RASPBERRY Pİ 3 İLE KUSURSUZ DÖNÜŞLER

**GİRİŞ**

Projemizde RC model arabaların ani dönüşlerde devrilmesini engellemeye çalıştık.

Projede araba düz giderken ya da normal şekilde viraj alırken servo motorun kontrolü kumanda üzerinden gelen bilgiden sağlanacak ama araba aşırı derecede yana yattığında yani jiroskopun x ekseninde gösterdiği değer -25 ve +25 değer aralığının dışına çıktığında servo motorun kontrolü Raspberry Pi 3 üzerinden gerçekleştirilecek ve eğer araba sağ tarafa devriliyorsa servo motoru sola çevirecek , sola devriliyorsa servo motoru sağa çevirecek bu şekilde araba devrilmekten kurtulacaktır.

**Gerekli Donanım Bileşenleri**

**1.** 1 adet Raspberry Pi 3

**2.**1 adet Servo Motor

**3.**1 adet MPU6050 Jiroskop(İvme Ölçer)

**4.**Jumper kablolar

**5.**Raspberry Pi 3 için Adaptör ve SD Kart

**6.**RC Model Araba

**7.**Arabanın Kumandası

**Kullanılan Bileşenlerin Özellikleri**

**1.**Projemizde Raspberry Pi 3 kullandık.Özellikleri :

-Broadcom BCM2837 SoC

-1.2 GHz 4 çekirdekli 64-bit ARM Cortex-A53 işlemci

-2 çekirdekli Videocore IV® Multimedia işlemcisi

-1 GB LPDDR2 bellek

-Dahili 802.11b/g/n  destekli WiFi

-Bluetooth 4.1, low-energy destekli

-10/100 Mbit destekli Ethernet portu

-HDMI portu (HDMI 1.4 destekli)

-Kompozit video ve ses çıkışı için 3.5mm TRRS (4 uçlu) konektör

-4 adet USB2.0 portu

-40 adet GPIO pini, önceki Raspberry Pi modelleri ile uyumlu

-WiFi/Bluetooth için dahili çip anten

-CSI (kamera) ve DSI (ekran) konektörleri

-Mikro SD kart yuvası

**2.**MPU6050 jiroskop(3 eksenli ivme ölçer) kullandık.İnternet üzerinden temin ettik.

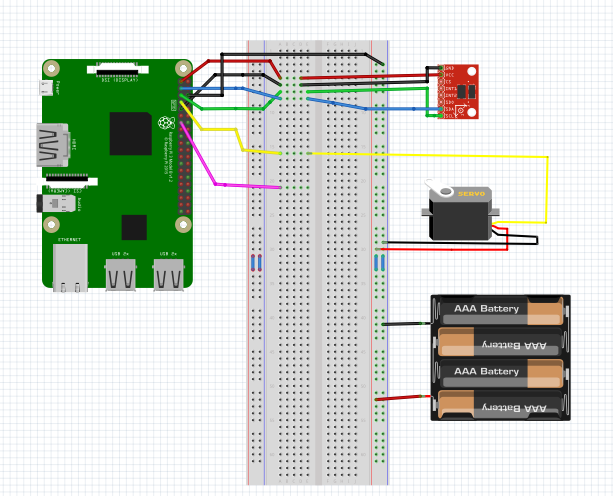
(<https://www.direnc.net/mpu6050-3-axis-gyro-ve-egim-sensoru>)

**3.**RC model araba kullandık.Kendimiz toparlamıştık daha önceden vardı.

**4.**Servo motor mevcuttu.Direksiyonu kontrol etmek amacıyla kullandık.Alınmak istenirse

(<https://www.direnc.net/mg995-servo-motor>) buradan alınabilir.

PROJEMİZİN ELEKTRONİK DEVRE ŞEMASI



**elektronikdevre.png**

**Sinyal Kablo durumları:**

Kırmızı Kablo : + Besleme

Siyah Kablo : - Besleme

Sarı Kablo : PWM sinyal

Yeşil ve Mavi Kablo : X eksenindeki eğimi ölçen sensörün kabloları

Pembe Kablo : Arabadan gelen sinyal kablosu

**Yapım Aşamaları :**

**1.**Raspberry Pi 3 ' e işletim sistemini yükledik.

**2.**Wi-Fi üzerinden bağlanabilmek adına VNC Viewer programını laptopumuza yükledik.

**3.**Raspberry üzerindeki ip adresi aracılığı ile laptop ve Raspberry arasında bağlantı kurup , Raspberry'nin ekran görüntüsünü laptopumuza aktardık.

**4.**Projemizde kullanacağımız kütüphaneleri ,gerekli olan bilgileri internet aracılığı ile öğrenmeye çalıştık.İşimize yarayabilecek kodları bir metin dosyasında sakladık.

**5.**Öncelikle ivme ölçerin çalışıp çalışmadığını kontrol ettik ve ivme ölçer üzerinden eğim verilerini ekrana yazdırdık.

**6.**Kullanacağımız servo motorun çalıştırılmasını gerçekleştirdik.

**7.**İvme ölçer ve servo motoru birlikte çalıştırarak aralarında veri alış verişine dayalı eğim bilgisine göre hareket elde ettik.

**8.**Elde ettiğimiz bu hareketi kumandadan bizim kontrolümüz dışında devreye girmesini sağladık.

PROJEDE KULLANILAN PYTHON KODLARI

1. **import** smbus
2. **import** math
3. **import** time
4. **import** RPi.GPIO as GPIO

Kodlarımızın çalışması için kütüphaneleri belirttik.

1. GPIO.setwarnings(False)
2. GPIO.setmode (GPIO.BOARD)
3. GPIO.setup(7, GPIO.OUT)
4. GPIO.setup(13, GPIO.IN)

GPIO ayarlarını yaptık.Girdiğimiz pin sayıları board üzerindedir.

1. pwm = GPIO.PWM(7, 100)
2. pwm.start(7.5)

PWM pin ve frekans değerini girdik.Başlangıç pwm değerini girdik.

1. power\_mgmt\_1 = 0x6b
2. power\_mgmt\_2 = 0x6c

Güç yönetim registerlari

1. **def** read\_byte(adr):
2. **return** bus.read\_byte\_data(address, adr)
3. **def** read\_word(adr):
4. high = bus.read\_byte\_data(address, adr)
5. low = bus.read\_byte\_data(address, adr+1)
6. val = (high << 8) + low
7. **return** val
8. **def** read\_word\_2c(adr):
9. val = read\_word(adr)
10. **if** (val >= 0x8000):
11. **return** -((65535 - val) + 1)
12. **else**:
13. **return** val
14. **def** dist(a,b):
15. **return** math.sqrt((a\*a)+(b\*b))
16. **def** get\_y\_rotation(x,y,z):
17. radians = math.atan2(x, dist(y,z))
18. **return** -math.degrees(radians)
19. **def** get\_x\_rotation(x,y,z):
20. radians = math.atan2(y, dist(x,z))
21. **return** math.degrees(radians)
22. bus = smbus.SMBus(1)
23. address = 0x68

Belirttiğimiz gibi MPU6050 I2C adresini belirledik.

1. bus.write\_byte\_data(address, power\_mgmt\_1, 0)

MPU6050 ilk calistiginda uyku modunda oldugundan, calistirmak icin bu komutu girdik.

1. **while** True:
2. time.sleep(0.5)
4. gyro\_xout = read\_word\_2c(0x43)
5. gyro\_yout = read\_word\_2c(0x45)
6. gyro\_zout = read\_word\_2c(0x47)
8. **print** "Jiroskop X : ", gyro\_xout, " olcekli: ", (gyro\_xout / 131)
9. **print** "Jiroskop Y : ", gyro\_yout, " olcekli: ", (gyro\_yout / 131)
10. **print** "Jiroskop Z: ", gyro\_zout, " olcekli: ", (gyro\_zout / 131)

13. accel\_xout = read\_word\_2c(0x3b)
14. accel\_yout = read\_word\_2c(0x3d)
15. accel\_zout = read\_word\_2c(0x3f)
17. accel\_xout\_scaled = accel\_xout / 16384.0
18. accel\_yout\_scaled = accel\_yout / 16384.0
19. accel\_zout\_scaled = accel\_zout / 16384.0
21. **print** "Ivmeolcer X: ", accel\_xout, " olcekli: ", accel\_xout\_scaled
22. **print** "Ivmeolcer Y: ", accel\_yout, " olcekli: ", accel\_yout\_scaled
23. **print** "Ivmeolcer Z: ", accel\_zout, " olcekli: ", accel\_zout\_scaled
25. **print** "X dondurme: " , get\_x\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled)
26. **print** "Y dondurme: " , get\_y\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled)
28. time.sleep(0.5

Sonsuz döngüye sokarak MPU6050 ivme ölçerden sürekli eğim bilgisini alıyoruz ve bu bilgiyi yazdırıyoruz.

1. **if** get\_x\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled) > 25 :
2. **print**("turn right now")
3. pwm.ChangeDutyCycle(12.5)
4. **elif** get\_x\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled) < -25 :
5. **print**("turn left now")
6. pwm.ChangeDutyCycle(2.5)

MPU6050'den okuduğumuz eğim değerlerine göre servo motorun sağ ve sol hareketini sağlıyoruz.Bu hareketi eğer '' x döndürme '' değerini if kalıplarına sokarak yapıyoruz.Eğer sağa doğru bir eğilme varsa servo arabanın yönünü sol tarafa çeviriyor , sola doğru bir eğim söz konusu ise servo arabanın yönünü sağa doğru çeviriyor.

1. **elif** (get\_x\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled) > -25) **and** (get\_x\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled) < 25) :
2. deger = 1
3. **while** deger:
4. **print**("stay there")
5. pwm.stop()
6. GPIO.output(7, GPIO.input(13))
7. time.sleep(0.5)
9. accel\_xout = read\_word\_2c(0x3b)
10. accel\_yout = read\_word\_2c(0x3d)
11. accel\_zout = read\_word\_2c(0x3f)
13. accel\_xout\_scaled = accel\_xout / 16384.0
14. accel\_yout\_scaled = accel\_yout / 16384.0
15. accel\_zout\_scaled = accel\_zout / 16384.0
16. **if** get\_x\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled) < -25:
17. deger = 0
18. **if** get\_x\_rotation(accel\_xout\_scaled, accel\_yout\_scaled, accel\_zout\_scaled) > 25:
19. deger = 0

Eğer bir eğim söz konusu değil ise servo motoruna giden pwm sinyali direksiyonu ortada tutacak şekilde arabanın düz hareketini sağlar.Bu hareket while döngüsü içerisinde gerçekleşir.Bu döngü içerisinde sensör bilgileri tekrar tekrar okunur.Eğer sensör bilgileri istenilen değerler altına veya üstüne gelirse while döngüsünden çıkılarak yukarıda anlatmış olduğumuz döndürme hareketini yapar.

**Kaynak Kodu**

Projemizin resimlerine,videolarına ve kaynak koduna [https://github.com/umtcsgn/Raspberry-Proje.git](https://github.com/umtcsgn/Raspberry-Proje.git%20) adresinden erişilebilir.

**Nasıl Kullanılır**

Yukarıda paylaştığımız bağlantıdan kaynak kodunu koplayınız.Kopyaladığınız kaynak kodunu Raspberry Pi 'nin içindeki pi klasörüne ".py" uzantılı olarak kaydediniz.Kaydettiğiniz bu dosyayı araba ile Raspberry arasındaki bağlantıları yaptıktan sonra komut satırında çalıştırınız.

asdasd.PNG

Arabayı kullanmaya başlayabilirsiniz.

**RESİMLER**

****

**araba2.png**



**araba1.png**