

Univerzitet u Sarajevu  
Elektrotehnički fakultet u Sarajevu  
Odsjek za automatiku i elektroniku  
*Studijska: 2020/2021. godina*

## Praktikum Elektronike

---

### Zmija1997 igra na PIC16F1939 mikrokontroleru

Student:

**Adnan Bogilović (17976)**

---

Profesor:

**r.prof.dr. Abdulah Akšamović dipl.ing.el.**

---

Sarajevo,  
januar, 2021.

# Sadržaj

<b>1</b>	<b>Sažetak</b>	<b>3</b>
<b>2</b>	<b>Definisanje projektnog zadatka</b>	<b>3</b>
<b>3</b>	<b>Korištene elektroničke komponente</b>	<b>3</b>
3.1	Naponski regulator 7805 . . . . .	3
3.2	Tranzistori 2N2907 (PNP) . . . . .	4
3.3	LE Diode . . . . .	4
3.4	Komandna dugmad . . . . .	5
3.5	Kristalni oscilator 8MHz . . . . .	5
3.6	Konektor za programiranje CONN-D9F . . . . .	5
3.7	Switch 3 terminala . . . . .	6
3.8	PIC16F1939 mikrokontroler . . . . .	6
<b>4</b>	<b>Realizacija softvera igre Zmija1997</b>	<b>7</b>
4.1	Pravila igre . . . . .	7
4.2	Ulazi i izlazi mikrokontrolera . . . . .	7
4.3	Tehnike programiranja pravila igre na mikroprocesoru . . . . .	8
<b>5</b>	<b>Realizacija hardvera igre Zmija1997</b>	<b>9</b>
5.1	Shema projekta Zmija1997 . . . . .	9
5.2	Izrada štampane pločice PCB . . . . .	10
<b>6</b>	<b>Zaključak</b>	<b>11</b>

## 1 Sažetak

Cilj ovoga projekta je da se realizuje popularna igra prošlog stoljeća - Zmija 1997 na odabranom mikrokontroleru PIC16F1939. Ova igra nam je svima poznata sa starih NOKIA 3310 mobilnih telefona. Implementovana su pravila igre i riješeni svi izazovi efikasnog prikaza igre na 8x8 LED ekranu. Igra sadrži šest nivoa težine (brzina kretanja zmije) i odabire sa na početku sesije. Igrom se upravlja pomoću dva dugmeta - desno i lijevo relativno skretanje zmije. Dioda koja u datom momentu prikazuje poziciju hrane za zmiju vidno treperi da bi se razlikovala od tijela zmije. Svaka sesija igre raspolaže sa novim sjemenom nasumičnog generatora brojeva. To sam postigao tako što sam koristio EEPROM memoriju. Ovo daje čar neregulativnosti pozicija hrane.

## 2 Definisane projektnog zadatka

Ovaj projekat je seminarski rad sa predmeta praktikum elektronike i važan je dio shvatanja i samostalnog rada za koji se ovaj predmet zalaže. Projekat se primarno radi zbog prikaza moći mikrokontrolera kao i naše vještine korištenja istog i sastoji od softverskog dijela - C kod i hardverskog dijela - šematski i PCB dizajn kao i izrada štampane pločice i spajanje/lemljenje svih dijelova u cjelinu.

Temu projekta sam izabrao jer sam uvijek htio napraviti ovu igru na ekranu kojeg čine LE Diode i koja se upravlja sa dva click dugmeta, bez arduina ili sličnih general purpose mikrokontrolera koji mogu biti i više nego potrebni (tzv. overkill) po snazi ili glomaznosti nego što nama treba. Zbog toga se arduino više koristi za prototipe ili pri učenju.

## 3 Korištene elektroničke komponente

Za ovaj projekat nije potrebno dosta različitih i/ili nesvakidašnjih komponenti, ali je potrebno prilično dosta istih komponenti. Na primjer LE Dioda nam je potrebno 64.

### 3.1 Naponski regulator 7805

Na samom početku, imamo obavezu napajati sklop sa 12V DC, a naš mikrokontroler radi na 5V, dakle prosto dolazimo do rješenja korištenjem naponskog regulatora 7805 koji snižava napon sa 12V na 5V. Prilično je efikasan, ali na svom kućištu takođe ima i metalni dio koji služi kao odvod toplote kojeg možemo ali i ne moramo zavrnuti na neki disipator toplote. Kondenzatori C1 i C2 su tu iz razloga ako je regulator lociran blizu izvora i poboljšavaju stabilnost i odziv.



Slika 1: Naponski regulator 7805

### 3.2 Tranzistori 2N2907 (PNP)

Jedna od glavnih komponenti za ovaj projekat kao i za svu elektronsku industriju su tranzistori. U ovom projektu je korišteno osam ovih tranzistora koji rade u prekidačkom režimu. Pomoću njih kontroliramo aktivne redove LED matrice u određenom trenutku.

Ovo je PNP tranzistor niske maksimalne voltaže (40 V), ali pristojne vrijednosti struje (600 mA). Ova struja je dovoljna jer kroz tranzistor nikada neće teći struja veća od  $8 \cdot 20 \text{ mA} = 160 \text{ mA}$ . Dakle maksimalna struja će proticati u slučaju kada su sve diode datog reda aktivne. Ovaj tranzistor dolazi u dva tipa pakovanja: TO-92 i TO-18.



Slika 2: Tranzistor 2N2907 (TO-92)

### 3.3 LE Diode

Glavni akteri ovog projekta su svjetleće diode koje daju vizuelni prikaz stanja igre. Aranžirane su u matricu 8x8 dakle potrebno nam ih je 64. One su i jedan od uzroka korištenja dvoslojne izrade pcb pločice iz razloga što su anode svake diode iz istog reda spojene, kao i katode svake diode iste kolone i naravno ne smiju se preklapati što dovodi do potrebe korištenja drugog sloja.



Slika 3: LE Dioda

Za ovaj projekat nije previše bitan model diode, dakle standardna svijetleća dioda 3mm-5mm sa padom napona 2.2V-3.0V pri propusnoj polarizaciji diode. Ovaj napon pri propusnoj polarizaciji nam je dosta bitan iz razloga što će katode, dakle kolone matrice dioda biti spojene preko odgovarajućih otpora sa PORTB. Otpore ćemo birati tako da struja koja ulazi u mikrokontroler na jednom pinu porta B ne bude veća od 20 mA (max. 25 mA).

Računamo otpornik diode na sljedeći način:

$$R_d = \frac{5V - 0.2V - 2.2V}{20mA} = 130\Omega \quad (1)$$

Gdje su: 5 V napon sa naponskog regulatora, 0.2 V napon  $V_{ec}$  pnp tranzistora odgovarajućeg reda, a 2.2 V napon pri propusnoj polarizaciji LE Dioda.

### 3.4 Komandna dugmad

Da bismo upravljali kretnjom zmije ili izborom težine igre na samom početku, koristimo dva dugmeta i to jedan za lijevo skretanje i jedan za desno skretanje zmije. Skretanje je relativno u odnosu na trenutni pravac i smijer, zbog čega se morala dizajnirati određena logička funkcija koju ćemo poslije spomenuti.

Prilikom pritiska dugmeta, na odgovarajućem pinu mikrokontrolera se mijenja napon sa 0V na 5V, a električni krug se zatvori preko odgovarajućeg otpornika ka GND.



Slika 4: Dugme

### 3.5 Kristalni oscilator 8MHz

Kristalni oscilator, osmo megahercni koristimo kao vanjski (primarni) takt našeg mikrokontrolera. To znači da jedan instrukcijski ciklus na ovom mikrokontroleru traje četiri puta duže od perioda kristalnog oscilatora  $4 \cdot T_o = 0.5 \text{ us}$ .

Odgovarajuće kondenzatore C3 i C4 koristimo da rezoniraju s induktivitetom kristala i uzrokuju titranje kristala u svom osnovnom paralelno-rezonantnom modu.

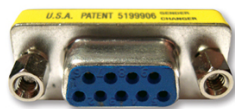


Slika 5: Kristalni oscilator 8 MHz

### 3.6 Konektor za programiranje CONN-D9F

Ovaj konektor koristimo kako bismo programirali naš mikrokontroler serijskom komunikacijom sa računara. Stariji računari podržavaju ovakav konektor. Postoje noviji programatori na usb (PicKit 2,3..).

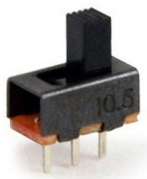
Od ponuđenih devet pinova koristimo pet. Većinu pinova vežemo preko 3-terminal malih prekidača, da bismo bili u stanju lako prelaziti iz moda programiranja u mod aktivnog rada mikrokontrolera.



Slika 6: Konektor CONN-D9F

### 3.7 Switch 3 terminala

Ove prekidače koristimo da bismo lako prelazili iz moda programiranja mikrokontrolera u mod aktivnog stanja istoga.



Slika 7: Prekidač 3-terminala

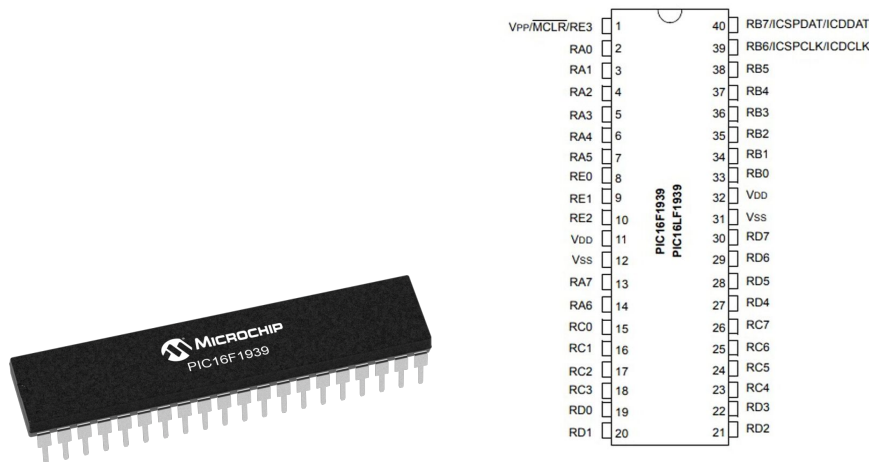
### 3.8 PIC16F1939 mikrokontroler

Glavni dio projekta jeste PIC16F1939 mikrokontroler. On je 8-bitni, (dakle memorijski registri su mu osmобitni) i koristi se u razvoju elektronskih aplikacija. Dakle programski kod koji je pisan u C-u se prvo prevede u assembly, a zatim u mašinski kod (.hex files) nakon čega ga uploadamo u mikrokontroler. Razvojno okruženje koje koristimo pri pisanju programa je MPLAB IDE.

Neke od glavnih specifikacija su mu:

- Besplatno razvojno okruženje MPLAB IDE.
- Sadrži dosta potrebnih periferija na čipu
- Programira se jednostavno
- Instrukcije jednociklusne i dvociklusne (grananja)
- Unutarnji 32MHz oscilator
- Jako male snage
- Četiri 8-bitna i jedan 16-bitni tajmer
- Širok opseg napona rada (1.8V-5.5V)
- Četrnaest kanalni 10-bitni ADC
- 1-UART, 1-SPI, 1-I2C1-MSSP(SPI/I2C) komunikacije
- Dva komparatora
- SRAM 1024 B, EEPROM 256 B

- Struja max. 25mA Source/Sink I/O



Slika 8: Mikrokontroler i njegova schema pinova

## 4 Realizacija softvera igre Zmija1997

### 4.1 Pravila igre

Pravila igre Zmija1997 su sljedeća:

- Zmija se kreće određenim pravcem i smjerom
- Pravac i smjer se mijenja ulaznim komandama
- Zmija kreće sa veličinom 1 i sakuplja hranu da bi se povećavala
- Pozicije hrane i početna pozicija zmije su nasumične
- Zmija se može teleportirati na suprotni kraj ekrana kada izađe van opsega ekrana/mape
- Igrač je pobijedio kada dostigne maksimalnu veličinu zmije (64)
- Igrač je izgubio kada zmija ugrize/udari u svoj dio tijela
- Težina igre (brzina zmije) se bira na početku svake igre

### 4.2 Ulazi i izlazi mikrokontrolera

Ulazi:

- PORTD (2 bita)

Da bismo kontrolirali zmiju potrebni su nam neki ulazi. Naime ova aplikacija zahtjeva 2 ulazna bita PORTD-a za dva dugmeta. Desno dugme je spojeno na RD0, a lijevo na RD1. Kada je logička jedinica na nekom od ovih pinova to znači da treba doći do odgovarajuće promjene pravca i smjera zmije.

Izlazi:

- PORTB (8 bita)

- PORTC (8 bita)

Ovi portovi se koriste za upravljanje aktivnog reda (PORTC) i njegovih kolona u datom trenutku (PORTB). Da bi neka dioda bila aktivna na (m,n) poziciji, onda se m-ti bit porta C postavlja na logičku nulu (0V) kao i n-ti bit porta B takođe na logičku nulu (0V). Naravno suprotno tome ako ne želimo da neka dioda bude aktivna u nekom trenutku onda postavljamo jedan od navedenih bita na 5V.

### 4.3 Tehnike programiranja pravila igre na mikroprocesoru

Za osvježavanje ekrana korištena je poznata tehnika osvježavanje 'red po red'. U nekom momentu takođe mogu biti aktivne sve diode nekog reda. Brzina osvježavanja ekrana je podesiva u kodu. Pri fizičkoj izvedbi korišću 25-30 FPS (frames per second) brzinu osvježavanja ekrana, što znači da će svaki red biti aktivan  $1/8 * 1/25$  sekundi.

Ova brzina od 25-30 FPS se koristi iz razloga što tromost oka ne primijeti tako brze promjene i samim tim neće uočiti brza paljenja i gašenja LE Dioda - izgledaće nam sasvim kontinualno i prirodno (kao i svaki drugi ekran koji mora raditi na principu osvježavanja).

Zmija se u nekom trenutku kreće jednim od četiri moguće kombinacije smijera i pravca. Možemo ga mijenjati sa ulaznim dugmadima koji su povezani na RD0 i RD1. Da igra ne bi zahtjevala četiri dugmeta, pritiskom na desno ili lijevo dugme moguće je promijeniti pravac i smijer zmije u datu stranu relativno prethodnom pravcu. Za to koristimo logičku funkciju.

Varijable i njihova značenja su: dpravac (trenutni pravac) može biti 0 (horizontalan) ili 1 (vertikalni), dsmijer (trenutni smijer) može biti 0 (negativan) ili 1 (pozitivan), bit B može biti 0 - pritisnuto lijevo dugme ili 1 - pritisnuto desno dugme.

a = dpravac, b = dsmijer, c = B

dsmijer\_novo =  $f(dpravac, dsmijer, B) = !a!bc + !ab!c + a!b!c + abc$

dpravac\_novo =  $!dpravac$

a	b	c	Output
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Slika 9: Logička funkcija za određivanje novog smijera

Pozicije dijelova zmije čuvamo u nizu parova x,y koordinata, odnosno red,kolona i to u formatu 2n tipa (32,16) umjesto (5,4) iz razloga što poslije pri osvježavanju ekrana samo saberemo kolone svih dijelova zmije čiji red odgovara trenutnom redu koji se osvježava. Zatim takav broj (sumu) prosljedimo negiran binarno na LATB. Kao i binarno negiran broj aktivnog reda na LATC. Hrana se takođe sabere sa pomenutom sumom ako je došao ciklus prikazivanja hrane i ako je



hrana pomenutom redu.

Radi zanimljivije igre isprogramiran je i animirani obrazac na prelazima igre, kao i obrasci (smješko) pri pobjedi i (ljutko) pri porazu. Težina igre jeste zapravo tromost zmiје. Ako je na početku sesije izabrana teža igra onda smo smanjili varijablu tromosti zmiје, dakle zmiје će se kretati brže.

Sve postavke (osvježavanje ekrana, treptanje diode hrane, težina igre) se moraju naknadno podesiti za fizički sklop. U simulaciji je to mnogo sporije iz razloga što se ne može lijepo vizuelno simulirati tako brz refresh rate.

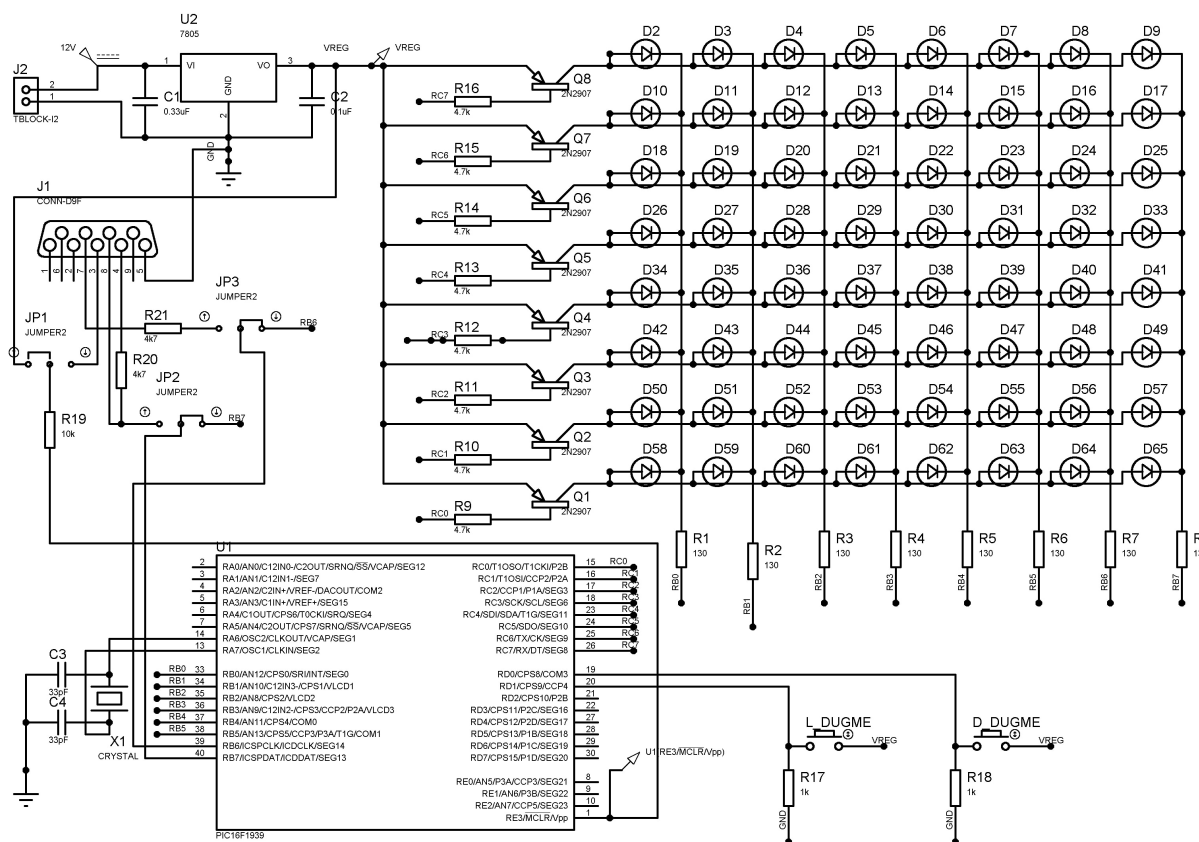
Nasumične pozicije hrane svake sesije se dobiju tako što se iz EEPROM-a uzima prošla vrijednost nasumičnog broja, zatim inicijaliziramo generator nasumičnih brojeva sa time, a potom spasimo u EEPROM novu nasumičnu vrijednost za idući put. Teoretski moguće je 256 konfiguracija jer toliko može postojati različitih 8-bitnih brojeva kao sjeme generatora.

C kod (komentiran) se nalazi u prilogu iz razloga što bi pdf bio jako dug kada bismo obradili sve dijelove koda ovdje.

## 5 Realizacija hardvera igre Zmija1997

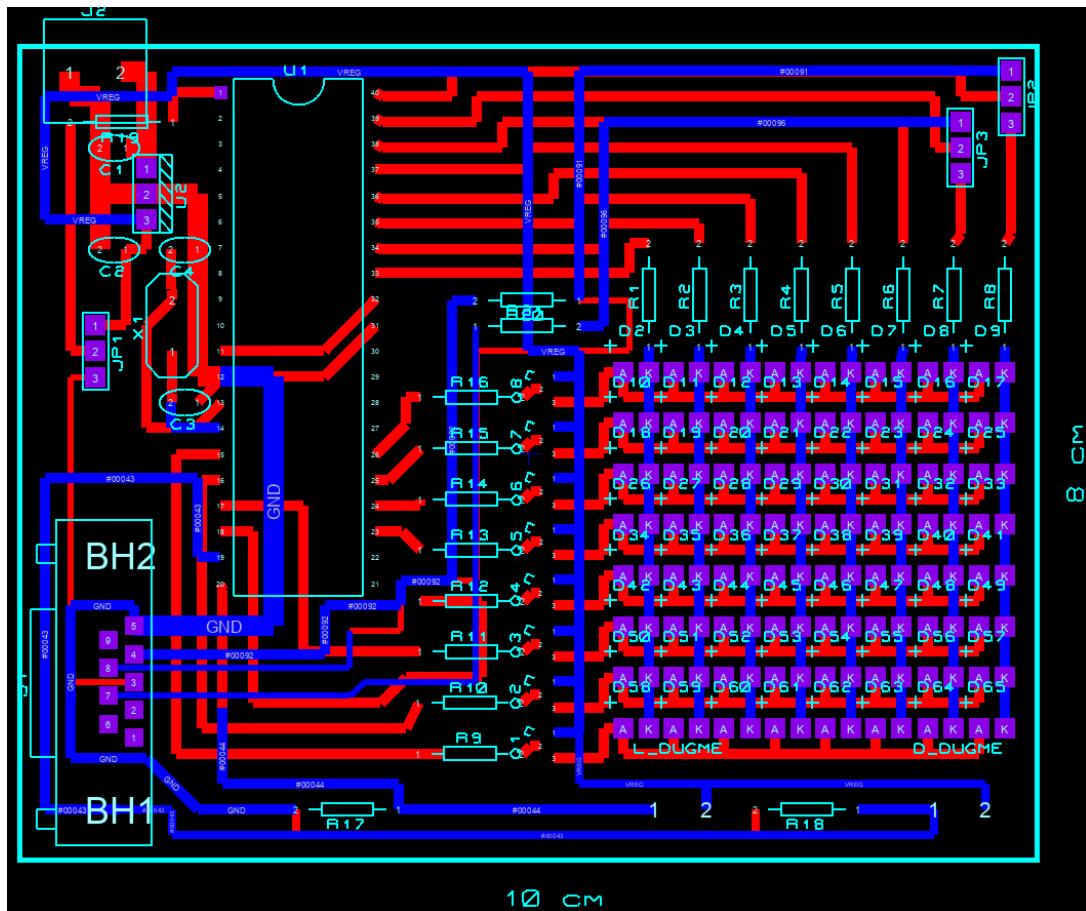
### 5.1 Shema projekta Zmija1997

Na sljedećoj slici vidimo kompletnu shemu projekta Zmija1997 na mikrokontroleru PIC16F1939.

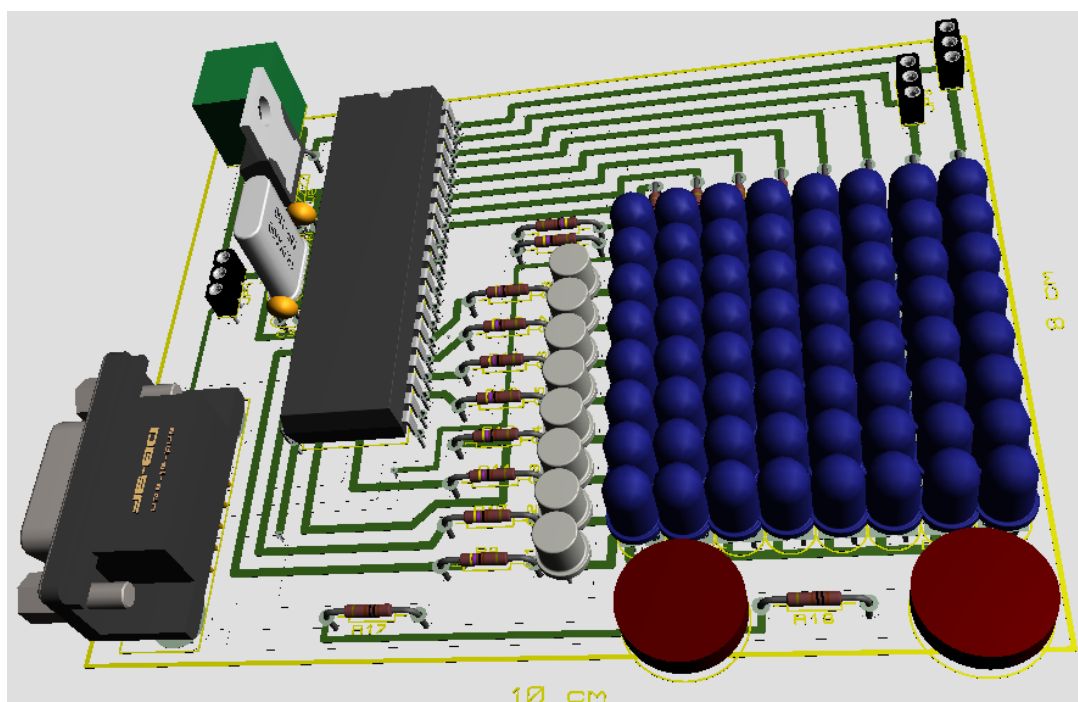


Slika 10: Shema projekta

## 5.2 Izrada štampane pločice PCB



Slika 11: PCB dizajn projekta (dvoslojni)



Slika 12: 3d prikaz projekta

## 6 Zaključak

Zmija1997 je jedan od mojih dražih urađenih projekata i planiram da ga uradim i u fizičkoj formi. Naime ovaj predmet me je naučio koristiti mikrokontrolere za razne aplikacije i svrhe koje mogu susresti u budućnosti. U budućnosti se može nadograditi program sa još nekim igrama koje se mogu prikazati na 8x8 matrici LE Dioda. Komentirani C kod i Proteus simulacioni fajl / PCB dizajn su dostupni u prilogu, kao i oba sloja za izradu PCB-a.