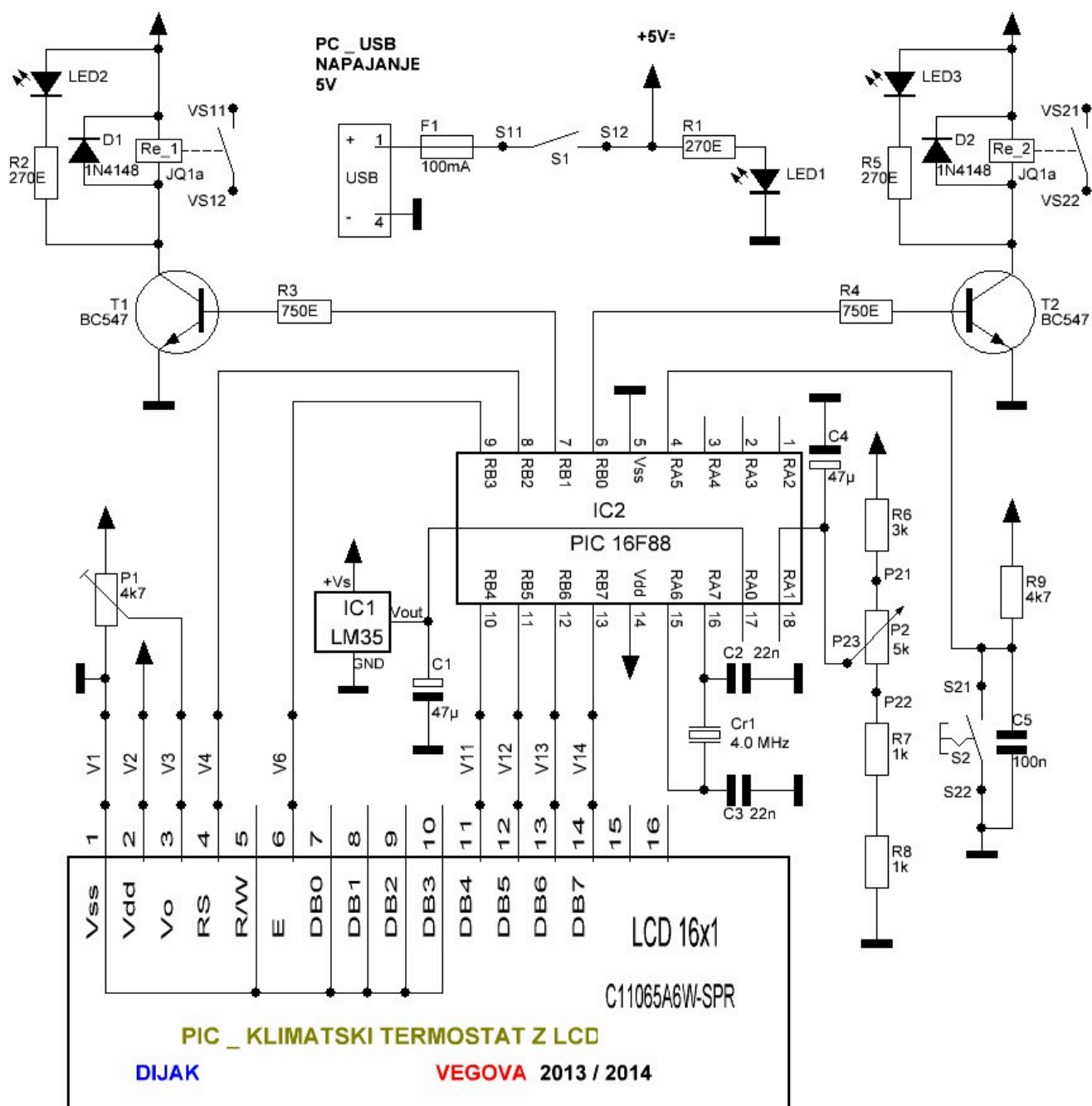


1. PIC klimatski termostat z LCD display-em _ električna shema (sPL70)



P1: LCD KONTRAST

P2: IZBIRA TEMPERATURE

IC1: TEMPERATURNI SENZOR

S2: PIC RESET

Re1: I VKLOP GRETJA

Re2: VKLOP HLAJENJA

S1: VKLOP NAPA JANJA

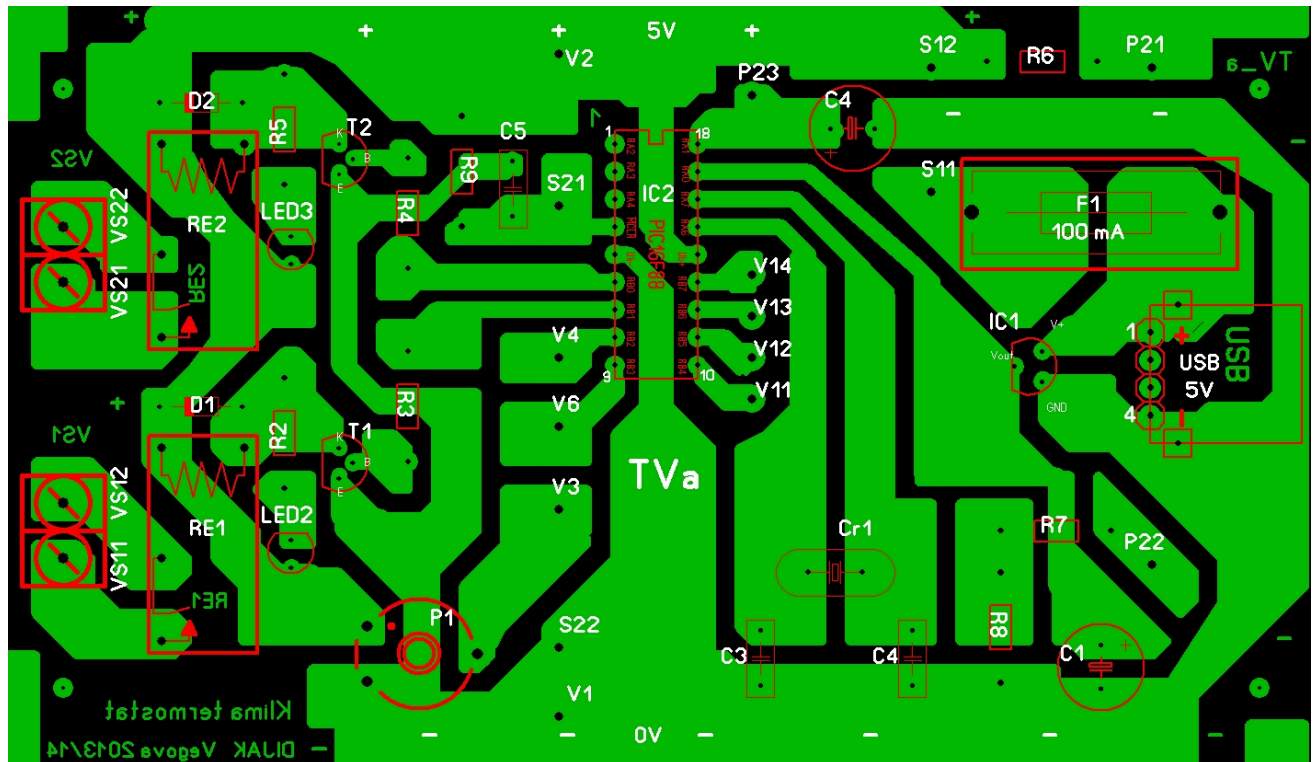
LED1: VKLOPLJENO NAPA JANJE

LED1: VKLOPLJENO GRETJE

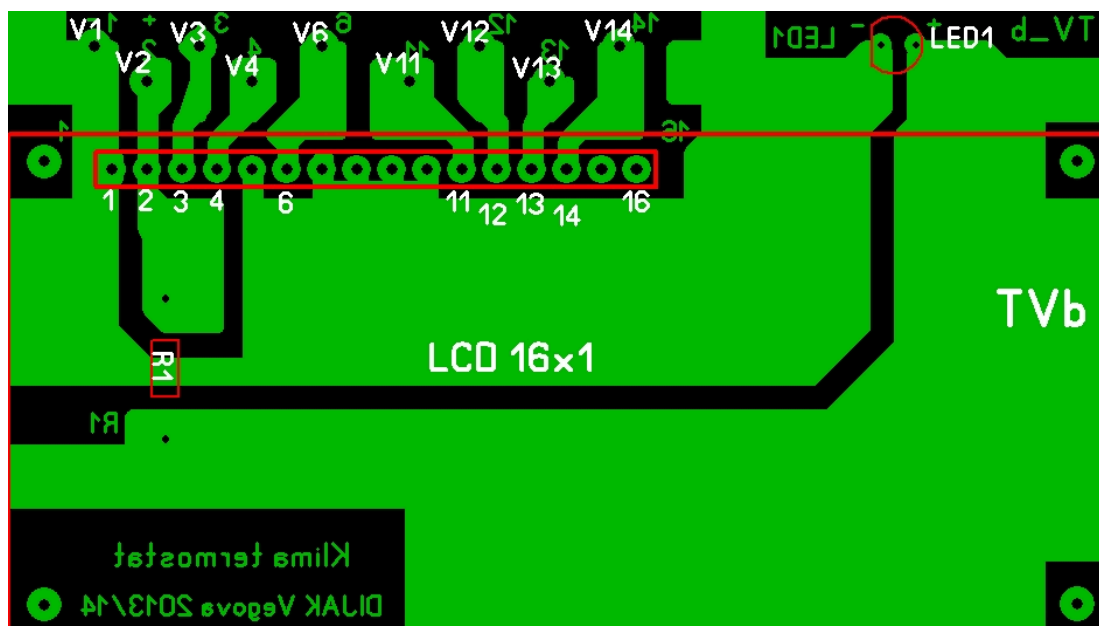
LED3: VKLOPLJENO HLAJENJE

2. PIC klimatski termostat z LCD display-em _ tiskani vezji (LAY50)

2.1 Tiskano vezje_a (TVa)



2.2 Tiskano vezje_b (TV b)



3. PIC klimatski termostat z LCD display-em _program za PIC16F88 (mikro C)

```
// 7 KLIMATSKI TERMOSTAT Z LCD DISPLAY-EM //
```

```
// Aplikacijski program 7TERMOSTATdecNORMAL1 // D E L U J O E A sintaksa //
```

```
// verzija z prioritetnim prikazom REF med nastavljanjem potenciometra //
```

```
// Narejeno iz testne verzije programa 7TERMOSTATdecNORMALtest1 //
```

```
// Termostat ima potenciometer(REF) vezan na analogni vhod A1, //
```

```
// temperatuurni sezor/merilni pretvornik LM35(T) pa na A0. //
```

```
// REF merimo na potenciometru, T merimo na izhodu LM35 //
```

```
// LCD izmenoma prikazuje REF, T in stanje regulacije. //
```

```
// Spreminjanje REF vklopi zaeasno zanko s prikazom REF brez //
```

```
// obdelave meritve T, ter brez prikaza T in stanja regulacije. //
```

```
// Regulacija krmili korekcijo temperature:  $T = REF + (-) \text{dif}T$ . //
```

```
// Digitalni izhod B0 preko releja vklaplja hlajenje, //
```

```
// digitalni izhod B1 pa vklaplja gretje. //
```

```
// Gretje označuje prižgana rdeča LED. //
```

```
// Hlajenje označuje modra LED. //
```

```
// delovanje še ni preverjeno za LCD funkcije v živo na vezju //
```

```
// testni Usart izpisi namesto LCD so pravilni //
```

```
// aplikacijska Verzija test je za odprte sponke na A0 dala 0 //
```

```
// meritev na A4 in prikaz deluje pravilno pri testni verziji1 //
```

```
// regulacija deluje pravilno, nastavljivost REG je ok //
```

```
// potrebna še naknadna optimizacija zakasnilnih časov //
```

```
// prvotno je bila REF na A4, vendar je zaradi lažje izdelave //
```

```
// vezja prestavljeno na A1 //
```

```
////////////////////////////////////
```

```
// ZAČETEK PROGRAMA //
```

```
void main()
```

```

{
// DEKLARACIJE IN PRIREDITVE SPREMENLJIVK //

float T = 51.0; // deklarira T in priredi vrednost 51.0 //

int Tx10 = 999; // kalkulacija decimalke //

int Tc = 99; // cifri celomestnega dela //

int Td = 9; // cifra decimalnega mesta //

float REF = 510.0; // deklarira REF in priredi vrednost 510.0 //

int REFx10 = 999; // kalkulacija decimalke //

int REFc = 99; // cifri celomestnega dela //

int REFd = 9; // cifra decimalnega mesta //

char podatek[5] = "26.0"; // znakovni niz za vrednost T //

char podatek0[5] = "25.0"; // znakovni niz za vrednost REF //

char beseda[9] = "T('C) ="; // znakovni niz napisa za prikaz T //

char beseda0[9] = "REF('C)="; // znakovni niz napisa za prikaz REF //

// znakovni nizi besedil za prikaz stanja regulacije: //

// char reg_M [9] = "OK "; // reg=3 : "ok ", regulacija miruje //

// char reg_G [9] = "GRETJE "; // reg=1 : "GRETJE ", regulacija deluje //

// char reg_H [9] = "HLAJENJE"; // reg=2 : "HLAJENJE", regulacija deluje //

// int reg = 0; // switch indikator za izbiro prikazov //

int difT; // +/- toleranca regulacije T //

// histereza je 2*difT/10 //

// opcionalni dodatek v pomoč nastavljanja REF: //

int oldr = 250; // int pretvorbo terjajo math funkcije //

int newr = 259; // //

int difREF; // sprememba nastavitve REF med dvema meritvama REF//

int prag; // prag odkritja spremembe REF //

// diskriminanta detekcije nastavljanja REF = prag / 10 //

```

```

// INICIALIZACIJE IN RESET          //

PORTB = 0;    // reset vseh pinov porta B    //

Lcd_Init (&PORTB); // default konfiguracija povezav LCD - PIC //


// NASTAVITEV MERITEV z uporabo ADC in analognih vhodov //

ANSEL = 0b00010011; // A1 in A0 definiramo za analogna vhoda //

ADCON1 = 0x80;    // definiramo uro in UREF ?    //

TRISA = 0b00000011; // A1 in A0 definiramo za vhoda    //

// ANSEL = 0b00010001; // A4 in A0 sta analogna vhoda //

// TRISA = 0b00010001; // A4 in A0 definiramo za vhoda //

// NASTAVITEV DIGITALNIH VHODOV IN IZHODOV //

TRISB = 0; // Vsi B pini so izhodi    //


// NASTAVITEV PARAMETROV - PRIREDITVE VREDNOSTI SPREMENLJIVK //

difT = 5; // nastavitev +/- tolerance regulacije T    //

//    histereza je 2*difT/10 v st C    //

//    opcionalni dodatek v pomoč nastavljanja REF:    //

difREF = 5; // začetna prireditve vrednosti spremembe REF    //

prag = 4; // nastavitev praga za odkritje spremembe REF    //

//    diskriminanta je prag / 10 v st C    //

//    detekcija nastavljanja pri difREF > prag    //

while(1)    // GLAVNA NEPOGOJNA ZANKA    //

{

// 1. ZAJEM MERITEV IN SHRANITEV V SPREMENLJIVKI //

// 1.1. meritev regulirane temperature T //

T = Adc_Read(0); // zajem meritve T na A0    //

T = T * 488 / 1000; // merilna konstanta 0.488 za 500 st C pri 5 V //

```

```

Tx10 = T * 10;    // del algoritma za izraeun decimalnega mesta //
Td = Tx10 % 10;   // ostanek deljenja z 10 da cifro decimalke //
Tc = Tx10 / 10;   // deljenje z 10 da celomestni del števila //
nastavljanje: //  opcionalni dodatek v pomoe nastavljanja REF: //

// 1.2. meritev referenene temperature REF //

REF = Adc_Read(1);    // zajem meritve REF na A1          //
REF = REF * 488 / 10000; // merilna konstanta 0.0488 za 50 st C pri 5 V //
REFx10 = REF * 10;    // del algoritma za izraeun decimalnega mesta //
REFd = REFx10 % 10;   // ostanek deljenja z 10 da cifro decimalke //
REFc = REFx10 / 10;   // deljenje z 10 da celomestni del števila //

// REF = Adc_Read(4); // zajem meritve REF na A4      //

1.3. Komentar določanja merilne konstante za T in REF: //

//LM 35 da 1,5 V pri 150 st C. 5 V na A0 predstavlja torej 500 st C.    //

//5 V (interna REF. napetost) da pri 10 bitni intiger meritvi vrednost 1023. //

//Za pravilno vrednost temperature T moramo to množiti z merilno konstanto K. //

//500 = K * 1023 => K = 500 / 1023 = 0.4888.                //

//Sedaj 5 V na analognem vhodu A0 da vrednost meritve 1023 * 0.488 = 500 st C, //

//0.5 V na izhodu LM35 pri 50 st C pa da torej 10x manjšo vrednost 50 st C. //

// //

//Meritev REF je nastavljiva v razponu 10 do 35 st C. //

//5 V na A1 predstavlja 50 st C (prvotno na A4).    //

//Merilna konstanta bo torej 10 x manjša, to je 0.0488. //

// //

// 2. FUNKCIJA REGULACIJE //

// 2.1 test prenizke temperature T          //

if (Tx10 < (REFx10 - difT)) // T nižja od spodnje meje //

```

```

{
//          vklop gretja          //

PORTB.F0 = 0; // B0 = 0 za predhodni reset hlajenja    //

Delay_ms(500);

PORTB.F1 = 1; // B1 = 1 za vklop gretja, gori rdeea LED  //

// reg = 1; // switch spremenljivka za napis case 1    //

}

//          izklop gretja          //

else

PORTB.F1 = 0; // B1 = 0 za reset gretja, ugasne rdeea LED  //

// 2.2 test previsoke temperature T          //

if (Tx10 > (REFx10 + difT)) // T višja od zgornje meje    //

{

//          vklop hlajenja          //

PORTB.F1 = 0; // B1 = 0 za predhodni reset gretja    //

Delay_ms(500);

PORTB.F0 = 1; // B0 = 1 za vklop hlajenja, gori modra LED  //

// reg = 2; // switch spremenljivka za napis case 2    //

}

//          izklop hlajenja          //

else

PORTB.F0 = 0; // B0 = 0 za reset hlajenja, ugasne modra LED //

// reg = 3; // switch spremenljivka za napis case 3    //

// ta sekvenca algoritma regulacije pomeni T = ok    //

// rdeea in modra LED ugasneta          //

// 3. FUNKCIJA PRETVORBE FLOAT ŠTEVILA V CHAR ZNAKOVNI NIZ //

```

```

// 3.1. Pretvorba vrednosti T v znakovni niz //

podatek[0] = Tc / 10 + 48; //dobim ASCII kodo za prvo levo celomestno cifro //
podatek[1] = Tc % 10 + 48; //dobim ASCII kodo za desno celomestno cifro //
podatek[2] = 46; //dodam ASCII kodo za decimalno piko //
podatek[3] = Td + 48; //dobim ASCII kodo za cifro decimalnega mesta //

// 3.2. Pretvorba vrednosti REF v znakovni niz //

podatek0[0] = refc / 10 + 48; //dobim ASCII kodo za prvo levo celomestno cifro //
podatek0[1] = refc % 10 + 48; //dobim ASCII kodo za desno celomestno cifro //
podatek0[2] = 46; //dodam ASCII kodo za decimalno piko //
podatek0[3] = refd + 48; //dobim ASCII kodo za cifro decimalnega mesta //

// 4. FUNKCIJA LCD prikazovalnika //

// 4.1. Prikaz REF //

Lcd_Cmd(Lcd_CLEAR); // Briše LCD (enkrat Lcd, enkrat LCD ?) //
Lcd_Cmd(LCD_CURSOR_OFF); // Izklopi prikaz kurzorja //
LCD_Out(1,1,beseda0); // PIC pošlje na LCD napis za REF (v=1, s=1) //
LCD_Out(1,10,podatek0); // PIC pošlje na LCD vrednost REF //
Delay_ms(2000); // prilagojen prikaz //

// detekcija nastavljanja REF in sprememba programskega toka //
// opcionalni dodatek v pomoč nastavljanja REF: //
difREF = abs(newr - oldr); // absolutna vrednost spremembe REF //
if ( difREF > prag ) // razlika 2 meritev REF > prag //
{
    // (prag = 0.4 st C) //
    goto nastavljanje; // programski preskok na 'nastavljanje' //
} // ( prikaz bolje sledi spremembi REF ) //

// zaeasna zanka meritve in prikaza REF ter regulacije //

// 4.2. Prikaz T //

```



```

Lcd_Cmd(Lcd_CLEAR); // Briše LCD (enkrat Lcd, enkrat LCD ?) //
Lcd_Cmd(LCD_CURSOR_OFF); // Izklopi prikaz kurzorja //
LCD_Out(1,1,beseda); // PIC pošlje na LCD napis za T (v=1, s=1) //
LCD_Out(1,10,podatek); // PIC pošlje na LCD vrednost T //
Delay_ms(2000); // prilagojen prikaz //

// 4.3. Prikaz stanja regulacije //

// odreže vse spremenljivke za prikaz (truncated) //

//Lcd_Cmd(Lcd_CLEAR); // Briše LCD (enkrat Lcd, enkrat LCD ?) //
//Lcd_Cmd(LCD_CURSOR_OFF); // Izklopi prikaz kurzorja //
//switch (reg) // prikaz besedila glede na stanje regulacije //
//{
//case 1: LCD_Out(1,1,reg_G); // GRETJE //
//break;
//case 2: LCD_Out(1,1,reg_H); // HLAJENJE //
//break;
//case 3: LCD_Out(1,1,reg_M); // OK //
//break;
//default: LCD_Out(1,1,"NAPAKA!"); // NAPAKA //
//}

//Delay_ms(500); // prilagojen prikaz //

GLAVNA NEPOGOJNA ZANKA //

}

// KONEC PROGRAMA //

////////////////////////////////////

```