

Yağmur Algılama ve Alarm Sistemi

Deneyin Amacı:

Bu deneyde, bir yağmur sensöründen gelen dijital sinyal kullanılarak yağmur algılandığında bir LED'in yanması ve hoparlör aracılığıyla sesli bir uyarı verilmesi sağlanacaktır. Transistör, hoparlörün kontrolünde anahtar görevi görecek ve Arduino'nun dijital çıkış kapasitesini aşmadan daha yüksek akım gerektiren bileşenlerin kontrolünü öğretecektir.

Devrede Kullanılan Elemanlar ve Görevleri

1. Arduino UNO

- **Görevi:**
- Deneyin ana kontrol birimi olarak çalışır. Yağmur sensöründen gelen veriyi işler ve LED ile hoparlörü kontrol eder. Arduino, dijital giriş ve çıkışlar aracılığıyla devre elemanlarının senkronize çalışmasını sağlar.

2. Yağmur Sensörü (Dijital Çıkışlı)

- **Görevi:**

Ortamdaki yağmuru algılar ve bunu dijital bir sinyal (0 veya 1) olarak Arduino'ya iletir.

- **Çıkış Durumları:**

- ♣ **HIGH (1):** Yağmur algılandı.
- ♣ **LOW (0):** Yağmur algılanmadı.

3. LED

- **Görevi:**

Yağmur algılandığında görsel bir uyarı sağlamak için yanar.

- **Bağlantı Şeması:**

LED, Arduino'nun D13 pinine bağlanır. LED'in anot bacağına bir **220Ω direnç** eklenmiştir.

- **Neden 220Ω Direnç Kullanılır?**

LED'in zarar görmesini ve aşırı akım çekmesini önler. Akımı sınırlayarak LED'in güvenli çalışmasını sağlar.

4. Hoparlör

- **Görevi:**

Yağmur algılandığında sesli bir uyarı sağlamak için kullanılır. Hoparlör, bir transistör üzerinden kontrol edilir ve **tone()** fonksiyonu ile belirli bir frekansta ses çıkarır.

5. NPN Transistör (Örnek: BC547)

- **Görevi:**

Hoparlör gibi daha yüksek akım gerektiren elemanların kontrolü için bir anahtar görevi görür. Arduino'nun D11 çıkışındaki düşük akımı kontrol sinyali olarak kullanır ve hoparlöre yüksek akım sağlar.

- **Çalışma Mantığı:**

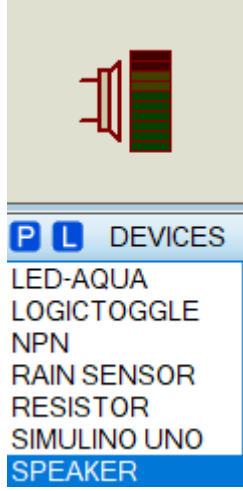
- o **Base (B):** Arduino'nun D11 pininden 220Ω direnç üzerinden kontrol sinyali alır.
- o **Collector (C):** Hoparlörün bir ucuna bağlanır.
- o **Emitter (E):** Arduino'nun GND pinine bağlanır.
- o Base'e sinyal geldiğinde transistör "açılır" ve hoparlör çalışır.

6. 220Ω Direnç (2 Adet)

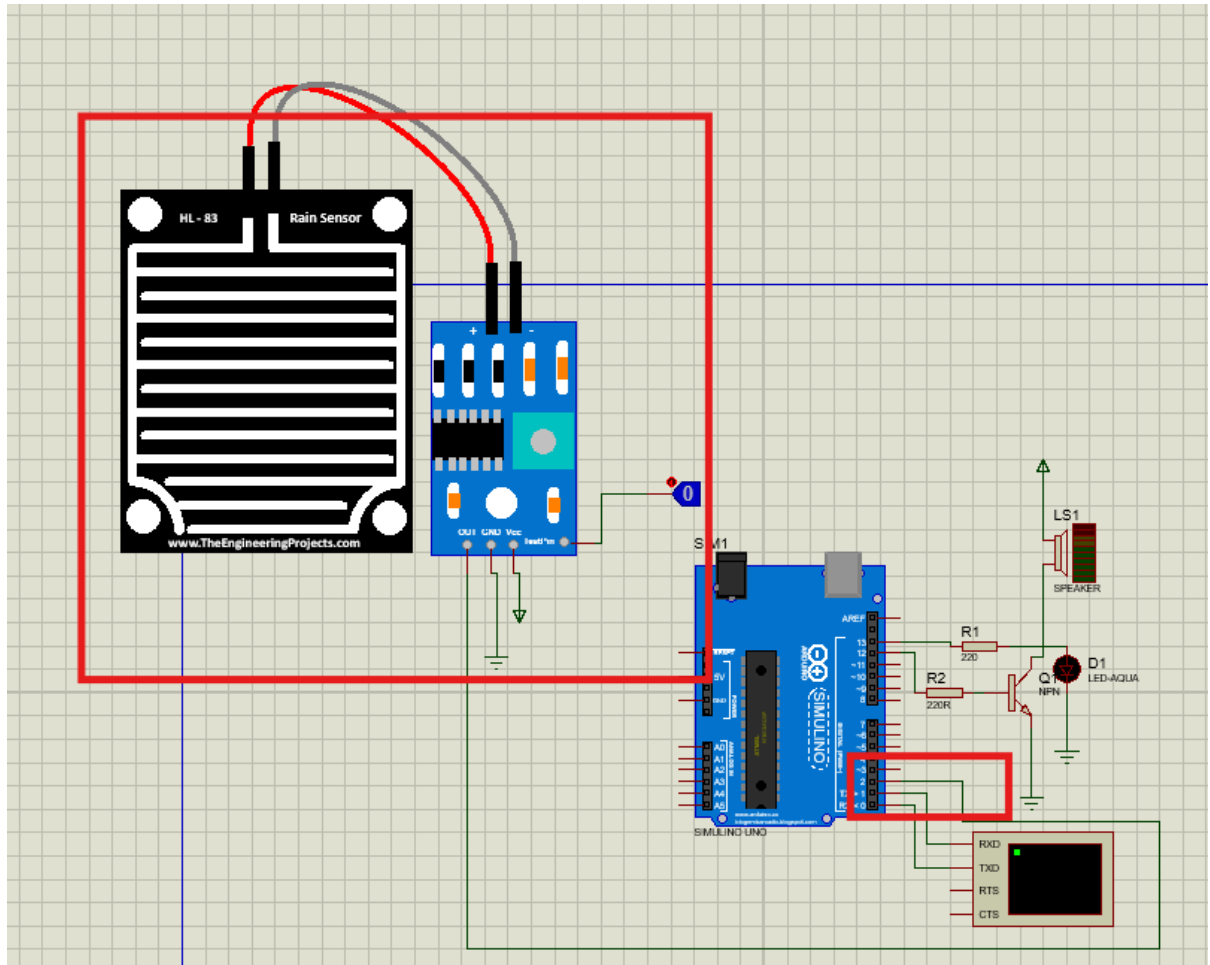
- **Görevleri:**

- o **LED'in Önünde:**
 - ♣ LED'e aşırı akım gitmesini önler. LED'in ömrünü uzatarak güvenli çalışmasını sağlar.
- o **Transistörün Base Ucunda:**
 - ♣ Transistörün base bacağına gelen akımı sınırlar. Arduino'nun dijital pinine zarar gelmesini engeller ve transistörün güvenli çalışmasını sağlar

Devre Elemanlarının Bağlantıları Ve Deneyin Yapılışı

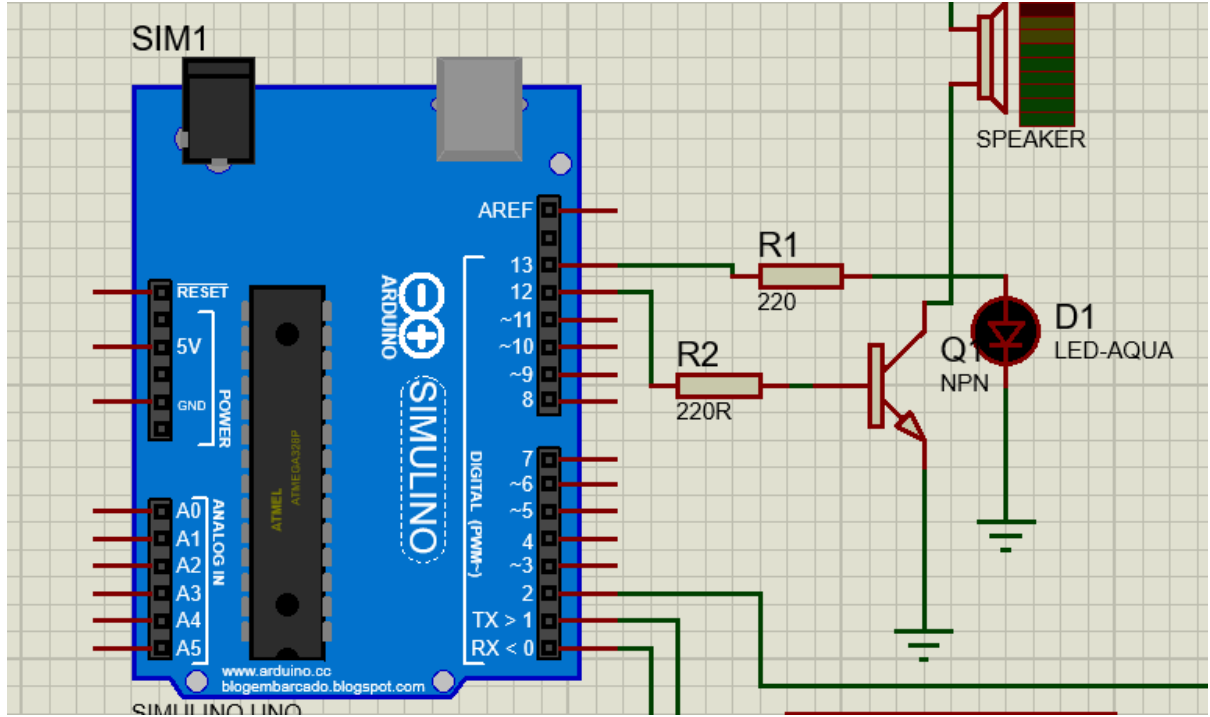


Devre elemanları seçilerek deneye başlanır.



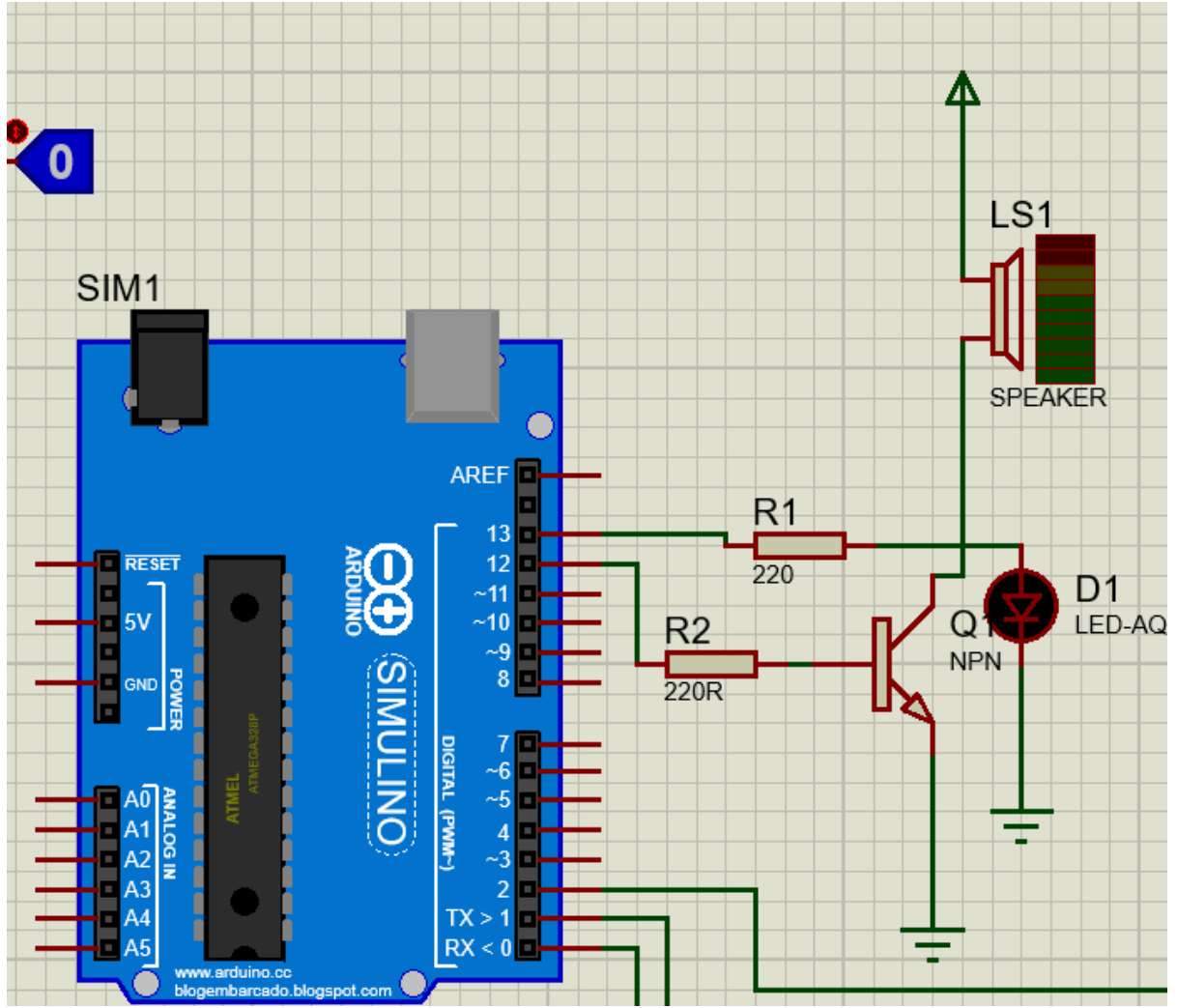
Yağmur Sensörü

- **VCC (Güç):** Arduino'nun **5V** pinine bağlanır.
- **GND (Toprak):** Arduino'nun **GND** pinine bağlanır.
- **OUT (Dijital Çıkış):** Arduino'nun **D2** pinine bağlanır



LED

- **Anot (+):** Arduino'nun **D13** pinine bağlanır.
- **Katot (-):** Bir **220Ω direnç** üzerinden Arduino'nun **GND** pinine bağlanır.



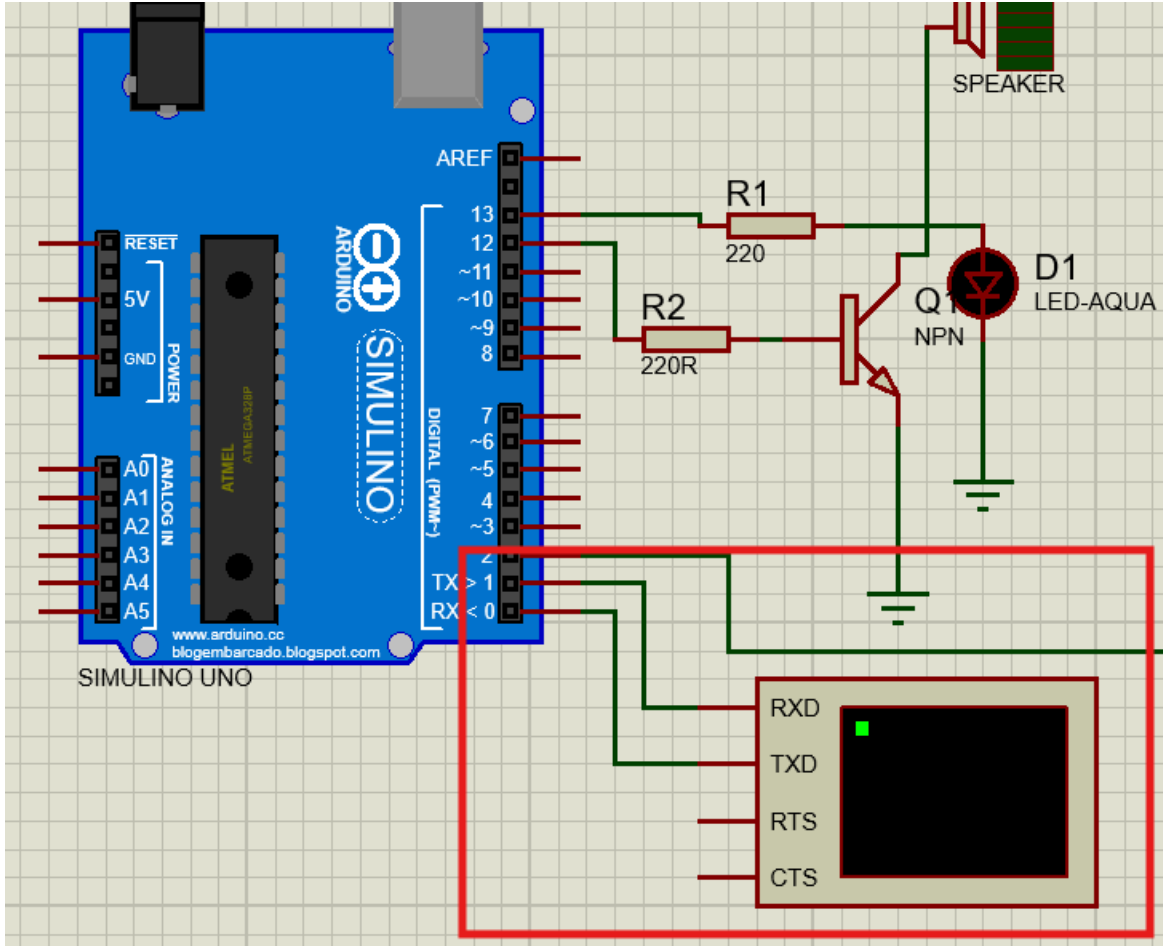
Hoparlör

- **Birinci Ucu:** Arduino'nun **5V** pinine bağlanır (güç sağlar).
- **İkinci Ucu:** Transistörün **Collector (C)** bacağına bağlanır.

5. NPN Transistör (Örnek: BC547)

- **Base (B):**
 - o Arduino'nun **D12** pininden gelen kontrol sinyali bir **220Ω direnç** üzerinden Base'e bağlanır.

- **Collector (C):**
 - Hoparlörün ikinci ucuna bağlanır. Bu bağlantı üzerinden hoparlöre güç akışı kontrol edilir.
- **Emitter (E):**
 - Arduino'nun **GND** pinine bağlanır. Bu bağlantı, transistörün akım devresini tamamlar.



6. Seri Monitör

Arduino UNO'nun **RX (D0)** ve **TX (D1)** pinleri, dahili olarak USB bağlantısına bağlıdır. Bu nedenle, harici bir bağlantı gerekmez.

Deney Kodları

1. Değişken Tanımlamaları:

```
const int ledPin = 13;          // LED'in bağlı olduğu pin tanımlanır.  
const int speakerPin = 12;     // Hoparlörün bağlı olduğu pin  
tanımlanır.
```

- **ledPin:** Arduino'nun D13 pinine bağlı LED'in kontrolü için tanımlanır.
- **speakerPin:** Arduino'nun D12 pinine bağlı hoparlörün kontrolü için tanımlanır.

2. setup() Fonksiyonu:

Bu fonksiyon, Arduino'nun başlangıç ayarlarını yapar.

```
void setup() {  
    pinMode(ledPin, OUTPUT);  
    pinMode(speakerPin, OUTPUT);  
    pinMode(2 , INPUT);          // Yağmur sensörünün bağlı olduğu  
pin giriş olarak ayarlanır.  
    Serial.begin(9600);          // Seri haberleşme başlatılır (9600  
baud hızı).  
}
```

- **pinMode(ledPin, OUTPUT):** LED'in bağlı olduğu pin çıkış olarak ayarlanır.
- **pinMode(speakerPin, OUTPUT):** Hoparlörün bağlı olduğu pin çıkış olarak ayarlanır.
- **pinMode(2, INPUT):** Yağmur sensörünün bağlı olduğu pin giriş olarak ayarlanır. Sensörden gelen dijital sinyal bu pin üzerinden okunur.
- **Serial.begin(9600):** Seri monitör üzerinden veri gönderimi ve alımı için haberleşme başlatılır.

3. Loop() Fonksiyonu:

Bu fonksiyon sürekli çalışarak yağmur sensöründen gelen veriyi işler ve bağlı elemanları kontrol eder.

a. Yağmur Durumunu Okuma:

`int rain = digitalRead(2); // Sensörden gelen dijital sinyal okunur.`

- **`digitalRead(2)`**: Arduino'nun D2 pinine bağlı sensörün durumu okunur.
 - **HIGH (1)**: Yağmur algılandı.
 - **LOW (0)**: Yağmur algılanmadı.
- Okunan değer, **rain** değişkenine atanır.

b. Yağmur Algılandığında (HIGH):

```
if (rain == 1) {  
    Serial.println("Its Raining"); // Seri monitöre "Its Raining"  
    yazdırılır.  
    digitalWrite(ledPin, HIGH); // LED yanar.  
    digitalWrite(speakerPin, HIGH); // Hoparlör çalışır.  
    delay(500); // 500 ms bekleme süresi.  
}
```

- **`Serial.println("Its Raining")`**: Seri monitörde yağmurun algılandığını bildiren mesaj yazdırılır.
- **`digitalWrite(ledPin, HIGH)`**: LED yakılarak görsel uyarı verilir.
- **`digitalWrite(speakerPin, HIGH)`**: Hoparlör aktif edilerek işitsel uyarı sağlanır.
- **`delay(500)`**: 500 milisaniyelik bekleme süresi.

c. Yağmur Algılanmadığında (LOW):

```
else if (rain == 0) {  
    Serial.println("No Rain"); // Seri monitöre "No Rain"  
    yazdırılır.  
    digitalWrite(ledPin, LOW); // LED söndürülür.  
    digitalWrite(speakerPin, LOW); // Hoparlör kapatılır.  
    delay(500); // 500 ms bekleme süresi.  
}
```

- **`Serial.println("No Rain")`**: Seri monitörde yağmurun algılanmadığını bildiren mesaj yazdırılır.
- **`digitalWrite(ledPin, LOW)`**: LED söndürülür, görsel uyarı kaldırılır.

- **digitalWrite(speakerPin, LOW):** Hoparlör kapatılır, işitsel uyarı durdurulur.
- **delay(500):** 500 milisaniyelik bekleme süresi.

Çalışma Mantığı:

1. **Sensör Çıkışı HIGH (1) ise:**
 - a. Seri monitörde **"Its Raining"** yazdırılır.
 - b. LED yanar, hoparlör çalışır (uyarı verilir).
2. **Sensör Çıkışı LOW (0) ise:**
 - a. Seri monitörde **"No Rain"** yazdırılır.
 - b. LED söner, hoparlör kapanır (uyarı kaldırılır).
3. **Bu süreç sürekli tekrar eder:** Arduino, sensör durumunu sürekli kontrol eder ve uyarı durumunu günceller.