

Uvod

Postoji velika razlika između mikrokontrolera i mikroprocesora. Najvažnija razlika je njegova funkcionalnost. Da bi mikroprocesor uopšte mogao da se upotrebi, potrebno je da sadrži i neke druge komponente kao što su memorije i komponente za prijem i slanje podataka. Mikroprocesor se smatra „srcem“ mikrokontrolera, zbog toga što je mikrokontroler dizajniran da sadrži sva kola koja su neophodna, sa pripadajućim periferijama, tako da mu nisu potrebna dodatna spoljašnja kola.

Mikrokontroler koji se koristi za ove vežbe je 18F25K22, a okruženje je Ready for PIC.

Da bi mikrokontroler imao primenu, mora biti povezan sa dodatnom elektronikom (električnim kolima). Ta dodatna kola povezuju se sa mikrokontrolerom preko pinova, a pinovi mikrokontrolera su grupisani u posebne registre koji se nazivaju portovi (A,B,C,D). U okruženju koje koristimo na ovim vežbama pinovi mikrokontrolera izvučeni su sa strane radi lakšeg rada.

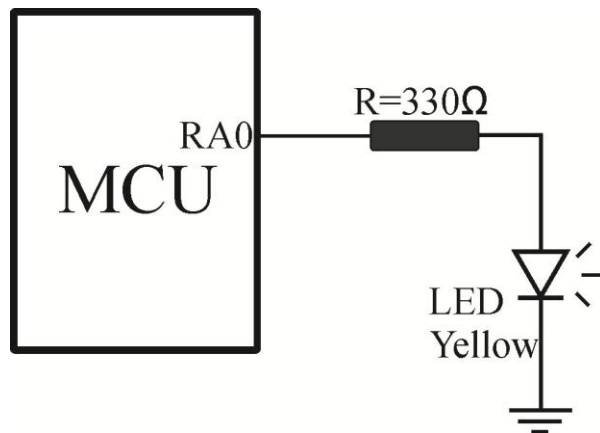
Svaki pin mikrokontrolera može biti:

- ulazni (očitanje spoljašnjeg stanja)
- izlazni (generisanje digitalnog stanja)
- korišćenje određenih pinova za A/D konverziju (pogledati datasheet)

*Pinovi su ulazni kada je za realizaciju nekog projekta potrebno da se očitava vrednost spoljašnjih signala, dok se za aktuatorski deo projekta koriste izlazni pinovi (u ovoj vežbi rad LED diode).

TREPTAČ

Potrebno je napraviti treptač koji trepće svakih 400 ms.



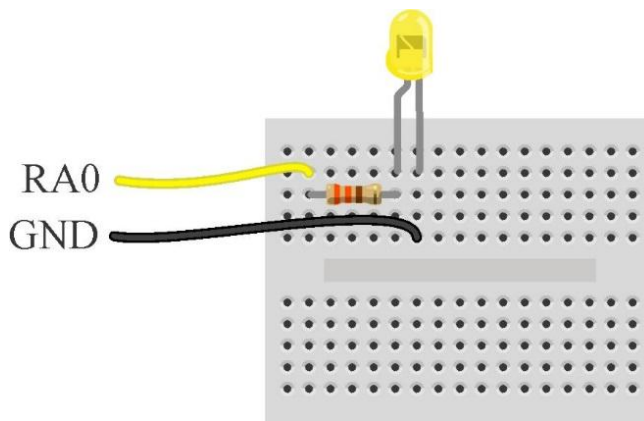
Slika 1: Električna šema treptača

Za ovu vežbu potrebno je povezati otpornik i LED diodu prema električnoj šemi sa slike. LED dioda će za ovu vežbu biti spojena na pin RA0. Pre početka pisanja koda programa, potrebno je da se zna da li je pin na kojem se nalazi dioda ulazni ili izlazni. S obzirom da nam je potrebno da određenim naponom delujemo na diodu, a ne želimo da merimo neki spoljašnji signal, potrebno je da pin na diodi bude izlazni (OUTPUT).

Pri pisanju koda programa potrebno je razlikovati PORT registar i TRIS registar.

PORT registar obezbeđuje da li se na određeni pin dovodi napon ili ne. Ako se dovede napon, pin će imati vrednost logičke 1, dok je odsustvo napona predstavljeno logičkom 0.

TRIS registar označava da li su pinovi ulazni ili izlazni. Logičkom 1 se označava da je pin ulazni, dok je za logičku 0 pin izlazni.



Slika 2: Povezivanje kola na protoploči

VAŽNO: Kada se povezuje kolo na protoploči, obavezno se mora okruženje isključiti sa USB napajanja!!!

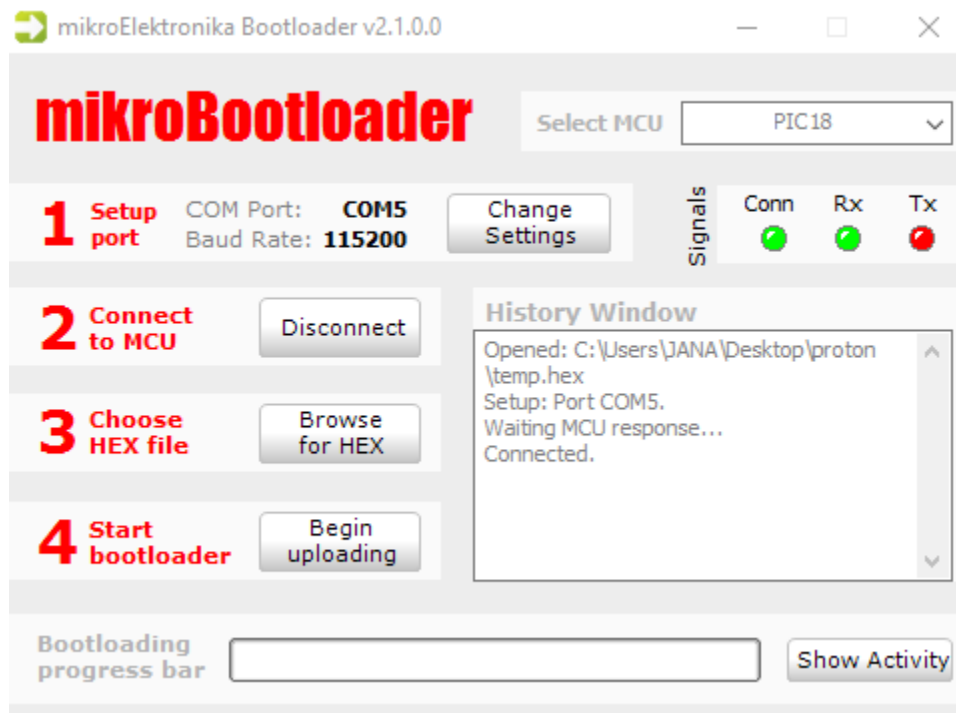
Programski kod

```

Device 18F25K22
Xtal 32
ANSELA=%00000000    `svi pinovi su isključeni sa A/D konvertora
TRISA=%11111110    `podešavanja I/O pinova, pin RA0 je izlazni
/=====
Start:                `naziv labele
/=====
PORTA.0=1            `uključi LED diodu
DelayMS 400          `čekaj 400 ms
PORTA.0=0            `isključi LED diodu
DelayMS 400          `čekaj 400 ms
GoTo Start           `vrati se u petlju
End

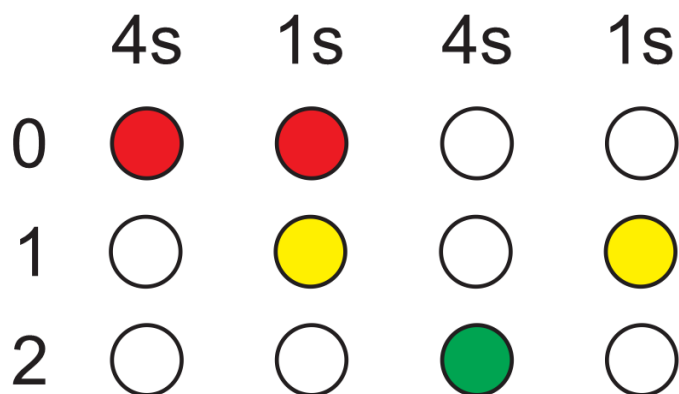
```

Nakon pisanja programa, potrebno je njegovo kompajliranje. Pokreće se mikroBootloader i prate se koraci za povezivanje sa okruženjem:

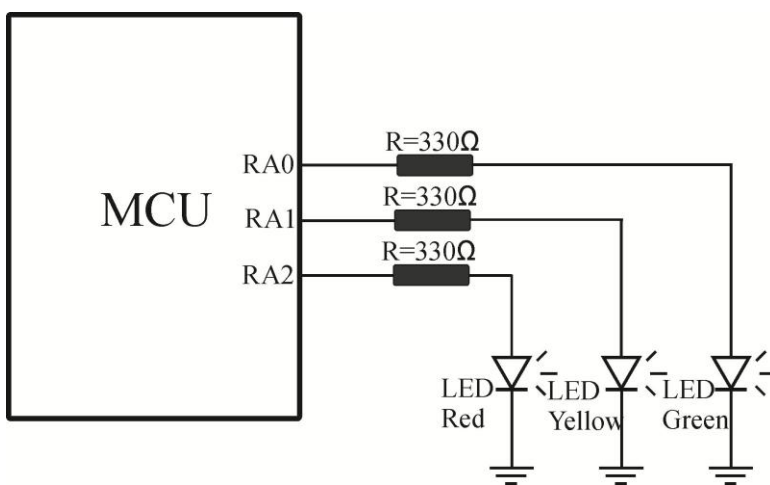


SEMAFOR

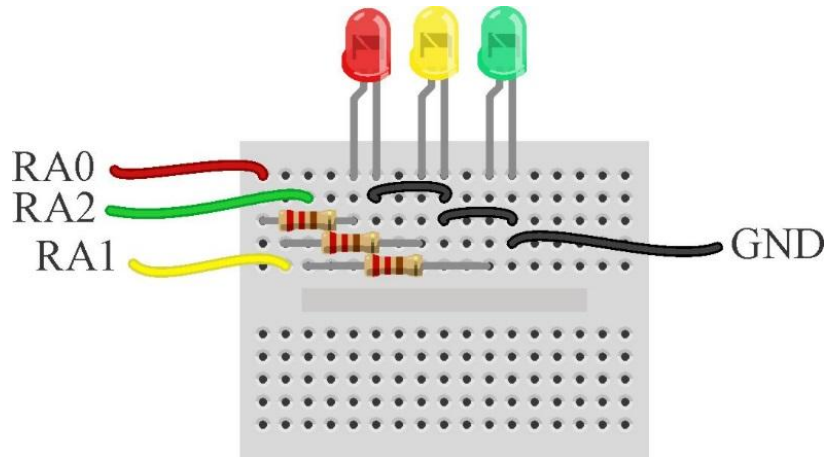
Drugi primer je pravljenje semafora. Potrebno je uključivati LED diode kao što je prikazano na slici 3, što predstavlja rad realnog semafora. Vreme koje je potrebno postaviti za crvenu i zelenu LED diodu je 4 s, a vreme za menjanje stanja između crvene i zelene je 1 s.



Slika 3: Rad semafora



Slika 4: Električna šema semafora



Slika 5 Povezivanje kola na protoploči

Programski kod

```
Device 18F25K22  
Xtal 32
```

```
ANSELA=%00000000  
TRISA=%11111000
```

```
Start:
```

```
PORTA=%00000001  
DelayMS 4000  
PORTA=%00000011  
DelayMS 1000  
PORTA=%00000100  
DelayMS 4000  
PORTA=%00000010  
DelayMS 1000  
GoTo Start  
End
```

```
`svi pinovi su isključeni sa A/D konvertora  
`RA0,RA1 i RA2 su postavljeni kao izlazni
```

```
`uključiti crvenu LED diodu (RA0)  
`čekaj 4 s  
`uključiti crvenu i žutu LED diodu (RA0 i RA1)  
`čekaj 1 s  
`uključiti zelenu LED diodu (RA2)  
`čekaj 4 s  
`uključiti žutu LED diodu (RA1)  
`čekaj 1 s
```

Kada se piše kod, postoji više načina za predstavljanje logičkih stanja pinova (primeri prikazani konkretno za semafor).

FOR petlja

Ako želimo da nam kao na realnom semaforu zelena LED dioda trepće nekoliko puta, to je moguće uraditi na nekoliko načina. Prvi način je da ponovimo naredbe za uključivanje i isključivanje diode npr. tri puta:

```
PORTA=%00000100
DelayMs 400
PORTA=%00000000
DelayMs 400
PORTA=%00000100
DelayMs 400
PORTA=%00000000
DelayMs 400
PORTA=%00000100
DelayMs 400
PORTA=%00000000
DelayMs 400
```

Pisanje programa za ovakav slučaj je moguće, ali ako bismo želeli 200 puta da ponovimo neku naredbu, onda ovakvo predstavljanje postaje besmisleno. U ovakvim slučajevima se koriste petlje. S obzirom da se zna da želimo 200 puta da ponovimo istu naredbu, iskoristiće se FOR petlja.

Primer 1: Ponavlja se deo programa u opsegu od 1 do 200, za promenljivu **X**

```
For X=1 To 100 (Step 1)
```

```
... kod
```

```
Next X
```

Step predstavlja korak brojača i ako se ne spomene važi da je korak 1. Korak brojača može biti bilo koji broj, u zavisnosti od potrebe. Brojanje je takođe moguće i unazad, ali je tada bitno da se stavi **Step -1**. Kada bi se samo stavilo da se broji od npr. 200 do 1, brojač bi pokušao da broji unapred, bez obzira na napisano.