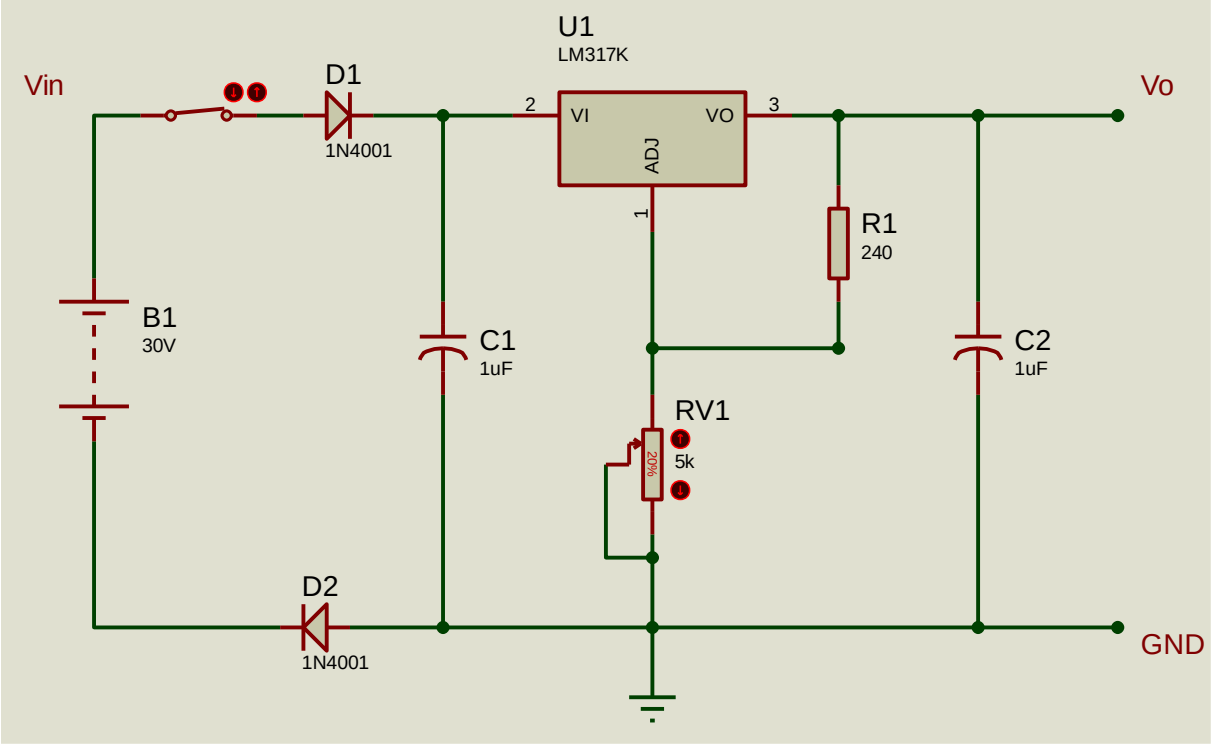
P.C.I.S. BİLGİ İŞLEM VE DANIŞMANLIK

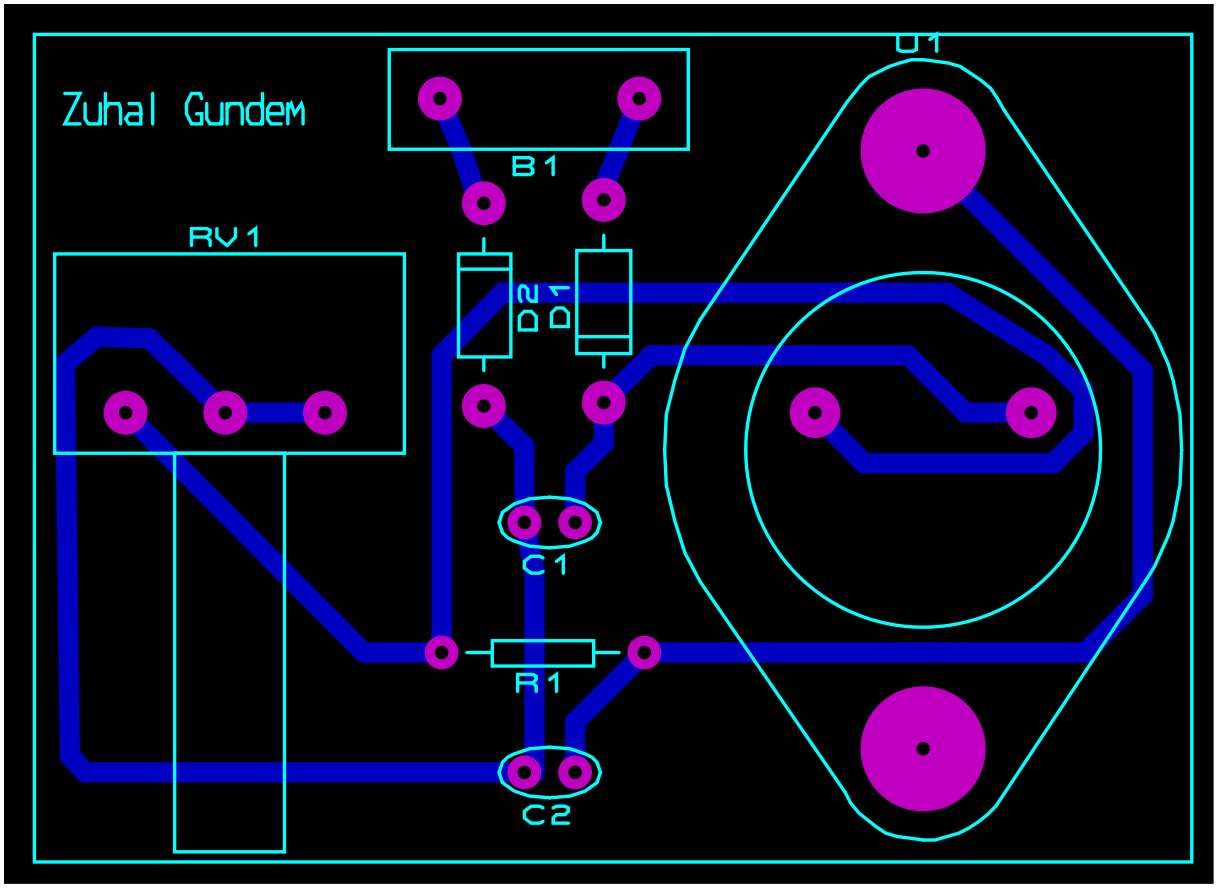
MÜLAKAT SORULARI

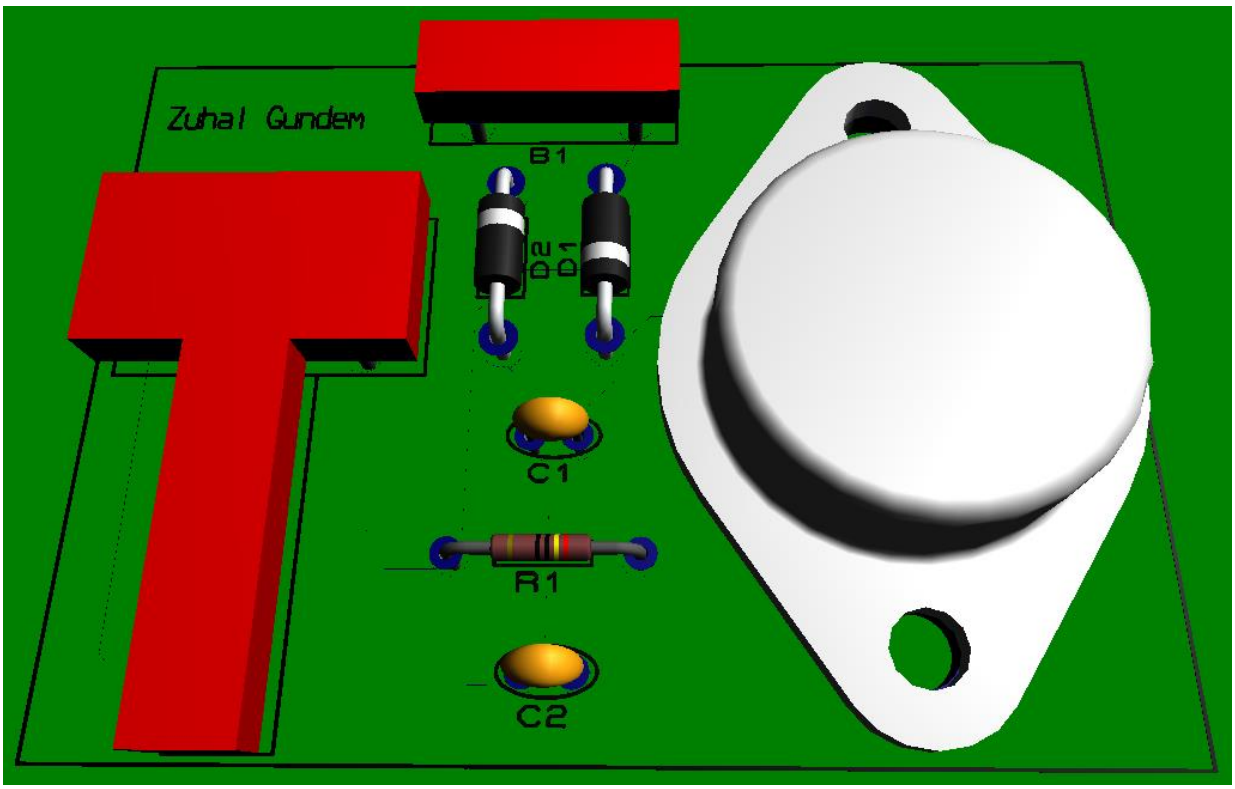
1 - Ters Voltaj Korumalı Ve Ayarlı Voltaj Regülatör Devresi

Şematik Devre:



Baskı Devre (PCB):





Devre İle İlgili Bilgiler:

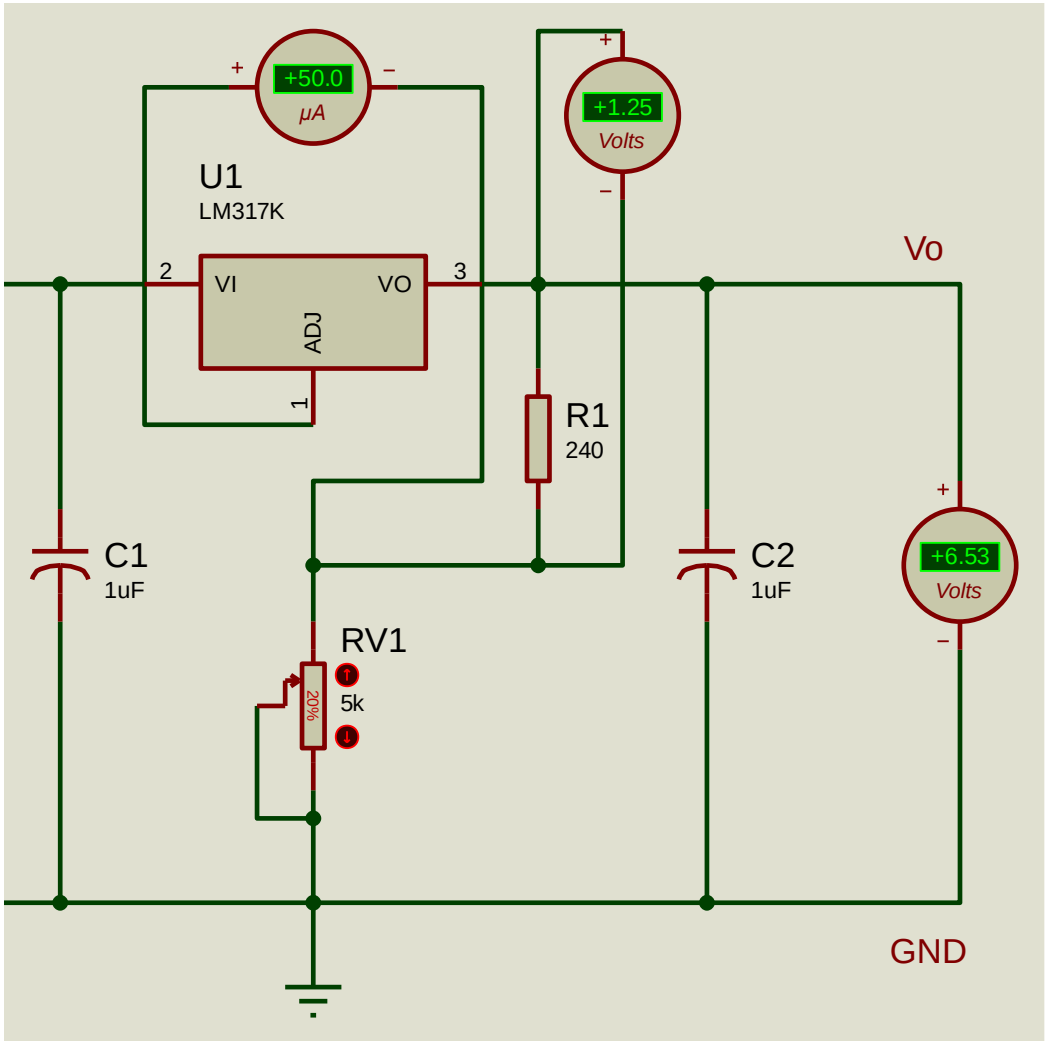
Bu devrede ayarlanabilir voltaj regülatörü olan LM317 devre elemanı kullanılmıştır. LM 317 elemanının datasheet'i incelenmiş ve uygun bağlantıları sağlanmıştır. Ayarlanabilir bacak olan 1. bacak ile toprak hattı arasına 5K potansiyometre bağlanarak datasheet'deki formüle uygun direnç oluşturulduğunda istenilen gerilim çıkışta alınmıştır. Bu entegre 1.25V ile 37V arasında çıkış gerilimi ve maksimum 1.5 amper akım vermektedir. Sisteme 30V giriş uygulandığı için maksimum çıkış bu devre için 27.2V olarak gözlenmiştir. Giriş ve çıkıştaki dalgalanmaları engellemek için kondansatör kullanılmıştır. Girişin + ve – uçlarına akıma uygun diyotlar konularak ters voltaj koruması sağlanmıştır.

Datasheet'teki formül aşağıdaki gibidir;

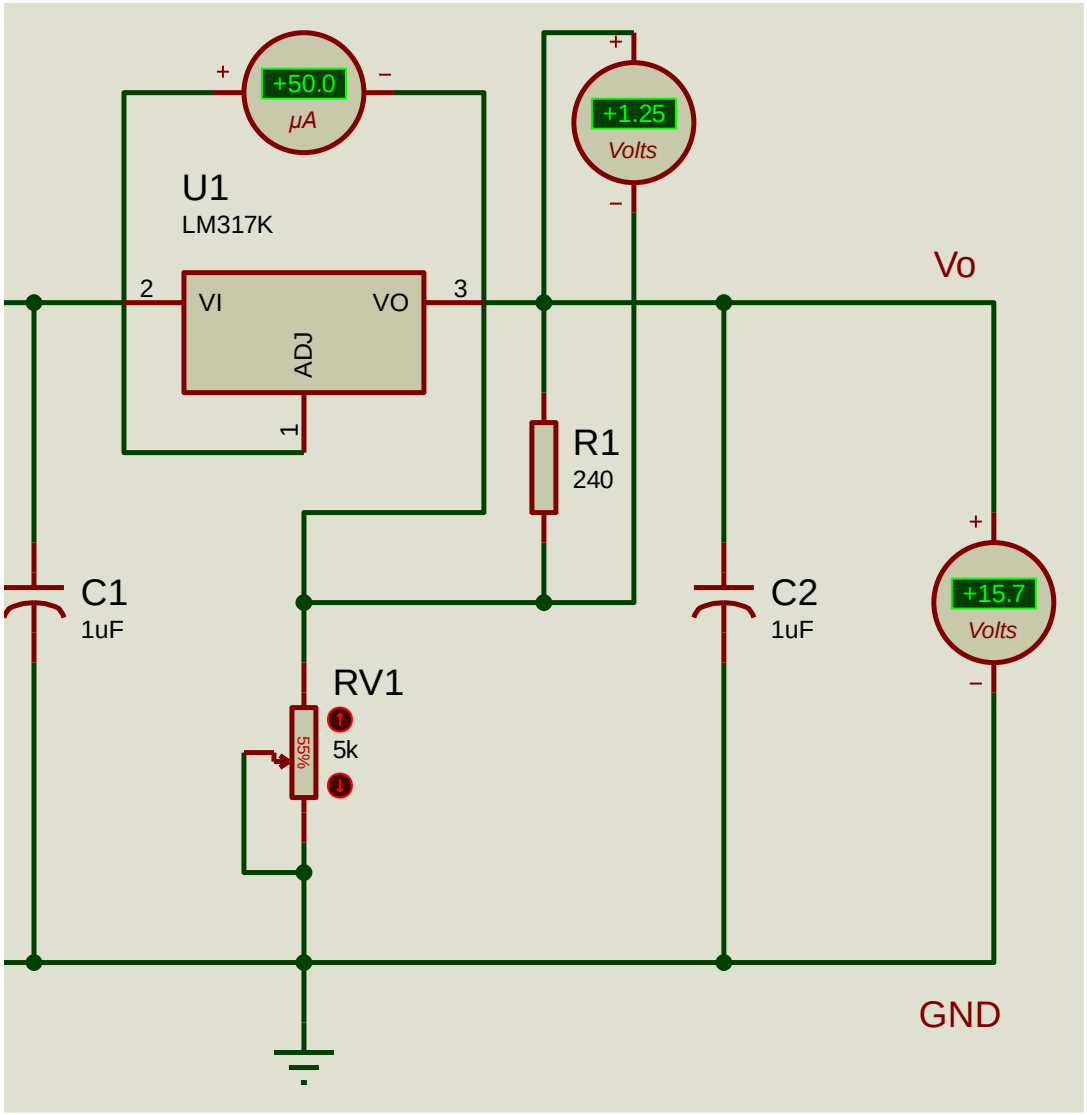
$$v_0 = v_{ref}\left(1 + \frac{R_2}{R_1}\right) + \left(I_{ADJ} * R_2\right)$$

- Formüldeki v_{ref} çıkış ve ayarlama bacakları arasındaki voltajtır. (1.25V)
- I_{ADJ} ayar bacağındaki akım yani 50 µA'dır. (Çok küçük olduğu için ihmal edilebilir.)
- R_1 sabit direnç 240 Ω ve R_2 5K'lık potansiyometredir.

Bu formüle göre potansiyometreyi 1000Ω'a ayarlarsak v_0 6.45V olur. Proteus simülasyonunda 6.53V gözlenmiştir.



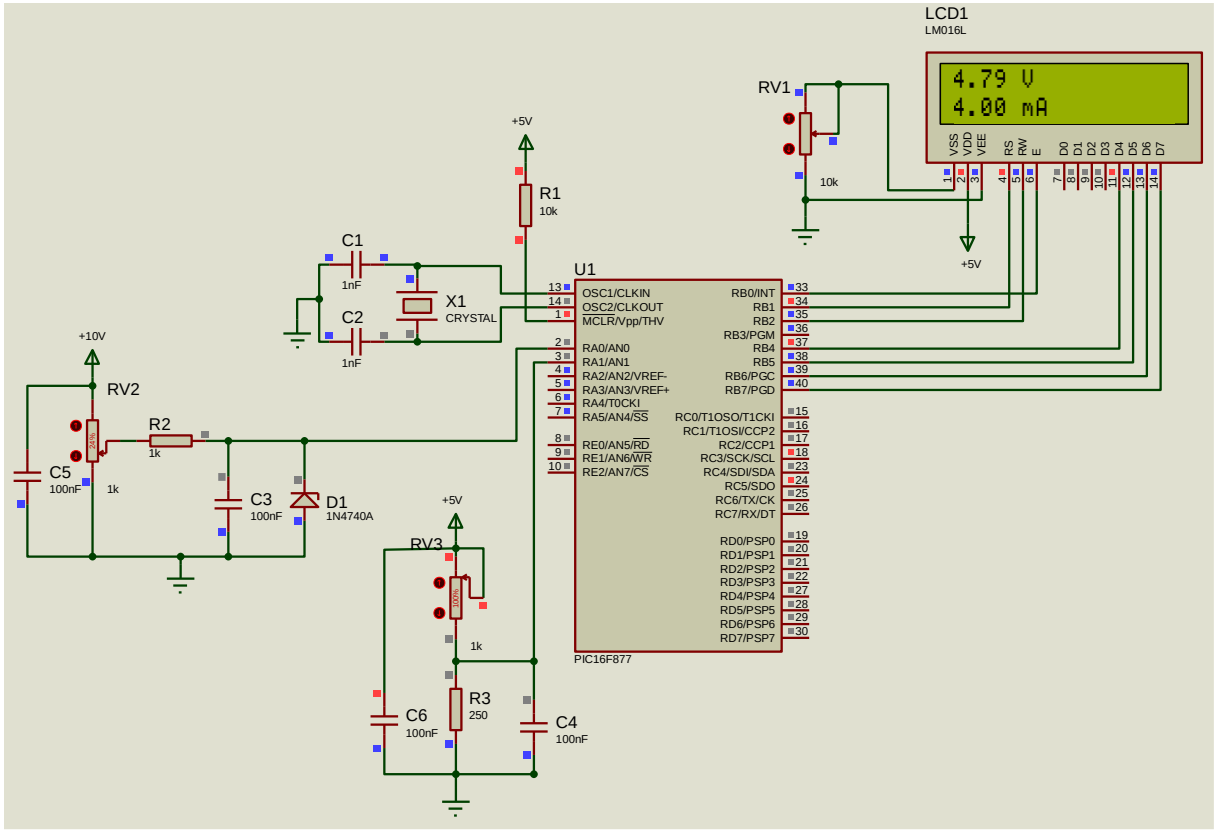
Formüle göre potansiyometreyi 2750Ω'a ayarlarsak v_0 15.57V olur. Proteus simülasyonunda 15.7V gözlenmiştir.



- Son olarak devre tasarımının gerber dosyası oluşturulmuş ve kaydedilmiştir.

2 - 3.3 V_{DC} İle Çalışan Mikro Denetleyici Analog Ölçüm Devresi

0-10V ve 4-20mA Ölçüm İçin Şematik Devre (PIC16F877 için):



Devre İle İlgili Bilgiler:

Analog ölçüm için mikrodenetleyici olarak PIC16F877 tercih edilmiş ve kodlama için CCSC programı kullanılmıştır. PIC16F877 mikrodenetleyicisi için datasheet incelenmiş ve Proteus programında pic için osilatör ve besleme gerilimi bağlantısı yapılmıştır. Analog ölçüm değerlerinin takibi için LCD bağantıları da yapılmıştır.

Analog girişlerinden A0 pinine 0-10V voltaj ölçümü için gerilim bölücü devre tasarlanmıştır.

$$Vo = Vin \frac{R2}{R1+R2}$$

Yukarıdaki gerilim bölücü formülü kullanılarak potansiyometre değeri değiştirilip sabit direnç üzerine düşen voltaj gözlenmiştir. Devredeki dalgalanmaları filtrelemek için giriş voltajına ve mikrodenetleyicinin adc bacağına kondansatör bağlanmıştır.

CCSC programında mikrodenetleyicinin gerekli ayarlamaları yapılmış, lcd kütüphanesi eklenmiş, değişkenler tanımlanmış, giriş ve çıkış pinleri belirtilmiştir. Daha doğru sonuç almak için yazılımsal olarak da filtreleme yapılmıştır. 20 kez adc değeri okunarak bir değişkene atılmış ve ortalaması alınmıştır. 10V giriş voltatjı adc dönüşümü için 1024'e (10 bit) bölünerek okunan değer ile çarpılmış ve lcd ekranına yazdırılmıştır.

A1 pininden ise 4-20mA akım ölçümü için başka bir gerilim bölücü devre tasarlanmıştır.

$(V = I * R)$

Yukarıdaki temel elektrik formülü ile 4 ve 20 mA değerlerini sağlayan direnç değerleri hesaplanmıştır.

5V için; 1250Ω -> 4mA

250Ω -> 20mA

Bu değerlere için 1k'lık potansiyometre ve 250Ω sabit direnç ile sabit direncin üzerinden geçen voltaj değeri okunmuş ve yazılımda yukarıdaki formül ile mA cinsine çevrilerek ekrana yazdırılmıştır. Yine devredeki dalgalanmaları önlemek amacı ile kondansatör kullanılmıştır.

Daha doğru sonuç almak için bu devrede de yazılımsal olarak da filtreleme yapılmıştır. 20 kez adc değeri okunarak bir değişkene atılmış ve ortalaması alınmıştır. 5V giriş voltatjı adc dönüşümü için 1024'e (10 bit) bölünerek okunan değer ile çarpılmıştır. Bulunan değer voltaj olduğu için sabit direnç olan 250Ω'a bölünmüş ve mA cinsine çevirmek için 1000 ile çarpılmıştır. Sonuç lcd ekranına yazdırılmıştır.

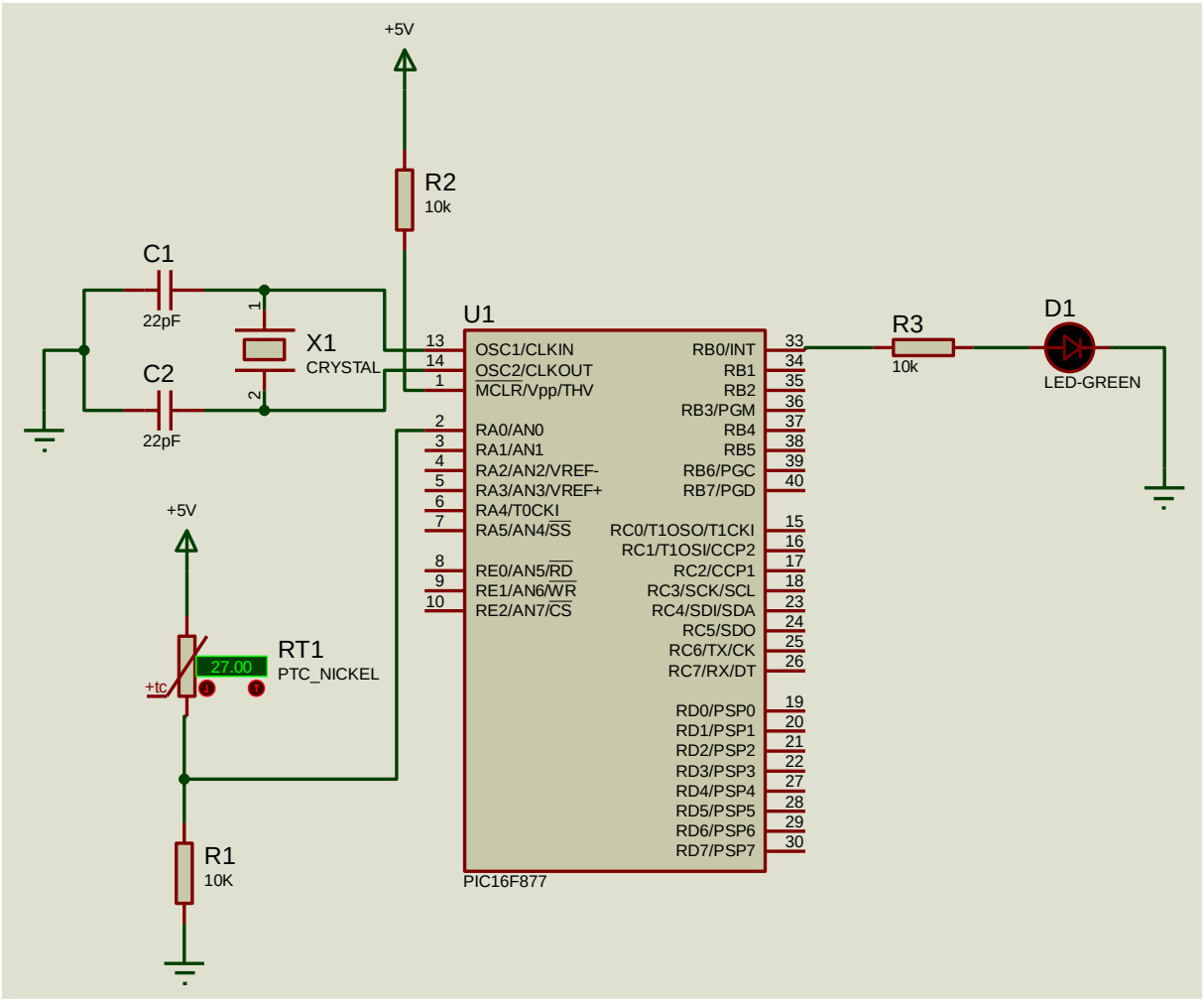
CCSC Kodları ve Açıklamaları:

```
#include <16f877.h>            // denetleyicinin baslik dosyasi tanimlama
#fuses XT, NOWDT, NOPROTECT, NOBROwNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPD       // Denetleyici konfigürasyon ayarlari

#device ADC=10
#include <math.h>           // logaritmik islemler icin matematik kutuphanesi tanitilmasi
#fuses HS
#use delay(clock=4000000)    // osilator frekansi 4Mhz ayarlanmasi

#use fast_io(a)             // A ve B portu icin port yonlendirme komutlari
#use fast_io(b)

#define use_portb_lcd TRUE      // Pportuna lcd tanimlandi
#include <lcd.c>                // lcd dosyası tanitildi

  int i = 0, r1=250;
  unsigned long int vo_okunan = 0, io_okunan = 0 ;    // analog deger okumak icin degisken tnaimlama
  float vo = 0, io = 0;

#int_AD
void ADC_Kesmesi(){             // ADC kesmesi
}

void main()
```

```
{
  setup_psp(PSP_DISABLED);
  setup_spi(SPI_SS_DISABLED);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
  setup_CCP1(CCP_OFF);
  setup_CCP2(CCP_OFF);

  set_tris_a(0b00000011);      // a portu 0. pini giris olarak tanimlama
  set_tris_b(0b00000000);      // b portu pinleri cikis olarak tanimlama
  output_b(0x00);              // b portu pinleri sifir yapma
  setup_adc(adc_clock_div_32);       // ADC clock frekansi fosc/32
  setup_adc_ports(ALL_ANALOG);       // Tum ADC girisleri analog

  enable_interrupts(INT_AD);
  enable_interrupts(GLOBAL);         // Tum kesmelere izin verme

  lcd_init();                  // lcd hazir

  while(TRUE)
  {

    set_adc_channel(0);        // Cevrim icin AN0 secildi
    delay_us(20);              // bekleme

      for(i=0;i<20;i++){               // filtrelemek için 20 adımlık döngü
        vo_okunan = read_adc();        // okunan analog degeri degiskene atama
        vo += vo_okunan;               // tum okunan degerlerin bir degiskende toplanmasi
      }

    vo = vo / 20;                      // okunan degerlerin ortalamasinin alinmasi
    vo = vo * 0.00976562;      // 10 bit 1024 10V'a bölünerek okunan deger ile çarpılır ADC donusumu

    printf(lcd_putc, "\f%f V",vo);         // okunan voltaj bilgisi lcd'ye yazdirilir
    delay_us(250);

    set_adc_channel(1);        // Cevrim icin AN1 secildi
    delay_us(20);              // bekleme
```

```
      for(i=0;i<20;i++){              // filtrelemek için 20 adımlık döngü

        io_okunan = read_adc();       // okunan analog degeri degiskene atama
        io += io_okunan;              // tum okunan degerlerin bir degiskende toplanmasi
      }

    io = io / 20;                 // okunan degerlerin ortalamasinin alinmasi

    io = io * 0.00488759;         // besleme 5V oldugu icin okunan deger (5/1024) sayisina bolundu
    io = io / r1;                 // okunan voltaj degeri r1= 250 direncine bolunerek akim degeri bulunuyor
    io = io * 1000;               //akim mA cinsine cevirmek icin 1000 ile carpilir
    printf(lcd_putc, "\n%f mA" io);       // hesaplanan amper bilgisi lcd'ye yazdirilir
    delay_ms(250);            //bekleme
  }
}
```

PTC Ölçüm İçin Şematik Devre (PIC16F877 için):



Devre İle İlgili Bilgiler:

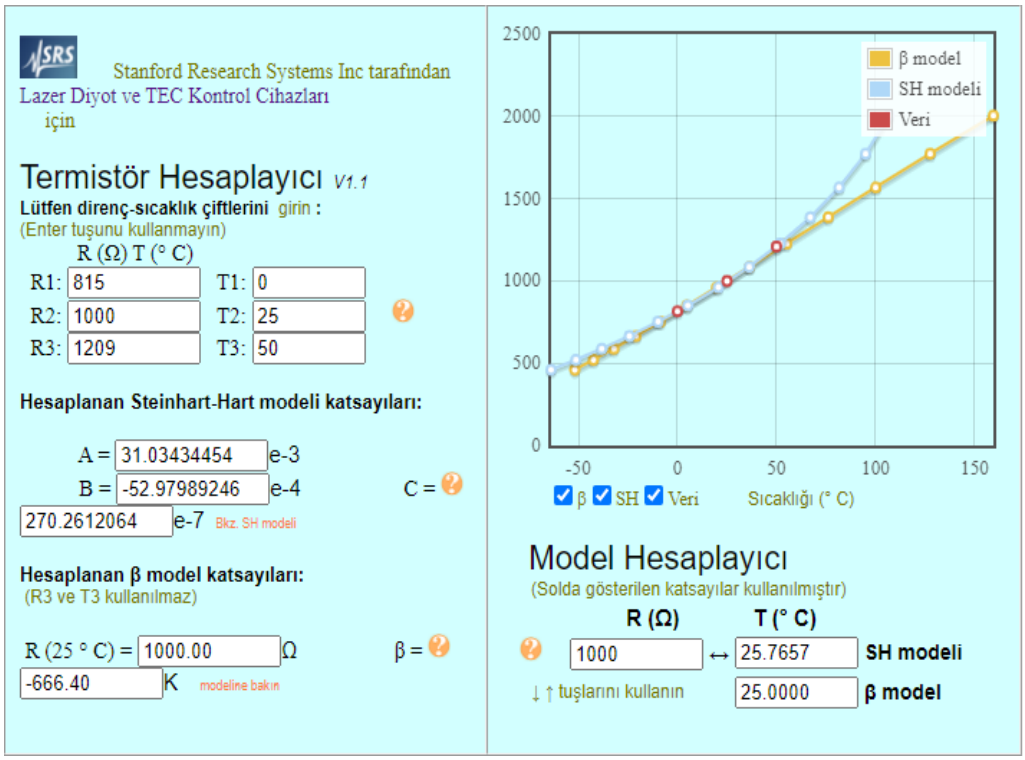
Analog ölçüm için mikrodenetleyici olarak PIC16F877 tercih edilmiş ve kodlama için CCSC programı kullanılmıştır. PIC16F877 mikrodenetleyicisi için datasheet incelenmiş ve Proteus programında pic için osilatör ve besleme gerilimi bağlantısı yapılmıştır. Analog girişlerinden A0 pinine PTC elemanı bağlanmış ve zarar görmemesi için toprak hattına direnç eklenmiştir. Analog değerin dijital değere dönüştürüldüğünü gözlemlemek için B portu çıkış portu yapılmış ve B0 pinine led bağlanmıştır. Sıcaklık 30^0C'den fazla ise led yanıyor, 30^0C'den az ise sönüyor şeklindedir.

10kΩ PTC için datasheet araştırması yapılmıştır. 10kΩ PTC için datasheet bilgisi bulunamamış ve 1kΩ PTC için ekteki datasheet ile değerler okunmuş ve Stein Hart Denklemi'ndeki değerleri hesaplamak için (https://www.thinksrs.com/downloads/programs/therm%20calc/ntccalibrator/ntccalculator.html) internet adresinde yazılarak A, B ve C değerleri bulunmuştur.

Stein Hart Denklemi;

$$\frac{1}{\mathrm{T}} = \mathrm{A} + \mathrm{B}\,(\ln R) + \mathrm{C}(\ln R)^3$$

	PTC 1K KTY81-110 1%@25°C	
T °C	R Ohm	dT ±°C
-50	515	2,9
-40	567	2,7
-30	624	2,6
-20	684	2,4
-10	747	2,1
0	815	1,9
10	886	1,7
20	961	1,4
25	**1000**	1,3
30	1040	1,4
40	1122	1,6
50	1209	1,9



Bu tabloya göre 0^0C - 25^0C - 50^0C sıcaklıkları için;

A= 31.03434454e-03

B= -52.97989246e-04

C= 270.2612064e-07 değerleri elde edilmiştir.

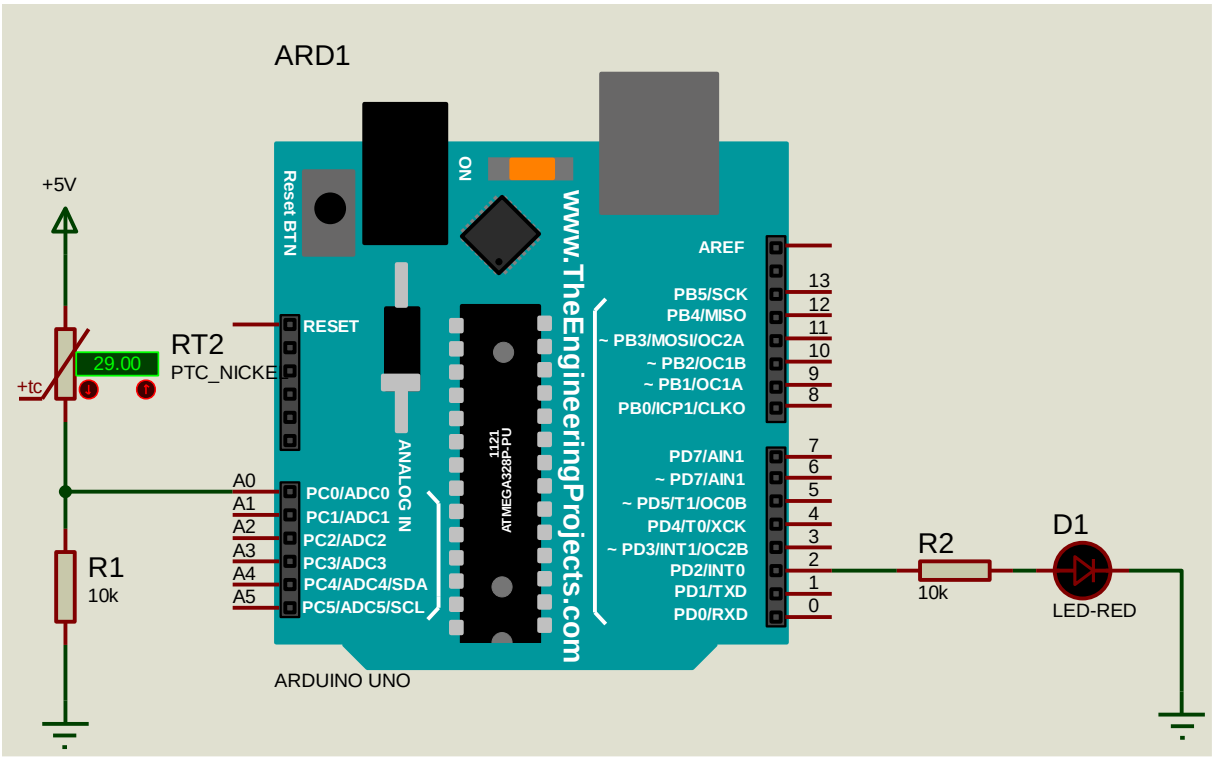
CCSC Kodları ve Açıklamaları:

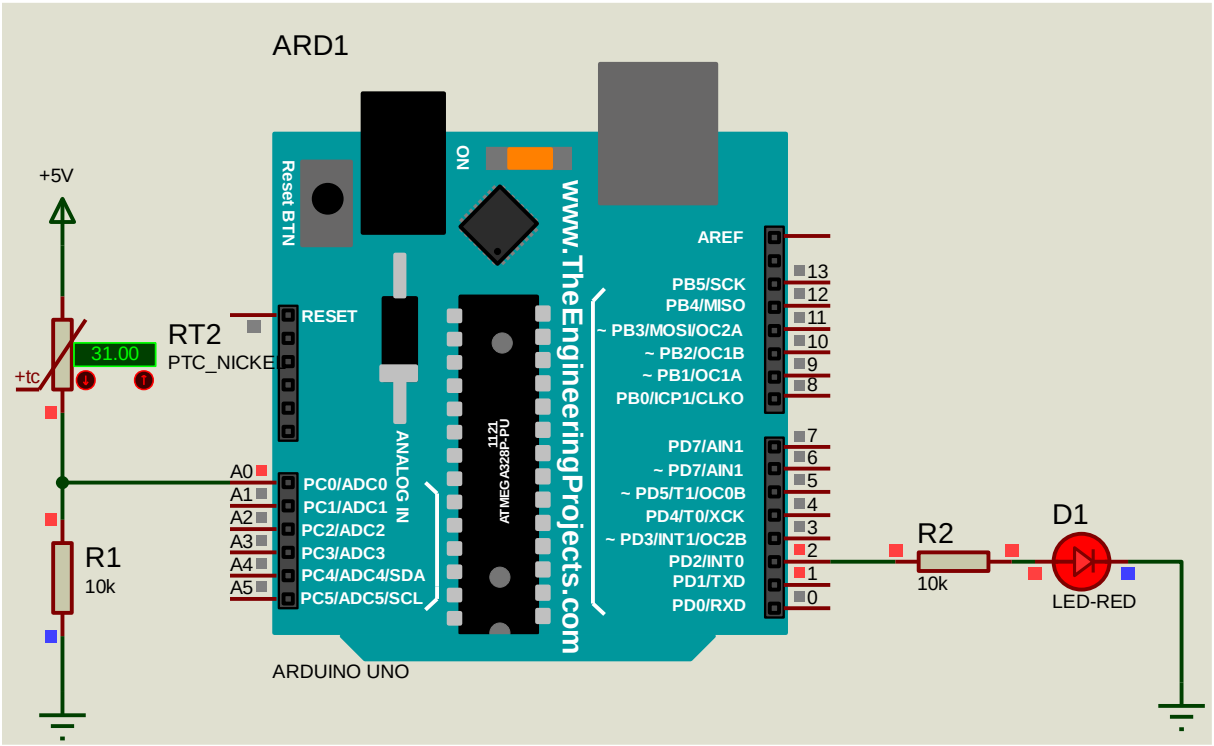
```
#include <16f877.h>            // denetleyicinin baslik dosyasi tanimlama
#fuses XT, NOWDT, NOPROTECT, NOBROwNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPD      // Denetleyici konfigürasyon ayarlari

#device ADC=10
#include <math.h>          // logaritmik islemler icin matematik kutuphanesi tanitilmasi
#fuses HS
#use delay(clock=4000000)    // osilator frekansi 4Mhz ayarlanmasi

#use fast_io(a)            // A ve B portu icin port yonlendirme komutlari
#use fast_io(b)


  unsigned long int okunan;    // analog deger okumak icin degisken tanimlama
```

```
  float sicaklik;              // asicaklik degeri okumak icin degisken tnaimlama
  float R1=10000;              // sabit direnc degeri tanÄ±mlama
  float A=31.03434454e-03, B= -52.97989246e-04, C=270.2612064e-07;       // Stein hart denklemindeki A-B-C degerleri

void main()
{
  set_tris_a(0b00000001);      // a portu 0. pini giris olarak tanimlama
  set_tris_b(0b00000000);      // b portu pinleri cikis olarak tanimlama
  output_b(0x00);              // b portu pinleri sifir yapma
  setup_adc(adc_clock_div_32);       // ADC clock frekansi fosc/32
  setup_adc_ports(ALL_ANALOG);       // Tum ADC girisleri analog

  while(TRUE)
  {
    set_adc_channel(0);        // CEvrim icin AN0 secildi
    delay_us(20);              // bekleme
    okunan=read_adc();         // okunan analog degeri degiskene atama

    sicaklik = log((10240000 / okunan) - R1);            // (8 bit (1024) / okunan deger) - direnc logaritma alinmasi
    sicaklik = 1 / (A + (B + (C * sicaklik * sicaklik)) * sicaklik);          // formulunden sicaklik degeri bulunmasi
    sicaklik = sicaklik - 273.15;                  // sicakligi santigrat dereceye cevirme

    if(sicaklik > 30){             // sicaklik 30 dereceden buyukse led yansin
      output_high(pin_b0);
    }
    else{                          // degilse led sonsun
      output_low(pin_b0);
    }

    delay_ms(100);                 // 10 ms bekleme
  }
}
```

- Proteus programında pic ile yaptığım devrede simülasyonunu gözlemleyemediğim için arduino ile tekrar devre hazırlayıp aynı kodlar ile devreyi test ettim.

Şematik Devre (Arduino için):





Arduino Kodları ve Açıklamaları:

```
#include<math.h>    // logaritmik islemler icin matematik kutuphanesi tanitilmasi
#define led 2       // 2.pinin led olarak tanimlanmasi
float R1=10000;     // sabit direnc degeri tanımlama
float A=31.03434454e-03, B= -52.97989246e-04, C=270.2612064e-07;    // Stein hart denklemindeki A-B-C degerleri
void setup() {
  Serial.begin(9600);       //haberlesme hizi
  pinMode(led,OUTPUT);      // ledin cikis olarak tanimlanmasi
}
double PTC(int okunan){     // matematiksel islemler icin fonk yazma
 double sicaklik;
 sicaklik = log(((10240000 / okunan) - R1));   // (8 bit (1024) / okunan deger) - direnc logaritma alinmasi
 sicaklik = 1 / (A + (B + (C * sicaklik * sicaklik)) * sicaklik);   // formulunden sicaklik degeri bulunmasi
 sicaklik = sicaklik - 273.15;          // sicakligi santigrat dereceye cevirme
 return sicaklik;
}
void loop() {
  int deger = analogRead(A0);       // A0 pinindeki analog degeri okuma
  double sicaklik = PTC(deger);     // PTC fonksiyonuna gitme
  if(sicaklik > 30){        // sicaklik 30 dereceden buyukse led yansin
    digitalWrite(led,HIGH);
  }
  else{                     // degilse led sonsun
    digitalWrite(led,LOW);
  }
  delay(250);               //0.25 sn bekleme
}
```