Tallinna Tehnikaülikool
Mehhatroonikainstituut

Mikrokontrollerid ja praktiline robootika (MHK0011) „Teeme ise 2011“ Aruanne

Joan Lääne

25.05.2011

Tallinn 2011

Sisukord

1. Sissejuhatus

Paar aastat tagasi (tegelikult juba varem) tekkis huvi selle vastu, kuidas koos tööle panna hammasrattad ja arvuti. Tavaline raadioteel juhitav auto ei olnud enam piisavalt põnev, tahtsin teha nii, et auto teeks ka midagi oma „peaga“. Enne ülikooli tulekut ei õnnestunud mul see.

Kui ma sain teada, et TTÜ Robotiklubi hakkab kevadel läbi viima koolitust, kus tuleb juttu praktilisest robotite programmeerimisest, teadsin, et see on mu võimalus teha seda, mis on huvi pakkunud juba aastaid. Sellepärast valisingi selle vabaaine.

2. Robotist endast

Iga tunni algul saime grupi peale (3 inimest) ühe Pololu 3π roboti ja selle programmeerija (SparkFun PocketProgrammer). Robot ise on väike, ümbermõõtu on tal ainult 3π –d, nagu nimigi seda ütleb. Robotil on kaks mootorit, mille otsas on rattad. Lisaks mootoritele on ka ekraan, buzzer (genereerib erinevaid noote ja sagedusi) , mõned nupud, IR - sensorid ja LED-id. Kõik need on programmeeritavad läbi C – keele või assembleri. Meie keskendusime koolituse vältel rohkem C- keelele, kuigi natuke õppisime ka assemblerit.

3. Lühike ülevaade õpitust

Kursuse vältel jõudsime õppida päris palju. Peale sissejuhatavaid loenguid oli esimeseks ülesandeks lihtne LED-i vilgutamine. Kuigi see oleks olnud ühe rea ülesanne Pololu teekidega, realiseerisime meie seda aga riistvaralähedase programmeerimisega. Nimelt pidime uurime andmelehelt kuhu on ühendatud antud LED, siis kirjutama koodi, mis tegeles registrite muutmisega. Näiteks LED-i vilgutamine seisnes selles, et oli vaja vastav LED-i viik (ingl. k *pin*) muuta kord madalaks ja kord kõrgeks.

Lisaks selle uurisime ka kuidas töötavad erinevad sensorid 3π robotil. Roboti all on olemas rida peegeldussensoreid (*reflectance sensor*). Sellega sai uurida, kas pind on hele või tume. Esimene katse, mis meie grupp sellega tegi, oli see, et robot pidi jääma musta joone ees seisma. Natuke muudatust koodis ja järgmisel hetkel juba robot pendeldas kahe musta joone vahel. Seda oli juba päris lõbus vaadata võrreldes lihtsa LED-i vilgutamisega.

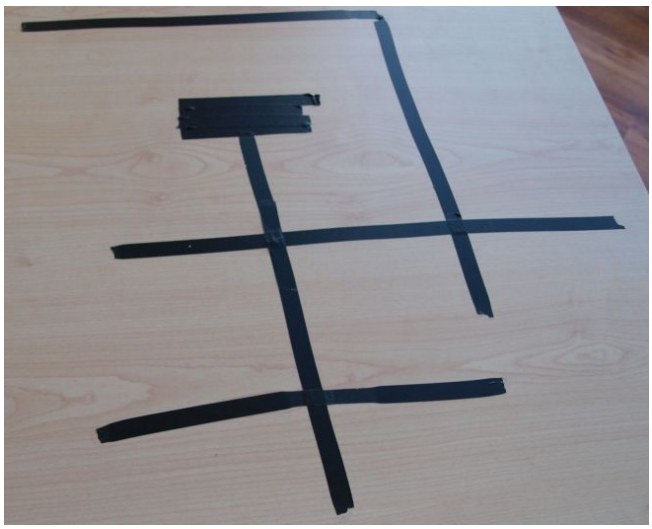
Ühes tunnis tutvusime ka infrapuna kaugussensoriga Sharp. See kujutas endast kahte IR-LED-i, kus üks saatis signaali välja ja teine proovis selle kinni püüda. Kui signaal jõudis tagasi, oli mingi objekt kiire ees. Sensori testimiseks kirjutasime sellise koodi, kus robot jäi 10 cm enne takistust seisma.

Järgmises tunnis selgitati meile PID – kontrolleri põhimõtet ja joonejärgimist. PID – kontroller koosneb kolmest komponendist (*Proportional, Integral, Derivative*), millega saab juhtida mootorite kiirusi. Kui robot hakkab üle joone minema, siis arvutatakse välja viga ja PID kontroller korrigeerib

mootorite kiirusi nii, et robot püsiks joonel. PID – konstantite leidmine oli natuke tülikas, aga see tasus end ära. Tulemuseks oli see, et 3π robot oskas juba kenasti joonel sõita.

4. Lõpuülesanne - labürindi läbimine

4.1 Ülesande püstitus



Ühes viimastes tundides saime teada oma viimase ülesande – labürindi läbimine. Oli vaja kirjutada kood, mis aitaks robotil labürindi edukalt läbida. Alguseks oli siis suvaline labürindi punkt ja lõpuks oli must ristkülik (*Pilt paremal*, Allikas: http://www.robotiklubi.ee/kursused/teeme_ise/2011).

4.2 Esimesed ideed

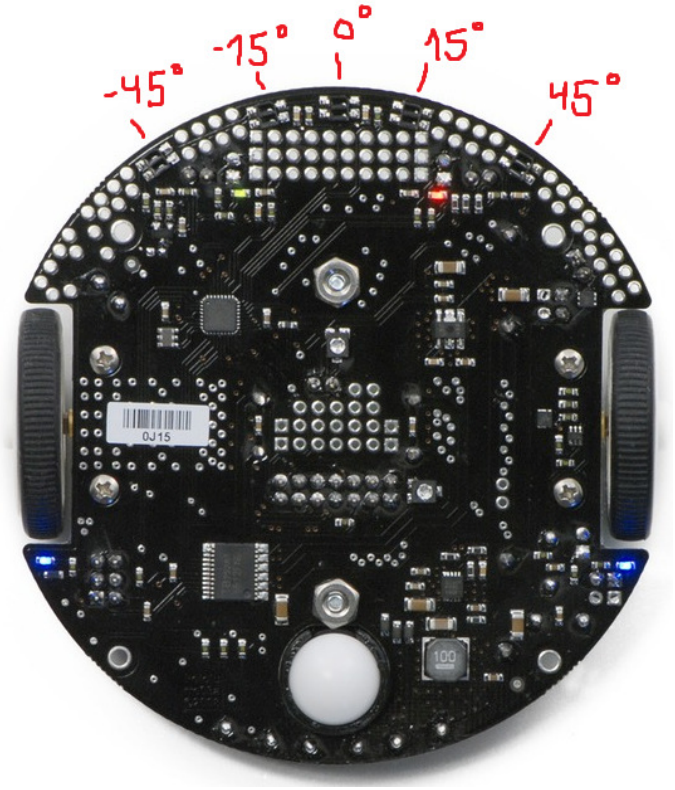
Alustada soovitati sellest, et kirjutada kood, millega robot jääks ristmikul seisma. Mõeldud – tehtud. Hiljem täiendasime seda nii, et robot oskaks ka kuhugi pöörata. Nimelt valisime labürindi läbimise strateegiaks „vasak käsi seinal", ehk saabudes ristmikule, pöörab robot alati vasakule. Pärast väikest muudatust nägi see koodijupp välja selline:

```
void follow_segment()
{
    int16_t speed_dif;
    while(1)
    {
            linesensors_read();
            speed_dif = pid();
            drive(60, speed_dif);
            clear();
            print("Forwd");

        if(linesensor[1] < 100 && linesensor[2] < 100 && linesensor[3] < 100)
            {
                  // Tupik, pööra 180 kraadi
            clear();
            print("D_end");
                  return;
            }
            else if(linesensor[0] > 200 || linesensor[4] > 200)
            {
                  // Ristmik, pöörame 90 kraadi vasakule
            clear();
            print("Ristm");
                  return;
            }
    }
```

4.3 PID –i integreerimine

Joonis 1. Sensorite paigutus



Nagu näha, siis eelmises funktsioonis kasutan alamfunktsiooni *pid();* . See on pärit sellest tunnist, kui õppisime PID-i. Kuna see oli olemas, siis selle saime lihtsalt sisse kopeerida. PID hoolitseb selle eest, et robot oleks mustal joonel ja ei kalduks rajast kõrvale.

Funktsioon *pid()* hoolitseb selle eest, et arvutab välja igal ajahetkel uued P, I ja D konstantide väärtused. *Pid()* ise kasutab veel alamfunktsiooni *error()* , mis tegeleb vea arvutamisega. Idee seisneb seal selles, et leitakse sensor, mis on kõige tumedam. See tagastab selle sensori nurga. Roboti all on 5 QTR Reflectance sensorit paigutatud nii, et nad on erinevate nurkade all (Joonis 1 (Allikas: http://www.pololu.com/picture/view/0J828)) Selle teabega saab arvutada vea välja ning funktsioonile *pid()* edasi anda.

```
int8_t error()
{
	uint8_t i;
	uint8_t	darkest = 0;
	int8_t angles[] = {45, 15, 0, -15, -45};

	for (i=1; i<5; i++)
	{
		if (linesensor[i] > linesensor[darkest])
		darkest = i;
	}

	return angles[darkest];
}

int16_t pid ()
{
	proportional = (float) error();
	integral += proportional;

	derivative = proportional - last_proportional;
	last_proportional = proportional;
	//arvutame uued Kp, Ki, Kd väärtused välja
	return (Kp*proportional + Ki*integral + Kd*derivative);
}
```

4.4 Lahendusalgoritm

Main funktsioonist lähme kohe üle alamfunktsiooni *m_solve()* , milles toimub lahenduse otsimine. Esiteks sõidab robot senikaua otse, kuni tuleb ristmik. Ristmik on siis, kui kumbki kõige äärmistest sensoritest on must. Edasi teeb 3π kindlaks, mis tüüpi ristmik on ja kuhu saab pöörata. Ristmiku tüüp olemas, sõidame veel nii palju edasi, et robot oleks täpselt ristmiku keskel. Nüüd kontrollime veel kahte olukorda: esiteks, kas on võimalus edasi sõita ja teiseks, kas oleme juba lõpuristkülikul.

Kui kõik andmed olemas, jääb üle ainult pöörata õigele poole (eelistatult vasakule), kui selleks vajadus.

```
void m_solve()
{
      // Tsüklis, kuni labürint on läbitud
      while(1)
      {
            follow_segment();
        //Sõidame natuke otse
            set_motors(50,50);
            delay_ms(50);

        //Muutujad, mis peavad järge, kus poolt on joont näha
            unsigned char found_left=0;
            unsigned char found_straight=0;
            unsigned char found_right=0;

            // Loeme sensoreid ja kontrollime ristmiku tüüpi
            unsigned int linesensor[5];
            linesensors_read();

            // Kontrollime, kas saab vasakule - paremale pöörata
            if(linesensor[0] > 100)
                  found_left = 1;
            if(linesensor[4] > 100)
                  found_right = 1;

            // Sõidame natuke veel edasi, piisavalt, et ristmiku keskele end seada.
            set_motors(40,40);
            delay_ms(200);

            // Kontrollime, kas saab otse sõita
            linesensors_read();
            if(linesensor[1] > 200 || linesensor[2] > 200 || linesensor[3] > 200)
            clear();
            print("Otse");
                  found_straight = 1;

            // Kontrollime, kas oleme laburindi lõpus
            // S.t kui kõik 3 keskmist sensorid on tumedad,
            // siis oleme lõpuristküliku peal
            if(linesensor[1] > 600 && linesensor[2] > 600 && linesensor[3] > 600)
            clear();
            print("SOLVED");
            while(1);
                  break;

            unsigned char dir = select_turn(found_left, found_straight, found_right);

            //Pöörame, eelistatult vasakule.
```

```
            turn(dir);
        }
}
```

5. Kokkuvõte

Kokkuvõtteks oskan TTÜ Robotiklubi mikrokontrollerite koolituse kohta ainult head öelda. Tunnid olid sisukad – oli nii kuulamist, kirjutamist klaviatuuril kui ka nuputamist. Koduülesanded aitasid kinnistada ja taaskasutada tunnis olnud teemasid.

Antud kursus motiveeris mind veel nii palju, et otsustasin ka ise endale mikrokontrolleri ja paar sensorit osta. Eks varsti näen ka kodus midagi joonel sõitmas või midagi veel põnevamat tegemas. Sest kontrollerit ostes mõtlesin nii – võimalused on lõputud. Kõik, mida suudan välja mõelda peaks saama ka teostada.

Olen rahul, et valisin selle vabaine. See aitas mul alustada tegelemist sellega, millest olin unistanud juba ammu.