

Univerza v Ljubljani  
Fakulteta za elektrotehniko

Jernej Rozman

# Trikanalni voltmetar

V Ljubljani, maja 2007

# KAZALO

<b>1.</b>	<b>UVOD</b> .....	<b>1</b>
<b>2.</b>	<b>OPIS VEZJA</b> .....	<b>1</b>
2.1.	VHODNI DEL VEZJA.....	1
2.2.	IZHODNI DEL VEZJA.....	1
<b>3.</b>	<b>MIKROKRMILNIK</b> .....	<b>2</b>
3.1.	IZBIRA MIKROKRMILNIKA.....	2
3.2.	OPIS MIKROKRMILNIKA.....	2
<b>4.</b>	<b>PROGRAM</b> .....	<b>3</b>
4.1.	KODA PROGRAMA.....	3
4.2.	OPIS PROGRAMA.....	8
<b>5.</b>	<b>PRILOGE</b> .....	<b>9</b>
5.1.	DIAGRAM POTEKA PROGRAMA.....	10
5.2.	ELEKTRIČNA SHEMA.....	11
5.3.	NAČRT TISKANEGA VEZJA.....	12
5.4.	MONTAŽNI NAČRT.....	13
5.5.	KOSOVNICA.....	14
5.6.	CENA MATERIALA.....	15
<b>6.</b>	<b>VIRI</b> .....	<b>16</b>

# 1. UVOD

Pri predmetu Elektronska vezja 2. del v četrtem letniku univerzitetnega študija elektronike sem izdelal trikanalni voltmeter, ki ga bom uporabil na domačem usmerniku za prikaz izhodnih napetosti.

## 2. OPIS VEZJA

Vezje je trikanalni voltmeter z dosegom napetosti 10V, 20V in 50V na posameznem kanalu. Izmerjeno napetost pa prikaže na treh 7-segmentnih LED prikazovalnikih. Za napajanje je uporabljen 5V napetostni regulator L7805 s pripadajočimi kondenzatorji C1, C2 in C3. Nujen za delovanje mikrokrmilnika vezja je tudi 20MHz kvarčni kristal Q4 s pripadajočima 22pF kondenzatorjema C4 in C5.

### 2.1. *Vhodni del vezja*

Na vhodnem delu vezja so trije uporovni delilniki (R13-R14, R15-R16, R17-R18) za ustrezno atenuacijo merjene napetosti na ustrezen nivo, ki ustreza mikrokrmilniku. Med vsakim parom serijsko vezanih uporov so tudi potenciometri (R19, R20 in R21), ki služijo kalibraciji in fini nastavitvi posameznega območja voltmetra. Da napetost na vhodu v mikrokrmilnik v nobenem primeru ne preseže napetosti, ki je v tolerančnih mejah mikrokrmilnika, so iz vhoda mikrokrmilnika proti masi vezane 5,1V zener diode D1, D2, D3. K vhodnemu delu vezja sodita še dve tipki (PB1 in PB2). Tipka PB1 je reset mikrokrmilnika, tipka PB2 pa služi za izbiranje vhodnih kanalov. Obe tipki stvorita negativno logiko, kar pomeni da je pri razklenjeni tipki na vhodu mikrokrmilnika logična 1 (5V), pri sklenjeni tipki pa 0 (0V).

### 2.2. *Izhodni del vezja*

Izhodni del vezja predstavljajo trije 7 segmentni LED prikazovalniki (DIS1, DIS2, DIS3) s skupno katodo. Upori R2 do R9 služijo za omejitev toka skozi posamezen segment prikazovalnika, upori R10, R11 in R12 skupaj s tranzistorji Q1, Q2 in Q3 pa služijo vklop in izklop posameznega prikazovalnika.

Električna shema celotnega vezja se nahaja v prilogi.

## **3. MIKROKRMILNIK**

### **3.1. Izbira mikrokrmilnika**

Za projekt sem izbral mikrokrmilnik Microchip PIC 16F877A. Glede na to, da je precej vhodov in izhodov mikrokrmilnika še prostih, bi bil primeren tudi manj zmogljiv mikrokrmilnik, ki ima vsaj 3 analogno-digitalne pretvornike in 12 vhodno/izhodnih priključkov. Vendar izbira mikrokrmilnikov, ki bi točno ustrezal tem zahtevam enostavno ni mogoča že zaradi slabe izbire v naših trgovinah z elektroniko.

### **3.2. Opis mikrokrmilnika**

Mikrokrmilnik 16F877A predstavlja vrh družine mikrokrmilnikov 16F8xx. Je 8-bitni mikrokrmilnik in je zmožen delovati s frekvencami do 20MHz. Mikrokrmilnik vsebuje 256 bytov EEPROM-a, 368 bytov pomnilnika RAM in 8K 14-bitnega programskega pomnilnika Flash.

Prav tako vsebuje 8-kanalni 10-bitni A/D pretvornik, ki lahko zajema podatke s hitrostjo 32k vzorcev na sekundo, zaradi česa je še posebej primeren za zahtevnejše aplikacije, ki potrebujejo hitre A/D pretvorbe.

Več informacij o samem mikrokrmilniku in njegovem delovanju pa je na voljo v njegovi dokumentaciji [ 2 ]

## 4. PROGRAM

Program je pisan v prevajalniku MikroC, ki je v omejeni različici na voljo brezplačno na internetu [ 4 ]. Združljiv je s standardom ANSI C in ima vgrajenih že kar nekaj knjižnic, ki olajšajo programiranje mikrokrmilnika (odskakovanje tipk – debouncing, itd).

### 4.1. Koda programa

```
unsigned short i, j, k, osv1, osv2, on, utrip, ston, t=0, gumb=0;
long adc_nap, vred=0, vredV, vredA, povpr=0, izpis=0;

void main() {
    INTCON = 0x00;           // Onemogoči prekinitve
    TRISB = 0xFF;           // Nastavitev PORTB kot vhod
    TRISD = 0x00;           // Nastavitev PORTD kot izhod
    PORTD = 0x00;           // Nastavitev začetnega stanja PORTD
    TRISC = 0x00;           // Nastavitev PORTC kot izhod
    PORTC = 0x00;           // Nastavitev začetnega stanja PORTC
    ADCON1 = 0x82;          // Nastavitev ADC kanalov in Vcc kot Vref+
                           // in GND kot Vref-

    // ***** Animacija zaslona ob vklopu *****

    j=0;
    while (j<2)
    {
        i=0;
        while (i<12)
        {
            PORTD = 0x08;
            PORTC = 0x01;
            Delay_ms(2);
            PORTC = 0x02;
            Delay_ms(2);
            PORTC = 0x04;
            Delay_ms(2);
            i++;
        }

        i=0;
        while (i<12)
        {
            PORTD = 0x04;
            PORTC = 0x01;
            Delay_ms(2);
            PORTC = 0x02;
            Delay_ms(2);
            PORTC = 0x04;
            Delay_ms(2);
            i++;
        }

        i=0;
        while (i<12)
        {
            PORTD = 0x40;
            PORTC = 0x01;
            Delay_ms(2);
            PORTC = 0x02;
            Delay_ms(2);
            PORTC = 0x04;
            Delay_ms(2);
            i++;
        }
    }
}
```

```

i=0;
while (i<12)
{
PORTD = 0x20;
PORTC = 0x01;
Delay_ms(2);
PORTC = 0x02;
Delay_ms(2);
PORTC = 0x04;
Delay_ms(2);
i++;
}

i=0;
while (i<12)
{
PORTD = 0x01;
PORTC = 0x01;
Delay_ms(2);
PORTC = 0x02;
Delay_ms(2);
PORTC = 0x04;
Delay_ms(2);
i++;
}

i=0;
while (i<12)
{
PORTD = 0x02;
PORTC = 0x01;
Delay_ms(2);
PORTC = 0x02;
Delay_ms(2);
PORTC = 0x04;
Delay_ms(2);
i++;
}

i=0;
while (i<12)
{
PORTD = 0x40;
PORTC = 0x01;
Delay_ms(2);
PORTC = 0x02;
Delay_ms(2);
PORTC = 0x04;
Delay_ms(2);
i++;
}

i=0;
while (i<12)
{
PORTD = 0x10;
PORTC = 0x01;
Delay_ms(2);
PORTC = 0x02;
Delay_ms(2);
PORTC = 0x04;
Delay_ms(2);
i++;
}

i=0;
if (j!=0)
{
while (i<12)
{
PORTD = 0x08;
PORTC = 0x01;
Delay_ms(2);
PORTC = 0x02;
Delay_ms(2);
PORTC = 0x04;
Delay_ms(2);
i++;
}
}
j++;
}

```

```

i=0;
while (i<12)
{
PORTD = 0x80;
PORTC = 0x01;
Delay_ms(2);
PORTC = 0x02;
Delay_ms(2);
PORTC = 0x04;
Delay_ms(2);
i++;
}

// ***** Konec animacije *****

do {
if (t>2) t=0; //v primeru t>2, nastavi t na 0
//izbira ADC vhoda glede na t

if (t==0)
{
adc_nap = ADC_read(0); //če je t=0, bere ADC 0
}
else if (t==1)
{
adc_nap = ADC_read(1); //če je t=1, bere ADC 1
}
else if (t==2)
{
adc_nap = ADC_read(2); //če je t=2, bere ADC 2
}

//računanje povprečja branih vrednosti
if (k<8)
{
//v primeru da je prebranih manj kot
//8 vzorcev, poveča k za 1 in prišteje
povpr += adc_nap; //trenjni vzorec k celotni vrednosti
}
else
{
vred = povpr >> 3; //ko je prebranih 8 vzorcev, premik v
povpr=0; //desno za 3 bite (deljenje z 8)
k=0; //reset spremenljivk k in povpr
}

//izračun dejanske vrednosti napetosti
if (t==0)
{
if (vred > 410) //če je vrednost z ADC>410 (410 = 10V)
{
utrip = 1; //vklopi utripanje prikazovalnika in
vredV = 102400; //nastvi prikaz vrednosti 10.0V
}
else
{
utrip = 0; //izklopi utripanje prikazovalnika in
on = 0; //izračunaj vredV
vredV = vred * 250; //250 zato, ker napetost 25V na prvem
//ADC vhodu da 5V na ADC
}
}
else if (t==1)
{
if (vred > 410) //če je vrednost z ADC>410 (410 = 20V)
{
utrip = 1; //vklopi utripanje prikazovalnika in
vredV = 204800; //nastvi prikaz vrednosti 20.0V
}
else
{
utrip = 0; //izklopi utripanje prikazovalnika in
on = 0; //izračunaj vredV
vredV = vred * 500; //500 zato, ker napetost 50V na prvem
//ADC vhodu da 5V na ADC
}
}
}

```

```

else
{
    if (vred > 410)                //če je vrednost z ADC>410 (410 = 50V)
    {
        utrip = 1;                //vklopi utripanje prikazovalnika in
        vredV = 512000;           //nastvi prikaz vrednosti 20.0V
    }
    else
    {
        utrip = 0;                //izklopi utripanje prikazovalnika in
        on = 0;                   //izračunaj vredV
        vredV = vred * 1250;      //1250 zato, ker napetost 125V na prvem
    }                             //ADC vhodu da 5V na ADC
}
vredA = vredV >> 10;             //deljenje s 1024

osv2 = 0;
do {
    osv1 = 0;
    do {
        if (utrip==1)             //če je vklopljeno utripanje
        {
            ++ston;               //poglej ali je trenutno prikazovalnik
            if (on==1)            //vklopljen in če je vklopljen že
            {                       //12 ciklov, potem ga v naslednjem
                if (ston>12)       //ciklu izklopi
                {
                    ston = 0;
                    on = 0;
                }
            }
        }
        else                       //če je vklopljeno utripanje
        {
            if (ston > 12)         //poglej ali je trenutno prikazovalnik
            {                       //izklopljen in če je izklopljen že
                ston = 0;         //12 ciklov, potem ga v naslednjem
                on = 1;           //ciklu vklopi
            }
        }
    }
}

//izpis prve cifre na prvem prikazovalniku

izpis = vredA / 100;
switch (izpis)
{
    case 0: PORTD = 0x00; break;
    case 1: PORTD = 0x06; break;
    case 2: PORTD = 0x5B; break;    // Nastavitev PORTD
    case 3: PORTD = 0x4F; break;    // glede na izračunano
    case 4: PORTD = 0x66; break;    // cifro
    case 5: PORTD = 0x6D; break;
    case 6: PORTD = 0x7D; break;
    case 7: PORTD = 0x07; break;
    case 8: PORTD = 0x7F; break;
    case 9: PORTD = 0x6F; break;
}
if (on==0)
{
    PORTC = 0x01;                 // če je prikazovalnik v ON stanju
    Delay_ms(5);                  // vklopi prvi prikazovalnik za 5ms
    PORTC = 0x00;
}
else
{
    PORTC = 0x00;                 // drugače samo zakasni za 5ms
    delay_ms(5);
}

```



```
//izpis druge cifre na drugem prikazovalniku
```

```
    izpis = (vredA / 10) % 10;
    switch (izpis)
    {
        case 0: PORTD = 0xBF; break;
        case 1: PORTD = 0x86; break;
        case 2: PORTD = 0xDB; break;        // Nastavitev PORTD
        case 3: PORTD = 0xCF; break;        // glede na izračunano
        case 4: PORTD = 0xE6; break;        // cifro
        case 5: PORTD = 0xED; break;
        case 6: PORTD = 0xFD; break;
        case 7: PORTD = 0x87; break;
        case 8: PORTD = 0xFF; break;
        case 9: PORTD = 0xEF; break;
    }
    if (on==0)
    {
        PORTC = 0x02;        // če je prikazovalnik v ON stanju
        Delay_ms(5);        // vklopi drugi prikazovalnik za 5ms
        PORTC = 0x00;
    }
    else
    {
        PORTC = 0x00;        // drugače samo zakasni za 5ms
        Delay_ms(5);
    }
    izpis = vredA % 10;
    switch (izpis)
    {
        case 0: PORTD = 0x3F; break;
        case 1: PORTD = 0x06; break;
        case 2: PORTD = 0x5B; break;        // Nastavitev PORTD
        case 3: PORTD = 0x4F; break;        // glede na izračunano
        case 4: PORTD = 0x66; break;        // cifro
        case 5: PORTD = 0x6D; break;
        case 6: PORTD = 0x7D; break;
        case 7: PORTD = 0x07; break;
        case 8: PORTD = 0x7F; break;
        case 9: PORTD = 0x6F; break;
    }
    if (on == 0)
    {
        PORTC = 0x04;        // če je prikazovalnik v ON stanju
        Delay_ms(5);        // vklopi drugi prikazovalnik za 5ms
        PORTC = 0x00;
    }
    else
    {
        PORTC = 0x00;        // drugače samo zakasni za 5ms
        Delay_ms(5);
    }
} while (osv1>0, osv1--);        // prva zanka za osveževanje zaslona

if (Button(&PORTB, 0, 1, 0)) gumb = 1;        //
if (gumb && Button(&PORTB, 0, 1, 1))        //detekcija pritiska
{
    ++t;        //gumba
    gumb = 0;        //če je gumb pritisnjen
    povpr=0;        //se spremeni stanje 0. bita
    k=0;        //na PORTB
}        //resetiranje spremenljivk

} while (osv2>0, osv2--);        //druga zanka za branje gumba
} while (1);
}
```

## 4.2. Opis programa

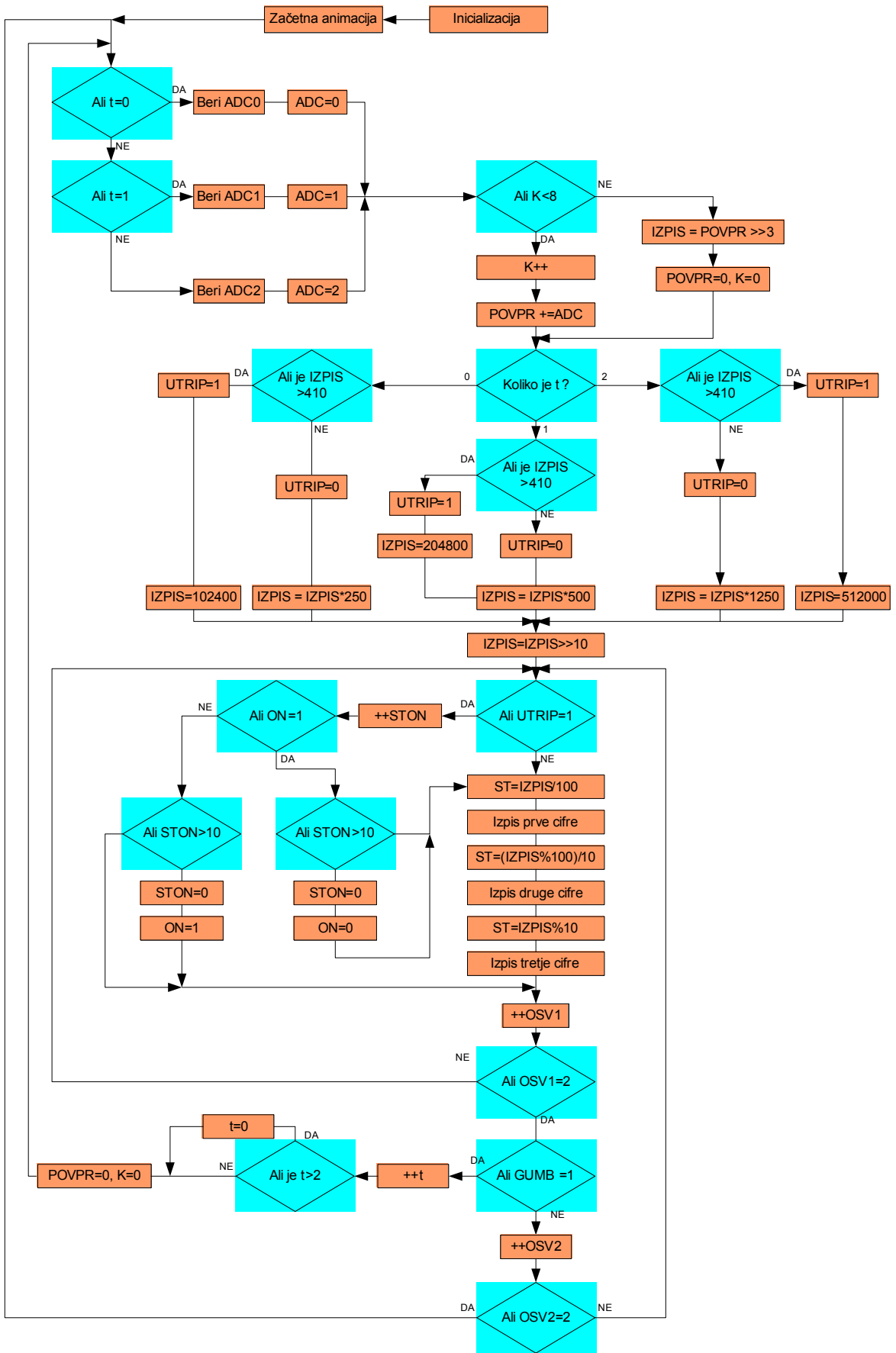
Program najprej na začetku inicializira ustrezne porte kot vhode ali izhode in postavi izhodom določene vrednosti. Nato na prikazovalniku ob vklopu izriše kratko animacijo. Sledi branje analogno-digitalnega pretvornika. Spremenljivka  $t$  določa iz katerega kanala program prebere vrednost.

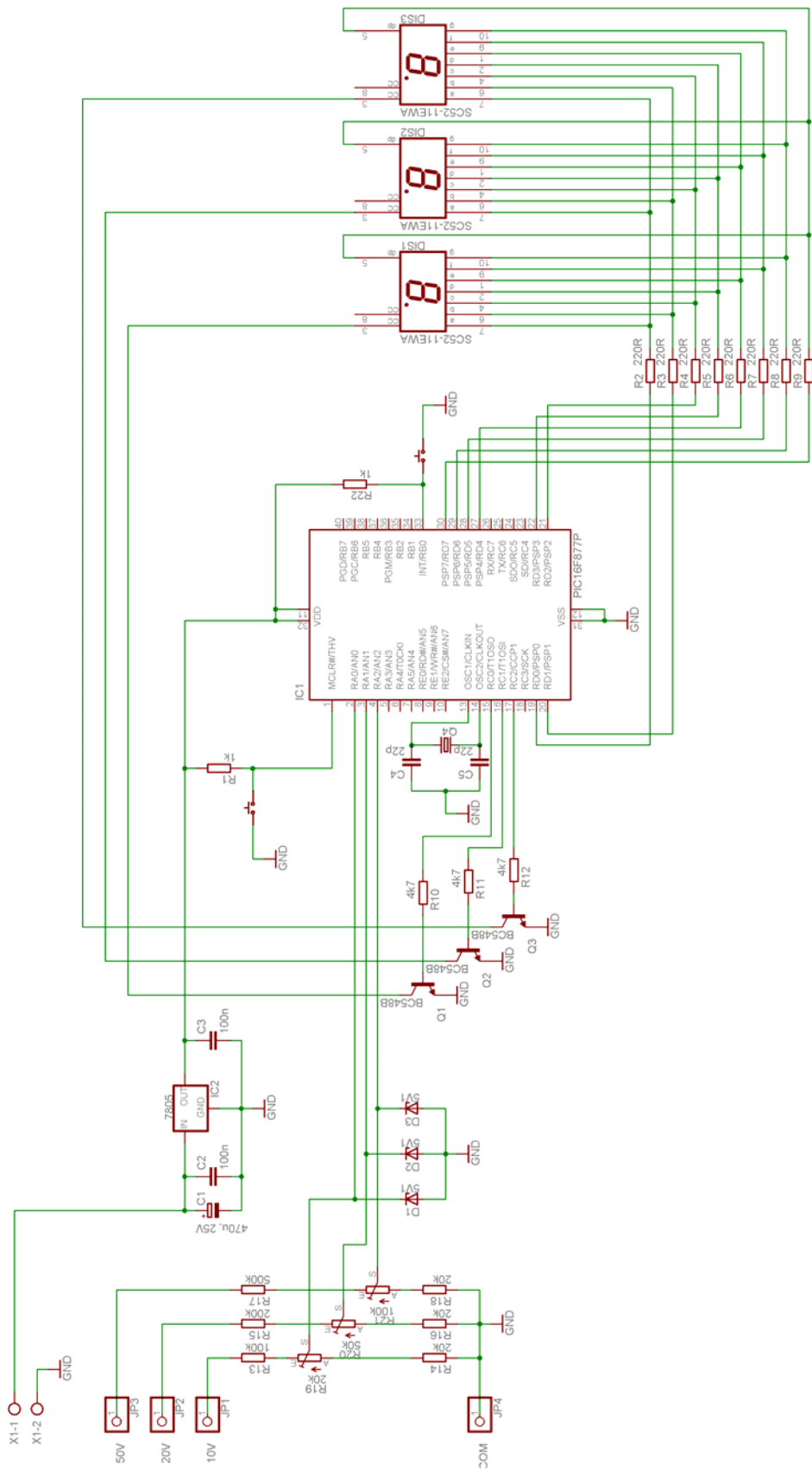
Po branju ADC sledi odločitveni stavek, ki izračuna povprečje 8 vzorcev prebranih iz ADC. Ko je povprečje izračunano, program glede na spremenljivko  $t$  določi na katerem intervalu merimo napetost in v primeru da je napetost višja kot je dano z intervalom, vklopi utripanje zaslona in izpiše maksimalno vrednost trenutnega intervala. Če je napetost v dopustnem intervalu, se sproži izračun vrednosti, ki je ustrezen za izpis na prikazovalniku. Program nato izvrši utripanje zaslona ali normalen izpis glede na izračunano vrednost napetosti. V primeru normalnega izpisa za 5ms izpiše na prvi prikazovalnik prvo cifro, nato za 5ms drugo cifro in še za 5ms tretjo cifro.

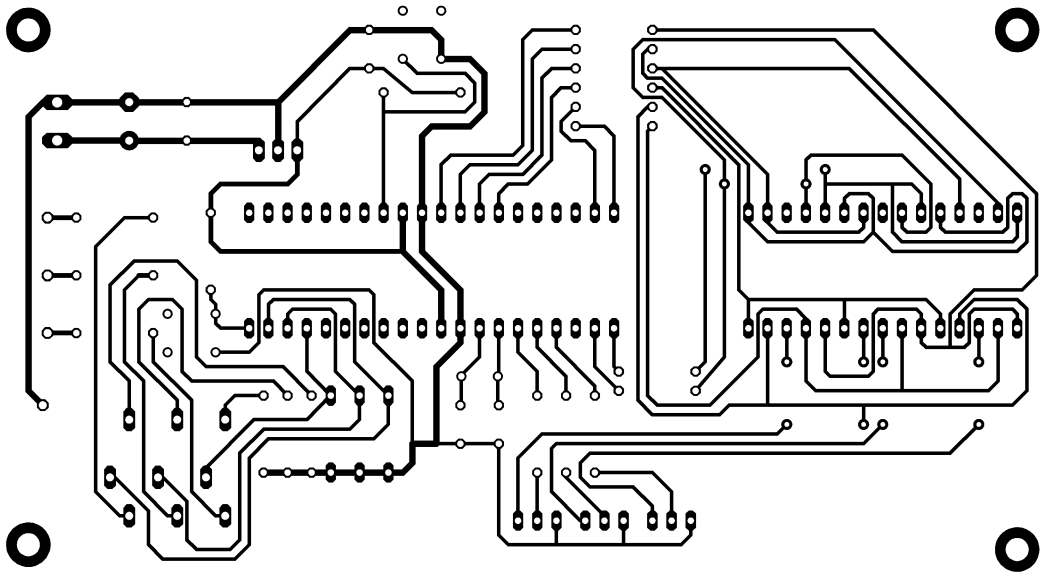
Osveževanje zaslona je urejeno tako, da se ponavlja s takšno hitrostjo, da na zaslonu ni opaziti utripanja. Po prikazu vrednosti na zaslonu program preveri še ali je slučajno pritisnjen gumb za preklon na drugi kanal in ustrezno temu nastavi spremenljivko  $t$ . Na koncu se celoten program spet ponovi.

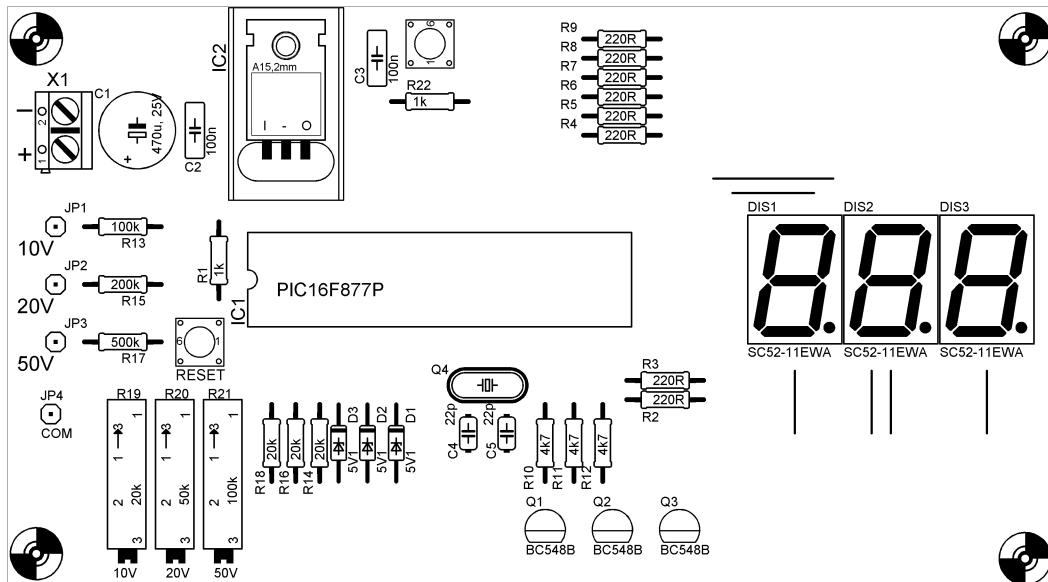
## 5. PRILOGE

- diagram poteka programa (flowchart);
- električna shema;
- načrt tiskanega vezja;
- montažni načrt;
- kosovnica;
- cena materiala.









## KOSOVNICA:

Upori	Opis	Količina
R2, R3, R4, R5, R6, R7, R8, R9	220Ω, 1/4W	8
R1, R22	1kΩ, 1/4W	2
R10, R11, R12	4,7kΩ, 1/4W	3
R14, R16, R18	20kΩ, 1/4W	3
R13	100kΩ, 1/4W	1
R15	200kΩ, 1/4W	1
R17	500kΩ, 1/4W	1
R19	20kΩ, multiturn potenciometer 43P	1
R20	50kΩ, multiturn potenciometer 43P	1
R21	100kΩ, multiturn potenciometer 43P	1
<b>Kondenzatorji</b>		
C1	470uF, 25V, elektrolit	1
C2, C3	100nF, 50V, keramika	2
C4, C5	22pF, 50V, keramika	2
<b>Polprevodniki</b>		
Q1, Q2, Q3	BC548, NPN tranzistor	3
Q4	Kvarčni kristal 20MHz	1
D1, D2, D3	Zener dioda 5,1V, 0,5W	3
IC1	Microchip PIC 16F877A	1
IC2	L7805	1
DIS1, DIS2, DIS3	SC52-11EWA, 7 segmentni LED prikazovalnik 18mm	3
<b>Ostalo</b>		
PB1, PB2	Tipka 6mm	2
JP1, JP2, JP3, JP4	Pin letvice	4
X1	Vrstna spojka 2-polna	1
<b>Razno</b>		
	Hladilnik HRU 15/25	1
	Podnožje 40-pin	1



## CENA MATERIALA

	kosov	cena za kos z DDV	skupaj
Upor 1/4W, 5%	13	0,04	0,52
Upor 1/4W, 1%	6	0,06	0,36
Multiturn potenciometer 43P	3	0,85	2,55
Kond. 470 $\mu$ F, 25V, elektrolit	1	0,19	0,19
Kond. 100nF, 50V, keramika	2	0,05	0,10
Kond. 22pF, 50V, keramika	2	0,05	0,10
Tr. BC548	3	0,15	0,45
L7805	1	0,71	0,71
PIC 16F877A	1	11,89	11,89
Zener dioda 5,1V, 1/2W	3	0,13	0,39
Kvarčni kristal 20MHz	1	0,65	0,65
7 segmentni LED display - SC52-11EWA	3	1,40	4,20
Tipka 6mm	2	0,15	0,30
Vrstna spojka 2-polna	1	0,40	0,40
Hladilnik HRU 15/25	1	1,60	1,60
Podnožje 40-pin	1	0,71	0,71
Pin letvice	4	0,02	0,08
Foto plošča 160x100mm - positiv	1	3,00	3,00
		Skupaj:	28,20€

Za celotno ceno projekta je potrebno k ceni materiala prišteti še pomožna sredstva za izdelavo ploščice tiskanega vezja (razvijalec, sredstvo za jedkanje, itd). Ocenjen čas, ki sem ga porabil za realizacijo tega projekta pa je okoli 40ur.

## 6. VIRI

- [ 1 ] F. Bergelj – Meritve II. Del
- [ 2 ] Microchip PIC16F87XA Data Sheet
- [ 3 ] [http://www.mikroe.com/en/books/picbook/0\\_Uvod.htm](http://www.mikroe.com/en/books/picbook/0_Uvod.htm)
- [ 4 ] <http://www.mikroe.com/en/compilers/mikroc/pic/>