kuvio 17. JSON paketti sanomajonoon

Kuviossa 18 on funktio, mikä hoitaa sanomajonojen liikenteen.

```

58     def handle_queues(self):
59         if self.ws is None:
60             logging.info("No connection. Reconnecting.")
61             self.connect()
62         else:
63             # Handle messages FROM cloud
64             if self.queue_in.empty() is True:
65                 # Input queue is empty. Do nothing.
66                 pass
67             else:
68                 # We have something in input queue. Send that to somewhere.
69                 pass
70
71             # Handle messages TO cloud. Loop until queue is empty
72             while self.queue_out.empty() is False:
73                 logging.info("Out_queue size: " + str(self.queue_out.qsize()))
74                 try:
75                     self.ws.write_message(str(self.queue_out.get()))
76                 except:
77                     logging.error("Error sending message to cloud: " + str(sys.exc_info()[0]))
78

```

Kuvio 18. Sanomajonojen hallinta

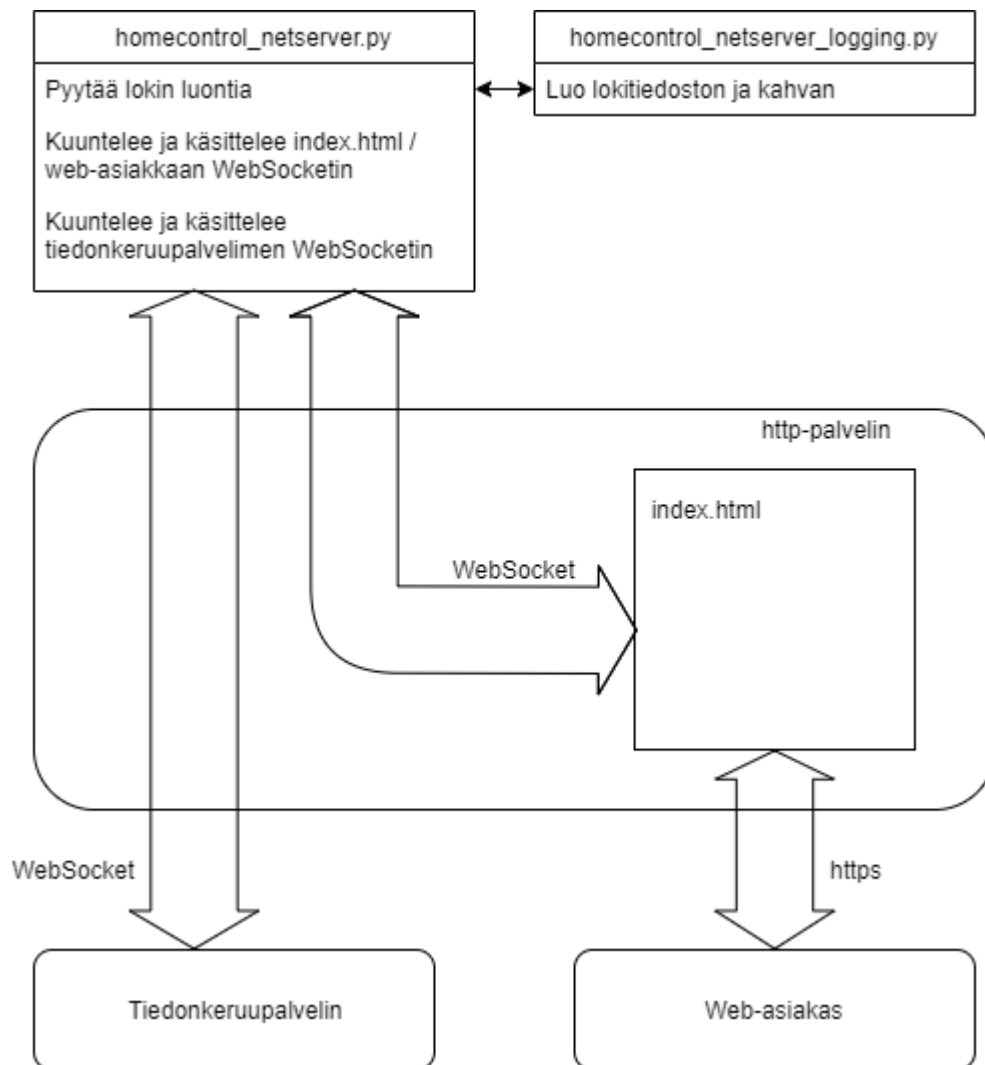
Kuviossa 17 JSON-tietorakenne laitettiin queue_out-sanomajonoon. Kuvion 18 funktiossa kyseinen sanomajono käsitellään riveillä 72-77, jossa jonosta otetaan tietoa yksi kerrallaan, kunnes jono on tyhjä ja jokainen otettu sanoma lähetetään aikaisemmin avatulla WebSocket-yhteydellä pilvipalveluun.

Kuviossa Kuvio 18 näkyy valmius sanomiin, jotka tulevat pilvipalvelusta. Tällä valmiudella varaudutaan siihen, että pilvipalvelusta voisi antaa komentoja tai muuttaa maalämpöpumpun parametreja.

Pilvipalveluun lähtevällä sanomajonolla voidaan varmistaa se, että jos pilvipalvelu on tavoittamattomissa niin sanomajono puskuroid mittausviestejä. Kun yhteys jälleen palautuu, puskuroidut viestit lähetetään pilvipalveluun.

5.2 Pilvipalvelu

Pilvipalvelun ohjelmiston toimintaperiaate on esitetty kuviossa 19.



Kuvio 19. Pilvipalveluusion toimintaperiaate

Pilvipalvelun ohjelmisto käynnistetään Linux-konsolista komennolla

```
python3.4 homecontrol_netserver.py
```

Käynnistyksen jälkeen alustetaan loki, jonka jälkeen luodaan Tornado web framework -kuuntelija tiedonkeruupalvelimen `WebSocket`-yhteydelle ja web-käyttöliittymän `WebSocket`-yhteydelle. Kuviossa 20 on `WebSocket`-kuuntelijoitten alustus.

```

127 class Homecontrol_server(tornado.web.Application):
128     def __init__(self):
129         try:
130             homecontrol_netserver_handlers = [(r'/homecontrol_client', Homecontrol_client_handler), (r'/homecontrol_server', Homecontrol_server_handler)]
131         except:
132             print("Handler setting failed " + str(sys.exc_info()[0]))
133         try:
134             tornado.web.Application.__init__(self, homecontrol_netserver_handlers)
135         except:
136             print("tornado.web.application failed " + str(sys.exc_info()[0]))
137

```

Kuvio 20. WebSocket kuuntelijoitten alustus

Kuuntelijoitten alustuksessa kerrotaan, mitä tehdään, kun tiettyyn url:ään avataan yhteys. Kuuntelijoitten käsittelyrutiineissa on on_message-funktio, joka suoritetaan aina, kun käsittelijälle saapuu WebSocket-sanoma. Kuviossa 21 on tiedonkeruupalvelimelta saapuvan WebSocket-sanoman on_message-funktio.

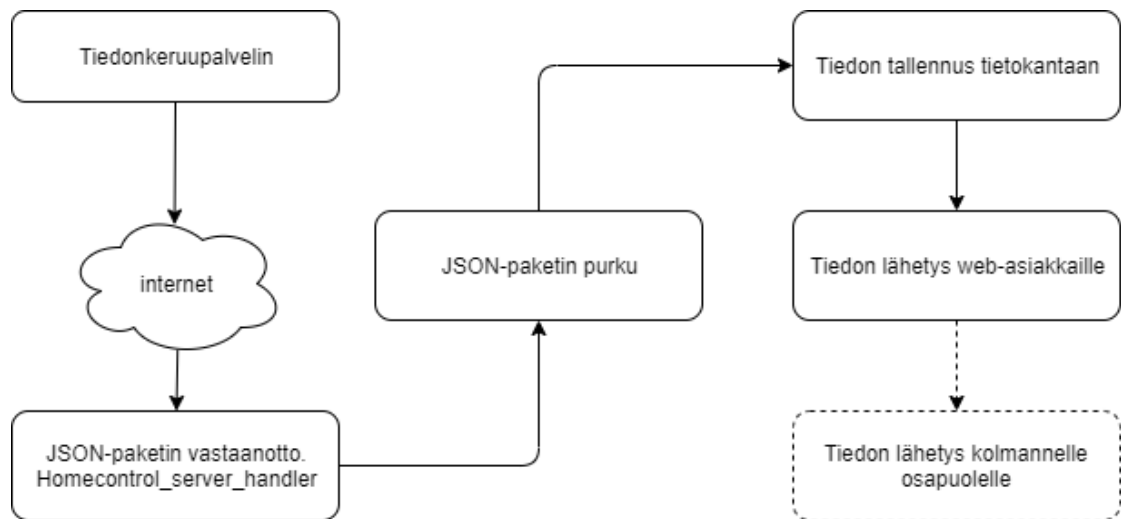
```

128     def on_message(self, message):
129         self.message = message
130         logging.info("Got message from homeserver " + str(self.x_real_ip_server_handler))
131         logging.debug("Got message from homeserver (" + str(self.x_real_ip_server_handler) + "): " + self.message)
132         self.parse_message(self.message)
133

```

Kuvio 21. on_message-funktio

Kun tiedonkeruupalvelimelta saapuu WebSocket-sanoma, ajetaan automaattisesti on_message-funktio. Kyseisessä funktiossa sanoman käsittely aloitetaan kutsumalla parse_message-funktiota, jossa JSON-sanoma puretaan ja mittatiedon kulku jatkuu kuvion 22 mukaisesti.



Kuvio 22. Mittausdatapaketin kulku

5.3 Verkkosivusto

Loppukäyttäjä näkee kerätyn tiedon verkkosivulla, jonne tieto päivittyy automaattisesti aina, kun uusi tieto saapuu palvelimelle. Sivuston toteutukseen on käytetty HTML5-, CSS- ja Javascript-kieltä.

Sivuston perusrunko on luotu www.layoutit.com -verkkosivuston Bootstrap builder -generaattorilla. Generaattori luo responsiivisen www-sivun, jolla on haluttu määrä rivejä ja sarakkeita.

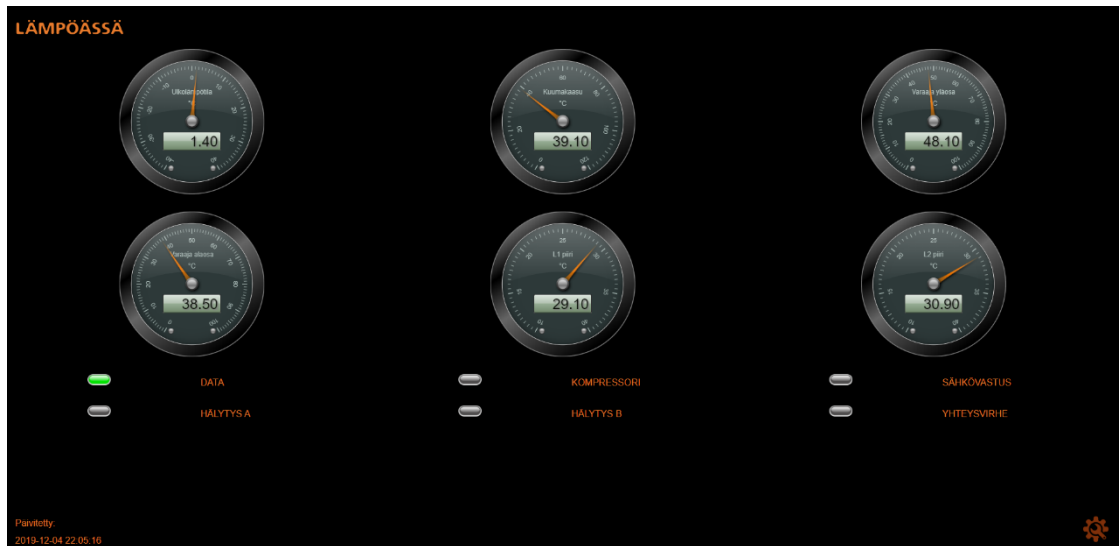
Sivustolla on graafisia komponentteja, joita päivitetään uuden tiedon saapuessa. Mittarit ovat SteelSeries open-source-komponentteja (Steelseries JavaScript edition 2011). Kuviossa 23 on esimerkki Steelseries-mittarin luonnista.

```
4 function init_gauges() {
5
6     gaugeOutdoorTemp = new steelseries.Radial('MeasOutdoorTemp',
7     {
8         size: 250,
9         area: steelseries.areas,
10        unitString: '\xB0C',
11        threshold: 100,
12        lcdVisible: true,
13        ledVisible: false,
14        thresholdVisible: false;
15    });
16
17    gaugeOutdoorTemp.setValue(0);
18    gaugeOutdoorTemp.setMinValue(-40);
19    gaugeOutdoorTemp.setMaxValue(40);
20    gaugeOutdoorTemp.setFrameDesign(steelseries.FrameDesign.TILTED_BLACK);
21    gaugeOutdoorTemp.setBackgroundColor(steelseries.BackgroundColor.SATIN_GRAY);
22    gaugeOutdoorTemp.setPointerColor(steelseries.ColorDef.ORANGE);
23    gaugeOutdoorTemp.setTitleString("Ulkolämpötila");
24    gaugeOutdoorTemp.setLcdColor(steelseries.LcdColor.STANDARD_GREEN);
25
```

Kuvio 23. Lämpömittarin luonti

Kun mittarin ilmentymä on luotu, sille annetaan mittarille soveltuvia ominaisuuksia, joista tärkeimmät ovat minimiarvo, maksiarvo ja mittarin otsikkokenttä. Mittarin luonnin jälkeen sille voidaan antaa arvoja ilmentymän setValue-metodilla.

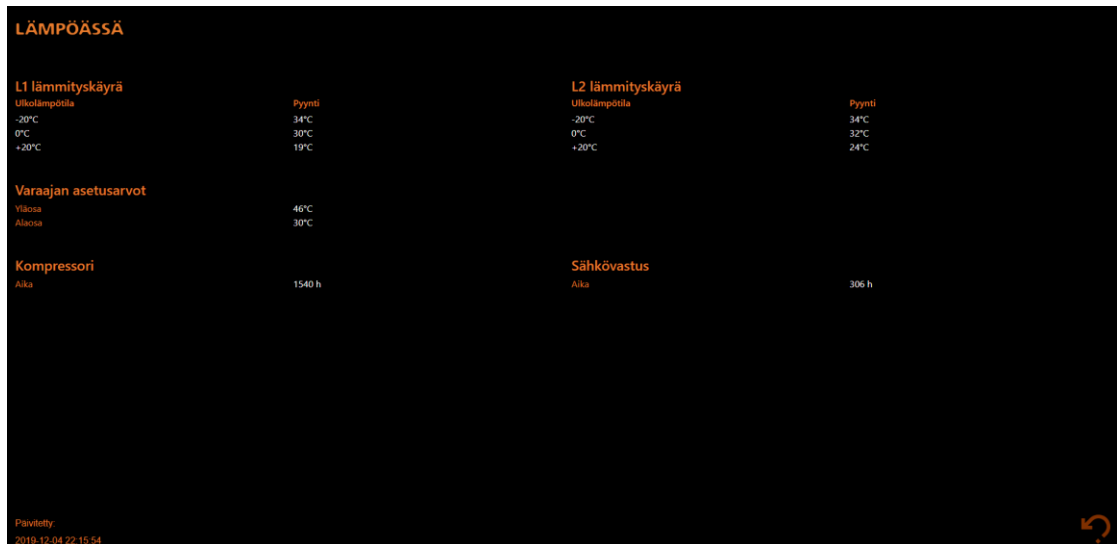
Kun käyttäjä avaa verkkosivun, avautuu normaali HTML-sivu ja samalla Javascript avaa taustalla WebSocket-yhteyden taustajärjestelmään. Taustajärjestelmä pitää kirjaa web-asiakkaista ja lähettää kaikille asiakkaille tiedonkeruupalvelimelta saapuvan uuden mittaustiedon. Verkkosivusto on single page -sivu, jossa on kaksi näkymää. Näkymää vaihdetaan oikean alareunan napilla ja varsinainen vaihto tehdään piilottamalla ja näyttämällä sivuston div-osioita. Käyttöliittymän etusivu on kuviossa 24.



Kuvio 24. Verkkosivuston mittaussivu

Mittaussivulla näkyy 6 eri lämpötilamittausta ja 6 eri tilatietoa. Lämpötilat esitetään mittareilla ja tilatiedot merkkivaloilla. Sivun vasemmassa alareunassa on kellon aika, jolloin viimeksi sivu on saanut uutta mittautietoa. Uuden mittautiedon saapuessa DATA-merkkivalo syttyy. Muut merkkivalot ovat KOMPRESSORI, joka palaa silloin, kun maalämpöpumpun kompressori on käynnissä. SÄHKÖVASTUS-merkkivalo palaa, kun maalämpöpumpun sähkövastus on päällä. Sähkövastus kytkeytyy päälle vain silloin kun on jokin vikatilanne. HÄLYTYS A- ja HÄLYTYS B-merkkivalot palavat, mikäli lämpöpumpussa on jokin hälytys päällä. Hälytykset ovat ns. summahälytyksiä ja tarkempi hälytyksen tyyppi pitää tarkistaa maalämpöpumpun käyttöpaneelista. YHTEYSVIRHE-merkkivalo palaa, mikäli käyttöliittymä ei ole saanut yhteyttä mittausjärjestelmän taustajärjestelmään.

Käyttöliittymän näkymää voi vaihtaa oikean alareunan napista. Asetussivulla (ks. kuvio 25) näkyy lämpöpumpun yleisiä asetustietoja sekä kaksi mittautietoa, kompressorin ja sähkövastuksen päälläoloajat.



Kuvio 25. Verkkosivuston asetussivu

Verkkosivuston värimaailmassa käytetään maalämpöpumpun valmistajan oranssia väriä ja valmistajan virallista logoa. Koska yksi päätavoitteista oli tehdä kiinteistön pukuhuonetiloihin kiinteä tablet-asennus, on sivuston yleisilme huomiota herättämätön tumman sävyinen.

6 Testaus ja vikatilanteista toipuminen

6.1 Vikatilanteet

Luvuissa 6.2-6.4 on testitapauksia sekä tiedonkeruupalvelimen, että pilvipalvelimen näkökulmasta tarkasteltuna. Jokaisen testitapauksen jälkeen on oletus, että järjestelmä palvelee loppukäyttäjää normaalisti. Indikaattorina on se, että loppukäyttäjän käyttöliittymän data-ledi vilkkuu.

6.2 Palvelinpää

Seuraavat testitapaukset on testattu pilvipalvelun näkökulmasta katsottuna.

6.2.1 Pilvipalvelimen uudelleenkäynnistys

Pilvipalvelimelle tehdään uudelleenkäynnistys komentamalla komentokehotteessa *reboot*.

Palvelin uudelleenkäynnistyy ja toiminnot palautuvat normaaleiksi.

6.2.2 Tiedonkeruupalvelimen uudelleenkäynnistys

Tiedonkeruupään uudelleenkäynnistämällä ei ole vaikutusta pilvipalvelun toimintaan.

6.2.3 Tietoliikennekatkos

Tietoliikennekatkolla ei ole vaikutusta pilvipalvelun toimintaan.

6.2.4 Tietokanta

Tietokantapalvelin sammutetaan hallitusti komentoriviltä.

Aiheutuu poikkeus ja yhteys tiedonkeruupalvelimeen katkeaa.

```
_mysql_exceptions.OperationalError: (2002, "Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (2)")
```

Tietokantapalvelimen käynnistämisen jälkeen tilanne palaa normaaliksi.

6.2.5 Tietomurtoyritykset

Generoidaan paljon liikennettä ja CPU -kuormaa palvelimelle. Generoidaan CPU-kuormaa ajamalla komentokehotteessa komento `dd if=/dev/zero of=/dev/null` jossa kopioidaan nollaa Linuxin /dev/null -laitteeseen.

CPU-kuorman lisäyksellä ei ollut vaikutusta.

CPU-kuorman lisäksi ajetaan ”ping floodia” palvelimen osoitteeseen.

Havaittavissa oli erittäin pientä viivettä käyttöliittymän päivityksessä, mutta oli kuitenkin käytettävä.

6.3 Tiedonkeruupää

Seuraavat testitapaukset on testattu tiedonkeruupään näkökulmasta katsottuna.

6.3.1 Tiedonkeruupalvelimen uudelleenkäynnistys

Tiedonkeruupalvelimelle tehdään uudelleenkäynnistys komentamalla komentokohotteessa *reboot*

Tiedonkeruuohjelmisto käynnistyy automaattisesti ja tilanne palautuu normaaliksi.

6.3.2 Pilvipalvelimen uudelleenkäynnistys

Pilvipalvelu saavuttamattomissa.

Tiedonkeruupää puskuroi mittaustietoa sanomajonoon. Kun pilvipalvelu on saavutettavissa, jono purkautuu ja tilanne palautuu normaaliksi.

6.3.3 Lämpöpumpun vikatilanne

Sammutetaan lämpöpumppu päävirtakytkimestä ja laitetaan takaisin päälle.

Tiedonkeruuohjelmisto pysyy päällä, mutta yhteys lämpöpumppuun katoaa, eikä palaudu.

6.4 Web-käyttöliittymä

6.4.1 Toimivuus

Tarkistetaan yleinen toimivuus muutamalla yleisimmällä selainversiolla, Windows, Linux, Android, IOS

Verkkosivusto toimii yleisillä selaimilla. Pienillä matkapuhelinnäytöillä käyttöliittymän elementit eivät sijoitu täydellisesti, mutta on kuitenkin käytettävä.

6.4.2 Useat sessiot

Avataan käyttöliittymiä useasta paikasta auki.

Testattu useilla käyttöliittymillä. Palvelu käytettävissä ja tiedot päivittyvä kaikkiin oikkein.

7 Yhteenveto

7.1 Tavoitteet ja tulokset

Tavoitteena oli toteuttaa ”vanhaan” maalämpöpumppuun etämonitorointi ja mahdollinen etäohjaus käyttäen jotakin maalämpöpumpun mahdollistamaa liityntärajapintaa.

Lopputuloksena oli monitorointitoteutus, jonka olisi varmasti pystynyt tekemään yksinkertaisemminkin. Toteutus on kuitenkin hyvin skaalautuva, siihen voi helposti integroida lukuisia muita laitteita ja mittauslähteitä. Tiedon tallennuspaikkoja on helppo lisätä ja vaihtaa.

Etäohjaustoiminnallisuutta toteutuksessa ei ole, mutta se oli kehitystyön aikana koko ajan taustalla mielessä. Sekä tiedonkeruu- että pilvipään ohjelmistoissa on varauksia

ja valmista ohjelmakoodiakin etäohjaukselle ja testeissä pilvipalvelusta on pystynyt kirjoittamaan sanomia tiedonkeruupään ohjelmistolle asti. Ainoa väli, mitä ei ole testattu ja mihin ei ole ohjelmistototeutusta on Modbus-kirjoitus maalämpöpumppuun.

Teknologiavalinnat olivat erityisen opettavaiset WebSocket-ohjelmoinnin ja sen käytön osalta. Websocketit olivat uusi ja mielenkiintoinen tuttavuus.

Ehkä hieman yllättäenkin, HTML- ja erityisesti CSS -ohjelmoinnin puolella oli projektin suurimmat haasteet. Nämä johtuivat siitä, että erityisesti CSS-maailma oli vieras ympäristö. Pienempiä haasteita oli varmenteilla toimivan https/websocket-yhteyden rakentamisessa ja konfiguroinnissa.

Kokonaisuutena toteutus onnistui hyvin eikä mitään isoja virhearviointeja tullut. Tärkeät, isossa roolissa olevat teknologiat, Modbus ja WebSocket olivat hyviä valintoja.

7.2 Jatkokehitysideoita

Selkein alue jatkokehitykselle on etäohjaus. Vaikka täydellistä etäohjausta ei usein tarvita, niin ehkä sellainen olisi hyvä olla. Varjopuolena on mahdollinen vihamielinen hyökkäys järjestelmään. Pumpun parametrit säätämällä ns ”tappiin asti”, saadaan helposti vahinkoa aikaan esimerkiksi päästämällä yli 50 asteista vettä lattialämmitystalon lattiaan. Vahinkoa pystyy tekemään myös ajamalla sunttiventtiilit talvella kesäasentoon, jolloin kiinteistö jäähtyy nopeasti. Tällaisten mahdollisuuksien minimoimiseksi etäohjauksen komennot tai parametrien etämuutokset tulisi varmistaa jollakin tavalla. Järjestelmä voisi esimerkiksi lähettää sähköpostin tiettyyn osoitteeseen ja siihen oikealla tavalla reagoimalla, etämuutokset menisivät vasta lämpöpumpulle asti.

Toinen hyödyllinen jatkokehityskohde olisi hälytysten toimittaminen eteenpäin joko sähköpostilla tai tekstiviestillä. Varsinkin talvella olevan pidemmän poissaolon aikana olisi tärkeää saada lämpöpumpun vikatilanteet mahdollisimman pian tietoon.

Hälytyksistä generoituva sähköposti olisi teknisesti varsin helppo toteuttaa heti tiedonkeruupäässä.

Kolmas jatkokehityskohde olisi historiatietojen haku ja lämpötilakäyrien generointi halutulla aikavälillä. Tästä ominaisuudesta olisi hyötyä kun haluttaisiin hienosäätää lämpöpumpun prosessia ja optimoida esimerkiksi pumpun käyntiaikoja.

Neljäs ja mielenkiintoisin kehityskohde olisi lämpöpumpun parametrien säätäminen sääennusteen mukaan. Tämä vaatisi minimissään parin seuraavan vuorokauden lämpötilat ja lisäksi lämpöpumpun parametrien kirjoitusfunktioden toteutuksen.

Lähteet

203GT Lämmönsäädin käsikirja. N.d. Suomen Lämpöpumpputekniikka Oy

History of geothermal energy.N.d. Verkkoartikkeli. Viitattu 2.12.2019.
<https://www.conserve-energy-future.com/geothermalenergyhistory.php>

EH-net. N.d. Esite. Ouman Oy:n verkkosivut. Viitattu 20.11.2019.
https://ouman.fi/documentbank/EH-net__brochure__fi.pdf

EH-203 Lämmönsäädin. N.d. Käsikirja, v2.18b. Ouman Oy:n verkkosivut. Viitattu 20.11.2019. http://ouman.fi/documentbank/EH-203__manual__fi.pdf

EH-200 säätimen Modbus-rajapinta. N.d. Käyttöohje. Viitattu 10.11.2019.
https://ouman.fi/wp-content/uploads/2018/08/MODBUS-200__manual__fi.pdf

HP ProLiant MicroServer Gen8 review. 22.5.2015. Katsaus. Techradar-verkkosivusto. Viitattu 20.11.2019. <https://www.techradar.com/reviews/pc-mac/peripherals/servers/hp-proliant-microserver-gen8-review-1254460/review>

Katsaus lämpöpumppuhistoriaan ja markkinaan. 2019. SULPU 20v Esitelmä. Suomen Lämpöpumppuyhdistys ry:n verkkosivut.
<https://www.sulpu.fi/documents/184029/5177362/B1%20Jussi%20Hirvonen%2C%20SULPUn%20historiaa%2C%20JH.pdf>

Latomäki, J & Vulli, M. 2016. AnturiTietokanta. Advanced Databases -opintojakson harjoitustyön loppuraportti.

Lämpöpumppujen hankintaopas-kunnat ja taloyhtiöt. 2018. Motiva Oy:n verkkosivut. Viitattu 20.11.2019.
https://www.motiva.fi/files/14752/Lampopumppujen_hankintaopas_kunnat_ja_taloyhtiöt.pdf

Lämpöpumppumarkkinat. 2019. SULPU 20v Esitelmä. Suomen Lämpöpumppuyhdistys ry:n verkkosivut.
<https://www.sulpu.fi/documents/184029/5177362/A1%20Jussi%20Hirvonen%2C%20OL%3%A4mp%3%B6pumppumarkkina%2C%2021.11.2019%2C%20f.pdf>

Lämpöä omasta maasta. 2012. Motiva Oy:n verkkosivut. Viitattu 2.12.2019.
https://www.motiva.fi/files/7965/Lampoa_omasta_maasta_Maalampopumput.pdf

Lämpöässä V7.0-V30.0. 2014. Käyttöohje. Suomen Lämpöpumpputekniikka Oy:n verkkosivut. Viitattu 11.11.2019. https://www.lampoassa.fi/wp-content/uploads/2015/08/Lampoassa_V_7-30_2014_04.pdf

Maalämpö. 2019. Wikipedia. Viitattu 2.12.2019.
<https://fi.wikipedia.org/wiki/Maalämpö>

Maalämpöpumppu. 2019. Motiva Oy:n verkkosivut. Viitattu 2.12.2019.
https://www.motiva.fi/koti_ja_asuminen/rakentaminen/lammitysjarjestelman_valinta/lammitysmuodot/maalampopumppu_mlp

Modbus Application Protocol Specification. 2012. Modbus Org verkkosivut. Viitattu 23.12.2019.
http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Modbus FAQ. N.d. Modbus Organization verkkosivut. Viitattu 3.21.2019.
<http://www.modbus.org/faq.php>

Modbus over serial line. 2006. Specification and implementation guide. Modbus Org verkkosivut. Viitattu 3.12.2019.
http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf

Modbus tutorial. N.d. Modbus turorial. RF Wireless World verkkosivut. Viitattu 3.12.2019. <https://www.rfwireless-world.com/Tutorials/Modbus-Protocol-tutorial.html>

openHAB addons. 2019. openHAB verkkosivut. Viitattu 3.12.2019.
<https://www.openhab.org/addons/>

Protocollo Modbus su RS485. 2017. Physical layer. Overdigit.com verkkosivut. Viitattu 3.12.2019. <https://www.overdigit.com/blog/2017/05/22/modbus-su-rs485/>

PyModbus – A Python Modbus Stack. 2017. Esittelysivu. PyModbus verkkosivut. Viitattu 3.12.2019. <https://pymodbus.readthedocs.io/en/latest/readme.html>

Steelseries JavaScript edition. 2011. Harmonic code verkkobloki. Viitattu 4.12.2019.
<https://harmoniccode.blogspot.com/2011/04/steelseries-javascript-edition.html>

Tornado Web Server. N.d. Esittelysivu. Tornado verkkosivut. Viitattu 3.12.2019.
<https://www.tornadoweb.org/en/stable/index.html>

USB-RS485 Comminications Module. N.d. Technical specification. Robot electronics - verkkosivut. Viitattu 20.11.2019. https://www.robot-electronics.co.uk/htm/usb_rs485_tech.htm

Vahala, T. Rakennusautomaation väyläratkaisuja. 2015. Insinööriyö. Viitattu 3.12.2019.
<https://www.theseus.fi/bitstream/handle/10024/91356/Vahala%20Teemu%20-%20Rakennusautomaation%20vaylaratkaisuja.pdf?sequence=1&isAllowed=y>

Tietokannan SQL-malli

