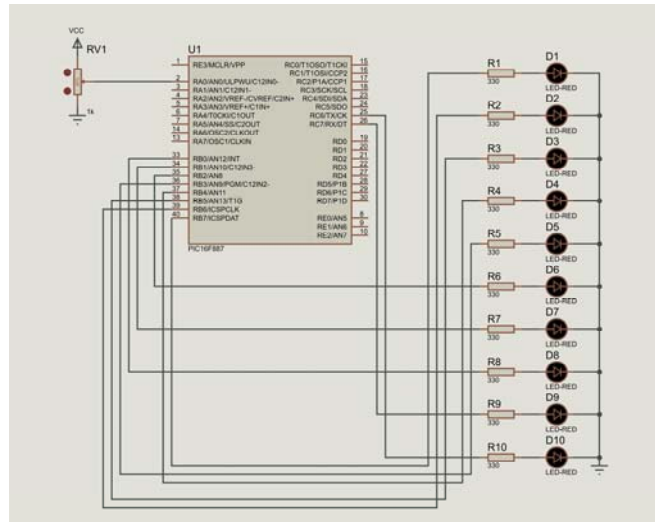


Za ovu nedelju zaplanirano je utvrđivanje A/D konverzije i povezivanje sa zadatkom iz prošle nedelje u smislu samostalne realizacije programa koji vrši A/D konverziju i kontroliše akviziciju napona pomoću prekidačke procedure.

1. U uputstvu za MikroC (mikroc_pic_pro_manual_v101.pdf) pogledati materijal dat na stranama 243 i 244. Realizovati simulacioni projekat sa strane 244. Elektronsko kolo je dato na sledećoj slici.

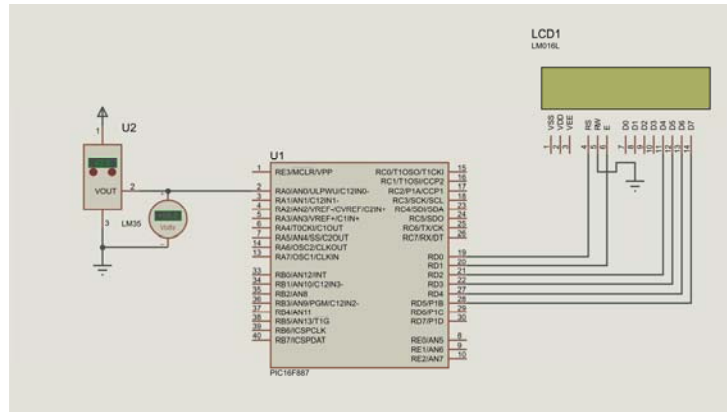


Programski kod je dat na strani 243. Napraviti simulacioni projekat i program za PIC16F887 i učitati ga u simulaciju. Program jednostavno čita analognu vrednost napona sa potenciometra, i odgovarajuću 10-bitnu vrednost predstavlja preko grupe LE dioda. Voditi računa o pinu na kojem se vrši A/D konverzija. U primeru iz .pdf fajla konverzija se vrši na A2 pinu a u šemi na slici na A0 pinu. Drugim rečima programski kod treba da izgleda:

```
unsigned int napon;
```

```
void main() {
    ANSEL = 0x01; // Konfiguracija A0 pina kao analognog
    TRISA = 0xFF; // postavljanje PORTA kao ulaznog
    ANSELH = 0;
    TRISC = 0x3F; // Postavljanje pinova RC7, RC6 kao izlaznih
    TRISB = 0; // PORTB je takođe izlazan
    do {
        napon = ADC_Read(0); // Čitanje 10-bitne vrednosti napona
        PORTB = napon; // Slanje nižih osam bitova na PORTB
        PORTC = napon >> 2; // Slanje 2 najviša bita na RC7, RC6
    } while(1);
}
```

2. Ponoviti primer koje smo radili na času. Akvizicija temperature pomoću LM35 senzora i predstavljanje temperature na LCD displej.



```

sbit LCD_RS at RD0_bit;
sbit LCD_EN at RD1_bit;
sbit LCD_D4 at RD2_bit;
sbit LCD_D5 at RD3_bit;
sbit LCD_D6 at RD4_bit;
sbit LCD_D7 at RD5_bit;
sbit LCD_RS_Direction at TRISD0_bit;
sbit LCD_EN_Direction at TRISD1_bit;
sbit LCD_D4_Direction at TRISD2_bit;
sbit LCD_D5_Direction at TRISD3_bit;
sbit LCD_D6_Direction at TRISD4_bit;
sbit LCD_D7_Direction at TRISD5_bit;

```

```

char Temperature[] = " 00.0 C";
unsigned int Temp;

```

```

void main() {
    OSCCON = 0X70;
    ANSEL = 1;
    Lcd_Init();
    Lcd_Cmd( LCD_CURSOR_OFF);
    Lcd_Cmd( LCD_CLEAR);
    lcd_out(1, 3, "Temperature:");

```

```

while(1) {

    Temp = ADC_Read(0) * 0.489;

    if (temp > 99)
        Temperature[0] = 1 + 48;
    else
        Temperature[0] = '.';
        Temperature[1] = (temp / 10) % 10 + 48;
        Temperature[2] = temp % 10 + 48;
        Temperature[5] = 223;
        lcd_out(2, 5, Temperature);
        delay_ms(1000);
    }
}

```

Napisati komentare u svakoj liniji programskog koda i objasniti šta koja komanda radi.

3. Napisati novi programski kod koji će sadržati prekidačku funkciju (interrupt procedure). Implementirana prekidačka funkcija treba da omogući osvežavanje LCD svake sekunde, za razliku od datog programskog koda koji osvežavanje LCD vrši proizvoljno.