

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutus

Matti-Pekka Apilo

RAVINNETASAPAINON JA KASVATUSOLOSUHTEIDEN
HALLINTA VESIKASVATUKSESSA KÄYTTÄEN IOT-RATKAISUJA

Opinnäytetyö
Toukokuu 2021



OPINNÄYTETYÖ
Toukokuu 2021
Tietojenkäsittelyn koulutus

Tikkarinne 9
80200 JOENSUU
+358 13 260 600

Tekijä(t)
Matti-Pekka Apilo

Nimeke
Ravinnetasapainon ja kasvatusolosuhteiden hallinta vesikasvatuksessa käyttäen IoT-ratkaisuja

Toimeksiantaja
Karelia-amk

Tiivistelmä

Opinnäytetyössä käsiteltiin vesikasvatuksen perusteita ja niiden soveltamista toteutuksessa, jossa rakennettiin vesikasvatukseen soveltuva mittari Arduino-alustalle. Opinnäytetyön tarkoituksena oli rakentaa Arduinon ympärille vesikasvatuksen hallintaan soveltuva IoT-laite, jolla pystyttiin mittaamaan sekä keräämään dataa vesikasvatukselle tärkeistä muuttujista. Datan keräämisen kannalta oli myös tärkeää, että dataa pystyttiin keräämään ja lähettämään laitteen ulkopuoliseen tietokantaan.

Laite toteutettiin käyttämällä Arduino Uno -alustaa, johon liitetyillä sensoreilla ja laitteilla se saatiin keräämään ja lähettämään dataa verkkopalvelimelle. Tiedon lähettäminen suoritettiin http POST-pyyntöillä ja verkossa sijaitsevalla PHP-tiedostolla, joka käsitteli tiedon ja lähetti sen SQL-kyselynä MySQL-tietokantaan.

Opinnäytetyön tuloksena oli tavoitteiden ja toimeksiannon mukainen laite. Laite pystyi mittaamaan ravinneliuoksen lämpötilaa, sähkönjohtavuutta sekä liukoisuutta tarvittavat muuttujat huomioiden, reagoimaan ravinnetasapainon muutoksiin sekä lähettämään niitä verkossa sijaitsevaan tietokantaan.

Kieli
Suomi

Sivuja 45
Liitteet 0
Liitesivumäärä 0

Asiasanat

hydroviljely, arduino, esineiden internet, tietokannat



THESIS
May 2021
Computing

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600

Author (s)
Matti-Pekka Apilo

Title
Management of Nutrient Balance and Growing Conditions in Hydroponics Using IoT Solutions

Commissioned by
Karelia UAS

Abstract

The thesis dealt with the principles of hydroponic growing and their application in the implementation, where a measuring device suitable for hydroculture was built on the Arduino development platform. The purpose of the thesis was to build an IoT device around Arduino suitable for nutrient solution management to measure and collect data and variables important for hydroponic growing. It was also important for data collection that the data could be collected and sent to an external online database.

The device was implemented using an Arduino Uno platform, with sensors and devices connected to it to collect and send data to a web server. The data was sent as an http POST request and an online PHP file that processed the data and sent it as an SQL query to a MySQL database.

The result of the thesis is a device that meets the goals and the commissioner's requirements for the assignment. The device is able to measure the temperature, electrical conductivity and solubility of the nutrient solution while taking into account the necessary variables. The device is also able to react to changes in the nutrient balance and send data to an online database.

Language

Finnish

Pages 45

Appendices 0

Pages of Appendices 0

Keywords

Hydroponics, Arduino, Internet of things, Databases

Sisältö

Lyhenteet	5
1 Johdanto	6
2 Vesiviljely eli hydroponiikka	7
2.2 Hydroponiikan edut	8
2.3 Hydroponiikan haasteita	12
2.4 Ravinneliuoksen pitoisuuksien mittaaminen sähkönjohtavuuden avulla 14	
3 Hydroponiikan käyttö kaupallisessa kasvatuksessa	15
4 Opinnäytetyön toteutus	16
5 Arduino	17
6 Sensorit ja käytetyt lisälaitteet.....	19
6.1 Liukoisuussensori	19
6.2 Lämpösensori	20
6.3 LCD-näyttö.....	21
6.4 ESP-01 WiFi-moduuli	21
6.5 Muistikortinlukija	22
6.6 Muut komponentit ja tarvikkeet	23
7 Laitteen kokoonpano ja kytkennät	25
8 Arduinin ohjelmointi	27
8.1 Arduinin ohjelmointikieli ja ympäristö	27
8.2 Ohjelmakoodi ja sen rakenne	28
9 PHP	34
9.1 Tiedonsiirto PHP:llä	34
9.2 Sensoridatan visualisointi PHP:llä	35
10 MySQL-tietokanta	36
10.1 Tietokanta	36
10.2 Tietokannan taulut	36
11 Tulokset ja yhteenveto	38
12 Pohdinta.....	39
12.1 Tavoitteet ja oppimistulokset.....	39
12.2 Kohdattuja ongelmia opinnäytetyössä	40
12.3 Mihin opinnäytetyön toteutusta voidaan käyttää	41
Lähteet.....	43

Lyhenteet

- PHP Hypertext Preprocessor. Verkkokehitykseen tarkoitettu ohjelmointikieli, joka mahdollistaa www-sivujen dynaamisen muokkauksen palvelimella (PHP 2021).
- HTML Hypertext Markup Language. Verkkosivujen ja sovelluksien rakennukseen ja muotoiluun tarkoitettu merkkikieli (Damantas 2019).
- HTTP Hypertext Transfer Protocol. Selainten ja palvelimien käyttämä tiedonsiirtoprotokolla, jota käytetään tiedonsiirtoon ja kommunikaatioon (W3Schools 2021).
- SQL Structured Query Language. Tietokantojen kanssa kommunikointiin tarkoitettu kieli, jota käytetään tiedon tallentamiseen, muokkaamiseen ja hakemiseen tietokannoista (SQLCourse 2021).

1 Johdanto

Opinnäytetyöni aiheena on ”Ravintetasapainon ja kasvatusolosuhteiden hallinta vesikasvatuksessa käyttäen IoT-ratkaisuja”. Työssä rakensin IoT-laitteen, jolla voidaan hallita vesikasvatusta. Laite kerää dataa vesiliuoksien ravintetasapainosta ja lämmöstä sekä tallentaa kerätyt tiedot pilvipalvelussa sijaitsevaan tietokantaan.

Valitsin opinnäytetyön aiheen koska olin jo suunnitellut vastaavaa projektia henkilökohtaisten vesikasvatusharrastuksieni takia ja koska se oli kiinnostava. Olen harrastanut hydroponista chilien ja muiden vihanneksien kasvattamista muutaman vuoden ajan ja tavoitteenani on optimoida kasvatusastioiden ravinneliuoksen ravintetasapainoa niin, että kasvit saisivat optimaalisen määrän ravinteita. Halusin selvittää, kuinka vesikasvatusjärjestelmiä voidaan hallita käyttäen IoT-ratkaisuja ja olin kiinnostunut rakentamaan jonkinlaisen mittarilaitteen jollekin alustalle, jotta voisin mitata ja seurata ravinneliuoksen koostumusta. Aihealue vastasi mielestäni todella hyvin suurta osaan digitaalisen liiketalouden koulutusohjelman kurssisisällöstä, joten halusin ehdottaa aihetta sovellettavaksi opinnäytetyöhön Karelia ammattikorkeakoululle.

Kasvit saavat hydroponisessa kasvatuksessa ravinteensa ravinneliuksesta, joka koostuu vedestä ja siihen lisätyistä ravinteista. Ravinneliuoksen ravinnepitoisuus laskee kasvien kulutuksen takia, joten liukseen täytyy lisätä ravinteita säännöllisin määräajoin. Tähän asti olin noudattanut minulle annettuja kiinteitä suosituksia lannoitteiden määrästä ja olin lisännyt ravinteita muutamien viikkojen välein manuaalisesti. Ravinteiden manuaalinen lisääminen oli kokemukseni mukaan joissakin tilanteissa johtanut kasveissa selkeisiin ravinteiden puutoksiin. Tämä on johtunut esimerkiksi kasvien nopeasta kasvusta tai lämpötilojen suuresta vaihtelusta, joka on johtanut liuoksen veden liialliseen haihtumiseen tai kasvien kasvaneeseen ravinteiden tarpeeseen.

Opinnäytetyön tavoitteena oli ottaa selvää tarvittavista muuttujista ja määreistä, joita vesikasvatuksen hallinnassa tarvitaan ja soveltaa niitä käytännön

toteutuksessa. Tähän tarvitsin pohjatietoa, kuten kuinka ravinteiden mittaaminen voidaan toteuttaa ja millaisia arvoja siitä voidaan mitata sekä miten tietoa voidaan käyttää tilanteen analysointiin ja korjaamiseen. Lisäksi olin todella kiinnostunut kasvatustieteen keräämisestä, joten laitteen yhdistäminen verkkoon oli tarpeellista.

2 Vesiviljely eli hydroponiikka

2.1 Hydroponiikka kasvatustekniikkana

Vesiviljely eli hydroponiikka on kasvatustekniikka, jossa kasvien kasvaminen tapahtuu mullan sijaan ravinneliuoksessa, joka koostuu vedestä ja siihen lisätyistä ravinteista. Vesiviljelyssä kasvit käyttävät siis välittäjäaineena ravinneliuosta, eivätkä ne täten tarvitse maa-ainesta kasvatusalustaksi. Kasveille voidaan haluttaessa myös lisätä keinotekoisesti tuki- ja välittäjäainemateriaaliksi esimerkiksi kivivillaa tai perliittiä, mutta kasvien ravinteiden saanti ei varsinaisesti tapahdu tästä materiaalista. (Jones 2005, 3–4.)

Vesikasvatukselle on monenlaisia erilaisia tekniikoita, joilla kasvatus voidaan toteuttaa. Eräs näistä tekniikoista on seisovassa ilmatussa vedessä kasvatus (Standing aerated nutrient solution), jossa ravinneliuos on paikallaan olevaa vettä, jota ilmataan esimerkiksi akvaariopumpulla. Toisena yleisenä tekniikkana käytössä on ravinnekalvotekniikka (Nutrient Film Technique, NFT), jossa ravinneliuos ohjataan valumaan hieman kallistetun kanavan pohjaa pitkin niin, että se tuo ravinteita juurille. (Jones 2005, 123–130.)

Jones (2005) esittelee mm. Jensenin (1981) sekä K. Wignarajahin (1995) listaamia etuja ja haittoja hydroponisesta kasvatuksesta. Käyn läpi näitä etuja ja haittoja seuraavissa kappaleissa.

2.2 Hydroponiikan edut

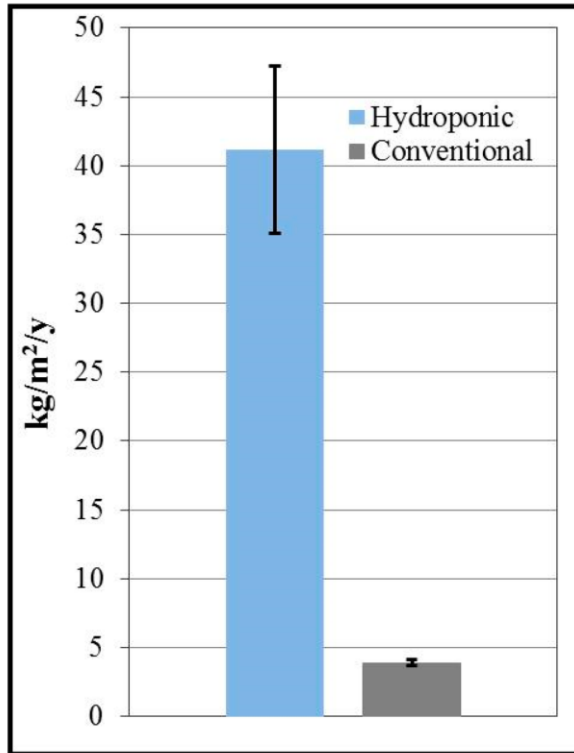
Hydroponiikka tuo teknologiana mukanaan monenlaisia hyötyjä ja mahdollisuuksia. Hydroponiikka mahdollistaa kasvien kasvamisen tarkan hallinnan ja sen monet esiintymismuodot mahdollistavat uusia innovaatioita eri aloilla ja sijainneissa. Hydroponisten ratkaisujen käyttämiseen viljelyssä liittyy todella paljon myös muita etuja. Teknologiana se on kehittyvä ja hyvä vaihtoehto tai ratkaisu moniin ongelmatilanteisiin, joihin voidaan joutua tavallisen maanviljelyn keinoin. (Jones 2005, 4–5.)

Hydroponiikan eräänä etuna on sen paikasta riippumattomuus. Hydroponiikka voidaan ottaa käyttöön muun muassa paikoissa, joissa ei ole saatavilla sopivaa maa-ainesta kasvatusalustaksi, vedensaanti on niukka tai joissa maaperä on saastunut esimerkiksi maaperässä leviävien tautien tai loisten, kuten sukkulamatojen takia. Maaperäisten kasvitautien leviäminen vesikasvatusjärjestelmässä on hallitumpaa, sillä niiden estäminen on helpompaa ja ne voidaan hävittää suoraan lisäämällä ravinneliuokseen niitä tuhoavia lisäaineita järjestelmään. (Jones 2005, 4–5.)

Hydroponiikka etuineen mahdollistaa myös kasvatuksen esimerkiksi kaupungeissa. Kasvatus on tekniikan ansiosta mahdollista myös asunnoissa tai muissa toimitiloissa, kuten kerrostaloissa, joissa perinteisiä kasvatusmenetelmiä ei voida käyttää tehokkaasti (Jones 2005, 4–5). Kaupunkikasvatuksen etuja ovat muun muassa kasvatustilan optimointi, toimitusten ja pakkauskulujen pienentäminen sekä paikallisen ruoan tuottaminen ympärivuotisesti. Kaupungeissa tapahtuva kaupallinen kasvattaminen hydroponisesti onkin eräs tapa, jolla tulevaisuudessa tullaan auttamaan kasvavan väestön ruoan tarvetta. Kasvattaminen kaupungeissa vähentää pakkaamisen ja kuljetuksien aiheuttamaa ekologista painetta ja parantaa kaupunkien ruoansaantia, sillä ruokaa voidaan tuottaa myös paikallisesti kaupunkien sisällä. Hallittujen kasvatusjärjestelmien käyttö auttaa myös esimerkiksi hankalasti saatavien kasvilajikkeiden kasvatuksessa ravintolakäyttöön, jolloin niitä ei tarvitse maahantuoda ulkomailta. (Venture City 2019.)

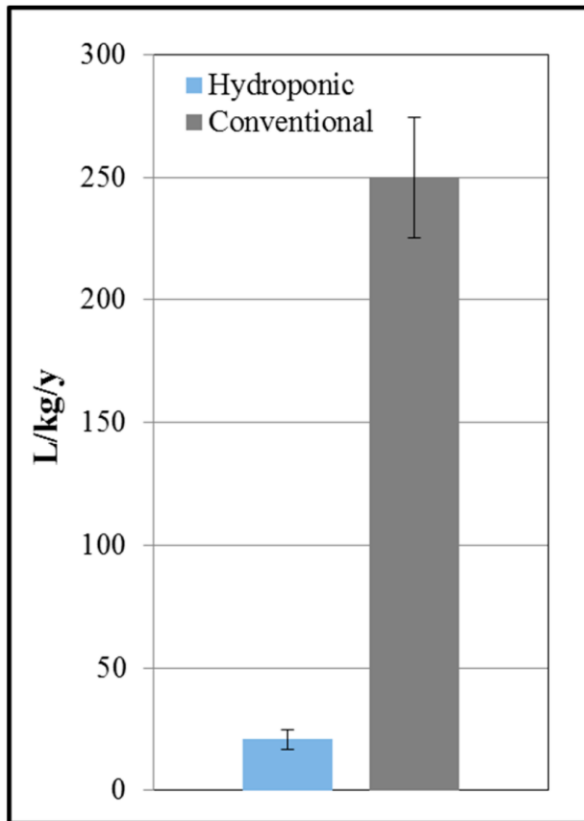
Toisena etuna hydroponisella kasvatuksella on fyysisen työn väheneminen. Fyysisen maan muokkaamisen, kuten kyntämisen, viljelyn, tuholistorjunnan ja kastelun sekä muiden kasvatukseen liittyvien prosessien työn määrä vähentyy huomattavasti, sillä oikein hallittuna järjestelmä on todella tehokas ja suhteessa työmäärällisesti kevyempi kuin monet muut kasvatusmenetelmät. (Jones 2005.) Maantyöstämisen ja perinteisten kasvatuksen vaatimien valmisteluiden puuttuminen mahdollistavat myös jatkuvan kasvatusketjun. Perinteisessä maakasvatuksessa sadonkorjuun ja uuden kasvin kasvatuksen välissä voi usein olla viikkojen katko. Katko johtuu vanhojen kasvien poistamisen, maan muokkaamisen, lannoittamisen sekä muiden työvaiheiden vaatimasta työajasta. Vesikasvatuksessa kasvien vaihtaminen uusiin kasveihin vie multakasvatukseen verrattuna huomattavasti vähemmän aikaa, sillä kasvit on helppo poistaa järjestelmästä ja uudet kasvit voidaan lisätä heti vanhojen kasvien poistamisen jälkeen. (Venture City 2019.)

Tehokkuus näkyy muun muassa hydroponiikan mahdollistamassa optimoidussa tuotannossa. Tuotannon optimointi mahdollistaa tuotannon pienemmillä pinta-aloilla ja mahdollistaa täten tuotannon kalliimmilla alueilla kasvien tuottavuuden ansiosta. (Jones 2005, 4–5). Esimerkkinä tästä on International Journal of Environmental Research and Public Healthin artikkelin vertailu perinteisen ja hydroponisen kasvatuksen vuosittaisen tuotannon välillä kiloissa neliometriä kohden. Hydroponisella kasvatusmenetelmällä tuotetun kausittaisen tuotannon määrä oli jopa 11-kertainen verrattuna perinteisellä kasvatusmenetelmällä tuotettuun määrään salaattia (kuvio 1). (Barbosa ym. 2015.)



Kuvio 1. Vuotuinen salaatin tuotanto hydroponisella ja perinteisellä kasvatusmenetelmällä Lounais-Arizonassa (Barbosa ym. 2015).

Tuotannon tehokkuus johtuu muun muassa järjestelmien paremmasta kontrollista, joka mahdollistaa kasvien tarpeiden tarkemman huomioimisen ja paremman hallinnan ravinnetasapainosta. Tämän ansiosta ravinneliuoksien pitoisuudet ovat myös multakasvatusta pienempiä ja ravinteita ei mene niin paljoa hukkaan. (Jones 2005, 4–5.) Ravinteiden saanti vesikasvatusympäristössä on myös parempi verrattuna multakasvatukseen, sillä ravinteiden saatavuus ja niiden käyttö vesikasvatuksessa olevilla kasveilla on tehokkaampaa. Myös veden kulutus on vähäisempää, sillä kasvien kasteluun ei tarvita yhtä paljon vettä. Vettä ei myöskään poistu järjestelmästä yhtä paljon kuin perinteisessä kasvatuksessa, sillä vesi on hallitussa suljetussa ympäristössä. (Jones 2005, 4–5.) International Journal of Environmental Research and Public Healthin artikkelin mukaan vedenkulutus oli esimerkiksi vesikasvatetussa salaatissa keskimäärin 13 kertaa pienempi kuin perinteisillä kasvatusmenetelmillä kasvatetussa salaatissa (kuvio 2) (Barbosa ym. 2015).



Kuvio 2. Vuotuinen vedenkulutus salaattikiloa kohden hydroponisella ja perinteisellä kasvatustavalla Lounais-Arizonassa (Barbosa ym.2015).

Myös australialaisen ABC Catalystin tekemän dokumentin mukaan perinteisillä tekniikoilla kasvatetun tomaatin kasvatukseen pelloilla voi kuluttaa jopa 200 litraa vettä tomaattikiloa kohden. Dokumentissa haastatellun australialaisen hydroponisen tomaattien tuottajan mukaan he kuluttavat vain 12 litraa vettä tomaattikiloa kohden. Perinteiseen kasvatukseen verrattuna vettä kuluu 94 % vähemmän. (ABC Catalyst 2020.)

Veden ja ravinteiden hallinta auttaa myös pitämään huolen ympäristöstä sekä vähentämään maan ja vesistöjen saastumista sekä niistä johtuvaa rehevöitymistä, sillä ravinneliuos ei pääse esim. karkaamaan vesistöihin sadeveden tai kastelun aiheuttaman huuhtoutumisen takia. Tämä johtuu siitä, että ravinneliuos on hallitussa ympäristössä ja se voidaan kierrättää hallitusti ja vastuullisesti. Hydroponiikan hallittavuus mahdollistaa myös korkeampien

suolapitoisuuksien käyttämisen kasvatuksessa, mikä oikein hallittuna voi parantaa kasvien tuotantoa. (Jones 2005, 4–5.)

Hydroponiikassa onkin siis todella paljon etuja, jotka tekevät siitä mielenkiintoisen vaihtoehdon perinteiselle maanviljelylle. Mielestäni mielenkiintoisimpana hyötynä vesikasvatuksessa ovat kasvatusalustan mahdollistama kontrolli ravinteiden saannista ja käytöstä sekä niiden tarjoamat optimointimahdollisuudet ja hallinnalliset tekniikat, joita maksimaalisen kasvun saavuttamiseen voidaan hyödyntää. Tämä lieneekin syy, jonka takia olen valinnut tämän aiheen opinnäytetyön aiheeksi. Jotta vesikasvatus olisi tehokasta, se vaatii tietoa ja taitoa ravinneliuoksien koostumuksista, kasvien ravinteellisten tarpeiden ja kulutuksen toiminnasta sekä järjestelmien toiminnasta. Tästä syystä siihen liittyy myös paljon huonoja ja kasvatusteknisesti haastavia puolia, joita käydään läpi seuraavissa kappaleissa.

2.3 Hydroponiikan haasteita

Hydroponisten ratkaisujen käyttäminen ei ole suoraan helppoa, vaan siihen liittyy luonnollisesti myös teknisiä vaikeuksia ja kasvatusteknillisesti haastavia puolia. Hydroponisten ratkaisujen alkuperäinen rakennuskustannus on laajoissa projekteissa suuri ja niiden hinta tavallisiin kasvatusmenetelmiin nähden on merkittävä. (Jones 2005, 4–5.)

Kalifornialainen kasvatusratkaisuja valmistava AMHydro kertoo webinaarissaan ”Cost and Profitability of Hydroponic Farming” kasvatusratkaisujen ja hydroponiikan vaatimista kuluista. Harris kertoo esimerkiksi Nutrient film -tekniikkaa käyttävän järjestelmän maksavan noin 10-12 \$/m² ja sankojärjestelmien maksavan noin 8-10 \$/m². Lisäksi kasvatukseen kuuluu myös monia muita rakennuksiin ja laitteistoihin liittyviä kuluja, kuten esimerkiksi kasvatuksen optimoinnin kannalta tärkeän kasvihuoneen rakentaminen. (Harris 2019.) Hydroponisen kasvatuksen onnistumisen takaamiseksi kasvien

kasvattaminen tapahtuu yleensä hallitussa ympäristössä, kuten kasvihuoneessa. Kasvihuonekasvatuksessa voidaan hallita monia kasvien vaatimia muuttujia ja ympäristö voidaan muokata kasvien vaatimuksien mukaiseksi. Kasvihuoneympäristössä voidaan hallita muun muassa lämpötilaa, valoisuutta ja valon laatua, vedenkulutusta, hiilidioksidipitoisuutta sekä ilman kosteutta. (UCANR 2019.)

Vesikasvatus vaatii myös paljon osaamista, sillä prosessien oikeaoppinen hallinta on vesikasvatuksessa todella merkittävässä osassa. Kasvien kasvamisen ja ravinneteknisten asioiden tunteminen ovat tärkeitä ja alan tuntemus ovat välttämättömiä, minkä takia hydroponisten järjestelmien ylläpitäminen voi olla haastavaa. Tietämystä tarvitaan esimerkiksi siihen, että suurin osa saatavilla olevista kasvilajikkeista, jotka on mukautettu hallittuihin kasvatusolosuhteisiin, tarvitsevat lajikohtaista tuntemusta ja asianmukaista kehitystä, jotta niiden kasvattaminen onnistuu kunnolla.

Lisäksi vesikasvatuksessa olevilla kasveissa tapahtuvat ravinnetasapainon muutokset ovat uskomattoman nopeita, joten kasvatettavat kasvit tarvitsevat päivittäistä seuranta puutostilojen ja yleisen kunnan tarkastamiseksi. Sama ongelma näkyy myös tautien leviämisessä. Mikäli järjestelmään pääsee maaperäinen tauti tai esimerkiksi sukkulamato tartunta voi se leviää myös muihin järjestelmiin todella nopeasti. Leviäminen voi tällöin saastuttaa kerralla koko ketjun, johon ravinneliuos on yhteydessä. (Jones 2005, 4–5.)

Hydroponisten järjestelmien edut tuottavat siis myös kasvatuksen näkökulmasta hankaluuksia viljelyyn verraten kasvatusta esimerkiksi multakasvatukseen. Järjestelmien hallitseminen on hankalaa, mutta sen onnistuessa tulokset ovat selkeästi merkittäviä ja useimmiten parempia kuin multakasvatuksessa. Tämäkin on mielestäni eräs lähtökohta, joka kiinnostaa minua todella paljon, sillä opinnäytetyön tarkoituksena on rakentaa laite, jolla kyseisiä seikkoja voidaan mahdollisesti parantaa.

2.4 Ravinneliuoksen pitoisuuksien mittaaminen sähkönjohtavuuden avulla

Vesikasvatuksen onnistumiseksi kasvien ravinneliuoksen ravinnetasapainoa ja ravinteiden määrää on hallittava tarkasti. Ravinnetasapainoa voidaan hallita monella eri tavalla ja tekniikalla, kuten vaihtamalla ravinneliuos säännöllisin väliajoin tuoreeseen liuokseen, jonka ravinnetaso on halutunlainen tai lisäämällä ravinneliuokseen lisää ravinteita tietyin aikajaksoin. Nämä eivät kuitenkaan ole itsessään optimaalisia ratkaisuja, sillä ravinneliuoksen tarkemmasta tilasta ei saada oikeanlaista ravinnetasapainoa kuvaavaa tietoa, eikä prosessissa päästä käyttämään kasvatusmenetelmän tuomia hallinnallisia etuja kunnolla. (Nemali 2018.)

Eräs tapa, jolla liuoksen ravinnepitoisuutta voidaan mitata, on käyttää hyödyksi ravinteiden lisäämää sähkönjohtavuutta liuoksessa. Puhdas vesi ei itsessään ole hyvä sähkönjohdin, mutta kun siihen lisätään ravinnesuoloja, sen sähkönjohtavuus kasvaa ravinteiden tuomien ionien vaikutuksesta. Sähkönjohtavuutta mitataan siemensseissä, joka on konduktanssin yksikkö. Sähkönjohtavuuden avulla voidaan mitata liuoksen kokonaissähkönjohtavuutta, mutta siitä voidaan tietyillä kaavoilla laskea myös ravinteiden yksittäisiä määriä liuoksessa, kuten typen määrää. Sen laskeminen ei kuitenkaan ole tarkkaa kuin vain tuoreessa liuoksessa kasvien epätasaisen ravinteiden kulutuksen takia. (Nemali 2018.)

Kasvit kuluttavat eri määriä eri ravinneaineita ja muun muassa typen, kaliumin, fosforin ja mangaanin kulutus on huomattavasti suurempi kuin kalsiumin, magnesiumin ja rikin kulutus. Tästä syystä nopeasti kuluvien ravinteiden määrä vähenee nopeammin suhteessa hitaasti kuluvien ravinteiden määrään. Ravinteiden epätasaisen kulutuksen takia ravinnetasapainoa ei voida pitää yllä ainoastaan sähkönjohtavuuden perusteella. Ravinteiden optimaalisen saannin kannalta on kuitenkin yleistä, että sähkönjohtavuutta ylläpidetään tasaisesti lisäämällä ravinneliuokseen lisää ravinteita, jotta ravinteiden saanti kasveille ei katkea. Ravinneliuoksen säätäminen ja täyttäminen tuoreella liuoksella eivät

kuitenkaan tuo pysyvää ratkaisua ravinteiden hallintaan, sillä hitaasti kuluvat lannoiteionit vääristävät pitkän ajan sisällä sähkönjohtavuutta liuoksessa. Tästä syystä liuos kannattaa vaihtaa tuoreeseen liuokseen viimeistään kymmenen vuorokauden välein. (Nemali 2018.)

3 Hydroponiikan käyttö kaupallisessa kasvatuksessa

Hydroponiikan käyttö muiden vastaavien mullattomien kasvatustekniikoiden, kuten aeroponiikan kanssa on kasvavan kiinnostuksen kohteena monestakin syystä. Tekniikoiden eräs käyttötarkoitus on kehittää kestävämpää tulevaisuutta kaupunkeihin, tuomalla lisää kasvatustilaa vastaamaan kasvavan populaation ruoantarvetta. Tekniikoiden tehokkuuden ansiosta, kasvien ja vihannesten kasvatusta onnistuu entistä pienemmillä pinta-aloilla, minkä takia kasvatustekniikoita voidaan käyttää myös kaupungeissa. Esimerkkinä tällaisesta kasvatustekniikasta on Lontoon alla sijaitsevassa, toisen maailmansodan aikaisessa suojassa toimiva *growing underground*, joka kasvattaa lehtivihanneksia, kuten herneenversoja ja sinapin lehtiä kaupalliseen käyttöön ja ravintoloihin ympärivuotisesti käyttäen hydroponista kasvatusteknologiaa. (Venture City 2019.)

Hydroponiikan käyttö on myös osana perinteistä kasvatusta. *Greenhouse Production Newsin* artikkelin mukaan jo vuonna 2012 kasvattajat ovat aloittaneet kasvatuksen monimuotoistamisen lisäämällä hydroponisten vihannesten kasvattamista kasvihuoneisiinsa. Kasvihuoneiden muuntaminen hydroponiseksi kasvatustekniikaksi nostaa kasvihuoneen tuottavuutta, pienentää työvoiman kustannuksia ja parantaa laatua. (Goldsher 2012.)

Muun muassa Hollannissa hydroponiikkaa käytetään paljon kasvihuonekasvatuksessa. Hollanti on maailman toiseksi suurin ruoan viejä USA:n jälkeen. Hollannissa on noin 93 km² kasvihuoneita, joissa hallittu kasvatusta mahdollistaa maan edistyneen tuotannon. (National Geographic 2017.)

Kasvihuoneiden määrä vastaa noin 80 % Hollannin käytetystä viljelysmaasta, eli ne kattavat todella suuren osan Hollannin kasvatusalasta (DW 2019). Hollannin kasvihuonetuotannosta 80 % tapahtuu nykyisin kivivillassa käyttäen pisarakastelumenetelmää, eli toisin sanoen myös suurin osa Hollannin tuotannosta tapahtuu vesikasvatusmenetelmää käyttäen (UCANR 2019).

4 Opinnäytetyön toteutus

Opinnäytetyössä toteutin ravinneliuksen tilan mittaamisen ja mittausdatan tallentamisen tietokantaan. Mittauksia ja datan keruuta varten rakensin IoT-laitteen käyttäen Arduino mikro-ohjainta ja siihen yhdistettyjä sensoreita sekä lisälaitteita. Sensorit mahdollistavat laiteessa eri muuttujien, kuten sähkönjohtavuuden ja lämpötilan mittaamista sekä keräämisen. Sensorien avulla saatavaa mittausdataa lähetetään pilvipalvelussa sijaitsevaan tietokantaan http POST-pyyntöä ja PHP-tiedostoja käyttäen. Tässä opinnäytetyössä rakennettu järjestelmä ei vielä mahdollista täyttä ravinteiden automaattista hallintaa. Ravinnetasapainoa järjestelmässä voidaan hallita lisäämällä ravinteita esimerkiksi kasvien ravinteiden kulutuksen mukaan. Opinnäytetyön toteutuksessa tätä toimintoa on simuloitu lisäämällä arvoja automaattisesti ohjelmakoodissa. Ravinnetasapainon muuttaminen voidaan toteuttaa käytännössä esim. pumpaamalla tuoretta ravinneliuosta kasvatusalustaan, jolla ravinneliuksen ravinnetasapainoa pyritään muuttamaan ja ylläpitämään.

Opinnäytetyön toteutuksessa käytettiin monenlaisia työkaluja. Opinnäytetyö on jaettavissa laitteen fyysiseen kokoonpanemiseen sekä suunnitteluun ja toiminnallisuuksien kehittämiseen ohjelmoimalla. Seuraavissa luvuissa käyn läpi opinnäytetyön fyysisen toteutuksen keskeisiä komponentteja ja niiden yhteensovittamista sekä niiden toiminnallisuutta ja ohjelmointia.

5 Arduino

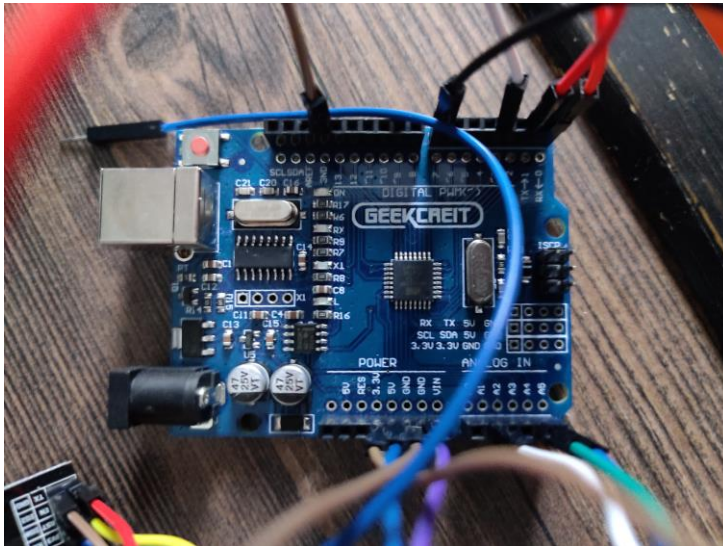
Arduino on avoimeen lähdekoodiin perustuva elektroniikka-alusta. Arduinon tarkoituksena on tarjota käyttäjille helppokäyttöinen ja edullinen laitteisto sekä ohjelmistoalusta, jonka päälle käyttäjät voivat rakentaa omia projektejaan ja sovelluksiaan. Käyttäjät voivat ohjelmoida Arduinon mikro-ohjaimia lukemaan syötteitä erilaisista sensoreista ja kytkimistä, sekä ohjaamaan erilaisia laitteita. Esimerkkinä tällaisesta kytkennästä voisi olla valoisuussensorin arvon perusteella syttyvä LED-valo tai vaikkapa Twitter -syötteestä aktivoituva moottori. (Arduino 2018.)

Arduino on kehitetty Ivrea Interaction Design Instituutista Italiassa. Arduino kehitettiin helpoksi työkaluksi opiskelijoille sekä opettajille ja siksi sen käyttö ei vaadi suurta tietämystä elektroniikasta tai ohjelmoinnista. Tästä huolimatta Arduino on suosittu alusta kehittyneempien käyttäjien parissa, sillä se antaa käyttäjille joustavan työalustan, jolle on helppo rakentaa erilaisia sovellutuksia ja prototyyppejä. Nykyisin Arduino-alusta tarjoaa tuotteita esimerkiksi IoT:n, puettavien teknologioiden, 3D-tulostuksen sekä sulautettujen järjestelmien tarpeisiin. Arduino-alusta ja siihen liittyvät sovellukset ovat täysin avoimeen lähdekoodiin perustuvia, joten käyttäjät voivat rakentaa niille täysin itsenäisesti ja muokata niitä omien tarpeidensa mukaan. (Arduino 2018.)

Arduino-alustan käytön etuina ovat Arduinon mikro-ohjainten edullisuus, niiden järjestelmäriippumattomuus, ohjelmointiympäristöjen helppokäyttöisyys, ohjelmiston avoin lähdekoodi ja laajennettavat ympäristöt sekä itse laitteiston avoin lähdekoodi, joka mahdollistaa piirien itsenäisen kehityksen Creative Commons Lisenssin alaisena (Arduino, 2018). Nämä edut olivat opinnäytetyöni ja kehityksen kannalta todella tärkeitä, sillä varsinkin Arduino-kehitysalustojen hinta on prototyyppien tekemisen kannalta todella tärkeä asia. Myös ympäristön helppokäyttöisyys ja avoimuus olivat prosessien kehittämisen kannalta todella merkittäviä tekijöitä. Tämä näkyy muun muassa siinä, että Arduinosta löytyy helposti tuki erilaisille laitteille ja sensoreille. Uusien kirjastojen ja lisälaitteiden lisääminen Arduinon ohjelmoinnissa ja kytkennässä on helppoa, sillä kirjastot

ovat yleensä yhteensopivia toistensa kanssa ja niiden löytäminen on helppoa Arduinon laajan ja avoimen käyttäjäyhteisön ansiosta. Arduino-alusta tuki myös opinnäytetyöni tarpeita todella hyvin, sillä opinnäytetyössäni tarvitsin alustaa, johon pystyin kytkemään erilaisia sensoreita ja laitteita saaden ne toimimaan yhdessä. Näistä syistä Arduino valikoitui kustannustehokkaaksi ja käytännölliseksi alustaksi opinnäytetyölleni.

Opinnäytetyöhöni valitsin Arduino UNO -kehitysalustan, johon voidaan lisätä toiminnallisuutta käyttäen sen piirilevyssä olevia pinnejä. Arduino Uno oli opinnäytetyöhön sopivin Arduino-malli, sillä se sisältää työhön nähden sopivasti niin analogisia kuin digitaalisiaakin kytkentäpinnejä, joten sen kytkennät riittivät kaikille lisälaitteille ja sensoreille. Arduino UNO sisältää 6 kappaletta analogisia kytkentäpinnejä ja 11 kappaletta digitaalisia kytkentäpinnejä. Opinnäytetyössäni käytin Geekcreitin valmistamaa Arduino UNO -kopiota (kuva 1).

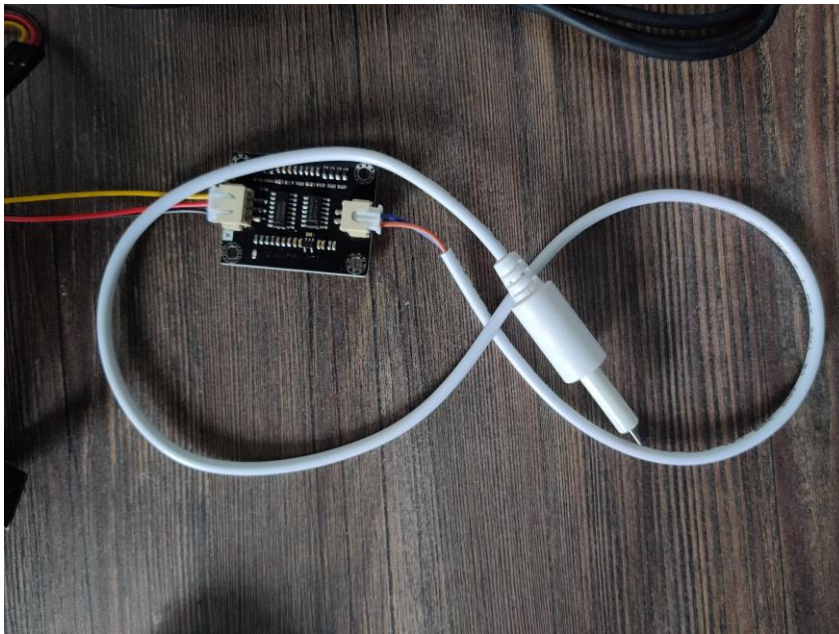


Kuva 1. Geekcreitin valmistama versio Arduino UNO piiristä.

6 Sensorit ja käytetyt lisälaitteet

6.1 Liukoisuussensori

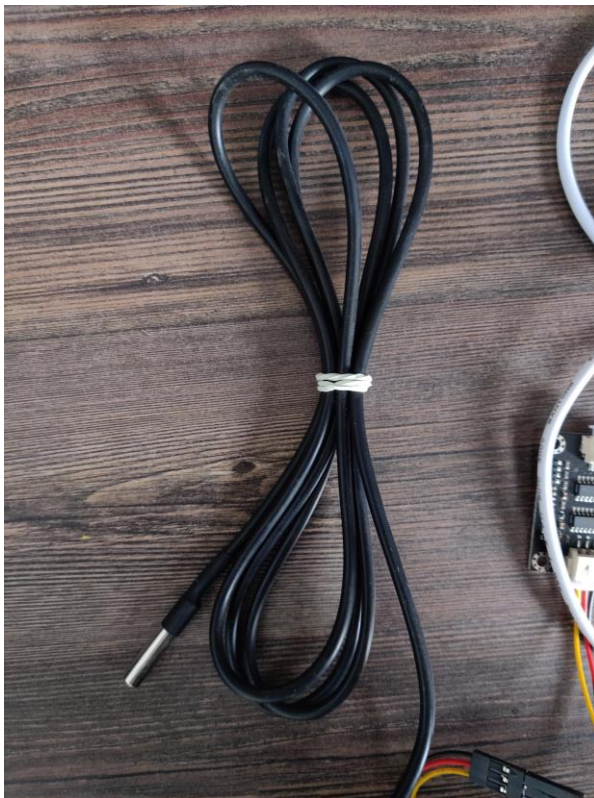
Opinnäytetyön toteuttamiseksi tarvitsin Arduino UNO -kehitysalustaan muutamia ulkoisia sensoreita ja komponentteja. Sähkönjohtavuuden ja liukoisuuksien mittauksia varten tarvitsin liukoisuussensorin, jolla voidaan mitata lienneiden aineiden määriä ravinneliuoksessa. Sensori mittaa sähkönjohtavuutta, jota voidaan lukea Arduinon analogisten pinnien kautta. Sähkönjohtavuutta voidaan muuntaa ohjelmallisesti muun muassa liukoisuudeksi, jota voidaan käyttää ravinnepitoisuuden sekä veden kovuuden mittaamiseen, sillä lienneet ravinnesuolat nostavat liuoksen kokonaissähkönjohtavuutta. Opinnäytetyössäni käytin Gravityn analogista liukoisuussensoria (kuva 2).



Kuva 2. Gravityn analoginen TDS-sensori, jota voidaan käyttää mittaamaan veteen lienneiden aineiden määrää ja sähkönjohtavuutta ravinneliuoksessa. Sensori sisältää piirilevyn ja vedenkestävän liukoisuussensori.

6.2 Lämpösensori

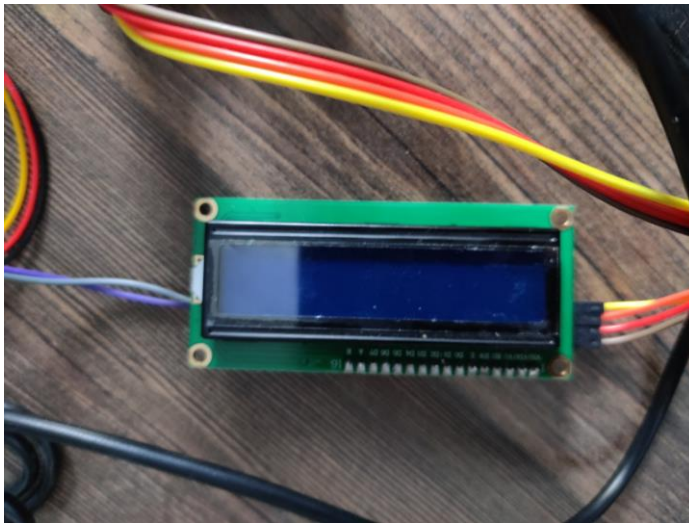
Toisena pääasiallisena sensorina käytin lämpötilasensoria. Lämpötilasensorilla voidaan mitata vesikasvatuksessa joko ravinneliuoksen tai ympäristön lämpötilaa. Lämpötilan mittaaminen liukoisuus- ja sähkönjohtavuusarvon kanssa on todella tärkeää sillä veden sähkönjohtavuus ja liukoisuus muuttuvat lämpötilan muuttuessa. Veden sähkönjohtavuus muuttuu veden ionien nopeamman liikkumisen johdosta veden ollessa lämmintä. Samoin myös veden reaktiivisuus nousee ja sen liukoisuuksien määrä kasvaa nostaen sähkönjohtavuutta. (Shoukat 2020.) Sähkönjohtavuus ravinneliuoksessa kasvaa jopa 2–4 % jokaista veden Celsius-astetta kohden (Ahmed 2017). Näiden muuttujien takia on tärkeää, että myös veden lämpötilaa mitataan säännöllisesti ja jotta tuloksia voidaan korjata oikeiden muuttujien mukaan. Tällaiseen käyttötarkoitukseen tulevan lämpömittarin tuli olla veden- ja lämmönkestävä, joten valitsin tarkoitukseen DS18B20-lämpöanturin (kuva 3).



Kuva 3. DS18B20 vedenkestävä lämpötila-anturi ravinneliuoksen tai ulkoisen lämpötilan mittaamiseen.

6.3 LCD-näyttö

Jotta mitattuja lukemia voitaisiin seurata reaaliajassa, tarvitsin jonkinlaisen näytön tai indikaattorin, jolla arvoja pystytään visualisoimaan. Valitsin näytöksi kaksirivisen nestekidenäytön, johon voin kirjoittaa ja joka näyttää lämpötilan-, sähkönjohtavuuden- sekä liukoisuudenarvoja sekä muuta tarvittavaa tietoa (kuva 4). Näyttö voidaan ohjelmoida näyttämään lukemia reaaliajassa ja tarvittaessa näytön tilaa voidaan animoida rullaamaan sivulle tilan lisäämiseksi.

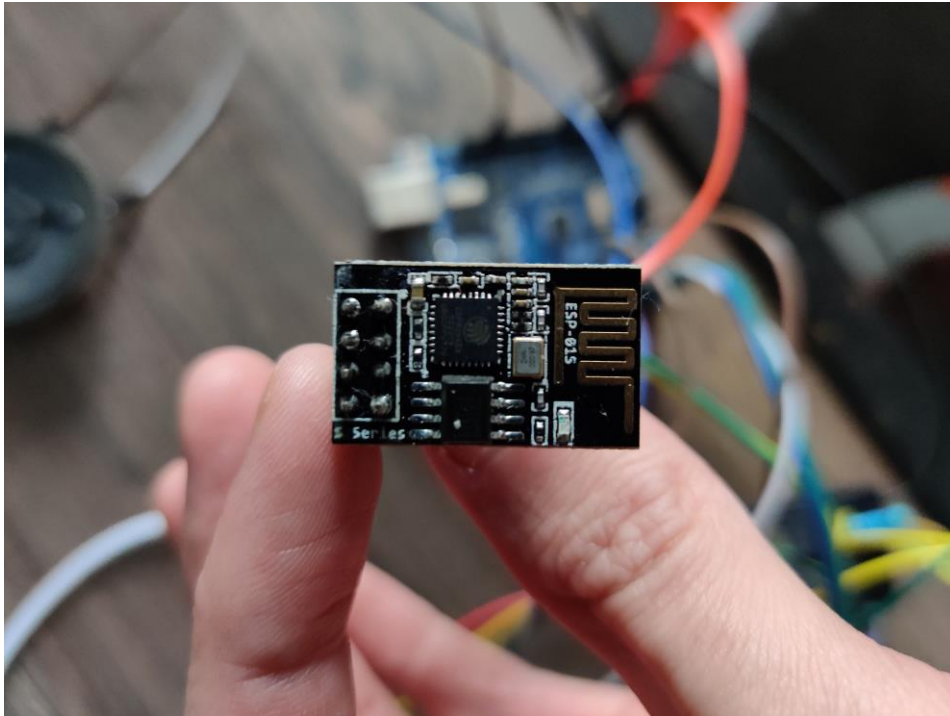


Kuva 4. Yksinkertainen taustavaloilla varustettu nestekidenäyttö, jossa voidaan näyttää reaaliaikaista lämpötilaa, sähkönjohtavuutta sekä liukoisuutta.

6.4 ESP-01 WiFi-moduuli

ESP-01 on ESP8266-piiriin perustuva WiFi-moduuli, joka mahdollistaa mikropiirien yhdistämisen WiFi-verkkoon ja tietoliikenneyhteyksien käyttämisen WiFi-verkkojen kautta. ESP8266-moduulia voi esimerkiksi ohjata mikropiiristä UART-sarjaliikennepiirin kautta käyttäen AT-komentoja. (Fabacademy 2015.) Opinnäytetyössäni käytin WiFiEsp Arduino -kirjastoa, jolla sain yhdistettyä moduulin WiFi-verkkoon. WiFiEsp kirjaston ansiosta AT-komentoja ei tarvinnut käyttää, vaan pystyin ohjaamaan moduulia WiFiEsp kirjaston metodeilla. Tarvitsin ESP-moduulia yhdistääkseni Arduinon verkkoon, jotta sain siirrettyä

lämpö- ja sähkönjohtavuusdataa verkossa sijaitsevalle tietokantapalvelimelle. ESP yhdistyy paikalliseen lähiverkkoon, jonka kautta se voi olla yhteydessä verkossa olevien laitteiden ja sovelluksien kanssa käyttäen esimerkiksi http-kutsuja. Eräänä ESP-01-piirin etuna on sen pieni koko ja sen helppo liitettävyys Arduino -ympäristöön (Kuva 5).

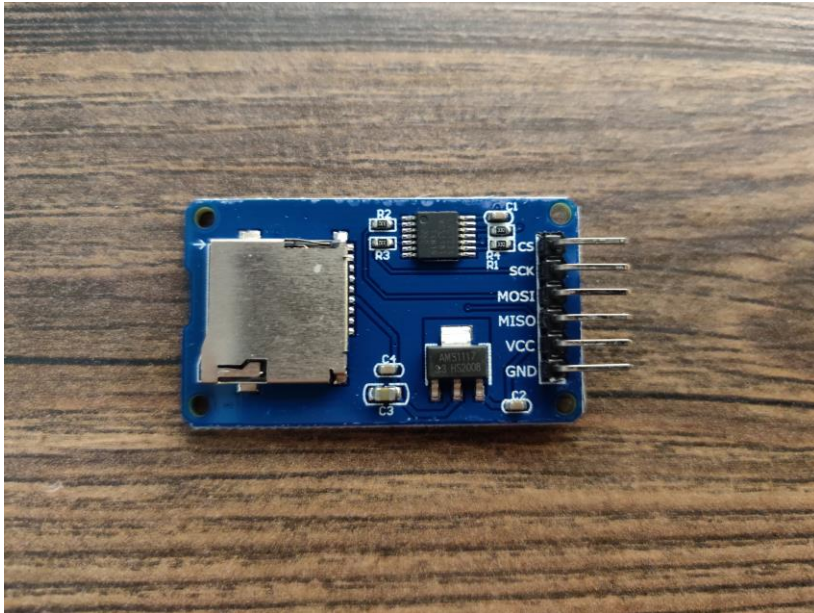


Kuva 5. ESP8266-piiriin perustuva ESP-01-piiri, jossa on sisäänrakennettu antenni.

6.5 Muistikortinlukija

Muistikortinlukija oli opinnäytetyössäni vaihtoehtoinen tapa kerätä dataa micro-SD-muistikortille, mikäli datan siirtäminen verkossa sijaitsevaan tietokantaan ei onnistu. Datan keräämistä varten hankin opinnäytetyöhöni varalle micro-SD-adapterin (Kuva 6). Datan kerääminen muistikortille oli myös eräs tapa varmuuskopioida kerättyä IoT-dataa paikallisesti, sekä kerätä tietoa myös tietoverkkojen yhteyksien ollessa epäkunnossa. Tiedot voidaan myöhemmin siirtää tietokantaan manuaalisesti tai niitä voidaan hyödyntää paikallisesti

muistokortilla sijaitsevasta tekstitiedostosta, johon Arduino voi tallentaa sen keräämää sensoridataa. Muistikortin lukija ei kuitenkaan ollut välttämätön komponentti opinnäytetyössä, joten sitä ei varsinaisesti otettu käyttöön lopullisessa toteutuksessa, mutta sen toimintaa ja lisäämistä tutkittiin edellä mainituista syistä.



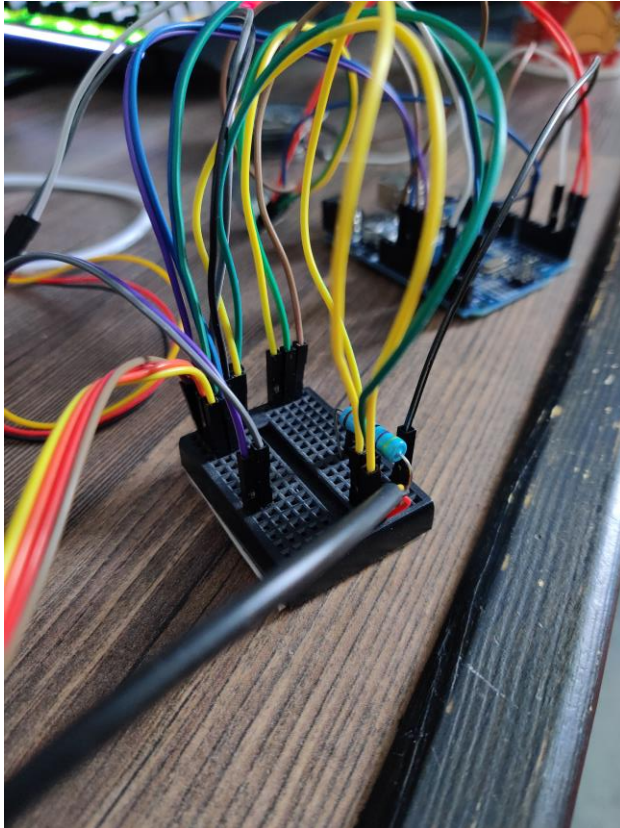
Kuva 6. Arduino yhteensopiva mikro-SD-adapteri, joka mahdollistaa kortilta lukemisen ja kirjoittamisen.

6.6 Muut komponentit ja tarvikkeet

Opinnäytetyön kehitys oli erittäin laajalti prototyyppien suunnittelua ja kehittämistä, joten tarvitsin joustavan kehitysratkaisun, johon tehdyt muutokset eivät olisi pysyviä. Päädyin käyttämään kehityksessä eräänlaisia koekytkentäalustaa, johon pystyin tekemään kytkentöjä Arduinosta käyttäen hyppylankoja. Alustan käyttäminen oli helppoa, koska hyppylankojen kytkeminen sekä poistaminen oli nopeaa ja se mahdollisti erilaisten kytkentöjen nopean testaamisen ja muuttamisen.

Koekytkentäalustan kytkentäkolot ovat yhteydessä toisiinsa riveittäin. Tästä syystä koekytkentäalustalle oli helppo luoda virta- ja maadoituskiskoja, joihin

laitteet voitiin kytkeä. Tämä tarkoittaa sitä, että esimerkiksi maadoitus voitiin suorittaa kytkemällä yksi Arduinin maadoitus pinni kiinni yhteen kiskoon, josta se voitiin jakaa moneen eri liitäntään ja laitteeseen (Kuva 7).



Kuva 7. Koekytkentäalusta, johon voidaan tehdä kytkentöjä hyppylangoilla.

Mikäli laitteistosta halutaan myöhemmin tehdä kiinteämpiä prototyypppejä, voidaan sen kehitys myös siirtää esimerkiksi täplä- tai nauhakuparoidulle prototyyppilevyille, jolloin hyppylangat voidaan juottaa kiinni piirilevyyn tai laitteelle voidaan tilata tai tehdä valmis piirilevy.

Muina komponentteina opinnäytetyössä tarvittiin muun muassa $4.7\text{K}\Omega$ etuvastus lämpösensorille, LED-valoja testaamiseen, hyppylankoja, sekä kaiutinta merkkiäänien tuottamiseen. Näistä komponenteista ainakin LED-valoja sekä kaiutinta voitiin käyttää niin testaamisen yhteydessä kuin myös paikallisina merkinä ravinnetasapainon muutoksien ilmaisemiseksi LCD-näytön lisäksi (Kuva 8).



Kuva 8. Geneerinen kaiutnelementti merkkiäänien tuottamiseksi.

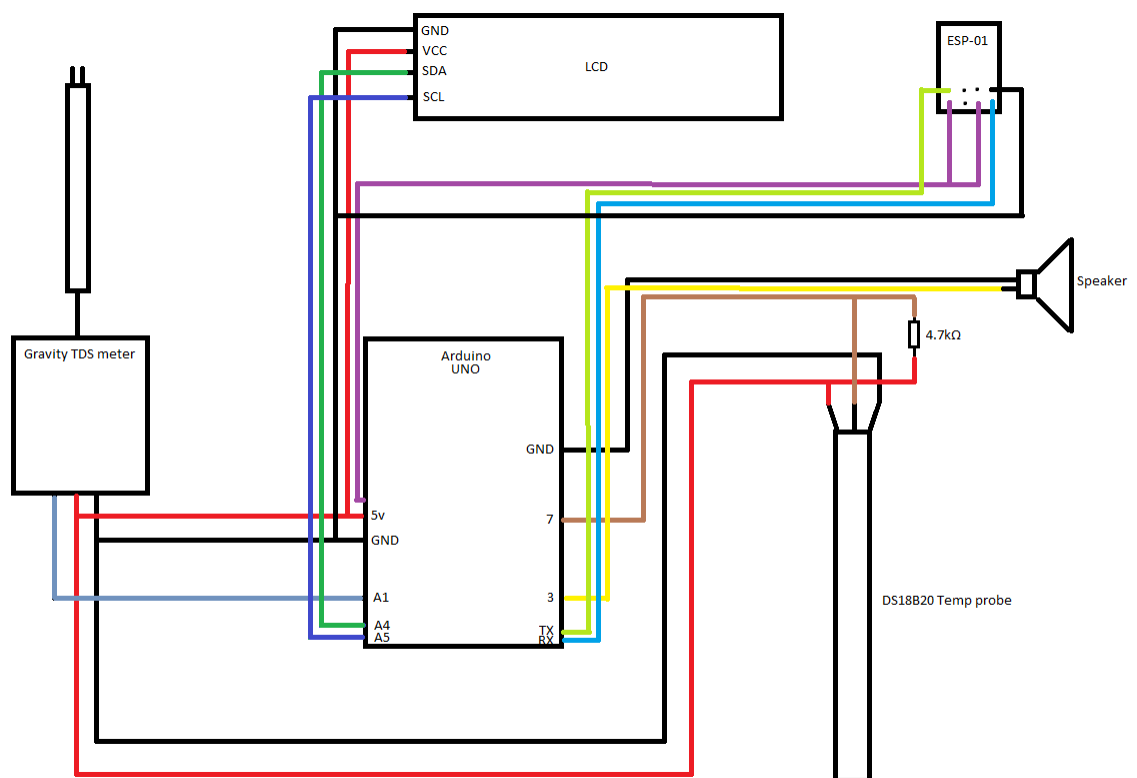
7 Laitteen kokoonpano ja kytkennät

Laitteiston kokoonpaneminen ja kytkeminen oli helppoa Arduinon liittimien ja koekytkentäalustan ansiosta. Kytkökset tehtiin pääasiassa Arduinosta koekytkentäalustalle ja koekytkentäalustasta lisälaitteisiin sekä sensoreihin. Laitteiden kytkeminen toisiinsa oli helposti ymmärrettävää, sillä laitteiden kytkennät voidaan tehdä koekytkentäalustan eri alueille ja ne on helppo tehdä kytkentäkaavion mukaan. Kytkentäkaavion laatiminen oli tärkeä prosessi kokonaisuuden kannalta, sillä se auttoi laitteen kokoonpanemisessa ja toimi selkeänä referenssinä toiminnallisuuden ohjelmointivaiheessa.

Suuri haaste laitteen suunnittelussa ja kytkentöjen toteuttamisessa oli kytkentäkaavion kehittämisessä. Arduinon kytkettäessä on tiedettävä minkä tyyppisiä liittäntöjä ollaan tekemässä, sillä jotkin laitteet vaativat analogisia liittäntöjä ja toiset digitaalisia liittäntöjä. On myös tärkeää tietää mihin pinniin laitteet ovat yhteydessä Arduinossa, sillä niitä tarvitaan ohjelmoinnin yhteydessä,

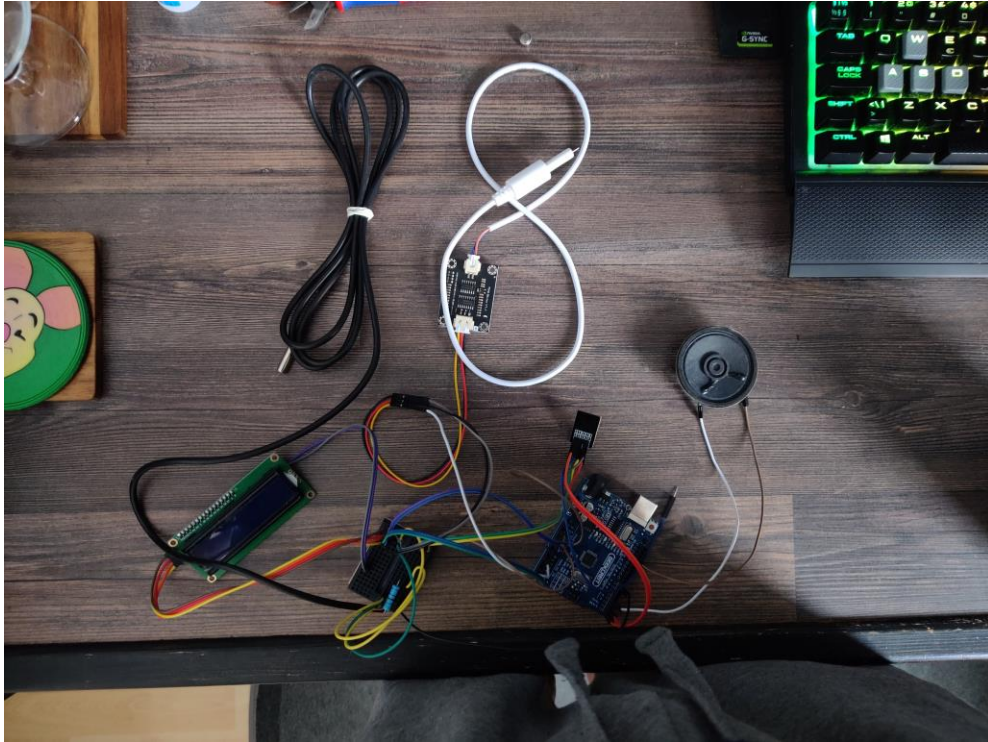
jossa laitteet otetaan käyttöön ja niiden pinneihin ohjelmoidaan toiminnallisuuksia.

Opinnäytetyöni aiheelle ja erillisille käyttämilleni sensoreille sekä laitteille löytyi paljon ohjeita verkosta, mutta mikään niistä ei kattanut suoraan opinnäytetyöni ja laitteistoni kaavaa. Tehdessäni kytkentäkaaviota jouduin muokkaamaan ja yhdistämään todella paljon ohjeita ja toiminnallisuuksia toimivaksi kokonaisuudeksi. Lopulta sain yhdistettyä kaikki laitteet yhteen, josta sain käyttämäni kytkentäkaavion (Kuvio 3).



Kuvio 3. Arduino järjestelmäni kytkentäkaavio.

Kun kaikki laitteet ovat kytkettyinä Arduinoon on laitteiden välisten kytkentöjen ja laitteiden erottaminen toisistaan helppoa. Laitteet muodostavat selkeästi ymmärrettävän ja erillään olevan kokonaisuuden (Kuva 9).



Kuva 9. Kokoonpantu Arduino-laite.

8 Arduinin ohjelmointi

8.1 Arduinin ohjelmointikieli ja ympäristö

Arduinin ohjelmointi oli teknisesti helppoa. Ohjelmakoodin syvempi ymmärtäminen ja kokonaisuuden hahmottaminen osoittautuivat kuitenkin vaikeammiksi kokonaisuuksien suurentuessa. Kuten kaikissa ohjelmointia sisältävissä tehtävissä, uuden kielen ja kirjastojen käyttäminen ja opiskelu olivat eräät suurimmista teknisistä haasteista ja ne kuluttivat huomattavasti eniten aikaa koko ohjelmointiprosessissa. Varsinkin ESP8266:n toiminnallisuuden ohjelmoinnin yhteydessä törmäsin monenlaisiin ongelmatilanteisiin, joiden selvittämiseen minulta meni paljon aikaa. Näistä teknisistä seikoista huolimatta kehitys oli tehokasta, sillä Arduinolla oli erittäin aktiivinen yhteisö ja toimiva ohjelmointialusta. Käytössäni oli todella paljon erilaisia laajoja ja helppokäyttöisiä

kirjastoja sekä resursseja, joiden avulla ohjelmakoodin kirjoittaminen oli ongelmista huolimatta lopulta kohtuullisen helppoa.

Arduinon ohjelmointi tapahtuu Arduino IDE -ohjelmointiympäristössä käyttäen C- ja C++-ohjelmointikielen ja Arduinon omiin metodeihin ja funktioihin pohjautuvaa Arduinon omaa ohjelmointikieltä. Arduinon ohjelmointikielenä voidaan myös käyttää muita ohjelmointikieliä, mutta päätin käyttää Arduinon omaa oletuskieltä. Arduinon oletuskielen käyttäminen oli huomattavasti helpompaa, sillä uutta alustaa tutkiessa olin valmistautunut kohtaamaan ongelmia. Myös ongelmien ratkaisu oli helpompaa oletuskielellä, sillä siihen löytyi todella hyvin ohjeita ja valmista koodia sekä esimerkkejä, joiden avulla ohjelmakoodia oli helppo testata ja jatkokehittää. Ohjelmoinnin jälkeen Arduino IDE kääntää kirjoitetun ohjelmakoodin Arduinon käyttämään sketch-tiedostomuotoon, jota Arduino pystyy käsittelemään konekieltä käyttäen. Tämän jälkeen sketch-tiedosto ladataan USB-kaapelia ja tietokoneen sarjaporttia käyttäen Arduinoon. Latauksen jälkeen ohjelmakoodi aktivoituu Arduinon käyttöön jokaisen käynnistyskerran yhteydessä.

8.2 Ohjelmakoodi ja sen rakenne

Arduinon ohjelmakoodi koostuu kahdesta pääfunktioista (Kuvio 4). Setup-funktiossa Arduino valmistellaan käyttöä varten. Setup-funktio suoritetaan vain kerran Arduinon käynnistyessä tai aina kun Arduino nollataan. Setup-funktio sisältää tästä syystä kaiken sellaisen toiminnallisen osuuden, jota ei haluta suorittaa kuin kerran, kuten laitteiston käyttöönotto ja verkkoihin kytkeytymiset. Loop-funktio taas sisältää virallisen toiminnallisen ohjelmakoodin, jota Arduino suorittaa. Loop-funktio toistaa sen ohjelmakoodia niin pitkään kuin Arduino on päällä ja sen ulkopuoliset funktiot suoritetaan kutsumalla niitä loop-funktion sisältä. Loop-funktio sisältää esimerkiksi sensorien arvojen mittaamisen ja sensoritietojen asettamisen muuttujiin, sekä niiden käyttämisen ja lähettämisen verkkoon.

void Setup()	void Loop()
Alustaa sarjaportin, LCD-näytön, Lämpömittarin, Wifi moduulin Valmistelelee ja yhdistää ESP-01 moduulin lähiverkkoon	Lukee sensorien arvot ja säättää sähkönjohtavuuden arvoa lämpötilan mukaan Testaa yhteyden palvelimeen Muotoilee ja lähettää HTTP Post-kutsun PHP tiedostoon

Kuvio 4. Arduino ohjelmakoodin pääasialliset toiminnot opinnäytetyössä

Toteutuksen Arduinon ohjelmakoodin tarkoitus on mitata lämpötilaa, liukoisuutta ja sähkönjohtavuutta sensoreiden avulla ja lähettää dataa eteenpäin käyttäen hyväksi Arduinon liitettyä ESP-01 WiFi-moduulia. Ohjelmakoodi yhdistää aluksi Arduino mikro-ohjaimen koodissa sijaitsevilla muuttujilla asetettuun WiFi-verkkoon ja valmistelelee sen toiminnan setup-vaiheessa. WiFi-verkkoon yhdistäminen tapahtui WiFiEsp kirjastoa hyväksi käyttäen, joka mahdollisti Arduinon yhdistetyn ESP-01 WiFi-moduulin käytön yksinkertaisilla komennoilla (Kuva 10).

```

// initialize serial for ESP module
Serial1.begin(9600);

// initialize ESP module
WiFi.init(&Serial1);

// check for the presence of the shield
if (WiFi.status() == WL_NO_SHIELD) {
  Serial.println("WiFi shield not present");
  // don't continue
  while (true);
}

// attempt to connect to WiFi network
while ( status != WL_CONNECTED) {
  Serial.print("Attempting to connect to WPA SSID: ");
  Serial.println(ssid);
  // Connect to WPA/WPA2 network
  status = WiFi.begin(ssid, pass);
}

Serial.println("You're connected to the network");

```

Kuva 10. Arduino yhdistyy WiFi-verkkoon WiFiEsp kirjastoa käyttäen.

Setup-vaiheen jälkeen Arduino alkaa toistamaan loop-vaihetta, jossa se kerää sensoridataa ja lähettää sitä verkkoon. Loop-vaiheen alussa Arduino kutsuu sensorien lukufunktiota, joka lukee ja käsittelee sensorien arvot tietyn säädettävän viiveen mukaan sekä kirjoittaa ne näkyviin LCD-näytölle. Arduino laskee sensorien antamien arvojen mukaan liuksen lämpötilan käyttäen DallasTemperaturen lämpöanturi-kirjastoa ja mittaa sähkönjohtavuuden analogi pinnien kautta. Sähkönjohtavuuden arvo käsitellään ja muunnetaan liukoisuus arvoksi ja sähkönjohtavuutta kompensoidaan lämpötilakertoimen avulla. Lämpökertoimen käyttö ja säätö ovat tärkeitä vesikasvatusympäristön hallinnan kannalta, jotta liukoisuusarvo saadaan mahdollisimman tarkaksi ja toimimaan niin, että se pysyy oikeana lämpötilasta riippumatta. Tämän jälkeen lukufunktio tulostaa mitatut mittariarvot LCD-ruudulle niille osoitettuihin pakkoihin (Kuva 11).

```

void readSensorValues() { //READ SENSOR VALUES

    dallasTemperature.requestTemperatures(); //read temperature
    sensor::waterTemp = dallasTemperature.getTempCByIndex(0);
    float rawEc = analogRead(pin::tds_sensor) * device::aref / 1024.0;
    // read the analog value more stable by the median filtering algorithm, and convert to voltage value
    float temperatureCoefficient = 1.0 + 0.02 * (sensor::waterTemp - 25.0);
    // temperature compensation formula: fFinalResult(25°C) = fFinalResult(current)/(1.0+0.02*(fTP-25.0));
    sensor::ec = (rawEc / temperatureCoefficient) * sensor::ecCalibration;
    // temperature and calibration compensation
    sensor::tds = (133.42 * pow(sensor::ec, 3) - 255.86 * sensor::ec * sensor::ec + 857.39 * sensor::ec) * 0.5;
    //convert voltage value to tds value
    Serial.print(F("TDS:")); Serial.println(sensor::tds);
    Serial.print(F("EC:")); Serial.println(sensor::ec, 2);
    Serial.print(F("Temperature:")); Serial.println(sensor::waterTemp,2);
    lcd.clear(); //Print values on LCD
    lcd.print("TDS  EC  Temp");
    lcd.setCursor(0,1);
    lcd.print(sensor::tds);
    lcd.setCursor(5,1);
    lcd.print(sensor::ec, 2);
    lcd.setCursor(11,1);
    lcd.print(sensor::waterTemp,2);
}

```

Kuva 11. readSensorValues funktio lukee sensorien arvot, kompensoi sähkönjohtavuuden lämpökertoimen mukaan ja tulostaa ne LCD-näytölle.

Seuraavaksi Arduino kutsuu addNutrients -funktioita, joka tarkastaa liukoisuusarvon suuruutta. Mikäli Arduino tunnistaa mitatun liukoisuusarvon olevan kynnysarvon alapuolella, järjestelmä antaa ilmoituksen ja lisää ravinneliuosta kasvatusalustaan. Koska opinnäytetyössä ei tällä hetkellä voitu toteuttaa fyysistä ravinteiden lisäystä, simuloitiin sen toimintaa ohjelmakoodissa. Ravinteiden lisäys toteutettiin nostamalla liukoisuusmuuttujan arvoa kerrallaan 25 liukoisuuden asteella, kunnes arvo ylittää kynnysarvon (Kuva 12). Fyysisessä toteutuksessa lisääminen voitaisiin toteuttaa esimerkiksi pumpuilla tai sähkömoottoreilla ja pumppaamalla tuoretta ravinneliuosta järjestelmään.

```

void addNutrients() { //SIMULATE NUTRIENT ADDITION

    if (sensor::tds < 200) { //Chenck if nutrients are below 200 TDS
        while (sensor::tds < 200) { //Add nutrients until you are at 200 TDS
            Serial.println("Water TDS under 200");
            Serial.print(F("TDS:")); Serial.println(sensor::tds);
            Serial.println("Adding nutrients");
            sensor::tds = sensor::tds + 25; //ADD 25 to TDS
            Serial.print(F("New TDS:")); Serial.println(sensor::tds);
            Serial.println();
            delay(3000);
        }
    }
}

```

Kuva 12. addNutrients -funktio simuloi ravinteiden lisäämistä liukseen lisäämällä sen liukoisuus arvoa.

Seuraavaksi Arduino ottaa yhteyden 000Webhost -palvelussa ylläpidettyyn verkkosivuun, johon se lähettää dataa. Arduino lähettää dataa sivustolla sijaitsevaan Post.php -tiedostoon http POST pyynnöllä, jonka se muotoilee sensoriarvoista ja muista tarvittavista muuttujista. Muuttujat kerätään content -merkkijonoon, joka lähetetään palvelimelle. Pyynnön lähettämisen jälkeen Arduino sulkee yhteyden ja jää odottamaan seuraavaa mittauskierrosta annetun viiveen mukaisesti (Kuva 13).

```

    Serial.println("Starting connection to server...");
    // if you get a connection, report back via serial
    if (client.connect(server, 80)) {
        Serial.println("Connected to server");

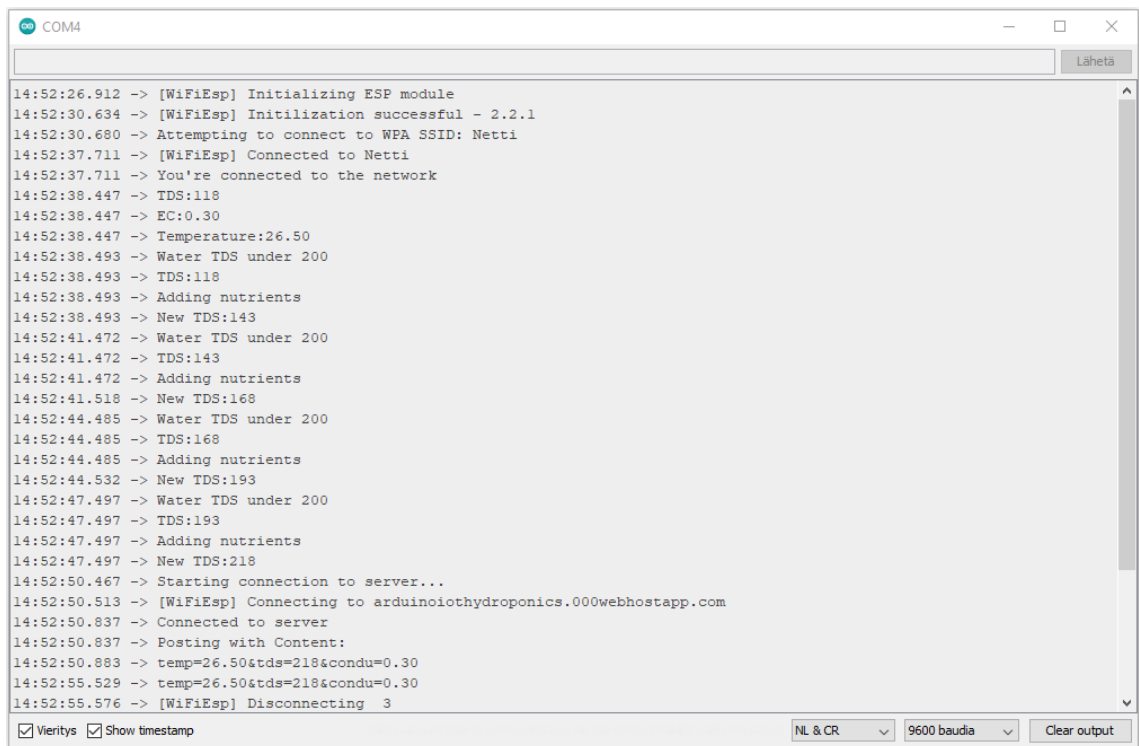
        String content = temp_post + sensor::waterTemp + "&" + tds_post + sensor::tds + "&" + condu_post + sensor::ec;
        Serial.println("Posting with Content:");
        Serial.println(content);

        client.println("POST /test/post.php HTTP/1.1");
        client.println("Host: arduinoiohydroponics.000webhostapp.com");
        //client.println("Accept: */*");
        //client.print("Content-Length: ");
        // client.println(content.length());
        client.println("Content-Type: application/x-www-form-urlencoded");
        client.print("Content-Length: ");
        client.println(content.length());
        client.println();
        client.println(content);
        Serial.println(content);
    }
    if (client.connected()) {
        client.stop();
    }
    delay(10000);
}

```

Kuva 13. http POST-pyyntö muodostetaan erilaisista tulosteista, jotka saavat arvonsa muuttujien ja annettujen osoitteiden mukaan, jonka jälkeen pyyntö lähetetään palvelimelle koottuun osoitteeseen.

Laitteen toimintaa voidaan seurata muun muassa Arduinon sarjamonitorista laitteen ollessa kytkettynä tietokoneeseen sarjaportin kautta. Arduino ohjelmoitiin tulostamaan erilaisia merkkijonoja sarjamonitoriin. Sarjamonitorista voidaan lukea missä vaiheessa suoritusta Arduino on ja minkälaisia arvoja se käsittelee. Sarjamonitorissa Arduino kertoo muun muassa mihin WiFi-verkkoon se yhdistää, mitkä ovat viimeisimmät sensoriarvot, mikä liukoisuuden tila on ja kuinka se yhdistää ja lähettää tietoa palvelimelle http POST-pyynnöllä (Kuva 14).



```
14:52:26.912 -> [WiFiEsp] Initializing ESP module
14:52:30.634 -> [WiFiEsp] Initialization successful - 2.2.1
14:52:30.680 -> Attempting to connect to WPA SSID: Netti
14:52:37.711 -> [WiFiEsp] Connected to Netti
14:52:37.711 -> You're connected to the network
14:52:38.447 -> TDS:118
14:52:38.447 -> EC:0.30
14:52:38.447 -> Temperature:26.50
14:52:38.493 -> Water TDS under 200
14:52:38.493 -> TDS:118
14:52:38.493 -> Adding nutrients
14:52:38.493 -> New TDS:143
14:52:41.472 -> Water TDS under 200
14:52:41.472 -> TDS:143
14:52:41.472 -> Adding nutrients
14:52:41.518 -> New TDS:168
14:52:44.485 -> Water TDS under 200
14:52:44.485 -> TDS:168
14:52:44.485 -> Adding nutrients
14:52:44.532 -> New TDS:193
14:52:47.497 -> Water TDS under 200
14:52:47.497 -> TDS:193
14:52:47.497 -> Adding nutrients
14:52:47.497 -> New TDS:218
14:52:50.467 -> Starting connection to server...
14:52:50.513 -> [WiFiEsp] Connecting to arduinoiohydroponics.000webhostapp.com
14:52:50.837 -> Connected to server
14:52:50.837 -> Posting with Content:
14:52:50.883 -> temp=26.50&tds=218&condu=0.30
14:52:55.529 -> temp=26.50&tds=218&condu=0.30
14:52:55.576 -> [WiFiEsp] Disconnecting 3
```

Kuva 14. Arduino on ohjelmoitu kirjoittamaan merkkijonoja ja muuttujien arvoja sarjamonitoriin, jotta sen toimintaa voidaan seurata. Kuvassa Arduinon tekemä kokonainen kierros ohjelmakoodia käynnistyksen yhteydessä, jossa se yhdistää WiFi-verkkoon ja mittaa sekä lähettää dataa palvelimelle http POST-pyynnöllä.

9 PHP

PHP on avoimen lähdekoodin yleiskäyttöinen ohjelmointikieli, jota käytetään muun muassa www-sivujen dynaamiseen muokkaukseen palvelimella ja sitä voidaan sisällyttää HTML-koodiin. HTML-koodin perusteella www-selaimet muotoilevat käyttäjille näkyvät sivut (PHP 2021). Opinnäytetyössäni tarvitsin PHP:tä koska tarvitsin jonkin alustan, jonka kautta lähettää tietoa Arduinosta tietokantaan. PHP toimi erittäin hyvänä alustana tietojen välittämiseen, sillä sen integrointi SQL-tietokantoihin oli todella tehokasta. PHP oli myös todella helppokäyttöinen alusta, johon löytyi paljon ohjeita ja tietoa muun muassa tietokanta yhteyksien tekemisestä ja POST-pyyntöjen vastaanottamisesta.

9.1 Tiedonsiirto PHP:llä

PHP:llä tein 000webhostissa sijaitsevalle verkkosivulleni POST.php -nimisen tiedoston. Kun POST.php -tiedostoa kutsutaan, se ottaa muuttujien arvot vastaanotetun POST pyynnön merkkijonosta ja tallentaa muuttujien arvot temp-html -sivulle (Kuva 15), sekä lähettää ne haluttuun tietokantaan SQL-yhteydellä. POST.php luo tietokantayhteyden SQL-tietokantaan mysqli_connect funktiolla. Funktio avaa SQL-tietokantayhteyden annettujen tunnusten mukaan ja yhteyden onnistuessa se lähettää tietokantaan muotoillun INSERT SQL-kyselyn, jolla muuttujien arvot voidaan lisätä haluttuun ennalta asetettuun tauluun tietokannassa.

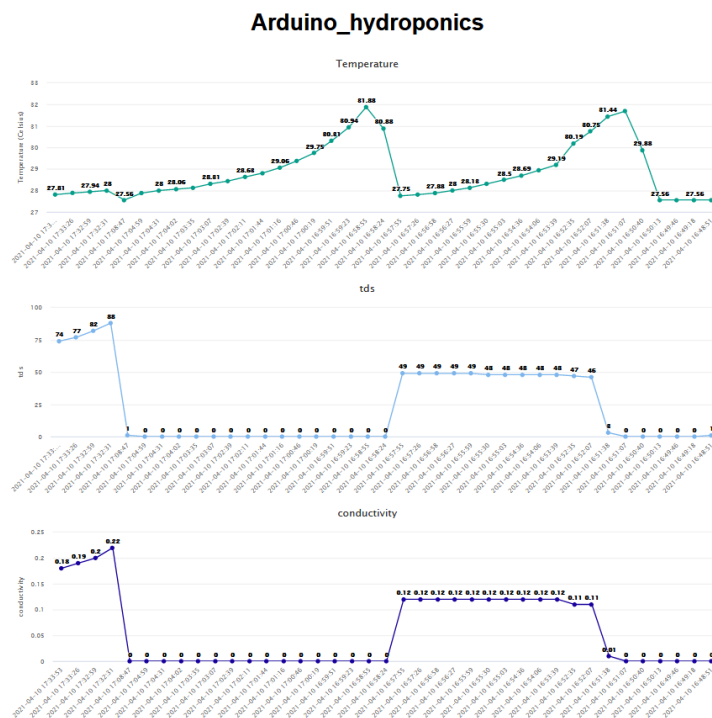


1619084740 - 26.44 - 71 - 0.18

Kuva 15. Temp.html sivu, jolle kirjoitetaan aina viimeisimmät vastaanotetut mittariarvot ja aikaleima ennen tietokantaan vientiä. Sivua voidaan käyttää esim. Arduinon ja PHP tiedoston testaamiseen sekä tiedon etälukemiseen.

9.2 Sensoridatan visualisointi PHP:llä

Eräänä lisäominaisuutena lisäksi palvelimelle myös Arduino_hydroponics.php tiedoston. Tiedoston avulla tietokannasta voi hakea kerättyä sensoridataa ja visualisoida sitä graafiseen muotoon. Arduino_hydroponics ottaa yhteyden mpa_1703349_arduino tietokantaan ja pyytää sieltä haluttujen kenttien arvoja. Kenttien arvoja käytetään Highcharts.js skriptissä kasvatustiedon graafien luomiseen. Highcharts tekee kerätyistä mittariarvoista datapisteitä ja muuntaa ne graafiseen muotoon tehden niistä viivakaavion (Kuvio 4).



Kuvio 4. Arduino_hydroponics.php luo visuaalisen viivakaavio tietokannan tauluista haetuista mittariarvokentistä ja päivämäärä kentästä.

10 MySQL-tietokanta

10.1 Tietokanta

MySQL-tietokantaa tarvittiin opinnäytetyössä Aduinon keräämän sensoridatan pidempiaikaiseen tallentamiseen. Opinnäytetyössä käytin tekemääni mpa_1703349_arduino MySQL-tietokantaa, joka on osa käyttämäni 000webhost palvelua. Tietokantaan loin taulun arduino_hydroponics, johon lisäsin sensoreista kerättyä dataa. Opinnäytetyön kehityksen yhteydessä testasin myös Karelian MariaDB-tietokanta alustaan pohjautuvaa tietokantaa, joka sijaitsee Microsoftin Azuren pilvipalvelussa. Molemmat yhteydet toimivat, mutta lopulta päädyin käyttämään omaa tietokantaani 000webhostingissa, sillä siihen datan kerääminen oli käytännöllisempää tulevien henkilökohtaisten datan visualisointien ja käyttämisten kannalta. Yhteys toimi kuitenkin myös Karelian tietokantaan ja sen tauluihin voitiin lisätä dataa PHP tiedoston kutsun kautta. Karelian tietokantaan tietoa lisätessä tarvittiin myös hieman erilaista lähestymistapaa, sillä tietokannan autocommit ei ollut päällä, joten tietokantaan tehdyt muutokset täytyi hyväksyä ennen kuin ne tallentuivat tauluun. Tämä onnistui avaamalla tietokantaan transaktio ennen SQL-kyselyä ja committaamalla muutokset kantaan kyselyn jälkeen.

10.2 Tietokannan taulut

Arduino_hydroponics-taulu sisältää ID-, lämpötila-, liukoisuus-, sähkönjohtavuus- sekä päivämääräkentät, joita käytetään datan keräämiseen (Kuva 16). ID-kenttää käytetään IoT-sensoridatan indeksointiin ja se toimii taulun perusavaimena. ID kentän arvo päivittyy automaattisesti uusien rivien liittämisen yhteydessä AUTO_INCREMENT toiminnon ansiosta. Lämpötila, liukoisuus ja sähkönjohtavuus -kentät sisältävät itse kerättävän sensoridatan liukulukuina. Tulevaisuudessa näiden tietojen lisäksi järjestelmään voitaisiin tallentaa

esimerkiksi arvot ulkoisesta lämpötilasta, valon määrästä ja ilman kosteudesta. Päivämääräkenttä sisältää datan tallennuspäivämäärän datetime-tyyppisenä.

#	Nimi	Tyyppi	Aakkosjärjestys	Attribuutit	Tyhjä	Oletusarvo	Kommentit	Lisätiedot	Toiminnot
<input type="checkbox"/>	1 ID	bigint(20)			Ei	None		AUTO_INCREMENT	Muokkaa Tuhoa Lisää
<input type="checkbox"/>	2 temp	float			Kyllä	NULL			Muokkaa Tuhoa Lisää
<input type="checkbox"/>	3 tds	float			Kyllä	NULL			Muokkaa Tuhoa Lisää
<input type="checkbox"/>	4 conductivity	float			Kyllä	NULL			Muokkaa Tuhoa Lisää
<input type="checkbox"/>	5 date	datetime			Ei	current_timestamp()			Muokkaa Tuhoa Lisää

Kuva 16. Arduino_hydroponics taulu ja sen rakenne.

Kun tietokantaan on lähetetty dataa, sitä voidaan tarkastella tekemällä tietokantaan SQL-kysely. Tässä tapauksessa kyselyksi riittää `SELECT*FROM 'arduino_hydroponics'`, joka näyttää kaikki arduino_hydroponics taulun kentät (Kuva 17).

✓ Näytetään rivit 0–24 (39 tulos, Kysely kesti 0.0013 sekuntia)

```
SELECT * FROM 'arduino_hydroponics'
```

1 > >> Näytä kaikki | Rivien määrä: 25 Suodata rivejä: H

ID	temp	tds	conductivity	date
10	27.56	1	0	2021-04-10 16:48:51
11	27.56	0	0	2021-04-10 16:49:18
12	27.56	0	0	2021-04-10 16:49:46
13	27.56	0	0	2021-04-10 16:50:13
14	29.88	0	0	2021-04-10 16:50:40
15	31.69	0	0	2021-04-10 16:51:07
16	31.44	3	0.01	2021-04-10 16:51:38
17	30.75	46	0.11	2021-04-10 16:52:07
18	30.19	47	0.11	2021-04-10 16:52:35
19	29.19	48	0.12	2021-04-10 16:53:39
20	28.94	48	0.12	2021-04-10 16:54:06
21	28.69	48	0.12	2021-04-10 16:54:36
22	28.5	48	0.12	2021-04-10 16:55:03
23	28.31	48	0.12	2021-04-10 16:55:30
24	28.13	49	0.12	2021-04-10 16:55:59
25	28	49	0.12	2021-04-10 16:56:27
26	27.88	49	0.12	2021-04-10 16:56:58
27	27.81	49	0.12	2021-04-10 16:57:26
28	27.75	49	0.12	2021-04-10 16:57:55
29	30.88	0	0	2021-04-10 16:58:24
30	31.88	0	0	2021-04-10 16:58:55
31	30.94	0	0	2021-04-10 16:59:23
32	30.31	0	0	2021-04-10 16:59:51
33	29.75	0	0	2021-04-10 17:00:19

Kuva 17. Kerättyä sensoridataa tietokanta näkymässä. Samat arvot visualisoituna myös Arduino_hydroponics.php kuviossa (kuvio 4).

11 Tulokset ja yhteenveto

Opinnäytetyön tavoitteena oli ottaa selvää tarvittavista muuttujista ja määreistä, joita vesikasvatuksen hallinnassa tarvitaan ja soveltaa niitä käytännössä. Opinnäytetyön toiminnallisessa osuudessa rakennettiin erillisistä komponenteista IoT-laite, joka pystyi keräämään Arduino-alustaan liitetyistä sensoreista dataa ja lähettämään sitä verkossa sijaitsevaan tietokantaan.

Työlle suunniteltiin ja rakennettiin kytkentäkaavio, sekä sen kokoonpano tehtiin toisistaan erillisten komponenttien ympärille. Pääasiallisena ohjaimena valittiin Arduino Uno -alusta, jonka ympärille kaikki laitteen toiminta rakentui. Opinnäytetyössä vaaditut komponentit valittiin projektin kannalta tarpeellisten toimintojen toteuttamiseksi. Opinnäytetyön tapauksessa näiksi toiminnoiksi valittiin ravinneliuoksen koostumuksen mittaamisen ja hallinnan kannalta oleelliset lämpötila ja sähkönjohtavuus sekä sen pohjalta laskettava liukoisuus.

Järjestelmän ohjelmisto ja sen tarvitsemat yhteydet ja tekniikat rakennettiin teknisen toteutuksen mukaan. Arduino ohjelmoitiin käyttämällä Arduinon omaa ohjelmointikieltä ja siihen perustuvia kirjastoja, joilla laitteelle saatiin niiden tarvitsemat toiminnot. Yhteyksien käyttö perustui paljon ESP-01 WiFi moduulin toiminnallisuuksiin ja ominaisuuksiin. ESP-01 mahdollisti Arduinon yhdistämisen WiFi-verkkoon, josta se pystyi kommunikoimaan onnistuneesti etäpalvelimien kanssa, jotka pystyivät jatkokäsittelymään lähetettyä dataa.

Opinnäytetyöhön rakennettu Arduino-laite pystyi mittaamaan ravinneliuoksen tilan kannalta tärkeitä ja haluttuja muuttujia sekä pystyi ottamaan huomioon niiden ristivaikutuksia, kuten sähkönjohtavuuden muutosta lämpötilan mukaan. Arduino-laite pystyi onnistuneesti mittaamaan lämpötilaa, liukoisuutta ja sähkönjohtavuutta sekä lähettämään kerättyä dataa http POST-pyyntöillä verkossa olevalle palvelimelle ja tallentamaan datan tietokantaan palvelimella olevan PHP-tiedoston avulla. Lisäksi tietoa voidaan hakea tietokannasta toisen PHP-tiedoston avulla esimerkiksi käyttäjälle visualisoitavaksi

12 Pohdinta

12.1 Tavoitteet ja oppimistulokset

Opinnäytetyöni tavoitteena oli luoda laaja kokonaisuus erilaisia tietojenkäsittelyssä tarvittavia ja koulutuksessamme läpikäytyjä tekniikoita ja soveltaa niitä yhdessä kokonaisuudessa. Halusin liittää tekniikat myös hyödylliseen käyttötarkoitukseen, joka liittyy johonkin nousevaan kehitystrendiin, tässä tapauksessa vesiviljelyyn. Opinnäytetyö antoi hyvän kuvan siitä, minkälaisia tekniikoita muun muassa dataa keräävien IoT-laitteiden kehittäminen ja ylläpitäminen vaatii. Sain myös hyvän käsityksen siitä, minkälaisia tietokantoihin pohjautuvia ratkaisuja kyseisten laitteiden ympärille voidaan rakentaa.

Vesikasvatuksen hallinnan kannalta opinnäytetyön Arduino-toteutus kattaa todella hyvin opinnäytetyön tavoitteet. Vesikasvatusjärjestelmien eräänä etuna on järjestelmien hallittavuus. Toteutettu laite kykenee mittaamaan kasvatusalustan liukoisuutta, sähkönjohtavuutta, lämpötilaa, sekä vaikuttamaan sen muutokseen keräten samalla dataa tietokantaan. Tietokantoihin kerättyä dataa voidaan käyttää myös erilaisten analyysien ja visualisointien tekemiseen, minkä toteutettu `Arduino_hydroponics.php` -esimerkki osoittaa. Kerätty data tukee kasvatusta antamalla käyttäjälle selkeää tietoa ravinneliuoksen ja kasvatusastian tilasta reaaliajassa mahdollistaen kasvien ravinteiden ja kasvun parannetun hallinnan ilman erillisiä analyysejä.

Opinnäytetyö oli myös todella opettavainen. Opinnäytetyö sisälsi paljon minulle vierasta teknologiaa ja paljon sellaisia tekniikoita, joita halusin soveltaa käytännön toteutuksessa. Arduino oli laitealustana minulle suhteellisen vieras, enkä ollut tehnyt kuin vain pieniä ohjauksia laitteella. Arduinon toimintaperiaatteet ja ohjelmakoodi tulivat minulla paljon selkeämmiksi opinnäytetyön aikana ja sain todella paljon tietoa, kuinka tämän tyyliä IoT-järjestelmiä voidaan rakentaa. Opin myös paljon erillisten laitteiden ja tekniikoiden toisiinsa yhdistämisestä ja tiedonsiirrosta.

Opin myös toteuttamaan tietokantayhteyksiä ja http-pyyntöjä käytännössä. Aikaisemmin tietoliikenneyhteyksien ja ohjelmointirajapintojen käyttö on ollut minulle hieman epäselvää käytännössä, vaikka nämä ovat keskeisiä ohjelmoijien tarvitsemia taitoja IT-alalla, mutta niiden tekninen toteutus on ollut minulle aina hieman epäselvää. Lisäksi opin paljon verkkopohjaisesta kehityksestä ja PHP-ohjelmoinnista sekä verkkosivujen käytöstä tiedonsiirrossa ja vastaanottamisessa.

Työn tuloksista on kuitenkin muutamia asioita, jotka on hyvä huomioida työn tuloksia tutkiessa. En ole vesikasvatuksen tai viljelyn ammattilainen, mutta olen harrastanut vesikasvatusta muutaman vuoden. Tältä pohjalta olen kartuttanut tietämystäni aiheesta. Lisäksi toteuttamani IoT-laite on vielä prototyyppi ja sitä ei ole kokeiltu oikeissa kasvatusolosuhteissa. Näistä syistä johtuen en osaa vielä päätellä, kuinka toimiva toteuttamani ratkaisu todellisuudessa on. Uskon kuitenkin, että voin käyttää ja kehittää sitä tulevissa vesikasvatusprojekteissani.

Toteutettu laite tarvitsee jatkokehitystä ainakin koteloinnin osalta. Kotelointi ei kuitenkaan ollut opinnäytetyöni teknisen toteutuksen tai testaamisen kannalta oleellista. Opinnäytetyön kannalta oleellista oli saada kokoon valmis laite, joka pystyy keräämään ja lähettämään sensori dataa verkossa sijaitsevaan tietokantaan.

12.2 Kohdattuja ongelmia opinnäytetyössä

Opinnäytetyössä kohtasin monenlaisia ongelmia, varsinkin kehitysvaiheessa. Laitteiden yhdistäminen toisiinsa oli suhteessa helppoa, mutta varsinkin ESP-01 WiFi-moduulin käyttämisessä törmäsin moneen ongelmatilanteeseen. Suurin osa verkosta löytyvästä materiaalista ei ollut soveltuvaa Arduinon ja ESP-01 moduulin yhteiskäyttöön, vaan ne vaativat yleensä joko Arduinon lisättävän WiFi tai Ethernet-kilven tai itsenäisen NodeMCU ESP8266 piirin. Tästä syystä suurin osa kirjastoista ja ohjeista oli yhteensopimatonta käyttämäni laitteen kanssa ja toimivan ratkaisun löytäminen vei paljon aikaa. Tästä johtuen kehityksessä tuli

monenlaisia takapakkeja ja kehityksen suunta muuttui monta kertaa opinnäytetyön aikana, sillä minulla oli vaikeuksia löytää tietoa, kuinka saisin ohjattua dataa verkkoon moduulia käyttäen. Eräänä alkuperäisenä suunnitelmana oli lähettää dataa suoraan MySQL-tietokantaan, mutta se ei onnistunut mainitun WiFi-kilven puuttuessa. Tämä aiheutti työssä viivästyksiä ja jopa viikkojen pysähdyksiä kehityksessä, sillä joudin hakemaan jokaisen epäonnistuneen yrityksen jälkeen aina uusia vaihtoehtoja. Lopulta löysin kuitenkin tavan lähettää tietoa http POST-pyyntöillä, joka toimi projektin kannalta todella hyvin.

Jos aloittaisin kehityksen uudelleen nyt, tutustuisin laitteistoon ja sen yhteyksiin sekä kirjastoihin paremmin ennen kehityksen aloittamista, jotta tällaisilta ongelmilta vältyttäisiin. Toisaalta opin paljon näiden ongelmien takia ja järjestelmä toimii nyt halutulla tavalla, joten olen tyytyväinen opinnäytetyön tulokseen.

12.3 Mihin opinnäytetyön toteutusta voidaan käyttää

Laitteen pääasiallisena tarkoituksena on pitää kasvatusalustan ravinnetasapaino tasaisena ravinneliuosten vaihtojen välissä. Laitetta tullaan käyttämään niin että ravinneliuoksille lasketaan lisättyjen ravinnemäärien mukaan optimaalinen seos, johon ravinteiden määrää tullaan vertaamaan. Tätä arvoa käytetään automaattisen ravinneliuoksen lisäämisen ja ravinneliuoksen koostumuksen seurantaan ja ravinneliuoksen koostumus pyritään pitämään sen tasolla. Tulevaisuudessa tätä arvoa voitaisiin myös automatisoida ja optimoida kasvien ravinteiden tarpeen mukaan ja arvoa voitaisiin muuttaa kasvuvaiheen mukaan.

Laitteella kerättyä dataa voidaan käyttää myös erilaisten ennusteiden ja analyysien tekemiseen. Esimerkkinä tällaisesta analyysistä voi olla vaikkapa ravinteiden kulutuksen muutosten seuranta esimerkiksi lämpötilojen tai säätilan mukaan. Analyysien tekeminen oli kuitenkin tämän työn kokonaisuuden, laajuuden ja tarkoituksen ulkopuolella, eikä niihin otettu tästä syystä enempää

kantaa tässä opinnäytetyössä. Siitä huolimatta on kuitenkin tärkeää ottaa huomioon myös minkälaisia hyötyjä ja käyttötarkoituksia tämäntyyppisellä laitteella voidaan tai voitaisiin tulevaisuudessa saada.

Lähteet

ABC Catalyst. 2020. Growing Tomatoes Indoors With 94% Less Water And No Soil. ABC Science.

<https://www.youtube.com/watch?v=5Fq6PQI7fr8> 27.1.2021.

Ahmed, R. 2017. The Impact of TDS & pH on Quality of Drinking Water. BlueEast. Medium. <https://medium.com/blueeast/tds-ph-and-their-impact-on-quality-of-drinking-water-68b2a7433043> 15.3.2021.

Arduino. 2018. What is Arduino?

<https://www.arduino.cc/en/Guide/Introduction> 15.3.2021.

Barbosa, G. Gadelha, F. Kablik, N. Proctor, A. Reichelm, L. Weissinger, E. Wohlleb, G. Halden, R. 2015. Comparison of Land, Water, and Energy Requirements of Lettuce Grown Using Hydroponic vs. Conventional Agricultural Methods. Environmental Research and Public Health. MDPI.

<https://www.mdpi.com/1660-4601/12/6/6879/htm#> 27.1.2021.

Damantas, G. 2019, What is HTML? The Basics of Hypertext Markup Language Explained. Hostinger.

<https://www.hostinger.com/tutorials/what-is-html> 7.5.2021.

Fabacademy. 2015. ESP8266 Introduction

<http://fabacademy.org/archives/2015/doc/networking-esp8266.html>. 15.3.2021.

Goldsher, E. 2019. Greenhouse production news

<https://gpnmag.com/article/hydroponics-revolutionizing-greenhouse-growing/>
1.2.2021.

Harris, J. 2019. Cost and Profitability of Hydroponic Farming. AmHydro

<https://www.youtube.com/watch?v=mIXh0vyvZfQ> 27.1.2021.

Jones, J. 2005. Hydroponics: A Practical Guide for the Soilless Grower. CRC Press.

https://books.google.fi/books?hl=fi&lr=&id=y_bKBQAAQBAJ&oi=fnd&pg=PP1&dq=hydroponics&ots=kAPFQUqOiG&sig=pilNNhvfZHPGCV0wRXa7F2rhRVY&edir_esc=y#v=onepage&q&f=false. 4.1.2021.

Lauremson, J. 2019. Could high-tech Netherlands-style farming feed the world? DW.

<https://www.dw.com/en/could-high-tech-netherlands-style-farming-feed-the-world/a-47105412> 1.2.2021.

Nemali, K. 2018. Details of Electrical Conductivity Measurements in Greenhouse Production. Purdue Extension

<https://www.extension.purdue.edu/extmedia/HO/HO-286-w.pdf> 4.1.2021.

PHP. 2021. What is PHP?

<https://www.php.net/manual/en/intro-what-is.php> 7.5.2021.

Shoukat, A. Shoukat Hussain, M. Shoukat, A. 2020 Effects of Temperature on Total dissolved Solid in water. ResearchGate

https://www.researchgate.net/publication/339510087_Effects_of_Temperature_on_Total_dissolved_Solid_in_water 15.3.2021.

SQLCourse. 2021. What is SQL?

<http://www.sqlcourse.com/intro.html> 7.5.2021.

UCANR. 2019. Glasshouse Vegetable Production in the Netherlands, Part 1

<https://youtu.be/nM8Qz-fzJ6M> 1.2.2021.

Venture City. 2019. Why the Future of Farming is in Cities - The Big Money in Vertical Farming. Venture City

<https://www.youtube.com/watch?v=LiNI-JUFtsA> 29.1.2021.

Viviano, F. 2019 THIS TINY COUNTRY FEEDS THE WORLD.
Nationalgeographic.

<https://www.nationalgeographic.com/magazine/2017/09/holland-agriculture-sustainable-farming/> 1.2.2021.

W3Schools. 2021. What is HTTP?

https://www.w3schools.com/whatis/whatis_http.asp 7.5.2021.