



Mika Ollikainen

# Kauko-ohjattava rakennusteline Körmy

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikan tutkinto-ohjelma

Insinöörityö

7.5.2023

## Tiivistelmä

Tekijä:	Mika Ollikainen
Otsikko:	Kauko-ohjattava rakennusteline Körmy
Sivumäärä:	33 sivua + 2 liitettä
Aika:	7.5.2023
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Ammatillinen pääaine:	Automaatiotekniikka
Ohjaajat:	Lehtori Antti Liljaniemi

---

Insinööriyön tavoitteena oli viedä loppuun innovaatioprojektina aloitettu ja Körmyksi nimetty kauko-ohjattavan rakennustelineen toteutus. Toteutus oli tarkoitus tehdä konseptitasoisena, joten tavoitteena oli saada Körmy perustoiminnallisuus liikkumisen ja ohjauksen osalta toimimaan. Projektin tilaajana toimi Rakennuspalvelu Ilmari Iho Oy. Asiakkaan käyttötarkoitukseen visioitua ratkaisua ei hänen omien tutkimustensa mukaan ole kaupallisesti saatavilla.

Projektissa selvitettiin kuluttajakäyttöön tarkoitetun sähköskootterin BLDC-napamootoreiden ohjauksen toteutus ja akuston soveltuvuus projektin tehölähteeksi. Ohjaimena käytetään PlayStation 4:n ohjainta. Ohjauslogiikasta vastaa Raspberry Pi 4 -minitietokone ja ohjelmointikielenä toimii Python.

Päättyöyössä kuvataan projektin eteneminen ja perustellaan tehtyjä ratkaisuja komponenttivalintojen ja ohjelmallisten toteutusten osalta. Projektin aikana tutustuttiin myös robottien ja mobiilirobottien historiaan ja mobiilirobottien hyödyntämiseen rakennuskohteissa.

Projektissa onnistuttiin saavuttamaan sille asetetut tavoitteet. Projektin ja siitä saatujen oppien pohjalta jatkokehitys edelleen on mahdollista.

Avainsanat: mobiilirobotti, kauko-ohjaus, rakennusteline, Python, Raspberry Pi

## Abstract

Author: Mika Ollikainen  
Title: Remotely Controlled Construction Platform Körmy  
Number of Pages: 33 pages + 2 appendices  
Date: 7 May 2023

Degree: Bachelor of Engineering  
Degree Programme: Electrical and automation engineering  
Professional Major: Automation engineering  
Supervisors: Antti Liljaniemi, Senior Lecturer

---

The scope of this engineering thesis project was to complete remotely controlled construction platform called Körmy. The project was started as innovation project. Project should be completed on concept level, which means that basic functionalities of Körmy, such as moving and turning, work. Körmy was ordered by Rakennuspalvelu Ilmari Iho Oy. Based on customer's own research, there are no construction platforms available on the market that match with customer's vision.

Part of the project was to investigate the control method for BLDC motor taken from commercial e-scooter and find out if the scooter's battery pack can be utilized as power source. PlayStation 4 controller was chosen as controller for the project. The main unit for control logic is Raspberry Pi 4 and used programming language is Python.

Thesis project includes description of how the project developed and motivations behind made decisions regarding used components and programming methods. A brief study about history of robotics and mobile robotics, as well as utilization of mobile robots in construction projects, was conducted during the project.

Goals of the project were achieved. Gathered knowledge and lessons learned from the project can be used to further develop Körmy.

Keywords: mobile robot, remote control, construction platform, Python, Raspberry Pi

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Robotti	2
2.1	Mobiilirobotti	3
2.2	Mobiili- ja kuljetusrobotit rakennusmaailmassa	4
3	Körmyn perusrakenne ja osat	5
3.1	Mekaaninen perusrakenne	5
3.2	Sähköjärjestelmän perusrakenne	7
3.3	Raspberry Pi 4	7
3.4	DUALSHOCK®4-ohjain	8
4	Körmyn toteutus	8
4.1	Ohjausjärjestelmä	8
4.1.1	DC-moottoreiden ohjaus yleisesti	8
4.1.2	Kääntömoottorin ohjaus	9
4.1.3	BLDC-moottori	10
4.2	Sähköjärjestelmä	14
4.2.1	Akusto ja sen testaus	14
4.2.2	Jännitetasojen määrittely ja toteutus	16
4.2.3	Elektroniikan signaalien kytkentä	18
4.2.4	Komponenttien asennus	20
4.3	Körmyn aivot	21
4.3.1	Raspberry Pi 4:n käyttöönotto ja vaadittavat asennukset ja kirjastot	21
4.3.2	Ohjausohjelmisto ja Python-koodi avattuna	23
5	Yhteenveto	30
	Lähteet	32
	Liitteet	
	Liite 1: Osalista	
	Liite 2: Piirikaavio	

## Lyhenteet

PWM: *Pulse Width Modulation*. Pulssinleveysmodulaatio on elektroniikassa käytettävä signaalityyppi. Signaali saa arvoksi joko 0, tai 1 ja pulssin pituus vaihtelee määritellyn aikayksikön sisällä.

BLDC: *Brushless DC electric motor*. Harjaton tasavirtamoottori ei nimensä mukaisesti vaadi harjoja kontaktoreille. Moottorin ohjaus vaatii staattorikämmien ajoituksen ja siten monimutkaisemman logiikan verrattuna tavalliseen DC-moottoriin.

# 1 Johdanto

Projekti Körmy on osana Big Flash -hanketta toteutettava liikkuva rakennusteline ja työtaso, jonka asiakasyritys Rakennuspalvelu Ilmari Iho Oy on tilannut Metropolialta. Pohjatyö robotin mekaanisen perusrakenteen ja ohjauksen osalta on toteutettu innovaatioprojektina keväällä 2022.

Asiakkaan visiona on käyttää Körmyä erilaisissa vaihtelevissa maalaus- ja rakennuskohteissa, joissa liikkuva rakennusteline mahdollistaa joustavamman työskentelyn sekä tuo selkeää ajansäästöä. Asiakkaan omien tutkimusten perusteella visioituihin tarpeisiin edes etäisesti sopivaa ratkaisua ei ole markkinoilta saatavilla.

Insinööriyön tavoitteena on suunnitella ja toteuttaa Körmyn kauko-ohjaus hyödyntäen Raspberry Pi -minitietokonetta ja langatonta PlayStation 4 -ohjainta. Robotin vetotavaksi valikoitui innovaatioprojektin yhteydessä kaksi erillistä kaupallisesta sähköpotkulaudasta irrotettua napamoottorilla varustettua vetävää pyörää. Ohjauksen toteutuksessa päädyttiin hyödyntämään tavallista harjallista tasavirtamoottoria. Vetävien pyörien moottorin tyyppi on harjaton tasavirtamoottori. Näitä kolmea moottoria ohjataan erillisillä moottorinohjauspiireillä, joille ohjaussignaali tuotetaan hyödyntämällä Raspberry Pi:n tuottamaa PWM-signaalia. Kokonaisuus on tarkoitus saada toimimaan konseptitasolla, jonka pohjalta myöhempi jatkokehitys on mahdollista.

Projektin aikana tullaan tutkimaan BLDC- ja DC-moottoreiden ohjausta yleisesti. Vetävien pyörien ominaisuudet tulee myös selvittää, sillä kaupallisesta tuotteesta irrotetun moottorin ominaisuuksia ei ole julkisesti saatavilla. Projektin aiemmassa vaiheessa on jo hankittu muutama ohjainkortti, joita on tarkoitus hyödyntää projektin toteutuksessa. Moottoreiden perusominaisuuksiin ja ohjaustapoihin tutustutaan hyödyntämällä aiempia tutkimuksia ja päättötyöprojekteja vastaavasta aiheesta. Näiden perusteella testataan ohjauskorteille syötettävän ohjaussignaalin toteutusta ja valitaan paras tapa kuhunkin tilanteeseen.

## 2 Robotti

Robotti-sanan etymologiseksi alkuperäksi mielletään tšekkiläisen näytelmäkirjailijan Karel Čapekin vuonna 1920 julkaistussa teoksessa Rossum's Universal Robots kuvailema ihmismäinen biologinen kone, jota kutsuttiin nimellä ”robota”. Tšekinkielinen robottia tarkoittava termi on yhdistelmä sanoista ”rabota”, joka tarkoittaa pakkotyötä ja ”robotnik”, joka tarkoittaa maaorjaa. On kuitenkin varsin selvää, että historian saatossa jo kauan ennen Čapekin popularisoimaa robotti-termiä ihmistä palvelevan tai automaattisesti työtä tekevän systeemin idea on ollut useamman keksijän ja insinöörin ajatuksissa ja sen vuoksi robotin idean alkuperää on käytännössä mahdotonta määritellä. [1. s. 1–2.]

Ajan saatossa idea siitä, mitä robotilla tarkoitetaan ja millaisia tehtäviä robotti voisi kyetä suorittamaan, on elänyt tieteen ja teknologian kehittyessä. Yksinkertaisimmillaan robotilla on tarkoitettu edistynyttä mekaanista konetta. Tämän määritelmän mukaisia koneita olivat esimerkiksi 1600- ja 1700-luvuilla rakennetut kellokoneistoihin perustuvat automaattit, jotka kykenivät matkimaan luonnollista liikettä, tai vaikka soittamaan pianoa. Yksi moderni robotin määritelmä on systeemi, jonka tulee samanaikaisesti olla autonominen, sijaita fyysisessä maailmassa, kyetä havainnoimaan ympäristöään ja kyetä suorittamaan sille määritettyjä tavoitteita. [1, s. 1–5.]

IFR – International Federation of Robotics on kansainvälinen voittoa tavoittelematon teollisuus- ja palvelurobotiikkaan keskittyvä yhteistyöelin, joka on mukana standardoimassa robotiikkaan liittyviä termejä ja teknologioita yhdessä kansainvälisen standardoimisorganisaation kanssa. IFR [2] nojaa robotiikkaan liittyvissä määritelmässä viralliseen ISO 8373 -standardiin [3], jonka mukainen määritelmä robotille on ohjelmoitava mekanismi, joka kykenee toimimaan jollain tasolla autonomisesti suorittaen liikkumista, käsittelyä tai paikoitusta. Robotti itsessään taas koostuu ohjausjärjestelmästä ja toimiyksiköstä, joka voi olla toimitusväline, mobiilialusta tai puettava mekanismi.

IFR:n määrittelyyn sopii esimerkiksi William Grey Walterin rakentamat Machina speculatrix -koneet, joita pidetään ensimmäisinä varsinaisina mobiilirobotteina. Hän rakensi ensimmäiset tällaiset robotit vuosien 1948–1949 aikana ja antoi niille nimet Elmer ja Elsie. Robotit koostuivat kolmesta renkaasta, fotodiodista sekä ohjauslogiikasta. Logiikan toteutuksesta johtuva fotofiilisyyys yhdistettynä niiden verkkaiseen liikenopeuteen ja ulkomuotoon sai Walterin antamaan niille lempinimen tortoise eli ”kilpikonna”. [4.]

## 2.1 Mobiilirobotti

IFR määrittelee mobiilirobotin robotiksi, joka kykenee matkustamaan itsenäisen ohjauksen varassa. Mobiilirobotti on alusta, jossa voi olla toimielimiä ja siinä voi myös olla mahdollisuus kauko-ohjaukselle. [2.]

Karkeasti mobiilirobotit voidaan jakaa kahteen pääryhmään. Perinteisempää teknologiaa edustaa AGV (Automated guided vehicle), joka seuraa ennalta määritettyä reittiä. Long ja Zhang [5. s. 3] listaavat ennalta määrittelyyn reitin luomiseksi useita käytössä olevia tekniikoita, jotka perustuvat fyysiseen rakentamiseen, tai langattomaan signaaliin, joita ovat esimerkiksi sähkömagneettinen rata, laserpulssein heijastus, kuvantunnistus ja GPS.

AMR (Autonomous mobile robot) kykenee reitinmäärittelyyn itsenäisesti. Intel määrittelee autonomisen robotin järjestelmäksi, joka kykenee ymmärtämään ympäristöään ja liikkumaan siellä itsenäisesti. Autonominen robotti käyttää edistynyttä yhdistelmää eri sensoreita, tekoälyä, koneoppimista ja laskentaa reitin suunnitteluun ja navigointiin ympäristössään. Kamerajärjestelmän ansiosta autonomiset robotit kykenevät väistämään reitille päätyneitä esteitä, kuten pudonneita laatikoita, tai ihmisiä ja jatkamaan tämän jälkeen suunnitellun tehtävän suorittamista. [6.]

Autonomisesti maalla liikkuvien robottien merkkipaaluihin lukeutuu W. G. Waltersin kilpikonniin lisäksi 1960-luvun lopulla kehitetty Shakey, joka oli varustettu kameralla, etäisyysmittareilla ja törmäystunnistimilla. Shakey osasi suorittaa



yksinkertaisia navigointitehtäviä ja joitakin käyttäjän sille määrittelemiä tehtäviä purkamalla annetun tehtäväkokonaisuuden pienemmiksi kokonaisuuksiksi ja suorittamalla kunkin kokonaisuuden erikseen. Samoihin aikoihin aloitettiin Stanford Cardin kehitys, joka valmistuessaan 1970-luvun lopulla osasi mm. suunnistaa tuoleilla täytetyn huoneen läpi ilman ihmisen avustusta. Suunnistus perustui kameralla otettuihin kuviin ja sen pohjalta luotuun kappaleiden etäisyysmalliin. 80-luvulla kehitettiin MIT:n toimesta Ghengis Khan, joka kykeni 6 raajansa ansiosta ylittämään erilaisia esteitä. [7. s. 8–10.]

80-luvulla kehitettiin useita erilaisia tutkimuskäyttöön suunnattuja mobiilirobotteja. 90-luvulle tultaessa yleistyivät tutkimukseen käytettävät robotit, joista esimerkiksi Carnegie Mellon yliopiston kehittämät Dante-1 ja Dante-2 suunniteltiin tutkimaan aktiivisia tulivuoria. Myös ensimmäinen Mars-kulkija Sojourner aloitti tehtävänsä 1990-luvun lopulla. Vuosituhannen vaihteessa näki päivänvalon entistä pienikokoisemmat ja kehittyneemmät robotit, kuten Sony Aibo ja Honda Asimo. [8, s. 2–5.]

Mobiilirobottien kehityksen kannalta olennaista on ollut niiden keräämän ja käsittelemän datamäärän kasvava hyödyntäminen. Aiemmin ongelmana oli rajallinen sensori- ja kamerateknologia yhdistettynä rajalliseen laskentakapasiteettiin. Nykyisin tämä ongelma on pienenemään päin, ja sensoreiden ja tekoälyn kehityksessä on otettu huomioon askeleita. Mobiilirobotiikan kärkeä edustaa tällä hetkellä kaupallisesti saatavilla oleva Boston Dynamicsin Spot [9] ja pitkäaikainen tutkimusprojekti Atlas [10], joka kykenee ketterään ihmismäiseen liikkumiseen ja asennon korjaamiseen sekä akrobatiaan.

## 2.2 Mobiili- ja kuljetusrobotit rakennusmaailmassa

Vaikuttaisi, että rakennusmaailmassa mobiilirobottien hyödyntäminen ei vielä ole kovin suurta huomiota saava aihe. Syynä tähän on varmaankin rakennusprojektien vaihteleva luonne ja haastava ympäristö. Robotin pitäisi kyetä toimimaan ja havainnoimaan ympäristöään yhdellä tavalla esimerkiksi tavallisen asuinkerrostalon rakennustyömaalla ja toisella tavalla vaikka vaativan

teollisuuskompleksin rakennustyömaalla. Saman robotin kyvykkyys toimia hyvin vaihtelevissa ympäristöissä on toki toteutettavissa, mutta oletettavasti käyttökohteeseen liittyvästä kustomoinnista aiheutuvat kustannukset ylittäisivät säästettävät hyödyt.

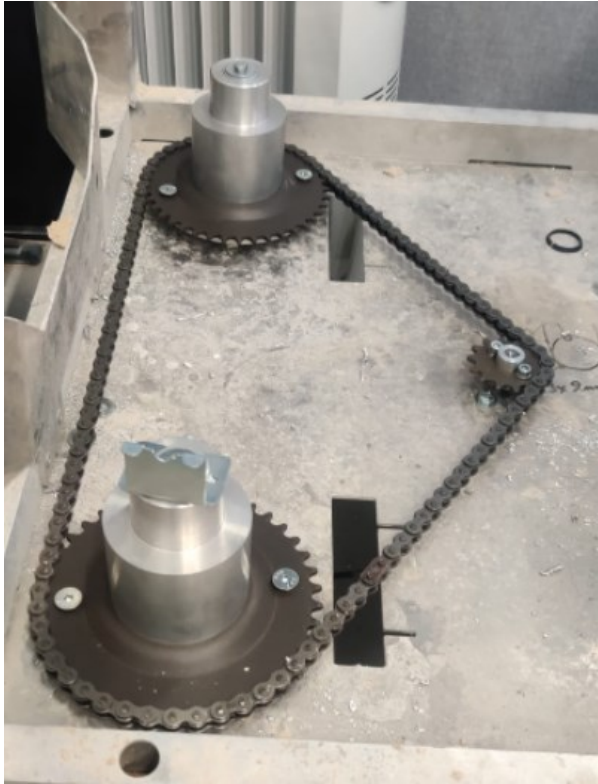
Körmylle visioituihin tehtäviin ei löytynyt markkinoilta suoraan vastaavaa tuotetta. Rakennusmaailmaan on toki saatavilla erilaisia kauko-ohjattavia alustoja ja trukkeja, mutta nämä on tarkoitettu pääasiassa raskaan tai muuten hankalasti siirrettävän materiaalin kuljetukseen. Vaikuttaisi, että merkittävä osa kaupallisesti saatavilla olevista kuljetusalustoista on samalta valmistajalta, mutta pienillä myyjäkohtaisilla kustomoinneilla varustettuja. Esimerkki tällaisesta on Ergo-Mover Wireless [11], joka myyjän mukaan kykenee 1500 kg kuorman käsittelyyn ja jonka ohjaus tapahtuu langattomalla ohjaimella. Toisena esimerkkinä löytyi nostotasolla varustettu langattomasti ohjattava kuljetusalusta PEF 500 [12].

### **3 Körmyrakenne ja osat**

#### **3.1 Mekaaninen perusrakenne**

Mekaanisen toteutuksen osalta Körmyrakenneeseen ei tämän projektin puitteissa ollut tarkoitus, eikä tarve koskea. Körmyrakenne pohjana toimii asiakkaan toimittama kaupallisesti saatavilla oleva alumiininen rakennusteline, jonka pohjalevy on kooltaan noin 1200 mm x 700 mm. Pohjalevyyn on projektin yhteydessä kiinnitetty 4 pyörää, ohjauselektronikka ja mekaaninen ohjausjärjestelmä.

Ohjauksen perusrakenne on toteutettu innovaatioprojektina keväällä 2022. Rakenne koostuu kahdesta napamoottorista, jotka on irrotettu kaupallisesta Xiaomi Essential Lite -sähköpotkulaudasta ja ne on liitetty Körmyrakenne pohjalevyyn tätä varten valmistetulla laakeriperällä. Laakeriperään on kiinnitetty ohjausrattaat, joihin ohjaus välittyy kääntömoottorilta ketjun välityksellä. Ohjauksen mekaaninen perusrakenne innovaatioprojektin tuloksena näkyy kuvissa 1 ja 2.



Kuva 1. Ohjauksen mekaaninen perusrakenne pohjalevyn yläpuolella.



Kuva 2. Ohjauksen mekaaninen perusrakenne pohjan puolelta.

## 3.2 Sähköjärjestelmän perusrakenne

Sähköjärjestelmän osalta projektin toteutuksessa pohdittiin eri vaihtoehtoja käytettävän jännitetason osalta. Haasteeksi muodostui kolme tarvittavaa jännitetasoa, joista napamoottoreiden ominaisjännite on 36 V, käänmömoottorin 24 V ja Raspberry Pi 4:n 5 V. Projektissa päädyttiin hyödyntämään Xiaomi-sähköskootterin akkua ja asettamaan jännitetasot jännitteenalentimilla halutuiksi. Projektin edetessä Xiaomin akut osoittautuivat melko haastaviksi, mikä johtuu niitä suojaavasta järjestelmäpiiristä, joka rajoittaa ainekin saatavilla olevaa hetkellistä virtaa. Tämän vuoksi myös napamoottoreiden jännite päädyttiin kuristamaan 24 V:n tasoon, joka testauksen perusteella todettiin riittäväksi projektin luonne huomioiden. Kustannussyistä ja projektin konseptiluoteen takia päädyttiin pitäytymään Xiaomin 36 V:n akussa sen sijaan, että eri jännitetasot asetetaan omilla akuillaan.

## 3.3 Raspberry Pi 4

Körmyn ohjausjärjestelmän keskusyksikkönä toimii Raspberry Pi 4. Raspberry Pi 4 on suorituskykyinen ja monipuolinen minitietokone, jolla on laaja tuki niin valmistajan kuin maailmanlaajuisen kehittäjäyhteisön toimesta. Laajan suosion takia sitä käytetään niin harrastelijaprojekteissa kuin teollisissa ja kaupallisissa toteutuksissa. Käytössä on myös analogisen signaalin tuottamiseen ja lukemiseen kykenevä ADC-DAC Pi Zero -lisäkortti.

Raspberry Pi 4:n käyttöjärjestelmänä toimii valmistajan oma Linux-jakelu Raspberry Pi OS. Käyttöjärjestelmä on kevyt ja siinä on valmiiksi asennettuna tärkeimmät oheisohjelmat ja ohjelmakirjastot. Ohjelmointikielistä valmistajalla on suora tuki ohjelmakirjastojen muodossa useille eri kielille, kuten Pythonille, C:lle ja Scratchille. Tässä projektissa ohjelmakieleksi valikoitui Python, joka ei välttämättä nopeutensa puolesta ole paras mahdollinen, mutta tämän projektin vaatimukseen enemmän kuin riittävä. Lisäksi kaikki tarvittavat ohjelmakirjastot löytyivät suoraan DUALSHOCK®4-ohjaimelle ja ADC-DAC-lisäkortille.

Körmyyn vaatimuksiin sopisi myös kevyempi ohjelmoitava mikrokontrolleri, kuten Arduino UNO, tai vastaava lukuisista eri vaihtoehdoista. Päädyin tämän projektin puitteissa kuitenkin pysymään Raspberry Pi 4:ssä sen tarjoaman laajan ohjelmakirjastotuen ja itselle tutun Python-kielen takia.

### 3.4 DUALSHOCK®4-ohjain

Ohjaimeksi valittiin PlayStation 4 -konsolin yhteydessä julkaistu langaton DUALSHOCK®4-ohjain innovaatioprojektivaiheessa. Ohjaimen etuna on helppo saatavuus ja laaja ohjelmakirjastojen tuki erilaisissa kehitysprojekteissa. Ohjain tukee langattomassa tiedonsiirrossa Bluetooth-yhteyttä ja sen yhdistäminen Raspberry Pi 4:ään on varsin suoraviivaista, mikä johtuu minitietokoneen rauta- ja käyttöliittymätason tuesta.

## 4 Körmyyn toteutus

### 4.1 Ohjausjärjestelmä

#### 4.1.1 DC-moottoreiden ohjaus yleisesti

Suuritehoisempia DC-moottoreita ohjataan erillisillä moottorinohjauspiireillä. Ohjauspiirejä on saatavilla kattavasti eri toimittajilta ja myyntipaikoista. Projektiin valikoitui Electromenin valikoimasta löytyvät ohjainkortit, joista kääntömoottorin ohjainkortti todettiin toimivaksi ja soveltuvaksi projektiin innovaatioprojektivaiheessa. Päättyöprojektissa tuli selvittää soveltuvatko valmistajan 2 valittua BLDC-ohjainkorttia sähköpotkulaudasta irrotettujen napamoottoreiden ohjaamiseen.

Xiaomin sähköskootterissa on luonnollisesti myös oma ohjainkortti. On todennäköistä, että Xiaomin oma moottorinohjauskortti vaatisi kännykkälinkitykseen perustuvan ajoneston ohittamista. Lisäksi ohjaussignaalin tyyppi vaatisi lisäselvitystä. Aikataulullisista syistä Xiaomin oman ohjauskortin toimintaa ei lähdetty

tarkemmin selvittämään, vaan päädyttiin kokeilemaan tilattujen ohjainkorttien soveltuvuutta projektiin.

#### 4.1.2 Kääntömoottorin ohjaus

DC-moottorin rakenne perustuu kiinteään staattoriin ja pyörivään roottoriin. Staattori pysyy nimensä mukaisesti paikallaan, ja se sisältää kestopagneetit, kun taas roottori on pyörivä osa koostuen akselista, harjoista ja roottorikämmistä. Luomalla jännite-ero roottorin käänin napojen välille sen läpi kulkee virta, joka saa aikaan magneettikentän muodostumisen käämiin. Virran kulkusuunnasta suhteessa staattorin magneettikentän suuntaan roottori alkaa pyörimään joko myötä- tai vastapäivään. Perinteinen DC-moottori on siis toiminnaltaan varsin yksinkertainen, ja sen ohjauksen toteutus on varsin suoraviivaista. Lyhykäisyydessään perinteisen DC-moottorin ohjaus tapahtuu säätämällä moottorille syötettävän jännitteen suuruutta ja sen napaisuutta. Erillisellä moottorinohjauskortilla voidaan perinteisen DC-moottorin ohjaukseen tuoda erilaisia lisäominaisuuksia, jotka voivat olla hyödyllisiä käyttökohteen tarpeesta riippuen.

Körmyin kääntömoottoriksi valittiin innovaatioprojektivaiheessa DC-moottori Mädler SFS 3 tyyppinumeroltaan 430 486 24. Moottorin ominaisjännite on 24 V, ja se kykenee tuottamaan 1,3 Nm jatkuvan väännön liikkeellelähtöväännön ollessa maksimissaan 25 Nm [13, s. 6]. Moottorin valinnasta ei ole innovaatioprojektivaiheen jäljiltä dokumentointia, mutta lukujen valossa sen pitäisi olla enemmän kuin riittävä projektin tarpeisiin.

Kääntömoottorin ohjauskortiksi valittiin niin ikään innovaatioprojektivaiheessa EM-176A. Ohjainkortti on suunniteltu 24 V:n ominaisjännitettä käyttäville DC-moottoreille, ja sen koko säätöalue on 0–29 V maksimivirran ollessa 10 A [14]. Ohjainkortissa on runsaasti lisäominaisuuksia, joista tämän projektin kannalta olennaisia ovat nopeussignaali, suunta ja vapaa. Kääntömoottorin nopeus tulee olemaan hidas, joten korkeimmalla mahdollisella pyörimisnopeudella ei ole tämän projektin kannalta merkitystä niin moottorin kuin moottorinohjaimen osalta.

Kääntömoottori ja ohjainkortti todettiin toimiviksi innovaatioprojektin aikana. Moottorinohjaukselle oli luotu yksinkertainen ohjauuskoodi ja ohjaussignaali syötettiin suoraan Raspberry Pi:n GPIO-liitimestä moottorinohjaimelle. Sinällään tämä on toimiva ratkaisu, joskaan ei optimaalinen, mikä johtuu Raspberry Pi:n lähdön alemmasta 3,3 V:n jännitetasosta moottorinohjaimen tulon ollessa 0–5 V.

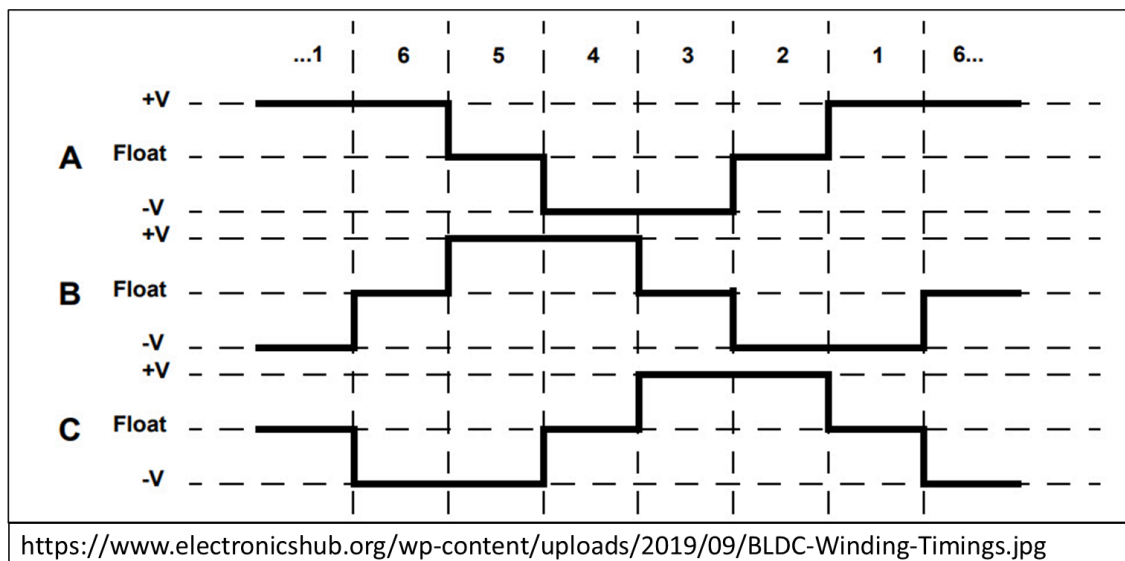
Omien testien yhteydessä havaitsin, että jostain syystä aiemmin tehty kääntömoottorin ohjaus ei toimi luotettavasti ja toistettavasti etenkin käytettäessä pienempiä nopeuksia. Moottorinohjaimen toimintaa päätettiin testata käyttämällä ulkoista tehölähdettä ja pienempää 24 VDC -moottoria. Testin aikana nopeuden säätämiseen käytettiin potentiometriä. Tällaisella testillä ohjainkortti todettiin toimivaksi odotetulla tavalla. Kuitenkin käytettäessä Raspberry Pi:n tuottamaa ohjaussignaalia havaittiin ongelmien uusiutuvan. Tämän oletetaan johtuvan moottorinohjaimen suodattamattomasta nopeuden ohjaussignaalin tulosta, mikä tarkoittaa sitä, että ohjaussignaalksi ei kelpaa Raspberry Pi:n tuottama PWM-signaali, vaan signaalin tulee olla DC-muotoista. Ongelma ratkaistiin käyttämällä Raspberry Pi:hin kytkettyä ADC-DAC-lisäkorttia ja sen tuottamaa analogista signaalia.

#### 4.1.3 BLDC-moottori

BLDC-moottorista käytetään myös nimitystä harjaton tasavirtamoottori. BLDC-moottorin toiminta perustuu kiinteään magneettiin perustuvaan roottoriin ja käämitettyyn staattoriin. Staattorin käämitys vaihtelee tyypillisesti 2–8 navan välillä. Voiman tuottamiseksi roottorille ohjataan virta staattorikäämeille vuoron perään ajoitetusti. Ajoituksessa käytetään roottorin paikkatietoa, joka saadaan Hall-antureiden antamasta signaalista. Hall-anturi antaa arvoksi 1 tai 0 riippuen, onko anturin kohdalla magneetin pohjois- vai etelänapa. [15.]

6-napaisen BLDC-moottorin ajoituskaavio yhden roottorin kierroksen ajalta näkyy kuvassa 3. Kaavion pohjalta voidaan todeta, että kun roottoria halutaan

liikuttaa, niin samanaikaisesti yhdelle käämille ohjataan positiivinen ja yhdelle käämille negatiivinen jännite yhden käämin jäädessä kytkeytymättä. [15.]



Kuva 3. Esimerkki BLDC-moottorin ajoituskaaviosta [15].

BLDC-moottorin ohjaus on siis jonkin verran monimutkaisempaa johtuen tarvittavasta ohjauslogiikasta. Nykyisin ohjauslogiikka ei kuitenkaan tuota ongelmia johtuen elektroniikan ja teknologian kehityksestä. Lisäksi selkeänä etuna verrattuna perinteiseen harjalliseen DC-moottoriin siinä on kuluviin mekaanisten osien rajatumpi määrä.

Körmyin vetävien pyörien napamoottori on tyypiltään BLDC-moottori. Kuvassa 4 näkyy napamoottori avoimena ja siitä on erotettavissa staattorin käämitys ja Hall-antureille tulevat johtimet.





Kuva 4. Xiaomin napamoottori avattuna.

Projektiä varten on tilattu 2 erillistä BLDC-moottorinohjauskorttia, joista päädyttiin testaamaan mallia EM-347A, mikä johtuu sen korkeammasta 36 V:n ominaisjännitteestä, joka vastaa Xiaomin akuston ominaisjännitettä. Teknisten tietojen perusteella EM-347A kykenee ohjaamaan jopa 60 A:n kuormaa [16]. Ohjainkortin tulo on puskuroitu, ja se kykenee käsittelemään PWM-muotoista signaalia. Ohjainkortissa on runsaasti ominaisuuksia ja tuloja, joista tämän projektin kannalta olennaisia ovat nopeus, suunta ja vapaa.

Xiaomin käyttämän napamoottorin puutteellisen dokumentoinnin takia moottorinohjainta testattiin ensin ulkoisella virtalähteellä, ja nopeudensäätö tehtiin potentiometrillä. Ensin pyrittiin selvittämään Hall-antureiden ja käämien johtojen järjestys kokeilemalla eri kytkentäkombinaatioita ja mittaamalla moottorinohjaimen ottamaa virtaa. Hall-antureiden osalta oletettiin, että johtojen järjestys olisi pääteltävissä kuvan 4 perusteella. Asian varmistamiseksi testaus suoritettiin siten, että kaikki käämien ja Hall-antureiden johtokombinaatiot testattiin. Lopputulokseksi saatiin seuraavaa:

- Hall A: sininen

- Hall B: vihreä
- Hall C: keltainen
- käämi A: sininen
- käämi B: ruskea
- käämi C: keltainen.

Tämä tuntuu loogiselle järjestykselle ja testien perusteella kyseinen kombinaatio oli ainoa, jolla rengas pyöri tasaisesti eteenpäin. Muilla kombinaatioilla napamoottori ei pyörinyt ollenkaan, tai liike oli nykivää ja virrankulutus oli korkea. Tuloksia yritettiin verrata internetin harrastelijafoorumeilta löytyviin projekteihin ja lähteisiin, mutta vaikka vastaavan oloinen napamoottori vaikuttaisi olevan yleisesti käytössä eri valmistajilla, niin silti moottoreiden sisäisissä kytkennöissä käytettävien johtojen värit vaihtelevat.

Körmyyn vaatimaa ohjausta varten EM-347A-moottorinohjaimen asetuksia muutettiin oletuksista poikkeavaksi kokeilujen ja teknisten tietojen pohjalta seuraavasti:

- 1. Mode: 0 (Open loop)
- 7. Scale gain: 30
- 11. Operating mode: 0
- 16. I-trip and overcurrent indication: 1.

Tämän ohjainkortin dip1-kytkin asetettiin ON-asentoon, jolloin jarrutuksen aikana syntyvää virtaa ei kerätä talteen tai hävitetä kuormavastuksella lämmöksi tehokkaamman jarrutuksen aikaansaamiseksi.

Raspberry Pi:llä tehtiin yksinkertainen ohjauskoodi ja tämän perusteella EM-347A todettiin toimivaksi ja sopivaksi projektia varten. Ominaisuuksiensa puolesta EM-347A on kaikelta osin ylimitoitettu projektin tarpeisiin nähden, mutta kun 1 tällainen kortti oli jo tilattu, niin päädyttiin tilaamaan myös toinen samanlainen toiselle napamoottorille. Aluksi pohdittiin mahdollisuutta käyttää toista projektia varten hankittua ohjainkorttia EM-151B ja testien perusteella myös se todettiin toimivaksi. Testien aikana kuitenkin huomattiin, että

moottorinohjaimet tuottavat eri suuruisen kierrosnopeuden samalla PWM-pulssinleveydellä. Tämä tarkoittaisi, että käytettäessä poikkeavia moottorinohjainkortteja pitäisi kummallekin ohjainkortille kirjoittaa erillinen nopeussignaalin ohjausrutiini. Tämä todettiin projektin puitteissa turhan aikaa vieväksi.

Roottorilta tarvittavan paikkatiedon ja ohjauksen luoteen vuoksi BLDC-moottoreita ei voi käytännössä ohjata kuin yhtä moottoria moottorinohjainta kohden. Ainoastaan hyvin teoreettisessa tilanteessa voisi olla mahdollista, että moottoreiden roottorit pyörivät täysin samassa vaiheessa läpi käyttöajan.

## 4.2 Sähköjärjestelmä

### 4.2.1 Akusto ja sen testaus

Sähköjärjestelmän toteutukseksi oli useita eri vaihtoehtoja, joiden välillä käytiin pohdintaa. Vaihtoehtoja oli Xiaomin sähköpotkulaudan akustoon perustuva 36 V:n järjestelmä, 24 V:n järjestelmä ja näiden yhdistelmä. Erillisten jännitetasojen omaavien akkujen käyttäminen olisi projektin puitteissa ollut mahdollista. Näin napamoottoreille olisi ollut oma 36 V:n akusto, kääntömoottorille oma 24 V:n akusto ja mahdollisesti myös Raspberry Pi:lle oma sen vaatima 5 V:n akusto. Alkuvaiheessa ei ollut varmaa, että riittääkö 24 V:n jännite tuottamaan tarvittavan väännön Xiaomin napamoottoreille luotettavan toiminnan takaamiseksi. Tähän liittyen ei löytynyt tarpeeksi kattavasti dokumentoitua testaamista netin eri lähteistä, ja ymmärrettävistä syistä Xiaomin oma dokumentaatio aiheesta on varsin rajallista. Tämän takia ja lisäksi oman mielenkiinnon ja kustannuskysymysten vuoksi projektissa päädyttiin testaamaan Xiaomin omaa sähköpotkulaudasta irrotettua akkua ja sen soveltuvuutta projektin sähkölähteeksi.

Xiaomin akku on tyyppiä NE1002-H. Vaikuttaisi, että kyseisestä akusta on useita eri versioita jotakuinkin identtisillä ominaisuuksilla eri sähköskoottereiden, -polkupyörien ja muiden henkilökohtaisen liikkumisen välineiden valmistajilla. Kuvassa 5 näkyvän akun tyyppitarrassa ilmoitetaan ominaisjännitteeksi 36 V ja

sen kapasiteetti on 5100 mAh. Xiaomi ilmoittaa Essential-sähköpotkulaudan maksimikapasiteetin olevan 100 kg ja napamoottorin teholuokaksi ilmoitetaan 250 W hetkellisen tehon ollessa 500 W [19]. Tämän perusteella yhden akun antama teho riittää Kõrmyn vaatimuksiin tämän projektin puitteissa.



Kuva 5. Xiaomin sähköskootterin akku.

Xiaomin käyttämä akku on niin kutsuttu älyakku. Akkukennojen yhteyteen on liitetty ohjauspiiri, joka hallinnoi tehonsyöttöä, latausta ja muita akun turvatoimintoja. Ohjauspiirin toiminnasta löytyi hyvin vähän dokumentoitua tietoa ja alkuvaiheessa pidettiin myös mahdollisena, että akusto on jollain tavalla liitetty Xiaomin omaan moottorinohjauspiiriin ja näin ajonestoon. Akun toimintaa selvitettiin mitaamalla sen antama kuormaton jännite ja kuormittamalla sitä kuristamalla jännite 24 V:iin ja pyörittämällä Kõrmyn käänttömoottoria alhaisella nopeudella vapaasti ja liikettä jarruttamalla. Lisäksi akkuun tehtiin oma latausliitäntä ja lataamista kokeiltiin Xiaomin omalla akkulaturilla. Kuormitustestin ja latauksen aikana mitattiin yleismittarilla akun ottama ja antama virta normaalin toiminnan varmistamiseksi. Testien perusteella akun käyttöä ei ole valmistajan toimesta rajoitettu ja sitä voidaan käyttää Kõrmyn jännitelähteenä.

Testausmielessä päädyttiin kokeilemaan kahden akun käyttämistä rinnakkain maksimitehon ja -kapasiteetin kasvattamiseksi. Tämä todettiin pian toimimattomaksi johtuen akkujen poikkeavasta jännitetasosta. Jännitetaso poikkesi noin

500 mV myös täyteen ladattuna. Ongelman tässä aiheutti akkujen järjestelmäpiiri, joka mitä ilmeisimmin tulkitsi akkujen rinnankytkennän oikosuluksi ja näin laukaisi molempien akkujen suojapiirit. Suojapiirien laukeamista kokeiltiin kiertää myös tasaamalla molempien akkujen jännitetaso täysin samaksi, mutta tästäkään ei ollut apua. Tämän vuoksi päädyttiin toteuttamaan sähköjärjestelmä yhdellä akulla.

Jatkotestailun aikana tuli eteen satunnaisesti toistuva ongelma, jossa virtojen kytkemisen jälkeen akun suojapiiri laukesi. Syyn oletetaan johtuvan napamoottoreiden ohjauskortin suurikapasiitivisten kondensaattoreiden täyttymisestä, joka rajoittamattomana saa aikaan akun ylivirtasuojauksen laukeamisen. Korjaukseksi tähän päätettiin kokeilla laajemmin napamoottoreiden toimivuutta 24 V:n jännitteellä. Testin perusteella napamoottorit vaikuttavat toimivan projektin kannalta riittävällä tavalla myös 36 V:n ominaisjännitettä alemmalla jännitteellä. Oletus oli, että jännitteenalennin toimisi myös virranrajoittajana virtojen kytkemisen jälkeen ja tätä testattiin seuraavaksi. Havainto oli, että oletus oli oikea ja satunnaisesta akun vikapiirin laukeamisesta päästiin eroon. Ratkaisu ei toki ole optimaalinen, mutta hyväksyttävä projektille määritellyssä ikkunassa.

#### 4.2.2 Jännitetasojen määrittely ja toteutus

Körmyin eri komponenttien jännitetasoiksi muodostui lopulta siis 24 V ja 5 V. Jännitetasot luotiin alentamalla akun 36 V:n jännite näihin tasoihin erillisillä jännitteenalenninpiireillä. Molempien jännitetasojen alentamiseen käytettiin yleisesti eri elektroniikkaa myyviltä tahoilta löytyviä piirejä. Projektissa käytettävät jännitteenalentimet päädyttiin hankkimaan kotimaiselta Partcolta.

Jännite alennettiin 24 V:n tasoon säädettävällä jännitteenalentimella, joka pohjautuu TI:n LM25116 synkronoituun hakkuripiiriin. Sen ominaisuuksiksi on lisätty myyjäliikkeen toimesta 300 W:n tehonkesto ja 20 A:n virrankesto. Sama jännitteenalennin vaikuttaisi olevan myynnissä samoilla ominaisuuksilla useissa eri paikoissa, joten ominaisuuksia voidaan tämän pohjalta pitää ainekin jokseenkin paikkaansa pitävänä. Koska Körmyllä ei vielä tässä vaiheessa ole

tarkoitus liikutella suuria kuormia, niin napamoottoreiden kuormitustestaukseen ja Xiaomin sähköpotkulaudan teknisiin tietoihin pohjautuen arvioitiin, että yksi jännitteenalennin riittää molemmille vetäville pyörille. Kääntömoottorille tulee oma vastaava jännitteenalennin.

Raspberry Pi suosittelee, että Pi 4:n jännitelähde kykenisi tuottamaan vähintään 3 A:n virran [17, s. 3]. Körmyn ohjauskoodi ei ole kovin vaativa ja näin Pi 4:n CPU:n rasitus tulee olemaan varsin maltillinen. Lisäksi käytön aikana ei ole tarvetta ulkoisille liittimille, kuten USB-porteille ja HDMI:lle tai muille virtaa käyttäville ulkoisille liittimille. Tähän pohjautuen päädyttiin saatavilla olevista vaihtoehdoista TI:n LM2596-regulaattoriin pohjautuvaan säädettävään jännitelähteeseen, joka kykenee tuottamaan maksimissaan 3 A:n virran.

Molempien jännitteenalentimien toimintaa testattiin niiden toiminnan varmistamiseksi mittaamalla niiden antama jännite eri kuormilla. Testin perusteella todettiin, että jännitteenalentimien antama lähtöjännite ei juurikaan tipu testatulla ja odotettavissa olevalla kuormalla. 5 V:n jännitteenalentimen kela pitää jonkin verran sirinää, mikä on edulliselle osalle melko odotettava ilmiö, kun jännite alennetaan yli 30 V lähtötasostaan. Komponenttien lämpötilaa mitattiin erillisellä yleismittarilla ja siihen kytketyllä lämpöanturilla. 24 V:n piiristä mitattiin hakkuriin ja FETien lämpötila ja 5 V:n piiristä mitattiin regulaattorin ja kelan lämpötila. Testauksen aikana ei havaittu kummankaan jännitteenalentimen komponenttien lämpenevän edes lähelle niille määritellyn toimintalämpötilan ylärajaa. Mittauksia ei standardoitu ja mittaustuloksia ei erikseen dokumentoitu. Testien tarkoituksena oli saada suuntaa antava tulos piirien toiminnasta ja soveltuvuudesta projektiin, eikä niinkään tarkkaa tietoa niiden toiminnasta eri kuormitusasteilla ja olosuhteissa.

Pääsähköjärjestelmän kaapelointi toteutettiin riviliittimen kautta, josta akun jännite jaetaan jännitteenalentimille. Riviliittimen asennuskiskoon sijoitettiin myös 20 A:n pääsulake, jolle suunniteltiin ja tulostettiin oma kiinnike.

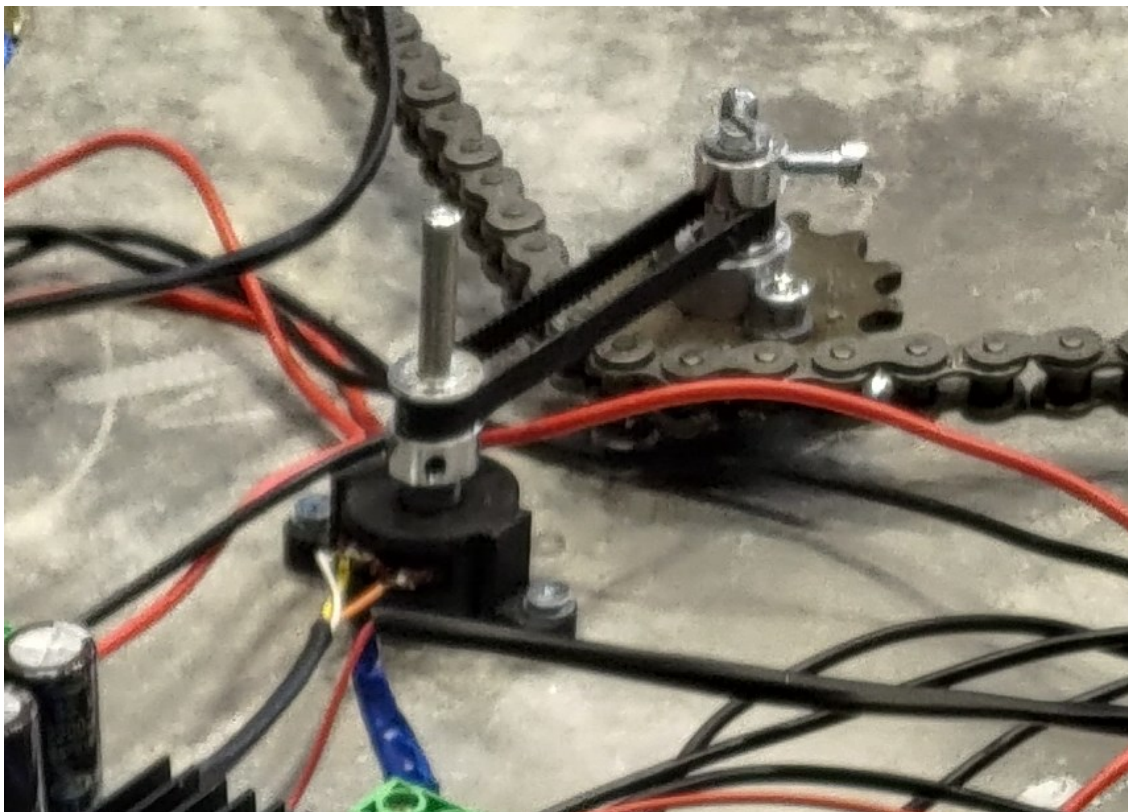
### 4.2.3 Elektroniikan signaalien kytkentä

Raspberry Pi 4:n GPIO-liitäntöjen jännitetaso on 3,3 V. Moottorinohjaimet vaativat 5 V:n signaalin ja signaalitason korotus 3,3 V:sta tehtiin erillisellä logiikkatasomuuntimella. Kääntömoottorin vaatimaan DC-muotoisen ohjaussignaalin tuottamiseen käytetään DAC-lisäkorttia, jonka lähtötaso on ohjelmallisesti säädettävissä välillä 0–3,3 V ja tämä on kytketty edelleen logiikkatasonmuuntimelle.

Kaikille kolmelle moottorinohjaimelle tulee 3 erillistä signaalia, joista kaksi tulee Raspberry Pi:ltä ja yksi tulee fyysiseltä kytkimeltä. Fyysiseltä kytkimeltä tuleva signaali on kytketty napamoottoreiden ohjainkorttien Freerun-tuloon ja kääntömoottorin ohjainkortin Disable-tuloon. Syy tämän signaalin käytölle fyysisellä kytkimellä johtuu Raspberry Pi 4:n käynnistyssekvenssistä, joka tuottaa GPIO-liittimille satunnaisen arvon, kunnes liittinten tila nollataan ohjelmallisesti. Fyysisellä kytkimellä varmistetaan, että moottorinohjauskortit vastaavat ohjaussignaaliin vasta, kun kytkimellä kytketään estosignaali pois päältä.

Asiakkaan toiveesta renkaiden kääntäminen haluttiin toteuttaa siten, että se seuraa ohjaimen joystickin asentoa. Tämän toteuttamiseksi kääntömekanismiin lisättiin erillinen potentiometri, joka on yhdistetty kääntömoottorin akseliin hihnalla. Toteutus on nähtävissä kuvassa 6. Näin renkaiden asento välittyy potentiometrille, jonka jännitteen perusteella voidaan tulkita renkaiden kääntökulma.

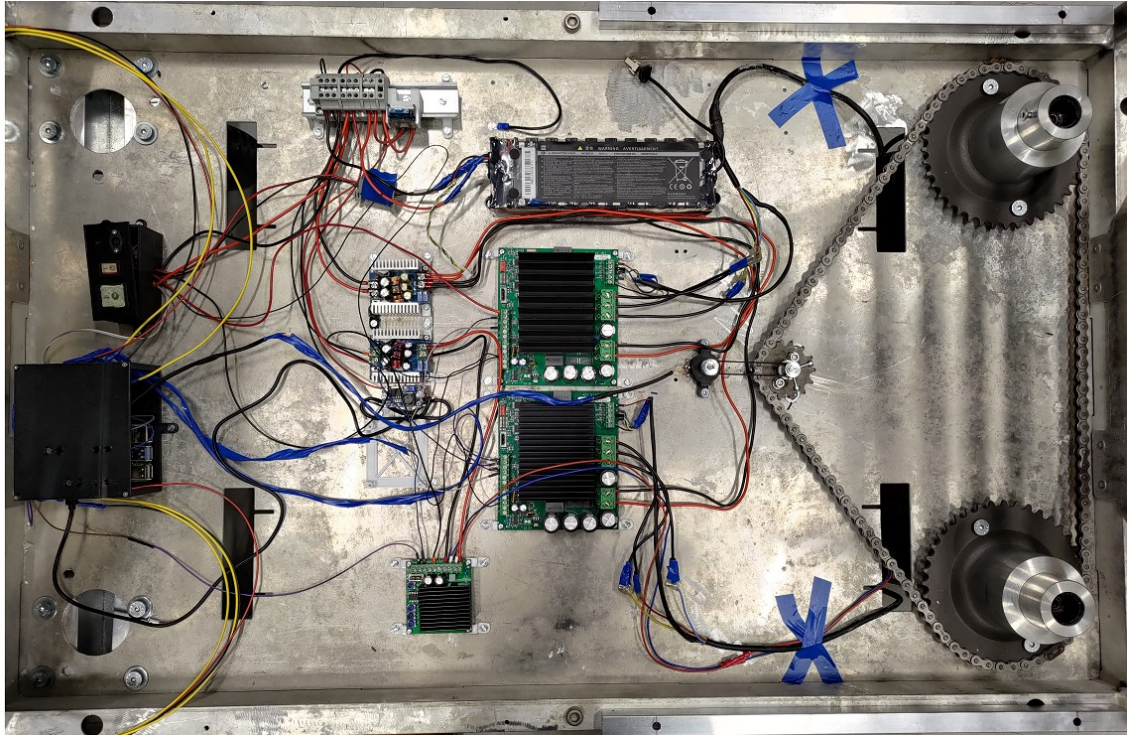
Potentiometrin liikerata rajoittaa jonkin verran renkaiden kääntökulmaa ja näin Kõrmyn kääntösädetä, mutta projektin puitteissa se todettiin riittäväksi. Potentiometrille syötetään DAC-kortilta 3,3 V:n jännite ja maa ja jännitetason mittaamiseen käytetään DAC-kortin tuloa IN1.



Kuva 6. Kääntömootorin akseliin hihnalla yhdistetty potentiometri.

Lähestymisantureina toimiville HC-SR04-ultraääniantureille tulee 3,3 V käyttöjännitteen ja maan lisäksi oma signaali pulssille ja kaiulle. Pulssi ja kaiu on kytketty suoraan Raspberry Pi:n GPIO-portteihin. Anturin käyttöjännitteeksi on ilmoitettu 5 V, mutta testien perusteella se toimii luotettavasti myös tätä alhaisemmalla 3,3 V:n jännitteellä. Ratkaisu ei ole optimaalinen, mutta aikataulullisista syistä ei ollut mahdollista tilata enempää logiikkatasomuuntimia, joten se päätettiin toteuttaa. Sähkökytkennän ja komponenttien sijoittelun lopputulos näkyy kuvassa 7.





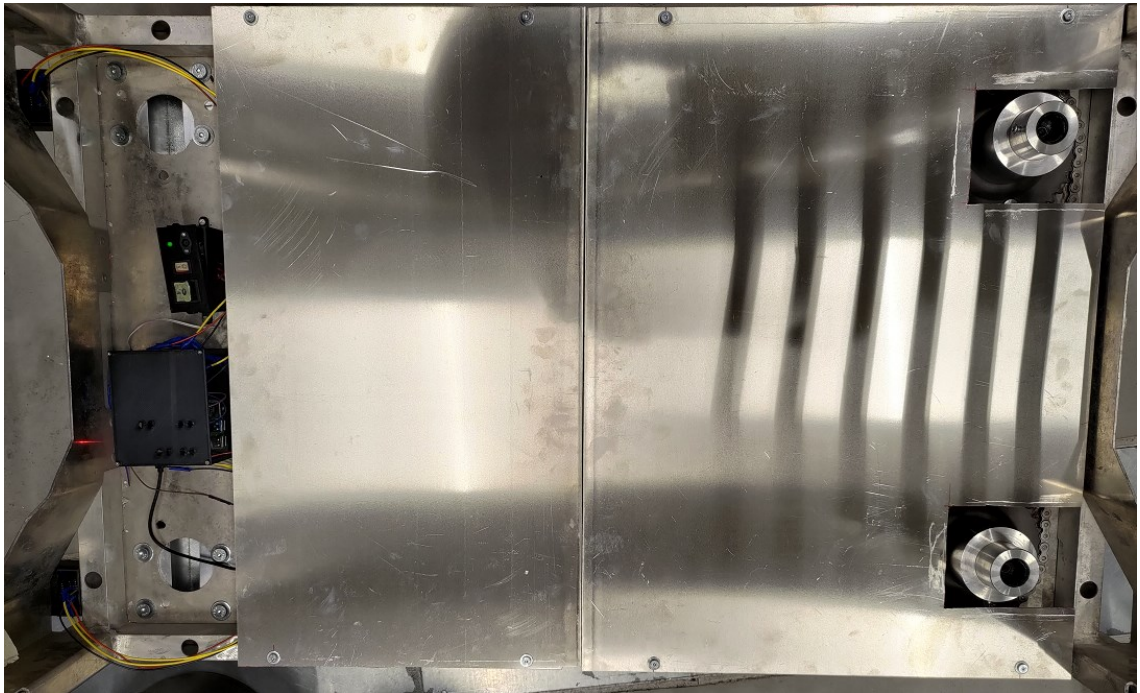
Kuva 7. Sähkökytkennät ja komponentit paikallaan.

#### 4.2.4 Komponenttien asennus

Komponenttien asennukseen ja kotelointiin pohdittiin erilaisia vaihtoehtoja kaupallisesti saatavilla olevista laiterasioista aina itse tehtyihin koteloihin. Lopulta aikataulun puitteissa päädyttiin yksinkertaisesti kiinnittämään komponentit ruuvaamalla Körmyin pohjalevyyn ja suojaamalla komponentit koko pohjalevyn peittävällä alumiinilevyllä. Komponenttien kiinnitystä varten suunniteltiin Solidwork-silla kiinnitysalustat, jotka tulostettiin 3D-tulostimella. 24 V:n jännitteenalentimille ja moottorinohjaimille käytettiin samaa yleiskäyttöistä kiinnityskappaletta. Raspberry Pi:lle, päävirtakytkimelle ja ultraäänisensoreille tehtiin omat kotelot.

Päävirtakytkimen kanssa samaan koteloon laitettiin myös moottorinohjainten estokytkin, latausliitin ja indikaattorina toimiva LED. Raspberry Pi:n ja päävirtakytkimen kotelo on sijoitettu muut komponentit peittävän alumiinisen levyn ulkopuolelle, jotta niihin pääse tarvittaessa helpommin käsiksi. Levy on kiinnitetty ja tuettu vain reunoiltaan, joten se ei kestä minkäänlaista painoa. Levyn ainoa

tarkoitus on suojata komponentteja. Lopullinen asennus koteloinnin kanssa näkyy kuvassa 8.



Kuva 8. Lopullinen kotelointi.

### 4.3 Kõrmyñ aivot

#### 4.3.1 Raspberry Pi 4:n käyttöönotto ja vaadittavat asennukset ja kirjastot

Projektissa päädyttiin käyttämään Raspberry Pi:n omaa Raspberry OS -käyttöjärjestelmää. Graafisen käyttöliittymän lataaminen ei nopeutensa puolesta ole välttämättä se optimaalisin ratkaisu, mutta projektin konseptiluonne huomioiden se ajaa asiansa.

Projektin aluksi Raspberry OS asennettiin alusta alkaen uudelleen ja päivitettiin kaikkien käyttöjärjestelmän komponenttien osalta ajan tasalle. Tällä haluttiin varmistaa, että käyttöjärjestelmäjärjestelmä on puhdas mahdollisista aiemmista projekteista ja niiden yhteydessä tehdyistä määrittelyistä. Päivityksellä myös haluttiin varmistaa, että käytössä on viimeisimmät ajurit esimerkiksi Bluetooth-yhteyttä varten. Käyttöjärjestelmään päivitettiin uudelleen asennuksen jälkeen

ensin kaikki graafisen käyttöliittymän ehdottamat päivitykset ja tämän jälkeen käytettiin vielä kuvassa 9 näkyviä komentoja komentotulkissa päivityksen tekemiseksi.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Kuva 9. Komentotulkin komennut Raspberry OS:n päivittämiseksi.

Python päivitettiin versioon 3.10. Projektin koodissa käytettävät komennot toimivat yleisesti Pythonin versiolla 3, joten version 3.10 tuoma tuki komennoille kuten match ei ole välttämättömyys. Päivitys tapahtui kuvassa 10 näkyvillä komennoilla komentotulkissa.

```
$ wget https://www.python.org/ftp/python/3.10.2/Python-3.10.2.tgz
$ tar -zxvf Python-3.10.2.tgz
$ cd Python-3.10.2
$ ./configure --enable-optimizations
$ sudo apt install -y build-essential tk-dev libncurses5-dev libncursesw5-dev libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev libffi-dev
$ sudo make altinstall
$ cd /usr/bin
$ sudo rm python3
$ sudo ln -s /usr/local/bin/python3.10 python3
```

Kuva 10. Python 3.10. asennus ja asetus käyttöjärjestelmän oletukseksi.

Projektia varten tarvittiin Raspberry Pi:n ja Pythonin omien ohjelmakirjastojen lisäksi oma kirjasto ADC-DAC-muuntimelle ja DUALSHOCK@4-ohjaimelle. Molemmille kirjastoille löytyy kattava dokumentointi kirjaston käytöstä niiden GitHub-projektisivuilta:

- Ohjain: <https://github.com/ArturSpirin/pyPS4Controller>
- ADC-DAC-muunnin: [https://github.com/abelectronicsuk/ABElectronics\\_C\\_Libraries/tree/master/ADCDACPi](https://github.com/abelectronicsuk/ABElectronics_C_Libraries/tree/master/ADCDACPi).

DUALSHOCK@4-ohjaimen ohjelma kirjaston asennus onnistuu kuvassa 11 näkyvällä komennolla:

```
sudo pip install pyPS4Controller
```

Kuva 11. Komentotulkin komento pyPS4Controller-ohjelmakirjaston asennukselle.

ADC-DAC-lisäkortin ohjelmakirjaston asentaminen onnistuu kuvan 12 komennolla:

```
sudo python -m pip install git+https://github.com/abelectronicsuk/ABElectronics_Python_Libraries.git
```

Kuva 12. Komentotulkin komento DAC-lisäkortin ohjelmakirjaston asennukselle.

Raspberry OS asetettiin käynnistämään Python-koodi automaattisesti aina, kun Körmyyn kytketään virrat pääkatkaisijasta. Tämä tehtiin muokkaamalla .bashrc-tiedostoa. Linuxin .bashrc-tiedostoon määritellyt komennot ajetaan automaattisesti, kun käyttäjä kirjautuu käyttöjärjestelmään sisälle. Kormyn tapauksessa kirjautuminen käyttöjärjestelmään tapahtuu automaattisesti. Tiedoston loppuun lisättiin kuvassa 12 näkyvä komento.

```
$ sudo python /home/Desktop/kormy/kehitysfilly.py
```

Kuva 13. .bashrc-tiedoston loppuun lisätty komento ohjauskoodin automaattiseksi käynnistämiseksi.

DUALSHOCK@4-ohjain lisättiin käyttöjärjestelmän luotettujen laitteiden listalle ja näin Bluetooth-yhteyden muodostaminen tapahtuu automaattisesti, kun ohjaimen kytketään virta päälle.

#### 4.3.2 Ohjausohjelmisto ja Python-koodi avattuna

Ohjauskoodin kehittäminen aloitettiin tutustumalla pyPS4Controller-ohjelmakirjaston dokumentointiin ja esimerkkeihin. Karkeasti kirjasto toimii siten, että

pääohjelmassa kirjaston alustamisen jälkeen se toimii luupilla. Ohjaimen tilan muutos saa aikaan keskeytyksen, joka oletuksena palauttaa keskeytyksen arvon tekstinä. Yksilöllisen koodin ajaminen keskeytyksen perusteella onnistuu luomalla luokka, joka perii pyPS4Controller-kirjaston Controller-luokan ja määrittelemällä tälle jäsenmuuttajat dokumentoinnin perusteella.

Kirjaston toimintaa testattiin ensin pelkän Pythonin tulkin avulla. Testailun tarkoituksena oli määrittellä ohjaimen analogisten toimintojen arvot ja tarvittavien keskeytysten jäsenmuuttujien toiminta. Tämän pohjalta saatiin määriteltyä mm. seuraavaa ohjaimen analogisten joystickien kokonaislukua vastaaviksi arvoiksi:

- vasen: pienin -337 ja ääriasento -32767
- oikea: pienin 337 ja ääriasento 32767
- ylös: pienin -337 ja ääriasento -32767
- alas: pienin 337 ja ääriasento 32767.

Kummassakin tapauksessa nähdään erillisten suuntien saavan joko negatiivisen tai positiivisen kokonaislukuarvon. Tämän perustella voidaan ohjauskoodissa määrittellä joystickin asennosta riippuva nopeussignaali ja suunta. Nopeussignaali määrittellään PWM-pulssijonon suhteella, jonka yksikkönä on prosenttia ilmaiseva kokonaisluku välillä 0–100. Nopeuden maksimiksi määriteltiin lukuarvo 50, joka kokeilun perusteella tuntuu järkevälle Kõrmyn maksiminopeudelle yhdessä luvussa 4.1.3. tehtyjen asetusten kanssa. Kõrmyn liikuttaminen vaatii aina sen, että R1-nappi on painettuna pohjaan. Tämä toimii ikään kuin kuolleenmiehenkytkimenä. Esimerkkikoodista 1 näkyy typistettynä esimerkki ohjaussignaalin luomisesta napamoottoreiden pyörittämiseksi eteen ja taakse. Esimerkkiin on koodin rakenteen vuoksi jätetty vain toiminnon esittämisen kannalta olennaiset kohdat.

```

class MyController(Controller):

    def __init__(self, **kwargs):
        Controller.__init__(self, **kwargs)

    #Controller.on_R3_up(self, value) #-337 pienin, -32767 suurin
    def on_R3_up(self, value):
        global DM_Switch #kuolleenmiehenkytkin
        global max_speed #maksiminopeus
        global l2_value #etäisyysmittauksen ohitus
        if DM_Switch == True:
            if l2_value == 32767:
                max_speed = 50
            if (value / -32767 - max_speed / 100) <= 0:
                R3_up_DutyCycleValue = int(value / -32767 * 100)
            else:
                R3_up_DutyCycleValue = int(max_speed)
            #suunta eteen päin
            GPIO.output(18, GPIO.LOW)
            #moottorinohjaimelle lähtevä signaali
            motor.ChangeDutyCycle(R3_up_DutyCycleValue)
        else:
            print("DM_Switch = False")

    #Controller.on_R3_down(self, value) #337 pienin, 32767 suurin
    def on_R3_down(self, value):
        global DM_Switch #kuolleenmiehenkytkin
        if DM_Switch == True:
            R3_down_DutyCycleValue = int(value / 32767 * 50)
            #suunta taakse päin
            GPIO.output(18, GPIO.HIGH)
            #moottorinohjaimelle lähtevä signaali
            motor.ChangeDutyCycle(R3_down_DutyCycleValue)
        else:
            print("DM_Switch = False")

```

### Esimerkkikoodi 1. Napamoottoreiden ohjaus.

Körmyrren renkaiden kääntämisen toiminto toteutettiin asiakkaan pyynnön pohjalta siten, että ohjaukskulma on riippuvainen ohjaimen joystickin asennosta. Näin esimerkiksi joystickin vapauttaminen saa renkaiden ohjaukskulman palautumaan oletusasentoon, joka on suoraan eteenpäin. Tämän toteuttamiseksi hyödynnetään ohjausjärjestelmään kiinnitetyn potentiometrin antamaa jännitesignaalia, joka muunnetaan digitaaliseksi DAC-kortilla. Kortin muunnin on 12-bittinen, joten se kykenee jakamaan 3,3 V:n jännitteen 4096 osaan. Näin ääriarvoilla 0 ja 4095 renkaat ovat kääntyneet jompaankumpaan ääriasentoon, ja arvolla 2048 renkaat osoittavat suoraan eteenpäin.

Kääntömoottorin liikenopeus on toteutustavan johdosta vakio. Sopivaa kääntönopeutta testattiin kokeilemalla ohjauksen toimintaa käytännössä Körmyä

liikuttamalla. Kokeilun perusteella tehtiin havainto, että kääntömoottori ei kykene asettumaan haluttuun asentoon, mikä johtuu komponenttien epätarkkuuksista ja ohjausjärjestelmän mekaaniseen rakenteen aiheuttamista viiveistä. Ohjaus ajautui jatkuvasti halutun asennon yli, ja näin ohjausmoottori jäi ikään kuin oskiloimaan halutun asennon ympärille sitä saavuttamatta. Korjaukseksi koodiin asetettiin toleranssirajat halutulle asennolle ja lisäksi määriteltiin normaalia liikenopeutta hitaampi lähestymisnopeus, jotta ohjausjärjestelmä saavuttaisi halutun ohjauskulman luotettavammin. Tämän pohjalta testausta jatkettiin sopivan kääntömoottorin nopeuden määrittämiseksi. Ohjausjärjestelmän nopeuden määrittämiseen vaikutti se, kuinka nopeasti renkaiden halutaan kääntyvään haluttuun ohjauskulmaan, ja lisäksi se, kuinka nopeasti ohjauskulman halutaan vakioituvan.

Testailun aikana havaittiin, että ohjaussignaalin koodi toimii toivotulla tavalla, kun samaan aikaan ei tuoteta ohjaussignaalia eteenpäin, tai taaksepäin olevalle liikkeelle. Kun molempia signaaleja halutaan tuottaa samaan aikaan, niin havaittiin koodin hidastuvan merkittävästi ja aiheuttavan satunnaisesti koko ohjausjärjestelmän jumiutumisen ja esimerkiksi renkaiden kääntyvän yli asetettujen rajojen, tai napamoottoreiden ohjaussignaalin jäävän päälle usean sekunnin ajaksi sen jälkeen, kun joystick on palautettu nolla-asentoon. Syyksi tähän epäiltiin molempien ohjaussignaalien ja pyPS4Controller-kirjaston luuppien rasittavan ohjelmistotulkin työjonoa siten, että se ei kyennyt ratkaisemaan työjonoa halutulla tavalla. Ratkaisuksi ohjaussignaali päätettiin toteuttaa säikeistämällä siten, että kääntömoottorin ohjaussignaali erotetaan omaksi säikeekseen ja napamoottoreiden ohjaussignaali omaksi säikeeksi. Tämä tarkoitti, että kääntömoottorin ohjaussignaali piti irrottaa omaksi funktioksi irti aiemmin esitetystä MyController-luokasta. Tieto joystickin asennosta tuodaan ohjaussignaalin funktiolle käyttämällä globaaleja muuttujia.

Seuraavalla sivulla on esitettyä esimerkikoodi 2, jossa näkyy ohjaussignaalin tuottamiseen käytettävä koodi. Muuttujat A ja B tuodaan MyController-luokasta. Niiden perusteella koodi määrittelee kääntömoottorin suunnan ja asennon.

```

def handler():
    global a #kääntö oikealle
    global b #kääntö vasemmalle
    global c #triggeri, renkaat on suorassa
    if c == 1:
        CurrentTurnPosition = adcdac.read_adc_raw(1, 0)
        target = 2048
        if CurrentTurnPosition < 2038:
            GPIO.output(11, GPIO.HIGH)
            adcdac.set_dac_raw(1, int(4095 * 0.3))
            if CurrentTurnPosition < 2048 and CurrentTurnPosition >
1600:
                adcdac.set_dac_raw(1, int(4095 * 0.075))
                print("Current position:", CurrentTurnPosition, "Turning
left")
        elif CurrentTurnPosition > 2058:
            GPIO.output(11, GPIO.LOW)
            adcdac.set_dac_raw(1, int(4095 * 0.4))
            if CurrentTurnPosition < 2600 and CurrentTurnPosition >
2058 :
                adcdac.set_dac_raw(1, int(4095 * 0.075))
                print("Current position:", CurrentTurnPosition, "Turning
right")
        else:
            adcdac.set_dac_raw(1, 0) # DACin nollaus
            GPIO.output(11, GPIO.LOW)
            c = 0
    else:
        CurrentTurnPosition = adcdac.read_adc_raw(1, 0)
        target = int(2048 - (b - a) / 32767 * 2048)

        if target != 2048:
            if CurrentTurnPosition > target:
                GPIO.output(11, GPIO.LOW)
                if CurrentTurnPosition > target + 500:
                    adcdac.set_dac_raw(1, int(4095 * 0.3))
                elif CurrentTurnPosition > target + 40:
                    adcdac.set_dac_raw(1, int(4095 * 0.075))
                else:
                    adcdac.set_dac_raw(1, 0)
            else:
                GPIO.output(11, GPIO.HIGH)
                if CurrentTurnPosition < target - 500:
                    adcdac.set_dac_raw(1, int(4095 * 0.3))
                elif CurrentTurnPosition < target - 40:
                    adcdac.set_dac_raw(1, int(4095 * 0.075))
                else:
                    adcdac.set_dac_raw(1, 0)

```

## Esimerkkikoodi 2. Kääntömoottorin ohjaus.

Etäisyysantureiden signaalin perusteella määritellään Kõrmyn nopeus lähestyt-  
täessä estettä. Etäisyysantureiden ohjaus vaatii oman ohjelmaluupin ja kääntö-  
moottorin ohjaussignaalin toteutuksen kokemusten perusteella etäisyysmittaus  
päättiin toteuttaa omana säikeenään. Etäisyyden mittaus tapahtuu



ultraäänellä siten, että ultraäänianturilla tuotetaan ensin lyhyt pulssi ja tämän jälkeen jäädään kuuntelemaan pulssin palautumista. Lähteyksen ja vastaanoton välinen aika lasketaan ja kerrotaan äänennopeudella, joka on 343 metriä sekunnissa. Tulo jaetaan edelleen kahdella, koska äänipulssi kulkee mitatun matkan kahteen kertaan. Kahden anturin mittausta suoritetaan 0,2 sekunnin välein ja pienempää mittaustulosta käytetään maksiminopeuden määrittelyssä. 0,2 sekunnin välein tapahtuva etäisyyden mittausta on riittävä huomioiden Kөрmylle kokeilemalla määritelty maksiminopeus. Mittausluupille on määritelty myös erillinen maksimiaika, joka on 0,02 sekuntia. Matkana tämä tarkoittaa noin 3,4 metrin etäisyyttä. Lähestymisanturin tarkoituksenmukaiselle toiminnalle tällä etäisyydellä on merkitystä ja pidemmän ajan mittaaminen olisi siksi tarpeetonta.

Etäisyyden mittauksen koodi on jaettu kahteen erilliseen funktioon, joista toinen ohjaa mittauksen ajoitusta sekä määrittelee mittauksen perusteella Kөрmy maksiminopeuden ja toinen suorittaa etäisyyden mittauksen. Maksiminopeus tallennetaan globaaliin muuttujaan, jota käytetään nopeudenohjaussignaalin luomisessa. Etäisyyden mittausta toimii varsin luotettavasti, ja sen tarkkuus on riittävä projektin tarpeisiin nähden.

Kөрmy edessä olevan kohteen etäisyyden ollessa alle 10 cm ei Kөрmy voi liikkua kuin taakse päin. Toiminnon ohittamiseksi lisättiin koodiin ominaisuus, joka mahdollistaa rajoitteen kiertämisen painamalla L2-nappi pohjaan.

Esimerkkikoodi 3 näyttää etäisyyden mittaukseen ja sen pohjalta tehtävään maksiminopeuden määrittelyyn liittyvän ohjelmakoodin.

```

def distance(triggeri, kuuntelija):
    GPIO.output(triggeri, True) #pulssin lähetys
    time.sleep(0.00001) #pulssin pituus
    GPIO.output(triggeri, False)

    # StartTime
    while GPIO.input(kuuntelija) == 0:
        StartTime = time.time()
        EndTime = time.time() + 0.02

    while GPIO.input(kuuntelija) == 1:
        StopTime = time.time()
        try:
            if time.time() > EndTime:
                StopTime = 0
                break
        except Exception:
            StopTime = 0

    try:
        TimeElapsed = StopTime - StartTime
    except Exception:
        TimeElapsed = 0
    etaisyyss = (TimeElapsed * 34300) / 2

    return etaisyyss

def Measurement_Handler():
    global max_speed
    global Distance_Override
    while True:
        dist1 = distance(GPIO_TRIGGER_1, GPIO_ECHO_1)
        if dist1 < 0:
            dist1 = 999
        dist2 = distance(GPIO_TRIGGER_2, GPIO_ECHO_2)
        if dist2 < 0:
            dist2 = 999
        print ("1 = %.1f" % dist1, "2 = %.1f cm" % dist2)
        min_distance = min(dist1, dist2)

        if min_distance < 10:
            if Distance_Override == 1:
                max_speed = 1
            else:
                max_speed = 0
                R3_up_DutyCycleValue = 0
        elif min_distance < 30:
            if Distance_Override == 1:
                max_speed = 1
            else:
                max_speed = 0
        elif min_distance < 60:
            max_speed = 1
        elif min_distance < 100:
            max_speed = 50
        time.sleep(0.2)

```

**Esimerkkikoodi 3. Etäisyyden mittauksen ja maksiminopeuden määrittelyn koodi.**

Ohjaukoodiin lisättiin myös viimeiseksi riviksi erillinen signaali, joka saa arvon HIGH, kun ohjelmakoodi pääsee kyseiselle riville asti. Tämän signaalin on tarkoitus kytkeä päävirtakytkimen kotelossa oleva indikaattoriledi päälle, joka toimii signaalina siitä, että käyttöjärjestelmä ja koodi on ladattuna se odottaa ohjaimen Bluetooth-yhteyden muodostamista. Yhteyden muodostamisen jälkeen mootto-reiden ohjauskorttien vapaa voidaan kytkeä pois päältä, ja näin Körmy reagoi ohjaimen komentoihin.

## 5 Yhteenveto

Kokonaisuutena projekti todettiin onnistuneeksi asiakkaan ja Big Flash -projektin projektipäällikön toimesta. Körmy saatiin liikkumaan ja tottelemaan langatonta ohjainta, renkaiden kääntäminen saatiin toteutettua asiakkaan toivomalla tavalla ja lisäksi loppuvaiheessa projektiin lisätty ultraääneen perustuva lähestymisanturi saatiin toimimaan. Kokonaisuus tehtiin käyttämällä pelkästään kaupallisesti saatavilla olevia osia, ja näin kustannukset saatiin pidettyä hallinnassa. Projekti oli kokonaisuutena mielenkiintoinen, ja se tarjosi sopivan haasteen suhteutettuna projektille asetettuun aikaikkunaan.

Haasteena tämänkaltaisessa projektissa, jossa luodaan jotakuinkin alusta alkaen jotain uutta, on aina aikataulu. Projektin alussa visioitu ja suunniteltu aikataulu havaittiin paikkaansa pitämättömäksi jo hyvin pian. Valtaosa ajasta oli tarkoitus käyttää ohjaukoodin hienosäätämiseen ja Körmyn toiminnan ja kykyjen testaamiseen kuorman kanssa. Projektin aikana jouduttiin kuitenkin tekemään jonkin verran mekaanisia töitä Körmyn rakenteellisen toiminnan varmistamiseksi ja lisäksi joidenkin komponenttien ominaisuudet aiheuttivat suunnittelematonta ajankäyttöä. Näiden seikkojen vuoksi ohjelmakoodin, johdotuksen ja mekaanisten toiminnallisuuksien luotettavuus ja varmuus eivät välttämättä ole parhaalla mahdollisella tasolla.

Projektissa opittiin, että uudenlaisten visioiden pohjalta voidaan lähteä kehittämään ja luomaan jotain uutta ja poikkeuksellista. Uudenlaisia robotiikan käyttökohteita ja mahdollisuuksia visioimaan ja toteuttamaan uskaltavat voivat olla

hyvissä lähtöasetelmissä, kun robotisaatio jatkaa yleistymistään ja arkipäiväistymistään kaikilla elämän ja teollisuuden osa-alueilla.

Körmyyn toiminnot toteutettiin projektin luonteen mukaisesti konseptitasoisesti, joten tuotantovalmiista tuotteesta ei siis ole kyse. Körmyyn mekaanisen rakenteen ja ohjaukoodin kehittäminen sellaiselle tasolle, että robottia voisi käyttää prototyypitasoisesti asiakkaan rakennuskohteissa vaatisi runsaasti aikaa ja resursseja. Se olisi kuitenkin tehtävissä tämän projektin pohjalta opittuja asioita hyödyntämällä.

Projektissa tehtyjen havaintojen pohjalta ehdotus mahdollisen jatkokehityksen osalta olisi seuraavien asioiden huomioiminen:

- Vetävien pyörien napamoottoriksi voisi olla parempi valita sellainen malli, jossa on sisäänrakennettuna välitystä pienentävä ratastus ja mahdollisuus mekaanisen jarrun toteutukselle.
- DUALSHOCK®4-ohjain ajaa asiansa varmasti prototyypitasolle asti, mutta rakennusmaailmaan sopivampi ja kantamaltaan parempi ohjain kannattaa ottaa harkintaan.
- Körmyyn mekaanisen rakenteen suunnittelu ja toteutus tukevammaksi. Tämänhetkinen mekaaninen rakenne ei sovellu kovin suurelle kuormalle.
- Pohtiminen, voisiko Körmyyn pohjana hyödyntää jotain olemassa olevaa alustaa ja kehittää tämän pohjalta asiakkaan toivomia ominaisuuksia. Esimerkki tällaisesta alustasta on ErgoMover [11].

Laajempi älytoimintojen, kuten autonomisen suunnistamisen, tai työntekijän automaattisen seuraamisen tuominen osaksi Körmyä olisi kokonaisuutena oma projektinsa. Älytoimintojen osalta saattaisi olla kannattavaa pohtia kaupallisesti saatavien teollisuusmobiilirobottien hyödyntämistä Körmyyn pohjaksi. Pääsääntöisesti teollisuudessa käytettävät mobiilirobotit on suunniteltu varastoihin ja vastaaviin ympäristöihin, mutta teknologian kehittyessä uskoisin, että jonkinlainen olemassa oleva tai konseptitasolla oleva ratkaisu voisi toimia Körmyyn kaltaisen rakennusmaailmaan tarkoitetun robotin pohjana.

## Lähteet

- 1 Matarić, Maja J. 2007. The Robotics Primer. London: The MIT Press.
- 2 Standardization. 2023. Verkkoaineisto. IFR. <<https://ifr.org/standardisation>>. Luettu 16.4.2023.
- 3 ISO 8373:2021(en) Robotics — Vocabulary. 2021. Verkkoaineisto. ISO. <<https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-3:v1:en>>. Luettu 16.4.2023.
- 4 Grey Walter and his tortoises. 2008. Verkkoaineisto. University of Bristol. <<http://www.bristol.ac.uk/news/2008/212017945378.html>>. Luettu 7.5.2023.
- 5 Long, Jing; Zhang Chun Liang. 2012. The Summary of AGV Guidance Technology. Verkkoaineisto. Scientific.Net. <<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c47e00e1ce9b5849907208e794da9b1ed8d88af0>>. Luettu 16.4.2023.
- 6 Autonomous Mobile Robots. 2023. Verkkoaineisto. Intel. <<https://www.intel.com/content/www/us/en/robotics/autonomous-mobile-robots/overview.html>>. Luettu 18.4.2023.
- 7 Chen, X.Q.; Chen Y.Q.; Chase, J.G. 2009. Mobile Robots – Past Present and Future. Verkkoaineisto. ResearchGate. <[https://www.researchgate.net/publication/221786528\\_Mobiles\\_Robots\\_-\\_Past\\_Present\\_and\\_Future](https://www.researchgate.net/publication/221786528_Mobiles_Robots_-_Past_Present_and_Future)>. Luettu 5.5.2023.
- 8 Sarkar Anirbar. 2020. Chronological History of Mobile Robots. Verkkoaineisto. Whataftercollege. <<https://whataftercollege.com/robotics-embedded-system/chronological-history-of-mobile-robotics/>>. Luettu 4.5.2023.
- 9 Spot. Verkkoaineisto. Boston Dynamics. <<https://www.bostondynamics.com/products/spot>>. Luettu 6.5.2023.
- 10 Atlas. Verkkoaineisto. Boston Dynamics. <<https://www.bostondynamics.com/atlas>>. Luettu 6.5.2023.
- 11 ErgoMover Wireless. 2023. Verkkoaineisto. GMV. <<https://gmvas.dk/en/ergomover-wireless-en/>>. Luettu 6.5.2023.
- 12 PEF 500. 2023. Verkkoaineista. GT Cranes. <<https://www.gt-cranes.com/en/crane-for-special-lifting/lift-crane-pef-500/>>. Luettu 6.5.2023.

- 13 Small Worm Geared Motors SFS Size 3 Data Sheets. 2011. Tuoteperheen tekniset tiedot. Mädler. <[https://smarthost.maedler.de/datenblaetter/SFS\\_3\\_EN.pdf](https://smarthost.maedler.de/datenblaetter/SFS_3_EN.pdf)>. Luettu 5.5.2023.
- 14 EM-176A DC-MOTOR CONTROLLER 12/24Vdc 10A. Tuotteen tekniset tiedot. <[https://electromen.com/files/3816/1129/6638/EN\\_em-176A.pdf](https://electromen.com/files/3816/1129/6638/EN_em-176A.pdf)>. Luettu 5.5.2023.
- 15 Introduction to Brushless DC Motors (BLDC Motor). 2019. Verkkoaineisto. Electronics Hub. <<https://www.electronicshub.org/brushless-dc-motor-blcd-motor/>>.
- 16 EM-347A BRUSHLESS DC-MOTOR DRIVER 12-36V 50A. Tuotteen tekniset tiedot. <[https://electromen.com/files/3216/0820/4489/EN\\_em-347A.pdf](https://electromen.com/files/3216/0820/4489/EN_em-347A.pdf)>.
- 17 Raspberry Pi 4 Computer. 2021. Verkkoaineisto. Raspberry Pi. <<https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf>>. Luettu 17.3.2023.
- 18 ADC-DAC Pi Zero. 2023. Verkkoaineisto. AB Electronics UK. <<https://www.abelectronics.co.uk/p/74/adc-dac-pi-zero-raspberry-pi-adc-and-dac-expansion-board>>. Luettu 17.4.2023.
- 19 Xiaomi Mi Electric Scooter Essential User manual. 2020. Sähköskootterin mukana toimitettu paperinen käyttöohje. Xiaomi.

**Osalista**

STEP-DOWN DC/DC-MUUNNIN 1,2-36V 20A CCV	<a href="https://www.partco.fi/fi/te-holaehheet/dcdc-jaennitemuuntimet/19265-dd-yd4020j.html">https://www.partco.fi/fi/te-holaehheet/dcdc-jaennitemuuntimet/19265-dd-yd4020j.html</a>
STEP-DOWN DC/DC-MUUNNIN 1,5-35V 3A MINI	<a href="https://www.partco.fi/fi/te-holaehheet/dcdc-jaennitemuuntimet/21119-dd-pch2596s.html">https://www.partco.fi/fi/te-holaehheet/dcdc-jaennitemuuntimet/21119-dd-pch2596s.html</a>
SparkFun LOGIIKKATASOMUUNNIN 1,8-5,5V (2-DIR)	<a href="https://www.partco.fi/fi/rakennussarjat/sparkfun/19079-spf-bob-12009.html">https://www.partco.fi/fi/rakennussarjat/sparkfun/19079-spf-bob-12009.html</a>
EM-176A	<a href="https://electromen.com/en/products/item/motor-controllers/dc-motor-below-10A/em-176A">https://electromen.com/en/products/item/motor-controllers/dc-motor-below-10A/em-176A</a>
EM-347A	<a href="https://electromen.com/en/products/item/motor-controllers/brushless-dc-motor/EM-347A">https://electromen.com/en/products/item/motor-controllers/brushless-dc-motor/EM-347A</a>
Potentiometri 10 kΩ	
Raspberry Pi 4	<a href="https://www.raspberrypi.com/products/raspberry-pi-4-model-b/">https://www.raspberrypi.com/products/raspberry-pi-4-model-b/</a>
ADC-DAC Pi Zero	<a href="https://www.abelectronics.co.uk/p/74/adc-dac-pi-zero-raspberry-pi-adc-and-dac-expansion-board">https://www.abelectronics.co.uk/p/74/adc-dac-pi-zero-raspberry-pi-adc-and-dac-expansion-board</a>
KEINUKYTKIN 2-NAPAINEN ON/OFF 16A 250VAC	<a href="https://www.partco.fi/fi/saehkoemekaniikka/kytkimet/keinukytkimet/7332-kyt-c1550atbb.html">https://www.partco.fi/fi/saehkoemekaniikka/kytkimet/keinukytkimet/7332-kyt-c1550atbb.html</a>
HC-SR04	<a href="https://www.amazon.de/-/en/ELEGOO-HC-SR04-Ultrasonic-Distance-Arduino/dp/B01M9CMJ9O/">https://www.amazon.de/-/en/ELEGOO-HC-SR04-Ultrasonic-Distance-Arduino/dp/B01M9CMJ9O/</a>

# Piirikaavio

