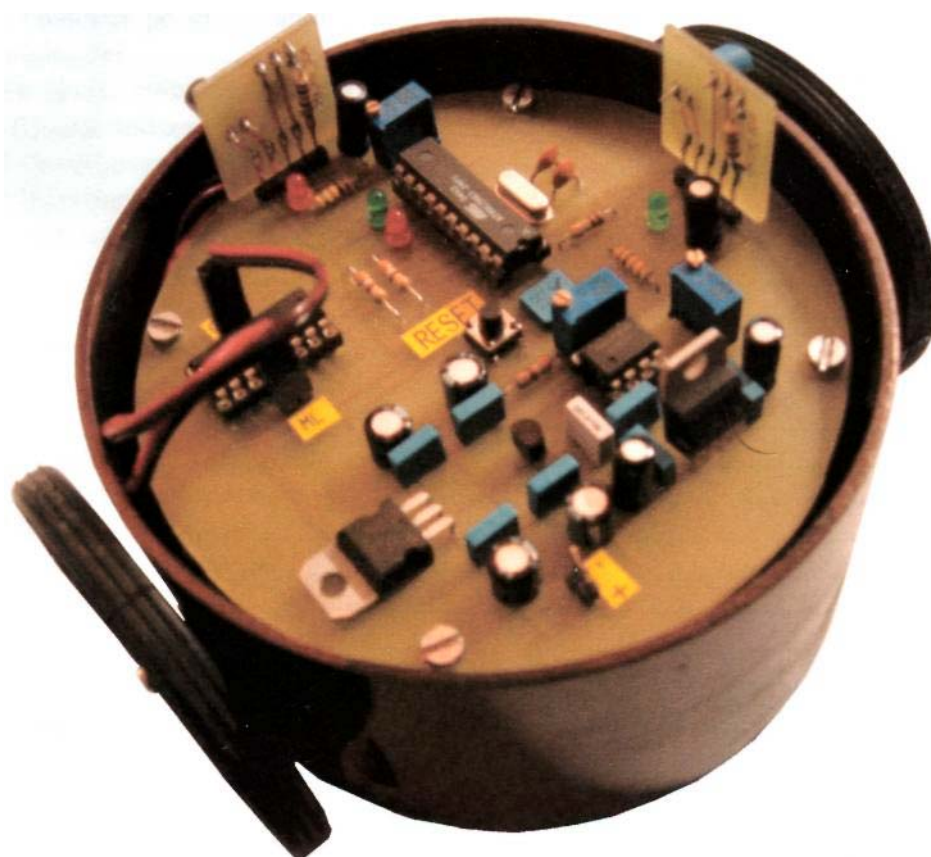


Šolski center Celje
PTEKŠ

Raziskovalna naloga

Mobilni robot RICHI 304



Mentor: Peter Kuzman, uni. dipl. ing.

Avtor: Matej Rajh, E-4.d

2005/2006

Kazalo

Povzetek	2
Zahvala	3
Uvod	4
Teoretični del	5
4. 1 Mikrokrmilnik ATMEL 89c2051	6
4. 2 Timer NE 555	6
4. 3 L293	7
4. 4 SFH5110	8
4. 5 Programiranje v zbirnem jeziku	9
4. 5. 1 Algoritem	9
4. 5. 2 Program	9
4. 5. 3 Postopek programiranja	9
Ekspperimentalni del	11
5. 1 Konstrukcija mobilnega robota	11
5. 1.1 Risanje tiskanega vezja	11
5. 1.2 Osvetljevanje ploščice	12
5. 1.3 Razvijanje vezja	12
5. 1.4 Jedkanje	12
5. 1.5 Vrtanje ploščice	13
5. 1.6 Zaščita ploščice	13
5. 1.7 Izdelava ohišja	13
5. 1.9 Predelava motorjev	14
5. 1.10 Izbira napajanja	15
5. 2 Opis vezja	16
5. 2.1 Oscilator in reset vezje	16
5. 2.2 Indikacija pozicije	16
5. 2.3 Krmiljenje motorjev	17
5. 2.4 Timer	17
5. 3 Umerjanje senzorjev	18
5. 4 Program za posamezne dele labirinta	19
5. 4.1 Vožnja naravnost in glavni program	19
5. 4.2 Zavoj levo	20
5. 4.3 Zavoj desno	21
Zaključek	22
Literatura	23
Priloge	24
8. 1 Program zapisan v zbirnem jeziku	24
8. 2 Seznam elementov	27
8.3 Shema	28
8. 4 Tiskano vezje	29

Povzetek

Namen raziskovalne naloge je bil izdelava mobilnega robota, ki bo v čim krajšem času prevozil labirint. Mobilni robot RICH304 sicer ni zelo hiter, je pa programsko zanesljiv, saj zaradi počasne vožnje manjša verjetnost napake pri zaznavanju senzorjev. Pri raziskovalni nalogi sem si pomagal z razno literaturo, ki mi je bila v pomoč.

Zahvala

Zahvaljujem se mentorju profesorju Petru Kuzman, uni. dipl. ing. pri odpravljanju problemov in dajanju koristnih nasvetov. Zahvaljujem se tudi učiteljem praktičnega pouka, predvsem gospodu Dušanu Bombač in gospodu Gvidu Par, ki sta mi omogočala razvijanje naloge tudi pri praktičnem pouku.

Uvod

Naloga mobilnega robota RICH304 je samostojno prevoziti labirint s pomočjo IR senzorjev. Robot se mora sam znajti v različnih situacijah, ki jih mora prepoznati sam, brez da mu zunanje pomagamo. Glavni program teče v zanki. Čez celoten program se preverja stanje na izhodih senzorjev. Program mora zajemati vožnjo naravnost, zavoje levo in desno za 90° in zavoje za 180° .

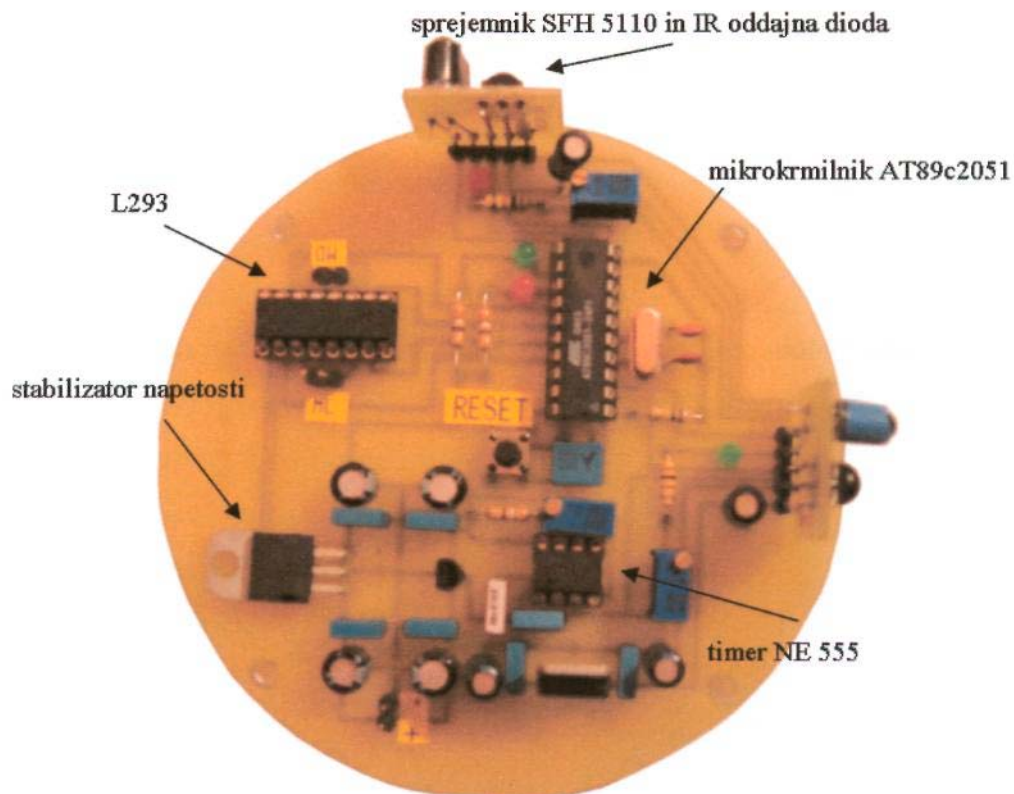
Hipoteza: Mobilni robot RICH304 mora s pomočjo IR senzorjev prevoziti labirint v čim krajšem času, biti pa mora tudi programsko zanesljiv.

Teoretični del

Naloga mobilnega robota je, da brez zunanje vodenja čim hitreje prevozi labirint. Pri izdelavi mobilnega robota je potrebno temeljito poznavanje njegovih sestavnih komponent.

Vezje mora vsebovati naslednje komponente:

- mikrokrmilnik, AT89c2051
- oscilator, NE555, ki deluje v stanju astabilnega multivibratorja
- gonilnik za motorje, L293
- senzorje, sprejemnik SFH5110 in IR oddajna dioda
- stabilizator napetosti
- ter ostale malo manj pomembne elemente

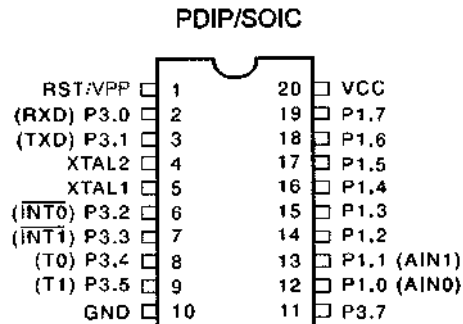


Slika 1: Najpomembnejše komponente

4.1 Mikrokrmilnik ATMELEL 89c2051

Mikrokrmilnik je mikroročunalnik, v katerem se nahaja centralno procesna enota CPE, ROM pomnilnik, RAM pomnilnik, paralelna vrata in serijski vmesnik. Ker je namenjen za vodenje naprav in procesov ima vgrajene še dodatne funkcijske enote kot so A/D pretvornik, D/A pretvornik, časovnik 0, časovnik 1, PWM, prekinitveni sistem, močnostne vhode in izhode.

Če potrebujemo še dodatne funkcijske enote, zgradimo zunanje vodilo, na katerega priključimo pomnilnike in vmesnike. Mikrokrmilniki družine Intel 8051 so 8 bitni. Imajo odlično programsko opremo, razvojno opremo in veliko število programsko kompatibilnih komponent.



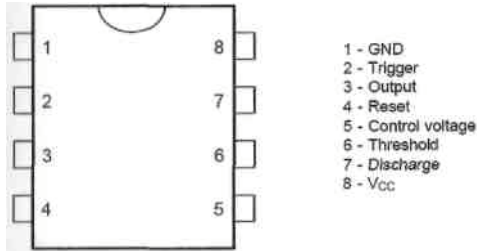
Slika 2: Priključki AT89c2051

Osnovni podatki mikrokrmilnika AT98c2051:

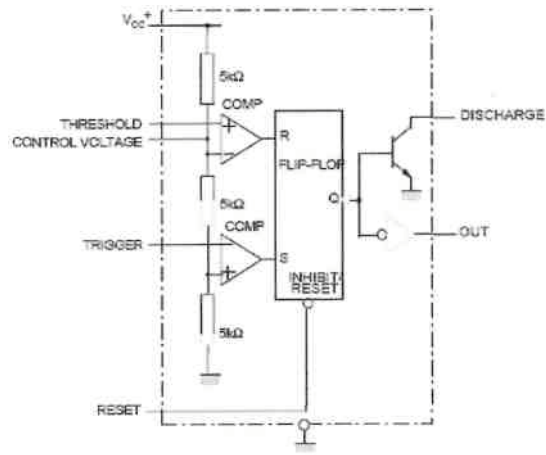
- sistemska frekvenca je med 1,2 in 16 MHz
- ukazni cikel traja 1/12 sistemske frekvence
- zunanji programski pomnilnik razširljiv do 64 K zlogov
- zunanji podatkovni pomnilnik razširljiv do 64 K zlogov
- 128 zlogov notranjega podatkovnega pomnilnika, 16 zlogov bitno naslovljivih
- 21 direktno naslovljivih posebnih registrov, 11 tudi bitno naslovljivih
- pet prekinitvenih izvorov z dvema prioriteta
- dva 16 bitna časovnika/števca
- serijska vrata z generatorjem takta za serijski prenos podatkov
- štiri 8 bitna paralelna vrata

4. 2 Timer NE 555

NE 555 je generator pravokotne napetosti, ki je potrebna za krmiljenje IR diode. Deluje kot astabilni multivibrator. Sposoben je generirati natančne časovne zakasnitve oscilacije. Njegova maksimalna izhodna frekvenca je 500 kHz. Ko deluje kot časovni zakasnilnik, njegov čas zakasnitve natančno kontrolirata zunanji upor in kondenzator. Da stabilno deluje kot oscilator, je potrebno njegovo frekvenco nadzorovati z dvema zunanjima uporoma in kondenzatorjem. Njegov izhodni tok je lahko največ 200 mA, kar pa zadostuje krmiljenju IR diode.



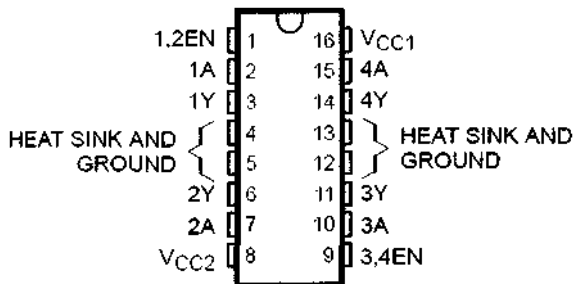
Slika 3: Priključki NE 555



Slika 4: Nadomestna shema časovnika NE 555

4.3 L293

L293 je visoko tokovni četveropolni krmilnik. Njegov izhodni tok je lahko največ 1 A, a vendar za zelo kratek čas. Razpon njegove napajalne oziroma delovne napetosti je med 4,5 V do 36 V. Namenjen je za krmiljenje induktivnih bremen kot so rele, enosmernih in bipolarnih koračnih motorjev, ter ostalih visokotokovnih bremen. Da je zmožen krmiliti tako velike tokove je njegov izhod krmiljen z darlington tranzistorjem.

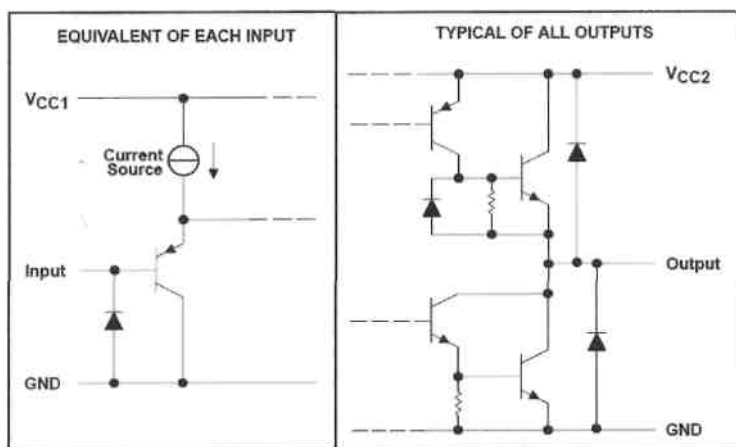


Slika 5: Priključki L293

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = doint care

Slika 6: Stanja na pinih



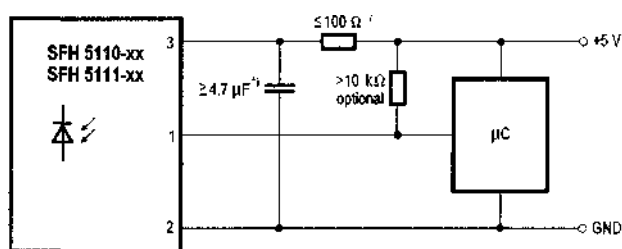
Slika 7: Nadomestna shema gonilnika za motorje

4.4 SFH 5110

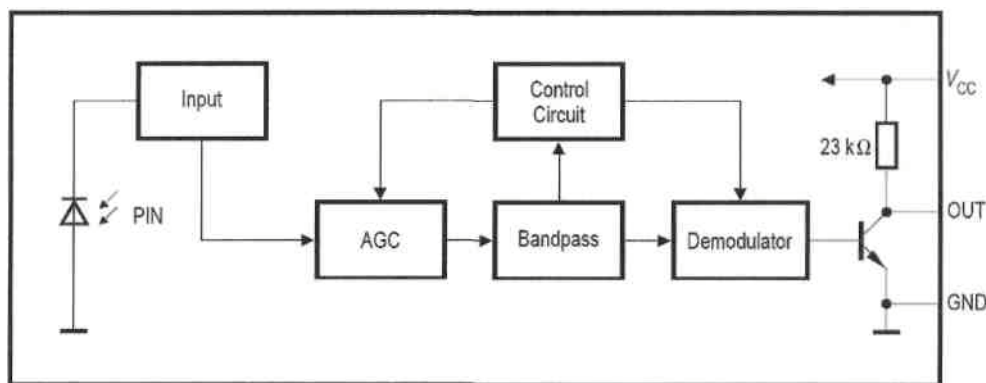
SFH 5110 je IR sprejemnik, ki zaznava IR svetlobo IR oddajnikov. Sam sprejemnik vsebuje fotodiodo, predojačevalnik, avtomatski nadzor ojačanja in povratno vezavo. Črne barve je zaradi tega, da nam ne zaznava dnevne svetlobe. Ima zelo veliko občutljivost, je TTL in CMOS kompatibilen. Njegov izhod krmili NPN tranzistor, kar pomeni da je aktiven pri logični ničli, saj nam izhod sklene na maso, ko zazna IR signal. Uporablja se v televizorjih, videorekorderjih, hi-fi sprejemnikih, med drugim se uporablja tudi kot optično stikalo.



Slika 8: SFH 5110



Slika 9: Vezja za stabilizacijo napetosti



Slika 10: Nadomestno vezje sprejemnika

4. 5 Programiranje v zbirnem jeziku 4.

5. 1 Algoritem

Algoritem pomeni splošen postopek za reševanje problemov. Predstavlja nam končen spisek navodil, ki jih lahko izvaja računalnik, da dobimo rešitev nekega problema.

Algoritme lahko predstavimo v naslednjih oblikah:

- v psevdo naravnem jeziku
- z diagrami poteka
- struktogrami
- s programom

4.5. 2 Program

Program, je opis algoritma v programskem jeziku. Ukazi so sestavljeni iz mnemonikov. Ko pišemo program, pišemo ukaze procesorju. Procesor bere zaporedje ukazov iz pomnilnika in jih izvaja enega za drugim.

Zapis programa v zbirniku

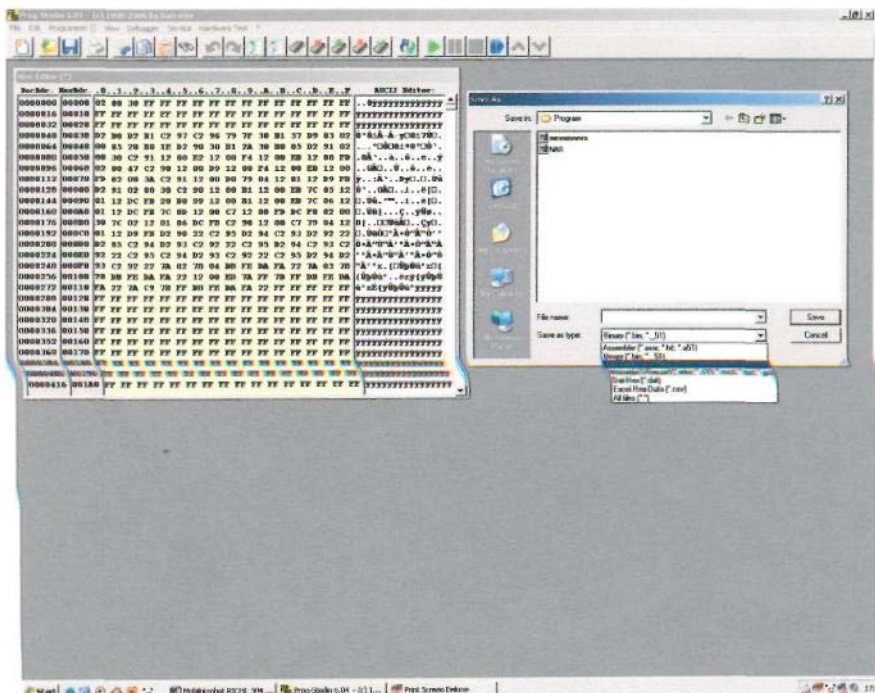
Program ponavadi pišemo v štirih stolpcih. V prvi stolpec pišemo labele, v drugi stolpec operacije, v tretji stolpec operande, v četrti pa komentar. Tako dobimo pregleden program, ki je razumljiv tudi drugim ne samo nam.

LABELA OPERACIJA OPERAND KOMENTAR

ZANKA: JNB P 1.1,ZANKA ;skoči na ZANKA, če je na vhodu P1.1 zaznana logična 0

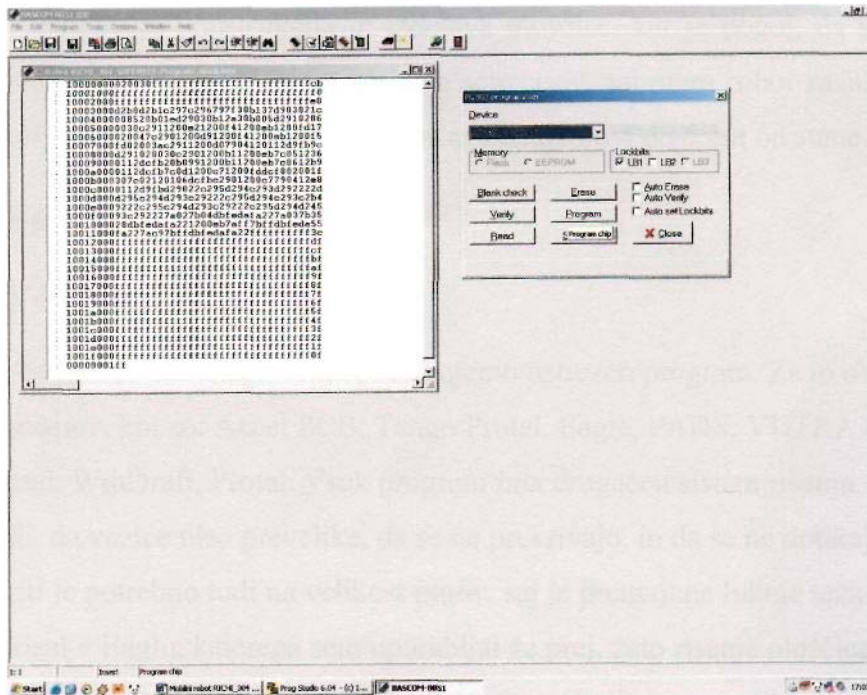
4. 5.3 Postopek programiranja

Pri programiranju v zbirniku se za pisanje programa v zbirniku uporablja razvojni sistem Prog-Studio. V prog studiu napišemo program po zgornji strukturi. Mnemonike je potrebno pretvoriti v strojno kodo, ki jo nato shranimo v formatu .hex, če za programiranje uporabljamo programator PG302, ali v formatu .bin, če uporabljamo BlowIT programator.



Slika 11: Programiranje v razvojnem sistemu Prog-Studio

Shranjeno datoteko odpremo v razvojnem sistemu Bascom, ki nam omogoča končno programiranje programa na mikrokrmilnik. Potem pa jo enostavno pošljemo na mikrokrmilnik.



Slika 12: Programiranje v razvojnem sistemu Bascom

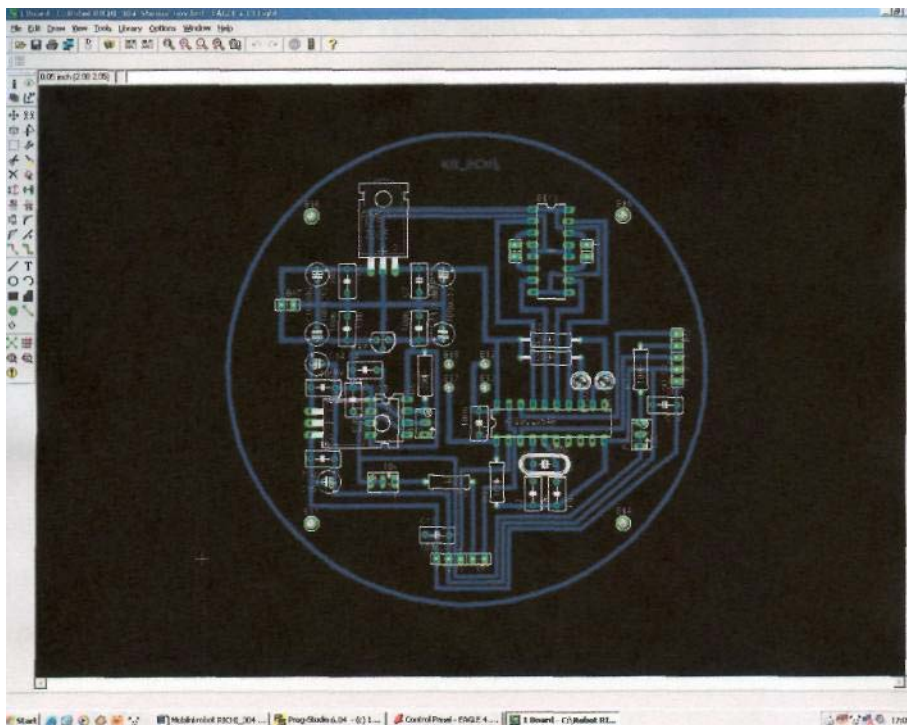
Eksperimentalni del

Nastajanje mobilnega robota je potrebno razviti od ideje do končnega projekta. Pri tem je potrebno vložiti veliko truda, predvsem v program, ki je potreben za njegovo delovanje. Pomembna je tudi sama konstrukcija in izdelava ploščice, kar je pogoj, da se ukvarjamo s pisanjem programa. Zelo težavno je umerjanje senzorjev, saj mora robot zaznavati nek zelen pas oddaljenosti od stene. Najboljše je, če zaznava nekje od 4 do 5 cm od stene.

5.1 Konstrukcija mobilnega robota

5.1.1 Risanje tiskanega vezja

Za računalniško risanje tiskanega vezja potrebujemo ustrezen program. Za to obstaja več različnih programov, kot so: Accel PCB, Tango Protel, Eagle, PADS, VUTRAX PCB, Easytrax, Circad, WinDraft, Protel. Vsak program ima drugačen sistem risanja. Pri risanju je potrebno paziti, da vežice niso prevelike, da se ne prekrivajo, in da se ne dotikajo, tam kjer se ne smejo. Paziti je potrebno tudi na velikost pinov, saj je premajhne luknje težko spajkati. Vezje sem narisal v Eaglu, katerega sem uporabljal že prej, zato risanje ploščice ni bilo tako težavno. Najprej sem si pripravil vse potrebne elemente, in jih razporedil na določena mesta. Nato sem elemente pravilno povezal, ter odpravil napake nastale pri risanju. Vsako vezje, ki ga narišemo je potrebno temeljito pregledati, saj nam lahko napake v vezju kasneje povzročajo nezaželene težave. Tako narisano vezje je potrebno natisniti. Tiskanje je mogoče na folijo ali na paus papir. Pri tiskanju na paus papir je potrebno vedeti, da se ploščica osvetljuje malo dlje, saj ni prozoren tako kot folija.



Slika 13: Risanje osnutka v Eaglu

5.1. 2 Osvetljevanje ploščice

Ploščica se osvetljuje z UV svetlobo. Osvetljevanje je možno na soncu, ali pa pod UV žarnico. Osvetljeval sem z razbito žarnico od javne razsvetljave. Na ploščico prikrito s fotolakom položimo pravilno obrnjeno že prej stiskano vezje, in ga prekrijemo s steklom, da lahko skozi pronica UV svetloba. Ploščico je potrebno osvetljevati približno dve minuti (odvisno od žarnice). Preveč osvetljena ploščica lahko poškoduje lak, in nam ploščica ne bo uspela.

5.1. 3 Razvijanje vezja

Vezje je po osvetlitvi treba razviti. Vezja se razvija v raztopini NaOH in vode. Razmerje mešanja je 7 gramov na liter vode. Osvetljeno vezje potopimo v mešanico, da ga prekrije. Pri razvijanju si lahko pomagamo z premikanjem posode ali pa z mehkim čopičem. Ko je vezje razvito, se vidi baker na ploščici. Po razvijanju je potrebno ploščico dobro izprati z vodo.

5.1. 4 Jedkanje

Jedkanje je najnevarnejše opravilo pri izdelavi vezij. Jedkalna tekočina običajno mešanica 35% HCl, 15% H₂O₂ in 50% vode. (HCl - solna kislina; H₂O₂ - vodikov peroksid). Postopek jedkanja pri tej raztopini je zelo hiter. Ploščico potopimo v raztopino. Ko se raztopina neha

peniti, je vezje izjedkano. Po jedkanju vezje speremo z vodo, da se postopek ustavi. Vezje lahko očistimo tako, da ga za nekaj časa damo nazaj pod UV žarnico, nato pa še v razvijalec, kar je dobro tudi za ploščico, saj se ploščica nevtralizira, ali pa ga očistimo z acetonom.

5.1. 5 Vrtanje ploščice

Ploščico vrtamo s svedri premera 0,8mm - 1,2mm. Pri vrtanju je potrebno paziti, da so svedri dovolj ostri saj lahko s skrhanimi svedri poškodujemo ploščico.

5.1. 6 Zaščita ploščice

Ploščico je potrebno po končanem vrtanju premazati z zaščitnim lakom ali raztopino smole in acetona, da jo zaščitimo pred oksidacijo in drugimi vplivi. Če ploščico premažemo s smolno raztopino se ploščica lažje spajka.

5. 1. 7 Izdelava ohišja

Izbira ohišja je zelo pomembna pri izdelavi mobilnega robota, saj mora biti majhno in spretno ker se robot ne sme zatikati ob stene in vogale. Zato sem izdelal ohišje iz navadne odtočne cevi in pleksi stekla. Cev sem odrezal z rezalnim strojem na višino 7cm. Nato sem označil luknje za os motorčkov, ki sem jih zvrtil z vrtnim strojem. Spodnja ploskev in kolesa so iz pleksi stekla debeline 3mm. Na spodnjo ploskev je pritrjena ploščica, ki je na distančnikih, motorčki in baterije. Najprej sem obrisal cev, da je bila velikost spodnje ploskve dovolj velika. Pleksi steklo sem obrezal z vbodno žago, nato pa ga zbrusil prvič z grobim, nato pa še z vodobrusnim brusilnim papirjem. Obe dve ploskvi sem zlepil skupaj, in jih pobarval z črno barvo v spreju.

5.1. 8 Dimenzioniranje koles

Kolesa morajo biti dovolj velika saj so servo motorčki zelo počasni. Kolesa so iz pleksi stekla, ki sem jih obrezal z vbodno žago, obrusil z brusilnim papirjem, da so čisto okrogla. Premer kolesje 6cm. Za boljši oprijem sem na kolesa pritrtil gumo, saj je površina koles zelo gladka.

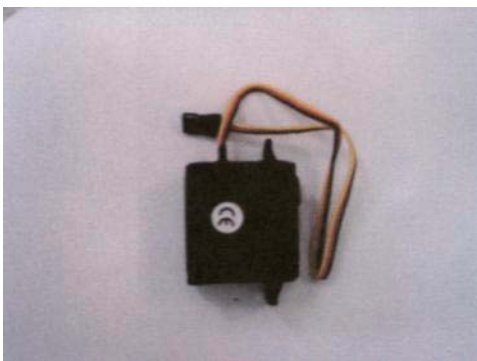


Slika 14: Kolo mobilnega robota

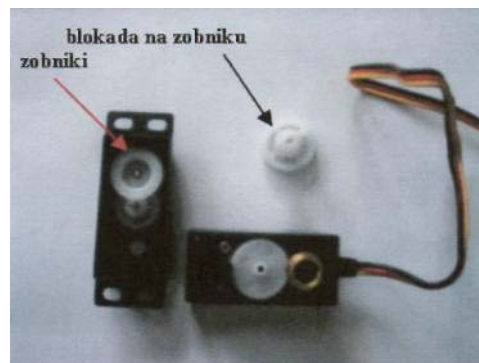
5.1.9 Predelava motorjev

Za gibanje mobilnega robota potrebujemo neko gibalno sredstvo, kar je v tem primeru navaden predelan servo motor. Za navaden DC motor se nisem odločil, ker se vrti zelo hitro, zaradi česar ga je zelo težko nadzirati. Za koračni motor pa se nisem odločil zato, ker ga ni tako enostavno krmiliti. Najlažje je bilo predelati servo motor. To pa smo naredili po naslednjem postopku:

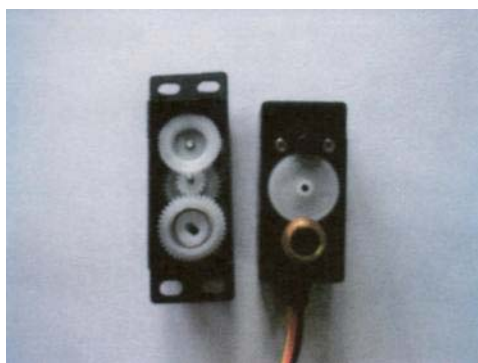
- motor razstavimo in ohišja
- razstavimo zobnike
- na zobnikih zbrusimo blokado
- odspajkamo mu krmilno žičko
- odstranimo krmilni potenciometer
- previdno nazaj sestavimo motor



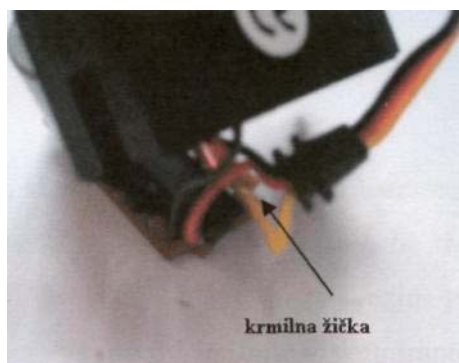
Slika 15: Motor mobilnega robota



Slika 16: Razstavljen motor



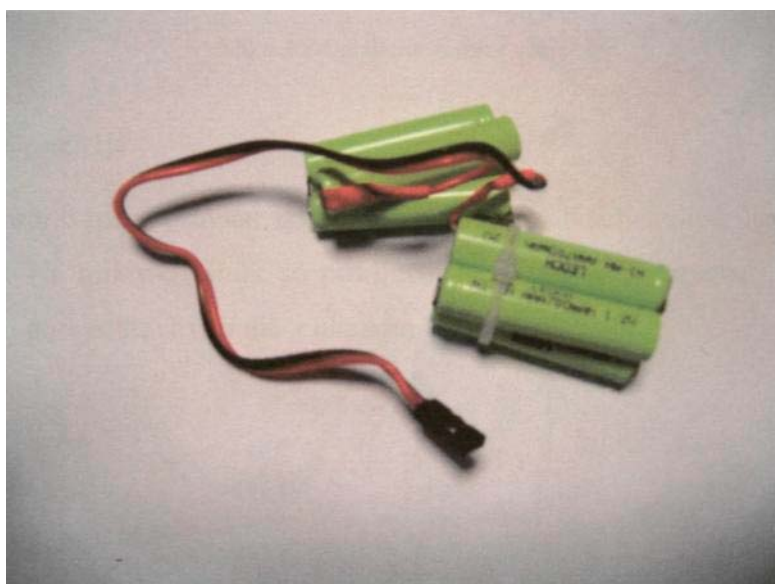
Slika 17: Razstavljen motor



Slika 18: Krmilno vezje motorja

5.1.10 Izbira napajanja

Zelo pomembna stvar pri izdelavi mobilnega robota je izbira napajanja, saj nam napajanje vezja določa hitrost vrtenja motorčkov. Napajanje mora biti čim bolj stabilizirano, imeti pa mora tudi dovolj veliko kapaciteto. Napajanje sestavlja sedem zaporedno vezanih 1,2 V baterij kapacitete 700 mAh, kar skupaj znaša 8,4 V. Zaradi majhne kapacitete se baterije hitro praznijo, kar pa ni dobro, saj je program napisan za neko določeno hitrost motorjev, ki pa jo določa napajanje.

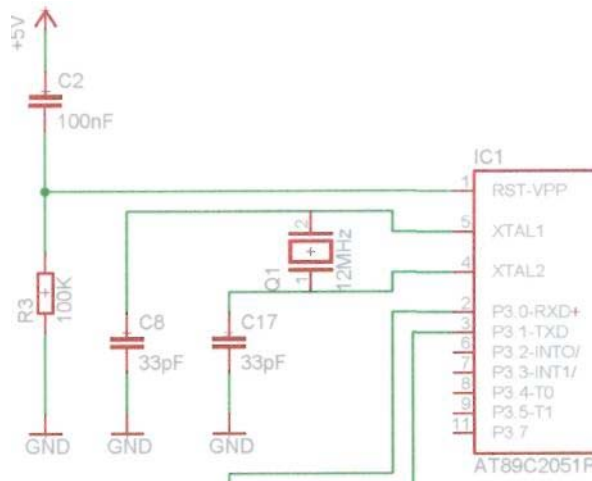


Slika 19: Baterije

5.2 Opis vezja

5.2.1 Oscilator in reset vezje

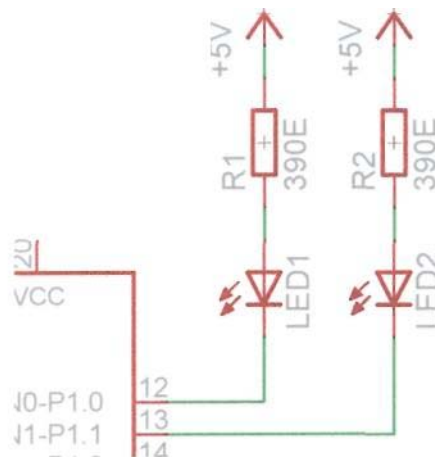
Za delovanje mikrokontrolerja je potreben oscilator, ki nam narekuje trajanje ukaznega cikla, ki je $1/12$ frekvence oscilatorja Q. Skrbi za sinhrono izvajanje ukazov in delovanje mikrokontrolerja. Zunanja priključka sta XTAL1 in XTAL2, ki ju v vezju povezuje enostopenjski inverter. Reset vezje skrbi, da se postavi mikrokontroler v definirano stanje. Impulz, ki postavi mikrokontroler v začetno stanje mora trajati vsaj dva strojna cikla.



Slika 20: Oscilator in reset vezje

5.2.2 Indikacija pozicije

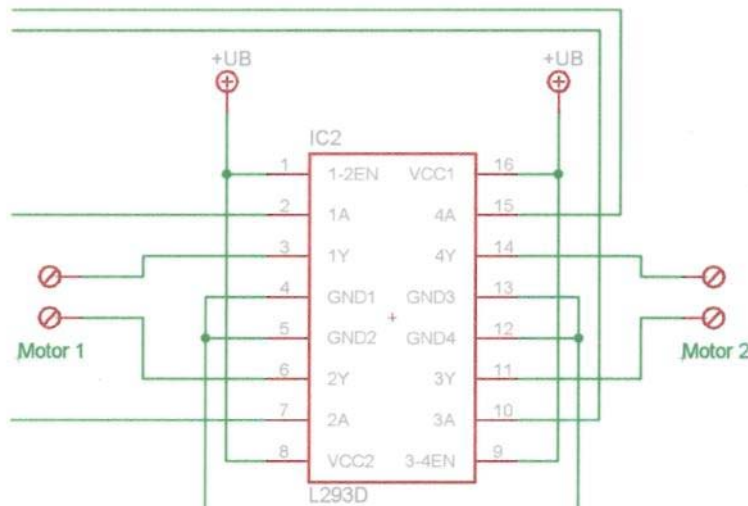
Da se ve katero rutino robot pravkar izvaja ponazarjajo indikacijske led diode priključene na port P1.0 in P1.1 Na mikrokontroler so priključene iz napajalne napetosti, zato moramo za njihovo aktivacijo port postaviti na nič z ukazom CLR P1.x.



Slika 21: Indikacijske led diode

5.2.3 Krmiljenje motorjev

Z regulatorjem L293 nadziramo hitrost in smer vrtenja motorčkov. Hitrost vrtenja nadziramo s pavzami v programu ter z napajalno napetost. Na pinih L293 mu dajemo kombinacije različnih logičnih stanj, s katerimi se vrtijo motorji naprej, nazaj ali pa stojijo pri miru.



Slika 22: Krmiljenje motorjev

5.2.4 Timer

S timerjem krmilimo oddajno IR diodo, ki oddaja signal s frekvenco 38 kHz. NE 555 deluje v načinu astabilnega multivibratorja. Z uporabo R5 in potenciometrom P1 polnimo kondenzatorja C1 in C3. Praznita pa se preko potenciometra P1, ter tako nastavljamo frekvenco oscilatorja.

Izračun frekvence oscilacije:

Čas vklopa:

$$t_1 = 0,693 (R_1 + R_2) C_1$$

Čas izklopa:

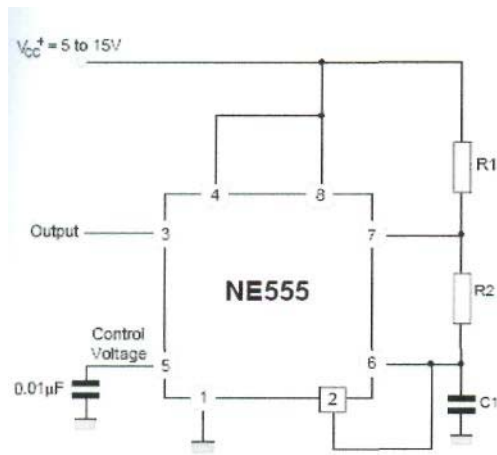
$$t_2 = 0,693 (R_2) C_1$$

Čas trajanja celotne periode:

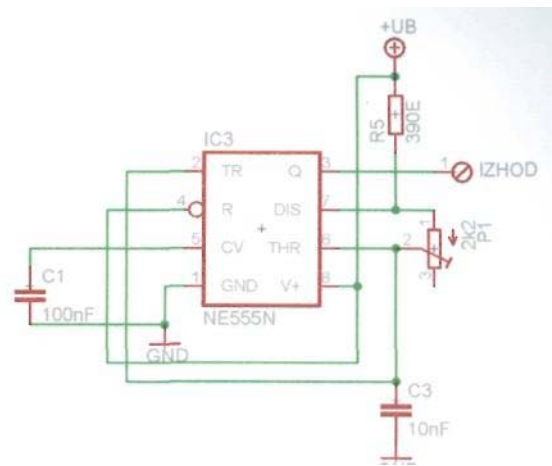
$$T = t_1 + t_2 = 0,693 (R_1 + 2R_2) C_1$$

Frekvenco oscilacije je potem:

$$f = 1/T = 1,44 / ((R_1 + 2R_2) C_1)$$



Slika 23: Shema vezja za izračun



Slika 24: Astabilni multivibrator

5.3 Umerjanje senzorjev

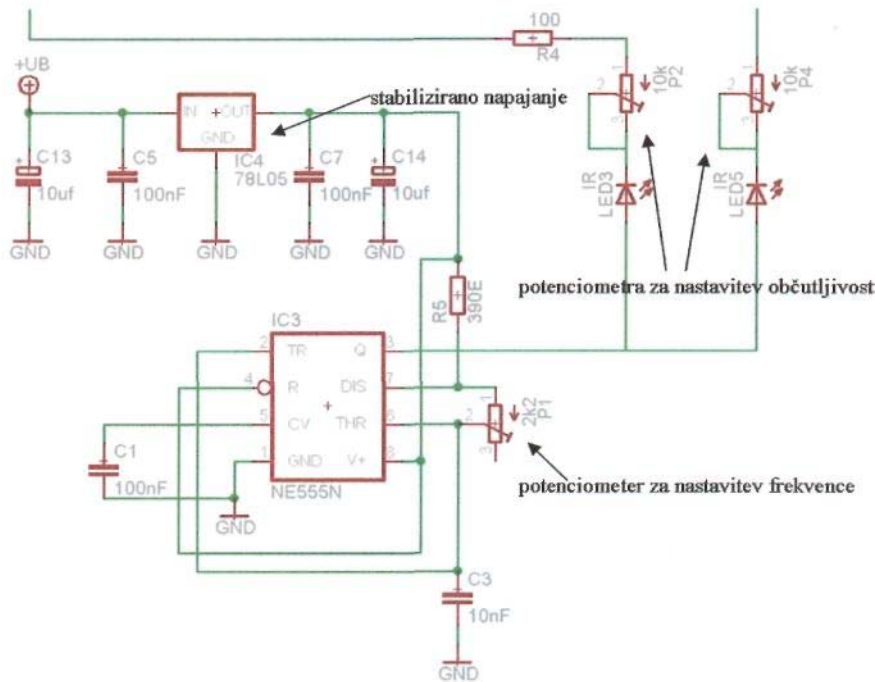
Najtežji del pri izdelavi mobilnega robota je umerjanje senzorjev. Odzivanje sprejemnika je zelo odvisno od osvetljenosti prostora v katerem je labirint. Senzorje sem umerjal pod nedirektno svetlobo žarnice. Oddajna IR dioda oddaja signal s frekvenco, ki jo nastavimo z astabilnim vezjem časovnika NE 555. Njegovo frekvenco določata kondenzatorja C3 in C1, ter fino nastavljivi potenciometer P1. Nastaviti je potrebno frekvenco enako frekvenci sprejemnika SFH 5110, kije 38 kHz. S fino nastavljivim potenciometroma P2 in P4 gladimo pravokotni signal, in tako nastavljamo odmik zaznavanja od stene. Četudi smo senzorje umerili, nam program ne deluje pravilno, saj je pod obremenitvijo obnašanje senzorjev spremenjeno, in je potrebno postopek umerjanja ponoviti, s preprostim programom. Umerjamo jih tako dolgo, dokler nam ne zaznavajo na takšni razdalji kot želimo.

Program za umerjanje senzorjev:

```
START:
JNB SD,ZELENA
JB SD,RDEČA
LJMP START
```

```
ZELENA:
CLR ILDD
SETB P1.2
CLR P1.3
SETB P1.4
CLR P1.5
SETB ILDD
LJMP START
```

```
RDEČA:
CLR ILDN
SETB P1.3
CLR P1.2
SETB P1.5
CLR P1.4
SETB ILDN
```



Slika 25: Umerjanje senzorjev

5. 4 Program za posamezne dele labirinta

Pri pisanju programa je zelo pomembno pravilno naslavljanje portov. Najlažje je, da si te porte predhodno poimenujemo s prireditvijo simbolov.

Primer prireditve simbolov:

```
ILDD EQU P1.0 ;indikacijska led dioda desno-zelena
ILDL EQU P1.1 ;indikacijska led dioda levo(naprej)-rdeča
```

Oddajni diodi oddajata signal čez celoten program. Stanje na sprejemnikih pa se preverja samo v glavnem programu. Program za mobilnega robota pričnemo pisati pri vožnji naravnost, nadaljujemo pa najprej na zavojih za 90° nato pa še na zavojih za 180°. Pri pisanju programa je priporočeno pisanje komentarjev, saj se drugače ne znajdemo, ko želimo program analizirati.

5. 4.1 Vožnja naravnost in glavni program

Glavni program, ki teče v zanki, je hkrati vožnja naravnost, le da med vožnjo naravnost program preverja stanje na pinih, da se robot pomika ob steni. Najprej preverja stanje izhoda sprednjega senzorja, nato pa desnega. Če robot steno zaznava teče podprogram odmikanja od stene. Ko pa stene ne zazna več takoj začne teči podprogram pomikanja k steni. K steni se

pomika tako dolgo dokler je ne *zazna*. Če se proti njej pomika predolgo (nekje več kot 200ms) pa pomeni da stene ni, zato sledi podprogram za zavoj.

Program vožnje naravnost:

```

CLR    IRDN                ;IR dioda za naprej oddaja signal
MOV    RI,#7Fh            ;v register RI je vpisana konstanta 7Fh

PO_P_D:
JNB    SN,ZAVOJ_LEVO     ;zavij levo, če je sprednji senzor zaznal steno
DJNZ   RI,NAPREJ        ;primerja konstanto vpisano v register in število ponovitev programa
LJMP   ZAVOJ_DESNO     ;če se program ponavlja predolgo časa pomeni, da ni več stene, zavije desno
NAPREJ:
JB     SD,POZ._DESNO    ;robot ne zaznava stene, zato se ji približuje
SETB   ILDD            ;ugasne led diodo, ki indicira pozicioniranje desno
PO_P_L:
JNB    SN,ZAVOJ_LEVO     ;zavij levo, če je sprednji senzor zaznal steno
JNB    SD,POZ._LEVO     ;robot zaznava steno, zato se ji oddaljuje
SETB   ILDL            ;ugasne led diodo, ki indicira pozicioniranje levo

```

5.4.2 Zavoj levo

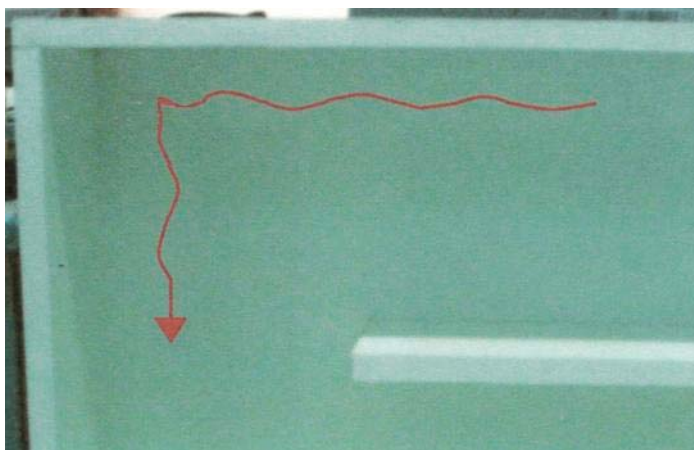
Robot zavije levo, če program zazna na pinu P 3.1 logično ničlo, kar pomeni da je preblizu stene in se mora obrniti ta 90° v levo. To stori tako da vrtil desni motor naprej, levega pa nazaj. To naredi obračanje zelo hitro. Dolžino obrata določa program, oziroma pavza obrata.

Program za zavoj levo:

```

ZAVOJLEVO:
CLR    ILDL                ;indicira zavijanje levo
LCALL  LEVOZ              ;;klic podprograma za zasuk za 90° v levo smer
MOV    RI,#04h           ;v register RI je vpisana konstanta 04h
Z_L:
LCALL  PAVZA_90          ;klic pavze
DJNZ   RI,Z_L            ;primerja konstanto vpisano v register in število ponovitev programa
SETB   ILDL             ;ugasne led diodo, ki indicira zavijanje levo
LJMP   START            ;skoči na začetek glavnega programa

```



Slika 26: Zavoj levo

5.4.3 Zavoj desno

Ko se robot vozi naravnost išče steno, poleg pa preverja tudi čas, ko ne zazna stene. Če je ta čas prevelik, pomeni, da ni stene in da mora zaviti desno. To nam meri sprejemnik SFH 5510, katerega izhod je priključen na pinu P3.0. Najprej se zapelje še malo naprej in zavije desno tako da se obrne v desno za 90°, ter se zapelje malo naprej. Sedaj preverja ali zazna steno, če jo, se nadaljuje vožnja naravnost. Če pa stene ne zazna, se še enkrat obrne za 90°, ter zapelje naprej ter nadaljuje vožnjo naravnost.

Program za zavoj desno:

```
ZAVOJ_DESNO:
  CLR    ILDD                ;indicira zavijanje DESNO
  LCALL  ZAVOJ_DESNO 90     ;klic podprograma za zavoj desno za 90°
  LCALL  NARAVNOST         ;klic podprograma za vrtenje koles naprej
  MOV    R1,#05h           ;v register R1 je vpisana konstanta 05h

MAL_N1:
  LCALL  PAVZA 90           ;klic pavze
  DJNZ   R1,MAL_N1         ;primerja konstanto vpisano v register in število ponovitev programa
  JNB    SD,START          ;skok na start če senzor SD zaznava steno
  LCALL  ZAVOJ_DESNO 90     ;klic podprograma za zavoj desno za 90°
  LCALL  NARAVNOST         ;klic podprograma za vrtenje koles naprej
  MOV    R1,#06h           ;v register R1 je vpisana konstanta 06h

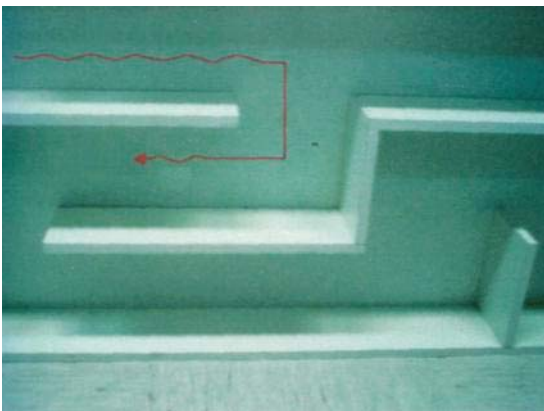
MAL_N2:
  LCALL  PAVZA 90           ;klic pavze
  DJNZ   R1,MAL_N2         ;primerja konstanto vpisano v register in število ponovitev programa
  MOV    R1,#0Dh           ;v register R1 je vpisana konstanta 0Dh

MAL_N3:
  LCALL  DESNOZ             ;klic podprograma za zasuk za 90° v desno smer
  LCALL  PAVZA_N_P         ;klic pavze
  DJNZ   R4,MAL_N3         ;primerja konstanto vpisano v register in število ponovitev programa
  SETB  ILDD               ;skoči na začetek glavnega programa
  LJMP   START

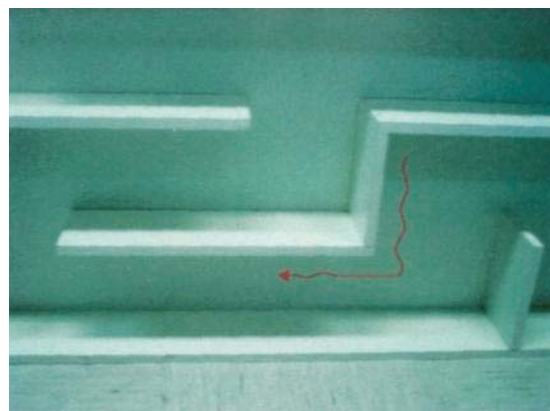
ZAVOJ DESNO 90:
  MOV    R1,#02h           ;v register R1 je vpisana konstanta 02h

MAL_N:
  LCALL  PAVZA_N           ;klic pavze
  DJNZ   R1,MAL_N         ;primerja konstanto vpisano v register in število ponovitev programa
  LCALL  DESNOZ             ;klic podprograma za zasuk za 90° v desno smer
  MOV    R1,#04h

Z D:
  LCALL  PAVZA 90           ;klic pavze
  DJNZ   R1,Z_D           ;primerja konstanto vpisano v register in število ponovitev programa
  RET
```



Slika 27: Zavoj desno za 180°



Slika 28: Zavoj desno za 90°

Zaključek

Robot prevozi labirint. Moti ga, če je svetloba prešibka, saj je takrat zaznavanje drugačno kot pri umerjeni svetlobi. Pri razvijanju projekta sem osvojil veliko znanja o programiranju ter se naučil razmišljanja pri reševanju problemov, ki se jih da rešiti z mikrokrmilniki.

Literatura

-zapiski iz predavanj digitalnih sistemov in krmilij, Celje 2003-2006

-Danilo Germ: Uvod v mikrokrmilnik 8051

-Peter Kuzman: Zbirka vaj Mikrokrmilnik 8051

-internet:

spletna stran www.datasheetcatalog.com

Priloge

8.1 Program zapisan v zbirnem jeziku

```

                                INCLUDE 89c52.mc

;MOBILNI ROBOT RICH304

;prireditve simbolov

ILDD EQU P1.0 ;indikacijska led dioda desno-zelena
ILDL EQUJ PL1 ;indikacijska led dioda levo(naprej)-rdeča
4A EQU P1.2 ;desni motor
3A EQU PL3
2A EQU P1.4 ;levi motor
1A EQU PL5
IRDN EQU P1.6 ;IR oddajna dioda naprej
IRDD EQU P1.7 ;IR oddajna dioda desno
SD EQU P3.0 ;senzor desno
SN EQU P3.1 ;senzor naprej

;uporabniški program
;organizacija zagona programa

ORG 0000h ;začetni naslov programskega pomnilnika
    LJMP START ;skok na glavni program

ORG 0030h ;začetek glavnega programa

START:
;inicializacija vrat

    SETB P3.0 ;port P3.0 je vhod
    SETB P3.1 ;port P3.1 je vhod

;glavni program

    CLR IRDD ;IR dioda za desno oddaja signal
    CLR IRDN ;IR dioda za naprej oddaja signal
    MOV RI,#7Fh ;v register RI je vpisana konstanta 7Fh

PO_P_D:
    JNB SN,ZAVOJ_LEVO ;zavij levo, če je sprednji senzor zaznal steno
    DJNZ RI,NAPREJ ;primerja konstanto vpisano v register in število ponovitev programa
    LJMP ZAVOJ_DESNO ;če se program ponavlja predolgo časa pomeni, da ni več stene, zavije desno

NAPREJ:
    JB SD,POZ_DESNO ;robot ne zaznava stene, zato se ji približuje
    SETB ILDD ;ugasne led diodo, ki indicira pozicioniranje desno

PO_P_L:
    JNB SN,ZAVOJ_LEVO ;zavij levo, če je sprednji senzor zaznal steno
    JNB SD,POZ_LEVO ;robot zaznava steno, zato se ji oddaljuje
    SETB ILDL ;ugasne led diodo, ki indicira pozicioniranje levo
    LJMP START ;skoči na začetek glavnega programa

;pomikanje ob steni

POZ_LEVO:
    CLR ILDL ;indicira pozicioniranje levo
    LCALL LEVO_P ;klic podprograma za vrtenje motorjev v levo
    LCALL PAVZA_P_Z ;klic pavze
    LCALL NARAVNOST ;klic podprograma za vožnjo naravnost
    LCALL PAVZA_N_P ;klic pavze
    LJMP PO_P_L ;skoči nazaj v glavni program

POZ_DESNO:
    CLR ILDD ;indicira pozicioniranje desno
    LCALL DESNO_P ;klic podprograma za vrtenje motorjev v desno
    LCALL PAVZA_P_Z ;klic pavze
    LCALL NARAVNOST ;klic podprograma za vožnjo naravnost

```

```

    LCALL PAVZA_N_P          ;klic pavze
    LJMP  PO P D            ;skoči nazaj v glavni program

;zavijanje
;zavoj levo
ZAVOJ_LEVO:
    CLR  ILDL              ;indicira zavijanje levo
    LCALL LEVO_Z          ;;klic podprograma za zasuk za 90° v levo smer
    MOV  RI,#04h          ;v register RI je vpisana konstanta 04h
Z_L:
    LCALL PAVZA_90        ;klic pavze
    DJNZ RI,Z_L           ;primerja konstanto vpisano v register in število ponovitev programa
    SETB ILDL             ;ugasne led diodo, ki indicira zavijanje levo
    LJMP START           ;skoči na začetek glavnega programa

;zavoj desno
ZAVOJ_DESNO:
    CLR  ILDD              ;indicira zavijanje DESNO
    LCALL ZAVOJ_DESNO_90 ;klic podprograma za zavoj desno za 90°
    LCALL NARAVNOST       ;klic podprograma za vrtenje koles naprej
    MOV  RI,#05h          ;v register RI je vpisana konstanta 05h
MALN1:
    LCALL PAVZA_90        ;klic pavze
    DJNZ RI,MAL_N1       ;primerja konstanto vpisano v register in število ponovitev programa
    JNB  SD,START         ;skok na start če senzor SD zaznava steno
    LCALL ZAVOJ_DESNO_90 ;klic podprograma za zavoj desno za 90°
    LCALL NARAVNOST       ;klic podprograma za vrtenje koles naprej
    MOV  RI,#06h          ;v register RI je vpisana konstanta 06h
MALN2:
    LCALL PAVZA_90        ;klic pavze
    DJNZ RI,MAL_N2       ;primerja konstanto vpisano v register in število ponovitev programa
    MOV  RI,#0Dh          ;v register RI je vpisana konstanta 0Dh
MALN3:
    LCALL DESNOZ          ;klic podprograma za zasuk za 90° v desno smer
    LCALL PAVZA_N_P       ;klic pavze
    DJNZ R4,MAL_N3       ;primerja konstanto vpisano v register in število ponovitev programa
    SETB ILDD             ;skoči na začetek glavnega programa
    LJMP START

ZAVOJ_DESNO_90:
    MOV  RI,#02h          ;v register RI je vpisana konstanta 02h
MALN:
    LCALL PAVZA_N         ;klic pavze
    DJNZ RI,MAL_N        ;primerja konstanto vpisano v register in število ponovitev programa
    LCALL DESNO_Z         ;klic podprograma za zasuk za 90° v desno smer
    MOV  RI,#04h
Z_D:
    LCALL PAVZA_90        ;klic pavze
    DJNZ RI,Z_D          ;primerja konstanto vpisano v register in število ponovitev programa
    RET                   ;vrnitev iz podprograma

;vrtenje motorjev pri zavijanju
;motorji desno
DESN0_Z:
    CLR  1A
    SETB 2A
    CLR  3A
    SETB 4A
    RET

;motorji levo
LEVO_Z:
    SETB 1A
    CLR  2A
    SETB 3A
    CLR  4A
    RET

;vrtenje motorjev pri pozicioniranju

```

;molorji desno

DESNO P:

```

CLR    1A
SETB   2A
CLR    3A
CLR    4A
RET

```

;motorji levo

LEVO P:

```

CLR    1A
CLR    2A
SETB   3A
CLR    4A
RET

```

;vrtenje motorjev pri vožnji naravnost

NARAVNOST:

```

CLR    1A
SETB   2A
SETB   3A
CLR    4A
RET

```

;PAVZE

;pavza za pozicioniranje

```

PAVZA_P_Z:
MOV    R2,#02h
PAVZA_P_Z2:
MOV    R3,#04h
PAVZA_P_Z1:
DJNZ   R3,PAVZA_P_Z1
DJNZ   R2,PAVZA_P_Z2
RET

```

;pavza za vožnjo naravnost pri pozicioniranju

```

PAVZA_N_P:
MOV    R2,#03h
PAVZA_N_P2:
MOV    R3,#28h
PAVZA_N_P1:
DJNZ   R3,PAVZA_N_P1
DJNZ   R2,PAVZA_N_P2
RET

```

;pavza za vožnjo naravnost

```

PAVZA_N:
LCALL  NARAVNOST
MOV    R2,#FFh
PAVZA_N2:
MOV    R3,#FFh
PAVZA_N1:
DJNZ   R3,PAVZA_N1
DJNZ   R2,PAVZA_N2
RET

```

;pavza za zavoj 90°

```

PAVZA_90:
MOV    R2,#C9h
PAVZA_902:
MOV    R3,#FFh
PAVZA_901:
DJNZ   R3,PAVZA_901
DJNZ   R2,PAVZA_902
RET

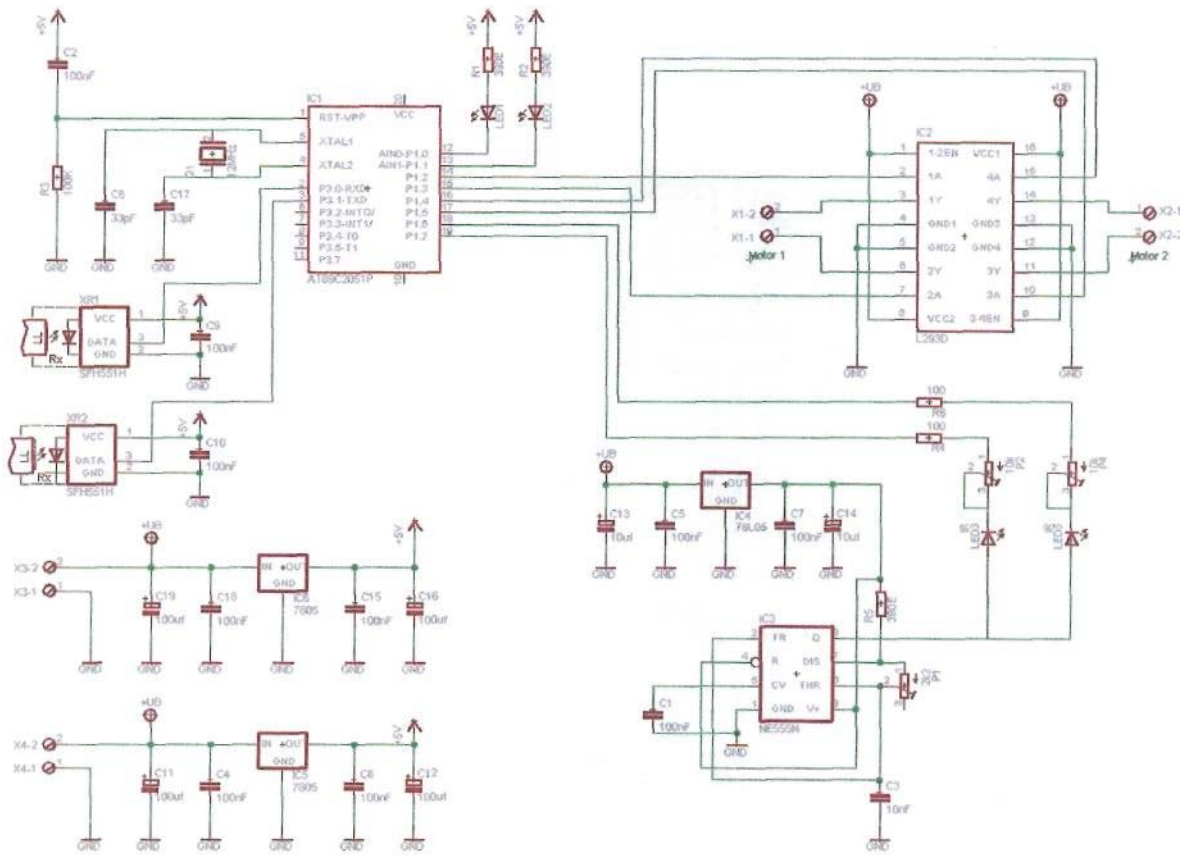
```

END

8. 2 Seznam elementov

Vrednost	Št. el.	Opis elementa
AT 89C2051	1	mikrokrmilnik
NE 555	1	timer
L293	1	driver
IR dioda	2	
SFH5110	2	IR sprejemnik
78L05	1	regulator napetosti
7805	2	regulator napetosti
10 uF	4	el. kondenzator
100 uF	2	el. kondenzator
100 nF	8	navaden kondenzator
10 nF	1	navaden kondenzator
33 pF	2	navaden kondenzator
100k	1	iupor
390	3	upor
100	4	upor
led dioda	2	
20 pin	1	podnožje
8 pin	1	podnožje
16 pin	1	podnožje

8. 3 Shema



Slika 29: Shema mobilnega robota R1CHI 304

8. 4 Tiskano vezje

Slika 30: Tiskano vezje

RICH304

