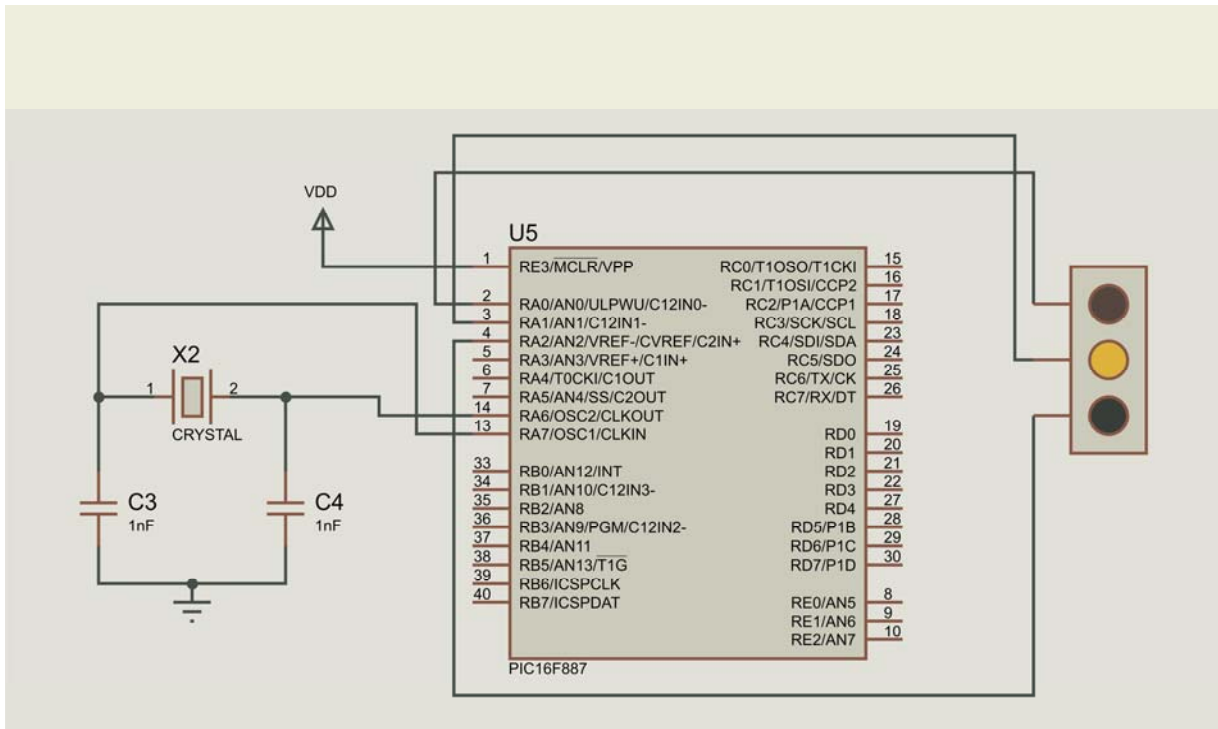


Priloženi primer je več rađen na predavanjima pre dve nedelje. Šematski prikaz je dat na slici 1.



Upravljanje semaforom preko mikrokontrolera je bilo realizovano na dva načina. Programski kod koji smo objasnili izgledao je ovako:

```
void main() {
    TRISA = 0x00;
    PORTA = 0x00;
    while(1)
    {
        PORTA = 0x01;
        Delay_ms(990);
        PORTA = 0x03;
        Delay_ms(990);
        PORTA = 0x04;
        Delay_ms(990);
        PORTA = 0x02;
        Delay_ms(990);
    }
}
```

Pričali smo o nedostacima ovakvog pristupa rešavanju problema. Zatim je isti problem rešen korišćenem Interrupt procedure. Programski kod je izgledao ovako:

```
unsigned long cnt;

void interrupt() { // Procedura se poziva svakih 10 ms
    if (TMR0IF_bit) {
        cnt++;
        TMR0IF_bit = 0;
        TMR0 = 100;
    }
}

void main() {

    OPTION_REG = 0x86; //
    TMR0 = 100; // Podešavanje odgovarajućih registara za pozivanje
Interrupt procedure svakih 10 ms
    INTCON = 0xA0; //

    ANSEL = 0;
    ANSELH = 0;

    C1ON_bit = 0;
    C2ON_bit = 0;

    TRISA = 0; // Podešavanje porta kao izlaznog
    PORTA = 0x00; // Postavljanje vrednosti porta na 0

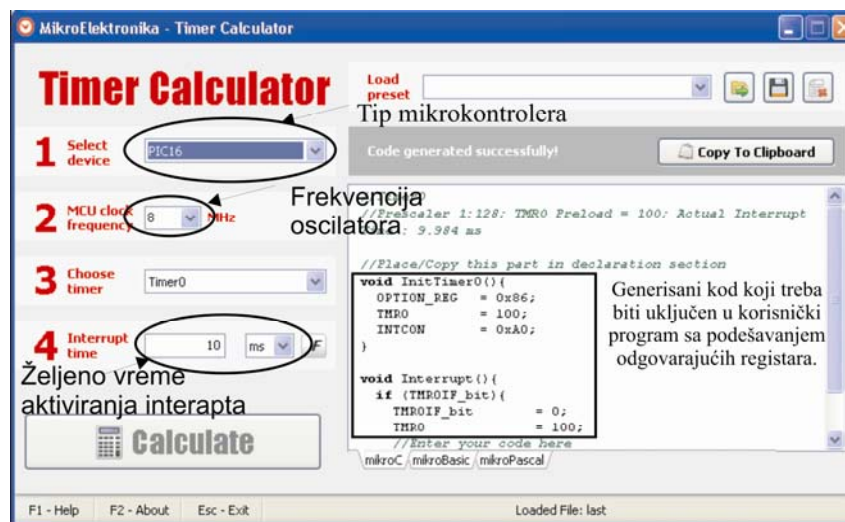
    cnt = 0; // Resetovanje brojac cija se vrednost inkrementira u
Interrupt proceduri svakih 10 ms

    do {
        if (cnt >= 100) { // Ukoliko je brojac dostigao vrednost 100, ukupno je
proteklo 100*10 ms = 1 s
            switch(PORTA) //
            {
                case 0x00: PORTA = 0x01; break;//
                case 0x01: PORTA = 0x03; break;//
                case 0x02: PORTA = 0x01; break;// Posle svake sekunde se menja stanje
na semaforu
                case 0x03: PORTA = 0x04; break;//
                case 0x04: PORTA = 0x02; break;//
                default: PORTA = 0; break;//
            }
            cnt = 0; // Reset cnt
        }
    } while(1);
}
```

U programskom kodu su Vam dati komentari najvažnijih blokova programa. Ukratko, program je koncipiran tako da se vrši promena stanja na semaforu svake sekunde. Interrupt procedure je podešena tako da se poziva svakih 10 ms. Kada brojač cnt dostigne vrednost 100,

ukupno proteklo vreme iznosi $100 \cdot 10 \text{ ms}$, odnosno 1 s i vrši se odgovarajuća promena stanja na semaforu.

1. Napraviti novi projekat sa šematskim prikazom sa slike 1. Napraviti novi projekat u MikroC kompajleru (<https://www.mikroe.com/mikroc-pic>). Izvršiti simulaciju i verifikovati ispravnost funkcionisanja semafora (bilo bi dobro da se verifikacija izvrši praćenjem signala na pinovima A0, A1 i A2 pomoću virtuelnog osciloskopa).
2. Izvršiti modifikaciju programskog koda tako da se odgovarajuća promena stanja na semaforu vrši na svakih 0.5 s a zatim na svake 2 s. Programski kod za ova druga dva slučaja je praktično isti kao u datom primeru. Jedina razlika je u podešavanju vrednosti odgovarajućih registara za interrupt mikrokontrolera. Da bi olakšali rešenje ovog problema koristiti gotov softver (<https://www.mikroe.com/timer-calculator>) prikazan na sledećoj slici.



3. Po uspešnom završetku zadatka datih pod tačkom 2, pogledati datasheet za PIC16F887, strana 75-77 gde je detaljno objašnjena struktura i namena odgovarajućih registara korišćenih u realizaciji ovog problema.