Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Matti Tampio

IoT-laitteen rakentaminen osaksi kotiautomaatiota

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Konetekniikka

Insinöörityö

7.1.2020

Tekijä Otsikko Sivumäärä Aika	Matti Tampio IoT-laitteen rakentaminen osaksi kotiautomaatiota 66 sivua + 7 liitettä 7.1.2020
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Konetekniikka
Ammatillinen pääaine	Koneensuunnittelu
Ohjaaja	Lehtori Ari Koistinen

Insinöörityön tavoitteena oli rakentaa oma IoT-laite osaksi kotiautomaatiota ja kirjoittaa opastyylinen kokonaisuus järjestelmän rakentamisesta. Työtä taustoitettiin käsittelemällä kotiautomaation nykytilannetta ja historiaa. Työ antaa hyödyllistä tietoa kotiautomaatiosta kiinnostuneille ja opastusta sen käyttöönottamiseksi kotona.

IoT-laitteen rakentaminen toteutettiin kokoamalla Raspberry Pi -tietokoneen ympärille sensoreiden ja komponenttien kokonaisuus, jota hallitaan Grafana-, Apple HomeKit- ja Homebridge-sovelluksilla. Laitteen rakentaminen on esitelty siten, että eri työvaiheita voi käyttää oppaana kyseisen järjestelmän rakentamiseen, tai vaihtoehtoisesti lukija ohjataan tutustumaan lähteeseen, josta tarvittavat tiedot löytyvät.

Esimerkkityössä rakennetulla IoT-laitteella voidaan helpottaa kasvien ylläpitoa, kun esimerkiksi kasvualustan kastelu voidaan hoitaa etäohjauksella tai se voidaan asettaa toimimaan automaattisesti ajastuksella tai kasvualustan kosteusarvon poiketessa asetetusta raja-arvosta. Laitteella seurataan kasveille tärkeiden muuttujien kuten lämpötilan, ilmankosteuden, kasvualustan kosteuden, hiilidioksidin, lannoitteen- ja valonvoimakkuuden muutoksia reaaliajassa. Lisäksi kaikki arvot tallennetaan InfluxDB-tietokantaan, josta kerättyjä arvoja analysoidaan Grafana-ohjelman käyttöliittymällä. Rakennettua järjestelmää voidaan käyttää myös kotiautomaation keskusyksikkönä, johon voidaan liittää uusia laitteita ja luoda näiden välille haluttua automaatiota ja hälytystoimintoja. Järjestelmää voidaan hallita sekä Android-laitteilla että Applen iOS-laitteilla.

Insinöörityön lopputuloksena saatiin edullinen ja toimiva kotiautomaatiojärjestelmä, jolla voidaan ohjata kodin sähkölaitteita ja valvoa sensoriarvoja. Järjestelmä on helposti laajennettavissa uusilla sensoreilla, automaatiotoiminnoilla ja yhteystekniikoilla, kuten Zigbee-yhteystekniikalla. Järjestelmä on myös siirrettävissä uudempiin Raspberry Pi -tietokoneisiin, kunhan LabVIEW-ohjelma jätetään pois ohjelmakokoonpanosta.

Avainsanat	Raspberry Pi, IoT-laite, kotiautomaatio, etäohjaus, tiedonkeruulaite, avoimen lähdekoodin ohjelmat

Author Title Number of Pages Date	Matti Tampio Building an IoT Device for Home Automation 66 pages + 7 appendices 7 January 2020
Degree	Bachelor of Engineering
Degree Programme	Mechanical Engineering
Professional Major	Machine Design
Instructor	Ari Koistinen, Senior Lecturer

The aim of this thesis was to build an IoT device for home automation and write a guide, how to construct the system. The thesis introduces the current state of home automation and examines its history. It also provides useful information for those interested in home automation and guidance on how to implement the system at home

The construction of the IoT device was accomplished by assembling a set of sensors and components around the Raspberry Pi computer. Grafana, Apple HomeKit and Homebridge applications provide user interfaces for the system. The construction of the device is presented in such a way that the various stages of work can be used as a guide or, alternatively, the reader is guided to the source where the necessary information can be found.

In this thesis, the IoT device was set to help plant maintenance. For example, plant watering can be remotely controlled or can be set to operate automatically by timing or when the soil moisture value deviates from the set limit. The device monitors real-time changes in variables important to plants, such as temperature, humidity, soil moisture, carbon dioxide, fertilizer, and luminous intensity. In addition, all values are stored in the InfluxDB database, from where the collected values can be analyzed using the Grafana user interface. The built IoT device can also be used as a central hub for home automation to which new devices, sensors, alarm functions and desired automation can be added. The system can be controlled with android and iOS devices.

As a result of this thesis, an inexpensive and functional home automation system was built that can control electrical appliances and monitor sensor values. The system is easily expandable with new sensors, automation functions and communication protocols such as Zigbee. The system is also transferable to newer Raspberry Pi computers, as long as the LabVIEW program is excluded from the program assembly.

Keywords	Raspberry Pi, IoT device, home automation, remote control, data logger, open source software

Sisällys

Lyhenteet

1 Johdanto 1

2 Kotiautomaatio 2

2.1 Kotiautomaation historiaa 2
2.2 Yleisimmät käyttökohteet 4
2.2.1 Valaistus 4
2.2.2 Lämmityksen ja energian käytön hallinta 5
2.2.3 Turvallisuus 6
2.2.4 Viihdejärjestelmät 6
2.3 Yleistä kotiautomaatiojärjestelmän käyttöliittymän valinnasta 7
2.4 Järjestelmien protokollat 8

3 Esimerkkitoteutus kotiautomaatioon lisättävästä IoT-laitteesta 10

3.1 Esitietovaatimukset asennusohjeiden seuraamiseen 10
3.2 Ohjelmaversiot ja tiedonsiirto 12
3.3 Laitteisto 14
3.3.1 Raspberry Pi 3 B 14
3.3.2 RTL-SDR-radiovastaanotin 15
3.3.3 2-kanavainen relekortti 16
3.3.4 Raspberry Pi -kameramoduuli 17
3.3.5 Gardena-lomakastelusarja 18
3.4 Kotelointi 19
3.4.1 Suunnittelu 19
3.4.2 Mallinnus ja 3D-tulostus 20
3.5 Sensorit 22
3.5.1 Hiilidioksidimittari 9050 22
3.5.2 Lämpö- ja kosteussensori AcuRite 06002M 23
3.5.3 HHCC Flower Care -sensori 24

3.6 Ohjelmat laitteille ja sensoreille 25
3.6.1 Raspberry Pi 3 B 25
3.6.2 RTL_443 25
3.6.3 Mi Flora Plant Sensor MQTT Client/Daemon 27
3.6.4 CO2Meter 29
3.6.5 MotionEyeOS 30
3.6.6 Dyfi-update 31
3.7 Ohjelmat datan käsittelyyn 33
3.7.1 LabVIEW 33
3.7.2 Node-RED 36
3.7.3 InfluxDB 40
3.8 Ohjelmat käyttöliittymään 43
3.8.1 Grafana 43
3.8.2 Apple HomeKit 49
3.8.3 Homebridge 51

4 Lopputulos 54

5 Yhteenveto 56

5.1 Pohdintaa rakennetusta järjestelmästä 56
5.2 Pohdintaa tämänhetkisistä kotiautomaatiojärjestelmistä 57
5.3 Järjestelmän parannussuunnitelmat 61
5.3.1 Parannukset laitteistolle 61
5.3.2 Parannukset ohjelmille ja tietoturvalle 61

Lähteet 63

Liitteet

Liite 1. LabVIEW-ohjelman JSON-syöte Grafanaa varten

Liite 2. LabVIEW-ohjelman JSON-syöte kaikilla toiminnoilla

Liite 3. Asetukset InfluxDB-tietokannan yhdistämisestä Grafanaan

Liite 4. Javascript-ohjelmakoodi painonappien luomiseen Grafanassa

Liite 5. Sisältö config.json-tiedostoon

Liite 6. Sisältö homebridge-tiedostoon

Liite 7. Sisältö homebridge.service-tiedostoon

Lyhenteet

CSV	*Comma Separated Value*. Tiedostomuoto, joka erottelee tietoa pilkulla ja rivinvaihdolla tekstitiedostoon.
DNS	*Domain Name System*. Järjestelmä, joka muuttaa selkokielisen nimen numeeriseksi IP-osoitteeksi.
IoT	*Internet of Things*. Esineiden internet.
JSON	*JavaScript Object Notation*. Avoimen standardin formaatti tiedonvälitykseen.
MAC	*Media Access Control*. Laitteen lähiverkossa yksilöivä koodi.
MQTT	*Message Queuing Telemetry Transport*. Avoimeen lähdekoodiin perustuva viestiprotokolla.
NoSQL	*No Structured Query Language*. Tavallisesta SQL-relaatiomallista poikkeava tietokanta.
UDP	*User Datagram Protocol*. Yhteydettömän palvelun toteuttava protokolla.
Wi-Fi	*Wireless Fidelity*. Langaton lähiverkko.

1 Johdanto

Tässä insinöörityössä esitellään yleisesti, miten teknisesti suuntautunut peruskuluttaja voi nykyisellään kotiautomaatiosta hyötyä ja miten kotiautomaation lisäämistä omaan kotiin voi lähteä toteuttamaan. Esittelystä rajataan ulkopuolelle kiinteästi asennettavat järjestelmät ja järjestelmät, joiden asentamiseen tarvitaan sähköalan ammattilaisen apua. Nykyisin kodinkoneita ja viihde-elektroniikkaa myyvistä kaupoista löytyy satoja erilaisia kotiautomaatiotuotteita, jotka on tarkoitettu peruskuluttajien asennettaviksi ja käytettäviksi. Tämä insinöörityö antaa hyödyllistä perustietoa näiden tuotteiden hyödyntämisestä, vaikka lukija ei ymmärtäisikään tämän työn kaikkia teknisiä yksityiskohtia.

Esimerkkityöksi rakennetaan myös kasvien ylläpitoon tarkoitettu Raspberry Pi -pohjainen etäohjattava IoT-laite, jonka toimintoja ovat mm. mittausdatan kerääminen useilta eri sensoreilta ja sen esittäminen eri käyttöliittymistä. Laite toimii itsessään kodinohjausjärjestelmän keskusyksikkönä, johon voidaan lisätä lisää laitteita ja luoda niiden välille haluttua automaatiota. Laite toimii myös tiedonkeruulaitteena sisältäen myös sähkölaitteiden ohjauksen relekortin välityksellä. Laite liitetään osaksi Applen HomeKit-kotiautomaatiosovellusta, mutta sitä voidaan hallita myös Android-laitteilla käyttäen Homebridge-ohjelman käyttöliittymää.

Esimerkkityö on esitelty siten, että eri työvaiheita voi käyttää myös oppaana, tai vaihtoehtoisesti työvaiheita selostettaessa ohjataan tutustumaan lähteeseen, josta tarvittavat tiedot löytyvät. Työn lukijan pitäisi hallita ohjelmoinnin perusteita siinä määrin, että hän ymmärtää tyypillisimpiä ohjelmointikielen käsitteitä, kuten erilaisia toistorakenteita ja tietotyyppejä. Ilman aikaisempaa kokemusta ohjelmoinnista voi olla hyvin vaikeaa seurata ja soveltaa esimerkkityössä esiteltyjä työvaiheita.

2 Kotiautomaatio

2.1 Kotiautomaation historiaa

Sähköllä toimivia kotona asumista helpottavia keksintöjä on ollut markkinoilla jo toista sataa vuotta. Esimerkkinä 1900-luvun alkupuolella yleistynyt jääkaappi, jossa ruoka pysyi kylmänä itsestään, tai ovikello, joka mahdollisti tiedonkulun talon sisällä sähköjohtoja pitkin. Jo tuolloin oli saatavilla sähkölaitteille kellokatkaisimia ja erilaisia sähköllä toimivia hälyttimiä, mutta varsinaista kotiautomaatiota ei tuon ajan sovellutuksista vielä löytynyt.

Varsinaisen automaation rantautumisen Suomeen saivat aikaan Lähi-idän sodat 1970-luvulla, kun länsimaille suuttuneet öljyntuottajamaat nostivat yllättäen öljyn hintaa. Tällöin öljystä oli pulaa ja silloinen sähköverkko oli antanut merkkejä ylikuormituksesta, kun öljyllä lämmittämisestä siirryttiin yllättäen sähkölämmitykseen. Tällöin valtiovalta asetti uusilla säädöksillä kiinteistönomistajat hakemaan keinoja energiakulujen pienentämiseksi. Samalla vuosikymmenellä yleistyivät myös mikroprosessorit, jotka mahdollistivat digitaalitekniikan mukaan tulon automaatioon. (Suomäki & Vepsäläinen 2018, 9; Lindfors 2018.)

1980─1990-luvulla kiinteistöissä alkoivat yleistyä ohjausjärjestelmät, jotka oli suunniteltu hoitamaan kiinteistöjen eri osa-alueita oman käyttöliittymän ja keskusyksikön kautta. Suosittuja järjestelmiä olivat mm. murtovalvonta-, kameravalvonta- ja kulunvalvontajärjestelmät. Vasta 2000-luvulla markkinoille ilmaantui järjestelmiä, joita alettiin yleisesti kutsua taloautomaatioksi tai kotiautomaatioksi. (Hattunen 2015: 17.)

2000-luvun kotiautomaatiojärjestelmät erosivat 1990-luvun järjestelmistä siten, että ennen erillisinä järjestelminä olleet toiminnot alkoivat löytyä saman järjestelmän piiristä. Tämän mahdollistivat eri sähkötarvikevalmistajien yhteistyö ja avoimet järjestelmästandardit, kuten väylätekniikalla toteutettu KNX. Tämä helpotti paljon järjestelmien suunnittelua ja rakentamista, kun laitteista voitiin muodostaa erilaisia järjestelmiä eri käyttötarkoituksiin valmistajasta riippumatta.

Alkuun kotiautomaatiojärjestelmän laitteet olivat yleensä aina kiinteästi asennettavia ja yhdistetty langallisesti. Langattomien kotiverkkojen yleistymisen myötä langattomasti

toimivien kotiautomaatiotuotteiden määrä on kasvanut huomattavasti. Langaton tekniikka teki laitteiden asentamisesta helppoa, koska useimmat laitteet voitiin asentaa jälkiasennuksena ilman sähköalan ammattilaisen apua ja laitteiden yhdistäminen kotiautomaatioon voitiin tehdä helposti järjestelmän käyttöliittymän kautta ilman sen kummempaa asiantuntemusta.

Kotiautomaatiotuotteista puhuttaessa tarkoitetaan usein älylaitteita, eli tietokonepohjaisista laitteista. Matemaatikko Alan Turing (1912–1954), jota pidetään modernin tietojenkäsittelyn yhtenä merkittävimmistä uranuurtajista, määritteli tietokoneen olevan älykäs vasta sitten, kun se läpäisee kokeen, jossa tietokoneen antamista vastauksista on mahdotonta erottaa ihmisen antamia vastauksia. Tietokoneen olisi siis kyettävä antamaan ihmismäinen vaikutelma itsestään, ennen kuin se voitaisiin mieltää älykkääksi. (Lehtonen 2018.) Nykyisellään laite voidaan mieltää ”älykkääksi”, jos se täyttää IoT-laitteen määritelmän.

Nykyisellään IoT-sanasta (Internet of Things) on muodostunut muotisana, ja sillä voidaan viitata useampiinkin eri asioihin, mutta kotiautomaatiosta puhuttaessa sillä usein tarkoitetaan fyysisiä laitteita. Alve (2018: 2) Suomen sähköteknillisestä standardisoimisyhdistyksestä (SESKO ry) määrittelee opetusmateriaaleissaan IoT-laitteen seuraavasti:

> IoT-laitteella tarkoitetaan mitä tahansa laitetta, joka voi kytkeytyä verkkoon ja kerätä ja jakaa tietoja verkossa. Yleensä sen ymmärretään olevan fyysinen esine, joka on kytketty internetiin ja jota voidaan ohjata internetin kautta.

IoT-käsite syntyi 1990-luvun loppupuolella, kun radiotaajuuksilla toimivat tekniikat ottivat uusia kehitysaskeleita mahdollistaen laitteiden välisen kommunikoinnin. Tuolloin kehitystä kuitenkin jarruttivat vielä kalliit komponentit kuten anturit ja tiedonsiirtolaiteet. Myös tehonkulutus esti useissa tapauksissa sovellusten kehittämisen. Nämä esteet ovat nyt pääosin poistuneet. (Seppä 2017.)

Nykyaikainen kotiautomaation toiminta perustuukin juuri esineiden internetiin. Tällä hetkellä kotiautomaation toiminnot ovat melko yksinkertaisia, suurimmaksi osaksi vain ”päällä/pois päältä” -tyyppisiä, ja niihin ei liity automaatiotoimintoja, ellei käyttäjä ole niitä itse määritellyt. Kotiautomaatio kuitenkin kehittyy koko ajan, ja tulevaisuudelta

odotetaankin jo seuraavaa vaihetta, jossa kotoa alkaisi löytyä aitoa robotiikkaa ja eri toiminnot osaisivat ”ajatella” itse. (Siren 2019.)

2.2 Yleisimmät käyttökohteet

Kotiautomaation tehtävä on ensisijaisesti tukea hyvinvointia ja asumisviihtyvyyttä. Myös kodin turvallisuuteen ja energiatehokkuuteen voidaan vaikuttaa lisäämällä tiettyjä kotiautomaatiosovelluksia. Kodin toiminnot voidaan jättää joko kokonaan tai osittain tietokoneen hoidettavaksi jättäen järjestelmänkäyttäjälle vain tarkkailijan rooli.

Kotiautomaatiotuotteiden valmistajat ovat keskittyneet kodin valaistuksen, lämmityksen, turvallisuuden, sähkönsyötön ja viihdejärjestelmien ohjaamiseen. Myös useat kodinkoneet voidaan yhdistää osaksi kodin automaatiojärjestelmää. Voi esimerkiksi yhdistää osaksi järjestelmää robotti-imurin tai pesukoneen ja ohjata niiden toimintoja kodin ulkopuolelta käsin. Se, kuinka pitkälle automaatiosovellutukset halutaan viedä kotioloissa, on vain käyttäjästä kiinni, koska nykyisillä kotiautomaatiosovelluksilla lähes kaikki on mahdollista. Olisikin tärkeä lähteä liikkeelle näkökulmasta, jossa punnitaan toteutusten järkevyys.

2.2.1 Valaistus

Ehkä yleisin ja helpoin tapa aloittaa kotiautomaatioon tutustuminen on valaistuksen ohjaus. Useat eri yhtiöt myyvät omia älyvalaistustuotteitaan monilla eri ominaisuuksilla. Tyypillisiä ominaisuuksia älylampuissa ovat himmennyksen säätömahdollisuus, valaisuvärin muuttaminen ja lamppujen hallinta useista käyttöliittymistä, kuten mobiililaitteista ja kodinohjausjärjestelmän näyttöpaneelista. Älyvalaistukseen siirtymisellä voidaan vaikuttaa mm. sähkönkulutukseen asettamalla valot syttymään liikesensoreiden aktivoimina tai säätämällä ajastuksia, joilla voidaan välttää lamppujen turha päällä oleminen. Myös tavalliset lamput voidaan muuttaa osaksi kodin automaatiota sijoittamalla älypistorasia lampun ja sähköverkon väliin tai asentamalla valokatkaisimen alle kytkinrele, joka on yhteydessä valojen ohjauslaitteeseen.

2.2.2 Lämmityksen ja energian käytön hallinta

Automaation avulla voidaan seurata tehokkaasti veden, sähkön ja lämmitysenergian kulutusta, mikä mahdollistaa myös merkittävien säästöjen aikaansaamisen monissa tapauksissa. Parhaimmillaan tehdyt automatisoinnit maksavat itsensä säästöinä takaisin jo muutamassa vuodessa.

Varsinkin omakotitalossa asuvan kannattaisi perehtyä tämän kategorian tuotteisiin, koska energiankulutus voidaan asettaa seurantaan ja näin energiaa paljon kuluttaviin toimintoihin voidaan alkaa vaikuttaa. Esimerkiksi nykyaikaisilta lämmönjakokeskuksilta ja sähkötauluista voidaan seurata energiankulutusta reaaliaikaisesti ja tuoda tämä tieto osaksi kotiautomaatiota. Kun mitattuihin tietoihin lisätään vielä sähkön SPOT-hinta ja säätilan seuranta, voidaan automaation avulla ohjata energiankäyttöä niihin hetkiin, kun se on hinnan ja ajankohdan vuoksi järkevintä. Yksittäiselle kerros- tai rivitaloasukkaalle energiakulut ovat yleensä osa hoitovastiketta, joten niihin ei juurikaan voida itse vaikuttaa (poikkeuksena sähkölämmitteiset taloudet).

Lämmitystä voidaan yksinkertaisimmillaan ohjata seinään kytkettävillä älypistokkeilla tai elektronisilla patteritermostaateilla (kuva 1), kun lämmitys asetetaan toimimaan talon asukkaiden elämän rytmin mukaan. Lämmitystä voidaan esimerkiksi asukkaiden poissaolon ajaksi laskea useilla asteilla tai yksittäistä huoneen lämpötilaa voidaan säätää etänä tarpeen mukaan.

Kuva 1. Netatmon valmistama älytermostaatti, joka asennetaan kodin vesipatteriin ja yhdistetään Wi-Fi-verkon kautta etäohjattavaksi (Netatmo 2019).

Kun kodin lämpötilaan ja ilmastointiin aletaan vaikuttaa, pitäisi tietää kiinteistön rakenteille sallitut poikkeamat lämpötiloissa ja ilman kosteudessa. Esimerkkinä epäonnistuneesta lämpötilojen ja ilmastoinnin ohjauksesta Suomäki ja Vepsäläinen (2018: 127) kertovat kirjassaan, että etenkin julkisella sektorilla säästettiin vuosikausia heikentämällä kiinteistöjen olosuhteita yli sallittujen arvojen. Myöhemmin huomattiin säätökohteiksi valikoituneissa kiinteistöissä vakavia sisäilmaongelmia.

2.2.3 Turvallisuus

Kodin turvallisuuteen voidaan vaikuttaa kotiautomaation keinoin vaihtamalla vanha lukko ja ovikello älykkäiksi sisältäen esim. kasvojen tunnistuksen ovikameralla tai etäavausmahdollisuuden älypuhelimella. Käyttäjä voi päästää esim. siivoojan sisään asuntoon työpaikaltaan tai saada ilmoituksen kännykkäänsä, kun lapsi on saapunut koulusta kotiin. Myös paljon erilaisia valvontakameroita on saatavilla eri käyttötarkoituksiin esim. ulko- ja sisävalvontaan, itkuhälyttimeksi tai lemmikkieläinten seuraamiseen. Kameratyypistä riippuen ne voivat sisältää omat liiketunnistimet tai mahdollisuuden yökuvaukseen ja kaksisuuntaiseen äänen lähettämiseen.

Myös paljon erilaisia hälyttimiä on saatavilla runsaasti esim. vesivuotojen, savukaasujen ja ikkunanavauksien valvontaan. Ikkunanavauksesta aiheutunut viesti voidaan haluttaessa lähettää käyttäjän lisäksi myös suoraan paikalliseen vartiointipalveluun.

2.2.4 Viihdejärjestelmät

Kun kotiin ostetaan älytoimintoja tukevia kaiuttimia, voidaan kaiuttimet asettaa toimimaan yhtenä langattomana äänentoistojärjestelmänä. Näin saadaan esim. sama musiikki soimaan talon eri osissa. Myös vanhat kaiuttimet voidaan liittää langattomasti äänentoistojärjestelmään lisäämällä Wi-Fi- tai Bluetooth-vastaanotin kaiuttimeen tai äänentoistojärjestelmään.

Nykyiset älytelevisiot voidaan myös asentaa osaksi kotiautomaatiota. Television eri toimintoihin voidaan vaikuttaa näin suoraan ilman kaukosäädintä. Television liittämisestä kotiautomaatioon saadaan pitkälti samanlaista hyötyä kuin valojen ohjaamisesta. Television voi vaikkapa halutessaan ajoittaa käynnistymään arkiaamuina näyttämään uutisia

tai television häiritessä isoäänisesti voi äänenvoimakkuutta säätää pienemmälle toisesta huoneesta käsin.

2.3 Yleistä kotiautomaatiojärjestelmän käyttöliittymän valinnasta

Jotta saadaan etäohjattava automaatiojärjestelmä, on kodista löydyttävä internetyhteys. Tämän lisäksi tarvitaan kotiautomaation keskusyksikkö, joka valmistajasta riippuen yhdistetään internetiin kaapelin tai Wi-Fi:n avulla. Keskusyksikkö linkittää käyttäjän kodin ulkopuolelta kodin laitteisiin. Keskusyksikköön myös tallennetaan muistiin käyttäjän määrittämät laitteet ja automaatiotoiminnot.

Kuluttajien harmiksi markkinoilla on tällä hetkellä monia erilaisia järjestelmiä, jotka eivät ole yhteensopivia toistensa kanssa. Tämä johtuu esimerkiksi siitä, että järjestelmät käyttävät erilaisia yhteystekniikoita tai ohjelmistot on suunniteltu siten, että ne tunnistavat vain tiettyjen valmistajien laitteita. On mahdotonta sanoa, mikä yhteystekniikka kannattaisi valita omaan kotiin, sillä jokaisessa tekniikassa on hyvät ja huonot puolensa. Yhteystekniikoita myös päivitetään jatkuvasti, mikä voi yllättäen nostaa vanhan tekniikan taas hyväksi vaihtoehdoksi. Tästä esimerkkinä yli 20 vuotta vanha Wi-Fi-verkko, joka on vuosien saatossa jatkuvasti nopeutunut tiedonsiirrossa ja saanut uusia ominaisuuksia, kuten viimeisimpänä Mesh-tekniikka ja WP3-salaus. Tällä hetkellä kotiautomaatiota hallitsevasta yhteystekniikasta käydään kovaa kilpailua, ja onkin hyvin mahdollista, että lähivuosina nähdään joidenkin tekniikoiden katoavan markkinoilta samaan tapaan kuin CD-teknologia hävisi USB-muistitikkujen tultua markkinoille.

Omaa kotiautomaatiojärjestelmää valittaessa pitäisi siis jo suunnilleen tietää, mitä laitteita tulee yhdistämään kotiautomaation piiriin, sillä sama yhteystekniikka kaikissa laitteissa helpottaisi huomattavasti järjestelmän hallintaa. Toki on myös mahdollista hankkia keskusyksikkö, josta löytyy tuki useampien yhteystekniikoiden hallintaan, mutta se tarkoittaa yleensä korkeita hankintakustannuksia ja kömpelöitä käyttöliittymiä, joiden pysyminen ajan tasalla päivitysten johdosta voi olla valmistajasta riippuen hyvin epävarmaa. Tarjottavat päivitykset ovat hyvin tärkeitä IoT-laitteille yleisen toiminnallisuuden ja tietoturvan takia (Savonia 2019).

Olisikin hyvä miettiä, mitä laitteita kotoa löytyy jo valmiiksi, eli onko kodin käyttäjillä esimerkiksi Androidi vai Apple iOS-laitteita. Esimerkiksi Apple Tv tai iPad toimivat jo sellaisenaan kotiautomaation keskusyksikkönä, ja tätä kautta älylaitteet saadaan hallittaviksi Apple HomeKit -sovellukseen. Huomionarvoista on myös, että valittaessa Apple HomeKit -sovellus käyttöliittymäksi, suljetaan Android-puhelinten käyttäjät järjestelmän ulkopuolelle, koska Apple HomeKit -sovellus on saatavilla vain Apple iOS -tuotteisiin. Android-käyttäjät voivat valita järjestelmäkseen minkä tahansa muun käyttöliittymän paitsi Apple HomeKit -sovelluksen.

Suosittua on myös käyttää avoimen lähdekoodin ohjelmistoja kotiautomaation hallintaan, kuten Home Assistant ja OpenHAB. Nämä voivat olla hyvä vaihtoehto, jos ollaan valmiita käyttämään aikaa käyttöliittymän rakentamisen ja eri laitteiden yhdistämisen opetteluun. Toisin kuin kaupallisten merkkien käyttöliittymissä, avoimen ohjelmakoodin käyttöliittymissä ei yhteensopivuusongelmia juuri ole, ja kotiautomaation keskusyksikkönä voidaan käyttää miltei mitä tahansa tietokonetta, esim. edullista yhden piirilevyn Raspberry Pi -tietokonetta. Avoimen lähdekoodin ohjelmistoja käyttämällä voidaan siis välttyä tilanteilta, että ajan kuluessa kotoa alkaa löytyä useita eri älylaitteiden keskusyksiköitä ja niiden ohjaamiseen tarvitaan kuhunkin oma sovelluksensa.

2.4 Järjestelmien protokollat

Tässä työssä ei syvennytä tarkemmin kotiautomaatiojärjestelmien käyttämiin protokolliin, eli yhteyskäytäntöihin. Kotiautomaatiolaitteiden hankintaa harkitsevan olisi kuitenkin hyvä tietää, että niitä on monia, ja jos eri protokollia halutaan yhdistää ohjattavaksi saman järjestelmän piiriin, törmätään todennäköisesti yhteensopivuusongelmiin. Seuraavassa on lueteltuna yleisimpiä protokollia:

- Z-Wave
- ZigBee
- Bluetooth
- Wi-Fi
- 433 MHz
- GSM.

Kaikki edellä mainitut ovat langattomasti toimivia tekniikoita. Useat kodinlaitteet, kuten televisio, saadaan yhdistettyä myös tutulla Ethernet-kaapelilla. Kotiautomaatioprotokollista puhuttaessa törmätään väistämättä myös käsitteeseen Mesh, eli laitteiden muodostamaan solmuverkkoon, jota tukevat ainakin Z-Wave-, ZigBee-, Bluetooth 5- ja jotkin Wi-Fi-laitteet. Solmuverkko eroaa tavanomaisesta verkosta siten, että laitteet ovat yhteydessä suoraan toisiinsa ja voivat toistensa kautta yhdistää toisiin laitteisiin. Tavanomaisessa verkossa jokainen laite on taas yhteydessä keskusyksikön kautta toisiin laitteisiin. Kun valitaan solmuverkkoa tukeva tekniikka kotiautomaation toteutukseen, saadaan kodin langaton verkko levittäytymään laitteiden mukana isollekin alalle.

3 Esimerkkitoteutus kotiautomaatioon lisättävästä IoT-laitteesta

Tässä esimerkissä rakennetaan tee-se-itse-hengessä ohjelmakokonaisuus kasvien ylläpitoon ja kodinohjaukseen Raspberry Pi -tietokoneella. Järjestelmän mittaukseen liittyvät toiminnot ovat

- kasvualustan kosteusprosentti
- kasvualustan EC-arvo (sähkönjohtavuusarvo)
- valon voimakkuus
- lämpötila
- ilmankosteus
- hiilidioksiditaso.

Tärkeimmät toiminnot ja ominaisuudet järjestelmälle ovat

- valvonta- ja ohjaus etäyhteydellä
- automaatiotoimintojen lisääminen
- hälytystoiminnot
- historiatietojen kerääminen mittausarvoista
- kuvaajien piirtäminen mittausarvoista
- etäyhteydellä hallittavat pistorasiat
- videokuva osana käyttöliittymää
- hallittavuus Apple iOS- ja Android-laitteilla
- laajennettavuus uusilla sensoreilla ja yhteystekniikoilla.

3.1 Esitietovaatimukset asennusohjeiden seuraamiseen

Esimerkkityön toteutus on selostettu siten, että sitä voitaisiin myös pitää oppaana kyseisen järjestelmän rakentamiselle tai vain joidenkin osien asentamiselle. Ohjeiden seuraamisen edellytyksenä on se, että hallitaan Linux-järjestelmän perustoiminnot

komentoriviltä käsin, koska kaikkia vaiheita ei ole välttämättä kirjoitettu tai ne oletetaan jo tehdyksi. Näitä ovat mm.

- yleisten järjestelmäasetusten määritys Raspberry Pi:lle
- hakemistoissa siirtyminen
- tiedostojen suorittaminen
- pakettien lataaminen ja asennus
- nano-tekstieditorin käyttö
- perustuntemus ohjelmien toistorakenteista ja tietotyypeistä
- kodin reitittimen laitekohtainen porttiohjaus.

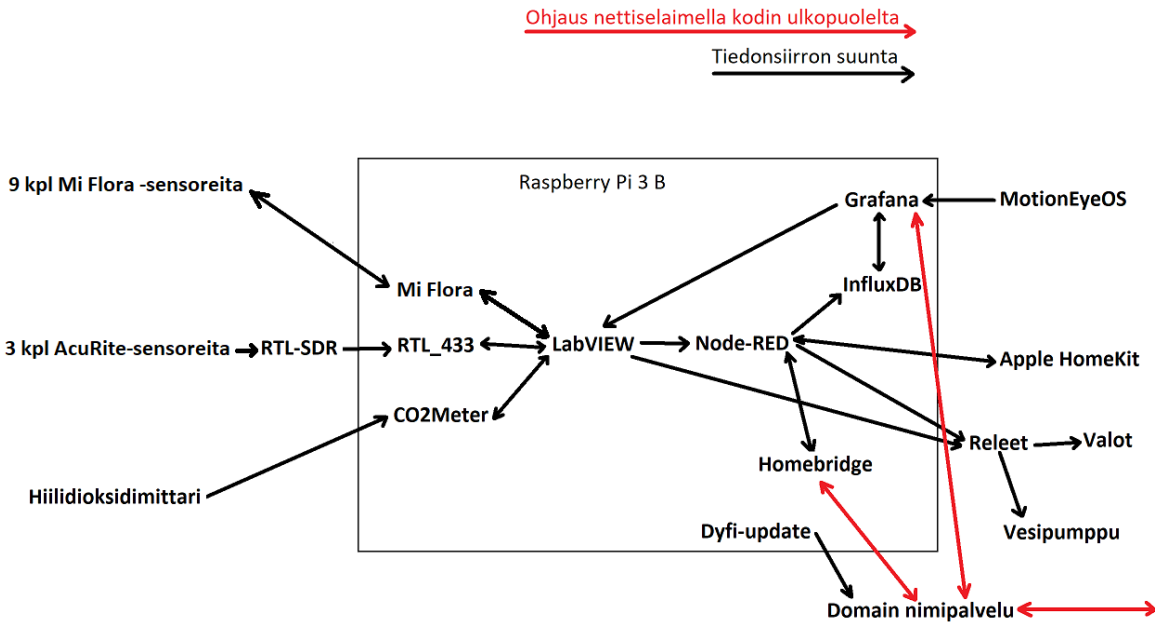
On myös mahdollista, että tiedostoja suoritettaessa tai asennettaessa törmätään yllättäviin virheviesteihin, esim. Python-tulkki ilmoittaa puuttuvasta moduulista. Näistä tilanteista pitäisi pystyä omin avuin pääsemään eteenpäin. Järjestelmää rakentavalta edellytetään myös englannin kielen taitoa, jotta voidaan lukea esimerkkityön ulkopuolisia ohjeita. Ehdottoman tärkeää olisi myös hallita sähkötekniikan perusteita siinä määrin, että osataan tehdä tarvittavat kytkennät oikein ja turvallisesti, mikäli relekortilla halutaan ohjata verkkovirtaa (230 V). Jos kokemusta ja taitoa ei ole, mutta intoa sähkötyöharjoituksiin kuitenkin löytyy, tulisi harjoitukset aloittaa matalilla n. 3–30 V:n tasajännitteillä, jotka eivät ole hengenvaarallisia ihmiselle. Sähköä voi ottaa harjoituksiin esimerkiksi kotoa löytyvistä akuista tai paristoista. Myös säädettävän laboratoriovirtalähteen (hinta n. 50 €) hankinta voi olla järkevää, mikäli uskoo harrastusinnostuksen jatkuvan.

3.2 Ohjelmaversiot ja tiedonsiirto

Jos tässä työssä käytettyjen ohjelmien versioista poiketaan, on mahdollista, että törmätään yhteensopivuusongelmiin. Projektissa käytetyt ohjelmistoversiot olivat seuraavanlaiset:

- Raspbian Jessie (2017-04-10)
- Node.Js (10.17.0)
- Npm (6.11.3)
- RTL_433 (18.12-207-g82d0f55)
- Mi Flora Plant Sensor MQTT Client/Daemon (1.0)
- CO2Meter (1.0)
- MotionEyeOS (20190911)
- Dyfi-update (1.2.0)
- LabVIEW LINX (3.0.1.192)
- Node-RED (0.20.8)
- InfluxDB (1.7.8)
- Grafana (6.3.6)
- Apple Homekit (13.1.3)
- Homebridge Config UI X (4.6.2).

Tiedonsiirto ohjelmien ja laitteiden välillä tapahtuu kuvassa 2 olevan periaatekaavion mukaisesti. Kuvassa Raspberry Pi:lle asennetut ohjelmat on piirretty mustan kehyksen sisäpuolelle ja ulkopuolelle on jätetty mm. relekortti, sensorit ja muissa laitteissa toimivat ohjelmat. Huomataan, että useissa tapauksissa tiedonsiirto tapahtuu vain yhteen suuntaan. Esimerkiksi AcuRite-sensorit lähettävät 433 MHz:n taajuudella mittausarvoja riippumatta siitä, oliko RTL-SDR vastaanottamassa signaaleja vai ei. Mi Flora -sensorit puolestaan lähettävät mittausarvot Bluetooth-yhteyden välityksellä vain silloin, kun Mi Flora -ohjelma pyytää arvoja sensorilta.



Kuva 2. Periaatekaavio tiedonsiirron suunnasta laitteiden ja ohjelmien välillä.

Mittausarvoja kerätään Mi Flora-, RTL_433- ja CO2Meter-ohjelmilla ja lähetetään UDP-protokollaa käyttäen LabVIEW-ohjelmalle käsiteltäviksi. Prosessoituaan arvot LabVIEW muodostaa JSON-syötteen (liite 1 ja 2), josta Node-RED käy lukemassa sensoriarvot kerran minuutissa. Node-RED syöttää tiedot tietokantaan, sekä Apple Homekit - ja Homebridge-käyttöliittymiin. Lisäksi Node-RED yhdistää käyttöliittymien ohjauksen releisiin.

Grafana toimii puolestaan käyttöliittymänä, joka lukee InfluxDB-tietokantaan syötettyjä arvoja. Lisäksi se saa videokuvan lähiverkossa toimivalta IP-kameralta ja voi lähettää vesipumppua ohjaavan käskyn suoraan LabVIEW-ohjelmaan.

Lisäksi järjestelmässä toimii itsenäisesti Dyfi-update-ohjelma, joka ottaa säännöllisesti yhteyden Domain-nimipalvelimelle kertoen Raspberry Pi:n julkisen IP-osoitteen. Näin ollen nimipalvelin osaa aina ohjata www.esimerkki.dy.fi-sivustolle tulleen henkilön oikeaan IP-osoitteeseen.

3.3 Laitteisto

3.3.1 Raspberry Pi 3 B

Raspberry Pi on Brittiläisen Raspberry Pi Foundation -säätiön kehittämä yhden piirilevyn tietokone. Ensimmäinen Raspberry Pi -tietokonemalli julkaistiin vuonna 2012, jonka jälkeen uusia malleja on julkaistu lähes vuosittain. Raspberry Pi eroaa tavanomaisista tietokoneista mm. hintansa, kokonsa, virrankulutuksensa ja kytkentöjensä kautta. Erityisen kiinnostavan Raspberry Pi:stä tekevät sen 40 GPIO -liitännät (kuva 3), jotka mahdollistavat eri laitteiden helpon liittämisen suoraan piirilevyyn. Myös pienen SD-kortin käyttömahdollisuus tietokoneen kovalevynä helpottaa järjestelmästä toiseen hyppäämistä, kun kortti on helposti vaihdettavissa. (Raspberry Pi Foundation 2019a.)

Kuva 3. Raspberry Pi 3 B -tietokone (Raspberry Pi Foundation 2019a).

Koska tässä työssä sensoreilta halutaan kerätä mittaustuloksia vuorokauden ympäri vuodesta toiseen, tulee Raspberry Pi:n olla aina päällä. Tässä yhteydessä korostuvat Raspberry Pi:n hyvät puolet: se on pienikokoinen, äänetön ja energiatehokas sekä kykenee useiden ohjelmien yhtäaikaiseen suorittamiseen. Tämän projektin kokoonpanoa energiamittarilla testattaessa Raspberry Pi 3 B toimi 4,5 W:n sähköteholla, joka vuositasolla maksaa noin 5,5 € (14 snt/kWh).

3.3.2 RTL-SDR-radiovastaanotin

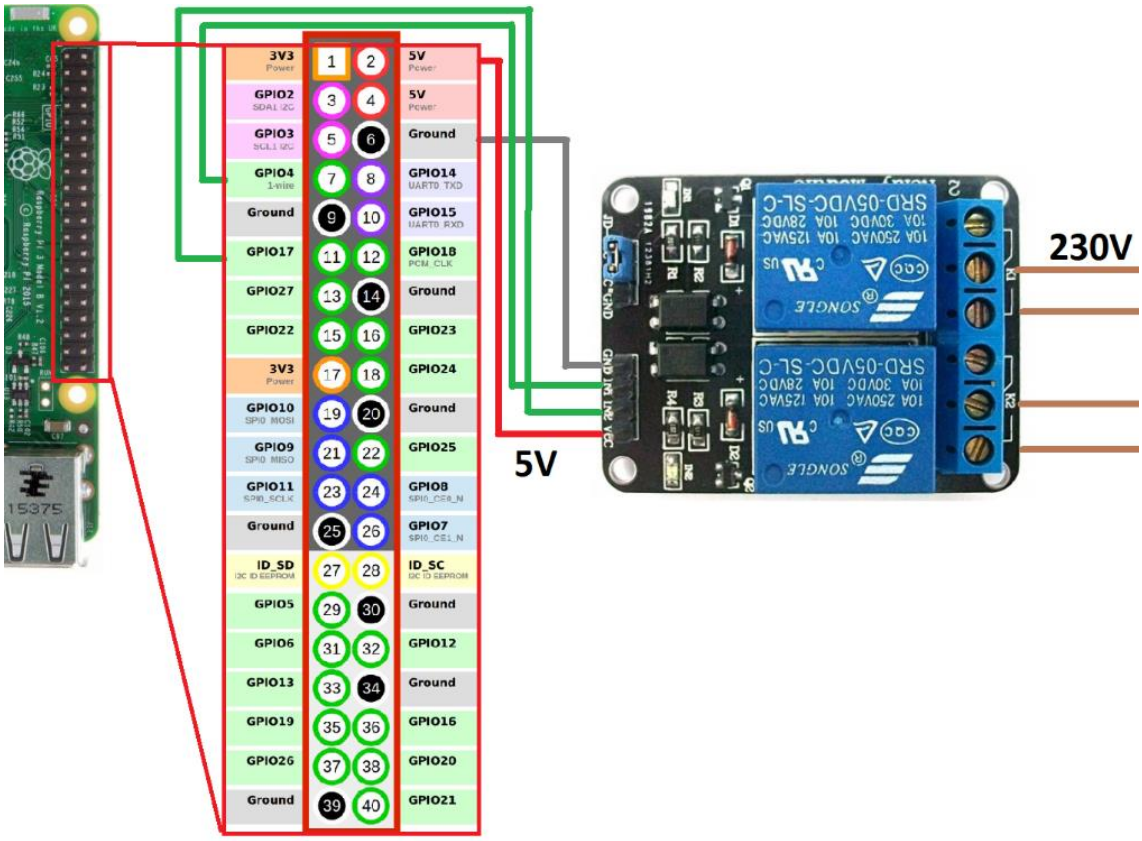
RTL-SDR-radiovastaanotin on alun perin tarkoitettu digitaali-tv-lähetysten vastaanottoon, mutta sitä voidaan käyttää myös yleiskäyttöisenä radiovastaanottimena (kuva 4). Laite on hyvin suosittu mm. radioamatöörien ja tee-se-itse-elektroniikkaharrastajien keskuudessa, ja laitteen käytöstä on kirjoitettu myös oma opaskirjansa (*The Hobbyist's Guide to the RTL-SDR: Really Cheap Software Defined Radio*). Laitteella voidaan mm. kuunnella lentokoneliikennettä, satelliitteja ja muuta radioliikennettä. Toiminta perustuu RTL2832U-piiriin, jonka vastaanottamiin I/Q-signaaleihin päästään käsiksi. Riippuen mallista sillä voidaan vastaanottaa radiosignaalia 500 kHz:n taajuudesta 1 770 000 kHz:iin. RTL-SDR:lle on saatavilla useita ohjelmistoja riippuen siitä, mitä laitteen vastaanottamilla signaaleilla halutaan tehdä. (RTL-SDR.COM 2019.)

Kuva 4. USB:llä Raspberry Pi:hin yhdistettävä RTL-SDR-radiovastaanotin.

Tässä projektissa RTL-SDR-tikulla seurataan kolmea 433 MHz:n taajuudella toimivaa lämpötilasensoria. Edullinen hinta (n. 10 €), valmis kotelointi ja ohjelmisto signaalin purkamiseen tekevät tästä helposti lisättävän ominaisuuden esimerkkiprojektiin.

3.3.3 2-kanavainen relekortti

Koska projektissa halutaan ohjata kasvivaloja ja vesipumppua, tarvitaan relekortti, jotta matalalla Raspberry Pi:n ohjausjännitteellä voidaan ohjata 230 V:n verkkovirtaa. Releen yhdistäminen Raspberry Pi:n liitäntöihin (kuva 5) onnistuu parhaiten käyttäen johtoja, joissa kummassakin päässä on pinneihin sopiva naarasliitin. Raspberry Pi:n 5 V -pinni (kuvassa punainen) yhdistetään relekortin VCC-liitäntään ja maajohto (kuvassa harmaa) relekortin GND-liitäntään. GPIO-johdot yhdistetään Raspberry Pi:n pinneistä 7 ja 11 relekortin IN1- ja IN2-liitäntöihin. Pienen hyppyliittimen (kuvassa sininen osa relekortin vasemmassa yläreunassa) pitää olla asetettuna välille JD─VCC. Verkkovirran vaihejohdot on kytketty relekortin NO-liitäntään (Normally Open). Tästä seuraa se, että virran hävitessä järjestelmästä myös releen kautta kulkeva verkkovirta katkeaa.



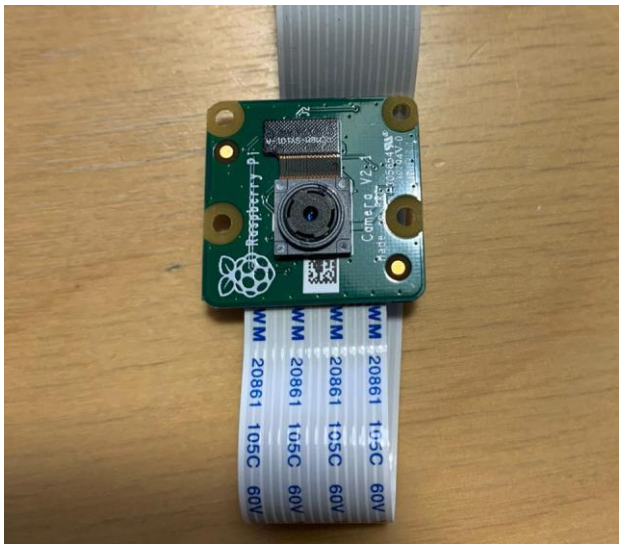
Kuva 5. Havainnollistamiskuva Raspberry Pi 3 B:n ja 2-kanavaisen relekortin yhdistämisestä.

Ennen sähkötöihin ryhtymistä täytyy muistaa, että 230 V:n verkkojännite on ihmiselle hengenvaarallinen ja jos on vähänkin epävarma liitäntöjen oikeellisuudesta, ei

sähkötöihin pitäisi ryhtyä ollenkaan. Työhön ryhdyttäessä pitäisi esimerkiksi ymmärtää, kuinka paljon virtaa mikäkin sähkökomponentti pystyy kestämään. Tämän projektin relekortti kestää maksimissaan vain 10 A:n virtaa, ja se ylittyy helposti, jos suuritehoisia sähkölaitteita kytketään kortin ohjattavaksi. Suomessa sähkötöiden tekeminen on myös säänneltyä toimintaa, jota saavat harjoittaa pääasiassa vain sähköalan ammattilaiset. Kuitenkin harrastustoimintana omaan käyttöön saa sähkölaitteen rakentaa. On hyvä muistaa myös, että jos laite aiheuttaa kotona onnettomuuden, joka menisi normaalisti vakuutusyhtiön korvattavaksi, voi vakuutusyhtiö alentaa korvausta siinä tapauksessa, että he voivat osoittaa onnettomuuden johtuneen väärin rakennetusta laitteesta. (Sähköturvallisuuslaki 2016; Linja-aho 2019.)

3.3.4 Raspberry Pi -kameramoduuli

Kun lisätään järjestelmään IP-kamera, voidaan kasvien kuntoa ja kastelutapahtumaa seurata etänä live streamin kautta. Kameraksi tähän projektiin käytetään toista Raspberry Pi 3 -tietokonetta. Kameran kuntoon laittamiseen riittää, että Raspberry Pi -kameramoduulin kaapeli asennetaan kiinni piirilevyyn. Tässä esimerkissä käytetään Raspberry Pi 8.0 Mpix v2 -kameraa (kuva 6).



Kuva 6. Raspberry Pi:n 8 megapikselin v2-kameramoduuli. Kameraa käytettiin tässä projektissa yhdessä MotionEyeOS-ohjelman kanssa.

3.3.5 Gardena-lomakastelusarja

Kasvien kasteluun käytetään Gardena-lomakastelusarjaa (kuva 7). Tähän kävisi myös mikä tahansa verkkovirtaan kytkettävä vesipumppu, mutta tällä paketilla saadaan kasvien kastelu toteutettua helposti, koska kyseessä on valmis paketti, joka sisältää oman virtalähteen, pumpun ja letkut. Kastelusarja on suunniteltu niin, että uppopumppu nostaa vettä säiliöstä kolmen eri runkoputken kautta, joista vesi haarautetaan edelleen pienempiin letkuihin, jotka johtavat kasviruukkuihin (kuva 8).

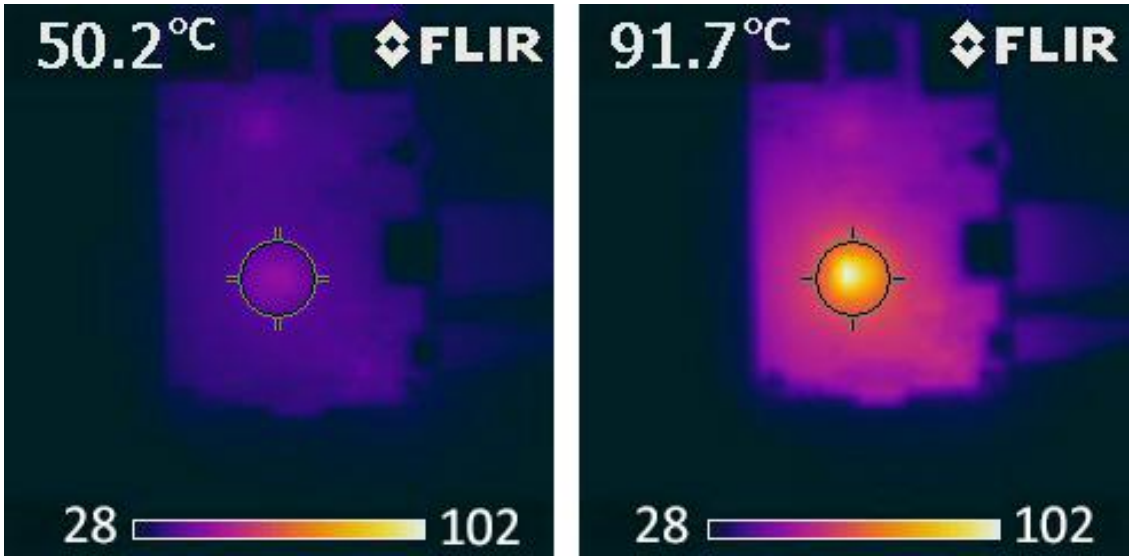
Kuva 7. Gardena-lomakastelusarjan sisältö (Gardena 2019).

Kuva 8. Gardena-lomakastelusarja asennettuna (ManoMano 2019).

3.4 Kotelointi

3.4.1 Suunnittelu

Yleensä helpointa on tilata valmis kotelo internetistä käyttötarkoituksen mukaan. Mutta koska tässä kokoonpanossa oli mukana myös erillinen relekortti, valmista sopivankokoista koteloa ei näyttänyt olevan verkossa tarjolla. Oman kotelon 3D-tulostus tuntui järkevältä vaihtoehdolta, jotta lopullinen kokoonpanon koko ei kasvaisi liian isoksi. Oman kotelon suunnittelu vie toki aikansa, mutta jos käytössä on mallinnusohjelma, johon voidaan tuoda myös koteloitavat komponentit, voidaan olla aika varmoja, että kotelo tulee kerralla valmiiksi. Koteloa suunnitellessa on hyvä kiinnittää huomiota etenkin ilmanvaihtoon kotelon sisällä. Mikäli Raspberry Pi 3:n prosessori saavuttaa 85 °C:n lämpötilan, laite suojaa itseään pudottamalla suorituskykyä tai sammuttamalla itsensä. Lämpökamerakuvasta nähdään (kuva 9), että piirilevyn komponenteista juuri prosessori kuumenee eniten rasituksessa.



Kuva 9. Vasemmalla lämpökamerakuva Raspberry Pi 3:sta lepotilassa ja oikealla tekemässä intensiivistä laskentatyötä (Denkel & Persaud 2018).

Raspberry Pi:n peruskäytössä ei pitäisi tulla ongelmia ylikuumenemisen vuoksi, mutta lopullinen vastuu jäähdytyksen riittävyydestä jää kuitenkin käyttäjälle. Valmistajan mukaan Raspberry Pi:n pitäisi toimia ongelmitta 0 °C – 70 °C:n lämpötiloissa. (Raspberry

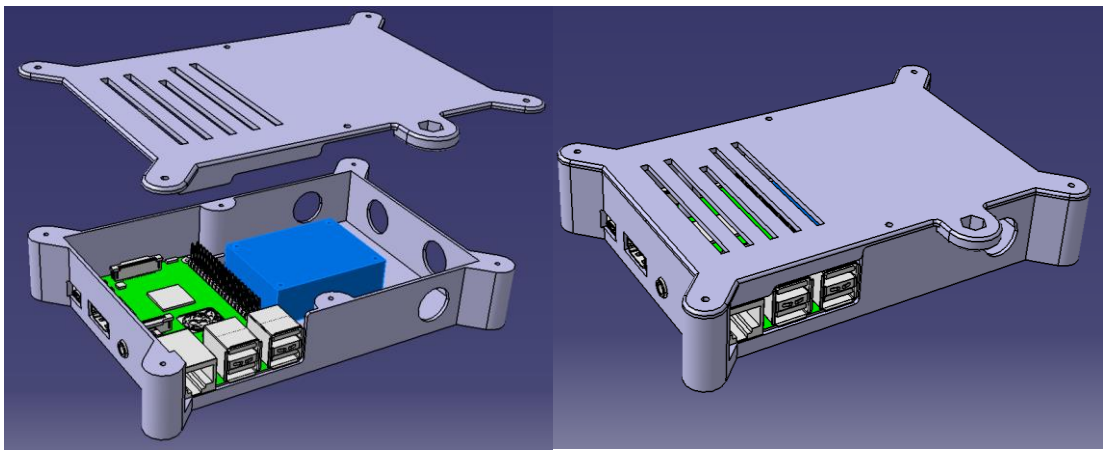
Pi Foundation 2019b). Jos lämpötilaa halutaan seurata käytön aikana, saadaan prosessorin lämpötila helposti esille komentoriviltä komennolla (esimerkkikoodi 1).

```
vcgencmd measure_temp
```

Esimerkkikoodi 1. Komentorivikomento, jolla saadaan selville Raspberry Pi:n prosessorin lämpötila.

3.4.2 Mallinnus ja 3D-tulostus

Kotelon mallinnus tehtiin käyttäen Catia v5 -mallinnusohjelmaa. Kotelomallin suunnittelua ohjaavia asioita olivat Raspberry Pi:n ja relekortin ulkomitat. Tärkeää oli myös saada verkkojännitteisille johdoille omat vedonpoistajat, jotta rasian sisällä oleviin liitäntöihin ei kohdistuisi vetoa tai työntöä (kuva 10).

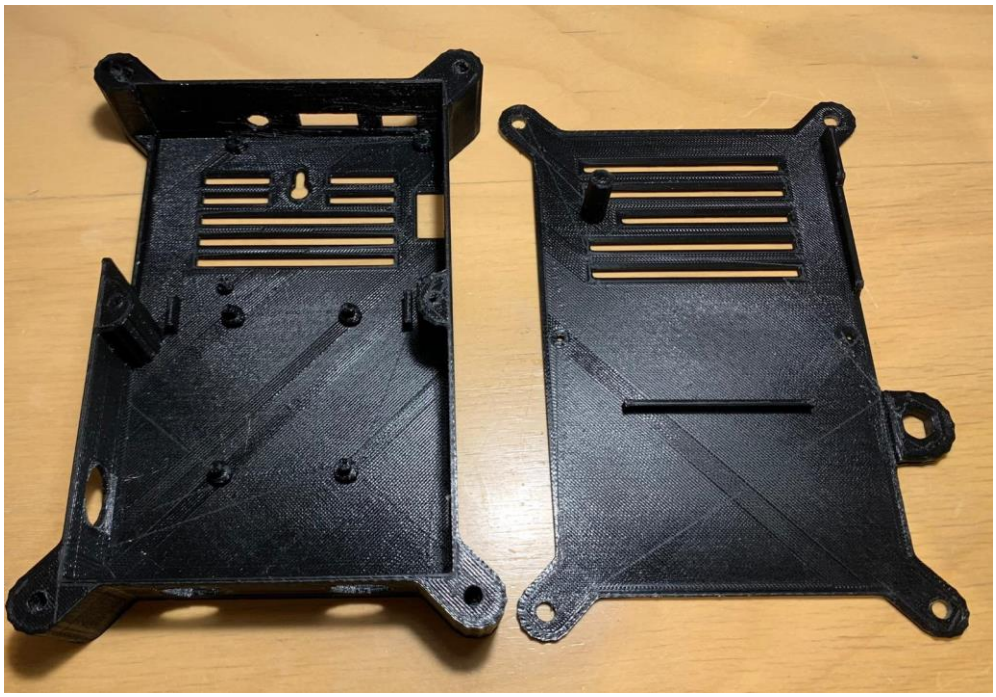


Kuva 10. Kokoonpanokuva kotelosta kannen ollessa poissa ja paikoillaan.

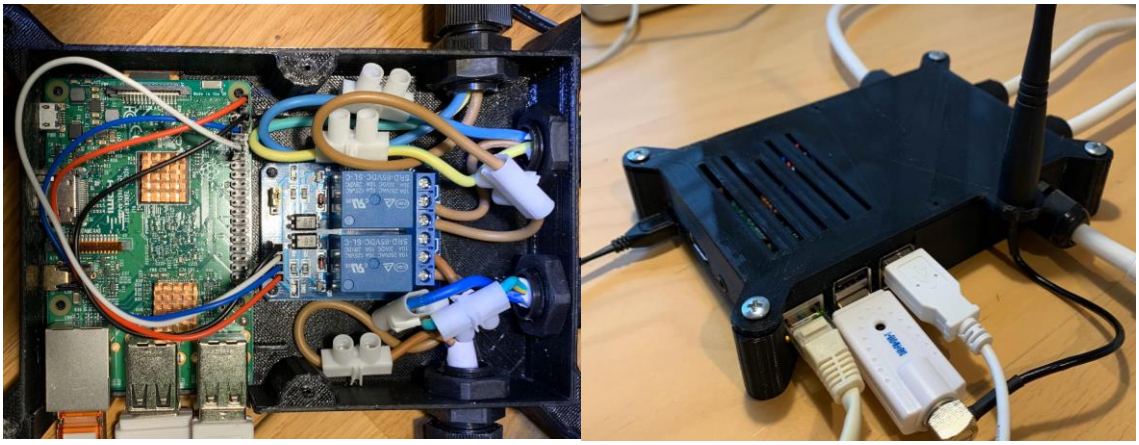
Kun mallinnettu kotelon malli todettiin valmiiksi, tallennettiin se stl-muodossa muistitikulle. Tämän jälkeen stl-tiedosto vietiin valmisteltavaksi Cura-ohjelmistoon, jossa määriteltiin 3D-tulostuksen asetukset käytettävälle tulostimelle.

Huomionarvoista tulostusasetuksia miettiessä on, sisältääkö malli reikiä tai onkaloita, joiden tulostaminen ilman tukimateriaalia olisi mahdotonta. Myös tulostusaikaan voidaan vaikuttaa merkittävästi mm. yhden muovikerroksen paksuuden säädöllä ja täyttöasteen määrällä. Asetukset ovat kuitenkin aina tulostinkohtaisia, ja niihin olisi syytä perehtyä ennen tulostuksen aloittamista.

Tässä työssä tulostimena käytetiin Ultimaker B.V:n valmistamaa Ultimaker 3 -tulostinta, ja tulostusajaksi kannen ja alaosan kanssa tuli noin 9 h. Tulostusasetuksina käytetiin 50 %:n täyttöastetta ja yhden kerroksen tulostuspaksuutena suurinta mahdollista (0,2 mm) ja tulostusmateriaalina PLA-muovia (kuvat 11 ja 12). Hinnaksi kotelon 3D-tulostukselle Helsingissä Pasilan kirjastossa tuli yhteensä 1,4 €.



Kuva 11. PLA-muovilla 3D-tulostettu kotelo Raspberry Pi:lle ja relekortille.



Kuva 12. Valmis kotelo kansi irrotettuna ja kannen kanssa kokoonpantuna.

3.5 Sensorit

Tiedonsiirtoa varten Raspberry Pi 3 sisältää integroituna Bluetoothin, Wi-Fi-yhteyden, Ethernet-portin ja 4 kpl USB-portteja sekä 40 kpl erilaisia GPIO-pinnejä. Nämä kaikki mahdollistavat erilaisten digitaalisen protokollien käyttämisen signaalien käsittelyssä. Raspberry Pi:n huonona puolena voidaan liitäntöjen osalta pitää sitä, että siitä puuttuu analogisen signaalin luku- ja lähettämismahdollisuus. Tämä voidaan kuitenkin tehdä mahdolliseksi, mikäli Raspberry Pi:hin lisätään erillinen A/D-muunnin, joka muuttaa esimerkiksi jännitteen vaihtelut digitaalisiksi arvoiksi. Tässä esimerkkityössä luetaan sensoreita käyttäen Bluetoothia, USB-porttia ja 433 MHz:n taajuutta.

3.5.1 Hiilidioksidimittari 9050

Hiilidioksidimittari 9050 on Suomen Lämpömittari Oy:n myymä hiilidioksidimittari, joka mittaa hiilidioksidipitoisuuden lisäksi myös lämpötilan ja kosteuden (kuva 13). Mittari sisältää ison näytön mitatuille lukemille ja LED-merkkivalot, jotka kuvaavat sisäilman hiilidioksiditasoa. Myös hälytysäänen mykistysnappi löytyy.

Kuva 13. Hiilidioksidimittari 9050 (Suomen Lämpömittari Oy 2019).

Tarkempi tutustuminen tuotteeseen ostamisen jälkeen paljasti hiilidioksidimittarin olevan taiwanilaisen ZyAura-yhtiön valmistama ZG1683R-hiilidioksidimittari (ZyAura 2019).

ZyAuran verkkosivuilta löytyy linkki, josta on ladattavissa Windowsille oma ohjelmisto, jossa mittausarvoja voidaan säätää, tallentaa ja monitoroida. Tästä syntyi ajatus siitä, voisiko mittarille olla olemassa myös Linux-pohjainen ohjelma, jolla mittausarvoja voitaisiin kerätä Raspberry Pi:n avulla. Ohjelmaratkaisusta lisää luvussa 3.6 Ohjelmat laitteille ja sensoreille.

3.5.2 Lämpö- ja kosteussensori AcuRite 06002M

Langaton AcuRite 06002M -sensori on yhdysvaltalaisen Chaney Instruments -yhtiön valmistama kosteus- ja lämpötilasensori (kuva 14). Sensori valittiin tähän työhön, koska sitä suositeltiin useilla internetin keskustelupalstoilla toiminnaltaan luotettavaksi yhdessä RTL-SDR-vastanottimen kanssa. Varmuutta toimivuuteen tuovat mm. säänkestäväksi tehty kotelointi, muuttumaton tunnistuskoodi ja laaja lämpötilan mittausalue (-40 °C – 70 °C), tiheä signaalin lähetys (3 kertaa 15 sekunnin välein) ja helposti vaihdettavat AA-paristot (2 kpl).

Kuva 14. AcuRite 06002M -lämpötila- ja kosteussensori (AcuRite 2019).

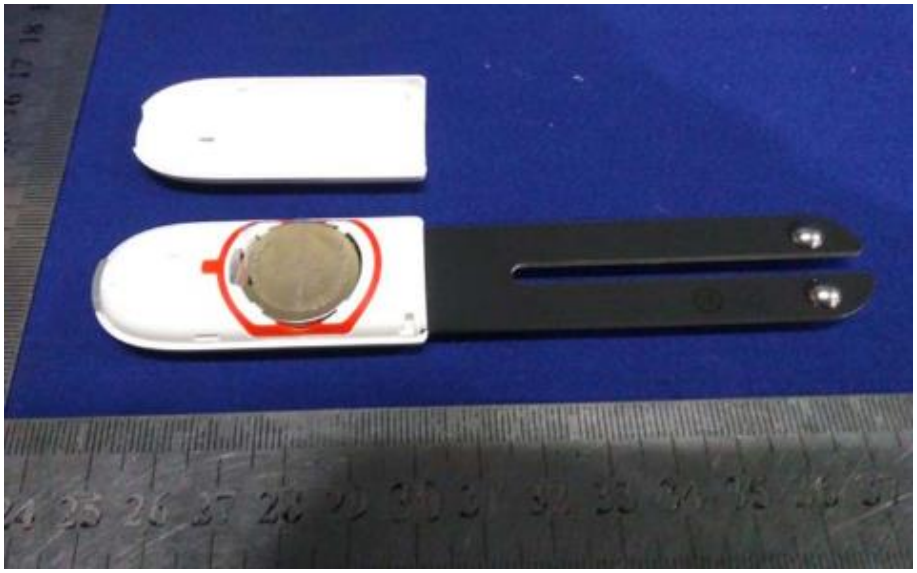
Valmistaja myös lupaa sensorin toimintaetäisyydeksi 100 m riippuen esteistä vastaanottimen ja sensorin välillä (AcuRite 2019). Projektia tehdessä kyseistä sensoria ei ollut saatavilla Euroopasta, joten niitä tilattiin Amazon-verkkokaupasta hinnan ollessa n. 12 €/kpl.

3.5.3 HHCC Flower Care -sensori

Kasvien kuntoa monitoroimaan ostettiin kiinalaisen HHCC Plant Technology -yhtiön valmistamia Flower Care -sensoreita (kuva 15). Sensorista mielenkiintoisen tekee se, että se pystyy mittamaan

- valaistusvoimakkuuden (lx)
- lämpötilan (°C)
- kasvualustan EC-arvon eli sähkönjohtavuusarvon (µs/cm)
- kasvualustan kosteusprosentin perustuen kapasitiiviseen mittaukseen.

Lisäksi se toimii langattomasti käyttäen Bluetooth-yhteyttä, ja virtalähteenä toimii CR2032-nappiparisto. Sensoria käytetään siten, että se upotetaan kasviruukun multaan sopivalle syvyydelle (HHCC Plant Technology 2016). Hinta yhtä sensoria kohti oli noin 14 €. Sensoreita on tarjolla useimmissa suosituissa kansainvälisissä elektroniikkaa myyvissä verkkokaupoissa, esimerkiksi Amazonissa ja Banggoodissa.



Kuva 15. Flower Care -sensori (HHCC Plant Technology 2016).

Sensori on suunniteltu toimimaan yhdessä Flower Care -sovelluksen kanssa, joka on ladattavista Applen App Storesta ja Google Play Storesta. Sensorille on myös kirjoitettu avoimen lähdekoodin ohjelma, jonka avulla sensoria voidaan käyttää myös Linux-ympäristössä ilman Flower Care -sovellusta.

3.6 Ohjelmat laitteille ja sensoreille

3.6.1 Raspberry Pi 3 B

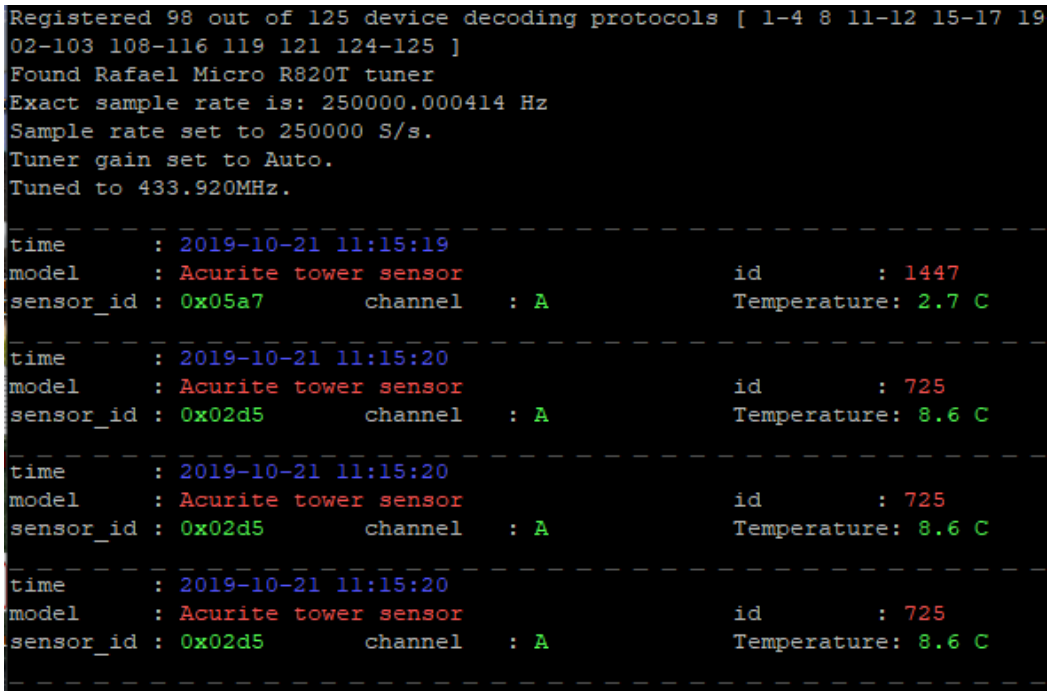
Projektissa käytettiin Raspberry Pi 3:n käyttöjärjestelmänä Raspbian Jessie 4.9.35 -versiota. Tämä siksi, että osa järjestelmän toiminnoista on toteutettu LabVIEW-ohjelmointiympäristössä ja ainoastaan Raspbian Jessie -käyttöjärjestelmä tukee tätä ohjelmointitapaa. Lisätietoa Raspberry Pi:n ja LabVIEWin yhteensopivuudesta löytyy National Instrumentsin ylläpitämiltä MakerHub-sivuilta. (LabVIEW MakerHub 2016a.)

Sensoreiden lukua varten käytettiin Python-tulkeiksi 2.7.9- ja 3.4.2 -versioita. Näiden pitäisi löytyä valmiiksi asennettuna levykuvatiedostosta, jos käyttöjärjestelmänä käytetään Raspberry Pi Foundationin virallisesti tukemaa Raspbian-käyttöjärjestelmää.

3.6.2 RTL_443

Yksinkertaisuudessaan RTL_433 on avoimen lähdekoodin purkuohjelma, joka toimii yhdessä RTL-SDR-radiovastaanottimen kanssa. Se purkaa 433 MHz:n taajuudelta tulevat radiosignaalit luettavaan muotoon (kuva 16) ennalta määrättyjen sääntöjen pohjalta. Se toimii MacOS-, Windows- ja Linux-käyttöjärjestelmissä. Ohjelman hyviksi puoliksi voidaan mainita mm.

- matala prosessorin kuormitus
- yli sata tuettua laiteprotokollaa
- käyttövalmius heti asennuksen jälkeen
- datan edelleenohjaus mm. JSON-, CSV-, MQTT- ja UDP-formaateissa
- käyttäjäyhteisön jatkuva kehitystyö.



Kuva 16. RTL_433-ohjelman testaaminen komentoriviltä. Kuvassa ohjelma on purkanut vastaanotetut AcuRite-sensorin lähettämät radiosignaalit luettavaan muotoon.

Ohjelman latauslinkki ja tarkempi dokumentointi ovat saatavissa Microsoftin omistamalta GitHub-verkkosivustolta, jossa voi myös ottaa kantaa ja osallistua itse ohjelman kehittämiseen (RTL_433 2019).

Tässä projektissa AcuRite-sensoreilta saadut mittausarvot ohjattiin komentoriviltä JSON-formaatissa Syslog-protokollaa käyttäen paikalliseen porttiin käynnistyskäskyllä (esimerkkikoodi 2).

```
rtl_433 -R 40 -F json -F syslog:192.168.1.227:8001
```

Esimerkkikoodi 2. RTL_433-ohjelman käynnistyskäsky, joka kirjoitetaan komentoriville.

Tässä esimerkkikoodissa ”rtl_433” on käynnistyskäsky, ”-R 40” rajaa signaalin vain AcuRite-sensoreihin ja ”-F json -F syslog:192.168.1.227:8001” määrittää formaatiksi JSON:n, joka lähetetään Syslog-protokollaa käyttäen osoitteeseen 192.168.1.227:8001, joka oli tässä tapauksessa Raspberry Pi:n oma staattiseksi määritetty IP-osoite.

Tämän jälkeen RTL_433-ohjelman lähettämä data on vastaanotettavissa sitä mukaa, kun ohjelma kääntää radiosignaaleja. On hyvä tietää, että Syslog käyttää UDP-protokollaa, mikä tarkoittaa, että lähetettäviin viesteihin ei saada kuittauksia eikä uudelleen lähettämistä suoriteta. UDP-protokolla on tässä mielessä epäluotettava, mutta hyvin nopea tapa siirtää tietoa, koska ylimääräisiä prosesseja ei ole (Helsingin yliopisto 2009). Jos halutaan varmistua siitä, että viestit tavoittavat vastaanottajan, tulisikin käyttää MQTT-yhteyttä ohjelmien väliseen kommunikointiin.

3.6.3 Mi Flora Plant Sensor MQTT Client/Daemon

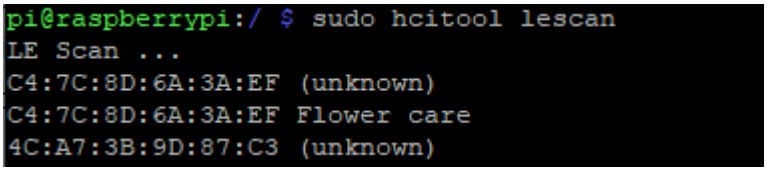
Mi Flora Plant Sensor MQTT Client/Daemon on avoimen lähdekoodin Python-skripti, jolla Mi Flora -sensoreiden mittausdataa luetaan Raspberry Pi:n Bluetooth-moduulin avulla (Mi Flora Plant Sensor MQTT Client/Daemon 2019).

Kun ohjelma on asennettu Raspberry Pi:lle GitHubissa olevien ohjeiden mukaisesti, aloitetaan sensoreiden määrittäminen asetuksiin. Jokaisella Mi Flora -sensorilla on yksilöllinen MAC-osoite (kuva 17), joka saadaan selville käyttämällä Bluetooth-laitteiden etsimiseen käytettävää komentoa (esimerkkikoodi 3).

```
sudo hcitool lescan
```

Esimerkkikoodi 3. Komentoriville kirjoitettava komento Bluetooth-laitteiden etsimiseen.

Sensorit kannattaa määrittää asettamalla paristot sensoreihin yksitellen, jotta voidaan olla varmoja siitä, mihin sensoriin löydetty MAC-osoite liittyy.



Kuva 17. Raspberry Pi on löytänyt yhden Flower care -sensorin.

Kun Mi Floran sensorit on kartoitettu, ne kirjoitetaan muistiin ohjelman config.ini-tiedostoon sensoriotsikon alapuolelle tiedoston oman esimerkin mukaisesti. Tämän jälkeen

muokataan itse pääohjelman (miflora-mqtt-daemon.py) Python-skriptiä UPD-yhteyttä varten. Yhteys muodostetaan lisäämällä skriptiin 5 riviä (esimerkkikoodi 4).

```
import socket

UDP_IP_ADDRESS = "localhost"
UDP_PORT_NO = 8002

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
```

Esimerkkikoodi 4. UDP-yhteyden muodostamista varten tarvittavat rivit, joissa määritellään vastaanottajan IP-osoite, vastaanottoportti ja viestin sisältö (Python Software Foundation 2013).

Lisättyjä rivejä muokataan vielä siten, että Message-kohta poistetaan ja siihen lisätään tiedot, joita halutaan lähettää UDP-yhteyttä pitkin. Jotta oikeat tiedot osataan lisätä Message-kohtaan, pitäisi käsiteltävän Python-skriptin toimintaperiaatetta tuntea siinä määrin, että halutut muuttajat osataan poimia Message-kohtaan. Usein Python-skriptit tulostavat mittausdatan suoraan terminaali-ikkunaan, ja täten voidaan pienellä tutkimustyöllä löytää Pythonin print-käsky, joka johtaa oikeiden muuttujien löytämiseen. Tässä työssä Message-kohta (esitetty esimerkkikoodissa 4) muokattiin seuraavanlaiseen muotoon (esimerkkikoodi 5).

```
sock.sendto(bytes('BT_sensor "{}": {}'.format(flora_name, json.dumps(data)),
"utf-8"), (UDP_IP_ADDRESS, UDP_PORT_NO))
```

Esimerkkikoodi 5. Message-kohtaan on muokattu halutut muuttujat viestin lähettämistä varten.

UDP-yhteyden kautta vastaanotettu viesti näyttää seuraavanlaiselta (esimerkkikoodi 6).

```
BT_sensor "id_9": {"mac": "C4:7C:8D:6B:07:23", "battery": 98, "temperature":
4.1, "firmware": "3.2.1", "moisture": 48, "timestamp": "2019-10-28 12:16:41",
"name": "id_9", "name_pretty": "id_9", "light": 2274, "conductivity": 220}
```

Esimerkkikoodi 6. UDP-yhteyden kautta vastaanotettu viesti, joka sisältää sensorin lähettämät tiedot.

Tämän jälkeen viestin sisällön jatkokäsittely voidaan aloittaa toisessa ohjelmassa. Tässä esimerkissä viestin jatkokäsittely tehtiin LabVIEW-ympäristössä, josta tarkemmin luvussa 3.7.1 LabVIEW.

3.6.4 CO2Meter

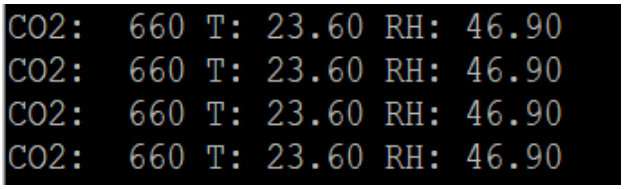
CO2Meter-ohjelmakoodi perustuu Henryk Plötzin projektiin, jossa hiilidioksidimittarin lähettämä data käännetään ihmissilmin luettavaan muotoon Python-skriptin avulla. Mittausdata otetaan vastaan suoraan Raspberry Pi:n USB-portista. (Plötz 2015.)

Ohjelma käynnistetään komentoriviltä kahdella erillisellä komennolla (esimerkkikoodi 7).

```
sudo chmod a+rw /dev/hidraw0
python zytemp.py /dev/hidraw0
```

Esimerkkikoodi 7. Hiilidioksidimittarin käynnistyskomennot. Ensimmäisellä rivillä määritellään käyttöoikeudet laitteen käsittelyyn ja seuraava käynnistää Python-skriptin.

Mikäli ohjelman suoritus onnistuu, pitäisi terminaaliin alkaa tulostua jatkuvalla syötöllä hiilidioksidimittarilta käännettyjä mittausarvoja (kuva 18).



Kuva 18. Python-ohjelman kääntämiä mittausarvoja tulostuneena komentoikkunaan.

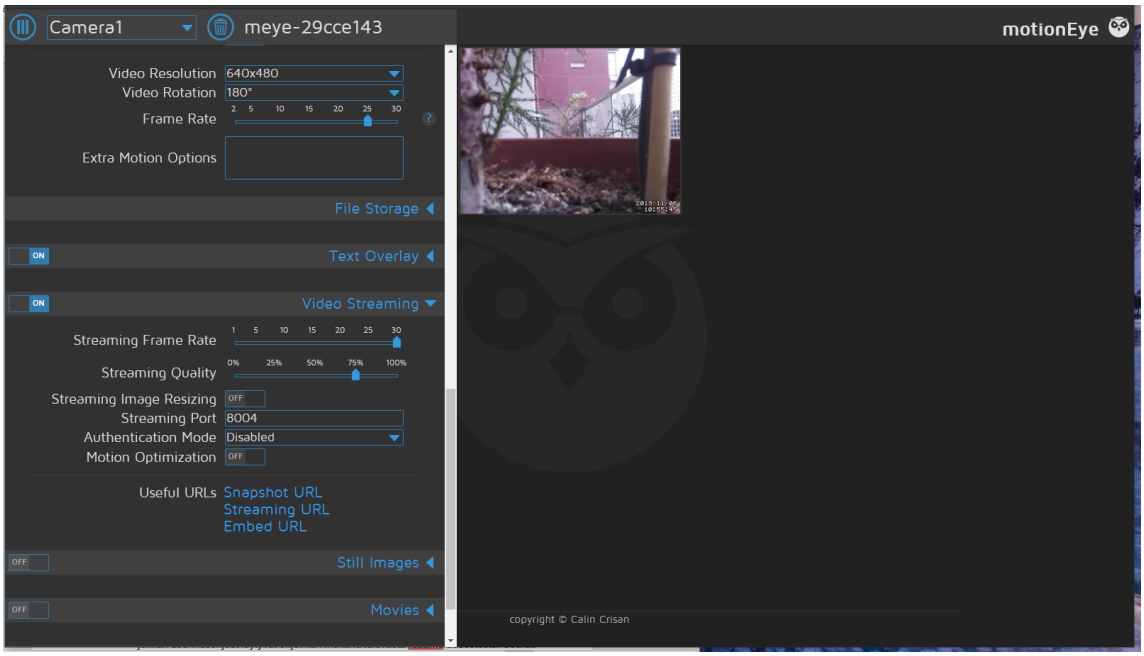
Tämän jälkeen voidaan aloittaa Python-skriptin muokkaaminen UDP-yhteyttä varten. Alkuperäisestä skriptistä puuttui aikaleima, joten se on syytä lisätä mukaan UDP-viestiin, jotta tiedetään jatkossa, milloin mittausarvo on lähetetty ohjelmasta. UDP-yhteys tehdään samojen periaatteiden mukaan kuin edellä Mi Flora Plant Sensor -ohjelmassa tehtiin, paitsi että vastaanottoportiksi määriteltiin eri portti (8003).

3.6.5 MotionEyeOS

MotionEyeOS on avoimen lähdekoodin ohjelma, joka muuttaa Raspberry Pi:n videovalvontajärjestelmäksi. Sen avulla voidaan mm. vastaanottaa, säätää ja tallentaa videokuvaa eri lähteistä. Ohjelman keveyden vuoksi se sopii hyvin juuri yhden piirilevyn tietokoneille, ja ohjelman käyttöliittymään päästään käsiksi tavallisella verkkoselaimella. (Crisan 2017.)

Tässä esimerkissä Raspberry Pi 3 B+ -tietokoneesta tehtiin web-kamera lisäämällä kokoonpanoon 8.0 Mpix v2 -kameramoduuli, joka lähettää jatkuvaa videokuvaa kasveista MotionEyeOS-ohjelman avulla. Videon välityksellä voitiin havainnoida yleisesti kasvien kuntoa ja seurata kasvien kastelutapahtumaa etänä samalla kun pumppu kytkettiin päälle.

Ohjelman käyttöönotto toteutetaan asentamalla levynkuvatiedosto tyhjälle SD-kortille. Kun SD-kortti asetetaan Raspberry Pi:hin ja laitteeseen kytketään virta, päästään verkkoselaimella kirjautumaan valvontajärjestelmän käyttöliittymään (kuva 19), kun yhteys otetaan Raspberry Pi:n IP-osoitteeseen.



Kuva 19. MotionEyeOS-ohjelman käyttöliittymä avattuna verkkoselaimella asetusten määrittämistä varten.

Suoratoistoa varten asetuksia muutettiin siten, että Video Streaming kytkettiin asetuksista päälle ja aikaleima asetettiin näkymään kuvan oikeaan reunaan. Video Streaming -otsikon alta löytyy suora URL-linkki videokuvaan, joka alkaa toimimaan sen jälkeen, kun uudet asetukset on otettu käyttöön.

3.6.6 Dyfi-update

Dyfi-update on IP-osoitteen päivitysohjelma, joka käyttää hyväksi Dy.fi-sivuston tarjoamaa ilmaista dynaamista DNS-palvelua. Ohjelma toimii siten, että se ilmoittaa aika ajoin Dy.fi-palvelimelle oman julkisen IP-osoitteen. Kun Dy.fi-sivustolta on varattu oma domain-nimi ja Dyfi-update-ohjelma on lähettänyt IP-osoitteen palvelimelle, päästään Raspberry Pi:lle suoraan käyttämällä Dy.fi-palvelusta varattua domain-nimeä. (Dy.fi 2019.)

Koska projektissa haluttiin ottaa yhteys Raspberry Pi:hin myös kodin ulkopuolelta ja IP-osoite vaihtuu internet-operaattorin toimesta satunnaisesti, oli jonkin DNS-palvelun käyttöönotto pakollista. Dyfi-update sopi hyvin tämän projektin tarpeisiin, koska se oli helposti asennettavissa Raspberry Pi:lle. Ohjelma on ladattavissa Dy.fi-sivustolta ja ohjelman asennusohjeet löytyvät ohjelman mukana tulevasta readme-tiedostosta. Seuraavaksi esitetään nopea asennusohje olettaen, että ohjelma halutaan asentaa Raspberry Pi:lle komentoriviltä käsin (esimerkkikoodi 8).

```
wget http://www.dy.fi/files/dyfi-update-pl-1.2.0.tar.gz
tar zxvf dyfi-update-pl-1.2.0.tar.gz
cd dyfi-update-pl-1.2.0
sudo make install
sudo make installboot2
```

Esimerkkikoodi 8. Dyfi-updaten asennus Raspberry Pi-tietokoneelle komentoriviltä käsin. Jokainen rivi on kirjoitettava komentoriville erikseen.

Kun asennus on suoritettu, täytyy lopuksi vielä käydä muokkaamassa ohjelman asetustiedostoa, johon asetetaan Dy.fi-palvelimelle tarvittavat tunnukset ja oma domain-nimi (tarkemmat ohjeet muokattavassa tiedostossa). Asetustiedostoon päästään käsiksi komennolla (esimerkkikoodi 9).

```
sudo nano /usr/local/etc/dyfi-update.conf
```

Esimerkkikoodi 9. Komento, jolla päästään käsiksi Dyfi-update-ohjelman asetustiedostoon.

Mikäli asennus ja asetusten määrittäminen onnistui, pitäisi Dydi-update-ohjelman käynnistyä automaattisesti, kun Raspberry Pi:hin kytketään virta. Myös varatun domain-nimen pitäisi alkaa heti toimimaan. Sivustolta www.dy.fi voidaan myös tarkastaa, onko ohjelma ottanut yhteyttä palvelimeen. Jos ohjelma ei ole ottanut palvelimeen yhteyttä yli 7 päivään, poistuu varattu domain-nimi pois käytöstä, eikä se enää ohjaa oikeaan IP-osoitteeseen.

3.7 Ohjelmat datan käsittelyyn

3.7.1 LabVIEW

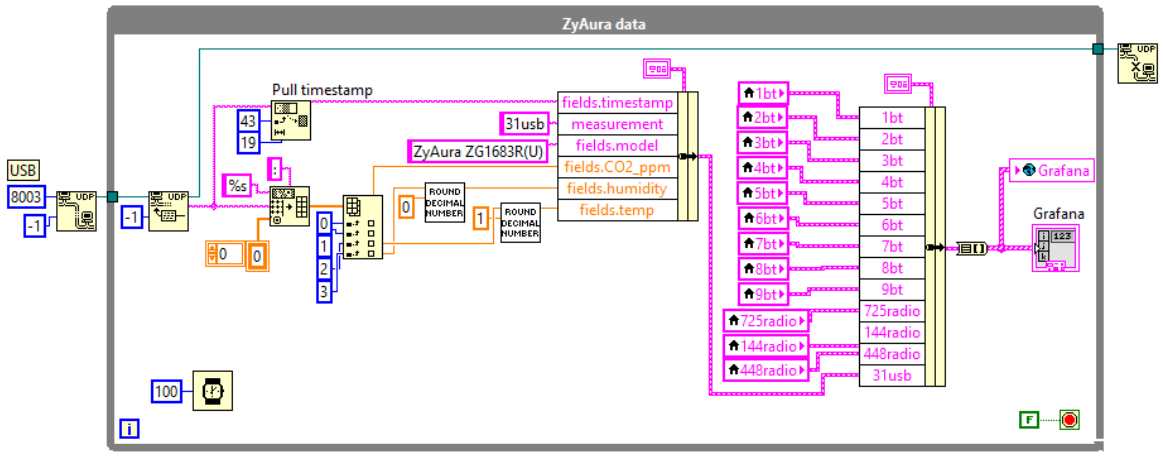
LabVIEWin on valmistanut National Instruments. Se on graafinen ohjelmointikieli. Ohjelmalla voi lukea ja analysoida mittaus- ja testausdataa. Sillä voi myös suunnitella koneita, teollisuuslaitteita ja langattomia tietojärjestelmiä.

LabVIEW-ohjelmaa käytettiin tässä esimerkissä ohjelmana, jolla kootaan kaikilta sensoreilta saadut mittausarvot. Tämä mahdollisti selkeän pääsyn kaikkien sensoreiden mittausarvojen hallintaan yhden ohjelman kautta, jossa voitiin mm.

- kalibroida, säätää ja pyöristää mittausarvoja
- luoda kasvien kasteluohjelma
- hallita Python-ohjelmia
- selvittää oma julkinen IP-osoite
- hallita vikatilanteita (esim. automaattinen järjestelmän uudelleenkäynnistys Python-ohjelman pysähtyessä virheviestiin).

Jotta LabVIEW saadaan toimimaan yhdessä Raspberry Pi:n kanssa, on asennettava oikeat ohjelmaversiot ja lisäosat. Ohjeet näiden asentamisesta löytyvät LabVIEW MakerHub -sivustolta. (LabVIEW MakerHub 2016a.)

Projektia varten tehty ohjelma sisälsi useita eri ohjelmarakenteita, mutta tässä esitellään niistä vain tärkeimmät. Tärkeimmässä osassa oli mm. UDP-yhteyden kautta saadut mittausarvot, joiden hallita tapahtui yksinkertaisimmillaan kuvan 20 mukaisesti. Kuvassa ohjelmakoodi liikkuu vasemmalta oikealle siten, että sensorilta saatu viesti tulee LabVIEWin while-silmukkaan portista 8003. Seuraavaksi viesti puretaan string-muotoon, ja tämän jälkeen kaksoispisteellä erotellut muuttujien arvot ohjataan sarakkeeseen. Edelleen käsittelemällä sarakkeen arvoja päästään käsiksi yksittäisiin arvoihin. Kuvan esimerkissä arvoja vain pyöristettiin ja määriteltiin sensoria yksilöivä tunniste (31usb). Lopuksi kaikki sensorit yhdistettiin samaan sarakkeeseen, josta ohjelmakoodin viesti ohjattiin Web Service -toiminnon julkaistavaksi.



Kuva 20. Esimerkkikoodi hiilidioksidimittarin keräämän mittausdatan hallinnasta LabVIEW-ympäristössä. Viesti otettiin vastaan vasemmalla, minkä jälkeen se käsiteltiin ja lähetettiin eteenpäin.

Web Service -toiminnon tuloksena saatiin rajapinta, joka tarjoaa JSON-syötettä. Rajapinnan avulla LabVIEWin prosessoimaa tietoa voidaan käyttää muissa ohjelmissa. JSON-syöte päivittyy sitä mukaa, kun LabVIEW prosessoi uusia sensoriarvoja.

Syöte on muodostettu JSON-standardin mukaan. Jos syöte muodostetaan väärin, ei siihen yhdistävä ohjelma pysty tulkitsemaan syötteen tarjoamaa tietoa. Syöte muodostuu hakasuluista, aaltosulkeista, pilkuista, kaksoispisteistä, numeroista ja tekstistä. Hakasuluilla määritellään sarakkeet ja aaltosulkeilla yksittäiset oliot. Pilkut erottelevat olion sisällä olevat avain-arvoparit. Avain-arvoparit sen sijaan erotellaan kaksoispisteellä sen avaimesta ja arvosta. Esim. ”{"temperature_C":20.4}”-oliossa ”temperature_C” on olion avain ja ”20.4” sen arvo.

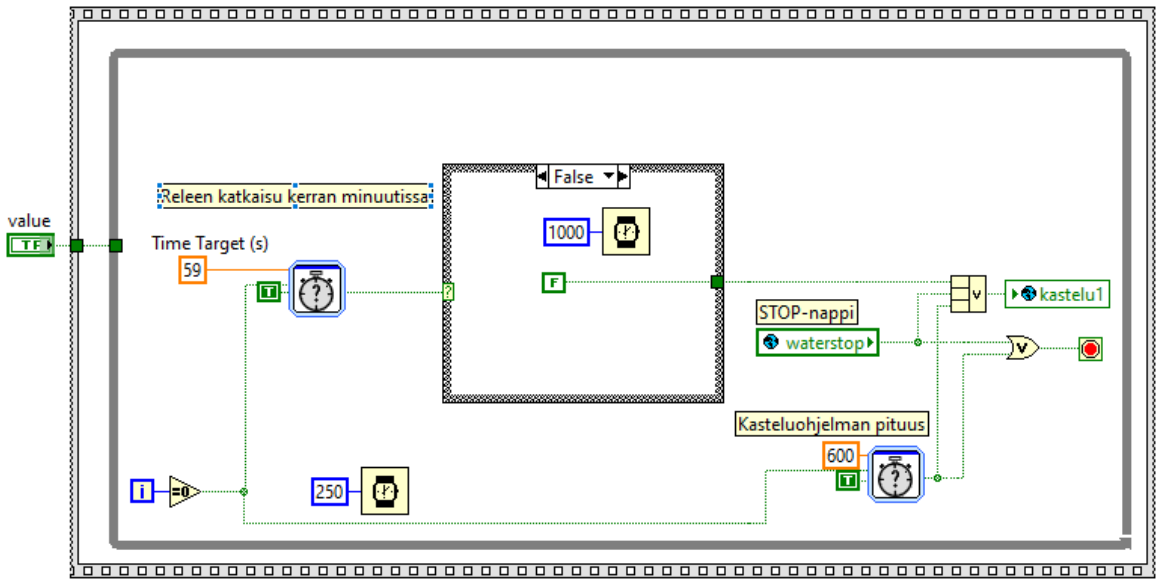
Tässä projektissa LabVIEWillä tehtiin kaksi JSON-syötettä, joista toinen tehtiin Node-REDiä varten (liite 1). Toinen JSON-syöte sisälsi kaiken tiedon LabVIEWiin luoduista toiminnoista, kuten kastelutoiminnoista, video-URL:sta sekä reboot- ja stop-napista (liite 2), jota käytettiin tämän työn ulkopuolelle rajatussa Kaste 2.0 -ohjelmassa (Tampio 2019).

Jotta Web Servicen julkaisemaa JSON-syötettä päästään käsittelemään Node-REDissä halutulla tavalla, on se muodostettava käytettävän tietokantaohjelman tarpeiden

mukaisesti. Tässä projektissa käytettiin InfluxDB-tietokantaohjelmaa, jonka käytöstä tarkemmin luvussa 3.7.3 InfluxDB.

Kasteluohjelmien luominen

Kasvien kasteluohjelmat toteutettiin LabVIEW-ympäristössä (kuva 21) siten, että Web servicen kautta tuleva boolean-viesti käynnistää while-silmukan, kunnes ajastin pysäyttää silmukan tai stop-nappia painetaan. Gardena-lomakastelusarjan muuntajaan on kiinteästi asetettu ajastin, joka ohjaa laitteen käynnistymään minuutiksi kerran vuorokaudessa. Tämän vuoksi LabVIEWin kasteluohjelma tuli rakentaa siten, että muuntaja nollattiin releen avulla kerran minuutissa. Näin voitiin luoda pidempiä kasteluohjelmia, kuin Gardena-kastelusarjalla voitiin alun perin toteuttaa.



Kuva 21. LabVIEW-ohjelmointiympäristössä toteutettu kasteluohjelma, jossa Web Servicen kautta saapunut boolean-viesti käynnistää while-silmukan. Silmukan pyöriessä Raspberry Pi ohjaa vesipumppua releen avulla määritellyn ajan.

3.7.2 Node-RED

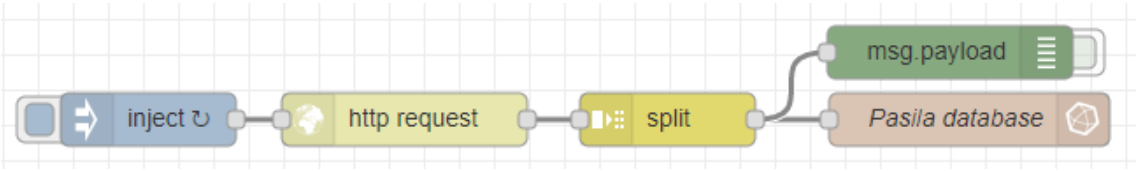
Node-RED on IBM:n kehittämä visuaalinen ohjelmointiympäristö, jossa ohjelmakoodia luodaan funktio-nodejen eli solmujen avulla. Jokaisella solmulla on ohjelmoinnissa oma tarkoin määritelty tarkoituksensa. Ohjelmakoodissa solmut ottavat vastaan liikkuvat viestit, käsittelevät ne määritellyllä tavalla ja lähettävät eteenpäin. Solmujen avulla saadaan esimerkiksi erilaiset protokollat yhdistettyä, ja näin voidaan luoda uusia sovellutuksia eri järjestelmien välille. Node-RED on saanut suuren suosion Raspberry Pi -tietokoneiden käyttäjien keskuudessa helpon, selkeän ja kevyen toiminnallisuutensa ansiosta. Lisäksi sen käyttö on avoimen lähdekoodin ohjelmien tapaan ilmaista. (Node-RED 2019a.)

Node-RED tulee Raspbian-käyttöjärjestelmissä valmiiksi asennettuna, ja sitä voidaan helposti laajentaa tarpeen mukaan erilaisilla lisämoduuleilla. Seuraavilla komennoilla päivitetään Node-RED ja siihen liittyvä runtime-ympäristö Node.js:n viimeisimpään versioon. Lisäksi asennetaan myöhemmin tarvittavat lisämoduulit Apple HomeKit -sovellusta, InfluxDB-tietokantaa ja valojen ajastamista varten. Määritellään myös Node-RED käynnistymään automaattisesti, kun Raspberryn Pi uudelleen käynnistetään (esimerkkikoodi 10).

```
update-nodejs-and-nodered
sudo apt-get install libavahi-compat-libdnssd-dev
node-red
ctrl + c (näppäinyhdistelmä)
cd ~/.node-red
npm install node-red-contrib-homekit
npm install node-red-contrib-influxdb
npm install node-red-contrib-bigtimer
sudo service nodered restart
sudo systemctl enable nodered.service
sudo service nodered start
```

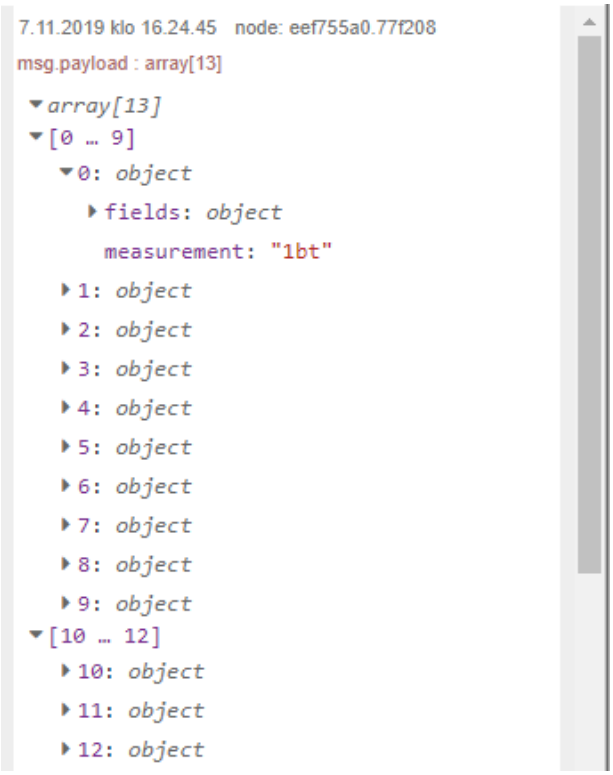
Esimerkkikoodi 10. Node-RED-ohjelman päivitys ja lisäosamoduulien asennus.

Tässä projektissa Node-REDiä käytettiin JSON-syötteen lukemiseen ja sen sisältämien arvojen tallentamiseen InfluxDB-tietokantaohjelmaan. JSON-syöte luotiin LabVIEWissä InfluxDB-solmun ohjeiden mukaisesti (Node-RED 2019b). Seuraavassa esimerkissä on esitetty, kuinka sensoreiden arvot tallennettiin tietokantaan. Samaan tyyliin kuin LabVIEW-ohjelmoinnissa, Node-REDissä ohjelmakoodin vaiheet liikkuvat aina vasemmalta oikealle (kuva 22).

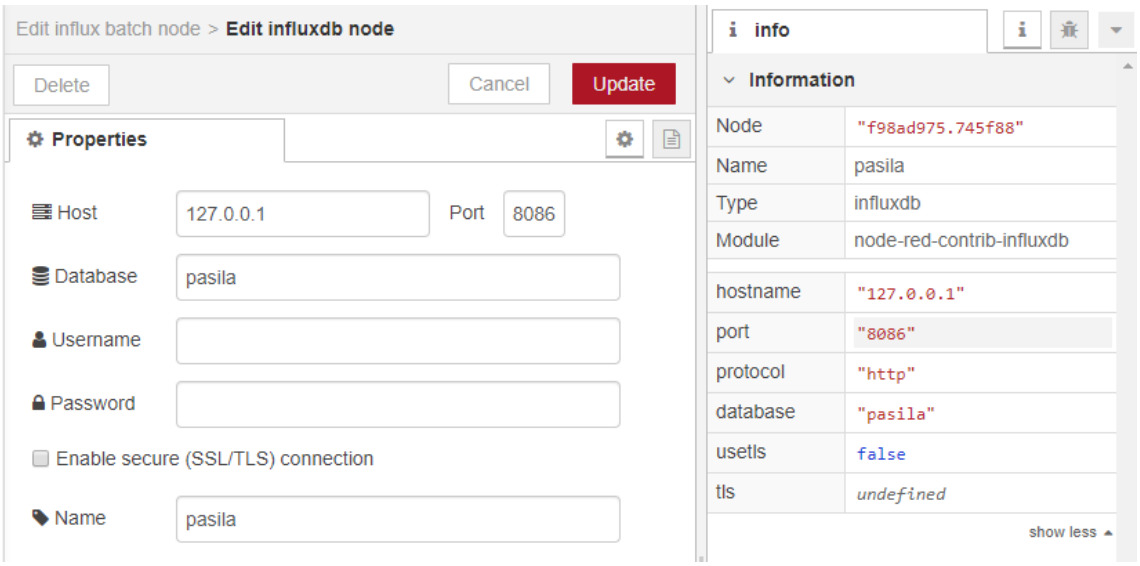


Kuva 22. Ohjelmakoodi, jossa Node-RED lukee JSON-syötteen ja tallentaa mittausarvot tietokantaan.

Alkaen vasemmanpuoleisimmasta solmusta, Inject-solmua käytettiin käynnistämään ohjelmakoodi kerran minuutissa. Ohjelmakoodin lähdettyä liikkeelle http request -solmu hakee LabVIEWin luomasta URL-osoitteesta JSON-syötteen. URL-osoitteena kannattaa käyttää paikallista IP-osoitetta, jotta yhteys pysyy yllä, vaikka internet-yhteys katkeaisikin. Http request -solmun jälkeen split-solmu pilkkoo JSON-syötteen yksittäisiksi objekteiksi, jotka vastaavat yksittäisiä sensoreita arvoineen (kuva 23). Lopussa näkyvän Pasila database -solmun asetukset määriteltiin (kuva 24) siten, että InfluxDB:n oletetaan olevan asennettuna samaan järjestelmään kuin Node-RED.



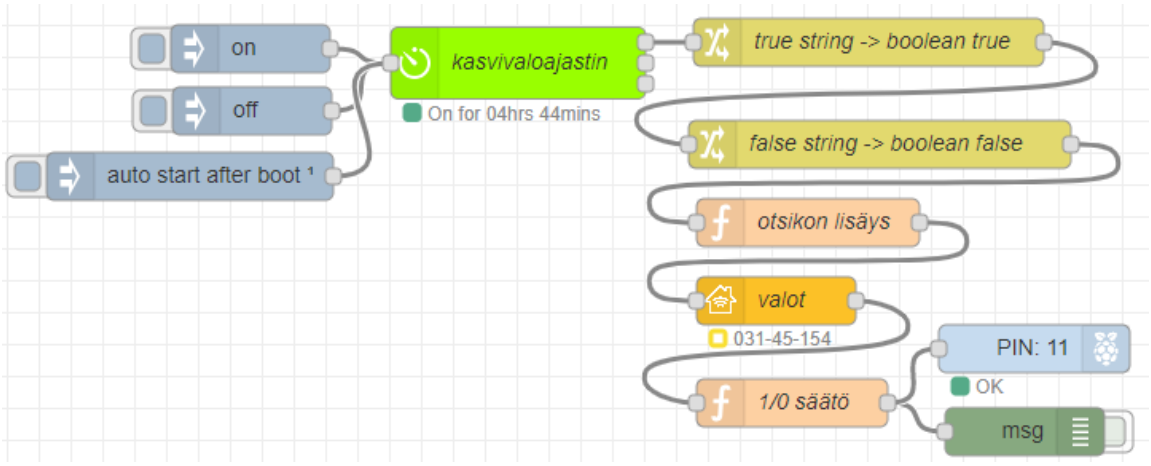
Kuva 23. Debug-solmulla tulostettu viestisisältö heti split-solmun jälkeen. Nähdään, että split-solmu on pilkkonut JSON-syötteen siten, että jokainen objekti vastaa eri sensoria. Tämä on valmis viesti ohjattavaksi suoraan InfuxDB-tietokantaan.



Kuva 24. InfluxDB-solmun asetukset määriteltynä siten, että se yhdistyy Rapberry Pi:lle asennettuun InfluxDB-tietokantaohjelmaan. Esimerkissä käytettiin pasila-nimistä tietokantaa.

Valojen ohjaus

Tässä työssä toteutettiin myös valojen ohjaus käyttäen Node-RED-ohjelmointia. Valojen ohjaaminen haluttiin toteuttaa siten, että valot toimisivat ajastetusti joka päivä, mutta tarvittaessa päästään myös ohittamaan ajastus manuaalisesti käyttöliittymän kautta (kuva 25). Haluttiin myös, että manuaalisen ohjauksen unohtuessa päälle ajastin voisi palauttaa järjestelmän takaisin säädettyyn päivärytmiin.



Kuva 25. Node-REDissä toteutettu valojen ohjausohjelma.

Ohjelmakoodi toimii siten, että käynnistyessään ajastin (kuvassa kasvivaloajastin) syöttää ajanhetkestä riippuen totuusarvon *true* tai *false*, joka ohjautuu homekit-solmun (kuvassa valot) kautta Raspberry Pi:n GPIO-ulostuloon nro 11, joka ohjaa releen toimintaa ”päällä/pois päältä” -tyyppisesti.

Ohjelmakoodi on vietävä homekit-solmun kautta, jotta saadaan päivitetty tieto valojen tilasta Apple HomeKit -sovellukseen. Jos valoja ohjataan sovelluksen kautta, ohittaa se ajastimen ohjauksen, koska viesti lähtee tällöin liikkeelle homekit-solmusta (kuvassa valot). Sovelluksen kautta tehdyt ohjaukset nollautuvat siinä vaiheessa, kun ajastin vaihtaa tilaansa. Esimerkiksi illalla manuaalisesti sammutetut valot syttyvät aamulla ajastimen mennessä automaattisesti ”On”-tilaan. Node-Redissä törmätään usein siihen, että tietotyypit eivät käy yhteen eri solmujen kesken. Tässä ohjelmassa ajastimen antama string-teksti muutettiin boolean-muotoon, ja homekit-solmun antama boolean-arvo käännettiin numeroksi.

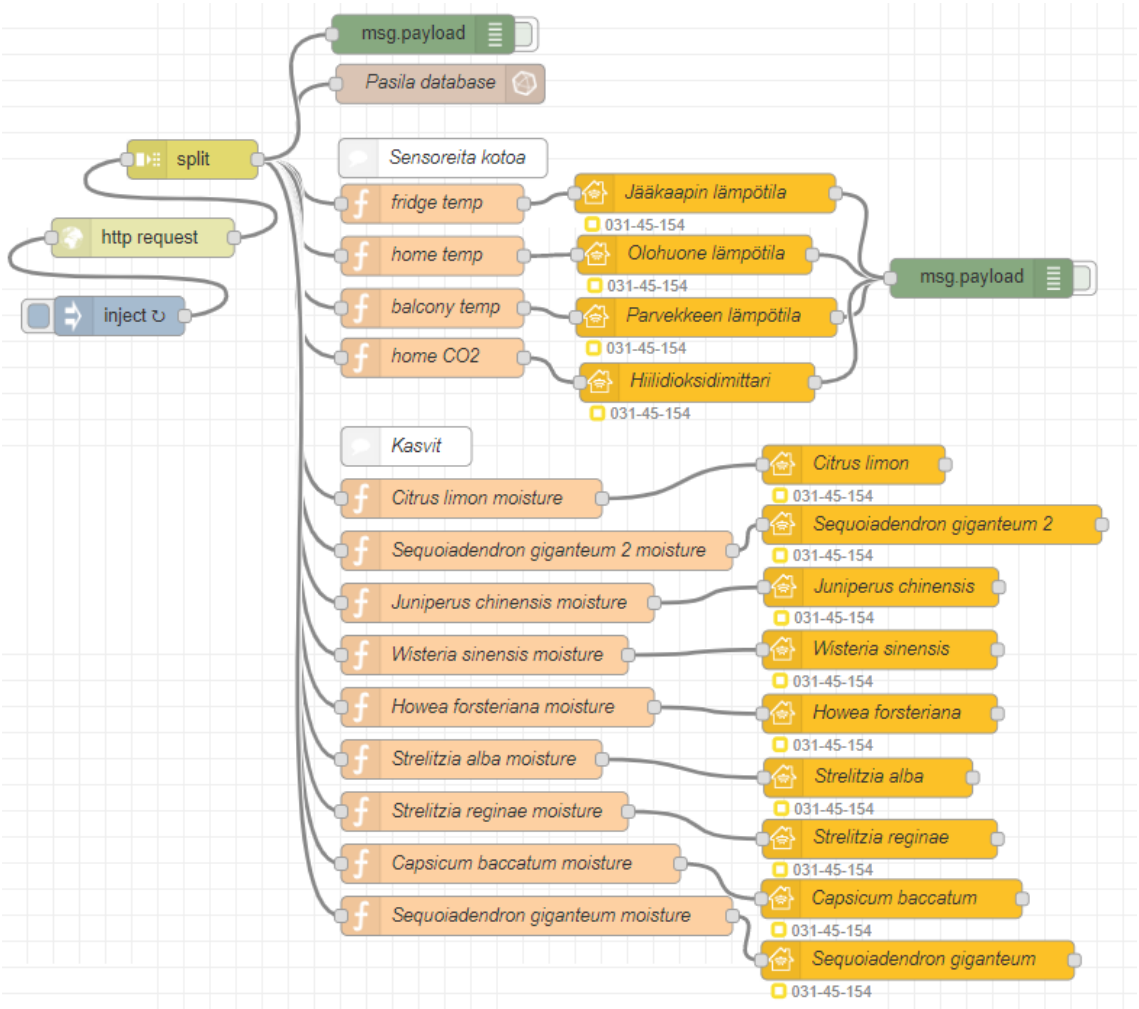
Sensoriarvojen vieminen Apple HomeKit -sovellukseen

Sensoriarvot poimittiin JSON-syötteestä function-solmujen avulla (kuva 26). Split-solmun jälkeen kaikki sensorit ovat samassa sarakkeessa, joten ne voidaan numeroidusti ohjata omiin homekit-solmuihin, käyttäen hyväksi function-solmuun kirjoitettua sarakkeen erotteluskriptiä (esimerkkikoodi 11).

```
msg.payload = {
   CurrentTemperature: (msg.payload[10].fields.temp)
}
return msg;
```

Esimerkkikoodi 11. Function-solmuun kirjoitettu skripti, jolla on poimittu sarakkeen kohdasta 10 kentän ”fields.temp”-arvo. Arvo ohjataan homekit-solmuun, joka päivittää arvon Homekit-sovellukseen.

JSON-syöte on muodostettu niin, että kaikki sensorit tulevat mukaan aina viestiin (ilman mittausarvoja), vaikka niihin ei olisi saatu yhteyttä. Tämä mahdollistaa sen, että sensorit ovat aina samoissa sarakkeen paikoissa, eivätkä ne pääse menemään sekaisin. Kun function-solmu on poiminut halutun arvon, siirtää se poimitun arvon homekit-solmuun.



Kuva 26. JSON-syötteestä on eroteltu sensorit function-solmujen avulla omiin homekit-solmuihin.

Edellisessä esitellyt ratkaisut perustuvat Anna Gerberin tekemään Node-RED-ohjelmointioppaaseen, joka on julkaistu IBM Developer -sivustolla (Gerber 2018). Sivustolla on tarkemmin kerrottu homekit-solmujen käytöstä.

3.7.3 InfluxDB

InfluxDB on InfluxData-yrityksen kehittämä NoSQL-tyyppinen tietokantaohjelma, joka on helposti asennettavissa ja yhdistettävissä tämän projektin muihin ohjelmiin (InfluxData 2019). Tässä projektissa InfluxDB-tietokantaa käytettiin sensoriarvojen tallentamiseen. Sensoriarvojen tarkasteluun käytettiin Grafana-ohjelmaa, josta tarkemmin luvussa 3.8.1 Grafana. Seuraavassa ohjeessa käydään läpi, kuinka InfluxDB asennetaan Raspberry

Pi:lle. Olettaen, että asennus tehdään komentoriviltä käsin, asentaminen aloitetaan kirjoittamalla seuraavat komennot (esimerkkikoodi 12).

```
curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -
source /etc/os-release
(test $VERSION_ID = "7" && echo "deb https://repos.influxdata.com/debian
wheezy stable" | sudo tee /etc/apt/sources.list.d/influxdb.list)
(test $VERSION_ID = "8" && echo "deb https://repos.influxdata.com/debian jes-
sie stable" | sudo tee /etc/apt/sources.list.d/influxdb.list)
sudo apt-get update && sudo apt-get install influxdb
sudo service influxdb start
sudo nano /etc/influxdb/influxdb.conf
```

Esimerkkikoodi 12. InfluxDB-tietokantaohjelman ensimmäisen vaiheen asennuskäskyt.

Tämän jälkeen muutetaan influxdb.conf-tiedostosta http-asetuksia, siten että #-merkki poistetaan kahdesta kohdasta (kuva 27). Tämän jälkeen tallennetaan muutokset ja poistutaan tekstieditorista. Edellinen vaihe oli tehtävä, jotta Grafana ja Node-RED saadaan yhdistettyä InfluxDB:hen.

```
[http]
  # Determines whether HTTP endpoint is enabled.
  enabled = true

  # Determines whether the Flux query endpoint is enabled.
  # flux-enabled = false

  # Determines whether the Flux query logging is enabled.
  # flux-log-enabled = false

  # The bind address used by the HTTP service.
  bind-address = ":8086"
```

Kuva 27. HTTP- otsikon alta on poistettu # - merkki kohdasta enabled ja bind-address.

Lopuksi luodaan vielä käyttäjäprofiili tietokannalle komentoriviltä käsin (esimerkkikoodi 13).

```
influx
CREATE USER "pi" WITH PASSWORD 'raspberry' WITH ALL PRIVILEGES
```

Esimerkkikoodi 13. Käyttäjäprofiilin luominen InfluxDB-tietokantaohjelmaan.

Taulukossa 1 luetellaan muutamia hyödyllisiä komentoja tietokannan selaamiseen komentoriviltä käsin. Komennot toimivat sillä edellytyksellä, että tietokantaan on tallennettu sensoriarvot Influx batch -solmun ohjeiden mukaan (Node-RED 2019b).

Tarkemmat ohjeet tietokannan käytöstä löytyvät InfluxDB:n omasta dokumentaatiosta. Huomioitavaa on, että InfluxDB käyttää tietokantamerkintöjä tehdessään millisekuntiaikaleimaa, joka ei ole tarkoitettu ihmissilmin luettavaksi. Millisekuntimerkintä voidaan väliaikaisesti muuntaa ihmissilmin luettavaan muotoon komentoriville kirjoitettavalla komennolla (taulukko 1). Millisekunnit voidaan myös kääntää ihmissilmin luettavaan muotoon internetistä löytyvällä muuntimella (xmillis.com 2019). Muunnin tulee käyttökelpoiseksi varsinkin silloin, jos halutaan etsiä mittausarvoja tietyltä aikaväliltä.

Taulukko 1. Muutamia hyödyllisiä komentorivikomentoja InfluxDB-tietokantaohjelman käyttöön. Taulukko on tehty InfluxDB:n dokumentaatiossa olevien esimerkkien pohjalta.

Komentorivin teksti	Selite
influx	Käynnistää InfluxDB-tietokantaohjelman hallinnan
CREATE DATABASE pasila	Luo tietokannan nimeltä pasila
CREATE USER "pi" WITH PASSWORD 'raspberry' WITH ALL PRIVILEGES	Luo käyttäjäprofiilin tietokannan käyttöön
SHOW DATABASES	Näyttää luotujen tietokantojen nimet
USE pasila	Ottaa käyttöön tietokannan nimeltä pasila
SHOW MEASUREMENTS	Näyttää sensorit, joille tehty tietokanta merkintöjä
SELECT * FROM "448radio" GROUP BY * ORDER BY DESC LIMIT 10	Näyttää 10 viimeisintä merkintää sensorilta 448radio
SELECT * FROM "448radio" LIMIT 10	Näyttää 10 ensimmäistä merkintää sensorilta 448radio
PRECISION RFC3339	Muuttaa millisekuntiaikaleiman muotoon päivämäärä + kellonaika
SELECT * FROM "448radio" WHERE TIME = 1573211901599604652	Näyttää sensorin 448radio mittausarvot ajanhetkellä 1573211901599604652
SELECT *FROM "448radio" WHERE TIME >= 1570545420000ms AND time <= 1570546020000ms LIMIT 100	Näyttää kaikki sensorin mittausarvot väliltä 1570545420000 -> 1570546020000. Näyttäen korkeintaan 100 merkintää
DELETE FROM "448radio" WHERE TIME = 1573211901599604652	Poistaa sensorin 448radio mittausarvot ajanhetkellä 1573211901599604652
DROP MEASUREMENT "448radio"	Tyhjentää kaiken mittausdatan sensorilta 448radio
DROP SERIES FROM /.*/	Tyhjentää käytössä olevan tietokannan kaikki merkinnät

3.8 Ohjelmat käyttöliittymään

3.8.1 Grafana

Grafana on avoimen lähdekoodin ohjelmisto, jossa voi tietokantaan kerättyjä tietoja esittää graafisesti ja määritellä niihin liittyviä hälytyksiä. Grafana on asennettavissa yleisimmille käyttöjärjestelmille (Linux, Windows ja macOS) ja sitä voidaan käyttää yleisimmillä verkkoselaimilla. Grafanan vahvuutena voidaan myös pitää sitä, että se on yhdistettävissä yli 30 eri tietokantaohjelmaan. (Grafana Labs 2019.)

Tässä esimerkkityössä Grafanaa käytettiin sensoriarvojen graafiseen esittämiseen, hälytysten asettamiseen ja videokuvan katsomiseen. Lisäksi luotiin myös virtuaaliset painonapit kasteluohjelmien käynnistämiseen. Grafanan asentaminen aloitettiin Raspberry Pi:lle seuraavilla komennoilla (esimerkkikoodi 14).

```
wget https://dl.grafana.com/oss/release/grafana_6.3.6_armhf.deb
sudo apt-get install -y adduser libfontconfig1
sudo dpkg -i grafana_6.3.6_armhf.deb
sudo apt-get update
sudo apt-get install Grafana
sudo systemctl daemon-reload
sudo systemctl start grafana-server
sudo systemctl enable grafana-server.service
```

Esimerkkikoodi 14. Grafanan asentamiseen kirjoitettavat komentorivikomennot.

Viimeisimmät ohjeet Grafanan asentamiseen kannattaa tarkistaa asennuksen yhteydessä Grafanan omasta dokumentaatiosta, jotta uusin saatavilla oleva versio saadaan asennetuksi. On myös hyvä huomioida asennustiedostoa valittaessa, että ARMv7-suorittimelle tehty asennustiedosto toimii Raspberry Pi 2:ssa ja sitä uudemmissa malleissa. Vanhemmille Raspberry Pi -malleille tulisi valita ARMv6-suorittimelle tehty asennuspaketti. Asennuksen jälkeen Grafanan käyttöliittymään päästään kirjautumaan menemällä URL-osoitteeseen (esimerkkikoodi 15).

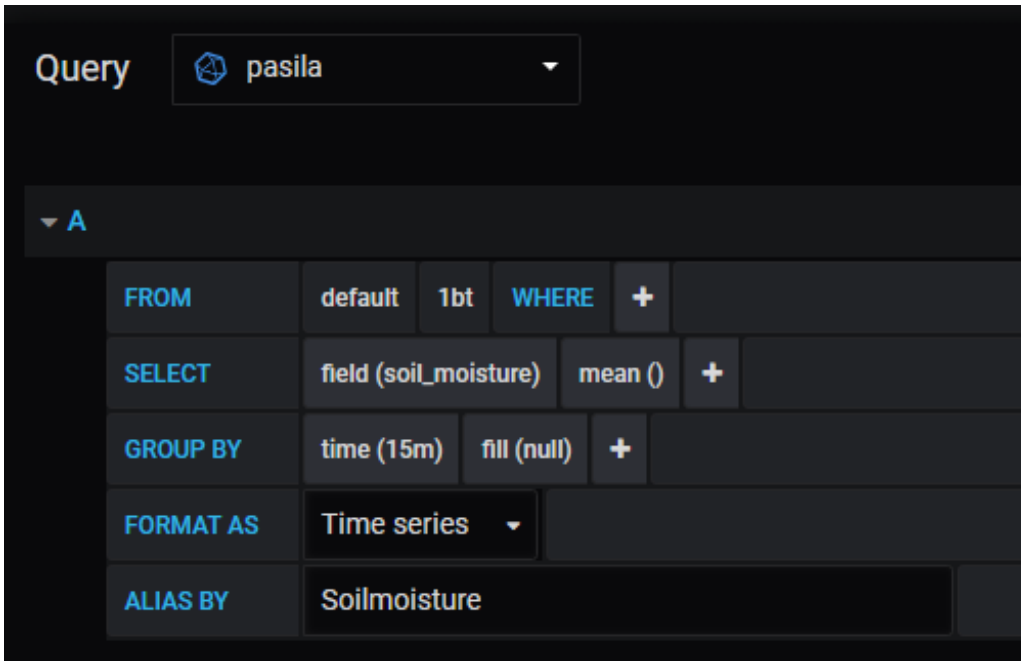
```
http://192.168.1.227:3000/login
```

Esimerkkikoodi 15. Grafanan käyttöliittymään päästään käsiksi kirjoittamalla Raspberry Pi:n oma IP-osoite ja portti nro 3000. Ensimmäisellä kirjautumisella oletuskäyttäjätunnus ja -salasana ovat kummatkin ”admin”.

Kun käyttöliittymään on kirjauduttu, kannattaa oletuksena olleet kirjautumistunnukset vaihtaa, ja tämän jälkeen voidaankin määritellä yhdistettävät tietokannat. Tässä projektissa tietokantaohjelmaksi määritettiin Raspberry Pi:llä toimiva InfluxDB (liite 3).

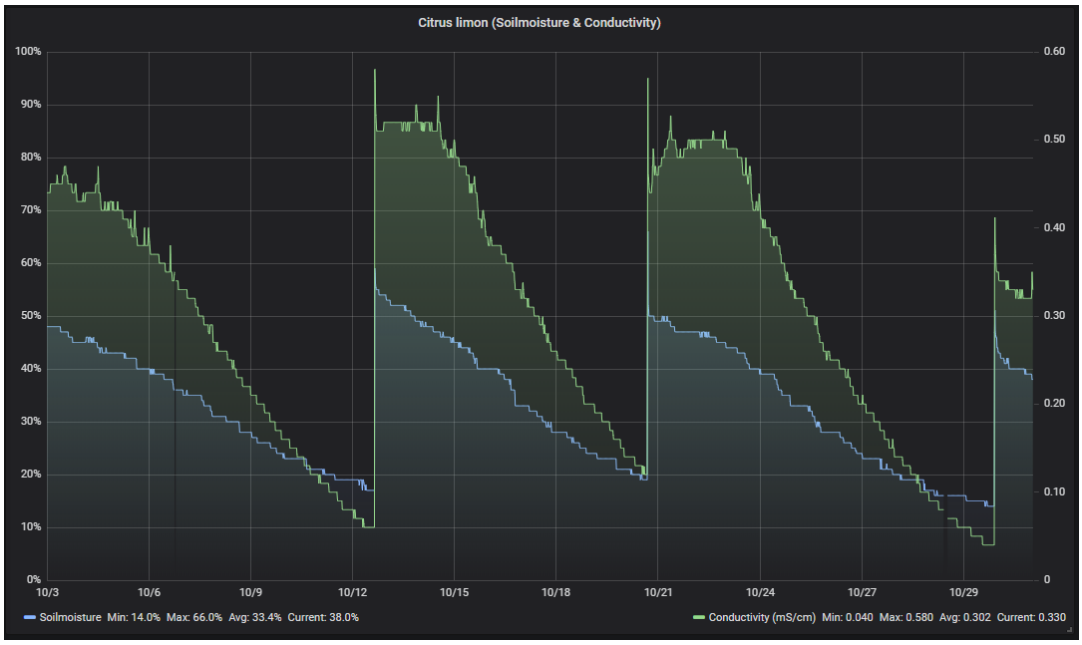
Kuvaajien muodostaminen

Kun tietokantaohjelmaan on saatu yhteys, voidaan aloittaa kuvaajien tekeminen. Kuvaajien tekemisestä löytyy kattavat ohjeet Grafanan omasta dokumentaatiosta, ja saatavilla on myös Grafanan julkaisemia opetusvideoita aloittelijoille. Tässä työssä ei tarkemmin opeteta kuvaajien tekemistä, mutta tähän asti tehtyjen vaiheiden pohjalta Grafanan tulisi nyt löytää InfluxDB:ssä määritellyt sensorit ja niiden sisältämät arvot (kuva 28). Kuvaajien muodostamisen tästä eteenpäin pitäisi onnistua Grafanan ohjeiden avulla.

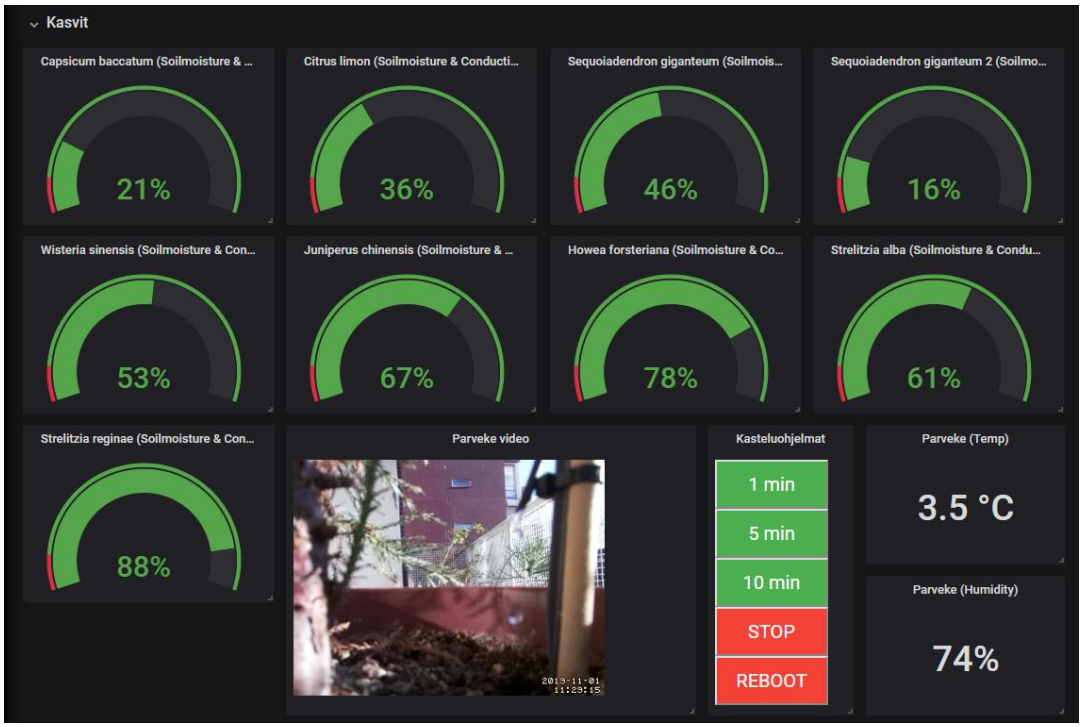


Kuva 28. Kuvaajan tekeminen aloitettu sensorille (1bt) ja tietokannasta haetuksi arvoksi valittu kasvualustan kosteusprosentti (soil_moisture).

Grafanassa on useita eri mahdollisuuksia esittää tietoa tietokannasta. Seuraavissa kuvissa on esimerkki sensorikohtaisesta kuvaajasta (kuva 29) ja käyttöliittymästä (kuva 30), joka sisältää kasvualustojen kosteusprosentin, videokuvan kasveista ja painonapit kasteluohjelmille.



Kuva 29. Mi Flora -sensorin mittaamat kosteus- ja sähkönjohtavuusarvot sitruunapuun kasvualustasta. Kuvaajasta nähdään päivien kuluessa tapahtunut kosteuden ja sähkönjohtavuuden lasku, jota on seurannut aina kasvualustan uudelleen kastelu.



Kuva 30. Grafanan käyttöliittymä manuaalista kasvien kastelua varten. Näytöstä nähdään viimeisimmät kasvikohtaiset kasvualustan kosteusprosentit ja videokuva, josta nähdään kastelutapahtuma painonappeja painaessa.

Videokuvan, varoitusviestien ja painonappien lisääminen käyttöliittymään

Jotta videokuva, hälytystoiminnot ja kastelunapit saadaan toimimaan, pitää Grafanan asetuksia käydä muuttamassa komentoriviltä käsin komennolla (esimerkkikoodi 16).

```
sudo nano /etc/grafana/grafana.ini
```

Esimerkkikoodi 16. Komentorivikomento, jolla päästään muutamaan Grafanan asetuksia.

Asetuksista muutettiin seuraavat kohdat:

- disable_sanitize_html = true
- SMTP / Email asetukset
- domain = esimerkki.dy.fi.

Tämän jälkeen voidaan Grafanan näytölle tuoda videokuva luomalla tekstipaneeli, joka on määritetty html-tilaan. Tässä työssä tekstilaatikkoon lisättiin rivi (esimerkkikoodi 17).

```
<img style="-webkit-user-select: none;margin: auto;"
src="http://kaste.dy.fi:8004/" width="330" height="250">
```

Esimerkkikoodi 17. Videokuvan lisääminen Grafanan tekstipaneeliin. Huomioitavaa on kameran IP-osoitteen määrittely ja haluttu kuvan koko.

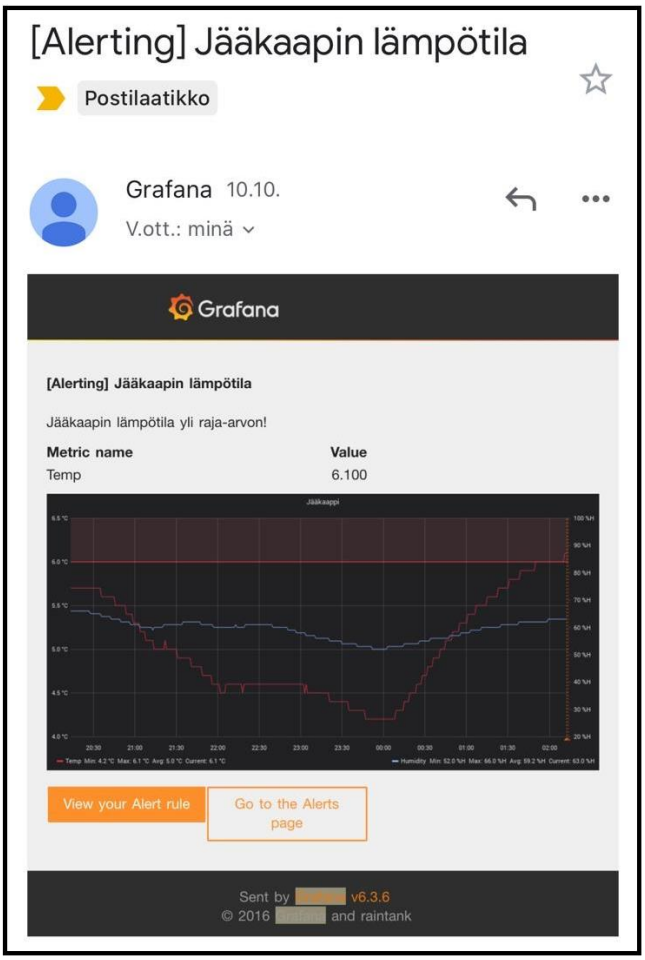
Painonapit saatiin luotua omaan tekstipaneeliin samalla tyylillä, kuin videokuva (liite 4 1(2)). Tässä työssä painonappien toimintaperiaate perustui LabVIEW-tutoriaalissa esitettyyn vaihtoehtoon, jossa LabVIEW-ohjelmalle syötetään numeroarvoa 1 tai 0 verkkoselaimen kautta, joka ohjaa vesipumpun sammumista ja käynnistymistä LabVIEW-ohjelmassa. (LabVIEW MakerHub 2016b.)

Grafana tarjoaa useita vaihtoehtoja hälytystoimintojen asettamiseen. Tässä esimerkissä hälytystoiminnot asetettiin siten, että Grafanaa varten luotiin uusi Gmail-tili, jonka kautta Grafana lähettää varoitusviestit. Luodun Gmail-tilin tunnukset on lisättävä Grafanan grafana.ini-tiedostoon, jotta hälytystoiminnot voisivat alkaa toimia (kuva 31).

```
################################### SMTP / Emailing ##########################
[smtp]
enabled = true
host = smtp.gmail.com:587
user = esimerkki@gmail.com
# If the password contains # or ; you have to wrap it with trippel quotes. Ex """#password;"""
password = 12345
```

Kuva 31. Varoitustoimintoa varten grafana.ini-tiedostoon on määriteltävä sähköpostitunnukset. Huomaa myös muut asetetut kohdat.

Tämän jälkeen varoitusten lähettämistä voidaan testata Grafanan käyttöliittymän kautta. Varoitukset määritellään sensorikohtaisesti, mutta tämän lisäksi on myös määriteltävä sähköpostiosoitteet, joihin varoitusviesti lähetetään. Varoitusten toimiessa Grafanan tulisi lähettää sähköpostia välittömästi, kun määritellyt raja-arvot ylittyvät (kuva 32).



Kuva 32. Grafanan sähköpostilla lähettämä varoitusviesti asetetun lämpötila-arvon ylittymisestä. Varoitusviesti sisältää kuvauksen varoituksesta, seurattavan lämpötilan arvon ja kuvan kaappauksen lämpötilan muutoksista varoitusta edeltäneeltä ajalta.

Grafanan 6.3.6-versiossa ongelmana oli, että kuvankaappaukset raja-arvojen ylityshetkiltä eivät sisältyneet varoitusviesteihin. Tämä johtui phantomjs-tiedoston puuttumisesta. Ongelma korjaantui seuraavilla komennoilla (esimerkkikoodi 18).

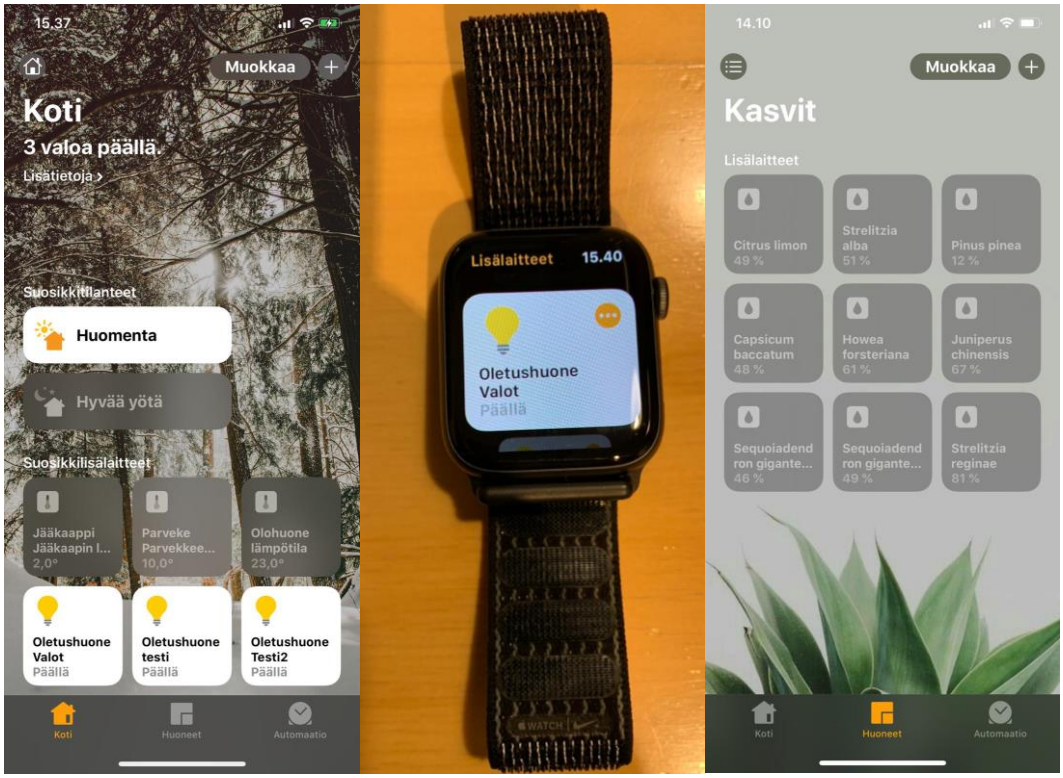
```
cd ~
wget https://github.com/fg2it/phantomjs-on-raspberry/raw/master/rpi-2-
3/wheezy-jessie/v2.1.1/phantomjs
chmod 755 phantomjs
sudo chown root: phantomjs
sudo mv phantomjs /usr/share/grafana/tools/phantomjs/
```

Esimerkkikoodi 18. Kuvankaappauksien lisääminen varoitusviesteihin korjaantui phantonjs-tiedoston lataamisella Raspberry Pi:lle.

Todennäköisesti Grafana Labs korjaa kuvankaappaukseen liittyvän ongelman tulevissa ohjelmaversioissaan.

3.8.2 Apple HomeKit

Teknologiayhtiö Apple on kehittänyt oman kotiautomaatiosovelluksen nimeltä HomeKit, johon voidaan yhdistää eri valmistajien laitteita. Sovelluksen kautta voidaan hallita ja seurata yhdistettyjen laitteiden tilaa, sekä luoda tilanteita, joiden avulla voidaan hallita useita laitteita samanaikaisesti (kuva 33). Esimerkiksi nukkumaan mennessä voi kertoa ääniavustaja Sirille ”Hyvää yötä”. Tämän seurauksena kaikki ”hyvää yötä” -tilanteeseen yhdistetyt laitteet muuttavat tilaansa ennalta määritellysti, esim. kodin ulko-ovet menevät lukkoon ja valot sammuvat. Käyttöliittymä voidaan myös jakaa kodin muille asukkaille kaikkine toimintoineen, mikäli heiltä löytyy yhteensopivia laitteita. (Apple 2019.)



Kuva 33. Apple HomeKit-sovelluksen käyttöliittymä iPhonesta ja Apple Watchista tarkasteltuna. Käyttöliittymästä voidaan mm. aktivoida yksittäisiä laitteita tai tilanteita tai tarkastella sensoriarvoja, kuten kasvien kasvualustan kosteusprosentteja ja parvekkeen lämpötilaa.

Suosituimmista ääniohjausteknologioista Applen Siri on vielä tällä hetkellä ainut, joka tukee ohjauksessa suomen kieltä. Kun tilanteita ja laitteita lisätään HomeKit-sovellukseen, osaa Siri ääniohjauksen kautta kertoa esimerkiksi paljonko jääkaapin lämpömittari näyttää, onko valot päällä tai pitäisikö huoneisto tuulettaa (hiilidioksiditason tunnistus).

Omassa käytössä olen ääniohjausta pitänyt hauskana lisäominaisuutena, mutta sitä ei tule käytettyä aktiivisesti. Usein Siri kuulee annetun käskyn väärin tai taustamelu estää Sirin aktivoitumisen kutsuttaessa.

Laitteiden yhdistäminen HomeKit-sovellukseen tapahtuu ”Lisää lisälaite”-toiminnon kautta. Node-REDissä luodut laitteet ovat löydettävissä käyttöönottokoodilla, joka määriteltiin laitteita luotaessa niiden asetuksiin.

Huomion arvoista Apple HomeKit-sovelluksessa on se, että se on asennettavissa vain Applen omille laitteille. Lisäksi, jos halutaan ohjata laitteita kodin ulkopuolelta tai käyttää HomeKit-sovelluksen automaatiotoimintoa, tarvitaan järjestelmän keskusyksiköksi Applen tuotteista, joko Homepod, Apple TV tai iPad. Tämä siksi, koska Applen etäyhteydessä tieto kulkee Applen oman pilvipalvelun kautta, johon Raspberry Pi:tä ei saa yhdistettyä. Apple HomeKit-sovelluksesta puuttuvat kokonaan myös graafiset kuvaajat ja historiatiedot ohjaustapahtumista.

Tässä työssä Apple-keskusyksikön puuttuminen kierrettiin tekemällä halutut automaatiot Node-REDissä, sekä asentamalla etäyhteyttä ja Android-laitteita varten Homebridge-ohjelma. (Homebridge 2019.) Graafiset kuvaajat taas löytyvät Grafana-ohjelmasta. Homebridgessä on saatavilla myös ohjaustapahtumien historiatiedot.

3.8.3 Homebridge

Homebridge-ohjelma mahdollistaa laitteiden ohjaamisen ja sensoriarvojen seuraamisen Apple HomeKit -sovelluksen lisäksi myös Android-laitteilla. Homebridgen avulla saatiin luotua myös yhteys laitteisiin kodin ulkopuolelta. Homebridgeä voidaan käyttää myös yksinään kodinohjausjärjestelmän keskuksena, ilman Applen tuotteita.

Homebridgen asennus

Homebridgen ja siihen liittyvien ohjelmistojen asennus tehtiin seuraavilla käskyillä Raspberry Pi:lle komentoriviltä käsin (esimerkkikoodi 19).

```
sudo apt-get update
sudo apt-get upgrade
curl -sL https://deb.nodesource.com/setup_10.x | sudo bash -
sudo apt-get install -y nodejs
sudo apt-get install libavahi-compat-libdnssd-dev
sudo npm install -g --unsafe-perm homebridge
sudo npm install -g --unsafe-perm homebridge-config-ui-x
sudo mkdir /var/homebridge
sudo nano /var/homebridge/config.json
```

Esimerkkikoodi 19. Homebridgen asennuksen ensimmäinen vaihe. Asennus loppuu vaiheeseen, jossa määritellään config.json-tiedoston sisältö.

Jatketaan asentamista liittämällä sisältö (liite 5) config-tiedostoon. Tallennetaan tiedosto ja jatketaan asentamista komennolla (esimerkkikoodi 20).

```
sudo nano /etc/default/homebridge
```

Esimerkkikoodi 20. Homebridgen asennus loppuu vaiheeseen, jossa määritellään homebridge-tiedoston sisältö.

Jatketaan asentamista liittämällä sisältö (liite 6) homebridge-tiedostoon. Tallennetaan tiedosto ja jatketaan asentamista komennolla (esimerkkikoodi 21).

```
sudo nano /etc/systemd/system/homebridge.service
```

Esimerkkikoodi 21. Homebridgen asennusvaihe, jossa luodaan homebridge.service-tiedosto.

Liitetään tiedostoon sisältö (liite 7) homebridge.service-tiedostoon. Tallennetaan tiedosto ja jatketaan asentamista komennolla (esimerkkikoodi 22).

```
sudo useradd --system homebridge
sudo chmod -R 0777 /var/homebridge
sudo systemctl daemon-reload
sudo systemctl enable homebridge
sudo systemctl start homebridge
sudo visudo
```

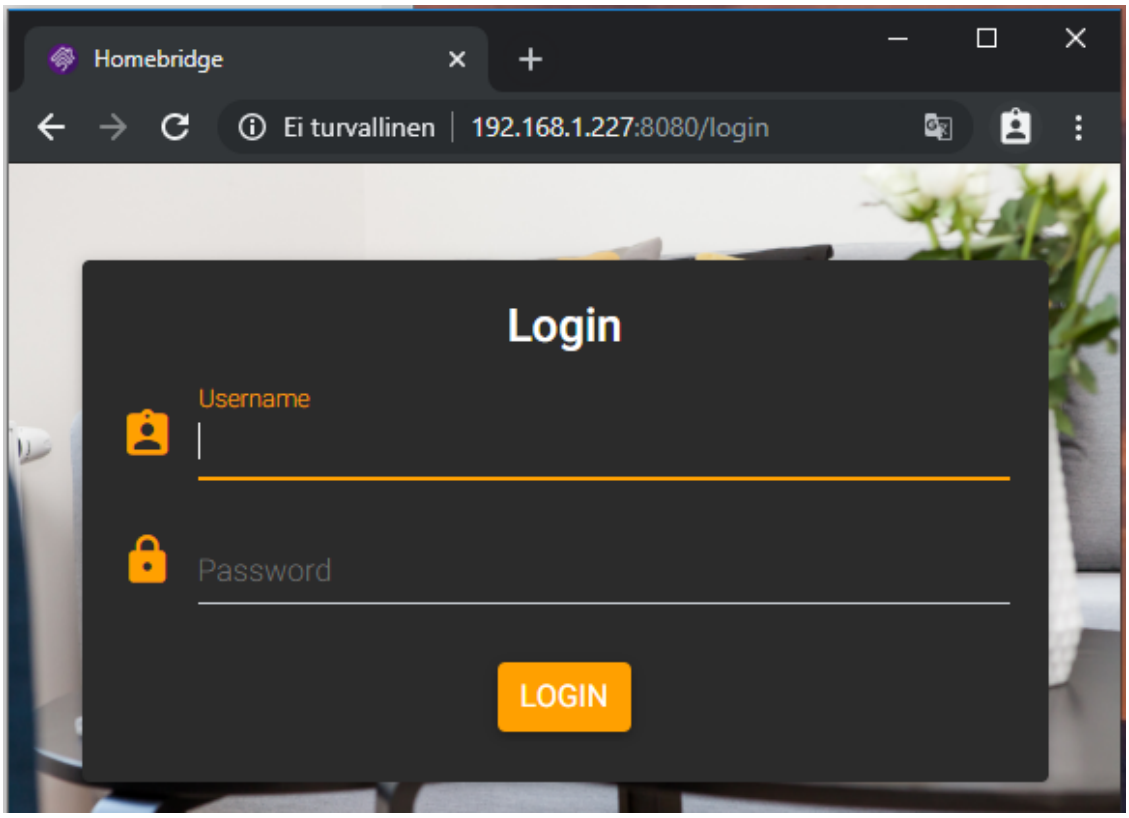
Esimerkkikoodi 22. Homebridge-ohjelman asennusvaihe, jossa on määritelty Homebridge käynnistymään automaattisesti Raspberry Pi:n käynnistyessä. Lopuksi avataan sudoers-tiedosto asetusten määrittämistä varten.

Tämän jälkeen lisätään seuraava sisältö sudoers-tiedoston loppuun omalle rivilleen (esimerkkikoodi 23).

```
homebridge     ALL=(ALL) NOPASSWD: ALL
```

Esimerkkikoodi 23. Sudoers-tiedoston loppuun lisättävä rivi. Ilman tätä rivilisäystä Homebridge ei pystynyt näyttämään järjestelmän lokitiedostoja verkkoselaimesta käsin.

Kun näiden vaiheiden jälkeen Raspberry Pi käynnistetään uudelleen, pitäisi Homebridgen käyttöliittymä olla käytettävissä verkkoselaimella paikallisen IP-osoitteen portista nro 8080 (kuva 34). Jatkossa asetusten määrittäminen ja lokitietojen seuraaminen voidaan tehdä verkkoselaimen kautta.



Kuva 34. Kirjautuminen Homebridgen käyttöliittymään. Oletussalasanat ovat kirjautumiseen admin ja admin, jotka kannattaa muistaa vaihtaa heti ensimmäisellä kirjautumisella.

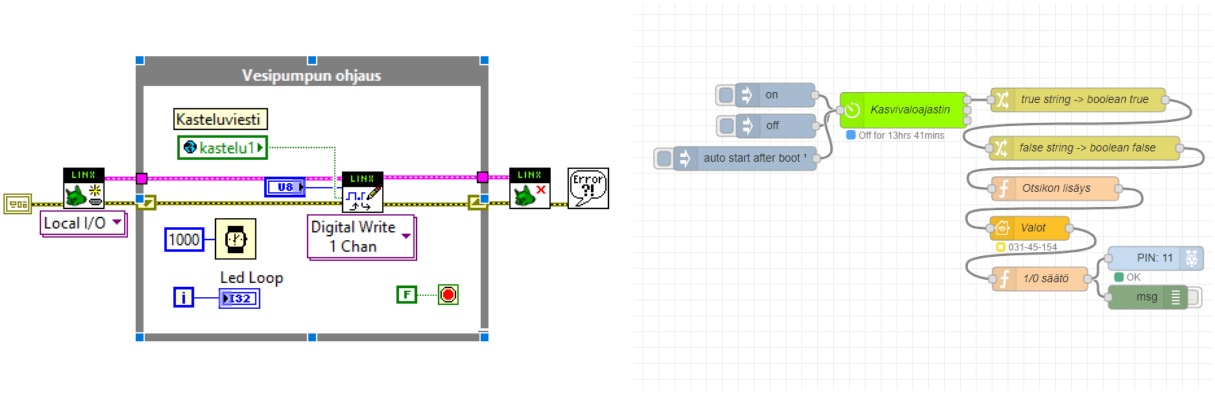
Yksityiskohtaiset ja päivitetyt asennusohjeet Homebridgen asennukseen löytyvät Homebridgen GitHub-sivuilta. (Homebridge 2019.)

Laitteiden lisääminen Node-REDistä Homebridgeen

Kun laitteita määritetään Node-RED-ohjelmassa homekit-solmujen avulla, tulevat ne sitä mukaa myös näkyviin Homebridge-ohjelmaan. Laitteiden asetuksia määritettäessä pitää käyttää Homebridgen antamaa PIN-koodia, tai muuten päädytään tilanteeseen, että laitteet kyllä näkyvät käyttöliittymässä, mutta ohjaus ei toimi.

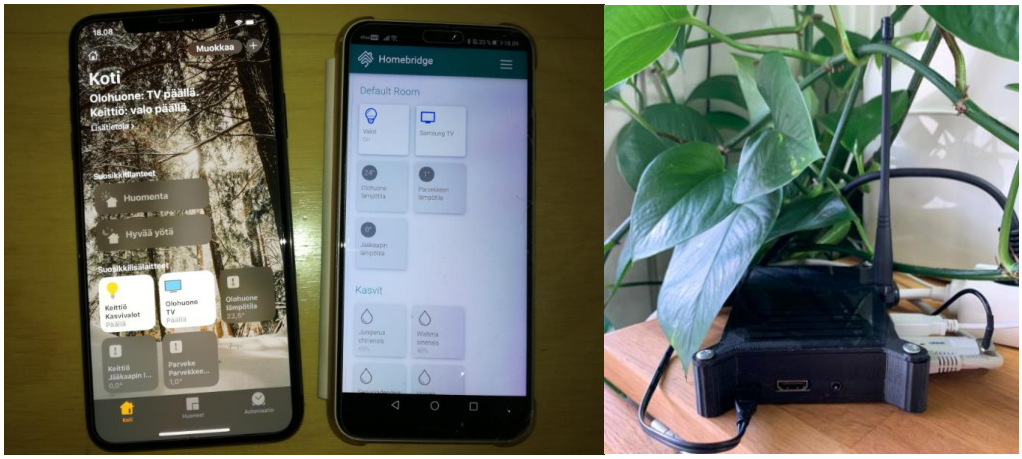
4 Lopputulos

Tämän insinöörityön tuloksena saatiin etäohjattava järjestelmä kasvien ylläpitoon, kun vesipumppu ja kasvivalot yhdistettiin (kuva 35) tietokoneohjelmien ohjattavaksi (LabVIEW ja Node-RED).



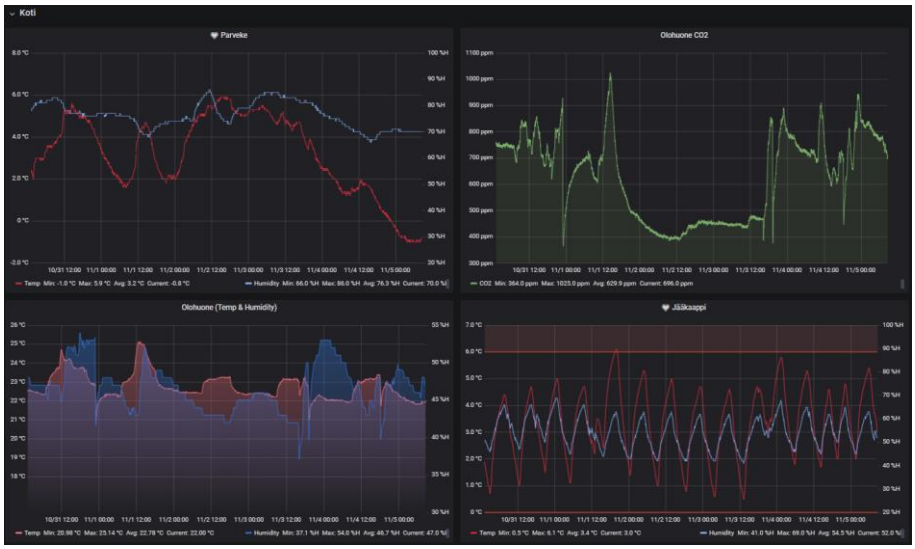
Kuva 35. Tässä työssä vesipumpun ohjaus toteutettiin käyttäen LabVIEW-ohjelmointia (kuvassa vasemmalla) ja valojen ohjaus Node-RED-ohjelmoinnilla.

Työssä ratkaistiin myös Apple HomeKit -sovelluksen yksipuolinen toiminta Applen laitteilla asentamalla rinnalle Homebridge-ohjelma, joka toimii käytännössä kaikilla laitteilla, joista löytyy nykyaikainen verkkoselain. Näin ollen samoja laitteita päästään ohjaamaan nyt Apple- ja Android-laitteilla (kuva 36). Homebridge luo myös käyttökelpoisen pohjan kodinohjausjärjestelmän laajentamiselle jatkossa.

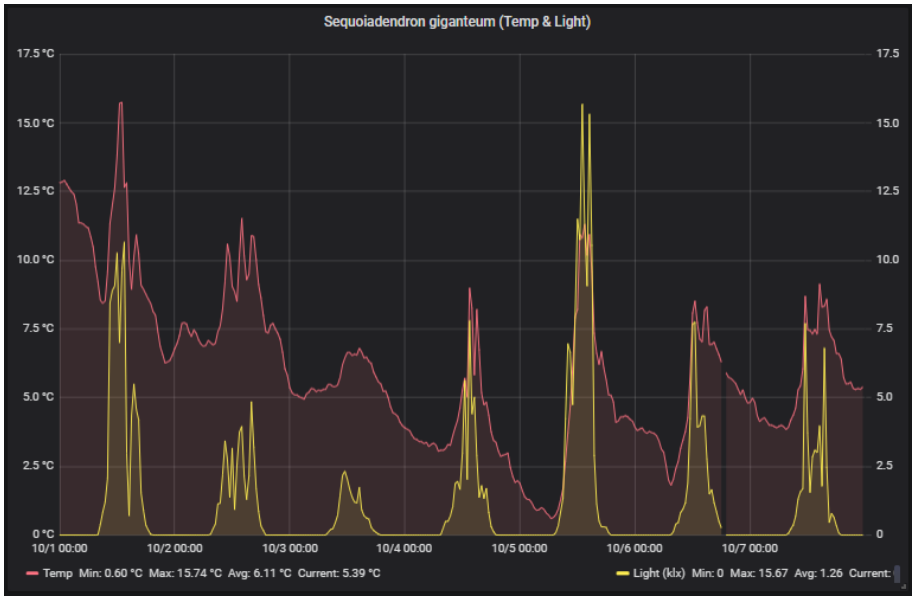


Kuva 36. Applen ja Huawein älypuhelimet yhdistettyinä hallitsemaan samoja laitteita eri käyttöliittymistä. Laitteiden hallinnan keskusyksikkönä toimii Raspberry Pi -tietokone.

Lisäksi saatiin tallennus-, monitorointi- ja varoitustoiminnot kaikille sensoreille, kun järjestelmään asennettiin InfluxDB-tietokantaohjelma ja tietokantaa lukeva Grafana-käyttöliittymä (kuvat 37 ja 38). Järjestelmään voidaan jatkossa lisätä uusia sensoreita tai vaihtaa rikkoutuneiden tilalle uusia, kunhan tunnetaan sensoreiden yhteensopivuus rakennetun järjestelmän kanssa.



Kuva 37. Grafana-ohjelman käyttöliittymä asetettuna näyttämään sensoriarvoja tietokannasta.



Kuva 38. Grafana-ohjelma on asetettu näyttämään Mi Flora -sensoreilta kerättyjä lämpötilan ja valonvoimakkuuden mittausarvoja. Kuvaajasta voidaan päätellä esimerkiksi, onko valittu ajanjakso ollut pilvinen vai aurinkoinen.

5 Yhteenveto

Tässä insinöörityössä rakennettiin etäohjattava järjestelmä kasvien ylläpitoon ja seurantaan. Rakennettua järjestelmää voidaan käyttää kodinohjausjärjestelmän pohjana, kun lisää laitteita liitetään järjestelmään. Lisäksi työssä perehdyttiin tämän hetken kotiautomaation tuomiin mahdollisuuksiin ja historiaan.

Tärkeimmät tavoitteet rakennetulle järjestelmälle olivat, että laitteiden tilaa voidaan valvoa ja ohjata etänä ja ohjauksiin voidaan luoda tarvittaessa tilanteeseen sopivaa automaatiota, esimerkiksi vesipumpun automaattinen käynnistys kasvualustan kosteusprosentin mukaan. Tärkeää oli myös pystyä tallentamaan sensoriarvot ja asettamaan näihin liittyviä varoitusviestejä.

Syventyminen joihinkin käsiteltyihin asioihin jäi varsin pinnalliseksi, työn laajuuden takia. Insinöörityö haluttiin kuitenkin pitää pakettina, jossa opastetaan kyseisen järjestelmän tekeminen alusta loppuun. Työtä voidaan käyttää myös esim. pikaoppaana tiedonkeruulaitteen nopeaan pystyttämiseen, kun valitaan asennettavaksi vain halutut ohjelmistot. Myös useille komentoriveille kirjoitettaville komennoille tämä insinöörityö toimii muistikirjana.

5.1 Pohdintaa rakennetusta järjestelmästä

Kaikki alkuperäisissä suunnitelmissa olleet tavoitteet rakennetulle ohjausjärjestelmälle saavutettiin. Työn tuloksena syntyneen oppaan pohjalta kyseisen järjestelmän rakentamisen pitäisi onnistua suoraviivaisesti. Järjestelmä on osoittautunut käytössä toimintavarmaksi, sillä se palautuu hyvin poikkeustilanteista, esimerkiksi jos laitteesta katkaistaan virta kesken toiminnan. Myöskään yksittäisten sensoreiden häviäminen järjestelmästä ei vaikuta mittausarvojen prosessointiin muilta sensoreilta, eikä yhden sensoriohjelman kaatuminen kaada muita ohjelmia. On todennäköistä, että työssä esitellyt ohjeet vanhenevat ajan kuluessa ja ohjeisiin on jäänyt virheitä, koska ohjausjärjestelmää ei asennettu kaikilta osin uusiksi raporttia kirjoittaessa.

Insinöörityötä tehdessä opittiin paljon uutta mm. sensoreista, tietokoneohjelmoinnista ja ohjelmien välisestä viestinnästä. Nämä aihealueet eivät olleet tämän insinöörityön

kirjoittajalle ennestään kovin tuttuja, eivätkä liittyneet varsinaisesti omaan syventymisaiheeseen insinööriopintojen loppuvaiheessa. Pohjaa insinöörityölle toivat kuitenkin harrasteprojekteissa opitut asiat, sekä sähkö-, ohjelmointi- ja automaatiotekniikan kursseilla opitut perusteet.

Projektin parhaiksi puoliksi nostaisin Node-RED-, InfluxDB- ja Grafana-ohjelmien käytön oppimisen. Yksin tai yhdistämällä nämä ohjelmat voidaan luoda nopeasti hyvinkin monimutkaisia, mutta selkeitä ohjelmia eri tilanteisiin. Uskoisin myös kyseisten ohjelmien tuen jatkuvan Raspberry Pi -tietokoneille vielä vuosikausia mm. suuren suosion ja kehittäjäyhteisön kautta tulevien lisälaajennusten ansiosta.

Tärkeitä opittuja asioita olivat myös säännölliset varmuuskopiot rakennettavasta järjestelmästä. Useissa tapauksissa tehdyt muutokset yhdessä ohjelmassa vaikuttivat toiseen ohjelmaan, minkä jälkeen haluttiinkin palata alkuperäiseen tilanteeseen. Joskus ohjelmien asennus myös epäonnistui siten, että helpoin ulospääsy tilanneesta oli formatoida käytetty muistikortti ja kirjoittaa siihen uusiksi edellisen varmuuskopion levykuvatiedosto.

Kesällä kastelujärjestelmän ollessa käytössä parvekkeella huomattiin, että kasvualustat kuivuvat eri nopeuksilla riippuen mm. säästä, kasvista, kasvualustantyypistä ja ruukun koosta. Tämä hankaloitti kasvien kastelua, koska pumpatessaan vettä vesipumppu kasteli samanaikaisesti kaikki ruukut. Ongelmaa hoidettiin ohjaamalla suurempiin ruukkuihin enemmän kastelupisteitä kuin pieniin ruukkuihin ja ajoittamalla kastelu ajankohtaan, jossa pienin ruukku oli kuivunut rutikuivaksi. Tästä huolimatta jotkut isoissa ruukuissa olleet kasvit kärsivät ylikastelusta. Tähän ongelmaan ei vielä keksitty hyvää ratkaisua.

5.2 Pohdintaa tämänhetkisistä kotiautomaatiojärjestelmistä

Työssä paneuduttiin myös tämän hetken kotiautomaatioon sen tuomien mahdollisuuksien ja hyötyjen kautta. Sanoisin, että jos kotiautomaatiotuotteet eivät erityisesti kiinnosta tai tarvetta jonkin asian automatisointiin ei ole, kannattaa pysytellä vielä muutama vuosi poissa näiltä markkinoilta. Useimmat kotiautomaatiotuotteet ovat tällä hetkellä niin kalliita, hankalasti säädettäviä ja huonosti yhteensopivia keskenään, ettei niiden hankinta muuten kuin harrastusmielessä olisi järkevää. Lisäksi markkinoilla on tuotteita, joissa on havaittu ongelmia ylikuumenemisen ja kestävyyden suhteen. Tästä esimerkkinä toimivat

Ikean Trådfri-tuotteet, jotka ovat osoittautuneet kestämään nykyisellään kotikäytössä vain alle kaksi vuotta (Pertilä 2019; Spiess 2017). Toki isossa talossa asuvalle tai liikuntarajoitteiselle henkilölle kotiautomaatiosovelluksilla voidaan saada suurta helpotusta elämään, kun kodin toimintoja voidaan ohjata vaikkapa kotisohvalta käsin. Järjestelmän säätäminen ja asentaminen tuo taas omat haasteensa, jos tekniikka on täysin uutta. Esimerkiksi pelkkä lampun vaihtaminen kodinohjausjärjestelmään, jossa yleiseksi himmennysprosentiksi on määritetty 90 %, voi olla tekniikkaa taitamattomalle varsin haastava prosessi, jossa pitäisi osata avata kodinohjausjärjestelmän asetukset siten, että asennettavan lampun tunnistekoodi päästään syöttämään järjestelmään. Vasta tämän jälkeen voidaan säätää uuden lampun himmennys siten, että se ei erotu muista lampuista palamalla kirkkaammin. Jos järjestelmän on asentanut kodin ulkopuolinen henkilö, kodin asukas ei välttämättä osaa mennä säätämään järjestelmän asetuksia ollenkaan.

Nykyinen laitteiden tietoturva herättää myös paljon epäilyksiä viimeaikaisten uutisten ja osaksi omien kokemuksien pohjalta. Esimerkiksi olen vahingossa peilannut puhelimeni näytön naapurin televisioon ja yhdistänyt naapurin Cromecast-laitteen omaan Google Home -sovellukseen ja näin aiheuttanut naapurille tahattoman ponnahdusviestin televisioruudulle. Lisäksi kerran etäohjausharjoituksissa reitittimen porttiohjaus unohtui osoittamaan Raspberry Pi:tä, jossa olivat käytössä oletustunnukset. Tästä seurasi Telia Security Operation Centerin yhteydenotto, jonka mukaan IP-osoitteestani oli aloitettu julkisen verkon porttiskannaus. Raspberry Pi:lle oli siis murtauduttu internetin kautta oletustunnuksilla ja asennettu verkkoa tutkiva ohjelmisto. Tämä tapahtuma toimi hyvänä muistutuksena sille, että eri ohjelmien oletustunnukset kannattaa aina vaihtaa.

Radiosignaaleja RTL-SDR-tikulla tutkiessani saan tietoja lähialueelta useista kymmenistä eri 433 MHz -taajuuksilla toimivista laitteista ja sensoreista. Samaan tapaan murtovaras voisi päätellä, onko talossa ketään kotona, jos asunnon hiilidioksidimittari lähettää tietoa matalalla olevasta tasosta tai ilmastointi on kytketty pitkäksi aikaa pois käytöstä. Myös langattomasti lähetettyjä ohjausviestejä voidaan helposti kaapata ja sen jälleen uudelleen lähettää. Näin voidaan esim. sekoittaa kiinteistöjen ilmastointilaitteita ja avata autojen ovia (Ikkala 2015).

Myös melkein joka kodista löytyvät WP2-salauksella toimivat langattomat lähiverkot ovat murrettavissa ainakin kahdella tavalla: joko murtamalla salasanat ohjelma-avusteisesti

eri salasanoja kokeilemalla tai käyttämällä hyväksi WP2-salauksen avainvarmistus vaihetta (Key Reinstallation Attack). Valitettavasti salauksen murtaminen on helppoa, eikä siihen tarvita erityistä tietoteknistä osaamista — riittää, että langattomaan kotiverkkoon murtautuva lataa tarvittavat ohjelmat internetistä ja lukee ohjeista, kuinka ohjelmia käytetään. (Kivelä 2018; 33.) Näin ollen kuka tahansa asiasta kiinnostunut voisi halutessaan hyökätä vieraaseen Wi-Fi-verkkoon ilman kiinnijäämisen pelkoa.

Edellä mainituista huonoista puolista ja kokemuksista huolimatta uskoisin kuitenkin erilaisten älylaitteiden saavuttavan seuraavan 15 vuoden aikana samantapaisen suosion ihmisten keskuudessa kuin matkapuhelimet pari vuosikymmentä sitten. Tämän hetken suosiota on vaikea arvioida, mutta kyselytutkimuksissa on saatu selville, että vain noin joka viides suomalainen omistaa jonkun kodin älylaitteen (Miinin 2018). Kun yhteiskuntaa rakennetaan etenevissä määrin digitaalisen teknologian varaan ja valtiovalta ohjaa kiinteistöjen rakentamista uusilla säädöksillä entistä energiatehokkaampaan suuntaan, automaatio tulee väistämättä lisääntymään kotitalouksissa.

Mielenkiintoista oli syventyä myös kotiautomaation historiaan ja sille vauhtia antaneeseen 70-luvun energiakriisiin. Historian tapahtumista tulee väistämättä mieleen tämän hetken ilmastokriisi. Olemme taas pakotettuna tilanteeseen, jossa pitäisi siirtyä käyttämään uusia energiamuotoja, vaikka korvaavien teknologioiden kehitys on edelleen pahasti kesken. Historia on osoittanut kehityksen edistyvän aina kriisiaikoina nopeimmin, joten ehkä näemme myös kiinteistöautomaatiossa merkittäviä uusia kehitysaskeleita lähivuosina. Kulkevathan automaatiosovellutukset väistämättä käsikädessä uusien teknologioiden kanssa.

Toivoisin tulevaisuuden älylaitteilta parempaa käyttäjäystävällisyyttä. Uusien laitteiden käyttöönoton ja asetusten määrittäminen tulisi olla nykyistä nopeampaa ja helpompaa. Tämä voitaisiin saada aikaan esimerkiksi yhdistämällä kasvojentunnistus, puheohjaus ja tekoäly kotiautomaation keskusyksikköön. Puheohjauksen ja tekoälyn avulla voitaisiin laitteiden hallinnasta ja käytöstä tehdä samanlainen tapahtuma kuin toisen ihmisen kanssa asioista sopiminen. Näin tekniikkaa vieroksuvatkin voitaisiin saada innostumaan kotiautomaation mahdollisuuksista ja vaikeasti ymmärrettävät järjestelmäasetukset osattaisiin säätää kenen tahansa toimesta, kun laite osaisi vuorovaikuttaa ihmisen tavoin. Jo nyt esimerkiksi Applen ääniavustaja Siriä voi ”opettaa” puheohjauksella muistamaan

käyttäjän perhesuhteita ja aktivoimaan eri toimintoja sovellusten välillä. Myös erilaisia laskutoimituksia voi antaa puhekomennon avulla Sirin laskettavaksi (toistaiseksi vain englannin kielellä). Viime vuosina on nähty jatkuvia kehitysaskelia puheentunnistuksessa ja siihen liittyvässä tekoälyssä. Uskoisin perinteisten ja graafisten ohjelmointikielien lisäksi ilmestyvän tulevaisuudessa myös puheohjattuja ohjelmointikieliä. Onhan tietokoneiden ohjelmointia pyritty aina helpottamaan, jotta ohjelmointi olisi mahdollista myös suuremmalle käyttäjäryhmälle. Tästä hyvänä esimerkkinä LabVIEW tai lapsille tarkoitettu Scratch-ohjelmointikieli, mikä mahdollistaa ”apurattailla” ohjelmoinnin samaan aikaan, kun ”konepellin alla” muodostuu todellinen ohjelmakoodi esim. C-ohjelmointikielellä. Ehkä lähitulevaisuudessa voimme ohjelmoida älykotimme toimintoja vain juttelemalla omille IoT-laitteillemme.

Myös tämänhetkiset tietoturvaongelmat pitäisi saada ratkaistua. Esimerkiksi kasvojentunnistusteknologia on jo nykyisellään osoittautunut toimivaksi tavaksi vahvistaa henkilöllisyys, vaikka sitä on onnistuttu huijaamaan eri keinoin, kuten 3D-tulostetuilla kasvoilla. Kuitenkin sitä pidetään yleisesti varmempana ja vaikeammin huijattavissa olevana tunnistustapana kuin PIN-koodi ja sormenjälkitunnistus. Hyödyt nähdään ainakin toistaiseksi uhkia suurempana. Uskosta kasvojentunnistukseen nousevana teknologiana kertoo esimerkiksi Suomen viranomaisten mielenkiinto kyseistä tekniikkaa kohtaan. Myös monet yritykset ovat kiinnostuneita liittämään kasvojentunnistuksen osaksi palveluitaan. (Hjelt 2019; Pajari 2019; Korhonen 2017.)

Kasvojentunnistusta voitaisiinkin hyväksikäyttää enemmän myös kotiautomaatiolaitteissa. Sillä voisi esimerkiksi nopeuttaa älylaitteiden käyttöönottoa ja parantaa tietoturvaa, koska laitteiden hallinta olisi silloin kytketty käyttäjän kasvojen tunnistetietoihin. Laitevalmistajien tulisi myös panostaa enemmän laitteiden yhteensopivuuteen, ja hinnat pitäisi saada sille tasolle, että ne houkuttelisivat ostamaan. Tätä hetkeä saadaan kuitenkin vielä odottaa.

5.3 Järjestelmän parannussuunnitelmat

5.3.1 Parannukset laitteistolle

Työn edetessä ja sen tullessa valmiiksi huomattiin useita laajennus- ja kehitysideoita käytetyille laitteille. Laitteeseen on mm. tilattu ZigBee-tikku (Kanters 2019), jonka avulla voidaan hallita erilaisia ZigBee-protokollaa käyttäviä laitteita. Tämän lisäyksen myötä esimerkiksi Ikean, Philipsin ja Xiaomin älylaitteita voidaan yhdistää järjestelmään ilman niiden omia keskusyksiköitä. Lisäksi kotiin olisi tarkoitus ostaa kiinteästi asennettava näyttö, josta kodin laitteiden tilaa voidaan tarkkailla ja niiden toimintaa ohjata.

Jatkossa tavallisten sähkölaitteiden ”päälle/päältä pois”-tyyppistä ohjaamista tullaan kokeilemaan myös langattomilla pistorasioilla. Tällä hetkellä kastelujärjestelmä ja kasvivalot toimivat vain Raspberry Pi:n ohjaaman pistorasian kautta, vaikka tarvetta järjestelmien erillään olemiseen tulee varmasti.

Kasvien seurannasta puuttuu kokonaan kasvualustan pH-arvon mittaus. Se on hyvin oleellinen muuttuja, joka vaikuttaa kasvien kykyyn ottaa ravinteita kasvualustasta. pH-mittaukseen on saatavilla useita sopivia sensoreita, joita tullaan kokeilemaan järjestelmässä.

5.3.2 Parannukset ohjelmille ja tietoturvalle

Ohjelmistokokonaisuutta tullaan kehittämään siten, että LabVIEW-ohjelmoinnilla tehdyt ratkaisut poistetaan kokonaisuudessaan. Tämä siksi, että samat ohjelmat voidaan tehdä myös Node-RED-ohjelmaa käyttäen ja LabVIEWin tuki Raspberry Pi -tietokoneita kohtaan näyttää olevan nykyisellään heikko (ei tukea Raspberry Pi 4:lle). Node-REDillä tehdyt ohjelmat ovat myös helpommin muokattavissa ja yhteen liitettävissä muiden protokollien kanssa. Node-RED tuli ensikerran projektiin mukaan siinä vaiheessa, kun LabVIEW-ohjelman prosessoimia arvoja haluttiin ohjata InfluxDB-tietokantaan. Huomattiin, että Node-REDissä kyseinen vaihe onnistuu huomattavasti helpommin. Myös releiden ohjaus, komentorivin suoritteet, ohjelman siirtäminen Raspberry Pi:lle ja ohjelmien vianmääritys onnistui Node-RED-ohjelmalla yksinkertaisemmin.

Myös työssä käytetystä JSON-syötteestä tullaan luopumaan, ja jatkossa kaikki ohjelmien väliset viestit kulkevat MQTT-protokollan mukaisin viestein. MQTT-protokollan käytön myötä mm. viestien toimitusvarmuuden pitäisi parantua huomattavasti ja yhdistettävyyden muihin sovellutuksiin helpottua entisestään, koska useat sensoriohjelmat on suunniteltu valmiiksi MQTT-protokollan käyttöön. Vanha JSON-syöte oli myös tietoturvan kannalta ongelmallinen, koska se sisälsi kaiken tiedon mm. IP-kamerasta, ohjauskäskyistä ja kodin eri sensoreiden tilasta.

Tietoturvaa tullaan kehittämään myös siten, että internetistä laitteelle kirjautuessa on mahdollista syöttää kirjautumistunnukset vain muutaman kerran, minkä jälkeen kirjautuminen sulkeutuu. Tällä hetkellä tietoturvasta pitää huolta vain Homebridgen ja Grafanan vaatima kirjautuminen, jotka voidaan murtaa salasananmurto-ohjelmalla.

On myös yleisesti tiedossa, että Raspberry Pi:n käytössä SD-kortit korruptoituvat vuosien mittaan. Tämän takia on harkittu tietokannan siirtämistä maksulliseen palvelinhotelliin, mikä toisi huomattavasti lisää turvaa ja laskentatehoa kerätyn datan käsittelyyn.

Lähteet

AcuRite. 2019. Instruction Manual for AcuRite Tower Sensor. Verkkodokumentti. <https://www.acurite.com/media/manuals/06002RM-instructions.pdf>. Luettu 16.10.2019.

Alve, Jukka. 2018. SESKO-akatemian opiskeluaineistot. Verkkodokumentti. <https://www.sesko.fi/files/1037/IoT-standardointi.pdf>. Luettu 20.10.2019.

Spiess, Andreas. 2017. IKEA Tradfri Smart lighting system. Verkkodokumentti. <http://www.sensorsiot.org/145-ikea-tradfri-hack-with-gateway/>. Luettu 20.11.2019.

Apple. 2019. Koti-apin ottaminen käyttöön ja sen käyttäminen. Verkkodokumentti. <https://support.apple.com/fi-fi/HT204893>. Luettu 13.11.2019.

Crisan, Calin. 2016. About motionEyeOS. Verkkodokumentti. <https://github.com/ccrisan/motioneyeos/wiki>. Luettu 28.10.2019.

Denkel, O & Persaud, S. 2018. Active cooling your Raspberry Pi 3. Verkkodokumentti. <https://microsoft.github.io/ELL/tutorials/Active-cooling-your-Raspberry-Pi-3/>. Luettu 14.10.2019.

Dy.fi. 2019. Usein kysytyt kysymykset. Verkkodokumentti. <https://www.dy.fi/page/faq>. Luettu 28.10.2019.

Gardena GmbH. 2019. Gardena-lomakastelusarjan tuote-esittely. Verkkodokumentti. <https://www.gardena.com/fi/tuotteet/kastelu/lomakastelu/lomakastelu-sarja/900902901/>. Luettu 14.10.2019.

Gerber, Anna. 2017. Get started developing IoT solutions. Verkkoaineisto. <https://developer.ibm.com/tutorials/iot-lp101-get-started-develop-iot-home-automation/>. Luettu 12.11.2019.

Grafana Labs. 2019. The analytics platform for all your metrics. Verkkodokumentti. <https://grafana.com/grafana/>. Luettu 10.11.2019.

Hattunen, Keijo. 2015. Talotekniikan oppimisympäristöjen kehittäminen. Insinöörityö. Jyväskylän ammattikorkeakoulu. Automaatioteknologian koulutusohjelma. Jyväskylä.

Helsingin yliopisto. 2009. Tietojenkäsittelytieteiden opetusmateriaali – Tietoliikenteen perusteet. <https://www.cs.helsinki.fi/u/marttine/tilpe/Kalvot/luku1c2.pdf>. Luettu 21.10.2019.

HHCC Plant Technology. 2016. Flower Care User Manual. Verkkodokumentti. <https://fcc.report/FCC-ID/2AJEPHHCCJCY01HHCC/3113866>. Luettu 16.10.2019.

Hjelt, Yrjö. 2019. Poliisi ja Tulli saivat oikeuden automaattiseen kasvojen tunnistamiseen ihmisvirrasta. Verkkodokumentti. <https://yle.fi/uutiset/3-10815487>. Luettu 11.12.2019.

Homebridge. 2018. Homebridge-ohjelman lähdekoodi. Verkkoaineisto. <https://github.com/nfarina/homebridge>. Luettu 11.11.2019.

Ikkala, Tapio. 2015. Autovarkaan unelmasta tuli totta. Verkkodokumentti. <https://www.tekniikkatalous.fi/uutiset/autovarkaan-unelmasta-tuli-totta-voi-murtautua-jokaiseen-kaukolukituksella-varustettuun-autoon/5140b1ac-541e-3e78-921f-fc219d8e79ca>. Luettu 20.11.2019.

InfluxData. 2019. Installation guide for InfluxDB. Verkkodokumentti. <https://docs.influxdata.com/influxdb/v0.9/introduction/installation/>. Luettu 12.11.2019.

Kanters, Koen. 2019. Zigbee2mqtt-ohjelman dokumentaatio. Verkkodokumentti. <https://www.zigbee2mqtt.io/>. Luettu 19.11.2019.

Kivelä, Severi. 2018. Langattoman verkon salauksen murtaminen. Insinöörityö. Lapin ammattikorkeakoulu. Tietojenkäsittelyn ja tietoliikenteen koulutusohjelma. Rovaniemi.

Korhonen, Suvi. 2017. Kasvojentunnistus korvaa lentolipun Helsinki-Vantaalla. Verkkodokumentti. <https://www.tivi.fi/uutiset/kasvojentunnistus-korvaa-lentolipun-helsinki-vantaalla-kokeilu-alkaa/efe472c3-e0cc-3317-ab2a-54f0a026c4c8>. Luettu 11.12.2019.

LabVIEW MakerHub. 2016a. The process of setting up a Raspberry Pi for LabVIEW. Verkkodokumentti. <https://www.labviewmakerhub.com/doku.php?id=learn:tutorials:libraries:linx:3-0:raspberry-pi-setup>. Luettu 11.11.2019.

LabVIEW MakerHub. 2016b. Creating a LabVIEW Web Service and hosting it. Verkkodokumentti. <https://www.labviewmakerhub.com/doku.php?id=learn:tutorials:libraries:linx:3-0:web-services>. Luettu 15.11.2019.

Lehtonen, Tommi. 2018. Asiantuntijablogit - tekoälyn jäljillä. Verkkodokumentti. <https://www.univaasa.fi/fi/blogs/expert/ajatusyhteys/tekoalyn_jaljilla/>. Luettu 4.12.2019.

Lindfors, Jukka. 2018. Yleisradion Oy:n artikkeli Suomen energiakriisistä vuonna 1973. Verkkodokumentti. <https://yle.fi/aihe/artikkeli/2006/09/08/energiakriisi-vuonna-1973>. Luettu 23.11.2019.

Linja-aho, Vesa. 2019. Harrastustoimintaan liittyvät sähkötyöt. Sähköpostiviesti 31.10.2019. Vastaanottaja M. Tampio. Autoelektroniikan lehtorin selvennys insinöörityön kirjoittajalle Metropolian Ammattikorkeakoulussa.

ManoMano. 2019. Verkkokaupan tuote-esittely Gardena-lomakastelusarjasta. Verkkodokumentti. <https://www.manomano.de/p/gardena-urlaubsbewaesserung-set-01265-20-6116967>. Luettu 14.10.2019.

Mi Flora Plant Sensor MQTT Client/Daemon. 2019. Ohjelman lähdekoodi. Verkkodokumentti. <https://github.com/ThomDietrich/miflora-mqtt-daemon#xiaomi-mi-flora-plant-sensor-mqtt-clientdaemon>. Luettu 28.10.2019.

Miinin, Elisa. 2019. Tulevaisuuden koti ajattelee puolestamme. Verkkodokumentti. <https://www.pirkka.fi/artikkeli/tulevaisuuden-koti-ajattelee-puolestamme>. Luettu 11.12.2019.

Netatmo. 2019. Additional smart radiator valve. Verkkodokumentti. <https://www.netatmo.com/en-eu/energy/additional-valve>. Luettu 28.11.2019.

Node-RED. 2019a. About Node-RED. Verkkodokumentti. <https://nodered.org/about/>. Luettu 14.12.2019.

Node-RED. 2019b. A Node-RED node to write and query data from an influxdb time series database. Verkkodokumentti. <https://flows.nodered.org/node/node-red-contrib-influxdb>. Luettu 12.11.2019.

Pajari, Katariina. 2019. Kiinassa valmistetaan ihmiskasvoista niin tarkkoja kopioita, että ne huijaavat kasvojentunnistusta. Verkkodokumentti. <https://www.hs.fi/ulkomaat/art-2000006119345.html>. Luettu 11.12.2019.

Pertilä, Timo. 2019. Älyvalojen kesto – IKEA Trådfri. Verkkodokumentti. <https://timo-pertila.com/2019/08/12/25-vuotta-alyvaloja-kestaako-ne/>. Luettu 20.11.2019.

Plötz, Henryk. 2015. Reverse-Engineering a low-cost USB CO_2 monitor. Verkkodokumentti. <https://hackaday.io/project/5301-reverse-engineering-a-low-cost-usb-co-monitor/log/17909-all-your-base-are-belong-to-us>. Luettu 28.10.2019.

Python Software Foundation. 2013. UDP Communication. Verkkodokumentti. <https://wiki.python.org/moin/UdpCommunication>. Luettu 28.10.2019.

Raspberry Pi Foundation. 2019a. About Us & Products. Verkkodokumentti. <https://www.raspberrypi.org>. Luettu 13.10.2019.

Raspberry Pi Foundation. 2019b. Temperature ranges for Raspberry Pi. Verkkodokumentti. <https://www.raspberrypi.org/documentation/faqs/>. Luettu 15.10.2019.

RTL_433. 2019. Ohjelman lähdekoodi. Verkkodokumentti. <https://github.com/merbanan/rtl_433>. Luettu 21.10.2019.

RTL-SDR.COM. 2019. RTL-SDR vastaanottimen kotisivut. Verkkodokumentti. <https://www.rtl-sdr.com>. Luettu 13.10.2019.

Savonen, Sonia. 2019. Mikä kaikki voi mennä oikeasti pieleen, kun ostat älylaitteen. Verkkodokumentti. <https://www.tivi.fi/uutiset/mika-kaikki-voi-menna-oikeasti-pieleen-kun-ostat-alylaitteen-kannattava-ostos-on-laite-joka-ei-ole-riippuvainen-nettiyhteydesta/08612ba2-bf26-453a-b97e-80d210ea85a5>. Luettu 9.12.2019.

Seppä, Heikki. 2017. VTT Blog - Onko esineiden internet oikeasti palveluiden internet. Verkkodokumentti. <https://vttblog.com/2017/06/22/onko-esineiden-internet-palveluiden-internet/>. Luettu 20.10.2019.

Siren, Jukka. 2019. Ihminen päättää, automaatio säätää. Verkkodokumentti. <https://www.kiinteistolehti.fi/ihminen-paattaa-automaatio-saataa/>. Luettu 20.10.2019.

Suomen Lämpömittari Oy. 2019. 9050 Hiilidioksidimittari. Verkkodokumentti. <https://www.suomenlampomittari.fi/hiilidioksidimittari/>. Luettu 16.10.2019.

Suomäki, J & Vepsäläinen, S. 2018. Talotekniikan automaatio – Käyttäjän opas. 4., muuttamaton painos. Helsinki: Kiinteistöalan Kustannus Oy ja kirjailijat.

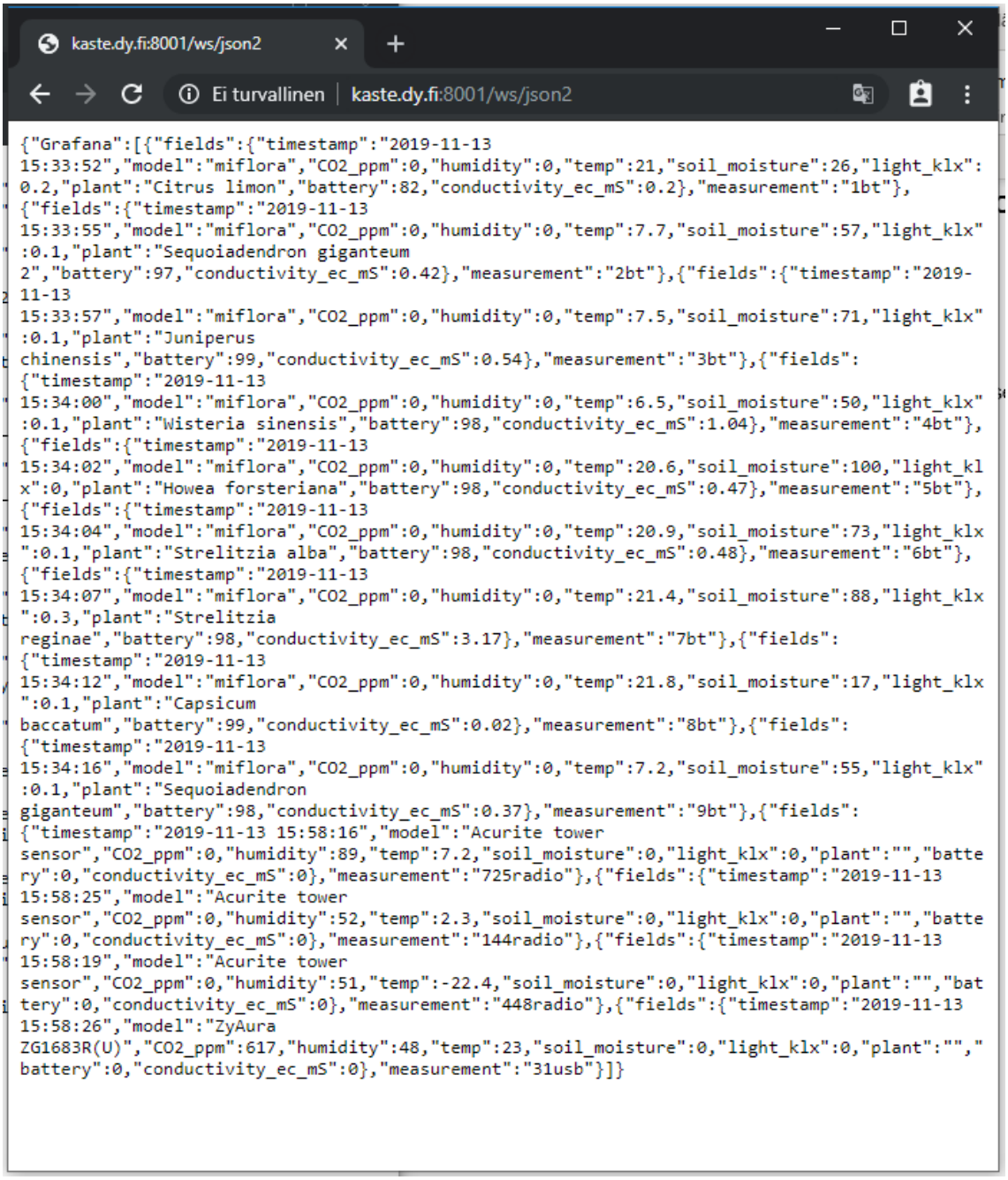
Sähköturvallisuuslaki. 2016. 16.12.2016/1135, § 56.

Tampio, Tommi. 2019. Kaste 2.0 käyttöliittymän lähdekoodi. Verkkodokumentti. <https://github.com/stutommi/kaste_2.0-backend>. Luettu 14.11.2019.

Xmillis.com. 2019. Verkkodokumentti. Muunto-ohjelma kalenteriajasta millisekunteihin. <https://currentmillis.com/>. Luettu 12.11.2019.

ZyAura. 2019. General Introduction for CO2 Monitor. Verkkodokumentti. <http://www.zyaura.com/products/ZG1683R.asp>. Luettu 16.10.2019.

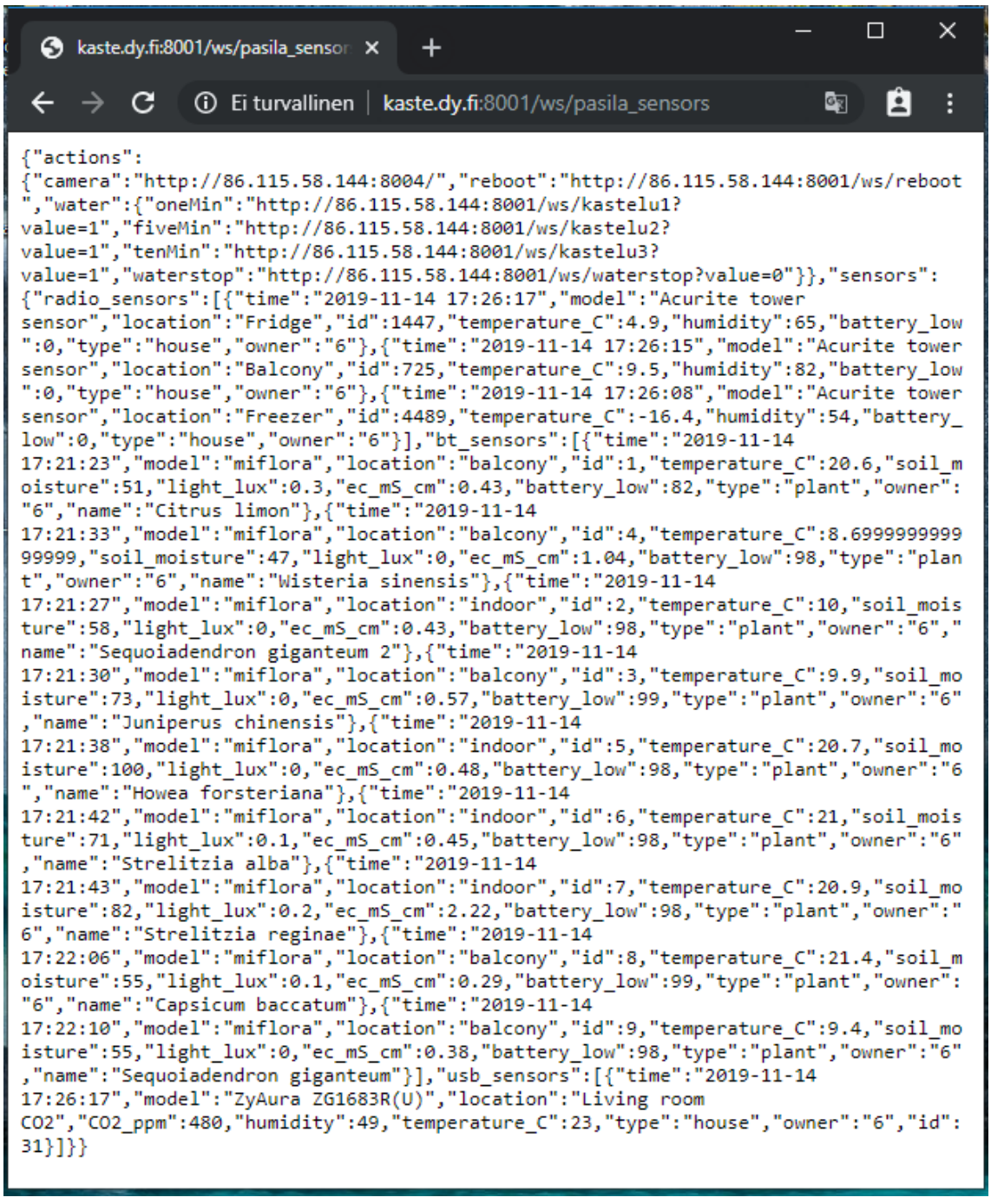
LabVIEW-ohjelman JSON-syöte Grafanaa varten



{"Grafana":[{"fields":{"timestamp":"2019-11-13
15:33:52","model":"miflora","CO2_ppm":0,"humidity":0,"temp":21,"soil_moisture":26,"light_klx":
0.2,"plant":"Citrus limon","battery":82,"conductivity_ec_mS":0.2},"measurement":"1bt"},
{"fields":{"timestamp":"2019-11-13
15:33:55","model":"miflora","CO2_ppm":0,"humidity":0,"temp":7.7,"soil_moisture":57,"light_klx"
:0.1,"plant":"Sequoiadendron giganteum
2","battery":97,"conductivity_ec_mS":0.42},"measurement":"2bt"},{"fields":{"timestamp":"2019-
11-13
15:33:57","model":"miflora","CO2_ppm":0,"humidity":0,"temp":7.5,"soil_moisture":71,"light_klx"
:0.1,"plant":"Juniperus
chinensis","battery":99,"conductivity_ec_mS":0.54},"measurement":"3bt"},{"fields":
{"timestamp":"2019-11-13
15:34:00","model":"miflora","CO2_ppm":0,"humidity":0,"temp":6.5,"soil_moisture":50,"light_klx"
:0.1,"plant":"Wisteria sinensis","battery":98,"conductivity_ec_mS":1.04},"measurement":"4bt"},
{"fields":{"timestamp":"2019-11-13
15:34:02","model":"miflora","CO2_ppm":0,"humidity":0,"temp":20.6,"soil_moisture":100,"light_kl
x":0,"plant":"Howea forsteriana","battery":98,"conductivity_ec_mS":0.47},"measurement":"5bt"},
{"fields":{"timestamp":"2019-11-13
15:34:04","model":"miflora","CO2_ppm":0,"humidity":0,"temp":20.9,"soil_moisture":73,"light_klx
":0.1,"plant":"Strelitzia alba","battery":98,"conductivity_ec_mS":0.48},"measurement":"6bt"},
{"fields":{"timestamp":"2019-11-13
15:34:07","model":"miflora","CO2_ppm":0,"humidity":0,"temp":21.4,"soil_moisture":88,"light_klx
":0.3,"plant":"Strelitzia
reginae","battery":98,"conductivity_ec_mS":3.17},"measurement":"7bt"},{"fields":
{"timestamp":"2019-11-13
15:34:12","model":"miflora","CO2_ppm":0,"humidity":0,"temp":21.8,"soil_moisture":17,"light_klx
":0.1,"plant":"Capsicum
baccatum","battery":99,"conductivity_ec_mS":0.02},"measurement":"8bt"},{"fields":
{"timestamp":"2019-11-13
15:34:16","model":"miflora","CO2_ppm":0,"humidity":0,"temp":7.2,"soil_moisture":55,"light_klx"
:0.1,"plant":"Sequoiadendron
giganteum","battery":98,"conductivity_ec_mS":0.37},"measurement":"9bt"},{"fields":
{"timestamp":"2019-11-13 15:58:16","model":"Acurite tower
sensor","CO2_ppm":0,"humidity":89,"temp":7.2,"soil_moisture":0,"light_klx":0,"plant":"","batte
ry":0,"conductivity_ec_mS":0},"measurement":"725radio"},{"fields":{"timestamp":"2019-11-13
15:58:25","model":"Acurite tower
sensor","CO2_ppm":0,"humidity":52,"temp":2.3,"soil_moisture":0,"light_klx":0,"plant":"","batte
ry":0,"conductivity_ec_mS":0},"measurement":"144radio"},{"fields":{"timestamp":"2019-11-13
15:58:19","model":"Acurite tower
sensor","CO2_ppm":0,"humidity":51,"temp":-22.4,"soil_moisture":0,"light_klx":0,"plant":"","bat
tery":0,"conductivity_ec_mS":0},"measurement":"448radio"},{"fields":{"timestamp":"2019-11-13
15:58:26","model":"ZyAura
ZG1683R(U)","CO2_ppm":617,"humidity":48,"temp":23,"soil_moisture":0,"light_klx":0,"plant":"","
battery":0,"conductivity_ec_mS":0},"measurement":"31usb"}]}

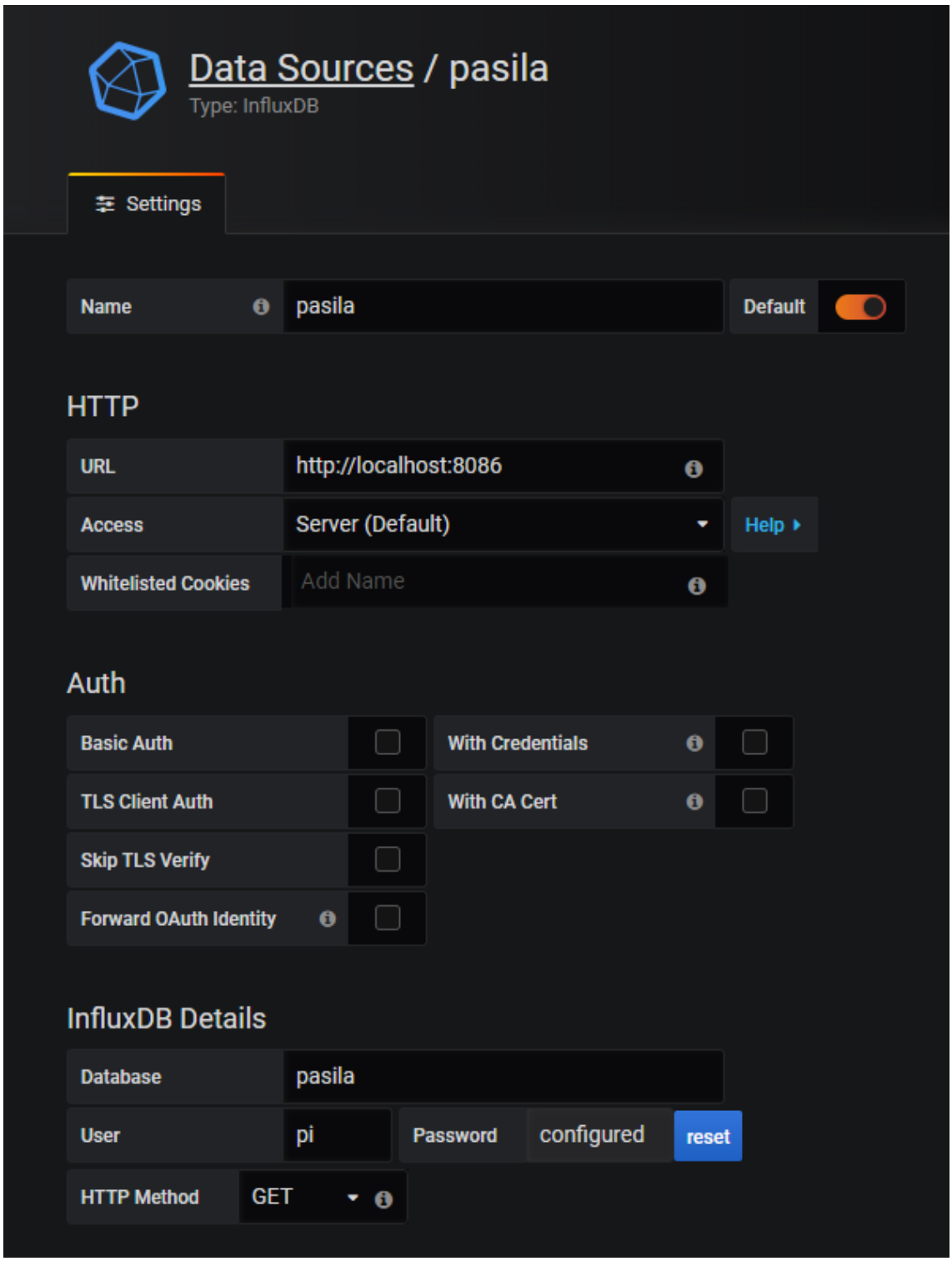
Kuvassa influxdb batch-solmun ohjeiden mukaan muodostettu JSON-syöte. Viesti sisältää LabVIEWin prosessoimat sensoriarvot. Sitä ei ole tarkoitettu ihmissilmin luettavaksi, mutta sitä voidaan tarvittaessa käyttää esim. vian paikantamiseen.

LabVIEW-ohjelman JSON-syöte kaikilla toiminnoilla



Tämä JSON-syöte tehtiin ulkopuolisella palvelimella toimivaa Kaste 2.0 käyttöliittymää varten (Tampio 2019). Syötettä käytettiin hyväksi myös vikojen etsintään ja LabVIEWiin tehtyjen toimintojen kartoittamiseen.

Asetukset InfluxDB-tietokannan yhdistämisestä Grafanaan



Javascript-ohjelmakoodi painonappien luomiseen Grafanaan

```
<script language="javascript" type="text/javascript"
src="js/jquery.js"></script>
<script type="text/javascript">
      $(function() {
            function fetchData(){}

            fetchData();

            $('#1').click(function(){
                  $.ajax({
                        url: "http://192.168.1.227:8001/ws/kastelu1",
                        type: "GET",
                        data: "value=1",
                  });
            });

            $('#2').click(function(){
                  $.ajax({
                        url: "http://192.168.1.227:8001/ws/kastelu2",
                        type: "GET",
                        data: "value=1",
                  });
            });

            $('#3').click(function(){
                  $.ajax({
                        url: "http://192.168.1.227:8001/ws/kastelu2",
                        type: "GET",
                        data: "value=1",
                  });
            });

            $('#4').click(function(){
                  $.ajax({
                        url: "http://192.168.1.227:8001/ws/waterstop",
                        type: "GET",
                        data: "value=0",
                  });
            });

            $('#5').click(function(){
                  $.ajax({
                        url: "http://192.168.1.227:8001/ws/reboot",
                        type: "GET",
                        data: "value=1",
                  });
            });
      });

</script>



<style>
.button {
  position: relative;
  background-color: #4CAF50; /* Green */
  border: 1,5px solid green;
```

```
  font-size: 20px;
  color: white;
  padding: 9px;
  width: 120px;
  text-align: center;
  -webkit-transition-duration: 0.4s; /* Safari */
  transition-duration: 0.4s;
  text-decoration: none;
  overflow: hidden;
  cursor: pointer;

}

.button2 {background-color: #4CAF50;} /* Green */
.button3 {background-color: #f44336;} /* Red */

.button:after {
  content: "";
  background: #f1f1f1;
  display: block;
  position: absolute;
  padding-top: 300%;
  padding-left: 350%;
  margin-left: -20px !important;
  margin-top: -120%;
  opacity: 0;
  transition: all 0.8s
}

.button:active:after {
  padding: 0;
  margin: 0;
  opacity: 1;
  transition: 0s
}
</style>
  <button id="1" class="button button2">1 min</button>
  <button id="2" class="button button2">5 min</button>
  <button id="3" class="button button2">10 min</button>
  <button id="4" class="button button3">STOP</button>
  <button id="5" class="button button3">REBOOT</button>
```

Sisältö config.json-tiedostoon

```
{
    "bridge": {
        "name": "Homebridge",
        "username": "CC:22:3D:E3:CE:30",
        "port": 51826,
        "pin": "031-45-154"
    },
    "accessories": [],
    "platforms": [
        {
            "name": "Config",
            "port": 8080,
            "auth": "form",
            "theme": "auto",
            "restart": "sudo -n systemctl restart homebridge",
            "tempUnits": "c",
            "sudo": true,
            "log": {
                "method": "systemd",
                "service": "homebridge"
            },
            "platform": "config"
        }
    ]
}
```

Sisältö homebridge-tiedostoon

```
# Defaults / Configuration options for homebridge
# The following settings tells homebridge where to find the config.json file
and where to persist the data (i.e. pairing and others)
HOMEBRIDGE_OPTS=-U /var/homebridge -I

# If you uncomment the following line, homebridge will log more
# You can display this via systemd's journalctl: journalctl -f -u homebridge
# DEBUG=*
```

Sisältö homebridge.service-tiedostoon

```
[Unit]
Description=Node.js HomeKit Server
After=syslog.target network-online.target

[Service]
Type=simple
User=homebridge
EnvironmentFile=/etc/default/homebridge
ExecStart=/usr/bin/homebridge $HOMEBRIDGE_OPTS
Restart=on-failure
RestartSec=10
KillMode=process

[Install]
WantedBy=multi-user.target
```