

Kullanıcıya Özgü Özelleştirilebilir Jest Tanıma Sistemi

Ali Burak Saraç (21401932)

Sude Melis Pilaz (22401992)

Galatasaray Üniversitesi, Bilgisayar Mühendisliği Bölümü

INF340 - Mikroişlemciler Dönem Projesi

Mayıs 2025

İçindekiler

1	Proje Amacı	2
2	Çalışma Prensipleri	2
3	Kullanılan Malzemeler ve Teknolojiler	3
4	Projede Kullanılacak Yazılım ve Donanım Altyapısı	4
4.1	Donanım Altyapısı	4
4.2	Yazılım Altyapısı	4
5	Projenin Teknik Tasarımı	4
5.1	Donanım Yapısı	4
5.2	Yazılım Mimarisi	5
5.2.1	Genel Yazılım Akışı	5
5.2.2	Dosya Yapısı	5
6	Karşılaşılan Sorunlar ve Çözümler	7
7	Potansiyel Geliştirmeler	8

1. Proje Amacı

Bu projenin amacı, kullanıcıların kendi hareket verilerini toplayarak kişiselleştirilmiş bir jest tanıma modeli oluşturabileceği esnek bir sistem geliştirmektir. Proje sayesinde kullanıcılar belirledikleri jestleri tanımlayıp, bu hareketlere karşılık gelen etkileşimleri sistemin öğrenmesini sağlayabilmektedir. MATLAB tabanlı uygulama, sensör verilerini alır, analiz eder ve kullanıcı tanımlı sınıflar üzerinden sınıflandırma işlemini gerçekleştirir. Sistem, eğitilmiş modelleri kullanarak yeni jestleri gerçek zamanlı olarak tanıyabilir ve bu jestlere karşılık gelen çıktıları üretir.

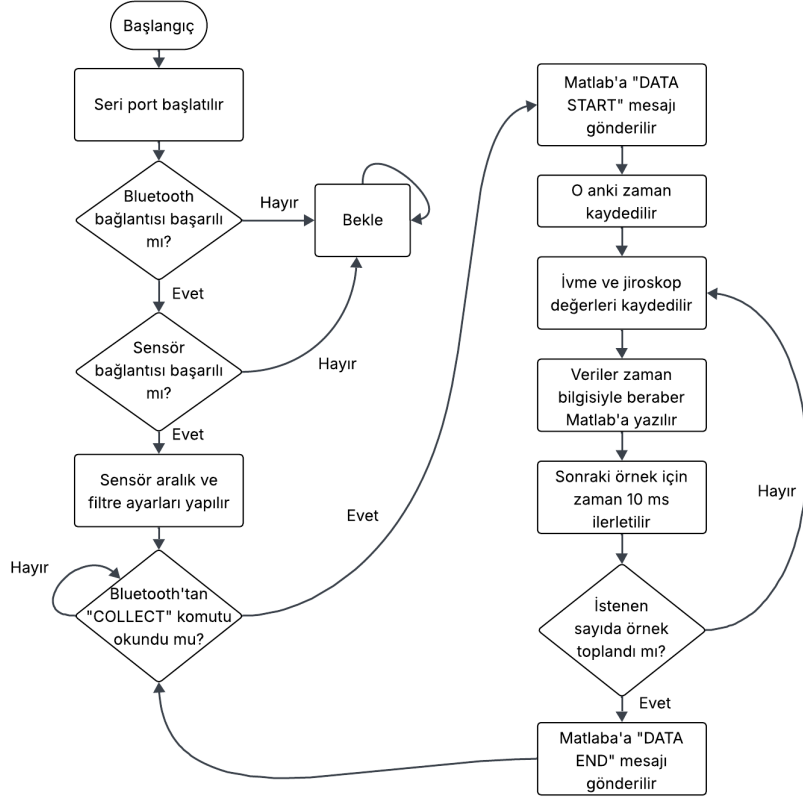
Bu sistem, görme engelliler gibi belirli gruplar için özelleştirilebildiği gibi, ev otomasyonu, oyun kontrolü, sanal gerçeklik etkileşimleri veya eğitimsel uygulamalarda da kişisel kullanım için uygun hale getirilebilir.

2. Çalışma Prensibi

Geliştirilen sistem, jest tanıma sürecini kullanıcıya özel ve esnek bir biçimde gerçekleştirecek şekilde tasarlanmıştır. Sistem, aşağıdaki donanım ve yazılım bileşenlerinden oluşur: bir adet ESP32 geliştirme kartı, MPU6050 ivmeölçer-jiroskop sensörü, breadboard ve bağlantı kabloları ile MATLAB yüklü bir bilgisayar, lityum pil ve şarj modülü.

Sistem şu şekilde çalışır:

- MPU6050 sensörü, kola sabitlenmiş şekilde konumlandırılır ve ESP32 üzerinden güç alır.
- Kullanıcı kolunu hareket ettirdikçe sensör, üç eksenli ivme ve açısal hız verilerini toplar.
- Toplanan veriler, ESP32'nin Bluetooth modülü aracılığıyla kablosuz olarak MATLAB yüklü bilgisayara aktarılır.
- MATLAB ortamında, bu veriler anlık olarak sınıflandırılabilir ya da yeni veri örnekleri toplanarak model eğitimi yapılabilir.
- Jest sınıflandırma sonuçları doğrudan MATLAB ekranında kullanıcıya yazılı olarak gösterilir.



Şekil 1: Sistem Akış Şeması

3. Kullanılan Malzemeler ve Teknolojiler

- **ESP32:** Sensör verilerini okumak, işlemek ve Bluetooth üzerinden bilgisayara iletmek için kullanılır.
- **MPU6050 Sensörü:** İvme ve açısal hız verisi sağlayan bu IMU sensörü, jest hareketlerinin algılanmasında kullanılır.
- **Breadboard ve Bağlantı Kabloları:** ESP32 ile MPU6050 arasında düzenli bağlantılar kurmak için kullanılır.
- **MATLAB Yüklü Bilgisayar:** Sensörden gelen verilerin alındığı, işlendiği ve sınıflandırma algoritmalarının çalıştırıldığı platformdur. Ayrıca kullanıcı bu ortamda model eğitimi de gerçekleştirebilir.
- **Bluetooth (ESP32 Dahili):** Kablosuz bağlantı sağlayarak sensör verilerinin bilgisayara iletilmesini sağlar. Harici modül gerektirmez.
- **Lityum Pil:** 3.7V 2000 mAh lityum polimer pil devrenin güç kaynağı olarak kullanılır.
- **Lityum Pil Şarj Devresi:** TP4056 micro usb şarj modülü pili şarj etmek için kullanılır.

4. Projede Kullanılacak Yazılım ve Donanım Altyapısı

Projede, hem donanım hem de yazılım altyapısı kullanıcıdan veri toplayabilen, bu veriyi analiz edebilen ve sınıflandırma yapabilen esnek bir sistem oluşturmaya yöneliktir.

4.1. Donanım Altyapısı

Kullanıcının koluna sabitlenen MPU6050 sensörü, üç eksenli ivme ve açısal hız verilerini toplar. Bu veriler, ESP32 geliştirme kartı tarafından I2C protokolü ile okunur ve Bluetooth üzerinden kablosuz olarak MATLAB yüklü bilgisayara aktarılır. ESP32, hem sensörle düşük seviyeli iletişimi hem de kablosuz haberleşmeyi üstlenerek sistemin merkezinde yer alır. Güç ihtiyacı, şarj edilebilir bir 3.7V lityum polimer pil ile karşılanır ve TP4056 şarj modülü ile güvenli şarj işlemi sağlanır. Tüm bileşenler breadboard üzerinde organize edilerek taşınabilir ve test edilebilir bir prototip oluşturulmuştur.

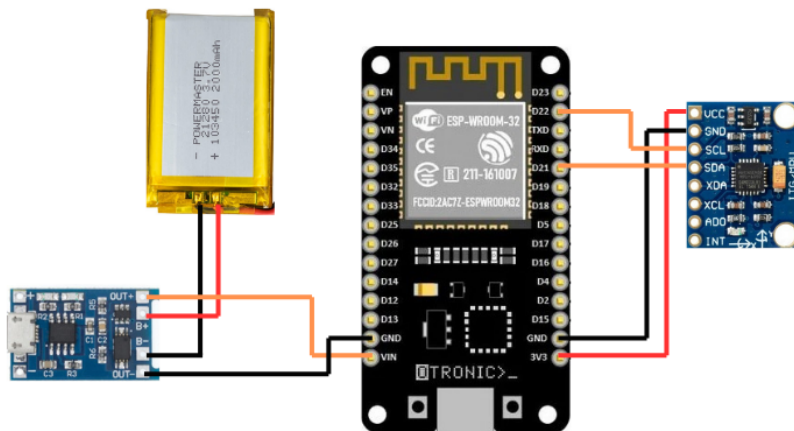
4.2. Yazılım Altyapısı

- **VS Code PlatformIO Eklentisi:** ESP32 üzerinde çalışan programın yazılması ve karta yüklenmesi için kullanılır.
- **MATLAB:** Bluetooth üzerinden alınan verilerin okunması, ön işlenmesi, özellik çıkarımı, sınıflandırma algoritmalarının çalıştırılması ve çıktıların gösterimi için kullanılır.

5. Projenin Teknik Tasarımı

5.1. Donanım Yapısı

Projede kullanılan donanım elemanları, bileğe takılabilecek boyutta olması gerektiği göz önüne alınarak tek bir breadboard üzerine yerleştirilmiştir. MPU6050 sensörü, kullanıcı koluna sabitlenecek şekilde konumlandırılmıştır. Bu sensörden çıkan bağlantılar, kısa kablolar aracılığıyla ESP32'ye bağlanmıştır (SCL: D22, SDA: D21). Gerekli güç bağlantıları (VCC ve GND) doğrudan ESP32 üzerinden sağlanmıştır.



Şekil 2: Devre Şeması

5.2. Yazılım Mimarisi

5.2.1 Genel Yazılım Akışı

`main.m` çalıştırılır ve `config.m` ile tüm parametreler (port, örnek sayısı, klasör yolları vb.) yüklenir. Kullanıcıdan mod seçimi (Veri Toplama, Sınıflandırma, Ham Akış, Özellik Çıkarımı, Grafik Oluşturma) alınır. `sensors.initializeSensors` ile ESP32 ile seri/Bluetooth bağlantısı kurulur; veri alışverişi için “DATA_START” / “DATA_END” işaretleri beklenir. Menüden istenen işlem seçilir:

1. Veri Toplama (Data Collection)

`runDataCollection` çağrıldığında, ESP32’ye “COLLECT” komutu gönderilir. `sensors.collectSamples` CSV satırlarını okuyarak `acc`, `gyro`, `ts` dizilerine dönüştürür. Veriler geçici dosyaya (`saveTemp`), ardından kullanıcı onayıyla `saveFinal` ve `savePlots` fonksiyonları ile kalıcı hale getirilir.

2. Gerçek Zamanlı Sınıflandırma (Classification)

Daha önce eğitilmiş model (`wideNeuralNetwork.mat`) yüklenir. Her pencere için veri toplanır, `classification.extractFeatures` ile özellik çıkarılır ve tahmin yapılır. Sonuç kullanıcıya ekranda gösterilir.

3. Ham Veri Akışı (Raw Streaming)

Sürekli olarak ESP32’den gelen CSV verisi okunur ve konsola basılır. Herhangi bir sınıflandırma veya grafik işlemi yapılmaz.

4. Özellik Seti Hazırlama (Extract Features)

`modelTraining.prepareGestureData` fonksiyonu ile:

- Tüm kayıtlar (`signalsStructFile.mat` & `metadata.csv`) yüklenir.
- Hedef jestlere göre filtreleme yapılır.
- Özellikler çıkarılıp tek bir tabloya (`featureTable`) dönüştürülür.
- MAT ve CSV formatında kaydedilir, `README.txt` oluşturulur.

5. Toplu Grafik Üretme (Plot Data)

`plotData.plot` fonksiyonu, tüm verileri tarar ve her kayıt için hız/ivme grafiklerini PNG olarak kaydeder.

5.2.2 Dosya Yapısı

Aşağıda, sistemde yer alan her bir MATLAB dosyasının işlevi ve sistem içerisindeki yeri açıklanmıştır.

`main.m`

Uygulamanın başlangıç noktasıdır. Kullanıcıya 5 mod sunar ve seçilen moda göre ilgili alt fonksiyonu çağırır ve donanım bağlantısını yönetir.

config.m

Tüm yapılandırma parametrelerini merkezi bir noktada toplar. Bu sayede sistemde sabit olarak kullanılan değerler tek yerden kontrol edilebilir:

- **Port ve Haberleşme Ayarları:** `cfg.port`, `cfg.baudRate`, `cfg.BluetoothName`, `cfg.BTChannel` — ESP32 ile seri/Bluetooth bağlantısı için gerekli tanımlamalar yapılır.
- **Örnekleme:** `cfg.SampleRate`, `cfg.TargetSamples`, `cfg.MedianFilterWindow` — Sensör verisi toplarken kullanılacak örnekleme frekansı, hedef örnek sayısı ve filtreleme parametreleri.
- **Dosya ve Klasör Yapısı:** `cfg.GestureFolder`, `cfg.MaxFileNameLength`, `cfg.learnerDir`, `cfg.LogFolder` — Verilerin kaydedileceği klasörler, dosya adı uzunluğu ve log klasörlerinin yolları tanımlanır.
- **Makine Öğrenmesi Parametreleri:**
 - `cfg.targetGestures` — Sınıflandırma işlemlerinde kullanılacak jest ID'leri.
 - `cfg.ClassificationMethod` — Varsayılan sınıflandırma algoritması (örn. SVM).
 - `cfg.CrossValidationFolds` — Eğitim sırasında çapraz doğrulama kat sayısı.
 - `cfg.ModelSavePath` — Eğitilen modelin kaydedileceği dosya yolu.
 - `cfg.modelData`, `cfg.trainedModel` — Eğitimli model dosyasının yüklenmesi ve hazır hâlde tutulması.
 - `cfg.threshold` — Tahmin güven eşiği.
 - `cfg.labelMap` — Tahmin sonucuna karşılık gelen jest etiketleri (örn: 0 → "13 - Come").
- **FFT ve Özellik Çıkarımı Ayarları:** `cfg.FFTWindowSize`, `cfg.PeakDetectionThreshold`, `cfg.FeatureNormalization` — Zaman-frekans analizlerinde kullanılan pencere boyutu, tepe algılama eşiği ve normalizasyon ayarı.
- **Loglama ve Hata Ayıklama:** `cfg.EnableDetailedLogging`, `cfg.LogFolder` — Sistem çalışırken ayrıntılı log kaydının tutulup tutulmayacağı ve logların saklanacağı klasör.

sensors.m

ESP32 ile haberleşmeyi yönetir:

- `serialport` üzerinden bağlantı kurar
- "DATA_START" ve "DATA_END" arasında gelen sensör verilerini okur
- Okunan verileri `acc`, `gyro`, `ts` matrislerine dönüştürür

gesture.m

Kullanıcıyla etkileşim sağlar:

- Jest ismi girdisi alır ve doğrular, gerekirse yeni klasör oluşturur
- Kullanıcıya kayıt yönergelerini adım adım sunar

files.m

Veri kaydetme ve dosya yönetimi işlemlerini yürütür:

- Geçici veri dosyasını oluşturur
- Kullanıcının onayıyla veriyi kalıcı dosyalara ekler (MAT & CSV)
- Sensör verilerini grafik olarak çizer ve PNG dosyaları olarak saklar
- Metadata girişi alır ve `metadata.csv` dosyasına yazar

classification.m

Sensör verisinden özellik çıkarımı yapar:

- **İstatistiksel Özellikler:** ortalama, standart sapma, RMS, aralık, vs.
- **Zamansal/Frekans Özellikleri:** FFT ile baskın frekans ve genlik
- Her eksen için ayrı ayrı hesaplama yapar

modelTraining.m

Kayıtlı verileri işler ve makine öğrenmesi modeli için hazırlar:

- Hedef jestleri filtreler, her kayıt için `classification` modülüyle özellik çıkarır
- Özellikleri tabloya dönüştürür ve `gesture_features.mat` + `.csv` olarak kaydeder
- Kullanıcıya model eğitimi için yönergeler içeren bir `README.txt` üretir

plotData.m

Tüm veriler için grafik üretimi yapar:

- Her bir kayıt için hız ve ivme grafiklerini oluşturur
- PNG formatında dosya olarak kaydeder

6. Karşılaşılan Sorunlar ve Çözümler

- **MPU6050 Temassızlık Problemi:** Başlangıçta sensör pinlerinde temassızlık sorunu yaşanmıştır. Bu durum, sensör pinlerinin lehimlenmesiyle kalıcı olarak çözülmüştür.
- **ESP32'nin Kola Yerleştirilmesinde Fiziksel Kısıt:** Geliştirme kartının genişliği nedeniyle optimal bir breadboard bulunamıştır. Bu nedenle ESP32'nin gücünü kesmek için kullanılması tasarlanan anahtar tasarımdan çıkarılmıştır.
- **Bluetooth Bağlantı Sorunu:** ESP32 güç kaynağından çıkarılmadan arka arkaya veri toplamaya çalışıldığında Bluetooth bağlantısı kurulamamaktaydı. Bu durum, MATLAB kodunda bağlantı kontrolü ve istisna yönetimi yapılarak başarılı şekilde çözülmüştür.
- **Yanlış Tahmin Riski:** Eğitim verilerinin yetersiz çeşitliliği, sınıflandırma doğruluğunu düşürebilir. Bu problem daha fazla ve çeşitli veri kullanılarak çözülebilir.

7. Potansiyel Geliştirmeler

Geliştirilen sistem modüler ve esnek yapısıyla farklı alanlarda kullanılabilecek şekilde genişletilebilir. İlerleyen aşamalarda aşağıdaki geliştirme fikirleri uygulanabilir:

- **Görme Engelliler İçin Sesli Destek:** Sisteme gelişmiş ses sistemleri entegre edilerek tanınan jestlerin sesli bildirimler ile aktarılması sağlanabilir. Bu şekilde görme engelli bireylerin sistemden etkin biçimde faydalanması mümkün olur.
- **İşaret Dili Tanıma ve Seslendirme:** Parmaklara ayrı ayrı yerleştirilecek IMU sensörleri ile jest tanımanın ötesine geçilerek, işaret dili ifadeleri tanınabilir ve sesli çıktılarına dönüştürülebilir. Bu, işitme ve konuşma engellilerle iletişim kurmayı kolaylaştıran bir araç haline gelebilir.
- **Trafik Kontrol Uygulamaları:** Trafik polislerine sağlanacak bu sistem sayesinde jest ve el hareketleriyle trafik akışını kontrol ederken ışıkların da senkronize edilmesi mümkün hale getirilebilir.
- **Oyun ve Eğlence Uygulamaları:** Geliştirilen sistem, vücut hareketleriyle oyun kontrolü sağlayarak hem eğlenceli hem de fiziksel etkileşimli bir deneyim sunabilir. Özellikle sanal gerçeklik ve artırılmış gerçeklik uygulamaları için potansiyel taşır.
- **Kullanıcı Arayüzü Geliştirme:** MATLAB arayüzü geliştirilebilir. Kullanıcıların veri toplama, model eğitimi ve test işlemlerini daha kolay ve görsel olarak gerçekleştirebileceği bir arayüz geliştirilebilir.
- **Donanım Sabitleme:** Şu anda breadboard üzerine kurulu olan sistem kola elastik bantlarla takılmaktadır. Bu yapının daha sağlam, ergonomik ve sabit bir forma dönüştürülmesi için özel 3D baskı aparatlar veya tekstil entegreli bileklikler kullanılabilir.