

**TUOTEKEHITYSPROSESSIEN
SOVELTAMINEN
KASVIENKASTELUJÄRJESTELMÄN
SUUNNITTELUSSA**

Tiivistelmä

Tekijä Salim, Dlovan	Julkaisun laji Opinnäytetyö, AMK Sivumäärä 36	Valmistumisaika Kevät 2020
Työn nimi Tuotekehitysprosessien soveltaminen kasvienkastelujärjestelmän suunnittelussa		
Tutkinto Konetekniikka, Insinööri (AMK)		
Tiivistelmä <p>Opinnäytetyössä suunniteltiin kasvienkastelujärjestelmä soveltamalla tuotekehitysprosessien elementtejä. Oppimistavoitteena oli ymmärtää tuotekehitysprosessien ideoiden ja konseptien luontivaihetta ja luoda yksinkertaistettu prosessimalli käytännön projektia varten.</p> <p>Opinnäytetyön tavoitteena oli selkeän rakenteen omaavan yksinkertaistetun tuoteidean luontiprosessin käyttäminen kasvienkastelujärjestelmän suunnittelussa, jonka tuloksena syntyi fyysinen prototyyppi sekä 3D-mallinnus. Tuotteesta voidaan jatkokehittää kuluttajamarkkinoille automaattinen kotikasvien kastelujärjestelmä, jonka toiminta-aika on kahdesta neljään viikkoon.</p> <p>Tuotekehitysteoriassa keskitytään kahden amerikkalaiseen konetekniikan insinöörien, Karl T. Ulrichin ja Steven D. Eppingerin teokseen aiheesta. Heidän menetelmänsä valittiin sen laajan sisällön sekä monialaisen näkökulman vuoksi. Teoriaosuuden jälkeen siirryttiin käytäntöön, jossa olennaisimmat työkalut olivat Arduino-mikrokontrolleri ja 3D-mallinnuksen hyödyntäminen suunnittelussa.</p> <p>Opinnäytetyön pääasialliset ongelmat liittyivät kustannushintojen alentamiseen, suunnittelukokemuksen puutteeseen ja isomman työryhmän tarpeeseen varmistamaan suunnittelutyön laatu.</p> <p>Lopussa todetaan yksinkertaistetun prosessimallin olevan hyvin supistettu, mutta toimiva vaihtoehto idean luontivaiheeseen. Tuotteen lopputuloksena on toimiva kastelusysteemi, jota voidaan käyttää markkinatarpeen arvioimisessa.</p>		
Asiasanat Tuotesuunnittelu, tuotekehitys, kastelujärjestelmä, tuoteidea		

Abstract

Author Salim, Dlovan	Type of publication Bachelor's thesis	Published Spring 2020
	Number of pages 36	
Title of publication Applying product development processes in designing plant irrigation system		
Name of Degree Bachelor's Degree Programme in Mechanical Engineering		
Abstract <p>The objective of this bachelor's thesis was to design a plant irrigation system by applying elements of product design and development processes. The learning goal was to understand the idea creation phase of product development and to create a simplified pipeline for a practical project.</p> <p>The goal of the thesis was to apply a product development process with a clear structure for a plant watering system, and to demonstrate this by building a physical prototype and a 3D model. This can then be further developed to provide the consumer market with an automatic irrigation system for indoor plants with an operating timeframe of two to four weeks.</p> <p>The theoretical view on product development and design focused on the work of the mechanical engineers Karl T. Ulrich and Steven D. Eppinger. Their product development method was chosen due to its broad content and multidisciplinary perspective. After these theoretical considerations, the simplified process has been applied to the practical project, in which an Arduino microcontroller and the utilization of 3D modeling have been the most relevant tools.</p> <p>The main problems were related to lowering production costs, lack of design experience and the need for a larger work group to guarantee the quality of design and iterate of ideas in each phase.</p> <p>In the end, the simplified product development process, in spite of being highly reduced, is effective when implementing it to the idea creation phase. The resulting product idea was a functional watering system, which can be used to assess market needs.</p>		
Keywords product design, product development, plant irrigation, idea generation		

SISÄLLYS

1	JOHDANTO	1
2	TUOTEKEHITYS	2
2.1	Tuotekehityksen prosessi	2
2.2	Yksinkertaistettu prosessi	3
2.2	Prosessin vaiheet	4
3	KASVIENKASTELUJÄRJESTELMÄN SUUNNITTELU	7
3.1	Ideointi ja tarve	7
3.2	Määrittely	9
3.3	Luova työ ja suunnittelu	10
3.3.1	Idea 1: KV-Pi	14
3.3.2	Idea 2: SV-Logic	15
3.3.3	Idea 3: VS-A	16
3.3.4	Luonnosten arviointi ja analysointi	17
3.3.5	Perusteet pisteytyksille	17
3.4	Systeemisuunnittelu	19
3.5	Prototyyppi	20
3.6	Yksityiskohtainen suunnittelu	26
3.7	Jatkokehitys	34
4	YHTEENVETO	35
	LÄHTEET	36
	LIITTEET	37

1 JOHDANTO

Tuotekehitysprosesseja on erilaisia, mutta niissä toistuvat samat elementit. Opinnäytetyössä tutkitaan ensin näitä elementtejä ottamalla niistä selvää aihetta käsittelevän kirjallisuuden avulla ja luomalla niistä yksinkertaisempaa mallia, jonka avulla voidaan luoda tuoteidea. Tuotetta aletaan luomaan käytännön kautta seuraamalla yksinkertaistetun prosessin tavoitteita sekä tehtäviä. Tavoitteena on esittelykelpoinen malli tuoteideasta, jonka avulla kyettäisiin mahdollisesti saamaan kiinnostusta idealle sekä toimiva yksinkertaistettu prosessimalli tuoteideoiden kehittämiseen.

Opinnäytetyö on itsenäisesti toteutettu projekti, jonka idea syntyi tarpeesta laitteelle. Kasvien kasteluun löytyy markkinoilta monia vaihtoehtoja, mutta ne eivät täytä kaikkia asiakastarpeita ja toiveita kategoriassaan. Tuotekehitys onkin usein jo olemassa olevien tuotteiden parantelemista tai muuttamista.

Opinnäytetyössä tuotekehitysprosessia on rajattu niin kutsuttuun innovaatioprosessiin, jossa keskitytään idean luomiseen, eikä edetä ideoinnin tasolta pidemmälle. Myös suunnittelumenetelmiä on rajattu yksinkertaisempaan muotoon nopeiden tuloksien saamiseksi. Tuotekehityksestä löytyy laaja määrä kirjallisuutta sekä erilaisia menetelmiä ja niistä keskitytään pääasiassa Karl T. Ulrichin ja Steven D. Eppingerin kehittämään menetelmään aiheesta.

Opinnäytetyö alkaa teoriaosuudella, jota tutkimalla tuotetaan kasvienkastelujärjestelmän kehittämistä varten yksinkertaistettu prosessi. Prosessin määrittämisen jälkeen siirrytään sen toteuttamiseen käytännössä. Lopuksi käsitellään tuotteen jatkokehityksen kannalta olennaisimmat vaiheet.

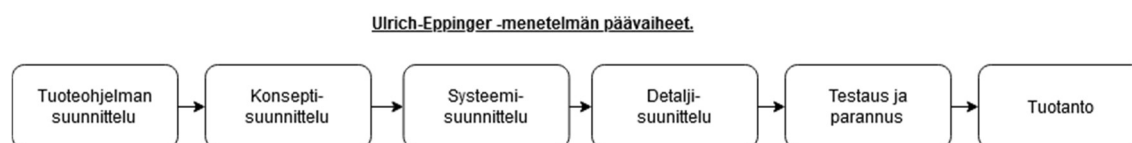
Aiheen tuotekehitystä voi suositella kaikille konetekniikan opiskelijoille sekä ammattilaisille, sillä se antaa työkalut luovuuden kehittämiseen sekä analyttiseen ja prosessimaiseen työntekoon. On myös hyödyllistä tutkia tuotekehitysprosesseissa toistuvia elementtejä, sillä niitä löytää työelämässäkin yritysten soveltamina omaan kehitystoimintaansa.

2 TUOTEKEHITYS

2.1 Tuotekehityksen prosessi

Tuotekehityksen teoria on hyvin laaja käsite, johon kuuluu iso kirjo erilaisia menetelmiä sekä tapoja ja siitä on paljon kirjallisuutta. Kaikissa kuitenkin toistuu samankaltaiset elementit, koska tuotekehitystoiminta on usein prosessimaista vaiheiden läpikäyntiä, ja tämä periaate keksittiin sekä kehitettiin jo 1960-luvulla NASA:ssa, minkä jälkeen onkin kehitetty monia eri prosessimalleja (Verworn 2005, 13). Tässä opinnäytetyössä ei kuitenkaan käydä teoriaa läpi syvällisesti, vaan keskitytään tuotekehityksessä osaan, jonka tarkoitus on luoda uusi tuoteidea. Tätä prosessia kutsutaankin enemmän innovaatioprosessiksi, kuin tuotekehitysprosessiksi, sillä tuotekehitysprosessiin kuuluu nykyään enemmän, kuin vain tuoteidean luominen ja kehittäminen (Hietikko 2008, 41), mutta aihe kuuluu kuitenkin tuotekehitysprosessiin ja kirjallisuudessa sitä edelleen kutsutaan siksi.

Tätä opinnäytetyötä varten on tutkittu Ulrich-Eppinger-menetelmää, joka perustuu Karl T. Ulrichin ja Steven D. Eppingerin kirjoittamaan kirjaan aiheesta. Teoksessa käsitellään ai-
hetta yksityiskohtaisesti ja laajasti alkaen tuotteen tarpeesta ja keksimisestä tuotannon käynnistämiseen asti. Menetelmän eri vaiheet näkyvät kuviossa 1.

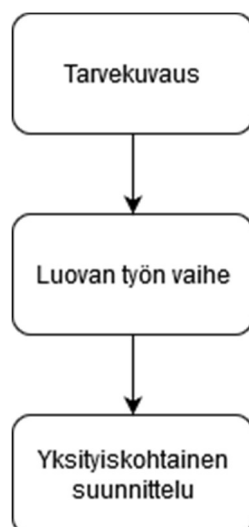


Kuvio 1. Ulrich-Eppinger-menetelmän vaiheet (Hietikko 2008, 42)

Kuvio 1 näyttää, kuinka projektissa tavallisesti edetään, mutta usein sen aikana joudutaan myös palaamaan edellisiin vaiheisiin, eli prosessissa on "takaisinkytkentöjä" odottamattomien ongelmien tai parannusideoiden syntyessä. On myös muita prosessimalleja, kuten spiraalinen työnkulku, jossa vaiheet kierretään useaan kertaan, kunnes tuote on hiottu lopulliseen muotoonsa. (Hietikko 2008, 41.)

Ulrich-Eppinger-menetelmä koostuu koko tuotekehitysprosessista, johon se tarjoaa monia eri työkaluja eri vaiheiden suorittamiseen. Tässä opinnäytetyössä käytetään kolmea vai-
hetta koko prosessista: 1. Konseptisuunnittelu, jossa tunnistetaan tarve ja kirjataan tuotteen tavoitteet, 2. Systeemis suunnittelu, jossa muun muassa mietitään tuotteen toimintaa, 3. Detaljisuunnittelu, jossa tulokset kerätään yhteen ja tuote suunnitellaan yksityiskohtaisesti (Hietikko 2008, 43).

Ulrich-Eppinger-menetelmästä valitut vaiheet sisältävät myös itsessään laajan määrän kä-
sitteitä, joten ne voidaan esittää yksinkertaisemmin kuvion 2 mukaisena prosessina.



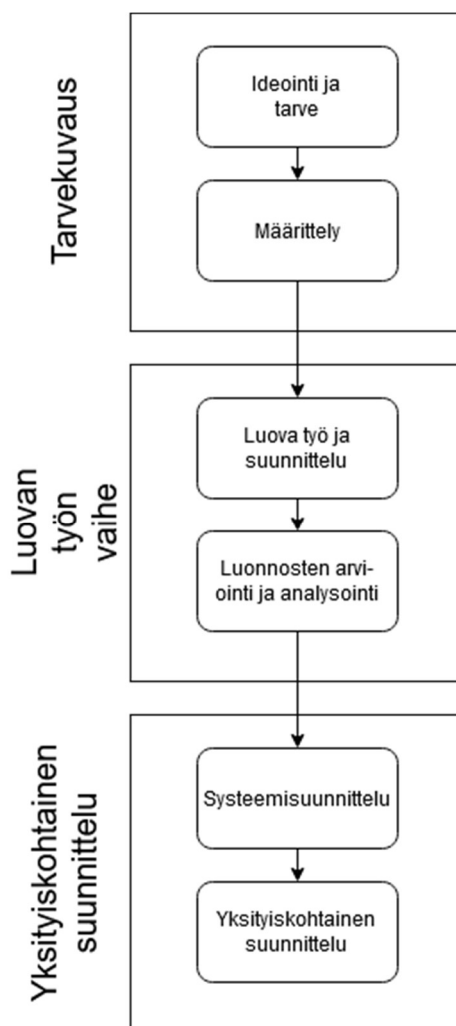
Kuvio 2. Yksinkertaisempi prosessikuvaus innovaatiokehityksestä

Kuvio 2 näyttää selkeämmin kolmen valitun vaiheen tarkoitukset ja tavoitteet, joita tarvitaan tässä opinnäytetyössä. Tarvekuvaus voi olla asiakkaan tarpeiden täyttäminen tai esimerkiksi markkinoiden tutkimisen jälkeen huomattu mahdollisuus luoda uusi tuote helpottamaan elämää. Kun tarve tai idea on määritelty, siirrytään luovan työn vaiheeseen, jossa tehdään luovaa työtä idean kehittämiseen ja ongelmien ratkaisemiseksi. Tähän vaiheeseen kuuluu muun muassa aivoriihet, luonnostelu ja jo olemassa olevien tuotteiden tutkiminen. Lopuksi suoritetaan yksityiskohtainen suunnittelu, jossa luovan työn vaiheen ideoita kehitetään pidemmälle. Jokaiseen vaiheeseen voidaan soveltaa iso kirjo menetelmiä niiden suorittamiseksi; esimerkiksi tarvekuvaus voi olla kyselylomakkeella suoritettu tutkimus ja luovan työn vaihe voi olla systemaattista olemassa olevien ratkaisujen yhdistämistä tai kokonaan uusien menetelmien tutkimista.

Kolme vaihetta eivät kuitenkaan yksinään riitä ohjaamaan innovaatioprosessia, vaan vaiheet ja niiden tavoitteet täytyy määritellä tarkemmin. Seuraavassa osassa käydään läpi ja luodaan Ulrich-Eppinger-menetelmään pohjautuvaa yksinkertaisempaa prosessia.

2.2 Yksinkertaistettu prosessi

Kuvion 2 esittämien vaiheiden sisään voidaan lisätä tehtäviä, joiden avulla päästään jokaisen vaiheen tavoitteisiin. Kuviossa 3 on lisätty jokaisen vaiheen sisälle kaksi tehtävää toteuttamisjärjestyksessä. Tässä prosessissa ei ole takaisinkytkentöjä – vaikka yleensä niitä on – koska opinnäytetyö on haluttu pitää selkeänä ja lineaarisena. Takaisinkytkennät ovat luonnollisia, sillä kaikkia määritelmiä ei aina voida täyttää ja uusia ideoita ja ongelmia ei voi välttämättä ennakoida.



Kuvio 3. Yksinkertaistettujen prosessivaiheiden sisäiset tehtävät

Kuvion 3 tehtävät suositeltaisiin tekemään pienessä ryhmässä, sillä hyvän tuotteen luominen vaatii ryhmän, jolla on erilaiset näkemykset, taidot ja luonteet. Näitä ominaisuuksia tarvitaan näkökulmien laajentamiseksi. (Ulrich & Eppinger 2012, 6 - 8.) Yhden henkilön suunnitellessa tuotetta laaja näkökulma puuttuu sekä itsekriittisyys voi jäädä vähemmälle; omia virheitä ei välttämättä huomaa helposti, vaikka virhe olisikin ilmeinen. Tosin yksittäisten ongelmien ratkominen yksin tuottaa enemmän ja parempia ratkaisuja kuin jos samaa ongelmaa ratkaistaisiin ryhmässä (Ulrich & Eppinger 2012, 128). Prosessia voidaan kuitenkin käyttää alustavan idean luomiseen ja kehittämiseen yksin, jolloin lopputulosta voidaan käyttää markkinatarpeen arvioimisessa.

2.2 Prosessin vaiheet

Prosessi alkaa tarvekuvaus-osiosta. Aluksi syntyy idea, kun esimerkiksi huomataan jokin tarve uudelle ratkaisulle tai parannukselle olemassa olevaan tuotteeseen. Tässä

vaiheessa ensimmäinen tehtävä on idean ja tarpeen kartoittaminen paperille, jonka tarkoituksena on konkretisoida ja demonstroida idea ryhmälle tai asiakkaalle.

Kun idea on alustavasti paperilla, siirrytään määrittelyvaiheeseen. Määrittelyvaiheessa tehtävänä on luoda tuotteelle mitattavissa olevat vaatimukset. Määrittelyt tulevat alkupe-
räisestä ideasta ja tarpeesta, mutta niiden tulee olla mitattavia ja vertailtavia suureita. Esimerkiksi sana "Hieno" ei riitä määritelmäksi ja vaatimukseksi itsessään, mutta sille voidaan asettaa mitattavissa oleva asteikko muun muassa kyselytutkimuksen avulla. Määritelmät antavat suunnittelulle lähtökohdat ja asettavat sen rajat.

Idealle on tässä vaiheessa asetettu lähtökohdat ja rajat, mistä siirrytään niiden soveltamiseen luovan työn vaiheessa. Vaiheen tarkoitus on luoda alustavia hahmotelmia laitteesta ja ratkaisuja sen toimintojen toteuttamiseen käyttämällä määritelmien antamia tietoja, luovuutta kuitenkin rajoittamatta. Luovaan työhön kuuluu positiivinen ja avoin asenne uusia ideoita kohtaan sekä rakentava kritiikki (Hietikko 2008, 85). Liiallinen tekninen ajattelu ja kova kritiikki sivuuttavat paljon uusia ideoita, jotka saattavat ensinäkemältä olla soveluttomampia, mutta kehitettäessä hyviä ja uniikkeja ratkaisuja.

Luovaan kehittämiseen löytyy paljon erilaisia menetelmiä sekä työkaluja fyysisistä aktiviteeteistä systemaattiseen taulukkojen käyttämiseen. Kilpailijoiden samassa kategoriassa olevien tuotteiden tutkiminen on myös hyvä menetelmä kehittää omaa ideaa, ja siihen sisältyy State-Of-The-Art-tuotteen löytäminen ja tarkasteleminen. State-Of-The-Art-tuotteella tarkoitetaan tuotekategoriansa parasta markkinoilla olevaa tuotetta. Tässä sanalla "paras" tarkoitetaan menestyneintä, sillä myös kuluttajamarkkinoilla ei tarkastella ainoastaan kehittyneintä teknologiaa, vaan menestys voi johtua esimerkiksi helppokäyttöisyydestä tai esteettisestä muotoilusta. Kaiken tämän taustatutkimuksen ja ideoiden syntymisen jälkeen päästään luovan työn ytimeen eli luonnosteluun. Luonnostelun, eli piirrosten tekemisen, tarkoitus on visualisoida ideat ja ajatukset paperille. Tässä vaiheessa ideoita ja ratkaisuja yhdistellään tuotekokonaisuuksiksi ja lopuksi luonnoksia analysoidaan ja verrataan toisiinsa. Näistä voidaan valita yksi tai useampi seuraavaan vaiheeseen, jossa niille suunnitellaan toimiva systeemi ja kehittämistä viedään eteenpäin.

Systeemisuunnittelussa suunnitellaan nimensä mukaisesti teknisesti toimiva kokonaisuus. Tämän tehtävän tarkoitus on pistää suunnittelija miettimään ja esittämään, mitä eri toimintoja laitteessa on, mitä ne tekevät ja miten ne toimivat. Systeemisuunnittelu on kirjallisuudessa osa tuotteen arkkitehtuuria, jossa suunnitellaan tuotteen systeemi ja jaetaan osat alueet moduuleihin, joissa mietitään, miten laitteen eri osat toimivat keskenään sekä osakokonaisuuksia ja niihin liittyviä tietoja (Hietikko 2008, 111). Systeemisuunnittelun jälkeen voidaan testata systeemin toimivuutta toiminnallisella prototyypillä.

Toiminnallisen prototyypin tarkoitus on testata systeemis suunnittelussa määritellyjä toimintoja. Prototyyppejä on muihinkin tarkoituksiin, ja niitä voi tehdä missä tahansa kohtaa koko tuotekehitysprosessin aikana, kun halutaan testata jotakin ideaa tai ratkaisua. Toiminnallisten prototyyppien lisäksi voi tehdä nopeita fyysisiä malleja, jotka ovat alkuvaiheessa tehtäviä havainnointimalleja esimerkiksi pahvista tai puusta muotoiltuna. Nopeita fyysisiä malleja on hyvä käyttää alkuvaiheessa palautteen keräämiseen. (Ulrich & Eppinger 2012, 172).

Lopuksi, kun paperilla on idea ja ideasta luotu toimiva systeemi, siirrytään yksityiskohtaiseen suunnitteluun. Yksityiskohtien hiomisen tavoitteena on komponenttien valinta, niiden asettelu ja tuotteen muodon luominen. Lopputuloksena pitäisi olla kokonainen ja esittelykelpoinen tuote, josta päästäisiin tekemään lopputuotteen prototyyppi testaukseen sekä markkinatarpeen arvioimiseen.

Tämän prosessin ulkopuolella, mutta seuraavana vaiheena on tuotteen jatkokehittäminen valmistuskelpoiseksi kokonaisuudeksi.

3 KASVIENKASTELUJÄRJESTELMÄN SUUNNITTELU

3.1 Ideointi ja tarve

Tässä projektissa, jossa suunnitellaan kasvienkastelua varten kotikone, käytetään opinäytetyön edellisessä osiossa määriteltyä yksinkertaista prosessia. Tarkoituksena on toteuttaa jokainen prosessivaihe käytännön kautta. Tämä tuoteidea ei ole uusi, ja markkinoilla onkin paljon valmiita vaihtoehtoja ja internetissä harrastelijoiden ohjeita oman kasvienkastelujärjestelmän rakentamiseen, jotka olivat tämänkin projektin perusta. Erona harrastelijoiden vaihtoehdoille on prosessimainen ja systemaattinen suunnittelu sekä esittelykelpoisen kokonaisuuden luominen.

Idea tuli suoraan asiakastarpeesta. Matkalle lähdessä oleva tai mökille viikonlopuksi muuttava henkilö todennäköisesti omistaa kotikasveja, jotka vaativat säännöllistä kastelua. Tuotetta ei tarvitse välttämättä omistaa itse, mutta sen sijaan, että ystävä tulee kastelemaan säännöllisesti, hän voi lainata omaa kastelujärjestelmäänsä.

Alkuperäinen kuvaus tuotteesta on esteettisesti nykyaikaisen kotikoneen näköinen laite, joka mahtuu ruokapöydälle tai ikkunalaudalle ja kastelee kasvit kahden viikon tai kokonaisen kuukauden ajan itsenäisesti. Laitteeseen ei toivota mitään ylimääräistä, sillä sen halutaan olevan mahdollisimman yksinkertainen ja helppokäyttöinen. Näiden ominaisuuksien toivotaan nopeuttavan käyttöönottoa. Asiakas ilmaisi, että markkinoilla olevat tuotteet ovat kalliita ja ne kuuluvat suurempiin kasvihuonejärjestelmiin, monimutkaisia ja hitaita asentaa tai epäluotettavia kastelupalloja, jotka perustuvat tasaiseen ja hitaaseen veden tippumiseen ottamatta huomioon kasvien vesimäärätarvetta. Tämän lisäksi olisi suotavaa, jos laitteen voi rikkoutuessa korjata itse ostamalla komponentit kuluttajille saatavilla olevilla vaihto-osilla.

Tarpeet kirjataan ylös tekemällä lyhyt raportti tai esitelmä, jonka tarkoituksena on visualisoida konkreettinen idea. Mission Statement -taulukolla, eli tehtävänantoa kuvaavalla taulukolla saadaan helposti ja nopeasti idean kuvaus, tavoitteet ja muut tiedot paperille ja sen tarkoitus on antaa kokonaiskuva projektin lähtökohdista. (Hietikko 2008, 45.) Taulukossa näkyy myös sidosryhmät eli kuka tulee olemaan lopputuotteen kanssa tekemisissä. Tämän projektin Mission Statement -taulukko näkyy taulukossa 1.

Näin prosessin varhaisessa vaiheessa voidaan myös luoda nopeita ja alustavia luonnoskuvia paperille. Kun ajatuksia ja ideoita syntyy mieleen, on hyödyllistä visualisoida ne piirtämällä. Lisäämällä avainsanoja piirustuksiin ideat palautuvat mieleen helpommin.

Taulukko 1. Tehtävänantoa kuvaava Mission Statement -taulukko

Mission Statement: Kasvienkastelujärjestelmä
Tuotteen kuvaus: <ul style="list-style-type: none"> - Yksinkertainen ja automaattinen kotikasvien kastelujärjestelmä lyhyiden matkojen ajaksi
Keskeiset tavoitteet: <ul style="list-style-type: none"> - Kustannushinta alle 100 euroa - Prototyyppi valmis ennen 2020 kesäkuuta
Päämarkkinat: <ul style="list-style-type: none"> - Kuluttajat, usein matkustelevat
Toissijaiset markkinat: <ul style="list-style-type: none"> - Kasvikauppa-yrittäjät - Puutarhurit
Oletukset ja rajat: <ul style="list-style-type: none"> - Kuluttajille saatavilla olevat komponentit tuotteeseen - Vähintään neljä kasvia kasteleva laite - Toimii kaksi viikkoa/kuukauden itsenäisesti - Nopea käyttöönotto
Sidosryhmä: <ul style="list-style-type: none"> - Loppukäyttäjät - Suunnittelija/tekijät

Taulukon 1 kohdat on täytetty asiakkaan kanssa, ja se on niin kutsutulla ”asiakkaan kielellä” kirjoitettu. Asiakkaan kielellä tarkoitetaan epätarkkaa määritelmää, jota ei voida sellaisenaan käyttää suunnittelemiseen. Seuraavaksi muunnetaan taulukon tieto tuotekehityksen kannalta olennaiseksi taulukoksi, jossa asetetaan tarkemmat määritelmät.

3.2 Määrittely

Kun tehtävänanto on annettu, sen pohjalta on tehtävä lista määritelmistä, joita voidaan sitten mitata. Määritelmät ovat alussa suuntaa antavia, mutta ne tulisi yrittää täyttää mahdollisimman hyvin. Usein osa jää täyttämättä, mutta vastaavasti odotukset voidaan joskus jopa ylittää. Tähän vaiheeseen prosessia usein palataan jälkikäteen korjaamaan määritelmiä (Ulrich & Eppinger 2012, 94).

Taulukko 2. Määritelmät

Määritelmä	Suure	Arvo
Kompakti	Tilavuus (m ³)	< 0,064 m ³
Automaattinen kastelu-aika	Aika (vk)	> 2 vk
Helppokäyttöinen/Yksinkertainen	Painikkeiden/säätökytkimien määrä (kpl)	< 3 kpl
Nopea asentaa toimintakuntoon	Aika (s)	< 30 s
Edullinen	Hinta (€)	< 30 €
Kevyt	Paino (kg)	< 5 kg
Monen kasvin kastelu	Kasteluputkien määrä (kpl)	> 4 kpl
Korjaamisen mahdollisuus	Vaihdettavat osakokonaisuudet (%)	> 70 %

Taulukossa 2 korjaamisen mahdollisuuden kohdalla vaihdettavilla osakokonaisuuksilla tarkoitetaan prosentuaalista määrää osista, jotka voidaan vaihtaa ostamalla osakokonaisuus valmiina kuluttajamarkkinoilta. Esimerkiksi jos laitetta ohjaava piirikorttia ei löydy valmiina markkinoilta, se täytyy tehdä itse ja ohjauskokonaisuus jää prosentuaalisesti pois.

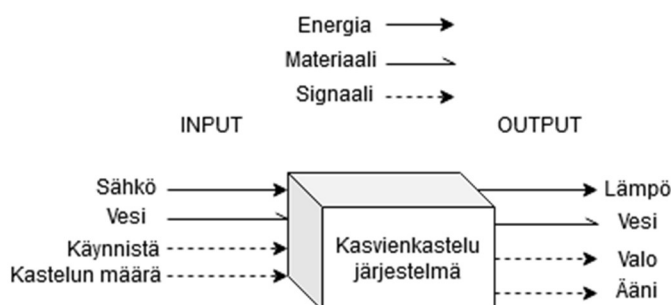
Taulukon 2 ollessa valmis voidaan siirtyä laitteen luomiseen ja kehittämiseen. Määritelmäosio voi rajoittaa hieman luovuutta, mutta sen on samalla tarkoitus toimia pohjana ideoinnille rajaamalla loputonta määrää mahdollisia ratkaisuja.

3.3 Luova työ ja suunnittelu

Internettiä selaamalla ja tutkimalla on helpoin tapa saada inspiraatiota ja ideoita nopeasti. Hakukoneella etsiessä kastelujärjestelmiä löytyy useita kotitekoisia sekä tuotekehitettyjä malleja. Laitteiden lisäksi löytyy monia menetelmiä kastella kasveja kotona, mutta yksi ongelma kuitenkin toistui eri laitteissa – aikaa vievä käyttöönotto. Toinen yleinen ongelma näytti olevan monimutkaisuus, joka johtui laitteen säätämisen tarpeesta. Tässä päästiin heti olemassa olevien tuotteiden tutkimisen perimmäiseen tarkoitukseen; olemassa olevista tiedoista ja tuotteista etsitään ongelmia ja tuotetaan uusia ideoita ja ratkaisuja (Hietikko 2008, 85). Internetistä saatu inspiraatio, ratkaisut ja ongelmat on hyvä kirjoittaa tai hahmotella muistivihkoon myöhempää aivoriihiä varten. Ensin on kuitenkin luotava kokonaiskuva kastelujärjestelmästä.

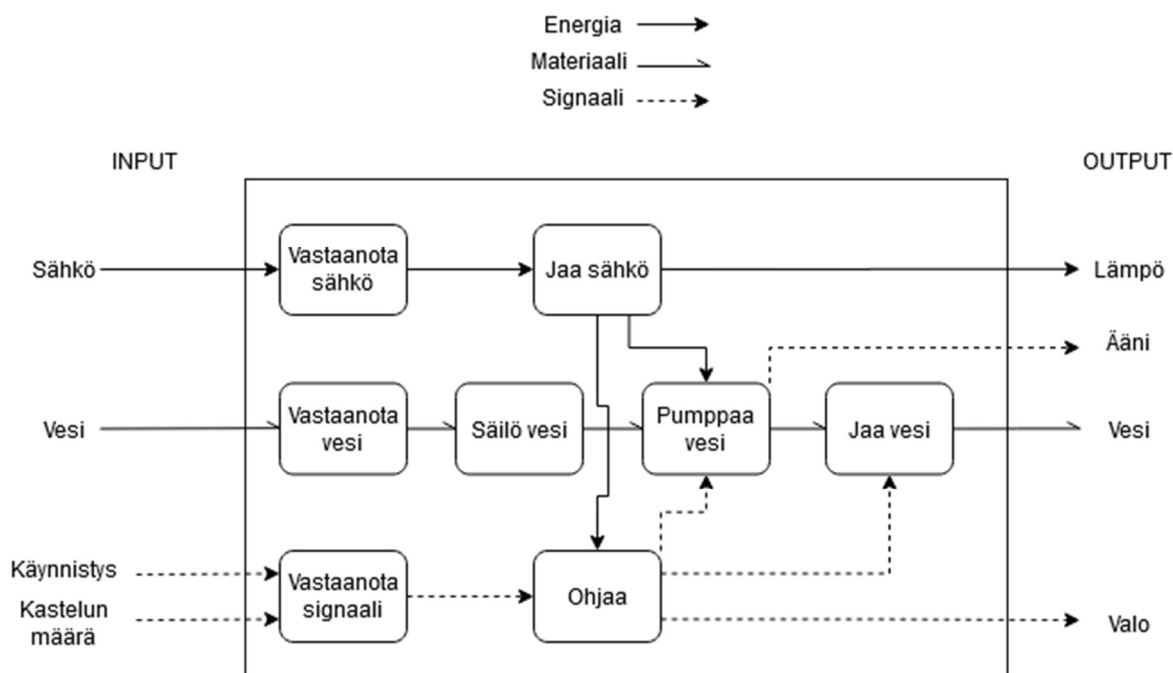
Kastelujärjestelmien tutkimisen jälkeen on hahmottunut kuva toimivista, jo olemassa olevista laitteista. Nämä tiedot riittävät luovan työn aloittamiseen luomalla abstrakti ja looginen kuva laitteesta Function Diagram -hahmotelman avulla. Tämä hahmotelma auttaa laitteen luomisessa tekemällä siitä yksinkertaisen abstraktin kokonaisuuden ilman tarvetta tietää siitä vielä kaikkea. Menetelmän perimmäinen tarkoitus on rikkoa isot ongelmat pienempiin ongelmiin, jotta voidaan helpommin hahmottaa eri osa-alueet ja keskittyä niihin erikseen. (Ulrich & Eppinger 2012, 121.)

Function Diagram -hahmotelman ensimmäinen vaihe on luoda kuviossa 4 oleva ”musta laatikko”, joka edustaa lopullista laitetta. Sen sisäiset toiminnot eivät ole olennaisia tässä vaiheessa, vaan mietitään sen ulkopuolelta tulevia asioita, joita se vastaanottaa (INPUT) ja sen sisältä ulospäin suuntaavia asioita (OUTPUT). Abstraktilla tasolla voidaan määrittää laitteen tuottavan sekä vapauttavan energiaa, materiaaleja ja signaaleja.



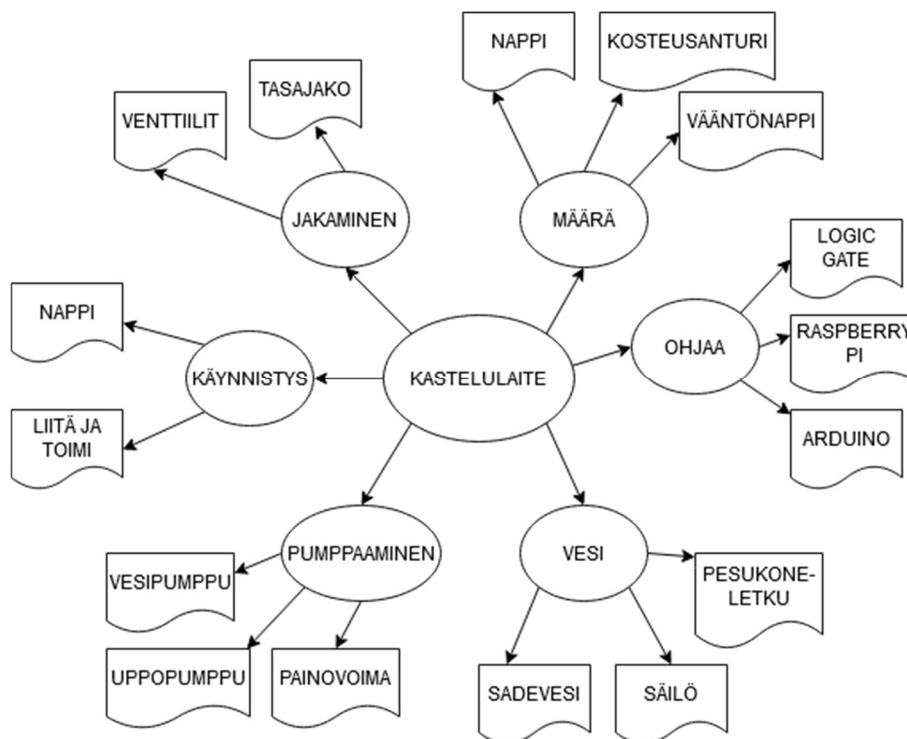
Kuvio 4. Kasvienkastelu järjestelmä mustana laatikkona

Kuvion 4 musta laatikko antaa tarpeelliset tiedot sen sisäisten toimintojen suunnitteluun. Seuraava vaihe on suunnitella sen sisällä loogisesti tapahtuvat asiat ilman, että pohditaan niitä liian yksityiskohtaisesti. Kuviossa 5 näkyy Function Diagram -hahmotelman lopputulos.



Kuvio 5. Function Diagram -hahmotelma kasvienkastelu järjestelmästä

Function Diagram -hahmotelman lopputuloksen ja internetin tutkimisen avulla voidaan aloittaa ideointi aivoriihellä. Kuvion 5 toiminnot antavat aivoriihelle pääotsikot ja niille keksitään lyhyessä ajassa erilaisia toteutustapoja näille mustan laatikon toiminnoille. Aivoriihen tulokset kerätään lopuksi miellekarttaan.



Kuvio 6. Aivoriihen tulokset miellekartalla

Kuviossa 6 näkyy aivoriihen lopputulos. Tuloksilla luodaan taulukko, joka jaetaan kategori-oihin ja joihin lisätään omat lokerot; ideat sitten hahmotellaan luonnostelemalla ne lokeroihin. Luonnostelu ja lokerointi auttavat seuraavan tehtävän suorittamisessa, jossa yhdistel-ään ideoita laitekokonaisuuksiksi valitsemalla yksi lokero jokaisesta kategoriasta. Ku- vassa 1 on aivoriihen tuloksista tehty taulukko hahmotelmineen.

	TALON OMA VESI- LÄTKU	SÄILÖ	SADEVESI
VESI			
KÄYNN- ISTYS	LIITÄ JA TOIMI 		NAPPI/ KYTKIN
OHJAUS	LOGIC GATE 	ARDUINO 	RASPBERRY PI
MÄÄRÄ	VÄÄNTÖNAPPI 	NAPPI 	KOSTEUS ANTURI
PUMPP- AUS	VESIPUMPPU 	PAINOVOIMA 	UPPOPUMPPU
JAKO	VÄNTTILI 		TASAJAKO

Kuva 1. Aivoriihen tuloksista tehty taulukko

Kuvan 1 taulukko ei ole ”virallinen” suunnittelutyökalu, mutta se mukailee Ulrich-Eppinger-menetelmän ”Sub Problem Solving” -osion työkaluja. Yleinen tavoite kummassakin on luoda erilaisista ratkaisuista matriisi, jota yhdistelemällä saadaan laitekokonaisuuksia.

Taulukon käyttö kuvan 1 tapauksessa etenee valitsemalla jokaiselta riviltä yksi vaihtoehto, jonka tuloksena saadaan laitekokonaisuus. Taulukko 3 esittää laitekokonaisuuksien valinnat kolmelle laitteelle eli idealle.

Taulukko 3. Laitekokonaisuuksien valinnat

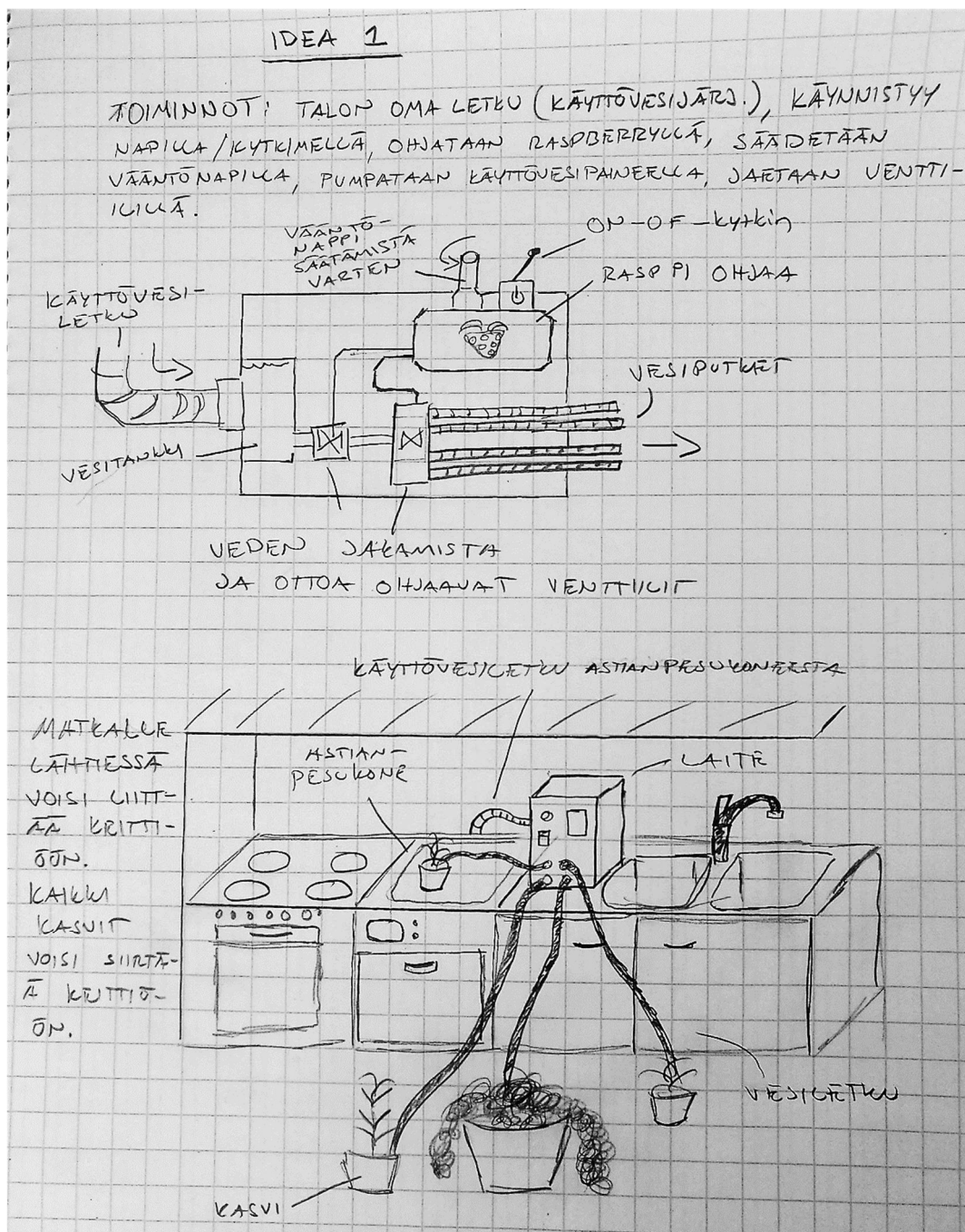
Toiminto	Idea 1	Idea 2	Idea 3
Vesi	Talon oma vesiletku (käyttövesijärjestelmä)	Sadevesi	Säiliö
Käynnistys	Painike/Kytkin	Liitä ja toimi	Painike/Kytkin
Ohjaus	Raspberry Pi	Logic gate	Arduino
Määrä	Vääntökytkin	Painike	Kosteusanturi
Pumppaus	Painovoima (käyttövesijärjestelmän oma paine)	Uppopumppu	Vesipumppu
Jako	Venttiili	Tasajako	Venttiili

Ideoita laitekokonaisuuksille on hyvä olla enemmän, kuin vain yksi. Yhden idean kehittäminen rajoittaa suunnittelijan mielikuvitusta ja kekseliäisyyttä, mikä saattaa jättää paljon hyviä ideoita tutkimatta. (Ulrich & Eppinger 2012, 128.) Tästä syystä taulukossa 3 on kolme ideaa, joita lähdetään kehittämään luonnosteluvaiheeseen.

Luonnostelujen on tarkoitus olla selkeitä ja selittäviä piirroksia ideoista, menemättä kuitenkaan liikaa yksityiskohtiin. Luonnosteluvaiheen piirrokset kannattaa tehdä käsin luovuuden vuoksi; tyyli on vapaa, mutta tekniset ohjelmistot saattavat rajoittaa luovuutta nopean piirroksen tekemisessä. Käsin piirtämistä tulisi harjoitella muutenkin, sillä se on nopea tapa saada idea visuaalisesti esille. Ideat myös nimetään, koska vertailtaessa ne menevät helposti sekaisin; numeroidut ideat ovat huono tapa muistaa, mikä laite oli kyseessä. Näin aikaisessa vaiheessa prosessia voidaan nimetä laitteet vedensääntömenetelmän alkukirjaimilla ja ohjausmenetelmää kuvaavilla kirjaimilla.

3.3.1 Idea 1: KV-Pi

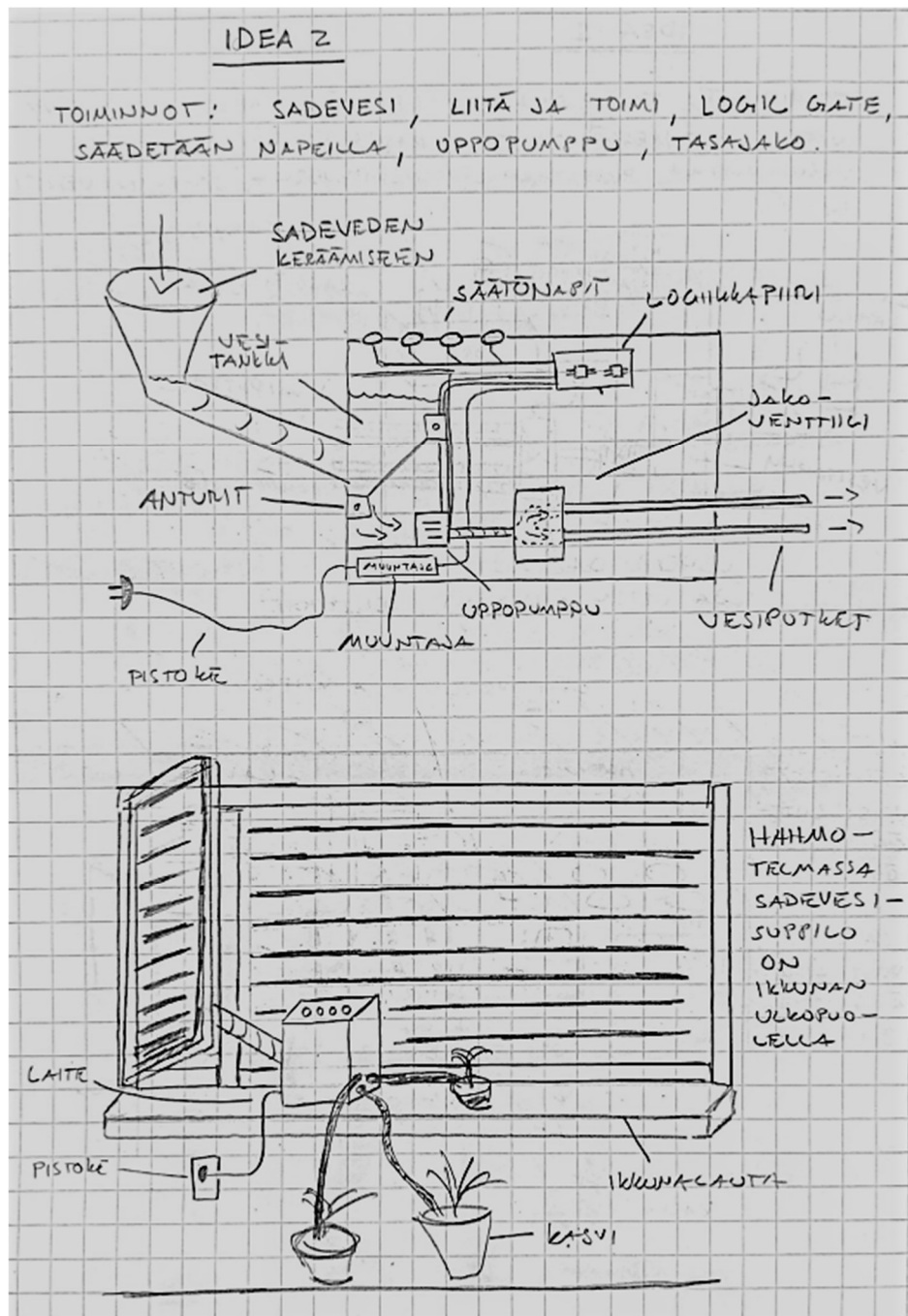
Kuva 2 näyttää, että luonnos syntyi kirjoittamalla ensin toiminnot ja sitten niiden perusteella hahmottelu kokonaisuudesta. Ensimmäisen idean vedensaanti tulee käyttövesijärjestelmästä, jonka oma paine korvaa pumpput ja määrää säädetään venttiileillä. Raspberry Pi -tietokone ohjaa kokonaisuutta, ja kastelumäärää voidaan säätää vääntökäytimellä. Laite myös käynnistetään ja sammutetaan erillisellä kytkimellä. Esimerkiksi keittiössä pesukoneen vesiputki voitaisiin mahdollisesti liittää laitteeseen.



Kuva 2. Luonnos ideasta 1

3.3.2 Idea 2: SV-Logic

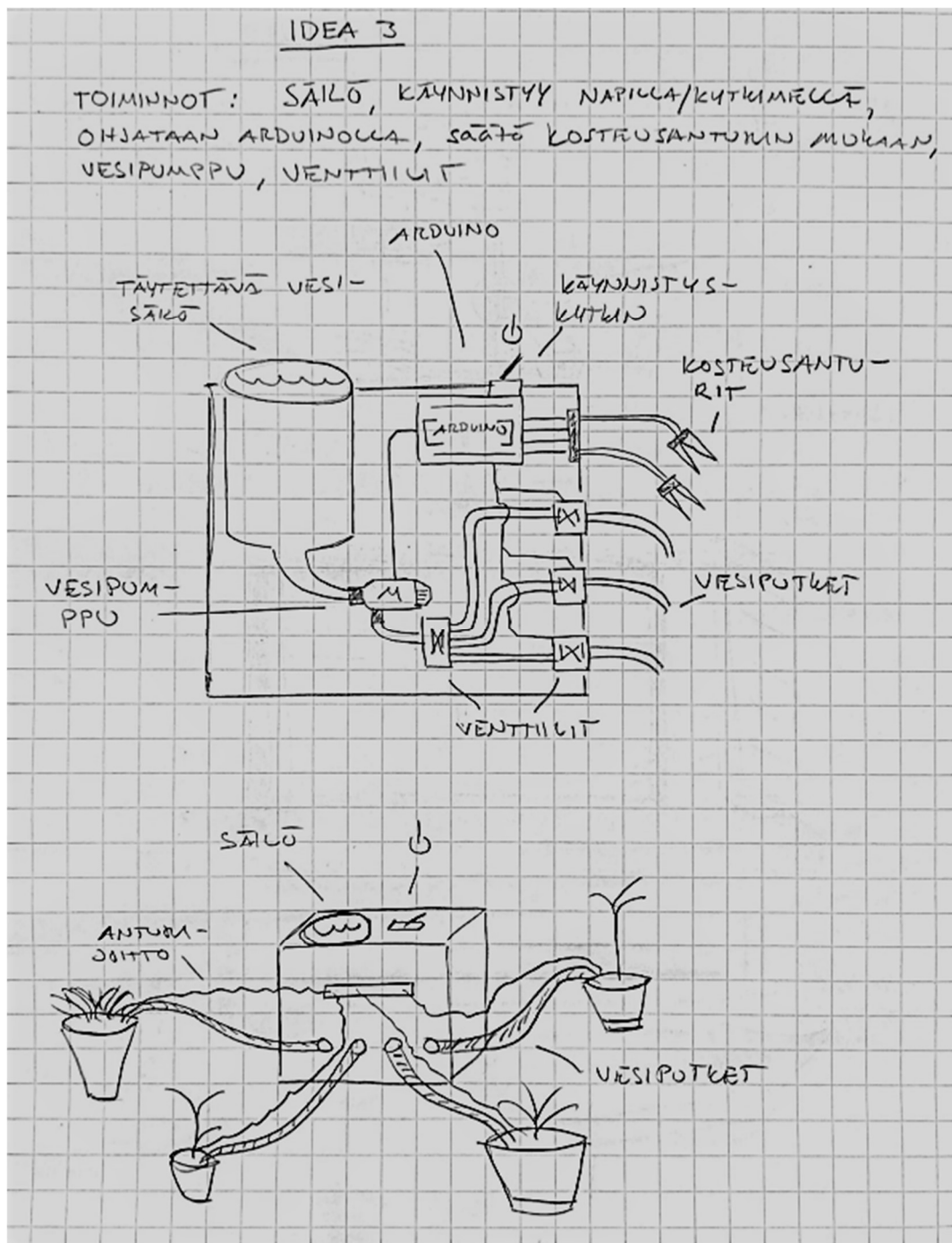
Sadeveden käyttö on oiva tapa säästää energiaa, ja idea perustuu sen hyödyntämiseen. Laite kerää sadevettä, mikä siirtyy vesitankkiin ja josta pumpataan vesi uppopumpulla. Laite alkaa toimimaan heti, kun sen liittyy seinän sähköpistokkeeseen. Kastelumäärää säädetään painikkeilla ja toimintaa ohjaa logiikkapiiri, joka kerää tiedot vesitankissa olevista antureista.



Kuva 3. Luonnos ideasta 2

3.3.3 Idea 3: VS-A

Luonnos ideasta 3 toimii normaalilla vesipumpulla, jota ohjataan Arduino-mikrokontrollilla. Vesi saadaan täytettävästä laitteesta olevasta vesisäiliöstä. Veden pumppausta, määrää ja jakamista ohjataan Arduinolla, joka kerää tiedot kasvien kosteudesta kosteusantureilla. Laite käynnistetään ja sammutetaan kytkimellä.



Kuva 4. Luonnos ideasta 3

3.3.4 Luonnosten arviointi ja analysointi

Kun ideoista on tehty laitetta kuvaavat luonnokset, siirrytään niiden vertailemiseen. Tavoitteena on valita yksi laiteideoista yksityiskohtaisempaan suunnitteluun. Vertailussa käytetään alussa asetettuja määritelmiä arvioimalla, kuinka kukin laiteidea täyttää asetetut määritelmät.

Tauluko 4. Laitteideoiden arviointi

Määritelmät	Ideat		
	KV-Pi	SV-Logic	VS-A
Kompakti	1	1	0
Automaattinen kasteluaika	1	1	1
Helppokäyttöinen/Yksinkertainen	1	0	1
Nopea asentaa toimintakuntoon	-1	0	1
Edullinen	-1	1	-1
Kevyt	1	1	0
Monen kasvin kastelu	1	0	1
Korjaamisen mahdollisuus	0	-1	1
Pisteet	3	3	4
(Pisteytys: Kyllä = +1, Ehkä = 0, Ei = -1)			

Taulukossa 4 näkyy arviointivaiheen lopputulos. Arvioinnissa on tehtävä taustatutkimusta siitä, miten eri määritelmät täyttyvät laiteideoissa. Tätä varten on katsottava esimerkiksi ohjaukseen liittyen karkeasti niiden tekniset tiedot ja ominaisuudet, sillä arvioinnin pisteytyksiä täytyy pystyä perustelemaan.

3.3.5 Perusteet pisteytyksille

Pisteytyksille annetaan perusteet taustatutkimuksen ja havaintojen perusteella. Perusteleminen pakottaa tarkastelemaan omista mieltymyksistä johtuvien valintojen kriittisesti. Käydään läpi kaikki kahdeksan määritelmää laiteideoiden osalta:

1. Kompakti: Eri laiteideoissa ”kompakti” määritelmän arvioinnissa on käytetty olemassa olevien osien karkeita mittoja. KV-Pi täyttää määritelmän kriteerit, mutta arvioinnissa ei ole huomioitu mahdollisia turvalaitteita, jotka täytyy asentaa johtuen käyttövesijärjestelmän käyttämisestä laitteessa. SV-Logic:ssa tilaa on runsaasti olettaen, että veden säilytykseen ei tarvitse suurta vesitankkia. VS-A saattaa viedä

enemmän tilaa, koska systeemin vedensaanti perustuu laitteessa olevaan täytettävään vesisäiliöön, johon ei saa lisää vettä käyttäjän ollessa poissa. Näin ollen vesisäiliön koko saattaa tehdä laitteesta suuremman, toisin kuin esimerkiksi KV-Pi:ssa, jossa säiliötä ei tarvita ollenkaan.

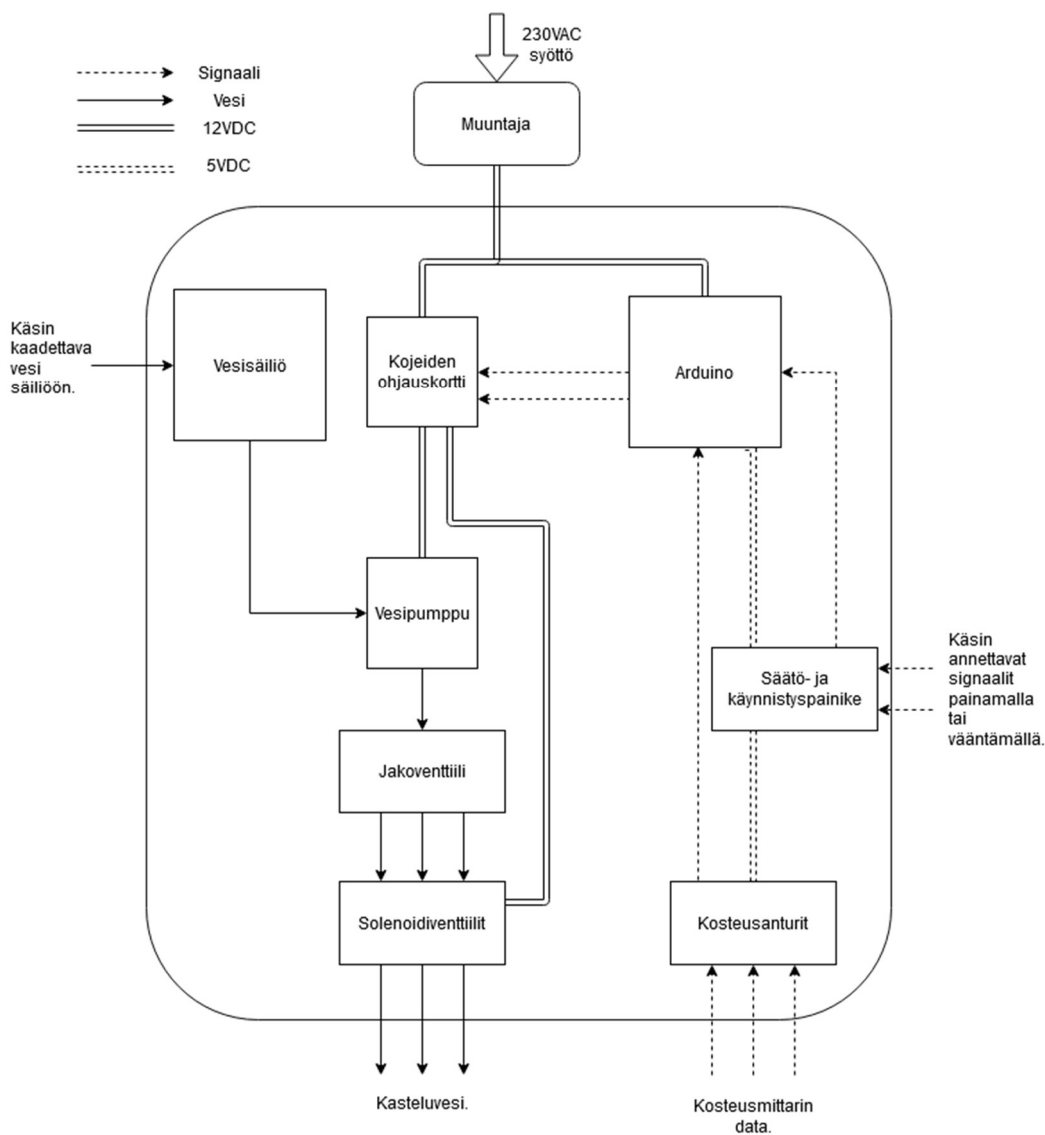
2. Automaattinen kastelu-aika: Kaikki ideat täyttävät tämän määritelmän kriteerit, mutta on huomioitava, että esimerkiksi SV-Logic:ssa kastelu on automatisoitu, niin itse kastelu on riippuvainen sateesta. VS-A:ssa kasteluajan pidentyessä on vesisäiliön kokoa suurennettava, mikä taas suurentaa laitteen kokoa. KV-Pi täyttää tämän kriteerin täydellisesti, sillä vedensaanti on taattu käyttövesijärjestelmästä.
3. Helppokäyttöinen/Yksinkertainen: Laitteet KV-Pi ja VS-A omaavat älykkäämmän ohjauksen, minkä ansiosta ne voivat yhden säätimen avulla toteuttaa tarkempaa säätämistä. SV-Logic tarvitsee enemmän painikkeita säätämiseen.
4. Nopea asentaa toimintakuntoon: KV-Pi vaatii laitteen asentamista käyttövesijärjestelmään, mikä taas vie aikaa ja saattaa vaatia putkimiehen paikalle pyytämistä. SV-Logic:n ajatus on nopea käyttöönotto liittämällä laite vain pistokkeeseen, mutta todellisuudessa sadevesisuppilon asettaminen paikoilleen sopivaan paikkaan vie enemmän aikaa. VS-A täyttää määritelmän, mutta on otettava huomioon, että laitteen käyttöönotossa on enemmän vaiheita (anturit, putket, veden täyttäminen ja laitteen säätäminen).
5. Edullinen: Tästä määritelmästä ainoastaan SV-Logic täyttää kriteerit. KV-Pi:n ja VS-A:n osat maksavat enemmän, esimerkiksi jo mikrokontrolleri ja tietokone maksavat enemmän, kuin logiikkapiirilevy.
6. Kevyt: VS-A:ssa on enemmän osia ja koko on isompi, mikä tuo painoa lisää. Venttiilit, vesipumppu ja vesisäiliö lisäävät myös laitteen kokoa ja näin ollen lisäävät painoa.
7. Monen kasvin kastelu: Käyttövesijärjestelmän tuottama vesi ja paine antavat mahdollisuuden kastella montaakin kasvia samanaikaisesti KV-Pi:ssa, kun taas SV-Logic:n vesimäärä riippuu täysin säästä. VS-A:ssa toistuu tässäkin määritelmässä suhde vesisäiliön kokoon.
8. Korjaamisen mahdollisuus: KV-Pi:n korjaamismahdollisuus on ongelmallista, johon tuen käyttövesijärjestelmän käyttämisestä laitteessa. Suuren paineen ja vesivahingon mahdollisuus tekevät laitteen korjaamisesta kotona vaarallista, mutta sisäisten venttiilien ja ohjauksen korjaaminen on mahdollista. SV-Logic vaatii piirilevyn tekemistä itse, koska markkinoilla ei ole suoraan tilattavaa kappaletta tähän

laitteeseen. VS-A:ssa on vaihdettavuus hyvä, koska kaikki laitteen osat ovat markkinoilla kuluttajille saatavilla ja eivät välttämättä tarvitse tiettyä osaa, vaan samankaltainen ja yhteensopiva osa riittää.

Lopputuloksessa näkyy, etteivät kaikki määritelmät täyty. Näitä ongelmia voidaan kuitenkin korjata yksityiskohtaisessa suunnittelussa, jossa parannellaan laiteidea ja jatkokehitetään kaikkia idean osa-alueita. Kaikista kolmesta ideasta jatkoon valittiin VS-A, mutta se ei tarkoita, etteikö muista ideoista voitaisi ottaa hyviä ratkaisuja käyttöön.

3.4 Systemisuunnittelu

Kun ideoista on valittu yksi, voidaan siirtyä systemisuunnitteluun. Tuotteen systeemiä voi alkaa suunnittelemaan edellä käytetyn Function Diagram -hahmotelman pohjalta. Kuvio 7 on tämän suunnitteluvaiheen lopputulos, joka kuvaa systeemiä ja sen toimintaa.



Kuvio 7. Systemistä tehty Function Diagram -hahmotelma

Kuviossa 7 on jaoteltu systeemi ohjauksen, sähkön ja vedenkulun mukaan. Muuntaja tuottaa 12V tasajännitettä Arduinolle, pumpulle ja solenoidiventtiileille. 12V tasajännitteen valitseminen oli kompromissi, sillä Arduinon suositeltu tulojännite on 7-12VDC (Arduino SA. 2020a) ja helposti saatavilla olevat pienet vesipumput, jotka on tarkoitettu esimerkiksi autojen tuulilasien pesunesteen pumpuiksi ovat myös samassa jänniteluokassa. Myös helposti saatavilla olevat ja halvat solenoidiventtiilit ovat 12-24VDC:n syötöllä. Arduinossa on lisäksi oma 5VDC jännitelähtö, johon voidaan liittää painikkeet sekä anturit (Arduino SA. 2020a). Arduinolla ei kuitenkaan voida suoraan ohjata pumppua ja solenoideja, joten väliin tarvitaan kuviossa 7 näkyvä kojeiden ohjauskortti, jossa on releitä.

Jotta laitteen sisällä säästettäisiin tilaa ja siitä saataisiin turvallisempi pitämällä sähkö alle suojajänniterajan (alle 24VDC), systeemissä on ulkoinen muuntaja. Muuntajasta voidaan kytkeä johto IP65 (tiiviytsluokitus) luokituksen ylittävään sähkökeskukseen ja jakaa sähkö muualle systeemiin sen sisältä.

Lopuksi systeemissä jää jäljelle veden kulkureitti. Kuviossa 7 näkyy manuaalinen vedenkaato säiliöön, josta sitä tarvittaessa pumpataan ja ohjataan venttiilien avulla oikeille kasveille.

Lopputuloksena on teoriassa toimiva systeemi, joka ei konkreettisesti rajoita suunnittelua, vaan antaa hyvät lähtökohdat yksityiskohtaisempaan suunnitteluun. Systeemiin voi vielä vaikuttaa komponenttien valitsemisella ja niiden määrällä. Tässä vaiheessa on myös helppoa tehdä työnjako ryhmässä, jossa kukin suunnittelija keskittyy yhteen systeemin osaluueeseen. Systeemisuunnittelun päätteeksi on mahdollista siirtyä nopeaan prototyypin tekemiseen, jolloin voidaan ennen yksityiskohtaista suunnittelua kokeilla systeemin toiminnallisuutta. Prototyyppi voidaan toteuttaa pienemmässä koossa, jossa esimerkiksi testataan vain pumpun ohjausta. Näin saadaan nopeita tuloksia.

3.5 Prototyyppi

Kiireisissä aikarajoissa kannattaa valmistaa nopea ja toiminnallinen prototyyppi. Tällaisella prototyypillä halutaan todistaa idean tai systeemin toiminta ja saada nopeita tuloksia. Prototyyppien tekemiseen kuuluu epäonnistumiset ja kompromissien löytäminen sekä rahaa kuluu yleensä enemmän, kuin osataan odottaa. Erilaisten ideoiden kokeilemista ei tulisi kuitenkaan estää mikään, sillä prototyyppien tekemisellä voidaan keksiä uusia ideoita ja huomata mahdollisia ongelmia varhaisessa vaiheessa.

Kasvienkastelujärjestelmän prototyypin materiaalit oli löydettävä mahdollisimman helposti ja läheltä, jotta saatiin aikaiseksi nopea prototyypin rakentaminen. Arduino, vesipumppu ja tehölähde olivat keskeisimmät prototyypin osat, ja kriteerinä oli 12V tasajännitteen

yhteensopivuus kaikissa kolmessa. Osien oli tarkoitus olla helposti vaihdettavissa ja mahdollisimman universaaleja, jotta osan rikkoutuessa ei tarvitsisi huolehtia tietyn mallin ja merkin saatavuudesta.

Prototyypin rakentaminen alkoi etsimällä sopivia osia paikallisista kaupoista ja internetistä, jonka jälkeen hahmoteltiin kokonaisuus paperille. Tässä vaiheessa tarkasteltiin myös sähkösuunnittelua sekä vesiletkujen liitoksia prototyypissä. Lähistöllä olevista rautakaupoista löytyi sopivat tuotteet. Kaikki prototyypin osat on listattu taulukossa 5.

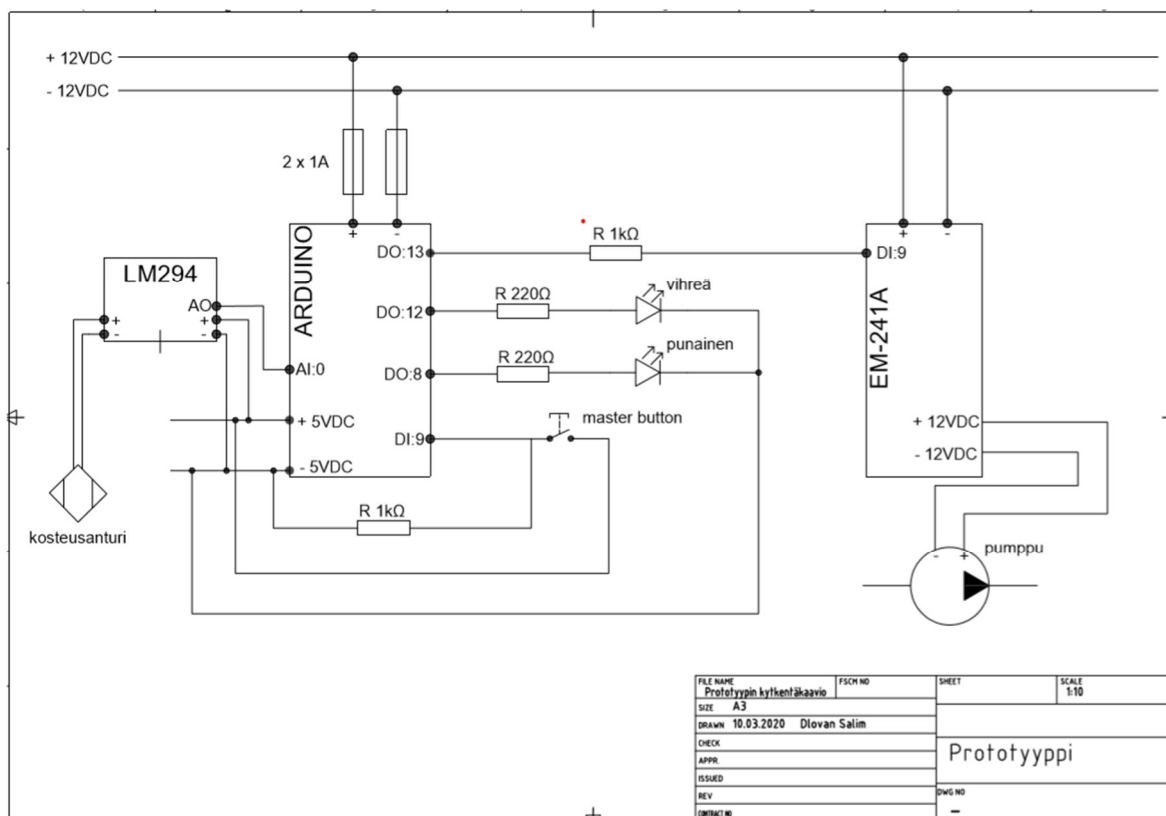
Taulukko 5. Prototyypin osaluettelo

PROTOTYYPIN OSALUETTELO			
tuote	tiedot	määrä (KPL)	hinta/kpl (€)
Arduino Uno	Arduino Uno R3	1	14,90
Vesipumppu	Pesulaitteen tuplapumppu TOPRAN 12VDC	1	7,99
Johdot	Koekytkentäkaapeli (pakkaus)	1	5,90
Vastukset	1-10kΩ vastuksia (pakkaus)	1	5,90
Vesisäiliö	Hamsterin juomapullo	1	3,99
Breadboard	Koekytkentälevy	1	5
Muuntaja	Virtalähde 12VDC 6,67A 90W	1	27,80
Lasisulake	1A lasiputkisulake	2	3,90
Sulakepesä	Muovinen sulakepesä lasiputkisulakkeelle	2	0,99
moottoriohjain	EM-241A v1.5 moottorinohjaukorkortti	1	43,99
Painike	Painonappi	1	3,90
LED	LED	2	1
KytKentärima	3-liitin kytkentärima	1	2,99
Putket	muoviletku 2m D8mm	1	1,99
Putkiliittimet	Y-liitin, muovi, D8mm	1	0,2

Taulukko 5 näyttää lopullisen osaluettelon, jonka hinnaksi tuli 130,44€. Prototyypin osien ostaminen ei kuitenkaan ole välttämättä suoraviivaista, vaan tässäkin on vain lopullinen osaluettelo. Todellisuudessa osia on paljon enemmän, mutta ne eivät joko sopineet tai toimineet odotetulla tavalla ja jätettiin pois prototyypistä.

Osia, etenkin pumppua ja elektroniikkaa, testattiin erikseen ennen niiden liittämistä toisiinsa. Esimerkiksi pumppua testattiin laboratorio teholähteellä, kun tutkittiin tarvittavaa virtamäärää suhteessa pumpattavaan vesimäärään. Arduinolla ohjaamista testattiin painikkeilla ja signaalin kulkua LED-lampuilla. Testaaminen oli tärkeä osa prototyypin rakentamista, sillä moni osa oli eri laitteiden varaosia ja kopiomalleja, joiden mukana ei tullut tai niistä ei löytynyt tarkkoja teknisiä tietoja.

Kun kaikki osat todettiin toimiviksi, niin luotiin kuvan 5 mukainen sähkökytkentäkaavio. Kaavion avulla kytkettiin kaikki osat toisiinsa, jonka jälkeen päästiin kirjoittamaan systeemin toimivuutta testaavaa ohjelmaa Arduinoon.



Kuva 5: Kuvakaappaus prototyypin sähköjen ja ohjauksen kytkentäkaaviosta

Arduino toimii C-pohjaisella kielellä (Arduino SA. 2020b), joka on helppo oppia internetissä olevien ohjeiden ansiosta. Aikaisessa vaiheessa testattiin yksinkertaisempia asioita ohjelmointikielen oppimisen vuoksi, esimerkiksi signaalien simulointia LED-lampuilla. Opettelun jälkeen oli tehtävä ohjelma, jolla koko systeemin toimintoja voitaisiin prototyypissä testata.

Arduinon ohjelmoinnissa auttaa sen yksinkertainen rakenne. Ohjelmassa on muuttujat, "Void Setup ()"- ja "Void Loop ()"-funktiot. Muuttujien asettamisen jälkeen täytetään "setup"-funktio, jonka käskyt suoritetaan ohjelmassa Arduinon käynnistyksen ja nollaamisen yhteydessä. Siinä määritetään, mitkä liitäntäpinnit lähettävät signaaleja ja mitkä vastaanottavat niitä, eli määritetään lähdöt ja tulot. Ohjelman viimeinen osa, "Void Loop ()"-funktio, on ohjelman toistuva käskylista. Ohjelma kiertää käskylistan alusta loppuun niin kauan, kun Arduino on päällä. Ohjelma on kirjoitettu englanniksi noudattaen hyvää ohjelmointitapaa ja sen toimintaa on kuvattu ohjelman sisäisillä kommentteilla, jotka alkavat // tai /* -merkeillä. Kuvassa 6 näkyy koko prototyypin ohjelma.

```

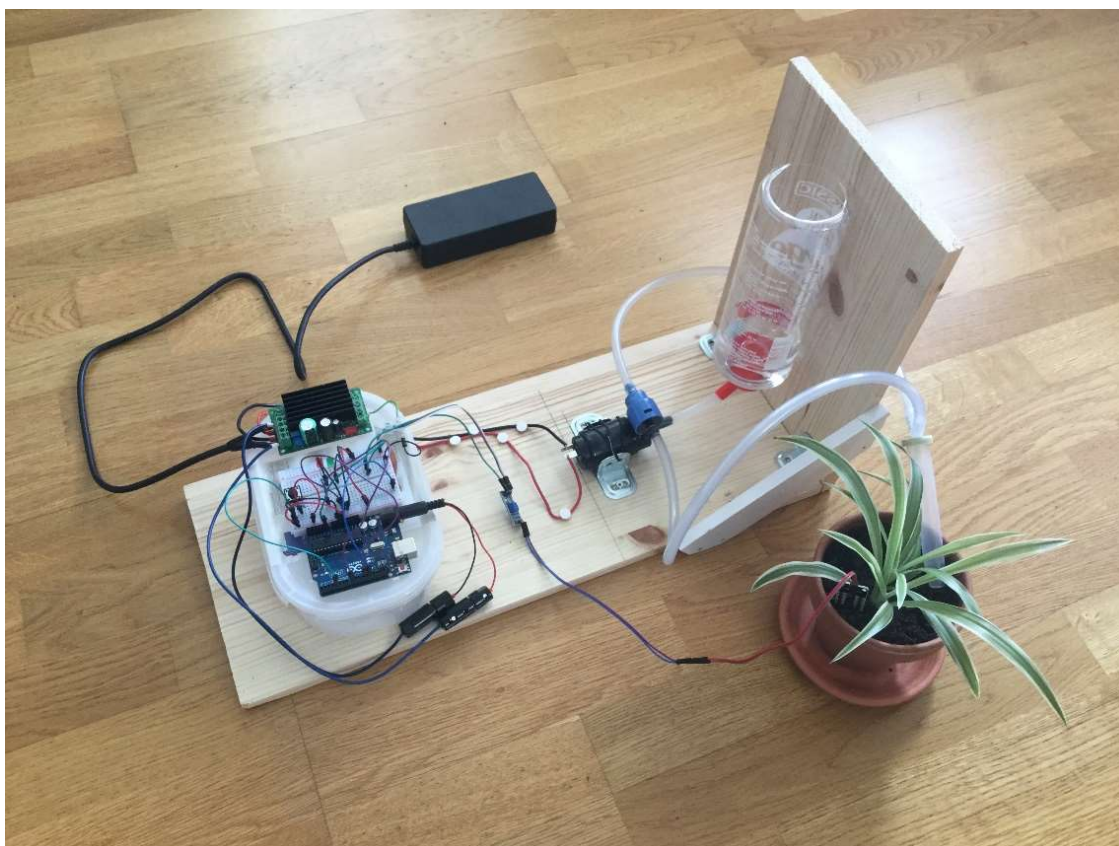
1 // Noudattaen hyvää ohjelmointitapaa, kirjoitin koodin englanniksi.
2 // in first part we declare the variables
3 int FWD_signal = 13; /* PIN - pump motor: forward signal */
4 int RUN_signal = 12; /* PIN - green led light if master button is on */
5 int WET_signal = 8; /* PIN - red led if plant is already wet */
6 int MASTERBUTTON = 4; /* PIN - start code */
7 int DAYS = 3; /* amount of for-loops (1 loop should be 1 day) */
8 int PUMP_TIME = 2000; /* pumping time in ms */
9 int WAIT_DAY = 86400000; /* pause between sensing the moisture in ms for 24 hours */
10
11 // moisture sensor variables
12 int sensorPin = A0; /* PIN - Analog pin for moisture sensor */
13 int sensorValue = 0; /* return value for moisture amount (1024 is dry) */
14 int MoistureLimit = 700; /* Moisture level limit for dryness, over 700 dry; activate pumping */
15
16
17 // in the second part we declare which variables are pin numbers and whether they are outputs or inputs
18 // setup() function only runs once; upon arduino start or restart
19 // serial.begin is for tuning the frequency between the terminal and arduino to monitor moisture values
20 void setup() {
21     Serial.begin(9600);
22     pinMode(FWD_signal, OUTPUT);
23     pinMode(RUN_signal, OUTPUT);
24     pinMode(WET_signal, OUTPUT);
25     pinMode(MASTERBUTTON, INPUT);
26 }
27
28 /* void loop is the code that runs repeatedly while arduino is on. Here
29 the first IF-statement is essential for the rest of the code to work.
30 MASTERBUTTON enables the first IF-statement. Then the code will start
31 a FOR-loop for set amount of days. One FOR loop will last 24 hours to
32 ensure that it checks the sensor data once per day. After the code has
33 finished, the MASTERBUTTON will reset and the code will wait until
34 MASTERBUTTON is enabled for the code to start working again.*/
35 void loop() {
36
37     if (digitalRead(MASTERBUTTON) == HIGH) {
38
39         digitalWrite(RUN_signal, 1); // turn on green led light
40
41
42
43         // LOOP THROUGH GIVEN AMOUNT OF DAYS
44
45         for (int i = 1; i <= DAYS; i++) {
46
47             // SETUP 1: sense moisture
48
49             sensorValue = analogRead(sensorPin); // activate & read sensor
50
51             Serial.println("Analog Value: ");
52             Serial.println(sensorValue); // print sensed value to console
53
54
55             // SETUP 2: decide if its wet or dry
56
57             if (sensorValue > MoistureLimit) { // limit hardcoded 700 > dry plant
58
59                 // IT IS DRY
60
61                 digitalWrite(FWD_signal, 1); // turn on pump
62                 delay(PUMP_TIME); // pump as long as pump time value
63
64                 digitalWrite(FWD_signal, 0); // turn off pump
65
66                 // for checking loops and which one it is running we need: Serial.println(i);
67                 Serial.println(sensorValue);
68             }
69             else {
70                 // IT IS WET
71
72                 digitalWrite(WET_signal, 1); // light up red led
73                 delay(1000); // wait 1 second
74                 digitalWrite(WET_signal, 0); // turn off red led
75             }
76
77             // STEP 3: WAIT FOR THE NEXT DAY (unless its the last day)
78
79             if (i < DAYS) {
80                 delay(WAIT_DAY); // wait for a day to pass
81             }
82
83
84         }
85
86     }
87
88     // END OF DAYS IN FOR-LOOP
89
90     digitalWrite(FWD_signal, 0); // turn off pump
91
92     digitalWrite(RUN_signal, 0); // turn OFF green light
93
94 }
95
96 // END OF PROGRAMM
97
98 }
99

```

Kuva 6. Kuvakaappaus prototyypissä Arduinoon ladatusta ohjelmasta

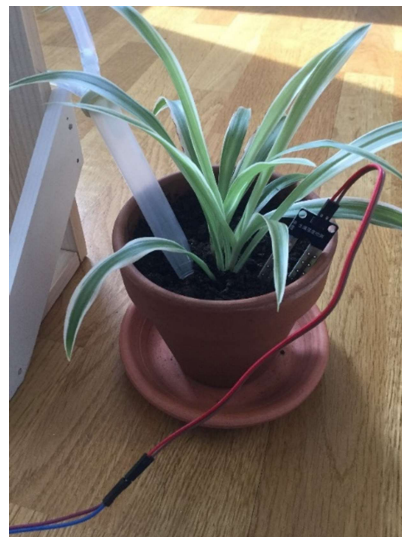
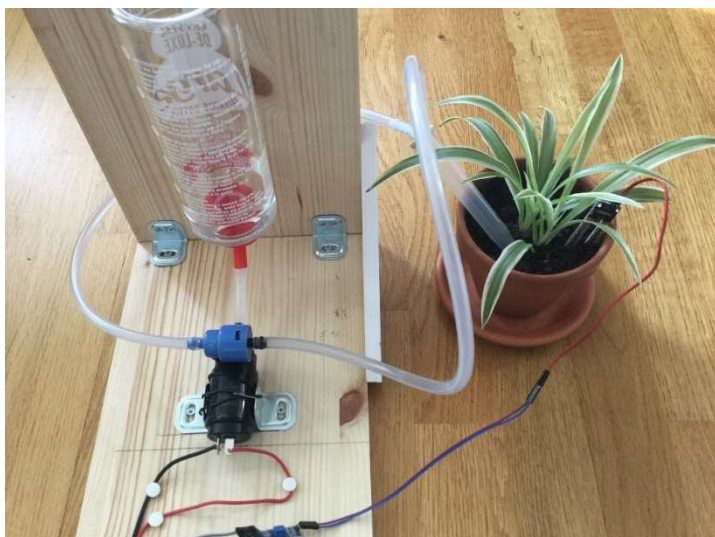
Ohjelma on kokonaan selitetty sen sisältämissä kommentteissa, mutta toiminnan keskeisimmät asiat ovat seuraavat: Ohjelma odottaa "Master Button" -painiketta käynnistääkseen kasteluohjelman, jonka jälkeen se tarkistaa, kuinka monta päivää automaattinen kastelu kestää "DAYS"-muuttujasta. Päiviä kuvaava muuttuja on nykyisessä ohjelmassa kolme päivää. Kasteluohjelman ollessa päällä tarkistetaan joka kahdeskymmenesneljäs tunti kasvin kosteus. Jos multa on kuivaa, niin pumppu käynnistetään "FWD_signal" -signaalilla kahdeksi sekunniksi. Kun kasteluohjelma lopettaa toimintansa kolmen päivän kulluttua, se jää odottamaan uudelleen "Master Button" -painikkeen painallusta.

Ohjelman valmistuttua se voitiin ladata Arduinoon. Lopuksi, kun koko prototyyppi oli valmis, sitä testattiin muutaman päivän ajan. Tuloksena oli toimiva kastelujärjestelmä.



kuva 7. Prototyyppi

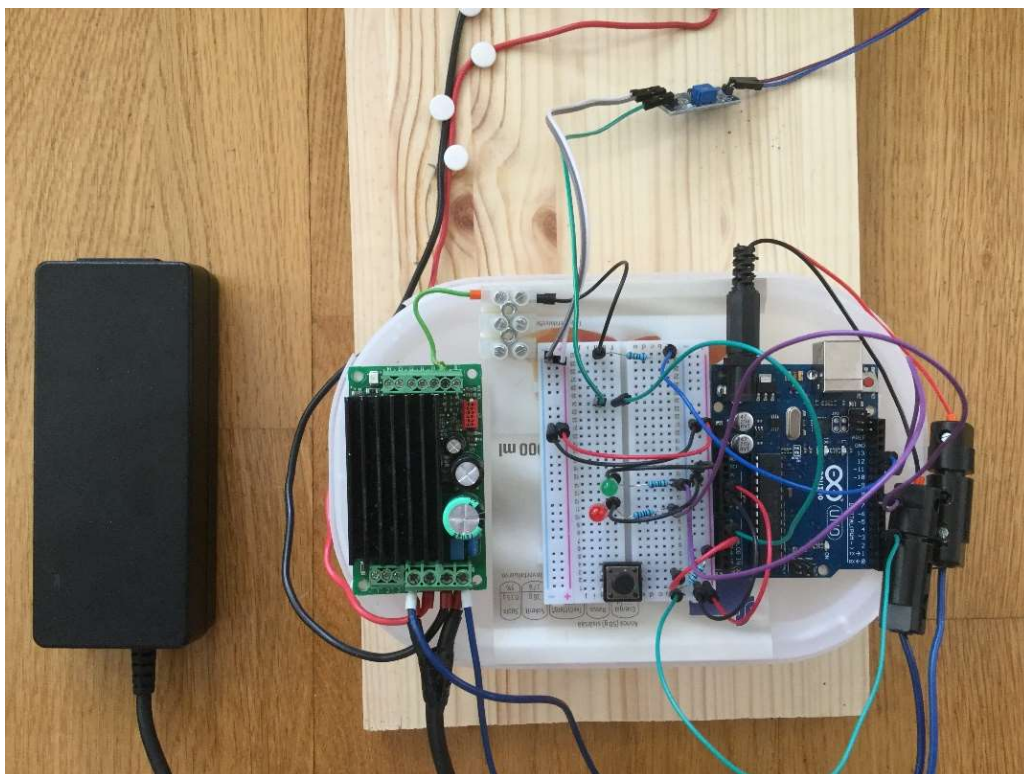
Kuvassa 7 näkyy prototyypin lopputuloksen kokonaisuus. Kuvasta puuttuu ainoastaan muuntajan verkkovirtaan kytkettävä johto. Prototyyppi oli vain systeemin testaamista ja idean todistamista varten, ja sellaisenaan se ei sovi kotikäyttöön.



Kuvat 8 (vas.) ja 9 (oik.). Veden kulkureitti säiliöstä kasviin

Kuva 8 on tarkennettu veden kulkureittiin. Järjestelmän pumpun ulostuloreikiä on kaksi, mutta ne on yhdistetty T-liittimellä. Yhdistäminen tehtiin, koska testauksessa huomattiin toisen ulostuloreiän vapauttavan vähemmän vettä. Kuvassa näkyy myös vesisäiliönä käytettävää hamsterin vesipulloa.

Kuvassa 9 näkyy kasvi ja siihen asetetut kosteusanturi sekä vesiletku. Näin pienessä ruukussa kosteusanturi huomaa kosteuden eron nopeasti kastelun jälkeen, mutta isoimmissa ruukuissa nämä molemmat tulisi asettaa toisiaan yhtä lähelle kuin kuvassa. Lisäksi huomattiin, että kasteluletkuissa ja kosteusantureissa olisi hyvä olla piikikkäät päät helpottamaan niiden kiinnitystä ja asettelua multa. Tämänkaltaiset havainnot ovat prototyypin rakentamisen ydin; huomataan uusia asioita ja saadaan ideoita yksityiskohtaisempaa suunnittelua varten.



Kuva 10. Sähkö ja ohjaus

Sähköjen ja ohjauksen kasaaminen oli Arduinon sekä koekytkentälevyn ansiosta nopeaa ja helppoa. Kuvassa 10 näkyy kaikki laitteen ohjaukseen tarvittavat osat. Kuvan osat vasemmalta oikealla ovat muuntaja, moottorinohjauskortti, koekytkentälevy, Arduino ja lasisulakepesät.

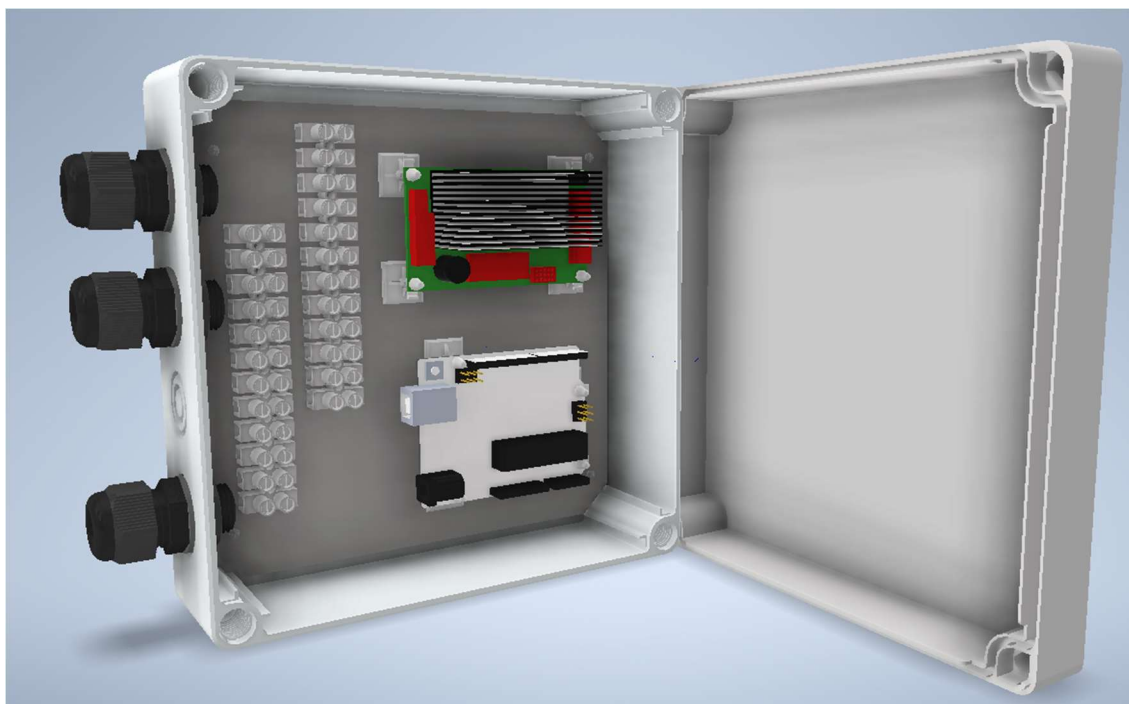
Prototyyppi toimii sellaisenaan yhden kasvin kasteluun, mutta kuten kuvissa näkyy, avoin johdotus ja paljaat osat eivät käy lopputuotteesta. Laitteeseen vaaditaan ulkokuoret sekä muita osia erottamaan sähköt ja vesi toisistaan. Prototyyppi ei ole myöskään käytännöllinen kuljettaa, sillä osat eivät ole tiukasti paikallaan. Seuraavassa osassa siirrytäänkin yksityiskohtaisempaan suunnitteluun, jossa otetaan prototyypillä testattu ja toimivaksi todettu systeemi ja viedään se askeleen pidemmälle suunnittelemalla kotelo, komponenttien kiinnitys ja niiden suojaus.

3.6 Yksityiskohtainen suunnittelu

Systemisuunnittelussa tehty kuvion 7 systeemistä valitaan joko valmiiksi tuotemerkkinoilla olevia osia tai ne suunnitellaan itse. Tässä projektissa valitaan mahdollisimman monta valmiiksi tuotemerkkinoilla olevia osia ”korjattavuus” määritelmän täyttämiseksi. Projektissa ei kuitenkaan tehdä tarkempaa osaluetteloa hintoineen, vaan siinä etsitään sopivat osat kokonaisuuden kartoittamiseksi. Lopputuloksen tavoite on olla esittelykelpoinen malli.

Ensin luodaan ohjaus- ja sähkökeskus Arduinosta ja kojeiden ohjauskortista. Valmistajat ja jälleenmyyjät usein tarjoavat omista tuotteistaan 3D-mallinnuksia, mutta joskus ne on mallinnettava itse – esimerkiksi ohjauskortista ei löytynyt valmista 3D-mallia.

Keskuksen suunnittelussa on mietittävä sen kokoa. Koon määrittämiseen kannattaakin ensin 3D-mallintaa kaikki komponentit ja luoda ohutlevymetallista niille kiinnitysalusta. Kiinnitysalustan koon perusteella valitaan sopiva kotelo. Kotelon valinnassa on mietittävä sen tiiviysluokkaa (IP-luokka), jonka tulisi olla tässä tapauksessa IP65. Tiiviyn varmistamiseksi johtojen täytyy kulkea koteloon myös tiiviin läpivientiholkin kautta.



Kuva 11. Ohjauskeskuksen mallinnus

Kuvassa 11 näkyy, että keskuksessa on prototyypissä olleen koeyhtentälevyn sijasta valittu kytkentärimat liittimiksi tilan säästämisen vuoksi. Keskus on hyvin pieni, jotta koko tuote saadaan mahdollisimman pieneen kokoon ”kompakti” määritelmän täyttämiseksi.

Kojeiden ohjaukseen ei löytynyt sopivaa moottorinohjauskorttia, jolla voitaisiin ohjata pumppua ja neljän eri solenoidiventtiiliä. Kuvassa 11 on käytetty prototyypissä ollutta moottorinohjauskorttia, mutta todellisuudessa se täytyy vaihtaa joko sopivaan tai itsetehtyyn piirikorttiin. Sopivan piirikortin tekeminen on kartoitettava ennen etenemistä ja siihen löytyikin sopiva ohjaustapa Darlington-transistori TIP120 avulla (Arduino SA. 2020c). Piirikortin valmistaminen vaatii jatkokehittelyä, mutta esittelykelpoisuutta varten voidaan jatkaa nykyisellä ohjauskortilla.

Elektroniikka on kiinnitetty kiinnitysalustaan muovisilla korokkeilla, joiden tarkoitus on poistaa ohutlevymetallin ja elektroniikan oikosulkuvaara. Ruuveja on käytetty ainoastaan kiinnitysläpivientien koteloon kiinnittämiseen. Keskuksen osaluettelo näkyy taulukossa 6.

Taulukko 6. Ohjaus- ja sähkökeskuksen osaluettelo

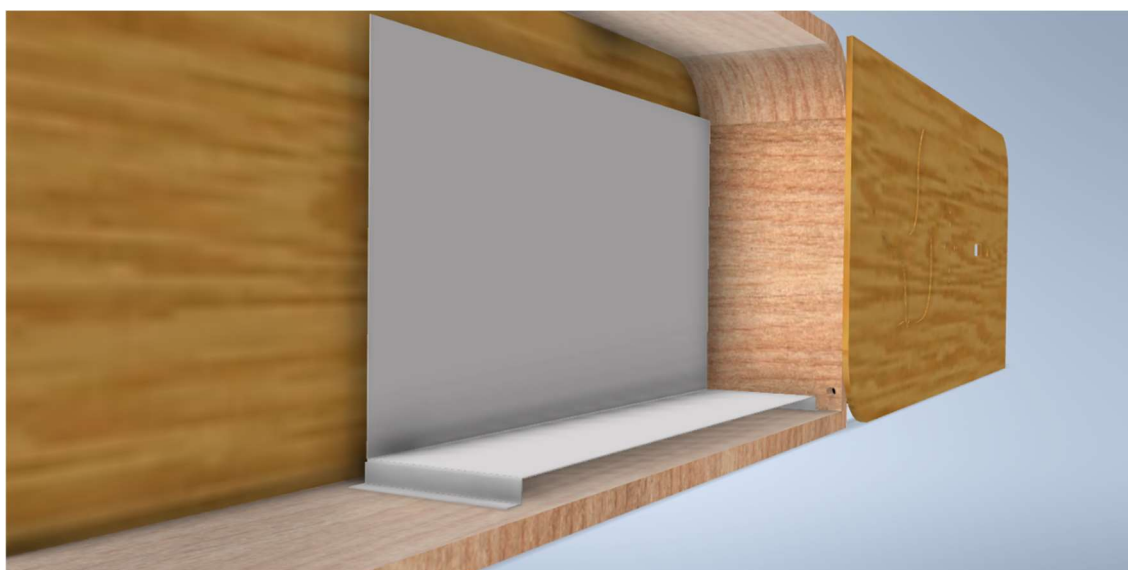
Osa	Määrä (kpl)
Kotelo (+kansi ja kiinnitysruuvit)	1
Ohutlevymetalli (kiinnitysalusta)	1
Läpivientiholkki	3
Kytkentärima	2
Arduino	1
Ohjauskortti	1
Muovikoroke	8
Ruuvit	4

Kuten edellä mainittiin, myös läpivientiholkkien on oltava IP65-luokiteltuja kosteuden takia. Johdon oma ulkoinen kuori suojaa johtoa, ja se kiristetään läpivientiholkista, joka toimii samalla vedonpoistajana. Kuvassa 12 oikealla näkyy puiseen koteloon tehty läpivientireikä muuntajan johtoa varten, joka kulkee kuvassa näkyvään mustaan läpivientiholkiin.



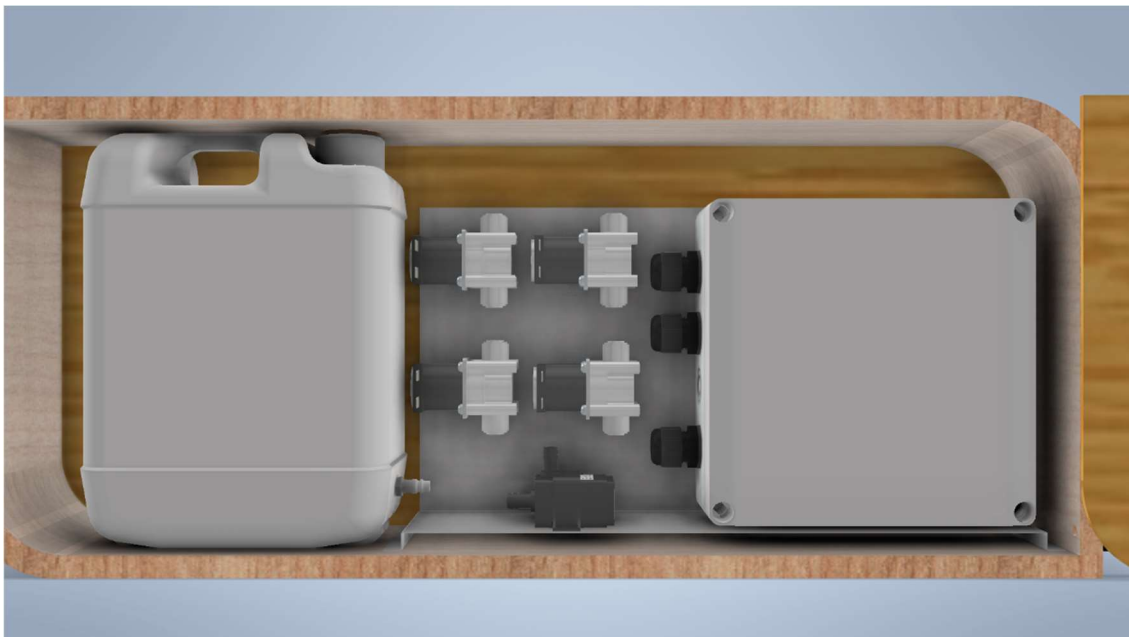
Kuva 12. Muuntajan sähköjohdon kulkureitti

Kun keskuksen suunnittelu on saatu valmiiksi, siirrytään suunnittelemaan laitteen sisäisten osien koteloon kiinnitystä. Kuten keskuksessa, tehdään taas ohutlevymetallista kiinnitysalusta kaikille osille helpottamaan kiinnitystä ja koteloon koon määrittämistä. Keskuksen lisäksi on myös pumppu ja venttiilit, jotka voidaan kiinnittää samaan kiinnitysalustaan. Kaikki osat voisi myös kiinnittää suoraan koteloon, mutta se vaikeuttaa osien asentamista, sillä tila voi olla ahdas. Kiinnityslevyn tarkoitus on helpottaa osien kiinnitystä ja vähentää koteloon ahtaaseen tilaan kiinnitettäviä osia. Lisäksi saadaan puhtaamman näköinen kotelo, jonka ulkokuoressa on vähemmän ruuveja, pultteja ja muttereita. Kuvassa 13 näkyy ohutlevymetallista tehty kiinnitysalusta. Alusta on suorasta levystä leikattu kappale, joka särmätään, eli taivutetaan taivutuslaitteella muotoonsa.



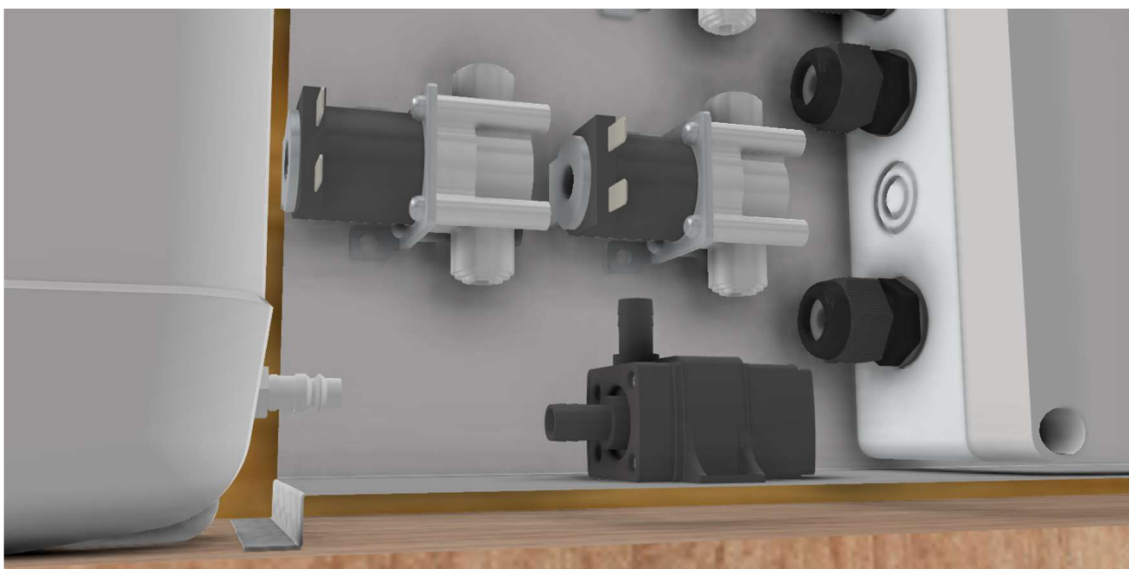
Kuva 13. Kiinnitysalusta mekaniikan kiinnitykseen

Kiinnitysalustaan voidaan kiinnittää kaikki neljä solenoidiventtiiliä, pumppu ja keskus. Laitteesta puuttuu enää vesisäiliö. Vesisäiliön valinnassa on myös mietittävä osien vaihdettavuutta "korjattavuus" määritelmän täyttämiseksi. Yksi mahdollisuus on käyttää tavallista 2,5 litran vesikanisteria. Kuvassa 14 näkyy kiinnityslevy osineen sekä vesisäiliönä toimiva kanisteri.



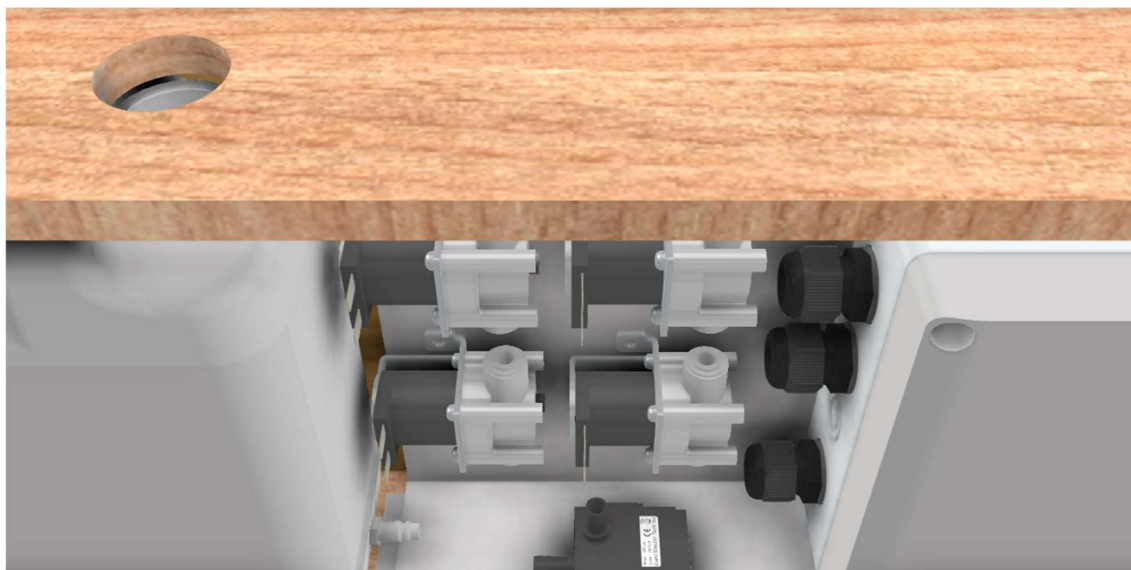
Kuva 14. Asettelyn mallinnus

Kanisteri sopii sellaisenaan laitteeseen, mutta veden pumppaamista varten siihen on tehtävä reikä. Reikään kiinnitetään vesiletkun kiinnityspala, joka samalla tekee reiästä vesitiiviin. Kuvassa 15 oikealla näkyy kanisteriin kiinnitetty letkuliitin. Liitin on mahdollisimman lähellä pumppua, jotta siitä voidaan helposti pumpata vesi ylös solenoidiventtiilien kautta kasveille. Solenoidiventtiilejä avataan ja suljetaan kasvien vesitarpeen mukaan.



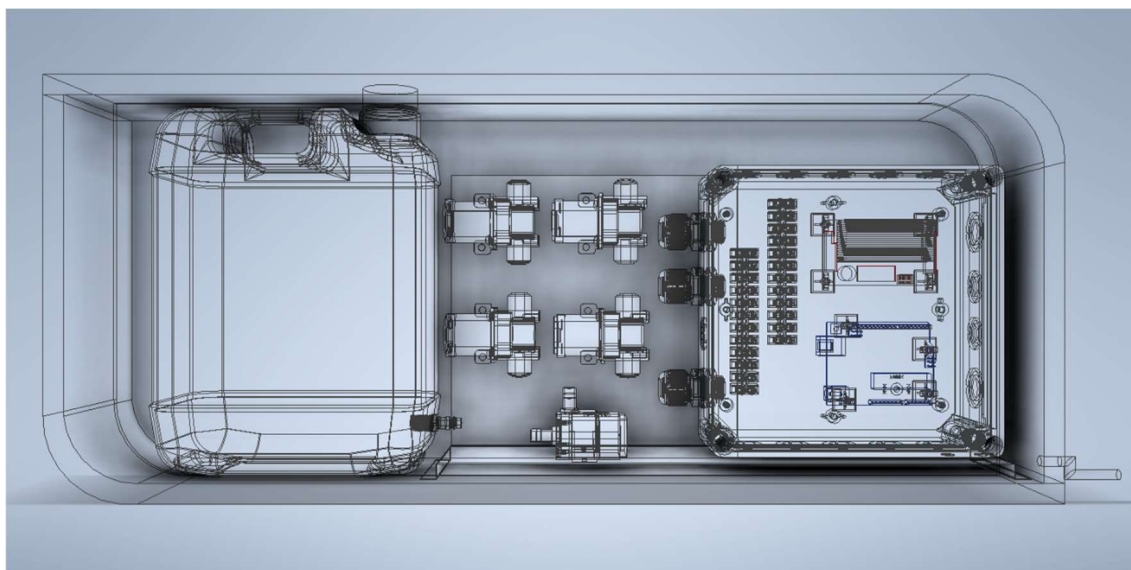
Kuva 15. Pumpun ja säiliön välinen letkuyhteys

Laitteessa kanisterin täyttö tapahtuu kanisterin omasta täyttöreikästä. Täyttämistä varten puukoteloon on tehty reikä, joka näkyy kuvassa 16.



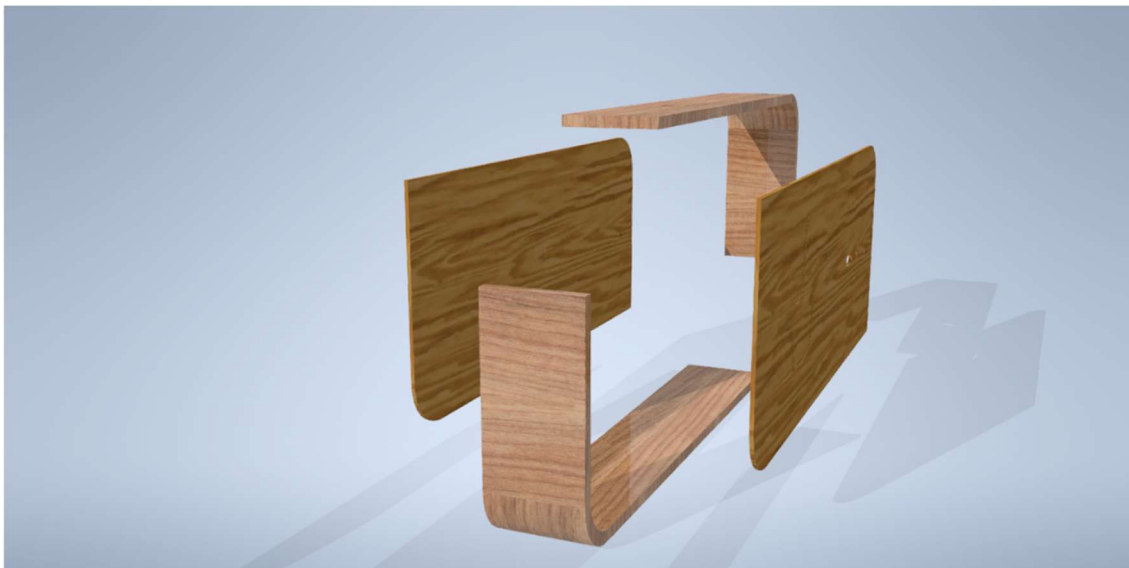
Kuva 16. Veden kulkureitti

Kanisterin tulee sopia tiiviisti laitteen koteloon sen liikkumisen estämiseksi. Liikkumisen estämistä voidaan myös parantaa lisäämällä kotelon sisälle metallisia kulmapaloja tai koteloon voidaan tehdä puusta erilaisia tukia kanisterille.



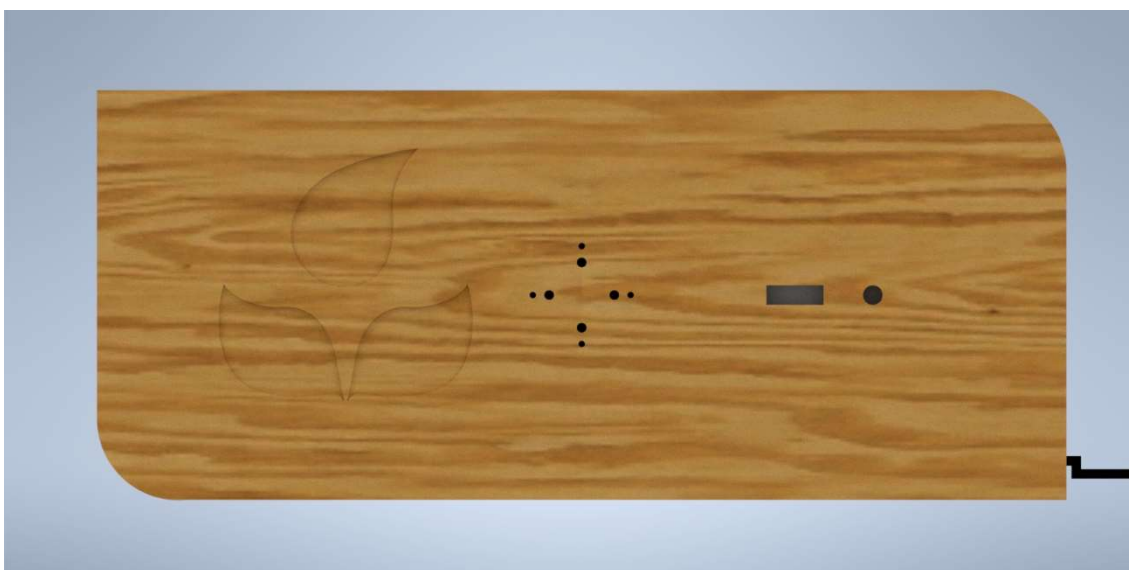
Kuva 17. Asettelusta läpinäkyvä mallinnus

Kun laitteella on kaikki olennaisimmat komponentit, voidaan niitä käyttää kotelon mitoittamiseen. Kotelon suunnitteleminen kannattaisi jättää ammattitaitoiselle muotoilijalle, varsinkin kuluttajatuotteiden kohdalla, mutta mekaniikkasuunnittelija voi alustavasti tehdä kotelon tuotteen esittelyä varten. Kuvassa 18 näkyy alustava idea kotelolle. Yksinkertaiset muodot ovat ajattomampia, ja puusta tehty kuori sopii hyvin kasviteemalle.



Kuva 18. Puiset ulkokuoret

Yksityiskohdat jäävät laitteen etupuolelle, jotka näkyvät kuvassa 19. Etupuolen kuvassa vasemmalla on kuvitteellinen logo, joka koostuu kahdesta kaiverretusta lehdestä ja vesipisarasta. Kuvassa näkyy keskellä rei'itys vesiletkuja ja kosteusanturien johtoja varten. Oikealla kuvassa näkyy säätö- ja käynnistyspainikkeita varten reiät.



Kuva 19. Laitteen kotelo edestäpäin



Kuva 20 (yllä) ja 21 (alla). Esittelykuvat

Laitteen kotelo näkyy valmiina esittelykuvissa 20 ja 21. Kuvat on mallinnettu ja käsitelty 3D-mallinnusohjelmalla. Kuvassa 28 on myös muuntaja ilman verkkovirtaan kytkettävää johtoa ja pöytä auttamaan laitteen koon hahmotusta. Lopputulos ei ole ammattitaitoisen muotoilijan luoma, mutta toimivaksi todettu systeemi on saatu 3D-suunniteltua koteloon ja laite on esittelyä varten valmis.

Kuvista ja suunnittelusta puuttuvat putket sekä painikkeet, sillä niiden valintaan vaikuttaa laitteen tarkempi muotoilu, jota ei viedä enempää eteenpäin tässä opinnäytetyössä. Tuloksena on laite, joka on esittelykelpoinen esimerkiksi yritykselle tai muulle rahoittajalle mahdollisesta tuoteideasta. Jos sopiva rahoittaja löytyy tai tuotetta haluaa itse valmistaa, niin sitä pitää jatkokehittää.

3.7 Jatkokehitys

Tuote on vasta ideatasolla valmis, ja sen tuominen markkinoille vaatii paljon jatkokehittämistä. Tässä osiossa käydään läpi jatkokehityksen kannalta olennaisimmat asiat ja seuraavat askeleet kasvienkastelujärjestelmän kehitysprosessissa.

Suunniteltuun laitteeseen tulisi tehdä teknisiä yksityiskohtia ja työstää 3D-mallin mittoja tarkemmaksi. Letkujen lisäksi tulisi miettiä johdotusta ja niiden kulkureittejä sekä osien kiinnitysmenetelmiä. Teknisiin yksityiskohtiin kannattaa ottaa DFMA-periaatteet (Design For Manufacture and Assembly) kustannushinnan laskemiseksi. DFMA-periaatteet ovat suunnitteluperiaatteita valmistuksen ja kasaamisen helpottamiseksi (Boothroyd, Dewhurst & Knight 2010, 1), mikä puolestaan madaltaa kustannushintoja. Teknisiin yksityiskohtiin sisältyy myös osavaihto, joita voidaan tarkastella, jos toimintaa tai hintaan halutaan vaikuttaa. Teknisten yksityiskohtien ohella joudutaan miettimään valmistusta ja materiaalivalintoja tarkemmin. Materiaalivalinnoilla voidaan vaikuttaa hyvin paljon tuotteen kustannushintaan (Boothroyd ym. 2010, 29 - 31) kuten myös tuotteen kestävyys- ja elinkaareen.

Laitteen jatkokehityksessä on myös tehtävä kompromisseja. Turvallisuuden kartoittamiseksi on tehtävä riskianalyysi, jonka jälkeen laitetta voidaan joutua muuttamaan hyvin paljon, ellei näitä asioita ole jo mietitty etukäteen. Automaattinen kotilaite, joka jää päälle omistajan ollessa poissa, ei välttämättä saa hyväksyntää vakuutusyhtiöiltä niin helposti. Hyväksyttävän dokumentaation tekeminen vaatii konedirektiivien ja paikallisten standardien tutkimista. Nämä asiat erottavat harrastelijoiden projektit tuotekehityksestä kuluttajatuotteista.

Teknisten yksityiskohtien ja turvallisuuden sekä laadun lisäksi on mietittävä tuotteen muotoilu, sillä se on hyvin olennainen osa laitteen jatkokehittämistä. Laite on tarkoitettu kuluttajille myytäväksi kotikoneena, ja sen halutaan vetävän kuluttajia puoleensa. Tästä syystä muotoilu kannattaa jättää ammattilaisille suunnittelemaan sen ulkomuotoa ja parantamaan käytettävyyttä. Vaikeakäyttöistä ja ulkomuodoltaan epämiellyttävää tai kaiken tyyliseen sisustukseen sopimatonta laitetta harva haluaa kotiinsa, varsinkin jos tuotemarkkinoilla on muita vaihtoehtoja.

4 YHTEENVETO

Opinnäytetyön tavoitteena oli tuotekehitysprosessin yksinkertaistetun innovaatio-osuuden soveltaminen kastelujärjestelmän suunnittelussa. Työssä käytiin läpi tuotekehityksen teoriaa, josta otettiin innovaatioprosessin olennaisimmat elementit ja luotiin niistä yksinkertaisempi malli. Yksinkertaista mallia käytiin läpi käytännön projektin avulla ja lopputuloksena saatiin esittelykelpoinen tuoteidea.

Vaikka aiheesta on paljon kirjallisuutta, kaikki tarvittavat tiedot ja menetelmät löytyivät Karl T. Ulrichin ja Steven D. Eppingerin teoksesta. Muistakin lähteistä löytyi samaa tietoa kuin mainittujen kirjailijoiden teoksesta, mutta niissä demonstroitiin hieman eri menetelmillä ja näkökulmilla. Yksinkertaistettu malli on hyvin supistettu versio alkuperäisestä prosessista, mutta tuotti halutun tuloksen.

Prosessin aikana ongelmia ilmaantui esimerkiksi prototyyppiä rakentaessa, kun alkuperäinen tavoite oli pitää kustannushinta alle annetun määritelmän arvon eli 100€. Osa etsiessä joutuu tyytymään yksilöhintaan, mutta valmistusvaiheessa suurina määrinä tilatut osat saattavat olla hieman halvempia. Tämän osalta ei kuitenkaan onnistuttu. Toinen ongelma oli kokemuksen puute, mikä hidasti kehitysprosessia merkittävästi. Suunnitteluprosessista ja prototyypin rakentamiseen liittyvistä aiheista piti etsiä iso määrä tietoa ja kokemuksen puutteen vuoksi tiedonhakua ei osattu kohdistaa oikein. Tästä löytyy kuitenkin positiivisia puolia, sillä koko opinnäytetyön sisältö projektia myöten oli hyvin opettavainen kokemus tuotekehitysmaailmasta.

Luotua tuoteideaa kasvienkastelujärjestelmästä voidaan esitellä etsiessä rahoittajaa tai tuotteen jatkokehittämiseen lisää suunnittelijoita, koska kuten opinnäytetyössä on mainittu, yhden henkilön kehitysprojektista jää laajempi näkemys ja kriittisyys vähemmälle. Prosessia voitaisiin kuitenkin käyttää ideoiden kehittämisen opettelemiseen. Toimiva prototyyppi todistaa laitteen systeemin toimivuuden ja 3D-malli sen mahdollisuudet ja potentiaalin jatkokehitykseen, josta mahdollisesti sitten siirryttäisiin kuluttajamarkkinoille. Tavoitteisiin päästiin, mutta samalla voidaan todeta, että yksin toteutettu kehitystyö ei ole kannattavaa. Kehitystyössä päästään pidemmälle ja saadaan enemmän tuloksia ryhmässä, jossa on monialainen osaaminen ja eri kokemustason omaavia henkilöitä.

Lopputulos osoittaa prosessin toimivuuden, mutta ei takaa sen laatua. Lopullisen arvioinnin tuotetusta tuotteesta antaa ostopäätöksellään kuluttajat, kuten aina tuotekehityksen asiakaslähtöisessä suunnittelussa.

LÄHTEET

Painetut lähteet:

Boothroyd, G., Dewhurst, P. & Knight, W.A. 2010. Product Design for Manufacture and Assembly. 3rd edition. New York: CRC Press.

Hietikko, E. 2008. Tuotekehitystoiminta. Kuopio: Savonia ammattikorkeakoulu.

Ulrich, K.T. & Eppinger, S.D. 2012. Product Design and Development. 5th edition. New York: The McGraw-Hill Companies.

Verworn, B. 2005. Die frühen Phasen der Produktentwicklung, Eine empirische Analyse in der Mess-, Steuer- und Regelungstechnik. Wiesbaden: Springer Fachmedien Wiesbaden GmbH.

Elektroniset lähteet:

Arduino SA. 2020a. Arduino UNO REV3 [viitattu 1.4.2020]. Saatavissa:
<https://store.arduino.cc/arduino-uno-rev3>

Arduino SA. 2020b. Introduction [Viitattu 1.4.2020]. Saatavissa:
<https://www.arduino.cc/en/Guide/Introduction>

Arduino SA. 2020c. Tutorials. Transistor Motor Control [viitattu 1.4.2020]. Saatavissa:
<https://www.arduino.cc/en/Tutorial/TransistorMotorControl>

LIITTEET

Liite 1 Kastelujärjestelmän prototyyppiin ladattu ohjelma kopiointia varten

Liite 2 Prototyypin kytkentäkaavio

Kastelujärjestelmän prototyyppiin ladattu ohjelma kopiointia varten:

```
// Noudattaen hyvää ohjelmointitapaa, kirjoitin koodin englanniksi.
// in first part we declare the variables
int FWD_signal = 13; /* PIN - pump motor: forward signal */
int RUN_signal = 12; /* PIN - green led light if master button is on */
int WET_signal = 8; /* PIN - red led if plant is already wet */
int MASTERBUTTON = 4; /* PIN - start code */
int DAYS = 3; /* amount of for-loops (1 loop should be 1 day) */
int PUMP_TIME = 2000; /* pumping time in ms */
int WAIT_DAY = 86400000; /* pause between sensing the moisture in ms for 24
hours */

// moisture sensor variables
int sensorPin = A0; /* PIN - Analog pin for moisture sensor */
int sensorValue = 0; /* return value for moisture amount (1024 is dry) */
int MoistureLimit = 700; /* Moisture level limit for dryness, over 700 dry;
activate pumping */

// in the second part we declare which variables are pin numbers and whether
they are outputs or inputs
// setup() function only runs once; upon arduino start or restart
// serial.begin is for tuning the frequency between the terminal and arduino to
monitor moisture values
void setup() {
  Serial.begin(9600);
  pinMode(FWD_signal, OUTPUT);
  pinMode(RUN_signal, OUTPUT);
  pinMode(WET_signal, OUTPUT);
  pinMode(MASTERBUTTON, INPUT);
}

/* void loop is the code that runs repeatedly while arduino is on. Here
the first IF-statement is essential for the rest of the code to work.
MASTERBUTTON enables the first IF-statement. Then the code will start
a FOR-loop for set amount of days. One FOR loop will last 24 hours to
ensure that it checks the sensor data once per day. After the code has
finished, the MASTERBUTTON will reset and the code will wait until
MASTERBUTTON is enabled for the code to start working again.*/
void loop() {

  if (digitalRead(MASTERBUTTON) == HIGH) {

    digitalWrite(RUN_signal, 1); // turn on green led light

    // LOOP THROUGH GIVEN AMOUNT OF DAYS

    for (int i = 1; i <= DAYS; i++) {

      // SETUP 1: sense moisture

      sensorValue = analogRead(sensorPin); // activate & read sensor
```

```

Serial.println("Analog Value: ");
Serial.println(sensorValue); // print sensed value to console

// SETUP 2: decide if its wet or dry

if (sensorValue > MoistureLimit) { // limit hardcoded 700 > dry plant

    // IT IS DRY

    digitalWrite(FWD_signal, 1); // turn on pump
    delay(PUMP_TIME); // pump as long as pump time value

    digitalWrite(FWD_signal, 0); // turn off pump

    // for checking loops and which one it is running we need:
Serial.println(i);
    Serial.println(sensorValue);
}
else {
    // IT IS WET

    digitalWrite(WET_signal, 1); // light up red led
    delay(1000); // wait 1 second
    digitalWrite(WET_signal, 0); // turn off red led
}

// STEP 3: WAIT FOR THE NEXT DAY (unless its the last day)

if (i < DAYS) {
    delay(WAIT_DAY); // wait for a day to pass
}

}

// END OF DAYS IN FOR-LOOP

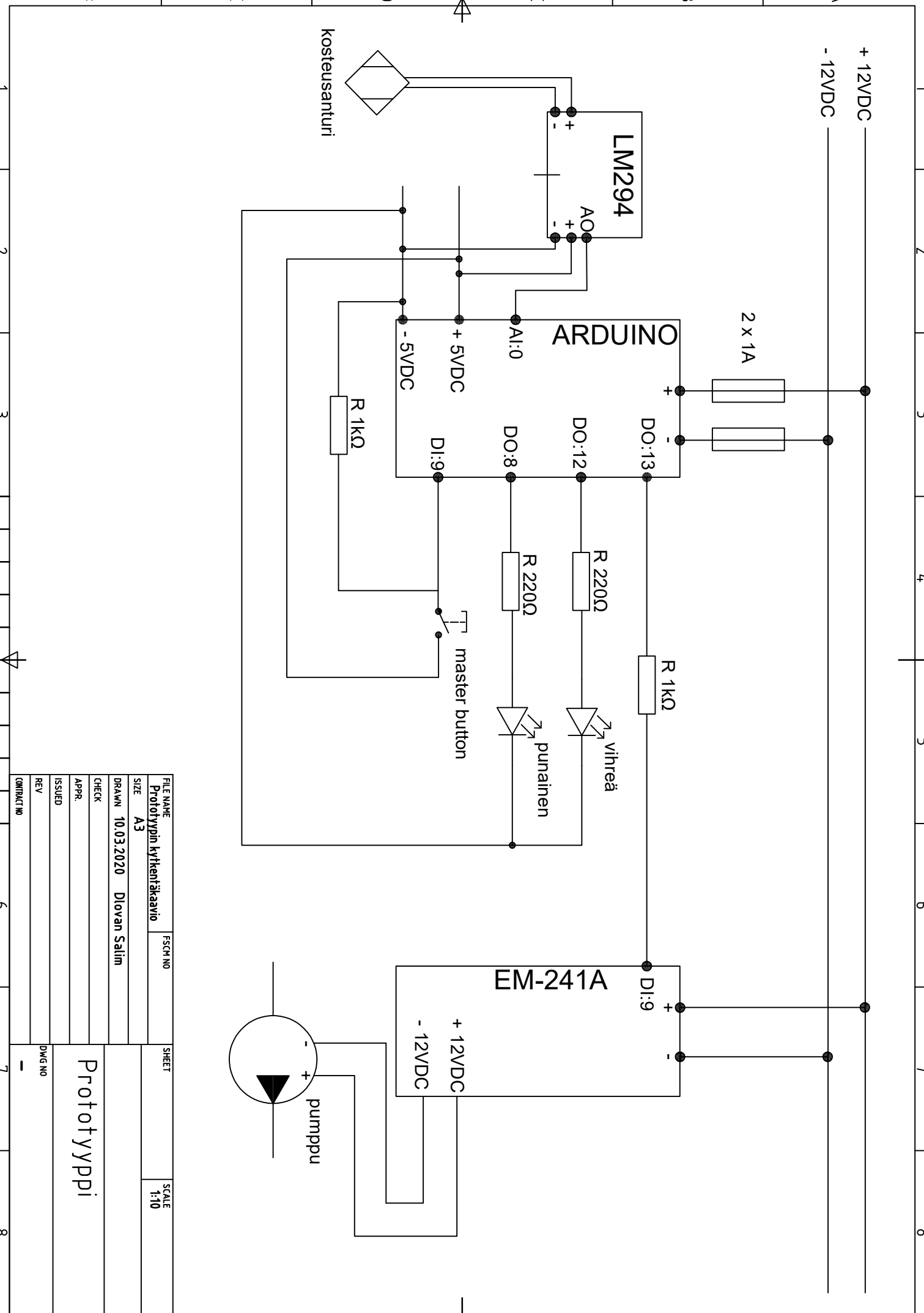
digitalWrite(FWD_signal, 0); // turn off pump

digitalWrite(RUN_signal, 0); // turn OFF green light

}

// END OF PROGRAMM
}

```



1 2 3 4 5 6 7 8

FILE NAME	Prototyypin kytkentäkaavio	FSCM NO		SHEET		SCALE	1:10
SIZE	A3						
DRAWN	10.03.2020	Dlovan Salim					
CHECK							
APPR.							
ISSUED							
REV							
CONTRACT NO							

Prototyyppi