

T.C.

ATATÜRK ÜNİVERSİTESİ

Mühendislik Fakültesi

GoPiGo Robot

LİSANS DÖNEM PROJESİ

Bilgisayar Mühendisliği

Proje Öğrencileri

120757011-Şeymanur YOLCU

120757019-Beyza AKSOY

110757001-Halil İbrahim YILMAZ

110757025-Selçuk POLAT

Proje Danışmanı

Yrd.Doç.Dr. Levent BAYINDIR

Erzurum–2017

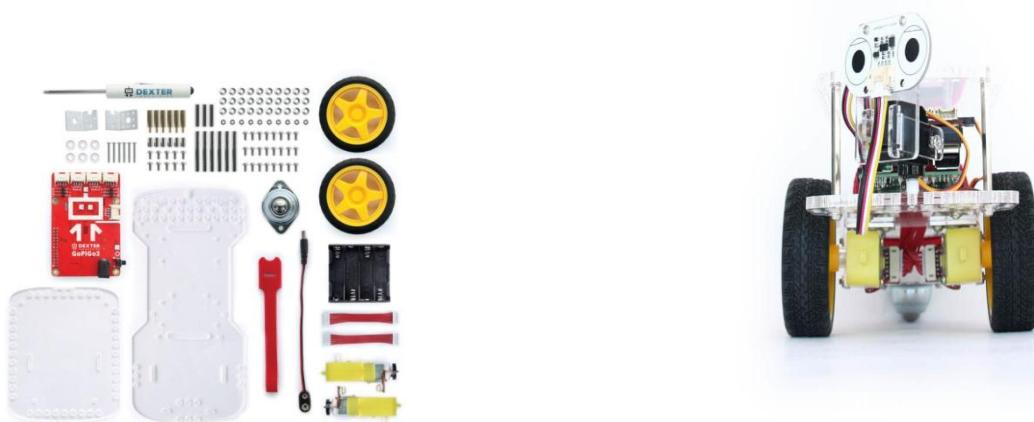
İçindekiler

| | |
|---|----|
| Gopigo nedir? | 4 |
| Raspberry nedir? | 4 |
| GoPiGo Robot Kitinin Kurulum Aşamaları | 5 |
| 1.GoPiGo Raspberry Pi Robot Kitini Birleştirme | 5 |
| Gerekli Malzemeler | 5 |
| 1.Adım : Kutu ambalajından çıkarıldı. | 5 |
| 2. Adım: Parçaların Birleştirilmesi | 6 |
| 3.Adım : Motorların Takılması | 7 |
| 4.Adım : Kodlayıcılar | 9 |
| 5. Adım : GoPiGo Kartının Takılması | 10 |
| 6. Adım : Caster Wheel in takılması | 10 |
| 7. Adım : Tekerleklerin Takılması | 11 |
| 8.Adım : Pil Kutusunun Takılması | 11 |
| 9. Adım : Gücün bağlanması ve testi | 12 |
| 10. ADIM : Raspberry Pi nin GoPiGo Robotuna Takılması | 14 |
| 11. Adım : Üst Parçanın Takılması | 14 |
| 12. Adım : Motorların Bağlanması | 14 |
| 2. Aşama SD Card | 15 |
| Windows için Sd Card kurulumu | 15 |
| İlk SD Kartınızı Çalıştırma | 18 |
| Seçenek 1: Dexter Industries'den bir SD Kartı | 18 |
| 2. Seçenek: Raspberry Pi 'de GoPiGo için image indirip kurun. | 19 |
| 3. seçenek: | 19 |
| 3. Aşama: Power UP(GoPiGo'yu çalıştırın) | 20 |
| 4.Aşama : GoPiGo ya bağlanma | 20 |
| Seçenek1: Masaüstü Olarak Kurulum | 21 |
| Seçenek 2: Bilgisayarınıza bir ethernet kablosuyla bağlama | 21 |
| Wifi Bağlanması | 23 |
| 5.Aşama : GoPiGo cihazını ilk defa başlatma | 27 |
| 6.Aşama : Opencv Kurulumu | 32 |
| Gopigo ile test ettiğimiz uygulamalar | 33 |
| Basic_Robot_Control | 33 |
| Basic_Robot_Control_GUI | 34 |
| Basic_Servo | 34 |
| Browser_Streaming_Robot | 35 |

| | |
|---|----|
| Mouse_Control..... | 37 |
| I2C Haberleşme Protokolü..... | 38 |
| Raspberry Pi ile Arduino I2C İletişim | 39 |
| Raspberry Pi ve Gopigo I2C Haberleşme Pinleri..... | 40 |
| GoPiGo I2C İletişiminin Donanımsal İncelenmesi..... | 41 |
| Yazılımsal Olarak I2C Haberleşme | 41 |
| RASPBERRY İLE ARDUINO HABERLEŞMESİ | 55 |
| Instal.sh | 55 |
| Instal_avrdude.sh..... | 65 |
| Setup.sh..... | 68 |
| GOPIGO İLE TOP TAKİBİ..... | 71 |
| Kaynakça..... | 78 |

Gopigo nedir?

GoPiGo, 400 seviyeli Bilgisayar Bilimi dersi için açık kaynaklı programlama ortamlarında pratik yapmaya yarayan bir robot kitidir.



Raspberry nedir?

Raspberry Pi kredi kartı boyutunda bir bilgisayardır. Televizyonunuza bağlayıp görüntü alabilir, bir klavye bağlayabilirsiniz. Yetenekli küçük bir bilgisayar diye adlandırdığımız Raspberry Pi ile normal masaüstü bilgisayarlarda yaptığınız işleri örneğin, sözcük işlemciler ve hesap programları (Word, Excel) ile çalışabilir çeşitli oyunlar oynayabilirsiniz. Ayrıca yüksek çözünürlüklü HD videolar oynatabilirsiniz. Ayrıca tüm dünyada çocukların alıp kullanabileceği, basit programlama yapabilecekleri hatta deneylerinde kullanabileceği uygun fiyatlı bir bilgisayar gibi düşünübilirsiniz.



GoPiGo Robot Kitinin Kurulum Aşamaları

1.GoPiGo Raspberry Pi Robot Kitini Birleştirme

2. SD Kartını Kurma .

4. GoPiGo'yu çalışma.

4. GoPiGo'ya bağlama .

5. GoPiGo cihazını ilk defa başlatma .

1.GoPiGo Raspberry Pi Robot Kitini Birleştirme

Gerekli Malzemeler

- Küçük Yıldız tornavida
- 8 AA pil

1.Adım : Kutu ambalajından çıkarıldı.

İçindeki Malzemeler

- 2 tekerlek
- GoPiGo Kartı
- 2 motor
- 2 büyük gövde parçası, 4 t şekilli parça, 1 uzun parça, 2 tekerlek
- 2 kutu donanım parçaları(4 uzun civata, 20 küçük vida, 8 küçük somun)
- Pil kutusu (8 AA pille uyumludur)



Resim 1

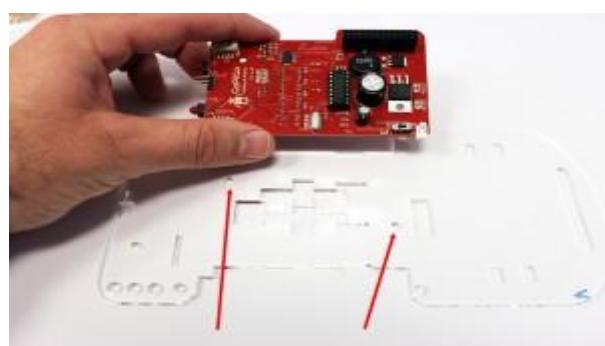
2. Adım: Parçaların Birleştirilmesi

Gopigo kartının montajı için 2 küçük somun,2 cıvataya ihtiyaç vardır.



Resim2

Bu iki delik GoPiGo Board'a uyumlu şekilde tasarlanmıştır.



Resim 3

Resim 4'de de görüldüğü gibi 2 küçük somun buraya monte edilir.



Resim 4

Son olarak kart alt tabaka ile birleştirip vidalanır.



Resim 5

3.Adım : Motorların Takılması

Motorları şasiye bağlamak için önce dört adet uzun civataları bulup, hazırlayacağımız motorları takmak için kitteki akrilik "T" ler kullanılmıştır. Motora vidalama işleminde dikkatli olun. Eğer aşırı sıkarsanız kırılabilir.



Resim 6

Her civataya 1 somun takarak civatalar hazırlandı.



Resim 7

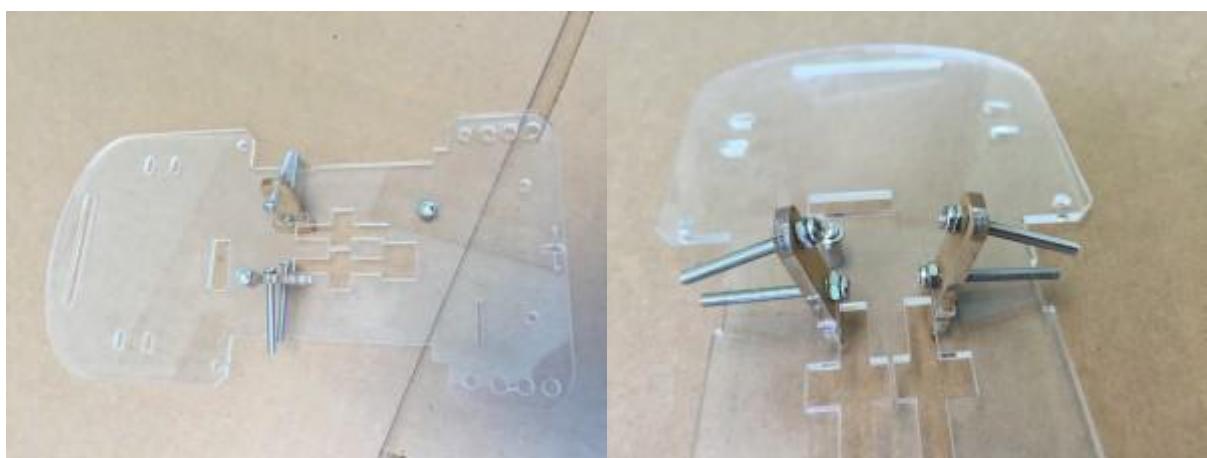
Vidalı kafaya kadar her civata üzerine tek bir somunu vidalayarak civataları hazırlayın.

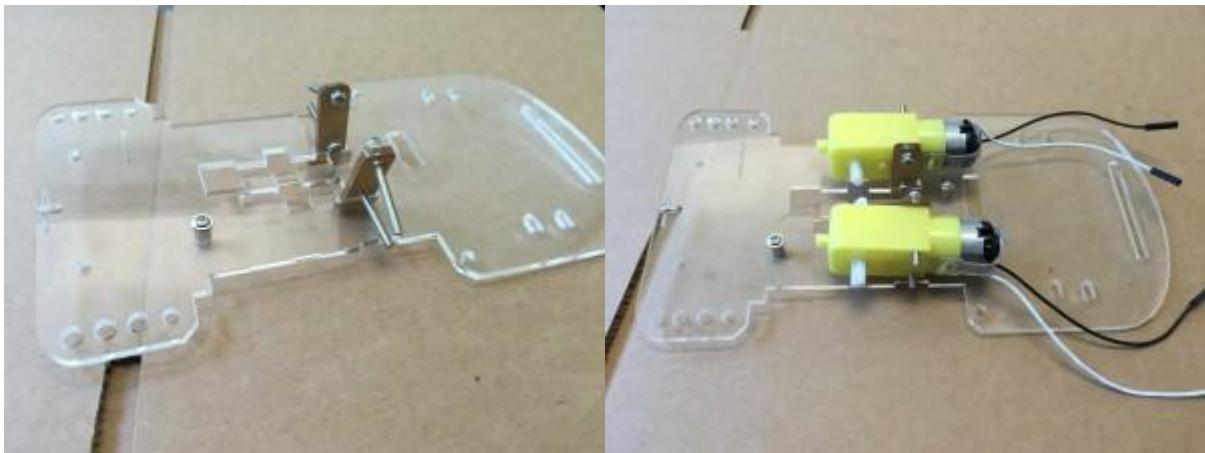
Her iki T'yi üstten takın ve akrilik gövdenin üzerine çevirin, böylece T'nin alt kısmı Resim 8'de olduğu gibi yapışır. Bu noktada sıkıştığınız mini metrik vida aşağı bakmalı ve gövdenin alt tarafında (motorları takacağınız yerlerde) çalışılmaktadır.



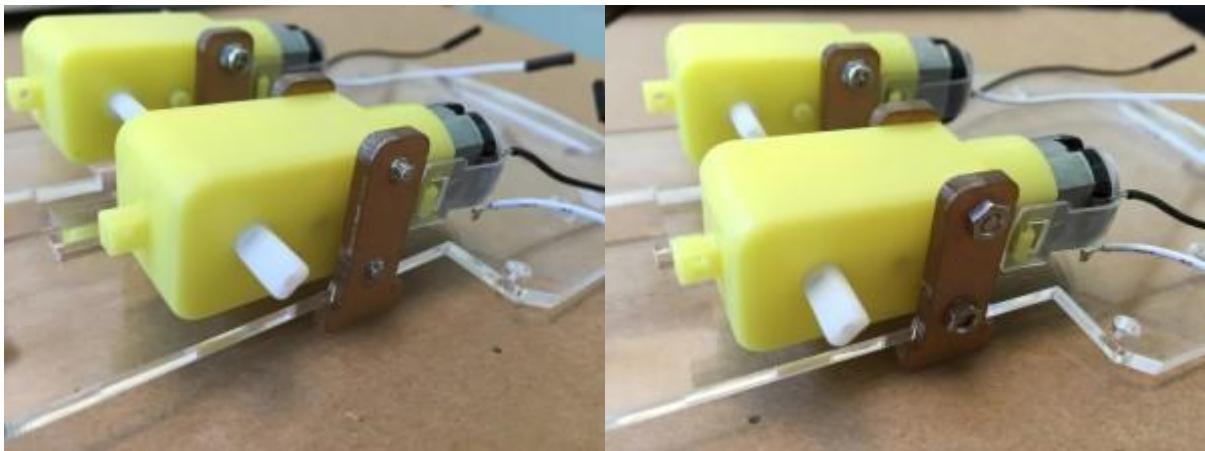
Resim 8

Daha sonra (somunları vidalayarak hazırladığınız) uzun civataları kullanın ve onları her T'nin her iki delidine doğru kaydırın. Aşağıdaki resimde gösterildiği gibi dışarı doğru yapışacak şekilde içten takılmıştır.





Motora yapışan cıvatalarla başka bir akrilik "T" yi kaydırarak parantez olarak hareket ettirin ve yerine tutun. T'yi yerinde tutmak için her cıvata somununu takın.



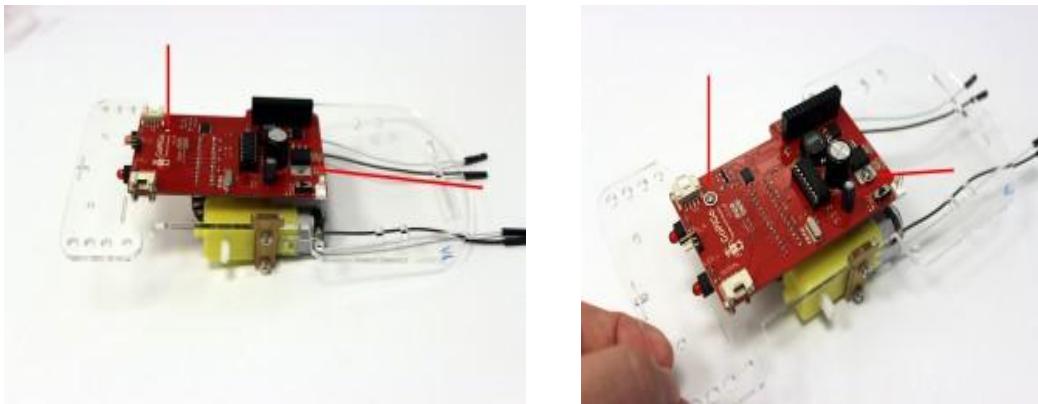
4.Adım : Kodlayıcılar

GoPiGo için kodlayıcılar siyah küçük bir tekerlek gibi görünüyor. Motorlar içinde uygun ve GoPiGo akrilik kasa ile uyum sağlamıştır. Bunlar motorun devir yönü hakkında geri bildirim sağlar. Motorlar bu kodlayıcılar olmadan çoğu zaman iyi çalışır, ama motorlar eylemi kontrol etmek ve iyileştirmek için kullanılabilir.



5. Adım : GoPiGo Kartının Takılması

1. adımda şasinin üstüne yerleştirdiğimiz mini metal direklere geri dönüyoruz. GoPiGo kartı ara parçalara yerleştirilip karttaki deliklerle hizalanmıştır. Kırmızı GoPiGo kartını eklemek için küçük vidalar kullanılmıştır.



6. Adım : Caster Wheel in takılması

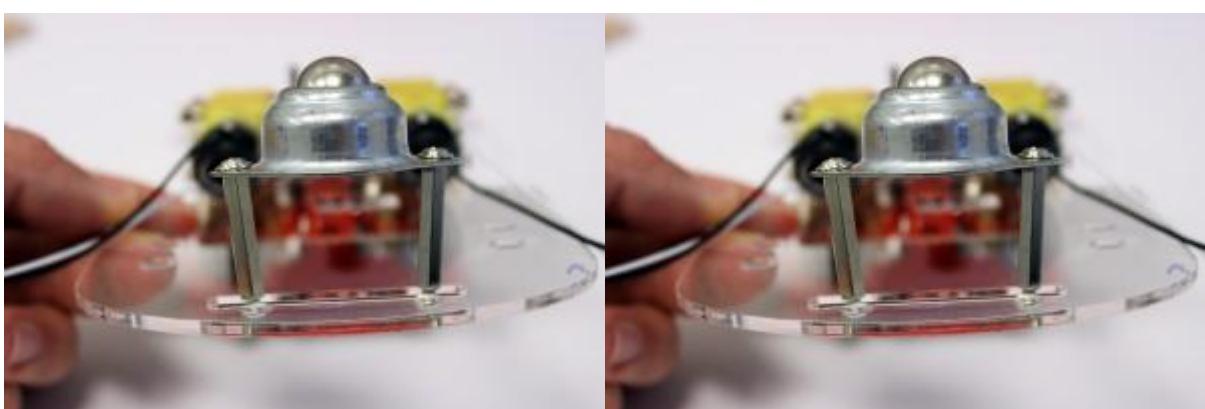


Resim 10

Kullanılan altigen destekler.

iki altigen destekler GoPiGo ‘nun ön kısmına takıldı.

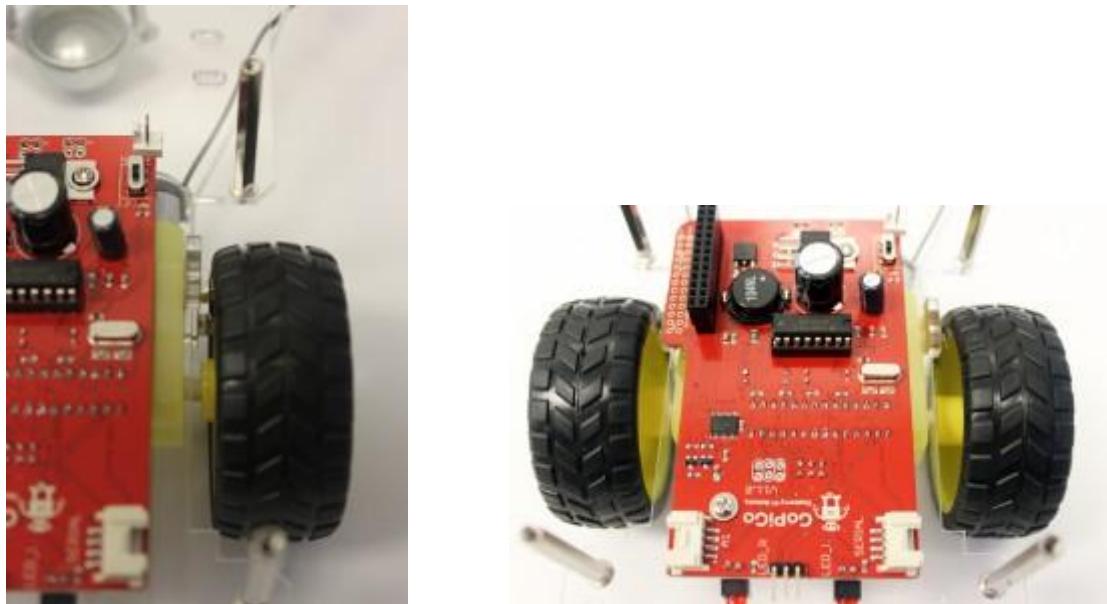
Diger ikisi ise arka kısmına takıldı.



Resim 11

7. Adım : Tekerleklerin Takılması

Boşluklar kontrol edilmiştir. Çünkü tekerlekler vidalara sürülmemelidir.



8.Adım : Pil Kutusunun Takılması

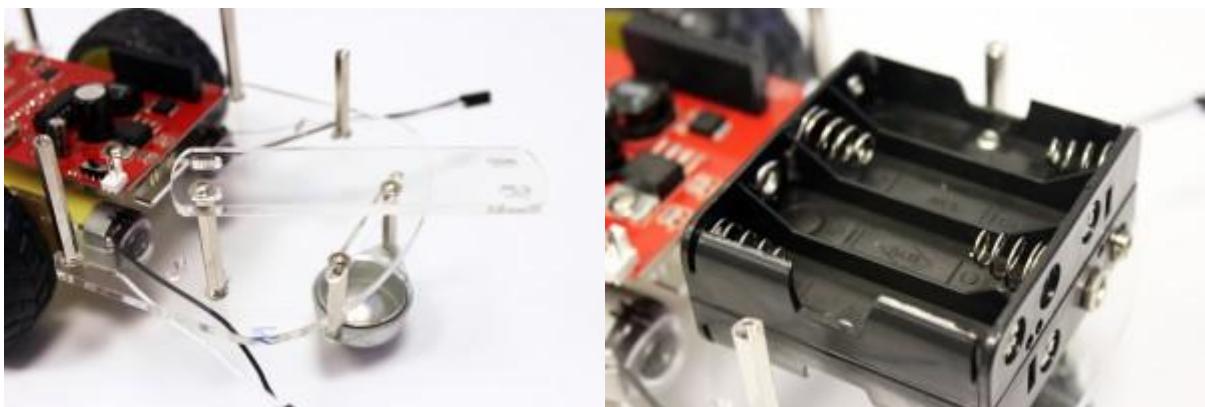
Akü kutusu için Adım 6'daki iki altigen ara parçası bulunmuştur. Bu parçalar GoPiGo tabakasına vidalanmıştır.

Önce, bu ara parçaları GoPiGo'ya bağlayacağız

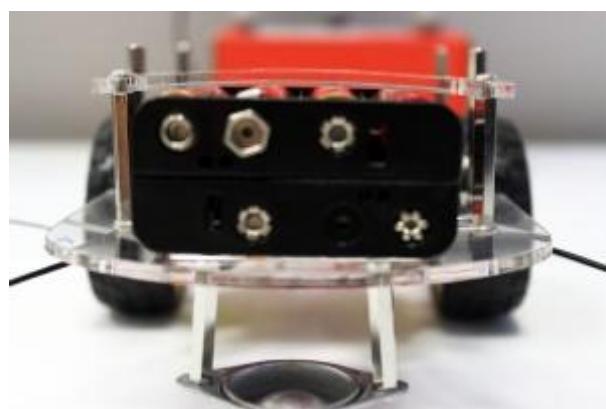


Spacers, GoPiGo'ya bağlandı.

Sonra, Pil Kutusu askısı takılmıştır. Bu, pil kutusu aralıklarına takılmış olan aşağıda gösterilen uzun bir akrilik parçadır. Önce vidayı gevşek bırakarak bir tarafı takılmıştır.



Pil kutusu kayışının diğer tarafını zıt taraftaki ara parçasına sıkıştırınız. Her iki vida da sıkıştırılsın. Kayış, pil kutusunun etrafına baskı yapmak ve yerine tutmak için tasarlanmıştır.

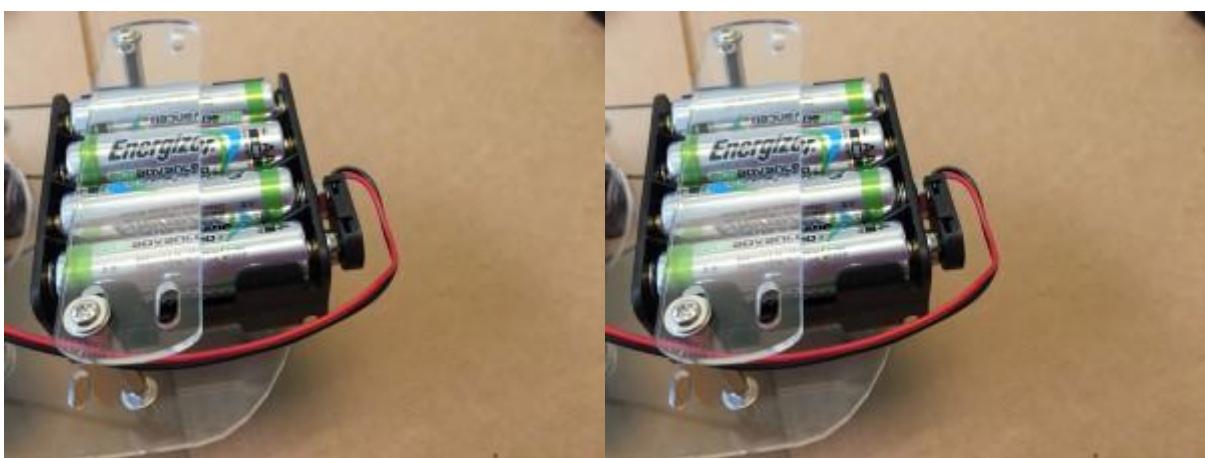


Resim 13

Pil kutusu sıkıca şaseye sabitlendi.

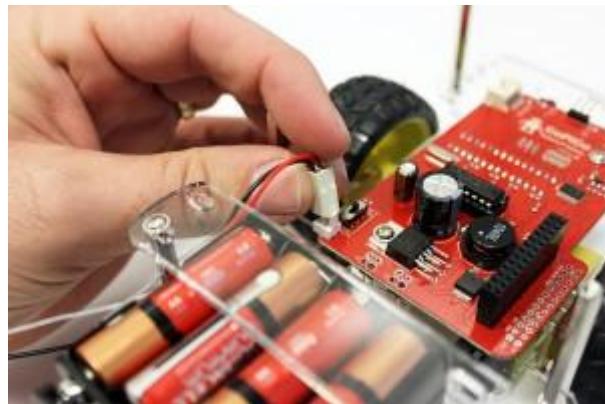
9. Adım : Gücün bağlanması ve testi

Güç kablosu pil takımına bağlandı.



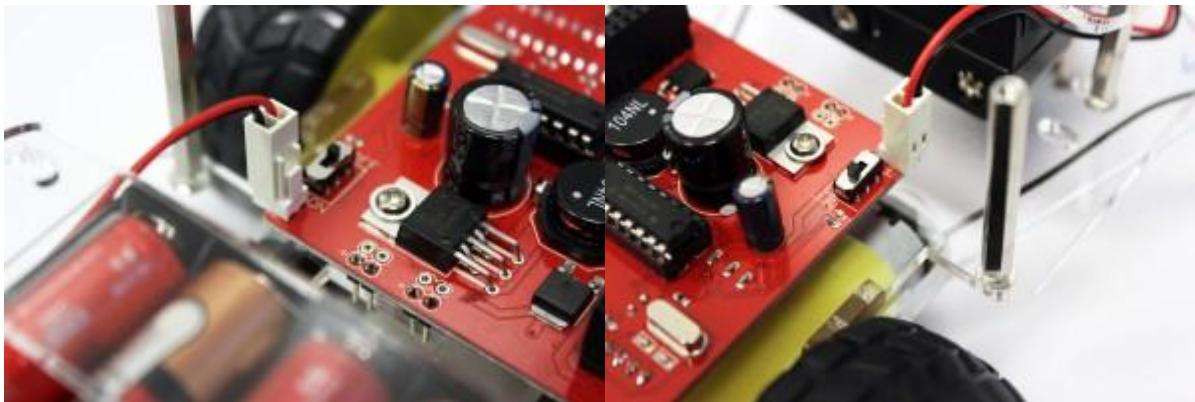
Resim 14

Güç kablosunun diğer ucu kırmızı GoPiGo kartına takıldı.

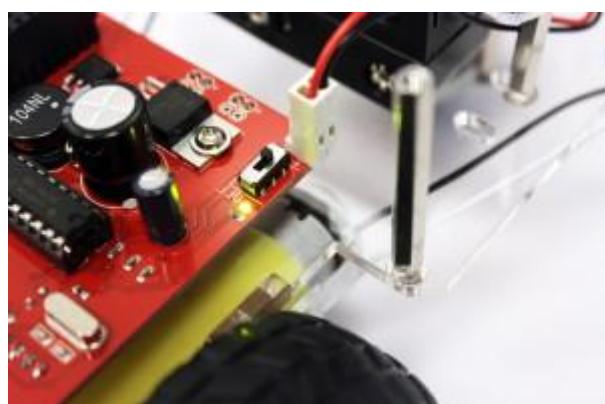


Resim 15

Güç adaptörünün GoPiGo kartına takması.



Güç adaptörü GoPiGo'ya düzgün bir şekilde takılmıştır. Kabloların renklerinin yönelimine dikkat edilmiştir.

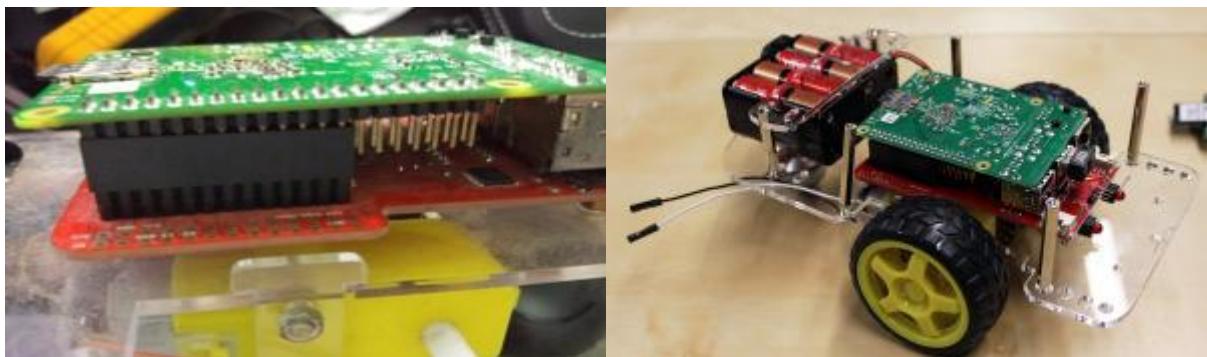


Resim 16

10. ADIM : Raspberry Pi nin GoPiGo Robotuna Takılması

Önce SD kartınızı Raspberry Pi'ye takınız

Daha sonra, gücün kapalı olduğundan emin olunuz ve Raspberry Pi'yi GoPiGo'nun üzerindeki siyah plastik dişli konektöre yerleştiriniz. GPO pinlerine yerleştirerek Raspberry Pi'yi GoPiGo'nun üzerine takınız.

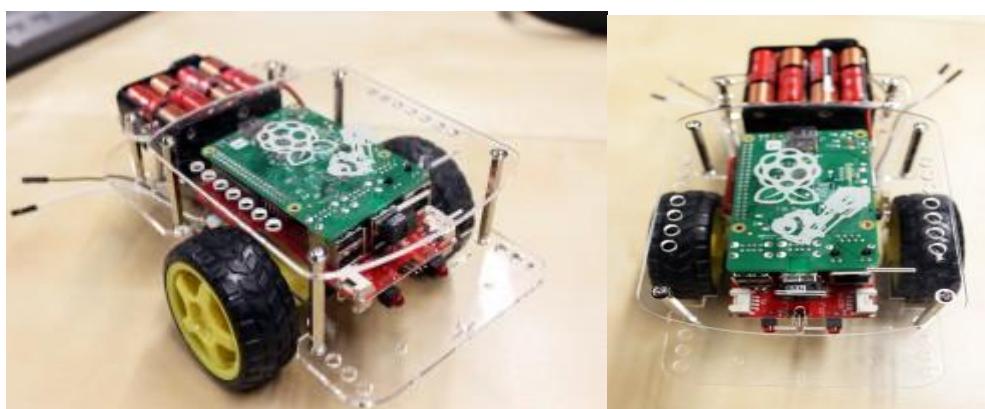


Resim 17

Raspberry Pi Model B + ile birlikte GoPiGo.

11. Adım : Üst Parçanın Takılması

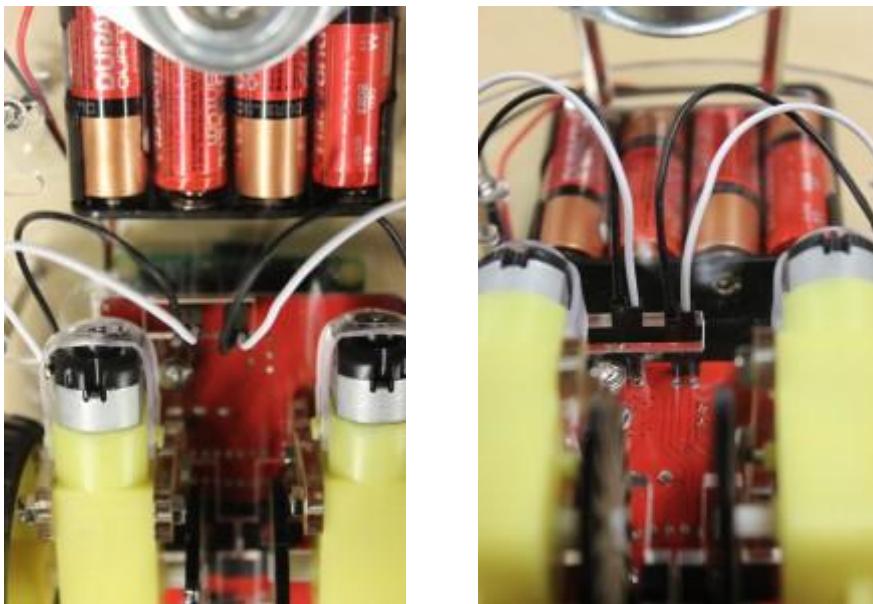
Eğer yeni başladığınız ve programınızı GoPiGo ya bağlamak istiyorsanız şimdilik üstteki akrilik parçayı bırakabilirsiniz. Eğer üstte olmazsa gücün bağlamak ve sd karta ulaşmak daha kolay olacaktır. Ancak tüm malzemeleri toparlamak , biraraya getirmek isterseniz de üste koymaya devam edebilirsiniz.



12. Adım : Motorların Bağlanması

Son olarak güçler kapalıken motoru bağlayacağız. Eğer tam şu an bağlamayı ve programı çalıştırmayı düşünüyorsanız muhtemelen GoPiGo için bilgisayarınıza bağlanmanız ve yazılım ve ürün yazılımı güncelleştirme çalıştmak için motorların bağlanması beklemelisiniz . Motorlar gösterildiği gibi bağlanmıştır. Öncelikle bağlanması için renkler dönüşümlü bir desene yerleştirilmelidir. Motor

konektörlerinin renginin siyah, beyaz, siyah, beyaz çalıştığını dikkat edilmiştir. GoPiGo'yu ilk kez çalıştırıldıkten sonra, motorların geriye doğru gittiğini tespit ederseniz, iki motor grubunu da tersine çevirmelisiniz. Motorlar düz çalışmaya çalışırken ters yönde ilerliyorsa, konektörlerin değişen bir yapıda olup olmadığı kontrol edilmiştir.



2. Aşama SD Card

Bizdeki microSD Kart 16GB 'dır. Bir microSD Kart kullanıyorsanız, kartın bilgisayarınıza sığması için bir microSD kart adaptörüne ihtiyacınız olabilir.

SD Card başlangıç aşamaları:

Windows için Sd Card kurulumu

Önce, burada Sourceforge'dan veya Google Drive'dan image kopyasını indirin .

"Dosyalar" ı tıklayın ve aşağıdaki gibi bir ekran göreceksiniz.

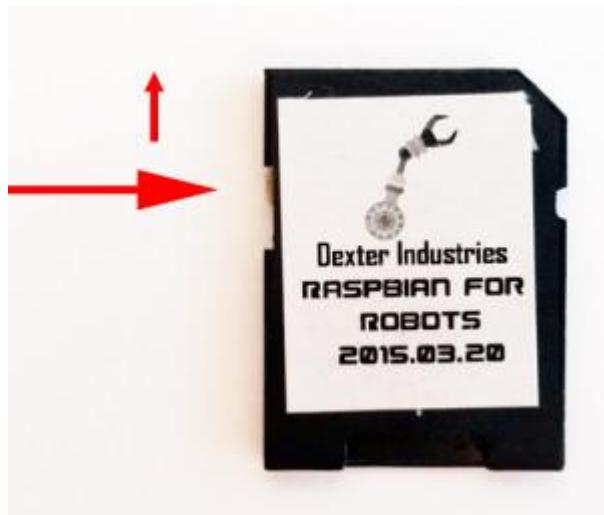
| Name | Modified | Size | Downloads / Week |
|---|------------|--------|------------------|
| 2015-02-17_Dexter_Industries_wheezy.zip | 2015-02-18 | 1.6 GB | 20 |
| 2015-02-17_Dexter_Industries_wheezy.zip | 2015-02-18 | 1.2 GB | 20 |
| 2014-07-30_Dexter_Industries_wheezy.zip | 2014-07-31 | 1.2 GB | 3 |
| 2014-07-30_Dexter_Industries_wheezy.zip | 2014-07-31 | 1.4 GB | 31 |
| 2014-06-15_Dexter_Industries_wheezy.zip | 2014-06-14 | 1.3 GB | 0 |

Resim 18

Sourceforge'de indirilecek dosyalar yukarıdaki gibidir.

PC kullanıcıları en son sıkıştırılmış dosyayı indirmelidir. Bu dizindeki dosyalar tarihe göre düzenlenir ve adlandırılır. En son dosyayı seçin ve farenizi dosya adının üzerine getirin. ".zip" ile biten son dosyayı seçin, üzerine tıklayın ve dosyayı indirin.

Dosya indirildikten sonra dosyanın sıkıştırmasını kaldırabiliriz. Ardından, SD Kart adaptörünüzün kilidi açık olduğundan emin olun.



Sonra, mikro SD kartını mikro SD kart adaptörünüze yerleştirin. SD kartı SD kart okuyucusuna takın.



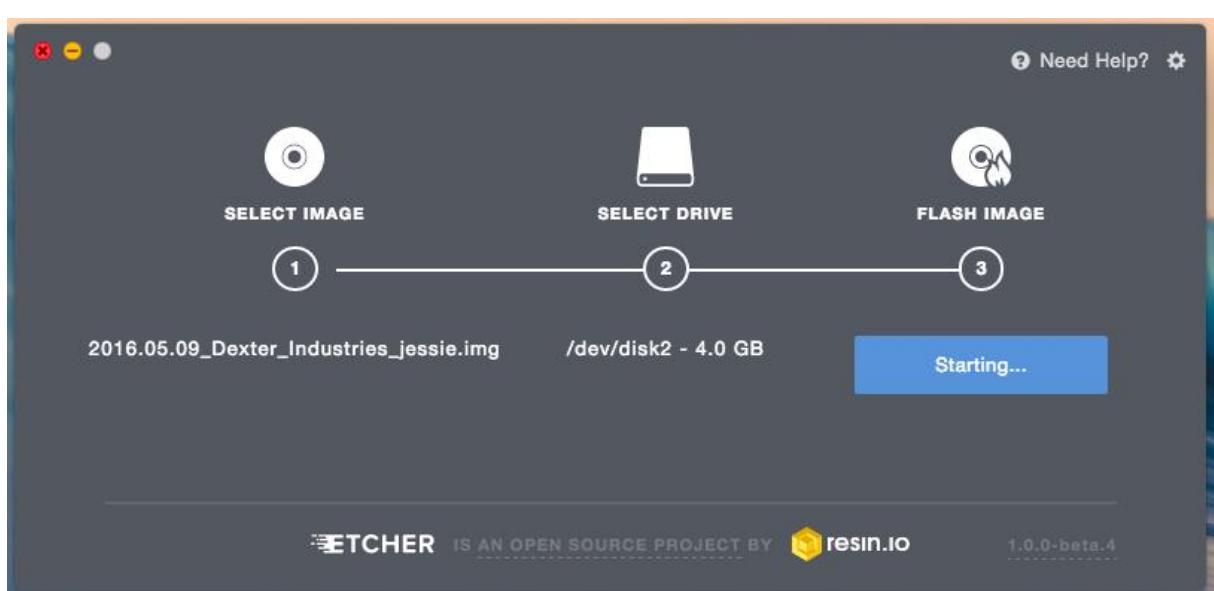
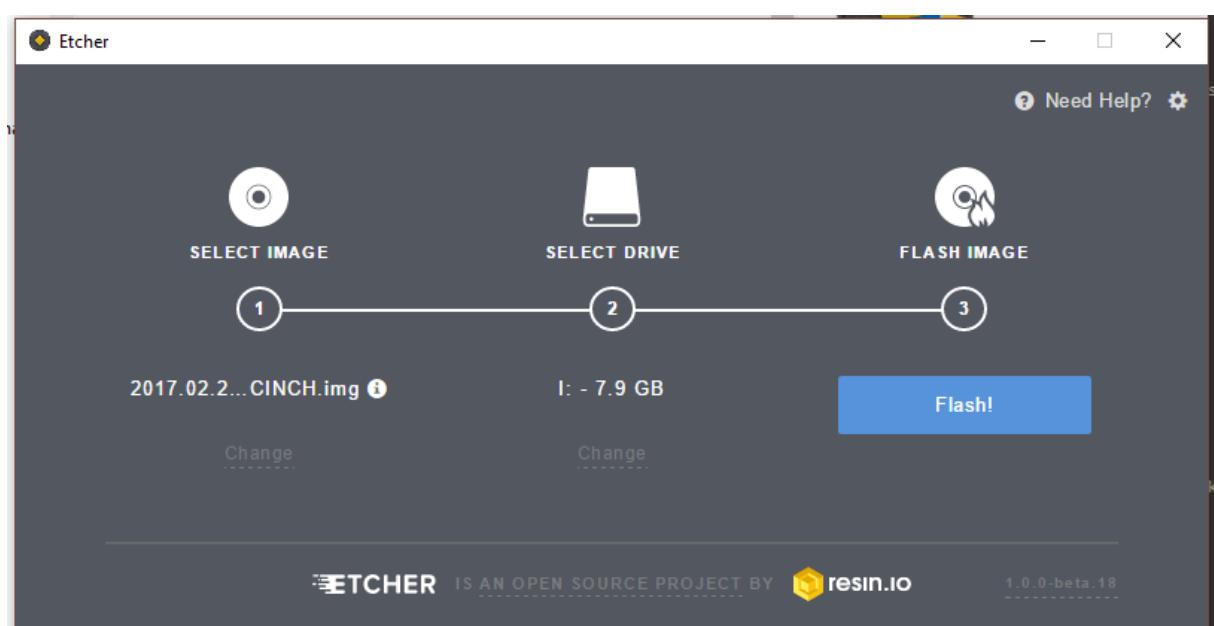
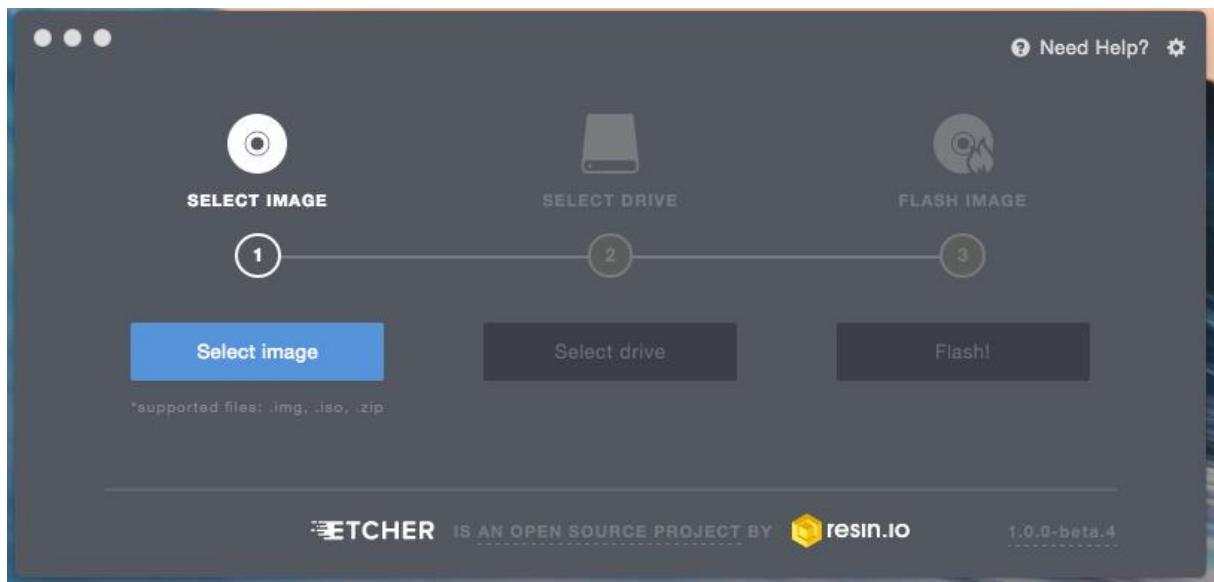
SD Kartını bilgisayarınızın SD Kart okuyucusuna takın.

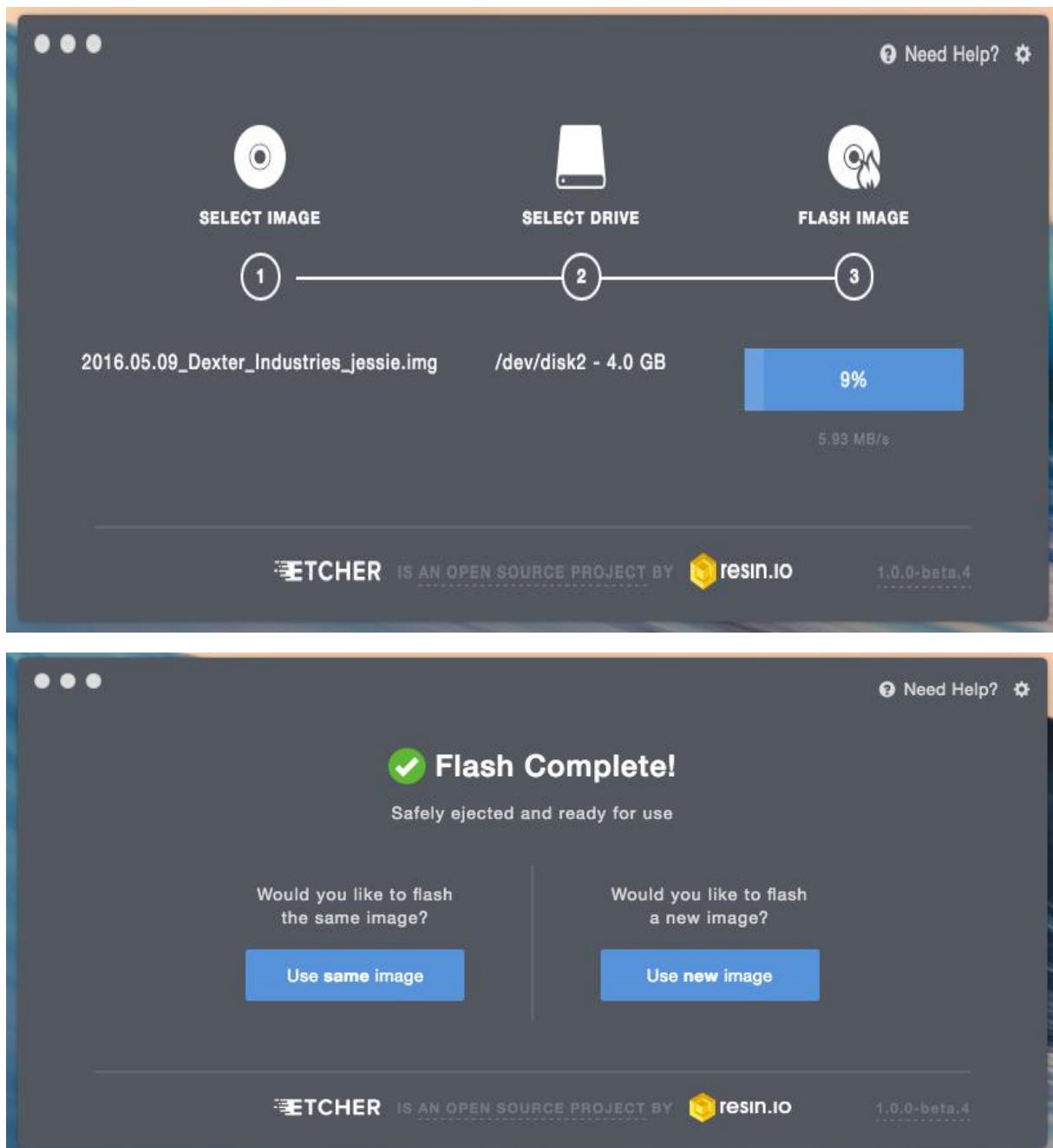
Sonra Etcher'i kuracağınız. etcher.io adresine gidiniz.

Sayfanın ortasındaki büyük yeşil düğmede "Etcher for PC" veya "Windows için İndir> = xx" seçeneğini seçin. Bu, indirilmeye başlamalıdır. Karşılidan yükleme toplam boyutu yaklaşık 80 MB'dır. Bu, `Etcher-1.0.0-beta.19-win32-x64.exe` gibi bir ad indirmelidir İndirme tamamlandıktan sonra, ".exe" dosyasına çift tıklayın ve Etcher'i PC'nize yüklemek için yönergeleri izleyin.

Etcher'i başlatın. Etcher programı görünmelidir.

Şimdi, daha önce sıkıştırmadığımız sıkıştırılmış ".img" dosyasını bulun. Etcher programındaki bu dosyayı mavi "Select Image" düğmesine sürükleyin. Mavi "Select image" düğmesini tıklayabilir ve dosya dizinizdeki ".img" dosyasını bulabilirisiniz.





İlk SD Kartınızı Çalıştırma

Seçenek 1: Dexter Industries'den bir SD Kartı

GoPiGo'nuz için bir Dexter Industries SD kartı satın aldiysanız, bu adımı atlayabilirsiniz: SD kartınız zaten yapılandırılmıştır!

Dexter Industries SD kartında zaten bu yazılım bulunmaktadır, ancak "görüntüsünü genişleteerek" daha fazla alan hazırlayabilirsiniz. Bu, Raspberry Pi'ye bağlandıktan sonra alabileceğiniz basit bir yapılandırma adımıdır.

Raspberry Pi'ye (bir VNC veya terminal üzerinden) bağlandıktan sonra, aşağıdaki komutu çalıştırarak Dexter Industries SD Kart genişletilmektedir:

```
sudo raspi-config
```

SD kartını genişlettikten sonra SD Karttaki tüm alana erişilmektedir.

2. Seçenek: Raspberry Pi 'de GoPiGo için image indirip kurun.

Bunu yaptıktan sonra Raspberry Pi'ye bağlandığınızda terminaldeki komutu çalıştırarak SD Kartınızın tam boyutuna kadar genişleyebileceğini unutmayın:

```
Sudo raspi-config
```

Bu, SD Kart üzerindeki tüm mevcut alanı kullanmanıza izin verir.

Daha gelişmiş ise ve yapılandırmınızı ve donanım profillerinizi güncellemek istenildiğinden, internete bağlanıldı ve aşağıdaki komutları çalıştırıldı:

Sudo apt-get update: apt-get update komutu sayesinde sisteminizde kurulu olan paketlerin, paket deposundaki versiyonları ile farklıları araştırılır ve liste güncellenir. Aslında tek yapılan liste güncellemedir. Herhangi bir güncelleme işleminden önce çalıştırılması mutlaka tavsiye edilir çünkü sisteminizde gerekli güncelleme ve bağımlılıkların çözümü için gerekli bir işlemidir.

apt-get update komutu ile sisteminize herhangi bir kurulum yapılmaz.

Sudo apt-get upgrade: Sisteminizde kurulu olan paketlerin hepsini, apt-get update komutu ile güncellediğiniz listede bulunan en son sürümé yükseltir. Burada dikkat edilmesi gereken nokta **KURULU** olan paketler üzerinde güncelleme işlemi yapıldığıdır. A.lib dosyasının 1.1 sürümünü 1.2 sürümüne günceller. Sadece kurulu olan paketleri en güncel hale getirir.

Bir takım yazılım geliştiriciler veya sistem yöneticileri ihtiyaçları olmayan paket veya kernel dosyalarını, sistemlerinin kararlığını düşünerek yüklemek istemezler. Böyle bir durumda upgrade oldukça kullanışlıdır. Yalnızca mevcut paketler en yeni sürümé yükseltilmiş olur. Önemli bir nokta da, sistemden hiçbir paketin silinmediğidir. Yalnızca mevcut paketler en yeni sürümé yükseltilmiş olur.

3. seçenek:

Dexter Image kullanılmazsa bu ayarlamalar yapılır.

Kendi wheezy kurulumunu kullanmak istenirse, Raspberry Pi'ye aşağıdaki değişiklikler yapmalıdır.

Yükleme komut dosyasını indirebilir ve çalıştırabilirsiniz . Bu, GoPiGo için gerekli tüm paketleri ve bağımlılıkları otomatik olarak indirip yükleyecek ve mevcut Raspbian resminize yükleyecektir.

1). Raspberry Pi üzerinde Terminal açın ve **klon** Dexter Industries GoPiGo Deposu.

```
Git clone https://github.com/DexterInd/GoPiGo.git
```

2). **GoPiGo** adlı bir klasör geçerli çalışma dizininde görünmelidir.

3). Açıldı ve **Kurulum** Dosyaları dizinine gidildi:

```
cd Setup
```

4). Yükleme komut dosyasına **install.sh** adı verildi . Çalıştırılabilir yapıldı

Sudo chmod +x install.sh

5). Komut dosyası çalıştırıldı

Sudo ./install.sh

6). Ekrandaki yönergeleri izleyin. Komut dosyası tüm bağımlılıkları yükleyecek ve tamamlandıktan sonra yeniden başlatılacaktır. Raspbianınız şimdi GoPiGo'yu çalıştırılmaya hazır olmalıdır.

3. Aşama: Power UP(GoPiGo'yu çalıştırın)

GoPiGo'ya güç sağlamak için üç yol vardır:

1. 8 xAA pil kullanımı

Bunlar, GoPiGo'yu çalıştırırken en iyi seçenekir ve taban kiti ile birlikte gelir. Şarj edilebilir iyi pillerle, GoPiGo'yu 8xAA pillerle çalıştırırken, GoPiGo'nun bir saatten daha uzun süre çalışmasını beklenilebilir.

GoPiGo'ya AA pillerle güç sağlamak için:

1. GoPiGo üzerindeki güç konektörünü bulun.
2. Pil adaptörü konektörünü beyaz fişe takın.
3. GoPiGo'yu açmak için güç anahtarını açın veya kapatın. Güç LED'i yanacaktır.

2. USB Micro kablosunu kullanma

GoPiGo'ya Raspberry Pi'den güç vermek için bir USB Micro adaptörü kullanabilirsiniz. Bu, GoPiGo'yu programlarken ve hata ayıklarken yararlıdır. Motorların ve sensörlerin çalışabilmesi için pillerin bağlı olması gereklidir!

3. Özel güç kaynağı kullanma

Daha ileri seviyedeyseniz, GoPiGo'ya güç sağlamak için 9V ve 12V arasında 2A veya daha yüksek bir özel güç kaynağı deneyebilirsiniz. Kendi özel güç kaynaklarınızı oluşturmak, GoPiGo'yu uzun süre kullanışlı bir şekilde test etmek istiyorsanız faydalı olabilir. GoPiGo, güç kaynağında ters polarite korumasına sahip değildir: kendi sorumluluğundadır kendi özel güç kaynağını oluşturulmalıdır.

Powerbank ile Raspberry Pi'a enerji sağlanabilir.

4.Aşama : GoPiGo ya bağlanma

GoPiGo robotunuza bağlanın.

Seçenek1: Masaüstü Olarak Kurulum

Bir monitör, fare, klavye kullanarak: Bu seçenek, Raspberry Pi'ye aşina olanlar için harika seçenektir. GoPiGo, klavye, monitör ve fare için yer açmak için tasarlanmıştır. GoPiGo'ya bir WIFI cihazı eklemek için USB hub'a ihtiyacınız olduğunu görebilirsiniz. Bu yapılandırmamanın avantajı, WIFI'yi Raspberry Pi'de yapılandırmamanız veya bilgisayarda uzaktan oturum açmak zorunda kalmamanızdır.

Seçenek 2: Bilgisayarınıza bir ethernet kablosuyla bağlama

Dexter Industries SD kart Bonjour, saqlar yazılım ile yapılandırılmış şekilde gelir.

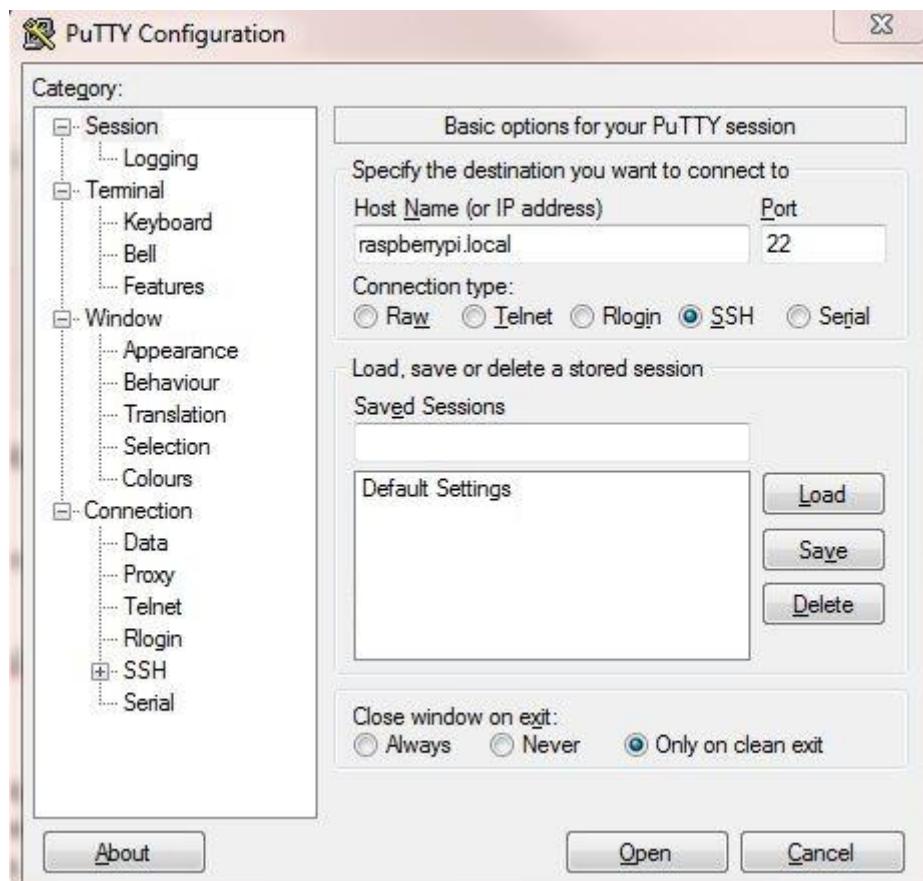
- Windows kullanıyorsanız ve yüklü iTunes a sahip Bonjour indirmeseniz burada kurun, ardından Aşağıdaki Windows bölümünü aşağı hareket ettirin.
- Mac kullanıyorsanız veya zaten iTunes yükleyse, ihtiyacınız olan her şeye sahipsiniz ve aşağıdaki Mac bölümüne geçmeniz yeterlidir.
- Kendi resminizi kullanıyorsanız , sadece Raspbain imajınıza avahi *-daemon* yükleyin :

```
Sudo apt-get update
```

```
sudo apt-get install libnss-mdns
```

Windows

1. GoPiGo kartının kapalı olduğundan, motorların bağlı olmadığından ve pillerin bağlı olmadığından emin olun.
2. SD kartınızın Raspberry Pi'de olduğundan emin olun.
3. Wifi dongle'inizi Raspberry Pi üzerindeki bir USB portuna takın.
4. Ethernet kablosunu bilgisayarınıza ve Raspberry Pi'ye bağlayın.
5. Güç adaptörünü Raspberry Pi'ye takın ve daha sonra, gücü açmak için yuvaya takın. Güç duvara bağlandıktan sonra, GoPiGo da açılacaktır. İsterseniz kendiniz de açabilirsiniz.
6. Bilgisayarınızda ,putty i açın . İmageniz 2015.11.01'den önce ise, [lütfen bu talimatları kullanarak güncelleşeyin](#) . Resminiz 2015.11.01'den sonra ise, ana makine adınız "dex.local" ve şifreniz "robots1234" şeklindedir. Open'a basın.



Bir terminal açacak bir Kullanıcı Adı ve Şifre isteyecektir. **Kullanıcı adı** "pi" ve şifre "**robots1234**". Kimlik bilgilerini girdikten sonra, Raspberry Pi terminaline giriş yapacaksınız.

```
pi@raspberrypi: ~
login as: pi
pi@raspberrypi.local's password:
Linux raspberrypi 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 armv6l

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: Sun May 11 19:28:36 2014
pi@raspberrypi ~ $
```

Wifi Bağlanma

(dexter sd kart imangenizi indirdiyseniz bu adımı atlayabilirsiniz)

GoPiGo'yu bilgisayarınıza ve duvara sürekli bağlı kalmadan uzaktan çalıştırınmak için Wifi'yi kurmak isteyeceğiz!

1. Wifi dongle'ınız zaten Raspberry Pi'de olmalı ve terminale bağlanmalısınız.

Wifi dongle'ınızı düzgün çalışacak şekilde ayarlamak için bazı yazılımlara ihtiyacınız vardır. Ayrıca, bilgisayarlarınızın hangi kablosuz ağ kullanıldığını ve bu kablosuz ağın şifresini bildiğinizden emin olun.

2. Şuraya gidin: <https://www.realvnc.com/download/vnc/>

3. Bilgisayarınızla eşleşen herhangi bir sürüm için **Download** düğmesini tıklayın .

4. Çalışmakta olduğunuz OS X sürümünü bilmiyorsanız, ekranınızın sol üst köşesindeki elma simgesini tıklayın ve "About This Mac" (Bu Mac Hakkında) seçeneğini seçin ve çalıştırın. Olduğunuz OS X sürümünü görüntüleyecektir.

5. VNC dosyasını tarayıcınızda açın, orada değilse, **Finder->Downloads** bölümüne gidin ve orada bulacaksınızdır.

6. **Continue** öğesini seçin.

7. **Continue** öğesini seçin.

8. Bu açılır pencerede **Select Agree** öğesini seçin.

9. **Continue** öğesini seçin.

10. Sadece VNC Viewer'ı görmek için VNC Server'ı kaldırın. Continue öğesini seçin.

11. Install öğesini seçin ve kurulumun başarılı olduğuna dair onay alacaksınız. Close öğesini seçin.

12. Finder-> Applications-> Real VNC -> VNC ye gidin ve VNC Görüntüleyicisi'ni açın.

13. VNC Sunucusu kutusuna şunu yazın: dex.local: 1

Bu, VNC Görüntüleyiciye aşağıdakileri ifade ediyor:

- "Dex" i arayın
- Yer "local" (bilgisayarınıza bağlı)
- Port = 1 (port ...)

Muhtemelen bir uyarı alacaksınız.

Sadece, Raspberry Pi'niz ile bilgisayarınız arasındaki bağlantı şifrelenmez ve bu da güvenli bir veri aktarımı yöntemidir. Evde ya da güvenilir bir yerde yapıyorsanız bu büyük bir endişe değil, çünkü başka hiç kimse bilgisayarınıza erişemez.

14. Continue (Devam) seçeneğini seçin (ve bu konuda tekrar uyarı olmak istemiyorsanız onay kutusunu işaretleyin).

15. Bilgisayarınıza bağlı Raspberry Pi "sabit sürücüye" (SD kartı) erişmek için bir şifre girmeniz istenecektir. Tüm Dexter Industry SD kartlarının şifresi: "robots1234"

16. Tür girin: robots1234

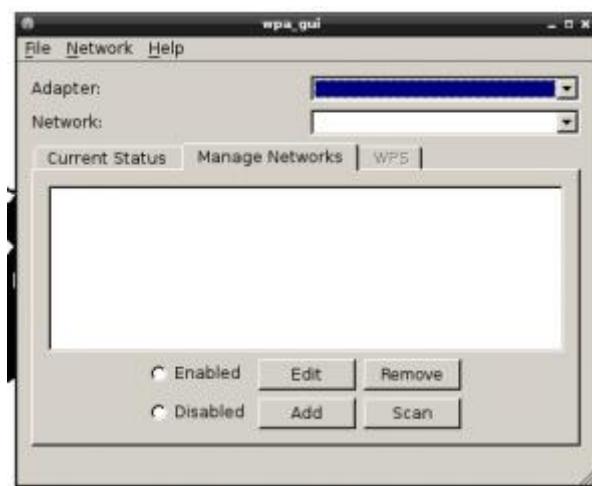
17. OK seçeneğini seçin.

Bu ekran VNC masaüstü. Unutmayın, VNC (sanal ağ bilgisayarı) temel olarak bilgisayarınızın içerisinde çalıştırığınız küçük bir sanal bilgisayardır.

Normal masaüstünüzün yaptığı gibi çalışır. Gördüğünüz dosyalar ve kisayollar, Raspberry Pi'nizdeki SD karttaki dosyaları açıp programları çalıştıracaktır.

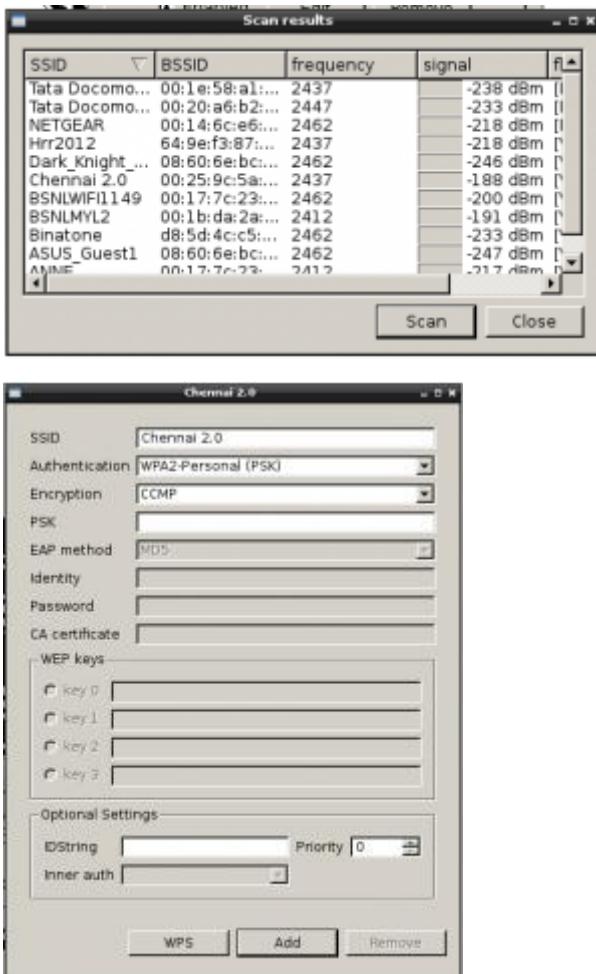
18. Masaüstünün ortasındaki "WiFi Config" yazan simgeyi tıklayın

19. İkinci sekmeyi seçin: **Manage Networks**. Scan düğmesini seçin. Program yakınlardaki kablosuz ağları tarayacaktır.



Bu biraz zaman alabilir ve bu arada ekranınız hala durgun olacak.

20. Bilgisayarınızın açık olduğu aynı ağı çift tıklatarak seçin ve PSK'sını (kablosuz ağınızın şifresi) gireceğiniz bir açılır pencere elde edin.



21. Tarama ekranına geri dönersiniz ve **Close** öğesini seçin .

22. İlk Olan Mevcut Durum sekmesini tıklayın, ardından Açıılır Ağ listesinden ağınızı seçin ve Connect öğesini seçin.

23. **Bu işlem birkaç dakika alabilir.** Bağlanmaya çalışırsa ve IP adresi olmadan geri gelirse, panik yapmayın. Tekrar bağlan'ı seçin ve 3-5 dakika sonra bir IP adresinin gösterilmesi gereklidir.

24. Gösterilecek bir IP adresi almazsanız, aşağıdakileri kontrol edin:

- Manage networks düğüm, çift tıklatarak kablosuz ağınızı seçin ve PSK'yi (şifre) tekrar girin. Daha önce şifreyi yanlış girmiş olabilirsiniz.

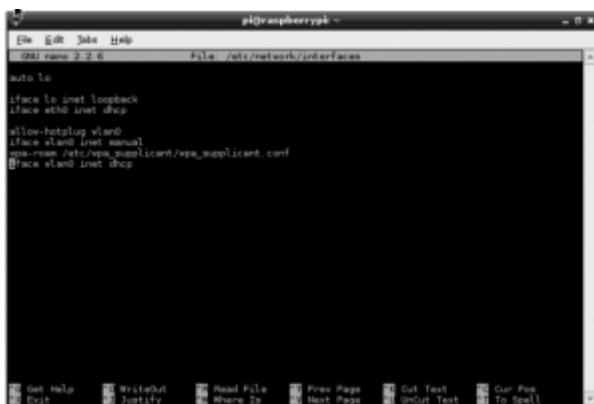
Bağlı olup olmadığını görmek için iyi bir test:

1. VNC masaüstündeki (Dexter Industries masaüstü) LXTerminal simgesini açın.
2. 'ping google.com' yazın.
3. Aşağıdaki resim gibi bir yanıt alırsanız, o zaman BAĞLANTIDASINIZ! Bunun durdurulması için "Ctrl + c" ye basın.
4. Wifi Dongle'ı kontrol edin ve güvenli bir şekilde takıldığından emin olun ve kırmızı yanıp sönen ışık olmalıdır.

Terminali veya SSH'yi kullanarak wifi kurulumu yapmak istiyorsanız, Terminalde, arayüz dosyanızı aşağıdaki komutu kullanarak düzenlemek üzere açın:

```
sudo nano /etc/network/interfaces
```

4.2 Interfaces File: Arayüz dosyanız şöyle görünmelidir:

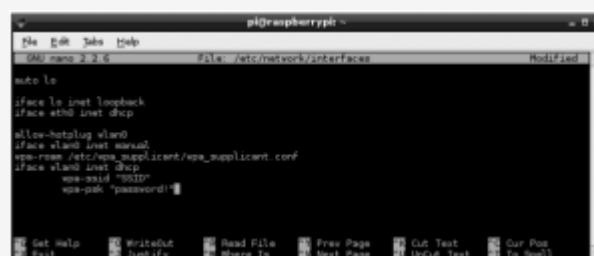


```
pi@raspberrypi: ~
File Edit Tasks Help
File nano 2.2.6      File: /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface wlan0 inet dhcp
```

4.3 Dosyayı düzenleyin. Dosyayı, ağ yapılandırma bilgilerini ve ağlarınızın SSID'sini ve son satırı altındaki şifre deyimini ekleyerek düzenleyin.

Wpa-ssid "ssid"

Wpa-psk "şifre"



```
pi@raspberrypi: ~
File Edit Tasks Help
File nano 2.2.6      File: /etc/network/interfaces      Modified
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface wlan0 inet dhcp
    wpa-ssid "SSID"
    wpa-psk "password!"
```

4.4 Dosyanızı kaydedin. Bilgilerinizi ekledikten sonra "Ctrl-x" yazıp dosyayı kaydetmeyi seçerek kaydedin ve çıxın.

4.5 Raspberry Pi'nizi yeniden başlatın.

4.6 Kablosuz Adaptörünüzü Test Edin. Pi'nizin WiFi Adaptörünü tanıdığını test edin. GoPiGo'nun terminalinde " ifconfig " yazarak test edebilirsiniz. "Wlan0" görüntüleniyorsa, wifi dongle tanınır.

```

pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet HWaddr 00:1c:29:eb:5e:0b
          inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.0.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:252 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:27506 (36.6 KiB) TX bytes:23024 (22.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
                  UP LOOPBACK RUNNING MTU:16436 Metric:1
                  RX packets:66 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:6404 (6.3 KiB) TX bytes:6404 (6.3 KiB)

wlan0     Link encap:Ethernet HWaddr 00:17:7c:22:81:f1
          inet addr:192.168.2.10 Bcast:192.168.2.255 Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:222 errors:0 dropped:6 overruns:0 frame:0
                  TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:27392 (26.7 KiB) TX bytes:9901 (9.6 KiB)

pi@raspberrypi ~ $ 

```

Wlan0, Raspberry Pi tarafından başarıyla tanınır. Raspberry Pi, bir wifi cihazının kurulu olduğunu biliyor.

5. Bağlantıyı kontrol edin. Ağa başarıyla bağlanıp bağlanmadığınızı kontrol edin. Komut satırına şunu yazın:

ifconfig

RX ve TX 0 bayt gösteriyorsa hiçbir veri alışverişi yapılmamıştır. Aşağıda başarıyla bağlanmış Raspberry Pi'nin bir kablosuz ağa bağlılığı görülmektedir.

```

wlan0      Link encap:Ethernet HWaddr 00:17:7c:22:81:f1
          inet addr:192.168.2.10 Bcast:192.168.2.255 Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:222 errors:0 dropped:6 overruns:0 frame:0
                  TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:27392 (26.7 KiB) TX bytes:9901 (9.6 KiB)

pi@raspberrypi ~ $ 

```

Bir ağa başarıyla bağlandı. Altaki RX baytları ve TX baytlarının 0'dan büyük değerlere sahip olduğuna işaret ederek, bağlı olduğumuzu ve bilgi gönderip alacağımızı belirttiğimizi unutmayın.

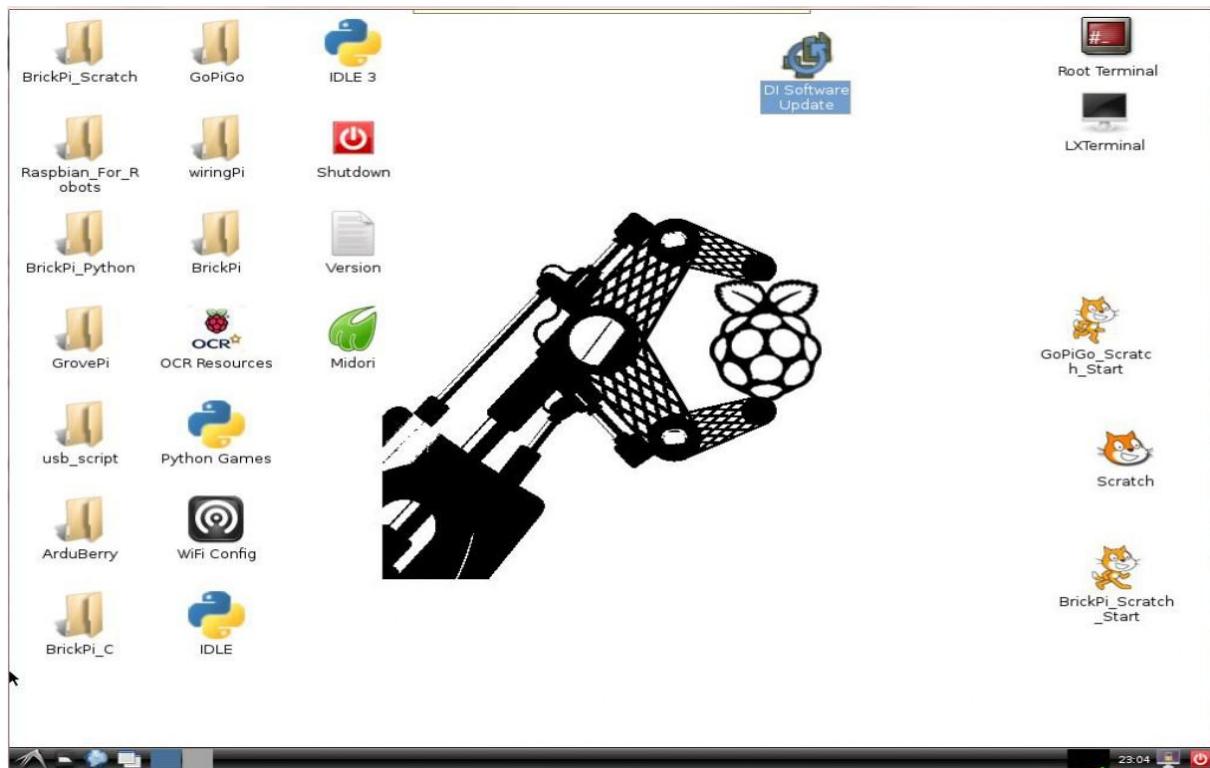
5.Aşama : GoPiGo cihazını ilk defa başlatma

GoPiGo'ya bağlandıktan sonra yapılması gereken ilk şey, GoPiGo'daki mevcut yazılımı ve firmware yi güncellemektir. GoPiGo yazılımını sürekli geliştirmiyoruz ve hataları düzeltiyoruz. Yazılımınızı düzenli olarak güncellemek iyi bir uygulamadır.

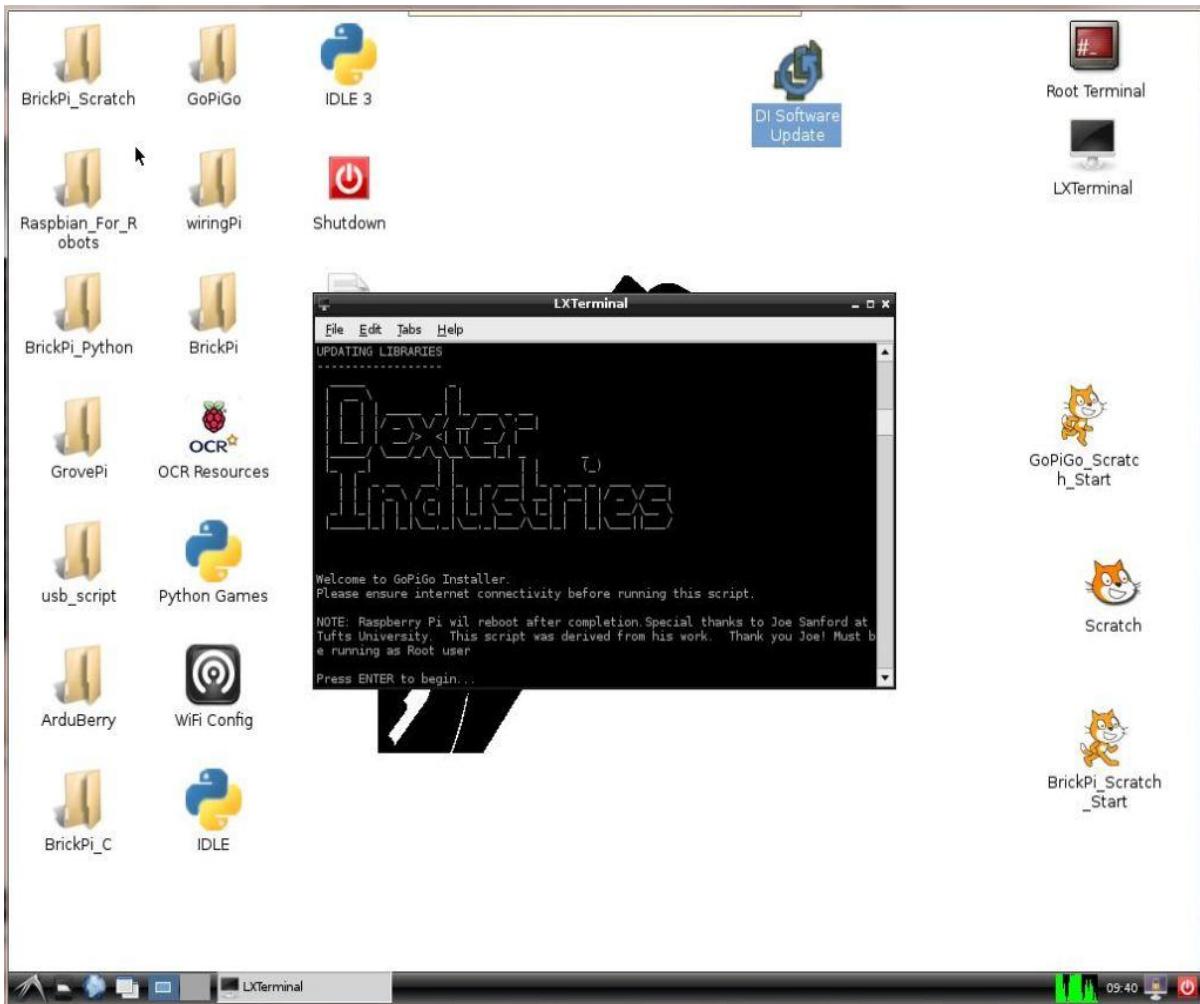
Yazılımı kullanmadan önce GoPiGo'daki Yazılım'ı ve ürün yazılımını güncellemenizi öneririz. Yazılımı ve firmware yi güncellemek için aşağıdaki adımları izleyin:

1. Rasberry Pi'yi GoPiGo'ya bağlayın ve açın.
NOT : Güncellenirken motoru GoPiGo'ya bağlayan tellerin bağlantısını kesmenizi öneririz . Aksi takdirde üretici yazılımı düzgün şekilde güncellenmeyebilir.

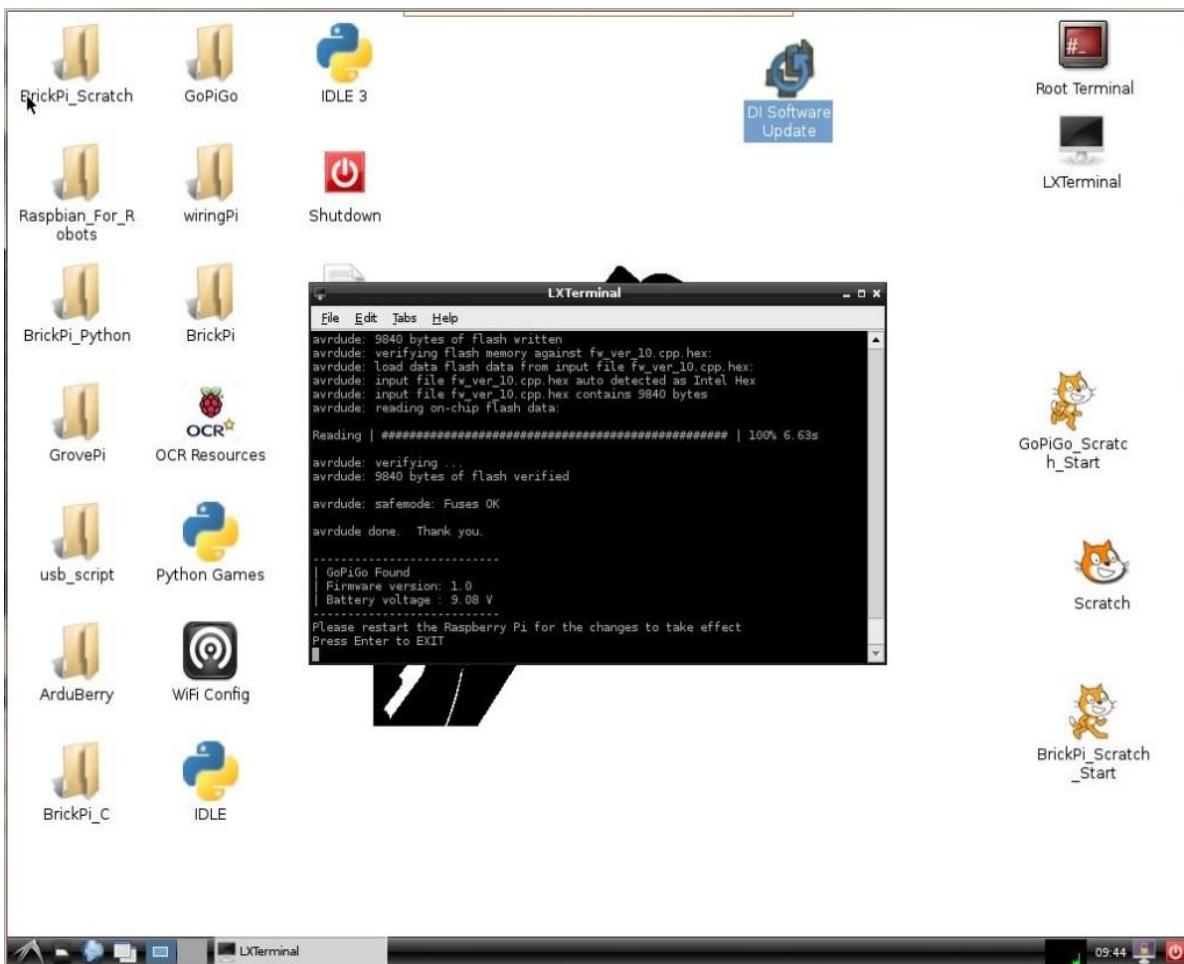
Dexter Industries SD kartını veya görüntüyü kullanıyorsanız, yazılımı güncellemek oldukça basit olmalıdır. Aşağıda Pi'nin **internete bağlı** olduğundan ve VNC'yi açığınızdan emin olun . **Masaüstündeki DI Software Update simgesini** çift tıklatın .



Güncelleme, internetten bir dosya getirip çalıştırır. İstendiğinde " **Enter** " a basın



Güncelme tamamlandığında, firmware sürümünü **1.1** olarak görmelisiniz. Başka bir şey görürseniz, bu güncellemenin düzgün bitmediği anlamına gelir.



Şimdi GoPiGo'yu yeniden başlatıp SSH ile WiFi veya Ethernet kullanarak GoPiGo'ya giriş yapın (GoPiGo'ya erişmek için putty gibi bir istemci kullanın)

```

pi@raspberrypi: ~
login as: pi
pi@169.254.127.191's password:
Linux raspberrypi 3.6.11+ #538 PREEMPT Fri Aug 30 20:42:08 BST 2013 armv6l

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: Mon Apr 7 18:10:00 2014 from john-hp.local
pi@raspberrypi ~ $ 

```

Dizini, Masaüstünde veya kopyaladığınız dizindeki GoPiGo klasörüyle değiştirin.

```
cd Desktop/GoPiGo/
```

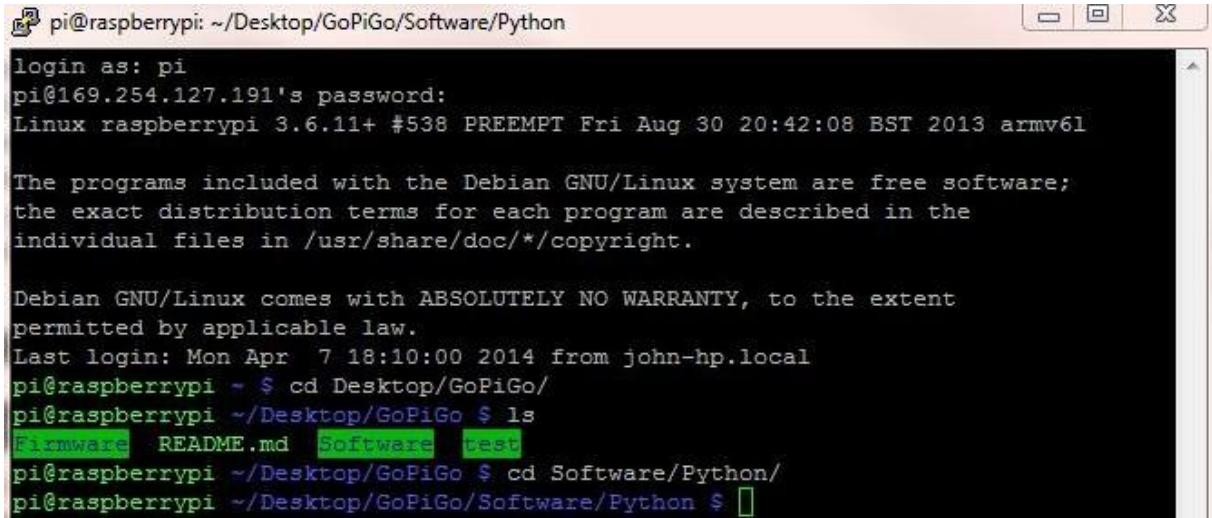
```

pi@raspberrypi ~ $ cd Desktop/GoPiGo/
pi@raspberrypi ~/Desktop/GoPiGo $ ls
Firmware README.md Software test
pi@raspberrypi ~/Desktop/GoPiGo $ 

```

Şimdi Yazılım klasöründeki Python Klasörüne gidin:

```
cd Software/Python/
```



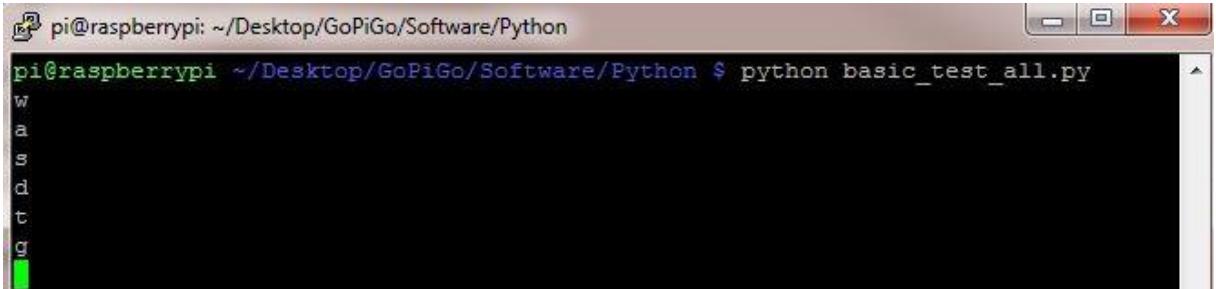
```
pi@raspberrypi: ~/Desktop/GoPiGo/Software/Python
login as: pi
pi@169.254.127.191's password:
Linux raspberrypi 3.6.11+ #538 PREEMPT Fri Aug 30 20:42:08 BST 2013 armv6l

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: Mon Apr  7 18:10:00 2014 from john-hp.local
pi@raspberrypi ~ $ cd Desktop/GoPiGo/
pi@raspberrypi ~/Desktop/GoPiGo $ ls
Firmware README.md Software test
pi@raspberrypi ~/Desktop/GoPiGo $ cd Software/Python/
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python $ 
```

Son olarak, robotu test etmek için temel python programını çalıştırın:

```
sudo python basic_test_all.py
```



```
pi@raspberrypi: ~/Desktop/GoPiGo/Software/Python
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python $ python basic_test_all.py
w
a
s
d
t
g
```

Press:

- “w” to move the GoPiGo forward
- “a” to move left
- “s” to move back
- “d” to move right
- “x” to stop
- “t” to increase speed
- “g” to decrease speed

6.Aşama : Opencv Kurulumu

1.Gerekli paketleri yüklemeden önce sistem güncellenir

```
sudo apt-get update
```

2. Güncellemeye tamamlandıktan sonra, OpenCV'yi derlemek için gerekli paketleri indirmeye devam edeceğiz.

```
sudo apt-get install build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale
```

```
sudo apt-get install python3.5-dev
```

```
python3.5-config --includes
```

3. Çıktı aşağıdakine benzer bir şeye benzeyecektir:

```
-I/usr/include/python3.5m -I/usr/include/x86_64-linux-gnu/python3.5m
```

4. Çıktının ilk kısmı beklenen konumdur ve ikinci bölüm yapılandırma dosyasının geçerli konumunu göstermektedir. Bu sorunu çözmek için dosyayı geçerli konumdan beklenen konuma kopyalayacağız.

```
sudo cp /usr/include/x86_64-linux-gnu/python3.5m/pyconfig.h /usr/include/python3.5m/
```

5. Opencv kaynak kodu inidirilir

```
mkdir OpenCV-tmp  
cd OpenCV-tmp  
git clone https://github.com/Itseez/opencv.git
```

6. Kaynak derlenmeden önce makefile dosyası oluşturulur

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..../opencv-3
```

7. Makefile oluşturulduktan sonra kurulum dosyalarını oluşturulur

```
make -j $(nproc --all)
```

8. Kurulum tamamlandıktan sonra, make dosyasını çalıştırarak OpenCV kurulur

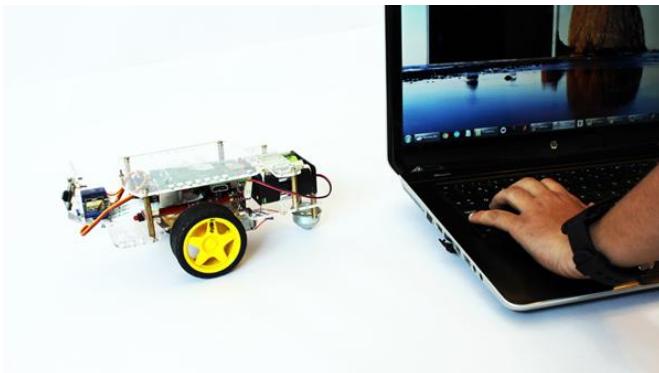
```
sudo make install
```

9. OpenCV kurulumu tamamlandıktan sonra, OpenCV'nin düzgün bir şekilde kurulduğundan emin olmak için kontrol edilir

```
1 >>> import cv2  
2 >>> cv2.__version__  
3 '3.1.0-dev'
```

Gopigo ile test ettiğimiz uygulamalar

Basic_Robot_Control



Bu uygulama gopigoyu kontrol etmek için temel işlevleri yerine getirir. İleri, Geri , Sol, Sağ, Dur ,Hızı Arttır, Hızı Azalt gibi komutları kullanarak gopigoyu haraket ettirmemizi sağlar.

Örneği çalıştırın:

1. Temel Robot Kontrolü klasörüğe gidin:

```
cd GoPiGo/Software/Python/Examples/Basic Robot Control/
```

```
pi@raspberrypi ~/Desktop $ cd GoPiGo/Software/Python/Examples/Basic\ Robot\ Cont .rol/
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Basic Robot Control $
```

2. Basic_robot_control.py örneğini başlatın

```
sudo python basic_robot.py
```

```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Basic Robot Control
$ sudo python basic_robot.py
This is a basic example for the GoPiGo Robot control
Press:
    w: Move GoPiGo Robot forward
    a: Turn GoPiGo Robot left
    d: Turn GoPiGo Robot right
    s: Move GoPiGo Robot backward
    t: Increase speed
    g: Decrease speed
    x: Stop GoPiGo Robot
    z: Exit

Enter the Command: w
Enter the Command: a
Enter the Command: s
Enter the Command: d
Enter the Command: x
Enter the Command: z
Exiting
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Basic Robot Control
$
```

Basic_Robot_Control_GUI

Bir arayüz oluşturur ve temel işlevleri yerine getirir.

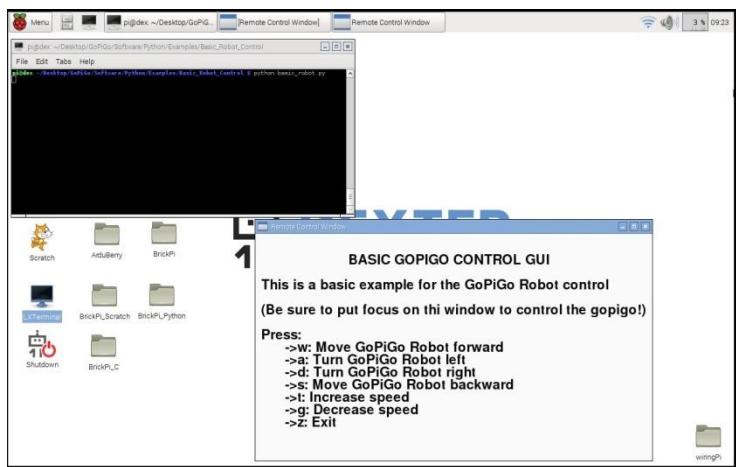
Örneği çalıştırın:

1. Basic_Robot_Control_GUI klasörüne gidin:

```
cd Desktop/GoPiGo/Software/Python/Examples/ Basic_Robot_Control_GUI/
```

2 Basic_Robot_Control_GUI örneğini başlatın

```
sudo python Basic_Robot_Control_GUI.py
```



Basic_Servo

Servo pozisyonuna göre sağa veya sola doğru belirli açılarla hareket etmesini sağlar.

```
cd Desktop/GoPiGo/Software/Python/Examples/Basic Servo/
```

```
pi@raspberrypi ~ $ cd Desktop/GoPiGo/Software/Python/Examples/Basic\ Servo/
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Basic Servo $
```

```
sudo python basic_servo.py
```

```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Basic Servo $ sudo
python basic_servo.py
CONTROLS
a: move servo left
d: move servo right
s: move servo home
Press ENTER to send the commands
a
a
a
d
d
d
s
^CTraceback (most recent call last):
  File "basic_servo.py", line 30, in <module>
    inp=raw_input()                                     # Get keyboard input.
KeyboardInterrupt
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Basic Servo $
```

Browser_Streaming_Robot

```
cd Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot/
```

```
pi@raspberrypi ~ $ cd Desktop/GoPiGo/Software/Python/Examples/Browser\ Streaming\ Robot/
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot $
```

```
sudo chmod +x robot_web_server.py
```

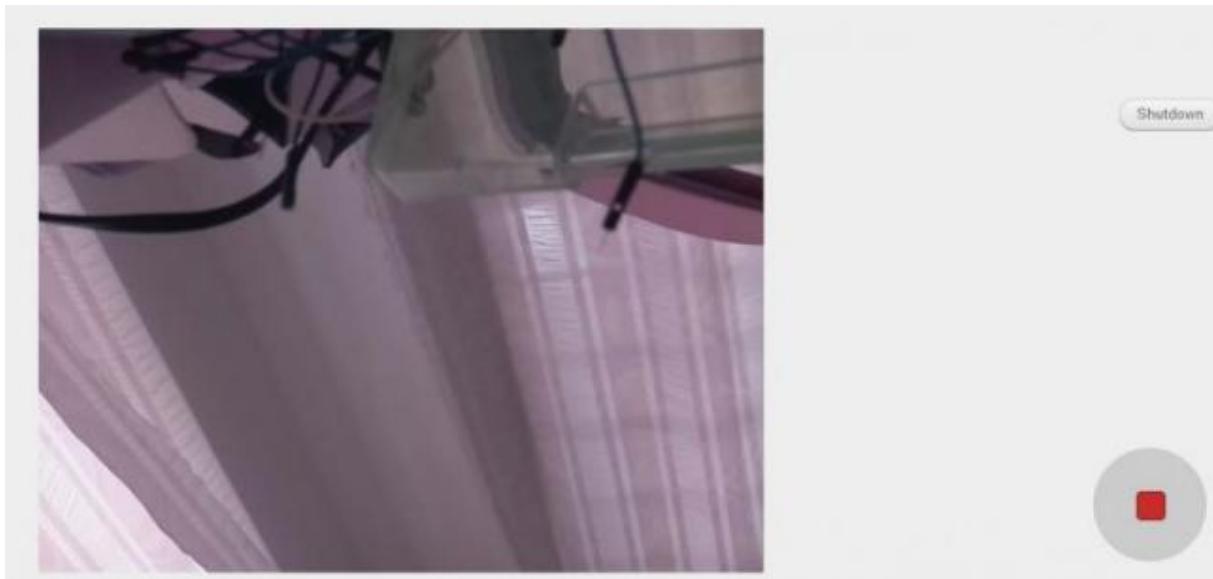
```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot $ sudo chmod +x ro
bot_web_server.py
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot $
```

```
sudo ./robot_web_server.py
```

```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot $ sudo ./robot_web_
server.py

sockjs.tornado will use json module
/home/pi/Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot/www
Starting web server...
```

Şimdi tarayıcıyı açın ve tarayıcıda **dex.local** yazın ve programı başlatmak için enter tuşuna basın. Ayrıca terminalde işlem yapmaya devam eder.(Tarayıcı ile raspberry aynı ağda olmalıdır)



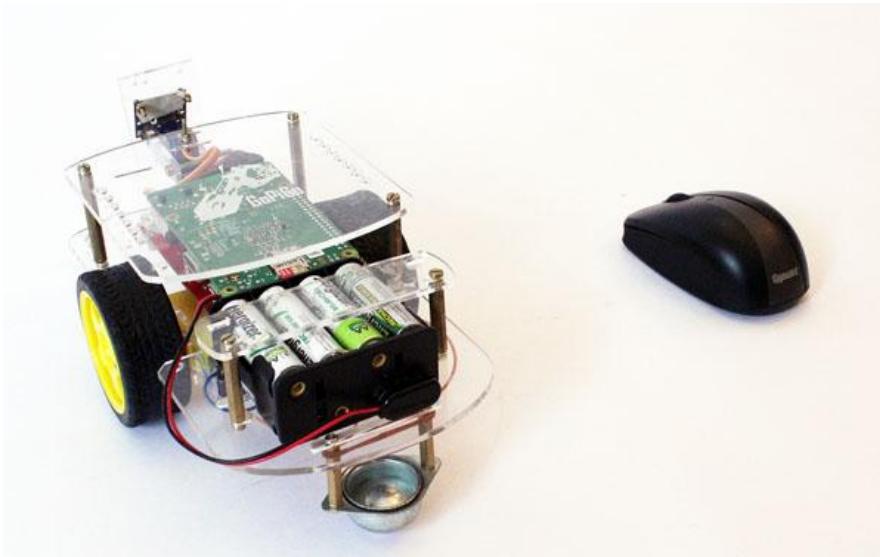
```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot $ server.py

sockjs.tornado will use json module
/home/pi/Desktop/GoPiGo/Software/Python/Examples/Browser Streaming Robot/www
Starting web server...
304 GET / (192.168.43.243) 80.54ms
200 GET /robot_control/info (192.168.43.243) 9.76ms
Image encoder buffer num 3, buffer size 81920
Image encoder buffer num 3, buffer size 81920
```

Telefondan kontrol etmet için robotun ip öğrendikten sonra tarayıcıya ip adresini girmemiz yeterli olacaktır.Şifre sorarsa(robots1234)



Mouse_Control



Mause haraketine göre sağa ve sola doğru haraket eder.

Diğer bir uygulaması ise mause tuşları ile kontrolünün sağlanmasıdır.

```
cd Desktop/GoPiGo/Software/Python/Examples/Mouse Control/
```

```
pi@raspberrypi ~ $ cd Desktop/GoPiGo/Software/Python/Examples/Mouse\ Control/  
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Mouse Control $
```

```
sudo python mouse_control_movement.py
```

```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Mouse Control $ sudo python mouse_control_m  
ovement.py  
Press Enter to start
```

```
pi@raspberrypi ~/Desktop/GoPiGo/Software/Python/Examples/Mouse Control $ sudo python mouse_control_m  
ovement.py  
Press Enter to start  
  
-1 1  
-2 0  
-4 0  
-5 0  
-1 0  
-3 0  
-6 1  
-2 1  
-3 1  
-6 2  
-3 0  
-3 1  
-7 2  
-3 0  
-3 1  
-5 0  
-2 1  
-3 0  
-1 0  
-2 0  
-2 0  
-3 1  
-4 0
```

Ctrl+C ile program sonlandırılır

I²C Haberleşme Protokolü

I²C protokolünde temel olarak iki hat vardır. Bunlar SCL ve SDA olup, SCL (SerialClock) veri senkronizasyonu için kullanılan clock darbeleri hattıdır, SDA (SerialData) ise veri hattıdır. Ayrıca elemanların kendine has yazma koruma(WP-Write Protect), çıkış izin(OE-Output Enable) gibi uçları bulunabilir. Ancak bu kontroller bu protokol için bir standart olmadığından kullanıcı bunları kendisi düzenlemek zorundadır.

I²C protokolünde gönderilen veri ve okunan veri aynı hat üzerinden edilir, yani SDA hattından transfer edilir. SDA pini sürekli olarak bir pull-up direnciyle V_{cc}'ye (TTL için +5V besleme gerilimi) bağlı olduğundan hattaki start ve stop bitlerinin anlaşılması hatta lojik 0 olup olmamasıyla anlaşılır. Ayrıca hattın sürekli yüksek(high) lojik olarak kurulması bazı elemanların yüksek empedans konumlarında bile hattan lojik 1 okunmasını sağlar.

I²C haberleşmesinin nasıl çalıştığını şu şekilde özetleyebiliriz:



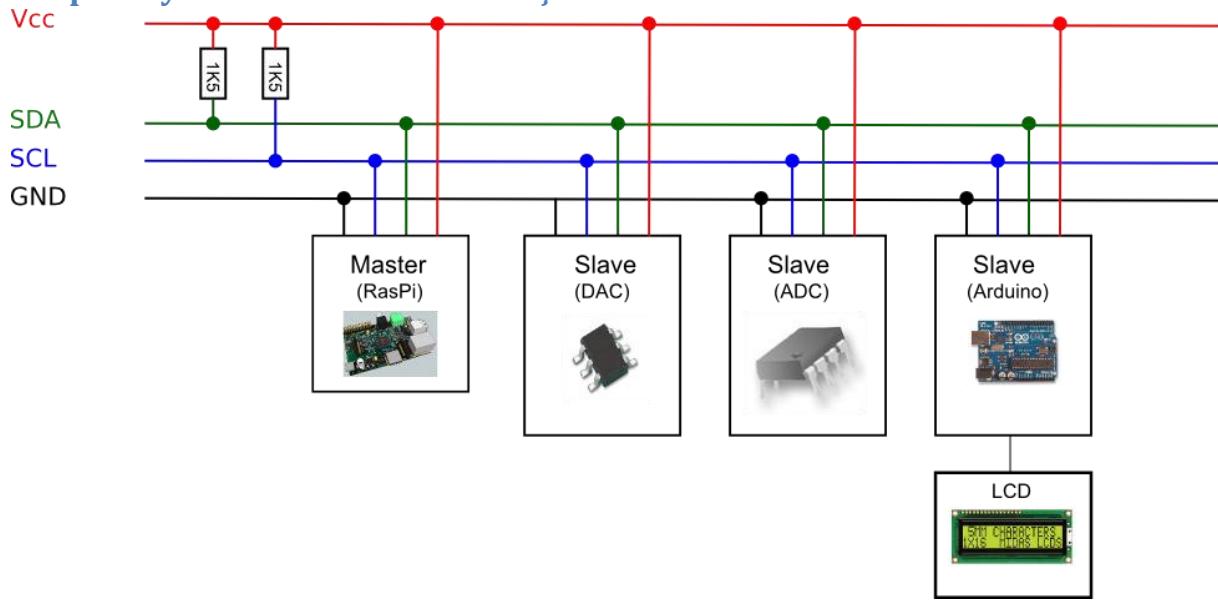
1. Öncelikle mikrodenetleyici SDA hattından "Start" biti gönderir. Bu durumda tüm slave IC'ler kendilerini alınacak adres bilgisi için hazırlarlar.
2. Mikrodenetleyici, haberleşmek istediği slave IC'nin adresini ve yapılacak işlemin yazma veya okuma olup olmadığı 1 bayt halinde SDA hattından IC veriyoluna aktarır. Slave IC'ler SDA hattından gelen adres bilgisini alırlar ve bunun kendi adresleri olup olmadığına karar verirler.
3. Masterin gönderdiği adrese sahip IC kendisinin veri yolunda olduğunu ve haberleşmeye hazır olduğunu bildirmek üzere SDA yoluna bir ACK sinyali gönderir. Diğer IC'ler mikrodenetleyiciden gelecek "Stop" bitini beklemeye koyulurlar ki , bu haberleşme tamamlanana kadar müdahalede bulunmasınlar ve bir sonraki haberleşmede kendileri çağrılabılırler.
4. Veri transferi gerçekleşir. İşlem yazma ise master'dan slave'e doğru, okuma ise tersi yönde veri transferi gerçekleşir. Transfer tamamlandıktan sonra Master(yani örneğimizde mikrodenetleyici) Stop bitini SDA hattından aktarır ve haberleşme sona erer.

Aşağıdaki şekilde elektronik olarak nasıl bilgi işlendiği görülmektedir;



Resim 25

Raspberry Pi ile Arduino I2C İletişim

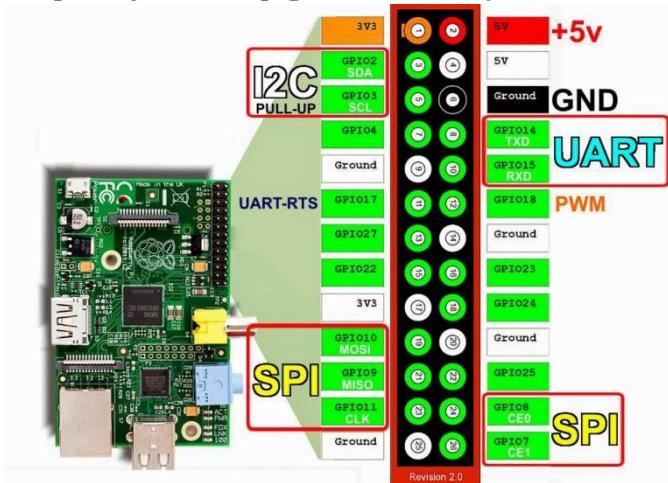


Resim 26

Yukarıdaki şekilde Raspberry ile I2C iletişimini kullanan elemanların bağlantısı görülmektedir. Görüldüğü gibi aynı hatta birden fazla cihaz iletişim sağlanabilmektedir.

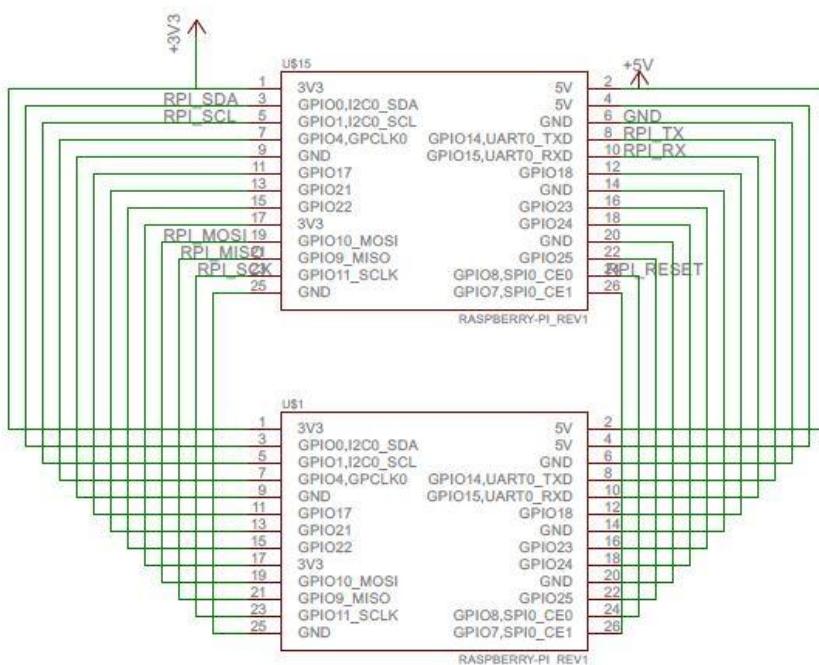
SDA bilgi giriş çıkış işlemleri SCL ucundaki saat sinyali sayesinde senkronize bir şekilde gerçekleşmektedir. Buradan anlaşıldığı gibi I2C protokolünde sadece 2 hat üzerinden yapılmaktadır. Bu protokolde entegreler master yada slave olabilirler. Her bir slave cihazın kendine özgü bir adresi vardır. Bu adres 7 veya 10 bit olabilir. Bazı cihazların adresleri fabrika çıkışında belirlenip sabittir. Bazı cihazların ise adresi kullanıcı tarafından cihazın belli uçları lojik-1 lojik-0 yapılarak belirlenir. I2C data yolu 8 bittir.

Raspberry Pi ve Gopigo I2C Haberleşme Pinleri



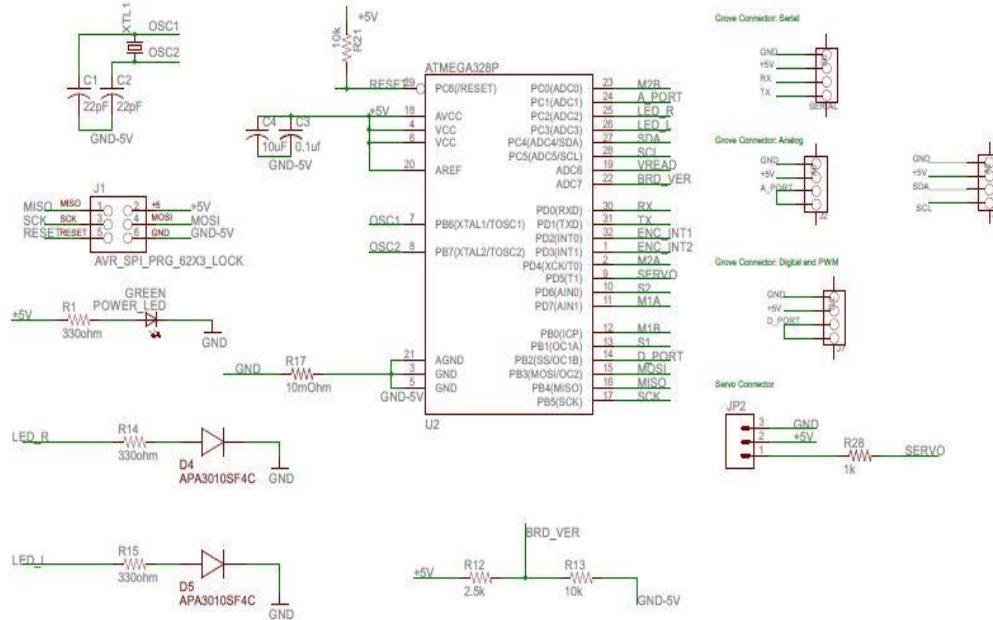
Raspberry Pi donanımsal olarak incelendiğinde pinlerinin GPIO2 SDA VE GPIO3 SCL olduğu görülmektedir. Bizim projemizde master konumundaki cihaz Raspberry Pi olup slave konumundaki cihaz ise GoPiGo'dur. Haberleşmenin gerçekleşmesi için SDA pinlerinin tek hatta aynı şekilde SCL pinleri tek bir hatta toplanması gereklidir.

Raspberry Pi Connector



Raspberry Pi'nin pinlerinin bir diğer gösterimi şekildeki gibidir.

GoPiGo I2C İletişiminin Donanımsal İncelenmesi



Şekilde GoPiGo'nun içerisinde bulunan PC4(ADC4/SDA) ve PC5(ADC5/SCL) pinlerinin i2c iletişimini sağladığını görmekteyiz.

Yazılımsal Olarak I2C Haberleşme

“fw_ver_16.ino” dosyası GoPiGo'nun yazılımını içeren dosyadır. Bu kod aynı zamanda i2c haberleşmesini gerektiren fonksiyonların bulunduğu yazılımdır. Bu kodu inceleyeceğimizde;

```
#include <avr/power.h>
#include <Wire.h> // (I2C protokolünü gerçekleştirmek için) For handling I2C
#include "SoftwareServo.h" // (Servo.h kullanma çünkü zamanlayıcıları devre dışı bırakır) Not using Servo.h because it disables timers
#include <EEPROM.h>
/* EEPROM belleğe bayt bazında erişimin yapıldığı bir hafıza tipidir.
 * Bu hafıza tipi sayesinde arduinomuzla çalışırken elde ettiğimiz verileri kaydedip
 * arduinom üzerinde güç kesilmesi durumlarında bile bu veriler silinip gitmemektedir.
Fakat belli bir veri tutma kapasitesi bulunmaktadır.
EEPROM üzerinde 0-1023 arasında bir adres bloğu bulunmaktadır */
```

Yukarıda tanımlanan kütüphanelere bakacak olursak Wire.h adında bir kütüphane bulunmaktadır. Bu kütüphanenin tanımlanma amacı I2C protokollerini gerçekleştirmektir.

```

//Raspberry Pi'den alınan komuta listesi//Command list received from the Raspberry Pi
/*PID(proportional,Integral,Derivative) oransal-integral-türevsel denetleyici PID
kontrol döngüsü yemeti , yaygın endüstriyel kontrol sistemlerinde kullanılan genel
bir kontrol döngüsü geribildirim mekanizmasıdır. */
#define fwd_cmd 119 //PID ile ileri git //Move forward with PID
#define motor_fwd_cmd 105 //PID olmadan ilerle//Move forward without PID
#define bwd_cmd 115 //PID ile geri git//Move back with PID
#define motor_bwd_cmd 107 //PID olmadan geri git//Move back without PID
#define left_cmd 97 //Bir moturu kapatarak sola dönün//Turn Left by turning off one motor
#define left_rot_cmd 98 //Her iki moturu da ters yönde çalışmasıyla sola dönün//Rotate left by running both motors in opposite direction
#define right_cmd 100 //Bir moturu kapatarak sağa dönüs yapın//Turn Right by turning off one motor
#define right_rot_cmd 110 //Her iki moturu da ters yönde çalışmasıyla sağa çevir//Rotate Right by running both motors in opposite direction
#define stop_cmd 120 //GoPiGo yu durdur//Stop the GoPiGo
#define inc_speed_cmd 116 //Hızı 10 artırmı//Increase the speed by 10
#define dec_speed_cmd 103 //Hızı 10 azaltın//Decrease the speed by 10
#define m1_cmd 111 //motor1 kontrol//Control motor1
#define m2_cmd 112 //motor2 kontrol//Control motor2
#define read_motor_speed_cmd 114 // Motor hızlarını okuyun/Read the motor speeds
#define volt_cmd 118 //Pillerin voltajını okuyun//Read the voltage of the batteries
#define us_cmd 117 //Ultrasonik sensörden mesafeyi okuyun//Read the distance from the ultrasonic sensor
#define led_cmd 108 //led yak sonucu//Turn On/Off the LED's
#define servo_cmd 101 //Servo yönünü belirle//Rotate the servo
#define enc_tgt_cmd 50 //Kodlayıcıyı hedeflemeyi ayarlama//Set the encoder targeting
#define fw_ver_cmd 20 //Ürün yazılımı sürümünü okuyun//Read the firmware version
#define en_enc_cmd 51 //Kodlayıcıları etkinleştir//Enable the encoders
#define dis_enc_cmd 52 //Kodlayıcıları devre dışı bırak//Disable the encoders
#define en_servo_cmd 61 //Servoyu etkinleştir//Enable the servo's
#define dis_servo_cmd 60 //Servoların devre dışı bırakılması//Disable the servo's
#define set_r_speed_cmd 71 //Sağ motorun hızını ayarlayın//Set the speed of the right motor
#define set_l_speed_cmd 70 //Sol motorun hızını ayarlayın//Set the speed of the left motor
#define enc_test 10 //Enkoder İmalat testi//Encoder Manufacturing test (not to be used by users)
#define serial_test 11 //Seri Port Üretim Testi//Serial Port Manufacturing test (not to be used by users)

#define enc_read 53 //Read the encoder value for motors

#define trim_test 30 //Test the trim value for a motor
#define trim_write 31 //Write the trim value to eeprom for motor m2
#define trim_read 32 //Read the trim value from eeprom

#define en_com_timeout_cmd 80 //İletişim zaman aşımını etkinleştir//Enable communication timeout
#define dis_com_timeout_cmd 81 //İletişim zaman aşımını devre dışı bırak//Disable communication timeout
#define timeout_status_cmd 82 //Zaman aşımı durumunu oku//Read the timeout status

#define digital_write 12 //Port to digital write yap//Digital write on a port
#define digital_read 13 //Portta digital read yap//Digital read on a port
#define analog_read 14 //Portta Analog read yap//Analog read on a port
#define analog_write 15 //Portta Analog read yap//Analog write on a port
#define pin_mode 16 //Set up the pin mode on a port

#define ep_sig1 0xDE
#define ep_sig2 0x14

#define ir_init_cmd 22//IR kumanda kontrolünü initialize(başlat) yap
#define ir_read_cmd 21//IR kumanda oku
#define cpu_speed_cmd 25//CPU hızı

```

Yukarıdaki tanımlanan komutlar Raspberry Pi ' den alınan komut listesidir. Burada Raspberry Pi'nin isteği doğrultusunda GoPiGo'da bir takım işlemler yaptırılır.

Aşağıdaki fonksiyon I2C üzerinden gelen komutları almak için tanımlanmıştır. Kod parçacığını inceleyecek olursak;

```

//Receive commands via I2C// I2C üzerinden komutlar al
void receiveData(int byteCount)
{
    while(Wire.available())
    {
        //When the buffer gets filled up with 4 bytes( the commands have to be 4 bytes in size), set the index back to 0 to read the next command
        // Arabellek 4 bayt ile dolduğunda, dizini bir sonraki komutu okumak için 0'a getirin
        if(Wire.available()==4)
        {
            index=0;
            if(timeout_f) //Refresh the communication time-out value// İletişim zaman aşımı değerini yenile
                last_t=millis();
        }
        cmd[index++]=Wire.read(); //Load the command into the buffer// Komutu arabelleğe yükle
    }

    // callback for sending data// veri göndermek için geri arama
    volatile int ind=0;
}

```

Aşağıdaki void sendData() fonksiyonu ise I2C üzerinden komut göndermek için tanımlanmıştır. Bu fonksiyonu inceleyecek olursak;

```

void sendData()
{
    //if(timeout_>)
    //    last_t=millis();
    //Sends 2 bytes back for the Ultrasonic read, voltage and firmware version
    if(bytes_to_send==2)
    {
        Wire.write(payload[ind++]);
        if(ind==2)
        {
            ind=0;
            bytes_to_send=0;
        }
    }
    else if(bytes_to_send==20)
    {
        Wire.write(payload,20);
        bytes_to_send=0;
    }
    // else if(bytes_to_send==21)
    // {
    //     Wire.write(payload[ind++]);
    //     if(ind==21)
    //     {
    //         ind=0;
    //         bytes_to_send=0;
    //         payload[0]=0;

    else// Send back the status flag
    {
        Wire.write(status_r);
    }
}

```

“gopigo.py” Dosyasının İncelenmesi

I2C haberleşmesinde Raspberry Pi tarafında gerçekleştiren dosyadır. Bu dosyada GoPiGo' ya gönderilmek istenen komut listesi bulunmaktadır.

```

import sys
import time
import math
import struct
import subprocess

WHEEL_RAD=3.25
WHEEL_CIRC=2*math.pi*WHEEL_RAD
PPR = 18 # encoder Pulses Per Revolution

if sys.platform == 'uwp':
    import winrt_smbus as smbus
    bus = smbus.SMBus(1)
else:
    import RPi.GPIO as GPIO
    import smbus

    # for RPI version 1, use "bus = smbus.SMBus(0)"
    rev = GPIO.RPI_REVISION
    if rev == 2 or rev == 3:
        bus = smbus.SMBus(1)
    else:
        bus = smbus.SMBus(0)

```

Yukarıdaki kod parçacığında tanımlanan “[smbus](#)” kütüphanesi I2C iletişimini gerçekleştiren kütüphanedir.

```

#GoPiGo Commands
fwd_cmd      =[119]      #Move forward with PID
motor_fwd_cmd =[105]      #Move forward without PID
bwd_cmd       =[115]      #Move back with PID
motor_bwd_cmd =[107]      #Move back without PID
left_cmd      =[97]       #Turn Left by turning off one motor
left_rot_cmd  =[98]       #Rotate left by running both motors in opposite direction
right_cmd     =[100]      #Turn Right by turning off one motor
right_rot_cmd =[110]      #Rotate Right by running both motors in opposite direction
stop_cmd      =[120]      #Stop the GoPiGo
ispd_cmd      =[116]      #Increase the speed by 10
dsdp_cmd      =[103]      #Decrease the speed by 10
m1_cmd        =[111]      #Control motor1
m2_cmd        =[112]      #Control motor2
read_motor_speed_cmd=[114] #Get motor speed back

volt_cmd      =[118]      #Read the voltage of the batteries
us_cmd        =[117]      #Read the distance from the ultrasonic sensor
led_cmd       =[108]      #Turn On/Off the LED's
servo_cmd    =[101]      #Rotate the servo
enc_tgt_cmd   =[50]       #Set the encoder targeting
fw_ver_cmd   =[20]       #Read the firmware version
en_enc_cmd   =[51]       #Enable the encoders
dis_enc_cmd  =[52]       #Disable the encoders
read_enc_status_cmd =[53]
en_servo_cmd =[61]       #Enable the servo's
dis_servo_cmd=[60]       #Disable the servo's
set_left_speed_cmd =[70]  #Set the speed of the right motor
set_right_speed_cmd =[71] #Set the speed of the left motor
en_com_timeout_cmd =[80]  #Enable communication timeout
dis_com_timeout_cmd =[81] #Disable communication timeout
timeout_status_cmd =[82] #Read the timeout status
enc_read_cmd  =[53]      #Read encoder values
trim_test_cmd =[30]      #Test the trim values
trim_write_cmd=[31]      #Write the trim values
trim_read_cmd =[32]

digital_write_cmd =[12]   #Digital write on a port
digital_read_cmd =[13]   #Digital read on a port
analog_read_cmd =[14]   #Analog read on a port
analog_write_cmd =[15]   #Analog read on a port
pin_mode_cmd   =[16]   #Set up the pin mode on a port

ir_read_cmd   =[21]
ir_recv_pin_cmd =[22]
cpu_speed_cmd =[25]

```

Yukarıdaki kod parçası GoPiGo için gerekli komut listesini içermektedir. Bu komutlar i2c ile istek doğrultusunda kullanılmak içindir.

Aşağıdaki kod parçacıklarını inceleyeceğiz olursak;

PinMode

1.Kod

```

# Setting Up Pin mode on Arduino
def pinMode(pin, mode):
    # if pin ==10 or pin ==15 or pin ==0 or pin ==1:
    if mode == "OUTPUT":
        write_i2c_block(address, pin_mode_cmd + [pin, 1, unused])
    elif mode == "INPUT":
        write_i2c_block(address, pin_mode_cmd + [pin, 0, unused])
    #time.sleep(.005)  #Wait for 5 ms for the commands to complete
    return 1
    # else:
    #     # return -2

```

2. Kod

```

//Pin Mode
//If "pin mode" received via I2C, set the pin specified in cmd[1] to pin state specified in cmd[2]
//  State: 0 - Input
//  State: 1 - Output
//e.g.:cmd()=[16,7,1]
//          ^-- pin_mode
//          ^-- pin ?
//          ^-- Output
if(cmd[0]==pin_mode)
{
    if(cmd[2]==0)
        pinMode(cmd[1], INPUT);
    else if(cmd[2]==1)
        pinMode(cmd[1], OUTPUT);
    cmd[0]=0;
}

```

1.Kod parçası gopigo.py den bir kısım ve 2.Kod parçası ise fw_ver_16.ino dosyasından bir kısımıdır. Bu iki kodu karşılaştırıldığımızda i2c haberleşmesi gerçekleştiğinde pinMode

komutunun 3 değer aldığı ve bu değerleri ise cmd[] dizisinde tuttuğumuzu görmekteyiz. cmd[0] komutun ismini tutarken cmd[1] pin numarasını cmd[2] ise input veya output değerini tutmaktadır.

Analog Read

Kod 1

```
# Read analog value from Pin
def analogRead(pin):
    #if pin == 1 :
    write_i2c_block(address, analog_read_cmd + [pin, unused, unused])
    time.sleep(.007)
    try:
        b1=bus.read_byte(address)
        b2=bus.read_byte(address)
    except IOError:
        return -1
    return b1* 256 + b2
#else:
#    return -2
```

Kod2

```
#Read board revision
#   return: voltage in V
def brd_rev():
    write_i2c_block(address, analog_read_cmd + [7, unused, unused])
    time.sleep(.1)
    try:
        b1=bus.read_byte(address)
        b2=bus.read_byte(address)
    except IOError:
        return -1
    return b1* 256 + b2
```

Kod3

```
//Analog Read
//If "analog read" received via I2C, read the pin specified in cmd[1] and save the data in payload
//e.g.:cmd[]={14,7}
//           14- analog_read
//           7- pin 7
```

```
//Analog Read I2C ile alınıyorsa cmd1 deki pini oku ve kaydet
if(cmd[0]==analog_read)
{
    temp=analogRead(cmd[1]);
    payload[0]=temp/256;
    payload[1]=temp%256;
    bytes_to_send=2; //Enable this to send bytes back */Byte i geri göndermek için etkinleştirme
    cmd[0]=0;
    cmd[1]=0;
}
```

Kod 1,Kod 2 ve Kod3 kod parçacıklarını incelersek python kodunda(kod1 ve kod2) adresle beraber gönderilen komut gopigo tarafında(kod3) tarafında bu değerler nasıl kullanıldığı gösterilmiştir.GoPiGo 8 bitlik değer okuma yaptığı için payload dizisine bboldük.

Stop GoPiGo

Kod 1

```

#Stop the GoPiGo
def stop():
    return write_i2c_block(address,stop_cmd+[0,0,0])

```

Kod 2

```

else if(cmd[0]==stop_cmd)
    stp();

//Set motor1 movement//motor1 momentini ayarla

```

Stop komutu için haberleşmeyi incelemek gerekirse 3 parametrenin olduğu görülmektedir. Gopigo tarafında stp() fonksiyonu çağrılarak durdurma işlemi gerçekleşir.

Motor 1 Kontrol

Kod 1

```

#Control Motor 1
def motor1(direction,speed):
    return write_i2c_block(address,m1_cmd+[direction,speed,0])

```

Kod 2

```

//Set motor1 movement//motor1 momentini ayarla
else if(cmd[0]==m1_cmd)
    motor1_control(cmd[1],cmd[2]);

```

Motor1 için Raspberry Pi'den gelen direction ve speed değerleri Gopigo motor1 kontrolü için kullanılmıştır.

Motor 2 Kontrol

Kod 1

```

#Control Motor 2
def motor2(direction,speed):
    return write_i2c_block(address,m2_cmd+[direction,speed,0])

```

Kod 2

```

else if(cmd[0]==m2_cmd)
    motor2_control(cmd[1],cmd[2]);

//Read voltage and load the result into the I2C buffer// Voltajı okuyun ve sonucu I2C tamponuna yükleyin

```

Motor2 için Raspberry Pi'den gelen direction ve speed değerleri motor1 de olduğu gibi Gopigo motor2 kontrolü için kullanılmıştır.

GoPiGo Forward

Kod 1

```

#Move the GoPiGo forward
def fwd(dist=0): #distance is in cm
    try:
        if dist>0:
            # this casting to int doesn't seem necessary
            pulse=int(PPR*(dist//WHEEL_CIRC) )
            enc_tgt(1,1,pulse)
    except:
        pass
    return write_i2c_block(address,motor_fwd_cmd+[0,0,0])

# support more explicit spelling for forward function
forward=fwd

#Move the GoPiGo forward without PID
def motor_fwd():
    return write_i2c_block(address,motor_fwd_cmd+[0,0,0])

```

Kod 2

```
//Set the motor controller to start moving forward. The speed is either constant with "nopid" or changing with "pid enabled"
// Motor kontrolunu ileri hareket etmeye başlayacak şekilde ayarlayın.
if(cmd[0]==fwd_cmd || cmd[0]==motor_fwd_cmd)
    forward();
```

Yukarıdaki kodlar Gopigo'nun ileri gitmesi için haberleşmenin nasıl gerçekleştiğini göstermektedir.

Move Gopigo Back

Kod 1

```
#Move GoPiGo back
def bwd(dist=0):
    try:
        if dist>0:
            # this casting to int doesn't seem necessary
            pulse=int(PPR*(dist//WHEEL_CIRC) )
            enc_tgt(1,1,pulse)
    except:
        pass
    return write_i2c_block(address,motor_bwd_cmd+[0,0,0])

# support more explicit spelling for backward function
backward=bwd

#Move GoPiGo back without PID control
def motor_bwd():
    return write_i2c_block(address,motor_bwd_cmd+[0,0,0])
```

Kod 2

```
//Set the motor controller to start moving backward. The speed is either constant wit "nopid" or changing with "pid enabled"
// Motor kontrolunu geriye doğru hareket etmeye başlayacak şekilde ayarlayın
else if(cmd[0]==bwd_cmd || cmd[0]==motor_bwd_cmd)
    backward();
```

Yukarıda gösterilen kod parçacıkları Gopigo'nun geri gitmesi için nasıl haberleşmesi gerektiğini göstermektedir.

Turn Gopigo Left

Kod 1

```
#Turn GoPiGo Left slow (one motor off, better control)
def left():
    return write_i2c_block(address,left_cmd+[0,0,0])
```

Kod 2

```
//Set the motor controller to turn left// Motor kontrolunu sola dönecek şekilde ayarlayın
else if(cmd[0]==left_cmd)
    left();
```

Kod1 ve Kod2 kod parçacıklarında sola dönme olayının gerçekleşmesi için kod1 den gelen verilerin kod2 de nasıl kullanıldığı görülmektedir.

Turn Gopigo Right

Kod 1

```
#Turn GoPiGo right slow (one motor off, better control)
def right():
    return write_i2c_block(address,right_cmd+[0,0,0])
```

Kod 2

```
//Set the motor controller to turn right// Motor kontrolunu sağa dönecek şekilde ayarlayın
else if(cmd[0]==right_cmd)
    right();
```

Kod1 ve Kod2 kod parçacıklarında sağa dönme olayının gerçekleşmesi için kod1 den gelen verilerin kod2 de nasıl kullanıldığı görülmektedir.

Rotate Gopigo Right

Kod 1

```
#Rotate GoPiGo right in same position both motors moving in the opposite direction)
def right_rot():
    return write_i2c_block(address,right_rot_cmd+[0,0,0])
```

Kod 2

```
//Set the motor controller to rotate right// Motor kontrol cihazını sağa döndürmek için ayarlayın
else if(cmd[0]==right_rot_cmd)
    right_rot();
```

Kod parçalarında aynı şekilde kod1 den gelen veri değerleri kod2 için kullanılmıştır.

Read Voltage

Kod 1

```
#Read voltage
#    return: voltage in V
def volt():
    write_i2c_block(address,volt_cmd+[0,0,0])
    time.sleep(.1)
    try:
        b1=bus.read_byte(address)
        b2=bus.read_byte(address)
    except IOError:
        return -1

    if b1!=-1 and b2!=-1:
        v=b1*256+b2
        v=(5*float(v)/1024)/0.4
        return round(v,2)
    else:
        return -1
```

Kod 2

```
//Read voltage and load the result into the I2C buffer// Voltajı okuyun ve sonucu I2C tamponuna yükleyin
else if(cmd[0]==volt_cmd)
{
    //MAKE COMPATIBLE WITH OLD FIRMWARE
    // volt=analogRead(0);
    volt=analogRead(voltage_pin);//voltaj pinini oku değeri degiskene ata
    payload[0]=volt/256;//olculen değeri volta donusturuyoruz
    payload[1]=volt%256;//olculen değer 256 ya tam bolunmuyorsa kalanı buraya kaydediyoruz
    bytes_to_send=2;
    cmd[0]=0;
}
```

Voltaj değerinin okunması için kod1 ile Gopigo'ya gönderilen veriler kullanımı kod2 de olduğu gibi kullanılır.

Read Voltage

Kod 1

```

#Read voltage
#    return: voltage in V
def volt():
    write_i2c_block(address,volt_cmd+[0,0,0])
    time.sleep(.1)
    try:
        b1=bus.read_byte(address)
        b2=bus.read_byte(address)
    except IOError:
        return -1

    if b1!=-1 and b2!=-1:
        v=b1*256+b2
        v=(5*float(v)/1024)/0.4
        return round(v,2)
    else:
        return -1

```

Kod 2

```

//Read voltage and load the result into the I2C buffer// Voltajı okuyun ve sonucu I2C tamponuna yükleyin
else if(cmd[0]==volt_cmd)
{
    //MAKE COMPATIBLE WITH OLD FIRMWARE
    // volt=analogRead(0);
    volt=analogRead(voltage_pin);//voltaj pinini oku degeri degiskene ata
    payload[0]=volt/256;//olculen degeri volta donusturuyoruz
    payload[1]=volt%256;//olculen deger 256 ya tam bolunmuyorsa kalani buraya kaydediyoruz
    bytes_to_send=2;
    cmd[0]=0;
}

```

Yukarıdaki kodlarda voltaj değeri okunur ve sonuç i2c tamponuna yüklenir.

Read Ultrasonic Sensor

```

#Read ultrasonic sensor
#    arg:
#        pin -> Pin number on which the US sensor is connected
#    return:    distance in cm
def us_dist(pin):
    write_i2c_block(address,us_cmd+[pin,0,0])
    time.sleep(.08)
    try:
        b1=bus.read_byte(address)
        b2=bus.read_byte(address)
    except IOError:
        return -1
    if b1!=-1 and b2!=-1:
        v=b1*256+b2
        return v
    else:
        return -1

```

Kod 2

```

//Read ultrasonic sensor and load the result into I2C buffer // Ultrasonik sensörü okuyun ve sonucu I2C tamponuna yükleyin
else if(cmd[0]==us_cmd)
{
    int _pin=cmd[1];//pin degerini cmd[1] den alıp degiskene atadık
    if(debug)//hata ayıklama
    {
        Serial.print("US ");
        Serial.print(_pin);
    }
    pinMode(_pin, OUTPUT);//pini output olarak tanımladık
    digitalWrite(_pin, LOW);
    delayMicroseconds(2);//2 mikrosaniye bekle
    digitalWrite(_pin, HIGH);
    delayMicroseconds(5);
    digitalWrite(_pin, LOW);
    pinMode(_pin, INPUT);//pini input olarak tanımladık
    long dur = pulseIn(_pin,HIGH);//pulseIn frekans okumaya yarar ve frekansın uzunluğunu ms olarak döndürür.
    int RangeCm = dur/29/2;//mesafe hesabı yapıldı

    payload[0]=RangeCm/256;//mesafe
    payload[1]=RangeCm%256;
    bytes_to_send=2;

    if(debug)
    {
        Serial.print(RangeCm);
        Serial.print(" ");
        Serial.print(payload[0]);
        Serial.print(" ");
        Serial.println(payload[1]);
    }
    cmd[0]=0;
}

```

Yukarıdaki kod parçalarında ise ultrasonic sensor okuma ve sonucun i2c tamponuna nasıl yükleniği gösterilir.

Set Servo Position

Kod 1

```

#Set servo position
#  arg:
#      position: angle in degrees to set the servo at
def servo(position):
    write_i2c_block(address,servo_cmd+[position,0,0])
#time.sleep(.05)

```

Kod 2

```

//Set the servo angle // Servo açısını ayarla
else if(cmd[0]==servo_cmd)
{
    value=cmd[1];
    if(debug)
    {
        Serial.println(value);
    }
    servo1.write(value);
    time_since_servo_cmd=millis();
    servo_flag=1;           //Set the servo flag so that it is refreshed every time in the main loop().
                           // Servo bayrağını, ana döngüde her seferinde yenilenmesini sağlayacak şekilde ayarlayın().
    cmd[0]=0;
}

```

Kod parçacıklarında servo açısının ayarlanması için kod1 den gelen verilerin kod2 de nasıl kullanıldığı görülmektedir.

Encoder Targeting

Kod 1

```
#Set encoder targeting on
#arg:
#    m1: 0 to disable targeting for m1, 1 to enable it
#    m2: 1 to disable targeting for m2, 1 to enable it
#    target: number of encoder pulses to target (18 per revolution)
def enc_tgt(m1,m2,target):
    # print("enc_tgt m1 {} m2 {} target {}".format(m1,m2,target))
    if m1>1 or m1<0 or m2>1 or m2<0:
        return -1
    m_sel=m1*2+m2
    write_i2c_block(address,enc_tgt_cmd+[m_sel,target//256,target%256])
    return 1
```

Kod 2

```
//Encoder targeting function// Encoder hedefleme fonksiyonu
//Target unit in encoder counts. 18 counts per rotation
if(cmd[0]==enc_tgt_cmd || tgt_flag==1)
{
    //If the command to set the encoder targeting was just received// Kodlayıcı hedeflemeyi ayarlama komutu yeni alındıysa
    if(cmd[0]==enc_tgt_cmd)
    {
        cmd[0]=0;                      //Command has been read// Komutu oku
        tgt_flag=1;                    //Enable the targeting flag// Hedefleme bayrağını etkinleştir
        m1_en=cmd[1]/2;                //Check if target on motor 1 set// motor1 setinde hedef olup olmadığını kontrol edin
        m2_en=cmd[1]%2;                //Check if target on motor 2 set// motor2 setinde hedef olup olmadığını kontrol edin
        tgt=cmd[2]*256+cmd[3];         //Set the target//hedefi ayarla
        e1=e2=0;                      //Set position of both the encoders to 0// Her iki enkoderin konumunu 0 olarak ayarlayın
    }
}
```

Encoder hedeflemede kod1 de m_sel değeri hesaplanıp i2c ile gönderilir kod2 de bu değer kullanılır.

Read Encoder

Kod 1

```
#Read encoder value
#    arg:
#        motor -> 0 for motor1 and 1 for motor2
#    return: distance in cm
def enc_read(motor):
    write_i2c_block(address,enc_read_cmd+[motor,0,0])
    time.sleep(.08)
    try:
        b1=bus.read_byte(address)
        b2=bus.read_byte(address)
    except IOError:
        return -1
    if b1!=-1 and b2!=-1:
        v=b1*256+b2
        return v
    else:
        return -1
```

Kod 2

```

//Encoder read//kodlayıcıyı oku
//If "enc_read" received via I2C, read the encoder value for motor in cmd[1] for and send the data in payload
//e.g.: cmd[]={1}
//      0- motor1
//      1- motor2
if(cmd[0]==enc_read)
{
    if(cmd[1]==0)//motor1 için
        temp=e1;
    else//motor2 için
        temp=e2;
    payload[0]=temp/256;
    payload[1]=temp%256;
    bytes_to_send=2; //Enable this to send bytes back// Bayt geri göndermek için bunu etkinleştirin
    cmd[0]=0;
}

```

Encoder okumada motor1 veya motor2 bilgisi Gopigo'ya verilir. Kod 2 de görüldüğü gibi kullanımı gerçekleşmiştir.

Firmware Version

Kod 1

```

#Returns the firmware version
def fw_ver():
    write_i2c_block(address, fw_ver_cmd+[0,0,0])
    time.sleep(.1)
    try:
        ver=bus.read_byte(address)
        bus.read_byte(address) #Empty the buffer
    except IOError:
        return -1
    return float(ver)/10

```

Kod 2

```

//Load the firmware version in payload[] buffer to be sent back// Gönderilecek firmware sürümünü payload[] buffer'a yükleyin
else if(cmd[0]==fw_ver_cmd)
{
    payload[0]=ver;
    bytes_to_send=2; //Enable this to send bytes back// Baytları geri göndermek için bunu etkinleştirin
    cmd[0]=0;
}

```

Firmare versiyonu için haberleşme yukarıdaki gibidir.

Enable Encoders

Kod 1

```

#Enable the encoders (enabled by default)
def enable_encoders():
    return write_i2c_block(address, en_enc_cmd+[0,0,0])

```

Kod 2

```

//If encoders are to be enabled, attach the interrupts to the pins// Enkoderler etkinleştirileceklese, kesmeleri pinlere ata
else if(cmd[0]==en_enc_cmd)//kesmeleri etkinleştir
{
    attachInterrupt(0, step1, RISING);
    attachInterrupt(1, step2, RISING);
    cmd[0]=0;
}
//To disable encoder interrupts, detach the interzupts
else if(cmd[0]==dis_enc_cmd)//kesmeleri etkisiz bizek
{
    detachInterrupt(1);
    detachInterrupt(0);
    cmd[0]=0;
}

```

Encoderları etkinleştirme için Raspberry Pi ve Gopigo haberleşmesi yukarıdaki gibidir.

Set Left Speed

Kod 1

```

#Set speed of the left motor
#    arg:
#        speed-> 0-255
def set_left_speed(speed):
    if speed >255:
        speed =255
    elif speed <0:
        speed =0
    return write_i2c_block(address,set_left_speed_cmd+[speed,0,0])

```

Kod 2

```

//Set speed of left motor
// Sol motorun hızını ayarla
else if(cmd[0]==set_l_speed_cmd)
{
    speed2=cmd[1];
    cmd[0]=0;
}

```

Sol motorun hızını kod1'den gelen hız bilgisiyle kod2'deki hız değeri belirlenir.

Set Right Speed

Kod 1

```

#Set speed of the right motor
#    arg:
#        speed-> 0-255
def set_right_speed(speed):
    if speed >255:
        speed =255
    elif speed <0:
        speed =0
    return write_i2c_block(address,set_right_speed_cmd+[speed,0,0])

```

Kod2

```

//Set speed of right motor
// Sağ motorun hızını ayarla
if(cmd[0]==set_r_speed_cmd)
{
    speed1=cmd[1];
    cmd[0]=0;
}

```

Sağ motorun hızını kod1'den gelen hız bilgisiyle kod2'deki hız değeri belirlenir.

Enable Com Timeout

Kod 1

```
def enable_com_timeout(timeout):
    return write_i2c_block(address,en_com_timeout_cmd+[timeout//256,timeout%256,0])
```

Kod 2

```
//Set the communication timeout// İletişim zaman aşımını ayarla
else if(cmd[0]==en_com_timeout_cmd)
{
    timeout=cmd[1]*256+cmd[2];
    cmd[0]=0;
    last_t=millis();
    timeout_f=1;
}
```

İletişim zaman aşımını ayarlamak için i2c iletişimini yukarıdaki şekilde gerçekleştirir.

Disable Com Timeout

Kod 1

```
#Disable communication time-out
def disable_com_timeout():
    return write_i2c_block(address,dis_com_timeout_cmd+[0,0,0])
```

Kod 2

```
//Disable the communication timeout// İletişim zaman aşımını devre dışı bırak
else if(cmd[0]==dis_com_timeout_cmd)
{
    cmd[0]=0;
    last_t=millis();
    timeout_f=0;
    tgt_flag=0;
}
```

İletişim zaman aşımını devre dışı bırakmak için i2c iletişimini yukarıdaki şekilde gerçekleştirir.

Cpu Speed

Kod 1

```
def cpu_speed():
    write_i2c_block(address,cpu_speed_cmd+[0,0,0])
    time.sleep(.1)
    try:
        b1=bus.read_byte(address)
        b2=bus.read_byte(address)
    except IOError:
        return -1
    return b1
```

Kod 2

```
//Read CPU speed in Mhz// Mhz'de CPU hızını oku
else if(cmd[0]==cpu_speed_cmd)
{
    payload[0]=F_CPU/1000000;
    payload[1]=0;
    bytes_to_send=2;
    cmd[0]=0;
}
```

CPU hızını okumak için i2c iletişimini yukarıdaki şekilde gerçekleştirir.

RASPBERRY İLE ARDUINO HABERLEŞMESİ

Instal.sh

```

identify_robot() {
    #robotun başlangıcı
    echo " _____ _____ _____ _____ _____"
    echo " | ____| |__|_____| |_____| |__|"
    echo " |_____| |_____| |_____| |_____| |_____|"
    echo " "
    feedback "Welcome to GoPiGo Installer."
    echo " "
}
```

```

check_root_user() {
    if [[ $EUID -ne 0 ]]; then
        feedback "FAIL! This script must be run as such: sudo ./install.sh"
        exit 1
    fi
    echo " "
    #root kullanıcısını kontrol et
    #güncelleme kısmı
}
```

```

check_internet() {

    if ! quiet_mode ; then
        feedback "Check for internet connectivity..."
        #internet bağlantısını kontrol etme
        feedback "====="
        wget -q --tries=2 --timeout=20 --output-document=/dev/null http://raspberrypi.org
        if [ $? -eq 0 ];then
            echo "Connected to the Internet"
        else
            echo "Unable to Connect, try again !!!"

```

```

    exit 0
fi
fi
}

display_welcome_msg() {
    feedback "Please ensure internet connectivity before running this script."
    if ! quiet_mode
    then
        feedback "NOTE: Raspberry Pi will need to be rebooted after completion."
    fi
    #internet bağlantısını sağlayıp yeniden başlatır

    feedback "Special thanks to Joe Sanford at Tufts University. This script was derived from his work.
Thank you Joe!"
    echo ""
}

install_dependencies() {
    if ! quiet_mode ; then
        sudo apt-get update
    fi
    echo ""
    feedback "Installing Dependencies"
    feedback "=====
    sudo apt-get install python-pip git libi2c-dev python-serial python-rpi.gpio i2c-tools python-smbus
arduino minicom libnss-mdns python-dev -y
    sudo pip install -U RPi.GPIO

    feedback "Dependencies installed"
    #güncelleyip kurma
}

```

```
install_DHT() { #dht kütüphanesi arduino için sensör kütüphanesi
#dht kütüphanesinin kurulması

# Install the DHT library

feedback "Installing DHT library"

pushd $ROBOT_DIR/Software/Python/sensor_examples/dht/Adafruit_Python_DHT >/dev/null

sudo python setup.py install

sudo python3 setup.py install

popd > /dev/null

}
```

```
install_wiringpi() {

#WiringPi, Raspberry Pi üzerindeki GPIO pinlerini kontrol edebilmenizi sağlayan c dili ile yazılmış bir
kütüphanedir.

#WiringPi'nin Yüklü olup olmadığını kontrol edin

# Check if WiringPi Installed

# using curl piped to bash does not leave a file behind. no need to remove it

# we can do either the curl - it works just fine

# sudo curl https://raw.githubusercontent.com/DexterInd/script_tools/master/update_wiringpi.sh
| bash

# or call the version that's already on the SD card

#sd kartta bulunan sürümü ayarlar ve tekrar yüklendi mi diye kontrol eder.
```

```
sudo bash $DEXTERSCRIPT/update_wiringpi.sh

#wiringPi yüklemü diye tekrar kontrol edilir
```

```

# done with WiringPi

# remove wiringPi directory if present
if [ -d wiringPi ]
then
    sudo rm -r wiringPi
fi

# End check if WiringPi installed
echo " "
}

install_spi_i2c() {
#I2C ve SPI seri arabirimleri üzerinden bir mikroişlemciye eklenebilen birçok arabirimleri vardır.
Bunlara sensörler , EEPROM'lar ve çeşitli ekran türleri dahildir .

    feedback "Removing blacklist from /etc/modprobe.d/raspi-blacklist.conf . . ."
    feedback "====="
    if grep -q "#blacklist i2c-bcm2708" /etc/modprobe.d/raspi-blacklist.conf; then
        echo "I2C already removed from blacklist"
    else
        sudo sed -i -e 's/blacklist i2c-bcm2708/#blacklist i2c-bcm2708/g' /etc/modprobe.d/raspi-
blacklist.conf
        echo "I2C removed from blacklist"
    fi
    if grep -q "#blacklist spi-bcm2708" /etc/modprobe.d/raspi-blacklist.conf; then
        echo "SPI already removed from blacklist"
    else
        sudo sed -i -e 's/blacklist spi-bcm2708/#blacklist spi-bcm2708/g' /etc/modprobe.d/raspi-
blacklist.conf
        #raspi-blacklist.conf çekirdeğin gerekli olmayan modülleri yüklemesini önlemek için
        kullanılır.
        echo "SPI removed from blacklist"
    fi
}

```

```

fi

#i2c ve spi baclistten kaldırıldı

#Adding in /etc/modules

echo " "

feedback "Adding I2C-dev and SPI-dev in /etc/modules . . ."

feedback "====="

if grep -q "i2c-dev" /etc/modules; then

    echo "I2C-dev already there"

else

    echo i2c-dev >> /etc/modules

    echo "I2C-dev added"

fi

if grep -q "i2c-bcm2708" /etc/modules; then

    echo "i2c-bcm2708 already there"

else

    echo i2c-bcm2708 >> /etc/modules

    echo "i2c-bcm2708 added"

fi

if grep -q "spi-dev" /etc/modules; then

    echo "spi-dev already there"

else

    echo spi-dev >> /etc/modules

    echo "spi-dev added"

fi

echo " "

feedback "Making I2C changes in /boot/config.txt . . ."

feedback "====="

echo dtparam=i2c1=on >> /boot/config.txt

echo dtparam=i2c_arm=on >> /boot/config.txt

```

```

sudo adduser pi i2c
echo " "
}

#i2c ve spi eklendi ve tekrar olup olmadığı kontrol edildi

install_avr() {
    #Adding ARDUINO setup files
    echo " "
    #####
    # Remove after the image is created for BrickPi3
    #####
    # feedback "Making changes to Arduino . . ."
    # feedback "====="
    # cd /tmp
    # wget http://project-downloads.drogon.net/gertboard/avrdude_5.10-4_armhf.deb
    # sudo dpkg -i avrdude_5.10-4_armhf.deb
    # sudo chmod 4755 /usr/bin/avrdude
    # cd /tmp
    # if [ -f /tmp/setup.sh ]; then
        # rm /tmp/setup.sh
    # fi
    # wget http://project-downloads.drogon.net/gertboard/setup.sh
    # chmod +x setup.sh
    # sudo ./setup.sh
    # #Enabling serial port in Arduino IDE
    # crontab -l > file; echo '@reboot ln -sf /dev/ttAMA0 /dev/ttyS0' >> file; crontab file
    # rm file
    #####
    source /home/pi/Dexter/lib/Dexter/script_tools/install_avrdude.sh
    create_avrdude_folder
    install_avrdude
    cd $ROBOT_DIR
}

```

```

echo "done with AVRDUDE "
}

#avrdude kurulum dosyaları ekleme

install_line_follower(){

    feedback "--> Installing Line Follower Calibration"

    # Install GoPiGo Line Follower Calibration

    delete_file /home/pi/Desktop/line_follow.desktop

    sudo cp /home/pi/Dexter/GoPiGo/Software/Python/line_follower/line_follow.desktop
    /home/pi/Desktop/

    sudo chmod +x /home/pi/Desktop/line_follow.desktop

    sudo chmod +x /home/pi/Dexter/GoPiGo/Software/Python/line_follower/line_sensor_gui.py

    # sensor_gui Hat ortasındaysa, dolaşmaya devam et

    # Hat sağdan biraz sola dönüyorsa, düz hareket etmeye devam edin

    # if the configuration files exist in the home directory

    # then move them to their new place

    # otherwise create new ones

    if file_exists "$PIHOME/black_line.txt"

        then

            sudo mv $PIHOME/black_line.txt $PIHOME/Dexter/black_line.txt

            #black_line white_line Dosyalardan varsayılan değerleri yükler

        else

            sudo touch $PIHOME/Dexter/black_line.txt

        fi

    if file_exists "$PIHOME/white_line.txt"

        then

            sudo mv $PIHOME/white_line.txt $PIHOME/Dexter/white_line.txt

        else

            sudo touch $PIHOME/Dexter/white_line.txt

```

```

fi

if file_exists "$PIHOME/range_line.txt"
then
    sudo mv $PIHOME/range_line.txt $PIHOME/Dexter/range_line.txt
        # range_line Eşik değerine bağlı olarak ham değerleri mutlak 0 ve 1'e dönüştürür
else
    sudo touch $PIHOME/Dexter/range_line.txt
fi

sudo chmod 666 $PIHOME/Dexter/*line.txt

}

# Yapılandırma dosyaları ev dizini içinde bulunuyorsa
# Sonra onları yeni yerine taşıyın
#Aksi takdirde yenilerini oluştur

install_control_panel(){

    sudo cp "$ROBOT_DIR/Software/Python/control_panel/gopigo_control_panel.desktop"
$PIHOME/Desktop
}

call_for_reboot() {
    if ! quiet_mode ; then
        feedback " "
        feedback "Please restart the Raspberry Pi for the changes to take effect"
        feedback " "
        feedback "Please restart to implement changes!"
        feedback " _____ "
        feedback " | \ / | / \ | \ / | \ / | \ / | \ / | \ / | "
        feedback " | | \ ) | | _ | ( _ | | / \ | | _ | | | "
        feedback " | _ / | _ | \ | | / \ | _ | / | | "
        feedback " | | \ \ | | | | | | | | | | | | "

```

```

feedback " |_| \\\_____|____/  |_/_\  \\\| \\\|_| "
feedback " "
feedback "Please restart to implement changes!"
feedback "To Restart type sudo reboot"
fi
}

#raspberrypi yi yeniden başlatın
#####
#####
#####identify_cie
#####identify_robot
#####check_root_user
#####display_welcome_msg
#####check_internet

echo "Installing GoPiGo software in ${ROBOT_DIR}"
echo ""

install_dependencies

#Copy Software Servo
cp -R $ROBOT_DIR/Firmware/SoftwareServo/ /usr/share/arduino/libraries/

chmod +x gopigo
cp gopigo /usr/bin

cd $ROBOT_DIR/Software/Python
python setup.py install
python3 setup.py install

```

```

install_DHT
install_wiringpi
install_spi_i2c
install_avr
install_line_follower
install_control_panel

#sudo rm -r /tmp/di_update

sudo chmod +x $ROBOT_DIR/Software/Scratch/GoPiGo_Scratch_Scripts/*.sh

call_for_reboot
#GoPiGo yazılımını $ {ROBOT_DIR} alanına kurma

```

İnstal_avrdude.sh

```

#!/usr/bin/env bash

# Create AVRDUDE folder. Create it if it does not exist

#AVRDude programıyla Programmer's Notepad'in içinden derlediğiniz programı doğrudan işlemciye yükleyebilirsiniz.

#avrdude klosörü var mı diye bakar yoksa baştan oluşturur.

create_avrdude_folder(){

    AVRDUDE_DIR='/home/pi/Dexter/lib/AVRDUDE'
    if [ -d "$AVRDUDE_DIR" ]; then
        echo $AVRDUDE_DIR" Found!"
    else
        DIRECTORY='/home/pi/Dexter'
        if [ -d "$DIRECTORY" ]; then
            # Will enter here if $DIRECTORY exists, even if it contains spaces
            echo $DIRECTORY" Directory Found !"
        else

```

```

echo "creating \"$DIRECTORY"
mkdir $DIRECTORY
fi

DIRECTORY='/home/pi/Dexter/lib'
if [ -d "$DIRECTORY" ]; then
    # Will enter here if $DIRECTORY exists, even if it contains spaces
    echo "$DIRECTORY" Directory Found!
else
    echo "creating \"$DIRECTORY"
    mkdir $DIRECTORY
fi

pushd $DIRECTORY
git clone https://github.com/DexterInd/AVRDUDE.git
popd
fi
}

# Install Avrdude 5.1 from Dexter repos
#avrdude 5.1 yükleyin
install_avrdude(){
    #Updating AVRDUDE
    #güncelleyin
    FILENAME=tmpfile.txt
    AVRDUDE_VER=5.10
    avrdude -v &> $FILENAME

    #Avrdude 5.1 mevcut değilse yalnızca yükleyin
    if grep -q $AVRDUDE_VER $FILENAME
    then
        echo "avrdude" $AVRDUDE_VER "Found"

```

```

else

echo "avrdude" $AVRDUDE_VER "Not Found,Installing avrdude now"

create_avrdude_folder

#####
#AVRDUDE kurulumu
#####
pushd /home/pi/Dexter/lib/AVRDUDE/avrdude

# Avrdude deb paketini kurun

# Dosyaların avrdude klasöründe olması gerektiği için wget'a gerek yok

# wget https://github.com/DexterInd/AVRDUDE/raw/master/avrdude/avrdude_5.10-
4_armhf.deb

sudo dpkg -i avrdude_5.10-4_armhf.deb

sudo chmod 4755 /usr/bin/avrdude

# Kurulum dosyalarını ayarlayın

# wget http://project-downloads.drogon.net/gertboard/setup.sh

chmod +x setup.sh

sudo ./setup.sh

# pushd /etc/minicom

# sudo wget http://project-downloads.drogon.net/gertboard/minirc.ama0

#makine tarafından üretilen dosya parametrelerini değiştirmek için= minicom

# sudo sed -i '/Exec=arduino/c\Exec=sudo arduino' /usr/share/applications/arduino.desktop

echo " "

popd

fi

rm $FILENAME

}

```

Setup.sh

```
#!/bin/bash
```

```
cd /tmp
```

```
doBackup() {  
    cd $1  
    echo -n " $2: "  
    if [ -f $2.bak ]; then  
        echo "Backup of $2 exists, not overwriting"  
        #yedekleme var üzerine yazılmaz  
    else  
        mv $2 $2.bak  
        mv /tmp/$2 .  
        echo "OK"  
    fi  
}
```

```
#Ahududu Pi'yi Gertboard'la çalışacak şekilde ayarlama  
echo "Setting up Raspberry Pi to make it work with the Gertboard"  
echo "and the ATmega chip on-board with the Arduino IDE."  
#Ve ATmega Arduino IDE ile birlikte geliyor.  
echo ""  
echo "Checking ..."  
#kontrol eder  
echo -n " Avrdude: "  
if [ ! -f /etc/avrdude.conf ]; then  
    #yükülü değilse yükleyin  
    echo "Not installed. Please install it first"  
    exit 1  
fi
```

```

fgrep -sq GPIO /etc/avrdude.conf

if [ $? != 0 ]; then
    #GPIO desteği yok. Lütfen doğru sürümü yüklediğinizden emin olun
    echo "No GPIO support. Please make sure you install the right version"
    exit 1
fi

echo "OK"

echo -n " Arduino IDE: "

if [ ! -f /usr/share/arduino/hardware/arduino/programmers.txt ]; then
    echo "Not installed. Please install it first"
    exit 1
fi

echo "OK"

echo "Fetching files:"

for file in boards.txt programmers.txt avrsetup ; do
    echo " $file"
    rm -f $file
    wget -q http://project-downloads.drogon.net/gertboard/$file
done

#avrsetup ve setup.sh ve avrdude bulunmaktadır.

echo "Replacing/updating files:"

#dosyaları güncelleme

rm -f /usr/local/bin/avrsetup
mv /tmp/avrsetup /usr/local/bin
chmod 755 /usr/local/bin/avrsetup

```

```

cd /etc
echo -n "inittab: "
if [ -f inittab.bak ]; then
    echo "Backup exists: not overwriting"
    #yedekleme
else
    cp -a inittab inittab.bak
    sed -e 's/^.*AMA0.*$/#\0/' < inittab > /tmp/inittab.$$
    mv /tmp/inittab.$$ inittab
    echo "OK"
fi

cd /boot
echo -n "cmdline.txt: "
if [ -f cmdline.txt.bak ]; then
    echo "Backup exists: not overwriting"
    #üzerine yazma
else
    cp -a cmdline.txt cmdline.txt.bak
    cat cmdline.txt | \
        sed -e 's/console=ttyAMA0,115200//' | \
        sed -e 's/console=tty1//' | \
        sed -e 's/kgdboc=ttyAMA0,115200//' > /tmp/cmdline.txt.$$
    mv /tmp/cmdline.txt.$$ cmdline.txt
    echo "OK"
fi

doBackup /usr/share/arduino/hardware/arduino boards.txt
doBackup /usr/share/arduino/hardware/arduino programmers.txt

echo "All Done."

```

```
echo "Check and reboot now to apply changes."  
#kontrol edin ve yeniden başlatın  
exit 0
```

GOPİGO İLE TOP TAKİBİ

```
# # Gerekli paketleri içe aktar  
from collections import deque #Nesnenin izlendiği n adet noktanın verimli şekilde depolanması için  
deque veri tipi kullanıldı  
import numpy as np #Dizi kullanımı için içe aktarıldı  
import argparse # Ekrana yönleri yazdırma için kullanıldı  
import imutils #Döndürme, yeniden boyutlandırma,konturları sıralama, kenarları algılama gibi  
g.işleme işlevlerini gerçekleştirmek için içe aktarıldı  
import cv2 #opencv kütüphanesinden gerekli fonksiyonları içe aktarır  
from picamera import PiCamera #picamera modülünün temel fonksiyonlarını içe aktarır  
from picamera.array import PiRGBArray  
import os #Dizin oluşturma silme değiştirme ve çeşitli işletim sistemleriin komutlarını python içinde  
kullanılmasını sağlar  
from gopigo import * #GoPigo kontrolü için temel fonksiyonların olduğu modül  
import sys      #Çalışan programı kapatmak için kullanılır  
import io #Herhangi bir giriş çıkış türleri için kullanılır  
os.system("v4l2-ctl --set-fmt-video=width=600,height=600,pixelformat=1")  
os.system("v4l2-ctl --set-parm=32") #fps 32 olarak ayarlandı  
os.system("sudo modprobe bcm2835-v4l2") #Kamera için v4l2 sürücüsünü aktif ettik  
# argümanları ayırma işlemleri  
ap = argparse.ArgumentParser()  
ap.add_argument("-v", "--video",  
                help="path to the (optional) video file")  
ap.add_argument("-b", "--buffer", type=int, default=64,  
                help="max buffer size")  
args = vars(ap.parse_args())
```

```
# mavinin renginin alt ve üst sınırları belirlendi
# topun HSV renk alanı
blueLower = (110,50,50)
blueUpper = (130,255,255)

# istenen noktaların listesini başlat,çerçeve sayıcı
# koordinat noktaları
pts = deque(maxlen=args["buffer"])
print ("Pts ",pts)
counter = 0
(dX, dY) = (0, 0)
direction = ""

# eğer bir video yolu sağlanmadıysa referans olarak al
# picamera default ayarları yapıldı
if not args.get("video", False):
    camera = PiCamera()
    camera.sharpness = 0
    camera.contrast = 0
    camera.brightness = 50
    camera.saturation = 0
    camera.ISO = 0
    camera.video_stabilization = False
    camera.exposure_compensation = 0
    camera.exposure_mode = 'auto'
    camera.meter_mode = 'average'
    camera.awb_mode = 'auto'
    camera.image_effect = 'none'
    camera.color_effects = None
    camera.rotation = 0
    camera.hflip = False
    camera.vflip = False
```

```

# camera.crop = (0.0, 0.0, 1.0, 1.0)
camera.resolution=(320,240)
#fps=FPS().start()
#vs=PiVideoStream().start()

# otherwise, grab a reference to the video file
else:
    camera = cv2.VideoCapture(args["video"])

#keep looping
frame=None

radius=0

while True:

    c=cv2.waitKey(1)%0x100
    if c==120:
        state=True
        print("Stop")
    elif c==10:
        state=False
        print("Start")

    if state == True:
        stop()
        continue

    # Geçerli çerçeveyi yakala
    # Eğer çerçeve yakalanmadıysa
    # ve videopunu sonuna ulaşıldıysa

```

```

# çerçeveyi yeniden boyutlandır,bulanıklaştır ve HSV'ye dönüştür
# renk alanı

#frame=vs.read()

rawCapture=PiRGBArray(camera,size=(320,240))

backstream=io.BytesIO()

camera.capture(backstream,format='jpeg',use_video_port=True)

datapak=np.fromstring(backstream.getvalue(),dtype=np.uint8)

frame=cv2.imdecode(datapak,1)

hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

# Mavi rengi için bir maskeleme işlemi gerçekleştir
mask = cv2.inRange(hsv, greenLower, greenUpper)
mask = cv2.erode(mask, None, iterations=2)
mask = cv2.dilate(mask, None, iterations=2)

# Maskede kontur(çevre) bulma ve mevcut
# (x, y) topun ortası
cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
                        cv2.CHAIN_APPROX_SIMPLE)[-2]
center = None

# En az bir kontur bulunursa devam et
if len(cnts) > 0:
    # Maskedeki en büyük konturu bulup kullan. En az çevreleyen çevreyi hesapla
    #radius=0
    tRadius =radius
    c = max(cnts, key=cv2.contourArea)
    ((x, y), radius) = cv2.minEnclosingCircle(c)
    M = cv2.moments(c)

```

```

center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))

# Eğer yarıçap minimum boyuta ulaşırsa devam et

if radius > 10:

    # Çemberi ve çemberin merkezini çiz

    # Sonra izlenen noktaların listesini güncelle

    cv2.circle(frame, (int(x), int(y)), int(radius),

               (0, 255, 255), 2)

    cv2.circle(frame, center, 5, (0, 0, 255), -1)

    pts.appendleft(center)

    durum=1

if (radius)<10:

    print ("Stop")

    stop()

    continue

else:

    #print('radius = ',radius)

    #print("T radius = ",tRadius)

    #print("FRAK = ",(tRadius-radius))

    stop()

    continue

    # İzlenen noktalar kümesi üzerinden döngü

    for i in np.arange(1, len(pts)):

        # İzlenen noktalardan biri yok ise, yoksay

        if pts[i - 1] is None or pts[i] is None:

            continue

        try:

            t= pts[-10]

        except IndexError:

            break

```

```

# Buffer içinde yeterli nokta birikmiş olup olmadığını kontrol et
if counter >= 10 and i == 1 and pts[-10] is not None:

    # X ve y arasındaki farkı hesapla
    # Yönü koordine et ve yeniden başlat
    dX = pts[-10][0] - pts[i][0]
    dY = pts[-10][1] - pts[i][1]
    (dirX, dirY) = ("", "")
    if np.abs(dX) > 20:
        left() if np.sign(dX) == 1 else right()

    # Hareket varsa
    # X-yönü
    #gopigoyu doğu ve batı yönünde ilerlet
    #else
    # Hareket varsa
    # Y-yönü
    #gopigoyu kuzey ve güney yönünde ilerlet

if np.sign(dX) == 1:
    dirX = "Left"
    print("Left")
    right() #sağ motoru çalıştırıp sola döner
    continue

elif np.sign(dY) == 1:
    dirX="Right"
    print("Right")
    left() #sol motoru çalıştırıp sağa döner
    continue

elif (tRadius-radius) > 0: #sonraki çemberin yarıçapından ilk çemberin yarıçapını
#çıkarıyor 0 dan büyükse ileri
    dirX="FWD"

```

```

print("ileri")

fwd() #iki motoruda ileri yönde çalıştırır

continue

elif (tRadius-radius) < 0:

    dirt="BWD"

    print("Geri")

    bwd() #iki motorda geri yönde çalışır.

    continue

else:

    stop() #diğer durumlarda durur

    continue #continue yapıp tekrar işi başa alır.

else:

    stop()

    continue

if dirX != "" and dirY != "":

    # X ve Y noktalarını ekrana yazdırma

    direction = "{ }-{ }".format(dirY, dirX)

else:

    # otherwise, only one direction is non-empty

# otherwise, compute the thickness of the line and

# draw the connecting lines

thickness = int(np.sqrt(args["buffer"] / float(i + 1)) * 2.5)

# diğer durumlarda hattın kalınlığını hesapla

# bağlantı çizgilerini çiz

# Çerçevenin hareket deltalarını ve hareket yönünü göster

```

```

cv2.putText(frame, direction, (10, 30), cv2.FONT_HERSHEY_SIMPLEX,
           0.65, (0, 0, 255), 3)

cv2.putText(frame, "dx: {} , dy: {}".format(dX, dY),
           (10, frame.shape[0] - 10), cv2.FONT_HERSHEY_SIMPLEX,
           0.35, (0, 0, 255), 1)

cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

counter += 1

# q tuşuna basınca döngüden çıkış
if key == ord("q"):
    break

# kamerayı temizle ve açık pencereleri kapat
camera.release()

cv2.destroyAllWindows()

```

Kaynakça

<https://github.com/DexterInd/GoPiGo>

<http://cyaninfinite.com/tutorials/installing-opencv-in-ubuntu-for-python-3/>

<https://www.dexterindustries.com/gopigo/>

<https://www.dexterindustries.com/gopigo-tutorials-documentation/>

www.pyimagesearch.com

<https://etcher.io>

http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html