

SVEUČILIŠTE U ZAGREBU



FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA



ZAVOD ZA ELEKTRONIČKE SUSTAVE I OBRADBU INFORMACIJA



LABORATORIJ ZA SUSTAVE I SIGNALE



PROJEKT IZ KOLEGIJA
PROGRAMSKA PODRŠKA MJERNIH I PROCESNIH SUSTAVA

Računalno upravljanje Mindstorms® NXT robotom

Tehnička dokumentacija

Kristina Bashota
Tibor Čordaš
Dalibor Jelača
Iva Jelenčić
Vedran Koruga
Damir Kušević

Predrag Pale
mentor

Zagreb, 06.02. 2008.

Sadržaj

Uvod	3
Opis komunikacije i rada robota	4
Osposobljavanje <i>web</i> sučelja	5
Prilog	6

Uvod

U svrhu izrade ovog projekta korišten je robot *Mindstorms*[®] *NXT*.

Kao dodaci za navedeni robot koriste se: ultrazvučni senzor, tipkalo, zvučni senzor i otvaranje i zatvaranje robotovih kliješta.

Upravljanje robotom programira se u programskom jeziku C++. Za to je korišten *Bloodshed Dev-C++ compiler* koji je besplatan i preuzet sa <http://www.bloodshed.net/devcpp.html>. Za jednostavnije programiranje preuzete su i već napravljene biblioteke sa <http://www.norgsgade14.dk/bluetoothlibrary.php>.

S robotom je preko *Bluetootha* moguće prilično jednostavno komunicirati. Pomoću C++ datoteka naredbe se direktno šalju robotu. Nakon uspostave veze potrebno je ustanoviti koji *COM* port robot koristi.

C++ datoteke omogućuju direktno slanje naredbi preko *Bluetootha*. Postoji 6 klasa koje omogućuju:

- uspostavljanje i prekidanje veze preko *Bluetootha*,
- kontrolu nad motorima,
- isčitavanje vrijednosti senzora,
- imenovanje *Bricka* i čitanje *firmware-a*,
- reprodukcija zvučnih zapisa i
- korištenje ultrazvučnog senzora.

Web sučelje koje je dizajnirano omogućava upravljanje robotom. Korisniku je omogućen izbor radnje robota (vožnja naprijed / nazad, zakretanje lijevo / desno, otvaranje / zatvaranje kliješta). Podaci dobiveni sa senzora pregledno se prikazuju na zaslonu računala, kako bi na temelju tih poruka korisnik mogao odlučiti o svom slijedećem koraku. Putem *web* sučelja omogućeno je da korisnik s udaljenog računala upravlja robotom koji komunicira sa serverskim računalom.

Opis komunikacije i rada

Bluetooth veza

COM port za *Bluetooth* komunikaciju otvara se funkcijom *int connect(char *port)*, a zatvara funkcijom *disconnect()*.

Upravljanje motorima

Robot ima 3 motora kojima upravljamo pomoću slijedećih funkcija. *Motor(int output_port, Serial *cp)* predaje broj porta i pokazivač serijskoj vezi. Funkcija *int on(int speed)* pokreće motor zadanom brzinom. Ako se doda predznak minus, motor se vrti u suprotnom smjeru. Funkcija *int stop()* zaustavlja motor, dok *int reset_rotation()* resetira rotaciju motora.

Brzina kretanja robota naprijed i natrag fiksirana je na vrijednost 35 (odabir brzine kretanja je proizvoljan, a raspon vrijednosti brzina je od 0 do 100). Brzina okretanja ulijevo i udesno smanjena je na četvrtinu fiksne vrijednosti radi korisnikovog jednostavnijeg upravljanja. Naredbe kretanja robota koje zadajemo s *web* sučelja zapisuju se u tekstualnu datoteku *upravljanje.txt* koju stvara sam program.

Upravljanje senzorima

Podatke koji su primljeni sa ultrazvučnog senzora i tipkala potrebno je proslijediti računalu koje će te podatke prikazati na *web* sučelju.

U beskonačnoj *while* petlji provjerava se stanje ultrazvučnog senzora. Pozivom funkcije *sonar.distance()* tražimo od robota da mjeri udaljenost do prepreke te se taj podatak *Bluetooth* vezom šalje računalu na daljnju obradu. (Navedena funkcija može vraćati vrijednost u centimetrima ili inčima.) Taj podatak zapisuje se u tekstualnu datoteku *udaljenost.txt* koju će stvoriti glavni program i koja je veza sa *web* sučeljem.

U navedenoj beskonačnoj petlji provjerava se i stanje tipkala. Tipkalo služi kao detektor niskih prepreka kao što su kablovi, knjige itd. Pozivom funkcije *touch.read()* tražimo da se očitava stanje tipkala, tj. kada tipkalo naiđe na prepreku ono se uključi i *Bluetooth* vezom računalu šalje poruku da je tipkalo uključeno. U trenutku otpuštanja tipkala računalu se šalje poruka da je tipkalo isključeno. Stanje tipkala također se zapisuje u tekstualnu datoteku *tipkalo.txt* koja je veza sa *web* sučeljem.

Upravljanje ostalim dodacima

Omogućeno je puštanje zvučnog signala naredbom *speaker.beep()*.



Otvaranje i zatvaranje kliješta omogućeno je koristeći naredbe *motorA.reset_rotation()* i *motorA.on(10,45)* (gdje 45 kut, a 10 brzina).

Osposobljavanje *web* sučelja

Nakon što je u nekom *html editoru* napravljeno sučelje *web* stranice, za osposobljavanje *online* komunikacije s robotom potrebno je pratiti slijedeće korake:

- Skinuti i instalirati *apache web server* (verzija 2.2.8). Adresa s koje skidate jest http://ftp.carnet.hr/misc/apache/httpd/binaries/win32/apache_2.2.8-win32-x86-no_ssl.msi

Instalacija je jednostavna, samo odabirati *next*.

- Skinuti i instalirati *php*. Adresa je

<http://www.php.net/get/php-5.2.5-win32-installer.msi/from/a/mirror>

Instalacija je jednostavna, samo odabirati *next* i u jednom trenutku će pitati za koji program da instalira modul. Tu samo označiti *apache 2.2.x*. Također, tražit će adresu gdje je konfiguracijska datoteka *apachea*, ona se nalazi u direktoriju *conf* u direktoriju gdje je instaliran *apache* (samo treba dati lokaciju *conf* direktorija).

- Kopirati direktorij s *web* stranicom unutar *htdocs* direktorija koji se nalazi u *apache* direktoriju. U *mstore/tmp* direktorij kopirati i pokrenuti glavni program za upravljanje robotom.

U ovom primjeru *html* stranica nazvana je *index.html* i spremljena u direktorij *mstore* sa svim pripadajućim potrebnim datotekama (tekstualne datoteke i *jpg* slike ikona *web* sučelja) da bi sučelje ispravno radilo. Kad se želi pristupiti stranici spojiti se na *localhost/mstore* sa lokalnog računala odnosno *IP-adresa/mstore* sa udaljenog računala.

Prilog

Dokumentaciji je priložen i komentirani kôd koji slijedi.

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <conio.h>
#include <stdlib.h>

#include "serial.hpp"
#include "brick.hpp"
#include "sound.hpp"
#include "motor.hpp"
#include "sonar.hpp"
#include "sensor.hpp"

Serial bluetooth;
Sonar sonar = Sonar(0,&bluetooth); //ultrazvučni senzor je na portu 1
Sensor touch = Sensor(1,&bluetooth); //tipkalo je na portu 2
Brick nxt = (&bluetooth);
Sound speaker = (&bluetooth);

Motor motorA = Motor(0,&bluetooth);
Motor motorB = Motor(1,&bluetooth);
Motor motorC = Motor(2,&bluetooth); //Motor C na portu 3
Sound zvuk;

using namespace std;
int main(int argc, char *argv[]){
    FILE *fp, *fp1, *fp2;

    int brz;
    int naredba; //upravljanje robotom
    int j,k,l; //zastavice za datoteke
    int i,x,y; //zastavica za naredbe
    int distance,tipkalo;
    char string [4];

    if(bluetooth.connect("COM6")){//Spaja se na NXT kroz com port 6
        cout << "Veza OK!" << endl;
        nxt.set_name("LEGO"); //Set the brickname to LEGO
        cout << "Firmware verzija: " << nxt.firmware_version() << endl;
```

```

touch.type_and_mode(TOUCH,BOOL_MODE); //inicijaliziranje tipkala
sonar.setup(); //inicijaliziranje ultrazvučnog senzora
//inicijalizacija zastavica i varijabli
brz=35;
naredba=13;
i=12;
j=0;
k=0;
l=0;
x=1;
y=1;
distance=0;
tipkalo=0;

// cout << "Postavite željenu brzinu!" << endl;
// cin >> brz;

while (naredba != 1){
    Sleep(10);
    //ucitavanje naredbe kroz datoteku "upravljanje.txt"
    if ((fp = fopen("upravljanje.txt", "r+")) !=NULL){ //kreiranje datoteke
        j=1;
    }

    if(j){

        naredba=fgetc(fp);

        fclose(fp); //zatvori datoteku
        fopen("upravljanje.txt", "w");
        fclose(fp);
        j=0;
    }

    //upisivanje udaljenosti u datoteku "udaljenost.txt"
    if ((fp1 = fopen("udaljenost.txt", "w"))!=NULL){//kreiranje datoteke
        k=1;

    }

    if(k){
        itoa(distance,string,10);
        fputs(string,fp1); //upisi podatak o udaljenosti u datoteku
        fclose(fp1); //zatvori datoteku
        k=0;
    }
}

```



```

}

//upisivanje vrijednosti tipkala u datoteku "tipkalo.txt"
if ((fp1 = fopen("tipkalo.txt", "w")) !=NULL){ //kreiranje datoteke
    l=1;
}

if(l){
    tipkalo=touch.read();

    fputc(tipkalo,fp1); //upisi vrijednost tipkala
    fclose(fp1); //zatvori datoteku
    l=0;
}

// upravljački i senzorski odlomci:

distance=sonar.distance(); //ocitavanje udaljenosti

//    if (distance < 35){
//        motorB.stop();
//        motorC.stop();
//    }
naredba=naredba-48;
if (naredba == 8){ //oba motora naprijed
    motorC.on(brz);
    motorB.on(brz);
}

if (naredba == 5){ //oba motora zaustavljena
    motorC.stop();
    motorB.stop();
}

if (naredba == 2){ //oba motora nazad
    motorC.on(-brz);
    motorB.on(-brz);

}

if (naredba == 4){ //kretanje u lijevo
    motorC.on(-brz/4);
    motorB.on(brz/4);

}

```

```

if (naredba == 6){ //kretanje u desno
    motorC.on(brz/4);
    motorB.on(-brz/4);

}

if (naredba == 0){ //zvučni signal
    speaker.beep(200);

}

if (naredba == 7){ //otvaranje kliješta
    if(x){
        motorA.reset_rotation();
        motorA.on(10,45);
        x=0;
        y=1;
    }
}

if (naredba == 9){ //zatvaranje kliješta
    if(y){
        motorA.reset_rotation();
        motorA.on(-10,45);

        y=0;
        x=1;
    }
}

if (naredba == 17){ //kretanje udesno
    motorC.on(brz/2,30);
    motorB.on(-brz/2,30);

}

if (naredba == 18 ){ //kretanje ulijevo
    motorC.on(-brz/2,30);
    motorB.on(brz/2,30);

}

if (naredba == 19 ){ //kretanje naprijed
    motorC.on(brz/2,135);
    motorB.on(brz/2,135);

}

}

```

```
    bluetooth.disconnect();//Odpaja se bluetooth
  }
  else{
    cout << "Bluetooth veza nije uspostavljena!" << endl;
  }

  Sleep(200);
  exit(1);
}
```