

# Akıllı Termostat Deneyi

## Deneyin Amacı

Bu deneyde, sıcaklık değerlerini ölçerek bir ısıtıcı veya soğutucunun kontrol edilmesi sağlanacaktır. LM35 sıcaklık sensöründen alınan veriler, LCD ekranında görüntülenecek ve bir röle yardımıyla ısıtıcı/sogutucu devreye girecektir. Deney, Arduino'nun dijital ve analog pinlerini, sıcaklık sensörü, LCD ekran, röle ve transistör gibi temel elektronik bileşenleri birleştirilerek bir akıllı sistemin nasıl çalıştığını göstermeyi amaçlar.

## Devrede Kullanılan Elemanlar ve Görevleri

### 1. Arduino UNO

- **Görevi:**
- Sistemin ana kontrol birimidir. Sıcaklık sensöründen gelen verileri işler, LCD ekrana yazdırır ve röleyi kontrol eder.
  - o Analog pinden sıcaklık verisi alır.
  - o Dijital çıkışları ile röleye ve LCD ekrana sinyal gönderir.

### 2. LM35 Sıcaklık Sensörü

- **Görevi:**
- Çevrenin sıcaklığını ölçer ve analog bir voltaj sinyali olarak Arduino'ya iletir.
- o 10 mV/°C hassasiyete sahiptir.
  - o Ölçülen sıcaklık, Arduino tarafından Celsius'a dönüştürülür.

### 3. 16x2 LCD Ekran

- **Görevi:**
- Kullanıcıya sıcaklık bilgisini ve sistem durumunu gösterir (ör. "Temp: 23.5°C", "Heater ON").
- o LCD ekran, Arduino üzerinden kontrol edilir ve 16 sütun, 2 satır şeklinde bilgi görüntüler.

### 4. Röle Modülü

- **Görevi:**
- Yüksek güç gerektiren cihazları (ör. ısıtıcı veya soğutucu) kontrol etmek için düşük güçle çalışan bir anahtar görevi görür.

- o Arduino'dan gelen sinyale göre elektrik devresini açar veya kapatır.

### 5. NPN Transistör (ör. BC547)

- **Görevi:**

Röleyi kontrol etmek için Arduino'nun çıkış voltajını yükseltir.

- o Arduino'nun dijital çıkışı doğrudan röleyi çalıştırmaya yeterli olmadığından, transistör bir anahtar gibi davranır.

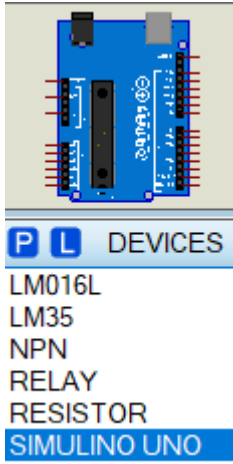
### 6. 10k Ohm Direnç

- **Görevi:**

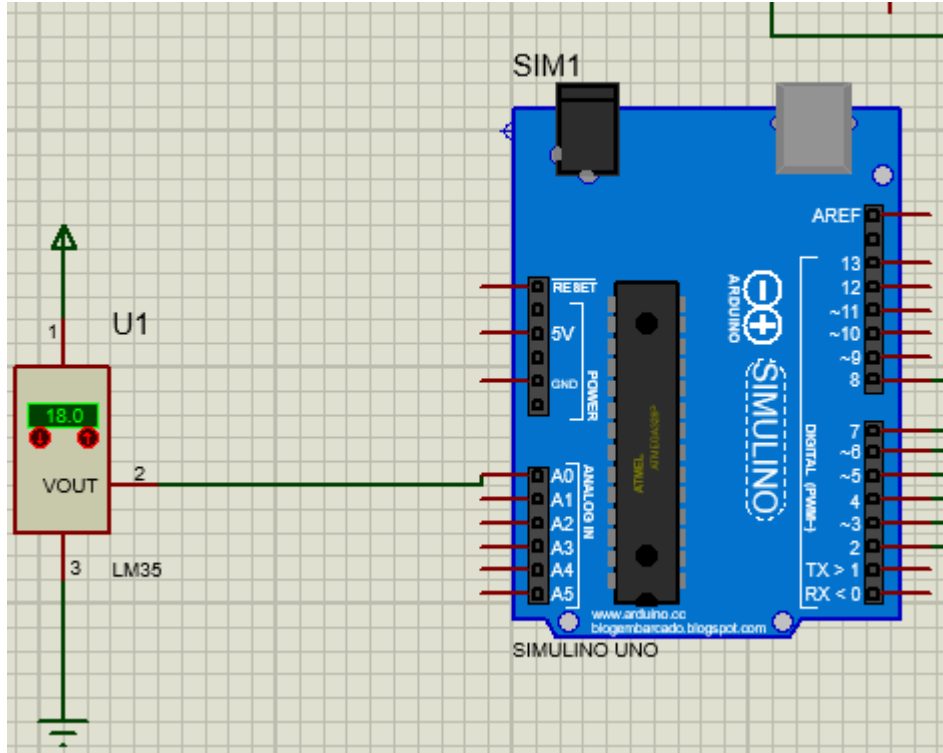
Transistörün tabanına gelen akımı sınırlayarak transistörün doğru çalışmasını sağlar.

- o Transistörün korunması ve doğru anahtarlama işleminin yapılması için kullanılır.
- o

## Devre Elemanlarının Bağlantıları Ve Deneyin Yapılışı

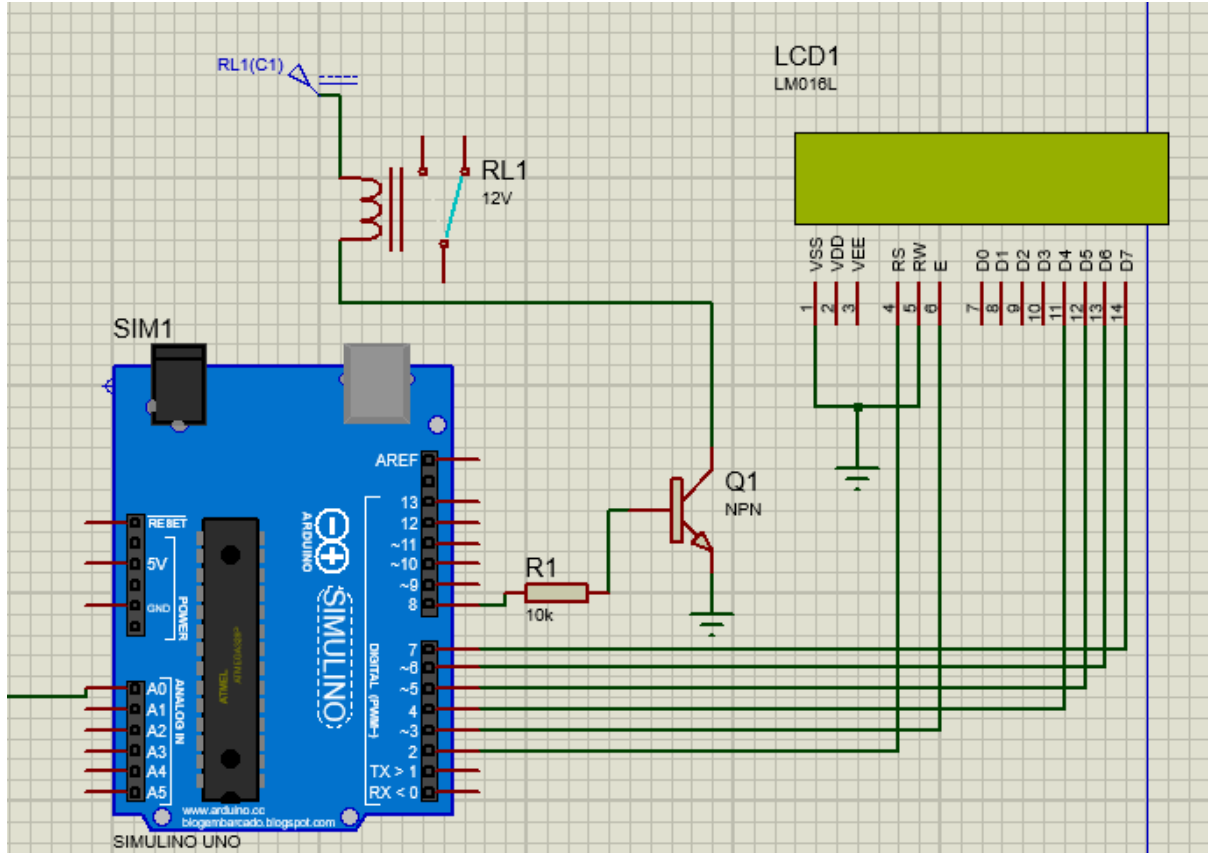


Devre elemanları seçilerek deneye başlanır.



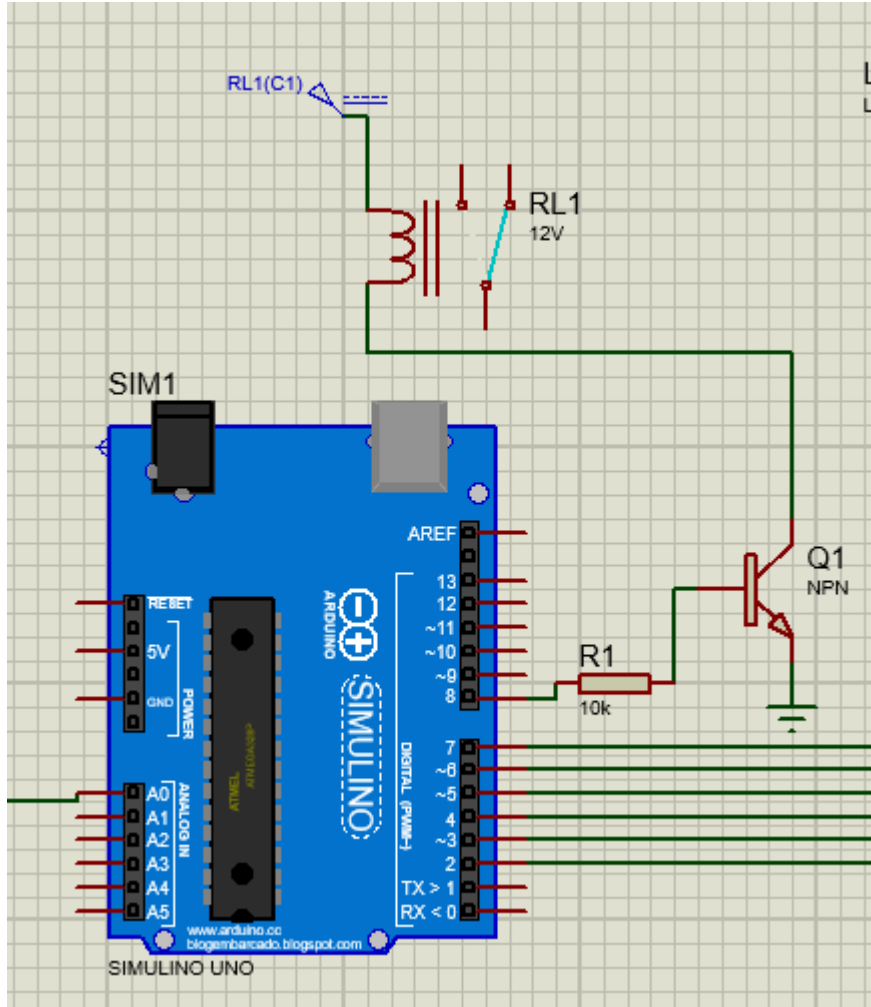
### 1. Arduino ile LM35 Bağlantısı

- **LM35'in VCC pini (1. pin):** Arduino'nun 5V çıkışına bağlanır.
- **LM35'in GND pini (3. pin):** Arduino'nun GND pinine bağlanır.
- **LM35'in VOUT pini (2. pin):** Arduino'nun analog giriş pini olan **A0**'a bağlanır.



## 2. Arduino ile LCD Ekran Bağlantısı

- **LCD'nin VSS pini (1. pin):** Arduino'nun GND pinine bağlanır.
- **LCD'nin VDD pini (2. pin):** Arduino'nun 5V çıkışına bağlanır.
- **LCD'nin RW pini (5. pin):** Arduino'nun GND pinine bağlanır (sadece yazma modunda kullanılacağı için).
- **LCD'nin RS pini (4. pin):** Arduino'nun dijital pin 2'sine bağlanır.
- **LCD'nin E pini (6. pin):** Arduino'nun dijital pin 3'üne bağlanır.
- **LCD'nin D4 pini (11. pin):** Arduino'nun dijital pin 4'üne bağlanır.
- **LCD'nin D5 pini (12. pin):** Arduino'nun dijital pin 5'ine bağlanır.
- **LCD'nin D6 pini (13. pin):** Arduino'nun dijital pin 6'sına bağlanır.
- **LCD'nin D7 pini (14. pin):** Arduino'nun dijital pin 7'sine bağlanır.



### Arduino ile Röle ve Transistör Bağlantısı

- **Rölenin GND pini:** Arduino'nun GND pinine bağlanır.
- **Rölenin VCC pini:** Harici bir 5V güç kaynağına veya Arduino'nun 5V pinine bağlanır (röle modülüne bağlı olarak).
- **Rölenin IN pini:** NPN transistörün **kolektör** ucuna bağlanır.
- **NPN Transistörün Tabanı:** Arduino'nun dijital pin 8'ine, 10k direnç üzerinden bağlanır.
- **NPN Transistörün Emitörü:** GND'ye bağlanır.

### Çalışma Prensibi

1. **Sıcaklık Okuma:**

2. LM35, sıcaklığı analog bir voltaj değeri olarak ölçer. Arduino'nun analog pinine bağlı LM35, sıcaklık değerini okur ve Celsius'a dönüştürür.
3. **LCD Ekran:**

Ölçülen sıcaklık değeri, LCD ekranın ilk satırında görüntülenir.

4. **Röle Kontrolü:**

- a. Sıcaklık **20°C'nin altına düştüğünde**, röle aktif hale gelir ve "Heater ON" mesajı görüntülenir.
- b. Sıcaklık **25°C'nin üzerine çıktığında**, röle pasif hale gelir ve "Cooler ON" mesajı görüntülenir.
- c. Aralıkta kaldığında ise "System Idle" mesajı gösterilir.

5. **Kod Akışı:**

Arduino, sıcaklık ölçümü ve röle kontrolünü 1 saniyelik döngülerle tekrar eder.

## Deney kodları ve açıklaması

### 1. Kütüphane ve Değişken Tanımlamaları

```
#include <LiquidCrystal.h>
```

- **Açıklama:**
- Arduino'nun 16x2 LCD ekranını kontrol etmek için kullanılan standart bir kütüphanedir. LCD ile iletişim kurmayı sağlar.

```
const int rs=2 , en = 3, d4 = 4, d5 = 5, d6 = 6 , d7 = 7;  
LiquidCrystal lcd(rs , en, d4, d5, d6, d7);
```

- **Açıklama:**
  - o **rs, en, d4, d5, d6, d7:** LCD'nin dijital pinlerle bağlantısını temsil eder.
  - o **LiquidCrystal lcd:** LCD'nin bu pinler üzerinden kontrol edilmesi için bir nesne oluşturur.

```
float temp;  
const int relayPin = 8;
```

- **Açıklama:**
  - o **temp:** LM35'ten alınan sıcaklık değerini saklayan bir değişken.

- o **relayPin:** Röle modülüne bağlı olan Arduino dijital pinini tanımlar.

## 2. setup() Fonksiyonu

```
void setup() {  
  Serial.begin(9600);  
  analogReference(INTERNAL);  
  lcd.begin(16, 2);  
  pinMode(relayPin, OUTPUT);  
  digitalWrite(relayPin, LOW);  
}
```

- **Açıklama:**

- o **Serial.begin(9600);**

Arduino'nun seri haberleşme protokolünü başlatır. Sıcaklık verileri seri port üzerinden gözlemlenebilir.

- o **analogReference(INTERNAL);**

LM35 için referans voltajını iç referans (1.1V) olarak ayarlar. Bu, düşük voltajlı sensörler için daha hassas ölçüm sağlar.

- o **lcd.begin(16, 2);**

16 sütun ve 2 satırdan oluşan LCD ekranı başlatır.

- o **pinMode(relayPin, OUTPUT);**

Röle pinini çıkış olarak ayarlar.

- o **digitalWrite(relayPin, LOW);**

Rölenin başlangıçta kapalı (pasif) olmasını sağlar.

## 3. loop() Fonksiyonu

### *Sıcaklık Verisinin Okunması ve Hesaplanması*

```
temp = analogRead(A0);  
temp = temp * 1100 / (1024 * 10);
```

- **Açıklama:**

- o **analogRead(A0);**

LM35'ten gelen analog sıcaklık verisini okur. Bu veri 0 ile 1023 arasında bir değerdir.

- o **temp \* 1100 / (1024 \* 10);**

Okunan analog veriyi sıcaklığa dönüştürür:

- ♣ LM35, sıcaklık başına 10 mV üretir.
- ♣ Arduino'nun ADC'si (Analog-Dijital Dönüştürücü), 1024 adımlık bir çözünürlüğe sahiptir.
- ♣ 1.1V referans kullanıldığından, her bir adım yaklaşık 1.074 mV'a eşittir.

Bu formül ile sıcaklık °C cinsinden hesaplanır.

### ***LCD Ekranda Sıcaklık Bilgisinin Gösterilmesi***

```
lcd.setCursor(0, 0);  
lcd.print("Temp: ");  
lcd.print(temp, 1);  
lcd.print(" C ");
```

- **Açıklama:**

- o `lcd.setCursor(0, 0);`

LCD ekranın 1. satırının başına imleci yerleştirir.

- o `lcd.print("Temp: ");`

Ekrana "Temp:" metnini yazar.

- o `lcd.print(temp, 1);`

Ölçülen sıcaklık değerini ondalık hassasiyetle yazar. Örneğin: "23.4".

- o `lcd.print(" C ");`

Sıcaklık birimini gösterir ve ekranın temiz görünmesini sağlar.

### ***Röle Kontrolü ve Sistem Durumu***

```
if (temp < 20) {  
    digitalWrite(relayPin, HIGH);  
    lcd.setCursor(0, 1);  
    lcd.print("Heater ON    ");  
} else if (temp > 25) {  
    digitalWrite(relayPin, LOW);  
    lcd.setCursor(0, 1);  
    lcd.print("Cooler ON    ");  
} else {  
    lcd.setCursor(0, 1);  
    lcd.print("System Idle  ");  
    digitalWrite(relayPin, LOW);  
}
```



}

- **Açıklama:**

Bu bölüm, sıcaklığa göre röleyi kontrol eder ve durumu LCD ekranına yazar.

- o **Sıcaklık 20°C'nin altındaysa:**

- ♣ **digitalWrite(relayPin, HIGH);**

Röle aktif hale getirilir (örneğin, bir ısıtıcı çalıştırılır).

- ♣ **lcd.print("Heater ON");**

LCD ekranın 2. satırına "Heater ON" yazılır.

- o **Sıcaklık 25°C'nin üzerindeyse:**

- ♣ **digitalWrite(relayPin, LOW);**

Röle pasif hale getirilir (örneğin, ısıtıcı kapatılır).

- ♣ **lcd.print("Cooler ON");**

LCD ekranın 2. satırına "Cooler ON" yazılır.

- o **Sıcaklık 20°C ile 25°C arasındaysa:**

- ♣ Röle kapalı tutulur ve sistemin bekleme durumunda olduğu belirtilir: **lcd.print("System Idle");**

### ***Döngü ve Ekranın Temizlenmesi***

```
delay(1000);
```

```
lcd.clear();
```

- **Açıklama:**

- o **delay(1000);**

Kodun her döngüde 1 saniye beklemesini sağlar.

- o **lcd.clear();**

LCD ekranı bir sonraki ölçüm döngüsü için temizler.