



ARDUINO IDE UYGULAMALARI EL KİTAPÇIĞI

KODLUYORUZ DERNEĞİ

İÇİNDEKİLER

1. LED Yakma(Blink)	2
2. Karaşımşek Uygulaması	7
3. Buzzer Uygulaması	10
4. Buton ile LED Yakma	13
5. Seri Port Üzerinden Haberleşme	16
6. Potansiyometre Kullanımı	19
7. Potansiyometre ile LED Kontrolü	22
8. RGB LED Kullanımı.....	25
9. LDR Kullanımı	28
10. LDR ile Sokak Lambası Yapımı.....	31

LED Yakma(Blink)

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Olayı başlatma
- Fonksiyonlar
- LED
- Direnç
- Delay

Gerekli Malzemeler

- Bilgisayar
- Arduino IDE programı
- Arduino Uno
- LED
- 220 ohm direnç
- Breadboard
- Jumper kablolar
- USB kablo

Amaç

- LED ve direnç bağlayarak devre kurmayı öğretmek.
- Kod kullanmadan ve kod kullanarak devreyi çalıştırmayı göstermek.

Kazanım

- Elektornik kart kavramını öğrenir.
- Arduino IDE programını tanır.
- Arduino IDE ile Arduino kart arasında bağlantı kurmayı öğrenir.
- Bu program ile yazılım dillerine giriş yapar.
- Temel elektronik devre elemanlarını tanır.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Temel düzeyde breadboard kullanır.

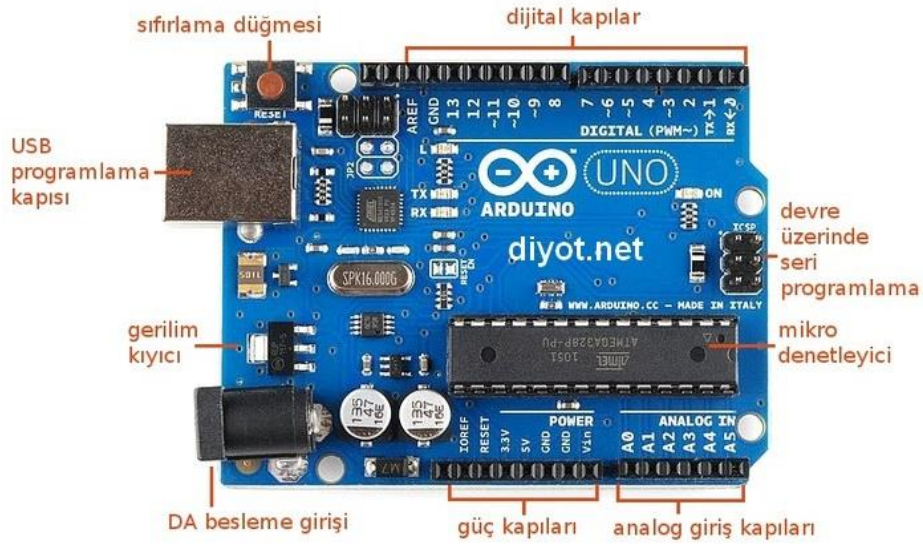
Arduino Nedir?

Arduino, her yaştan ve her eğitim seviyesindeki kişilerin, elektronik projeleri kolaylıkla yapılabilmesi için geliştirilmiş olan açık kaynaklı bir mikro kontrolcü platformudur.

Arduino ile sensörlerden gelen değerleri okuyabilir ve bu değerleri işlemlere sokabiliriz, ayrıca ışık, motor gibi araçları çalıştırabilir ve bu araçlarla aklımıza gelebilecek tüm elektronik projeleri gerçekleştirebiliriz.

Arduino Kart

Arduino, birçok sensörden girdi alarak çevreyi algılar ve ışıkları, motorları ve diğer aktüatörleri kontrol ederek çevresini etkiler.

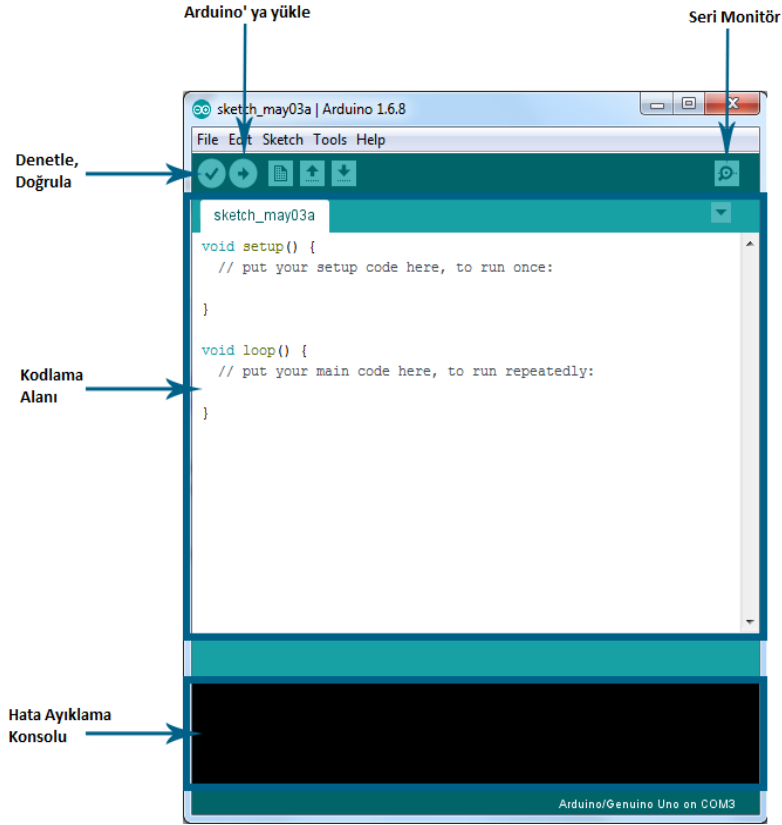


Arduino Software

Arduino programlama dilinde kod yazarak ve Arduino geliştirme ortamını kullanarak neler yapılacağını anlatabilirsiniz.

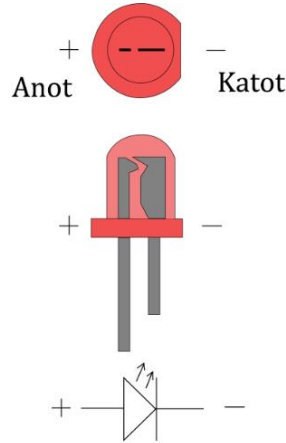
Arduino programının geliştirilmesinden önce yapmanız gereken ilk şey Arduino IDE yazılımı kurmak. Bu yazılım, Arduino programını geliştirmek için gerekli temel geliştirme ortamını sağlar. Arduino IDE'yi indirmek için <http://www.arduino.cc> web adresine girerek “SOFTWARE” sekmesinin altında bulunan “DOWNLOADS” bölümüne tıklayarak indirme ekranına ulaşabiliriz.

Daha sonra karşımıza gelen ekranda farklı işletim sistemi platformlarına uygun olarak ayrı linkler karşımıza çıkıyor. İşletim sisteminize uygun olan linki tıklayarak programın kurulum dosyalarını indirebilirsiniz. İndirdiğimiz dosya ile kurulumu sağladıktan sonra karşımıza gelen ekran aşağıdaki gibi olacaktır.



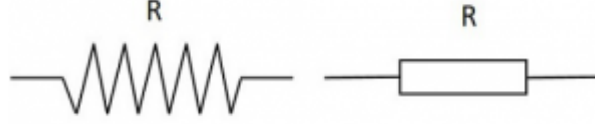
Led Nedir?

LED, ışık yayan diyot anlamına gelen Light Emitting Diode sözcüğünün baş harflerinden oluşan bir kısaltmadır. LED'lerin uzun bacak +(anot) ve kısa bacak -(katot) şeklinde 2 bacağı bulunur. Uzun bacak, Arduino kart üzerindeki dijital sinyal pinine, kısa bacak ise yine kart üzerindeki GND pinine bağlanmalıdır. Ancak dikkat etmemiz gereken bir nokta var. Arduino pinlere 5V enerji gönderir, LED ler ise 3V ile çalışmaktadır. İşte aradaki 2V kadar enerjiyi direnç ile dengeliyoruz.



Direnç Nedir? (Kaynakça: <https://maker.robotistan.com/direnc/>)

Dirençler, elektrikli devrelerde akımı sınırlayarak belli bir değerde tutmaya yararlar. Bunun haricinde hassas devre elemanlarının üzerlerinden yüksek akım geçmesini önlerler, besleme gerilimini ve akımı bölmek için de kullanılırlar. Direncin birimi Ohm (Ω)'dur. Denklemlerde **R** harfi ile gösterilir. Elektronik devrelerde direncin sembolü 2 farklı şekilde gösterilebilir:



Temel Arduino IDE Fonksiyonları:

Setup()

Değişkenleri başlatmak, pin modlarını başlatmak, kütüphaneleri kullanmak vb. için kullanılır. Setup() fonksiyonu, her bir güç açıldıktan veya Arduino kartının sıfırlanmasından sonra sadece bir kez çalışır.

Loop()

Loop() fonksiyonu, setup fonksiyonu çalıştırıldıktan sonra çalıştırılır ve bir sonsuz döngü işlevi görür. Loop fonksiyonun bu sonsuz döngü özelliği, sürekli tekrar edecek olan işlemlerimizin gerçekleşmesini sağlar.

pinMode()

Belirtilen pini, bir giriş veya çıkış olarak davranacak şekilde yapılandırır.

digitalWrite()

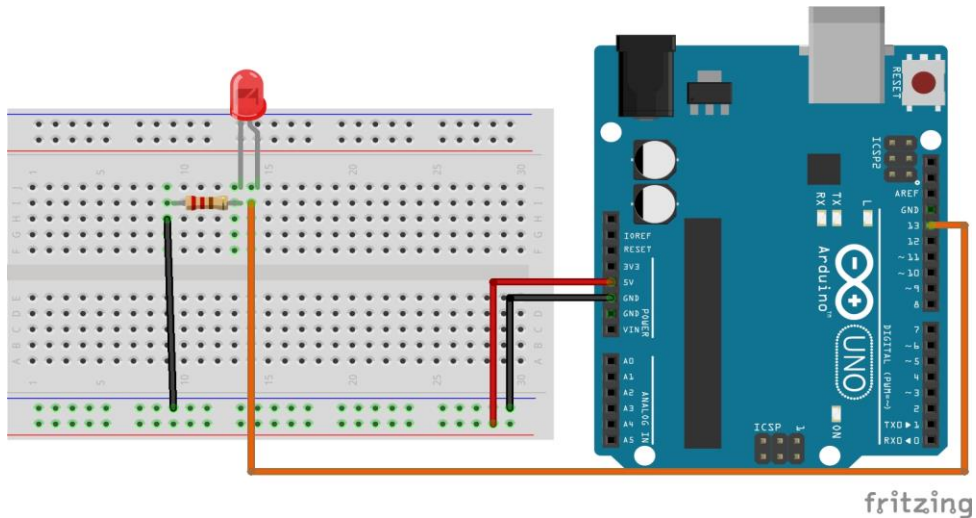
Belirtilen dijital pine HIGH (5V) veya LOW(0V) değerlerini gönderir.

Delay()

Programı, parametre olarak belirtilen süre (milisaniye cinsinden) için duraklatır.

***1 Saniyede 1000 milisaniye var.

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra LED yakma(blink) projemizin kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.
 - Kodların yanında her bir satır için açıklamalar bulunmaktadır.

```
void setup() { // Arduino çalıştığında
  pinMode(7, OUTPUT); // 7. pini çıkış olarak tanımlıyoruz
}

void loop() { // Arduino Çalıştığı sürece
  digitalWrite(7, HIGH); // 7. pine elektrik gönder
  delay(1000); // 1 saniye bekle
  digitalWrite(7, LOW); // 7. pine giden elektriği kes
  delay(1000); // 1 saniye bekle
}
```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino'ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, breadboard üzerindeki LED'in 1 saniye yanıp, 1 saniye söndüğünü göreceğiz.

Karaşimşek Uygulaması

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Olayı başlatma
- Fonksiyonlar
- LED
- Direnç
- PinMode
- Değişken tipi
- Delay

Gerekli Malzemeler

- Bilgisayar
- Arduino IDE programı
- Arduino Uno
- 8 adet LED
- 8 adet 220 ohm direnç
- Breadboard
- Jumper kablolar
- USB kablo

Amaç

- Birden fazla LED i programlama diliyle kontrol etmek

Kazanım

- Temel elektronik devre elemanlarını tanır.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.
- OUTPUT, INPUT ve HIGH-LOW komutlarını öğrenir.
- void setup ve void loop fonksiyonlarını öğrenir.
- int veri tipi ile tam sayı tipi değişken oluşturmayı öğrenir.
- for döngüsü ve dizileri bir arada kullanır.

İlk proje olarak, Arduino'yu programlayarak bir LED yanıp sönmeyi nasıl yapacağımızı öğrendik. Bu proje ile, LED'lerin akışın etkisini göstermesini sağlamak için 8 LED'i kontrol etmek için Arduino kullanacağız.

For Döngüsü

Programlamanın en temel yapılarından birisi olan döngüler, birbirini tekrar eden işlemleri düzenli ve sistematik şekilde yapabilmemizi sağlar. Ayrıca döngüler sayesinde bazen binlerce kod satırı yazma yükünden de kurtuluruz.

For döngüsü daha çok döngü sayısı ve aralığını bildiğimiz durumlarda kullanılır. Döngüdeki sayacın başlangıç, bitiş ve artış, azalış değerleri belirtebiliriz.

Döngünüzün yapısı aşağıdaki gibidir.

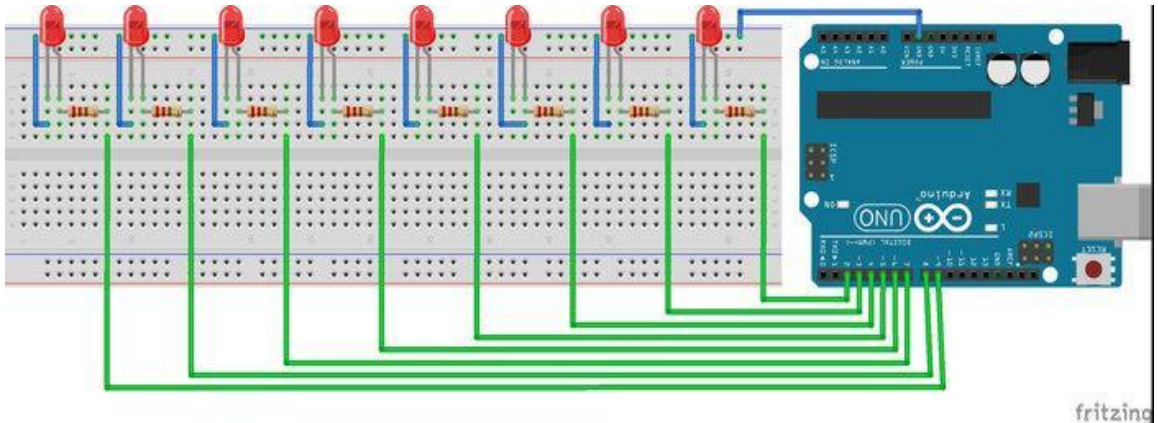
for(başlangıç değeri; koşul; artış miktarı)

```
{  
    Tekrar edilmesi istenilen komut/komutlar;  
}
```

Örnek: 9 defa 1'er saniye aralıklarla LED'lerin yanıp sönmesini sağlayan kod dizisi.

```
int a;  
for(a=0; a<10;a=a+1)  
{  
    digitalWrite(led,HIGH);  
    delay(1000);  
    digitalWrite(led,LOW);  
    delay(1000);  
}
```

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra karaşimşek projemizin kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.
 - Kodların yanında her bir satır için açıklamalar bulunmaktadır.

```
const int LEDdizisi[] = {2,3,4,5,6,7,8,9};

void setup () {

    for(int i=0; i<8; i++) /*LED dizisi elemanlarına ulaşıyoruz */
    {
        pinMode(LEDdizisi[i], OUTPUT); /* LED pinleri çıkış olarak ayarlandı */
    }
}

void loop() {
    for(int i=0; i<8; i++){ /* Tüm LED'leri sırayla 50 milisaniye yakıp söndürüyoruz */
        digitalWrite(LEDdizisi[i], HIGH);
        delay(50);
        digitalWrite(LEDdizisi[i], LOW);
    }

    for(int j=7; j>=0; j--) /* LED'leri geri yönde 50 milisaniye yakıp söndürüyoruz */
    {
        digitalWrite(LEDdizisi[j], HIGH);
        delay(50);
        digitalWrite(LEDdizisi[j], LOW);
    }
}
```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino'ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, breadboard üzerindeki LED'lerin sırası ile yanıp daha sonra sırası ile söndüğünü göreceğiz.

Buzzer Uygulaması

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Olayı başlatma
- Fonksiyonlar
- Buzzer
- PinMode
- Değişken tipi
- Delay

Gerekli Malzemeler

- Bilgisayar
- Arduino IDE programı
- Arduino Uno
- Buzzer
- Breadboard
- Jumper kablolar
- USB kablo

Amaç

- Kodlama yardımıyla devreden ses çıktısı almak

Kazanım

- Temel elektronik devre elemanlarını tanır.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Temel düzeyde breadboard kullanır.
- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.
- OUTPUT, INPUT ve HIGH-LOW komutlarını öğrenir.
- void setup ve void loop fonksiyonlarını öğrenir.
- delay kavramını öğrenir.

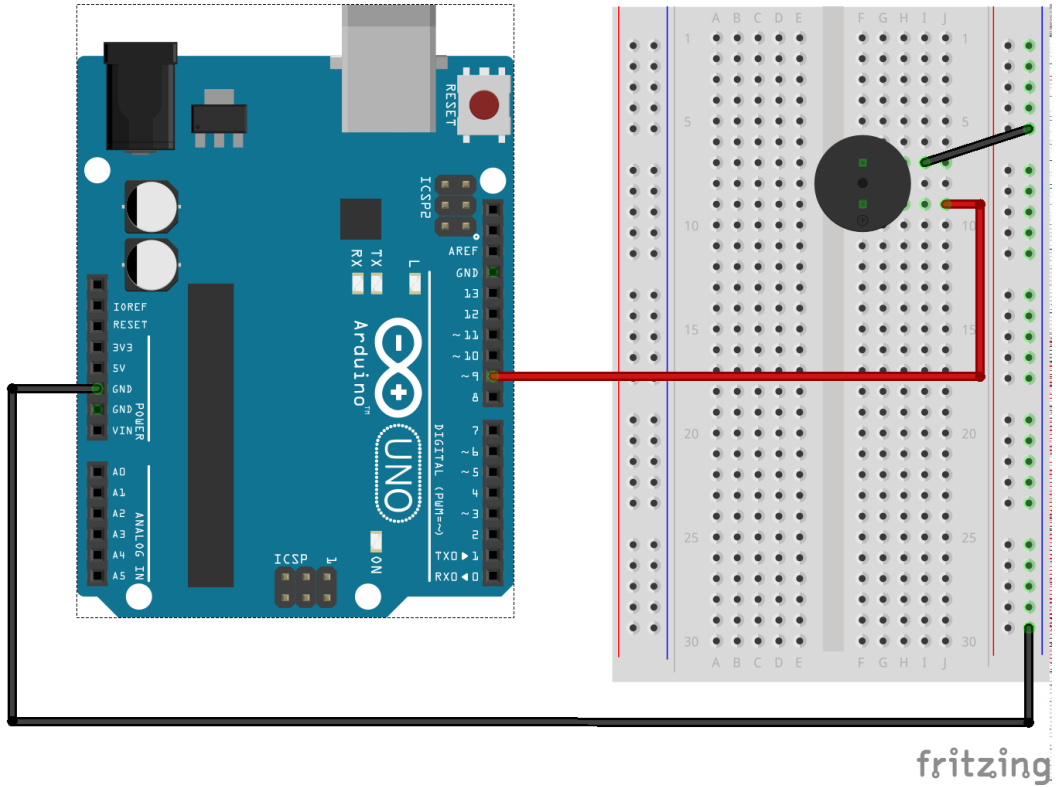
Bu çalışmada Buzzer kullanımını öğreneceğiz.



Buzzer; projelerimizde uyarı sesleri çıkarabilmek amacı ile kullanılan mini hoparlördür. 2 çeşit buzzer vardır. Bunlar, aktif ve pasif buzzerlardır. Aktif buzzer istenilen notaya göre ses çıkarabilir ve programla çeşitli müziklerin yapılması mümkün olan buzzerlardır. Pasif buzzer ise, sadece tek bir tonda ses çıkabilen ve sadece uyarı amacı ile kullanılan buzzerlardır.

Biz bu projemizde farklı notalar kullanacağımız için buzzerın aktif buzzer olmasına dikkat etmeliyiz.

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra buzzer projemizin kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.
 - Kodların yanında her bir satır için açıklamalar bulunmaktadır.

```
int buzzerPin=8; // 8 numaralı dijital pine bağladık
void setup()
{
  pinMode(buzzerPin, OUTPUT); // 8 numaralı pini çıkış olarak tanımladık
}
void loop()
{
  digitalWrite(buzzerPin, HIGH);
  delay(2000);
  digitalWrite(buzzerPin, LOW);
  delay(2000);
}
```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino'ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, breadboard üzerindeki buzzerın 2 saniye ses verip, 2 saniye sustuğu uygulamamızı göreceğiz.

Buton ile LED Yakma

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Olayı başlatma
- Fonksiyonlar
- Kontrol yapıları
- PinMode
- Değişken tipi
- LED
- Direnç
- Buton

Gerekli Malzemeler

- Bilgisayar
- Arduino IDE programı
- Arduino Uno
- Led
- Button
- 220 Ohm direnç
- 10K direnç
- Breadboard
- Jumper kablolar
- USB kablo

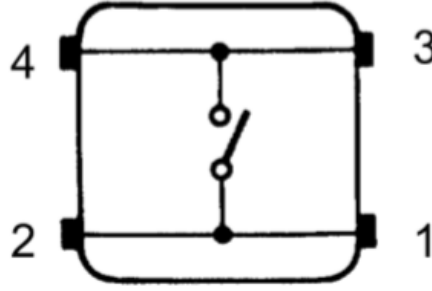
Amaç

- Arduino IDE programını kullanarak bir buton yardımıyla devreyi çalıştırmayı göstermek.
- Fonksiyonlar ve koşullar konularına giriş yapmak.

Kazanım

- Temel elektronik devre elemanlarını tanıır.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Temel düzeyde breadboard kullanır.
- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.
- OUTPUT, INPUT ve HIGH-LOW komutlarını öğrenir.
- void setup ve void loop fonksiyonlarını öğrenir.
- if-else koşullarıyla komut vermeyi öğrenir.
- Fonksiyonlar ve koşullar konularına giriş yapılır.
- int veri tipi ile tam sayı tipi değişken oluşturmayı öğrenir.

Butona basıldığında LED yanacak, butondan el çekildiğinde LED sönecek.



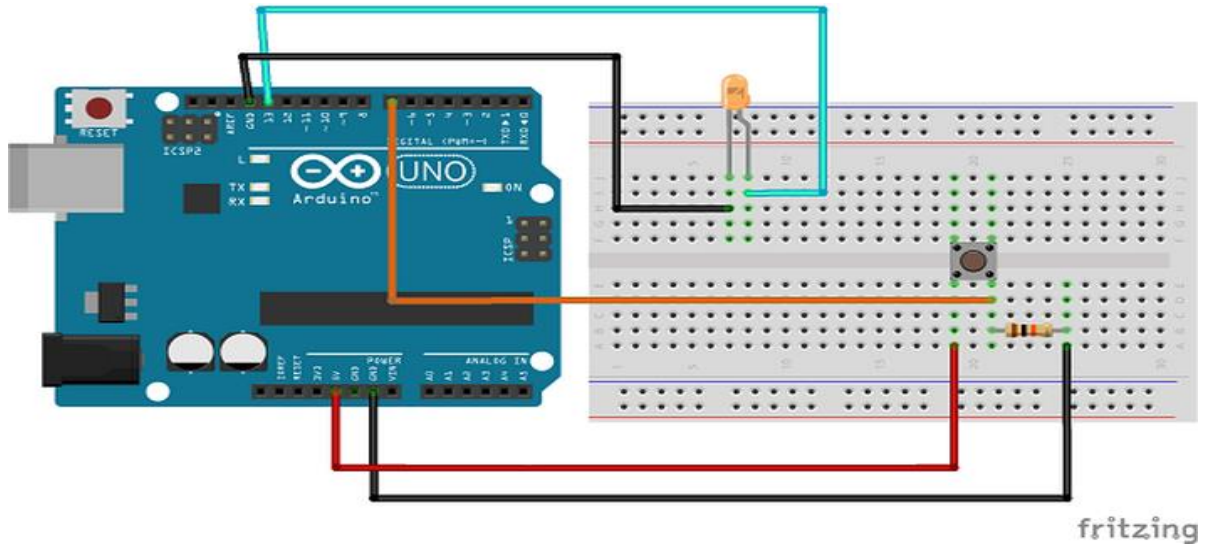
Butonlar, elektronik cihazları kontrol etmek için kullanılan yaygın bir bileşendir. Devreleri bağlamak veya ayırmak için genellikle anahtar rolünde kullanılırlar.

Etrafımızdaki birçok kullandığımız elektrikli ya da elektronik cihazların üzerinde buton mekanizması bulunur. Bunlara; akıllı telefonların ana ekran tuşu ve ses açma - kapama tuşları; televizyonların uzaktan kumandalarında bulunan tuşlar; otobüs, metro, tramvay gibi toplu taşıma araçlarında bulunan acil durum butonları, günlük yaşamınızdan örnekler olabilir.

digitalRead(pin)

Dijital pinlerden gelen sinyalleri okumak için kullanılır. Pinin YÜKSEK veya DÜŞÜK değerini döndürür.

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra buton ile LED yakma projemizin kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.

```
const int buttonPin =7;
const int ledPin = 13;
int buttonState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino'ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, breadboard üzerindeki LED'in butona bastığımızda yanacak, butonu bıraktığımızda sönecek şekilde çalıştığını göreceğiz.

Seri Port Üzerinden Haberleşme

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Seri Port
- Seri Port haberleşmesi
- Fonksiyonlar
- Delay

Gerekli Malzemeler

- Arduino UNO
- Breadboard
- USB kablo

Amaç

- Seri port kullanımını öğrenme

Kazanım

- Arduino ile sensörler arasındaki bağlantıyı görmeyi öğrenir.
- Sensörlerden gelen değerleri görür.
- Arduino ile farklı cihazlar arasında haberleşmeyi öğrenir.
- void setup ve void loop fonksiyonlarını öğrenir.

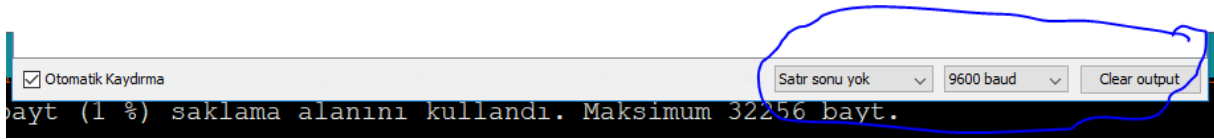
Projelerimizde Arduino ile sensörler arasındaki bağlantıyı ve sensörlerden gelen değerleri görebilmek için seri port ekranını kullanırız.

Seri haberleşme ile bu protokolü destekleyen tüm cihazlarla haberleşebiliriz. Biz genellikle projelerimizde, Bluetooth, ESP8266, USB gibi haberleşme protokollerini kullanacağız.

Not: Arduino' nun 0 ve 1 numaralı yani Rx ve Tx pinleri seri haberleşmeyi sağlamaktadır. Bu pinler aynı zamanda Arduino' nun bilgisayarla haberleşmesini sağlayan USB hattına da bağlıdır. 0 ve 1 numaralı pinler başka bir yere bağlı olduğunda, Arduino bilgisayarla haberleşmesini sağlayamamaktadır. Bu yüzden Arduino' ya kod atarken bu pinlerin bir yere bağlı olmamasına dikkat edilmelidir.

USB ile Bilgisayara Veri Gönderme:

Arduino' nun USB kablosu üzerinden bilgisayara veri aktaracağız. Bunun için öncelikle haberleşme hızını (Baud Rate) ayarlamalıyız.



Bu ayarın sadece bir kere yapılması yeterli olduğu için, haberleşme hızı setup fonksiyonu içerisinde ayarlanmalıdır. Artık bilgisayara veri aktarmaya hazırız.

“Serial.begin(9600)”

Uygulama:

- İlk olarak Arduino IDE programını açıyoruz.
- Arduino' yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın “Araçlar” sekmesinden “Kart” kısmına gelerek “Arduino/Genuino Uno” seçeneğini seçiyoruz. Son olarak da “Araçlar” sekmesinden “Port” kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino' yu kullanıma hazırlamış oluyoruz.
- Daha sonra seri port ile haberleşmemizi sağlayacak olan kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.
 - Kodların yanında her bir satır için açıklamalar bulunmaktadır.

```
void setup() {  
    Serial.begin(9600); /* Haberleşmeyi başlattık. Haberleşme hızı 9600*/  
}  
void loop() {  
    Serial.println("Arduino ile Programlama Öğreniyorum");  
    /* Görüntülenmesi istenen yazı;  
    Serial.println: Alt satıra yazı yazmak için kullanılır.  
    Serial.print: Aynı satırda yazdırmak için kullanılır*/  
    delay(1000); // Milisaniye cinsinden bekleme süresi komutudur. 1000ms=1sn  
}
```

- Kod yazma işlemi tamamlandıktan sonra üst sağ kısmında bulunan tik(🔍) 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok(➡) 'a tıklayarak kodumuzu Arduino' ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, seri port ekranını açarak ekran çıktısını görebiliriz.

Ekran Çıktısı:

The screenshot shows the Arduino IDE interface. The code editor on the left contains the following code:

```
1 void setup() {  
2   Serial.begin(9600); /* Haberleşmeyi başlatmak için 9600 baud hızı kullanıyoruz */  
3 }  
4 void loop() {  
5   Serial.println("Arduino ile Programlama Öğreniyorum");  
6   /*  
7   Serial.println: Alt satıra yazı yazmak için kullanılır.  
8   Serial.print: Aynı satırda yazdırmak için kullanılır.  
9   */  
10  delay(1000); // Milisaniye cinsinden bekleme süresi  
11 }  
12
```

The Serial Monitor window on the right shows the output of the code:

```
Arduino ile Programlama Öğreniyorum  
Arduino ile Programlama Öğreniyorum  
Arduino ile Programlama Öğreniyorum  
Arduino ile Programlama Öğreniyorum
```

The status bar at the bottom indicates that the upload was successful: "Yükleme tamamlandı." It also shows the program size: "Çalışmanız programın 1666 bayt (5 %) saklandı." and the global variable memory usage: "Global değişkenler belleğin 224 byte kadarını (10%) kullanıyor. Diğer değişkenler için 1024 byte yer kalıyor."

Potansiyometre Kullanımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Analog pin
- Potansiyometre
- Fonksiyonlar
- Değişken tipi

Gerekli Malzemeler

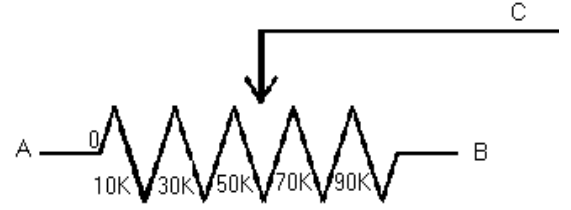
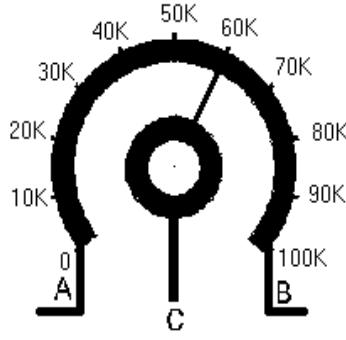
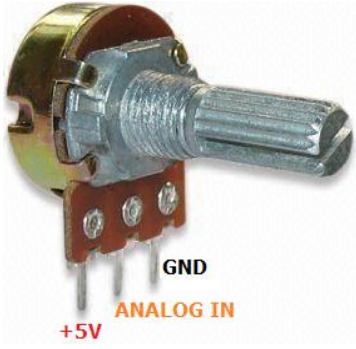
- Arduino UNO
- Breadboard
- USB kablo
- Potansiyometre
- Jumper Kablo

Amaç

- Potansiyometreyi kontrol etmek

Kazanım

- Potansiyometre kavramını öğrenilir.
- Günlük hayatta kullandığımız ayar düğmelerinin potansiyometre mantığıyla çalıştığını öğrenir.
- int veri tipi ile tam sayı tipi değişken oluşturur.
- void setup ve void loop fonksiyonlarını öğrenir.



Potansiyometre ayarlı direnç ya da sürgülü reosta olarak bilinen bir devre elemanıdır. İç yapısında bir direnç malzemesi bulunur. Bu sayede istenilen seviyeye getirilebilir ve bize sinyal değerleri verebilir. Potansiyometre, günlük kullandığımız birçok cihazda var olan ve elle ayarlama yapabildiğimiz bir direnç türüdür.

Örneğin; araç radyolarında sesi arttırmak veya azaltmak için çevirdiğimiz düğme bir potansiyometre dir.

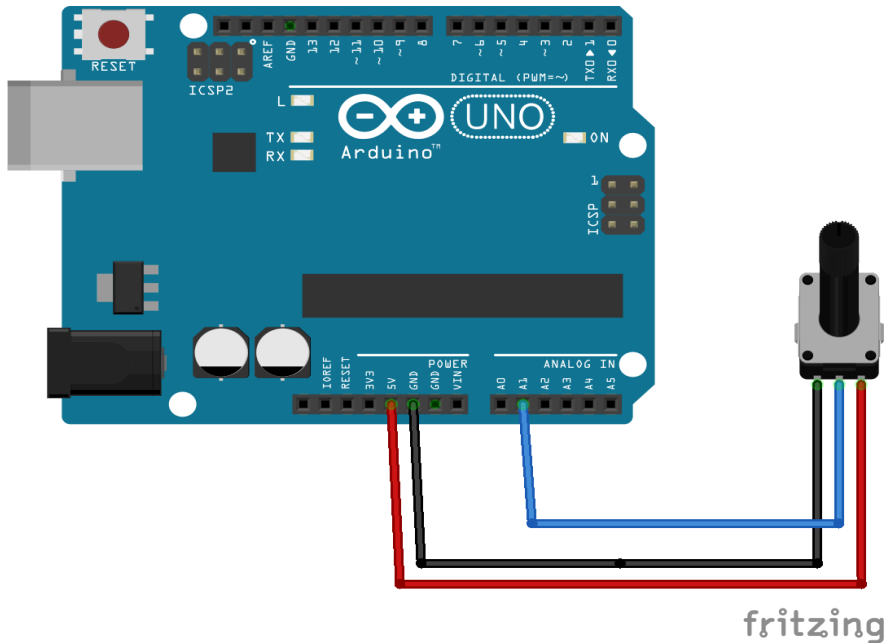
Bağlantısı; A Bacağı 5V, B Bacağı GND, C Bacağı Analog Sinyal

Analog pinlere bağlanan C Bacağından 0-1023 arasında değerler elde edilir.

Çalışma prensibi; A veya B uçlarından herhangi birine gerilim diğer uca ise toprak bağlantısı yapılıyor. Hareketli ibreye bağlı olan w ucu ise devremize bağlanıyor. Böylelikle ibreyi gerilim kaynağı bağlı olan uca doğru çevirdiğimiz zaman direnç değerimiz azalıyor ve devremize giden akım artmış oluyor. Aksi yönde yani toprak bağlı uca doğru çevirdiğimizde ise gerilim bağlı ucla devremiz arasında yüksek direnç düşük akım oluşuyor. Diğer yandan toprakla devre arasında ki dirençte azalmış, devrenin ucu ibreyi bu yöne doğru çevirdikçe gittikçe topraklanmış oluyor.

Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra potansiyometre kullanımı projemizin kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.
 - Kodların yanında her bir satır için açıklamalar bulunmaktadır.

```
const byte pot_pin=A1;
int pot_deger=0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  pot_deger=analogRead(potansiyometre_pin); //Analog pinden gelen değeri okutulur. 10 bitlik veri getirir.
  int deger=map(pot_deger,0, 1023, 0, 100); /* Map fonksiyonu ile potansiyometreden 0 ile 1023 arasında okunan analog değeri 0 ile 100 arasında indirgemiş olduk. 8 bitlik veriye dönüştürme işlemi yapıyoruz*/
  Serial.print("Potansiyometre Değeri= ");
  Serial.println(deger);
}
```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino'ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, seri port ekranını açarak, potansiyometrenin ayar kolunu çevirebiliriz. Böylece potansiyometrenin değerlerini anlık olarak ekranda görebiliriz.

Potansiyometre ile LED Kontrolü

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Olayı başlatma
- Fonksiyonlar
- Değişken tipi
- Analog pin
- Potansiyometre
- LED
- Direnç
- Delay

Gerekli Malzemeler

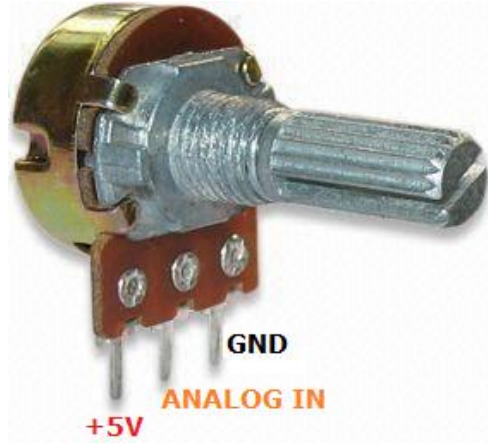
- Arduino UNO
- Breadboard
- USB kablo
- LED
- Direnç
- Potansiyometre
- Jumper Kablo

Amaç

- Potansiyometre ile LED parlaklığını kontrol etmek

Kazanım

- Kodlama yardımıyla devreyi çalıştırmayı öğrenilir.
- Potansiyometre kavramını öğrenilir.
- Yanan LED'in parlaklığını azaltıp artırır.
- int veri tipi ile tam sayı tipi değişken oluşturur.
- void setup ve void loop fonksiyonlarını öğrenir.
- analogRead, delay ve analogWrite kavramlarını öğrenir.



analogRead() Fonksiyonu:

analogRead() fonksiyonu Arduino' ya Analog değerler girmemizi sağlar. Bu fonksiyon, okuduğu değere karşılık 0 ile 1023 arasında bir sayı üretir.

map() Fonksiyonu:

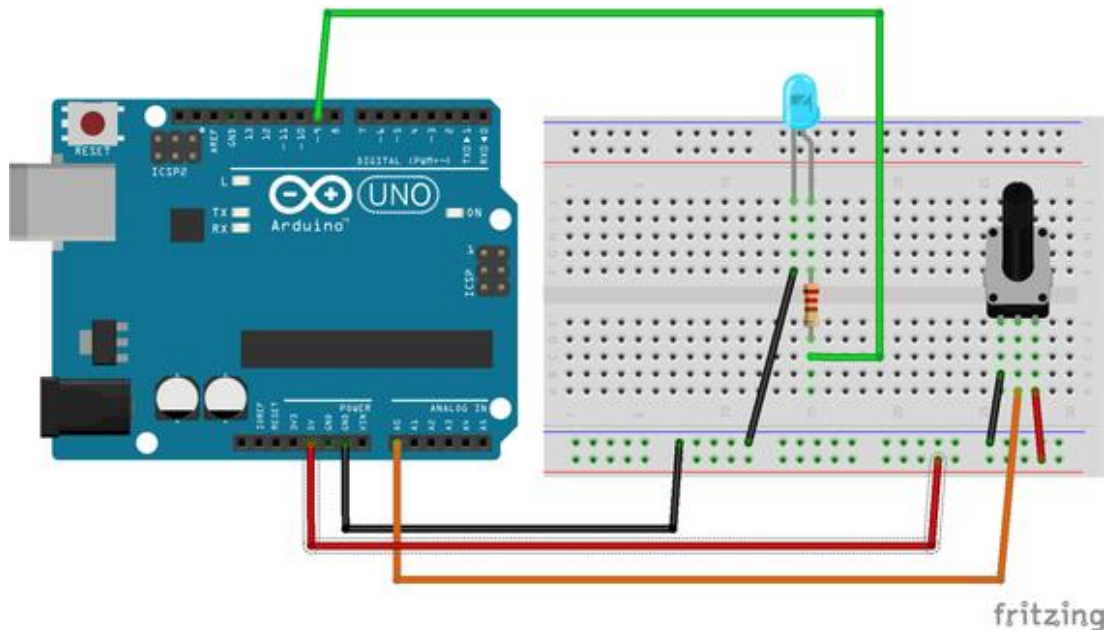
Bu fonksiyon bir oranlama fonksiyonudur. Belirli bir aralıktaki sayının, başka bir aralıkta hangi sayıya karşılık geldiğini hesaplar. Map() fonksiyonuna parametre olarak beş değer girilir.

map (sayi, aralik_1_altdeger, aralik_1_ustdeger, aralik_2_altdeger, aralik_2_ustdeger);

analogWrite Fonksiyonu:

Bu fonksiyon ise 0-255 arası arasında bir değeri çıkışlardan verir.

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra potansiyometre ile LED parlaklığı ayarlama projemizin kodları yazmaya başlayabiliriz.
 - Arduino IDE kod satırları aşağıda gösterilmektedir.

```
int led=10;
int pot=A0;
int potDeger;
void setup(){
}
void loop(){
  potDeger=analogRead(pot);
  potDeger=map(potDeger, 0, 1023, 0, 255);
  analogWrite(led, potDeger);
  delay(10);
}
```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino'ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, breadboard üzerindeki LED'in yanmaya başlayacak. Potansiyometrenin ayar kolunu çevirdiğimizde LED'in parlaklığının değiştiğini gözlemleyebiliriz.

RGB LED Kullanımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- RGB LED
- Potansiyometre
- Olayı başlatma
- Fonksiyonlar
- Kontrol yapıları
- PinMode
- Değişken tipi
- Delay

Gerekli Malzemeler

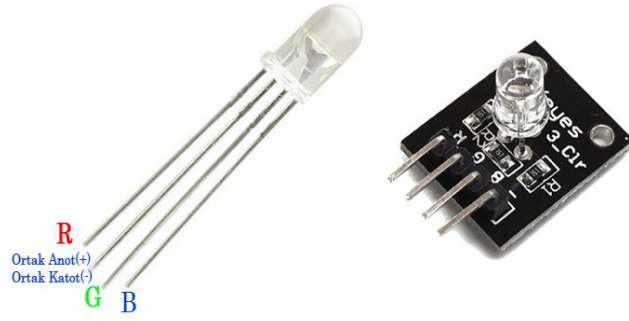
- Bilgisayar
- Arduino UNO
- Breadboard
- USB kablo
- RGB LED
- Potansiyometre
- Jumper Kablo

Amaç

- RGB LED kontrolünü öğrenebilmek.

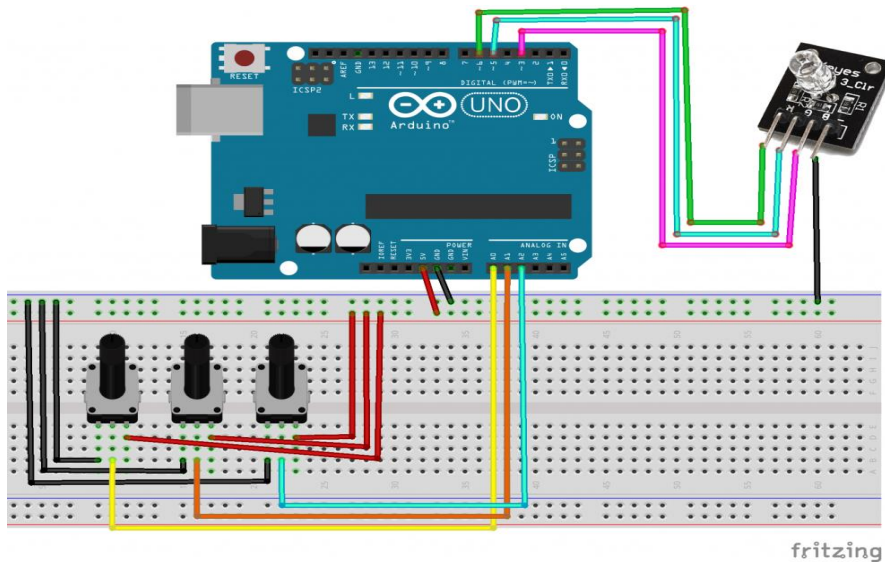
Kazanım

- RGB LED kavramını öğrenir.
- RGB LED'i potansiyometre ile kontrol edebilir.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.
- Değişken oluşturmaya öğrenir.



Potansiyometreden alınan analog değerler “map” komutu ile 0-255 arasındaki PWM sinyallerine dönüştürülerek, Arduino üzerindeki PWM pinlerine bağlı RGB LED pinlerine aktararak RGB LED kontrol edilebilir. Bu sayede kırmızı, yeşil ve mavi dışında tüm ara renkleri de potansiyometreleri hareket ettirilerek ara renkler elde edilebilmektedir.

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın “Araçlar” sekmesinden “Kart” kısmına gelerek “Arduino/Genuino Uno” seçeneğini seçiyoruz. Son olarak da “Araçlar” sekmesinden “Port” kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra RGB LED'i potansiyometre ile kontrol edebileceğimiz kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.

```

int r=3;
int g=5;
int b=6;
const byte potRed=A0;
const byte potGreen=A1;
const byte potBlue=A2;
int potDegerRed;
int potDegerGreen;
int potDegerBlue;
void setup() {
  pinMode(r,OUTPUT);
  pinMode(g,OUTPUT);
  pinMode(b,OUTPUT);
}
void loop() {
  int potDegerRed=map(analogRead(potRed, 0,1023, 0, 255));
  int potDegerGreen=map(analogRead(potGreen, 0,1023, 0, 255));
  int potDegerBlue=map(analogRead(potBlue, 0,1023, 0, 255));
  analogWrite(r, potDegerRed);
  analogWrite(g, potDegerGreen);
  analogWrite(b, potDegerBlue);
  delay(100);
}

```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino' ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, her bir potansiyometre bir rengi temsil ediyor olacak. Yani kırmızı, yeşil ve mavi renkleri potansiyometrelere tanımlanmış olacak. Böylece potansiyometreleri çevirdikçe LED imiz, bu 3 ana renk ile ışık teorisine uygun olarak, olabilecek bütün renklerde yanmaya başlayacak.

LDR Kullanımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- LDR
- Olayı başlatma
- Fonksiyonlar
- Kontrol yapıları
- PinMode
- Değişken tipi
- Delay

Gerekli Malzemeler

- Bilgisayar
- Arduino UNO
- Breadboard
- USB kablo
- LDR
- Direnç
- Jumper Kablo

Amaç

- LDR ışık şiddetini ölçme işlemini öğrenmek

Kazanım

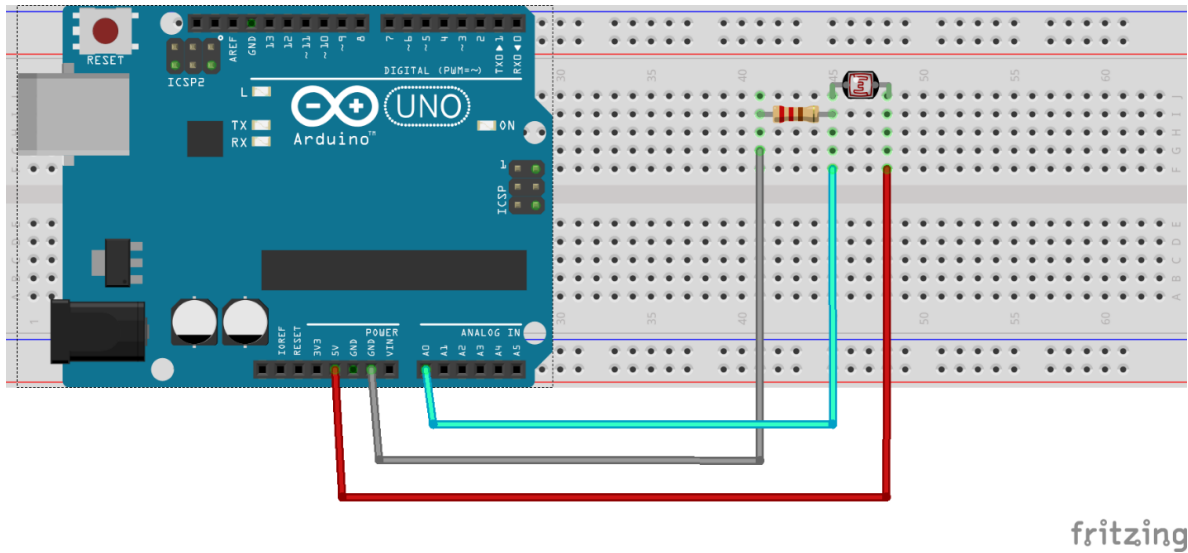
- LDR LED kavramını, kontrol yapılarını, fonksiyonları ve değişken tipi oluşturmayı öğrenir.
- Işık şiddetini ölçebilir.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.



Işığa bağlı olarak değerleri değişen dirençlere fotodirençler denilmektedir. LDR(Light Dependent Resistor)'nin üzerine düşen ışık şiddeti ile biz verdiği direnç değeri ters orantılıdır. Yani ışık şiddeti; arttığında direnç düşer, azaldığında ise direnç artar.

LDR, ışık ile kontrol gerektiren projelerde ve sistemlerde basitçe kullanılabilecek bir sensördür. Sokak lambaları, gece lambaları, kumanda sistemleri günlük kullanımlara örnek olarak verilebilirler.

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra LDR'nin verdiği değerleri seri port ekranında görebilmek için gereken kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.

```
int Fotores = A0;

void setup() {
  Serial.begin(9600);
}

void loop () {
  int Ldr=analogRead(Fotores);
  Serial.println(Ldr);
  delay (1000);
}
```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino' ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, seri port ekranı üzerinden LDR sensörden gelen değerleri görebiliriz. Böylece karanlık ve aydınlıkta verdiği değerleri buradan görebiliriz.

LDR ile Sokak Lambası Yapımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- LDR
- Olayı başlatma
- Fonksiyonlar
- Kontrol yapıları
- PinMode
- Değişken tipi
- Delay

Gerekli Malzemeler

- Bilgisayar
- Arduino UNO
- Breadboard
- USB kablo
- LED
- LDR
- Direnç
- Jumper kablo

Amaç

- LDR ışık şiddetini ölçme işlemini öğrenmek
- Sokak lambalarının çalışma mantığını öğrenmek

Kazanım

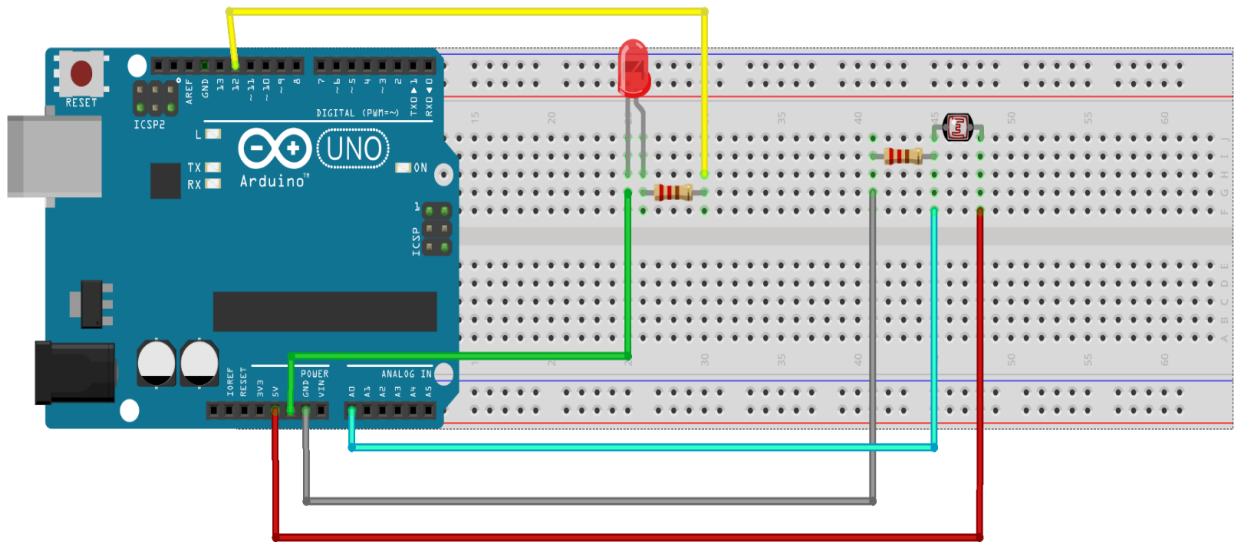
- LDR LED kavramını, kontrol yapılarını, fonksiyonları ve değişken tipi oluşturmayı öğrenir.
- Işık şiddetini ölçebilir.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.
- Günlük hayat ile bağlantı kurar.



Işığa bağlı olarak değerleri değişen dirençlere fotodirençler denilmektedir. LDR(Light Dependent Resistor)'nin üzerine düşen ışık şiddeti ile biz verdiği direnç değeri ters orantılıdır. Yani ışık şiddeti; arttığında direnç düşer, azaldığında ise direnç artar.

LDR, ışık ile kontrol gerektiren projelerde ve sistemlerde basitçe kullanılabilecek bir sensördür. Sokak lambaları, gece lambaları, kumanda sistemleri günlük kullanımlarımıza örnek olarak verilebilirler.

Devre Şeması:



Uygulama:

- Devre kurulumu bittikten sonra Arduino IDE programını açabiliriz.
- Arduino'yu USB kablo ile bilgisayarımıza bağlıyoruz. Programımızın "Araçlar" sekmesinden "Kart" kısmına gelerek "Arduino/Genuino Uno" seçeneğini seçiyoruz. Son olarak da "Araçlar" sekmesinden "Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra "Sokak Lambası" projemizin kodları yazmaya başlayabiliriz.
- Arduino IDE kod satırları aşağıda gösterilmektedir.

```

int led=12;

int ldr=A0;

int deger=0;

void setup() {
    pinMode(led,OUTPUT);
    pinMode(ldr,INPUT);
    Serial.begin(9600);
}

void loop() {
    deger=analogRead(ldr);
    Serial.print("ldr led değeri: ");
    Serial.println(deger);
    if(deger<30){
        digitalWrite(led,HIGH);
    }else{
        digitalWrite(led,LOW);
    }
    delay(1000);
}

```

- Kod yazma işlemi tamamladıktan sonra üst sağ kısmında bulunan tik() 'e tıklayarak yazılan kodu kontrol edebiliriz. Son olarak da yine sağ üst kısımdaki ok()' a tıklayarak kodumuzu Arduino' ya yüklüyoruz.
- Yükleme işlemi tamamlandıktan sonra, breadboard üzerindeki LED eğer ortam aydınlıkta sönecek, karanlıkta da yanmaya başlayacak. Böylece sokak lambalarında da olduğu gibi karanlıkta otomatik olarak yanan ve aydınlık ortamda sönen bir ışık sistemi oluşturmuş oluruz.