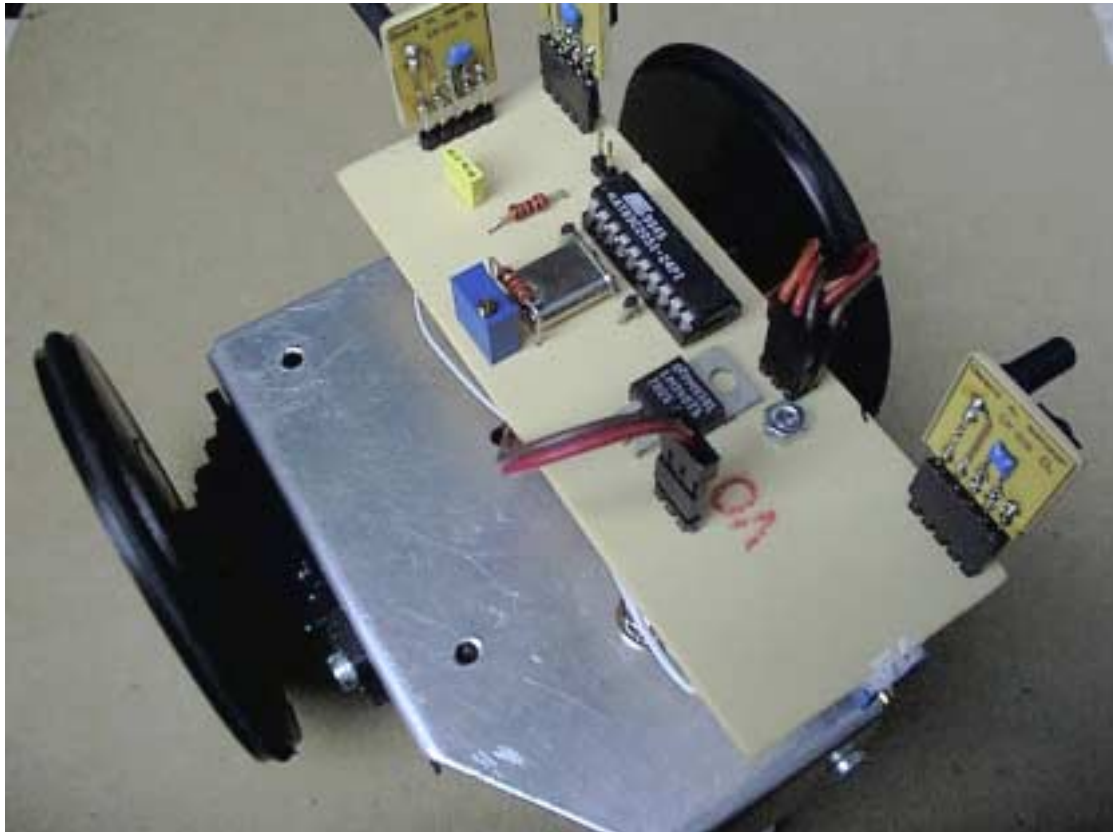


## MIMOR - MINI MOBILNI ROBOT LASTNE GRADNJE

### Uvod

Beseda robot zajema široko področje, pod katerim si predstavljamo, da neko delo opravlja stroj. V našem primeru je to delo premagovanje labirinta, za katerega je izdelan mini robot, ki s svojim gibanjem premaguje ovire. Glavni nadzor nad njegovim delovanjem opravlja Atmelov mikrokontroler 89C2051, enostavno programiranje pa nam omogoča program BascomLT, zato je ta članek primeren tudi za začetnike.



### Mehanski del robota

Robot ima stik s tlemi samo s tremi kolesi. Prva dva služita za pogon, tretje kolo pa je prostovrteče (*slika 1*).

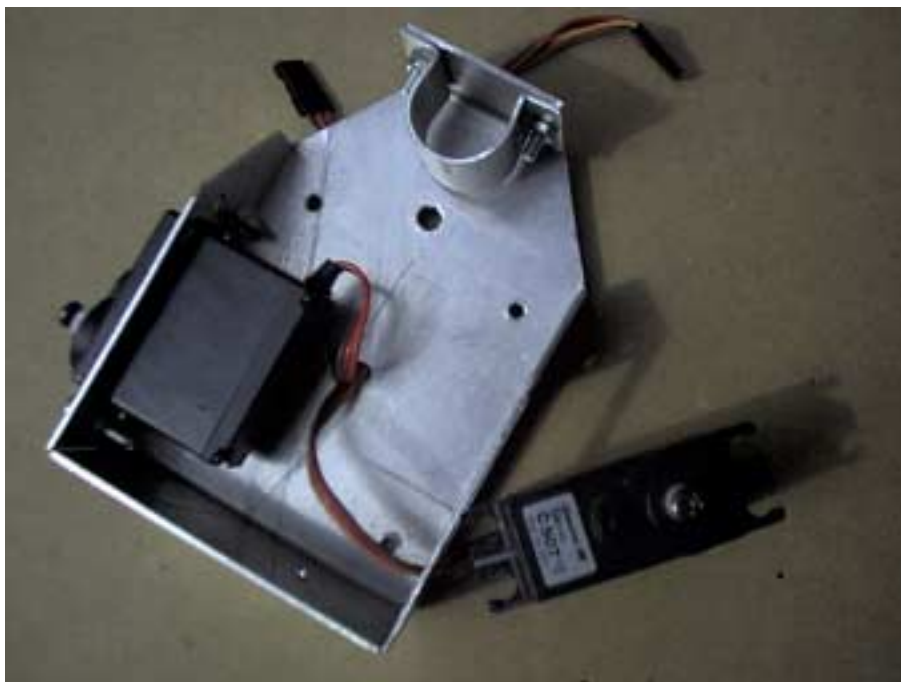


**Slika 1:** Na desni prostovrteče kolo; na levi pa kolo iz plastične mase obdano z gumo, na sredini pa je prilepljen element za pritrditev na servo-motor.

Prednost takšne izdelave je v tem, da se robot lahko obrača na mestu, to pa koristi kadar se znajde čisto blizu ovire. Za pogon sta uporabljena dva servo-motorja, ki sta pritrjena na aluminijast okvir. Oblika okvirja je na *sliki 2a*, ki se ga lahko poljubno spremeni, pomembno je to, da lahko nanj pritrdimo dva servo-motorja (*slika 2b*). Okvir služi samo toliko, da dele robota drži skupaj.



**Slika 2a:** Prikaz okvirja.



**Slika 2b:** V okvir vgrajen servo-motor.

Servo-motor vsebuje že svoje elektronsko vezje, katerega krmilimo preko treh priključkov, dva za napajanje (+5V), tretji pa za krmiljenje (PWM). Velikost zasuka servo-motorja (0-90°) je odvisno od širine pulza, ki se giblje od 1 do 2ms. Širina pulza se preverja vsake 20ms. Če na vhodu ni signala, potem se motor počasi ustavi, če pa želimo, da se hitro ustavi, mu moramo zagotoviti širino pulza dolgo približno 1.5ms (odvisno od motorja). Ker pa je robot lahek je pri obeh načinih ustavljanja dolžina zavorne poti enaka. Opisani servo-motor je iz radijsko vodenih R/C modelov (*Slika 3*).

V 2. odstavku smo ugotovili, da se s širino pulza (1 do 2ms) servo-motor različno odkloni, mi pa potrebujemo takšen način delovanja, da se bo servo-motor lahko ustavil, vrtel samo v eno ali samo v drugo smer, kolikor časa bomo hoteli. Zato je potrebna manjša predelava (odklop pozicijske povratne vezave).

Najprej odvijemo vijake in odstranimo zgornji pokrov. Zobnik, na katerega pride pritrjeno kolo, je povezan z potenciometrom. To vezavo moramo ločiti od potenciometra, tako da se ne vrtita skupaj. Nato zavrtimo potenciometer približno na sredino in vse skupaj sestavimo nazaj.

### Kolesa in hitrost

Za kolesa lahko uporabimo kar element, ki ga dobimo poleg servo-motorja (slika 3), če pa želimo imeti večjo hitrost robota, pa moramo imeti večja kolesa. Naredimo jih lahko iz plastične mase, na sredino pa prilepimo element, da lahko kolo pritrdimo na servo-motor (slika 1). Hitrost lahko povečamo tudi tako, da povišamo napetost na servo-motorjih (do 7.2V), vendar je potem potrebno v vezje vstaviti še 5-voltni stabilizator, da lahko napajamo ostale elemente.



Slika 3: Servo-motor iz R/C modelov in element, ki ga lahko uporabimo za kolo.

### Testiranje servo-motorja

Primer programa, ki ga bomo lahko izvajali kasneje, ko bomo imeli sestavljeno vezje.

*'Vrtenje motorja v eno smer z največjo hitrostjo*

```
Do          'DO-LOOP zanka
PI.3 = 1    'Na tretjem pinu porta 1 je signal velikosti +5V
Waitms 2    'Širina pulza je 2 ms
PI.3 = 0    'Na tretjem pinu porta 1 je signal velikosti 0V
Waitms 20   'Po 20ms je potrebna ponovna širina pulza
Loop
End         'Neskončna zanka
```

*'Vrtenje motorja v drugo smer z največjo hitrostjo*

```
Do          'DO-LOOP zanka
PI.3 = 1    'Na tretjem pinu porta 1 je signal velikosti +5V
Waitms 1    'Širina pulza je 1 ms
PI.3 = 0    'Na tretjem pinu porta 1 je signal velikosti 0V
Waitms 20   'Po 20ms je potrebna ponovna širina pulza
Loop
End         'Neskončna zanka
```

*'Počasna ustavitev motorja*

```
PI.3=0      'Na vhodu je vseskozi 0V
```

End

V primeru, da se motor ne vrti je lahko napaka v vezju oz., če motor samo 'trzne', je napaka lahko pri predelavi servo-motorja.

### Krmilni in logični del mobilnega robota

Ena izmed najbolj zanimivih nalog robota je 'premagati' labirint, zato potrebuje senzorje s katerimi zaznava okolico. Za dan primer zadostujejo IR (infra-rdeči) senzorji (*slika 4*), katerih uporaba je enostavna. Na oddajni IR diodi mora biti bužirka, zato da je snop bolj usmerjen. Deluje pri 35kHz (največji domet), s spreminjanjem frekvence pa lahko domet zmanjšamo. Sprejemna dioda ima tri priključke, dva napajanje (+5V), tretji pa je TTL izhod. Ko dioda zazna IR žarek se izhod postavi na 0V, v nasprotnem primeru je +5V. Ker imamo opravka s TTL napetostnimi nivoji, lahko senzorje priključimo neposredno na mikrokontroler.

Za napajanje lahko uporabimo štiri do šest Ni-Cd baterij po 1.2V velikosti AA, še boljše pa je uporabiti Ni-Mh baterije, ki imajo kapaciteto od 1.3 do 1,6Ah. Pritrdimo jih lahko pod vozilo, lahko pa tudi drugam.



Slika 4: IR sprejemna in oddajna dioda.

### Opis vezja

Shema sprejemnega dela z mikrokontrolerjem je na *sliki 6*, shema frekvenčnega generatorja pa je na *sliki 7*. Glavno vlogo v vezju ima Atmelov mikrokontroler, ki glede na zaznavanje IR sprejemnikov krmili oba servo-motorja. Za frekvenčni generator je uporabljen timer NE555, ki ga lahko natančno nastavimo z preciznim trimmerjem (domet). Če uporabimo tri IR elemente, potem je boljše narediti dva frekvenčna generatorja, zato da lahko enemu oz. dvema senzorjema nastavljamo različni domet. Servo-motorja sta priključena na P1.0 in P1.1. Ker mikrokontroler na teh dveh pinih zahteva zunanji pullup, sta proti Vcc priključena še dva upora. Na pinih od P1.2 do P1.7 ima mikrokontroler že notranje pullup-e, tako da IR sprejemniki pridejo priključeni direktno brez uporov. Za urin signal mikrokontrolerja služi kvarc in dva kondenzatorja.

**Če zvišamo napetost, da se bodo servo-motorji vrteli hitreje, potem moramo v vezje vstaviti še 5-voltni stabilizator (LM7805).**

### Seznam elementov:

Oznaka	Vrednost	Opis
IC1	Atmel 89C2051	mikrokontroler
IC2	NE555	timer
C1,C2	33pF	kondenzator
C3	1000 $\mu$ F/16V	ELKO
C4	10 $\mu$ F/63V	ELKO
C5,C6,C7	0.1 $\mu$ F	kondenzator

C8	10nF	kondenzator
R1,R2,R3	10K	upor
R4,R5	1K	upor
R6,R7	470E	upor
POT1	1K	potenciometer
IR1,IR2,IR3	PNA 4602M	IR sprejemnik (Panasonic)
IR4,IR5	QT QEC113	IR oddajna dioda
Q1	12MHz	kvarc

### Program, ki je bil sprogramiran v mikrokontrolerju na tekmovanju

*'Program za orientacijsko delovanje*

*Dim K As Integer*

*Dim D As Integer*

*Dim F As Integer*

*D = 0*

*F = 0*

*Do*

*If P1.4 = 0 Then*

*'Sprednji stranski senzor*

*If P1.2 = 0 Then*

*'Zadnji stranski senzor*

*Incr D*

*K = 5*

*Gosub Naprej*

*'Da se robot nebi zaletel v steno, se obrne čisto malo levo in*

*If D = 10 Then*

*'nadaljuje pot*

*K = 1*

*Gosub Levo*

*D = 0*

*End If*

*Else*

*'Če zadnji senzor ne zazna stene se obrne za malo večji kot levo*

*K = 2*

*Gosub Levo*

*End If*

*Else*

*' Ta else stavek je od prvega if stavka in je v delovanju, če*

*K = 3*

*'stranski sprednji senzor ne zazna stene*

*Gosub Naprej*

*K = 9*

*Gosub Desno*

*K = 15*

*Gosub Naprej*

*Incr F*

*If F = 8 Then*

*If P1.4 = 1 Then*

*Do*

*K = 5*

*Gosub Naprej*

*K = 11*

*Gosub Desno*

*K = 10*

*Gosub Naprej*

*F = 0*

*Loop Until P1.4 = 0*

*Else*

*F = 0*

*End If*

*End If*

*End If*

*If P1.3 = 0 Then*

*'Zaznavanje sprednje stene*

*K = 10*

*Gosub Desno*

```

K = 10
Gosub Nazaj
K = 10
Gosub Levo
K = 22
Gosub Levo
End If
K = 5
Gosub Naprej
If P1.3 = 0 Then           'Zaznavanje sprednje stene
  K = 10
  Gosub Desno
  K = 10
  Gosub Nazaj
  K = 10
  Gosub Levo
  K = 22
  Gosub Levo
End If
If P1.3 = 0 Then         'Zaznavanje sprednje stene
  K = 10
  Gosub Desno
  K = 10
  Gosub Nazaj
  K = 10
  Gosub Levo
  K = 22
  Gosub Levo
End If
Loop
End

                                     'Podprogrami za premikanje robota

Naprej:
Do
P1.0 = 1
P1.1 = 1
Waitms 1
P1.1 = 0
Waitms 1
P1.0 = 0
Waitms 18
Decr K
Loop Until K = 0
Return

Nazaj:
Do
P1.0 = 1
P1.1 = 1
Waitms 1
P1.0 = 0
Waitms 1
P1.1 = 0
Waitms 18
Decr K
Loop Until K = 0
Return

Desno:
Do

```

```

Decr K
P1.0 = 1
P1.1 = 1
Waitms 1
P1.1 = 0
P1.0 = 0
Waitms 20
Loop Until K = 0
Return

```

```

Levo:
Do
Decr K
P1.0 = 1
P1.1 = 1
Waitms 2
P1.1 = 0
P1.0 = 0
Waitms 20
Loop Until K = 0
Return

```

```

Stoj:
P1.0 = 0
P1.1 = 0
Return

```

Na prvi pogled izgleda program dolg in zapleten, vendar so v glavni zanki predvsem »if – then – else« stavki, ki se tudi ponavljajo. Običajno začnemo s kratko verzijo programa, potem pa program popravljamo in dopisujemo glede na reagiranje robota. Tako nastane iz kratkega programa malo daljši in bolj uporaben. Program, ki sem ga napisal je ena izmed možnosti, lahko pa ga napišemo na veliko drugih načinov. Tema je zelo odprta, zato lahko vsak pokaže svoje ideje in rešitve.

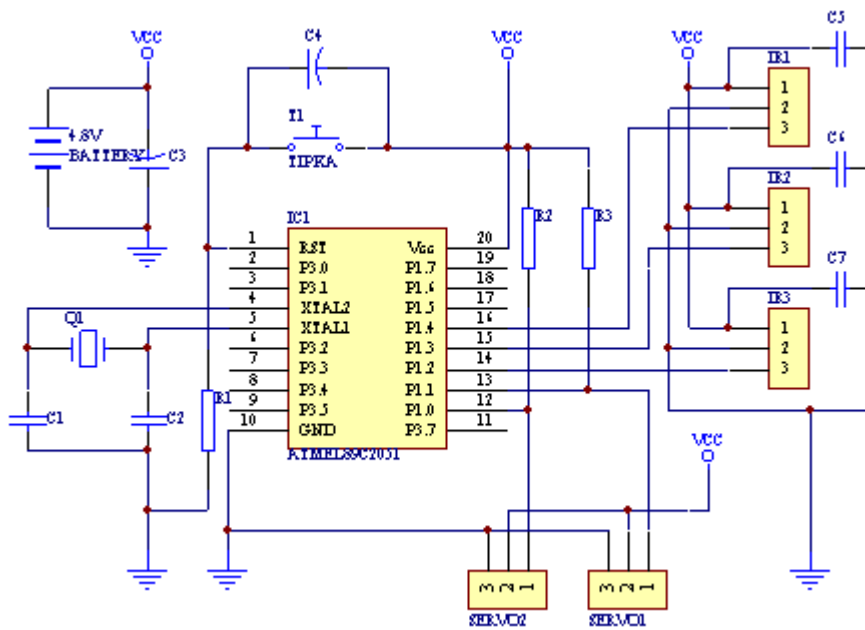
Krajša razlaga programa:

Robot ima tri senzorje, ki so priključeni na P1.2 (stranski zadnji), P1.3 (sprednji) in P1.4 (stranski sprednji). Da robot zanesljivo najde izhod iz labirinta, je najboljše, da vseskozi sledi steni (samo levi ali samo desni), zato sta oba senzorja na desni strani. Začetni del programa preverja samo stranska senzorja, in to tako dolgo (pomika se malo naprej in malo desno), dokler s sprednjim stranskim senzorjem ne zazna stene. Ko jo zazna, se robot obrne malo levo in malo naprej, dokler s stranskim zadnjim senzorjem ne zazna stene. Tako se robot na začetku postavi vzporedno s steno. Ko obe stranski sprejemni diode zaznavata steno (odbiti snop IR žarka), se robot pelje naravnost, tako dolgo, dokler sprednji senzor ne zazna ovire. Ker se vozilo drži desne stene pomeni da se mora pri zaznavi sprednje stene obrniti za 90 stopinj levo (če so stene labirinta postavljene med seboj za 90°). Del programa, ki se začne na *If P1.3 = 0 Then...Endif*, se večkrat ponovi, ker zaznava sprednjo oviro in jo mora pravočasno odkriti. V primeru, da robot izgubi desno steno, se ponovno nastavi tako, kot je opisano. Program je v DO-LOOP (neskončna zanka) zanki, kar pomeni, da se izvaja, dokler ne ugasnemo stikala. Konstanti D in F sta uporabljeni zato, da lahko dosežemo v določeni ponovitvi programa drugačno reagiranje robota, s konstanto K pa določamo število ponovitev DO-LOOP zanke klicanega podprograma, kar nam pomeni dolžino premika.

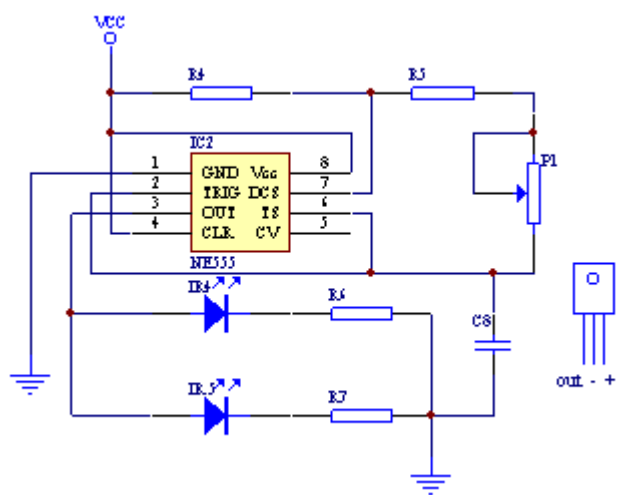
V podprogramu za premikanje robota naprej ali nazaj se pojavi problem, ker moramo enemu motorju priskrbeti 1ms širok pulz, drugemu pa 2ms. Na začetku oba postavimo na logično eno, potem pa po 1ms postavimo na logično 0 najprej prvi motor in po 2ms še drug servo-motor. Ko se bo zanka ponovila, bo zopet na začetku na obeh motorjih +5V in zato pride do razlike za 1ms. Da se izognemo zapletenemu podprogramu sem namesto *waitms 20* napisal *waitms 18* in program prav tako deluje pravilno.

## Zaključek

Atmelov mikrokontroler nam omogoča različne možnosti izboljšave. Z uporabo timerja lahko natančneje določimo širino pulza, zato lahko robotu spreminjamo hitrost gibanja, kar pomeni da se robot lahko ustavlja (pospešuje) počasi. Izdelek lahko poljubno preprogramiramo in ga uporabimo za druge namene (npr. vožnja po dani poti, itd.).

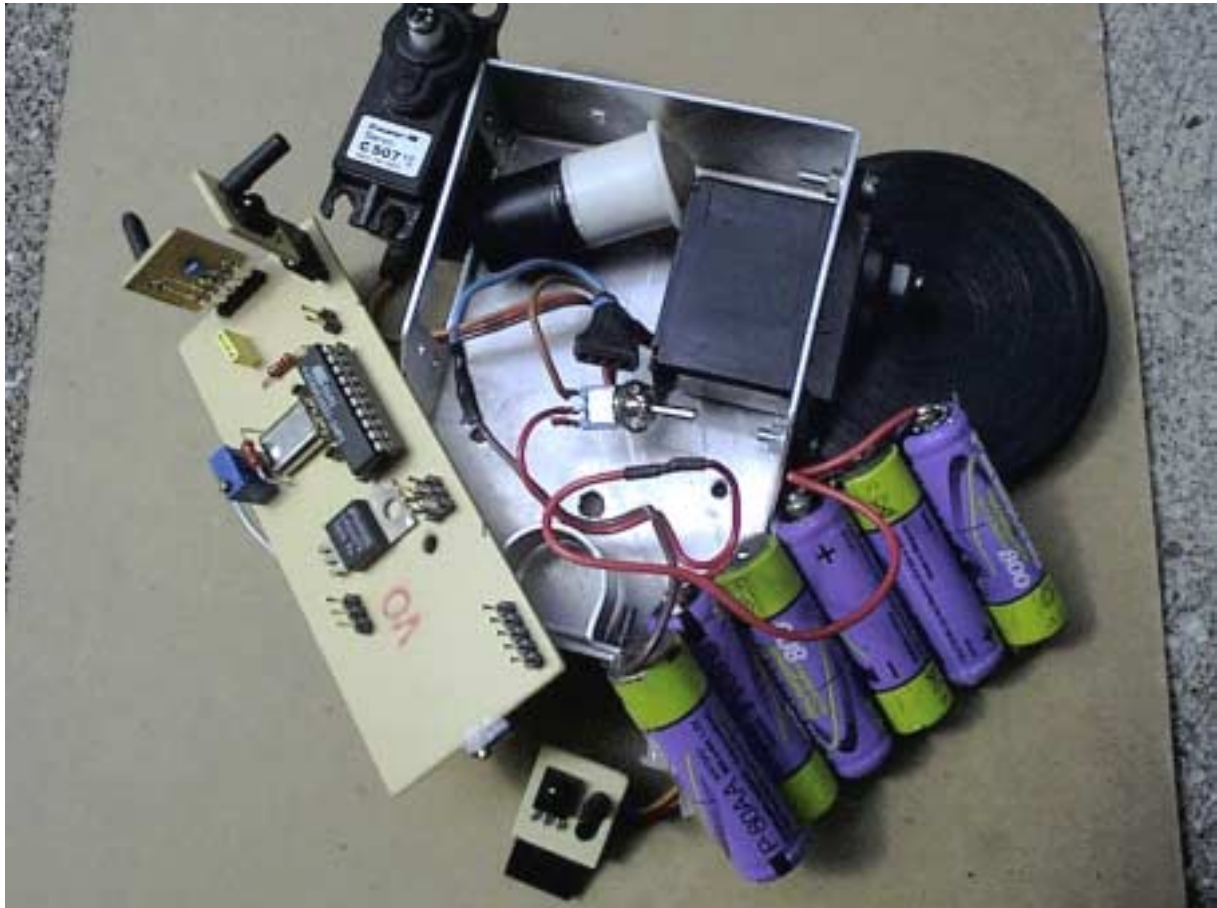


Slika 5: Shema elementov povezanih na mikrokontroler.



Slika 6: Frekvenčni generator in razpored pinov na IR sprejemniku.





**Slika 7:** Pogled na razstavljenega robota

Uroš Derstvenšek , študent 1. letnika UNI programa Elektrotehnika

E-mail: [Uros.Derstvenssek@mailandnews.com](mailto:Uros.Derstvenssek@mailandnews.com)

Mentor: mag. Janez Pogorelc, univ. dipl. inž.

E-mail: [Janez.Pogorelc@uni-mb.si](mailto:Janez.Pogorelc@uni-mb.si)

UNIVERZA V MARIBORU

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>