

# ARDUINO

## TEMEL ATÖLYE UYGULAMALARI

2015



**İÇİNDEKİLER**

:

LED UYGULAMALARI.....	5
TRAFİK LAMBASI.....	9
TAKİPÇİ 10 LED.....	11
RGB LED UYGULAMALARI.....	13
POTANSİYOMETREDEN DEĞER OKUMAK.....	17
NTC UYGULAMALARI.....	19
LDR UYGULAMALARI.....	21
DC MOTOR UYGULAMALARI.....	25
STEP MOTOR UYGULAMALARI.....	29
SERVO MOTOR UYGULAMALARI.....	31
RTC UYGULAMALARI.....	33
LCD VE TFT EKRAN UYGULAMALARI.....	35
YANGIN ALARMI UYGULAMASI.....	41
SU SEVİYE ÖLÇÜMÜ UYGULAMASI.....	43
SERIAL.AVAILABLE ( ) FONKSİYONU UYGULAMASI.....	45
LCD GÖSTERGELİ TERMOMETRE UYGULAMASI.....	49



# LED UYGULAMALARI

## 1

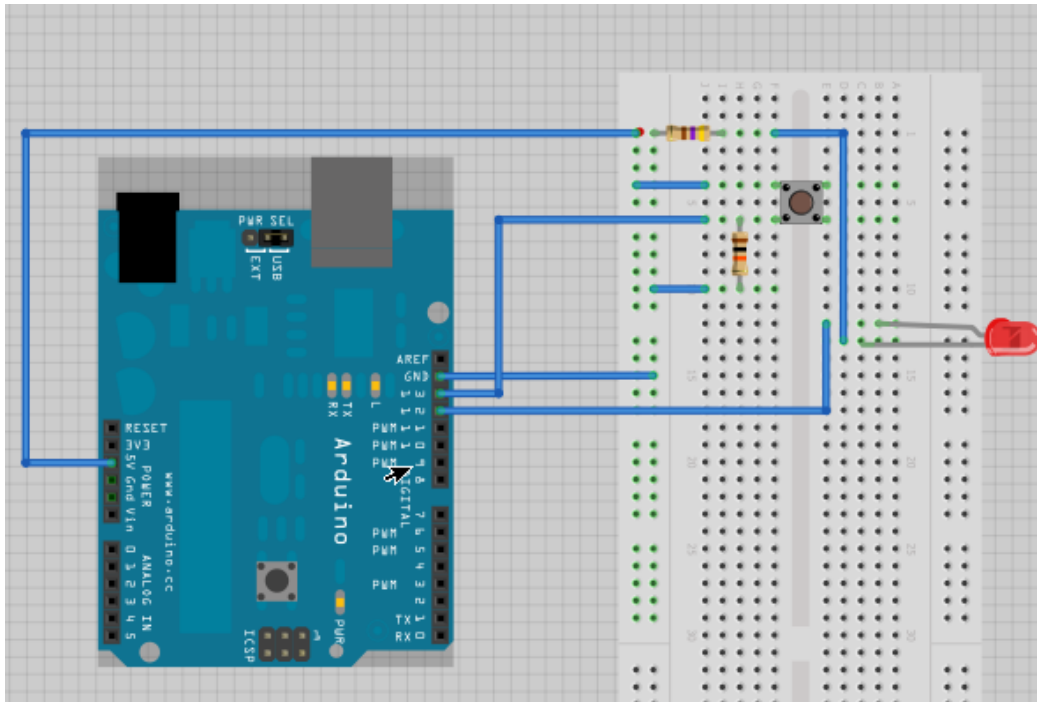
### 1.1- BUTON İLE LED YAKMA

**Açıklama:** Arduino nun üzerinde ki 5V ve gnd uçlarını borda bağlayınız. Led in anot ucunu arduino nun 12 nolu pinine, katot ucunu 470  $\Omega$  direnç ile gnd ye bağlanır.

#### Devrede Kullanılan Malzemeler:

- 1 x 470  $\Omega$
- 1 x 10 K $\Omega$
- 1 x çift çift buton
- 1 x led
- 1 x Arduino UNO

#### Devre Şeması:



#### Devrenin Yazılımı:

```
// pin numaralarını sabit olarak tanımlıyoruz
const int butonPin = 13;      // butonun bağlı olduğu pin numarası
const int ledPin = 12;      // LED'in bağlı olduğu pin numarası
```

```
int buttonDurumu = 0;           // Buton girişinin durumu (HIGH veya LOW)
void setup() {

    // Buton pinini giriş olarak ayarlıyoruz:
    pinMode(buttonPin, INPUT);
    // LED pinini çıkış olarak ayarlıyoruz

    pinMode(ledPin, OUTPUT);
}

void loop() {

    // Buton durumunu oku

    buttonDurumu = digitalRead(buttonPin);

    /* Butona basıldığında butonun durumu HIGH olacaktır. Bu durumda
    LED çıkışını HIGH yapıyoruz. Ters durumda ise LOW yapıyoruz
    */

    if (buttonDurumu == HIGH) {
        digitalWrite(ledPin, HIGH);
    }
    else {
        digitalWrite(ledPin, LOW);
    }
}
```

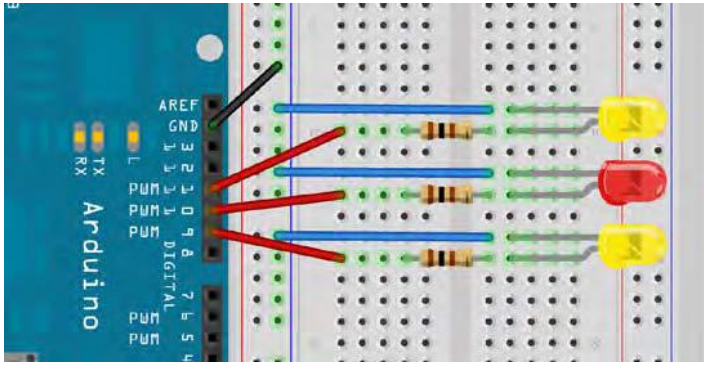
## 1.2 - ALEV EFEKTİ

**Açıklama:** PWM aracılığıyla, LED'lere titreşen rastgele bir ışık etkisi vereceğiz. Örneğin evinizde ki bir dekoratif şöminede, filmlerde, sahne oyunlarında, model demiryolları vb. gibi yerlerde bu uygulamayı kullanabilirsiniz.

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X Kırmızı LED
- 2 X Sarı LED
- 3 X 220  $\Omega$  Direnç

### Devre Şeması:



### Devrenin Yazılımı:

```
// LED'ler Arduino' nun PWM çıkışı veren uçlarına bağlanır
int ledPin1 = 9;
int ledPin2 = 10;
int ledPin3 = 11;
void setup()
{
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
}
void loop()
{
  /* Ana program döngüsü içerisinde LED'lere 0 ile 120 arasında rastgele bir
  değer gönderilir; LED parlaklığını tam olarak almak için 135 değeri eklenir
  */
  analogWrite(ledPin1, random(120)+135);
  analogWrite(ledPin2, random(120)+135);
  analogWrite(ledPin3, random(120)+135);
  delay(random(100));
}
```

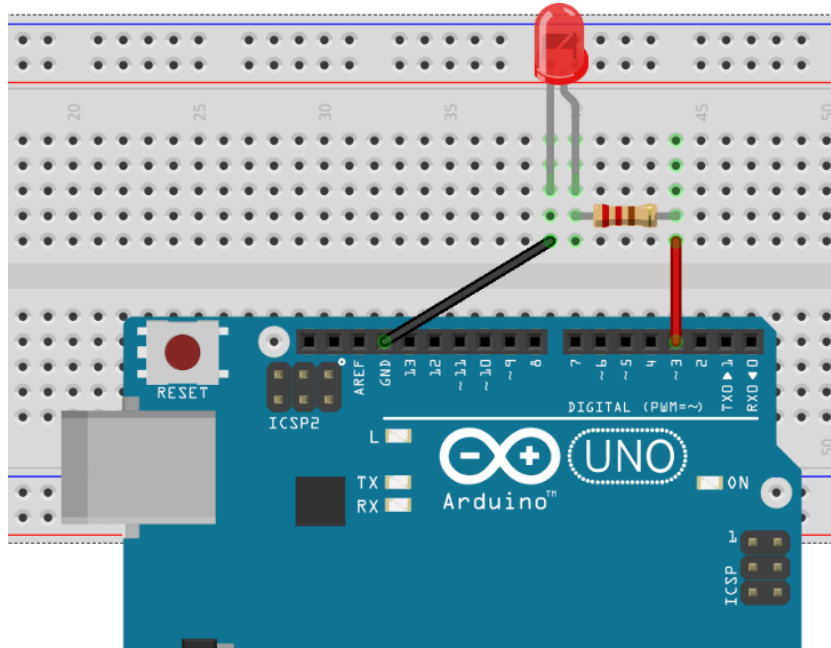
### 1.3- PWM İle Işık Şiddeti Kontrolü

**Açıklama:** Led in bağlanacağı pin 3 nolu pwm pinidir. Loop metodu içerisinde for döngüsüyle i sayaç değeri 0 dan başlatılarak 255 e kadar saydırılıyor. Led en az ışık düzeyinden en yüksek düzeye çıkarak bu işlemi devam ettirir. 10 ms lik gecikme süresini değiştirerek gözlemlenebilirsiniz.

#### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X led
- 1 x 220  $\Omega$

#### Devre Şeması:



#### Devrenin Yazılımı:

```
const int led=3; // 3 nolu pwm çıkışına led bağlanacak.
void setup()
{
  pinMode(led,OUTPUT); //3 nolu pin çıkış olarak ayarlandı.
}
void loop()
{
  for (int i=0;i<=255;i++)
  {
    analogWrite(led,i); //pwm değeri sayaç değişkeni ile belirleniyor.
    delay(10); //10 ms lik gecikme sağlanıyor
  }
}
```

Öğrencinin	DEĞERLENDİRME				TOPLAM	
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	RAKAM	YAZI
Adı :						
Soyadı :						
Nu :						
Sınıfı :						
	Ders Öğretmeni:				İmza:	



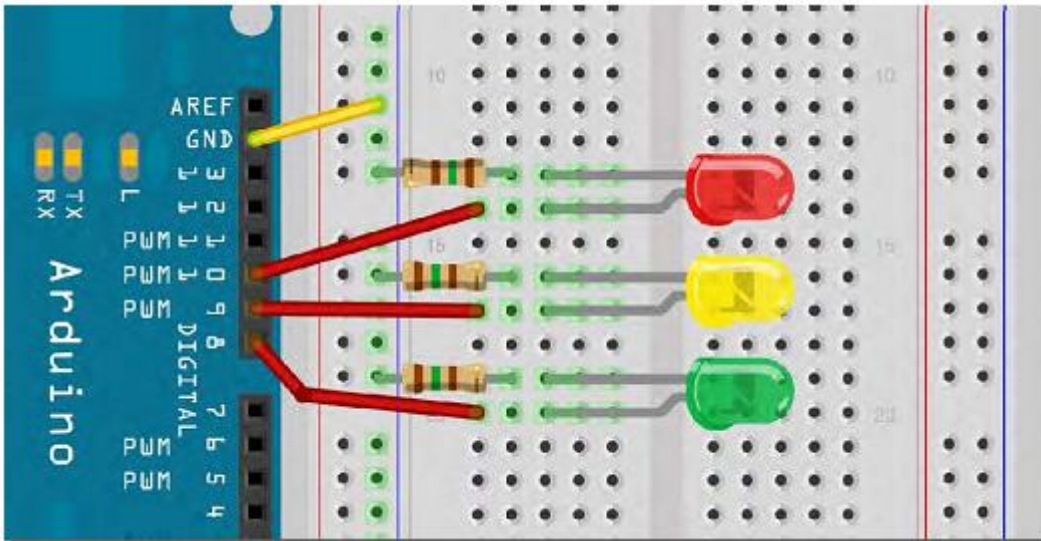
# TRAFİK LAMBASI

## 2

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 3 X led (Kırmızı, Sarı, Yeşil)
- 3 x 150  $\Omega$

### Devre Şeması:



### Devrenin Yazılımı:

```
int ledDelay = 10000; // geçişler arasındaki gecikmeye değer atıyoruz
int kirmiziPin = 10;
int sariPin = 9;
int yesilPin = 8;
```

```
// ledleri çıkış olarak tanımlıyoruz
void setup() {
  pinMode(kirmiziPin, OUTPUT);
  pinMode(sariPin, OUTPUT);
  pinMode(yesilPin, OUTPUT);
}
```

```

void loop() {
digitalWrite(kirmiziPin, HIGH); // öncelikle kırmızı led yanıyor
delay(ledDelay); // 10 saniye bekler

digitalWrite(sariPin, HIGH); // sarı led yanar
delay(2000); // 2 saniye bekler

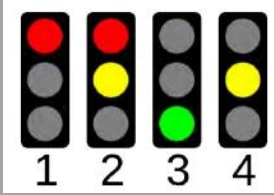
// şimdi yeşil ledi yakarak kırmızı ve sarı ledi söndürüyoruz
digitalWrite(yesilPin, HIGH);
digitalWrite(kirmiziPin, LOW);
digitalWrite(sariPin, LOW);
delay(ledDelay);

// sarı ledi yakıp yeşil ledi söndürüyoruz
digitalWrite(sariPin, HIGH);
digitalWrite(yesilPin, LOW);
delay(2000);

digitalWrite(sariPin, LOW); // sarı led söner ve döngü baştan başlar
}

```

**Açıklama:** Sarı ledin yanıp sönmeye süresi 2 saniyedir. Led sürelerini istediğiniz gibi “delay(.....)” komutu içerisinde yazarak değiştirebilirsiniz. Ledler aşağıdaki sırada çalışır.



### Notlar:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Öğrencinin	DEĞERLENDİRME				TOPLAM	
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	RAKAM	YAZI
Adı :						
Soyadı :						
Nu :						
Sınıfı :						
Ders Öğretmeni:					İmza:	

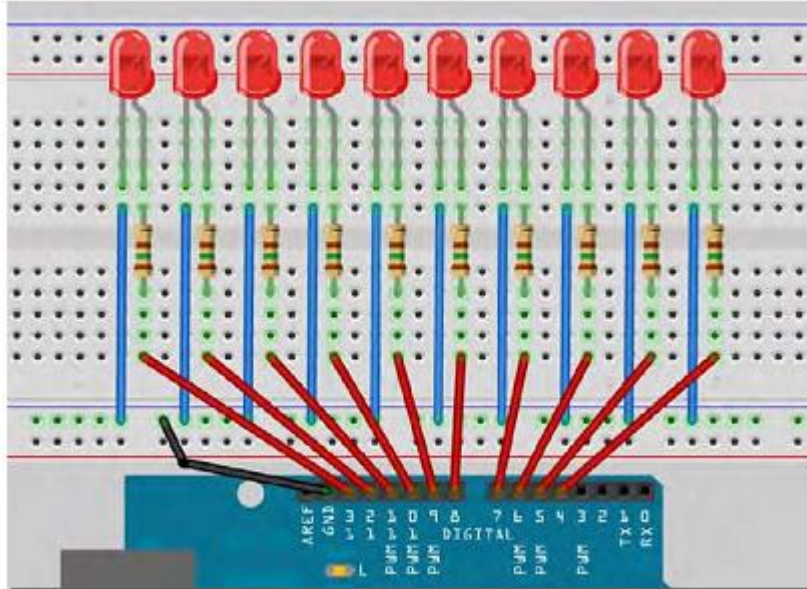
# TAKİPÇİ 10 LED

# 3

## Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 10 X led
- 10 X 150  $\Omega$

## Devre Şeması:



## Devrenin Yazılımı:

```
byte ledPin[] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13}; /* LED pinleri için dizi oluşturuyoruz*/
```

```
int ledDelay(65); // değişiklikler arasındaki gecikmeyi belirliyoruz  
int yon = 1;  
int currentLED = 0;  
unsigned long degismeZamani;
```

```
void setup() {  
  for (int x=0; x<10; x++) { // bütün pinleri çıkış olarak tanımlıyoruz  
    pinMode(ledPin[x], OUTPUT); }  
  degismeZamani = millis();  
}
```

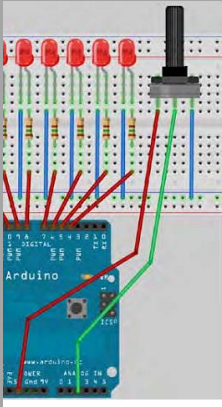
```

void loop() {
  if ((millis() - degismeZamani) > ledDelay) {
    changeLED();
    degismeZamani = millis();
  }
}

void changeLED() {
  for (int x=0; x<10; x++) { // tüm ledler sönmük
    digitalWrite(ledPin[x], LOW);
  }
  digitalWrite(ledPin[currentLED], HIGH);
  currentLED += yon; // yön değerinde arttırıyoruz
  if (currentLED == 9) {yon = -1;}
  if (currentLED == 0) {yon = 1;}
}

```

**Açıklama:** Bu uygulamada ledler birbirini sırasıyla takip etmektedir. “currentLED” 9. ve 0. Bite bağlı ledin lojik 1 durumuna göre yön değiştirmektedir. Ledlerin hızını bir potansiyometre ile kontrol etmek istiyorsanız aşağıdaki bağlantıyı yaparak verilen kodları ekleyin.



```

byte ledPin[] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
int ledDelay;
int potPin = 2;
.
.
void loop() {
  ledDelay = analogRead(potPin);
  if ((millis() - degismeZamani) > ledDelay) {

```

#### Notlar:

.....

.....

.....

.....

.....

.....

.....

Öğrencinin	DEĞERLENDİRME				TOPLAM	
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	RAKAM	YAZI
Adı :						
Soyadı :						
Nu :						
Sınıfı :						
Ders Öğretmeni:					İmza:	

# RGB LED UYGULAMALARI

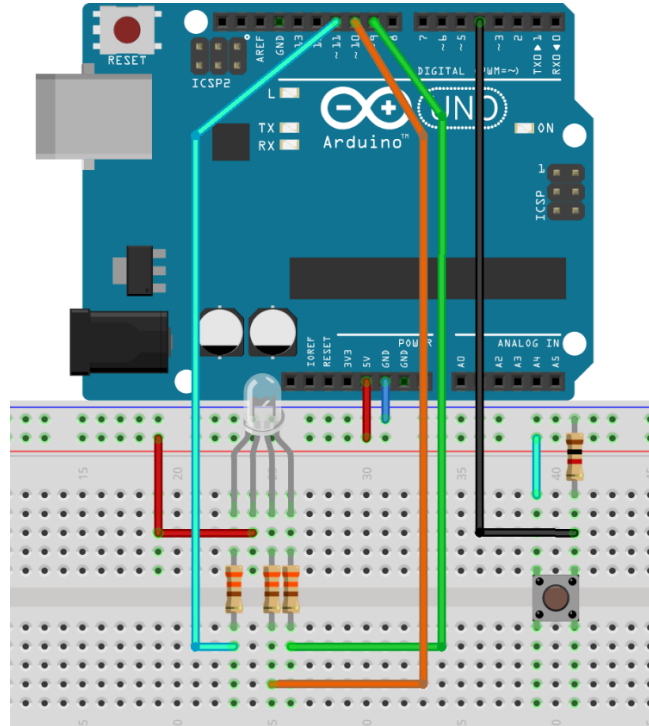
## 4

### Devrede Kullanılan Malzemeler:

- 1 X Arduino uno
- 3 X 330  $\Omega$
- 1 X 1 K $\Omega$
- 1 X RGB led
- 1 X çift çift buton

**Açıklama:** Kullandığımız RGB LED anotları ortak bağlanmış 3 LED' in tek bir kılıf içerisinde yerleştirilmiş halidir. Üç rengin ışık şiddeti dijital olarak kontrol etmemiz mümkündür. Ayrıca PWM tekniği kullanılarak istenilen renkler elde edilebilir.

### Devre Şeması:



### Devrenin Yazılımı - 1:

*/\*İlk olarak yukardaki devreyi kuracağız ve RGB LED' in her rengini 1 saniye aralıklarla yakıp söndüreceğiz. Eğer beyaz ışık görüntülemek istiyorsak tüm LED' leri açmamız gerekir.\*/\**

```
const int MaviLed=11; //mavi ledi 11 nolu pine bağlıyoruz
```

```
const int YesilLed=10; //yeşil ledi 10 nolu pine bağlıyoruz
const int Kirmiziled=9; //kırmızı ledi 9 nolu pine bağlıyoruz

// LED' lerin bağlı olduğu pinleri çıkış olarak tanımlıyoruz ;
void setup()
{
  pinMode(MaviLed,OUTPUT);
  pinMode(YesilLed,OUTPUT);
  pinMode(Kirmiziled,OUTPUT);
}

// Döngüyü başlatıyoruz
void loop()
{
  //Kırmızı rengi görüntülüyoruz
  digitalWrite(MaviLed,HIGH);
  digitalWrite(YesilLed,HIGH);
  digitalWrite(Kirmiziled,LOW);
  delay(1000);

  //Yeşil rengi görüntülüyoruz
  digitalWrite(MaviLed,HIGH);
  digitalWrite(YesilLed,LOW);
  digitalWrite(Kirmiziled,HIGH);
  delay(1000);

  //Mavi rengi görüntülüyoruz
  digitalWrite(MaviLed,LOW);
  digitalWrite(YesilLed,HIGH);
  digitalWrite(Kirmiziled,HIGH);
  delay(1000);

  //Tüm ledleri aktif hale getirerek beyaz rengi görüntülüyoruz.
  digitalWrite(MaviLed,LOW);
  digitalWrite(YesilLed,LOW);
  digitalWrite(Kirmiziled,LOW);
  delay(1000);
}
```

**Devrenin Yazılımı – 2:**

*/\*Bu yazılımda her butona basılmasıyla pwm tekniğini kullanarak ara renklerde elde edeceğiz.\*/*

```
const int BlueLed=11;
const int GreenLed=10;
const int RedLed=9;
const int buton=4;
boolean oncekiBtnDurum=LOW;
boolean simdikiBtnDurum=LOW;
int ledModu=0;

void setup()
{
  pinMode(BlueLed,OUTPUT);
  pinMode(GreenLed,OUTPUT);
  pinMode(RedLed,OUTPUT);
  pinMode(buton,INPUT);
}

boolean butonKontrol(boolean durum)
{
  boolean simdikiBtn = digitalRead(buton);
  if(durum!=simdikiBtn)
  {
    delay(5);
    simdikiBtn=digitalRead(buton);
  }
  return simdikiBtn;
}

void rgbAyarla(int renk)
{
  //kırmızı
  if(renk==1)
  {
    digitalWrite(BlueLed,HIGH);
    digitalWrite(GreenLed,HIGH);
    digitalWrite(RedLed,LOW);
  }
  else if(renk==2)
  {
    digitalWrite(BlueLed,LOW);
    digitalWrite(GreenLed,HIGH);
    digitalWrite(RedLed,HIGH);
  }
  else if(renk==3)
  {
    digitalWrite(BlueLed,HIGH);
    digitalWrite(GreenLed,LOW);
    digitalWrite(RedLed,HIGH);
  }
  else if(renk==4)
  {
```

```

analogWrite(BlueLed,127);
analogWrite(GreenLed,0);
analogWrite(RedLed,127);
}
else if(renk==5)
{
analogWrite(BlueLed,127);
analogWrite(GreenLed,127);
analogWrite(RedLed,0);
}
else if(renk==6)
{
analogWrite(BlueLed,0);
analogWrite(GreenLed,127);
analogWrite(RedLed,127);
}
}
}
void loop()
{
  simdikiBtnDurum=butonKontrol(oncekiBtnDurum);
  if(oncekiBtnDurum==LOW && simdikiBtnDurum==HIGH)
  {
    ledModu++;
  }
  oncekiBtnDurum=simdikiBtnDurum;
  if(ledModu==7)
    ledModu=0;

  rgbAyarla(ledModu);

}

```

**Notlar:**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Öğrencinin	DEĞERLENDİRME				TOPLAM	
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	RAKAM	YAZI
Adı :						
Soyadı :						
Nu :						
Sınıfı :						
Ders Öğretmeni:					İmza:	



# POTANSİYOMETREDEN DEĞER OKUMAK

## 5

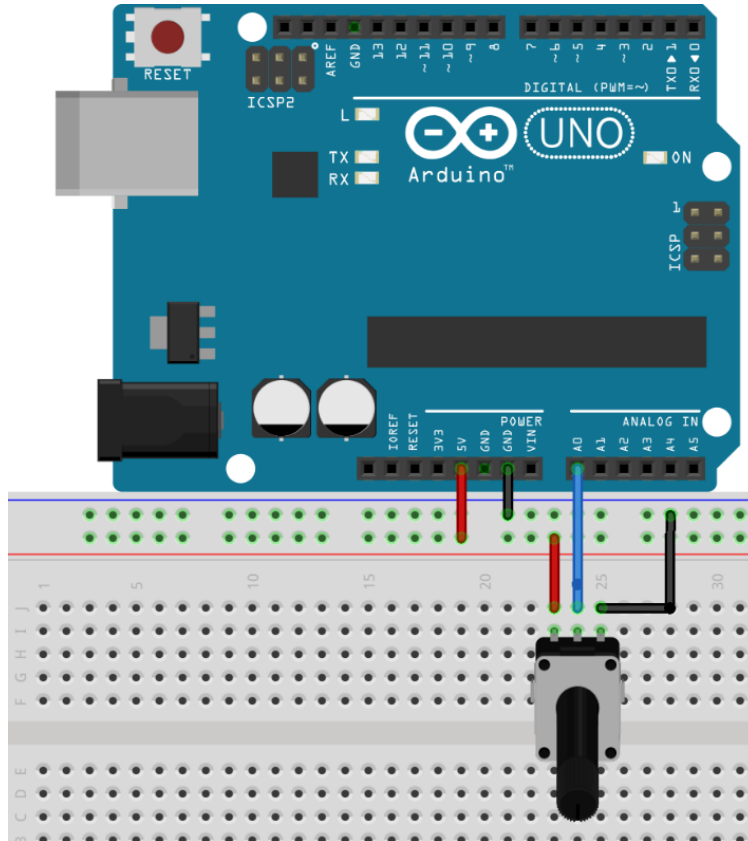
### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X 100 KΩ POT

**Açıklama:** Potansiyometre değişen direnç değeri ile bu uygulamada gerilim bölücü olarak çalışır. Ayar ucundan alınan analog değerleri dijitale çevirmek için `analogRead()` komutunu kullanacağız. Potun verdiği her değeri PC ortamına aktarıp serial monitörde gözlemleyeceğiz.

Okunan değer 10 bit olduğu için  $2^{10} = 1024$  farklı değer alır. Giriş voltajı 5V olduğunda üretilen dijital verinin değeri 1023 olur. Her 1 saniyede okunan değer PC'ye gönderilir. Bu değerleri görüntülemek için Arduino IDE programındaki [Seri Port Ekranı](#) butonuna tıklamanız yeterlidir.

### Devre Şeması:





# NTC

## UYGULAMALARI

# 6

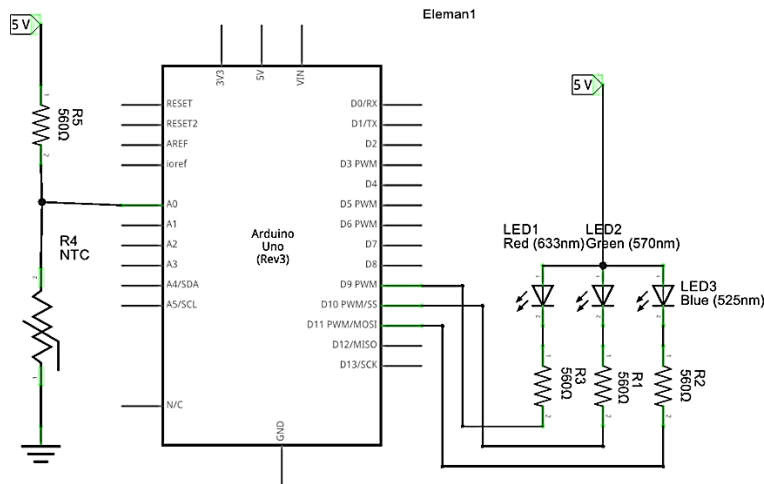
### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X NTC
- 3 X LED ( KIRMIZI, YEŞİL, MAVİ )
- 4 X 560  $\Omega$

**Açıklama:** NTC' ler ( **N**egative **T**emperature **C**oefficient) sıcaklığı arttıkça direnci azalan devre elemanlarıdır. Ntc' den okumak istediğimiz analog gerilimi alabilmemiz için, değişen direnç değerini gerilim bölücü direnç sayesinde analog gerilime dönüştürmemiz gerekir.

Aşağıdaki devremizde ntc ile 560  $\Omega$  luk direncin birleştiği yerden A0 pinine bağlantı sağlanarak gerilim bölücü elde edilmiştir. Ntc' nin oda sıcaklığındaki direnci 500  $\Omega$  civarında olduğundan sıcaklık arttıkça A0 pinine giden gerilim düşecektir. Bu sayede elde ettiğimiz analog sinyalleri yazılımla yorumlayacağız.

### Devre Şeması:



### Devrenin Yazılımı:

```
int sensorSicaklik=0; /*sensörden okunan gerilim değeri bu değişkene kaydedilir*/
int kirmiziled=9; //kirmizi ledin bağlı olduğu pin
int yesilled=10; //yeşil ledin bağlı olduğu pin
int maviled=11; //mavi ledin bağlı olduğu pin
void setup()
{
```

```

Serial.begin(9600); //Seri iletişim ayarlarını yapıyoruz
//ledlerin bağlı olduğu pinleri çıkış olarak ayarlıyoruz
pinMode(kirmiziLed,OUTPUT);
pinMode(yesilLed,OUTPUT);
pinMode(maviLed,OUTPUT);

}
void loop()
{
  sensorSicaklik=analogRead(0); //sensörden gelen veriyi okuyoruz
  if(sensorSicaklik > 400 && sensorSicaklik < 520) /*okunan değer 520 ile
400 arasında ise*/
  {
    digitalWrite(yesilLed,LOW); /*yeşil Led yanar. Ölçülen sıcaklık oda
sıcaklığındadır*/
    digitalWrite(kirmiziLed,HIGH);
    digitalWrite(maviLed,HIGH);
    Serial.println("ölçülen sıcaklık oda sıcaklığında"); /* serial monitörde
göreceğimiz uyarı yazısı*/
  }
  else if(sensorSicaklik > 0 && sensorSicaklik <= 400) /*okunan değer 400
den küçükse*/
  {
    digitalWrite(yesilLed,HIGH);
    digitalWrite(kirmiziLed,LOW); /*kırmızı Led yansın. Ölçülen sıcaklık
oda sıcaklığından yüksek*/
    digitalWrite(maviLed,HIGH);
    Serial.println("ölçülen sıcaklık oda sıcaklığından yüksek"); /* serial
monitörde göreceğimiz uyarı yazısı*/
  }
  else if(sensorSicaklik>520 ) //okunan değer 520'den büyükse
  {
    digitalWrite(yesilLed,HIGH);
    digitalWrite(kirmiziLed,HIGH);
    digitalWrite(maviLed,LOW); //mavi Led yansın. Ölçülen sıcaklık oda
sıcaklığından düşük
    Serial.println("ölçülen sıcaklık oda sıcaklığından düşük"); /* serial
monitörde göreceğimiz uyarı yazısı*/
  }
  delay(100); // 100ms bekle
}

```

Öğrencinin	DEĞERLENDİRME				
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	TOPLAM
Adı :					RAKAM
Soyadı :					YAZI
Nu :					
Sınıfı :					
Ders Öğretmeni:					İmza:

# LDR UYGULAMALARI

# 7

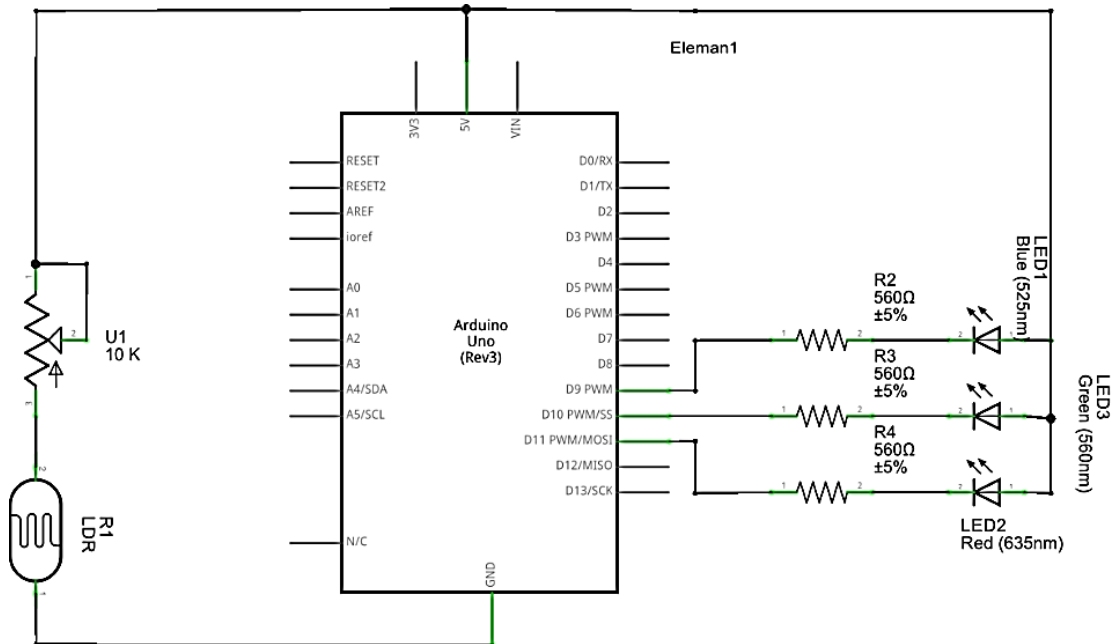
## 7.1 – Işık Etkili RGB LED

**Açıklama:** LDR'lerin ışık şiddetine bağlı olarak direnç değerleri değişmektedir. Üzerine düşen ışığın şiddetiyle ters orantılıdır. Maksimum ışık şiddetinde direnci minimumken, karanlıkta direnci maksimumdur. Aşağıdaki devrede Potansiyometre ile ldr gerilim bölücü olarak kullanılmıştır. Bu sayede ışık şiddetine göre hassasiyeti ayarlanabilen ışık etkili bir devre oluşturuyoruz. Devrede ldr üzerine düşen gerilim, ışığın şiddetine ve potansiyometrenin ayarına bağlıdır.

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X LDR
- 1 X 10 KΩ POT
- 3 X LED ( KIRMIZI, YEŞİL, MAVİ )
- 4 X 560 Ω

### Devre Şeması:



### Devrenin Yazılımı:

```
int ldrDeger=0;//sensörden okunan gerilim değerini bu değişkene kaydedeceğiz
int kirmiziLed=11;
int yesilLed=10;
```

```
int maviLed=9;
void setup()
{
  Serial.begin(9600); //Seri iletişim ayarları yapılıyor
  pinMode(kirmiziLed,OUTPUT);
  pinMode(yesilLed,OUTPUT);
  pinMode(maviLed,OUTPUT);
}
void loop()
{
  ldrDeger=analogRead(0); //sensörden gelen veri okunuyor
  //okunan değer 500 ile 400 arasında ise ortam ışık şiddeti az seviyede
  if(ldrDeger>400 && ldrDeger<500)
  {
    //yeşil Led yansın.
    digitalWrite(yesilLed,LOW);
    digitalWrite(kirmiziLed,HIGH);
    digitalWrite(maviLed,HIGH);
    Serial.println("ortam isik siddeti az"); /*s.m. de göreceğimiz uyarı
yazısını yazdırıyoruz*/
  }
  //okunan değer 400 den küçükse
  else if(ldrDeger>0 && ldrDeger<=400)
  {
    digitalWrite(yesilLed,HIGH);
    //ışık şiddeti yüksekse kırmızı Led yansın.
    digitalWrite(kirmiziLed,LOW);
    digitalWrite(maviLed,HIGH);
    Serial.println("ortam aydinlik");
  }
  else if(ldrDeger>=500 ) //okunan değer 500'den büyükse
  {
    digitalWrite(yesilLed,HIGH);
    digitalWrite(kirmiziLed,HIGH);
    //hava karardığında mavi Led yansın.
    digitalWrite(maviLed,LOW);
    Serial.println("ortam karanlik");
  }
  delay(200);
}
```

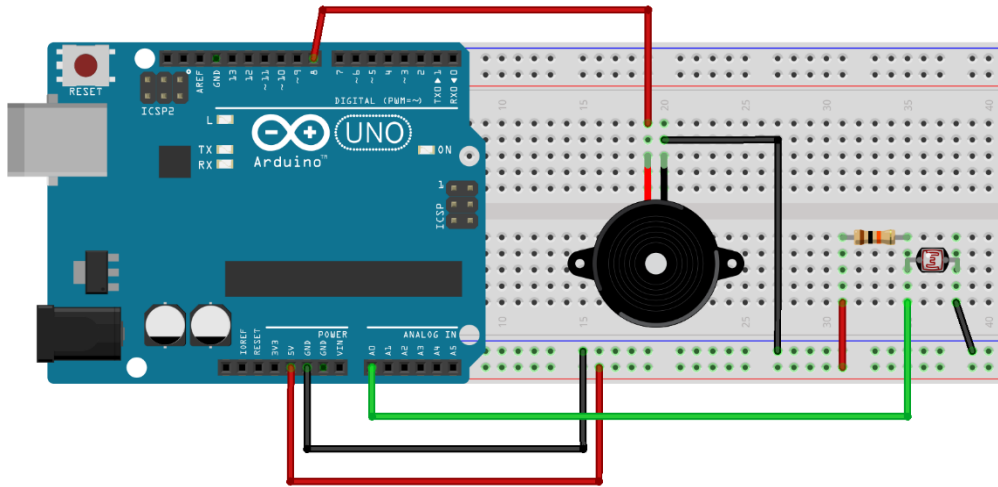
## 7.2 – Işık Etkisiyle Sesli Geri Bildirim

**Açıklama:** Bu uygulamada ışığı algılamak için LDR, algılanan ışık miktarına göre sesli geribildirim elde etmek için piezo sounder kullanacağız.

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X Piezo Sounder veya Buzzer
- 1 X LDR ( Light Dependent Resistor )
- 1 X 10KΩ Direnç

### Devre Şeması:



### Devrenin Yazılımı:

```
int piezoPin = 8; // Piezonun bağlandığı pin D8
int ldrPin = 0; // LDR' nin bağlandığı pin A0
int ldrValue = 0; // LDR değerini okuyacağımız değişkeni tanımlıyoruz
void setup() {
}
void loop() {
  ldrValue = analogRead(ldrPin); // LDR' den analog değeri okuyoruz
  tone(piezoPin,1000); // piezodan 1000Hz değerinde ses elde ediyoruz
  delay(25);
  noTone(piezoPin); // 25ms bekledikten sonra sesi kapatıyoruz
  delay(ldrValue); // LDR' den okunan değerın milisaniyesi kadar bekliyoruz
}
```

Öğrencinin	DEĞERLENDİRME				
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	TOPLAM
Adı :					RAKAM
Soyadı :					YAZI
Nu :					
Sınıfı :					
	Ders Öğretmeni:				İmza:





# DC MOTOR UYGULAMALARI

## 8

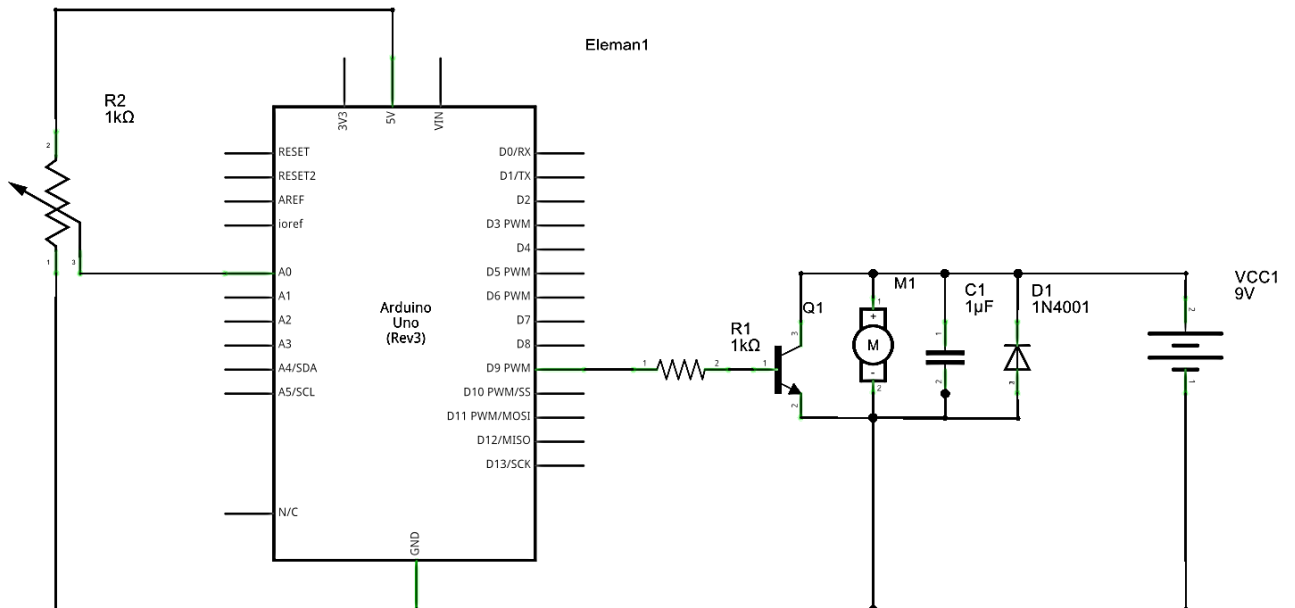
### 8.1- Potansiyometre ile DC Motor Hız Kontrolü

#### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X DC motor
- 1 X 1 K $\Omega$  Direnç
- 1 X 1 K $\Omega$  POT
- 1 X 1 N 4001
- 1 X 1 $\mu$ f
- 1 X BC237 Transistör
- 9 V pil

**Açıklama:** Bu devremizde **A0** pinine bağlı potun orta ucu, potun değeri değiştiğinde 0 – 5V arasında gerilimler üretir. ADC sonucu çıkan sayısal değerler aralığı 0 – 1023 arasındadır. Aşağıdaki kodumuzda **map** fonksiyonu kullanarak **deger** değişkeninin 0 – 1023 arasındaki değerler 0 – 255 arasında sınırlandırılmıştır. Bunun amacı pwm değerinin 0 – 255 arasında olabilmesidir. Buna göre pottan okunan değer 0 – 255 arasında pwm çıkışı olarak transistörün beyzine aktarılır.

#### Devre Şeması:



**Devrenin Yazılımı:**

```
const int motorPin=9; // Motoru kontrol edecek pwm pinini tanımlıyoruz
const int pot=0; //potun ayar ucunun bağlandığı A0 girişi
int deger =0; //Analog giriş değerini saklayan değişken
void setup()
{

    pinMode(motorPin,OUTPUT); //pwm pini çıkış olarak ayarlanıyor
}
void loop()
{
    deger=analogRead(pot); /*A0 girişinden analog değer okunup dijitale
dönüştürülüyor*/
    /* map metodu ile deger değişkenin değerinin 0 ila 1023 arasındaki
karşılığı 0 ila 255 arasında sınırlandırılıyor */
    deger=map(deger,0,1023,0,255);
    analogWrite(motorPin,deger); /* motorun hızı deger değişkeni içerisindeki
verinin değerine göre hızlanır veya yavaşlar */
    delay(10);
}
```

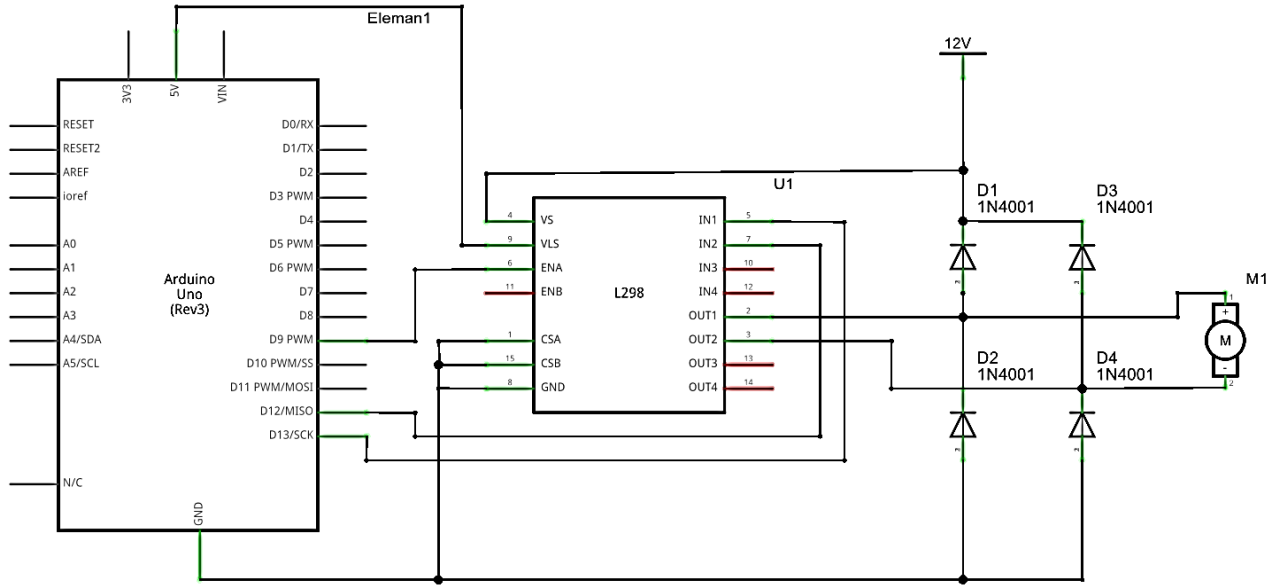
## 8.2- L298 İle Motor Hız ve Yön Kontrolü

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X DC motor
- 4 X 1 N 4001
- 1 X L298 Motor Sürücü

**Açıklama:** Aşağıdaki devrede **L298** entegresinin sadece A köprüsü kullanılmıştır. A köprüsü için; motor uçları out1 ve out2 çıkışına bağlanır. EnA pini Arduino' nun 9 nolu pwm pinine bağlanmıştır. IN1 ve IN2 pinleri motor dönüş yönünü kontrol eder ve Arduino' nun 12. 13. Pinlerine bağlanmıştır. Bu bağlantı şekliyle motorun hem hız hem de yön kontrolünü sağlayacağız.

### Devre Şeması:



### Devrenin Yazılımı:

```
const int ENA=9; //Pwm çıkışı L298 in ENA girişine bağlanıyor
const int IN1=13; //13 nolu pin L298 in in1 girişine bağlanıyor
const int IN2=12; //12nolu pin L298 in in1 girişine bağlanıyor
void setup()
{
  pinMode(ENA,OUTPUT);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
}
void loop()
{
  //motoru ileri yönde döndürüyoruz
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);

  //for döngüsü ile motorun dönüş hızı pwm ile ayarlanıyor

  for(int i=0;i<=255;i++)
```



# STEP MOTOR UYGULAMALARI

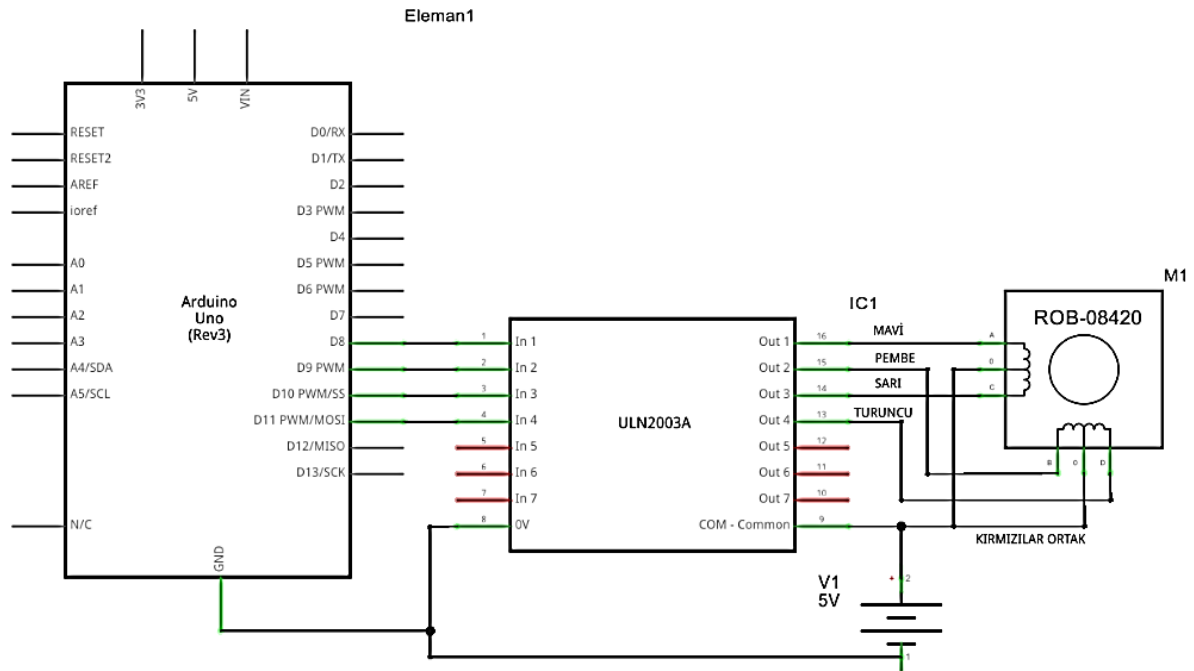
## 9

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X STEP Motor
- 1 X ULN2003A Motor Sürücü

**Açıklama:** Aşağıdaki devrede step motorun çalışması için gereken; motorun bobinlerine sırayla Arduino üzerinden pals'ler göndermek ve motor sürücü entegresi bu pals'leri lojik olarak tersleyerek step motora aktarır, motorun dönmesini sağlar. Biz bu uygulamada step motor kullanmak için hazır kütüphaneden faydalanacağız. Hazır kütüphane kullanımı bir çok uygulamada kodları hızlı bir şekilde hazırlamamızı sağlar.

### Devre Şeması:



### Devrenin Yazılımı:

```
#include <Stepper.h>
//motor pinleri için değişkenler tanımlanıyor
int motorPin1 = 8;    // Mavi pin 1
int motorPin2 = 9;    // Pembe pin 2
int motorPin3 = 10;   // Sarı pin 3
int motorPin4 = 11;   // Turuncu pin 4
```



# SERVO MOTOR UYGULAMALARI

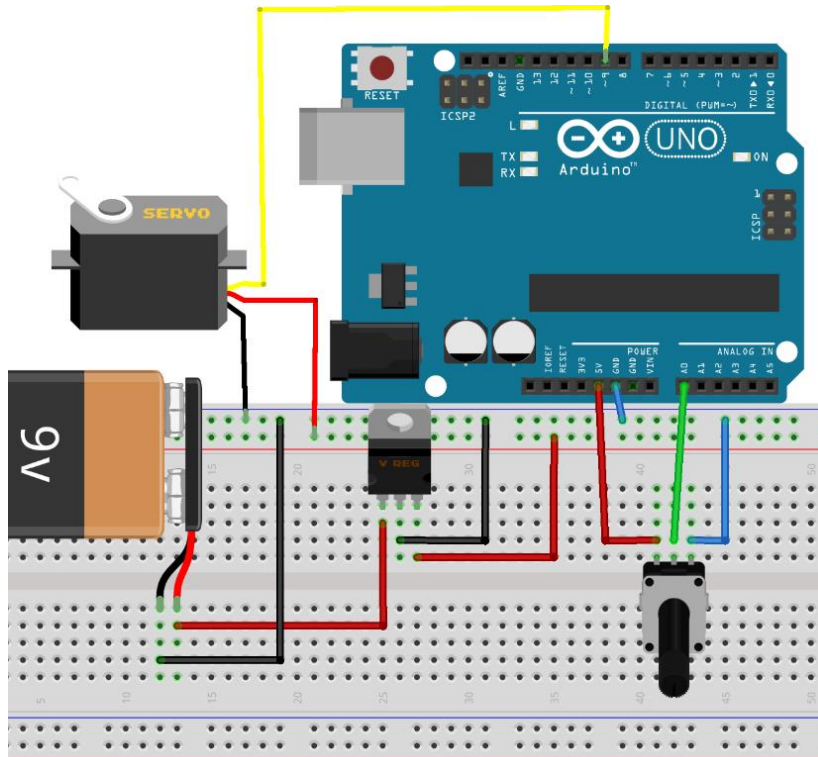
## 10

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X 7805
- 9 Volt pil
- Servo motor
- 10 K $\Omega$  pot

**Açıklama:** Uygulamamızda kullandığımız motor türü, verilen komuta göre açısız olarak hareket eder ve istediğimiz pozisyonda kalabilir. 3 bağlantı pinine sahiptirler. Bu pinler, GND ( kahve, siyah ), 5V ( kırmızı ) ve sinyal ucudur (beyaz ya da turuncu). Biz devremizde motorun beslemesini harici bir güç kaynağından vereceğimiz için 7805 regülatör entegresi kullanacağız. Aşağıdaki devrede potansiyometreden okunan 0 – 1023 arasındaki analog verilerin karşılığı olan değerler, map metoduyla 0 – 179 derece arasında sınırlandırılır. Bu sayede potansiyometrenin değeri değıştikçe servo motor 0 – 180 derece arasında hareket eder.

### Devre Şeması:







# RTC UYGULAMALARI

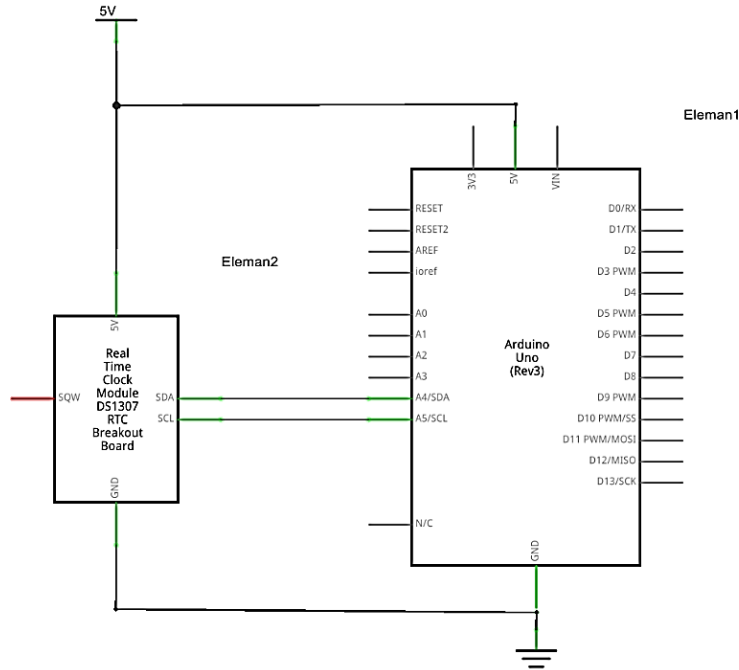
## 11

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X RTC Modülü

**Açıklama:** *Real Time Clock* ( gerçek zaman bilgisi ) ile uygulamamızda zaman bilgisini kullanabiliriz. Örneğin istenen gün/tarih/saat/dakika da çalışmasını istediğimiz bir röleyi programlayabilir ya da sensörden okunan verinin hangi zaman diliminde hangi değerleri aldığını RTC ile kolaylıkla görebiliriz.

### Devre Şeması:



### Devrenin Yazılımı:

```

/* RTC modülden saat ve tarih bilgisini okuyacağız
#include "Wire.h" // I2C için gerekli wire kütüphanesi
ekleniyor.
#define DS1307_ADRES 0x68 //ds1307 nin adres bilgisi belirtiliyor.
void setup(){
  Wire.begin();
  Serial.begin(9600);
}
void loop(){

```

```

    tarihsaatYazdir(); //anlık tarih ve saat bilgisini yazdır.
    delay(1000);
}
byte bcdToDec(byte val) {
    // binary olarak kodlanmış desimal veri (BCD) desimale çevriliyor.
    return ( (val/16*10) + (val%16) );
}

void tarihsaatYazdir(){

    // Register sayacı resetlenir.
    Wire.beginTransmission(DS1307_ADRES);
    byte zero = 0x00;
    Wire.write(zero);
    Wire.endTransmission();

    Wire.requestFrom(DS1307_ADRES, 7); /*Belirtilen adresten itibaren 7 byte
    veri okunması isteniyor.*/

    int saniye = bcdToDec(Wire.read()); //Okunan ilk veri, saniye
    int dakika = bcdToDec(Wire.read()); //Okunan ikinci, dakika
    int saat = bcdToDec(Wire.read() & 0b111111); //24 saat sistemine göre saat
    verisi
    int haftaninGunu = bcdToDec(Wire.read()); //0-6 -> 0.gün pazar. 6. gün
    cumartesi
    int gun = bcdToDec(Wire.read()); //ayın kaçınıcı gunu olduğu
    int ay = bcdToDec(Wire.read()); //yılın hangi ayı olduğu
    int yil = bcdToDec(Wire.read()); //hangi yılda olduğu verilerini ayarlar.

    //şu anda yüklü olan saat ve tarihi görüntüler
    Serial.print(ay);
    Serial.print("/");
    Serial.print(gun);
    Serial.print("/");
    Serial.print(yil);
    Serial.print(" ");
    Serial.print(saat);
    Serial.print(":");
    Serial.print(dakika);
    Serial.print(":");
    Serial.print(saniye);
    Serial.print("haftanın gunu:");
    Serial.println(haftaninGunu);

}

```

Öğrencinin	DEĞERLENDİRME				
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	TOPLAM
Adı :					RAKAM
Soyadı :					YAZI
Nu :					
Sınıfı :					
	Ders Öğretmeni:				İmza:

# LCD VE TFT EKRAN UYGULAMALARI

## 12

**Açıklama:** LCD ekranda 16 bağlantı pini vardır. 16 sütun, 2 satırdan oluşan bu ekran aynı anda 32 karakter görüntüleyebilir. Paralel LCD modüllerde 8 adet data gönderim hattı (DB0...DB7) bulunmaktadır. LCD ekran pinleri aşağıdaki gibidir.

PİN NUMARASI	PİN İSMİ	PİN ÖZELLİĞİ
1.	VSS	GND
2.	VDD	+5 Volt
3.	VO	Arka plan ışık kontrastı
4.	RS	Register seçici (Karakter/Komut)
5.	RW	Read/Write
6.	EN	Enable
7.	D0	Data hatları
8.	D1	
9.	D2	
10.	D3	
11.	D4	
12.	D5	
13.	D6	
14.	D7	
15.	A	Arka ışık anot
16.	K	Arka ışık katot

- Register seçici pini ( **RS** ) ile LCD ye komut ya da karakter verisi gönderilebilir. Sonraki satıra geçmesi ya da istenilen sütundan itibaren yazdırma işlemi için komut kullanılır.
- **RW** pini **GND**' ye bağlanarak LCD ekrana yazma işlemi yapılır. Okuma işlemi yapılmaz.
- **EN** pini LCD' ye veri hazır olduğunu söyler.
- **Data** pinleri **D4-D7** veri göndermek için kullanılır ( D0-D3 arasındaki pinler kullanılmaz ).
- LCD ekranın arka plan ışığını açmak için **5V A** pinine ve **GND K** pinine bağlanır.

Aşağıda Arduino ve LCD pin bağlantıları verilmiştir.

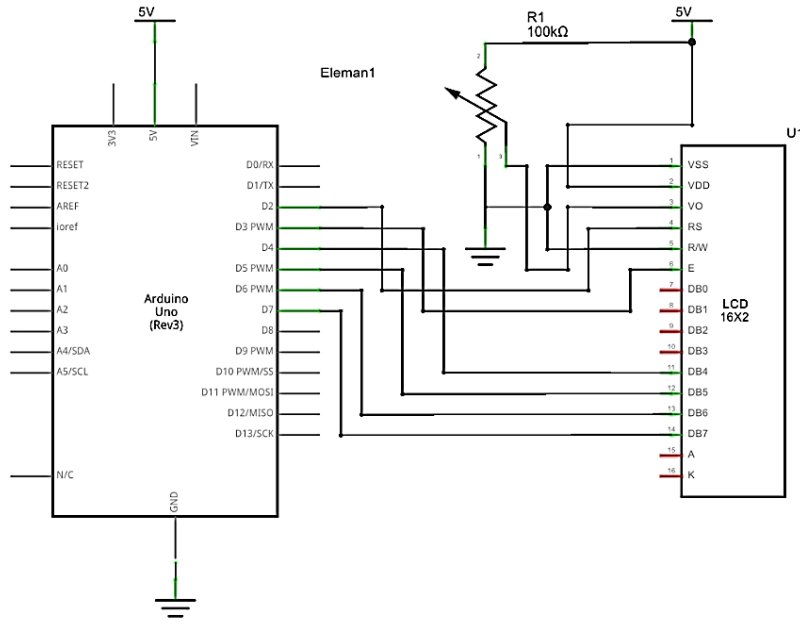
LCD PİN	ARDUİNO PİN
RS	PIN2
EN	PIN3
D4	PIN4
D5	PIN5
D6	PIN6
D7	PIN7

## 12.1- LCD Ekranı Veri Göndermek

### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X LCD Ekran
- 100 K $\Omega$  pot

### Devre Şeması:



### Devrenin Yazılımı:

```
#include <LiquidCrystal.h> //LCD kütüphanesi ekleniyor
int time=0; //Başlangıç zamanı 0 olarak ayarlandı
//LCD ekran ayarları yapılıyor
LiquidCrystal LCD(2,3,4,5,6,7);
void setup()
{
//16x2 ekran kullanılıyor.
LCD.begin(16,2);
//Lcd ekrana veri yazdırılıyor
LCD.print("**GİRESUN MTAL**");
}
void loop()
{
// cursoru 2. Satıra çekiyoruz
LCD.setCursor (0,1);
// anlık zaman sayacı ekrana yazdırılıyor
LCD.print(time);
//1 saniye bekleniyor
delay(1000);
//zaman sayacı 1 arttırılıyor
time++;
}
```



### 12.3- TFT Ekran Veri Yazdırma ve Şekil Çizme

**Açıklama:** Bu uygulamada kullanacağımız dokunmatik ekran SSD1289 LCD ekran sürücüsüne sahiptir. Modüldeki dokunmatik panel kontrolü ve kart okuyucu SPI veri yolu kullanarak iletişim sağlarken, ssd1289 LCD ekran paralel veri iletişimi kullanmaktadır. Aşağıda TFT ekran ve Arduino Mega2560 bağlantıları verilmiştir.

TFT EKРАН PİNLERİ	ARDUİNO PİNLERİ
LED-A	5V arka plan ışığı
VCC	Besleme gerilimi ( 3V3 )
RD	3V3 ile HIGH konumda tutulur
GND	GND
DB0, DB1, DB2, DB3, DB4, DB5, DB6, DB7	D37, D36, D35, D34, D33, D32, D31, D30 VERİ PİNLERİ
DB8, DB9, DB10, DB11, DB12, DB13, DB14, DB15	D22, D23, D24, D25, D26, D27, D28, D29 VERİ PİNLERİ
RS ( REGİSTER SEÇİCİ )	D38
WR	D39
CS (PIN 15)	D40
REST	D41

#### Devrede Kullanılan Malzemeler:

- 1 X Arduino MEGA2560
- 1 X TFT Ekran

#### Devrenin Yazılımı:

```
#include <UTFT.h>
// kullanılacak fontlar bildiriliyor.
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];
// bu değerler kullandığınız dokunmatik ekranın özelliğine göre değişebilir.
```

```
UTFT myGLCD(ITDB32S,38,39,40,41);
```

```
void setup()
{
  myGLCD.InitLCD();
  myGLCD.clrScr();
}
void loop()
{
  myGLCD.setColor(65, 255, 0);
  myGLCD.setBackColor(125,35, 50);
  myGLCD.fillScr(30,60,120);
  myGLCD.setFont(BigFont);
  myGLCD.print("GİRESUN", CENTER, 32);
  myGLCD.setFont(BigFont);
  myGLCD.print("ARDUİNO UYGULAMALARI", CENTER, 202);
  myGLCD.drawRoundRect(5,5,315,235);
  myGLCD.drawCircle(160,120,45);
  myGLCD.fillCircle(160,120,15);
```











# SU SEVİYE ÖLÇÜMÜ UYGULAMASI

## 14

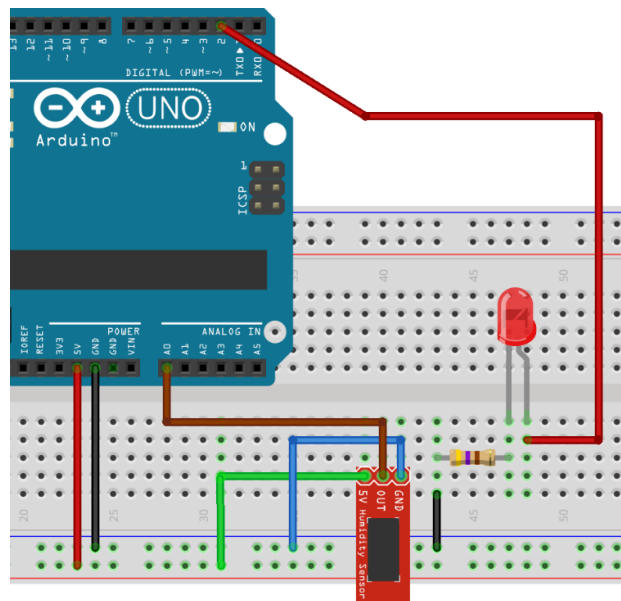
### Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 1 X Su Seviye Sensörü
- 1 X Led
- 1 X 470  $\Omega$  Direnç

**Açıklama:** Su seviye sensörünün çalışma mantığı; bakır plakalar arasında normal şartlarda elektriksel olarak bağlantı yokken, su içerisinde çalışmaya başladığında bakır plakalar arasında ki su seviyesine göre üzerinden geçen elektrik akımı artmakta böylece çıkış ucundan, su seviyesine göre değişen analog gerilim elde edilmektedir. Bu gerilim 0 – 3.3 Volt arasında değişim gösterir.



### Devre Şeması:





# SERIAL .AVAILABLE ( ) FONKSİYONU UYGULAMASI

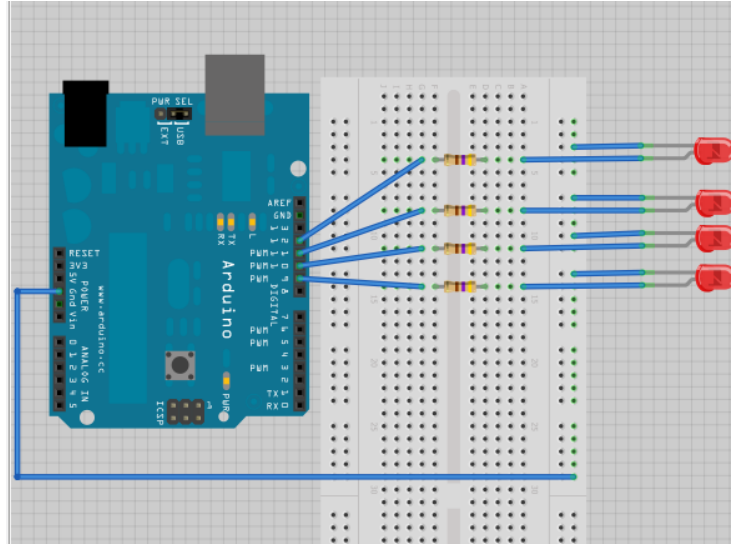
# 15

## Devrede Kullanılan Malzemeler:

- 1 X Arduino UNO
- 4 X Led
- 4 X 470  $\Omega$  Direnç

**Açıklama:** Bu uygulamada seri haberleşme üzerinden gönderilen karakterlere göre istenilen LED'i yakan bir program üzerinde bu fonksiyonun nasıl çalıştığını göreceğiz. Arduino ya bağlı 4 ledi seri haberleşme üzerinden ASCII karakterlerine göre yakıp söndüreceğiz. Gönderilen karakterlerden her biri ( A,B,C,D ) bir ledi kontrol edecek. Karakterleri göndermek için serial monitörü kullanacağız. SM penceresindeki üst bölüme göndereceğimiz karakteri yazıp Send butonuna bastığımızda yazdığımız karakter Arduino ya gönderilecektir.

## Devre Şeması:



## Devrenin Yazılımı:

```

/*****

```

*Serial Monitor üzerinden gönderilen karakterlere göre 4 LED'den birisinin durumunu değiştirilir.*

*Bağlantılar:*

- Arduino Dijital 9 - LED1*
- Arduino Dijital 10 - LED2*
- Arduino Dijital 11 - LED3*
- Arduino Dijital 12 - LED4*

Gönderilecek Karakterler:

A => 1.LED

B => 2.LED

C => 3.LED

D => 4.LED

\*\*\*\*\*/

// Gelen verinin aktarılacağı karakter değişkeni

char karakter = 0;

//LED Pinlerinin numaraları

const byte LEDpinleri[4] = { 9,10,11,12};

// LED'lerin durumlarını tutan değişkenler

// true = LED Yanık, false = LED Sönük

boolean LEDDurum[4] = { false,false,false,false};

void setup() {

Serial.begin(9600); // Seri haberleşmeyi başlatıyoruz(9600 bps)

pinMode(LEDpinleri[0],OUTPUT);

pinMode(LEDpinleri[1],OUTPUT);

pinMode(LEDpinleri[2],OUTPUT);

pinMode(LEDpinleri[3],OUTPUT);

}

void loop() {

// Veri gelmiş mi?

if (Serial.available() > 0) {

    // gelen veriyi oku

    karakter = Serial.read();

    // Gelen karakteri yazdır

    Serial.print("Gelen karakter: ");

    Serial.println(karakter);

    //Gelen Karakterler göre ilgili LED'i yakıyoruz

    switch(karakter)

    {

    case 'A':

        LEDDurumDegistir(0);

        break;

    case 'B':

        LEDDurumDegistir(1);

        break;

    case 'C':

        LEDDurumDegistir(2);

        break;

    case 'D':

        LEDDurumDegistir(3);

        break;

    default:

        Serial.println("Gecersiz bir karakter gonderildi!");









```

* LCD RS      ==> digital-12
* LCD Enable ==> digital-11
* LCD D4      ==> digital-5
* LCD D5      ==> digital-4
* LCD D6      ==> digital-3
* LCD D7      ==> digital-2
* LCD R/W     ==> toprak (GND)
* 10K potansiyometre 0-5V bağlantı:
* orta ucu LCD'nin V0 pinine(pin 3)
*****/

#include <LiquidCrystal.h>
//Değişkenleri tanımlıyoruz
float sicaklik;
float gerilim;
int sicaklikPin = 0;

// LCD bağlantılarımızı ile LCD'yi başlatıyoruz
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{
  Serial.begin(9600); //Seri haberleşmeyi başlat (9600 bps)
  // LCD'mizin başlangıç ayarı 16 karakter x 2 satır
  lcd.begin(16, 2);
}
void loop()
{
  gerilim = analogRead(sicaklikPin); // Sensörden gelen değeri oku
  gerilim = (gerilim / 1023) * 5000; // mV cinsinden gerilimi buluyoruz
  sicaklik = gerilim / 10.0; // Gerilim değerini sıcaklığa çeviriyoruz

  lcd.setCursor(0,0);
  lcd.print("Sicaklik:");
  lcd.setCursor(0,1);
  lcd.print(sicaklik);
  delay(1000); //Saniyede bir ölçüm gönderiyoruz
}

```

**Notlar:**

.....

.....

.....

.....

.....

Öğrencinin	DEĞERLENDİRME				TOPLAM	
	Bilgi (%.....)	İşlem Basamakları (%.....)	İş Alışkanlığı (%.....)	Süre (%.....)	RAKAM	YAZI
Adı :						
Soyadı :						
Nu :						
Sınıfı :						
	Ders Öğretmeni:				İmza:	