

Mobilni robot R5400

Avtor: Davorin Arcet
e-pošta: dav.a@email.si

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerze v Mariboru je lani že petič organizirala državno študentsko tekmovanje v konstrukciji in vožnji z mobilnimi roboti RoboT 2004. Na tem tekmovanju morajo majhni roboti lastne izgradnje čimprej prepeljati pot skozi postavljen labirint. Kar se sprva komu morda zdi dokaj preprosta naloga, lahko kmalu postane zahteven projekt, za katerega lahko porabimo ogromno časa.

Uvod

Sam sem se z mobilnimi roboti prvič srečal že v srednji šoli, in sicer v Zimski počitniški šoli Avtomatike in Robotike, ki jo vsako leto organizira UM-FERI v Mariboru v času zimskih šolskih počitnic. Majhni robotki, ki so sami vozili po labirintu, so me takoj začeli zanimati. In ko sem leto kasneje prebral članek v reviji Svet elektronike o mini mobilnem robotu, sem se takoj odločil, da bom na naslednjem tekmovanju sodeloval tudi sam. Seveda je uporabno, če imaš vsaj nekaj osnovnega predznanja o elektroniki in mikrokontrolerjih. Brez tega znanja pa se vseeno ni treba vdati v usodo, saj sem se tudi sam z malo raziskovanja in s pomočjo nasvetov mentorja ter izkušenih tekmovalcev z UM-FERI hitro naučil dovolj, da sem lahko sam sestavil konkurenčnega mobilnega robota.

Zgradba robota

Na začetku sem se moral odločiti, kakšnega robota sploh želim zgraditi. Uporabil bi lahko modelarske servomotorčke ali pa navadne (DC) elektromotorčke. Odločil sem se za slednje, saj sem jih že imel doma, pa tudi predelati jih ni bilo treba. Pri servomotorčkih bi moral najprej prekiniti povezavo med zobniki in potenciometrom, ki meri položaj ročice. Šele takrat bi jih lahko uporabil za pogon robota, saj se šele takrat lahko kolo zavrti za celih 360 stopinj. Mislim, da je najboljšje, če ima robot dva motorja in dve pogonski kolesi, ki se vrtita neodvisno eno od drugega, in če se le da, tudi v obe smeri. Takšen robot se lahko obrača na mestu in tudi zavira, ter se tako v ozkih hodnikih labirinta veliko lažje giblje. Zelo pomembna sta tudi krmilni in senzorski del robota. Na razpolago je veliko različnih krmilnikov in senzorjev, ki se zelo razlikujejo po zmogljivostih in ceni. Vse te dele

moramo pritrditi na primerno ohišje, ki je lahko kovinsko, leseno, iz pleksi stekla, ali pa, kot je letos naredila ena od ekip, ki je vse skupaj vgradila v odslužen in napol prežagani model avtomobila.

Moj robot

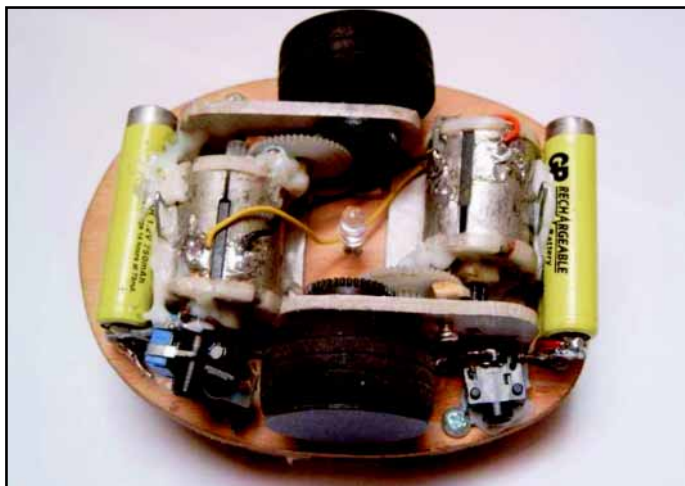
Kot sem že navedel, sem za robota uporabil dva elektromotorčka, ki pa sta se vrtela občutno prehitro, da bi jima lahko kolesa namestil direktno na gred. Da bi zmanjšal hitrost in pri tem še povečal navor, sem moral uporabiti zobnike (reduktor). Najprej sem poskusil kar z zobniki od kock LEGO, ki so bili zelo dobri, le da so zavzeli nekoliko preveč prostora. Kasneje mi je uspelo sestaviti boljši komplet zobnikov, ki zasedajo bistveno manj prostora in se tudi bolj tekoče vrtijo.

Za vodenje robota sem na prejšnjem tekmovanju uporabil mikrokontroler AT89C2051, letos pa zmogljivejši PIC16F876. Oba se da razmeroma enostavno programirati in tudi za oba si lahko sestavimo zelo enostaven programator. Največje težave so mi povzročali senzorji, ki mi na začetku niso in niso hoteli delovati. Uporabil sem infrardeče senzorje SFH 5110. Lahko bi uporabil tudi drage IR senzorje, s katerimi bi lahko meril razdaljo, vendar sem letos dokazal, da se da tudi s temi doseči zelo dober rezultat. Ohišje je eden najpomembnejših sestavnih delov robota, saj se na ohišje pritrdijo vsi ostali sestavni deli, z razporeditvijo le-teh pa določamo tudi težišče robota. Sam sem iz navadne vezane plošče izrezal ovalno obliko robota ter na eno stran pritrdil pogonski sistem in akumulatorje, na drugo stran pa vezje. Dober robot mora imeti čim nižje težišče. Na takšen način sem lahko vse najtežje dele robota (oba motorja in baterije) namestil čisto pri tleh. Pomembna je tudi oblika ohišja. Ker se robotom velikokrat zgodi, da se zaletijo v kakšno steno, je pametno, če je oblika ohišja takšna, da se robot ne zatakne ob oviro. Vsekakor je dobro, da je robot čim manjši.

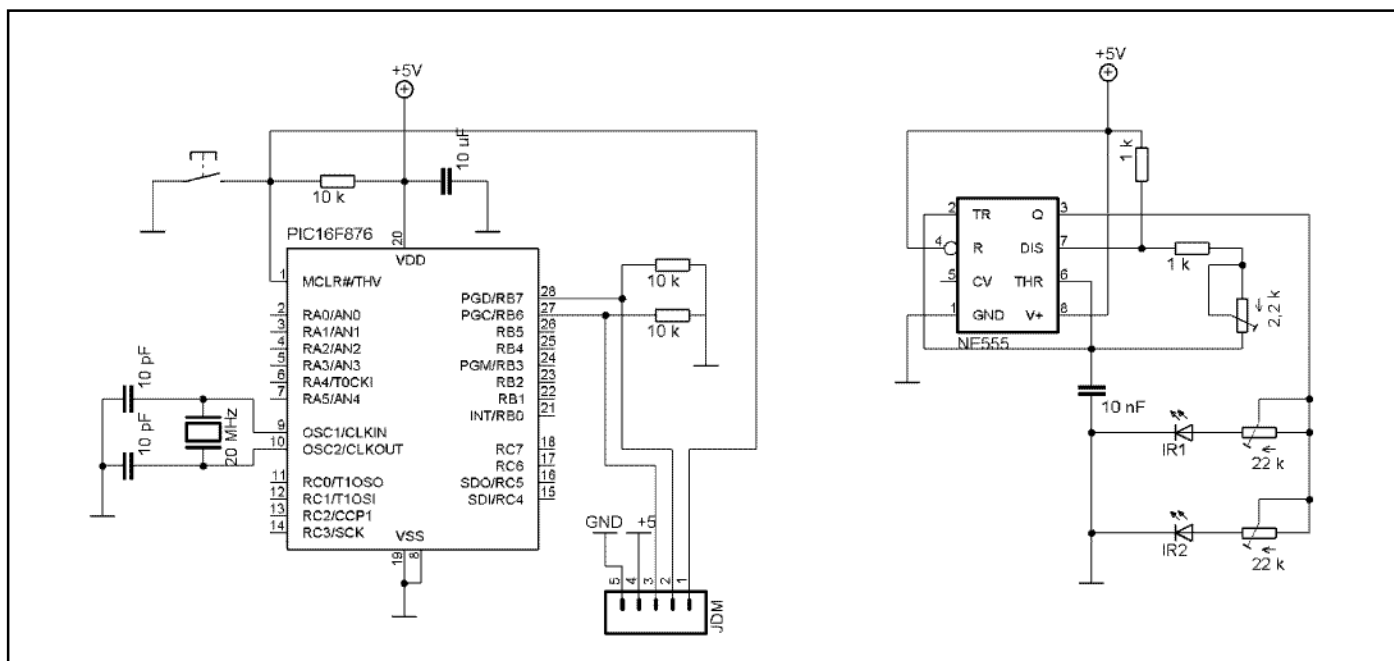
Za napajanje sem uporabil štiri 750 mAh NiMH akumulatorje (AAA). Za napajanje elektronskega vezja sem baterijsko napetost še stabiliziral. Ker sta oba elektromotorčka porabila veliko energije, sem med posameznimi vožnjami v labirintu vedno polnil baterije, saj je robot le tako dosegel svojo polno moč in hitrost.

Vežje

Mikrokontroler PIC16F876 za delovanje ne potrebuje kakega kompliciranega vezja, le nekaj najosnovnejših komponent. Malenkost drugače je pri IR senzorjih, kjer je prisotna tudi oddajna dioda. Dioda oddaja infrardečo svetlobo. Le-ta se odbije od stene labirinta in se vrne do sensorja, ki pri tem spremeni logično stanje na izhodu. Uporabil sem 36 kHz senzorje in zato sem oddajne diode moral napajati s signalom te frekvence. Kot oscilator



Slika 1: Pogonski sklop robota in akumulatorji za napajanje.



Slika 2: Osnovno vezje za mikrokontroler PIC 16F876 s konektorjem za »IN-CIRCUIT« programiranje z JDM programatorjem ter vezje z NE 555 za napajanje IR oddajnih diod.

sem uporabil čip NE555. S spreminjanjem frekvence sem lahko spreminjal domet senzorjev, saj je domet največji pri optimalni frekvenci, pri nekoliko višji ali nižji frekvenci pa se domet zmanjša. Ker so tako vsi senzori merili na isto razdaljo, sem pred vsako diodo še dodatno vezal potenciometer, s katerim sem vsaki diodi posebej določal moč.

Oba motorja sem napajal preko dveh lastno sestavljenih H-mostičev in sta se tako lahko vrtela v obe smeri. Uporabil sem tranzistorje BD 675A in BD 676A. Ker pa motorja zaradi nenehnega vklapljanja povzročata velike motnje v vezju, sem dodal še množico različnih kondenzatorjev. Prav tako sem stabiliziral napetost za senzorje in mikrokontroler. Ko sem še preizkušal senzorje, mi je velike težave in mnogo glavobolov povzročal sam NE 555, ki je v napajalno napetost vnašal velike motnje in pri tem onesposobil IR senzorje. To težavo sem odpravil s 1.000 mF elektrolitskim kondenzatorjem, vezanim na napajalne sponke.

Program

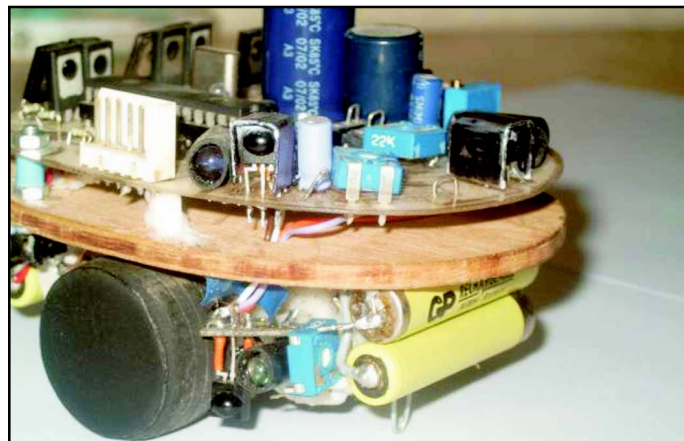
Če želiš robotu dopovedati, kako naj se glede na zaznavanja senzorjev obnaša, je najbolje uporabiti programirljiv mikrokontroler, v katerega s programatorjem vpišemo program. Na začetku sem si mislil, da takšnega programa ne bo ravno pretirano težko napisati. Kasneje sem ugotovil, da to velja le, če se ne pojavi želja po zmagi. Tako sem letos kontroler PIC16F876 programiral s pomočjo JDM programatorja, ki omogoča tudi IN-CIRCUIT programiranje in mi tako mikrokontrolerja ni bilo treba vedno jemati iz vezja. Program sem pisal v C-ju v programskem okolju PCWH, pri čemer sem večinoma uporabljal le bolj preproste stavke, ki se jih, z malo truda, lahko nauči vsak.

Celoten program za robota, ki sem ga uporabil na tekmovanju, zasede kar štiri strani in je tudi »poslovna skrivnost«, zato ga tukaj ne morem objaviti. Lahko pa približno opišem, kako naj bi se robot v labirintu obnašal.

Verjetno je najlažje, če se robot vozi ob eni steni. Če se recimo vozi ob desni steni, naj vseskozi dela rahel ovinek v desno. Ko senzor na desni strani zazna steno, pa se malenkost zavrti v levo.

S takšno vožnjo se lahko robot lepo pelje ob ravni steni. Če bi tako pripeljal do pravokotnega levega ovinka, bi se robot zaletel, saj ne bi delal dovolj velikega ovinka v levo. To težavo reši senzor, nameščen na sprednji strani robota, ki tako avtomatsko zaznava, kdaj je robot pripeljal do levega ovinka in pri tem naredi nekoliko večji ovinek v levo. V takšnem primeru je priročno, če se lahko robot obrača na mestu, saj se lahko zgodi, da se pri večjih hitrostih zaleti v steno. Drugo težavo predstavljajo desni ovinki. Takrat robot na desni ne zaznava stene in bi vseskozi delal rahel desni ovinek. Pri hitri vožnji bi se tako spet zaletel. Če na desno stran namestimo še en senzor, ki meri na večjo razdaljo kot prejšnji, lahko zaznavamo, kdaj je stena na desni strani in kdaj je ni. Naprej je enako kot pri sprednjem senzoru. Če na desni strani zmanjka stene, mora robot narediti večji ovinek desno. Senzorja na desni strani sem namestil enega nad drugim, saj bi se drugače preveč motila. Za prej opisano vožnjo robota skozi labirint so zadostovali trije senzori.

Z opisanim programom bi robot lahko lepo prepeljal labirint od starta do cilja, le da bi za to verjetno porabil kar precej časa. Ko se pojavi tekmovalni duh in množica ostalih tekmovalcev, se ti



Slika 3: Namestitev IR senzorjev.

vseskozi zdi, da je robot prepočasen. Seveda hitrejšega robota ni težko narediti. Le zamenjšaj zobnike, povečaj napetost baterij, ali pa spremeniš razmerje pri pulznem napajanju motorčkov. Tako hitrega robota je težje kontrolirati. Največ časa pri izdelavi robota sem porabil ravno za pisanje programa, s katerim bi hitrega robota lahko dovolj zanesljivo vodil po labirintu in tako dosegel dober rezultat. Med programiranjem sem vseskozi testiral v doma postavljenem labirintu in na srečo se je robot skoraj enako obnašal tudi v tekmovalnem labirintu. Na koncu mi je prvo mesto na tekmovanju ušlo le za pet stotink sekunde oziroma za malenkost napačno nastavitve senzorja.

Videz

V vsakega robota je vložena veliko truda. Ampak na koncu vsak gledalec najprej opazi le zunanji videz robota. In ta je, kot

opazam že vsa leta, zadnja skrb večine tekmovalcev. Včasih kakšen robot izgleda, kot da bi pravkar končal ravnanje s pobesnelim psom. Žice se da prav lepo skriti ali pa jih vsaj malo lepše speljati. Tudi kakšne zanimive dodatke se lahko namesti na robota. LED diode, LCD zaslončki, spojlerji in zastavice so povsod zanimive. Sam sem kot zanimivost na spodnjo stran robota namestil močno, rdečo LED diodo, tako da je robot spominjal bolj na avto iz spektakularnih hollywoodskih filmov, kot pa na zvestega predstavnika svoje vrste. In vsakdo, ki je prišel pogledat mojega robota, me je najprej povprašal po tej posebnosti.

Zaključek

Gradnja mobilnega robota je verjetno eden najboljših načinov za spoznavanje mikrokontrolerjev, elektronike in mehanike. Pri tem sem se naučil tudi, da je za izdelavo tako kompleksnega izdelka potrebno veli-

ko načrtovanja in še več testiranj, že dosti pred samim začetkom sestavljanja robota. Prav zaradi tega bi priporočal gradnjo takšnega robotka čisto vsem, ki jih vsaj malo zanimajo mikrokontrolerji in naprave, v katerih bi jih lahko uporabili. Potruditi se splača tudi zaradi samega programiranja robota. Nič ni bolj zabavnega, kot robot s slabo napisanim programom, ki noče in noče peljati tako, kot smo si zamislili. Mentor: mag. Janez Pogorelc, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko

Povezava na vire

Stran tekmovanja:

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

Forum, kjer sem dobil veliko informacij:

<http://www.elektro-n.com/>

Proizvajalec mikrokontrolerjev
PIC16F876:

<http://www.microchip.com/>

Progea Movicon (2)

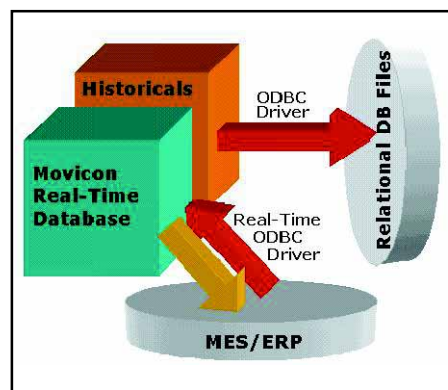
Avtor: Jernej Tavčar

E-pošta: jernej.tavcar@genera.si

Ena izmed prednosti, ki jo je prinesel razvoj računalniške opreme, je zmožnost shranjevanja velike količine podatkov, kar sodobni SCADA sistemi s pridom izkoriščajo in s tem omogočajo dostop do trenutnih in preteklih vrednosti procesnih spremenljivk. V prejšnji številki smo spoznali lastnosti programske opreme Progea Movicon, v tokratnem prispevku si bomo nekatere podrobneje ogledali. Ob izdelavi preproste aplikacije bomo spoznali uporabo podatkovnih baz in funkcij, povezanih z njihovo uporabo.

Za primer vzemimo enostaven proces, pri katerem merimo nivo tekočine v rezervoarju. Celotna aplikacija ima samo eno spremenljivko, to je nivo, ki jo opazujemo na nadzornem računalniku. Želimo tudi, da se vrednosti nivoja shranjujejo v bazo in izrisujejo na grafu.

Pri delu z bazami programska oprema Movicon omogoča enostavne in zmogljive rešitve. V celoti podpira ODBC standard (Open Data Base Connectivity) za shranjevanje podatkov v formatu katerekoli druge aplikacije, ki podpira ODBC standard. Poleg tega omogoča povezavo v realnem času z zunanjimi bazami podatkov. S tem odpravimo težave, povezane s prenosom podatkov z različnimi formati. Zapis baze spremenljivk v programu Movicon omogoča shranjevanje v formatu baze uporabnika, naprimer MS Access. Tako lahko uporabnik sam pregleduje in ureja podatke, ki so bili zapisani v programu Movicon. Za željen format zapisa baze morajo biti nameščeni ustrezni ODBC gonilniki. Ob



Slika 1: Povezava podatkov po ODBC standardu

namestitvi programa Movicon se namestitvo ODBC gonilniki za Access in Excel, na izbiro pa je tudi namestitvev gonilnikov za SQL Server MSDE. Za ostale formate baz moramo gonilnike posebej namestiti s programom ODBC Data Source, kjer pa ima že sam operacijski sistem pod privzetimi nastavitvami naloženih veliko ODBC gonilnikov (Paradox, FoxPro, dBase itd.),

tako da je le v redkih primerih potrebno namestiti dodatne gonilnike.

Orodje za upravljanje z bazami v Movicon programskem okolju se imenuje Data Logger. To je zmogljivo in preprosto orodje za urejanje in parametriziranje nastavitvev baz. Bazo lahko uporabimo za shranjevanje podatkov, ki jih potrebujemo pri kreiranju poročil, obrazcev, risanju diagramov ali pri raznih drugih analizah v projektu. Z Data Logger orodjem upravljamo preko Data Logger Editor funkcije. V vsaki bazi moramo določiti ime baze, ime tabele baze, imena stolpcev baze (te predstavljajo spremenljivke), maksimalno število shranjenih vrednosti na posamezni stolpec baze in določiti, kdaj naj se vrednost spremenljivke zapiše v bazo. Vsaka baza lahko vsebuje več tabel, v katere se zapisujejo spremenljivke. Podatki se zapisujejo v bazo po določenem redu. Ko je doseženo maksimalno število podatkov, se začnejo novi podatki beležiti čez najstarejše podatke. Do podatkov v bazi lahko dostopamo tudi