

**T.C.**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**Gömülü Sistemler ve Uygulama Ödevi**

**ÖĞRENCİNİN;**

**ADI SOYADI : Ahmet YEŞİLYURT**

**OKUL NO : 171215005**

**RASPBERRY Pİ ile LAZER ODAKLAMA SİSTEMİ**

**Giriş :**

“Silahlı Lazer Odaklama Sistemi” projesi hareketli bir nesnenin(kişinin) lazer sistemi ile takibinin sağlar. Ülkemizin içindeki ve sınırındaki terör unsurlarının çoğalması ile çatışma bölgesi artmaktadır. Çatışma bölgelerinin artması ile askerimiz ve bölücü terör unsurları ile yakın çatışmalar sıkça olmaktadır. Bu çatışmalarda da birçok şehit vermiş bulunmaktayız. “Lazer Odaklama Sistemi” de bu soruna yönelik bir çözümdür. Sistem sabit kullanılabilir veya hareketli bir sistem üzerine kolayca entegre edilebilir. Sisteme silah entegre edilerek tespit edilen düşmanları etkisiz hale getirir.

Projenin ilk aşamasında kameradan alınan görüntü, raspberry pi de open cv kütüphanesi ile işlenerek görüntüdeki yüzleri algılar. Algılanan yüz kare içine alınır ve bu karenin merkez koordinatları bulunur.

Bir sonraki adımda ise alınan bu merkez koordinatlar 2 eksende hareket eden bir mekanik sisteme aktararak, mekanik sistem üzerindeki lazerin algılanan hareketli kişiyi sürekli şekilde takip etmeyi sağlayacaktır.

**Gerekli Donanım Bileşenleri :**

* 1 adet Raspberry Pi
* 1 adet Raspberry Pi Kamera
* 1 adet Lazer
* 2 adet DC Motor
* 1 adet L298N

**Gerekli Yazılım Bileşenleri :**

-Raspbian Jessie OS (https://www.raspberrypi.org/downloads/raspbian/)

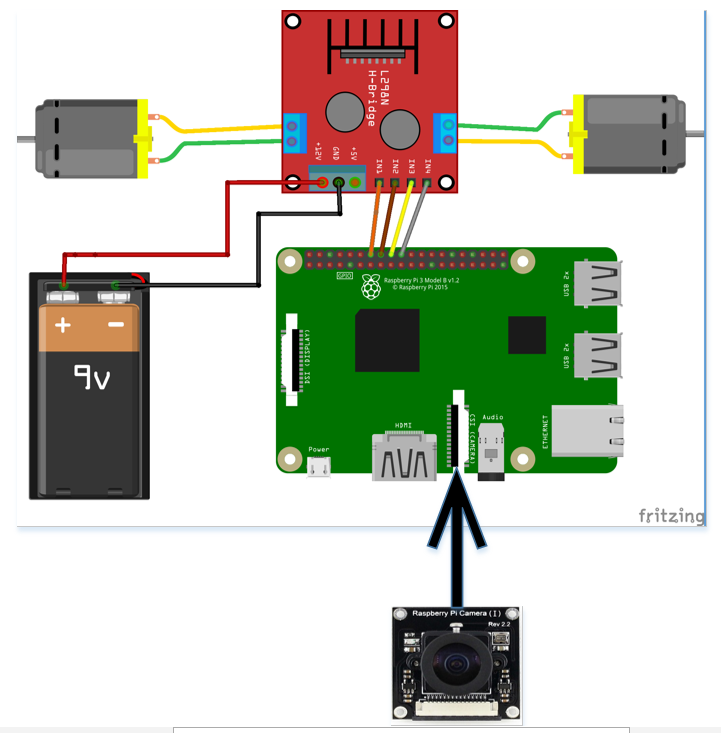
Raspbian, Raspberry Pi için optimize edilmiş Debian'a dayalı tamamen ücretsiz bir işletim sistemidir. Bu işletim sistemi, Raspberry Pi'nizi çalıştıran temel programlar ve yardımcı programlar kümesidir.

**Kullanılan Bileşenlerin Özellikleri :**

1.Raspberry Pi 3 ; Raspberry Pi bir tek kart bilgisayardır. Bunun anlamı, bir bilgisayar için gerekli olan işlemci, RAM bellek, giriş/çıkışlar gibi tüm birimler tek bir devre kartı üzerinde toplanmıştır. Raspberry Pi 3’te 1.2 GHz 64-bit 4 çekirdekli ARM Cortex-A53 işlemci ve 1 GB RAM’e sahiptir.

2. Web Kamera ; Herhangi bir web kamera kullanılabilir.

**Şematik Çizimi :**



**Resim 1**

**Yapım Aşamaları :**

***-Open CV kurulumu***

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında İntel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmektedir. Kameradan alınan görüntüyü bu kütüphanedeki komutlar yardımı ile işleyerek daha kolay bir şekilde çıktılara alırız.

***Adım 1 :***

İlk olarak Rasbian OS sistemini güncellemek için aşağıdaki kodu terminalde çalıştırıyoruz.

* sudo apt-get update

***Adım 2 :***

Raspberry Pi kartımıza open cv kütüphanesini kurmak için aşağıdaki linkteki işlemleri sırasıyla yapıyoruz

link: (<http://www.life2coding.com/install-opencv-3-4-0-python-3-raspberry-pi-3/>)

***Adım 3 :***

Bu adımda open cv kütüphanesini doğru yükleyip yüklemediğimizi kontrol ediyoruz. Aşağıdaki kodları terminal arayüzünde sırasıyla çalıştırıyoruz.

* python3
* import cv2
* cv2.\_\_version\_\_

Eğer hata almıyorsa open cv kütüphanesini doğru yüklemişizdir.

***-Python Kodu***

|  |
| --- |
| import cv2  import RPi.GPIO as GPIO  face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')  cap = cv2.VideoCapture(0)  while 1:  ret, img = cap.read()  gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  faces = face\_cascade.detectMultiScale(gray, 1.5, 3)  for (x,y,w,h) in faces:  cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)    #Yüzü kare içine aldıktan sonra, karenin merkezini bulma  x\_eksen=x+(w/2)  y\_eksen=y+(h/2)  print ("[",int(x\_eksen),int(y\_eksen),"]")  # Kullandığımız ekran boyutumuz 640x480  #Görüntünün merkez noktası ise (320,240)  #Motorları hareket ettirmek için L298N motor sürücüsüne sinyal yolluyuruz  if(x\_eksen < 320):  GPIO.wait\_for\_edge(14, GPIO.RISING)  GPIO.wait\_for\_edge(15, GPIO.FALLING)  if(x\_eksen > 321):  GPIO.wait\_for\_edge(15, GPIO.RISING)  GPIO.wait\_for\_edge(14, GPIO.FALLING)  if(y\_eksen < 240):  GPIO.wait\_for\_edge(16, GPIO.RISING)  GPIO.wait\_for\_edge(28, GPIO.FALLING)  if(y\_eksen > 241):  GPIO.wait\_for\_edge(28, GPIO.RISING)  GPIO.wait\_for\_edge(16, GPIO.FALLING)    cv2.imshow('img',img)  k = cv2.waitKey(30) & 0xff  if k == 27:  break  cap.release()  cv2.destroyAllWindows() |

Yüzün algılanmasında öğretilmiş yüz dosyası olan .xml uzantılı dosyayı kullanıyoruz. Bu dosyaya aşağıdaki github linktenden ulaşılabilir.

link: (<https://github.com/opencv/opencv/tree/master/data/haarcascades>)

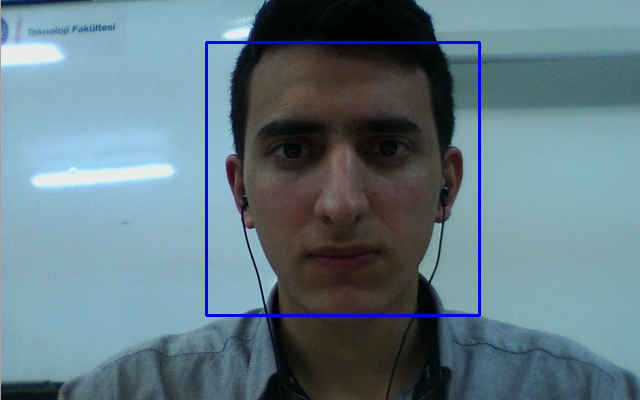
Yazılan python kodunun dosyası ile .xml uzantılı dosya aynı dizinde yer alması lazım. Eğer faklı dizinde bulunuyor ise .xml dosyanın tam adresini vererek de kodumuzu derleyebiliriz.

Kamera çalıştığında ekrandaki yüzü algılayarak bir kare içerisine alıyoruz. Bu karenin köşelerini değişkenler yardımı ile alarak ve bu parametreler ile matematiksel bazı hesaplamalar ile karenin merkezini buluyoruz.

Karenin merkezini bularak algılanan yüzün merkez noktasını bulmuş oluyoruz. Daha sonra bu merkez noktayı referans alınarak 2 eksenli mekanik sitemdeki motorlara enerji verilerek, merkez noktanın sabit kalınması sağlanıyor. Böylelikle mekanizma da bulunan lazer, algılanan yüze odaklanarak yüzün-kişinin takibini sağlamış olur.

Motor kontrolü L298N motor sürücüsü ile yapılmaktadır. L298N motor sürücüsü ile motorları ileri-geri yönde kontrol etmeye imkan sağlıyor. Böylece 2 eksenli mekanizmayı +x, -x, +y ve –y yönlerinde hareket ettirerek 180 derecelik bir alanda yüz takibi sağlamaktadır.

***-Çalışma Görüntüleri***

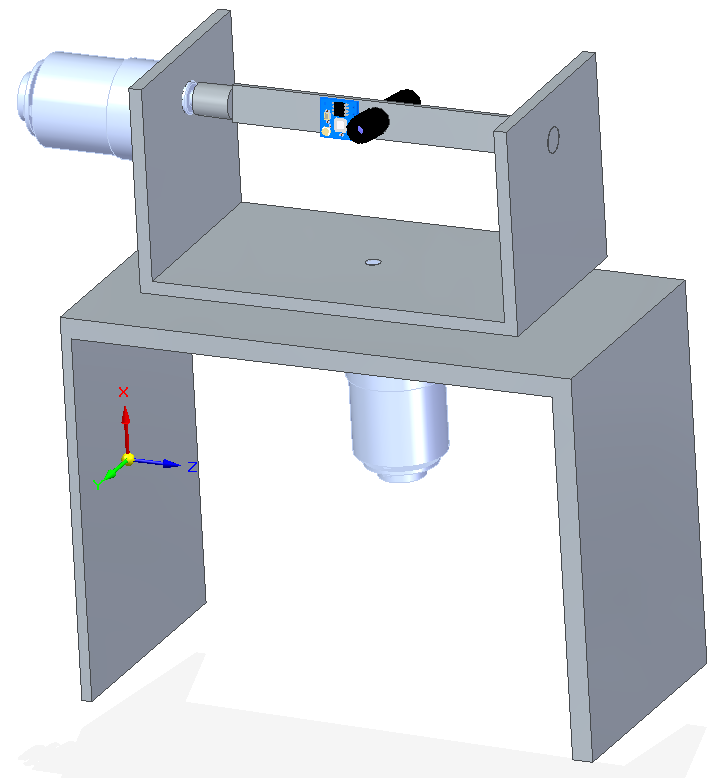
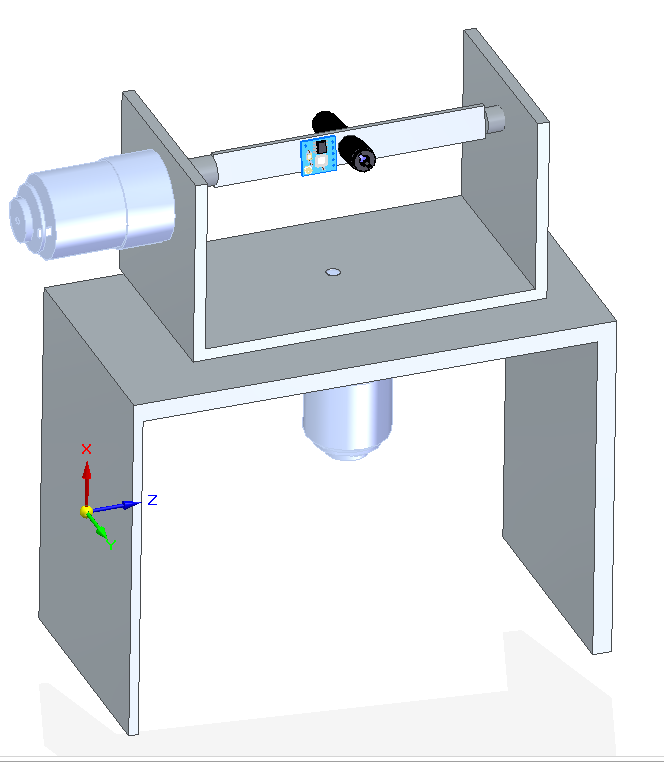


Resim2 . kareAlma

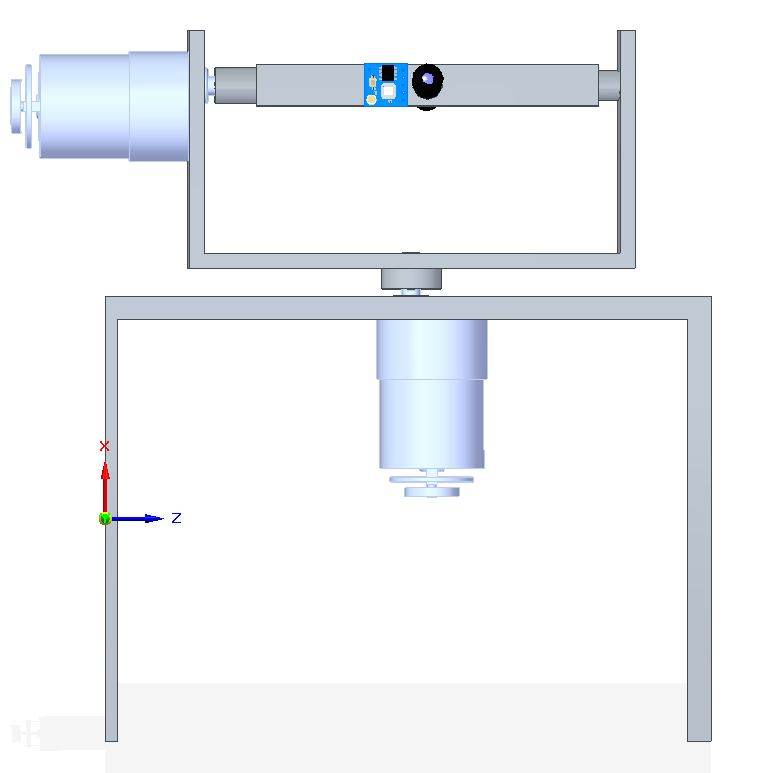


Resim 3 . Koordinat

**2 Eksenli Hareket Mekanizması :**



**Resim 4 Resim 5**



**Resim 6**

**Kaynak Kodu :**

GitHub linkinden proje dosyalarına ve kodlara erişebilirsiniz.

( <https://github.com/AhmetYesilyurt/Lazer_Takip_Sistemi_2> )

Bu linkten projenin çalışır videosuna ulaşabilirsiniz.

( <https://drive.google.com/open?id=1m1VlzNHK065TohZmULgXq_spL0Nu-VjU> )

**Projenin Geliştirilmesi :**

Sistemde dost-düşman tanıma sistemi olmadığı için tehlikeli durumlar söz konusu olabilir. Bunu önlemek amacıyla sistemi geliştirip dost-düşman sistemi eklenebilir.