



**HİTİT**  
ÜNİVERSİTESİ

Mühendislik Fakültesi  
Bilgisayar Mühendisliği

2023-2024 Bahar Dönemi

**BIL 306 – Mikrodenetleyiciler Laboratuvarı**

# Uygulama Ödevi

## Mikrodenetleyici Tabanlı Sistem Tasarımı

	1. Öğrenci	2. Öğrenci
Ad	<b>Oğuzhan</b>	<b>Muhammet Akif</b>
Soyad	<b>YALÇIN</b>	<b>AYAN</b>
Ödev Linki	<a href="https://www.youtube.com/watch?v=UFVGz6wSowM">https://www.youtube.com/watch?v=UFVGz6wSowM</a>	

Dr. Öğr. Üyesi Serkan DIŞLİTAŞ  
Ders Sorumlusu

2024  
Çorum / Türkiye

# DHT11 Sensörü ile Sıcaklık ve Nem Ölçümü Projesi Raporu

## 1. Proje Özeti

Bu proje, DHT11 sıcaklık ve nem sensörü kullanarak ortamın sıcaklık ve nem değerlerini ölçmeyi amaçlamaktadır. Ölçülen değerler 16x2 LCD ekranda ve Blynk uygulaması üzerinden mobil ve web arayüzlerinde gösterilmektedir. Sıcaklık 32°C'yi aştığında, yeşil LED yanmakta ve buzzer belirli bir melodi çalmaktadır.

## 2. Kullanılan Malzemeler

- NodeMCU ESP8266
- 16x2 LCD Ekran (I2C ile bağlantılı)
- DHT11 Sıcaklık ve Nem Sensörü
- Yeşil LED
- Jumper Kablolar
- Mikro USB Kablo
- 100 Ohm Direnç
- Buzzer

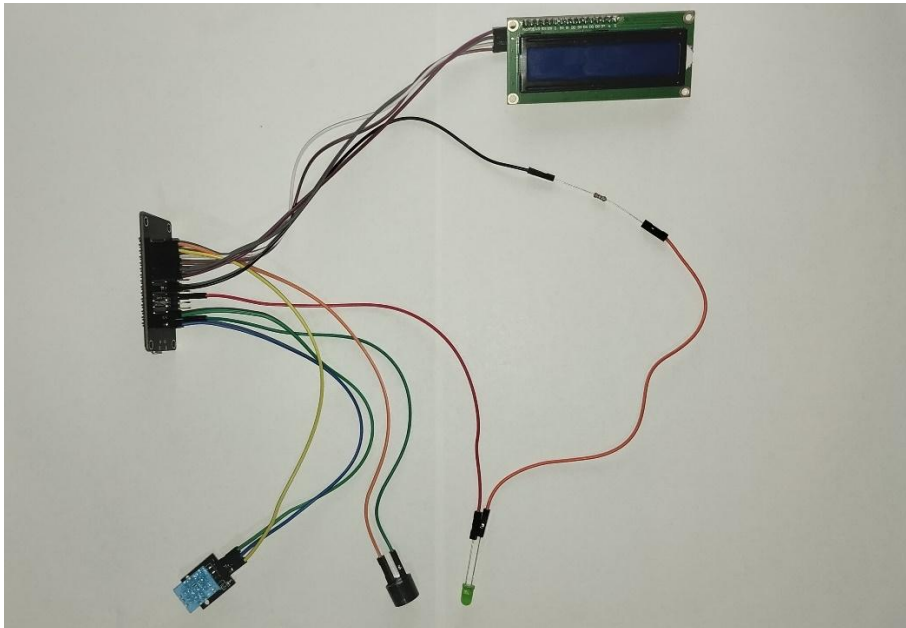
## 3. Kullanılan Uygulamalar

- Arduino IDE
- Blynk
- Proteus

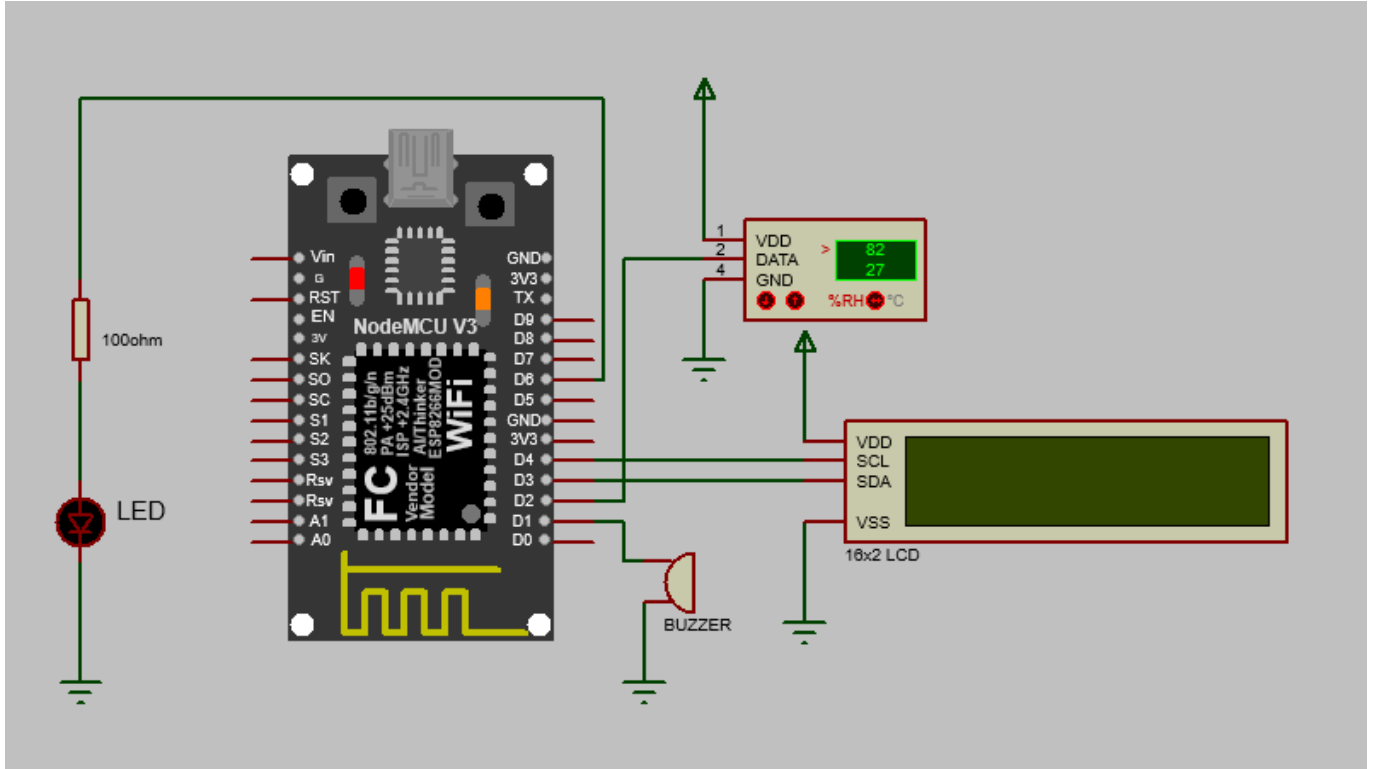
## 4. Proje Kapsamı

Bu proje, sıcaklık ve nem değerlerinin ölçülmesi ve bu değerlerin LCD ekran ve Blynk uygulaması aracılığıyla görselleştirilmesini kapsamaktadır. Ayrıca, sıcaklık belirli bir seviyeyi aştığında kullanıcıyı uyarmak amacıyla LED ve buzzer kullanılmaktadır.

## 5. Devre Şeması



## 6. Simülasyon Görüntüsü



## 7. Proje Yöntemi

Projede NodeMCU ESP8266 mikrodnetleyicisi kullanılarak DHT11 sensöründen sıcaklık ve nem verileri alınmıştır. Bu veriler, 16x2 LCD ekran ve Blynk uygulaması üzerinden kullanıcıya gösterilmiştir. Arduino IDE kullanılarak yazılan kod ile sıcaklık 32°C'yi aştığında yeşil LED'in yanması ve buzzerin belirli bir melodi çalması sağlanmıştır.

### 7.1. Kodlar ve Açıklamaları

#### 7.1.1. Gerekli Kütüphanelerin Dahil Edilmesi

```
#define BLYNK_TEMPLATE_ID "TMPL6K8JefXnv"
#define BLYNK_TEMPLATE_NAME "proje"

#include <DHT.h>
#include <Wire.h> // I2C kütüphanesi
#include <LiquidCrystal_I2C.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

Bu kütüphaneler, Wi-Fi bağlantısı, Blynk platformu, DHT sensörü ve LCD ekran için gereklidir.

### 7.1.2. Değişkenlerin Tanımlanması ve Bağlantı Bilgileri

```
#define DHTPIN 4 // D2 pinine bağladığınızı varsayıyoruz (GPIO4)
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

// LCD ekran için gerekli parametreler
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C adresi 0x27 olan 16x2 LCD ekran

// Blynk için gerekli değişkenler
char auth[] = "7mpv1z5AZ5a1iRVuub1dZFLO9enyx1fc"; // Blynk Auth Token
char ssid[] = "YALÇIN"; // WiFi SSID
char pass[] = "UOyalcin.19"; // WiFi Password

// Buzzer ve LED pinlerini tanımlayın
#define BUZZER_PIN 5 // D1 pinine bağladığınızı varsayıyoruz (GPIO5)
#define LED_PIN 12 // D6 pinine bağladığınızı varsayıyoruz (GPIO12)

// Super Mario melodisi için notalar ve süreleri
int melody[] = {
  262, 262, 0, 262, 0, 209, 262, 0, 330, 0, 392, 0, 0, 0,
  523, 0, 392, 0, 330, 0, 262, 0, 196, 0, 0, 0,
  392, 0, 370, 0, 349, 0, 311, 0, 330, 0, 0, 0,
  262, 0, 0, 0, 196, 0, 0, 0, 262, 0, 0, 0, 311, 0, 349, 0,
  330, 0, 262, 0, 330, 0, 392, 0, 523, 0, 0, 0
};
int noteDurations[] = {
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4
};
```

Burada, DHT sensörünün ve LCD ekranın bağlantı pinleri belirtilmiş ve Blynk yetkilendirme tokeni, Wi-Fi SSID ve şifresi tanımlanmıştır. Ayrıca, Super Mario melodisi için notalar ve süreler tanımlanmıştır.

### 7.1.3. Ayarların Yapılması ve Başlatılması

```
void setup() {
  Serial.begin(9600); // Seri iletişimi başlat ve baud hızını 9600 olarak ayarla

  // I2C pinlerini ayarla
  Wire.begin(0, 2); // SDA -> D3 (GPIO0), SCL -> D4 (GPIO2)

  dht.begin(); // DHT sensörünü başlat

  // LCD ekranı başlat
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("DHT11 sensor test");

  // Buzzer ve LED pinlerini çıkış olarak ayarla
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW); // Başlangıçta buzzer kapalı

  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW); // Başlangıçta LED kapalı

  // Blynk'i başlat
  Blynk.begin(auth, ssid, pass);
}
```

Bu kısımda, seri haberleşme, I2C pinleri, DHT sensörü, LCD ekran, Buzzer ve LED başlatılmaktadır. Blynk bağlantısı da burada sağlanmaktadır.

#### 7.1.4. Melodi Çalma Fonksiyonu

```
void playMelody() {
    // Super Mario melodisini çal
    int melodyLength = sizeof(melody) / sizeof(melody[0]);
    for (int thisNote = 0; thisNote < melodyLength; thisNote++) {
        int noteDuration = 1000 / noteDurations[thisNote];
        tone(BUZZER_PIN, melody[thisNote], noteDuration);
        int pauseBetweenNotes = noteDuration * 1.30;
        delay(pauseBetweenNotes);
        noTone(BUZZER_PIN);
    }
}
```

Bu fonksiyon, Super Mario melodisini buzzer aracılığıyla çalmaktadır.

#### 7.1.5. Döngü Fonksiyonu

```
void loop() {
    Blynk.run(); // Blynk işlevini çalıştır
    delay(2000); // Okumalar arasında 2 saniye bekle

    // Nem ve sıcaklık değerlerini oku
    float humi = dht.readHumidity();
    float temperature_C = dht.readTemperature();

    // Okumanın başarılı olup olmadığını kontrol et
    if (isnan(temperature_C) || isnan(humi)) {
        Serial.println("Failed to read from DHT sensor!");
        lcd.setCursor(0, 1);
        lcd.print("Failed to read");
    } else {
        Serial.print("Sıcaklık: ");
        Serial.print(temperature_C);
        Serial.print(" °C, Nem: ");
        Serial.print(humi);
        Serial.println(" %");
    }
}
```

```
// LCD ekranına sıcaklık ve nem değerlerini yazdır
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Sıcaklık:");
lcd.print(temperature_C);
lcd.print(" °C");

lcd.setCursor(0, 1);
lcd.print("Nem: ");
lcd.print(humi);
lcd.print(" %");

// Blynk uygulamasına sıcaklık ve nem değerlerini gönder
Blynk.virtualWrite(V0, temperature_C); // Sıcaklık için Virtual Pin V0
Blynk.virtualWrite(V1, humi);           // Nem için Virtual Pin V1
Blynk.virtualWrite(V2, LED_PIN);

// Sıcaklık 32 dereceyi geçtiğinde buzzer'ı çalıştır ve LED'i yak
if (temperature_C > 32) {
    digitalWrite(LED_PIN, HIGH); // LED'i yak
    playMelody(); // Tehlike alarmı melodisini çal
} else {
    noTone(BUZZER_PIN); // Sesi kapat
    digitalWrite(LED_PIN, LOW); // LED'i kapat
}
}
```

Bu döngü fonksiyonunda, sıcaklık ve nem değerleri okunmakta ve LCD ekranda gösterilmektedir. Ayrıca, bu değerler Blynk platformuna gönderilmektedir. Sıcaklık 32°C'nin üzerinde ise LED yanmakta ve buzzer belirli bir frekansta ses çıkarmaktadır.

## 8. Sonuçlar

Bu projede, DHT11 sensörü ile ortam sıcaklığı ve nemi başarıyla ölçülmüş ve bu veriler LCD ekran ve Blynk uygulaması üzerinden kullanıcıya sunulmuştur. Sıcaklık 32°C'yi aştığında, kullanıcıyı uyarmak için LED yanmakta ve buzzer belirli bir melodi çalmaktadır.