

ARDUİNO VE ROBOTİK KODLAMA ETKİNLİK KİTABI 2

Bekir BAŞKURT

Cebrail AKAY

2025



Bu kitap, Arduino ve robotik kodlama dünyasına adım atmak isteyen herkes için kapsamlı bir rehber niteliğindedir. Temel elektronik bilgileri, mikrodenetleyiciler, devre simülasyonları ve Arduino platformları gibi konuları içeren bu eser, hem teorik hem de pratik bilgilerle donatılmıştır.

Kitabımız, adım adım ilerleyen etkinliklerle okuyucuların kendi projelerini geliştirmelerine olanak tanır. Her bölümde yer alan uygulamalı örnekler ve projeler sayesinde, okuyucular öğrendiklerini hemen pratiğe dökebilirler.

Bu kitabı hazırlarken, robotik kodlama alanında kendini geliştirmek isteyen gençlerimize ve yetişkinlere ilham vermeyi amaçladık. Umarız ki, sizlere yeni ufuklar açar ve projelerinizde size rehberlik eder.

ALGORİTMA

ALGORİTMA NEDİR ?

Bir sorunu çözmek veya belirlenmiş bir amaca ulaşmak için tasarlanan yola, takip edilen işlem basamaklarına algoritma denir. Amaca ulaşmak için işlenecek çözüm yolları ve sıralamaları belirlenir ve algoritma bu sırayı takip ederek en mantıklı çözüme ulaşır.

Algoritma Niçin Kullanılır? Ne İşe yarar?

Algoritma matematikte bir sonuca ulaşmak için takip edilmesi gereken işlem basamaklarını belirlemede kullanılır. Örneğin; 3 ve 5 sayılarının ortalamasını bulmak istediğinizde önce $(3+5)$ işlemini yaptıktan sonra sırada “çıkan sonuç/2” işlemini yapmanız gerekmektedir.

Algoritma ve Programlama Bağlantısı




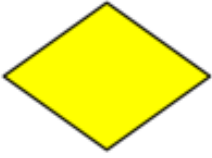
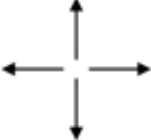


Tüm programlama dillerinin temelinde algoritma vardır. Algoritmalar, programlama dillerinin vasıtasıyla uygulanabilirler. Programda kullanılan dil ve kullanım alanı ne olursa olsun algoritması olmayan bir program yoktur.

Algoritmada 3 temel bileşen vardır:

- **Değişkenler:** Dışarıdan girilen ve bizim oluşturduğumuz değerleri tutan elemanlardır.
- **Algoritma:** Kısaca gerekli adımların mantıksal bir sıra ile yazılmasıdır
- **Akış Diyagramı:** Birbirine oklar gösterilerek algoritmaları ve şemaları birbirine bağlayan kutulara denir. Kısaca çözümün şekiller ile tasviridir.

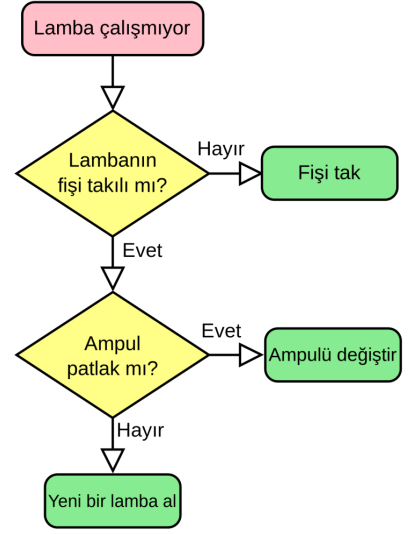
ALGORİTMA

Akış Şeması Şekilleri

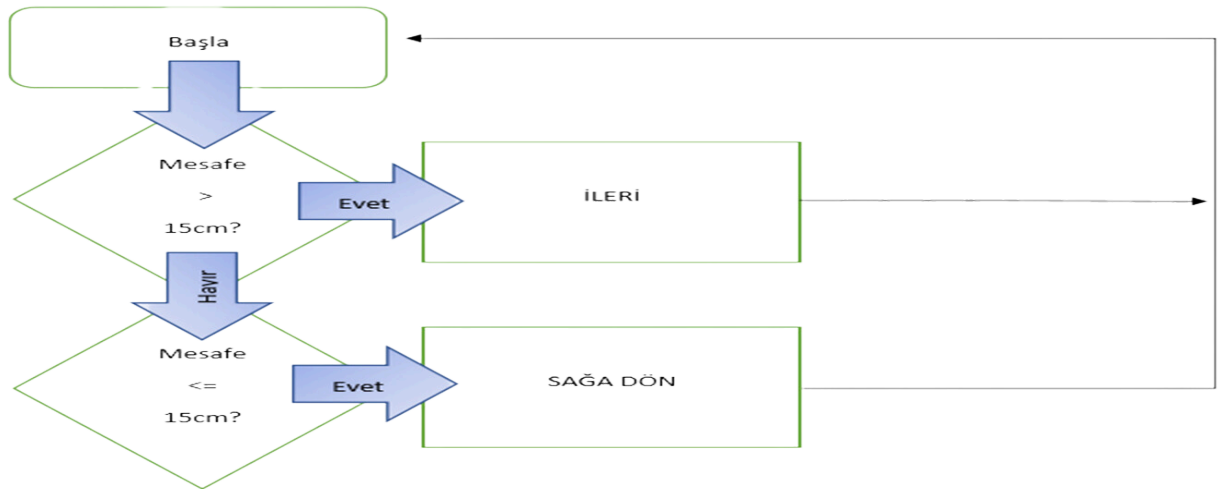
	Programın başlangıç ve bitişi için kullanılır.
	Bilgi giriş çıkışı için kullanılır.
	Aktarma, aritmetik hesaplama, işlem
	Karar alma için kullanılır.
	Birleştirme çizgileri
	Yazdır
	Bağlantı

ALGORİTMA

Örnek 1 : Lambanın çalışıp çalışmadığını kontrol eden algoritma.



Örnek 2 : Engelden Kaçan Robot uygulamasındaki algoritmayı inceleyelim. Robottaki sensörün gördüğü mesafe değeri 15 cm'den büyük ise ileri gitsin, 15 cm'ye eşit veya 15 cm'den küçük ise robot sağa dönsün.



ALGORİTMA

Örnek 3: Verilen iki sayıyı toplamını veren algoritma

- BAŞLA
- SAYI 1 GİR
- SAYI2 GİR
- TOPLAM= SAYI1+SAYI2
- BİTİR



Örnek 4: Girilen 2 sayıdan büyük olanını ekranda gösteren algoritma

- BAŞLA
- SAYI1 GİR
- SAYI2 GİR
- EĞER SAYI1 >SAYI2 İSE YAZ SAYI1
- DEĞİLSE YAZ SAYI2
- BİTİR

Algoritma

Örnek 5: Çay Demleme Algoritması

- 1: Başla
- 2: Çaydanlığa su doldur.
- 3: Ocağı yak.
- 4: Çaydanlığı ocağa koy.
- 5: Bir süre suyun kaynamasını bekle.
- 6: Su kaynadı mı? Kaynamadıysa(şart ifadesi) 5. adıma git.**
Kaynadıysa(şart ifadesi) 7. adıma git.
- 7: Çayı demle.
- 8: Çaydanlığa su ilave et.
- 9: Çaydanlığı tekrar ocağa koy.
- 10: Bir süre suyun kaynamasını bekle.
- 11: Su kaynadı mı? Kaynamadıysa(şart ifadesi) 10. adıma git.**
Kaynadıysa(şart ifadesi) 12. adıma git.
- 12: Bir süre çayın demlenmesini bekle.
- 13: Çay demlendi mi? Demlenmediyse 12. adıma git. Demlendiyse 14. adıma git.**
- 14: Çayı bardaklara koyup servis et.
- 15: **Bitir.**

ALGORİTMA OLUŞTURURKEN DİKKAT EDİLMESİ GEREKEN HUSUSLAR

• Etkinlik

Bilgisayarlar düşünemez. Bu yüzden algoritmanın her adımı anlaşılır, basit ve kesin bir biçimde ifade edilmiş olmalıdır. Yorum gerektirmemeli ve belirsiz ifadelerle sahip olmamalıdır. Gereksiz tekrarlarda bulunmayan diğer algoritmalar içerisinde de kullanılabilir olmalıdır. Bir çocuğa, ilk defa öğreneceği birşeyi öğretirken ki hassasiyet ve basitliği örnek alabilirsiniz.

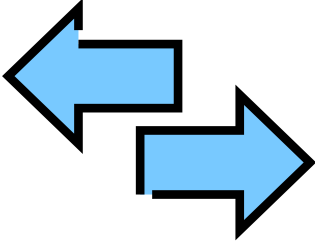


• Sonluluk

Her algoritmanın bir başlangıç noktası, belirli işlem adımı ve bir bitiş noktası içermelidir. Sonsuz döngüye girmemelidir. Başlatılan bir iş kesinlikle neticelendirilmelidir.

• Kesinlik

İşlem sonucu kesin olmalı, aynı veri için her yeni çalıştırmada aynı sonucu üretmelidir.



• Giriş – Çıkış

Algoritma giriş (üzerinde işlem yapılacak değerler) ve çıkış (yapılan işlemler neticesinde üretilen sonuç değerler) değerlerine sahip olmalıdır.

• Başarı – Performans

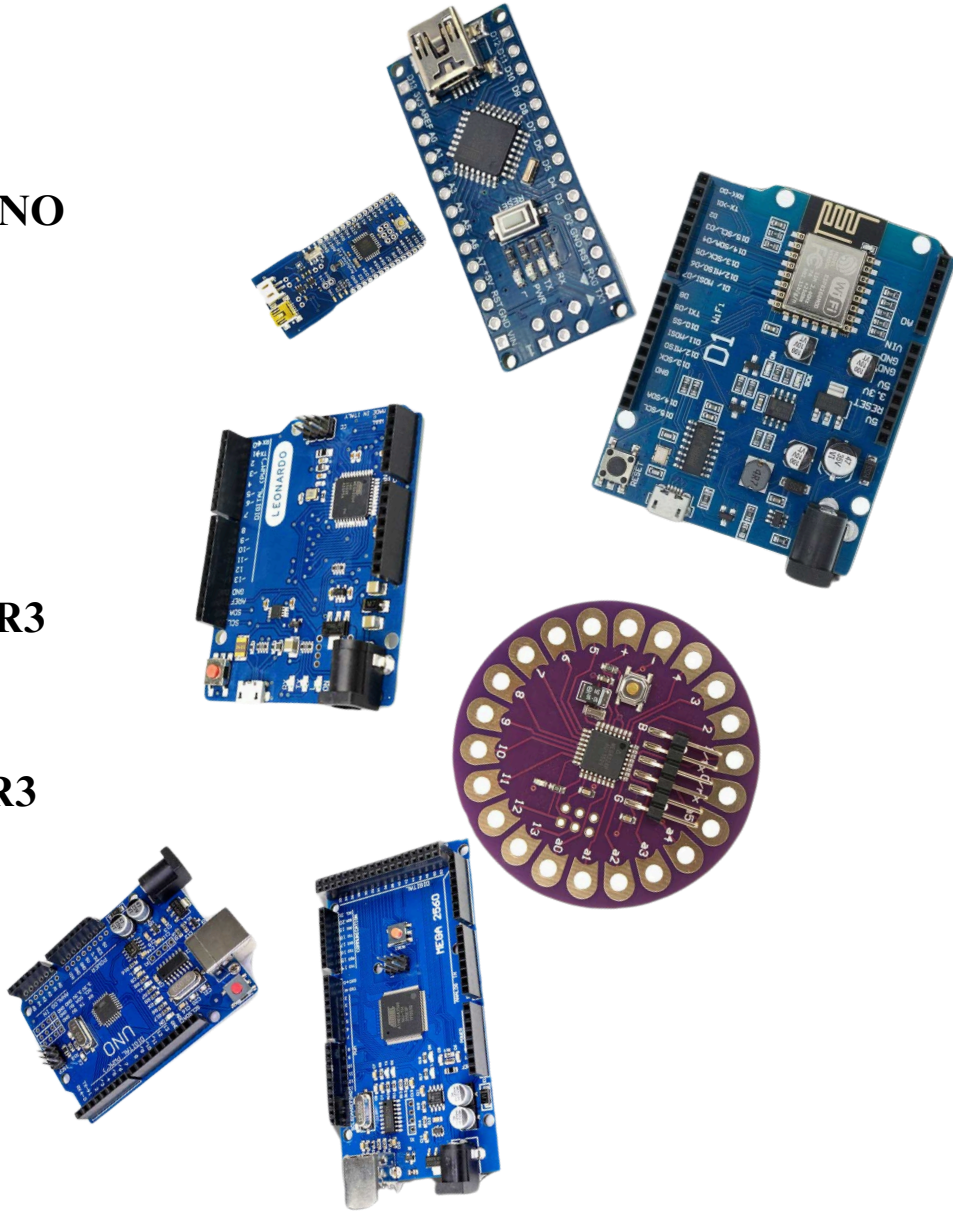
Amaç donanım gereksinimi (bellek kullanımı gibi), çalışma süresi gibi performans kriterlerini dikkate alarak yüksek başarımlı programlar yazmak olmalıdır. Gereksiz kod yığınlarından kurtulup en sade kodları tercih etmek önemlidir.

ARDUİNO

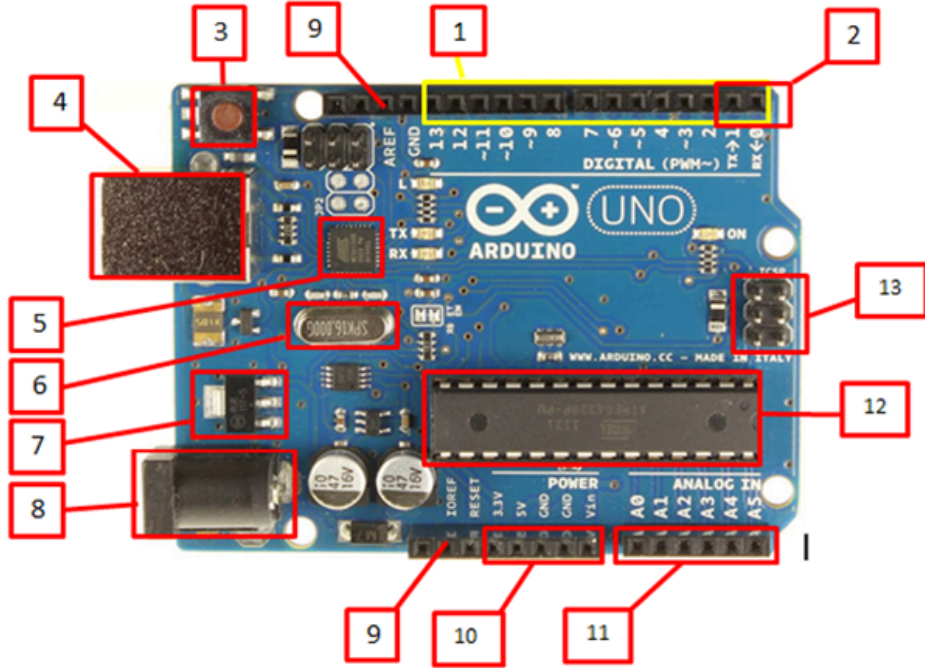
Arduino, elektronik ile ilgili olan her insanın kolayca kullanabilmesi için geliştirilmiş açık kaynaklı bir **mikrokontrolcü platformudur**. Arduino kullanarak çeşitli sensörlerden gelen sinyalleri okuyabilir, ışık yakıp söndürebilir, motor çalıştırabilir; kısacası aklınıza gelebilecek tüm elektronik uygulamaları yapabilirsiniz.

Arduino Çeşitleri

- **Arduino / Genuino UNO R3**
- **Arduino Leonard**
- **Arduino Due**
- **Arduino Yun**
- **Arduino Esplora**
- **Arduino Micro**
- **Arduino Mega ADK R3**
- **Arduino Ethernet**
- **Arduino Robot**
- **Arduino Mega 2560 R3**
- **Arduino Mini**
- **Arduino Nano**
- **Arduino Pro Mini**
- **Arduino Fio**
- **Arduino Pro**



ARDUİNO



1. **Dijital Giriş/ Çıkış Pinleri** (Yanında ~ işareti olan pinler PWM çıkışı olarak kullanılabilir.) PWM digital çıkışlar olmasına rağmen analog pine ihtiyaç duyduğumuzda kullanabileceğimiz çıkışlardır.

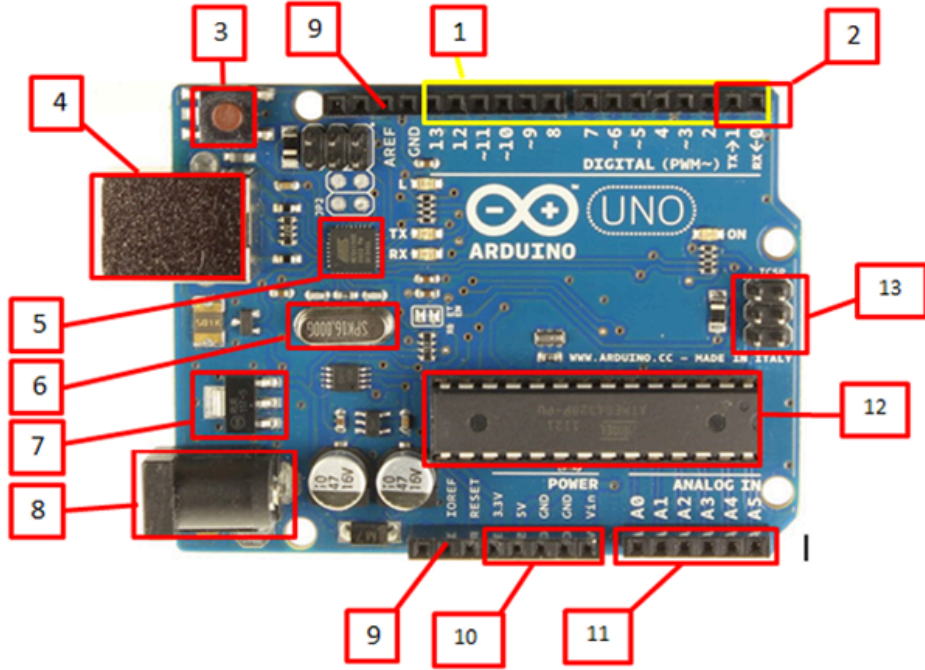
2. **TX/RX Pinleri:** Seri haberleşme pinleridir USB bağlı iken ihtiyaç duyulmaz. Bluetooth sensörü kullanılacaksa tercih edilir.

3. **Reset Butonu:** Mikrodenetleyiciyi sıfırlamak için kullandığımız buton.

4. **USB Jaki:** Bilgisayar ile kart arasında ki USB kablosunun bağlantı girişi.

5. **USB Seriport Çevirici (Haberleşme Çipi):** Seri port haberleşme çipi

ARDUİNO



6.16 Mhz Kristal: Mikro denetleyicinin çalışma osilatörü

7.Voltaj Regülatörü: Voltaj düzenleyici entegre

8.Power Jaki (7-12 V DC): Harici enerji girişi (pil ya da enerji kaynağı)

9.AREF Pini: Bu pin analog pinlerdeki voltaj ölçümü için referans pinidir.

10.Power Pinleri:Mikrodenetleyiciden 5V, 3.3V ve (-) eksi uçlarının bağlandığı noktalar

11.Analog Girişler: Analog olarak dışarıdan mikro işlemciye bilgi girişi yapılan uçlar

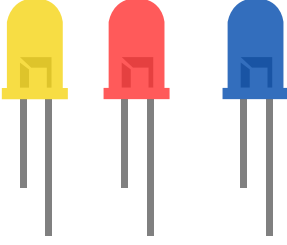
12.Mikrodenetleyicinin işlemcisi

13.Programlama Pinleri: Mikrokontrolcüyü doğrudan programlamak için kullanılan pinlerdir. Bu pinler sayesinde Arduino kartına bir programcı yani başka bir kart bağlayarak mikrokontrolcüye doğrudan erişim sağlanır ve kod yüklemesi yapılabilir.

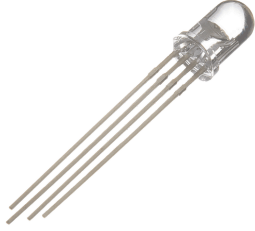
ANALOG VE DİJİTAL PİNLER

- Pinler, Arduino'nun kendisine takılan bileşenlerine bağlanmasını sağlıyor.
- Dijital ve Analog Pinler arasındaki fark ise verilen sinyal tipidir.
- Analog Sinyal 0-255 arasındaki değerlere sahip olabilirken, dijital için 1 ve 0 vardır

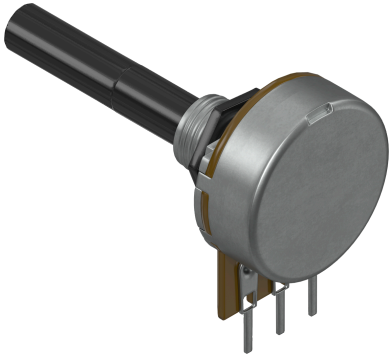
DEVRE ELEMANLARI



Led: anot ve katot olmak üzere iki farklı bacağı olan, bunlardan anodu pozitif gerilime yani + uca, katot ise negatif gerilime yani – uca ya da devredeki toprak hattına (GND, Ground) bağlanan devre elemanıdır. Parlaklık ve ömür seviyesi de üzerinden geçen akıma göre değişir.



Rgb Led: Kırmızı, yeşil, mavi (Red, Green, Blue) renklerdeki 3 adet ledi içerisinde barındıran Ledlere RGB led denir. Bu üç rengin farklı oranlarda karıştırılması ile her rengi elde edebiliriz.



Potansiyometre(ayarlı direnç), direnç değerini ayarlamak ve böylece bir devredeki elektrik akışını kontrol etmek için kullanılan pasif bir elektronik bileşendir..

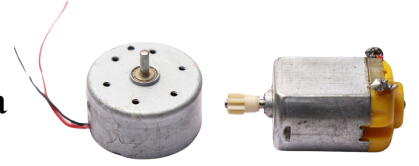


Servo motor, açısal veya doğrusal pozisyonun hassas bir şekilde kontrol edilmesini sağlayan bir motor türüdür. Servo motorlar, belirli bir pozisyonda sabit kalabilme ve hassas hareket gerektiren uygulamalarda sıkça tercih edilir. 180 derecelik dönüş kapasitesi vardır.

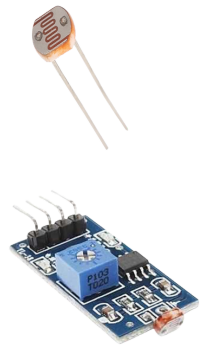
Doğru akım motoru, dünya literatüründe DC motor olarak bilinen elektrikli motorlardır.

Fırçalı ve fırçasız olmak üzere iki ana kategoriye ayrılır:

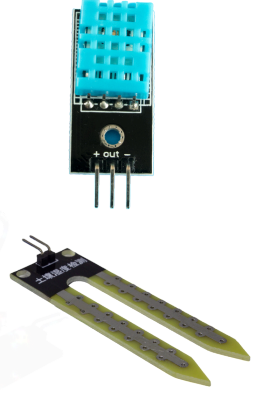
Fırçalı DC Motorlar: Daha eski bir teknoloji olup, fırçalar aracılığıyla akım iletilir. Fırçalar zamanla aşınabilir ve değiştirilmesi gerekebilir.



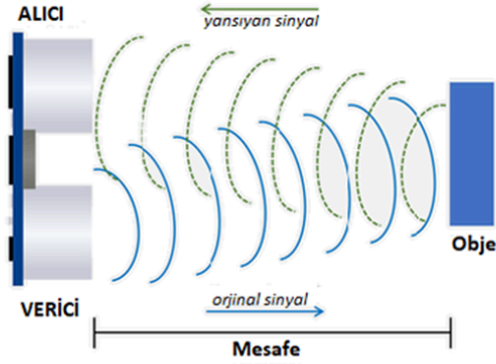
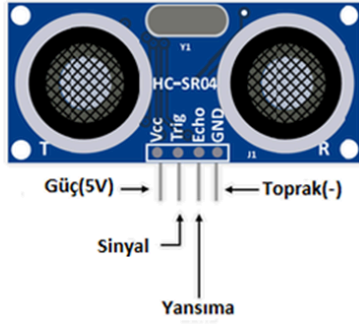
Fırçasız DC Motorlar: Daha yeni bir teknoloji olup, elektronik hız kontrol devreleri (ESC) kullanılarak çalışır. Daha verimli ve uzun ömürlüdür. Dronelar ve model uçaklarda yaygın olarak kullanılır.



Işık Sensörü (LDR): Foto dirençler, üzerlerine düşen ışık şiddetiyle ters orantılı olarak dirençleri değişen elemanlardır. Foto direnç, üzerine düşen ışık arttıkça direnç değeri lineer olmayan bir şekilde azalır. LDR'nin aydınlıkta direnci minimum, karanlıkta maksimumdur. Hem AC devrede, hem DC devrede aynı özellik gösterir

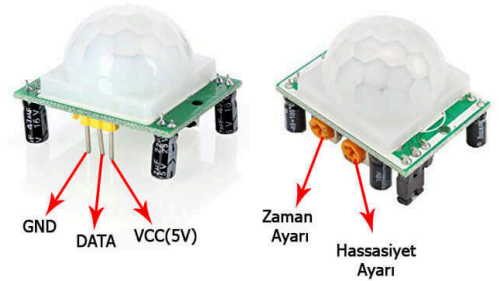


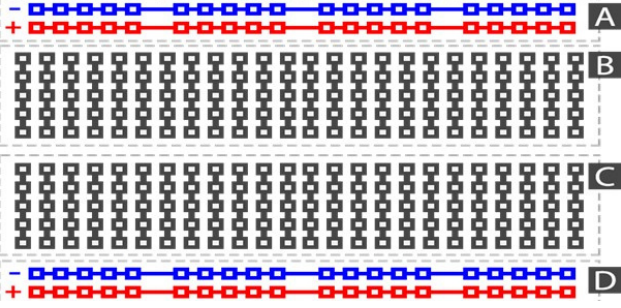
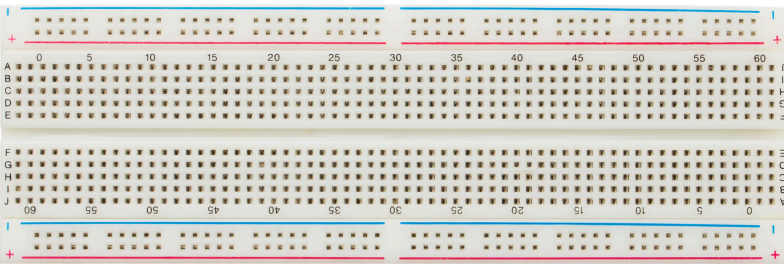
Nem sensörleri, iki adet sıcaklık sensöründen meydana gelir. Bu sensörler ortamdaki nem oranını oluşturan elektrik akımına göre ölçer ve hesaplar. Sensörlerden bir tanesi kuru nitrojen içinde yer alırken diğeri ortam sıcaklığını ölçecek şekilde yerleştirilmiştir. Aradaki fark ortamın nem değerini verir.



Mesafe sensörü: Ultrasonik ses dalgaları, insan kulağının duyamayacağı frekans bandına sahiptir. Bu ses dalgalarını kullanan sensör ile nesnelere herhangi bir temas sağlamadan mesafe ölçümleri yapabiliriz. Bunun için, hareket problemlerinde olduğu gibi " $X(\text{yol})=V(\text{hız}) \cdot t(\text{zaman})$ " formülüne göre çalışmaktadır. Ses dalgalarının belirli koşullardaki hızının biliniyor olmasından faydalanarak, ses dalgasının gidip, daha sonra geri gelmesi arasında geçen sürenin yarısı ile hızının çarpımı sonucu bize mesafeyi vermektedir.

Pır hareket sensörü: "Passive Infrared Sensor" ifadesinin kısaltması olan PIR sensörler çokça tercih edilen pratik bir hareket sensörüdür. Türkçe anlamı "Pasif Kızılötesi Sensör" olan PIR sensörler bir nesneden yayılan kızılötesi ışınları algılayabilen elektronik bir cihazdır. Ortamda meydana gelen hareketlilikleri algılar ve bağlı bulunduğu sisteme uyarı göndererek, gerekli tepkiyi ortaya koyar.

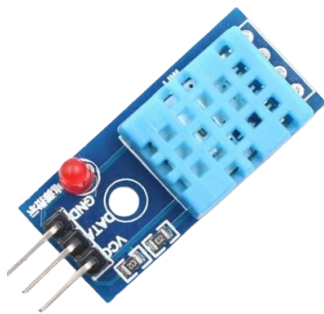
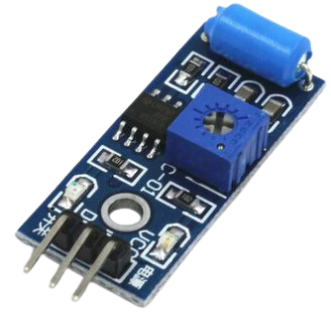




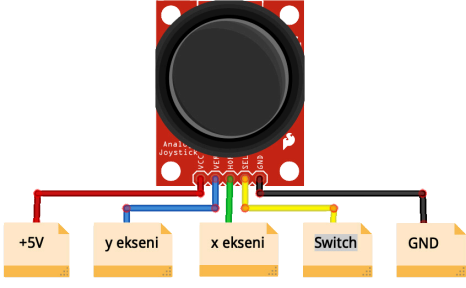
Breadboard İç Bağlantı Şekli

Breadboard, üzerinde elektronik devre tasarlayabileceğimiz bir araçtır. Kurduğumuz devreleri birbirlerine lehimlemeden kolaylıkla test etmemizi sağlar. Tasarladığımız devreleri baskı devre veya delikli plakette üzerine aktarmadan önce denememize olanak sağlar.

SW-420 Hareket Titreşim Sensörü: Harekete karşı tetik üreten bir modüldür. Çok yaygın alanda kullanabilirsiniz. Titreşim, hırsız alarmı, araba alarmı, deprem alarmı, motosiklet alarmı gibi alarm uygulamalarında hareketin varlığını algılayabileceğiniz bir üründür.



DHT11, sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimdir. Yüksek güvenilirlikte ve uzun dönem çalışmalarda dengelidir.



Joystick, iki eksende hareket etmeyi saęlayan iki potansiyometre bulunduran ve dikey hareketinde ise buton görevi gören bir modüldür. Joystick'leri genellikle oyun konsollarında ve uzaktan kumandalı araçlarda görürüz.

**ETKİNLİKLERE
BAŞLAYALIM MI ?**



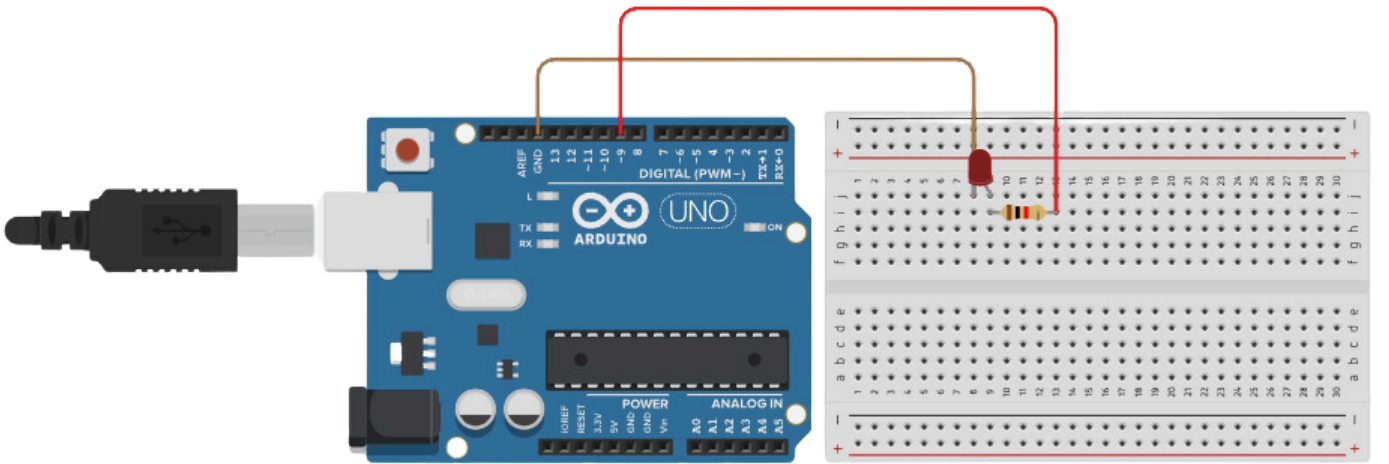
ETKİNLİK 1

**Birer saniye
aralıklarla Led
yakıp söndürelim !**

Kullanılacak Malzeme

- Arduino uno
- BreadBord
- Jumper kablo
- Led
- Direnç

Devre Şeması



Mblock Kodları



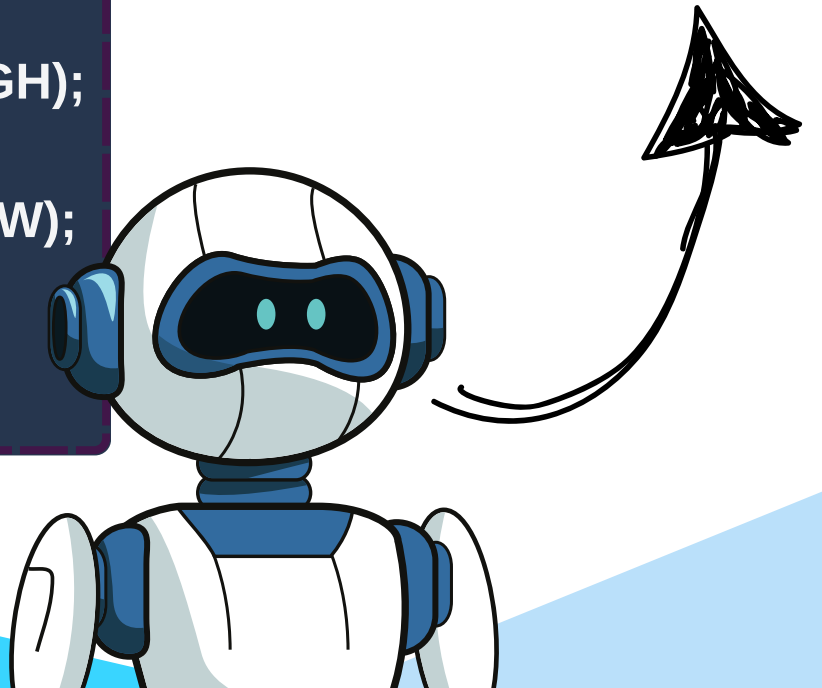
Metin Tabanlı Kodları

```
int ledPin = 10;

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

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

LED, ışık yayan diyot anlamına gelir. Diyot ise yarı iletken malzemeden oluşur. Yarı iletken malzemenin katmanlarından birinde fazla elektron olur diğer katman ise elektronlardan arındırılır. Katmanlar arası elektron seviyelerindeki fark, elektronların bir katmandan diğerine hareket etmesini sağlayarak elektronik uyarım yoluyla ışık yayar.



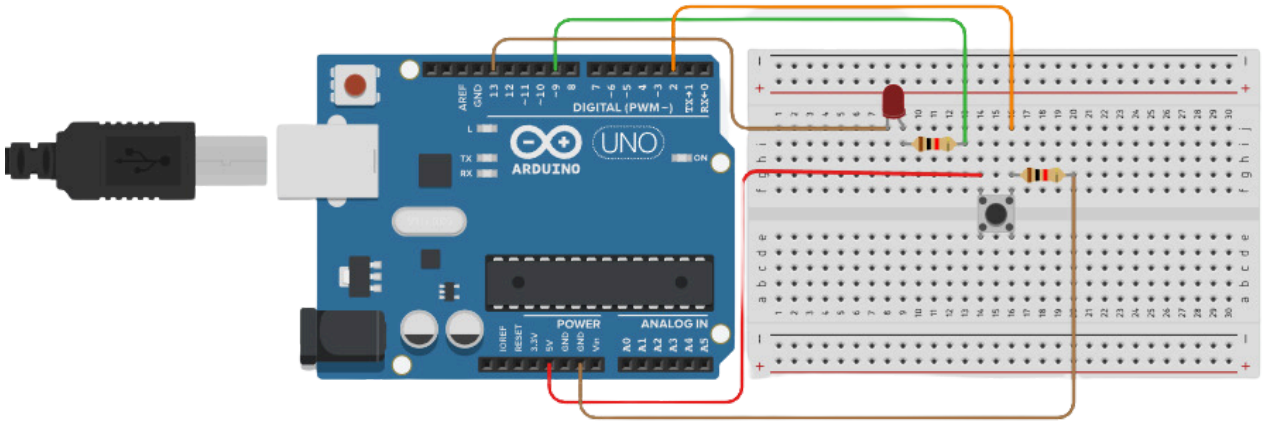
ETKİNLİK 2

Buton ile led yakıp söndürelim !

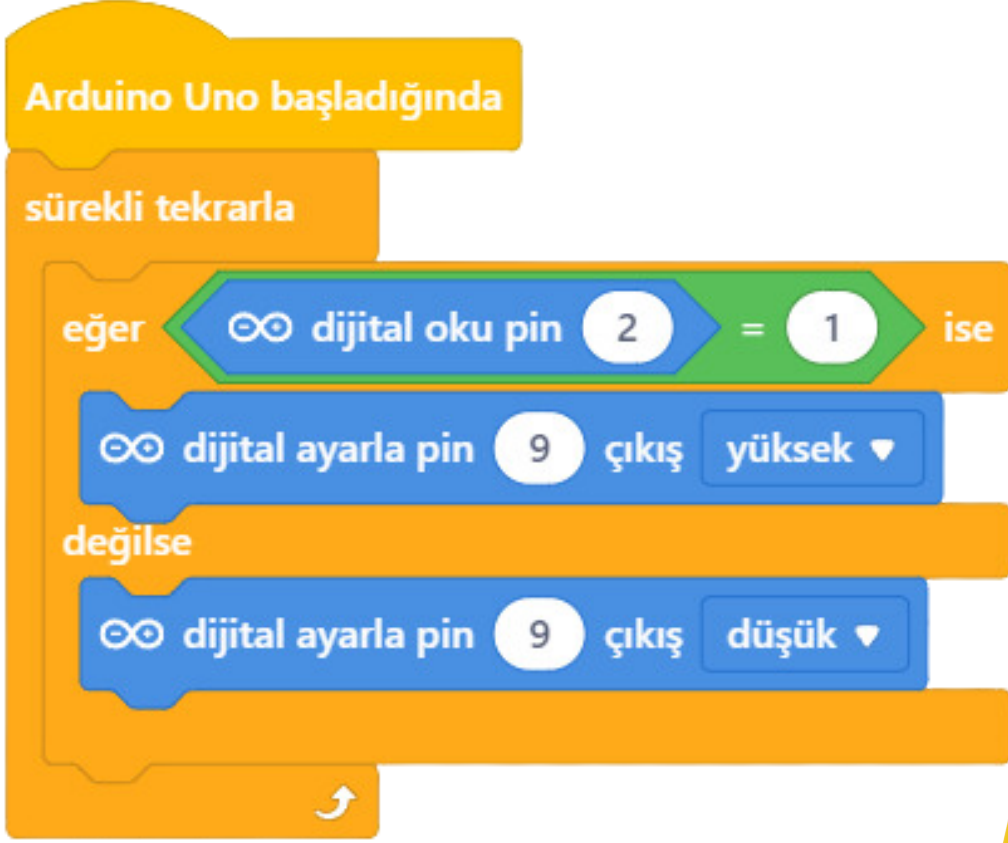
Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Led
- Direnç
- Buton

Devre Şeması



Mblock Kodları



Blok tabanlı kodlama ile metin tabanlı kodlamanın arasındaki en önemli fark **anlaşılabilirlik ve zamandır**. Metin tabanlı kodlama ile hazırlanan bir kodu daha kısa sürede ve kodlama mantığını daha kolay kavrayarak blok tabanlı kodlama ile yapılabilir. Blok tabanlı programlamanın dezavantajı da ne kadar projeler geliştirilse de sınırlı kalacaktır.

Metin tabanlı programlamada geliştirebileceğimiz proje sayısı sınırsızdır.



Metin Tabanlı Kodları

```
int ledPin = 10;
int buttonPin = 2;

int ledDurumu = LOW;
int buttonDurumu;

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

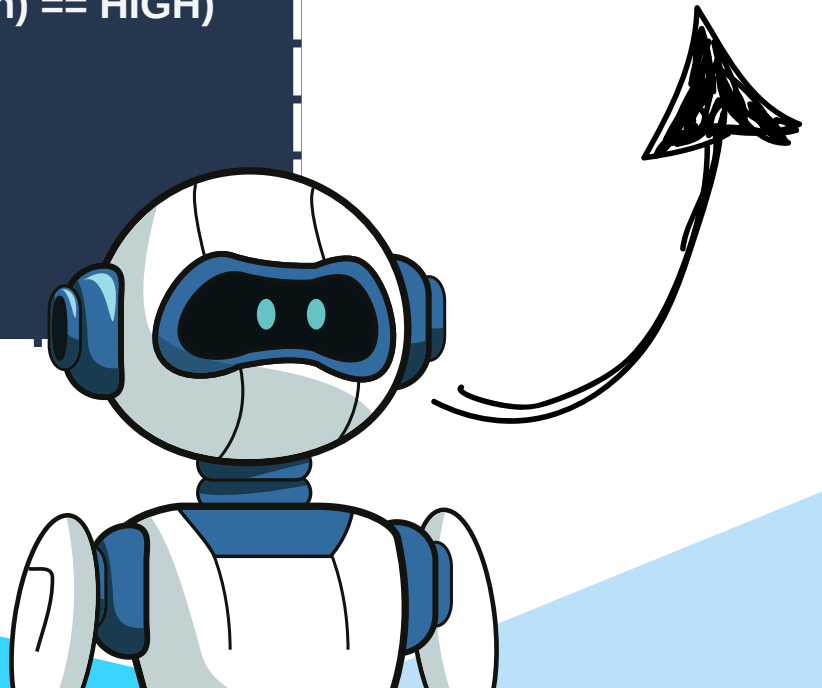
void loop() {
  buttonDurumu = digitalRead(buttonPin);

  if (buttonState == HIGH) {

    ledDurumu = !ledDurumu;
    digitalWrite(ledPin, ledDurumu);

    while (digitalRead(buttonPin) == HIGH)
    {
      delay(10);
    }
  }
}
```

Dirençler, elektrikli devrelerde akımı sınırlayarak **belli bir değerde tutmaya yararlar**. Bunun haricinde hassas devre elemanlarının üzerlerinden yüksek akım geçmesini önlerler.



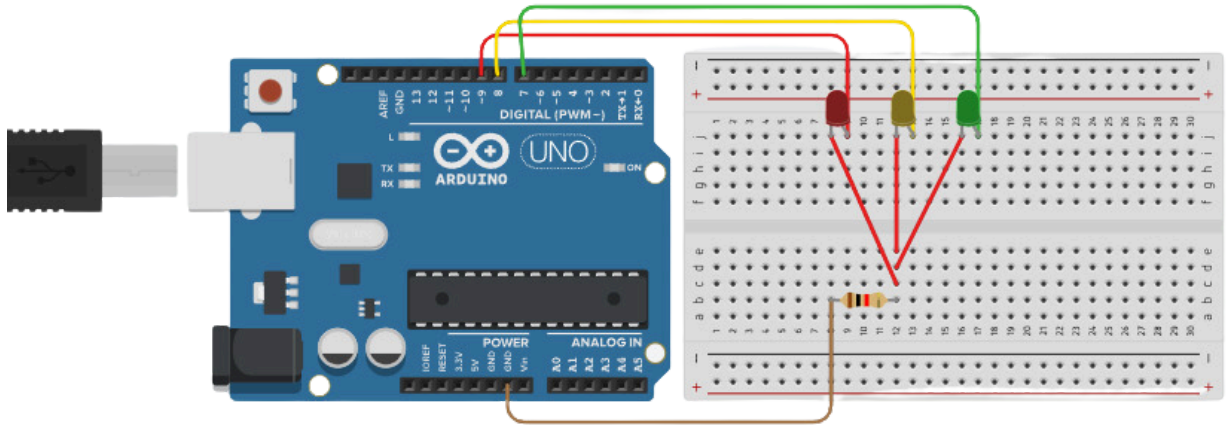
ETKİNLİK 3

Hadi! Trafik ışığı yapalım !

Kullanılacak Malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Led
- Direnç

Devre Şeması



Mblock Kodları

```
Arduino Uno başladığında  
sürekli tekrarla  
dijital ayarla pin 9 çıkış yüksek ▼  
3 saniye bekle  
dijital ayarla pin 9 çıkış düşük ▼  
dijital ayarla pin 8 çıkış yüksek ▼  
1 saniye bekle  
dijital ayarla pin 8 çıkış düşük ▼  
dijital ayarla pin 7 çıkış yüksek ▼  
2 saniye bekle  
dijital ayarla pin 7 çıkış düşük ▼
```

LED, “Light Emitting Diode” yani “Işık Yayan Diyot” anlamına gelir.LED’ler trafik ışıklarından televizyon ekranlarına, dekoratif aydınlatmalardan tıbbi cihazlara kadar geniş bir yelpazede kullanılır.



Metin Tabanlı Kodları

```
int led1Pin = 8;
int led2Pin = 9;
int led3Pin = 10;

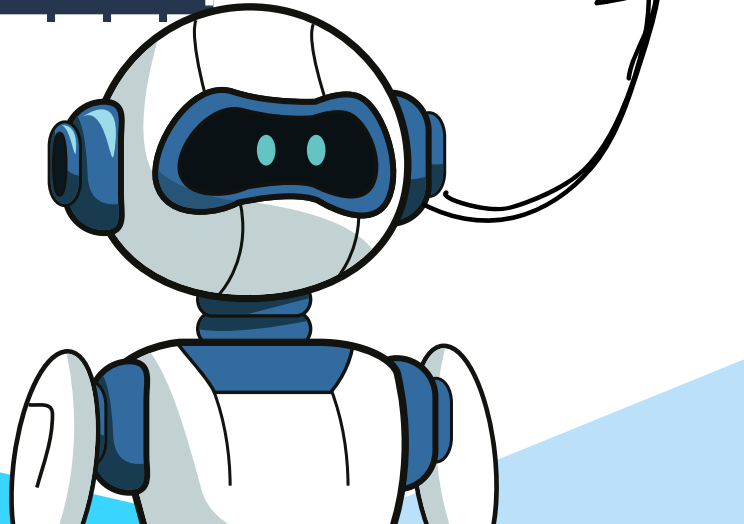
void setup() {
  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
}

void loop() {
  digitalWrite(led1Pin, HIGH);
  delay(1000);
  digitalWrite(led1Pin, LOW);

  digitalWrite(led2Pin, HIGH);
  delay(1000);
  digitalWrite(led2Pin, LOW);

  digitalWrite(led3Pin, HIGH);
  delay(1000);
  digitalWrite(led3Pin, LOW);
}
```

Makineler, genellikle göründükleri kadar zeki değildir. İnsan zekası ve sezgisi gibi karmaşık düşünme yetenekleri yoktur. Bu nedenle, algoritmalar oluşturulurken her adımın **anlaşılır, sıralı ve net** bir şekilde tanımlanması önemlidir.



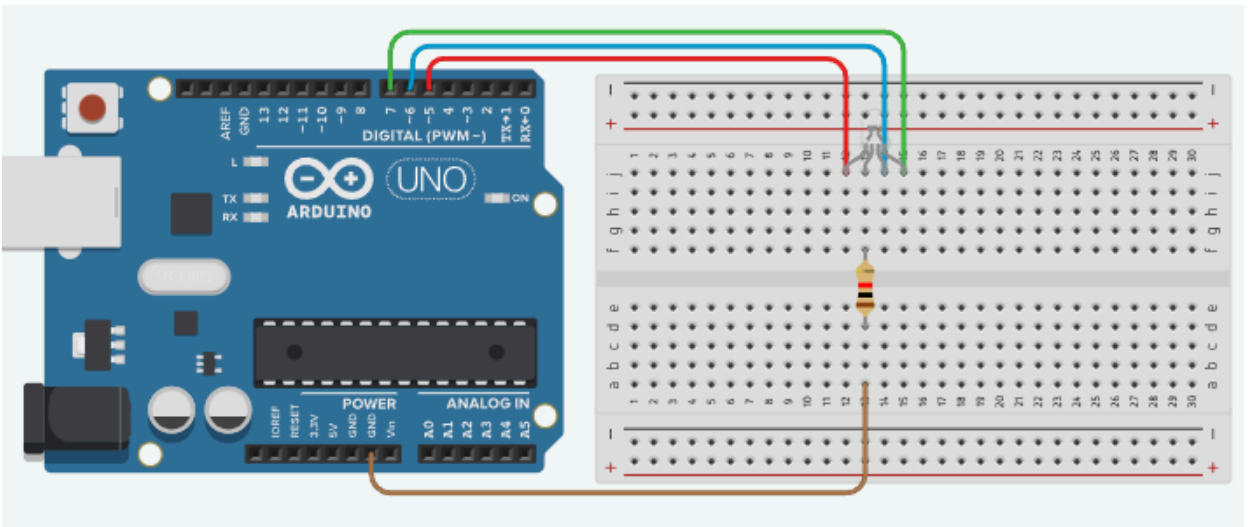
ETKİNLİK 4

RGB led çalıştıralım!

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Direnç
- RGB Led

Devre Şeması



Mblock Kodları

```
Arduino Uno başladığında  
sürekli tekrarla  
∞ dijital ayarla pin 5 çıkış yüksek ▼  
1 saniye bekle  
∞ dijital ayarla pin 5 çıkış düşük ▼  
∞ dijital ayarla pin 6 çıkış yüksek ▼  
1 saniye bekle  
∞ dijital ayarla pin 6 çıkış düşük ▼  
∞ dijital ayarla pin 7 çıkış yüksek ▼  
1 saniye bekle  
∞ dijital ayarla pin 7 çıkış düşük ▼
```

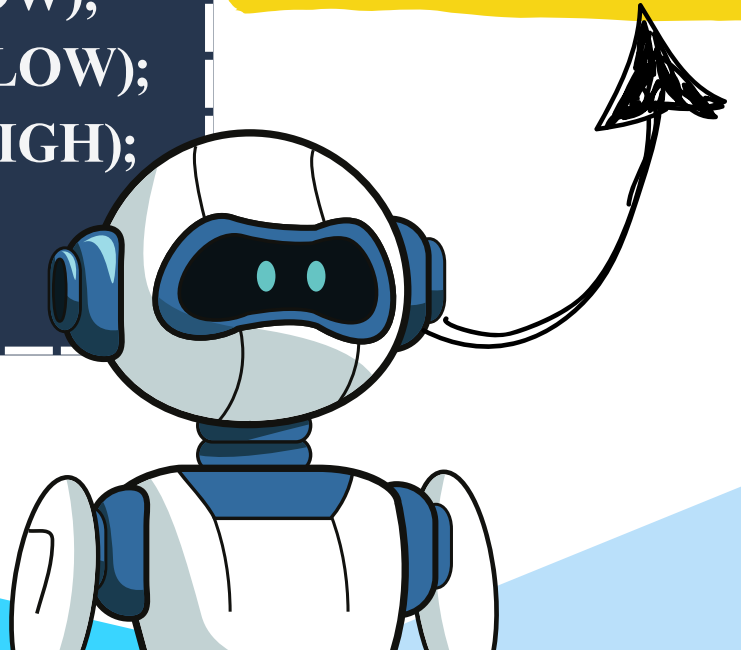
RGB LED'ler, sadece üç ana rengi kullanarak beyaz dahil birçok renk oluşturabilir. Örneğin, kırmızı ve yeşil LED'ler birlikte sarı rengi oluşturur.



Metin Tabanlı Kodları

```
int redPin = 7;
int greenPin = 8;
int bluePin = 9;
void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}
void loop() {
  digitalWrite(redPin, HIGH);
  digitalWrite(greenPin, LOW);
  digitalWrite(bluePin, LOW);
  delay(1000);
  digitalWrite(redPin, LOW);
  digitalWrite(greenPin, HIGH);
  digitalWrite(bluePin, LOW);
  delay(1000);
  digitalWrite(redPin, LOW);
  digitalWrite(greenPin, LOW);
  digitalWrite(bluePin, HIGH);
  delay(1000);
}
```

C++ programlama dili kullanarak Arduino IDE programında Arduino kartlarımızı programlayabiliriz. Ayrıca C++, oyun geliştirme, grafik uygulamaları, işletim sistemleri, sürücüler, gömülü sistemler ve daha birçok alanda kullanılır.



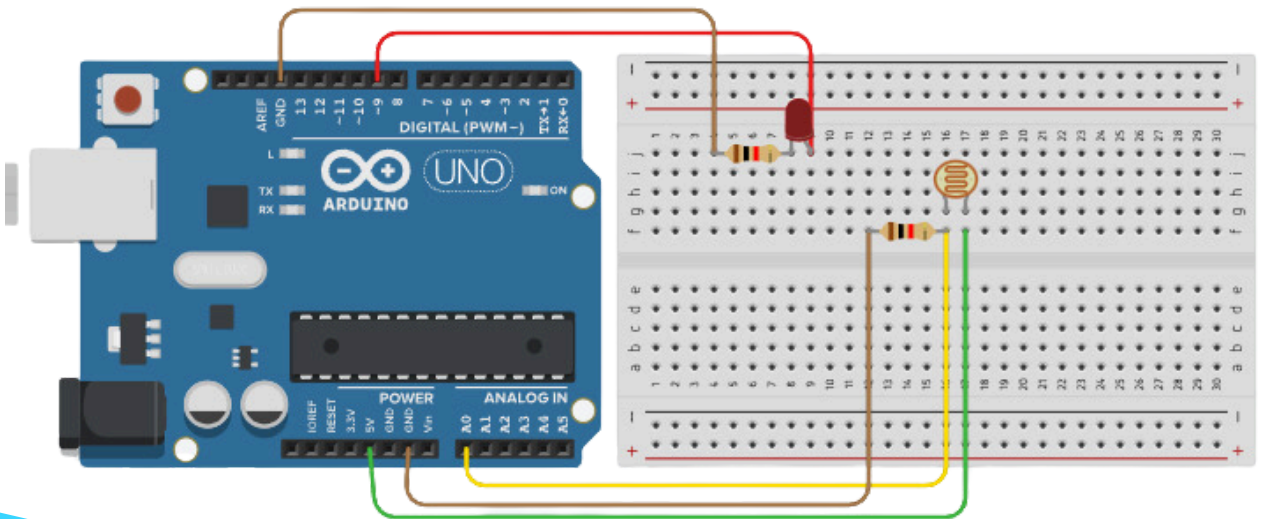
ETKİNLİK 5

Işığa duyarlı sokak lambası yapalım!

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Led
- Direnç
- LDR Led

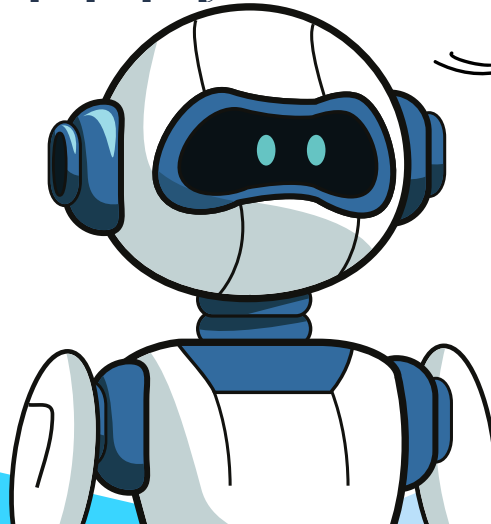
Devre Şeması



Metin Tabanlı Kodları

```
int ldrPin = A0;
int ledPin = 7;
int ldrValue = 0;
int LdrDegeri= 500;
void setup() {
pinMode(ledPin, OUTPUT);
Serial.begin(9600);
}
void loop() {
ldrValue = analogRead(ldrPin);
Serial.println(ldrValue);
if (ldrValue < LdrDegeri) {
digitalWrite(ledPin, HIGH);
}
else {
digitalWrite(ledPin, LOW);
}
delay(100);
}
```

LDR'ler, otomatik aydınlatma sistemleri, ortam ışığı sensörleri, fotoğraf ekipmanlarında pozlama ayarları ve hatta güvenlik sistemlerinde yaygın olarak kullanılır. Örneğin, sokak lambalarının otomatik olarak açılıp kapanmasını sağlayan sistemlerde LDR'ler kullanılır. Sıcak havalarda dikkatli olmakta fayda var 60°C üzeri sıcaklıklarda arıza verebilir.



ETKİNLİK 6

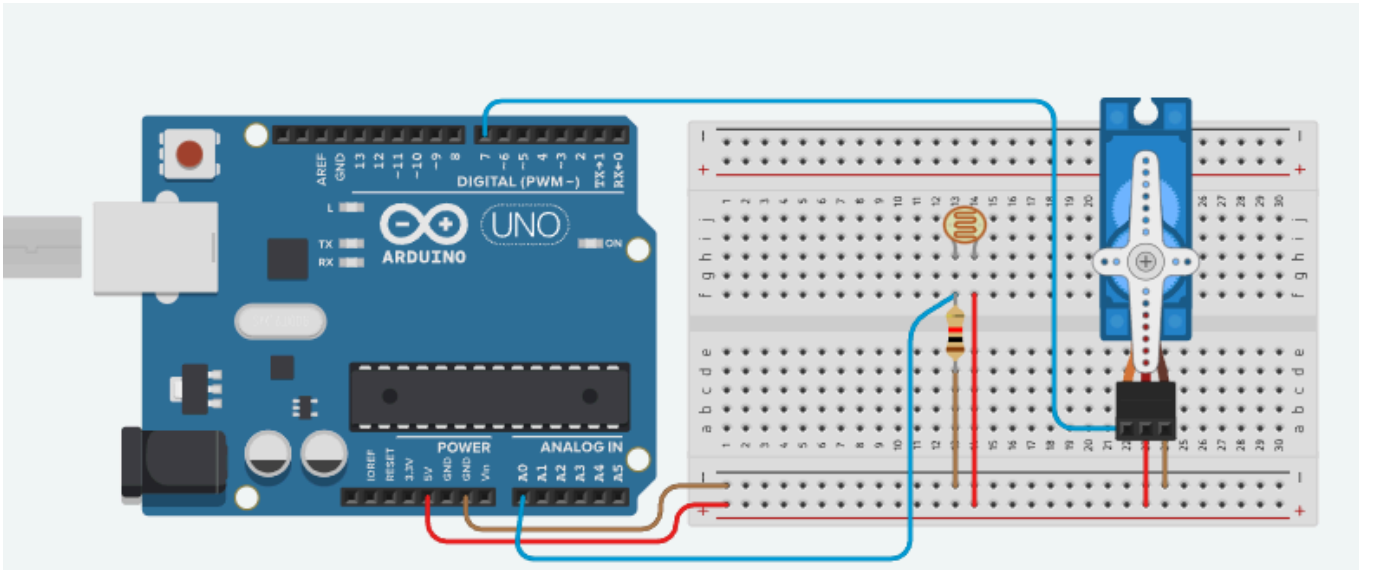
Ldr ile servo kontrol edelim !

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Servo motor
- Direnç
- LDR Led

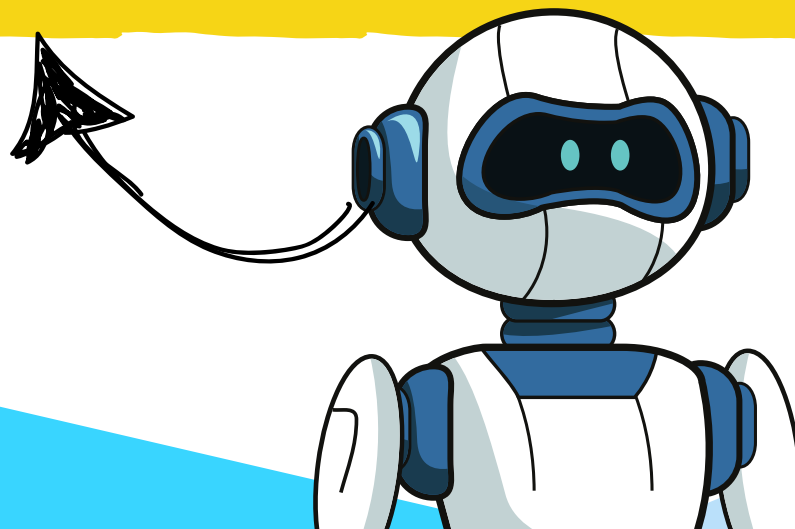


Devre Şeması



Metin Tabanlı Kodları

```
#include <Servo.h>
int ldrPin = A0;
int servoPin = 9;
Servo myServo;
void setup() {
  myServo.attach(servoPin);
}
void loop() {
  int ldrValue = analogRead(ldrPin);
  int servoPosition =map(ldrValue, 0, 1023, 0, 180);
  myServo.write(servoPosition);
  delay(100);
}
```

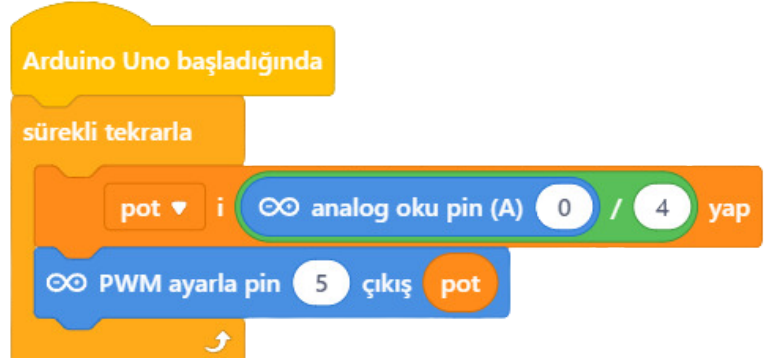


ETKİNLİK 7

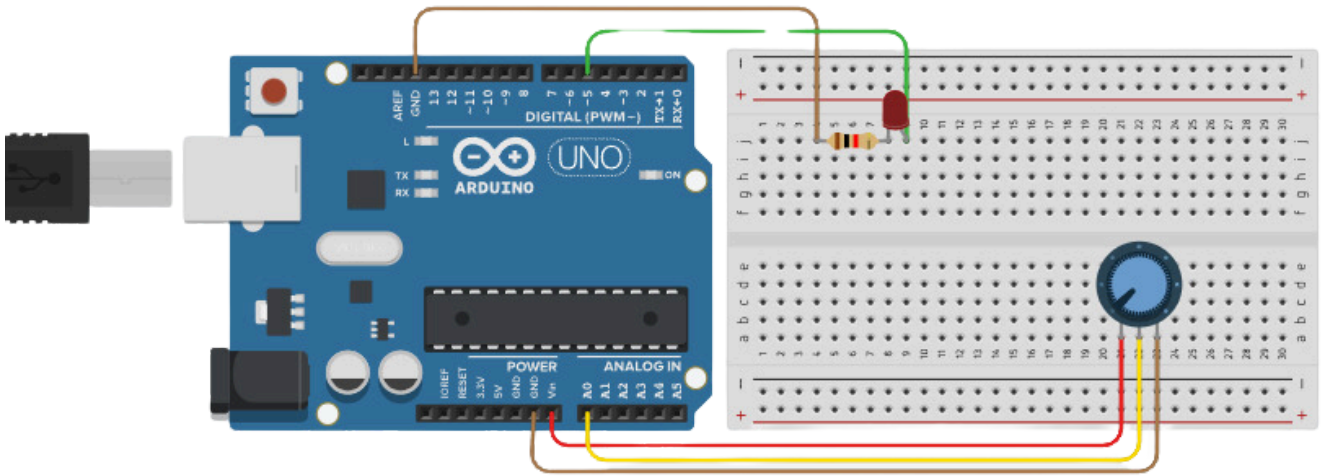
Potansiyometre ile led ışık kontrol edelim!

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Led
- Direnç
- Potansiyometre



Devre Şeması

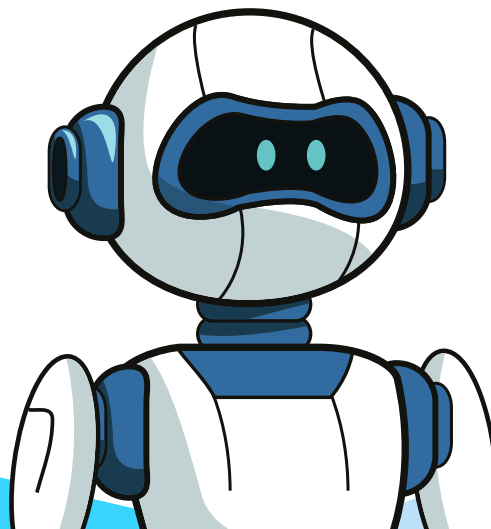


Metin Tabanlı Kodları

```
int potPin = A0;
int ledPin = 9;
int potValue = 0;
int ledBrightness = 0;

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

void loop() {
  potValue = analogRead(potPin);
  ledBrightness = map(potValue, 0, 1023, 0, 255);
  analogWrite(ledPin, ledBrightness);
  delay(10);
}
```



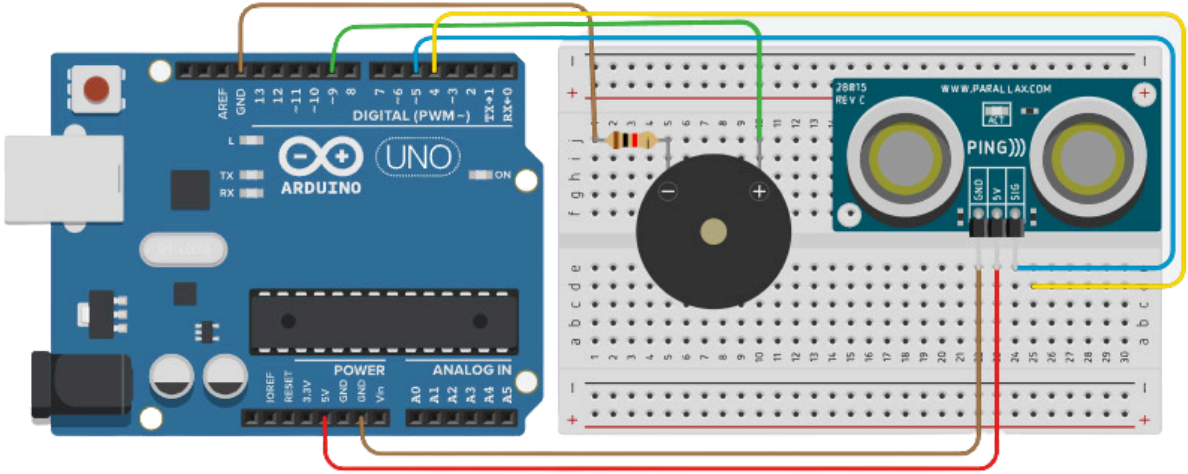
ETKİNLİK 8

Ultrasonic mesafe sensörü ile buzzer çalıştıralım.

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Buzzer
- Direnç
- Ultrasonic mesafe

Devre Şeması



Mblock Kodları

```
Arduino Uno başladığında  
sürekli tekrarla  
mesafe i mesafe algılayıcı 4 trig pin 5 echo pin yap  
eğer mesafe < 30 ise  
dijital ayarla pin 9 çıkış yüksek  
değilse  
dijital ayarla pin 9 çıkış düşük
```



Metin Tabanlı Kodları

```
#define trigPin 9
#define echoPin 10
#define buzzerPin 11
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  long sure, mesafe;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

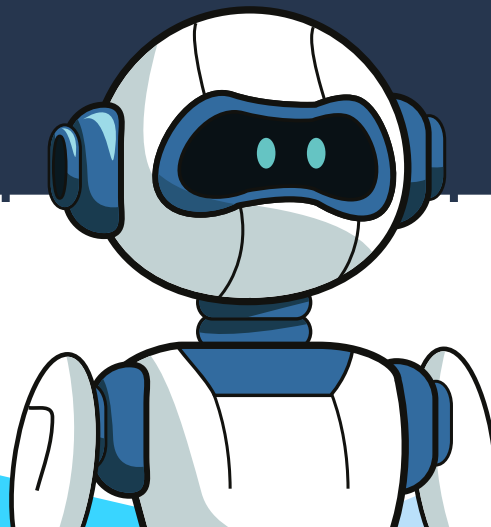
  sure = pulseIn(echoPin, HIGH);

  mesafe = (sure / 2) / 29.1;

  Serial.print("Mesafe: ");
  Serial.print(mesafe);
  Serial.println(" cm");

  if (mesafe < 20 && mesafe > 0) {
    digitalWrite(buzzerPin, HIGH);
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(100);
}
```



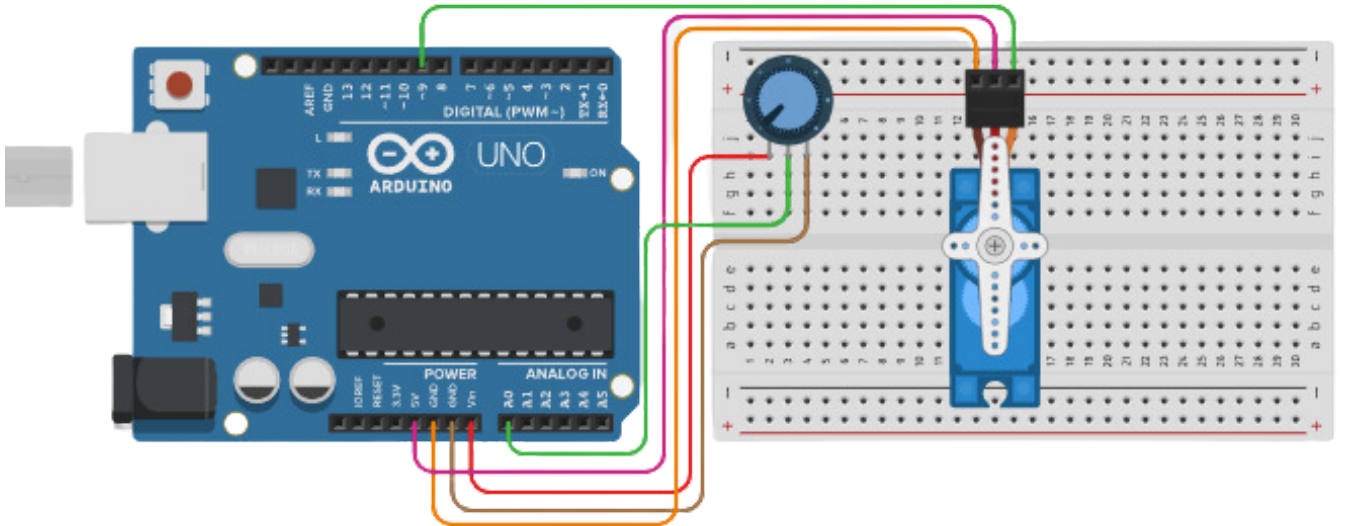
ETKİNLİK 9

Potansiyometre ile Servo motoru kontrol edelim !

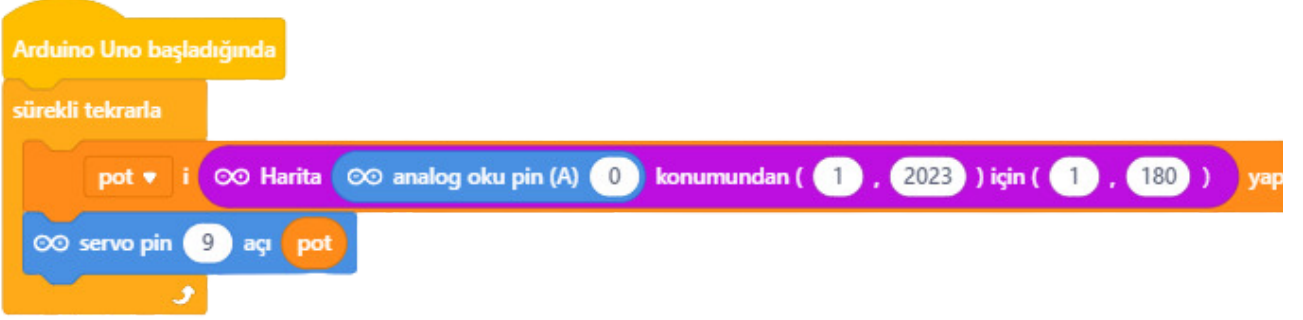
Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Direnç
- Potansiyometre
- Servo motor

Devre Şeması



Mblock Kodları



Metin Tabanlı Kodları

```
#include <Servo.h>
```

```
int potPin = A0;
```

```
int servoPin = 9;
```

```
Servo servoMotor;
```

```
void setup() {
```

```
  servoMotor.attach(servoPin);
```

```
}
```

```
void loop() {
```

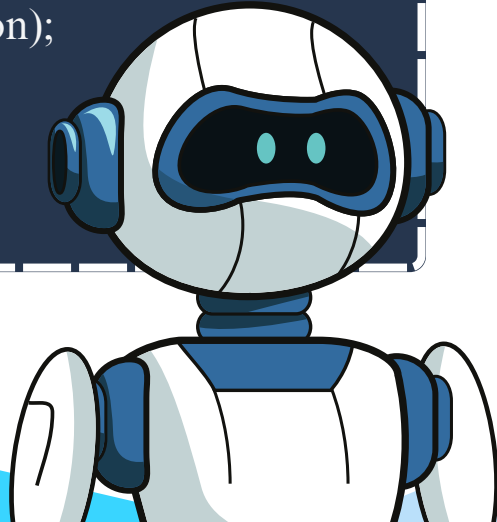
```
  int potValue = analogRead(potPin);
```

```
  int servoPosition = map(potValue, 0, 1023, 0, 180);
```

```
  servoMotor.write(servoPosition);
```

```
  delay(15);
```

```
}
```



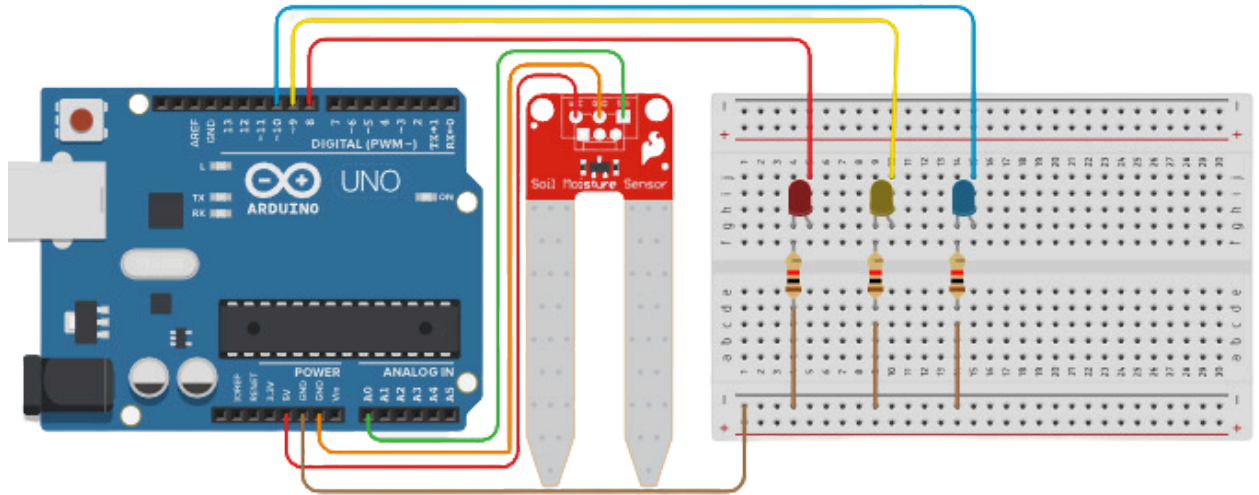
ETKİNLİK 10

Nem sensörü ile Ledleri Yakalım !

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Direnç
- Sıvı sensörü
- Led

Devre Şeması



Mblock Kodları

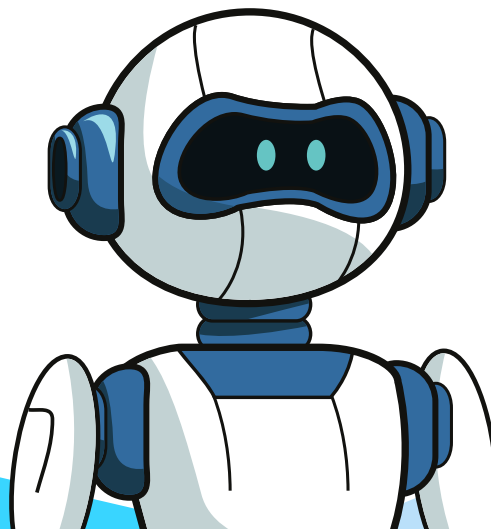
```
Arduino Uno başladığında
sürekli tekrarla
değer i analog oku pin (A) 0 yap
eğer değer < 100 ise
  dijital ayarla pin 8 çıkış yüksek
  dijital ayarla pin 9 çıkış düşük
  dijital ayarla pin 10 çıkış düşük
eğer değer < 200 ve değer > 100 ise
  dijital ayarla pin 8 çıkış düşük
  dijital ayarla pin 9 çıkış yüksek
  dijital ayarla pin 10 çıkış düşük
eğer değer > 200 ise
  dijital ayarla pin 8 çıkış düşük
  dijital ayarla pin 9 çıkış düşük
  dijital ayarla pin 10 çıkış yüksek
```



Metin Tabanlı Kodları

```
int toprakNemPin = A0;
int ledPin = 9;
int nemDegeri = 0;
int esikDegeri = 500;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  nemDegeri = analogRead(toprakNemPin);
  Serial.print("Nem Degeri: ");
  Serial.println(nemDegeri);
  if (nemDegeri < esikDegeri) {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
  delay(1000);
}
```



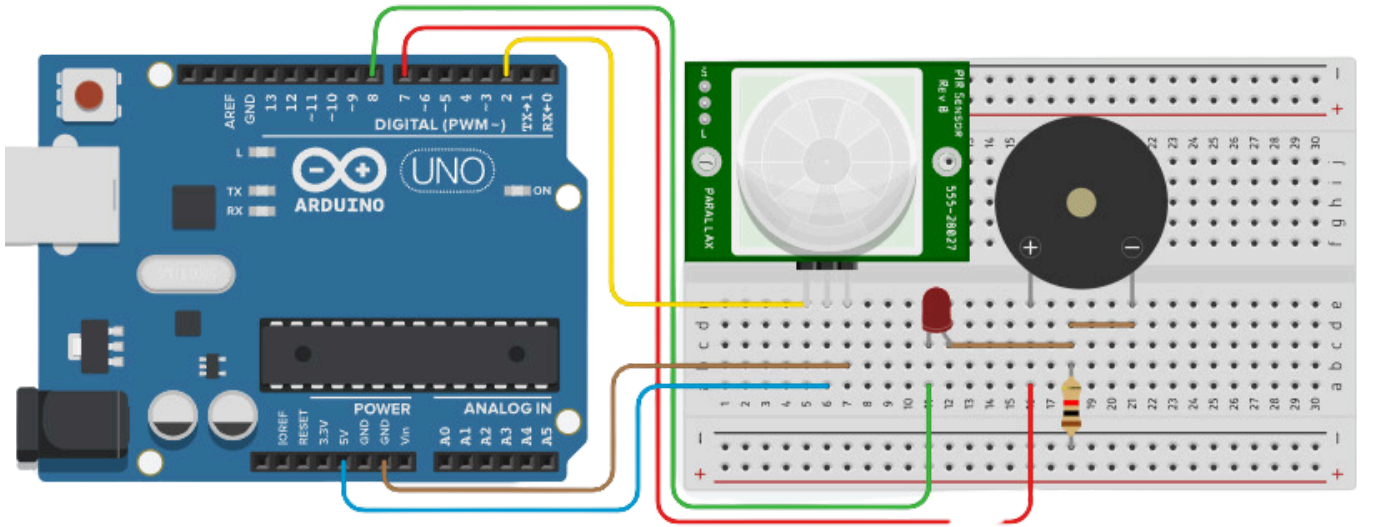
ETKİNLİK 11

Pır Hareket sensörü ile Buzzer ve Led çalıştıralım

Kullanılacak malzeme

- Arduino uno
- BreadBord
- Jumper kablo
- Direnç
- Buzzer
- Pır hareket sensörü
- Led

Devre Şeması



Mblock Kodları

```
Arduino Uno başladığında  
sürekli tekrarla  
eğer dijital oku pin 2 = 1 ise  
dijital ayarla pin 9 çıkış yüksek ▼  
dijital ayarla pin 8 çıkış yüksek ▼  
1 saniye bekle  
dijital ayarla pin 9 çıkış düşük ▼  
dijital ayarla pin 8 çıkış düşük ▼  
1 saniye bekle  
değilse  
dijital ayarla pin 9 çıkış düşük ▼  
dijital ayarla pin 8 çıkış düşük ▼
```



Metin Tabanlı Kodları

```
#define pirPin 2
#define buzzerPin 9
#define ledPin 10

void setup() {
  pinMode(pirPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

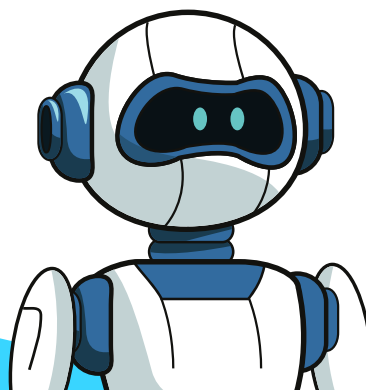
void loop() {
  int hareketDurumu = digitalRead(pirPin);

  if (hareketDurumu == HIGH) {

    digitalWrite(buzzerPin, HIGH);
    digitalWrite(ledPin, HIGH);
    Serial.println("Hareket Algılandı!");
  } else {

    digitalWrite(buzzerPin, LOW);
    digitalWrite(ledPin, LOW);
    Serial.println("Hareket Yok");
  }

  delay(100);
}
```

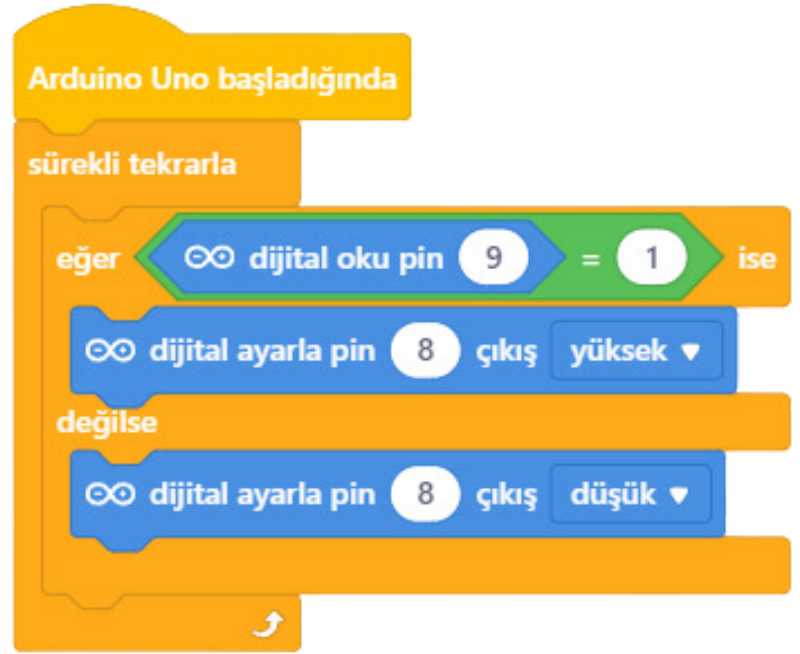


ETKİNLİK 12

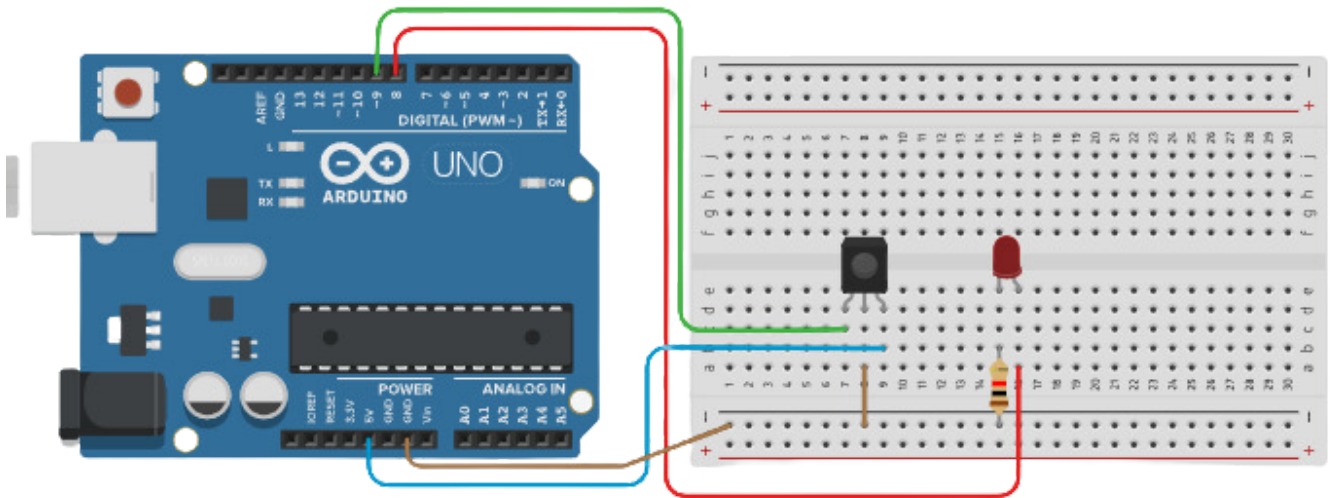
Mikrofon ile led kontrolü yapalım

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Direnç
- Led
- Ses Mikrafon sensörü



Devre Şeması



Metin Tabanlı Kodları

```
int ledPin = 9;
int mikrofonPin = A0;
int mikrofonDegeri = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  mikrofonDegeri = analogRead(mikrofonPin);
  Serial.println(mikrofonDegeri);
  if (mikrofonDegeri > 500) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }

  delay(100);
}
```

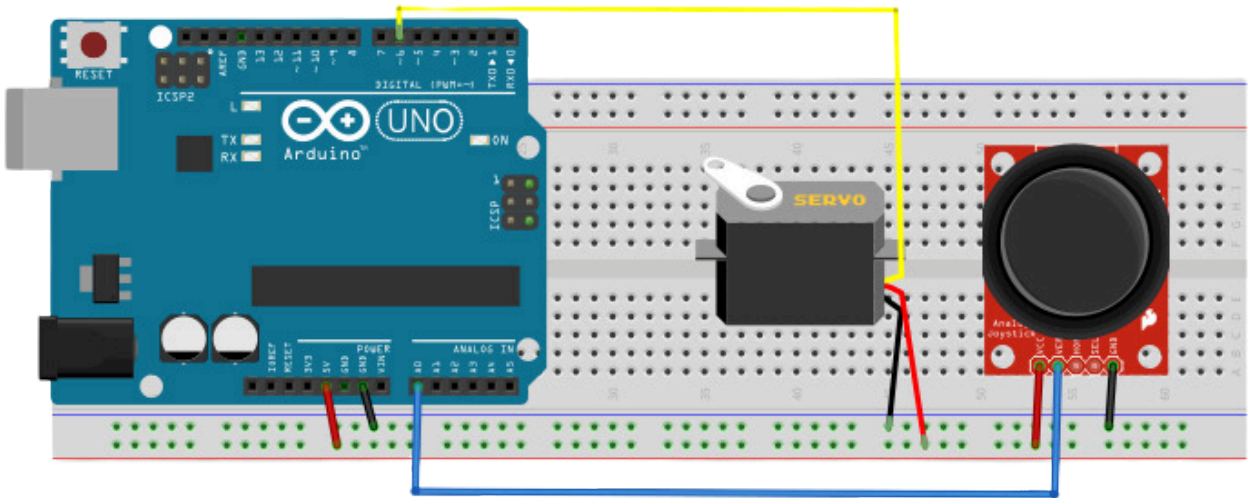
ETKİNLİK 13

Joystick ile servo kontrol edelim!

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Joystick
- Servo motor

Devre Şeması



Mblock Kodları

```
Arduino Uno başladığında  
sürekli tekrarla  
x eksenini i analog oku pin (A) 1 / 6 yap  
servo pin 2 açısı x eksenini
```



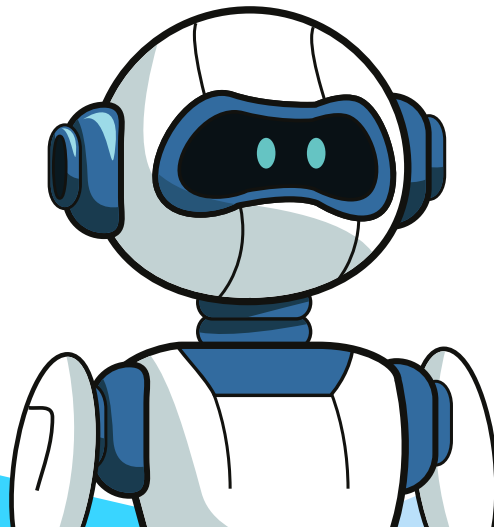
Metin Tabanlı Kodları

```
#include <Servo.h>

Servo servoMotor;
int joystickXPin = A0;
int joystickDegeri;

void setup() {
  servoMotor.attach(6);
  Serial.begin(9600);
}

void loop() {
  joystickDegeri = analogRead(joystickXPin);
  int servoAci = map(joystickDegeri, 0, 1023, 0, 180);
  servoMotor.write(servoAci);
  Serial.println(servoAci);
  delay(15);
}
```



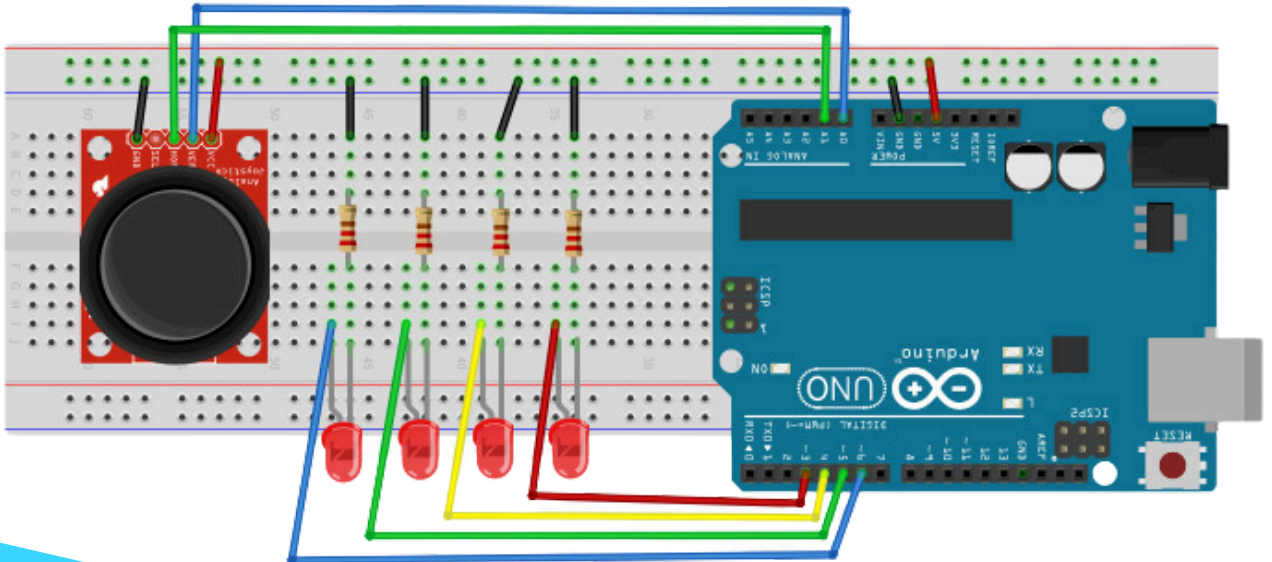
ETKİNLİK 14

Joystick ile led kontrol edelim!

Kullanılacak malzeme

- Arduino uno
- Bread Bord
- Jumper kablo
- Direnç
- Joystick
- Led

Devre Şeması



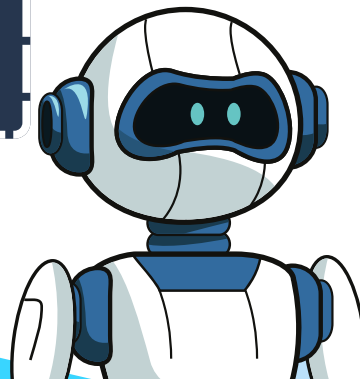
Mblock Kodları

```
Arduino Uno başladığında
sürekli tekrarla
  x eksenini i analog oku pin (A) 0 yap
  y eksenini i analog oku pin (A) 1 yap
  eğer x eksenini > 510 ise
    dijital ayarla pin 3 çıkış yüksek
  değilse
    dijital ayarla pin 3 çıkış düşük
  eğer x eksenini < 510 ise
    dijital ayarla pin 4 çıkış yüksek
  değilse
    dijital ayarla pin 4 çıkış düşük
  eğer y eksenini > 510 ise
    dijital ayarla pin 5 çıkış yüksek
  değilse
    dijital ayarla pin 5 çıkış düşük
  eğer y eksenini < 510 ise
    dijital ayarla pin 6 çıkış yüksek
  değilse
    dijital ayarla pin 6 çıkış düşük
```

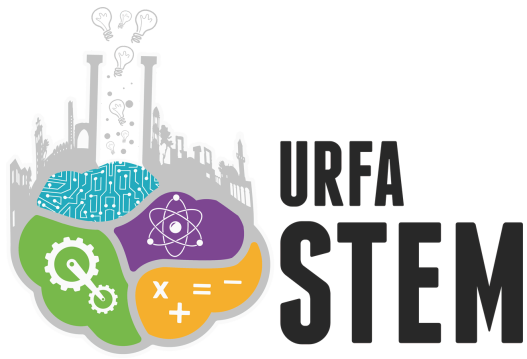


Metin Tabanlı Kodları

```
int xPin = A0; int yPin = A1; int ledK = 2;
int ledM = 3; int ledY = 4; int ledS = 5;
void setup() {
  pinMode(xPin, INPUT);
  pinMode(yPin, INPUT);
  pinMode(ledK, OUTPUT);
  pinMode(ledM, OUTPUT);
  pinMode(ledY, OUTPUT);
  pinMode(ledS, OUTPUT);
}
void loop() {
  int xPozisyon = analogRead(xPin);
  int yPozisyon = analogRead(yPin);
  if (xPozisyon < 10) {
    digitalWrite(ledK, HIGH);
  } else {
    digitalWrite(ledK, LOW);
  }
  if (xPozisyon > 1000) {
    digitalWrite(ledM, HIGH);
  } else {
    digitalWrite(ledM, LOW);
  }
  if (yPozisyon < 10) {
    digitalWrite(ledY, HIGH);
  } else {
    digitalWrite(ledY, LOW);
  }
  if (yPozisyon > 1000) {
    digitalWrite(ledS, HIGH);
  } else {
    digitalWrite(ledS, LOW);
  }
}
```







ŞANLIURFA İL MİLLİ EĞİTİM MÜDÜRLÜĞÜ
STEM VE BİLİM MERKEZİ