



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Eemil Karppelin

# Omakotitalon lämmityskustannusten optimoiminen automaation avulla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

21.11.2019

Tekijä Otsikko	Eemil Karppelin Omakotitalon lämmityskustannusten optimoiminen automaation avulla
Sivumäärä Aika	61 sivua + 10 liitettä 21.11.2019
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine	Automaatiotekniikka
Ohjaajat	Lehtori Timo Kasurinen
<p>Tämän insinööriyön pääasiallisena tavoitteena oli optimoida kahden omakotitalon lämmityksestä aiheutuvia kustannuksia automaation avulla. Insinööriyö tehtiin kiinteistöistä vastaavalle yksityishenkilölle tilaustyönä. Työ tehtiin kahteen eri kiinteistöön kuitenkin niin että eri sijainnista huolimatta päämäärä kummassakin oli sama eli lämmittämisestä aiheutuvien kustannusten alentaminen. Toinen kiinteistö on vakituksessa asutuskäytössä ja toista käytetään vain kesäaikoina. Toisena suurena tavoitteena työssä oli paikan päältä tulevan informaation analysoiminen ja lämmityksen ohjaus etänä.</p> <p>Opinnäytetyössä molempiin testiympäristöihin mietittiin vaihtoehtoja säästömahdollisuuksille sekä toteutettiin tutkimus, jossa ympärivuotisessa käytössä olevaan omakotitaloon päädyttiin vanhan öljylämmityksen rinnalle asentamaan aurinkoenergiajärjestelmä sekä sähkön hintaa tarkkaileva automatiikka. Vapaa-ajan asunnon testijärjestelmä ratkaisussa päädyttiin automaattiseen kuivanapitolämmitysjärjestelmään, joka seuraa sekä sisä- että ulkolämpötilaeroa, mutta myös sisäilman suhteellista kosteutta ja toteuttaa säädön mittaustulosten perusteella.</p> <p>Saadut tulokset osoittavat, että automaattisella lämmityksenohjauksella ja sen monitoroinnilla voidaan saavuttaa huomattavia säästöjä lämmityskustannuksissa sekä parantaa samalla energiatehokkuutta ja ympäristöystävällisyyttä. Erityisesti toteutetun kuivanapitolämmitysjärjestelmän tuottamat energiasäästöt suhteessa järjestelmän hintaan ovat huomattavia ja järjestelmän takaisinmaksuaika erittäin lyhyt.</p>	
Avainsanat	kuivanapitolämmitys, etäohjaus, energian säästö, hybridilämmitys

Author Title Number of Pages Date	Eemil Karppelin Optimizing Heating Costs of a Detached House Using Automation 61 pages + 10 appendices 21 November 2019
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Engineering
Instructors	Timo Kasurinen, Senior Lecturer
<p>The main goal of this thesis was to lower the heating costs of two detached houses using automation. Study was commissioned by a private person responsible for the properties. Study was done to two different locations and goal was the same on both: lowering the heating costs. One of the properties is used regularly over the year and the other is used for living only in summer time. Secondary goal in this work was to analyze the system data and to control the heating system remotely.</p> <p>In this study, different ways to save energy and costs were clarified. It was decided to install a solar panel system to the regularly lived house, as an addition to the old oil heated boiler. Automatic system, which monitors electric stock market prices and controls the heating based on that was also created. On the other location, the solution was to implement an automatic heating system, which monitors in and outside temperatures, but also the humidity inside, and based on these controls the heating radiators of the house.</p> <p>Results of the study point out that using automated heating control and remote monitoring can make significant difference in consumed energy and heating costs, but also makes the system information more accessible and useful.</p>	
Keywords	Remote control, Energy saving, Hybrid heating system

## Sisällys

Lyhenteet

Tiivistelmä

1	Johdanto	1
2	Omakotitalon hybridilämmitys	2
2.1	Maalämpö	3
2.2	Öljylämmitys	4
2.3	Sähkölämmitys	5
2.4	Poistoilmalämpöpumppu	6
2.5	Aurinkoenergia	7
2.6	Sopivan lämmitysratkaisun valinta	7
3	Kuivanapitolämmitys	9
3.1	Kuivanapitolämmitykseen liittyvät käsitteet	9
3.2	Kuivanapitolämmityksen tavoitteet	10
4	Tutkimusasetelma	12
5	Testijärjestelmä	14
5.1	Komponentit	19
5.1.1	Kapasitiivinen kosteusanturi	19
5.1.2	433 MHz:n radioaaltolähetin	21
5.1.3	Kehitysalusta	22
5.1.4	Aurinkokenno	28
5.1.5	Akusto	29
5.1.6	Invertteri	30
5.1.7	Ryhmäkeskus	32
5.2	Ohjelmisto	35
5.2.1	Palvelin	40
5.2.2	Käyttöliittymä	44

6	Tulokset	51
6.1	Mäntyharjun kiinteistö	51
6.2	Uvilan kiinteistö	56
7	Yhteenveto	59
	Lähteet	61

#### Liitteet

Liite 1. Uvilan ryhmäkeskus ja ohjauskaappi

Liite 2. Keskuksen 3-vaiheinen päävirtapiiri

Liite 3. Keskuksen 1-vaiheinen päävirtapiiri

Liite 4. Palvelimen API rajapinta

Liite 5. Palvelimen pääohjelma (osa)

Liite 6. Uvilan Photonilla ajettava ohjelma

Liite 7. Fortumin hintatiedon välittävä ohjelma

Liite 8. Mäntyharjun Photonilla ajettava ohjelma

Liite 9. Käyttöliittymän HTML koodi

Liite 10. Käyttöliittymän JavaScript koodi (osa)

## Lyhenteet

Hystereesi	Järjestelmän ominaisuus, jonka tarkoituksena on hidastaa muutoksiin reagoimista.
Input/Output	(I/O) Inputit ovat prosessoivaan systeemiin tulevia signaaleita tai dataa. Outputit ovat systeemin lähettämiä signaaleita tai dataa.
Tietokanta	Tietokanta on määrättyä tarkoitusta varten koottu tiedostokoelma, josta hakumenetelmillä saadaan tieto tai tietoyhdistelmä ja jota voidaan muuttaa tai täydentää.
SQL	(Structured Query Language) on kyselykieli, jolla relaatiotietokantaan voidaan tehdä erilaisia hakuja, muutoksia ja lisäyksiä

## 1 Johdanto

Suomessa asumiseen kului kaikkiaan reilu 66 terawattituntia energiaa vuonna 2017. Asumisen energiankulutuksesta noin 68 prosenttia kohdistuu asuinrakennusten ja -tilojen lämmittämiseen. Sähköä tästä on vajaan 23 TWh, josta 48 prosenttia menee tilojen lämmitykseen. (1.)

Lämmitystä tarvitaan niin kokoaikais- kuin vapaa-ajan asunnoissakin ympäri vuoden. Energian säästömahdollisuudet sen sijaan ovat kummassakin erilaiset. Ainoastaan kesäasuntona käytössä olevissa omakotitaloissa suurimmat energiasäästöt saavutetaan käyttämällä jonkin tyyppistä kuivanapitolämmitystä, joka on tärkeää rakenteiden kunnossa pysymisen kannalta. Vakituisesti asutussa omakotitalossa säästöjä voidaan puolestaan kerryttää pidemmällä aikavälillä esimerkiksi vaihtamalla lämmitysjärjestelmää tai lisäämällä vanhan järjestelmän rinnalle nykyaikainen hybridilämmitysratkaisu.

Suomessa on yli puoli miljoonaa vapaa-ajan asuntoa, joista 40 – 50 prosenttia on arvioitu lämmitettävän sähköllä vuoden 2025 loppuun mennessä (2). Suurin osa mökkien sähkökulutuksesta kuluu lämmitykseen ja lämmitysenergian huippu sijoittuu kovimmille pakkaskeleille. Tällöin käyttämättömät kesäasutukseen tarkoitetut asunnot on perinteisesti lämmitetty vakiolämmöllä, jotta rakenteet pysyisivät kunnossa. Tästä aiheutuu tarpeeseen nähden turhaa energian kulutusta ja ylimääräisiä kustannuksia.

Usein ongelmaan haetaan ratkaisua vakioteholämmityksestä, jolla rakennuksen lämmittämiseen kulutettavaa energiaa saadaan vähennettyä. Vakioteholämmitys ei kuitenkaan sovellu kaikkiin rakennuskohteisiin. Automatisoidulla lämmityksenohjausjärjestelmällä, päästään usein parempiin lopputuloksiin ja se on myös sovellettavampi useammassa kohteessa. (2.)

Hybridilämmityksellä tarkoitetaan vähintään kahden eri lämmitysmuodon vaihtelua esimerkiksi vuoden- tai vuorokaudenajan mukaisesti, jolla pyritään hyödyntämään kummankin lämmitysmuodon parhaat puolet ja olosuhteet. Hybridilämmityksen lämmityksen ohjaus on vaativampaa verrattuna yhden lämmitysmuodon järjestelmiin, mutta markkinoille on kuitenkin viime vuosina tullut erilaisia ratkaisuja, jotka vuorottelevat tai täydentävät toisiaan käyttäen automaatiota. (3.) Automaation avulla voidaan esimerkiksi

seurata paljon vaihtelevien suureiden kuten sähkön hintaa ja lämmittää sillä tavalla, joka on kyseisellä hetkellä ostoenergialtaan halvinta.

Kodin lämmityksen valvonta ja etäohjaus ovat vahvasti nykypäivää. Etäohjausjärjestelmät yhdistettynä kodin automatiikkaan tuovat kodin tai vapaa-ajan asuntoon viihtyvyyttä sekä energiatehokkuutta ja sen valvonta mahdollisuuden. Esimerkiksi mobiililaitteella saunan lämmitys pakkasesta tullessa tai lisälämmityksen päälle kytkeminen ennen mökille lähtöä voi lisätä mukavuutta huomattavasti. Talviaikaan asumattomassa rakennuksessa myös ilman kosteuden seuranta voi olla hyödyllistä, jotta voidaan varmistaa rakenteiden ja kalusteiden kunnossa pysyminen talven yli ja reagoida ilmaantuviin ongelmiin nopeasti.

Tässä insinööriyössä käsitellään vapaa-ajan asuntojen kuivanapitolämmitystä ja asutun omakotitalon hybridi lämmitysratkaisuja yleisesti. Lisäksi opinnäytetyössä raportoidaan tutkimuksesta, jolla selvitetään testiympäristöissä tehtyjen toteutusten avulla automaatiolla saavutettavia energian ja kustannusten säästömahdollisuuksia. Testijärjestelmän toteutus kohteista Mäntyharjulta ja Ulvilasta kerätään reaaliaikaista ja historiatyyppistä mitta- ja tilastodataa etäluettavaan muotoon.

Opinnäytetyön tavoitteena on lisätä tietoisuutta kuivanapitolämmityksestä ja omakotitalon lämmityskustannusten alentamisesta automaation avulla sekä helpottaa pienemmän hintatason kotiautomaatiojärjestelmien käyttöönottoa.

## **2 Omakotitalon hybridilämmitys**

Lämmitysjärjestelmä vaikuttaa keskeisesti asumisen mukavuuteen ja siihen liittyviin käyttökustannuksiin. Lämpöä voidaan tuottaa polttamalla erilaisia bio- tai fossiilisia polttoaineita. Lämpöenergiaa voidaan myös ottaa käyttöön auringon lämpösäteilystä suoraltaan tai sen lämmittämästä maasta tai vedestä ja paikoitellen myös geotermisestä energiasta. (4.)

Monessa omakotitalossa on mahdollista vähentää lämmityksestä aiheutuvia kustannuksia vaihtamalla lämmitysjärjestelmä nykyaikaisemmaksi, joko uusimmalla laitteistoa tai lisäämällä esimerkiksi automaatiota vaihtamaan kahden energiamuodon välillä, jotta

saadaan hyödynnettyä molemmista parhaat puolet. Keskimäärin omakotirakentajalla on kaksi eri energianlähdettä. (5.) Mietittäessä lämmitykseen käytettäviä mahdollisia ratkaisuja monille on myös tärkeää lämmitysjärjestelmän ympäristöystävällisyys.

Hybridilämmitys tarkoittaa, että talon vesikiertoisessa lämmönjaossa ja lämpimän käyttöveden tuotossa käytetään hybridivaraajaa, johon on liitetty vähintään kaksi eri lämmönlähdettä. Hybridi lämmitysjärjestelmä on muokattavissa, ja sillä pyritään hyödyntämään mahdollisimman paljon omavaraisenergiaa ostoenergian sijasta. Hybridilämmitysratkaisut voivat tarjota varteen otettavan vaihtoehdon esimerkiksi maalämmölle tai sen rinnalle. Joka vuosi yli 5000 öljylämmittäjää päivittää lämmitysjärjestelmänsä toiseksi. Näin lämmityskulut voivat tippua jopa kolmasosaan aiemmasta. Monissa tapauksissa on järkevää säilyttää silti vanha öljylaitteisto ja liittää sen rinnalle esimerkiksi aurinkokeräimiä. (6)

Tässä luvussa kuvattavat jo itsessään automaatiota sisältävät lämmitysratkaisut voidaan liittää osaksi hybridilämmitysjärjestelmää ja esimerkiksi automaation avulla hyödyntää useammista lämmitysmuodoista saatavia etuja. Monesti järjestelmät myös liitetään osaksi automaatiojärjestelmää monitoroinnin ja säädön helpottamiseksi.

## 2.1 Maalämpö

Maalämpö on uusiutuvaa maaperän tai veden massaan varastoitunutta, lähes kokonaan aurinko- ja osittain myös geotermistä lämpöenergiaa (7). Maan sisäinen lämpö on peräisin radioaktiivisesta hajoamisesta ja maanmuokkautumisesta jatkuvasti aiheutuvista lämpöhäviöistä (8). Maaperä säilyttää sijainnin keskimääräisen vuotuisen lämpötilan, muutamien metrien syvyydessä. Ennen kaikkea talvisin maaperä on merkittävästi ulkoilmaa lämpimämpää. Maalämpöjärjestelmällä saatavaa lämpöenergiaa voidaan käyttää rakennuksen, käyttöveden ja sisäilman lämmittämiseen. Maalämpöä voidaan käyttää kaikentyyppisissä rakennuksissa, ja se on kaukolämpöä edullisempaa elinkaarensa aikana. (7.)

Maalämpöä voidaan tuottaa taloudellisesti kannattavasti hajautetusti tai rakennuskohtaisissa yksiköissä. Järjestelmä on kuitenkin aina sitä kannattavampi, mitä suurempi rakennus ja energiankulutus kohteessa ovat. (7.)

Maalämpöjärjestelmissä käytetään apuna maalämpöpumppua, joka on keskeinen lämmitys ja/tai jäähdytys moduuli, joka siirtää lämpöä maasta tai maahan (9). Lämpöpumpun kompressorin avulla maaperästä saatu lämpöenergia pystytään puristamaan kuumemmaksi ja siirtämään lämmitysjärjestelmään ja lämpimään käyttöveteen (7).

Lämmön siirtämiseen maasta kohteeseen käytetään lämmönkeruupiiriä. Lämmönkeruupiiri on pintamaahan, peruskallioon tai veteen sijoitettava putkisto. Ympäröivä maaperä lämmittelee sen sisällä kiertävää nestettä (7). Kiertäessään putkistossa neste kerää lämpöenergiaa ympäröivästä maa-aineksesta, joka pumpataan lämpöpumpulla lämpövaraajaan. (7)

Maalämpö on yleisin uusissa omakotitaloissa käytetty lämmitystapa (9). Maalämpöä käytettäessä etuina nähdään taloudellisuus, ympäristöystävällisyys ja uusiutuva energianlähde. Haittoina taas ovat suuri alkuinvestointi sekä se, että se vaatii tilaa ja porausta maapiirille. Se valitaan myös tuttavien suosittelun vaikutuksesta. Omakotirakentajista yli 50 % valitsee maalämmön. (5.)

## 2.2 Öljylämmitys

Öljylämmitysjärjestelmän pääosat ovat öljypoltin, öljykattila, lämmönsäätöautomaatiikka, öljysäiliö ja savuhormi. Öljylämmityksen energianlähteenä on jokin maaöljystä jalostettu tuote, joka on yleensä kevyttä polttoöljyä eli lämmitysöljyä. Öljy on energiatiivis sekä helposti varastoitava ja kuljettava energiamuoto. 1000 litraa kevyttä polttoöljyä voidaan verrata energia-arvioltaan 10 000 kWh sähköä, 2150 kiloa puupellettejä tai 5,8 kuutiota koi-vuhalkoja. (10.)

Öljylämmityksestä saatava lämpö siirretään lämmitettävään tilaan vesikiertoisella lämmönjako järjestelmällä tai vaihtoehtoisesti ilmalämmityksellä. Öljylämmityksen toteutus vaatii erillisen palosuojatun teknisen tilan, eli pannuhuoneen, jossa järjestelmän laitteet sijaitsevat. Öljy säiliö voidaan sijoittaa ulos, sisälle tai maan alle. (10.)

Öljylämmityksen hyötysuhde on nykyaikaisella kattilalla noin 95 %. Kondenssikattilalla, joka on kykenevä ottamaan palamiskaasuista otetun lämmön talteen ja syöttämään sen lämmityspiiriin, päästään jopa lähes 100 %:n hyötysuhteeseen. (10.)

Suomessa öljylämmityksen tuottamasta energiasta lämmitykseen kuluu suurin piirtein  $\frac{3}{4}$  ja lämpimän käyttöveden tuotantoon noin  $\frac{1}{4}$ . Öljy on fossiilinen polttoaine, jonka poltto-prosessista aiheutuu ilmastomuutosta nopeuttavia kasvihuonepäästöjä. Öljylämmitys-järjestelmään voidaan kytkeä avuksi aurinkoenergiajärjestelmä tai öljykattilaksi voidaan valita myös puuta polttava kaksoispesäkattila. (10.)

Asuintilojen ja käyttöveden pelkästään öljyllä lämmittäminen ei ole enää nykyaikaista siitä aiheutuvien kasvihuonepäästöjen vuoksi. Nykypäivänä tarjolla on kuitenkin monia mahdollisuuksia tehdä järjestelmästä hybridi, jolloin lämmitykseen käytetään kahta eri energialähdettä. Pääsääntöisesti öljyllä lämmittäminen ei enää sovellu uusien omakoti-talojen lämmitysratkaisuihin (11), mutta Suomessa öljyllä lämmitettäviä rakennuksia on vielä vajaa neljäsosa kaikista rakennuksista, sillä se on rikitöntä ja palaminen nykyaikai-sella lämmitysjärjestelmällä tehokasta (10).

### 2.3 Sähkölämmitys

Sähkölämmitys on rakennuksen lämmöntuottojärjestelmä, jossa lämpö tuotetaan sähköllä. Tässä lämmitysmuodossa ensisijaisena energianlähteenä on sähköenergia, erotuksena muihin lämmitysmuotoihin esimerkiksi maa-, aurinko- tai ilmalämpöön, jotka myös voivat vaatia sähköenergian käyttöä toimiakseen, mutta paljon pienemmän määrän. Sähkölämmitys liittyy lämmön tuottotapaan. Lämmönjako voi tapahtua muillakin menetelmillä kuin sähköä hyväksi käyttäen, mutta myös tällöin puhutaan sähkölämmityksestä. (12.)

Lämmitykseen on mahdollista käyttää suoraa, varaavaa tai osittain varaavaa sähkölämmitysjärjestelmää. Suorassa sähkölämmitysjärjestelmässä lämmitys tapahtuu heti, tarpeen mukaan esimerkiksi termostaatilla ohjattuna. Lämpö luovutetaan halutussa sijainnissa esimerkiksi pattereilla. (12.)

Varaavassa sähkölämmitysjärjestelmässä pyritään hyödyntämään varaavaa massaa, kuten esimerkiksi lattialaatta tai vesisäiliö ja ennen kaikkea sähkön hinnan tariffit. Massa varataan lämpöä silloin kun sähkön hinta on halpaa, esimerkiksi yö aikaan. Massa luovuttaa lämmön päivän mittaa ja se varataan jälleen yö aikaan. (12.) Osittain varaava järjestelmä on aikaisempien sekoitus.

Päälämmitystapana sähköä käytetään uusissa omakotitaloissa enää alle 10 % tapauksista. Sähkölämmitys on todella helppohoitoinen ja edullinen hankkia ja siitä on paljon aikaisempia kokemuksia (11). Toisaalta jos sähköä käytetään päälämmitystapana käyttökustannukset nousevat korkeammaksi, kuin muilla yleisillä lämmitystavoilla (12).

#### 2.4 Poistoilmalämpöpumppu

Poistoilmalämpöpumppu (PILP) ottaa lämmitysenergiaa talteen talosta poistettavasta ilmasta. Lämpöpumppu siirtää lämpöä tuloilmaan, lämpimään käyttöveteen ja/tai vesikiertoiseen lämmitysjärjestelmään. Tämä järjestelmä vaatii aina, että ilmaa vaihdetaan riittävästi. Kannattavuus poistoilmalämpöpumpulle riippuu rakennuksen muusta lämpötaloudesta. Rakennuksen ilmanvaihto toteutetaan myös poistoilmalämpöpumpulla. Usein poistoilmalämpöpumppua ei käytetä ainoana lämmönlähteenä vaan mukana on vähintään yksi apulämmitin. (13.)

Lämmönlähteenä on aina talon sisäilma, joka on noin 21 asteista, minkä vuoksi lämpöpumppu tuottaa lämpöä vakioteholla vuodenajasta ja ulkolämpötilasta riippumatta. Poistoilmalämpöpumppu on ilmanvaihtojärjestelmien suhteen hyvin mukautuvainen ja siksi se on mahdollista asentaa myös jälkiasennuksena vanhoihin rakennuksiin vähäisin muutoksia. (13.)

Poistoilmalämpöpumppu on kohtalaisen edullinen hankinta- ja käyttökustannuksiltaan. Se on usein talotehtaan suosittelema ja helppohoitoinen lämmitysmuoto (5). Poistoilmalämpöpumppu on usein hyvä valinta uusiin, hyvin eristettyihin taloihin koska näissä tilojen lämmitys ei vaadi suurta energiamäärää vuositasolla. Poistoilmalämpöpumpulla ei kuitenkaan voida tuottaa kaikkea talon tarvitsemaa energiaa (14). Omakotitalonrakentajista noin 18 % hankkii poistoilmalämpöpumpun (5).

## 2.5 Aurinkoenergia

Aurinkoenergiassa auringon säteilemä energia käytetään hyväksi sähkö- tai lämpöenergiana. Monesti tällä tarkoitetaan suoraa säteilyenergian hyödyntämistä aurinkokennon tai -keräimen avulla. Aurinkoenergia on uusiutuvaa energiaa, jonka tuotannosta syntyy jätettä sekä päästöjä vain valmistus vaiheessa sekä kierrätyksessä. Aurinkoenergia on pitkään ollut suhteellisen hintavaa sen hyödyntämiseen tarvittavien aurinkopaneelien vuoksi, mutta hinta on kuitenkin ollut laskusuunnassa lähes fossiilisten polttoaineiden tasolle. (15)

Aurinkosähköä tuotetaan tavallisesti aurinkokennoilla ja nykypäivänä jopa yli 80 prosenttia niistä on liitettyä sähköverkkoon. Aurinkokennoja on mahdollista käyttää myös verkosta irrallaan ja akkujen avulla turvata sähkön saannin jatkuvuutta. (15.)

Aurinkoenergian positiivisina puolina pidetään saasteettomuutta, sitä että se on käytössä ilmaista sekä siitä saatavaa ennustettavuutta sähkön hankintaan. Huonoina puolina nähdään investointikulujen korkeus ja se että aurinkoenergia soveltuu Suomessa ainoastaan täydentäväksi energiamuodoksi. Suomessa eniten energiaa aurinkopaneelit tuottavat kesällä ja energiahuippu kohdistuu kesä – heinäkuuhun. Talvella energian tuotanto paneeleista on vähäistä. (16.)

## 2.6 Sopivan lämmitysratkaisun valinta

Omakotirakentajien tärkeimmät viisi perustetta lämmitystavan valinnassa ovat huolettomuus/vaivattomuus, edulliset käyttökustannukset, helppohoitoisuus, ympäristöystävällisyys, sekä se ettei näkyviä pattereita saa olla (5). Uusissa omakotitaloissa kannattavimpana lämmitysratkaisuna pidetään hybridilämmitystä (17).

Suuri osa uutta lämmitysratkaisua miettivistä omistaa kuitenkin vanhemman omakotitalon, mutta haluaa kehittää talon lämmitysratkaisua säästöjen ja ympäristöystävällisyyden vuoksi. Näissä tapauksissa tilannetta voidaan tarkastella esimerkiksi laskemalla lämmityksestä aiheutuvat kumulatiiviset kustannukset 15 vuoden ajalle.

Taulukko 1. 1960-luvun talon suuntaa antavat investointi- ja käyttökustannukset. (18)

Lämmitysmuoto	Investointikustannus	Käyttökustannus / vuosi	Kumulatiiviset kustannukset 15
Maalämpö	15 000 €	1 511 €	37 665€
Kaukolämpö	7 500 €	2 906 €	51 090€
Öljy	9 000 €	3 015 €	54 225€
Sähkö	4 000€	4 016 €	64 240€
Poistoilma-lämpöpumppu ja sähkö	10 000 €	2 514 €	47 710€

Taulukko 1 kertoo esimerkkilaskelmasta kunkin lämmitysmuodon suuntaa antavat investointikustannukset ja lämmityskulut 1960-luvulla rakennetulle omakotitalolle. Laskelmat on tehty neljän hengen esimerkkiperheelle, joka asuu Keski-Suomessa. (18.)

Laskelmista voidaan helposti päätellä, että maalämpöpumppu on selvästi halvin tapa lämmittää omakotitaloa pitkällä aikavälillä, mutta samalla se on suurin kertainvestointi. Otettaessa huomioon investointi ja lämmityskulut maalämpöpumppu tulee edullisemmaksi 1960-luvulla rakennetussa omakotitalossa jo alle kymmenessä vuodessa. Kallein lämmitysmuoto pitkällä aikavälillä vertailun mukaan on sähkö, jonka alkukustannukset ovat vertailun halvimmat, mutta käyttökustannukset suurimmat. (18.)

Esimerkki laskelmissa on kuitenkin otettava huomioon, että ne ovat vain suuntaa antavia ja että todellisuudessa lopputulokseen vaikuttaa monta muuttuvaa ja vaikeasti

ennustettavaa tekijää. Laskennassa ei myöskään ole otettu huomioon korkoja, kun lasketaan kumulatiivisia kustannuksia useiden vuosien ajalta. (18.)

Aurinkosähköjärjestelmän kustannuksista noin 40 % tulee työstä ja 60 % materiaaleista. Hinta koostuu aurinkopaneeleista, muista laitteista ja materiaaleista kuten inverttereistä ja sähköjohdoista sekä suunnittelusta ja asennuksesta. Vuonna 2014 kokonaishinta Suomessa FinSolarin raportin mukaan oli 1,21–1,8 €/Wp. (19.)

Testijärjestelmän osalta aurinkoenergiajärjestelmään päädyttiin toteamalla järjestelmä nykyaikaiseksi täydentäväksi energiamuodoksi, joka on helppohko asentaa kohteeseen siihen soveltuvien tilojen ja komponenttien ollessa valmiina. Toisena vaikuttavana tekijänä nähtiin järjestelmän kokeilun hyvä mahdollisuus ja helppo laajennus mahdollisuus positiivisten tulosten ilmetessä. Lisäksi aurinkoenergiajärjestelmä todettiin soveltuvaksi öljylämmityksen mahdollisesti tulevaisuudessa korvaavan maalämpöjärjestelmän rinnalle. Näistä syistä aurinkoenergia nähtiin kokonaisinvestointina optimaalisimpana vaihtoehtona kohdekiinteistölle Ulvilassa.

### 3 Kuivanapitolämmitys

Asumattomassa ja lämmittämättömässä asunnossa sisäilman suhteellisen kosteuden nousu saattaa aiheuttaa rakenteisiin pysyviä kosteusvaurioita, joita on kallista ja hankalaa korjata. Kuivanapitolämmitys on lämmitysmuoto, joka tähtää rakenteiden ja kalusteiden kunnossa pysymiseen energiaa säästäen.

#### 3.1 Kuivanapitolämmitykseen liittyvät käsitteet

*Vakioteholämmityksellä* rakennus pyritään pitämään kuivana talven yli asettamalla sisätiloihin nimensä mukaisesti lämmitys, joka lämmittää jatkuvasti samalla teholla ja pitää näin sisä- ja ulkolämpötilaeron muutamassa asteessa. Lämmityksen vaatima teho määräytyy muun muassa alapohjan tyypistä ja sen kyvystä tuulettua. Noin kolmen asteen lämpötilaero saavutetaan 5–10 W:n lämmitysteholla neliötä kohden. (20.)

Perinteisesti mökissä ollaan totuttu pitämään *peruslämpöä*, joka on usein 5–7 astetta. Tämäntyyppinen ratkaisu ei kuitenkaan ota millään tavalla huomioon ulkolämpötilaa ja etenkin pakkasilla siitä aiheutuu huomattavaa liikalämmitystä.

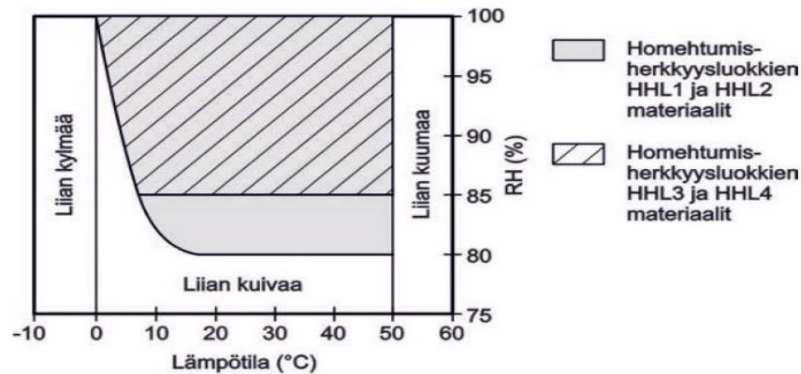
*Suhteellinen kosteus* tarkoittaa tämän hetkistä absoluuttista kosteutta suhteessa korkeimpaan mahdolliseen absoluuttiseen kosteuteen. Korkein mahdollinen kosteus on riippuvainen vallitsevan ilman lämpötilasta. 100 prosenttinen suhteellinen kosteus tarkoittaa, että ilma on täysin kylläinen vesihöyryä eikä voi varastoida enempää, jolloin syntyy mahdollisuus sateelle. (21.) Sisäilman suhteellinen kosteus vaikuttaa oleellisesti rakenteiden ja kalusteiden säilymiseen (2). Useimmiten kaikista kriittisintä aikaa ovat syksy-, kevät- sekä talviaika silloin kun ei ole pakkasta.

### 3.2 Kuivanapitolämmityksen tavoitteet

Kuivanapitolämmityksen ideana on sisätilan lämpötilan pitäminen koko talvikauden ajan vain joitakin asteita ulkolämpötilaa korkeammalla (22). Tämä tarkoittaa käytännössä sitä, että sisälämpötila saa ajoittain painua vaikka pakkasen puolelle. Tästä huolimatta ulko- ja sisälämpötilaero on riittävä estämään kosteuden aiheuttamat haitat, ja energiaa voi säästyä jopa satoja euroja talvikaudessa verrattuna peruslämmön ylläpitoon. (22.) Painovoimaisessa ilmanvaihdossa tämä on myös tärkeää, jotta ilma kiertää suunnitellusti ja rakenne pääsee kuivumaan oikein (2).

Kuivanapitolämmitysratkaisun, jossa lämpötila päästetään ajoittain laskemaan pakkaselle, on otettava huomioon muun muassa tilassa sijaitsevat hanat, putket, vesivaraajat ja pesukoneet. Nämä laitteet on voitava tyhjentää vedestä, jotta pakkasen ei pääse rikkomään niitä. Useimmat elektroniset laitteet puolestaan sietävät pakkasta, kunhan niiden annetaan lämmitä rauhassa ennen käyttöä. (22.) Kuivanapitolämmitykseen tarvitaan lämmitin tai lämmittimiä sekä mahdollisuus säätää niitä. Nykyään useissa ratkaisuissa käytetään jonkin tyyppisiä antureita ja tekoälyä ohjamaan lämmittimiä. (22.)

Kuivanapitolämmitystä mietittäessä on tärkeää ottaa huomioon homeen kasvu mahdollisuudet kussakin tapauksessa. Kuten kuvasta 1 voidaan havaita, VTT:n teettämän tutkimuksen mukaan homeen kasvu on mahdollista 0...50 °C lämpötilassa silloin kun suhteellinen kosteus prosentti on yli 80.



Kuva 1. Suomalaisen homeallin kuvaamat suhteellisen kosteuden ja lämpötilan raja-arvot mikrobikasvulle. (23)

Kyseisen tutkimuksen teettämän suomalaisen homekasvumallin perusteella homekasvua ei ole havaittavissa, jos sisäilman lämpötila pysyy normaaleissa 5–20 asteen lämpötiloissa ja suhteellinen kosteus alle 80 %. Kuivanapitolämmityksellä tavoitteena on pysyä noin 10–15 % riskirajan alapuolella, jolloin pienet säätövirheet voidaan sallia ja huomioidaan myös mahdolliset mittauspisteitä kosteammat kohdat rakenteissa ilman riskejä (2).

Lisäksi Tampereen teknillisen yliopiston teettämässä tutkimuksessa ollaan havaittu, että jo 3–5 asteen lämpötilaero on riittävä säilyttämään sisäilman suhteellisen kosteuden turvallisesti todetulla 60–70 prosentin tasolla ja näin ollen rakenteet eivät pääse homehtumaan. (20)

Näiden tutkimusten perusteella voitaisiin kehittää useampikin säätömalli, jolla varmistuttaisiin, että homeetta ei pääse syntymään. Esimerkkinä voitaisiin seurata sisä- ja ulkolämpötilojen eroa ja kun ero on liian pieni, käynnistetään lämmitys. Toisena esimerkkinä ohjaukselle olisi laskea kosteusarvo kussakin lämpötilassa, säätää se turvalliselle tasolle ja tarkkailla myös lämpötilaeroa, jolla pidetään huolta painovoimaisesta ilmanvaihdosta. Tällöin lämpötilaero voitaisiin tiputtaa vielä ensimmäistäkin esimerkkiä hieman alhaisemmaksi ja saavuttaa lisäsäästöjä. Tämäntyyppinen ratkaisu vaatii kuitenkin enemmän anturointia ja laskentaa kuin pelkkä lämpötilaeron seuraaminen.

Aiemmin kuvattuihin ongelmiin on useita erilaisia mahdollisia lämmitystapoja. Tilojen kuivanapidossa lämmittimenä voidaan käyttää esimerkiksi diesel- tai sähkökäyttöistä mökilämmintä ja/tai valtakunnanverkkoon kytkettäviä sähköpattereita.

#### **4 Tutkimusasetelma**

Tässä työssä tehtiin tutkimus, jossa selvitettiin erilaisia tapoja kohdekiinteistöjen lämmityksestä aiheutuvien kustannusten optimoimiseen ja ympäristöystävällisyyden parantamiseen automaation avulla. Tutkimus tehtiin kahteen eri omakotitaloon, joista toinen oli vapaa-ajan käytössä ja toinen ympärivuotisessa asutus käytössä. Toisena tavoitteena työssä oli mahdollistaa järjestelmän tuottaman datan lukeminen etänä analysoinnin helpottamiseksi sekä vikatilanteisiin reagoinnin ja nopeuttamiseksi.

Tutkimus sijoittui varsinaisesti ajalle 9.9.2019–08.11.2019, joka on osaltaan lyhyt aika lopullisten tulosten määrittämiseksi, mutta tältä ajalta saaduista tuloksista voidaan laskea aiempien tutkimusten perusteella pidemmän aikavälin suuntaa antavat arviot. Tutkimuksesta saatavia tuloksia verrataan kiinteistönomistajan hallussa oleviin edellisten vuosien vastaavalta ajalta mitattuihin energiankulutuksiin.

Työn tarkoituksena oli selvittää, onko automaattisen kuivanapitolämmitys- ja aurinkoenergiajärjestelmien toteuttaminen ja lämmitystavan vaihtaminen öljyn ja sähkön välillä taloudellisesti kannattavaa. Toisena päätavoitteena oli järjestelmien hallinnan ja seuraamisen helpottaminen ja sen etänä mahdollistaminen. Lisäksi tarkoituksena oli lisätä tietämystä omakotitaloon asennettavan kuivanapitolämmityksen sekä aurinkoenergiajärjestelmän toiminnasta.

## Mäntyharjun kiinteistö

Mäntyharjun omakotitalo sijaitsee Etelä-Savon maakunnassa, joka on yksi Suomen suosituimmista mökkikunnista. Kiinteistö on ainoastaan kesällä asutuskäytössä, jolloin suurin osa rakennuksen energian kulutuksesta menee rakennuksen ollessa tyhjillään. Kiinteistön omistaja on kiinnostunut laskemaan rakennuksen lämmityksestä aiheutuvia kuluja. Tähän asti kiinteistöä ollaan lämmitetty peruslämpö periaatteella, pitäen huoneiston lämpötila termostaatilla noin 13 asteessa. Tämäntyyppinen ratkaisu ei kuitenkaan ole ympäristö- eikä energiaystävällinen.

Kiinteistö sijaitsee huomattavan matkan päässä kiinteistön omistajan vakituisesta asuinpaikasta (Helsingistä) ja huoneiston tilanteesta halutaan pysyä jatkuvasti ajan tasalla. Näistä syistä rakennuksen mennyttä ja nykyistä kosteuden ja lämpötilan tilaa halutaan pystyä seuraamaan ja tarpeen tullessa säätämään etänä mobiililaitteella.

## Uvilan kiinteistö

Uvila on Satakunnan maakunnassa noin kymmenen kilometriä Porista kaakkoon sijaitseva kaupunki. Kuten edellisessä sama kiinteistön omistaja on päättänyt optimoida kiinteistön lämmitykseen käytettäviä kuluja ja lisätä ympäristöystävällisyyttä mutta haluaa myös mahdollisuuden seurata järjestelmän toimintoja, kuten tuotettua energiamäärää sijainnista riippumatta.

1970-luvulla rakennetun kiinteistön asuintilojen patteriverkoston vettä sekä käyttövettä on mahdollista lämmittää polttoöljyn tai sähköisen lämmitysvastuksen avulla. Tämä avaa mahdollisuuden vaihtaa lämmitystapaa käyttäen automaatiota. Öljylämmityksestä halutaan myös osittain siirtyä käyttämään ympäristöystävällisempiä vaihtoehtoja.

Pörssisähkön spot-hinta vaihtelee paljon kysynnän ja tarjonta mahdollisuuksien mukaan. Vaikuttavia tekijöitä ovat muun muassa suuret teollisuuslaitokset sekä vuorokauden- ja vuodenaajat. Spot-markkinoilla hinta määräytyy Pohjoismaiden ja Baltian maiden yhdessä omistamasta sähköpörssissä Nord Poolissa (24). Esimerkkinä hinnan vaihtelun suuruudesta voidaan ottaa 22.05.2019, jolloin hinta vaihteli yhden päivän aikana 29,00 ja 199,91 EUR/Mwh välillä (25).

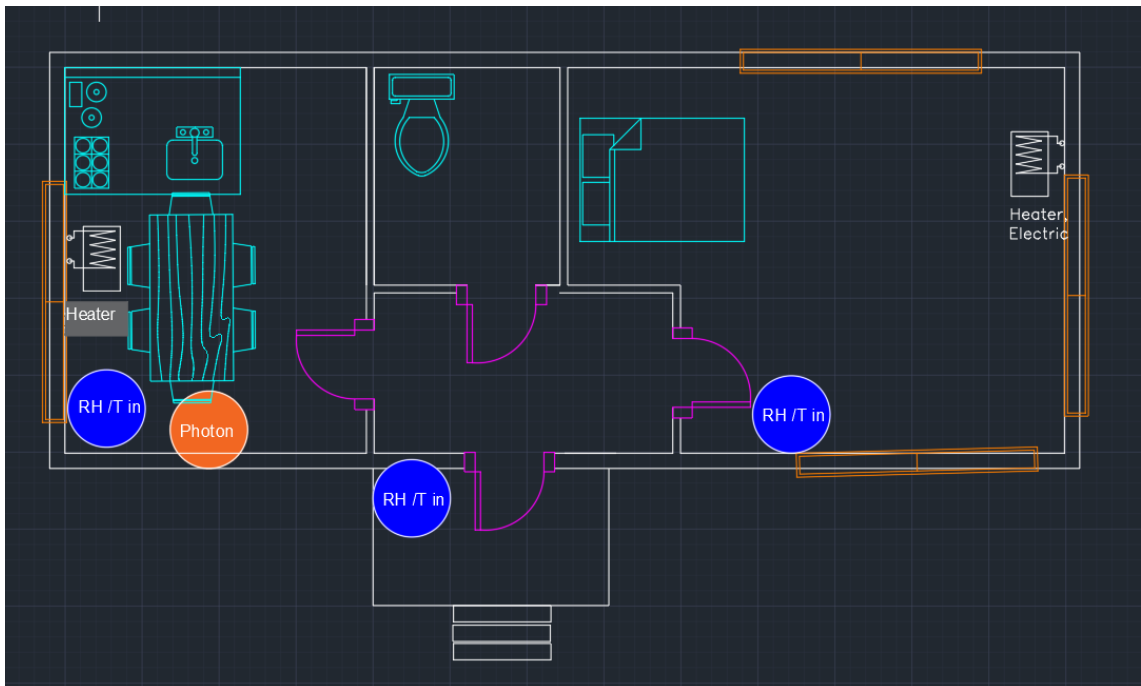
Tutkimuksen tavoitteena on selvittää, millaisia säästöjä automaation avulla mahdollista saavuttaa vaihtamalla lämmitystapaa sähkön ja öljyn välillä sähkön senhetkisen hinnan perusteella. Toisena päätavoitteena on selvittää käytännössä aurinkoenergiajärjestelmän taloudellinen ja ympäristöystävällinen kannattavuus sekä helpottaa järjestelmän tuottaman datan tulkitsemista ja siihen käsiksi pääsyä.

## 5 Testijärjestelmä

Tässä luvussa pureudutaan testijärjestelmän toimintaan ensin pääpiirteittäin ja sen jälkeen tarkemmin komponenttitasolla. Suuressa roolissa myös tehty ohjelmisto ja sen yksityiskohtainen toiminta.

Järjestelmä toteutettiin vastaamaan aiemmin tutkimusasetelmassa kuvattuihin ongelmiin ja vaatimuksiin. Järjestelmän toteutuksessa on huomioitu tilaajan tarve monitoroida ja hallinnoida kahta eri kiinteistöä mahdollisimman helposti ja nopeasti sijainnista riippumatta. Näin ollen toteutuksessa kummastakin sijainnista tuotettu data on tehty saman käyttöliittymän avulla etäluettavaksi. Käyttöliittymästä pystytään myös vaikuttamaan Mäntyharjun automaattisen säädön käyttäytymiseen sekä asettamaan lämmitys päälle tai pois manuaalisesti.

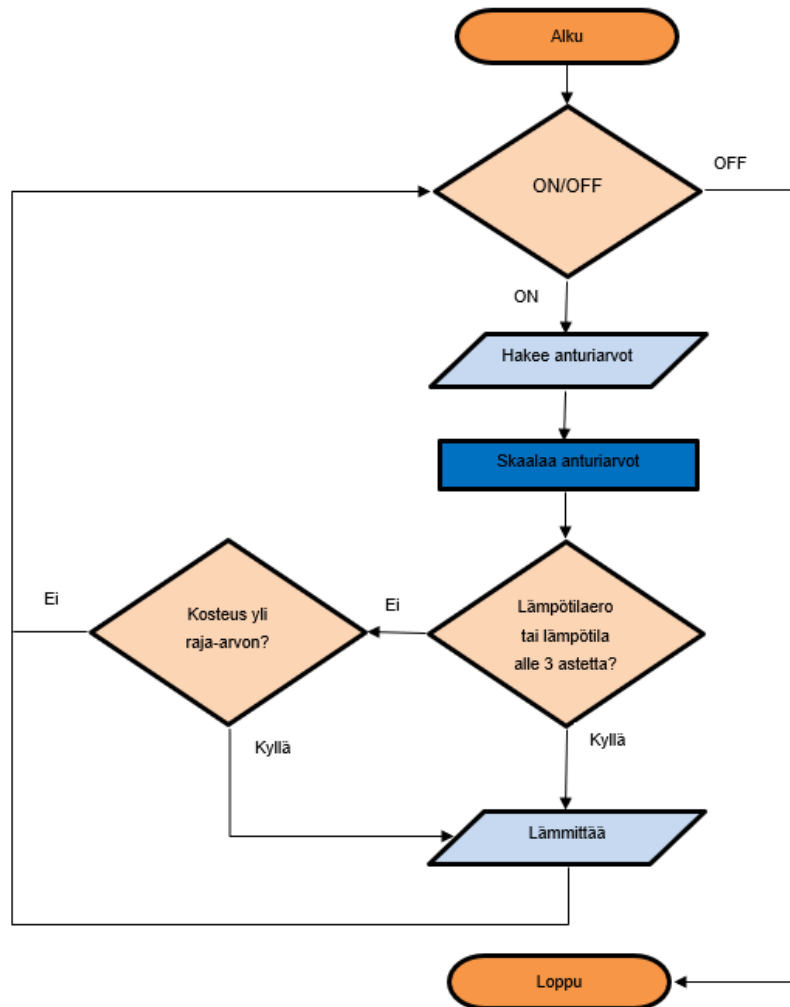
Testijärjestelmän monet osat ovat riippuvaisia Photonin toiminnasta, jonka ympärille Mäntyharjun järjestelmä (kuva 2) rakentuu. Photon on esineiden internet-alusta, joten se soveltuu tilaajan tarpeisiin erinomaisesti.



Kuva 2. Komponenttien suunniteltu sijoittelu Mäntyharjulla

Mikrokontrolleri asennettiin keittiöön mahdollisimman vähän näkyvälle paikalle. Kuvassa 2 näkyvät lämpötila ja kosteusanturit on pyritty sijoittelemaan lähelle lattiaa ja siten, että ne havaitisivat mahdollisimman hyvin koko rakennuksen kosteuden tilan, jotta lämmitystä osataan säätää oikein. Kuvassa näkyvät myös sähköiset lämmityspatterit, joiden pistorasioita ohjataan langattomasti Photonilla radiolähetintä apuna käyttäen. Radiolähetin ollaan asennettu Photonin kanssa yhteiseen kytkentäalustaan.

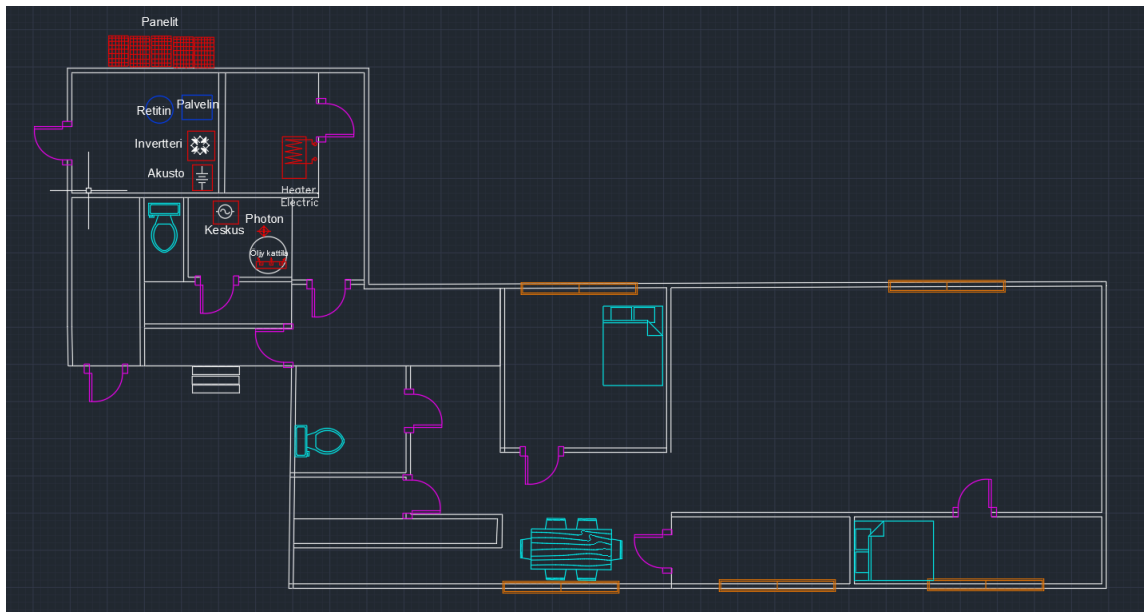
Photon ohjaa kiinteistön lämmitystä pyrkien pitämään sisäilman suhteellisen kosteuden turvallisissa rajoissa. Lisäksi Photonin ohjaamalla säädöllä pyritään varmistamaan huoneiden painovoimaisen ilmanvaihdon oikeanlainen toiminta. Lämmitystä on mahdollista tarkkailla ja ohjata käyttöliittymän (s.46, kuva 18) avulla joko manuaalisesti tai automaattisesti. Automaattijolla lämmitystä ohjataan Kuvassa 3 näkyvän vuokaavion mukaisesti.



Kuva 3. Vuokaavio Mäntyharjun lämmitysjärjestelmän ohjausmallista automaattijolla

Kun laite kytketään päälle ja automattiajolle se alkaa hakemaan huoneiden kosteus- ja lämpöarvoja toistuvasti. Järjestelmä priorisoi minimaalista lämpötilaeroa sisä- ja ulkoilman välillä ja lämmitää eron ollessa alle 3 astetta, joka on riittävä, koska kosteutta ei päästetä nousemaan haitallisen korkealle. Järjestelmä ei kuitenkaan päästä huoneiden lämpötilaa laskemaan pakkaselle, vaan lämmitää lämpötilan laskiessa alle 3 asteen. Järjestelmä tarkkailee myös mitattua kosteusarvoa, jonka noustessa asetettua rajaa korkeammaksi käynnistetään lämmitys. Photonin ohjelman yksityiskohtainen toiminta esitellään liitteessä 8.

Säädössä lämmitystä ohjaavissa ehdoissa on hyvä huomioida noin 2 asteen hystereesi.



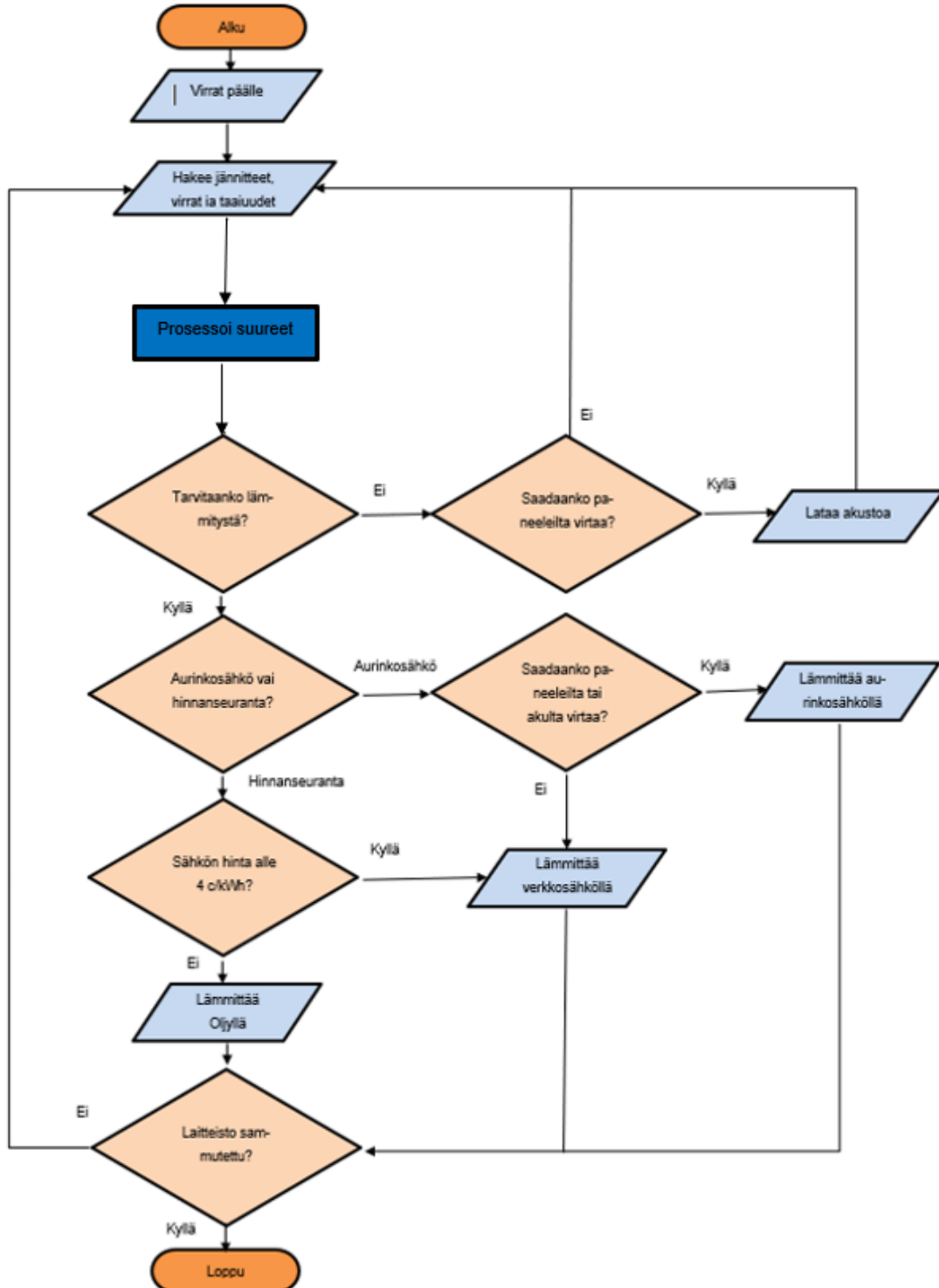
Kuva 4. Komponenttien sijoittelu Ulvilassa

Uvilan lämmitysjärjestelmän ohjaamisen kannalta keskeisimpinä komponentteina voidaan pitää Kuvassa 4 näkyvää invertterinä ja ryhmäkeskusta sekä sen sisään kytkettyä Photonia. Kuvasta nähdään myös muut järjestelmän datan käsittelyyn, energian varastointiin, -tuottamiseen ja -kulutukseen käyttämät komponentit.

Komponenttien sijoittelussa pyrittiin käyttämään suurimmaksi osaksi tyhjillään olevaa varastointitilaa, joka on myös keskeisellä paikalla lämmityskattilaa ja kiuasta (kuvassa Heater electric) ajatellen. Keskus asennettiin lämmityskattilan viereen, jolloin hyödynnettiin mahdollisuus käyttää aikaisemmin vedettyjä kaapeleita. Photon asennettiin keskuksen sisään, jolloin tilaa saatiin säästettyä ja visuaalisuutta parannettua. Paneelit suunnattiin etelään noin neljänkymmenen asteen kulmassa maanpintaan nähden, jolloin niistä saatava teho on teoriassa suurimmillaan kyseisessä sijainnissa.

Järjestelmän etäluettavaksi keräämää dataa havainnollistaa käyttöliittymän ”Inverter View” -näkyvä, (s.48, kuva 20) jossa näkyvät arvot tuotetaan invertterillä, josta ne kysellään palvelimelle RS-232 -sarjaväylän kautta. Tarkempaa kuvausta järjestelmien komponenttien välisestä kommunikaatiosta ja tiedonsiirrosta ohjelmisto-osiossa.

Lämmitysjärjestelmällä voidaan lämmittää 3 eri tavalla: Öljyllä, Aurinkosähköllä tai verkkosähköllä. Hyödyntäen tämän tyyppisen hybridilämmityksen tuomia mahdollisuuksia, järjestelmällä pyritään vähentämään öljyllä lämmittämisen määrää seuraavan vuokaavion 5 kuvailemin tavoin.



Kuva 5. Vuokaavio Ulvilan lämmitysjärjestelmän ohjausmallista

Kun Järjestelmä käynnistetään Invertteri alkaa hakea paneelien, akuston ja verkon suureita ja ne prosessoidaan. Jos lämmityskattilan lämmitysveden lämpötilaksi mitataan asetettua arvoa pienempi, on vettä syytä alkaa lämmittämään. Järjestelmän toteuttaman ohjauksen peruste valitaan paikalliselta ryhmäkeskukselta (s.33, kuva 13). Sähköllä lämmitys voidaan toteuttaa joko sähkön hinnan perusteella tai käyttämällä aurinkoenergiaa. Kun keskukselta ollaan valittu käytettäväksi aurinkosähköä, invertteri tutkii, saadaanko virtaa suoraan aurinkopaneeleilta tai akustolta. Jos ei, invertteri syöttää kuorman ottaman virran verkosta. Jos puolestaan keskukselta ollaan valittu hinnan perusteella ohjattu lämmitys, vertaa Photon sähkön c/kWh hintaa asetettuun raja-arvoon ja kytkee lämmityksen, joko verkkosähkölle tai verkkosähkön pois päältä. Jos verkkosähköllä ei lämmitetä ja öljykattilan veden lämpötila laskee, kytkeytyy öljypoltin päälle oman termostaatinsa vaikutuksesta. Jos lämmitys tarvetta ei esiinny, mutta paneelit tuottavat silti energiaa se varastoidaan akustoon.

## 5.1 Komponentit

Tässä luvussa käydään läpi testijärjestelmään kuuluvat pääkomponentit, niiden toiminnan perusteet sekä se, miten mitäkin komponenttia on ratkaisussa hyödynnetty. Tässä osiossa käydään myös läpi mitä komponentin valinnassa ollaan otettu huomioon ja perustellaan komponentti valinnat.

### 5.1.1 Kapasitiivinen kosteusanturi

Kapasitiiviset suhteellisen kosteuden anturit ovat laajalti käytössä muun muassa teollisuudessa ja sään tulkinta sovelluksissa (26). Niiden toiminta perustuu nimensä mukaisesti kapasitanssin muutokseen, jonka anturi havaitsee.

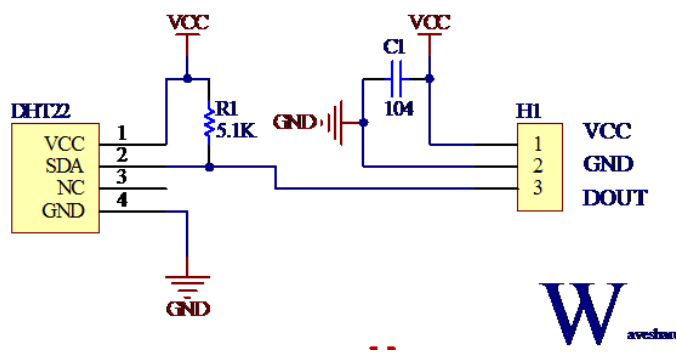
Anturit koostuvat kahdesta lasisesta tai keraamisesta elektrodista, joiden välissä ohut polymeerikalvo on eristeenä. Ohutkalvopolymeeri absorboi tai vapauttaa vesihöyryä seuraten ympäröivän ilman suhteellista kosteutta. Sen eristeominaisuudet riippuvat siihen absorboituneen veden määrästä, jolloin anturin kapasitanssi muuttuu, kun anturia ympäröivä suhteellinen kosteus muuttuu. (27.)

Anturi pystyy muuttamaan kapasitanssiarvon AD-muuntimen tai muun IC:n avulla digitaaliseen muotoon ja tällöin kapasitanssiarvo tai jo kosteuslukemaksi muutettu arvo voidaan lähettää mittalaitteelle. (27.)

DHT22-lämpötila- ja kosteusanturi

DHT-kosteusanturien tuoteperhe koostuu kapasitiivisista kosteusantureista ja termistoreista (28). Niiden koteloiden ulkomitat ovat noin 15 mm x 25 mm, josta ulos tulee tyypillisesti kolme käytettävää johdinta. Nämä ovat virransyöttö-, datan ulostulo- ja mittausjohdin. DHT22-anturin lämpötilan mittausalue ulottuu -40:stä 125:een asteeseen celsiusista, 0.5 asteen tarkkuudella. Sillä voidaan myös mitata kosteutta koko alueelta 0:sta 100 prosenttiin 2 – 5 prosentin resoluutiolla. (28)

### Temperature-Humidity Sensor (B)



Kuva 6. DHT22 anturin kaavio kuva (29)

Toiselta nimeltään AM2302 ei tarvitse kommunikaation osalta kuin yhden tiedonsiirtojohdinten tiedon siirtoon ja sen kontrolloimiseen. Anturille lähetään aloituspulssi, ja se vastaa 40-bittisellä paketilla, joka sisältää sekä kosteuden että lämpötilan. Usein oikeanlaisen toiminnan varmistamiseksi suuruudeltaan noin  $5k\Omega$  –  $10k\Omega$  ulkoinen vastus kytketään tiedonsiirto ja virransyöttö johtimien välille (kuva 6). (29.)

Lämpötilan ja suhteellisen kosteuden anturien valinnassa huomioitiin erityisesti kohteeseen soveltuvuutta. Tämä tarkoitti tässä tapauksessa sitä, että signaalin voimakkuuden oli oltava riittävä, koska paikoittain vedot saattoivat olla suhteellisen pitkiäkin, käyttäjännitteeseen nähden anturin ja kontrollerin välillä. Toisena tärkeänä tekijänä pidettiin

anturin antamien mittaustulosten tarkkuutta. Edellä mainituista syistä päädyttiin DHT-tuoteperheen hintavampaan malliin AM2302:een. Anturit on sijoitettu mittaamaan kosteutta makuuhuoneesta ja keittiöstä läheltä lattian rajaa, koska siellä arvot nousevat todennäköisimmin kriittiselle alueelle.

### 5.1.2 433 MHz:n radioaaltolähetin

Radioaaltomoduuli (RF module) on pieni elektroninen laite, jota käytetään lähettämään tai vastaanottamaan radio signaaleja kahden laitteen välillä. Sulautetuissa järjestelmissä halutaan usein suorittaa kommunikointi laitteiden välillä langattomasti. Langaton kommunikointi voidaan suorittaa käyttäen optista- tai radiotaajuuskommunikointia. Monissa sovelluksissa suositaan radiotaajuutta, koska siinä ei tarvita näköyhteyttä. (30.)



Kuva 7. 433 MHz:n langaton radiotaajuuslähetin (31)

Radiotaajuuslähetysohjausmoduuli (kuva 7) on painettu piirilevy, joka on kykenevä lähettämään ja mukauttamaan radioaaltoa, jotta dataa pystytään kuljettamaan. Lähettimiä käytetään usein mikrokontrollerien kanssa, jotka tuottavat datan, joka moduulilla lähetetään. (30.) Sillä on monia vaihtelevia käyttösovelluskohteita, esimerkiksi valojen etähallinta, pitkän matkan RFID sekä langattomat hälytys- ja turvajärjestelmät (32). Laite operoi 433 MHz taajuudella.

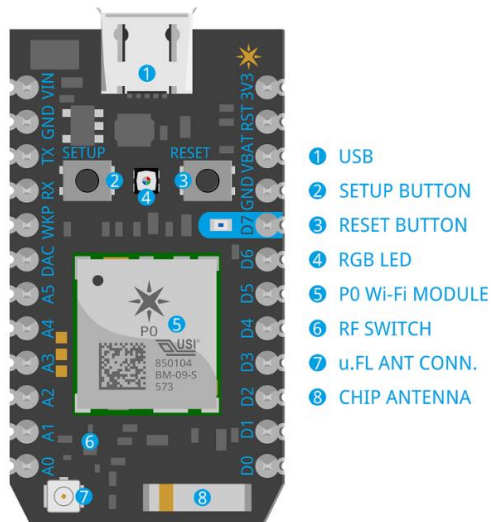
Lähetin on mitoiltaan hyvin pieni ja sen käyttöjännite on 3 voltista 12 volttiin. Sillä voidaan lähettää signaalia jopa sataan metriin asti, riippuen käytettävästä antennista, käyttöympäristöstä ja käyttöjännitteestä. (31.) Lähettimen kokonaisvaltaiseen toimintaan kytetään virta-, maa- ja datajohtimet.

Tässä toteutuksessa radiolähettimen ainoa toiminto oli kuljettaa päälle tai pois käsky samalla radiotaajuudella ohjattavalle pistorasialle langattomasti. Niinpä siis radiolähettimen osalta vaikuttavimpina tekijöinä pidettiin hintaa, sekä mikrokontrollerin kanssa vaadittua yhteensopivuutta. Edellä kuvatuista syistä päädyttiin nopeasti 3 – 12 V:n jännitteellä toimivaan 433 MHz:n radiolähettimeen, johon vastaanottimen löytäminen oli helppoa ja nopeaa.

Radiolähetin kytkettiin Photonin kanssa samaan piirilevyyn, jolloin sen sijainti oli keskeinen pistorasioiden ohjaukseen tarvittavan signaalin voimakkuuden kannalta, mutta samalla piilossa, ettei siitä aiheutuisi visuaalista häiriötä.

### 5.1.3 Kehitysalusta

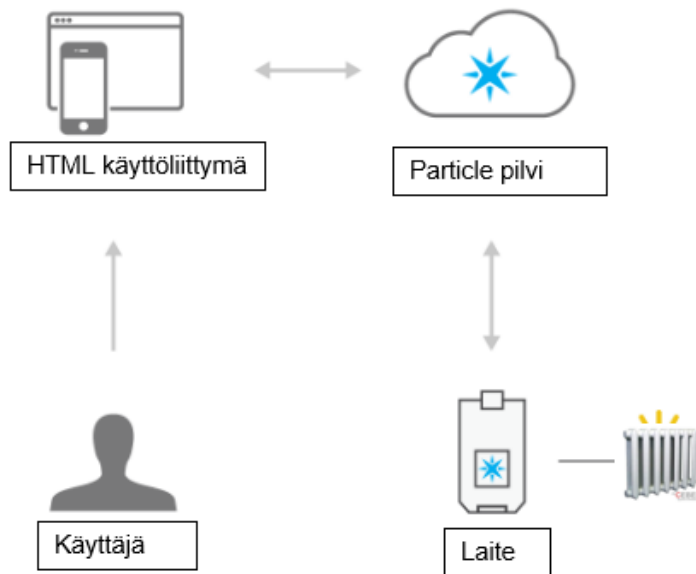
Kehitysalusta on painettu piirikortti, joka sisältää mikroprosessorin sekä minimaaliset tukikomponentit, jotta käyttäjä pystyy tutustumaan prosessoriin ja oppii ohjelmoimaan sitä. Toisin kuin yleisissä käyttöjärjestelmissä, kuten kotitietokoneissa, kehitysalustat sisältävät hyvin vähän, jos ollenkaan ”hardwarea”, joka on omistettu käyttöliittymälle. Kehitysalustoissa on aina mahdollisuus vastaanottaa ja ajaa ohjelma. Tämä voidaan toteuttaa esimerkiksi lataamalla ohjelma sarjaportin kautta jonkintyyppiseen ohjelmoitavaan muistiin. (33.)



Kuva 8. Particle Photon kuvattuna ylhäältä päin sekä pinnien merkinnät (34)

Particlen valmistamassa avoimen lähdekoodin kehitysalustassa Photonissa yhdistyvät ARM Cortex M3-mikrokontrolleri sekä Wi-Fi -siru (kuva 8) (34). ARM Cortex-M3 on yleinen reaaliaikaisissa sovelluksissa käytetty 32-bittinen prosessori ja se sisältää 1MB sisäistä flash -ja 128KB RAM muistia (35). Photon kytetään 3,6 – 5,5 V:n jännitelähteeseen, jolloin siitä saadaan otettua ulos 3,3 V (34). Photonin ja ulkomaailman välinen vuorovaikutus tapahtuu pinnien avulla. Photonin voidaan tuoda input ja output sekä analogisessa että digitaalisessa muodossa. Wi-Fi -moduuli mahdollistaa kommunikoinnin mikrokontrollerin kanssa verkon yli.

Particle Photon nojaa vahvasti omaan pilvipalvelu ratkaisuunsa. Minkä tahansa Particlen tuotteen oikeanlaiseen toimintaan tarvitaan vähintään neljä osapuolta, jotka ovat Particlen pilvi, kehitetty sovellus, ohjattava laite sekä käyttäjä. (36.)



Kuva 9. Kaavio, jossa sovellus kommunikoi suoraan Particlen API:n kanssa (36, muokattu)

Jotta pilven kanssa kommunikointi onnistuu, voidaan käyttää esimerkiksi ”yksinkertaista” tunnistautumista, jolloin sovelluksella ei ole omaa palvelinta tai tausta-arkkitehtuuria vaan verkko tai selain sovelluksella tunnistaudutaan suoraan Particlen API:hin yhteyden ylläpidon sekä laitteen kanssa vuorovaikutuksen saavuttamiseksi. Kuvassa 9 on kaavio yksinkertaisesta tunnistautumisesta korkealla tasolla. (36.)

#### Particle Build verkkokehitysympäristö ja IFTTT integraatio

Particle tarjoaa monipuolisen selaimella toimivan C/C++ -ohjelmointiympäristön (IDE), joka on valmiiksi konfiguroitu, jotta voidaan heti lähteä kirjoittamaan ohjelmia, mutta myös kääntää ja ladata ne kontrollerille samasta paikasta. Ohjelmointiympäristöä voi siis hyödyntää millä tahansa laitteella, jolla on mahdollista käyttää selainta. Ohjelman kääntö ja lataus kontrollerille tapahtuu Particlen pilven kautta, jolloin tiedostot viedään pilveen, jossa ne käännetään ja palautetaan binäärinä (37).

Particle on luonut integraation IFTTT (if this then that) -alustan kanssa, jolla on mahdollista luoda resepti, joka integroituu satojen eri verkkoalustojen kanssa. Näiltä alustoilta etsitään reseptillä asetettuja laukaisevia tekijöitä, joita käytetään reseptin toimintojen toteuttamiseen. Laukaiseva tekijä voi olla melkein mikä tahansa, esimerkiksi toiminto älylaitteella, uusi Facebook-status tai vaikka Photonin pilveen julkaisema tapahtuma. (38.)

**✳ New event published**

This Trigger fires when an interesting event comes from a particular device. Send events using Particle.publish.

**If (Event Name)**

Temperatures

Fill in your published event name; ex: monitoring a washing machine?  
Event Name = Wash\_Status

**is (Event Contents)**

The contents of the published event, "Data"; ex: monitoring a washing machine? Event Contents = Done

**Device Name or ID**

Abe

An optional id for a particular device

---

**📄 Add row to spreadsheet**

This action will add a single row to the bottom of the first worksheet of a spreadsheet you specify. Note: a new spreadsheet is created after 2000 rows.

**Spreadsheet name**

Temperatures

Will create a new spreadsheet if one with this title doesn't exist **Add ingredient**

**Formatted row**

EventContents ||| CreatedAt

Use "|||" to separate cells **Add ingredient**

**Drive folder path (optional)**

Particle

Format: some/folder/path (defaults to "IFTTT") **Add ingredient**

Kuva 10. Esimerkki IFTTT sovelluksen konfiguroinnista

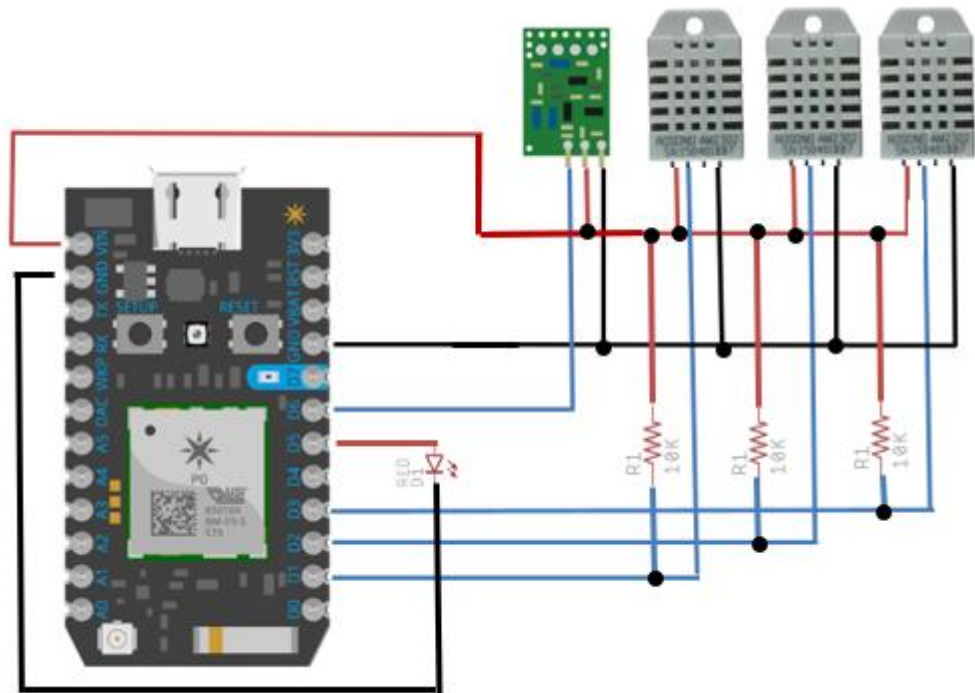
IFTTT -alustalle konfiguroitiin kuvan 10 tavoin resepti, jolla havaitaan Particlen pilveen julkaistu tapahtuma ja sen perusteella toteutettu toiminto. Reseptiin määriteltiin ehto "if" eli kun "Device Name"-kohtaan määritelty laite julkaisee Particlen pilveen "Temperatures"-nimisen tapahtuman toteutetaan toiminto, jolla Googlen laskentataulukoon lisätään uusi rivi, jolle kirjoitetaan "Formatted row"-kohdassa määritelty data ja tapahtuman kellonaika.

## Logiikka ja kytkennät

Photonilla kerättiin kaikki mittalaitteilta tuleva data, ja sitä verrattiin käyttöliittymästä tulevaan kosteuden asetusarvoon. Kosteus- ja lämpöarvot tulevat molemmat samaa datajohdinta käyttäen Photonin digitaaliseen inputtiin. Jotta antureilta tuleva data saatiin ihmisen tulkittavaan muotoon, lisättiin ohjelmaan Adafruitin tarjoama DHT -kirjasto. Kaikille muuttujille, jotka haluttiin viedä pilveen ja sitä kautta käyttöliittymään luotiin Particle.variable tyyppinen muuttuja. Vastaavasti toiminnot, joita käyttöliittymästä haluttiin hallita esimerkiksi manuaaliohjaus, kirjoitettiin muotoon Particle.function.

Radiolähettimen ohjaaminen Photonilla vaatii myös kirjaston. RCSwitch -kirjaston avulla pystytään luomaan olio, jonka avulla voidaan lähettää binääridataa langattomasti radiolähtimellä. Photonilla ajettu ohjelma löytyy kokonaisuudessaan liitteestä 8.

Huoneiston kosteudesta haluttiin myös kerätä historia tietoa. Photon vie kuuden tunnin välein senhetkiset kosteus- ja lämpötilatiedot pilvipalvelu Google Sheetsiin, käyttäen IFTTT -integraatiota. Historia voidaan sen jälkeen hakea käyttöliittymään Sheetsistä. Taulukko, joka halutaan hakea verkon yli, voidaan ensin julkaista, jotta sen sisältämää dataa pystytään tarkastelemaan vapaasti. GSX2JSON on API-palvelu, joka yhdistetään haluttuun taulukkoon tunnuksen perusteella. API-rajapintaa voidaan käyttää ulkopuolisen isännöimänä tai pystyttää se itse. API:n päätepisteeseen voidaan liittää minkä tahansa julkaistun Google Sheetsin "Spreadsheet\_Id" ja "sheet number". Palvelun kautta oman taulukon data voidaan hakea JSON-muotoisena vastauksena lähettämällä JSON pyyntö palvelun päätepisteeseen (s.38, kuva 15).



Kuva 11. Photonin kytkentäkaavio Mäntyharjulla (2, muokattu)

Kuvan 11 kytkentäkaaviossa esitellään Mäntyharjun kiinteistön sähköiset kytkennät lämmityksen ohjausjärjestelmän osalta. Näihin kuuluu :

- kolme suhteellisen kosteuden ja lämpötilan anturia, jotka mittaavat kosteutta ja lämpötilaa eripuolilta rakennusta
- 443 MHz:n radiolähetin, jolla ohjataan pistorasioita, joihin sähköiset lämmityspatterit on kytketty.
- Kuvassa näkyy myös ledi, joka indikoi onko laite yhteydessä verkkoon vai ei.

Punaisella kuvattuna USB-väylään syötetty jännite, sinisellä digitaalisiin inputteihin kytketyt datajohtimet ja mustalla maa.

Työlle asetuista vaatimuksista yksi suurimmista oli järjestelmän etäohjaus mahdollisuus. Kuten edellä on mainittu, Particlen valmistama Photon on pilvipalvelupohjainen mikrokontrolleri, jolloin tiedon siirtäminen pilveen on pyritty tekemään mahdollisimman helppoksi. Etäohjauksen kannalta on tärkeää, että datan saa vietyä helposti jollekin ulkoverkosta käsiksi päästävälle palvelimelle, jolloin halutun datan pystyy hakemaan omalle laitteelle sijainnista riippumatta verkon yli. Lisäksi todettiin, että Particlella on hyödyllisiä integraatioita sekä IFTTT -kanava, jolla pystytään liittymään muihin palveluihin, kuten esimerkiksi Google Sheets'iin, jota myös hyödynnettiin tässä työssä.

#### 5.1.4 Aurinkokenno

Aurinkokenno on laite, jonka avulla pystytään muuttamaan auringon säteily sähköenergiaksi valosähköisen ilmiön avulla. Aurinkokennon toimintaa voidaan kuvata kahdella pääperiaatteella: saapuva fotoni synnyttää absorboivaan puolijohteeseen varauksenkuljettajia, jotka käytetään kulutuslaitteissa tai kerätään akustoon. Yleisimmät aurinkokennon materiaalit ovat yksikiteinen pii, monikiteinen pii, galliumarsenidi sekä amorfinen pii. (39.)

Yksittäisen aurinkokennon tuottama jännite on kohtalaisen pieni, joten usein aurinkokennot kytketään sarjaan ja asennetaan kehikkoon. Tällöin niitä kutsutaan aurinkopaneeleiksi. Valosähköisestä aurinkosähköstä puhuttaessa käytetäänkin yleensä termiä aurinkopaneeli kennon sijasta. (39.)

Tyypillinen käyttöikä aurinkokennolle on noin 30 vuotta. Ne voidaan jakaa tasapaksuihin levykennoihin, joissa auringonvalo tuottaa sähköä suoraan ja keskittäviin aurinkokennoihin, joissa auringonvalo keskitetään tai tiivistetään sähkön tuottamista varten. (39.)

Aurinkokennojen tehokkuus on riippuvainen auringon säteilyenergian määrästä. Tähän vaikuttavia tekijöitä ovat alueen leveysaste, paikallinen ilmasto ja ilman saasteiden määrä. Parhaimmillaan aurinkoenergiajärjestelmät ovat maissa, joissa on paljon aurinkoa ja jotka sijaitsevat päiväntasaajan molemmin puolin. (39.)

Aurinkopaneelit valittiin vastaamaan työhön varattua investointimahdollisuutta. Järjestelmää on suunniteltu laajennettavan ja sitä kautta paneeleita lisättävän, jos testausvaiheen jälkeen paneeleilta saatava energia vastaa tai ylittää niiltä odotetut vaatimukset. Paneeleita valittaessa on tärkeää ottaa huomioon käytettävän aurinkoinvertterin koko ja käyttöjännite sekä suunnitellun kaapeloinnin virrankestokyky. Näitä asioita silmällä pitäen aurinkopaneelit on valittu invertterin kanssa yhteensopiviksi.

#### 5.1.5 Akusto

Akut määritellään sähkökemiallisiksi energiavarastoiksi ja ne koostuvat kahden elektrodin, katodin ja anodin yhdistelmästä. Yleensä elektrolyytti on joko geelimäistä tai neste-mäistä ja se sijaitsee elektrodien välillä. (40.)

Akkujen luokittelussa käytetään usein elektrodien materiaalia. Haptoakussa elektrodit ovat lyijyä ja lipeäakussa esimerkiksi nikkeliä, rautaa tai metallihybridia.

Jos akku kytketään virtapiiriin katodilla tapahtuu pelkistymisreaktio, ja anodilla hapettumisreaktio. Hapettuessa luovutetaan elektroneja, jotka siirtyvät anodin kautta virtapiiriin, ja sen jälkeen katodille synnyttäen sähkövirran. (40.)

Akusto on tasavirtalähde, joka syntyy, kun useita akkuja kytketään sarjaan tai rinnan käyttötarpeesta riippuen.

Ratkaisussa käytetty akusto koostuu kahdesta 100 Ah:n kapasiteetin omaavasta sarjaan kytketystä 12 V:n lyijyakuista. Invertteriin asetetuissa parametreissa määritellään invertteri syöttämään pääasiallisesti aurinkoenergiaa kuormaan sen akustoon varastoimisen sijasta ja näin saadaan vähennettyä akuston kapasiteetin tarvetta. Ajoittain suurempi akuston kokonaiskapasiteetti voisi olla hyödyllinen ja sitä on myös tarkoitus laajentaa, jos testauksissa havaitaan investoinnin kannattavuus.

### 5.1.6 Invertteri

Invertteri eli vaihtosuuntaaja on tehoelektroninen laite tai kytkentä, joka muuttaa tasavirran vaihtovirraksi. Sisään tuotava jännite, ulos lähtevä jännite ja taajuus sekä kokonaisvaltainen tehon käsittelykyky riippuvat kyseessä olevan laitteen tai kytkennän suunnittelusta. Invertteri ei itsessään tuota sähköä, vaan sähkö saadaan tasajännitelähteestä. (41.) Joillakin inverttereillä on hyvä käynnistää moottoreita ja toisilla sysäysvirta saattaa olla hyvin rajoitettu. Suunnittelijan on hyvä tarkentaa invertterin koko ja tyyppi käyttötarkoituksesta riippuen. (42.) Sen lisäksi että vaihtosuuntaajalla muutetaan tasasähkö vaihtosähköksi, sen on myös tärkeää tuottaa hyvälaatuista siniaaltoista verkkovirtaa mahdollisimman pienin häviöin.

Aurinkovaihtosuuntaaja on tasapainottava komponentti aurinkosähköjärjestelmässä ja sitä voidaan käyttää joko kytkettynä verkkoon tai irti verkosta. Vaihtosuuntaajalta ulos saatava teho voidaan siis syöttää valtakunnanverkkoon tai käyttää paikallisesti omaan tarpeeseen.

Aurinkosähköinvertterit voidaan jakaa kolmeen laajaan tyyppiin:

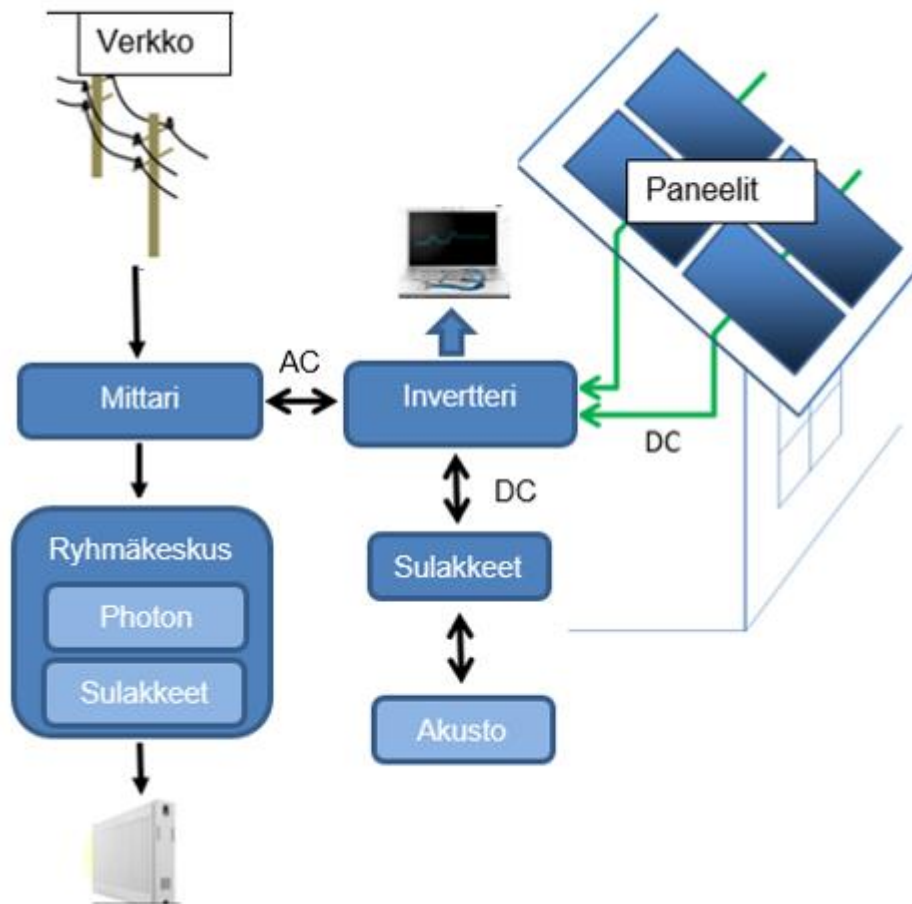
Irrallisia eli verkosta irti kytkettyjä inverttereitä käytetään eristetyissä järjestelmissä, jolloin invertteri ottaa energiansa akustosta, jota ladataan aurinkopaneeleilla integroidun akkuvaraajan kautta (43).

Verkkoon sidotut invertterit ovat koko ajan samassa vaiheessa verkon siniaallon kanssa. Ne on suunniteltu sammumaan automaattisesti turvallisuus syistä, jos jännitelähde tippuu pois. Ne eivät myöskään tuota varavirtaa sähkökatkosten aikana. (43.)

Akustovaraajainvertterit on erityisesti suunniteltu ottamaan virtaa akustosta, pitämään yllä akuston varausta mukana sisään rakennetulla laturilla ja syöttämällä ylimääräisen energian verkkoon. Tämäntyyppiset invertterit pystyvät syöttämään kuormaan virtaa myös sähkökatkosten aikana. (43.)

Älykkäät hybridi-invertterit käsittelevät aurinkokennoja sekä akuston varausta ja verkkoa, jotka on kaikki kytketty suoraan yksikköön. Nämä modernit ”kaikki samassa” järjestelmät

ovat monipuolisia ja niitä voidaan käyttää verkkosidottujen-, irrallisten- tai tuki-invertteerien tavoin, mutta niiden päätoiminto on itsekulutus akuston avulla. (40.)



Kuva 12. Yksinkertaistettu kaavio invertteristä osana Ulvilan aurinkoenergiajärjestelmää

Kuvassa 12 on yksinkertaistettu kaavio invertterin toiminnasta osana Ulvilan aurinkopaneelijärjestelmää. Kun keskukselta (kuva 13) järjestelmän käyttämäksi ohjaustavaksi on asetettu aurinkosähkö, keskuksen sisällä oleva Photon ei vaikuta lämmityksen ohjaukseen, vaan kaikki kuormaan syötettävä teho siirtyy invertterin läpi. Paneeleilta tuleva tasasähkö syötetään invertterin tasasähköpiiriin, jolloin invertteri siirtää asetettujen parametrien mukaan saatavan energian joko kuormaan tai akustoon sulakkeiden kautta. Hybridi-invertteri kytketään myös verkkoon, jolloin jos esimerkiksi akusto on käytetty tyhjäksi eikä paneeleilta saada energiaa, mutta halutaan silti lämmittää, alkaa invertteri

syöttää kuormaan sähköä verkosta. Ratkaisussa ei ollut tarkoitus syöttää paneeleilla tuotettua energiaa verkkoon päin, vaikka se mahdollista olisikin. Kuvaan on myös lisätty invertterin mittaamia arvoja etäluettavaksi julkaiseva palvelin, joka käyttää RS-232 -sarjaväylää invertterin kanssa kommunikointiin.

Aurinkoenergiajärjestelmän pääasiallisena tarkoituksena ei ollut olla kiinteistön päälämmitystapa vaan ympäristöystävällinen lämmityslisä. Tämän lisäksi aurinkoenergiajärjestelmään kuluvat investointikulut haluttiin pitää kohtuullisina, mutta kuitenkin varata hieman tilaa myös laajennukselle, odotuksia positiivisemmän lopputuloksen ilmetessä. Näistä syistä työhön valikoitui Mpp Solarin valmistama Pip-Hsp 24v-sarjan 2424HSP nimellisteholtaan 2,4 kilowattinen hybridi-invertteri. Kyseisiä inverttereitä on myös mahdollista kytkeä useampi rinnakkain tarpeen vaatiessa.

#### 5.1.7 Ryhmäkeskus

Ryhmäkeskusta kutsutaan kansankielessä monesti myös sulakekaapiksi tai -tauluksi. Se on sähköpääkeskuksen alakeskus, josta jaetaan sähköä tietylle alueelle tai ryhmälle. Omakotitaloissa ryhmäkeskuksia on monesti vain yksi, kun taas kerrostaloissa ryhmäkeskus on jokaisessa asunnossa ja isommissa toimistoissa voi olla jopa useampi keskus jokaisessa kerroksessa. Tällä helpotetaan järjestelmän hallintaa ja vähennetään kaapelointi tarvetta verrattuna tilanteeseen, jossa kaikki sulakkeet ja muut keskuskomponentit sijaitsisivat samassa paikassa. (44.)



Ryhmäkeskus kytkettiin pannuhuoneeseen öljylämmityskattilan viereen, jolloin keskus voitiin helposti kytkeä sähkönsyötön ja sähköisen lämmitysvastuksen väliin. Tämä mahdollisti ohjauksen uudella tavalla toteuttamisen sekä kuormaan invertteriltä tulevan energian liittämisen. Tämän lisäksi laitteiden kaapeloiminen oli helppoa ja nopeaa, sillä uudet vedot olivat lyhyitä ja pystyttiin käyttämään paljon jo vedettyjä kaapeleita. Ohjaukseen käytettävät painonapit kytkettiin liitteessä 1 näkyvään ohjauskaappiin komponenttien sijoittelun ja käytettävyyden helpottamiseksi.

Kuvassa 13 esitellään Ulvilaan asennetun ryhmäkeskuksen ohjausvirtapiirin kytkentä AutoCad Electricalilla piirrettynä.

Kytkenän tarkoituksena oli vaihtaa kattilan lämmitykseen käytettävää lämmitysmuotoa öljyn ja verkkosähkön välillä automaattisesti sähkön hinnan perusteella. Lisäksi kytkennällä mahdollistettiin aurinkopaneeleilta tulevan energian hyödyntäminen osana lämmitysjärjestelmää, kaaviossa 5 (s.18) kuvatuin tavoin. Kytkennässä käytettiin apuna kahta öljykattilaan valmiiksi asennettua termostaattia, jotka mittaavat lämmitysjärjestelmässä kiertävän veden lämpötilaa. Sähkövastuksen termostaatti, joka näkyy päävirtapiiri kuvissa liitteissä 2 ja 3 säädettiin lämmittämään hieman öljypolttimen termostaattia aikaisemmin. Sähkön hintaa tarkkailtiin ja sen perusteella kärkeä K6 ohjattiin Photonilla. Näin ollen, kun sähkön hinta on korkea, eikä sähköllä sen vuoksi lämmitetä, kytketty öljypoltin päälle. Huomioitavaa on, että verkkosähköllä lämmitettäessä vastukseen kytketään 3 vaihetta, jolloin vastuksesta saatava teho on huomattavasti yhtä vaihetta suurempi.

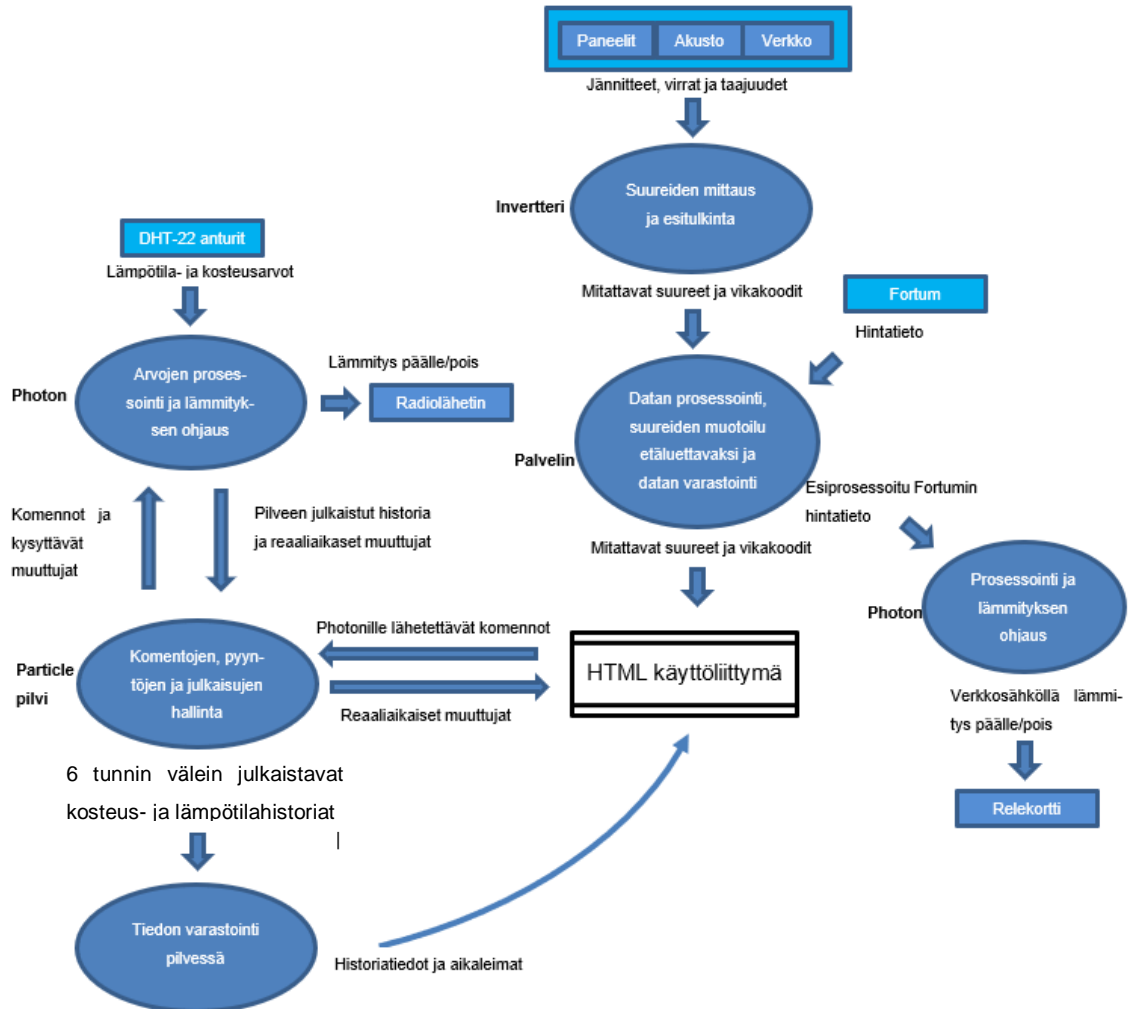
Pääkytkimen ja suojalaitteen jälkeen kuvassa vasemmalla on lämmityskattilan suhteen ristiin kytkentä, joka estää kattilan lämmityksen samanaikaisesti verkko- ja aurinkosähköllä. Painonapilla S1 voidaan ohjata kattila lämpenemään aurinkoenergialla. S2 puolestaan ohjaa apureleen K7 vetämään, joka kehittää itselleen pitopiirin ja kytkee verkkosähkölämmityksen ns. "automaattitilaan", jolloin Photonin hinnanseuranta vastaa lämmitysreleen K1 päälle ohjaamisesta. Jos resettiä R1 painetaan kattilan lämmitys kytketty pois päältä.

Kiukaan ohjauksessa käytetään apuna vaihtokytkintä. Kun vaihtokytkin on 0-asennossa kiuas on pois päältä. Kun vaihtokytkin on asennossa 1, aikareleen ajastin laskee asetetun ajan, jona kiuasta lämmitetään aurinkoenergialla. Kun aikareleen aika on kulunut, ohitetaan vaihtokytkimen asento 2 ja kiuasta lämmitetään verkkosähköllä. Asennossa 2 kiuas kytketään lämpenemään heti verkkosähköllä. Releiden edessä on myös ristiinlukitukset aurinko- ja verkkosähkön samanaikaisen päälläolon välttämiseksi.

## 5.2 Ohjelmisto

Ohjelmiston tarkoituksena on kerätä paikallisilta mittalaitteilta mielenkiintoinen data talteen ja viedä se etäluettavaan muotoon. Samaan aikaan ohjelmistolla monitoroidaan samaa sovellusta (käyttöliittymää) käyttäen sekä Mäntyharjun että Ulvilan kiinteistöjen tapahtumia, mutta myös ohjataan Mäntyharjun kuivanapitolämmitysjärjestelmää etänä. Ohjaus perustuu Mäntyharjun huoneistosta kerättävään anturidataan, jota käytetään rakennuksen lämmitystarpeen tarkkailuun.

Kuvasta 14 voidaan nähdä, mistä järjestelmä hakee dataa, mitä sillä tehdään ja mihin se päätyy. Järjestelmään saapuu dataa kolmesta eri lähteestä: lämpötila- ja kosteusmittauksesta, jännite ja virtamittauksesta sekä Fortumin palvelimelta ja ne on merkattu kuvaan vaalealla pohjalla.



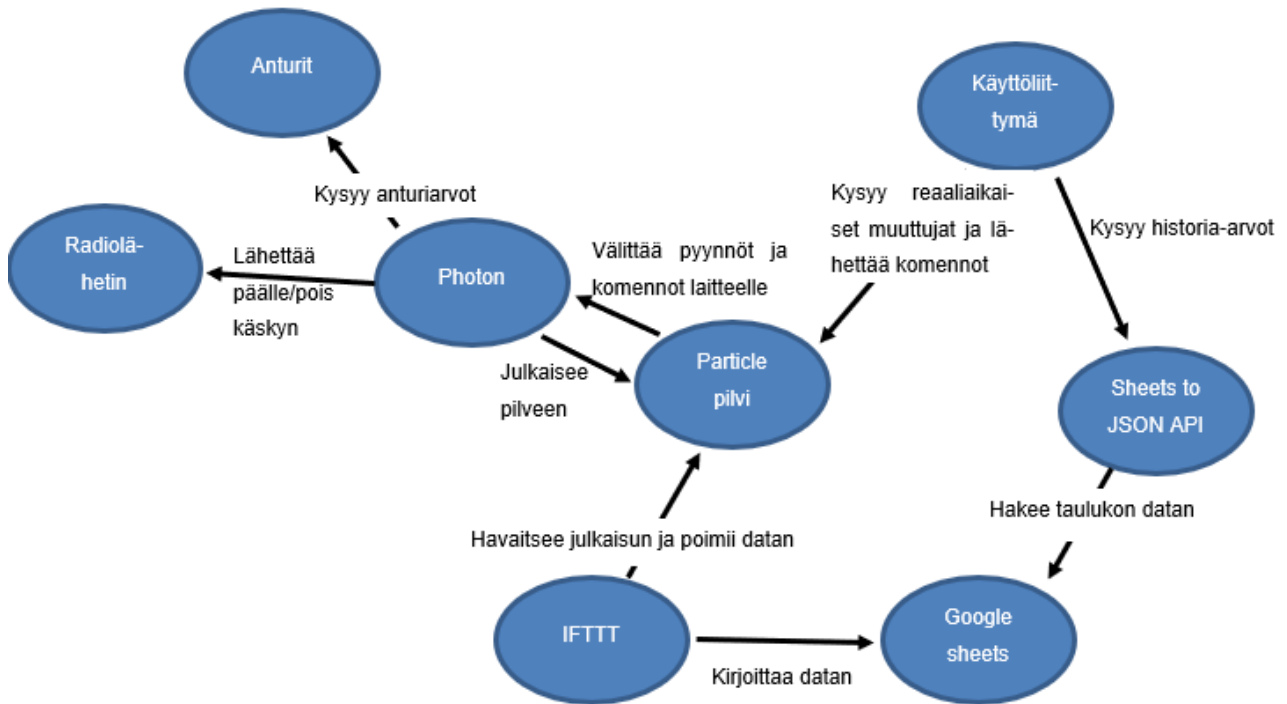
Kuva 14. Järjestelmän yleiskuvas

Järjestelmä prosessoi, lähettää, vastaanottaa ja toteuttaa säätöjä saapuvan datan perusteella. Kuvassa vasemmassa yläreunassa lämpötila- ja kosteusarvot prosessoidaan ja lämmitystä ohjataan radiolähtetimen välityksellä Mäntyharjulle asennetun Photonin toimesta. Photon on yhdistetty paikalliseen lähiverkkoon, jonka välityksellä se on yhteydessä Particlen pilveen. Pilven toimintoja on havainnollistettu kuvassa käyttöliittymän ja Photonin välissä. Kaikki Photonin kanssa tapahtuva kommunikointi tehdään Particlen

pilven kautta, jossa komennot ja pyynnöt käsitellään ja jonne julkaisut viedään. Kuitenkin reaaliaikaisten ja historiatyyppisten tietojen käsittely toteutetaan eri tavoilla. Komennot lähetetään laitteelle ja reaaliaikaiset muuttujat haetaan laitteelta suoraan Particlen API:n välityksellä. Historiatiedot viedään Particlen pilvestä Googlen pilvipalveluun varastoitavaksi, josta ne haetaan eri API:n kautta kuvassa 15 kuvattuun tyyliin.

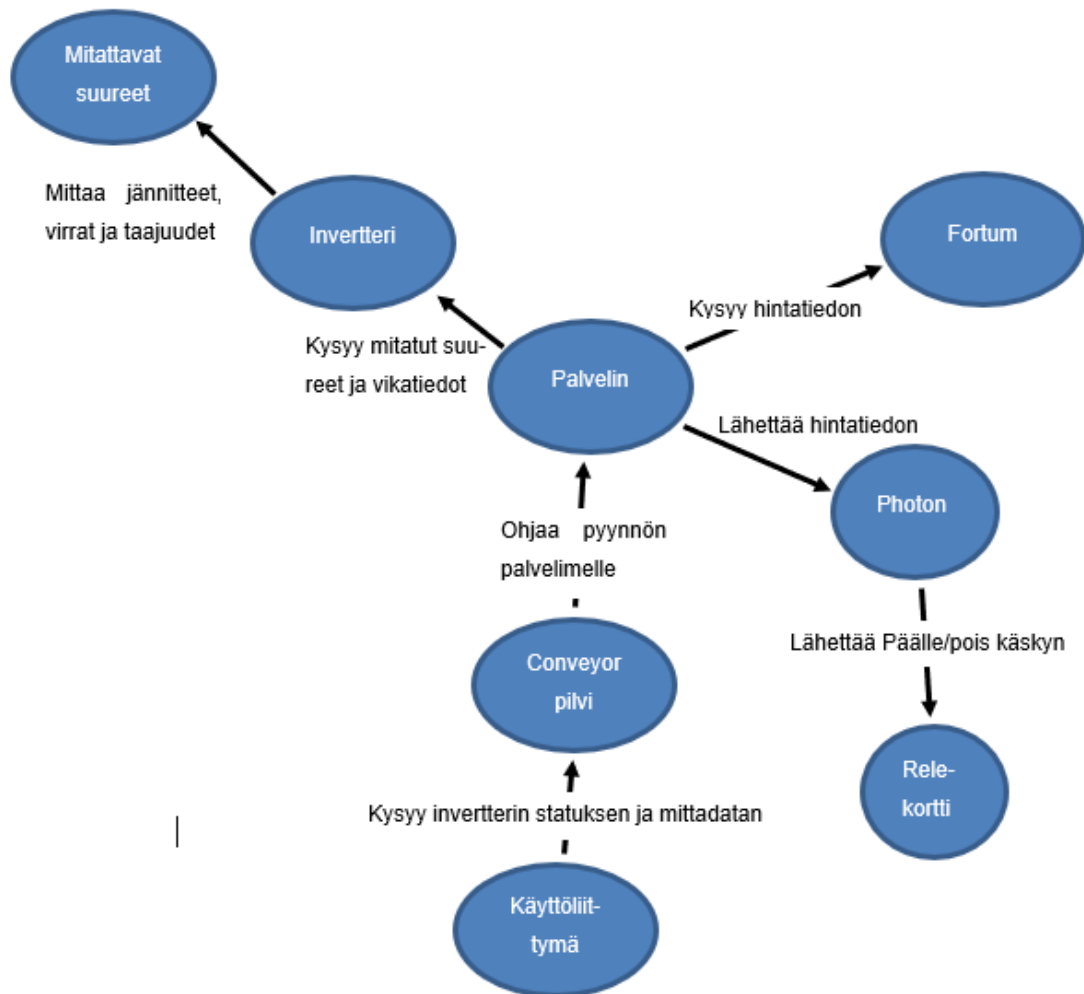
Kuvan oikeassa yläreunassa on havainnollistettu invertterin toimintoja. Invertteri mittaa jänniteitä, virtoja ja taajuuksia, tulkitsee niitä ja generoi vikakoodin havaitessaan ongelman. Invertteriltä data kysellään eteenpäin palvelimelle, jota havainnollistetaan kuvassa Invertterin ja käyttöliittymän välissä. Data prosessoidaan, jonka jälkeen se tallennetaan palvelimen tietokantaan. Prosessoidut mittadatat ja vikakoodit haetaan käyttöliittymään kuvassa 16 kuvatuin tavoin.

Palvelimelle haetaan samanaikaisesti rinnakkaissovelluksella Fortumin hintatieto, joka välitetään eteenpäin Ulvilan lämmitystä ohjaavalle Photonille. Photonin tekemän ohjauksen vaikutus voidaan havaita kuvasta 13, jossa sen ohjaama rele on K6. Kuvasta nähdään, että Photon ohjaa vain verkkosähköllä toimivaa lämmitystä, joka on erillään aurinkosähkön syötöstä. Photon laskee halvimman tavan lämmittää ja ohjaa sen perustella joko lämmityksen päälle verkkosähköllä tai sähkön hinnan ollessa korkealla öljypolttimella termostaatin avulla. Ulvilan koko lämmitysjärjestelmän ohjausta ollaan havainnollistettu kaaviolla 5 sivulla 18.



Kuva 15. Mäntyharjun komponenttien välinen integraatio

Kuvassa 15 esitellään Mäntyharjun lämmityksen ohjaukseen käytettävien komponenttien välinen integraatiomalli. Kuvassa nuolen suunta kuvaa laitteiden välistä kontrollivaltaa. Photon kysyy antureilta lämpötila ja kosteusarvot sekä ohjaa näiden perusteella radiolähtetimen kautta lämmityspattereita. Photon julkaisee kuuden tunnin välein sen hetkiset lämpötila- ja kosteusarvot historiatiedoiksi Particlen pilveen. IFTTT -sovellus havaitsee julkaisun ja reagoi siihen siirtämällä julkaisun sisältämän datan Google Sheetsiin varastoitavaksi. Käyttöliittymästä tehdään tietyin väliajoin kyselyitä Particlen pilveen tämän hetkisistä eli usein päivittyvistä muuttujista. Pilven API ohjaa pyynnöt laitteelle, joka palauttaa muuttujan arvon JSON-muotoisena. Käyttöliittymästä voidaan lähettää samalla periaatteella myös komentoja laitteelle, jolloin laite suorittaa kyseisen toiminnon ja palauttaa paluuarvon. Käyttöliittymästä lähetään myös pyyntö hakea historia dataa Sheetsistä. Pyyntö lähetään API:n kautta, joka hakee taulukon datan, muotoilee sen JSON-muotoiseksi ja palauttaa arvot.



Kuva 16. Ulvilan komponenttien välinen integraatio

Kuvasta 16 nähdään Ulvilan komponenttien välinen integraatio. Edellisen kuvan tavoin mallilla havainnollistetaan kunkin komponentin kontrollia toisiinsa nähden nuolen avulla. Keskeisessä osassa oleva palvelin kyselee sekä invertterin mittaamaa, että sen tuottamaa dataa. Invertterin tuottama data kysytään käyttöliittymästä Conveyor -pilven kautta, joka mahdollistaa palvelimen kanssa verkon yli kommunikoinnin. Samaan aikaan palvelin lähettää Fortumin palvelimelle JSON-muotoisen pyynnön, jolla haetaan sähkön hinnan arvo. Lisäksi palvelin lähettää tietyin väliajoin keräämänsä hintatiedon Photonille, joka vertaa arvoa asetettuun 4 c/kWh rajaan ja tekee päätöksen verkkosähkölämmityksen ohjaukselle.

### 5.2.1 Palvelin

Palvelimen ohjelmointi toteutettiin Microsoftin Visual Studio ohjelmointiympäristössä C# ohjelmointikieltä käyttäen.

Microsoftin kehittämä Visual Studio on ohjelmankehitysympäristö, jossa voidaan käyttää useita eri ohjelmointikieliä kuten, C++ ja C#. Kyseisellä työkalulla voidaan esimerkiksi tehdä web-, Windows-, sekä mobiilisovelluksia ja siihen voidaan integroida useiden valmistajien täydentäviä ominaisuuksia kuten Intel Visual Fortran. (45)

Palvelin pystytettiin käyttämättä jääneelle Windows 7 -käyttöjärjestelmää käyttävälle kannettavalle tietokoneelle. Tämän palvelimen toiminta voidaan jakaa neljään päätoimintoon: aurinkoenergiajärjestelmään liittyvän datan kyseleminen invertteriltä, datan jäsentely ja varastointi, datan julkaisu sekä hinnan seurantasovelluksen ajaminen.

Invertteriltä datan hakeminen RS-232 -sarjaväylän kautta

RS-232 on 1960-luvulla kehitetty standardi kahden laitteen välistä kommunikointia ja tiedonsiirtoa varten. Standardin mukainen siirrettävä tavu koostuu aloitusbitistä, joka on aina 0, databiteistä, joita on 5-9, pariteetti bitistä ja yhdestä tai puolestatoista loppubitistä. Tietoa vastaanottaa ja lähettää siihen tarkoitettu UART-piiri, joka huolehtii lukemisen ja kirjoittamisen ajoittamisesta. Standardin mukaisesti tietoa on mahdollista siirtää eri baudinopeuksilla, joista yleisimpiä ovat 1200, 2400, 4800, 9600 ja 19200 bittiä sekunnissa. Baudinopeus tarkoittaa sekunnissa siirrettyjen bittien lukumäärää. (46.)

Käytettävältä invertteriltä ei ollut varsinaisesti tarkoitettu hakea dataa muuten kuin sen mukana tulevaan PC-ohjelmistoon. PC-ohjelmistosta invertterin tietoja oltaisiin kuitenkin voitu lukea ainoastaan paikallisesti, joka ei riittänyt työlle asetettuihin vaatimuksiin. Laitteen kanssa muun kuin valmistajan ohjelmiston avulla käytävästä kommunikaatiosta ei asiakkaille ole saatavilla minkäänlaista virallista dokumentaatiota. Viitattuja invertterin ja pystytetyn palvelimen välisiä komentoja ei voida vahvistaa virallisista lähteistä vaan ne on joko löydetty keskustelupalstoilta tai päätelty itse. Työssä invertterille lähetetyissä komennoissa on kuitenkin käytetty pohjana kommunikaatioprotokollamanuaalia (47).

Ongelmaa lähdettiin purkamaan valmistajan toimittaman ohjelmiston avulla. Invertteriltä tulevaan RS-232-sarjaväylään kytkettiin USB-sarjaporttisovitin ja Windowsiin asennettiin tätä varten ajuri. Valmistajan ohjelmiston ja invertterin välille pystytettiin kommunikaatioyhteys, jonka avulla invertteriltä tulevaa dataa pystyttiin havainnollistamaan.

Vaihtoehtoja pohdittiin vastaavan kommunikaatioyhteyden saavuttamiseksi Visual Studiassa ajettavan ohjelman kanssa, jotta data saataisiin osaksi käyttöliittymää.

Toteutuksen PIP-4048 (MPPSolar) Invertteri tulkitsee 16-bittistä CRC-algoritmiä, jota käyttäen sille voidaan lähettää komentoja manuaalissa mainitussa muodossa. Tämä tarkoittaa käytännössä sitä, että mikä tahansa manuaalissa mainittu String voidaan muuttaa hexadesimaaliksi ja sen perään lisätä saman komennon CRC-arvo sekä paluu arvo hexana, jolloin saadaan invertteri ymmärtämään komento ja myös vastaamaan siihen ihmisen ymmärtämässä muodossa. CRC-arvo voidaan laskea esimerkiksi netistä löytyvällä laskurilla.

CRC-algoritmiä onnistuttiin tulkitsemaan ja sen avulla toteuttamaan komentoja terminaali emulaattoreilla kuten Tera Termillä. Varsinaisen palvelimen pystyttäminen Visual Studiassa alkoi sarjaväylään kirjoittamisella ja sieltä lukemisella, joka toteutettiin Microsoftin "SerialPort Class"-esimerkin pohjalta.

Ohjelman alussa luodaan sarjaportti olio, joka määrittellään kuuntelemaan invertteriin kytkettyä porttia baud-nopeudella 2400. Muut erikseen määrittelemättömät sarjaportin arvot kuten "parity" tai "Data bits" määräytyvät oletusarvojen mukaan.

```
_serialPort.Write(new byte[] { 0x51, 0x50, 0x49, 0x47, 0x53, 0xb7, 0xa9, 0x0d }, 0, 8); //QPIGS
```

```
string message = _serialPort.ReadExisting();
string[] splittedMessage = message.Split(' ');
//To get the whole reading
while (splittedMessage.Length < 18)
{
    message += _serialPort.ReadExisting();
    splittedMessage = message.Split(' ');
    Thread.Sleep(200);
}
```

Esimerkkikoodi 1. Sarjaporttioliolla kirjoittaminen ja sillä sarjaportin lukeminen

Portin aukaisemisen jälkeen käynnistetään silmukka, joka lähettää sarjaporttiin komennon tietyin väliajoin ja sen jälkeen lukee vastausta niin pitkään kunnes oletettu pituus saavutetaan kuten esimerkikoodissa 1. Sarjaportista saatu vastaus sisältää välilyönnin jokaisen arvon välissä, jolloin se voidaan jakaa ja sen perusteella laskea viestin pituus. Vastaus määritellään "message" muuttujan arvoksi.

#### Datan jäsentely ja tallentaminen SQL-tietokantaan

Muuttuja "splittedMessage" on string-tyyppinen taulukko, ja sen sisältämiin arvoihin voidaan viitata yksiulotteisen taulukon tavoin. Kommunikaatiomanuaalia apuna käyttäen koko vastaus voidaan jaotella indeksien avulla haluttuihin ryhmiin ja tallentaa sitten tietokantaan, jolloin tiedon hakeminen helpottuu huomattavasti.

Visual Studiossa on mahdollista luoda paikallinen SQL-tietokanta yksinkertaisesti lisäämällä projektiin "Service-based Database" ja luomalla siihen uusi taulukko, johon lisätään halutut arvot.

```

        str = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFile-
name=C:\Users\source\repos\Application\WebApplication\App_Data\Data-
base1.mdf;Integrated Security=True";
        con = new SqlConnection(str);
        con.Open();
        Console.WriteLine("Database connected");
        string query = "INSERT INTO AlarmsAndFaults (Warnings)
VALUES ('" + QPIWSmessage + "')";
        SqlCommand ins = new SqlCommand(query, con);

        ins.ExecuteNonQuery();

        Console.WriteLine("\n Data stored Into Database");
        con.Close();

```

#### Esimerkkikoodi 2. Datan tallentaminen tietokantaan SQL komennolla

Esimerkkikoodissa 2. nähdään, kuinka tietokannan luonnin jälkeen siihen luodaan uusi yhteys, jonka jälkeen valittu data (tässä tapauksessa "QPIWSmessage") viedään taulukon "AlarmsAndFaults" haluttuun sarakkeeseen "INSERT INTO"-komennon avulla. Konsoliin saadaan tieto datan onnistuneesta tallennuksesta.

## Datan hakeminen SQL-tietokannasta ja sen julkaisu

Palvelimen säilömaa dataa voidaan hakea API-rajapinnan (liite 4) kautta, joka vastaa sen päätepisteeseen lähetettyihin "GET" tyyppisiin http(s) pyyntöihin. Jotta palvelimelle on mahdollista lähettää julkisen verkon puolelta kyselyitä, siihen voidaan luoda ulkoverkosta käsiksi päästävä päätepiste esimerkiksi Visual Studioon asennettavan Conveyor-lisäosan avulla. Luotuun Conveyor-pilven päätepisteeseen voidaan lähettää pyyntöjä samaan tyyliin kuin reitittimen julkiseen IP-osoitteeseen johon lähiverkon portti on jaettu, mutta Conveyor "tunneloi" myös palomuurin ohi.

```
public string GetHistory()
{
    SqlConnection con;
    con = new SqlConnection(str);
    con.Open();
    string q = "SELECT TOP 5 timeStampHis, EnergyHis FROM His-
tory ORDER BY Id DESC";
    SqlCommand view = new SqlCommand(q, con);
    SqlDataReader dr = view.ExecuteReader();
    while (dr.Read())
    {
        theData += dr.GetValue(0).ToString() + " " + dr.Get-
Value(1).ToString() + " ";
    }
    con.Close();
    return theData;
}
```

### Esimerkkikoodi 3. Datan hakeminen tietokannasta SQL-kyselyllä

Kun palvelimen API:n päätepisteeseen lähetetään esimerkiksi esimerkkikoodin 4 tyyppi-  
nen pyyntö se prosessoi käskyn ja ajaa päätepiesteen mukaisen toiminnon läpi. Esimerk-  
kikoodissa 3 tietokannasta haetaan 15 viimeisintä päivittäin laskettua energian kulutus  
arvoa ja päivämäärää SELECT kyselyn avulla.

Palvelimella ajetusta pääohjelmasta on lisätty liitteeseen 5, ohjelma osuus, joka kysyy invertteriltä mittadataa ja tallentaa sen tietokantaan. Liitteen lopussa on ohjelmisto osuus, jolla tietokannasta haetaan tallennetut arvot API-rajapinnan avulla. Palvelimen API-rajapinta löytyy liitteestä 4. Ohjelmiin on tehty muokkauksia, kuten tiedostopolkujen muutoksia työhön liitettäessä.

#### Fortum tarkka hintatiedon hakeminen

Täysin irrallaan Visual Studiossa ajettavasta ohjelmasta on selaimella ajettava Fortumin jälleenmyymää pörssisähkön hintaa seuraava sovellus (liite 7). HTML:llä ja JavaScriptillä toteutettu sovellus lähettää tietyin väliajoin esimerkkikoodin 4 tyyppisen kyselyn Fortumin palvelimelle ja poimii takaisin tulevasta vastauksesta kyseisellä hetkellä vallitsevan sähkönhinnan. Tämän jälkeen sovellus lähettää saman hintatiedon kattilan lämmitystä ohjaavalle Photonille. Sovelluksella pystytään myös asettamaan hinnan raja-arvo, johon Photon vertaa senhetkistä pörssihintaa. Photoniin on ohjelmoitu yksinkertainen silmukka, joka tarkkailee jatkuvasti hintaa ja sille sovelluksella asetettua raja-arvoa ja kytkee releen K6 kärjen (s.33, kuva 13) sen perusteella kiinni tai auki. Lisäksi Photon ajaa joka minuutti läpi lyhyen toiminnon, jolla lasketaan, kuinka pitkään on ollut halvempaa lämmitä sähköllä ja kuinka paljon säästöä siitä kertyy. Photonin ajama ohjelma on kokonaisuudessaan liitteessä 6.

#### 5.2.2 Käyttöliittymä

Käyttöliittymä päädyttiin toteuttamaan käyttäen HTML:llä ja Javascriptiä, koska näin käyttöliittymä voidaan muokata haluttuun muotoon niin visuaalisesti kuin toiminnallises-  
tikin. Lisäksi käyttöliittymästä saadaan äärimmäisen joustava eri laitteiden käyttöä ajatellen, sillä sitä voidaan käyttää millä tahansa laitteella, jolla on mahdollisuus selaimen. Näiden ominaisuuksien vuoksi käyttöliittymästä saadaan oivallinen etähallinnointiin vaikka puhelimella.

## HTML

HTML (Hypertext Markup Language) on avoin standardoitu kuvaus kieli, jolla kuvataan hyperlinkkejä sisältävää tekstiä. Tätä voidaan kutsua myös hypertekstiksi, josta suomen­nettu nimi eli hypertekstin merkintäkieli muodostuu. HTML:llä voidaan merkitä myös tekstin rakenne eli muun muassa mikä osa tekstistä on otsikkoa ja mikä leipätekstiä. HTML on erityisesti suunniteltu kieleksi, jolla kirjoitetaan internetsivuja. (48.)

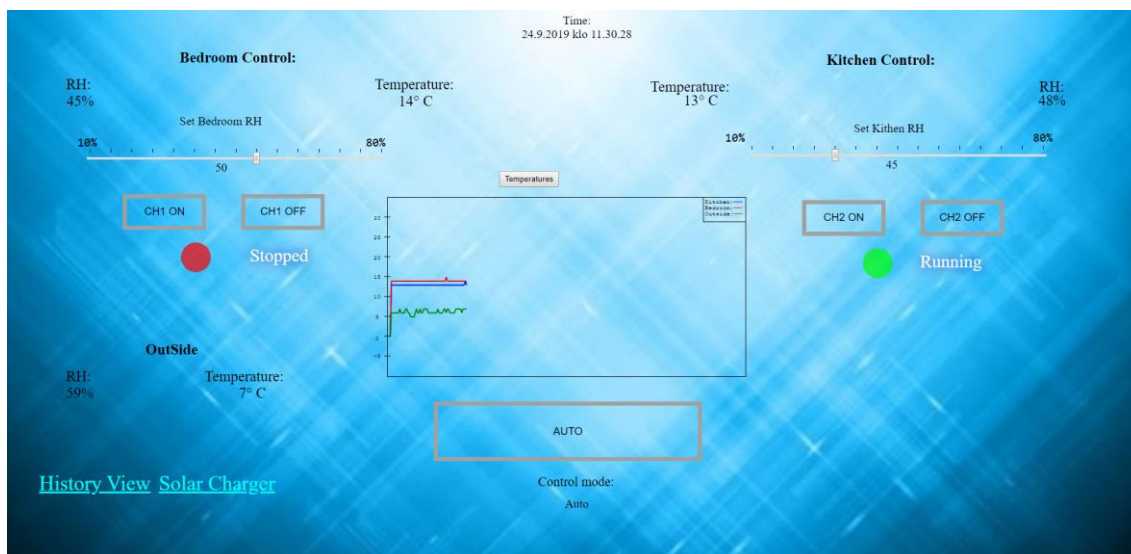
HTML:llä kirjoitettu koodi on rakenteista tekstiä ja koostuu elementeistä, jotka kirjoitetaan sisäkkäin tai peräkkäin. Koodaustasolla elementtejä edustavat kulmasulkeilla merkityt tunnisteet, joihin voidaan lisätä erilaisia attribuutteja, jotka vaikuttavat elementin tyyppiin ja ominaisuuksiin sekä mahdollisesti ulkonäköön. (48.)

```
<tunniste attribuutti="arvo"> elementin sisältö </tunniste>
```

Kuva 17. Tunniste, josta HTML sivu rakentuu (48)

HTML:ää tulkitaan selaimella. Selaimet käyttävät tunnisteita teknisinä ohjeina, joiden mukaan sivun varsinainen sisältö jäsenellään.

HTML:ää käytettiin käyttöliittymässä elementtien muodostamiseen ja niiden sommittele­miseen, jotta Javascriptillä haettu tai sillä lähetetty data voitiin visualisoida. HTML:llä to­teutettiin myös käyttöliittymän taustan visualisointi ja sivua selkeyttävä otsikointi kuten kuvissa 18, 19 ja 20 alla.

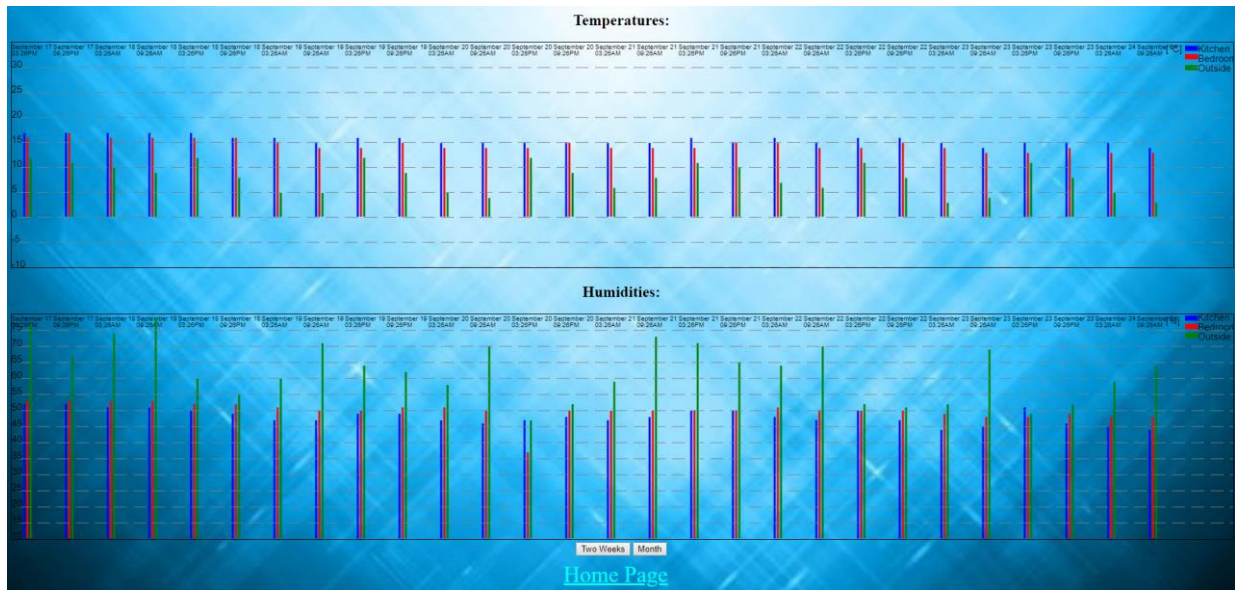


Kuva 18. Etähallinta PC-käyttöliittymän päänäkömä

Kuvassa 18 esitellään Mäntyharjun kiinteistön makuuhuoneen, keittiön ja ulkoterassin lämpötila- sekä kosteusseuranta ja hallinta, PC-valvomotyypisenä ratkaisuna toteutettuna.

Käyttöliittymän päänäkömästä voidaan lukea reaaliaikaisia kosteus- ja lämpötila-arvoja makuuhuoneesta, keittiöstä ja ulkoa. Lisäksi näkömästä voidaan ohjata keittiöön ja makuuhuoneeseen kytkettyjä sähköisiä lämmityspattereita joko manuaalisesti tai automaattisesti. Kummankin huoneen lämmittimiä voidaan kontrolloida päälle tai pois erikseen tai vaihtoehtoisesti järjestelmä voidaan asettaa automaattille historiaikkunan alapuolelta. Automaatilla säätö tapahtuu kaavion 3 (s.16) kuvailemalla tavalla, johon asetettu kosteuden raja-arvo määräytyy kummallakin huoneella erikseen liukusäätimellä asetetun arvon perusteella. Tämän hetkinen asetettu kosteuden raja-arvo ilmoitetaan liukusäätimen alapuolella. Säätömuodon nykyinen tila nähdään käyttöliittymän keskellä lähellä alareunaa.

Lämmityksen tämänhetkinen tila (päällä tai pois) nähdään tilatiedosta "Running" tai "Stopped". Lyhyttä historiaa lämpötiloista piiryy keskellä olevaan kuvaajaan ja pidemmän aikavälin historia voidaan valita "History View"-linkistä.



Kuva 19. PC-käyttöliittymän historianäkymä

Kuva 19 on PC-käyttöliittymän historia näkymästä. Tarkoituksena on havainnollistaa järjestelmän toimivuutta ja korjata puutoksia tai havaittuja epäkohtia paikan päältä tallennetun historian avulla. Näkymästä voidaan tulkita aiempaa lämpötilan ja kosteuden kehitystä. Eri huoneisiin sijoitetut anturit on merkattu eri väreillä. Keittiön kosteudet ja lämpötilat havaitaan sinisellä piirretyllä pylväällä. Makuuhuoneen vastaavat arvot on merkitty punaisella ja ulkotilan vihreällä.

Historianäkymän näytteitä otetaan kuuden tunnin välein. Alareunassa näkyvillä painonapeilla voidaan valita historiatiedot joko kahden viikon tai kuukauden ajalta, sen ollessa oletuksena yhden viikon tarkkuudella. "Home Page"-linkistä voidaan siirtyä takaisin päänäkymään (kuva 18).



Kuva 20. PC-käyttöliittymän aurinkopaneelijärjestelmän näkymä

Kuvassa 20 esitellään Ulvilan kiinteistöön liittyvä käyttöliittymäosuus. Näkymä on toteutettu, jotta pystytään seuraamaan järjestelmän tuottamaa energiamäärää ja siinä esiintyviä vikoja. Näiden tietojen perusteella voidaan tehdä kehityssuunnitelmia tehostamaan hyötyvaikutuksia, mutta pystytään myös havaitsemaan järjestelmään ilmaantuvia vikoja huomattavasti nopeammin pelkkään paikan päällä olevaan paneeliin verrattuna. Vasemmassa yläreunassa nähdään verkon jännite ja taajuus, johon invertteri on kytketty sekä invertterin kuormaan syöttämät vastaavat arvot ja prosentuaalinen osuus, jonka kuorma ottaa invertterin maksimi tehontuotosta. Kyseisten arvojen alla akustolta mitattu jännite sekä sinne syötettävä latausvirta. Kuvassa näkyvässä tilanteessa, jossa aurinkopaneeleita saadaan virtaa, mutta akustoon sitä ei mene, voidaan päätellä, että energia syötetään suoraan kuormaan.

Historiaikkunaan mittausarvojen alle haetaan 15 päivän ajalta paneeleilla tuotettu energiamäärä kultakin päivältä erikseen. Keskelle lasketaan kumulatiivisesti tuotettu energiamäärä järjestelmän käyttöönotto päivästä lähtien, joka auttaa ymmärtämään paneeleilta tosiasiaa tulevaa energiamäärää ja sitä kautta helpottaa mahdollisten tulevien investointien tekemistä.

Oikeaan reunaan saadaan reaaliaikainen tieto invertterin tekemistä vika-, hälytys-, tai varoitusilmoituksista. Näistä edellä kuvatuista mittauksista sekä laitteelta kysyttävistä vikakoodeista voidaan päätellä, vastaako invertterin toiminta oletettua vai tarvitaanko kenties huoltoa. Käyttöliittymän lähdekoodia voidaan tarkastella liitteessä 9, jossa näkyvät kaikkien kolmen sivun elementit ja osa muotoiluista.

## JavaScript

Netscapen alun perin kehittämä JavaScript on pääasiassa Web-ympäristössä käytettävä dynaaminen komentosarjakieli. Tärkein sovellus JavaScriptille on mahdollisuus lisätä Web-sivuille dynaamista toiminallisuutta. JavaScriptiä käytetään tavallisesti osana verkkoselaimia, joiden toteutukset sallivat asiakaspuolen sovellusten interaktion käyttäjän kanssa, selaimen rajoitetun hallinnan ja käyttäjälle näytettävän dokumentti sisällön muokkaamisen. JavaScriptiä käytetään myös muun muassa palvelinten verkko-ohjelmoinnissa sekä työpöytä ja mobiilisovellusten luomisessa. JavaScript-tiedostojen tiedosto liite on yleensä .js. (49.)

Javascriptillä toteutettiin kaikki käyttöliittymän toiminnallisuudet muun muassa lämpötila- ja kosteusarvojen sekä jännite- ja taajuusarvojen haku. Näiden lisäksi käyttöliittymässä on kolme eri näkymää ja Javascriptillä ohjataan mitä elementtejä kullakin sivulla näytetään. Javascriptillä myös pilkottiin käyttöliittymästä lähetettävä ja siihen vastaanotettava data sekä muokattiin se oikeanlaiseen muotoon, joka oli useimmiten JSON. Käyttöliittymään piirrettiin myös kuvaajia menneestä ja nykyisestä datasta Javascriptiä apuna käyttäen. Esimerkkikoodissa 4 on esimerkki JavaScriptillä kirjoitetusta ohjelman toiminnosta, jonka tyyppisillä kyselyillä monet muuttujista haettiin toiselta palvelimelta.

```

function GetGridValues() {
    var serviceUrl = 'https://applicatio.conveyor.cloud/api/values/1';
    var method = $('#method').val();
    $.ajax({
        type: method,
        url: serviceUrl
    }).done(function (data) {
        var splittedGrid = data.split(' ');
        var GridDisplay = splittedGrid[0].substr(1); //To
remove the first ( sign
        $('#GridVoltageText').text(GridDisplay);
        $('#GridFrequencyText').text(splittedGrid[1]);
    }).fail(function (jqXHR, textStatus, errorThrown) {
        $('#GridFrequency').text(jqXHR.responseText || textStatus);
    });
}

```

Esimerkkikoodi 4. JavaScript toiminto, jolla haetaan verkon jännite ja taajuus palvelimelta käyttööliittymään

Automaattisesti tietyin väliajoin ajettava scriptti (koodiesimerkki 4) tekee AJAX (Asynchronous JavaScript And XML) -pyynnön määriteltyn osoitteeseen. AJAX ei ole ohjelmointikieli, se vain käyttää selaimen sisään rakennettua XMLHttpRequest -oliota kysyäkseen dataa Web-palvelimelta internetin yli. Palvelimella pyyntö käsitellään, jonka jälkeen luodaan vastaus, joka lähetetään takaisin selaimen (50). Kun vastaus palvelimelta saadaan, käytetään JavaScriptiä datan pilkkomiseen sekä näyttämiseen käyttäjälle. Jos vastausta ei saada tiettyyn aikaan mennessä pyynnön lähettämisestä, käyttäjälle näytetään virhe koodi.

Käyttöliittymän Javascriptillä toteutettu kosteuksien ja lämpötilojen kysely sekä sen perusteella tehty kuvaajan piirto on liitteessä 10. Lisäksi liitteen lopussa esitellään historia arvojen kysely ja vastauksen muotoilu sille varattuun tilaan.

Palvelin on pystytetty julkiseen verkkoon, joten sen yksityisyyttä on haluttu suojella välttämällä palvelimen URL -osoitteen jakamista. Ulkopuolisten palvelimeen yhdistämisestä ei pitäisi koitua varsinaista haittaa, sillä oikeudet on rajoitettu ainoastaan lukuoikeuksiin ulkopuolisille.

## 6 Tulokset

### 6.1 Mäntyharjun kiinteistö

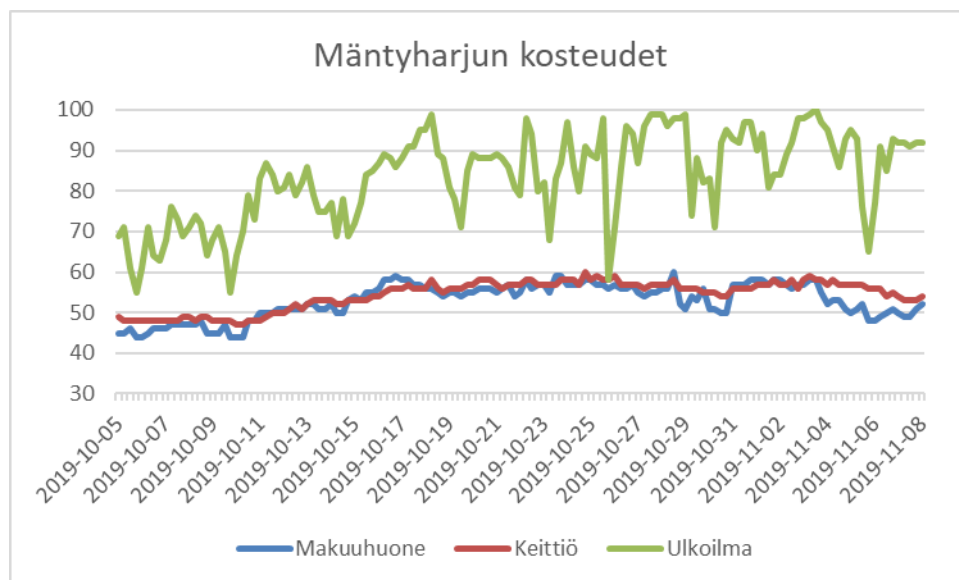
Rakennuksesta kerättiin kosteus-, sisä- ja ulkolämpötila-arvoja etäluettaviksi ajalta 05.10.2019 – 08.11.2019. Kuivanapitojärjestelmän kaikki olennainen data saatiin julkaisua etäluettavaan muotoon ja halutut toiminnot kuten lämmittimien päälle/pois ohjaus toteutettua. Järjestelmä vie tulkintakelpoista dataa historiatiedoista pilveen, josta ne voidaan hakea käyttöliittymään tarkasteltavaksi. Käytettävyyden helpottamiseksi manuaaliohjausta lukuun ottamatta käyttöliittymä osiossa esitellyt toiminnot tuotiin myös mobiililaitteella käytettävään muotoon. Mobiilisovellus toteutettiin täysin PC-käyttöliittymän tavoin ja ainoana eroavaisuutena HTML -ohjelmassa on otsikko, joka skaalaa elementit puhelimen ruudulle sopiviksi.

Rakennukseen oli suunniteltu asennettavan kolme lämpötila- ja kosteusanturia, yksi keittiöön, yksi makuuhuoneeseen ja yksi ulos terassin suojiin. Kolmesta asennettavasta anturista ainoastaan yksi saatiin sijoitettua täysin sille suunnitellulle paikalle. Makuuhuoneeseen suunniteltua anturia ei voitu sijoittaa makuuhuoneeseen, sillä seinän rakenteelliset ominaisuudet eivät soveltuneet siihen suunnitellun reiän poraamiseen. Kyseinen anturi asennettiin sen sijaan seinän toiselle puolelle, jolloin anturin mittaamien arvojen tulkittiin olevan riittävän lähellä makuuhuoneen arvoja, kun ilmavirtauksesta huolehdittiin. Ulkolämpötilaa mittaava anturi kokeiltiin sijoittaa terassin katoksen ja ikkunapuitteiden alle, jolloin se olisi suojassa ulkoisilta häiritsejiltä kuten sateelta ja tuulelta. Suunnittelussa ei kuitenkaan otettu huomioon anturin pinnalle ajoittain tiivistyvää kosteutta, jonka havaittiin vaikuttavan anturin antamiin tuloksiin haitallisesti. Anturin sijoitus muutettiin tuplalasisen ikkunan puitteiden väliin. Ulommassa ikkunan karmissa ei ollut eristettä, jolloin anturin antamien arvojen todettiin vastaavan toleranssiensa rajoissa todellisia ulkona esiintyviä arvoja.

Myrskysäällä ajoittain esiintyvät sähkökatkokset vaikuttivat lopulliseen ratkaisuun. Photon kytkettiin alustavasti suoraan oman virtalähteensä kautta verkkoon. Ongelman esiintyttyä kytkentään lisättiin väliin varavirtalähde, jolla Photonille saatiin syötettyä virtaa usein vain muutamien tuntien kestävien sähkökatkojen ajan.

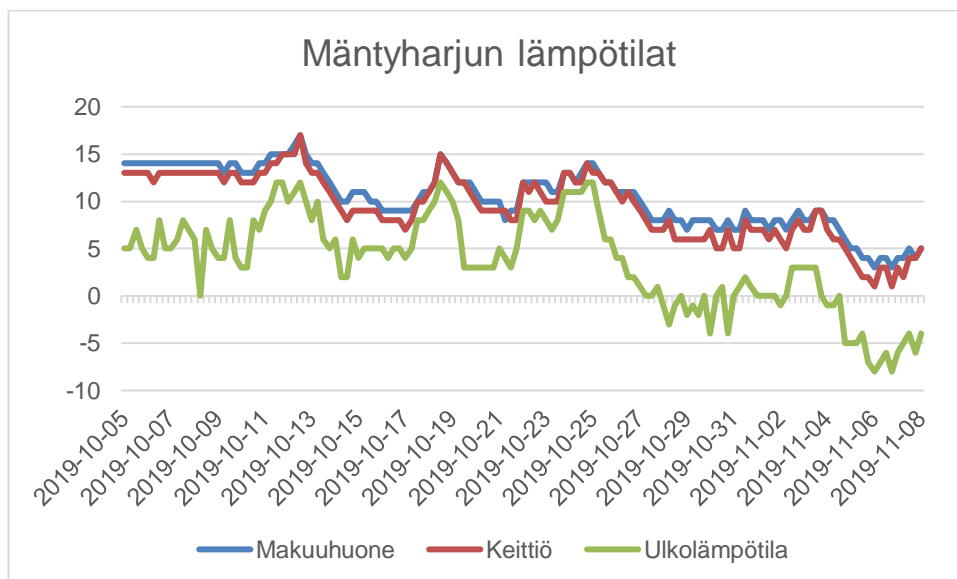
Photonin julkaisemissa historia tiedoissa paljastui noin viikon testijakson jälkeen mielenkiintoinen ongelma, jossa mitattuun ulkolämpötila arvoon ilmestyi lisäksi jotain siihen kuulumatonta todellisten mittaustuloksen eteen, kun arvoja tallennettiin Google Sheets'iin. Vika paikannettiin aiheutuvan IFTTT -applikaatiosta, sillä Particlen pilveen julkaistut lukemat vastasivat oletettuja. Ratkaisu päivitettiin käyttöliittymän puolelle niin että kaikki Sheets'istä haetut lämpöhistoria arvot ajetaan läpi ja ylimääräiset lukemat suodatetaan pois. Pääperusteena suodatukselle käytettiin oletusarvoa, jossa lämpötilan oletettiin aina olevan alle 50 astetta.

Aiemmin kuivanapitolämmitys kappaleessa kuvatuista syistä huoneiston kosteuden raja-arvoksi asetettiin 60 %, jonka arvioitiin pitävän huoneiston lämpötila pääsääntöisesti ulkoista korkeammalla ja asunto kuivana. Koska kiinteistön ilmastointi on painovoimalla toimiva, järjestelmässä on huomioitu myös ilmavirtauksen oikeasuunta. Tämä toteutettiin vertailemalla sisä- ja ulkolämpötila eroa ja eron ollessa liian pieni kytkettiin lämmitys päälle. Näin ollen huoneiston ilmavirtaus säilyy jatkuvasti ja ilma vaihtuu paine-eron vaikutuksesta, jolloin ylimääräinen kosteus poistuu tehokkaammin. Kosteusrajan asetuksessa huomioitiin myös mittauspisteitä hieman kosteampien kohtien mahdollisuus sekä ikkunankarmeille tiivistyvä kosteus, jossa kosteutta saattaa esiintyä jopa 15 % muuta huonetta enemmän.



Kuva 21. Mäntyharjun kiinteistön makuuhuoneen, keittiön ja ulkoilman suhteellisen kosteuden käyrät

Kuvaajasta 21 nähdään suhteellisen kosteuden kehitys kuuden tunnin välein mitattuna ajalta 5.10 – 08.11 ensin vakiolämmöllä ja sitten kosteuden ja lämpötilaeron perusteella säädettyinä. Aikavälillä 5.10 – 11.10, kun kosteutta rajoitettiin vielä vanhaan tyyliin, suhteellinen kosteus pysyy melko tasaisena lämpötilan pysyessä termostaatilla säädettyssä noin 13 asteessa. Tämän jälkeen, kun kosteuden säätö toteutetaan mittaamalla suhteellista kosteutta ja lämpötilaeroa huomataan kosteuden alkavan nousta lähelle säädettyä 60 % rajaa. Anturien mittaamat arvot liikkuvat noin  $\pm 2\%$  todellisista ja säädössä on 6 % hystereesi alaspäin, jonka vuoksi kosteusarvo vaihtelee hieman säädettyä alhaisemalla tasolla. Pidempinä yhtäjaksoisina pakkaskausina suhteellinen kosteus putoaa huomattavasti asetettua alemmas, koska automatiikka pitää huoneiston lämpötilan plusalla ja näin ollen myös lämpötilaero on suurempi. Koska säädössä seurataan sekä kosteutta että lämpötilaa kosteuskäyrä ei kerro kaikkea säädön käyttäytymisestä. Lämpötilan vaikutus näkyy selvästi 18.10 ja 19.10 välisenä aikana, jolloin ulkolämpötila on ollut selvästi aikaista korkeampi (kuva 22).

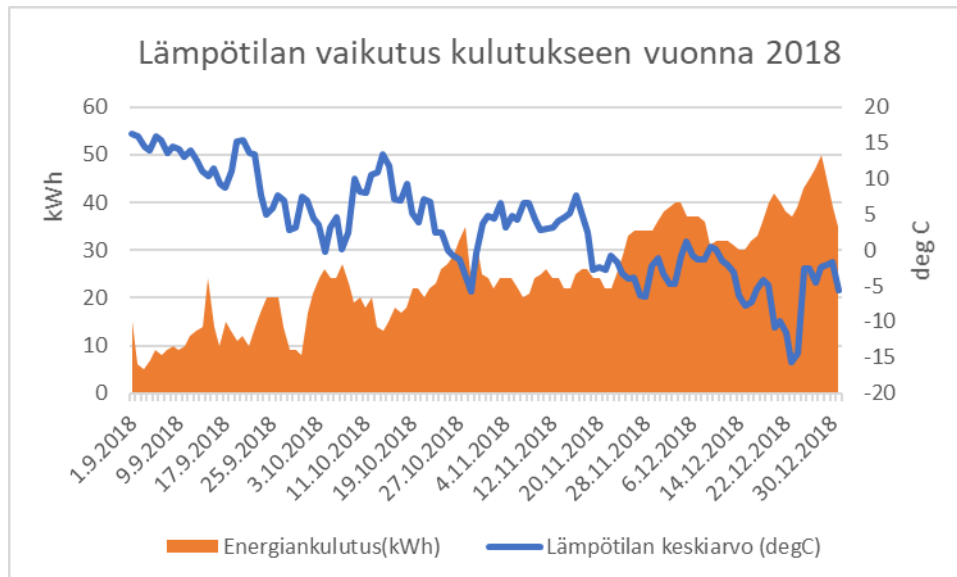


Kuva 22. Mäntyharjun kiinteistön makuuhuoneen, keittiön ja ulkoilman lämpötila käyrät

Kuvasta 22 havaitaan ulkolämpötilan nousseen 12 asteeseen ja tällöin myös sisäilman lämpötilaa on pyritty nostamaan. Näin ollen myös sisäilman kosteus alkaa laskea. 12.10 havaittava nopea ja lyhytaikainen lämpötilan nousu aiheutui säädön muuttamisesta

termostaattiohjatusta kosteus- ja lämpötilaeroperusteiseksi. Tällöin rakennusta lämmitettiin manuaaliajolla muutaman tunnin ajan. Tämän jälkeen sääto alkaa tiputtaa lämpötilaa oletetusti. Lämpötilaeron ollessa suuri lämpötilan laskiessa nopeasti pakkasen puolelle, kuten 25.10 – 27.10 välisenä aikana, myös kosteus poistuu rakenteista tehokkaasti, seuraten lämpötilaeron muutosta hieman jäljessä. Tällöin ilmankosteus putoaa jopa 10 % kohtalaisen lyhyessä ajassa. Aikoina, jolloin pakkasjakso kestää useampia päiviä, kuten 5.11 – 08.11 huomataan kuinka järjestelmä pitää rakennuksen sisäilman jatkuvasti plusasteiden puolella.

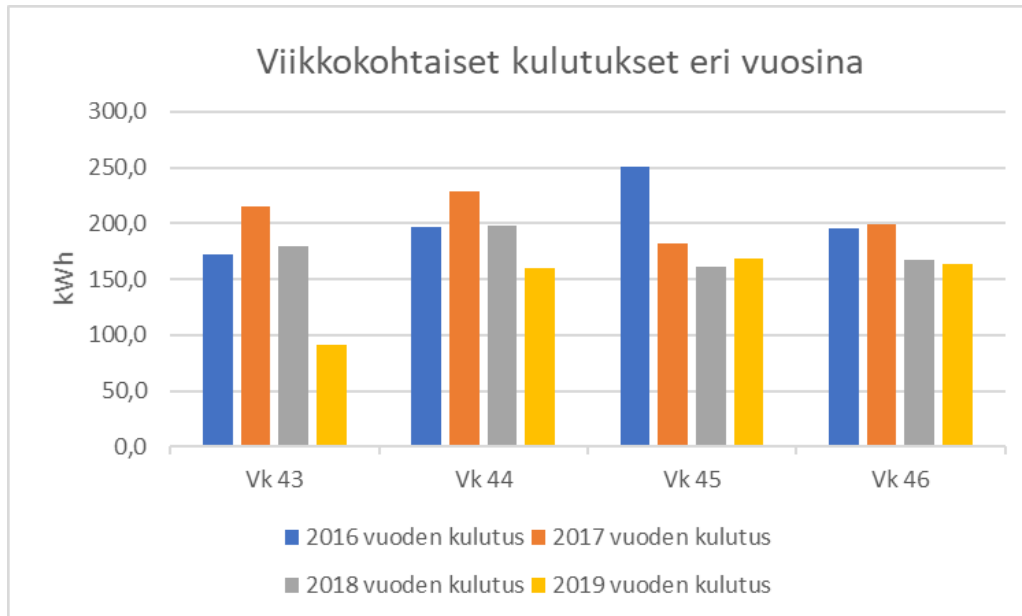
Yhtenä suurena vaikuttajana kulutukseen käytettäessä vakioämpötilassa pidettävää lämmitystä on ulkolämpötila, johon testausjakson ajalta saatuja tuloksia verrataan. Mikkelin sääasemalta (lähinnä kohdetta) tunnin välein kerätyn datan perusteella voidaan huomata, kuinka lämpötila vaikuttaa vakioämpötilalla lämmitetyn mökin kustannuksiin (kuva 23).



Kuva 23. Ulkolämpötilan vaikutus Mäntyharjun kiinteistön kulutukseen termostaattiohjauksella vuonna 2018

Kuvasta 23 havaitaan, kuinka merkittävästi ulkolämpötila vaikuttaa kiinteistön energian kulutukseen. Kustannuslaskelmissa on myös otettava huomioon ostoenergian hinnan kallistuminen muihin vuodelle nähden sen suuren kysynnän vuoksi. Vakioämpötilan verrattuna säädöllä, jolla lämpötilaero pyritään pitämään vakiona, saavutetaan siis aina suurempia säästöjä, mitä kylmempää ulkona on.

Testausjakson (viikko 43 - 46) ajalta kerättiin sähköyhtiön (Fortumin) toimittamasta sovelluksesta Mäntyharjun viikkokohtaiset energian kulutukset ja lämpötilat. Ensimmäisen viikon lämpötilakeskiarvoksi saatiin vuodelle 2016 noin 2,6 astetta. Vastaavalle ajalle lämpötiloiksi saatiin vuosina 2017 noin -0,5, Vuonna 2018 noin 1,1 ja 2019 noin 5,7.



Kuva 24. Eri vuosien kulutukset viikon välein mitattuna

Kuvasta 24 nähdään Mäntyharjulta viikoittain mitatut energiankulutukset Fortumin toimittaman sovelluksen perusteella. Kuvasta nähdään, että jo viikkotasolla 2019 vuoden kulutus on alhaisempi kuin useina muina vuosina. Lukuun ottamatta vuotta 2018 siirryttäessä viikosta 43 viikkoon 44 lämpötila on pudonnut ja on vaikuttanut välittömästi kulutukseen nostamalla sitä. Viikolla 45 vuoden 2016 ja 2019 lämpötilat ovat reilusti pakkasella ja muina vuosina on ollut huomattavasti lämpimämpää. Myös viikolla 46 vuoden 2018 lämpötilakeskiarvo on muuta huomattavasti korkeampi, joten myös kulutus on alhaisempi.

Vuoteen 2017 verrattuna testausjakson aikana energiaa säästettiin noin 240 kWh eli noin 30 %. Järjestelmän alkuinvestoinnin ollessa vain noin 100 euroa voidaan säästöä pitää huomattavana. Vuoden 2019 testausjakson tuloksia verrattiin vuoteen 2017 koska näillä vuosilla testausjakson aikaiset lämpötilakeskiarvot olivat lähimpinä toisiaan. Vuonna 2017 testausjaksonaikainen keskiarvo oli noin 0,70 ja vuonna 2019 noin 0,75 astetta

## 6.2 Ulvilan kiinteistö

Rakennukseen asennettiin aurinkoenergiajärjestelmä ja kerättiin tutkimuskäyttöön sen kokonaistuottamaa sekä päiväkohtaista energiamäärää etäluettavaksi 13.9.2019–21.10.2019. Koska aurinkoenergian tuotanto on heikkoa talviaikaan, järjestelmään liitettiin rinnalle myös sähkön pörssihinnan perusteella toteutettu lämmityksen ohjaus.

Aurinkoenergiajärjestelmään kuuluva invertteri mittasi aurinkopaneeleilla generoitua jännitettä ja virtaa, siihen liitetyn akuston jännitettä ja sieltä ulos otettavaa virtaa sekä verkon jännitettä ja taajuutta, mutta myös monia muita suureita. Kaikki mielenkiitoinen data saatiin paikan päälle asennetun palvelimen avulla etäluettavaan muotoon, johon suuri osa tutkimuksen tuloksista pohjautuu. Järjestelmän tuottamaa dataa on mahdollista tarkastella myös mobiililaitteella.

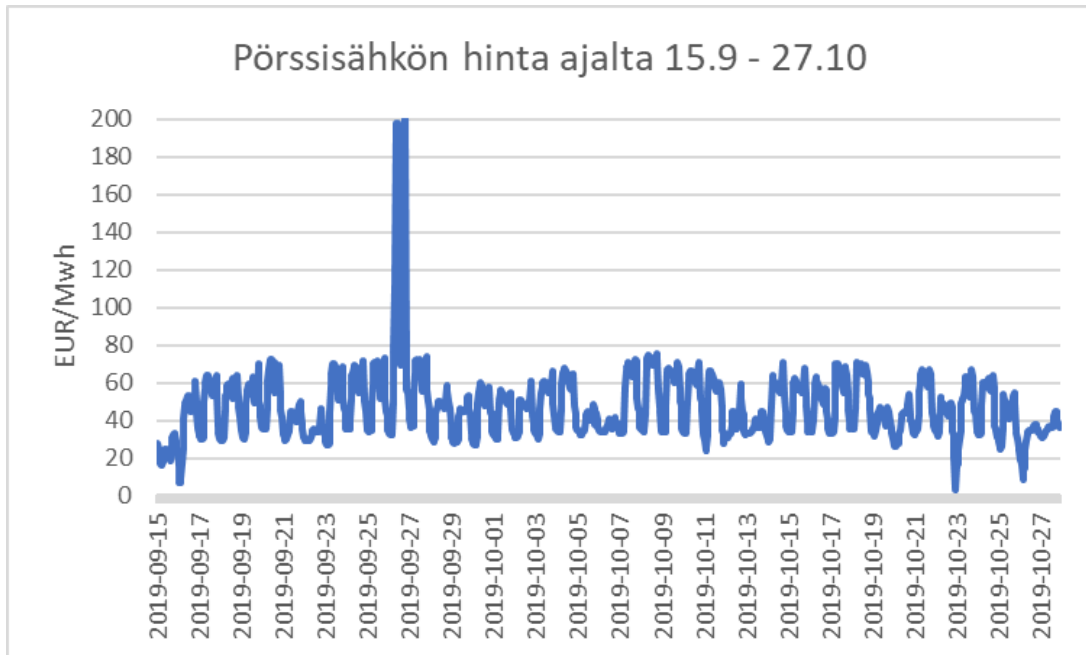
Asennetuilta paneeleilta kerätty laadukas mittajakso sijoittui 11.9.2019 – 1.10.2019, jolloin jokaiselta päivältä saatiin mitattua tuotettu energiamäärä. Tuotettu energiamäärä riippuu luonnollisesti pääsääntöisesti vuodenajasta ja siitä paistaako aurinko vai onko pilvistä. Mittaustulokset vaihtelivat 0,786 kWh ja 3,184 kWh välillä. Tämä mittajakso kesti 21 päivää ja ajalta laskettu keskiarvo oli 1,718 kWh/pvä. Tästä saadaan laskettua syyskuun energia tuotannoksi 51.54 kWh. Fortumin (51) kyseiselle alueelle tehdyn aurinkolaskuri kyselyn mukaan voidaan laskea, missä suhteessa energiaa pystytään tuottamaan kussakin kuukaudessa. Esimerkiksi kesäkuun kuukauden keskiarvon tulisi olla noin 2 -kertainen syyskuuhun nähden. Näin ollen vuoden aikana paneeleilla tuotettu energiamäärä on noin 600 kWh.

Takaisinmaksuaikaa laskettaessa on lukuisia tekijöitä, jotka vaikuttavat investoinnin lopulliseen kustannukseen. Näitä ovat luonnollisesti paneelien hankinta määrä ja laatu, muiden komponenttien saatavuus, asennuksen helppous sekä energian hinta tuotetulla hetkellä. Tämän työn kokonaisinvestointi kuluiksi arvioidaan noin 1400 euroa. Pienemmät komponentit löytyivät valmiina, joten varsinaisena investointina olivat invertteri, paneelit sekä akusto. Fortumin aurinkolaskuri sivustolla yhden kilowattitunnin arvoksi on määritetty 15 senttiä, jos kaikki energia käytetään. Näillä arvoilla laskettuna takaisinmaksuajaksi saataisiin noin 15,6 vuotta.

Järjestelmään asennettiin myös lämmitystapaa hinnan perusteella muuttava systeemi, joka toteuttaa säädön halvimalla käytössä olevalla tavalla. Järjestelmällä tuotettuja säästöjä tarkkailtiin syyskuun puolivälistä lokakuun loppuun Photonin laskurilla ja sähköpörssistä.

Talvilaatuisenlämmitysöljyn hinta vaihtelee sijainnista ja tilauspaikasta riippuen hieman 1,1 €/l alapuolella. Tästä öljylle laskettu kilowattituntihinta noin 11 c/kWh. Sähkön hinta puolestaan koostuu pörssisähkösopimuksella perusmaksusta (€/kk), sähköverosta (c/kWh), siirtomaksusta (c/kWh) ja sähköpörssihinnasta (c/kWh). Kiinteistön kokonaiskulutukseksi on arvioitu aiempien vuosien energialaskutusten perusteella noin 30 000 kWh. Näin ollen yö sähkö tai kausi sopimusta hyväksi käyttämällä pörssisähkön hinnan raja-arvoksi, jolla saavutettaisiin taloudellista hyötyä, määräytyi noin 4,0 c/kWh.

Pörssisähkön hinta määräytyy Nord Pool -sähköpörssissä. Sähköä ei pysty ostamaan suoraan pörssistä vaan se tilataan sähköyhtiön kautta, joka lisää sopimuksen mukaisen maksun (joitakin euroja kuukaudessa) sekä noin 0,3 c/kWh jälleenmyytyyn hintaan. Nord Poolissa hinnat määräytyvät tunneittain ja hintojen vaihteluväli on suuri. Kesäaikaan kun kulutus on pienempi, hinnat saattavat useinkin olla alle 1 c/kWh ja talviaikaan kulutuksen kasvaessa hinnat saattavat puolestaan olla jopa 20 c/kWh. Myös muina vuoden aikoina hintavaihtelu on suurta.



Kuva 25. Tunneittain vaihtelevan pörssisähkön kuvaaja

Tutkimusjakson aikana hinnat vaihtelivat 0,7 c/kWh ja 20 c/kWh välillä. Kuvaajasta 25 nähdään Nord Poolista haettu pörssinsähkön hinta, johon sähköyhtiö lisää aikaisemmin mainitut lisät. Näin ollen määritelty hintaraja alitettiin tarkkailujakson aikana useasti, mutta lopulliset taloudelliset hyödyt jäivät kohtalaisen pieniksi, sillä hinta määräytyi pääsääntöisesti määriteltyä rajaa ylemmäksi. Vastaavia tuloksia saatiin Photonilla ajettavalla laskurilla, jolla havainnollistettiin aikaa, jolloin jälleenmyyntihinta on ollut alle 4,0 c/kWh.

Hinnan tarkkailulla saavutettavat säästöt sijoittuvatkin suunnitellun talviajan sijasta enimmäkseen loppukesään ja alkukevääseen, koska silloin sähkön hinta on usein halvempaa. Tällöin tosin myös lämmitystarve on talviaikaa pienempi mutta kulutusta on silti. Toisaalta hinnanseurantaan käytetty investointi oli vain kymmeniä euroja ja näin ollen takaisin maksuajaksi arvioidaan 1 – 2 vuotta.

## 7 Yhteenveto

Insinööriyön tavoitteena oli toteuttaa tutkimus, jolla selvitetään tapoja säästää omakotitalon lämmityskustannuksissa ja pienentää hiilijalanjälkeä hyödyntäen automaatiota. Tavoitteena oli myös parantaa lämmitysjärjestelmän etävalvontamahdollisuuksia ja lisätä näin tietoisuutta rakenteiden kunnosta ja lämmitysjärjestelmän toiminnasta sekä tehdä järjestelmän optimoiminen helpommaksi. Työ tehtiin kahteen eri kiinteistöön, joista toinen oli vain vapaa-ajan ja toinen vakituksessa käytössä.

Tiedonsiirto lämmitysjärjestelmästä pilveen toteutettiin molemmissa kiinteistöissä eri tyyliin. Mäntyharjun vapaa-ajan asunnosta dataa vietiin käyttöliittymään kahdella tavalla. Historia tiedot vietiin Google Sheets:iin julkaisemalla Particlen pilveen Photonilla mitatut lämpö- ja kosteus lukemat ja käyttämällä julkaisua laukaisevana tekijänä IFTTT -sovellukselle, jonka avulla data kirjoitettiin taulukkoon pilveen. Järjestelmästä haettiin käyttöliittymään myös reaaliaikaista dataa, kuten lämmittimien tilatietoja sekä lämpötilojen ja kosteuksien nykyarvoja, jolloin data haettiin Particlen pilven API:n kautta. Lisäksi laitteelle pystyttiin lähettämään komentoja, kuten päälle tai pois käskyjä sekä erilaisia sääntöön vaikuttavia raja-arvoja. Vastaavasti myös komennot välitettiin laitteelle Particlen pilven kautta.

Uvilan järjestelmässä käyttöliittymään tuotu data tuotettiin paikan päälle asennetulla invertterillä. Samaan sijaintiin asennettu palvelin kyselee Visual Studiossa ajettavalla ohjelmalla invertteriltä siihen kytkettyjen laitteiden jänniteitä, virtoja ja taajuuksia sekä vika-koodeja RS-232 -sarjaväylän kautta. Palvelimella ajettu ohjelma tallentaa kyselyn tulokset SQL-tietokantaan, josta data haettiin käyttöliittymään Conveyor pilven kautta. Palvelin ajaa myös rinnakkaissovellusta, jolla tarkkaillaan pörssisähkön hintaa Fortumin sivuilta. Palvelin lähettää hakemansa ja esiprosessoimansa datan lämmitystä ohjaavalle Photonille, jonka perusteella verkkosähkölämmitys kytketään päälle tai pois.

Saatujen tulosten perusteella voidaan koko järjestelmä todeta taloudellisen kannattavuuden osalta hyväksi pitkällä aikavälillä. Monesti aurinkoenergiajärjestelmän käyttöiäksi arvioidaan kymmeniä vuosia ja kalleimmilla osilla eli paneeleilla tehontuotto takuu on usein 25 vuotta. Järjestelmän takaisinmaksuaika todettiin noin 15,6 vuodeksi, jolloin järjestelmään tehtävää laajennusta voidaan pitää kannattavana. Järjestelmää laajennettaessa on tärkeää välttää ylimitoittaminen, koska sähkön verkkoon myyntiä ei ainakaan

lähitulevaisuudessa nähdä investoinnin suuruuteen nähden kannattavaksi. Ulvilan ratkaisussa myös öljyn kulutusta ja sitä kautta myös pienhiukkaspäästöjä saatiin vähennettyä.

Monesti hybridilämmitykseen asennettava lisäenergialähteen kuten aurinkosähkön tai poistoilmapumpun takaisinmaksuaika vaihtelee 15 vuoden molemmin puolin riippuen monista tekijöistä, kuten omakotitalon iästä, eristyksistä ja kiinteistön koosta. Omavaraisenergian investoinnin takaisinmaksuaikaa tulee myös lyhentämään tulevaisuudessa arvioitu vaihtoehtokustannusten kuten, sähkön hinnan kallistuminen.

Toisaalta esimerkiksi maalämpöpumpulla voidaan tulevaisuudessa korvata rakennuksen päälämmitystapa ja liittää aurinkoenergiajärjestelmä osaksi sitä ja näin saavuttaa vielä huomattavasti suurempia säästöjä pitkällä aikavälillä. Huomioitavaa on myös öljyn litrahinta tulevaisuudessa, joka tulee mahdollisesti nousemaan siitä aiheutuvien ympäristövaikutusten toimesta. Lisäksi öljylämmitteisiä rakennuksia kannustetaan siirtymään muihin lämmitysmuotoihin ja fossiilisen öljyn käytöstä pyritään luopumaan asteittain vuoden 2030 alkuun mennessä.

Mäntyharjun ratkaisussa puolestaan järjestelmän takaisinmaksuaika määräytyi äärimäisen lyhyeksi (noin 3kk), alkuinvestoinnin ollessa noin 100 euroa ja energiankulutuksen laskiessa kymmeniä prosentteja talvikaudessa. Toisaalta järjestelmän hintaan huomattavasti vaikuttavia tekijöitä ovat anturien laatu ja niiden määrä sekä asennukseen ja käyttöönottoon vaadittu aika. Näin ollen halvemman järjestelmän toimintakuntoon pystyttäminen vie enemmän aikaa verrattuna parempilaatuisiin antureihin ja kontrollereihin. Tämä johtuu pitkälti antureilta tulevien virheiden ja häiriöiden suodattamisesta, mikä aiheuttaa ylimääräistä työtä.

Kuivanapitolämmitysjärjestelmän tuomiin säästöihin vaikuttavat myös olennaisesti vallitseva sää, rakennuksen normaali energiankulutus, säätömalli, energian hinta ja järjestelmän hinta.

## Lähteet

- 1 Asumisen energiankulutus laski hieman vuonna 2017. Verkkoaineisto. Tilastokeskus. <[https://www.stat.fi/til/asen/2017/asen\\_2017\\_2018-11-22\\_tie\\_001\\_fi.html](https://www.stat.fi/til/asen/2017/asen_2017_2018-11-22_tie_001_fi.html)>. Luettu 03.10.2019.
- 2 Ojanen Miika & Duktig Anssi. 2018. Älykäs kuivanapitolämmitys. Insinööriyö. Turun Ammattikorkeakoulu. Theseus-tietokanta.
- 3 Hybridilämmitys. 2018. Verkkoaineisto. Energiatehokas koti. <[https://www.energiatehokaskoti.fi/suunnittelu/talotekniikan\\_suunnittelu/lammitys/hybridilammitys](https://www.energiatehokaskoti.fi/suunnittelu/talotekniikan_suunnittelu/lammitys/hybridilammitys)>. Luettu 04.10.2019.
- 4 Lämmitys. 2018. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/L%C3%A4mmitys>>. Luettu 23.09.2019.
- 5 Pientalon lämmitysjärjestelmät. 2019. Verkkoaineisto. Suomi Rakentaa. <<https://www.suomirakentaa.fi/omakotirakentaja/laemmitys/laemmityksen-valinta>>. Luettu 24.09.2019.
- 6 Hybridilämmitys. 2019. Verkkoaineisto. Eko Lämmöx. <<https://www.ekolammox.fi/hybridilammitys/>>. Luettu 04.10.2019.
- 7 Maalämpö. 2019. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/Maal%C3%A4mp%C3%B6>>. Luettu 23.09.2019
- 8 Geothermal energy. Verkkoaineisto. Wikipedia. <[https://en.wikipedia.org/wiki/Geothermal\\_energy](https://en.wikipedia.org/wiki/Geothermal_energy)>. Luettu 23.09.2019.
- 9 Geothermal heat pump. 2019. Verkkoaineisto. Wikipedia. <[https://en.wikipedia.org/wiki/Geothermal\\_heat\\_pump](https://en.wikipedia.org/wiki/Geothermal_heat_pump)>. Luettu 23.09.2019.
- 10 Öljylämmitys. 2019. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/%C3%96ljyl%C3%A4mmitys>>. Luettu 24.09.2019.
- 11 Lämmitysjärjestelmän valinta. 2018. Verkkoaineisto. Energiatehokas koti. <[https://www.energiatehokaskoti.fi/suunnittelu/talotekniikan\\_suunnittelu/lammitys](https://www.energiatehokaskoti.fi/suunnittelu/talotekniikan_suunnittelu/lammitys)>. Luettu 24.09.2019.
- 12 Sähkölämmitys. 2018. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/S%C3%A4hk%C3%B6l%C3%A4mmitys>>. Luettu 25.9.2019.

- 13 Poistoilmalämpöpumppu. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/Poistoilmal%C3%A4mp%C3%B6pumppu>>. Luettu 25.9.2019.
- 14 Poistoilmalämpöpumppu. 2019. Verkkoaineisto. Motiva. <[https://www.motiva.fi/ratkaisut/uusiutuva\\_energia/lampopumput/lampopumpputeknologiat/poistoilmalampopumppu](https://www.motiva.fi/ratkaisut/uusiutuva_energia/lampopumput/lampopumpputeknologiat/poistoilmalampopumppu)>. Luettu 25.9.2019
- 15 Aurinkoenergia. 2019. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/Aurinkoenergia>>. Luettu 10.10.2019.
- 16 Hybridilämmitys kannattavin lämmitysratkaisu uusissa omakotitaloissa. 2019. Verkkoaineisto. Suomela. <<https://www.suomela.fi/hybridilammitys-kannattavin-lammitysratkaisu-uusissa-omakotitaloissa/>>. Luettu 23.9.2019.
- 17 Niskanen, Saara. 2016. Maalämpö on edullisin ja sähkö kallein tapa lämmittää omakotitaloa pitkällä aikavälillä, suuntaa antava laskelma paljastaa. Verkkoaineisto. Taloussanomat. <<https://www.is.fi/taloussanomat/art-2000001923312.html>>. Päivitetty 4.10.2016. Luettu 24.9.2019.
- 18 Laaksonen, Iko. 2017. Diplomityö. Tampereen teknillinen yliopisto. dspace. <<https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/24779/Laaksonen.pdf?sequence=1&isAllowed=y>>. Luettu 20.09.2019.
- 19 Vinha, Juha. 2016. Vakiotehoinen kuivanapitolämmitys on mökeillä parempi ratkaisu kuin perinteinen peruslämpö. Verkkoaineisto. Meillä kotona. <<https://www.meillakotona.fi/artikkelit/kuivanapitolammitys-kannattaa-mokilla>>. Luettu 10.10.2019.
- 20 Capacitive Humidity Sensors. 2015. Verkkoaineisto. Metropolia. <<https://wiki.metropolia.fi/display/sensor/Capacitive+Humidity+Sensors>>. Luettu 19.9.2019
- 22 Kuivanapitolämmitys säästää ja kesämökki säilyy kunnossa yli talven. 2018. Verkkoaineisto. Arki virtaa. <<https://energiaa.ksoy.fi/kuivanapitolammitys-pitaa-kesamokin-kunnossa/>>. Luettu 18.9.2019
- 23 Nyman, Antti. 2016. Suomalainen homemalli. Verkkoaineisto. Rakennusfysiikka <<http://www.anttinyman.fi/category/rakennusfysiikka/>>. Luettu 10.10.2019.
- 24 Nord Pool. Verkkoaineisto. 2019. Wikipedia. <[https://fi.wikipedia.org/wiki/Nord\\_Pool](https://fi.wikipedia.org/wiki/Nord_Pool)>. Luettu 30.9.2019.
- 25 Day-ahead prices. 2019. Verkkoaineisto. Nord Pool. <<https://www.nordpool-group.com/Market-data1/Dayahead/Area-Prices/FI/Hourly/?view=table>>. Luettu 30.9.2019.

26 Choosing a Humidity Sensor: A Review of Three Technologies. 2001. Verkkoaineisto. FierceElectronics. <<https://www.fierceelectronics.com/components/choosing-a-humidity-sensor-a-review-three-technologies>>. Luettu 19.9.2019.

27 Kiiski, Tuomas. 2018. Kosteusanturien kestävyys pitkäkestoisessa betonin kosteusmittauksessa. Insinööriyö. Turun Ammattikorkeakoulu. Theseus-tietokanta.

28 DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino. 2016. Verkkoaineisto. How To Mechatronics. <<https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/>>. Luettu 19.9.2019

29 DHT22 Temperature-Humidity Sensor. 2018. Verkkoaineisto. Waveshare. <[https://www.waveshare.com/wiki/DHT22\\_Temperature-Humidity\\_Sensor](https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor)>. Luettu 19.9.2019

30 RF module. 2019. Verkkoaineisto. Wikipedia. <[https://en.wikipedia.org/wiki/RF\\_module](https://en.wikipedia.org/wiki/RF_module)>. Luettu 20.9.2019.

31 How to Interface With RF Transmitter and Receiver. 2019. Verkkoaineisto. Instructables. <<https://www.instructables.com/id/How-to-Interface-With-RF-Transmitter-and-Receiver/>>. Luettu 20.9.2019.

32 RF Module – Transmitter & Receiver. Verkkoaineisto. 2019. Elprocus. <<https://www.elprocus.com/rf-module-transmitter-receiver/>>. Luettu 20.9.2019.

33 Microprocessor development board. Verkkoaineisto. 2019. Wikipedia. <[https://en.wikipedia.org/wiki/Microprocessor\\_development\\_board](https://en.wikipedia.org/wiki/Microprocessor_development_board)>. Luettu 16.9.2019.

34 Photon datasheet. 2019. Verkkoaineisto. Particle docs. <<https://docs.particle.io/datasheets/wi-fi/photon-datasheet/#pinout-diagrams>>. Luettu 16.9.2019

35 Particle Photon. 2019. Verkkoaineisto. Zerynth Docs. <[https://docs.zerynth.com/latest/official/board.zerynth.particle\\_photon/docs/index.html](https://docs.zerynth.com/latest/official/board.zerynth.particle_photon/docs/index.html)>. Luettu 20.9.2019.

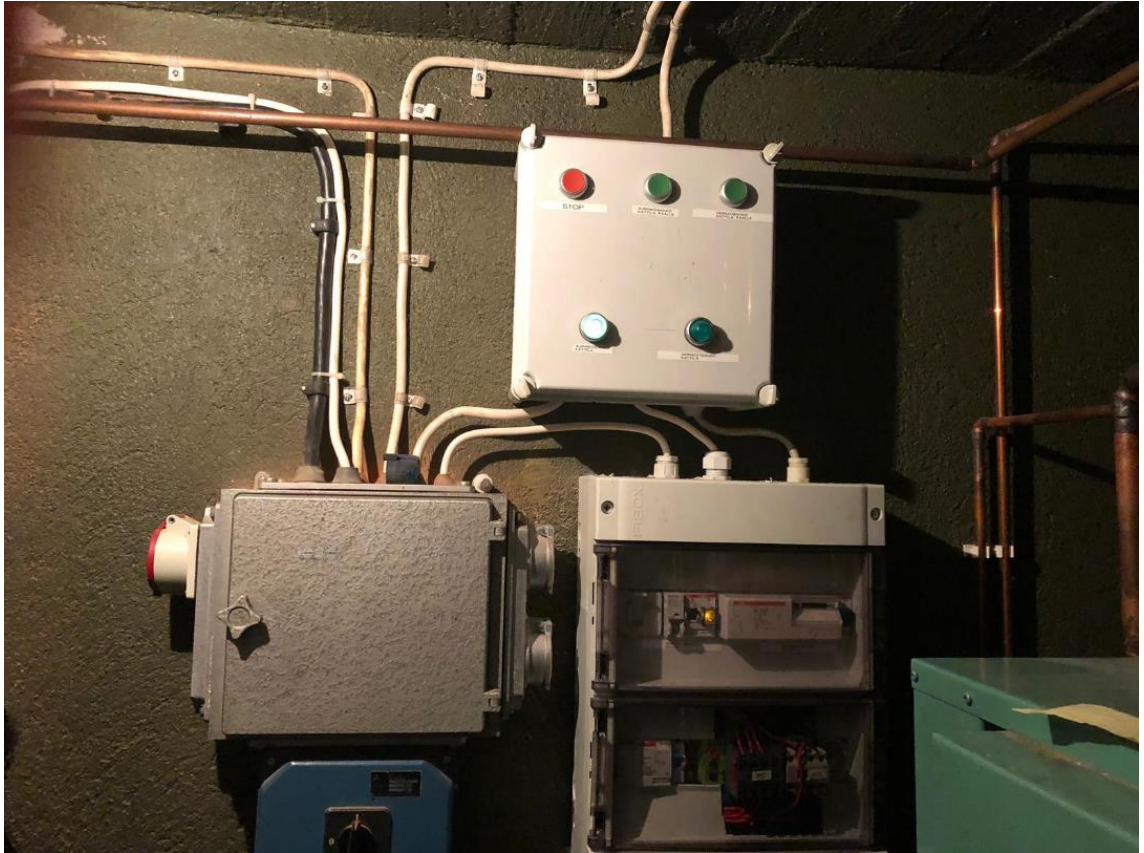
36 Authentication. 2019. Verkkoaineisto. Particle docs. <<https://docs.particle.io/tutorials/device-cloud/authentication/>>. Luettu 16.9.2019.

37 Desktop IDE. 2019. Verkkoaineisto. Particle Docs. <<https://docs.particle.io/tutorials/developer-tools/dev/>>. Luettu 16.9.2019.

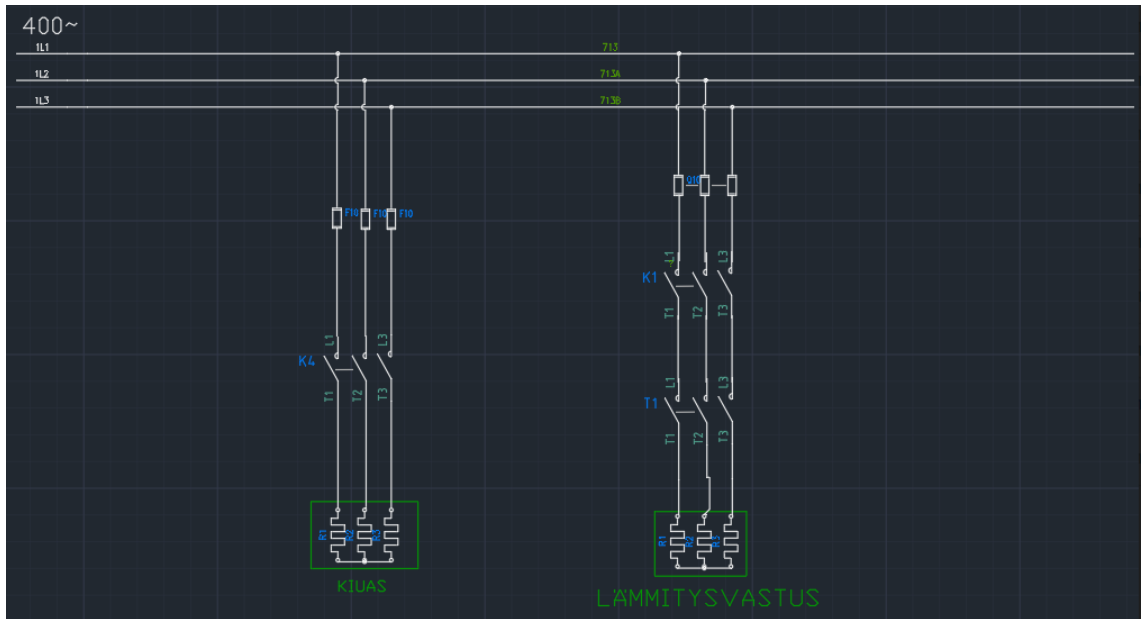
38 Put the Internet to Work with IFTTT & Particle. 2016. Verkkoaineisto. Core Electronics. <<https://core-electronics.com.au/tutorials/using-ifttt-with-particle.html>>. Luettu 03.10.2019.

- 39 Aurinkokenno. 2019. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/Aurinkokenno>>. Luettu 23.9.2019
- 40 Niemi, Miro. 2016. Sähköajoneuvojen akusto ja kestoikä. Insinööriyö. Metropolia Ammattikorkeakoulu. Theseus tietokanta
- 41 Power inverter. 2019. Verkkoaineisto. Wikipedia. <[https://en.wikipedia.org/wiki/Power\\_inverter](https://en.wikipedia.org/wiki/Power_inverter)>. Luettu 28.9.2019.
- 42 Inverter. 2019. Verkkoaineisto. ScienceDirect. <<https://www.sciencedirect.com/topics/engineering/inverter>>. Luettu 28.9.2019
- 43 Solar inverter. 2019. Verkkoaineisto. Wikipedia. <[https://en.wikipedia.org/wiki/Solar\\_inverter](https://en.wikipedia.org/wiki/Solar_inverter)>. Luettu 28.9.2019.
- 44 Ryhmäkeskus. 2016. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/Ryhm%C3%A4keskus>>. Luettu 29.9.2019.
- 45 Microsoft Visual Studio. 2019. Verkkoaineisto. Wikipedia. <[https://fi.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://fi.wikipedia.org/wiki/Microsoft_Visual_Studio)>. Luettu 3.10.2019.
- 46 Karttunen, Henri. 2009. Opinnäytetyö. Lahden Ammattikorkeakoulu. Theseus-tietokanta.
- 47 Communication Manual. 2017. Verkkoaineisto. offgrid. <[http://www.offgrid.casa/wp-content/uploads/2017/10/HS\\_MS\\_MSX\\_RS232\\_Protocol\\_20140822\\_after\\_current\\_upgrade.pdf](http://www.offgrid.casa/wp-content/uploads/2017/10/HS_MS_MSX_RS232_Protocol_20140822_after_current_upgrade.pdf)>. Luettu 10.10.2019.
- 48 HTML. Verkkoaineisto.2019. Wikipedia. <<https://fi.wikipedia.org/wiki/HTML>>. Luettu 17.9.2019.
- 49 JavaScript. 2019. Verkkoaineisto. Wikipedia <<https://fi.wikipedia.org/wiki/JavaScript>>. Luettu 17.9.2019
- 50 AJAX Introduction. 2019. Verkkoaineisto. W3schools. <[https://www.w3schools.com/js/js\\_ajax\\_intro.asp](https://www.w3schools.com/js/js_ajax_intro.asp)>. Luettu 17.9.2019.
- 51 Laske aurinkoenergian tuotanto talossasi. 2019. Verkkoaineisto. Fortum. <<https://aurinkolaskuri.fortum.fi/>>. Luettu 3.10.2019.

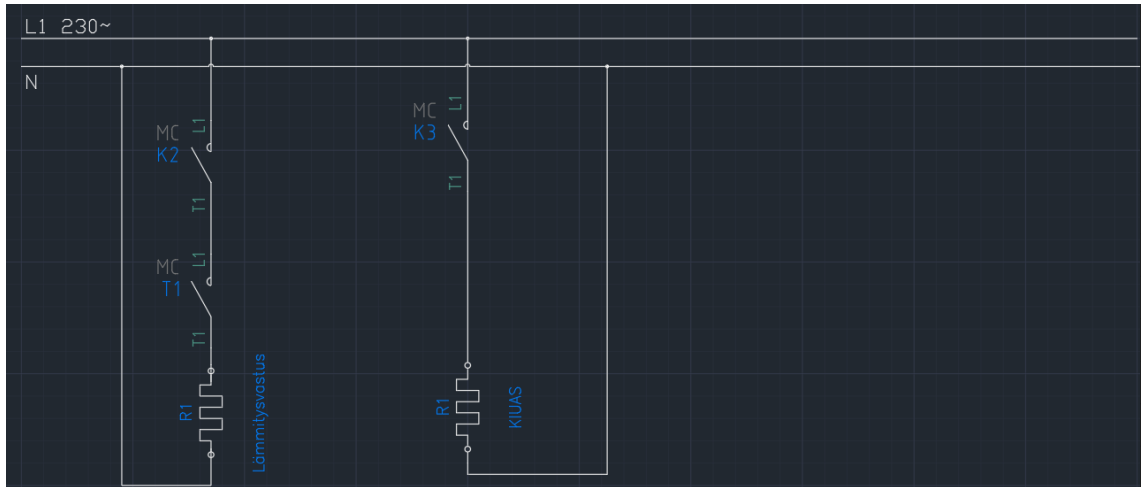
Liite 1. Ulvilan Ryhmäkeskus ja ohjauskaappi



### Liite 2. Keskuksen 3-vaiheinen päävirtapiiri



### Liite 3. Keskuksen yksivaiheinen päävirtapiiri



**Liite 4. Palvelimen API rajapinta**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace WebApplication.Controllers
{
    public class ValuesController : ApiController
    {
        // GET api/values
        public string Get()
        {
            ReadAndPublish GetData = new ReadAndPublish();
            return GetData.GetGridValuesSql();
        }

        // GET api/values/5
        public string Get(int id)
        {
            ReadAndPublish GetData = new ReadAndPublish();
            // Get grid values
            if (id == 1)
            {
                return GetData.GetGridValuesSql();
            }
            //Get Output values
            else if (id == 2)
            {
                return GetData.GetOutputValuesSql();
            }
            //Get Output load percent values
            else if(id == 3)
            {
                return GetData.GetOutputLoadPrecentSql();
            }
            //Get battery values
            else if (id == 4)
            {
                return GetData.GetBatteryValuesSql();
            }
            //Get Pv values
            else if (id == 5)
            {
                return GetData.GetPvValuesSql();
            }
            else if (id == 6)
            {
                return GetData.FaultsAndAlarms();
            }
            else if (id == 7)
            {
                return GetData.Commands();
            }
        }
    }
}
```

```
    }
    else if (id== 8)
    {
        return GetData.GetEnergyObtained();
    }
    else if(id == 9)
    {
        return GetData.GetHistory();
    }

    else
    {
        return "No response, check the Id called";
    }
}

// POST api/values
public void Post([FromBody]string value)
{
}

// PUT api/values/5
public void Put(int id, [FromBody]string value)
{
}

// DELETE api/values/5
public void Delete(int id)
{
}
}
```

**Liite 5. Palvelimen pääohjelma (osa)**

```
public class ReadAndPublish
{
    static SerialPort _serialPort;
    public string theData;
    public static int i;
    public static int ApiCommands = 0;
    public static float passedTime = 1/60F;           //Time count in hours
    public static float ReadGainedEnergy;
    public static string WriteGainedEnergy;
    public static float HistoryEnergy = 0F;
    public static bool onStartup = false;
    public static string PvValues;
    public static bool QPIGSRead = false;

    public static void Main()
    {
        //Count the amount of energy gained from solar panels
        ReadAndPublish CallKwhObj = new ReadAndPublish();
        Thread CallKwhCounter = new Thread(CountKwh);
        CallKwhCounter.Start();
        _serialPort = new SerialPort();
        // Allow the user to set the appropriate properties.
        _serialPort.PortName = SetPortName(_serialPort.PortName);
        _serialPort.BaudRate = 2400;
        _serialPort.Open();
        SqlConnection con;
        string str;

        while (true)
        {

            //First Read QPIGS!

            _serialPort.Write(new byte[] { 0x51, 0x50, 0x49, 0x47,
0x53, 0xb7, 0xa9, 0x0d }, 0, 8); //QPIGS

            string message = _serialPort.ReadExisting();
            // Console.WriteLine(message);
            string[] splittedMessage = message.Split(' ');
            //To get the whole reading
            while (splittedMessage.Length < 18)
            {
                message += _serialPort.ReadExisting();
                splittedMessage = message.Split(' ');
                Thread.Sleep(200);
            }
            //To avoid index error
            if (splittedMessage.Length >= 18)
            {
```

```

        Console.WriteLine(message);

        //Value meanings from left to right Grid voltage, Grid
frequency
        string GridValues = splittedMessage[0] + " " + split-
tedMessage[1];
        //Output voltage, output frequency
        string OutputValues = splittedMessage[2] + " " +
splittedMessage[3];
        //OutputLoad percent [W% or VA%]
        string OutputLoadPercent = splittedMessage[6];
        //Battery voltage, battery charging current
        string BatteryValues = splittedMessage[8] + " " +
splittedMessage[9];
        //Pv input current for battery and Pv input voltage
        PvValues = splittedMessage[12] + " " + splittedMes-
sage[13];

        str = @"Data Source=(LocalDB)\MSSQLLocalDB;At-
tachDbFilename=C:\Users\WebApplication\WebApplication\AppData\Data-
base1.mdf;Integrated Security=True";
        con = new SqlConnection(str);
        con.Open();
        Console.WriteLine("Database connected");
        string query = "INSERT INTO MyUsbValues
(GridValues,OutputValues,OutputLoadPercent,BatteryValues,PvValues)
VALUES('" + GridValues + "', '" + OutputValues + "', '" + Output-
LoadPercent + "', '" + BatteryValues + "', '" + PvValues + "')";

        SqlCommand ins = new SqlCommand(query, con);

        ins.ExecuteNonQuery();
        // Outputins.ExecuteNonQuery();
        Console.WriteLine("\n Data stored Into Database");

        con.Close();
        QPIGSRead = true;
    }

    Thread.Sleep(25000);
}

//Use api to get grid values from sql
public string GetGridValuesSql()
{
    SqlConnection con;
    // str = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFile-
name=C:\Users\WebApplication\WebApplication\AppData\Data\Datase1.mdf;In-
tegrated Security=True";
    con = new SqlConnection(str);
    con.Open();
    // string q = "SELECT * FROM UsbValues";

```

```
        string q = "SELECT TOP 1 MyUsbValues.GridValues FROM
MyUsbValues ORDER BY Id DESC";
        SqlCommand view = new SqlCommand(q, con);
        SqlDataReader dr = view.ExecuteReader();
        while (dr.Read())
        {
            theData = dr.GetValue(0).ToString();
        }
        con.Close();
        return theData;
    }
    //Use api to get output values from sql
    public string GetOutputValuesSql()
    {
        // Console.WriteLine("You are now in the readSQL part");
        SqlConnection con;
        // str = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFile-
name=C:\Users\WebApplication\WebApplication\AppData\Databasel.mdf;In-
tegrated Security=True";
        con = new SqlConnection(str);
        con.Open();
        // string q = "SELECT * FROM UsbValues";
        string q = "SELECT TOP 1 MyUsbValues.OutputValues FROM
MyUsbValues ORDER BY Id DESC";
        SqlCommand view = new SqlCommand(q, con);
        SqlDataReader dr = view.ExecuteReader();
        while (dr.Read())
        {
            theData = dr.GetValue(0).ToString();
        }
        con.Close();
        return theData;
    }
    //Use api to get OutputLoadPercent values from sql
    public string GetOutputLoadPrecentSql()
    {
        // Console.WriteLine("You are now in the readSQL part");
        SqlConnection con;
        // str = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFile-
name=C:\Users\WebApplication\WebApplication\AppData\Databasel.mdf;In-
tegrated Security=True";
        con = new SqlConnection(str);
        con.Open();
        // string q = "SELECT * FROM UsbValues";
        string q = "SELECT TOP 1 MyUsbValues.OutputLoadPercent FROM
MyUsbValues ORDER BY Id DESC";
        SqlCommand view = new SqlCommand(q, con);
        SqlDataReader dr = view.ExecuteReader();
        while (dr.Read())
        {
            theData = dr.GetValue(0).ToString();
        }
        con.Close();
    }
```

```
        return theData;
    }
    //Use api to get BatteryValues values from sql
    public string GetBatteryValuesSql()
    {
        Console.WriteLine("You are now in the readSQL part");
        SqlConnection con;
        // str = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFile-
name=C:\Users\WebApplication\WebApplication\AppData\Data\Database1.mdf;In-
tegrated Security=True";
        con = new SqlConnection(str);
        con.Open();
        // string q = "SELECT * FROM UsbValues";
        string q = "SELECT TOP 1 MyUsbValues.BatteryValues FROM
MyUsbValues ORDER BY Id DESC";
        SqlCommand view = new SqlCommand(q, con);
        SqlDataReader dr = view.ExecuteReader();
        while (dr.Read())
        {
            theData = dr.GetValue(0).ToString();

        }
        con.Close();
        return theData;
    }
    //Use api to get Pv values from sql
    public string GetPvValuesSql()
    {

        SqlConnection con;
        // str = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFile-
name=C:\Users\WebApplication\WebApplication\AppData\Data\Database1.mdf;In-
tegrated Security=True";
        con = new SqlConnection(str);
        con.Open();
        // string q = "SELECT * FROM UsbValues";
        string q = "SELECT TOP 1 MyUsbValues.PvValues FROM
MyUsbValues ORDER BY Id DESC";
        SqlCommand view = new SqlCommand(q, con);
        SqlDataReader dr = view.ExecuteReader();
        while (dr.Read())
        {
            theData = dr.GetValue(0).ToString();

        }
        con.Close();
        return theData;
    }
}
```

**Liite 6. Ulvilan Photonilla ajettava ohjelma**

// This #include statement was automatically added by the Particle IDE.

```
#include <TimeAlarms.h>
```

```
int heater = D5;
```

```
int pos = 1000;
```

```
String position = "00";
```

```
int value = 650;
```

```
int Price = 0;
```

```
int status = 0;
```

```
bool automatic = false;
```

```
Timer minutesUsedLoweredCosts(60000,LoweredCosts);
```

```
int loweredCostsCountInMinutes = 0;
```

```
void setup()
```

```
{
```

```
  Time.zone(1);
```

```
  pinMode(test,OUTPUT);
```

```
  //Eurojen eli pilkun vasemman puoleisen arvon haku
```

```
  Particle.function("heater",heaterControl);
```

```
  pinMode(heater, OUTPUT);
```

```
  //Senttien eli pilkun oikean puoleisen arvon haku
```

```
  Particle.function("led",ledControl);
```

```
  //Vertailuarvon haku
```

```
  Particle.function("setValue",setPrice);
```

```
  Particle.variable("PriceValue", &Price, INT);
```

```
  aika = EEPROM.get(addr, saveAika);
```

```
  //Jotta voidaan seurata aikaa jolloin automaatti on kytkenyt sähkön päälle ja säästetään
```

```
  loweredCostsCountInMinutes = EEPROM.get(6, loweredCostsCountInMinutes);
```

```
  Particle.variable("MinutesSavedEnergy", &loweredCostsCountInMinutes, INT);
```

```
  Particle.function("ResetMinutesSavedEnergy", &ResetSavedCosts);
```

```
minutesUsedLoweredCosts.start();

}

void loop()
{

    if(Price < value)
    {
        digitalWrite(heater, HIGH);
        status = 1;
    }

    else if (Price > value)
    {
        digitalWrite(heater, LOW);
        status = 0;
    }

}

}

//vasen arvo
int heaterControl(String priceValue)
{
    pos = priceValue.toInt();
    // Muutetaan int string muotoon ja lisätään numerot peräkkäin jonka jälkeen takaisin int muotoon.
    String posstring = String(pos);
    String positionstring = String(position);
    String price = posstring + positionstring;
```

```
    Price = price.toInt();
    return -1;

}
//oikean puolen arvo
int ledControl(String ndpriceValue)
{
    position = ndpriceValue;
    String posstring = String(pos);
    String positionstring = String(position);
    String price = posstring + positionstring;
    Price = price.toInt();
    return 1;

}
//Hinnan asetus
int setPrice(String compValue)
{
    value = compValue.toInt();
    return 1;

}
//Jotta voidaan seurata aikaa joka ollaan säästetty lämmityksessä. 4.0c/kWh on rajana pörssi
sähkön hinnalle jolloin lämmityssähköllä tulee halvemmaksi
void LoweredCosts(){
    if(Price <= 400){
        loweredCostsCountInMinutes += 1;
        EEPROM.put(6, loweredCostsCountInMinutes);
    }
}
```

**Liite 7. Fortumin hintatiedon välittävä ohjelma**

```
<!DOCTYPE html>
<html lang="en">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<head>
  <!--<link rel="stylesheet" href="fortumAjax6.css">-->

</head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <title>Fortumin sähkön hinnan seuranta ja ohjaus</title>

<body>
<div class="outputs" align="center">
<h3>Pörssisähkön hinta</h3>
Hinta juuri nyt:<br>
<span id="output1"></span>
<div style="display: inline;" id="comma">,</div>
<div style="display: inline;" id="output2"></div>c/kwh<br><br>
Päivitetty:<br>
<div id="output3"></div>
<span id="output4"></span><br><br>

Aseta haluttu hinnan raja-arvo ilman pilkkuja, jonka mukaan lämmitystä ohjataan sähköllä<br>

<input type="text" id="num" size="3" maxlength="4"><br>
<button id="Set" onclick="CalcValue()">Aseta arvo</button><br><br>
<h4>Tällä hetkellä asetettu arvo:</h4>
<span id="output5"></span>c/kwh<br><br>

</div>

<script>

window.setInterval (function){
```

```

var serviceUrl = 'https://fortum.heydaypro.com/tarkka/graph.php';
var method = $('#method').val();
$.ajax({
  type: method,
  url: serviceUrl
}).done(function (data) {

    var strFirst = data.split('>')[494];
    var output1 = strFirst.split('<')[0];
    var strNd = data.split('>')[496];
    var output2 = strNd.split('<')[0];
    var strRd = data.split('>')[503];
    var output3 = strRd.split('<')[0];

    $('#output1').text(output1);
    $('#output2').text(output2);
    $('#output3').text(output3);

  }).fail(function (jqXHR, textStatus, errorThrown) {
    $('#GridFrequency').text(jqXHR.responseText || textStatus);
  });
},5000);

var deviceId = "35*****3937";
var accessToken = "b**b568*****333";
var setFunc = "heater";
var setndFunc = "led";
var setValue = "setValue";

//Ensimmäinen numero (eli pilkun vasemman puoleinen) lähetetään particlelle 20s välein
window.setInterval (function(setValue) {

    var setValue = document.getEle-
mentById('output1').innerHTML;

    var requestURL =
"https://api.particle.io/v1/devices/" +deviceId + "/" + setFunc + "/";

```

```
$.post( requestURL, { params: setValue, access_token: accessToken });
}, 21000);

// toinen numero (eli pilkun oikean-
puoleinen) lähetetään partillelle 20s välein.
window.setInterval (function(){
    var setndValue = document.getElementById('output2').innerHTML;
    var requestURL = "https://api.particle.io/v1/devices/" +deviceId + "/" + setndFunc + "/";
    $.post( requestURL, { params: setndValue, access_token: accessToken });
}, 20000);

//Raja-arvon säätö textBoxila. Lähetetään particlelle painettaessa aseta arvo nappia.
function CalcValue(){
    var field=document.getElementById("num").value;
    sparkSetPrice(field);
}

function sparkSetPrice(field){
    document.getElementById('output5').innerHTML = field;
    var requestURL = "https://api.particle.io/v1/devices/" +deviceId + "/" + setValue + "/";
    $.post( requestURL, { params: field, access_token: accessToken });
}

</script>
</body>
</html>
```

### Liite 8. Mäntyharjun Photonilla ajettava ohjelma

```
#include "Particle.h"
SYSTEM_MODE(AUTOMATIC);
SYSTEM_THREAD(ENABLED);
#define WAIT_FOR_CONNECTION_SECONDS 60

// This #include statement was automatically added by the Particle IDE.
#include <TimeAlarms.h>

// This #include statement was automatically added by the Particle IDE.
#include <RCSwitch.h>

// This #include statement was automatically added by the Particle IDE.
#include <Adafruit_DHT.h>

//DHT parameters
#define DHTPIN2 2
#define DHTTYPE2 DHT21
#define DHTPIN3 3
#define DHTTYPE DHT21
#define DHTPIN3 4
#define DHTTYPE3 DHT21
// initialisation
int analogvalue;
int digitalvalue;
int commands = 0;
int pos = 30;
int humLimit2 = 30;
int degrees;
int heaterstatus = 0;
int heaterstatus2 = 0;
String strTemperature, strTemperature2, strTemperature3;
String wrappedTemperatures;
String strHumidity, strHumidity2, strHumidity3;
String wrappedHumidities;
```

```
String strHistoryTemperature="0", strHistoryTemperature2="0", strHistoryTemperature3="0",
strHistoryHumidity="0", strHistoryHumidity2="0", strHistoryHumidity3="0";
String wrappedHistory;

//inputs
int temperature, temperature2, temperature3;
int humidity;
int humidity2;
int humidity3;
//outputs
int heater = D5;
//int led = D7;
int WriteTemp;
int tempSum, tempSum2, tempSum3;
int tempSumCnt, tempSumCnt2, tempSumCnt3, tempSumCnt4; //sumcount 4 on makuuhuoneelle ja 3 keittiölle
int tempLimit = 1;
int humHysteresis = 5;

// DHT sensor
DHT dht(DHTPIN, DHTTYPE);
DHT dht2(DHTPIN2, DHTTYPE2);
DHT dht3(DHTPIN3, DHTTYPE3);

// Radio frequency 443Mhz initialisation
RCSwitch mySwitch = RCSwitch();

//Ajastin lähettää sen hetkisen ohjauksen uudestaan 15min välein
int TimerTime = 900000; //900000 = 15min
Timer repeatBedroomHeaterHigh(TimerTime, BedroomHeaterOn);
Timer repeatBedroomHeaterLow(TimerTime, BedroomHeaterOff);
Timer repeatKitchenHeaterHigh(TimerTime, KitchenHeaterOn);
Timer repeatKitchenHeaterLow(TimerTime, KitchenHeaterOff);
```

```
void setup()
{
  //Particle.connect();
  Particle.function("connectedHTMLS",connectedHTMLS);
  Particle.variable("NumberOfConnections", &NumberOfConnections,INT);
  //Asetusarvon muuttuja
  Particle.variable("humidityLimit",&pos,INT);
  Particle.variable("ndhumidityLimit",&humLimit2,INT);
  Particle.variable("state",&commands,INT);
  //Kerätään kaikki 3 int lämpötilamuuttujaa yhteen stringiin ja lähetetään particlelle
  Particle.variable("wrappedTemperatures", &wrappedTemperatures, STRING);
  //Kerätään kaikki 3 int kosteusmuuttujaa yhteen stringiin ja lähetetään particlelle
  Particle.variable("wrappedHumidities", &wrappedHumidities, STRING);

  //Kosteuden 1(MH) säätöfunktio.
  Particle.function("settemp",settemp);
  //Kosteuden 2 (K) säätöfunktio.
  Particle.function("setHum2",setHumidity2);
  //Lämmittimen 1 oloarvo
  Particle.variable("heaterstatus", &heaterstatus, INT);
  //Lämmittimen 2 oloarvo
  Particle.variable("heaterstatus2", &heaterstatus2, INT);
  //Lämmitin ohjaus
  Particle.function("heater",heaterControl);

  pinMode(heater, OUTPUT);
  pinMode(D7, OUTPUT);
  pinMode(D0, OUTPUT);

  Time.zone(1);
  Alarm.timerRepeat(21600, Repeats); //Every 6 hours = 21600s

  // Transmitter is connected to Arduino Pin #D6
  mySwitch.enableTransmit(D6);
```

```
// Start DHT sensor
dht.begin();
dht2.begin();
dht3.begin();

// Set outlets to 0
mySwitch.send("0001010100010101010100"); // Pois CH 1
delay(200);
mySwitch.send("0100010101000101010100"); // Pois CH 2

}

void Restart(){

    System.reset();
}

//Historian muuttujiin haetaan uusi arvo 6 tunnin välein
void Repeats(){

    Particle.publish("Temperatures",strHistoryTemperature + "," + strHistoryTempera-
ture2+"," +strHistoryTemperature3);
    delay(1000);
    Particle.publish("Humidities",strHumidity+"," +strHumidity2+"," +strHumidity3);

}

//Joka kierroksella edellistä arvoa verrataan uuteen ja jos poikkeamaa on ylös tai alaspäin yli 3
yksikköä arvoa ei kirjata historiaan, vaan edellinen kelvollisempi arvo.
int prevTemp, prevTemp2, prevTemp3;
int prevHum, prevHum2, prevHum3;
bool firstRound = false;
```

```
int TempAndHum(){

    if(temperature - prevTemp <= 3 && temperature - prevTemp >= -3 && -5 < temperature <
25){
        strHistoryTemperature = temperature;
    }
    prevTemp = temperature;

    if(temperature2 - prevTemp2 <= 3 && temperature2 - prevTemp2 >= -3 && -5 < tempera-
ture2 < 25){
        strHistoryTemperature2 = temperature2;
    }
    prevTemp2 = temperature2;

    if(temperature3 - prevTemp3 <= 3 && temperature3 - prevTemp3 >= -3 && -5 < tempera-
ture3 < 25 && -5 < strHistoryTemperature3.toInt() < 20){
        strHistoryTemperature3 = String(temperature3);
    }
    prevTemp3 = temperature3;

    if(humidity - prevHum <= 3 && humidity - prevHum >= -3 && 10 < humidity < 100){
        strHumidity = humidity;
    }
    prevHum = humidity;

    if(humidity2 - prevHum2 <= 3 && humidity2 - prevHum2 >= -3 && 10 < humidity2 < 100){
        strHumidity2 = humidity2;
    }
    prevHum2 = humidity2;

    if(humidity3 - prevHum3 <= 3 && humidity3 - prevHum3 >= -3 && 10 < humidity3 < 100){
        strHumidity3 = humidity3;
    }
    prevHum3 = humidity3;
```

```
    temperature = dht.getTempCelcius(); //Makuuhuone
    temperature2 = dht2.getTempCelcius(); //Keittiö
    temperature3 = dht3.getTempCelcius(); //ulko
    humidity = dht.getHumidity(); //Makuuhuone
    humidity2 = dht2.getHumidity(); //Keittiö
    humidity3 = dht3.getHumidity(); //ulko

}
int controlPrevHum, controlPrevHum2;
int controlPrevTemp, controlPrevTemp2, controlPrevTemp3;
int twoBehindTemp, twoBehindTemp2, twoBehindTemp3;
bool tempHeating = false;
bool tempHeating2 = false;
int tempSumCntCompare = 20; //Aseta kaikkiin muuttuja vertailu arvoksi
int humSum, humSum2;
int humSumCnt, humSumCnt2;

void loop()
{

    //Haetaan lämpötilat ja kosteudet
    TempAndHum();

    Alarm.delay(100);

    //Kaikki kolme lämpötilaa lähetetään yhdessä string muuttujassa
    strTemperature = String(temperature);
    strTemperature2 = String(temperature2);
    strTemperature3 = String(temperature3);
    wrappedTemperatures = strTemperature + "," + strTemperature2 + "," + strTempera-
    ture3+","+String(pos)+","+String(humLimit2)+","+String(commands);

    //Kaikki kolme kosteusarvoa lähetetään yhdessä string muuttujassa.
    strHumidity = String(humidity);
    strHumidity2 = String(humidity2);
```

```
strHumidity3 = String(humidity3);  
wrappedHumidities = strHumidity + "," + strHumidity2 + "," + strHumidity3;
```

```
//Heater controlling by automation  
if (commands == 1)  
{  
  
    //Tehdään häiriönpoisto vertailu kaikille kolmelle lämpötilalle  
    tempSum = temperature - twoBehindTemp;  
    tempSum2 = temperature2 - twoBehindTemp2;  
    tempSum3 = temperature3 - twoBehindTemp3;  
    humSum = humidity - controlPrevHum;  
    humSum2 = humidity2 - controlPrevHum2;  
    //Temp 1  
    if(-2 < tempSum < 2){  
        tempSumCnt += 1;  
    }  
    if (3 < tempSum || -3 > tempSum){  
        tempSumCnt = 0;  
    }  
  
    //Temp 2  
    if(-2 < tempSum2 < 2){  
        tempSumCnt2 += 1;  
    }  
    if (3 <= tempSum2 || tempSum2 <= -3){  
        tempSumCnt2 = 0;  
    }  
  
    //Temp 3  
    if(-2 < tempSum3 < 2){  
        tempSumCnt3 += 1;  
        tempSumCnt4 += 1;  
    }  
    if (3 <= tempSum3 || tempSum3 <= -3){
```

```
tempSumCnt3 = 0;
tempSumCnt4 = 0;
}
//Hum 1
if(-2 < humSum < 2){
    humSumCnt += 1;
}
if(3 < humSum || -3 > humSum){
    humSumCnt = 0;
}
//Hum 2
if(-2 < humSum2 < 2){
    humSumCnt2 += 1;
}
if(3 < humSum2 || -3 > humSum2){
    humSumCnt2 = 0;
}

//Lämpötilalla ohjausta //////////////////////////////////////
//Jos makuuhuoneen ja ulkolämpötilan ero on vähemmän kuin 3 astetta lämmitys kytke-
tään päälle.
//controlPrevTempillä pyritään vähentämään häiriöstä aiheutuvia säätöjä
//Kun lämpötila yli 0
if (temperature3 >= -15 && temperature3 < 20 && tempSumCnt >= tempSumCntCompare &&
tempSumCnt4 >= tempSumCntCompare){
    //Lämmitys päälle
    if(temperature - temperature3 <= tempLimit && tempHeating == false && temperature
>= 1 && temperature < 14 || (temperature <= 2 && tempHeating == false && temperature >= 1){
        repeatBedroomHeaterLow.stop();
        mySwitch.send("0001010100010101010101"); //Päälle CH 1
        digitalWrite(heater, HIGH);
        tempHeating = true;
        heaterstatus=1;
        repeatBedroomHeaterHigh.start();
        delay(1000);
    }
    //Lämmitys pois
```

```
    else if(temperature - temperature3 - 2 > tempLimit && tempHeating == true && temper-
ature > 5 && temperature < 17){
        repeatBedroomHeaterHigh.stop();
        mySwitch.send("000101010001010101010100"); // Pois CH 1
        digitalWrite(heater, LOW);
        tempHeating = false;
        heaterstatus=0;
        repeatBedroomHeaterLow.start();
        delay(1000);
    }
    tempSumCnt = 0;
    tempSumCnt4 = 0;
}

twoBehindTemp = controlPrevTemp;
controlPrevTemp = temperature;

if(tempHeating == false){
    // Makuuhoneen kosteuden säätö. +6 on asetettu hystereesi. Kun kosteus nousee yli asetetun
raja-arvon lämmitin kytkeytyy päälle. Ja kun kosteus tulee alla asetetun rajan +3 se kytkeytyy
pois.
    //Kosteusarvon täytyy olla järkevissä rajoissa jotta säätö toteutetaan.
    if(humidity >= pos && heaterstatus == 0 && humidity > 10 && humidity < 80 && humSumCnt
>= tempSumCntCompare)
    {
        //Jotta saadaan väärin mittatulosten perusteella tehdyt säädöt karsittua
        // if(humidity - controlPrevHum <= 2 && humidity - controlPrevHum >= -2){
        repeatBedroomHeaterLow.stop();
        mySwitch.send("000101010001010101010101"); //Päälle CH 1
        digitalWrite(heater, HIGH);
        heaterstatus=1;
        repeatBedroomHeaterHigh.start();
        delay(1000);
        // }
        humSumCnt = 0;
    }
}
```

```
else if (humidity + humHysteresis <= pos && heaterstatus == 1 && humidity > 10 && humidity  
< 80 && humSumCnt >= tempSumCntCompare)  
{  
    // if(humidity - controlPrevHum <= 2 && humidity - controlPrevHum >= -2){  
        repeatBedroomHeaterHigh.stop();  
        mySwitch.send("000101010001010101010100"); // Pois CH 1  
        digitalWrite(heater, LOW);  
        heaterstatus=0;  
        repeatBedroomHeaterLow.start();  
        delay(1000);  
    // }  
    humSumCnt = 0;  
}  
}  
  
//Määritellään vertailun jälkeen edellisen kierroksen arvo nykyiseksi  
controlPrevHum = humidity;
```

```
//Lämpötilan perusteella ohjausta  
////////////////////////////////////  
if (temperature3 >= -15 && temperature3 < 20 && tempSumCnt2 >= tempSumCntCompare &&  
tempSumCnt3 >= tempSumCntCompare){  
    //Lämmitys päälle  
    if(temperature2 - temperature3 <= tempLimit && tempHeating2 == false && tempera-  
ture2 >= 1 && temperature2 < 14 || temperature2 <= 2 && tempHeating2 == false && tempera-  
ture2 >= 1){  
        repeatKitchenHeaterLow.stop();  
        mySwitch.send("010001010100010101010101"); // Päälle CH 2  
        //digitalWrite(heater, HIGH);  
        tempHeating2 = true;  
        heaterstatus2=1;  
        repeatKitchenHeaterHigh.start();  
        delay(1000);  
    }  
}
```

```
    }  
    //Lämmitys pois  
    else if(temperature2 - temperature3 - 2 > tempLimit && tempHeating2 == true && tem-  
perature2 > 5 && temperature2 < 17){  
        repeatKitchenHeaterHigh.stop();  
        mySwitch.send("010001010100010101010100"); // Pois CH 2  
        //digitalWrite(heater, LOW);  
        tempHeating2 = false;  
        heaterstatus2=0;  
        repeatKitchenHeaterLow.start();  
        delay(1000);  
    }  
    tempSumCnt2 = 0;  
    tempSumCnt3 = 0;  
  
}
```

```
twoBehindTemp2 = controlPrevTemp2;  
controlPrevTemp2 = temperature2;
```

```
if(tempHeating2 == false){  
    //Keittiön kosteuden säätö. +3 on asetettu hystereesi. Kun kosteus nousee yli asetetun raja-  
arvon lämmitin kytkeytyy päälle. Ja kun kosteus tulee alla asetetun rajan +3 se kytkeytyy pois.  
    //Kosteusarvon täytyy olla järkevissä rajoissa, jotta säätö toteutetaan  
    if(humidity2 >= humLimit2 && heaterstatus2 == 0 && humidity2 > 10 && humidity2 <  
80 && humSumCnt2 >= tempSumCntCompare)  
    {  
        // if(humidity2 - controlPrevHum2 <= 2 && humidity2 - controlPrevHum2 >= -2){  
        repeatKitchenHeaterLow.stop();  
        mySwitch.send("010001010100010101010101"); // Päälle CH 2  
        //digitalWrite(heater, HIGH);  
        heaterstatus2=1;  
        repeatKitchenHeaterHigh.start();  
        delay(1000);  
        // }  
        humSumCnt2 = 0;
```

```
    }  
  
    else if (humidity2 + humHysteresis <= humLimit2 && heaterstatus2 == 1 && humidity2 >  
10 && humidity2 < 80 && humSumCnt2 >= tempSumCntCompare)  
    {  
        // if(humidity2 - controlPrevHum2 <= 2 && humidity2 - controlPrevHum2 >= -2){  
        repeatKitchenHeaterHigh.stop();  
        mySwitch.send("010001010100010101010100"); // Pois CH 2  
        //digitalWrite(heater, LOW);  
        heaterstatus2=0;  
        repeatKitchenHeaterLow.start();  
        delay(1000);  
        // }  
        humSumCnt2 = 0;  
    }  
}  
  
controlPrevHum2 = humidity2;
```

```
twoBehindTemp3 = controlPrevTemp3;  
controlPrevTemp3 = temperature3;  
}
```

```
delay(1000);
```

```
}
```

```
int settemp(String tempValue)  
{  
    pos = tempValue.toInt();  
    return -1;  
}
```

```
int setHumidity2(String humidityValue2)
{
    humLimit2 = humidityValue2.toInt();
    return -1;
}
// Heater control by manual
int heaterControl(String command)
{

    repeatBedroomHeaterLow.stop();
    repeatBedroomHeaterHigh.stop();
    repeatKitchenHeaterLow.stop();
    repeatKitchenHeaterHigh.stop();
    if (command=="1" && heaterstatus == 0) {
        mySwitch.send("000101010001010101010101"); //Päälle CH 1
        heaterstatus=1;
        commands = 0;
    }

    else if (command=="0" && heaterstatus == 1) {
        mySwitch.send("000101010001010101010100"); // Pois CH 1
        heaterstatus=0;
        commands = 0;
    }

    else if (command=="2")
    {
        commands = 1;
    }

    else if (command=="3" && heaterstatus2 == 0)
    {
        mySwitch.send("010001010100010101010101"); // Päälle CH 2
        heaterstatus2 = 1;
        commands = 0;
    }
}
```

```
else if (command=="4" && heaterstatus2 == 1)
{
    mySwitch.send("010001010100010101010100"); // Pois CH 2
    heaterstatus2 = 0;
    commands = 0;
}

}

//Automaattiohjauksen käskyt toistetaan 15min välein jotta varmistutaan ohjauksen meneminen
päälle
int BedroomHeaterOn(){
    mySwitch.send("000101010001010101010101"); //Päälle CH 1
    heaterstatus=1;
    delay(500);
}
int BedroomHeaterOff(){
    mySwitch.send("000101010001010101010100"); // Pois CH 1
    heaterstatus=0;
    delay(500);
}
int KitchenHeaterOn(){
    mySwitch.send("010001010100010101010101"); // Päälle CH 2
    heaterstatus2=1;
    delay(500);
}
int KitchenHeaterOff(){
    mySwitch.send("010001010100010101010100"); // Pois CH 2
    heaterstatus2=0;
    delay(500);
}
int remoteRestart(String values){
    System.reset();
}
```

**Liite 9. Käyttöliittymän HTML koodi**

```
<!DOCTYPE HTML>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
<head>
<link rel="stylesheet" href="HumidityAndInverterControl.css">
</head>

<body background="lightBlueRotated2.jpg">

<div id="Page1">
<div class="hideMany">

<!--Kosteus- ja lämpötila tekstien luokka, näkymä ja AIKA-->
<div class="Time" align="center">Time:
  <br>
  <span id="tstamp"></span><br><br><br>
  <div class="loader"></div>
</div><br>

<h1 id="BRC" style="font-size: 25px;">Bedroom Control:</h1>

<div id="BedroomRH">RH:</div>
<div id="BedroomT">Temperature:</div>

<!--Kosteus- ja lämpötila arvojen luokka ja näkymä-->
<span id="hum"></span>
  <span id="temp"></span>
  <br><br>

  <div id="pSB" style="font-size: 20px;" style="margin-top: -10px;">Set Bedroom RH</div>

  <span id="currentlySetBedValue"></span>

<!--Lämpötilan asetusarvo ja sen muotoilu-->
```





```
<option value="75">
<option value="80">
</datalist><br>

<button class="btn heater2On" style="z-index: 2;" onclick="Control(3);askState2()">CH2
ON</button>
<button class="btn heater2Off" onclick="Control(4);askState2()">CH2 OFF</button>

<div id="container10">
  <div id="highlight10"></div>
</div>
<div id="container20">
  <div id="highlight20"></div>
</div>

  <span id="kaynnissa2"></span>
  <span id="kaynnissa3"></span>

</div>
<br><br><br><br><br>

<h1 id="Ocss">OutSide</h1>

<div class="hideOutsides">
<div id="ORH">RH:</div>
<div id="OT">Temperature:</div>

<span id="hum3"></span>
  <span id="temp3"></span><br>

  </div>
</div>

<!-- Painonappit-->

<div align="center">
```

```
<div class="hideAuto">
  <button class="automatic" onclick="Control(2);askState();askState2()">AUTO</button><br>

  <p id="saatimenTila">Control mode:</p>

  <span id="AutoState"></span>
</div>

  <span id="output10" style="position: fixed; font-size: 50px; color: red; margin-top: -350px; margin-left: -350px;"></span><br>
  <span id="output11" style="position: fixed; font-size: 50px; color: red; margin-top: -320px; margin-left: -350px;"></span>
  <span id="output12"></span>

</div>

<div class="hideHistories">
<!--Historian attribuutit-->
<canvas id="Canvas" width="600" height="300" style="border:1px solid #000000; margin-left: 33%;"></canvas>
<canvas id="Canvas2" width="600" height="300" style="border:1px solid #000000; margin-left: 33%;"></canvas>
<canvas id="Canvas3" width="70" height="40" style="border:1px solid #000000; margin-left: 60.6%;"></canvas>
<canvas id="Canvas4" width="70" height="40" style="border:1px solid #000000; margin-left: 60.6%;"></canvas>
<canvas id="Canvas5" width="30" height="300" style="margin-left: 31.7%;"></canvas>
<canvas id="Canvas6" width="30" height="300" style="margin-left: 31.7%;"></canvas>

</div>

<a href="#" style="color:cyan; background: transparent; padding: 10px; font-size: 35px; margin-left: 40px; onclick=" return show('Page2','Page1');">History View</a>

<a href="#" style="font-size: 35px; color: cyan; background: transparent;" id="ShowInverter" onclick="showPage3();RunInverterSide()">Solar Charger</a>
```

```
</div>
```

```
<div id="Page2" style="display:none">
  <div align="center">
    <h2>Temperatures:</h2>
    <canvas id="tempCanvas" width="1905" height="350" style="border:1px solid
#000000;"></canvas><br>
    <h2>Humidities:</h2>
    <canvas id="humCanvas" width="1905" height="350" style="border:1px solid
#000000;"></canvas><br>
    <button onclick="myScope(56)">Two Weeks</button>
    <button onclick="myScope(120)">Month</button>
```

```
</div>
```

```
  <div align="center">
    <a style="color:cyan; background: transparent; padding: 10px; font-size: 35px;margin-left: -
100px; position: fixed;" href="#" onclick=" return show('Page1','Page2');">Home Page</a></div>
  </div>
```

```
<!-- Page 3 is for inverter parameters!-->
```

```
<div id="Page3" style="display: none">

  <div align="center">

    <h2 style="">Energy from panels till this day:</h2>
    <textarea id="KwhText" style="text-align: center; background: transparent; height: 36px; width:
160px;font-weight:bolder; font-size: 30px;"></textarea><p style="display: inline; font-size: 35px;
margin-left: 15px;">Kwh</p><br><br>
    <button id="Kwh" style="margin-left: -70px; margin-right: 30px;" onclick="GetKwh()">Refresh
Kwh</button>
    <button id="History" onclick="GetHistory()">Refresh History</button>
  </div>
```

```
<div id="GeneralStatusValues" style="margin-top: -150px;">
<!--Grid values displayed in textbox-->
<h3 style="display: inline; margin-right: 40px">Grid voltage:</h3> <h3 style="display: in-
line;">Grid frequency:</h3><br>
<textarea id="GridVoltageText" style="text-align: center;margin-left: 20px; background: trans-
parent; height: 22px; width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></tex-
tarea><p style="display: inline; margin-left: 5px;font-size: 25px">V</p> <textarea id="GridFre-
quencyText" style="text-align: center;margin-left: 30px; background: transparent; height: 22px;
width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></textarea><p style="display: in-
line; margin-left:5px;font-size: 25px ">Hz</p><br><br>

<!--InverterOutput displayed in textbox-->
<h3 style="display: inline; margin-right: 18px">Output voltage:</h3> <h3 style="display: in-
line;margin-right: 18px">Output frequency:</h3><h3 style="display: inline;">Output load per-
cent:</h3><br>
<textarea id="OutputVoltageText" style="text-align: center;margin-left: 20px; background:
transparent; height: 22px; width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></tex-
tarea><p style="display: inline; margin-left: 5px;font-size: 25px">V</p> <textarea id="OutputFre-
quencyText" style="text-align: center;margin-left: 30px; background: transparent; height: 22px;
width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></textarea><p style="display: in-
line; margin-left:5px;font-size: 25px ">Hz</p>
<textarea id="OutputLoadPerText" style="text-align: center;margin-left: 50px; background:
transparent; height: 22px; width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></tex-
tarea><p style="display: inline; margin-left:5px;font-size: 25px ">%</p><br><br>

<h3 style="display: inline; margin-right: 18px">Battery voltage:</h3> <h3 style="display: in-
line;">Battery input current:</h3><br>
<textarea id="BatteryVoltageText" style="text-align: center;margin-left: 20px; background:
transparent; height: 22px; width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></tex-
tarea><p style="display: inline; margin-left: 5px;font-size: 25px">V</p> <textarea id="Bat-
teryCharCurrentText" style="text-align: center;margin-left: 30px; background: transparent; height:
22px; width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></textarea><p style="dis-
play: inline; margin-left:5px;font-size: 25px ">A</p><br><br>

<h3 style="display: inline; margin-right: 60px">Pv voltage:</h3> <h3 style="display: inline;">Pv
charging current:</h3><br>
```

```
<textarea id="PvVoltageText" style="text-align: center;margin-left: 20px; background: transparent; height: 22px; width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></textarea><p style="display: inline; margin-left: 5px;font-size: 25px">V</p> <textarea id="PvCharCurrentText" style="text-align: center;margin-left: 30px; background: transparent; height: 22px; width: 75px;font-weight:bolder; font-size: 18px; display: inline;"></textarea><p style="display: inline; margin-left:5px;font-size: 25px ">A</p><br><br>
```

```
<h1 style=" margin-left:150px; ">History:</h1>
```

```
<p style="font-size: 20px; display: inline; margin-right: 170px; margin-left: 80px;">Date:</p><p style="font-size: 20px; display: inline;">Energy(Kwh):</p><br>
```

```
<textarea id="HistoryText" style="text-align: left;margin-left: 30px; background: transparent; height: 400px; width: 400px;font-weight:bolder; font-size: 18px; border: 7px solid #A1A1A1;"></textarea>
```

```
</div>
```

```
<h1 style=" position: fixed; margin-left:1350px; margin-top: -700px">Alarms and faults:</h1>
```

```
<textarea id="AlarmsAndFaultsText" style="width: 600px; height: 600px; position: fixed; margin-top: -600px; margin-left: 1250px; font-size: 50px "></textarea>
```

```
<div align="center">
```

```
<button style="background: transparent; border: 7px solid #A1A1A1; color: cyan; font-size: 30px; width: 400px; height: 100px;" id="ShowHome" onclick="showHome()">Home Page</button>
```

```
</div>
```

```
</div>
```

```
<span style="margin-left: 500px;" id="testing"></span>
```

**Liite 10. Käyttöliittymän JavaScript koodi (osa)**

```
var DrawHT2 = 0;
var DrawHT1 = 0;
var DrawHT3 = 0;

var firstTempNumber;
var secondTempNumber;
var thirdTempNumber;
//Lämpötilojen ja säätimen tilan haku
function FC7_Lämpötilat(){
window.setInterval(function() {

    requestURL = "https://api.particle.io/v1/devices/" + deviceId + "/" + wrappedTemperatures +
"/?access_token=" + accessToken;
    $.getJSON(requestURL, function(json) {

        var json = json.result;
        var split = json.split(",");
        firstTempNumber = split[0];
        secondTempNumber = split[1];
        thirdTempNumber = split[2];
        var number7 = split[3];
        var number8 = split[4];
        var number9 = split[5];

        document.getElementById("currentlySetBedValue").innerHTML = number7;
        document.getElementById("currentlySetKitchenValue").in-
nerHTML = number8;

        document.getElementById("temp").innerHTML = firstTempNumber+ "&deg; C";
        DrawHT2 = firstTempNumber;
```

```
document.getElementById("temp2").innerHTML = secondTempNumber+ "&deg;  
C";  
  
DrawHT1 = secondTempNumber;  
  
document.getElementById("temp3").innerHTML = thirdTempNumber+ "&deg; C";  
DrawHT3 = thirdTempNumber;
```

```
var AutomaticState;  
if(number9==0)  
    {AutomaticState="Manual";}   
else if(number9==1)  
    {AutomaticState="Auto";}   
  
document.getElementById('AutoState').innerHTML = AutomaticState;  
  
    });  
  
}, 3000);  
}
```

```
var firstNumber = 0;  
var ndNumber = 0;  
var rdHumNumber = 0;  
var firstRound = 0;  
  
//Haetaan kaikki kolme kosteusarvoa yhdellä stringillä.  
function FC1_KosteusArvojenHaku(){  
    window.setInterval(function() {  
  
        requestURL = "https://api.particle.io/v1/devices/" + deviceId + "/" + wrappedHumidities +  
        "?access_token=" + accessToken;
```

```
$.getJSON(requestURL, function(json) {

    var prevFirstNumber = firstNumber;
    var prevndNumber = ndNumber;
    var prevrdNumber = rdHumNumber;
    var json = json.result;
    var split = json.split(",");
    firstNumber = split[0];
    ndNumber = split[1];
    rdHumNumber = split[2];

    document.getElementById("hum").innerHTML = firstNumber+ "%";;
    document.getElementById("hum2").innerHTML = ndNumber+ "%"; ;
    document.getElementById("hum3").innerHTML = rdHumNumber + "%";;

    $("#showCanvas2").show();

    $(".loader").hide();
});

}, 3500);
}
```

//Lämpötilojen piirto

```
var canvas = document.getElementById('Canvas');
var ctx = canvas.getContext('2d');
ctx.translate(0, canvas.height);
ctx.scale(1, -1);
```

//Keittiön kuvaajan funktiot

```
var x = 4;
var dx = 2;
var lastX, lastY;
```

//Makuuhuoneen kuvaajan funktiot

```
var x2 = 4;
```

```
var previousX, previousY;

//Ulkolämpötilan kuvaajan funktiot
var x3 = 4;
var beforeX, beforeY;

window.setInterval(function() {
    if(DrawHT1 < 35 && DrawHT1 > -10){
//Keittiön lämpötilan piirto
    var y = (300/45) * (DrawHT1-(-10));
    }

    ctx.beginPath();
    ctx.moveTo(lastX,lastY);
        ctx.lineTo(x, y);
        ctx.strokeStyle = 'blue';
        ctx.stroke();

        lastX = x; lastY=y;

        x+=dx;

//Makuuhuoneen lämpötilan piirto.
    if(DrawHT2 < 35 && DrawHT2 > -10){
        var y2 = (300/45) * (DrawHT2-(-10)) ;
    }

        ctx.beginPath();
    ctx.moveTo(previousX,previousY);
        ctx.lineTo(x2,y2);
        ctx.strokeStyle = 'red';
        ctx.stroke();
```

```
previousX = x2; previousY=y2;

x2+=dx;

//Ulkolämpötilan piirto

if(DrawHT3 < 35 && DrawHT3 > -10){
var y3 = (300/45) * (DrawHT3-(-10));
}

ctx.beginPath();
ctx.moveTo(beforeX,beforeY);
ctx.lineTo(x3,y3);
ctx.strokeStyle = 'green';
ctx.lineWidth = 2;
ctx.stroke();

beforeX = x3; beforeY=y3;

x3+=dx;

//Jos piirtokankaan reuna saavutetaan aloitetaan
piirto alusta

if(x > canvas.width || x2> canvas.width){
ctx.clearRect(0, 0, innerWidth, innerHeight);
x=4;
lastX=4;
lastY=4;

x2=4;
previousX=4;
previousY=4;

x3=4;
beforeX=4;
```

```

        beforeY=4;
    }
}, 3000);

function GetHistory() {
    var serviceUrl = 'https:// example.conveyor.cloud/api/values/9';
    var method = $('#method').val();

    $.ajax({
        type: method,
        url: serviceUrl
    }).done(function (data) {
        var splittedHistory = data.split(' ');

        // To split data nicely to the text box
        $('#HistoryText').text(splittedHistory[0] + "
        " + splittedHistory[1]
        + "\n" + splittedHistory[2] + "
        " + splittedHistory[3] + "\n"
        + splittedHistory[4] + "
        " + splittedHistory[5] + "\n"
        + splittedHistory[6] + "
        " + splittedHistory[7] + "\n"
        + splittedHistory[8] + "
        " + splittedHistory[9] + "\n"
        + splittedHistory[10] + "
        " + splittedHistory[11] + "\n"
        + splittedHistory[12] + "
        " + splittedHistory[13] + "\n"
        + splittedHistory[14] + "
        " + splittedHistory[15] + "\n"
        + splittedHistory[16] + "
        " + splittedHistory[17] + "\n"
        + splittedHistory[18] + "
        " + splittedHistory[19] + "\n"
        + splittedHistory[20] + "
        " + splittedHistory[21] + "\n"
        + splittedHistory[22] + "
        " + splittedHistory[23] + "\n"
        + splittedHistory[24] + "
        " + splittedHistory[25] + "\n"
        + splittedHistory[26] + "
        " + splittedHistory[27] + "\n"
        + splittedHistory[28] + "
        " + splittedHistory[29] + "\n");

    }).fail(function (jqXHR, textStatus, errorThrown) {
        $('#HistoryText').text(jqXHR.responseText || textStatus);
    });
});

```