

KATEDRA ZA ELEKTRONIKU

Laboratorijske vežbe

OSNOVI DIGITALNE ELEKTRONIKE (13S042ODE)

Vežba br. 2

MIKROKONTROLER PIC18F45K22 - TAJMERI, PREKIDI, KONTROLA ČETVOROCIFRENOG 7-SEGMENTNOG DISPLEJA

Pribor:

1. Laptop	1 kom.
2. Razvojni sistem EasyPic v7 sa PIC18F45K22 mikrokontrolerom	1 kom.
3. USB kabl	1 kom.

ZADATAK 1

U razvojnom okruženju "MikroC PRO for PIC" kreirati novi projekat pod imenom "Vezba2_1", selektovati "Device Name" P18F45K22, i projekat snimiti u odgovarajući folder po želji. U projekat dodati novi ".c" fajl pod imenom "Vezba2_1", i uključiti u projekat sve raspoložive biblioteke - opcija "Include All (Default)".

U fajl "Vezba2_1.c" uneti sledeći source kod:

```
unsigned short i;  
  
//--- Returns mask for common cathode 7-seg. display  
unsigned short mask(unsigned short num) {  
    switch (num) {  
        case 0 : return 0x3F;  
        case 1 : return 0x06;  
        case 2 : return 0x5B;  
        case 3 : return 0x4F;  
        case 4 : return 0x66;  
        case 5 : return 0x6D;  
        case 6 : return 0x7D;  
        case 7 : return 0x07;  
    }
```

```

        case 8 : return 0x7F;
        case 9 : return 0x6F;
    }
}

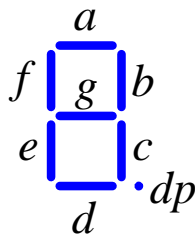
void main() {
    ANSELA = 0;           // Configure PORTA pins as digital
    ANSELB = 0;           // Configure PORTB pins as digital
    ANSELD = 0;           // Configure PORTD pins as digital
    TRISA = 0;            // Configure PORTA as output
    LATA = 0;             // Clear PORTA
    TRISD = 0;            // Configure PORTD as output
    LATD = 0;             // Clear PORTD

    do {
        for (i=0; i<8; i++) {
            LATA = 0;      // Turn off all 7seg displays
            LATD = mask(i); // bring appropriate value to PORTD
            LATA = 1;      // turn on appropriate 7seg. display
            Delay_ms(1000); // one second delay
        }
    } while(1);
}

```

- Povezati razvojni sistem EasyPic v7 sa laptopom preko USB kabla (preko USB konektora CN2).
- Povezati pinove RA0, RA1, RA2 i RA3 mikrokontrolera na selekzione linije cifara četvorocifrenog sedmosegmentnog displeja uključivanjem prekidača SW4.1, SW4.2, SW4.3 i SW4.4.
- Uključiti napajanje na razvojnom sistemu, odraditi build unetog source koda i izvršiti programiranje mikrokontrolera.
- Proanalizirati uneti source kod i ponašanje razvojnog sistema.
- Objasniti ulogu funkcije `unsigned short mask(unsigned short num)`.
- Šta se postiže upisivanjem vrednosti 1 u registar specijalne namene LATA, a šta upisivanjem vrednosti 0?
- Koju vrednost je potrebno upisati u registar specijalne namene LATA, a koju u registar specijalne namene LATD, da bi se uključila decimalna tačka na sedmosegmentnom displeju koji predstavlja cifru DIS2?

Detaljan opis značenja pojedinih bita registara specijalne namene koji kontrolišu portove mikrokontrolera se može videti u datasheet-u mikrokontrolera "40001412G.pdf" u sekciji 10 ("I/O Ports"). Opis četvorocifrenog sedmosegmentnog displeja se može naći u "easypic-v7-manual-v104d.pdf" na strani 27. Oznake za pojedine segmente svake od cifara sedmosegmentnog displeja su prikazane na slici 1. Kompletna električna šema razvojnog sistema se može naći u dokumentu " easypic-v7-schematic-v104c.pdf".



Slika 1

ZADATAK 2

Napisati program u programskom jeziku C za mikrokontroler PIC18F45K22 koji će obezbediti da se najpre na sedmosegmentnom displeju koji predstavlja cifru DIS3 prikaže karakter '1', zatim nakon 500ms da se na sedmosegmentnom displeju koji predstavlja cifru DIS2 prikaže karakter '2', zatim nakon 500ms da se na sedmosegmentnom displeju koji predstavlja cifru DIS1 prikaže karakter '3' i da se potom nakon 500ms na sedmosegmentnom displeju koji predstavlja cifru DIS0 prikaže karakter '4'. Nakon pauze od 2s tokom koje treba da budu isključena sva 4 sedmosegmentna displeja, ponoviti opisani ispis karaktera na 4 sedmosegmentna displeja, ali ovog puta sa karakterima '9', '8', '7' i '6'.

Opisani scenario treba da se ciklično ponavlja sa pauzama od 2s (tokom kojih treba da budu isključena sva 4 sedmosegmentna displeja).

Isprogramirati mikrokontroler, testirati i debugovati (po potrebi) napisani program.

Po završetku zadatka, pozvati dežurnog nastavnika i demonstrirati mu rad programa na razvojnom sistemu.

ZADATAK 3

U razvojnom okruženju "MikroC PRO for PIC" kreirati novi projekat pod imenom "Vezba2_3", selektovati "Device Name" P18F45K22, i projekat snimiti u odgovarajući folder po želji. U projekat dodati novi ".c" fajl pod imenom "Vezba2_3", i uključiti u projekat sve raspoložive biblioteke - opcija "Include All (Default)".

U fajl "Vezba2_3.c" uneti sledeći source kod:

```
unsigned short digit, changeDigitFlag, tick;

//--- Returns mask for common cathode 7-seg. display
unsigned short mask(unsigned short num) {
    switch (num) {
```

```

    case 0 : return 0x3F;
    case 1 : return 0x06;
    case 2 : return 0x5B;
    case 3 : return 0x4F;
    case 4 : return 0x66;
    case 5 : return 0x6D;
    case 6 : return 0x7D;
    case 7 : return 0x07;
    case 8 : return 0x7F;
    case 9 : return 0x6F;
}
}

// This interrupt occurs every 8.192 ms (256*64/2000000=0.008192)
void interrupt() {
    LATA = 0;           // Turn off all 7seg displays
    LATD = mask(digit); // bring appropriate value to PORTD
    LATA = 2;          // turn on appropriate 7seg. display

    if (tick < 100) {
        tick++;
    }
    else {
        tick = 0;
        changeDigitFlag = 1;
        // This flag is set every 0.8192 sec (256*64*100/2000000=0.8192)
    }

    TMR0L = 0;           // Clear TMR0L
    TMR0IF_bit = 0;      // Clear TMR0IF
}

void main() {
    ANSELA = 0;          // Configure PORTA pins as digital
    ANSELD = 0;          // Configure PORTD pins as digital
    TRISA = 0;           // Configure PORTA as output
    LATA = 0;            // Clear PORTA
    TRISD = 0;           // Configure PORTD as output
    LATD = 0;            // Clear PORTD

    TOCON = 0xC5;        // Set TMR0 in 8bit mode, assign prescaler to TMR0
    TMR0L = 0;           // Clear TMR0L
    digit = 0;
    changeDigitFlag = 0;
    tick = 0;

    GIE_bit = 1;         // Enable all unmasked interrupts
    TMR0IE_bit = 1;      // Enable TMR0 overflow interrupt
}

```

```

do {
    if (changeDigitFlag) {
        if (digit == 9)
            digit = 0;
        else
            digit++;
        changeDigitFlag = 0;
    }
} while(1);
}

```

- Povezati pinove RA0, RA1, RA2 i RA3 mikrokontrolera na selekzione linije cifara četvorocifrenog sedmosegmentnog displeja uključivanjem prekidača SW4.1, SW4.2, SW4.3 i SW4.4.
- Uključiti napajanje na razvojnom sistemu, odraditi build unetog source koda i izvršiti programiranje mikrokontrolera.
- Proanalizirati uneti source kod i ponašanje razvojnog sistema.
- Objasniti način izračunavanja periode pojavljivanja prekida.
- Objasniti način izračunavanja periode setovanja flega changeDigitFlag. Koja je uloga tog flega?
- Objasniti na koji način radi "Timer0" za konfiguraciju bita u registrima specijalne namene navedenu u source kodu.

Detaljan opis značenja pojedinih bita registara specijalne namene koji kontrolišu portove mikrokontrolera se može videti u datasheet-u mikrokontrolera "40001412G.pdf" u sekciji 10 ("I/O Ports"). Detaljan opis modula "Timer0" mikrokontrolera se može videti u datasheet-u mikrokontrolera "40001412G.pdf" u sekciji 11 ("Timer0 Module"). Detaljan opis prekida kod mikrokontrolera se može videti u datasheet-u mikrokontrolera "40001412G.pdf" u sekciji 9 ("Interrupts"). Opis četvorocifrenog sedmosegmentnog displeja se može naći u "easypic-v7-manual-v104d.pdf" na strani 27. Oznake za pojedine segmente svake od cifara sedmosegmentnog displeja su prikazane na slici 1. Kompletna električna šema razvojnog sistema se može naći u dokumentu " easypic-v7-schematic-v104c.pdf".

ZADATAK 4

Napisati program u programskom jeziku C za mikrokontroler PIC18F45K22 koji će omogućiti da se na četvorocifrenom sedmosegmentnom displeju prikazuju četvorocifreni brojevi koji su elementi unapred definisanog niza u programskoj memoriji. Niz se sastoji od 5 elemenata, čiji prikaz treba da se ciklično smenjuje na četvorocifrenom sedmosegmentnom displeju svakih 5s.

Konfigurisati "Timer0" kao 16-bitni tajmer. Multipleksiranje cifara četvorocifrenog sedmosegmentnog displeja i merenje vremenskog intervala od 5s odraditi korišćenjem prekida. Nije dozvoljeno koristiti funkciju `Delay_ms()`.

Isprogramirati mikrokontroler, testirati i debugovati (po potrebi) napisani program.

Po završetku zadatka, pozvati dežurnog nastavnika i demonstrirati mu rad programa na razvojnom sistemu.