## Računala i procesi

## 1. domaća zadaća

Ema Popović 0036506085 20. 10. 2020.

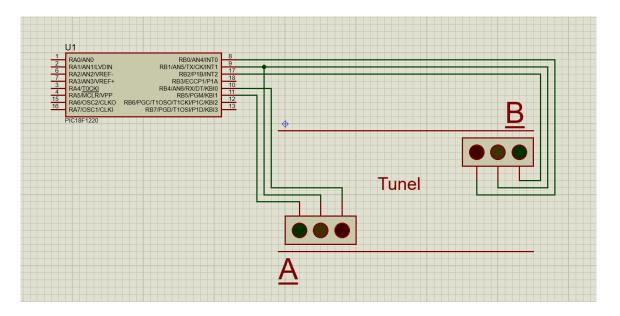
1. U lookup table mehanizmu u dio koda koji opisuje tablicu se upisuje niz konstanti oblika "retlw [konstanta]". U radni registar W se sprema redni broj željene konstante (počinju od 0). Poziva se tablica te se pomoću vrijednosti iz W računa koja je željena konstanta ("addwf PCL, 1"). Programskom brojilu se dodaje potrebni odmak od početka tablice. Npr. ako je vrijednost u W jednaka 2, odabire se treća konstanta. Željena vrijednost se kroz retlw sprema u registar W i nastavlja se izvoditi glavni program.

Ovaj mehanizam štedi vrijeme jer je alternativa često kompleksniji izvorni kod. Ranije upoznati mikrokontroleri nisu imali ovu mogućnost, već su računali potrebne vrijednosti kada su bile potrebne i vrijednosti spremali u memoriju, registar opće namjene ili stog.

2. BTFSS LATB, RB3 BSF LATB, RB3 BCF LATB, RB4

Prva linija provjerava je li bit 3 registra opće namjene LATB postavljen u 1. Ako je, preskače se iduća linija. Ako nije, izvodi se druga linija koja postavlja bit 3 tog registra u 1. Treća linija se izvodi u svakom slučaju i postavlja bit 4 registra LATB u 0.

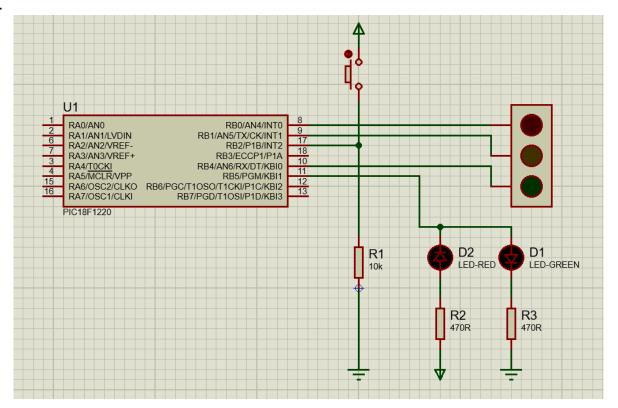
3.



Slika 1. Shema tunela u Proteusu

4. Sintaksne pogreške: U liniji 19 piše ime naredbe clfr umjesto clrf. U liniji 26 piše ime registra LATAB umjesto LATB. U liniji 30 piše ime labele wati umjesto wait.

Funkcionalna pogreška: Na semaforu B nikad nije svjetlilo zeleno svijetlo. U stanju 3 (linija 40) je netočno bila napisana konstanta koja se vraća te je palila crveno/narančasto i narančasto, a ne crveno i zeleno. Kada se vrijednost promijenila iz retlw 0x12 u retlw 0x14 semafor je počeo normalno funkcionirati.



Slika 2. Shema dva semafora u Proteusu

Na početku programa se potrebne postavke postavljaju upisom 0x00 ili 0xFF na potrebna mjesta - isključena A/D pretvorba, digitalni ulazi, raspored ulaza/izlaza... Stvara se state, na čiju adresu se upisuje broj stanja semafora,. Početno stanje je 0, te u njemu gori crveno svijetlo na oba semafora. Svijetla se postavljaju uz pomoć lookup tablice. Nakon toga se omogućuju prekidi i glavni program se vrti u beskonačnoj petlji. Na prekid (stisnut gumb) se mijenja stanje semafora. Uz tu promjenu se mora obaviti provjera je li vrijednost stanja između 0 i 5. Postoji 6 stanja nakon kojih slijedi nulto i kreću od početka. Novo stanje se sprema u state i prikladna svijetla se aktiviraju na semaforima. Opet se dozvoljava prekid i čeka se novi.

## KOD:

LIST p=18f1220; cilini procesor

CONFIG PWRT = ON, WDT = OFF, MCLRE = OFF

;konfiguracija mikrokontrolera

;WDT = OFF = iskljuci watchdog timer

;PWRT = ON = ukljuci Power Up Timer

:MCLRE = OFF = interno spoji ~MCLR na Vcc

#include "P18f1220.INC"; datoteka s definicijama konstanti za nas PIC

Ram EQU h'0'; rezervacija okteta za varijable

state EQU Ram+0; registri opce namjene pocinju od adrese 0h

org 0; obavezno

goto start ; postavljanje pocetnih vrijednosti, normalan rad

org 8; prekid

btfss INTCON3, 1; ako prekid nije od INT2 vrati u beskonacnu petlju

goto kraj

incf state,W; povecaj stanje, rezultat stavi u W

andlw 0x07; uzmi samo potrebne bitove

movwf state; spremi stanje natrag u memoriju

btfss state, 2; ako je stanje manje od 4 mijenjaj semafor

goto svjetla

btfss state, 1; ako je stanje manje od 6 mijenjaj semafor

goto svietla

movlw d'0'; ako je stanje 6 ili 7 stavi ga u 0 movwf state; spremi stanje natrag u memoriju

svjetla call getmask; izracunaj bit-masku za trenutno stanje

movwf LATB; zapisi masku na porte

bcf INTCON3, 1; ocisti prekidnu zastavicu retfie; povratak iz prekida, vraćanje konteksta

movlw 0xFF; iskljuci A/D start

movwf ADCON1; 0xff, svi ulazi su digitalni

movlw d'0'; resetiraj registar W

clrf LATB; pocisti port B clrf TRISB; sve linije izlazne clrf ADCON0; iskljuci A/D clrf state: obrisi stanje

call getmask; prvo stanje crveno potrebno 1 pozvat, nakon toga pomocu prekida

movwf LATB; zapisi masku na port

movlw 0x10

movwf INTCON3 ; dozvoli INT2 prekide

movlw 0x80

movwf INTCON ; dozvoli globalne prekide

loop goto loop; vrti beskonacno

movf state, W; dohvati stanje u registar W. getmask

addwf state, 0 ; pomnozi s 2 zbog sirine instrukcije(16 bita)

addwf PCL, 1; dodaj W registru PCL, rezultat u PCL.

retlw 0x01; 000 0001; state==0 upaljeno crveno autima i pjesacima.

retlw 0x21; 010 0001; state==1 upaljeno crveno autima, zeleno pjesacima

retlw 0x01; 000 0001; state==2 upaljeno crveno autima i pjesacima.

retlw 0x03; 000 0011; state==3 upaljeno crveno/narancasto autima, crveno

pjesacima.

kraj

retlw 0x10; 001 0000; state==4 upaljeno zeleno autima, crveno pjesacima. retlw 0x02: 000 0010: state==5 upalieno narancasto autima, crveno piesacima.

**END** 

6. 6.1 
$$V = \frac{N_V * (V_{MAX} - V_{MIN})}{2^B} + V_{MIN} = \frac{N_V * (250 - 20)}{2^{10}} + 20 = 0.2246 * N_V + 20$$

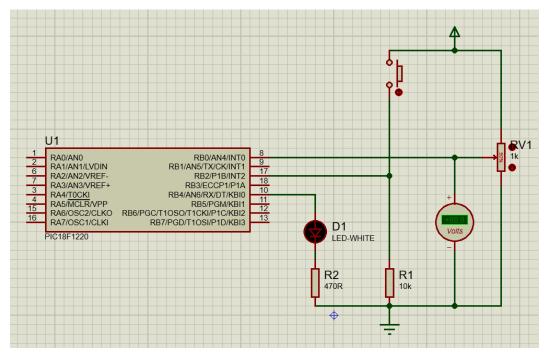
N<sub>V</sub> = izlazna binarna vrijednost A/D pretvornika

B = razlučivost A/D pretvornika

V<sub>MIN</sub>, V<sub>MAX</sub> = najniža i najviša mjeriva vrijednost A/D pretvornika

6.2 
$$V_0 = \frac{V_{MAX} - V_{MIN}}{2^B} = \frac{230}{2^{10}} = 0.2246 \text{ km/h}$$

6.2 
$$V_0 = \frac{V_{MAX} - V_{MIN}}{2^B} = \frac{230}{2^{10}} = 0.2246 \text{ km/h}$$
  
6.3  $V = 60 \frac{km}{h} \rightarrow N_V = \frac{V - V_{MIN}}{\frac{T_{MAX} - T_{MIN}}{2^B}} = \frac{V - V_{MIN}}{V_0} = \frac{60 - 20}{0.2246} = 178.09 \rightarrow N_V = 178$ 



Slika 3. Shema osjetila brzine u Proteusu

Program počinje definiranjem i aktivacijom ulaza (mjerni pretvornik i tipkalo) i izlaza (dioda – alarm). Omogućuju se A/D pretvorba i prekidi koji se javljaju pri stiskanju tipkala. Glavni program ulazi u beskonačnu petlju. A/D pretvorba i provjeravanje brzine događa se na pritisak tipkala i odlazak u prekidnu rutinu. U prekidnoj rutini prvo se vrši A/D pretvorba. Slijedi ispitivanje očitane brzine i usporedba s gore izračunatom vrijednosti 178<sub>10</sub> koja predstavlja 60 km/h. Ako je brzina veća, pali se alarm, te se u memorijsku lokaciju 10<sub>16</sub> upisuju jedinice. Ako je brzina manja ili jednaka 60 km/h, alarm ne svijetli te se u lokaciju 10<sub>16</sub> upisuju nule. Omogućavaju se budući prekidi i izlazi se iz prekidne rutine.

```
LIST p=18f1220; ciljni procesor
CONFIG PWRT = ON, WDT = OFF, MCLRE = OFF;
konfiguracija mikrokontrolera
;WDT = OFF = iskljuci watchdog timer
```

;PWRT = ON = ukljuci Power Up Timer

;MCLRE = OFF = interno spoji ~MCLR na Vcc

#include "P18f1220.INC"; datoteka s definicijama konstanti za nas

PIC

Ram EQU h'0'; rezervacija okteta za varijable

brzina EQU 0x10 ; lokacija za upis oznake brzine (0 / FF)

org 0; obavezno

goto start ; postavljanje pocetnih vrijednosti, normalan rad

org 8; prekid

btfss INTCON3, 1; ako prekid nije od INT2 vrati u beskonacnu petlju goto kraj

bsf ADCON0, GO; pokreni novu pretvorbu

pretvori btfsc ADCON0,GO; je li pretvorba gotova

goto pretvori ; vrti dok traje pretvorba

btfsc ADRESH, 1; je li najvisi bit 1? (brzina > 512?)

goto brza ; skoci na obradu prevelike brzine

btfsc ADRESH, 0; je li drugi najvisi bit 1? (brzina > 256?)

goto brza; skoci na obradu prevelike brzine

movlw d'178'; 178 (izracunata oznaka za 60km/h) u W

subwf ADRESL, 0; usporedi ocitanu brzinu sa 178 (60kmh)

btfss STATUS, C; je li brzina < 60kmh

goto spora; (je) skoci na obradu male brzine

btfsc STATUS, Z; je li brzina = 60kmh

goto spora ; (je) skoci na obradu male brzine

brza movlw b'00010000'; maska – paljenje LED

movwf LATB; aktiviraj LED

movlw 0x10 ; lokacija za upis podatka o brzini je 0x10

movwf BSR; odaberi 0x10 kao lokaciju

movlw 0xFF; sve jedinice

movwf brzina; upisi jedinice na lokaciju

goto gotovo; zavrsi prekid

spora movlw b'00000000'; maska - ne gori LED

movwf LATB; deaktiviraj LED

movlw 0x10; lokacija za upis podatka o brzini je 0x10

movwf BSR; odaberi 0x10 kao lokaciju

movlw 0x00; sve nule

movwf brzina; upisi nule na lokaciju

gotovo bcf INTCON3, 1; ocisti prekidnu zastavicu

;bsf ADCON0, GO; pokreni novu pretvorbu

kraj retfie ; povratak iz prekida, vraćanje konteksta

start clrf LATB; pocisti port B

movlw b'00000101'; maska – ulazi su RB0/AN4 (A/D) i RB2/INT2

movwf TRISB; postavi registar smjera GPIO porta B movlw b'00010001'; AD uključen na kanalu AN4 movwf ADCON0; analog input AN4, Fosc/2 za A/D

movlw b'011011111'; AN4 je A/D, ostali GPIO

movwf ADCON1

movlw b'10001101'; Fosc/16, 8 TAD, desno poravnano

movwf ADCON2

bsf PIE1, ADIE; omogući generiranje prekida od A/D jedinice

;bsf ADCON0, GO; pokretanje pretvorbe

movlw 0x10

	movwf INTCON3	; dozvoli INT2 prekide	
	movlw 0x80		
	movwf INTCON	; dozvoli globalne prekide	
loop a	roto loop i viiti bookonoo		
100р д	goto loop ; vrti beskonacno		
	END		
	LIND		