

UNIVERZITET SINGIDUNUM
SOFTVERSKO I INFORMACIONO INŽENJERSTVO



SISTEMI ZA RAD U REALNOM VREMENU

- Ispit -

Lazar Smiljković 2017201826

Objašnjenje:

1. „Neophodno je napraviti sistem koji se sastoji od 4 LE diode (RC4-RC7), 7-segmentnog displeja sa 4 polja i 2 tastera (RB0 – RB1). Pinove RB4, RB5, RB6 i RB7 treba definisati kao ulazne i za njih treba koristiti IOC(Interrupt-On-Change), kada se na sva 4 pina detektuje visok naponski nivo tada je sistem u normalnom modu rada. U slučaju da je detektovan nizak naponski nivo na nekom od četiri pina(RB4-RB7) potrebno je da sistem predje u mod smanjenih kapaciteta. U slučaju da je u isto vreme na sva 4 pina detektovan nizak naponski nivo, potrebno je odmah ugasiti sistem.“

-Za početak na PIC18F45K22 Mikrokontroler povezao sam 4 diode na portovima RC4-RC7, nakon toga ubacio sam 7-segmentni displej sa 4 polja i 2 tastera povezan na odgovarajuće portove, na portove RB4-RB7 realizovan je mehanizam prekida (Interrupt On Change), dodavanje tastera i povezivanje istih na napon i uzemljenje, te portove sam na početku stavio na definisan HIGH tj. 1 preko mikroC programa jer se traži visok napon na njima i to ceo sistem stavlja u normalni mod rada (**STANDARDNI_MOD**) koji je definisan kao jedna od dve opcije **stanja** i vraća ga u isti ako se svi ti portovi vrate na visok napon ponovo pomoću tastera koji menjaju napon koji dolazi na te portove.

-Ako bilo koji od tih portova vratimo na nizak napon, pomoću if naredbe koja proverava da li je bilo koji od njih promenio naponski nivo, sistem će se prebaciti u **stanje SMANJENI_KAPACITET**, a ako su sva 4 IOC porta postavljena na nizak napon sistem će prebaciti brojač na displeju na 0 i sistem će se ugasiti(svi portovi će biti ugašeni preko Stop() metode).

2. „**Standardni mod** – U ovom modu sve diode treba da svetle. A na sedmosegmentnom displeju treba da odbrojava sekunde koliko se nalazi u ovom modu.“

-U ovom modu kako bi svetlele sve diode svi njihovi portovi su preko mikroC programa postavljeni na visok napon (dok neka If naredba to ne promeni kada zatreba).

-Za displej je dodat msBroj koji služi za prikazivanje i odbrojavanje cifri (digita) na displeju, takođe je dodata inicijalizacija timera i ubačen je svaki karakter za displej.

3. „**Smanjeni kapacitet** – U ovom modu na svakom pinu porta B na kome je detektovan nizak naponski nivo odgovarajuća dioda na portu C treba da treperi (RB4 -> RC4, RB5 -> RC5, RB6 -> RC6, RB7 -> RC7). U ovom modu postaju aktivna 2 tastera povezana na RB0 i RB1, pomocu ova dva tastera povećavamo i smanjujemo frekvenciju treperenja dioda. Na sedmosegmentom ekranu treba da krene odbrojavanje od 10 – 0 sekundi do gašenja sistema. Ukoliko se pre isteka vremena ponovo detektuje visok naponski nivo na svim pinovima ponovo se vraćamo u standardni mod.“

-Mod smanjenog kapaciteta pokreće se stavljanjem nekog od RB4 – RB7 portova na nizak napon, on spaja portove IOC sa portovima dioda tako što pri menjanju napona na portovima RB menjamo kako portovi RC reaguju na njih. Ako je odgovarajući port RB na visokom naponu, dioda na RC portu koji odgovara tom RB portu će svetleti, u slučaju da se port RB stavi na nizak napon odgovarajuća dioda će da treperi uz pomoć If naredbe i uključivanja treperenja sa naredbom:

LATC#_bit = ~LATC#_bit;

-Kada smo to završili dodajemo dva tastera na portove RB0 i RB1 koji će služiti za povećavanje ili smanjivanje frekvencije treperenja, to smo postigli ubacivanjem **treperenje** koja služi za ubrzavanje ili smanjivanje treperenja dioda u zavisnosti od komande koju želimo tako što taj broj povećamo ili smanjimo.

-Na displeju se **broj** postavlja na 11000ms što je 11 sekundi i to je urađeno kako bi prva cifra prikazivanja na displeju bila 10. Kada **broj** dostigne 0 aktivira se metoda Stop() koja gasi sve portove na mikrokontroleru.

-Ukoliko portove IOC tj. RB4 – RB7 stavimo na visok napon pre isteka vremena program će preko If naredbe proveriti to i vratiće se u rad Standardnog moda nastavljajući da broji vreme rada.

Opis koda:

```
#define HIGH 0x1
#define LOW 0x0
#define STANDARDNI_MOD 0
#define SMANJENI_KAPACITET 1
#define LIMITUP 250
#define LIMITDOWN 1500
```

-Ovaj deo koda koristi se za definisanje:

HIGH - 1 (visok napon)

LOW – 0 (nizak napon)

STANDARDNI_MOD 0 – Standardni mod može preko **stanje** opcije da se koristi kao 0

SMANJENI_KAPACITET 1 – Smanjeni kapacitet mod može da se koristi i kao 1

LIMITUP 250– Minimalna moguća frekvencija treperenja diode

LIMITDOWN 1500– Maksimalna moguća frekvencija treperenja diode

```
const unsigned char SEG_MAP[] = {
    0x01, //0
    0x4F, //1
    0x12, //2
    0x06, //3
    0x4c, //4
    0x24, //5
    0x20, //6
    0x0F, //7
    0x00, //8
    0x04 //9
};
```

-Mapiranje brojeva za 7-segmentni displej od 0 – 9

```
unsigned char
display[4]={SEG_MAP[0],SEG_MAP[0],SEG_MAP[0],SEG_MAP[0]};
unsigned char indexDisplay = 0;
unsigned short position;
int digit;
```

-Inicijalizovanje displeja, njegovog indeksa displeja koji može biti od 0-3, pozicije koju menjamo i digita tj. cifre koja se prikazuje

```
unsigned int broj= 0;
unsigned int treperenje= 200;
bit stanje;
```

-Postavljanje **broj** – broj prikazan na displeju koji može da se dodaje ili oduzima u zavisnosti šta nam je potrebno, **treperenje**– broj u ms koji se koristi za menjanje frekvencije treperenja dioda, stanje – može biti 1 ili 0 u zavisnosti da li se sistem stavlja na STANDARDNI_MOD tj. 0 ili SMANJENI_KAPACITET tj.1.

```
void InitTimer0(){
    T0CON = 0x88;
    TMR0H = 0xB1;
    TMR0L = 0xE0;
    GIE_bit = 1;
    TMR0IE_bit = 1;
}
```

-Kod koji pravi timer interruptove . Kod se generiše u preko Timer Calculatora gde se odabere koliku frekvenciju i interrupt time želimo i u zavisnosti od toga on nam generiše kod

```
void Stop(){
    GIE_bit = 0;
    TRISA = 0;
    TRISD = 0;
    PORTA = 0;
    PORTB = 0;
    PORTC = 0;
    PORTD = 0;
}
```

-Metoda Stop koja se koristi za gasenje svih dioda tj. Postavljanje napona na svim diodama mikrokontrolera na 0 kako bi se postigao efekat gašenja programa.

```

void Interrupt(){
if(TMROIF_bit){
    TMROIF_bit = 0;
    TMROH = 0xB1;
    TMROL = 0xE0;
    LATA= 0x00;
    LATD = display[indexDisplay];
    LATA = position;
    position <= 1;
    if(position > 8){
        position = 0x01;
    }
    indexDisplay++;
    if(indexDisplay > 3){
        indexDisplay = 0;
    }
}
}

```

-Prekid u kome se gase aktivni portovi, tj. da nam nije aktivan ni jedan port sa desne strane pa onda na poziciji D stavimo vrednost i aktiviramo sledeću aktivnu poziciju (u jednom trenutku je samo jedna aktivna) i ona se pomera za jednu poziciju i kada stigne do 16 vraća se na prvobitnu poziciju i ide u krug. Index takođe isto radi sem što ide od 0-3.

```

if(RBIF_bit) {
    RBIF_bit = 0;
    if(PORTB.F4 == 0 || PORTB.F5 == 0 || PORTB.F6 == 0 || PORTB.F7 == 0){
        stanje = SMANJENI_KAPACITET;
        broj = 11000;
    }
    if(PORTB.F4 == 1 && PORTB.F5 == 1 && PORTB.F6 == 1 && PORTB.F7 == 1){
        stanje = STANDARDNI_MOD;
        LATC = 0xF0;
        broj = 0;
    }
}
}
}

```

-Ovaj deo koda proverava da li je bilo koji od portova na niskom naponu, ako jeste postavlja se timer na 10 i stanje se menja u SMANJENI_KAPACITET.

-U slučaju da su svi portovi vraćeni na visok napon stanje se menja u STANDARDNI_MOD, uključuju se sve diode i kreće odbrojavanje od 0.

```

void Start(){
    TRISB = 0xF0;
    ANSELB = 0x00;
    ANSELD = 0;
    TRISA = 0;
    TRISD = 0;
    TRISC = 0x0F;
    LATC = 0xF0;
    position = 0x01;
    ADC_Init();
    InitTimer0();
    RBIE_bit = HIGH;
    RBIF_bit = LOW;
    GIE_bit = HIGH;
    IOCB4_bit = HIGH;
    IOCB5_bit = HIGH;
    IOCB6_bit = HIGH;
    IOCB7_bit = HIGH;
    stanje = STANDARDNI_MOD;
}

```

-Metoda start se pokreće čim se pokrene program, ona služi za uključivanje pojedinih portova i takođe gašenje nekih, pokretanje timera i stavljanje celog sistema na STANDARDNI_MOD pri pokretanju programa.

```

if( Button(&PORTB, 0, 1, 1)) {
    if(treperenje> LIMITDOWN){
        treperenje= LIMITDOWN;
    }else{
        treperenje+=100;
    }
}
if( Button(&PORTB, 1, 1, 1)) {
    if(treperenje> LIMITUP){
        treperenje= LIMITUP;
    }else{
        treperenje-=100;
    }
}
}

```

-U ovom delu koda prikazuje se logika iza 2 tastera koji služe za menjanje frekvencije treperenja dioda(RB0 i RB1), pomoću na početku definisanih LIMITDOWN i LIMITUP dobijamo limit do kog dioda može da ide sa treperenjem, zatim menjamo treperenje u zavisnosti od toga da li želimo da smanjimo ili povećamo frekvenciju (+ ili -)


```

if(stanje == SMANJENI_KAPACITET){
    broj--;
    if(PORTB.F4 == 0 && PORTB.F5 == 0 && PORTB.F6 == 0 && PORTB.F7 == 0){
        GIE_bit = 0;
        Stop();
    }

    if(PORTB.F4 == 1 && PORTB.F5 == 1 && PORTB.F6 == 1 && PORTB.F7 == 1){

        TRISB = 0xF0;
        ANSELB = 0x00;
        ANSELD = 0;
        TRISA = 0;
        TRISD = 0;
        TRISC = 0x0F;
        LATC = 0xF0;
        RBIE_bit = HIGH;
        RBIF_bit = LOW;
        GIE_bit = HIGH;
        IOCB4_bit = HIGH;
        IOCB5_bit = HIGH;
        IOCB6_bit = HIGH;
        IOCB7_bit = HIGH;
        stanje = STANDARDNI_MOD;
        broj = 0;
    }

    if(PORTB.F4 == 0){
        if(broj % treperenje == 0){
            LATC4_bit = ~LATC4_bit;
            if(broj == 0){
                Stop();
                return;
            }
        }
    }
}

if(PORTB.F5 == 0){
    if(broj % treperenje == 0){
        LATC5_bit = ~LATC5_bit;
        if(broj == 0){
            Stop();
            return;
        }
    }
}

if(PORTB.F6 == 0){
    if(broj % treperenje == 0){
        LATC6_bit = ~LATC6_bit;
        if(broj == 0){
            Stop();
        }
    }
}

```

```

        return;
    }
}
}
if(PORTB.F7 == 0){
    if(broj % treperenje== 0){
        LATC7_bit = ~LATC7_bit;
        if(broj ==0){
            Stop();
            return;
        }
    }
}
} else{
    broj ++;
}
}

```

-Ako je **stanje** sistema SMANJENI_KAPACITET **broj** se postavlja da oduzima umesto što dodaje brojeve na displeju i u isto vreme se aktivira prebacivanje **broj** na 10 iz prethodnog koda.

-U ovom kodu proveravamo na početku da li je svaki od IOC ulaza (RB4 – RB7) na niskom naponu, ako jeste aktivira se Stop() metoda koja gasi sve pinove na mikrokontroleru i tako se dobija gasenje sistema. Ako to nije slučaj dalje se proverava da li je neki od tih portova pojedinačno ugašen (ili više njih odjednom) i kada se nađe taj određeni port određena dioda se stavlja na treperenje u zavisnosti od B porta koji je na niskom naponu, postavlja se If naredba koja proverava da li je **broj** (broj na displeju) dostigao nulu, i ako jeste poziva se Stop() metoda koja gasi ceo sistem.

-U drugom slučaju proveravamo da li je visok napon vraćen na sva 4 porta, u slučaju da jeste sistem se vraća u STANDARDNI_MOD, portovi se uključuju i isključuju i displej ponovo kreće da broji vreme rada sistema.

-U slučaju da ni jedno od ovih nije slučaj, nista od ponuđenog se ne dešava već se samo **broj** vrati na prvobitan Standardni mod i računa vreme provedeno u njemu.