

SCUTTLE Denge Robotu - Yazılım Kurulum Kılavuzu

1. Adım : Raspberry Pi'nin SD kartını Bilgisayarımıza takıyoruz

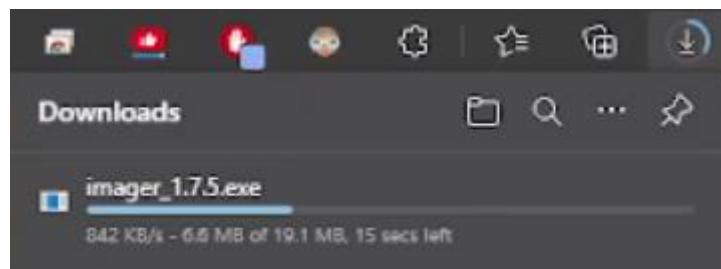
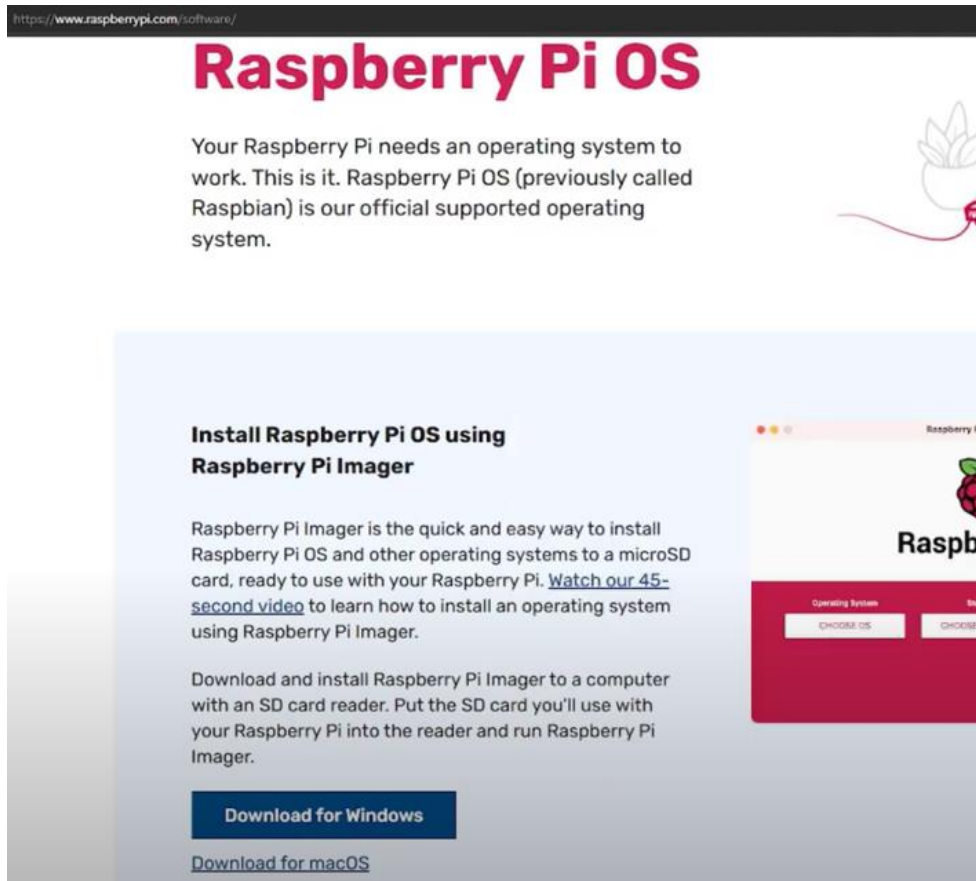
Minimum 4 GB bir SD kart yeterli olacaktır.

2. Adım : Raspberry Pi'nin web sitesinden Raspbian işletim sisteminin Lite versiyonunu indirip Raspberry Pi'nin SD kartına kurulumunu yapıyoruz, gerekli seçenekleri ve bilgileri giriyoruz.

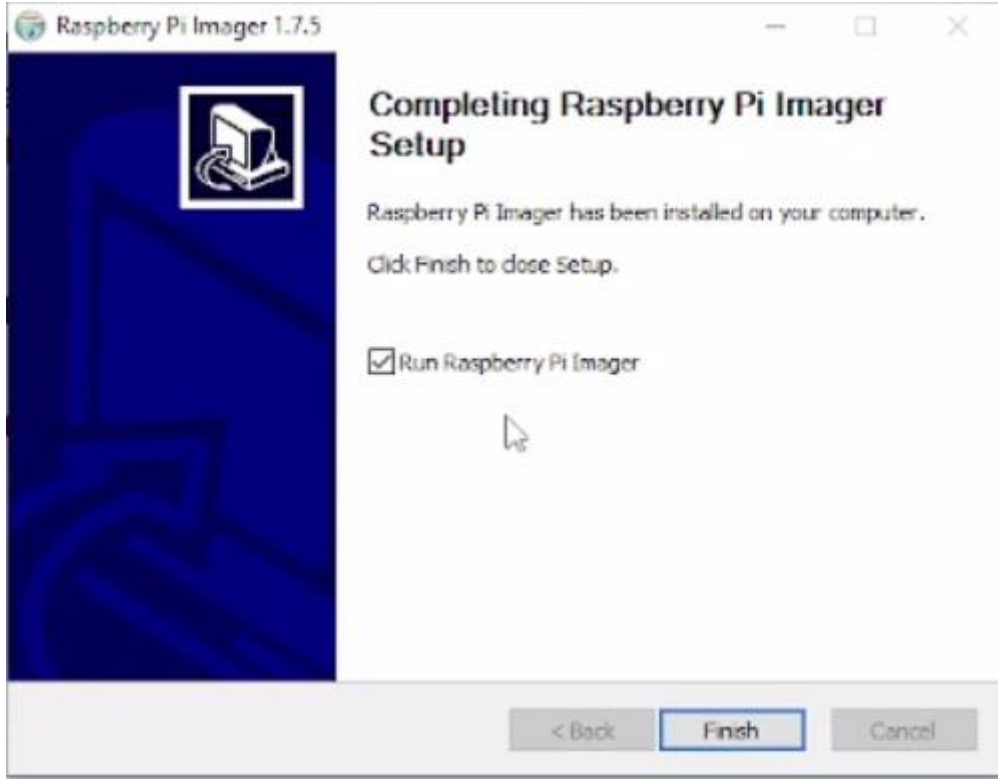
a. İlk olarak Raspberry Pi'nin web sitesini açıyoruz ve "software"i seçiyoruz.

<https://www.raspberrypi.com/software/>

b. Bu bölümden Raspberry Pi Imager programını indiriyoruz.



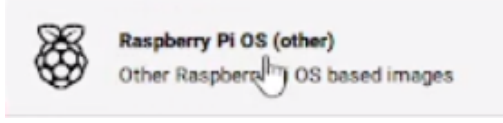
- c. Daha sonra bu "Imager_x.x.x.exe" dosyasını çalıştırıyoruz ve kurulumu başlatıyoruz.



- d. Kurulum tamamlandıktan sonra "Raspberry Pi Imager X.X.X" dosyasını çalıştırıyoruz. "Run Raspberry Pi Imager" seçili iken "Finish" butonuna basmak programı başlatmak için yeterlidir.



- e. İlk olarak “Choose OS” butonuna basarak işletim sistemini seçeceğimiz pencereyi açıyoruz. Burada önce “Raspberry Pi OS (other)” seçeneğini seçiyoruz.



- f. Daha sonra “Raspberry Pi OS Lite (32-bit)” işletim sistemini seçiyoruz. Siz dilerseniz “Raspberry Pi OS (32-bit)” işletim sistemini de seçebilirsiniz.



- g. Ardından işletim sisteminin nereye kurulacağını seçmek için “Choose Storage” butonuna basıyoruz.



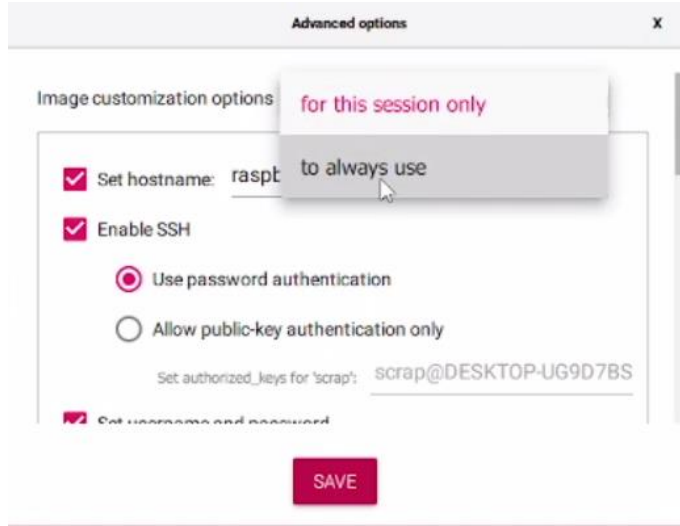
- h. SD kartımızı seçiyoruz.



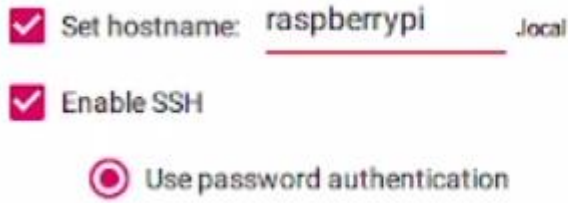
- i. Diğer işleme geçiyoruz ve “Ayarlar” butonuna basıyoruz.



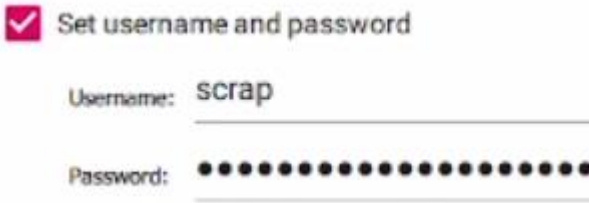
- j. Açılan pencerede ilk olarak “image customization options” için “to always use” seçeneğini seçiyoruz. Böylece bu yaptığımız ayarları her seferinde yeniden yapmamıza gerek kalmıyor ve bu değişiklikler kalıcı şekilde sisteme kaydediliyor.



k. Diğer ayarlar ise şöyle;



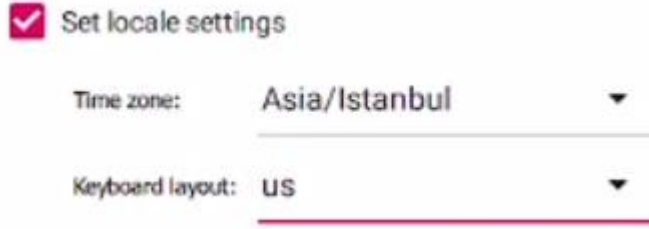
l. Bu bölümde bir kullanıcı adı ve şifre belirliyorsunuz.



m. Bu bölümde ise kablosuz bağlantı için kullanacağınız Wi-Fi yerel ağı belirliyoruz ve şifresini giriyoruz.



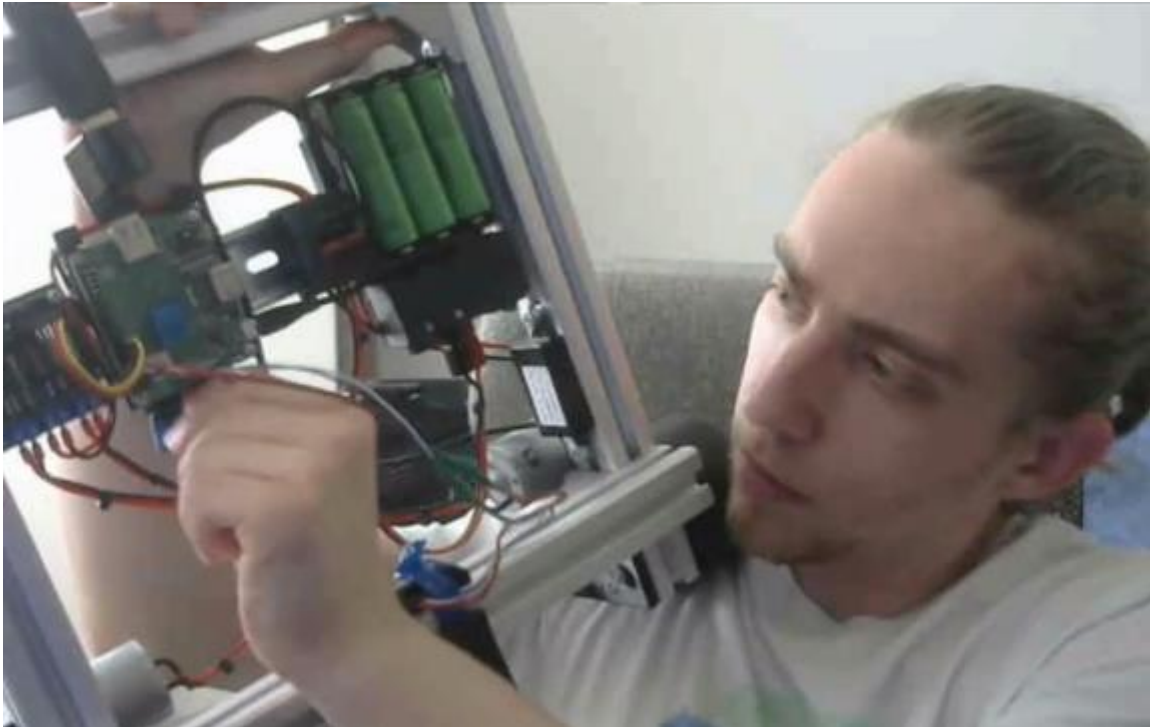
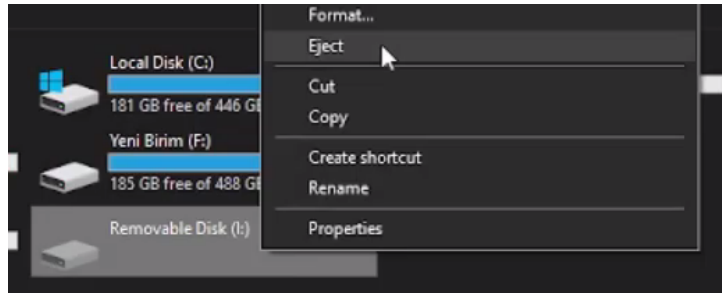
n. Bu bölümde zaman bölgenizi ve klavyenizi seçiyorsunuz.



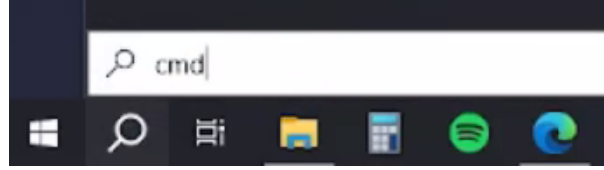
o. Son olarak "write" butonuna basarak bu değişiklikleri kaydediyoruz. İşlem sonuçlanana kadar bekliyoruz.



3. Adım : SD kartı Bilgisayardan çıkarıp Raspberry Pi'ye takıyoruz. Ve Raspberry Pi'yi çalıştırıyoruz.

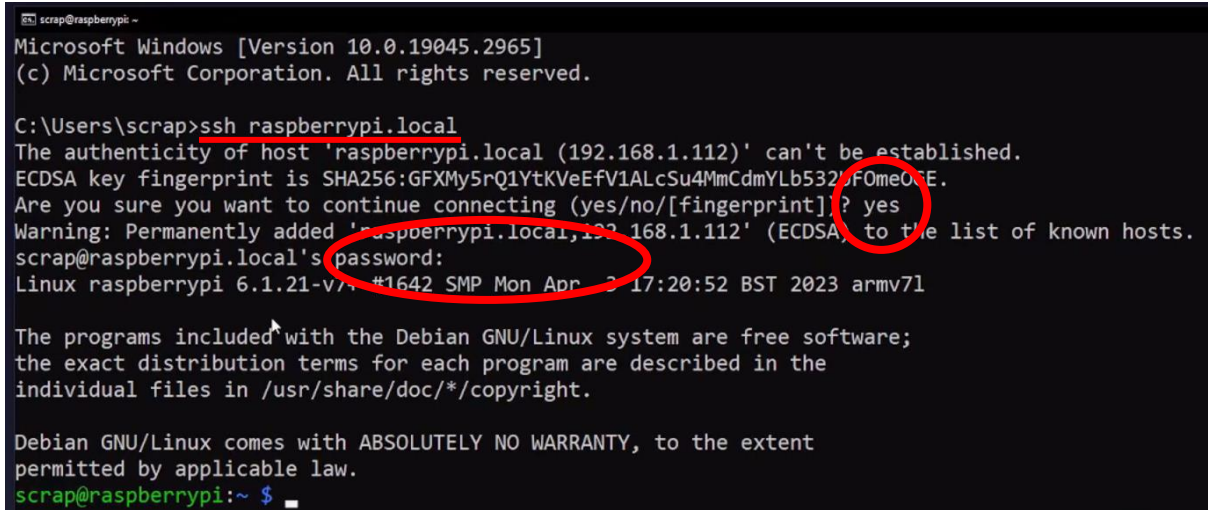


4. Adım : Raspberry Pi açılırken bu sırada bilgisayarımızda "cmd comment prompt"u çalıştırıyorum.



5. Adım : "cmd comment prompt" üzerinden Raspberry Pi'ye bilgisayarımızdan bağlanıyoruz.

Bağlanmak için bu komutu kullanıyoruz : "ssh raspberrypi.local"



Bağlantıyı gerçekleştirdik.

6. Adım : Raspberry üzerinde "Update" güncelleme ve "Upgrade" yükseltme gerektiren bütün yazılımların güncelleme ve yükseltmelerini yapıyorum.

Bunun için kullandığım komutlar şunlar :

"sudo apt-get update" ve "sudo apt-get upgrade"

```
scrap@raspberrypi:~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org/debian bullseye InRelease [23.6 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Get:3 http://archive.raspberrypi.org/debian bullseye/main armhf Packages [316 kB]
Get:4 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf Packages [13.2
Fetched 13.6 MB in 16s (824 kB/s)
Reading package lists... Done

scrap@raspberrypi:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  base-files libpam-systemd libsystemd0 libudev1 libwebp6 libwebpdemux2
  systemd systemd-sysv systemd-timesyncd udev
13 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 9,588 kB of archives.
After this operation, 1,053 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.raspberrypi.org/debian bullseye/main armhf raspber
Get:3 http://archive.raspberrypi.org/debian bullseye/main armhf rpi-eepr
Get:2 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf base-
Get:4 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf syste
Get:5 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libpa
Get:6 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libsy
Get:7 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf syste
Get:8 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf syste
Get:9 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf udev
Get:10 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libu
Get:11 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libw
Get:12 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libw
Get:13 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf libw
Fetched 9,588 kB in 7s (1,470 kB/s)
Reading changelogs... Done

scrap@raspberrypi:~ $
```

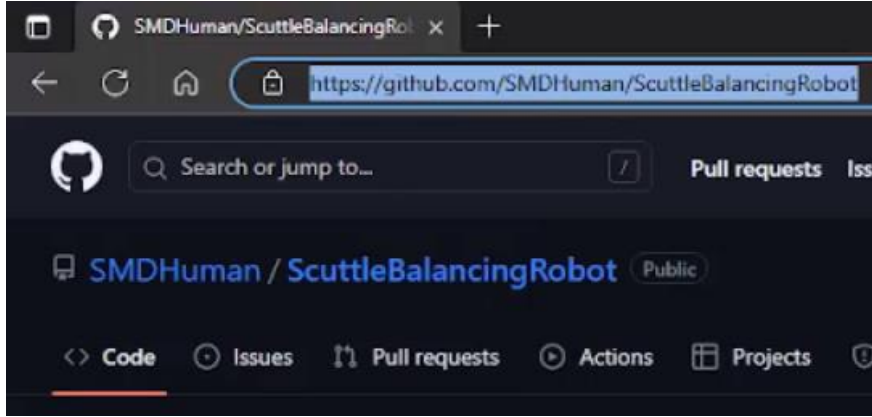
7. Adım : GitHub dan dosyaların indirilebilmesini sağlayan “Git” programını Raspberry Pi’ye yükliyorum.

Bunun için kullandığım komut şu : “sudo apt install git”

```
scrap@raspberrypi:~ $ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 6,564 kB of archives.
After this operation, 33.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
0% [Waiting for headers]
```

8. Adım : Projenin GitHub sayfasının Linkini kopyalayıp GIT programı ile bu projenin Python yazılımlarını dahil bütün dosyalarını Raspberry Pi'ye yüklüyorum.

Bunun için kullandığım komut şu : “git clone <github projesi web adresi linki>”



```
scrap@raspberrypi:~ $ git clone https://github.com/SMDHuman/ScuttleBalancingRobot
```

yükledikten sonra dosyaları görebilirsiniz.

```
scrap@raspberrypi:~ $ cd ScuttleBalancingRobot/  
scrap@raspberrypi:~/ScuttleBalancingRobot $ ls  
docs images README.md requirements.txt source  
scrap@raspberrypi:~/ScuttleBalancingRobot $ cd source/  
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ ls  
EasySMXController.py mainBalance.py MPU6050Manager.py
```

9. Adım : Python için gerekli kütüphaneleri yükleyen ve kontrol etmemizi sağlayan PIP programını Raspberry Pi'ye yüklüyorum.

Bunun için kullandığım komut şu : “sudo apt install pip”

```
scrap@raspberrypi:~ $ sudo apt install pip  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Note, selecting 'python3-pip' instead of 'pip'  
The following additional packages will be installed:  
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore  
  python-pip-whl python3-dev python3-distutils python3-lib2to3 python3-pip  
Suggested packages:  
  apache2 | lighttpd | httpd python-setuptools-doc  
The following NEW packages will be installed:  
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore  
  python-pip-whl python3-dev python3-distutils python3-lib2to3 python3-pip  
  python3.9-dev  
0 upgraded, 15 newly installed, 0 to remove and 0 not upgraded.  
Need to get 7,224 kB of archives.  
After this operation, 23.0 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
0% [Working]
```


10. Adım : “Requirement.txt” dosyasında belirttiğim gerekli Python kütüphanelerini PIP komutu ile Raspberry Pi’ye indiriyorum.

Bunun için kullandığım komut şu : “pip install -r requirement.txt”

```
scrap@raspberrypi:~/ScuttleBalancingRobot $ ls
docs images README.md requirements.txt source
scrap@raspberrypi:~/ScuttleBalancingRobot $ cat requirements.txt
RPi.GPIO==0.7.1
evdev==1.6.1
mpu6050-raspberrypi==1.2
smbus==1.1.post2
```

```
scrap@raspberrypi:~/ScuttleBalancingRobot $ pip install -r requirements.txt
asd
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting RPi.GPIO==0.7.1
  Downloading https://www.piwheels.org/simple/rpi-gpio/RPi.GPIO-0.7.1-cp39-cp39-linux_armv7l.whl (66 kB)
    |-----| 66 kB 293 kB/s
Collecting evdev==1.6.1
  Downloading https://www.piwheels.org/simple/evdev/evdev-1.6.1-cp39-cp39-linux_armv7l.whl (80 kB)
    |-----| 80 kB 425 kB/s
Collecting mpu6050-raspberrypi==1.2
  Downloading https://www.piwheels.org/simple/mpu6050-raspberrypi/mpu6050_raspberrypi-1.2-py3-none-any.whl (6.5 kB)
Collecting smbus==1.1.post2
  Downloading https://www.piwheels.org/simple/smbus/smbus-1.1.post2-cp39-cp39-linux_armv7l.whl (39 kB)
Installing collected packages: smbus, RPi.GPIO, mpu6050-raspberrypi, evdev
Successfully installed RPi.GPIO-0.7.1 evdev-1.6.1 mpu6050-raspberrypi-1.2 smbus-1.1.post2

scrap@raspberrypi:~/ScuttleBalancingRobot $
scrap@raspberrypi:~/ScuttleBalancingRobot $
scrap@raspberrypi:~/ScuttleBalancingRobot $ asd
-bash: asd: command not found
scrap@raspberrypi:~/ScuttleBalancingRobot $
```

11. Adım : Raspberry Pi’nin konfigrasyon arayüzüne girerek, MPU6050’ye bağlanmasını sağlayacak I2C pinlerini etkinleştiriyorum.

a. Bunun için kullandığım komut şu : “sudo raspi-config”

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ sudo raspi-config
```

b. Konfigrasyon ekranından “3 Interface Options”ı seçiyoruz.

```
Raspberry Pi 3 Model A Plus Rev 1.0

Raspberry Pi Software Configuration Tool (raspi-config)

1 System Options          Configure system settings
2 Display Options         Configure display settings
3 Interface Options       Configure connections to peripherals
4 Performance Options     Configure performance settings
5 Localisation Options   Configure language and regional settings
6 Advanced Options        Configure advanced settings
8 Update                  Update this tool to the latest version
9 About raspi-config      Information about this configuration tool
```

c. Burada "I5 I2C Enable/disable"ı seçiyoruz.

```
Raspberry Pi Software Configuration Tool (raspi-config)

I1 Legacy Camera Enable/disable legacy camera support
I2 SSH           Enable/disable remote command line access using SSH
I3 VNC          Enable/disable graphical remote access using RealVNC
I4 SPI          Enable/disable automatic loading of SPI kernel module
I5 I2C          Enable/disable automatic loading of I2C kernel module
I6 Serial Port  Enable/disable shell messages on the serial connection
I7 1-Wire       Enable/disable one-wire interface
I8 Remote GPIO  Enable/disable remote access to GPIO pins
```

d. "ARM I2C arayüzü"nü etkinleştirmek için "Yes" tuşuna basıyorum.

```
Would you like the ARM I2C interface to be enabled?

<Yes> <No>
```

e. "ARM I2C" etkinleştirildi.

```
The ARM I2C interface is enabled

<Ok>
```

12. Adım : "mainBalance.py" Python yazılım dosyasını çalıştırıyorum. Ve Denge Programı çalışmaya başlıyor. Bunun için kullandığım komut şu : "python mainBalance.py"

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ python mainBalance.py
```

13. Adım : Problem 1 : Kumanda ile iletişim kurmamızı sağlayan "Dongle" takılı olmadığında oluşacak problemin çözümünü burada anlatıyorum. Kumanda hangi isim ile Raspberry Pi tarafından tanınırsa bu ismin kod içinden güncellemesinin nasıl yapıldığını burada anlatıyorum.

a. İlk olarak "Dongle"ı robotumdan çıkarıyorum.

b. "nano mainBalance.py" komutunu vererek "mainBalance.py" dosyasını açıyoruz.

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ nano mainBalance.py
```

c. Burada Controller “event0” olarak görülmektedir.

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source
GNU nano 5.4 mainBalance.py
leftBackwardPWM = GPIO.PWM(leftBackwardPin, 5000)
rightForwardPWM = GPIO.PWM(rightForwardPin, 5000)
rightBackwardPWM = GPIO.PWM(rightBackwardPin, 5000)

speedBias = 0
biasAngle = 0
targetAngle = -15.6
tagretSpeed = 0

moveDir = [0, 0]

pad = Controller('/dev/input/event0')
imu = IMUManager()
```

d. “ls /dev/input” komutu ile sadece “event0” görülebiliyor.

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ ls /dev/input/
by-path event0 mice
```

e. “Dongle” tekrar Raspberry Pi’ye takıyorum.

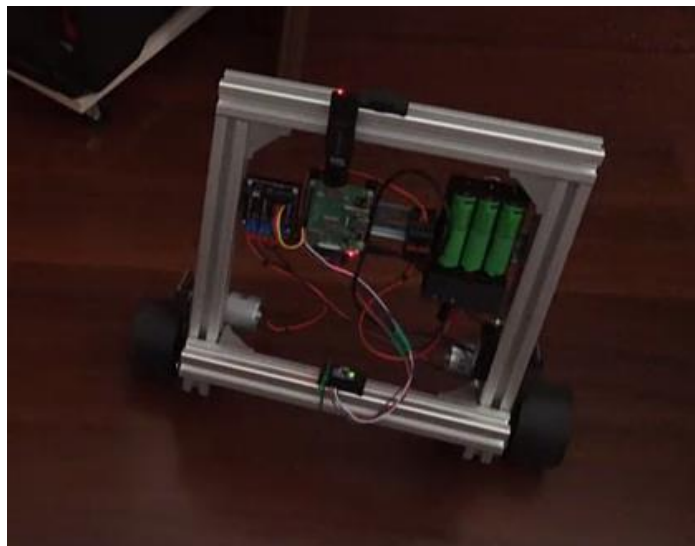
f. “ls /dev/input” komutu ile artık baktığımızda “event1” olarak yeni kontrol kumandası sisteme eklendiğini görebiliriz.

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ ls /dev/input/
by-id by-path event0 event1 js0 mice
```

g. “mainBalance.py” dosyasında bu satır bu şekilde düzeltildiğinde Kontrol Kumandasının sistem tarafından görülmesi sağlanır. “pad = Controller(‘/dev/input/event1’)”

```
pad = Controller('/dev/input/event1')
```

14. Adım : Robotum sorunsuz çalışıyor.



15. Adım : Problem 2 : Kumanda markası ile ilgili sorun yaşanması durumunda hangi yazılım dosyasının düzenlenmesi gerektiğini burada anlatıyorum.

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ ls
EasySMXController.py  mainBalance.py  MPU6050Manager.py  __pycache__
```

“EasySMXController.py” dosyasından gerekli düzenlemeler yapılır ve Yeni farklı marka kumanda sisteme tanıtılabilir.

16. Adım : Kumandayı devre dışı bırakmak isterseniz kod içinde yapılması gereken değişiklikler bunlardır..

a. İlk olarak . “nano mainBalance.py” komutunu ile “mainBalance.py” dosyasını açıyoruz.

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source $ nano mainBalance.py
```

b.” from EasySMXController import Controller” satırının başına “#” koyuyorum, satırı devre dışı bırakmak için

```
scrap@raspberrypi:~/ScuttleBalancingRobot/source
GNU nano 5.4 mainBalance.py *
import RPi.GPIO as GPIO
from time import sleep
from MPU6050Manager import IMUManager
#from EasySMXController import Controller
```

c.”pad= Controller('/dev/input/event1')” satırının başına “#” koyuyorum, satırı devre dışı bırakmak için

```
#pad = Controller('/dev/input/event1')
```

d. “speedPID.target” ve “moveDir[1]” değişkenlerine 0 atıyorum

```
speedPID.target = 0
moveDir[1] = 0
```

17. Adım : Robotumuzun sabit bir hızda gitmesini yada sabit yerinde kalmasını, aşağıdaki şekilde kod içinde yapacağımız değişikliklerle sağlayabiliriz.

“speedPID.target” robotun hızını kontrol ettiğim bir değişken. Bu değişkene 0 değeri atarsam robot yerinde sabit durur ve hareket etmez.

```
speedPID.target = (pad.leftJoyY-128) * 40 / 128
```

```
speedPID.target = 10
```

18. Adım : Robotun ne kadar sağa yada ne kadar sola döneceğini, yada dönmeden sabit kalmasını kod içinde yapacağımız şu değişikliklerle sağlıyoruz

“speedPID.target” robotun sağı sola dönüşlerini kontrol ettiğim bir değişken. Bu değışkene 0 değeri atarsam robot yerinde sabit durur ve sağı sola dönmez.

```
moveDir[1] = (pad.leftJoyX - 128) * 50 / 128
```

```
moveDir[1] = 12
```

19. Adım : PID ince ayarlarını kod üzerinde aşğıdaki şekilde yapabiliriz.

PID değerlerine ince ayar gerektiğinde p,i,d ve sp,si,sd değışkenlerinin değerlerini değıştirerek en başarılı değerleri bulabilirsiniz.

```
scrap@raspberrypi: ~/ScuttleBalancingRobot/source
GNU nano 5.4 mainBalance.py
import RPi.GPIO as GPIO
from time import sleep
from MPU6050Manager import IMUManager
from EasySMXController import Controller

p, i, d = 16.5, 0.01, 30
sp, si, sd = 0.018, 0.0000004, 0.0002
```

20. Adım : Tebrikler. Artık SCUTTLE robotunuzun iki teker üzerinde dengede kalmasını sağlayabilirsiniz. Sizlerden gelen öneriler ile bu dokümanı güncel tutmak istiyorum. Lütfen gördüğünüz eksiklikleri ya da geliştirilmesi gereken yerleri reductalimenitis@gmail.com mail adresime gönderin, mutlaka üzerlerinde çalışacağım. İyi eğlenceler.