

Test pitanja za vežbu u vezi bloka tema 6

1. Označi šta je sve ključno kod definisanja funkcije za obradu prekida:

- vektor prekida
- povratak iz prekidne funkcije
- čuvanje sadržaja ključnih registara
- prenos parametara po vrednosti ili referenci

2. Početna adresa za obradu prekida zavisi od:

- toga da li je adresa vektora prekida u RAM ili ROM memoriji
- toga kako je podesi programer
- direktne ili odložene obrade
- izvora prekida

3. Pri obradi prekida označi tačne iskaze:

- obrada prekida treba da traje što kraće
- obrada prekida ne može biti prekinuta
- ulazak u obradu prekidne rutine je uvek moguć
- dugačke obrade prekida se rešavaju podelom izvršavanja prekida na kraće delove
- iz funkcije za obradu prekida je moguće pozivati običnu funkciju

4. Označi netačne iskaze:

- direktna obrada prekida je poželjna samo za kratke obrade prekida
- odložena obrada prekida se ne sme koristiti za dugačke obrade prekida
- odložena obrada prekida se realizuje pozivom funkcije iz prekidne rutine
- pri direktnoj obradi prekida je neophodno koristiti isključivo lokalne promenljive

5. Koji problem se rešava sledećom prekidnom rutinom tajmera:

```
void isr_timer(void)
{
    if (--tmr_val)
    {
        tmr_val = TIME1;
        // izvršavanje
        // akcije
    }
}
```

- interval hardverskog tajmera je prekratak
- potreba za odloženim izvršavanjem
- interval hardverskog tajmera je predugačak
- direktna obrada bi bila predugačka

6. Kod funkcija za obradu prekida prenos parametara se vrši:

- po referenci
- lokalnim promenljivama
- koriste se globalne promenljive
- ne prenose se parametri

7. Izbaci uljeza:

```
void isr_func(void) interrupt 5 using 2
void isr_func(void) __interrupt (5)
void interrupt isr_func(void) @ 0x10
__interrupt void isr_func(void)
ISR(PCINT_vect)
void reentrant interrupt(void)
```

8. Zaokružiti tačne iskaze vezane za prekidne funkcije.

- Dva osnovna elementa u definisanju prekidne funkcije su početna adresa (vektor prekida) i mašinska instrukcija za povratak iz prekida.
- Svaka prekidna funkcija (ili prekidni potprogram u assembleru) mora sačuvati sadržaj ključnih registara.
- Za mikrokontrolere kod kojih imamo samo jednu prekidnu funkciju neophodno je definisati i adresu te funkcije.
- Ako se iz prekidne funkcije zove neka funkcija koja se može pozvati i iz osnovnog programa, takva funkcija mora biti definisana kao rembrant. :-)
- Uobičajeno je pravilo da se prekidna funkcija izvršava što brže.
- Prekidne funkcije ne mogu da koriste globalne promenljive koje su pristupačne i iz ostalih programskih blokova (zajedničke promenljive).

9. Označi netačne iskaze:

- Indirektno adresiranje prekidne funkcije se koristi kod manjih (slabijih) mikrokontrolera, a podrazumeva da je prekidna funkcija smeštena na fiksnu adresu.
- Rad sa prekidima na način odložene obrade prekida omogućava pravovremeno reagovanje na više različitih i čestih prekida, bez opasnosti da će dugotrajna obrada pojedinačnog prekida u kraćem ili dužem vremenskom intervalu blokirati ostale programske blokove.
- Najčešće korišćeni prekidi su prekidi tajmera.
- Ako se u prekidnoj funkciji samo registruje razlog prekida, uz eventualnu brzu delimičnu obradu, a osnovnom programu se prepusti preostali deo obrade prekida koji nije vremenski kritičan, reč je o direktnoj obradi prekida.

10. U datom primeru, prekidna rutina tajmera podešena je na 1 ms. Unutar prekidne funkcije koriste se dva brojača. Ako je vrednost tmr1 = 10, kolika je dužina prvog intervala?

Koliko treba da je vrednost pomenljive tmr2, tako da drugi interval bude dužine 50 ms?

```
void isr_timer(void) // 1 ms interval
{
    if (!(--tmr1))
    {
        tmr1 = 10;
        isr_flag1 = 1;
        if (!(--tmr2))
        {
            tmr2 = __ ; //popuniti prazninu
            isr_flag2 = 1;
        }
    }
}
```

```
}
```

- A) Duzina intervala jedan je 1 ms, a tmr2 = 5.
- B) Duzina intervala jedan je 1 ms, a tmr2 = 50.
- C) Duzina intervala jedan je 10 ms, a tmr2 = 5.
- D) Duzina intervala jedan je 10 ms, a tmr2 = 50.

11. Ukoliko je frekvencija prekida tajmera 200 Hz. Koju vrednost treba da ima konstanta TIME da bi se akcija izvršila svake sekunde? Akcija nije specificirana, ali je njeno mesto u kodu označeno komentaram.

```
void isr_timer(void)
{
    if (!(--tmr_brojac))
    {
        tmr_brojac = TIME;
        // izvršavanje
        // akcije
    }
}
```

- a) 200
- b) 150
- c) 50
- d) 20

12. Komande za zabranu i dozvolu prekida kod Arduina su:

- a) EA() i EI()
- b) cli() i sei()
- c) DI() i EI()
- d) cle() i sea()