

OTONOM ARAZI ROBOTU

2020
BİLGİSAYAR MÜHENDİSLİĞİ
BİTİRME PROJESİ TEZİ

Uğur SÜRÜL-Ali Can PAMUK

OTONOM ARAZİ ROBOTU

Uğur SÜRÜL-Ali Can PAMUK

**Karabük Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümünde
Bitirme Projesi Tezi
Olarak Hazırlanmıştır.**

**KARABÜK
HAZİRAN 2020**

Uğur SÜRÜL ve Ali Can PAMUK tarafından hazırlanan “OTONOM ARAZI
ROBOTU” başlıklı bu projenin Bitirme Projesi Tezi olarak uygun olduğunu
onaylarım.

Dr. Öğr.Üyesi Rafet DURGUT

.....

Bitirme Projesi Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

...../...../2020

Bilgisayar Mühendisliği Bölümü , bu tez ile, Bitirme Projesi Tezini onamıştır

Dr.Öğr.Üyesi Hakan KUTUCU
Bölüm Başkanı

.....

“Bu projedeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Uğur SÜRÜL- Ali Can PAMUK

ÖZET

Bitime Projesi Tezi

OTONOM ARAZI ROBOTU

Uğur SÜRÜL-Ali Can PAMUK

Karabük Üniversitesi

Bilgisayar Mühendisliği

Bilgisayar Mühendisliği Bölümü

Tez Danışmanı:

Dr.Öğr.Üyesi Rafet DURGUT

Haziran 2020,

Raspberry Pi kullanarak Dagu Wild Thumper 4WD 4 tekerden çekiş özelliğine sahip arazi aracının motor sürücüler yardımı ile otonom hareket ettirilerek TensorFlow ile nesne tespitinin sağlanması çalışması yapılmıştır.

ABSTRACT

Senior Projcet Thesis

AUTONOMIC LAND ROBOT

Uğur SÜRÜL-Ali Can PAMUK

Karabük University

Faculty of Engineering

Department of Computer Engineering

Project Supervisor

Asts. Prof. Assistant Rafet DURGUT

Jun 2020

Using the Raspberry Pi, Dagu Wild Thumper 4WD is used to operate object detection with TensorFlow by moving autonomously about the motor use of the all-wheel drive off-road vehicle.

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde, oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Dr.Öğr. Üyesi Rafet DURGUT’a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	5
ABSTRACT	6
TEŞEKKÜR	7
İÇİNDEKİLER	viii
ŞEKİLLER DİZİNİ	x
BÖLÜM 1	1
GİRİŞ.....	1
1.1.Literatür Taraması.....	1
1.2.Dagu Wild Thumper 4WD	2
1.3.BTS7960B 20 Amper Motor Sürücüsü	4
1.4.Raspberry Pi 3 model:	6
1.5.Raspberry Pi Camera.....	9
1.6. MZ80 80cm Menzilli Kızılötesi Sensör.....	10
BÖLÜM 2	13
KURULUM.....	13
2.1. Raspbian Kurulumu	13
2.2. System Sekmesi.....	15
2.3. Interfaces Sekmesi:	16
2.4. Performance Sekmesi:	17
2.5. Localisation Sekmesi:	18
2.6. Ağ Ayarları:.....	19
BÖLÜM 3	21
RASPBERRY Pİ 3 KURULUM VE KODLANMASI.....	21
3.1. Raspberry Pi3 ile Donanım Bağlantıları:.....	21
3.2. Raspberry Pi3 Kodlaması:.....	21
BÖLÜM 4	30
NESNE TESPİTİ.....	30
4.1. Raspberry Pi güncellenmesi	30
4.2.TensorFlow Yüklenmesi	30
4.3.OPENCv Yüklenmesi	33

4.4.Protobuf Derlenmesi ve Yüklmesi.....	34
4.5.TensorFlow Dizin Yapısını ve PYTHONPATH Değişkenini Ayarlama.....	37
4.6. Nesnelerin Algılanması.....	40

ŞEKİLLER DİZİNİ

Sayfa

BÖLÜM 1	1
Şekil 1.1.Dagu Wild Thumper 4WD.....	3
Şekil 1.2. Dagü Wild Thumper 4WD Motor Kısmı	3
Şekil 1.3. Dagü Wild Thumper 4WD Motor Kısmı	4
Şekil 1.2.1. BTS7960B Blok Diyagram.....	5
Şekil 1.2.2. BTS7960B Uygulama Örneği	6
Şekil 1.2.1. Raspberry Pi 3 Genel Görünümü	7
Şekil 1.2.1. Raspberry Pi GPIO Başlığı.....	8
Şekil 1.4.1. Raspberry Pi Kamera Ön Görünüm	9
Şekil 1.4.2. Raspberry Pi Kamera Arka Görünüm.....	10
Şekil 2.1.1. Raspbian Kurulumu.....	13
Şekil 2.1.2. Raspbian Kurulumu.....	14
Şekil 2.1.3. Raspbian Kurulumu.....	14
Şekil 2.2.1. Raspbian Temel Ayarlar	15
Şekil 2.3.1. Raspbian Temel Ayarlar	16
Şekil 2.4.1. Raspbian Temel Ayarlar	17
Şekil 2.5.1. Raspbian Temel Ayarlar	18
Şekil 2.6.1. Raspbian Temel Ayarlar	19
Şekil 3.1. Donanım Bağlantıları	21
Şekil 3.2.1. Yazılan kodlar	24
Şekil 3.2.2. Sistemin Çalışması	27

BÖLÜM 1

GİRİŞ

1.1.Literatür Taraması

Mobil robotlar günümüzde tarım, mayın tarama, askeri uygulamalar, uzay araştırmaları ve insanların erişemediği veya nükleer santraller gibi insanlara zararlı ortamlarda kullanılmaktadırlar. Ancak bu robotların bazıları bir operatörden bağımsız olarak, diğer bir deyişle otonom olarak çalışmaktadırlar. Dünyada çeşitli alanlarda kullanılmak üzere otonom robot geliştirmek için bir yarış vardır.

Bu tezde çeşitli alanlarda kullanılmak üzere (öncelikli olarak arazi alanında) otonom robot tasarlanmıştır. Tasarlanan robot Tensorflow ile kodalanarak arazide otonom olarak hareket ederken nesne tanınması gerçekleştirecektir.

Projede kullanılacak olan donanımlar;

- Dagu Wild Thumper 4WD
- BTS7960B 20 Amper motor sürücüsü
- Raspberry Pi 3 model
- Raspberry Pi Camera
- MZ80 80cm Menzilli Kızılötesi Sensör

1.2.Dağı Wild Thumper 4WD

Dayanıklı ve 4 tekerden çekiş özelliğine sahip bu robot gövdesini arazi aracı projelerinizde güvenle kullanabilmemize olanak sağlayacaktır. 120 mm çaplı dişli lastiklere sahip her bir tekerlek, birbirinden bağımsız süspansiyona sahip olduğundan her türlü arazi koşullarında sorunsuzca hareket imkanı sağlar. 2mm kalınlığında anodize alüminyum gövde, 10mm aralıklarla yerleştirilmiş 4mm çaplı delikler barındırır. Bu sayede elektronik kartlarınızı ve diğer aksesuarlarınızı kolayca monte etmemizi sağlayacaktır.

Özellikler:

Boyutlar: 280 × 300 × 130 mm

Ağırlık: 1.9 kg

Zeminden yükseklik (hafif yük durumunda): 60 mm

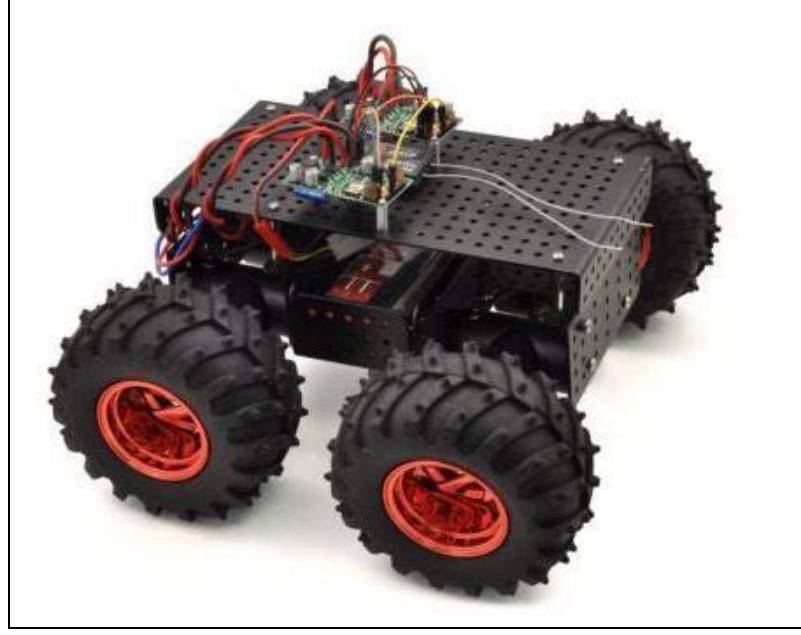
Tavsiye edilen motor gerilimi: 2-7.2V

Zorlanma akımı (@7.2V): 6.6A (motor başına)

Yüksüz akım (@7.2V): 420mA (motor başına)

Yüksüz devir: 350 RPM

Zorlanma torku: 5 kg-cm



Şekil 1.1.Dagu Wild Thumper 4WD



Şekil 1.2. Dagu Wild Thumper 4WD Motor Kısmı



Şekil 1.3. Dagu Wild Thumper 4WD Motor Kısım

Siyah renk şasi, 75:1 redüksiyon oranına sahip 25mm redüktörlü DC motorlar ve krom renkli jantlara sahip dişli lastik tekerlekler ile gelmektedir. Ürün içeriği robot gövdesi, 4 adet motor, 4 adet tekerlek ve montaj ekipmanlarından oluşmaktadır.

7.2V NiMH pil ile kullanıldığında, üzerindeki motorlar ile 3 km/h maksimum sürata çıkabilir. Motorların zorlanma torku yaklaşık olarak 11 kg-cm'dir. Şase üzerinde RC arabalarda sıklıkla kullanılan 6 hücreli 7.2V NiMH piller için iki adet bölme yer almaktadır. Ayrıca motor sürücüsü ve pil bağlantılarını kolayca yapabilmemiz için üzerinde vidalı klemensler mevcuttur. Motorlar yüksüz durumda 420mA, zorlanmada ise 6.6A akım çekmektedir. Kalkış ve yön değiştirme durumlarında ise bu değer, 18A'e kadar çıkabilmektedir. Bu yüzden, kanal başına 20A akım verebilen sürücüler kullanmamız gerekir .

1.3.BTS7960B 20 Amper Motor Sürücüsü

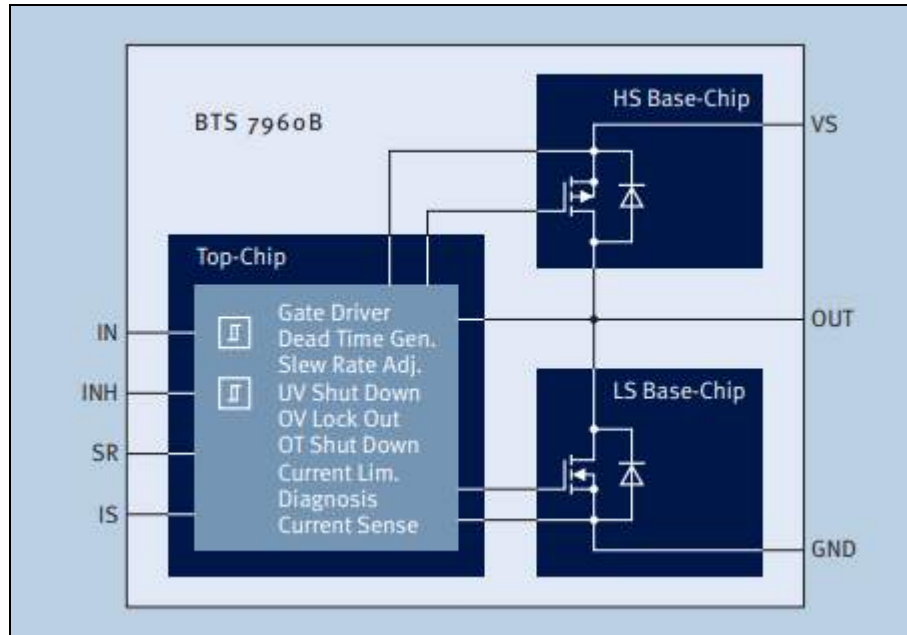
BTS7960B motor sürücü kartı sumo ve arazi robotlarında kullanılan, yüksek güçlü ve yüksek akım verebilen sürücü kartıdır. 28V'a kadar olan besleme gerilimi altında çalışabilen motor sürücü kartı, 20 ampere kadar akım çeken motorların kontrolünde kullanılabilir

Motor ve besleme bağlantıları;

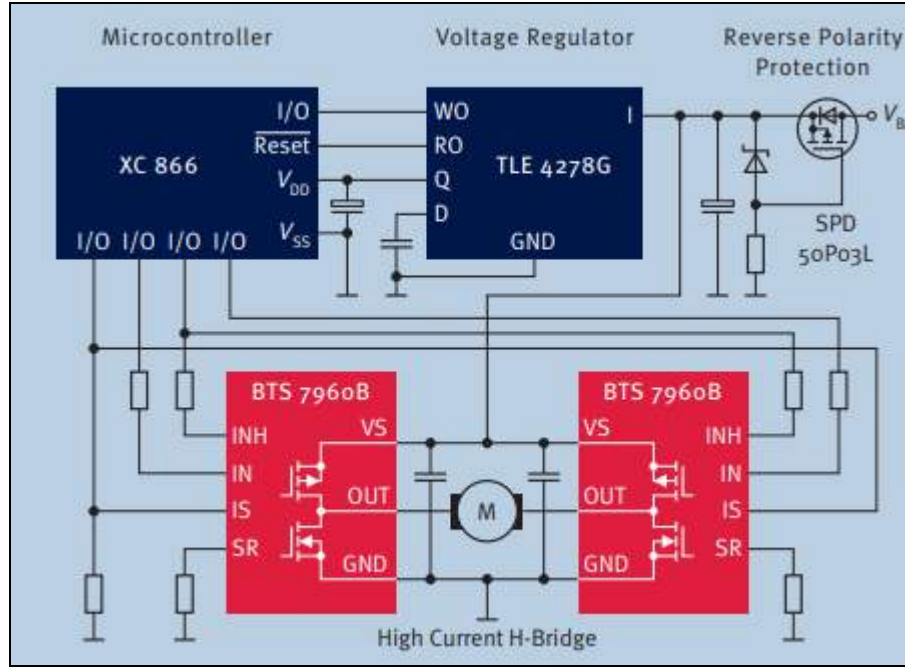
- B+ : Besleme Gerilimi (28V'a kadar)
- B- : Besleme Toprağı
- M+ : Motor+ Ucu
- M- : Motor- Ucu

Kontrol Pinleri;

- VCC: +5 VDC
- GND: Toprak Bağlantısı
- IS1,IS2: Akım Test Pinleri
- EN1,EN2: PWM girişleri
- IN1,IN2: Motor yön kontrol pinleri



Şekil 1.2.1. BTS7960B Blok Diyagram



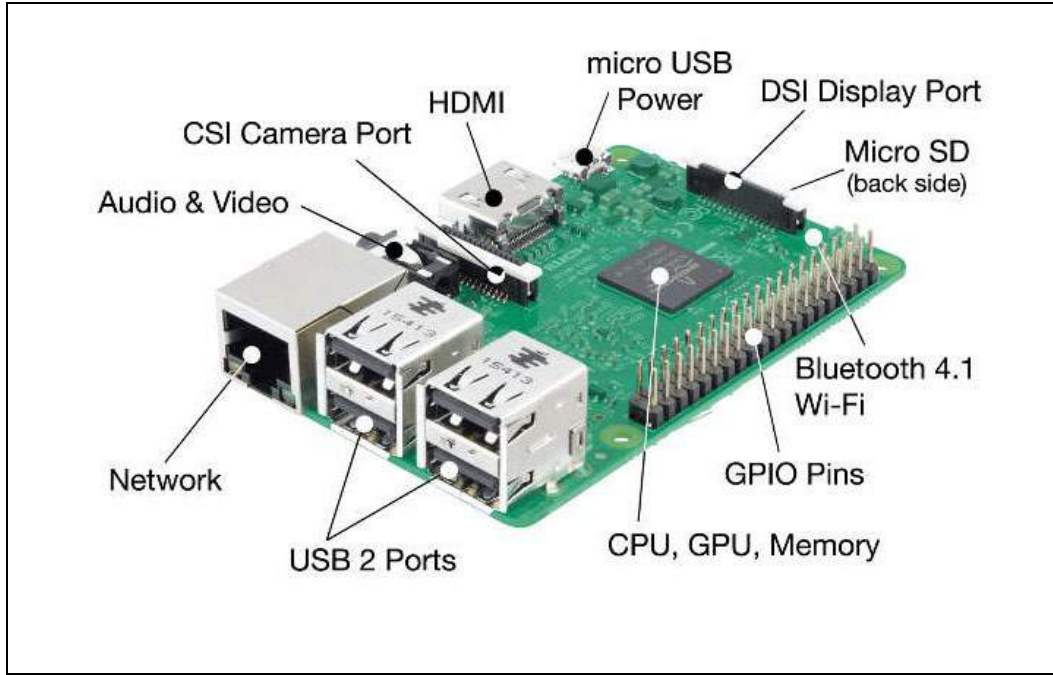
Şekil 1.2.2. BTS7960B Uygulama Örneği

Özellikler

- Min yüksek tepe akımı kapasitesi. 33 A
- Tip yol direnci. 16 m $^{\circ}$ @ 25 $^{\circ}$ C
- Düşük tipte sakin akım. 25 $^{\circ}$ C de 7 uA
- Aktif olarak 25 kHz'e kadar PWM özelliği
- Anahtarlama mod akım sınırlaması azaltılmış aşırı akımda güç tüketimi
- Mevcut anlamda durum bayrağı teşhisi kabiliyet
- Aşırı sıcaklık mandalla kapatıldı davranış
- Lojik seviye girişli sürücü devresi
- Optimize edilmiş EMI için ayarlanabilir dönüş hızı

1.4.Raspberry Pi 3 model:

Raspberry Pi genellikle robotik çalışmalarda kullanılan kredi kartı boyutunda bir bilgisayardır. Masaüstü veya dizüstü bilgisayarların yapabileceği çoğu işlemi yapabilme kapasitesine sahiptir.



Şekil 1.2.1. Raspberry Pi 3 Genel Görünümü

Raspberry Pi 3 Teknik 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
- Tüm Raspberry Pi uyumlu Linux dağıtımlarını ve Windows 10 IoT Core işletim sistemini destekler
- Boyutlar: 85 x 56 x 17 mm

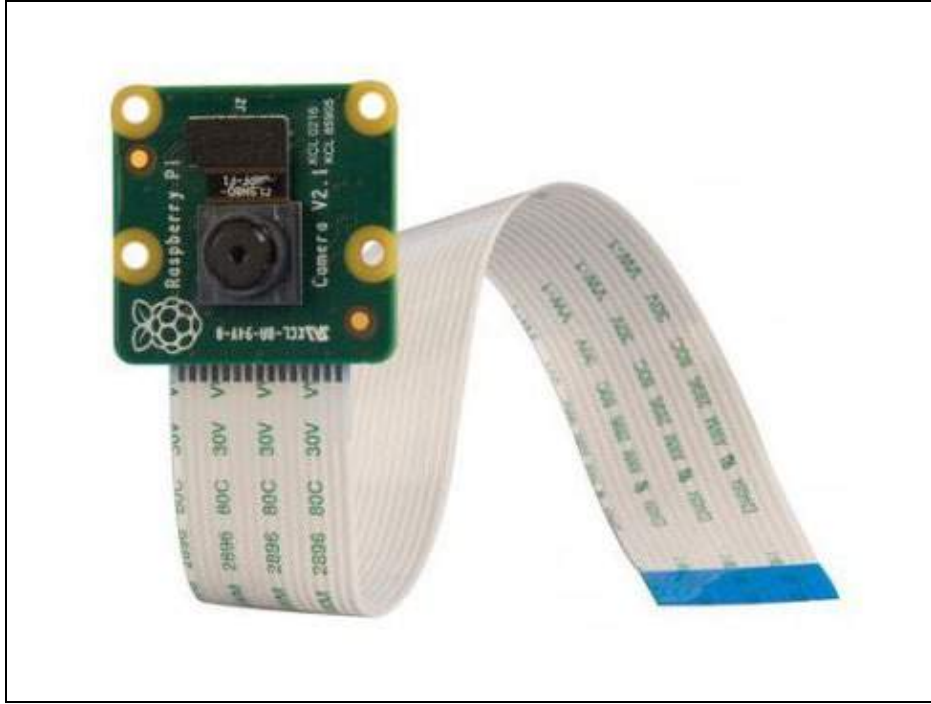
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Şekil 1.2.1. Raspberry Pi GPIO Başlığı

Raspberry Pi'nin şimdiye kadarki en güçlü modelidir. Model B+ ile aynı kart dizilimine sahip olmasına karşın, bu kartta ARMv7 mimarisine sahip 4 çekirdekli işlemci ve 1GB RAM bellek bulunmaktadır. Son zamanlarda bunla müteakiben pi 3 ve pi 4 modelleri de çıkmıştır. Bizim projemiz için pi 3 modeli en uygun model olacaktır. Bu modelin diğerlerinden en büyük farkı dahili Wi-Fi ve Bluetooth bağlantıya sahip olmasıdır. Ayrıca ARMv8 64-bit mimarisine sahip 4 çekirdekli işlemcisi, 1.2GHz frekansında çalışmaktadır ve 1GB RAM belleğe sahiptir.

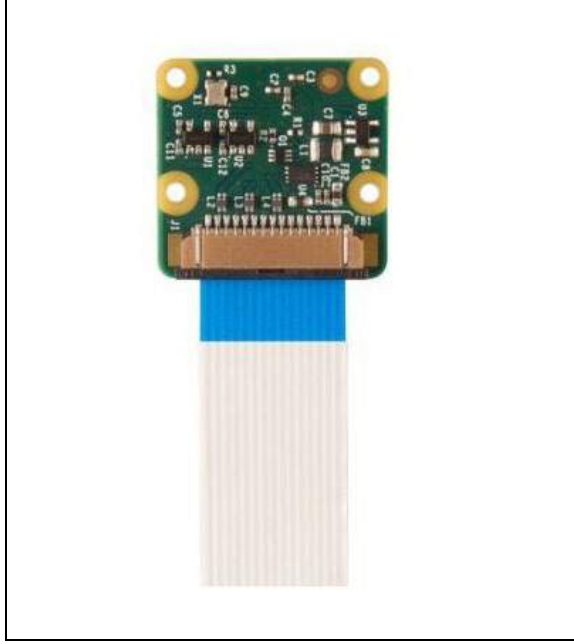
1.5.Raspberry Pi Camera

Raspberry Pi yüksek çözünürlüklü kamera(camera), üzerinde CSI konektörü bulunan tüm modeller ile uyumludur. Bu kamerayı, fotoğraf ve HD çözünürlükte video çekiminde kullanabiliriz. Time-lapse video ve ağır çekim için mükemmel bir tercihtir.



Şekil 1.5.1. Raspberry Pi Kamera Ön Görünüm

Kamera modülü üzerinde Sony tarafından üretilen yüksek hassasiyetli ve yüksek hızlı video desteği sunan IMX219PQ görüntü algılayıcı sensör bulunmaktadır. Ayrıca otomatik beyaz dengesi, pozlama telafisi kontrolü ve ortam ışığı algılama gibi özellikleri de bulunmaktadır.



Şekil 1.5.2. Raspberry Pi Kamera Arka Görünüm

Raspberry Pi Camera Özellikleri:

- Yüksek kaliteli görüntü algılama
- Büyük veri işleme kapasitesi
- 8 megapiksel sabit odak noktalı
- 1080p, 720p60 ve VGA90 destekli
- Sony IMX219PQ CMOS görüntü algılayıcı
- 15-pin şerit kablo

Kamera modülümüzü Pi'mize CSI (Camera Serial Interface) konektörü üzerinden bağlamalıyız. Bağlantı yapıldıktan sonra, Multi-Media Abstraction Layer (MMAL) veya Video for Linux (V4L) API'leri üzerinden kameraya erişiminiz mümkün olacaktır. Ayrıca internet üzerinden Picamera Python gibi farklı API'ler ile de kullanılabilir.

1.6. MZ80 80cm Menzilli Kızılötesi Sensör

MZ80 Ayarlanabilir Kızılötesi Sensör, 80cm'e kadar ölçüm yapabilen bir sensördür. Ölçüm aralığı üzerindeki potansiyometreden ayarlanabilir. Ayarlanan mesafe aralığına bir engel geldiği zaman dijital voltaj çıkışı verir.



Şekil 1.6.1. MZ80 80cm Menzilli Kızılötesi Sensör

Özellikler:

- Dijital çıkışlı, yüksek kaliteli endüstriyel sensördür.
- Arkasındaki trimpot ile menzili 3-80 cm arasında ayarlanabilir.
- NPN çıkışlıdır.
- 5V ile çalışır, tepki süresi oldukça düşüktür.(2 ms)
- Montajlama aparatıyla rahatça montajlanabilir.
- Arduino ve PIC projelerinizle tam uyumludur.

Kablo Bağlantıları:

Kahverengi +5V

Mavi: GND

Siyah: Data Çıkışı

Kırmızı: +5V

Yeşil: GND

Sarı: Data Çıkışı



Şekil 1.6.2. MZ80 Ayarlanabilir Kızılötesi Sensör Bağlantısı

Şekil 1.6.2 de görüldüğü gibi mavi kablo GND girişine, kahverengi kablo 5V'a ve sarı kablo data çıkışı olarak Raspberry'de 16. Pine jumper kablolar ile bağlandı.

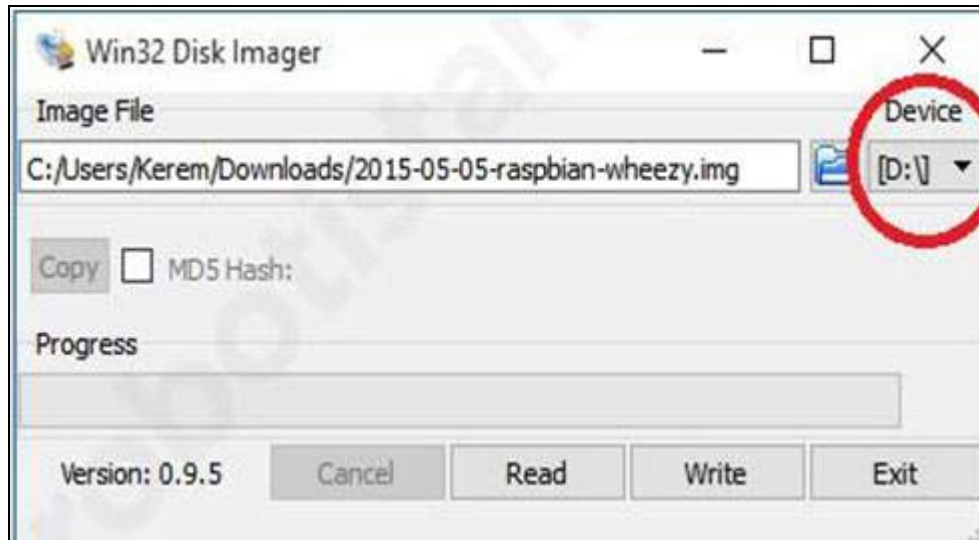
BÖLÜM 2

KURULUM

Projemizde açık kaynak kodlu bir Linux dağıtımını kullanmayı tercih ettiğimizden bu dağıtım Raspberry Pi için özelleştirilmiş ve bu tarz işlemler için en uygun dağıtım olan Raspbian olmalıdır.

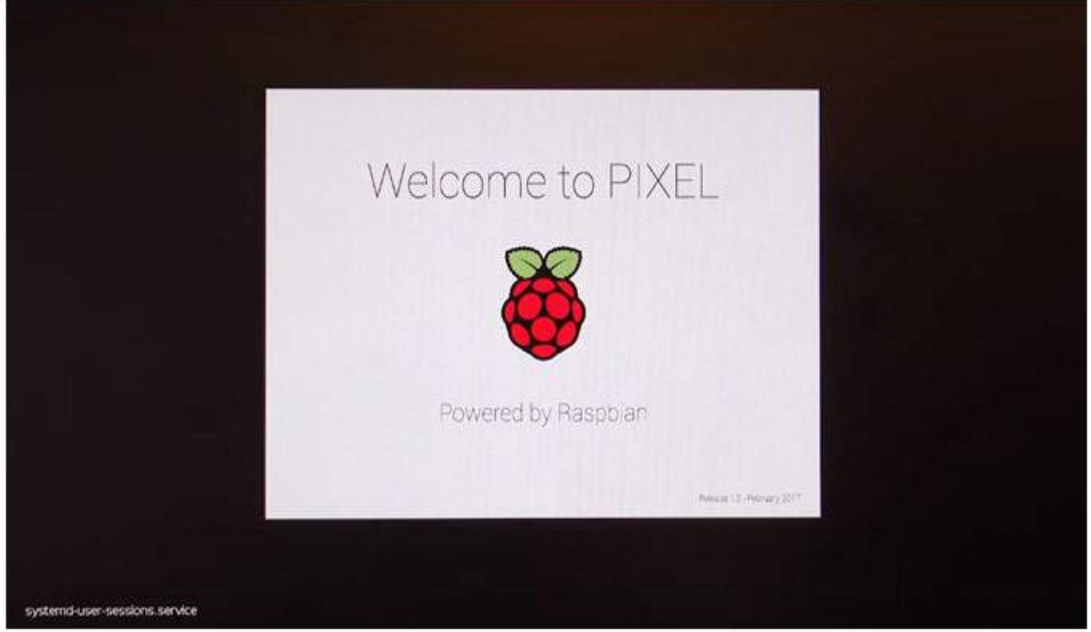
2.1. Raspbian Kurulumu

İlk önce SD kartımızı Windows yüklü bilgisayarımıza takıyoruz. Öncelik olarak bilgisayarımıza SD Cart Formartter programını kuruyoruz. Sonra bilgisayarımıza taktığımız SD kartımızı SD Cart Formatter programı sayesinde formatlıyoruz. Daha sonra Raspberry' nin kendi sitesinden <https://www.raspberrypi.org/downloads/raspbian/> linkten Raspbian işletim sistemin indirmemiz gerekiyor. Raspberry'nin kendi işletim sistemi olan Raspbian'ı indirdikten sonra Win32 Disk Manager program yardımı ile SD kartımızın içine yazdırıyoruz.

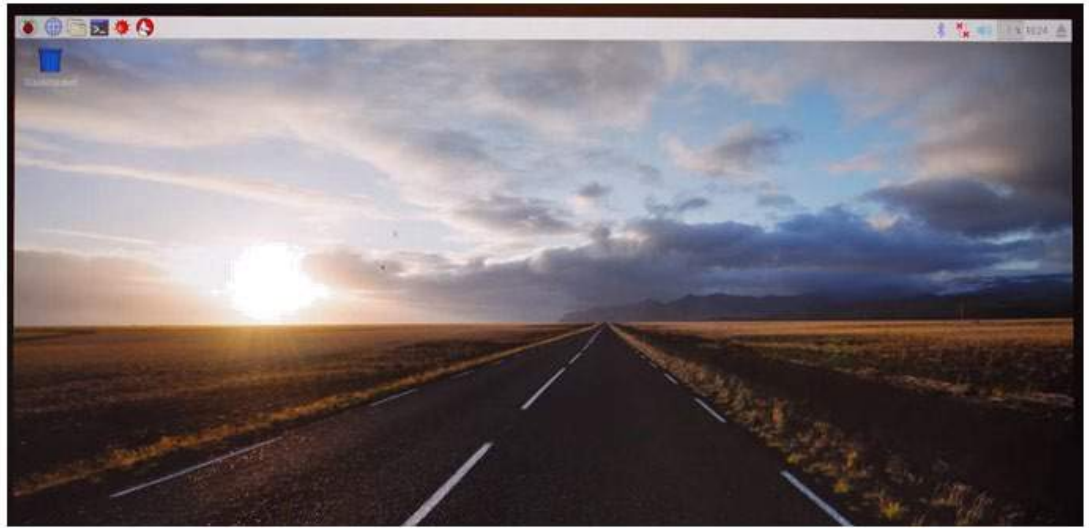


Şekil 2.1.1. Raspbian Kurulumu

Raspberry Pi mizde bulunan SD Cart girişine kartımızı takıp Raspberry Pi'yi bir adet HDMI kablo, kablosuz Mouse, Klavye ekipmanlarını ve güç adaptörünü takıp bilgisayarımıza bağlıyoruz. Ufak bi işletim sistemi yüklemesinden sonra masaüstümüze aşağıdaki şekildeki gibi kullanılabilir halde gelecek.



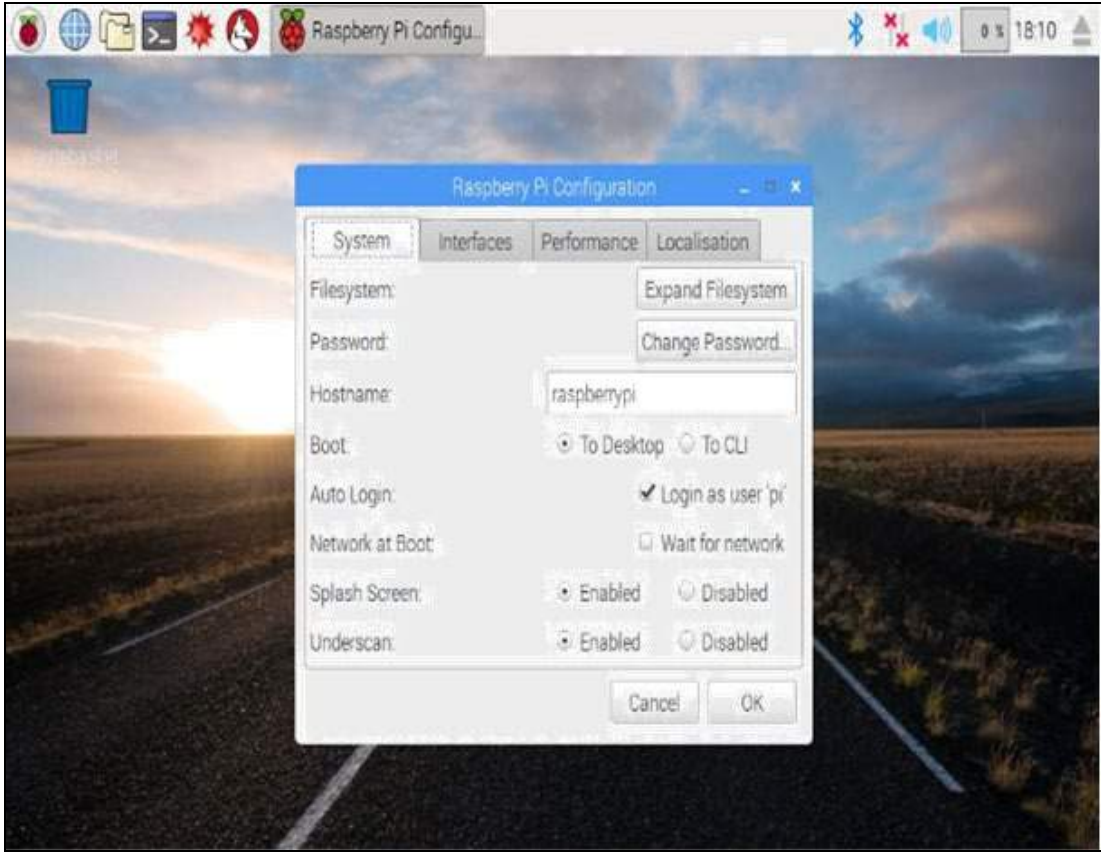
Şekil 2.1.2. Raspbian Kurulumu



Şekil 2.1.3. Raspbian Kurulumu

Raspbian işletim sistemini kurduktan sonra Raspbian'ın PIXEL sürümü ile birlikte Pi'miz artık doğrudan masaüstüne açılmakta. Ayarlamalarımızı Ana Menü -> Preferences -> Raspberry Pi Configuration adımları ile ulaşacağımız program ile yapılabilir. Bunun yanı sıra, komut satırından raspi-config programını da kullanmak mümkündür.

2.2. System Sekmesi



Şekil 2.2.1. Raspbian Temel Ayarlar

Expand filesystem: SD kartına imajın kartın boyundan küçük olması durumunda (örn. 8GB veya 16GB büyüklüğünde bir karta yükleme yapmamız durumunda) kartın tamamının kullanılmasını sağlar.

Password: Varsayılan kullanıcı olan “pi”nin şifresini değiştirmeye yarar. Raspbian’da varsayılan olarak pi kullanıcısının şifresi “raspberrypi”dir. Güvenlik sebebiyle değiştirmenizi önemle tavsiye ederim.

Hostname: Pi’de ağda tanınmak için kullandığı ismi buradan değiştirilebilir.

Boot: Raspbian’ın masaüstü (To Desktop) veya komut satırına (To CLI) açılmasını ayarlanır.

Autologin: Açılışta varsayılan kullanıcı olan “pi” olarak oturum açılmasını buradaki ayar ile değiştirilebilir, “pi” kullanıcısının şifresini değiştirilirse de otomatik oturum açma işlemi çalışmaya devam edecektir.

Network at Boot: Pi açılışta kablolu veya kablosuz ağ bağlantısını oluşturmayı beklemesini buradan tercih edilebilir. Masaüstüne açılmayı tercih edildiyse, bu seçenek kapalı tutulur.

Splashscreen: Raspbian’ın açılışta sistem mesajlarını göstermesini veya bunun yerine logo görünmesini ayarlanabilir.

Underscan: Eğer monitörünüzün görüntüsü ekrandan taşıyorsa bu seçeneği aktif hale getirilir.

2.3. Interfaces Sekmesi:



Şekil 2.3.1. Raspbian Temel Ayarlar

Bu sekmede, Pi’de bulunan yazılım ve donanım arayüzlerini açıp kapatılabilir. Varsayılan olarak tüm arayüzler kapalı durumdadır.

Camera: CSI konektörü aracılığıyla bağlanan kamerayı açıp kapatmada kullanılır.

SSH: Ağ üzerinden uzak komut satırı bağlantısı sağlar. SSH arayüzünü güvenlik sebebiyle “pi” kullanıcısının şifresini değiştirmeden aktifleştirilmesi gerekir.

VNC: Ağ üzerinden uzak masaüstü bağlantısı sağlar.

SPI: Donanımsal SPI bağlantısını aktifleştirir. GPIO pinleri aracılığı ile SPI bağlantısına sahip bir cihaz kullanmak istenirse (sensör, RFID okuyucu, LCD ekran v.b.) aktifleştirilmesi gerekir..

I2C: Donanımsal I2C bağlantısını aktifleştirir. GPIO pinleri aracılığı ile I2C bağlantısına sahip bir cihaz kullanılması istenirse (sensör, RTC entegresi v.b.) aktifleştirilmesi gerekir.

Serial: GPIO pinlerinde yer alan UART bağlantısı üzerinden komut satırına erişilmesini sağlar. UART’ı başka bir amaçla kullanılıyorsa (XBee veya GSM modülü v.b.) kapalı olması gerekir.

1-Wire: DS18B20 gibi 1-wire protokolünü kullanan sensörler kullanılması istenirse açmamız gerekir.

Remote GPIO: pigpio isimli servisi kontrol eder. Bu servis aracılığı ile ağ bağlantısı üzerinden Pi GPIO pinlerini kontrol edilebilir.

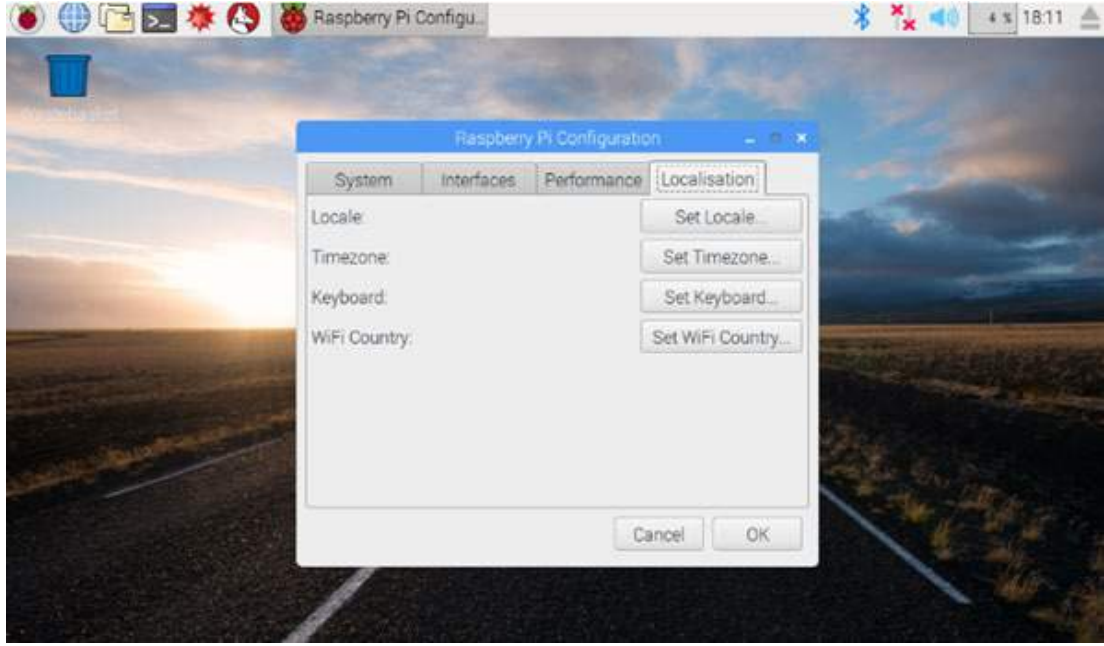
2.4. Performance Sekmesi:



Şekil 2.4.1. Raspbian Temel Ayarlar

Performance sekmesinde hız aşırtma ve grafik işlemciye ayrılan bellek miktarını değiştirilebilir.

2.5. Localisation Sekmesi:



Şekil 2.5.1. Raspbian Temel Ayarlar

Bu sekmede Raspbian'ın dil seçenekleri bulunmakta.

Locale: Sistemin dilini buradan değiştirilebilir. Türkçe'yi seçilebilir, fakat tüm programlar Türkçe desteği sunmadığından yüzde yüz Türkçe olmayacaktır.

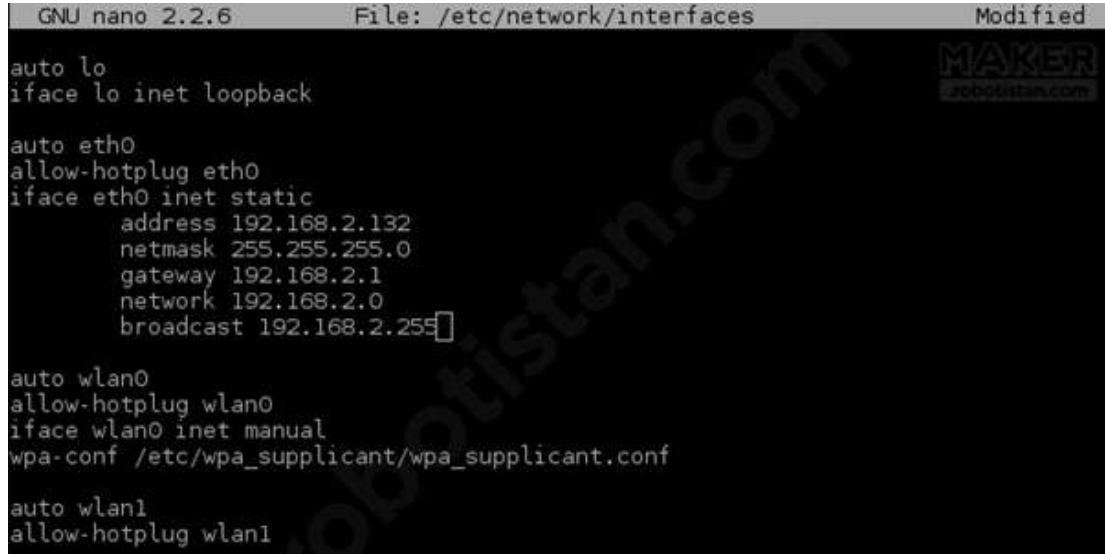
Timezone: Sistemin saat diliminin seçilmesini sağlar. Türkiye için artık kalıcı yaz saati uygulaması geçerli olduğundan Etc/GMT+3 seçilebilir.

Keyboard: Klavyenin tuş dizilimini bu seçenekten değiştirilebilir. Türkçe Q dizilimi için Turkey/Turkish, Türkçe F dizilimi için Turkey/Turkish (F) seçilebilir.

WiFi Country: WiFi kanallarının frekansları farklı ülkelerin kanunlarına göre belirlendiği için buradan Türkiye seçilmesi gerekir.

Program, ayarların geçerli olması için Pi yeniden başlatılır.

2.6. Ağ Ayarları:

A screenshot of the GNU nano 2.2.6 text editor. The title bar shows 'File: /etc/network/interfaces' and 'Modified'. The editor content shows the configuration for the eth0 interface. It is set to 'auto eth0' and 'allow-hotplug eth0'. The interface is configured as 'iface eth0 inet static' with the following settings: 'address 192.168.2.132', 'netmask 255.255.255.0', 'gateway 192.168.2.1', 'network 192.168.2.0', and 'broadcast 192.168.2.255'. Below this, there are configurations for wlan0 and wlan1, both set to 'auto' and 'allow-hotplug' with 'manual' interface type. The wlan0 configuration also points to a wpa-conf file. A 'MAKER' watermark is visible in the top right corner of the editor window.

```
GNU nano 2.2.6 File: /etc/network/interfaces Modified
auto lo
iface lo inet loopback

auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.2.132
    netmask 255.255.255.0
    gateway 192.168.2.1
    network 192.168.2.0
    broadcast 192.168.2.255

auto wlan0
allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

auto wlan1
allow-hotplug wlan1
```

Şekil 2.6.1. Raspbian Temel Ayarlar

Gerekli temel ayarları yaptıktan sonra Raspberry Pi'nin ağ ayarlarını yapılması gerekir.

Raspberry Pi'yi ağa bağlamak için en kolay yol kablolu bağlantıdır. Modemin herhangi bir boş Ethernet portuna bağlanacak ve kablunun diğer ucunu Raspberry Pi'ye ttakıldığında bağlantı otomatik olarak gerçekleşir.

Raspberry Pi kablolu bağlatın da sabit IP vermekistenirse console ekranına `sudo nano /etc/network/interfaces` komutu ile açılacak olan dosyadaki `iface eth0 inet manual` satırını “`iface eth0 inet static address 192.168.2.132 netmask 255.255.255.0 gateway 192.168.2.1 network 192.168.2.0 broadcast 192.168.2.255`” olarak değiştirilir. Buradaki adress kısmına IP adresini, gateway kısmına modem yada router'ın IP adresini, network kısmına IP adresinin ilk 3 hanesini son hanesini 0 yazarak ve broadcast kısmına da IP adresinin ilk 3 hanesini son hanesini 255 yaparak girilir. CTRL + O ile yapılan değişiklikler kaydedilir, CTRL +X ile nano programından kapatılır. Grafik arayüzden sabit ip ayarları için resimde görüldüğü gibi adımları takip edilir.

Kablolu ağı seçmek için en sağdaki seçim kutusundan “eth0”ı seçilir.

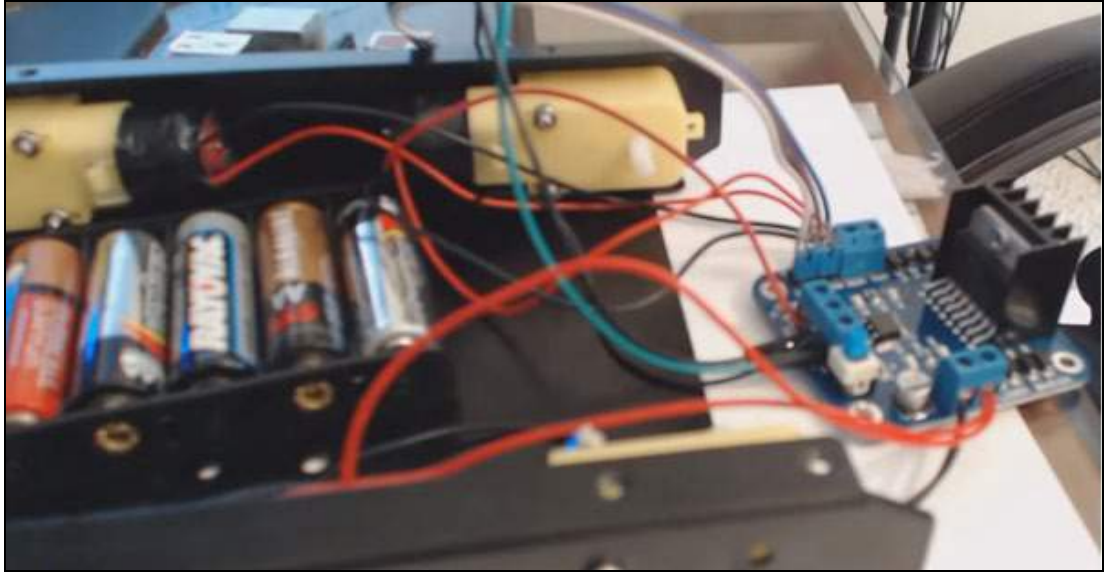
Kablosuz bağlantı için ise, tüm yazılımlar güncel Raspbian sürümünde kurulu olarak gelmektedir. Gerekli olan Raspberry Pi uyumlu bir USB Wi-Fi adaptörüdür. http://elinux.org/RPi_USB_Wi-Fi_Adapters#Working_USB_Wi-Fi_Adapters adresinden Raspberry Pi ile sorunsuzca kullanılacak USB Wi-Fi adaptörlerinin listesine erişilebilir. Bazı adaptörler için sürücü kurulumu gerekebilir.

BÖLÜM 3

RASPBERRY Pİ 3 KURULUM VE KODLANMASI

3.1. Rasberry Pi3 ile Donanım Bağlantıları:

Raspberry Pi'ye, GPIO pinlerine doğru şekilde yerleştirdikten sonra Raspberry Pİ açılarak ilk senaryomuzu oluşturulur.



Şekil 3.1. Donanım Bağlantıları

3.2. Rasberry Pi3 Kodlaması:

- RPI GPIO

```
Sudo apt-get install python-RPi.GPIO
```

```
Ls
```

```
mkdir robotics
```

```
cd robotics
```

```
sudo nano robot1.py
```

```
İmport RPi.GPIO as gpio
```

```
İmport time
```

```
gpio.setmode (gpio.BOARD)
gpio.setup (7,gpio.OUT)
gpio.setup (11,gpio.OUT)
gpio.setup (13,gpio.OUT)
gpio.setup (15,gpio.OUT)
time.sleep (0.5)
gpio.cleanup()
```

robot1.py dosyasını kapatarak tekrar komut satırına ;

```
sudo nano robot1.py
sudo python robot1.py
sudo python robot1.py
sudo python robot1.py
sudo python robot2.py
İmport RPi.GPIO as gpio
İmport time
def init()
gpio.setmode (gpio.BOARD)
gpio.setup (7,gpio.OUT)
gpio.setup (11,gpio.OUT)
gpio.setup (13,gpio.OUT)
gpio.setup (15,gpio.OUT)
def forward (tf) => ileri gidebilmek için
init()
gpio.output(7,False)
gpio.output(11,True)
gpio.output(13,True)
gpio.output(15,False)
time.sleep(tf)
gpio.cleanup()
def reverse (tf) => geri gidebilmek için
init()
gpio.output(7,True)
```



```
gpio.output(11,False)
gpio.output(13,False)
gpio.output(15,True)
time.sleep(tf)
gpio.cleanup()
forward(1)
reverse(1)
```

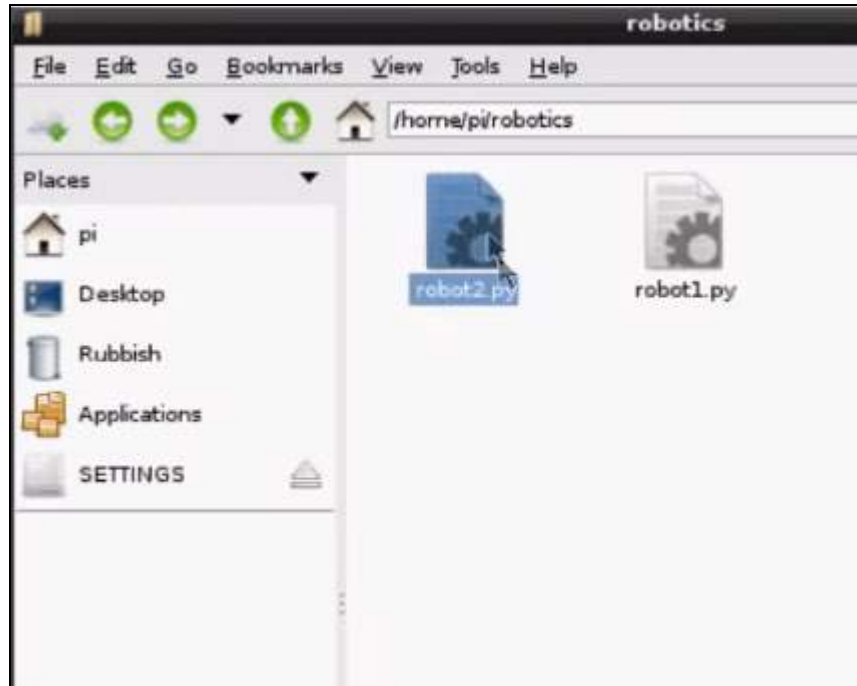
- Yön Kodlarının Yazılması

```
def turn_left(tf)
init()
gpio.output(7,True)
gpio.output(11,True)
gpio.output(13,True)
gpio.output(15,False)
time.sleep(tf)
gpio.cleanup()
def turn_right(tf) => sağa dönebilmemiz için
init()
gpio.output(7,True)
gpio.output(11,False)
gpio.output(13,False)
gpio.output(15,True)
time.sleep(tf)
gpio.cleanup()
def pivot_left(tf) => Sola keskin dönüş
init()
gpio.output(7,True)
gpio.output(11,False)
gpio.output(13,True)
gpio.output(15,False)
time.sleep(tf)
```

```
gpio.cleanup()
def pivot_right(tf)=>Sağa keskin dönüş
init()
gpio.output(7,False)
gpio.output(11,True)
gpio.output(13,False)
gpio.output(15,True)
time.sleep(tf)
gpio.cleanup()
```

Akıllı etmenin hareket fonksiyonlarını sağladıktan sonra Kullanıcı kontrolü eklenmesi gerekmektedir. Kontrolün klavyeden W harfine ileri hareket, S harfine geri hareket, D harfine sağa hareket, A harfine sola hareket, Q harfine pivot left, E harfine ise pivot right yapılmasını sağlanır.

File Manager seçeneğinde;



Şekil 3.2.1. Ekran Görüntüsü

Açılan pencereye robot2.py 'da yazılan kodları kopyalayıp python dosyası açıp aşağıdaki kodlar yazılır.

```

import RPi.GPIO as gpio
import time
import sys => gerekli olan 2 tane kütüphane eklendi
import Tkinter as tk

def init()
    gpio.setmode (gpio.BOARD)
    gpio.setup (7,gpio.OUT)
    gpio.setup (11,gpio.OUT)
    gpio.setup (13,gpio.OUT)
    gpio.setup (15,gpio.OUT)
    def forward (tf)
        gpio.output(7,False)
        gpio.output(11,True)
        gpio.output(13,True)
        gpio.output(15,False)
        time.sleep(tf)
        gpio.cleanup()
    def reverse (tf)
        gpio.output(7,True)
        gpio.output(11,False)
        gpio.output(13,False)
        gpio.output(15,True)
        time.sleep(tf)
        gpio.cleanup()
    def turn_left(tf)
        gpio.output(7,True)
        gpio.output(11,True)
        gpio.output(13,True)
        gpio.output(15,False)
        time.sleep(tf)
        gpio.cleanup()
    def turn_right(tf)
        gpio.output(7,True)

```

```

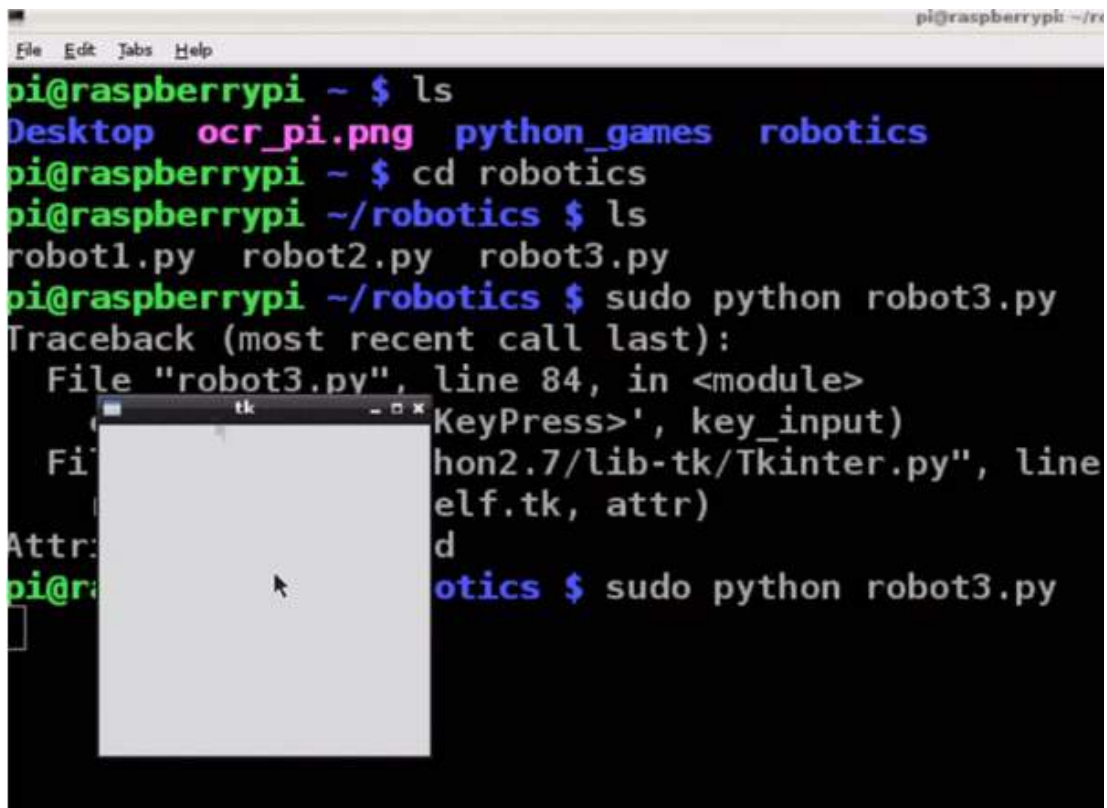
gpio.output(11,False)
gpio.output(13,False)
gpio.output(15,True)
time.sleep(tf)
gpio.cleanup()
def pivot_left(tf)
gpio.output(7,True)
gpio.output(11,False)
gpio.output(13,True)
gpio.output(15,False)
time.sleep(tf)
gpio.cleanup()
def pivot_right(tf)
gpio.output(7,False)
gpio.output(11,True)
gpio.output(13,False)
gpio.output(15,True)
time.sleep(tf)
gpio.cleanup()
def key_input(event):
init()
print 'Key:', event.char
key_press=event.char
sleep_time=0.030
if key_press.lower() == 'w':
    forward(sleep_time)
elif key_press.lower() == 's':
    reverse(sleep_time)
elif key_press.lower() == 'a':
    turn_left(sleep_time)
elif key_press.lower() == 'd':
    turn_right(sleep_time)
elif key_press.lower() == 'q':

```

```

    pivot_left(sleep_time)
elif key_press.lower() == 'e':
    pivot_right(sleep_time)
else:
    pass
command = tk.Tk()
command.bind('<KeyPress>', key_input)
command.mainloop()

```



The screenshot shows a terminal window on a Raspberry Pi. The user has navigated to the 'robotics' directory and run 'sudo python robot3.py'. The terminal output shows a traceback error: 'File "robot3.py", line 84, in <module> ...'. A small Tkinter window titled 'tk' is visible in the foreground, showing a mouse cursor. The terminal window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'.

```

pi@raspberrypi ~ $ ls
Desktop ocr_pi.png python_games robotics
pi@raspberrypi ~ $ cd robotics
pi@raspberrypi ~/robotics $ ls
robot1.py robot2.py robot3.py
pi@raspberrypi ~/robotics $ sudo python robot3.py
Traceback (most recent call last):
  File "robot3.py", line 84, in <module>
    KeyPress>', key_input)
  File "/usr/lib/python2.7/lib-tk/Tkinter.py", line
    elif tk, attr)
  File "robot3.py", line 84, in <module>
    KeyPress>', key_input)
pi@raspberrypi ~/robotics $ sudo python robot3.py

```

Şekil 3.2.2. Sistemin Çalışması

Yazılan kod ile birlikte akıllı etmenin klavyeden uzaktan bir kullanıcı ile artık kontrol edilebilir.

- MZ80 Ayarlanabilir Kızılötesi Sensör Kodları

```
import RPi.GPIO as GPIO

import time
GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)
GPIO.setup(16, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(36,GPIO.OUT)
GPIO.setup(33,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)

while True:

    if GPIO.input(16) == False :
        GPIO.cleanup
        print("engel var")
        print("ARAC DURUYOR")
        GPIO.output(18,0)
        GPIO.output(36,0)
        GPIO.output(33,1)
        GPIO.output(13,1)
        GPIO.cleanup
        time.sleep(0.4)
        time.sleep(0.4)
    #    print("ARAC GERI GIDIYOR")
    #    GPIO.output(18,0)
    #    GPIO.output(36,1)
    #    GPIO.output(22,0)
    #    GPIO.output(11,1)
    #
    #    time.sleep(0.4)
    print("SOLA DONUYOR")
    GPIO.output(18,0)
    GPIO.output(36,1)
    GPIO.output(33,1)
    GPIO.output(13,0)
    time.sleep(0.5)

    if GPIO.input(16) == True :
        print("engel yok")
        print("ARAC ILERLIYOR")
        GPIO.output(18,1)
```

```
GPIO.output(36,0)
GPIO.output(33,0)
GPIO.output(13,1)
GPIO.cleanup
```

BÖLÜM 4

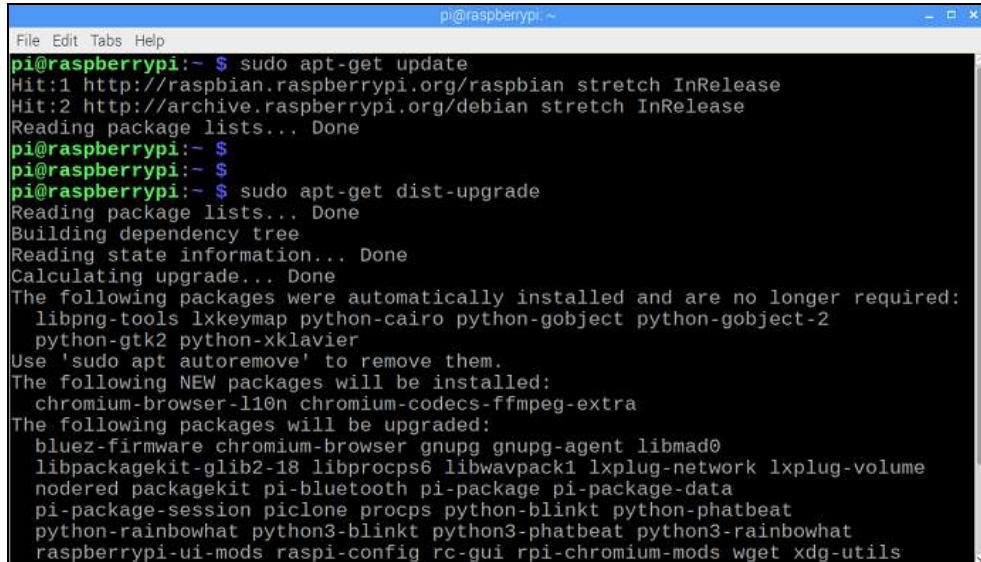
NESNE TESPİTİ

4.1. Raspberry Pi güncellenmesi

Raspberry Pi'nin tamamen güncellenmesi için Terminal penceresine;

`sudo apt-get update`

`sudo apt-get dist-upgrade` kodları yazılır.



```
pi@raspberrypi:~$ sudo apt-get update
Hit:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease
Hit:2 http://archive.raspberrypi.org/debian stretch InRelease
Reading package lists... Done
pi@raspberrypi:~$ sudo apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libpng-tools lxkeymap python-cairo python-gobject python-gobject-2
  python-gtk2 python-xklavier
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  chromium-browser-110n chromium-codecs-ffmpeg-extra
The following packages will be upgraded:
  bluez-firmware chromium-browser gnutls gnutls-agent libmad0
  libpackagekit-glib2-18 libprocps6 libwavpack1 lxplug-network lxplug-volume
  nodered packagekit pi-bluetooth pi-package pi-package-data
  pi-package-session piclone procps python-blink python-phatbeat
  python-rainbowhat python3-blink python3-phatbeat python3-rainbowhat
  raspberrypi-ui-mods raspi-config rc-gui rpi-chromium-mods wget xdg-utils
```

Şekil 4.1.1. Raspberry Pi Kurulum Ekranı

4.2. TensorFlow Yüklenmesi

TensorFlow dosyasını https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/installation.md linkinden Raspberry Pi'ye indirip TensorFlow kurulumu yapılır. TensorFlow'un LibAtlas paketine aşağıdaki komutu vererek kurulur.

`sudo apt-get install libatlas-base-dev`


```
pi@raspberrypi: ~/tf
File Edit Tabs Help
Setting up libreoffice-gtk2 (1:5.2.7-1+rpil+deb9u3) ...
Setting up python3-uno (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-base-drivers (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-systray (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-gtk (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-draw (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-avmedia-backend-gstreamer (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-style-galaxy (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-java-common (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-impress (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-math (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-base-core (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-sdbc-hsqldb (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-calc (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-writer (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-base (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice-report-builder-bin (1:5.2.7-1+rpil+deb9u3) ...
Setting up libreoffice (1:5.2.7-1+rpil+deb9u3) ...
Processing triggers for libc-bin (2.24-11+deb9u3) ...
pi@raspberrypi:~ $
pi@raspberrypi:~ $ mkdir tf
pi@raspberrypi:~ $ cd tf
pi@raspberrypi:~/tf $ wget https://github.com/lhelontra/tensorflow-on-arm/releases/download/v1.8.0/tensorflow-1.8.0-cp35-none-linux_armv7l.whl
```

Şekil 4.2.1. TensorFlow Kurulum Ekranı

```
pi@raspberrypi: ~/tf
File Edit Tabs Help
Location: https://github-production-release-asset-2e65be.s3.amazonaws.com/99042851/b7008c04-4d6c-11e8-986a-e92fa32379d9?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20180629%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180629T043727Z&X-Amz-Expires=300&X-Amz-Signature=6343a603f364a9dce2b8136c1050cd0d048a702ec8e823ba283484aeb2b5523e&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dtensorflow-1.8.0-cp35-none-linux_armv7l.whl&response-content-type=application%2Foctet-stream [following]
--2018-06-29 04:37:27-- https://github-production-release-asset-2e65be.s3.amazonaws.com/99042851/b7008c04-4d6c-11e8-986a-e92fa32379d9?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20180629%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180629T043727Z&X-Amz-Expires=300&X-Amz-Signature=6343a603f364a9dce2b8136c1050cd0d048a702ec8e823ba283484aeb2b5523e&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dtensorflow-1.8.0-cp35-none-linux_armv7l.whl&response-content-type=application%2Foctet-stream
Resolving github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.216.129.187
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|52.216.129.187|:443... connected
HTTP request sent, awaiting response... 200 OK
Length: 53425471 (51M) [application/octet-stream]
Saving to: 'tensorflow-1.8.0-cp35-none-linux_armv7l.whl'

tensorflow-1.8.0-cp 92%[=====>] 47.19M 1.89MB/s eta 2s
```

Şekil 4.2.2. TensorFlow Kurulum Ekranı

TensorFlow için gerekli kütüphaneler ve aşağıdaki API'sı yüklenir.

Protobuf 3.0.0

Python-tk

Yastık 1.0

lxml

tf-slim (<https://github.com/google-research/tf-slim>)

ince ("tensorflow / modeller / araştırma /" kasasında bulunur)

Jupyter dizüstü bilgisayar

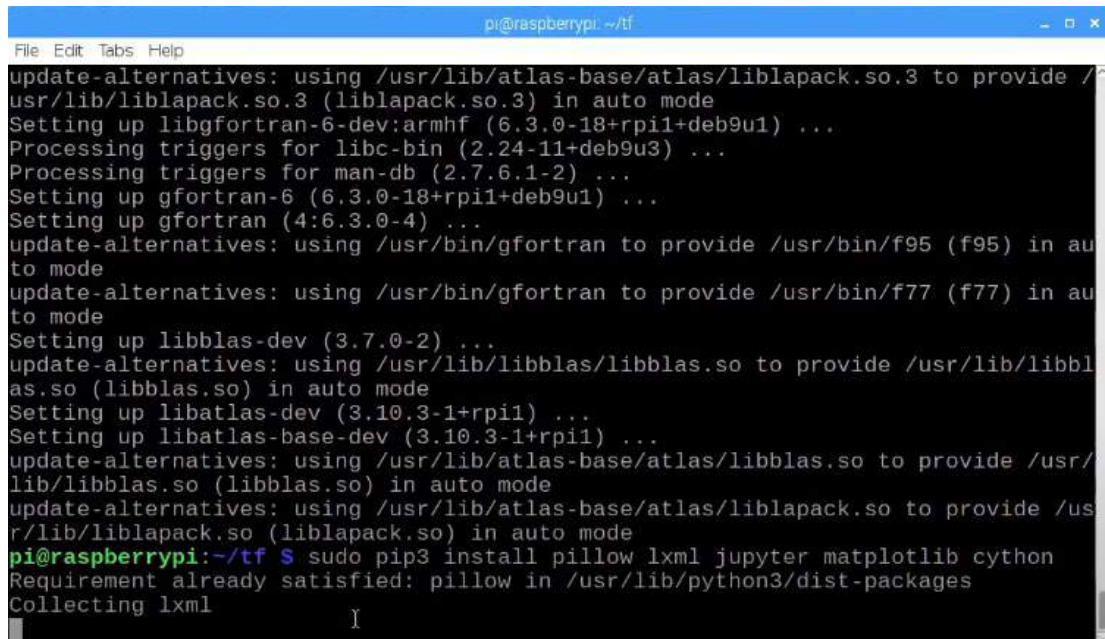
matplotlib

Tensorflow (1.15.0)

Cython

contextlib2

cocoapi



```
pi@raspberrypi: ~/tf
File Edit Tabs Help
update-alternatives: using /usr/lib/atlas-base/atlas/liblapack.so.3 to provide /
usr/lib/liblapack.so.3 (liblapack.so.3) in auto mode
Setting up libgfortran-6-dev:armhf (6.3.0-18+rpi1+deb9u1) ...
Processing triggers for libc-bin (2.24-11+deb9u3) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up gfortran-6 (6.3.0-18+rpi1+deb9u1) ...
Setting up gfortran (4:6.3.0-4) ...
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f95 (f95) in au
to mode
update-alternatives: using /usr/bin/gfortran to provide /usr/bin/f77 (f77) in au
to mode
Setting up libblas-dev (3.7.0-2) ...
update-alternatives: using /usr/lib/libblas/libblas.so to provide /usr/lib/libbl
as.so (libblas.so) in auto mode
Setting up libatlas-dev (3.10.3-1+rpi1) ...
Setting up libatlas-base-dev (3.10.3-1+rpi1) ...
update-alternatives: using /usr/lib/atlas-base/atlas/libblas.so to provide /usr/
lib/libblas.so (libblas.so) in auto mode
update-alternatives: using /usr/lib/atlas-base/atlas/liblapack.so to provide /us
r/lib/liblapack.so (liblapack.so) in auto mode
pi@raspberrypi:~/tf $ sudo pip3 install pillow lxml jupyter matplotlib cython
Requirement already satisfied: pillow in /usr/lib/python3/dist-packages
Collecting lxml
I
```

Şekil 4.2.2. TensorFlow Kütüphane Yükleme

```
sudo pip3 install pillow lxml jupyter matplotlib cython
```

```
sudo apt-get install python-tk
```

komutu yazılarak TensorFlow'un Raspberry ye kurulumu tamalanır.

4.3.OPENCV Yüklenmesi

TensorFlow'un nesne algılama örnekleri genellikle görüntüleri görüntülemek için matplotlib kullanır, OpenCV kullanılmasıdaki amaç daha az hataya açık olması ve GitHub deposundaki nesne algılama komut dosyalarının OpenCV kullanıyor olmasıdır.

```
sudo apt-get update
```

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
```

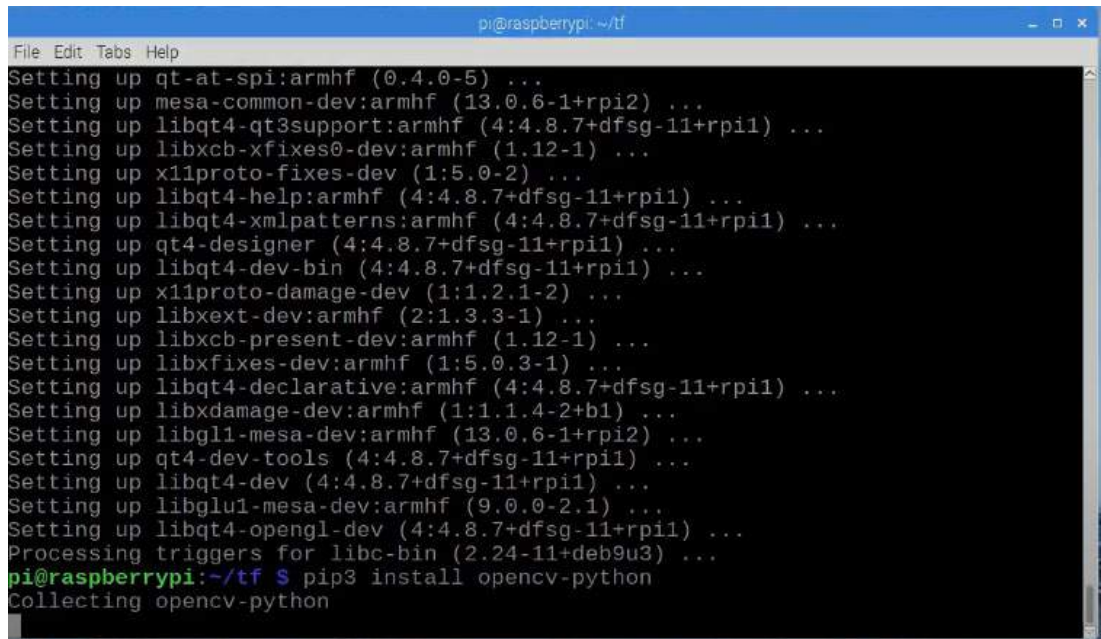
```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

```
sudo apt-get install libxvidcore-dev libx264-dev
```

```
sudo apt-get install qt4-dev-tools libatlas-base-dev
```

yukarıdaki komutlar yüklendikten sonra OpenCV kurulumu yapılır.

```
sudo pip3 install opencv-python
```



```
pi@raspberrypi: ~/tf
File Edit Tabs Help
Setting up qt-at-spi:armhf (0.4.0-5) ...
Setting up mesa-common-dev:armhf (13.0.6-1+rpi2) ...
Setting up libqt4-qt3support:armhf (4:4.8.7+dfsg-11+rpi1) ...
Setting up libxcb-xfixes0-dev:armhf (1.12-1) ...
Setting up x11proto-fixes-dev (1:5.0-2) ...
Setting up libqt4-help:armhf (4:4.8.7+dfsg-11+rpi1) ...
Setting up libqt4-xmlpatterns:armhf (4:4.8.7+dfsg-11+rpi1) ...
Setting up qt4-designer (4:4.8.7+dfsg-11+rpi1) ...
Setting up libqt4-dev-bin (4:4.8.7+dfsg-11+rpi1) ...
Setting up x11proto-damage-dev (1:1.2.1-2) ...
Setting up libxext-dev:armhf (2:1.3.3-1) ...
Setting up libxcb-present-dev:armhf (1.12-1) ...
Setting up libxfixes-dev:armhf (1:5.0.3-1) ...
Setting up libqt4-declarative:armhf (4:4.8.7+dfsg-11+rpi1) ...
Setting up libxdamage-dev:armhf (1:1.1.4-2+b1) ...
Setting up libgl1-mesa-dev:armhf (13.0.6-1+rpi2) ...
Setting up qt4-dev-tools (4:4.8.7+dfsg-11+rpi1) ...
Setting up libqt4-dev (4:4.8.7+dfsg-11+rpi1) ...
Setting up libglu1-mesa-dev:armhf (9.0.0-2.1) ...
Setting up libqt4-opengl-dev (4:4.8.7+dfsg-11+rpi1) ...
Processing triggers for libc-bin (2.24-11+deb9u3) ...
pi@raspberrypi:~/tf $ pip3 install opencv-python
Collecting opencv-python
```

Şekil 4.3.1. OpenCV Kurulum Ekranı


```
pi@raspberrypi: ~/tf
File Edit Tabs Help
File "/usr/share/python-wheels/CacheControl-0.11.7-py2.py3-none-any.whl/cachecontrol/adapters.py", line 47, in send
    resp = super(CacheControlAdapter, self).send(request, **kw)
File "/usr/share/python-wheels/requests-2.12.4-py2.py3-none-any.whl/requests/adapters.py", line 423, in send
    timeout=timeout
File "/usr/share/python-wheels/urllib3-1.19.1-py2.py3-none-any.whl/urllib3/connectionpool.py", line 643, in urlopen
    _stacktrace=sys.exc_info()[2])
File "/usr/share/python-wheels/urllib3-1.19.1-py2.py3-none-any.whl/urllib3/util/retry.py", line 315, in increment
    total -= 1
TypeError: unsupported operand type(s) for -=: 'Retry' and 'int'
pi@raspberrypi:~/tf $ pip3 install opencv-python
Collecting opencv-python
  Using cached https://www.piwheels.org/simple/opencv-python/opencv_python-3.4.1.15-cp35-cp35m-linux_armv7l.whl
Collecting numpy>=1.12.1 (from opencv-python)
  Downloading https://www.piwheels.org/simple/numpy/numpy-1.14.5-cp35-cp35m-linux_armv7l.whl (6.3MB)
100% |████████████████████████████████████████| 6.3MB 53kB/s
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.14.5 opencv-python-3.4.1.15
pi@raspberrypi:~/tf $
```

Şekil 4.3.2. OpenCV Kurulum Ekranı

4.4.Protobuf Derlenmesi ve Yüklenmesi

TensorFlow nesne algılama API'sı, Google'ın Protokol Arabelleği veri biçimini uygulayan bir paket olan Protobuf'u kullanır.

sudo apt-get install protobuf-compiler

```
pi@raspberrypi: ~/tf
File Edit Tabs Help
File "/usr/share/python-wheels/requests-2.12.4-py2.py3-none-any.whl/requests/adapters.py", line 423, in send
    timeout=timeout
File "/usr/share/python-wheels/urllib3-1.19.1-py2.py3-none-any.whl/urllib3/connectionpool.py", line 643, in urlopen
    _stacktrace=sys.exc_info()[2])
File "/usr/share/python-wheels/urllib3-1.19.1-py2.py3-none-any.whl/urllib3/util/retry.py", line 315, in increment
    total -= 1
TypeError: unsupported operand type(s) for -=: 'Retry' and 'int'
pi@raspberrypi:~/tf $ pip3 install opencv-python
Collecting opencv-python
  Using cached https://www.piwheels.org/simple/opencv-python/opencv_python-3.4.1.15-cp35-cp35m-linux_armv7l.whl
Collecting numpy>=1.12.1 (from opencv-python)
  Downloading https://www.piwheels.org/simple/numpy/numpy-1.14.5-cp35-cp35m-linux_armv7l.whl (6.3MB)
100% |████████████████████████████████████████| 6.3MB 53kB/s
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.14.5 opencv-python-3.4.1.15
pi@raspberrypi:~/tf $ sudo apt-get install autoconf automake libtool curl
Reading package lists... Done
Building dependency tree... 50%
```

Şekil 4.4.1. Protobuf Yükleme Ekranı

Yüklü olduğunu doğrulamak için bittiğinde protoc --version komutunu çalıştırılarak Libprotoc 3.6.1 veya benzeri bir yanıt alınması beklenir.

```
pi@raspberrypi:~/tf $ wget https://github.com/google/protobuf/releases/download/v3.6.0/protobuf-all-3.6.0.tar.gz
--2018-07-01 00:31:26-- https://github.com/google/protobuf/releases/download/v3.6.0/protobuf-all-3.6.0.tar.gz
Resolving github.com (github.com)... 192.30.255.112, 192.30.255.113
Connecting to github.com (github.com)|192.30.255.112|:443... connected.
HTTP request sent, awaiting response...
```

```
HTTP request sent, awaiting response... 200 OK
Length: 6727974 (6.4M) [application/octet-stream]
Saving to: 'protobuf-all-3.6.0.tar.gz'

protobuf-all-3.6.0. 100%[=====>] 6.42M 1.82MB/s in 4.8s

2018-07-01 00:31:32 (1.33 MB/s) - 'protobuf-all-3.6.0.tar.gz' saved [6727974/6727974]
```

```
pi@raspberrypi:~/tf $ tar -zxvf protobuf-all-3.6.0.tar.gz
```

```
protobuf-3.6.0/java/util/src/main/java/com/google/protobuf/util/Durations.java
protobuf-3.6.0/java/util/src/main/java/com/google/protobuf/util/TimeUtil.java
protobuf-3.6.0/java/util/src/main/java/com/google/protobuf/util/FieldMaskTree.java
protobuf-3.6.0/java/util/src/main/java/com/google/protobuf/util/Timestamps.java
protobuf-3.6.0/java/README.md
protobuf-3.6.0/BUILD
protobuf-3.6.0/compile
pi@raspberrypi:~/tf $ cd protobuf-3.6.0
pi@raspberrypi:~/tf/protobuf-3.6.0 $ ./configure
```

```
File Edit Tabs Help
config.status: creating build-aux/config.h
config.status: executing depfiles commands
config.status: executing libtool commands
pi@raspberrypi:~/tf/protobuf-3.6.0 $ make
make all-recursive
make[1]: Entering directory '/home/pi/tf/protobuf-3.6.0'
Making all in .
make[2]: Entering directory '/home/pi/tf/protobuf-3.6.0'
make[2]: Leaving directory '/home/pi/tf/protobuf-3.6.0'
Making all in src
make[2]: Entering directory '/home/pi/tf/protobuf-3.6.0/src'
depbases=`echo google/protobuf/stubs/bytestream.lo | sed 's|^[^/]*$|.deps/&;s|\.lo$||'`;\
/bin/bash ../libtool --tag=CXX --mode=compile g++ -DHAVE_CONFIG_H -I. -I..
-pthread -DHAVE_PTHREAD=1 -DHAVE_ZLIB=1 -Wall -Wno-sign-compare -O2 -g -std=c++11 -DNDEBUG -MT google/protobuf/stubs/bytestream.lo -MD -MP -MF $depbases.Tpo -c
-o google/protobuf/stubs/bytestream.lo google/protobuf/stubs/bytestream.cc &&\
mv -f $depbases.Tpo $depbases.Plo
```

```
pi@raspberrypi: ~/tf/protobuf-3.6.0
File Edit Tabs Help
# TOTAL: 7
# PASS: 7
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
make[3]: Leaving directory '/home/pi/tf/protobuf-3.6.0/src'
make[2]: Leaving directory '/home/pi/tf/protobuf-3.6.0/src'
make[1]: Leaving directory '/home/pi/tf/protobuf-3.6.0/src'
pi@raspberrypi:~/tf/protobuf-3.6.0 $
pi@raspberrypi:~/tf/protobuf-3.6.0 $ sudo make install
Making install in .
make[1]: Entering directory '/home/pi/tf/protobuf-3.6.0'
make[2]: Entering directory '/home/pi/tf/protobuf-3.6.0'
make[2]: Nothing to be done for 'install-exec-am'.
/bin/mkdir -p '/usr/local/lib/pkgconfig'
/usr/bin/install -c -m 644 protobuf.pc protobuf-lite.pc '/usr/local/lib/pkgconf
ig'
make[2]: Leaving directory '/home/pi/tf/protobuf-3.6.0'
make[1]: Leaving directory '/home/pi/tf/protobuf-3.6.0'
Making install in src
```

Sudo make install işlemi bittikten sonra TensorFlow dizin yapısını ve PYTHONPATH değişkenini ayarlanır.

```
/bin/mkdir -p '/usr/local/include/google/protobuf/compiler/objectivec'
/usr/bin/install -c -m 644 google/protobuf/compiler/objectivec/objectivec_gene
rator.h google/protobuf/compiler/objectivec/objectivec_helpers.h '/usr/local/inc
lude/google/protobuf/compiler/objectivec'
/bin/mkdir -p '/usr/local/include/google/protobuf/compiler'
/usr/bin/install -c -m 644 google/protobuf/compiler/code_generator.h google/pr
otobuf/compiler/command_line_interface.h google/protobuf/compiler/importer.h goo
gle/protobuf/compiler/parser.h google/protobuf/compiler/plugin.h google/protobu
f/compiler/plugin.pb.h '/usr/local/include/google/protobuf/compiler'
make[2]: Leaving directory '/home/pi/tf/protobuf-3.6.0/src'
make[1]: Leaving directory '/home/pi/tf/protobuf-3.6.0/src'
pi@raspberrypi:~/tf/protobuf-3.6.0 $ cd python
pi@raspberrypi:~/tf/protobuf-3.6.0/python $ export LD_LIBRARY_PATH=./src/.libs
pi@raspberrypi:~/tf/protobuf-3.6.0/python $ python3 setup.py build --cpp_impleme
ntation
-warn-string-overflow -Wno-invalid-offsetof -Wno-sign-compare -DPYTHON_PROTO2_CPP_IMPL
V2
cc1plus: warning: command line option '-Wstrict-prototypes' is valid for C/ObjC
but not for C++
arm-linux-gnueabihf-g++ -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions -Wl,-z
,relro -Wl,-z,relro -g -fdebug-prefix-map=/build/python3.5-RUBMX3/python3.5-3.5.
3=. -fstack-protector-strong -Wformat -Werror=format-security -Wdate-time -D_FOR
TIFY_SOURCE=2 build/temp.linux-armv7l-3.5/google/protobuf/internal/api_implemen
tation.o -o build/lib.linux-armv7l-3.5/google/protobuf/internal/_api_implemen
tation.cpython-35m-arm-linux-gnueabihf.so
pi@raspberrypi:~/tf/protobuf-3.6.0/python $
pi@raspberrypi:~/tf/protobuf-3.6.0/python $ python3 setup.py test --cpp_implemen
tation
```



```
pi@raspberrypi: ~/tf/protobuf-3.6.0/python
File Edit Tabs Help

--cpp_out=OUT_DIR      is mapped to the given executable even if
--csharp_out=OUT_DIR   the executable's own name differs.
--java_out=OUT_DIR      Generate C++ header and source.
--js_out=OUT_DIR        Generate C# source file.
--objc_out=OUT_DIR      Generate Java source file.
--php_out=OUT_DIR       Generate JavaScript source.
--python_out=OUT_DIR    Generate Objective C header and source.
--ruby_out=OUT_DIR      Generate PHP source file.
@<filename>            Generate Python source file.
                        Generate Ruby source file.
                        Read options and filenames from file. If a
                        relative file path is specified, the file
                        will be searched in the working directory.
                        The --proto_path option will not affect how
                        this argument file is searched. Content of
                        the file will be expanded in the position of
                        @<filename> as in the argument list. Note
                        that shell expansion is not applied to the
                        content of the file (i.e., you cannot use
                        quotes, wildcards, escapes, commands, etc.).
                        Each line corresponds to a single argument,
                        even if it contains spaces.
pi@raspberrypi:~/tf/protobuf-3.6.0/python $ sudo reboot now
```

4.5.TensorFlow Dizin Yapısını ve PYTHONPATH Değişkenini Ayarlama

TensorFlow dizinini kurarak Ana dizininize geri dönüp ardından “tensorflow1” adlı bir dizin oluşturulur.

```
pi@raspberrypi: ~/tensorflow1
File Edit Tabs Help
pi@raspberrypi:~ $ mkdir tensorflow1
pi@raspberrypi:~ $ cd tensorflow1/
pi@raspberrypi:~/tensorflow1 $ git clone --recurse-submodules https://github.com/tensorflow/models.git
Cloning into 'models'...
```

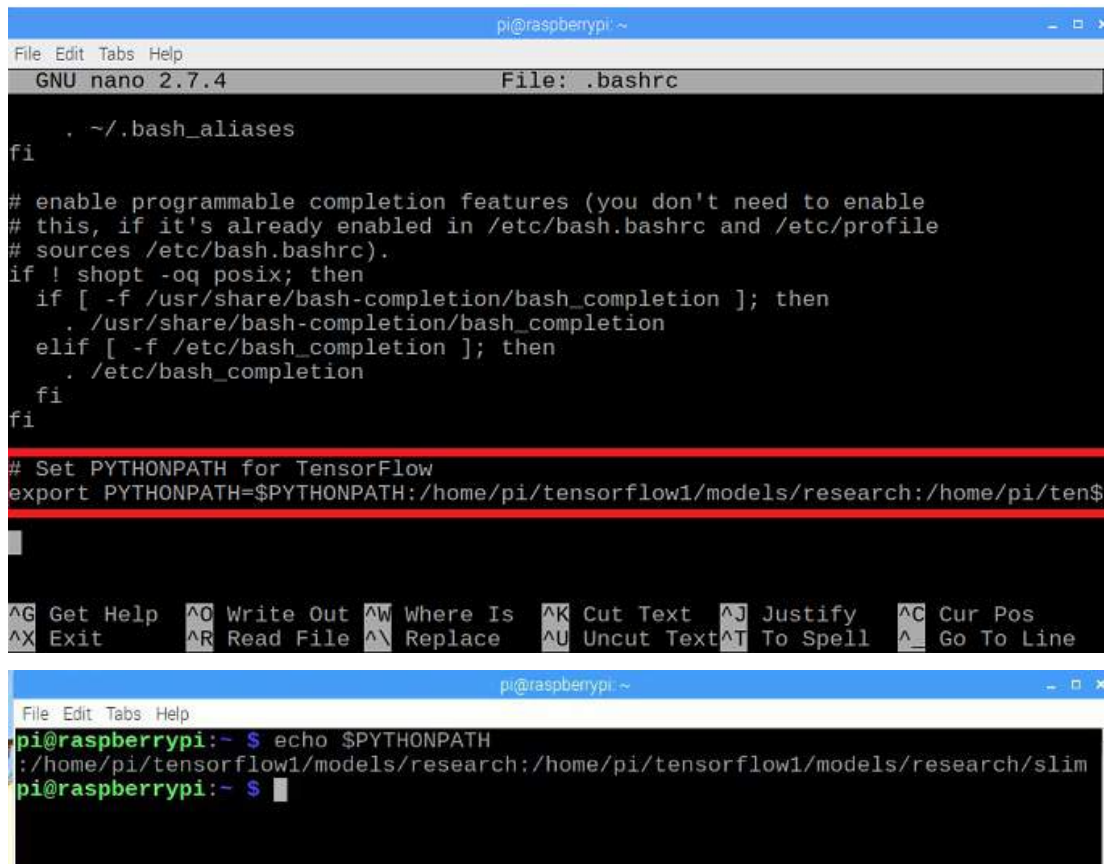
Oluşturulan dizine

git clone --depth 1 <https://github.com/tensorflow/models.git> komutu yazılır.

Burada yapılan işlem TensorFlow deposunu github dan indirmemiz sağlıyor. Daha sonra TensorFlow deposu içindeki bazı dizinleri gösterecek şekilde PYTHONPATH ortam değişkenini değiştirilmesi gerekir. Terminali açıldığında PYTHONPATH'ın açılması için; bashrc dosyasını değiştirilmesi gerekir. Bunun için sudo nano ~/.bashrc ile dosyanın sonuna aşağıdakiler yazılır.

export

PYTHONPATH=\$PYTHONPATH:/home/pi/tensorflow1/models/research:/home/pi/tensorflow1/models/research/slim



```
pi@raspberrypi ~  
File Edit Tabs Help  
GNU nano 2.7.4 File: .bashrc  
  
. ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
if ! shopt -oq posix; then  
  if [ -f /usr/share/bash-completion/bash_completion ]; then  
    . /usr/share/bash-completion/bash_completion  
  elif [ -f /etc/bash_completion ]; then  
    . /etc/bash_completion  
  fi  
fi  
  
# Set PYTHONPATH for TensorFlow  
export PYTHONPATH=$PYTHONPATH:/home/pi/tensorflow1/models/research:/home/pi/ten$  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line  
  
pi@raspberrypi ~  
File Edit Tabs Help  
pi@raspberrypi:~$ echo $PYTHONPATH  
:/home/pi/tensorflow1/models/research:/home/pi/tensorflow1/models/research/slim  
pi@raspberrypi:~$
```

Object Detection API'sı tarafından kullanılan Protokol Buffer (.proto) dosyalarını derlemek için Protoc kullanılması gereklidir. Proto dosyaları / Research / object_detection / protos konumunda bulunur, ancak komutu / Research dizininden yürütülmesi gerekir.

```
cd /home/pi/tensorflow1/models/research  
protoc object_detection/protos/*.proto --python_out=.
```

Bu komut, tüm "name" .proto dosyalarını "name_pb2" .py dosyalarına dönüştürür.

Ardından, object_detection dizinine,
cd /home/pi/tensorflow1/models/research/object_detection

SSD_Lite modelini TensorFlow deposundan indirilir. Model deposu, Google'ın çeşitli hız ve doğruluk düzeylerine sahip önceden eğitilmiş nesne algılama modelleri koleksiyonudur. Raspberry Pi'nin zayıf bir işlemcisi var, bu yüzden daha az işlem gücü gerektiren bir model kullanılması gerekiyor. Model daha hızlı çalışmasına rağmen, daha düşük hassasiyete sahip olma dengesinde geliyor. Bu eğitimde, mevcut en hızlı model olan SSDLite-MobileNet'i kullanılacaktır.

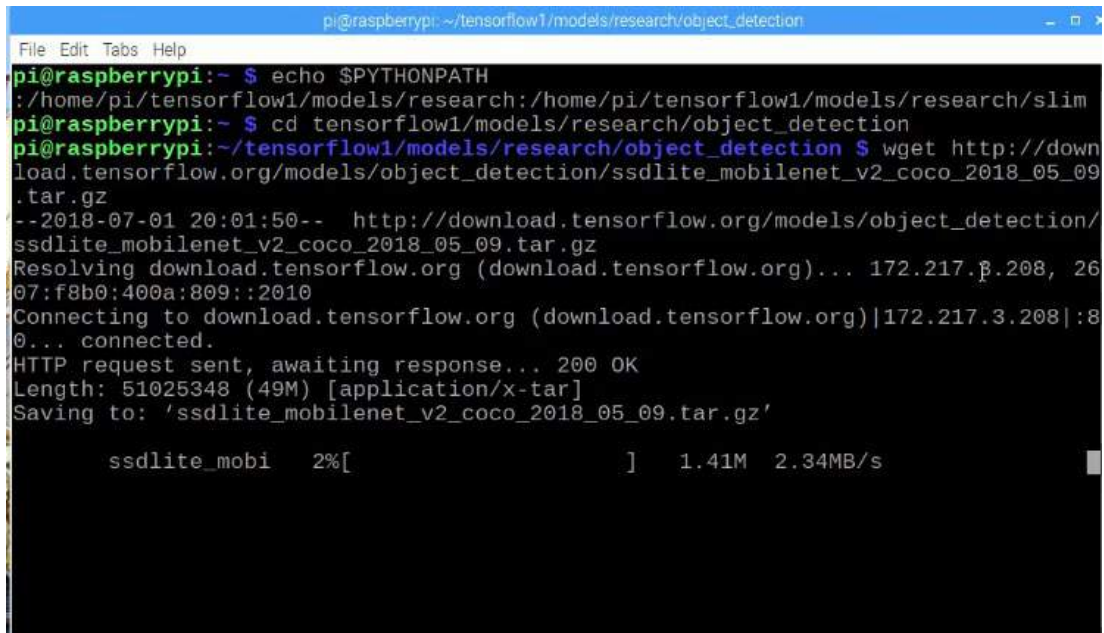
Google, geliştirilmiş hız ve performansa sahip modelleri sürekli olarak yayınlamaktadır, bu nedenle daha iyi model olup olmadığını görmek için model deposunu güncel tutulmalıdır.

SSDLite-MobileNet modelini indirilip düzenlenmesi;

wget

http://download.tensorflow.org/models/object_detection/ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz

tar -xzf ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz



```
pi@raspberrypi:~/tensorflow1/models/research/object_detection
File Edit Tabs Help
pi@raspberrypi:~ $ echo $PYTHONPATH
:/home/pi/tensorflow1/models/research:/home/pi/tensorflow1/models/research/slim
pi@raspberrypi:~ $ cd tensorflow1/models/research/object_detection
pi@raspberrypi:~/tensorflow1/models/research/object_detection $ wget http://download.tensorflow.org/models/object_detection/ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz
--2018-07-01 20:01:50-- http://download.tensorflow.org/models/object_detection/ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz
Resolving download.tensorflow.org (download.tensorflow.org)... 172.217.3.208, 2607:f8b0:400a:809::2010
Connecting to download.tensorflow.org (download.tensorflow.org)|172.217.3.208|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 51025348 (49M) [application/x-tar]
Saving to: 'ssdlite_mobilenet_v2_coco_2018_05_09.tar.gz'

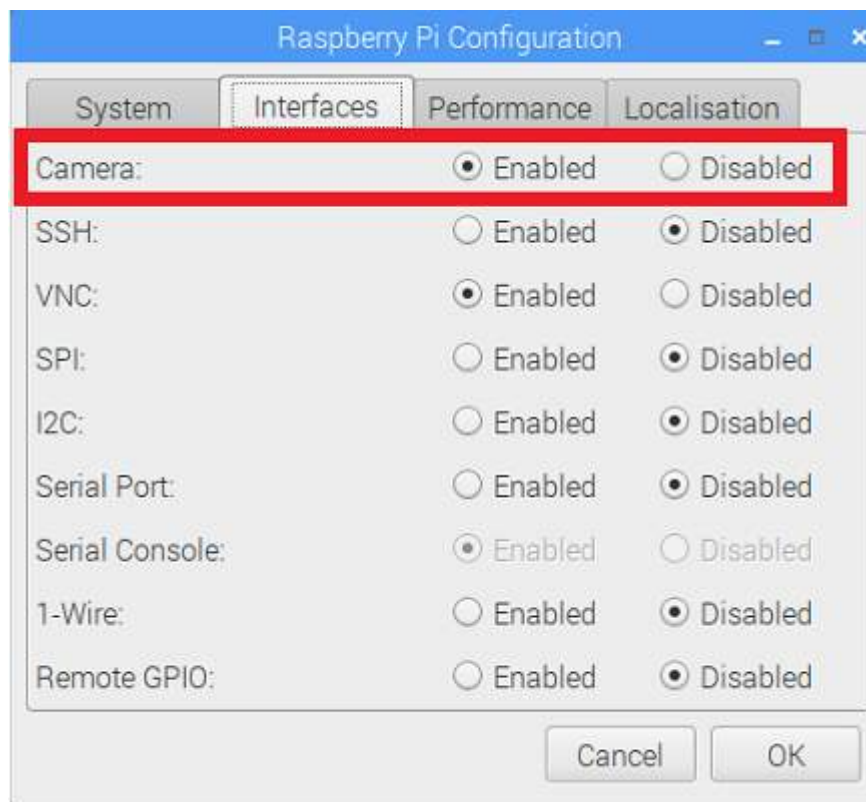
ssdlite_mobi  2%[          ] 1.41M 2.34MB/s
```

```
pi@raspberrypi: ~/tensorflow1/models/research/object_detection/protos
File Edit Tabs Help
pi@raspberrypi:~ $ cd tensorflow1/models/research/object_detection/protos
pi@raspberrypi:~/tensorflow1/models/research/object_detection/protos $ ls
anchor_generator.proto      matcher.proto
argmax_matcher.proto       mean_stddev_box_coder.proto
bipartite_matcher.proto    model.proto
box_coder.proto            multiscale_anchor_generator.proto
box_predictor.proto        optimizer.proto
eval.proto                 pipeline.proto
faster_rcnn_box_coder.proto post_processing.proto
faster_rcnn.proto          preprocessor.proto
graph_rewriter.proto       __pycache__
grid_anchor_generator.proto region_similarity_calculator.proto
hyperparams.proto          square_box_coder.proto
image_resizer.proto        ssd_anchor_generator.proto
__init__.py                ssd.proto
input_reader.proto         string_int_label_map.proto
keypoint_box_coder.proto   train.proto
losses.proto
pi@raspberrypi:~/tensorflow1/models/research/object_detection/protos $ cd ~
pi@raspberrypi:~ $ cd tensorflow1/models/research
pi@raspberrypi:~/tensorflow1/models/research $ protoc object_detection/protos/*.
proto --python_out=.
pi@raspberrypi:~/tensorflow1/models/research $ cd object_detection/protos/
pi@raspberrypi:~/tensorflow1/models/research/object_detection/protos $
```

4.6. Nesnelerin Algılanması

Object_detection_picamera.py içindeki Python betiği, bir Picamera veya USB web kamerasından canlı yayınlardaki nesneleri algılar. Temel olarak, komut dosyası model ve etiket eşlemesine giden yolları belirler, modeli belleğe yükler, Picamera'yı başlatır ve ardından Picamera'dan her video karesinde nesne algılama yapmaya başlar.

Picamera kullanıldığı için Raspberry Pi yapılandırma menüsünde camera seçeneğini aktifleştirilmesi gerekir.



Pi kamerayı aktif hale getirdikten sonra `Object_detection_picamera.py` dosyasını `object_detection` dizinine indirilir.

wget

https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi/master/Object_detection_picamera.py

```

pi@raspberrypi: ~/tensorflow1/models/research/object_detection
File Edit Tabs Help
pi@raspberrypi:~$ cd tensorflow1/models/research/object_detection/
pi@raspberrypi:~/tensorflow1/models/research/object_detection$ wget https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi/master/Object_detection_picamera.py
--2018-07-01 22:42:00-- https://raw.githubusercontent.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi/master/Object_detection_picamera.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.52.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.52.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8180 (8.0K) [text/plain]
Saving to: 'Object_detection_picamera.py'

Object_detection_pi 100%[=====] 7.99K --.-KB/s in 0.001s

2018-07-01 22:42:01 (10.5 MB/s) - 'Object_detection_picamera.py' saved [8180/8180]

pi@raspberrypi:~/tensorflow1/models/research/object_detection$

```

Komut dosyasına `python3 Object_detection_picamera.py` komutu yazılarak çalıştırılır.

