



TALLINNA TEHNIKAÜLIKOOL  
INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

## **Mobiilse platvormi testrakenduse väljatöötamine**

Development of Test Application for Mobile Platform

BAKALAUREUSETÖÖ

Üliõpilane: Andrei Kokurin

Üliõpilaskood: 163909MAHB

Juhendaja: Taavi Möller, insener

## AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“.....” ..... 201.....

Autor: .....

/ allkiri /

Töö vastab bakalaureusetööle esitatud nõuetele

“.....” ..... 201.....

Juhendaja: .....

/ allkiri /

Kaitsmisele lubatud

“.....” .....201....

Kaitsmiskomisjoni esimees .....

/ nimi ja allkiri /

## Elektroenergeetika ja mehhatroonika instituut

### LÕPUTÖÖ ÜLESANNE

**Üliõpilane:** Andrei Kokurin 163909MAHB

**Õppekava, peeriala:** MAHB mehhatroonika

**Juhendaja:** Insener, Taavi Möller, 620 3706, taavi.moller@taltech.ee

#### Lõputöö teema:

(eesti keeles) Mobiilse platvormi testrakenduse väljatöötamine.

(inglise keeles) Development of test application for mobile platform.

#### Lõputöö põhieesmärgid:

1. Uurida hetkel liikuvate robotite ja droonide juhtimiseks kasutatavate kasutajaliideste võimalusi.
2. Luua testlahendus mobiilsete platvormide juhtimiseks.
3. Valmistada 2-3 mobiilset platvormi süsteemi kontrollimiseks.
4. Testida ning hinnata saadud rakendust.

#### Lõputöö etapid ja ajakava:

| Nr | Ülesande kirjeldus   | Tähtaeg    |
|----|--|------------|
| 1. | Enamlevinud robotite ja droonide juhtimiseks kasutatavate kasutajaliideste ülevaade. Testrakenduse kirjeldus.  | 17.03.2019 |
| 2. | Kasutajaliidese testlahenduse arendamine platvormi juhtimiseks. Lisada võimalus saata, salvestada ja laadida käsud. Lisada võimalus jälgida ning peatada käsu täitmist reaajas. Lisada mitu võimalust mobiilse platvormi juhtimiseks. Lisada roboti positsiooni kuvamist kaardil. Võimalus juhtida mitu robotit korraga. | 14.04.2019 |
| 3. | Mobiilsete platvormide valmistamine süsteemi testimiseks. Kirjeldada lahendust roboti asendi tagasiside saamiseks. Testimisplatsi valmistamine. Mobiilse platvormi juhitavuse kontrollimine testimisplatsil.   | 28.04.2019 |
| 4. | Lõputöö dokumentatsiooni täitmine, koodi ülevaade, viimased parandused ja kokkuvõte.   | 12.05.2019 |

**Töökeel:** eesti

**Lõputöö esitamise tähtaeg:** "21" mai 2019a

**Üliõpilane:** Andrei Kokurin ..... "....." .....201....a

/allkiri/

**Juhendaja:** Taavi Möller ..... "....." .....201....a

/allkiri/

**Konsultant:** ..... "....." .....201....a

/allkiri/

## **ANNOTATSIOON**

Bakalaureusetöö eesmärgiks on luua testrakendus mobiilse platvormi juhtimiseks. Testrakendus see on süsteem, mis koosneb kasutajaliidesest roboti juhtimiseks, mobiilsest testplatvormist ning testimisplatsist. Süsteemi hakatakse kasutama robotika õpetamiseks ning on mõeldud robotika harrastajate projektidele, kus kasutatakse väikse mäluga kontrollereid. Arendatav süsteem võimaldab juhtida platvormi liikumist, platvormi peale paigaldatud täiturit ning lugeda anduri väärtusi. Antud töö on sisendiks süsteemi edasiseks arendamiseks mobiilsete platvormide juhtimisel.

Töö kirjutamise ajendiks on kasutaja segasus, mis tuleneb osaliselt keerulisest kasutajaliidesest, süsteemi püsti saamisest ning süsteemi osade kõrgest hinnast.

Tulemusena antakse ülevaade erinevatest arvutiprogrammidest, mida kasutatakse robotika harrastajate projektides. Lisaks luuakse testrakendus, mida saab kasutada teistes robotika projektides.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 41 leheküljel, 29 joonist ja 16 tabelit.

## **ABSTRACT**

### **Development of Test Application for Mobile Platform**

The aim of this thesis is create test application for mobile platform control. The test application is an example of system, which consist of robot controlling graphical user interface, mobile platform and testing ground. The target group of this work are robotics educational and hobby projects, which use low memory microcontrollers. This system allows to control platform movements, but also platform based actuators and sensors. This thesis is an input into mobile platform control further development.

The reason behind this thesis is users confusion, which comes from difficult graphical user interface, complex system set up and required parts high cost.

As a result of this thesis overview of different computer programs for hobby robotics will be completed. The test application system will be built and tested additionally. The test application part could be used in other projects.

This thesis is in Estonian language and contains 41 pages of text, 29 figures and 16 tables.

# SISUKORD

|  |    |
|--|----|
| Lühendite ja tähiste loetelu                                 | 8  |
| Jooniste loetelu   | 9  |
| Tabelite loetelu   | 10 |
| SISSEJUHATUS   | 11 |
| 1.ÜLEVAADE LEVINUD DROONIDE JA ROBOTITE JUHTIMISTARKVARADEST | 12 |
| 1.1 Vaadeldud programmid                                     | 12 |
| 1.1.1 ROS virtuaalne platvorm                                | 12 |
| 1.1.2 Ardupilot  | 13 |
| 1.1.3 Open robot   | 14 |
| 1.1.4 ZigBee+Solar+Robot                                     | 14 |
| 1.1.5 Logo   | 15 |
| 1.1.6 MATLAB Mobile Robotics Simulation Toolbox              | 16 |
| 1.1.7 BT Robot Controller v1.2                               | 17 |
| 1.2 Programmide ülevaade                                     | 17 |
| 1.3 MAVLink sideprotokoll                                    | 19 |
| 2.TESTRAKENDUS   | 21 |
| 2.1 Testrakenduse funktsionaalsus                            | 21 |
| 2.2 Testrakenduse moodulid                                   | 22 |
| 2.3 Mobiilne testplatvorm                                    | 24 |
| 2.3.1 Elektroonika   | 25 |
| 2.3.2 Mikrokontrollerite püsivara                            | 29 |
| 2.4 Sideprotokoll ja andmevahetus                            | 31 |
| 2.5 Kasutajaliidese kirjeldus                                | 34 |
| 2.5.1 Juhtimise vahekaart                                    | 34 |
| 2.5.2 Programmi täitmine käsureast                           | 37 |
| 2.5.5 Jadaliidese monitoorimine                              | 38 |
| 2.5.6 Jadaliidese seadistamine                               | 39 |
| 2.5.7 Tööriistade seadistamine                               | 41 |
| 2.5.8 Kaardi seadistamine                                    | 43 |
| 2.6 Testimisplats  | 44 |
| 2.6.1 Navigatsioon ja positsiooni jälgimine                  | 45 |

|  |    |
|--|----|
| 2.7 Testrakenduse katsetamine                                    | 47 |
| KOKKUVÕTE  | 49 |
| SUMMARY  | 51 |
| KASUTATUD KIRJANDUS  | 53 |
| LISAD  | 56 |
| Lisa 1 - Qt Creatori Graafilise disaini tööriist                 | 56 |
| Lisa 2 - Open Robot kasutajaliidese ekraanipilt                  | 57 |
| Lisa 3 - Zigbee+Solar+Robot projekti kasutajaliidese ekraanipilt | 58 |
| Lisa 4 - Käskude salvestamine ja laadimine                       | 59 |
| Lisa 5 - BT Robot Controller v1.2 kasutajaliidese ekraanipilt    | 60 |
| Lisa 6 - Logo kasutajaliidese ekraanipilt ja baaskäsud           | 61 |
| Lisa 7 - Testrakenduse failid                                    | 63 |
| Lisa 8 - Elektroonika skeemid ja joonised                        | 64 |
| Lisa 9 - Käsurea programmide näited                              | 68 |
| Lisa 10 - Mobiilse platvormi nurga leidmise katse analüüs.       | 69 |

## Lühendite ja tähiste loetelu

|             |   |
|-------------|---|
| RFID        | — RFID on lühend sõnadest <i>Radio Frequency IDentification</i> . See on viis objektide puutevabaks identifitseerimiseks [4].   |
| COM port    | — <i>Communication port</i> . Arvuti port kommunikatsiooni jaoks välisseadmetega.   |
| baud kiirus | — side kiiruse mõõtmise ühik elektroonikas, mis näitab mitu korda signaal muutub 1 sekundi jooksul.   |
| ROS         | — ROS on lühend sõnadest <i>Robot Operating system</i> . Virtuaalne platvorm roboti süsteemide arendamiseks [29].   |
| PWM         | — <i>Pulse-Width Modulation</i> on pulsilaiusmodulatsioon. PWMi kasutatakse väljundpinge reguleerimiseks kiirete sisse- ja väljalülitamiste abil [27].  |
| GPIO        | — GPIO on lühend sõnadest <i>General Purpose Input/Output</i> . Kõik GPIO sisend väljundid valitakse (tarkvaras) sisendina või väljundina ning nende kasutusala on lai ulatus [8].  |
| UART        | — UART on lühend sõnadest <i>Universal Asynchronous Receiver-Transmitter</i> . See on andmeedastuse standard, mida kasutatakse mikrokontrollerite omavaheliseks andmesideks ja ühenduseks muude väliste seadmetega [28].                                      |
| ASCII tabel | — ASCII on lühend sõnadest <i>American Standard Code for Information Interchange</i> . See on tabel, mida kasutatakse ingliskeelsete tähemärkide, numbrite, märkide ja klaviatuuri funktsioonide edastamiseks ja kuvamiseks. Tabelis on kokku 128 märki [26]. |



## Jooniste loetelu

Joonis 1. MavLink saadetise paketti formaat.

Joonis 2. Testrakenduse moodulid.

Joonis 3. Mobiilne testplatvorm.

Joonis 4. Elektroonika I osa trükkplaadi laotus KiCad programmis koos selgitustega.

Joonis 5. Elektroonika I osa mikrokontrolleri programmi põhifunktsionaalsuse plokkskeem.

Joonis 6. Elektroonika II osa mikrokontrolleri programmi põhifunktsionaalsuse plokkskeem.

Joonis 7. Kasutajaliidese juhtimise vahekaart.

Joonis 8. Võimalikud tööriistad.

Joonis 9. Programmi saatmise algoritmi plokkskeem.

Joonis 10. Jadaliidese monitoorimise vahekaart.

Joonis 11. Jadaliidese parameetrite seadistamise vahekaart

Joonis 12. Jadaliidese avamise algoritm.

Joonis 13. Kasutajaliidese nuppude seadistamise vahekaart.

Joonis 14. Liugurite ja andurite parameetrite seadistamise aknad.

Joonis 15. Platvormi liikumiste kaart.

Joonis 16. Testimisplats.

Joonis 17. Tagasiside mobiilse platvormi positsioonist.

Joonis 18. Nurga leidmise algoritm.

Joonis 19. Qt Creator graafilise disaini tööriist.

Joonis 20. Open Robot kasutajaliidese ekraanipilt.

Joonis 21. Zigbee+Solar+Robot Projekti kasutajaliidese ekraanipilt.

Joonis 22. Käsu salvestamise aken.

Joonis 23. Käsu laadimise aken.

Joonis 24. Bt Robot Controller kasutajaliidese ekraanipilt.

Joonis 25. Logo programmi kasutajaliidese ekraanipilt.

Joonis 26. Elektroonika I osa skeem KiCad programmis.

Joonis 27. Elektroonika II osa skeem KiCad programmis.

Joonis 28. Elektroonika I osa trükkplaat sobib nii samm-mootorite, kui ka alalisvoolu mootorite jaoks.

Joonis 29. USB-Bluetooth signaali konverter.

## **Tabelite loetelu**

Tabel 1. Programmide ülevaade tabel

Tabel 2. Elektroonika I osa komponentide nimekiri koos selgitusega

Tabel 3. Komponentide tabel

Tabel 4. Sideprotokolli saadetise pakett

Tabel 5. Sideprotokolli saadetise segmentide kirjeldus

Tabel 6. *MSG* tüüpide tabel

Tabel 7. Saadetise näited

Tabel 8. Juhtimise vahekaarti peamised elemendid

Tabel 9. Jadaliidese monitoorimise vahekaardi elemendid

Tabel 10. Jadaliidese parameetrite seadistamise vahekaardi elemendid

Tabel 11. Juhtimise tööriistade seadistamise vahekaardi elemendid

Tabel 12. Kaardi seadistamise vahekaardi elemendid

Tabel 13. Logo baaskäsud

Tabel 14. Vajalikud AT käsud moodulite HC-06 ühendamiseks

Tabel 15. Käskude ja programmide näited

Tabel 16. Nurga asendi leidmise algoritmi katsetulemused.

## SISSEJUHATUS

Bakalaureusetöö eesmärk on luua testrakendus mobiilse platvormi juhtimiseks. Testrakendust luuakse robotika õpetamiseks ning robotika harrastajate projektide tarbeks. Praeguste lahenduste puudusteks on spetsiifilise riistvara kasutamine, keeruline süsteemi osade paigaldamine ja juhtimisloogika ning mõnel juhul funktsioonide piiratud valik.

Lahendus peab sobima erineva taseme projektidele. Lähtuvalt mõistlikkuse printsiibist tuleb lahenduses kasutada eelkõige lihtsaid, arusaadavaid ning töökindlaid lahendusi. Testrakendus see on süsteem, mis koosneb järgmistest osadest: mobiilne testplatvorm, kasutajaliides ja testimisplats. Süsteem on näidslahendus, mida robotika harrastajad saavad võtta aluseks enda projektides. Süsteemi osasid saab muuta vastavalt projekti nõuetele, kuid seda kindlas piirides. Mobiilne platvorm on alus, mille peale kasutaja saab vastavalt projektile lisada täitureid: kaamera, andurid, haaratsid jne. Kasutajaliides on vajalik kasutaja ja kontrolleri vahelise suhtluse tagamiseks. Eesmärgi saavutamiseks lahendatakse järgmised alamülesanded:

1. Vaatlus. Vaatluse eesmärk on tutvuda olemasolevate lahendustega, määrata süsteemi põhinõudmised ja vajalik funktsionaalsus. Vaadeldakse nii kerged kui ka tööstuslikke robotite seadistamiseks kasutatavaid lahendusi. Antud osa tulemusena koostatakse ülevaade tabel.

2. Testrakenduse süsteemi kirjeldus:

1. Mobiilse testplatvormi kirjeldus. Elektroonika skeemi kavandamine.

2. Testrakenduse kasutajaliidese kirjeldus.

3. Platvormi juhtimissüsteemi kavandamine ja valmistamine.

4. Testimisplatsi kirjeldus.

5. Testrakenduse katsetamine. Ülevaade õnnestunud süsteemi osadest. Arendamist vajavate kohtade leidmine ning võimalikke lahenduste pakkumine.

Töö esimese osas tutvustatakse ning võrreldakse robotite ja droonide juhtimiseks robotika harrastajate projektides kasutatavaid programme. Teises osas kirjeldatakse ning valmistatakse süsteemi testlahendust mobiilsete platvormide juhtimiseks, mis koosneb mobiilsest platvormist, roboti juhtimise tarkvarast ning testimisplatsist. Lisaks analüüsitakse tehtud testsüsteemi. Lõpuks tuuakse välja kõik arendamist vajavad aspektid ning leitakse suund projekti edasiseks arendamiseks.

# 1.ÜLEVAADE LEVINUD DROONIDE JA ROBOTITE JUHTIMISTARKVARADEST

Selles peatükis antakse üldine ülevaade arvuti programmidest, mida kasutatakse mobiilsete robotite juhtimiseks. Peatükis vaadeldakse olemasolevaid lahendusi, tuvastatakse lahenduste tugevad ja nõrgad kohad, uuritakse funktsioonide kasutust roboti juhtimise rakendustes. Töö eesmärgiks on luua optimaalne lahendus, mis jääb lihtsate ja keeruliste süsteemide vahele. Süsteemi peamine nõue on roboti riistvara vahetatavus. Ülevaates ei kajastata projekte, kus tarkvara on koos robotiga ning juhtusid, kus robot või selle osad ei ole vahetatavad.

## 1.1 Vaadeldud programmid

### 1.1.1 ROS virtuaalne platvorm

ROS – avatud lähtekoodiga projekt. ROS põhineb universaalse tarkvara kirjutamise ideel, mis eeldab töötamiseks erinevate robotitega väikest muudatust koodis. ROS on selline virtuaalne alus, millele lisatakse vajalikud funktsioonidega pakid [29].

Funktsioonipakkide eesmärk on lahendada selliseid robootika probleeme nagu koordinaadistiku teisendamine, liikumise planeerimine, kommunikatsioon, andurite integreerimine jne. Kasutaja saab projektis vajamineva tööriistade paki alla laadida või ise kirjutada ning lisada baasprogrammile. Kõik ROS funktsioonid ja tööriistad on universaalsed ning lisatavad ka teistesse ROS projektidesse. Selle süsteemi eelis on lihtsus, sest paljud funktsioonid on juba loodud ning neid ei pea nullist uuesti looma. Esialgelt loodi ROS Stanfordini ülikooli tehisintellekti laboris, praegu on ROS virtuaalne platvorm tasuta nii harrastajate kui ka kommerts projektide jaoks. ROS kogukond kasutab laia valikut erinevaid funktsioone, kuid programmi kasutaja peab tundma programmeerimist kõrgel tasemel [30].

ROS baasile saab integreerida ka teised avatud lähtekoodiga projektid:

- *Move it* – virtuaalne platvorm roboti liikumise planeerimise teostamiseks. Sellega saab määrata roboti liikumise alguspunkti ning lõpp-punkti. Algoritmi põhjal genereeritakse optimaalne teekond takistuste vahel. Lisaks juhitakse paljusid tööstuslikke käe-laadseid manipulaatoreid. Sellega saab manipuleerida 3d objektidega ning luua dünaamilisi 3d vaateid robotist ning keskkonnast [31].
- *Gazebo* – virtuaalne platvorm, mida kasutatakse roboti käitumise simulatsiooniks erinevates olukordades. Virtuaalne platvorm võimaldab testida algoritme, õpetada roboti disaini ning tehisintellekti, kasutades realistlikke stsenaariume [32].

## 1.1.2 Ardupilot

Ardupilot on nii kogukond kui ka avatud lähtekoodiga tarkvara, mis pakub lahendusi iga võimaliku sõidukitüübi juhtimiseks. Ardupilot tarkvara on laialdaselt testitud ning töökindel. Selles kogukonnas on sõiduki juhtsüsteemid, mis koosnevad järgmistest osadest: riistvara, püsivara ja tarkvara. Riistvara moodustub drooni mikrokontrollerist ning selle külge ühendatud anduritest, GPS moodulitest, sidemoodulitest, mootoritest ja teistest täituritest. Paljud komponendid on tihti integreeritud kontrolleri trükkplaadile. Püsivara on programm, mis laaditakse mikrokontrollerile. Püsivara on olemas järgmiste seadmete jaoks: neljarattaline maastiku droon, helikopter, lennuk, lendav droon (3-9 propellerit) ja drooni positsioneerimise tarbeks kasutatav majakas. Ardupilot tarkvara on arvuti- või nutitelefoni programm, mida kasutatakse drooni manuaalseks või autonoomseks juhtimiseks. Drooni juhtimiseks autonoomselt määratakse teekond märgitud läbitavate punktidega ning seda teekonda nimetatakse missiooniks. [18].

Arvutit või nutitelefoni kasutatakse sisuliselt puldina. Arvuti ja kontrolleri vaheline andmevahetus toimub läbi MAVLink side protokoll. Täpsemalt MAVLink protokollist saab lugeda peatükis 1.3. Ardupilot pakutavatest erinevatest tarkvaradest on kõige levinumad järgmised tarkvarad [34]:

- Mission Planner – seda kasutatakse enamasti lendavate droonide juhtimiseks. Ardupilot kogukonna levinuim tarkvara.
- QGroundControl – seda saab kasutada nii arvuti kui ka Android seadmetel. See on avatud lähtekoodiga programm, mida saab muuta vastavalt oma vajadustele Qt Creator tarkvara arendusplatvormis.
- UgCS – *Universal Ground Control Station* suudab ühtida mitu drooni korraga, lubab ühendada mitu juhti korraga. Esimene kasutaja saab tegeleda juhtimiseks ning teine pöörata kaamerat ning pildistada.
- Tower – selle rakenduse eripära on aerofotode tegemine ning objektide 3d kaardistamine.
- AndroPilot – see rakenduse töötab android seadmetega.

Kasutaja seisukohalt on kõik eelpool nimetatud programmid sarnased. Programmi on integreeritud navigatsioonisüsteem, mille puhul saab maailma kaardilt tuvastada drooni asukohta ning valida drooni teekond kaardile märgitavate punktidega. Kui kasutaja vajutab “Alusta missiooni” nupule, siis droon püüab etteantud punktid läbida. Missiooni alustamiseks on vaja teada alguspunkti asukohta. Kasutajaliides võimaldab jälgida güroskoobi ning kõrgusanduri väärtusi, samuti määrata drooni tööpiirkonda, mille ületamisel droon püüab jõuda tagasi, signaali

kaotamisel naaseb droon alguspunkti. Arvuti ühendamiseks drooniga tuleb COM-porti ühendada SiK telemeetriline raadio, tööala raadiusega 300 meetrit [23].

Ardupilot kogukond soovib ka elektroonikat droonide juhtimiseks. Maastiku drooni jaoks soovib kogukond kasutada Pixhawk plaati. Selle plaadil on 32-bitine ARM mikrokontroller, 256 KB RAM mälu ja 2 MB Flash mälu. Anduritest on Pixhawk plaadil güroskoop, digitaalne kompass, kiirendusandur ja baromeeter. Sisseehitatud andurite eelis on kontaktide kindlus värelemise ohu eest. Plaadil on sisseehitatud 5 UART sisendit, GPS mooduli sisend ja teised sisendid [33].

### **1.1.3 Open robot**

Open robot eesmärk on luua lihtne ja mugav kasutajaliides roboti juhtimiseks. Open Robot kasutab side jaoks ZigBee moduleid [5]. Roboti ühendamiseks kasutatakse nii interneti IP aadressi kui ka arvuti COM-porti. Lisaks on kasutajaliides varustatud RFID operatsioonidega, see võimaldab teostada lugemise ja kirjutamise operatsioone automaatselt.

Roboti mootorite kiirust reguleeritakse PWM töötsükli abil. Mobiilse platvormi rataste pöörete jälgimiseks kasutatakse enkooderid. Kasutajaliidese aknasse on integreeritud kalkulaator, millega saab konverteerida enkoodri pulseerimisi kiiruseks või nurga asendiks. Kasutaja sisestab ratta enkoodri pulseerimiste arvu ning kalkulaator teisendab selle vastavaks väärtuseks kasutajale arusaadavates ühikutes.

Kasutaja saab nuppudega manipuleerida kontrolleri sisendi/väljundi nivood kõrgeks või madalaks, samuti vaadata andurite hetkeväärtuseid. Anduri väärtused sisalduvad seitsme elemendiga massiivis, kus iga element vastab kindla anduri väärtusele ning on mahuga üks bait. Algoritmiliselt on võimalik luua erinevad sündmused, mis sõltuvad andurite väärtusest. Näiteks kui ühe või mitme kaugusanduri väärtus ületab ettenähtud piiri, siis robot peatub või väldib tuvastatud takistust. Parameetreid saab salvestada kontrolleri EEPROM mällu, need parameetrid jäävad kontrolleri mällu hetkeni, kui need sealt kustutatakse. Roboti osade juhtimiseks projektis kasutatakse ainult PIC18F4520 mikrokontrollerit [6]. Selle projekti kasutajaliidese aknaga saab tutvuda lisa 2.

### **1.1.4 ZigBee+Solar+Robot**

Selle projekti eesmärk on luua neljarattaline mobiilne platform koos käe-manipulaatoriga, mille juhtimine toimub arvuti programmi kaudu. Selle projekti raames juhitakse neljarattalist vankrit nooltega kasutades arvutihiirt, klaviatuuri või pihukonsooli. Manipulaatori positsiooni on jälgitav

ekraanilt, millel on dünaamiline pilt ning aknad iga lüli nurga määramiseks. Selle projekti kasutajaliidese aknaga saab tutvuda lisas 3. Veel saab määrata manipulaatori tööriista lõpp-punkti koordinaate ning programm teeb ise kõik arvutused vastavasse asukohta liikumiseks. Manipulaatori juhtimiseks kasutatakse liugurit. Roboti kiirust kuvatakse protsentides maksimaalse kiiruse suhtes. Sidemooduli signaali tugevust ning aku mahtu kuvatakse olekuribal. Side roveri ja arvuti vahel toimib XBee mooduli abil. Roveri osade juhtimiseks kasutatakse PIC16F kontrolleri, millel on 16 GPIO sisend/väljundeid [19].

### 1.1.5 Logo

Logo on üks vanematest programmeerimise keeltest. Esialgu oli see mõeldud robotite juhtimiseks, kuid maailmas sai see tuntuks arvutiprogrammina. Logo lahendust kasutati aktiivselt programmeerimise loogika õpetamiseks. Hetkel on sellest keelest loodud sajad versioonid, kusjuures esimene versioon loodi aastal 1967 [11]. Roboteid, mis kasutavad Logo käske enda töös, nimetatakse *turtle* robotiteks ehk kilpkonnadeks. Baaskäskudeks on näiteks: forward, back, left, right, print, repeat jne. Logo programmi tüüpiline kasutajaliides ja baaskäskude näited on esitatud lisas 6. Selle keele eripäradeks on interaktiivsus, modulaarsus, pikendatavus ja andmetüüpide paindlikkus [12]:

1. Interaktiivsus - See väljendub tagasisides käsu õigsuse kohta. Näiteks, kui kasutaja sisestab käsu *"bac"*, siis tuleb vastuseks *"I don't know how to bac"*, ehk *"Ma ei tea mida bac tähendab"*.

Järgmisena sisestatakse käsk *"back"*, vastuseks tuleb *"Not enough inputs to forward"*, ehk *"Käsu back käivitamiseks puudub sisend"*. See tähendab, et Logo tuvastas käsu, kuid ei saa seda käivitada, kuna puudub lisainformatsioon.

Järgnevalt lisatakse *"back"* käsule informatsiooni ja trükitakse *"back 200"*, selle tulemusena Logo saab käivitada ning liigutab robotit 200 sammu tagasi [12].

2. Modulaarsus ja pikendatavus. Logo programm koosneb protseduuridest, mida saab ühendada mooduliteks. Need moodulid saab ühendada omakorda suuremateks mooduliteks. Näiteks koostame moodulit ristkülik, mida kasutatakse ristküliku trajektoori läbimiseks:

*"to ristkülik*

*repeat 4 [forward 100 left 90]*

*end*

Eelnevalt defineeritud moodulit ristkülik saab kasutada uue mooduli parkla kirjutamiseks:

*“to parkla*

*repeat 10 [forward 100 ristkülik]*

*end*

Varasemalt defineeritud moodulid lisatakse Logo baaskäskude nimekirja ning neid saab edaspidistes moodulites kasutada juba baaskäsuna. See võimaldab luua suuri ja keerulisi programme, kasutades väikesi mooduleid [12].

### 3. Andmetüüpide paindlikkus

Käskude parameetrid saab sisetada nii numbrit, sõnade ja massiividena. Käskud pole tundlikud parameetri andmetüüpi vastu ning ise konverteerivad neid [12].

## 1.1.6 MATLAB Mobile Robotics Simulation Toolbox

MATLAB (lühend MATrix LABoratory) on programm insenerlike ja teaduslike arvutuste teostamiseks. MATLAB lubab valmistada oma kasutajaliidest protsesside kontrollimiseks. MATLAB tihti kasutatakse manipulaatorite juhtimiseks, kuna selles on lihtne teostada arvutusi lülide pöördmaatriksitega. Samuti saab juhtida ka mobiilsete robotite käitumist. MATLAB kasutajaliidest saab ühendada nii lihtsate kontrolleritega COM-pordiga, kui ka ROS süsteemiga [20].

MATLAB eelisteks võrreldes teiste arvuti programmidega on:

1. Programmi koodi on suhteliselt kerge kirjutada, pole vaja kasutada eraldi kompilaatoreid koodi konverteerimiseks masinkoodiks.
2. MATLAB on täielik programm, mis töötab erinevates arvutisüsteemides.
3. Defineeritud funktsioonid. Programmis on suur valik defineeritud baasfunktsioone. MATLAB kogukonnast saab laadida ka uusi funktsioone. Näiteks on olemas “Mobile Robotics Simulation Toolbox” funktsioonide pakett, millega saab simuleerida mobiilse roboti kinemaatikat ja sensoreid 2D keskkonnas [21].
4. Graafiku ehitamise tööriistad. MATLAB programmis on sisseehitatud tööriistad graafikute ehitamiseks.
5. Saab valmistada kasutajaliidese protsesside simulatsiooniks ning juhtimiseks.
6. Suur valik erinevatest lisapakettidest.



MATLAB on mugav tööriist projektide arendamisel, juhul, kui projektis osaleb mitu arendajat, kuid robotite arv on piiratud. Kasutades MATLABi tööriistu saab genereerida simulatsiooni, kus robotil on keha ja keskkonna 3d mudelid koos mehaanika omadustega. Näiteks võimaldab SIMULINK luua takistusi vältiva roboti loogika ning seejärel genereerida ruumi, kus testida programmi loogikat [22]. Samuti on olemas valmis projektid mobiilsete robotite juhtimiseks. Tasub meeles pidada, et MATLAB kasutajaliidesed töötavad ainult siis, kui arvuti peal on paigaldatud õige MATLAB versioon. Puuduseks on see, et MATLAB litsents on tasuline.

### **1.1.7 BT Robot Controller v1.2**

BT Robot Controller v1.2 on kasutajaliides lihtsate mänguautode või robotite juhtimiseks. Kasutajaliidese ekraanipilt on toodud lisa 5. Programm on universaalne ning sobib erinevate kontrollerite jaoks. Arvuti ja roboti vahelise side jaoks kasutatakse USB-UART konverterit. Kasutajaliides on selgelt tehtud, see on arusaadav ja kergesti kasutatav. Programmi osad on teineteisest eraldatud nende rakenduse põhimõttel [25]. Kokku programm on jaotatud neljaks vahekaardiks:

1. Juhtimise vahekaardilt saab robotit nooltega liigutada ning lülitada sisse ja välja, lugeda toitepinge väärtust ning käivitada erinevaid LEDe, vilkureid ja sireene, vahetada režiime autonoomse ja manuaalse vahel. Toitepinge väärtuse lugemiseks tuleb vajutada vastavale nupule. Nuppude signaale pole võimalik muuta. Programmi alumises osas kuvatakse side olek ja viimati saadetud signaali väärtust.
2. Terminali vahekaardilt loetakse COM-porti saabuvasid signaale ning samuti saab saata enda käsked.
3. Programmi seadistamise vahekaardil saab valida side parameetreid ning arvuti ja roboti sidevahetuse COM porti. Programmi käivitamisel ilmub programmi seadistamise vahekaart esimesena ekraanile. Roboti juhtimiseks tuleb esmalt ühendada sidemoodul.
4. Programmi kirjelduses saab tutvuda nuppude signaalide kirjeldusega. Lisaks kuuluvad alajaotisesse lingid programmi allika veebilehele ning kasutajatoele.

## **1.2 Programmide ülevaade**

Tabelis 1. on toodud juhtimistarkvarade olulisemad omadused: universaalsus, kasutamise keerukus, võimalikud funktsioonid, arendamise võimalused, kasutatav mikrokontroller.

Tabel 1. Programmide ülevaade tabel

|   | Universaalsus  | Kasutamise keerukus                                       | Võimalikud funktsioonid   | Arendamise võimalused   | Roboti mikrokontroller   |
|---|--|---|---|---|--|
| ROS   | ROS põhjal on tehtud üle 100 erineva roboti.   | Nõuab tugevat programmeerimise oskust kasutajalt.         | Lai valik erinevatest funktsioonide ja tööriistade pakenditest. ROS on alus, mille külge saab neid tööriistad ühendada. | ROS on avatud lähtekoodiga projekt, mille arendamisega tegeleb palju inimesi. | ROS kasutab keerulisi algoritme. Seega juhtimiseks kasutatakse mikroarvuteid või tööstuslikke kontrollereid. |
| Open robot                                  | Sobib kasutusele ainult esialgse projekti raames   | Suhteliselt kerge   | Funktsioonid katavad mobiilse platvormi juhtimist, RFID märkide lugemist ja kirjutamist.                                | Arendamine on keeruline, kuna tegemist pole avatud lähtekoodi projektiga.     | kasutatakse PIC18F4520 mikrokontrollerit   |
| ZigBee +Solar +Robot                        | Sobib kasutusele ainult esialgse projekti raames   | Suhteliselt kerge   | Funktsioonid katavad mobiilse platvormi juhtimist, kui ka manipulaatori juhtimist                                       | Arendamine on keeruline, kuna tegemist pole avatud lähtekoodi projektiga.     | Projektis kasutatakse PIC16F mikrokontrollerit   |
| Ardupilot                                   | Ardupilot põhjal on tehtud üle 100 erineva drooni.   | Käivitamise protsess on kirjeldatud väga üksikasjalikult. | Lai valik funktsioone drooni autonoomse liikumise jaoks.  | Kogukond pidevalt areneb  | Ardupilot pakub kasutada lai valik erinevatest plaatidest: Pixhawk, The Cube, Pixracer, F4BY jne.            |
| Robotite juhtimine MatLab kasutajaliidesega | MatLab on väga universaalne programm. ning sisaldab palju erinevaid tööriistu, mida saab ühendada. | Piisavalt keeruline. Tuleb tunda MATLAB keskkonda.        | Funktsioonide valik on väga lai. Sisaldab tööriistad mobiilsete robotite käitumise modelleerimiseks.                    | Matlab pidevalt arendab. Uue tööriista või funktsiooni lisamine on kerge.     | Sobivad kõik lihtsad programmeeritavad mikrokontrollerid, UART jadaliidese sisenditega.                      |
| BT Robot Controller v1.2                    | Sobib kõikidele robotitele ning on ettenähtud oma roboti kasutamiseks                              | Programmi kasutamine on kerge ja arusaadav.               | Funktsioonide valik on väike, kuid mängu roboti juhtimiseks piisab.   | Arendamine on keeruline, kuna tegemist pole avatud lähtekoodi projektiga.     | Sobivad kõik lihtsad programmeeritavad mikrokontrollerid, UART jadaliidese sisenditega.                      |

Lisaks tasub mainida, et ROS ja Ardupilot projektide suureks eeliseks on nende kogukonnad veebis, kus inimesed saavad küsida nõuandeid enda projekti kohta ning teha koostööd ühise projekti loomisel.

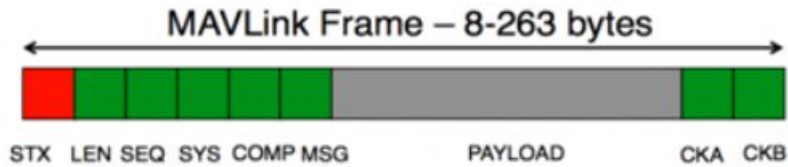
On tõenäoline, et robotika valdkonna algaja jaoks on parim programm BT Robot Controller v1.2. Selle programmi suureks eeliseks on kasutajaliidese lihtsus, võimalus kasutada erinevate mikrokontrolleritega. Programmi alumises osas kuvatakse side olekut ja viimati saadetud signaali väärtust. Puuduseks on roboti juhtimise tööriistade ja funktsioonide tagasihoidlik valik. Lisaks tööriistaid pole võimalik seadistada vastavalt projekti nõuetele. BT Robot Controller v1.2 ei ole avatud lähtekoodiga programm ning kasutajal puudub võimalus lisada uusi tööriistu enda projekti juhtimiseks.

### 1.3 MAVLink sideprotokoll

Distantsjuhtimise jaoks kasutatakse sideprotokollid, sest juhtimise, andurite, navigatsiooni süsteemide ja muid signaale tuleb teineteisest eristada. Sideprotokollid jaotatakse kaheks rühmaks: sümbolilised ja binaarsed.

1. Sümboliliste sideprotokollide eelisteks on loetavus ja realiseerimise lihtsus, kuid puuduseks on kasulikku koormuse ebaefektiivsus. Arvud tuleb konverteerida massiividest teisteks andmetüüpideks. Näiteks käsus *forward 100* iga sümboli suurus on 1 bait. Number 100 tuleb eraldada käsuraadist ning kasutamiseks, seda tuleb konverteerida sobivaks formaadiks [15].
2. Binaarsed protokollid saab jaotada kaheks tasemeks: konteineri tase ja andmete tase. Konteineri tase tegeleb paketi leidmisega baitide voos ning paketi sisu kontrollimisega. Andmete tase sisaldab ennast kasulikku infot. Teise taseme struktuur sõltub lahendatavatest eesmärkidest. Saadetise suurus võib olla fikseeritud või muutuv. Binaarse protokollid eelisteks on info efektiivne paigutus saadetises, kuid sellel on keerulisem loetavus ja realiseerimine. Tänapäeval binaarsed protokollid on rohkem levinud, kui sümbolilised [15].

MAVLink on binaarne sideprotokoll, mis on aktiivselt kasutusel ArduPilot projektide jaoks, et vahendada teavet drooni ja maapealse juhtimise jaama (*ground control*) vahel. Seda saab kasutada nii ArduPilot, kui ka ROS projektides. MAVlink teateid saab saata Wifi, 900 MHz raadio ja teiste jadaühendus vahenditega (*Serial communication*). Paketi formaat on toodud joonisel 1. Paketti vastuvõtmine pole garanteeritud, seetõttu saatja ei tea, kas käsk on täidetud või mitte. Saadetise pakk on jaotatud osadeks. Iga osa peale kasulikku koormuse suurus on 1 bait ehk 8 bitti. Kasulikku koormuse maksimaalseks suurus on 255 baiti [16].



Joonis 1. MavLink saadetise paketti formaat [16].

(STX) - Käsulähtepunkt. Versioonis v0.9 see on bait 0x55 HEX,

(LEN) - Käsulähtepunkti pikkus,

(SEQ) – Käsulähtepunkti lugeja. Aitab leida saadetise kaod,

(SYS) – Saadaja number,

(COMP) – Saadaja komponendi number, juhul kui saadajal on mitu mooduleid,

(MSG) – Saadetise tüüp, mis näitab kasutajale mida andmetega tuleb teha. Mavlink protokollis on oma xml tabel, kus kõik need tüübid on kirjeldatud,

(PAYLOAD) - käsulähtepunkt, sõltub saadetise tüübist,

(CKA ja CKB) on paketti kontrollimise bitid [16].

SYS ja COMP segmendid on alati täidetud, et süsteem tunneks, kust info pakett tuli. Juhul, kui side on püstitatud, siis iga seade saadab signaali (HEARTBEAT) sagedusega 1 Hz, sellega saab kontrollida, kas ühendus veel püsib [16]. Hetkel on olemas ka uuem versioon sellest protokollist – MAVLink2, mis võimaldab saata suurema mahuga informatsiooni.

MavLink protokoll arusaamist teeb keeruliseks asjaolu, et see on mõeldud nii lendavate droonide, maa robotite, kui ka droonilaadijate jaoks. Seega dokumentatsioonis on segamini pandud saadetised erinevate seadmete jaoks. Dokumentatsioon on mõeldud rohkem tarkvaraarendajale, kui tavakasutajale [17].

## 2.TESTRAKENDUS

Antud peatükis antakse detailne ülevaade testsüsteemi osadest, et kasutaja saaks valida millist osa ta soovib enda projektis kasutada ja millist mitte. Süsteem on näidislahendus, mida robotika õppijad ja harrastajad saaks võtta seda aluseks enda projekti arendamiseks. Kuna antud süsteem luuakse bakalaureusetöö raames, nimetatakse valmivat süsteemi näidislahendust testrakenduseks mobiilse platvormi juhtimiseks. Kasutajaliides on arendava süsteemi olulisem osa. Robotika harrastajad saavad ehitada oma mobiilset platvormi või sidemooduli või testimisplatsi, kuid kasutajaliides peab olema sama. Vastasel juhul tegemist on juba teise süsteemiga.

### 2.1 Testrakenduse funktsionaalsus

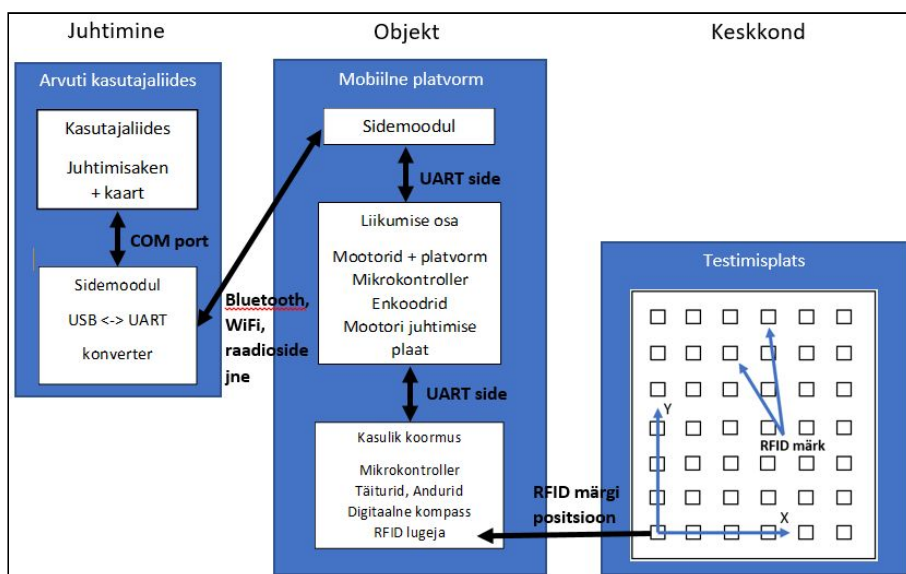
Toimiva süsteemi väljatöötamiseks on vaja kasutada olemasolevaid lahendusi. Süsteemi juhtimise eeskujuks võetakse BT Robot Controller v1.2 programmi kasutajaliides. Selle programmi eelised ja puudused on toodud peatükis 1.1.7. Selle peamiseks puuduseks on funktsioonide piiratud valik. Arendava süsteemi funktsioonide valik peab olema suurem ning peab sisaldama kõiki robotika projektides enim kasutatavaid funktsioone. Arendava süsteemi funktsioonid ja elemendid:

- Arendatav süsteem lubab juhtida nii mobiilse platvormi liikumist, kui ka selle peale paigaldatud täitureid/andureid.
- Mobiilse platvormi ja selle täitureite käitumist saab juhtida nii kasutajaliidese elementidega (nuppude ja liuguritega) kui ka programmi koodiga käsureast.
- Käsu saatmise aluseks on võetud Logo programmeerimiskeel. Selle eelisteks on interaktiivsus, modulaarsus, pikendatavus ja andmetüüpide paindlikkus [12]. Logo baaskäsud on täiesti asendamatud roboti juhtimise rakenduses. Logo programmeerimiskeele loogikal puudus on suur erinevus arvuti keskkonna ja reaalse elu vahel. Arvutis täidab robot kõike sisestatud käske. Reaalsel elus tuleb Logo käsude täitmist kontrollida. Arendatav süsteem ei lase saata uut käsu, kuni eelmine käsk on täidetud. Täpsemalt Logo programmeerimiskeelest saab lugeda peatükis 1.1.5.
- Käskudest saab kirjutada programmi, mida saab salvestada ning hiljem üles laadida.
- Sisestatud käsude programmi täitmise jälgimiseks saab kasutada olekuriba, mille peal kuvatakse käsu täitmist protsentides ning analoog kujul. Selline lahendus on arusaadav ning piltlik. Seega selline olekuriba on üks süsteemi jälgitavuse viisidest.

- Mobiilse platvormi positsiooni saamiseks saab kasutada RFID-märke. RFID-märgid on odavad ning nendega saab määrata positsiooni väga täpselt. Passiivsed RFID-märgid on töökindlad, sest töötamiseks ei ole vaja toitepinget. RFID-märgid on optimaalne positsioneerimise lahendus robotika harrastajate projektides, võrreldes teiste ruumis positsioneerimise viisidega.
- Andurite väärtused, käsud ning muud saadetised tuleb omavahel eristada. Selle eesmärgi täidavad sideprotokollid. MAVLink on levinud sideprotokoll droonide juhtimiseks, kuid tegemist on võrdlemisi palju mälu kasutava lahendusega, mille ülekandeid ei saa lugeda ilma vastava dokumentatsioonita. MAVlink ei sobi arendatavasse süsteemi oma keerukuse ja mahukuse tõttu, seetõttu kasutatakse lihtsamat sideprotokollit. Lähemalt MAVLink protokollist saab lugeda peatükis 1.3.
- Kasutajaliides kujutab ennast alust, mille peale asetatakse erinevaid tööriistaid. Tööriistadeks on erinevad nuppude, liugurite ja andurite kuvarite kombinatsioonid. Kasutaja ise valib tööriista, mida kasutatakse mobiilse platvormi ning selle peal asuva kasuliku koormuse juhtimiseks.
- Kasutaja saab muuta iga kasutajaliidese elemendi (nuppude, liugurite ja andurite) signaali vastavalt projekti nõuetele.
- Arendatav testsüsteem lubab juhtida mitut mobiilset platvormi korraga. Platvormid esitatakse ühel kaardil, et saaks hinnata nende omavahelist asendit ruumis. Käsud saab täita mitmel platvormil korraga. Ühe platvormi jaoks salvestatud käsud on avatud kasutamiseks ka teistel platvormidel.
- Mobiilne platvorm kasutab madala arvutusvõimsusega ja programmi mäluga mikrokontrollereid. Kuna need on tavalised soodsad ning on levinud robotika algajate projektides.

## 2.2 Testrakenduse moodulid

Mobiilse platvormi testrakendus koosneb kolmest moodulist: mobiilne platvorm, kasutajaliides selle platvormi juhtimiseks ning testimisplats. Testrakenduse moodulid ja nende omavahelised seosed on esitatud joonisel 2.



Joonis 2. Testrakenduse moodulid.

Mobiilne platvorm üldjuhul koosneb kahest osast: liikuv osa ja kasulik koormus. Liikuvaks osaks on mobiilse platvormi alus, mille eesmärgiks on kasuliku koormuse logistika. Aluse peale pannakse täiturid, milleks võib olla kinnitatud kaamera, andurid, haaratsid, teised täiturid või nende elementide kombinatsioon.

Üldjuhul liikuva osa ja kasuliku koormuse juhtimiseks kasutatakse üht juhtimise rakendust, sest ühe roboti juhtimine mitme seadmega pole mugav ega mõistlik. Erinevatel juhtimisseadmetel on oma eelised ja puudused:

- Distsantsjuhtimise pult tõstab projekti hinda. Puldi signaalide valik on piiratud mehhaanilise juhtkangi ja nuppude arvuga. Programmeeritavad puldid on eriti kallid.
- Nutitelefon projekti hinda ei tõsta, kuna need on levinud ning nende kasutusala on lai. Juhtimistarkvara versiooni saab vajadusel kiiresti uuendada. Nutitelefonidel on sisseehitatud *Bluetooth* moodul, mida saab ühendada mobiilse platvormiga. Teised sidemoodulid tuleb ühendada nutitelefoni *USB mini* pordi kaudu. Nutitelefoni ekraan on väike ning sellel on ebamugav avada mitu akent korraga.
- Arvuti funktsionaalsus ja arvutusvõimsus on suurem, kui distantsjuhtimise puldil ja nutitefonil. Roboti projekti arendamisel mikrokontrolleri programmi kirjutamiseks ja laadimiseks on vaja kasutada arvutit. Lisaks on arvuti ekraan suurem, on rohkem ühendatavaid sisendporte ning tarkvara versioonide uuendamine kiire. Paljudel arvutitel on olemas sisseehitatud *Bluetooth* moodul. Arvuti puuduseks on selle suurus, mille tõttu on välistingimustes kasutamine ebamugav. Kasutamaks muud side viisi, tuleb osta konverter antenniga.

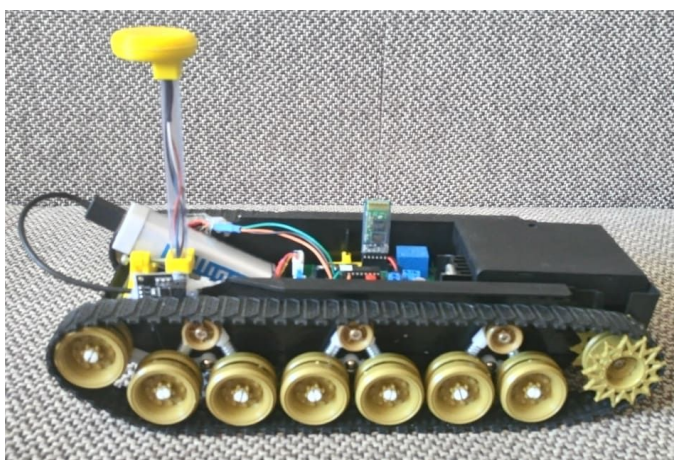
Antud töös otsustati kasutada juhtimisseadmena arvutit. Arvutile kirjutati kasutajaliidese programm, mis lubab juhtida mobiilset testplatvormi mööda COM-porti.

Sidemoodulid võivad olla erinevad. Kõige odavamaks variandiks on kasutada *Bluetooth* moodulit. Näiteks HC-06 moodulit. Neid *Bluetooth* mooduleid ei saa vahetult arvuti külge ühendada. Seetõttu tuleb kasutada USB - UART konverterit. UART see on side protokoll, mis on laialt levinud elektroonikas. Samuti kasutatakse UART side mobiilse platvormi erinevate moodulite infovahetuseks.

Roboti asukohainfot saab mitmel viisil: raadiomajakad, roboti tuvastamine kaameraga, Lidar süsteem, NFC märgid, RFID märgid jne. Enamus süsteemid on keerulised, kallid, kitsa kasutusala ja nõuavad lisatoidet, erandiks on passiivsed RFID märgid, mis töötavad ilma vooluallikata. Iga RFID kiibile kirjutatakse selle positsioon testimisplatsil ning need asetatakse vastavale asukohale põrandal. Robotile pannakse RFID märkide lugeja, millega saab märgi eelpool defineeritud positsiooni lugeda ning platvormi asukohta määrata.

## 2.3 Mobiilne testplatvorm

Süsteemi ja kasutajaliidese testimiseks valmistatakse mobiilne testplatvorm, milleks on roomikutel roboti alus. Roomikutel on suurem hõõrdumine ning rataste libisemist esineb vähem. Iga roomiku küljele on ühendatud enkoodrid. Töö jaoks valmistati kaks mobiilset testplatvormi. Mobiilse testplatvormi suurus on 0.3x0.15x0.08 m. Mobiilse testplatvormi pilt on toodud joonisel 3.



Joonis 3. Mobiilne testplatvorm.

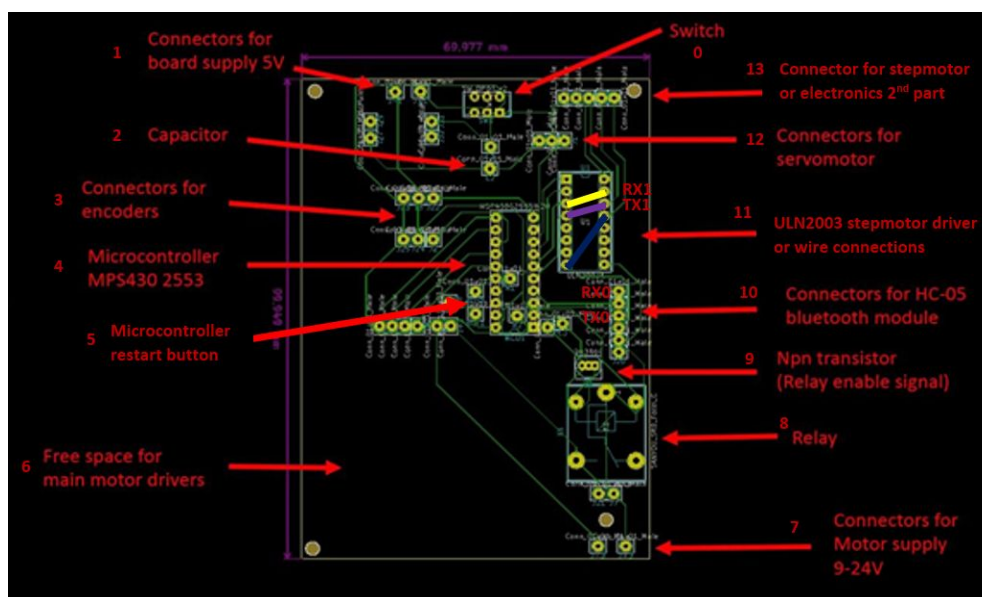


### 2.3.1 Elektroonika

Mobiilse platvormi elektroonika koosneb kahest osast:

1. Elektroonika I osa vastutab platvormi liikumise eest. Elektronika I osa jaoks valmistati trükkplaat selleks, et muuta elektroonika komponentide paigutust kompaktsemaks ning kontaktide ühendust kindlamaks. Trükkplaadi mikrokontroller on Texas Instruments tootevalikust mudel MSP430G2553. Trükkplaadis kasutatud komponentide nimekiri, koos selgitustega on toodud tabelis 2. Elektroonika plaate droonide juhtimiseks on palju erinevaid, kuid tavaliselt nendel on kõrge hind ning kõik komponendid on joodetud plaadi peale. Seega kasutaja ei saa võtta ja kasutada selle plaadi mikrokontrollerit ja side moodulit teistes projektides. Elektroonika I osa skeem on toodud lisa 8. Trükkplaadi laotus on toodud joonisel 4.
2. Elektroonika II osa on platvormi kasulik koormus, millel on oma mikrokontroller. Kasulikuks koormuseks võib olla andurid, manipulaator, haarats, kaamera liikuv kinnitus või muu täitur. Kasulikku koormust kasutaja valib ise vastavalt oma nõuetele. Elektroonika II jaoks trükkplaati pole mõistlik valmistada, kuna selle komponendid ning omadused sõltuvad kasutaja projektist. Antud töös teise osana olen kasutanud Arduino Uno või Esp8266 WeMos R1 D2 mini arendusplaati, mille külge ühendati RC522 RFID kaartide lugeja ning lms303 digitaalne kompass. Viimast 2 komponenti kasutatakse kaardil navigeerimiseks. Suhtlemine I ja II osade vahel toimub UART protokolliga. Antud töös kasutatud elektroonika II osa skeem on toodud lisa 8.

Loomulikult on võimalik valmistada ühe mikrokontrolleri juhitud mobiilne platvorm, aga madala arvutusvõimsusega mikrokontrollerite programmi täitmises võivad tekkida vead ja ajaline viide, kuna on üks algoritm, mis täidab väga paljusid ülesandeid. Teiseks elektroonika jaotamine mooduliteks võimaldab üht mobiilset platvormi lihtsamini ümber ehitada selliselt, et vahetatakse elektroonika II osa komponendid. Seetõttu on kontrollerite vahel jaotatud ülesannetega lahendus kiirem ning tagab teatava universaalsuse ja võimaluse täitureid vahetada. Seega süsteem, milles ülesanded on jaotatud kontrollerite vahel on parem. Elektroonika skeemide ja trükkplaadi disainimiseks valiti KiCad trükkplaatide arendamise keskkond. KiCad on tasuta tarkvara elektroonika skeemide ja trükkplaatide laotuse disainimiseks. Väljundis saadakse gerber failid CNC pinkide jaoks, mille alusel hakatakse trükkplaati valmistama.



Joonis 4. Elektroonika I osa trükkplaadi laotus KiCad programmis koos selgitustega.

Tabel 2. Elektroonika I osa komponentide nimekiri koos selgitusega.

| Nr. | Komponendi nimetus  | Selgitus  |
|-----|---|---|
| 1   | Toite kontaktid mikrokontrolleri jaoks, Mikrokontrolleri toiteallikad | Mobiilse testplatvormi valmistamisel selgus, et ühest toiteallikast ei piisa, kuna mootorite suur energiatarve põhjustas häireid sidemooduli töös, lisati veel teine toiteallikas pingega 5 VDC, millega toidetakse mikrokontrollerit, releed, sidemoodulit ning kõiki täitureid ehk elektroonika II osa. |
| 2   | Kondensaator  | Kondensaatorit mahtuvusega 100 uF, kasutatakse kontrolleri ja sidemooduli sujuvaks tööks.   |
| 3   | Enkoodrite sisend   | Enkoodrite signaalid lähevad otse mikrokontrolleri <i>interrupt</i> sisenditesse. Programmi kood interrupt sisendite ühendamiseks kasutati Energia näidet [2].  |
| 4   | Mikrokontroller   | Elektroonika I osa mikrokontrollerina kasutatakse Texas Instruments MSP430G2553 kontrolleri. Sellel mikrokontrolleril on 16 kB <i>flash</i> mälu ja 512 RAM mälu. Kõik kontrolleri 14 <i>GPIO</i> sisend/väljundit toetavad <i>interrupt</i> sündmust. Kontrolleri toide pingeks on 3-6V [35].            |
| 5   | Mikrokontrolleri taaskäivitamise nupp                                 | Katkestab ning käivitab algusest mikrokontrolleri töö   |
| 6   | Mootorite juhtplaat   | Motoritena saab kasutada nii alalisvoolu mootoreid kui ka samm-mootoreid. Mootorite ühendamise joonis on esitatud lisas 8.  |
| 7   | Toiteallikas mootorite jaoks  | Seda toiteallikat kasutatakse mootorite toitmiseks. Selle pingeväärtus sõltub kasutatavast mootorist. Alalisvoolumootori juhtimise plaat L298N pingevahemik on 9-24V.   |

| Nr. | Komponendi nimetus  | Selgitus   |
|-----|---|--|
| 8   | Relee   | Releega hakatakse lülitama mootorite toidet  |
| 9   | NPN tüüpi transistor  | Transistori kasutatakse kontrolleri signaali võimendamiseks selleks, et lülitada releed.   |
| 10  | Sisend sidemooduli ühendamiseks   | Andmevahetuseks platvormiga sobivad kõik UART R232 standardiga sidemoodulid. Nendeks võivad olla näiteks HC-06 bluetooth moodulid, TB387 2.4GHz sidemoodulid, ESP 01 moodulid, ZigBee moodulid jne. HC-06 moodulite ühendamine AT-käskudega on kirjeldatud lisa 8. UART side on valitud seetõttu, et enamus mikrokontrolleritest omavad sisend/väljundeid selle side tüübi jaoks. Antud töös kasutati HC-06 ja TB387 sidemoduleid. |
| 11  | Kiibi pesa ULN2003 samm-mootori juhtimise kiibile või juhtmetele          | Kiibi pesale saab vajaduse korral ühendada samm-mootori juhtimise kiibi ULN2003 või kasutada need sisendite pesad kontaktide ühendamiseks.   |
| 12  | Servomootori sisend   | Servomootoreid kasutatakse teistes projektides mobiilse platvormi esimeste rataste suuna muutmiseks.   |
| 13  | Sisend samm-mootori jaoks või UART ühendamiseks elektroonika teise osaga. | Seda sisendit saab kasutada samm-mootori juhtimiseks või teise mikrokontrolleriga ühendamiseks UART sideprotokolliga. UART side jaoks on vaja ühendada 4 juhet: +5V, GND, TX ja RX.  |
| 0   | Lüliti  | Lülitab mikrokontrolleri sisse   |

Sidemooduli ühendamiseks on kolm viisi:

1. Sidemoodul ühendatakse elektroonika I osa trükkplaadile ning teise plaadiga suhtlemine toimub esimese plaadi kontrolleri kaudu. Infovahetus kahe mikrokontrolleri vahel toimub UART side abil.
2. Sidemoodul ühendatakse elektroonika teise osa trükkplaadile ning esimese plaadiga suhtlemine toimub teise plaadi kontrolleri kaudu. Infovahetus kahe mikrokontrolleri vahel toimub UART side abil.
3. Elektroonika mõlemal osal on oma sidemoodul. Mobiilset platvormi saavad juhtida kaks operaatorit, üks vastutab liikumise eest ja teine vastutab täiturite eest.

Testrakenduse eraldi teemaks on sidemooduli ühendamine arvuti külge. Tavalisel arvutil UART sisend/ väljundid puuduvad. Seega tuleb kasutada USB-UART konverterit. USB-UART konverteriks saab valida CH340G kiibi. USB-UART konverteril on eraldi toite väljundid 5 V,

seega üht side moodulit sai toita vahetult arvutist. USB-UART konverteri joonis on toodud lisa 8. Tabelis 3 on esitatud riistvara komponendid, mida kasutati testimisplatsi, 2 sidemooduli ja 2 mobiilse testplatvormi valmistamiseks.

Tabel 3. Komponentide tabel

| Komponent                          | Tüüp                                | Kogus |
|------------------------------------|-------------------------------------|-------|
| Launchpad programmaator            | MSP430G perekonna kontrolleri jaoks | 1     |
| Mikrokontroller                    | MSP430G2553                         | 2     |
| Arendusplaat                       | Arduino Uno                         | 1     |
| Arendusplaat                       | ESP8266 WeMos R1 D2 mini            | 1     |
| Trükkplaat                         | PBC, värv: roheline, tekstoliidist  | 2     |
| Bluetooth moodul                   | HC-06                               | 2     |
| USB to UART konverteri moodul      | CH340                               | 2     |
| Sidemoodul                         | TB387                               | 2     |
| Takisti                            | 330 Ω                               | 2     |
| Lüliti                             | SPDT                                | 2     |
| Lüliti                             | surunupp                            | 2     |
| Pin Socket                         | 20 sisendit                         | 2     |
| Pin Socket                         | 16 sisendit                         | 4     |
| Kondensaator                       | 100uF                               | 2     |
| Alalisvoolumootori juhtimise plaat | L298N                               | 2     |
| Relee                              | Songle SRD-05VDC-SL-C               | 2     |
| Transistor                         | 2N2222                              | 2     |
| Roboti alus                        | roomikutel, SN100                   | 2     |
| RFID lugeja                        | RC522                               | 2     |
| RFID märgid                        | RFID kaart 0.085*0.055 m            | 90    |

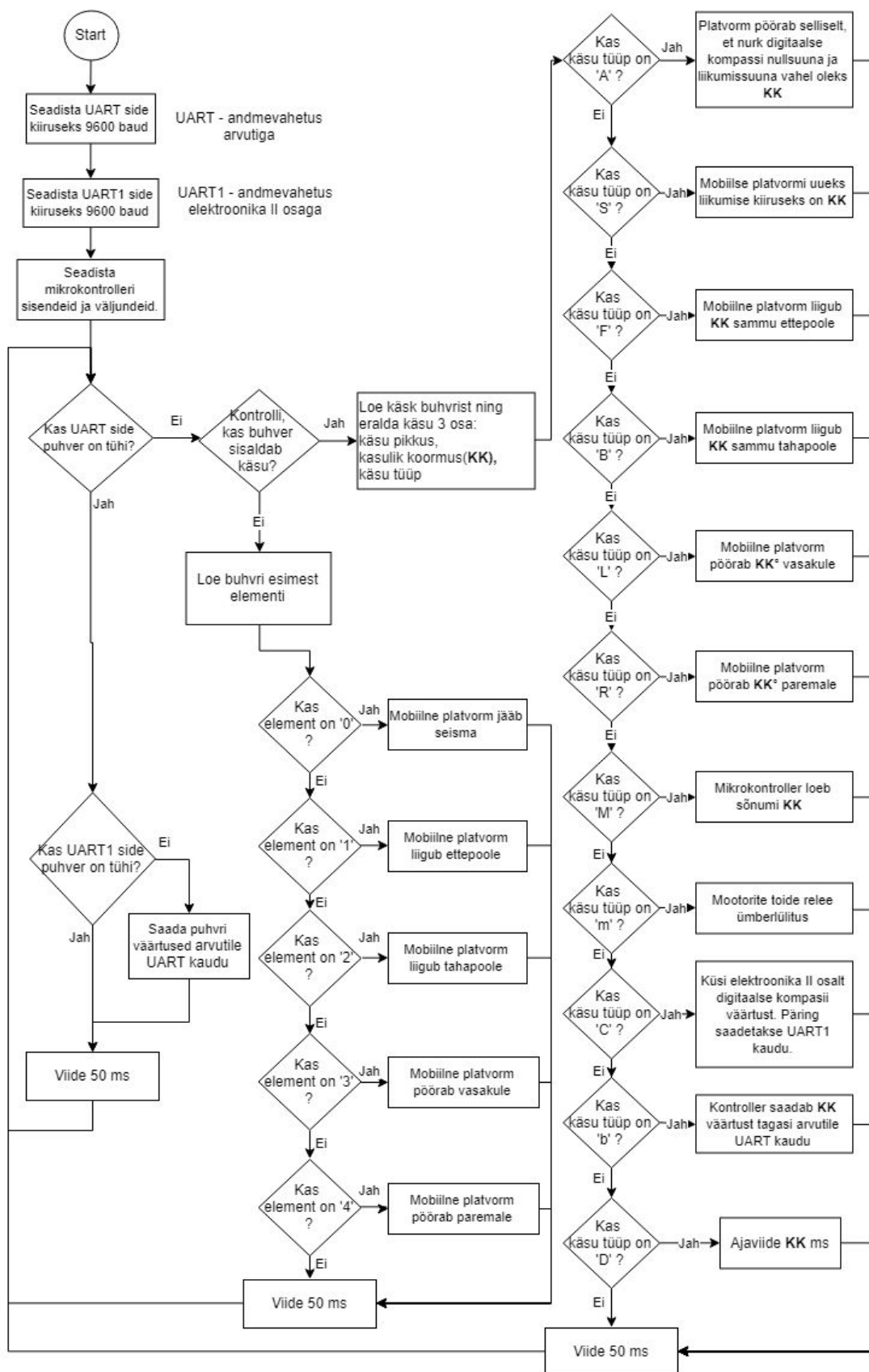
### 2.3.2 Mikrokontrollerite püsivara

Mobiilse testplavormi püsivara ehk mikrokontrollerile laetav programm on arendatava süsteemi oluline osa. Mobiilne platvorm kasutab kahte mikrokontrollerit, mida kasutatakse vastavalt elektroonika I ja II osade juhtimiseks. See eeldab, et kirjutatakse mõlema mikrokontrolleri jaoks programm. Täpsemalt kasutatavast elektroonikast saab lugeda peatükis 2.3.1.

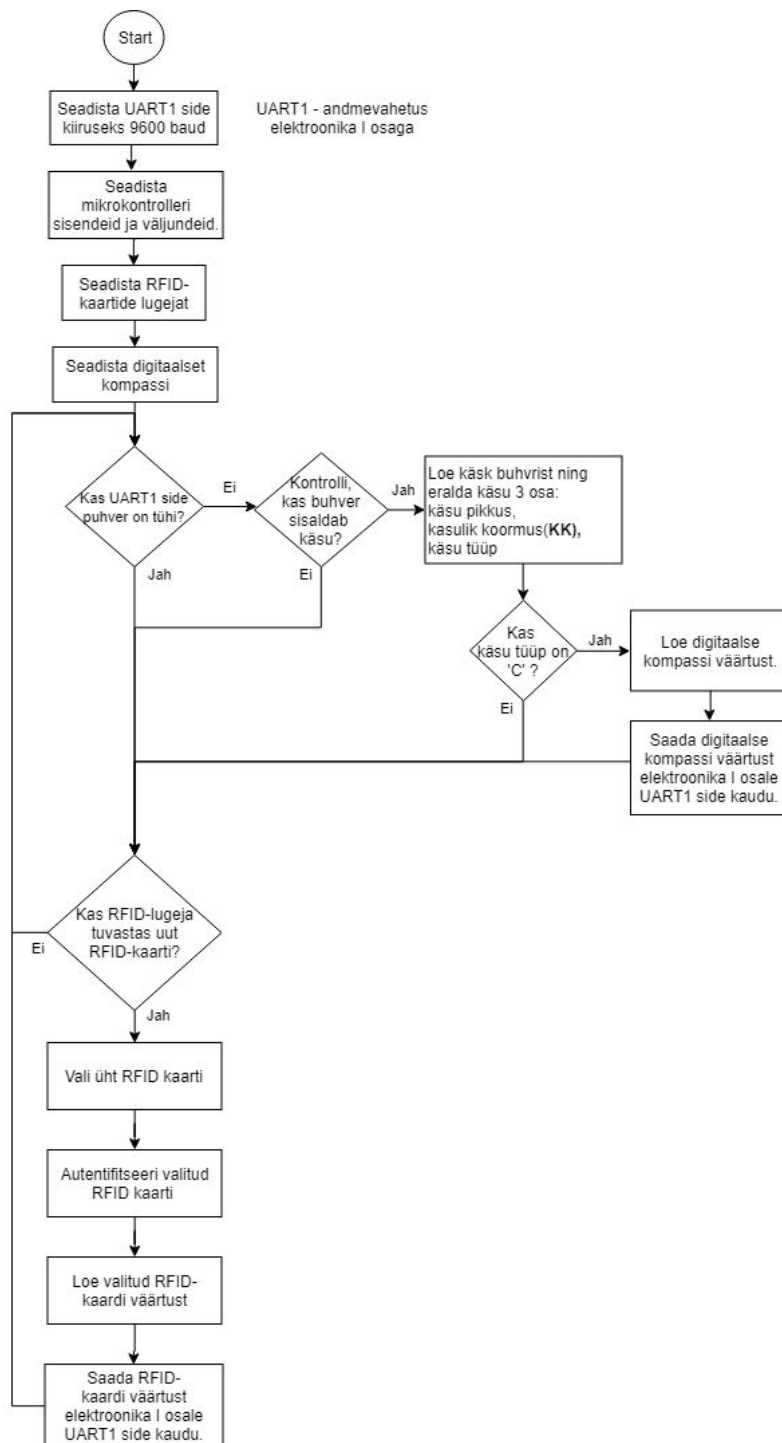
Elektroonika I osa mikrokontrolleri programm juhib mobiilse platvormi liikumist. Erinevate käsude eristamiseks kasutatava sideprotokolli kohta on kirjutatud peatükis 2.4. Programm pidevalt ootab jadapordi kaudu sisenevaid käskusid, mida programm tuvastab ja täidab. MSP430G2 tüüpi mikrokontrollerite programmeerimiseks kasutati Energia mikrokontrollerite tarkvara arendamisplatvormi.

Elektroonika II osa mikrokontrolleri programm loeb andurite väärtusi. Anduriteks on RFID-kaartide lugeja ja digitaalne kompass. Lugeja tuvastab aktiivselt RFID-kaarte. Kui RFID-kaart on tuvastatud, siis program püüab lugeda kaardi peale kirjutatud positsiooni asukoha väärtust, mis saadakse elektroonika I osale. Digitaalse kompassi lugemiseks tuleb saata päring. Arduino UNO ja ESP8266 WeMos mini arendusplaatide mikrokontrollerite programmeerimiseks kasutati Arduino IDE mikrokontrollerite tarkvara arendamise platvormi.

Kasutaja saab võtta need programmid enda projekti aluseks. Püsivara põhifunktsionaalsust kirjeldavate algoritmide plokskeemid on toodud joonisel 5 ja joonisel 6.



Joonis 5. Elektroonika I osa mikrokontrolleri programmi põhifunktsionaalsuse plokk skeem.



Joonis 6. Elektroonika II osa mikrokontrolleri programmi põhifunktsionaalsuse plokk skeem.

## 2.4 Sideprotokoll ja andmevahetus

Infovahetuse jaoks kasutatakse sideprotokollid, sest juhtimise, andurite, navigatsiooni süsteemide ja muid signaale tuleb teineteisest eristada. Sideprotokolli saatmise paketi struktuur võiks

sarnaneda MAVlink protokolliga. MAVlink protokoll kirjeldusega saab tutvuda peatükis 1.3. MAVlink protokoll puuduseks on halb loetavus ning keerulisus. MAVlink protokoll saadetisele käsu tõlgendamiseks on vaja vastavat dokumentatsiooni. Saadetise loetavuse tõstmiseks kasutatakse sümbolilist protokoll. Täpsemalt sideprotokollide eri tüüpidest on kirjutatud peatükis 1.3. Antud peatükis püütakse luua lihtsamat ja loetavat protokoll MAVLink protokoll struktuuri võttes. Tabelis 4. on toodud sideprotokoll saadetise paketi segmentid. Tabelis 5. on toodud saadetise segmentide kirjeldus.

Tabel 4. Side protokoll saadetise pakett

|                    | Saadetise pakett |     |     |         |     |
|--------------------|------------------|-----|-----|---------|-----|
| Segmentide nimetus | STX              | LEN | MSG | PAYLOAD | CKB |
| Suurus baitides    | 1                | 1   | 1   | 0-205   | 1   |

Tabel 5. Sideprotokoll saadetise segmentide kirjeldus

| Segmenti lühend | Segmenti täisnimetus       | Selgitus   |
|-----------------|----------------------------|--|
| STX             | Käsu alguse bait           | Selle väärtuseks on 0x55 HEX formaadis ehk täht 'U' ASCII tabelis                                |
| LEN             | Kasulik koormuse pikkus    | Selle väärtus algab baiti väärtusest 0x30 HEX ehk täht '0' ASCII tabelis                         |
| MSG             | Saadetise tüüp             | Saadetise tüüp näitab süsteemile mida andmetega tuleb teha. Saadetise tüüpe saab näha tabelis X. |
| PAYLOAD         | kasulik koormus            | Andmed nende sisu ja kasutamise viis sõltub saadetise tüübist                                    |
| CKB             | paketti kontrollimise bait | Kasulik koormuse ülemine bait  |

MAVlink protokoll aktsepteerib kasutaja poolt koostatud käske, see tähendab et paar MSG saadetise tüüpi on reserveeritud kasutaja käskude tarbeks. Antud sideprotokoll korral otsustati teha vastupidi. Üheksa MSG saadetise tüüpi on baastüübid. Baastüüpide aluseks võib nimetada Logo programmeerimise keele baaskäskusid. Viimased on kirjeldatud peatükis 1.1.5. Ülejäänud 246 baiti on vabad ning kasutaja saab programmeerida nendele oma tegevust. Tabelis 6. on esitatud saadetise tüüpide näited ja tabelis 7. on toodud saadetise näited. Käskude saatmiseks kasutatakse nuppe, liugureid ja käsuresisendit. Erinevatest juhtimise viisidest saab täpsemalt lugeda peatükis 2.5.1



Tabel 6. MSG tüüpide tabel

| Saadetise tüübi täisnimetus | Saadetise tüüp char formaadis | Saadetise saajad    | Selgitus   |
|-----------------------------|-------------------------------|---------------------|--|
| <i>Position</i>             | P                             | arvuti              | Saadetis platvormi positsioonist. Seda kasutatakse platvormi liikumiseks kaardil.  |
| <i>Angle</i>                | A                             | arvuti              | Käsk platvormile, et tuleb valida uut suunavektor. Suunavektor on antud nurgakraadides.  |
| <i>Compass</i>              | C                             | arvuti/<br>platvorm | Saadetis arvutile. See sisaldab kompassi nurga väärtust kraadides. Käsku kasutatakse kompassi hetkeväärtuse uuendamiseks.                          |
| <i>Forward</i>              | F                             | arvuti/<br>platvorm | Käsku kasutatakse edasi liikumiseks, lisainformatsiooniks käsu juures on number, mis näitab mitu sammu platvorm edasi liigub.                      |
| <i>Back</i>                 | B                             | arvuti/<br>platvorm | Käsku kasutatakse tagasi liikumiseks, lisainformatsiooniks käsu juures on number, mis näitab, mitu sammu platvorm tagasi liigub.                   |
| <i>Right X</i>              | R                             | arvuti              | Käsk platvormile, et tuleb liikuda X° paremale.  |
| <i>Left X</i>               | L                             | arvuti              | Käsk platvormile, et tuleb liikuda X° vasakule.  |
| <i>Speed</i>                | S                             | arvuti/<br>platvorm | Käsk platvormile, et tuleb liikuda määratud kiirusega. Kinnitus arvutile, et platvorm liigub määratud kiirusega.                                   |
| <i>Message</i>              | M                             | arvuti/<br>platvorm | Saadetis platvormile või arvutile  |
| <i>Delay</i>                | D                             | platvorm            | Saadetis platvormile ajaviitega millisekundites  |
| <i>Action</i>               | kasutaja poolt valitud        | platvorm            | Kasutaja tegevusele reserveeritud MSG saadetise tüüp. Käsk platvormile, et tuleb alustada tegevust.  |
| <i>Stop/Start</i>           | kasutaja poolt valitud        | platvorm            | Kasutaja tegevusele reserveeritud MSG saadetise tüüp. Käsk platvormile, et tuleb lülitada mootorit sisse või välja.                                |
| <i>Slider</i>               | kasutaja poolt valitud        | arvuti/<br>platvorm | Täiturite juhtimiseks reserveeritud 5 saadetise MSG tüüpi. Käsk platvormile, täituri uue positsiooniga. Tagasiside arvutile täituri positsiooniga. |
| <i>Sensor</i>               | kasutaja poolt valitud        | arvuti/<br>platvorm | Sensorite juhtimiseks reserveeritud 5 MSG saadetise tüüpi.   |
| <i>Buttons</i>              | kasutaja poolt valitud        | platvorm            | Kasutaja tegevusele reserveeritud MSG saadetise tüüpi. Kokku 8 tükki iga nuppu jaoks Käsk platvormile, et tuleb alustada mingit tegevust.          |

Tabel 7. Saadetise näited

| Saadetis char massiivina | Kasulik koormus | Kirjeldus  |
|--------------------------|-----------------|--|
| U6PX100Y20               | X100Y20         | Saadetis arvutile: Platvorm asub positsioonil X100 Y20 |
| U4F100;                  | 100;            | Saadetis platvormile: Liigu 100 sammu ettepoole        |
| U5D1000;                 | 1000;           | Saadetis platvormile: Oota 1000 ms                     |
| U3L45;                   | 45;             | Saadetis platvormile: Pööra 45° vasakule               |
| U9MObstacle!             | Obstacle!       | Saadetis arvutile: Eespool on takistus!                |
| U<MKuidas laheb?         | Kuidas laheb?   | Saadetis platvormile: Kuidas laheb?                    |

## 2.5 Kasutajaliidese kirjeldus

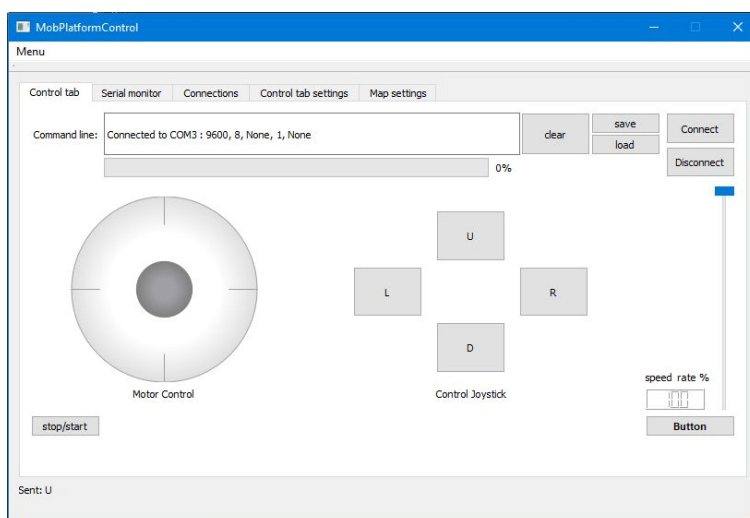
Kasutajaliides koosneb kahest aknast - juhtimise ning kaardiga aken. Juhtimise aknas saab avada uued juhtimise aknad, mida on kokku saab olla neli. Kasutajal on võimalik vajadusel kaardi aken sulgeda. Platvormi distantsjuhtimiseks on vaja läbida neli etappi: leida COM-port andmevahetuseks, avada vastav port, luua kasutajaliidest, mis lubab saata ning lugeda signaale COM-pordist. Kasutajaliidese juhtimise aken on jaotatud 5 vahekaardiks:

1. *Control tab* vahekaart - sellelt saab juhtida platvormi liikumist erinevate nuppudega.
2. *Serial monitor* vahekaart - sellelt saab lugeda kõiki mobiilsest platvormist tagastatud signaale.
3. *Connections* vahekaart - sellel toimub jadaliidese parameetrite seadistamine
4. *Control tab settings* vahekaart - kasutaja saab seadistada mobiilse platvormi juhtimiseks kasutatavad nupud.
5. *Map settings* vahekaart - siin saab seadistada platvormi kuju parameetrid kaardil.

Graafilise kasutajaliidese arendamisel kasutati Qt Creator tarkvaraarendus platvormi. Qt Creator programmi lühitutvustus on toodud lisas 1. Kasutajaliidese programmeerimiseks kasutati C++ keelt.

### 2.5.1 Juhtimise vahekaart

Juhtimise vahekaarti kasutatakse roboti liigutamiseks ning selle peale paigaldatud täiturite ning andurite kasutamiseks. Juhtimise vahekaardi pilt on esitatud joonisel 7. Juhtimise vahekaardi nuppude toimingu kirjeldus on esitatud tabelis 8.



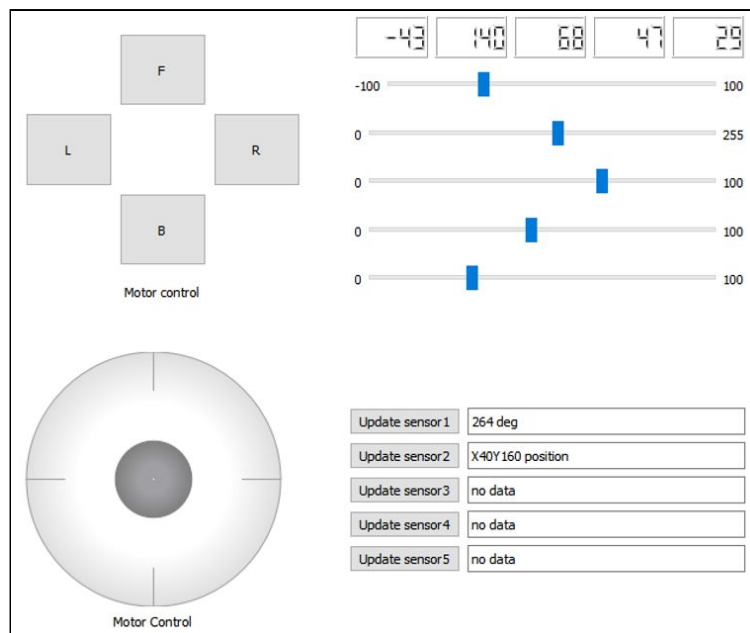
Joonis 7. Kasutajaliidese juhtimise vahekaart.

Tabel 8. Juhtimise vahekaarti peamised elemendid

| Elemendi nimetus        | Elemendi toimingu kirjeldus  |
|-------------------------|--|
| <i>Connect</i>          | Nupp tekitab ühenduse seadme ja programmi vahel COM-pordi kaudu. Ilma ühendusest ei ole programmiliselt võimalik platvormi juhtida ega infot vastu võtta.  |
| <i>Disconnect</i>       | Nupp ühendab COM-porti lahti. Pärast sellele nupule vajutamist antud porti saab kasutada teiste ülesannete jaoks.  |
| <i>Save</i>             | Nuppu kasutatakse käsude salvestamiseks. Täpsemalt käsu salvestamisest saab lugeda lisa 4.   |
| <i>Load</i>             | Nuppu kasutatakse käsude laadimiseks. Täpsemalt käskude laadimisest saab lugeda lisa 4.  |
| <i>Motor Control</i>    | Tööriist platvormi juhtimiseks. Kasutaja valib nuppude rühma või juhtkangi.  |
| <i>Control Joystick</i> | Tööriist platvormi täiturite ning andurite kasutamiseks. Kasutajal on neli valikut: nuppude rühm, juhtkang, liugurid ja andurite kuvarid.  |
| <i>Command line</i>     | Käsu sisend, mida saab kasutada oma käsude saatmiseks. Täpsemalt käsu saatmisest saab lugeda peatükis 2.5.2. Käsu sisestamiseks tuleb vajutada <i>Enter</i> klahvile. Lisaks seda kasutatakse erinevate teadete kuvamiseks. Näiteks:<br><i>Port is not connected</i> - COM-port pole ühendatud<br><i>Connected to COM25 : 9600, 8, None, 1, None</i> - COM-port 25 on valmis lugemiseks/kirjutamiseks. |
| <i>Button</i>           | Nupp, mille nimetus ning toiming on kasutajal võimalik muuta.  |
| <i>Speed rate</i>       | Kuvab platvormi hetkekiirust protsentides. 100% on platvormi maksimaalne kiirus, 0% platvorm ei liigu ning kiirus on null.   |
| <i>Käsu olekuriba</i>   | Olekuriba kuvab programmi täitmist protsentides. Selle arvutamiseks kasutatakse valemit 1. Täpsemalt käsu olekuriba täitmisest saab lugeda peatükis 2.5.2.   |

Juhtimise tööriistadeks on erinevad nuppude, anduri kuvarite ja liugurite kombinatsioonid, mida kasutatakse mobiilse platvormi ning selle peal paigaldatud täiturite/andurite juhtimiseks. Õiget tööriista saab valida *Control tab settings* vahekaardis. Hetkel lisatud tööriistade pildid on esitatud joonisel 8. Hetkel on võimalikud järgmised tööriistad:

1. Nuppude kombinatsioon. See on ettenähtud mobiilse platvormi liikumiste juhtimiseks, kuid võimalik täita ka muid ülesandeid.
2. Juhtkang. See on ettenähtud mobiilse platvormi liikumiste juhtimiseks, kuid võimalik täita ka muid ülesandeid. Iga juhtkangi suunale saab määrata signaali. Juhtkangi lahti laskmisel, läheb see tagasi null asendisse ja saadab *stop* signaali. *Minimum/Maksimum* märgid näitab liuguri liikumise vahemikku nurgakraadides. Juhtkangi programmi kirjutamiseks on kasutatud teise autori koodi [1].
3. Liugurid. Nende eesmärgiks on erinevate täiturite juhtimine. Näiteks haaratsite ja manipulaatorite juhtimiseks. Paljud haaratsid ja manipulaatorid kasutavad lülide pööramiseks servomootoreid. Antud tööriist lubab korraga juhtida 5 servomootorit.
4. Andurite väärtuste lugemiseks on kasutuses eraldi tööriist. Kokku saab lugeda 5 andurit. Iga vajutus nupule saadab platvormile päringu anduri väärtuse kohta. Samas on võimalus saata päringut automaatselt igal sekundil. Anduri väärtuseks võivad olla nii numbrid, kui ka tähed. Lisaks saab määrata väärtuste ühikuid.



Joonis 8. Võimalikud tööriistad.

## 2.5.2 Programmi täitmine käsureast

Traditsioonilise Logo programmeerimiskeele süntaksi käsud on teineteisest eraldatud uue reaga. Selline lahendus tõstab programmi loetavust, kuid võtab rohkem ruumi juhtimise aknal. Seega käskude eraldamiseks kasutatakse “;” märki. Käsk koosneb kolmest osast: käsu tüüp *MSG*, käsu parameeter ja käsu lõpu märk. Käskude ja saadetise *MSG* tüüpide tähised on samad. Need on kirjeldatud peatükis 2.4. Programmide ja käskude näited saab vaadata lisas 9. Programmi täitmist saab jälgida käsu olekuribal. Programmi täitmist kuvatakse protsentides ning analoog kujul olekuribal.

$$S = \frac{n}{N} * 100\% \quad (1)$$

kus,  $S$  – Programmi täitmise staatus protsentides,

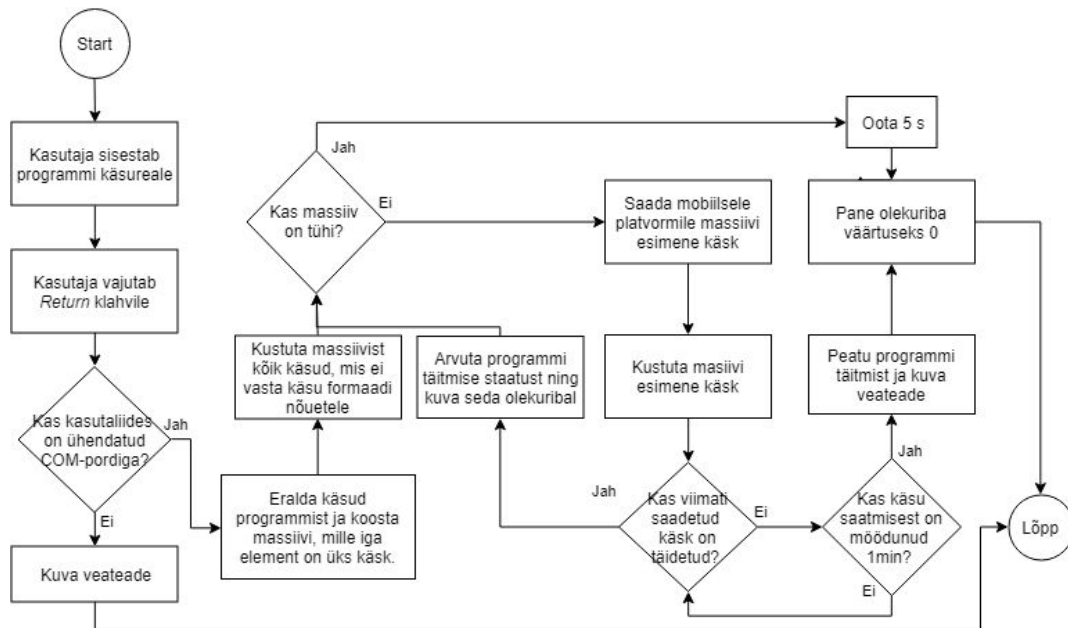
$n$  – Viimati täidetud käsu number,

$N$  – Tuvastatud käskude arv sisestatud programmis.

Käsude programmi saatmise järjekord:

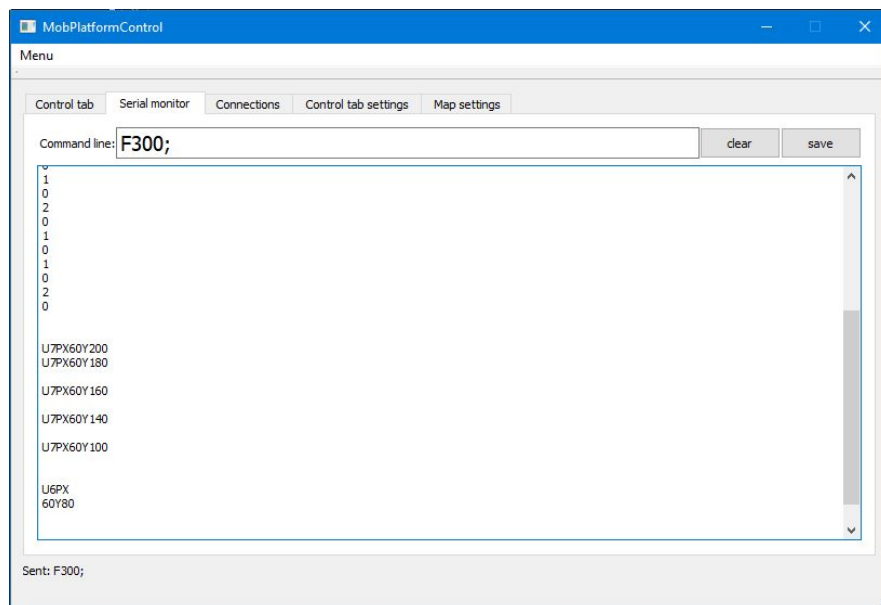
1. Tuvastatud käskudest koostatakse massiiv, mille iga element on üks käsk.
2. Platvormile saadakse käsk.
3. Mikrokontroller tuvastab käsu ning hakkab seda täitma.
4. Käsu täitmise kinnitus saadakse tagasi arvutile.
5. Arvuti arvutab programmi täitmise staatuse protsentides ning kuvab seda olekuribal.
6. Programmi järgmine käsk saadetakse mobiilsele platvormile. Seega uue käsu käivitamine pole võimalik enne, kui eelmine käsk on täidetud.

Käsurealt sisestatu kontrollimise ja kontrolleriile saatmise algoritmi plokk skeem on esitatud joonisel 9. Programmide näited on esitatud lisas 9.



Joonis 9. Programmi saatmise algoritmi plokk skeem.

## 2.5.5 Jadaliidese monitoorimine



Joonis 10. Jadaliidese monitoorimise vahekaart.

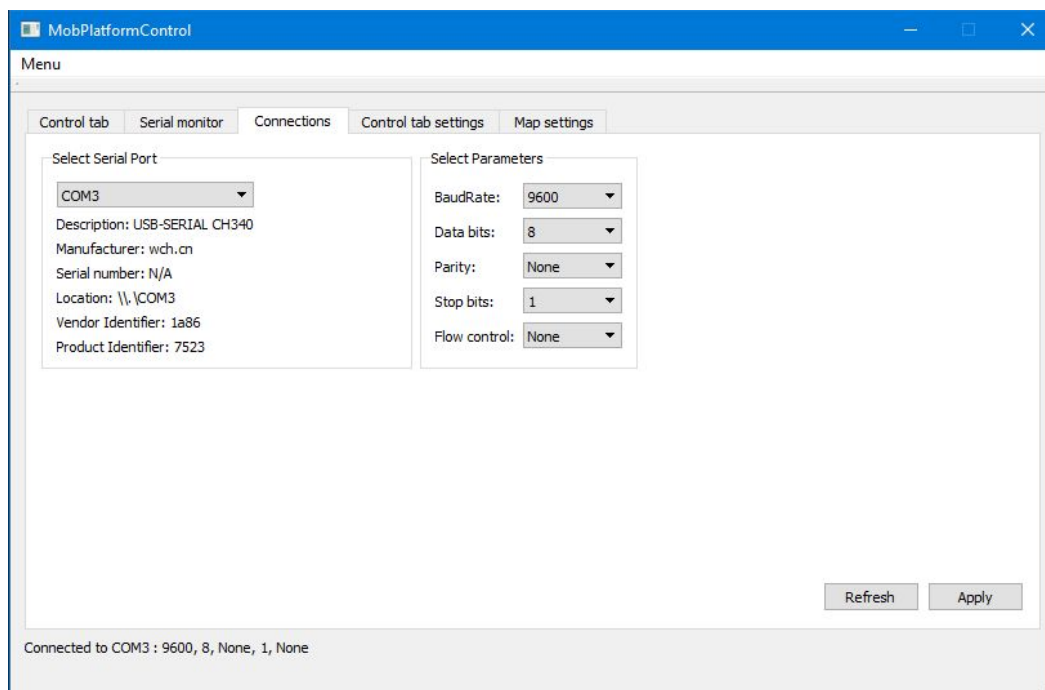
Jadaliidese monitori kasutatakse infovahetuse signaalide kuvamiseks. Kasutaja näeb kõiki mobiilsest platvormist tagastatud signaale ning arvuti poolt viimati saadetud signaale. Jadaliidese monitoorimise vahekaart on toodud joonisel 10 ja selle elementide kirjeldus on toodud tabelis 9.

Tabel 9. Jadaliidese monitoorimise vahekaardi elemendid

| Elemendi nimetus    | Elemendi toimingu kirjeldus   |
|---------------------|---|
| <i>Command line</i> | Käsu sisend, mida saab kasutada oma käsude saatmiseks. Kõik signaalid saadetakse muudetuseeta. Käsk sisestatakse vajutades <i>Enter</i> klahvile. |
| jadaliidese monitor | Siin kuvatakse kõik COM-pordist tulnud signaalid, mida saab kopeerida ning kasutada mujal.  |
| <i>Save</i>         | Nupp kasutatakse käsude salvestamiseks.   |
| <i>Clear</i>        | Nupp kasutatakse käsu sisendi ja jadaliidese monitori tühistamiseks.  |

## 2.5.6 Jadaliidese seadistamine

Arvuti infovahetus sidemooduliga toimub COM-pordi kaudu. Sidemoodul on seade, mida kasutatakse arvuti ühendamiseks mobiilse platvormiga. Sellel ühendusel on oma parameetrid, mida saab valida jadaliidese parameetrite seadistamise vahekaardil. Sellega saab tutvuda joonisel 11. Vahekaardi graafiliste elementide toiminguga saab tutvuda tabelis 10. Kasutaja määrab COM-porti, mille külge on ühendatud sidemoodul. Jadaliidese seadistamiseks kasutati Qt Creatori näide *Terminal* koodi[24].

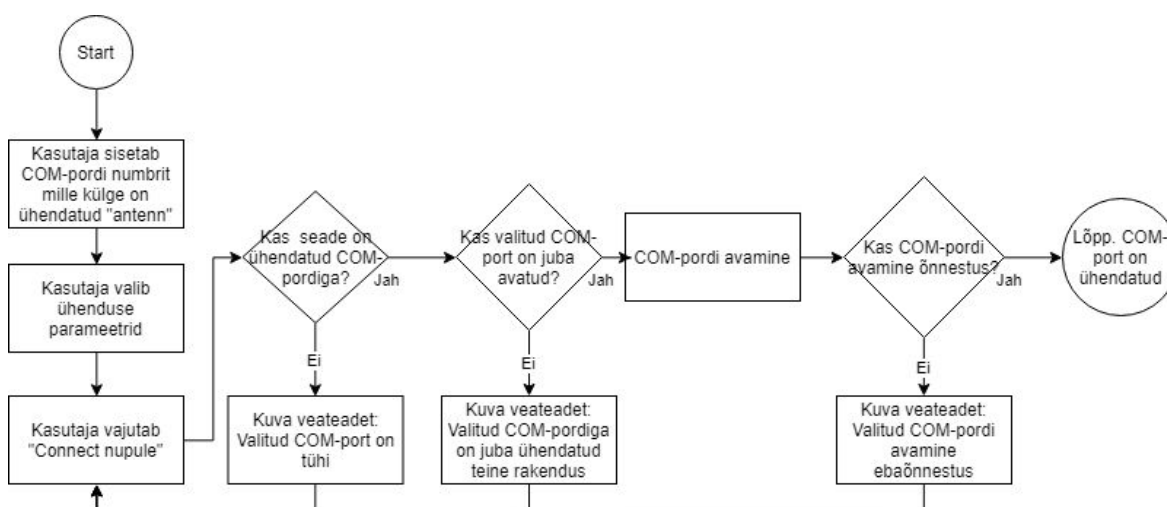


Joonis 11. Jadaliidese parameetrite seadistamise vahekaart.

Tabel 10. Jadaliidese parameetrite seadistamise vahekaardi elemendid

| Elemendi nimetus          | Elemendi toimingu kirjeldus   |
|---------------------------|---|
| <i>Select Serial Port</i> | Selle valikuga määratakse infovahetuse jaoks kasutatav COM-port. Lisaks on võimalik näha seadme ning tootja nimetust.   |
| <i>Select Parameter</i>   | Selle valikuga määratakse side parameetreid. Olulisemateks parameetriteks on <i>baud</i> kiirus ja bittide arv. Baud on side kiiruse mõõtmise ühik elektroonikas, mis näitab mitu korda signaal muutub 1 sekundi jooksul. Bittide arv näitab mitu bitti saadetakse korraga. |
| <i>Refresh</i>            | Nupp uuendab kõikide ühendatud portide nimekirja  |
| <i>Apply</i>              | Nupp rakendab tehtud muudatused   |

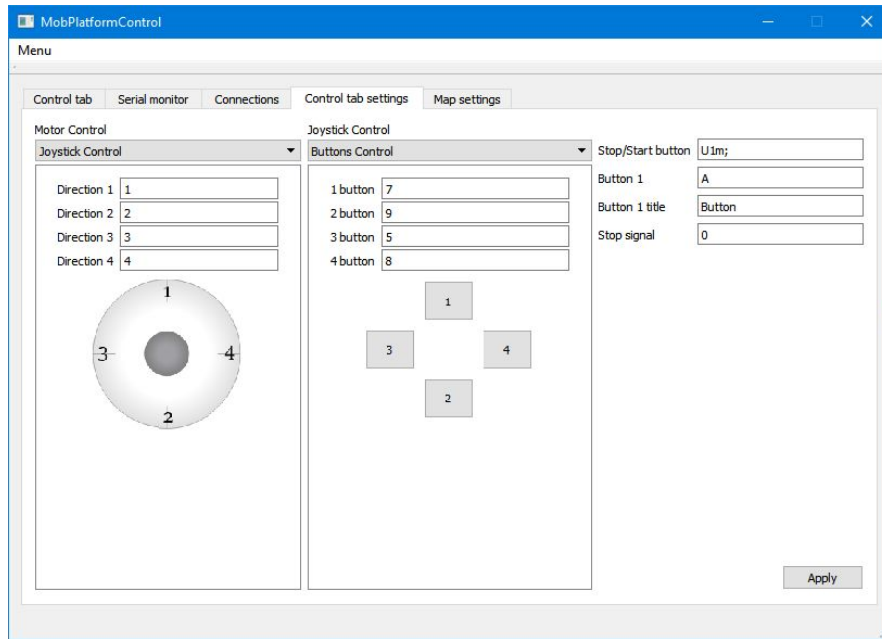
Jadaliidese avamine on keeruline protseduur. Esmalt kontrollitakse sidemooduli ühendust valitud COM-pordiga. Teiseks kontrollitakse, kas valitud COM-port on juba kasutusel mingis teises arvuti rakenduses. Jadaliidese avamise protseduuri algoritm on esitatud joonisel 12.



Joonis 12. Jadaliidese avamise algoritm.

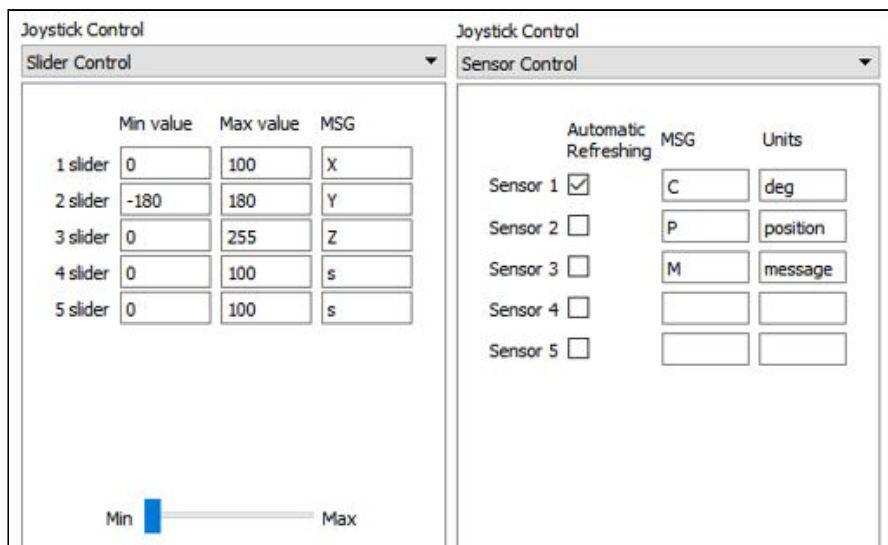


## 2.5.7 Tööriistade seadistamine



Joonis 13. Kasutajaliidese nuppude seadistamise vahekaart.

Kasutajaliidese üheks eeliseks on võimalus valida erinevaid tööriistu vastavalt projekti nõuetele. Kasutaja saab muuta iga kasutajaliidese elemendi (nuppude, liugurite ja andurite) signaali. Joonisel 13. on toodud juhtkangi ja nuppude seadistamise parameetrid. Joonisel 14. on toodud liugurite ja andurite seadistamise parameetrid. Tabelis 11. on toodud tööriistade seadistamise elementide kirjeldus.

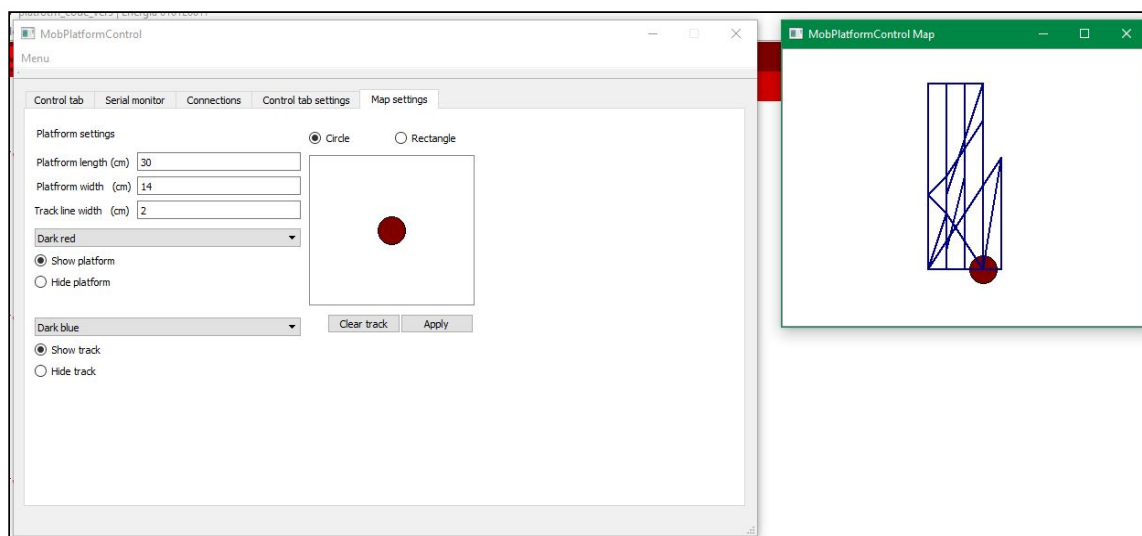


Joonis 14. Liugurite ja andurite parameetrite seadistamise aknad.

Tabel 11. Juhtimise tööriistade seadistamise vahekaardi elemendid

| <i>Elemendi nimetus</i>     | Elemendi toimingu kirjeldus   |
|-----------------------------|---|
| <i>Motor Control</i>        | Selles parameetrite rühmas saab valida juhtimise nupud platvormi liikumise jaoks. Valida saab nuppude ja juhtkangi vahel.   |
| <i>Joystick Control</i>     | Selles parameetrite rühmas saab valida juhtimise nupud platvormi kasuliku koormuse jaoks. Valida saab nuppude, juhtkangi, liugurite ning andurite kuvarite vahel.   |
| <i>Direction</i>            | Selle sisendisse saab määrata signaali, mida saadakse, kui juhtkang on vastavas piirkonnas. Signaal saab koosneda mitmest sümbolist.  |
| <i>button,</i>              | Selle sisendisse saab määrata signaali, mida saadakse nupule vajutamisel. Signaal saab sisaldada mitu märki.  |
| <i>Button 1</i>             | Selle sisendisse saab määrata signaali, mida saadakse nupule <i>Button 1</i> vajutamisel  |
| <i>Button 1 title</i>       | Selle sisendisse saab määrata nuppu <i>Button 1</i> nimetuse  |
| <i>Stop Signal</i>          | Selle sisendisse saab määrata signaali, mis peatab mobiilse platvormi tegevust.   |
| <i>Min value</i>            | Liuguri minimaalne väärtus nurgakraadides   |
| <i>Max value</i>            | Liuguri maksimaalne väärtus nurgakraadides  |
| <i>MSG (liugurid)</i>       | MSG tüüpi käsu kasutatakse informatsiooni edastamiseks, sisuliselt saadab kasutaja täiturite juhtimiseks käsu. MSG tüüpi käsuga määrab kasutaja, missugust täiturit ta soovib juhtida ning valitud täiturit hakatakse juhtima edaspidi vastava liuguri abil. Täpsemalt saadetise tüüpidest saab lugeda peatükis 2.4. Lisas 7. on toodud link video lõigule demo2, milles näidatakse kolme servomootori juhtimist. |
| <i>Automatic refreshing</i> | Antud valikut kasutatakse selleks, et võimaldada informatsiooni värskendamine automaatselt. Automaatne väärtuste tuvastamine toimub sekundis korra.   |
| <i>MSG (andurid)</i>        | MSG tüüpi käsku andurite puhul kasutatakse andurite info lugemiseks. Kasutaja saadab MSG käsuga informatsiooni, missuguse anduri infot ta näha soovib ning jadaliidese kaudu tagastatakse valitud anduri väärtus. Täpsemalt saadetise MSG tüüpidest saab lugeda peatükis 2.4.   |
| <i>Units</i>                | Anduri väärtustele saab lisada ühikuid  |

## 2.5.8 Kaardi seadistamine



Joonis 15. Platvormi liikumiste kaart.

Mobiilse platvormi liikumise ajalugu testimisplatsil joonistatakse üles ning kuvatakse ekraanil nii, nagu on esitatud joonise 15. paremal poolel. Antud ekraanil kuvatavat informatsiooni kasutatakse platvormi positsiooni määramiseks 2D kaardil. Kõik programmis avatud mobiilsed platvormid kuvatakse samal ekraanil punktidenähtuna, kus iga punkt näitab mobiilse platvormi asukohta. Asukohad testimisplatsil loetakse jadaliidese kaudu, platsist on räägitud peatükis 2.6.

Liikumise ajalugu kujutavad jooned salvestatakse massiivina esinemise järjekorras, iga element on üks joon. Kaardil on võimalik platvormi kuju muuta, saab kujutada nii ringina kui ka ristkülikuna, samuti saab muuta platvormi suurust, värvi ning jälje parameetreid nagu nähtavust, värvi, joone jämedust. Kaardi seadistamise vahekaart on nähtav joonisel 15. Sarnased funktsioonid on olemas ka traditsioonilises Logo programmis, millega on võimalik tutvuda lisas 6. Täpsemalt positioneerimisest testimisplatsil saab lugeda peatükis 2.5.1. Vahekaardi graafika elementide kirjeldus on toodud tabelis 12.

Tabel 12. Kaardi seadistamise vahekaardi elemendid

| Elemendi nimetus         | Elemendi toimingu kirjeldus  |
|--------------------------|--|
| <i>Platform setting</i>  | Antud alajaotise alt on võimalik muuta platvormi pikkust, laiust ning liikumisajalugu märkiva joone jämedust.            |
| <i>Ülemine rippmenüü</i> | Klõpsates rippmenüü järel oleva noole peale, tekib värvide nimekiri, mille alusel saab muuta platvormi kujutamise värvi. |
| <i>Clear track</i>       | Nupp kustutab platvormi liikumise ajaloo.  |

| Elemendi nimetus          | Elemendi toimingu kirjeldus   |
|---------------------------|---|
| <i>Show/Hide platform</i> | Antud käsu abil saab valida, kas platvorm on kaardilt nähtav või mitte. Punkt valiku Show platform ees tähendab, et platvormi tingmärki kuvatakse ekraanil.             |
| Alumine rippmenüü         | Klikkides rippmenüü järel oleva noole peale, tekib värvide nimekiri, mille alusel saab kasutaja valida endale meelepärase värvi liikumise ajaloo kuvamiseks.            |
| Show/Hide track           | Antud valikuga saab valida, kas kaardil kuvatakse platvormi liikumise ajalugu või mitte. Kui Show track ees on punkt, tähendab, et kaardil kuvatakse liikumise ajalugu. |
| <i>Circle/ Rectangle</i>  | Antud valikuga saab määrata, kas platvormi kuvatakse kaardil ringina või ristkülikuna.  |

## 2.6 Testimisplats

Testimisplats on keskkond, kus mobiilne platvorm hakkab sõitma. Eesmärgiks on luua selline testimisplats, et roboti juhtimiseks ei ole vajadust robotit realselt näha, kogu asukoha informatsioon kajastub arvutis olevast kasutajaliidesest. Testimisplatsi pilt on toodud joonisel 16.



Joonis 16. Testimisplats.

## 2.6.1 Navigatsioon ja positsiooni jälgimine

Mobiilses robotikas on positsioneerimine üks olulisemaid ja samas keerulisemaid valdkondi. Kaasaegses robotikas kasutatakse kombineeritud positsioneerimise meetodit, mis tähendab, et kasutatakse mitut erineva tööpõhimõttega seadet positsiooni määramiseks ruumis.

Platvormi positsiooni määramine toimub enkoodrite ja digitaalse kompassi abil. Enkooder ühendatakse mobiilse testplatvormi roomikute külge, mootori pöörete pealt enkoodri väärtused muutuvad. Igaale nivoo ümberlülitusele vastab teatud kaugus. Enkoodrite väärtuste muutused saadetakse arvutile, kus teisendatakse väärtuse muutus läbitud vahemaaks. Läbitud vahemaa põhjal liigutatakse mobiilset platvormi kasutajaliidese kaardil, trajektoori pikkus on võrdelises seoses kaardile joonistatava joonega. Liikumise suund määratakse digitaalse kompassi abil.

Nagu eelpool mainitud, et vahemaa arvutamiseks on valem, siis ainult enkoodrite väärtusest ei piisa vahemaa arvutamiseks. Probleemiks on võimaliku vea akumulatsioon aja jooksul, sisuliselt mida pikem vahemaa läbitakse, seda suurem on viga reaalse ja arvutatud asukoha vahel. Positsioneerimise täpsuse tõstmiseks kombineeritakse mitu positsioneerimise meetodit. Lisaks enkoodritele kasutatakse positsiooni määramiseks passiivseid RFID-kaarte, mille kiibile on kirjutatud koordinaat, mille järgi saab määrata täpse roboti XY positsiooni ruumis. RFID-kaardi eelised: madal hind, paindlikkus, töökindlus ja lihtne tööpõhimõte [3]. Vajaduse korral robotika arendaja saab kasutada RFID-kaarte ka teistes projektides. Kuna neil on lai kasutusala.

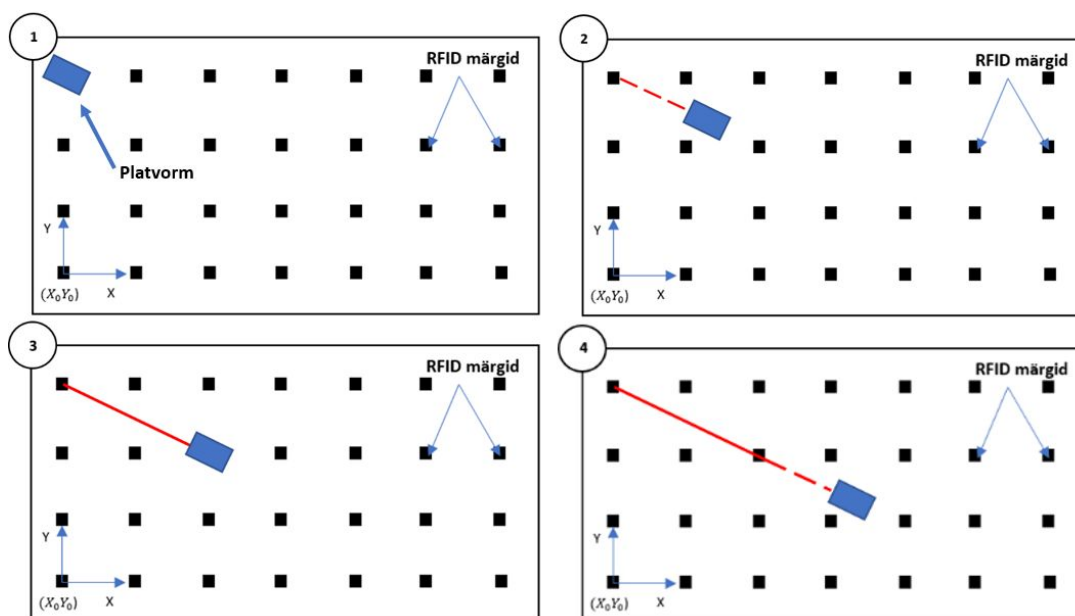
Valmistatud testimisplatsi suurus on 3.0x1.2 m. Ühe RFID kaardi suurus on 0.08x0.05 m. Kaugus kahe RFID-kaardi vahel on 0.2 m. Mida rohkem RFID-kaarte asub testimisplatsil, seda täpsem on tulemus. Kaartide paigutamiseks jaotatakse testimisplats võrdsete pindaladega ruutudeks, mille keskele pannakse RFID-märk

Kasutades ainult RFID-märke, ei saa roboti positsiooni märkide vahel määrata ja tekib olukord, kus robot saab liikuda ainult ühe märgi pealt teise märgi peale. RFID-märkide tuvastamiseks on need tähistatud ja sisaldavad informatsiooni oma asukoha kohta. Kuna RFID-märkidega koordinaatsüsteemi ei saa rakendada iseseisvalt, siis platvormi positsiooni määramiseks kasutatakse suurema täpsuse tagamiseks nii RFID-märke kui ka enkoodereid, mis võimaldab saata roboti võimalikult täpselt suvalisele koordinaadile [3]. Joonisel 17 on esitatud 4 pilti platvormi liikumisesttestimisplatsil, millega selgitatakse, kuidas määratakse roboti asukoht kaardil.

1. Esimesel pildil platvorm alustab liikumist. Platvormi liikumise suund määratakse digitaalse kompassiga.
2. Teisel pildil platvorm on liikunud valitud suunas. Sel etapil ei ole tee peale sattunud RFID-märke ja kaugus määratakse enkoodritelt saadud informatsiooni põhjal. Kuna

pikkade vahemaade puhul ei ole mõttekas iga enkoodri väärtust kuvada, siis uuendamine toimub pärast teatud enkoodri väärtuste muutust. Kasutajaliidese kaardil kuvatakse ainult enkoodri abil saadud väärtused punktiirjoonega, sest näidatakse roboti arvutuslikku positsiooni, mis ei pruugi olla täpselt sama nagu roboti tegelik positsioon ruumis.

3. Kolmandal pildil platvorm läbib RFID-kaarti. Kuna platvormi alla on paigutatud RFID-kaardi lugeja ja RFID-kaardile on kirjutatud kaardi positsioon, siis selles etapis määratakse platvormi tegelik positsioon. Eelpool tekitatud katkendjoon muutub pidevaks, sest platvormi asukoht on kinnitatud.
4. Neljandal pildil on näha, et roboti RFID-kaardist edasi liikumisel läbitud teekond on jälle esitatud punktiirjoonega, sest see on arvutuslik.



Joonis 17. Tagasiside mobiilse platvormi positsioonist.

Platvormi positsiooni määramine toimub kahel viisil:

- Absoluutne kaugus. Kaugus saadakse RFID-kaartide abil, kuna kaardid on kinnitatud kindlatele asukohtadele ja sisaldavad koordinaadi informatsiooni, siis saadakse roboti täpne asukoht nullpunkti suhtes. Saadetise *MSG* tüübiks on 'P'.
- Inkrementaalne kaugus. Kaugus saadakse enkoodrite põhjal, see on kaugus mis arvutatakse viimasest kindlalt läbitud punktist. Saadetise *MSG* tüübiks on 'F' - ettepoole või 'B' - tahapoole.

Nurga informatsioon saadakse digitaalse kompassi abil. Nurga informatsioon on vajalik selleks, et määrata platvormi liikumise suund sihtpunkti suunas. Nurka informatsioon saadakse *MSG* tüübiga 'C'. Platvormi andmevahetusest on detailsemalt kirjeldatud peatükis 2.4.

## 2.7 Testrakenduse katsetamine

Peatükis teostatakse katsed eelmistes peatükkides kirjeldatud testsüsteemiga ning analüüsitakse katsete tulemusi. Lõpptulemusena antakse hinnang loodud süsteemile ning tuuakse välja peamised arendamist vajavad kohad.

Mobiilse platvormi elektroonika liikumise juhtimise osa on valmistatud trükkplaadina, sest trükkplaat muudab kontaktide ühenduse kindlaks ning lõpptulemuse kompaktsemaks. Trükkplaat lubab kasutada kaht tüüpi mootoreid mobiilse platvormi liikumise tagamiseks: DC mootoreid ja samm-mootoreid. Trükkplaadi külge ühendati mõlemad mootorid ning nad töötasid antud trükkplaadiga korralikult. DC mootori kasutamisel mootori pöörete arvu saab loendada enkoodriga. Samm-mootori puhul enkoodrid pole vaja kasutada. Samm-mootorina kasutati Nema 17, kuid mobiilsele platvormile seda pole pandud. Trükkplaadi komponente õnnestus ühendada omavahel ning need töötasid sujuvalt.

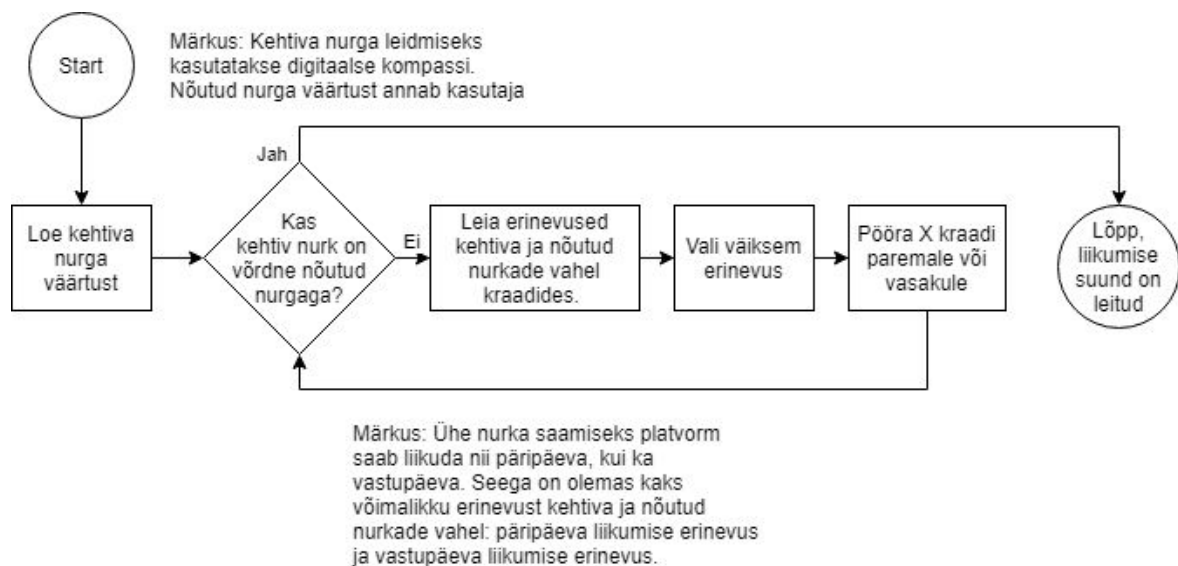
Mobiilse platvormi juhtimiseks valmistati kasutajaliides, mis lubab saata mobiilsele platvormile signaalid ning võtta neid vastu. Jadaühendus arvuti külge ühendatud antenniga töötab tõrgeteta. Paralleelselt on võimalik juhtida kuni 4 mobiilset platvormi. Paralleelse töö katsetamiseks valmistati kaks mobiilset testplatvormi. Käsureast sisetatud programmide saatmine platvormidele ja nende täitmine toimus üheaegselt. Kasutajaliidese liugurite töö katsetamiseks ühendati 3 servomootorit ning katsetati nende juhitavust. Servomootorid pöörlesid vastavalt liuguri asendile. Projekti edasi arendamisel tuleb lisada võimalust määrata nimetused kõikidele juhtimisnuppudele ja valida 2d kaardi taustaks pilt.

Testimisplatsile paigaldati RFID-kaardid, millele sisestati asukohale vastav positsioon testimisplatsil. Mobiilne platvorm sõitis testimisplatsil ning loetud RFID-kaardi väärtus saadeti arvutile, kuid selgus, et suurel kiirusel RFID-lugejal ei õnnestunud kaardi väärtust lugeda. Eriti probleemne oli pikkade numbritega koordinaadide edastamine, sest jadaliides ei suutnud neid ühe korraga lugeda ning edastada. Seda probleemi õnnestus tarkvaraliselt lahendada. Lisaks viidi RFID-lugeja madalamale, platsile pinnale lähemale. Nende lahenduste tulemusena muutus RFID-kaartide lugemine töökindlamaks.

Mobiilse platvormi roomikute külge ühendati enkoodrid, mida kasutatakse sammude lugemiseks. Käsurealt saab liigutada mobiilset platvormi määratud arv sammu. Üheks sammuks on enkoodri nivoo ümberlülitus. Mikrokontrolleril on oma loendur iga enkoodri jaoks. See loeb sammude arvu, mida üks ratas on teinud. Katsetamise tulemusena selgus, et erinevus kasutaja poolt määratud ja tehtud sammude vahel on kuni 4 sammu, kusjuures reaalsete sammude arv on suurem, sest platvorm ei peatu koheselt. Peale käsu täitmist mobiilne platvorm saadab kinnitust, et käsk on

täidetud arvutile. Sammude erinevus lineaarsel liikumisel pole probleem, see on lahendatav mikrokontrolleri tarkvara arendamisega. Probleemiks osutuvad eksisammud mobiilse platvormi pööramise juhul, sest eksimused põhjustavad liikumise suuna erinevusi võrreldes eeldatava suunaga. Lisaks tuleb lisada võimalus saata käsud kauguse ühikutes, mitte sammu ühikutes.

Mobiilsele platvormile saab määrata liikumise suunavektori põhja suuna suhtes. Sellist funktsiooni on vaja autonoomseks liikumiseks ühest punktist teise. Hetkel selline funktsioon puudub. Suunavektori leidmise algoritmi plokkskeem on esitatud joonisel 18. Selle algoritm on realiseeritud mobiilse testplatvormi mikrokontrolleri programmis. LSM303 digitaalse kompassi väärtuste lugemiseks on kasutatud teise autori koodi lõiku [10]. Kompassi tuli kalibreerida, et määrata õige põhja suund. Lõpptulemusena oli digitaalse kompassi ja magnetkompassi põhja suundade erinevus kuni 5°. Omavahel võrreldi 3 digitaalset kompassi. Nende väärtused erinesid kaks kuni kolm kraadi. Algoritm töötab korralikult, kuid digitaalne kompass annab valed väärtused mobiilse platvormi liikumise ajal, tõenäoliselt on põhjus platvormi pinnal olevates vooluga juhtmetes, milles tekib Ampere'i jõud. Lisas 7. on toodud link video lõigule demo3, milles mobiilne platvorm püüab saada vastavat nurka, antud algoritmi alusel. Lisas 10. on toodud tabel katsetustega. Katsete keskmine vahe kasutaja poolt määratud nurga ja saadud nurga vahel on 4.33°. Erinevuste standardhälve on 5.55°. Seda ei saa nimetatada täpseks, kuid üldine liikumissuund on leitud õigesti.



Joonis 18. Nurga leidmise algoritm.



## KOKKUVÕTE

Bakalaureusetöona sai valmis testrakendus mobiilse platvormi juhtimiseks. Testrakendus loodi robotika õpetamiseks ning robotika harrastajate projektide tarbeks. Testrakendus on näidislahendus süsteemist, mille osad robotika harrastajad saavad võtta aluseks enda projektides. Mobiilne platvorm on kohandatav vastavalt kasutaja projektile, sellele saab lisada erinevaid täitureid nagu: kaamera, andurid, haaratsid jne. Lõputöö kirjeldab projekti esimest etappi ehk näidissüsteemi väljatöötamist ja selle osade ühendamist.

Antud töö on jagatud osadeks, kus esmalt vaadeldi erinevaid lahendusi, siis kirjeldati testrakenduse loomise protsessi ja katsetamise protsessi. Kõigepealt analüüsiti mobiilsete platvormide juhtimiseks kasutatavaid erineva tasemega programme ning tutvuti enimkasutatava MAVLink sideprotokolliga . Analüüsi tulemusena võeti kasutajaliidese loomise eeskujuks programm BT Robot Controller v1.2.

Töö teises osas on kirjeldatud testrakenduse süsteemi loomist. Süsteem koosneb järgmistest moodulitest: mobiilne testplatvorm, kasutajaliides ja testimisplats. Infovahetuse sideprotokollina kasutatakse lihtsamat ja loetavat protokollit MAVLink protokollit struktuuri võttes. MAVLink protokollit puuduseks on halb loetavus ning keerulisus.

Mobiilne testplatvorm on roomikutel liikuv robot. Mobiilse platvormi elektroonika koosneb kahest osast, mõlemal oma mikrokontroller, ühega juhitakse liikumist, teisega andureid ja täitureid. Mikrokontrolleritel on programmid, mille tööpõhimõtte on kirjeldatud plokk skeemidel. Elektroonika liikumise juhtimise osa on valmistatud trükkplaadina, sest trükkplaat muudab kontaktide ühenduse kindlaks ning lõpptulemuse kompaktsemaks. Elektroonika täiturite ja andurite osa on lahendatud tavaliste ühendustega, sest täiturite ja andurite kasutamine sõltub konkreetsest rakendusest ja peab olema kasutajal kohandatav vastavalt roboti rakendamisalale.

Kasutajaliideselt on võimalik platvormi juhtida kahel viisil: nuppudega või käsureast. Mobiilsele platvormile paigaldatud täiturite ja andurite juhtimiseks on võimalik valida neli tööriista: nuppude kombinatsioon, juhtkang, liugurid ja andurite kuvarid. Kasutajal on võimalik nupuvajutuse toiminguid kohandada. Käskudega juhtimise loogika aluseks on Logo programmeerimiskeel ning sisestatud käskude täitmine on reaajas jälgitav olekuribalt. Käsurealt saab juhtida nii mobiilse platvormi liikumist kui ka selle peale paigaldatud täitureid. Käske saab salvestada kasutajaliidese mälus. Kasutajaliides võimaldab juhtida neli mobiilset platvormi korraga. Mobiilse platvormi positsiooni nullpunkti suhtes on võimalik kuvada eraldi ekraanil. See on sisendiks platvormi kuvamiseks 2d kaardil. Kõik platvormid kuvatakse ühel ekraanil.

Testimisplats on paberileht, mille pindala on ühtlaselt kaetud RFID-märkidega. RFID-märgi kiibi peal on kirjas selle asukoht testimisplatsi nullpunkti suhtes. Mobiilse testplatvormi põhjal asub RFID-märkide lugeja. Mobiilse platvormi liikumisel RFID-kaardi kohale, loetakse selle kiibi peal olevat asukoha väärtust ning saadakse kasutajaliidesesse.

Lõputöö eesmärk on suures osas täidetud, sest suur osa süsteemi tarkvarast, mikrokontrollerite püsivarast, elektroonikast ja algoritmidest on välja arendatud ja valmistatud. Süsteemi moodulid ühendati omavahel ning katsetati nende tööd. Töö autor kavatseb jätkata süsteemi arendamist, edaspidi on plaanis testida kasutajaliidest, tagada selle töökindlus ning anda ligipääs teistele kasutajatele. Kasutajaliidese kaardile tuleb lisada võimalus panna pilti taustaks. Lisaks tuleb parandada digitaalse kompassi lugemist ja platvormi positsioneerimist ruumis.

## SUMMARY

The aim of this thesis is to create a test system for controlling a moving platform. This test system is created with the purpose of robotics learning and for robotics enthusiasts. The test system is designed to be an example of a system that can be used as a starting point for their projects. Mobile platform can be changed in relation to the user projects. It can be modified with a wide selection of tools. Such as: camera, various sensors and grippers. This work describes the first stage of the project, which is to develop an example system and its attachments.

This thesis is divided into sections. First part is a review of previous studies in this field. The following section describes the creation of the prototype system and the process of testing. The review was conducted in the field of moving robots control systems and “MAVLink” communication protocol – software that is used to control drones. As a result, a “BT Robot Controller v1.2” was chosen as a user interface example.

The creation of the software prototype is presented in the second part of this thesis. The system consists of the following modules: a moving platform, graphical interface and test environment. MAVLing is used as a good example of the communication protocol. Though, it required a few upgrades for a simpler and more legible functioning. The raw MAVLink is too complicated and difficult for new users.

The moving platform represents a robot on tracks. Its inner electronics consist of two part. Each part has a separate microcontroller. One is used to control the movement of the robot, and the second one is used to control the sensors and actuators. Each microcontroller has its own encoding. These are presented and described in the block diagrams. First electronics section required printing of a circuit board. Basic components that control the movements of the platform were mounted on this circuit. The part that is responsible for the moving platform control was mounted on this printed circuit board, because this type of circuit allows a more tight placement of the pin connections and makes the end product more reliable. The second part that is responsible for actuators and sensors was connected to the microcontroller using usual connection. There was no need for another printed board circuit in the second part because it is assumed to be very project specific and should be produced by the users themselves.

The graphical interface allows two methods of controlling the platform: simple press-buttons design and a command line. User can choose four different tools to control the actuators and sensors: button combination, joystick, sliders or sensor displays. It is possible for the user to choose what signal the buttons send when pressed. The logic of command line programming is

based on “Logo” programming language. The performance of the imputed command can be monitored in status bar in real time.

Using the command line, it is possible to control the movement of the moving platform and its mounts. Imputes can be saved in the memory of the user interface. The graphical interface allows to control up to four moving platforms at the same time. It is possible to track the platforms position relative to the zero point in a separate window. The positions of all platforms are displayed on one screen using 2D map.

The test environment is a sheet of paper, which is covered with RFID-tags. RFID-tag chip contains information regarding the RFID-tag location on the paper. RFID-tag reader is located under the moving platform. When the platform passes above the RFID-card its chip position information is read and sent to the computer. Overall, it is believed that the work has achieved most of initially set goals. System software, encoding of the microcontrollers, electronics and algorithms, which are the major parts of the project were developed and successfully produced. System modules were connected to each other and tested.

The author is planning to continue developing the project. Future project plans include testing the user interface for bugs and providing access to other users. As one of the possible improvements, a possibility to add a background to the user graphical interface map could be included. Additionally, digital compass reading and accuracy of platform positioning in the room could be improved.

## KASUTATUD KIRJANDUS

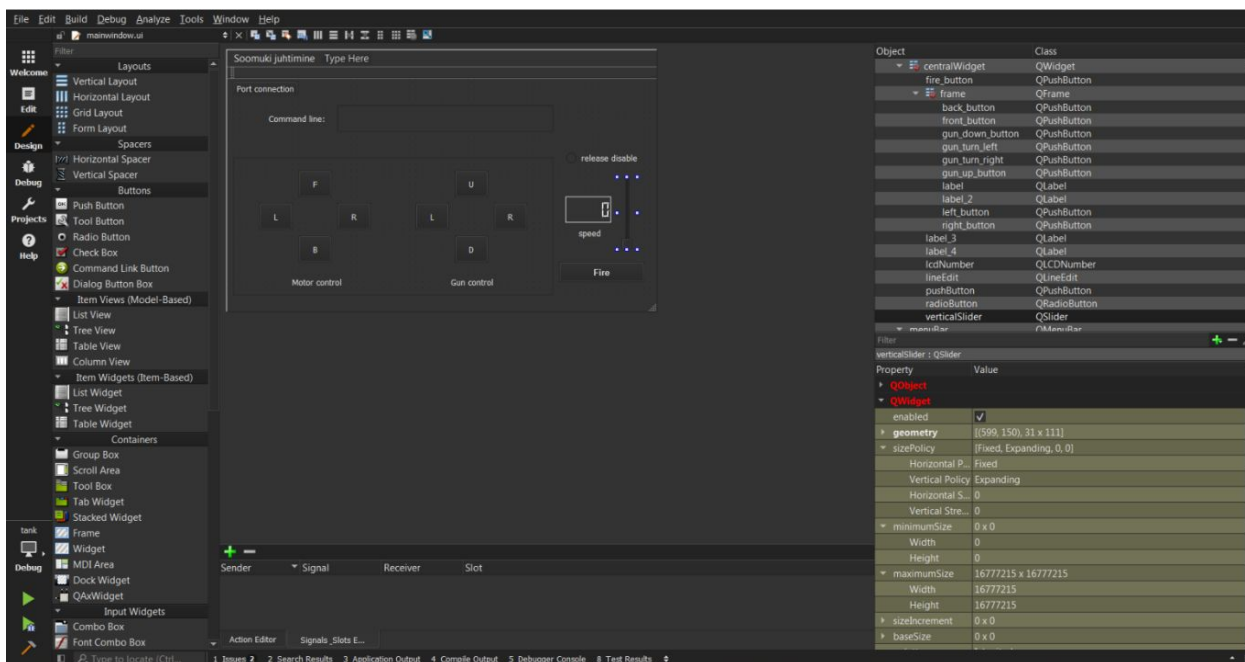
- [1] Joypad-Widget [võrgumaterjal] Available: <https://github.com/tobiasThomer/Joypad-Widget> [Kasutatud 10.03.2019].
- [2] Description, Digital Pins With Interrupts [võrgumaterjal] Available: <https://energia.nu/reference/en/language/functions/external-interrupts/attachinterrupt/> [Kasutatud 20.03.2019].
- [3] INTAN SURIA BINTI ZAKARIA, RFID-BASED INDOOR POSITIONING OF AUTONOMOUS AID FOR DISABLE PEOPLE, 2013 [võrgumaterjal] Available: [http://eprints.uthm.edu.my/3648/1/INTAN\\_SURIA\\_BINTI\\_ZAKARIA.pdf](http://eprints.uthm.edu.my/3648/1/INTAN_SURIA_BINTI_ZAKARIA.pdf) [Kasutatud 20.03.2019].
- [4] What is RFID? [võrgumaterjal] Available: <https://web.archive.org/web/20120322194318/http://www.explania.com/en/channels/technology/detail/what-is-rfid/> [Kasutatud 20.03.2019].
- [5] Open roboti projekti veebileht [võrgumaterjal] Available: <https://code.google.com/archive/p/open-robot-zigbee-gui/> [Kasutatud 30.03.2019].
- [6] Open roboti kasutajaliidese kirjeldus [võrgumaterjal] Available: [https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/open-robot-zigbee-gui/OpenRobot\\_GUI\\_Manual.pdf](https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/open-robot-zigbee-gui/OpenRobot_GUI_Manual.pdf) [Kasutatud 30.03.2019].
- [7] Zigbee+Solar+Robot kasutajaliidese pilt [võrgumaterjal] Available: [http://3.bp.blogspot.com/\\_fTdr602nenQ/R0pA-b2t1JI/AAAAAAAAANU/sJTzf-1rfR4/s1600/Robot+GUI\\_Oct07.jpg](http://3.bp.blogspot.com/_fTdr602nenQ/R0pA-b2t1JI/AAAAAAAAANU/sJTzf-1rfR4/s1600/Robot+GUI_Oct07.jpg) [Kasutatud 30.03.2019].
- [8] What is GPIO? [võrgumaterjal] Available: <https://community.estimote.com/hc/en-us/articles/217429867-What-is-GPIO-> [Kasutatud 01.04.2019].
- [9] mswlogo programmi ekraanipilt [võrgumaterjal] Available: [https://images.sftcdn.net/images/t\\_app-cover-l\\_f\\_auto/p/867a5048-9b24-11e6-8122-00163ec9f5fa/959072388/mswlogo-screenshot.png](https://images.sftcdn.net/images/t_app-cover-l_f_auto/p/867a5048-9b24-11e6-8122-00163ec9f5fa/959072388/mswlogo-screenshot.png) [Kasutatud 01.04.2019].
- [10] lsm303 digitaalse kompassi väärtuste lugemise arduino kood [võrgumaterjal] Available: <https://github.com/pololu/lsm303-arduino/blob/master/examples/Heading/Heading.ino> [Kasutatud 05.04.2019].

- [11] LOGO Software and Hardware [võrgumaterjal] Available:  
[https://el.media.mit.edu/logo-foundation/resources/software\\_hardware.html](https://el.media.mit.edu/logo-foundation/resources/software_hardware.html)  
[Kasutatud 12.04.2019]
- [12] LOGO programming [võrgumaterjal] Available:  
[https://el.media.mit.edu/logo-foundation/what\\_is\\_logo/logo\\_programming.html](https://el.media.mit.edu/logo-foundation/what_is_logo/logo_programming.html)  
[Kasutatud 12.04.2019]
- [13] LOGO programmeerimise keele baaskäsud [võrgumaterjal] Available:  
<https://glinsklogoprogramming.weebly.com/logo-commands.html> [Kasutatud 12.04.2019]
- [14] LOGO Commands 1 [võrgumaterjal] Available:  
<http://www.mit.edu/~h1b/MA562/commands.html> [Kasutatud 12.04.2019]
- [15] Denis Kiltitsakov, Andmevahetuse sideprokollid, 2012 [võrgumaterjal] Available:  
<https://habr.com/ru/post/138533/> [Kasutatud 13.04.2019]
- [16] Mavlink Basics [võrgumaterjal] Available: <http://ardupilot.org/dev/docs/mavlink-basics.html>  
[Kasutatud 13.04.2019]
- [17] MAVLINK Common message set [võrgumaterjal] Available:  
<https://mavlink.io/en/messages/common.html> [Kasutatud 13.04.2019]
- [18] ArduPilot, 2019 [võrgumaterjal] Available: <http://ardupilot.org/ardupilot/index.html>  
[Kasutatud 14.05.2019]
- [19] Zigbee + solar + robot, 2010 [võrgumaterjal] Available:  
<http://intillegencerobots.blogspot.com/2010/12/zigbee-solar-robot.html>  
[Kasutatud 14.05.2019]
- [20] Hairi Azdy Bin Aezman, Mobile Robot Simulation and Controller Design with Matlab Simulink, 2005 [võrgumaterjal] Available:  
[https://www.academia.edu/6834350/Mobile\\_Robot\\_Simulation\\_and\\_Controller\\_Design\\_with\\_Matlab\\_Simulink](https://www.academia.edu/6834350/Mobile_Robot_Simulation_and_Controller_Design_with_Matlab_Simulink) [Kasutatud 14.04.2019]
- [21] Getting start with the mobile robotics simulation toolbox [võrgumaterjal] Available:  
<https://se.mathworks.com/videos/matlab-and-simulink-robotics-arena-getting-started-with-the-mobile-robotics-simulation-toolbox-1523443253912.html> [Kasutatud 14.04.2019]
- [22] Video: Mobile Robot Simulation for Collision Avoidance with Simulink [võrgumaterjal]  
Available: <https://www.youtube.com/watch?v=s350OktFuaw> [Kasutatud 14.04.2019]

- [23] SiK Telemetry Radio [võrgumaterjal] Available:  
<http://ardupilot.org/rover/docs/common-sik-telemetry-radio.html#common-sik-telemetry-radio>  
[Kasutatud 15.04.2019]
- [24] Terminal Example [võrgumaterjal] Available:  
<https://doc.qt.io/qt-5/qtserialport-terminal-example.html> [Kasutatud 16.04.2019]
- [25] BT Robot Controller v1.2 project post [võrgumaterjal] Available:  
<https://blazinge.wordpress.com/2013/04/19/bt-robot-controller-software/>  
[Kasutatud 29.04.2019]
- [26] ASCII (American Standard Code for Information Interchange) [võrgumaterjal] Available:  
<https://whatis.techtarget.com/definition/ASCII-American-Standard-Code-for-Information-Interchange> [Kasutatud 30.04.2019]
- [27] PULSILAIUSMODULATSIOON (PWM) ja "Analoog väljund" [võrgumaterjal] Available:  
<http://libahunt.ee/et/eka-arduino-algajatele-2016S-PWM> [Kasutatud 14.05.2019]
- [28] USART - järjestikühenduse liides [võrgumaterjal] Available:  
<https://www.robotiklubi.ee/juhendid/avr-cpp-lib/usart> [Kasutatud 14.05.2019]
- [29] What is ROS?, 2018 [võrgumaterjal] Available:  
<http://wiki.ros.org/ROS/Introduction> [Kasutatud 14.05.2019]
- [30] Yahya Tawil, An Introduction to Robot Operating System (ROS), 2017 [võrgumaterjal] Available:  
<https://www.allaboutcircuits.com/technical-articles/an-introduction-to-robot-operating-system-ros/> [Kasutatud 14.05.2019]
- [31] Moveit, Frequently Asked Questions [võrgumaterjal] Available:  
<https://moveit.ros.org/documentation/faqs/> [Kasutatud 14.05.2019]
- [32] Why Gazebo? [võrgumaterjal] Available: <http://gazebosim.org/#getstarted>  
[Kasutatud 14.05.2019]
- [33] Pixhawk overview [võrgumaterjal] Available:  
<http://ardupilot.org/rover/docs/common-pixhawk-overview.html#common-pixhawk-overview>  
[Kasutatud 14.05.2019]
- [34] Choosing a Ground Station [võrgumaterjal] Available:  
<http://ardupilot.org/rover/docs/common-choosing-a-ground-station.html>  
[Kasutatud 14.05.2019]
- [35] MSP430 Launchpad [võrgumaterjal] Available:  
<https://components101.com/microcontrollers/msp430-launchpad-pinout-datasheet>  
[Kasutatud 18.05.2019]

# LISAD

## Lisa 1 - Qt Creatori Graafilise disaini tööriist

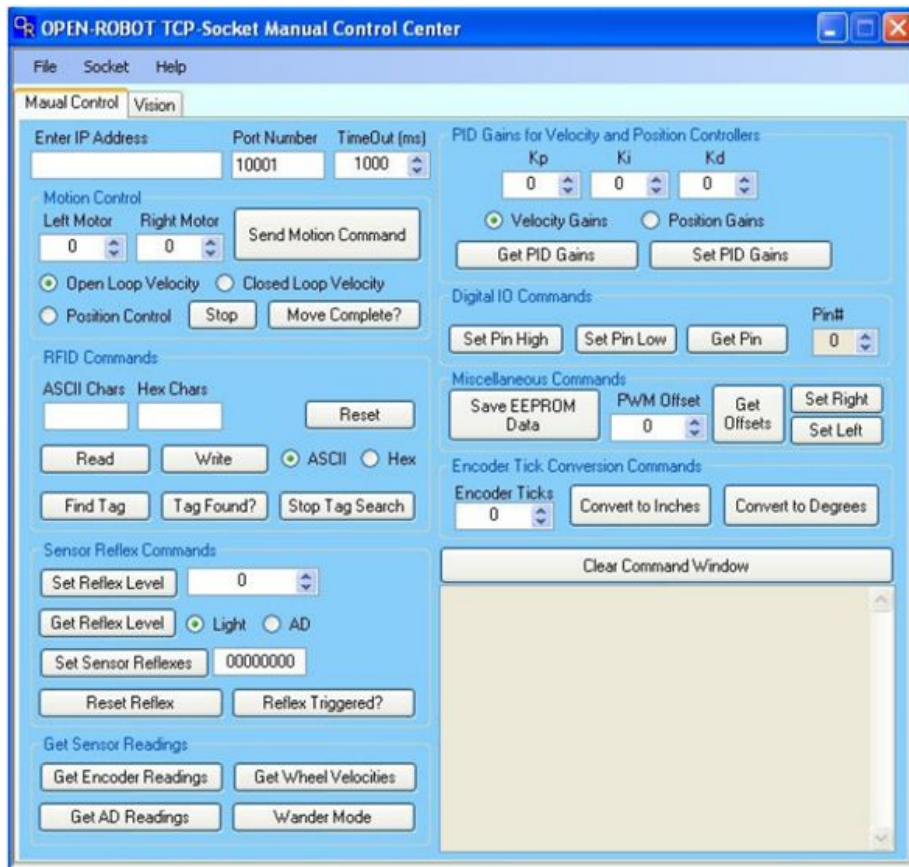


Joonis 19. Qt Creator graafilise disaini tööriist.

Qt Creator tarkvaraarendus platvorm on levinud automaatika alal tööstusrakenduste valmistamisel. Näiteks kasutatakse Qt Creator LG televiisorite, Mercedes-Benz autoarvuti ekraani valmistamisel. Programmeerimise keelena siin kasutatakse C++ ja Python keelt. Qt Creator programmis saab kasutada graafilise disaini tööriista. Joonisel 19. on esitatud Qt Creator graafilise disaini tööriist. Selle tööriistaga on mugavam lisada nuppe, liugureid, olekuribasid, käsu sisendeid ja teisi graafika elemente. Ekraani vasakul poolel asuvad graafika elemendid, mida saab kasutada. Ekraani keskmisel osal asub arendatav kasutajaliides. Ekraani paremal ülemisel poolel asub kasutatud elementide nimekiri. Ekraani paremal alumisel poolel saab konfigurierida valitud elemendi omadusi. Elemendi omadused sõltuvad selle rakendusest. Näiteks nuppu omaduseks on suurus, positsioon ekraanil, stiil, nimetus, tegevus, mis toimub vajutamisel, jne.

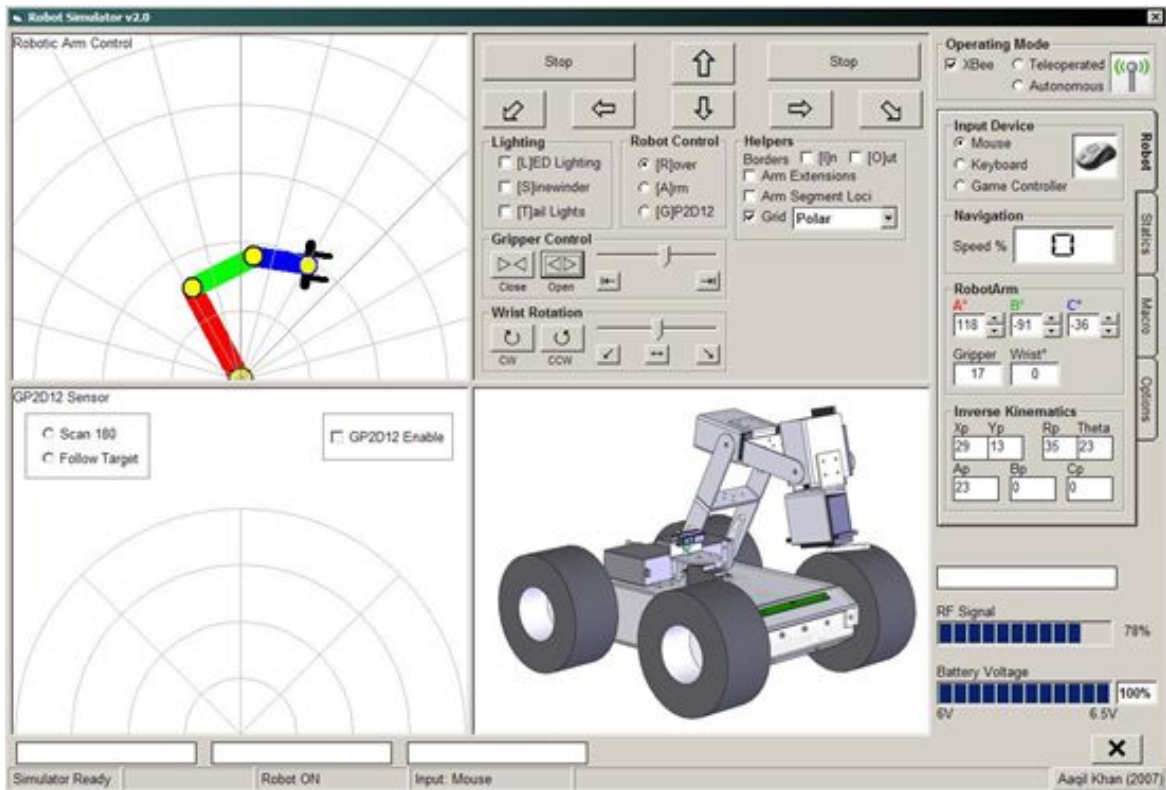


## Lisa 2 - Open Robot kasutajaliidese ekraanipilt



Joonis 20. Open Robot kasutajaliidese ekraanipilt [6].

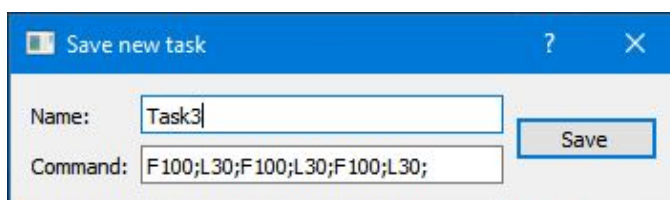
### Lisa 3 - Zigbee+Solar+Robot projekti kasutajaliidese ekraanipilt



Joonis 21. Zigbee+Solar+Robot projekti kasutajaliidese ekraanipilt [7].

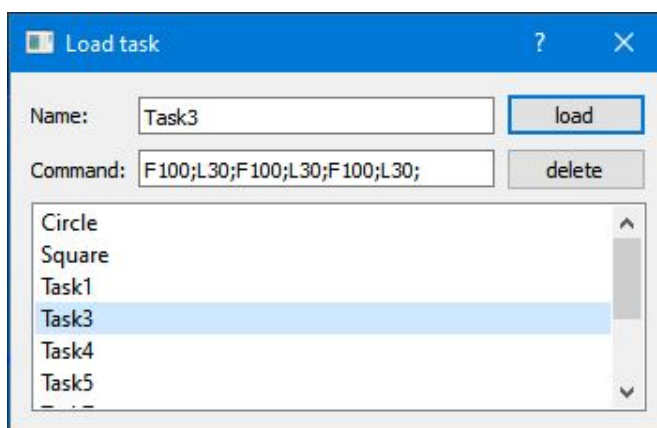
## Lisa 4 - Käskude salvestamine ja laadimine

Kõik kirjutatud käsud salvestatakse faili nimega tasks.txt ning need saab laadida mikrokontrollerile, käsu laadimisel ja salvestamisel vastavust süntaksile ei kontrollita, kuid reaalselt täita saab ainult käskusid, mis vastavad programmeerimiskeele süntaksile. Ühe platvormi jaoks kirjutatud käskusid saab kasutada juhtimaks ka teisi platvorme, igale platvormile tuleb juhtimise aknast saata valitud käsk. Joonisel 22. ja 23. on esitatud käsu laadimise ja salvestamise aknad.



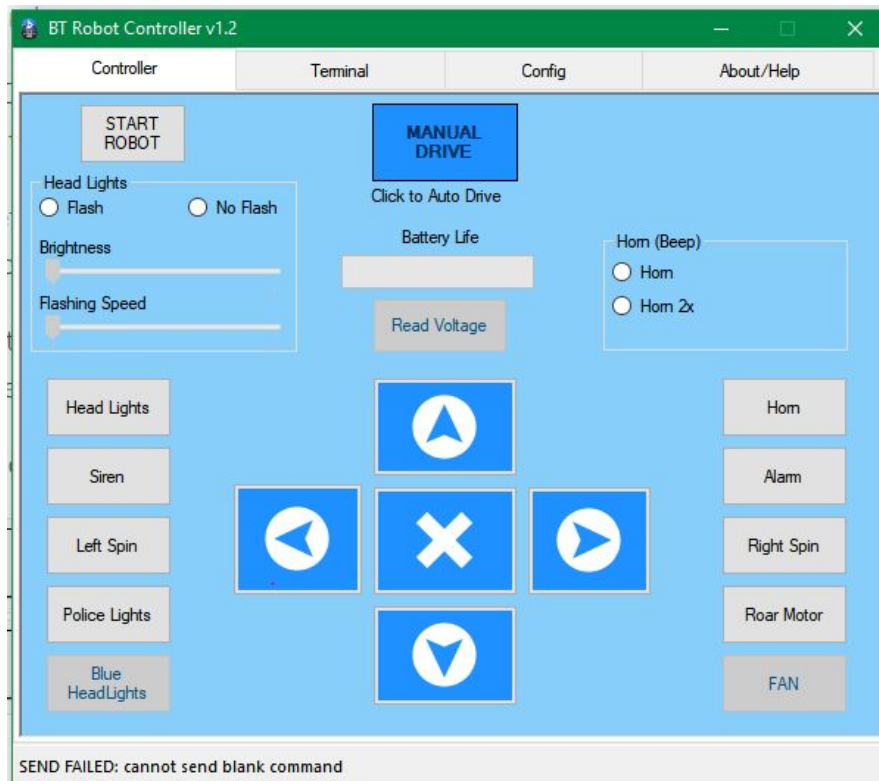
Joonis 22. Käsu salvestamise aken.

Käsu laadimise aknas saab laadida salvestatud käske ning kustutada neid.



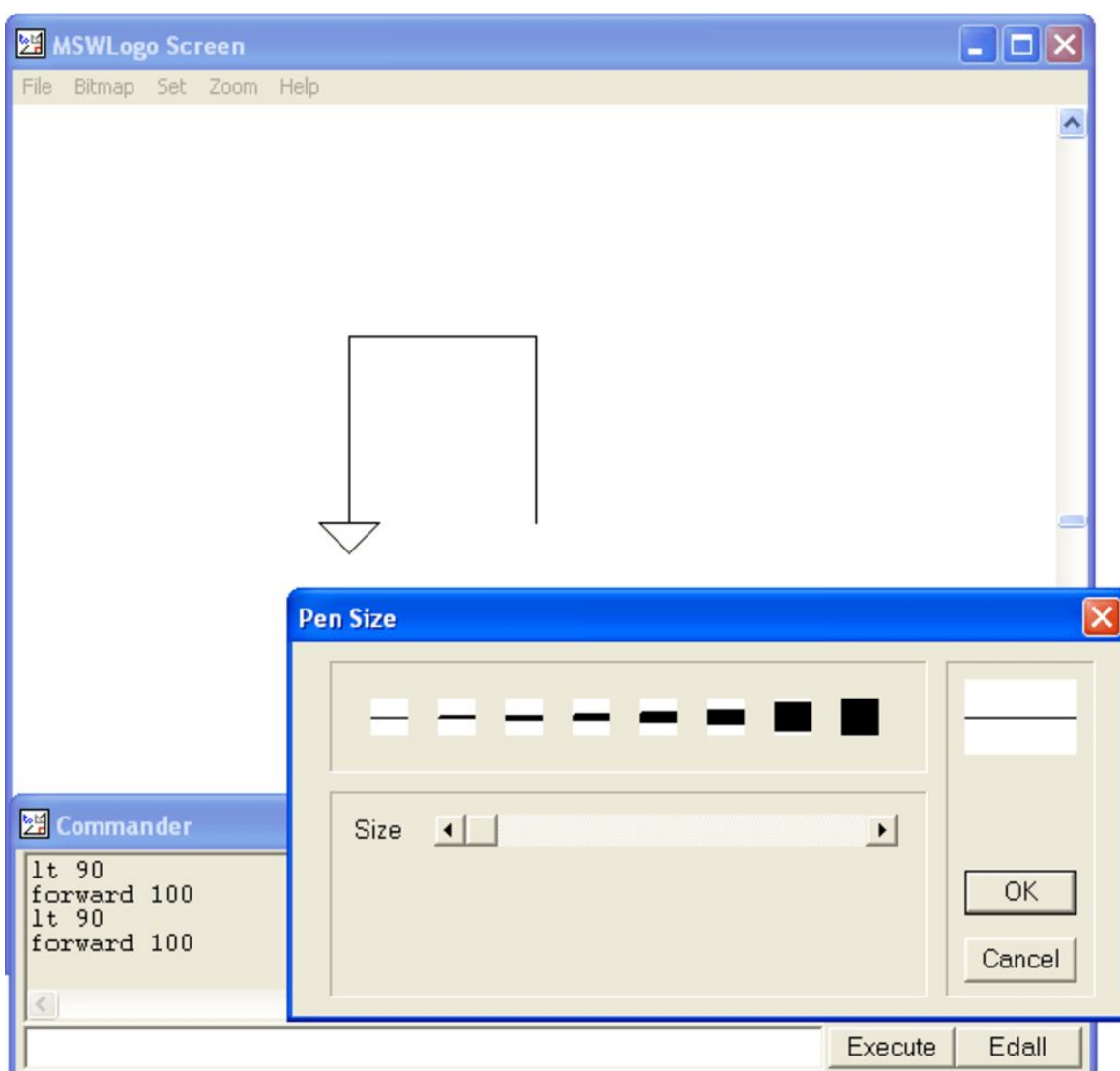
Joonis 23. Käsu laadimise aken.

## Lisa 5 - BT Robot Controller v1.2 kasutajaliidese ekraanipilt



Joonis 24. Bt Robot Controller kasutajaliidese ekraanipilt [25].

## Lisa 6 - Logo kasutajaliidese ekraanipilt ja baaskäsud



Joonis 25. Logo programmi kasutajaliidese ekraanipilt [9].

*Commander* aknas saab kirjutada koodi, mida kilpkonn hakkab täitma. Kaardil saab näha tulemust. Käsudeks on:

- *lt 90* - pööra vasakule 90°
- *forward 100* - mine 100 sammu ettepoole

Vajutades *Execute* nupule saab käivitada kirjutatud koodi. Logo programmeerimiskeele baaskäsud on toodud tabelis 13.

Tabel 13. Logo baaskäsud

| Käsu nimetus                            | Lühend | Kirjeldus   |
|---|--------|---|
| Forward 100                             | FD 100 | 100 sammu ettepoole   |
| Back 100                                | BK 100 | 100 sammu tahapoole   |
| Right 90                                | RT 90  | Pööra paremale 90°  |
| Left 45                                 | LT 90  | Pööra vasakule 90°  |
| PenUp                                   | PU     | Pliiats üles  |
| PenDown                                 | PD     | Pliiats alla  |
| SetPenColor 1                           | SETPC  | Valitakse pliiatsi värv, kus parameetrik on värvi number            |
| SetBackgroundColor 1                    | SETBC  | Valitakse taustavärv, kus parameetrik on värvi number               |
| HideTurtle                              | HT     | Peida kilpkonn  |
| ShowTurtle                              | ST     | Näita kilpkonn  |
| Clear screen                            | CS     | Tühista ekraan  |
| Home                                    | -      | Kilpkonn liigub koju, mis asub ekraani keskel koordinaatidel [0][0] |
| bye                                     | -      | Lõpetab programmi tööd ja sulgeb programmi                          |
| save "C:/LogoTasks/TASK1.TXT            | -      | Salvestab programmi failisse  |
| load "C:/LogoTasks/TASK1.TXT            | -      | Laadib programmi üles failist                                       |
| repeat 4 [ <i>forward 100 left 90</i> ] | -      | Korda tegevust  |

Tegelikult käskude variandid sõltuvad logo versioonist, kuid nende versioonide paljud käsud on samad [13] [14].

## Lisa 7 - Testrakenduse failid

Juhitavuse testimine. Video demo1:

<https://youtu.be/hfOqMDmskuk>

Liugurite töö testimine. Video demo2:

[https://youtu.be/qKbmo3\\_PyncPlatvormi](https://youtu.be/qKbmo3_PyncPlatvormi)

Liikumise suuna leidmine. Video demo3:

[https://youtu.be/C6TZwn\\_WGf4](https://youtu.be/C6TZwn_WGf4)

Käsu salvestamise, laadimise ja täitmise jälgimise demonstratsioon. Video demo4:

<https://youtu.be/1NafkwWTYRM>

Kahe platvormi paralleelse töö demonstratsioon. Video demo5:

<https://youtu.be/jqQbFumAcRM>

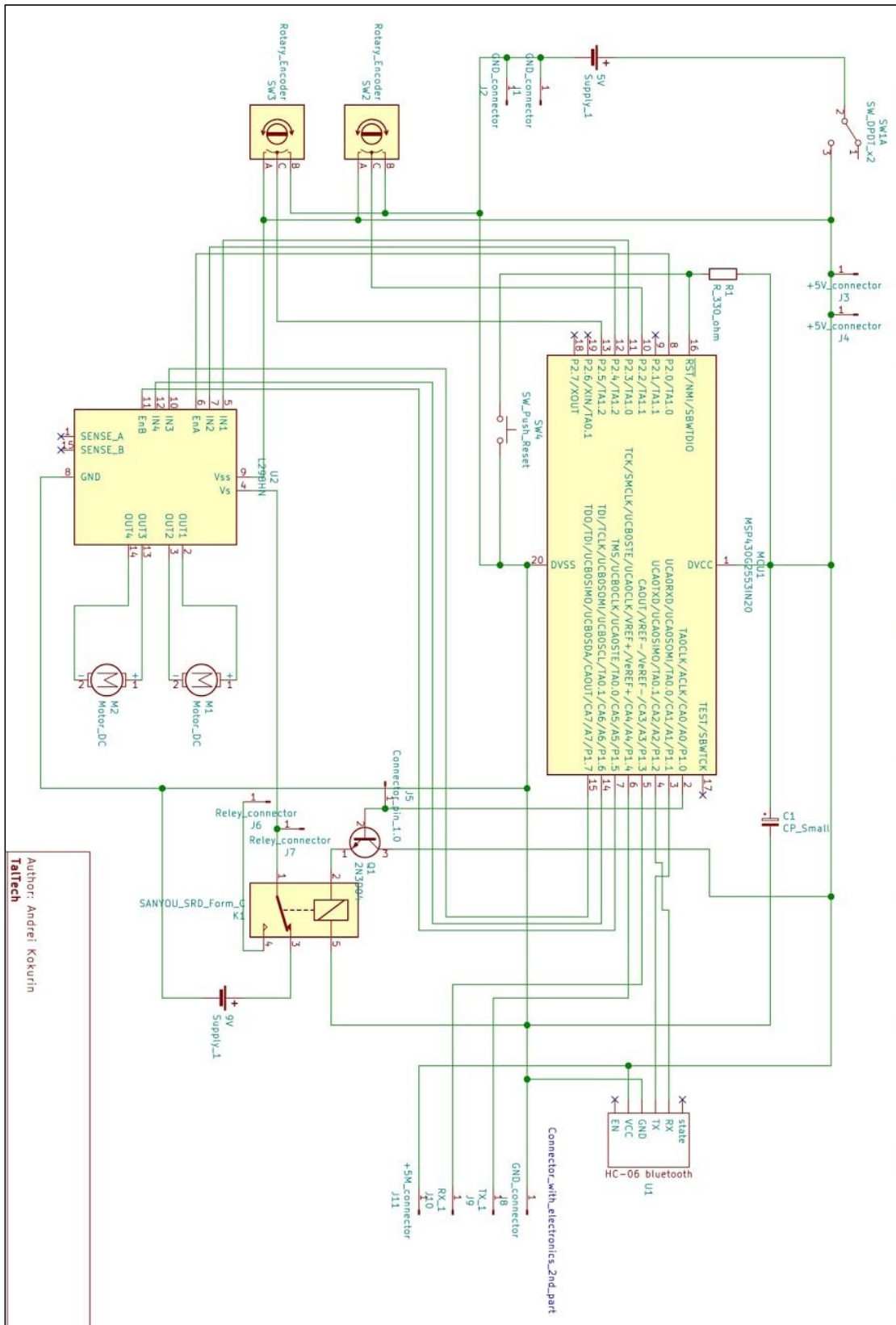
Kasutajaliidese versioonide link:

[https://drive.google.com/open?id=11WBBg\\_K\\_dmF4HOQv\\_zNB17ZOXWpOvbXx](https://drive.google.com/open?id=11WBBg_K_dmF4HOQv_zNB17ZOXWpOvbXx)

Link trükkplaadi failide kaustale:

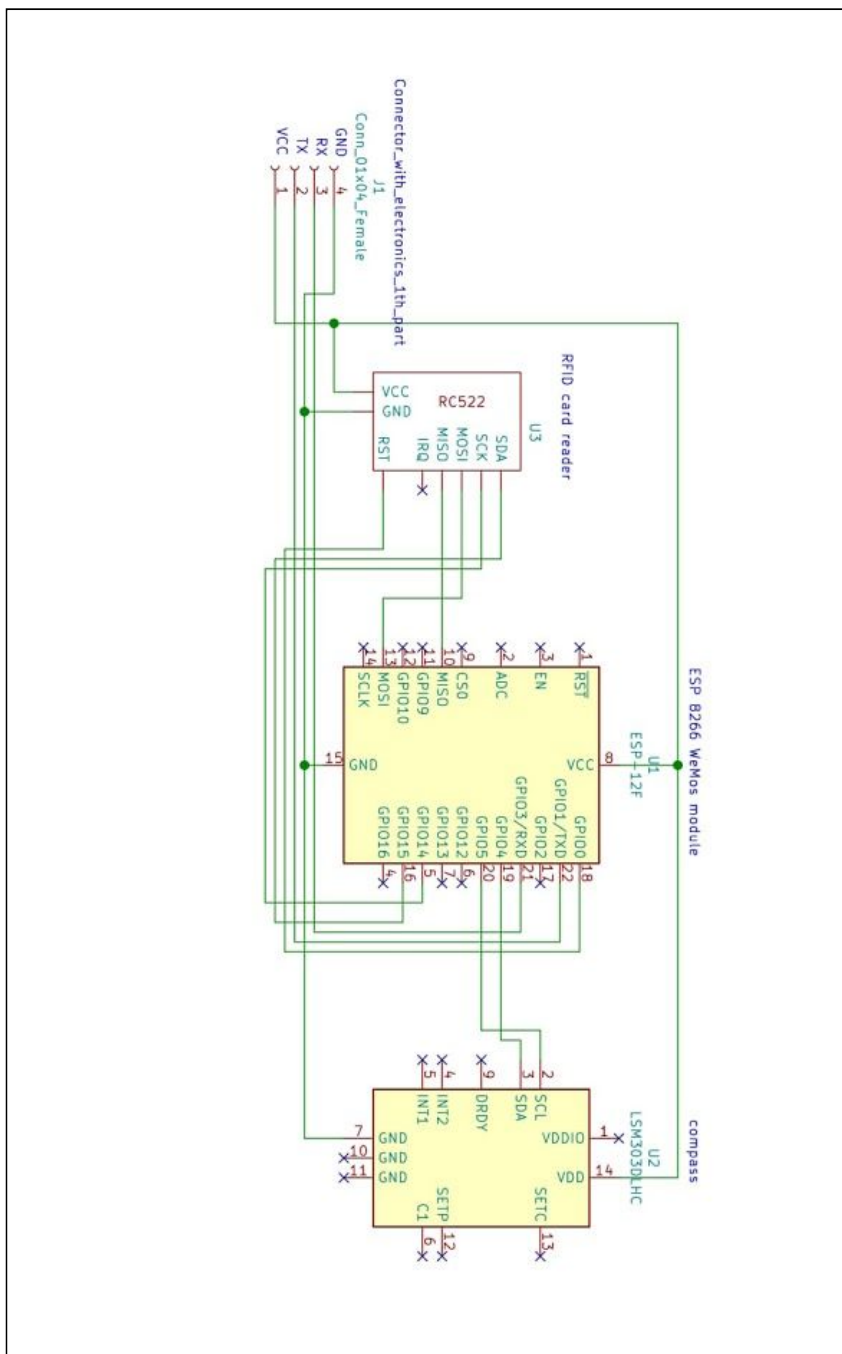
<https://drive.google.com/open?id=1jAFNSHX2px2kHhCJSziW7mje-vlMA1gr>

# Lisa 8 - Elektroonika skeemid ja joonised

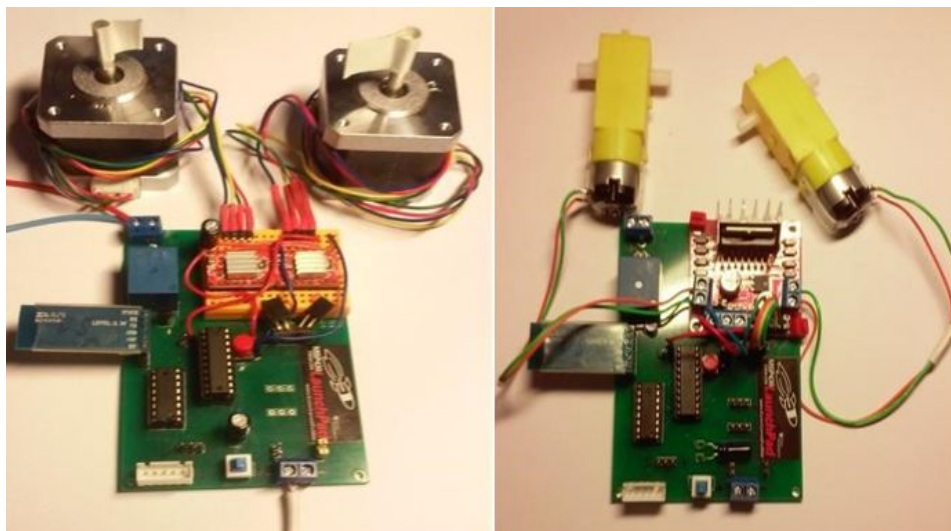


Joonis 26. Elektroonika I osa skeem KiCad programmis.

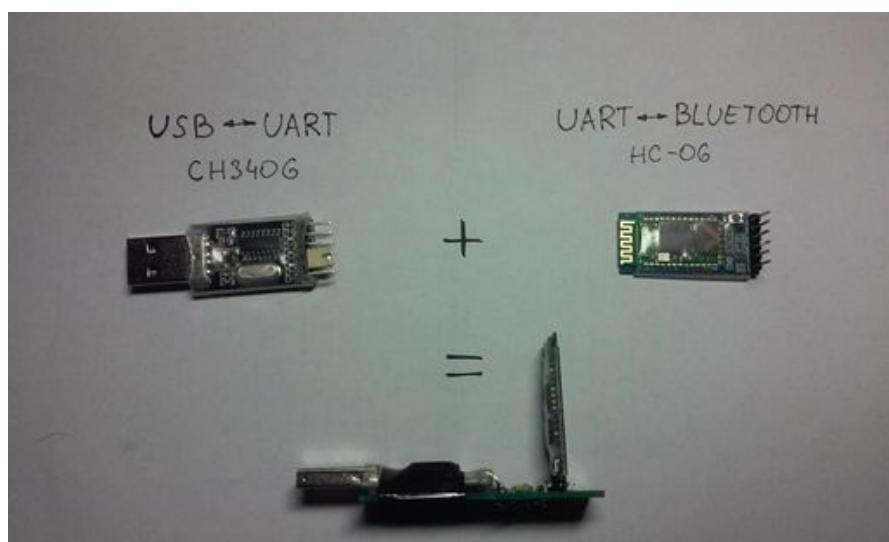




Joonis 27. Elektroonika II osa skeem KiCad programmis.



Joonis 28. Elektroonika I osa trükkplaat sobib nii samm-mootorite, kui ka alalisvoolu mootorite jaoks



Joonis 29. USB-Bluetooth signaali konverter

HC-06 moodulite ühendamise. HC-06 funktsionaalsus on lubab seadistada HC-06 moodulite paari selliselt, et nad ühendaks omavahel automaatselt. Selleks tuleb kasutada AT käskude. Üks moodul on seadistatud klient (master) rolliks ja teine serveri (slave) rolliks. Klient püüab ühendada automaatselt server mooduliga. Server lubab end siduda ainult ühe kliendiga. Moodulit tuleb ühendada programmatoriga või Arduino kontrolleri AT käskude saatmiseks. AT käsud saadakse ja võetakse vastu kiirusega 38400 baud. Tabelis 14. on toodud vajalikud AT käsud moodulite HC-06 ühendamiseks.

Tabel 14. Vajalikud AT käsud moodulite HC-06 ühendamiseks

| AT käsk                     | Vastus AT käsule | Selgitus   |
|-----------------------------|------------------|--|
| AT                          | OK               | Kontrollitakse, kas moodul on õigesti ühendatud  |
| AT+VERSION?                 |                  | Kontrollitakse moodulite versiooni. Omavahel on võimalik ühendada ainult HC-06 moodulit, kuid nende välimus on sarnane HC-05 moodulitega. Seega mõned müüjad saavad HC-06 asemele HC-05  |
| AT+BAUD<index>              | OK<baud kiirus>  | Valitakse side kiirust. Indeksina valitakse number, kusjuures 1-1200 baud, 2-2400 baud, 3-4800 baud, 4-9600 baud, 5-19200 baud   |
| AT+ADDR?                    | <seade aadress>  | Aadressiks on arvrida. Näiteks 14:2:11007. Ühendamiseks on vaja teada ainult serveri aadressi. Seda kirjutatakse kliendi mällu, selleks et klient otsiks vastavat serverit ühendamiseks. |
| AT+ROLE?                    | 0 või 1          | 0 - server, 1 - klient, algolekus on 0   |
| AT+ROLE = 1                 | OK               | Mooduli rolli seadistamine   |
| AT+RESET                    | -                | Pärast rolli muutmist tuleb taaskäivitada moodulit   |
| AT+BIND = <kliendi aadress> | OK               | Seda käsu kasutatakse kliendi pidevaks ühendamiseks serveriga  |
| AT+CMODE = 0                | OK               | See käsk keelab kliendil ühendada teiste seadmetega  |

## Lisa 9 - Käsurea programmide näited

Tabel 15. Käskude ja programmide näited

| Käsu näide                | Saadetise tüüp | Platvormile saadetakse pakett   | Selgitus   |
|---------------------------|----------------|---|--|
| F100;                     | F - forward    | U4F100;<br>U3b100   | Mine 100 sammu ettepoole   |
| B100;                     | B - back       | U4B100;<br>U3b100   | Mine 100 sammu tahapoole   |
| L90;                      | L - left       | U3L90;<br>U3b100  | Pööra 90° vasakule   |
| R90;                      | R - right      | U3R90;<br>U3b100  | Pööra 90° paremale   |
| D1000;                    | D - delay      | U5D1000;<br>U3b100  | Oota 1000 ms   |
| A45;                      | A - angle      | U3A45;<br>U3b100  | Pööra selliselt, et liikumissuuna nurk oleks 45° nulli suhtes  |
| MHow are you?;            | M - message    | U=MKuidas laheb?;<br>U3b100   | Saadab sõnumi "Kuidas laheb?"  |
| F100;R90;F100;            | -              | U4F100;<br>U2b33<br>U3R90;<br>U2b66<br>U4F100;<br>U3b100  | Mine 100 sammu ettepoole<br>Pööra 90° paremale<br>Mine 100 sammu ettepoole   |
| F100;L120;F100;L120;F100; | -              | U4F100;<br>U2b20<br>U3L120;<br>U2b40<br>U4F100;<br>U3b60<br>U4L120;<br>U3b80<br>U4F100;<br>U3b100 | Mine 100 sammu ettepoole<br>Pööra 120° vasakule<br>Mine 100 sammu ettepoole<br>Pööra 120° vasakule<br>Mine 100 sammu ettepoole |
| lalalalaF100;lalalalal    | F - forward    | U4F100;<br>U3b100   | Mine 100 sammu ettepoole   |

## Lisa 10 - Mobiilse platvormi nurga leidmise katse analüüs.

Tabel 16. Nurga asendi leidmise algoritmi katsetulemused.

| Katse nr. | Stardi-<br>positsiooni<br>nurk kraadides | Nõutav nurk<br>kraadides | Tegelikult<br>saavutatud<br>nurk kraadides | Nõutava ja tegeliku nurga<br>erinevuse absoluutväärtus |
|-----------|--|--------------------------|--|--|
| 1         | 0  | 180                      | 171  | 9  |
| 2         | 171                                      | 90                       | 85   | 5  |
| 3         | 85                                       | 225                      | 220  | 5  |
| 4         | 220                                      | 45                       | 47   | 2  |
| 5         | 47                                       | 0                        | 1  | 1  |
| 6         | 1  | 135                      | 132  | 3  |
| 7         | 132                                      | 315                      | 318  | 3  |
| 8         | 318                                      | 0                        | 357  | 3  |
| 9         | 357                                      | 180                      | 173  | 7  |
| 10        | 173                                      | 90                       | 89   | 1  |
| 11        | 89                                       | 315                      | 311  | 4  |
| 12        | 311                                      | 225                      | 222  | 3  |
| 13        | 222                                      | 45                       | 50   | 5  |
| 14        | 50                                       | 90                       | 89   | 1  |
| 15        | 89                                       | 270                      | 269  | 1  |
| 16        | 269                                      | 135                      | 135  | 0  |
| 17        | 135                                      | 0                        | 7  | 7  |
| 18        | 7  | 180                      | 183  | 3  |
| 19        | 183                                      | 45                       | 45   | 0  |
| 20        | 45                                       | 90                       | 88   | 2  |
| 21        | 88                                       | 255                      | 258  | 3  |
| 22        | 258                                      | 315                      | 313  | 2  |
| 23        | 313                                      | 45                       | 38   | 7  |
| 24        | 38                                       | 180                      | 188  | 8  |
| 25        | 188                                      | 315                      | 311  | 4  |
| 26        | 311                                      | 135                      | 104  | 31   |
| 27        | 104                                      | 135                      | 137  | 2  |
| 28        | 137                                      | 270                      | 268  | 2  |
| 29        | 268                                      | 45                       | 42   | 3  |
| 30        | 42                                       | 0                        | 3  | 3  |
|           |  |                          | aritmeetiline<br>keskmine                  | 4.33°  |
|           |  |                          | standardhälve                              | 5.55°  |

Lisas 7. on toodud link videole demo3, milles kasutaja määrab mobiilse platvormi suuna absoluutväärtusena kompassi põhja suuna suhtes. Mobiilne platvorm pöörab end vastavalt kasutaja ette antud suunale. Tabelisse on märgitud mobiilse platvormi katse alguses olnud asend, siis soovitud suunavektori nurk, kuhu platvorm end pöörama peab, ning platvormi realselt saavutatud suunavektori nurk. Tabelis võrreldakse erinevust soovitud ja tegeliku nurga vahel, et anda hinnang algoritmi toimimisele, mille tööpõhimõtte on kirjeldatud joonisel 18.

Katsetulemuste erinevuse aritmeetiline keskmine on  $4.33^\circ$  ja standardhälve on  $5.55^\circ$ , nagu tabelist on näha, siis katsetulemuste erinevused varieeruvad, on tulemusi, kus erinevus on minimaalne, praktiliselt null kraadi, kuid ühe katse tulemusena oli erinevus lausa 31 kraadi.