

# PROJE RAPORU

**Proje Adı:** Kestirimli Hedef Tespit Uygulaması

**Yazar:** Özgür DEMİRCAN

**Tarih:** 7 Eylül 2025

## Özet

Bu rapor, dağıtık bir sistem mimarisi kullanarak iki boyutlu uzayda bir hedefin sensör verileriyle tespit edilmesini simüle eden projenin aşamalı gelişimini ve çoklu teknoloji entegrasyonunu sunmaktadır.

Projenin ilk aşamasında, sistemin temel mimarisini ve algoritmalarını doğrulamak amacıyla C/C++ ve POSIX kütüphaneleri (Sockets, Pthreads) kullanılarak, komut satırı üzerinden çalışan bir arka plan (backend) sistemi geliştirilmiştir. Bu sistem, mimarinin ve temel algoritmaların doğruluğunu kanıtlamıştır.

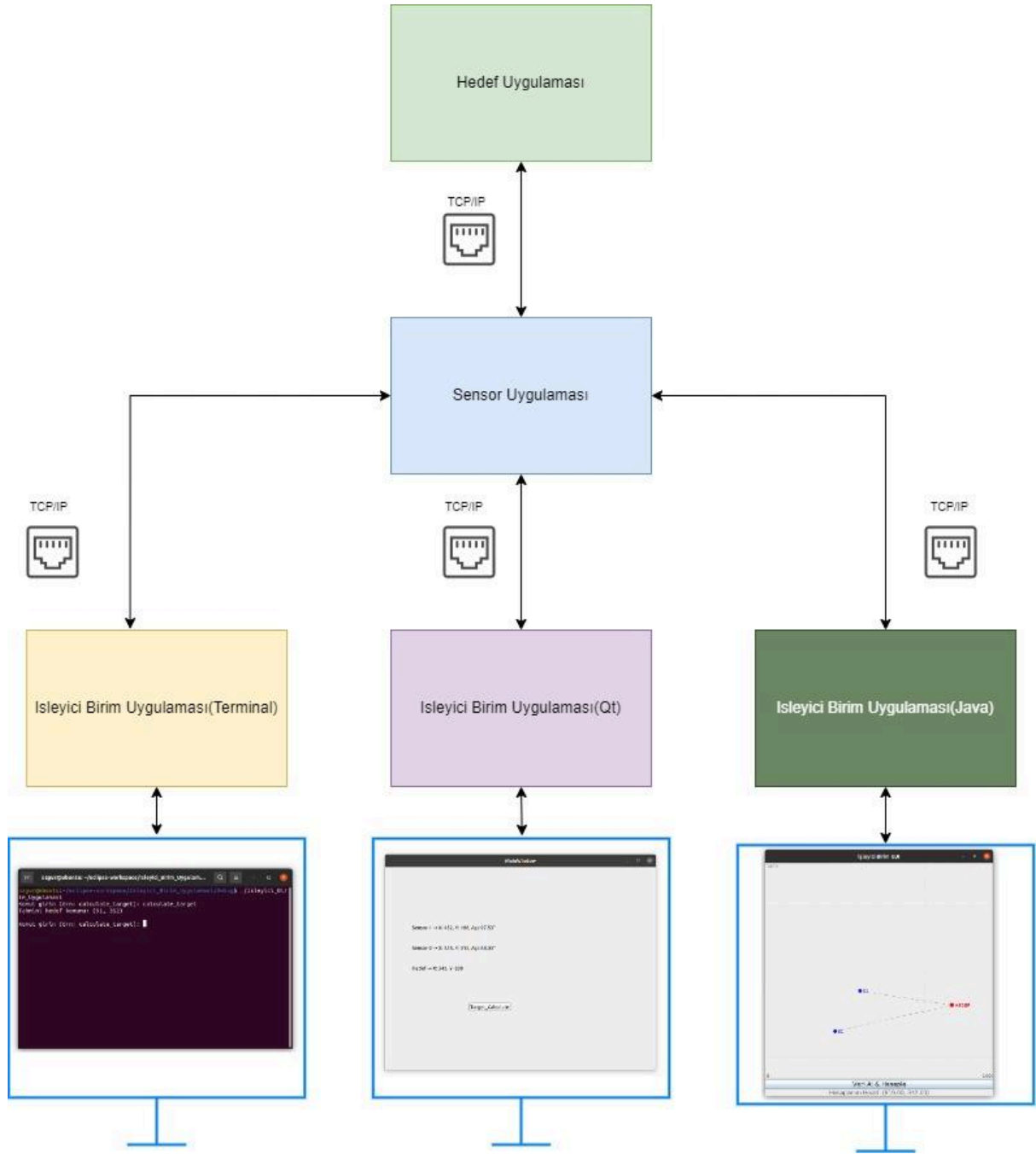
Ardından, projenin platform bağımsızlığını ve kullanıcı etkileşimini artırmak için iki farklı grafik kullanıcı arayüzü (GUI) geliştirilmiştir. Birincisi, C++ tabanlı Qt framework'ü kullanılarak oluşturulan interaktif bir uygulamadır. İkincisi ise, tüm altyapının Java'ya taşınmasıyla geliştirilen, Java Swing ile zenginleştirilmiş bir GUI uygulamasıdır.

Rapor, her üç uygulamanın (C/C++, Qt, ve Java) mimari tasarımını, kullanılan teknolojileri, uygulamanın dayandığı matematiksel temelleri ve bu çok aşamalı geliştirme sürecinde karşılaşılan zorluklara bulunan çözümleri detaylandırmaktadır.

## 1. Giriş

Bu rapor, bir hedefin konumunu tespit etmek için dağıtık bir sistem mimarisi kullanan uygulamanın bileşenlerini, işleyişini ve temel prensiplerini açıklamaktadır. Sistem, TCP/IP protokolü üzerinden haberleşen farklı modüllerden oluşmaktadır.

## 2. Sistem Mimarisi ve Veri Akışı



Şekil 1:Sistem Tasarımı

Sistem, üç ana katmanlı bir yapıya sahiptir: veri kaynağı, merkezi işlem birimi ve istemci uygulamaları.

- **Hedef Uygulaması:** Sistemin hedef verisini sağlayan kaynaktır. Rastgele (x, y) koordinatları üretir ve kendisine gelen istekler üzerine bu verileri TCP üzerinden Sensör Uygulaması'na gönderir.

- **Sensör Uygulaması:** Sistemin merkezi işlem birimidir. Hedef Uygulamasından aldığı konum bilgisini, kendi yazılımındaki sensör konumları ile karşılaştırır. Sensör ve hedef arasındaki açı bilgilerini hesaplar ve bu veriyi bağlı olan tüm İşleyici Birim uygulamalarına iletir.
- **İşleyici Birim Uygulamaları:** Farklı teknolojilerle geliştirilmiş, Sensör Uygulamasından gelen verileri işleyip kullanıcıya sunan istemcilerdir.
  - **C/C++ Uygulaması:** Terminal tabanlı bir uygulamadır. Gelen verilerle hedef konumunu matematiksel olarak hesaplar ve sonuçları metin olarak gösterir.
  - **Qt Uygulaması:** Grafik kullanıcı arayüzü (GUI) sunan bir uygulamadır. Verileri kullanarak hedefi ve sensörleri görselleştirir.
  - **Java Uygulaması:** Hedef ve sensör konumlarını bir grafik üzerinde noktalar (points) halinde gösterir. Sensörlerden hedefe doğru yönü belirten bir ok çizerek görselleştirmeyi zenginleştirir.

### 3. Temel Matematiksel Prensipler

Sistem, geometri ve trigonometriye dayalı iki temel matematiksel işlemi kullanır:

- **Açı Hesaplaması:** Sensör Uygulaması, bir sensör konumu ( $S_x, S_y$ ) ve hedef konumu ( $H_x, H_y$ ) verildiğinde, aralarındaki açıyı atan2 fonksiyonu ile hesaplar. Bu, sensörden hedefe olan vektörün yatay eksenle yaptığı açıyı bulmayı sağlar.

$$\theta = \text{atan2}(H_y - S_y, H_x - S_x)$$

- **Hedef Konumu Tespiti:** İki sensörün konumları ve bu sensörlerden hedefe olan açılar bilindiğinde, hedefin konumu iki doğrunun kesişim noktası olarak bulunur.

**Sensör 1:** Konumu ( $x_1, y_1$ ) ve hedefe olan açısı  $\theta_1$ .

**Sensör 2:** Konumu ( $x_2, y_2$ ) ve hedefe olan açısı  $\theta_2$ .

Öncelikle, her iki sensörden hedefe uzanan doğruların eğimleri ( $m_1$  ve  $m_2$ ) hesaplanır. Eğim, açının tanjantına eşittir.

$$m_1 = \tan(\theta_1) \quad m_2 = \tan(\theta_2)$$

Bu eğimler kullanılarak, hedefin ( $x, y$ ) koordinatları aşağıdaki formüllerle bulunur:

$$x = (m_1 x_1 - m_2 x_2 - y_1 + y_2) / (m_1 - m_2)$$

$$y = (m_1 m_2 (x_1 - x_2) + m_1 y_2 - m_2 y_1) / (m_1 - m_2)$$

**Not:** Bu formüller, her iki doğruyun eğimlerinin farklı olduğu ( $m_1 \neq m_2$ ) durumda geçerlidir. Eğer eğimler eşitse ( $m_1 = m_2$ ), bu, sensörlerin aynı doğru üzerinde olduğunu ve doğruların paralel olduğunu gösterir, bu durumda tek bir kesişim noktası bulunamaz.

### 3. İletişim Protokolü

Tüm sistem bileşenleri arasındaki veri akışı, TCP (Transmission Control Protocol) üzerinden sağlanır. TCP'nin kullanımı, verilerin güvenilir ve sıralı bir şekilde iletilmesini garanti eder, bu da hassas konum ve açı bilgilerinin doğru bir şekilde iletilmesi için kritik öneme sahiptir.

### 4. Geliştirme Ortamı

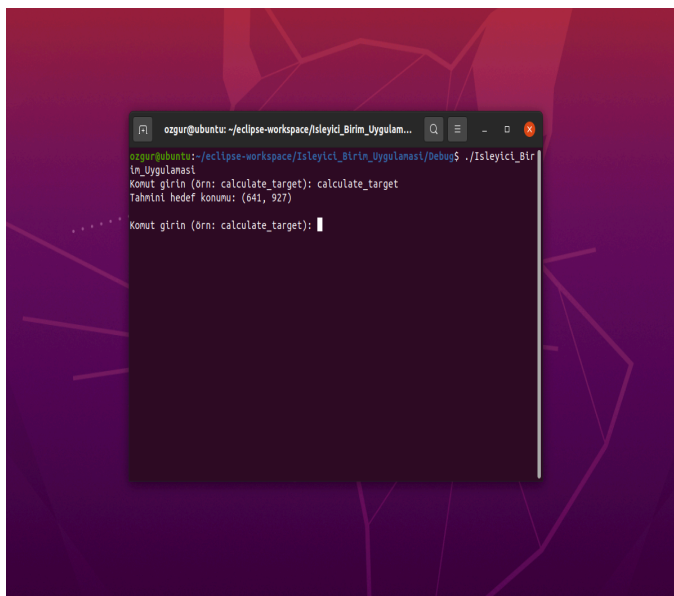
Projenin tüm geliştirme süreçleri Ubuntu işletim sistemi üzerinde gerçekleştirilmiştir.

- **C/C++ ve Terminal Uygulaması:** Uygulamanın arka plan ve terminal arayüzü, Ubuntu'nun yerleşik araçları ve GCC/G++ derleyicisi kullanılarak komut satırı ortamında geliştirilmiştir.
- **Qt Uygulaması:** Grafik kullanıcı arayüzü, Ubuntu üzerine kurulan Qt Creator IDE ve Qt 5 kütüphaneleri ile geliştirilmiştir.
- **Java Uygulaması:** Java tabanlı tüm bileşenler, bir Entegre Geliştirme Ortamı (IDE) kullanılmadan, doğrudan Ubuntu terminalinden JDK komutları (javac ve java) ile derlenip çalıştırılarak geliştirilmiştir.

### 5. Uygulama Çıktıları ve Kaynak Kod

Aşağıda, uygulamanın farklı arayüzlerinden alınan ekran görüntüleri ve proje kodlarının bulunduğu bağlantı yer almaktadır.

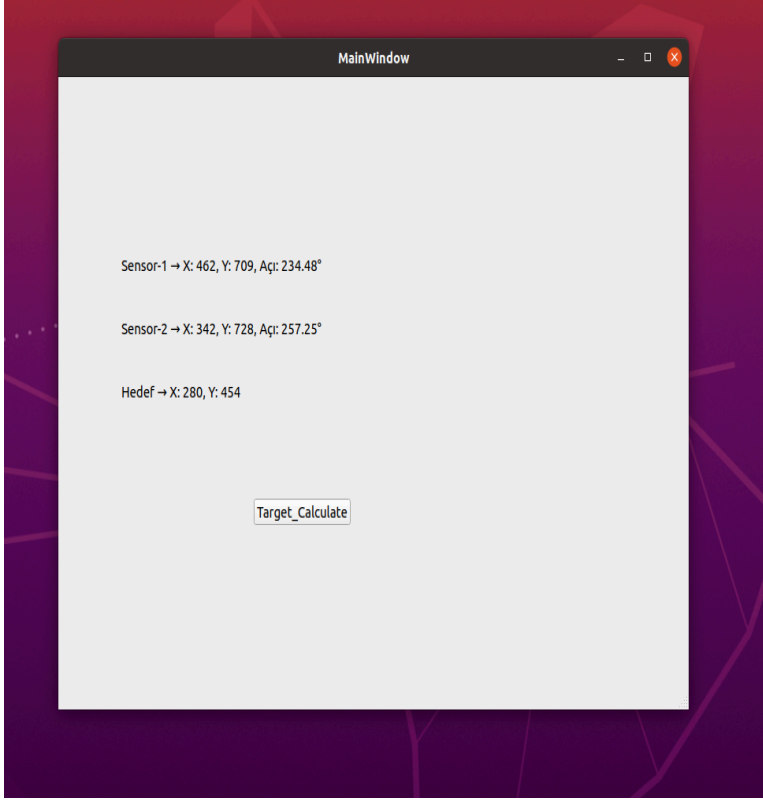
- **Terminal Çıktısı:**



```
ozgur@ubuntu:~/eclipse-workspace/Isleyici_Birim_Uygulam...  
ozgur@ubuntu:~/eclipse-workspace/Isleyici_Birim_Uygulamasi/Debug$ ./Isleyici_Bir_Uygulama  
Komut girin (örn: calculate_target): calculate_target  
Tahmini hedef konumu: (641, 927)  
Komut girin (örn: calculate_target):
```

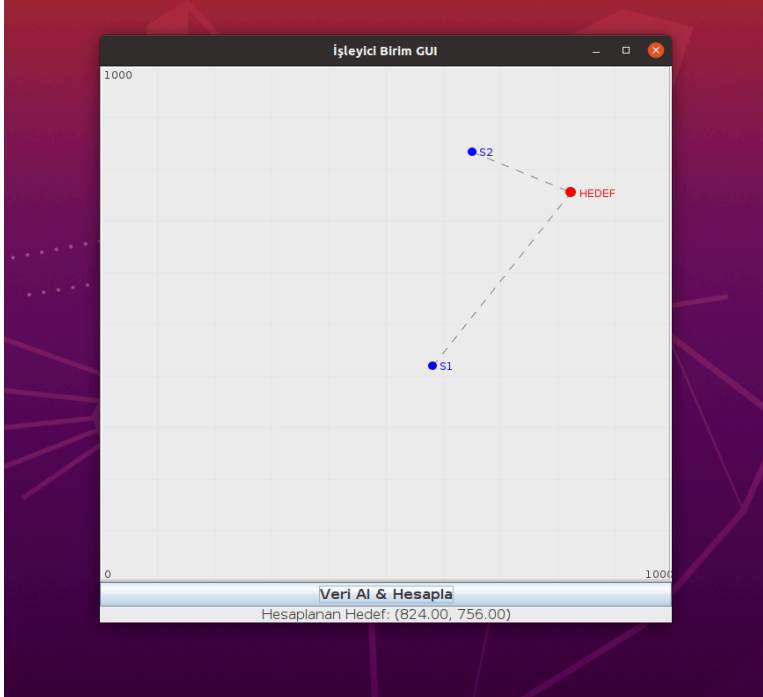
Şekil 2: Ekran Görüntüsü

- **Qt Uygulaması Ekran Görüntüsü:**



Şekil 3: Ekran Görüntüsü

- **Java Uygulaması Ekran Görüntüsü:**



Şekil 4: Ekran Görüntüsü

Projenin tüm kaynak kodlarına aşağıdaki adresten erişilebilir:  
**<https://github.com/ozgurdmrcn/Hedef-Tespit-Uygulamasi->**

## **5. Sonuç**

Bu proje, yazılım mühendisliği prensiplerini (modülerlik, dağıtık sistemler) ve temel matematiksel hesaplamaları bir araya getiren başarılı bir uygulamadır. Farklı diller ve teknolojiler kullanarak aynı veriyi işleyebilme yeteneği, sistemin esnekliğini ve genişletilebilirliğini kanıtlamaktadır.