

3. ArmClient Service

```
// ArmClient.cs

using System;
using System.IO;
using System.Net.Sockets;
using System.Text.Json;
using System.Threading;
using System.Threading.Tasks;
using RoboticArmWeb.Models;

public class ArmClient : IDisposable
{
    private readonly TcpClient _tcp;
    private readonly StreamWriter _writer;
    private readonly StreamReader _reader;
    private readonly CancellationTokenSource _cts = new();

    private Telemetry _latest = new Telemetry();

    public ArmClient(string ip = "192.168.4.1", int port = 115200)
    {
        _tcp = new TcpClient();
        _tcp.Connect(ip, port);
        var stream = _tcp.GetStream();
        _writer = new StreamWriter(stream) { AutoFlush = true };
        _reader = new StreamReader(stream);
    }
}
```

```

        Task.Run(ReadLoopAsync);
    }

    private async Task ReadLoopAsync()
    {
        while (!_cts.IsCancellationRequested)
        {
            try
            {
                var line = await _reader.ReadLineAsync();
                if (line is null) break;
                var tel = JsonSerializer.Deserialize<Telemetry>(line);
                if (tel != null) _latest = tel;
            }
            catch
            {
                // ignore malformed JSON or network hiccups
            }
        }
    }

    public Telemetry GetLatestTelemetry() => _latest;

    public async Task SendCommandAsync(Command cmd)
    {
        var json = JsonSerializer.Serialize(cmd);
        await _writer.WriteLineAsync(json);
    }

```

```
public void Dispose()
{
    _cts.Cancel();
    _tcp.Close();
}
}
```