

Inteligenten Robotiziran Izvrševalec – F.I.R.E.

(Fully Intelligent Robotized Executer)

Uvod

Letos maja je bilo na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru že drugo tekmovanje z mobilnimi roboti. Seveda sem se ga kot lanski veteran ponovno udeležil, kajti pri konstrukciji lastnega robota se naučiš ogromno. Seveda je v meni tlela tudi neustavljiva želja po zmagi, kar je na koncu koncev tudi želja vsakega tekmovalca.



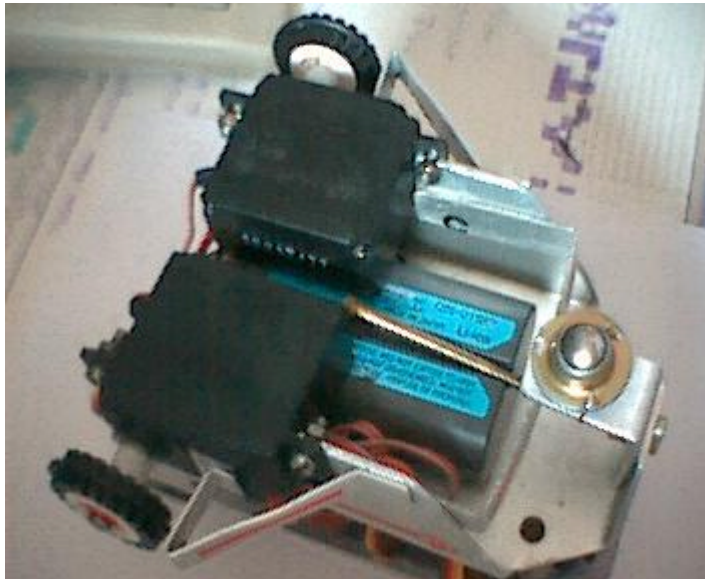
Slika 1: Robotek FIRE v vsej svoji lepoti

Ideja

Cilj tekmovanja je, da samostojen robot lastne konstrukcije čimprej pride iz labirinta. Pri tem mu ne sme direktno pomagati človek. Preden sem se lotil same konstrukcije robota, sem pri sebi razmislil, kako bi to nalogo rešil človek. Predpostavil sem, da so stene gladke in razdalje identične. Ugotovil sem, da pri prvem sprehodu skozi labirint ni druge možnosti, kot da človek sledi nekemu zidu. Točno to idejo sem nato implementiral na robota.

Konstrukcija

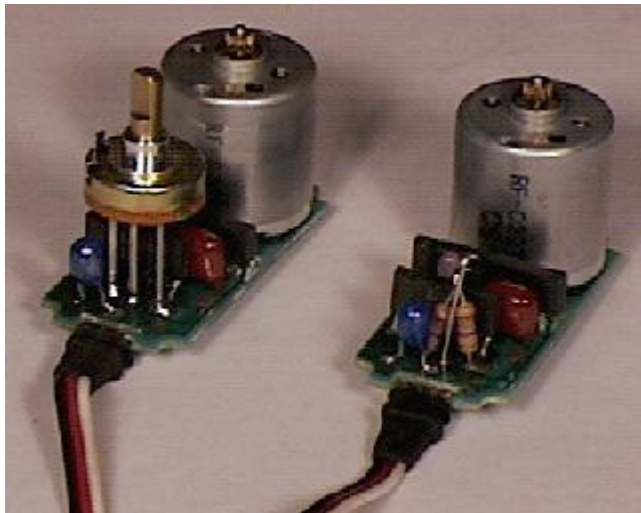
Po daljšem razmisleku sem ugotovil, da je najbolj optimalno uporabiti tri kolesa. Dve kolesi sta pogonski, eno pa je pomožno in se prosto vrti. Ta pogon je najbrž idealen za tekmovanje, ker se robot lahko obrača na mestu in ga je enostavno krmiliti. Za vožnjo po terenu bi mu bilo treba namestiti gosenice, a na srečo v labirintu ni bilo blatnih ovir. Robot se premika tako, da je pomožno kolo spredaj (večina ga je imela zadaj). Pred pogonska kolesa sem vgradil tudi ščitnike, saj bi se robot lahko zataknil ob rob zidu in bi obtičal.



Slika 2: Postavitev servo-motorjev in pomožnega kolesa

Pogonski del in kolesa

Za pogonski del sem uporabil navadne servo-motorje in sicer najbolj cenene od Graupner/JR. Modelarski servo-motorji imajo izjemne karakteristike: majhna poraba in izvedba, velika moč, nizka cena (okoli 3000 SIT za kos), vsebujejo pa tudi že velik del krmilne logike. Težava je le, da jih je treba predelati, tako da se bodo kontinuirano vrteli. Ko jih kupimo, se lahko namreč obračajo le za 180° . Predelamo jih tako, da namesto vgrajenega potenciometra namestimo dva upora po 2,2 k (slika 3).



Slika 3: Servu odstranimo potenciometer in vstavimo upora

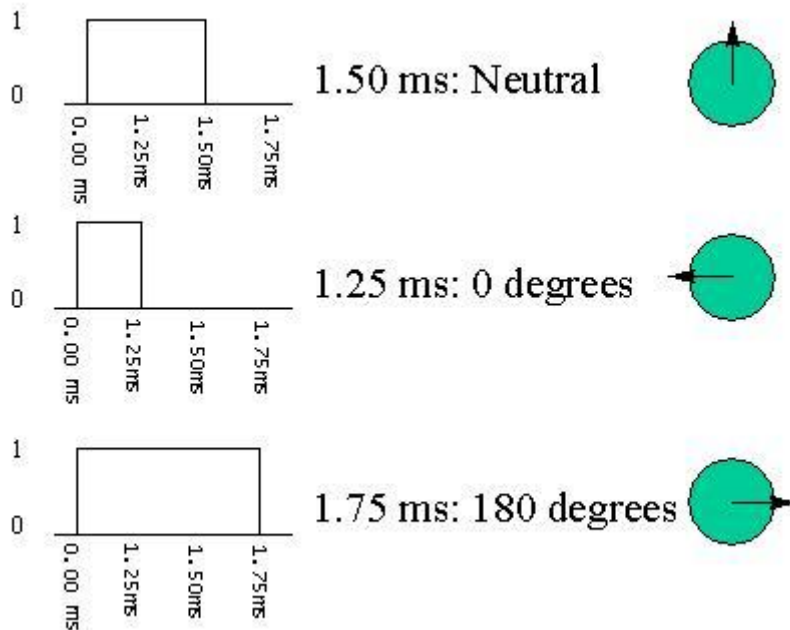
Potrebno je le še modificirati glavni in največji zobnik, na katerem je mehanska zaščita.



Slika 4: Zobniku odstranimo košček, ki onemogoča rotacijo

Težava s servo-motorji je tudi ta, da imajo sicer ogromen navor, a so zelo počasni. To se da rešiti na dva načina. Lahko uporabimo velika kolesa, kot je to storila večina tekmovalcev. Sam sem ubral drugo pot, tako da sem zamenjal zobnike v servu. Doma sem imel večje število zobnikov, ki sem jih skombiniral tako, da imajo sedaj motorji manjši navor, a izjemno visoko hitrost. Robot je bil tako zmožen potovati celo s hitrostjo 5 km/h (1.4 m/s). Zaradi te predelave sem uporabil manjša kolesa iz Lego sistema, na katera sem prilepil mehko gumo, da niso drsela. Za čim večjo hitrost sem uporabil napajalno napetost 8.4 V namesto predvidenih 5V. Tako so motorji sicer bolj trpeli in tudi poraba je bila višja, a se je vseeno izšlo.

Servo-motorje je treba ustrezno krmiliti. Na prvi pogled se bo morda komu zdelo zapleteno, a je v resnici preprosto. Servo ima 3 žice: dve sta napajalni (+5V), ena pa je krmilna. Servo pričakuje pulz na največ 20ms. Če je ta pulz (ki ga torej pošljemo vsakih 20ms) dolg 1.5ms, bo servo miroval. Če je krajši ali daljši, se bo servo vrtel v eno ali drugo smer (slika 5).



Slika 5: Odmik serva v odvisnosti od dolžine pulza

Kako pa izgleda del programa, ki krmili servote s pomočjo Parallax BASIC Stamp II krmilnika? Preprosto:

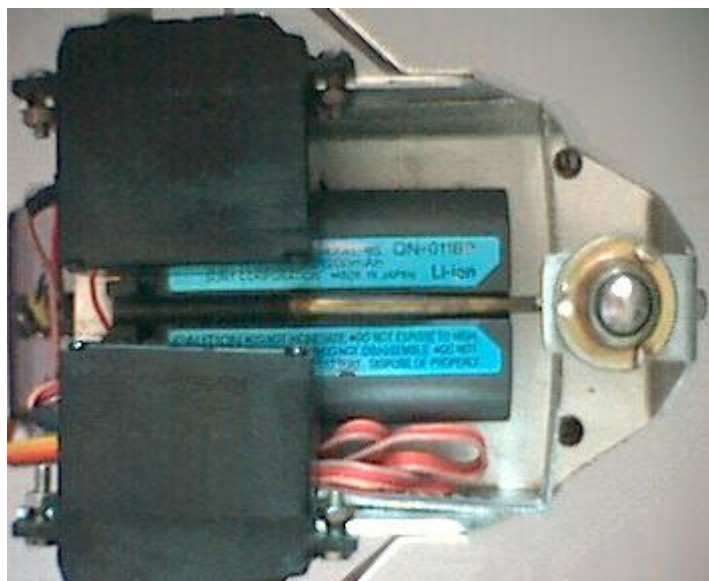
naprej:

```
pulsout levi_servo,650 ' ta ukaz pošlje servu pulz, katerega dolžino ponazarja številka 650
pulsout desni_servo,750 ' tukaj pa je številka 750, zato se bo ta servo vrtel v drugo smer
pause 20                ' servo pričakuje pulz vsakih 20 ms, zato je potrebna 20ms pavza
goto naprej             ' neskončna zanka
```

S tem delom programa, napisanega v Parallax Basic-u, bi se robotek premikal naprej.

Napajanje

Napajanje je ključnega pomena. Če se le-to med iskanjem poti samo za hip prekine, se bo kontroler resetiral in robotek bo nemočen obtičal. Najbolje je, če so akumulatorji tovarniško povezani ali pa zalotani. Povezava akumulatorjev v raznih škatlicah ni najboljša rešitev, saj se zaradi vibracij hitro prekine stik. Daleč najbolj optimalni so Li-Ion akumulatorji, ki so najlažji in se hitro polnijo. Sam sem uporabil komplet iz odsluženega GSM telefona, kajti njegova baterija ima kar 8.4 V in 1100 mAh, kljub temu pa je zelo lahka. Ker so pa Li-Ion akumulatorji še vedno zelo dragi, so dobra alternativa Ni-Mh, pa tudi Ni-Cd. Zelo pomembno pa je tudi, kam se akumulator vgradi. Najbolje je, da je težišče nad pogonskimi kolesi, kajti drugače se lahko vrtijo v prazno (slika 6).



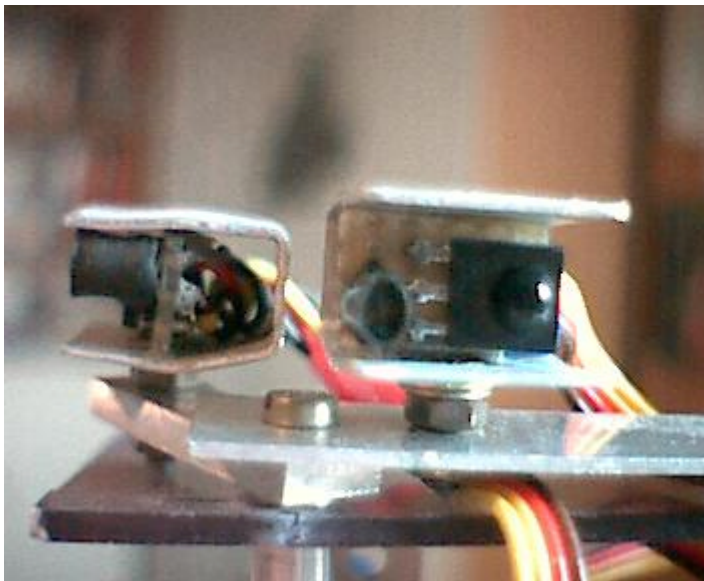
Slika 6: Postavitev baterije pod ohišje

Senzorji

Eden izmed najbolj zapletenih delov so gotovo senzorji (in procesor). Ko sem razmišljal, katere senzorje uporabiti, sem poskušal iskati podobnosti v naravi. Takrat sem tudi ugotovil, kako zelo daleč smo še od oči in tipal, ki jih ima vsak človek. Za robota sem nameraval uporabiti kamero, a se to za sam labirint ne izplača. V igri so bili senzorji na ultrazvok (ki lahko merijo tudi razdaljo) ali pa infra-rdeči (IR) senzorji. Odločil sem se za slednje, ki nam jih je tudi prijazno sponzoriral Inštitut za robotiko naše fakultete.

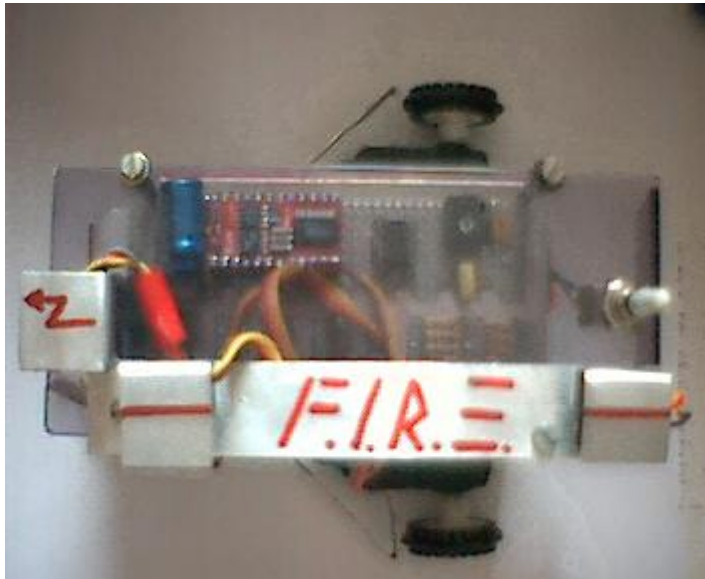
Ti IR senzorji so sestavljeni iz IR diode, ki s pomočjo oscilatorja oddaja svetlobo pri 35kHz, in sprejemnika, ki zaznava to svetlobo (ko se npr. odbije od stene). Domet oddajne diode je od približno 5 cm do kakšnih 30 cm, spreminjamo pa ga z nastavljivim uporom. Kadar sprejemnik zazna svetlobo, vrne stanje 0 V, ko pa ne zazna ničesar, vrne +5V, kar ustreza TTL logičnim nivojem. Ta TTL nivo je odličen tudi zato, ker lahko izhod iz sprejemnika speljemo direktno na logični vhod krmilnika.

Ker so senzorji občutljivi, jih je dobro vgraditi v nekakšno ohišje (slika 7). Sam sem za to uporabil košček aluminija ter vanj zalepil tiskano vezje z IR oddajnikom in sprejemnikom na njem. Na ta način sem sprejemnik tudi dodatno zaščitil pred vplivi okoliške svetlobe, kajti direktno sonce lahko senzorje popolnoma »oslepi«. Moduli senzorjev se dajo tudi enostavno prestaviti na poljubno mesto, saj so pritrjeni z vijakom. Bužirka na IR oddajniku je potrebna, da je snop bolj ozko usmerjen. Lahko bi se namreč zgodilo, da bi se senzorji med sabo motili.



Slika 7: Senzorja v kovinski zaščiti

Ker je bila naloga sledenje zidu, je potrebno senzorje še pametno razporediti na konstrukcijo robotka. Postavil sem jih tako, da je eden usmerjen naprej, dva pa na levo stran (slika 8). Tako lahko robot z lahkoto sledi levemu zidu.



Slika 8: Postavitev senzorjev - eden naprej, dva na stran

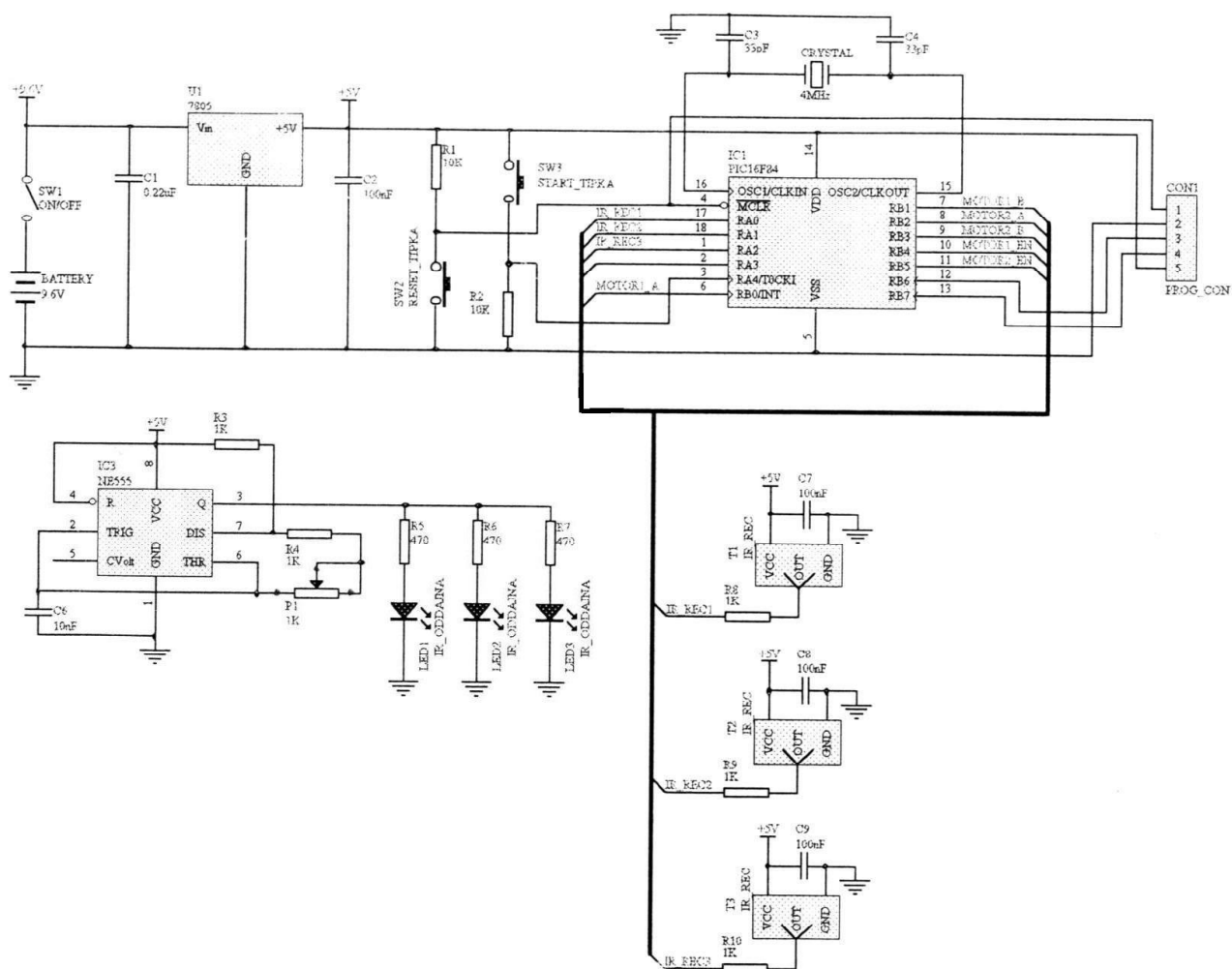
Vežje in mikro-krmilnik

Najvažnejši del vezja je mikro-krmilnik Parallax Basic Stamp II. Letos sem se odločil za tega, kajti prejšnje leto sem uporabil PIC 16F84 in nisem bil preveč zadovoljen. BS2 ima veliko funkcij namreč integriranih. V njem je interpreter, ki nam ukaze v PBasicu pretvarja v zbirnik, na njem je 5V stabilizator in kvarc, ki mu daje takt. BS2. Je eden najboljših čipov v tem razredu, tudi zaradi preprostosti. Čeprav nima interuptov in je nekoliko počasnejši od ostalih (vseeno zmore obdelati 4000 inštrukcij/sec), ga je zelo lahko programirati. To pomeni, da lahko napišemo razmeroma zapleten program v izjemno kratkem času. PBasic ima tudi veliko funkcij že vgrajenih, tako da jih ne rabimo pisati sami (npr. krmiljenje servotov, podpora zvoku, ...). Sam sem sicer kupil BS2e, ki ima 16k EEPROMa, saj sem imel dolg program. Navadni BS2 ima 2k EEPROMa. Problematična je lahko cena, ki znaša okoli 50 USD (okoli 13.000 SIT). Za primerjavo, PIC 16F84 stane okoli 700 SIT. Kot vedno, dobre stvari so drage.

Vežje je torej v bistvu sestavljeno iz mikro-krmilnika, oscilatorja NE555 (daje pulzirajočo svetlobo IR diodam), IR diod in sprejemnikov, 5-voltnega stabilizatorja napetosti (LM7805), nekaj kondenzatorjev in uporov. Na vezje sem zalotal še 9-polni konektor, ki ga s kablom povežem na COM vrata od računalnika. Servote in IR-sprejmenike se veže direktno na BS2, saj so tranzistorji že vgrajeni v servote. Načrt, ki je na sliki, je bil sicer v osnovi za PIC 16F84, a namesto slednjega vstavimo BS2 in odstranimo kvarc. Vežje za BS2 je tako še bolj preprosto.

Element	Količina	Opis
Basic Stamp II	1	mikrokontroler
NE555	1	oscilator
LM7805	1	5V stabilizator

1K potenciometer	1	spreminjanje dometa
QT QTEC 113	3	IR-dioda
PNA 4602M	3	IR-sprejemnik (Panasonic)
470E upor	3	pred IR-diodami
1K upor	5	izhod iz IR-sprejemnikov
100nF kondenzator	3	pri sprejemniku
10nF kondenzator	1	
220μF ELKO	1	za filtracijo napajanja
9-pin konektor	1	komunikacija z računalnikom



Slika 9: Shema vezja (v vezju je sicer PIC16F84)

Program

Napisati program je ena izmed lažjih nalog, čeprav na prvi pogled izgleda obratno. Preprost program v PBasicu lahko vsak napiše v par urah. Kot sem že omenil, ravno v preprostosti programiranja je največja moč BS2. Meni se je na samem tekmovanju pripetila nezgoda, ki bila lahko usodna. Nepričakovano mi je namreč odpovedal sprednji senzor. Težavo sem premostil tako, da sem ga zamenjal z zadnjim in modificiral program tako, da je robot zmagal samo z dvema senzorjema (sprednji in stranski-sprednji). Ta program, ki sem ga za lažje razumevanje močno oklestil in priredil, bom predstavil tudi tukaj. Če kdo želi celoten program, naj mi pošlje e-mail in z veseljem mu ga pošljem.

```
l var word          'deklaracija spremenljivk
r var word
right_IR var in3    'desni senzor je na vhodu 3
center_IR var in5   'sprednji senzor je na vhodu 5
left_servo con 7    'levi servo je na izhodu 7
right_servo con 6   'desni servo je na izhodu 6

l=639               'to je številka, ki drži servo v nevtralni legi
r=661

'opomba: senzorji so na levi strani

zaznavaj:          'glavni del, tukaj berem senzorja
if center_IR=0 then desno 'ko sprednji zazna steno, gre robot desno
if right_IR=0 then desno  'ko stranski zazna steno, gre desno, da se ne zaleti
if right_IR=1 then levo   'ko pa stranski izgubi stik s steno, pa se ji skusa priblizat

'vidimo, da robot nikoli ne vozi ravno, ampak cik-cak

levo:              'gre levo
pulsout left_servo,l-100
pulsout right_servo,r+100
pause 20
goto zaznavaj     'ko naredi en »korakec« v levo, skoci nazaj v zaznavanje, da
                  'preveri stanje

desno:             'gre desno
pulsout left_servo,l-100
pulsout right_servo,r+100
pause 20
goto zaznavaj     'ko naredi en »korakec« v desno, skoci nazaj v zaznavanje, da
                  'preveri stanje
```

Kot je razvidno, je program v bistvu res enostaven. Na samem tekmovanju sem uporabil še rutine, ki so poskrbele, da se robot ni izgubil, zataknil ali pa kje obtičal. Nameravam pa

sprogramirati, da si bo pot beležil v pomnilnik in bo tako še hitreje ter predvsem zanesljivo prišel do cilja. Beleženje poti pa zahteva že veliko več znanja in izkušenj.

Zaključek

Z izdelavo takšnega robotka sem pridobil neverjetno količino praktičnega znanja. V veliko veselje je, ko vidiš, da se tvoj umotvor premika in izogiba oviram. V robotih še leži velik potencial, a je naša tehnologija še premalo razvita. Najbolj je v zaostanku predvsem mehanika. Napovedi so, da bo okrog leta 2017 pravcati naval gospodinjskih robotov. Kdo ve, morda pa boste prav Vi nekoč slaven konstruktor, ki jim bo vdihnil inteligenco...



Slika 10: Ponosen konstruktor s svojim "konjičkom"

Literatura:

www.parallax.com

www.robotics.com

www.seattlerobotics.org/

www.nwlink.com/~kevinro/guide/

www.ro.feri.uni-mb.si/tekma/

Jani Bajec, študent 2. letnika UNI programa Računalništvo in informatika, Jani.Bajec@uni-mb.si

Mentor: mag. Janez Pogorelc, univ. dipl. inž., Janez.Pogorelc@uni-mb.si

Fakulteta za elektrotehniko, računalništvo in informatiko

Inštitut za avtomatiko in robotiko

Smetanova ul. 17, 2000 Maribor

Spletna stran: <http://www.ro.feri.uni-mb.si>