

WPF Tabanlı TCP Koordinat ve RTSP Video Yayını Uygulaması Kullanım Kılavuzu

Erenalp Özcan

25 Temmuz 2025

Giriş

Bu belgede, .NET 8 ile geliştirilen iki masaüstü uygulamadan (ServerApp ve ClientApp) oluşan staj projesinin kullanım senaryosu detaylı ve teknik düzeyde açıklanacaktır. Uygulamalar TCP/IP üzerinden haberleşmekte, istemci tarafında RTSP protokolü ile canlı video yayını başlatılmakta ve sunucu tarafında bu yayın alınmaktadır. Ayrıca harita üzerinden koordinat seçimi, hedef işaretleme ve canlı saat ile mermi bilgisi gösterimi gibi özellikler mevcuttur.

1 Teknik Mimari

Proje aşağıdaki temel bileşenlerden oluşmaktadır:

- **ClientApp:** WPF tabanlı, kullanıcıdan koordinat alır ve TCP/IP üzerinden ServerApp'e gönderir.
- **ServerApp:** TCP sunucu olarak çalışır, gelen koordinatları işler, harita ve video akışı ile görsel olarak sunar.
- **Video Yayını:** ClientApp tarafından RTSP protokolüyle başlatılır, ServerApp tarafından LibVLCSharp kullanılarak alınır.
- **Harita Entegrasyonu:** LeafletJS ve WebView2 ile entegre edilmiştir.
- **Kamera Erişimi:** ServerApp tarafında OpenCV üzerinden canlı görüntü alınır.

2 İletişim Akışı (Özet)

Uygulamalar arası tipik bir iletişim senaryosu şu adımlardan oluşur:

1. Kullanıcı ClientApp üzerinden sunucuya bağlanır.
2. Harita üzerinden koordinat seçimi yapılır.
3. Açıklama ile birlikte koordinatlar sunucuya TCP üzerinden gönderilir.
4. Sunucu, gelen veriyi çözümleyerek haritada hedef simgesi oluşturur.

5. Kullanıcı, video yayını başlatır ve RTSP üzerinden aktarım başlar.
6. Sunucu video akışını alır ve ekranda gösterir.
7. Kullanıcı “Ateş Et” komutu gönderir, sunucu hedefi işaretler ve mermi sayısını azaltır.

3 Kullanılan Teknolojiler

- .NET 8 (WPF - C)
- TCP/IP Socket Programlama
- RTSP (VLC Media Player ile video yayını)
- LibVLCSharp (.NET ile video oynatma)
- LeafletJS (Harita arayüzü)
- OpenCVSharp (Kamera erişimi ve görüntü işleme)
- WebView2 (HTML harita gösterimi)
- SQLite (veri saklama altyapısı)

4 1. Uygulamaların Başlatılması

4.1 1.1 ServerApp’ın Otomatik Başlatılması

ServerApp çalıştırıldığında, TCP dinlemesi otomatik başlar. Kod içerisindeki ilgili satır:

```
StartTcpServer(); // MainWindow constructor içinde arlr .
```

Harita, RTSP video akışı, hedef işareti, canlı saat ve mermi sayacı gibi overlay bileşenleri uygulama başladığında otomatik olarak yüklenir.



4.2 1.2 ClientApp Bağlantısı

ClientApp açıldığında, IP ve port bilgileri girilip "Bağlan" butonuna basılır. Bu işlem ile sunucuya TCP bağlantısı kurulur:

```
_tcpClient = new TcpClient();  
_tcpClient.Connect(ip, port);  
_stream = _tcpClient.GetStream();
```

Başarıyla bağlanıldığında "Durum: Bağlı" yeşil renkte görünür. Aynı anda heartbeat mesajları gönderilmeye başlanır.



5 2. Harita Üzerinden Koordinat Seçimi

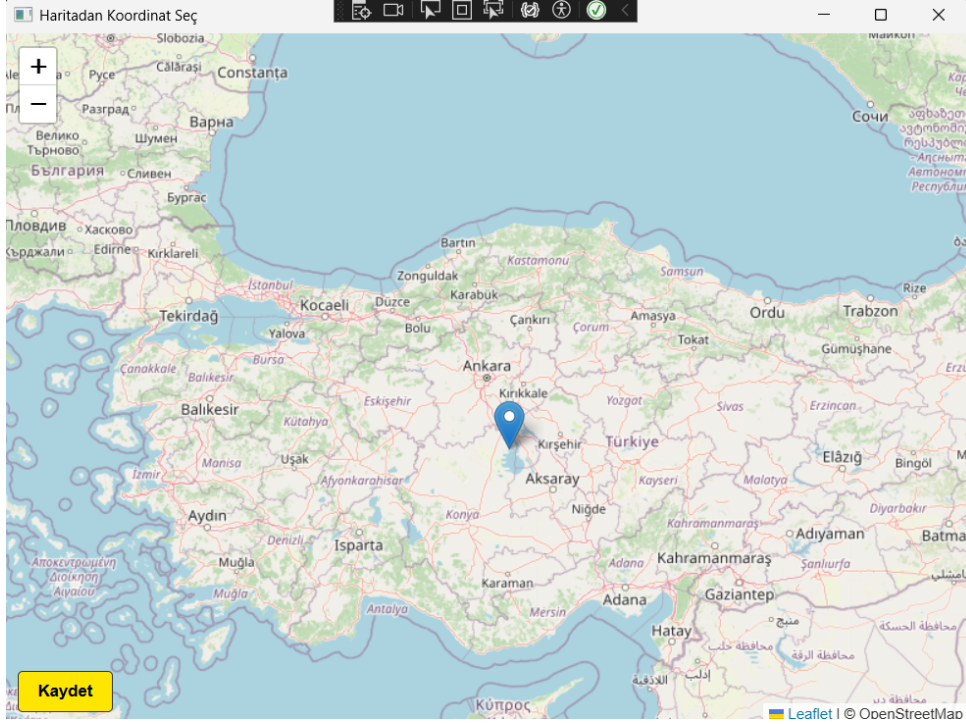
ClientApp içerisindeki "Haritadan Seç" butonuna basıldığında, 'Map/map.html' dosyası WebView2 ile açılır. Kullanıcı haritadan bir alan seçer:

HTML tarafında seçilen koordinat şu şekilde C tarafına aktarılır:

```
window.chrome.webview.postMessage({ lat: selectedLat, lng:  
    selectedLng });
```

C tarafında bu koordinatlar almarak ilgili textbox'lara yazılır:

```
LatitudeTextBox.Text = lat.ToString(CultureInfo.InvariantCulture)  
    ;  
LongitudeTextBox.Text = lng.ToString(CultureInfo.InvariantCulture  
    );
```

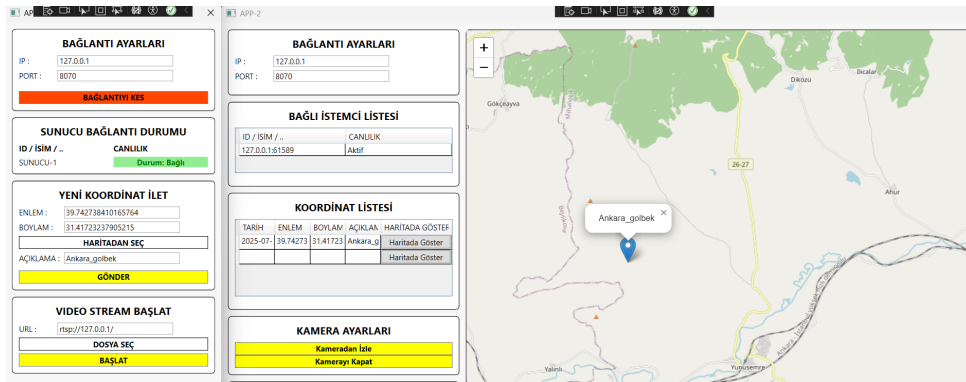


6 3. Koordinat ve Açıklama Gönderme

Kullanıcı koordinat seçip açıklama girdiğinde “Gönder” butonuna basar. Aşağıdaki gibi TCP ile veri gönderilir:

```
writer.Write((byte)MessageType.Coordinate);
writer.Write(lat);
writer.Write(lon);
writer.Write(descBytes.Length);
writer.Write(descBytes);
writer.Flush();
```

Sunucu taraf nda gelen veri zlererek harita zerine hedef simgesi yerle tirilir.

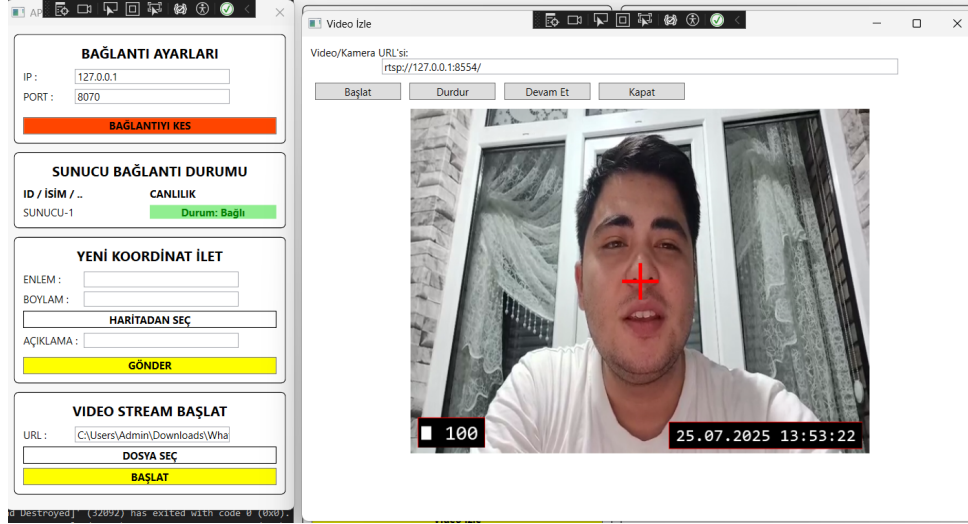


7 4. RTSP Video Yayını Başlatma

ClientApp üzerinden “Video Seç” ve ardından “Yayını Başlat” butonlarına basılır. VLC player ile video yayını başlatılır:

```
string args = $"\"{videoPath}\" _ --sout=\"#rtp{{sdp=rtsp://{ip}:8554/}}\"";
```

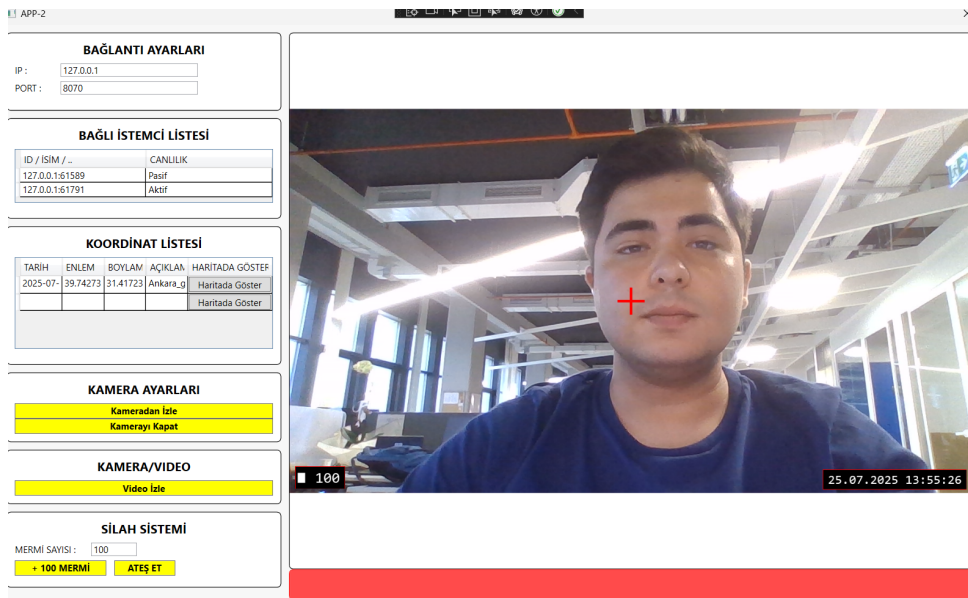
Sunucu tarafında bu RTSP yayını alınır ve LibVLCSharp ile gösterilir:



8 5. Kameradan İzle (OpenCV ile)

ServerApp tarafında, istemciden bağımsız olarak OpenCV ile kamera erişimi sağlanır:

```
_videoCapture = new VideoCapture(0); // 0 = varsayılan kamera  
var frame = _videoCapture.RetrieveMat();  
Cv2.ImShow("Kamera", frame);
```

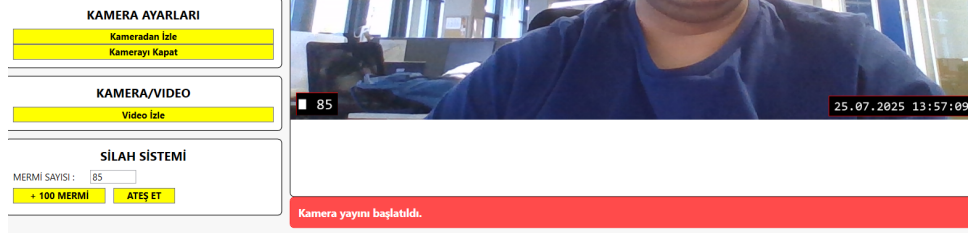


Kullanıcı bu görüntüyü “Kameradan İzle” sekmesinden izleyebilir.

9 6. Ateş Et ve Mermi Sayısı

ClientApp üzerindeki “Ateş Et” butonuna basıldığında sunucuya mermi ateşleme bilgisi gönderilir. Sunucu tarafında ekranda mermi sayısı güncellenir ve hedef üzerine animasyon çizilir:

```
// ServerApp iindeki overlay izimi
_drawContext.DrawText($"Mermi: {bulletCount}", ...);
_drawContext.DrawEllipse(...); // hedef i areti
```

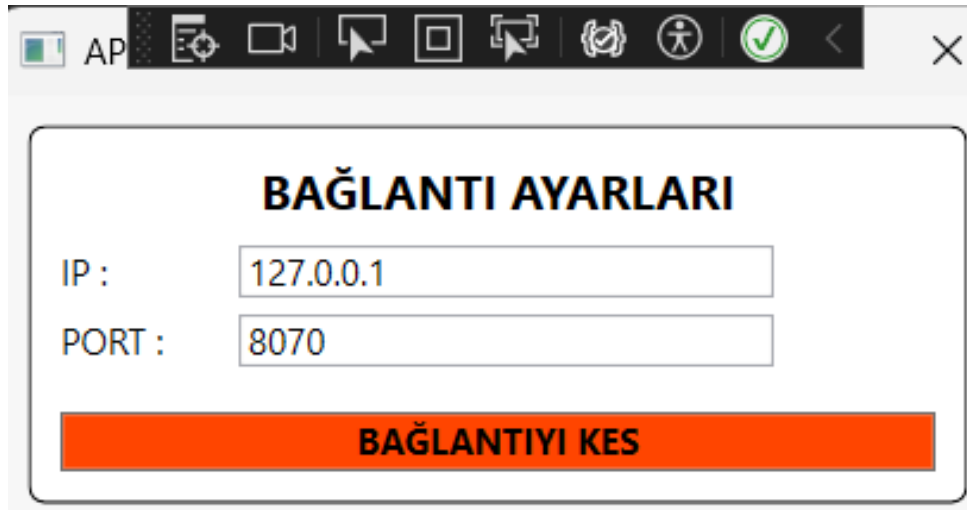


10 7. Bağlantı Kesme ve Kapanış

İstemci kapatıldığında:

- Heartbeat thread durur
- TCP bağlantısı kapanır
- VLC işlemi sonlandırılır
- Server ekranında bağlantı durumu güncellenir

```
if (_vlcProcess != null && !_vlcProcess.HasExited)
{
    _vlcProcess.Kill();
    _vlcProcess.Dispose();
}
```



BAĞLI İSTEMCİ LİSTESİ	
ID / İSİM / ..	CANLILIK
127.0.0.1:61867	Pasif

Sonuç

Bu dökümanda geliştirilen istemci ve sunucu uygulamalarının adım adım kullanım senaryosu detaylandırılmıştır. RTSP video yayını, koordinat işaretleme, kamera gösterimi, harita üzerinde hedef seçimi ve mermi takibi gibi tüm işlevler ekran görüntüleriyle desteklenmiş ve kod seviyesinde açıklanmıştır.