

Juha Tienvieri

RADIO-OHJATTAVA RUOHONLEIKKURI

Työnnettävän akkukäyttöisen ruohonleikkurin muuntaminen radio-ohjattavaksi ruohonleikkuriksi

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tieto- ja viestintäteknikan koulutusohjelma
Syyskuu 2018**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	Aika Syyskuu 2018	Tekijä/tekijät Juha Tienvieri
Koulutusohjelma Tieto- ja viestintäteknikka		
Työn nimi RADIO-OHJATTAVA RUOHONLEIKKURI. Työnnettävän akkukäyttöisen ruohonleikkurin muuntaminen radio-ohjattavaksi ruohonleikkuriksi		
Työn ohjaaja Hannu Ala-Pönttiö	Sivumäärä 44 + 2	
Työelämäohjaaja Juha Itäpalo		
<p>Opinnäytetyön tavoitteena oli muuntaa työnnettävä akkukäyttöinen ruohonleikkuri radio-ohjattavaksi ruohonleikkuriksi. Opinnäytetyön lopputuloksena tuli valmis prototyyppi yksityiskäyttöön ja, se on suunnattu ison ja vaikeamaastaisen nurmikon leikkaamiseen.</p> <p>Opinnäytetyön teoriaosassa on käyty läpi teoriaa radio-ohjauksesta, Arduino-mikrokontrollerista sekä tasasähkömoottoreiden ohjaustavoista. Käytännön osuus on jaettu mekaaniseen suunnitteluun ja ohjaussuunnitteluun. Mekaanisessa suunnittelussa on mitoitettu työssä käytettävät vääntömoottorit sekä hihnat ja hammashihnapyörät. Ohjaussuunnittelussa on käyty läpi radio-ohjauksen liittäminen Arduinoon, ohjausohjelman teko, ruohonleikkurin käynnistäminen radio-ohjauksen kautta ja turvallisuuden huomioiminen työssä. Viimeisessä luvussa on käsitelty työn lopullista onnistumista ja mahdollisia parantamismahdollisuuksia.</p>		
Asiasanat Arduino-mikrokontrolleri, mekaaninen suunnittelu, ohjaussuunnittelu, radio-ohjaus, tasasähkömoottori		

ABSTRACT

Centria University of Applied Sciences	Date September 2018	Author Juha Tienvieri
Degree programme Information and communications technology		
Name of thesis RADIO CONTROLLABLE LAWN MOWER. Pushable battery-powered lawn mower transformation to radio controllable lawn mower		
Instructor Hannu Ala-Pöntiö	Pages 44 + 2	
Supervisor Juha Itäpalo		
<p>The object of this thesis was to transform pushable battery-powered lawn mower to radio controlled lawn mower. The end result of this thesis was a ready prototype for private use and it is targeted to big and difficulty terrain's lawn cutting.</p> <p>The theoretical part of this thesis has dealt with the theory of radio controlling, Arduino microcontrollers and control methods of direct current motors. The practical part is divided between mechanical design and control design. The torque motors, which have been used in the work, as well as the belts and the belt gears have been dimensioned in the mechanical design part. The control design part has focused on connecting the radio controller to Arduino, making the control program, starting the lawn mower with radio control and on the safety planning. The last chapter has dealt with the work's final success and potential improvement possibilities.</p>		
Key words Arduino microcontroller, control design, direct current motor, mechanical design, radio control		

KÄSITTEIDEN MÄÄRITTELY

AD-muunnin	Analog-to-digital converter. Analogia-digitaalimuunnin.
AM	Amplitude modulation. Amplitudimodulaatio.
DC-moottori	Direct current motor. Tasavirtamoottori.
FM	Frequency modulation. Taajuusmodulaatio.
MCU	MicroController Unit. Mikrokontrolleri.
Nomogrammi	Graafinen laskentakaavio. Käytetään matemaattisen funktion likimääräiseen graafiseen laskentaan.
PWM	Pulse-Width Modulation. Pulssinleveysmodulaatio.
RC	Radio Control. Radio-ohjaus.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
2 RADIO-OHJAUS.....	2
2.1 Lähetin	3
2.2 Vastaanotin	4
3 ARDUINO-MIKROKONTROLLERI	6
4 DC-MOOTTOREIDEN OHJAUSTAVAT	9
4.1 DC-moottorin ohjaus magnetointivirran avulla	10
4.2 DC-moottorin ohjaus ankkurijännitteen avulla.....	10
4.3 DC-moottorin ohjaus pulssinleveysmodulaation avulla	11
5 MEKAANINEN SUUNNITTELU	14
5.1 Alustavat toimenpiteet	16
5.2 Vääntömoottoreiden valinta.....	18
5.3 Hihnan mitoitus ja hammashihnapyörien valinta	21
6 OHJAUSSUUNNITTELU	29
6.1 Uusien laitteiden hankinta.....	29
6.2 Arduinon moottorinohjainyksikön, vastaanottimen sekä vääntömoottoreiden kytkeminen Arduinon.....	33
6.3 PWM-signaalin määrittely Arduinolla	35
6.4 Ohjausohjelman teko	36
6.5 Ruohonleikkurin käynnistäminen	38
6.6 Turvallisuussuunnittelu	40
7 YHTEENVETO	42
LÄHTEET	43
LIITTEET	
KUVIOT	
KUVIO 1. Lähettimen toimintaperiaate.....	4
KUVIO 2. Vastaanottimen toimintaperiaate.....	5
KUVIO 3. Magnetointivirran ohjausmenetelmä.....	10
KUVIO 4. Ankkurijännitteen ohjausmenetelmä.....	11
KUVIO 5. Pulssinleveysmodulaation periaate	12
KUVIO 6. Pulssisuhteiden synkronointi Arduinon 0–255 skaalausalueelle	13
KUVIO 7. Nomogrammi	25
KUVIO 8. Sauvojen asentojen PWM-arvot.....	36
KUVIO 9. Ruohonleikkurin ohjaus, radio-ohjaimen sauvojen asennot	37
KUVAT	
KUVA 1. Ampiltudi- ja taajuusmodulaation lähetteen erot kaavioina.....	2

KUVA 2. Tasasähkömoottorin rakenne.....	9
KUVA 3. FXA:n akkukäyttöinen ruohonleikkuri.....	14
KUVA 4. Ruohonleikkurin 40 Voltin akkutila.....	16
KUVA 5. Eturattaat, jotka on siirretty taakse.....	17
KUVA 6. Vapaan pyörän kiinnitys ruohonleikkurin runkoon.....	17
KUVA 7. E192.12.18-DC-moottorit.....	19
KUVA 8. Vääntömoottorien kiinnitys kiinnitysnaulauslevyyn.....	21
KUVA 9. Kuvassa ovat sorvatut välilaidat ja vetopyörät.....	23
KUVA 10. Vetopyörä, välilaidat ja ruohonleikkurin pyörä.....	23
KUVA 11. Hihnat ja hammashihnapyörät.....	27
KUVA 12. Hihnojen kiristys.....	28
KUVA 13. Arduino Motor Shield.....	30
KUVA 14. Arduino Uno Rev 3.....	31
KUVA 15. Futaban T4YWD-nelikanavainen radio-ohjain ja R213GF-E-nelikanavainen vastaanotin.....	32
KUVA 16. Powercellin 12 voltin ladattava lyijyakku.....	34
KUVA 17. Kuvassa Arduino Uno Rev 3 ja Arduino Motor Shield ovat liitettyinä toisiinsa.....	34
KUVA 18. SRD-05-VDC-SL-C-relekortti.....	39
KUVA 19. Asennustarvikkeita, LEG-5F-rele sekä diodi 1N4007.....	39
KUVA 20. HC-SR03-ultraäänianturi.....	40
KUVA 21. Ruohonleikkurin testausajo.....	41

TAULUKOT

TAULUKKO 1. FXA-ruohonleikkurin tekniset tiedot.....	15
TAULUKKO 2. E192.12.18-DC-moottorin tekniset tiedot.....	19
TAULUKKO 3. Arduino Motor Shield tekniset tiedot.....	30
TAULUKKO 4. Arduino Uno Rev 3 tekniset tiedot.....	31
TAULUKKO 5. Futaban lähettimen ja vastaanottimen tekniset tiedot.....	33
TAULUKKO 6. HC-SR04 ultraäänianturin tekniset tiedot.....	40

1 JOHDANTO

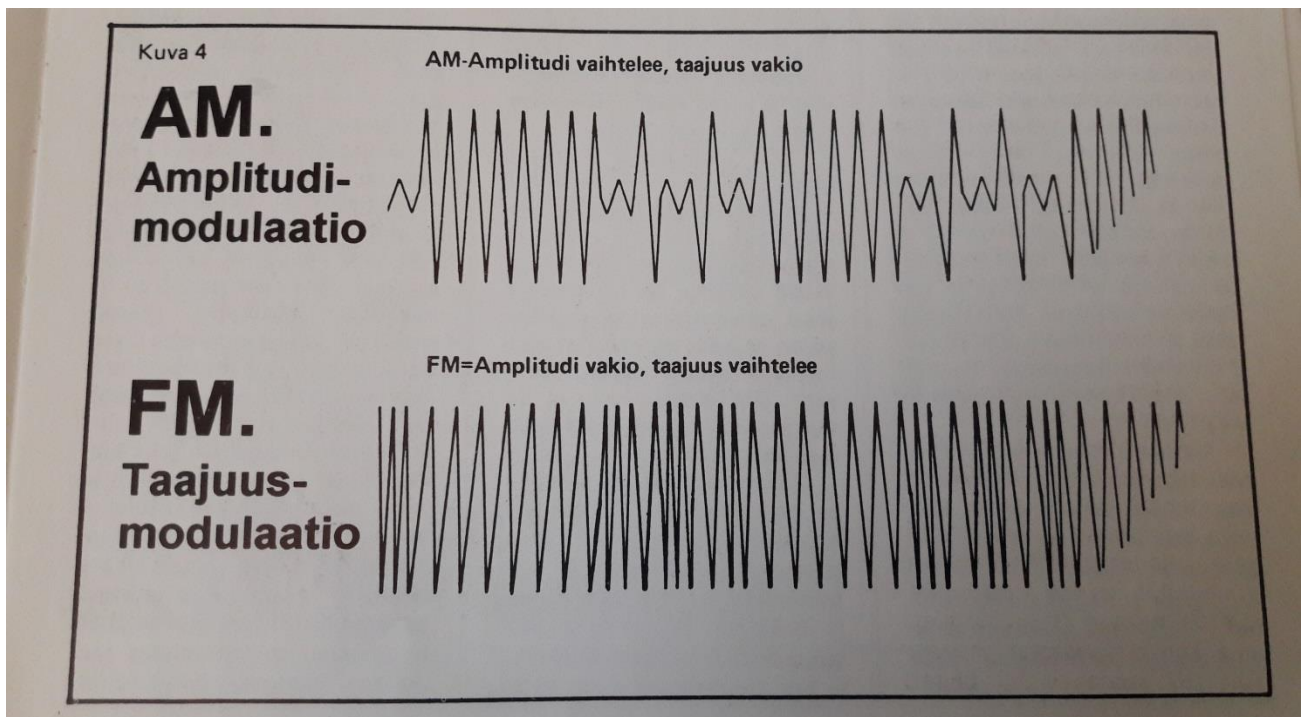
Opinnäytetyön tavoitteena on muuntaa työnnettävä akkukäyttöinen ruohonleikkuri radio-ohjattavaksi ruohonleikkuriksi. Työn tilaus tuli sukulaiseltani, jolla on iso ja vaikeamaastoinen tontti ja jonka nurmikon leikkaamiseen eivät perinteiset markkinoilla saatavissa olevat rajatulla alueella toimivat robottiruohonleikkurit sovellu. Kokonaiskustannuksiltaan radio-ohjattavan ruohonleikkurin prototyypin tekeminen tulisi myös olemaan keskimäärin halvempi vaihtoehto kuin valmiin robottiruohonleikkurin ostaminen. Radio-ohjattavan ruohonleikkurin nopeus pyrittiin saamaan myös nopeammaksi kuin robottiruohonleikkureilla, jotta se vastaisi käyttäjän omaa työntövuuhua. Työn kannalta oleellista oli myös saada radio-ohjattava ruohonleikkuri mahdollisimman turvalliseksi, jotta sillä ei aiheutuisi henkilövahinkoja.

Tämän opinnäytetyön teoriaosuudessa käsitellään ensiksi radio-ohjausta. Siinä käsitellään radio-ohjauksessa käytettäviä modulointitapoja ja radio-ohjauspakettiin kuuluvia tärkeimpiä komponentteja. Seuraavana asiana teoriaosuudessa käsitellään Arduino-mikrokontrolleria, jota on hyödynnetty tämän opinnäytetyön ohjaussuunnittelussa sekä ruohonleikkurin turvallisuuden huomioimisessa. Kolmantena asiana teoriaosuudessa on käsitelty tasasähkömoottoreiden ohjaustapoja, joista ensimmäisenä käydään läpi tasasähkömoottorin ohjaaminen magnetointivirralla. Toisena ohjaustapana käydään läpi tasasähkömoottorin ohjaaminen ankkurijännitettä muuttamalla. Viimeisenä tapana käydään läpi tasasähkömoottorin ohjaaminen pulssinleveysmodulaatiotekniikan avulla, jota hyödynnettiin tässä opinnäytetyössä.

Opinnäytetyön käytännön osuus on jaettu mekaaniseen suunnitteluun ja ohjaussuunnitteluun. Mekaaninen suunnittelu on rajattu tasasähkömoottoreiden mitoittamiseen sekä hihnojen ja hammashihnapyörien mitoittamiseen, koska opinnäytetyön tekijällä ei ole käytännön kokemusta mekaanisesta suunnittelusta ja koska se ei myöskään ole opinnäytetyön tekijän opiskeltava ala. Ohjaussuunnittelussa käydään läpi vaihe vaiheelta radio-ohjauksen liittäminen Arduinoon, ohjausohjelman tekeminen, ruohonleikkurin käynnistäminen radio-ohjauksen kautta. Lopuksi käsitellään turvallisuuden huomioiminen työssä ottamalla huomioon radio-ohjauksen kantaman menettäminen, törmäysvaara kohteeseen sekä muita mahdollisia inhimillisistä virheistä aiheutuvia vaaratilanteita.

2 RADIO-OHJAUS

Radio-ohjaus tulee sanoista Radio Control, joista käytetään lyhennystä RC. Radio-ohjauksessa ilmenee kaksi käsitettä, jotka ovat nimeltään suhteellisuus ja digitaalisuus. Suhteellisuuden merkitys ilmenee siten, että lähettimellä muutetaan ohjausliikettä ja verrannollisesti vastaanottopäässä tapahtuu vastaavanlainen ohjaustapahtuma. Digitaalisuudella tarkoitetaan ohjaussignaalin lähettämistä pulssijonoina. Mikäli ohjaussignaalit lähetettäisiin analogiasignaaleina, aiheutuisi siitä enemmän ulkoisia häiriöitä. Ulkoiset häiriöt on saatu pienenemään myös siirtymällä amplitudimodulaatiosta taajuusmodulaatioon ja siitä edelleen digitaaliseen spektrin modulaatioon. (James 2017; Meder 1993, 5.)



KUVA 1. Amplitudi- ja taajuusmodulaation läheteiden erot kaavioina (mukaihen Meder 1993, 7)

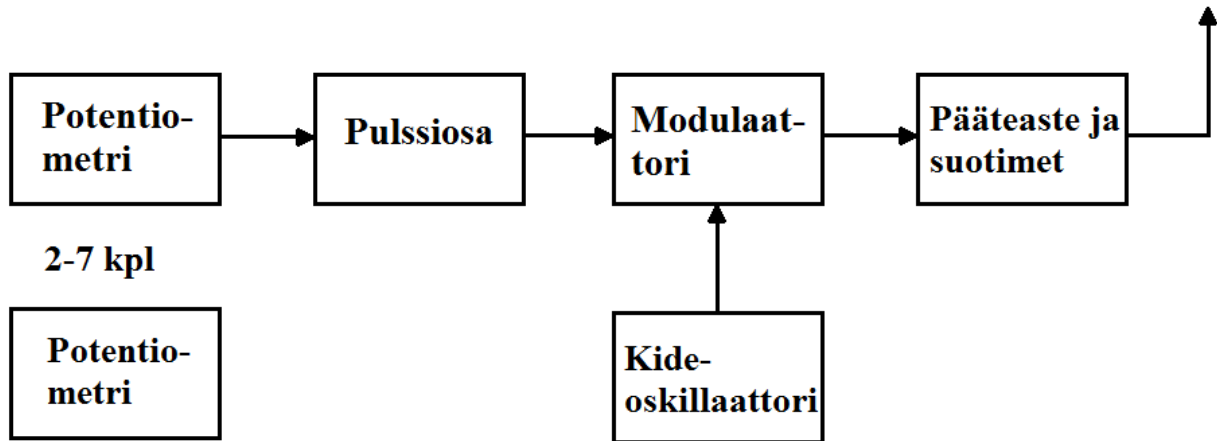
Lähetettäessä digitaalisignaalia amplitudimodulaatiossa (KUVA 1) katkotaan lähtevää radiosignaalia. Tämä voi kasvattaa radion aiheuttamia häiriöitä sekä ulkoisia häiriöitä. Ulkoiset häiriöt ilmenevät siten, että voimakkaammat radiolähteet voivat täyttää pulssien välit, ja tämän seurauksena vastaanotinpuoli ei enää tunnista oman lähettimensä signaalia. Taajuusmodulaatiossa lähettimen signaalin voimakkuus pysyy samanlaisena koko ajan. Häiriötilanne voi syntyä taajuusmodulaatiossa, mikäli lähete on saamantaajuinen, jatkuva sekä omaa lähetettä voimakkaampi. Digitaalisen spektrin modulaati-

on toteutustavat jaetaan taajuushyppely-tekniikkaan ja suorasekventointi-tekniikkaan. Taajuushyppelyssä lähetin lähettää kapeakaistaisen signaalin ja hyppää nopeasti seuraavaan taajuuteen. Suorasekventoinnissa järjestelmä toimii yhdellä taajuudella suurella alueella. (James 2017; Meder 1993, 7.)

Radio-ohjauslaitteissa on kanavia, joiden lukumäärä kertoo, kuinka monta erilaista toimintoa radio-ohjauslaitteella voidaan tehdä. Yleisimmät kanavien määrät radio-ohjauslaitteissa ovat 2–7 kanavaa. 2–3-kanavaiset laitteet on suunnattu aloittelijoille, ja ne ovat hinnaltaan halvempia. 2–3 kanavaisia laitteita voidaan käyttää yksinkertaisempiin laitteisiin ja yli nelikanavaisia vaativampiin ja kehittyneempiin laitteisiin. Radio-ohjauslaitteisiin kuuluvat tyypillisesti lähetin, vastaanotin, servoja kanavien määrän mukaan, asennustarvikkeita sekä tarpeen mukaan lisälaitteita, kuten vaihtokiteitä sekä sähkömoottorin nopeuden- ja suunnanvaihtolaitteita. (Meder 1993, 5.)

2.1 Lähetin

Lähetimissä on vaihdettava kide, joka määrittää lähettimen taajuuden. Yleinen käytettävä taajuus on 27 MHz. Käytettäviä taajuuksia ovat myös 49 MHz, 50 MHz, 72 MHz, 75 MHz sekä nykyisin paljon käytetty 2.4 GHz. 2.4 GHz -lähetimissä kidettä ei tarvitse vaihtaa. 2.4 GHz -lähetimet eivät myöskään häiritse toisiaan laajan taajuusalueen ansiosta. Lähettimen malleja ovat muun muassa perinteinen sauvoilla varustettu malli tai rattimalli. Kaksikanavaisissa lähetimissä oikeanpuoleinen sauva liikkuu sivuttaissuunnassa ja vasemmanpuoleinen sauva liikkuu pystysuunnassa. Sauvat on voitu varustaa keskitysrousella, joka palauttaa sauvan keskiasentoon, kun käyttäjä päästää sauvasta irti. Sauvat voidaan varustaa keskitysrousen sijaan myös rytmijarruilla. Rytmijarru pidättää sauvan sen hetkiseen asentoon, kun käyttäjä päästää sauvasta irti. Lähetimet on varustettu myös trimmausvivuilla, joita on yhtä monta kuin kanavaakin. Trimmausvivuilla siirretään sähköisesti vastaanotinpään ohjaussuureen keskipistettä. Nelikanavaisissa lähetimissä molemmat sauvat on yhdistetty sekä pystysuuntaisella että vaakasuuntaisella toiminnolla. Mikäli kanavia on esimerkiksi 5, 6 tai 7, ne voivat olla liukupotentiometrejä tai vaikkapa kolmiasentoisena kytkimenä. (James 2018; Meder 1993, 6.)



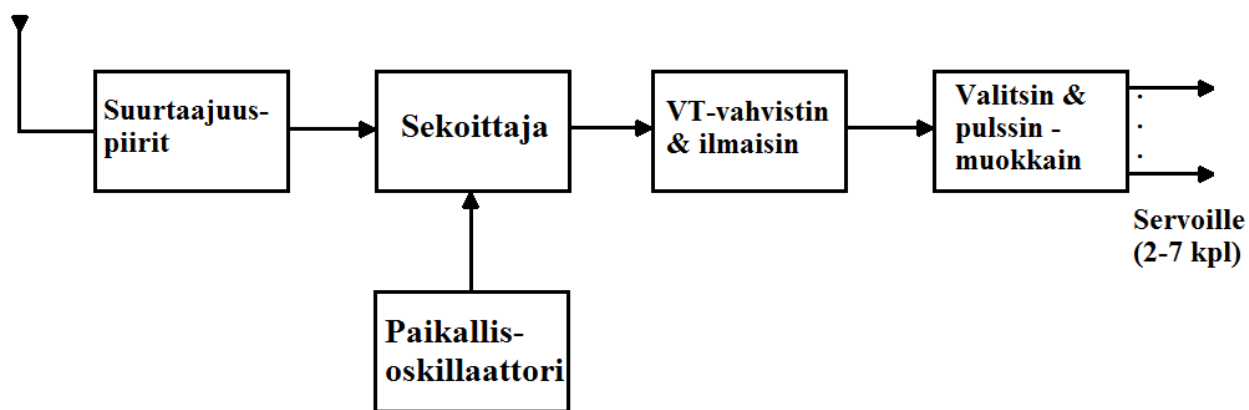
Lähettimen lohkokaavio

KUVIO 1. Lähettimen toimintaperiaate (mukaillen Meder 1993, 7)

Lähetin (KUVIO 1) toimii siten, että ohjauspotentiometrejä on yhtä monta kuin kanavia. Elektronisesti pulssiosa kiertää vuorottelevasti keräämässä lyhyen näytteen potentiometrin asennosta. Kideoskillaattori ohjaa radiotaajuutta suurtaajuusosassa. Modulaattori yhdistää pulssiosan ja kideoskillaattorin signaalit. Lopuksi signaali vahvistetaan pääteasemassa, minkä jälkeen se vielä suodatetaan ja toimitetaan antennin avulla ulos. (Meder 1993, 6–7.)

2.2 Vastaanotin

Vastaanottimet ovat pienikokoisia ja niissä käytetään mikropiirejä. Vastaanottimissa on kullattuja liittimiä, joita vastaanottimessa on radiolähettimen kanavien mukainen määrä. Vastaanottimen kide pysytään vaihtamaan. Lähettimien käyttämän pulssitekniikan ansiosta vastaanotin voi olla kanavamäärältään pienempi kuin lähetin. Vastaanotin (KUVIO 2) toimii siten, että ensiksi vastaanotin vastaanottaa antennipiirin kautta tulevan signaalin. Signaali summataan sekoittajassa vastaanotinkiteen ohjaaman paikallisoskillaattorin muodostamaan signaaliin. Tätä signaalia kutsutaan välitaajuudeksi. Sen jälkeen välitaajuus vahvistetaan ja ilmaistaan. Tässä vaiheessa välitaajuudesta on tullut pulssijono. Pulssijono siirretään valitsimeen ja pulssinmuokkaimen, jotka leikkaavat pulssit oikean korkuisiksi ja jakavat ne ohjauslaitteelle. Tahdistin nollautuu lepopulssin aikana. (Meder 1993, 8.)



Vastaanottimen lohko

KUVIO 2. Vastaanottimen toimintaperiaate (mukaillen Meder 1993, 8)

3 ARDUINO-MIKROKONTROLLERI

Mikrokontrolleri on pienikokoinen tietokone, joka sisältää prosessorin ja muistia. Mikrokontrollereita käytetään sulautetuissa järjestelmissä, joita ovat muun muassa kodinkoneet, matkapuhelimet, mikroaaltouunit ja GPS-paikantimet. Sulautetut järjestelmät sisältävät mikrokontrollereiden lisäksi sensoreita ja lähtöjä. Sensoreilla mitataan ympäristöstä esimerkiksi etäisyyttä, kiihtyvyyttä, valoa ja liikettä. Lähdoilla vaikutetaan fyysisesti ympäristöön ohjaamalla esimerkiksi moottoreita. Mikrokontrollerin sisältämän prosessorin ja muistin ansiosta sillä pystytään ajamaan käyttäjän itse tekemiä ohjelmia. Suurimmaksi osin sulautetut järjestelmät käsittelevät tarvittavat tiedot heti, koska erilaisten ohjauksien on tapahduttava välittömästi. Laitteen määrittelemisen sulautetuksi järjestelmäksi tai tietokoneeksi on vaikeaa, koska moniin sulautettuihin järjestelmiin on tullut tietokoneen kaltaisia ominaisuuksia. Sulautetuksi järjestelmäksi lasketaan yleensä laite, joka omaa rajoitetun skaalan ennalta määritellyjä toimintoja, joiden muuntaminen toisenlaiseksi on hankalaa. (Karvinen & Karvinen 2010, 9, 11, 34.)

Mikrokontrolleista käytetään lyhennettyä nimeä MCU, joka tarkoittaa MicroController Unit. Mikrokontrollereiden etuna esimerkiksi logiikkapiireihin nähden on se, että niillä pystytään toteuttamaan todella monimutkaisia toimintoja sekä suorittamaan laskutoimituksia, joita muilla piireillä ei pystytä tekemään. Osa mikrokontrollereista on rakennettu helposti liitettäväksi tietokoneeseen, josta sitä voidaan ohjelmoida halutulla tavalla. Asioiden hallitseminen ohjelmoinnilla helpottaa elektronisten laitteiden rakentamista. Suuret ongelmat pystytään mikrokontrollereiden avulla selvittämään askel askeleelta. Mikrokontrollerit sisältävät myös tulo- ja lähtöportteja, tulo- ja lähtöpinnejä sekä lisälaitteita, joihin kuuluvat yleensä AD-muunnin, sarjadataan ohjaamiseen tarkoitettu yksikkö, pulssinleveysmodulaationohjain, lämpötila-anturi ja ajastimia. Mikrokontrollereiden muisti on jaettu ohjelmamuistiin ja työmuistiin. (Hutasu.net 2017; Karvinen ym. 2010, 34.)

Mikrokontrollereiden prosessori toimii kellotaajuudella. Prosessori suorittaa yhden kellojakson aikana useita toimintoja. Ohjelmamuisti antaa käskyjä mikrokontrollerille. Ohjelmamuistista ladataan lukuja mikroprosessorin rekistereihin. Prosessori tulkitsee lukuja käskyiksi. Käskyinä siirretään dataa toiminoiksi, siirretään dataa toiseen paikkaan tai suoritetaan laskentoja. Mikrokontrollereiden ohjelmoimiseen käytetään tietokonetta, jolla käyttäjän tekemä ohjelma käännetään mikrokontrollerin ymmärtämään muotoon. Käännetty tiedosto ladataan mikrokontrollerin ohjelmamuistiin tietokoneen USB-portin tai ohjelmointityökalun kautta. Mikrokontrolleri voi suorittaa tämän jälkeen ohjelmaa. (Hutasu.net 2017.)

Mikrokontrollerin käyttämisessä ja ohjelmoinnissa tarvitaan varsinaisen mikrokontrollerin lisäksi tietokone, ohjelmointityökalut, ohjelmointilaite tai ohjelmointikaapeli sekä lisäkomponentteja. Ohjelmointityökaluina käytetään yleensä ilmaisia ohjelmointityökaluja tai kaupallisia versioita. Ohjelmointikaapelina on joko sarjakaapeli tai USB-kaapeli. Mikrokontrollerit ovat joko 4-, 8-, 16- tai 32-bittisiä. Mikrokontrollerin käyttöönotto aloitetaan yleensä asentamalla ohjelmointiympäristö, johon sisältyvät tarvittavien ohjelmointityökalujen asentaminen tietokoneelle sekä ohjelmointikaapelin ja mikrokontrolleri-alustan hankkiminen. (Hutasu.net 2017.)

Mikrokontrollereista sopivimmat vaihtoehdot aloittelijalle ovat Basic Stamp tai Arduino. Basic Stamp on kehitetty 1990-luvun alussa ja se on alan harrastajien suosima mikrokontrolleri. Basic Stamp käyttää ohjelmointikielenä Basiciä. Basic on helppokäyttöinen mutta rajoittunut ohjelmointikieli verrattuna C-kieleen, jota käytetään Arduinossa. Arduino muistuttaa käyttötarkoitukseltaan Basic Stampia, ja Basic Stampin perinteiset ongelmat on ratkaistu Arduinossa. Arduino on myös neljä kertaa halvempi Basic Stampiin verrattuna, ja Arduinon prosessori on tehokkaampi. Lisäksi Arduino sisältää enemmän muistia kuin Basic Stamp. Myös kooltaan Arduino on pienempi kuin Basic Stamp, ja sen takia se soveltuu paremmin moniin erilaisiin projekteihin. (Karvinen ym. 2010, 34–35.)

Arduino on kehitetty Ivrea Interaction Design Institute -oppilaitoksessa helpoksi työkaluksi, jota voidaan hyödyntää prototyyppien teossa, ja se on myös suunnattu opiskelijoille, joilla ei ole ennestään kovin suurta elektroniikka- ja ohjelmointitietämystä. Arduinon ohjelmisto (IDE) on avointa lähdekoodia. Arduinon ohjelmiston pystyy asentamaan erilaisille tunnetuille käyttöjärjestelmille kuten Windows, Linux ja Macintosh OSX. Suurin osa muista mikrokontrollereista pystytään asentamaan ainoastaan Windows-käyttöjärjestelmille. (Arduino a; Karvinen ym. 2010, 36.)

Arduinosta on saatavilla useita erilaisia malleja erilaisiin käyttötarkoituksiin. Yleisimpiä aloitusmalleja ovat mm. Arduino Uno, Arduino Leonardo ja Arduino 101. Sen lisäksi on saatavilla laajempia toimintoja sisältäviä malleja kuten Arduino Mega 2560. Lisäksi on olemassa moduuleja, jotka ovat pienempiä malleja alkuperäisistä kehitysalustoista. Pienempiä malleja on esimerkiksi Arduino Nano, joka soveltuu pieniin projekteihin. Sen haittapuolena on se, että se on isompaa kehitysalustaa herkempi häiriöille, ja pinnien merkintöjen lukeminen on hankalampaa. (Arduino c; Karvinen ym. 2010, 36, 47.)

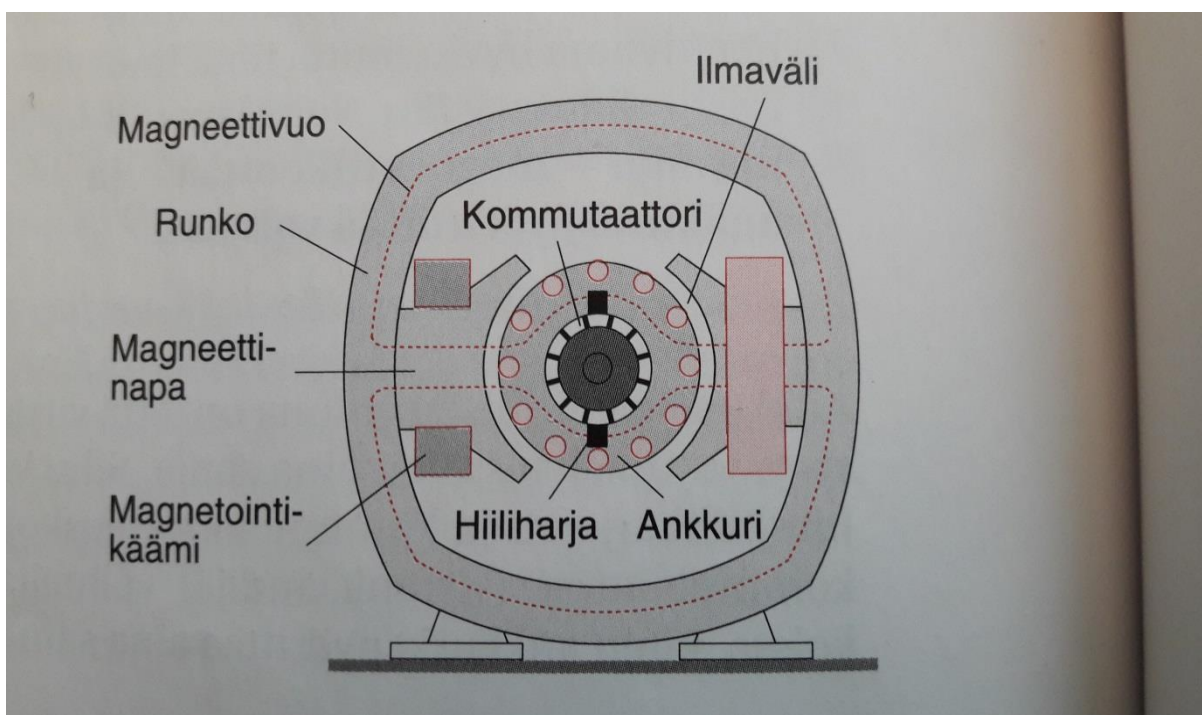
Tässä opinnäytetyössä käytetty Arduinon malli Arduino Uno Rev3 on suositelluin ja käytetyin Arduinon kehitysalustan malli aloitettaessa elektroniikka- ja ohjelmointityöskentelyä. Arduino Unosta löytyy myös eniten dokumentoitua tietoa verrattuna muihin Arduinon malleihin. Arduino Uno perustuu

ATmega328P:hen. Arduino Unossa on 14 digitaalisääntulo-/ulostulo-pinniä, joista kuutta voidaan käyttää PWM-ulostuloina. Arduino Unossa on myös 6 analogisisääntuloa ja 16 MHz:n kvartsikide. Arduino Unoön sisältyvät myös muun muassa virtaliitin, USB-yhteys ja reset-painike. Arduino Uno ottaa käyttövirtansa tietokoneen USB-portin, adapterin tai muun vastaavan 5 voltin tasajännitteen kautta. Arduino Uno on myös ensimmäisiä Arduino-kehitysalustoja, jotka hyödyntävät USB:tä. (Arduino d.)

Tässä opinnäytetyössä käytettiin lisäosista Arduino Motor Shield Rev3 -moottorinohjausyksikköä. Arduinon Moottorinohjausyksiköllä pystytään ohjaamaan muun muassa tasavirta- ja askelmoottoreita. Arduinon oma moottorinohjausyksikkö perustuu L298-mikropiiriin, joka mahdollistaa tasavirtamoottoreiden nopeudenohjaamisen sekä suunnanmuutoksen. Moottorinohjausyksikössä on kaksi erillistä kanavaa, jotka käyttävät yhteensä kahdeksaa pinniä Arduinosta. Kahdeksalla pinnillä voidaan ohjata kahden moottorin pyörimissuuntaa, jarruttaa moottoreita, muuttaa PWM-arvojen pulssisuhdetta sekä mitata moottoreiden läpi menevää virtaa. (Arduino b.)

4 DC-MOOTTOREIDEN OHJAUSTAVAT

Tasasähkömoottorin toimintaperiaate perustuu magnetismiin. Tasasähkömoottorin avulla muutetaan sähköinen energia mekaaniseksi energiaksi. Magneettikenttään sijoitetaan toisesta päästä avoin johdinsilmukka. Avoin johdinsilmukka on kiinnitetty tasasähkömoottorin akseliin. Pyörivää osaa kutsutaan roottoriksi. Johtimen päät on kiinnitetty kommutaattoriin, joka on mekaaninen virrankäytäjä. Kommutaattorissa on kaksi renkaan puolikasta, jotka ovat kommutaattorilamelleja. Ne on eristetty toisistaan. Hiiliharjojen kautta johdetaan virta johtimille. Kun ankkuri on pyörähtänyt siten, että hiiliharjat eivät ole enää kontaktissa kommutaattorin lamelleihin, pyöräyttää liikemäärämomentti ankkurin nollakohdan yli, minkä jälkeen kommutaattorin lamellit koskettavat vastakkaisia hiiliharjoja. Näin virran suunta käämissä vaihtuu, mutta voiman suunta pysyy samana. Lisäämällä johdinkierroksia ja pienentämällä ilmaväliä sijoittamalla johtimet raudasta tehdyn ankkurin uriin saadaan ankkurin pyöritysvoimaa kasvatettua. (Ahoranta 2014, 263–264; Inkinen, Manninen & Tuohi 2012, 136.)



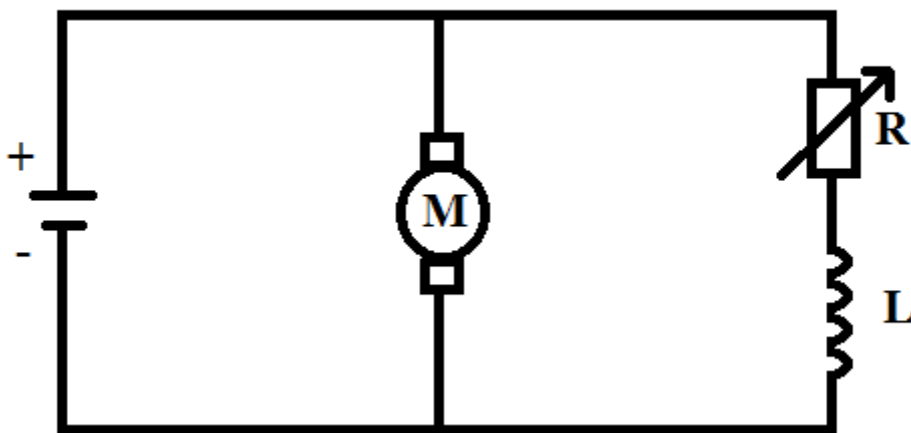
KUVA 2. Tasasähkömoottorin rakenne (mukaillen Ahoranta 2014, 264)

Kestomagneettimoottoreissa magneettikenttä muodostetaan kestopagneettien avulla. Kestomagneeteissa magneettikenttä on muuttumaton, ja tämän takia sitä ei voida ulkoisesti muuttaa. Kestomagneet-

timootorit koostuvat staattorista ja ankkurista. Kestomagneetit on asennettu staattorin ympärille kiinni. Kestomagneettimoottoreissa ei käytetä käämiä, joten tämän johdosta kestopagneettimoottoria ei voida ohjata magnetointivirran avulla. (Electrical4u.com 2018.)

4.1 DC-moottorin ohjaus magnetointivirran avulla

Yksi tasasähkömoottoreiden ohjaustapa on säätää tasasähkömoottorin magnetointivirtaa (KUVIO 3). Käämin magneettivuota muuttamalla saadaan muutettua tasasähkömoottorin pyörimisnopeutta. Käämin magneettivuota saadaan pienennettyä pienentämällä käämiin menevää virtaa. Se tapahtuu siten, että säätövastus kytketään sarjaan käämin kanssa, jonka avulla rajoitetaan virtaa. Tasasähkömoottorin nopeus kasvaa magneettivuon pienentyessä. Nopeus tässä tapauksessa ylittää moottorin normaalin nopeuden. Liiallinen nopeuden nostaminen vahingoittaa tasasähkömoottorin kommutaattoria. (Agarwal 2014; Daware 2016.)

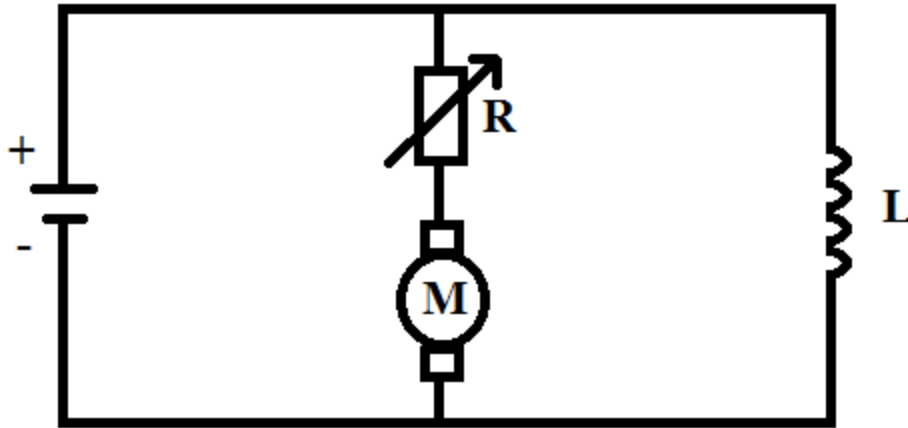


KUVIO 3. Magnetointivirran ohjausmenetelmä (mukaiillen Daware 2016)

4.2 DC-moottorin ohjaus ankkurijännitteen avulla

Toinen tasasähkömoottorin ohjaustapa on säätää tasasähkömoottorin ankkurijännitettä (KUVIO 4). Tämä tapahtuu siten, että säätövastus kytketään sarjaan tasasähkömoottorin ankkuriosan kanssa. Kun säätövastuksen resistanssia kasvatetaan, pienenee ankkurijännite. Ankkurijännitteen pienentyessä no-

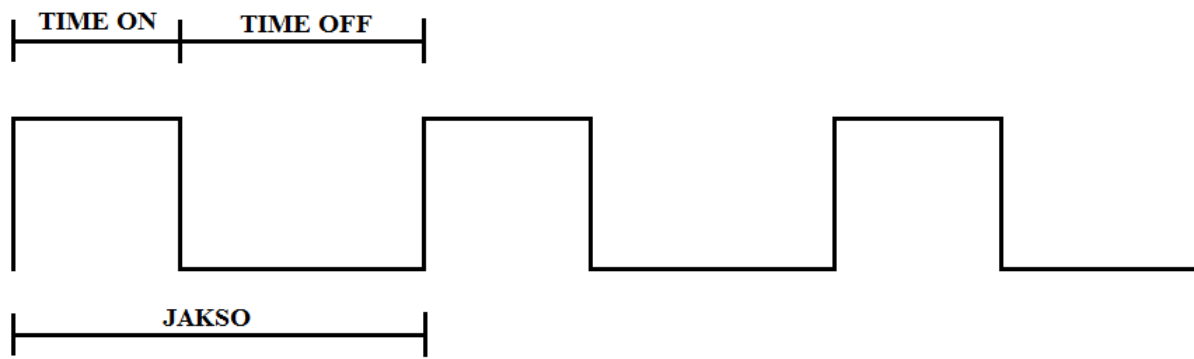
peuskin pienenee. Ankkurijännite on verrannollinen tasasähkömoottorin pyörimisnopeuteen. Ankkurijänniteohjausmenetelmän haittapuolena ovat suuret tehohäviöt, jotka aiheutuvat säätövastuksesta tasasähkömoottorin ankkuriin kytkemisestä. (Agarwal 2014; Daware 2016.)



KUVIO 4. Ankkurijännitteen ohjausmenetelmä (mukaillen Daware 2016)

4.3 DC-moottorin ohjaus pulssinleveysmodulaation avulla

Kolmas tasasähkömoottorin ohjaustapa on pulssinleveysmodulaatio (KUVIO 5). Pulssinleveysmodulaatio tulee englanninkielisistä sanoista Pulse-Width Modulation, ja siitä käytetään lyhennystä PWM. Pulssinleveysmodulaatiolla muutetaan pulssin leveyttä ja siten laitteelle annettavaa voimaa. Mikrokontrollereilla, kuten Arduinolla, pystytään hyödyntämään pulssinleveysmodulaation tekniikkaa. Pulssinleveysmodulaatiota käytetään tasasähkömoottoreiden nopeuden ohjaamisessa erityisen paljon. Pulssinleveysmodulaation avulla saadaan pidettyä tehohäviöt pieninä. (Agarwal 2014; Hirzel; Lin 2018.)



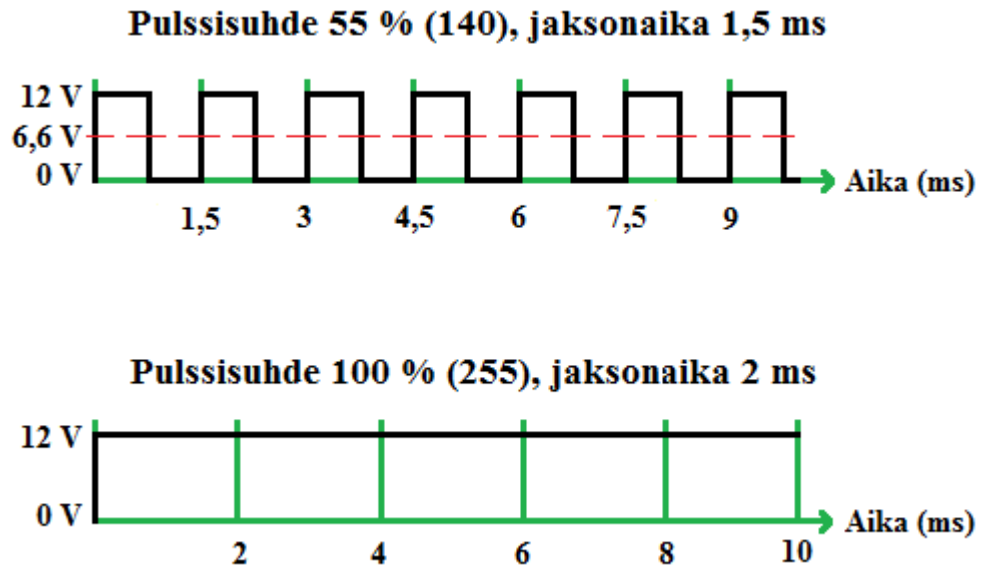
Jakso = Time ON + Time OFF

Taajuus = 1/Jakso

Pulssisuhde = (Time ON/Jakso) * 100%

KUVIO 5. Pulssinleveysmodulaation periaate (mukaillen EBLDC.com 2018)

Pulssisuhde tarkoittaa sitä, kuinka kauan pulssisignaali on ylätilassa suhteessa yhden kokonaisen jakson pituuteen nähden (KUVIO 6). Pulssisuhde ilmoitetaan yleensä prosentteina. Taajuudella ilmaistaan, kuinka nopeasti pulssin suoritus tapahtuu. Maksimipulssisuhde on 100 %, ja minimipulssisuhde on 0 %. Tämä tarkoittaa sitä, että maksimipulssisuhteella kohde, esimerkiksi moottori, saa jatkuvasti täydet jännitteet käyttöönsä ja minimiprosentilla ei jännitettä lainkaan. Jos taas pulssisuhde on 50 %, moottorin saama jännite on puolet maksimista. Pulssisuhde antaa siis ylä- ja alatilien välisen keskiarvon jännitettä kohteelle. Mikrokontrollerin digitaaliohjauksilla tuotetaan neliöaaltosignaaleja, joilla simuloitetaan jännitettä. Radio-ohjaimen lähetin ja vastaanotin käyttävät myös pulssinleveysmodulaatiota servojen asentojen ohjaamiseen. Radio-ohjaimissa yleensä käytetään 1 ms:n ja 2 ms:n välistä pulssin jaksonaikaa. (Avayan 2018; Hirzel; Lin 2018.)



KUVIO 6. Pulssisuhteiden synkronointi Arduinin 0–255-skaalausalueelle

5 MEKAANINEN SUUNNITTELU

Opinnäytetyön suunnitelmana oli tarkoitus hankkia akkukäyttöinen ruohonleikkuri, jolle tehtäisiin muunnosprosessi työnnettävästä mallista kauko-ohjattavaksi. Opinnäytetyötä aloitettaessa vastaan tuli sopivaksi vaihtoehdoksi FXA:n akkukäyttöinen ruohonleikkuri (KUVA 3, TAULUKKO 2). FXA:n ruohonleikkuri hankittiin K-Raudasta, ja leikkurin hinnaksi tuli 299 €. Muita etuja hinnan lisäksi tässä ruohonleikkurissa olivat ruohonleikkurin päältä löytyvä tila, johon voidaan sijoittaa tarvittavat komponentit, kuten akut (KUVA 4), vetomoottorit sekä Arduino. Tämä ruohonleikkuri ei ollut itsevetävä, joten ruohonleikkurin vetäminen toteutettiin vääntömoottoreiden avulla, joilla saatiin ohjaus yksinkertaiseksi.



KUVA 3. FXA:n akkukäyttöinen ruohonleikkuri

TAULUKKO 1. FXA-ruohonleikkurin tekniset tiedot (mukaillen K-Rauta; Redfem 2017, 24)

Tiedot	Arvo
Jännite	40 V DC
Pakkauksen nettopaino	24,4 kg
Tuotteen pakkauksen mitat (syvyys, leveys, korkeus)	87.0 x 54.5 x 41.5 cm
Tyyppi	Työnnettävä
Käyttövalmiin tuotteen paino (kg)	24,4
Terän pituus (cm)	43
Leikkuuleveys (cm)	43
Rengaskoko	180/250
Keruusäkki (l)	50
Virtalähde	Akku
Joutokäyntinopeus	3100 min ⁻¹
Kuormittamaton käyttöaika	23 min (2,5 Ah akku)
Korkeuden säätö	6 asentoa (25-75 mm)
Moottorin tyyppi	Harjaton moottori



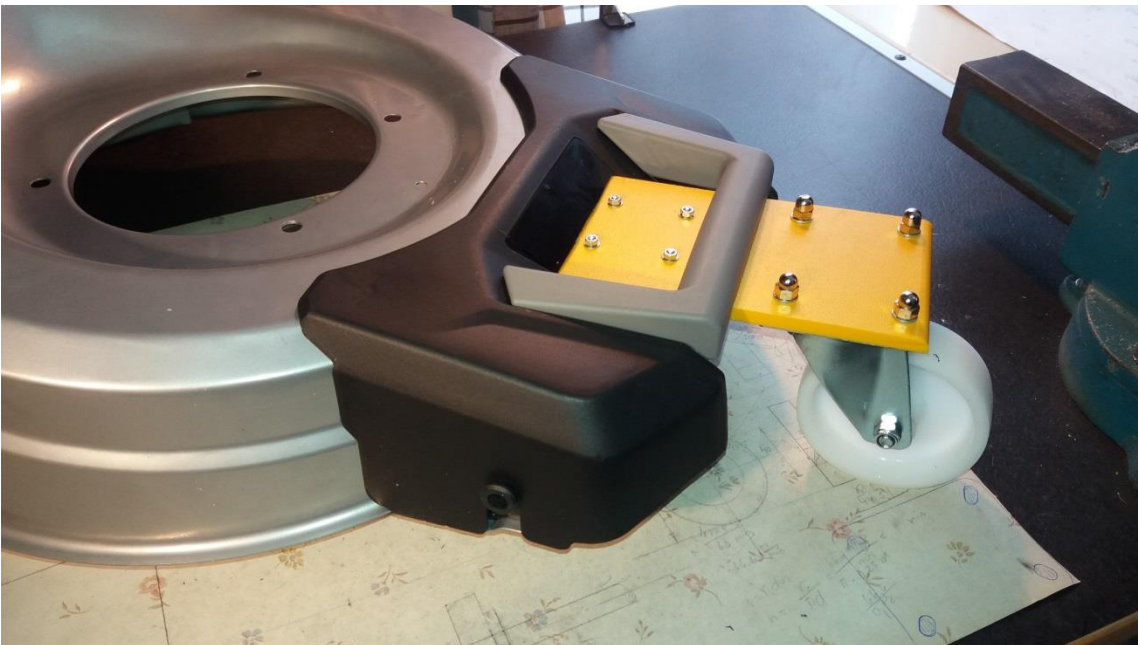
KUVA 4. Ruohonleikkurin 40 voltin akun/akuston akkutila

5.1 Alustavat toimenpiteet

Muutosprosessi ruohonleikkurille aloitettiin purkamalla pois työntöaisa, käynnistyskytkin, moottori, terä sekä takarattaat. Eturattaat, joiden halkaisija oli 180 mm, siirrettiin taakse (KUVA 5). Alkuperäiset takarattaat, joiden halkaisija oli 250 mm, poistettiin ja samalla ruohonleikkurin rungosta poistettiin korkeudensäätö. Ruohonleikkurin leikkuukorkeus saatiin näin pysymään vakiona 50 mm:ssä, ja vetosysteemi mahtui nyt leikkuriin paremmin ilman korkeudensäätöä. Eteen valittiin yksi muovinen vapaa ratas, joka oli halkaisijaltaan 100 mm ja leveydeltään 30 mm (KUVA 6). Vapaa ratas kiinnitettiin muovikappaleeseen, joka oli pituudeltaan 190 mm ja leveydeltään 105 mm. Muovikappale kiinnitettiin koneen runkoon kulmakiinnikkeellä.



KUVA 5. Eturattaat, jotka on siirretty taakse



KUVA 6. Vapaan pyörän kiinnitys ruohonleikkurin runkoon

5.2 Vääntömoottoreiden valinta

Ruohonleikkurin ohjaus päätettiin toteuttaa vääntömoottoreita hyödyntäen laittamalla molemmille tarkarattaille oma vääntömoottori. Vääntömoottorit kiinnitettiin ruohonleikkurin runkoon naulauslevyillä (KUVA 8). Naulauslevyt olivat kooltaan 80 mm x 240 mm x 2 mm. Vääntömoottoreiden mitoituksen laskeminen aloitettiin laskemalla tarvittava voima seuraavalla kaavalla

$$F = m * a \quad (1)$$

jossa m on massa (kg) ja a on kiihtyvyys (m/s^2). Voiman yksikkö on newton (N).

Ennen voiman laskemista piti selvittää yhtälössä olevat puuttuvat termit. Ruohonleikkurin massa pyöristettiin ylöspäin 25 kiloon. Lisäksi massaun lisättiin lisäakkujen paino, jolloin massaksi saatiin 30 kg. Nopeus arvioitiin 4 km/h:ssa ja sen saavuttamaksi ajaksi 3 sekuntia. Kiihtyvyys laskettiin seuraavalla kaavalla

$$a = \Delta v / \Delta t \quad (2)$$

jossa Δv on muutosnopeus (m/s) ja Δt on muutokseen kulunut aika (s). Kiihtyvyyden yksikkö on metriä neliösekunnissa (m/s^2).

Kiihtyvyyden laskenta:

$$a = \Delta v / \Delta t = 4000 \text{ m} / (3600 \text{ s} * 3 \text{ s}) = \underline{\underline{0,37 \text{ m/s}^2}}$$

Voiman laskenta:

$$F = m * a = 30 \text{ kg} * 0,37 \text{ m/s}^2 = \underline{\underline{11,1 \text{ N}}}$$

Seuraavaksi ratkaistiin tarvittava momentti seuraavalla kaavalla

$$M = F * r \quad (3)$$

jossa r on pyörien säde (m) ja F on voima (N). Momentin yksikkö newtonmetri (Nm).

Momentin laskenta:

$$M = F * r = 11,1 \text{ N} * 0,09 \text{ m} = 0,999 \text{ Nm} \approx \underline{\underline{1 \text{ Nm}}}$$

Seuraavaksi ratkaistiin pyörien pyörimisnopeus seuraavalla kaavalla

$$n = v / \pi d \quad (4)$$

jossa v on nopeus (m/s), π on matemaattinen vakio likiarvoltaan kahden desimaalin tarkkuudella 3,14 ja d on pyörien halkaisija (m). Pyörimisnopeuden yksikkö on kierrosta sekunnissa (1/s).

Pyörimisnopeuden laskenta:

$n = v / \pi d = 4000 \text{ m} / 3600 \text{ s} * \pi * 0,180 \text{ m} = 1,96 \text{ 1/s} * 60 \text{ s} = \underline{118 \text{ 1/min}}$, valitaan 100 1/min, jolloin nopeudeksi saadaan 3,4 km/h, joka todettiin riittäväksi nopeudeksi.

Laskentojen perusteella sopiva moottorikandidaatti löytyi Elfa Distreleciltä, Micromotorsin valmistama vaihteistolla varustettu DC-moottori E192.12.18 (KUVA 7, TAULUKKO 3). Moottoreita ostettiin 2 kpl yhteishintaan 189,4 €.



KUVA 7. E192.12.18-DC-moottorit

TAULUKKO 2. E192.12.18-DC-moottorin tekniset tiedot (mukaan Elfa Distrelec)

Nimellisjännite	12 VDC
Nimellisvirta	1650 mA
Pyörimisnopeus	155...218 min ⁻¹
Nimellismomentti	0,6 Nm
Akselin ø	8 mm
Mitat ø x P	40.5 x 93 mm
Alennussuhde	18:1
Toimintalämpötila	-20...+60 °C

(jatkuu)

TAULUKKO 2. (jatkuu)

Liitäntälaji	Juotoskorvat
Tuoteperheen tiedot	Akselissa kaksinkertainen kuulalaakerointi Asennettavissa kaikkiin asentoihin Pyörimissuunnan vaihto napaisuutta muuttamalla Integroitu EMC-suoja

Lopuksi laskettiin moottorin ja pyörän välitys seuraavalla kaavalla

$$i = n_1 / n_2 \quad (5)$$

jossa n_1 on moottorin pyörimisnopeus (1/min) ja n_2 on pyörän pyörimisnopeus (1/min). Välytyssuhteella kuvataan pyörimisnopeuksien suhdetta.

Välytyssuhteen laskenta:

Moottorin keskimääräiseksi pyörimisnopeudeksi arvioitiin 180 1/min.

$$i = n_1 / n_2 = 180 \text{ 1/min} / 100 \text{ 1/min} = \underline{1,8}$$

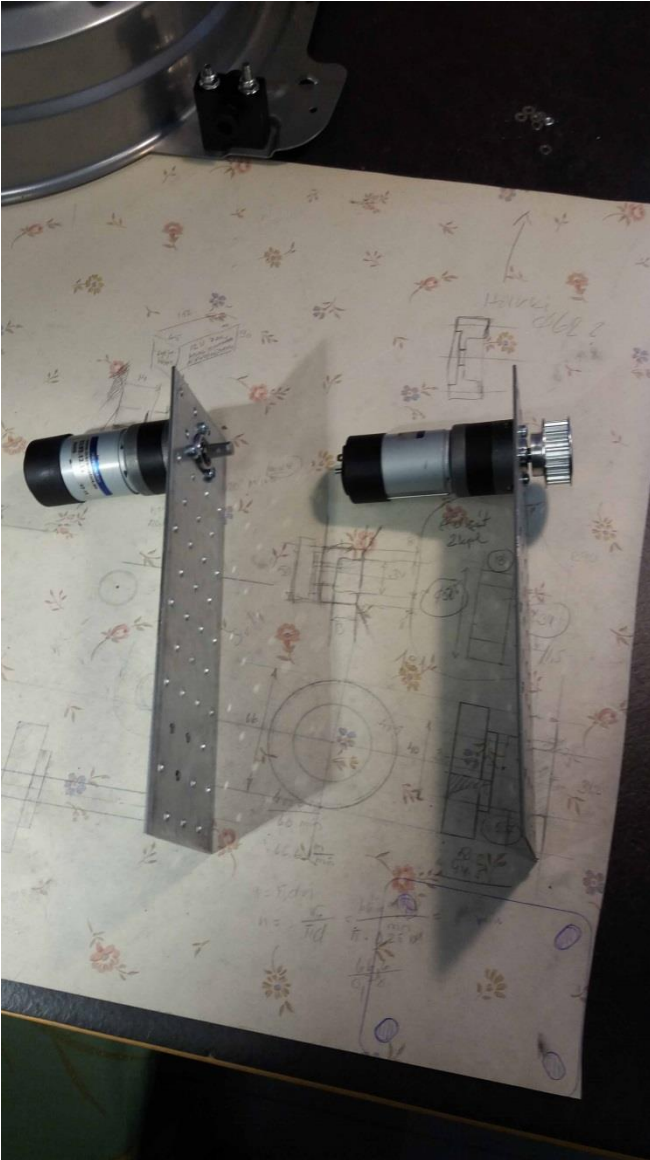
Tämän jälkeen laskettiin pyörän vääntömomentti välytyssuhteen avulla seuraavalla kaavalla

$$M_2 = i * M_1 \quad (6)$$

jossa M_1 on käytävän hammaspyörän vääntömomentti (Nm) ja i on välytyssuhde. M_2 on käytetyn pyörän vääntömomentti (Nm).

Vääntömomentin laskenta:

$$M_2 = i * M_1 = 1,8 * 0,6 \text{ Nm} = \underline{1,08 \text{ Nm}}$$



KUVA 8. Vääntömoottorien kiinnitys kiinnitysnaulauslevyyn

5.3 Hihnan mitoitus ja hammashihnapyörien valinta

Vääntömoottorit ja takapyörät yhdistettiin hammashihnoilla ja vetopyörillä. Isot vetopyörät kiinnitettiin ruohonleikkurin takapyörien akseliin. Pienet vetopyörät kiinnitettiin vääntömoottoreiden akseleihin. Sorvatut välilaidat laitettiin isojen vetopyörien ja ruohonleikkurin pyörien väliin (KUVA 10). Vetopyörien voimansiirto toteutettiin hammashihnoilla. Ensiksi lähdettiin etsimään sopivia hihnapyöriä. SKS Megapowerilta löytyi polyuretaanihihnaikäytöt pdf-dokumenttihihnojen mitoitukseen ja hammashihnapyörien valintaan. Vääntömoottoreiden mitoituksessa saatiin välityssuhteeksi 1,8. Tällä tu-

loksella tarkistettiin vetopyörien valitsemisen yhteydessä, että valitut vetopyörät ovat sopivia välityksen suhteen. Hihnan riittäväksi leveydeksi arvioitiin 10 mm.

Hammashihnapyörät valittiin SKS Megapowerin moduulista T5 esiporatut hammashihnapyörät. Näiden hammashihnapyörien materiaali oli alumiinia. Pienemmäksi pyöräksi valittiin hammashihnapyörä merkinnällä 21T5/22-2. Merkintä tarkoittaa sitä, että pyörän kokonaisleveys on 21 mm, jako on 5 mm, hammasluku on 22 ja laippamäärä on 2 (SKS Mekaniikka Oy b, 76). Vastaavasti isommaksi hammashihnapyöräksi valittiin hammashihnapyörä merkinnällä 21T5/40-2. Arvot tässä pyörässä ovat muuten samat, paitsi että hammasluku on 40. Tämän jälkeen tehtiin tarkistuslaskelma välityssuhteelle seuraavalla kaavalla

$$i = D_2 / D_1 \quad (7)$$

jossa D_1 on käytettävän hihnapyörän jakohalkaisija (mm) ja D_2 on käytettävän hihnapyörän jakohalkaisija (mm). i on välityssuhde.

Tarkistuslaskelma välityssuhteelle:

$i = D_2 / D_1 = 62,85 \text{ mm} / 34,25 \text{ mm} = \underline{1,84}$. Välityssuhde näillä pyörillä oli sopiva. Molempia hammashihnapyöriä tilattiin 2 kpl Tauno Tasanto Oy:ltä (KUVA 9). Molempien pyörien sisäreikää piti suurentaa, koska ne olivat liian pienet akseleita varten. Pienemmän pyörän sisäreiän alkuperäinen halkaisija oli 6 mm ja vääntömoottorin akseli oli 8 mm, joten reikä porattiin suuremmaksi. Vastaavasti isommat pyörät, jotka olivat alun perin 8 mm, porattiin isommiksi. Näin ne sopivat takapyörien akselille, jonka halkaisija oli 12 mm.



KUVA 9. Kuvassa ovat sorvatut välilaidat ja vetopyörät



KUVA 10. Vetopyörä, välilaiippa ja ruohonleikkurin pyörä

Hihnan mitoitus aloitettiin selvittämällä ensiksi akselinväli. Akselinvälin selvittämiseen käytettiin kah- ta akselinvälisuosituskaavaa sekä SKS Gates -hammashihnakäytön suunnitteluopasta. Ensimmäisenä akselinvälisuosituskaavana oli seuraava kaava

$$c \geq 2 \dots 3 * (D_1 + D_2) \quad (8)$$

jossa D_1 on käytettävän hihnapyörän jakohalkaisija (mm) ja D_2 on käytettävän hihnapyörän jakohalkaisi- ja (mm). Akselinvälin yksikkö on millimetri (mm).

Akselinvälin laskenta ensimmäisellä kaavalla:

$$c \geq 2 \dots 3 * (D_1 + D_2) = 2 \dots 3 * (34,25 \text{ mm} + 62,85 \text{ mm}) = \underline{194,2 \text{ mm} \dots 291,3 \text{ mm}}$$

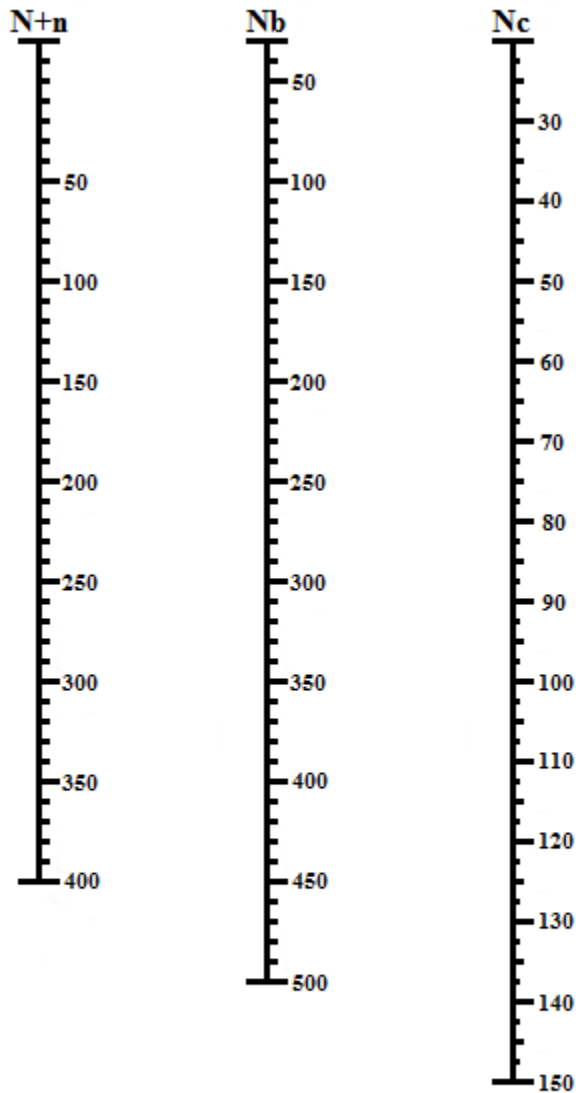
Toisena akselinvälisuosituskaavana oli seuraava kaava

$$c \geq 4 * D_2 \quad (9)$$

jossa D_2 on käytettävän hihnapyörän jakohalkaisija (mm). Akselinvälin yksikkö on millimetri (mm).

Akselinvälin laskenta toisella kaavalla:

$c \geq 4 * D_2 = 4 * 62,85 \text{ mm} = \underline{251,4 \text{ mm}}$. Tällä arvolla tarkistettiin ja todettiin, että akselinväli ei tule sopimaan laitteeseen, joten seuraavaksi päätettiin käyttää SKS Gates -hammashihnakäytön suunnitte- luopasta apuna, jotta saataisiin tarkempi tulos. Suunnitteluoppaasta löytyi suunnittelunomogrammi (KUVIO 7), jonka avulla pystyi arvioimaan akselinvälin, jos tiesi isompien ja pienempien vetopyörien hammaslukujen summan. Aikaisemmin valittujen vetopyörien hammaslukujen summa oli 62, joten suunnittelunomogrammin akselinväliksi jakopituuksina saataisiin tässä tapauksessa 40 mm:ä. Jako ilmoitettiin 5 mm pyörien kohdalla. Se tarkoittaa, että varsinainen akselinväli on viisinkertainen, jos se ilmoitetaan jakopituuksina. Lopullinen akselinväli oli siis 200 mm.



N+n = Ison ja pienen pyörän hammaslukujen summa

Nb = Hihnan hammasluku

Nc = Akseliväli jakopituuksina

KUVIO 7. Nomogrammi (mukaillen SKS Mekaniikka Oy a, 23)

Seuraavaksi lähdettiin selvittämään kosketuskulma, jonka avulla pystytään ratkaisemaan hihnan pituus seuraavalla kaavalla

$$\beta = 2 * \arccos[(t * (z_2 - z_1)) / (2 * \pi * c)] \quad (10)$$

jossa t on hammasjako (mm), z_1 on käyttävän hihnapyörän hammasluku, z_2 on käytettävän hihnapyörän hammasluku, c on akseliväli (mm) ja π on matemaattinen vakio likiarvoltaan kahden desimaalin tarkkuudella 3,14. Kosketuskulma ilmaistaan tasokulmana, jonka tunnus on β .

Kosketuskulman laskenta:

$$\beta = 2 * \arccos[(t * (z_2 - z_1)) / (2 * \pi * c)] = 2 * \arccos[(5 * (40 - 22)) / (2 * \pi * 200 \text{ mm})] = \underline{171,79^\circ}$$

Lopuksi selvitettiin hihnan pituus seuraavalla kaavalla

$$L = 2 * c * \sin(\beta / 2) + t / 2 * [z_1 + z_2 + (1 - \beta / 180^\circ) * z_2 - z_1] \quad (11)$$

jossa c on akseliväli (mm), β on kosketuskulma (asteina), t on hammasjako (mm), z_1 on käyttävän hihnapyörän hammasluku ja z_2 on käytettävän hihnapyörän hammasluku. Hihnan pituuden yksikkö on millimetri (mm).

Hihan pituuden laskenta:

$$L = 2 * c * \sin(\beta / 2) + t / 2 * [z_1 + z_2 + (1 - \beta / 180^\circ) * z_2 - z_1]$$

$$= 2 * 200 \text{ mm} * \sin(171,79^\circ / 2) + 5 \text{ mm} / 2 * [22 + 40 + (1 - 171,79^\circ / 180^\circ) * 40 - 22] = \underline{556,03 \text{ mm.}}$$

SKS Megapowerin oppaan taulukon mukaan lähimmäksi hihnan pituudeksi sopisi lasketun tuloksen perusteella 560 mm pituinen hihna. Valinnassa päädyttiin kuitenkin 575 mm pituiseen hihnaan, koska hihnanpituuden on hyvä olla hiukan laskettua arvoa pidempi, sillä akseliväli voi heittää asennuksen yhteydessä puoleen tai toiseen. Ylimääräinen pituus pystyttiin kompensoimaan hihnan kiristimellä. Tällä pituudella hihnan hammasluvuksi saadaan 115, kun hihan pituus jaetaan 5 mm:n hammasjakoilla. SKS Gatesin suunnittelunomogrammin hihnan hammasluku osuisi myös suurin piirtein 115:een, kun hammaslukujen summa on 62 ja akseliväli 40 mm jakopituuksina. Hihnoja tilattiin 2 kpl Tauno Tasanto Oy:stä (KUVA 11). Hihnoiniin laitettiin vipuvarrella varustettu jousikuormitteinen kiristysrulla poly muovista sorvaamalla (KUVA 12).



KUVA 11. Hihnat ja hammashihnapyörät



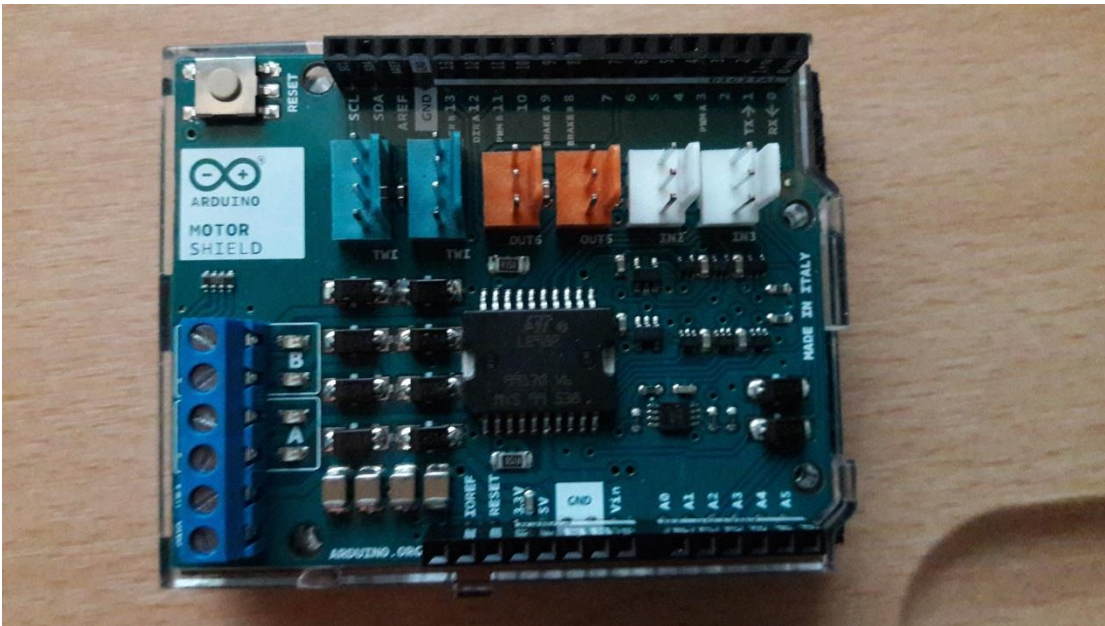
KUVA 12. Hihnojen kiristys

6 OHJAUSSUUNNITTELU

Ohjaussuunnittelussa mietittiin ja käytiin läpi erilaisia vaihtoehtoja siitä, miten ohjaus toteutettaisiin. Mekaaninen toteutus servoilla oli yksi varteenotettava ratkaisu. Nopeudensäädintä oli myös ajateltu. Turvallisuuspuolta varten projektiin oli hankittu Arduino-mikro-ohjain. Valitsemalla mekaanisen ratkaisun olisi Arduinoa varten pitänyt tässä projektissa syventyä elektroniikkapuoleen saattamalla jännite sopivaksi jänniteregulaattorin avulla. Erilaisia moottorinohjaimia läpikäydessä vastaan tuli Arduinoa varten tehty Arduinon oma moottorinohjain. Koska Arduinoa käytettiin jo turvallisuuspuolen suunnittelussa, valitut vääntömoottorit olivat käyttöjännitteeltään 12 V. Arduinon moottorinohjaimen toimintajännite oli puolestaan 5–12 V. Näiden tekijöiden vuoksi päätettiin ottaa kaikki hyöty irti Arduinosta toteuttamalla ohjaus kokonaan sillä.

6.1 Uusien laitteiden hankinta

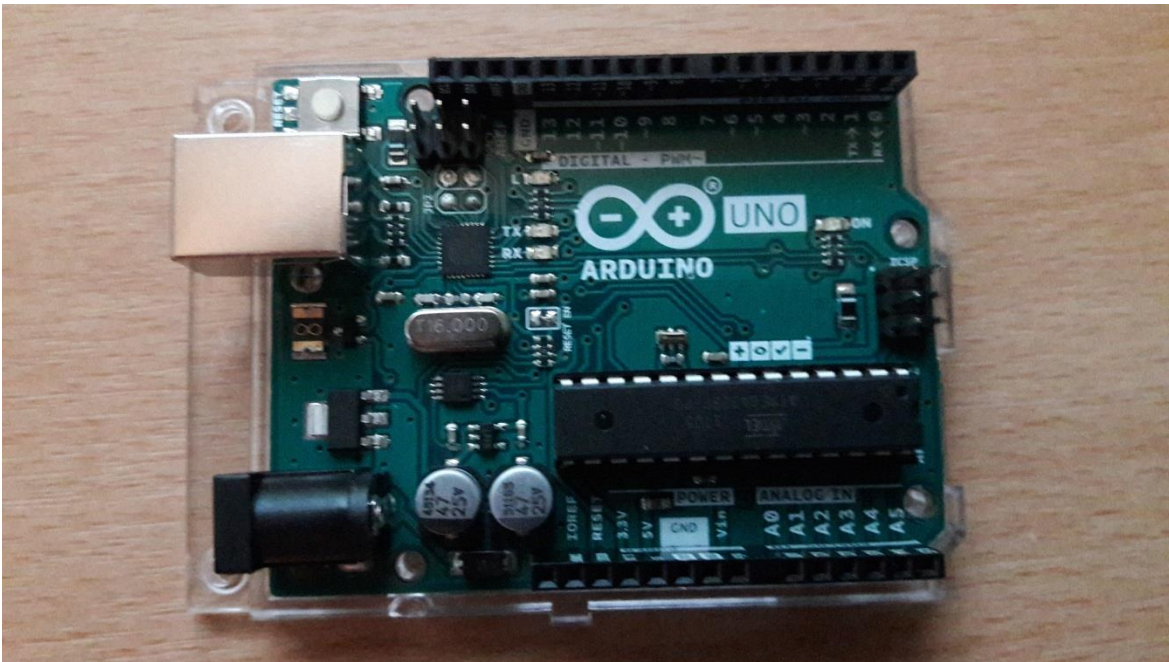
Projektiin oli ennestään hankittuna Arduinon Starter Kitin mukana tullut vanhempi versio Arduino Unon kehitysalustasta sekä muita elektroniikkakomponentteja, laitteita ja tarvikkeita. Vääntömoottoreita pystyi ohjaamaan Arduinolla Arduinon oman moottorinohjainyksikön kautta. Arduinon moottorinohjainyksikkö Arduino Motor Shield (KUVA 13, TAULUKKO 4) ostettiin Partcon verkkokaupasta hintaan 38,90 €. Samalla ostettiin uudempi versio Arduinosta, joka oli Arduino Uno Rev3 (KUVA 14, TAULUKKO 5) hintaan 29 €, koska Arduinon moottorinohjainyksikkö oli suoraan yhteen liitettävä tämän Arduinon version kanssa. Vanhempikin versio olisi toiminut, mutta silloin olisi pitänyt tehdä pieni muutosprosessi Arduinoon, koska vanhemmasta versiosta puuttui muutama pinni.



KUVA 13. Arduino Motor Shield

TAULUKKO 3. Arduino Motor Shield tekniset tiedot (mukaillen Arduino b)

Käyttöjännite	5 V–12V
Moottorinohjain	L298P, ohjaa kahta DC-moottoria tai yhtä askelmoottoria
Maksimivirta	2 A kanavaa kohden, 4 A maksimissaan (erillinen virtalähde)
Virranmittaus	1,65 V/A
Vapaa pysäytys ja jarrutusfunktiot	



KUVA 14. Arduino Uno Rev 3

TAULUKKO 4. Arduino Uno Rev 3 tekniset tiedot (mukaan Arduino b)

Mikrokontrolleri	ATmega328P
Käyttöjännite	5 V
Tulojännite (suositus)	7–12 V
Tulojännite (raja)	6–20 V
Digitaalinen I/O -pinnit	14 (joista kuudessa on PWM-ulostulo)
PWM Digitaalinen I/O -pinnit	6
Analogiset sisääntulopinnot	6
DC-virta I/O -pinnejä kohden	20 mA
DC-virta 3.3 V -pinneille	50 mA
Flash-muisti	32 KB (ATmega328P), josta 0.5 KB käyttää alku- latausohjelmaa
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Kellotaajuus	16 MHz

(jatkuu)

TAULUKKO 4. (jatkuu)

LED_BUILTIN	13
Pituus	68.6 mm
Leveys	53.4 mm
Paino	25 g

Projektissa oli myös ennestään Futaban T2ER-kaksikanavainen radio-ohjain varustettuna Futaban R152JE-kaksikanavaisella vastaanottimella, jonka taajuus oli 27 MHz. Kuitenkin alusta asti oli selvää, että kanavia tarvittaisiin vähintään kolme. Yhtä kanavaa käytettäisiin ruohonleikkurin ohjaamiseen eteenpäin ja taaksepäin, toista kanavaa käytettäisiin ruohonleikkurin kääntämiseen ja kolmatta kanavaa ruohonleikkurin käynnistämiseen ja sammuttamiseen. Arduinon ohjelmoinnin pystyi aloittamaan hyvin projektissa ennestään mukana olleella radio-ohjaimella, mutta ohjelmoinnin edetessä hankittiin uusi Futaban T4YWD-nelikanavainen radio-ohjain varustettuna Futaban R214GF-E-nelikanavaisella vastaanottimella (KUVA 15, TAULUKKO 6), jonka taajuus oli 2,4 GHz. Radio-ohjain ostettiin eurorc.comista hintaan 139,90 €.



KUVA 15. Futaban T4YWD-nelikanavainen radio-ohjain ja R213GF-E-nelikanavainen vastaanotin

TAULUKKO 5. Futaban lähettimen ja vastaanottimen tekniset tiedot (mukaillen Futaba 2013, 20)

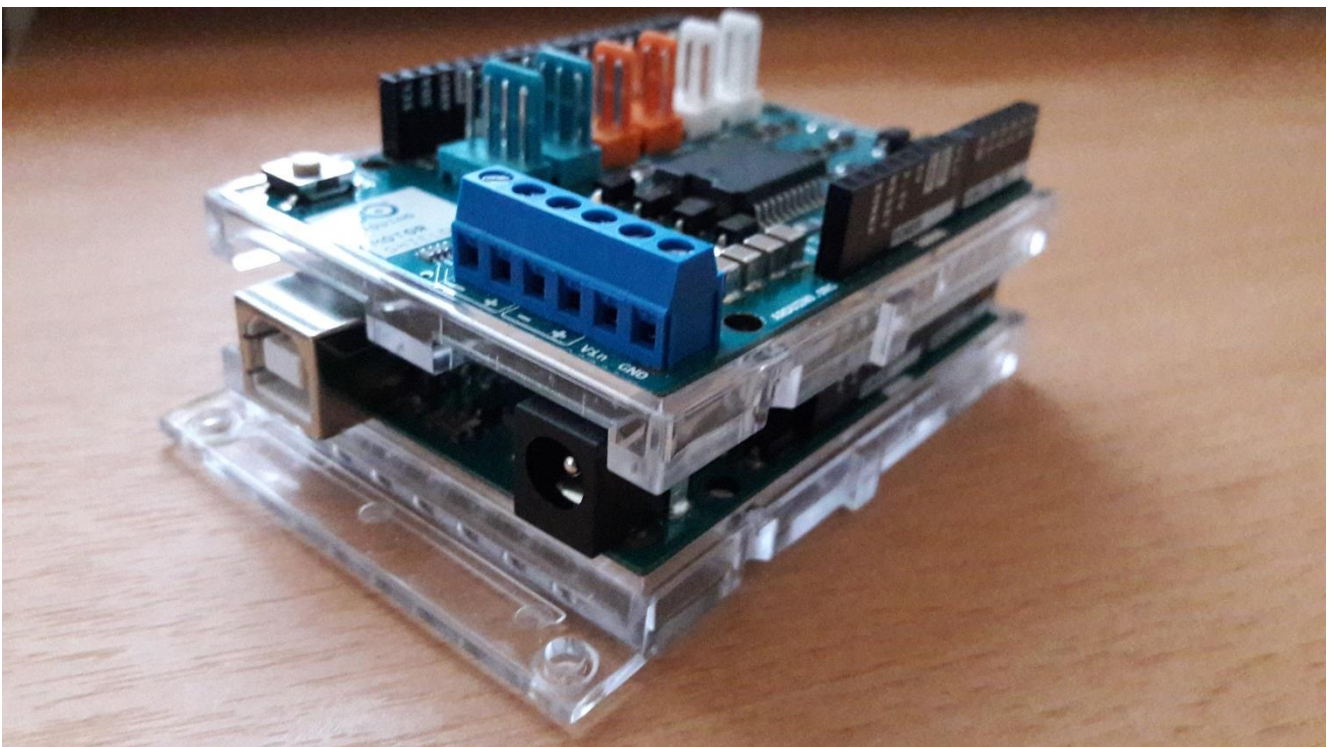
Lähettimen taajuus	2,4 GHz
Lähettimen tehovaatimus	Kuivaparisto x 4 (6 V)
Lähettimen virran tyhjentyminen	100 mA
Lähettimen antenni	$1/2 \lambda$
Vastaanottimen käyttöjännite	4,8 V - 7,8 V
Vastaanottimen paino	6 g

6.2 Arduinon moottorinohjainyksikön, vastaanottimen sekä vääntömoottoreiden kytkeminen Arduinoon

Arduinon moottorinohjainyksikkö painettiin Arduinoon kiinni moottorinohjainyksikössä olevien piikki-
rimojen kautta (KUVA 17). Moottorinohjainyksikön 5 voltin jännite tuotiin Futaban vastaanottimen
neljännen kanavan/patterikanavan keskimmäiseen plus-nastaan, ja moottorinohjainyksikön GND tuotiin
vastaanottimen oikeanreunimmaiseen miinus-nastaan. Vastaanottimen ensimmäistä kanavaa käytettiin
ruohonleikkurin ohjaamiseen eteen-/taaksepäin. Sen kanavan vasemmanreunimmaisen PWM-signaalin
nasta kytkettiin moottorinohjainyksikön pinniin 6. Vastaanottimen toista kanavaa käytettiin ruohon-
leikkurin kääntämiseen. Se kytkettiin moottorinohjainyksikön pinniin 7. Vastaanottimen kolmas kanava
käytettiin ruohonleikkurin käynnistämiseen. Se kytkettiin moottorinohjainyksikön pinniin 2. Vääntö-
moottorit ja 12 voltin akku kytkettiin moottorinohjainyksikön päädyssä oleviin liittimiin. Varsinaises-
sa käytössä Arduino sai käyttöjännitteensä moottorinohjainyksikön kautta samasta akusta kuin vääntö-
moottorit. Akkuna käytettiin Powercellin 12 voltin ladattavaa lyijyakkua (KUVA 16).



KUVA 16. Powercellin 12 voltin ladattava lyijyakku



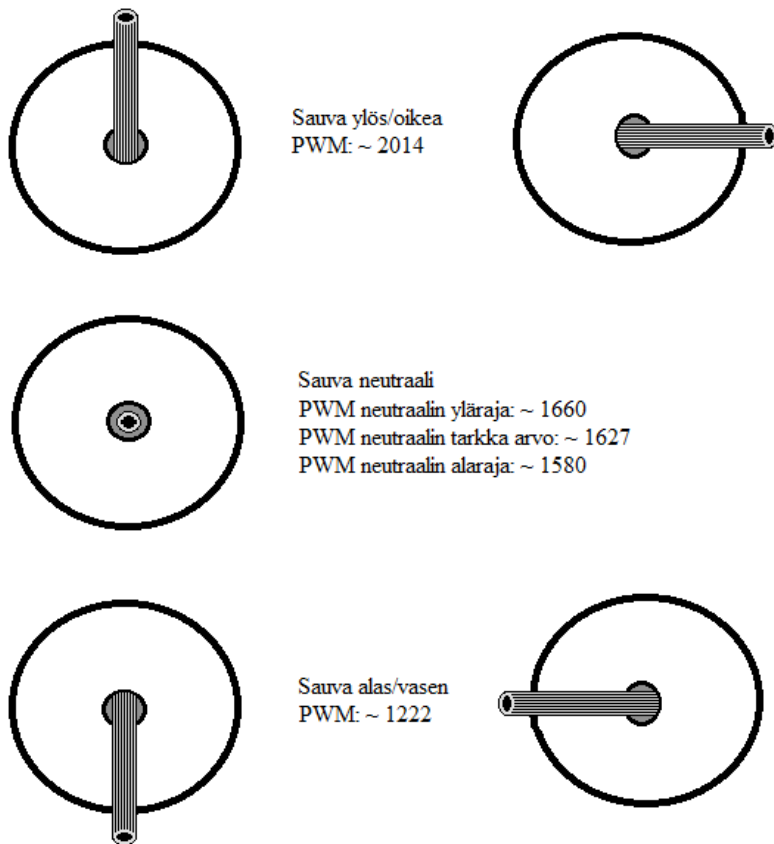
KUVA 17. Kuvassa Arduino Uno Rev 3 ja Arduino Motor Shield ovat liitettyinä toisiinsa

6.3 PWM-signaalin määrittely Arduinolla

Ohjauksen ydinideana oli ohjata PWM-signaalin pulssiarvojen perusteella ruohonleikkuria. Ensiksi piti määrittellä mitä arvoja Arduino lukee vastaanottimesta `pulseIn()`-funktion avulla. Kaksikanavaisella radio-ohjaimella ilmeni sellainen ongelma, että vastaanottimen puoli pysyi päällä, vaikka radio-ohjain sammutettiin tai vietiin kantaman ulkopuolelle. Lisäksi Arduinon lukemat arvot rupesivat heittelemään suuresti tässä tilanteessa, ja se olisi aiheuttanut varsinaisessa ohjauksessa sen, että vääntömoottorit olisivat ruvenneet pyörimään omin päin kantaman kadotessa. Uudemmassa nelikanavaisessa radio-ohjaimessa tätä ongelmaa ei ilmennyt, koska vastaanottimen puoli sulki itse itsensä, jos radio-ohjain sammutettiin tai kantama kadotettiin. Kantama oli myös uudemmalla ohjaimella huomattavasti suurempi, kun vastaanottimen taajuus oli 2,4 GHz 27 MHz sijaan.

Radio-ohjaimen kanavien trimmer-nimiset säädinpainikkeet asetettiin yläasentoon kahdesta syystä. Tällä tavalla tehtäessä ei tarvinnut itse arvioida sitä, ovatko jokaisen kanavan säätimet juuri samassa kohtaa. Pienikin heitto säätimen asennossa sotkisi ajatellun logiikan ohjelmoinnista. Toiseksi, jos käyttäjä vahingossa muuttaa säätimen asentoa, se on helppo ja tarkempi siirtää takaisin päätyasentoon kuin esimerkiksi puoleenväliin. Radio-ohjaimen servo reversing switch -nimisestä kääntökytkinmoduulista piti kääntää toisen ja kolmannen kanavan kytkimet rev-asentoon ja jättää ensimmäisen ja neljännen kanavan kytkimet nor-asentoon, jotta PWM-pulssiarvot kasvavat ja pienenevät kaikilla kanavilla samansuuntaisesti. Arduinon `pulseIn()`-funktiolla selvitettiin kummankin PWM:n maksimipulssiarvoksi noin 2014 sauvan ollessa ylhäällä tai oikealla (KUVIO 8). Sauvojen ollessa alhaalla tai vasemmalla saatiin vastaavaksi arvoksi noin 1222. Sauvojen ollessa koskemattomina pulssiarvoksi saatiin noin 1627.

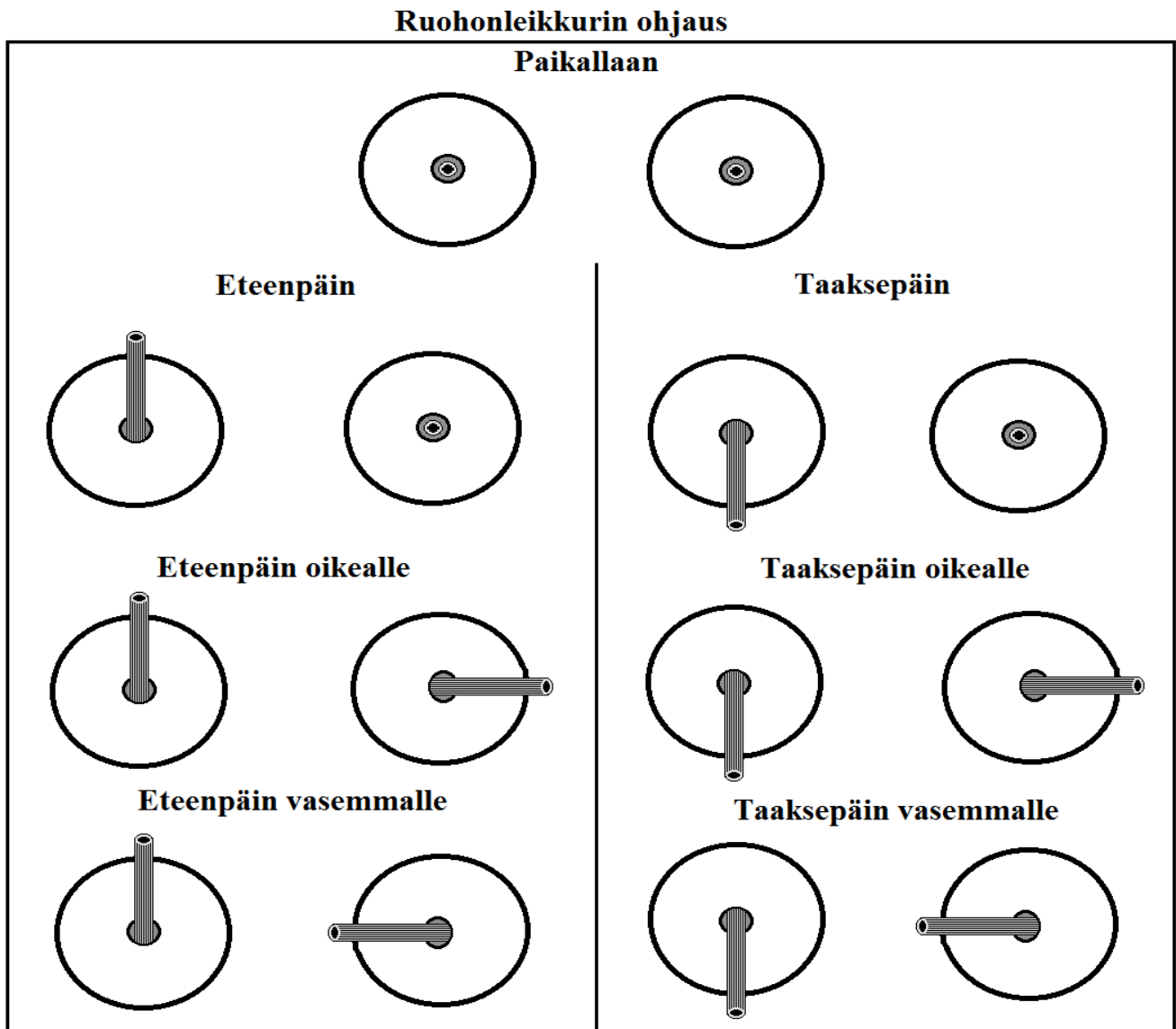
Sauvojen asentojen PWM-arvot



KUVIO 8. Sauvojen asentojen PWM-arvot

6.4 Ohjausohjelman teko

Ensi alkuun ladattiin Adafruitin Internet-osoitteesta Adafruitin tekemä AF_Motor-niminen moottoriohjainyksikköön tarkoitettu kirjasto, joka tallennettiin Arduinon libraries-kansioon. Varsinaisessa ohjelmassa tehtiin seitsemän eri radio-ohjaimen ohjaussauvojen asentojen määrittelyä (KUVIO 9), minkä perusteella ruohonleikkuria ohjattiin halutulla tavalla. Ensimmäisenä määriteltiin ruohonleikkurin pysyminen paikallaan. Ruohonleikkurin pysyminen paikallaan vaati, ettei käyttäjästä katsottuna vasempana olevaa ohjaussauvaa liikutettaisi pystysuuntaisesti ja oikeaa ohjaussauvaa vaakasuuntaisesti. Kääntämällä vasemman sauvan joko ylös tai alas ruohonleikkuri määriteltiin kulkemaan joko eteenpäin tai taaksepäin ja kääntämällä oikean sauvan joko oikealle tai vasemmalle määriteltiin kääntymisen joko oikealle tai vasemmalle.



KUVIO 9. Ruohonleikkurin ohjaus, radio-ohjaimen sauvojen asennot

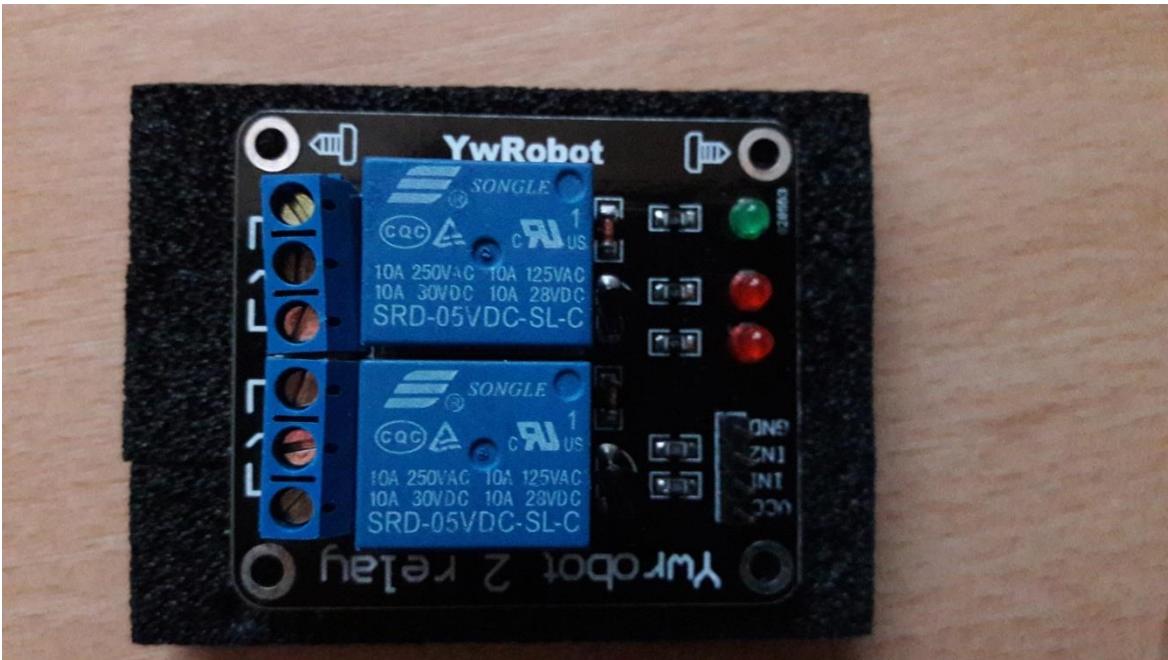
Ruohonleikkurin ollessa paikallaan PWM-arvot määriteltiin välille 1580–1660, jotta Arduino tulkitsisi, ettei vääntömoottoreita pyöritettäisi. Vääntömoottoreiden pyöriminen eteenpäin laitettiin alkamaan, jos PWM-arvo ylittää 1660. Vastaavasti vääntömoottoreiden pyöriminen taaksepäin laitettiin alkamaan, jos PWM-arvo alittaa 1580. Lisäehto täytyi lisätä PWM-arvon alitustilanteeseen, koska Arduino tulkitsisi radio-ohjaimen ja vastaanottimen sammuttamisen tai kantaman menetyksen ruohonleikkurin peruuttamiseksi. Arduinon lukemat PWM-arvot vastaanottimen ollessa sammuneena olivat 0, joten lisäehdoksi lisättiin, ettei Arduino aseta ruohonleikkuria peruuttamaan, jos PWM-arvo on alle 100. Map()-funktion avulla pulssisuhteet synkronoitiin Arduinon 0–255-skaalausalueelle siten, että vääntömoottoreiden nopeuden lähtönopeudeksi asetettiin 140 ja loppunopeudeksi 255. Testivaiheessa huomattiin, että ohjauksesta tulee vikkelämpinen nopeuden alkaessa 140 kuin nolasta. Map()-funktio synkronoi

eteenpäin kulkemisessa PWM-arvon 1660 lähtönopeudeksi 140 ja PWM-arvon 2014 loppunopeudeksi 255. Vastaava synkronointi tapahtui myös toiseen suuntaa peruuttamisessa. Peruuttamisessa PWM-arvo 1580 asetettiin lähtönopeudeksi ja PWM-arvo 1222 loppunopeudeksi.

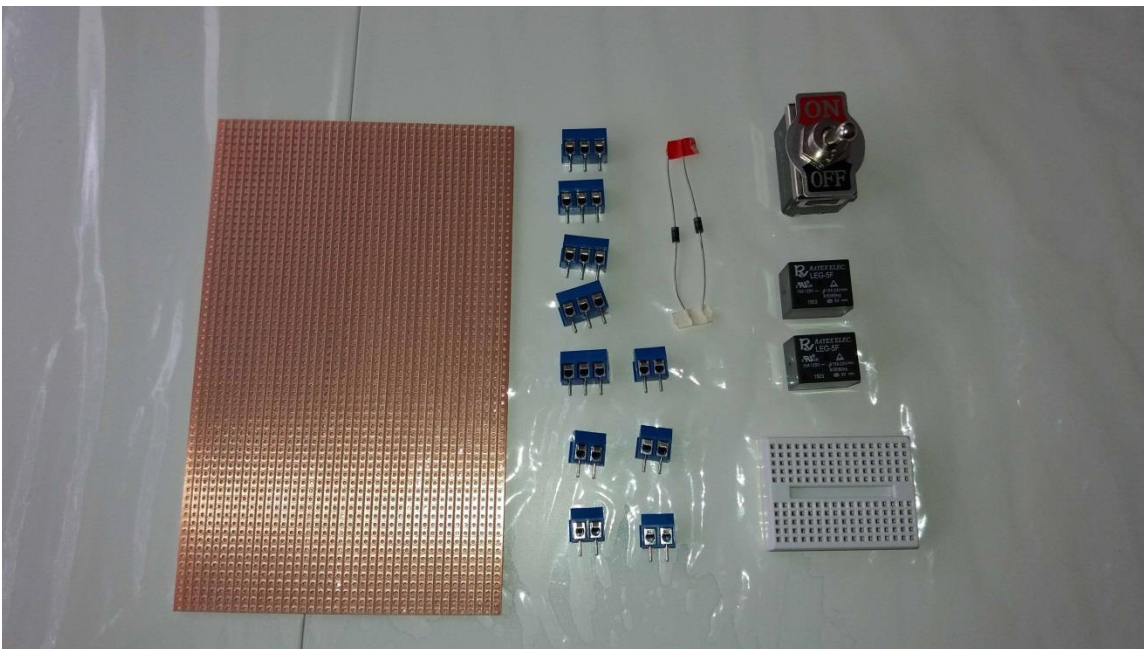
6.5 Ruohonleikkurin käynnistäminen

Ruohonleikkurin käynnistämässä mietittiin ensimmäisenä vaihtoehtona servojen avulla toteutettavaa mekaanista käynnistyskytkimen käynnistämistä. Lopulta päädyttiin kuitenkin releohjattuun käynnistämismvaihtoehtoon. Projektiin oli jo valmiiksi hankittu SRD05VDC-SL-C-niminen 5 voltin kelajännitteen omaava relekortti (KUVA 18), jolla pystyi ohjaamaan 10 ampeerin virtaa 30 voltin tasajännitteellä. Kaiken lisäksi tämä relekortti oli yhteensopiva Arduinon kanssa. Koska ruohonleikkurin akun jännite oli 40 V, ostettiin SP-Elektroniikasta toinenkin rele, jota käytettiin ohjauksessa. Releen nimi oli LEG-5F (KUVA 19), ja sillä pystyi ohjaamaan 15 ampeerin virtaa 24 voltin tasajännitteellä. Releeseen kolvattiin kiinni estosuuntaan 1N4007-diodi suojaamaan kelaa ylijännitepiikeiltä.

Relekortti kytkettiin kiinni Arduinon moottorinohjainyksikköön, jonka kautta se sai kelan tarvitseman 5 voltia. Relekortin piikkiriman IN1-piikki kytkettiin moottorinohjainyksikön pinniin 10. Arduino ohjelmoitiin kytkemään rele vetämään radio-ohjaimen kolmannen kanavan PWM-arvon mukaan. PWM-arvo määriteltiin siinä käynnistämään rele, kun PWM-arvo ylittää arvon 1990, ja sammuttamaan rele PWM-arvon alittaessa arvon 1270. PWM-arvojen rajat laitettiin tarkoituksella korkeiksi, jotta kolmas kanava, joka on radio-ohjaimessa oikea ohjaussauva pystysuuntaisesti, ei häiriintyisi, jos sauvaa käänetään vaakasuuntaisesti ja käyttäjän ohjaimelle antama sauvan liike ei ole millintarkkaa. Näin ruohonleikkurin käynnistämistä ja sammuttamista ei tapahdu tahattomasti inhimillisen virheen takia. Relekortilla ohjataan 5 voltin jännite LEG-5F-releelle ja tällä releellä ohjataan 40 voltin akun jännite ruohonleikkurille.



KUVA 18. SRD-05-VDC-SL-C-relekortti



KUVA 19. Asennustarvikkeita, LEG-5F-rele sekä diodi 1N4007

6.6 Turvallisuussuunnittelu

Ruohonleikkurin turvallisuussuunnittelussa oli jo huomioitu ohjelmoinnin yhteydessä kantaman kaatoaminen/ohjaimen sammuminen sekä se, ettei ohjain toimisi käyttäjällä huolimattomasti. Kuitenkin näiden lisäksi ajateltiin parantaa turvallisuutta lisäämällä Arduinon Starter Kitin mukana tullut ultraäänianturi HC-SR04 (KUVA 20, TAULUKKO 7) projektiin mukaan suojaamaan sitä, ettei käyttäjä pysty ohjaamaan ruohonleikkuria suoraan toista ihmistä tai muuta kohdetta päin. Ultraäänianturi kytkettiin Arduinon moottorinohjainyksikköön, josta se sai tarvitsemansa 5 voltia. Anturin TRIG-liitin kytkettiin moottorinohjainyksikön pinniin 4 ja anturin ECHO moottorinohjainyksikön pinniin 5.



KUVA 20. HC-SR03-ultraäänianturi

TAULUKKO 6. HC-SR04-ultraäänianturin tekniset tiedot (mukaien SparkFun Electronics)

Käyttöjännite	5 VDC
Käyttövirta	15 mA
Mittauskulma	15°
Mittausetäisyys	2 cm–4 m

PulseIn()-funktiolla laitettiin Arduino lukemaan kohteen ja anturin välinen aika. Sen jälkeen saadun ajan avulla laitettiin Arduino laskemaan matka oheisella muunnoskaavalla:

$$S = t / 2 / 29,1 \quad (12)$$

jossa t on aika. Matkan yksikkö on metri (m).

Muunnoskaava ilmoitti Arduinolle kohteen ja anturin välin senttimetreinä. Lisäehto lisättiin ruohonleikkurin käynnistämiseen, sammuttamiseen sekä vääntömoottoreiden pyörimiseen eteenpäin. Ehdoksi laitettiin, että 10 senttimetrin etäisyys estää ruohonleikkurin käynnistämisen, ja jos ruohonleikkuri on jo käynnissä, sammuttaa ultraäänianturi ruohonleikkurin 10 senttimetrin etäisyydellä olevan kohteen takia. Lisäehto esti myös ruohonleikkurin ohjaamisen suoraan kohteeseen ultraäänianturin suuntaisesti.



KUVA 21. Ruohonleikkurin testausajo

7 YHTEENVETO

Opinnäytetyön lopullisena tuloksena tuli valmis radio-ohjattava ruohonleikkurin prototyyppi. Opinnäytetyö oli erittäin haastava kokonaisuus, joka piti sisällään sekä mekaanista suunnittelua että ohjaussuunnittelua. Testausajot suoritettiin keväällä lumien sulettua (KUVA 21). Ruohonleikkurin ohjauspuoleen oltiin täysin tyytyväisiä. Ruohonleikkurin vauhti ja ketteryys olivat ideaaliset. Ohjausohjelma toimi kuten oli haluttu eli radio-ohjaimen kaksi kanavaa riittivät hyvin ruohonleikkurin ohjaamiseen eteen- ja taaksepäin sekä kääntämiseen sivuille. Kolmannella kanavalla saatiin ruohonleikkurin käynnistäminen toimimaan radio-ohjaimen kautta hyödyntämällä releitä kytkennässä. Ruohonleikkurista saatiin myös turvallinen siten, ettei ruohonleikkuri jatka matkaa itseksensä eikä ruohonleikkurin terä jatka leikkaamista radio-ohjaimen kantaman kadottua tai sammuttua. Leikkurin terä ei myöskään jatka leikkaamista eikä ruohonleikkuria pysty ohjaamaan suoraan esteeseen, jos ultraäänianturi, joka sijoitettiin ruohonleikkurin eteen, tunnistaa kohteen.

Mekaaniseen puoleen ruohonleikkurista oltiin melko tyytyväisiä. Tasaisella nurmikolla tai alustalla ruohonleikkurin vääntömoottorit ovat riittävän voimakkaat kääntääkseen ruohonleikkuria. Pitkällä nurmikolla tai muuten epätasaisessa maastossa ruohonleikkurin vääntömoottorit olivat liian heikot ruohonleikkurin kääntämiseen. Toinen huomattu ongelmia tuottanut asia oli edessä oleva vapaa pyörä, joka ei ollut kaikista ideaalein tämän tyyppisiin projekteihin. Vapaana pyörinyt pyörä oli liian jäykkä kääntymään näin painavan ruohonleikkurin kohdalla. Mahdollisia parantamiskeinoja tähän ruohonleikkuriin ovat siis vahvempien vääntömoottoreiden hankinta sekä toisenlaisen vapaan pyörän hankinta.

Kokonaisuutena opinnäytetyö opetti minulle paljon mekaniikkaan liittyvästä mitoittamisesta, radio-ohjauksesta, tasavirtamoottoreiden ohjaamisesta, mikrokontrollereiden ohjelman teosta sekä ennen kaikkea laitteen tekemisestä vaihe vaiheelta lopulliseksi prototyypiksi.

LÄHTEET

- Agarwal, T. 2014. Know about 3 Important Ways for DC Motor Speed Control. Www-dokumentti. Saatavissa: <https://www.elprocus.com/what-are-the-best-ways-to-control-the-speed-of-dc-motor/>. Viitattu 28.5.2018.
- Ahoranta, J. 2014. Sähkötekniikka. 1.–13., painos. Helsinki: Sanoma Pro Oy.
- Arduino. About Us. Www-dokumentti. Saatavissa: <https://www.arduino.cc/en/Main/AboutUs>. Viitattu 22.5.2018.
- Arduino. Arduino Motor Shield Rev3. Www-dokumentti. Saatavissa: <https://store.arduino.cc/arduino-motor-shield-rev3>. Viitattu 2.5.2018.
- Arduino. Arduino Products. Www-dokumentti. Saatavissa: <https://www.arduino.cc/en/main/products>. Viitattu 23.5.2018.
- Arduino. Arduino Uno Rev3. Www-dokumentti. Saatavissa: <https://store.arduino.cc/arduino-uno-rev3>. Viitattu 3.5.2018.
- Avayan. 2018. Understanding PWM. Www-dokumentti. Saatavissa: <https://eblcdc.com/?p=48>. Viitattu 29.5.2018.
- Daware, K. 2016. Speed control methods of DC motor. Www-dokumentti. Saatavissa: <https://www.electrical4u.com/2014/01/speed-control-methods-of-dc-motor.html>. Viitattu 28.5.2018.
- Electrical4u.com. 2018. Permanent Magnet DC Motor or PMDC Motor | Working Principle Construction. Www-dokumentti. Saatavissa: <https://www.electrical4u.com/permanent-magnet-dc-motor-or-pmdc-motor/>. Viitattu 14.6.2018.
- Elfa Distrelec. Vaihteistolla varustettu DC-moottori 40,5 mm 18:1 12 VDC, E192.12.18, Micro Motors. Www-dokumentti. Saatavissa: <https://www.elfadistrelec.fi/fi/vaihteistolla-varustettu-dc-moottori-40-mm-18-12-vdc-micro-motors-e192-12-18/p/15445515>. Viitattu 7.2.2018.
- Futaba. 2013. Attack 4YWD. Frequency Hopping Spread Spectrum. Pdf-dokumentti. Saatavissa: https://www.rc.futaba.co.jp/english/dl_manual/4ywd_e.pdf. Viitattu 6.5.2018.
- Hirzel, T. PWM. Www-dokumentti. Saatavissa: <https://www.arduino.cc/en/Tutorial/PWM>. Viitattu 29.5.2018.
- Hutasu.net. 2017. Mikä on mikrokontrolleri? Www-dokumentti. Saatavissa: <https://www.hutasu.net/elektroniiikka/sulautettu-elektroniiikka/mika-on-mikrokontrolleri/>. Viitattu 22.5.2018.
- Inkinen, P., Manninen, R. & Tuohi, J. 2012. Momentti 2. Insinöörifysiikka. 2.–5., painos. Helsinki: Kustannusosakeyhtiö Otava.

James, M. 2017. What Are DSM RC Controllers and Receivers and What Do They Do? Www-dokumentti. Saatavissa: <https://www.thoughtco.com/dsm-rc-controllers-and-receivers-2862526>. Viitattu 21.5.2018.

James, M. 2018. Radio Frequencies in the US for Radio Controlled Vehicles. Www-dokumentti. Saatavissa: <https://www.thoughtco.com/radio-frequencies-in-the-us-for-radio-controlled-vehicles-2862530>. Viitattu 21.5.2018.

Karvinen, K. & Karvinen, T. 2010. Sulautetut. Opi rakentamaan robotteja ja muita sulautettuja järjestelmiä. 2., painos. Helsinki: Readme.fi.

K-Rauta. AKKURUOHONLEIKKURI FXA 40V 2,5AH 43CM METALLIRUNKO. Www-dokumentti. Saatavissa: <https://www.k-rauta.fi/rautakauppa/akkuruohonleikkuri-fxa-40v-2-5ah-43cm-metallirunko>. Viitattu 5.2.2018.

Lin, A. 2018. Pulse-Width Modulation (PWM). Www-dokumentti. Saatavissa: <http://www.airsupplylab.com/embedded/61-pulse-width-modulation-pwm.html>. Viitattu 29.5.2018.

Meder, M. 1993. Radio-ohjaustekniikan käsikirja 2. 2., uusittu painos. Järvenpää: Harraste ja Tekniikka.

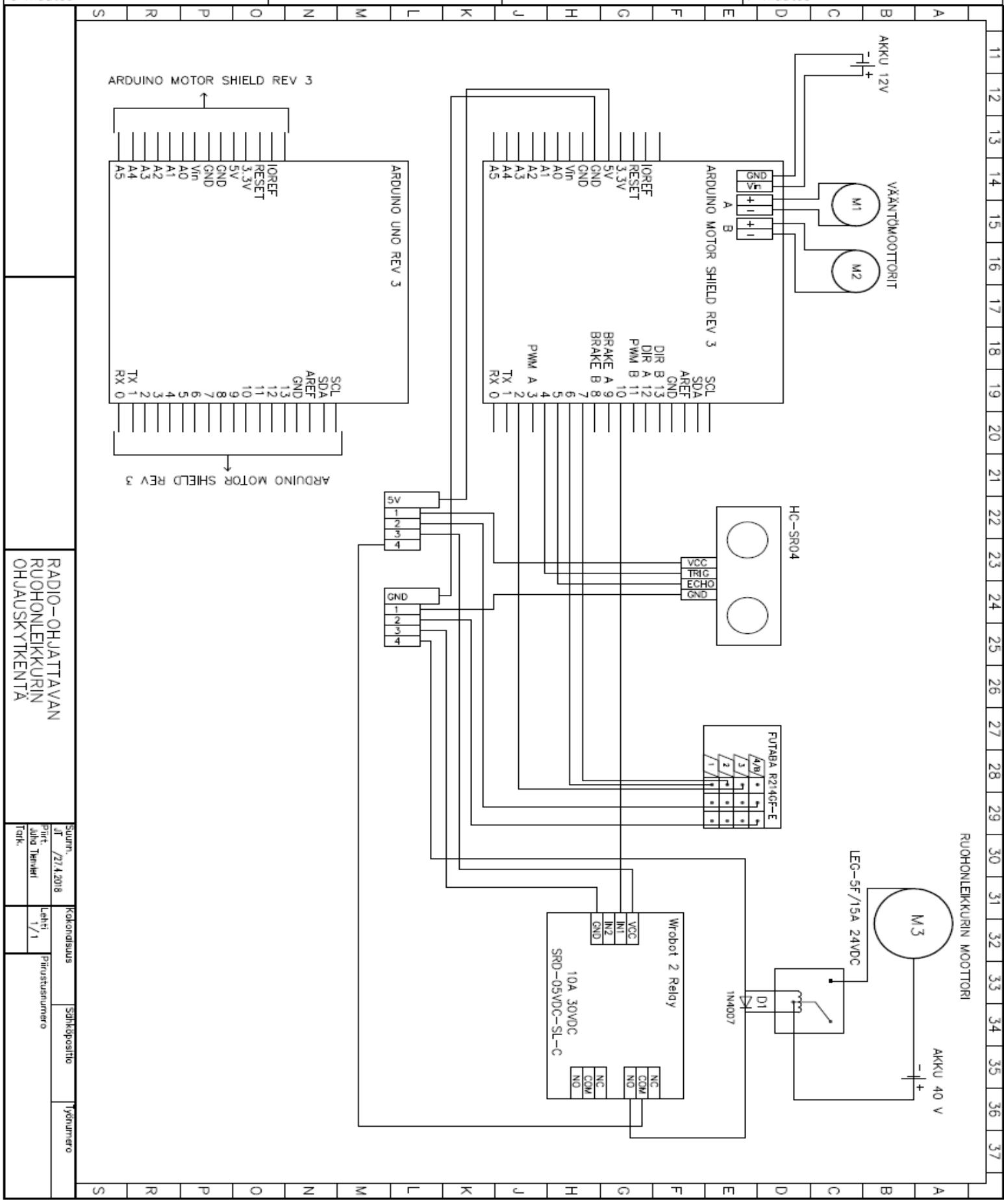
Redfem, R. 2017. FXA. FXA2ALR03-2R3S-430. Käyttöohje. Yhdistynyt kuningaskunta: Nap Brands Ltd.

SKS Mekaniikka Oy. Hammashihnakäytön suunnitteluopas. Pdf-dokumentti. Saatavissa: [https://www.sks.fi/www/sivut/186CEF516D83C523C2257B6B003E6E36/\\$FILE/Hammashihnak%C3%A4ytt%C3%B6jen%20suunnitteluopas.pdf](https://www.sks.fi/www/sivut/186CEF516D83C523C2257B6B003E6E36/$FILE/Hammashihnak%C3%A4ytt%C3%B6jen%20suunnitteluopas.pdf). Viitattu 21.2.2018.

SKS Mekaniikka Oy. Polyuretaanihihnekäytöt. Pdf-dokumentti. Saatavissa: [https://www.sks.fi/www/sivut/186CEF516D83C523C2257B6B003E6E36/\\$FILE/PU_hihnak%C3%A4ytt%C3%B6t_esite.pdf](https://www.sks.fi/www/sivut/186CEF516D83C523C2257B6B003E6E36/$FILE/PU_hihnak%C3%A4ytt%C3%B6t_esite.pdf). Viitattu 19.2.2018.

SparkFun Electronics. Ultrasonic Sensor - HC-SR04. Www-dokumentti. Saatavissa: <https://www.sparkfun.com/products/13959>. Viitattu 7.5.2018.

A muutos	D muutos
B muutos	E muutos
C muutos	F muutos



RADIO-OHJATTAVAN
RUOHONLEIKKURIN
OHJAUSKYTKENTÄ

Siunaus JT / 27.4.2018	Kokonaissuus Lehti 1/1	Sähköpostiosoite	Yhteydennumero
Print Alkio / Terhi	Talk.		

```
[code]
#include <AFMotor.h> //Adafruitin moottoriohjain kirjasto

AF_DCMotor motor1(1); //Toinen moottori M1
AF_DCMotor motor2(2); //Ensimmäinen moottori M2

int pin1 = 7; //Lähettimen kanava kakkonen ns. kaasutus
int pin2 = 6; //Lähettimen kanava ykkönen ns. ohjaus
int pin3 = 2; //Lähettimen kanava kolmonen ns. käynnistys

int in1 = 10;

int neutralh = 1660; //Neutraalin arvon yläraja
int neutrall = 1580; //Neutraalin arvon alaraja
int maxval = 2014; //Ohjaimen kahvat maximissa
int minval = 1222; //Ohjaimen kahvat minimissä

int releh = 1990; //Releen käynnistymisen yläraja
int relel = 1270; //Releen sammuttamisen alaraja

int trigPin = 4; //Ultraäänisensorin Triggeri
int echoPin = 5; //Ultraäänisensorin Echo
long duration, cm; //Etäisyys

void setup()
{
  Serial.begin(9600); //Sarjamonitori 9600 bittiä sekunnissa

  motor1.run(RELEASE); //Toisen moottorin vapautus
  motor2.run(RELEASE); //Ensimmäisen moottorin vapautus

  pinMode(pin1, INPUT); //Kaasutuksen alustus
  pinMode(pin2, INPUT); //Ohjauksen alustus

  pinMode(12, OUTPUT); //Ensimmäisen moottorin suunnanvaihdon alustus
  pinMode(13, OUTPUT); //Toisen moottorin suunnanvaihdon alustus

  pinMode(trigPin, OUTPUT); //Triggerin alustus
  pinMode(echoPin, INPUT); //Echon alustus

  pinMode(in1, OUTPUT); //Releen alustus
  digitalWrite(in1, HIGH); //Releen asettaminen sulkeutuvaksi
}

void loop()
{
```

```

// Map-funktion arvojen alustus
int sV1; //Kaasutus eteen
int sV2; //Kaasutus oikealle
int sV3; //Kaasutus vasemmalle
int sV4; //Peruutus taakse
int sV5; //Peruutus oikealle
int sV6; //Peruutus vasemmalle

int speedValue = pulseIn(pin1, HIGH); //PWM-pulssin lukeminen kanavasta 1
int directionValue = pulseIn(pin2, HIGH); //PWM-pulssin lukeminen kanavasta 2
int start = pulseIn(pin3, HIGH); //PWM-pulssin lukeminen kanavasta 3

Serial.println(speedValue); //Kaasutus arvon virheen testaus
Serial.println(directionValue); //Ohjaus arvon virheen testaus
Serial.println(start); //Käynnistysarvon virheen testaus

if (speedValue >= neutralh && speedValue <= neutralh || speedValue < 100) //Molemmat kahvat
neutraalina tai ohjain sammutetaan, leikkuri pysyy paikallaan
{
  motor1.run(RELEASE); //Toisen moottorin vapautus
  motor2.run(RELEASE); //Ensimmäisen moottorin vapautus
  motor1.setSpeed(0); //Asettaa ensimmäiselle moottorille arvon 0
  motor2.setSpeed(0); //Asettaa toiselle moottorille arvon 0
  delay(50); //50 mikrosekunnin tarkastus
}

if (speedValue > neutralh && directionValue >= neutralh && directionValue <= neutralh && cm >=
10) //Vasen kahva ylöspäin ja oikea kahva neutraalina
//leikkuri kulkee eteenpäin
{
  sV1 = map(speedValue, neutralh, maxval, 140, 255); //Mapfunktio lukee PWM-pulssiarvon ja
asettaa neutraalin ylärajaarvon moottorin lähtönopeudeksi ja
//maksimi arvon moottorin loppunopeudeksi
//Lisäksi lähtönopeus laitetaan alkamaan jo 140, jotta ohjauksesta
tulee vikkellämpi

  if(sV1 > 255) //Ennakkovarotoimi mahdollisten PWM-pulssibugien varalta, jotta moottori ei ylitä
maksiminopeutta
  {
    sV1 = 255;
  }

  motor1.setSpeed(sV1); //Asettaa ensimmäiselle moottorille ohjatun nopeuden
  motor2.setSpeed(sV1); //Asettaa toiselle moottorille ohjatun nopeuden
  motor1.run(FORWARD); //Laittaa ensimmäisen moottorin kulkemaan määritetyllä nopeudella
  motor2.run(FORWARD); //Laittaa toisen moottorin kulkemaan määritetyllä nopeudella
  Serial.println("A"); //Monitorointia varten, eteenpäin on yhtäkuin A, seuraava B jne.
  digitalWrite(13, LOW); //Toisen moottorin pyörimissuunta vastapäivään
  digitalWrite(12, HIGH); //Ensimmäisen moottorin pyörimissuunta myötäpäivään

```

```
delay(50); //50 mikrosekunnin tarkastus

}

else if (cm <= 10)

{

motor1.setSpeed(0);
motor2.setSpeed(0);

}

if (directionValue > neutralh && speedValue > neutralh && cm >= 10) //Vasen kahva ylöspäin ja
oikea kahva oikealla

                                //leikkuri kulkee eteenpäin oikealle

{
sV2 = map(speedValue, neutralh, maxval, 140, 255); //Eteenpäin oikealle

if(sV2 > 255)
{
sV2 = 255;
}

motor1.setSpeed(sV2);
motor2.setSpeed(0);
motor1.run(FORWARD);
motor2.run(FORWARD);
Serial.println("B");
digitalWrite(13, LOW);
delay(50);
}

else if (cm <= 10)

{

motor1.setSpeed(0);
motor2.setSpeed(0);

}

if (directionValue < neutrall && speedValue > neutralh && cm >= 10) //Vasen kahva ylöspäin ja
oikea kahva vasemmalle

                                //leikkuri kulkee eteenpäin vasemmalle

{
sV3 = map(speedValue, neutralh, maxval, 140, 255); //Eteenpäin vasemmalle
```

```

if(sV3 > 255)
{
  sV3 = 255;
}

```

```

motor1.setSpeed(0);
motor2.setSpeed(sV3);
motor1.run(FORWARD);
motor2.run(FORWARD);
Serial.println("C");
digitalWrite(12, HIGH);
delay(50);
}

```

```

else if (cm <= 10)

```

```

{

motor1.setSpeed(0);
motor2.setSpeed(0);

}

```

```

if (speedValue < neutrall && directionValue >= neutrall && directionValue <= neutralh && direc-
tionValue > 100 && speedValue > 100) //Vasen kahva alaspäin ja oikea kahva neutraalina

```

```

//leikkuri kulkee

```

```

taaksepäin

```

```

//Lisäksi, jos oh-

```

```

jain sammutetaan estetään ehdoilla

```

```

//peruutus

```

```

{
sV4 = map(speedValue, neutrall, minval, 140, 255); //Peruutus

```

```

if(sV4 > 255)
{
  sV4 = 255;
}

```

```

motor1.setSpeed(sV4);
motor2.setSpeed(sV4);
motor1.run(BACKWARD);
motor2.run(BACKWARD);
Serial.println("D");
digitalWrite(12, LOW);
digitalWrite(13, HIGH);
delay(50);
}

```

```

else if (directionValue < 100 || speedValue < 100)

```

```

{

```

```

motor1.setSpeed(0);
motor2.setSpeed(0);
}

```

```

if (directionValue > neutralh && speedValue < neutrall && directionValue > 100 && speedValue
> 100) //Vasen kahva alaspäin ja oikea kahva oikealle

```

//leikkuri kulkee taaksepäin oikealle
//Sisältää myös saman lisäehdon kuin

edellisessä

```
{
```

```
sV5 = map(speedValue, neutralh, minval, 140, 255); //Taaksepäin oikealle
```

```

if(sV5 > 255){
  sV5 = 255;
}

```

```

motor1.setSpeed(sV5);
motor2.setSpeed(0);
motor1.run(BACKWARD);
motor2.run(BACKWARD);
Serial.println("E");
digitalWrite(13, HIGH);
delay(50);
}

```

```

else if (directionValue < 100 || speedValue < 100)
{
  motor1.setSpeed(0);
  motor2.setSpeed(0);
}

```

```

if (directionValue < neutrall && speedValue < neutrall && directionValue > 100 && speedValue >
100) //Vasen kahva alaspäin ja oikea kahva vasemmalle

```

//leikkuri kulkee taaksepäin vasem-
malle

edellisessä

```
{
```

```
sV6 = map(speedValue, neutralh, minval, 140, 255); //Taaksepäin vasemmalle
```

```

if(sV6 > 255){
  sV6 = 255;
}

```

```

motor1.setSpeed(0);
motor2.setSpeed(sV6);
motor1.run(BACKWARD);
motor2.run(BACKWARD);

```



```
Serial.println("F");
digitalWrite(12, LOW);
delay(50);
}
else if (directionValue < 100 || speedValue < 100)
{
motor1.setSpeed(0);
motor2.setSpeed(0);
}

if (start > releh && cm >= 10)
{
digitalWrite(in1, LOW);
}
else if(start < relel || cm <= 10){
digitalWrite(in1, HIGH);
}

//Ultraäänianturi lähettää kymmenen mikrosekunnin välein pulssin
//Lyhyellä viiden mikrosekunnin LOW pulssilla varmistetaan puhdas HIGH pulssi

digitalWrite(trigPin, LOW);
delayMicroseconds(5);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

//Lukee signaalin ultraäänianturista, HIGH pulssin kesto ilmoitetaan mikrosekunneissa, Echolla lue-
taan etäisyys kohteesta takaisin
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);

cm = (duration/2) / 29.1; //Ajan muuttaminen matkaksi
//inches = (duration/2) / 74;
//Serial.print(duration);
//Serial.print("duration");
//Serial.print(inches);
//Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(350);
}
[/code]
```