



TALLINNA TEHNIKAÜLIKOO

INSENERITEADUSKOND

Virumaa kolledž

## **Markerite abil navigeeriv laorobot**

### **Warehouse robot navigating with markers**

TELEMAATIKA JA ARUKATE SÜSTEEMIDE ÕPPEKAVA LÕPUTÖÖ

Üliõpilane: Kiur Seiler

Üliõpilaskood: 183540EDTR

Juhendaja: Sergei Pavlov, lektor

# 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.

"...." ..... 20.....

Autor: .....

/ allkiri /

Töö vastab rakenduskõrgharidusõppe lõputööle/magistritööle esitatud nõuetele

"...." ..... 20.....

Juhendaja: .....

/ allkiri /

Kaitsmisele

lubatud

"...." ..... 20.....

Kaitsmiskomisjoni esimees .....

/ nimi ja allkiri /

# **LIHTLITSENTS LÕPUTÖÖ ÜLDSUSELE KÄTTESAADAVAKS TEGEMISEKS JA REPRODUTSEERIMISEKS**

Mina Kiur Seiler (sünnikuupäev: 29.01.1999)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

Markerite abil navigeeriv laorobot, mille juhendaja on Sergei Pavlov,

1.1. reprodutseerimiseks säilitamise ja elektroonilise avaldamise eesmärgil, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta kolmandate isikute intellektuaalomandi ega isikuandmete kaitse seadusest ja teistest õigusaktidest tulenevaid õigusi.

# TalTech Inseneriteaduskond Virumaa kolledž

## LÕPUTÖÖ ÜLESANNE

**Üliõpilane:** Kiur Seiler, 183540

Õppekava, peeriala: EDTR Telemaatika ja arukad süsteemid

Juhendaja(d): Lektor, Sergei Pavlov, sergei.pavlov@taltech.ee

### Lõputöö teema:

(eesti keeles) *Markerite abil navigeeriv laorobot*

(inglise keeles) Warehouse robot navigating with markers

### Lõputöö põhieesmärgid:

1. Teha suurem robot, millel on kandevõime
2. Luua markerite abil navigeerimise algoritm

### Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Roboti kere konstruktsioon	30.10.22
2.	Puldi juhtkood	13.11.22
3.	Markerite abil navigeerimiskood	18.12.22
4.	Kirjaliku osa täiendus	14.03.23
5.	Kirjaliku osa viimistlus ja vormindus	01.04.23

**Töö keel:** Eesti keel

**Lõputöö esitamise tähtaeg:**

"5"juuni 2023a

**Üliõpilane:** .....  
/allkiri/

"....."..... 20.....a

**Juhendaja:** .....  
/allkiri/

"....."..... 20.....a

**Programmijuht:** .....  
/allkiri/

"....."..... 20.....a

# SISUKORD

EESSÕNA .....	7
LÜHENDITE JA TÄHISTE LOETELU .....	8
1. SARNASED OLEMASOLEVAD ROBOTID .....	10
2. ERINEVAD NAVIGATSIOONI VÕIMALUSED .....	12
2.1. GPS .....	12
2.2. Lidar .....	12
2.3. Jooned .....	12
2.4. Odomeetria .....	12
2.5. Kaamerad .....	12
3. ROBOTI EHITUS .....	13
3.1. Roboti raam .....	13
3.2. Roboti elektroonika .....	14
3.2.1. Elektroonika diagramm .....	18
3.3. Tekkinud probleemid ja muudatused robotile .....	20
3.3.1. Raspberry Pi 4 toite probleem .....	20
3.3.2. Robot ei suuda pöörata .....	20
3.4. Muudatused .....	21
3.4.1. Ekraani lisamine .....	21
3.4.2. Led tuled .....	21
3.4.3. Akrüülpaneelid .....	22
3.4.4. Roboti juhtmestik .....	22
3.4.5. Mootori toed .....	23
3.4.6. Ultraheli sensorite paigaldus .....	23
4. ROBOTIS KASUTUSEL OLEV TEHNOLOOGIA .....	24
4.1. I2C SCK/SDA .....	24
4.2. UART .....	24
4.3. ArUco markerid .....	25
4.4. OpenCV .....	25
4.5. OpenCV installatsioon .....	26
4.6. Ultraheli sensorid .....	28
4.7. ThunderBorg mootori kontrollrite seadistamine .....	29
5. ÄPP .....	30
5.1.1. Puldi ja/või äpi juhtkood .....	32

5.1.2. Navigatsioonikood .....	32
6. KOMPONENTIDE MAKSUMUS .....	37
7. EDASISED ARENDUSED.....	38
7.1. Tugevam kere.....	38
7.2. Takistuste vältimine.....	38
7.3. Äpp .....	38
7.4. Kaamera ümberpaigutamine .....	38
KOKKUVÕTE .....	39
SUMMARY.....	40
KASUTATUD KIRJANDUSE LOETELU .....	41
LISAD .....	43

## **EESSÕNA**

Uurimistöö eesmärk on kavandada ja konstrueerida töötav robot, mis oleks väikeettevõtetele taskukohasem ning suudab autonoomselt liikuda laohoonete keskkonnas ning efektiivselt tuvastada ja kasutada markereid navigeerimisel.

Käesoleva lõputöö sõnastamine põhineb autori algatusel ning kooskõlastati lõputöö juhendajaga, kelle nõuanded ja tagasiside on olnud toeks lõputöö teostamise vältel.

Märksõnad: robot, autonoomsus, navigatsioon, Raspberry Pi, markerid, diplomitöö.

## **LÜHENDITE JA TÄHISTE LOETELU**

GPS – (Global Positioning System) tehnoloogia, mis võimaldab täpset positsioneerimist.

I2C – (Inter-Integrated Circuit) kommunikatsiooniprotokoll elektrooniliste seadmete vahel.

Lidar – (Light Detectidon and Ranging) seade, mis kasutab valguskiirt, et leida kaugus objektist.

UART – (Universal Asynchronous Receiver/Transmitter) kommunikatsiooniprotokoll elektrooniliste seadmete vahel.



## SISSEJUHATUS

Tööstusrobotite kiire arenguga muutuvad tööstus- ja laorobotid aina uhkemaks ja kõrgtehnoloogilisemaks, jättes maha väikeettevõtjad kellel pole võimalust hankida kalleid roboteid ja sammu pidada pidevate muutustega.

Lõputöö eesmärgiks on luua robot, mis suudab kanda pakke, ning ei vaja interneti ühendust ega kalleid sensoreid, et oma keskkonnas navigeerida.

Lõputöö ülesandeks on luua robot ja sellega kaasnev navigatsioonilahendus. Võimaluse korral lisada takistuste vältimise süsteem ja mobiiläpp, millega robotit juhtida ja käsklusi saata.

Eeldatavaks tulemuseks on robot, mis navigeerib laoruumides kasutades markereid ning jälitab töötajat määratud markeri alusel. Robot ei vaja välise süsteemi sekkumist navigeerides ja suudab liikuda punktist a punkti b kasutades vaid roboti küljes olevaid süsteeme.

Tulemuse saavutamiseks kasutab autor OpenCV paketi põhjal ArUco markerite tuvastust ja nende positsiooni määramist kaamera vaateväljas.

Antud lõputöö tulemusena valmiv robot ja selle navigeerimissüsteem pakub odavamat alternatiivi tänapäevastele lao- ja tööstusrobotitele ning aitab väike- ja keskmise suurusega ettevõtetel integreerida roboteid oma töökeskkonda ning lihtsustada füüsilist töökäiku.

Esimeses peatükis kirjeldatakse olemasolevaid lahendusi ning nende navigatsiooni seadmeid. Lisaks kirjeldatakse erinevate navigatsioonivahendite tööpõhimõtteid.

Teises peatükis kirjeldatakse valminud roboti ehitust, selle komponente ja robotis kasutatud tehnoloogiat. Lisaks on välja toodud roboti komponentide maksumused. Edasi kirjeldatakse töö käigus tekkinud probleeme ning nende lahendust ja nende käigus tekkinud muudatusi robotile.

Edasi kirjeldatakse robotis kasutusel olevate koodide tööpõhimõtteid.

Lõpetuseks kirjeldatakse plaanis olevaid edasiarendusi.

## 1. SARNASED OLEMASOLEVAD ROBOTID

Tehnoloogia kiire areng toob kaasa moodsate ja tehnikast täidetud robotite tootmise, mis omakorda tähendab seda, et väiksematel ettevõtetel on raske roboteid implementeerida oma töökeskkonda. Tänapäevased robotid on täidetud kallite anduritega, mis tõstavad robotite hinda mitme tuhande euroni ja mõnikord isegi mitmekümnete tuhande eurodeni. Allpool on välja toodud tabel roboti liideste, kasutusala ja hinnaga.

Olemas olevad robotid ja nende võimalused:

- Starship – kasutab navigeerimiseks mitut kaamerat ja kaugusandureid. Lisaks internet ühendus ja GPS.  
Peamine kasutusala: Toidu transport.  
Hind: pole müügiks
- Gitamini – Kasutab kaameraid ja sensoreid, et ära tuvastada oma omanik ja sõita tema järgi.  
Peamine kasutusala: Personaalne pakikandja.  
Hind: \$1,850
- Effibot – kasutab Lidar süsteemi, et oma keskkonnas navigeerida ja inimest jälitada.  
Peamine kasutusala: Laorobot  
Hind: Pole teada
- RB-VOGUI – kasutab Lidar süsteemi, mitmeid kaameraid, GPS, raadio kontrolli, et navigeerida oma keskkonnas ning suuri rattaid, mis lubavad liikuda ka ebatasasel pinnal.  
Peamine kasutusala: Laorobot, kuller, sõjavägi  
Hind: Pole teada

Tabel 2.1 Olemasolevate robotite tabel

Robot	Liideste nimekiri	Peamine kasutusala	Hind
Starship	Kaamerad, GPS, kaugusandurid, internet	Toidu transport	Pole müügiks
gitamini	Kaamerad, radar	Personaalne pakikandja	1850
RB-vogui	Lidar, Kaamerad, GPS, radio kontroll	Laorobot, kuller, sõjavägi	Pole teada
Effibot	Lidar, kaamerad	Laorobot	Pole teada
Kiva (Amazon)	Skänner, odomeetria, kaugussensor	laorobot	Pole teada
InVia	Kaamerad, odomeetria	laorobot	Pole teada

Autori valik:

Selleks, et luua võimalikult taskukohast robotit, jätab autor välja kallid sensorid nagu lidar ja kasutab navigatsiooniks markereid mida robot näeb kaamera abil ja vastavalt käsklusele navigeerida oskab. Takistuste vältimiseks kasutab autor ultraheli sensoreid.

## **2. ERINEVAD NAVIGATSIOONI VÕIMALUSED**

### **2.1. GPS**

GPS navigeerimine töötab tänu rohkem kui kolmekümnele satelliidile mis ümber maailma tiirlevad ja pidevalt signaale saadavad. GPS vastuvõtja kuulab neid signaale ja kui see on saanud neli või rohkem signaali erinevatelt satelliitidelt, on võimalik välja arvutada asukoht maa peal [1]. Robotid saavad kasutada seda informatsiooni, et liikuda ühest punktist teise.

### **2.2. Lidar**

Lidar on seade, mis mõõdab kaugust, saates välja laserit ja mõõtes millal see laser tagasi peegeldub. Seda informatsiooni kasutades saab robot aru kui kaugel on takistus. Lidarit saab kasutada kaardi loomiseks, mida robot kasutab navigeerimiseks. [2]

### **2.3. Jooned**

Joonejälgi sensor toimib kasutades infrapuna LED'i, mis annab infrapuna valgust ja infrapuna sensorit, mis loeb tagasipeegelduva valguse intensiivsust. Näiteks hele põrand peegeldab tagasi rohkem kui must joon põrandal. Kasutades neid sensoreid ravis, saab robot teada kas see on joone peal või tuleb sõidusuunda kohendada. [3]

### **2.4. Odomeetria**

Odomeeter loeb rataste pöördeid, kasutades selleks spetsiaalseid infrapuna sensoreid, aga on võimalik ka kaameraga. Teades rataste suurust saab välja arvutada kui palju on robot liikunud ja välja arvutada roboti positsioon. Odomeetria ei ole väga täpne ning on efektiivne teiste navigatsiooni võimalustega koostöös. [4]

### **2.5. Kaamerad**

Kaamerad on robotikas mitmekülgsed vahendid. Neid saab kasutada markerite äratundmiseks, aga ka masinägemise abil esemete ja objektide tuvastamiseks. Veel on olemas kaameraid mis tuvastavad kaugust. Robotikas kasutatavad kaamerad on disainitud pilditeabe kogumiseks ning saadab selle töötlemiseks kokku surumata kujul. Tavaliste tarbija kaameratega surutakse pildiandmed kokku, mis tagab sujuva pildi, kuid ei paku piisavalt kvaliteeti robotikarakenduste jaoks [5]. Parema tulemuse saavutamiseks võiks kaamerat kasutada teiste navigatsiooni võimalustega koostöös.

### 3. ROBOTI EHITUS

#### 3.1. Roboti raam

Roboti raami ehitamiseks on valitud alumiinium torud ning plastikust ühendusnurgad (vt Joonis 3.1).

Alumiiniumtorud on valitud nende odava hinna, kerge töötlemise ja piisava tugevuse pärast, mille tõttu on need sobivad prototüübi loomisel.

Plastikust ühendusnurgad mahuvad täpselt alumiiniumtorude sisse ning on hea lahendus raami kokku sobitamisel, sest need annavad võimaluse raami kokku panna ja lahti võtta, et raami vajadusel kohandada või muuta.



Joonis 3.1 Raami materjal

Roboti põhjaks on kasutatud kodus olemasolevat vineerplaati, kuhu on kinnitatud roboti elektroonika (vt Joonis 3.2).



Joonis 3.2 Valmis raam koos vineerplaadiga

## 3.2. Roboti elektroonika

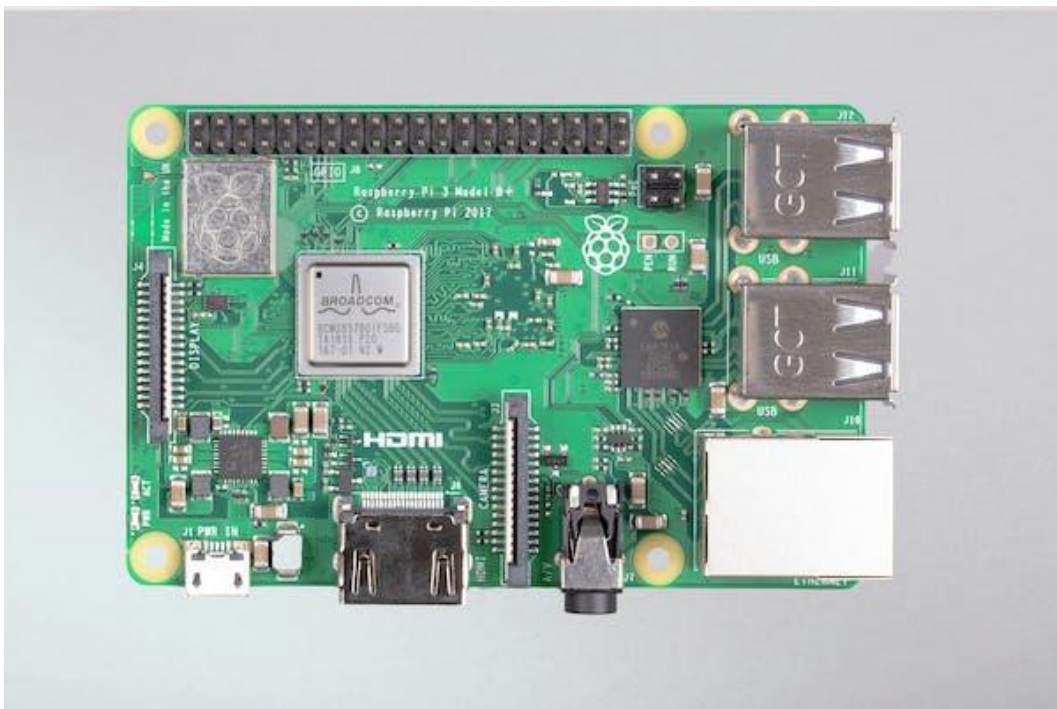
Järgnevaiks kirjeldatakse autori valitud komponente, nende otstarvet robotis ning miks need on valitud.

- Raspberry Pi 3

Roboti tegevust juhib Raspberry Pi 3B+ (vt Joonis 3.3) mikroprotsessor, mis tegeleb markerite tuvastusega ja juhib mootori kontrollrite tööd läbi I2C protokoll.

Algselt valitud Raspberry Pi 4B+ oleks olnud parem valik selle parema videotötluse pärast, mis on vajalik markerite tuvastuseks aga selle suurem voolutarbimine tähendas, et see ei saanud mootori kontrolleriilt piisavat toidet.

Raspberry Pi platvorm on valitud selle taskukohasuse ning autori olemasolevatest kogemustest platvormiga.



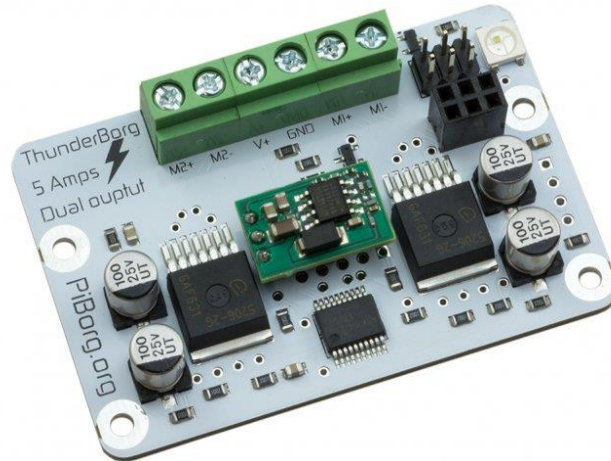
Joonis 3.3 Raspberry Pi 3B+ [6]

- Thunderborg mootori kontrolleri

Mootoreid kontrollivad 2 Thunderborg (vt Joonis 3.4) kontrolleri, igale rattapaarile üks, mis suudavad väljastada 5 amprit mootori kohta. Kontrolleriid suhtlevad omavahel I2C SCK/SDA protokoll abil ja toidavad ka Raspberry Pi'd. Maksimum voolu pinge on 35V.

Kontrollerid valik tuleneb sellest, et need on tehtud PiBorg firma poolt, kes tegutsevad Raspberry Pi robotika valdkonnas. Kontrollerid on dokumenteeritud ja ülessättimine on hästi kirjeldatud.

Testimise käigus selgus, et 5 amprit jääb natuke väheks, aga kontrollerid vähendavad automaatselt mootori toidet kui 5 amprit on ületatud.



Joonis 3.4 ThunderBorg mootori kontroller [7]

- Mootorid

Robot liigub tänu neljale 12V 450 RPM alalisvoolu mootorile vt (Joonis 3.5).

Mootorid on valitud sellepärast, et need olid küllaltki taskukohased ja usaldusväärset müüjalt. Lisaks on neile sisseehitatud reduktor, mis vähendab kiirust ja seega suurendab jõudu.

Kuna mootorid on küllaltki kiired, siis pole vaja neid täisvõimsusel kasutada. Lõputöö esitamise hetkel on mootori kasutus 20 kuni 50 protsenti mootori täisvõimsusest.

Testimise käigus selgus, et mootorid tõmbavad umbes 2.5 amprit voolu otse sõites ja umbes 5 amprit pöörates või kinni jäädes.

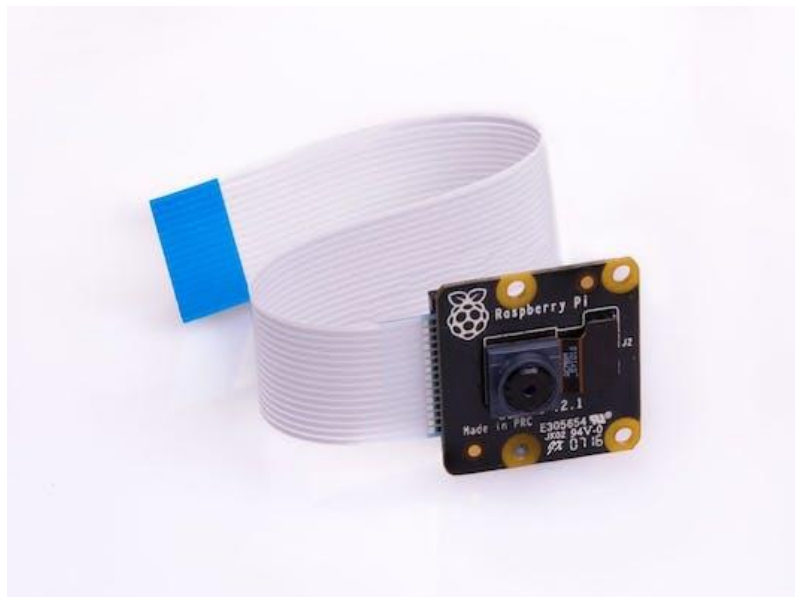


Joonis 3.5 Alalisvoolu mootor [8]

- Kaamera

Nägemiseks kasutab robot Raspberry Pi NoIr kaamerat (vt Joonis 3.6), mis teeb lihtsamaks ka pimedas nägemise.

Valitud sellepärast, et tulevikus katsetada infrapuna lampidega pimedas navigeerimist. Lisaks on Raspberry Pi kaamerad juba kokku sobivad Raspberry Pi mudelitega ja ei vaja erilist seadistust, et kasutada.



Joonis 3.6 NoIr kaamera [9]



- Raspberry Pi Pico

Ultraheli sensorite tööd robotis kontrollib Raspberry Pi Pico mikrokontroller (vt Joonis 3.8).

Mikrokontroller jälgib Bluetooth mooduli saadetud andmeid ja ultraheli sensorite andmeid ja saadab vastavalt vajalikud käsklused Raspberry Pi 3'le.

Mikrokontroller on valitud selle taskukohase hinna ja laialdase saadavuse pärast. Lisaks on Pico võimsam kui Arduino.

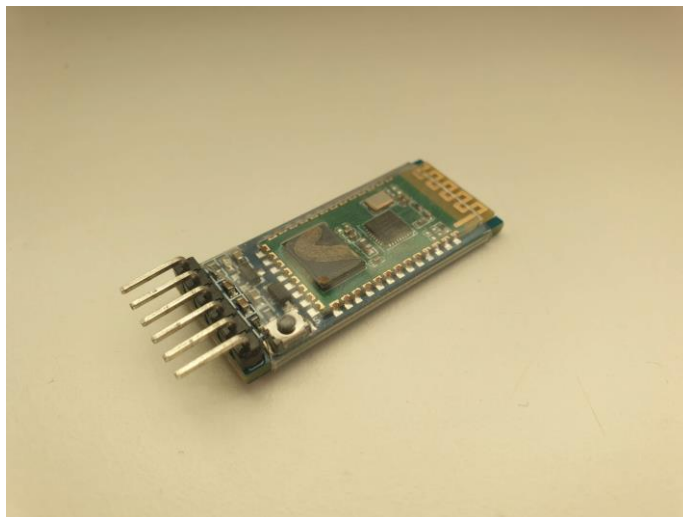


Joonis 3.7 Raspberry Pi Pico [10]

- HC-05

HC-05 Bluetooth moodul (vt Joonis 3.8) on robotis kasutusel äpiga suhtlusel. Moodul edastab äpist saadud andmed Raspberry Pi Pico mikrokontrollerile kasutades selleks UART ühendust.

Valitud selle lihtsa kasutusviisi ning autori varasema kasutuskogemuse pärast.

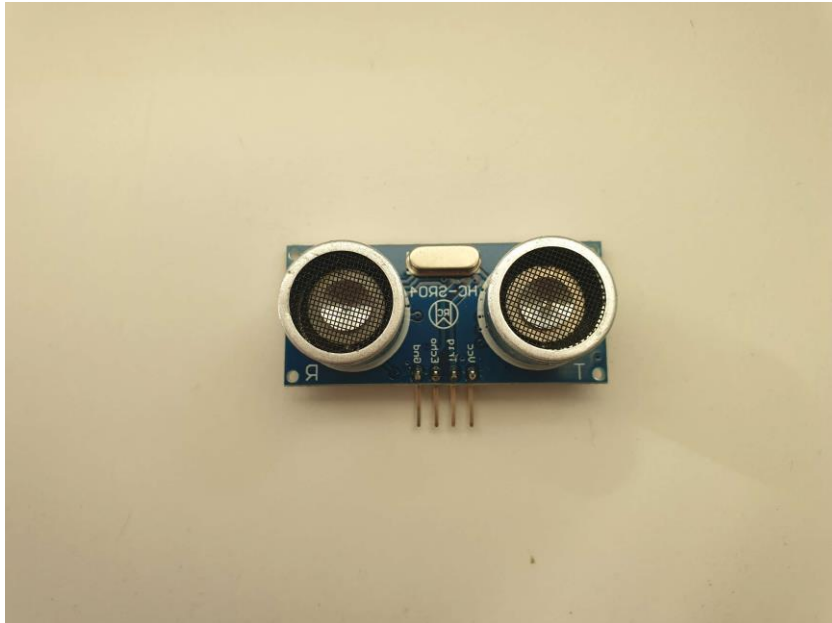


Joonis 3.8 HC-05 Bluetooth moodul

- HC-SR04 ultraheli sensorid

Takistuste vältimiseks kasutab robot kolme HC-SR04 ultraheli sensorit (vt Joonis 3.9), mis asetsevad ees ja külgedel. Sensorid saadavad info Raspberry Pi Picole mis vastavalt andmetele saadab juhiseid Raspberry Pi 3'le.

Sensordid on valitud nende taskukohase hinna ning laialdase saadaval oleva informatsiooni pärast.



Joonis 3.9 HC-SR04 ultraheli sensor

### **3.2.1. Elektroonika diagramm**

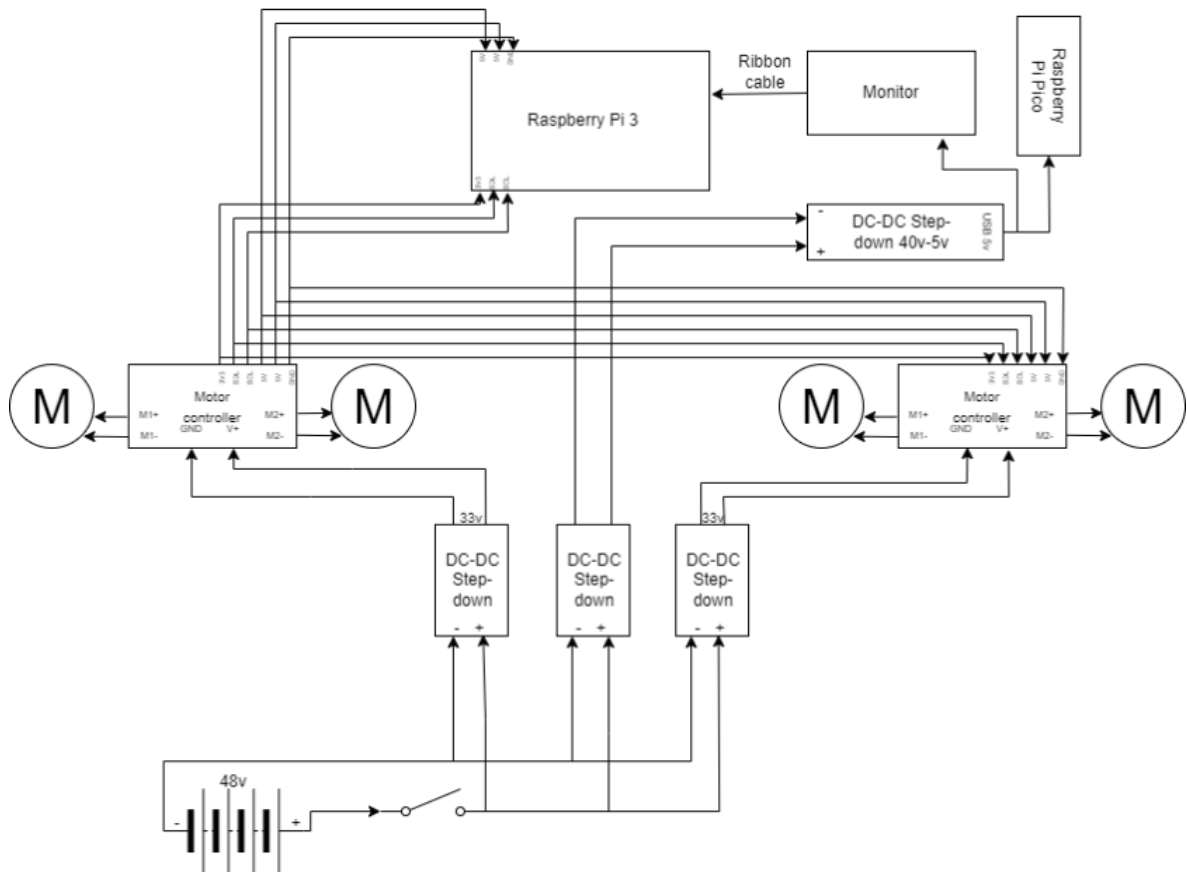
Allolev skeem (vt Joonis 3.10) illustreerib kuidas roboti elektroonika on ühendatud.

Aku toidab kahte ThunderBorg mootorikontrollerit. Mõlemal kontrolleril on individuaalne DC/DC step-down moodul, mis madaldab akust tulevat pinget kontrollerile aktsepteeritavale tasemele.

Lisatud on kolmas DC/DC step-down moodul, mis jookseb veel ühte DC/DC step-down moodulisse ja annab välja 5V. selle pealt saavad toite Raspberry Pi 3'ga ühendatud ekraan ja Raspberry Pi Pico, mis kontrollib ultraheli andureid ja Bluetooth moodulit millest tulevad käsklused äpist.

Mõlemad kontrollerid juhivad kahte mootorit ning neid saab individuaalselt kontrollida.

Kontrollerid ühenduvad Raspberry Pi'ga läbi kuue juhtme, millest neli on kasutusel suhtluse jaoks ja kaks toidavad Raspberry Pi'd. Kontrollerid ühenduvad üksteisega samamoodi.



Joonis 3.10 Roboti elektronika diagramm

### **3.3. Tekkinud probleemid ja muudatused robotile**

#### **3.3.1. Raspberry Pi 4 toite probleem**

Mootori kontrollid ei suutnud toita Raspberry Pi 4 ning see lülitus välja kui mootoreid juhtida. Selle lahendamiseks asendasin Raspberry Pi 4 Raspberry Pi 3 vastu, mille voolutarbimine on väiksem.

#### **3.3.2. Robot ei suuda pöörata**

Testimise käigus selgus, et robot on liiga pikk ning seetõttu ei suuda efektiivselt pöörata koha peal. Selle lahendamiseks tegin roboti lühemaks ning eemaldasid ühe ratta paari, nüüd liigub robot nelja ratta peal (vt Joonis 3.11).

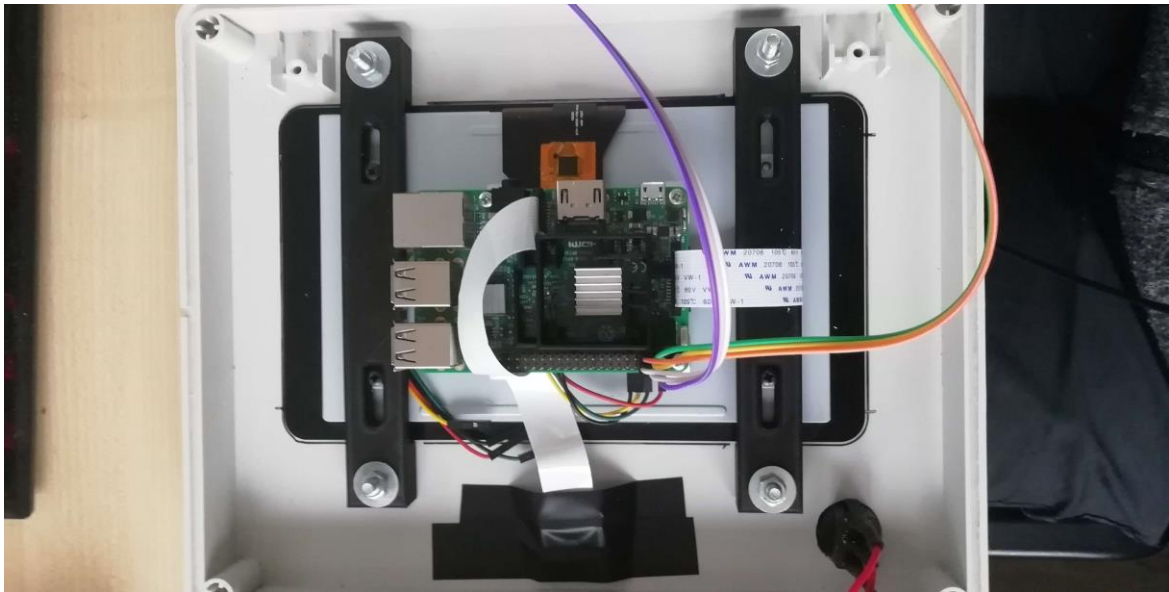


Joonis 3.11 Robot peale muudatusi

## 3.4. Muudatused

### 3.4.1. Ekraani lisamine

Robotile on lisatud ekraan, mis töö tegemise ajal ei paku suurt funktsionaalsust peale süsteemi otse kontrollimise. Ekraani kinnitamise jaoks disainis ja 3D printis autor klambrid (vt Joonis 3.12) (vt Lisa 1), mis hoiavad ekraani ning kinnituvad karbi külge kus asetsevad Raspberry Pi ning selle kaamera. Ekraani lisamisega kaasnes ka fikseeritud kinnituspunkt Raspberry Pi jaoks, mis varasemalt oli kinnitatud teibiga. Tulevikus soovib autor seda ekraani kasutada kasutajaliidesena ning lisada robotile animeeritud silmad, et robotil oleks rohkem iseloomu.



Joonis 3.12 Ekraani kinnitus

### 3.4.2. Led tuled

Robotile on lisatud ka led tuled (vt Joonis 3.13), mis töö esitamise ajaks ei ole integreeritud roboti voolusüsteemi.

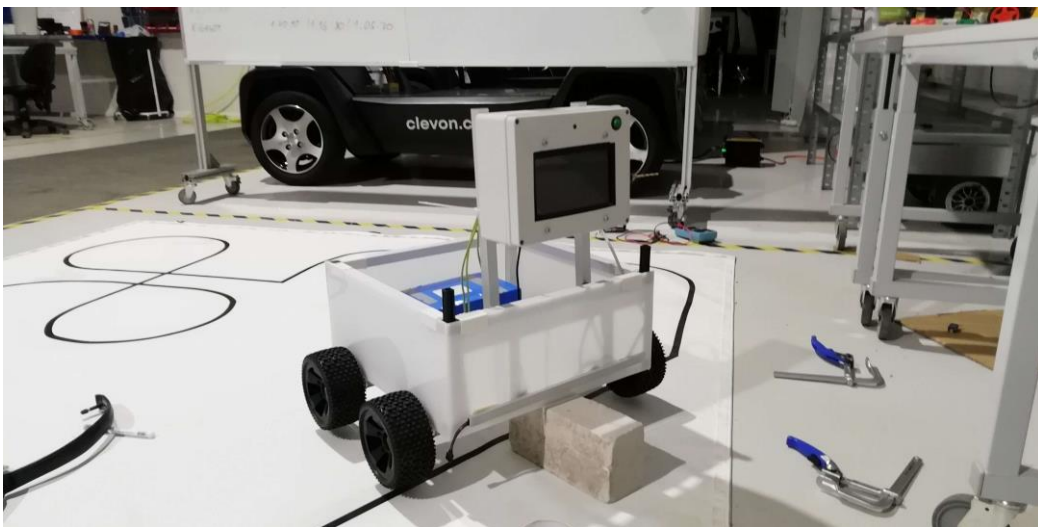
Led tuled teevad roboti laokeskkonnas rohkem nähtavaks ning tulevikus, kui autor lisab adresseeritavad led tuled, hakkavad need teavitama roboti järgmist liigutust.



Joonis 3.13 Led tuled

### 3.4.3. Akrüülpaneelid

Robotile on lisatud akrüülplaatidest korpus (vt Joonis 3.14). See on dekoratiivne muutus, mis peidab roboti elektroonika, aga annab ka raamile jäikust juurde.

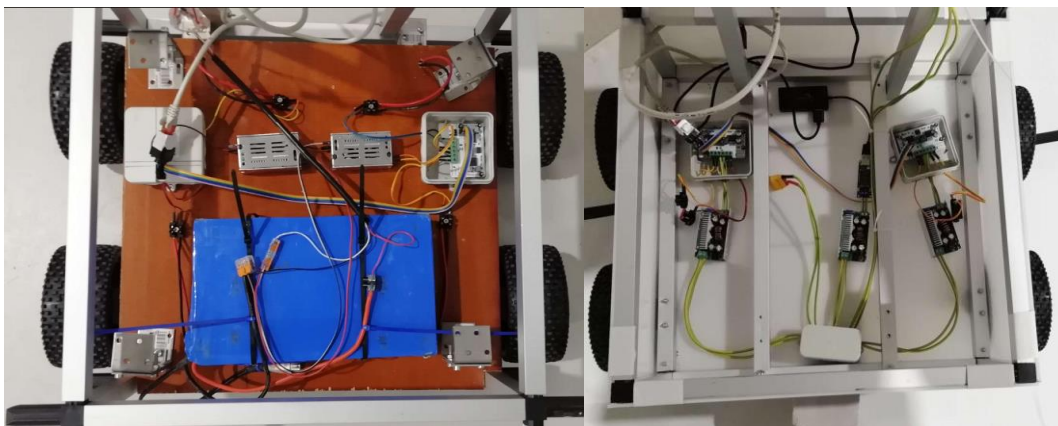


Joonis 3.14 Robot koos akrüülpaneelidega

### 3.4.4. Roboti juhtmestik

Roboti juhtmestik (vt Joonis 3.15) on välja vahetatud, seetõttu paremini organiseeritud ja visuaalselt selgemini mõistetav. Ühtlasi on mugavam akut eemaldada tänu lisatud pistikule.

Lisaks on paigaldatud kaks tugilatti, mis hoiavad akut üleval, et selle alt juhtmeid läbi viia.



Joonis 3.15 Roboti juhtmestik enne ja pärast

### 3.4.5. Mootori toed

Testimise käigus selgus, et mootori kinnitused on liiga pehmed, et mootoreid stabiilselt paigal hoida ja mootorid hakkasid raskuse all viltu vajuma. Selle parandamiseks sai modelleeritud ja 3D printitud klambrid (vt Joonis 3.16) (vt Lisa 2), mis toetavad mootori vajuvat otsa.



Joonis 3.16 Mootorid enne ja pärast tugede lisamist

Pildilt on näha kuidas mootorid vajusid enne toe lisamist ning peale tugede lisamist on maapinnaga paralleelselt.

### 3.4.6. Ultraheli sensorite paigaldus

Ultraheli sensorite paigaldamiseks sai modelleeritud ja 3D printitud ümbrised (vt Joonis 3.17) (vt Lisa 3).



Joonis 3.17 Ultraheli sensori ümbrised

## 4. ROBOTIS KASUTUSEL OLEV TEHNOLOOGIA

### 4.1. I2C SCK/SDA

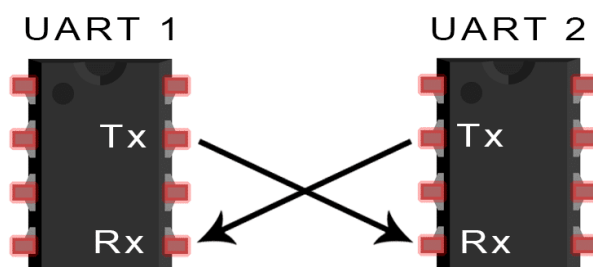
I2C on elektrooniliste seadmete suhtlus protokoll, mis laseb *master* seadmetel suhelda *slave* seadmetega. Mis teeb selle populaarseks on see, et ainult kahe juhtmega saab kontrollida paljusid *slave* seadmeid. *Master* seadmeid võib ka olla palju aga need peavad käskude jagama kordamööda. [11]

Robotis on 2 Mootori kontrolleriit, mis töötavad *slave* seadmetena ja Raspberry Pi 3 on *master* seade.

### 4.2. UART

UART (Universal Asynchronous Receiver/Transmitter) on kommunikatsiooniprotokoll, mis kasutab kahte juhet, et edastada andmeid. See kasutab Tx juhet andmete saatmiseks ja Rx juhet andmete vastu võtmiseks (vt Joonis 4.1). UART töötab asünkroonselt, mis tähendab, et see ei kasuta kella funktsiooni andmete sünkroniseerimiseks, vaid kasutab algus- ja lõppbitti. Selleks, et suhtlus töötaks peavad juhtmed ühendama ühe seadme Tx ühenduse, teise seadme Rx ühendusega ja vastupidi. [12]

Robotis on kasutusel UART ühendus Raspberry Pi Pico ja Bluetooth mooduli suhtlusel.



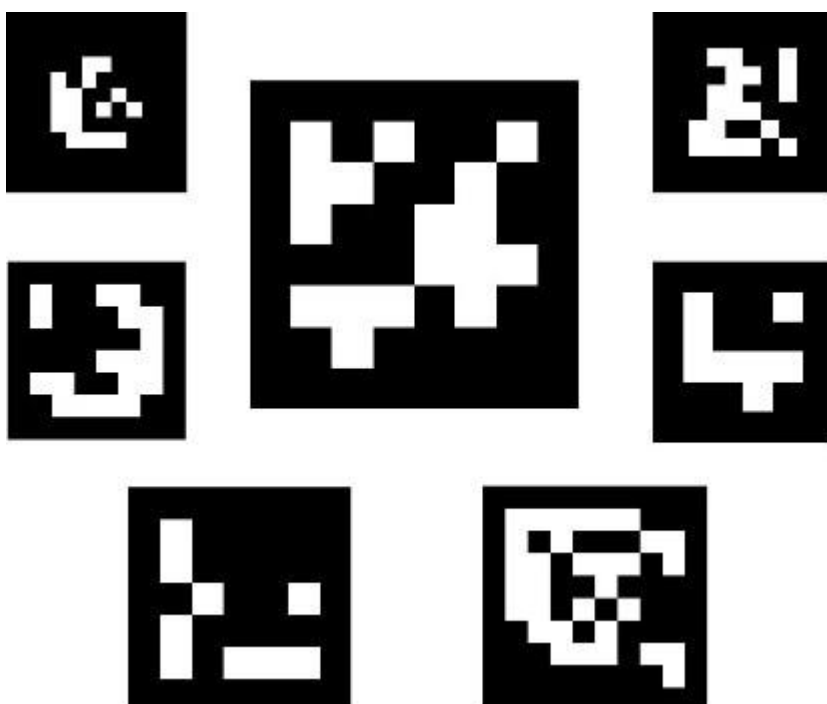
Joonis 4.1 UART ühendus [12]



### 4.3. ArUco markerid

ArUco markerid (vt Joonis 4.2) on binaar ruudustik markerid mida kasutatakse kaamera positsiooni hindamiseks. Marker koosneb mustast välisraamist, mis teeb nende tuvastamise kaamera jaoks lihtsaks ja sisemist binaarmaatriksist mis määrab markeri ID. Markerid võivad olla keskkonnas mistahes asendis ning sisemise maatriksi järgi on võimalik määrata, millises asendis marker asetseb. Nende põhieelis on see, et nende tuvastus on kiire, lihtne ja töökindel. [13][14]

Robot kasutab neid markereid, et leida oma positsioon laoruumides.



Joonis 4.2 ArUco markeri näited [13]

### 4.4. OpenCV

OpenCV on masinnägemise ja masinõppe vabavara. OpenCv loodi masinnägemisrakenduste ühise infrastruktuuri pakkumiseks ning kiirendada masintajumise kasutust kommertstoodetes. Kuna OpenCV on Apache 2 litsentsiga toode, on ettevõtetal lihtne seda kasutada ja kohandada oma vajadustele.

OpenCV' l on üle 2500 optimeeritud algoritmi. Nende algoritmide abil on võimalik tuvastada nägusid, objekte, klassifitseerida inimeste liigutusi, jälgida kaamera liikumist, luua piltidelt 3D objekte ja palju muud. [15]

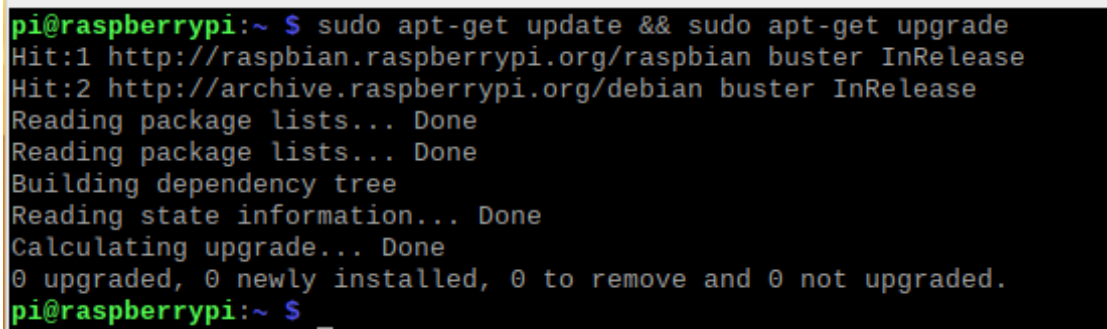
Antud projektis on OpenCV kasutusel, et tuvastada markereid ja nende järgi liikuda

## 4.5. OpenCV installatsioon

OpenCV installimisel tuleb esmalt kontrollida kas süsteem on uuendatud.

Selleks kasutame käsklust:

```
sudo apt-get update && sudo apt-get upgrade
```



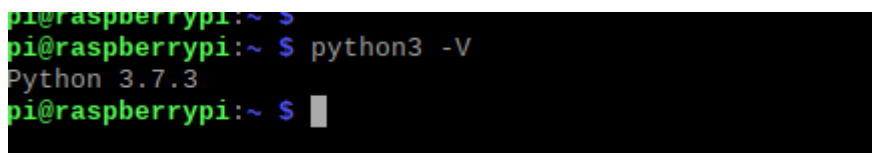
```
pi@raspberrypi:~ $ sudo apt-get update && sudo apt-get upgrade
Hit:1 http://raspbian.raspberrypi.org/raspbian buster InRelease
Hit:2 http://archive.raspberrypi.org/debian buster InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $
```

Joonis 4.3 OpenCV installatsioon 1

Pildilt (vt Joonis 4.3) on näha, et antud süsteem on juba uuendatud.

Järgmiseks tuleks kontrollida kas Python 3 versioon on olemas käsuga:

```
python3 -V
```



```
pi@raspberrypi:~ $ python3 -V
Python 3.7.3
pi@raspberrypi:~ $
```

Joonis 4.4 OpenCV installatsioon 2

Pildilt (vt Joonis 4.4) on näha, et süsteemis on olemas Python version 3.7.3

Järgmiseks tuleb luua virtuaalne keskkond OpenCV jaoks, sest see võib vastasel juhul segada süsteemi normaalset toimimist.

Selleks tuleb installeerida virtuaalkeskonna pakett, teha kaust ning aktiveerida virtuaalkeskkond käskudega:

```
sudo apt-get install python3-pip python3-virtualenv
```

```
mkdir project
```

```
cd project
```

```
python3 -m pip install virtualenv
```

```
python3 -m virtualenv env
```

```
source env/bin/activate
```

```
pi@raspberrypi:~ $ mkdir project
pi@raspberrypi:~ $ cd project
pi@raspberrypi:~/project $ python3 -m pip install virtualenv
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: virtualenv in /usr/lib/python3/dist-packages (15.1.0)
pi@raspberrypi:~/project $ python3 -m virtualenv env
Running virtualenv with interpreter /usr/bin/python2
New python executable in /home/pi/project/env/bin/python2
Also creating executable in /home/pi/project/env/bin/python
Installing setuptools, pip, wheel...done.
pi@raspberrypi:~/project $ source env/bin/activate
(env) pi@raspberrypi:~/project $
```

Joonis 4.5 OpenCV installatsioon 3

Pildil (vt Joonis 4.5) on näha loodud kaust nimega project, sinna navigeeritud, installitud ja loodud virtuaalkeskond nimega env, ja see aktiveeritud.

Järgmiseks on vaja hulk erinevaid pakette OpenCV jaoks.

Need saab installeerida ühe pika käsuna:

```
sudo apt install -y build-essential cmake pkg-config libjpeg-dev libtiff5-dev libpng-dev
libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev
libfontconfig1-dev libcairo2-dev libgdk-pixbuf2.0-dev libpango1.0-dev libgtk2.0-dev
libgtk-3-dev libatlas-base-dev gfortran libhdf5-dev libhdf5-serial-dev libhdf5-103
libqt5gui5 libqt5webkit5 libqt5test5 python3-pyqt5 python3-dev
```

```
Selecting previously unselected package automake.
Preparing to unpack .../010-automake_1%3a1.16.1-4_all.deb ...
Unpacking automake (1:1.16.1-4) ...
Selecting previously unselected package autopoint.
Preparing to unpack .../011-autopoint_0.19.8.1-9_all.deb ...
Unpacking autopoint (0.19.8.1-9) ...
Selecting previously unselected package cmake-data.
Preparing to unpack .../012-cmake-data_3.16.3-3~bpo10+1_all.deb ...
Unpacking cmake-data (3.16.3-3~bpo10+1) ...
Selecting previously unselected package libjsoncpp1:armhf.
Preparing to unpack .../013-libjsoncpp1_1.7.4-3_armhf.deb ...
Unpacking libjsoncpp1:armhf (1.7.4-3) ...
Selecting previously unselected package librhash0:armhf.
Preparing to unpack .../014-librhash0_1.3.8-1_armhf.deb ...
Unpacking librhash0:armhf (1.3.8-1) ...
Selecting previously unselected package libuv1:armhf.
Preparing to unpack .../015-libuv1_1.24.1-1+deb10u1_armhf.deb ...
Unpacking libuv1:armhf (1.24.1-1+deb10u1) ...
Selecting previously unselected package cmake.
Preparing to unpack .../016-cmake_3.16.3-3~bpo10+1_armhf.deb ...
Unpacking cmake (3.16.3-3~bpo10+1) ...
Progress: [ 5%] [##.....]
```

Joonis 4.6 OpenCV installatsioon 4

Pildilt (vt Joonis 4.6) on näha pakettide installatsioon.

See võtab umbes 2 minutit aega.

Kuna antud robot kasutab Raspberry Pi kaamerat, siis tuleb installeerida vastav pakett:

```
pip3 install "picamera[array]"
```

viimaks saab installeerida OpenCV käsuga:

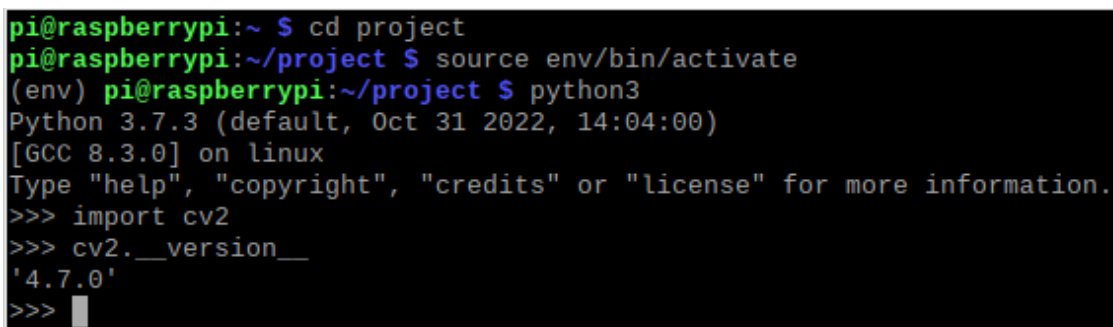
```
pip3 install opencv-python
```

Nüüd saab kontrollida kas kõik läks hästi.

```
python3
```

```
import cv2
```

```
cv2.__version__
```



```
pi@raspberrypi:~ $ cd project
pi@raspberrypi:~/project $ source env/bin/activate
(env) pi@raspberrypi:~/project $ python3
Python 3.7.3 (default, Oct 31 2022, 14:04:00)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.7.0'
>>>
```

Joonis 4.7 OpenCV installatsioon 5

Pildilt (vt Joonis 4.7) on näha, et installatsioon õnnestus ja süsteemis on OpenCV versioon 4.7.0. [16][17]

## 4.6. Ultraheli sensorid

Ultraheli sensoreid kasutatakse kauguste tuvastamiseks. Need töötavad kajalokatsiooni põhjal, üks pool saadab välja ultraheli signaali ja teine pool võtab selle vastu, ajavahemikku mis saatmise ja vastuvõtmise vahele jääb kasutatakse kauguse mõõtmiseks.

Robot kasutab nelja HC-SR04 ultraheli sensorit, et tuvastada tee peal olevaid takistusi. Sensorid on valitud nende laialdase kasutuse ja taskukohase hinna tõttu.

Sensori miinimum mõõdetav distants on 2 sentimeetrit ja maksimum mõõdetav distants on 4 meetrit. Mõõdetava ala laius on 15 kraadi.

Distants leitakse järgneva valemi järgi.

$$\text{Distants} = (\text{aeg} * \text{heli kiirus}) / 2$$

[18]

## 4.7. ThunderBorg mootori kontrolleri seadistamine

ThunderBorg kontrolleri tarkvara saab alla laadida ja installida käsklusega:

```
bash <(curl https://www.piborg.org/installer/install-thunderborg.txt)
```

alternatiivselt saab seda teha manuaalselt:

```
mkdir ~/thunderborg
```

```
cd ~/thunderborg
```

```
wget http://www.piborg.org/downloads/thunderborg/examples.zip
```

```
unzip examples.zip
```

```
chmod +x install.sh
```

```
./install.sh
```

Selleks, et mitut kontrolleri kasutada saaks on vaja igale kontrolleri määrata aadress kolmest sajaüheksateistkümnendi, selleks on vaja Raspberry Pi ühendada üksikshaaval kontrolleri ja teha käsklused:

```
cd ~/thunderborg
```

```
python
```

```
import ThunderBorg
```

```
ThunderBorg.SetNewAddress(11)
```

Esimesele kontrolleri on nüüd määratud aadress 11.

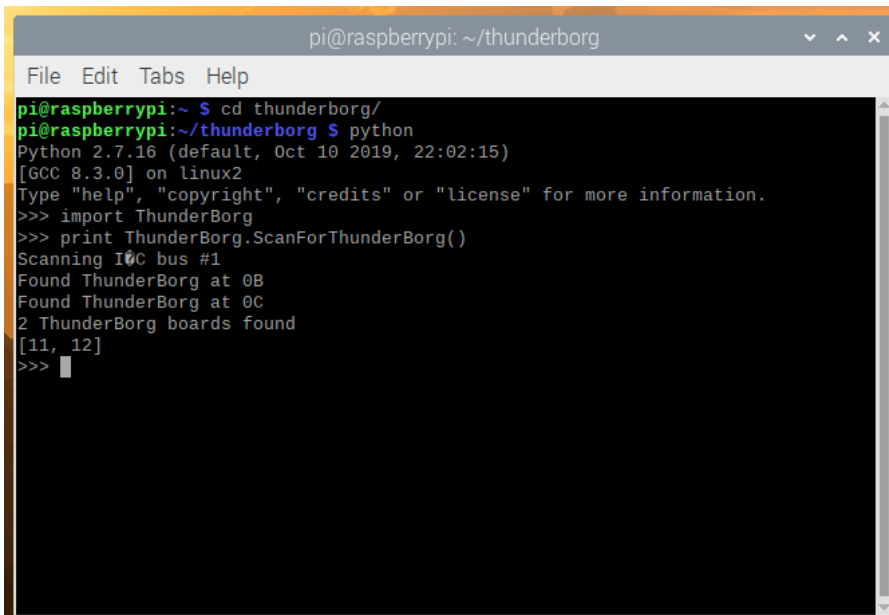
Kordame protsessi teise kontrolleri, aga viimane rida on:

```
ThunderBorg.SetNewAddress(12)
```

Nüüd on aadressid antud ja kui kõik kontrolleri on ühendatud, saab anda käskluse:

```
print ThunderBorg.ScanForThunderBorg()
```

Eduka seadistamise korral prinditakse välja kõik määratud aadressid.



```
pi@raspberrypi: ~/thunderborg
File Edit Tabs Help
pi@raspberrypi:~$ cd thunderborg/
pi@raspberrypi:~/thunderborg$ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import ThunderBorg
>>> print ThunderBorg.ScanForThunderBorg()
Scanning I2C bus #1
Found ThunderBorg at 0B
Found ThunderBorg at 0C
2 ThunderBorg boards found
[11, 12]
>>>
```

Joonis 4.8 ThunderBorg aadressid

Pildilt (vt Joonis 4.8) on näha, et on leitud kaks ThunderBorg kontrollereid aadressidega 11 ja 12, seega seadistamine õnnestus.[19]

ThunderBorg tarkvara kasutab Python 2, aga OpenCV kasutab Python 3, mis tähendab, et need ei tööta samas koodis. Selleks on saadaval ThunderBorg3 pakett. Selle saab alla laadida käsklusega:

```
cd ~/thunderborg
```

```
wget -O ThunderBorg3.py
```

```
http://forum.piborg.org/downloads/thunderborg/ThunderBorg3.py.txt
```

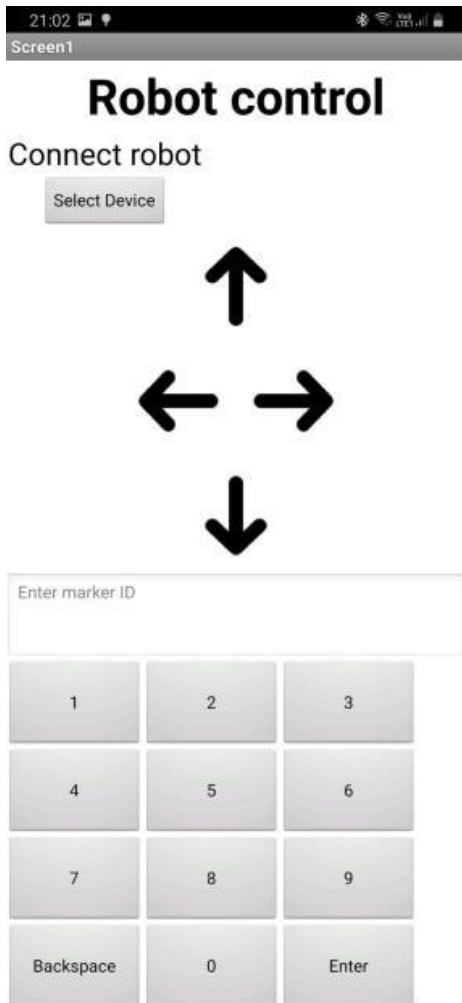
Ning koodides tuleb importida õige versioon:

```
import ThunderBorg3 as ThunderBorg
```

[20]

## 5. ÄPP

Äpp roboti juhtimise äpp (vt Joonis 5.1) on tehtud MIT App Inventor veebikeskkonnas ning võimaldab robotile üle Bluetooth ühenduse saata marker ID'd ning robotit juhtida manuaalselt. Rakenduse programmeerimine on näha lisade all (vt Lisa 4).



Joonis 5.1 Kuvatõmmis äpist

Pildi peal (vt Joonis 5.1) on näha nupp "Select Device", millele vajutades avaneb nimekiri telefoniga seotud Bluetooth seadetest. Menüüst valides robotiga ühendatud HC-05 bluetooth mooduli saab robotiga ühendus loodud ning seejärel saab äpist robotit juhtida. Lõputöö esitamise hetkeks ei ole Raspberry Pi 3 ja Raspberry Pi Pico vahel edukat ühendust loodud ning seetõttu ei ole veel võimalik äpist robotit kontrollida. Siiski on võimalik veenduda äpi toimimises läbi Raspberry Pi Pico.

```
1 from machine import Pin,PWM,UART
2 import time #importing time

Shell x
>>> %Run -c $EDITOR_CONTENT
b'right'
b'stop'
b'left'
b'stop'
b'12'
b'38'
b'right'
b'stop'
b'left'
b'stop'
```

Joonis 5.2 Raspberry Pico väljastatud andmed

Pildi (vt Joonis 5.2) pealt on näha Raspberry Pi Pico poolt väljastatud andmeid mis on saadetud äpist. Peale iga nupu lahti laskmist saadetakse käsklus "stop", mis roboti seisma paneb, seetõttu liigub robot ainult nuppu all hoides. Pildilt on samuti näha äpist saadetud numbrid mida robot hakkab kasutama sihtpunktidena.

### 5.1.1. Puldi ja/või äpi juhtkood

Käesoleva töö esitamise hetkeks on robotit võimalik juhtida Bluetooth puldiga. Tulevikus saab robotit kontrollida Bluetooth äpiga ja saata ka robotile käsk otsida kindlat markerit või jälitada kindlat markerit.

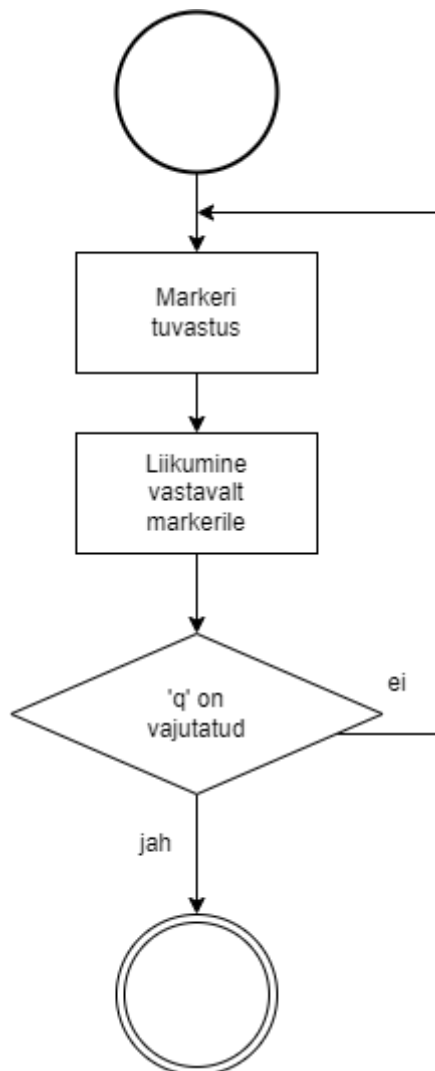
### 5.1.2. Navigatsioonikood

Robot navigeerib oma keskkonnas markerite järgi, millel on oma kindel id number. Robot suudab hinnata markeri distantssi selle suuruse järgi ja kaamera pildilt määrata markeri suuna ning selle järgi kohandada sõidusuunda.

Süsteem on ehitatud nii, et markerid ID'dega 10, 20, 30 jne tähistavad riulivaheid laokeskkonnas ja vahepealsed ID'd tähistavad vahepeatuseid riulikoridorides.

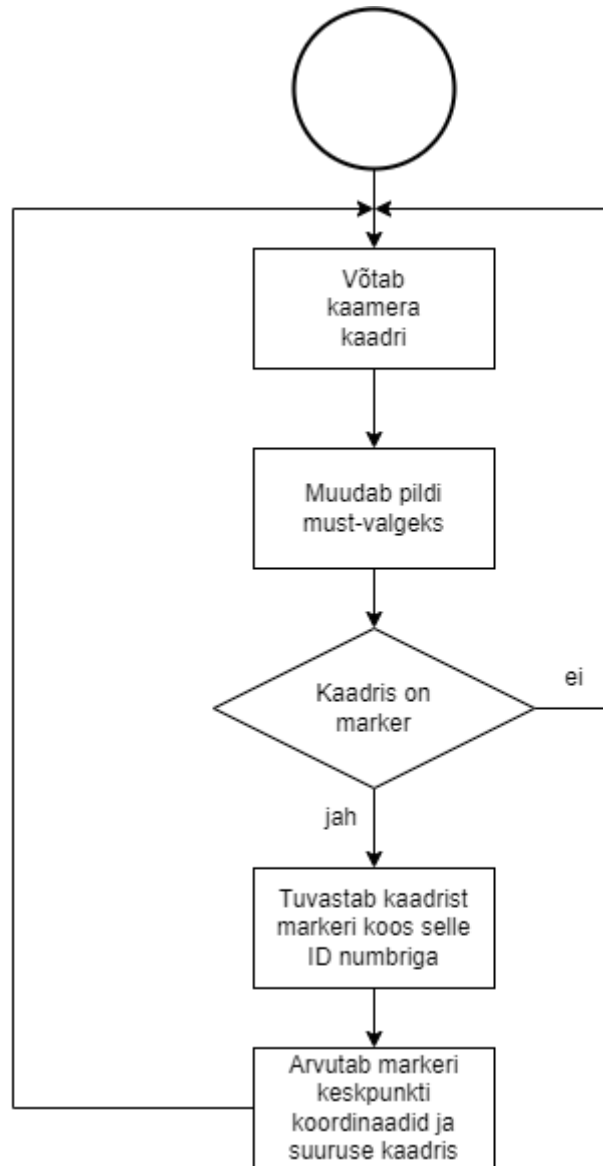


Joonisel (vt Joonis 5.3) on näha roboti töötamise põhikoodi diagramm, kus on kujutatud igavene ringlus kuni klahv Q on vajutatud. Esmalt tuvastatakse marker ning seejärel liigutakse vastavalt markerile.



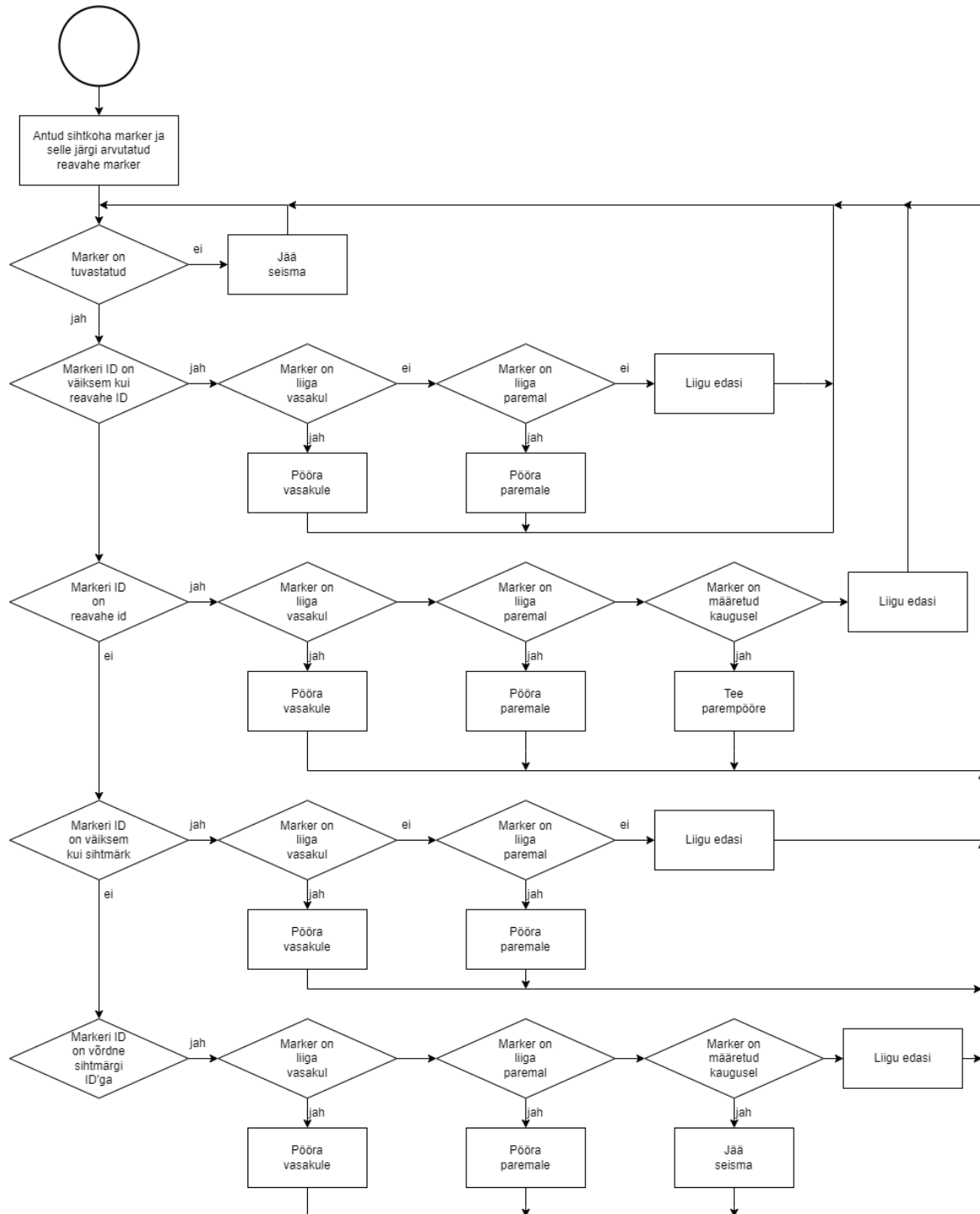
Joonis 5.3 Põhikoodi diagramm

Joonisel (vt Joonis 5.4)(vt Lisa 5) on näha roboti markerituvastuse algoritm. Süsteem võtab kaamera kaadri ja muudab selle *grayscale*’i. seejärel kontrollitakse kas kaadris on marker. Kui marker leitakse, siis tehakse kindlaks selle nurgad ja identifikaator. Järgmiseks leitakse markeri keskpunkti koordinaadid ning markeri suurus kaadris.



Joonis 5.4 Roboti markerituvastuse algoritm

Joonisel (vt Joonis 5.5)(vt Lisa 6) on näha roboti liikumisloogika markerite põhjal. Robot võtab sisestatud markeri ID ja kasutab valemit, et teada saada millises riulivahe asub sisestatud marker, sejärel suundub robot konkreetse riulivahe juurde ning keerab riulikoridori poole ja sõidab niikaua kuni jõuab otsitava markeri juurde.



Joonis 5.5 Roboti liikumisloogika

Tulevikus võtab robot ka arvesse takistuste kaugust ultraheli sensorite järgi ning kohandab sõidusuunda ka nende järgi või jääb seisma kui takistus ette tuleb.

## 6. KOMPONENTIDE MAKSUMUS

Allolevas tabelis (vt Tabel 5.1) on välja toodud roboti ehitamiseks ostetud materjalid ja komponendid.

Tabelis ei ole DC/DC step-down mooduleid ega akut, sest need on laenatud.

Tabel 5.1 Roboti komponentide maksumus

<b>Komponent</b>	<b>Maksumus</b>	<b>Kogus</b>	<b>Kokku</b>
<i>Raami materjal</i>	20 EUR		20 EUR
<i>Raspberry Pi 3</i>	60 EUR	1	60 EUR
<i>Raspberry Pi Pico</i>	7 EUR	1	7 EUR
<i>Kaamera</i>	25 EUR	1	25 EUR
<i>HC-05 bluetooth moodul</i>	16 EUR	1	16 EUR
<i>ThunderBorg kontrollor</i>	43 EUR	2	86 EUR
<i>Mootorid</i>	23 EUR	4	92 EUR
<i>Step-Down moodul 5V 2A</i>	13 EUR	1	13 EUR
<i>Mootori kinnitused</i>	5.80 EUR	4	23.2 EUR
<i>Ratta kinnitused</i>	2.90 EUR	4	11.6 EUR
<i>Rattad (paar)</i>	31.41 EUR	2	62.82 EUR
<i>Harukarp suur</i>	9 EUR	1	9 EUR
<i>Harukarp keskmine</i>	2.25 EUR	2	4.50 EUR
<i>Harukarp väike</i>	0.74 EUR	1	0.74 EUR
<i>Kaabel</i>	0.45 EUR/m	8 m	3.60 EUR
<i>Akrüülplaadid</i>	17.09 EUR	2	34.18 EUR
<i>3D printeri filament</i>	23 EUR/kg	122g	2.8 EUR
		KOKKU:	471.44 EUR

## **7. EDASISED ARENDUSED**

### **7.1. Tugevam kere**

Kuna roboti kere ei olnud piisavalt tugev, on plaanis ehitada tugevamast materjalist kere ning kinnitada rattad eraldi telgede peale ja ühendada kettülekandega mootorite külge.

### **7.2. Takistuste vältimine**

Kuigi robot suudab näha markereid ja teab mis suunas minna, on see ikka pime takistuste osas. Roboti külge on juba paigaldatud ultraheli sensorid aga need pole veel töökorras. Nende töökorda saamiseks on vaja tööle saada Raspberry Pi 3 ja Raspberry Pi Pico omavaheline edukas andmeedastus.

### **7.3. Äpp**

Selleks, et robotit oleks võimalik lihtsasti kontrollida, ilma spetsiaalse puldita, on plaanis luua äpp mis võimaldab operaatoril robotit juhtida ja sellele käsked anda.

### **7.4. Kaamera ümberpaigutamine**

Hetkel on roboti vaatenurk üsna väike, selleks, et robot suudaks oma keskkonnas paremini liikuda tuleks liigutada kaamera väljapoole Raspberry Pi 3 karpi ja panna see servo otsa, et robot saaks ringi vaadata.

## KOKKUVÕTE

Lõputöö tegemise käigus tutvuti suuremate robotite ehitusega, elektroonika komponentide seadistamisega, I2C protokolliga töötamisega roboti navigatsioonivõimalustega, äpi loomisega, modelleerimisega ja 3D printimisega.

Käesoleva lõputöö tulemusena valmis robot, mis suudab liigelda markerite abil laokeskkonnas ning jälitada operaatorit. Robot on iseseisev, mis tähendab, et sellel pole vaja eraldi olevat süsteemi, et opereerida ning erinevalt enamikest teistest laorobotitest suudab valminud robot liigelda ebatasase pinnase peal ning liigelda õues.

Robotit saab edasi arendada takistuste vältimise osas, lõputöö esitamise hetkeks see süsteem pole valmis aga on arenduses. Arendus annab robotile võimaluse vältida takistusi, aga ka võimaluse robotit juhtida äppist ning äppi kaudu saata robotile sihtkoha markeri ID.

Lisaks vajab edasi arendust roboti raam, kuna robot keerab endiselt vaevaliselt ja seetõttu on mõned funktsioonid ebatäpsed.

## **SUMMARY**

During the process of completing this thesis, the author familiarized themselves with the construction of larger robots, configuring electronic components, working with the I2C protocol for robot navigation capabilities, creating an application, as well as modeling and 3D printing.

As a result of this thesis, a robot was successfully developed, capable of navigating in a warehouse environment using markers and tracking an operator. The robot is self-contained, meaning it does not require a separate system to operate. Unlike most other warehouse robots, the developed robot can move on uneven surfaces and operate outdoors.

Further development can be carried out to enhance obstacle avoidance capabilities. At the time of thesis submission, this system is still under development. The development aims to provide the robot with the ability to avoid obstacles and control the robot through a mobile application, enabling the sending of destination marker IDs to the robot.

Additionally, the robot frame requires further refinement, as the robot's turning mechanism is currently cumbersome, leading to some inaccuracies in its functions.

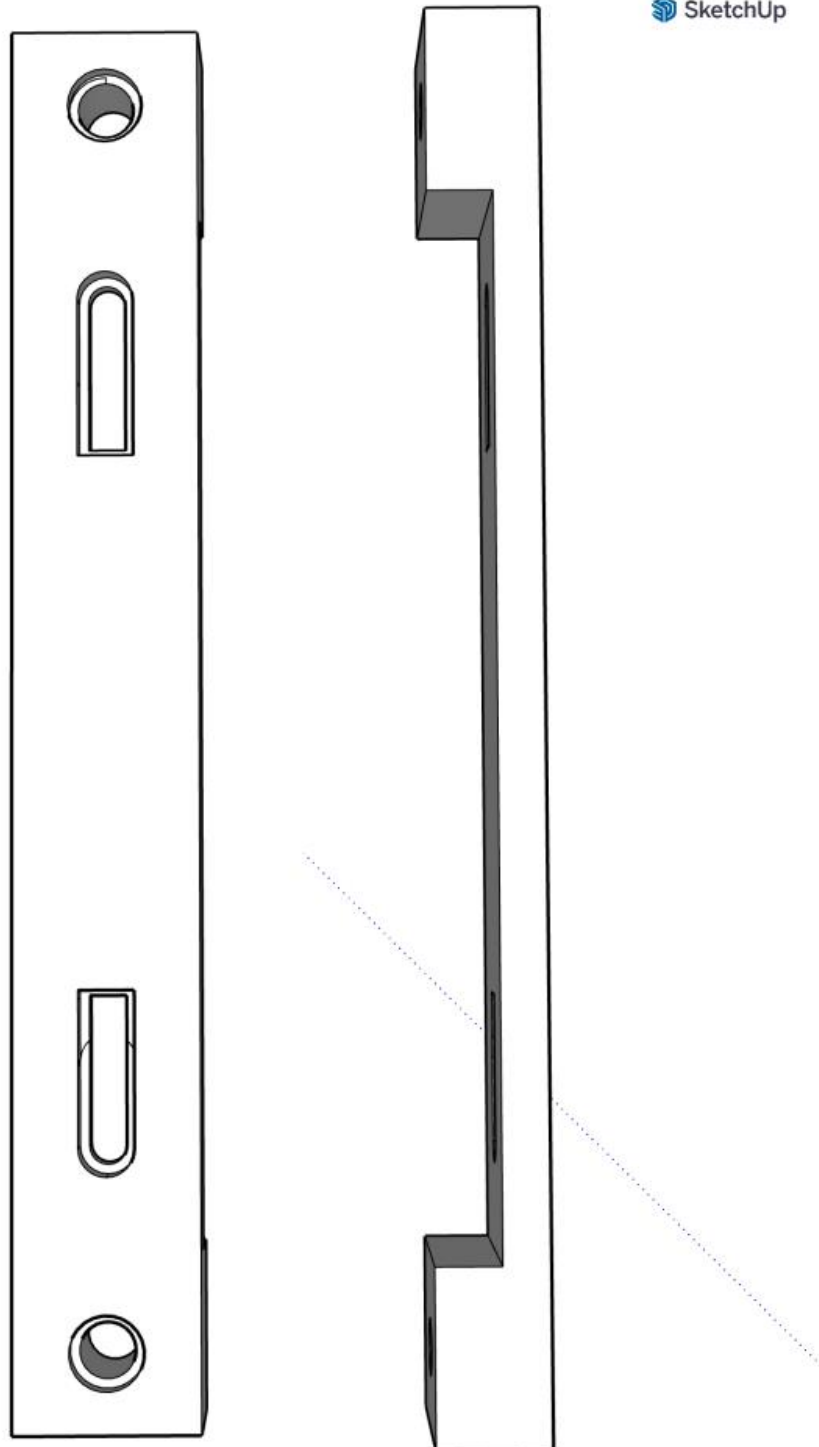


## KASUTATUD KIRJANDUSE LOETELU

1. Nasa Space Place. How does GPS Work? [Online] <https://spaceplace.nasa.gov/gps/en/#:~:text=GPS%20is%20a%20system%20of,figure%20out%20where%20you%20are> (27.09.2022).
2. Mapix Robotics. LiDAR sensors for robotic systems [Online] [https://www.mapix.com/lidar-applications/lidar-robotics/#:~:text=LiDAR%20\(Light%20Detection%20and%20Ranging,doors%2C%20people%20and%20other%20objects](https://www.mapix.com/lidar-applications/lidar-robotics/#:~:text=LiDAR%20(Light%20Detection%20and%20Ranging,doors%2C%20people%20and%20other%20objects) (11.09.2022).
3. Innovation One. line follower kit [Online] [http://cmra.rec.ri.cmu.edu/products/cortex\\_video\\_trainer/lesson/media\\_files/line\\_follower\\_ig.pdf](http://cmra.rec.ri.cmu.edu/products/cortex_video_trainer/lesson/media_files/line_follower_ig.pdf) (11.09.2022).
4. MIT, Robo-Rats Locomotion: Odometry [Online] <https://groups.csail.mit.edu/drl/courses/cs54-2001s/odometry.html> (04.04.2001).
5. Smith, Brett. A Guide to the Cameras Used in Robotics [Online] <https://www.azorobotics.com/Article.aspx?ArticleID=321#:~:text=A%20camera%20for%20robotics%20is,quality%20required%20for%20robotics%20applications> (18.12.2019).
6. Raspberry Pi. Raspberry Pi 3B+ [Online] <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/> (20.10.2022).
7. PiBorg. ThunderBorg mootori kontrollor [Online] <https://www.piborg.org/motor-control-1135/thunderborg> (20.10.2022).
8. PiBorg. 12V alalisvoolu mootorid [Online] <https://www.piborg.org/motors-mounts-and-wheels-1140/piborg-motor-12v-450rpm-42mm> (20.10.2022).
9. Raspberry Pi. Raspberry Pi. Raspberry Pi NoIR kaamera [Online] <https://www.raspberrypi.com/products/pi-noir-camera-v2/> (21.04.2023).
10. Raspberry Pi. Raspberry Pi Pico [Online] <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html> (21.04.2023).
11. Etheridge, Ian. What is I2C? [Online] <https://digilent.com/blog/what-is-i2c/> (09.01.2023).
12. Campbell, Scott. BASICS OF UART COMMUNICATION [Online] <https://www.circuitbasics.com/basics-uart-communication/> (21.04.2023).

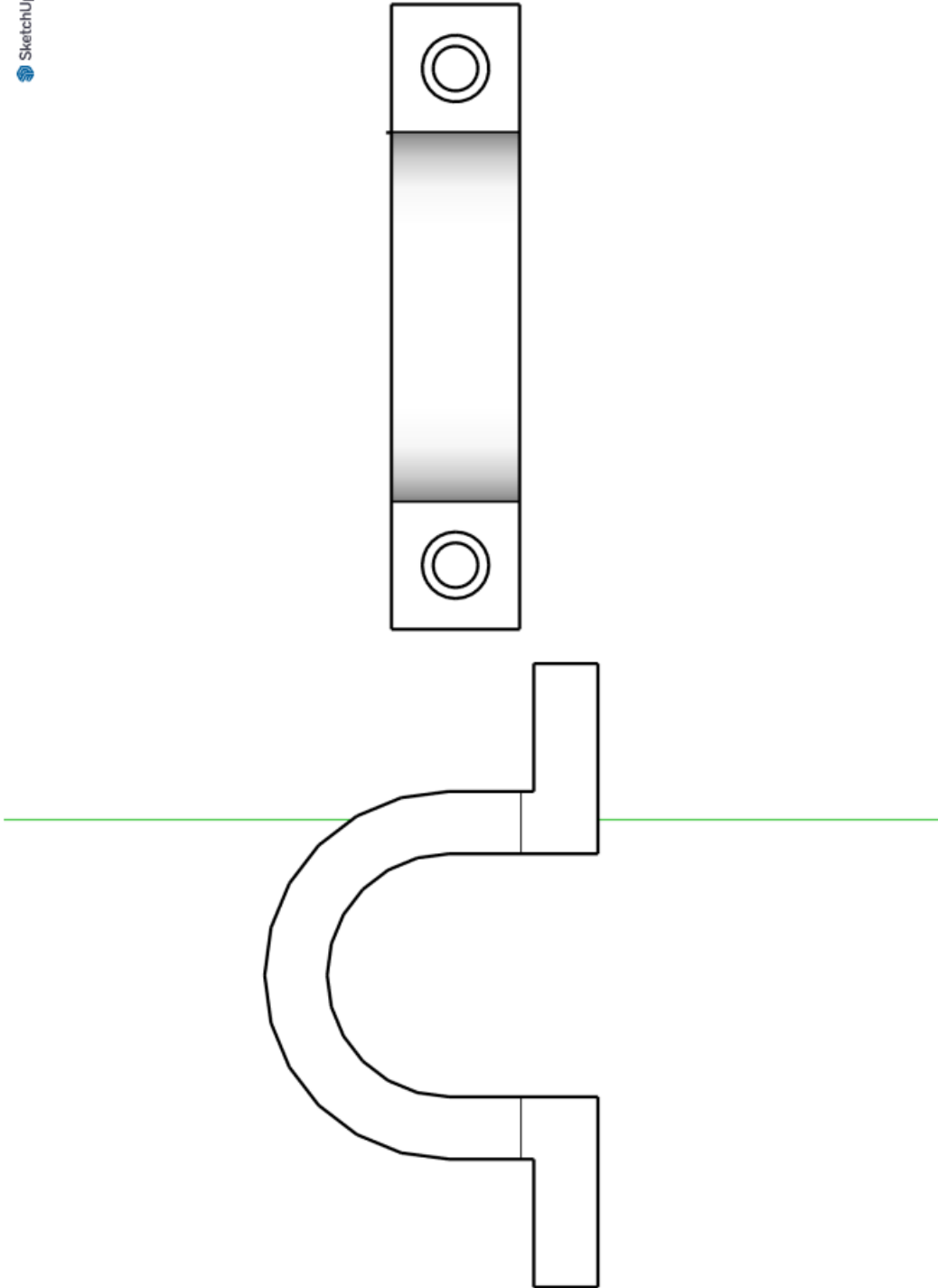
13. OpenCV. ArUco marker detection (aruco module) [Online]  
[https://docs.opencv.org/4.x/d9/d6d/tutorial\\_table\\_of\\_content\\_aruco.html](https://docs.opencv.org/4.x/d9/d6d/tutorial_table_of_content_aruco.html)  
(15.10.2022).
14. OpenCV. Detection of ArUco Markers [Online]  
[https://docs.opencv.org/4.x/d5/dae/tutorial\\_aruco\\_detection.html](https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html) (15.10.2022).
15. OpenCV, About [Online] <https://opencv.org/about/> (15.10.2022).
16. Tony G. How to install OpenCV on Raspberry Pi 4 [Online]  
<https://singleboardbytes.com/647/install-opencv-raspberry-pi-4.htm>  
(04.02.2022).
17. Sam Westby Tech. How to Install OpenCV on a Raspberry Pi [Online]  
<https://www.youtube.com/watch?v=QzVYnG-WaM4> (04.02.2022).
18. ElecFreaks. Ultrasonic Ranging Module HC - SR04 datasheet [Online]  
<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> (14.04.23).
19. PiBorg. ThunderBorg, Getting Started [Online]  
<https://www.piborg.org/blog/thunderborg-getting-started> (11.08.2022).
20. PiBorg. ThunderBorg3 [Online]  
<http://forum.piborg.org/downloads/thunderborg/ThunderBorg3.py.txt>  
(05.03.2023)

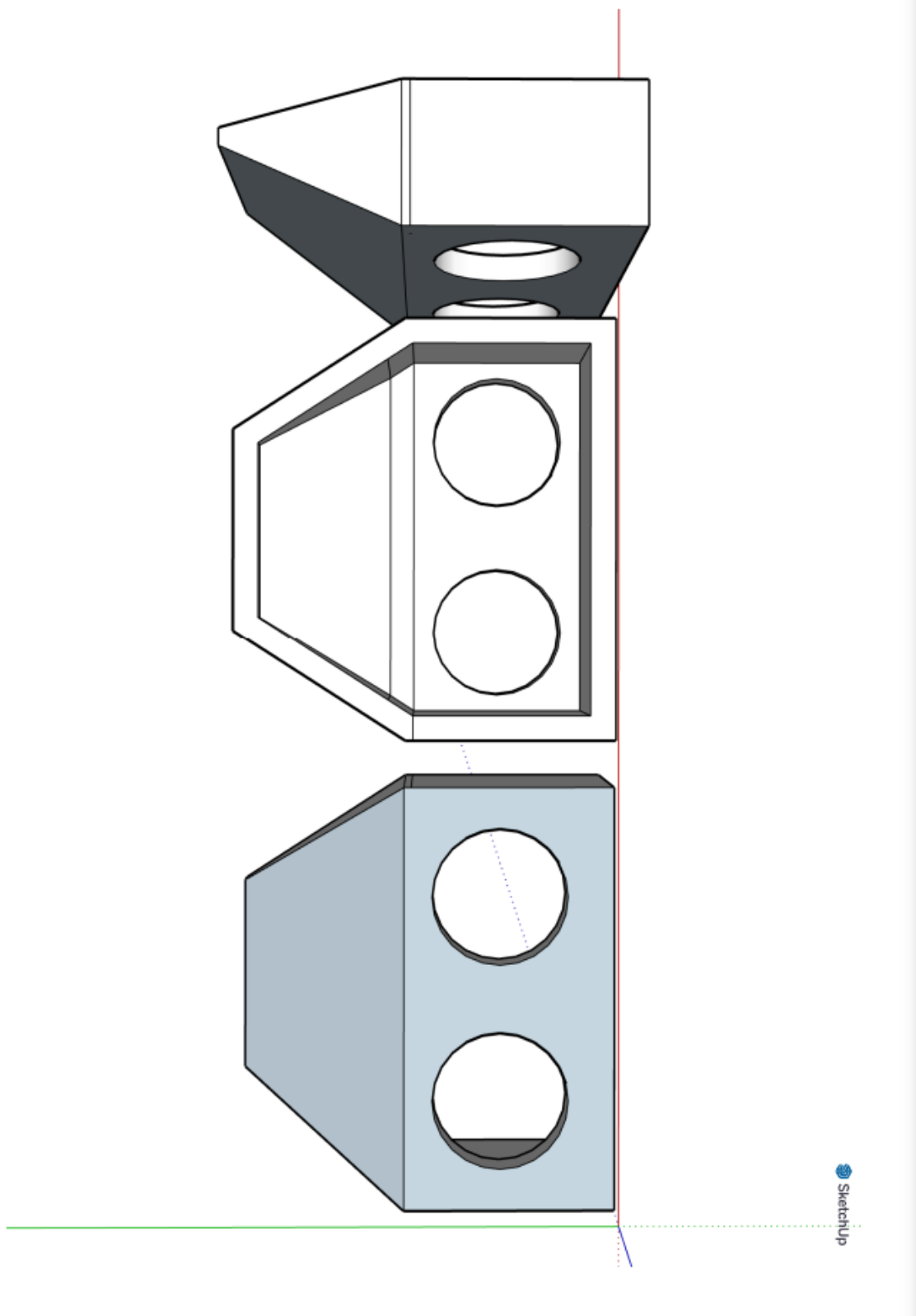
 SketchUp



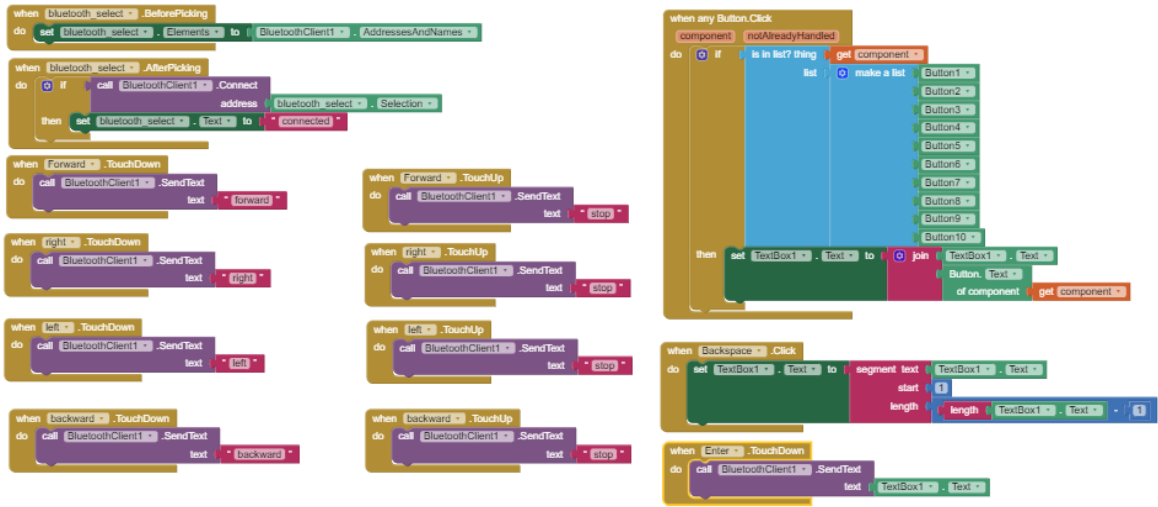
## Lisa 2 Mootori klambrid

SketchUp





## Lisa 4 MIT App Inventor kuvatömmis



## Lisa 5 Markerite tuvastamise kood

```
60 dictionary = cv.aruco.getPredefinedDictionary(cv.aruco.DICT_6X6_250)
61 parameters = cv.aruco.DetectorParameters()
62 detector = cv.aruco.ArucoDetector(dictionary, parameters)
63
64 width = 320
65 height = 240
66
67 screen_right = (width / 2) + 30
68 screen_left = (width / 2) - 30
69
70 camera = cv.VideoCapture(0)
71 camera.set(cv.CAP_PROP_FRAME_WIDTH, width)
72 camera.set(cv.CAP_PROP_FRAME_HEIGHT, height)
73
74 while True:
75     # creates an "img" var that takes in a camera frame
76     _, img = camera.read()
77     markerCorners, markerIds, rejectedCandidates = detector.detectMarkers(img)
78     # if a tag is found...
79     if markerIds is not None:
80         # for every tag in the array of detected tags...
81         for i in range(len(markerIds)):
82             print(markerIds[i])
83             # get the center point of the tag
84             center = markerCorners[i][0]
85             M = cv.moments(center)
86             cX = int(M["m10"] / M["m00"])
87             cY = int(M["m01"] / M["m00"])
88             # convert each of the (x, y)-coordinate pairs to integers
89             topRight = (int(topRight[0]), int(topRight[1]))
90             bottomRight = (int(bottomRight[0]), int(bottomRight[1]))
91             bottomLeft = (int(bottomLeft[0]), int(bottomLeft[1]))
92             topLeft = (int(topLeft[0]), int(topLeft[1]))
93
94
```

## Lisa 6 Roboti liikumise kood

```
99 for x in markerIds:
100     if x < vahekaik:
101         if x in markerIds and cX < screen_left:
102             print("slightly left")
103             print(cX)
104             slightly_left()
105         elif x in markerIds and cX > screen_right:
106             print("slightly right")
107             print(cX)
108             slightly_right()
109         else:
110             forward()
111             print(cX)
112             print('soida edasi')
113
114     elif x == vahekaik:
115
116         if distance < 60 and screen_left < cX < screen_right:
117             print("forward")
118             forward()
119         elif vahekaik in markerIds and distance < 60 and cX > screen_right:
120             slightly_right()
121             print("right")
122         elif vahekaik in markerIds and distance < 60 and cX < screen_left:
123             slightly_left()
124             print("left")
125         elif vahekaik in markerIds and 60 < distance < 100:
126
127             right()
128         elif distance > 100:
129             print("backwards")
130             backward()
131
132     elif target > x > vahekaik:
133
134         if distance < 45 and screen_left < cX < screen_right:
135             print("forward")
136             forward()
137         elif x in markerIds and distance < 45 and cX > screen_right:
138             slightly_right()
139             print("right")
140         elif x in markerIds and distance < 45 and cX < screen_left:
141             slightly_left()
142             print("left")
143         elif x in markerIds and 45 < distance < 100:
144             right()
145
146     else:
147         stop()
148
```