

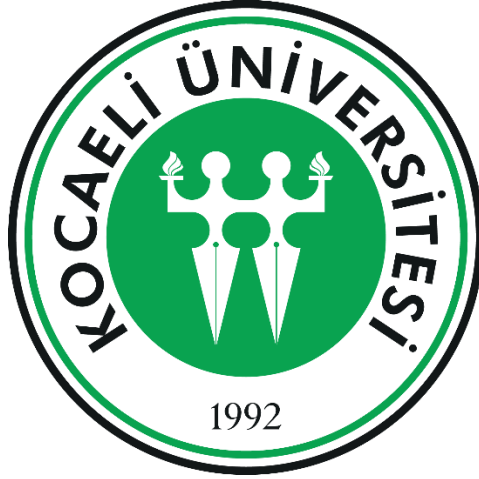
# KOCAELİ ÜNİVERSİTESİ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

MİKROİŞLEMCİLER DERSİ

PROJE ÖDEVİ

KONU: ORTAM SICAKLIĞINA GÖRE RGB ŞERİT LED ÜZERİNDE  
FARKLI RENK VE ANİMASYON YAPILMASI.



HAZIRLAYANIN

ADI SOYADI : KAAN GÜLER

NUMARASI : 190208045

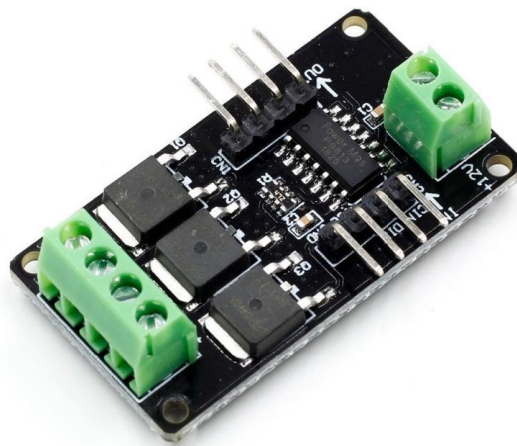
## İçindekiler

GİRİŞ .....	3
Kullanılacak Malzemeler .....	5
Projenin Yapılışı .....	6
1.)Kodlama Aşaması .....	6
2.) PCB Aşaması.....	9
Projenin Bitmiş Hali .....	11

## GİRİŞ

Projenin genel yapısı kısaca řu řekildedir, LM35 sıcaklık sensörünün Vout bacağından MSP430G2553 mikroişlemcisine gelen Analog veriyi mikroişlemcimizin içindeki ADC(Analog to Digital Converter) yardımı ile okuyup, LM35 in datasheetindeki özelliklerine dikkat edilerek derece cinsinden sıcaklık değeri ni elde edip, bu sıcaklık değeri ne göre mikroişlemcimizin içindeki Timer yardımı ile RGB ledimizin red, green ve blue kanallarına gidecek olan ve bu ledlerin parlaklıklarını farklı řekilde ayarlayıp tüm ara renkleri elde etmemize yardımcı olacak olan PWM sinyali ni üreteceğiz.

**Not:** Projenin pratik kısmında kod olarak herhangi bir fark yaratmayacağından dolayı ben tekli olan rgb lerden bir modül oluşturdum ve bu modül ile projemi test ettim, ama eğer bir řerit led kullanılmak istenirse koymuş olduğum rgb soketine bir strip led driver modülü bağlanarak aynı řekilde řerit led te çalışabilir.(Bu konu hakkında Abdulrezzak hocaya mail atıp izin aldım, mail bir sonraki sayfada.)



Örnek rgb strip led driver



## Re: mikroişlemci proje



[abdulrezzak.duran@kocaeli.edu.tr](mailto:abdulrezzak.duran@kocaeli.edu.tr) (9 Mayıs 2021 21:43)



Kime: [190208045@kocaeli.edu.tr](mailto:190208045@kocaeli.edu.tr)

Olur

9 Mayıs 2021 18:46, [190208045@kocaeli.edu.tr](mailto:190208045@kocaeli.edu.tr) yazdı:

Hocam merhabalar, benim mikroişlemcilerdeki projem "Ortam sıcaklığına göre RGB şerit LED üzerinde farklı renk ve animasyon yapılması." bunun hakkında bir sorum vardı ve ayhan hocaya mail atmıştım ama görmedi sanırım o yüzden size sormak istedim. Bu projede şerit led olması zorunlu mu ? çünkü araştırdığım kadarıyla ARGB dışındaki RGB şerit ledlerde tüm ledler aynı anda kontrol ediliyor tekli rgb ledlerden tek farkı şerit ledler 12 volt istiyor. Sormak istediğim şey şerit ledlerin fiyatı tekli ledlere göre çok fazla olduğu için projemde şerit led yerine tekli satılan rgb ledlerden kullanabilir miyim ?

## **Kullanılacak Malzemeler**

- 1 adet MSP430g2553
- 1 adet 4 pin diři header
- 1 adet LM317T voltaj regülatörü
- 1 adet LM35DZ sıcaklık sensörü
- 1 adet LM7805 voltaj regülatörü
- 1 adet 220 uF kondansatör
- 2 adet 1 uF kondansatör
- 1 adet 10 uF kondansatör
- 1 adet 240 ohm direnç
- 1 adet 390 ohm direnç
- 1 adet 47k ohm direnç
- 1 adet anahtar
- Şerit led kullanılacaksa: Rgb şerit led ve rgb şerit led sürücü
- Tekli led kullanılacaksa: Rgb led modülü(üstünde direnci olan)

# Projenin Yapılışı

## 1.)Kodlama Aşaması

Kodlama işlemi IAR idesi kullanılarak yapıldı, kodlar ve açıklamaları aşağıdaki gibidir;

```
#include <msp430g2553.h> //msp430g2553 kütüphanesi aktif edildi.
```

```
void pwmmayar();
void pwm_kirmizi(int value);
void pwm_yesil(int value);
void pwm_mavi(int value);
void setrgb(double rv,double gv,double bv);
void sicaklikrenksec();
unsigned long sicaklik = 0;
unsigned long adcdeger = 0;
unsigned long sicaklikl = 0;
void adc();
double rv1 = 0;
double gv1 = 0;
double bv1 = 0;
double rv2 = 0;
double gv2 = 0;
double bv2 = 0;
double rv3 = 0;
double gv3 = 0;
double bv3 = 0;
int d = 0;
```

```
void main()
{
    WDTCTL = WDTPW + WDTHOLD; //watchdog time kapatildi
    BCSCTL1 = CALBC1_1MHZ; // islemcinin çalışma frekansi olarak 1MHz seçildi.
    DCOCTL = CALDCO_1MHZ;

    P2DIR |= (BIT1 | BIT5); //BIT1 ve BIT5 degerlerini P2DIR ile cikis
    P2SEL |= (BIT1 | BIT5); // olarak tanimliyorum ve pwm sinyali için de P2SEL ile tanimliyorum.

    P1DIR |= BIT6; // BIT6 degerini P1DIR ile cikis olarak tanimliyorum
    P1SEL |= BIT6; //ve pwm sinyali için de P1SEL ile tanimliyorum.
    adc(); // ADC modülünün ayarlarinin olduğu fonksiyonu çağırdım.
    pwmmayar(); // PWM sinyali üretilecek timerin ayarlarinin olduğu fonksiyonu çağırdım.

    __bis_SR_register(GIE); // interruptlari aktif ettim.
    while(1){ // while(1) ile fonksiyon sonsuz döngüye alindi yani sürekli sicaklik degerleri ve renkler güncellenecek.
        ADC10CTL0 |= ENC + ADC10SC; // ENC biti ile ADC tetiklendi.
        sicaklikrenksec(); // Sicaklik degerleri kullanarak rgb renklerin üretilecegi fonksiyon çağırildi.
    }
}
```

```

void adc() {
    ADC10CTL0 &= ~ENC; // ADC ayarlari yapilmaya baslanmadan ENC biti 0 landi.
    ADC10CTL1 = INCH_5 + ADC10SSEL_3 + ADC10DIV_4; //INCH_5 komutu ile adc için 5. kanali seçtim,
                                                    //ADC10SSEL_3 ile clock olarak SMCLK seçtim(çalışma frekansi 1MHz),
    ADC10CTL0 = ADC10SHT_1 + ADC10ON + ADC10IE; //ADC10DIV_4 ile de adc'nin deger olusturma frekansi olarak
                                                    //çalışma frekansinin 4 te biri yani 250KHz ayarladim.
    ADC10AE0 = BIT5; //ADC10ON ile adc aktif hale getirdim, ADC10IE ile interruptlari aktif hale getirdim,
                    //ADC10SHT_1 ile de adc örneğinin kaç adc clock darbesinde alınacağı modunu
                    //01 modu olan 8 x ADC10CLKs olarak ayarladim.
                    // ADC girişi olarak BIT5(1.5) pinini belirledim.
}

#pragma vector = ADC10_VECTOR // ADC interrupt fonksiyonu çağırıldı
__interrupt void ADC10_ISR (void) {
    adcdeger = ADC10MEM; //ADC10MEM registerinden okunan degerleri adcdeger ismi verdiğim degiskene aktardim.
    sicaklikl = (adcdeger*3300)/1024; // okudugumuz degerler 0-1024 arasindadir bu yüzden bu degerleri mV cinsine çevirdim.
    sicaklik = sicaklikl/10; //LM35DZ sicaklik sensöründe her 10 mV degerinde 1 derece desisim olur bu yüzden
                            //mV' a çevirdiğim degeri 10 a böldüm.
}

void pwmayar(){
    TA1CCR0 = 1000; //PWM sinyallerini üretmek için timer kullaniriz.
                  //TA1CCR0 ve TA0CCR0 registerlarimiza atadigimiz degerler ile PWM sinyalinin periyodunu,
    TA1CCTL1 = OUTMOD_7; //TA1CCR1,TA1CCR2 ve TA0CCR1 registerlarına atadigimiz degerler ile de
                        //PWM sinyalinin darbe genişliğini(pulse width) belirleriz.
    TA1CCTL2 = OUTMOD_7; //TA1CCTL1,TA1CCTL2 ve TA0CCTL1 registerlarını OUTMOD_7(reset/set) ye ayarladim,
                        //bu mod timerin TACCR1 degerine ulasana kadar output=set olmasını,
    TA1CTL = TASSEL_2 | MC_1; //TACCR1 ve TACCR0 arasında output=reset olmasını saglar
                        //ve bu reset ve set oranlarını kullanarak ledlerin parlakliklarını ayarlarız.
    TA0CCR0 = 1000; //TASSEL_2 ile clock olarak SMCLK (1MHz) seçilir , MC_1 ile de up mode(0 dan TACCR0 a kadar sayma) aktif edilir.
    TA0CCTL1 = OUTMOD_7; //TA0CCR0 ve TA1CCR0 registerlarına 1000 degeri vererek PWM sinyalinin frekansini 1KHz olarak ayarladim.

    TA0CTL |= TASSEL_2 | MC_1;
}

void sicaklikrenksec(){
    if (sicaklik <= 19){ //Burada ADC modülünden okunup derece cinsine çevrilen sicaklik degerine uygun if kosulunu
        setrgb(0,0,255); //saglayan kisim aktif olur ve o if'in içindeki setrgb fonksiyonu çalışır.
    }
    if (sicaklik > 19 && sicaklik <= 22){
        setrgb(0,127,255);
    }
    if (sicaklik > 22 && sicaklik <=25){
        setrgb(0,255,255);
    }
    if (sicaklik > 25 && sicaklik <= 28){
        setrgb(0,255,127);
    }
    if (sicaklik > 28 && sicaklik <= 31){
        setrgb(0,255,0);
    }
    if (sicaklik > 31 && sicaklik <= 34){
        setrgb(127,255,0);
    }
    if (sicaklik > 34 && sicaklik <= 37){
        setrgb(255,255,0);
    }
    if (sicaklik > 37 && sicaklik <= 40){
        setrgb(255,127,0);
    }
    if (sicaklik > 40){
        setrgb(255,0,0);
    }
}

```

```

void setrgb(double rv,double gv,double bv){
    rv2 = (rv - rv1)/ 30;           //burada animasyon olarak, yeni bir renk degeri atandigi zaman önceki
                                     //renk degeriyle arasindaki tonlar görülebilecek sekilde
    gv2 = (gv - gv1)/ 30;           //kademeli olarak geçilmesini kullandim. Bu şöyle çalışıyor
                                     //örneğin kırmızı için yeni gelen rv degerinin önceki rv degeriyle
    bv2 = (bv - bv1)/ 30;           //yani rv1 degerile farki aliniyor, sonrasında bu fark 30 a bölünüyor
                                     //ve bu deger rv2 ye kaydediliyor.

    while(d < 30){
        d++;                         //burada ise aradaki farki 30 a böldüğümüz için 30 defa çalışacak bir
                                     //while fonksiyonu olusturuluyor ve önceki renk degeri(rv1) her bir
        rv1 = rv1 + rv2;             //döngüde farkin 30 a bölümüyle(rv2) toplanip rv1 e aktariliyor bu
        gv1 = gv1 + gv2;             //sekilde en basta yollanan rv degeri ile rv1 in arasindaki 30 degerin de
        bv1 = bv1 + bv2;             //renk degeri olusturularak rgb ledde geçis animasyonu olusturulmus oluyor.

        rv3 = ((rv1/255)*1000);      //rgb renk tonlari 0-255 arasinda tanimlaniyor fakat bizim olusturdugumuz
        gv3 = ((gv1/255)*1000);      //pwm sinyali için periyodumuz 1000 bu yüzden 0-255 arasindaki degeri
        bv3 = ((bv1/255)*1000);      //255 e bölüp 1000 ile çarptigimizda o renk için gerekli olan
                                     //pwm sinyalinin degerini bulmus oluruz.

        pwm_kirmizi((int)rv3);        //bulmus oldugumuz pwm sinyal degerlerini tamsayiye çevirip,
        pwm_yesil((int)gv3);         //renkleri olusturan pwm fonksiyonlarına yollariz.
        pwm_mavi((int)bv3);
        __delay_cycles(100000);
    }
    d = 0;
}

```

```

void pwm_kirmizi(int value)          //bu 3 fonksiyonla pwm sinyalinin periyot oranlarini
{                                     //belirleyen degerleri belirleyerek rgb renklerimiz elde ederiz.
    TA1CCR1 = value;                //kırmızı led (2.1)
}

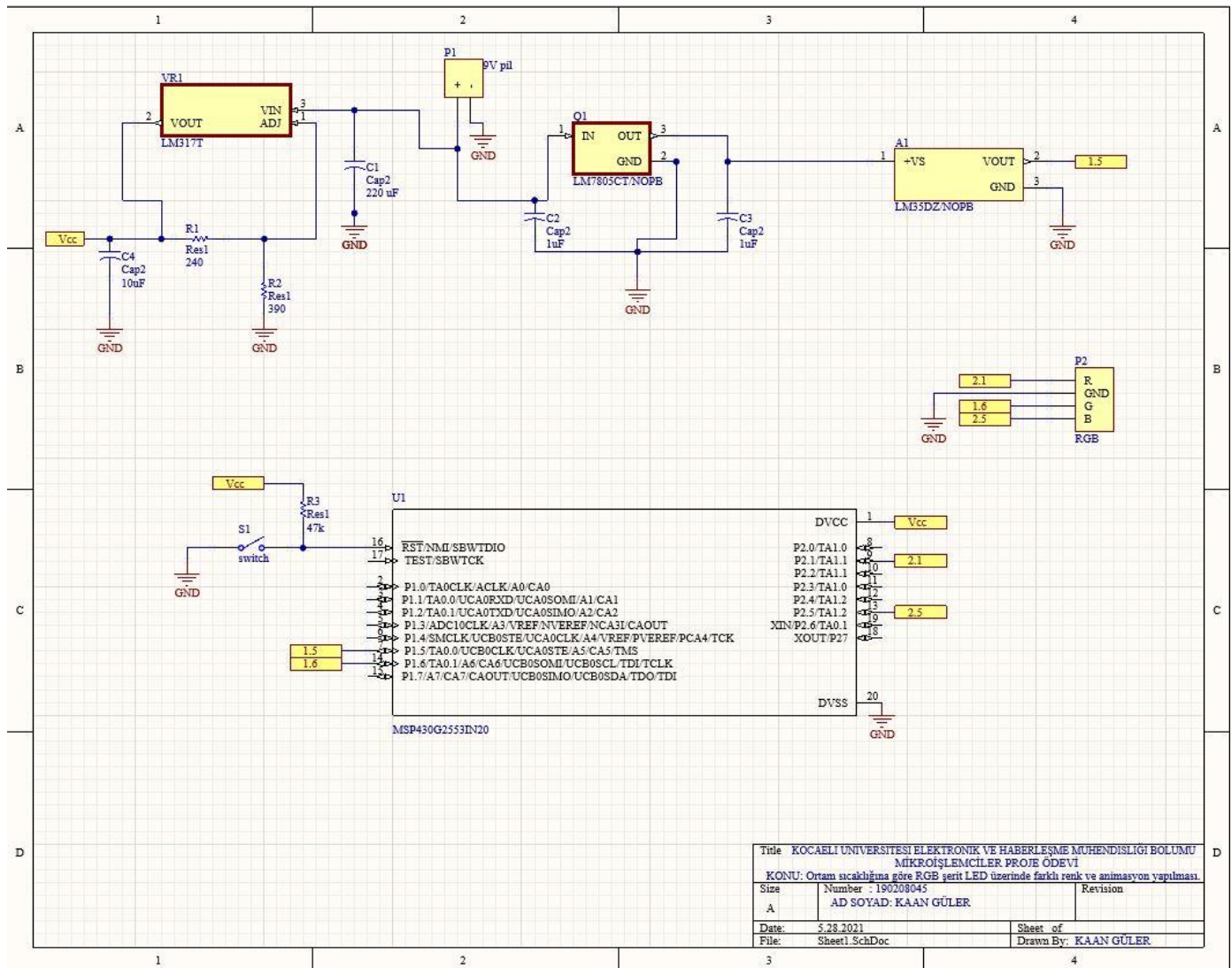
void pwm_yesil(int value){
    TA0CCR1 = value;                //yesil led (1.6)
}

void pwm_mavi(int value){
    TA1CCR2 = value;                //mavi led (2.5)
}

```



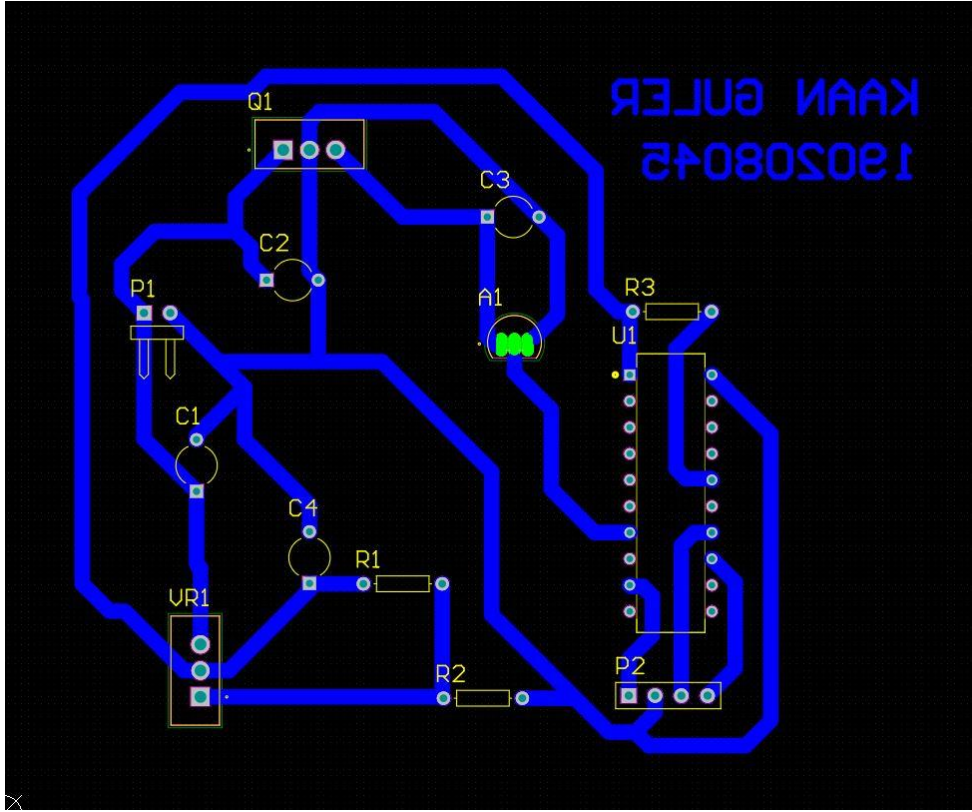
**PCB yi tasarlarken Altium programını kullandım. Şematik aşağıdaki gibi;**



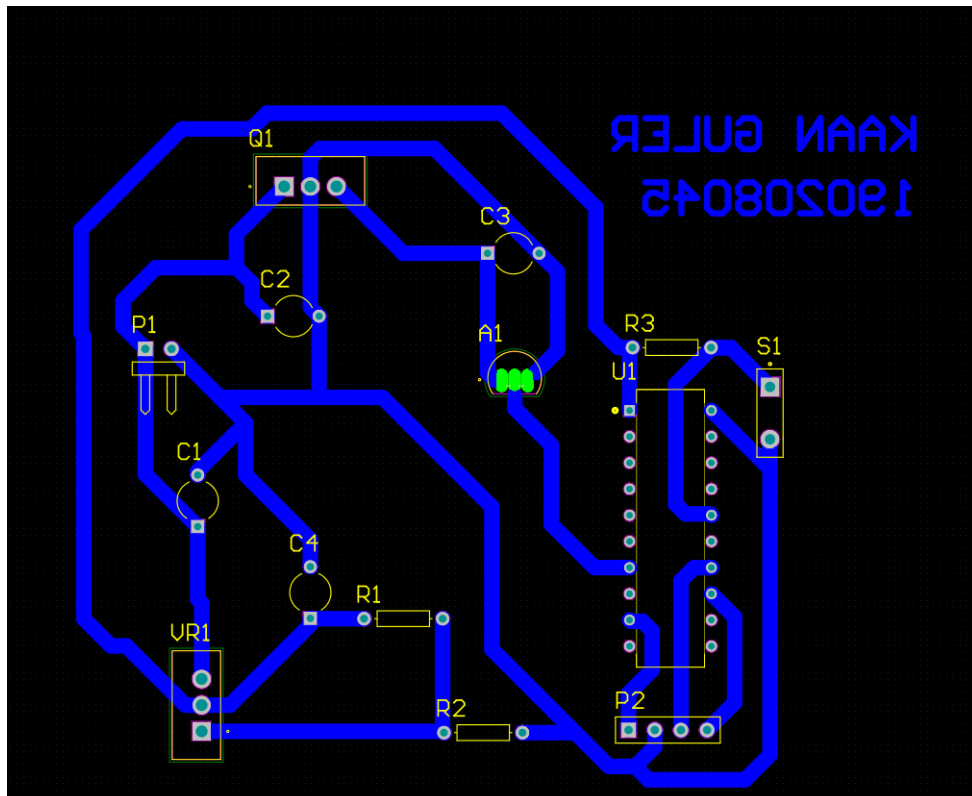
Şematikte üst kısımda LM317 ve LM7805 voltaj regülatörleri kullanarak girişte verdiğimiz 9V değeri MSP430G2553 ün vcc ayağına 3.3V, LM35DZ nin Vin ayağına ise 5V olacak şekilde ayarlanmıştır.

**Alt kısımda ise msp430g2553 ün pinleri bağlanmıştır.**

**PCB görüntüsü aşağıdakiler gibidir;**



**PCB yi normalde bu şekilde basarak projemi yapmışım fakat ayrıyeten resetleme özelliği istendiği için şematikteki gibi RST pinini grounda bir anahtar kullanarak bağladım, yani son hali aşağıdaki şekilde oldu.**



## Projenin Bitmiř Hali

