

Elektrotehnički fakultet  
Patre 5, Banja Luka  
**[www.etf.unibl.org](http://www.etf.unibl.org)**

# MIKROPROCESORSKI SISTEMI 2

## CONNECT FOUR

*Dejan Đekanović*

# CONNECT FOUR

## Sadržaj

|         |   |
|---------|---|
| Uvod    | 2 |
| Hardver | 3 |
| Softver | 6 |

## Uvod

Tema projekta je izrada *Connect four* igre upotrebom mikrokontrolera *8051* i *8x8* matričnog displeja. Potrebno je projektovati odgovarajući hardverski dizajn, zatim napisati softver koji će upravljati datim hardverskim komponentama. Za realizaciju datog projekta potrebne su odgovarajuće hardverske komponente, softverski alati, kompajleri, simulatori o čijim detaljima ćemo u nastavku.

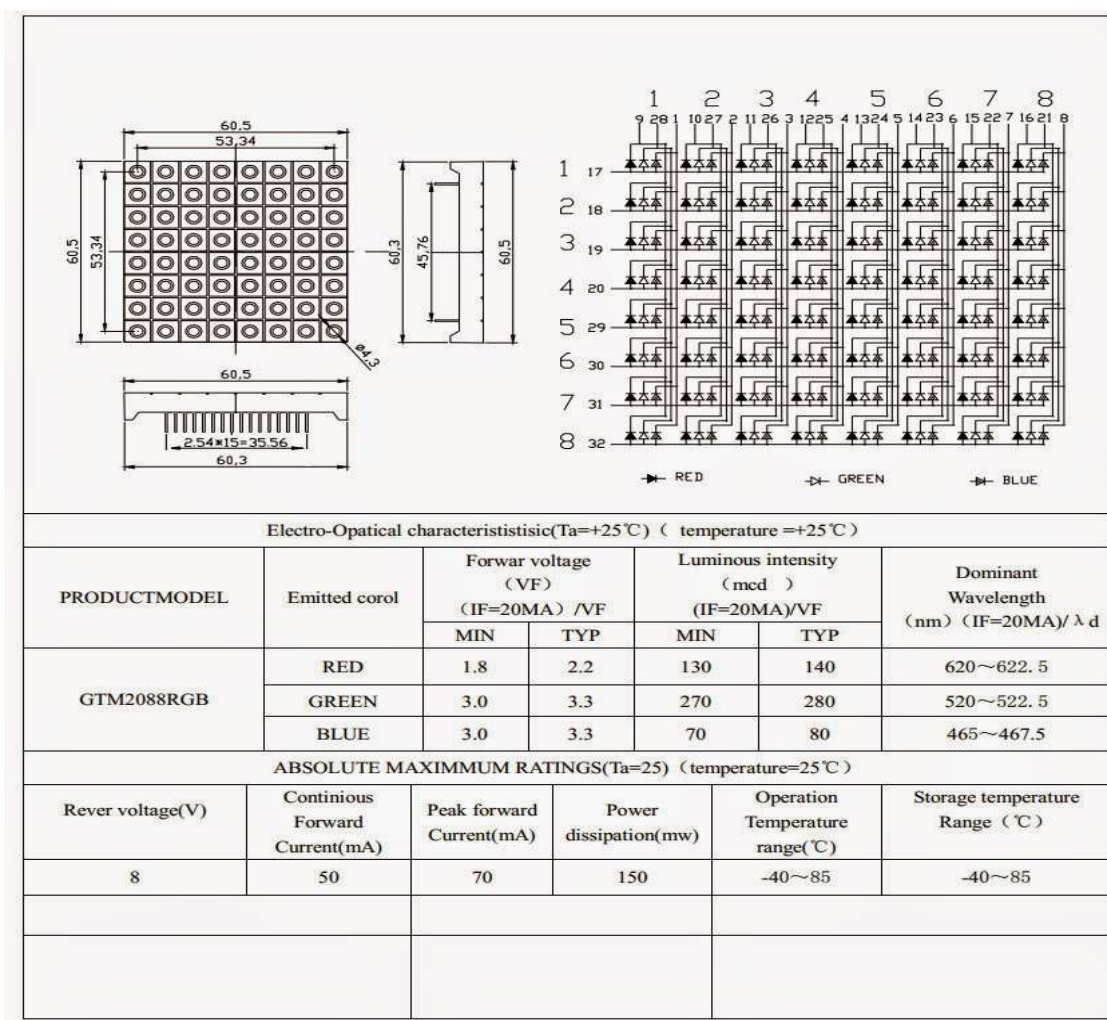
Prije samog ulaska u detalje implementacije datog problema potrebno je objasniti kako funkcioniše sama igra. Igra je osmišljena tako da dva igrača ubacuju svoje diskove, koji su različite boje (obično žuti i crveni), u tablu koja ima šest redova i sedam kolona. Diskovi se ubacuju sa gornje strane ploče u jednu od kolona. Cilj igre je da se sklope četiri diska jedan do drugog vertikalno, horizontalno ili po dijagonalama.

Ova igra spada u skupinu takozvanih *solved* igara, jer prvi igrač može uvijek da pobijedi izborom pravih poteza. Postoje mnoge verzije ove igre sa različitim veličinama ploča, i različitim pravilima, ali najkorištenija je ona sa tablom od *7x6* kolona u kojoj se moraju sklopiti diska zajedno.

## Hardver

U zadatku je navedeno da se igra realizuje pomoću mikrokontrolera iz familije 8051 i 8x8 matričnog displej. Što se tiče samog mikrokontrolera koristili smo verziju koji ima četiri izlazna porta.

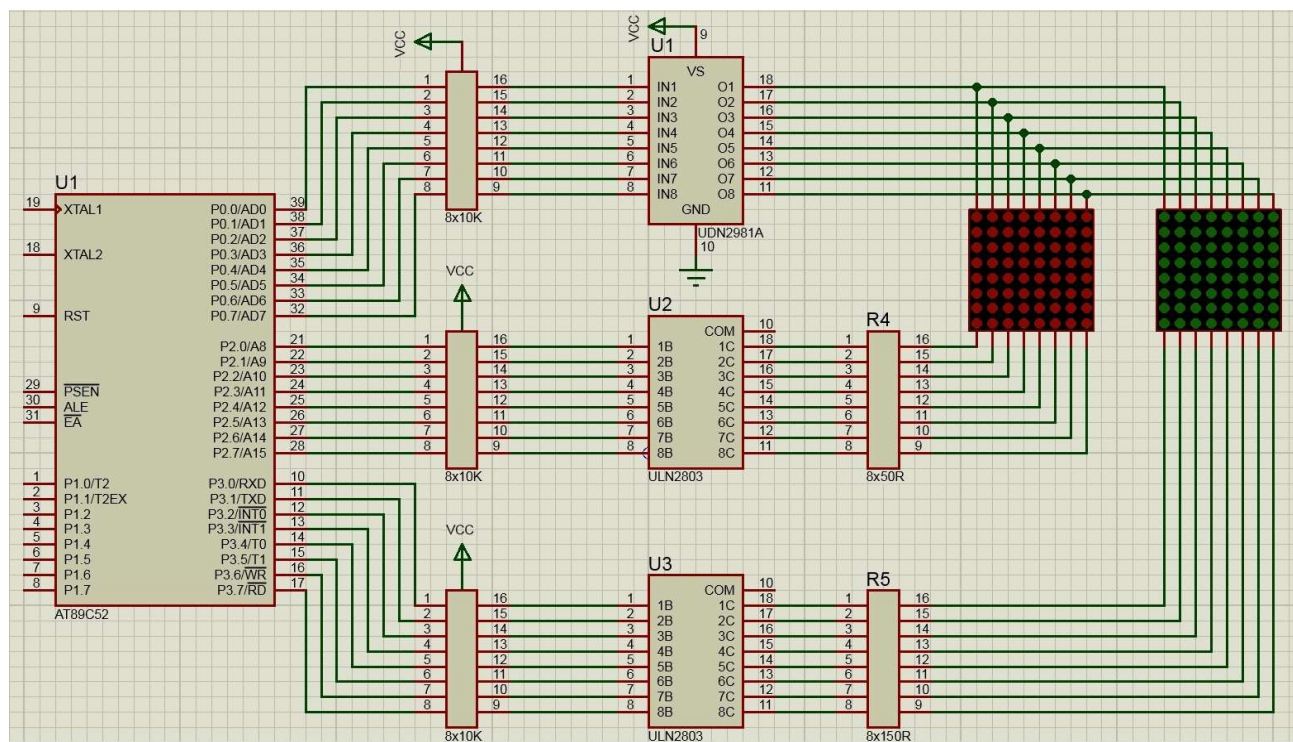
Za prikaz slike koristi se 8x8 matrični displej oznake 2088RGB-5. Da bi prikazali sliku na datom displeju potrebno je vršiti multipleksiranje, iz razloga što u jednom trenutku može biti aktivan samo jedan red. To se realizuje softverski o čemu ćemo kasnije detaljnije pričati.



Slika 1.2 – Matrični displej 2088RGB-5 izgled i specifikacije

# CONNECT FOUR

Dati displej posjeduje osam izlaza za selektovanje odgovarajućeg reda, po osam izlaza za selektovanje kolone sa crvenom, plavom i zelenom diodom, tj. ukupno trideset dva izlaza. Kako su nam za datu igru potrebne samo crvena i zelena boja, i njihova kombinacij žuta, možemo eliminisati plavu boju. Izbacivanjem plave boje ostaje nam dvadeset četiri izlaza na displeju koje je potrebno na neki način povezati sa mikrokontrolerom. Dati mikrokontroler posjeduje četiri porta sa po osam izlaza/ulaza, međutim jedan port je potrebno iskoristiti za povezivanje tastature, tako da ostala tri porta možemo iskoristiti za povezivanje displeja sa mikrokontrolerom.



Slika 1.1 – Šema spajanja odgovarajućih hardverskih komponenti

Povezivanje mikrokontrolera sa displejom nije moguće izvršiti direktno jer mikrokontroler daje veoma malu izlaznu snagu, nekoliko  $mW$ , dok je snaga koja je potrebna da dioda svijetli zadovoljavajućim intenzitetom nekoliko stotina  $mW$ . Iz navednog razloga se koriste tzv. drajveri koji služe da se pomoću male ulazne snage kontrolišu odgovarajući izlazi koji mogu da daju dosta veće snage.

Za povezivanje mikrokontrolera sa displejom koriste se drajveri oznake *UDN2981A* i *UNL2803A*. Prvi drajver se koristi za povezivanje sa redovima matričnog displeja, dok se drugi tip drajvera koristi za povezivanje sa kolonama odgovarajuće boje, za svaku boju po jedan. Razlika između ova dva tipa drajvera je u tome što prvi tzv. *source type*, ukoliko je na nekom od ulaza logička jedinica, spaja odgovarajući izlaz na *Vcc*, dok drugi tzv. *sink type*, ukoliko je na nekom od ulaza logička jedinica, spaja odgovarajući izlaz na masu.

Diode imaju veoma malu otpornost, pa bi njihovim direktnim povezivanjem na izvor napajanja došlo do pojave prevelike struje, što bi dovelo do njihovog uništenja. Iz tog razloga potrebno je dodati otpornike koji će ograničiti struju kroz date diode.

Za zelenu boju koriste se otpornici od  $150\Omega$ , dok se za crvenu koriste oni od  $50\Omega$ . Razlog tome je što zelena dioda daje jači intenzitet svjetlosti od crvene, pri istim strujama. Takođe, pad napona na zelenoj i crvenoj diodi nisu isti i iznose  $3,3V$  za zelenu i  $2,2V$  za crvenu diodu.

Prilikom realizacije datog zadatka korištena je *Easy8051 v6* razvojna ploča. Međutim, nije neophodno koristiti datu ploču. Umjesto toga potreban je oscilator od  $10MHz$  koji služi za generisanje takta, naponski izvor od  $5V$  i programator za *8051* familiju mikrokontrolera preko koga je moguće "flešovati" programski kod na dati mikrokontroler.

## Softver

Nakon projektovanja i spajanja odgovarajućeg hardvera potrebno je realizovati softver koji će upravljati radom navedenog hardvera. Pri izradi softverskog rješenja korišteni su *MikroC* razvojno okruženje i kompajler, dok je za “flešovanje” koda korišten *8051FLASH* softver. Radi lakšeg održavanja i ponovne upotrebljivosti programskog koda poželjno je isti organizovati u module.

Programsko rješenje za dati zadatak sastavljeno je iz četiti modula:

- **timer.c** – moduo za mjerenje proteklog vremena
- **display.c** – drajver za matrični displej
- **graphics.c** – moduo za apstrakciju prikaza grafičkih elemenata
- **connecfour.c** – moduo za apstrakciju igre

### TIMER.C

```
void Timer_Init();  
unsigned long Timer_Miliseconds();  
void Timer_Tick();
```

Moduo za mjerenje proteklog vremena koristi tajmer sa oznakom *T1* koji u odgovarajućim vremenskim intervalima, u ovom slučaju svake milisekunde, generiše prekid. Vremenski interval u kojima se dešava prekid zavisi od vrijednosti registara *TL1* i *TH1*. Funkcija *Timer\_Init* služi za inicijalizaciju navedenih vrijednosti, kao i za podešavanje odgovarajućih registara, kao što su *IE* (omogućavanje prekida) i *TMOD* (mod rada tajmera). Pozivom ove funkcije startuje se odgovarajući tajmer.

Funkcija *Timer\_Miliseconds* vraća trenutni broj milisekundi proteklih od trenutka startovanja tajmera. Funkcija *Timer\_Tick* predstavlja prekidnu rutinu koja se poziva kada se desi prekid odgovarajućeg tajmera. Prilikom njenog poziva inkrementuje se trenutni broj milisekundi.

# CONNECT FOUR

## DISPLAY.C

```
void Display_Init();  
void Display_SetOutput(unsigned char row, unsigned char red,  
                        unsigned char green);  
void Display_Refresh();
```

Drajver za matrični displej koristi tajmer sa oznakom *T0* koji u odgovarajućim vremenskim trenucima, u ovom slučaju svake dvije milisekunde, generiše prekid. Funkcija *Display\_Init* služi za startovanje odgovarajućeg tajmera.

Kao i u prethodnom modulu funkcija *Display\_Refresh* predstavlja prekidnu rutinu koja se izvršava kada se desi prekid na odgovarajućem tajmeru. Prilikom njenog poziva postavljaju se odgovarajuće vrijednosti izlaznih portova pomoću metorde *Display\_SetOutput*. U ovom slučaju vrši se multipleksiranje displeja koje je neophodno jer se radi o matričnom displeju.

Svake 2ms se dešava prekid, tako da dobijamo da je za iscrtavanje čitavog displeja potrebno 16ms, što je nešto više od 60 slika u sekundi. Da ne bi primijetili treperenje slike potrebno je da broj slika u sekundi bude veci od 25-30.

## GRAPHICS.C

```
void Graphics_SetPixel(unsigned char row, unsigned char column, Color color);  
void Graphics_VerticalLine(unsigned char column, unsigned char rowStart,  
                           unsigned char rowEnd, Color color);  
void Graphics_HorizontalLine(unsigned char row, unsigned char columnStart,  
                             unsigned char columnEnd, Color color);  
void Graphics_ClearDisplay();
```



# CONNECT FOUR

Moduo za apstrakciju prikaza grafičkih elemenata posjeduje funkciju *Graphics\_SetPixel* za iscrtavanje piksela u odgovarajućem redu i koloni i sa određenom bojom. Ona u zavisnosti od proslijeđenih parametara postavlja vrijednosti tzv. video memorije. Za iscrtavanje horizontalnih i vertikalnih linija koriste se metode *Graphics\_HorizontalLine* i *Graphics\_VerticalLine*, koje kao parameter primaju odgovarajuće koordinate za iscrtavanje i boju linije. *Graphics\_ClearDisplay* služi za brisanje sadržaja displeja, odnosno postavljanje svih piksela na crnu boju.

## CONNECTFOUR.C

```
void ConnectFour_Init();
unsigned char ConnectFour_InsertDisc(unsigned char column);
void ConnectFour_SwitchPlayer();
unsigned char ConnectFour_CheckWinner();
void ConnectFour_TurnWinnersDiscs(unsigned char onOff);
void ConnectFour_NewGame();
```

Navedeni moduo sadrži metode za inicijalizaciju igre *ConnectFour\_Init*, umetanje diska u odgovarajuću kolonu *ConnectFour\_InsertDisc*, prebacivanje na narednog igrača *ConnectFour\_SwitchPlayer*, uključivanje i isključivanje pobjednikovih diskova *ConnectFour\_TurnWinnersDiscs* i metodu za startovanje nove igre *ConnectFour\_NewGame*.

Implementaciju navedenih finkcija, slike, video, *doxy* dokumentaciju koda i ostalo, možete naći u *github* repozitorijumu na sledećem linku [www.github.com/djn21/ConnectFour](https://www.github.com/djn21/ConnectFour).