



Tero Nikkola

Robo Garagen mobiilirobottialustan sähkösuunnittelu ja ohjauksen kehittäminen

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Konetekniikan tutkinto-ohjelma

Insinöörityö

8.3.2026

Tiivistelmä

Tekijä:	Tero Nikkola
Otsikko:	Robo Garagen mobiilirobottialustan sähkösuunnittelu ja ohjauksen kehittäminen
Sivumäärä:	53 sivua + 2 liitettä
Aika:	8.3.2026
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Konetekniikka
Ammatillinen pääaine:	Koneautomaatio
Ohjaajat:	Lehtori Antti Liljaniemi

Tämän opinnäytetyön tavoitteena oli palauttaa puretusta mobiilirobotista toimiva mobiilirobotiikan kehitysalusta, jota voidaan hyödyntää robotiikan opetuksessa, opiskelijaprojekteissa sekä järjestelmän jatkokehityksessä Metropolia Ammattikorkeakoulun Robo Garagessa. Työssä suunniteltiin ja toteutettiin robotin sähköjärjestelmä, ohjausjärjestelmä sekä modulaarinen ohjelmistoarkkitehtuuri.

Robotin sähkösuunnittelu toteutettiin kolmelle 36 voltin akulle perustuvana järjestelmänä, jossa moottorinohjaus, ohjaustietokone ja anturijärjestelmät on jaettu toiminnallisiin kokonaisuuksiin. Järjestelmä sisältää päävirtapiirin, suojaukset sekä kaksikanavaisen hätäseis-järjestelmän. Robotin liikkeestä vastaavat BLDC-napamoottorit, joita ohjataan moottorinohjaimien kautta Modbus RS-485 -väylän avulla.

Robotin ohjausjärjestelmä toteutettiin ZED Box Orin NX -tietokoneen ympärille. Ohjelmisto kehitettiin Python-kielellä modulaarisena järjestelmänä, jossa robotin toiminta on jaettu sensori-, havainnointi-, päätöksenteko- ja ohjausmoduuleihin. Robotin käyttöliittymä toteutettiin selainpohjaisena sovelluksena NiceGUI-kirjaston avulla. Järjestelmän toimintaa testattiin moottorien testausohjelmalla, Modbus-väyläkommunikaation testauksella sekä Docker-kontissa toteutetulla anturisimulaatiolla.

Työn tuloksena syntyi toimiva mobiilirobotiikan kehitysalusta, jonka laitteisto ja ohjelmisto on suunniteltu ja dokumentoitu jatkokehitystä varten. Kehitetty järjestelmä mahdollistaa robotiikan ohjelmisto- ja laitteistokehityksen opetuskäytössä sekä tarjoaa perustan järjestelmän laajentamiselle esimerkiksi paikannukseen, kartoitukseen ja autonomiseen navigointiin.

Avainsanat: mobiilirobotiikka, robottialusta, sähkösuunnittelu, robottiohjaus, Python, Modbus

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Tero Nikkola
Title: Design of the Electrical System and Control Architecture for the Robo Garage Mobile Robot Platform
Number of Pages: 53 pages + 2 appendices
Date: 8 March 2026

Degree: Bachelor of Engineering
Degree Programme: Mechanical Engineering
Professional Major: Machine Automation
Supervisor: Antti Liljaniemi, Senior Lecturer

The objective of this thesis was to restore a previously dismantled mobile robot into a functional development platform for robotics education, student projects, and further research and development at Metropolia University of Applied Sciences' Robo Garage. The work focused on the design and implementation of the robot's electrical system, control system, and modular software architecture.

The electrical system of the robot was designed around three 36-volt batteries. The system separates the motor drives, control computer, and sensor subsystems into functional modules in order to support modular development and system expandability. The electrical design includes a main power circuit, protection components, and a dual-channel emergency stop system. The robot is driven by BLDC hub motors controlled through motor drivers via a Modbus RS-485 communication bus.

The control system is built around a ZED Box Orin NX embedded computer. The robot software was implemented in Python using a modular architecture consisting of sensor, perception, decision, and control modules. A browser-based user interface was developed using the NiceGUI framework to enable monitoring and control of the robot through a web interface.

The system was tested using a dedicated motor testing program, Modbus communication tests, and a sensor simulation environment implemented in a Docker container. The results demonstrate that the developed system provides a functional foundation for robotics development and experimentation.

The developed platform can be further extended with sensor data processing, localization, mapping, and autonomous navigation capabilities.

Keywords: mobile robotics, robot platform, electrical system design, robot control, Python, Modbus

Sisällys

Lyhenteet ja käsitteet

1	Johdanto	1
2	Mobiilirobotiikka	3
2.1	Mobiilirobotiikan käsite	3
2.2	Mobiilirobotiikka vuonna 2025	5
2.3	Mobiilirobotiikan mahdollisuudet	7
3	Funktionaalinen suunnittelu ja ohjelmointi	8
4	Robo Garagen mobiilirobottialusta "Jaska"	10
4.1	Sähkösuunnittelu	10
4.1.1	Sähkönsyöttö ja päävirtapiiri	11
4.1.2	Anturointi	13
4.1.3	Hätäseis-järjestelmä	14
4.1.4	Moottorit ja moottorien ohjaus	16
4.2	Ohjausjärjestelmä	17
4.3	Runkorakenne	18
4.3.1	Pyörätuenta	19
4.3.2	Sähkökotelointi	20
4.3.3	Lisäkotelo	22
4.3.4	Anturijalusta	23
4.4	Ohjelmisto	24
4.4.1	Moottoreiden testausohjelma	26
4.4.2	Robotin käyttöohjelman ohjelmistorakenne	28
4.4.3	Main.py:n ja ohjaussilmukan toiminta	30
4.4.4	Käyttöliittymä	33
4.5	Robotin käyttö	39
4.6	Testaus ja toimivuuden arviointi	40
4.6.1	Moottorien ja Modbus-väylän testaus	41
4.6.2	Ohjelmiston toiminnan testaus ja tilakone	42
4.6.3	Anturisimulaation testaus	42
4.6.4	Testauksen yhteenveto	43
5	Jatkokehitys ja kehityskohteet	45
5.1	Laitteistoon liittyvät jatkokehityskohteet	45
5.2	Ohjelmiston jatkokehitys	47
6	Yhteenveto	49

Liitteet

Liite 1: Sähkökuvat

Liite 2: Ehdotus pääpiirin kytkentämuutoksesta

Lyhenteet ja käsitteet

- AGV: *Automated Guided Vehicle*. Automaattisesti ohjattu ajoneuvo, joka liikkuu ennalta määritellyä reittiä pitkin.
- AMR: *Autonomous Mobile Robot*. Autonominen mobiilirobotti, joka pystyy navigoimaan ympäristössään itsenäisesti ja tekemään päätöksiä esteiden ja tehtävien mukaan.
- BLDC: *Brushless Direct Current Motor*. Harjaton tasasähkömoottori, jota käytetään robottien moottoreissa sen kestävyden, tehokkuuden ja hallittavuuden vuoksi.
- Docker: Konttipohjainen virtualisointiympäristö, jossa sovellukset ja niiden riippuvuudet voidaan eristää isäntäkoneesta ja ajaa turvallisesti. Työssä käytetty robotin anturisimulaation ajamiseen tarkoitettu ympäristö.
- FastAPI: Python-pohjainen verkkorajapintakehys, jolla voidaan toteuttaa tehokkaita HTTP-pohjaisia ohjelmointirajapintoja (API).
- GNSS: *Global Navigation Satellite System*. Maailmanlaajuinen satelliittipaikannusjärjestelmä, joka mahdollistaa robotin sijainnin määrittämisen GPS-tyyppisesti.
- Hall-anturi: Moottorin anturi, joka mittaa magneettikentän muutoksia ja mahdollistaa moottorin kierrosnopeuden laskennan.
- IMU: *Inertial Measurement Unit*. Inertiamittausyksikkö, joka mittaa kiihtyvyyksiä ja kulmanopeuksia, ja jota käytetään robotin liikkeen ja asennon arviointiin.

- LiDAR:** *Light Detection and Ranging*. Etäisyydenmittausmenetelmä, jossa lasersädettä käytetään ympäristön kolmiulotteisen pistepilven muodostamiseen esteiden ja kohteiden havaitsemiseksi.
- Modbus:** Teollisuusväyläprotokolla, jota käytetään ohjaimien ja laitteiden väliseen tiedonsiirtoon, esimerkiksi moottorinohjaimien ohjaukseen.
- NiceGUI:** Python-pohjainen selainkäyttöliittymäkirjasto, jota käytettiin robotin ohjaus- ja monitorointisovelluksen rakentamiseen.
- PWM:** *Pulse Width Modulation*. Pulssinleveysmodulaatio, jota käytetään moottorien nopeuden ohjaukseen.
- ROS:** *Robot Operating System* Robottiikan avoimen lähdekoodin ohjelmistokehys, joka tarjoaa kirjastoja ja työkaluja robottisovellusten kehittämiseen.
- SLAM:** *Simultaneous Localization and Mapping*. Menetelmä, jossa robotti muodostaa karttaa ympäristöstään samalla kun se arvioi omaa sijaintiaan kartalla anturidatan, kuten LiDARin tai kameran, perusteella.
- SSH:** *Secure Shell*. Suojattu etäyhteysprotokolla, jonka avulla voidaan muodostaa yhteys toiseen tietokoneeseen esimerkiksi komentoriviltä ja hallita sitä etänä.
- UAV:** *Unmanned Aerial Vehicle*. Miehittämätön ilma-alus, kuten lennokki tai drone, joka voi liikkua itsenäisesti tai kauko-ohjauksella.
- UGV:** *Unmanned Ground Vehicle*. Miehittämätön maa-ajoneuvo, kuten robotti, joka liikkuu maassa ilman kuljettajaa.
- UUV:** *Unmanned Underwater Vehicle*. Miehittämätön sukellusvene, jota käytetään vedenalaisiin tutkimuksiin ja tehtäviin.

UWV: *Unmanned Water Vehicle*. Miehitämätön vesialus, esimerkiksi robottivene, jota käytetään tutkimus- ja tarkkailutehtävissä.

Opinnäytetyön kirjoitusprosessissa on hyödynnetty OpenAI:n ChatGPT-kielimallia (GPT-5.2) tukena tekstin muotoilussa, rakenteen selkeyttämisessä sekä vaihtoehtoisten sanamuotojen ja otsikointien tuottamisessa. Tekoälyä on käytetty erityisesti teknisten kuvausten hiomisessa ja saavutettavuuteen liittyvien vaihtoehtoisten tekstien laatimisessa. Opinnäytetyön tekijä on vastannut työn sisällöstä, teknisestä toteutuksesta, johtopäätöksistä sekä lopullisista tekstivalinnoista.

1 Johdanto

Mobiilirobotit ovat keskeinen osa modernia robotiikkaa ja autonomisia järjestelmiä. Niitä hyödynnetään laajasti esimerkiksi teollisuudessa, tutkimuksessa ja opetuksessa. Robotiikan opetuksessa käytännönläheiset kehitysalustat ovat erityisen tärkeitä, sillä niiden avulla opiskelijat voivat perehtyä robottijärjestelmien suunnitteluun, toteutukseen ja testaamiseen käytännössä. Tällaiset alustat mahdollistavat myös erilaisten robotiikan osa-alueiden, kuten mekaniikan, sähköjärjestelmien, ohjelmistokehityksen ja anturoinnin, tarkastelun yhtenä kokonaisuutena.

Tämän opinnäytetyön lähtökohtana oli Metropolia Ammattikorkeakoulun Robo Garagessa ollut mobiilirobotti, joka oli purettu eikä ollut enää käyttövalmiissa kunnossa. Tämän vuoksi järjestelmää ei voitu hyödyntää opetuksessa tai kehitystyössä ilman uudelleenrakentamista.

Työn tarkoituksena oli palauttaa mobiilirobotista toimiva kehitysalusta, jota voidaan hyödyntää robotiikan opetuksessa, opiskelijaprojekteissa sekä järjestelmän jatkokehityksessä. Tavoitteena oli suunnitella ja toteuttaa robotille toimiva laitteisto- ja ohjelmistorakenne sekä mahdollistaa robotin manuaalinen ohjaus ja yksinkertainen ensimmäisen asteen automaatio. Lisäksi tavoitteena oli toteuttaa järjestelmä siten, että sitä voidaan myöhemmin laajentaa esimerkiksi autonomiseen navigointiin, kartoitukseen ja erilaisten anturijärjestelmien hyödyntämiseen.

Työssä hyödynnettiin funktionaalista suunnittelua, jossa järjestelmä jaettiin erillisiin toiminnallisiin moduuleihin. Moduulirakenne helpottaa järjestelmän hallintaa, testaamista ja jatkokehitystä sekä mahdollistaa yksittäisten järjestelmäosien päivittämisen ilman merkittäviä muutoksia koko järjestelmään.

Työn laajuutta rajattiin siten, että varsinaista anturidatan hyödyntämistä, kuten stereokameran syvyystiedon tai IMU- ja GNSS-datan käyttöä, ei toteutettu täysimääräisesti. Näille toiminnoille suunniteltiin kuitenkin ohjelmistoon valmiit

rajapinnat ja rakenteet, jotta järjestelmää voidaan laajentaa myöhemmissä projekteissa.

Työn toisessa luvussa esitellään mobiilirobotiikan käsitettä, nykytilaa sekä mobiilirobotiikan tarjoamia mahdollisuuksia. Kolmannessa luvussa käsitellään funktionaalista suunnittelua ja ohjelmointia, joita hyödynnettiin työn toteutuksessa. Neljännessä luvussa esitellään Robo Garagen mobiilirobottialusta Jaska sekä sen laitteisto- ja ohjelmistorakenne. Viidennessä luvussa tarkastellaan järjestelmän mahdollisia jatkokehityskohteita. Kuudennessa luvussa esitetään työn yhteenveto ja keskeiset johtopäätökset.

2 Mobiilirobotiikka

2.1 Mobiilirobotiikan käsite

Mobiilirobotit ovat robottijärjestelmiä, jotka kykenevät liikkumaan ympäristössään ja muuttamaan sijaintiaan ilman kiinteää asennusta. Toisin kuin paikallaan toimivat teollisuusrobotit, mobiilirobotit pystyvät siirtymään paikasta toiseen liikkumismekanisminsa avulla. [1, s. 8.] Robotin liikkumista ohjataan ohjelmiston avulla antureista saadun tiedon perusteella. Mobiilirobotteja voidaan luokitella myös niiden navigointitavan perusteella. Karkeasti mobiilirobotit voidaan jakaa Automated Guided Vehicle (AGV)- ja Autonomous Mobile Robot (AMR) -luokkiin. AGV-järjestelmissä navigointi perustuu ennalta rakennettuun infrastruktuuriin, kun taas AMR-robotit pystyvät liikkumaan dynaamisesti muuttuvassa ympäristössä ilman ennalta määriteltyä reittiä.

Automaattisesti ohjatut ajoneuvot (AGV) ovat kiinteäreittisiä robotteja, jotka seuraavat ennalta määriteltyjä reittejä esimerkiksi lasereiden, majakoiden, viivakoodien tai lattiaan upotettujen magneettinauhujen avulla. [2, s. 142.] Tyypillinen esimerkki AGV-järjestelmästä ovat Rocla AGV-trukit, joita käytetään tavaroiden siirtelyyn logistiikkakeskuksissa, varastoissa ja tehtaissa (Kuva 1). [3].



Kuva 1. Rocla AGV

Autonomiset mobiilirobotit (AMR) eivät tarvitse ennalta määriteltyjä reittejä, vaan ne kykenevät anturointinsa avulla ympäristön kartoittamiseen ja autonomiseen navigointiin [2, s. 142–145]. Kotitalouksissa käytettävät robotti-imurit ovat hyvä esimerkki AMR-teknologiasta (Kuva 2). Ne kykenevät itse kartoittamaan siivottavan alueen ja suorittamaan siivouksen autonomisesti ympäristön havainnointiin tarkoitettujen järjestelmien avulla. [4.]



Kuva 2. Roborock Saros Z70 robotti-imuri.

Mobiilirobotit voidaan luokitella myös toimintaympäristön perusteella maalla liikkuviin (UGV), ilmassa toimiviin (UAV) sekä vedessä toimiviin alustoihin (UUU ja UWV). Maalla liikkuvat robotit käyttävät liikkumiseen tyypillisesti pyöriä, teloja tai jalkoja [5]. Liikkumismekanismi valitaan käyttöympäristön ja tehtävän vaatimusten perusteella. Pyörälliset rakenteet ovat energiatehokkaita ja yleisesti käytettyjä sekä sisä- että ulkokäytössä, kun taas telat ja jalkarakenteet soveltuvat paremmin epätasaiseen ja vaativaan maastoon.

Autonomisessa toiminnassa mobiilirobotin tulee kyetä jatkuvasti havainnoimaan ympäristöään. Tämä toteutetaan erilaisten antureiden avulla, kuten LiDAR-, IMU- ja GPS-antureiden avulla sekä moottoreihin integroiduilla enkoodereilla, joiden avulla robotti saa tietoa ympäristöstään ja omasta liikkeestään. [6, s. 30–31.]

2.2 Mobiilirobotiikka vuonna 2025

Vuonna 2025 mobiilirobotiikka on siirtynyt kokeellisista ratkaisuista osaksi arkipäiväisiä palveluja, logistiikkaa ja turvallisuuskriittisiä sovelluksia. Kehitystä on vauhdittanut autonomisten ja etäohjattujen mobiilirobotien kyky toimia ympäristöissä, joissa ihmistyö on hidasta, vaarallista tai kustannuksiltaan kallista.

Suomessa yksi näkyvimmistä mobiilirobotiikan sovelluksista on autonominen jakelurobotiikka. Esimerkiksi Starship Technologies on ottanut käyttöön kaupunkiympäristössä toimivia jakelurobotteja, jotka hyödyntävät karttapohjaista navigointia ja anturointia. [7.] Toimintaa on laajennettu myös yhteistyössä ruokakuljetuspalveluiden, kuten Woltin, kanssa [8], mikä osoittaa mobiilirobottien soveltuvuuden osaksi olemassa olevia digitaalisia palveluketjuja.

Mobiilirobotiikka on noussut merkittävään rooliin myös puolustus- ja turvallisuussektorilla. Esimerkiksi Laykka-AMPGV on autonominen ja etäohjattava maastoajoneuvo, jota voidaan käyttää muun muassa tiedusteluun,

huoltotehtäviin ja vaikuttamistehtäviin. Järjestelmän kehityksessä korostuvat kustannustehokkuus ja uhrattavuus, mikä mahdollistaa robottien käytön tehtävissä, joissa ihmishenkien suojaaminen on ensisijaista. Vastaavanlainen kehitys on ollut nähtävissä myös kansainvälisesti, erityisesti Ukrainan sodan yhteydessä, jossa maassa liikkuvia robottijärjestelmiä on hyödynnetty tiedustelussa ja miinanraivauksessa. [9,10.]

Siviilisovelluksissa mobiilirobotteja hyödynnetään yhä enemmän terveydenhuollossa ja julkisissa tiloissa. Autonomiset kuljetusrobotit voivat siirtää tarvikkeita, ateroita ja lääkkeitä monikerroksisissa rakennuksissa, joissa infrastruktuuri ei ole alun perin suunniteltu automaatiota varten. [11, 12.] Lisäksi autonomiset siivousrobotit ovat yleistyneet erilaisissa julkisissa ympäristöissä.

Teollisuudessa ja logistiikassa autonomiset mobiilirobotit ovat vakiintumassa osaksi sisälogistiikkaa. Varastoissa ja tuotantolaitoksissa ne korvaavat tai täydentävät manuaalisia lavansiirtovaunuja ja haarukkatrukkeja, mahdollistaen joustavamman materiaalivirran ilman kiinteää infrastruktuuria. Tällaiset AMR-järjestelmät pystyvät toimimaan samoissa tiloissa ihmisten kanssa ja mukautumaan muuttuviin tuotantoprosesseihin. [13.]

Lisäksi mobiilirobotiikkaa hyödynnetään tutkimuksessa ja ympäristöseurannassa, joissa robotit toimivat mittausalustoina vaikeapääsyisissä tai toistettavuutta vaativissa kohteissa [13].

Esimerkit osoittavat, että mobiilirobotiikan sovellukset ovat laajentuneet nopeasti useille toimialoille. Teknologian kehitystä ohjaavat erityisesti autonomian lisääminen, kustannustehokkaat rakenteet sekä järjestelmät, jotka kykenevät toimimaan luotettavasti dynaamisissa ja osittain ennakoimattomissa ympäristöissä.

2.3 Mobiilirobotiikan mahdollisuudet

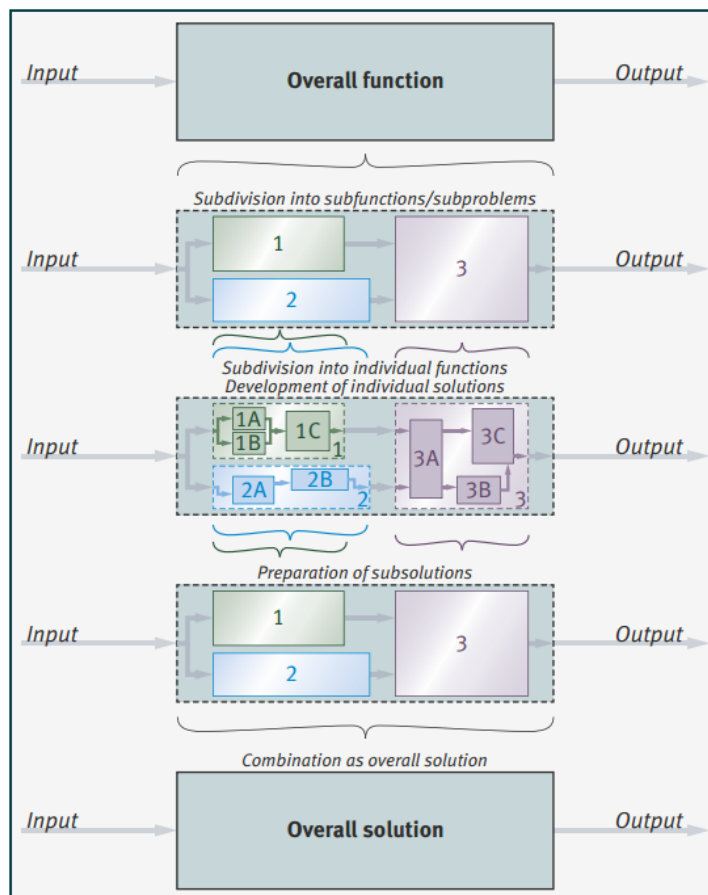
Mobiilirobotiikan tulevaisuuden mahdollisuudet liittyvät erityisesti robottien kykyyn toimia itsenäisemmin, monipuolisemmissa ympäristöissä ja aiempaa laajemmilla sovellusalueilla. Teknologinen kehitys mahdollistaa mobiilirobottien hyödyntämisen tehtävissä, jotka ovat aiemmin olleet teknisesti tai taloudellisesti haastavia. Autonomian ja sensoritekniiikan kehittyminen parantaa robottien kykyä havainnoida ympäristöään ja mukautua muuttuviin olosuhteisiin ilman jatkuvaa ihmisen ohjausta. Tämä mahdollistaa luotettavan toiminnan myös vaativissa ympäristöissä, kuten ulkokohteissa ja haastavissa ilmasto-olosuhteissa, sekä lisää robottien tehokkuutta ja turvallisuutta.

Erilaisten liikkumisratkaisujen, kuten pyörä-, tela- ja jalkarakenteiden, kehitys laajentaa mobiilirobottien käyttömahdollisuuksia ympäristöihin, joihin perinteiset ajoneuvot eivät sovellu. Erityisesti jalalliset mobiilirobotit mahdollistavat kartoitus-, tarkastus- ja tutkimustehtäviä epätasaisessa ja vaikeakulkuisessa maastossa, joissa liikkumiskyky on keskeinen rajoittava tekijä.

Edellä kuvatut teknologiset ja toiminnalliset mahdollisuudet heijastuvat myös mobiilirobotiikan markkinoiden nopeaan kasvuun. Vuonna 2024 mobiilirobotiikan globaalin markkinan kooksi arvioitiin noin 25,4 miljardia Yhdysvaltain dollaria, ja sen ennustetaan kasvavan noin 73,7 miljardiin dollariin vuoteen 2030 mennessä. Markkinan arvioitu vuotuinen kasvu vuosina 2025–2030 on 20,7 %, mikä kuvastaa automaation lisääntyvää hyödyntämistä muun muassa teollisuudessa, logistiikassa ja terveydenhuollossa. [14.]

3 Funktionaalinen suunnittelu ja ohjelmointi

Tässä työssä hyödynnettiin funktionaalista suunnittelua sekä järjestelmän rakenteellisessa suunnittelussa että ohjelmistokehityksessä. Lähestymistavan perusidea on jakaa kokonaisuus pienempiin, hallittaviin funktioihin, joista kukin vastaa tietyistä järjestelmän toiminnosta. Funktiot voidaan suunnitella, toteuttaa ja testata erikseen, ja niitä voidaan monistaa tai käyttää osana toisia funktioita (Kuva 3). [15, s. 442–444.]



Kuva 3. Funktionaalinen hajautus ja funktioiden yhdistäminen osaksi kokonaisjärjestelmää

Suunnittelun näkökulmasta funktionaalinen lähestymistapa helpottaa mobiilirobotin kehitystyön jäsentämistä selkeisiin ja hallittaviin kokonaisuuksiin.

Kehitys voidaan jakaa aluksi laitteisto- ja ohjelmistotasolle.

Laitteistokokonaisuus voidaan edelleen jakaa runkorakenteeseen, sähköjärjestelmään ja anturointiin. Vastaavasti ohjelmisto voidaan jakaa pääohjelmaan, antureiden lukemiseen sekä robotin ohjaukseen liittyviin ohjelmistomoduuleihin. Funktionaalinen rakenne mahdollistaa mobiilirobotin käytön opetusalustana sekä erillisten opiskelijaprojektien toteuttamisen ilman, että koko järjestelmää tarvitsee muuttaa. Tämä rakenne tukee myös syvällistä perehtymistä tiettyihin osa-alueisiin, kuten LiDAR-teknoologiaan, ilman vaikutusta muihin järjestelmän komponentteihin.

Lisäksi mobiilirobotti toimii monialaisena projektialustana: tietotekniikan opiskelijat voivat keskittyä ohjelmistoihin ja käyttöliittymiin, sähkötekniikan opiskelijat sähköjärjestelmiin, tehon- ja lämmönhallintaan, ja kone- sekä ajoneuvotekniikan opiskelijat runko- ja rengasratkaisuihin sekä mekaanisten tukirakenteiden kehittämiseen.

4 Robo Garagen mobiilirobottialusta ”Jaska”

Opinnäytetyön kohteena olevan Robo Garagen mobiilirobottialustan ”Jaska” kehitys pohjautuu alun perin Big Flash -hankkeeseen, jossa kehitettiin konenäköön liittyviä ratkaisuja Oinride Oy:lle. Hankkeen yhteydessä Metropolia Ammattikorkeakoulu hankki AutoJoe-mobiilirobotin mobiilirobottikehitysalustaksi MiR100-robotin rinnalle.

MiR100 on valmis kaupallinen ratkaisu, joka on suunniteltu pääasiassa sisätiloihin ja logistiikkakäyttöön, kun taas AutoJoe on ulkokäyttöön suunniteltu Mars-rover-tyyppinen mobiilirobottialusta, jossa on hyvä maavara ja yli 100 kg:n kantokyky. AutoJoe soveltuu erityisesti vaativiin käyttöympäristöihin ja avaa mahdollisuuksia robotiikkaratkaisujen kehittämiseen uusiin ympäristöihin.

Metropolian Robo Garagella AutoJoe-alustaa on kehitetty edelleen mekaniikan ja elektroniikan osalta, ja se toimii epävirallisesti laitteen viidentenä prototyyppiversiona. Tässä opinnäytetyössä alustaa kehitettiin eteenpäin ja saatettiin uudelleen toimintakuntoon opetuskäyttöön soveltuvaksi mobiilirobottialustaksi, joka mahdollistaa mobiilirobotiikan tutkimisen, kehittämisen ja jatkoprojektit eri tutkintosuuntausten opiskelijoille. [16.]

4.1 Sähkösuunnittelu

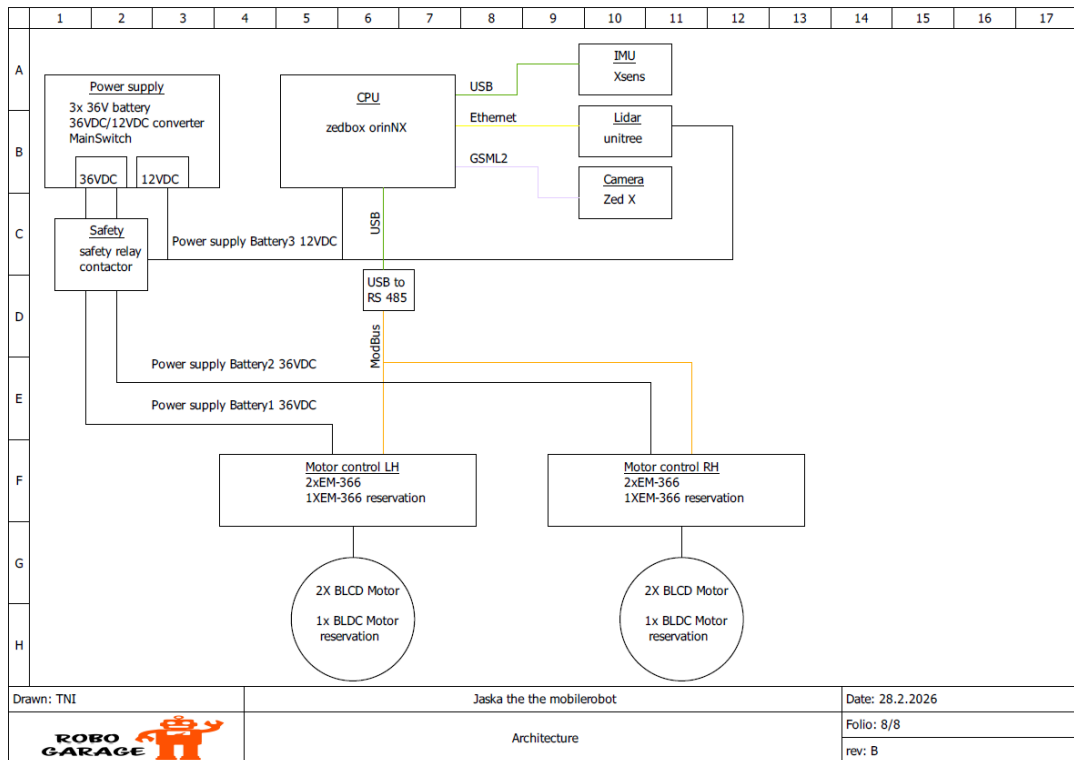
Mobiilirobotin sähkösuunnittelu on keskeinen osa kokonaisjärjestelmän toimivuutta, turvallisuutta ja laajennettavuutta. Tässä opinnäytetyössä sähkösuunnittelun tavoitteena oli toteuttaa luotettava ja turvallinen sähköjärjestelmä, joka tukee mobiilirobotin opetuskäyttöä Robo Garagessa sekä mahdollistaa järjestelmän jatkokehityksen ja erilaisten opiskelijaprojektien toteuttamisen. Suunnittelussa kiinnitettiin erityistä huomiota sähkönsyötön rakenteeseen, suojauksiin, komponenttien mitoitukseen sekä järjestelmän modulaarisuuteen.

Sähköjärjestelmä suunniteltiin siten, että mobiilirobotin keskeiset toiminnalliset kokonaisuudet, kuten moottorihjaus, anturointi ja ohjausjärjestelmä, ovat selkeästi eroteltavissa toisistaan. Tämä tukee funktionaalista suunnittelumenetelmää ja helpottaa vianhakua, huoltoa sekä yksittäisten osaluokkien kehittämistä ilman vaikutuksia koko järjestelmään.

Tässä luvussa käsitellään mobiilirobotin sähkönsyöttöä ja päävirtapiiriä, käytettyä anturointia, hätäseis-järjestelmää sekä moottoreita ja niiden ohjausta. Sähköjärjestelmän kokonaisuus, kytkennät sekä osaluettelo on esitetty liitteessä 1.

4.1.1 Sähkönsyöttö ja päävirtapiiri

Mobiilirobotin sähkönsyöttö toteutettiin kolmella 36 V:n akulla, joiden miinusnavat on yhdistetty yhteiseen potentiaaliin. Yksi akku syöttää virran vasemmanpuoleisille moottoreille, toinen oikeanpuoleisille moottoreille ja kolmas akku eristetylle DC-DC-muuntimelle (Orion-Tr 48/12-20A), joka muodostaa 12 VDC-jännitteen käyttöjärjestelmälle (Kuva 4).

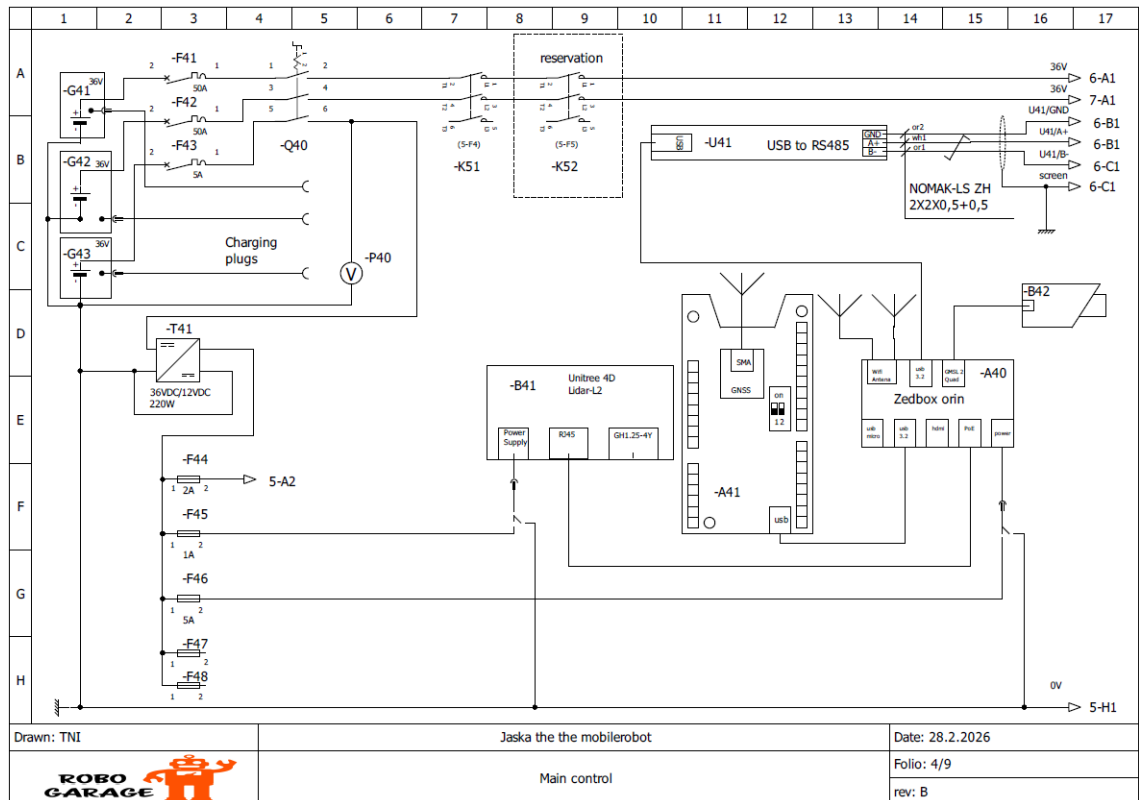


Kuva 4. Robotin järjestelmäarkkitehtuuri

36 V jännitetaso valittiin moottoreiden vaatimusten mukaisesti ja sen yhteensopivuuden vuoksi muiden komponenttien kanssa. 12 V taso puolestaan palvelee keskeisiä komponentteja, kuten tietokonetta ja LiDAR-anturia, ilman tarpeettoman monimutkaista väliportaiden käyttöä.

36 V puolen suojaukset toteutettiin Schneider Electricin C60H -sarjan DC-suojakatkaisijoilla, joista jokaisella akulla ja moottoriohjaimella on oma. 12 V puolen suojauksessa käytettiin ATO-lattasulakkeita, jotka erottelevat ohjauspiirin, tietokoneen ja LiDAR-anturin syötön. Lisäksi sulakekoteloon jätettiin kaksi vapaata paikkaa tuleville lisälaitteille (Kuva 5).

Järjestelmän pääkytkimenä toimii kolminapainen ABB OT63FT3, joka sijaitsee pääsulakkeiden jälkeen ja mahdollistaa koko järjestelmän erottamisen turvallisesti huolto- ja vikatilanteissa.



Kuva 5. Robotin pääpiiri

4.1.2 Anturointi

Mobiilirobotin anturointi mahdollistaa ympäristön havainnoinnin, liikkeen seurannan ja tulevat automaatio- sekä autonomiatoiminnot. Anturointi tukee sekä manuaalista ohjausta että jatkokehitystä kohti autonomisempaa toimintaa.

Pääasiallinen ympäristön havainnointianturi on Unitree 4D LiDAR L2, joka mittaa etäisyyksiä lasersäteiden avulla ja mahdollistaa 3D-kartoituksen sekä esteiden havaitsemisen SLAM-menetelmillä.

Näköpohjaista havainnointia varten käytetään ZED X -stereokameraa, joka tuottaa syvyystietoa esteiden havaitsemiseen. Kamera liitetään NVIDIA ZedBox Orin NX -tietokoneeseen, joka käsittelee kameradatan reaaliaikaisesti. Kameran kestävä alumiinirunko ja IP67-luokitus mahdollistavat myös kenttäkäytön.

Liikkeen ja asennon mittaamiseen käytetään Xsens MTi-7-0I-DK -IMU-anturia, sisältäen kiihtyvyyys-, kulmanopeus- ja asentoanturit sekä GNSS-paikannuksen. IMU liitetään tietokoneeseen USB-väylän kautta, jolloin data on välittömästi käytettävissä ohjausjärjestelmässä.

4.1.3 Hätäseis-järjestelmä

Mobiilirobotin hätäseis-järjestelmä mahdollistaa moottorien virransyötön nopean katkaisun vaaratilanteessa. Järjestelmä koostuu hätäseis- ja kuittauspainikkeesta, hätäseis-releestä sekä kontaktorista, joka ohjaa moottorihjaimien syöttöjännitettä. Painikkeet sijaitsevat robotin takapaneelissa (Kuva 6).

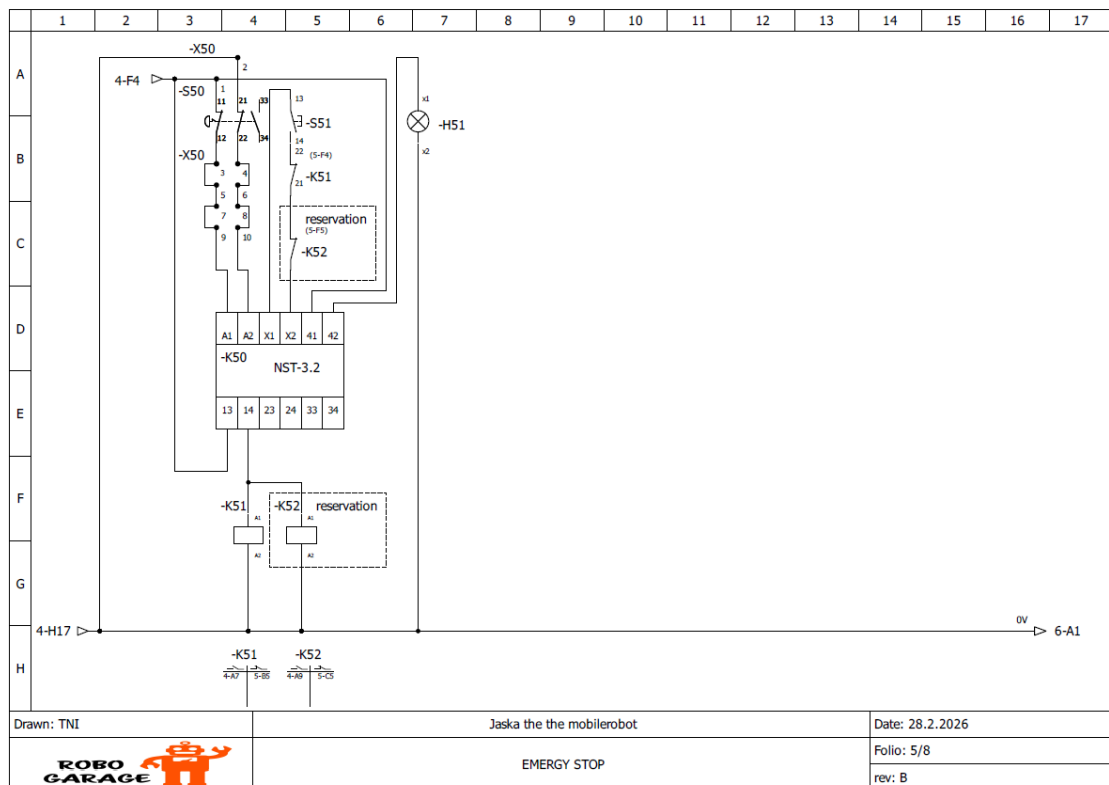


Kuva 6. Robotin takapaneeli.

Järjestelmä on toteutettu kaksikanavaisena, jolloin hätäseis-releen aktivoituminen edellyttää molempien turvakavien olevan ehjiä. Kun hätäseis-

piiri on kunnossa ja kuittauspainiketta painetaan, rele aktivoituu ja antaa ohjausjännitteen kontaktorille. Kuva 7 havainnollistaa hätäseis-piirin kaksikanavaista toteutusta. Kontaktorin vetäessä virtapiiri sulkeutuu ja moottorit saavat käyttöjännitteen.

Kun hätäseis-piiri aktivoituu, moottorien syöttö katkeaa välittömästi, mutta robotin tietokone ja anturit pysyvät jännitteellisinä (Kuva 5).

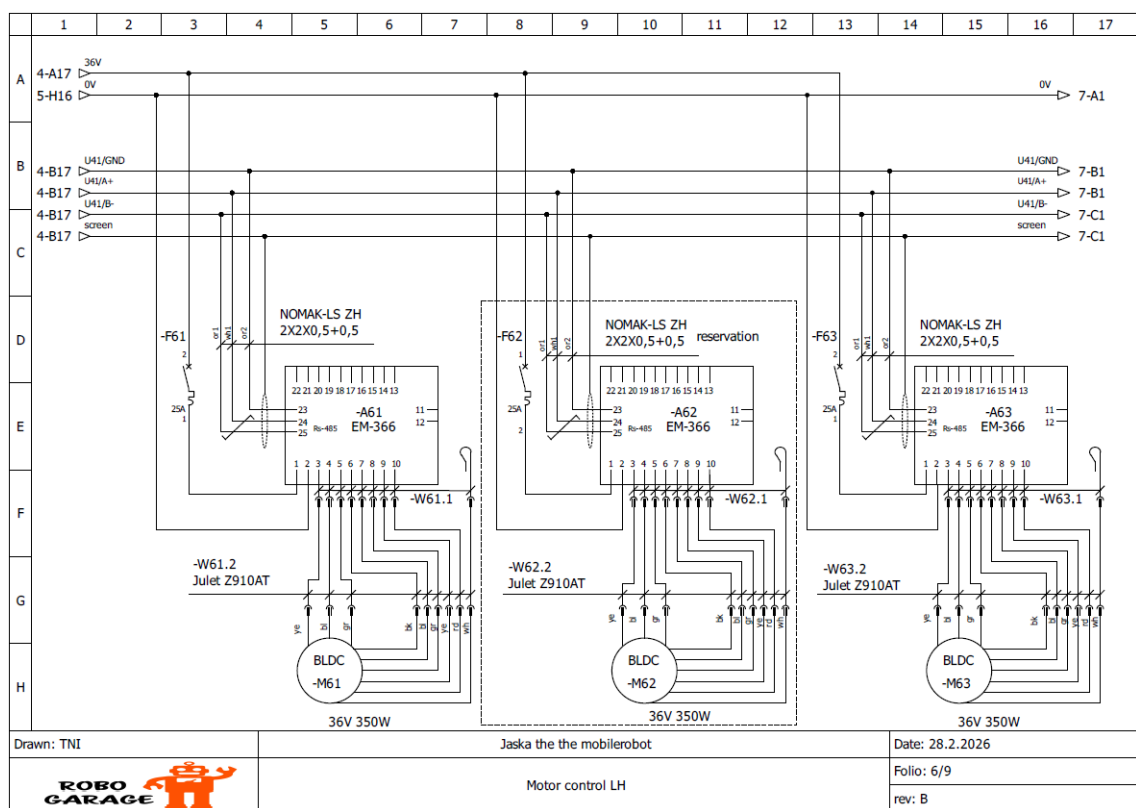


Kuva 7. Robotin hätäseispiiri

Järjestelmä on suunniteltu siten, ettei sitä voi kuitata, jos hätäseis-painike on painettuna tai kontaktori ei ole palautunut lepotilaansa. Piiri on laajennettavissa muiden turvakomponenttien liittämiseksi. Turvapiirin jälkeistä ohjausjännitettä voidaan hyödyntää myös muiden turvallisuuskriittisten laitteiden ohjaukseen. Tämä tukee turvallista ja joustavaa käyttöä opetuksessa ja opiskelijaprojekteissa.

4.1.4 Moottorit ja moottorien ohjaus

Mobiilirobotissa on neljä Ozo-electric 36 V, 350 W BLDC -napamoottoria, joille suunniteltiin tätä opinnäytetyötä varten. Suunnitelmiin tehtiin myös optio kahdelle lisämoottorille. Moottoreiden ohjauksesta vastaavat EM-366A Brushless DC-Motor Driver -ohjaimet, joita hallitaan Modbus RS-485 -väylän kautta (Kuva 8). Modbus-väylän avulla ohjataan moottoreita ja luetaan ohjaimilta tietoja, kuten jännite, virta ja taajuus, ja ohjaimia voidaan myös parametroida. Ohjaimet on valmistettu ELECTROMEN Oy:n toimesta.



Kuva 8. Piirikaavio robotin vasemmanpuoleisten moottorien ja ohjaimien kytkennöistä

EM-366A -ohjaimet (Kuva 9) ovat energiatehokkaita ja kompakteja BLDC-moottorien ohjaimia, joissa on hall-antureiden tuki sekä neljän kvadrantin toiminta. Tämä mahdollistaa moottorin nopeuden ja momentin hallinnan molempiin liikesuuntiin sekä regeneratiivisen jarrituksen, jossa liike-energia voidaan palauttaa akkujärjestelmään [17, s. 5]. Ohjaimet tukevat sekä avoimen

että takaisinkytketyn suljetun ohjauksen nopeuden hallintaa ja tarjoavat analogiset sekä digitaaliset tulot joustavaan ohjaukseen. Lisäksi ohjaimissa on suojaustoiminnot ylijännitteelle, alijännitteelle ja ylikuumenemiselle sekä häiriönilmaisuu- ulostulo. [18.]



Kuva 9. EM-366A moottorin ohjain

Vaikka suunnitelmat on tehty uusille Ozo-electric -moottoreille, opinnäytetyön aikaiseen kokoonpanoon käytettiin vanhoja hoverboard-moottoreita. Opinnäytetyön loppupuolella opiskelijaryhmä sai uudet pyörätuennat valmiiksi ja asensi uudet moottorit. Ajan ja sopivien renkaiden puutteen vuoksi koeajoja voitiin kuitenkin suorittaa vain siten, että robotti seisoj jalustan päällä, moottoreiden renkaat ilmassa.

4.2 Ohjausjärjestelmä

Mobiilirobotin ohjausjärjestelmä vastaa liikkeen hallinnasta, moottoreiden ohjauksesta sekä anturidatan reaaliaikaisesta käsittelystä. Järjestelmän keskiössä toimii ZED Box Orin NX -tietokone, joka vastaanottaa tietoa LiDAR-,

stereokamera- sekä IMU/GNSS-antureilta ja ohjaa moottorinohjaimia Modbus RS-485 -väylän kautta.

ZED Box Orin NX on Stereolabsin kehittämä sulautettu tietokone, joka perustuu NVIDIA Jetson Orin NX -alustaan. Se on ARM-pohjainen yhden piirikortin tietokone, suunniteltu erityisesti robotiikka- ja konenäkösovelluksiin. Käyttöjärjestelmänä toimii Jetson Linux, ja alusta tukee NVIDIA:n JetPack-ohjelmistopakettia, sisältäen muun muassa CUDA- ja konenäkökiihdytyskirjastot. [19.]

Tässä mobiilirobotissa käytetty 16GB-malli tarjoaa riittävän laskentatehon useiden antureiden samanaikaiseen käsittelyyn. Tietokoneen liitännät on hyödynnetty seuraavasti:

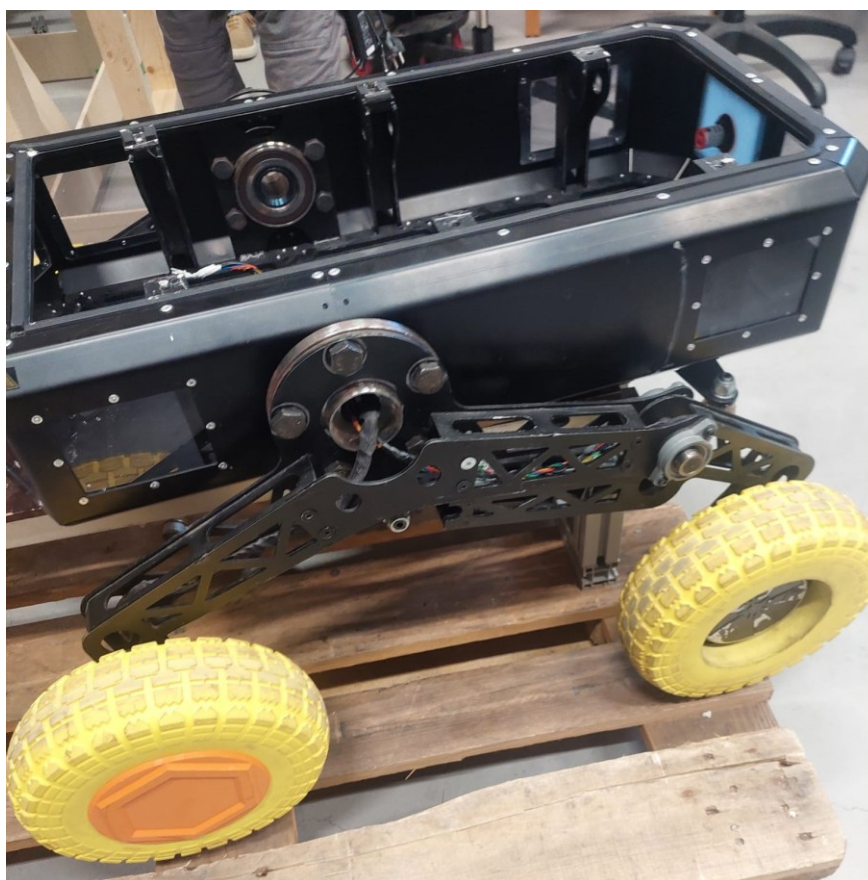
- Stereokamera on liitetty yhteen GMSL2 FAKRA-Z -liitännöistä, ja järjestelmään on mahdollista lisätä lisää kameroita tarpeen mukaan.
- IMU/GNSS-yksikkö kytketään USB 3.0 Type-A -liitäntään.
- Modbus RS-485 toteutetaan USB RS-485 -sovittimen kautta toisessa USB 3.0 -liitännässä.
- LiDAR-anturi liitetään Ethernet-liitäntään.
- HDMI-liitäntään on kytketty ”dummy plug”, jotta etäyhteyden kautta voidaan käyttää graafista käyttöliittymää ilman fyysistä näyttöä. [20.]

4.3 Runkorakenne

Tässä luvussa tarkastellaan mobiilirobotin mekaanista runkorakennetta ja siihen liittyviä rakenteita, kuten pyörätuennat, sähkökotelot, lisäkotelot sekä anturien kiinnitysrakenteet. Robotin runko perustuu Oinride Oy:n alkuperäiseen suunnitteluun, jota on Metropolia Ammattikorkeakoulussa muokattu ja täydennetty tätä mobiilirobottia varten. Tarkastelu keskittyy rakenteisiin, jotka vaikuttavat komponenttien sijoitteluun, asennettavuuteen ja järjestelmän kokonaisuuteen tässä työn toteuttamassa kokoonpanossa.

4.3.1 Pyörätuenta

Mobiilirobotin pyörätuentaratkaisu perustuu rocker-bogie -tyyppiseen rakenteeseen, jota käytetään yleisesti epätasaisessa maastossa liikkuvissa roboteissa. Alkuperäisessä suunnittelussa rakenne on toteutettu kuudelle moottoridulle pyörälle, mutta käytännön kokemusten perusteella kokoonpanossa on käytetty vain neljää moottoria. Kuusipyöräisen kokoonpanon havaittiin heikentävän robotin kääntyvyyttä, minkä vuoksi järjestelmää yksinkertaistettiin (Kuva 10).

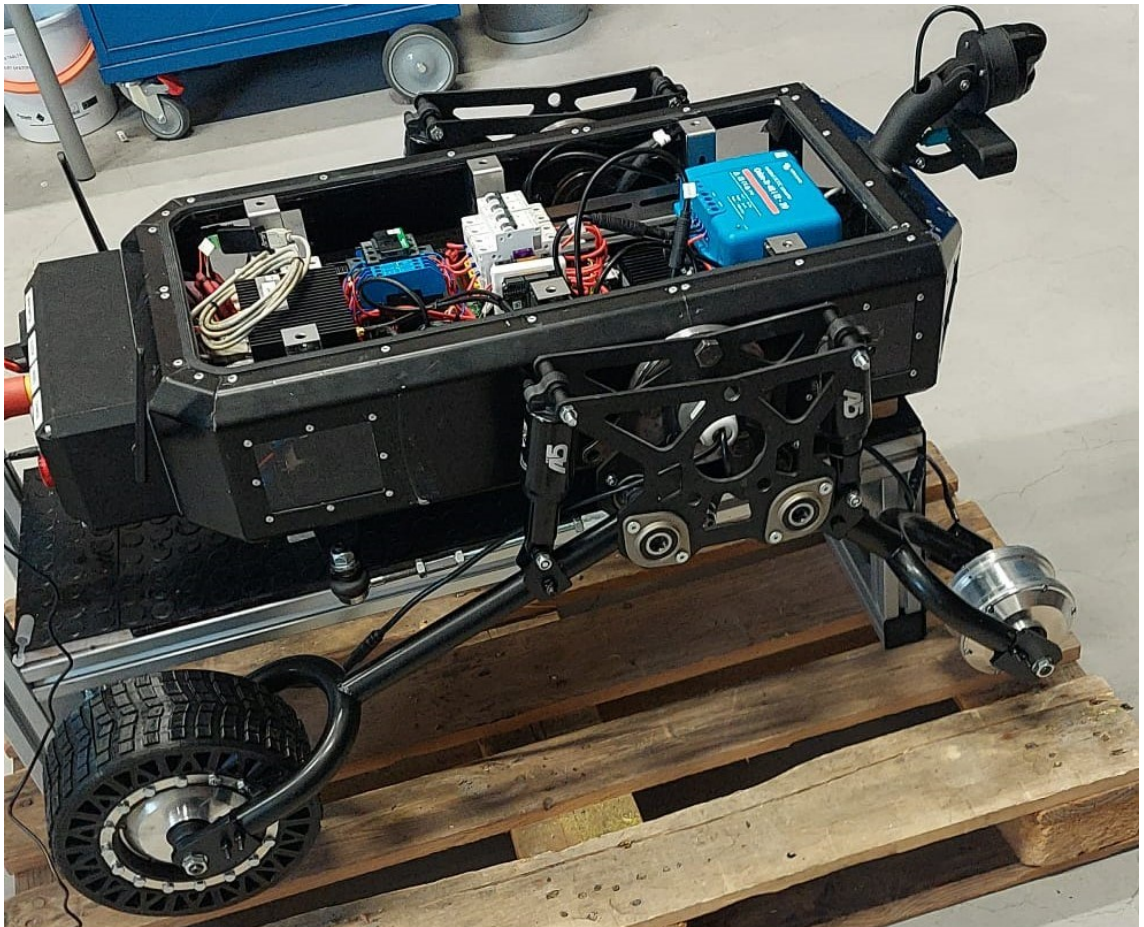


Kuva 10. Robotin pyörätuentaratkaisu sivulta kuvattuna.

Pyörien tukirakenteet on valmistettu laserleikatusta levyateriaalista, ja yksittäiset osat on liitetty toisiinsa hitsaamalla. Rakenne toimii alustana napamoottoreille sekä mahdollistaa pyörien kiinnittämisen rocker-bogie -

mekanismiin. Tässä työssä pyörätuenta ei muutettu rakenteellisesti, vaan se on hyödynnetty sellaisenaan osana mobiilirobotin kokoonpanoa.

Opinnäytetyön aikana oli käynnissä erillinen opiskelijaprojekti, jossa kehitettiin vaihtoehtoista pyörätuentaratkaisua. Kyseessä oli neljälle uudelle moottorille suunniteltu iskunvaimennettu putkirunkoinen malli (Kuva 11.), joka ei valmistumisen ajankohdan vuoksi ehtinyt koeajoon.



Kuva 11. Uusi pyörätuenta

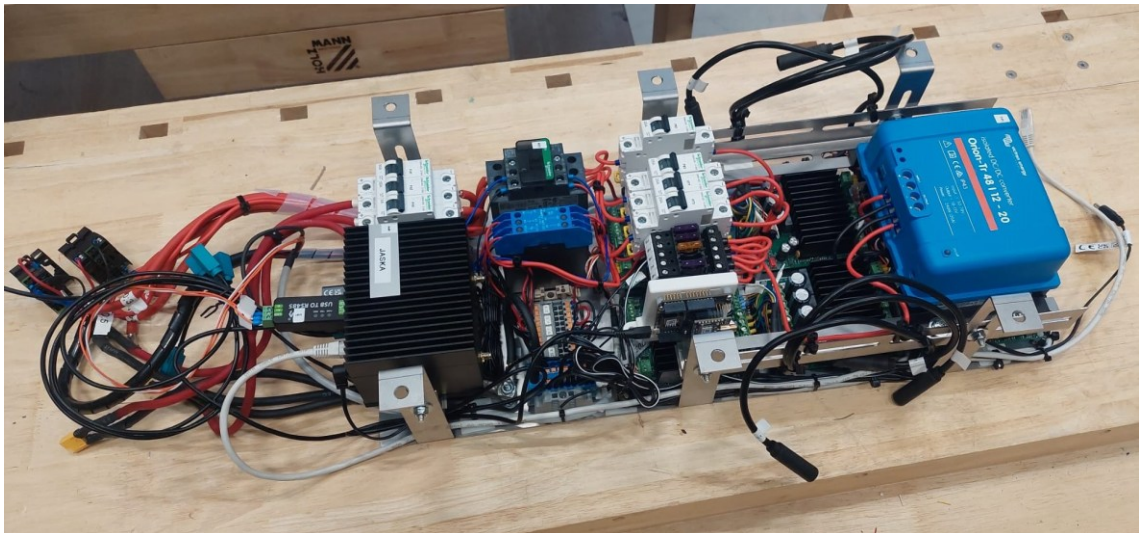
4.3.2 Sähkökotelointi

Mobiilirobotin runkorakenne toimii samalla sähkökomponenttien kotelona. Kotelo on valmistettu mustaksi pinnoitetusta alumiinilevystä, joka on taitettu ja niitattu kasaan. Saumoja on tiivistetty silikonilla, mutta tiivistys ei ole yhtenäinen,

joten kotelo ei ole IP-luokiteltu eikä tarjoa täyttä suojaa pölyä tai roiskevettä vastaan. Suojausta voidaan parantaa jatkokehityksessä.

Pyörien moottorikaapelit kulkevat kotelon sisään molemmilta puolilta laakeroitujen läpivientien kautta. Läpiviennit on varustettu 3D-tulostetuilla tulpilla ja M25-vedonpoistoholkilla kaapelien turvalliseen läpivienttiin.

Sähkökomponentit on kiinnitetty irrotettavaan asennuslevyyn, mikä helpottaa huoltoa, vianhakua ja kaapeloinnin muokkaamista. Kaikki komponentit on niputettu levyn yhteyteen, ja levy voidaan irrottaa, kun robotin kansi ja tarvittavat kaapelit on irrotettu (Kuva 12).

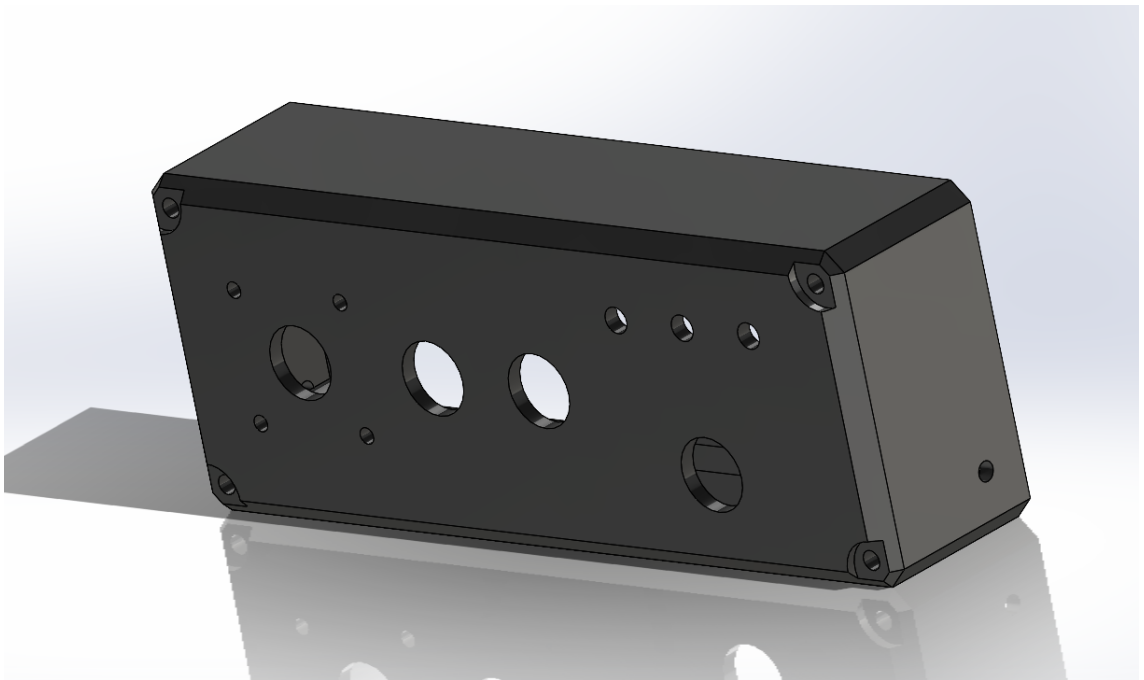


Kuva 12. Robotin sähköjärjestelmän asennuslevy ennen asennusta runkoon

Ilmanvaihtoa ja lämmönhallintaa ei huomioitu tässä prototyypivaiheessa, sillä tuuletusratkaisut olisivat vaatineet runkorakenteen muutoksia, ja tarkkaa tietoa komponenttien lämpenemisestä ei ollut saatavilla. Tulevissa kehityshankkeissa asia tulee ottaa huomioon, ja koteloon on jätetty varaus mahdolliselle tuulettimelle.

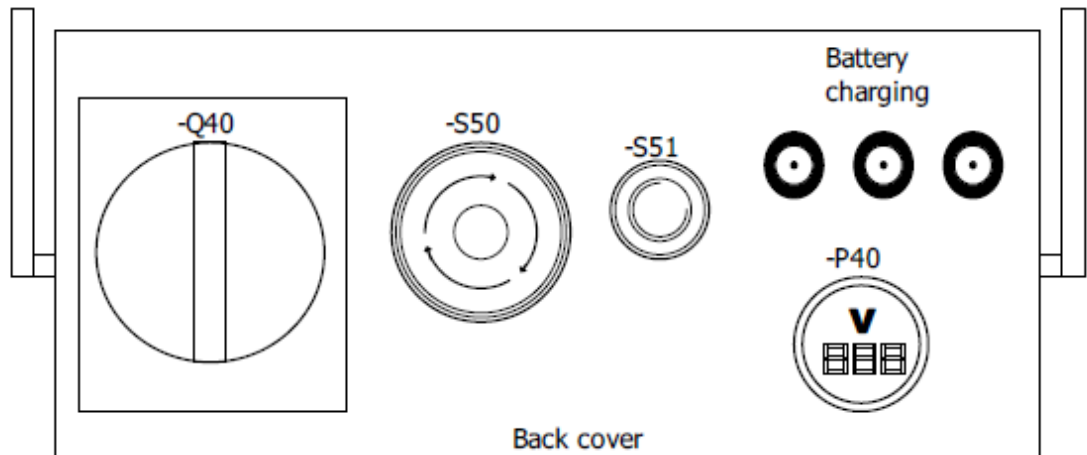
4.3.3 Lisäkotelo

Mobiilirobotin runkoon on lisätty erillinen lisäkotelo robotin takaosaan (Kuva 13). Lisäkotelo on toteutettu saman opiskelijaryhmän toimesta, joka suunnitteli ja valmisti sähkökomponenttien asennuslevyn. Lisäkotelon tarkoituksena on lisätä tilaa sähköisille ja käyttöön liittyville komponenteille sekä parantaa niiden saavutettavuutta.



Kuva 13. 3D-malli mobiilirobotin lisäkotelosta

Lisäkoteloon on sijoitettu robotin pääkytkin (-Q40), hätäseis-painike (-S50), kuittauspainike (-S51), jännitemittari (-P40) sekä akkujen latauspistokkeet (Kuva 14). Näiden komponenttien sijoittaminen erilliseen koteloon selkeyttää robotin käyttöliittymää ja mahdollistaa keskeisten hallintalaitteiden käytön ilman varsinaisen runkokotelon avaamista.



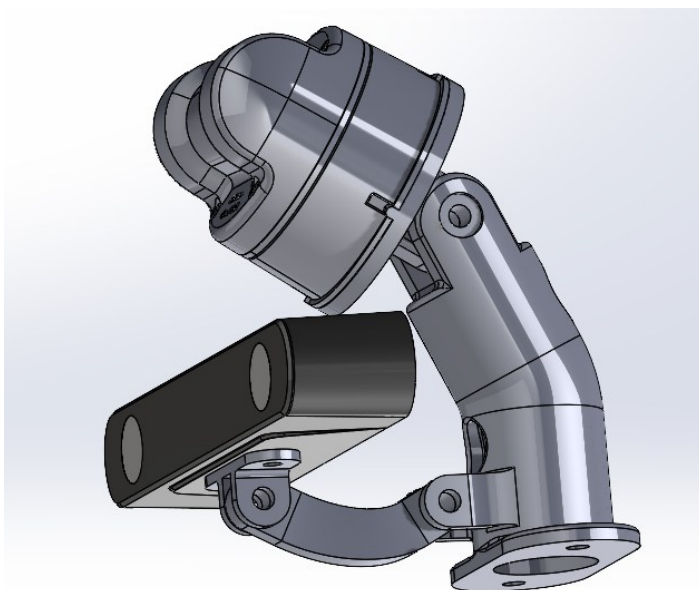
Kuva 14. Robotin takapaneelin komponenttien sijoittelukaavio

Lisäkoteloon on lisäksi toteutettu paikat tietokoneen WiFi-antennien asennusta varten. Antennien sijoittaminen kotelon ulkopuolelle parantaa langattoman yhteyden toimintaa verrattuna tilanteeseen, jossa antennit sijaitisivat metallisen runkorakenteen sisällä.

Lisäkotelo on mekaanisesti kiinteä osa robotin runkorakennetta ja se täydentää kokonaisuutta tarjoamalla lisätilaa tulevia laajennuksia ja muutoksia varten.

4.3.4 Anturijalusta

Mobiilirobotin keskeiset anturit on kiinnitetty erilliseen 3D-tulostettuun anturijalustaan (Kuva 15), joka sijaitsee robotin keulassa. Jalustaan on integroitu kiinnityspaikat sekä stereokameralle että LiDAR-anturille, mikä keskittää ympäristön havainnointiin käytettävät anturit yhteen rakenteeseen.



Kuva 15. Anturijalustan 3D-malli, jossa LiDAR ja kamera asennettuna

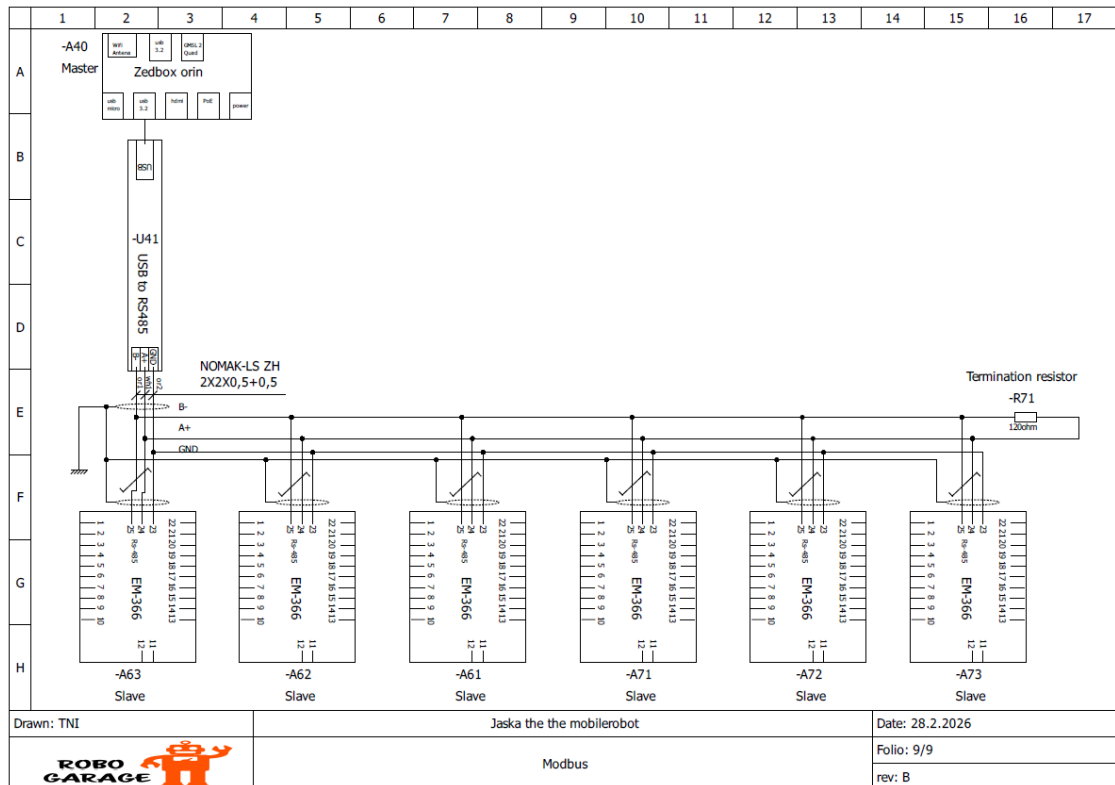
LiDAR-anturin kiinnitysalusta on käännettävissä pysty- ja vaaka-asentoon. Vaaka-asennossa LiDAR mittaa erityisesti etupuolen ja maanpinnan muodot, kun taas pystyasennossa se kattaa laajemmin myös robotin sivut ja taakse, vaikka eteenpäin suuntautuva havainnointi on tarkkuudeltaan rajallisempaa. Käännettävä mekanismi mahdollistaa eri mittaustarpeet ilman rakenteellisia muutoksia.

Anturien kaapelit kulkevat jalustan varren sisällä ja johdetaan robotin runkoon TET-läpiviennin kautta, mikä parantaa tiiveyttä ja suojaa kaapeleita teräviltä reunoilta.

4.4 Ohjelmisto

Mobiilirobotin ohjelmisto ohjaa robotin liikettä ja moottoreita sekä hallinnoi käyttäjän ja järjestelmän välistä vuorovaikutusta. Ohjelmisto ajetaan ZED Box Orin NX -tietokoneella ja se kommunikoi moottorinohjainten kanssa Modbus RTU RS-485 -väylän kautta (Kuva 16). Modbus on avoin ja laajasti käytetty teollinen viestintäprotokolla, joka perustuu isäntä-orjamalliin ja kysely-

vastausperiaatteeseen [21]. Modbus-väylä soveltuu erityisesti teollisiin ja sulautettuihin järjestelmiin hyvän häiriönsietokyvyn ja usean laitteen liitettävyyden ansiosta. Robotin tietokone toimii Modbus-isäntänä ja EM-366A-moottorinohjaimet orjalaitteina, joiden kautta moottoreiden toimintaa ohjataan ja ohjaimilta luetaan käyttö- ja tilatietoja. Kuva 16 havainnollistaa Modbus RTU RS-485 -väylän kytkentää, jossa tietokone toimii isäntälaitteena (master) ja moottorinohjaimet orjalaitteina (slave).



Kuva 16. Modbus-väylän kytkentäkuva

Antureiden, kuten LiDARin, stereokameran ja IMU/GNSS-yksikön datan varsinainen lukeminen rajattiin työn ulkopuolelle ajanpuutteen vuoksi. Ohjelmistoarkkitehtuuriin sisällytettiin kuitenkin valmiit rajapinnat ja loogiset paikat antureille sekä tuleville digitaalisille tuloille, jotta niiden käyttöönotto ja jatkokehitys onnistuvat myöhemmissä projekteissa ilman merkittäviä muutoksia ohjelmiston rakenteeseen.

Anturien simulointi toteutettiin ohjelmallisesti FastAPI-rajapinnan avulla, joka on Python-pohjainen kevyt web-rajapintakehys [22]. FastAPI mahdollistaa simuloitun anturidatan välittämisen ohjelmiston eri osien välillä samalla tavalla kuin fyysisiltä antureilta tulisi dataa. Simulaatio ajetaan Docker-kontissa, erillisessä ja eristetyssä virtuaalisessa ympäristössä [23]. Docker-kontin avulla voidaan varmistaa, että simulointi käyttää aina samoja riippuvuuksia ja ohjelmistoversioita, eikä se aiheuta ristiriitoja muiden järjestelmässä käytettävien ohjelmistojen kanssa. Kontti tekee simulaatiosta myös helposti siirrettävän ja uudelleenkäytettävän eri koneilla. Arvoja voidaan muuttaa ja lukea komentorivipohjaisella ohjelmalla, mikä mahdollistaa ohjauslogiikan testaamisen ilman fyysisiä antureita.

Moottoreiden testaukseen tarkoitettu ohjelma toteutettiin itsenäiseksi työkaluksi, jonka avulla moottoreiden, moottorinohjainten ja väyläkommunikaation toimintaa voidaan testata. Robotin käyttöohjelma muodostaa laajemman ohjelmistokokonaisuuden, joka sisältää moduuleja robotin ohjaamiseen ja käyttöliittymän hallintaan. Käyttöliittymä toteutettiin NiceGUI-kirjastoa hyödyntäen, jolloin graafinen käyttöliittymä voidaan ajaa suoraan tietokoneella tai etäyhteyden kautta verkkoselaimessa.

Docker- ja FastAPI-ratkaisut tarjoavat yhdessä turvallisen, modulaarisen ja hallitun tavan simuloida antureita ja kehittää robotin ohjelmistoa riippumatta fyysisistä laitteista.

Tässä luvussa kuvataan mobiilirobotin ohjelmiston rakenne ja keskeiset toiminnot. Alaluvuissa esitellään erikseen moottoreiden testausohjelma sekä laajempi käyttöohjelma, joka hallinnoi antureiden ja Modbus-väylän lukemista, anturidataa, robotin tiloja ja päätöksiä sekä ohjauskomentoja.

4.4.1 Moottoreiden testausohjelma

Moottoreiden testaus toteutettiin itsenäisellä Python-ohjelmalla, jossa käyttöliittymä on rakennettu NiceGUI-moduulilla. Ohjelmaa voidaan käyttää

SSH-yhteyden kautta selaimella, jolloin testaus onnistuu tietokoneella, tabletilla tai älypuhelimella. Testausohjelmassa hyödynnettiin Metropolia Ammattikorkeakoulun opiskelijan Samuli Ahokkaan luomia moduuleja ModbusDriver.py ja Config.py.

ModbusDriver-moduulin avulla mahdollistettiin kommunikaatio moottorinohjainten kanssa Modbus-väylän kautta, mukaan lukien moottorin pyörimissuunnan valinta, nopeusohjeen antaminen sekä ohjaimelta luettavien tietojen, kuten jännitteen, virran, jarruvirran, PWM-arvon ja hall-pulssitaajuuden, lukeminen. Moottorin kierrosnopeus laskettiin hall-antureilta saatavan pulssitaajuuden perusteella, kun moottorin napaparien määrä ja välityssuhde tunnettiin (kaava 1).

$$n = 60 \cdot \frac{f}{p} \cdot i \quad (1)$$

missä

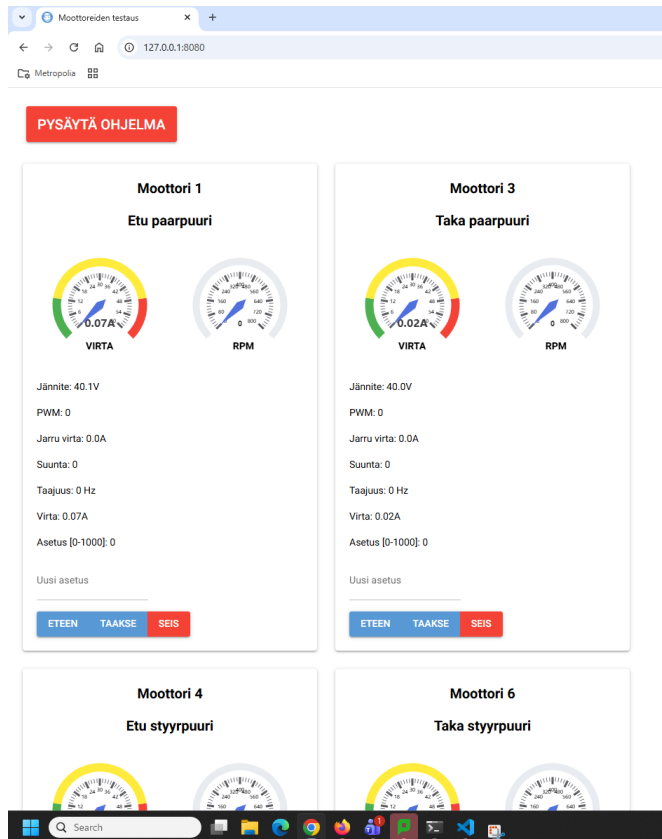
n = moottorin kierrosluku [1/min]

f = hall-pulssitaajuus [Hz]

p = napaparien määrä

i = välityssuhde

Käyttöliittymässä jokaiselle neljälle moottorille on oma hallintakortti, jossa ohjauspainikkeet ja sekunnin välein päivittyvät mittausarvot on esitetty (Kuva 17). Moottoreita voidaan ohjata yksitellen tai samanaikaisesti, ja ne voidaan pysäyttää yksittäin tai kaikki kerralla. Koko ohjelma voidaan tarvittaessa pysäyttää käyttöliittymästä. Näin moottoreiden toiminta voidaan testata ja vianetsintä suorittaa ilman robotin varsinaista liikettä.



Kuva 17. Moottoreiden testausohjelman käyttöliittymä

4.4.2 Robotin käyttöohjelman ohjelmistorakenne

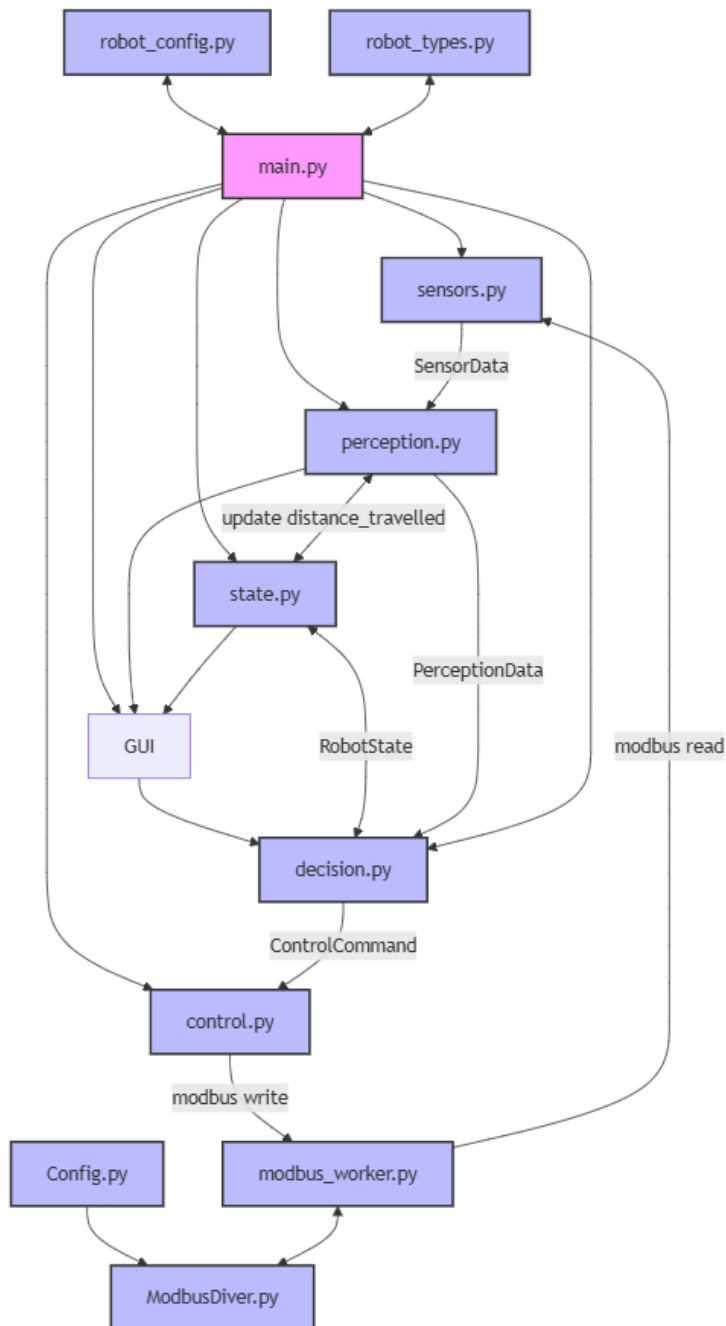
Varsinainen robotin käyttöohjelma toteutettiin modulaarisena Python-sovelluksena, joka koostuu pääohjelmasta, useista toiminnallisista moduuleista sekä selainpohjaisesta käyttöliittymästä. Ohjelmisto on suunniteltu siten, että havainnointi, päätöksenteko ja ohjaus erotettiin toisistaan selkeiksi kokonaisuuksiksi, mikä tukee jatkokehitystä ja testattavuutta.

Ohjelman suoritus käynnistyy main.py-pääohjelmasta, joka huolehtii robotin jatkuvasta ajosilmukasta, moduulien yhdistämisestä, tilan välittämisestä kierrokselta toiselle sekä käyttöliittymän käynnistämisestä, mutta ei sisällä varsinaista päätöksentekologiikkaa.

Ohjelmiston keskeiset moduulit ja niiden tehtävät ovat:

- **Sensors:** Vastaa anturidatan lukemisesta ja yhtenäistämistä muille moduuleille. Anturidata voidaan simuloida, ja käyttöliittymä sekä I/O-rajapinnat mahdollistavat testauksen ilman fyysisiä antureita.
- **Perception:** Jalostaa sensoridatan robottia ohjaaviksi havainnoiksi, kuten esteiden havaitsemiseksi, mitatun nopeuden määrittämiseksi sekä suuntiman arvioimiseksi. Moduuli muuntaa raakadatatusta helposti päätöksentekoon hyödynnettävät tiedot.
- **Decision:** Sisältää robotin päätöksentekologiikan ja tilakoneen. Moduuli valitsee robotille sopivan toiminnon havainnoista ja nykyisestä tilasta riippuen, esimerkiksi pysähtymisen, eteen ajon tai käännöksen vasemmalle tai oikealle.
- **Control:** Muuntaa päätökset konkreettisiksi moottori- ja ohjauskomennoiksi ja välittää ne moottorinohjaimille Modbus-väylän kautta. Moduuli käsittelee nopeus- ja kääntöarvot sekä huolehtii ohjausrajoituksen päivityksistä.
- **State:** Ylläpitää robotin sisäistä tilaa ja muistia, joka kulkee kierrokselta toiselle. Tallentaa muun muassa nykyisen toimintatilan, edellisen liikkeen, käännöksen suunnan sekä tilassa vietetyn ajan.
- **Robot_types:** Määrittelee ohjelmiston yhtenäiset tietorakenteet moduulien väliseen tiedonsiirtoon. Moduuli varmistaa, että kaikki moduulit käsittelevät dataa yhtenäisessä muodossa.
- **Robot_config:** Sisältää kaikki säädettävät parametrit ja raja-arvot, jotka eivät muutu ajonaikana, kuten renkaan koko, maksimi- ja perusnopeudet sekä esteen minimietäisyys. Luo lisäksi config.json-tiedoston niille muuttujille, joita voidaan säätää käyttöliittymästä.
- **ModbusDriver:** Sisältää varsinaiset Modbus-funktiot moottorinohjainten ohjaamiseen ja tietojen lukemiseen.
- **Modbus_worker:** Keskittää Modbus-väylän luku- ja kirjoituspyynnöt yhteen paikkaan, mikä helpottaa tiedon hallintaa ja estää ristiriidat väylän käytössä.
- **Config:** Sisältää väylän asetukset ja muut konfigurointiin liittyvät parametrit.
- **GUI (NiceGUI):** Selainpohjainen käyttöliittymä robotin ohjaukseen, ajonaikaisten parametrien säätöön ja robotin tilan sekä virheilmoitusten seurantaan.

Kuva 18 esittää ohjelmiston kokonaisrakenteen ja moduulien väliset suhteet vuokaaviona. [24.]



Kuva 18. Vuokaavio ohjelmistomodulien riippuvuuksista

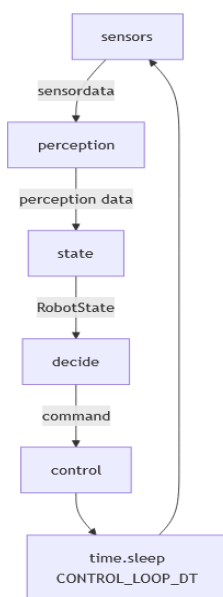
4.4.3 Main.py:n ja ohjaussilmukan toiminta

Keskeinen osa robotin käyttöohjelmaa on main.py, joka toimii ohjelmiston ajonaikaisena koordinaattorina. Main.py ei sisällä varsinaista robottikohtaista

ohjaus- tai päätöksentekologiikkaa, vaan sen tehtäväksi on määriteltävä ohjelmiston eri osien käynnistäminen ja yhdistäminen sekä ohjelman elinkaaren hallinta.

Ohjelman käynnistyessä main.py suorittaa ohjaussilmukan omassa säikeessään, käynnistää käyttöliittymän sekä aktivoi Modbus-väylän hallinnan modbus_worker-säikeessä. Main.py huolehtii myös ohjelman hallitusta sammuttamisesta, jolloin käynnissä olevat säikeet pysäytetään ja moottorit asetetaan tarvittaessa turvalliseen tilaan.

Ohjaussilmukka muodostaa robotin toiminnan rungon. Silmukka suoritetaan jatkuvana (Kuva 19), ja sen alussa Sensors-moduuli kerää käytettävissä olevan anturidatan. Kerätty data välitetään Perception-moduulille, jossa raakadata muunnetaan ohjausta tukeviksi havainnoiksi, kuten mitatuksi nopeudeksi ja esteiden havaitsemiseksi.



Kuva 19. Vuokaavio ohjaussilmukasta

Havaintojen perusteella State-moduuli päivittää robotin sisäisen tilan, joka säilyy silmukalta toiselle. Decision-moduuli muodostaa ohjauskäskyn robotin nykyisen tilan ja havaintojen perusteella.

Käsky välitetään Control-moduulille, joka muuntaa sen konkreettisiksi moottorikomennoiksi ja lähettää ne Modbus-väylän kautta moottorinohjaimille.

Silmukan lopussa suoritetaan määritelty aikaviive, joka pitää ohjaussilmukan ajotaajuuden vakiona ja prosessorin kuormituksen hallittuna. Tämän jälkeen silmukka alkaa uudelleen.

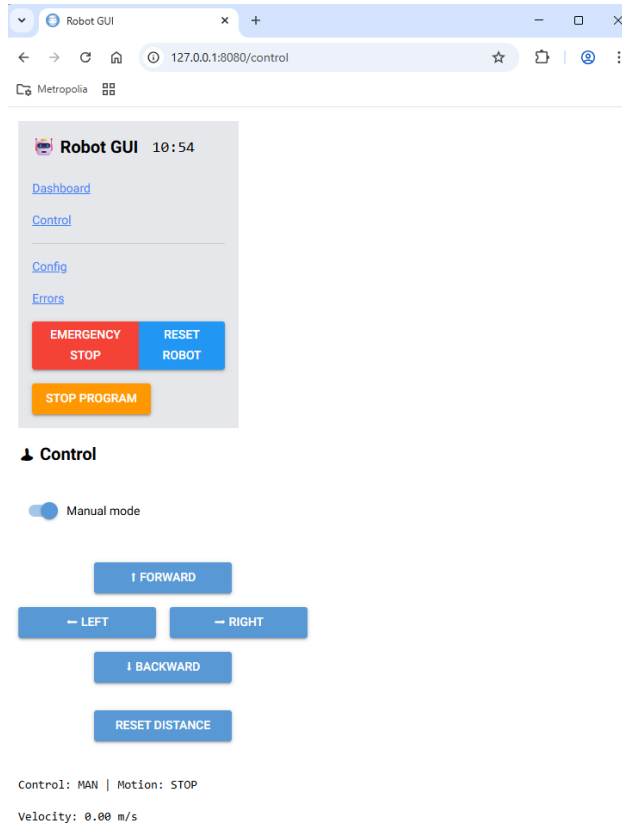
Ohjaussilmukka mahdollistaa sekä manuaalisen ohjauksen että automaatiotoimintojen toteuttamisen saman ohjelmistorungon sisällä. Ohjaustapa määräytyy robotin tilan perusteella ja voidaan vaihtaa ajonaikaisesti käyttöliittymän kautta. Main.py:n ja ohjaussilmukan toteutus on dokumentoitu projektin julkisessa versionhallinnassa. [25.]

4.4.4 Käyttöliittymä

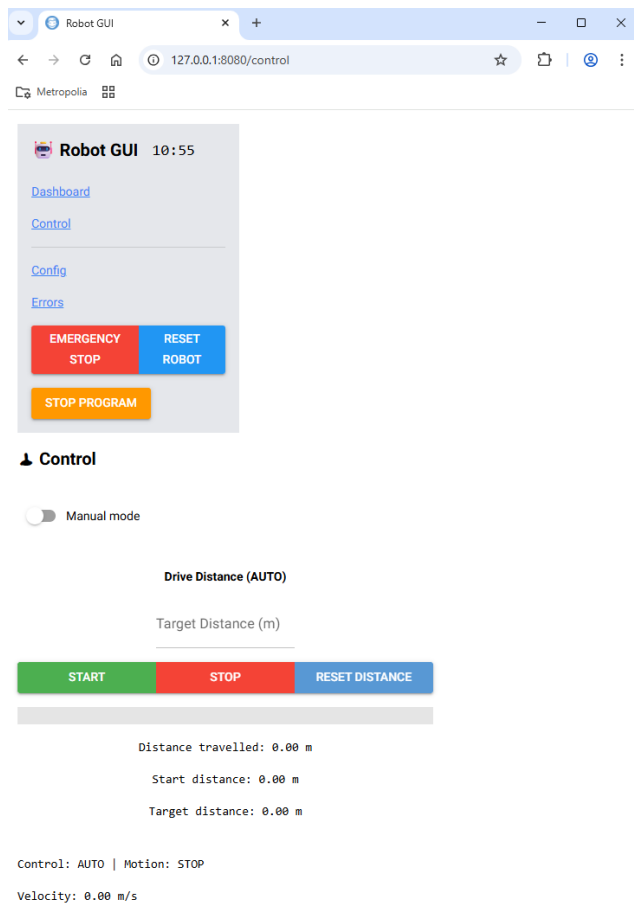
Robotin käyttöliittymä toteutettiin selainpohjaisena NiceGUI-kirjastoa hyödyntäen. Käyttöliittymää voidaan käyttää tietokoneella, tabletilla tai älypuhelimella etäyhteyden kautta.

Käyttöliittymä on jaettu eri näkymiin, jotka tukevat robotin hallintaa, tilan seuranta ja parametrien säätöä. Ohjausnäkyvässä robotin liikkumista voidaan

hallita manuaalisesti (Kuva 20) tai automaattiajotilassa (Kuva 21), ja käyttäjä voi valita ohjaustilan sekä käynnistää halutun liikkeen.

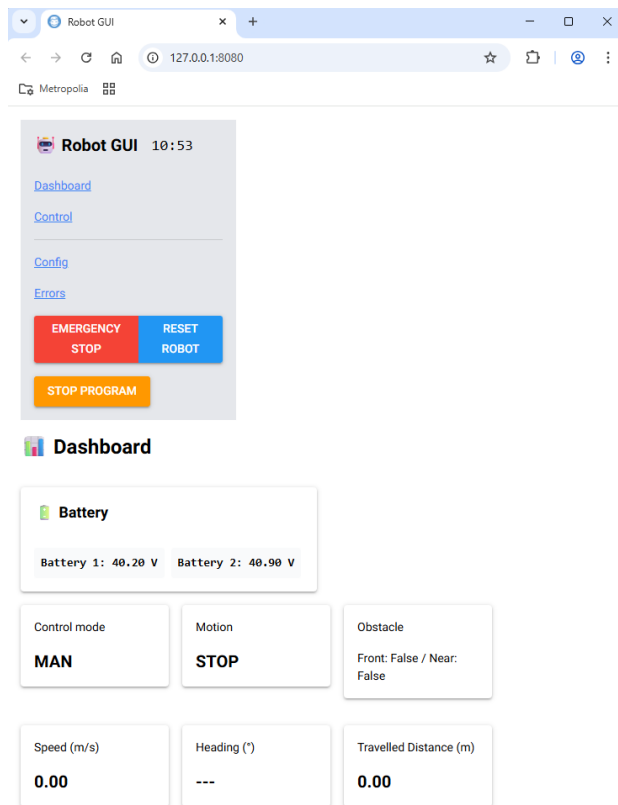


Kuva 20. Käyttöliittymän Control-näkymä manuaaltilassa



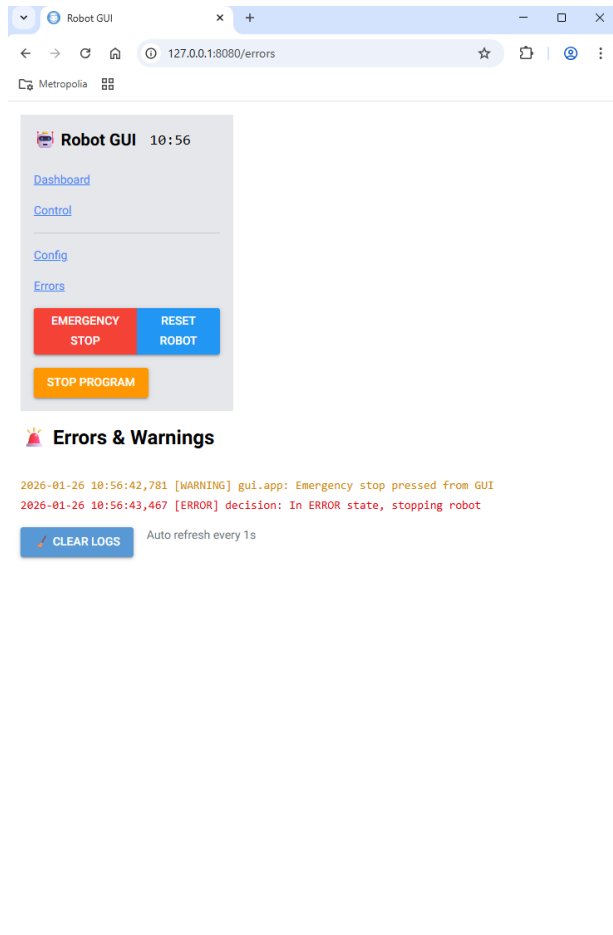
Kuva 21. Käyttöliittymän Control-näkymä automatiikkatilassa

Dashboard-näkymässä robotin tila esitetään reaaliajassa, esimerkiksi akkujen jännitteet, kuljettu matka, käynnissä oleva liike, nopeus ja valittu ohjaustyyppi (Kuva 22).



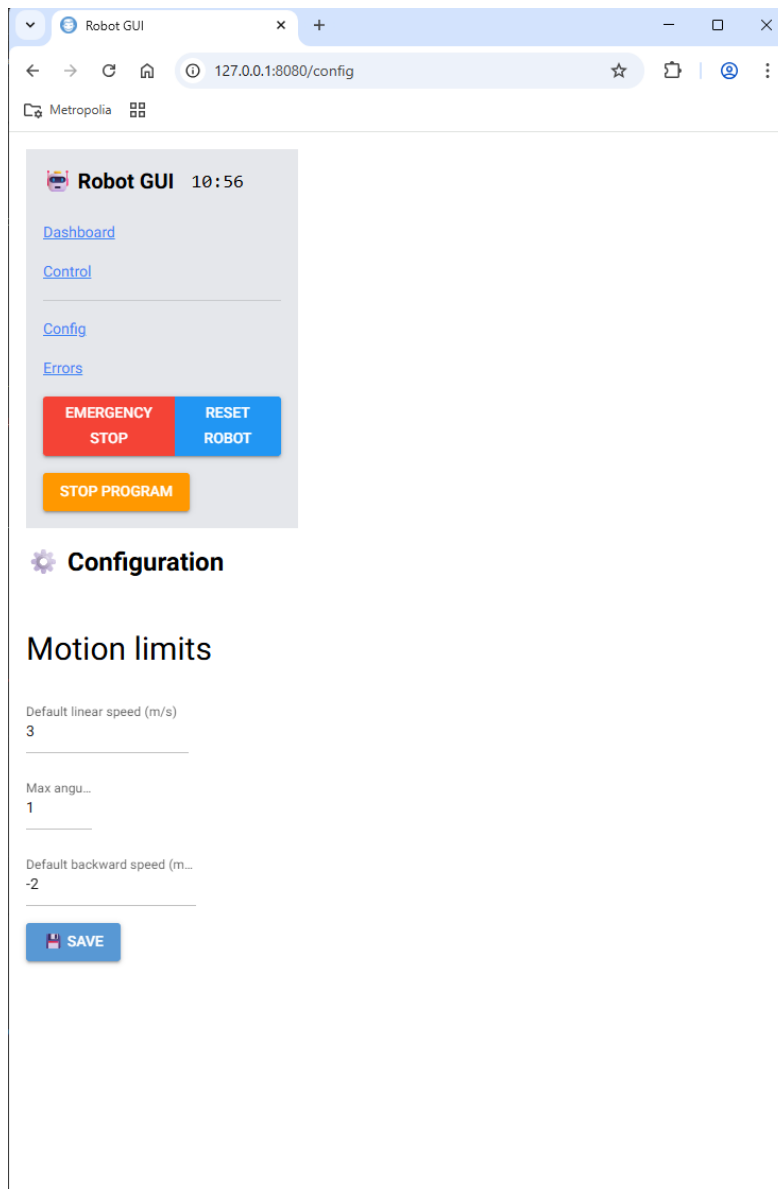
Kuva 22. Käyttöliittymän Dashboard-sivu

Virhe- ja hälytysnäkylässä esitetään järjestelmän hälytykset ja varoitukset yhdellä sivulla, jolloin robotin toiminnasta ja mahdollisista vioista saadaan kokonaiskuva (Kuva 23).



Kuva 23. Käyttöliittymän virhe- ja varoitussivu

Konfigurointinäkymässä voidaan säätää ajonaikaisia parametreja, kuten vakionopeuksia, jolloin robotin käyttäytymistä voidaan hienosäätää tarpeen mukaan (Kuva 24).



Kuva 24. Käyttöliittymän Configuration-sivu.

Käyttöliittymä hyödyntää ohjelmiston modulaarista rakennetta ja vastaanottaa tiedot reaaliajassa State- ja Perception-moduuleilta. Päätökset ja ohjaukset välitetään Control-moduulin kautta, mutta itse käyttöliittymä ei sisällä ohjauslogiikkaa, vaan toimii havainnointi- ja hallintapintana.

4.5 Robotin käyttö

Jaska-mobiilirobotin käyttö edellyttää yhteyden muodostamista robotin tietokoneeseen. Yhteys muodostetaan esimerkiksi SSH-yhteyden avulla, minkä jälkeen robotin ohjaussovellus käynnistetään manuaalisesti robotin tietokoneella. Käyttöliittymää käytetään erilliseltä päätelaitteelta, joka on samassa lähiverkossa robotin kanssa.

Robotti yhdistyy langattomaan Robo_Device_net_5G -verkkoon, jonka kautta käyttöliittymään päästään käsiksi.

Robotin käynnistys tapahtuu kytkemällä virta päälle robotin peräpaneelissa sijaitsevasta pääkytkimestä (MAIN SWITCH). Tämän jälkeen akkujen jännite ilmestyy jännitemittariin ja varmistetaan, että hätäseis-painike (E-STOP) on vapautettu. Kuittauspainikkeen merkkivalon syttyä hätäseis-piiri kuitataan painamalla kuittauspainiketta, jolloin jännite kytkeytyy moottorinohjaimille (Kuva 6). Kun robotin tietokone on käynnistynyt, järjestelmä on käyttövalmis.

Ohjelmistossa on havaittu USB-laitteiden alustukseen liittyvä ajoittainen ongelma. Tämän vuoksi suositellaan, että tietokoneen käynnistyksen jälkeen USB-laitteet irrotetaan ja kytketään uudelleen ennen ohjelmiston käynnistämistä.

Yhteys robotin tietokoneeseen muodostetaan esimerkiksi SSH-yhteyden avulla paikallisen tietokoneen komentoriviltä suorittamalla seuraava komento:

```
ssh user@192.168.123.175
```

Yhteyden muodostamisen jälkeen robotin ohjaussovellus voidaan käynnistää robotin tietokoneella.

Robotin ohjausohjelma käynnistetään suorittamalla robotin tietokoneen komentorivillä seuraava komento:

```
jaskagui
```

Komento käynnistää ohjelmiston sekä siihen liittyvän käyttöliittymän.

Käyttöliittymä avataan selaimella laitteella, joka on samassa verkossa kuin robotti. Selain ohjataan robotin IP-osoitteeseen porttiin 8080, esimerkiksi: [24.]

<http://192.168.123.175:8080>

Käyttöliittymän kautta robottia voidaan ohjata ja sen tilaa seurata reaaliaikaisesti. Control-sivulla käyttäjä valitsee liukukytkimen avulla manuaalisen tai automaattisen ohjaustilan.

Manuaalitulassa robottia ohjataan painikkeilla eteen- ja taaksepäin sekä kääntymällä vasemmalle tai oikealle. Liike tapahtuu niin kauan kuin painiketta pidetään painettuna, ja pysähtyy välittömästi painikkeen vapauttamisen jälkeen.

Automaattitulassa käyttäjä syöttää halutun ajomatkan metreinä ja käynnistää liikkeen start-painikkeella. Robotti etenee suoraviivaisesti määritellyn matkan. Liike voidaan pysäyttää kesken suorituksen, ja käynnistää uudelleen, jolloin robotti jatkaa kulkuaan, kunnes asetettu tavoitematka on saavutettu.

4.6 Testaus ja toimivuuden arviointi

Robotin ohjelmiston ja ohjausjärjestelmän toimivuutta testattiin vaiheittain kehitystyön aikana. Testauksen tavoitteena oli varmistaa moottorinohjausten, Modbus-väyläkommunikaation, ohjelmiston peruslogiikan ja käyttöliittymän toiminta ennen varsinaista robotin käyttöä. Testaus toteutettiin sekä fyysisellä laitteistolla että ohjelmallisten simulointien avulla.

4.6.1 Moottorien ja Modbus-väylän testaus

Testaus aloitettiin erillisellä moottorien testausohjelmalla, jonka tavoitteena oli ModbusDriver-moduulin toiminnan ymmärtäminen sekä Modbus RS-485 -väylän toimivuuden varmistaminen. Ohjelman avulla varmistettiin, että moottorinohjaimet vastaanottavat ohjauskäskyjä ja että moottoreilta voidaan lukea mittausrvoja, kuten jännite- ja virta-arvot. Lisäksi testattiin moottoreiden kytkennät ja pyörimissuunta.

Aluksi testauksessa käytettiin ainoastaan vanhoja moottoreita, sillä uusia moottoreita ei ollut vielä mahdollista liittää järjestelmään. Vanhoilla moottoreilla testattiin yksittäisten moottoreiden pyörimissuuntaa, nopeusohjeiden toteutumista sekä moottorinohjaimelta saatujen mittausrvojen järkevyyttä. Moottorien laskettua kierrosnopeutta verrattiin takometrillä mitattuihin arvoihin laskennan oikeellisuuden varmistamiseksi.

Myöhemmin uudet moottorit liitettiin sen verran, että niiden toimivuutta voitiin testata pyörittämällä niitä, mutta varsinaista ajoa ei voitu suorittaa, koska renkaiden ei ollut vielä asennettu. Alkuperäisessä oletuksessa kaikkien moottoreiden pyörimissuunta olisi voitu vaihtaa kytkentää muuttamalla yhdellä komennolla, ja vanhojen moottoreiden ylikuormittuminen oli herättänyt epäilyn niiden mahdollisesta viasta. Käytännössä samaa menetelmää ei voitu käyttää uusillakaan moottoreilla, joten suuntakomennot määritettiin ohjelmallisesti ja moottorit kytkettiin yhdenmukaisesti.

Lisäksi havaittiin ero vanhojen ja uusien moottoreiden laskettujen kierroslukuarvojen välillä. Vanhoissa moottoreissa, joissa ei ollut välityssuhdetta, laskettu kierrosluku vastasi hyvin todellista pyörimisnopeutta. Uusissa moottoreissa, joissa oli välityssuhde, laskettu kierrosluku oli aluksi liian suuri, mutta välityssuhteen huomioimisen jälkeen arvot vastasivat paremmin todellista pyörimisnopeutta. Moottorien virta-arvojen todettiin olevan suuntaa antavia, mutta riittäviä testaus- ja vianetsintätarkoituksiin.

4.6.2 Ohjelmiston toiminnan testaus ja tilakone

Robotin käyttöohjelman toimivuutta testattiin erityisesti tilakoneen ja päätöksentekologiikan osalta. Testauksessa varmistettiin, että robotti reagoi odotetusti eri tilanteisiin sekä manuaali- että automaattiajossa.

Estehavaintojen toiminta testattiin siten, että automaattiajossa robotti pysäytettiin esteen havaitsemisen yhteydessä, ja liike sallittiin vasta esteen poistuttua. Manuaaliajossa liike eteenpäin estetään esteen ollessa edessä. Mikäli este havaittiin lähellä, nopeutta hidastettiin sekä manuaali- että automaattitilassa.

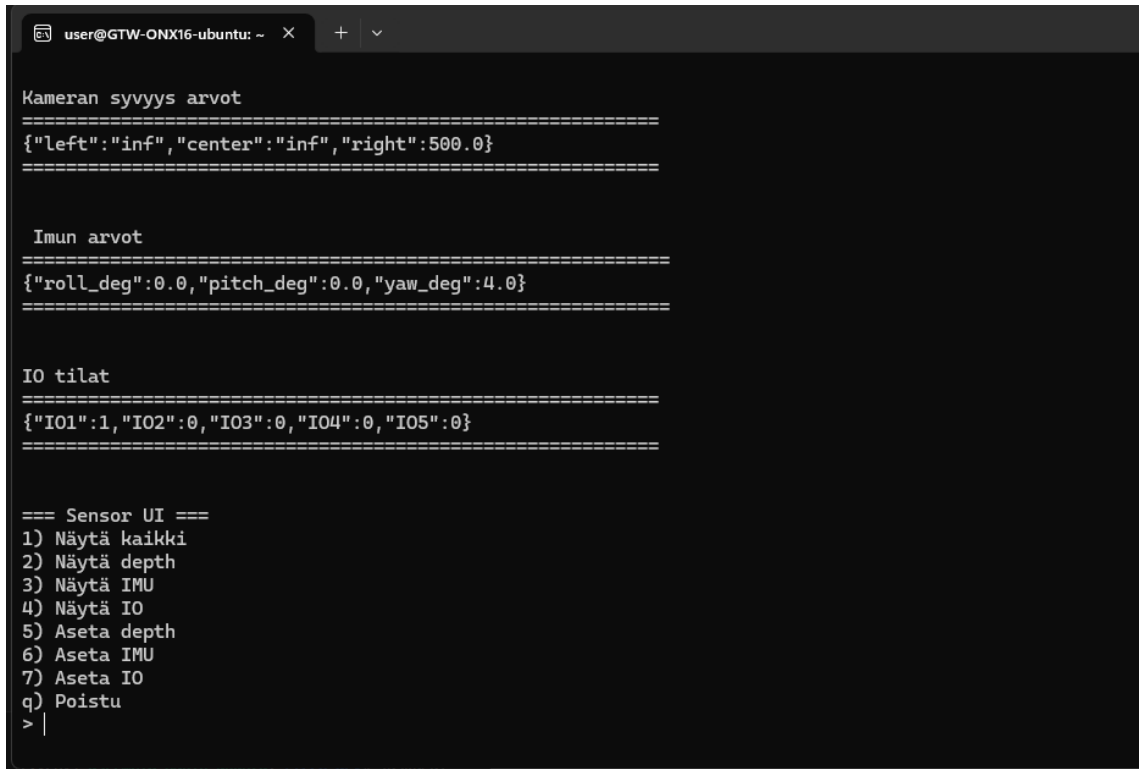
Hätäseis-painikkeen painaminen katkaisi moottoreiden ja ohjaimien virran, mutta sen signaalia ei voitu lukea ohjelmiston tilalogiikkaan, koska käytettävää IO-liitäntää ei ollut. Ohjelmistoon oli kuitenkin varattu paikat hätäseis- ja kuittauspainikkeiden signaaleille, ja niiden toiminta testattiin simuloidusti anturisimulaation avulla. Hätäseis-painikkeen simuloinnilla liike pysäytettiin ja järjestelmä siirrettiin virhetilaan, ja kuittauspainikkeen simuloinnilla järjestelmä palautettiin toimintavalmiiksi.

IMU/GNSS-yksikön testaus rajoittui suuntimatiedon näyttämiseen käyttöliittymän dashboard-näkymässä. Anturidatan varsinaista hyödyntämistä ohjauslogiikassa ei tässä työssä toteutettu.

4.6.3 Anturisimulaation testaus

Koska varsinaisten antureiden datan lukeminen ja hyödyntäminen rajattiin työn ulkopuolelle, testauksessa käytettiin anturisimulaattoria. Simulointi toteutettiin erillisessä Docker-kontissa robotin tietokoneella, ja kontti tuotti anturidataa FastAPI-rajapinnan kautta. Simuloituja anturiarvoja voitiin muokata ja lukea erillisellä komentorivipohjaisella `sensor_sim_ui` -ohjelmalla, jota ajettiin toisessa terminaalissa (Kuva 25).

Käyttöliittymä mahdollisti kaikkien anturiarvojen tarkastelun, yksittäisten anturien syötön muuttamisen sekä ohjauslogiikan ja tilakoneen testaamisen ilman fyysisiä antureita. Esimerkiksi, jos IO1:n tila muutettiin arvosta 1 arvoon 0, hätäseis-toiminnot aktivoituivat, ja jos kameran syvyysarvo laski alle 300, robotin eteenpäin ajo estyi. Näin moottorin ohjauslogiikkaa ja turvallisuustoimintoja voitiin testata hallitusti.



```

user@GTW-ONX16-ubuntu: ~
=====
Kameran syvyys arvot
=====
{"left": "inf", "center": "inf", "right": 500.0}
=====

Imun arvot
=====
{"roll_deg": 0.0, "pitch_deg": 0.0, "yaw_deg": 4.0}
=====

IO tilat
=====
{"IO1": 1, "IO2": 0, "IO3": 0, "IO4": 0, "IO5": 0}
=====

=== Sensor UI ===
1) Näytä kaikki
2) Näytä depth
3) Näytä IMU
4) Näytä IO
5) Aseta depth
6) Aseta IMU
7) Aseta IO
q) Poistu
> |

```

Kuva 25. sensor_sim_ui-anturisimulaattorin käyttöliittymä

4.6.4 Testauksen yhteenveto

Testauksessa moottorien ohjaus, Modbus-väyläkommunikaatio ja ohjelmiston perusrakenne todettiin toimiviksi. Moottoreiden testausohjelmaa havaittiin hyödylliseksi työkaluksi sekä käyttöönoton että vianetsinnän yhteydessä. Varsinaisen käyttöohjelman todettiin reagoivan oikein estehavaintoihin, käyttötilan muutoksiin ja turvallisuustoimintoihin.

Testauksessa ohjelmiston modulaarisen rakenteen ja anturisimulaation todettiin mahdollistavan testauksen etenemisen ilman kaikkien laitteistokomponenttien täysimääräistä käyttöönottoa. Tämä tukee ohjelmiston jatkokehitystä ja uusien anturien liittämistä tulevissa projekteissa.

5 Jatkokehitys ja kehityskohteet

Robotin ohjelmisto- ja laitteistoratkaisu todettiin perustoiminnoiltaan toimivaksi. Työn aikana havaittiin kuitenkin useita kehityskohteita ja rajoituksia, jotka ohjaavat järjestelmän jatkokehitystä. Kehitystarpeet voidaan jakaa kahteen pääalueeseen: fyysiseen laitteistoon ja ohjelmistoon.

5.1 Laitteistoon liittyvät jatkokehityskohteet

Mobiilirobotin laitteistoratkaisut todettiin perustoiminnaltaan toimiviksi. Kehitystyön aikana havaittiin useita kohtia, joissa rakenteita ja komponenttien sijoittelua olisi tarkoituksenmukaista kehittää edelleen. Havaitut kehitystarpeet liittyvät erityisesti kotelointiin, turvallisuuteen, huollettavuuteen ja järjestelmän laajennettavuuteen.

Nykyinen runkorakenne toimii myös sähköisenä kotelointina, mutta sen todettiin olevan rakenteeltaan melko ahdas ja vaikea saada täysin tiiviiksi. Osa saumoista vaatii lisätiivistystä, ja kokonaisuutena kotelointi hyötyisi uudelleensuunnittelusta. Jatkokehityksessä olisi perusteltua tarkastella väljempää rakennetta tai erilaista sähkökoteloa, joka helpottaisi komponenttien asentamista, huoltoa ja tiivistämistä pölyä ja roiskevettä vastaan.

Akuston sijoittelu samaan tilaan muun elektroniikan kanssa todettiin haastavaksi. Jatkokehityksen kannalta suositeltavaa olisi erottaa akusto omaksi moduulikseen, erilleen ohjauselektroniikasta. Tämä parantaisi sekä tilankäyttöä että turvallisuutta, sillä vikatilanteessa, kuten akuston ylikuumenemisessa tai syttymisessä, riskit muun elektroniikan vaurioitumiselle olisivat pienemmät.

Lämmönhallintaa ei tässä työssä suunniteltu tai toteutettu erikseen. Jatkossa on tarpeen selvittää, syntyykö järjestelmässä käytön aikana lämpökuormia, jotka edellyttävät ilmanvaihtoa, jäähdytystä tai lämpötilan seurantaa. Tämä korostuu erityisesti pidempiaikaisessa käytössä tai kuormittavissa olosuhteissa.

Ohjelmistoon on toteutettu valmius fyysisten hätäseis- ja kuittaussignaalien lukemiseen, mutta laitteistossa ei ollut käytettävissä soveltuvaa IO-liitäntää. Jatkokehityksessä tarvittavat IO-rajapinnat tulisi lisätä, jotta fyysiset hätäseis- ja kuittauspainikkeet voidaan liittää suoraan ohjelmiston tilakoneeseen ja turvallisuustoiminnot voidaan hallita kokonaisvaltaisesti.

Moottorinohjaimiin liittyen jarruvastusten tarve, tyyppi sekä mahdollinen sijoittaminen tulisi selvittää. Jarruvastuksia voidaan tarvita erityisesti kuormitustilanteissa, joissa moottorit toimivat generaattorina, ja niiden suunnittelulla järjestelmän luotettavuutta ja turvallisuutta voidaan parantaa.

Nykyisessä toteutuksessa kutakin akkua ladataan erillisestä liittimestä, mikä tekee latauksesta epäkäytännöllistä. Jatkokehityksessä voidaan suunnitella latauspiiri, jonka avulla kaikki akut voidaan ladata tasaisesti yhdestä latausliitännästä. Fyysistä jännitemittaria voidaan kehittää siten, että eri akkujen jännitteitä voidaan tarkastella valintakytkimen avulla. Tämä parannus voidaan toteuttaa erillisenä ja melko yksinkertaisena muutoksena pienellä kytkennällisellä muutoksella (Liite 2), ilman laajempaa elektroniikan uudelleensuunnittelua.

Turvallisuuden parantamiseksi testiajoissa hätäseis-piiriin voidaan lisätä talutushihnatyyppinen ratkaisu. Tällöin hätäseis laukeaa vetämällä hihnasta, ja robotti voidaan pysäyttää myös fyysisellä voimalla. Ratkaisu olisi hyödyllinen lisä erityisesti kokeellisessa käytössä ja kehitysvaiheen testauksessa.

Renkaiden kehittäminen on tunnistettu yhdeksi keskeiseksi jatkokehityskohteeksi, ja niiden parissa on tehty kehitystyötä myös aiemmissa projekteissa. Renkaiden ominaisuuksilla vaikutetaan merkittävästi robotin ajettavuuteen, pidon hallintaan ja kokonaiskäyttämiseen, minkä vuoksi renkaiden kehittämistä ja erilaisten rengasratkaisujen testaamista on perusteltua jatkaa tulevissa projekteissa.

5.2 Ohjelmiston jatkokehitys

Ohjelmiston osalta keskeisimmät jatkokehitystarpeet liittyvät anturidatan varsinaiseen hyödyntämiseen, autonomian laajentamiseen sekä robotin ympäristötietoisuuden parantamiseen. Tässä työssä ohjelmistoarkkitehtuuri rakennettiin siten, että antureille ja lisätoiminnoille on varattu rakenteelliset paikat, mutta varsinaista anturidatan lukemista ja hyödyntämistä ei toteutettu ajanpuutteen vuoksi.

Ensimmäinen selkeä jatkokehityskohde on anturidatan lukeminen ja käsittely. Ohjelmistoon on toteutettu sensors- ja perception-moduulit, mutta niiden toiminta perustui tässä työssä simuloituun dataan. Fyysiset anturit, kuten LiDAR, stereokamera ja IMU, voitaisiin liittää osaksi ohjelmiston päätöksentekoa, mikä mahdollistaisi järjestelmän todellisen ympäristöhavainnoinnin.

Stereokameran osalta jatkokehitystä tarvitaan erityisesti syvyystiedon hyödyntämisessä. Ohjelmistossa on varattu paikka stereokameran datalle, mutta syvyysdatan välittäminen FastAPI-rajapinnan kautta Docker-kontissa osoittautui haasteelliseksi. Jatkokehityksessä stereokameran syvyystietoa voitaisiin hyödyntää esteiden tunnistamiseen, etäisyyksien arviointiin ja ympäristön mallintamiseen. Lisäksi kameran kuvavirran näyttäminen käyttöliittymässä parantaisi robotin käytettävyyttä ja tilannetietoisuutta.

IMU-anturi on kytketty laitteistoon, mutta sen dataa ei tässä työssä luettu eikä hyödynnetty. Jatkokehityksessä IMU:n tarjoamaa suunta-, kiihtyvyyss- ja kulmanopeustietoa voitaisiin hyödyntää robotin liikkeen tarkempaan seurantaan, ajosuunnan vakauttamiseen sekä odometrian tukena.

Odometrian ja mahdollisesti SLAM-toiminnallisuuden lisääminen muodostaa merkittävän jatkokehityssuunnan. Moottoreiden pyörintätiedon, IMU-datan ja mahdollisen anturidatan yhdistämisen avulla robotin sijaintia ja liikkumista suhteessa ympäristöön voitaisiin arvioida. Tämä loisi perustan kartan muodostamiselle ja itsenäisemmälle navigoinnille.

Autonomian tasoa voidaan lisätä vaiheittain, kun anturidata integroidaan luotettavasti ohjelmistoon. Laajennuksiin voidaan sisällyttää esimerkiksi esteiden kiertäminen, reitin suunnittelu, erilaisten navigointitehtävien suorittaminen sekä tilannekohtaisen käyttäytymisen laajentaminen. Modulaarinen ohjelmistorakenne mahdollistaa tällaiset laajennukset ilman merkittäviä muutoksia olemassa olevaan koodipohjaan.

Yksi mahdollinen jatkokehityssuunta robotin ohjelmistolle on Robot Operating System (ROS) -ympäristön käyttöönotto. ROS on robotiikan ohjelmistokehityksessä laajasti käytetty kehitysalusta, joka tarjoaa valmiita työkaluja esimerkiksi anturidatan käsittelyyn, paikannukseen, kartoitukseen sekä autonomiseen navigointiin. [26.]

Tässä työssä robotin ohjelmisto toteutettiin kevyempänä modulaarisena järjestelmänä, jossa toiminnot on jaettu erillisiin ohjelmistomoduuleihin. Ratkaisu soveltuu hyvin kehitysalustan perustoimintoihin ja mahdollistaa järjestelmän laajentamisen vaiheittain.

Jatkokehityksessä järjestelmä voitaisiin kuitenkin integroida ROS-ympäristöön, jolloin robotin navigointiin, paikannukseen ja ympäristön mallintamiseen voitaisiin hyödyntää valmiita ROS-kirjastoja ja -paketteja. Tämä helpottaisi erityisesti anturidatan yhdistämistä sekä autonomisten toimintojen kehittämistä.

Toisaalta ROS-järjestelmän käyttöönotto lisäisi ohjelmiston monimutkaisuutta ja edellyttäisi nykyisen ohjelmistorakenteen osittaista uudelleensuunnittelua. Tästä syystä tässä työssä päädyttiin toteuttamaan kevyempi ohjelmistoarkkitehtuuri, joka soveltuu hyvin opetuskäyttöön ja kehitysalustaksi. Tulevissa projekteissa ROS-pohjainen toteutus voisi kuitenkin tarjota tehokkaamman pohjan edistyneemmälle mobiilirobotiikan tutkimukselle ja autonomisille toiminnoille.

6 Yhteenveto

Tässä opinnäytetyössä pyrittiin palauttamaan puretusta mobiilirobotista toimiva mobiilirobotiikan kehitysalusta, jota voidaan hyödyntää opetuksessa ja jatkokehityksessä. Alusta toteutettiin siten, että siinä on manuaalinen ohjaus sekä yksinkertainen ensimmäisen asteen autonomia, ja sen avulla opiskelijoiden osaamista mobiilirobotiikan eri osa-alueilla voidaan kehittää joko yksittäisiin järjestelmän osiin keskittyen tai kokonaisuutta tarkastellen. Työssä hahmotettiin myös mobiilirobotiikan tilaa vuonna 2025 ja luotiin pohja järjestelmän tulevalle kehitykselle.

Työn tuloksena kehitettiin toimiva mobiilirobotiikan kehitysalusta, jossa on selkeä laitteisto- ja ohjelmistorakenne, dokumentoidut sähköratkaisut sekä modulaarinen ohjelmistorunko. Robotin moottoreita ja Modbus-väyläkommunikaatiota testattiin erillisellä testausohjelmalla, ja varsinaista käyttöohjelmaa testattiin sekä fyysisellä laitteistolla että anturisimulaation avulla. Vanhoilla moottoreilla robottia ohjattiin rajoitetusti, ja uusien moottoreiden toimivuutta testattiin pyörittämällä niitä robotin ollessa pukkien päällä. Lisäksi ohjelmiston reagoitua erilaisiin anturitiloihin ja poikkeustilanteisiin testattiin hallitusti simulaation avulla.

Työhön liittyi myös selkeitä rajoitteita. Robotin ajotestaus jäi osittain puutteelliseksi, koska vanhojen moottoreiden toiminta ei ollut luotettavaa ja uusien moottoreiden käyttöönotto viivästyi mekaanisten ratkaisujen sekä renkaiden puuttumisen vuoksi. Anturidatan hyödyntäminen jäi rajalliseksi: stereokameran syvyystiedon lukeminen ja IMU-datan käyttö jäivät toteuttamatta ajan ja osaamisen rajallisuuden vuoksi. Erityisesti kameran käyttö olisi edellyttänyt syvällisempää Docker-osaamista, johon ei ollut mahdollista perehtyä riittävästi muun työn ohessa.

Opinnäytetyö tarjosi merkittävän oppimiskokemuksen mobiilirobotiikan kokonaisuuksista. Työn aikana ymmärrystä modulaarisen laitteisto- ja ohjelmistoarkkitehtuurin merkityksestä syvennettiin sekä havainnollistettiin,

miten robottijärjestelmä kannattaa jakaa selkeisiin kokonaisuuksiin, kuten anturointiin, havainnointiin, tilanhallintaan, päätöksentekoon ja ohjaukseen. Lisäksi Docker-konttien perusteita ja konttipohjaisen lähestymistavan hyötyjä monimutkaisten järjestelmien kehityksessä käsiteltiin. Projektityöskentely, vianetsintä ja dokumentoinnin merkitys korostuivat myös työn aikana.

Kehitetty mobiilirobotin alusta tarjoaa hyvän pohjan jatkokehitykselle ja uusille opiskelijaprojekteille. Robotiikkaosaamisen ja siihen liittyvän innovoinnin tarve on merkittävä Suomessa, ja kehitysalustaa voidaan hyödyntää eri koulutusohjelmien johdanto-, innovaatio- ja opinnäytetyöprojekteissa. Lisäksi alusta mahdollistaa Robo Garage Clubin kehittää robottia edelleen ja osallistua tulevaisuudessa esimerkiksi European Rover Challenge -tyyppisiin kilpailuihin.

Lähteet

- 1 Nehmzow, Ulrich. 2003. Mobile Robotics: A Practical Introduction. 2nd Edition. London: Springer-Verlag.
- 2 Röning, Juha. 2023. Liikkuvat robotit. Teoksessa Välimäki, Kari & Niemelä Marketta (toim.). Teollisuuden robotiikka. Helsinki: Suomen Robotiikkayhdistys.
- 3 Automated Guided Vehicles. Verkkoaineisto. Mitsubishi Logisnext Europe Oy. <<https://www.mitsubishilogisnexteurope.fi/fi/>>. Luettu 5.1.2026.
- 4 Saros Z70. Verkkoaineisto. Roborock. <<https://fi.roborock.com/pages/roborock-saros-z70#section6>>. Luettu 5.1.2026.
- 5 Liljaniemi, Antti. 2022. Autonominen robotiikka. Opintomateriaali. Big-Flash
- 6 Hutri ym. 2025. Mobiilirobotiikan mahdollisuudet. Automaatioväylä 5/2025, s. 30-32. (Suomenkielinen lehtiartikkeli)
- 7 Lehtiniitty, Markus. 2025. Verkkoaineisto. Mobiili.fi. <<https://mobiili.fi/2025/01/03/s-ryhman-kayttamat-starshipin-kuljetusrobotit-kulkeneet-jo-yli-miljoonaa-kilometria-suomessa/>>. 3.1.2025. Luettu 21.3.2026.
- 8 Starship Technologies ja Wolt aloittavat päivittäistavaroiden robottikuljetukset asiakkaille. 2025. Verkkoaineisto. Wolt. <<https://press.wolt.com/fi-FI/246778-starship-technologies-ja-wolt-aloittavat-paivittaistavaroiden-robottikuljetukset-asiakkaille/>>. 12.2.2025. Luettu 21.3.2026.
- 9 Vainio, Olli. 2023. Suomessa kehitetty taistelurobotti tuhoaa panssarivaunuja. Verkkoaineisto. Tivi. <<https://www-tivi-fi.ezproxy.metropolia.fi/uutiset/a/6fbd9a1a-dc16-4ed4-b918-0c0665b5d20c>>. 1.1.2023. Luettu 21.3.2026
- 10 Stasevskij, Justas. 2024. Ukraina uskoo sotarobottien vallankumoukseen. Verkkoaineisto. Yle. <<https://yle.fi/a/74-20116816>>. 24.10.2024. Luettu 21.3.2026.
- 11 Mobiilirobotit tehostavat kuljetuksia myös vanhoissa sairaaloissa. Verkkoaineisto. Sweco.

- <<https://www.sweco.fi/ajankohtaista/uutiset/mobiilirobotit-tehostavat-kuljetuksia-myos-vanhoissa-sairaaloissa/>>. Luettu 21.3. 2026.
- 12 Uutiset. Verkkoaineisto. Dimalog.
<<https://www.dimalog.com/blog/?lang=fi>>. Luettu 21.3.2026
 - 13 Mobiilirobotiikka. Verkkoaineisto. Probot.
<<https://probot.fi/mobiilirobotiikka>>. Luettu 21.3.2026
 - 14 Mobile Robotics Market (2025 - 2030). Verkkoaineisto. Grand view research <<https://www.grandviewresearch.com/industry-analysis/mobile-robotics-market>>. Luettu 22.2.2026
 - 15 Design elements. 2021. PDF. *Technical pocket guide*. 3., uudistettu painos. Herzogenaurach: Schaeffler Technologies AG & Co. KG. Luettu 2.1.2026.
 - 16 Hutri, Niko. 2023. Silmät Mobiilirobotille. Verkkoaineisto. Big Flash.
<https://bigflash.metropolia.fi/blogi_oinride/>. 8.6.2023. Luettu 4.2.2026
 - 17 Tekninen opas nro 8 - Sähköinen jarrutus. 2001. Verkkoaineisto. ABB Automation Group Ltd.
<<https://search.abb.com/library/Download.aspx?DocumentID=3BFE64455567&LanguageCode=fi&DocumentPartId=1&Action=Launch>>. 15.3.2001. Luettu 6.2.2026
 - 18 EM-366A Brushless DC-Motor Driver with Modbus RS-485. Verkkoaineisto. Electromen Oy. <<https://electromen.com/fi/motor-controllers/brushless-dc-motor-controllers/em-a-brushless-dc-motor-driver-with-modbus-rs>>. Luettu 6.2.2026.
 - 19 ZED Box - Embedded AI Computer with NVIDIAN Jetson. 2026. Verkkoaineisto. StereoLabs. <<https://www.stereolabs.com/en-fi/series/zed-box>>. Luettu 9.2.2026.
 - 20 Zed Box - Datasheet. 2025. Verkkoaineisto. StereoLabs.
<<https://cdn.sanity.io/files/s18ewfw4/staging/f072216c5fb87ecc593104ab3ec6fd2a4950f5fc.pdf/ZED%20Box%20-%20Datasheet%20-%20v4.pdf>>. 2025. Luettu 9.2026.
 - 21 What is Modbus?. Verkkoaineisto. Modbus Organization.
<<https://www.modbus.org/faq>>. Luettu 2.32026
 - 22 FastAPI. Verkkoaineisto. tiangolo.
<<https://fastapi.tiangolo.com/#additional-optional-dependencies>>. Luettu 8.3.2026

- 23 What is Container?. Verkkoaineisto. Docker inc.
<<https://www.docker.com/resources/what-container/>>. Luettu 8.3.2026
- 24 Nikkola, Tero. 2026. Jaska_Automation. Verkkoaineisto. GitHub.
<https://github.com/TNuppi/Jaska_Automation>. Päivitetty 29.1.2026.
Luettu 12.2.2026
- 25 Nikkola, Tero. 2026. Jaska_Automation - main.md. Verkkoaineisto.
GitHub.
<https://github.com/TNuppi/Jaska_Automation/blob/d7dfa34f453058c6a5d460abb5c63a8dc23e769c/docs/main.md>. Päivitetty 29.1.2026. Luettu 12.2.2026.
- 26 Why ROS?. Verkkoaineisto. Open Robotics.
<<https://www.ros.org/blog/why-ros/>>. Luettu 7.3.2026.

Sähkökuvat

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	Label	Function	Textual description	Article number	Manufacturer	Folio number	Position										
	-A40	Computer	Zedbox OhnX		Stereolabs	4/9	E16										
	-A41	IMU +GNSS	IMU+GNSS	MTI-7-0I-DK	Xsens	4/9	E12										
B	-A61	Motor controller	EM-366 BRUSHLESS DC-MOTOR DRIVER 12-48V 30/25A		ELECTROMEN Oy	6/9	D6										
	-A62	Motor controller	EM-366 BRUSHLESS DC-MOTOR DRIVER 12-48V 30/25A		ELECTROMEN Oy	6/9	D11										
	-A63	Motor controller	EM-366 BRUSHLESS DC-MOTOR DRIVER 12-48V 30/25A		ELECTROMEN Oy	6/9	D16										
	-A71	Motor Controller	EM-366 BRUSHLESS DC-MOTOR DRIVER 12-48V 30/25A		ELECTROMEN Oy	7/9	D5										
C	-A72	Motor Controller	EM-366 BRUSHLESS DC-MOTOR DRIVER 12-48V 30/25A		ELECTROMEN Oy	7/9	D10										
	-A73	Motor Controller	EM-366 BRUSHLESS DC-MOTOR DRIVER 12-48V 30/25A		ELECTROMEN Oy	7/9	D16										
	-B41	Lidar	Unitree 4D lidar-L2		Unitree	4/9	E9										
D	-B42	Camera	Zed X		Stereolabs	4/9	D16										
	-F41	Miniature circuit breaker	C60H DC - 1p- 50 A-C	A9N61518	Schneider Electric	4/9	A3										
	-F42	Miniature circuit breaker	C60H DC - 1p- 50 A-C	A9N61518	Schneider Electric	4/9	B3										
E	-F43	Miniature circuit breaker	C60H DC - 1p- 5 A-C	A9N61505	Schneider Electric	4/9	B3										
	-F44	Fuse	UK 6-FSJ/C + ATO2A	3118203	PHOENIX CONTACT	4/9	F3										
	-F45	Fuse	UK 6-FSJ/C + ATO1A	3118203	PHOENIX CONTACT	4/9	F3										
F	-F46	Fuse	UK 6-FSJ/C + ATO5A	3118203	PHOENIX CONTACT	4/9	G3										
	-F47	Fuse	UK 6-FSJ/C + ATO5A	3118203	PHOENIX CONTACT	4/9	H3										
	-F48	Fuse	UK 6-FSJ/C + ATO5A	3118203	PHOENIX CONTACT	4/9	H3										
G	-F61	Miniature circuit breaker	C60H DC - 1p- 25 A-C	A9N61513	Schneider Electric	6/9	E3										
	-F62	Miniature circuit breaker	C60H DC - 1p- 25 A-C	A9N61513	Schneider Electric	6/9	E8										
	-F63	Miniature circuit breaker	C60H DC - 1p- 25 A-C	A9N61513	Schneider Electric	6/9	E13										
H	-F71	Miniature circuit breaker	C60H DC - 1p- 25 A-C	A9N61513	Schneider Electric	7/9	E2										

Drawn: TNI

Jaska the the mobilienobot

Date: 28.2.2026




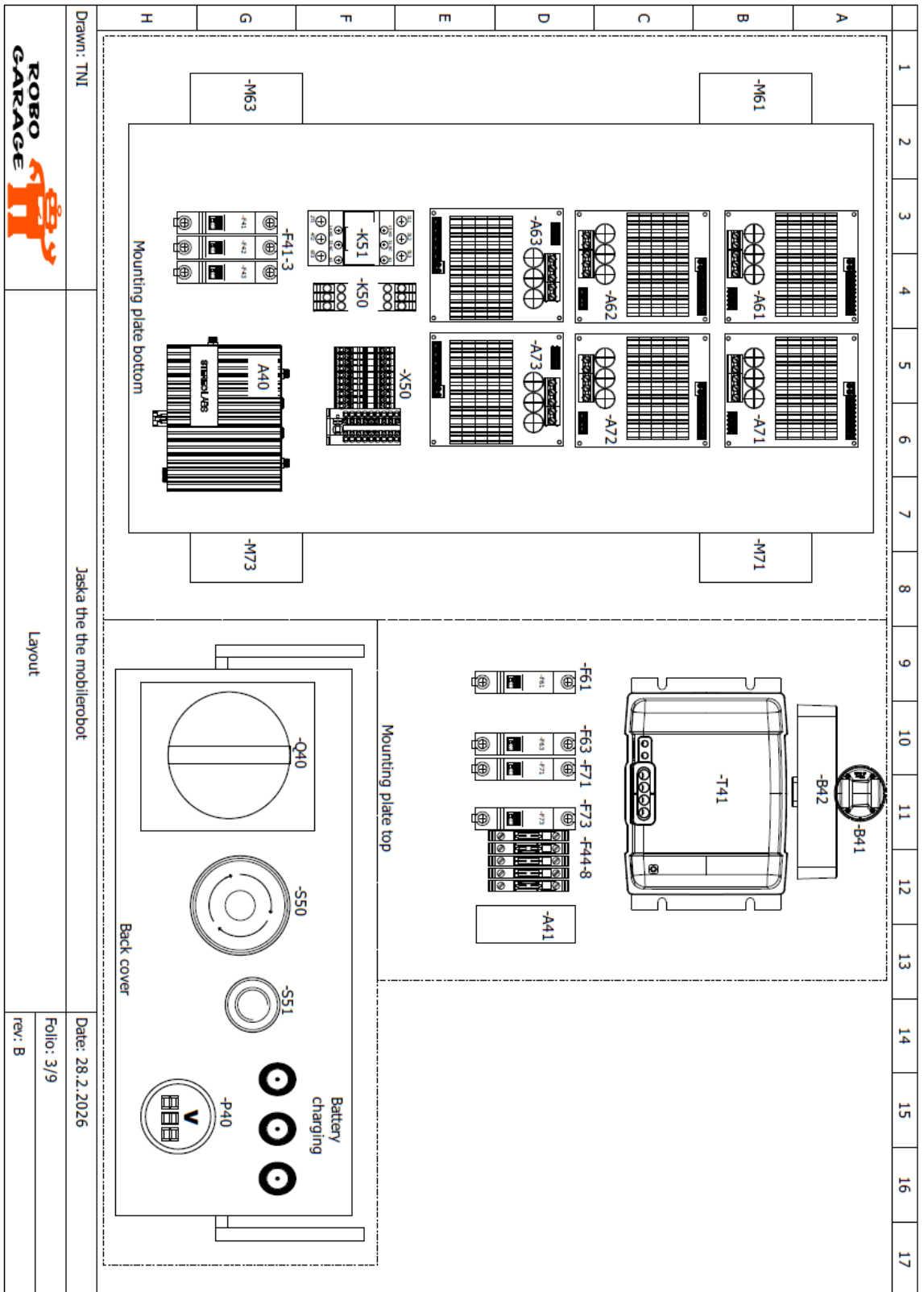
BOM

Folio: 1/9

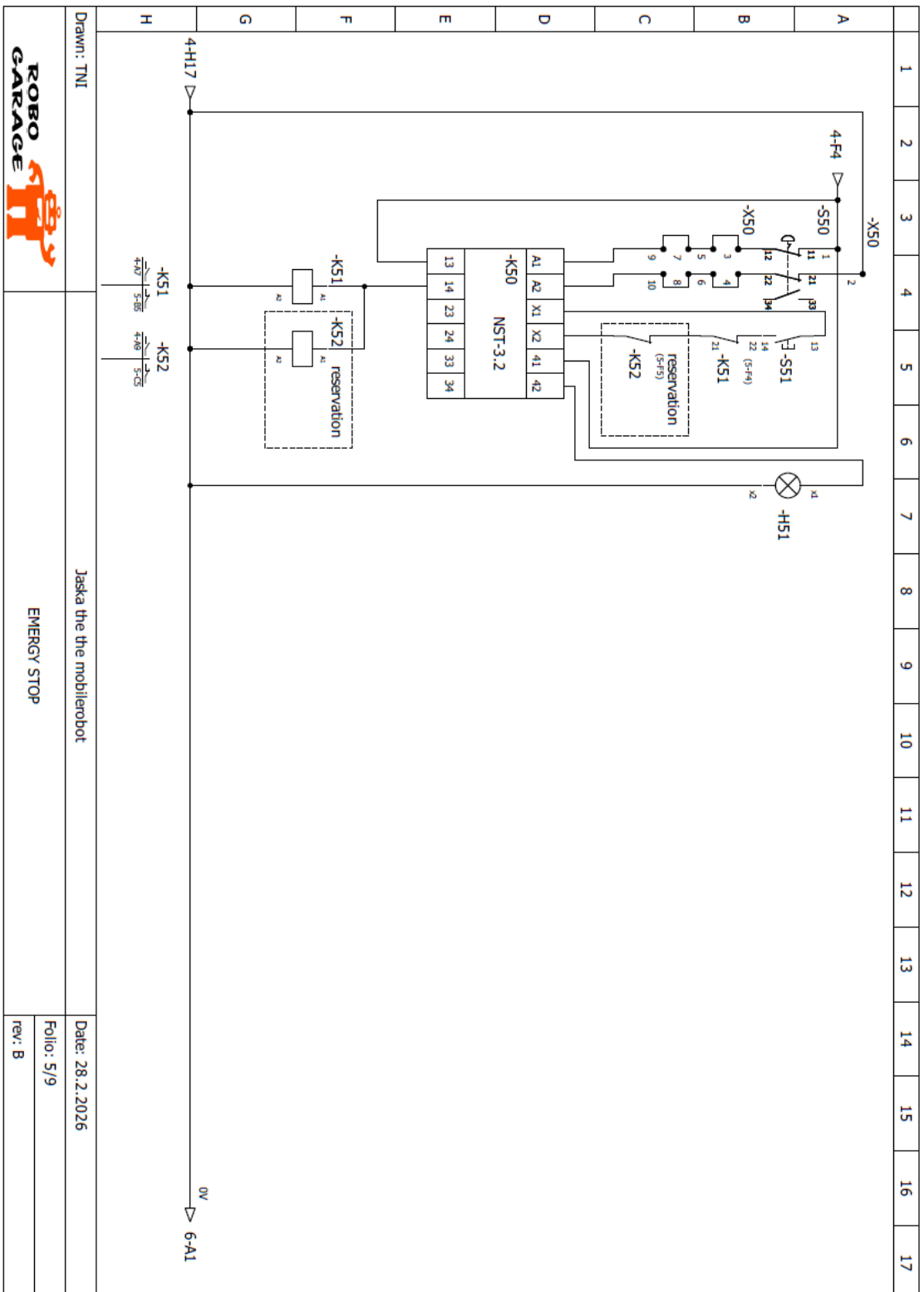
rev: B

Kuva 1. Osaluettelon ensimmäinen sivu

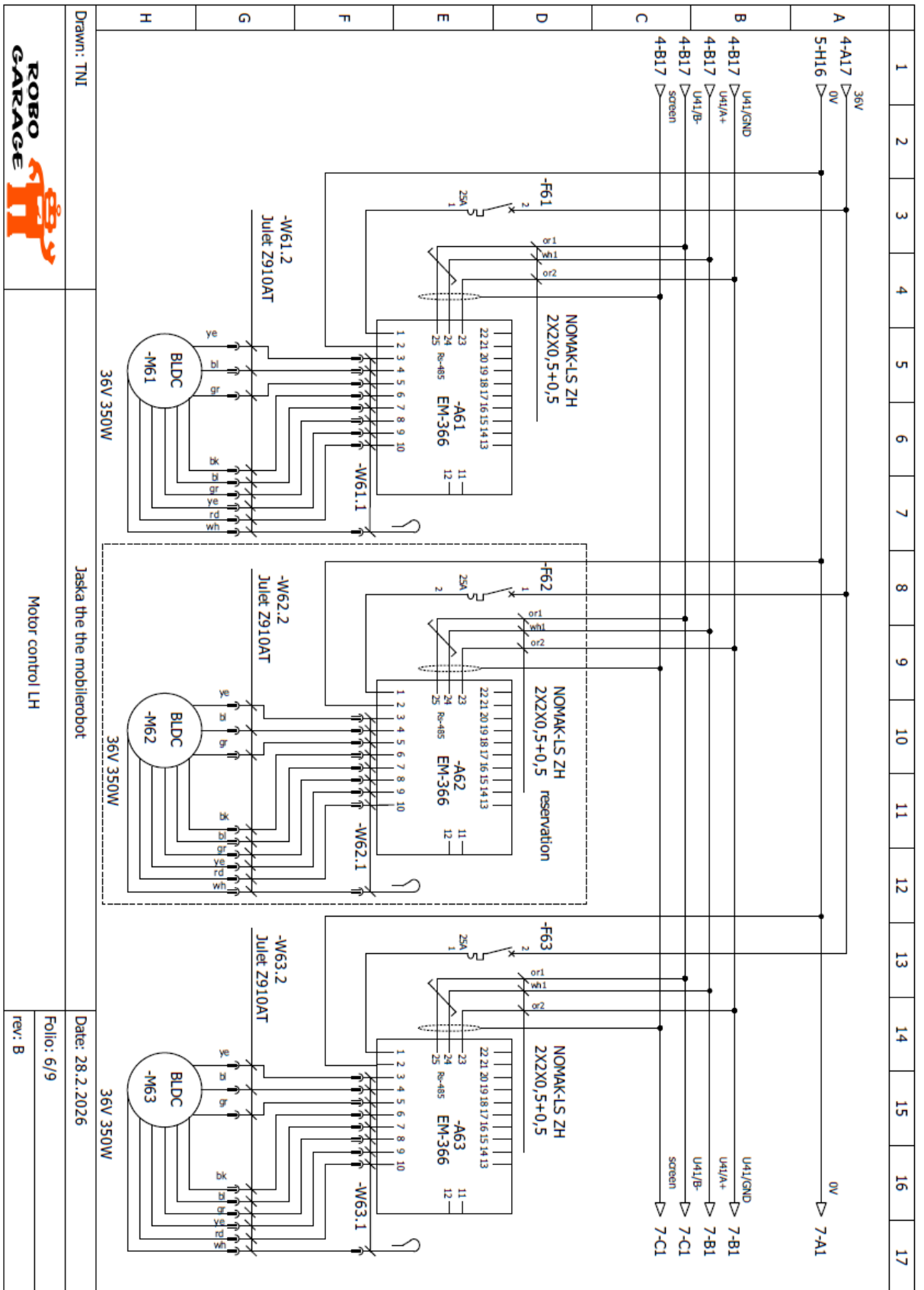
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	Label	Function	Textual description	Article number	Manufacturer	Folio number	Position										
	-F72	Miniature circuit breaker	C60H DC - 1p- 25 A -C	A9N61513	Schneider Electric	7/9	E7										
	-F73	Miniature circuit breaker	C60H DC - 1p- 25 A -C	A9N61513	Schneider Electric	7/9	E13										
	-G41	Battery	36V 30Ah			4/9	B1										
	-G42	Battery	36V 30Ah			4/9	B1										
	-G43	Battery	36V 30Ah			4/9	C1										
	-H51	Light	M22-LED-W	216557	Eaton	5/9	B7										
	-K50	Safety relay	NST-3-2		Duelco	5/9	E5										
	-K51	Contactora	Contactora 38A Tesys D LC1D38JD	LC1D38JD SNRO 3782355	Schneider Electric	5/9	F4										
	-K52	Contactora	Contactora 38A Tesys D LC1D38JD	LC1D38JD SNRO 3782355	Schneider Electric	5/9	F5										
D	-M61	BLDC Motor	HUB Motor 6"160mm		Ozo-electric	6/9	H6										
	-M62	BLDC Motor	HUB Motor 6"160mm		Ozo-electric	6/9	H10										
	-M63	BLDC Motor	HUB Motor 6"160mm		Ozo-electric	6/9	H15										
E	-M71	BLDC Motor	HUB Motor 6"160mm		Ozo-electric	7/9	H5										
	-M72	BLDC Motor	HUB Motor 6"160mm		Ozo-electric	7/9	H10										
	-M73	BLDC Motor	HUB Motor 6"160mm		Ozo-electric	7/9	H16										
F	-P40	Voltagemeter				4/9	C5										
	-Q40	Main switch	OT63FT3 + OHBS2RJ		ABB	4/9	A5										
	-R71	Resistor	120ohm			7/9	D14										
G	-S50	E-stop button	M22-PV/+ 2xCK01+M22-A	171880+216385+216374	Eaton	5/9	B3										
	-S51	Reset Button	M22-DL-B+M22-CK20+M22-A	216931+1078998+216374	Eaton	5/9	B5										
	-T41	DC/DC Converter	Orion-tr 48/12-20A		Victron energy	4/9	E3										
H	-U41	Signal converter	USB to RS485			4/9	B14										
Drawn: TNI		Jaska the the mobilirobot										Date: 28.2.2026		Folio: 2/9			
		BOM										rev: B					



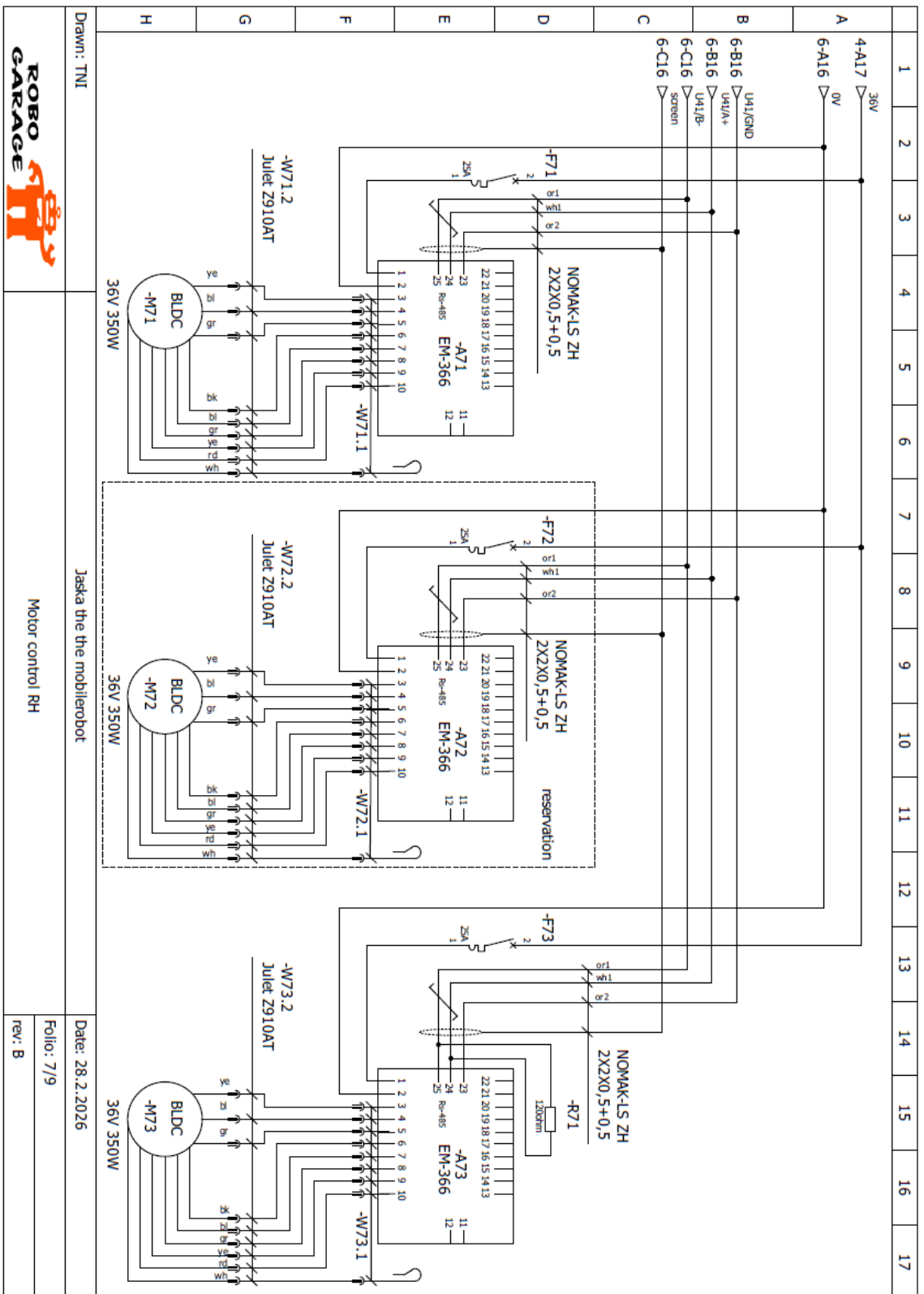
Kuva 3. Komponenttien sijoittelu



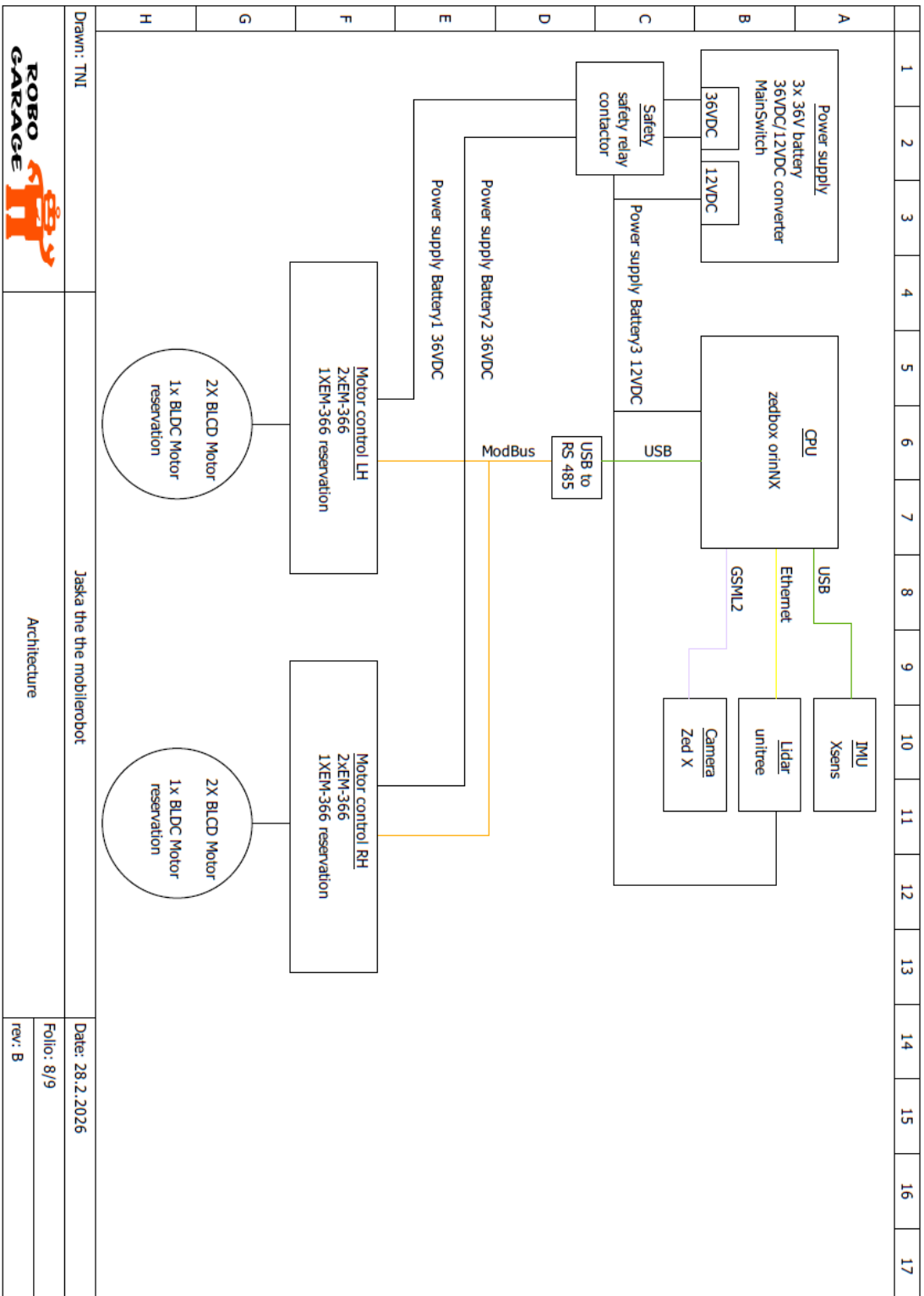
Kuva 5. Hätäseis-piiri



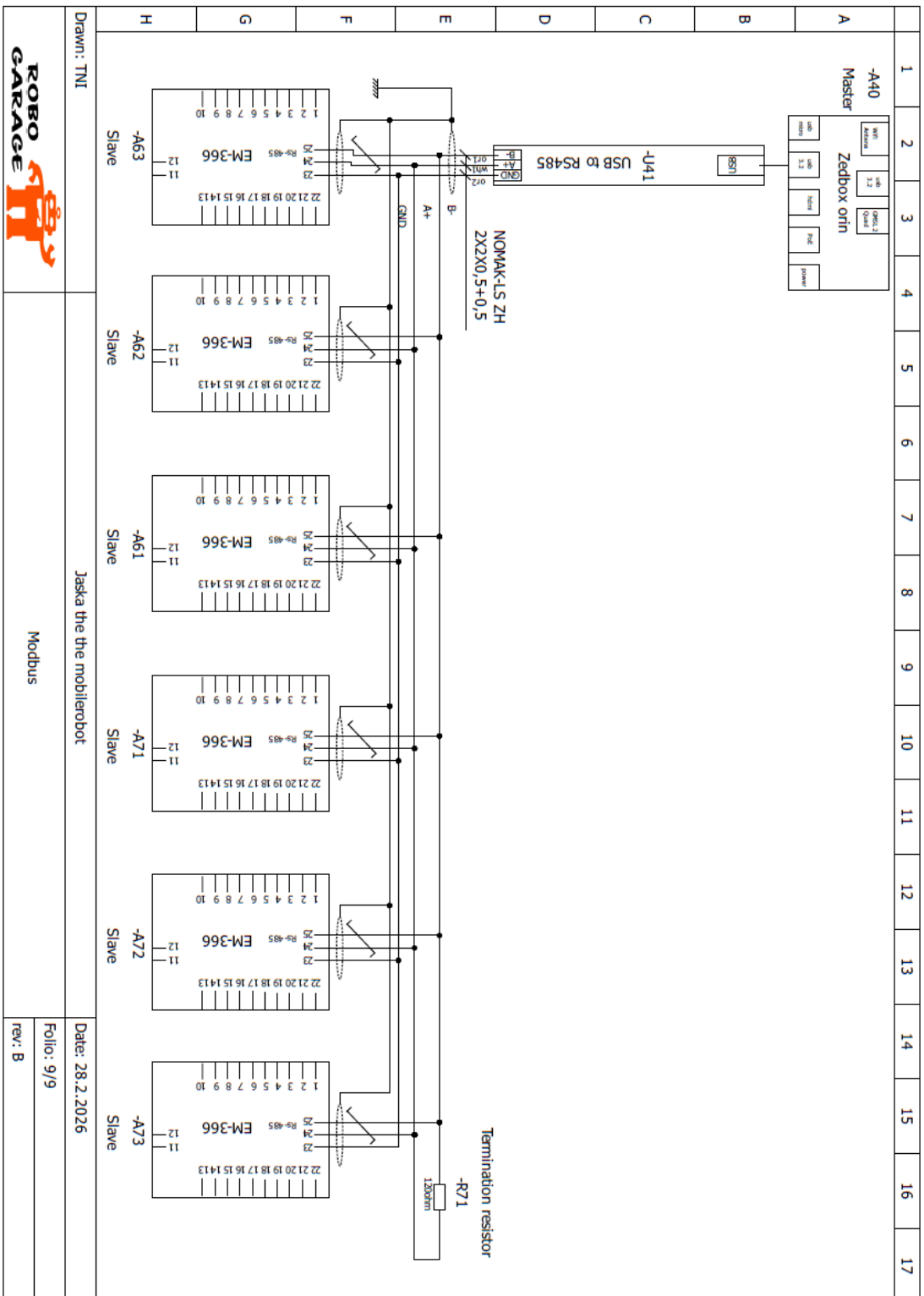
Kuva 6. Vasemmanpuoleisten moottoreiden kytkennät



Kuva 7. Oikeanpuoleisten moottoreiden kytkennät

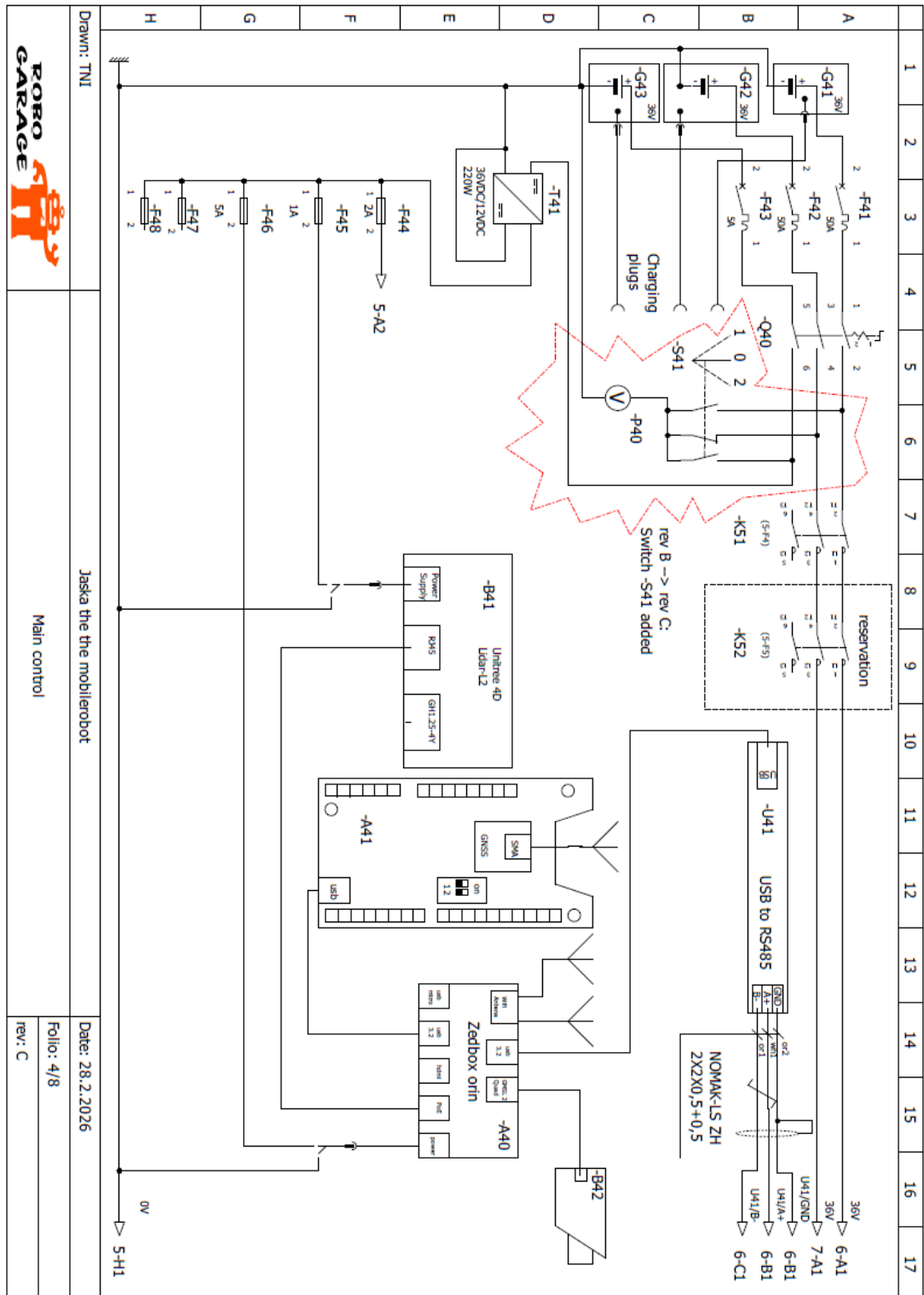


Kuva 8. Järjestelmä arkkitehtuuri



Kuva 9. Modbus-väylä

Ehdotus pääpiirin kytkentämuutoksesta



Kuva 1. Kytentämuutosehdotus