Tallinna Tehnikaülikool

„Mikrokontrollerid ja praktiline robootika“

Roomba sumoroboti aruanne

Ester Uibopuu
082807
Rain Ellermaa
082775
Marius Siigur
082802
Madis Tuul
082806

MAHB-21

Tallinn 2009

Sisukord

Ülesanne

Võistkonnal tuli Roomba tolmuimeja muuta sumo võistlusrobotiks. Selleks tegime erinevaid muudatusi roboti konstruktsiooni osas, kus oli vaja elemente eemaldada, lisada ja ka muuta. Suur rõhk oli ka programmeerimisel. Kõike neid muudatusi tuli teha samaaegselt jälgides võistluse ülesannet, reegleid, nõudeid ja võimalusi.

Võistlusel on ühe matši ajal kaks võistkonda, kes on pannud kumbki ühe Roomba kolmeks roundiks võistlustulle. Ainult üks võistkonnaliige võib sumoringile läheneda, teised liikmed jäävad eemale publikuks. Liige asetab roboti ringile enda poole, ning kohtuniku märguande peale lülitavad nad robotid tööle ja peale viit sekundit pausi võivad robotid liikuma hakata. Nende sekundite jooksul peab võistleja ringi piirkonnast eemalduma. Matš lõppeb ja taasalustab kohtuniku märguannete peale.

Võistkondade robotid peavad olema nende endi poolt ehitatud kindlate reeglite järgimisel. Matš algab kohtuniku käsu peale ning kestab enamasti kuni 3 minutit, või kuni kohtunik teeb võitja ise selgeks.

Võistlusplatsiks on sumoring. See on kindlate mõõtmetega, musta matt värviga ringikujuline metallplaat, raadiusega 148cm. Selle plaadi äärtmööda jookseb 5cm paksune valge piirjoon.

Sumoringis on robotite eesmärgiks sensorite abiga tunda ära vastase roboti asukoht.
Võitmiseks peab üks robotitest suutma teise vajalikke manöövreid tehes üle valge piirjoone lükata ning ise tuleb kindlasti sumoringi sisse jääda. Siin maksab programmeerimine ning hõõrdejõud kõige enam, hiljem tulevad teised disainilised eripärad nagu sahad jne.

Robotite modifitseerimisel tuleb arvestada paljusid reegleid nagu näiteks:

- ei tohi olla kaugjuhitav
- ei tohi muuta mootorit, rattaid
- ei tohi muuta energiaallikaid – ainult originaalsed akud on lubatud
- roboti kaal ei tohi ületada esialgsest kaalust 30% (maximum 3.8kg)
- roboti mõõtmeid ei tohi suurendada, välja arvatud kõrgus
- tohib kasutada lisasensoreid

- tohib kasutada lisaelemente (mikrokontrollereid) või elektroonilisi elemente, mis pole teiste nõuetega vastuolus
- ebavajalikud osad võib eemaldada nt: tolmukoguja, mootoriga hari ei tohi kahjustada teist robotit ega sumoringi

Ideed

Kasutada kolme Sharp IR kaugus andurit vastase leidmiseks.

Kaks paikneksid roboti ees ja vaataksid paralleelselt ette, et saaksid liikumist korrigeerida vastavalt kauguste vahele ja üks taga, et vastast märgates oskaks, end ümber pöörata.

Kinnitada rattad nii, et keeramisel robot ei kerkiks ja püsiks parem haarduvus.

Raskus asetada keskele.

Kogu raskus oleks suunatud vedavatele ratastele, samuti haarduvuse parandamiseks.

Jälgida väljaku servi cliff sensoritega.

Roomba originaalid, ei mingit leiutamist. Kui pind alt kaob, kohe tagurdab.

Kasutada pamperit selleks, et ära tunda, kas vastane on vastas või mitte.

Sharp infrapuna kaugusmõõdikutega tunneb ära ja sõidab otse vastase poole. Kui pamper aktiivne läheb rünnakule ja vastavalt missugune pamperi osa aktiivne on, selle järgi korrigeerib asendit vastase suhtes, et robot oleks otse ees.

Kaal

Maksimum kaal- raske, aeglane, peab kiirelt vaenlase leidma,

Midagi vahepealset- parim kaalu-kiiruse suhe, katsetamine teiste TTÜ Robotiklubi roombadega

Minimum kaal- kiire, kiired manöövrid,

Strateegiad

Vaenlase otsimine- keerab kohapeal kuni näeb vähemalt ühe IR sensoriga teist robotit.

Vaenlase enda ette võtmine- kui vaenlane oli eelmises tsükklis vasakul, keerab veel natuke, kuni vaenlane mõlemas IR sensoris, kui enne oli paremal, keerab samamoodi natukene paremale.

Vaenlase rammimine- kihutab vaenlase poole, hoides teda enda ees, kui juba kontakt vaenlasega, käib edasine vaenlase suuna määramine mikrolülititega.

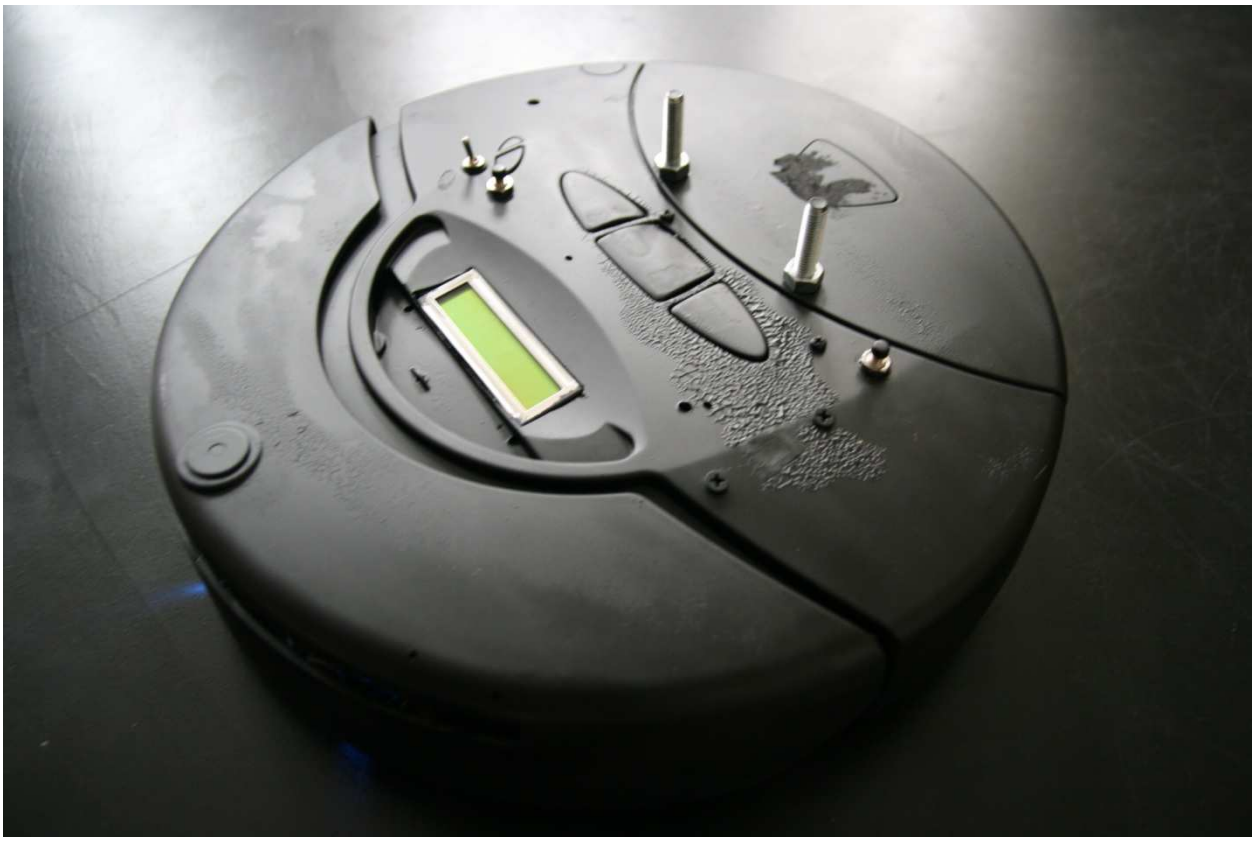
Ringis püsimine- kui tunnetab edasi minnes joont, sõidab tagurpidi ja keerab vastavalt selles suunas, kummal pool väljaku lõppu nähti: kui äär paremal, siis robot keeras vasakule ja vastavalt teistpidi.

Kavalus- kui vaenlane rammib ja näeb tagant joont, keerab end ümber joonest kaugemal oleva ratta ja rammib vaenlast.

Kaalusime ka tagumise ratta panekut, kuid õnneks ei teinud seda, ilma rattata tuli hiljem kasuks.

Väljanägemine

Roomba värvida matt musta värvi, LCD ekraan peal, 3W *side emitting power LED* põhjavalgustus all (sinine).



Spetsifikatsioon

Roboti kirjeldus

Roombalt sai algselt ära võetud osad,mis täitsid ainult koristamisülesannet, nagu harjased põhja alt.

Ees pamperis on 12mm laiune ja 250mm pikkune pilu Sharp anduritele.

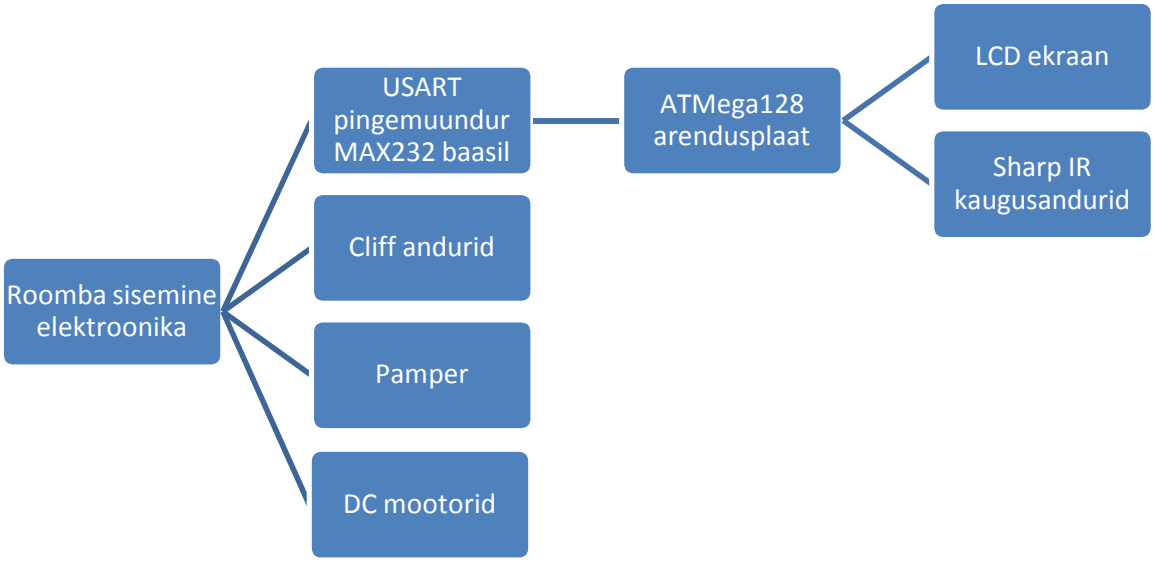
Vedavavatelt ratastelt on eemaldatud vedrud ja rattad fikseeritud kindlalt kere külge, et manööverdamisel kumbki külg kerkida ei saaks, sama on tehtud ka esimese tugirattaga. Taha ratast ei paigaldatud.

Algselt eemaldati ka, roomba originaal nupud, et saaks ruumi ekraani jaoks, mis asub eemaldatud käepideme süvikus. Ekraani oli tarvis andurite kalibreerimiseks ja juhtpogrammi mõistmiseks. Plaan oli üldse nuppude plaadist lahti saada, aga katsetamise käigus tuli välja, et asi ei toiminud ilma, mistõttu hiljem asus nuppude plaat põhja all. Start ja Clean nupp oli väja toodud nupplülitega korpuse pinnale. Kumminupud jäid, kuid need on ainult augu täiteks.

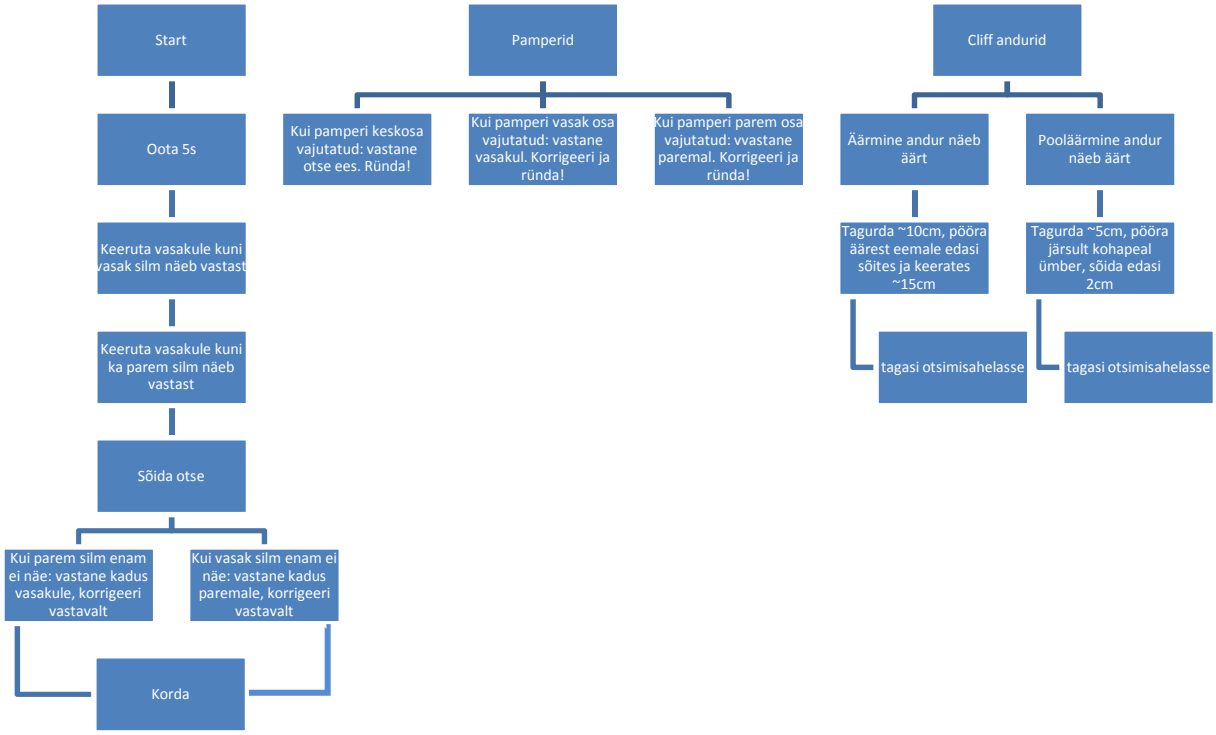
Tolmukast sai kinnitatud kahe M8 poldiga, plaan oli paigaldada raskus nende poltidega põhja alla, aga jäi ära, robot osutus liiga raskeks. Selle asemel lisasime väiksema massiga raskused topelt teibiga korpuses olevasse tühimikku ja tolmukasti.

Öö enne võistlus värviti robot mustaks.

Elektroonika üldskeem



Algoritmi selgitus



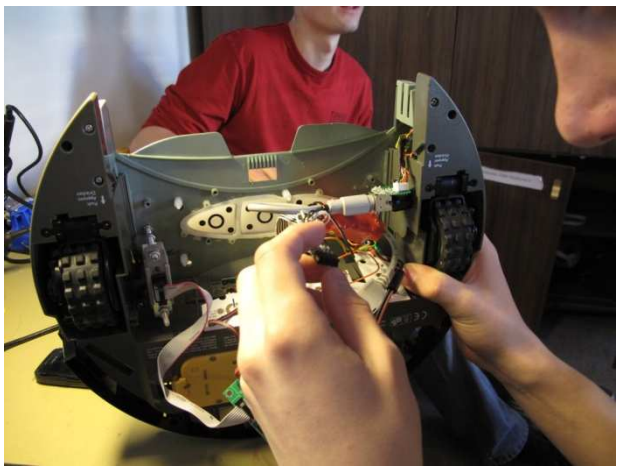
Programmikoodi tähtsamad osad

Näide programmi ründamise algoritmist:

```
if(state & ATTACK)
{
	lcd_clrscr();
	lcd_puts("Ryndan");
	roomba.Drive(TURBO, OTSE);
	timeout = clock.GetTimestamp();

	if(LeftDistance < VASAK_OLEMAS)
	{
		lcd_clrscr();
		lcd_puts("Kuhu kadus?\n");
		lcd_puts("Paremale");

		roomba.Drive(SPEED, PAREM);
		state = OTSING_PAREMALT;
	}
	else if(RightDistance < PAREM_OLEMAS)
	{
		state = KADUND_VASAKULE;
		lcd_clrscr();
		lcd_puts("Kuhu kadus?\n");
		lcd_puts("Vasakule");

		roomba.Drive(SPEED, VASAK);
		state = OTSING_VASAKULT;
	}
}
```

Fotod

Pilt 1 - Eestvaade



Pilt 3 - Altvaade. Ehitusprotsess

Pilt 2 – Pealtvaade

Pilt 4 – Võistlusmoment (Putting, 2009)

Kokkuvõte

Roomba oli algul ainult robottolmuimeja, mis suutis ainult seinapõrkest aru saada, et takistus on ees. Peale mõningast pusimist saime talle 2 IR sensorit ette vaatama ja ka väljanägemise saime sellel korda. Võistlusel oli meie Roomba ainuke ilma sahata Roomba mis suutis sahaga Roombale ära teha, põhjuseks oli tagumise tugiratta puudumine. Kahjuks oli meie algne kaaluseadistus natukene mööda, sest võistlusväljak oli teine, kui see milliel harjutasime, sellest ka esimene kaotus, kuid peale kaalu tõstmist näitas meie Roomba end tõelise võitjana. Kohaks oli 5-6, mis tegelikult oleks ka võinud parem olla, kui ainult oleksime osanud arvestada et värskelt värvitud pinnal on haakuvus väga palju parem. Eks igast eksimusest õpib tulevaste võistluste jaoks, seepärast arvan, et võistlusega võib rahule jääda.

Lisa 1. Põhialgoritmi lähtekood.

```
/*****************************************************************************\

        Sumo Roomba

\*****************************************************************************/

#include "common/Main.h"
#include "LCD.c"

#define object_modifier extern
#include "common/HardwareObjects.h"
#undef object_modifier

/**
 * Robot main program.
 **/

#define SPEED   300                             // sõidukiirus
#define TURBO   512                             // rammimiskiirus

#define VASAK_OLEMAS   80
#define PAREM_OLEMAS   20

#define OTSING_VASAKULT         1
#define OTSING_PAREMALT         2
#define MAYBE_VASAKUL  4
#define MAYBE_PAREMAL  8
#define ATTACK                  16
#define KADUND_VASAKULE         32
#define KADUND_PAREMALE         64


void Program::Run()
{

        roomba.Init();


        uint16_t LeftDistance;
        uint16_t RightDistance;

        uint64_t timeout = 0;
        uint64_t timer = 0;

        uint8_t state = OTSING_VASAKULT;

        roomba.LED.Clean(0);
        roomba.LED.Spot(0);
        roomba.LED.Max(0);

        lcd_init(LCD_DISP_ON_CURSOR_BLINK);
        roomba.LED.Clean(1);
        lcd_clrscr();
        roomba.LED.Spot(1);
        lcd_puts("LCD Test Line 1\n");
        roomba.LED.Max(1);

        //clock.Delay(1000);
        roomba.LED.Clean(0);
        roomba.LED.Spot(0);
        roomba.LED.Max(0);


        lcd_clrscr();
        lcd_puts("Valmis. Mollame\n VAJUTA!");

        while(true)
        {
                roomba.ReadSensors(SUBSET_2);
                roomba.LED.DirtDetect(1);
                if(roomba.Sensors.Max)
                {
                        break;
                }
        }

        clock.Delay(5000);

        roomba.Drive(SPEED, VASAK);
        timeout = clock.GetTimestamp();
```

```
        while (false)
        {
                roomba.FullMode();
                roomba.ReadSensors(SUBSET_ALL);
                // vasakust IR andurist loetud kaugus
                LeftDistance = irLeft.GetCorrectedRawDistance(10);
                RightDistance = irRight.GetCorrectedRawDistance(10);

                char Left[6], Right[6];

                itoa(LeftDistance, Left, 10);
                itoa(RightDistance, Right, 10);

                lcd_clrscr();
                lcd_puts("Left: ");
                lcd_puts(Left);
                lcd_puts("\nRight:");
                lcd_puts(Right);
        }

        while (true)
        {
                roomba.FullMode();
                roomba.ReadSensors(SUBSET_ALL);
                // vasakust IR andurist loetud kaugus
                LeftDistance = irLeft.GetCorrectedRawDistance(10);
                RightDistance = irRight.GetCorrectedRawDistance(10);

                timer = clock.GetTimestamp();

                char Left[6], Right[6];

                itoa(LeftDistance, Left, 10);
                itoa(RightDistance, Right, 10);

                if(roomba.Sensors.CliffFrontLeft &&
(roomba.Sensors.BumpRight||(roomba.Sensors.CliffFrontRight&&roomba.Sensors.BumpLeft)))
                {
                        roomba.Drive(-TURBO,OTSE);
                        clock.Delay(500);
                        roomba.Drive(TURBO,VASAK);
                        clock.Delay(100);
                        state = OTSING_PAREMALT;
                        roomba.Drive(300,PAREM);
                }

                else if(roomba.Sensors.CliffFrontRight &&
(roomba.Sensors.BumpLeft||(roomba.Sensors.CliffFrontRight&&roomba.Sensors.BumpLeft)))
                {
                        roomba.Drive(-TURBO,OTSE);
                        clock.Delay(500);
                        roomba.Drive(TURBO,PAREM);
                        clock.Delay(100);
                        state = OTSING_VASAKULT;
                        roomba.Drive(300,VASAK);
                }

                else if(roomba.Sensors.CliffFrontLeft)
                {
                        roomba.FullMode();
                        lcd_clrscr();
                        lcd_puts("Cliff Left");
                        roomba.Drive(-TURBO, OTSE);
                        clock.Delay(500);
                        state = OTSING_PAREMALT;
                        roomba.Drive(SPEED, PAREM);
                }

                else if(roomba.Sensors.CliffLeft)
                {
                        roomba.FullMode();
                        lcd_clrscr();
                        lcd_puts("Cliff Left");
                        //roomba.Drive(-TURBO, OTSE);
                        //clock.Delay(300);
                        roomba.Drive(-TURBO, -129);
                        clock.Delay(500);
                        roomba.Drive(-TURBO, 100);
                        clock.Delay(100);
```

```
                state = OTSING_PAREMALT;
                //roomba.Drive(SPEED, PAREM);
        }

        else if(roomba.Sensors.CliffFrontRight) //
        {
                roomba.FullMode();
                lcd_clrscr();
                lcd_puts("Cliff Right");
                roomba.Drive(-TURBO, OTSE);
                clock.Delay(500);
                state = OTSING_VASAKULT;
                roomba.Drive(SPEED, VASAK);
        }

        else if(roomba.Sensors.CliffRight)
        {
                roomba.FullMode();
                lcd_clrscr();
                lcd_puts("Cliff Right");
                //roomba.Drive(-TURBO, OTSE);
                //clock.Delay(1200);
                roomba.Drive(-TURBO, 160);
                clock.Delay(800);
                roomba.Drive(-TURBO, -129);
                clock.Delay(100);
                state = OTSING_PAREMALT;
                //roomba.Drive(SPEED, VASAK);
        }



        roomba.FullMode();
        if(roomba.Sensors.BumpRight&&roomba.Sensors.BumpLeft)
        {
                state = ATTACK;
        }

        else if(roomba.Sensors.BumpRight )//&& !(state & ATTACK)
        {
                roomba.Drive(-SPEED, OTSE);
                roomba.Drive(-TURBO, VASAK);
                state = OTSING_PAREMALT;
        }

        else if(roomba.Sensors.BumpLeft )//&& !(state & ATTACK)
        {
                roomba.Drive(-SPEED, OTSE);
                roomba.Drive(-TURBO, PAREM);
                state = OTSING_VASAKULT;
        }

        roomba.FullMode();
        if((state & OTSING_VASAKULT)&&((timeout-1000) <
timer)&&!(roomba.Sensors.BumpRight||roomba.Sensors.BumpLeft))
        {
                lcd_clrscr();
                lcd_puts("Otsin vasakult\n");
                lcd_puts(Left);
                lcd_puts("  ");
                lcd_puts(Right);
                if(LeftDistance > VASAK_OLEMAS)
                {
                        state = MAYBE_VASAKUL;
                        lcd_clrscr();
                        lcd_puts("Maybe vasakul\n");
                        lcd_puts(Left);
                        lcd_puts("  ");
                        lcd_puts(Right);
                }
                else
                        state = OTSING_VASAKULT;
        }

        else if((state & OTSING_PAREMALT)&&((timeout-1000) < timer))
        {
                lcd_clrscr();
                lcd_puts("Otsin paremalt\n");
                lcd_puts(Left);
                lcd_puts("  ");
```

```
                        lcd_puts(Right);
                        if(RightDistance > PAREM_OLEMAS)
                        {
                                state = MAYBE_PAREMAL;
                                lcd_clrscr();
                                lcd_puts("Maybe paremal\n");
                                lcd_puts(Left);
                                lcd_puts("  ");
                                lcd_puts(Right);
                        }
                        else
                                state = OTSING_PAREMALT;
                }

                else if(((RightDistance > PAREM_OLEMAS)&&(state &
MAYBE_VASAKUL))||((LeftDistance > VASAK_OLEMAS)&&(state & MAYBE_PAREMAL)))
                {
                        state = ATTACK;
                }


                if(state & ATTACK)
                {
                        lcd_clrscr();
                        lcd_puts("Ryndan");
                        roomba.Drive(TURBO, OTSE);
                        timeout = clock.GetTimestamp();

                        if(LeftDistance < VASAK_OLEMAS)
                        {
                                lcd_clrscr();
                                lcd_puts("Kuhu kadus?\n");
                                lcd_puts("Paremale");

                                roomba.Drive(SPEED, PAREM);
                                state = OTSING_PAREMALT;
                        }
                        else if(RightDistance < PAREM_OLEMAS)
                        {
                                state = KADUND_VASAKULE;
                                lcd_clrscr();
                                lcd_puts("Kuhu kadus?\n");
                                lcd_puts("Vasakule");

                                roomba.Drive(SPEED, VASAK);
                                state = OTSING_VASAKULT;
                        }
                }

                roomba.FullMode();
        }

        while(true)                                             // võistlus läbi!
        {

        }
}
```