

# Gerçek Zamanlı Deprem Algılama ve Erken Uyarı Sistemi

## Deneyin Amacı:

Bu deneyin amacı, piezoelektrik sensör kullanarak titreşim veya basınç algılamaktır. Bu algılama sonucunda bir hoparlör (buzzer) ile uyarı verilir ve belirli durumlarda LED'ler aracılığıyla risk seviyeleri görsel olarak gösterilir. Ayrıca bir sıfırlama butonu ile sistemin durumu yeniden başlatılabilir.

## Kullanılan Malzemeler ve Görevleri:

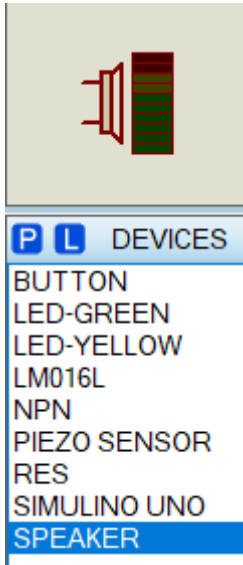
- Piezoelektrik Sensör (PZ1):**
  - Titreşim veya basıncı algılar ve bunu elektrik sinyaline dönüştürür.
  - Görev:** Algıladığı titreşim seviyesine göre uyarı sistemini çalıştırır.
- Hoparlör (Speaker):**
  - Titreşim algılandığında sesli uyarı verir.
  - Görev:** Durumu sesli olarak bildirmek.
- LED'ler:**
  - Yeşil LED (D1):** Normal durum göstergesi.
  - Sarı LED (D2):** Orta risk uyarısı.
  - Görevleri:** Durumun görsel bildirimini sağlar.
- Dirençler (R1, R2, R5, R6):**
  - Akımı sınırlamak ve devre elemanlarını korumak için kullanılır.
  - Görev:** LED'leri ve diğer elemanları uygun voltajda çalıştırır.
- Sıfırlama Butonu (SW1):**
  - Sistemi sıfırlamak için kullanılır.
  - Görev:** Devredeki tüm uyarıları kapatır.
- Arduino UNO:**
  - Sensör ve diğer devre elemanlarının çalışmasını kontrol eden mikrodenetleyici.
  - Görev:** Titreşim seviyesini ölçmek, uyarı koşullarını değerlendirmek ve çıkışlara sinyal göndermek.
- Transistör (Q2 - NPN):**
  - Hoparlörü çalıştırmak için gereken akımı kontrol eder.
  - Görev:** Arduino'nun düşük akımlı dijital sinyalini yükselterek hoparlörün çalışmasını sağlar. Transistör, anahtar görevi görür.

- c. **Not:** Arduino'nun çıkış pinleri, hoparlör gibi yükleri doğrudan sürebilecek kadar güçlü değildir. Bu nedenle, transistör bir sürücü eleman olarak kullanılır.

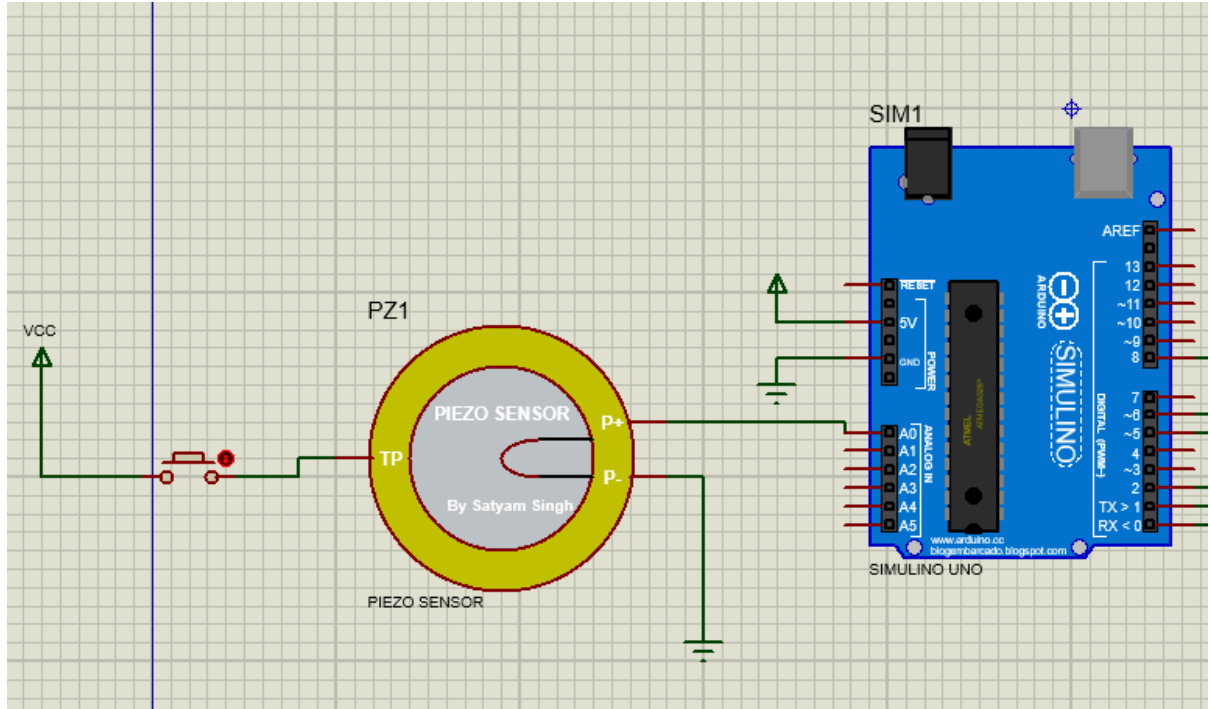
7. **Seri Monitör (PC Ekranı):**

- a. Arduino'dan gelen verileri görüntüler.  
b. **Görev:** Piezo sensörün okuduğu titreşim değerlerini ve uyarı durumlarını kullanıcıya gösterir.

## Devre Elemanlarının Bağlantıları Ve Deneyin Yapılışı

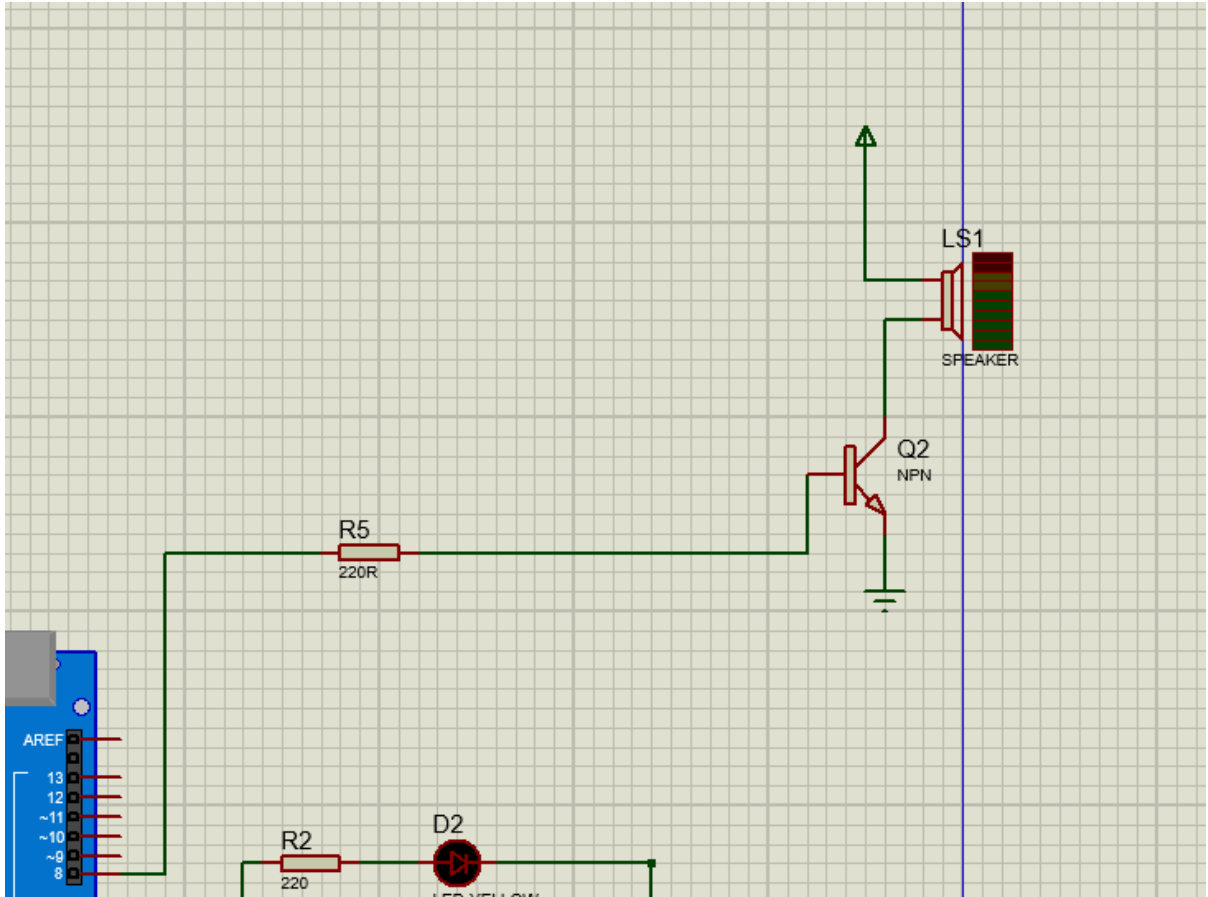


Devre elemanları seçilir.



### Piezoelektrik Sensör:

- **+ Ucu:** A0 analog pinine bağlanır.
- **- Ucu:** GND'ye bağlanır.
- **TP Ucu :** Bir buton ile güç kaynağına bağlanır.

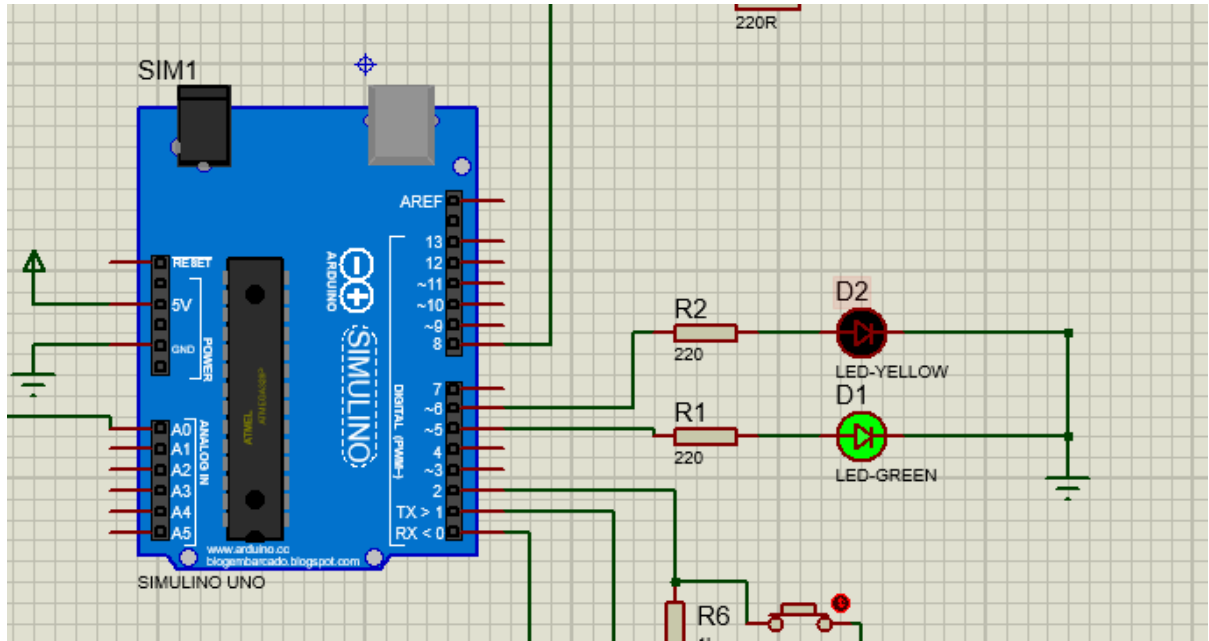


### Hoparlör (Speaker):

- **+ Ucu:** +5V kaynağına bağlanır.
- **- Ucu:** Transistörün kollektör (C) ucuna bağlanır.

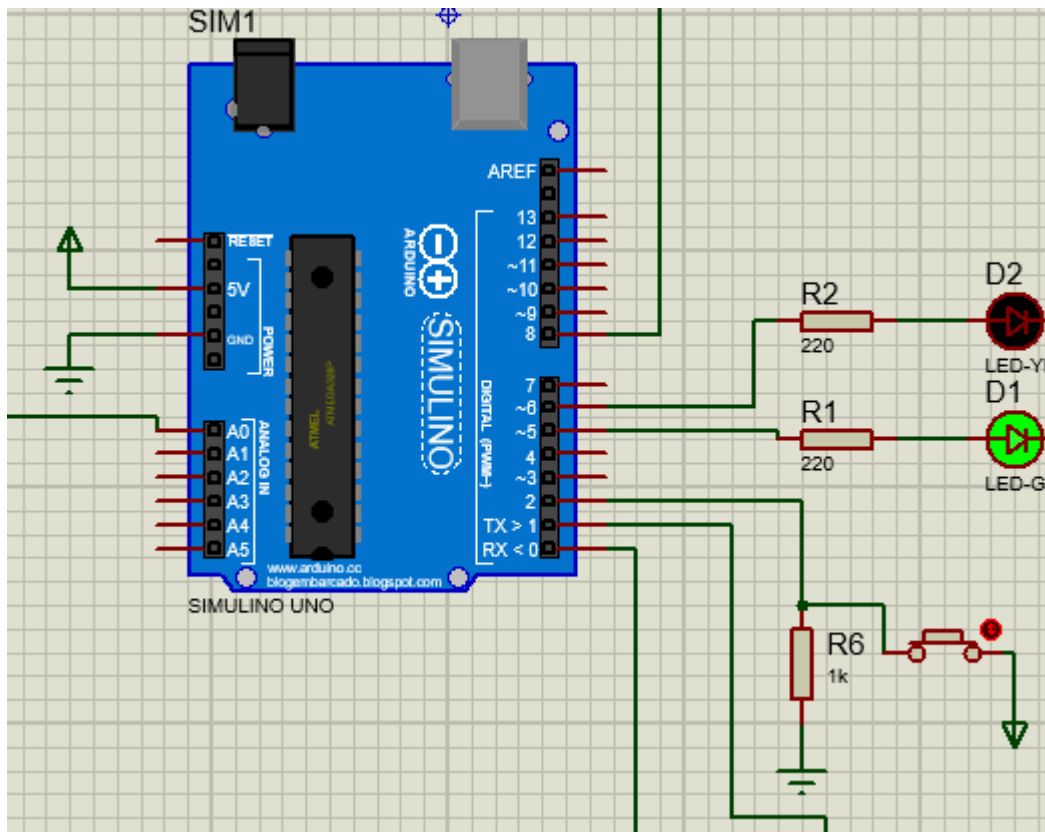
### Transistör (NPN):

- **Baz (B):** Arduino'nun **Pin 8** çıkışına, bir direnç (R5) aracılığıyla bağlanır.
- **Kollektör (C):** Hoparlörün negatif ucuna bağlanır.
- **Emiter (E):** GND'ye bağlanır.

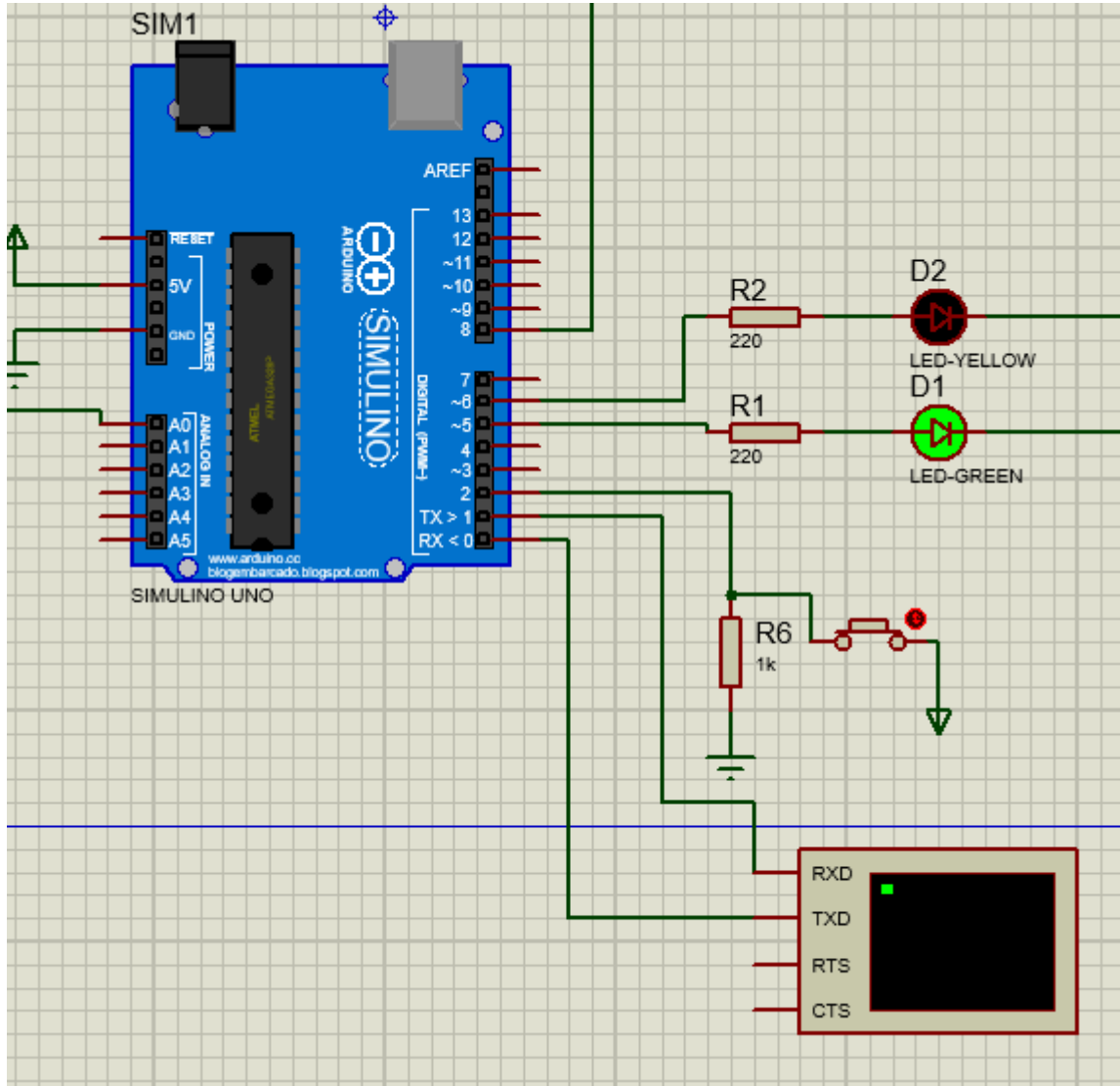


### LED'ler:

- **Yeşil LED (D1):** Pozitif ucu **Pin 5**, negatif ucu direnç üzerinden GND.
- **Sarı LED (D2):** Pozitif ucu **Pin 6**, negatif ucu direnç üzerinden GND.

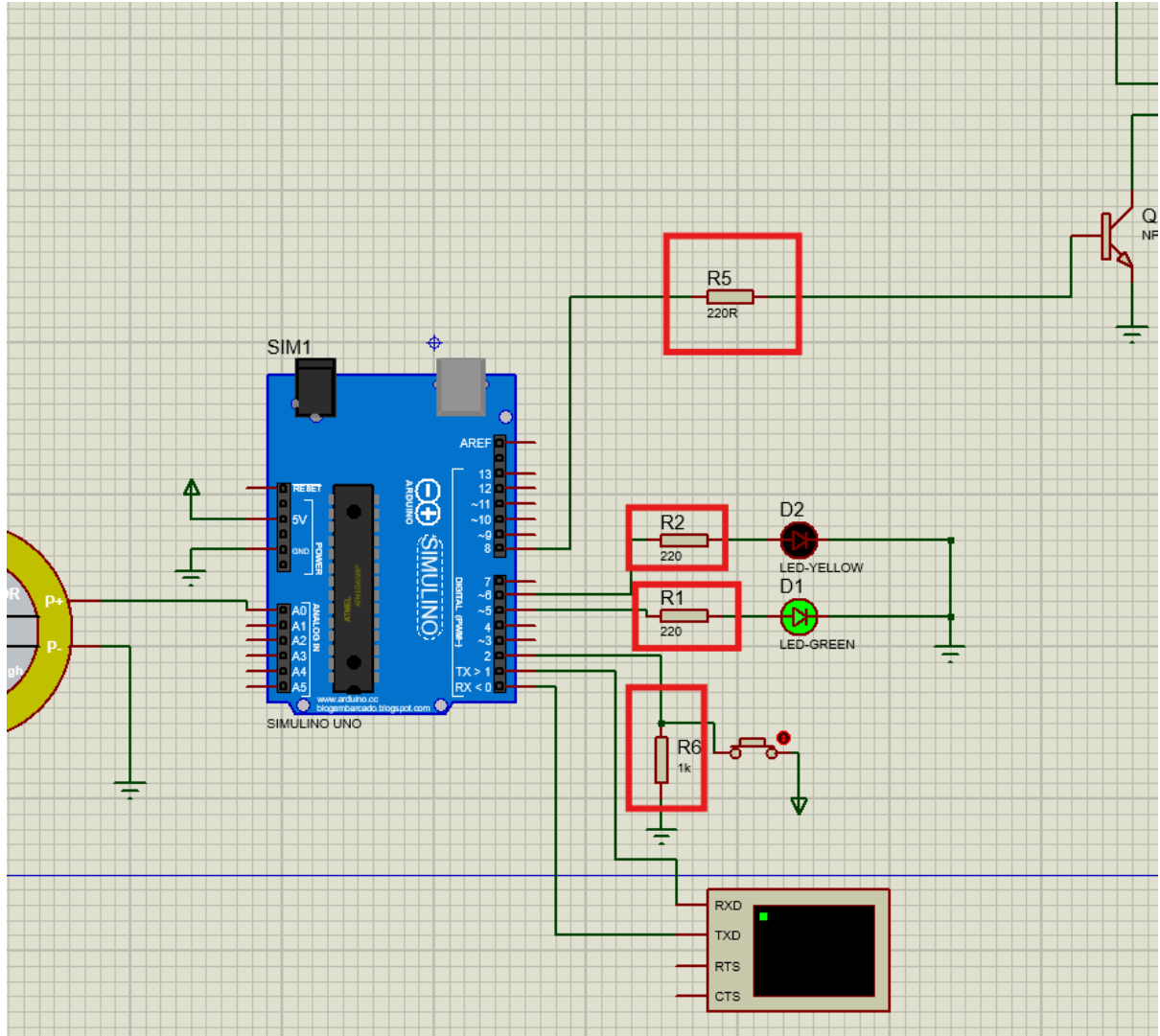


- **Sıfırlama Butonu (SW1):**
- Bir ucu **Pin 2**, diğer ucu GND.



#### Seri Monitör:

- Arduino'nun USB bağlantısı üzerinden bilgisayara bağlanır.
- **Baud Hızı:** Kodda `Serial.begin(9600);` ile ayarlanmıştır.



### Dirençler:

- LED'ler ve transistör girişinde dirençler kullanılarak uygun akım sağlanır.

## Deney Kodları

### 1. Kodun Genel Yapısı

Kodda aşağıdaki temel işlevler ve özellikler bulunmaktadır:

- **Piezoelektrik sensör** kullanılarak titreşim verilerinin ölçümü.
- Titreşim değerlerine göre **LED'ler** ve **hoparlör** ile uyarı verilmesi.
- Sistemi sıfırlamak için bir **buton** kullanılması.
- Titreşim seviyelerinin seri monitörde görüntülenmesi.

## 2. Tanımlamalar ve Ayarlar

### *Değişken Tanımlamaları:*

```
const int piezoPin = A0;           // Piezoelektrik sensörün bağlı  
olduğu pin  
const int speakerPin = 8;         // Hoparlör için dijital pin  
const int ledGreen = 5;           // Yeşil LED  
const int ledYellow = 6;          // Sarı LED  
const int resetButton = 2;        // Sıfırlama butonu  
  
int thresholdLow = 500;           // Orta risk eşiği  
int thresholdHigh = 800;          // Yüksek risk eşiği
```

- **piezoPin:** Piezoelektrik sensörün bağlı olduğu analog giriş pini (A0). Bu pin üzerinden titreşim değerleri okunur.
- **speakerPin:** Hoparlörün bağlandığı dijital pin (Pin 8). Hoparlör üzerinden sesli uyarı verilir.
- **ledGreen** ve **ledYellow:** LED'lerin bağlı olduğu dijital çıkış pinleri. Durumları göstermek için kullanılırlar.
- **resetButton:** Sistemi sıfırlayan butonun bağlı olduğu pin (Pin 2). Dahili pull-up direnci ile yapılandırılmıştır.
- **thresholdLow** ve **thresholdHigh:** Titreşim seviyelerini değerlendirmek için kullanılan eşik değerleridir.

## 3. setup() Fonksiyonu

```
void setup() {  
    pinMode(speakerPin, OUTPUT);    // Hoparlörü çıkış olarak ayarla  
    pinMode(ledGreen, OUTPUT);      // Yeşil LED'i çıkış olarak  
    ayarla  
    pinMode(ledYellow, OUTPUT);     // Sarı LED'i çıkış olarak  
    ayarla  
    pinMode(resetButton, INPUT_PULLUP); // Sıfırlama butonunu giriş  
    olarak ayarla (dahili pull-up aktif)  
  
    Serial.begin(9600); // Seri iletişimi başlat (9600 baud hızı)  
}
```

### Görevleri:



- Sistemdeki giriş ve çıkış pinlerini yapılandırır.
- **Hoparlör** ve **LED'ler** çıkış olarak tanımlanır.
- **Sıfırlama butonu**, dahili pull-up direnci etkinleştirilerek giriş olarak yapılandırılır (buton açıkken HIGH, basıldığında LOW seviyesindedir).
- **Seri Monitör** için haberleşme başlatılır.

#### 4. loop() Fonksiyonu

Bu ana döngü, sürekli olarak sensör değerlerini okur, uyarı koşullarını değerlendirir ve gerekli işlemleri gerçekleştirir:

```
void loop() {  
    int vibration = analogRead(piezoPin); // Sensör değerini oku  
  
    // Seri monitörde titreşim seviyesini görüntüle  
    Serial.print("Titresim Seviyesi: ");  
    Serial.println(vibration);  
    delay(100);  
  
    // Titreşim değerine göre uyarı seviyesi belirle  
    if (vibration > thresholdLow) {  
        triggerAlert(ledYellow); // Uyarı durumunu başlat  
    } else {  
        normalOperation(); // Normal durum işlemi  
    }  
  
    // Sıfırlama butonuna basıldığında tüm uyarıları kapat  
    if (digitalRead(resetButton) == LOW) {  
        resetSystem(); // Sistemi sıfırla  
    }  
  
    delay(1000); // Sensör okuma hızını kontrol etmek için kısa  
    gecikme  
}
```

#### Adım Adım Çalışma:

1. **Titreşim Ölçümü:** analogRead(piezoPin) ile piezo sensörden analog titreşim değeri okunur ve vibration değişkenine atanır.
2. **Seri Monitör Çıkışı:** Okunan titreşim değeri seri monitörde görüntülenir.

### 3. Uyarı Koşulları:

- a. Eğer titreşim değeri **thresholdLow** değerinden büyükse, `triggerAlert()` fonksiyonu çağrılır ve sarı LED ile hoparlör aktive edilir.
- b. Titreşim değeri düşükse, `normalOperation()` fonksiyonu çağrılır ve yalnızca yeşil LED yanar.

### 4. Sıfırlama Kontrolü:

- a. Sıfırlama butonuna basıldığında (LOW okunur), `resetSystem()` çağrılır ve tüm LED'ler ve hoparlör kapatılır.

## 5. Yardımcı Fonksiyonlar

### *triggerAlert() Fonksiyonu*

```
void triggerAlert(int ledPin) {  
    digitalWrite(ledGreen, LOW);        // Yeşil LED'i kapat  
    digitalWrite(ledYellow, LOW);       // Sarı LED'i kapat  
    digitalWrite(ledPin, HIGH);         // Belirtilen LED'i yak  
  
    digitalWrite(speakerPin, HIGH);     // Hoparlörü aç  
  
    Serial.print("Titresim var");        // Seri monitöre uyarıyı yazdır  
    Serial.println();  
}
```

### Görevleri:

- Belirli bir LED'i (örneğin sarı LED) aktif eder.
- Hoparlörü açar ve uyarı sesi verir.
- Uyarı durumunu seri monitöre yazdırır.

### *normalOperation() Fonksiyonu*

```
void normalOperation() {  
    digitalWrite(ledGreen, HIGH);       // Yeşil LED açık  
    digitalWrite(ledYellow, LOW);       // Sarı LED kapalı  
    digitalWrite(speakerPin, LOW);      // Hoparlör kapalı  
  
    Serial.println("Titresim yok");     // Seri monitöre normal durumu yazdır
```

}

#### Görevleri:

- Yeşil LED'i yakar (normal durum göstergesi).
- Sarı LED'i ve hoparlörü kapatır.
- Seri monitörde "Titreşim yok" mesajını gösterir.

#### *resetSystem() Fonksiyonu*

```
void resetSystem() {  
    digitalWrite(ledGreen, LOW);    // Yeşil LED kapalı  
    digitalWrite(ledYellow, LOW);   // Sarı LED kapalı  
    digitalWrite(speakerPin, LOW);   // Hoparlör kapalı  
  
    Serial.println("System Reset!"); // Seri monitöre sıfırlama mesajı  
    yazdır  
}
```

#### Görevleri:

- Tüm LED'leri ve hoparlörü kapatır.
- Seri monitöre "System Reset!" mesajını gönderir.

## 6. Çalışma Prensipleri

### 1. Normal Durum:

- a. Titreşim algılanmadığında yeşil LED yanar, hoparlör kapalıdır.
- b. Seri monitörde "Titreşim yok" yazısı gösterilir.

### 2. Titreşim Algılandığında:

- a. Titreşim değeri thresholdLow seviyesini aşarsa, sarı LED ve hoparlör aktif hale gelir.
- b. Seri monitörde "Titreşim var" mesajı görüntülenir.

### 3. Sıfırlama:

- a. Kullanıcı sıfırlama butonuna bastığında tüm uyarılar kapatılır.
- b. Seri monitöre "System Reset!" mesajı yazdırılır.

Bu kod, basit ama etkili bir **titreşim algılama ve uyarı sistemi** oluşturur. Sensör, LED, hoparlör ve buton gibi temel elemanlarla birden fazla çıktı tipi (görsel, işitsel, dijital) sağlar.