



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Jere Laaksonen

AUTOMATISOITU KASVIEN KASTELUJÄRJESTELMÄ

Tekniikka
2022

TIIVISTELMÄ

| | |
|--------------------|--|
| Tekijä | Jere Laaksonen |
| Opinnäytetyön nimi | Automatisoitu kasvien kastelujärjestelmä |
| Vuosi | 2022 |
| Kieli | suomi |
| Sivumäärä | 59 + 6 liitettä |
| Ohjaaja | Jukka Matila |

Opinnäytetyön tavoitteena oli tehdä automatisoitu kastelujärjestelmä huonekasveille, mikä osaisi huolehtia itsenäisesti huonekasvien kastelusta. Kastelujärjestelmä on suunniteltu ESP8266-mikro-ohjaimelle. ESP on yhteydessä Raspberry Pi:hin, joka toimii MQTT-välittäjänä. Antureiden keräämä data voidaan lähettää MQTT:lle. Kun data on saapunut välittäjälle, käytetään tietosiltaa apuna, jotta data saadaan siirtymään oikeassa muodossa tietokantaan. Grafana on yhdistetty tietokantaan. Grafanan avulla voidaan lukea antureiden dataa graafisesti. Lopuksi Grafana on linkitetty itse tehdylle WordPress-sivuille. WordPressin sivujen tarkoituksena on seurata antureiden lähettämää dataa eri vuorokauden aikana.

ABSTRACT

| | |
|--------------------|--|
| Author | Jere Laaksonen |
| Title | Automated irrigation system for house plants |
| Year | 2022 |
| Language | Finnish |
| Pages | 59 + 6 Appendices |
| Name of Supervisor | Jukka Matila |

The purpose of this thesis was to make an automated irrigation system for house plants that could take care of the irrigation of houseplants automatically. The irrigation system is designed for the ESP8266 microcontroller. ESP is connected to Raspberry Pi, which acts as an MQTT broker. The data collected by the sensors can be sent to the MQTT. Once the broker has received the data, the data bridge is used as an aid to get the data to the database in the correct format. Grafana is connected to the database. Grafana can be used to read sensor data graphically. Finally, Grafana is linked to self-made WordPress pages. The purpose of WordPress pages is to track the data sent by the sensors at different times of the day.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVA- JA TAULUKKOLUETTELO

LIITELUETTELO

TERMIT JA LYHENTEET

| | | |
|---|---|----|
| 1 | JOHDANTO..... | 10 |
| 2 | KASTELUJÄRJESTELMÄN KÄYTTÖTARKOITUS..... | 11 |
| 3 | KASTELUJÄRJESTELMÄN SUUNNITTELU | 12 |
| | 3.1 Järjestelmäkaavio..... | 13 |
| | 3.2 Lohkokaavio | 14 |
| | 3.3 Laitevaatimukset..... | 15 |
| | 3.4 Verkkokäyttöliittymän suunnittelu..... | 16 |
| 4 | KASTELUJÄRJESTELMÄN KOMPONENTIT JA TESTAUS | 18 |
| | 4.1 NodeMCU ESP8266..... | 18 |
| | 4.2 Raspberry Pi | 18 |
| | 4.3 Kapasitiivinen maakosteusanturi ja vesipumppu | 19 |
| | 4.4 DS18B20-lämpötila-anturi | 21 |
| | 4.5 HC-SR04-etäisyysanturi..... | 22 |
| | 4.5.1 Vedenpinnan mittaaminen | 24 |
| | 4.5.2 Vesisäiliö..... | 27 |
| | 4.5.3 Ohjelmointi..... | 27 |
| | 4.6 DHT11-lämpötila- ja kosteusanturi..... | 29 |
| 5 | MQTT-PALVELIN RASPBERRY PI:LLE | 30 |
| | 5.1 Mosquitto MQTT-välittäjä | 30 |
| | 5.1.1 Asennus | 31 |
| | 5.2 ESP8266 MQTT-asiakas..... | 32 |
| | 5.2.1 ESP8266 yhdistäminen MQTT-välittäjään..... | 33 |

| | | |
|-------|-------------------------------------|----|
| 5.2.2 | ESP8266-aihe..... | 35 |
| 5.3 | Node-RED..... | 36 |
| 5.4 | InfluxDB..... | 38 |
| 5.5 | Grafana..... | 39 |
| 5.6 | WordPress..... | 40 |
| 6 | KASTELUJÄRJESTELMÄN VALMISTUS | 42 |
| 6.1 | Kytkentäkaavio..... | 42 |
| 6.2 | Kytkentälevyn valmistus | 43 |
| 6.3 | Varavirtalähde..... | 44 |
| 6.4 | Järjestelmän kotelointi | 45 |
| 7 | TESTAUS..... | 47 |
| 8 | JATKOKEHITYS..... | 51 |
| 8.1 | Huonekasvit | 51 |
| 8.2 | Kasvihuonekasvit | 51 |
| 9 | JOHTOPÄÄTÖKSET | 53 |
| | LÄHTEET | 54 |
| | LIITTEET | 57 |

KUVA- JA TAULUKKOLUETTELO

| | |
|--|----|
| Kuva 1. Järjestelmäkaavio..... | 13 |
| Kuva 2. Lohkokaavio. | 14 |
| Kuva 3. Verkkosivujen suunnitelma..... | 17 |
| Kuva 4. ESP8266-mikro-ohjain..... | 18 |
| Kuva 5. Raspberry Pi. | 19 |
| Kuva 6. Kapasitiivinen maakosteusanturi..... | 19 |
| Kuva 7. Kapasitiivisen anturin ja vesipumpun kytkentä. | 20 |
| Kuva 8. DS18B20-anturi. | 21 |
| Kuva 9. DS18B20-anturin kytkentä..... | 21 |
| Kuva 10. HC-SR04-ultraäänianturi. | 22 |
| Kuva 11. HC-SR04-anturin aikakaavio. (ElectronicWings, 2021). | 23 |
| Kuva 12. HC-SR04-anturin toimintaperiaate. (Random nerd tutorials, 2021). | 23 |
| Kuva 13. HC-SR04-anturin kytkentäkaavio. | 24 |
| Kuva 14. HC-SR04-etäisyyden mittaaminen vedenpinnasta. (AutomationWiki, 2018). | 25 |
| Kuva 15. Plastex:in vesiastia. | 27 |
| Kuva 16. HC-SR04-vedenkorkeuden mittaaminen. | 28 |
| Kuva 17. DHT11-anturi..... | 29 |
| Kuva 18. DHT11-anturin kytkentä. | 29 |
| Kuva 19. Datán siirtyminen..... | 30 |
| Kuva 20. Raspberry Pi:n IP-osoite..... | 31 |
| Kuva 21. Mosquiton asennuskomento..... | 31 |
| Kuva 22. Käyttäjätunnuksen ja salasanan luominen. | 32 |
| Kuva 23. Salasanán salaaminen..... | 32 |
| Kuva 24. MQTT-viestintä. | 33 |
| Kuva 25. Kirjastojen, Wi-Fi:n ja MQTT:n määrittäminen..... | 34 |
| Kuva 26. ESP8266 yhdistäminen Wi-Fi-verkkoon..... | 34 |
| Kuva 27. MQTT-aihe. | 35 |
| Kuva 28. MQTT-aiheen tulostaminen..... | 35 |

| | |
|---|----|
| Kuva 29. Lämpötilan mittaaminen..... | 36 |
| Kuva 30. MQTT-välittäjälle saapunut lämpötila. | 36 |
| Kuva 31. Node-RED anturin data tietokantaan. | 37 |
| Kuva 32. Node-RED kaikki anturit. | 37 |
| Kuva 33. InfluxDB-tietokanta. | 38 |
| Kuva 34. Antureiden data Grafanassa. | 39 |
| Kuva 35. Verkkosivujen aloitussivu..... | 40 |
| Kuva 36. Verkkosivujen kasvisivu 1/2. | 41 |
| Kuva 37. Verkkosivujen kasvisivu 2/2. | 41 |
| Kuva 38. KytKentä. | 42 |
| Kuva 39. KytKentälevylle tehty juotos. | 43 |
| Kuva 40. ESP8266 liitettynä kytKentälevylle..... | 43 |
| Kuva 41. Järjestelmä kotelossa. | 46 |
| Kuva 42. Kastelujärjestelmä toiminnassa. | 46 |
| Kuva 43. Veden määrän mittaaminen. | 48 |
| Kuva 44. Veden määrän tulokset MQTT-välittäjässä. | 48 |
| Kuva 45. Maakosteuden mittaaminen. | 49 |
| Kuva 46. Maan kosteuden tulokset COM-portissa. | 50 |
| Kuva 47. Maan kosteuden tulokset MQTT-välittäjässä. | 50 |
| | |
| Taulukko 1. Mittaustulokset..... | 26 |

LIITELUETTELO

LIITE 1. 3D-suunniteltu DHT11-lämpötila-anturin kansi.

LIITE 2. 3D-suunniteltu DHT11-lämpötila-anturin kotelo.

LIITE 3. 3D-suunniteltu HC-SR04-etäisyysanturin kansi.

LIITE 4. 3D-suunniteltu HC-SR04-etäisyysanturin kotelo.

LIITE 5. 3D-suunniteltu maakosteusanturin kansi.

LIITE 6. 3D-suunniteltu maakosteusanturin kotelo.

TERMIT JA LYHENTEET

| | | |
|-------------|-------------------------------------|--------------------------------------|
| MQTT | Message Queuing Telemetry Transport | viestintä- ja tiedonvaihtoprotokolla |
| IoT | Internet of Things | esineiden internet |
| NTC | Negative Temperature Coefficient | negatiivisen lämpötilan kerroin |
| API | Application Programming Interface | ohjelmointirajapinta |
| ARM | Advanced RISC Machines | mikroprosessoriarkkitehtuuri |

1 JOHDANTO

Opinnäytetyön tarkoituksena oli rakentaa automatisoitu huonekasvien kastelujärjestelmä, joka osaisi kastella kasvit ilman ihmisen apua sekä kerätä erilaisten antureiden avulla dataa kasveista, mikä lähetetään MQTT-välittäjään. MQTT-välittäjä käyttää hyödyksi tietosiltaa, joka siirtää antureiden keräämän datan tietokantaan. Tietokanta on yhdistetty visualisointiohjelmaan, jonka avulla voidaan esittää antureiden keräämä data graafisesti. Kun data on saatu graafiseen muotoon, on se linkitettävä WordPress-sivulle. Käyttäjä voi seurata WordPress-sivujen avulla antureiden keräämää dataa eri vuorokauden aikana ja lisäksi käyttäjä voi lisätä järjestelmään omia kasveja.

Opinnäytetyö koostuu tehdystä automaatiojärjestelmästä, jossa on asiakkaana toimiva ESP8266-mikro-ohjain. ESP8266 lähettää antureiden keräämän datan Raspberry Pi:lle, joka toimii MQTT-välittäjänä. MQTT ohjaa antureiden datan tietosiltaa käyttäen tietokantaan, jonka jälkeen ne ohjataan visualisointiohjelmaan. Visualisointiohjelma muuttaa antureiden tallennetun tiedon graafiseen muotoon. Lopuksi visualisointiohjelma on linkitetty WordPress-sivuille, jonne se tulee päivittymään reaaliajassa.

2 KASTELUJÄRJESTELMÄN KÄYTTÖTARKOITUS

Tässä luvussa pohditaan mihin kastelujärjestelmää oikein tarvitaan ja millaisia on aikaisemmin valmistettu.

Käyttämällä automaattista kastelujärjestelmää huonekasveille autetaan kasveja voimaan paremmin sekä pysymään terveenä. Ihmisen heikkous kasvien kastelussa on ollut se, että joko kasveja kastellaan liian useasti tai liian harvoin. Kastelujärjestelmän avulla käyttäjän ei tarvitse enää huolehtia kasvien veden saannista vaan järjestelmä osaa kastella automaattisesti kasvin aina kun kasvilla on tarve saada vettä. Parasta tässä on se, että säästetään kallisarvoista aikaa sekä vettä.

Automaattinen huonekasvien kastelujärjestelmä ei ole mikään uusi idea. Laitteita on tehty lukuisia erilaisia niin elektroniikkaa harrastavien toimesta kuin myös kaupalliseen myyntiin. Suuremmissa mittakaavassa samaa periaatetta hyödyntäen on valmistettu maanviljelijöille kastelulaitteita suurille kasvihuoneille.

3 KASTELUJÄRJESTELMÄN SUUNNITTELU

Melkein kaikki ovat varmasti unohtaneet joskus kastella huonekasveja ajallaan. Kasvien pitäminen elossa ja terveenä voi olla haastavaa, jos ei ole aikaa kastella niitä säännöllisin väliajoin. Kasvien kastelu tulee vielä vaikeammaksi, jos kasveja on monia eri lajeja, jotka vaativat erilaisia ilmasto-olosuhteita sekä kastelumääriä.

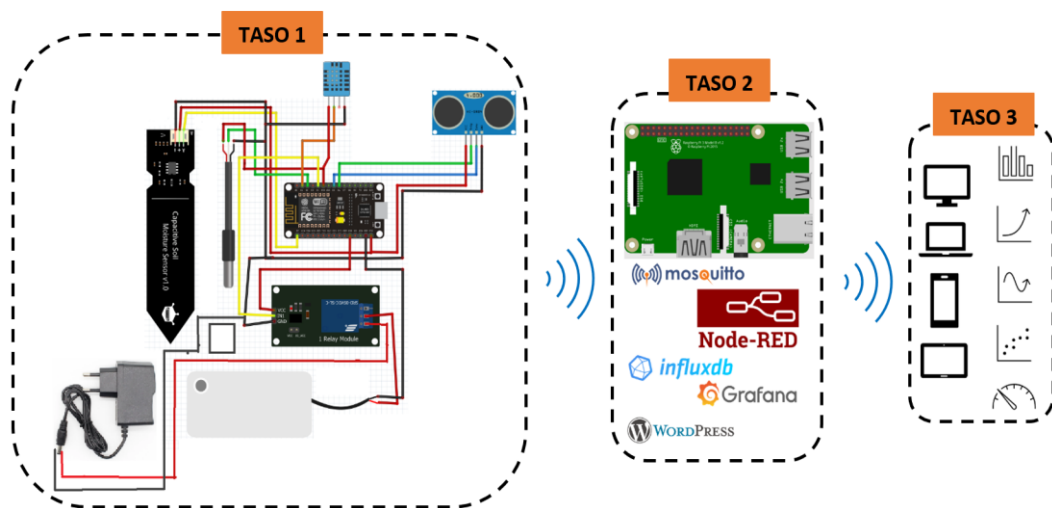
Kastelujärjestelmän pääsääntöinen idea on huolehtia automaattisesti kasvin kastelusta ja helpottaa arkea, jolloin jää aikaa muihin arjen askareisiin. (Instructables circuits, 2022 a).

Kasveja voidaan lisätä laitteelle enemmänkin. Jokaiselle kasville tulee antaa kaksi anturia, jotka ovat kapasitiivinen maakosteusanturi ja lämpötila-anturi. Lisäksi kasveille on laitettava oma vesipumppu sekä vesijohto, joka on yhteydessä vesiastiaan. Antureiden keräämä data tallennetaan tietokantaan, jotta voitaisiin seurata kasvien kasteluaikaväliä sekä lämpötilaa. Tietokantaan tallennetut datat on lähetetty Grafanaan, jotta anturien keräämä data saadaan graafiseen muotoon. Grafana on linkitetty omalle verkkosivulle, jota ihminen voi helposti ja ymmärrettävästi seurata. Verkkosivulle on lisättävä uusi kasvi ja haettava oikeat anturin data-aulut Grafanasta. Tämän jälkeen kasvien seuranta on helppoa koska voit seurata verkkosivuja millä tahansa laitteella, kunhan sillä pääsee kotisi sisäverkkoon esimerkiksi puhelimella, tabletilla ja tietokoneella.

ESP8622-mikro-ohjaimen tehtävä on hallita etäisyysanturia, vesipumppua, maaperän lämpötilaa, kapasitiivista maankosteutta ja huoneen lämpötilaa/kosteutta. Etäisyysanturi lukee vesiastian tilaa, jonka avulla saadaan selville kuinka paljon vettä on astiassa vielä jäljellä. ESP8266 on kytketty rele, joka käynnistää vesipumpun, jos kapasitiivinen maakosteusanturi havaitsee mullan olevan liian kuiva. (Instructables circuits, 2022 a).

3.1 Järjestelmäkaavio

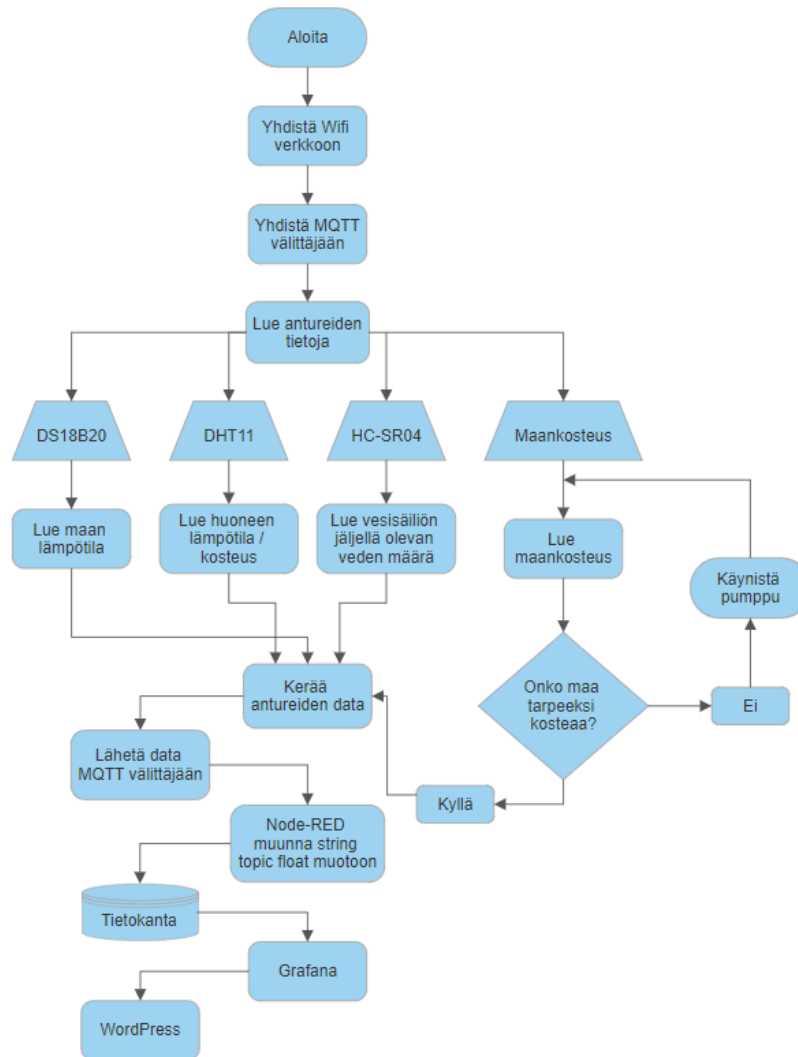
Tässä on kuvattuna kastelujärjestelmän yleiskuvaus järjestelmän toimivuudesta sekä niiden välisistä yhteyksistä. Kuvassa 1 järjestelmä on lajiteltu kolmeen eri tasoon. Tasossa 1 on kuvattuna järjestelmän elektronikka. Elektronikka koostuu ESP8266-mikro-ohjaimesta, johon on liitettyä anturit. Tasossa 2 on kuvattuna Raspberry Pi, joka toimii järjestelmän MQTT-välittäjänä. Lisäksi Raspberry Pi:hin on asennettuna tietosilta, tietokanta, visualisointiohjelma sekä sisällönhallintaohjelmisto. Tasossa 3 on kuvattuna millä eri laitteilla käyttäjällä on mahdollista seurata kasvin hyvinvointia.



Kuva 1. Järjestelmäkaavio.

3.2 Lohkokaavio

Kastelujärjestelmälle on oltava tehtynä toimivuuden kannalta asianmukainen lohkokaavio. Kuvassa 2 on kuvattuna lohkokaavio, joka kertoo laitteen yleisen käsityksen mitä missäkin vaiheessa tapahtuu.



Kuva 2. Lohkokaavio.

Ohjelma alkaa käynnistyksellä, joka tarkoittaa, että mikro-ohjain on käynnissä. Ohjelma suorittaa aluksi laitteen yhdistämisen Wi-Fi verkkoon, jonka jälkeen laite yhdistetään MQTT-välittäjään. Yhdistettyään laitteen onnistuneesti MQTT-välittäjään laite lukee antureiden keräämää dataa. Maakosteusanturi mittaa maan kosteutta. Anturin havaitessa maan olevan liian kuiva anturissa oleva 555-ajastin lähettää signaalin mikro-ohjaimelle, joka käynnistää vesipumpun. Anturi lukee maan kosteutta myös silloinkin, kun vesipumppu on käynnissä. Kun maan kosteus on riittävän kostea anturin 555-ajastin lähettää uuden signaalin mikro-ohjaimelle, joka sammuttaa vesipumpun. Anturin keräämät tiedot lähetetään MQTT-välittäjään, josta ne siirtyvät tietosiltaan, joka muuttaa saadun string-muotoisen topicin float-muotoon. Float-muodossa oleva data lähetetään tietokantaan. Tietokanta on yhdistetty Grafanaan, jotta antureiden data voitaisiin visualisoida. Lopuksi visualisoitu data on yhdistetty WordPress-sivulle.

3.3 Laitevaatimukset

Laitteisto vaatimuksilla kuvataan yleisesti mitä ominaisuuksia ja toimintoja järjestelmältä vaaditaan.



Ohjelmoitava alusta on oltava helppokäyttöinen, missä olisi sisäänrakennettu Wi-Fi sekä laitteen olisi oltava virraltaan pienikulutuksinen.












Raspberry Pi toimii MQTT-välittäjänä, joka saa virtansa virtalähteestä, joka on kytkettynä verkkovirtaan.



Raspberry Pi:hin on asennettuna tietokanta, johon antureiden data tallentuu.



Raspberry Pi:hin on asennettava visualisointiohjelma, joka osaa ottaa tietokannasta antureiden datan ja muuttaa sen graafiseksi. Lisäksi WordPress, jolla voi luoda omat verkkosivut.

-  Raspberry Pi:ssä on oltava 7” tuuman näyttö, josta voit seurata luodun verkkosivun avulla antureiden keräämää dataa.
-  Varavirtalähteen akun on kestävä vähintään 4–5 päivää.
-  Laitteen kotelossa on oltava virtalähde paikka 9V virtalähteelle, jotta vesipumppu saa virtansa.
-  USB-A liitäntä, joka mahdollistaa varavirtalähteen lataamisen aina tarvittaessa.
-  Lämpötila on kerättävä kasvin maaperästä sekä huoneen lämmöstä.
-  Huoneen kosteus olisi myöskin kerättävä.
-  Jäljellä olevan veden mittaaminen vesiastiasta.
-  Maaperän kosteuden mittaaminen.
-  Vesipumppu osaa käynnistyä itsenäisesti ja myös sammua.

3.4 Verkkokäyttöliittymän suunnittelu

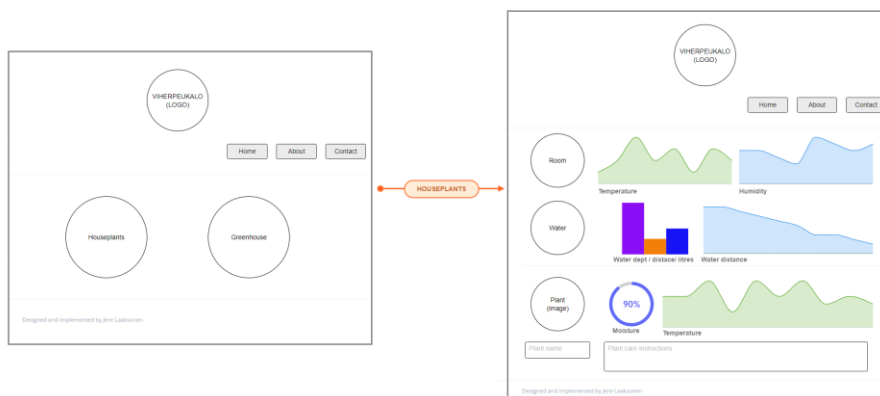
Verkkokäyttöliittymän suunnittelussa oli otettava huomioon ulkoasun lisäksi mikä on sivun käyttötarkoitus, onko sivujen käyttö käyttäjäystävällistä ja millä laitteella sivuja voidaan käyttää.

Verkkokäyttöliittymän ideana olisi se, että käyttäjä voisi seurata reaaliajassa huonekasvien hyvinvointia sekä huoneen lämpötilaa/kosteutta. Kasvien hyvinvoinnilla tarkoitetaan maan kosteutta, kasvien maalämpötilaa sekä vesiastiassa olevan veden määrää. Lisäksi hyvinvointiin kuuluu vesipumpun toimivuus, joka vaikuttaa maan kosteuteen.

Käyttäjä voi ottaa kuvia omista huonekasveista, jotka voidaan lisätä verkkosivuille kasvin kuvan kohdalle. Lisättyään oman kasvikuvan oikeaan paikkaan kasville voidaan antaa haluttu nimi. Lisäksi kasville voidaan lisätä kasvatusohjeita. Esimerkiksi millainen on valoisuuden tarve, tarvitseeko kasvi lannoitusta ja onko kasvi syötävä vai ei. Tämä toiminto helpottaa käyttäjää muistamaan tietyn kasvin tarpeet.

Verkkosivulla ei suoraan näy antureiden graafista näkymää. Graafisen näkymän saa sivuille mutta se on itse lisättävä. Jotta voisi saada graafisesti näkymään kasvin kasteluajavälin eri vuorokauden aikoina, on kyseisen anturin graafinen taulukko haettava Grafanasta ja linkitettävä se WordPress-sivulle. Tällä tavoin voidaan kaikkien muidenkin antureiden graafiset taulukot lisätä.

Verkkosivuja voidaan käyttää millä tahansa älylaitteella millä pääsee kodin sisäverkkoon. Esimerkiksi puhelin, tabletti ja tietokone. Verkkosivujen käyttö vaatii siis sisäverkon, joten sisäverkon ulkopuolelta ei sivulle ole mahdollista päästä. Tämä on toteutettu kustannussyistä vain sisäverkolle. Kuvassa 3 voi havaita verkkosivujen suunnitelman.



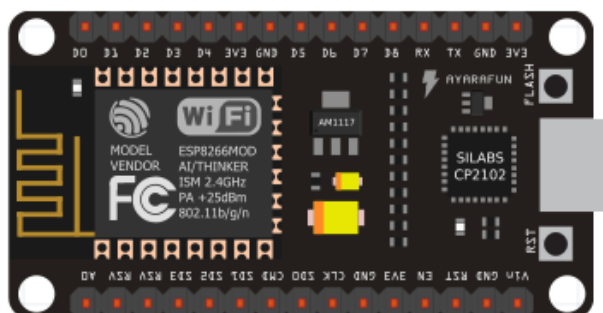
Kuva 3. Verkkosivujen suunnitelma.

4 KASTELUJÄRJESTELMÄN KOMPONENTIT JA TESTAUS

Kastelujärjestelmälle oli tehty vaatimusluettelo, jonka avulla pystyttiin valitsemaan oikeat komponentit järjestelmän toimivuuden kannalta. Tässä käydään työhön valikoituneet komponentit ja niiden toimintaperiaate.

4.1 NodeMCU ESP8266

NodeMCU on avoimen lähdekoodin IoT-alusta ESP-mikro-ohjaimille. Kuvassa 4 on ESP8266-mikro-ohjain, joka julkaistiin vuonna 2013. ESP8266-mikro-ohjaimella voi luoda omia elektroniikkaprojekteja edullisen hintansa sekä helppokäyttöisyyden ansiosta. Mikro-ohjaimen hyödyllisiä ominaisuuksia on Wi-Fi-ominaisuus, joka mahdollistaa laitteen yhdistämisen Wi-Fi-verkkoon. (NodeMCU Documentation, 2021).

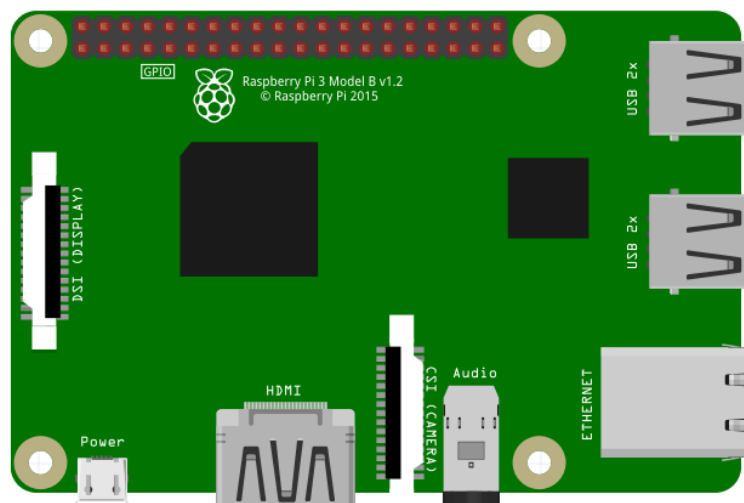


Kuva 4. ESP8266-mikro-ohjain.

4.2 Raspberry Pi

Raspberry Pi on pienikoinen yhden piirilevyn tietokone, jonka on kehittänyt brittiläinen Raspberry Pi Foundation vuonna 2012. Raspberry Pi on tehokas luottokortin kokoinen laite, jonka ansiosta ihmiset voivat opetella tekemään omia elektroniikkaprojekteja ja oppia ohjelmoimaan. Laite voidaan liittää tietokoneen näyttöön tai vaikka televisioon. Raspberry Pi pystyy tekemään kaiken sen mitä voit

tehdä tavallisella tietokoneella. Raspberry Pi käyttää ARM-mikroprosessoria. (Raspberry Pi Foundation, 2021) Kuvassa 5 on kuvattuna Raspberry Pi.



Kuva 5. Raspberry Pi.

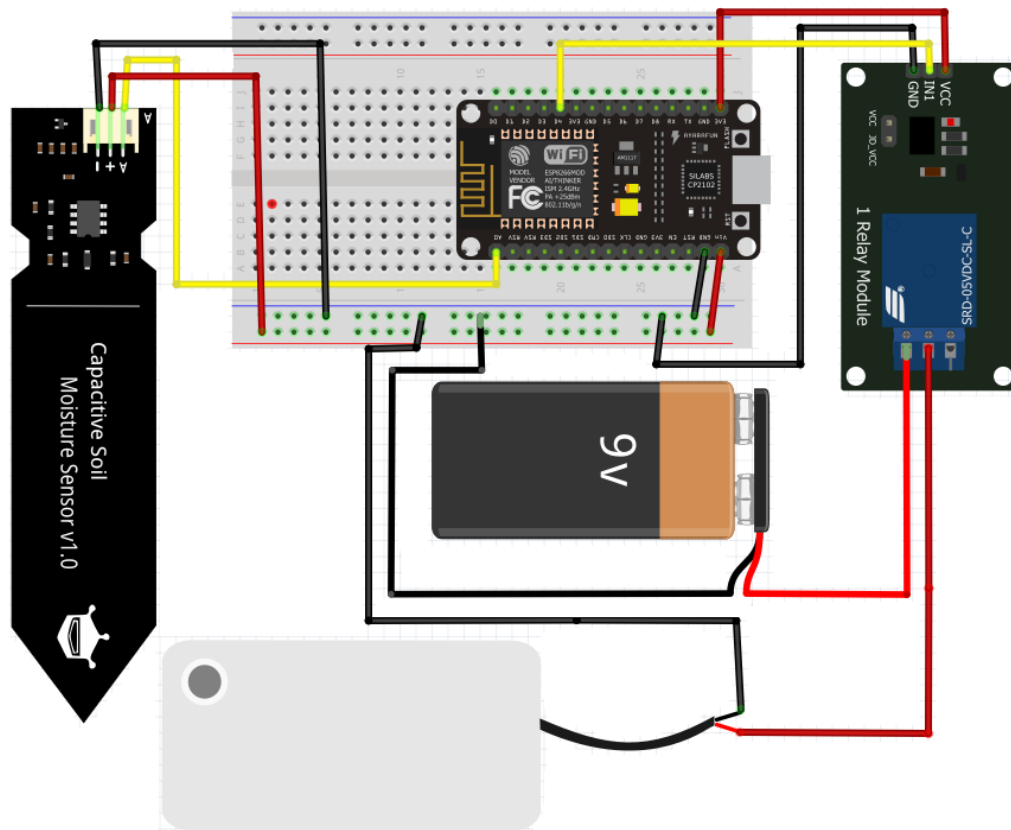
4.3 Kapasitiivinen maakosteusanturi ja vesipumppu

Kuvassa 6 on kuvattuna maakosteusanturi eli virallisesti kapasitiivinen maaperän kosteusanturi. Anturilla ei suoraan mitata maaperän kosteutta, vaan kapasitanssin aiheuttamaa muutosta. Muutos johtuu veden ja maan välisestä sähkövarauksesta. Eli siis lyhyesti sanottuna kapasitanssi vaihtelee maaperän vesipitoisuuden perusteella. (SwitchDoc Labs, 2020).



Kuva 6. Kapasitiivinen maakosteusanturi.

Automaattisessa kasvinkastelujärjestelmässä tarvitaan maakosteusanturin lisäksi vesipumppu ja rele. Anturin kapasitanssi mitataan 555-ajastinpohjaisella piirillä, joka tuottaa jännitteen, mikä on verrannollinen maaperään asetettuun kondensaattoriin ja voidaan tulkita maaperän kosteudeksi. Kapasitiivisen maakosteusanturin havaitessa maan kosteuden olevan liian kuiva niin 555-ajastinpiiri lähettää signaalin mikro-ohjaimelle, joka käynnistää vesipumpun automaattisesti releen avulla. Kun maan kosteus on riittävän kostea, sama 555-ajastinpiiri lähettää signaalin mikro-ohjaimelle, joka sammuttaa vesipumpun automaattisesti releen avulla. Kuvassa 7 on kuvattuna anturin ja vesipumpun kytkentä. (Circuit Schools, 2022 a), (Circuit Schools, 2022 b).



Kuva 7. Kapasitiivisen anturin ja vesipumpun kytkentä.

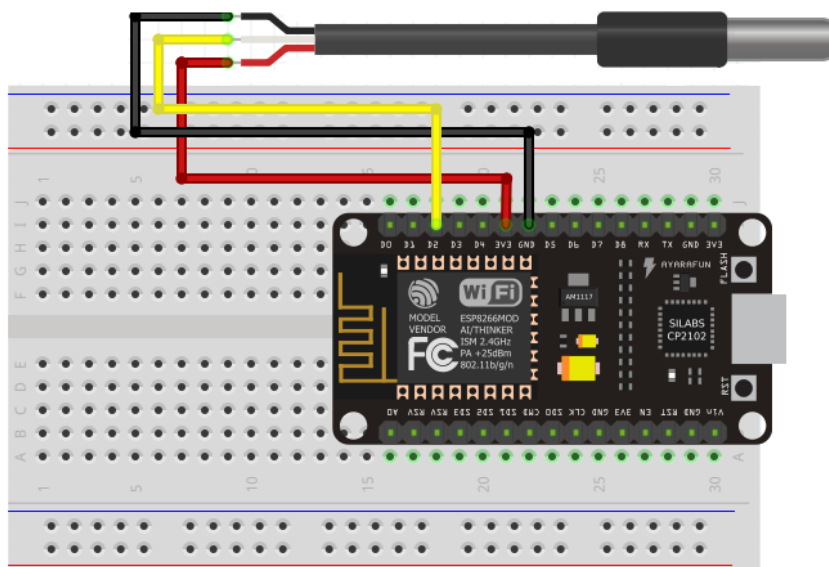
4.4 DS18B20-lämpötila-anturi

Kuvassa 8 on työhön valikoitunut DS18B20-yksijohtiminen digitaalinen lämpötila-anturi. Yksijohtimisella tarkoitetaan sitä, että tarvitaan vain yksi datajohto ja yksi GND-johto kommunikointiin esp8266:n kanssa. DS18B20 on päällystetty PVC-muovilla. Anturi soveltuu mainiosti maan lämpötilan mittaamiseen veden pitävyytensä takia. (Elecrow, 2019 a).

Jokaisella DS18B20-anturilla on oma 64-bittinen sarjakoodi, joka mahdollistaa sen, että voidaan kytkeä useita DS18B20-antureita samaan dataporttiin. Eli voidaan saada lämpötila useista eri kohteista käyttämällä vain yhtä pinniä. Kuvassa 9 on kuvattuna anturin kytkentä. (Elecrow, 2019 a).



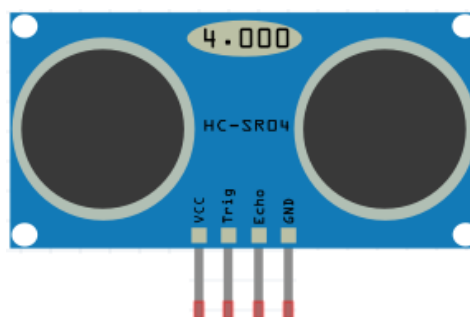
Kuva 8. DS18B20-anturi.



Kuva 9. DS18B20-anturin kytkentä.

4.5 HC-SR04-etäisyysanturi

Etäisyysanturilla tarkoitetaan HC-SR04-ultraäänianturia. Anturia käytetään ESP8266-kehitysalustan kanssa. HC-SR04-ultraäänianturi käyttää kaikuluotainta etäisyyden mittaamiseen kohteesta. Anturi pystyy lukemaan 2–400 cm:n päähän noin 0,3 cm tarkkuudella, Etäisyysanturin ominaisuuksia hyödyntäen voidaan mitata vesiastian tilavuutta ja ilmoittaa jäljellä olevan veden määrän litroina. Tämä toiminto tarvitsee ylimääräisiä laskutoimituksia. Laskutoimituksesta kerrotaan myöhemmin lisää. Kuvassa 10 on kuvattuna ultraäänianturi. (How To Mechatronics, 2022).

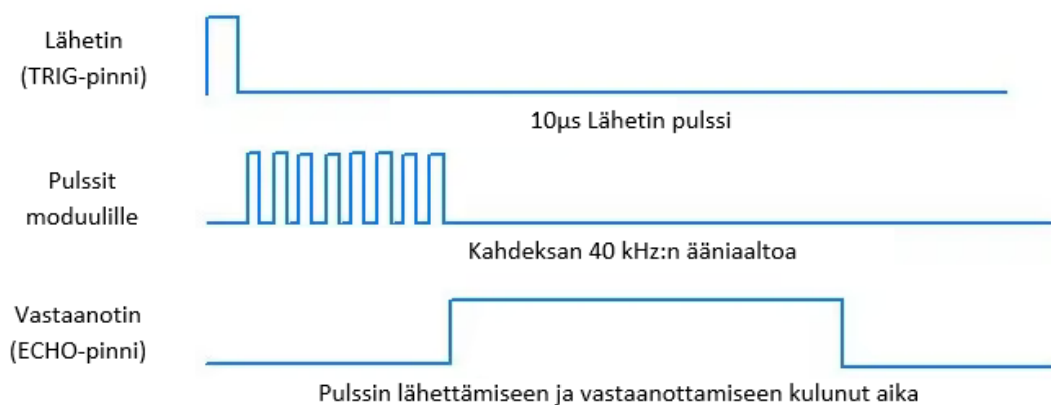


Kuva 10. HC-SR04-ultraäänianturi.

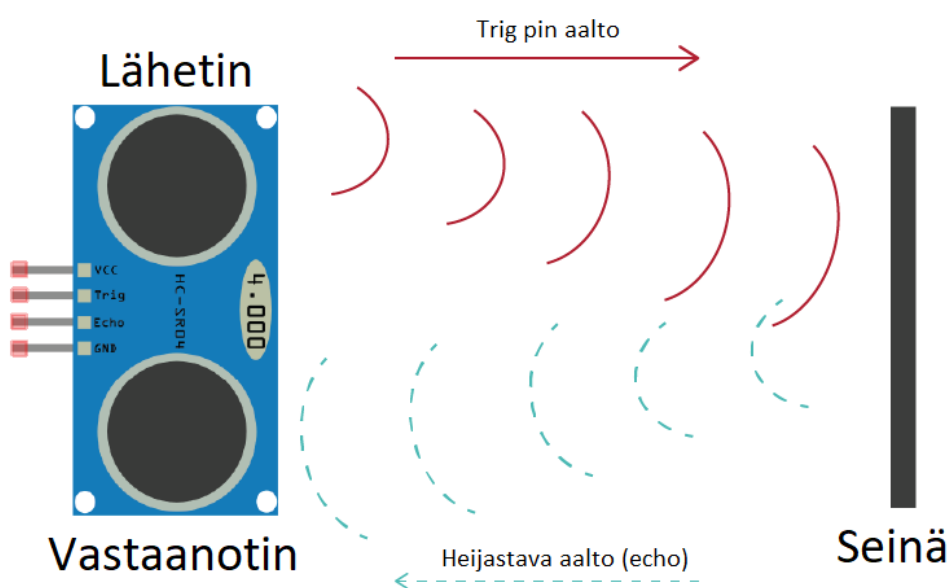
Kuvassa 11 ja kuvassa 12 on kuvattuna HC-SR04-anturin toiminta, joka toimii siten että ultraäänianturin lähetin (TRIG-pinni) lähettää 40 kHz pulssisignaalin, ihmiskorva ei kuitenkaan voi kuulla ääntä koska se on korkeataajuinen. TRIG-pinnin lähettämä signaali kulkee ilman läpi. Taajuuden osuttua objektiin äänitaajuus palaa takaisin ultraäänianturin vastaanottimeen (ECHO-pinniin), joka vastaanottaa heijastuneen äänen. Jotta pystytään saamaan oikea etäisyys, on otettava huomioon myös äänen nopeus ilmassa ja matkan aika. Matkan ajalla tarkoitetaan taajuuden lähettamisestä ja vastaanottamisesta kulunutta aikaa. Etäisyyden voi laskea kaavalla 1. (ElecFreaks, 2014.)

$$\text{äänen nopeus ilmassa} = 343 \text{ m/s}$$

$$etäisyys = \frac{(343 \text{ m/s} * aika)}{2} \quad (1)$$



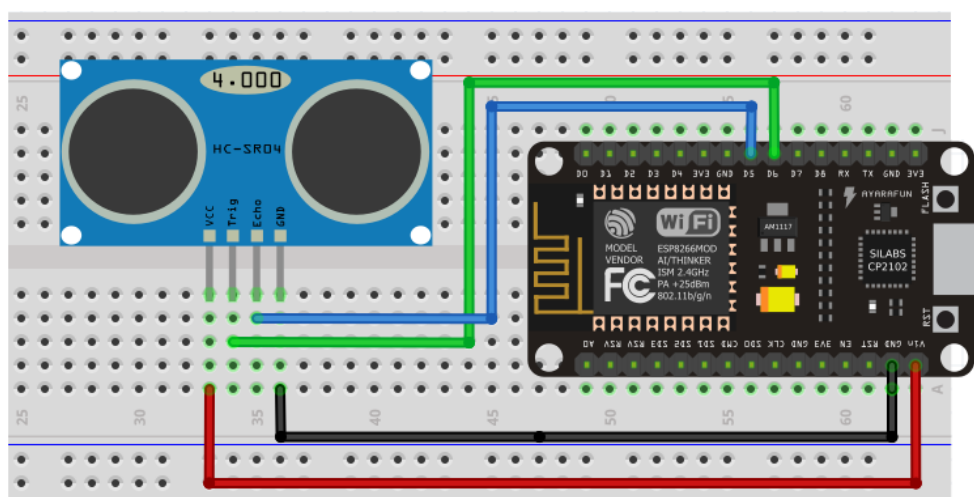
Kuva 11. HC-SR04-anturin aikakaavio. (ElectronicWings, 2021).



Kuva 12. HC-SR04-anturin toimintaperiaate. (Random nerd tutorials, 2021).

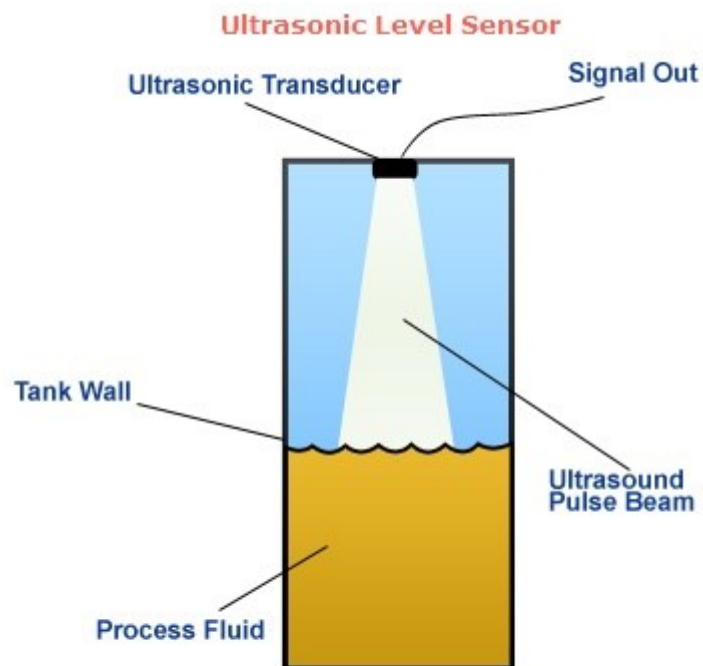
4.5.1 Vedenpinnan mittaaminen

Saadakseen selville vesiastiassa jäljellä olevan veden litramäärän, oli tiedettävä vedenpinnan taso. Tämän saavuttamiseksi oli käytettävä ultraäänianturia. Kuvassa 13 on kuvattuna anturin kytkentäkaavio.



Kuva 13. HC-SR04-anturin kytkentäkaavio.

Ultraäänianturin tarkoituksena oli mitata lähetin-vastaanottimen ja nesteen välinen etäisyys. Kuten aiemmin selitetty ultraäänianturin toimintaperiaate, lähetetään lyhyt ultraäänipulssi ja mitataan sen matka-aika lähettimestä nesteeseen ja takaisin vastaanottimeen. Ultraäänipulssi heijastuu nesteen tasolta, koska ultraäänipulssin edetessä tiheys muuttuu. Eli ultraäänipulssi kulkee ensin ilman läpi ja heijastuu ilmaa tiheämmällä nesteellä takaisin lähettimeen. Koska veden tiheys on suurempi kuin ilman, suurin osa pulssista heijastuu pois. Kuvassa 14 on kuvattuna etäisyyden mittaaminen vesiastiasta. (Instructables circuits, 2022 b).



© 2010 Chipkin Automation Systems Inc.

Kuva 14. HC-SR04-etäisyyden mittaaminen vedenpinnasta. (AutomationWiki, 2018).

Käytettäessä ultraäänianturia oli otettava huomioon kaksi ongelmaa. Yksi ongelmista oli lähettimen lähettämä pulssin pituus. Jos pulssin pituus oli liian pieni vastaanotin ei pysty vastaanottamaan pulssia. Ongelman pystyy korjaamaan helposti sijoittamalla anturin veden maksimitasosta muutamaa senttimetriä korkeammalle, jotta vastaanottimella on aikaa vastaanottaa. Toinen ongelma on säiliön halkaisijan koko. Jos säiliön halkaisija on liian pieni, signaali voi heijastua säiliön seinästä, joka johtaa virheellisiin lukemiin. (Instructables circuits, 2022 b). Mitäustulokset on kirjattu taulukkoon 1.

Taulukko 1. Mittaustulokset.

| Mittaus | Etäisyys | Mitattu etäisyys | Virhe arvo |
|----------------|-----------------|-------------------------|-------------------|
| 1. | 5 cm | 4,45 cm | 11 % |
| 2. | 8 cm | 7,51 cm | 6,13 % |
| 3. | 10 cm | 9,47 cm | 5,30 % |
| 4. | 12 cm | 11,63 cm | 3,08 % |
| 5. | 14 cm | 13,58 cm | 3 % |
| 6. | 16 cm | 15,64 cm | 2,25 % |
| 7. | 18 cm | 17,65 cm | 1,94 % |
| 8. | 20 cm | 19,7 cm | 1,50 % |
| 9. | 22 cm | 21,68 cm | 1,45 % |
| 10. | 25 cm | 24,67 cm | 1,32 % |

4.5.2 Vesisäiliö

Kastelujärjestelmän vesisäiliönä käytetään kuvassa 15 olevaa Plastex'in valmistamaa muovista vesiastiaa. Säiliöön mahtuu 5 L vettä. Säiliön kanteen on porattava kolme reikää, joista kaksi on varattu ultraäänianturille, jotta anturi voi lukea lähettämänsä pulssin veden pinnasta. Viimeinen porattu reikä on varattu vesijohdolle, jotta vesipumppu voisi pumpata vettä kasville aina tarvittaessa.



Materiaali: Muovi

Korkeus: 25 cm

Leveys: 15 cm

Syvyys: 17 cm

Tilavuus: 5L

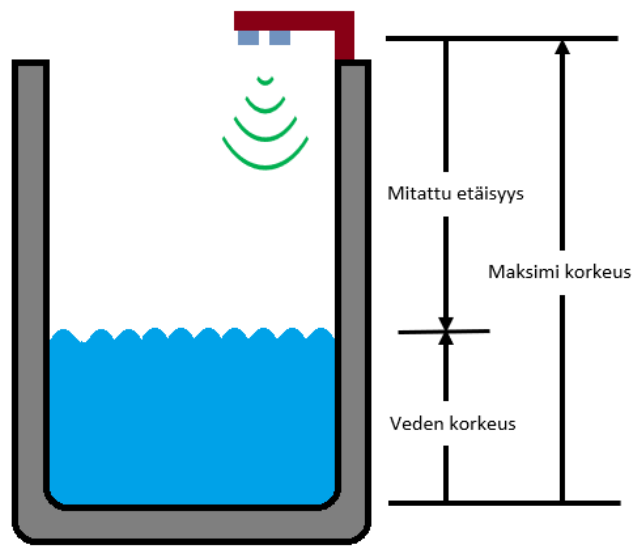
Kuva 15. Plastex'in vesiastia.

4.5.3 Ohjelmointi

Ultraäänianturille löytyi jo valmis testikoodi etäisyyden mittaamiseen, johon anturi on alun perin tarkoitettu. Otin testikoodin käyttöön mutta tarvitsin kuitenkin anturin mittaamaan vedenkorkeutta, joten tein muutamia muutoksia ohjelmaan.

Ohjelman toimintaperiaate on hyvin simppele. Aluksi oli saatava aika lähetetyn signaalipulssin ja vastaanotetun signaalin väliltä, jotta se voidaan muuttaa senttimetreiksi. Eli siis anturi lähettää signaalipulssin kohti veden pintaa, josta se heijastuu takaisin vastaanottimelle. Tämän jälkeen, kun etäisyys veden pintaan on saatu, on

seuraavaksi tiedettävä veden korkeus. Tiedämme astian maksimikorkeuden, josta on vähennettävä aikaisemmin mitattu etäisyys. Tämä voidaan laskea kaavalla 2. Jäljellä olevan vesimäärän laskeminen litroiksi lasketaan tilavuuden kaavalla eli kaavalla 3, missä on muistettava, että korkeuden tilalle on laitettava vedenkorkeus eikä vesiastian korkeutta. Lopuksi lasketaan jäljellä oleva veden määrä prosentteina kaavalla 4. Kuvassa 16 on kuvattuna millä tavalla jäljellä oleva veden määrä on mitattu.



Kuva 16. HC-SR04-vedenkorkeuden mittaaminen.

$$K = \text{Korkeus} \quad L = \text{Leveys} \quad S = \text{Syvyys}$$

$$\text{Vedenkorkeus} = K - \text{etäisyys} \quad (2)$$

$$\text{Vesiastian Tilavuus} = \frac{K * L * S}{1000}$$

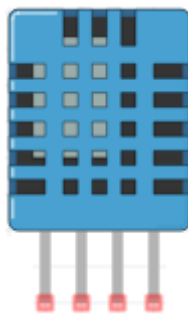
$$\text{Jäljellä oleva veden määrä litroina} = \text{Vedenkorkeus} * \left(\frac{\text{Tilavuus}}{K} \right) \quad (3)$$

$$\text{Jäljellä oleva veden määrä prosentteina} = \frac{\text{Vedenkorkeus} * 100}{K} \quad (4)$$

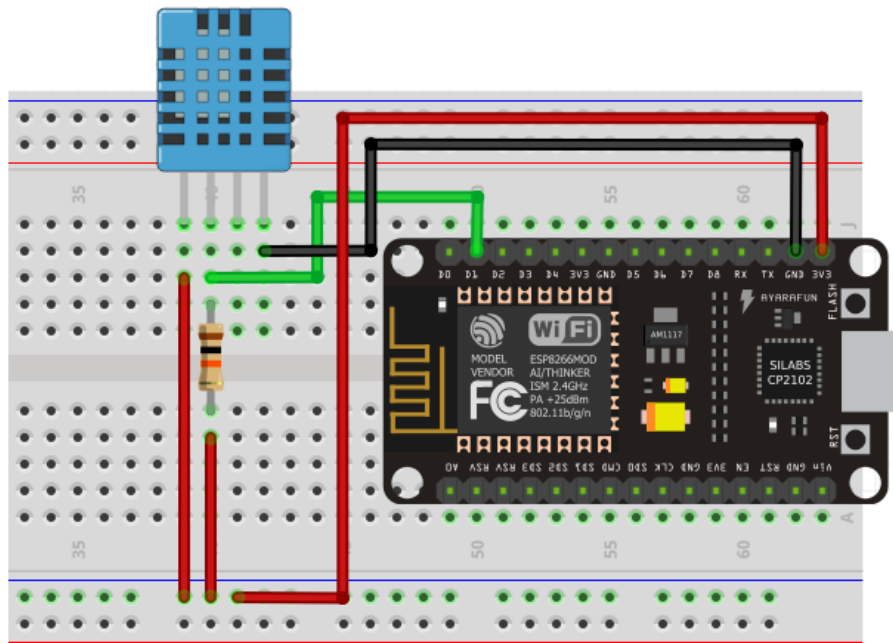
4.6 DHT11-lämpötila- ja kosteusanturi

Kuvassa 17 on kuvattuna DHT11-lämpötila- ja kosteusanturi, jolla nimensä mukaisesti mitataan huoneen lämpötilaa ja kosteutta. Anturissa on kalibroitu digitaalinen signaalilähtö. Tämän ansiosta voimme varmistaa mittaustuloksiin korkean luotettavuuden. Lisäksi anturissa on kosteuden mittaamiseen perustuva resistiivinen elementti ja lämpötilan mittaamiseen perustuva NTC-vastus. (Elecrow, 2019)

b) Kuvassa 18 on kuvattuna lämpötila- ja kosteusanturin kytkentä.



Kuva 17. DHT11-anturi.

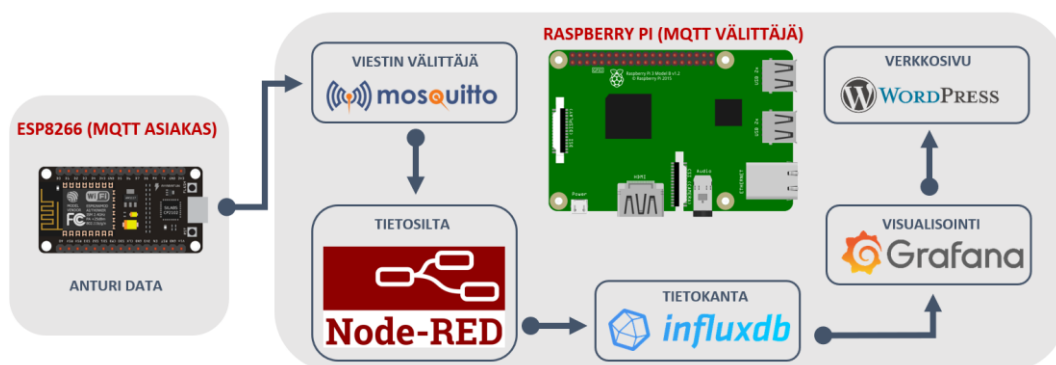


Kuva 18. DHT11-anturin kytkentä.

5 MQTT-PALVELIN RASPBERRY PI:LLE

MQTT (Message Queuing Telemetry Transport) on viestintä- ja tiedonvaihtoprotokolla, jota käytetään yleisesti IoT-laitteiden väliseen viestintään. MQTT:n protokolla on suunniteltu julkaisu-/tilausviestintäkuljetukseksi (publish/subscribe), jonka avulla verkon reunalaitteet voivat julkaista välittäjälle (broker). Asiakkaat (client) muodostavat yhteyden välittäjään (broker), joka sitten lähettää viestintää kahden laitteen välillä. Jokainen laite voi tilata tai rekisteröityä tiettyihin aiheisiin (topic). Kun toinen asiakas julkaisee viestin tilatusta aiheesta, välittäjä välittää viestin edelleen mille tahansa asiakkaalle, joka on tilannut. (MQTT, 2020).

Opinnäytetyössä julkaisijana toimii ESP8266. Välittäjänä on käytössä Raspberry Pi, johon oli asennettu Mosquitto broker-ohjelma. Lisäksi Raspberry Pi:hin oli asennettu Node-Red, influxdb, Grafana sekä WordPress. joista lisää myöhemmin. Kuvassa 19 on kuvattuna kuinka anturi data siirtyy paikasta toiseen.



Kuva 19. Datan siirtyminen.

5.1 Mosquitto MQTT-välittäjä

Eclipse Mosquitto on avoimen lähdekoodin viestinvälittäjä. Mosquitto hyödyntää MQTT-protokollaa, jonka avulla pystytään suorittamaan viestintää käyttämällä julkaisu-/tilausmallia. (Mosquitto, 2021).

5.1.1 Asennus

Ennen kuin oli mahdollista asentaa Mosquito-välittäjä, oli tiedettävä Raspberry Pi:n IP-osoite. IP-osoitetta tarvittiin myöhemmin, jotta MQTT-asiakas voitiin yhdistää MQTT-välittäjään. Kuvassa 20 on Raspberry Pi:n IP-osoite. (SuperHouse, 2019).

```
seed@planter:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.255 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::bba9:cc48:7fcb:2c04 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:13:fd:c0 txqueuelen 1000 (Ethernet)
    RX packets 2507778 bytes 785855127 (749.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1260732 bytes 176360068 (168.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Kuva 20. Raspberry Pi:n IP-osoite.

Kun IP-osoite oli saatu selville, asennettiin Raspberry Pi:hin ”mosquitto broker” ja ”mosquitto-clients”. Mosquitto-välittäjä olisi kyllä riittänyt mutta asiakkaan avulla olisi mahdollista tehdä testejä käyttäen terminaalin komentoriviä. Seuraavalla komennolla asennettiin mosquitto, joka näkyy kuvassa 21. (SuperHouse, 2019).

```
seed@planter: ~ $ sudo apt install -y mosquitto mosquitto-clients
```

Kuva 21. Mosquiton asennuskomento.

Mosquitto-välittäjä toimisi jo näin suoraan asennuksen jälkeen. Mutta se ei ole oletukseltaan kovinkaan turvallinen, joten oli hyvä määrittää laitteelle käyttäjätunnus ja salasana. Suojauksen tekeminen oli hyvin yksinkertainen, Tarvitsi vain tehdä tekstitiedosto, joka sisälsi käyttäjätunnuksen ja salasanan, jotka erotetaan kaksoispisteellä. Kuvassa 22 olevalla komennolla mahdollistettiin tehdä laitteelle käyttäjätunnus ja salasana. (SuperHouse, 2019).

```
seed@planter: ~ $ echo "mqtt_omakäyttäjä:mqtt_omasalasana" > pwfile
```

Kuva 22. Käyttäjätunnuksen ja salasanan luominen.

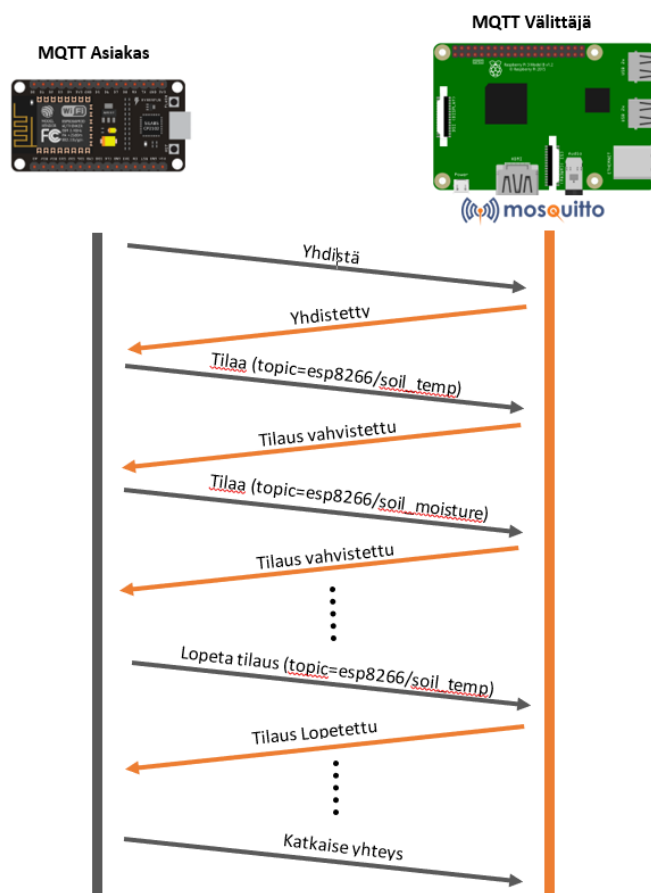
Jotta laitteesta saatiin vielä turvallisempi, oli "pwfile" tiedosto salattava passwd-apuohjelmaa käyttäen, ohjelma tuli mosquito-paketin mukana suoraan. Seuraavalla komennolla käyttäjätunnus ja salasana salataan. Eli suomeksi sanottuna käyttäjätunnuksesta ja salasanaasta tuli lukukelvotonta hölynpölyä mitä ei ole ihmisen mahdollista lukea. Kuvassa 23 on komento salasanan salaamiseksi. (SuperHouse, 2019).

```
seed@planter: ~ $ mosquito_passwd -U pwfile
```

Kuva 23. Salasanan salaaminen.

5.2 ESP8266 MQTT-asiakas

ESP8266-mikro-ohjain ottaa yhteyden palvelimena toimivaan Raspberry Pi tietokoneeseen ja tilaa (subscribe) "/kysely" MQTT-topicin. Halutessaan anturidataa, Raspberry Pi lähettää viestin "/kysely"-topic:iin, jota ESP8266 seuraa. Viestin saapuessa, ESP8266 tulkitsee viestin ja lähettää vastauksen "/vastaus"-topic:iin, jota vastaavasti Raspberry Pi seuraa. Periaate Raspberry Pi:n ja ESP8266 välisestä MQTT-keskustelusta on esitetty kuvassa 24.



Kuva 24. MQTT-viestintä.

5.2.1 ESP8266 yhdistäminen MQTT-välittäjään

Yhdistäessä ESP8266-mikro-ohjaimen MQTT-välittäjään, jota ylläpitää Raspberry Pi. ESP8266 oli ohjelmoitu Arduino IDE:n avulla.

Saadakseen yhteyden MQTT-välittäjään oli ihan aluksi ladattava ESP8266Wifi.h- ja PubSubClient.h-kirjastot Arduino IDE:n Tools -> Library Manager paikasta. ESP8266Wifi.h-kirjasto mahdollisti ESP8266:n yhdistämisen Wi-Fi-verkkoon, ja PubSubClient.h-kirjasto mahdollisti ESP8266:n yhteyden MQTT-välittäjään, jolla mahdollistettiin viestien julkaiseminen ja aiheiden tilaaminen. (Techtutorialsx, 2022), (Random nerd tutorials, 2022).

Kun kirjastot oli saatu ladattua, oli seuraavaksi asetettava oman verkon Wi-Fi nimi ja salasana verkolle. Lisäksi oli annettava Raspberry Pi:n IP-osoite sekä MQTT-välittäjän käyttäjänimi ja salasana ja portin numero. Kuvassa 25 on kuvattuna tarvittavat kirjastot ja verkon määrittäminen. (Techtutorialsx, 2022).

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

#define WIFI_SSID "OMA_SSID"
#define WIFI_PASSWORD "OMA_SALASANA"

#define MQTT_HOST "192.168.1.XXX" // Raspberri Pi Mosquitto MQTT Broker
#define MQTT_PORT 1883
#define MQTT_USER "OMA_KÄYTTÄJÄ"
#define MQTT_PASSWORD "OMA_SALASANA"
```

Kuva 25. Kirjastojen, Wi-Fi:n ja MQTT:n määrittäminen.

Tämän jälkeen, kun Wi-Fi ja MQTT-välittäjän määriykset on saatu tehtyä, oli otettava yhteys Wi-Fi-verkkoon. Yhteyden muodostuttua Wi-Fi-verkkoon ottaa mikroohjain seuraavaksi yhteyden MQTT-välittäjään, jonka jälkeen laita on valmis julkaisemaan viestit ja tilamaan MQTT-välittäjän. Kuvassa 26 on kerrottuna, miten yhteyden ottaminen on toteutettu. (Techtutorialsx, 2022).

```
void setup_wifi()
{
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
  client.setServer(MQTT_HOST, MQTT_PORT);
  client.setCallback(callback);
  while (!client.connected())
  {
    Serial.println("Connecting to MQTT...");
    if (client.connect("ESP8266Client", MQTT_USER, MQTT_PASSWORD))
    {
      Serial.println("connected");
    }
    else
    {
      Serial.print("failed with state ");
      Serial.print(client.state());
      delay(2000);
    }
  }
}
```

Kuva 26. ESP8266 yhdistäminen Wi-Fi-verkkoon.

5.2.2 ESP8266-aihe

Jokaiselle anturille oli määriteltävä oma aihe eli *topic*. Aiheella tarkoitetaan UTF-8-merkkijonoa, jota välittäjä käyttää viestien suodattamiseen. Aihe koostuu yhdestä tai useammasta aihetasosta. Aihetasot on erotettava vinoviivalla. Kaikkien antureiden aiheet ovat kuvassa 27. (Steve's Internet Guide, 2022), (Random nerd tutorials, 2022).

```
#define SOIL_TEMPERATURE_TOPIC "esp8266/soil_temp"
#define ROOM_TEMPERATURE_TOPIC "esp8266/room_temp"
#define ROOM_HUMIDITY_TOPIC "esp8266/room_hum"
#define SOIL_MOISTURE_TOPIC "esp8266/soil_moisture"
#define SOIL_MOISTURE_PERCENT_TOPIC "esp8266/soil_moisture_percent"
#define WATER_DISTANCE_TOPIC "esp8266/water_distance"
#define WATER_DEPTH_TOPIC "esp8266/water_depth"
#define WATER_LITRES_TOPIC "esp8266/water_litres"
#define WATER_PERCENT_TOPIC "esp8266/water_percent"
#define WATER_PUMP_MESSAGE_TOPIC "esp8266/water_pump_message"
```

Kuva 27. MQTT-aihe.

Kun antureille oli saatu annettua MQTT-aiheet, voitiin aiheen nimi tulostaa sarjaporttiin, jonka jälkeen vastaanotettujen tietojen jokainen tavu tulostetaan kuvassa 28 olevalla tavalla. (Random nerd tutorials, 2022).

```
void callback(char* topic, byte* payload, unsigned int length)
{
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);

  Serial.print("Message:");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
  }

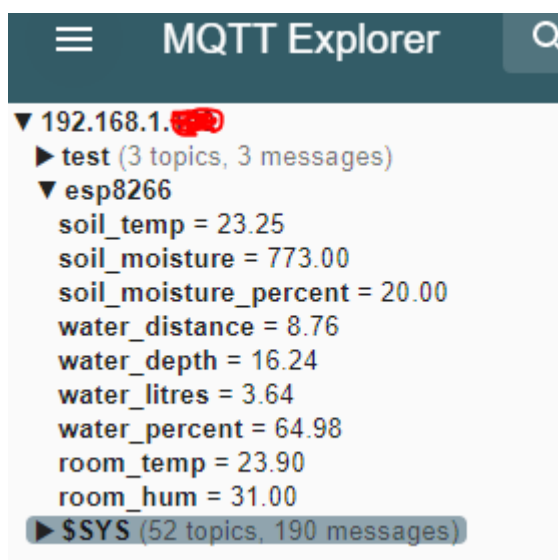
  Serial.println();
  Serial.println("-----");
}
```

Kuva 28. MQTT-aiheen tulostaminen.

Esimerkki kuvassa 29 miten maalämpötila-anturi mittaa lämpötilaa ja lähettää sen MQTT-välittäjälle. Kuvassa 30 on nähtävissä MQTT-välittäjälle vastaanotettu lämpötila.

```
void temperature()
{
  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0);
  Serial.print("Soil Temperature = ");
  Serial.print(temp);
  Serial.println(" °C");
  client.publish(SOIL_TEMPERATURE_TOPIC, String(temp).c_str(), true);
}
```

Kuva 29. Lämpötilan mittaaminen.

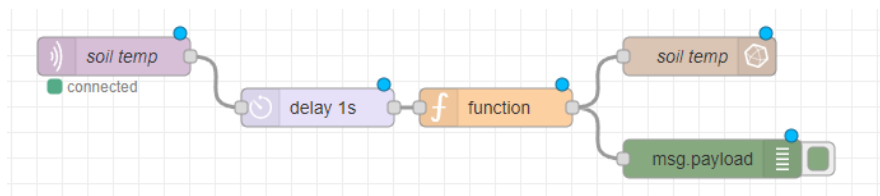


Kuva 30. MQTT-välittäjälle saapunut lämpötila.

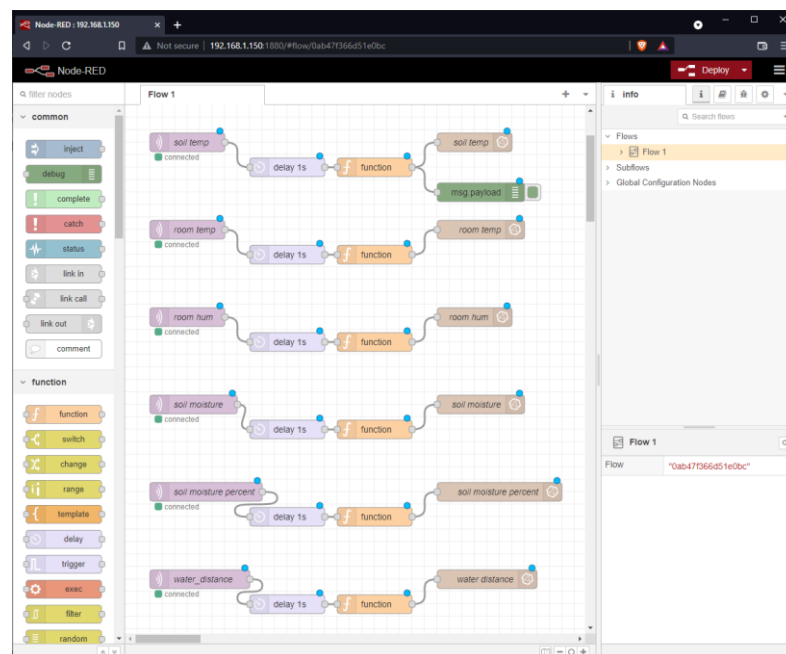
5.3 Node-RED

Node-RED on virtauspohjainen ohjelmointityökalu laitteistojen, API sekä verkkopalvelujen yhdistämiseen. Ohjelmointityökalun on alun perin kehittänyt IBM vuonna 2013. (Node-RED, 2021).

Kuvassa 31 on kuvattuna viisi laatikkoa. Ensimmäinen laatikko on yhdistetty MQTT-välittäjään. Lisäksi laatikolle on annettava tilattavan anturin "topic" jotta voimme lähettää oikean anturin datan tietokannan oikeaan paikkaan. Toisessa laatikossa on nimensä mukaisesti 1 sekunnin viive. Kolmannessa laatikossa on funktio, johon on kirjoitettu pieni koodinpätkä, joka osaa muuttaa saapuvan anturin datan string-muodosta float-muotoon. Neljäs laatikko on tietokanta, jonne on määritetty oikea paikka kyseisen anturin datalle. Viides eli vihreän laatikon merkitys on vain se, että voit seurata tulostuuko anturin data oikein. Kuvassa 32 on kuvattuna jokaisen anturin datan siirtyminen tietokantaan (SuperHouse, 2019).



Kuva 31. Node-RED anturin data tietokantaan.



Kuva 32. Node-RED kaikki anturit.

5.4 InfluxDB

InfluxDB on avoimen lähdekoodin aikasarjatietokanta. InfluxDB-tietokannan on kehittänyt InfluxData vuonna 2013. Aikasarjatietokannan avulla pystytään tallentamaan aikaleimattua antureiden dataa. Tämän ansiosta saadaan selville jokaisen anturilta tulevan datan aikaleimat, milloin tulos on saapunut. (Influxdata, 2021).

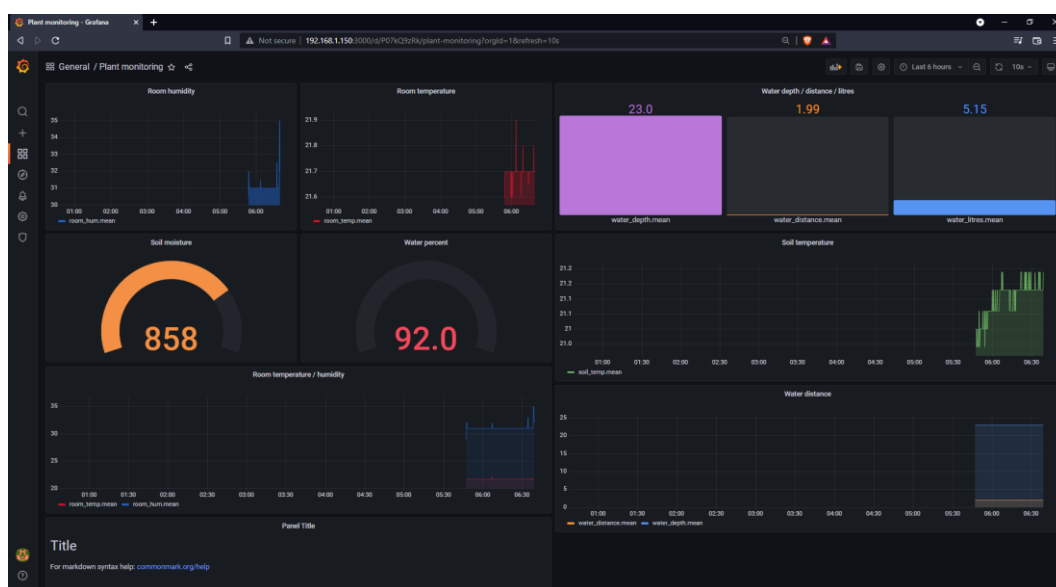
Kuvassa 33 on otettu yhteys tietokantaan. Tietokannassa näkyy jokaisen anturin ”topic”. Esimerkiksi on otettu maalämpötila-anturin 10 viimeisintä topic ”soil_temp”-viestiä. Kuten kuvasta voi nähdä jokaisella arvolla on oma aikaleima.

```
>
> SHOW DATABASES
name: databases
name
----
_internal
sensors
> SHOW SERIES
key
---
esp8266/soil_temp
room_hum
room_temp
soil_moisture
soil_moisture_percent
soil_temp
water_depth
water_distance
water_litres
water_percent
> select * from "soil_temp" limit 10
name: soil_temp
time                value
----                -
1638283735622080422 20.88
1638283755336241703 20.75
1638283775333972918 20.81
1638283795526870307 20.81
1638283815367367673 26.44
1638283835348432879 30.31
1638283855334580680 32.25
1638283875987294039 31.88
1638283895352281479 30.81
1638283915368889277 29.94
> █
```

Kuva 33. InfluxDB-tietokanta.

5.5 Grafana

Grafana on avoimen lähdekoodin visualisointiohjelma. Grafana julkaistiin vuonna 2014. Kun Grafana yhdistetään tietokantaan, tässä tapauksessa InfluxDB aikasarjatietokantaan saadaan hyödynnettyä visualisointiohjelman hyödyllisiä toimintoja. Voidaan luoda antureille halutun näköisiä graafisia esityksiä tai vaikka lämpökarttoja. (Grafana Labs, 2021), (SC& SCAND, 2021). Kuvassa 34 on kuvattuna millä tavalla antureiden data näkyy Grafanassa.

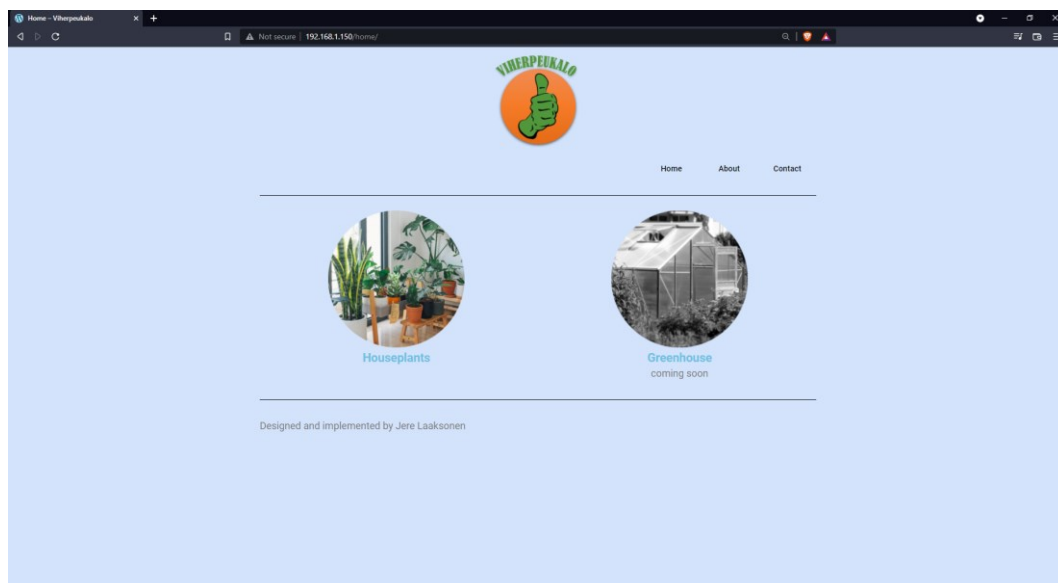


Kuva 34. Antureiden data Grafanassa.

5.6 WordPress

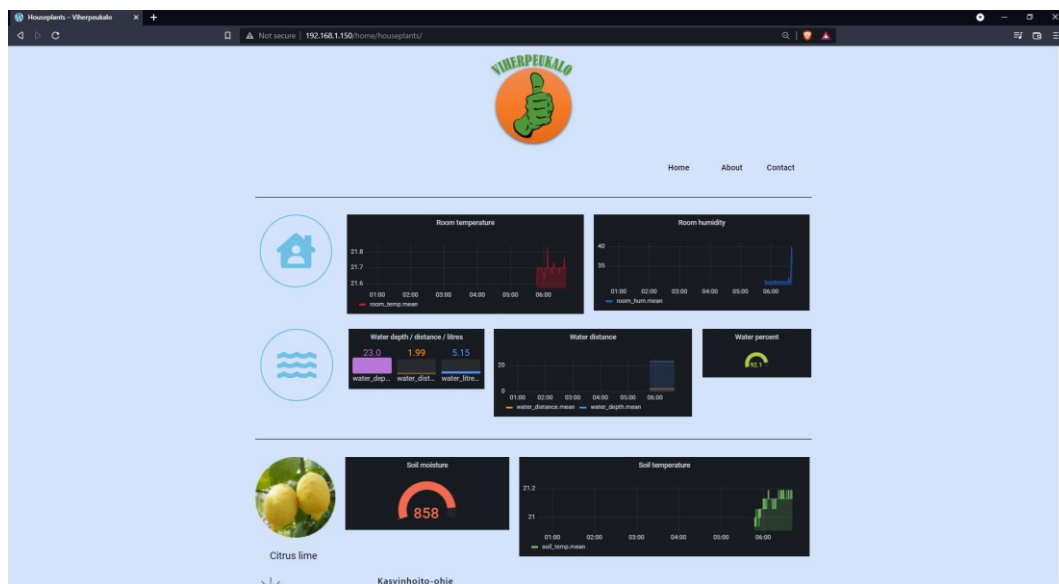
WordPress on sisällönhallintajärjestelmä, jonka avulla voit luoda omia verkkosivustoja. WordPress on julkaistu vuonna 2003. (WordPress.org, 2021).

Kuvassa 35 on kuvattuna WordPressillä luotujen omien verkkosivujen aloitussivu, josta voi valita haluaako siirtyä huonekasveihin vai kasvihuoneen kasveihin.



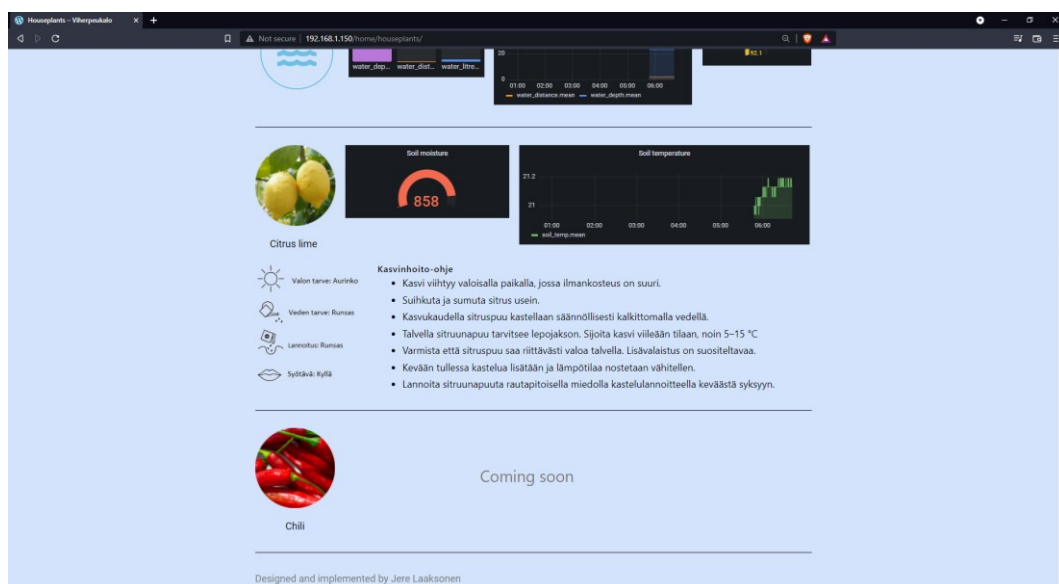
Kuva 35. Verkkosivujen aloitussivu.

Kuvassa 36 on kuvattuna mitä tietoja on mahdollista nähdä huonekasvien kohdalla. Sivun alussa näkyy yleisesti ottaen huoneen lämpötila sekä kosteus. Alempana on mainittu vesiastiassa jäljellä olevan veden määrän tietoja. Kasvin kohdalla on mainittuna maan kosteus sekä maan lämpötila. Jokaiselle kasveille on laitettava omat anturit, jotta jokaista kasvia voidaan seurata.



Kuva 36. Verkkosivujen kasvisivu 1/2.

Kuvassa 37 on kuvattuna miten jokaiselle kasville on mahdollista antaa omia hoito-ohjeita, jotta eri kasvien hoito tarpeen muistaa paremmin.



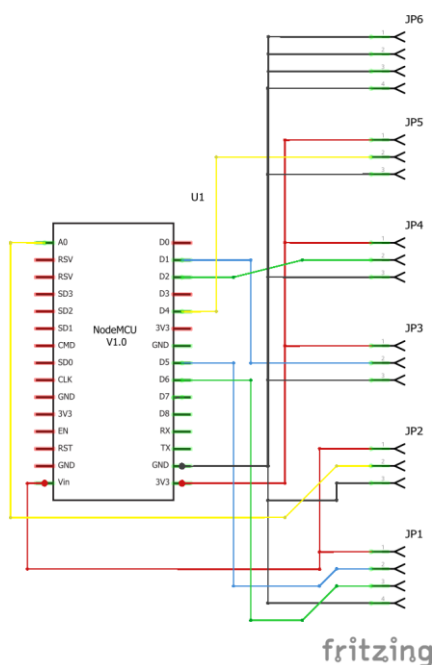
Kuva 37. Verkkosivujen kasvisivu 2/2.

6 KASTELUJÄRJESTELMÄN VALMISTUS

Kastelujärjestelmän valmistelussa oli tehtävä kytkentäkaavio, jotta pystyttiin valmistamaan kytkentälevy. Lisäksi oli laskettava varavirtalähteen kulutus, jonka avulla saadaan selville, kuinka kauan akku kestää ja kauanko latausaika on varavirtalähteellä täyteen ladattuna. Kastelujärjestelmän käyttöä varten on kolmelle anturille suunniteltu ja tulostettu kotelot Solidworks-ohjelmalla, jotta antureiden komponentit eivät joudu kosketuksiin ihmisten eikä myöskään lian kanssa. Tämä pitää anturit hyväkuntoisena pidempään ja lisää tietenkin käyttöikä. Antureiden 3D-piirrokset ovat nähtävissä liitetiedostoina.

6.1 Kytkentäkaavio

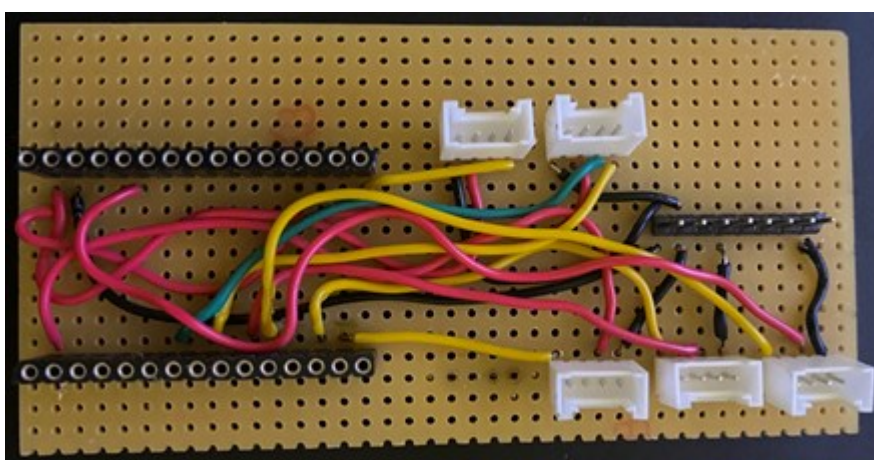
Kuvassa 38 on kuvattuna NodeMCU ESP8266-mikro-ohjain kytkentäkaavio, johon on kytketty 5 kappaletta Crowtailin 4-pinnisiä runkoliittimiä. Ainoastaan 1 liitin tarvitsee kaikki 4 pinniä. Ylhäällä oleva JP6 on maa, joten kaikki pinnit menevät samaan paikkaan. Kytkentä on piirretty fritzing-ohjelmalla.



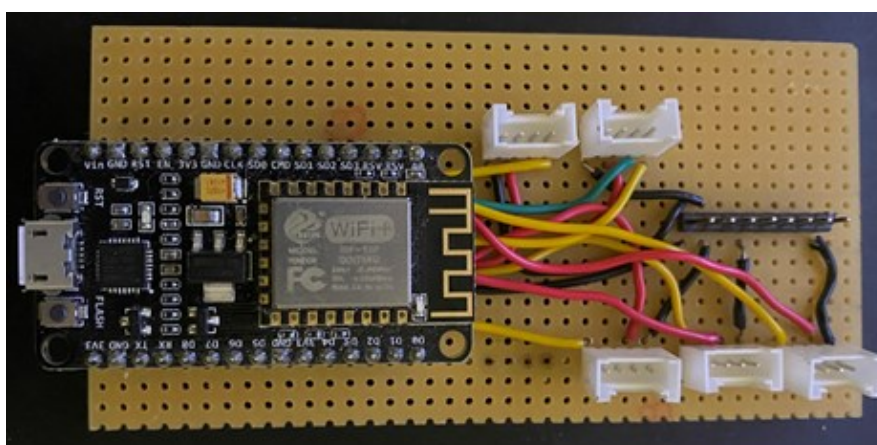
Kuva 38. Kytkentä.

6.2 KytKentälevyn valmistus

Kuvassa 39 on kytKentälevylle juotettu 5 kappaletta Crowtailin 4-pinnisiä runkoliittimiä. Runkoliittimet ovat juotettuna oikeille pinnipaikoille. Keskellä levyssä on maajohdoille lisättyä nastoja, joihin tulee kytkeä vesipumppu ja sähkövirtajohto. Runkoliittimien avulla anturit on helppo kytkeä oikeille paikoille. Jokaiselle anturille on juotettu vastakappale, joka liitetään runkoliittimeen. Kuvassa 40 on ESP8266 liitettynä kytKentään.



Kuva 39. KytKentälevylle tehty juotos.



Kuva 40. ESP8266 liitettynä kytKentälevylle.

6.3 Varavirtalähde

Opinnäytetyöhön valikoitui Puro:n 20 000mAh varavirtalähde. Varavirtalähteessä on 2 kpl USB-A-porttia, joista molemmista saa ulostulona 5V/2,4A -12W.

ESP8266-mikro-ohjaimen käyttövirran kulutus on 80mA – 400mA riippuen käyttö-tarkoituksesta. Lepotilassa virrankulutus on 20 μ A.

Jotta voidaan saada selville kauanko akku kestää varavirtalähteessä on selvitettävä akun kapasiteetti, joka on ilmoitettu jo aikaisemmin.

Lisäksi oli arvioitava, kuinka suuri sähkövirran kuorma keskimäärin kulkee. Alapuo- lella on laskettu kaavalla 5 akun kesto eri virrankulutustilanteissa.

$$Akun\ kesto = \frac{Akun\ kapasiteetti\ (mAh)}{Virrankulutus\ (mA)} \quad (5)$$

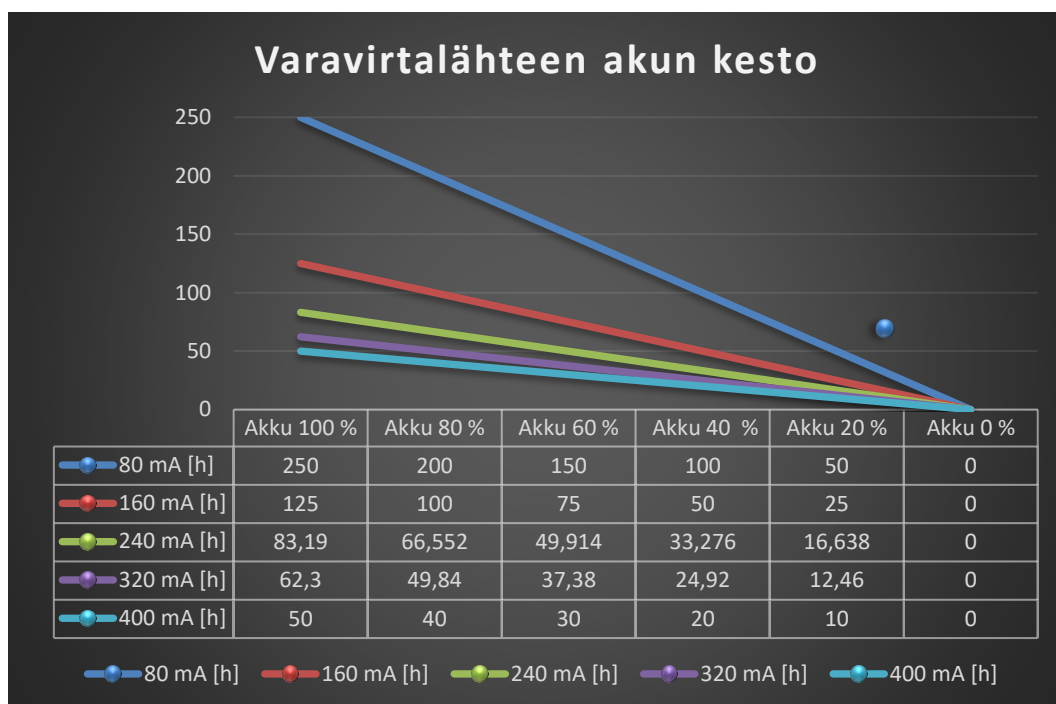
$$Akun\ kesto\ 80\ mA\ kulutus = \frac{20\ 000\ mAh}{80\ mA} = 250\ h$$

$$Akun\ kesto\ 160\ mA\ kulutus = \frac{20\ 000\ mAh}{160\ mA} = 125\ h$$

$$Akun\ kesto\ 240\ mA\ kulutus = \frac{20\ 000\ mAh}{240\ mA} = 83\ h\ 19\ min$$

$$Akun\ kesto\ 320\ mA\ kulutus = \frac{20\ 000\ mAh}{320\ mA} = 62\ h\ 30\ min$$

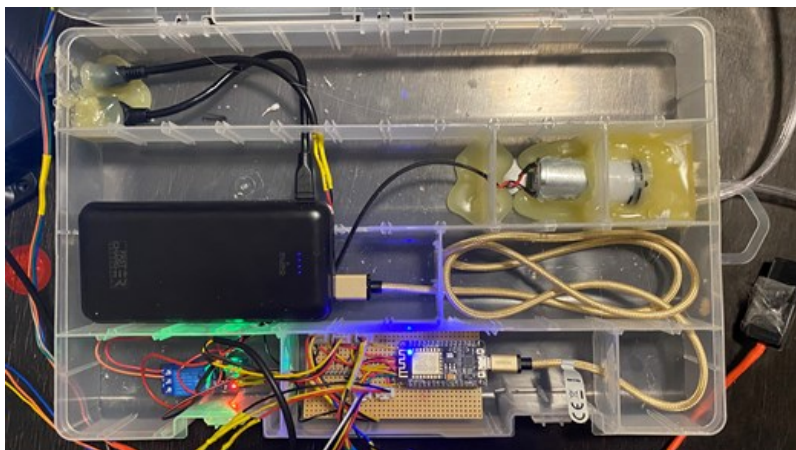
$$Akun\ kesto\ 400\ mA\ kulutus = \frac{20\ 000\ mAh}{400\ mA} = 50\ h$$



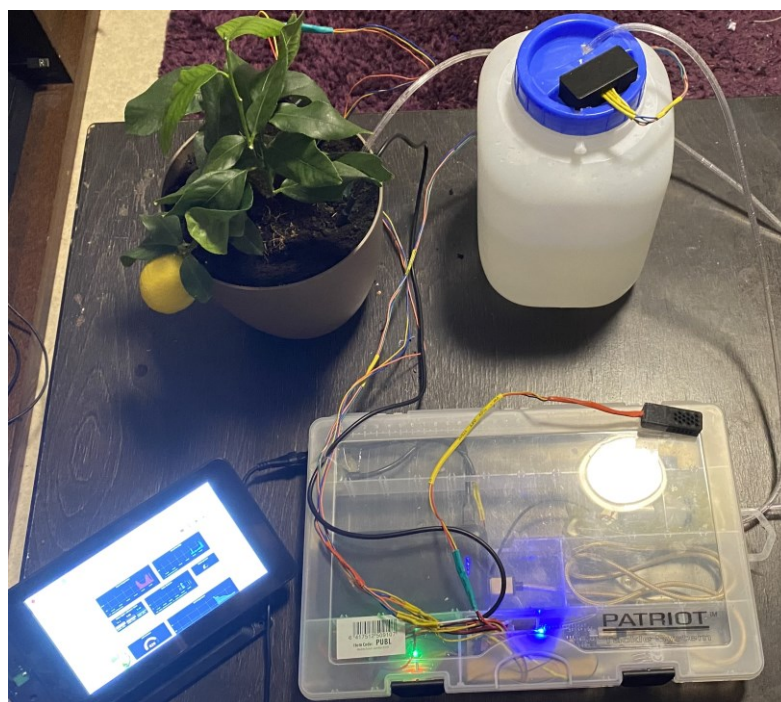
Varavirtalähdettä testattiin myös itse käytössä. Kastelujärjestelmä toimi akun avulla itsenäisesti testin mukaan 6 päivää. Tämän perusteella virran kulutus on alle 160mA.

6.4 Järjestelmän kotelointi

Kuvassa 41 on kastelujärjestelmä rakennettu muovisen kotelon sisälle. Kytkentälevy saa virtansa akkupankista. Kotelon vasemmassa reunassa on latausportti akkupankille sekä virtajohto vesipumpulle. Vesipumppu on vuorattuna, jotta mahdollisilta vesivuodolta vältyttäisiin. Sähköhän ei tunnetusti oikein veden kanssa ole mieluinen yhteistyökumppani. Kuvassa 42 on kuvattuna kastelujärjestelmän toimivuus.



Kuva 41. Järjestelmä kotelossa.



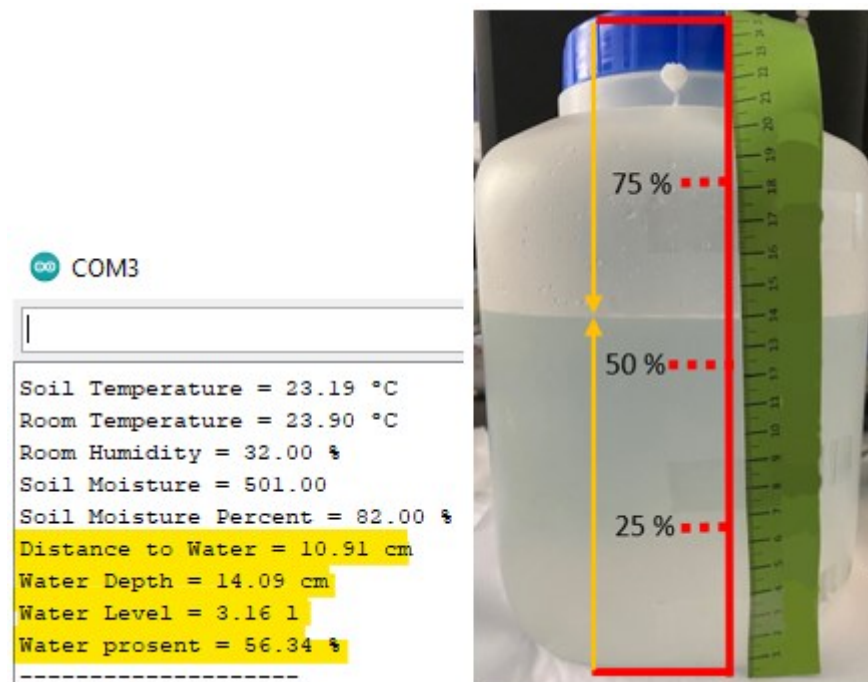
Kuva 42. Kastelujärjestelmä toiminnassa.

7 TESTAUS

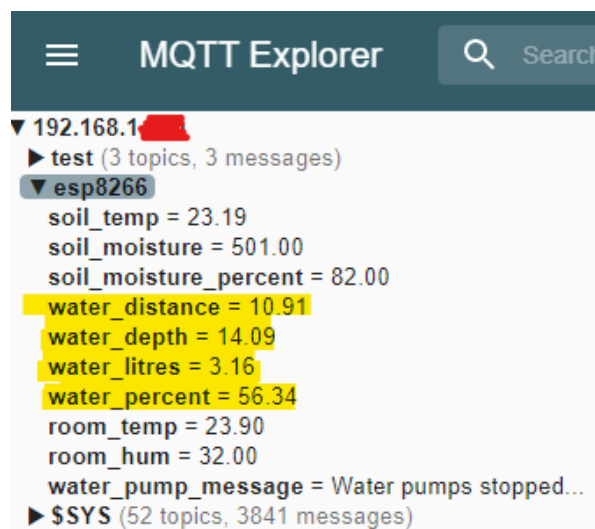
Kastelujärjestelmän luotettavan toimivuuden kannalta oli tärkeää testata jokaisen anturin toimivuus eri tilanteissa sekä anturidatan lähettäminen MQTT-välittäjälle onnistuneesti sekä datan siirtyminen tietokantaan. Tällä tavoin pystyttiin helposti todistamaan järjestelmän toimivuus ja lisäksi löytämään järjestelmän viat, mikäli sellaisia ilmaantui.

HC-SR04-ultraäänianturilla etäisyyden mittaamista vedenpinnasta oli aluksi testattava mittaamalla anturin lukema tyhjästä vesiastiasta. Vesiastian maksimikorkeudeksi mitattiin 25 cm, joka oli myös valmistajan ilmoittama korkeus. Kuvassa 43 on ilmoitettu etäisyysanturin mittaama tulos vedenpinnasta 10,91 cm, joka oli lähes sama kuin vieressä olevassa vesiastiassa, jos katsoo silmämääräisesti mittanauhan tulosta. Maksimikorkeus oli jo tiedossa, joten oletetusti kun maksimikorkeudesta vähennetään mitattu etäisyys, saataisiin selville veden syvyys, joka on anturin mukaan 14,09 cm. Tämä voidaan varmentaa vähentämällä 25,00 cm:stä 10,91 cm, jonka tulos on 14,09 cm. Tämän perusteella voidaan todeta anturin toimivan oikein.

Tietokoneelle oli asennettuna MQTT Explorer-ohjelma testaamista varten, jolla mahdollistettiin antureiden toimivuus helposti. Tämän ansiosta ei tarvinnut käyttää Raspberry Pi:n terminaalia eikä tarvinnut antaa erikseen komentoja vaan kaikki näkyivät suoraan kerralla. MQTT Explorer on siis monipuolinen ohjelma, joka tekee jäsenneilyn yleiskatsauksen MQTT-aiheista. Lisäksi ohjelman avulla työskentelestä välittäjän antureiden kanssa tulee helppoa. (MQTT Explorer, 2021.). Kuvassa 44 voi nähdä samat tulokset kuin COM-portista mitattuna.



Kuva 43. Veden määrän mittaaminen.



Kuva 44. Veden määrän tulokset MQTT-välittäjässä.

Kuvassa 45 on kuvattuna kapasitiivisen maakosteusanturin toimivuuden testaaminen, jossa samalla testattiin myös vesipumpun toimivuus. Oli aluksi saatava selville anturin lukema kuivimmillaan sekä kosteimmillaan kalibrointia varten. Anturin tulokset kuivimmillaan tuli 860 ja kosteimmillaan 421. Arvot voivat siis vaihdella anturista riippuen. Anturihan ei mittaa prosentteja, joten olen arvioinut, että jos 860 on kuivin niin sen on oltava 0 % ja kosteimmillaan 421 eli 100 %. Muut prosenttiarvot tulevat 860–421 väliltä. Vesipumppu on ohjelmoitu niin että pumppu menee päälle, jos maankosteus on alle 10 % ja sammuu kun maankosteus saavuttaa 85 %. Jos kuitenkin käy niin, että vesipumppu ei sammuu anturin lukemasta huolimatta, on varatoimeksi ohjelmoitu aikaraja, jonka ylittäessä vesipumppu sammuu.



Kuva 45. Maakosteuden mittaaminen.

Kuvassa 46 on kuvattuna maakosteusanturin tulokset sekä vesipumpun toimivuus COM-portista mitattuna. Kuvasta voi päätellä anturien lukemat kuivana, kosteana sekä märkänä. Samalla tuloksista näkyy vesipumpun sammuneen, kun maakosteus on tarpeeksi suuri.

| COM3 | COM3 | COM3 |
|--|--|--|
| Water Depth = 23.10 cm Water Level = 5.17 l Water present = 92.38 % ----- Soil Temperature = 23.19 °C Room Temperature = 23.90 °C Room Humidity = 31.00 % Soil Moisture = 817.00 Soil Moisture Percent = 10.00 % Distance to Water = 1.90 cm Water Depth = 23.10 cm Water Level = 5.17 l Water present = 92.38 % ----- Soil Temperature = 23.19 °C | Water present = 92.38 % ----- Soil Temperature = 23.13 °C Room Temperature = 23.90 °C Room Humidity = 31.00 % Soil Moisture = 630.00 Soil Moisture Percent = 52.00 % Distance to Water = 1.90 cm Water Depth = 23.10 cm Water Level = 5.17 l Water present = 92.38 % ----- Soil Temperature = 23.19 °C Room Temperature = 23.80 °C Room Humidity = 31.00 % | Water Level = 5.17 l Water present = 92.38 % ----- Soil Temperature = 23.13 °C Room Temperature = 23.80 °C Room Humidity = 31.00 % Soil Moisture = 448.00 Soil Moisture Percent = 94.00 % Water pumps stopped... Distance to Water = 1.90 cm Water Depth = 23.10 cm Water Level = 5.17 l Water present = 92.38 % ----- Soil Temperature = 23.19 °C |
| Kuiva | Kostea | Märkä |

Kuva 46. Maan kosteuden tulokset COM-portissa.

Kuvassa 47 on kuvattuna samat mitatut tulokset MQTT-välittäjästä. Välittäjästä näkee, että vesipumppu on päällä, kun maa on kuiva ja kostea kun taas märkänä ollessa vesipumppu on sammunut.

| MQTT Explorer | MQTT Explorer | MQTT Explorer |
|---|---|---|
| ▼ 192.168.1.1 ▶ test (3 topics, 3 messages) ▼ esp8266 soil_temp = 23.13 soil_moisture = 817.00 soil_moisture_percent = 10.00 water_distance = 1.90 water_depth = 23.10 water_litres = 5.17 water_percent = 92.38 room_temp = 23.90 room_hum = 31.00 water_pump_message = Water pumps running... ▶ \$SYS (52 topics, 1608 messages) | ▼ 192.168.1.1 ▶ test (3 topics, 3 messages) ▼ esp8266 soil_temp = 23.13 soil_moisture = 630.00 soil_moisture_percent = 52.00 water_distance = 1.90 water_depth = 23.10 water_litres = 5.17 water_percent = 92.38 room_temp = 23.90 room_hum = 31.00 water_pump_message = Water pumps running... ▶ \$SYS (52 topics, 2607 messages) | ▼ 192.168.1.1 ▶ test (3 topics, 3 messages) ▼ esp8266 soil_temp = 23.13 soil_moisture = 448.00 soil_moisture_percent = 94.00 water_distance = 1.90 water_depth = 23.10 water_litres = 5.17 water_percent = 92.38 room_temp = 23.80 room_hum = 31.00 water_pump_message = Water pumps stopped... ▶ \$SYS (52 topics, 2923 messages) |
| Kuiva | Kostea | Märkä |

Kuva 47. Maan kosteuden tulokset MQTT-välittäjässä.

8 JATKOKEHITYS

Jatkokehityksen kannalta löytyy paljon kehitysideoita mitkä olisi hyvä toteuttaa. Kehitysideat on jaettu kahteen eri kategoriaan, huonekasveihin ja kasvihuonekasveihin.

8.1 Huonekasvit

Tällä hetkellä kapasitiivinen maakosteusanturi on ohjelmoitu lukemaan maan kosteutta tiettyjen arvojen väliltä esimerkiksi, jos maa on alle 10 % kostea vesipumppu käynnistyy ja sammuu kun maan kosteus on 85 %. Olisi hyvä, jos jatkossa olisi kasveille oma tietokanta, jossa olisi erilaisia kasveja ja jokaiselle olisi määritetty oma kastelumäärä ja aikataulu, kuinka usein kasvia pitäisi kastella. Eli käyttäjä voisi jatkossa valita listalta oikean kasvilajin, jonka jälkeen automaatio hoitaa automaattisesti kastelun.

Kasvien hoitoon olisi jokaiselle kasville tehty itsesuunniteltu kukkaruukku. Kukkaruukun sisäpuolen yläreunaan olisi tehtynä vesijohdolle oma paikkansa. Vesijohdossa olisi pieniä reikiä, jolloin pumpattu vesi pääsisi kasville. Lisäksi järjestelmän elektroniikka olisi suunniteltava kompaktikokoon, joka mahtuisi siististi kukkaruukun alle.

8.2 Kasvihuonekasvit

Kotipihoille tarkoitetuille kasvihuoneille voisi kehittää oman kastelujärjestelmän, joka hoitaa kasvin kastelun automaattisesti tai vaikka millä tahansa älylaitteella napin painalluksella.

Kasvihuoneen sisällä olisi normaalisti kasveille tarkoitetut ruukut rivissä. Ruukkujen yläpuolelle pitäisi tehdä kiskot, jonne vedettäisiin vesijohtoa ja jokaisen kasvin

kohdalle laitettaisiin oma sprinkleri minkä avulla kasvi saisi vettä. Jokaiselle kasville tulisi oma kapasitiivinen maakosteusanturi sekä pH-anturi. Lisäksi kasvihuoneen sisälle pitäisi laittaa palo-/hätä-/savuvaroitin, joka ilmoittaa omistajalle, jos on esimerkiksi savua kasvihuoneessa. Tämä varatoimenpide sen varalta, jos vaikka omistaja on pois kotoa tai mökiltä.

Kasvihuoneen ulkopuolella olisi tehtynä sadevedelle keräys, jota voitaisiin hyödyntää kastelussa. Kasvihuoneen katolle olisi laitettu rännit ja johdettu vesi suureen vesiastiaan mikä olisi kaivettu osittain maan sisälle. Vesipumpun pitäisi olla yhteydessä vesiastiaan sekä kasvihuoneen sisälle olevaan vesijohtoon. Virran saamiseksi vesipumpulle on kaivettava sähköjohto kasvihuoneen viereen.

9 JOHTOPÄÄTÖKSET

Tämä projekti on ollut minulle pitkä ja valtava oppimiskokemus, työ vaati paljon yritystä ja lukuisia määriä virheitä ja ei tietenkään pidä pois sulkea turhautumista, jota tuli myös matkan varrella. Mutta loppujen lopuksi automatisoitu kastelujärjestelmä toimi vaaditulla tavalla.

Ongelmina tuli aluksi vastaan miten maakosteusanturi ja vesipumppu saadaan toimimaan yhdessä, jotta anturi osaisi kertoa milloin vesipumpun olisi mentävä päälle. Ongelman syyksi paljastui rele sekä pienimuotoinen virhe koodissa.

HC-SR04-ultraäänianturin kohdalla suurimpia ongelmia oli sen virheellisyys vedenmittaamiseen koska anturi toimii taajuuspulssina eli virheellisyys tulee siitä, että anturi on vesiastian yläpuolella ja kun taajuuspulssi liikkuu äänen tavoin, on mahdollista, että taajuuspulssi osuu vesiastian reunoihin ensin eikä veteen, jolloin se palaa takaisin anturille mikä johtaa virheelliseen lukemaan koska anturin pulssi ei missään vaiheessa osunut vedenpintaan.

Grafanan datan lähettäminen WordPress-sivuille ei ollut mitenkään helppoa. Grafanassa oli jokaiselle anturille oma graafinen taulukko ja taulukoissa oli oma linkkinsä. Linkin lisääminen suoraan WordPress-sivuille ei toiminut sillä taulukko ei päivittynyt ollenkaan. Ongelman syyksi paljastui, että Grafanan omista asetuksista oli erikseen sallittava taulukon linkittäminen.

LÄHTEET

AutomationWiki, 2018. Ultrasonic Level Measurement. Viitattu 1.1.2022 [http://automationwiki.com/index.php/Ultrasonic Level Measurement](http://automationwiki.com/index.php/Ultrasonic_Level_Measurement)

Circuit Schools, 2022 a. Interface Capacitive Soil moisture sensor. Viitattu 22.4.2022 <https://www.circuitschools.com/interface-capacitive-soil-moisture-sensor-v1-2-with-arduino-lcd-and-oled/>

Circuit Schools, 2022 b. Automatic Plant watering system using Arduino. Viitattu 22.4.2022 <https://www.circuitschools.com/automatic-plant-watering-system-using-arduino-or-esp8266/>

Elecfreaks, 2014. HC-SR04 User Guide. Viitattu 26.10.2021 <https://datasheetspdf.com/pdf-file/909919/ELECFreaks/HC-SR04/1>

Elecrow, 2019 a. DS18B20 Programmable Resolution 1-Wire Digital Thermometer Viitattu 26.10.2021 <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Temp/DS18B20.pdf>

Elecrow, 2019 b. DHT11. Viitattu 28.11.2021 <https://www.elecrow.com/wiki/index.php?title=DHT11>

ElectronicWings, 2021. Ultrasonic Module HC-SR04. Viitattu 1.1.2022 <https://www.electronicwings.com/sensors-modules/ultrasonic-module-hc-sr04>

Grafana Labs, 2021. Why Grafana. Viitattu 10.1.2022 <https://grafana.com/grafana/>

How To Mechatronics, 2022. Ultrasonic Sensor HC-SR04 and Arduino. Viitattu 22.4.2022 <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

Influxdata, 2021. Act in Time Build on InfluxDB. Viitattu 1.12.2021 <https://www.influxdata.com/>

Instructables circuits, 2022 a. Arduino Plant Monitor. Viitattu 22.4.2022 <https://www.instructables.com/Arduino-Plant-Monitor/>

Instructables circuits, 2022 b. Measuring Water Level With Ultrasonic Sensor. Viitattu 22.4.2022 <https://www.instructables.com/Measuring-water-level-with-ultrasonic-sensor/>

Mosquitto, 2021. Eclipse Mosquitto. Viitattu 28.11.2021 <https://mosquitto.org/>

MQTT Explorer, 2021. An all-round MQTT client that provides a structured topic overview Viitattu 26.2.2022 <http://mqtt-explorer.com/>

MQTT, 2020. MQTT: The Standard for IoT Messaging. Viitattu 28.11.2021 <https://mqtt.org/>

NodeMCU Documentation, 2021. NodeMCU Documentation. Viitattu 10.1.2022 <https://nodemcu.readthedocs.io/en/release/>

Node-RED, 2021. About Viitattu 30.11.2021 <https://nodered.org/about/>

Random nerd tutorials, 2021. ESP8266 NodeMCU with HC-SR04 Ultrasonic Sensor with Arduino IDE. Viitattu 1.1.2022 <https://randomnerdtutorials.com/esp8266-nodemcu-hc-sr04-ultrasonic-arduino/>

Random nerd tutorials, 2022. ESP8266 NodeMCU MQTT. Viitattu 22.4.2022 <https://randomnerdtutorials.com/esp8266-nodemcu-mqtt-publish-bme280-arduino/>

Raspberry Pi Foundation, 2021. What is a Raspberry Pi. Viitattu 10.1.2022 <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>

SC& SCAND, 2021. Grafana Everything You Need to Know. Viitattu 10.1.2022 <https://scand.com/company/blog/scand-commits-grafana/>

Steve's Internet Guide, 2022. Understanding MQTT Topics. Viitattu 22.4.2022 <http://www.steves-internet-guide.com/understanding-mqtt-topics/>

SuperHouse, 2019. #41: Dataloggin whit MQTT, Node-RED, InfluxDB, and Grafana. Viitattu 22.4.2022 <https://www.superhouse.tv/41-datalogging-with-mqtt-node-red-influxdb-and-grafana/>

SwitchDoc Labs, 2020. Tutorial Using Capacitive Soil Moisture Sensors on the Raspberry Pi. Viitattu 2.1.2022 <https://www.switchdoc.com/2020/06/tutorial-capacitive-moisture-sensor-grove/>

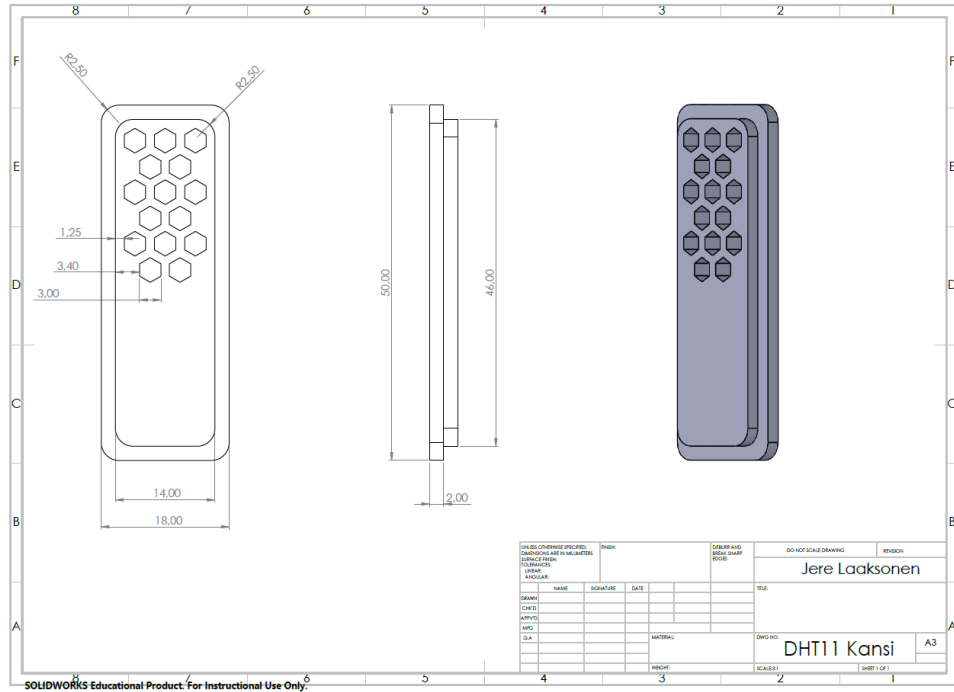
Techtutorialsx, 2022. ESP8266: Connecting to MQTT broker. Viitattu 22.4.2022 <https://techtutorialsx.com/2017/04/09/esp8266-connecting-to-mqtt-broker/>

WordPress.org, 2021. Our Mission. Viitattu 10.1.2022 <https://wordpress.org/about/>

LIITTEET

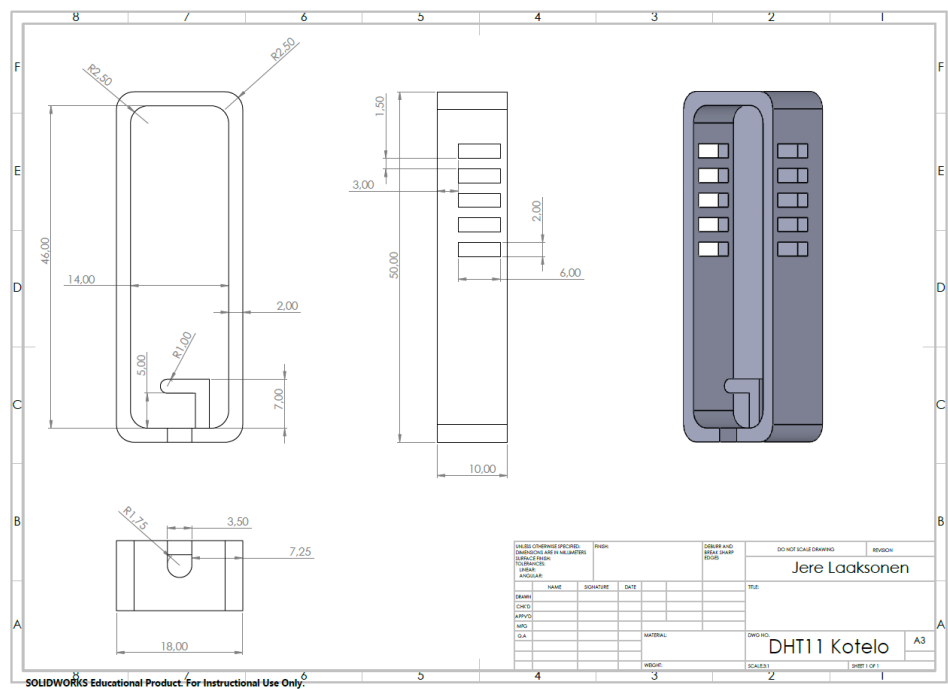
LIITE 1

3D-suunniteltu DHT11-lämpötila-anturin kansi.



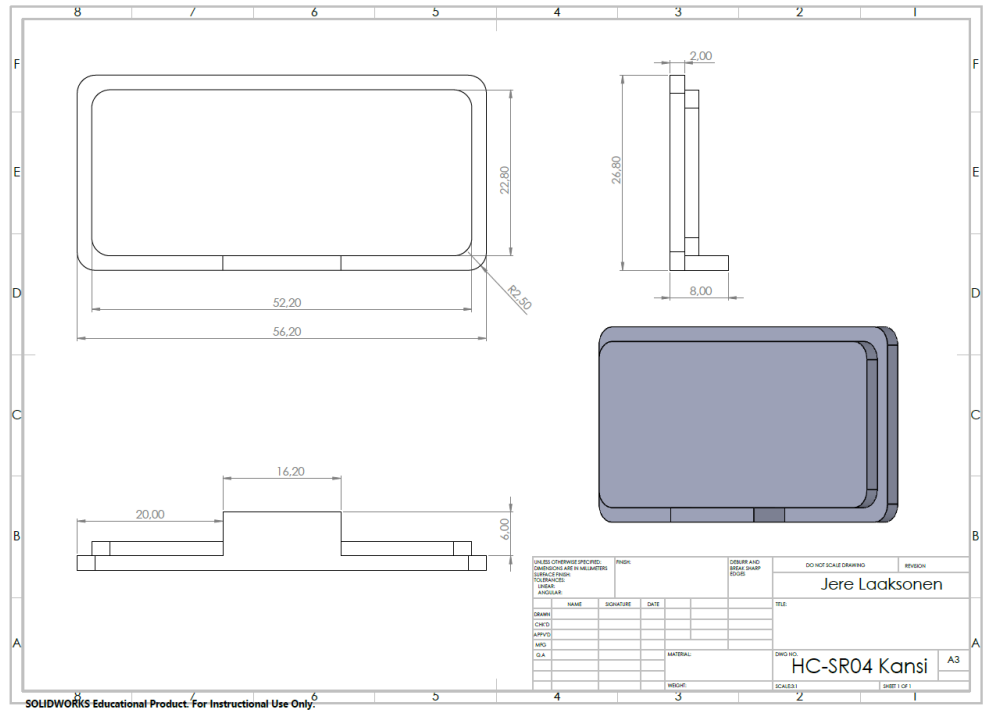
LIITE 2

3D-suunniteltu DHT11-lämpötila-anturin kotelo.



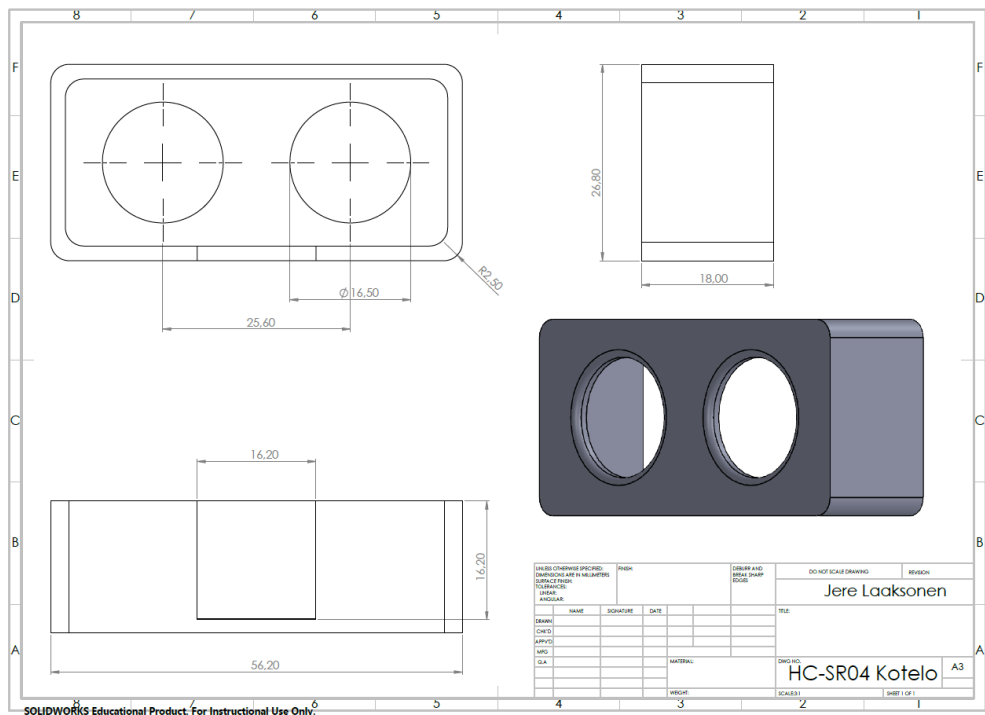
LIITE 3

3D-suunniteltu HC-SR04-etäisyyssanturin kansi.



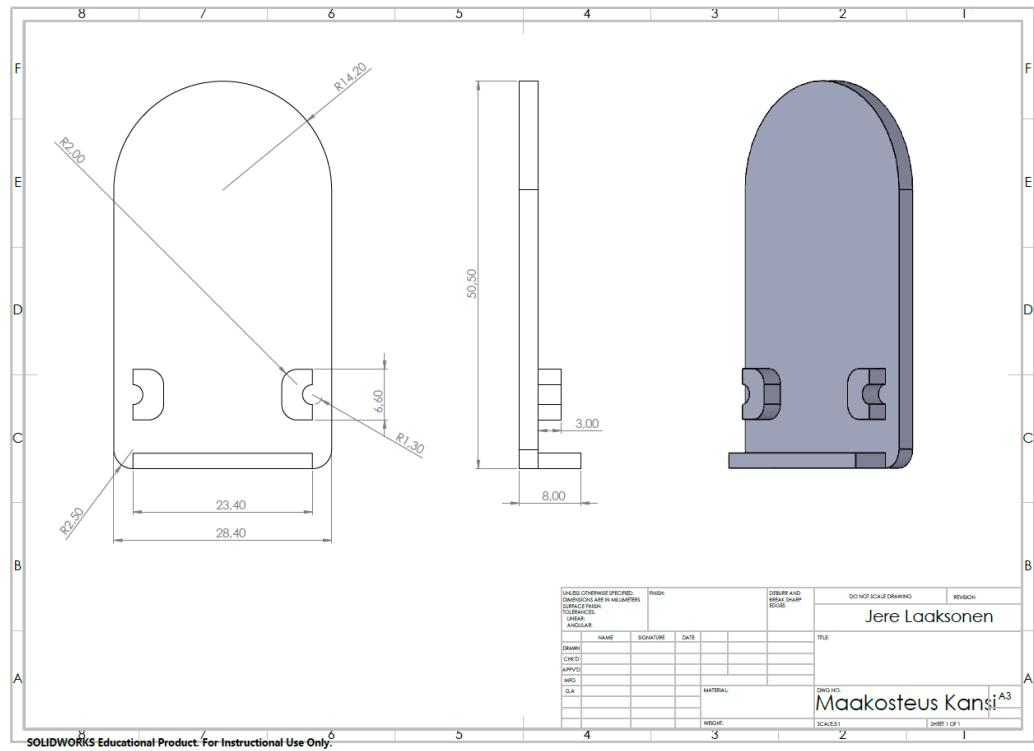
LIITE 4

3D-suunniteltu HC-SR04-etäisyyssanturin kotelo.



LIITE 5

3D-suunniteltu maakosteusanturin kansi



LIITE 6

3D-suunniteltu maakosteusanturin kotelo.

