

Tallinna Tehnikaülikool  
Mehhatroonika instituut, Robotitehnika õppetool

Aruanne aines  
Praktiline Robotika, teemal  
**Roomba robotid ja nende programmeerimine**

Õpilane: Viktor Beldjajev, 040379 AAAM41  
Juhendaja: Indrek Tubalkain

Tallinn 2009

# 1. Sisukord

1. Sisukord .....	2
2. Ülesanne.....	3
3. Ideed.....	3
4. Spetsifikatsioon.....	4
4.1    Roboti kirjeldus.....	4
4.2    Elektroonika üldskeem.....	5
4.3    Algoritmi selgitus.....	6
4.4    Fotod .....	7
5. Kokkuvõte.....	8
6. Kasutatud kirjandus.....	9

## 2. Ülesanne

Meie ülesandeks oli tolmuimeja tüüpi robot *Roomba* sedaviisi ümber ehitada, et tolmu koristamise asemel tegeleks ta sumovõistlustel teiste samasuguste robotitega võistlemisega ning lükkaks neid üle lahinguväljaku ääre. Ümberehitamise ajal tuli järgida võistluse reegleid ning modifitseerida *Roomba* just niipalju, et mitte reeglitega pahuksisse minna. Näiteks oli äratoodud *Roomba* maksimaalselt lubatud kaal, milleks oli 3,8 kg.

Kõik kursusest osavõtjad jaotati tiimideks, kus üks robotiklubi liige oli juhendaja ja teised olid muudelt kursustelt tudengid. Ülesanne oli kõigil tiimidel üks ja sama, kuid lahendusvariandid sõltusid suuresti grupis olevate inimeste fantaasiast ning oskustest. Robotit käidi ehitamas 1 kord nädalas planeeritud aegadel, meil oli selleks neljapäev kell 18.00

Meie tiim oli nr. 2, kuhu kuulusid:  
Indrek Tubalkain  
Viktor Beldjajev  
Imre Drovtar

Roboti nimeks oli valitud Baruto, Eesti tuntud sumomaadleja järgi.

## 3. Ideed

Arutluse käigus oli meil tekkinud palju võimalikke ideid roboti ümberehitamiseks, osadest mõttetest pidime loobuma kas nende realiseerimatuse, keerukuse, reeglitevastasuse või ebaratsionaalsuse tõttu. Siinkohal tooksin mõningaid mõtteid, mis leidsid poolehoidu, ning neid, mis ei leinud ja miks.

- Panna robotile ette 2 IR tüüpi andurit vastase nägemiseks. Mõte leidis poolehoidu, kuna 2 anduri abil oli robot võimeline määrama ka suuna, kus vastane asub.
- Panna robotile taha üks analoogandur UltraSonic määramaks ohtu ning selle kaugust tagantpoolt. Mõte leidis poolehoidu ning oli ka realiseeritud. Kahjuks pidime sellest loobuma tekkinud raskuste tõttu see anur õigesti tööle saada. Asendasime selle anduri tavalise IR anduriga, mille ühendamine kontrolleriiga ning tööle saamine oli lihtne.
- Võtta ära tolmuimemiseks vajalikud lisaseadmed. Tegime selle ära ruumi võitmiseks.
- Võtta rataste alt ära vedrud ning liimida rataste toed korpuse külge, et vedrud ei viskaks roboti tagaosa üles, kui ta peaks üle ääre sõitma. Hiljem osutus, et vedrud oleksid pidanud sinna jääma, kuna tagaksid parema vastupanu sellistele robotitele, kes kasutasid sahka teise roboti ülestõstmiseks.

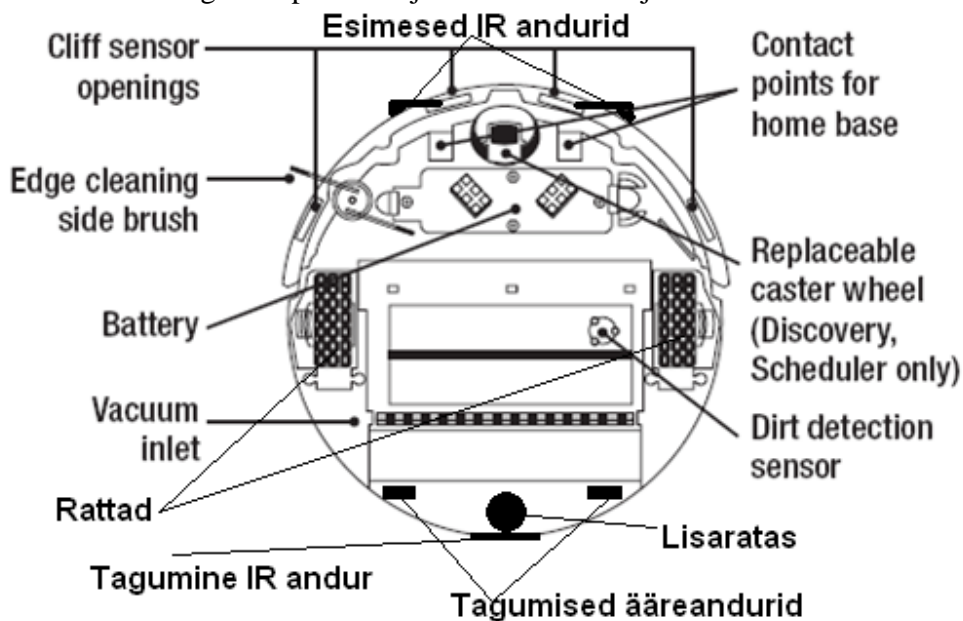
- Lisada robotile lisakaal. Tegime ära, lisades ühe aku ning koti erinevate kruvidega, kaaluks tuli 3,782 kg.
- Ehitada pinget tõstva toiteploki ning rakendada kõrgemat pinget mootorile lühiajaliselt ründamise hetkel. Mõttest loobusime põhjusel, et reeglid seda teha ei lubanud.
- Loobuda kahest esimesest ääreandurist. Kõik jäid alles, kuid programmis sai ühe ääre omad grupeeritud 1-ks ja teise ääre omad teiseks.
- Lisada mõni vastase andureid segadusse ajav signaalitekitaja, et vastane ei näeks meie robotit. Sellest mõttest loobusime reeglite mittevastavuse tõttu.

## 4. Spetsifikatsioon

Järgnevalt on ära toodud roboti tähtsamad osad, mis tegid temast võitlusvõimelise masina.

### 4.1 Roboti kirjeldus

Roboti ehitus ning tema põhiosad ja lisad on toodud joonisel 4.1.



Joonis 4.1. Roomba ehitus ning põhiosad ja lisad

Roomba mootorid ja elektroonika saavad toite 14.4 V akupatareist. Meie robotile oli alles jäänud 2 mootorit, mis panid liikuma vedavad rattad. Mootorid võimaldasid Roombal liikuda edasi-tagasi, keerata vasakule ja paremale. Maksimaalne kiirus on 500 mm/s. Rataste alt võtsime ära vedrud ning kinnitasime rataste šarniirid limiga korpuse külge. Roombale panime juurde veel ühe lisaratta tahapoole.

Roombasse on sisse ehitatud mitu erinevat andurit. Roomba ees on 2 kokkupõrkeandurit, mis tuvastavad, kas robot on millegagi kokku põrkanud (*Bump sensor*). Lisaks esimesel

äärel on 4 infrapunast ääreandurit(*Cliff sensor*), mis jälgivad, et robot ei kukuks üle väljaku ääre. Omalt poolt ehitasime robotile ette veel 2 infrapunaandurit, et saaks määrata vastase asukohta. Üritasime panna tööle *Ultrasonic* tüüpi andurit, et võimaldada Roombal tuvastada taganpoolt lähenevat vastast, kuid kahjuks ei saanud seda tarkvaraliselt tööle, seega pidime roomba tagumisele poolele monteerima samasuguse infrapunaanduri, nagu ettegi. Lisaks sellele monteerisime Roombale taha alumisele äärele ka kaks infrapuna andurit valge ääre tuvastamiseks, vältimaks roboti väljasõitmist tagurdamisel.

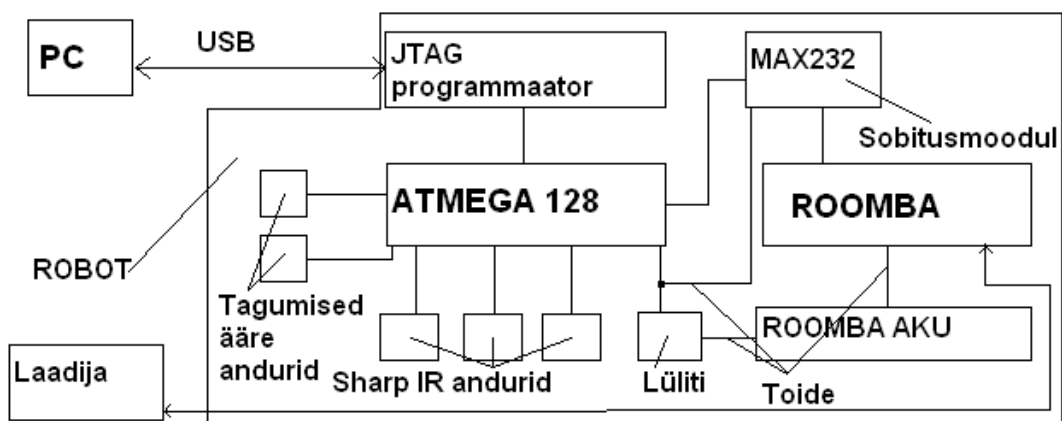
Kuna tühimass oli Roombal ligikaudu 2,5 kg siis lisasime talle lisaraskuseid, et muuta ta vastastele raskemaks. Selleks kinnitasime ühe aku roomba peale ning sisemusse lisasime kruve ja polte täistopitud kotikese ning kaaluks osutus 3,782 kg.

## 4.2 Elektroonika üldskeem

Roomba sisseehitatud elektroonikale lisasime Atmega128 mikrokontrolleriga arendusplaadi. See oli põhiliseks juhtmooduliks. Suhtluseks Roomba ja lisakontrolleri vahel sai kasutatud jadaliidest ja MAX232 integraalskeemil baseeruvat ühituslülitust. Lisakontrolleri programmeerimiseks sai kasutatud JTAG liidest koos vastava USB to JTAG mooduliga. Vastase leidmiseks kasutasime digitaalseid infrapuna andureid SHARP GP2D12 (2 tk roboti ees osas ja 1 taga osas). Lisaks sai ise valmistatud 2 valge ääre tuvastamise andurit, mis koosnesid: IR diodist, mis kiirgas IR valgust anduri all olevale pinnale ja IR foto transistorist, mille takistus muutus vastavalt pinnalt tagasipeegeldunud valgusele.

Lisaks sai kontrolleri arendusplaadi stabilisaatorile valmistatud suur alumiiniumist radiaator, kuna stabilisaatorile jäi suur pingelang (u 10 V) ja selle tulemusel kuumenes stabilisaator üle.

Järgneval joonisel on toodud roboti üldine blokk skeem:



Joonis 4.2 - Roboti üldine blokk skeem

Programmeerimiseks kasutasime WinAVR tarkvara koos vastava teegiga, programmeerimiskeeleks oli C++.

### **4.3 Algoritmi selgitus**

Kuna eesmärgiks oli luua võitlusrobot, siis algoritm baseerub võitlustehnika sooritamisel. Pärast seda, kui said defineeritud ja süsteemi integreeritud kõik roboti külge monteeritud andurid, saime hakata välja töötama nende signaalide loogikal baseeruvat programmi. Programmi täitmine toimus tsükliliselt, seega tuli eelkõige vältida kinniste tsüklite tekkimist programmis, mis võisid põhjustada vigu programmi täitmisel. Samuti loobusime kasutamast sellist käsku nagu *Waittime*, kuna see pidurdab programmi täitmise ning informatsioon osadelt anduritelt ei pruugi jõuda kontrollerini.

Algoritm nägi välja nii: Kui Roomba lülitati sisse, siis ta algul kalibreeris tagumised ääreandurid musta pinnaga, et valgele pinnale jõudmisel sealt kohe väljuda. Seejärel hakkas ta täiskiirusel ringi pöörlema, kusjuures suund anti kätte käivitamisel vajutades vastavat nuppu, ning otsima vastast. Kui üks eesmistest infrapunaanduritest vastase registreeris, siis hakkas Roomba liikuma mööda kaarekujulist trajektoori kiirusega 300 mm/s vastase suunas vastavalt sinna suunas, millise suuna andur signaali andis. Juhul kui mõlemad andurid registreerisid vastase otse ees, mindi otse vastase suunas.

Kokkupõrke tekkimisel hakkas robot täiskiirusel suruma vastast. Kui reageeris üks kokkupõrkeanduritest, siis pööras Roomba vastavas suunas, et vältida küljesuunalise tõuke tekkimist vastase poolt. Kui mõlemad kokkupõrkeandurid reageerisid, siis surus Roomba täiskiirusel otsesuunas.

Kogu programmi täitmise prioriteet oli suunatud eesmistele ja tagumistele ääreanduritele jälgimisele. Juhul kui täites ükskõik millist programmi osa reageerisid ääreandurid, sai Roomba otsekohe käsu liikuda äärest eemale, vastavalt siis kas edasi või tagasi suunas.

Osade funktsioonide realiseerimiseks kasutasime sisseehitatud käskude nagu näiteks *GetTimeStamp*, *GetDistanceInCm*.

#### 4.4 Fotod



Joonis 4.2. Baruto võistlusväljakul



Joonis 4.3. Roomba

## 5. Kokkuvõte

Võistlusel ei saavutanud Baruto paraku mitte ühtegi autasustavat kohta. Põhjuseks oli Leedu vastaste osav ehitus, milles oli kasutatud sahka ning mille vastu Baruto oli kahjuks võimetu. Küll aga võitis Baruto nii mõnegi lahingu, kus vastasteks olid „klassikalist” (nügimisega) sumostiili harrastavad robotid. Võistlustel juhtus ka nii, et kui mindi otsadega kokku sarnaste robotitega, siis miskipärast vedas meid meie roboti mehaanika alt, sest meie robot ei suutnud suurt vastupanu osutada võrdesel lükkamise situatsioonis. Põhjuseks võis olla rihmülekande kulumine.

Järeldus.

Kursus oli väga huvitav. Kursuse ajal õppisime tundma roboti ehitust, mehaanilisi ülekandeid, toitesüsteemi, elektroonikat. Samuti õppisime andurite paigaldamist robotile ning nende integreerimist roboti üldisesse juhtimissüsteemi. Huvitav oli roboti programmeerimine, nii erinevate objektide lisamine süsteemile kui ka võistlusalgoritmi väljatöötamine. Selleks olid meile abiks robotiklubi liikmed.

Kursus andis ka hea ülevaate roboti ümbruskonna tajumisest ning selle modifitseerimisest, pani mõtlema teatud kitsenduste ja piirangutele, näiteks roboti liikumiskiiruse valikul pidi ka arvesse võtma, et maksimaalsel kiirusel ei pruugi robot alati õigel ajal märgata piiri ning võib jõuda liiga kaugele ja üle ääre sõita.

Kindlasti oleks soovitatav sellise kursuse edaspidinegi läbiviimine. Võibolla võiks järgmisel korral teha programmeerimisega nii, et seletatakse kõigile ja korraga, sest see võib kokku hoida palju aega ja arusaamatusi, mis tekkivad, kui rääkida sellest kõigile inimestele eraldi.

Suured tänud kõigile korraldajatele.



## 6. Kasutatud kirjandus

1. Kurt T.E. Hacking Roomba. Wiley Publishing Inc. USA 2007. 436 lk.
2. Roomba Serial Pordi kasutusjuhend
3. Sharp IR sensorite kasutusjuhend
4. AVR mikrokontrolleri kasutusjuhend
5. *UltraSonic* anduri kasutusjuhend