

Tarmo Merimaa

Seuranta- ja hälytysjärjestelmä viljan kuivaukseen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

28.4.2018

Tekijä Otsikko	Tarmo Merimaa Seuranta- ja hälytysjärjestelmä viljan kuivaukseen
Sivumäärä Aika	27 sivua 28.4.2018
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Sulautettu tekniikka
Ohjaaja	Lehtori Keijo Länsikunnas
<p>Insinööritöiden tarkoituksena oli suunnitella ja rakentaa järjestelmä, joka mahdollistaa lämpötilan seurannan ja tallentamisen. Järjestelmän piti myös lähettää sähköpostihälytyksiä, kun tietyt ehdot täyttyivät, jolloin pystyttiin automatisoimaan osa kuivausprosessiin liittyvistä työnvaiheista. Työ tehtiin maanviljelijälle.</p> <p>Seuranta- ja kuivausjärjestelmä rakennettiin käyttäen Raspberry Piä ja DS18B20-antureita. Anturit kytkettiin Raspberry Piin One-wire-tekniikan avulla. Raspberry Pi luki anturien lämpötilat tasaisin väliajoin ja tallensi ne SQLite-tietokantaan.</p> <p>Työ aloitettiin tallentamalla lämpötiloja SQLite-tietokantaan, minkä jälkeen lisättiin ensimmäinen näyttö työssä tehtyyn laitteeseen. Viljan kuivausprosessi eroaa viljalajikkeesta riippuen, minkä takia laitteeseen lisättiin napit ja toinen näyttö. Tämä mahdollisti helpon tavan muuttaa asetuksia, jotka aiheuttavat sähköpostihälytyksen.</p> <p>Koekäytössä haasteet olivat viljan kuivurin pölyisyys ja ympäristö. Viljan kuivurin ovia pidettiin aina auki, mikä johti siihen, että laitteisto oli ulkoilmassa ja asiakkaan kanssa arveltiin, että linnut aiheuttivat napeille vaurioita, minkä vuoksi pöly pääsi nappien sisälle ja tämä johti nappien toimimattomuuteen. Pölyä pääsi myös kotelon sisälle, mikä todennäköisesti aiheuttaa ongelmia pidemmällä aikavälillä. Antureiden pitkät johdot keräsivät myös häiriöitä ympäristöstä, mikä aiheutti ongelmia.</p> <p>Tuotteen prototyyppi on ollut koekäytössä kesällä 2016 ja 2017 ja tuotteen kehitystä jatketaan. Jatkossa palataan tutkimaan langattomia antureita, jotka mahdollistaisivat itse laitteiston siirtämisen kauemmaksi viljan kuivurista ja lähemmäksi internetyhteyden omaavaa rakennusta.</p>	
Avainsanat	sulautettu ohjelmointi, SQLite, Raspberry Pi, DS18B20

Author Title	Tarmo Merimaa Monitoring and alarm system for crop drying process
Number of Pages Date	27 pages 28 April 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Embedded Systems
Instructor	Keijo Länsikunnas, Senior Lecturer
<p>The purpose of the thesis was to design and build a system, which allows temperature monitoring and recording. The system also had to send e-mail alerts when certain conditions were met to allow automation of some of the work steps associated with the drying process. The work was done to the farmer.</p> <p>The monitoring and drying system was built using the Raspberry Pi and DS18B20 sensors. The sensors were connected to Raspberry Pi by One-wire technology. Raspberry Pi read the temperature of the sensors at regular intervals and stored them in the SQLite database.</p> <p>The project was started by storing the temperatures in the SQLite database, after which the first display was added to the job. The grain drying process differs depending on the grain variety, which is why the buttons and the second display were added to the work. This provided an easy way to change settings that cause an e-mail alert.</p> <p>Challenges in pilot testing were the dustiness and environment of the crop dryer. The grain dryer door was always kept open, which led to the fact that the equipment was in the open air and it was thought that the birds caused damage to the buttons, causing the dust to get inside the buttons that led to the failure of the buttons. Dust was also inside the enclosure, which is likely to cause problems in the longer term. The long wires of the sensors also collected disturbances from the environment, which caused problems.</p> <p>The prototype of the product has been in trial for 2016 and 2017 in the summer and product development continues. In the future, we will return to explore wireless sensors, which will enable the hardware to be moved away from grain drying and closer to an internet-connected building.</p>	
Keywords	embedded programming, Raspberry Pi, SQLite, DS18B20

Sisällys

Lyhenteet

1	Johdanto	6
2	Viljan kuivausprosessi	7
3	Insinööriyössä käytetty tekniikka	8
3.1	Laitteet ja komponentit	8
3.2	Ohjelmat	14
4	Seuranta- ja valvontalaitteen toteutus	16
4.1	Ohjelma	17
4.2	Piirilevy	24
4.3	Kotelo	27
4.4	Ratkaisut	28
4.5	Käyttöönotto ja jatko	29
5	Yhteenveto	32
	Lähteet	33

Lyhenteet

LCD	Liquid crystal display. Nestekidenäyttö.
UTF-8	Unicode. Merkistöstandardi.
IDE	Integrated development environment. Ohjelmointiympäristö, jolla voidaan vähintään editoida ja kääntää ohjelmakoodi.
GPIO	General-purpose input/output. Ennalta määräämätön pinni (NASTA), jonka käyttötapaa voidaan itse määrittää.
SSH	Secure Shell. Salattu yhteys, jota käytetään yleisimmin kirjautumiseen etälaitteeseen.
VPN	Virtual private network. Salattu yhteys kahden verkkolaitteen välillä.

1 Johdanto

Insinööriyön tarkoituksena on tehdä viljankuivausprosessiin valvontajärjestelmä, joka tallentaa lämpötilavaihtelut ja ilmoittaa prosessin vaiheesta sähköpostitse. Järjestelmälle syötetään eri prosessin vaiheissa tavoitelämpötilat ja rajat, joiden sisällä lämpötilan on pysyttävä, ja poikkeustilanteessa lähetetään sähköpostiviesti, jolloin pystytään puuttumaan muutokseen välittömästi. Prosessin vaiheesta ilmoitetaan myös sähköpostitse, jolloin vältytään tilanteilta, joissa laitteisto on tyhjäkäynnillä. Järjestelmä on myös helposti ylläpidettävä ja laajennettavissa. Modulaarinen rakenne mahdollistaa muun muassa anturien määrän kasvattamista tarvittaessa ja uusien ominaisuuksien lisäämistä.

Työ tehdään maatilalle, jossa vesi pyritään lämmittämään itse ja kylmä vesi otetaan kaivosta. Veden lämmitykseen käytetään hakepoltinta, joka käyttää polttoaineena puusta saatavaa haketta. Hakepoltin sijaitsee erillisessä rakennuksessa, jonka takana on hakesäiliö. Hakesäiliön pohjalla on liukuhihna, joka kuljettaa haketta polttokammioon. Hakepoltin on automaattinen, eli energian tuottoa ei tarvitse itse ohjata vaan hakepoltin säättää energian tuoton lämpövesisäiliön lämpötilan perusteella.

Etäisyydet ovat työpisteiden välillä pitkiä, minkä takia etävalvonta ja hälytysjärjestelmä tehostaisivat työntekoa, koska ei ole tarvetta käydä kuivurilla säännöllisesti tarkastamassa tilannetta. Vilja kuivataan siilossa, jossa on mahdollista kierrättää ilmaa. Kuivurissa on tuuletin, joka puhaltaa ilman radiaattorin läpi, joka lämpenee hakepolttimen lämmittämällä vedellä. Lämmin ilma ohjataan siilon pohjaan ja siitä ylöspäin siilon. Samalla kun lämmin ilma kulkee viljan läpi, se nostaa viljan lämpötilaa ja nopeuttaa veden haihtumista viljasta. Kun päästään tavoitelämpötilaan, pystytään arvioimaan tarkasti viljan kosteus, ja se varmistetaan myös mittamaalla jyvien kosteus erillisellä laitteella. Tällöin siilon katossa oleva luukku avataan ja siilon puhalletaan ulkoilmaa, jolloin vilja alkaa jäähtyä. Eri viljalajikkeilla on eri varastointilämpötilat, ja tavoitelämpötilaan jäähtyttyä vilja siirretään varastoon odottamaan kuljetusta eteenpäin.

Tilalla on käytössä vanha laitteisto, ja nykyisen järjestelmän automatisointi vaatisi kuivurin täydellisen remontin ja uusimisen, mikä aiheuttaisi kymmenien tuhansien eurojen lisäkulut. Tämä insinööriyö mahdollistaa osittaisen automaation huomattavasti edullisemmin.

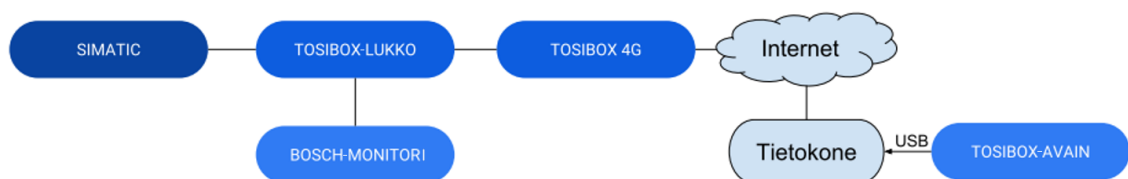
2 Viljan kuivausprosessi

Viljan kuivausprosessissa on kolme vaihetta, jotka ovat valvontajärjestelmälle oleellisia. Prosessissa on tyhjä vaihe, jolloin kuivuri on sammuksissa eikä viljaa ole kuivumassa. Tästä vaiheesta ei tarvita tallennustietoja, ja se pitää huomioida, kun automatisoidaan ehtoja, joilla tietoja tallennetaan.

Lämmitysvaiheessa viljan lämpötilaa nostetaan kuumalla ilmalla, jolloin kosteutta haihtuu viljasta ja ilma jäähtyy. Kun poistoilma saavuttaa viljaan vietävän ilman lämpötilan, tiedetään, että on tavoitettu haluttu kosteus, ja avataan poistoilmaluukku. Jäähdytysvaiheessa viljan lämpötilaksi halutaan varastointilämpötila, joka on väliltä 20–25 astetta lajikkeesta riippuen. Viljan jäähtyttyä se voidaan siirtää varastoon säilytystä ja myyntiä varten.

Kuivausprosessin automaatiota tarjoavat muun muassa Siemensin SIMATIC-tuotesarja ja Mitsubishi Electric. Nämä laitteet perustuvat logiikkapiireihin, jotka ostetaan moduuleina ja asentaja asettaa perusasetukset kohdilleen. Järjestelmien muutokset myöhemmässä vaiheessa vaativat jälleenmyyjän palveluita, sillä uudelleenohjelmointi vaatii järjestelmän oman ohjelmointisovelluksen, ja usein järjestelmät on mitoitettu juuri siihen, mitä ostaja haluaa, eli tarvitaan myös lisämoduuleita. [1; 2.]

Asiakkaan maatilalla hakepolttimen automaatio on toteutettu Simatic-laitteilla. Simatic syöttää haketta polttouuniin tarpeen mukaan ja ylläpitää tulta. Simatic tarjoaa laitteilleen myös etähallinnan, joka on toteutettu 3G/4G-verkon ja VPN:n välityksellä. VPN-laitteena Simatic käyttää Tosibox-laitteita kuvan 1 mukaisesti.



Kuva 1. Hakepolttimen ohjaus Simatic-laitteella asiakkaan maatilalla.

Tosibox-avain on muistitikun näköinen laite, joka paritetaan Tosibox-lukon kanssa laittamalla avain Tosibox-lukon USB-porttiin. Tällöin lukko kopioi salausavaimen

Avaimelle ja tämän jälkeen avain on käyttövalmis. Avain tukee useampaa Tosibox-lukkoa. VPN-yhteys muodostetaan kytkemällä avain tietokoneeseen, ja Tosibox-sovelluksesta voi valita kohteen mihin haluaa yhdistää. Tosibox tarjoaa turvallisen tavan kytkeä laite internetiin, sillä Tosibox toimii palomuurina samalla ja sallii VPN-yhteyden ainoastaan salausavainta ja erillistä salasanaa vastaan. [3]

VPN eli Virtual Private Network on tapa, jolla voidaan yhdistää erillisiä verkkoja yhdeksi julkisen verkon läpi. Tämä mahdollistaa esimerkiksi etätoimiston, joka on virtuaalisesti kytketty samaan verkkoon päätoimiston kanssa, ja näin voi käyttää esimerkiksi tiedostopalvelimia helposti.

Simatic-laitteisto toimii ohjelmoitavien logiikkapiirien avulla. Laitte ohjelmoidaan STEP 7 -ohjelmiston avulla. Simatic-yksikkö koostuu kontrollerista ja siihen kytkettävistä moduuleista.

3 Insinööriyössä käytetty tekniikka

3.1 Laitteet ja komponentit

Insinööriyössä oli tarkoituksena rakentaa edullinen luetettava järjestelmä, ja nykyisin harrastelijoille tarkoitettuja kehitysalustoja on tarjolla paljon. Varteenotettava vaihtoehto olisi ollut Particle Photon -moduuli [4], jonka avulla olisi voinut yhdistää anturit langattomasti Raspberry-laitteeseen. Tämä ratkaisu olisi tuottanut käytännön ongelmia virransyötön kanssa, sillä virtapistokkeita ei ollut kohteessa kuin yhdessä paikassa, ja paristoilla toimivat anturit olisivat olleet epäkäytännöllisiä. Particle Photonille olisi myös pitänyt rakentaa kotelo, sillä kohdeympäristö oli erittäin pölyinen.

Raspberry Pi 2

Raspberry Pi on minitietokone eli laite, joka on rakennettu yhdelle piirille ja sen takia erittäin pienikokoinen. Raspberry Pille on olemassa monia eri käyttöjärjestelmiä. Linux-ympäristöjä on eniten tarjolla, mutta myös Microsoft on tuomassa IoT-käyttöjärjestelmää markkinoille. [5; 6.]

Tässä projektissa käytettiin Raspberry Pi 2:ta, joka tarjoaa 40 GPIO-nastaa ja jolla on ARM Cortex-A7-prosessori [7]. Raspberry Pi 2:nastaa 7 tarjoaa 1-wire-tuen, joka on tarvittava ominaisuus projekteissa ja jota kautta luetaan 1-wire-protokollaa käyttäviä DS18B20-antureita [8]. Tähän projektiin valittiin käyttöjärjestelmäksi Linux-pohjainen Debian.

Debianin ydin pidetään ajan tasalla unattended-upgrades-paketin avulla. Unattended-upgrades mahdollistaa ytimen automaattisen päivityksen skriptien avulla. Skriptit päivittävät säännöllisesti kirjastot, ja jos uudempi versio on saatavilla, skripti lataa sen automaattisesti. Asennuksen voi myös automatisoida. [9.]

Raspberry laitetaan omaan suojaavaan muovikoteloon [kuva 2], jotta komponentit eivät kärsisi kulumavaurioita lopullisessa kotelossa. Raspberryyn liitetään myös USB WiFi -kortti, joka mahdollistaa internetiin yhdistämisen ilman verkkojohtoa. Virran laite saa mikro-USB-laturista, joka yhdistetään Raspberryyn mikro-USB:lla. Piirilevyn yhdistäminen tehdään lattakaapelilla. Raspberry Pi 2 laitetaan lopulliseen koteloon, joka tarjoaa lisäsuojan pölyä vastaan.



Kuva 2. Raspberry Pi 2 suojaavassa muovikotelossa.

DS18B20-lämpötila-anturi

Insinööriyössä antureina käytettiin DS18B20-lämpötila-antureita, jotka hankittiin vesitiiviissä paketissa 15 m:n johdolla varustettuna. Nämä anturit tarjoavat tarkan mittauksen ja toimivat digitaalisena 1-wire-protokollan kautta. [Liite 1.] Anturit määritettiin etukäteen ohjelmassa, ja teipillä merkittiin anturien lopullinen sijoituskohde. [Kuva 3.]



Kuva 3. DS18B20-anturi vesitiiviissä kotelossa ja merkittynä.

DS18B20-anturi liitettiin koteloon 3,5 mm:n liittännällä, jossa on 3 napaa, virta, data ja maa [kuva 4]. Liittimessä kärki on virralle, keskiosa datalle ja juuri maalle.

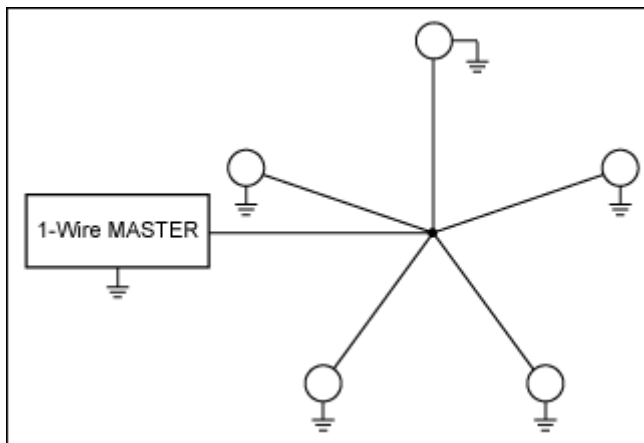


Kuva 4. DS18B20-anturin liitin.

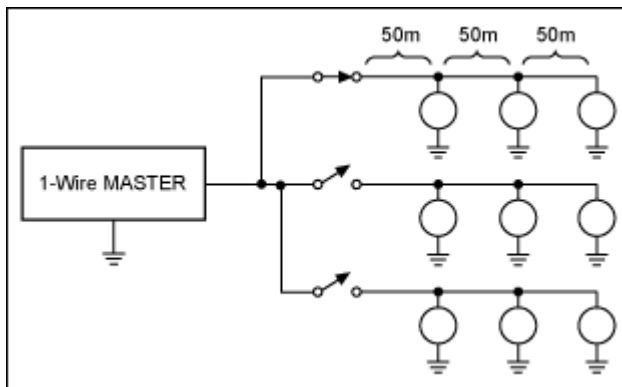
Anturien 15 metriä pitkä johto aiheutti ongelmia, sillä anturin käyttöjännite on alhainen, ja siksi pitkissä johdoissa kulkeva signaali kerää häiriöitä ympäristöstä. Pitkä johto aiheuttaa myös signaalin vääristymiä [10] ja sen vuoksi Raspberry ei pystynyt lukemaan

luotettavasti useampaa kuin kahta anturia. Tämä saatiin korjattua lisäämällä väliin SN74HC4066-piiri, joka poisti ongelman.

Kuvassa 5 on esiteltyä tähtitopologiaa. Tähtitopologiaa ei yleisesti suositella [10] käytettäväksi ilman releitä tai kytkimiä, koska johdoissa on impedanssia, joka aiheuttaa tiedon korruptoitumista, sillä signaali tavoittaa osan antureista myöhemmin, jolloin ne vastaavat pidemmällä viiveellä kuin lähimmät ja signaali sotkeutuu. Suositeltavampaa on käyttää kytkintopologiaa [kuva 6], koska tällöin voidaan kommunikoida yhden johdon takana olevien antureiden kanssa, jolloin minimoidaan impedanssin aiheuttama viive.



Kuva 5. Tähtitopologia [10].



Kuva 6. Kytkin topologia [10].

LCD 20 x 4 -näyttö

Näyttöinä oli kaksi LCD 20 x 4 -nestekidenäyttöä. Niissä on HD44780U-ohjain, jossa on valmiina eri merkistöjä. Näyttöä voidaan käyttää 4-bittisellä ohjauksella ja 8-bittisellä ohjauksella. Projektissa päädyttiin käyttämään näyttöä 4-bittisenä, jotta jäisi enemmän

nastoja Raspberrysta muihin tarkoituksiin. Näytön-ohjain toimii 8-bittisillä komennoilla, eli valinnan vuoksi näytönkirjoitusnopeus puolittui, mutta tässä tapauksessa siitä ei ollut haittaa.

Näyttöjen nastat ovat taulukon 1 mukaiset. Taulukkoon on merkitty myös nastat, joita ei käytetä tässä työssä.

Taulukko 1. LCD-näytön nastat.

Nastan numero	Symboli	Kuvaus	Käytössä
1	VSS	Maa	Kyllä
2	VDD	Käyttöjännite	Kyllä
3	VO	Kontrastin säätö	Kyllä
4	RS	Data / Komentotila	Kyllä
5	R/W	Luku / Kirjoitus	Kyllä
6	E	Aktiivinen	Kyllä
7	DB0	Ohjausbitti	Ei
8	BD1	Ohjausbitti	Ei
9	BD2	Ohjausbitti	Ei
10	BD3	Ohjausbitti	Ei
11	BD4	Ohjausbitti	Kyllä
12	BD5	Ohjausbitti	Kyllä
13	BD6	Ohjausbitti	Kyllä
14	BD7	Ohjausbitti	Kyllä
15	A	Taustavalon käyttöjännite	Kyllä
16	K	Taustavalon maa	Kyllä

Ylemmän näytön tehtävänä [kuva 7] on toimia statusnäyttönä, ja näyttöä voidaan kierrättää eri ikkunoiden välillä painamalla nappia. Näkyville saa lämpötilat ja ohjelman tilan sekä tiedot, onko ohjelma huoltotilassa. Alemman näytön tehtävänä on toimia valikkona, jota hallinnoidaan sen vieressä olevilla napeilla [kuva 7]. Asetuksista voidaan määrittää hälytysrajat eri tilanteissa sekä SQL-kantaan tallennusväli.

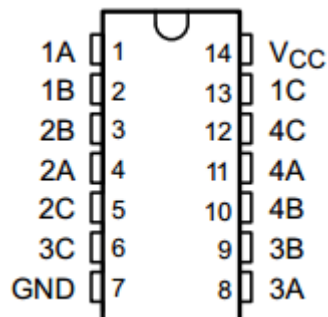


Kuva 7. LCD-näytöt paneelissa.

SN74HC4066-kytkin

SN74HC4066 on kaksisuuntainen analoginen kytkin. Kaksisuuntaisuus on tarvittava ominaisuus, sillä kommunikaatiota tapahtuu 1-wire-tekniikassa kumpaankin suuntaan. Piiri toimii eristävänä anturien välillä.

Kuvassa 8 ovat piirin kytkennät. Vcc on jännite, GND on maa ja A:lla ja B:llä merkityt ovat I/O-portteja eli signaalin sisääntuloja ja ulosmenoja. C:llä merkityt ovat hallintapinnejä eli ne kytkevät A ja B vastineet päälle ja pois.



Kuva 8. SN74HC4066-kytkimen nastat.

4066-piiri toimii kytkimenä kuvan 6 tapaan, eli se mahdollistaa yhteyden sulkemisen antureihin, jolloin ei ole niin paljon häiriötä. Tämä mahdollistaa useamman DS18B20:n kytkemisen pitkällä johdolla. Työssä oli käytössä viisi anturia eli piirejä tarvittiin kaksi, koska yhdessä piirissä on neljä porttia. Kuvassa 9 näkyy piirin paketti.



Kuva 9. SN74HC4066-piiri.

3.2 Ohjelmat

Insinööriyön ohjelmointi tehtiin Windows 10 -ympäristössä käyttäen Microsoft Visual Studio 2015 -ohjelmaa. Lisäosaksi Microsoft Visual Studio 2015:een hankittiin Visual GDB, joka mahdollistaa helpon tavan kääntää koodi Linux-ympäristöön sopivaksi. Ohjelmointi tehtiin C/C++-kielellä. Työssä oli tarvetta tallentaa tietoja ja tietokannaksi valittiin SQLite.

Kehitysympäristö

Kehityksessä käytettiin virtuaalikonetta, joka toimi Hyper-V:n päällä. Tämä mahdollisti fyysisen tietokoneen vaihdon ilman, että kehitysympäristö rikkoontuisi ja vaatisi uudelleenrakentamisen.

Visual Studio

Visual Studio 2015 on Microsoftin kehittämä kehitysympäristö, joka mahdollistaa koodaamisen lisäksi myös tuen testaukselle. Visual Studiioon voi ladata eri kielille omat paketit, jolloin saa peruskirjastot kielelle ja syntaksitarkistuksen. Visual Studiolla voi

kehittää sovelluksia muille alustoille, esimerkiksi Androidille ja iOS:lle, lataamalla siihen tarvittavat paketit. [11.]

VisualGDB

VisualGDB on Visual Studioon asennettava lisäosa, joka mahdollistaa helpon tavan kääntää ohjelma eri alustoille hyödyntäen Cross-Platform-menetelmää. Merkittävin ominaisuus lisäosassa on tuki sulautetuille järjestelmille, Linuxille ja Androidille.

Ohjelman testaukseen VisualGDB:ssä voi kirjoittaa testejä eri osa-alueille, eli voi koodin kirjoitusvaiheessa korjata virheitä, jotka saattavat aiheuttaa ohjelmassa virheitä. Linux-ympäristössä VisualGDB:llä ladataan kohteen kirjastot ja CMakeen laitteelle, jolla ohjelmoidaan, jotta pystytään tarkastamaan syntaksit ja suorittamaan testejä. Lopuksi ohjelma käännetään laitteella, jolla ohjelmaa kehitetään, ja SSH-yhteydellä se siirretään Linux-tietokoneelle, jolla se voidaan suorittaa ja myös testata.

SQLite

SQLite on yhä kehityksen alla oleva kirjasto, joka mahdollistaa tietokannan käytön ympäristöstä riippumatta. SQLite ei vaadi erillistä palvelinta, joten se toimii heti. Sen tietokannan voi siirtää laitteelta toiselle ilman minkäänlaista lisätyötä. Vahvuuksiin kuuluu myös se, että SQLite ei vaadi erillistä prosessia, joka hallitsee kyselyitä, ja tietokannan hallinta tapahtuu suoraan tiedostoon kirjoittamalla tai lukemalla. [12.]

KiCad EDA

KiCad on kokonaisvaltainen piirilevy-suunnittelu-ohjelma. KiCadin käyttöön päädyttiin sen takia, että ohjelma on ilmainen ja helppokäyttöinen. KiCad tukee myös GitHubista komponenttien automaattista hakua, joka tuo käyttöön huomattavan laajan komponenttikirjaston ilman, että pitää suunnitella itse komponentit.

Piirilevyn suunnittelussa ensimmäinen vaihe on kytkentäpiirustuksien teko. Kytkentäpiirustus kertoo sen, miten komponentit on kytketty keskenään. Kytkentäpiirustuksen avulla luodaan net-lista, joka koostuu komponenttien nastojen nimistä ja niiden pareista.

Seuraavaksi kytkentäpiirustuksissa esiintyvät komponentit liitetään oikeisiin komponentteihin. Tämä on tärkeä vaihe, koska samaa komponenttia voi saada monessa eri kotelossa.

Kolmas vaihe on itse piirilevyn suunnittelu. Tämä vaihe on työläs ja vaatii tarkkuutta, sillä se mallintaa fyysistä piirilevyä. Suunnittelussa pitää huomioida muun muassa jännite ja virta, joka tulee kulkemaan vedettävissä linjoissa, sillä liian pieni linja voi johtaa kuumenemiseen ja kuparin rikkoontumiseen. Kun piirilevy on suunniteltu, siitä tehdään gerber-tiedostot, jotka ovat standardi tiedostomuoto piirilevyn valmistuksessa ja joita isoin osa valmistuksessa käytettävistä laitteista tukee.

Gerber-tiedostot pitävät sisällään koordinaatit. Koordinaatteja käytetään suoja-pinnoitteen laittamiseen, jolloin kuparia syövytetään vain halutuista kohdista. Myös suoja-pinnoitteet ja kirjoitettavat tekstit vaativat koordinaatit. Gerber-tiedostot eivät kuitenkaan sisällä porauspisteitä, eli usein joutuu toimittamaan erikseen nämä koordinaatit.

4 Seuranta- ja valvontalaitteen toteutus

Insinööriyössä oli monta eri osa-aluetta, jotka pystyttiin tekemään erillään. Tämä mahdollisti projektin etenemisen, sillä aikaa, kun toinen osa-alue odotti komponentteja tai muita osia. Pisin odotusaika tuli piirilevyn tilaamisessa, sillä sen saapuminen Suomeen kestää 2-4 kk.

Seuranta- ja valvontalaitteen käyttö piti tehdä helpoksi, eli siihen tehtiin yksinkertainen käyttöliittymä ja säätö. Virta ja anturien liittimet piti tehdä kestäviksi, ja tuotteen piti olla mahdollisimman pölytiivis, koska se sijoitetaan erittäin pölyiseen tilaan.

Laitteessa on kaksi näyttöä, joista toinen toimii infonäyttönä. Infonäyttöä hallitaan yhden napin avulla, joka vaihtaa näytettävää infoa. Infotietoja ovat mm. lämpötilat, hälytykset ja ohjelman tila. Päänäyttöä hallitaan kuudella napilla: nuolinäppäimet, OK- ja peruutannappi. Päävalikossa voidaan valita hälytysväli, joka määrää, kuinka usein tarkastetaan hälytysrajat ja lähetetään sähköpostiviesti. Toinen toiminto päävalikossa on tallennusvälin asettaminen, jolla pystytään säätämään SQL-tietokantaan tallentamisen tiheys.

Kolmas valinta vie hälytysrajojen muokkausvalikkoon, ja siellä pystyy määäämään radiaattoriveden hälytysrajan, jolloin saadaan tieto, kun sisään tulevan veden lämpötila laskee liian alhaiseksi. Radiaattorin lämpötilaa seurataan kuivausvaiheen aikana, jolloin tarvitaan lämmintä vettä lämmittämään viljan sekaan puhallettavaa ilmaa.

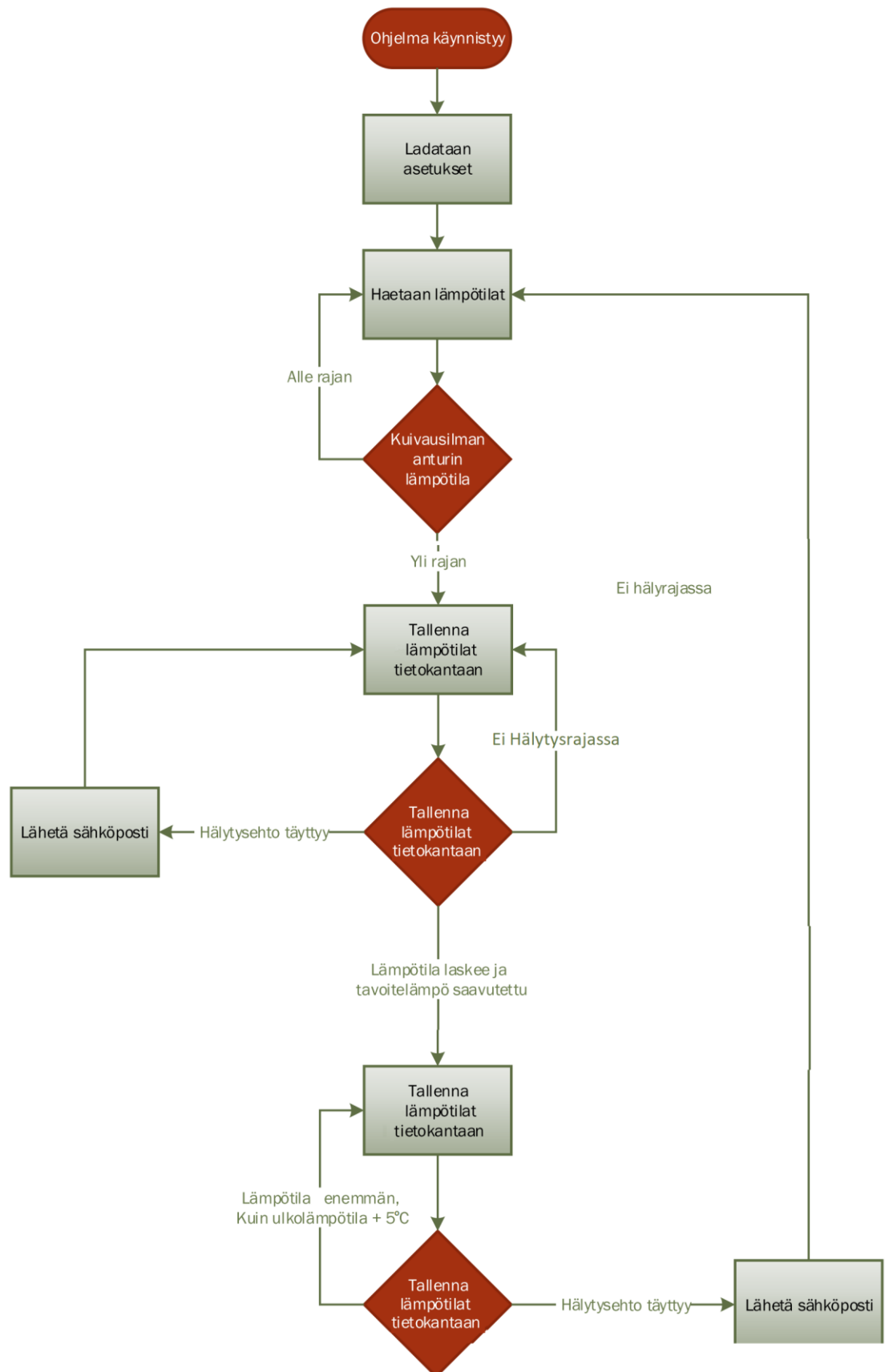
Toinen säädettävä hälytysraja on tavoitelämpö. Se on lämpötila, johon poistoilman halutaan nousevan, ja sitä kautta saadaan tietää, että vilja on kuivunut tarpeeksi. Tavoitelämpöhälytys lähettää myös ennakkohälytyksen astetta ennen tavoitetta, jolloin asiakas voi aloittaa matkan kuivaussiiilolle ennakkoon ja näin säästää aikaa. Tavoitelämpöä seurataan vain kuivausvaiheessa.

Kuumina päivinä saattaa käydä niin, ettei tavoitetta saavuteta ja se halutaan tietää. Kolmas valinta on siis ajan määäryitys sille, kuinka kauan ohjelma sallii kuivauksen pysyä käynnissä, ennen kuin se ilmoittaa asiakkaalle, ettei päästä tavoitteeseen. Tämä hälytys on myös käytössä vain kuivauksessa.

Viimeinen säädettävä hälytys on jäähdytysvaiheessa seurattava lämpötila. Lämpötila-anturi seuraa viljan lämpöä ja ilmoittaa, kun vilja on jäähtynyt haluttuun lämpötilaan. Tämä hälytysraja määää myös toisen hälytyksen, joka ilmoittaa, jos vilja ei voi jäähtyä niin alas. Kun ulkolämpötilan ja halutun lämpötilan ero on alle viisi astetta, ei välttämättä päästä haluttuun lämpötilaan.

4.1 Ohjelma

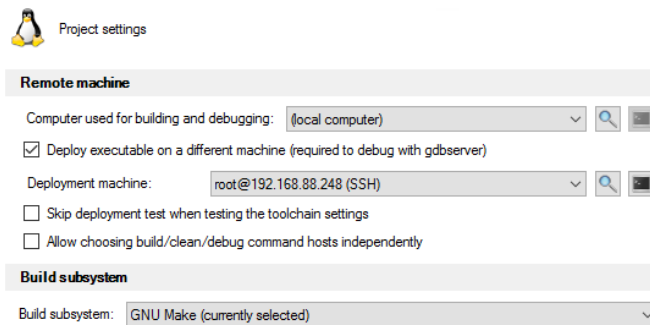
Kuivausohjelman koodi tehtiin modulaarisena ja oliota käyttäen. Tämä mahdollistaa ohjelman eri osien päivittämisen ilman, että koko ohjelmaa pitää muuttaa, koska olioiden rajapinnat voidaan pitää samana. Ohjelmassa on myös lokin kirjoitus, eli ongelmatilanteissa ei tarvitse lähteä tutkimaan paikan päälle, missä tilanteessa ohjelmalle sattui jotain ja minkä takia. Lokitiedostoon kirjoitetaan kaikki merkittävimmät tapahtumat ja muutokset, joissa on riski, että jokin asia aiheuttaa esimerkiksi ohjelman kaatumisen. Lokitiedoston koko pidetään tuhat riviä pitkänä, jotta se ei veisi paljon tilaa, eli toisin sanoen ohjelman tapahtumista viimeiset tuhat pidetään tiedossa. Kuvassa 10 on esitetty ohjelman toimintalogiikka.



Kuva 10. Kuivausohjelman toiminta.

VisualGDB:n käyttöönotto

VisualGDB:n käyttöönotto on tehty helpoksi. Ensimmäiseksi lisätään kohdelaitteen eli tässä tapauksessa Raspberry Pin IP-osoite ja käyttäjätunnus. Kuvassa 11 on esitettyä projektiasetukset, joissa voi myös määrittää, siirretäänkö suoritettava ohjelma toiselle koneelle. Tässä tapauksessa joutuu latamaan kohdelaitteen kirjastot omalle koneelle, jotta saa koodin testauksen toimimaan.

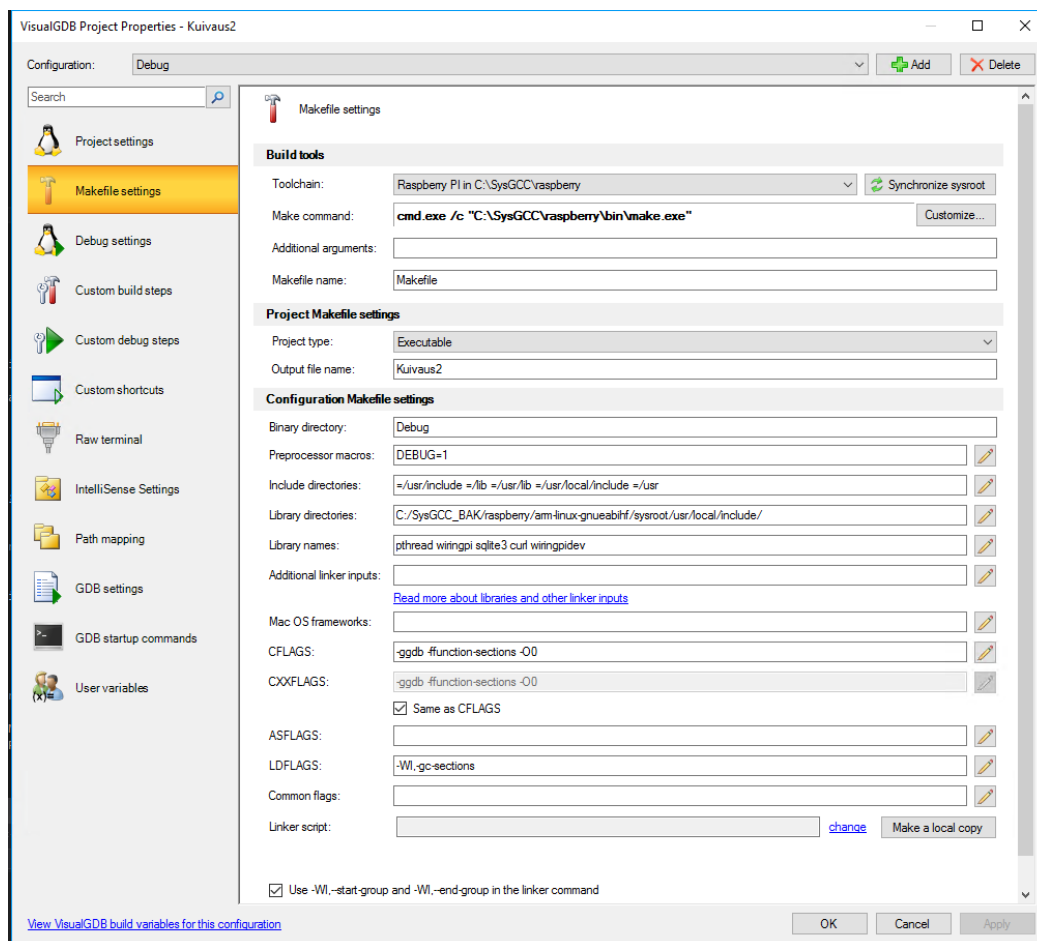


Kuva 11. VisualGDB:n Projektiasetukset-välilehti insinööriyössä.

Seuraavaksi määritetään Make-asetuksia ja käytettävät kirjastot. Ohjelmointityökaluina käytetään tässä työssä paikalliselle koneelle ladattuja GCC-työkaluja, jotka mahdollistavat ohjelmantestauksen ja ohjelman kääntämisen Raspberry Pille paikallisesti. Käännös työkaluihin ei tehdä muita muutoksia, sillä ei ole tarvetta lisätä parametreja. Tämän jälkeen synkronoidaan järjestelmän juuri.

VisualGDB tarjoaa myös mahdollisuuden tehdä kirjastoja Linux-ympäristöön Windows-koneella, mutta tämä insinööriyö on suoritettava ohjelma eli projektityyppinä pidetään kuvan 12 mukaisesti suoritettava ohjelma.

Jos käyttää useita kolmannen osapuolen kirjastoja, voi seuraava vaihe aiheuttaa vaikeuksia, sillä jokaisen kirjaston sijainti pitää tietää ja lisätä Makefilen asetuksiin. Jotkin kirjastot asentuvat oletuksena eri paikkoihin, kuin missä oletuskirjastot ovat. Tässä työssä käytettiin wiringpi-, sqlite3-, curl- ja wiringpdev-kirjastoja. Kun nämä asetukset on tehty, synkronoidaan vielä kerran muut paikat ja tarkastetaan asetusten eheys.



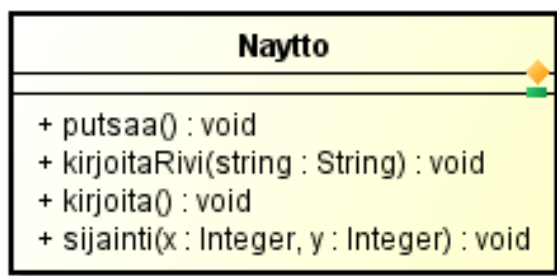
Kuva 12. VisualGDB:n Makefile-asetukset insinööriyössä.

Näyttö

Insinööriyön laitteistossa käytettiin kahta näyttöä, ja työn helpottamiseksi tehtiin funktioita erilaisia tilanteita varten. Näyttöolio käyttää hyväksi WiringPi-kirjastoon kuuluvaa lcd.h-funktioita. Näyttöolioita luotiin kaksi eli kummallekin näytölle oma. Näytöt käyttävät samoja datalinjoja. Ainoa ero olioiden välillä luomisen jälkeen oli enable-pinni, koska sillä voidaan määrittää, kumpi näytöistä on kirjoitustilassa, jolloin toinen näytöistä ei kirjoita tulevaa informaatiota esille vaan hylkää tulevat viestit.

Kuvassa 13 on esitetty näyttöolion rakenne koodissa. Putsaa-funktio tyhjentää näytön merkeistä, jolloin kirjoitettaessa lyhempää tekstiä kuin aiemmin ei jää vanhaa tekstiä esille. KirjoitaRivi-funktiolla saadaan kirjoitettua rivi kerrallaan, jolloin lyhyttä tekstiä kirjoittaessa ei tarvitse huolehtia loppurivillä olevasta tekstistä, sillä se tyhjennetään. Kirjoita-funktio puolestaan kirjoittaa joko numerot, desimaalit, kokonaisluvut tai tekstiä

näytölle kohtaan, jossa osoitin on, eikä koske muuhun, eli tämä soveltuu hyvin tilannetietojen päivittämiseen.



Kuva 13. Naytto-luokan rakenne.

Mittarit

Mittareina käytetyissä DS18B20-antureissa oli vesitiivis kotelo [kuva 14] ja 15 metrin johdot. Vesitiivisyys tarjoaa suojan pölyä ja kosteutta vastaan, jolloin ei ajan saatossa aiheudu oikosulkuja tai vahinkoa anturille tai laitteelle. Kaikki viisi anturia kytkettiin SN74HC4066-piiriin [kuva 9], joka tarjoaa pientä suodatusta signaalille eli pitää anturista saatavan tiedon lukukelpoisena, sillä 15 m pitkät johdot keräävät ympäristöstä huomattavasti häiriötä. Anturit kytkettiin tähtitopologiaan, joka on hankalin saada toimimaan luotettavasti häiriöiden takia. Suosituksena on käyttää kytkintä anturien välissä, ja luettaessa anturia pidetään yhteys vain yhteen anturiin.



Kuva 14. Anturi kotelossaan.

Anturien datalinja kytkettiin Raspberry Pin 7-nastaan, johon oletuksena ladataan käynnistyksen yhteydessä One-Wire-tuki. Nasta 7 valittiin sen takia, että siihen on kytketty myös kello, joka pitää kommunikaatiosyklin tahdissa ja näin mahdollistaa usean laitteen kanssa kommunikoinnin samaa johtoa pitkin. [8.]

Raspberry Pin oma One-Wire-tuki määrittää One-Wire-laitteiden nimet 28-alkuisiksi, ja tätä hyödynnettiin ohjelmassa. Nimen loppuosa muodostetaan laitteeseen määritetyn sarjanumeron avulla, joten jokainen laite saa eri nimen. Ohjelmaan on kovakoodattu anturien nimet ja tieto siitä, mitä anturi mittaa.

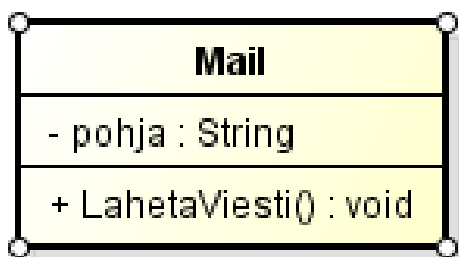
Ohjelman käynnistyksessä antureiden lämpötilat haetaan ensimmäisen kerran, minkä jälkeen lämmöt haetaan käyttäjän määrittämän ajan välein. Kuvassa 15 on Mittarit-olion luokka, josta näkee mittarin eri toiminnot. Lämpötila voidaan hakea desimaalilukuna tai tekstinä riippuen käyttötarkoituksesta. Kun mittareita luodaan, ne myös numeroidaan ja niille määritetään käytön helpottamiseksi kuvaavat nimet, jotka usein ovat mitattavan kohteen nimiä.

Mittari
- nimi : String - oikeaNimi : String
+ getLampomenuStatus() : String + getLampoMenuStatus() : double + getNimi() : String ~ Mittari() : void

Kuva 15. Mittari-luokan rakenne.

Sähköposti

Insinööriyössä sähköpostina käytettiin Gmail-tiliä, johon Raspberry Pi ottaa yhteyden libcurl-kirjastoa hyödyntäen. Otsikko on vakio, ja siinä ilmaistaan selkeästi, että viesti on laitteelta. Viesti saadaan parametrina, kun jokin hälytyksen ehto täyttyy, ja se liitetään viestiin ja lähetetään. Kuvasta 16 näkyy Sähköposti-olion rakenne.



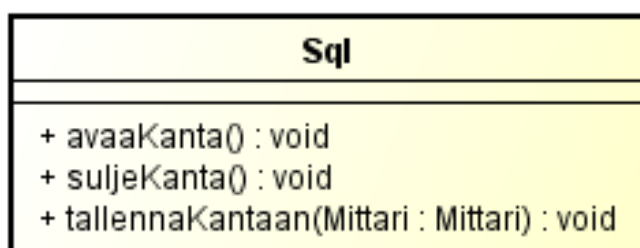
Kuva 16. Mail-luokan rakenne.

Curl on avoin komentokehotetiedostonsiirtosovellus, joka tukee monia protokollia. Koska Curl on avointa lähdekoodia, siitä on saatavilla myös kirjastot, jotka mahdollistavat Curlin implementoinnin ohjelmaan. Curlin kirjastoa kutsutaan libcurliksi, ja se on saatavilla .h-tiedostona eli C/C++-kirjastona.

Tietokanta

Insinööriyössä tietokantana käytettiin SQLite3:a, joka tarjoaa helpon hallittavuuden sovelluksessa ja siihen on helppo liittää WEB-pohjainen liittymä. Kanta avataan ohjelman alussa ja suljetaan, kun ohjelma sammutetaan. Ohjelman aikana on mahdollista lähettää kannalle mittarin sijainti muistissa, jolloin kantaan tallennetaan senhetkinen lämpötila kutsumalla Mittarin funktiota tallennusvaiheessa.

Tietokannan käyttö tehtiin helpoksi, eli sille lähetetään ainoastaan tieto, minkä mittarin lämpötila halutaan tallentaa ja SQL-olio hoitaa käsittelyn loppuun. Kuvassa 17 on esitelty SQL-olio.



Kuva 17. SQL-luokan rakenne.

SQL-komennot rakennetaan tallennusvaiheessa. Komento rakennetaan string-muuttujaan, joka sen jälkeen suoritetaan. Onnistuneesta tai epäonnistuneesta

Piirilevyn komponentit on selkeyden ja muokkaamisen helpottamiseksi sijoitettu osaluokittain. Piirustuksien tekeminen aloitettiin lisäämällä Raspberry Pin nastat, joihin liitettiin halutut verkot. Tämän jälkeen oli helppo lähteä rakentamaan ympärille muita osia. Piirilevylle otetaan virta itse Raspberry Pista.

Piirilevyn oikeassa yläkulmassa ovat näyttöihin menevät nastat. Yhteiset nastat ovat tiedonsiirtolinjat ja rekisterin valintanasta. Rekisterinasta kertoo näyttöjen HD44780-piirille, eli ohjaimelle, mihin halutaan kirjoittaa. Enable-nastalla ohjataan sitä, kumpi näytöistä on aktiivinen eli kumpi kirjoittaa. Jos enable-nasta viedään maihin, näyttö ottaa vastaan tietoa, mutta se ei kirjoita sitä näytölle. Enable-nastan ohjaukseen riittää 3,3 V:n jännite. Virran näytöt saavat Raspberry Pista, joka antaa tarvittavan 5 V:n jännitteen. Kumpaankin näyttöön on kytketty kaksi 10 k Ω :n potentiometriä. Toinen potentiometri ohjaa näytön taustavalaistusta ja toinen LCD-kiteiden intensiteettiä eli sitä, kuinka vahvalla virralla kiteitä ohjataan, mikä vaikuttaa siihen, miten teksti näkyy. Virta on liian korkea, kun kaikki ruudut näkyvät, ja liian matala, kun tekstiä ei näy lainkaan.

Ledejä, jotka ovat kuvassa 13 näyttönastojen oikealla puolella, ohjataan Raspberry Pin GPIO-nastoilla. GPIO-nastat on yhdistetty LED-valoihin ja vastuksen yli maahan. P4 – P6 ovat kiinnitysreikiä, joilla piirilevy voidaan kiinnittää tukevasti koteloon. Painikkeet, jotka on nimetty toiminnan perusteella, on yhdistetty 3,3 V:n jännitteeseen ja siitä Raspberry Pin GPIO-nastoihin. Nappia painaessa Raspberry Pin nastan jännite nousee 0 V:sta noin 3,3 V:iin, jolloin se rekisteröidään. Vivuilla Poisto ja Halytila voidaan ohjata ohjelman toimintaa. Vivut toimivat samalla tavoin kuin painikkeet sillä poikkeuksella, että vivut eivät palaudu oletustilaan vaan pitävät asetuksen.

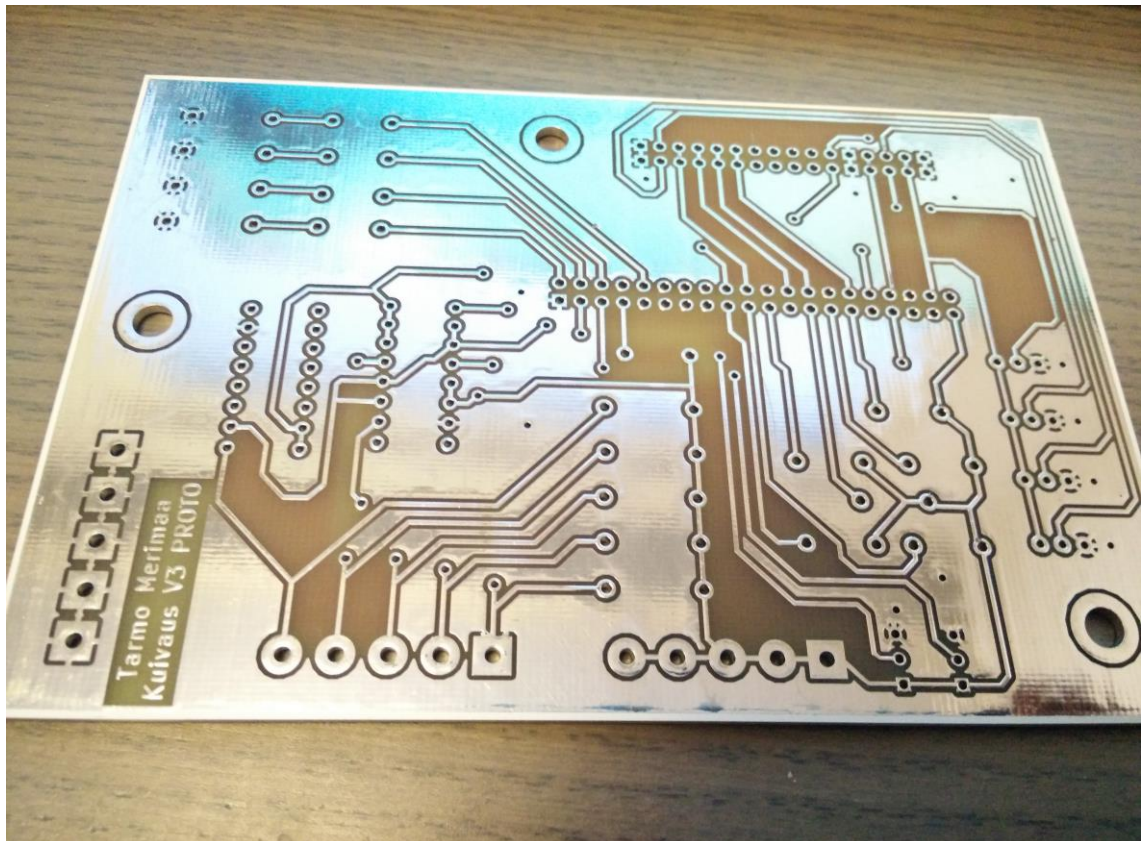
Piirilevyssä vasemmalla alhaalla ovat antureille menevät kytkennät. Jännite ja maa on kytketty suoraan Raspberry Pihin. Tiedonsiirtolinjat vaativat ylösvetovastuksen. Sen tarkoituksena on varmistaa signaalin pysyminen ylhäällä, kun se ei lähetä tietoja. Ylösvetovastuksen resistanssiksi on valittu 4,7 k Ω , joka on määritetty datalehdessä. Tiedonsiirtolinjat on yhdistetty myös SN74HC4066-piiriin, joka toimii kaksisuuntaisena analogisena porttina.

Oikealla alhaalla ovat SN74HC4066-piirit, jotka toimintatapansa takia luovat signaalin uudestaan. Tämä ominaisuus mahdollistaa usean pitkän anturin kiinnittämisen, koska muutoin signaali keräisi häiriöitä eikä antureista saataisi minkäänlaista tietoa. Kuvan 8 mukaisesti nastaan 14 yhdistettiin 3,3 V:n käyttöjännite ja nastaan 7 maa. Piirin nastat, jotka on merkitty A- tai B-kirjaimella, ovat tiedonsiirtonastoja eli niistä kulkee tieto sisään

ja ulos. C-nastat puolestaan ovat enable-nastoja eli niillä ohjataan, kulkeeko tieto A- ja B-nastan välillä. Kuvassa 8 numerot kertovat myös sen, että mitkä nastat ovat liitettyä toisiinsa.

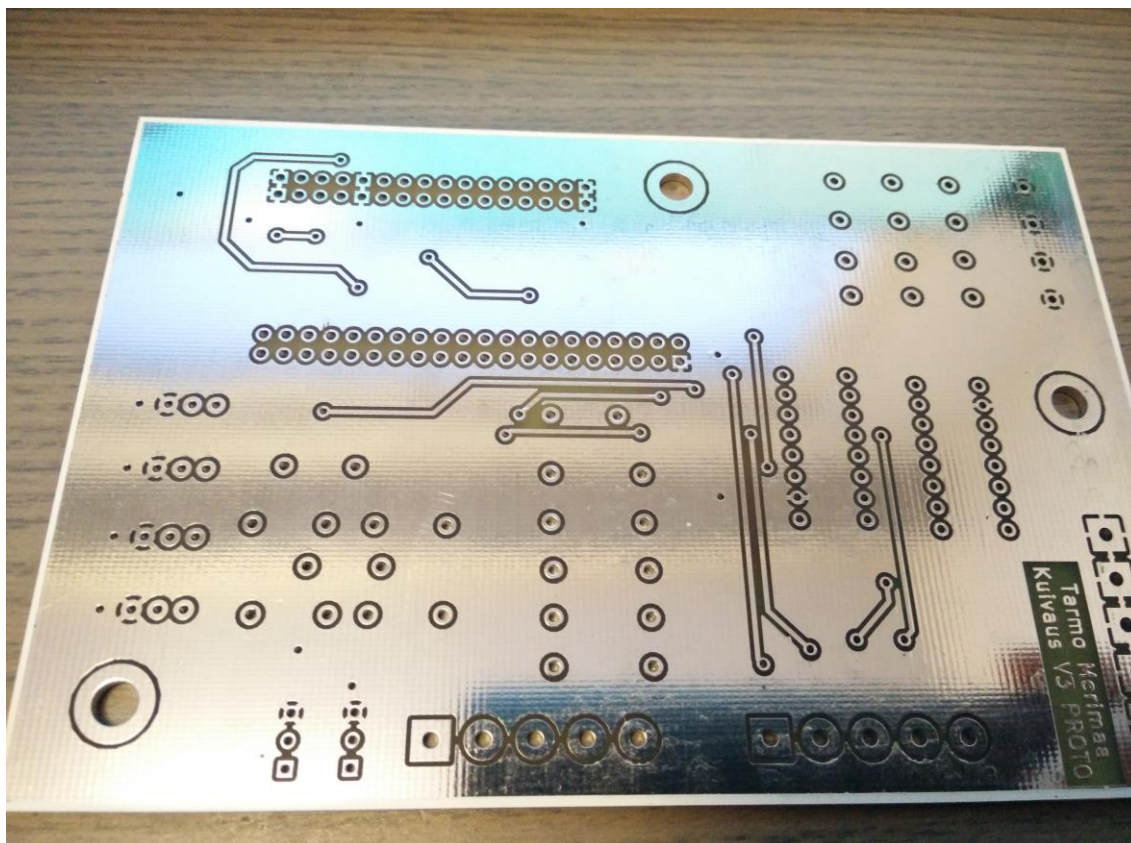
Piirilevyt tilattiin Seeds Studiosta, josta saa halvalla tilattua, kun toimittaa gerber-tiedostot. Piirilevy on mitoiltaan 83 mm x 123 mm x 1,6 mm. Piirilevy on myös kaksipuolinen, ja siinä on kolme ruuvipaikkaa, joista piirilevyn voi kiinnittää koteloon. Piirilevyn materiaali on FR-4 TG130. Päälystysmenetelmäksi valittiin HASL-tekniikkaa.

Kuvassa 19 on piirilevyn takapuoli, jossa isoin osa linjoista kulkee. Linjat menevät etupuolelle vain, jos joutuvat ohittamaan alapuolella olevia linjoja. Maa-alueet eli isot alueet ovat kytkettynä toisiinsa, ja alueet toimivat samalla maana.



Kuva 19. Insinööriyön piirilevyn takapuoli.

Piirilevyn etupuolelle [kuva 20] on jätetty enimmäkseen maata ja mahdollisuus kolvata komponentit "väärältä" puolelta.



Kuva 20. Insinööriyön piirilevyn etupuoli.

4.3 Kotelo

Insinööriyön kotelo on valmistettu ABS-muovista, ja kotelon etu- ja takapuolen välissä on tiiviste, jottei pöly pääse kotelon sisälle ja aiheuta oikosulkua tai heikennä jäähdytystä. Kotelon alkuperäinen luokitus oli IP67, mutta muokkauksien jälkeen luokitusta ei voida todentaa. Insinööriyön pitäisi täyttää tilan takia vähintään IP6-luokituksen, sillä tilassa esiintyy erittäin hienoajakoista pölyä.

IP-luokitus perustuu standardiin SFS-EN 60529, joka määrittelee menetelmät, miten luokitus testit tehdään. Luokituksessa ensimmäinen numero kertoo pölyn ja vieraiden esineiden pääsyn koteloon. Luokitus määritellään väliltä 0 – 6, jossa 0 on suojaamaton ja 6 on täysin tiivis. Toinen numero kertoo suojauksen vettä vastaan. Luokitus on väliltä 0 – 8, jossa 0 on suojaamaton ja 8 kestää jatkuvaa veden alla olemista. Tässä luokituksessa 1 – 2 on vedenkestävyys tietyistä suunnista ja 3 on sateenkestävyys. [13.]

Kuvissa 7 ja 21 on kotelon etupaneeli, jossa on kaksi näyttöä. Ylempi näyttö kertoo yleistietoa ohjelmasta ja senhetkiset lämpötilat ja ohjelmatilan. Ylemmän näytön sivuja voi selata näytön alapuolella olevalla napilla.

Alempi näyttö toimii päänäyttönä, ja sitä hallinnoidaan kuudella napilla, jotka ovat oikealla. Nuolipainikkeilla selataan asetuksia läpi, ja punainen nappi toimii peruuta- tai taaksepäin-painikkeena ja vihreä hyväksy-painikkeena.

Ylemmän näytön oikealla puolella on neljä eriväristä lediä, joista voi päätellä ohjelman tilan nopeasti. Eri tiloilla on omat värikoodit.



Kuva 21. Kotelon etupaneeli.

Kotelon johdotus tehtiin mahdollisimman lyhyillä johdoilla. Näytöt kytkettiin piirilevyyn Dupont-kaapeleilla, jotka teipattiin lattakaapeliksi. Ledit ja napit yhdistettiin piirilevyyn pikaliittimien kautta, jotta tarvittaessa vaihto olisi helpompaa. 3,5 mm:n liittimistä tulevat johdot niputettiin ja ne tulevat laitaa pitkin ruuviliittimiin.

4.4 Ratkaisut

Insinööriyön seuranta- ja valvontalaitteessa voitaisiin myös käyttää langattomia antureita, jolloin välttyttäisiin johtojen vedosta. Yksi mahdollinen laite olisi Particle Photon, jonka saisi liitettyä langattomaan verkkoon ja siihen liitettyä lämpösensorin. Tästä

ratkaisusta luovuttiin jo suunnitteluvaiheessa, sillä se, että paristoja jouduttaisiin vaihtamaan, ei sopinut asiakkaalle.

Lämpöantureissa päädyttiin käyttämään DS18B20-anturia 1-wire-tekniikan takia, sillä haluttiin mahdollisuus lisätä antureita tulevaisuudessa. DS18B20-anturi on myös hintansa takia hyvä valinta ja sitä sai valmiissa kotelossa. Anturit ovat tähtitopologiassa, koska mitattavat kohteet ovat hajautettuna tavalla, joka ei anna mahdollisuutta erilaisiin topologioihin pidentämättä johtoa monikertaiseksi.

Raspberry Pille olisi ollut useampiakin ratkaisuja, muun muassa kloonituotteet, esimerkiksi Orange Pi ja Banana Pi, mutta Raspberry valittiin luotettavuuden ja hyvien ominaisuuksien takia. Raspberry Pi oli myös valmiina insinööriyön tekijän hyllyssä.

Ensimmäisissä versioissa napit olivat juotettuna suoraan piirilevyyn, jolloin säästettiin tilaa, mutta heikkoudeksi muodostui nappien vaihto tarvittaessa. Ensimmäisellä testauskerralla huomattiin, että napit kärsivät vahinkoa ja oli tarve vaihtaa niistä kaksi. Liitokset vaihdettiin pikaliittimiksi, jolloin vaihto onnistuu huomattavasti helpommin. Samalla päädyttiin myös siihen, että anturien johdot kotelon sisällä liitetään pikaliittimillä. Tämä mahdollistaa kuluviin osien nopean vaihdon ilman juotinta, mikä pienentää samalla piirilevyn vahingoittamisen riskiä.

Selvityksen alla on myös mahdollisuus siirtää Raspberry Pin internetyhteys Tosibox-laitteen taakse, jolloin saataisiin tietoturvaso huomattavasti korkeammalle. Tosibox-laitte tarjoaa palomuurin ja VPN-palvelun, joka rajoittaisi pääsyä Raspberry Pihin, mutta samalla estäisi asiattomien pääsyn laitteeseen. Tällä hetkellä Raspberry Pi on kytketty kytkimeen, joka ei ole yhteydessä internetiin.

4.5 Käyttöönotto ja jatko

Seuranta- ja valvontalaite asennettiin viljakuivurin sähkökeskuksen viereen, josta saadaan virta. Anturit vedettiin katon rajaa pitkin mitattaviin kohteisiin.

Laitteen kehitystä jatketaan tulevaisuudessa. Tarkoitus on lisätä poistoilmaluukun ohjaus, jolloin viljan jäähdytykseen siirryttäisiin automaattisesti. Selvityksen alla on myös itse kuivurin ohjaus, jolloin olisi mahdollista automatisoida tuulettimien hallinta ja kuivausprosessi tapahtuisi kokonaisuudessa nappia painamalla. Tarkoituksena on myös

lisätä luotettavuutta kytkemällä anturit releen avustuksella tai muulla yhteyden katkaisevalla menetelmällä, sillä nykyisessä ratkaisussa esiintyy ongelmia häiriöiden takia.

Koekäytössä tuli esille, että kotelo ei ole tarpeeksi tiivis, koska pölyä pääsi koteloon sisälle. Todennäköisimmät paikat vuodoille ovat 3,5 mm:n liitännät ja reiät joihin napit oli kiinnitetty. Näytöt oli kiinnitetty koteloon silikonilla, joka toimi eristävänä materiaalina myös.

Nappien nupit kärsivät myös vahinkoa. Pöly, joka meni nappien mekaanisiin osiin, aiheutti jäykkyyttä nappeihin ja osa nappien nupuista oli kadonnut. Nappien nappien katoamisesta epäillään lintuja tai jyrsijöitä. Ratkaisu tähän ongelmaan olisi hipaisunapit, mutta työhanskat estäisivät työn käyttämisen tällöin. Kumi- tai silikonikupu nappien päälle ehkäisisi pölyn menemistä nappien mekaanisiin osiin, mutta se ei suojaa eläimiltä. Seuraavaan versioon kokeillaan nappeja, jotka ovat osittain kotelon sisällä, jolloin nuppia ei saa irti, ja silikonin- tai kumikupu päällä, joka suojaa pölyltä.

Työssä käytetyt anturit ovat kiinteät ja pölytiivit, ja ainoa riski on, että jos pölyiseen aikaan anturi irrotetaan, itse liittimeen sisään pääsee pölyä, joka voi aiheuttaa oikosulun tai kontaktihäiriöitä. Insinööriyössä käytetään jatkossa samoja 3,5 mm:n liittimiä ja tutkitaan mahdollisuutta langattomiin antureihin, esim. Bluetooth LE, jolla olisi mahdollista saada pitkä akunkesto.

Raspberry Pin datan analysointiin käytetään Grafana-ohjelmaa, sillä koekäytössä käyttäjä piti Grafanaa kaikkein mielekkäimpänä käyttöliittymänä. Grafana on avointa lähdekoodia ja suunniteltu aikapohjaisen tiedon analysointiin. Se hakee tiedot tietokannasta SQL-kyselyllä. Kyselyn tekeminen on Grafanassa helppoa, mikä antaa käyttäjälle mahdollisuuden rakentaa omia tilastoja. Kyselyn rakentaminen on mahdollista tehdä graafisesti. Grafana osaa hakea tallennetut tietotyypit ja laitteiden nimet valmiiksi, eli käyttäjän ei tarvitse kuin valita tieto ja laitteen nimi. Järjestely laitteen nimen ja ajan mukaan on oletuksena, eli niitä ei tarvitse muuttaa. [14.]

Grafanan koekäytössä huomattiin hitautta, varsinkin jos on useita tilastoja sivulla ja haluaa hakea pidemmältä aikajaksolta tietoja. Hitaus johtuu Raspberry Pin prosessorin rajallisesta tehosta. Kantaan tallennetaan arvoja muutaman sekunnin välein, mutta Grafana on asetettu kysymään arvot minuutin välein. Tätä arvoa voidaan muuttaa myöhemmin, jotta kyselyt olisivat nopeampia. Toinen mahdollinen pullonkaula voi

muodostua muistista, sillä kyselyn aikana muistin käyttö kasvaa huomattavasti, eli erittäin pitkän aikajakson tarkkailussa muisti voi loppua kesken, sillä kyselyn aikana tulokset tallennetaan muistiin.

5 Yhteenveto

Insinööriyön tarkoituksena oli luoda asiakkaalle viljankuivausprosessiin lämpötilanseurantajärjestelmä, joka toimittaa samalla tietoa asiakkaalle. Työn oli tarkoitus olla myös laajennettavissa tulevaisuudessa, jotta kuivausprosessin automatisoimista voidaan viedä pidemmälle.

Tuote, joka luovutettiin asiakkaalle, seuraa viljan lämpötilaa ja kuivaustilaan siirryttäessä aloittaa lämpötilan seurannan. Asiakas voi itse säätää tallennusvälin tiheyttä, jolloin saadaan haluttu määrä tietoa prosessin etenemisestä. Kuivausprosessin aikana seurataan radiaattoriveden, sisääntuloilman, viljan, ulkoilman ja poistoilman lämpötilaa. Vilja on tarpeeksi kuivaa varastoon, kun poistoilman lämpötila saavuttaa tavoitteen, joka riippuu viljalajikkeesta. Kuivauksen aikana seurataan myös sen veden lämpötilaa, jolla lämmitetään kuivausilma. Jos veden lämpötila laskee liikaa, lähetetään sähköpostiviesti asiakkaalle, jolloin asiakas on tietoinen, että hakepoltin ei pysty lämmittämään tarpeeksi vettä ja tarvitaan lämmintä vettä vesiverkostosta.

Kun lähestytään tavoitelämpötilaa, laite lähettää sähköpostiviestin asiakkaalle, jolloin asiakas voi mennä kuivurille ja sammuttaa kuivauksen ja avata kattoluukun, jolloin vilja jäähtyy. Tuote siirtyy tässä vaiheessa myös jäähdystilaan, sillä kuivausilman lämpötila alkaa laskea ja viljan sekaan puhalletaan viileämpää ilmaa. Tässä vaiheessa seurataan poistoilman lämpötilaa. Asiakkaalle lähetetään sähköpostiviesti, kun vilja on jäähtynyt tarpeeksi tai kun viljan lämpötila ei enää laske. Se, että vilja ei jäähdy tarpeeksi, johtuu yleensä kuumasta päivästä, jolloin viljan annetaan olla kuivurissa yöhön asti, jolloin se siirretään varastoon. Kun prosessi on valmis, laite siirtyy lepotilaan odottamaan uutta viljaerää.

Testeissä tuli vastaan erinäisiä ongelmia, muun muassa viljan pöly, joka pääsee kaikkialle. Seuraavaksi olisi tavoitteena tutkia erilaisia ratkaisuja, miten laite saadaan kauemmaksi kuivurista siten, että lämpötiloja voidaan yhä seurata. Tavoitteena on myös lisätä poistoilmaluukun moottorin ohjaus sekä saada laite eri sähköverkkoon kuin kuivuri, koska kuivurin aiheuttamat häiriöt saattavat välillä aiheuttaa insinööriyön uudelleenkäynnistymisen. Lopulliseen laitteeseen on myös tarkoituksena kehittää joko ohjelma tai selainpohjainen hallinta, joka mahdollistaisi myös itse laitteen paketoimisen koteloon, johon ei pääsisi viljapölyä.

Lähteet

- 1 Simatic. Verkkodokumentti. Siemens.
<http://w3.siemens.com/mcms/topics/en/simatic/pages/default.aspx>
Luettu 23.12.2017.
- 2 MELSEC. Verkkodokumentti. Mitsubishi Electric
<http://www.mitsubishielectric.com/fa/products/cnt/plc/index.html>
Luettu 23.12. 2017.
- 3 Tosibox. <https://www.tosibox.com/> Luettu 4.3.2018
- 4 Photon-datasheet: Verkkodokumentti. Particle-photon
<https://docs.particle.io/datasheets/photon-datasheet/>. Luettu 16.10. 2016.
- 5 RPi Distributions. 2016. Verkkodokumentti. Elinux. <
http://elinux.org/RPi_Distributions>. Luettu 10.5.2016.
- 6 Omien laitteitteesi Internet. Verkkodokumentti. Microsoft. <
<https://developer.microsoft.com/fi-fi/windows/iot>>. Luettu 10.5.2016
- 7 Raspberry Pi 2 model B: Verkkodokumentti. Raspberry.
<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. Luettu 16.10. 2016.
- 8 Hale, Ted B. New Raspberry Pi GPIO Pinout Diagram. 2015.
Verkkodokumentti. <
<http://rasberrypihobbyist.blogspot.fi/2015/03/new-raspberry-pi-gpio-pinout-diagram.html>>. Luettu 10.5.2016
- 9 Unattended Upgrades. Verkkodokumentti. Debian wiki.
<https://wiki.debian.org/UnattendedUpgrades> Luettu 4.3.2018
- 10 Guidelines for reliable long line 1-wire networks. Verkkodokumentti.
Maxim Integrated <https://www.maximintegrated.com/en/app-notes/index.mvp/id/148>. Luettu 16.10. 2016.
- 11 Visual Studio. Verkkodokumentti. Visual Studio.
<https://www.visualstudio.com>. Luettu 12.11. 2016.
- 12 About. Verkkodokumentti. SQLite. <https://sqlite.org/about.html>.
Luettu 12.11. 2016.
- 13 IP-luokitus. Verkkodokumentti. Virtuaali ammattikorkeakoulu.
<http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/030503/1133959973706/1133960605288/1133961558641/1133961579677.html>
Luettu 23.12. 2017