



T.C.
KIRŞEHİR AHI EVRAN ÜNİVERSİTESİ
MHENDİSLİK MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİLİĞİ
LİSANS

Bluetooth Kontrollü
6 Eksenli Robot Kol

202511040
Kadir Bedirhan US

LİSANS PROJESİ

KIRŞEHİR
2023

İÇİNDEKİLER DİZİNİ	II
1. ÖZET.....	III
2. GİRİŞ	III
3. KULLANILAN BİLEŞENLER.....	IV
4. PROJE MİMARİSİ	V
5. MEKANİK YAPI.....	VI
6. YAZILIM VE KODLAMA.....	VII
7. MOBİL UYGULAMA.....	8
8. SONUÇ VE DEĞERLENDİRME.....	9
9. KAYNAKLAR	10
10. EKLER.....	11
EK-1	
Proje Fotoğrafları.....	11
EK-2	
Proje Kodları.....	13

1.ÖZET

Proje kapsamında, 6 eksenli bir robot kol geliştirilmiştir. Robot kol, farklı görevleri yerine getirebilmek için SG90 ve MG995 tipi toplam 6 adet servo motor kullanmaktadır. Bu motorlar, Arduino Uno ve Sensor Shield kullanılarak kontrol edilmektedir. Servo motorlara güç sağlamak için 5V bir güç kaynağı kullanılmaktadır. Robotun kontrolü ise bir mobil uygulama ve HC-05 Bluetooth modülü aracılığıyla gerçekleştirilmektedir. Mobil uygulama, pozisyon kaydetme ve bu pozisyonları döngü şeklinde oynatma gibi özellikler sunmaktadır.

Anahtar Kelimeler: Robot Kol, Servo Motor, Arduino, Bluetooth

2. GİRİŞ

Teknolojinin hızla ilerlediği günümüzde, robotik sistemler endüstri ve günlük yaşamda önemli bir yer edinmiştir. Robot kollar, bu sistemlerin en temel bileşenlerinden biridir ve üretim hatlarından sağlık sektörüne kadar geniş bir kullanım alanına sahiptir. Bu çalışma, 6 eksenli bir robot kol tasarımı ve uygulanmasını ele almaktadır.

Projenin temel amacı, servo motorlar, Arduino mikrodenetleyici ve mobil uygulama entegrasyonu kullanılarak fonksiyonel ve kullanıcı dostu bir robot kol geliştirmektir. Çalışma, robot kolların hareket kontrolü, hassasiyet ve esneklik gibi temel kavramlarını anlamaya yöneliktir. Ayrıca, Bluetooth bağlantısı üzerinden mobil cihazlarla iletişim kuran bir sistemin nasıl geliştirileceği detaylı şekilde açıklanmıştır.

Bu rapor, projenin teknik detaylarını, kullanılan bileşenleri, yazılım altyapısını ve mekanik yapıyı kapsamlı bir şekilde sunmaktadır. Çalışmanın, robotik sistemler üzerine yapılan ileri düzey projelere başlangıç niteliğinde bir rehber olması hedeflenmiştir.

3. KULLANILAN BİLEŞENLER

3.1 Servo Motorlar

- **SG90 Servo Motorlar:** Daha küçük ve hafif yükler için kullanılan bu motorlar, robot kolun hassas hareketlerini sağlamaktadır.
- **MG995 Servo Motorlar:** Daha yüksek tork gerektiren hareketlerde kullanılan güçlü motorlardır. Robot kolun daha ağır yükleri taşımasını mümkün kılar.

3.2 Arduino ve Sensor Shield

- **Arduino Uno:** Robot kolun kontrol işlemlerini yöneten mikrodenetleyicidir. Servo motorların açılarının kontrol edilmesi ve Bluetooth modülünden gelen verilerin işlenmesi gibi görevleri üstlenir.
- **Sensor Shield:** Arduino'ya servo motorların ve diğer bileşenlerin kolay bir şekilde bağlanmasını sağlar.

3.3 HC-05 Bluetooth Modülü

- Bu modül, Arduino ile mobil uygulama arasında kablosuz iletişim sağlar. Kullanıcı, uygulama üzerinden gönderdiği komutlarla robot kolun pozisyonlarını ayarlayabilir ve kayıt edebilir

4. PROJE MİMARİSİ

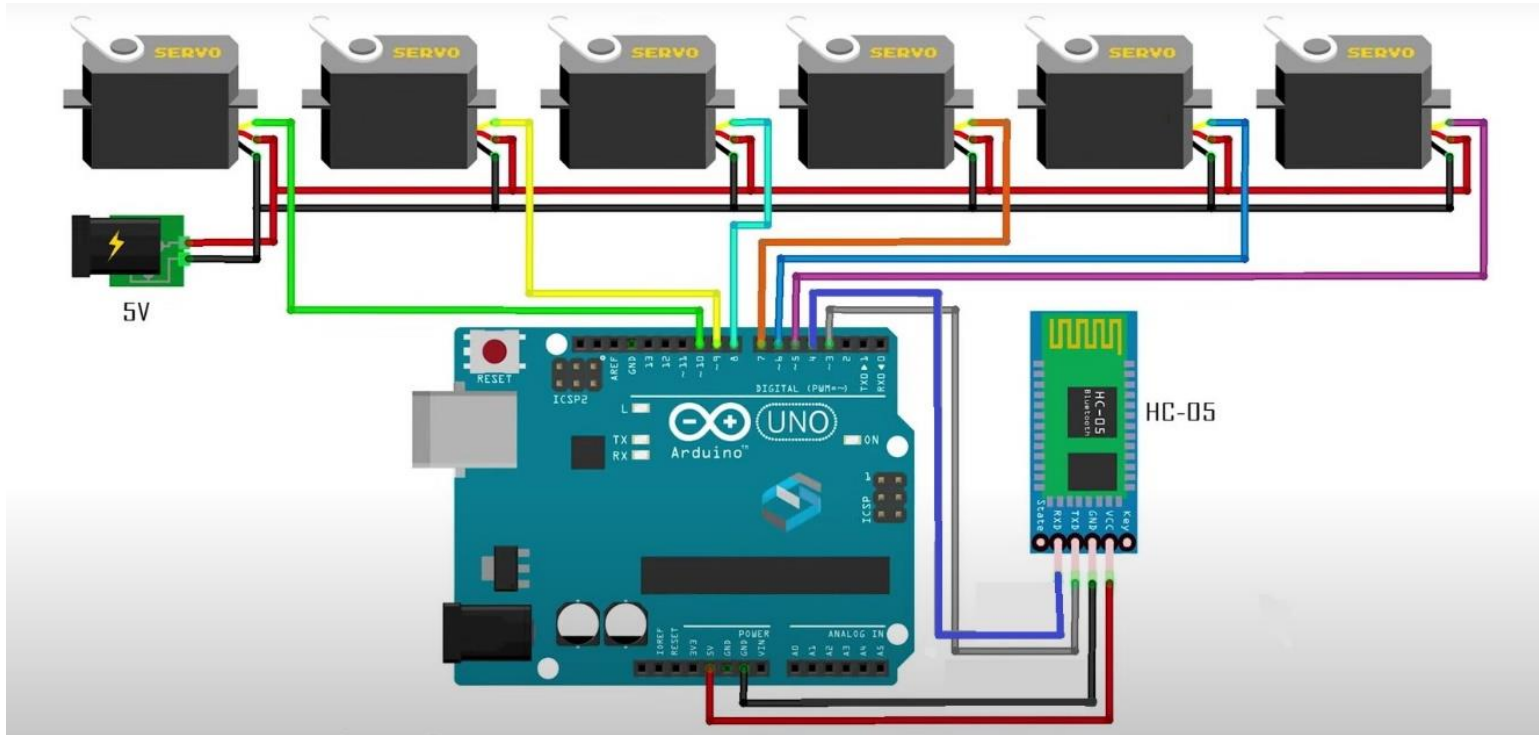
4.1 Bileşenler

Proje, aşağıdaki bileşenlerden oluşan bir mimariye sahiptir:

1. **Servo Motorlar:** Her bir eksenin hareketini sağlar.
2. **Arduino Uno ve Sensor Shield:** Kontrol birimi ve bağlantı noktası olarak görev yapar.
3. **Bluetooth Modülü:** Kablosuz iletişim sağlar.
4. **Mobil Uygulama:** Kullanıcı arayüzü sunarak robot kolun hareketlerini kontrol etme imkanı verir.

4.2 Elektriksel Bağlantılar

- Servo motorlar 5V güç kaynağı ile beslenmektedir.
- Arduino Uno üzerinden Sensor Shield aracılığıyla servo motorlara sinyal iletilmektedir.
- HC-05 Bluetooth modülü, Arduino'nun RX ve TX pinlerine bağlanmıştır.



5.MEKANİK YAPI

Robot kolun mekanik yapısı, farklı görevleri yerine getirmek üzere optimize edilmiştir. Yapı şu şekilde detaylandırılabilir:

- **Taban:** Robot kolun sabit bir yüzeye monte edilmesini sağlar ve yatay ekseninde dönme hareketini gerçekleştirir. Bu bölüm, kolun diğer bölümleri için destek ve stabilite sağlar.
- **Omuz Bölgesi:** Robot kolun üst kısmı ile taban arasındaki bağlantıyı sağlar. Bu bölüm yukarı-aşağı hareketi gerçekleştirir ve geniş açılı hareketleri desteklemek için güçlü bir motor ile donatılmıştır.
- **Dirsek:** Omuz ile bilek arasında yer alır ve ileri-geri hareketlerin yanı sıra yukarı-aşağı hareketlerin yapılmasını sağlar. Bu bölüm, kolun esnekliğini artırır ve karmaşık hareketlerin gerçekleştirilmesine olanak tanır.
- **Bilek:** İnce hareketlerin yapılması için optimize edilmiştir. Çeşitli açıları destekleyerek robot kolun küçük nesneleri kavramasını veya manipüle etmesini sağlar.
- **Tutuş Mekanizması (Kıskaç):** Nesneleri kavrama ve taşıma işlevi görür. Bu mekanizma, robot kolun kullanım amacına göre değişiklik gösterebilir.
- **Malzeme ve Tasarım:** Mekanik yapı, hafif ancak dayanıklı malzemelerden (örneğin, alüminyum veya ABS plastik) üretilmiştir. Bu, kolun yük kapasitesini artırırken aynı zamanda hareket hassasiyetini korur.

Her bir bölüm, bir servo motor ile hareket ettirilir. Bu motorlar, yüksek hassasiyet ve kontrol sağlamak amacıyla farklı tork ve hız seviyelerine göre seçilmiştir. Bağlantılar, dayanıklılık ve kolay montaj için vidalı düzeneklerle desteklenmiştir.

6. YAZILIM VE KODLAMA

Robot kolun hareketlerini kontrol etmek için Arduino programlama dili kullanılmıştır. Kodda, servo motorların belirli hızlarla hareket ettirilmesi ve pozisyonlarının kaydedilmesi sağlanmıştır.

6.1 Temel Özellikler

1. **Servo Kontrolü:**
 - Her bir servo motorun mevcut ve önceki pozisyonları ayrı ayrı izlenmektedir.
 - Servo motorların hareketi, hız parametreleri ile yavaşlatılmıştır.
2. **Pozisyon Kaydetme:**
 - Bluetooth üzerinden gelen komutlar doğrultusunda, mevcut pozisyonlar kaydedilmektedir.
 - Kaydedilen pozisyonlar bir dizi içinde saklanır.
3. **Döngüsel Çalışma:**
 - Kaydedilen pozisyonlar, RUN komutu ile sırasıyla oynatılmaktadır.
 - Döngüsel olarak tekrar eden bir çalışma prensibi benimsenmiştir.
4. **Resetleme:**
 - RESET komutu ile tüm pozisyonlar sıfırlanabilmektedir.

7. MOBİL UYGULAMA

7.1 Özellikler

Mobil uygulama, kullanıcıların robot kolun pozisyonlarını kolayca kontrol edebilmesini sağlamaktadır. Uygulama, Bluetooth üzerinden Arduino'ya komutlar gönderir. Başlıca özellikleri şunlardır:

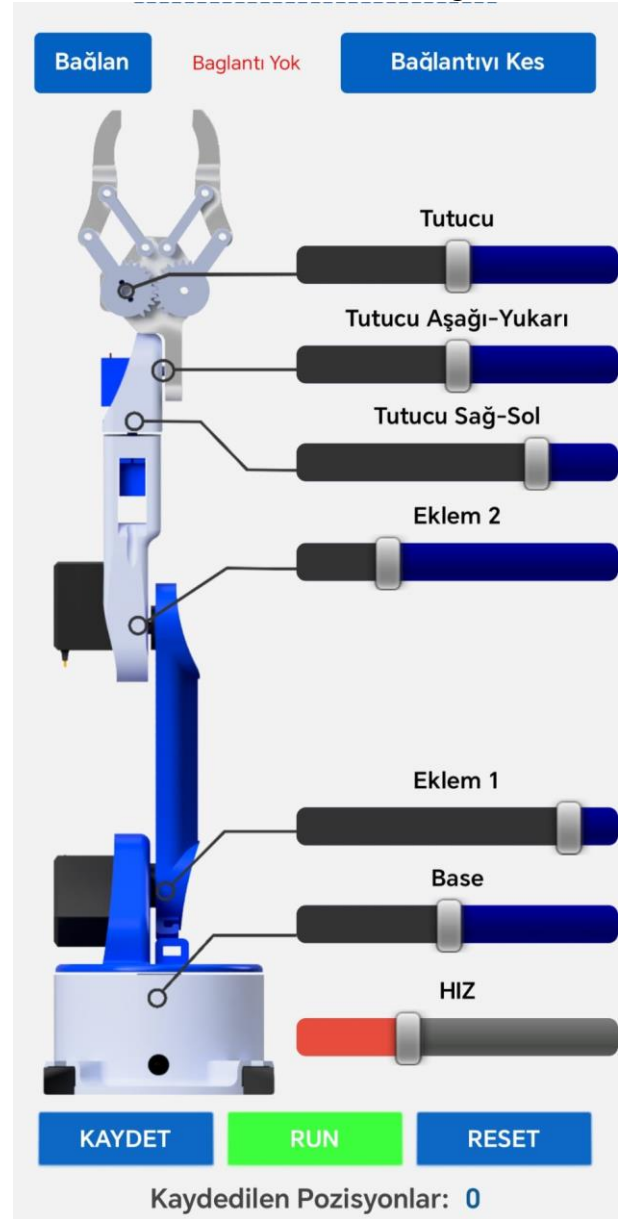
- **Manuel Kontrol:** Kullanıcı, her bir eksenin açısını manuel olarak ayarlayabilir.
- **Pozisyon Kaydetme:** Belirli bir pozisyonu kaydederek daha sonra tekrar kullanmak üzere saklar.
- **Döngü Oynatma:** Kaydedilen pozisyonları otomatik olarak tekrar eden bir döngü içinde oynatır.

7.2 Kullanıcı Arayüzü

Mobil uygulamanın arayüzü, kullanıcı dostu bir tasarıma sahiptir. Her bir eksen için ayrı bir kaydırma çubuğu veya buton bulunmaktadır. "SAVE", "RUN" ve "RESET" gibi temel işlemler için özel butonlar eklenmiştir.

7.3 Uygulama Senaryosu

1. Kullanıcı, mobil uygulamayı açar ve Bluetooth ile HC-05 modülüne bağlanır.
2. Servo motorları kontrol ederek robot kolu istenen pozisyona getirir.
3. Pozisyon "SAVE" butonuna basılarak kaydedilir.
4. "RUN" butonuna basılarak kaydedilen pozisyonlar otomatik olarak oynatılır.



8. SONUÇ VE DEĞERLENDİRME

Proje, temel olarak servo motorların kontrolü ve robotik sistemlerin çalışma prensiplerinin öğrenilmesi amacıyla gerçekleştirilmiştir. Kullanıcı dostu bir mobil uygulama ile entegre edilmesi, projeyi uygulama açısından daha işlevsel kılmıştır. Gelecekte, aşağıdaki geliştirmeler üzerinde çalışılabilir:

1. Robot kolun hassasiyetinin artırılması.
2. Daha karmaşık görevler için programlama yapılması.
3. Görüntü işleme veya sensör tabanlı bir kontrol sistemi ile entegrasyon.

Bu çalışma, robotik ve otomasyon alanında temel bir uygulama olarak değerlendirilebilir ve daha büyük projelerin başlangıç noktası olabilir.

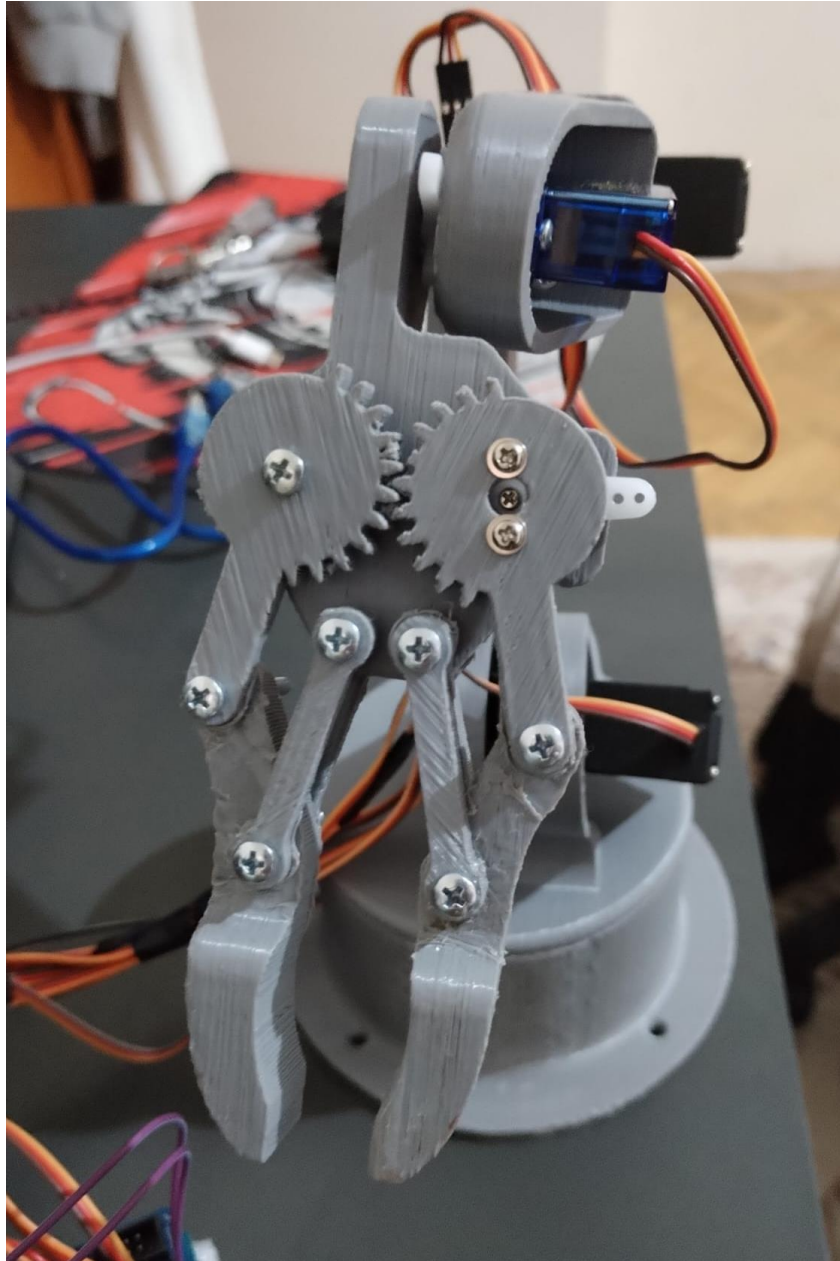
9. KAYNAKLAR

1. Arduino Documentation: <https://www.arduino.cc>
2. Servo Motor Kullanımı Kılavuzu.
3. HC-05 Bluetooth Modülü Teknik Dokümanı.

10.EKLER

EK-1 Proje Fotoğrafları





EK-2 Proje Kodları

```
void setup() {
  Serial.begin(115200);
  Bluetooth.begin(9600);
  Bluetooth.setTimeout(10);

  servo1.attach(2, 510, 1200);
  servo2.attach(3, 650, 1400);
  servo3.attach(4, 650, 1400);
  servo4.attach(5, 650, 1400);
  servo5.attach(6, 950, 1400);
  servo6.attach(7, 600, 1290);
  s1Ant = 90;
  servo1.write(s1Ant);
  s2Ant = 110;
  s3Ant = 70;
  servo2.write(s2Ant);
  servo3.write(s3Ant);
  s4Ant = 115;
  servo4.write(s4Ant);
  s5Ant = 70;
  servo5.write(s5Ant);
  s6Ant = 50;
  servo6.write(s6Ant);
  s7Ant = 110;
  servo7.write(s7Ant);
  delay(50);
}

void loop() {

  if (Bluetooth.available() > 0) {
    bt = Bluetooth.readString();

    if (bt.startsWith("s1")) {
      btS = bt.substring(2, bt.length());
      s1Act = btS.toInt();
      if (s1Ant > s1Act) {
        for (int j = s1Ant; j >= s1Act; j--) {
          servo1.write(j);
          delay(s1Vel);
        }
      } else {
        for (int j = s1Ant; j <= s1Act; j++) {
          servo1.write(j);
          delay(s1Vel);
        }
      }
    }
  }
}
```

```

s1Ant = s1Act;    }

if (bt.startsWith("s2")) {
    btS = bt.substring(2, bt.length());
    s2Act = btS.toInt();
    Serial.println(s2Act);
    if (s2Ant > s2Act) {
        for (int j = s2Ant; j >= s2Act; j--) {
            servo2.write(j);
            servo3.write(180 - j);
            delay(s2Vel);
        }
    } else {
        for (int j = s2Ant; j <= s2Act; j++) {
            servo2.write(j);
            servo3.write(180 - j);
            delay(s2Vel);
        }
    }
}

s2Ant = s2Act;    }

if (bt.startsWith("s4")) {
    btS = bt.substring(2, bt.length());
    s4Act = btS.toInt();

    if (s4Ant > s4Act) {
        for (int j = s4Ant; j >= s4Act; j--) {
            servo4.write(j);
            delay(s4Vel);
        }
    } else {
        for (int j = s4Ant; j <= s4Act; j++) {
            servo4.write(j);
            delay(s4Vel);
        }
    }
}

s4Ant = s4Act;
}

if (bt.startsWith("s5")) {
    btS = bt.substring(2, bt.length());
    s5Act = btS.toInt();

    if (s5Ant > s5Act) {
        for (int j = s5Ant; j >= s5Act; j--) {
            servo5.write(j);

```

```

        delay(s5Vel);
    }
} else {
    for (int j = s5Ant; j <= s5Act; j++) {
        servo5.write(j);
        delay(s5Vel);
    }
}

s5Ant = s5Act;
}

if (bt.startsWith("s6")) {
    btS = bt.substring(2, bt.length());
    s6Act = btS.toInt();
    Serial.println(s6Act);
    if (s6Ant > s6Act) {
        for (int j = s6Ant; j >= s6Act; j--) {
            servo6.write(j);
            delay(s6Vel);
        }
    } else {
        for (int j = s6Ant; j <= s6Act; j++) {
            servo6.write(j);
            delay(s6Vel);
        }
    }

    s6Ant = s6Act;
}

    if (bt.startsWith("SAVE")) {
s1[index] = s1Ant;
s2[index] = s2Ant;
s4[index] = s4Ant;
s5[index] = s5Ant;
s6[index] = s6Ant;
s7[index] = s7Ant;
Serial.print("Pozisyon Kaydedildi: ");
Serial.print("s1: "); Serial.print(s1[index]);
Serial.print(" s2: "); Serial.print(s2[index]);
Serial.print(" s4: "); Serial.print(s4[index]);
Serial.print(" s5: "); Serial.print(s5[index]);
Serial.print(" s6: "); Serial.print(s6[index]);
Serial.print(" s7: "); Serial.println(s7[index]);
index++;
}

    if (bt.startsWith("RESET")) {
        memset(s1, 0, sizeof(s1));

```

```

    memset(s2, 0, sizeof(s2));
    memset(s4, 0, sizeof(s4));
    memset(s5, 0, sizeof(s5));
    memset(s6, 0, sizeof(s6));
    memset(s7, 0, sizeof(s7));
    index = 0;
}

if (bt.startsWith("RUN")) {
while (true) {
    for (int i = 0; i < index; i++) {
        servo1.write(s1[i]);
        servo2.write(s2[i]);
        servo4.write(s4[i]);
        servo5.write(s5[i]);
        servo6.write(s6[i]);
        servo7.write(s7[i]);
        delay(2500);
    }
    Serial.println("RUN döngüsü tamamlandı. Tekrar ediyor...");

    if (Bluetooth.available() > 0) {
        String stopCommand = Bluetooth.readString();
        if (stopCommand.startsWith("PAUSE")) {
            Serial.println("RUN döngüsü durduruldu.");
            break;
        }
    }
}
}
}
}

```