

# **IOT project**

By

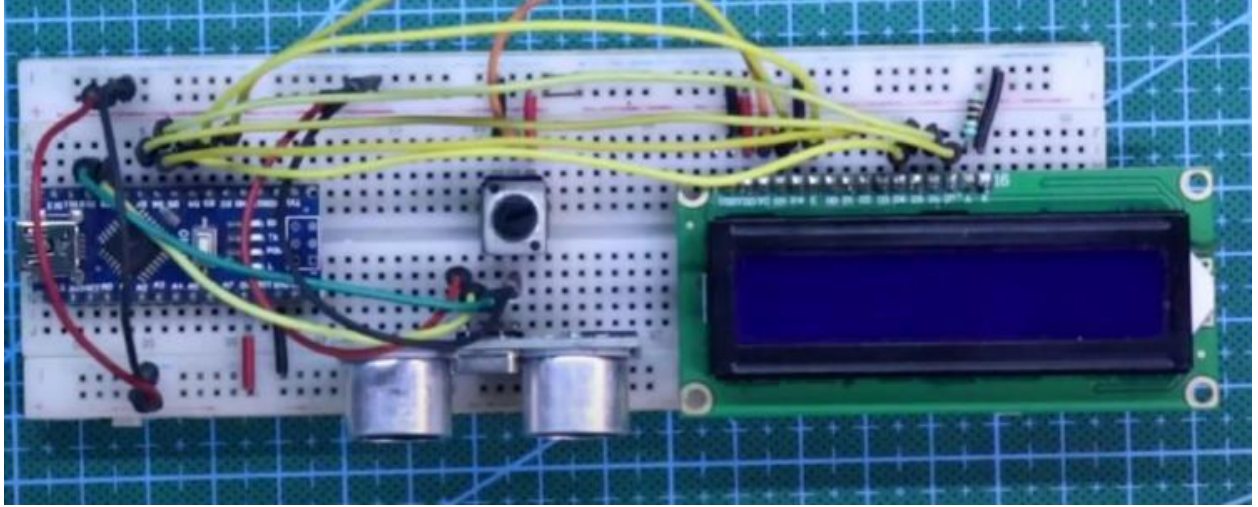
AbdulSamet Gök

## İçindekiler

• Mevcut IoT Projesi.....	3
• Projenin İşlevi.....	3
• Projede Kullanılan Malzemeler.....	4
• Devre Bağlantıları.....	4
• Projede Kullanılan Kod.....	5
• Çalışma Prensibi.....	10
• Ana İşlevler.....	10
• Kullanım Alanları ve Uygulamalar.....	10
• Çalışan Devre Kanıtı.....	10
• Genel Bakış.....	11

## Mevcut IoT Projesi:

- **Arduino ile Geliştirilen HC-SR04 Ultrasonik Mesafe Sensörü ve 16x2 LCD Projesi.**



## Projenin İşlevi:

Bu projede HC-SR04 ultrasonik mesafe sensörü kullanılarak bir nesnenin uzaklığı ölçülmekte ve ölçülen mesafe Arduino ile kontrol edilen 16x2 LCD ekranda santimetre ve inç cinsinden görüntülenmektedir.

## Projede Kullanılan Malzemeler:

- Arduino Nano
- Breadboard
- 16x2 LCD Ekran
- HC-SR04 Ultrasonik Mesafe Sensörü
- Buton
- Bağlantı Kabloları
- Dirençler

## Devre Bağlantıları:

### 1. HC-SR04 Ultrasonik Sensör → Arduino Nano

HC-SR04 Pin	Arduino Nano Pin	Açıklama
VCC	5V	Güç kaynağı (5V)
GND	GND	Toprak
TRIG	Dijital Pin	Tetikleme pini (Arduino'dan dijital çıkış)
ECHO	Dijital Pin	Echo pini (Arduino'ya dijital giriş)

### 2. 16x2 LCD Ekran → Arduino Nano

LCD Pin	Arduino Nano Pin	Açıklama
VSS	GND	Toprak
VDD	5V	Güç kaynağı (5V)
RS	Dijital Pin	Register seçimi
RW	GND	Yazma modu (Write)
E	Dijital Pin	Enable pini
D4	Dijital Pin	Veri pini 4
D5	Dijital Pin	Veri pini 5
D6	Dijital Pin	Veri pini 6
D7	Dijital Pin	Veri pini 7
A (LED+)	5V üzerinden direnç	LCD arka ışık anodu (~220Ω direnç ile)
K (LED-)	GND	LCD arka ışık katodu

### 3. Buton → Arduino Nano

- Bir ucu dijital giriş pinine, diğer ucu GND'ye bağlanır.

#### Projede Kullanılan Kod:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

const int trigPin = 9;

const int echoPin = 10;

long duration;

int distanceCm, distanceInch;

void setup() { lcd.begin(16,2);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() { digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distanceCm= duration*0.034/2;
```

---

```
distanceInch = duration*0.0133/2;
```

---

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

---

```
lcd.print("Distance: ");
```

---

```
lcd.print(distanceCm);
```

---

```
lcd.print(" cm"); delay(10);
```

---

```
lcd.setCursor(0,1);
```

---

```
lcd.print("Distance: ");
```

---

```
lcd.print(distanceInch);
```

---

```
lcd.print("inch");
```

---

```
delay(10);
```

---

```
}
```

---

## Proje Geliştirme Fikri:

- Ekipman sınırlamaları nedeniyle projeden buton çıkarıldı.
- Nesnenin mesafesi 10 cm'den az olduğunda kırmızı LED, 10 cm'den fazla olduğunda yeşil LED yanacak şekilde proje geliştirildi.
- Devre ve kod, buton olmadan çalışacak şekilde güncellendi.

## Projede Kullanılan Malzemeler:

- Arduino Uno R3
- Breadboard
- 16x2 LCD Ekran
- HC-SR04 Ultrasonik Mesafe Sensörü
- 220  $\Omega$  Direnç
- Bağlantı kabloları (Jumper Wires)
- USB Kablosu
- LED'ler (Yeşil, Kırmızı)

### 4. Arduino Uno → 16x2 LCD

LCD Pin	Arduino Pin	Açıklama
VSS	GND	Toprak
VDD	5V	Güç
RS	D2	Register seçimi
RW	GND	Yazma modu
E	D3	Enable pini
D4	D4	Veri pini 4
D5	D5	Veri pini 5
D6	D6	Veri pini 6
D7	D7	Veri pini 7
A (LED+)	5V üzerinden direnç	LCD arka ışık anodu (~220 $\Omega$ direnç ile)
K (LED-)	GND	LCD arka ışık katodu

## 1. Kod Açıklaması:

Kütüphane ve pin tanımlamaları yapıldı.

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

```
const int trigPin = 9;
```

```
const int echoPin = 10;
```

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
```

- LCD'yi kontrol etmek için LiquidCrystal kütüphanesi kullanıldı.
- Ultrasonik sensör için trigPin (çıkış) ve echoPin (giriş) tanımlandı.
- LCD, 6 pin üzerinden yapılandırıldı.

## 2. Setup Fonksiyonu.

```
void setup() { lcd.begin(16, 2);
```

```
pinMode(trigPin, OUTPUT);
```

```
pinMode(echoPin, INPUT);
```

```
lcd.print("Distance:");
```

- LCD başlatıldı ve sensör pinleri yapılandırıldı.
- Başlangıçta LCD ekranda "Distance:" yazısı gösterildi.

## 3. Loop Fonksiyonu İşleyişi.

1. Sensör tetikleme işlemi yapıldı:

```
digitalWrite(trigPin, LOW);
```

```
delayMicroseconds(2);
```

```
digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);
```



- Ölçümü başlatmak için HC-SR04 sensörünün trig pinine 10 mikrosaniyelik bir darbe gönderildi.

## **2. Echo pininden gelen sinyalin süresi ölçüldü.**

```
duration = pulseIn(echoPin, HIGH);
```

- pulseIn() fonksiyonu, echo pininin HIGH (yüksek) durumda kaldığı süreyi mikrosaniye cinsinden ölçer.

## **3. Mesafe santimetre ve inç olarak hesaplandı.**

```
distance_cm = duration * 0.034 / 2;
```

```
distance_inch = distance_cm / 2.54;
```

- Hesaplama, ses dalgasının nesneye gidip geri dönmesi nedeniyle 2'ye bölünerek yapıldı.

## **4. Ölçülen mesafe değerleri LCD ekranda görüntülendi.**

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

```
lcd.print("CM: ");
```

```
if (distance_cm >= 2 && distance_cm <= 400) {
```

```
lcd.print(distance_cm, 1);
```

```
} else {
```

```
lcd.print("Out of range");
```

```
}
```

- Ölçülen mesafe HC-SR04 sensörünün geçerli aralığında (2–400 cm) mı kontrol edildi.
- Geçersiz ölçümler için LCD ekranda “Out of range” mesajı gösterildi.

## Çalışma Prensipleri:

- HC-SR04 ultrasonik sensör, nesneden geri dönen yankının süresini ölçmek için ultrasonik bir darbe gönderir.
- Arduino, ölçülen süreyi kullanarak mesafeyi şu formülle hesaplar:  
**Mesafe = (Zaman × Ses Hızı) / 2**
- Hesaplama hem santimetre hem de inç cinsinden yapılır.

Ölçülen değerler 16x2 LCD ekranda gerçek zamanlı olarak gösterilir; ilk satırda mesafe santimetre, ikinci satırda inç cinsinden görüntülenir.

## Ana İşlevler:

- **Mesafe Ölçümü:** HC-SR04 ultrasonik sensör, nesneden geri dönen yankının süresini ölçer ve Arduino bu süreyi kullanarak mesafeyi hesaplar.
- **Gerçek Zamanlı Gösterim:** Ölçülen mesafe 16x2 LCD ekranda gerçek zamanlı olarak santimetre ve/veya inç cinsinden görüntülenir.
- **Mikrodenetleyici Kontrolü:** Arduino, sensörü kontrol eder, veriyi işler ve LCD ekranı günceller.
- **LED Göstergeler:** Nesnelerin uzaklığı hakkında görsel bir gösterge sağlar.
- **Alternatif Çıkışlar:** LED'ler yerine, nesnelere olan yakınlığa bağlı olarak sesli uyarı vermek için buzzer'lar kullanılabilir.

## Kullanım Alanları ve Uygulamalar:

- Robotik uygulamalarda engel tespiti
- Park asistan sistemleri
- Seviye ölçümü (ör. su tankı)
- Güvenlik ve otomasyon (ör. izinsiz giriş veya varlık tespiti)
- Sensör arayüzü ve mesafe ölçümünü göstermek için eğitim projeleri

## Çalışan Devre Kanıtı:

[Çalışan Devre Fotoğrafı ve Videoları için Drive Linki](#)

## **Genel Bakış:**

Bu proje, ultrasonik dalgalar kullanarak temassız mesafe ölçümü yapmanın pratik bir yolunu göstermektedir. Ölçülen değerlerin gerçek zamanlı olarak görsel çıktısının sağlanması, projeyi eğitim amaçlı ve robotik, otomasyon ile ölçüm sistemleri gibi gerçek dünya uygulamaları için değerli bir araç haline getirmektedir.