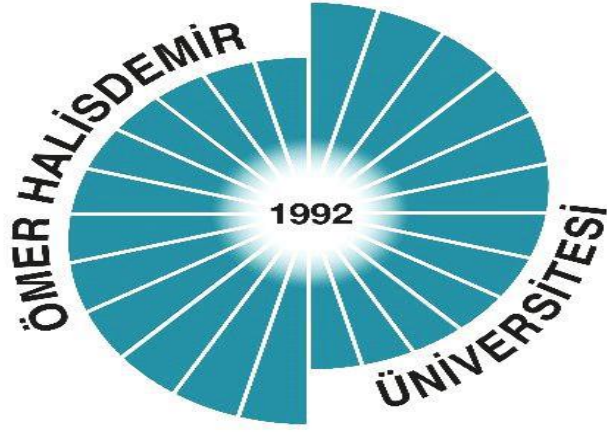




**ÖMER HALİSDEMİR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

TEZ ADI

HEDEF TAKİP VE BELİRLEME SİSTEMİ

HAZIRLAYAN

İMRAN GÜLSOY 130607045
FADİMANA ÜNLÜ 130607056
ÜMMÜ BETÜL ZENGİN 130607050

DANIŞMAN

Yrd. Doç. Dr. Mehmet Kürşat YALÇIN

NİĞDE 2017

T.C.

**ÖMER HALİSDEMİR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ
BİTİRME TEZİ KABUL VE ONAY BELGESİ**

Bölümümüz

.....
.....nu
maralı
öğrencileri
.....
.....
'nın,
.....başlıklı Bitirme Tezi çalışması aşağıdaki
jüri
üyeleri tarafındanMühendisliği Bölümü'nde Bitirme Tezi olarak Oy
Birliği/Oy Çokluğu ile kabul edilmiştir.

Danışman :

Üye :

Üye :

Tezin savunulduğu Tarih:

Bitirme Tezi dersi kapsamında yapılan bu çalışma, ilgili jüriler tarafından
değerlendirme sonucunda.....Mühendisliği Bölümü'nde Bitirme
Tezi çalışması olarak kabul edilmiştir...../...../.....

**İmza
Bölüm Başkanı**

DOĞRULUK BEYANI

Bitirme tezi olarak sunduğumuz bu çalışmayı tüm akademik kurallara ve Niğde Üniversitesi Yayın Etiği Komisyonu Yönergesi'ne uygun olarak gerçekleştirdiğimizi ve sunduğumuzu; bu kurallar ve ilkelere aykırı hiç bir yol ve yardıma başvurmaksızın bizzat hazırladığımızı beyan ederiz.

Tezimle ilgili yaptığım beyana aykırı bir durum saptanırsa ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımızı bildiririz. ... /.../...

İMİRAN GÜLSOY

FADİMANA ÜNLÜ

ÜMMÜ BETÜL ZENGİN

TEŞEKKÜR

Bu çalışma, Ömer Halisdemir Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği bölümünde Bitirme Tezi kapsamında hazırlanmıştır. Bu çalışmanın hazırlanması sırasında yardımlarını hiçbir zaman bizden esirgemeyen değerli hocamız Yrd. Doç. Dr. Mehmet Kürşat YALÇIN' a teşekkür ederiz.

Ayrıca çalışmalarımız sırasında bizden hiçbir zaman desteklerini esirgemeyen bölümümüz öğrencilerinden Davut YÜKSEKKAY'a , Muhammed Ali ÇALIŞIR ' a , makina bölümü öğrencisi Muhammed Onur OKTAY'a vebizden her türlü maddi ve manevi desteğini esirgemeyen ailelerimize teşekkür ederiz.

GİRİŞ

Bu projede daha önceden belirlenmiş bir cismin görüntü işleme algoritmaları yardımı ile tespit edilir. Cismin konumunu ve mesafesini tespit etmek için sistemin önüne yerleştirilen kameradan alınan görüntüler RASPBERRY Pİ de çalıştırılmaktadır. Proje kapsamında geliştirilen sistem daha önceden belirlenmiş bir cismin konumunu görüntü işleme ile belirleyip hedefi takip etmektedir. Bu sistem geliştirilerek askeri sistemlerde, araç takip sistemlerde vb. sistemlerde kullanılmaktadır. Daha önceden belirlenmiş sistemlerde hedef takip ve hedef tespitinde farklı çalışmalar bulunmaktadır.

Konu alınan projede hedef ve hedef konumu tespitinde görüntü işleme algoritmaları kullanılmaktadır. Görüntü işleme için OpenCV seçilmesinin temel sebebi OpenCV'nin bu konuda oldukça hızlı olmasıdır. Hedef daha önceden bilindiğinden, hedefin geometrik şeklinden merkez koordinatları baz alınarak hedef (düşman) belirlenmektedir. Hedef daire şeklindedir. Bu durumun sebebi diğer geometrik şekillere göre daha iyi bir sonuç elde etmemizi sağlamaktadır. Ayrıca kamera görüntü açısı içerisinde birden fazla cismin bulunması durumunda bize hedef tespit konusunda engel teşkil etmemektedir. Sistemimizde sorun teşkil eden durum ise eğer kamera görüntüsü içerisinde hedef tespit edilemez ise algoritma hata vermektedir. Hedefe ait veriler OpenCV'de elde edildikten sonra buradan ethernet kablosu yardımıyla arduinoda seri haberleşme elde edilip veriler servo motorumuza aktarılmaktadır. Kamera görüntüsü geri besleme olarak kullanılmaktadır. Hedef tam olarak görüntü karesinin ortasına denk gelene kadar sistem tek ekseninde gerekli olan yöne dönmektedir. Bu dönme işlemini de motorumuz gerçekleştirmektedir.

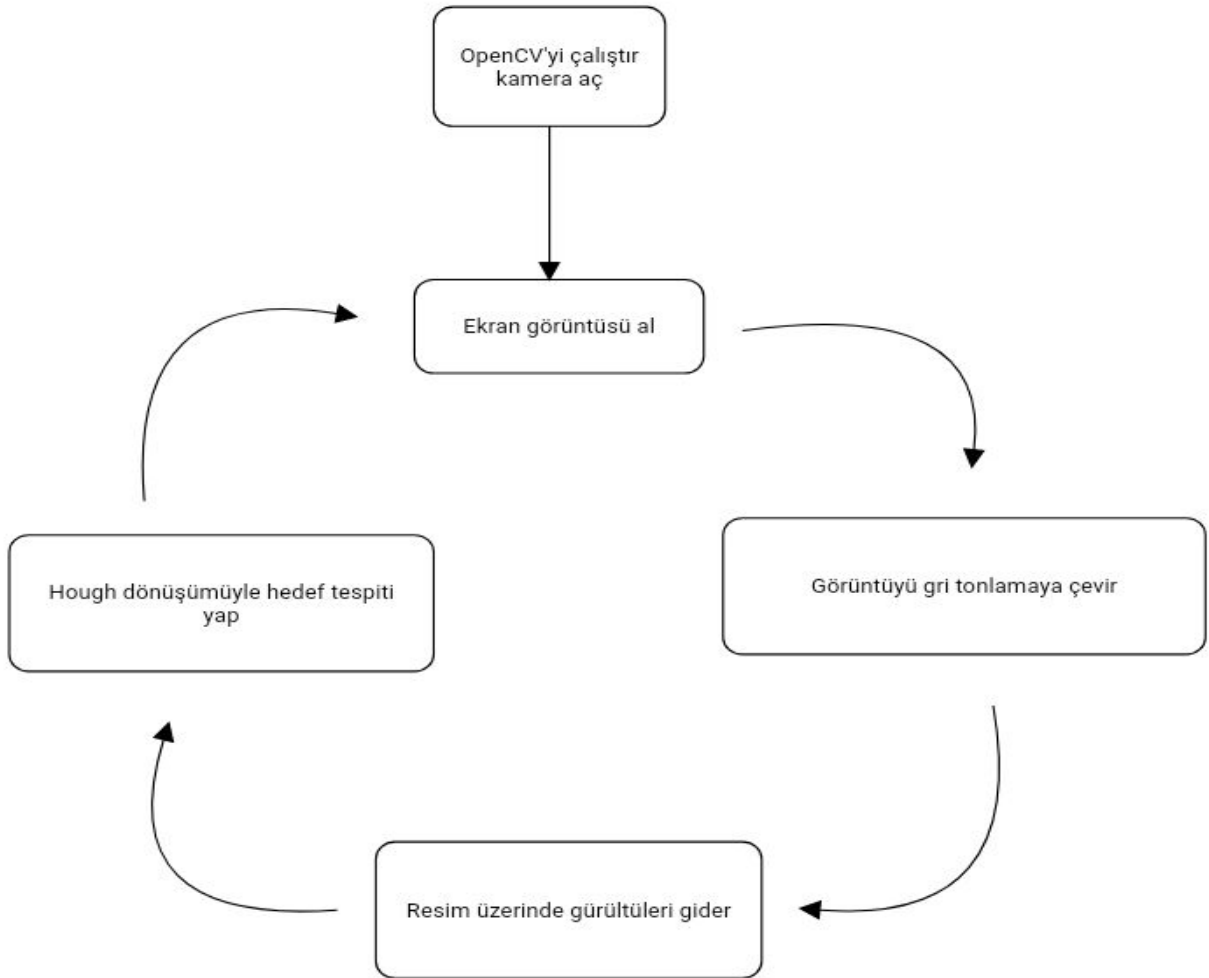
LİTERATÜR TARAMASI

Görüntü işleme sinyal işleme kapsamında yer alan ve hızlı gelişen bir alandır. Görüntü işleme sayısal bir resim haline getirilmiş olan gerçek yaşamdaki görüntülerin bir giriş resmi olarak işlenerek o resmin özelliklerinin ve görüntüsünün değiştirilmesi sonucunda yeni bir resim oluşturulmasıdır. On yıl önce Türkiye de bir Ar-Ge mühendisi için görüntü işleme hobi olarak photoshop programı ile birkaç iyileştirme yapmak dışında pek bir anlam ifade etmezdi. Lakin yıllar ilerledikçe görüntü işleme alanında yapılan çalışmalar geliştirilerek artış göstermiştir. Artık daha çok alanda kullanılmaya başlanmıştır. Bu kullanım alanlarından bazıları ise şunlardır:

- Sayısal görüntü işleme alanındaki örnek uygulamalar,
- Tasarım ve imalat uygulamaları,
- Savunma ve güvenlik uygulamaları,
- Tıp alanında görüntü inceleme,
- Mimari uygulamalar vb. birçok uygulamalarda kullanılmıştır.

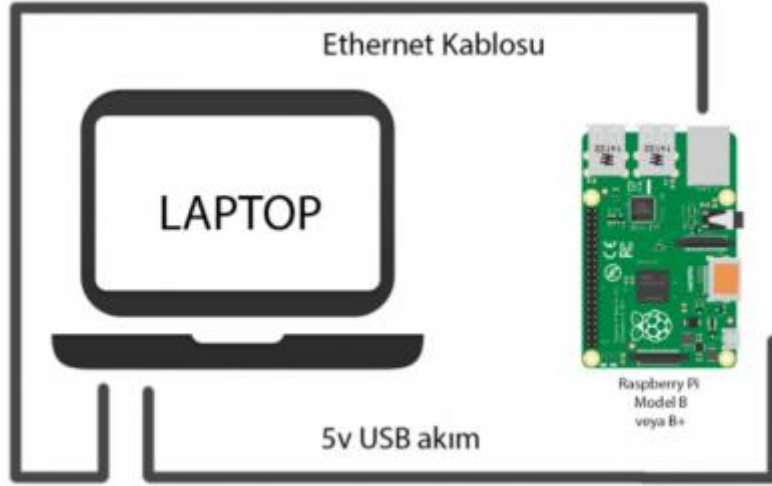
GÖRÜNTÜ İŞLEME

Görüntü işleme teknikleri ile , sayısal görüntüler iyileştirilerek nesne tanıma, hedef tanıma gibi işlemler gerçekleştirilebilmektedir. OpenCV, Intel tarafından geliştirilerek BSD lisansı ile lisanslanmış, “Bilgisayarla Göru/ Görme” kütüphanesidir. Özellikle gerçek zamanlı uygulamalar hedef alınarak geliştirilmiş olması, ticari kullanımı dahil ücretsiz olması ve Windows, Linux, MacOS X gibi farklı platformlarda kullanılabilmesi bu kütüphaneyi diğer görüntü işleme araçlarından bir adım öne çıkarması sebebiyle projemizde kullanılmıştır. Görüntü işlemede kullanılan akış şeması aşağıdaki gibidir.

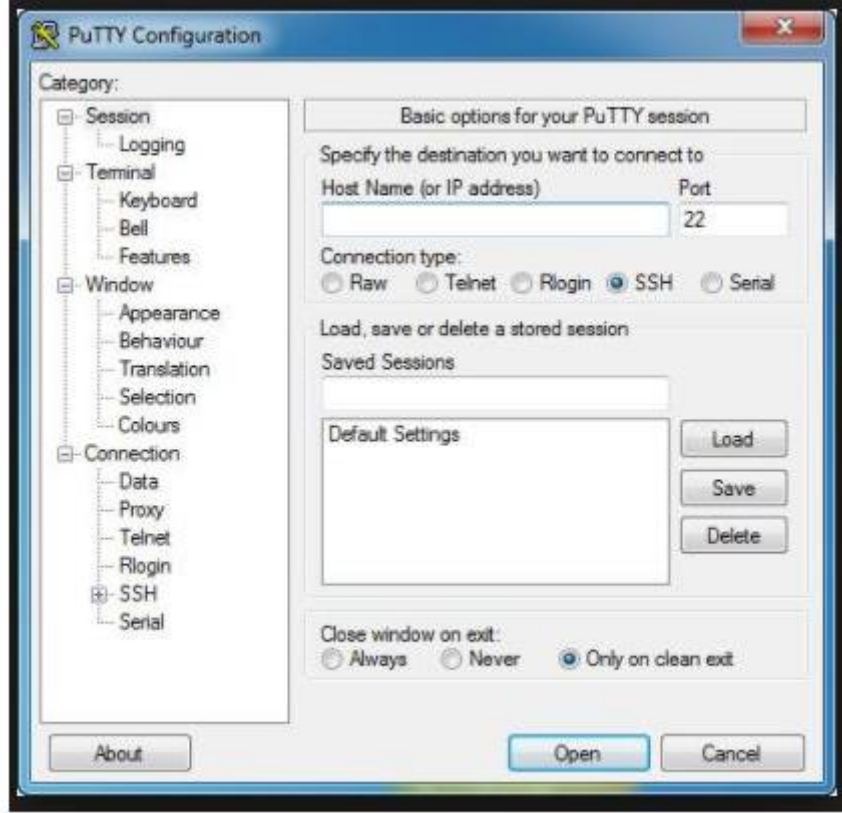


Akış şemasını açıklamadan önce Bazı programsal gereksinimleri yerine getirmemiz gerekmektedir.

Öncelikle RASPBERRY Pi'nin Ip adresine ihtiyaç vardır bunun için RASPBERRY Pi'ye ağ kablosu ile aşağıda şeması gösterildiği gibi doğrudan bağlanılıp ip adresi elde edildi.



RASPBERRY Pi'nin ip adresinin elde edilmesinin ardından RASPBERRY Pi'nin terminal ekranına SSH ile bağlanması gerekmektedir. Bu işlem PUTTY adlı program aracılığıyla sağlandı. PUTTY hakkında bilgi verecek olunur ise , PUTTY Unix/Linux sunuculara ve ağ cihazlarına kadar SSH, Telnet ve COM bağlantıları ile çalışan cihaz ve bilgisayarlara uzaktan oturum açarak işlemler yapmamızı sağlayan ücretsiz bir yazılımdır. Genellikle Windows işletim sisteminde SSH bağlantısı desteklenmediğinden dolayı SSH bağlantıları yapmak için kullanılmaktadır.



```
pi@raspberrypi-kBQN9wiMbS: ~  
login as: pi  
pi@192.168.43.243's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat May 27 06:15:04 2017  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
pi@raspberrypi-kBQN9wiMbS:~$
```

RASPBERRY Pİ'ye her seferinde ethernet kablosu aracılığıyla uzun işlemler sonucunda bağlanmak yerine kolaylık olması açısından telefonun hotspotu aracılığıyla bağlanması uygun görülmüştür. Bağlantının telefonun hotspotuyla yapılması için gerekli olan ayarlamalar şu şekildedir: Terminale şu komutu verip belgeyi düzenlemek:

```
pi@raspberrypi-kBQN9wiMbS: ~  
login as: pi  
pi@192.168.43.243's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat May 27 06:15:04 2017  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@raspberrypi-kBQN9wiMbS:~ $ sudo nano /etc/network/interfaces
```

Bundan sonra açılan belgeyi şu şekilde değiştireceğiz:

```
GNU nano 2.2.6 File: /etc/network/interfaces  
auto lo  
iface lo inet loopback  
  
#auto eth0  
#iface eth0 inet static  
#address 169.254.0.2  
#netmask 255.255.0.0  
#gateway 169.254.0.1  
  
allow-hotplug wlan0  
iface wlan0 inet manual  
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf  
    wpa-ssid "betul"  
    wpa-psk "Bb130607"  
allow-hotplug wlan1  
iface wlan1 inet manual  
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf  
  
[ Read 20 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^I To Spell
```

ctrl+x ve y diyip enter'a basıp komut satırına iniyoruz ve yeniden başlatıyoruz. Bu işlemler sonucunda RASPBERRY Pİ'ye her seferinde telefonumuzun hotspotuyla bağlantı yapılabilmesi sağlandı.

RASPBERRY Pİ ÜZERİNDE VNC SERVER BAĞLANTISI

VNC (Virtual Network Computing) temel anlamda RFB (Remote Framebuffer) protokolünün biraz gelişmiş platform bağımsız ve açık kaynak kodlu bir uygulaması. Proje uzun süre önce AT&T Cambridge Labs.'deki mühendisler tarafından başlatılmıştı ve bir süre sonra da BSD benzeri bir lisans ile açık kaynak kodu ile duyurularak gelişimini sürdürmüştü. Daha sonra AT&T'de VNC üzerine çalışan mühendislerden birkaçı bu kuruluştaki görevlerinden ayrılarak [RealVNC Ltd.](#) adındaki şirketi kurdular ve VNC kodundan türeterek geliştirdikleri VNC ürününü bu şirket içerisinde geliştirmeye devam ediyorlar. VNC, özetle herhangi bir ağ sunucusu üzerinde çalışan grafik arayüz uygulamalarına herhangi bir başka ağ üzerindeki bir bilgisayardan kolaylıkla ve hızla erişerek bu uygulamaları kullanabilmemizi ve yönetmemizi sağlayan platform , bağımsız bir yapıdır.

RASPBERRY Pİ nin masa üstü ekranına erişimi VNC server (Uzak masaüstü bağlantısı) yoluyla yapıldı. VNC server'ı windows üzerine kurmak için SSH bağlantısı gerekmektedir.

RASPBERRY Pİ ye VNC SERVER kurulumu için;

Öncelikle RASPBERRY Pİ'ye VNC server uygulaması indirildi bunun için

sudo apt-get install tightvncserver

Komutunu kullanıyoruz. Kurulum tamamlandığında da

Tightvncserver

yazarak uygulamamızı başlatıyoruz. VNC server uzak masaüstü bağlantısı olduğu için şifre ile korunması şarttır. Bu nedenle bir şifre girildi. Bu şifre view only password'tür. Bu şifreyi giren kişi ekranı görebilir ancak müdahale edemez.

```
pi@raspberrypi: ~  
auto mode  
update-alternatives: using /usr/bin/tightvncpasswd to provide /usr/bin/vncpasswd  
(vncpasswd) in auto mode  
Setting up x11-xserver-utils (7.7~3) ...  
Setting up xfonts-base (1:1.0.3) ...  
Processing triggers for menu ...  
root@raspberrypi:/home/pi# tightvncserver  
  
You will require a password to access your desktops.  
  
Password:  
Verify:  
Would you like to enter a view-only password (y/n)? y  
Password:  
Verify:  
xauth: file /root/.Xauthority does not exist  
  
New 'X' desktop is raspberrypi:1  
  
Creating default startup script /root/.vnc/xstartup  
Starting applications specified in /root/.vnc/xstartup  
Log file is /root/.vnc/raspberrypi:1.log  
root@raspberrypi:/home/pi#
```

“:” işareti ile ayrılan nokta ekran numarasıdır. Bu sayede birden fazla ekran başlatmak ve farklı farklı kullanmak mümkün olmaktadır.

Ekran boyutunu ve kalitesini ayarlamak için

vncserver :0 -geometry 1920×1080 -depth 24

komutu kullanıldı. Burada 1920×1080 yerine makul olan herhangi bir ekran boyutu yazılabilir. Bu ayar HD görüntü için seçilmiştir.

24 ise renk derinliğidir. Burada eğer ağı yormak istemiyorsa boyutu küçülterek daha hızlı bir bağlantı yapılması sağlanabilir. Sonuçta veri miktarı ekran boyutuyla ve kalitesiyle doğru orantılıdır.

Artık VNC server kullanılabilir. Ancak yeniden başlattığımızda tekrar programı başlatmamız ve ekran ayarını yapmamız gerekecekti. Bunun önüne geçmek için öncelikle sistem klasörlerine erişim için

sudo bash

komutunu kullanarak gerekli izinsaylandı

sonrasında değişiklik yapılacak dizine gidildi

cd /etc/init.d

burada

nano

yazarak editör açıldı.

Aşağıda verilen kod yazılmalıdı.

```
### BEGIN INIT INFO
# Provides: vncboot
# Required-Start: $remote_fs $syslog
# Required-Stop: $remote_fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start VNC Server at boot time
# Description: Start VNC Server at boot time.
### END INIT INFO

#! /bin/sh
# /etc/init.d/vncboot

USER=root
HOME=/root

export USER HOME

case "$1" in
  start)
    echo "Starting VNC Server"
    #Insert your favoured settings for a VNC session
    /usr/bin/vncserver :1 -geometry 1280x800 -depth 16 -pixelformat rgb565
    ;;
  stop)
    echo "Stopping VNC Server"
    /usr/bin/vncserver -kill :1
    ;;
  *)
    echo "Usage: /etc/init.d/vncboot {start|stop}"
    exit 1

```

sonrasında bu dosya kaydedildi.

chmod 755 /etc/init.d/vncboot

yazarak bu dosyanın sahip olduğu yetkiler değiştirildi. Burada 755 yetkinin cinsi.

update-rc.d vncboot defaults

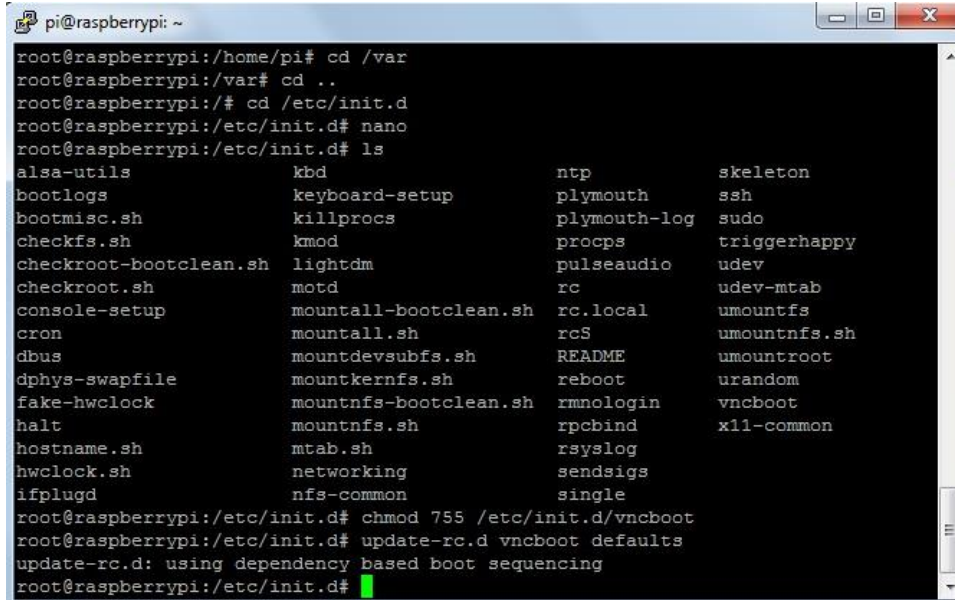
ile başlangıçta default olarak çalışması sağlandı. Burada aldığımız uyarı önemlidir. Eğer

update-rc.d: error: unable to read /etc/init.d//etc/init.d/vncboot

hatası alınırsa

update-rc.d /etc/init.d/vncboot defaults

komutu kullanılmalıdır. Bu hatanın oluşma nedeni farklı konumda bulunmamızdan kaynaklanır.

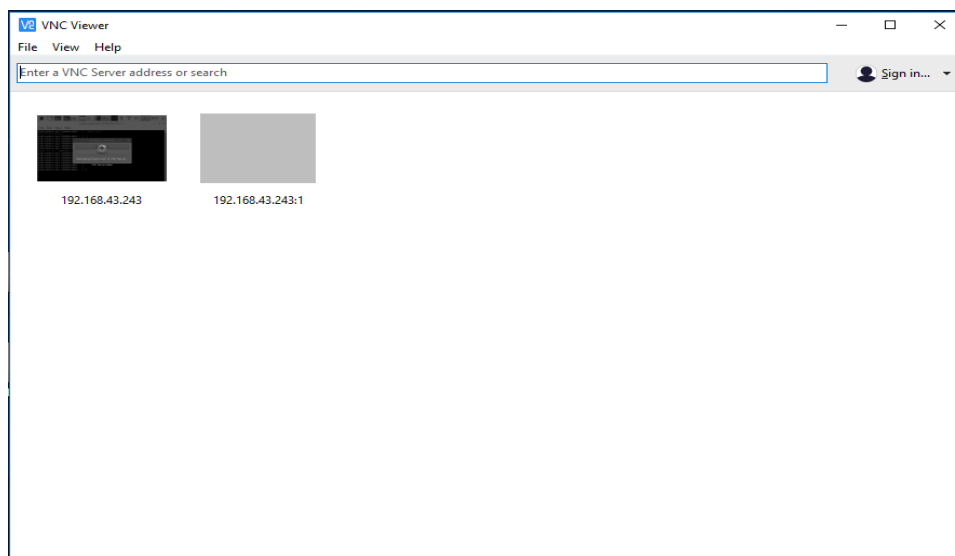


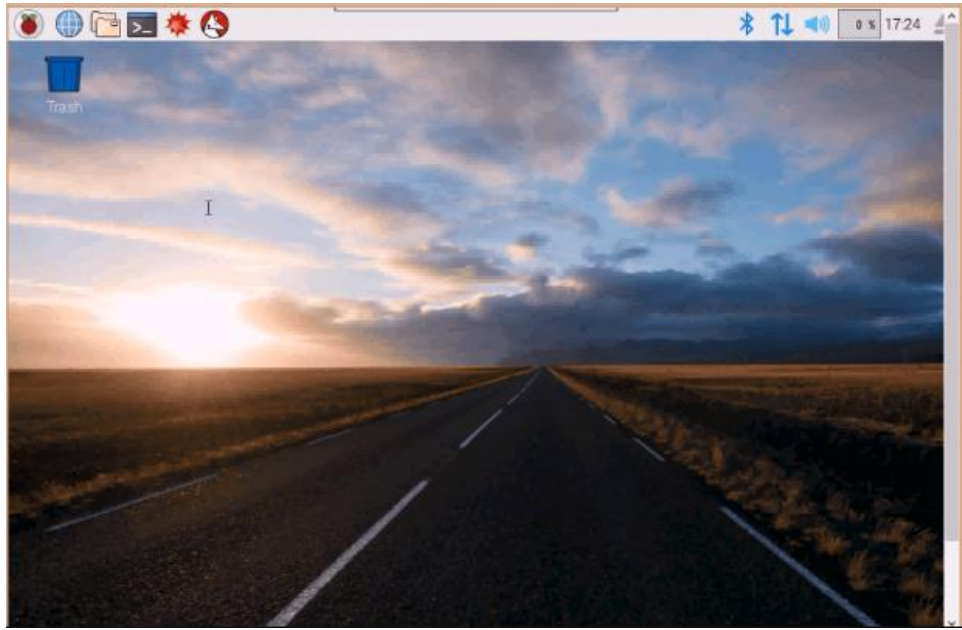
```
pi@raspberrypi: ~  
root@raspberrypi:/home/pi# cd /var  
root@raspberrypi:/var# cd ..  
root@raspberrypi:/# cd /etc/init.d  
root@raspberrypi:/etc/init.d# nano  
root@raspberrypi:/etc/init.d# ls  
alsa-utils          kbd                  ntp                  skeleton  
bootlogs            keyboard-setup      plymouth            ssh  
bootmisc.sh         killprocs           plymouth-log        sudo  
checkfs.sh          kmod                procps              triggerhappy  
checkroot-bootclean.sh lightdm             pulseaudio          udev  
checkroot.sh        motd                rc                  udev-mtab  
console-setup       mountall-bootclean.sh rc.local            umountfs  
cron                mountall.sh         rcS                 umountnfs.sh  
dbus                mountdevsubfs.sh   README             umountroot  
dphys-swapfile      mountkernfs.sh     reboot              urandom  
fake-hwclock        mountnfs-bootclean.sh rmnologin           vncboot  
halt                mountnfs.sh         rpcbind             x11-common  
hostname.sh         mtab.sh            rsyslog  
hwclock.sh          networking          sendsigs  
ifplugd             nfs-common          single  
root@raspberrypi:/etc/init.d# chmod 755 /etc/init.d/vncboot  
root@raspberrypi:/etc/init.d# update-rc.d vncboot defaults  
update-rc.d: using dependency based boot sequencing  
root@raspberrypi:/etc/init.d#
```

shutdown -r now

Komutuyla cihaz yeniden başlatılarak test edildi.

Aşağıda görüldüğü üzere windowsta kurulmuş olan VNC SERVER açılarak RASPBERRY Pİ nin ipsi girilmesi suratiyle RASPBERRY Pİ masaüstü ekranına başarıyla ulaşım sağlandı.





RASPBERRY Pİ'YE OPEN CV'NİN KURULUMU

Öncelikle gerekli bağlantıları kurmamız gerekti. Bağlantılar da birbirine bağlı olduğundan iki aşamada gerçekleştirildi:

1	<code>sudo apt-get -y install build-essential cmake cmake-qt-gui pkg-config libpng12-0 libpng12-dev libpng++-dev libpng3 libpnglite-dev zlib1g-dbg zlib1g zlib1g-dev pngtools libtiff4-dev libtiff4 libtiffxx0c2 libtiff-tools</code>
1	<code>sudo apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg libjpeg-progs ffmpeg libavcodec-dev libavcodec53 libavformat53 libavformat-dev libgstreamer0.10-0-dbg libgstreamer0.10-0 libgstreamer0.10-dev libxine1-ffmpeg libxine-dev libxine1-bin libunicap2 libunicap2-dev libdc1394-22-dev libdc1394-22 libdc1394-utils swig libv4l-0 libv4l-dev python-numpy libpython2.6 python-dev python2.6-dev libgtk2.0-dev pkg-config</code>

En güncel OpenCV kaynak kodunu indirildi

1	<code>wget http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.11/opencv-2.4.11.zip</code>
---	---

Şimdi de OpenCV'yi açmak aşağıdakiler yazıldı:

1	<code>unzip opencv-2.4.11.zip</code>
2	<code>rm opencv-2.4.11.zip</code>
3	<code>cd opencv-2.4.11</code>
4	<code>mkdir build</code>
5	<code>cd build</code>

CMAKE ile buildi oluşturmak için terminalden aşağıdaki komutu vererek standart ayarları ile build oluşturuldu:

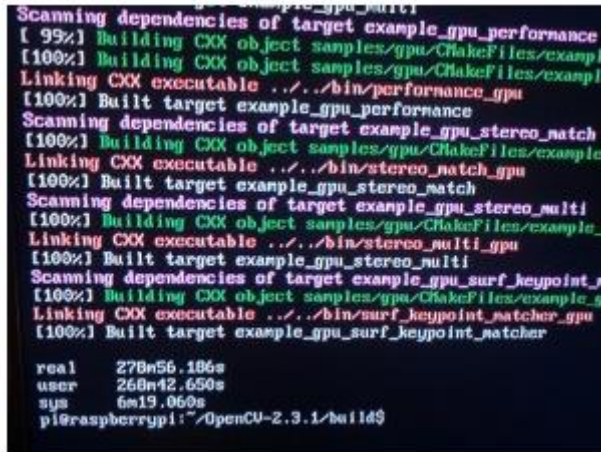
1	<code>cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D BUILD_PYTHON_SUPPORT=ON -D BUILD_EXAMPLES=ON ..</code>
---	--

Alternatif olarak gui arayüzünden de yapabilmek için cmake-gui komutunu kullanıldı. Terminale aşağıdaki komut girildi:

1	<code>cmake-gui ..</code>
---	---------------------------

Daha sonra gui arayüzünden configure'a basarak ve de dahil olmasını istediğimiz bileşenleri seçip yeniden 'configure'a basarak logları da kontrol edip bulunamayan modül olmadığından emin olundu, şimdi derlemeye ve kurulumu hazır.

1	<code>make</code>
2	<code>sudo make install</code>



```
Scanning dependencies of target example_gpu_performance
[ 99%] Building CXX object samples/gpu/CMakeFiles/example_gpu_performance
[100%] Building CXX object samples/gpu/CMakeFiles/example_gpu_performance
Linking CXX executable ../../bin/performance_gpu
[100%] Built target example_gpu_performance
Scanning dependencies of target example_gpu_stereo_match
[100%] Building CXX object samples/gpu/CMakeFiles/example_gpu_stereo_match
[100%] Building CXX object samples/gpu/CMakeFiles/example_gpu_stereo_match
Linking CXX executable ../../bin/stereo_match_gpu
[100%] Built target example_gpu_stereo_match
Scanning dependencies of target example_gpu_stereo_multi
[100%] Building CXX object samples/gpu/CMakeFiles/example_gpu_stereo_multi
[100%] Building CXX object samples/gpu/CMakeFiles/example_gpu_stereo_multi
Linking CXX executable ../../bin/stereo_multi_gpu
[100%] Built target example_gpu_stereo_multi
Scanning dependencies of target example_gpu_surf_keypoint_matcher
[100%] Building CXX object samples/gpu/CMakeFiles/example_gpu_surf_keypoint_matcher
[100%] Building CXX object samples/gpu/CMakeFiles/example_gpu_surf_keypoint_matcher
Linking CXX executable ../../bin/surf_keypoint_matcher_gpu
[100%] Built target example_gpu_surf_keypoint_matcher

real    278m56.186s
user    268m42.650s
sys     6m19.060s
pi@raspberrypi:~/OpenCV-2.3.1/build$
```

Derleme ve kurulum sonrası ayar dosyasında birkaç değişikliğe gitmemiz gerekti: Öncelikle terminalden şu komutu girerek düzenleme sayfasını açıldı:

1	<code>sudo nano /etc/ld.so.conf.d/opencv.conf</code>
---	--

en altına dosyanın şu komutu eklendi:

1	<code>/usr/local/lib</code>
---	-----------------------------

Daha sonra şu komutu çalıştırıp sistem genelinde bashrc ye eklendi:

1	<code>sudo nano /etc/bash.bashrc</code>
---	---

Açılan sayfada en altına şu komutu eklendi:

1	<code>PKG_CONFIG_PATH=\$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig export PKG_CONFIG_PATH</code>
---	--

Artık konfigürasyon hazır.Başarıyla OpenCV yüklendi.

Yapılan işlemler sonucunda OpenCVde görüntü işleme kısmına geçmek için gereken programsal gereksinimler yerine getirilmiş durumdadır.

HOUGH DÖNÜŞÜMÜ İLE DAİRESEL ŞEKİL TESPİTİ

Sayısal görüntülerde geometrik şekil bulmanın görüntü işlemede birçok uygulama alanı bulunmaktadır. İris bulma, plaka bulma, saha üzerinde top bulma ve benzeri uygulamalar buna örnek olarak verilebilir.

Bir kamera veya benzeri bir algılayıcı yardımı ile elde edilen görüntülerdeki şekillerin her zaman eksiksiz yer alması mümkün olmamaktadır ve şekillerdeki kopukluklar şekil tespitini zorlaştırmaktadır. Bu noktada hough dönüşümü ile görüntünün tamamının görülebilir olmadığı durumlarda da olası şekiller tespit edilebilmektedir. Bu projede şekil bulmada kullanılan hough dönüşümünün dairesel şekil(hedef) tespitinde kullanımı üzerinde durulmuştur. Bir [resimdeki daireleri algılamak](#) için OpenCV fonksiyonu olan [HoughCircles](#) kullanıldı.

Hough dönüşümü temelde kenarların olası geometrik şekilleri oylaması mantığı ile çalışmaktadır. Hough dönüşümü kullanılarak şekil tespiti genel olarak aşağıdaki adımlar ile özetlenebilir:

Çizgi algılama durumu bir satır, iki parametre tarafından tanımlandı (r, θ) . Dairesel durumda, bir daire tanımlamak için üç parametre gerekir:

$C : (x_{center}, y_{center}, r)$

Burada (x_{center}, y_{center}) merkez konumu r bir daireyi tamamen tanımlamamızı sağlayan yarıçaptır:

- Videodan görüntü alınır.
- Gri ölçeğe çevrilir.
- Gürültüyü azaltmak ve yanlış daire tespitini önlemek için görüntü bulanıklaştırılır.
- *Hough Circle Transform* bulanık görüntüye uygulanır .
- Algılanan dairelerin çevresi çizdirilir.(Çemberler kırmızıya ve orta noktalar küçük bir yeşil nokta ile çizdirildi.)
- Algılanan daire (ler) gösterilir.

Uygulamalarda yalnızca istenen yarıçap aralığındaki çemberlerin aranması ile hem istenmeyen şekillerin bulunması önlenmiş hem de başarımda iyileştirme sağlanmıştır.

Tüm bunların sonucunda OpenCV çalıştırılarak RASPBERRY Pİ'ye bağlı kamera modülünden gayet başarılı bir şekilde daire (hedef) tespiti sağlanmış olundu.

GÖRÜNTÜ İŞLEME YAZILIMI YARDIMIYLA NESNE TAKİBİ

OpenCV ile algılanan hedefin kamera görüntüsü düzleminde koordinatları belirlenmiştir. Belirlenen koordinatlar gönderilmiştir. Bu koordinatları USB ile göndermemiz için öncelikle bir mantık kontrolüyle değerlerin değişip değişmeyeceğini kontrol etmemiz gerekir. Eğer koordinat değişkenleri değişmiyorsa servo motorlarımız hareket etmeyeceğinden bu bilginin sürekli gönderilmesine gerek yoktur. Yani hedefimiz hareketsiz konumdaysa hedef koordinatları gönderilmeyecektir. Hedefimiz ne zaman hareket etmeye başlarsa hareketin algılandığı bölgenin koordinat bilgileri bir değişkenlerine atanacağından bu bilgileri USB aracılığıyla Arduino'ya ve Arduino ile servo motorlara gönderilecektir.

Bir başka husus ise servo motorumuzun 360 derece dönme kabiliyeti vardır. Ancak biz görüş açımız 120 derece olduğundan 0-120 arası değerler almasını sağladık. Kamera çözünürlüğü X'de 640 ve Y'de 480 dir. Aldığımız bu bilgiler Arduino'ya bağlı servo motora iletilir. Gönderilen koordinat bilgileri servo için her 120 pils 20 derece olacak şekilde dönme hareketi yapar. Servo motorlara gelen koordinat bilgisi belirlediğimiz eşik sınırında kamerayı y koordinatında 220'den küçükse yukarı , 260'dan büyük ise aşağı yönde servo moturu döndürür. Servo motorumuzun Arduino geliştirme kartı aracılığıyla kontrolü kısaca bu şekildedir.



OpenCV kodları:

```
#define PID_H
class PID
{
private:
    float u;
    float kP;
    float kI;
    float kD;
    float error_1;
    float uP;
    float uI;
    float uD;
    boolean pidEnabled;
    float outMax;
    float outMin;
public:
    PID(float kp = 0, float ki = 0, float kd = 0)
    {
        setKp(kp);
        setKi(ki);
        setKd(kd);
        pidEnabled = false;
        outMax = 255;
        outMin = -255;
        initialize();
    }
    float compute(float error, float deltaT)
    {
        if (!pidEnabled) return 0;
        //P
        uP = kP * error;
        //
        //I
        uI += kI * error * deltaT / 1000.0;
        //Integral
        //Integral += error * deltaT / 1000.0;
        //uI = kI * Integral;
        /*if (Integral > integralMax)
            uI = outMax;
        else if (uI < outMin)
            uI = outMin;*/
        if (uI > outMax)
            uI = outMax;
        else if (uI < outMin)
            uI = outMin;
        //D
        uD = kD * (error - error_1) * 1000.0 / deltaT;
```

```

    error_1 = error;
    //
    u = uP + ul + uD;
    if (u > outMax)
    {
        u = outMax;
    }
    else if (u < outMin)
    {
        u = outMin;
    }
    return u;
}
float getU()
{
    return u;
}
void initialize()
{
    u = 0;
    ul = 0;
    error_1 = 0;
}
void enablePID()
{
    initialize();
    pidEnabled = true;
}
void disablePID()
{
    pidEnabled = false;
    u = 0;
}
void setKp(float kp)
{
    if (kp < 0) return;
    kP = kp;
}
void setKi(float ki)
{
    if (ki < 0) return;
    kI = ki;
}
void setKd(float kd)
{
    if (kd < 0) return;
    kD = kd;
}
#endif
#ifdef __linux__ || defined(LINUX) || defined(__APPLE__) || defined(ANDROID)
#include <opencv2/imgproc/imgproc.hpp> // Gaussian Blur
#include <opencv2/core/core.hpp> // Basic OpenCV structures (cv::Mat, Scalar)

```

```

#include <opencv2/videoio/videoio.hpp>
#include <opencv2/highgui/highgui.hpp> // OpenCV window I/O
#include <opencv2/features2d/features2d.hpp>
#include <opencv2/objdetect/objdetect.hpp>
#include <stdio.h>
#include <string>
#include <vector>
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <termios.h>
#include <unistd.h>
#include "opencv2/imgcodecs.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <iostream>
#define DISPLAY_STRING
int set_interface_attribs(int fd, int speed)
{
    struct termios tty;
    if (tcgetattr(fd, &tty) < 0) {
        printf("Error from tcgetattr: %s\n", strerror(errno));
        return -1;
    }
    cfsetospeed(&tty, (speed_t)speed);
    cfsetispeed(&tty, (speed_t)speed);
    tty.c_cflag |= (CLOCAL | CREAD); /* ignore modem controls */
    tty.c_cflag &= ~CSIZE;
    tty.c_cflag |= CS8; /* 8-bit characters */
    tty.c_cflag &= ~PARENB; /* no parity bit */
    tty.c_cflag &= ~CSTOPB; /* only need 1 stop bit */
    tty.c_cflag &= ~CRTSCTS; /* no hardware flowcontrol */
    /* setup for non-canonical mode */
    tty.c_iflag &= ~(IGNBRK | BRKINT | PARMRK | ISTRIP | INLCR | IGNCR | ICRNL | IXON);
    tty.c_lflag &= ~(ECHO | ECHONL | ICANON | ISIG | IEXTEN);
    tty.c_oflag &= ~OPOST;
    /* fetch bytes as they become available */
    tty.c_cc[VMIN] = 1;
    tty.c_cc[VTIME] = 1;
    if (tcsetattr(fd, TCSANOW, &tty) != 0) {
        printf("Error from tcsetattr: %s\n", strerror(errno));
        return -1;
    }
    return 0;
}
void set_mincount(int fd, int mcount)
{
    struct termios tty;
    if (tcgetattr(fd, &tty) < 0) {

```

```

    printf("Error tcgetattr: %s\n", strerror(errno));
    return;
}
tty.c_cc[VMIN] = mcount ? 1 : 0;
tty.c_cc[VTIME] = 5;    /* half second timer */
if (tcsetattr(fd, TCSANOW, &tty) < 0)
    printf("Error tcsetattr: %s\n", strerror(errno));
}
using namespace std;
using namespace cv;
const string WindowName = "Face Detection example";
class CascadeDetectorAdapter: public DetectionBasedTracker::IDetector
{
public:
    CascadeDetectorAdapter(cv::Ptr<cv::CascadeClassifier> detector):
        IDetector(),
        Detector(detector)
    {
        CV_Assert(detector);
    }
void detect(const cv::Mat &Image, std::vector<cv::Rect> &objects)
{
    Detector->detectMultiScale(Image, objects, scaleFactor, minNeighbours, 0, minObjSize,
maxObjSize);
}
virtual ~CascadeDetectorAdapter()
{}
private:
    CascadeDetectorAdapter();
    cv::Ptr<cv::CascadeClassifier> Detector;
};
int main(int, char** )
{
    char *portname = "/dev/ttyACM0";
    int fd;
    int wlen;
    fd = open(portname, O_RDWR | O_NOCTTY | O_SYNC);
    if (fd < 0) {
        printf("Error opening %s: %s\n", portname, strerror(errno));
        return -1;
    }
    /*baudrate 115200, 8 bits, no parity, 1 stop bit */
    set_interface_attribs(fd, B115200);
    //set_mincount(fd, 0);          /* set to pure timed read */
    usleep(3000000);
    /* simple output */
    wlen = write(fd, "1 29 1\n", 7);
    if (wlen != 7) {
        printf("Error from write: %d, %d\n", wlen, errno);
    }
    tcdrain(fd); /* delay for output */
}

```

```

    wlen = write(fd, "1 30 1\n", 7);
    if (wlen != 7) {
        printf("Error from write: %d, %d\n", wlen, errno);
    }
    tcdrain(fd); /* delay for output */
    namedWindow(WindowName);
    VideoCapture VideoStream(0);
    if (!VideoStream.isOpened())
    {
        printf("Error: Cannot open video stream from camera\n");
        return 1;
    }
    std::string cascadeFrontalfilename = "../data/lbpcascades/lbpcascade_frontalface.xml";
    cv::Ptr<cv::CascadeClassifier> cascade =
    makePtr<cv::CascadeClassifier>(cascadeFrontalfilename);
    cv::Ptr<DetectionBasedTracker::IDetector> MainDetector =
    makePtr<CascadeDetectorAdapter>(cascade);
    cascade = makePtr<cv::CascadeClassifier>(cascadeFrontalfilename);
    cv::Ptr<DetectionBasedTracker::IDetector> TrackingDetector =
    makePtr<CascadeDetectorAdapter>(cascade);
    DetectionBasedTracker::Parameters params;
    DetectionBasedTracker Detector(MainDetector, TrackingDetector, params);
    if (!Detector.run())
    {
        printf("Error: Detector initialization failed\n");
        return 2;
    }
    Mat ReferenceFrame;
    Mat GrayFrame;
    vector<Rect> Faces;
    Mat dst;
    while(true)
    {
        //VideoStream >> ReferenceFrame;
        VideoStream >> dst;
        flip(dst, ReferenceFrame, 0);
        cvtColor(ReferenceFrame, GrayFrame, COLOR_RGB2GRAY);
        //Detector.process(GrayFrame);
        //Detector.getObjects(Faces);
        for (size_t i = 0; i < Faces.size(); i++)
        {
            rectangle(ReferenceFrame, Faces[i], Scalar(0,255,0));
        }
        medianBlur(ReferenceFrame, ReferenceFrame, 5);
        //cvtColor(ReferenceFrame, cimg, COLOR_GRAY2BGR);
        vector<Vec3f> circles;
        HoughCircles(GrayFrame, circles, HOUGH_GRADIENT,1, 10,
            100, 30, 15, 20 // change the last two parameters
            // (min_radius & max_radius) to detect larger circles
        );
        for( size_t i = 0; i < circles.size(); i++ )

```

```

{
    Vec3i c = circles[i];
    circle( ReferenceFrame, Point(c[0], c[1]), c[2], Scalar(0,0,255), 3, LINE_AA);
    cout << c[0] << " " << c[1] << " radius: " << c[2] << endl;
    circle( ReferenceFrame, Point(c[0], c[1]), 2, Scalar(0,255,0), 3, LINE_AA);
    if(c[1] > 260)
    {
        wlen = write(fd, "1 3 -5\n", 7);
        if (wlen != 7) {
            printf("Error from write: %d, %d\n", wlen, errno);
        }
        tcdrain(fd); /* delay for output */
    }

    if(c[1] < 220)
    {
        wlen = write(fd, "1 3 5\n", 6);
        if (wlen != 6) {
            printf("Error from write: %d, %d\n", wlen, errno);
        }
        tcdrain(fd); /* delay for output */
    }
}
imshow(WindowName, ReferenceFrame);
if (waitKey(30) >= 0) break;
}
//Detector.stop();
return 0;
}
#else
#include <stdio.h>
int main()
{
    printf("This sample works for UNIX or ANDROID only\n");
    return 0;
}
#endif

```

PID kodları:

```

#define PID_H
class PID
{
private:
    float u;
    float kP;
    float kI;

```

```

float kD;
float error_1;
float uP;
float uI;
float uD;
boolean pidEnabled;
float outMax;
float outMin;
public:
PID(float kp = 0, float ki = 0, float kd = 0)
{
    setKp(kp);
    setKi(ki);
    setKd(kd);
    pidEnabled = false;
    outMax = 255;
    outMin = -255;
    initialize();
}
float compute(float error, float deltaT)
{
    if (!pidEnabled) return 0;
//P
    uP = kP * error;
    //
//I
    uI += ki * error * deltaT / 1000.0;
    //Integral
    //Integral += error * deltaT / 1000.0;
    //uI = ki * Integral;
/*if (Integral > integralMax)
    uI = outMax;
else if (uI < outMin)
    uI = outMin;*/
if (uI > outMax)
    uI = outMax;
else if (uI < outMin)
    uI = outMin;

//D
    uD = kD * (error - error_1) * 1000.0 / deltaT;
    error_1 = error;
    //
u = uP + uI + uD;
if (u > outMax)
{
    u = outMax;
}
else if (u < outMin)
{
    u = outMin;
}

```

```

    }
    return u;
}
float getU()
{
    return u;
}
void initialize()
{
    u = 0;
    ul = 0;
    error_1 = 0;
}
void enablePID()
{
    initialize();
    pidEnabled = true;
}
void disablePID()
{
    pidEnabled = false;
    u = 0;
}
void setKp(float kp)
{
    if (kp < 0) return;
    kP = kp;
}
void setKi(float ki)
{
    if (ki < 0) return;
    kI = ki;
}
void setKd(float kd)
{
    if (kd < 0) return;
    kD = kd;
}
float getKp()
{
    return kP;
}
float getKi()
{
    return kI;
}
float getKd()
{
    return kD;
}
float getUp()

```

```

    {
        return uP;
    }
float getUi()
{
    return ul;
}
float getUd()
{
    return uD;
}
boolean getPIDStatus()
{
    return pidEnabled;
}
void setOutMinMax(float outmin, float outmax)
{
    if (outmin < outmax)
    {
        outMin = outmin;
        outMax = outmax;
    }
}
};
#endif // PID_H

```

ARDUİNO kodları:

```

/*
  DAYKY-TECH Motor Driver Firmware
  Written by Davut Yuksekkaya and M. Kursat Yalcin
*/
#include <PinChangeInt.h>
#include "PID.h"
#include "DEFINITIONS.h"
int PWMpin = 5; //Speed control
int IN1 = 6; //Direction
int IN2 = 7; //Direction
int STBY = 4; //standby
#define encPinA 2 //If you change enc pins, you have to change the codes in the
corresponding isr
#define encPinB 3
#define LOG_SIZE_MAX 200
#define LOG_POSITION 0
#define LOG_CURRENT 1
//defns for the state of the driver
#define DISABLED 0
#define ENABLED 1
volatile long encoderPosition = 0;
long posRef = 0;

```

```

long currentPosition;
long prevPosition = 0;
float currentVel;
float prevVel = 0;
float velFilterInput;
float velRef = 0;
long profPosRelRef = 0;
//long profPosAbsRef = 0;
boolean logEnable = false;
boolean logWhat = LOG_POSITION;
boolean sendLogToMatlab = false;
unsigned int logCounter = 0;
unsigned int logSize = LOG_SIZE_MAX;
unsigned long logForMatlab[LOG_SIZE_MAX];
unsigned long currentTime;
unsigned long oldTime = 0;
unsigned long deltaT;
byte loopRate = 3; //delta t for PID calc. in ms
unsigned int printTick = 0;
boolean longReportPos = false;
boolean longReportVel = false;
boolean shortReportPos = false;
boolean shortReportVel = false;
#define MAX_SERIAL_INT_DATA 5
float serialFloatData[MAX_SERIAL_INT_DATA];
int serialFloatIndex = 0;
PID posPID(15.0, 3.0, 0.0); //kP, kI, kD
//PID posPID(4.0, 5.0, 0.05); //kP, kI, kD
PID velPID(0.03, 0.05, 0.0);
boolean userPosDir = true;
boolean motorDir = false;
float u = 0;
float uManual = 0;
long printingPeriod = 1000;
float coeff[4];
unsigned long profInitialTime;
long profInitialPos;
long profFinalPos;
float profTimeInSec = 0;
float relTime;
boolean profPositionModeFired = false;
#define MAX_PROF_ACCEL 80000.0 //Enc pulse per second asuare
#define MAX_PROF_VEL 26000.0 //Enc pulse per second
boolean driverState = DISABLED;
int operatingMode = MANUAL_MODE;
int serialDataAvailable = 0;
unsigned short int rs485id = 0;
void setup (void)
{
    pinMode(STBY, OUTPUT);
    digitalWrite(STBY, HIGH);

```

```

    pinMode(PWMPin, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    //Home and Limit Switch Inputs
    //Serial Port Settings
    Serial.begin (115200);
    Serial.setTimeout(100);
    rs485id = 1;
    Serial.print(rs485id);
    Serial.println("!");
    //Pullups are enabled for Encoder Inputs to prevent miscountings
    pinMode(encPinA, INPUT_PULLUP);
    pinMode(encPinB, INPUT_PULLUP);
    //Interrupt Settings for pin change
    attachPinChangeInterrupt(encPinA, isrForEncPinA, CHANGE);
    attachPinChangeInterrupt(encPinB, isrForEncPinB, CHANGE);
} // end of setup
void loop (void)
{
    currentTime = millis();
    deltaT = currentTime - oldTime;
    if (deltaT >= loopRate)
    {
        oldTime = currentTime;
        printTick++;
    }
    //-----
    //Position Measurement
    currentPosition = getEncoderPosition();
    //Velocity Measurement
    velFilterInput = (float)(currentPosition - prevPosition) / (float)deltaT * 1000; //encPulse per
second
    prevPosition = currentPosition;
    //velocity filtering
    float alpha = 0.2;
    currentVel = alpha * velFilterInput + (1 - alpha) * prevVel;
    prevVel = currentVel;
    if (driverState == ENABLED) {
        switch (operatingMode)
        {
            case MANUAL_MODE:
                u = uManual;
                break;
            case POSITION_MODE:
                u = posPID.compute((float)(posRef - currentPosition), (float) deltaT);
                break;
            case PROFILE_POSITION_MODE :
                if (profPositionModeFired)
                {
                    relTime = (float)currentTime - (float)profInitialTime;
                    if (relTime < profTimeInSec * 1000)
                        posRef = evalPos(coeff, relTime / 1000);
                }
            }
        }
    }
}

```

```

        else
        {
            profPositionModeFired = false;
            posRef = profFinalPos;
            Serial.println("!");
        }
    }
    u = posPID.compute((float)(posRef - currentPosition), (float) deltaT);
    break;
case VELOCITY_MODE:
    u = velPID.compute(velRef - currentVel, (float) deltaT);
    break;
case PROFILE_VELOCITY_MODE:
    //Do nothing
    break;
case CURRENT_MODE:
    //Do nothing
    break;
case HOMING_MODE:
    //Do nothing
    break;
}
}
else
{
    u = 0;
}
uToPWM(u);
if (logEnable)
{
    switch (logWhat)
    {
        case LOG_POSITION:
            logForMatlab[logCounter] = (unsigned long)getEncoderPosition();
            break;
        case LOG_CURRENT:
            break;
    }
    if (++logCounter == logSize)
    {
        logEnable = false;
        logCounter = 0;
    }
}
}
if (printTick >= (printingPeriod / loopRate)) //printing period default 1000 ms
{
    printTick = 0;
    if (longReportVel)
    {
        Serial.print("ID ");
    }
}

```

```

Serial.print(rs485id);
Serial.print(" opMode ");
Serial.print(operatingMode);
Serial.print(" driverState ");
Serial.print(driverState);
Serial.print(" velRef ");
Serial.print(velRef);
Serial.print(" vel ");
Serial.print(currentVel);
Serial.print(" error ");
Serial.print(velRef - currentVel);
Serial.print(" u_p ");
Serial.print(velPID.getUp());
Serial.print(" u_i ");
Serial.print(velPID.getUi());
Serial.print(" u_d ");
Serial.print(velPID.getUd());
Serial.print(" u ");
Serial.print(velPID.getU());
Serial.print(" uManual ");
Serial.print(uManual);
Serial.print(" deltaT ");
Serial.print(deltaT);
Serial.print(" Kp ");
Serial.print(velPID.getKp());
Serial.print(" Ki ");
Serial.print(velPID.getKi());
Serial.print(" Kd ");
Serial.println(velPID.getKd());
}
if (longReportPos)
{
    Serial.print("ID ");
    Serial.print(rs485id);
    Serial.print(" opMode ");
    Serial.print(operatingMode);
    Serial.print(" driverState ");
    Serial.print(driverState);
    Serial.print(" posRef ");
    Serial.print(posRef);
    Serial.print(" pos ");
    Serial.print(getEncoderPosition());
    Serial.print(" error ");
    Serial.print(posRef - getEncoderPosition());
    Serial.print(" u_p ");
    Serial.print(posPID.getUp());
    Serial.print(" u_i ");
    Serial.print(posPID.getUi());
    Serial.print(" u_d ");
    Serial.print(posPID.getUd());
    Serial.print(" u ");
}

```

```

    Serial.print(posPID.getU());
    Serial.print(" uManual ");
    Serial.print(uManual);
    Serial.print(" deltaT ");
    Serial.print(deltaT);
    Serial.print(" Kp ");
    Serial.print(posPID.getKp());
    Serial.print(" Ki ");
    Serial.print(posPID.getKi());
    Serial.print(" Kd ");
    Serial.println(posPID.getKd());
}
if (shortReportPos)
{
    Serial.print("ID ");
    Serial.print(rs485id);
    Serial.print(" opMode ");
    Serial.print(operatingMode);
    Serial.print(" driverState ");
    Serial.print(driverState);
    Serial.print(" pos ");
    Serial.print(getEncoderPosition());
    Serial.print(" current ");
}
if (shortReportVel)
{
    Serial.print("ID ");
    Serial.print(rs485id);
    Serial.print(" opMode ");
    Serial.print(operatingMode);
    Serial.print(" driverState ");
    Serial.print(driverState);
    Serial.print(" vel ");
    Serial.print(currentVel);
    Serial.print(" current ");
}
}
if (sendLogToMatlab)
{
    sendLogToMatlab = false;

    for (int j = 0; j < logSize; j++)
    {
        Serial.write(logForMatlab[j]);
        Serial.write(logForMatlab[j] >> 8);
        Serial.write(logForMatlab[j] >> 16);
        Serial.write(logForMatlab[j] >> 24);
    }
}
while (Serial.available() > 0)

```

```

{
    serialFloatData[serialFloatIndex++] = Serial.parseFloat();
    if (Serial.read() == '\n')
    {
        serialDataAvailable = 1;
    }
}
if (serialDataAvailable)
{
    if (((char)serialFloatData[0]) == rs485id)
    {
        switch ((int) serialFloatData[1])
        {
            case COM_SET_DRIVER_STATE:
                if (serialFloatIndex == 3)
                {
                    switch ((int)serialFloatData[2])
                    {
                        case 0://disable driver
                            driverState = DISABLED;
                            uManual = 0;
                            operatingMode = MANUAL_MODE;
                            profPositionModeFired = false;
                            break;
                        case 1://enable driver
                            uManual = 0;
                            operatingMode = MANUAL_MODE;
                            profPositionModeFired = false;
                            driverState = ENABLED;
                            break;
                    }
                }
                break;
            case COM_SET_OPERATING_MODE:
                if (serialFloatIndex == 3)
                {
                    uManual = 0;
                    switch ((int)serialFloatData[2])
                    {
                        case MANUAL_MODE:
                            operatingMode = MANUAL_MODE;
                            posPID.disablePID();
                            velPID.disablePID();
                            break;
                        case POSITION_MODE:
                            operatingMode = POSITION_MODE;
                            posRef = getEncoderPosition();
                            posPID.enablePID();
                            velPID.disablePID();
                            break;
                        case PROFILE_POSITION_MODE:

```

```

    profPositionModeFired = false;
    operatingMode = PROFILE_POSITION_MODE;
    posRef = getEncoderPosition();
    posPID.enablePID();
    velPID.disablePID();
    break;
case VELOCITY_MODE:
    operatingMode = VELOCITY_MODE;
    posPID.disablePID();
    velRef = currentVel;
    velPID.enablePID();
    break;
case PROFILE_VELOCITY_MODE:
    operatingMode = PROFILE_VELOCITY_MODE;
    //do nothing
    break;
case CURRENT_MODE:
    operatingMode = CURRENT_MODE;
    //do nothing
    break;
case HOMING_MODE:
    operatingMode = HOMING_MODE;
    //do nothing
    break;
}
}
break;
case COM_REL_MAX_PROF_POS_REF:
    if (serialFloatIndex == 3)
    {
        profPosRelRef = (long)serialFloatData[2];
        profInitialPos = getEncoderPosition();
        profFinalPos = profInitialPos + profPosRelRef;
        profTimeInSec = sqrt(abs(profPosRelRef) / 6000.0) * 0.7; //6000 darbe öteye 0.7 saniye tatlı
        geçiş, o yüzden ref
        if (((abs(profPosRelRef) / 6000.0) / (profTimeInSec / 0.7) * 16070.0) > MAX_PROF_VEL) //6000
        darbe 0.7 saniye için max hız 16070
        {
            profTimeInSec = (abs(profPosRelRef) / 6000.0) / (MAX_PROF_VEL / 0.7) * 16070;
        }
        generatePosProfile(profInitialPos, profFinalPos, profTimeInSec, coeff);
        float maxComputedAccel = evalAccel(coeff, profTimeInSec / 4.73);
        if (abs(maxComputedAccel) < MAX_PROF_ACCEL)
            profPositionModeFired = true;
        else
        {
            Serial.print("<");
            Serial.print(rs485id);
            Serial.print(">");
        }
    }
    profInitialTime = millis();

```

```

}
    break;
case COM_ABS_MAX_PROF_POS_REF:
    if (serialFloatIndex == 3)
    {
        profFinalPos = (long)serialFloatData[2];
        profInitialPos = getEncoderPosition();
        profPosRelRef = profFinalPos - profInitialPos;
        profTimeInSec = sqrt(abs(profPosRelRef) / 6000.0) * 0.7;
        if (((abs(profPosRelRef) / 6000.0) / (profTimeInSec / 0.7) * 16070.0) > MAX_PROF_VEL) //6000
        darbe 0.7 saniye için max hız 16070
        {
            profTimeInSec = (abs(profPosRelRef) / 6000.0) / (MAX_PROF_VEL / 0.7) * 16070;
        }
        generatePosProfile(profInitialPos, profFinalPos, profTimeInSec, coeff);
        float maxComputedAccel = evalAccel(coeff, profTimeInSec / 4.73);
        if (abs(maxComputedAccel) < MAX_PROF_ACCEL)
        profPositionModeFired = true;
        else
            Serial.println("?");
        profInitialTime = millis();
    }
    break;
case COM_REL_PROF_POS_REF:
    if (serialFloatIndex == 4)
    {
        profPosRelRef = (long)serialFloatData[2];
        profTimeInSec = serialFloatData[3];
        if (profTimeInSec > 0)
        {
            profInitialPos = getEncoderPosition();
            profFinalPos = profInitialPos + profPosRelRef;
            generatePosProfile(profInitialPos, profFinalPos, profTimeInSec, coeff);
            float maxComputedAccel = evalAccel(coeff, profTimeInSec / 4.73);
            if (abs(maxComputedAccel) < MAX_PROF_ACCEL)
                profPositionModeFired = true;
            else
                Serial.println("?");
            profInitialTime = millis();
        }
    }
    break;
case COM_ABS_PROF_POS_REF:
    if (serialFloatIndex == 4)
    {
        profFinalPos = (long)serialFloatData[2];
        profTimeInSec = serialFloatData[3];
        if (profTimeInSec > 0)
        {
            profInitialPos = getEncoderPosition();
            generatePosProfile(profInitialPos, profFinalPos, profTimeInSec, coeff);

```

```

float maxComputedAccel = evalAccel(coeff, profTimeInSec / 4.73);
    if (abs(maxComputedAccel) < MAX_PROF_ACCEL)
        profPositionModeFired = true;
    else
        Serial.println("?");

    profInitialTime = millis();

}
}
break;
case COM_REL_VEL_REF:
    if (serialFloatIndex == 3)
    {
        velRef += serialFloatData[2];
    }
    break;
case COM_ABS_VEL_REF:
    if (serialFloatIndex == 3)
    {
        velRef = serialFloatData[2];
    }
    break;
case COM_REL_POS_REF:
    if (serialFloatIndex == 3)
    {
        posRef += (long)serialFloatData[2];
    }
    break;
case COM_ABS_POS_REF:
    if (serialFloatIndex == 3)
    {
        posRef = (long)serialFloatData[2];
    }
    break;
case COM_SET_POS_GAINS:
    if (serialFloatIndex == 5)
    {
        posPID.setKp(serialFloatData[2]);
        posPID.setKi(serialFloatData[3]);
        posPID.setKd(serialFloatData[4]);
    }
    break;
case COM_SET_VEL_GAINS:
    if (serialFloatIndex == 5)
    {
        velPID.setKp(serialFloatData[2]);
        velPID.setKi(serialFloatData[3]);
        velPID.setKd(serialFloatData[4]);
    }

```

```

    }
    break;
case COM_PRINT_LONG_POS_REPORT:
    if (serialFloatIndex == 2)
    {
        longReportPos = true;
        longReportVel = false;
        shortReportPos = false;
        shortReportVel = false;
    }
    break;
case COM_PRINT_LONG_VEL_REPORT:
    if (serialFloatIndex == 2)
    {
        longReportPos = false;
        longReportVel = true;
        shortReportPos = false;
        shortReportVel = false;
    }
    break;
case COM_PRINT_SHORT_POS_REPORT:
    if (serialFloatIndex == 2)
    {
        longReportPos = false;
        longReportVel = false;
        shortReportPos = true;
        shortReportVel = false;
    }
    break;
case COM_PRINT_SHORT_VEL_REPORT:
    if (serialFloatIndex == 2)
    {
        longReportPos = false;
        longReportVel = false;
        shortReportPos = false;
        shortReportVel = true;
    }
    break;
case COM_PRINT_NO_REPORT:
    if (serialFloatIndex == 2)
    {
        longReportPos = false;
        longReportVel = false;
        shortReportPos = false;
        shortReportVel = false;
    }
    break;
case COM_SEND_DATA_FOR_MATLAB:
    if (serialFloatIndex == 2)
    {
        sendLogToMatlab = true;

```

```

    }
    break;
case COM_SET_U:
    if (serialFloatIndex == 3)
    {
        uManual = constrain((long)serialFloatData[2], -255, 255);
        logEnable = true;
    }
    break;
case COM_PRINT_HELP:
    if (serialFloatIndex == 2)
    {
        longReportPos = false;
        longReportVel = false;
        shortReportPos = false;
        shortReportVel = false;
        print_help();
    }
    break;
case COM_GET_POSITION_LONG:
    if (serialFloatIndex == 2)
    {
        unsigned long tempPos = getEncoderPosition();
        Serial.write(tempPos);
        Serial.write(tempPos >> 8);
        Serial.write(tempPos >> 16);
        Serial.write(tempPos >> 24);
    }
    break;
case COM_LOG_POSITION:
    if (serialFloatIndex == 2)
    {
        logWhat = LOG_POSITION;
    }
    break;
case COM_LOG_CURRENT:
    if (serialFloatIndex == 2)
    {
        logWhat = LOG_CURRENT;
    }
    break;
case COM_TEST_MOTOR_CONNECTION:
    if (serialFloatIndex == 3)
    {
        longReportPos = false;
        longReportVel = false;
        shortReportPos = false;
        shortReportVel = false;
        posPID.disablePID();
        velPID.disablePID();
        uManual = 0;
    }

```

```

    byte uTemp = constrain((long)serialFloatData[2], 0, 255);
    long tempPos = getEncoderPosition();
    uToPWM(uTemp);
    delay(100);
    uToPWM(0);
    if ((getEncoderPosition() - tempPos) > 0)
        Serial.println("passed");
    else
    {
        if (motorDir)
            motorDir = false;
        else
            motorDir = true;
        Serial.println("First failed but corrected automatically");
    }
    delay(100);
    uToPWM(uTemp);
    delay(100);
    uToPWM(0);
}
break;
case COM_GET_LOG_SIZE_INTEGER:
    if (serialFloatIndex == 2)
    {
        Serial.write(logSize);
        Serial.write(logSize >> 8);
    }
    break;
case COM_GET_LOOPRATE_UINT8:
    if (serialFloatIndex == 2)
    {
        Serial.write(loopRate);
    }
    break;
case COM_CHANGE_USER_POS_DIR:
    if (serialFloatIndex == 2)
    {
        encoderReset();
        if (userPosDir)
            userPosDir = false;
        else
            userPosDir = true;
        if (motorDir)
            motorDir = false;
        else
            motorDir = true;
    }
    break;
case COM_SET_LOOPRATE:// veri tiplerini gözden geçir
    if (serialFloatIndex == 3)
    {

```

```

    char loopRateTemp = (long)serialFloatData[2];
    if (loopRateTemp > 0)
        loopRate = (byte)loopRateTemp;
    }
    break;
case COM_SET_PRINTING_PERIOD:
    if (serialFloatIndex == 3)
    {
        long printingPeriodTemp = (long)serialFloatData[2];
        if (printingPeriodTemp > 0)
            printingPeriod = printingPeriodTemp;
        }
        break;
case COM_RESET_ENCODER_POS:
    if (serialFloatIndex == 2)
    {
        encoderReset();
    }
    break;
case COM_GET_POS_GAINS:
    if (serialFloatIndex == 2)
    {
        float tempFloat;
        unsigned long tempHex;
        tempFloat = posPID.getKp();
        tempHex = *(unsigned long *)&tempFloat;
        Serial.write((unsigned long)tempHex);
        Serial.write(((unsigned long)tempHex) >> 8);
        Serial.write(((unsigned long)tempHex) >> 16);
        Serial.write(((unsigned long)tempHex) >> 24);
        tempFloat = posPID.getKi();
        tempHex = *(unsigned long *)&tempFloat;
        Serial.write((unsigned long)tempHex);
        Serial.write(((unsigned long)tempHex) >> 8);
        Serial.write(((unsigned long)tempHex) >> 16);
        Serial.write(((unsigned long)tempHex) >> 24);
        tempFloat = posPID.getKd();
        tempHex = *(unsigned long *)&tempFloat;
        Serial.write((unsigned long)tempHex);
        Serial.write(((unsigned long)tempHex) >> 8);
        Serial.write(((unsigned long)tempHex) >> 16);
        Serial.write(((unsigned long)tempHex) >> 24);
    }
    break;
case COM_GET_VEL_GAINS:
    if (serialFloatIndex == 2)
    {
        float tempFloat;
        unsigned long tempHex;
        tempFloat = velPID.getKp();
        tempHex = *(unsigned long *)&tempFloat;

```

```

        Serial.write((unsigned long)tempHex);
        Serial.write(((unsigned long)tempHex) >> 8);
        Serial.write(((unsigned long)tempHex) >> 16);
        Serial.write(((unsigned long)tempHex) >> 24);
        tempFloat = velPID.getKi();
        tempHex = *(unsigned long *)&tempFloat;
        Serial.write((unsigned long)tempHex);
        Serial.write(((unsigned long)tempHex) >> 8);
        Serial.write(((unsigned long)tempHex) >> 16);
        Serial.write(((unsigned long)tempHex) >> 24);
        tempFloat = velPID.getKd();
        tempHex = *(unsigned long *)&tempFloat;
        Serial.write((unsigned long)tempHex);
        Serial.write(((unsigned long)tempHex) >> 8);
        Serial.write(((unsigned long)tempHex) >> 16);
        Serial.write(((unsigned long)tempHex) >> 24);
    }
    break;
}
}
serialFloatIndex = 0;
serialDataAvailable = 0;
}

} // end of loop

//float getFloat(long num, long den, long sign, float inFloat)
//{
// if (den == 0)
// return inFloat;
// else
// return sign ? (0.0 - (float)num) / (float)den : (float)num / (float)den;
//}
//
//long getLong(long num, long den, long sign, long inLong)
//{
// if (den == 0)
// return inLong;
// else
// return sign ? (0.0 - num) / den : num / den;
//}

void print_help(void)
{
    Serial.println("help: Print this Help :)");
    // Serial.println("p: new value for Kp, e.g, p 4.2");
    // Serial.println("i: new value for Ki, e.g, i 0.5");
    // Serial.println("d: new value for Kd, e.g, d 0.7");
}

```

```

// Serial.println("piden: enables pid computation, close the loop");
// Serial.println("aref: sets new absolute pos ref");
// Serial.println("relref: sets new relative pos ref");
// Serial.println("piddis: disables pid computation, break the loop");
// Serial.println("u: sets open loop motor Voltage as PWM duty cycle (0-255) -> (0-100)%
duty");
// Serial.println("long: tell the drive 'print all debug messages'");
// Serial.println("short: tell the drive 'print just only current encoder pos'");
// Serial.println("no: tell the drive 'don't print any debug messages'");
// Serial.println("matlab: sends logged data to matlab as long int values (32bit-signed)");
// Serial.println("getpos: drive sends current encoder pos as a long int value (32bit-signed)");
}
void uToPWM(float uln)
{
    digitalWrite(STBY, HIGH); //disable standby
    boolean inPin1 = LOW;
    boolean inPin2 = HIGH;
    if (((uln < 0) ^ motorDir)) {
        inPin1 = HIGH;
        inPin2 = LOW;
    }
    digitalWrite(IN1, inPin1);
    digitalWrite(IN2, inPin2);
    analogWrite(PWMPin, abs((int) uln));
}
void isrForEncPinA() {
    encoderPosition = encoderPosition + 1 - (((PIND & B00001000) >> 2) ^ ((PIND & B00000100) >>
1));
}
void isrForEncPinB() {
    encoderPosition = encoderPosition - 1 + (((PIND & B00001000) >> 2) ^ ((PIND & B00000100) >>
1));
}
long getEncoderPosition()
{
    if (userPosDir)
        return encoderPosition;
    else
        return 0 - encoderPosition;
}
void encoderReset()
{
    longReportPos = false;
    longReportVel = false;
    shortReportPos = false;
    shortReportVel = false;
    posPID.disablePID();
    velPID.disablePID();
    uManual = 0;
    encoderPosition = 0;
}

```

```

void generatePosProfile(long initialPos, long targetPos, float profTime, float *coeff)
{
    coeff[0] = 12 * ((float)(targetPos - initialPos)) / (2 * pow(profTime, 5));
    coeff[1] = -30 * ((float)(targetPos - initialPos)) / (2 * pow(profTime, 4));
    coeff[2] = 20 * ((float)(targetPos - initialPos)) / (2 * pow(profTime, 3));
    coeff[3] = (float) initialPos;
}

float evalPos(float *coeff, float evalTime)
{
    return coeff[0] * pow(evalTime, 5) + coeff[1] * pow(evalTime, 4) + coeff[2] * pow(evalTime, 3)
    + coeff[3];
}

float evalVel(float *coeff, float evalTime)
{
    return 5.0 * coeff[0] * pow(evalTime, 4) + 4.0 * coeff[1] * pow(evalTime, 3) + 3.0 * coeff[2] *
    pow(evalTime, 2);
}

float evalAccel(float *coeff, float evalTime)
{
    return 20.0 * coeff[0] * pow(evalTime, 3) + 12.0 * coeff[1] * pow(evalTime, 2) + 6.0 * coeff[2] *
    evalTime;
}

```

SİSTEMİN ELEKTRİKSEL MEKANİZMASI

Kullanılacak Elemanlar Ve Bağlantı Elemanları

Bağlantı elemanları olarak kullandığımız elemanlar;

- 1-Arduino Uno
- 2-Dc Motor 6v;
- 3-Delikli Pertinaks
- 4-Motor Sürücü;
- 5-Raspberry Pi ve Kamera Modülü;
- 6-Kamera;

1-Arduino ;

Arduino, üzerinde Atmel mikrokontrolörleri bulunan açık kaynak kodlu donanımdır. PCB baskı devre ve şemaları, yerleşim planları ve üzerindeki programlanabilir elemanların kodları bütün detayları ile kullanımına rahatça ulaşabileceği bir şekildedir. Hazır alınabileceği gibi, aynısı da hazırlanabilir. Arduinoki mikrokontrolör (AtmegaXX) sayesinde Arduino temel C programlama dili ile programlanır ve bu program Processing tabanlı Arduino Yazılım Geliştirme Ortamı (IDE) yardımı ile karta yüklenir. Ayrıca her projeye uyumlu kart olabileceği gibi ve bu kartlara uygun shield (katman) olarak adlandırılmış kullanım kolaylığı sağlayan ek donanımlar da mevcuttur. (Bluetooth, Wireless, Ethernet, Motor sürücü katmanları vs.).

Arduino'un Avantajları ;

- Önyükleyiciye (bootloader) sahip olduğundan USB port bağlantısı ile mikroişlemciye program yükleyebilir. Bu sayede mikroişlemcilerdeki çıkarılıp takılmalardan kaynaklanan bacak kırılmaları riski ortadan kalkar.
 - Diğer programlamalara göre kolay ve daha ucuzdur.
- Ve biz bahsi geçen projemizde arduino uno ile bağlantılarımızı gerçekleştireceğiz.

Arduino Uno Arduino Uno R3 kartı üzerinde 8 bitlik Atmega328 işlemcisi, 14 digital Input-Output pini, bunlardan 6 tanesi PWM çıkışı, 6 analog giriş pini bulunmaktadır. Ve 16 Mhz kristal osilatör, USB bağlantısı, regüle edilmiş 5V, ICSP başlığı ve reset butonu bulunmaktadır. Çalışması için gerekli gücü USB bağlantısı ile PC'den veya 7-12 Volt'luk DA güç kaynağından sağlar. Besleme gerilimi için alt ve üst sınırlar 6-20 Volttur. Giriş çıkış pini başına akım 40 mA'dır. Kart üzerinde regüle edilmiş 3.3 V çıkışı ve bunun için çıkış akımı 50 mA'dır. FLASH Hafıza 32KB, SRAM 2 KB, EEPROM 1KB olarak hafıza büyüklükleri de belirtilmiştir. Ayrıca; harici bir güç kaynağı kullanılacağı zaman Arduino kartı üzerindeki VIN pininden giriş yapılabilir.

2-DC Motor 6v;

Elektrik enerjisini, mekanik enerjiye çeviren elektrik makinasına doğru akım motoru denir. Doğru akım motorlarına DC motor denilmektedir.

DC motor statorun yapısına göre sabit mıknatıslı ve elektromıknatıslı olarak DC motor üreticileri tarafından, DC motor imalatı yapılır. Elektromıknatıslı ve sabit mıknatıslı arasında stator dışında yapıları açısından farklılık yoktur. Doğru akım motorları, duran ve dönen kısımlardan meydana gelir. Duran kısım stator, dönen kısım ise rotordur.

Endüktör(stator);

Stator makinanın dış tarafında kalan, duran kısmıdır. Karkas üstüne yerleştirilen ana, yardımcı kutuplar ve bunlar üstündeki sargılardan meydana gelir. Mıknatıs alanını sağlar. Doğru akım motorlarında manyetik alan endüktörde (kutup) oluşur. Stator karkasına yerleştirilen kutuplar, demir sac levhaların paketlenmesiyle yapılmıştır. Orta ve büyük güçteki makinalarda ana kutuplar ve bunların arasına yerleştirilmiş yardımcı kutuplar vardır. Doğru akım motorlarında, kutup sayısı alternatif akım makinelerindeki gibi gerilim hız ve indüklenen akımın frekansına bağlı değildir. Kutup sayısı makinenin devir sayısına ve gücüne göre değişim gösterir.

Endüvi (Rotor);

İletkenleri taşıyan ve gerilim indüklenen dönen kısımdır. Kalınlığı 0,30-0,70 mmdir. Endüvi milinden mekanik enerji alınır. Rotor mıknatıs alanında döner. Stator ile rotorun bir ortak yönü vardır. Her ikisinde de saç paket ve bobinler bulunur. Endüvi, çelik bir mil üzerine preslenmiş rotor sac paketlerinden, bu sac paketi oluklarına döşenmiş endüvi sargılarından ve sargı uçlarının bağlandığı bir kollektörden oluşur. Rotor akımı, fırça adını verdiğimiz (yayların bastırdığı) kömür çubuklarla kollektör denilen bakır lamalardan meydana gelen kısımdan verilir.

Kollektör

Kollektör sayesinde doğru akım motorlarında endüviye gerilimin iletilmesi sağlanır. Statorda üretilen alternatif gerilim kolektör tarafından doğrultulur ve kolektör üzerine temas eden fırçalar yardımı ile dış devreye iletilir. Kollektör birbirinden tek tek yalıtılmış sert bakır levhalardan oluşur. Kollektör dilimleri yüksek ısıya dayanıklı sert bakırdan yapılır. Yalıtkanlık için bakır dilimlerin arasına 0,5 –1,5 mm yalıtkan maddeler kullanılır. Kollektör dilimlerine değen fırçalar, bağlama elemanlarını oluşturur.

DC Motor Çeşitleri;

Fırçasız DC Motorlar (Brushless DC Motor)

- i) Şönt Motor
- ii)Seri Motor
- iii) Kompunt Motor (Ters Kompunt, Eklemeli Kompunt)
- iiii) Fırçalı DC motor
- Sabit mıknatıslı motor
- Elektromıknatıslı motor

Doğru akım motorlarını çalıştırmak için, DC motor sürücüleri ile motor sürücü kartı gerekmektedir. DC elektrik motorları özelliklerine göre DC sürücüleri yapılır. Örneğin; Şönt motor, seri motor, kompunt motor DC servo motor v.s.

Motor Sürücü;

TB6612FNG motor sürücüsü, 1.2A'lık sabit bir akımda (3.2A tepe noktası) iki DC motoru kadar kumanda edebilir. Motoru iki fonksiyon modundan birinde (CW, CCW, kısa-fren ve durma) kontrol etmek için iki giriş sinyali (IN1 ve IN2) kullanılabilir. İki encoder çıkışı (A ve B) ayrı ayrı kontrol edilebilir, her motorun hızı 100kHz'e PWM giriş sinyali ile kontrol edilir. Motoru bekleme modundan çıkarmak için STBY pimi yüksekte çekilmelidir.

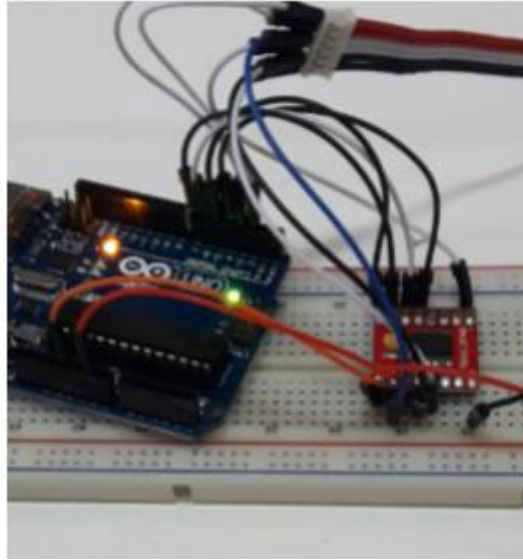
Motor gerilimi (VCC) 2.7 - 5.5VDC aralığında olabilirken, motor beslemesi (VM) maksimum 15VDC gerilimle sınırlandırılmıştır.(sistemde 12v kullanılacaktır).

Çıkış akımı kanal başına 1.2A (veya kısa, tek darbe için 3.2A'ya kadar) olarak derecelendirilmiştir.

Kurulu, gösterilen şekilde tüm bileşenleri takılmış şekilde gelir. Ayırma kapasitörleri her iki besleme hattına dahildir. TB6612FNG'nin tüm pimleri giriş pimleri bir tarafta ve çıkış pimleri diğerinde olacak şekilde düzenlenmiştir.

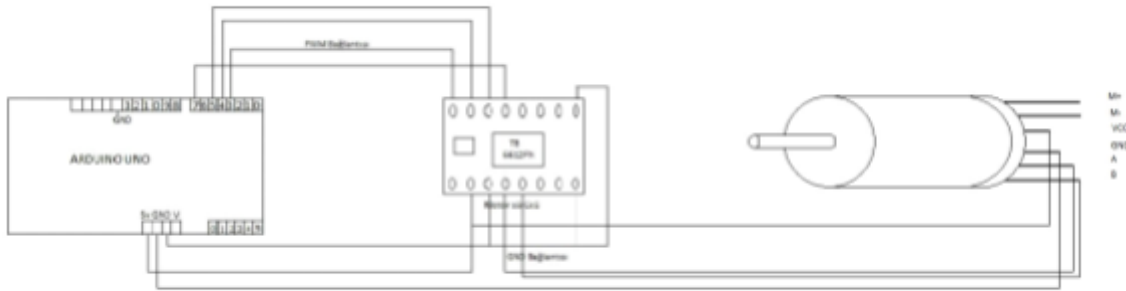
Kullanılacak Elemanların bağlantı Şemaları;

Sistemimizin tamamlanması için yapacağımız son işlemlerden biri de elektronik kısmı idi. Elektronik kısmı yaparken öncelikli olarak motor sürücü ve arduino arasındaki geçişler lehim yapılmadan jumper yardımıyla bağlantıları hazırlandı. Bu bağlantıların sorunları varsa tespit edildi. Çıkabilecek tek sorun jumperların yanlış pimlere bağlanıp arduino veya motor sürücünün yanması idi fakat ilk bağlantıda hiçbir sorunla karşılaşmadı.



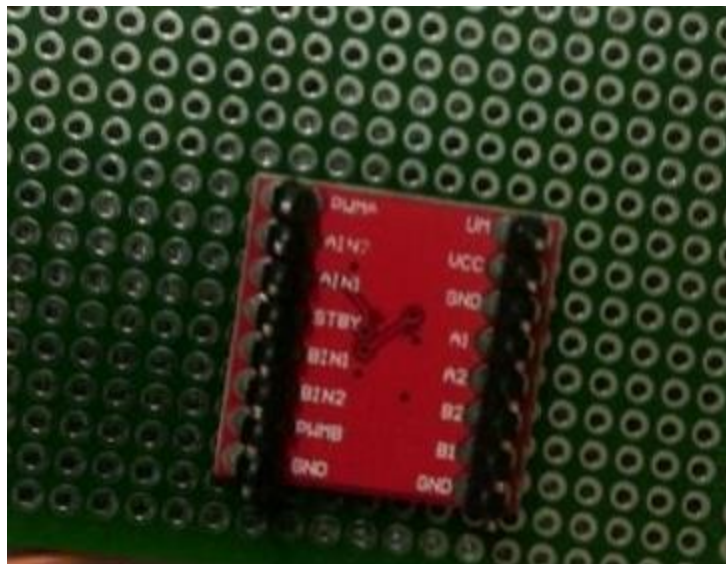
Bu bağlantı şemasından sonra sistemimizde bir sıkıntı olmadığı gözlemlenmiş olup diğer bağlantımız olan motor bağlantısına geçilmiştir. Motorumuzda bir adet GND (toprak bağlantısı) , bir adet VCC (+5V girişi) , iki adet encoder ucu (A ve B olmak üzere) ve güç girişlerimiz (M+ ve M-) mevcuttur. Encoder girişlerimizi motor sürücümüze bağladık. (A ucunu motor sürücümüzdeki A1 pimine , B ucunu ise yine motor sürücümüzdeki A2 pimine bağladık). GND ucunu ise arduino' muzdaki GND pimine bağladık. VCC ucunu da yine aynı

şekilde arduino'muzda mevcut olan 5V pimine bağlamış olduk. Güç girişlerimiz olan M+ ve M- uçlarımızı da dışardan bir güç kaynağı ile sağladık.

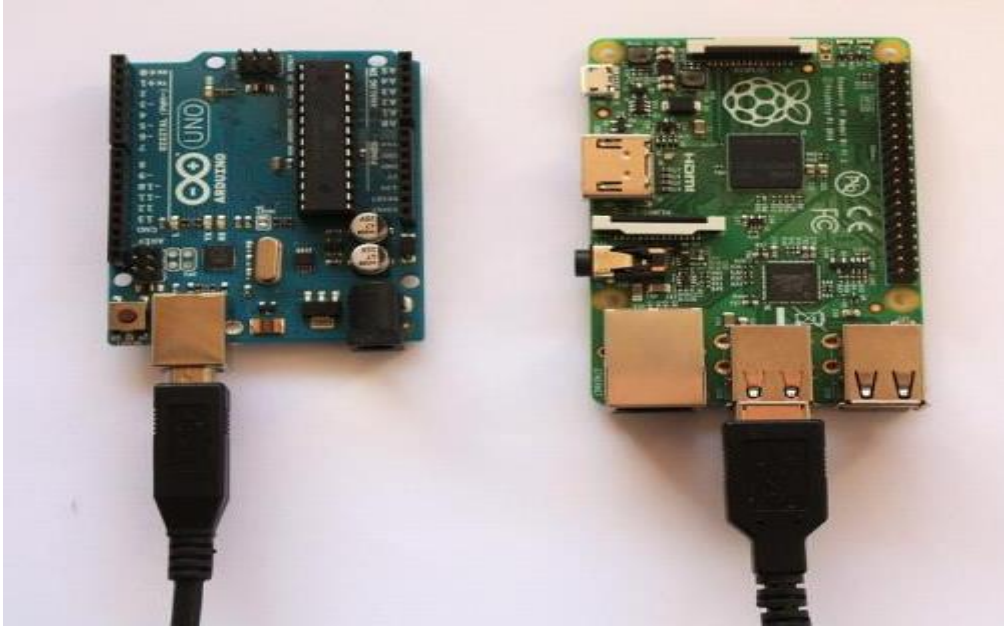


3- Delikli Pertinaks;

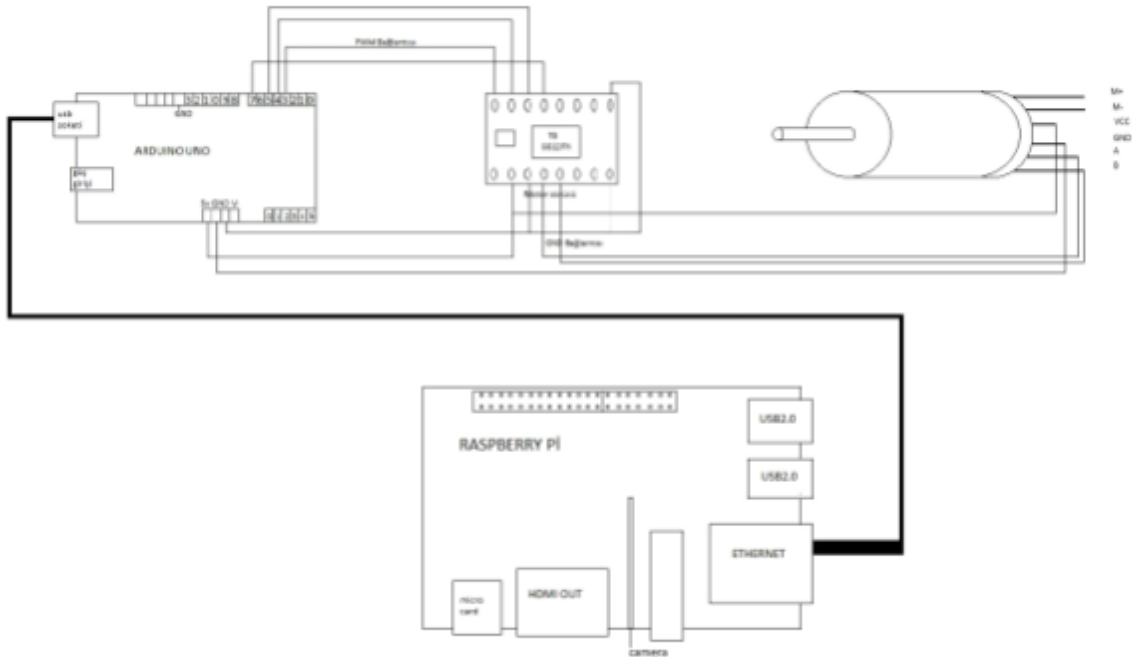
Delikli pertinaks, baskı devre de kullanılan bakır levhaların bir benzeridir. Bir çok parçayı bir arada tutmak için kullanılmaktadır. Biz ise projemizde motor sürücü arduino ve ara bağlantıları kullanmak için delikli pertinaksa ihtiyaç duyacağız.



Ve projemizin son bağlanan elemanımız ise Raspberry Pi olacaktır. Raspberry Pi'yi Arduino' ya seri olarak bağlayarak tüm elektronik devreyi tamamlamış olacağız.



Raspberry Pi' yi Arduino' ya seri bağlamamızın sebeplerinden biri ise daha hızlı veri aktarımı olmasıdır. Bir diğer sebebi ise; nasıl Arduino' yu bilgisayarımıza taktığımızda Arduino ' muzu kolayca programlayabiliyorsak, arduino ' yu Raspberry Pi ' ye seri taktığımızda da aynı işlemi Raspberry pi içinde de kolaylıkla yapabilme imkanını bize vermesidir. Raspberry Pi Linux tabanlı bir işletim sistemine sahip olduğundan, bu şekilde Arduino' yu programlamak da bizim için kolaylaşacaktır.Ve sistemimizin son hali de şu şekilde olmaktadır;



SİSTEMİN MEKANİZMASI

Projemizde kullandığımız malzemelerin montajının yapıldığı parçamızın çizimini SOLIDWORKS programı kullanılarak dizaynı yapılmıştır.Yaptığımız Solidworks çizimini daha sonra 3D Printer da üretimi yapıldı. Parçamızın malzemesi plastik malzemeden üretilmiştir.

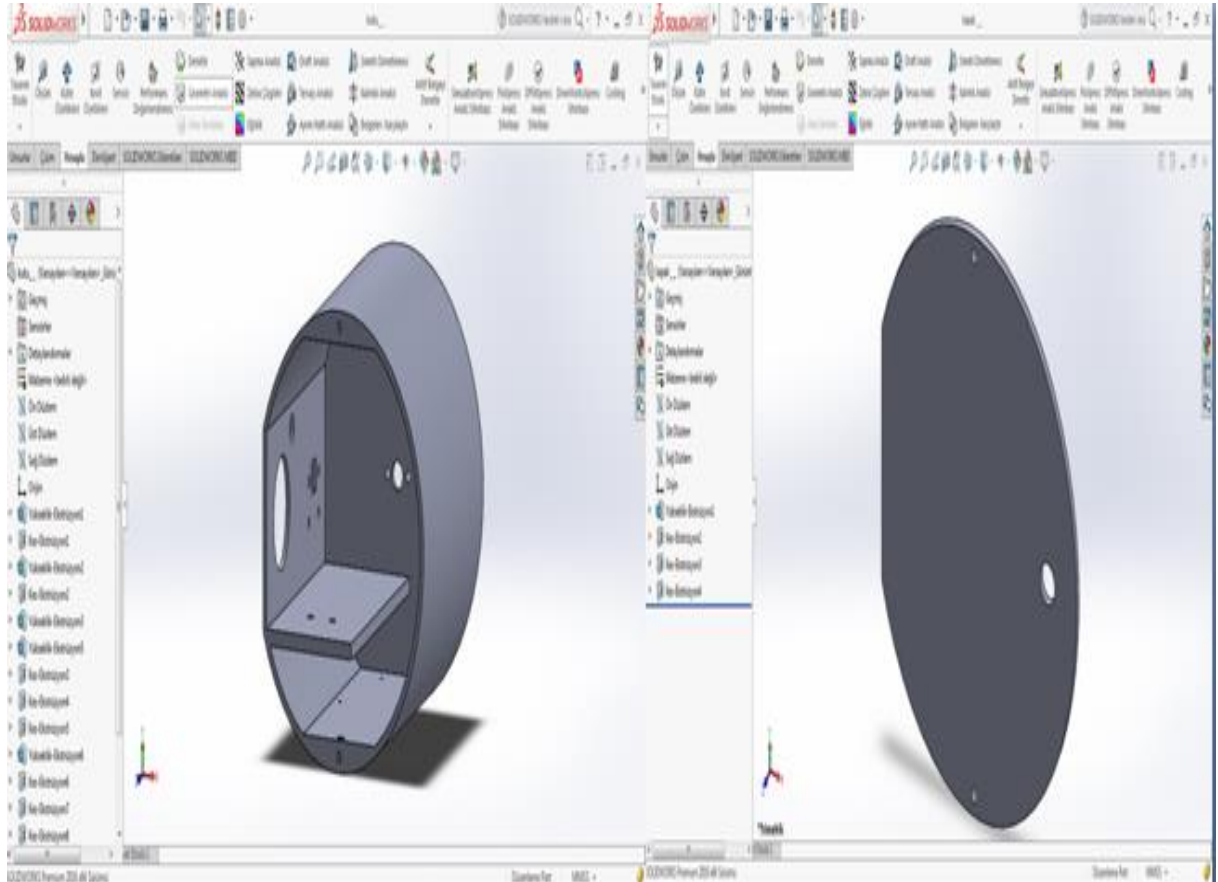
Dizaynımız hareket eden katı kütlelerin havayla etkileşimlerini inceleyen aerodinamik bilim dalı baz alınarak tasarlanmıştır.Dizayn da montajını yaptığımız malzemelerin kütle çekim kuvvetlerine dikkat edilerek ağırlık merkezine göre çizimi yapılmıştır.

Aşağıda görüldüğü gibi çizimimizin ölçüleri ;

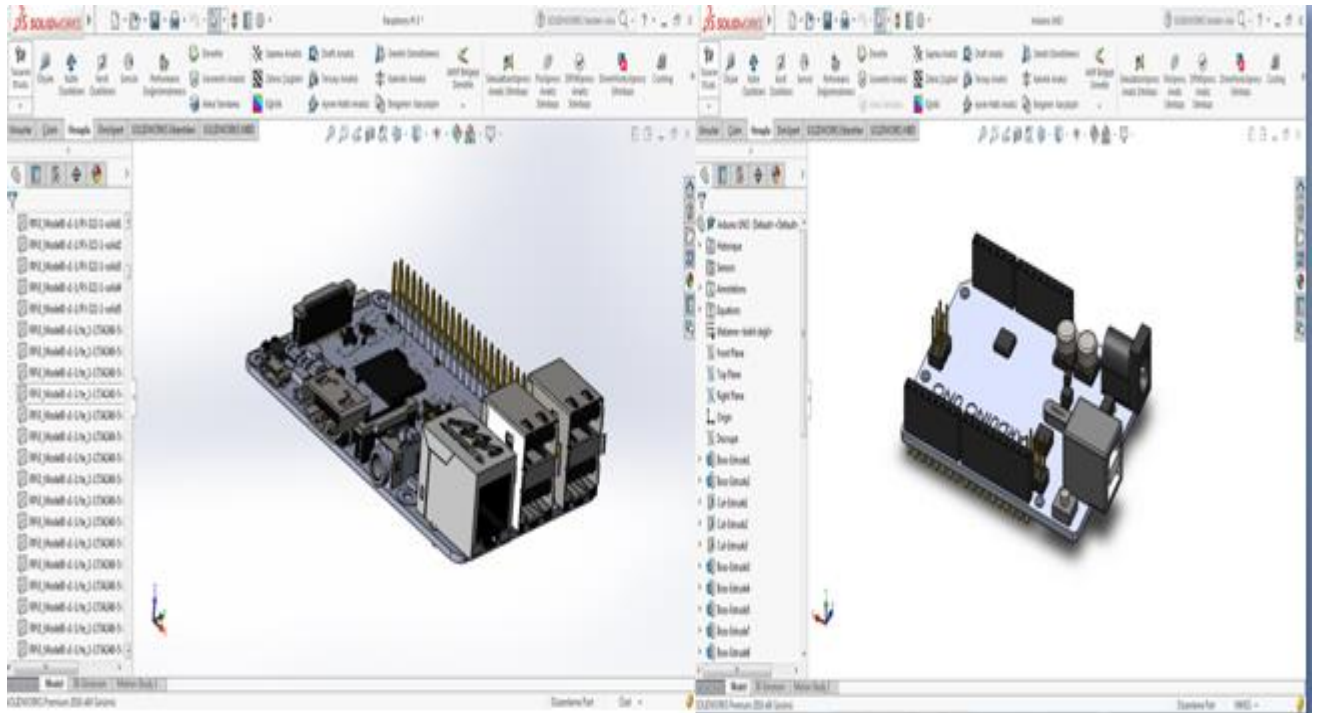
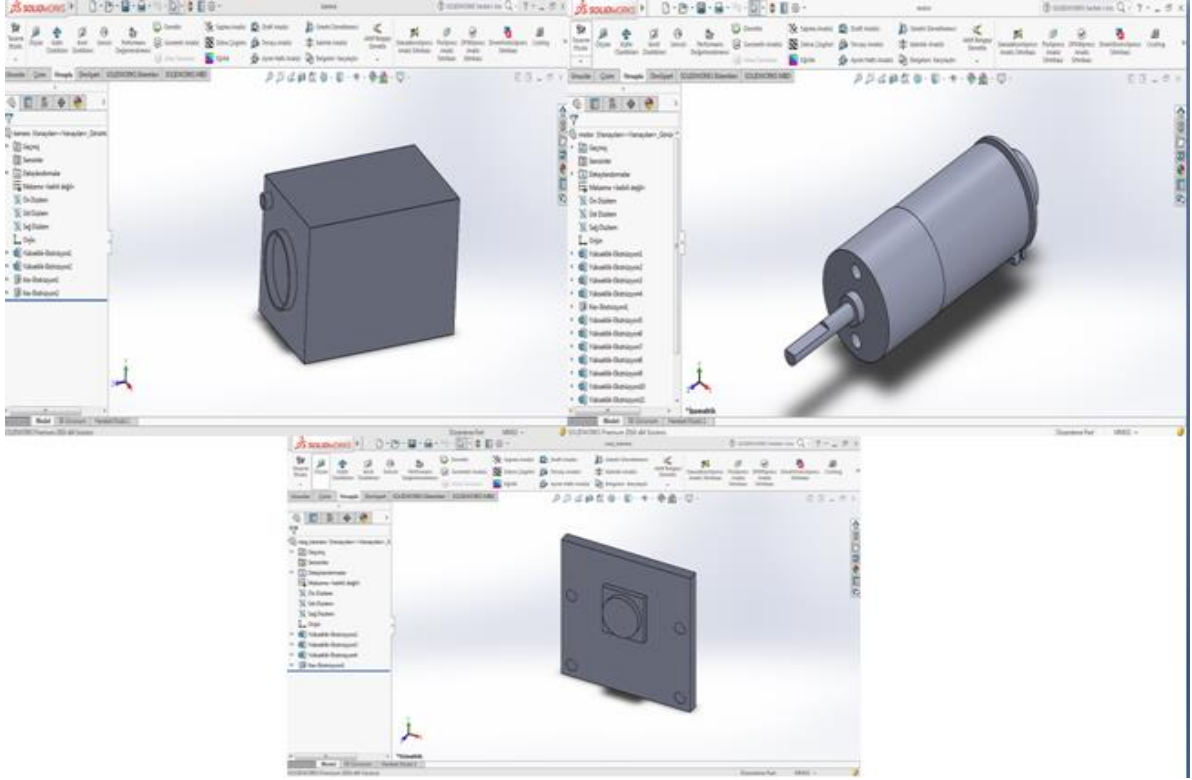
Genişliği : 140 mm

Uzunluğu :130 mm

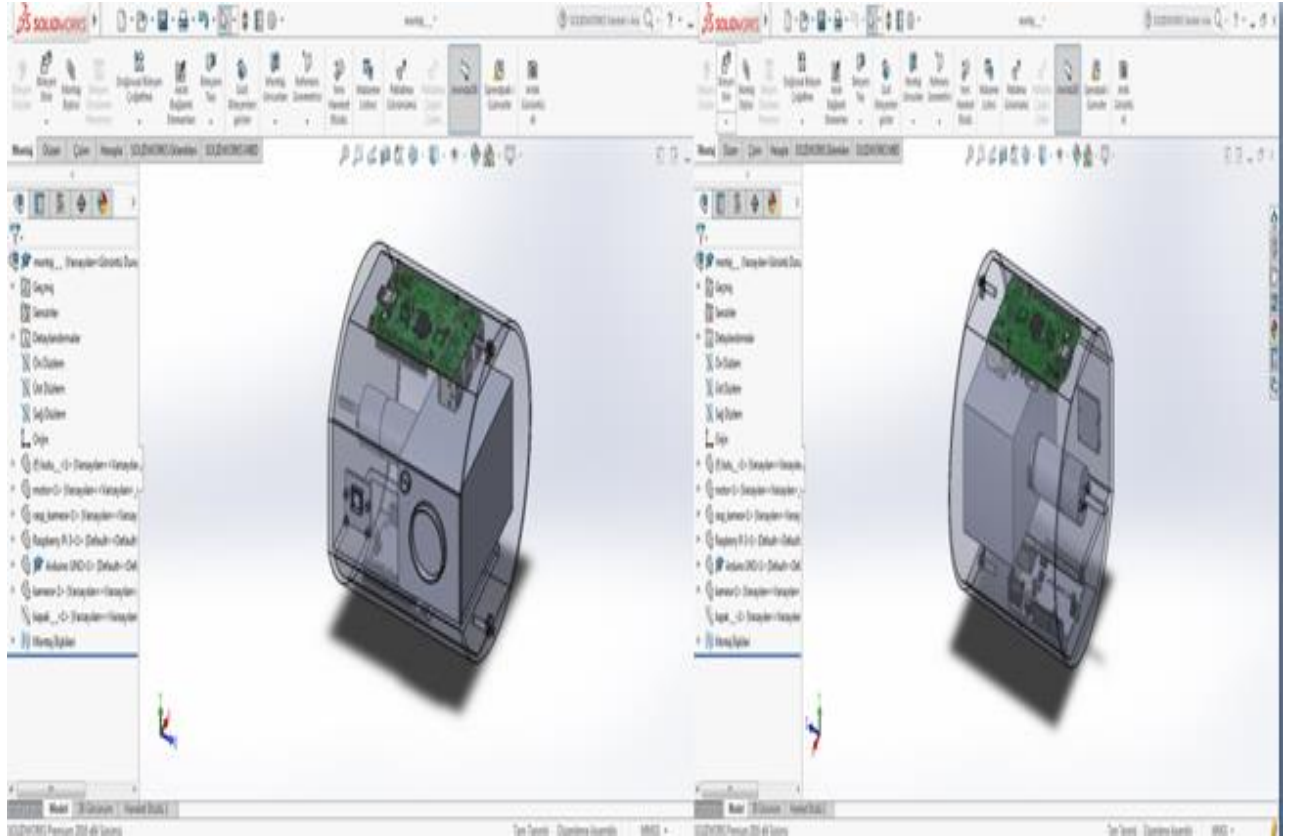
Kalınlığı:3 mm olacak şekilde tasarlanmıştır.



Sistemimiz de kullanılan malzemelerin de Solidworks çizimi yapılmıştır.

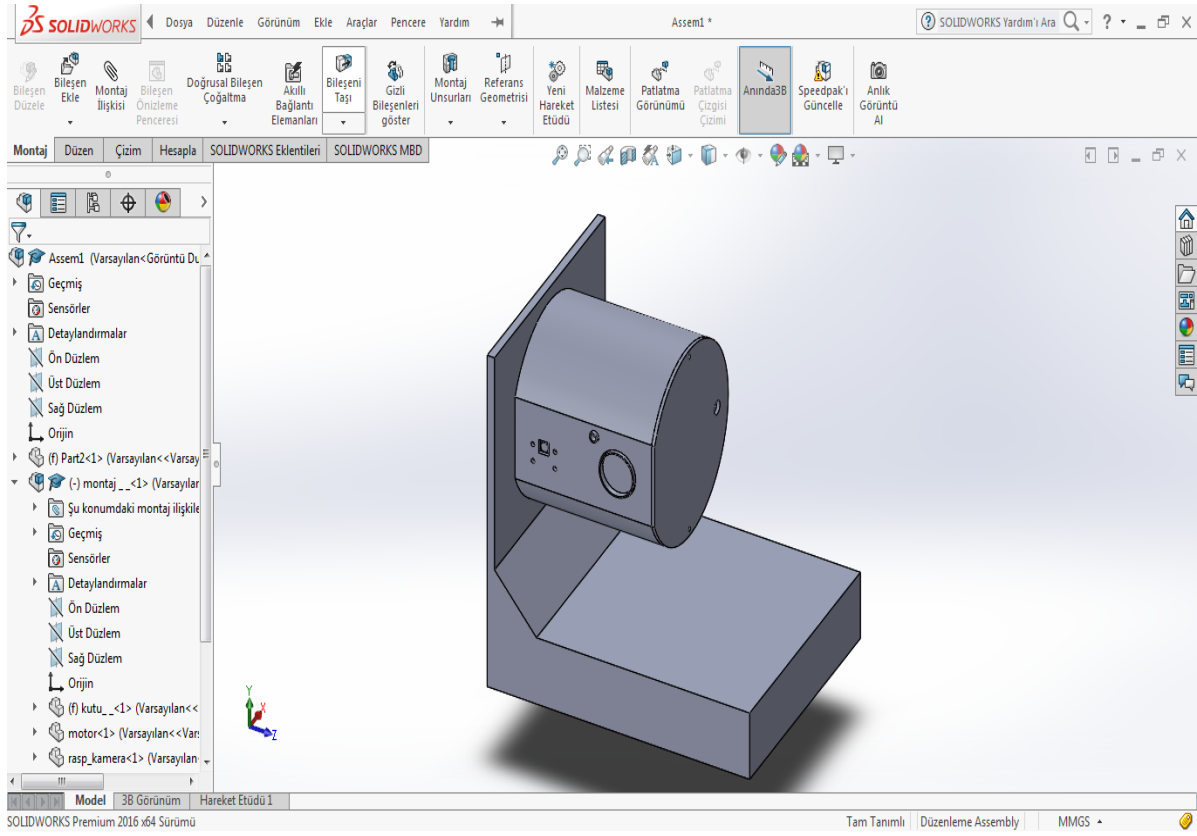


Daha sonra tasarladığımız parçamızla sistemimiz de olan malzemelerin montajı yapılmıştır.



En son olarak da çizdiğimiz parçayı bir yere sabitlemek için farklı bir parça dizayn edildi.Parçamızın üretimi sanayi de yapıldı.Parçamızın malzemesi ise mdf den yapılmıştır.

En son olarak da parçalarımızı birleştirerek bir sistem haline getirildi.



Sistemin son haline ulaşılabilmesi için, alınmış olan malzemelerin listesi, isimleri, adetleri, alındığı yerleri ve fiyatları aşağıdaki tabloda verilmiştir.

NUMARA	MALZEME ADI	ADET	ALINDIĞI YER	FİYAT (TL)
1	Raspberry Pi 3	1	Robotistan	162.84 TL
2	Raspberry Pi Kızılötesi Kamera Modülü V2	1	Robotistan	157.78 TL
3	MakeBlock DC Encoder Motor	1	domiRobot	108.54 TL
4	TB6612-FNG Çift Motor Sürücü Modülü	1	domiRobot	16.61 TL
5	Micro SD Card	1	Vatan	50 TL
6	Mekanik Dizayn	1	Niğde / Merkez	30 TL

KAYNAKÇA

[1] Müller H, Michoux N, Bandon D, Geissbuhler A. "A Review of Content-Based Image Retrieval Systems in Medical Applications-Clinical Benefits and future Directions". International Journal of Medical Informatics, 73(1), 1-23, 2004

[2] INTEL CORPORATION: Intel researcher teaches computers to 'read lips' to improve accuracy of speech recognition software. M2 Presswire, Coventry, Apr 28, 2003, pg1.

[3] M. Şanlı, F. Zengin, O. Urhan, M. K. Güllü, "Web Kamerasıyla Hareketli Nesne Takibi, Elektronik ve Haberleşme Mühendisliği Bölümü Mühendislik Fakültesi, İzmit/KOCAELİ, 2005

[4] Bradski, G. and Kaehler, A., "Learning OpenCV: Computer Vision with the OpenCV Library", O'Reilly Media, Amerika Birleşik Devletleri, 16-17 (2008).

[5] A. Beyoğlu, A. Kantarcı ve K. Erciyes "Telsiz Duyarga Ağlarında Çok Hızlı Hareket Eden Hedefler için Hedef Takip Algoritmaları," 2012

[6] http://akizilkaya.pamukkale.edu.tr/B%C3%B6l%C3%BCm4_goruntu_isleme.pdf

[7] <http://www.bilisim-kulubu.com/sozluk/>

[8] Gonzalez RC, Woods RE. Digital Image Processing. 3rd ed. London, England, Prentice Hall, 2008