

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

Kristjan Harri Laur 143089IASB

MINI SUMO ROBOTI PROJEKTEERIMINE JA EHTAMINE

Bakalaurusetöö

Juhendaja: Peeter Ellervee
Ph.D

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kristjan Harri Laur

15.05.2017

BAKALAUREUSETÖÖ ÜLESANNE

Üliõpilane: Kristjan Harri Laur

Üliõpilaskood: 143089IASB

Lõputöö teema eesti keeles: Minisumo roboti projekteerimine ja ehitamine

Lõputöö teema inglise keeles: Design and Implementation of Minisumo Robot

Juhendaja (nimi, töökoht, teaduslik kraad, allkiri): Prof. Peeter Ellervee, Tallinna Tehnikaülikool, Arvutisüsteemide instituut, Ph.D.

Konsultandid:

Lahendatavad küsimused ning lähtetingimused: Roboti projekteerimine lähtudes Robotex'i nõuetest. Roboti kokkupanek ja programmikoodi kirjutamine. Roboti testimine.

Eritingimused: Mõõtmed peavad jääma pikkuse ja laiuse mõõtmetes 10cm x 10cm sisse, roboti kaal peab jääma alla 500 grammi.

Nõuded vormistamisele: Vastavalt Infotehnoloogia teaduskonnas kehtivatele nõuetele

Lõputöö esitamise tähtaeg: 22.05.2017

Ülesande vastu võtnud: _____ kuupäeva:
(lõpetaja allkiri)

Annotatsioon

Antud lõputöö eesmärgiks on mini sumo roboti projekteerimine ning ehitamine. Selle all hõlmatakse roboti projekteerimist, ehitamist, programmi koodi loomist ning roboti testimist vastavalt ürituse Robotex 2016, aasta mini sumo võistlusreeglitele.

Töö käigus kirjeldatakse reegleid, mille järgi robot disainitakse, robotite disainitüüpe, ehitust, koostisosasid, programmeerimist ja testimist ning tehakse järeldus roboti ehituse õnnestumisest.

Töö tulemusena valmis esmane versioon mini sumo robotist, mida saab mingil määral edasi arendada ja millega saab tulevastel võistlustel võistelda.

Lõputöö on kirjutatud eesti keeles keeles ning sisaldab teksti 26 leheküljel, 5 peatükki, 12 joonist, 2 tabelit.

Abstract

Design and Implementation of a Mini sumo Robot

The aim of this thesis is to design and implement a mini sumo robot. This thesis covers the ruleset that limits the design of the mini sumo robot, which is based on the tournament Robotex 2016 rules, the possible and most often used designs of the mini sumo robots, the design process, building of and programming on this particular robot as well as the testing done during the thesis.

The thesis gives a good overview on the rules that govern the design limits on mini sumo robots, as well as some of the terminology and process of a mini sumo match.

In the second chapter the author first describes the main types of designs for mini-sumo robots, based on common use of these designs and based on visual aspect and behavioral aspect of the robots design. In visual aspect the author describes the two common strategies of design, what he calls inclined and boxed robots, the first being a robot with a medium or low incline and the second with no or a really steep incline. In the behavioral aspect he describes two types of behavior for the sumo robots: offensive and counter-offensive.

In the third chapter is the description of the general build of the robot as well as the general algorithm of the program the author is using. This chapter is ended with the description of problems the author faced during the development of the machine.

In the last chapter the author puts the robot through its paces, testing it fully. The author tests the robot in multiple ways including putting the robot in a small tournament, during which a number of matches are played and some problems arise. The author describes these problems as well as some of the fixes for them. He also puts forward some ways of further development that can be done.

The design and implementation of the mini sumo robot is a success and the author concludes the thesis that the goal was reached despite the problems that arose and that there is still further development to do.

The thesis is in Estonian and contains 26 pages of text, 5 chapters, 12 figures, 2 tables.

Lühendite ja mõistete sõnastik

USB	<i>Universal Serial Bus</i> , Universaalne järjestiksin
ADC	<i>Analog-Digital Converter</i> , Analoog-digitaalkonverter
IR	<i>Infrared</i> , infrapuna(-ne)
Li-Po	Liitiumpolümeer
PWM	<i>Pulse Width Modulation</i> , Pulsi-/Impulsilaiusmodulatsioon
GPIO	<i>General Purpose Input/Output</i> , Üldotstarbeline sisend/väljund
CAD	<i>Computer-aided design</i> , Raalprojekteerimine

Sisukord

Sissejuhatus	10
1 Mini sumo robot	11
1.1 Robotexi reeglid	11
1.1.1 Mini sumo roboti füüsilisi mõõtmeid piiravad reeglid.....	11
1.1.2 Võistlusväljak (Dohyo Jyonai)	12
1.1.3 Matši käik	14
2 Projekteerimine.....	15
2.1 Robotite erinevad disainivõimalused.....	16
2.1.1 Kaldpinnaga robotid	16
2.1.2 Karprobotid.....	17
2.1.3 Käitumuslik liigitus	17
2.2 Valitud roboti disain	18
2.2.1 Vajalikud elektroonsed komponendid	20
2.2.2 Mitteelektroonsed komponendid	23
3 Roboti ehitamine ja programmi koodi kirjutamine	26
3.1 Roboti kokkupanek.....	26
3.2 Programmi kood	28
3.3 Ehitusel ja programmeerimisel tekkinud probleemid.....	32
4 Roboti testimine.....	33
4.1 Võistlusolukord	33
4.2 Edasiarendus	34
5 Kokkuvõte	36
Kasutatud allikad	37
Lisa 1 – Roboti kere	38
Lisa 2 – Roboti sahk	39
Lisa 3 – Roboti sisemus pealtvaates	40
Lisa 4 – Roboti skaneerimise koodid	41

Jooniste loetelu

Joonis 1: Võistlusväljak. [4]	13
Joonis 2: Mini sumo robot.	18
Joonis 3: Minisumo sisemus eestvaates	19
Joonis 4: Roboti sisemus nurga pealt vaadates.....	19
Joonis 5: Mikrokontroller Arduino Nano väljaviikude skeem. [6]	21
Joonis 6: Pololu DRV8835 mootori draiver ja selle ühendused. [7].....	22
Joonis 7: Mini sumo roboti kere.	24
Joonis 8: Mini sumo roboti sahk.	25
Joonis 9: Roboti elektriskeem.	27
Joonis 10. Valmis ja kokkupandud mini sumo robot.	28
Joonis 11: Roboti juhtimisalgoritm.	30
Joonis 12: Roboti juhtimisalgoritm jätk.	31

Tabelite loetelu

Tabel 1: Sumo klasside mõõtude ja masside piirangud. [4]	12
Tabel 2: Sumo väljakute parameetrid. [4]	12

Sissejuhatus

Antud lõputöö eesmärgiks on mini sumo roboti loomine, mis oleks kooskõlas ürituse Robotex mini sumo võistluse reeglitega ning mida saaks tulevikus kasutada antud võistluse raames. Mini sumo roboti loomiseks läbitakse roboti projekteerimise, ehitamise ning testimise etapid.

Sumo roboteid hakati esimesena looma 1980'ndatel Jaapanis, kus peeti ka esimesed võistlused. 90'ndatel ja 2000'ndatel levisid robot-sumo turniirid ka mujale maailma ning tekkisid ka erinevad sumo-klassid. Praeguseks on sumorobotite võistlused ühed suurimad maailmas, meelitades võistlema tuhandeid inimesi. [1]

Robotex on Tallinna Tehnikaülikooli ja Tartu ülikooli poolt korraldatav iga-aastane robotivõistluste üritus. Alguse on see saanud 2001. aastal ja tänaseks on see üks Euroopa suurimaid robotivõistlusi, kus peale võistluste toimuvad ka tehnoloogiaäritus, teaduskonverentsid ning töötoad. [2]

Antud üritusel toimuvad mitmed võistlused nagu sumod, mida on 2017. aastal 7 erinevat klassi¹, labürint, joonejärgimine, folkraace, veeralli ja mitmed muud. Igale võistlusele on oma reeglid, mida tuleb järgida. [3]

Töö käigus räägin üldiselt antud robotiklassiga seotud reeglitest ja matšide ning raundide käigust, üldistest ning tihedamini kasutatavatest disainivõimalustest, valitud roboti disainist ning komponentidest, millest see robot koosneb, roboti kokkupanekust ning programmeerimisest ja viimaks selle testimisest.

¹ Antud töös on kasutatud 2016 aasta Robotex reegleid. Töö tegemise ajal pole 2017 aasta reeglid veel avalikustatud.

1 Mini sumo robot

Mini sumo robotite ülesandeks on teise roboti võistlusalast välja löömine, ilma, et ise välja lükataks. Mini sumo robotitele on pandud peale piirangud, et need liiga suureks ei läheks, mis on määratud võistluste reeglites. Roboteid saab mitmete erinevate viisidega ehitada, neil on mitmeid erinevaid disaine. Robotid ise liiguvad üsnagi kiiresti, ühest väljaku otsast teise liigub ligi sekundiga, ning enamus raunde lõpevad paari sekundiga.

1.1 Robotexi reeglid

Antud robot on koostatud Robotexi reeglite järgi, mis on enamasti kooskõlas rahvusvaheliste reeglitega. Ürituse reeglid on olemas ürituse kodulehel ning ka Tallinna Tehnikaülikooli Robotiklubi kodulehel. [4]

Igale robotiklassile on omad mõõtmeid mõjutavad reeglid.

1.1.1 Mini sumo roboti füüsilisi mõõtmeid piiravad reeglid

Kõikidele sumorobotitele on määratud reeglitekohaselt füüsilisi mõõtmeid piiravad reeglid. Mini sumo puhul on nendeks 10-sentimeetrine pikkus ning laius ja kaal 500 grammi. Kõrgust antud klassil piiratud ei ole. Tuleb märkida, et sellised mõõtmed kehtivad ainult raundi stardihetkeni, ehk stardihetkel peab roboti mõõtmed kuuluma etteantud piirangutesse. Roboti mõõtmeid kontrollitakse 10 cm x 10 cm puukarbiga, roboti kaalu mõõdetakse kaaluga. [4]

Pärast stardihetke võivad roboti mõõtmed muutuda. Kuna kõrgus stardihetkel pole piiratud, siis võib teha niimoodi, et stardihetkel on külgede juures tõstetud üles sahad või muud vastase alistamisel aitavad vahendid, mis on reeglitekohased, ning mis lastakse raundi alguses alla. Antud juhul võib küll juhtuda, et, kuna robotid üsna kiiresti liiguvad, abivahendid ei jõua oma vajalikesse positsioonidesse, enne kui vastane kohale jõuab.

Antud töö puhul ei minda sellist teed pidi ning pigem üritatakse teha võimalikult madal robot. Roboti kõrguse tõstmine võib olla halb, kuna mida kõrgem on robot, seda kergem on ühel robotil leida teine üles ning välja lükata.

Tabel 1: Sumo klasside mõõtude ja masside piirangud. [4]

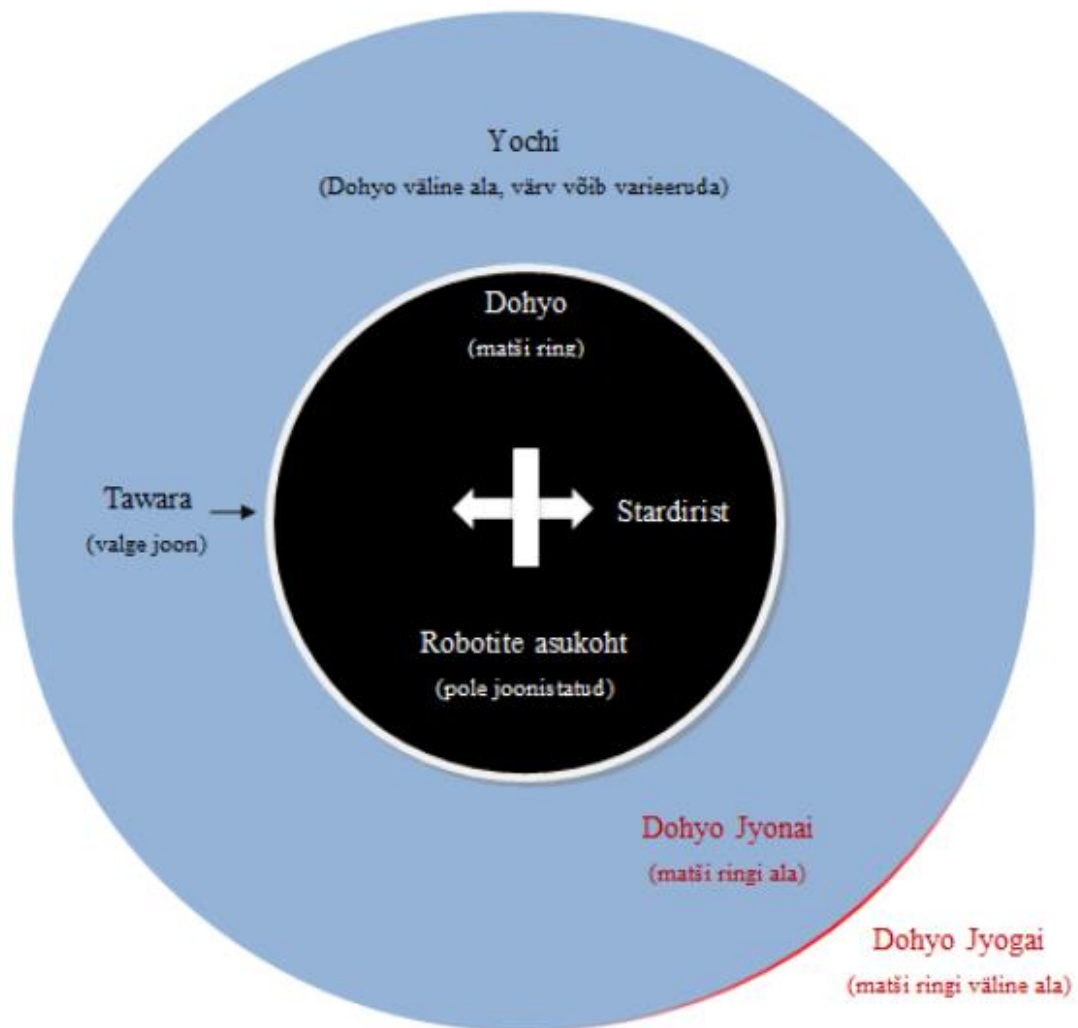
Klass	Mass	Pikkus*	Laius *	Kõrgus
3 kg Sumo	3,0 kg	20 cm	20 cm	piiramatu
Mini Sumo	0,5 kg	10 cm	10 cm	piiramatu
Mikro Sumo	0,1 kg	5 cm	5 cm	5 cm
iRobot Sumo	4,0 kg	originaalne väliskuju		piiramatu
LEGO Sumo	1,0 kg	15 cm	15 cm	piiramatu
3 kg LEGO Sumo	2,0 - 3,0 kg	20 cm	20 cm	piiramatu

1.1.2 Võistlusväljak (Dohyo Jyonai)

Tabel 2: Sumo väljakute parameetrid. [4]

Klass	Kõrgus	Diameeter	Väljaku materjal
3 kg Sumo	5 cm	154 cm	teras plastikust kattega
Mini Sumo	1 - 5 cm	77 cm	puit/plastik
Mikro Sumo	1 - 5 cm	38,5 cm	puit/plastik
iRobot Sumo	5 cm	154 cm	teras plastikust kattega
LEGO Sumo	1 - 5 cm	77 cm	puit/plastik
3 kg LEGO Sumo	5 cm	154 cm	teras plastikust kattega

Kõikide sumo robotite võistlused toimuvad kindlal võistlusväljal. Antud robotite klassi puhul on väljak diameetriga 77 sentimeetrit ja kõrgusega 1-5 mm. Lisaks on väljak tehtud puust ja plastikust, nii et näiteks magnetid on pinna ligidale hoidmiseks antud juhul kasutatud. [4]



Joonis 1: Võistlusväljak. [4]

Dohyo Jyonai, mis eesti keeli on võistlusala, koosneb kahest osast: *Dohyo*'st, ehk matši ringist, ja *Yochi*'st, ehk *Dohyo*'st väljaspoole jäävast alast. *Dohyo* on ring, mis on kaetud musta värvi kattega ja kus toimub robotite omavaheline võitlus. Ringi alla kuulub ka *Tawara*, ehk valge joon, mis ümbritseb võistlusala ning mis määrab võistlusringi ääre. *Tawara* laius on mini sumo klassis 2,5 cm. *Yochi* on mini sumo klassis vähemalt meetrise diameetriga ala ning võib koosneda mistahes värvi riidest, kuid ei tohi olla valge, viimane sellepärast, et *Tawara* on valge ja et see roboteid segadusse ei ajaks. Kui üks robotitest kukub sellele alale, on antud raund läbi. [4]

1.1.3 Matši käik

Matš koosneb tavaliselt kolmest raundist ja kestab kuni kolm minutit. Võistkond, kes saab esimesena kaks *Yuko* punkti (efektiivset punkti) matši aja jooksul, on võitja. Matši aega mõõdetakse raundide ajal, mitte nende vahel. Kui matši aja lõpuks on saadud ainult üks *Yuko* point, on võitjaks võistkond, kellel see on. Kui matši aja jooksul ei võida kumbki võistkond ühtegi raundi, selgitatakse võitja vastavalt *Yusei* (domineerimine) situatsioonile. Kui *Yusei*'d ei saa otsustada või võidetud raundide arv on mõlemal võistkonnal sama, siis pikendatakse matši kolme minuti võrra. Kui pikendatud aja jooksul läheb üks võistkond *Yuko* punktidega juhtima, siis see võistkond võidab. Võistlejatel on raundide vahel roboti hoolduseks aega maksimaalselt 30 sekundit. [4]

Matš algab vastavalt kohtuniku käsule. Võistlejad kummardavad teineteisele enne sisenemist *Dohyo Jyonai* alasse. Iga raundi eel panevad võistlejad vastavalt kohtuniku käsule robotid samaaegselt *Dohyo*'le. Robotid peavad paiknema vastastikustes sektorites, mis on näidatud vastavalt kohtuniku poolt paigutatud stardiristile ja vähemalt mõni roboti osa peab jääma valge joone peale. Roboteid ei või pärast nende asetamist liigutada. Osalejad lahkuvad *Dohyo Jyonai* alast pärast robotite asetamist. Kohtunik alustab raundi start-käsu saatmisega ametlikust infrapuna puldist. Robotid võivad alustada liikumist pärast start-käsu kättesaamist. [4]

Raundide käigus toimub robotite omavaheline võitlus ning olenevalt reeglitest antakse punkte. Raundi võitja valitakse järgmistel puhkudel:

- Kui vastane on *Dohyo*'st välja lükatud (Robot puutub *Dohyo*'st väljas olevat ala).
- Kui vastane kukub ise *Dohyo*'st välja ja puutub *Dohyo*'st väljas olevat ala.
- “*Shinitai*” olukorras. *Shinitai* on olukord, kus roboti ratas või rattad on *Dohyo*'lt väljas ja roboti ei suuda tagasi tulla väljakule.
- “*Yusei* (domineerimine)” olukorras. Antakse, kui ei selgu kindlat võitjat ja valitakse roboti liikumise, strateegia järgi.

- Kui “*Keikoku* (hoiatus)” antakse vastasele kaks korda. Kui vastane rikub reegleid mitu korda.
- Kui esineb *Hansoku* (rikkumine). Kui robot läheb katki (näiteks osa robotist tuleb lahti) või avaldatakse soovi alistuda või robotid ei puutu kokku liikumisel.
- Kui võitja selgub ilma matsita, saab võitja kaks *Yuko* punkti (kui võitjal juba on üks *Yuko* punkt, saab ta juurde ainult ühe). Kaotaja olemasolev(ad) *Yuko* punkt(id) säilivad. [4]

Matši käigus võivad ka toimuda raundi kordused. Raundi korratakse järgnevates situatsioonides:

- Mõlemad robotid on vastakuti ja liikumine on takistatud või seda ei toimu.
- Mõlemad robotid kukuvad *Dohyo*’lt korraga välja.
- Muud situatsioonid, kus võitu/kaotust pole võimalik selgitada.
- Kui võitjat ei saa selgitada pärast *Torinaoshi*’t, võib kohtunik ise paigutada robotid ja jätkata matši selleks määratud aja jooksul. [4]

Lisaks võivad matsid lõppeda ka tänu „*Sikkaku*“ ehk diskvalifitseerimine käigus, näiteks kui robot ei vasta nõuetele või osaleja käitub ebaväärikalt. [4]

Ametlikult lõpetatakse matš, kui kohtunik annab selleks käsu, vajutates kohtuniku IR puldil stopp nuppu. Seejärel peavad osalejad võtma oma roboti *Dohyo*’lt, kummardama teineteisele ja lahkuma *Dohyo Jyonai* alast. [4]

2 Projekteerimine

Antud peatükk käsitleb mini sumo robotite disaine, peamiselt enimkasutatud tüüpe, räägitakse valitud disaini projekteerimisest antud töös ning kirjutatakse selle disaini ja vajalike osade valikut ning nende tööde kohta roboti sees.

2.1 Robotite erinevad disainivõimalused

Enamus sumo roboteid koostatakse peamiselt kahe keredisaini järgi, mida kõige rohkem kasutatakse ja mis kõige edukamad siimaani on olnud. Neid kahte disaini võib liigitada nii-öelda kaldpinnaga ning karprobotid. On ka muid disaini proovitud, aga need pole eriti edukad olnud ning tavaliselt on ikka tulnud asi lõpuks kahe valiku vahele. Veel liigitatakse robotid ründe- ja vasturünde robotiteks, mida määrab just roboti käitumine ja osaliselt ka disain.

2.1.1 Kaldpinnaga robotid

Kaldpinnaga robotid on enim kasutatavad sumorobotid. Sellise roboti peamiseks ülesandeks on vastane leida ja võimalikult kiiresti vastaseni jõuda, saada selle alla ning vastane ümber või välja lükata. Selleks tehakse antud disain selline, et roboti kere oleks, vähemalt eesotsast, kaldu, mingi ligi 45-kraadise nurga all maapinna suhtes. Tavaliselt pole küll see kereosa maad puudutav, selleks on mõeldud sahk, aga antud disain aitab roboti allasaamisel. Kui robot suudab saada vastase alla, siis tähendab see tavaliselt seda, et vastase robotil pole enam korralikku kokkupuudet maapinnaga ning sellisel juhul on kergem vastase robot välja võistlusalt lükata.

Antud disaini plussiks on just, et sellega võib kergemini saada vastase roboti põhja alla, mille abil saab vastase välja lükata.

Küll aga on sellel disainil probleemiks see, et kallaku tegemine etteotsa vähendab ruumi, mida roboti sees saab kasutada. Enamus komponente mahub küll sisse, aga tihti on nii, et mõned komponendid, näiteks aku, tuleb robotist välja jätta. Teine võimalus selle parandamiseks on roboti kõrguse tõstmine ning vajalike osade mitmekihiline asetamine. Mõlemal juhul on probleemiks see, et roboti kõrgus suureneb, mistõttu on vastasel kergem robotit leida. Lisaks, esimesel juhul, on probleem veel see, et kui mingid osad on asetatud robotist välja, näiteks katusele, on võimalus, et vastase robot lööb selle komponendi küljest ära, hullemal juhul ära lõhkudes. Kuna robotite küljest ei tohi ühtegi juppi ära tulla raundi ajal, siis on see võimalus üsnagi halb ja suur riskivõtmise. Tavaliselt jäetakse, kui seda teed pidi mindakse, välja roboti aku, mis kinnitatakse mingit viisi pidi roboti katusele. Seda võib teha näiteks takjagröpsuga.

2.1.2 Karprobotid

Nii-öelda karprobotid on teine disainitüüp, mida saab kasutada, ning mida tihedamini kasutataksegi. Antud tüübil on peamiseks erisuseks eelneva tüübi suhtes see, et kaldpind on ligi 45-kraadise nurga asemel ligi 75- kuni 90-kraadise nurgaga. Antud tüübil on tavaliselt ka sahk suhteliselt väike ning selle peamiseks tööks on just see, et vastane antud roboti alla ei saaks. Tavaliselt sellist tüüpi järgides minnakse jõuga vastase ringist väljalükkamise peale, kuigi võib ka minna esmase kokkupõrke peale välja, lootes, et vastase robot põrkub ebamugavalt tagasi ning siis saab vastast välja lükata.

Antud disaini positiivsed küljeks on roboti välimise ehituse lihtsus, pole vaja mingeid suuri kaldpindu. Lisaks on antud disaini sisemuses rohkem ruumi, eriti võrreldes kaldpinnaga robotitega, kuhu saab komponente panna.

Negatiivsest poolest on raske või lausa võimatu vastase alla saada. Sellisel juhul tuleb vastast välja lükata jõuga ja kui robot on valesti ehitatud, võib see tähendada seda, et robotil pole jõudu, et seda teha.

2.1.3 Käitumuslik liigitus

Käitumusliku poole pealt on kõige enam kasutatavad nii-öelda ründerobotid. Need töötavad just põhimõttel, et vastast võimalikult kiirelt rünnata ja ta ümber või välja lükata. Sellised robotid on kiired, väikese jõumomendiga, disaini poolest väikeste, tihti laiade, ratastega ning kaldpinnaga kerega. On tehtud ka ründeroboteid, millel on suured rattad või ligi vertikaalse esiosaga, aga need pole võistlustel eriti läbi löönud.

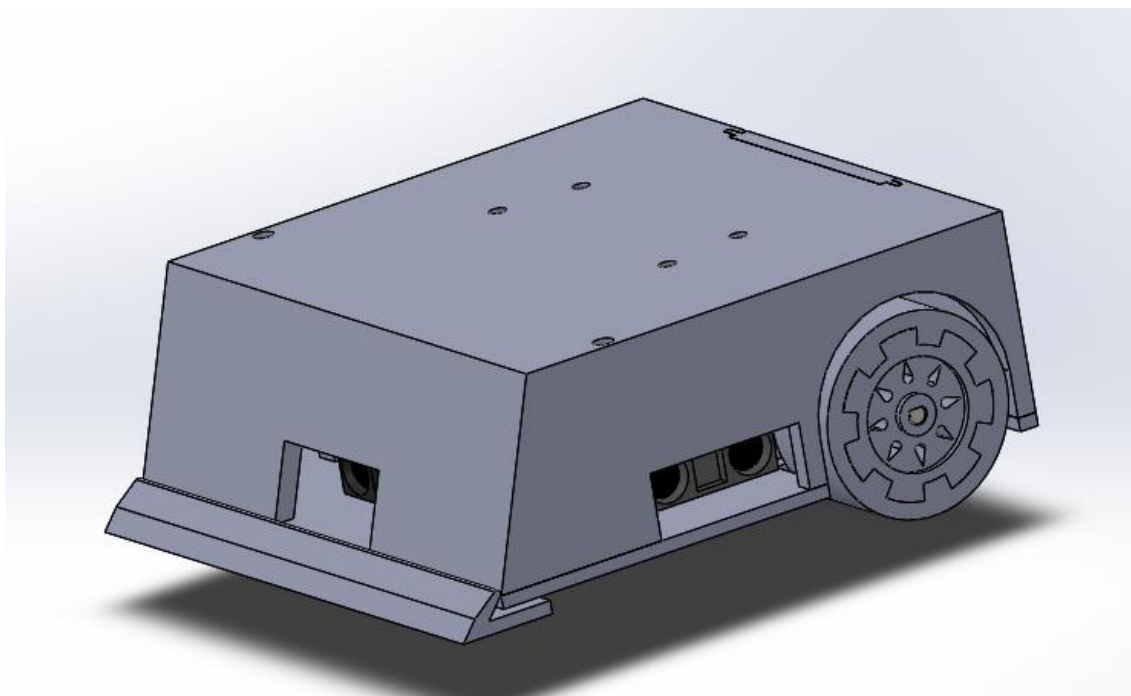
Teine tihedamini kasutatav käitumuslik viis on vasturünnakut teevad robotid. Sellised robotid on ehitatud põhimõttega, et nad suudavad vastu võtta vastase lööki ning siis kas jõuga välja lükata või, lootes, et vastane põrkab halvasti tagasi ja maandub halvasti, nurga alt välja lükata. Enamus roboteid, mis sellise käitumisega on, kasutavad just enam-vähem vertikaalset esiosa või on ehitatud just kõrgemad. Antud puhul on tüüpiliseks aeglane või keskmise kiirusega liikumine, mis on pigem suunatud suurele jõumomendile.

2.2 Valitud roboti disain

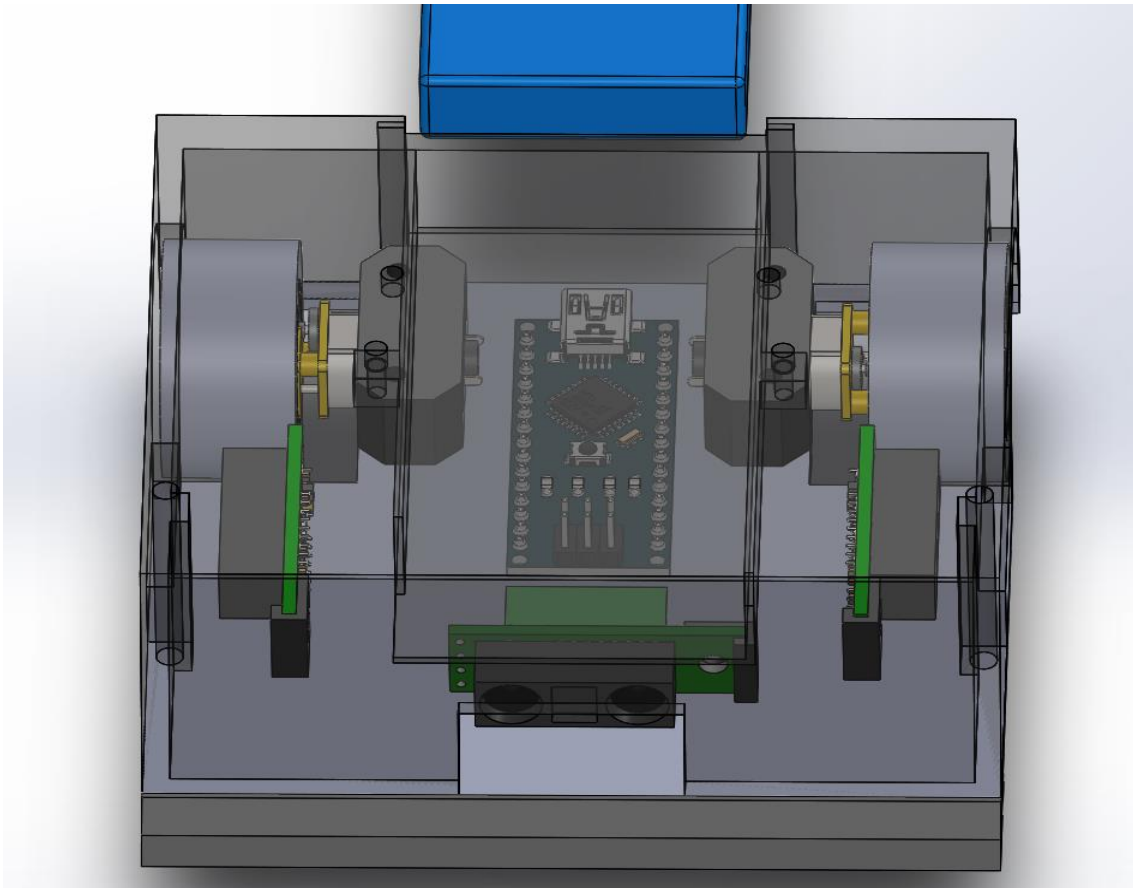
Roboti disainimisel võtsin eelduseks, et robot tuleks teha võimalikult madal. Selle saavutamiseks tuleb roboti sees teha kõik võimalikult kompaktselt ning seetõttu on see ka disainitud niimoodi, et kõik komponendid mahuksid roboti sisse. Lisaks oli soov, et kõik andurid panna võimalikult maa põhja ligidale, et oleks võimalik vastast võimalikult hästi tuvastada. Disain on väga karproboti sarnane, aga käitumusliku poole pealt tahtsin teha ründeroboti. Disainitud on see nii, et esimesel kokkupõrkel suudaks antud robot vastase ebasoodsasse positsiooni lüüa ning siis vastane välja lükata.

Isedisainitud komponentide hulka kuuluvad roboti kere, sahk, akupes, osade sensorite kinnitused, mootorite kinnitused ning rattad. Muud osad, mis on kasutuses on makettplaat, mis läheb ka kasutusele põhjana, ning mitmed elektroonsed komponendid.

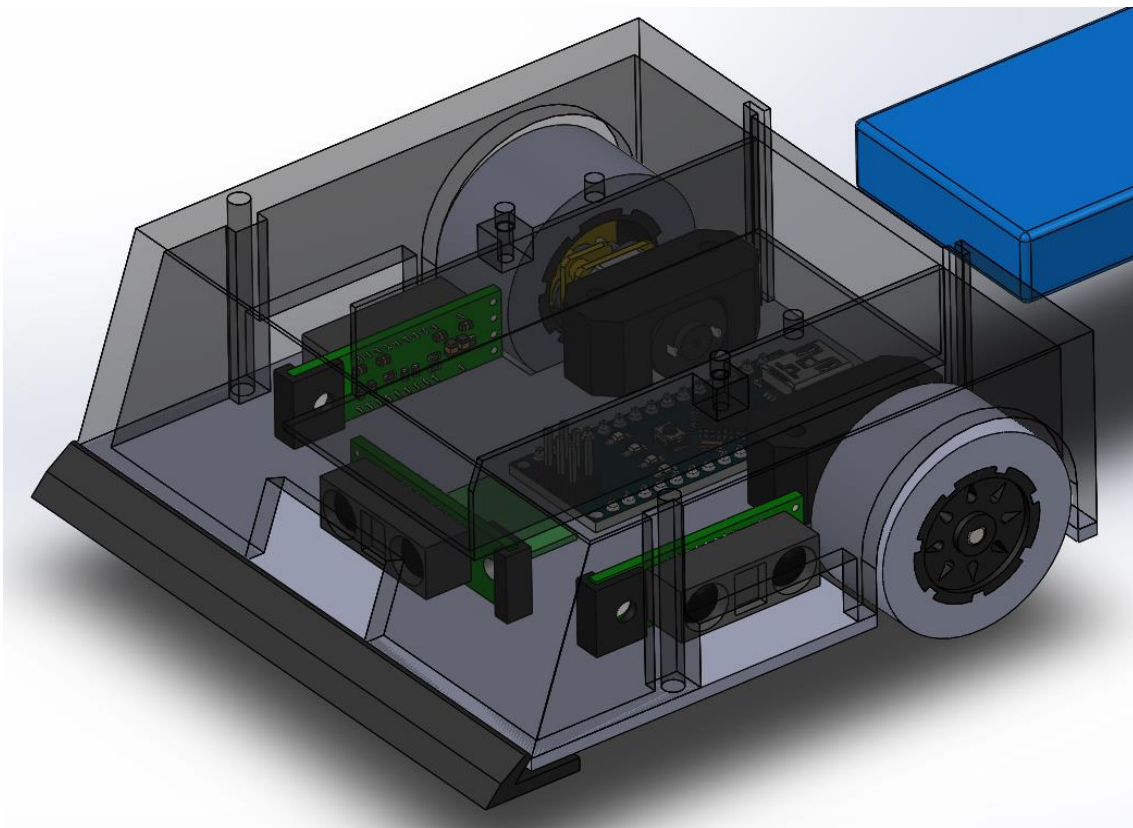
Roboti osade disainimiseks ning üldiseks roboti projekteerimiseks kasutasin 3D CAD tarkvara Solidworks 2016 Edition.



Joonis 2: Mini sumo robot.



Joonis 3: Minisumo sisemus eestvaates



Joonis 4: Roboti sisemus nurga pealt vaadates

2.2.1 Vajalikud elektroonsed komponendid

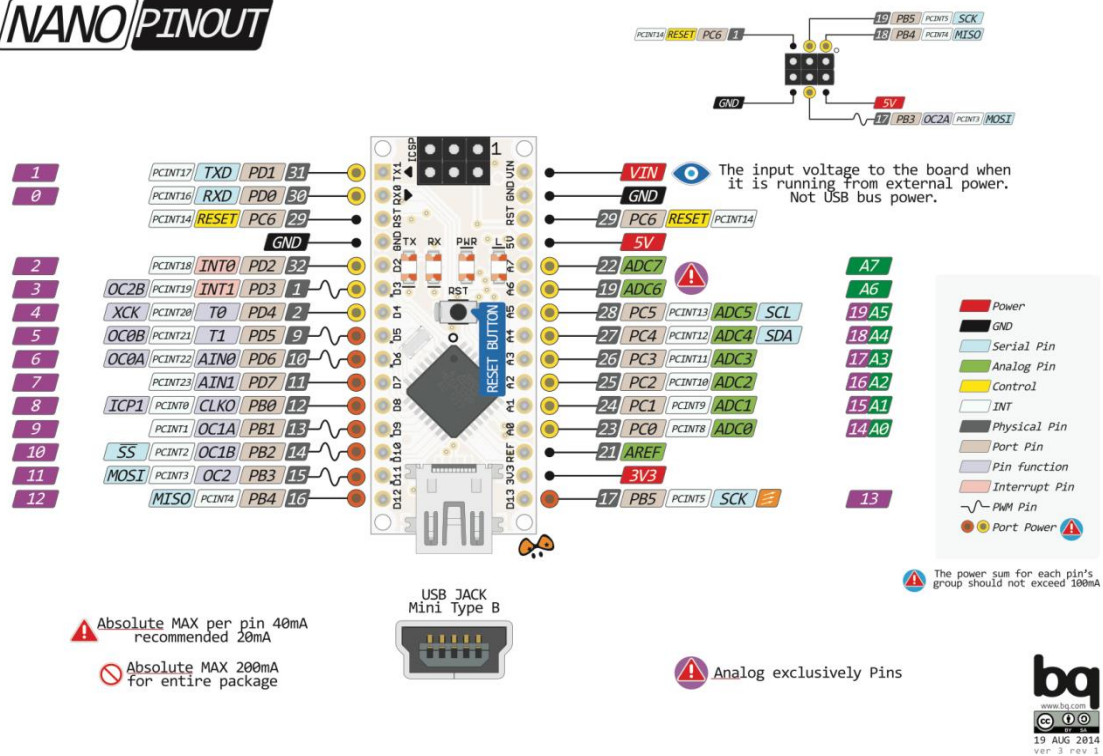
Mini sumo robotile on vaja mitmeid erinevaid komponente. Nende alla kuuluvad mikrokontroller, mootorite draiver, mootorid, jooneandurid ja infrapunaandurid ning lisaks sellele aku ning lülitid ja infrapuna vastuvõtja.

Mikrokontroller: Antud roboti jaoks on kasutuses Arduino Nano'1 põhinev mikrokontroller, mille protsessoriks on Atmega328P. Kontroller on 8-bitine kontroller, millel on 30 sisend/väljundporti pluss 6 suuremaks programmeerimiseks mõeldud porti, mille kaudu saab programeerijaga programmi peale panna. Lisaks viimasele saab ka läbi USB juhtme programmi peale laadida. [5]

Kontroller ise võimaldab välja anda nii 3,3-voldist pinget, kui ka 5-voldist pinget võttes ise sisendiks 5-voldist pinget. Kontrolleril saab ka kasutada 23 sisend-/väljundporti(ning -viiku), mis on digitaalse sisendi-/väljundiga ning millest 8 pordil on olemas ka ADC'd ehk analoog-digitaal muundurid, ehk mis suudavad muuta sisse tulevat analoogsignaali kontrollerile sobivaks digitaalsignaalsiks. Lisaks on veel 6'1 pordil võimalik väljastada PWM signaali. [5]

Igal pordil on olemas ka sisseehitatud *pull-up/pull-down* takisti, mida saab sisse lülitada, kui port muudetud sisendiks. See välistab osadel juhtudel välise resistori vajalikkuse.

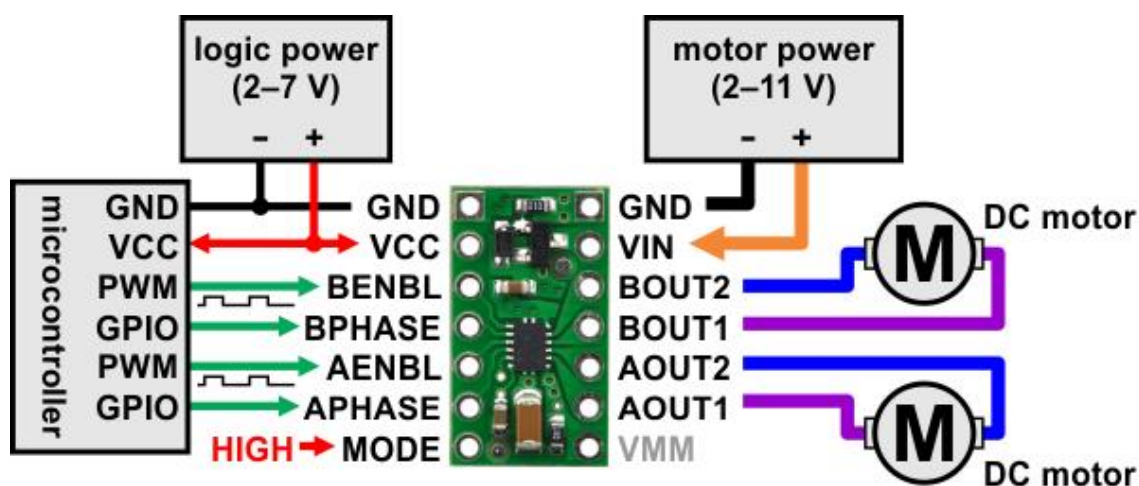
NANO PINOUT



Joonis 5: Mikrokontroller Arduino Nano väljaviikude skeem. [6]

Mootorite draiver: Mootorite draiverina kasutame Pololu DRV8835 plaati, mis on üks paremaid ja odavaimaid, mida saab antud robotites mootorite juhtimisel kasutada. Draiver on ühendatud mikrokontrolleriga ja mootoriga ning juhib mootoreid muutes kontrollerist tulnud madalvoolulise signaali sobivaks kõrgvooluliseks signaaliks mootorite juhtimiseks. Mootori draiveriga on lisaks ühendatud ka aku, kuna Arduino Nano'st tulev vool ei ole piisav mootorite liigutamiseks.

Draiveril on kokku 14 ühendusviiku, millest kasutusel on 13 viiku. 7 neist on ühendatud mikrokontrolleriga ning 2 on ühendatud akuga ja 4 mootoriga. Draiverisse tuleb mikrokontrollerist toide, PWM signaalid ning GPIO signaalid ja draiveri signaalide juhtimise signaal. Viimaseks on 5-voldine signaal, mis on ühendatud mikrokontrollerist draiverisse läbi pull-up takisti, mis hoiab ära sisendi „ujumise“ staatilise elektri ja muude häirete tõttu. Takistik on 1 kilo-oomine takisti. Ülejäänud 6 sisendit/väljundit on aku kaudu toide ning mootorid. Aku ise mootori draiverit ei toida, vaid toidab läbi draiveri mootoreid. [7]



Joonis 6: Pololu DRV8835 mootori draiver ja selle ühendused. [7]

Mootorid: Antud roboti jaoks on kasutusel 2 mootorit, milleks on Pololu kõrgpingeline 6-voldine vahelduvvoolumootor, millel on ka külge pandud redaktor. 6-voldise pingel peal suudab mootor teha 625 pööret minutis. Mootorite hammasrataste suhe on 51,45:1. [8]

Mootorite juhtimine toimub eeltoodud draiveri abil. Mootoril endal on vaja vähemalt 5-6-voldist pinget ning korraliku voolu. Kuna kontrolleri ei suuda välja anda piisavat voolu (maksimum on 200 mA) ning pinget (maksimaalne on 5 V), siis on töötamiseks vaja toiteallikast, ehk akust, läbi draiveri suuremat voolu ja pinget.

Jooneandurid: Jooneanduritena on kasutusel Pololu QTR-1A peegeldusandureid. Andur saadab välja IR kiiri ja selle all olevalt objektilt peegelduvad kiired tagasi. Olenevalt objekti värvusest ja materjalist peegeldub kiir tagasi ning anduril olevad fototransistorid tunnevad, kui palju kiirest tagasi jõudis. Seejärel saadab andur läbi oma OUT pordi analoogsignaali kontrolleri porti, kus seda saab analüüsida. Antud töös on kasutusel 2 andurit, mis paigutatakse roboti esiosas. [9]

Infrapunaandurid: Infrapunaanduriteks on kasutusel Pololu poolt loodud kandjat, millele on peale pandud Sharp GP2Y0A60SZLF analoog-kaugussensorit. Antud sensor suudab tuvastada, et objekt on selle ees kuni 150 sentimeetrini, mis on piisav mini sumo robotite jaoks, ning suudab täpselt tuvastada kaugust. Tuleb aga märkida, et miinimumkaugus, mis see suudab mõõta on 10 cm. Sellest vähem tuvastab ta küll ära, et mingi objekt on ees, aga ei suuda kindlat distantsi öelda. [10] Antud töös on kasutusel 3

sensorit ning nende väljundid on suunatud kolme kontrolleri analoogsignaali vastuvõtvasse porti. Andurid paiknevad roboti külgedel ning esiosas.

Andur saadab pidevalt välja infrapunakiiri. Kui mingi objekt ees juhtub olema, siis põrkab antud kiir tagasi ning anduris olev vastuvõtja saab selle kätte. Seejärel saadab ta välja kindla suurusega arvu, mis näitab, kaugel objekt on, mille analoogport kontrolleri kätte saab. Sealt edasi saab seda töödelda ning edasi käituda. [10]

Muud elektroonsed komponendid: Lisaks eeltoodud osadele on kasutusel ka väiksemad osad, mis ei ole eriti kasutuses matši ajal, vaid peamiselt ainult raundide alguses ja võib-olla ka lõpus. Nendeks osadeks on IR vastuvõtja(-d) ja lüliti. Infrapuna vastuvõtjad on kasutusel selleks, et raundi alguses oleks võimalik alustada ja selle lõpus peatuda, ehk saada kohtuniku puldi käest kätte stardi ning peatumissignaali. Töötavad need sagedusel 38 kHz.

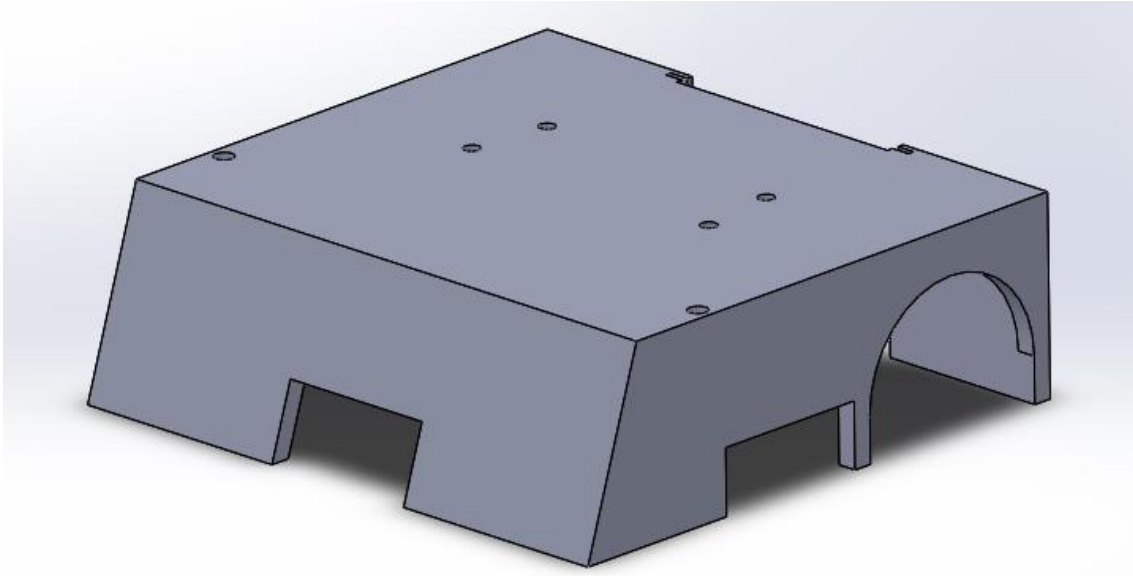
Lüliti on aga kasutusel selleks, et robot eraldi tööle panna ning, et see kogu aeg, mil toide, antud juhul aku, robotiga ühendatud on, pidevalt ei töötaks.

Lisaks nendele on veel märkimata ka kasutusel olev toide, milleks on aku. Antud robotis on kasutusel Turnigy liitiumpolümeeraku(Li-Po aku), mille maht on 1000 mAh ning mahutavus 20 C. Antud aku koosneb kahest Li-Po *cell*'st ning mis suudab välja anda 7,4 V'st pinget.

2.2.2 Mitteelektroonsed komponendid

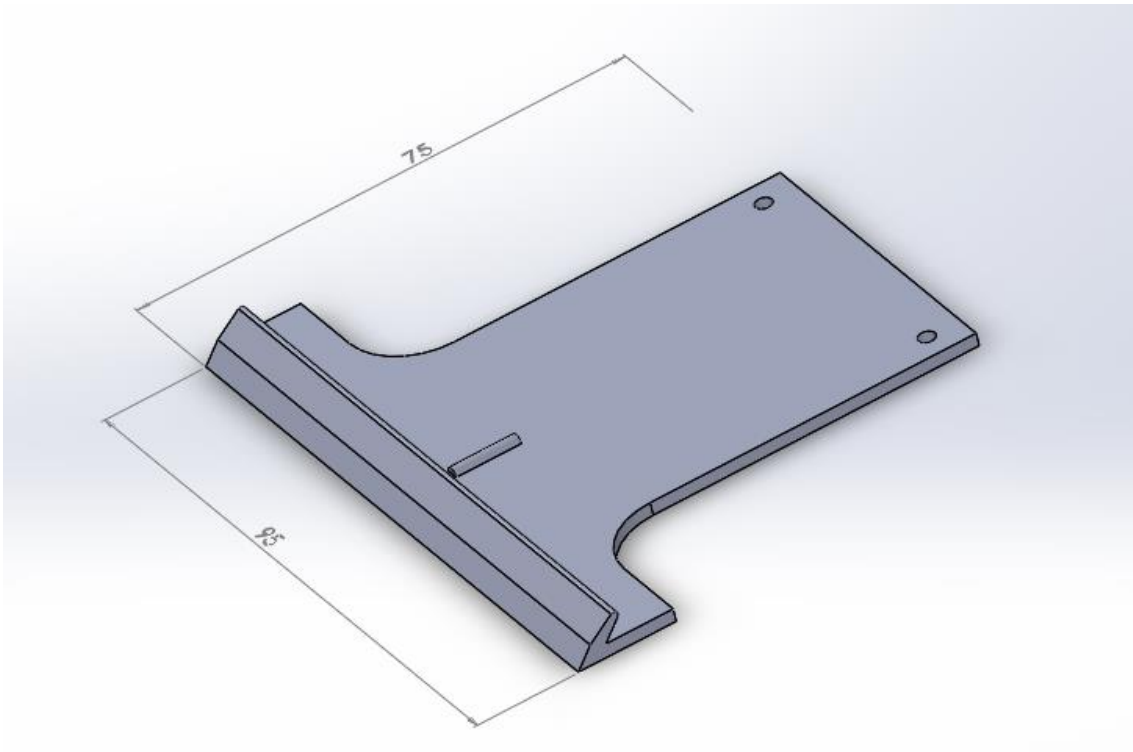
Mitteelektroonsete komponentide alla kuuluvad antud juhul suuremad komponendid nagu kere, sahk ning rattad ja rehvid ning lisaks mõned väiksemad osad. Kõik kolm peamist osa on eraldi disainitud ja 3D-printitud, välja arvatud kere esiosale minev kate, mis tuleb metallist.

Kere kirjeldus: Roboti kere on tehtud 3D-printeri abil. Kere paksus on tehtud 3 mm paksuseks ning selle tõttu on see suhteliselt korraliku tugevusega. Kere põhja laius on 95 mm ja pikkus on 90 mm. Katus on aga 80 mm pikkune ning kallak on umbes 80 kraadise nurga all võrreldes katusega. Kere tagaosas on ka luuk, mille saab ära võtta, ning mille kaudu saab vajutada lüliteid ning programmeerida mikrokontrollerit.



Joonis 7: Mini sumo roboti kere.

Sahk: Sahk on samuti tehtud 3D-printimise abil. See kinnitub roboti põhja külge ning pikendab roboti eesotsa ligi 10 mm võrra. Sahk katab osa roboti põhjast ära ja kaitseb seeläbi elektriliste komponentide ühendusi. Saha eesots läheb osaliselt kere esiotsa peale, moodustades väikese kaldpinna, mis viib eesmise kaldpinna peale. Eesmise jooneandurite blokeerimise vältimiseks sai sahk kujundatud nii, et see anduritest mööda läheks ja nende tegutsemist ei takistaks. Sahk on kerega ühendatud peitkruvi abil ning läheb läbi kere.



Joonis 8: Mini sumo roboti sahk.

Rattad ja rehvid: Roboti rattad tehakse 3D-printimise abil. Rataste diameetriks on 2,5 cm, pikkuseks 2 cm. Lisaks tuleb teha ratta sisemine osa osaliselt tühi selleks, et mootor koos reduktoriga ära mahuks. Rehvide poolest tuleb luua vorm, millesse valades tuleb kokku välja silikoonkumm, mis on roboti puhul rehviks.

Roboti rataste mõõtmed on valitud nii selleks, et robotil võimalikult hea hõõrdumine ja et robot ei libiseks ning oleks võimalikult hea jõumoment. Tavaliselt on kasutusel rehvidena silikoonrehvid, mida kasutatakse tihti selle hea vormimise võimaluse pärast ja hõõrdejõu pärast.

Muud komponendid: Muud mitteelektronsed komponendid, mis on ka kasutuses, on infrapunasensorite kinnitused, mis on vaja eraldi välja printida, kuna andurid lähevad põhimõtteliselt külje peale ning nende kinnitamiseks on vaja eraldi kinnitusi, kerele lisaks tagaplaat, mille saaks võistluste ajal kergesti ära võtta ja siis vastavalt kas akut vahetada või mikrokontrollerile programmi peale laadimiseks ühendust saada.

3 Roboti ehitamine ja programmi koodi kirjutamine

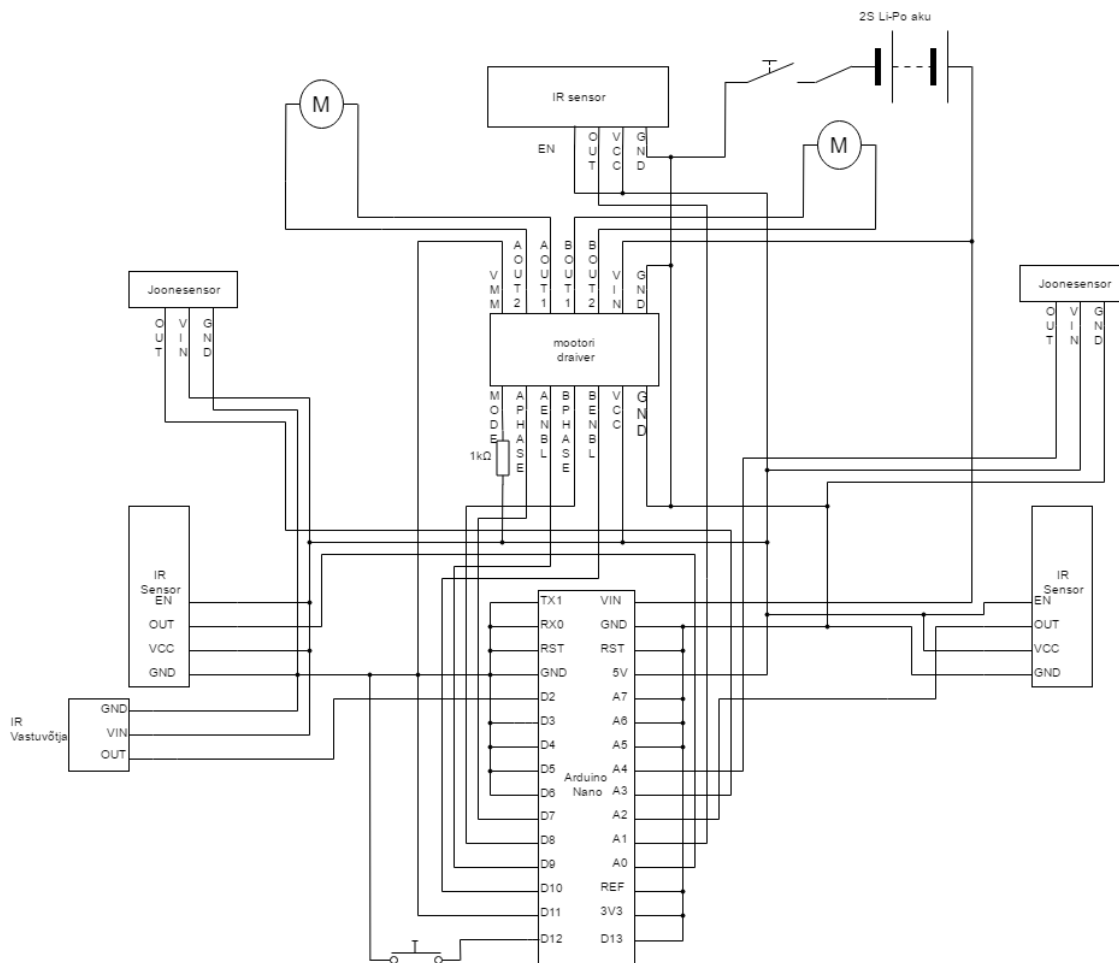
Antud peatükis käsitletakse roboti ehitamist ning programmi ning selle käigus tekkinud probleeme.

3.1 Roboti kokkupanek

Roboti kokkupanekul oli esimeseks etapiks roboti kere ja muude komponentide ühildumise kontrollimine ja vajadusel muutmine. Juba enne kokkupanekut võis arvata, et sellega võib tekkida probleeme ning need ilmsitki esimesel kokkupanekul. Nendest probleemidest võib lugeda peatükis 3.3. Pärast roboti kereosade vigade parandamist ning seejärel esimest täielikku assembleerimist sai liikuda edasi järgmiste asjade juurde.

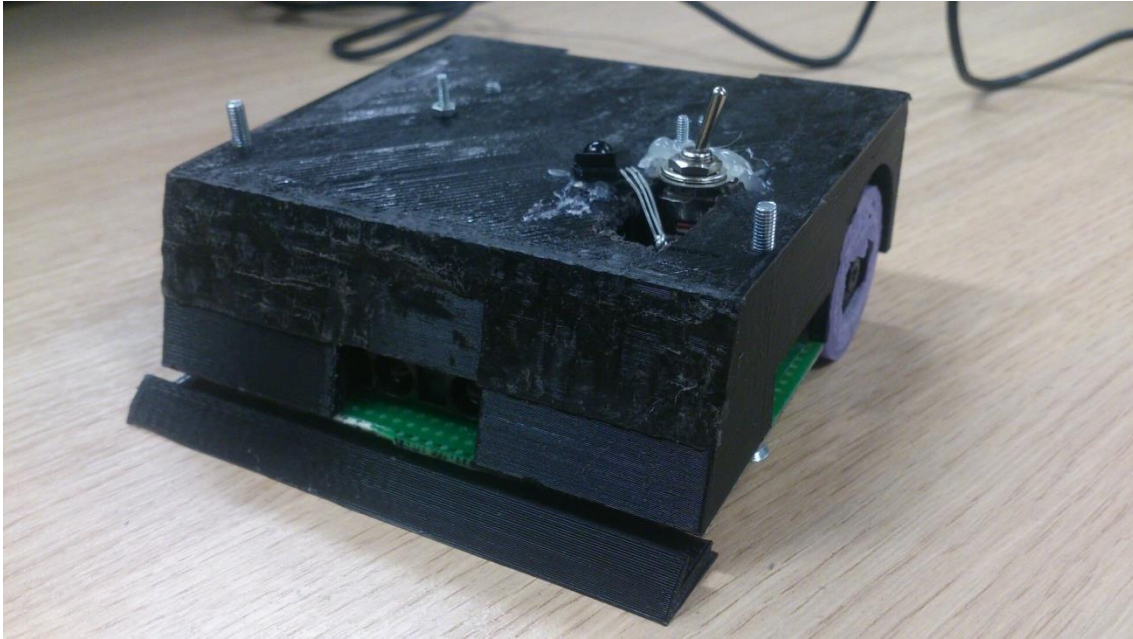
Järgmisena tuli teha roboti ratastele rehvid. Rehvid sai ise valatud silikoonist ning nende tegemiseks oli vaja teha luua ka vorm, mis sai selleks puhuks 3D-prinditud. Silikooni tahkestamiseks oli vaja see segada katalüsaatoriga. Kokkuvõttes tuli silikooni ja katalüsaatori segu teha 1:1 suhtega, ehk segusse oli vajalik panna silikooni ja katalüsaatorit sama hulk. Rehvid sai lõpuks tehtud lillast silikoonist, mis oli võimalikust valikust kõige sobivam antud suurusega rehvidele oma pehmuse ja tugevuse poolest (teisteks valikuteks olid roosa ja roheline, mis olid natuke pehmemad ja sobivamad rohkem paksematele rehvidele). Kokkuvõttes tegin kokku kolm rehvi, kuna esimese puhul sai valatud vigane rehvi, valamisega sai võetud liiga palju aega ja silikoon hakkas lõpus juba tahkestuma (antud silikoon hakkas tahkestuma 5 minuti peal ja oli täiesti tahke 10 minuti peal).

Järgmisena tuli teha ära elektroonsete osade jootmine. Esmalt sai ära joodetud suuremad komponendid nagu kontrollid ja mootori draiver, seejärel andurid ning muud väiksemad osad ning viimaks teha kõik vajalikud ühendused. Ühenduste skeem on ära näitatud järgneval joonisel.



Joonis 9: Roboti elektriskeem.

Pärast roboti jootmist sai lõpuks kokku panna. Vahepeal sai tehtud juurde ka roboti kere lakke auk, kust tuleb lae peale roboti toite lüliti ning IR vastuvõtja. Sellise muudatuse pidi tegema, kuna muidu poleks/oleks roboti sisselülitamine natuke halvasti tehtav tegevus, see poleks kusagilt hästi kättesaadav, ning kuna stardikäsk puldist antakse tavaliselt kõrgelt ülevalt, siis on katus vastuvõtjale sobiv positsioon.



Joonis 10. Valmis ja kokkupandud mini sumo robot.

Nagu enne sai öeldud, olid roboti kokkupanemisel väikesed probleemid. Kokkupanekul tekkinud probleemid sai aga lahendatud väikesekere muutmisega. Sellest probleemist on kirjutatud samuti punktis 3.3. Kokkuvõttes jäi roboti esialgseks kaaluks 237,5 grammi ning robot mahtus etteantud 10 cm x 10 cm piiridesse.

3.2 Programmi kood

Programm on roboti juures väga tähtis, kuna see määrab ära roboti käitumise. Antud töös sai roboti programmeerimiseks kasutatud tarkvara Atmel Studio 7.0, mis on firma Atmel poolt loodud tarkvara spetsiifiliselt nende poolt loodud protsessorite jaoks ning sobib antud töös oleva Atmel Atmega328P protsessori jaoks, mida kasutavad Arduino Nanod.

Antud robot on käitumusliku poole pealt valitud ründavaks ning programmi kood tuli kirjutada vastavalt sellele. Koodis tuleb ka ära märkida kontrolleri portide ja viikude orientatsiooni, kas need on sisendid või väljundid. Sisenditeks on siinkohal deklareeritud sensoritega seotud pordid (mis tuleb sättida ka ADC portideks, vaikimisi on need digitaalportid) ning lülititega seotud pordid. Väljunditeks on pordid, mis suunavad vajalikud mootorite liikumiskäsud edasi mootori draiverile. Peamised koodiosad on liikumise kood ning skaneerimiste koodid (IR sensorite ja jooneandurite

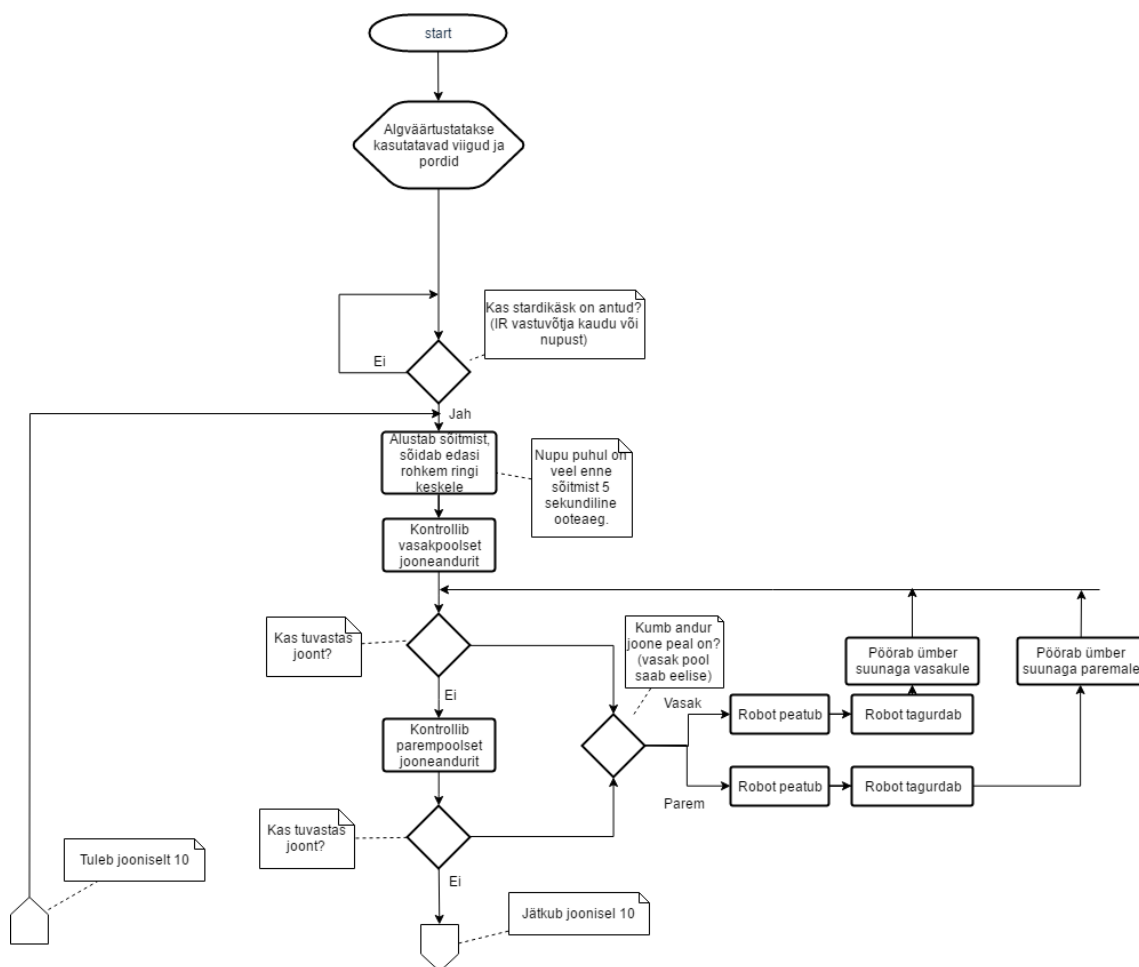
jaoks). Viimases on vaja enne sättida pordid vastavalt analoogsignaali sissevõtivateks portideks. Siinkohal tuleb öelda, et kontrolleri peal olevad viikude nimed ei ole tervenisti kooskõlas programmeeritavate portide ja viikudega (näiteks viigud D8-D13 kontrollerial on programmeeritud pordid/viigud B0-B5). Analooisisendid on üheks eranditeks sellisel juhul.

Enne liikumist teeb kontroller läbi skaneerimise tsüklid. Kõige esimesena läbitakse joonejälgimise tsüklid, kus vaadatakse läbi kontrolleri ADC viigud/pordid A3 ja A4, kuhu on ühendatud jooneandurid. Kui tuvastatakse valge joon, siis kontrollitakse, kumb andur joont tuvastas, ning vastavalt sellele pööratab robot ümber. Kui tuvastas vasak andur, pööratakse paremale ja vastupidi. Samal ajal kontrollitakse ka, et kas tuvastatakse vastase robot. Kui see tuvastatakse, lõpetab robot vastava funktsiooni ning hakkab järgima IR sensorite järgi. Vastasel juhul jätkab robot, kuni ta on ennast ümber pööranud. Pärast seda tegevust läheb robot tagasi normaalse käitumise juurde.

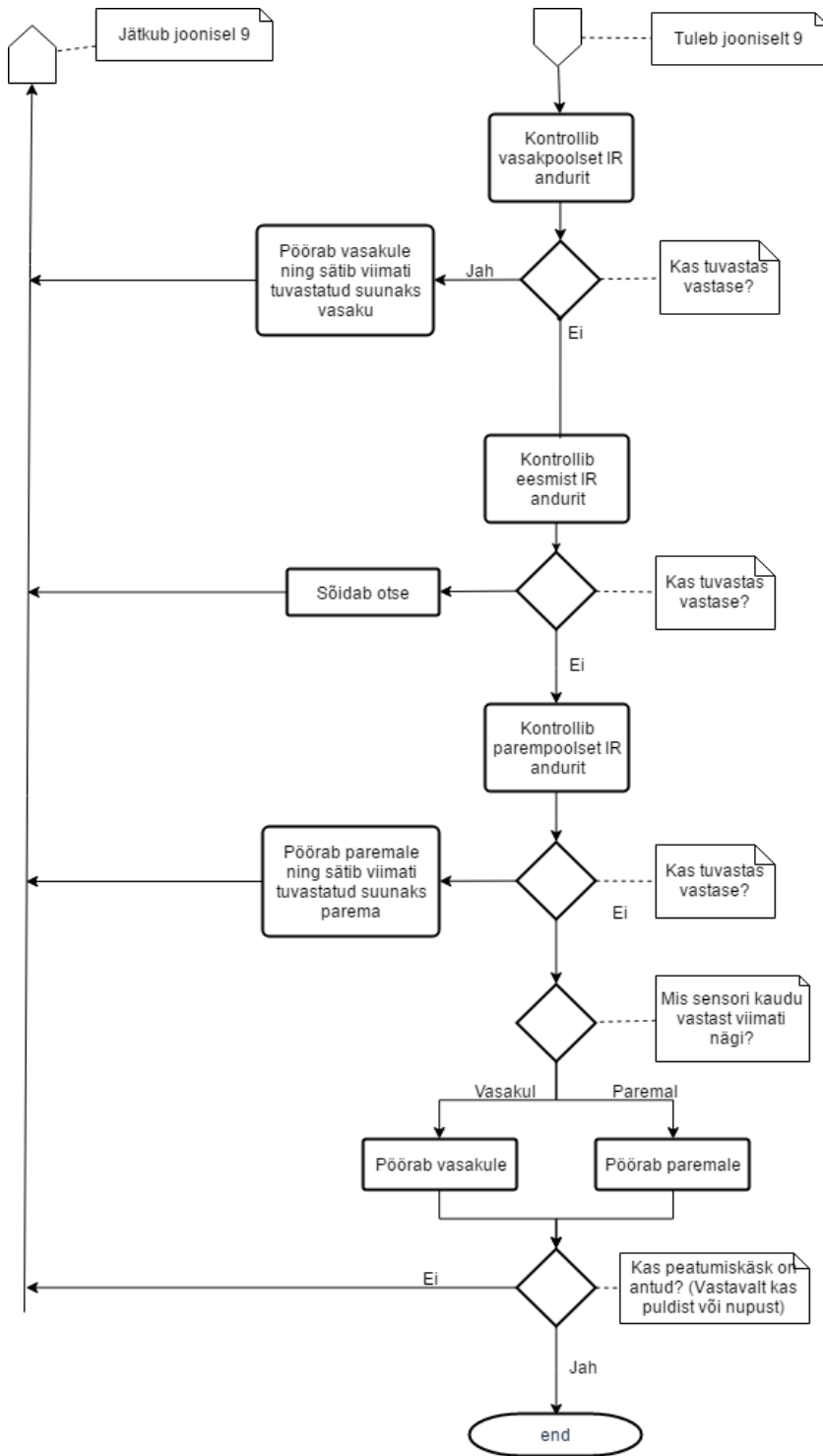
Kui joont ei tuvastatud, siis liigutakse edasi IR sensorite (viigud/pordid A0-A2) juurde ja hakatakse nende kaudu tuvastama vastast. Sellest, milline sensor vastast tuvastas, oleneb ka roboti käitumine. Kui vastase tuvastab eesmine sensor, liigub robot otse edasi vastase poole. Kui vasak või paremal, siis pööratakse vasakule või paremale nii, et roboti esiots oleks vastase poole. Kui vastast ei tuvastata, siis hakkab robot ümber oma telje tiirutama, kuni ta vastase leiab. Tiirutamise suund sõltub sellest, kus suunas robot viimati vastast nägi. See tagab ka selle, et roboti eesnurkades olev „pime koht“ midagi eriti suurt ei mõjuta. Kui näiteks on vastane tuvastatud vasakul, siis hakkab robot pöörama vasakule ning kui jõuab pimekohani, kus kumbki nii eesmine kui vasak andur seda vastast ei tuvastata, siis jätkab robot vasakule pööramist, kuni eesmine hakkab tuvastama vastast.

Liikumise poole pealt on defineeritud ära pordid, mis on ühendatud draiveril olevate portidega. Selleks vajalikud pordid on kontrollerial D7, D8, D9 ja D10 (programmeeritud pordid D7 ja B0-B2). Pordid D7 ja D8 on siinkohal signaali viigud, mis lubavad mootori draiveril mootorit tööle panna ning D9 ja D10 on PWM signaali viigud, mille kaudu antakse edasi PWM signaal draiverile ja öeldakse, kummas suunas tuleb mootorit keerata. Koodi poole pealt on ära määratud kindlad kiirused, millel mootorid võivad töötada.

Muud väiksemad funktsioonid, mida programmis on kasutatud, on rohkem seotud lisafunktsioonidega, nagu startifunktsioonid. Esialgu on olemas kaks viisi startimiseks, nupuvajutusega ning IR-vastuvõtja kaudu. Mõlemal juhul on need ühendatud mingisse sisendviiku. Nupuvajutusega stardi korral antakse kontrollerrisse signaal, mis alustab 5 sekundilist taimerit, ja siis hakkab tööle. Selline start on rohkem testimiseks teiste masinatega, millel on sarnane startifunktsioon. Teisel juhul võtab IR vastuvõtja vastu IR-signaali(puldilt) ning alustab kohe tegutsemist. Sellist starti kasutatakse turniiridel ning see jääb edaspidi põhiliseks startifunktsiooniks ning hiljem ka raundide peatamiseks.



Joonis 11: Roboti juhtimisalgoritm.



Joonis 12: Roboti juhtimisalgoritm jätk.

3.3 Ehitusel ja programmeerimisel tekkinud probleemid

Roboti kokkupanekul ning programmeerimisel tekkis mitmesuguseid probleeme. Kui ehitusel tekkinud probleemid ilmnisid just töös räägitava roboti kohta, siis programmeerimist sai peamiselt testitud ka paralleelselt tehtud teise mini sumo roboti peal, mida sai tehtud TTÜ Robotiklubi isetegemiskursusel suurema grupiga. Seetõttu sai osa programmeerimisega seotud probleemid enne õige roboti peale panemist ära parandada.

Roboti ehitusel tekkisid väikesed probleemid roboti enda struktuuriga. 3D-printimisel tekkinud väikeste vigade tõttu tuli teha roboti keres väikesi muudatusi. Peamisteks probleemideks olid kinnitusaukude mitteühildumine, seda peamiselt kere ning sahaga. Selle parandamiseks pidi saha kinnituse tegema natuke suuremaks niimoodi, et need augud ühilduks.

Lisaks tekkis probleem esialgselt valitud põhjaga, mis osutus juba parajaks löikamisel liiga nõrgaks. Nimelt parajaks löikamisel murdus osa makettplaadist, mis oleks pidanud minema roboti põhjaks, ära ning seda polnud võimalik ega arukas parandada. Edaspidi sai ostetud paksem ja tugevam makettplaat, mis sobis paremini põhjaks.

Pärast kokkupanekut ilmnis veel probleem, kus rataste rehvid läksid vastu roboti kere külge. Selle parandamiseks sai rataste eest ära lihvitud tremliga sein, kuni rattad enam kere vastu ei käinud.

Programmeerimisel tekkinud peamiseks probleemiks oli roboti juhtimisel käskude mittejärgimine. Väiksemal määral oli probleeme ka anduritelt info kätte saamisega, aga see probleemi lahenduseks oli lihtsalt portide õigesti sättimine ning selle sai kähku tehtud. Teise probleemi puhul oli aga lahendust raskem leida. Testimise käigus leidsin, et anduritelt tulevat infot töödeldakse õigesti ning see jõuab õigetesse ning vajalikesse kohtadesse koodis. Küll aga tekkis probleem, et mootorid ei tahtnud selle järel korralikult töötada.

Esialgu oli arvamus, et mootorid olid valedesse portidesse ühendatud või koodis oli valed pordid pandud ja osaliselt oli see ka tõene. Suurem probleem siiski püsis. Viga oli põhimõtteliselt selline, et mootorid ei reageerinud neile antud käskudele õigesti, ehk informatsioon jõudis küll vajalikesse punktidesse, aga kui tuli näiteks vasakule pöörata,

siis sellele vajalikku tegutsemist ei toimunud. Otse sõitis robot ainult siis, kui tuvastas kohe startides, et midagi tal ees on. Muudel juhtudel tegi ta ainult paremale pööramist ning sellest ta välja ei tulnud, isegi kui tuli midagi, mis oleks tal keelanud liikumise või öelnud, et tuleb liikuda teises suunas. Paremale hakkas ta ka pöörama siis, kui otse sõites tuvastas ta midagi paremal.

See probleem tekkis sellest, et vahepealsel koodimuutmisel oli süntaksis tekkinud viga. Nimelt oli vahepeal sätitud kood nii, et igal juhul, isegi kui oleks pidanud tagurdama, läks mootoritesse käsk, et tuleb otse sõita. Koodi olid PWM pordid valesti sätestatud ning seetõttu andis kontrolleri mootoritele ainult ühes suunas liikumise käsud.

4 Roboti testimine

Robotit sai testitud mitmel eri viisil. Kõige pealt sai testitud kõiki üksikuid komponente eraldi ning lõpuks ka kokkupanduna. Eraldi testimisel olid küll väikesed probleemid, peamiselt info kättesaamisega anduritelt, kuid need said kiiresti lahendatud programmis väikeste muudatuste tegemisega.

Kokkupanemisel sai ka testitud robotit, esialgu küll ilma kereta, kuna see oli kõige lihtsam viis tervet süsteemi testida. Selline testimine mingeid probleeme ei ilmutanud ja kõik töötas korralikult. Nii sai edasi liikuda järgmisse testimisetappi.

Pärast kokkupanekut ilmensid aga mõned väiksemad probleemid, mis robotil olid. Üheks probleemiks oli see, et roboti rattad, mis olid 3D-prinditud ja seepärast plastmaterjalist, hakkasid testide käigus võlli juurest kuluma ning üks ratastest seetõttu ei keerlenud. Selle parandamiseks sai liimitud ratas võlli külge. See on küll ajutine parandus ning tulevikus tuleks teha korralik parandus ratta sisse puurimise ja kinnituskrugi lisamise näol.

4.1 Võistlusolukord

Samuti sai robotit pandud võistlusolukordadesse. See kujutas endas matši simuleerimist, kus panin enda roboti vastamisi mitmete teiste sumorobotitega ning raundide

väljamängimist. Kokku sai võideldud 6 sumoroboti vastu, mis olid loodud TTÜ Robotiklubi isetegemiskursusel paralleelselt antud töös loodud robotiga. Seetõttu pidi kasutama antud „turniiril“ antud kursuse reegleid, mis üldiselt olid samad Robotex'i reeglitega. Ainsaks erinevuseks oli see, et juhtpuldi ja IR vastuvõtja kaudu stardikäsu asemel anti start roboti peal oleva nupu ning 5 sekundilise viite näol. Selleks on ka antud roboti juures olemas selleks vajalik nupp.

Teiste robotite vastu võideldes suutis roboti teistele vastu olla, kokkuvõttes võitis see kahte teist robotit ja jäi alla teisele neljale, aga sedagi enamasti ainult 1-2 kaotusseisuga. Peamiseks probleemideks, mis antud võistluse ajal ilmnesisid, oli roboti kaalu vähesus ning väikesed probleemid sensorite töötamisel.

Esimene probleem esines selles, et kuna robot kaalub ligi pool lubatust ning osad teised robotid olid raskemad, siis esines olukordi, kus robot lükati massiga välja. Teine olukord, mis juhtus, oli see, et pööramisel ja/või joonepealt eemaldumisel tõstis robot enda esiotsa natuke üles, mistõttu sai vastane kergemini roboti alla. Selle parandamiseks tuleb panna robotile raskusi juurde.

Sensorite poole pealt ei reageerinud/reageeris robot halvasti paremal pool olevatele vastastele. Siinkohal on probleem ilmselt selles, et kuna vasakpoolne ning eesmine andur uuritakse enne läbi, siis kui tuvastatakse ühes neist siis parempoolset üldse ei kontrollitagi. See probleem võib võimenduda, kui üks andur on suunatud allapoole (see probleem esineb rohkem eesmise anduriga), kuna sellisel juhul võib andur tuvastada maapinda. Selle parandamiseks tuleb andurid ükshaaval üle käia ja vaadata ega ükski neist maad ei tuvasta ning, kui vajalik, programmi muuta.

4.2 Edasiarendus

Nagu näha on, pole tehtud robot veel kõige paremas potentsiaalis, veel saab selle juures teha edasiarendusi. Esmasteks eesmärkideks on kindlasti roboti loomise ja testimise käigus tekkinud probleemide likvideerimine. Lisaks sellele võib robotile teha veel teisi edasiarendusi, näiteks uue saha loomine ning kere üldine tugevdamine.

Esmalt oleks hea aga parandada vead, mis robotil ehitamis- ja testimisjärgus tekkisid. Ehitusvigade kaotamine on küll esialgselt mahukam töö, kuna kere ja enamus muid sellega seotud asju on 3D-prinditud ning nende parandamiseks oleks vajalik mingil

määral kere ehituse muutmine ning uuesti printimine või hoopis ise tegemine mõnest muust materjalist. Ehitusvigade parandamine on küll mahukas töö, aga esialgu mitte eriti tähtis, kuna vead, mis tekkisid olid üsnagi väikesed ja roboti struktuuri ja terviklikkust mitte mõjutavad.

Küll saab parandada testimisel tekkinud probleemi seoses ratastega. Praegu sai see probleem parandatud liimiga, küll aga on see ilmselt ainult ajutine parandus. Lõplikuks paranduseks peaks olema kinnituskrugi lisamine rattasse, mis hoiaks ratast võlli küljes korralikult kinni. Selleks tuleks puurida (vajadusel läbi silikooni) rattasse ning lisada antud kinnituskrugi tehtud puurauku.

Roboti sensoritega tekkinud probleem on likvideeritav aga sensorite asetuse muutmise ning vajadusel koodi muutmisega. Sensorite asetust saab põhimõtteliselt muuta andureid ise liigutades, mis on küll natuke raskem tegevus, kuna nad on põhja külge kinnitatud, või siis tervet kere esiosa tõstes mõne toetuspunkti abil, näiteks kruvi läbi roboti põhja. Viimasel juhul tuleks aga arvestada sellega, et selle liiga kõrgele tõstmisel võib roboti saha ja kere vahele tekkida vahe, millesse vastane saab ennast suruda.

Programmeerimise poolest saab olukorda parandada sellega, et uurida ning sättida paremini väärtused, millest alates loetakse vastast ning tema poole liigutakse. Siinkohal eriti muutusi teha enam ei saa.

Lisaks eeltoodud edasiarendustele võib välja tuua veel uue saha loomise ning kere tugevdamise ning mingil määral programmi arendamine ja uuendamine. Antud robotile oleks hea luua uus sahk, mis läheks esiteks sujuvamalt roboti esiotsa peale ning oleks parema amortisatsiooniga. Hetkene nii-öelda vedrustus on kere raskus, mis saha peale suurelt osalt toetub. See aga pole eriti hea variant ning selle parandamiseks tuleks välja mõelda mingi teine lahendus. Lisaks on sahk kõrguse poolest natuke lühike ning seda tuleks tõsta.

Kere tugevdamist saab teha pealmiselt lisakihtide asetamisega kere külgedele. Seda oleks hea teha metallplaatide/-tükkide lisamise näol kere esiosale ning külgedele, kuna need on kõige rohkem lööke saavad osad. Kere tugevdamine suurendaks kere vastupidavust ja vähendaks võimalust, et vastase robot suudaks rünnakul roboti seinad ja/või esiosa katki lüüa. Lisaks võib roboti sahale panna metallitüki külge, seda nii saha

enda tugevdamiseks, kui ka saha maadpuutuva otsa teravdamiseks, nii, et vastasele oleks võimalik kergemini põhja alla saada.

Programmeerimise poolest saab robotit edasi arendada parandades ning muutes koodi, mille järgi roboti töötab. Peamised kohad, mida saab ja ilmselt tuleks arendada ja muuta, on stardiolukord ja joone tuvastamise olukord. Need olukorrad on hetkel tehtud üsnagi lihtsustatult ja roboti edasiarendamisel tuleks need kindlasti teha teistmoodi. On ka muid, väiksemaid, asju, mida koodis muuta, aga need kaks on hetkel peamised.

Praegu on stardiolukord seatud niimoodi, et stardimärguande peale liigub robot äärejoone juurest rohkem ringi keskele ja hakkab seal tiirutama, kuni leiab vastase. Seda saaks aga arendada kergelt niimoodi, et robot sõidab edasi, kuni ta vastase tuvastab või kuni ta valge jooneni jõuab. Lisaks võiks robot esialgu sõita väiksema kiirusega ja vastast tuvastades kiiremini.

Jooneolukord on teine, mida tuleks kindlasti edasi arendada. Hetkel on see tehtud niimoodi, et robot peatub joonel, tagurdab ja pöörab ümber, alati kindla aja ja pöörduse võrra. Sel ajal ta aga ei otsi vastast. Tulevikus tuleks see teha aga nii, et ta pööramise ajal jätkab vastase otsinguid ning kui ta tuvastab selle, siis hakkab ta sinna liikuma.

5 Kokkuvõte

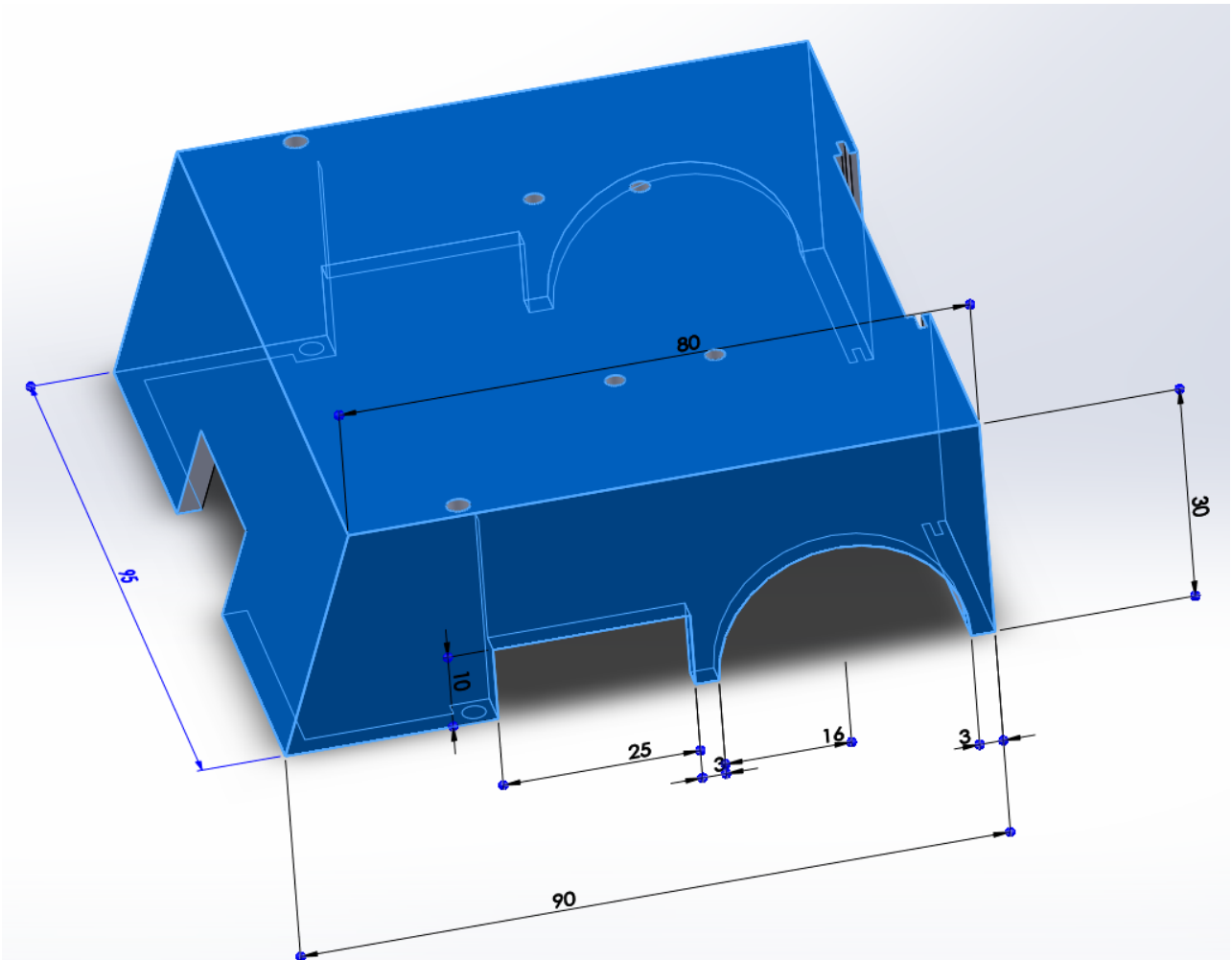
Antud töö eesmärgiks oli disainida, ehitada ja testida mini sumo robot, mis oleks ürituse Robotex reeglitele vastav ning tulevikus ka selleks kasutatav. Robot sai disainitud võimalikult madal, et vastasel oleks võimalikult raske robotit tuvastada ning võimalikult kompaktne, ehk nii, et kõik võimalikud komponendid mahuks roboti kere sisse.

Vaatamata probleemidele, mis tekkisid roboti loomisel ja testimisel, võib öelda, et võistlusolukorras suutis robot ennast tõestada. Siiski on veel kohti, kus robotit võib edasi arendada ja viimistleda, seda eelkõige roboti ehituses ning programmis.

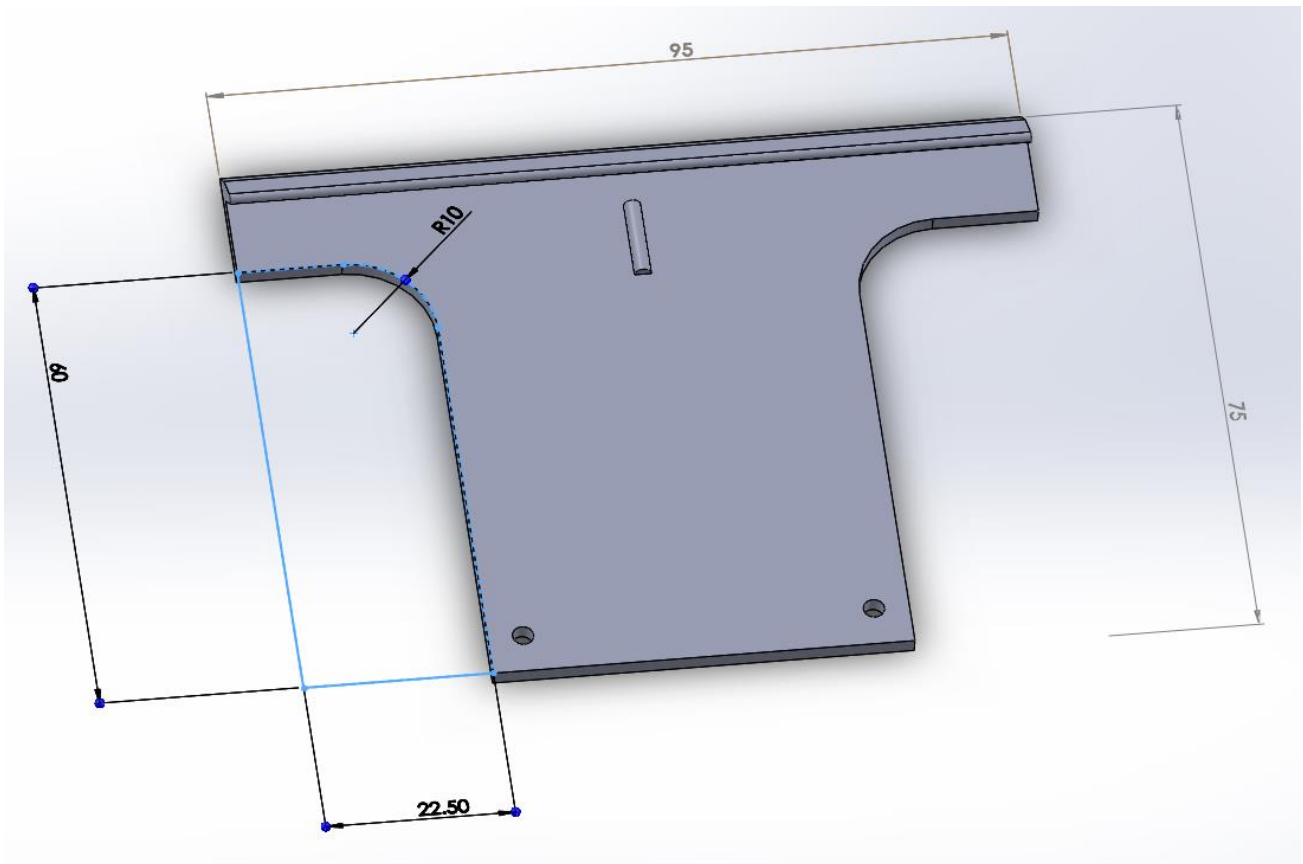
Kasutatud allikad

- [1] McGraw-Hill Education, „McGraw-Hill Education,“ 17 11 2005. [Võrgumaterjal]. Available: http://books.mcgraw-hill.com/downloads/products/007222617X/007222617X_ch01.pdf. [Kasutatud 14 Mai 2017].
- [2] MTÜ Robotex, „Tutvustav Materjal,“ 2016. [Võrgumaterjal]. Available: <http://2016.robotex.ee/docs/file/2016/tutvustav%20materjal/Robotex%202016%20tutvustav%20materjal.pdf>. [Kasutatud 14 Mai 2017].
- [3] MTÜ Robotex, „Robotex,“ MTÜ Robotex, 2017. [Võrgumaterjal]. Available: <http://robotex.ee/voistlused/>. [Kasutatud 14 Mai 2017].
- [4] R. Pääru, „Robotex sumo reeglid,“ MTÜ Robotex, 25 September 2016. [Võrgumaterjal]. Available: <http://www.robotex.ee/docs/file/2016/Reeglid/Robotex%202016%20Sumo%20reeglid.pdf?>. [Kasutatud 8 Märts 2017].
- [5] Arduino, „Arduino,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardNano>. [Kasutatud 10 märts 2017].
- [6] Pighixxx, „Pighixxx,“ 19 August 2014. [Võrgumaterjal]. Available: <http://www.pighixxx.com/test/pinouts/boards/nano.pdf>. [Kasutatud 8 Märts 2017].
- [7] Pololu, „Pololu Robotic and Electronics,“ [Võrgumaterjal]. Available: <https://www.pololu.com/product/2135>. [Kasutatud 8 Märts 2017].
- [8] Pololu, „Pololu Robotics and Electronics,“ Pololu, [Võrgumaterjal]. Available: <https://www.pololu.com/product/3073>. [Kasutatud 10 märts 2017].
- [9] Pololu, „Pololu Robotics and Electronics,“ Pololu, [Võrgumaterjal]. Available: <https://www.pololu.com/product/2458>. [Kasutatud 10 märts 2017].
- [10] Pololu, „Pololu Robotics and Electronics,“ Pololu, [Võrgumaterjal]. Available: <https://www.pololu.com/product/2474>. [Kasutatud 10 märts 2017].

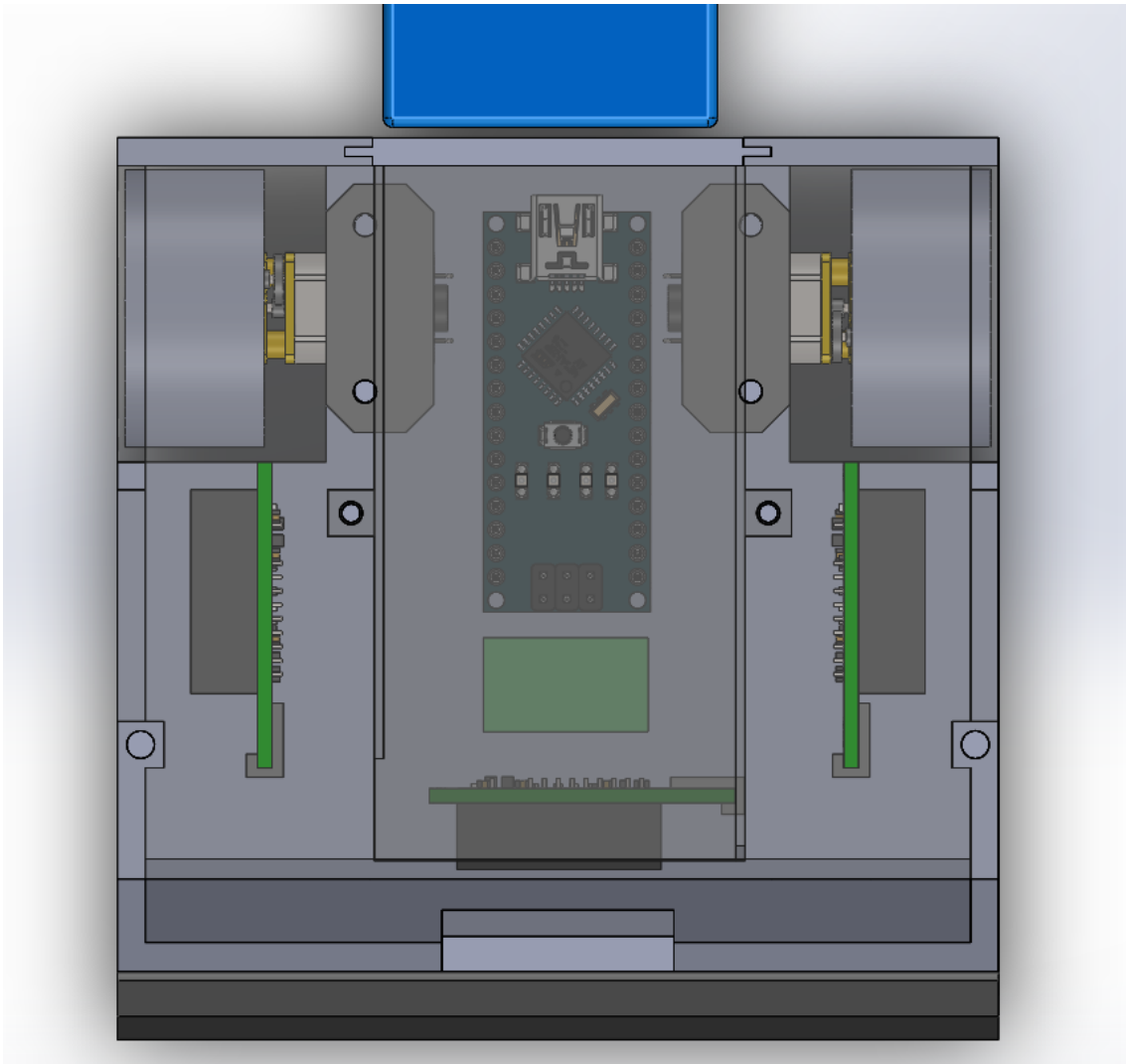
Lisa 1 – Roboti kere



Lisa 2 – Roboti sahk



Lisa 3 – Roboti sisemus pealtvaates



Lisa 4 – Roboti skaneerimise koodid

```
#include "scan.h"
uint8_t port ; // Port number
uint16_t adc_val = 0 , direction, white, stopVal ;
// Value received through ADC; direction markers(0 front, 1 left, 2 right, 4
nowhere to be
// found), marker for the port that saw the line and
// stopValue-ground marker(black or white)(white means it's about to go off
the ring)
uint8_t IRScan( ) // IR sensor function
{
    direction = 4 ;// Default direction if it can't see the enemy is 4
    for ( port = 0 ; port < 3 ; port++ ) // As long as the port number is
under 3(signal for IR sensors comes through ports 0-2)
    { // 5-7 is unused
        adc_val = read_adc( port ); // Reads through a channel
        if(adc_val >= 250 && port == 1) // if it detects something in
front
        {
            direction = port ; // it determines a direction, based on
the port/sensor that has seen the enemy(0 is forward, 1 is left, 2 is right)
            break ;// Breaks from the function
        }//If it's zero, it will continue
        else if(adc_val >= 450 && port != 1) //if it reads from any
other ports(0 or 2)
        {
            direction = port ;
            break ;
        }
        else if( adc_val == 0xffff ) //Error code, sets the port number
back to 0 to check again
        {
            port = 0 ; // sets port number to 0
        }
    } //If the enemy isn't seen, the direction
marker will stay the same and robot will act accordingly
    return direction ; // returns the direction
}
uint8_t lineScan( ) // Function to scan for the white line
{
    for ( port = 3 ; port < 5 ; port++ ) // Uses ports 3 and 4 for
information
    {
        adc_val = read_adc( port ); // Reads from the port

        if( adc_val >= 550 && adc_val < 1024 ) // Max value is 1024, if
550 or up
        {
            stopVal = 0 ; // the robot is on black
        }
    }
}
```

```

else if ( adc_val <= 549) // if the value is lower than this
value
{
    stopVal = 1 ; // the robot is on the white line
    break;
}
else if ( adc_val == 0xffff ) // if it happens to get an error,
{
    port = 3 ; // and also set the port back to 3
}
}
return stopVal;
}
uint8_t portScan( ) // Function to check, which port encountered the white
lines
{
    for ( port = 3 ; port < 5 ; port++ ) // Uses ports 3 and 4 for
information
    {
        adc_val = read_adc( port ); // Reads from the port

        if ( adc_val <= 549) // if the value is lower than this value
        {
            left one
            if(port == 3 ) // if the port that saw the line is the
            {
                white = port ; // Sets the white value to 3
                break ;
            }
            right one
            if (port == 4 ) // if the port that saw the line is the
            {
                white = port ; // Sets the white value to 4
            }
        }
        else if ( adc_val == 0xffff ) // if it happens to get an error,
        {
            port = 3 ; // and also set the port back to 3
        }
    }
    return white ; // Returns the port number that saw the white line
}

```