

TTÜ Robotiklubi

# Teeme ise 2013

## “Jalgpall”

aruanne

Marianne Nugis

Arthur Randjärv

Kristjan Jansons

Tallinn 2013

## Sisukord

Sisukord .....	1
Töö üldpõhimõte.....	2
Roboti valmistamine .....	2
Roboti tööpõhimõte .....	3
Roboti programmeerimine .....	3
Põhi programmi lihtsustatud skeem: .....	3
Kaardistamine.....	4
Leitud liikumis nimekirja täitmine.....	4
Lühima tee leidmine uuele asukohale .....	4
Põhi probleem programeerimisel .....	4
Kokkuvõte .....	4

## Töö üldpõhimõte

Eesmärgiks oli luua labürindis navigeeriv robot, mis peab aja peale leidma üles labürindi lõpp ruudu. Labürint koosnes 7\*7 ruudustikust ja jga ruudu suurus on 160x160mm. Põhi reeglid mida täitma peab on:

- Robot peab mahtuma 160x160mm ruutu
- Robot peab olema täielikult autonoomne
- Robot ei tohi endast maha jätta mitte midagi
- Robot ei tohi liikuda üle seinte

## Roboti valmistamine

Robot tuli komplekteerida järgnevatest komponentidest:

- Pisi-Xbee4 – Elektroonika mille abil juhitakse robotit
- Pololu 3Pi mootor
- Pololu rattad
- Pololu tugikuul
- IR diodid
- IR Fototransistorid

Roboti kere otsustasime me teha trükkplaadist millele on lihtne juurte lisada Pisi-Xbee4 moodult. Esialgseks kriteeriumiteks oli võimalikult väike kere ja võimalikult vähe juhtmetega ühendusi, et roboti välimus oleks parem. Idee et robot peab olema väike tuli sellest, et väiksema robotiga on lihtsam labürindis navigeerida.

Labürindi seinte tuvastuseks kasutasime 6 IR Fototransistorit, nii et neist 2 olid suunatud roboti külgedele, järgmised 2 olid suunatud otse ette ja viimased 2 olid diagonaalis. Reaalselt diagonaalis olevaid andureid oma koodis ei kasuta aga see lisab roboti edasiarenguks võimalusi. Kõik Fototransistorid asuvad roboti põhi kere all eraldi moodulil, mis on kinni keeratav kruvidega.

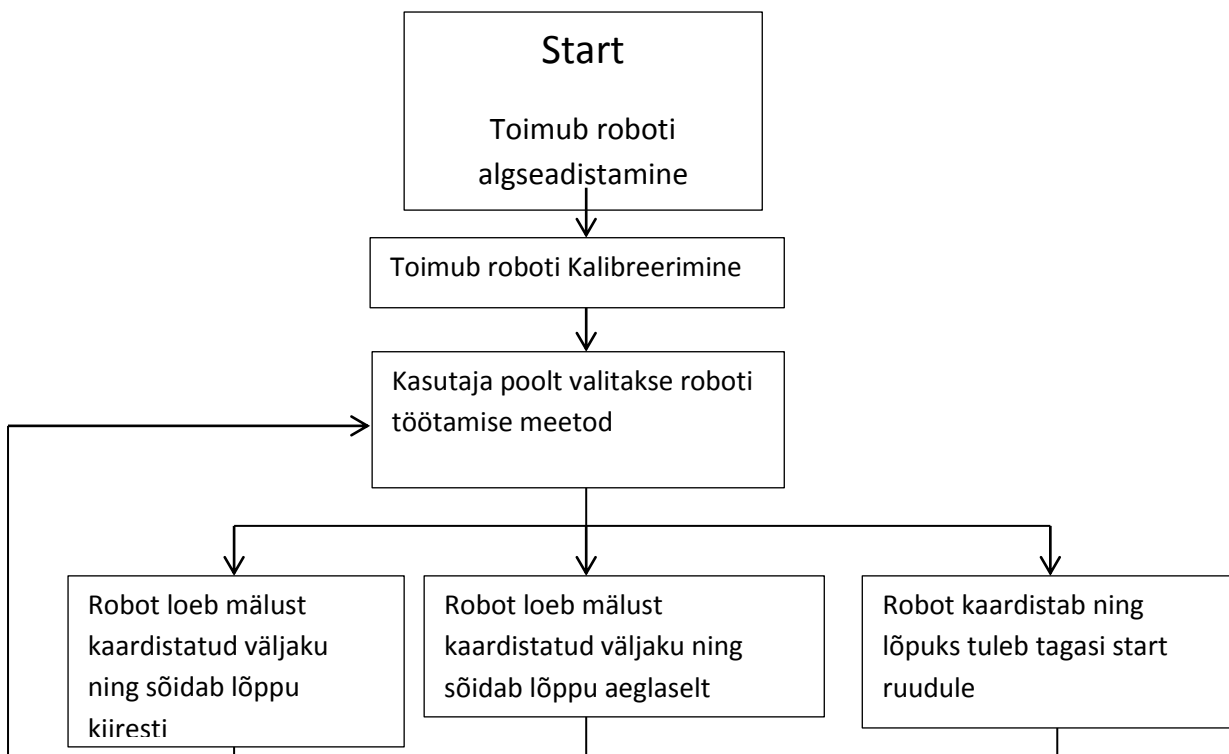
Roboti põhi kere trükkplaadil on IR LEDid ja nende lülitamiseks vajaminev elektroonika. Lisaks on kere külge pandud ka toite lüliti ja sinine LED mis põleb siis kui toite lüliti on sisse lülitatud. Idee et igat IR LEDi oleks võimalik sisse ja välja lülitada tuli sellest, et sedasi on võimalik tuvastada müravalgust mida võib mänguväljakul tekkida, kuid lõpuks tekitas see pigem probleeme kuna tekkis sensorite vahel lugemi erinevus.

## Roboti tööpõhimõte

Roboti tööpõhimõteks on kõigepealt kaardistada labürinti ja see järel otsida kõige otsem tee algusest lõpuni. Kaardistamine käib põhimõttel, et robot jätab meelde kõik kohad kus tal on veel käimata ja käib need lõpuks läbi. See tagab täieliku kaardistamise. Mis jäi kaardistamise puhul puudulikuks oli vea parandus. Kui roboti enda asukoht läheb paigast ära siis kirjutatakse alati üle ka vana juba uuritud kaardi osa ja seega väikse vea tõttu võib rikkuda kogu kaardistamise protsessi. Lisaks tuli ka lõpuks välja, et kõike lühem tee ei ole alati kõige kiirem ja peaks muutma teekonna otsimise algoritmi nii, et otsitakse kõige kiiremat teekonda mitte kõige lühemat. Üheks plussiks oli roboti disaini puhul tema mõttmed kuna väike robot suudab paremini liikuda kitsasdes tingimustes, kuid suureks veaks osutus halb informatsioon sensoritelt ja seetõttu ei püsinud robot tihti käikude keskel ja vahepeal isegi keeras ennast risti seinaga mis muutis kaardistamise keerukaks sest kaardistades on vaja täpset roboti asukohta hoidmist. Kindlasti aitab selle vea vastu uute takistite määramine fotransistor kollektori ja toite pinge vahele. Praegune valik oli halb sest fototrasnistor töötas tööpiirkonnas, kus toimus äkiline muutus umbes 10 cm kauguses. Lisaks oli sensori väärtuste põhi tööpiirkond ka liiga väike umbes 30-200. Kõige optimaalsem oleks olnud 100-900.

## Roboti programmeerimine

### Põhi programmi lihtsustatud skeem:



## Kaardistamine

Väljaku kaardistamiseks hoitakse robotil mälus seinte mälu masiivi ja masiivi kuhu kirjutatakse juba käidud kohad. Kaardistamisel vaatab robot alati enda ette ja kõrvale ning kirjutab nähtud endale mällu. Seejärel juba mälust vaatab üle kõik enda ümbruses olevad ruudud ning seal kus pole käidub need lisab nimekirja kuhu robot peab veel liikuma. Kui ümbrus on uuritud siis võetakse nimekirjast viimane element kuhu on vaja minna ning arvutatakse teekond sinna jõudmiseks. Kui teekond on arvutatud täidab robot teekonna nimekirja ja liigub uuele ruudule, kus hakkab robot jälle ümbrust jälgima. Kõik see kordub niikaua kuni nimiekiri, kus on kirjas järgmine asukoht on tühi.

## Leitud liikumis nimekirja täitmine

Liikumise nimekirjas on kirjas  $x$  ja  $y$  koordinaat kuhu robot peab järgmisena liikuma. Selles nimekirjas peab olema alati kahe koordinaadi vaheline kaugus 1 ja diagonaalis liikumine pole võimalik. Selleks et liikuda järgmisele ruudule leitakse kõigepealt nurk, kui palju peab robot pöörama, et sirgelt liikudes jõuaks määratud ruudule. Nii järjest iga ruudu keskel pöörates ja ruudult ruudule sõites liigutakse ühest asukohast teisele.

## Lühima tee leidmine uuele asukohale

Kõigepealt kasutatakse uputamise meetodit roboti enda asukohast kuni määratud asukohani kuhu peab jõudma. Uputamis meetod on selline, kus järjest täidetakse ära igal ruudul kui mitme ruudu kaugusel on robot, kasutades selleks ära kõrval oleva ruudu kaugust. Seejärel kui on üles kirjutatud käikude arv ruutudele siis hakatakse määratud ruudust järjest otsima ruutu mille puhul peab tegema kõige vähem käike niikaua kuni on leitud täielik teekond. Kõike seda tehakse funktsiooniga  $\text{FindWayToPoint}(x,y)$ . See funktsioon kirjutab leitud teekonna roboti liikumise nimekirja (Movelist) ja tagastab käikude arvu. Kui teekonda ei leita on käikude arvuks 0.

## Põhi probleem programmeerimisel

Enamus programmeerimisele planeeritud ajast kulus sellele, et tööle saada loogika, mis pidi leidma teekonda labürindis suvalisest punktist teise punkti. Põhi aeg kulus vea otsimisele millest ei saanud enne lahti, kui lihtsalt pidi koodi selle osa täiesti ümber kirjutama. Ilma selle veata oleks ka kindlasti jäänud rohkem aega roboti kiiruse arendamiseks. Lõppvõistluseks sai valmis põhiliselt kaardistamine ja aeglaselt liikumine algusest lõppu. Kuna keset võistlust sai veel oma robotit natuke muuta sai kiiruga tehtud kood mille abil suutis robot läbida labürindi ajaga 12.7 sekundit.

## Kokkuvõte

Tänu kursusele „Teeme ise 2013“ sai proovida realselt ühe roboti kokku panemist peaaegu nullist ja oli palju millest õppida, et järgmine kord robotit tehes enam samu vigu ei teeks. Lisaks oli väga kasulik Eagle CADi kursused, kus sai selgeks skeemide kokku panemine ja trükplaadi disaini loomine, mis on kindlasti tulevikus väga vajalik oskus. Veel sai proovida ühte võimalust kuidas väljamõeldud trükplaadi disaini on võimalik reaalsele trükplaadile saada. Ning lõpuks sai ka proovida komponentide jootmist trükplaadile.