

**MRM4002 – End striyel Haberle me ve SCADA****DENEY 4 – ESP32 ile Motor veya Ara  Kontrol *****Laboratuvar Raporu***

Deneyi Yapanlar:

**Muhammed  eyho lu 170221997**

** brahim Helli 170221924**

**Mohammad Abulimiti 170220994**

**1. Giri **

Bu projede, bir ESP32 mikrodene leyicisi kullanılarak BJT transist r aracılı ıyla k   k bir DC motorun kontrol  ger ekle tirilmi tir. Projenin amacı, web tabanlı bir kontrol aray z  olu turarak motorun a ılıp kapatılabilmesini sa lamaktır. ESP32, bir web sunucusu olarak kullanılmı  ve kullanıcılar a   zerinden motorun durumunu kontrol edebilmi tir.

**2. Deney Y ntemi****2.1 Kullanılan Malzemeler**

- 1 adet ESP32 mikrodene leyicisi
- 1 adet BJT transist r (2N2222)
- 1 adet DC motor
- 1 adet 2.2 k  diren 
- 1 adet 9V pil ve pil ba lantı aparatları
- Ba lantı kabloları
- Breadboard

**2.2 Ba lantı  emaları**

Ba lantı  emaları a a ıdaki gibi ger ekle tirilmi tir:

- ESP32 Pin 16: Transist r n base ucuna 2.2 k  diren   zerinden ba landı.
- Transist r n Emitter Ucu: Topra a (GND).
- Transist r n Collector Ucu: DC motorun bir ucuna ba landı.
- DC Motorun Di er Ucu: 9V pilin pozitif ucuna ba landı.
- 9V Pilin Negatif Ucu: ESP32'nin GND pinine ba landı.

## Bağlantı Sağlama:

### Kod:

```
Lab4.ino
1  #include <WebServer.h>
2  #include <WiFi.h>
3  const char* ssid = "ssid";
4  const char* password = "password";
5  WebServer server(80);
6  const int motorPin = 16;
7  bool motorState = false;
8
9  void setup() {
10     Serial.begin(115200);
11     pinMode(motorPin, OUTPUT);
12     digitalWrite(motorPin, LOW);
13     WiFi.begin(ssid, password);
14     while (WiFi.status() != WL_CONNECTED) {
15         server.on("/", handleRoot);
16         server.on("/motor_on", handleMotorOn);
17         server.on("/motor_off", handleMotorOff);
18         server.begin();
19     }
20     void loop() {
21         server.handleClient(); }
22     void handleRoot() {
23         String html = "<html>\n
24         <head>\n
25         <title>Motor Kontrol</title>\n
26         </head>\n
27         <body>\n
28         <h1>ESP32 Motor Kontrol</h1>\n
29         <p>Motor durumu: " + String(motorState ? "ON" : "OFF") + "</p>\n
30         <button onclick=\"location.href='/motor_on'\">Motor ON</button>\n
31         <button onclick=\"location.href='/motor_off'\">Motor OFF</button>\n
32         </body>\n
33         </html>";
34         server.send(200, "text/html", html); }
35     void handleMotorOn() {
36         motorState = true;
37         analogWrite(motorPin, 255);
38         handleRoot(); }
39
40     void handleMotorOff() {
41         motorState = false;
42         analogWrite(motorPin, 0);
43         handleRoot(); }
```

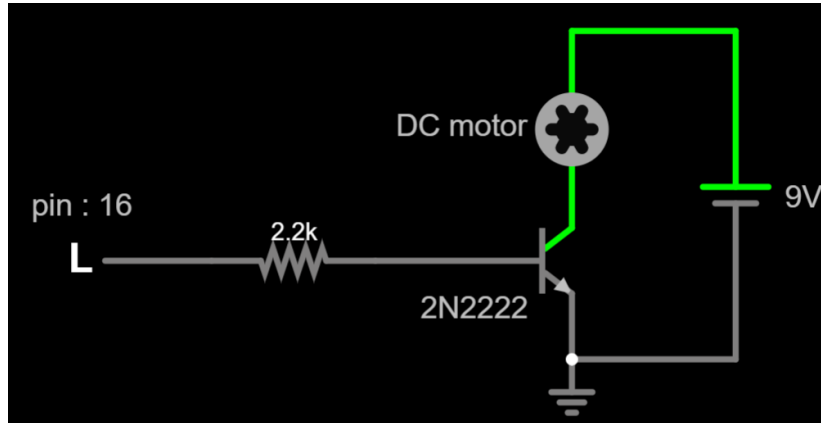
**Gerekli Donanımlar ve Kurulum:** Arduino IDE'de ESP32 modülünü tanıtarak ESP32 Dev Module kartını ve COM4 portunu seçtik.

Board: "ESP32 Dev Module"

Port: "COM4"

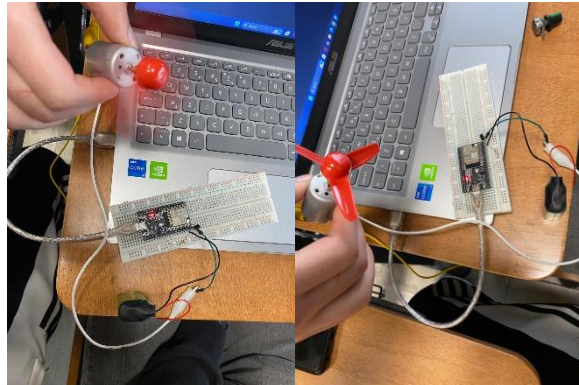
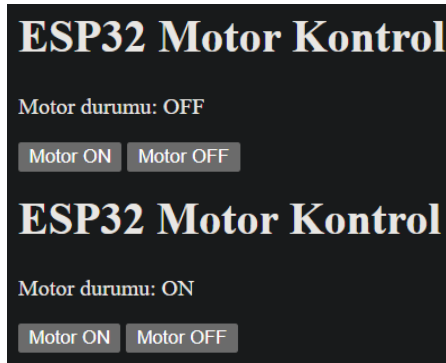
**Seri Port Kullanımı ve Bağlantı Hızı:** Seri port üzerinden veri alışverişi sağlamak için baud hızını 115200 bps olarak belirledik.

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



### 3. DENEY SONUÇLARI

ESP32 kullanarak bir motorun Wi-Fi üzerinden kontrol ettik. Motor kontrolü için motorPin (16. pin) çıkış pini olarak ayarladık ve başlangıçta motor kapalı olacak şekilde LOW durumuna getirdik. ESP32 üzerinde bir web sunucusu çalıştırdık ve bu sunucu üç farklı URL'ye yanıt verir: / adresi ana sayfa olup motorun durumu (ON veya OFF) görüntülenir ve motor kontrol düğmeleri bulunur; /motor\_on adresi motoru açar; /motor\_off adresi ise motoru kapatır. HTML sayfasında motor durumu dinamik olarak görüntülenir ve kullanıcı düğmelere tıklayarak motoru kontrol edebilir.



### 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.