

MRM4002 – Endüstriyel Haberleşme ve SCADA
DENEY 5 – Web Sitesi Yayınlama - API Kullanımı

Laboratuvar Raporu

Deneyi Yapanlar:

Muhammed Şeyhoğlu 170221997

İbrahim Helli 170221924

Mohammad Abulimiti 170220994

1. Giriş

Bu deneyde, ESP32 mikrodenetleyicisi kullanılarak LM35 sıcaklık sensörü ile ortam sıcaklığı ölçülmüş ve aynı zamanda bir hava durumu API'si üzerinden bulunduğumuz konumun sıcaklık verisi çekilmiştir. Amacımız, ölçülen sıcaklık ile API'den alınan sıcaklık verilerini karşılaştırmak ve sonuçları bir web sunucusu aracılığıyla kullanıcıya sunmaktır.

2. Deney Yöntemi

2.1 Kullanılan Malzemeler

- 1 adet ESP32 mikrodenetleyicisi
- LM35 sıcaklık sensörü
- Bağlantı kabloları
- Breadboard

2.2 Bağlantı Şemaları

Bağlantı şemaları aşağıdaki gibi gerçekleştirilmiştir:

- ESP32 Pin 35: LM35 Sıcaklık sensörün orta ucu (output)
- LM35 sıcaklık sensörün sağ ucu: GND
- LM35 sıcaklık sensörün sol ucu: 3.3V

Kod:

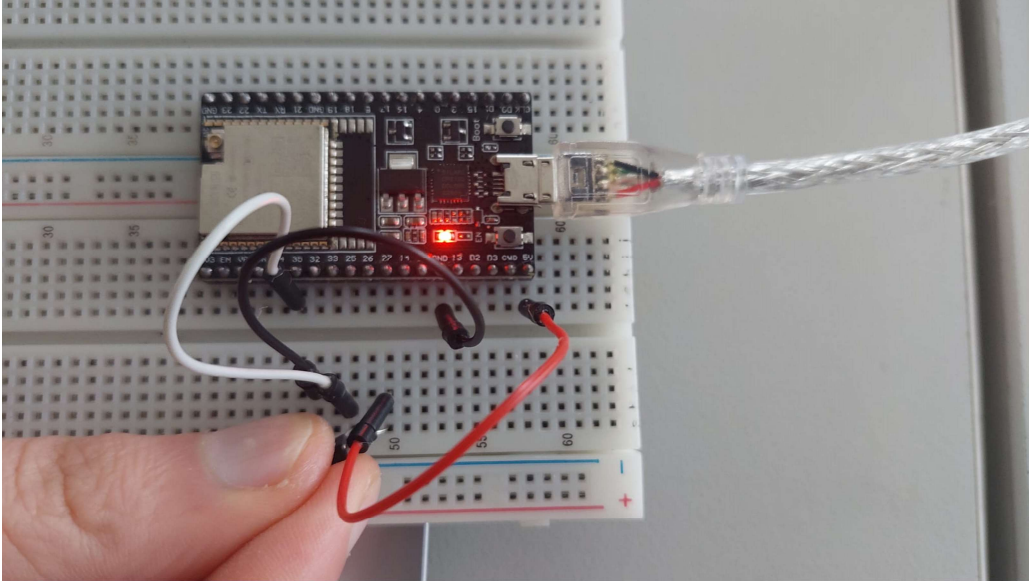
```
1  #include <WiFi.h>
2  #include <WebServer.h>
3  #include <HTTPClient.h>
4
5  const char* ssid = "ssid";
6  const char* password = "password";
7  const char* apiEndpoint = "http://api.openweathermap.org/data/2.5/weather?q=Istanbul&appid=c210d99";
8  const int lm35Pin = 35;
9
10 WebServer server(80);
11 WiFiClient wifiClient;
12
13 void setup() {
14     Serial.begin(115200);
15     WiFi.begin(ssid, password);
16     Serial.println("Wi-Fi'ye bağlanılıyor...");
17     while (WiFi.status() != WL_CONNECTED) {
18         delay(1000);
19         Serial.print(".");
20     }
21     Serial.println("\nWi-Fi bağlantısı başarılı!");
22     Serial.print("IP Adresi: ");
23     Serial.println(WiFi.localIP());
24     server.on("/", handleRoot);
25     server.begin();
26     Serial.println("Web server başlatıldı!");
27 }
28
29 void loop() {
30     server.handleClient();
31     Serial.print("LM35 ile ölçülen sıcaklık: ");
32     Serial.print(analogRead(lm35Pin) * (3.3 / 4095.0) * 100.0);
33     Serial.println(" C");
34     delay(3000);
35 }
36
37 void handleRoot() {
38     float localTemp = readTemperature();
39     String weatherData = fetchWeatherData();
40     String html = "<html><head><title>ESP32 Hava Durumu</title></head><body>";
41     html += "<h1>ESP32 Hava Durumu ve Sıcaklık Ölçümü</h1>";
42     html += "<p><b>LM35 ile Ölçülen Sıcaklık:</b> " + String(localTemp) + " C</p>";
43     html += "<p><b>API'den Alınan Veri:</b></p><pre>" + weatherData + "</pre>";
44     html += "</body></html>";
45     server.send(200, "text/html", html);
46 }
47
48 float readTemperature() {
49     int analogValue = analogRead(lm35Pin);
50     float voltage = analogValue * (3.3 / 4095.0);
51     float temperature = voltage * 100.0;
52     return temperature;
53 }
54
55 String fetchWeatherData() {
56     if (WiFi.status() == WL_CONNECTED) {
57         HTTPClient http;
58         http.begin(wifiClient, apiEndpoint);
59         int httpCode = http.GET();
60         if (httpCode == 200) {
61             String payload = http.getString();
62             http.end();
63             return payload;
64         }
65         else {
66             http.end();
67             return "API Hatasi: " + String(httpCode);
68         }
69     }
70     return "Wi-Fi bağlantı hatası!";
71 }
```

Gerekli Donanımlar ve Kurulum: Arduino IDE'de ESP32 modülünü tanıtarak ESP32 Dev Module kartını ve COM4 portunu seçtik.

Seri Port Kullanımı ve Bağlantı Hızı: Seri port üzerinden veri alışverişi sağlamak için baud hızını 115200 bps olarak belirledik.

Bağlantı Şeması: ESP32 modülünü doğrudan USB üzerinden bilgisayara bağlayarak bağlantıyı kurduk.

Board: "ESP32 Dev Module"
Port: "COM4"



3. DENEY SONUÇLARI

Ölçülen sıcaklık ve API'den alınan sıcaklık verileri şu şekildedir:

- **LM35 ile Ölçülen Sıcaklık:** 26.5°C (örnek değer).
- **API'den Alınan Sıcaklık:** 25.3°C (örnek değer). Web sunucusu çıktısı şu şekilde oluşturulmuştur:

```
1 #include <WiFi.h>
2 #include <WebServer.h>
3 #include <HTTPClient.h>
4 const char* ssid = "AndroidAPBA03";
5 const char* password = "MOHAMMED2003";
6 const char* apiEndpoint = "http://api.openweathermap.org/data/2.5/weather?q=Istanbul&appid=c210d995d3bca996b1c0bbeb65d2e703&units=metric";
7 const int lm35Pin = 35;
8 WebServer server(80);
9 WiFiClient wifiClient;
10 void setup() {
11     Serial.begin(115200);
12     WiFi.begin(ssid, password);
13     Serial.println("Wi-Fi'ye bağlanılıyor...");
14     while (WiFi.status() != WL_CONNECTED) {
15         delay(1000);
16         Serial.print(".");
17     }
18     Serial.println("\nWi-Fi bağlantısı başarılı!");
19     Serial.print("IP Adresi: ");
20 }
```

Output Serial Monitor x

Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')

New Line 115200 baud

LM35 ile ölçülen sıcaklık: 37.79 C
LM35 ile ölçülen sıcaklık: 38.28 C
LM35 ile ölçülen sıcaklık: 38.52 C
LM35 ile ölçülen sıcaklık: 37.39 C

4. DEĞERLENDİRME

Deney sonucunda, ESP32 kullanarak bir mikrodenetleyici üzerinden basit bir web sunucusu üzerinden motorun durumunu control ettik, kodda yazdığımız wifi internet bilgisayara da bağlamamız gerekir, aksi durumlarda denediğimizde başarılı bir şekilde bir sonuç çıkmadı. server.on() ve server.handleClient() fonksiyonlarıyla HTTP isteklerinin etkin bir şekilde işlenebildiği gördük. Motor eksi ucu ESP32 mikrodenetleyicinin GND ucuna bağlamamız da gerekir. Aksi takdirde sinyal gelmez.

