



# Platform ve Faydalı Yük Elektronik Tasarım Departmanı

## Güneş Takip Sisteminin Geliştirilmesi

---

Sefa SAYRACI

Stajyer

12 Eylül 2023

Tübitak UZAY

Resim ve tablo numaraları ekle.

USART EKLE

Söyleyeceklerini deftere not al.

RS Haberleşme ve Uzaydaki kamera modülleri kaldı.

• Ben Kimim.....	1
• Proje Detay .....	
• Proje Bölümleri .....	
• Teknik Araştırmalar .....	
• Aviyonik Yazılım Alt Tasarımı .....	
• Kontrol Arayüz Yazılım Alt Tasarımı .....	
• Donanım Alt Tasarımı .....	
• Kullanılan Elektronik Komponentleri .....	
• Elektronik Kart Tasarımı .....	
• IMU 3D Kontrol Yazılım Alt Tasarımı .....	
• Hedef Takip Sistemi (ESP CAM Modülü) .....	
• Hedef Takip Sistemi (Harici Kamera Modülü) .....	
• Kaynakça .....	
• İnternet Kaynakçaları .....	
• Teşekkürler .....	

## Ben Kimim?

Ben Sefa SAYRACI

Gazi Üniversitesi Elektrik Elektronik Mühendisliği 4. sınıf

1. zorunlu staj: Baykar Teknoloji Yapay Zeka Yazılım Departmanı
2. gönüllü staj: Group R & D and Energy Control Systems Gömülü Yazılım Departmanı
3. zorunlu staj: TÜBİTAK Uzak Platform ve Faydalı Yük Elektroniği Tasarım Departmanı

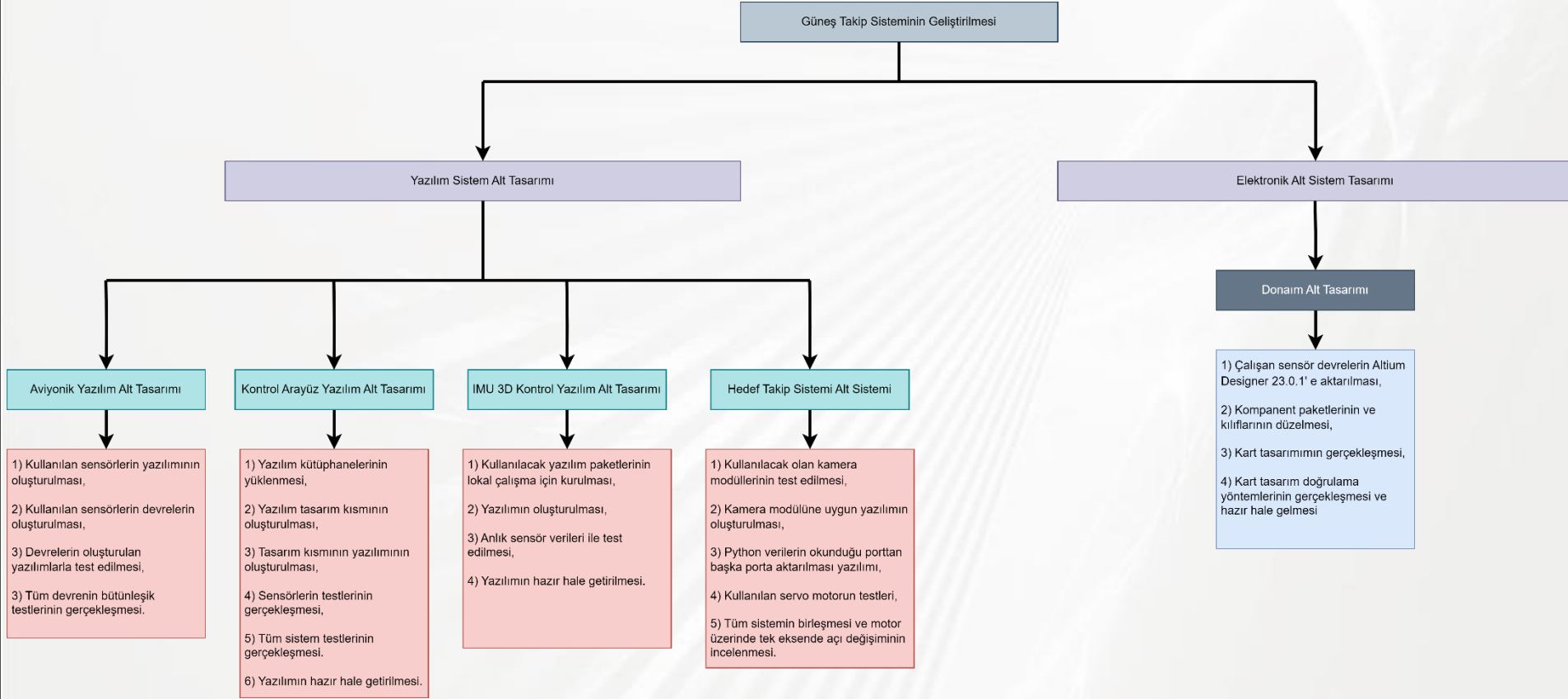
## Yaptığım Projeler

- TÜBİTAK 2209-A: Güneş Takip Sisteminde Derin Öğrenme ile Hedef Tespiti ve Fpga ile Sistem Kontrolünün Geliştirilmesi
- Real Time sistemlerde Integrity, GIS, VXWorks ile teoriksel araştırma yaptım. Bunun doğrultusunda RTOS'un mikrodeneleyiciye iplemente oluşu hakkında yazılımsal giriş ve atanan taskları gerçekleştirmesi üzerinedir.
- SimpleFOC Kütüphanesi ile Manyetik Motor Üzerinden Hedef Takip Sistemi-BLDC Motor
- Projemde motoru step motor şeklinde kullanacağımdan dolayı yazılımsal olarak SimpleFOC kütüphanesi ile yazılımsal üzerinde geliştirdim. Bu sisteme PID ekleyerek motorun istediğim açısal hareketini gerçekleştirdim.

Proje kapsamında «Güneş Takip Sistemi» hedefinde geliştirme yapılmıştır. Bu süreçte güneş takip sistemi üzerinde mevcut olacak telemetri sensörleri için oluşturulmuş olan C tabanlı yazılım ile verilerin anlık olarak gösterileceği bilgisayar üzerine iletilip, C# üzerinden geliştirilmiş olan arayüz uygulamasında verilerin anlık olarak text ve grafik şeklinde yansıtılması hedeflenmiştir. Ayrıca bu verilerin hepsi Excell üzerine kaydedilmektedir. Telemetri sensöründen gelen anlık açı verileri ile Python üzerinde poly3D kütüphanesi import edilerek anlık eksenel görünümü de gösterilmiştir. Bu sensörlerin daha kolay kontrolünün sağlanması için Altium Designer 23.0.1 üzerinden elektronik kart tasarımı gerçekleştirilmiştir.



**Resim 1.** Tübitak UZAY İMECE ve GÖKTÜRK-2 Resmi



Tablo 1. Proje Bölümleri

## Uzayda Kullanılan Kamera Sistemleri:

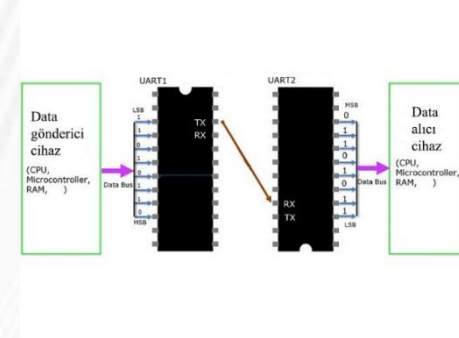
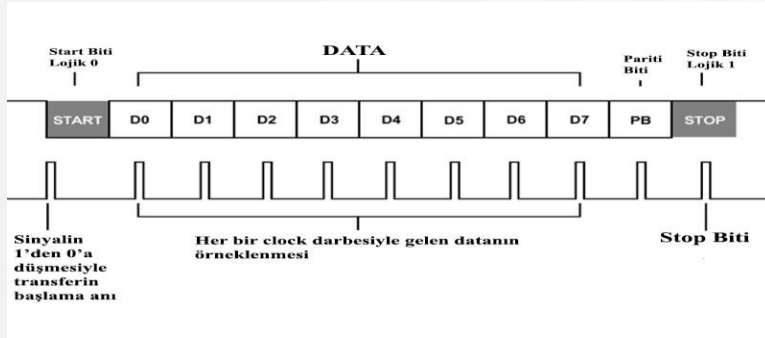
## Haberleşme Protokolleri:

**UART:** UART açılımı Universal Asynchronous Receiver Transmitter olarak isminden de Asenkron haberleşme olduğu anlaşılmaktadır. Asenkron olması sonucunda da ortak bir clock sinyali yoktur. Amacı verileri seri şekilde iletmektir. İki cihaz arasında her seferinde bir bit veri göndermenin bir yoludur. İki adet Tx (Gönderici) ve Rx (Alıcı) olarak pine sahiptir. UART'lar genellikle mikrodenetleyicileri bilgisayarlar, sensörler ve aktüatörler gibi diğer cihazlara bağlamak için kullanılır.

RS-232 ara yüzleri ve harici modemler ve bunlara benzer kullanan cihazlar da UART kullanılan yerlerin çok sık örnekleri olmaktadır.

Düşük voltaj seviyelerinde kullanılır.

Parite bit opsiyonel kullanılır. Veri bitlerin toplamının çift veya tek olmasını anlamak için kullanılır. Alıcı tarafında orijinal verilerde hata tespiti ve verilerin düzeltilmesi amacıyla kullanılır.



Resim 1.1 UART Protokolü



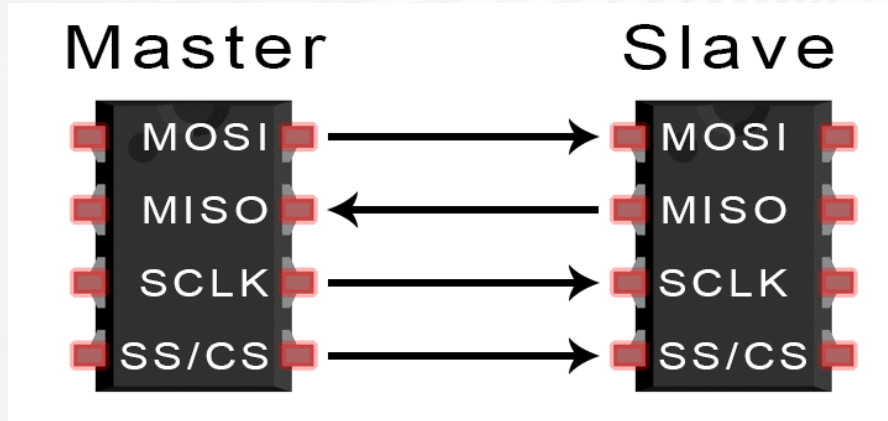
**SPI:** SPI açılımı Serial Peripheral Interface olarak bilinmektedir. Seri çevresel arayüz (SPI), yüksek hızlı bir seri iletişim protokolüdür. SPI genellikle mikrodenetleyicileri flash bellek ve SD kartlar gibi bellek aygıtlarına bağlamak için kullanılır.

Senkron haberleşme yapar.

Yüksek hızlarda veri iletimini destekler ve tam çift yönlü iletişim sağlar, yani aynı anda hem veri gönderme hem de almayı mümkün kılar.

Full Duplex modda çalışır.

SPI ile bir master cihaz birden fazla slave cihazı kontrol edebilir. Ana cihaz, iletişim kurmak istediği slave cihazı seçer.



SCLK: Serial Clock  
MOSI: Master out Slave in  
MISO: Master in Slave out  
SS/CS: Slave Select/Chip Select

Resim 1.2 SPI Protokolü

**I<sup>2</sup>C:** I2C açılımı Inter-Integrated Circuit, Entegre devreler arası (I2C) düşük hızlı bir seri iletişim protokolüdür. I2C genellikle mikrodenetleyicileri sensörlere ve aktüatörlere bağlamak için kullanılır.

Senkron haberleşmedir.

Kısa mesafelerde iyidir.

Çok master destekler.

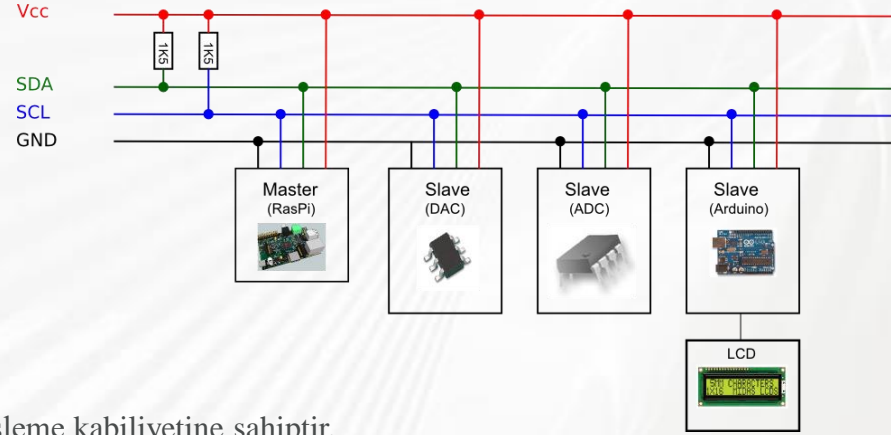
Uzun mesafelerde iyiler.

1 adet data 1 adet clock sinyali vardır.

Veri yolu üzerinde çoklu Master ve çoklu Slave'leri işleme kabiliyetine sahiptir.

Clock senkronizasyonu gibi önemli durumları mevcuttur.

Master tarafından start ve stop koşullarıyla haberleşme başlar ve biter.



**Resim 1.3** I2C Protokolü Örnek Master Slave Dağılımı



SCL: Serial Clock Line  
SDA: Serial Data Line

**Resim 1.4** I2C SCL SDA Sinyalleri

## I<sup>2</sup>C ve SPI Arasındaki Farklar:

İkisi de senkron haberleşmedir. Arasındaki farkları inceleyelim;

- I<sup>2</sup>C half duplex, SPI full duplex
- I<sup>2</sup>C 2 tele ihtiyaç duyar, SPI 3 ya da 4.
- I<sup>2</sup>C SPI'ya göre yavaştır.
- I<sup>2</sup>C SPI'ya göre daha fazla güç çeker.
- I<sup>2</sup>C gürültüye SPI'ya göre daha az duyarlıdır.
- I<sup>2</sup>C SPI'dan daha ucuzdur.
- I<sup>2</sup>C tel ve logic üzerinde çalışır ve pull-up direnci vardır. SPI'da pull-up direncine gerek yoktur.
- I<sup>2</sup>C haberleşmesinde her bytetan sonra onay biti alırız. Bunu SPI desteklemez.
- I<sup>2</sup>C verinin karşıdan alındığını garantiler. SPI bunu doğrulamaz.
- I<sup>2</sup>C çok master destekler. SPI sadece 1 master.
- I<sup>2</sup>C aynı bus üzerinde aygıt adresleri sayesinde çok aygıtı destekler. SPI chip select pinine ihtiyaç duyar.
- I<sup>2</sup>C arbitration destekler, SPI desteklemez
- I<sup>2</sup>C clock stretching destekler, SPI desteklemez.
- I<sup>2</sup>C start-stop bitleri ile fazla yüke sahiptir.
- I<sup>2</sup>C uzun mesafe, SPI kısa mesafe için daha iyidir.

## Asenkron ve Senkron Haberleşme:

Asenkron	Senkron
Alıcı verici arasında ortak bir saat sinyali yoktur.	İletişim paylaşılan bir saat ile yapılır.
Bir sefer 1 byte gönderir.	Veriyi bloklar halinde gönderir.
Senkrona göre yavaştır.	Asenkrona göre hızlıdır.
Start stop biti nedeniyle ek yük.	Daha az ek yük.
Uzak mesafeler.	Asenkrona göre daha kısa mesafeler.
Veri senkronizasyonu için start stop biti.	Veri senkronizasyonu için ortak saat kullanımı.
Ekonomik	Pahalı
RS232, RS485	I2C, SPI

**Tablo 1.1** Asenkron ve Senkron Haberleşme

**RS232:** Veri iletimi için kullanılan seri iletişim protokolüdür.

Maksimum 15 metre civarı haberleşme gerçekleştirir.

Sinyalizasyonu dengesizdir.

Tek katlı hat yapısına sahiptir.

Simplex ya da Full Duplex modda çalışır.

**RS422:** Bu standart, RS-232C'deki kısa iletim mesafesi ve yavaş iletim hızı gibi sorunları düzeltir. "EIA-422A" olarak da adlandırılır. Sinyal hatlarının amacı ve zamanlaması tanımlanır, ancak konektörler tanımlanmaz. öncelikle D-sub 25-pin ve D-sub 9-pin konektörleri benimser. RS-422 diferansiyel sinyaller kullanır ve RS-232 dengesiz toprak referanslı sinyaller kullanır. Diferansiyel iletim, sinyal göndermek ve almak için iki kablo kullanır RS-232 ile karşılaştırıldığında, gürültüye daha iyi direnebilir ve daha uzun iletim mesafesine sahip olabilir. Daha iyi gürültü bağıřıklığı ve endüstriyel ortamlarda daha uzun iletim mesafeleri büyük bir artıdır.

**RS485:** RS-485 (EIA-485 standardı) RS-422'nin bir gelişmesidir, çünkü cihaz sayısını 10'dan 32'ye çıkarır ve yeterli sağlamak için maksimum cihaz sayısındaki elektriksel karakteristikleri tanımlar. Birden fazla cihazın yeteneğı sayesinde, bir cihaz ağı kurmak için tek bir RS-422 portu kullanabilirsiniz. Mükemmel gürültü önleme ve çoklu cihaz özellikleri endüstriyel uygulamalarda, diğeri veri toplama kontrol cihazlarında, HMI'larda veya diğeri işlemlerde PC'lere dağıtılmış bir cihaz ağı kurarken, RS-485 seri bağlantıdır.

RS-485 dezavantajı , programlamak daha zor olmasıdır, çünkü veri göndermek ve almak için aynı 2 kabloyu kullanır. Ve herhangi bir ağda, yalnızca bir düğüm veri iletebilir, diğeri düğümler yalnızca o anda alabilir. On avantajları tarafında, herhangi bir sorun ile uzun mesafe iletişimi destekler. Ayrıca, arayüz devresinin zarar görmesini zorlařtıran RS-232'den daha düşük arayüz sinyal seviyeleri kullanır.

RS-485, RS-422'nin bir üst kümesidir, bu nedenle tüm RS-422 cihazları RS-485 tarafından kontrol edilebilir. RS-485, 1200 metre telin üzerindeki seri trafik için kullanılabilir. **NOTLARIMDAN YAZI EKLENECEK ve RESİM EKLENECEK**

## RS232 ve RS485 Arasındaki Farklar:

	RS232	RS485
<b>Hat Yapısı</b>	Tek katlı	Diferansiyel
<b>Aygıt Sayısı</b>	1 alıcı, 1 gönderici	32 alıcı, 32 gönderici
<b>Çalışma Şekli</b>	Simplex ya da full duplex	Simplex ya da half duplex
<b>Max. Kablo Uzunluğu</b>	15 m	1200 m
<b>Max. Veri Hızı</b>	20 Kbits/s	10 Kbits/s
<b>Sinyalizasyon</b>	Dengesiz	Dengeli
<b>Tipi Logic Seviyeleri</b>	+5 ~ +15V	+1.5 ~ +6V
<b>Min. Alıcı Giriş Empedansı</b>	3 ~ 7 K-ohm	12 K-ohm
<b>Alıcı Duyarlılığı</b>	+3V	+200mV

**Tablo 1.2** RS232 ve RS485 Arasındaki Farklar

```
#include <Wire.h>
#include <Adafruit_BMP085.h> // Adafruit BMP085 Library
#include "I2Cdev.h" //I2C kütüphanesi
#include "MPU6050.h" //Mpu6050 kütüphanesi
#include "Wire.h"

#define Magnetometer_mX0 0x03
#define Magnetometer_mX1 0x04
#define Magnetometer_mZ0 0x05
#define Magnetometer_mZ1 0x06
#define Magnetometer_mY0 0x07
#define Magnetometer_mY1 0x08

#define Magnetometer 0x1E //I2C 7bit address of HMC5883

Adafruit_BMP085 bmp;
MPU6050 accelgyro; // Mpu6050 sensör tanımlama

int16_t ax, ay, az; //ivme tanımlama
int16_t gx, gy, gz; //gyro tanımlama
int mX0, mX1, mX_out;
int mY0, mY1, mY_out;
int mZ0, mZ1, mZ_out;

float heading, headingDegrees, headingFiltered, declination;
float Xm,Ym,Zm;
void setup(){

  Wire.begin();
  delay(100);

  Serial.begin(9600);
  accelgyro.initialize();
```

```
  if (!bmp.begin()) {
    while (true) {}
  }

  Wire.beginTransmission(Magnetometer);
  Wire.write(0x02); // Select mode register
  Wire.write(0x00); // Continuous measurement mode
  Wire.endTransmission();

}

void loop(){
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  //---- X-Axis
  Wire.beginTransmission(Magnetometer); // transmit to
device
  Wire.write(Magnetometer_mX1);
  Wire.endTransmission();
  Wire.requestFrom(Magnetometer,1);
  if(Wire.available()<=1)
  {
    mX0 = Wire.read();
  }
}
```



```
Wire.beginTransaction(Magnetometer); // transmit to device
Wire.write(Magnetometer_mX0);
Wire.endTransmission();
Wire.requestFrom(Magnetometer,1);
if(Wire.available()<=1)
{
    mX1 = Wire.read();
}
//----- Y-Axis
Wire.beginTransaction(Magnetometer); // transmit to device
Wire.write(Magnetometer_mY1);
Wire.endTransmission();
Wire.requestFrom(Magnetometer,1);
if(Wire.available()<=1)
{
    mY0 = Wire.read();
}
Wire.beginTransaction(Magnetometer); // transmit to device
Wire.write(Magnetometer_mY0);
Wire.endTransmission();
Wire.requestFrom(Magnetometer,1);
if(Wire.available()<=1)
{
    mY1 = Wire.read();
}
```

```
//----- Z-Axis
Wire.beginTransaction(Magnetometer); // transmit to device
Wire.write(Magnetometer_mZ1);
Wire.endTransmission();
Wire.requestFrom(Magnetometer,1);
if(Wire.available()<=1)
{
    mZ0 = Wire.read();
}
Wire.beginTransaction(Magnetometer); // transmit to device
Wire.write(Magnetometer_mZ0);
Wire.endTransmission();
Wire.requestFrom(Magnetometer,1);
if(Wire.available()<=1)
{
    mZ1 = Wire.read();
}

//----- X-Axis
mX1=mX1<<8;
mX_out =mX0+mX1; // Raw data
// From the datasheet: 0.92 mG/digit
Xm = mX_out*0.00092; // Gauss unit
/* Earth magnetic field ranges from 0.25 to 0.65 Gauss,
so these are the values that we need to get approximately.
//----- Y-Axis
mY1=mY1<<8;
mY_out =mY0+mY1;
Ym = mY_out*0.00092;
```

```
//---- Z-Axis
mZ1=mZ1<<8;
mZ_out =mZ0+mZ1;
Zm = mZ_out*0.00092;
// =====
//Calculating Heading
heading = atan2(Ym, Xm);

// Correcting the heading with the declination angle
depending on your location
// You can find your declination angle at:
https://www.ngdc.noaa.gov/geomag-web/
// At my location it's 4.2 degrees => 0.073 rad
declination = 0.073;
heading += declination;

// Correcting when signs are reversed
if(heading <0) heading += 2*PI;
// Correcting due to the addition of the declination
angle
if(heading > 2*PI)heading -= 2*PI;
headingDegrees = heading * 180/PI; // The heading in
Degrees unit
// Smoothing the output angle / Low pass filter
```

```
headingFiltered = headingFiltered*0.85 + headingDegrees*0.15;
//Sending the heading value through the Serial Port to
Processing IDE
//Serial.println(headingFiltered);

delay(50);
Serial.print(bmp.readTemperature());Serial.print("/");
Serial.print(bmp.readPressure());Serial.print("/");
Serial.print(bmp.readAltitude());Serial.print("/");
Serial.print(ax); Serial.print("/");
Serial.print(ay); Serial.print("/");
Serial.print(az); Serial.print("/");
Serial.print(gx); Serial.print("/");
Serial.print(gy); Serial.print("/");
Serial.print(gz); Serial.print("/");
Serial.println(headingFiltered);

delay(1000);
}
```

Elektronik donanım kısmında seçilmiş olan elektronik komponentlerden verileri alabilmemiz için tanımlamış olduğumuz sensör kütüphanelerinden istediğimiz fonksiyonları çağırarak kullanmamızı sağlamaktadır. Bu yazılım dili ise C tabanlı yazılım dilidir.

25.90/91187/880.38/-840/-508/20632/-252/223/-30/46.43  
25.90/91190/880.47/-816/-404/20544/-236/235/-31/85.87

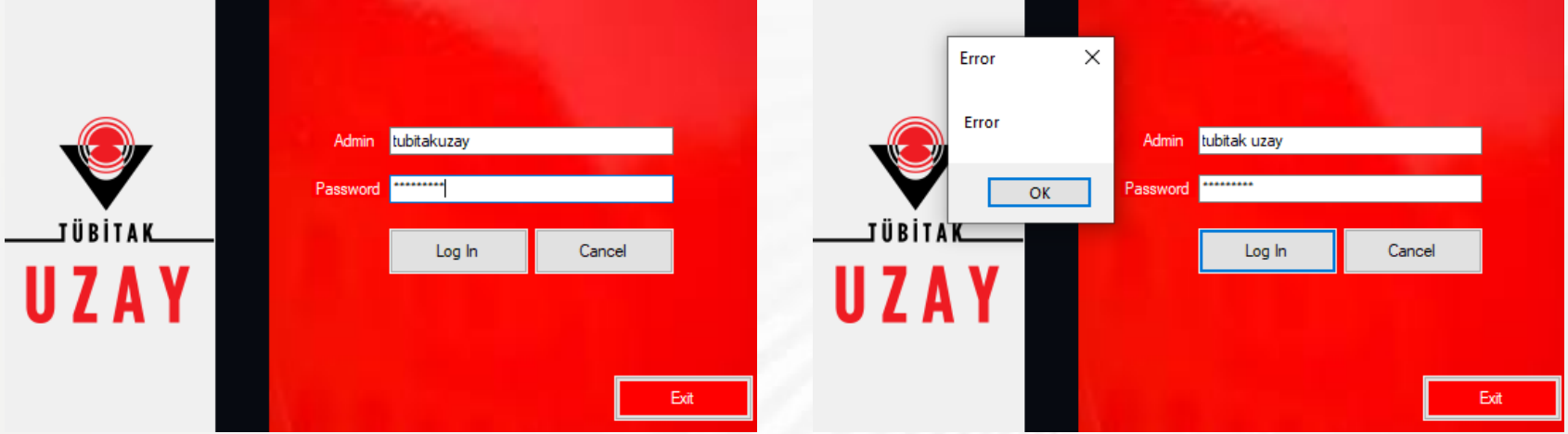
☒ Autoscroll ☐ Show timestamp

Both NL & CR

9600 baud

Clear output

**Video 1.** Aviyonik Yazılım Tasarımı Verilerin Geliş Formatı



**Resim 1.5** Kontrol Arayüzü Giriş Ekranı

Telemetri kontrol arayüzüne erişim sağlayabilmek için şifreli giriş yöntemi ekledim. Burada aktif olarak kayıtlı kullanıcı adı ve şifresi;

Kullanıcı Adı: tubitakuzay

Şifre: 123456789

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace interface_code
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void login_Click(object sender, EventArgs e)
        {
            /*
            telemetry form = new telemetry();
            form.Show();
            this.Hide();
            */


            if (textBox1.Text=="tubitakuzay" &&
            textBox2.Text=="123456789")
            {
                telemetry form = new telemetry();
                form.Show();
                this.Hide();
            }
        }

        else
        {
            MessageBox.Show("Error", "Error");
            textBox1.Clear();
            textBox2.Clear();
        }
    }

    private void cancel_Click(object sender, EventArgs e)
    {
        textBox1.Clear();
        textBox2.Text = "";
    }

    private void exit_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }
}
```

# Kontrol Arayüz Yazılım Alt Tasarımı



Setting

Connect COM3

Disconnect 4800

Exit Disconnect

Data Group

	Packet No	Time	Date	Temperature	Pressure	Altitude	aX	aY	aZ	gX	gY	gZ	Magno
*													

< >

Telemetry

Temperature

Pressure

Altitude

aX

aY

aZ

gX

gY

gZ

Magnetometer

Control Button

Excell Saved

Excell Clear

Temperature

Series1

Pressure

Series1

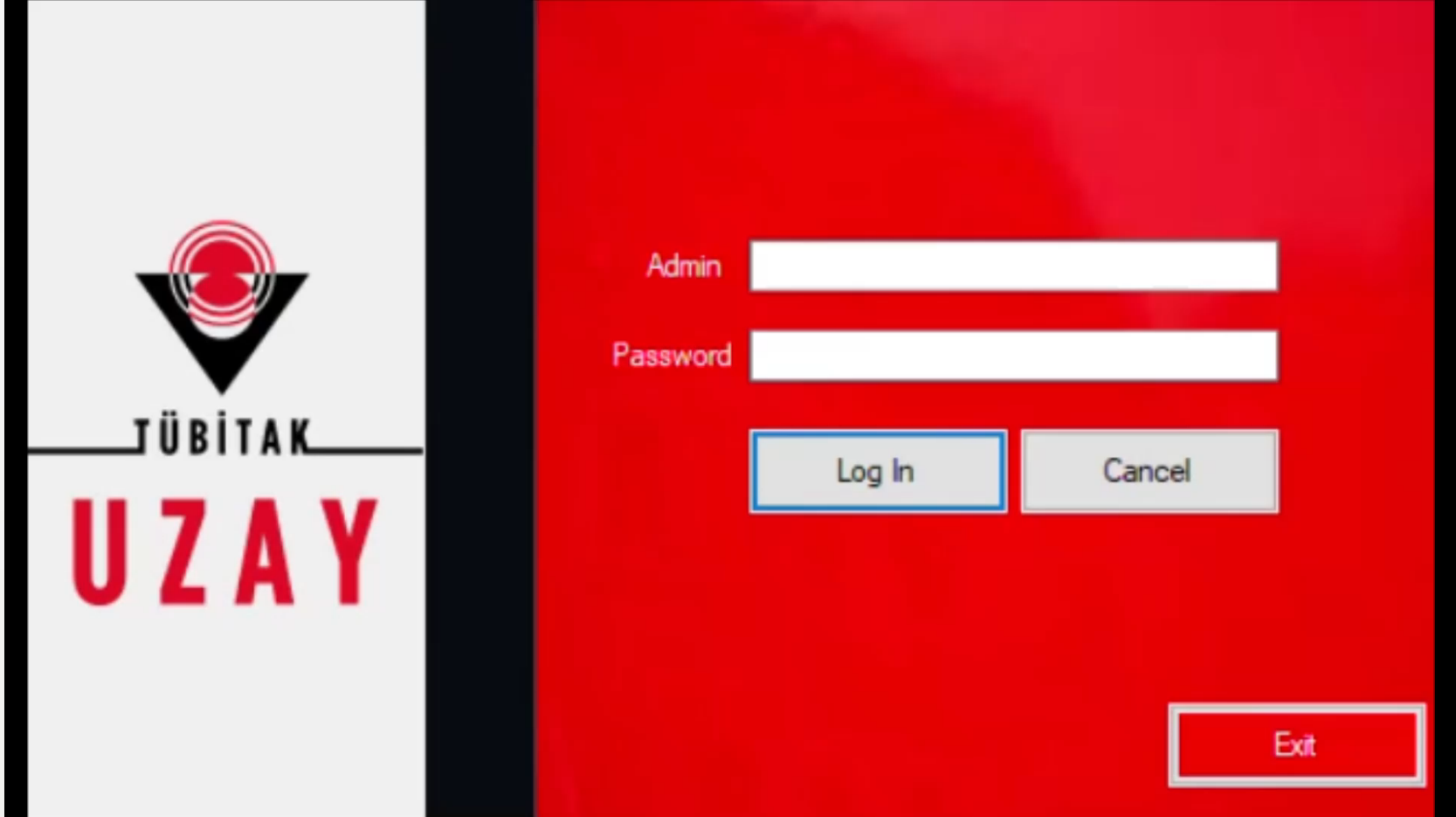
Altitude

Series1

Magnetometer

Series1

Resim 1.6 Kontrol Arayüzü Telemetri Alanı



Admin


Password

Log In Cancel

Exit

Video 1.1 Kontrol Arayüzü Sensör Test Videosu (MPU6050)

# Kontrol Arayüz Yazılım Alt Tasarımı



Setting

Connect COM3

Disconnect 4800

Exit Disconnect

Data Group

Packet No	Time	Date	Temperature	Pressure	Altitude	aX	aY	aZ	gX	gY	gZ	Magn
*												

Telemetry

Temperature

Pressure

Altitude

aX

aY

aZ

gX

gY

gZ

Magnetometer

Control Button

Excell Saved

Excell Clear

Temperature

Series1

Pressure

Series1

Altitude

Series1

Magnetometer

Series1



Setting

Connect COM3

Disconnect 4800

Exit label1 Disconnect

Resim 1.7 Baudrate Seçim Alanı

Telemetri verilerin serial porttan okunması için UART ile birlikte bağlı olan portun bulunması ve ayarlanan baudrate ile birlikte verilerin alınmasını sağlamak için oluşturulmuş olan kısımdır.

Data Group

Satir No	Time	Date	Pressure	Height	Temperature	aX	aY	aZ	gX	gY	gZ	Headin
*												

Resim 1.8 Kontrol Arayüzü Telemetri Alanı

Control Button

Excell Saved

Excell Clear

Resim 1.9 Veri Export Alanı

Telemetri verilerin gelmesi sonucunda verilerin sıralı formatta kaydedilmesi için Excel yöntemine başvurdum. Burada Excel formatına gelen veriler anlık olarak eklenmekte ve kaydedilmektedir.

M10

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Satir No	Time	Date	Pressure	Height	Temperat	aX	aY	aZ	gX	gY	gZ	Heading		
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														

Resim 2. Export Sonuç Alanı

# Kontrol Arayüz Yazılım Alt Tasarımı

Packet No	Time	Date	Temperat	Pressure	Altitude	aX	aY	aZ	gX	gY	gZ	Magnome
1	15:09:32	9/8/2023	26	91190	880.65	-776	-416	20568	-248	227	-53	309.39
2	15:09:33	9/8/2023	26	91193	880.2	-772	-436	20592	-243	230	-75	309.41
3	15:09:34	9/8/2023	26	91189	880.2	-728	-492	20444	-239	197	-34	309.41
4	15:09:35	9/8/2023	26	91182	879.92	-832	-440	20392	-251	235	-54	309.44
5	15:09:36	9/8/2023	26	91188	880.11	-796	-404	20524	-246	246	-53	309.41
6	15:09:37	9/8/2023	26	91188	880.47	-824	-376	20576	-235	210	-38	309.41
7	15:09:38	9/8/2023	26	91186	880.29	-796	-444	20552	-240	229	-22	309.38
8	15:09:39	9/8/2023	26	91188	880.2	-848	-444	20604	-262	225	-51	309.31
9	15:09:41	9/8/2023	26	91188	880.11	-792	-384	20492	-247	222	-55	309.28
10	15:09:42	9/8/2023	26	91192	880.2	-840	-484	20504	-253	220	-33	309.39
11	15:09:43	9/8/2023	26	91185	880.02	-796	-396	20564	-280	240	-51	309.37
12	15:09:44	9/8/2023	26	91188	880.29	-768	-540	20572	-237	234	-45	309.45
13	15:09:45	9/8/2023	26	91187	880.11	-856	-496	20528	-237	217	-36	309.46
14	15:09:46	9/8/2023	26	91195	880.29	-840	-544	20576	-268	223	-52	309.42
15	15:09:47	9/8/2023	26	91188	880.29	-824	-512	20560	-256	224	-50	309.45
16	15:09:49	9/8/2023	26	91190	880.2	-800	-540	20512	-263	240	-63	309.44
17	15:09:50	9/8/2023	26	91188	880.29	-800	-424	20416	-261	230	-36	309.55
18	15:09:51	9/8/2023	26	91187	880.47	-848	-504	20620	-226	216	-32	309.55
19	15:09:52	9/8/2023	26	91186	880.74	-804	-532	20612	-265	224	-54	309.52
20	15:09:53	9/8/2023	26	91186	880.56	-852	-408	20480	-238	213	-26	309.49
21	15:09:54	9/8/2023	26	91188	880.2	-900	-424	20560	-259	224	-55	309.49
22	15:09:55	9/8/2023	26	91188	880.11	-784	-444	20452	-249	213	-37	309.46
23	15:09:57	9/8/2023	26	91184	880.11	-728	-380	20508	-233	204	-43	309.46
24	15:09:58	9/8/2023	26	91183	880.2	-728	-508	20548	-233	220	-41	309.51
25	15:09:59	9/8/2023	26	91190	880.65	-732	-452	20476	-284	246	-50	309.59
26	15:10:00	9/8/2023	26	91186	880.2	-872	-452	20504	-249	224	-40	309.61
27	15:10:01	9/8/2023	26	91187	880.29	-720	-352	20592	-264	237	-23	309.62
28	15:10:02	9/8/2023	26	91182	880.11	-796	-508	20584	-260	213	-70	309.67
29	15:10:03	9/8/2023	26	91188	880.11	-784	-424	20444	-261	257	-41	309.76
30	15:10:05	9/8/2023	26	91186	880.56	-744	-392	20516	-243	229	-56	309.70
31	15:10:06	9/8/2023	26	91188	880.38	-812	-504	20460	-257	232	-32	309.70
32	15:10:07	9/8/2023	26	91184	880.74	-788	-496	20680	-238	226	-19	309.72
33	15:10:08	9/8/2023	26	91188	880.74	-728	-456	20620	-263	231	-36	309.69
34	15:10:09	9/8/2023	26	91191	880.2	-752	-476	20520	-242	251	-75	309.79

Telemetri verilerin gelmesi sonucunda veriler sonuçlarım yandaki tabloda gösterilmektedir.

A3	D6												
D3	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Packet No	Time	Date	Temperature	Pressure	Altitude	aX	aY	aZ	gX	gY	gZ	Magnetom
2	14:38:55	9/8/2023	26.4	91185	881.1	-744	-336	20524	-252	232	-22	309.47	
3	14:38:57	9/8/2023	26.4	91183	880.74	-608	-332	20600	-245	219	-43	309.46	
4	14:38:59	9/8/2023	26.4	91188	880.74	-756	-348	20504	-250	222	-61	129.90	
5	14:39:01	9/8/2023	26.4	91188	879.92	-692	-352	20556	-279	236	-40	160.75	
6	14:39:03	9/8/2023	26.4	91181	880.56	-772	-292	20560	-255	211	-15	186.86	
7	14:39:05	9/8/2023	26.4	91182	880.2	-772	-344	20512	-235	209	-50	209.08	
8	14:39:07	9/8/2023	26.4	91186	880.38	-752	-384	20540	-260	239	-42	228.00	
9	14:39:09	9/8/2023	26.4	91186	879.92	-660	-284	20588	-225	233	-44	244.18	
10	14:39:11	9/8/2023	26.4	91189	880.65	-696	-364	20568	-249	205	-37	257.87	
11	14:39:13	9/8/2023	26.4	91185	880.29	-792	-400	20568	-255	233	-52	269.55	
12	14:39:15	9/8/2023	26.4	91185	880.2	-792	-228	20552	-253	208	-41	279.54	
13	14:39:17	9/8/2023	26.3	91186	879.92	-760	-304	20572	-263	236	-49	287.97	
14	14:39:19	9/8/2023	26.3	91185	880.2	-756	-336	20496	-248	228	-44	295.12	
15	14:39:21	9/8/2023	26.3	91189	879.92	-768	-336	20464	-252	217	-19	301.24	
16	14:39:23	9/8/2023	26.3	91186	879.65	-676	-368	20428	-238	211	-31	306.45	
17	14:39:25	9/8/2023	26.3	91190	880.29	-792	-320	20532	-261	234	-49	310.91	
18	14:39:27	9/8/2023	26.4	91187	879.75	-800	-304	20536	-236	234	-46	314.69	
19	14:39:29	9/8/2023	26.3	91189	879.83	-756	-276	20606	-238	211	-31	317.82	
20	14:39:31	9/8/2023	26.4	91184	880.65	-828	-396	20548	-258	240	-43	320.47	
21	14:39:33	9/8/2023	26.3	91185	879.92	-716	-360	20556	-259	253	-32	322.86	
22	14:39:35	9/8/2023	26.3	91183	880.29	-756	-380	20488	-258	240	-31	324.82	
23	14:39:37	9/8/2023	26.3	91186	879.92	-776	-220	20516	-249	254	-42	326.53	
24	14:39:40	9/8/2023	26.4	91188	879.83	-816	-404	20696	-233	241	-17	327.87	
25	14:39:42	9/8/2023	26.4	91180	881.01	-732	-288	20600	-240	228	-35	329.03	
26	14:39:44	9/8/2023	26.3	91190	880.29	-832	-504	20616	-250	181	-40	329.99	
27	14:39:46	9/8/2023	26.3	91189	880.74	-2968	-10944	21476	11522	2340	1481	331.65	
28	14:39:48	9/8/2023	26.4	91189	880.11	-11132	72	16108	-1647	3639	-2755	328.79	
29	14:39:51	9/8/2023	26.3	91188	880.2	-15760	-124	6832	971	-2955	-448	325.72	
30	14:39:53	9/8/2023	26.3	91189	880.2	-708	-1340	19756	384	112	-44	327.52	
31	14:39:55	9/8/2023	26.3	91188	880.2	-736	-264	20508	-232	203	-32	329.12	
32	14:39:57	9/8/2023	26.4	91191	879.83	-728	-316	20544	-243	207	-35	330.50	
33	14:39:59	9/8/2023	26.4	91187	880.11	-828	-320	20536	-239	220	-42	331.65	
34	14:40:01	9/8/2023	26.3	91189	880.02	-744	-460	20512	-247	251	-36	332.51	
35	14:40:03	9/8/2023	26.3	91187	879.65	-764	-356	20596	-241	231	-55	333.23	
36	14:40:05	9/8/2023	26.3	91185	879.75	-788	-344	20424	-272	208	-37	333.86	
37	14:40:08	9/8/2023	26.4	91192	879.56	-816	-360	20564	-253	230	-56	334.43	
38	14:40:10	9/8/2023	26.3	91186	880.2	-704	-232	20572	-245	234	-48	334.86	

Tablo 1.3 Kontrol Arayüzü Data Sonuçları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using System.IO;
using System.IO.Ports;
using System.Drawing.Drawing2D;
using System.Runtime.ConstrainedExecution;
```

```
namespace interface_code
{
    public partial class telemetry : Form
    {
        DateTime yeni = DateTime.Now;

        long maximum = 30;
        long minimum = 0;

        int zaman = 0;
        int satir = 1;
        int sutun = 1;
        int satirNo = 1;
        int k = 0;
```

```
        string[] portlar = SerialPort.GetPortNames();
        public telemetry()
        {
            InitializeComponent();

            private void telemetry_FormClosing(object sender,
            FormClosingEventArgs e)
            {
                Application.Exit();
            }

            private void telemetry_Load(object sender, EventArgs e)
            {
                foreach (string port in portlar) // Veriler string
                geldi.
                {
                    comboBox1.Items.Add(port);
                    comboBox1.SelectedIndex = 0;

                    comboBox2.Items.Add("2400");
                    comboBox2.Items.Add("4800");
                    comboBox2.Items.Add("9600");
                    comboBox2.Items.Add("14400");
                    comboBox2.Items.Add("57600");
                    comboBox2.Items.Add("115200");
                    comboBox2.SelectedIndex = 1;
                    label2.Text = "Disconnect";
                }
            }
        }
    }
}
```

```
Task ExcellKayit()
{
    return Task.Run(() =>
    {
        Microsoft.Office.Interop.Excel.Application excel =
new Microsoft.Office.Interop.Excel.Application();

        excel.Visible = true;

        Microsoft.Office.Interop.Excel.Workbook workbook =
excel.Workbooks.Add(System.Reflection.Missing.Value);

        Microsoft.Office.Interop.Excel.Worksheet sheet1 =
(Microsoft.Office.Interop.Excel.Worksheet)workbook.Sheets[1];

        int StartCol = 1;
        int StartRow = 1;

        for (int j = 0; j < dataGridView1.Columns.Count;
j++)
        {
            Microsoft.Office.Interop.Excel.Range myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow,
StartCol + j];

            myRange.Value2 =
dataGridView1.Columns[j].HeaderText;

        }

        StartRow++;
    }
}
```

```
for (int i = 0; i < dataGridView1.Rows.Count; i++)
{
    for (int j = 0; j < dataGridView1.Columns.Count;
j++)
    {
        try
        {
            Microsoft.Office.Interop.Excel.Range
myRange =
(Microsoft.Office.Interop.Excel.Range)sheet1.Cells[StartRow + i,
StartCol + j];

            myRange.Value2 = dataGridView1[j,
i].Value == null ? "" : dataGridView1[j, i].Value;

        }
        catch
        {

        }

    }

}

});
```

```
private void connect_Click(object sender, EventArgs e)
{
    connect.Enabled = false;
    disconnect.Enabled = true;
    timer1.Start();
    if (serialPort1.IsOpen == false)
    {
        if (comboBox1.Text == "")
            return;
        serialPort1.PortName = comboBox1.Text;
        serialPort1.BaudRate =
Convert.ToInt16(comboBox2.Text);

        try
        {
            serialPort1.Open();
            label2.Text = "Connect";
        }
        catch (Exception syntax)
        {
            MessageBox.Show("Error:" + syntax.Message);
        }
        // serialPort1.DiscardInBuffer();
    }

    else
    {
        label2.Text = "Connected";
    }
}

private void serialPort1_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
}
```

```
private void timer1_Tick(object sender, EventArgs e)
{
    // X eksenini değer etiketlerini görünür yapma
    chart1.ChartAreas[0].AxisX.LabelStyle.Enabled = true;
    // X eksenini değer rengini değiştirme
    chart1.ChartAreas[0].AxisX.LabelStyle.ForeColor =
Color.Red;
    //////////////////////////////////////
    //////////////////////////////////////Kontrol Edilecek.

    // Y eksenini değer etiketlerini görünür yapma
    chart1.ChartAreas[0].AxisY.LabelStyle.Enabled = true;
    // Y eksenini değer rengini değiştirme
    chart1.ChartAreas[0].AxisY.LabelStyle.ForeColor =
Color.Red;

    // X eksenini değer etiketlerini görünür yapma
    chart2.ChartAreas[0].AxisX.LabelStyle.Enabled = true;
    // X eksenini değer rengini değiştirme
    chart2.ChartAreas[0].AxisX.LabelStyle.ForeColor =
Color.Red;
    //////////////////////////////////////
    //////////////////////////////////////Kontrol Edilecek.

    // Y eksenini değer etiketlerini görünür yapma
    chart2.ChartAreas[0].AxisY.LabelStyle.Enabled = true;
    // Y eksenini değer rengini değiştirme
    chart2.ChartAreas[0].AxisY.LabelStyle.ForeColor =
Color.Red;
```

```
// X eksenı değır etiketlerini görünır yapma
chart3.ChartAreas[0].AxisX.LabelStyle.Enabled = true;
// X eksenı değır rengini değıştirme
chart3.ChartAreas[0].AxisX.LabelStyle.ForeColor =
Color.Red;
////////////////////////////////////
/Kontrol Edilecek.
```

```
// Y eksenı değır etiketlerini görünır yapma
chart3.ChartAreas[0].AxisY.LabelStyle.Enabled = true;
// Y eksenı değır rengini değıştirme
chart3.ChartAreas[0].AxisY.LabelStyle.ForeColor =
Color.Red;
```

```
// X eksenı değır etiketlerini görünır yapma
chart4.ChartAreas[0].AxisX.LabelStyle.Enabled = true;
// X eksenı değır rengini değıştirme
chart4.ChartAreas[0].AxisX.LabelStyle.ForeColor =
Color.Red;
////////////////////////////////////
////////////////////////////////////Kontrol Edilecek.
```

```
// Y eksenı değır etiketlerini görünır yapma
chart4.ChartAreas[0].AxisY.LabelStyle.Enabled = true;
// Y eksenı değır rengini değıştirme
chart4.ChartAreas[0].AxisY.LabelStyle.ForeColor =
Color.Red;
```

```
//Chart1
chart1.ChartAreas[0].AxisX.Minimum = minimum;
chart1.ChartAreas[0].AxisX.Maximum = maximum;

chart1.ChartAreas[0].AxisY.Minimum = 0;
chart1.ChartAreas[0].AxisY.Maximum = 100;

chart1.ChartAreas[0].AxisX.ScaleView.Zoom(minimum,
maximum);
```

```
//Chart2
chart2.ChartAreas[0].AxisX.Minimum = minimum;
chart2.ChartAreas[0].AxisX.Maximum = maximum;

chart2.ChartAreas[0].AxisY.Minimum = 0;
chart2.ChartAreas[0].AxisY.Maximum = 100;

chart2.ChartAreas[0].AxisX.ScaleView.Zoom(minimum,
maximum);
```

```
//Chart3
chart3.ChartAreas[0].AxisX.Minimum = minimum;
chart3.ChartAreas[0].AxisX.Maximum = maximum;

chart3.ChartAreas[0].AxisY.Minimum = 0;
chart3.ChartAreas[0].AxisY.Maximum = 100;

chart3.ChartAreas[0].AxisX.ScaleView.Zoom(minimum,
maximum);
```

```
//Chart4
chart4.ChartAreas[0].AxisX.Minimum = minimum;
chart4.ChartAreas[0].AxisX.Maximum = maximum;

chart4.ChartAreas[0].AxisY.Minimum = 0;
chart4.ChartAreas[0].AxisY.Maximum = 100;

chart4.ChartAreas[0].AxisX.ScaleView.Zoom(minimum,
maximum);
```



```
try
{
    string data = serialPort1.ReadLine();
    string[] series = data.Split('/');
    //label1.Text = data + "";

    textBox1.Text = series[0];
    textBox2.Text = series[1];
    textBox3.Text = series[2];
    textBox4.Text = series[3];
    textBox5.Text = series[4];
    textBox6.Text = series[5]; /*
    textBox7.Text = series[6];
    textBox8.Text = series[7];
    textBox9.Text = series[8];
    textBox10.Text = series[9];
    textBox11.Text = series[10];
    textBox12.Text = series[11]; */

    this.chart1.Series[0].Points.AddXY((minimum +
maximum) / 2, series[0]);
    this.chart2.Series[0].Points.AddXY((minimum +
maximum) / 2, series[1]);
    this.chart3.Series[0].Points.AddXY((minimum +
maximum) / 2, series[2]);
    this.chart4.Series[0].Points.AddXY((minimum +
maximum) / 2, series[3]);

    maximum++;
    minimum++;
}
```

```
dataGridView1.Invoke(new Action(() =>
{
    satir = dataGridView1.Rows.Add(data);

    dataGridView1.Rows[satir].Cells[0].Value = satirNo;

    dataGridView1.Rows[satir].Cells[3].Value =
series[0];
    dataGridView1.Rows[satir].Cells[4].Value =
series[1];
    dataGridView1.Rows[satir].Cells[5].Value =
series[2];
    dataGridView1.Rows[satir].Cells[6].Value =
series[3];
    dataGridView1.Rows[satir].Cells[7].Value =
series[4];
    dataGridView1.Rows[satir].Cells[8].Value =
series[5];
    dataGridView1.Rows[satir].Cells[9].Value =
series[0];
    dataGridView1.Rows[satir].Cells[10].Value =
series[1];
    dataGridView1.Rows[satir].Cells[11].Value =
series[2];
    dataGridView1.Rows[satir].Cells[12].Value =
series[3];

    dataGridView1.Rows[satir].Cells[1].Value =
DateTime.Now.ToLongTimeString(); // EN BAŞA STRİNG OLARAK
EKLEDİĞİMİZ KOD KISMINDA SÜRE İLERLEMEDİĞİ İÇİN UZUN ZAMAN ZARFI
OLARAK DATATİME İ BURAYA EKLEDİM
    dataGridView1.Rows[satir].Cells[2].Value =
yeni.ToShortDateString();

    satir++;
    satirNo++;

}));
```

```
/*Thread is process with background because deleting data.*/
    System.Threading.Thread.Sleep(1000);

//System.Threading.Thread.Sleep(1500);////////////////////
////////////////////Buraya bakılacak
    Application.DoEvents();

    serialPort1.DiscardInBuffer();
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
    timer1.Stop();
}

}

private void disconnect_Click(object sender, EventArgs e)
{
    disconnect.Enabled = false;
    connect.Enabled = true;

    timer1.Stop();

    System.Threading.Thread.Sleep(1000);
    //System.Threading.Thread.Sleep(1500);
    Application.DoEvents();

    serialPort1.DiscardInBuffer();

    if (serialPort1.IsOpen == true)
    {
        serialPort1.Close();
        label2.Text = "Disconnect";
    }
}
```

```
private void telemetry_FormClosed(object sender,
FormClosedEventArgs e)
{
    if (serialPort1.IsOpen == true)
    {
        serialPort1.Close();
    }
}

private void exit_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true)
    {
        serialPort1.Close();
    }

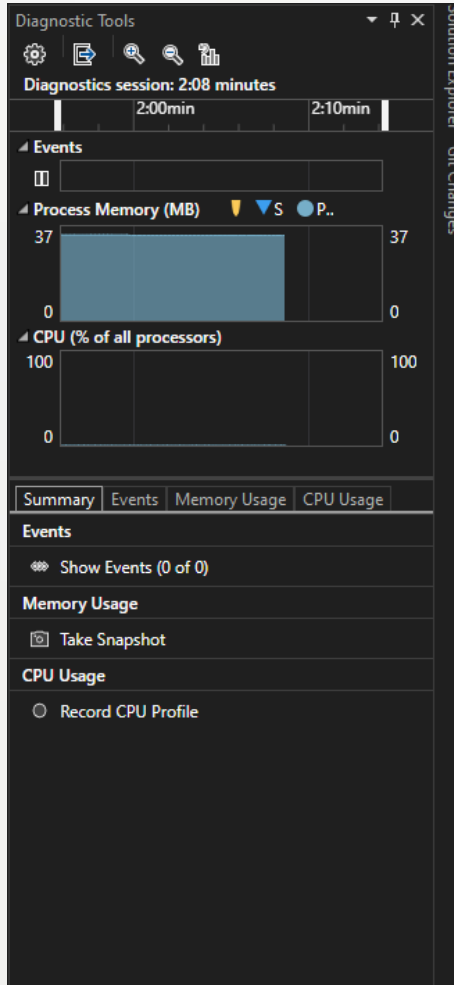
    Application.Exit();
}

////////////////////
////////////////////
private async void excellsaved_Click(object sender,
EventArgs e)
{
    await ExcellKayit();
}

////////////////////
////////////////////
private void excell_clear_Click(object sender, EventArgs e)
{
    dataGridView1.Rows.Clear();
}

}
```





Oluşturmuş olduğum kontrol arayüzü kodunu özgün bir şekilde geliştirdim.

Burada sensörlerden gelen telemetri verilerini serial port üzerinden okunması için verileri, split edebileceğim formatta string bir ifade şeklinde gönderdim. Bunun sonucunda veriler gelirken, her veri arasında '/' şeklinde ifade olmasını C yazılımı tarafında belirledim. Verileri serial port üzerinde okurken '/' bu işaretin okunması durumunda verileri split ederek C# tarafında ayırdım. Bu durumda verilerim düzgün gelmesini sağladım. Ayrıca veri kaçırma durumumda oluşmamaktadır.

Arayüz üzerinde birçok veri aynı anda belirlediğim Hz (1Hz=1 saniye) üzerinden geldiği için arayüz donmakta ve işlevsiz hale gelmektedir. Bu durumu ortadan kaldırmak ve veri güvenliğini korumak amacıyla verileri Excell üzerine aktarmasını ve daha sonra oluşturduğum thread (işlem parçacığı) ile bu verileri 1500 milisaniyede silmek istedim. Bunu sonucunda kodda saklanan verilerin bufferı şişmedi ve arayüz fonksiyonel çalışmasına devam etti.

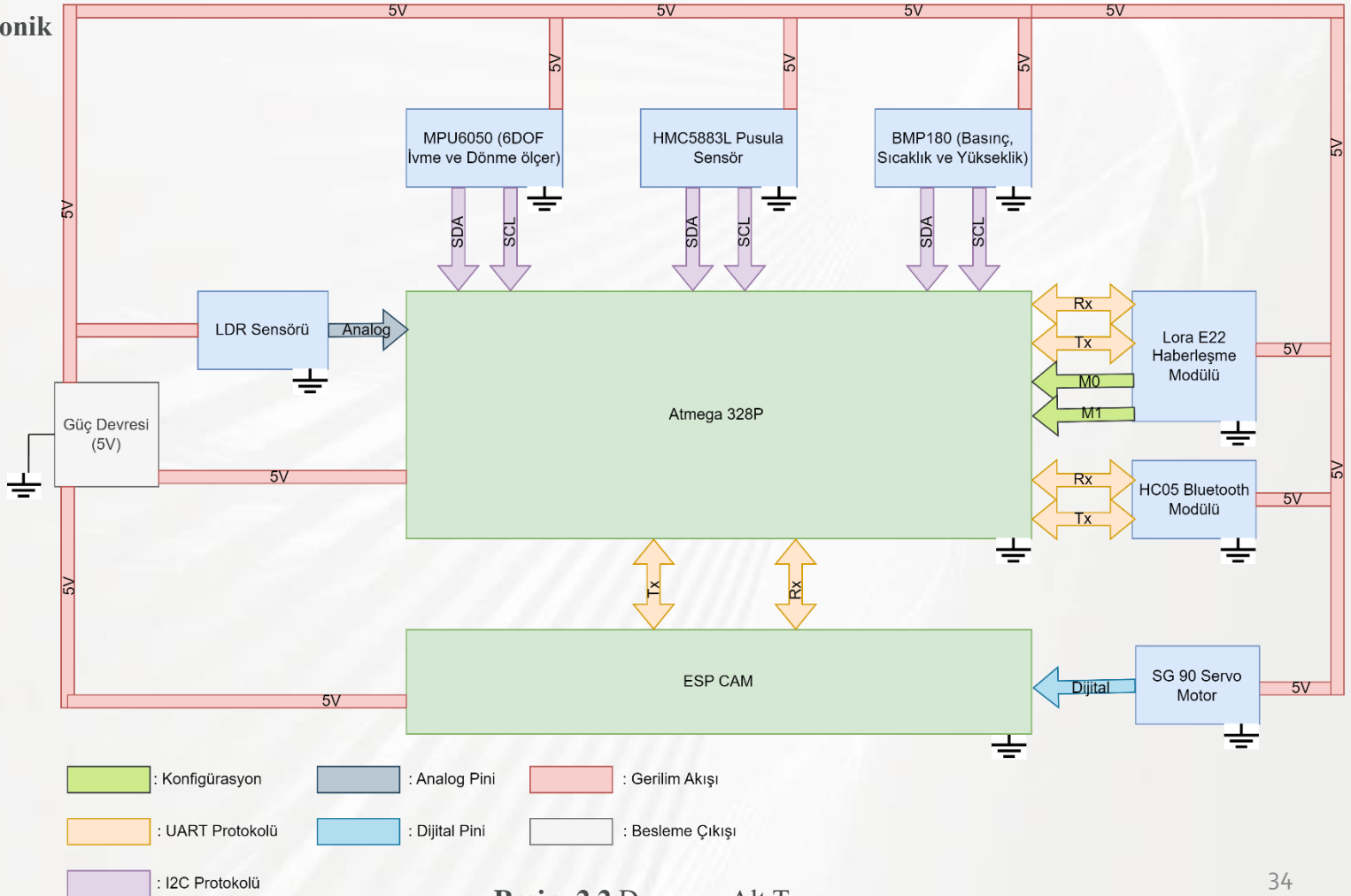
```
/*Thread is process with background because deleting data.*/  
System.Threading.Thread.Sleep(1500);  
//System.Threading.Thread.Sleep(1000); Application.DoEvents();
```

Bunun sonucuna da yandaki resimde verilen resimde işlem hafızasını incelediğimizde çok az bir miktarda (37 mb) bellekte çalıştığı gözlemlenmiştir.

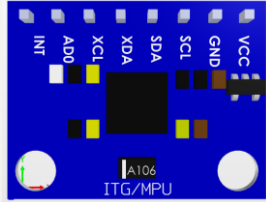
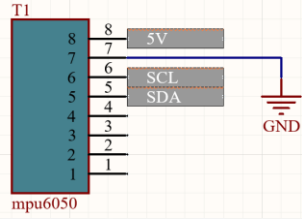
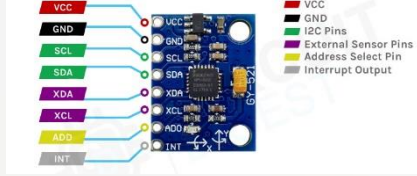
## Kullanılan Elektronik

### Ürünler:

1. MPU6050
2. BMP085
3. HMC5883L
4. ESP CAM
5. Arduino Nano
6. LDR



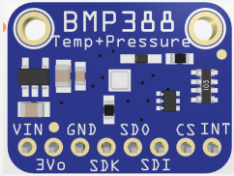
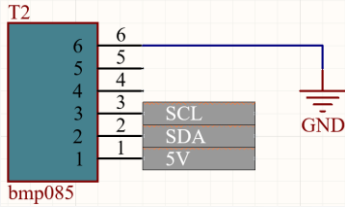
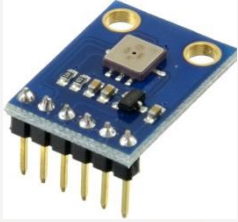
Resim 2.2 Donanım Alt Tasarımı



Resim 2.3 MPU6050 Sensör

Özellik	Değer
İletişim Arabirimi	I2C / TWI
Çalışma Gerilimi	3.3V (Volt)
İletişim Gerilimi Seviyeleri	3.3V (I2C)
Gyroskop Hassasiyeti	$\pm 250$ , $\pm 500$ , $\pm 1000$ , $\pm 2000$ dps
Hızlandırmaölçer Hassasiyeti	$\pm 2g$ , $\pm 4g$ , $\pm 8g$ , $\pm 16g$
Örnekleme Hızı	8 kHz (Gyroskop), 1 kHz (Hızlandırmaölçer)
Sıcaklık Duyarlılığı	$\pm 0.5^{\circ}\text{C}$
İletişim Protokolleri	I2C (TWI), DMP (Digital Motion Processor) ile veri işleme özelliği
Çalışma Sıcaklık Aralığı	$-40^{\circ}\text{C}$ ila $+85^{\circ}\text{C}$
Boyutlar	4 mm x 4 mm
Dijital Çıkış Veri Aralığı	16-bit
Power Management	Düşük güç tüketimi için enerji tasarrufu modları
Interrupt Pin	Programlanabilir interrupt pinleri
DMP (Digital Motion Processor)	6 eksenlik hareket takibi ve işleme yeteneği

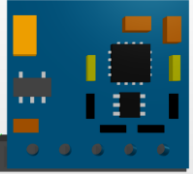
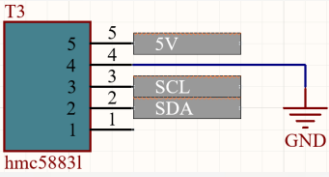
Tablo 1.4 MPU6050 Sensör Özellikleri



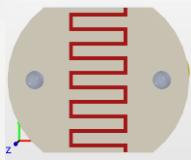
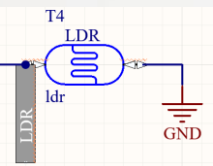
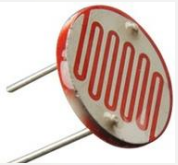
Resim 2.4 BMP085 Sensörü

Özellik	Değer
İletişim Arabirimi	I2C (TWI) veya SPI
Çalışma Gerilimi	3.3V (Volt)
İletişim Gerilimi Seviyeleri	3.3V (I2C) veya 3.3V/5V (SPI)
Basınç Aralığı	300 hPa ila 1100 hPa
Sıcaklık Aralığı	-40°C ila +85°C
Basınç Hassasiyeti	0.03 hPa (yaklaşık)
Sıcaklık Hassasiyeti	0.1°C (yaklaşık)
Ölçüm Çözünürlüğü	1 hPa (Basınç), 0.1°C (Sıcaklık)
Örnekleme Hızı	1 Hz (varsayılan)
Power Management	Düşük güç tüketimi için enerji tasarrufu modları
Boyutlar	Genellikle 5 mm x 5 mm
İletişim Arabirimi	I2C (TWI) veya SPI

Tablo 1.5 BMP085 Sensör Özellikleri



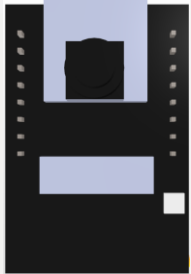
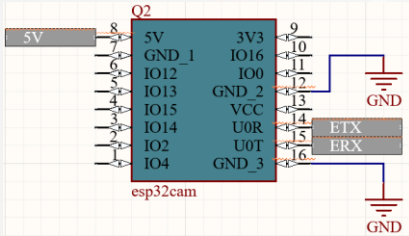
Resim 2.5 HMC5883L Sensörü



Resim 2.6 HMC5883L Sensörü

Özellik	Değer
İletişim Arabirimi	I2C (TWI)
Çalışma Gerilimi	3.3V (Volt)
İletişim Gerilimi Seviyeleri	3.3V (I2C)
Ölçüm Aralığı	$\pm 1.3 - \pm 8.1$ gauss (G)
Hassasiyet	0.2 milli-gauss (mG)
Örnekleme Hızı	15 Hz (varsayılan), 75 Hz (yüksek hız)
Ölçüm Çözünürlüğü	12-bit
Power Management	Düşük güç tüketimi için enerji tasarrufu modları
Boyutlar	Genellikle 3 mm x 3 mm
İletişim Arabirimi	I2C (TWI)
Özellik	Değer
Çalışma Prensibi	Işık yoğunluğuna duyarlı direnç
Işık Hassasiyeti	Ölçülen ışığın yoğunluğuna bağlı olarak değişir
Işık Spektrumu	Genellikle görünür ışık spektrumu
İletişim Arabirimi	N/A (Analog değer sağlar)
Çalışma Gerilimi	Genellikle 5V veya 3.3V (model ve tipine bağlı)
Direnç Değişikliği	Işık yoğunluğu arttıkça direnç azalır
Hızlı Yanıt Zamanı	Genellikle hızlı yanıt verir

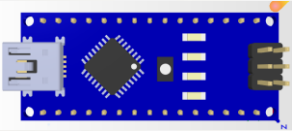
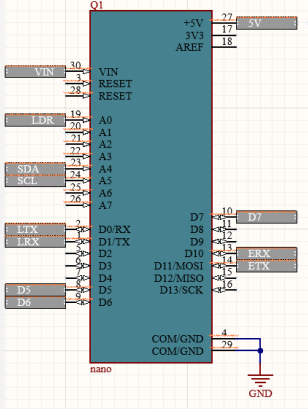
Tablo 1.6 HMC5883L ve LDR Sensör Özellikleri



Resim 2.7 ESP CAM Modülü

Özellik	Değer
Mikrodenetleyici	ESP32 mikrodenetleyici
Kamera Modülü	Genellikle OV2640 veya OV7670 gibi kamera modülleri kullanılır
Kamera Çözünürlüğü	Model ve kamera modülüne bağlı olarak değişebilir (genellikle 2MP)
İletişim Arabirimi	SPI, I2C, UART ve GPIO pinleri
Kablosuz İletişim	Wi-Fi (ESP32'nin entegre Wi-Fi desteği)
Depolama	MicroSD kart yuvası
Çalışma Gerilimi	Genellikle 5V veya 3.3V (model ve tipine bağlı)
Güç Tüketimi	Model ve çalışma koşullarına bağlı olarak değişebilir

Tablo 1.7 ESP CAM Özellikleri

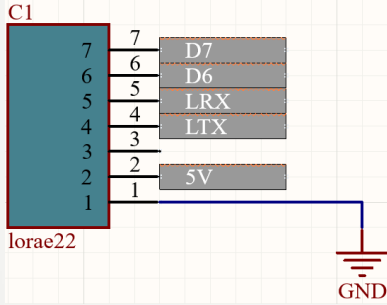


**Resim 2.8** Atmega 328P Mikrodenetleyicisi

Özellik	Değer
Mikrodenetleyici	ATmega328P veya ATmega168P mikrodenetleyici (model ve üreticiye bağlı)
İşlemci Hızı	Genellikle 16 MHz (ATmega328P) veya 8 MHz (ATmega168P)
Flash Bellek Kapasitesi	Genellikle 32 KB (ATmega328P) veya 16 KB (ATmega168P)
SRAM Kapasitesi	Genellikle 2 KB (ATmega328P) veya 1 KB (ATmega168P)
EEPROM Kapasitesi	Genellikle 1 KB (ATmega328P) veya 512 byte (ATmega168P)
İletişim Arabirimleri	USB (seri iletişim) ve GPIO pinleri
Giriş/Çıkış Pinleri	Genellikle 14 dijital giriş/çıkış ve 8 analog giriş (model ve üreticiye bağlı)
Analog Giriş Hassasiyeti	10 bit (0-1023)
Dijital Giriş Hassasiyeti	Yüksek ve düşük seviye (1 veya 0)
Çalışma Gerilimi	5V (ATmega328P tabanlı) veya 3.3V (ATmega168P tabanlı)
Boyutlar	Genellikle 45 mm x 18 mm (model ve üreticiye bağlı olarak değişebilir)
Mikrodenetleyici	ATmega328P veya ATmega168P mikrodenetleyici (model ve üreticiye bağlı)

**Tablo 1.8** Arduino Nano Özellikleri

# Kullanılan Elektronik Komponentleri

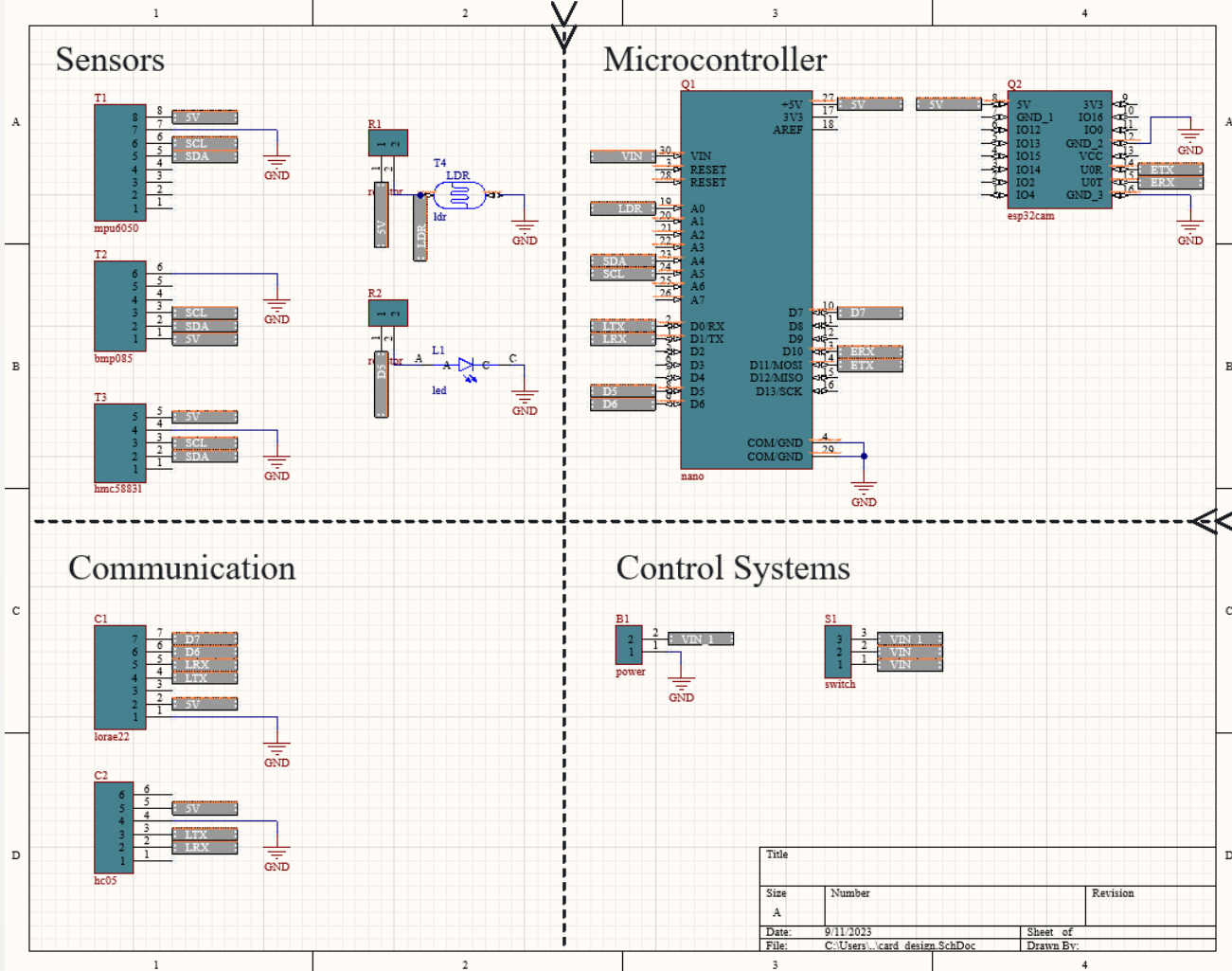


Resim 2.9 Lora E22 Haberleşme Modülü

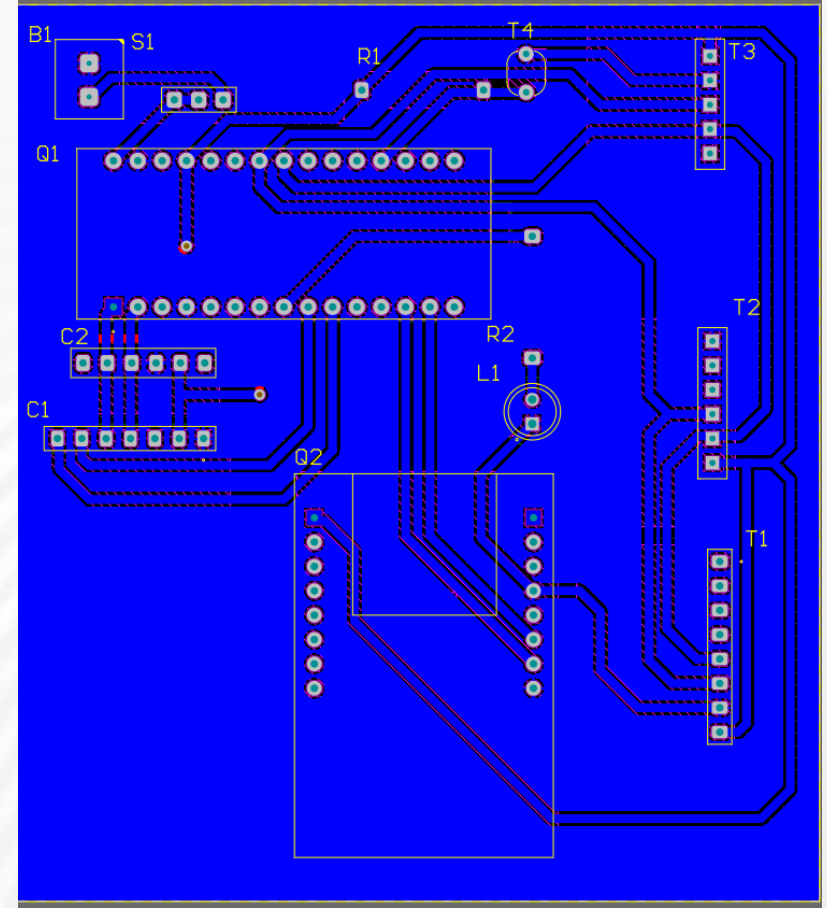
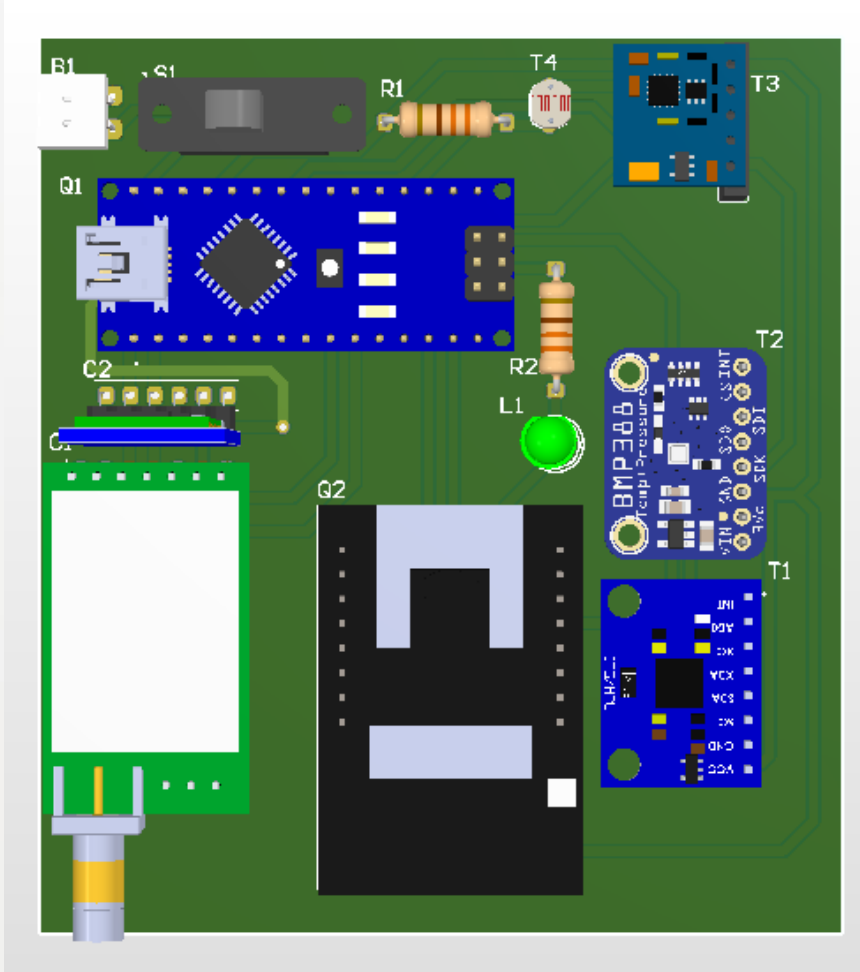
Özellik	Açıklama
Frekans Aralığı	433MHz veya 868MHz (seçeneklere bağlı olarak)
Modülasyon	LoRa (Uzun Menzilli)
Çıkış Gücü	433MHz versiyon için 20dBm, 868MHz versiyon için 20dBm (ayarlanabilir)
Alıcı Hassasiyeti	-148dBm (SF12, 125kHz)
Verici Gücü Tüketimi	Ortalama 30mA (transmisyon sırasında)
Alıcı Güç Tüketimi	Ortalama 10mA (bekleme modunda)
İletişim Arayüzü	UART (Serial)
Çalışma Gerilimi	3.3V (5V tolere edebilir, ancak seviye dönüştürme gerekebilir)
Veri Hızı	1200bps ila 115200bps (ayarlanabilir)
Anten Konnektörü	SMA dişi konektör (anten dahil değil)
Maksimum İletişim Mesafesi	Tipik olarak 5km ila 10km (çevresel koşullara bağlı olarak)
Çalışma Sıcaklığı Aralığı	-40°C ila 85°C

Tablo 1.9 LORA E22 Özellikleri

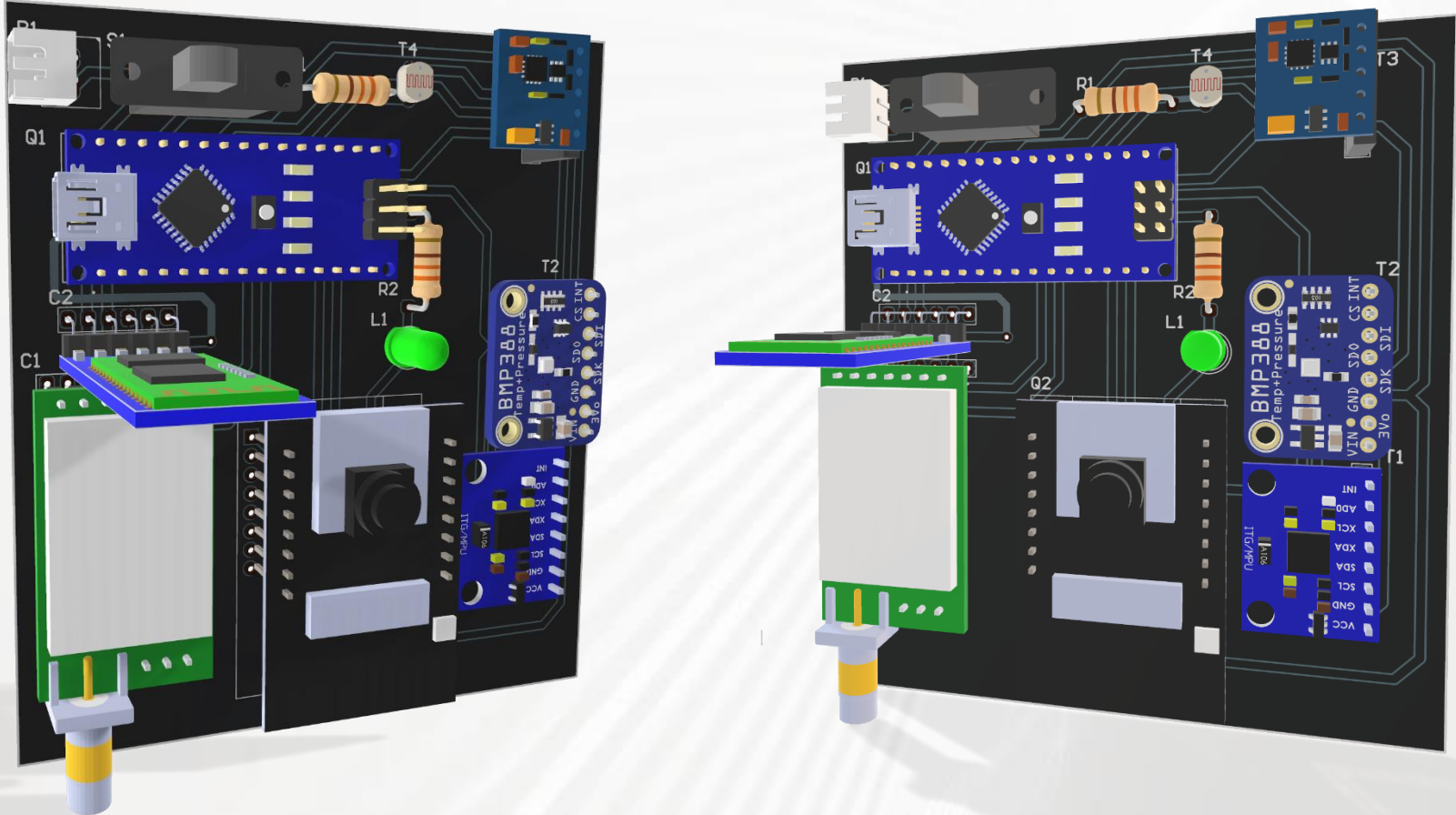




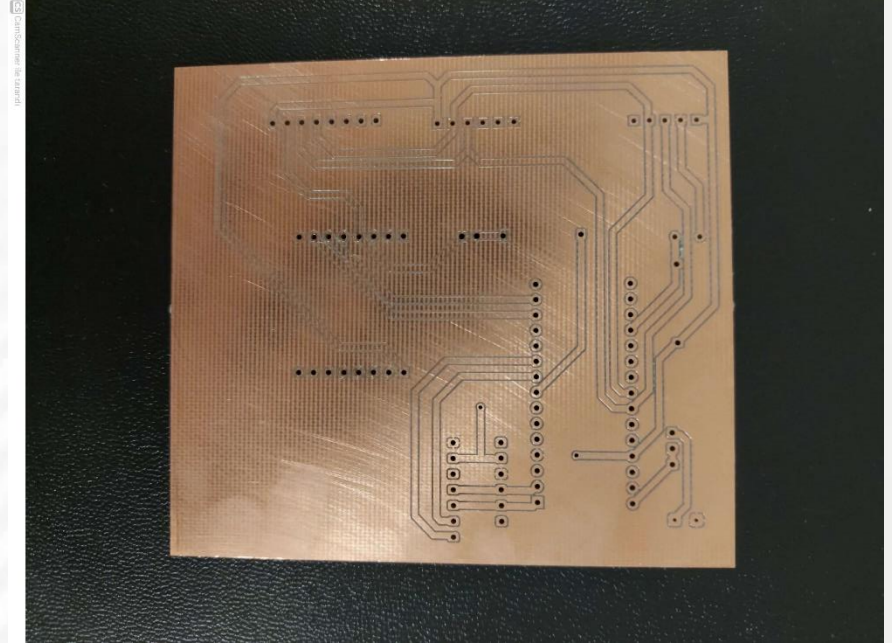
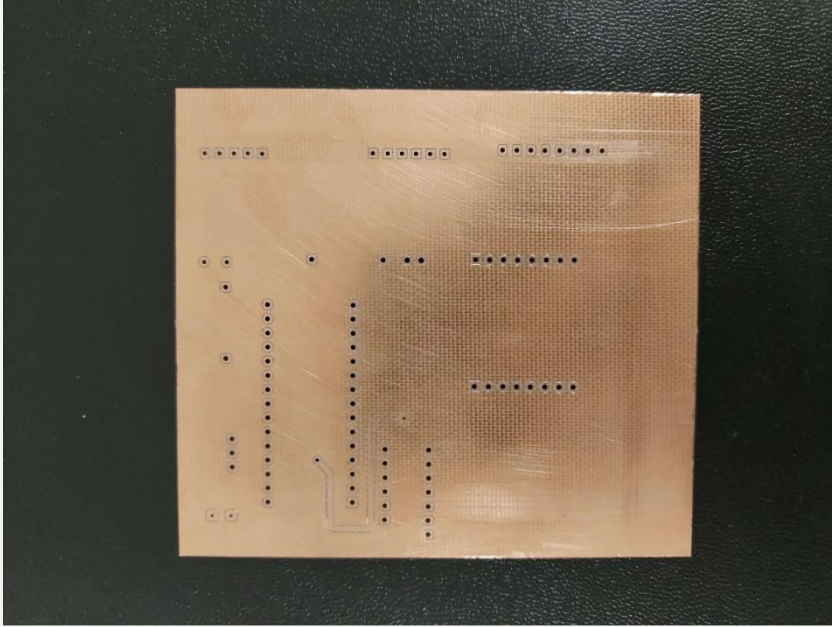
Resim 3. Elektronik Kart Tasarım Şematığı



Resim 3.1 Elektronik Kart Tasarım 2 Boyutlu Görünümü

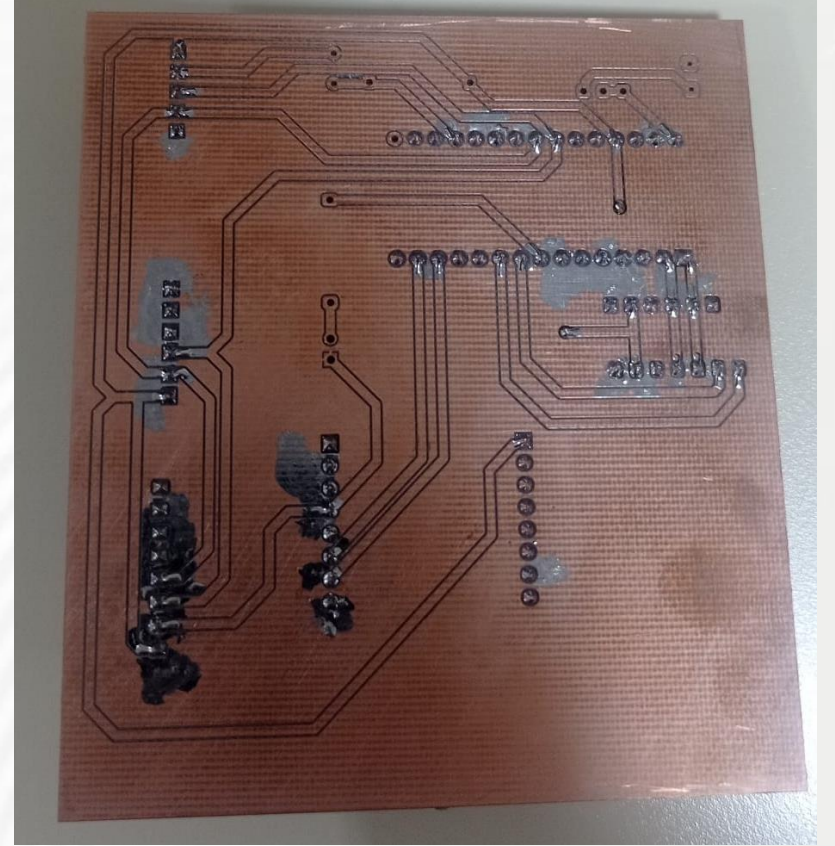
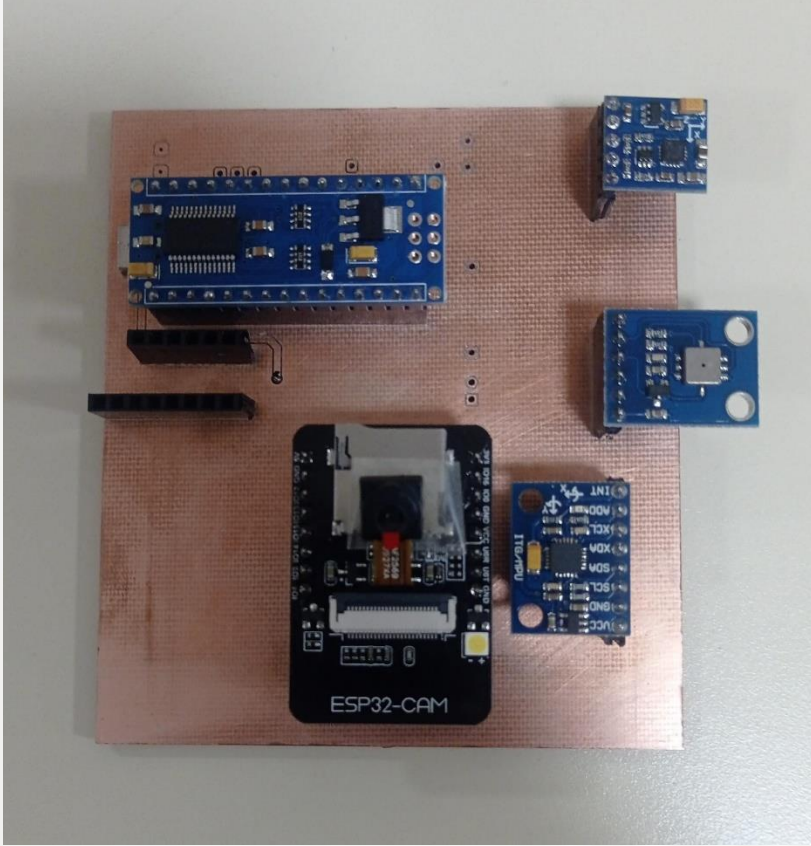


Resim 3.2 Elektronik Kart Tasarım 3 Boyutlu Görünümü



**Resim 3.3** Elektronik Kart Tasarımı Baskı Sonucu





Resim 3.4 Elektronik Kart Tasarım Komponent Entegre Görseli

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

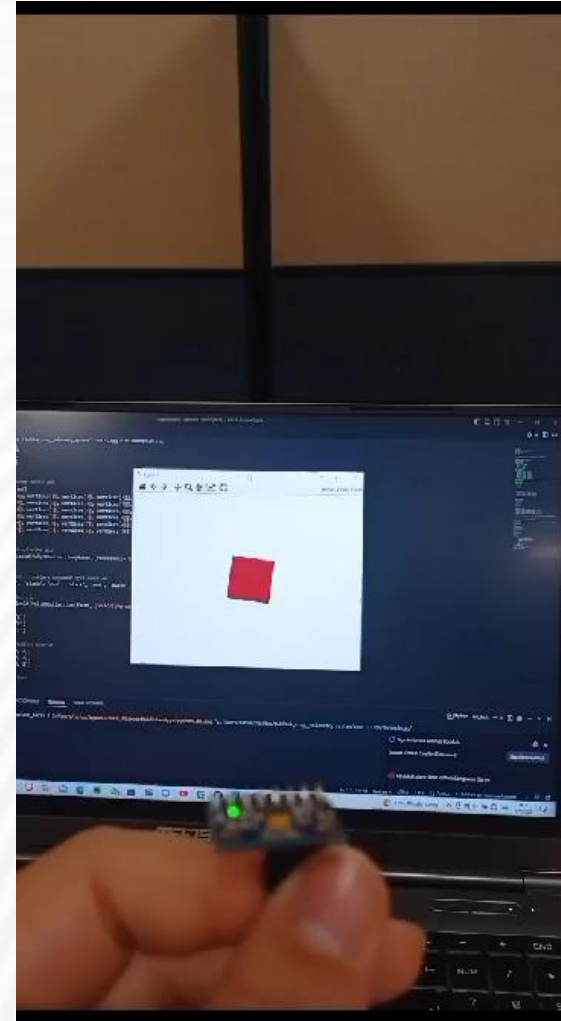
void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
}

void loop() {
  int ax, ay, az;
  mpu.getAcceleration(&ax, &ay, &az);

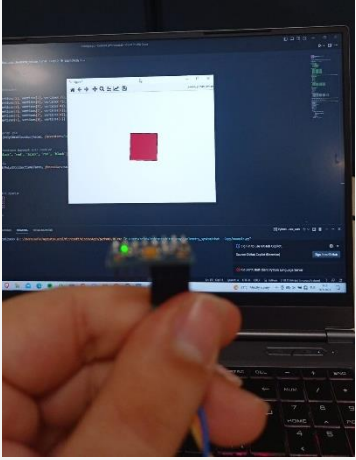
  float roll = atan2(ay, az) * 180.0 / PI;
  float pitch = atan2(-ax, sqrt(ay * ay + az * az))
  * 180.0 / PI;

  Serial.print(roll);
  Serial.print("/");
  Serial.println(pitch);

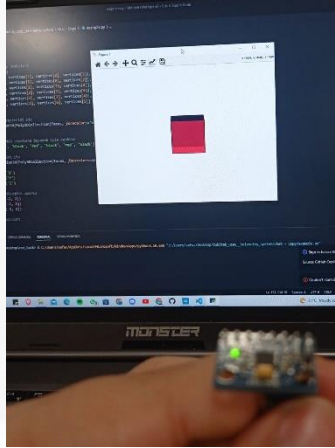
  delay(100);
}
```



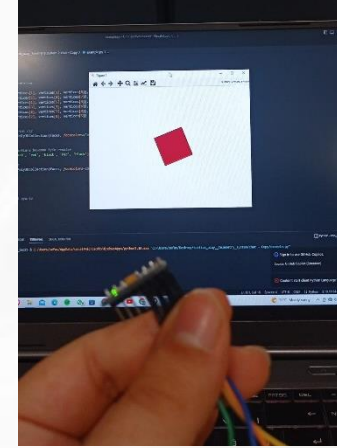
Video 1.3 IMU 3D Kontrol Algoritma Test Sonuçları



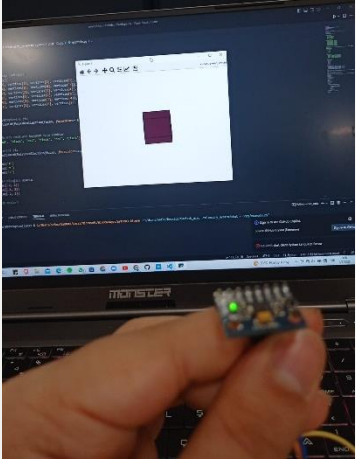
Resim 3.4 Başlangıç Konumu



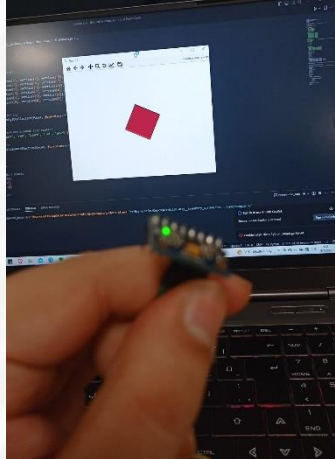
Resim 3.5 Roll Eksenindeki Değişim



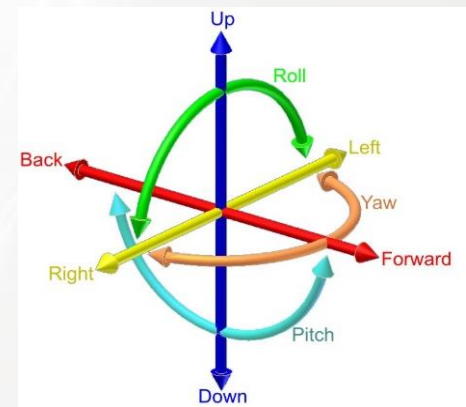
Resim 3.6 Pitch Eksenindeki Değişim



Resim 3.7 Roll Eksenindeki Değişim



Resim 3.8 Pitch Eksenindeki Değişim



Resim 3.9 Roll, Pitch, Yaw Eksenleri

```
import serial
import time
from mpl_toolkits.mplot3d.art3d import
Poly3DCollection
import matplotlib.pyplot as plt
import numpy as np
import sys
import math

# Arduino ile seri iletişim kur
ser = serial.Serial('COM9', 9600) # Port ve baud
rate'i uygun şekilde ayarlayın

# 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Dikdörtgen yüzey noktaları
vertices = np.array([
    [-1, -1, -1],
    [1, -1, -1],
    [1, 1, -1],
    [-1, 1, -1],
    [-1, -1, 1],
    [1, -1, 1],
    [1, 1, 1],
    [-1, 1, 1]
])
```

```
# Dikdörtgen yüzey indisleri
faces = np.array([
    [vertices[0], vertices[1], vertices[2], vertices[3]], #
    Alt yüzey
    [vertices[4], vertices[5], vertices[6], vertices[7]], #
    Üst yüzey
    [vertices[0], vertices[1], vertices[5], vertices[4]], # Ön
    yüzey
    [vertices[2], vertices[3], vertices[7], vertices[6]], #
    Arka yüzey
    [vertices[0], vertices[3], vertices[7], vertices[4]], #
    Sol yüzey
    [vertices[1], vertices[2], vertices[6], vertices[5]] #
    Sağ yüzey
])
```



```
# Dikdörtgen yüzeylerini çiz
ax.add_collection3d(Poly3DCollection(faces,
facecolors='blue', linewidths=1,
edgecolors='black', alpha=0.6))
```

```
# Yüzeyleri farklı renklere boyamak için renkler
colors = ['red', 'black', 'red', 'black', 'red',
'black']
```

```
# Küp yüzeylerini çiz
ax.add_collection3d(Poly3DCollection(faces,
facecolors=colors, linewidths=1,
edgecolors='black', alpha=0.6))
```

```
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
```

```
# Eksenlerin ölçeğini ayarla
```

```
ax.set_xlim([-2, 2])
ax.set_ylim([-2, 2])
ax.set_zlim([-2, 2])
```

```
# Küp dönüş açıları
roll = 0
pitch = 0
```

```
# Eksenleri kaldır
ax.set_axis_off()
```

```
# Ana döngü
try:
```

```
    while True:
        data = ser.readline().decode().strip()
```

```
        if data:
            roll, pitch = map(float, data.split('/'))
            ax.view_init(elev=pitch, azimuth=roll)
            plt.draw()
            plt.pause(0.01)
```

```
        if 0xFF == ord('q'):
            break
```

```
except KeyboardInterrupt:
    ser.close()
    print("Serial communication closed.")
```

```
plt.show() # Pencereyi göster
```

# Hedef Takip Sistemi (ESP Kamera Modülü)



TTL dönüştürücü yardımıyla ESP CAM pin üzerinden UART (Tx, Rx) Protokolü ile haberleştiğimiz ESP CAM üzerinde bu bağlantılara ek olarak haberleşme sırasında IO0 pini parazitlik olmaması için GND bağlantısı gerçekleştirilir.

Yanda verilen resimde de görüleceği üzere aynı TTL Dönüştürücüye sahip olduğum için ESP CAM kablo renk eşleştirilmesi sağlanmıştır.

Resim 4. Kablolu TTL Dönüştürücü

```
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
#define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
// ** Espressif Internal Boards **
#define CAMERA_MODEL_ESP32_CAM_BOARD
#define CAMERA_MODEL_ESP32S2_CAM_BOARD
#define CAMERA_MODEL_ESP32S3_CAM_LCD
#define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
#define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
#include "camera_pins.h"
```

Resim 4.1 ESP CAM Kütüphaneleri

```
const char* ssid = "sefa";
const char* password = "123456789";

void startCameraServer();
void setupLedFlash(int pin);

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();
}
```

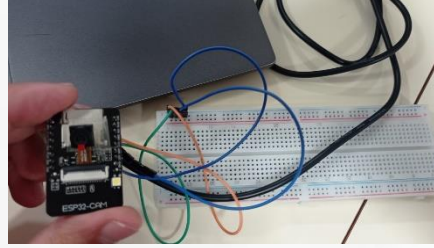
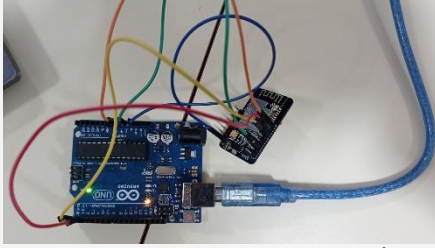
Resim 4.2 ESP CAM Erişim Sağlanması

ESP Cam Server kodundan benim kullanacağım modülünün kütüphanelerini aktif hale getirdim.

Bağlantıları tamamladıktan sonra ESP Cam Server Kodunu çalıştırdığımızda, ESP'yi herhangi bir internet ağına bağlamamız gerekmektedir. Bunun için kod üzerinde düzenleme yapılması gerekmektedir.

Kodun yüklenmesi aşamasında baudrate'i 115200 olarak seçtim. Sistem harici olarak 9600 kullandığım için çakışma durumuna dikkat ettim.

# Hedef Takip Sistemi (ESP Kamera Modülü)



**Resim 4.3** ESP CAM Kodun İmplemente Edilmesi

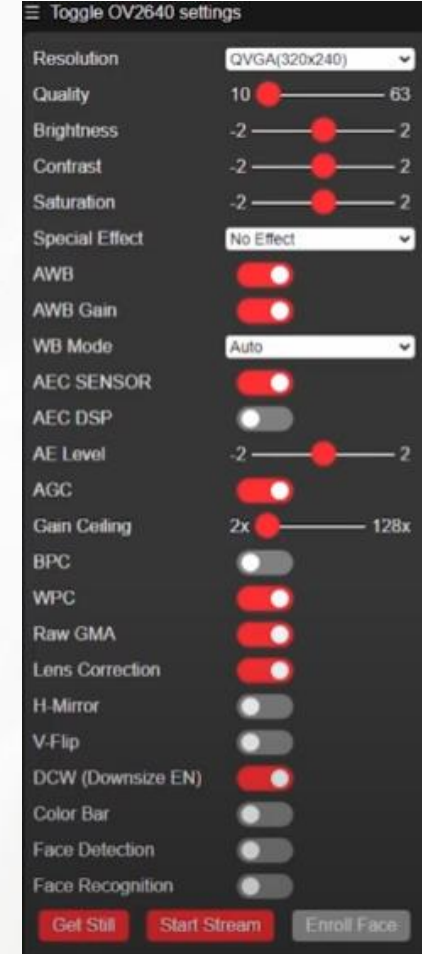
Ayrıca ESP Cam üzerinde programlama için herhangi bir Atmega işlemcisini de UART pinlerini kullanarak TTL dönüştürücü olarak kullanabiliriz.

```
COM10
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
no 0 tail 12 room 4
load:0x40078000,len:9720
no 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8

...
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.1.26' to connect

[ ] Otomatik Kaydırma [ ] Zaman damgasını göster [Yeni Satır] [115200 baud] [Çıkış temizle]
```

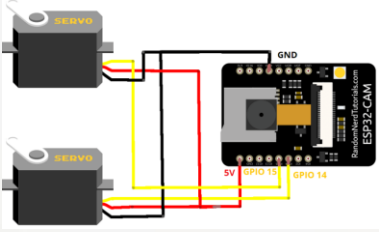
Kod yüklendikten sonra serial port ekranından ESP Cam kendi kütüphanelerini default yükledikten sonra bize yönlendireceği API adresine girerek kamera çalıştırabilir. Yan tarafta verilen görselde bu kamera sistemi için API üzerinden ayarı yapabileceğimiz gözükmektedir.



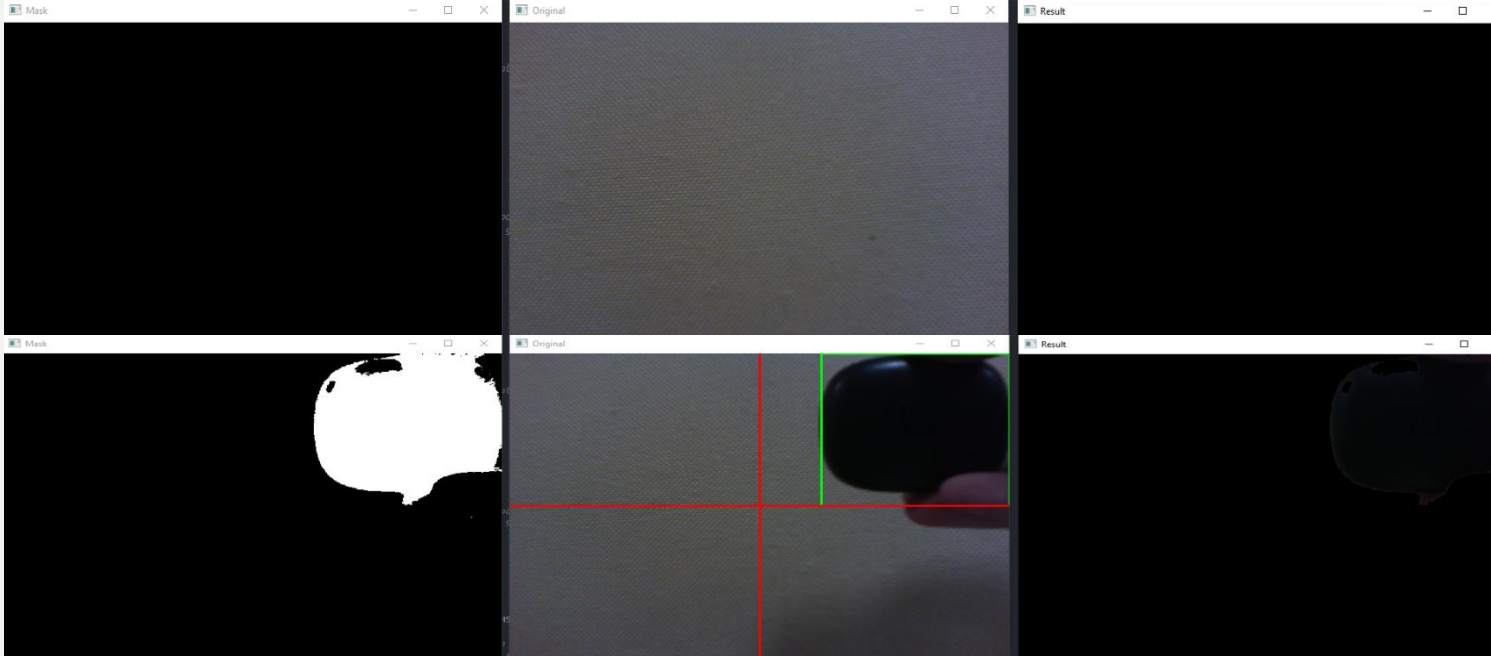
**Resim 4.4** ESP CAM Sunucusu

# Hedef Takip Sistemi (Harici Kamera Modülü)

## Servo Motor Bağlantı Devresi:

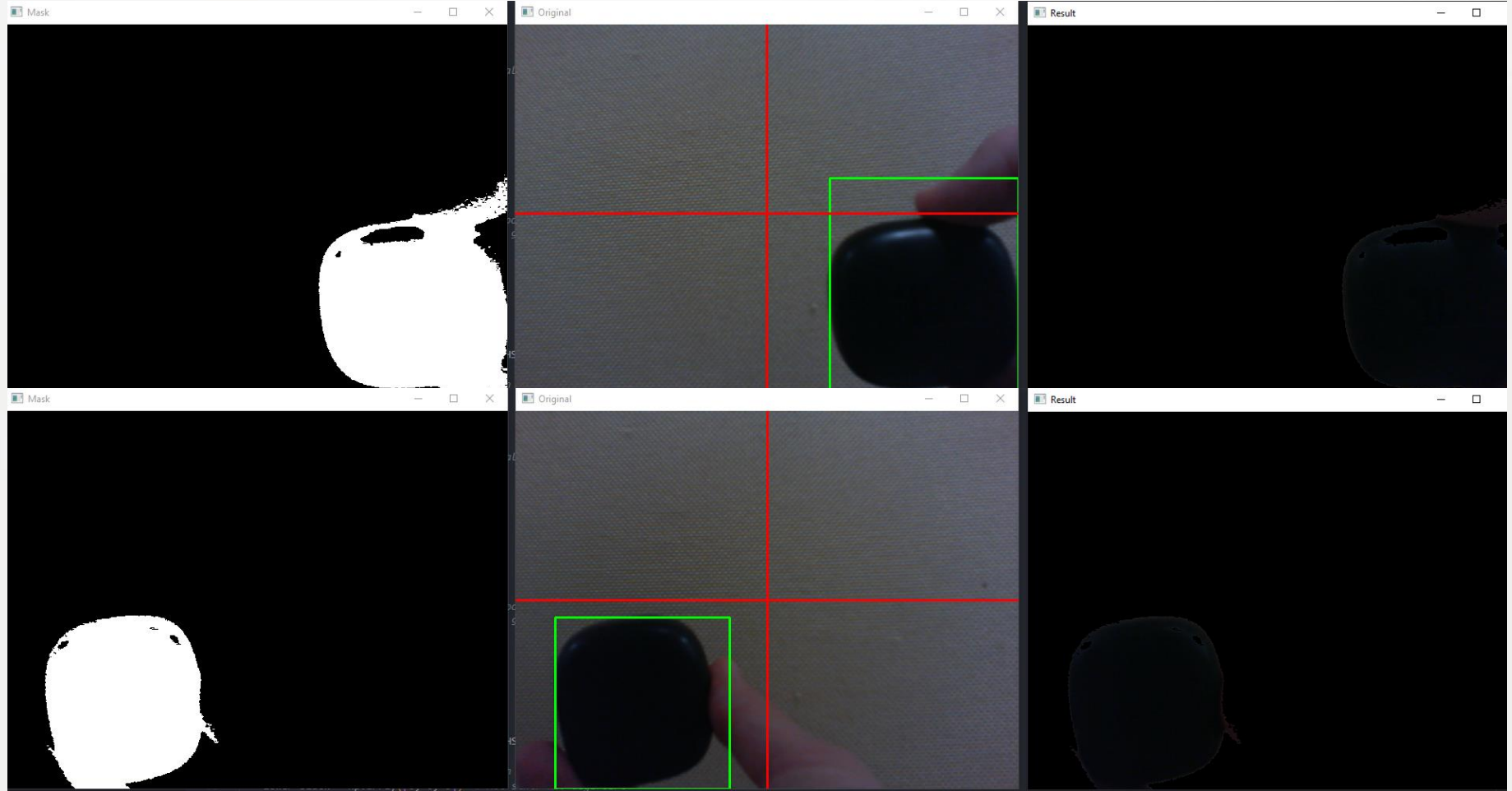


Servo motorların çalışma mantığı, kontrol sinyali ile geri besleme sistemi arasındaki hata sinyalini kullanarak istenen konuma veya açıya hassas bir şekilde ulaşma yeteneğine dayanır. Kontrol sinyali, motorun hedef konumu veya açısını belirtirken, geri besleme sistemi mevcut konumu sürekli olarak izler. Denetleyici, hata sinyalini işler ve motorun güç kaynağına uygun komutları iletir, böylece motor istenen konumu veya açıyı elde eder.



Resim 4.5 Hedef Takip Sistem Koordinatları

# Hedef Takip Sistemi (Harici Kamera Modülü)



Resim 4.6 Hedef Takip Sistem Sonuçları





# Hedef Takip Sistemi (Harici Kamera Modülü)

```
import cv2
import numpy as np
import serial

# Harici kamera cihaz numarasını belirtin
# (genellikle 0, 1, 2 gibi)
camera_device = 1

# Kamerayı başlatın
cap = cv2.VideoCapture(camera_device)

# Ekranın genişlik ve yükseklik bilgilerini alın
screen_width = int(cap.get(3))
screen_height = int(cap.get(4))

# Bölge sınırlarını hesaplayın
center_x = screen_width // 2
center_y = screen_height // 2
quarter_width = screen_width // 2
quarter_height = screen_height // 2

# Seri portu açın (Arduino'nun bağlı olduğu porta
göre değiştirin)
ser = serial.Serial('COM9', 9600) # COM3 ve 9600
hızınızı ve port numaranızı ayarlayın
```

```
while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Görüntüyü HSV renk uzayına dönüştürün
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # Siyah rengin HSV değerlerini belirleyin
    lower_black = np.array([0, 0, 0]) # Alt sınır HSV
    değerleri
    upper_black = np.array([180, 255, 30]) # Üst sınır HSV
    değerleri

    # Belirlenen siyah renk aralığını maskeleyin
    mask = cv2.inRange(hsv, lower_black, upper_black)

    # Maskeyi kullanarak nesne takibini gerçekleştirin
    result = cv2.bitwise_and(frame, frame, mask=mask)

    # Siyah nesne konturlarını bulun
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
```

# Hedef Takip Sistemi (Harici Kamera Modülü)

```
# En büyük konturu bulun
if contours:
    largest_contour = max(contours,
key=cv2.contourArea)

# Konturun merkez koordinatlarını bulun
M = cv2.moments(largest_contour)
if M["m00"] != 0:
    cX = int(M["m10"] / M["m00"])
    cY = int(M["m01"] / M["m00"])
else:
    cX, cY = 0, 0

# Nesnenin hangi bölgede olduğunu kontrol
edin
if cX < center_x and cY < center_y:
    # Sol üst bölge
    servo_acisi = 0
elif cX >= center_x and cY < center_y:
    # Sağ üst bölge
    servo_acisi = 0
elif cX < center_x and cY >= center_y:
    # Sol alt bölge
    servo_acisi = 180
else:
    # Sağ alt bölge
    servo_acisi = 180
```

```
# Nesneyi kutu içine alın
x, y, w, h = cv2.boundingRect(largest_contour)
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255,
0), 2)

# Merkezden geçen iki çizgi çizim
cv2.line(frame, (center_x, 0), (center_x,
screen_height), (0, 0, 255), 2)
cv2.line(frame, (0, center_y), (screen_width,
center_y), (0, 0, 255), 2)

# Servo açısını Arduino'ya gönderin
ser.write(str(servo_acisi).encode())

# Sonuçları gösterin
cv2.imshow('Original', frame)
cv2.imshow('Mask', mask)
cv2.imshow('Result', result)

if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Seri bağlantıyı kapatın
ser.close()

# Video yakalama işlemini serbest bırakın ve pencereleri
kapatın
cap.release()
cv2.destroyAllWindows()
```



# Hedef Takip Sistemi (Harici Kamera Modülü)

```
#include <Servo.h>

Servo servoMotor;

int servoPin = 9; // Servo sinyal pini (başka bir
pin kullanabilirsiniz)
int servoAcisi = 0; // Servo açısı

void setup() {
    servoMotor.attach(servoPin);
    Serial.begin(9600);
}

void loop() {
    if (Serial.available() > 0) {
        servoAcisi = Serial.parseInt(); // Python
        kodundan servo açısını alın

        // Servo açısını sınırlayın (örneğin, 0 ile 180
        arasında)
        servoAcisi = constrain(servoAcisi, 0, 180);

        // Servo motorunu belirtilen açıya hareket
        ettirin
        servoMotor.write(servoAcisi);
    }
}
```

- [1] Yılmaz, M., İki Eksenli Güneş Takip Sistemlerinde Takip Verimliliğın Arttırılması. Batman Üniversitesi Yaşam Bilimleri Dergisi Cilt 7, Sayı 1/2 (2017).
- [2] Oral, G., Nuri U., O.(2012). Güneş Takip Sistemleri ve Prototip Gerçekleştirme. İstanbul Aydın Üniversitesi Dergisi, 4(15), 1 – 14.
- [3] Boyacı, Ö., Kocaman, Ç.(2017). Matlab / Simulink Üzerinden Gerçek Zamanlı Gömülü Sistem Tabanlı Güneş Takip Sisteminin Tasarımı ve Uygulaması. ANKA e-DERGİ Journal of Phoenix (Teknik ve Sosyal Bilimler Dergisi) Cilt 3/Sayı 1 ISSN- 2148-7138
- [4] Pakfiliz, A.G.(2019). Uçan Nesnelerin Otomatik Tespit ve Takibi İçin Yeni Bir Yaklaşım. Pamukkale Üniversitesi Mühendislik Bilim Dergisi, 25(5),553-559.
- [5] Gökçe, B., Sonugür, G.(2016). GPS Destekli İmge Çakıştırma Yöntemi İle Hareketli Nesnelerin Tespiti. Pamukkale Üniversitesi Mühendislik Bilim Dergisi, 22(5), 353-360.
- [6] Mengi Ö., O., Altaş, İ. H.(2006). Hareketli Nesneler İçin Konum Tespiti ve İzleme Denetimi. Elektrik Elektronik Makina Bilgisayar Dergisi, 74-76.

## İnternet Kaynakları:

- <https://www.megaplc.com.tr/uart-usart-haberlesme-protokolu-nedir/>
- <https://ozdenercin.com/2019/01/25/i2c-seri-haberlesme-protokolu/>

Bu çalışma, Tübitak UZAY için verilmekte olan stajyer projesi için hazırlanmıştır. Yazar 4. Sınıfa geçmiş olup stajyer öğrencisidir. Çalışma APA formatını baz alarak hazırlanmış ve düzenlenmiştir. Bu ödevin konusu önemi hakkında araştırma, öğrenme ve kendimizi geliştirme fırsatı sağlayan Platform ve Faydalı Yük Elektroniği Tasarım Grubu Baş Araştırmacısı ve Grup Lideri Mustafa CEYLAN' a teşekkürü borç bilir ve saygılarımı sunarım.

## Teşekkürler