



T.C.

BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ

ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

EEM433 MİKROİŞLEMCİ UYGULAMALARI

RASPBERRY Pİ 2 DENEY FÖYÜ

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	Hata! Yer işareti tanımlanmamış.
TEŞEKKÜR	Hata! Yer işareti tanımlanmamış.
KISALTMALAR ve SİMGELER.....	v
1.GİRİŞ.....	1
2. RASPBERRY Pİ.....	2
3. RASPBERRY Pİ İŞLETİM SİSTEMİ KURULUMU.....	4
3.1. HAFIZA KARTINI BİÇİMLENDİRME.....	5
3.2. İMAJIN KURULMASI	5
4. RASPBERRY Pİ'NİN ETHERNET BAĞLANTISI İLE KULLANIMI.....	6
5. RASPBERRY Pİ MATLAB İLE KULLANIMI	12
6. DENEYLER.....	19
6.1. Deney 1: RASPBERRY Pİ İLE LED YAKMA	19
6.1.1. Amaç	19
6.1.2. Genel Bilgiler	19
6.1.3. Deney Devresi	20
6.1.4. Kod	21
6.1.5. Rpi.GPIO as GPIO Kütüphanesi	21
6.1.6. Time kütüphanesi kullanıldığında program kodu	22
6.1.7. Time Kütüphanesi	22
6.2. Deney 2: BUTON KULLANIMI	23
6.2.1. Amaç	23
6.2.2. Genel Bilgiler	23
6.2.3. 1. Deney Devresi	24
6.2.4. Kod	25
6.2.5. 2. Deney devresi.....	26
6.2.6. Kod	27
6.3. Deney 3: ANAHTAR KULLANIMI	28
6.3.1. Amaç	28
6.3.2. Genel Bilgiler	28
6.3.3. Deney Devresi	28
6.3.4. Kod	29

6.4. Deney 4: PWM İLE LED SÜRME	31
6.4.1. Amaç	31
6.4.2. Genel Bilgiler	31
6.4.3. Deney Devresi	32
6.4.4. Kod	33
6.5. Deney 5: TRAFİK LAMBALARI	35
6.5.1. Amaç	35
6.5.2. Genel Bilgiler	35
6.5.3. Deney Devresi	36
6.5.4. Kod	36
6.6. Deney 6: FOTO DİRENÇ'TEN DEĞER OKUMA.....	39
6.6.1. Amaç	39
6.6.2. Genel Bilgiler	39
6.6.3. Deney Devresi	40
6.6.4. Kod	41
6.7. Deney 7: YÜZ ALGILAMA.....	42
6.7.1. Amaç	42
6.7.2. Genel Bilgiler	42
6.7.3. Deney Devresi	42
6.7.4. KOD:	43
6.8. Deney 8: L298N MOTOR SÜRÜCÜ KARTI İLE MOTOR KONTROL	45
6.8.1. Amaç	45
6.8.2. Genel Bilgiler	45
6.8.3. Deney Devresi	46
6.8.4. KOD	47
6.9. Deney 9: ULTRASONİK SES SENSÖRÜ KULLANIMI	49
6.9.1. Amaç	49
6.9.2. Genel Bilgiler	49
6.9.3. Deney Devresi	50
6.9.4. KOD	51

6.10. Deney 10: ISI SENSÖRÜ KULLANIMI.....	53
6.10.1. Amaç	53
6.10.2. Genel Bilgiler	53
6.10.3. Deney Devresi	55
6.10.4. KOD	56
6.11. Deney 11: HAREKET SENSÖRÜ VE KAMERA KULLANIMI.....	59
6.11.1. Amaç	59
6.11.2. Genel Bilgiler	59
6.11.3. Deney Devresi	60
6.11.4. KOD	61
6.12. Deney 12: ANALOG SİNYAL OKUMA	62
6.12.1. Amaç.	62
6.12.2. Genel Bilgiler	62
6.12.3. Deney Devresi	65
6.12.4. KOD	65
6.13. Deney 13: I2C ile 4 Digit 7 Segment Display Kontrol Etme	67
6.13.1. Amaç.	67
6.13.2. Genel Bilgiler	67
6.13.3. Deney Devresi	69
6.13.4. KOD	70
6.14. Deney 14: Basit Voltmetre Yapma	71
6.14.1. Amaç	71
6.14.2. Genel Bilgiler	71
6.14.3. Deney Devresi	71
6.14.4. KOD	73
SONUÇ.....	Hata! Yer işareti tanımlanmamış.
KAYNAKÇA.....	74

ÖZET

Raspberry Pi Birleşik Krallık'ta Raspberry Pi Vakfı tarafından geliştirilmiş ARM tabanlı USB, ethernet, HDMI, 3.5 mm ses çıkışı, kompozit video çıkışı gibi çevre birimleri bulunan kredi kartı büyüklüğünde fansız olarak üretilmiş tek kartlı bir mini bilgisayardır. Linux tabanlı özelleştirilmiş işletim sistemleri ile kullanılmaktadır. Üzerinde bulunan GPIO'lar sayesinde elektronik devreleri kontrol edebilmek mümkündür. Yüksek seviye dillerle (C, Python gibi) programlanabilmektedir.

Bu çalışmada raspberry pi 2 ile gerçekleştirilmiş 14 tane deney bulunmaktadır. Deneyler GPIO kullanımı, pull up direnç ve pull down direnç, buton ve anahtar kullanımı, ultrasonik ses sensörü kullanımı, ısı ve nem sensörü kullanımı, motor sürücü kartı kullanımı, raspberry pi kamera modülü kullanımı, PIR hareket sensörü kullanımı, PWM kullanımı, analog dijital konverter kullanımı, 4 digit 7 segment display kullanımı, SPI haberleşme ve I2C haberleşme konularını içermektedir.

Anahtar Kelimeler:

Raspberry Pi 2, Python, PIR, DHT11, HC-SR04, MCP3008, PWM, I2C, SPI, 4 digit 7 segment display, Foto direnç, Yüz algılama

KISALTMALAR ve SİMGELER

KISALTMALAR

GPIO: General Purpose Input-Output (Genel Amaçlı Girdi-Çıktı)

PWM: Pulse Width Modulation (Darbe Genişlik Modulasyonu)

LDR: Light Dependent Resistor (Işık Bağımlı Direnç)

LED: Light Emitting Diod (Işık Yayan Diyot)

CSI: Camera Serial Interface (Kamera Seri Arayüzü)

GHz: Gigahertz

KHz: Kilohertz

NTC: Negative Temperature Coefficient (Negatif Sıcaklık Katsayısı)

PIR: Passive InfraRed (Pasif Kızılötesi)

SAR:Successive Approximation Register (Ardışık Yaklaşımlı Kaydedici)

I2C:Inter-Integrated Circuit (Gömülü Entegre Devre)

SPI:Serial Peripheral Interface (Seri Çevrebirim Arayüzü)

SİMGELER

Ω : Ohm (Direnç Birimi)

μ : Mikro (Ölçü birimi)

1. GİRİŞ

Raspberry Pi, ilk ürünü 29 şubat 2012 tarihin de piyasaya sürülmüş çocuklara bilgisayar bilimini ve programcılığını öğretmek amacıyla tasarlanmış kredi kartı büyüklüğünde tek kart bir mini bilgisayardır. Raspberry pi'nin A, A+, B, B+, Pi2, Pi3, PiZero modelleri bulunmaktadır. Ethernet girişi, USB girişi, Sdcard girişi, HDMI girişi ve kompozite video girişi gibi bağlantı birimleri bulundurmaktadır. Raspberry pi üzerin de bulundurduğu GPIO (General Purpose Input Output)' lar sayesinde elektronik, robotik ve mekatronik alanlarında da oldukça ilgi görmektedir.

Bu çalışma da raspberry pi'nin herhangi bir linux tabanlı işletim sistemiyle ve matlab ile kurulumu, SSH (Secure Shell) bağlantısı ile bilgisayar ekranından raspberry pi'nin kullanımı hakkında bilgi verilmektedir. Ayrıca GPIO'ların kullanımı üzerine 14 tane deney gerçekleştirilmiş ve bu deneyler hakkında da bilgiler verilmiştir. Deneyler, genel olarak GPIO'ların kullanımı, SPI haberleşme (Serial Peripheral Interface), I2C haberleşme (Inter-Integrated Circuit), PWM sinyali, kamera modülünün kullanımı, sensör kullanımı gibi konuları içermektedir.

2. RASPBERRY Pİ

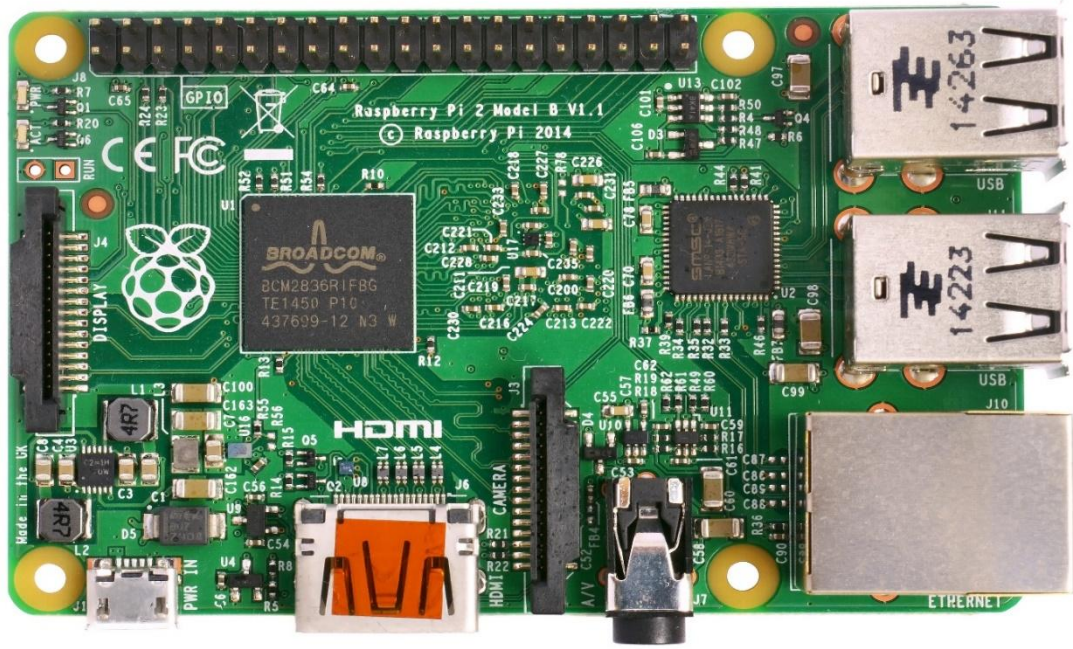
Raspberry Pi Birleşik Krallık'ta Raspberry Pi Vakfı tarafından geliştirilmiş ARM tabanlı USB, ethernet, HDMI, 3.5 mm ses çıkışı, kompozit video çıkışı gibi çevre birimleri bulunan kredi kartı büyüklüğünde fansız olarak üretilmiş tek kartlı bir bilgisayardır. Asıl üretim amacı okullarda bilgisayar bilimini ve programlamayı öğretmek olsada sahip olduğu gpio pinleri ile elektronik sistemlerle haberleşme amacıyla kullanılması sayesinde, elektronik ve mekatronik alanında kendine yeni bir kullanım sahası bulmuştur. Python, BBC Basic, C ve Perl programlama dilleri ve matlab ile programlanabilmektedir. Raspberry pi Raspbian (Debian Wheezy tabanlı), Pidora (Fedora tabanlı), Snappy Ubuntu Core, Pardus ARM, Arch Linux ARM ve Windows 10 IoT Core ve daha pek çok işletim sistemi ile çalışabilmektedir.

Raspberry pi nin model A, model B, model B+, pi2, pizero ve pi3 gibi modelleri bulunmaktadır.

Raspberry pi 2 nin özellikleri:

- İşlemci: Broadcom BCM2836 SoC 900 MHz quad-core ARM Cortex A7
- Grafik İşlemci: Broadcom VideoCore IV GPU (HD 1080 piksel destekli)
- Ram: 1 GB (GPU ile paylaşımlı)
- USB: 4 x USB2.0 Port
- Video Çıkışları: Kompozit Video Çıkışı (PAL and NTSC), HDMI or Raw LCD (DSI)
- Ses Çıkışı: 3.5mm fiş veya Audio over HDMI üstü ses sistemi yardımıyla Audio Çıkışı
- Bellek: MicroSD
- Ethernet: 10/100 Ethernet (RJ45)
- GPIO pinleri (40 adet)
- Çalışma gerilimi: +5V DC

Raspberry pi nin diğer modellerinde işlemci, RAM, pin sayısı, USB sayısı gibi farklılıklar vardır.



Şekil 2.1. Raspberry Pi 2

3. RASPBERRY Pİ İŞLETİM SİSTEMİ KURULUMU

Raspberry pi bir bilgisayardır ve dahili bir hafızası yoktur bu yüzden seçtiğimiz işletim sisteminin imajını internetten indirip hafıza kartına kurmalıyız sonra raspberry pi ye takarak kullanmaya başlayabiliriz. Burada işletim sistemini nasıl indirip kurmanız gerektiği anlatılacaktır.

Öncelikle aşağıdaki linkten kurulacak işletim sisteminin imajı indirilir. Raspbian en çok kullanılan dağıtımdır ve internet ortamında bir çok kaynak bulunabilir olduğu için bu dağıtımı tercih etmeniz daha iyi olur. Daha sonra hafıza kartını biçimlendirmek için SDformatter V4.0 programı indirilir ve hafıza kartı biçimlendirilir kullandığınız hafıza kartı sorun yaşamamanız için en az 4 GB tercihen 8 GB olmalıdır. Hafıza kartını biçimlendirdikten sonra win32diskimager programıyla indirdiğini işletim sistemi hafıza kartına kurulur.

Kullanılacak programların linkleri

Raspbian indirme linki:

<https://www.raspberrypi.org/downloads/raspbian/>

SDformatter V4.0 programının linki:

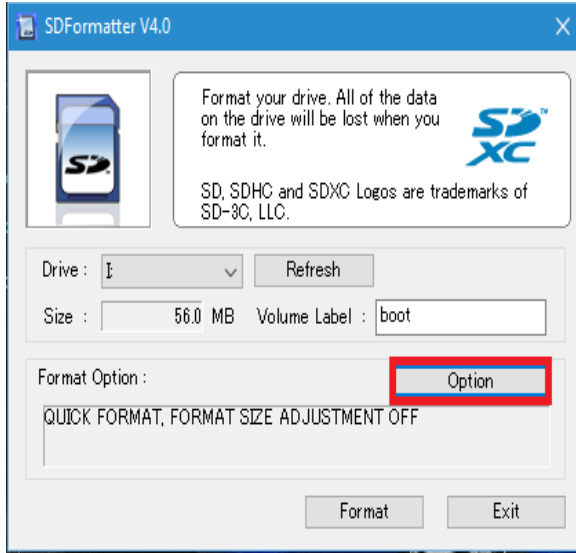
https://www.sdcard.org/downloads/formatter_4/eula_windows/index.html

win32diskimager programının linki:

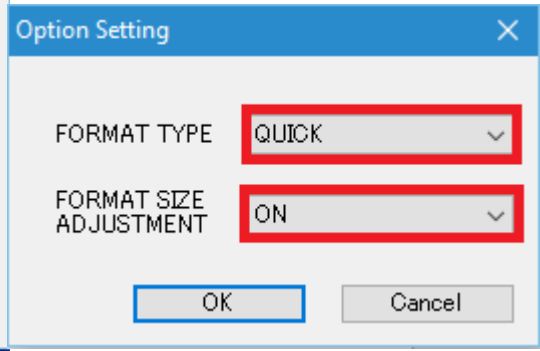
<https://sourceforge.net/projects/win32diskimager/>

3.1. HAFIZA KARTINI BİÇİMLENDİRME

SDformatter V4.0 programını indirip kurduktan sonra aşağıdaki gibi bir ekran açılır. Açılan ekranda options butonuna tıklanarak format size adjustment **ON** format type **QUICK** yapılır ve hafıza kartı formatlanır. (Not: hafıza kartının dosya biçimi **FAT** olmalıdır.)



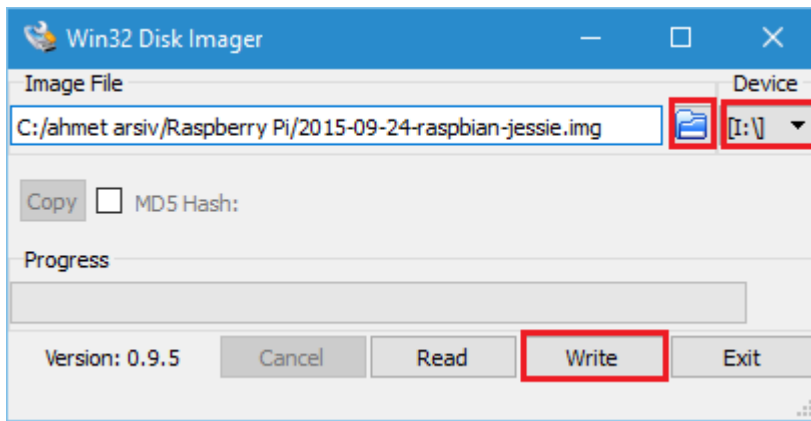
Şekil 3.1. SDFormatter



Şekil 3.2. SDFormatter ayarları

3.2. İMAJIN KURULMASI

Win32diskimager programını çalıştırdığımızda aşağıdaki ekran açılacaktır. Açılan ekranda device dan hafıza kartınızın konumunu seçip, klasör ikonuna tıklayarak imajın konumunu belirttikten sonra imajı kartınıza yazdırabilirsiniz. (Not: İmajınız **.img** yada **.iso** uzantılı olmalı **.zip** gibi bir uzantıyla sıkıştırılmışsa önce açmanız gerekmektedir)



Şekil 3.3. Win32 Disk Imager

Yukarıda anlatılan işlemleri yaptıktan sonra pi'ye hafıza kartını takıp klavye, mouse HDMI ile bir ekrana bağlayıp kullanmaya başlayabilirsiniz. Eğer harici bir ekran yerini bilgisayarınızın ekranını, mouse ve klavyesini kullanmak isterseniz raspberry Lan kablosu ile bilgisayarınıza bağlayarak kullanabilirsiniz.

4. RASPBERRY Pİ'NİN ETHERNET BAĞLANTISI İLE KULLANIMI

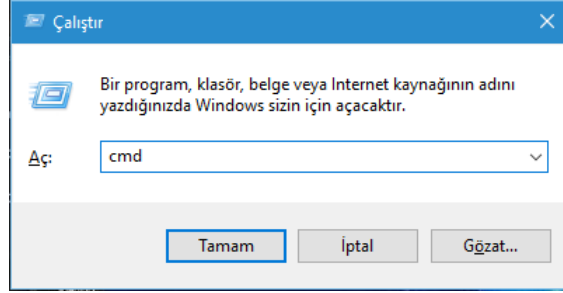
Raspberry pi yi SSH (Secure Shell) bağlantısıyla bilgisayarınızın ekranından kontrol etmeniz için Xming pencere uygulamasına ve putty sunucu bağlantı programlarına ihtiyacınız vardır. SSH bağlantısı bilgisayar-bilgisayar, bilgisayar-sunucu, sunucu-sunucu bağlantı çeşitleri içerisinde güvenli iletişimi sağlayan bir protokoldür. Sunucunuza SSH ile bağlanmak için gereken yetkilere (kullanıcı adı ve şifre) sahip olmanız gerekir. SSH'ın gelen olarak yetkili kullanıcısı root'tur.

Programların indirme linkleri

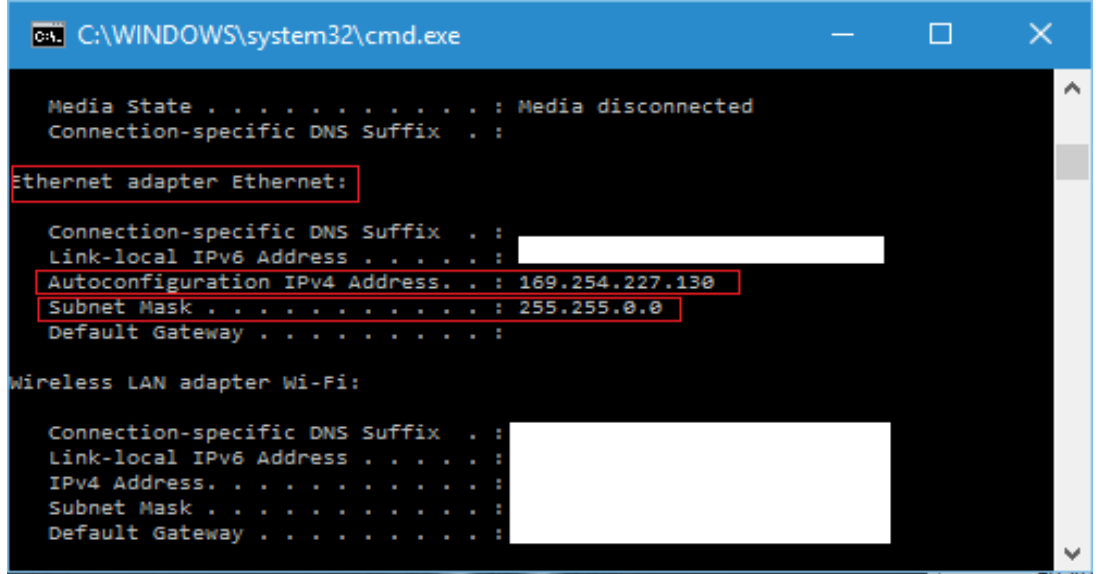
Xming indirme linki: <https://sourceforge.net/projects/xming/>

Putty indirme linki: <http://www.putty.org/>

Putty ve Xming'ı kurduktan sonra bağlantıyı kurabilmek için raspberry pi ye ip adresi atamak gerekir. Bunun için bilgisayarınızın aldığı ip ve subnet maskesi gibi bilgileri öğrenmemiz gerekiyor. Önce imajı yazdırdığımız hafıza kartını raspberriye takıp raspberryyi direk olarak lan kablosu ile bilgisayarımıza bağlıyoruz ve raspberryyi enerjilendiriyoruz (raspberrypi direk bilgisayarınızın USB girişinden veya 5V 2A değerlerinde bir adabtörle besleyebilirsiniz.) Daha sonra bilgisayarımızda subnet maskesi ve ip değerlerini öğrenmek için başlat/çalıştır a girin "cmd" yazıp komut istemini açın, komut istemi penceresinde "ipconfig" komutuyla ethernet üzerinden aldığımız ip ve subnet maskenizi not edin.



Şekil 4.1. Windows Çalıştır Penceresi



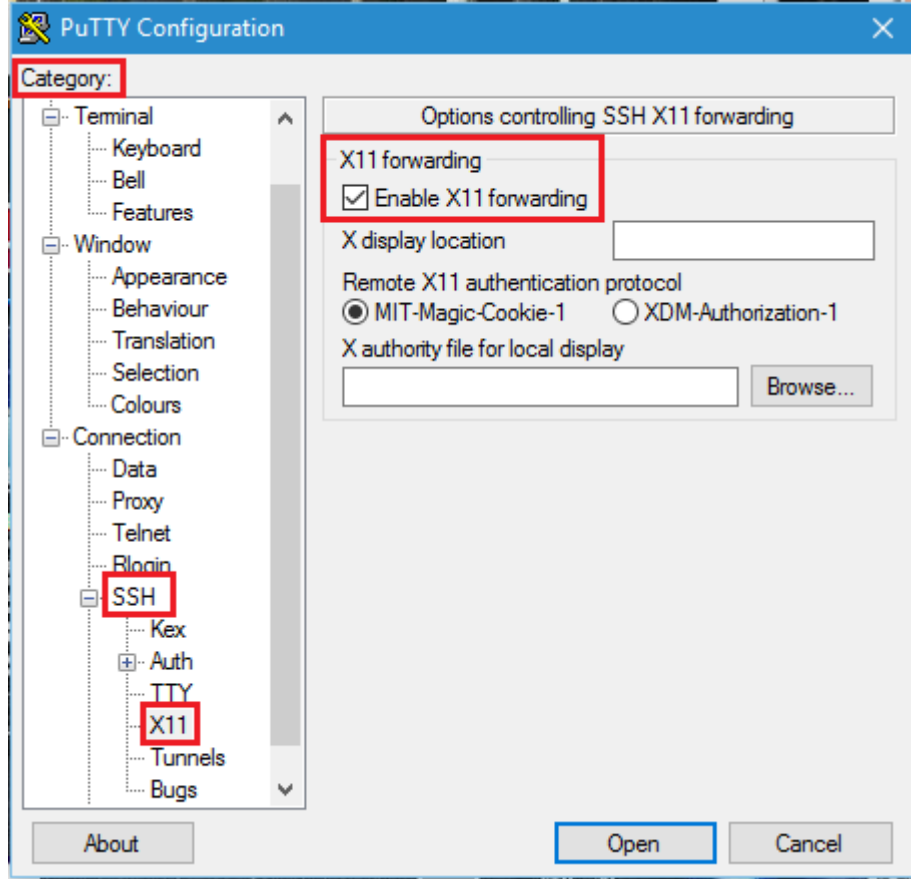
Şekil 4.2. Komut İstemi Penceresi

Yukarıdaki resimde görüldüğü gibi ethernet üzerinden ip adresi “169.254.227.130”. Raspberry için rastgele kendimiz bir ip adresi belirleyebilir ancak belirleyeceğimiz ip adresinin ilk üç bölümü bilgisayarımızın ethernet üzerinden aldığı ip adresinin ilk üç bölümüyle aynı olmalı yani bilgisayarın ip adresi A.B.C.130 gibi ise bizim raspberry için belirleyeceğimiz rastgele ip değeride A.B.C.100 gibi olmalı burada son bölümü bilgisayarın ip adresi ile aynı olmamak şartıyla kendimiz belirleyebiliriz.

Yukarıda anlatılanları yaptıktan sonra raspberry pi yi kapatın ve hafıza kartını bilgisayara takın hafıza kartının içinde cmdline.txt adında bir text dosyası var öncelikle herhangi bir sorun yaşamamak için o dosyayı yedekleyin sonrasında dosyaya girin dosyanın içinde şöyle bir metin karşılayacak sizi “dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait”

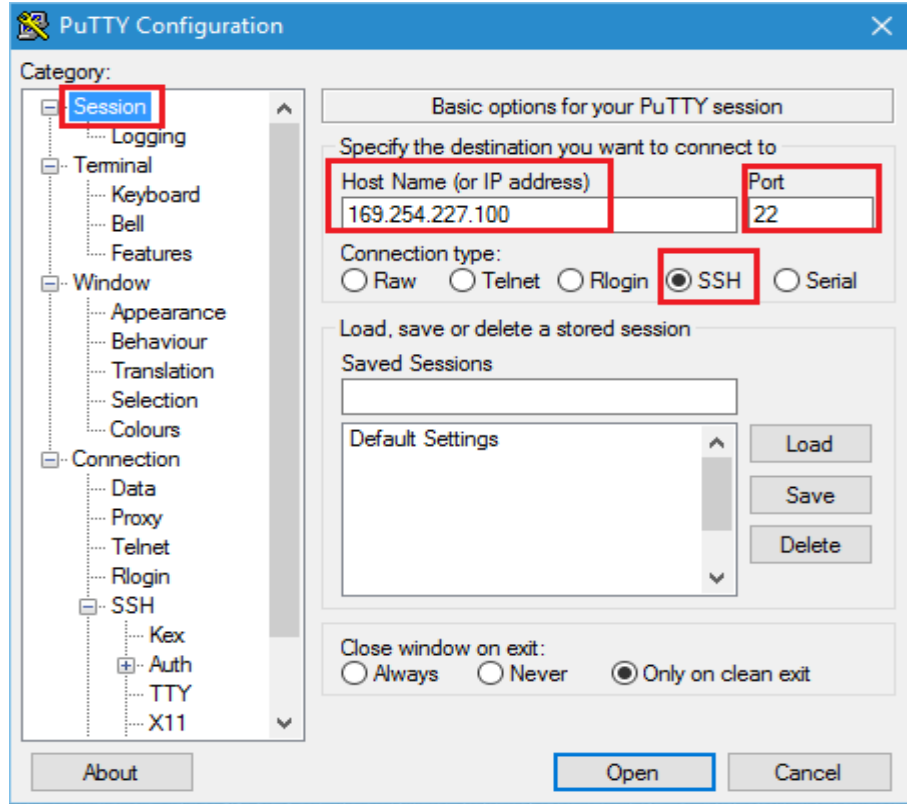
Burada rootwait kelimesinden sonra bir satır boşluk bırakıp “ip=raspberry pi için belirlediğimiz ip adresi::bilgisayarın ip adresi” şeklinde ip adreslerini yazın yani “ip=169.254.227.100::169.254.227.130” şeklinde yazılması gerekmektedir. (Not: Metnin altında boş bir satır var onu kesinlikle silmeyin ve kontrol etmeyi unutmayın) cmd dosyasının içinin son hali “dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait ip=169.254.227.100::169.254.227.130” oldu. Anlatılan şekilde yaptıktan sonra hala raspberry’e bağlanamadıysanız cmdline.txt dosyasının içine rootwait kelimesinden sonra “ip=169.254.227.100::255.255.0.0:rpi:eth0:off” yazın. Burada girdiğiniz ip adresi raspberry için belirlediğimiz ip, sonrasında girilen adres ise bilgisayarın subnet maskesidir.

Yukarıda anlatılan işlemleri tamamladıktan sonra hafıza kartını takıp raspberryyi çalıştırın. Raspberriye bağlanabilmek için önce Xming programını çalıştırın (bu pencere uygulamasıdır arka planda açılır ve bekler) daha sonra putty programını çalıştırın sol tarafta katagoriden SSH bloğunu açın X11 e tıklayın ve sağ tarafta gelen Enable X11 forwarding kutusuğunu tıklayın.



Şekil 4.3. Putty Ayarları

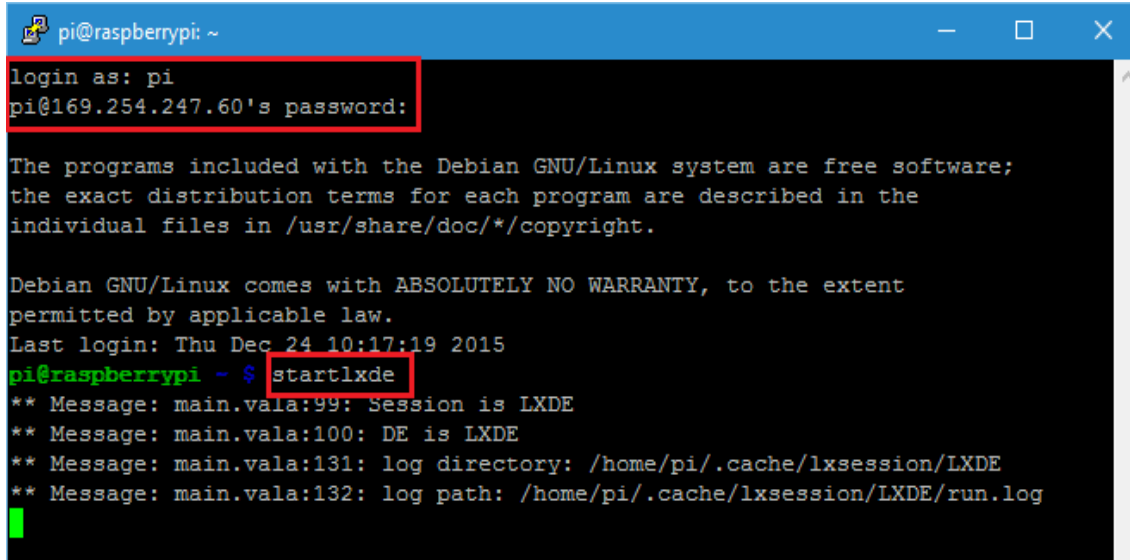
Daha sonra sol taraftaki katagoriden session gelin ve host name kısmına raspberry için belirlediğiniz ip adresini girin, port yerine 22 yazıp ssh'ı seçtikten sonra open'a tıklayın.



Şekil 4.4. Putty Ayarları 2

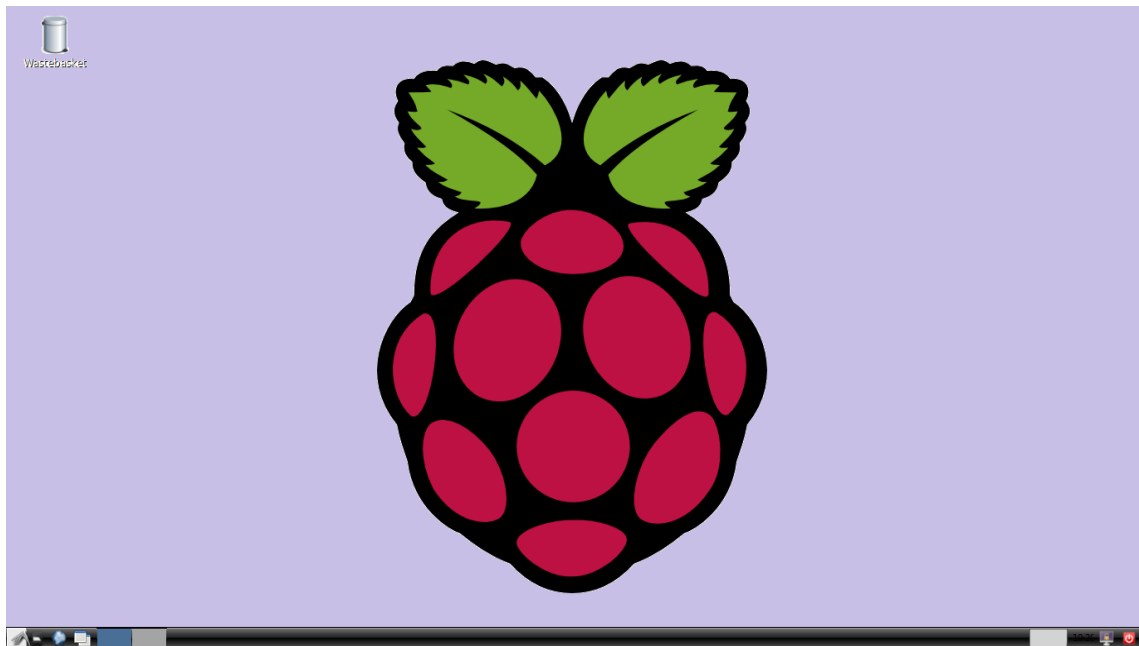
Bu işlemleri yaptıktan sonra sizi siyah bir pencere karşılayacak bura sizden kullanıcı adı ve parola istenecek raspbian imajının kullanıcı adı “pi” ve şifresi “raspberry” dir. (Not: Şifreyi girerken ekrana herhangi bir çıktı gelmemektedir.)

Kullanıcı adı ve şifreyi girdikten sonra terminal üzerinden raspberry piye erişebilirsiniz Xming pencere uygulamasında masaüstünü açabilmek için “startlxde” veya “lxsession” komutu girilir.



```
pi@raspberrypi: ~  
login as: pi  
pi@169.254.247.60's password:  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Dec 24 10:17:19 2015  
pi@raspberrypi ~ $ startlxde  
** Message: main.vala:99: Session is LXDE  
** Message: main.vala:100: DE is LXDE  
** Message: main.vala:131: log directory: /home/pi/.cache/lxsession/LXDE  
** Message: main.vala:132: log path: /home/pi/.cache/lxsession/LXDE/run.log
```

Şekil 4.5. Raspberry Pi Terminal Penceresi

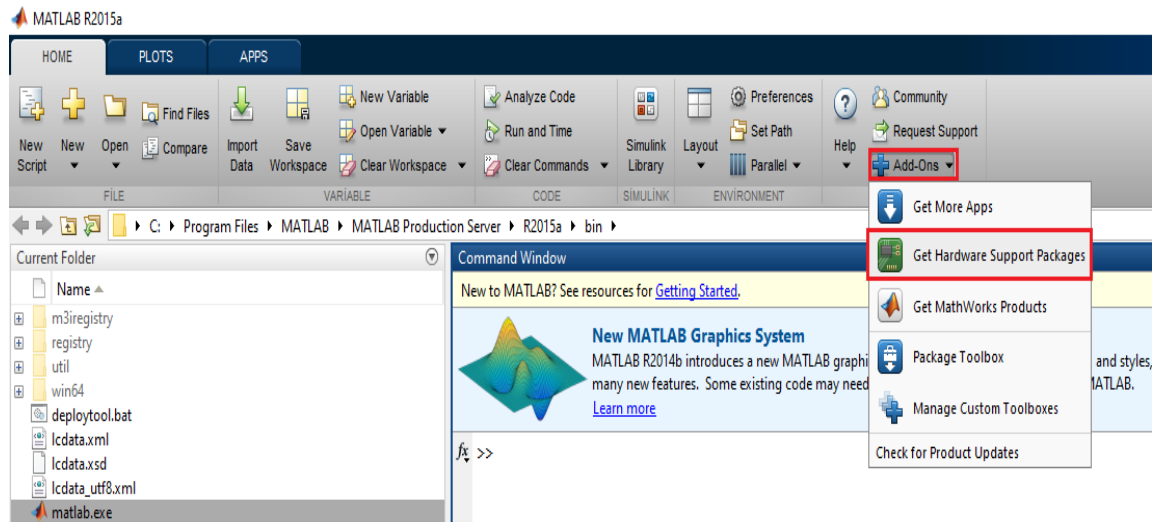


Şekil 4.6. Raspbian Masa Üstü Görüntüsü

5. RASPBERRY Pİ MATLAB ile KULLANIMI

Raspberry pi'nin GPIO pinleri ve kamera modülü gibi çevre birimleri MATLAB programının R2015a versiyonuyla da kontrol edilebilmektedir. Raspberry pi ethernet kablosu ile bilgisayara bağlanarak MATLAB kodlarıyla programlanmaya başlanabilir. MATLAB programı raspberry pi ile bağlantısını kurabilmek için ayarlanmış bir raspbian imajını indirerek hafıza kartının içine kendisi kurmaktadır.

Raspberry pi'yi MATLAB ile kontrol edebilmek için bilgisayarınıza MATLAB R2015a versiyonunu kurun. Raspberry pi kurulumu kısmında anlatıldığı gibi hafıza kartınızı biçimlendirin. MATLAB programını çalıştırın ve aşağıdaki şekilde gösterildiği gibi “Add-Ons” kısmına tıklayarak “Get Hardware Support Packages” kısmını seçin.



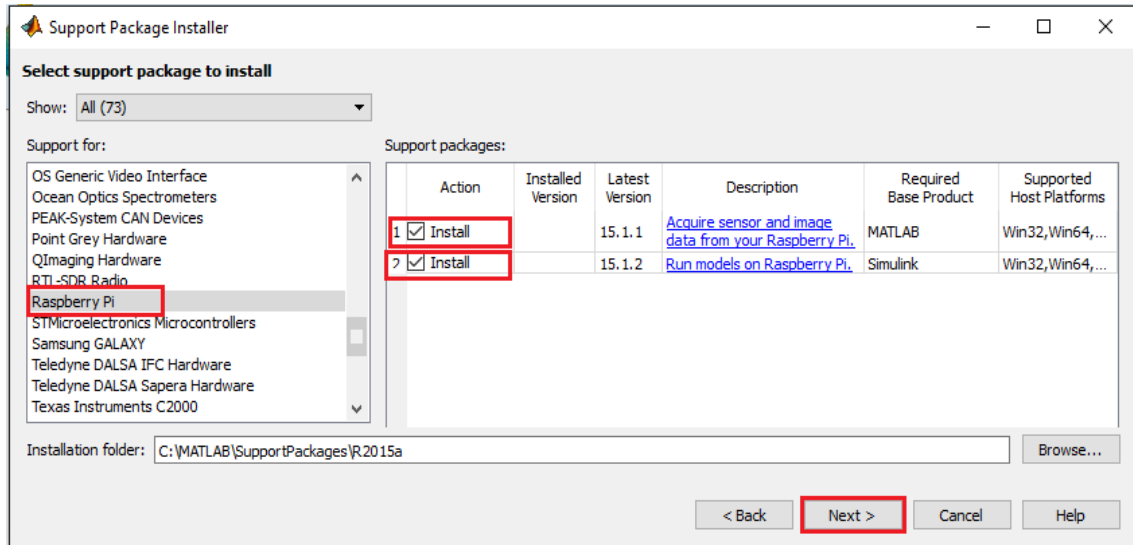
Şekil 5.1. Matlab Donanım Paketi Ekleme

“Get Hardware Support Packages” i seçtikten sonra açılan pencerede “Install from Internet”i seçin ve next’e basın.



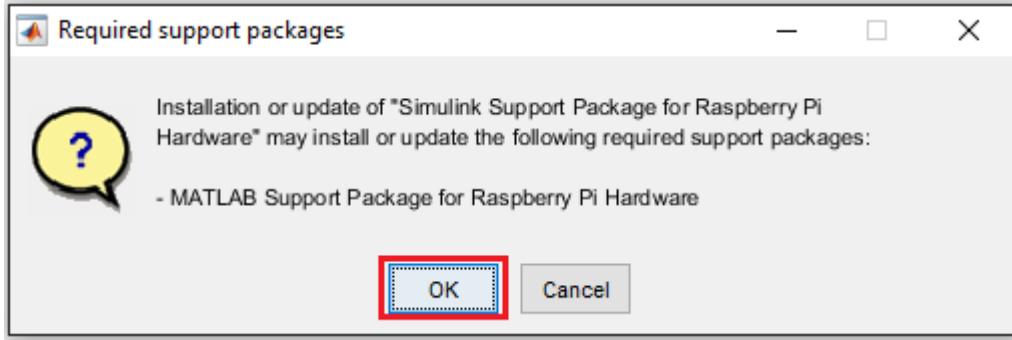
Şekil 5.2. Matlab Support Pakgace Installer Penceresi 1

“Install from Internet”i seçtikten sonra açılan pencerede destek paketleri bulunmaktadır. Sol menüde bulunan bu destek paketleri listesinden “Raspberry Pi”yi seçin raspberry pi için iki tane (Matlab ve Simulink için) destek paketi görülecektir, ikisini de seçtikten sonra next’e basın.



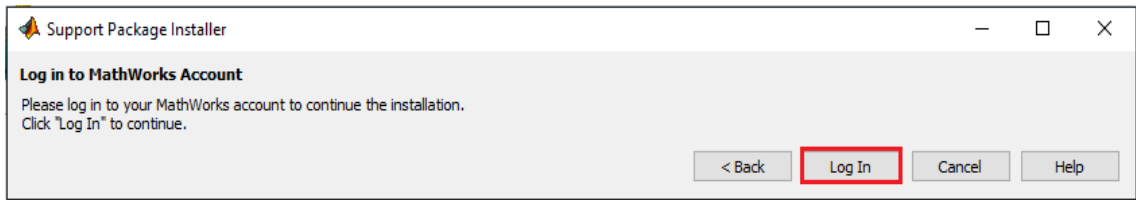
Şekil 5.3. Matlab Support Pakgace Installer Penceresi 2

Raspberry pi destek paketlerini seçtikten sonra açılan pencerede “OK” butonuna tıklayın.



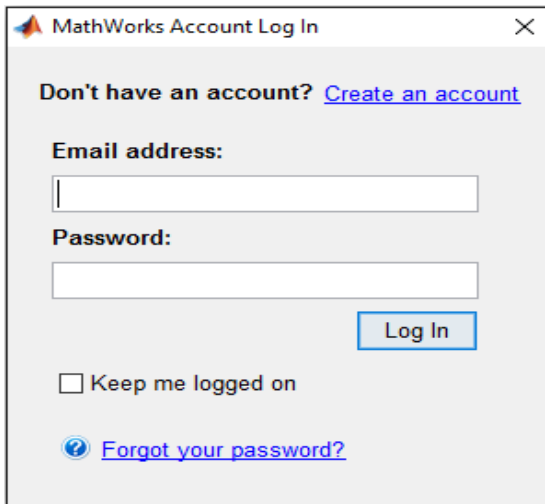
Şekil 5.4. Matlab Support Pakgace Installer Penceresi 3

Devamında sizi aşağıdaki resimde gösterilen pencere karşılayacaktır, “Log In” butonuna tıklayın. “MathWorks Account Log In” penceresi açılacaktır.

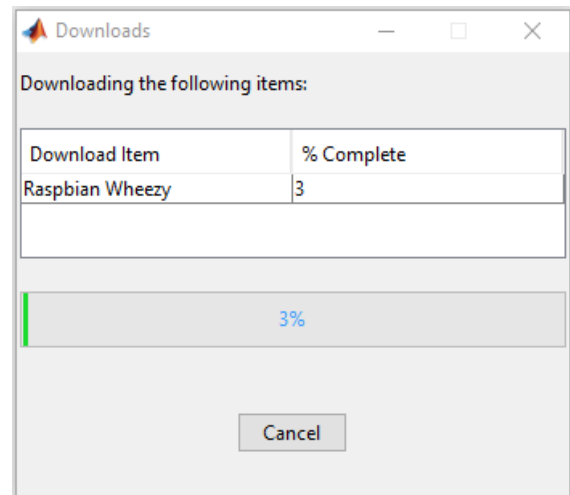


Şekil 5.5. Matlab Support Pakgace Installer Penceresi 4

“MathWorks Account Log In” penceresinde mail adresinizi ve şifrenizi girerek oturum açın. Oturumunuz kontrol edildikten sonra lisans anlaşması penceresi açılacaktır, kabul edip indirme işlemine başlayın.

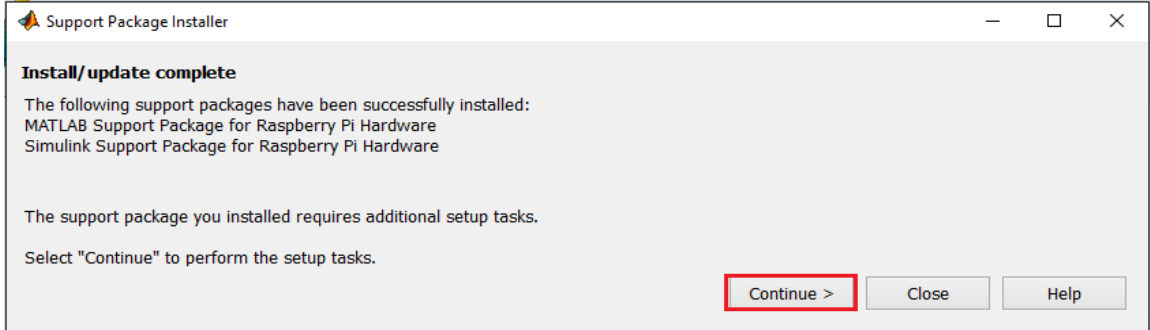


Şekil 5.6. MathWorks Oturum Penceresi

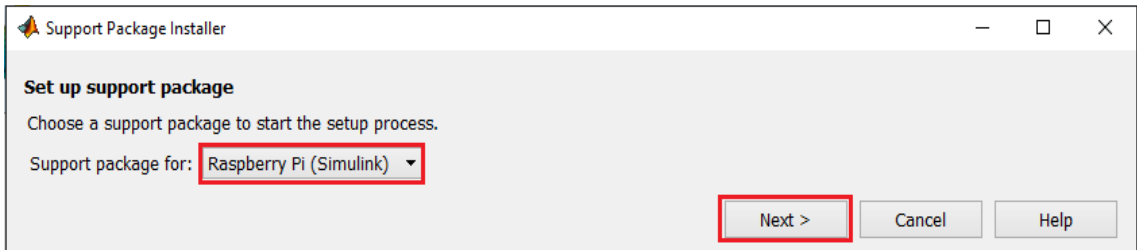


Şekil 5.7. Raspbian İmajı İndirme Penceresi

İndirme işlemi tamamlandıktan sonra açılan pencerede “Continue” butonuna tıklanır ve gelen pencerede support package for kısmı “Raspberry pi (Simulink)” seçilerek next butonuna basılır.

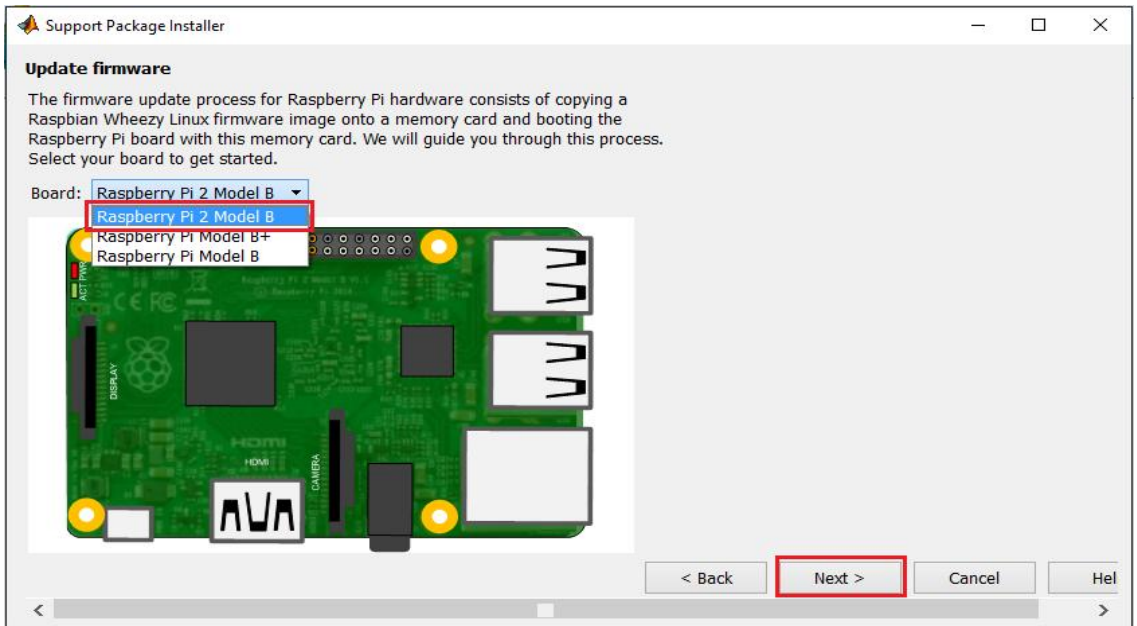


Şekil 5.8. Matlab Support Pakgace Installer Penceresi 5



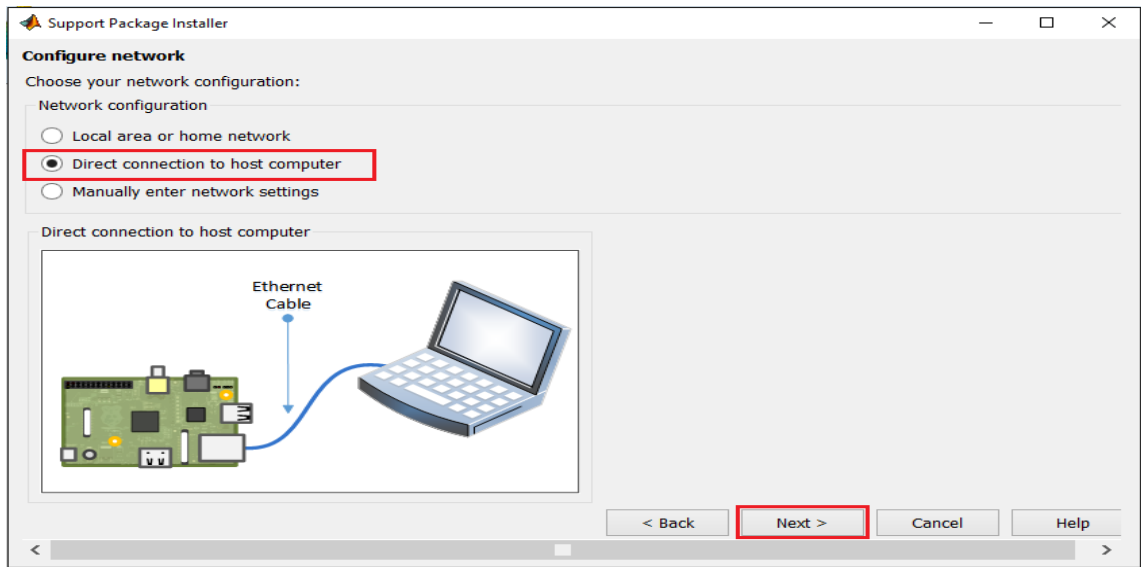
Şekil 5.9. Matlab Support Pakgace Installer Penceresi 6

Açılan pencerede “Board” kısmından kullanılan raspberry pi modeli seçilir ve next butonuna basılır.



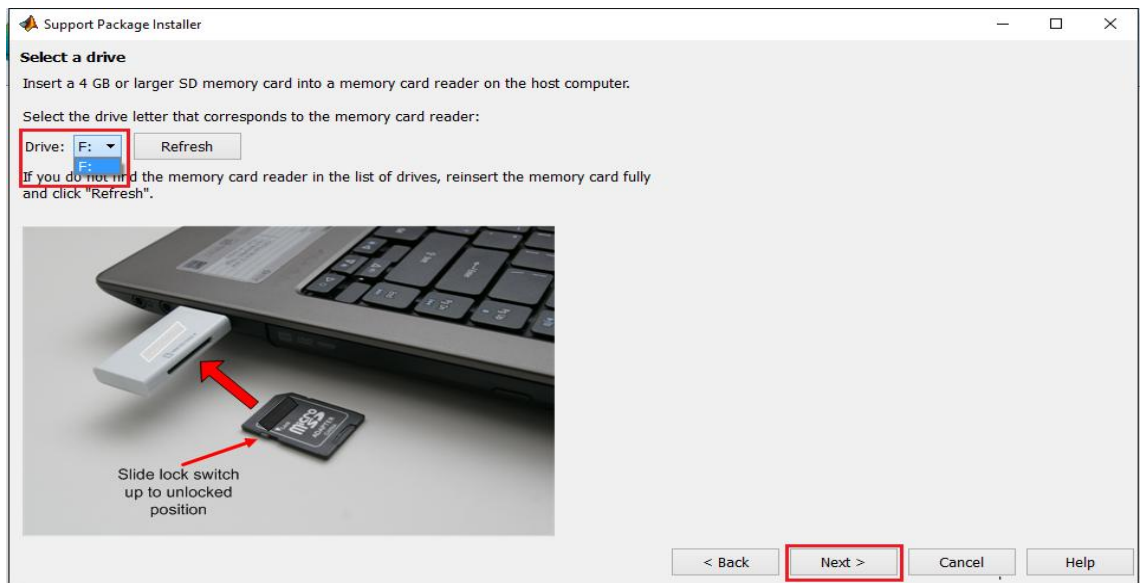
Şekil 5.10. Matlab Support Pakgace Installer Penceresi 7

Raspberry pi modeli seçildikten sonra açılan pencerede raspberry pi ile bilgisayar arasındaki bağlantı seçilecektir. “Direct connection to host computer” seçildikten sonra next butonuna basılır. Raspberry pi bilgisayarla direkt olarak değilde bir modem veya router aracılığıyla bağlanmak istenirse “Local area or home network” seçilmelidir veya “Manually enter network settings” seçeneğiyle bağlantı için gerekli IP adres, subnet mask ve dns adres değerlerini kendiniz belirleyerek de bağlantı gerçekleştirilebilir.

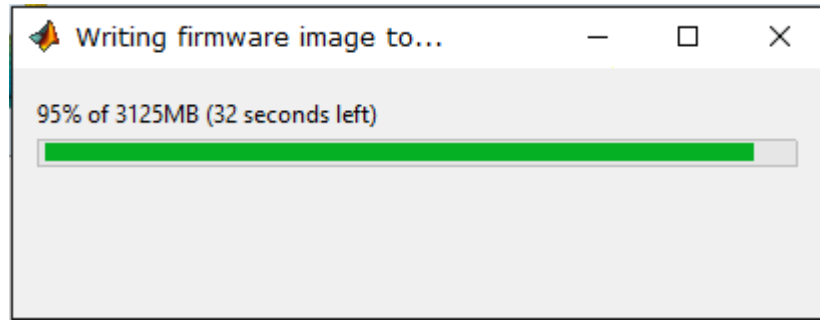


Şekil 5.11. Matlab Support Pakgace Installer Penceresi 8

Raspberry pi bağlantı ayarları yapıldıktan açılan pencerede “Drive” kısmından imajın yazdırılacağı hafıza kartının konumu seçilir ve next butonuna basılır.

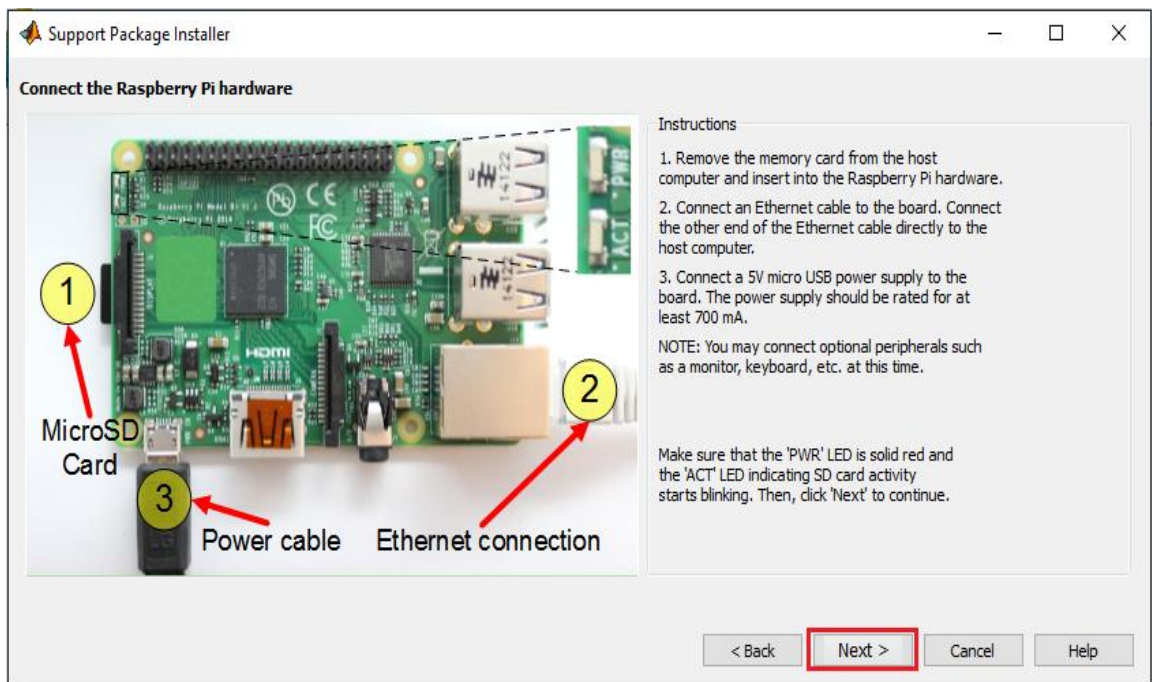


Şekil 5.12. Matlab Support Pakgace Installer Penceresi 9



Şekil 5.13. İmajı Hafıza Kartına Yazdırma Penceresi

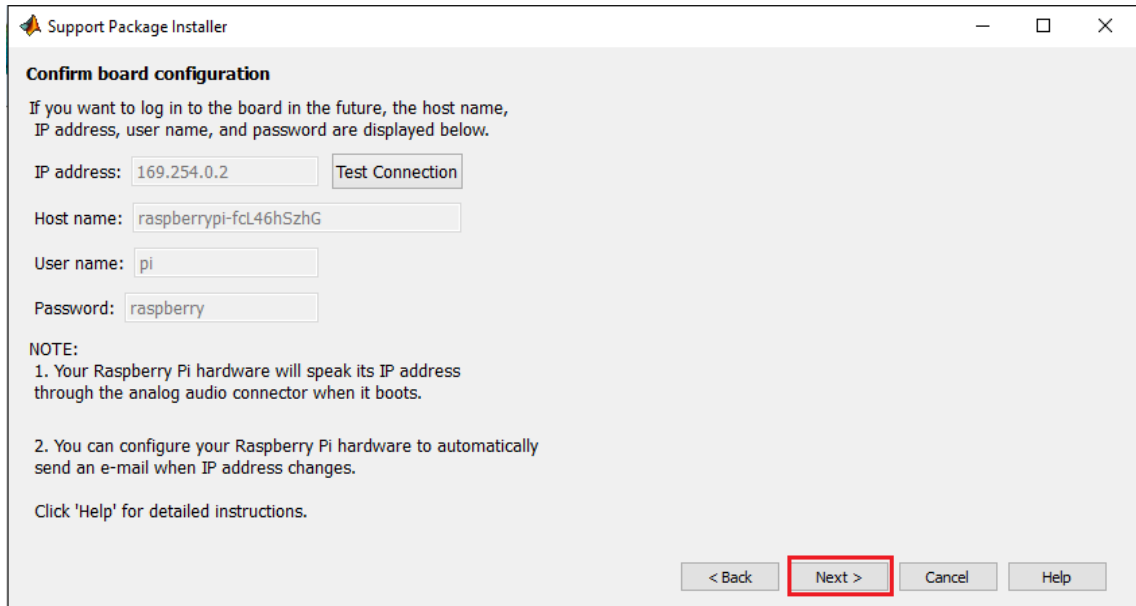
Hafıza kartına yazma işlemi tamamlandıktan sonra aşağıdaki resimde görünen pencere açılacaktır. Bu pencere açıldıktan sonra hafıza kartı raspberry pi'ye takılır, ethernet kablosuyla bilgisayara bağlanır ve raspberry pi'ye enerji verilir, raspberry pi üzerindeki PWR ve ACT ledleri yandıktan sonra next butonuna basılır.



Şekil 5.14. Matlab Support Pakgace Installer Penceresi 10

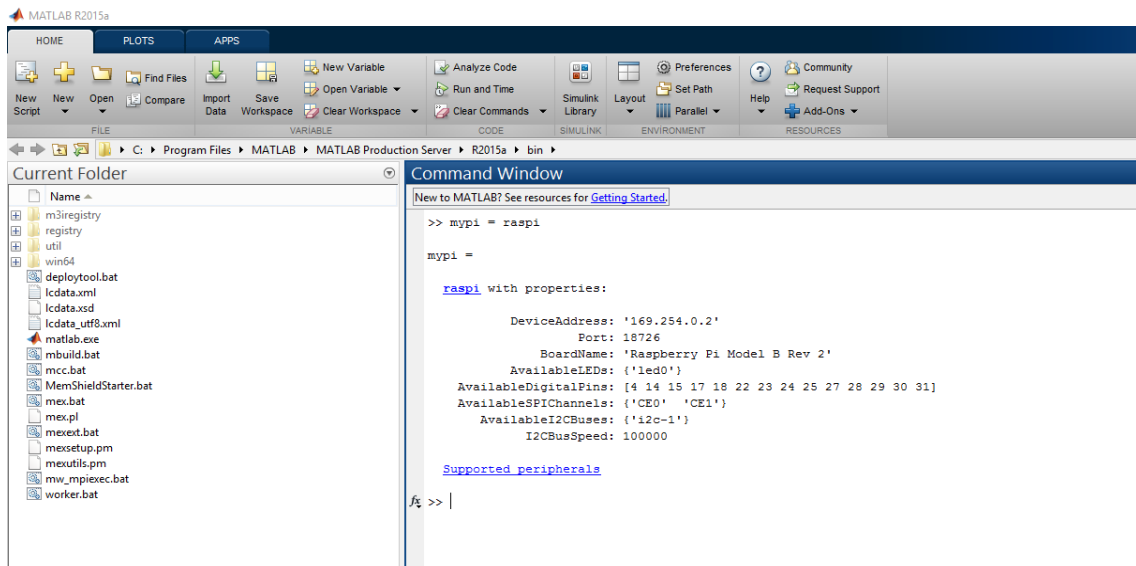
Ağ bağlantısı konfigure edildikten sonra aşağıda resimde görünen pencere açılacaktır. Bu pencerede raspberry pi'nin IP adresi, bağlantı adı, kullanıcı adı ve parola

bilgileri görülür next butonuna basıldıktan sonra kurulum ve bağlantı işlemi tamamlanmış olur.



Şekil 5.15. Matlab Support Pakgace Installer Penceresi 11

MATLAB’da “Command Window”da veya “Editor”de mypi=raspi komutuyla raspberry pi bağlantısı kurulur burada “mypi” raspberry pi’ye atana deışkendir, mypi yerine istenilen başka bir deışkende atanabilir (Örn: benimpi=raspi) bu işlem gerçekleştirildikten sonra program yazmaya başlanır.



Şekil 5.16. Matlab Editör Penceresi

6. DENEYLER

6.1. Deney 1: RASPBERRY Pİ İLE LED YAKMA

6.1.1. Amaç: GPIO kullanımını öğrenme ve python ile program yazma

Kullanılacak Malzemeler:

- Raspberry pi
- 1 adet led
- 1 K Ω direnç

6.1.2. Genel Bilgiler:

Raspberry pi üzerinde GPIO (General Purpose Input-Output) bulundurması sebebiyle elektronik sistemlerle haberleşmede kullanılabilirdiği için elektronik ve mekatronik alanlarında popülerlik kazanmıştır bu deneyde GPIO'lar yardımıyla led yakıp söndürerek GPIO'ların kullanımı hakkında bilgi edinilecektir.

GPIO (General Purpose Input-Output): Entegre sistemlerde temel veri giriş ve çıkışı yapılabilen, ve de bunların kontrol edilip okunabildiği sisteme denir. Bu sistem sayesinde elektronik devreler kontrol edebilir, entegre donanımlara sisteme çevre birimler bağlanabilir, kontrol edebilir ve de kombine olarak kullanılabilir. GPIO'lar C , Python gibi yüksek seviye dillerle kontrol edilip programlar yazılabilir. GPIO pin'leri aksi belirtilmedikçe 3.3V gerilim ile çalışır, ve de yüksek akım koruması yoktur. Bu pin'lere yüksek değerde akım uygulanırsa cihaza kalıcı olarak zarar verilebilir.

Raspberry Pi GPIO Şeması

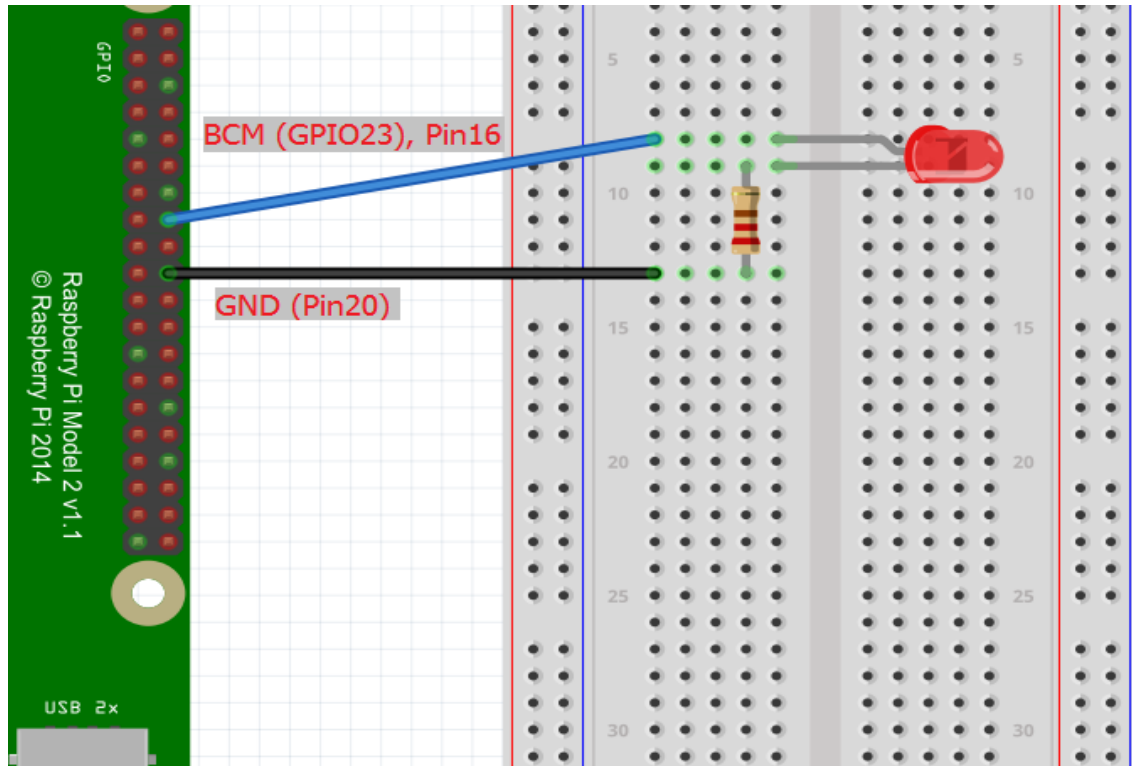
Raspberry Pi2 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power	⬇️	DC Power 5v	02
03	GPIO02 (SDA1 , PC)	⬇️	DC Power 5v	04
05	GPIO03 (SCL1 , PC)	⬇️	Ground	06
07	GPIO04 (GPIO_GCLK)	⬇️	(TXD0) GPIO14	08
09	Ground	⬇️	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	⬇️	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	⬇️	Ground	14
15	GPIO22 (GPIO_GEN3)	⬇️	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	⬇️	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	⬇️	Ground	20
21	GPIO09 (SPI_MISO)	⬇️	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	⬇️	(SPI_CE0_N) GPIO08	24
25	Ground	⬇️	(SPI_CE1_N) GPIO07	26
27	ID_SD (PC ID EEPROM)	⬇️	(PC ID EEPROM) ID_SC	28
29	GPIO05	⬇️	Ground	30
31	GPIO06	⬇️	GPIO12	32
33	GPIO13	⬇️	Ground	34
35	GPIO19	⬇️	GPIO16	36
37	GPIO26	⬇️	GPIO20	38
39	Ground	⬇️	GPIO21	40

Rev. 1
26/01/2014
http://www.element14.com

Yandaki şemada pinler, pin sıralamalarına göre alınır (Örnek: GPIO16) BOARD sıralaması kullanılmış olur, pin isimlerine göre alınır (Örnek GPIO23) BCM yani işlemci üzerindeki sıralaması kullanılmış olur (burada GPIO16 ve GPIO23 aynı pini temsil etmektedir.) programlama yaparken hangi sıralamanın seçildiğine dikkat ederek GPIO'lar kodlanmalıdır. GPIO'ların isimlerinin yanında parantez içinde belirtilenlerse o pinin SPI ve I2C gibi sistemler için kullanılıp kullanılmayacağını belirtmektedir.

Şekil 6.1. Raspberry Pi GPIO Şeması

6.1.3. Deney Devresi



Şekil 6.2. 1. Deney Devresi

Deneyin Yapılışı

Raspberry pi'yi çalıştırdıktan sonra yukarıdaki devreyi breadboard üzerine kurun ve raspberry ile bağlantılarını yapın daha sonra terminalden “sudo nano ledyakma.py” (“ledyakma” yazılan kısım oluşturmak istediğiniz python dosyasının ismidir istediğiniz ismi verebilirsiniz “.py” kısmı python uzantısıdır.) komutuyla python shell'i açın açılan pencerede aşağıdaki kodları girip kaydedin.

6.1.4. Kod:

```
#!/usr/bin/env python
```

```
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
```

```
GPIO.setmode(GPIO.BOARD)
```

```
# GPIO isimlendirmesi board üzerindeki numaralandırma
```

```
GPIO.setup(16, GPIO.OUT) # 16 numaralı pini çıkış olarak ayarladık
```

```
try: # Try - except bloğu programın ctrl+c komutuyla sonlandırılması için eklendi
```

```
    while True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
```

```
        GPIO.output(16, True) # 16 numaralı pini aktif ettik lojik bir gönderdik
```

```
except KeyboardInterrupt:
```

```
    GPIO.cleanup() # Pinler bir sonraki kullanım için temizlendi
```

5.1.5. Rpi.GPIO as GPIO Kütüphanesi

RPi.GPIO as GPIO kütüphanesi GPIO'ların kontrol edilebilmesi için oluşturulmuş bir python kütüphanesidir. Bu kütüphane GPIO ları giriş, çıkış olarak ayarlama GPIO'ya lojik 1 yada 0 gönderme gibi temel işlem komutlarını içerir.

- GPIO.setmode(GPIO.BOARD) komutu ile GPIO isimlendirmelerinin board sıralamasına göre yapılması sağlanır.
- GPIO.setmode(GPIO.BCM) komutu ile GPIO isimlendirmelerinin BCM (işlemci) sıralamasına göre yapılması sağlanır.
- GPIO.setup(A, B) bu komut pini çıkış yada giriş olarak ayarlamaya yarar. Bu komut kullanılırken A yerine ayarlanacak pin numarası (örnek:16), B yerine pinin giriş pini veya çıkış pini olarak kullanılma durumu (örnek:GPIO.OUT çıkış olarak ayarlar, GPIO.IN giriş olarak ayarlar.) girilir.
- GPIO.output(A, B) bu komut çıkış olarak ayarlanan pine lojik 1 yada 0 yollamaya yarar. Bu komut kullanılırken A yerine çıkış olarak ayarlanan pinin numarası (örnek:16), B yerine çıkışa gönderilecek lojik 1 yada 0 olma durumu (örnek:True lojik 1 durumu, False lojik 0 durumu) girilir.

Time (zaman) kütüphanesi de eklenerek kod tekrar yazılırsa belirtilen zaman süresince ledin yandığı sonra söndüğü gözlenecektir.

6.1.6. Time kütüphanesi kullanıldığında program kodu:

```
#!/usr/bin/env python
```

```
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
```

```
import time # Time kütüphanesi çağırılır
```

```
GPIO.setmode(GPIO.BOARD) # GPIO isimlendirmesi board üzerindeki numaralandırma
```

```
GPIO.setup(16, GPIO.OUT) # 16 numaralı pini çıkış olarak ayarladık
```

```
GPIO.output(16, True) # 16 numaralı pini aktif ettik lojik bir gönderdik
```

```
time.sleep(5) # Bir üstteki komutu 5 saniye boyunca çalıştırılır
```

6.1.7. Time Kütüphanesi:

Time kütüphanesi komutun belirli bir süre boyunca çalışması için kullanılan bir kütüphanedir.

- `time.sleep(A)` bu komutta A yazılan yere zaman girilmektedir. (örnek: 5) A yerine girilen değer saniye olarak belirlenmelidir.

6.2. Deney 2: BUTON KULLANIMI

6.2.1. Amaç: Buton kullanımını ve pull-up, pull-down olayını anlama

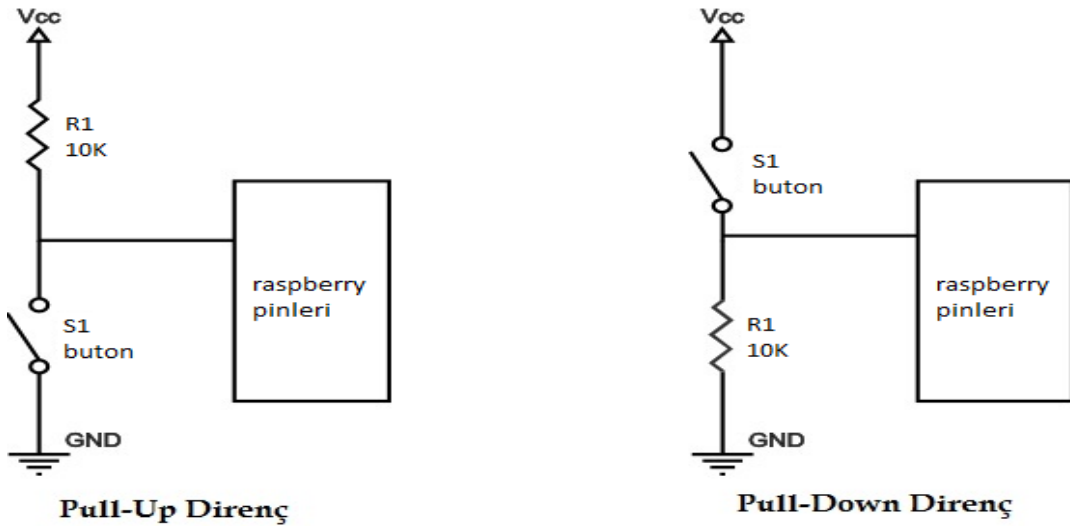
Kullanılacak Malzemeler:

- 2 adet buton
- 2 adet 10 K Ω direnç

6.2.2. Genel Bilgiler:

Raspberry pi de buton kullanılırken pull-up direnç ve pull-down direnç olaylarından bahsedilebilir. Bu deneyde pull-up, pull-down dirençlerini kullanarak ve kullanmadan buton kontrol etme öğrenilecektir.

Pull-up ve Pull-down Dirençleri Kullanılarak Buton Kontrol Etme

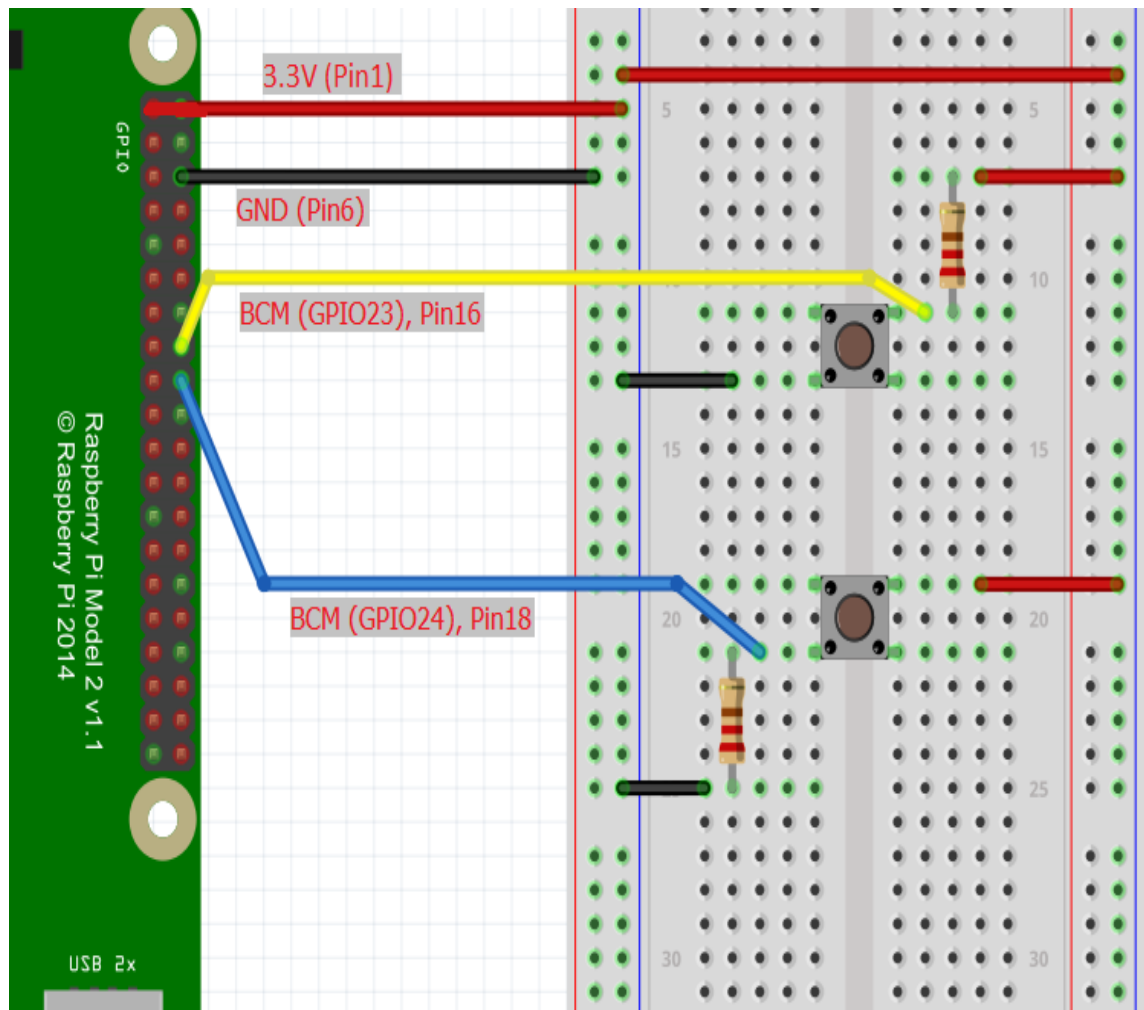


Şekil 6.3. Pull Up, Pull Down Direnç Devreleri

Pull-up direnç: Yukarıdaki şekilde gösterildiği gibi besleme>direnç>pin>buton>toprak şeklindeki bağlantı biçimine pull-up direnç adı verilmektedir. Burada butona basılmazken (devre açık durumda) besleme gerilimi direk raspberry pininde görülmektedir bu durumda raspberry giriş olarak ayarlanmış pininde lojik 1 görür ve yazılan programa göre butona basılmamış olduğunu anlar. Butona basıldığında ise besleme gerilimi pini görmeden anahtar üzerinden toprağa gitmektedir bu durumda raspberry giriş olarak ayarlanan pininde lojik 0 görür ve yazılan programa göre anahtara basılmış olduğunu anlar.

Pull-down direnç: Yukarıdaki resimde gösterildiği gibi besleme>buton>pin>direnç >toprak şeklinde bağlantı biçimine pull-down direnç adı verilmektedir. Burada butona basılmazken (devre açık durumda) raspberry, giriş olarak ayarlanmış pininde lojik 0 görür ve yazılan programa göre butona basılmamış olduğunu anlar. Butona basıldığında ise besleme gerilimi 10 K Ω direnç yüzünden toprağa gidemez raspberrynin pinine gider bu durumda raspberry giriş olarak ayarlanan pininde lojik 1 görür ve yazılan programa göre butona basılmış olduğunu anlar.

6.2.3. 1. Deney Devresi



Şekil 6.4. 2. Deneyin 1. Devresi

Yukarıdaki deney devresinde mavi kablolar raspberry'nin 1 numaralı pinine (3.3V) bağlı besleme kablolarıdır.

Siyah kablolar raspberry'nin 6 numaralı pinine (GND) bağlıdır.

Sarı kablo 16 numaralı pine (BOARD sıralamasına göre GPIO16, BCM sıralamasına göre GPIO23) bağlıdır.

Gri kablo 18 numaralı pine (BOARD sıralamasına göre GPIO18, BCM sıralamasına göre GPIO24) bağlıdır.

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python butonkontrol.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.2.4. Kod:

```
#!/usr/bin/env python
```

```
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
```

```
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
```

```
GPIO.setup(23, GPIO.IN) # 23 numaralı pin giriş pini olarak ayarlandı
```

```
GPIO.setup(24, GPIO.IN) # 24 numaralı pin giriş pini olarak ayarlandı
```

```
try: # Try - except bloğu programın ctrl+c komutuyla sonlandırılması için eklendi
```

```
    While True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
```

```
        if (GPIO.input(23) == False ): # 23 numaralı pinin lojik 0 olma durumu koşulu
```

```
            print (“Birinci butona basildi.”) # Koşul sağlandığında ekran çıktısı
```

```
        time.sleep(1)
```

```
        if (GPIO.input(24) == True ): # 24 numaralı pinin lojik 1 olma durumu koşulu
```

```
            print (“İkinci butona basildi”) # Koşul sağlandığında ekran çıktısı
```

```
        time.sleep(1)
```

```
            time.sleep(2)
```

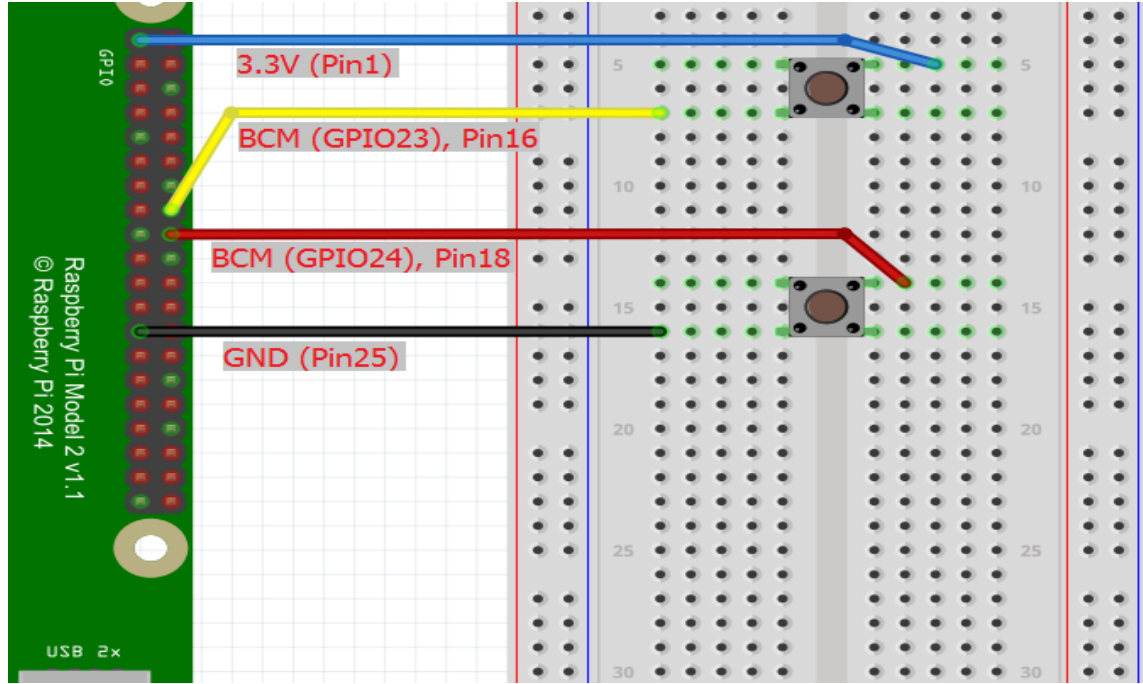
```
except KeyboardInterrupt:
```

```
    GPIO.cleanup() # Pinler bir sonraki kullanım için temizlendi
```

Direnç Kullanmadan Buton Kontrol Etme

Buton kullanımını dirençleri kullanmadan da yapmak mümkündür burada pull-up, pull-down olayı kodlarla yapılır.

6.2.5. 2. Deney devresi



Şekil 6.5. 2. Deneyin 2. Devresi

Yukarıdaki deney devresinde siyah kablolarla raspberry'nin 25 numaralı pinlerine (GND) bağlıdır.

Mavi kablo 16 numaralı pine (BOARD sıralamasına göre GPIO16, BCM sıralamasına göre GPIO23) bağlıdır.

Sarı kablo 18 numaralı pine (BOARD sıralamasına göre GPIO18, BCM sıralamasına göre GPIO.24) bağlıdır.

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python butonkontrol2.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.2.6. Kod:

```
#!/usr/bin/env python
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
# 23 numaralı pin yüksek voltaj beklemeye ayarlandı
GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)
# 24 numaralı pin düşük voltaj beklemeye ayarlandı
try: # Try - except bloğu programı ctrl+c komutuyla sonlandırılması için eklendi
    While True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
        if (GPIO.input(23) == True ): # 23 numaralı pinin lojik 1 olma durumu koşulu
            print ("Birinci butona basıldı.") # Koşul sağlandığında ekran çıktısı
            time.sleep(1)
        if (GPIO.input(24) == False ): # 24 numaralı pinin lojik 0 olma durumu koşulu
            print ("İkinci butona basıldı") # Koşul sağlandığında ekran çıktısı
            time.sleep(1)
        time.sleep(2)
except KeyboardInterrupt:
    GPIO.cleanup() # Pinler bir sonraki kullanım için temizlendi
```

NOT: # işaretinden sonraki kısımlar komutların açıklamalarıdır, programa dahil değildir.

6.3. Deney 3: ANAHTAR KULLANIMI

6.3.1. Amaç: Anahtar kullanımını öğrenme

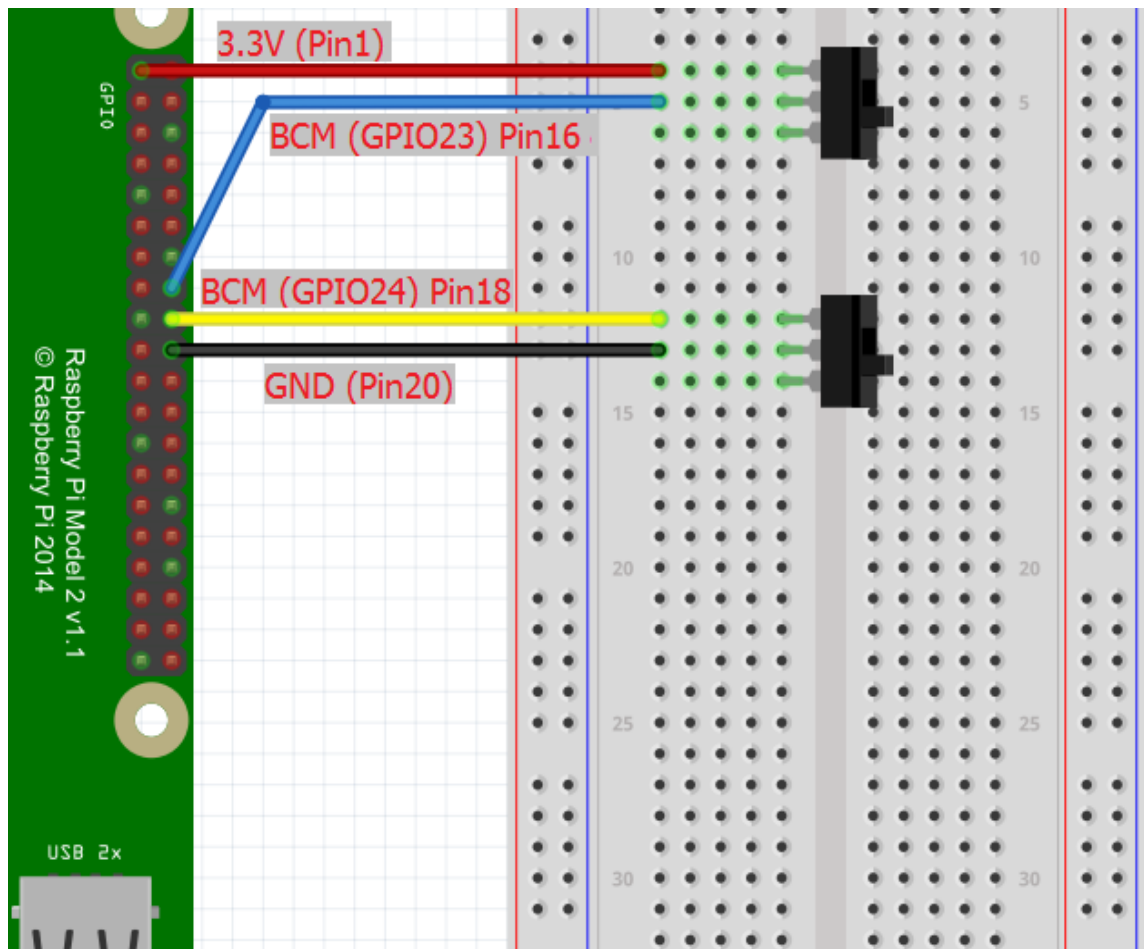
Kullanılacak Malzemeler:

- 2 adet anahtar

6.3.2. Genel Bilgiler:

Raspberry pi de anahtar kullanılırken devrenin tamalanması butondaki gibi anlık olmadığı için burada bir edge (kenar) yakalayıp o edge'nin değişimine göre anahtarın çalışmasını programlamalıyız. Edge olarak bahsedilen olay düşükten yükseğe veya yüksekte düşüğe bir elektrik sinyalinin durum değişimidir, anahtar kullanmak için yazacağımız programda girişin değerinden çok durumundaki değişime ihtiyaç duyarız.

6.3.3. Deney Devresi



Şekil 6.6. 3. Deney Devresi

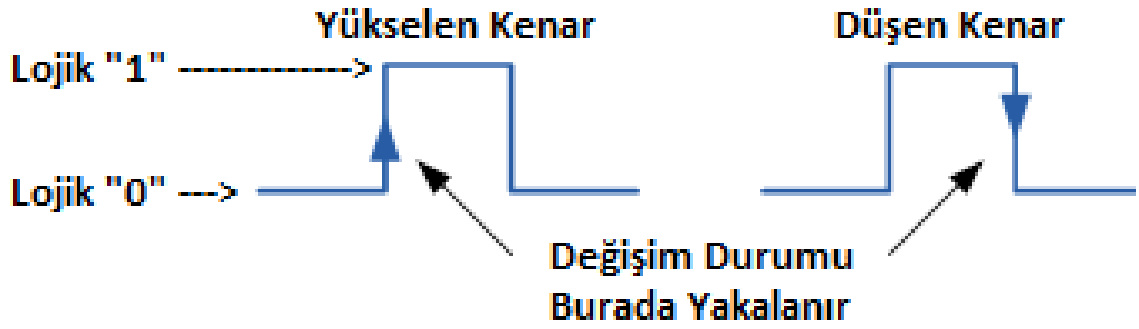
Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python anahtarkontrol.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.3.4. Kod:

```
#!/usr/bin/env python
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
GPIO.setup(18, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
# 18 numaralı pin yüksek voltaj beklemeye ayarlandı
GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)
# 24 numaralı pin düşük voltaj beklemeye ayarlandı
try: # Try - except bloğu programı ctrl+c komutuyla sonlandırılması için eklendi
    While True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
        GPIO.wait_for_edge(18, GPIO.RISING)
# 18 numaralı pinde yükselen kenar belirlenmesi
        print("Birinci buton devreyi tamamladı")
        GPIO.wait_for_edge(18, GPIO.FALLING)
# 18 numaralı pinde düşen kenar belirlenmesi
        print("Birinci buton devresi koptu")
        GPIO.wait_for_edge(24, GPIO.FALLING)
# 24 numaralı pinde düşen kenar belirlenmesi
        print("İkinci buton devreyi tamamladı")
        GPIO.wait_for_edge(24, GPIO.RISING)
# 24 numaralı pinde yükselen kenar belirlenmesi
        Print("İkinci buton devresi koptu")
except KeyboardInterrupt:
    GPIO.cleanup() # Pinler bir sonraki kullanım için temizlendi
```

Yukarıdaki deneyde 18 numaralı pin pull-down ve 24 numaralı pin pull-up direnç olarak ayarlanmıştır ve pinler GPIO.wait_for_edge komutuyla düşen kenar (GPIO.FALLING) veya yükselen kenar (GPIO.RISING) algılanması için ayarlanmıştır. GPIO.wait_for _edge komutu bir kenar değişimi durumu algılanana kadar programın yürütülmesini engellemek için tasarlanmıştır. GPIO.RISING (yükselen kenar) ve GPIO.FALLING (düşen kenar) kodlarıyla kenar (edge) değişimi belirlenir.



Şekil 6.7. Yükselen Kenar – Düşen Kenar Durumu

6.4. Deney 4: PWM İLE LED SÜRME

6.4.1. Amaç: PWM ne olduğunu ve raspberry de pwm kullanımını öğrenme

Kullanılacak Malzemeler:

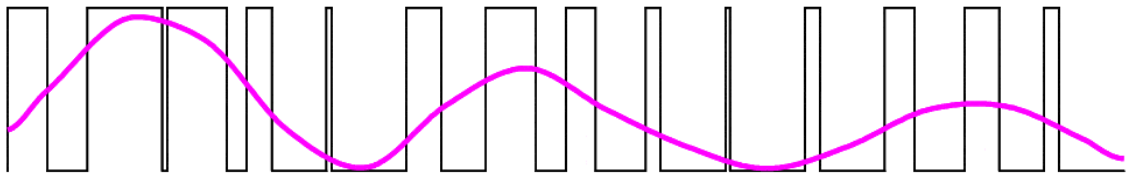
- 1 adet led
- 1 adet $1K\Omega$

6.4.2. Genel Bilgiler:

Raspberry pi de PWM ile led kontrol edilirken PWM sinyalinin görev periyodundaki değişimden dolayı led'in yavaş yavaş yanıp sönmeye olayı gözlemlenecektir.

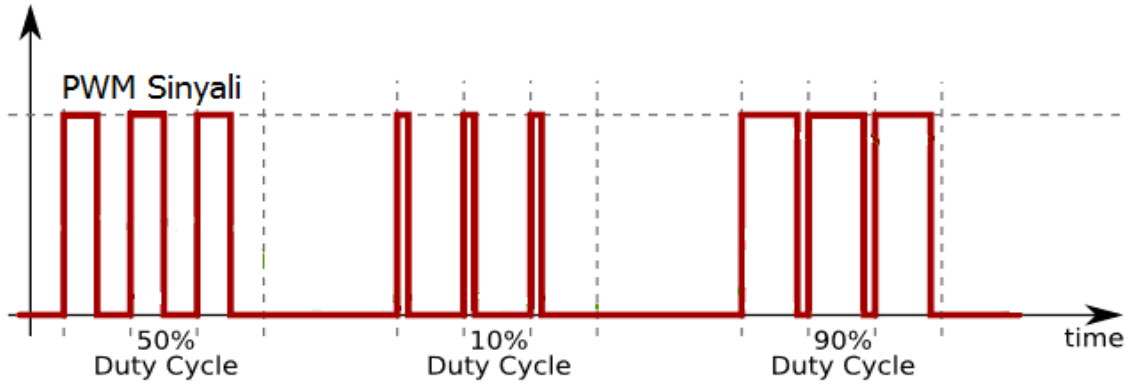
PWM: PWM (Pulse Width Modulation), analog bileşenlerin dijital sinyallerle kontrol edilmesini sağlayan bir yoldur. Sinyalin frekansının belirlenen değerlerde değiştirilmesi ile dijital sinyalin analog sinyali örnekleme gibi de düşünülebilir. Analog sinyal bir sinüs eğrisi gibi dalgalıdır, fakat dijital sinyal sadece "1" ve "0" lardan oluşmaktadır. PWM yönteminde "1" ve "0" lar kullanılarak analog sinyale yakın bir sinyal elde edilmeye çalışılır.

PWM basitçe belli frekanstaki dijital bir akımı on ve off duruma getirerek analog sinyale benzetmeye çalışmak olarak ifade edilebilir.



Şekil 6.8. PWM Sinyali Örneği 1

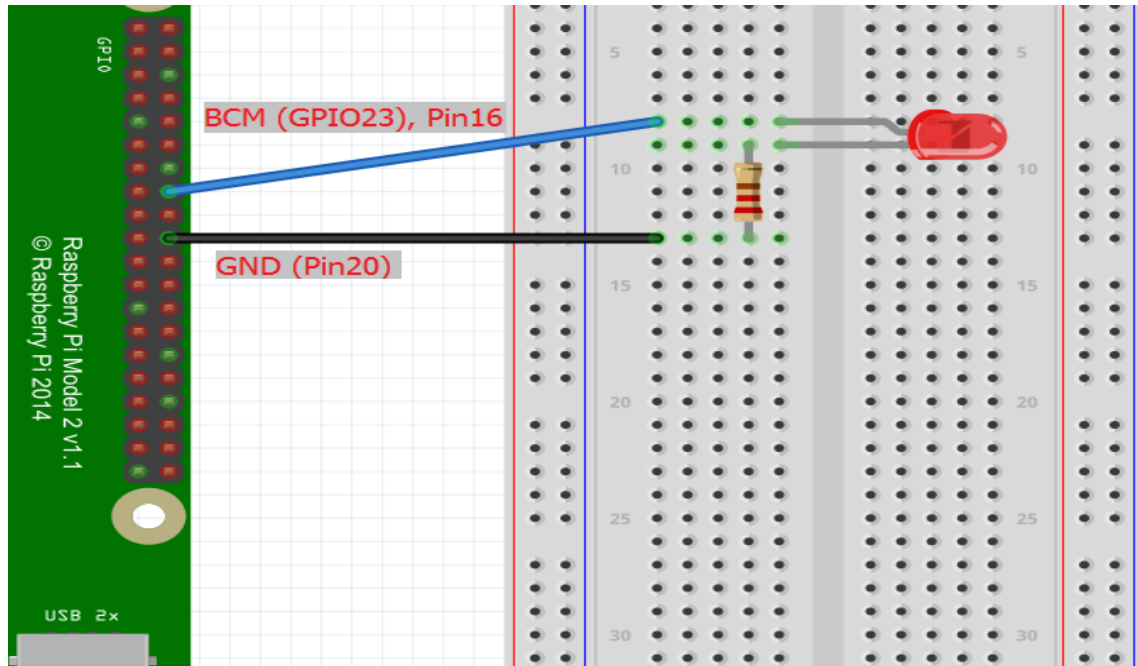
Yukarıdaki şekilde görüldüğü gibi dijital sinyalin "1" olma durumunun, miktarı değiştirilerek, analog sinyal dijital sinyal ile simüle edilir. PWM sinyalinin temelinde görev periyodu (duty cycle) yatar. Görev periyodu 100 birimlik bir cycle'da sinyalin "1" olma miktarını temsil etmektedir (Örnek: %10 duty cycle 1 saniyelik bir periyodun %10'unda sinyalin "1" durumunda geri kalanında "0" durumunda olduğunu belirtir).



Şekil 6.9. PWM Sinyali Örneği 2

Bu deneyde PWM sinyalinin görev periyodu değişiminden faydalanılarak led yakılacaktır. Burada led'in yavaşça yanıp söndüğü gözlenecektir. Bu işlem PWM sinyalinin görev periyodunu “%0” dan başlatarak “%100”e saniyenin “%2”si süresinde 1’er birimlik artışlarla, aynı şekilde “%100”den de tekrar “%0”a saniyenin “%2”si süresinde 1’er birimlik azalmalarla gerçekleştirilecektir. Burada led'in yavaşça yanıp söndüğünü görmemiz, gözümüzün küçük görev periyodundaki yanıp sönmeleri algılayamamasından kaynaklanmaktadır.

6.4.3. Deney Devresi



Şekil 6.10. 4. Deney Devresi

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python pwmledsurme.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.4.4. Kod:

```
#!/usr/bin/python env
```

```
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
```

```
import time
```

```
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
```

```
GPIO.setup(23, GPIO.OUT) # 23 numaralı pin çıkış olarak atanır
```

```
p= GPIO.PWM(23, 50) # 23 numaralı pin “p” değişkenine 50 Hz’lik PWM sinyali olarak atanır (1.14 Hz ile 19.2 MHz arası)
```

```
p.start(0)
```

```
# Görev periyodu 0 olarak sinyal başlatılır.
```

```
(Buraya yazılan değer 0-100 arasında olmalıdır görev periyodunu temsil eder.)
```

```
try: # Try - except bloğu programı ctrl+c komutuyla sonlandırılması için eklendi
```

```
    while True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
```

```
        for i in range(100): # “i” değeri 0’dan 100’e saydırıldı
```

```
            p.ChangeDutyCycle(i) # “i” görev periyodu değişimine tanımlandı
```

```
            time.sleep(0.05) # Bir sonraki “i” değeri için 0.05sn bekler
```

```
        for i in range(100): # “i” değeri 0’dan 100’e saydırıldı
```

```
            p.ChangeDutyCycle(100-i) # “100-i” görev periyodu değişimine tanımlandı
```

```
            time.sleep(0.05) # Bir sonraki “i” değeri için 0.05sn bekler
```

```
except KeyboardInterrupt:
```

```
    pass
```

```
p.stop() # PWM sinyali durduruldu
```

```
GPIO.cleanup()# Pinler bir sonraki kullanım için temizlendi
```

Yukarıda ki deneyde belirli bir frekans deęerindeki PWM sinyalinin, görev periyodu deęişiminden yararlanılarak, led'in yavaşça yakılıp söndürülmesi sağlandı. Deneyin koduna bakıldığında 23 numaralı pin çıkış olarak ayarlandıktan sonra 50 Hz'lik bir sinyal üretecek şekilde "p" deęişkenine atandı. PWM sinyali "p.start(0)" komutuyla "0" görev periyoduyla başlatıldı. Görev periyodundaki deęişimi sağlamak için "for" döngüsünde "i" deęeri "0'dan 100'e" saydırıldı ve "i" deęeri görev periyodu deęişim fonksiyonuna atandı. "i" deęerinin bir sonraki deęeri için o anki i deęerinden sonra "0.05sn" beklendi. Raspberry pi de PWM pinin frekans aralığı 1.14 Hz ile 19.2 MHz arasındadır.

6.5. Deney 5: TRAFİK LAMBALARI

6.5.1. Amaç: Python'da fonsiyon oluřturmayı öğrenme ve ledleri trafik ışık kurallarına göre yakıp söndürmek.

Kullanılacak Malzemeler:

- 1 adet kırmızı led
- 1 adet sarı led
- 1 adet yeřil led
- 3 adet 1K Ω direnç

6.5.2. Genel Bilgiler:

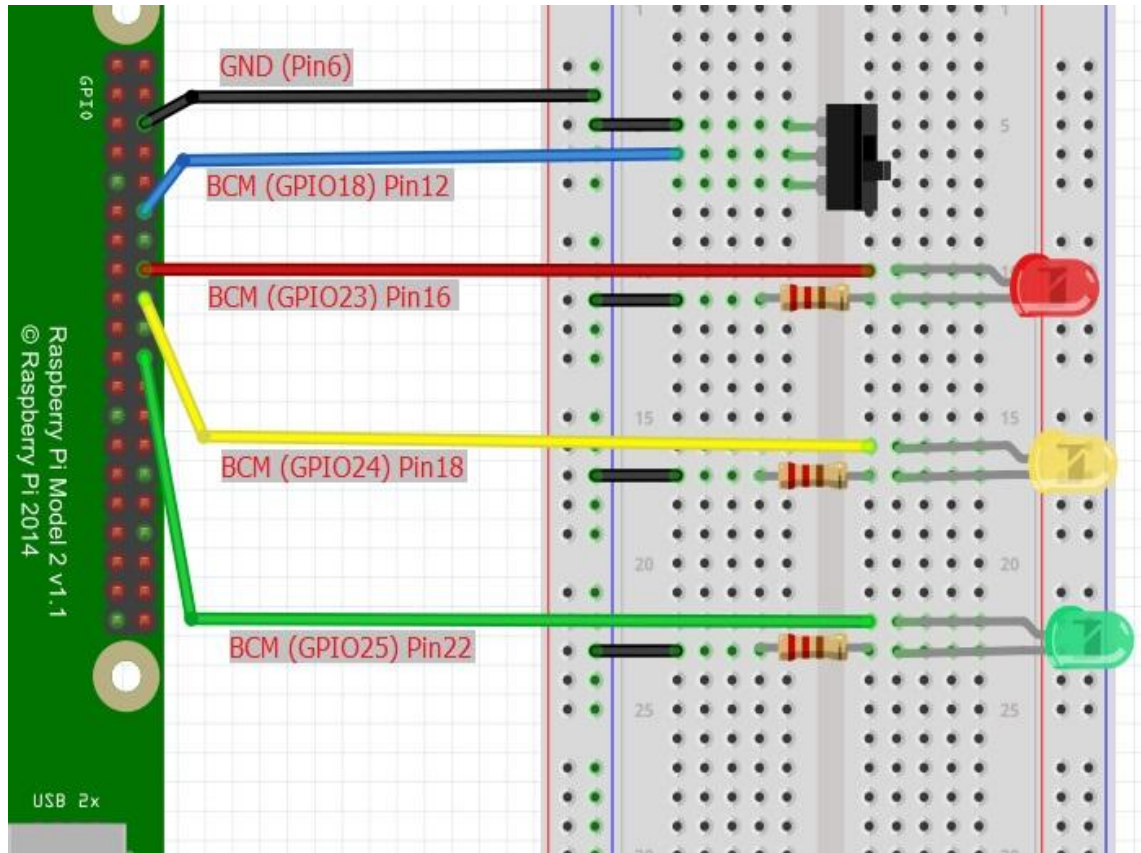
Bu deneyde bir anahtarla sistem çalıřtırılıp sonrasında trafik ışığı kurallarına göre ledlerin yanıp sönmeye saęlanacaktır. Ledlerin yanma sıraları ařaęıdaki kurallara göre gerçekteřtirilecektir.

- Kırmızıda duran araçlar dur'dan geç'e hazırlanırken önce kırmızı ve sarı led aynı anda yanar ve söndükten sonra yeřil led yanar.
- Yeřilde hareket halinde olan araçlar geç'ten dur'a hazırlanırken yeřil led söner, sarı led yanar ve sarı led söndükten sonra kırmızı led yanar.

Sistem anahtar vasıtası ile çalıřtırılır. Anahtar on durumuna alındığında kırmızı led 1 saniye boyunca yanar, ardından 1 saniye boyunca kırmızı ve sarı led beraber yanar, kırmızı ve sarı led söner ve ardından yeřil led 5 saniye boyunca yanar (Dur'dan geç'e hazırlanma durumu).

Yeřil led 5 saniye yandıktan sonra sistem dur'dan geç'e hazırlanma durumunu sonlandırır ve geç'ten dur'a hazırlanma durumu çalıřmaya bařlar, ilk önce yeřil led söner sonrasında sarı led 1 saniye boyunca yanar ve sarı led söner, sarı led söndükten sonra kırmızı led 5 saniye boyunca yanar (Geç'ten dur'a hazırlanma durumu), sistem sonsuz döngüde sırasıyla bu işlemleri tekrar tekrar gerçekteřtirir.

6.5.3. Deney Devresi



Şekil 6.11. 5. Deney Devresi

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python trafiklambalari.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.5.4. Kod:

```
#!/usr/bin/env python
```

```
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
```

```
import time # Zaman kütüphanesini çağırılır
```

```
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
```

```
anahtar = 18
```

```
kirmizi = 23
```

```
sari = 24
```

```
yesil = 25
```

```
GPIO.setup(anahtar, GPIO.IN, pull_up_down = GPIO.PUD_UP)
```

```

# Anahtarın bağlı olduğu 18 numaralı pin düşük voltaj beklemeye ayarlandı
GPIO.setup(kirmizi, GPIO.OUT) # 23 numaralı pin çıkış olarak ayarlandı
GPIO.setup(sari, GPIO.OUT) # 24 numaralı pin çıkış olarak ayarlandı
GPIO.setup(yesil, GPIO.OUT) # 25 numaralı pin çıkış olarak ayarlandı
def sifirla(): # GPIO pinlerini temizlemek için sıfırlama fonksiyonu oluşturuldu
    GPIO.output(kirmizi,GPIO.LOW)
    GPIO.output(sari,GPIO.LOW)
    GPIO.output(yesil,GPIO.LOW)
def kirmizidanYesile(): # Dur'dan geç'e hazırlanma fonksiyonu oluşturuldu
    print('kirmizidan yesile donuyor..')
    sifirla()
    GPIO.output(kirmizi,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(sari,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(kirmizi,GPIO.LOW)
    GPIO.output(sari,GPIO.LOW)
    GPIO.output(yesil,GPIO.HIGH)
def yesildenKirmiziya(): # Geç'ten dur'a hazırlanma fonksiyonu oluşturuldu
    print('yesilden kirmiziya donuyor..')
    sifirla()
    GPIO.output(yesil,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(yesil,GPIO.LOW)
    GPIO.output(sari,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(sari,GPIO.LOW)
    GPIO.output(kirmizi,GPIO.HIGH)
try: # Try - except bloğu programı ctrl+c komutuyla sonlandırılması için eklendi

```

```

sifirla() # Pinler temizlenir
while True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
    GPIO.wait_for_edge(anahtar, GPIO.FALLING)
# 18 numaralı pinde düşen kenar belirlenmesi
    while True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
        kirmizidanYesile() # Dur'dan geç'e hazırlanma fonksiyonu çalıştırıldı
        time.sleep(5)
        yesildenKirmiziya() # Geç'ten dur'a hazırlanma fonksiyonu çalıştırıldı
        time.sleep(5)
except KeyboardInterrupt:
    print('ctrl+c yapmis')
except:
    print('Bir hata var!')
finally:
    GPIO.cleanup()

```

Yukarıdaki deneyde trafik lambası için belirlenen koşullara göre ledlerin yanıp sönmeye sağlandı. Python kodu hazırlanırken dur'dan geç'e hazırlanma, geç'ten dur'a hazırlanma durumları için ve pinleri temizlemek için fonksiyonlar oluşturuldu ve sonsuz döngüde sırayla işletildi.

6.6. Deney 6: FOTO DİRENÇ'TEN DEĞER OKUMA

6.6.1. Amaç: Fotor direnç'in çalışma mantığını öğrenme

Kullanılacak Malzemeler:

- 1 adet foto direnç (LDR)
- 1 adet $1\mu F$ kondansatör

6.6.2. Genel Bilgiler:

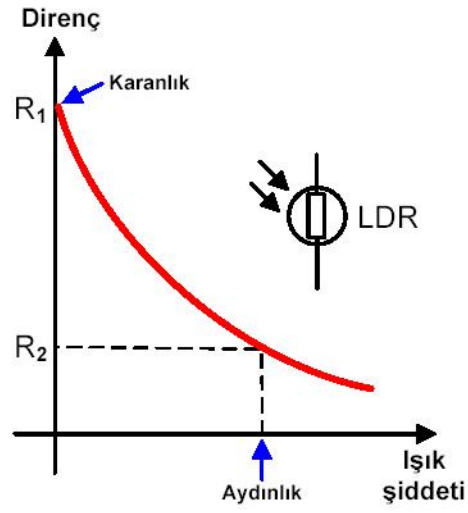
Foto direnç ışık şiddetine göre direnç değeri değişen bir devre elemanıdır. Ortamdaki ışığın şiddetine göre üzerine düşen direnç değerini ters orantılı olarak ayarlayabilen en basit optik sensör çeşididir. Üzerine ışık düştüğünde direnç değeri minimuma iner, karanlıkta ise maksimuma çıkar.

Deneyde kullanılan foto direnç analog bir devre elemanıdır fakat raspberry pi üzerinde analog sinyal girişi bulunmamaktadır, bu nedenle analog bir sensör kullanabilmek için analog dijital çeviriciye (analog digital converter “ADC”) ihtiyaç duyulur.

Bu deneyde analog dijital çevirici kullanmadan, bir kondansatör yardımıyla foto direnç'ten değer okunacaktır.

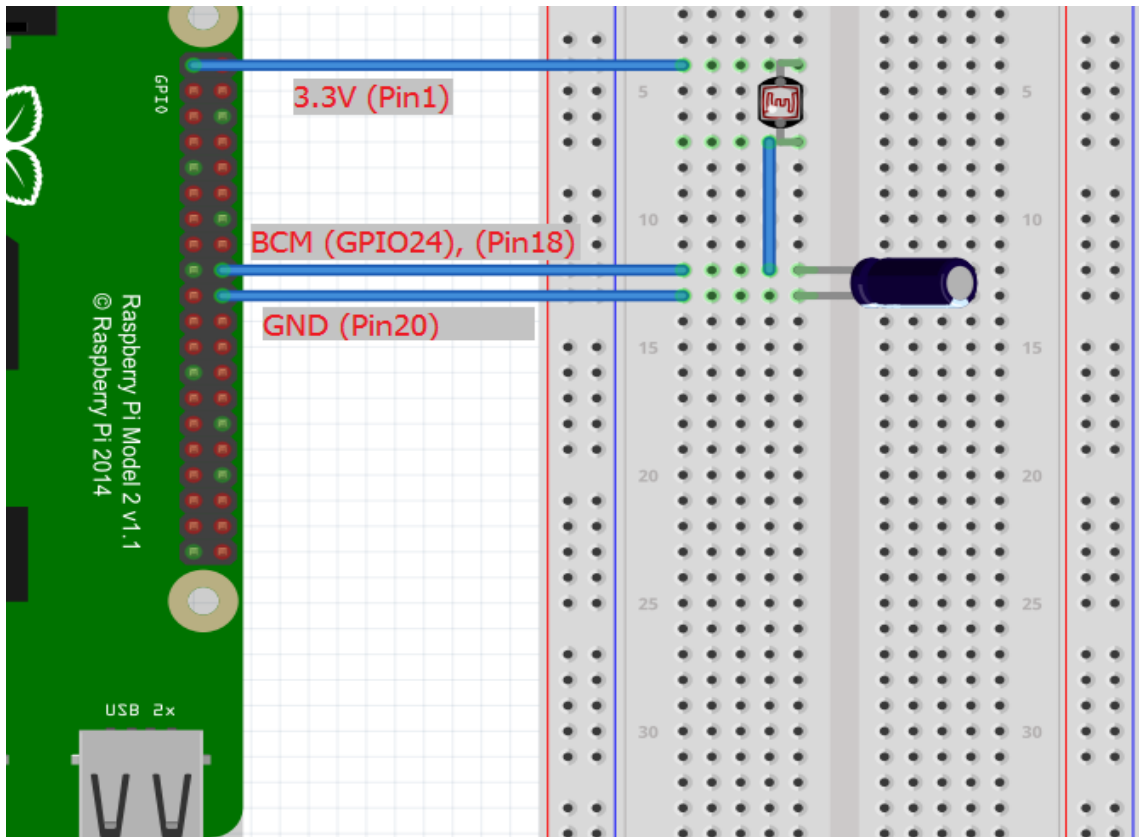
Foto Direnç Özellikleri:

- Üzerine düşen ışık şiddeti ile direnç değeri ters orantılı olarak değişir.
- Direnç değişimi lineer değildir.
- AC ve DC akımda aynı özellikleri gösterir.
- Yapısında kadmiyum sülfat” ($CdSO_4$) yarı iletken madde bulunur.
- Direnç değerleri karanlıkta mega ohm seviyelerindeyken aydınlıkta $5\ \Omega - 10\ \Omega$ değerlerine düşer.



Şekil 6.12. LDR Direnç Grafiği

6.6.3. Deney Devresi



Şekil 6.13. 6. Deney Devresi

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python fotodirenç.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.6.4. Kod:

```
#!/usr/bin/env python
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
import time # Zaman kütüphanesini çağırılır
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
def sensoroku(pin): # Foto direnç'ten deger okumak için oluşturulan fonksiyon
    sayac = 0
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, False)
    time.sleep(0.1)# Analog deđer okuma hızı 0.1 saniye 10 Hz
    GPIO.setup(pin, GPIO.IN)
    while (GPIO.input(pin) == False):
        sayac += 1
    return sayac
try:
    while True:
        print sensoroku(23)
except KeyboardInterrupt:
    GPIO.cleanup()
```

Deneyde kondansatör yardımıyla direnç deđerindeki deęişimi ekrana bastırdık. Devrede geçen akım kondansatörde birikecek, ve de birikene kadar 0.1 saniye aralıklarla kontrol edilecek ve eđer giriş deđerı “0” a eşitse ekrana çıktı verilecek. Burada foto direnç'in ışık durumuna göre direnç deęişimi akım deđerini de etkiler, ışığa göre akımın birikme süresi deęişeceęinden ışık deđerini sayaç referansından okumuş oluruz.“time.sleep(0.1)” komutuyla sistem 0.1 saniyede denetlenerek sistemin çalışma frekansı 10 Hz olarak ayarlanmıştır. Bu deđer bize analog deđer okuma hızını verir.

6.7. Deney 7: YÜZ ALGILAMA

6.7.1. Amaç: MATLAB programı ile raspberry pi ve kamera modülünü kontrol ederek yüz algılama

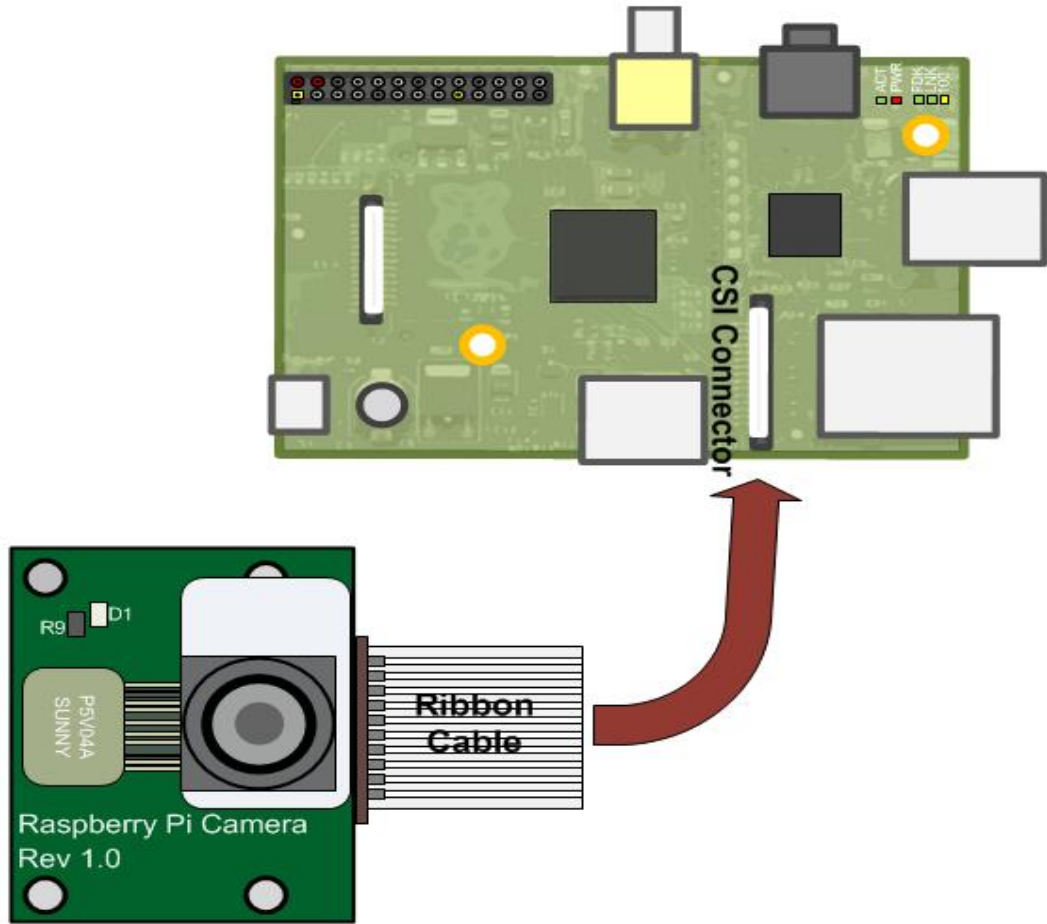
Kullanılacak Malzemeler:

- Raspberry Pi kamera modülü
- Raspberry Pi

6.7.2. Genel Bilgiler:

Raspberry pi de CSI konnektörü bulunmaktadır bu sayade kamera modülünü bağlayarak fotoğraf ve video çekebilmek mümkündür. Bu deneyde kamera modülüyle MATLAB programında yüz algılama işlemi yapılacaktır. Yüz algılama (face detection) görüntüde bulunan yüzleri tanımlama işlemidir. Yüz algılama işlemi yüz tanıma (face recognition) işleminin en önemli ara basamaklarından biridir.

6.7.3. Deney Devresi



Şekil 6.14. 7. Deney Devresi

Şekilde görüldüğü gibi kamera modülü raspberry pi'ye bağlanır ve aşağıdaki matlab kodları editörde çalıştırılır.

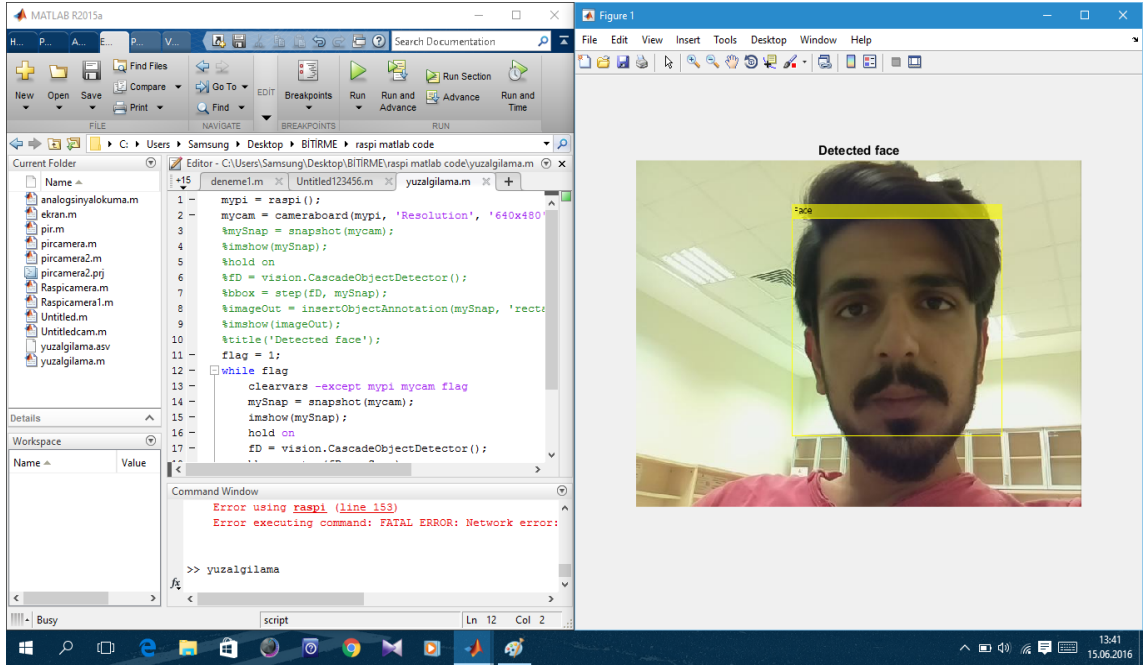
6.7.4. KOD:

```
mypi = raspi();  
mycam = cameraboard(mypi, 'Resolution', '640x480');  
flag = 1;  
while flag  
    clearvars –except mypi mycam flag  
    mySnap = snapshot(mycam);  
    imshow(mySnap);  
    hold on  
    fD = vision.CascadeObjectDetector();  
    bbox = step(fD, mySnap);  
    imageOut = insertObjectAnnotation(mySnap, 'rectangle', bbox, 'Face');  
    imshow(imageOut);  
    title('Detected face');  
    drawnow  
end
```

Not:

Raspberry pi'yi matlab ile kullanabilmek için matlab'ın R2015a ve üzeri sürümlerinin kullanılması gerekmektedir.

Bu deneyi gerçekleştirebilmek için “computer vision toolbox” ve “image processing toolbox” kütüphanelerinin matlab'da kurulu olması gerekmektedir.



Şekil 6.15. Yüz Algılama Örneği

Deney gerçekleştirildiğinde yukarıdaki resimde ki gibi bir pencere açılacaktır, açılan pencerede program kamera açısı içindeki yüzleri bularak çerçeve içerisine alır.

6.8. Deney 8: L298N MOTOR SÜRÜCÜ KARTI İLE MOTOR KONTROL

6.8.1. Amaç: L298N motor sürücü kontrol kartını ve motor kontrol etmeyi öğrenme

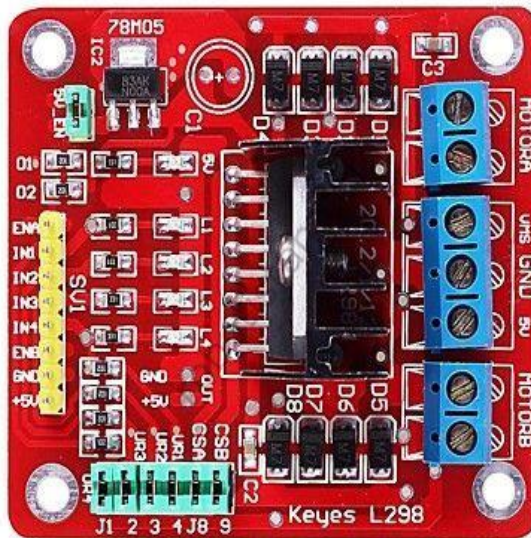
Kullanılacak Malzemeler:

- L298N motor sürücü kartı
- 5V DC motor

6.8.2. Genel Bilgiler:

Raspberry pi GPIO pinleri maksimum 40 mA civarında akım vermektedir. Bu nedenle 40 mA'den çok akımla çalışan cihazlar direkt olarak GPIO pinleri ile kontrol edilemez.

Deneyde motor direkt pinlere bağlanamadığı için L298N motor sürücü kartı ile kontrol edilir. Motor sürücü kartı yardımıyla motor dışarıdan bir kaynak ile beslenir ve GPIO pinleri aracılığıyla motor sürücü kartı kontrol edilir. L298N motor sürücü kartı ile iki adet motor kontrol edilebilir. Aşağıdaki şekilde L298N motor sürücü kartı ve bağlantı şekli gösterilmiştir.



VMS: Motor besleme girişi “+”

GND: Motor besleme girişi “-”

MOTORA: 1. Motor kanalı

MOTORB: 2. Motor kanalı

ENA: 1. Motor kanalını aktif etme pini

ENB: 2. Motor kanalını aktif etme pini

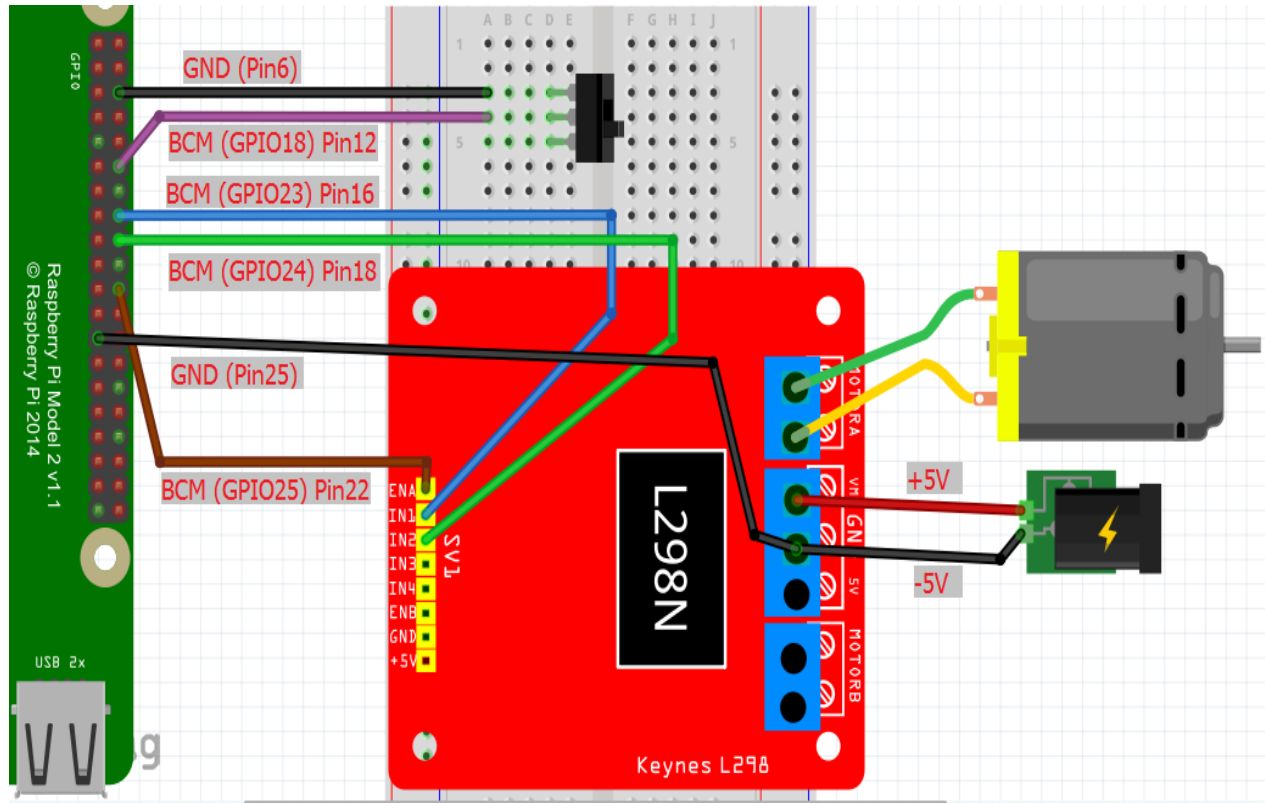
IN1, IN2: 1. Motor’u kontrol etme pinleri

IN3, IN4: 2. Motor’u kontrol etme pinleri

Şekil 6.16. L298N Motor Sürücü Kartı

Motor için gerekli olan besleme gerilimi (Bu deneyde 5V) “VMS” ve “GND” arasına uygulanır. Motorun “+” ve “-” uçları “MOTORA” veya “MOTORB” kısmına bağlanır. Motor kanallarını aktif etmek için “ENA” ve “ENB” pinlerine lojik 1 gönderilir. Motor ileri yönde döndürülmek istenirse 1. Motor kanalının kontrol pinleri olan “EN1” e lojik 1 “EN2”ye lojik 0 gönderilir, motor geri yönde döndürülmek istenirse 1. Motor kanalının kontrol pinleri olan “EN1” e lojik 0 “EN2”ye lojik 1 gönderilir aynı şekilde 2. Motor kanalı kullanılıyorsa motoru ileri yönde döndürmek için 2. Motor kanalının kontrol pinleri olan “EN3” e lojik 1 “EN4” e lojik 0 gönderilir, geri yönde döndürmek için ise 2. Motor kanalının kontrol pinleri olan “EN3” e lojik 0 “EN4” e lojik 1 gönderilir.

6.8.3. Deney Devresi



Şekil 6.17. 8. Deney Devresi

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python motorkontrol.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.8.4. KOD:

```
#!/usr/bin/env python
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
import time # Zaman kütüphanesini çağırılır
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
anahtar = 18
motor1A = 23
motor1B = 24
motor1E = 25

GPIO.setup(anahtar, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(motor1A, GPIO.OUT)
GPIO.setup(motor1B, GPIO.OUT)
GPIO.setup(motor1E, GPIO.OUT)

def sıfırla(): # Pinleri temizlemek için sıfırlama fonksiyonu oluşturuldu
    GPIO.output(motor1A,GPIO.LOW)
    GPIO.output(motor1B,GPIO.LOW)
    GPIO.output(motor1E,GPIO.LOW)

def ileri(): # Motoru ileri yönde döndürmek için ileri fonksiyonu oluşturuldu
    print 'motor ileri dönüyor'
    sıfırla()
    GPIO.output(motor1A,GPIO.HIGH)
    GPIO.output(motor1B,GPIO.LOW)
    GPIO.output(motor1E,GPIO.HIGH)
```

```

def geri(): # Motoru geri yönde döndürmek için geri fonksiyonu oluşturuldu
    print 'motor geri dönüyor'
    time.sleep(2)
    sifirla()
    GPIO.output(motor1A,GPIO.LOW)
    GPIO.output(motor1B,GPIO.HIGH)
    GPIO.output(motor1E,GPIO.HIGH)

try:
    while True: # Aşağıdaki koşullar sonsuz döngüye sokuldu
        GPIO.wait_for_edge(anahtar, GPIO.FALLING)
    # GPIO18 düşen kenar beklemeye ayarlandı
        time.sleep(2) # 2 saniye bekler
        ileri() # İleri fonksiyonu çalıştırılır
        GPIO.wait_for_edge(anahtar, GPIO.RISING)
    # GPIO18 düşen kenar beklemeye ayarlandı
        time.sleep(2) # 2 saniye bekler
        geri()# İleri fonksiyonu çalıştırılır

except KeyboardInterrupt:
    sifirla()
    GPIO.cleanup()

```

Not: L298N motor sürücü kartı entegresinin datasheet'ine aşağıdaki adresten bakılabilir.

<http://www.alldatasheet.com/datasheet-pdf/pdf/22440/STMICROELECTRONICS/L298N.html>

6.9. Deney 9: ULTRASONİK SES SENSÖRÜ KULLANIMI

6.9.1. Amaç: HC-SR04 ultrasonik ses sensörünü kullanmayı öğrenme ve mesafe ölçme

Kullanılacak Malzemeler:

- HC-SR04 ultrasonik ses sensörü
- 1 adet 330Ω direnç
- 1 adet 470Ω direnç

6.9.2. Genel Bilgiler:

Deneyde ultrasonik ses sensörü kullanılarak mesafe ölçümü yapılacaktır. Ses dalgaları sınıflandırılmasında 20 KHz-1 GHz arasındaki ses sinyalleri ultrasonik ses olarak tanımlanmıştır. Ultrasonik sesler insan kulağı tarafından algılanamazlar. HC-SR04 ultrasonik ses sensörü 40KHz frekansında 340 m/s hızında ultrasonik ses üretmektedir. HC-SR04 tüm haliyle bir transdüserdir. Transdüser ile sensör arasındaki fark ise sensör herhangi fiziksel bir değişimi algılayan bir elemandır, transdüser ise bu değişimi algılayıp elektrik, ısı vs. bir çok enerji formuna dönüştürebilen bir elektronik devredir.

Ultrasonik ses sensörünün teknik özellikleri:

- DC 5V besleme gerilimi
- 2cm – 4m ölçüm aralığı
- 15 derece ölçüm açısı
- Ölçüm hassasiyeti 0.3cm
- 40 KHz, 340 m/s ses sinyali üretir
- Bir ses sinyali üretici ve bir ses sinyali algılayıcısından oluşur.

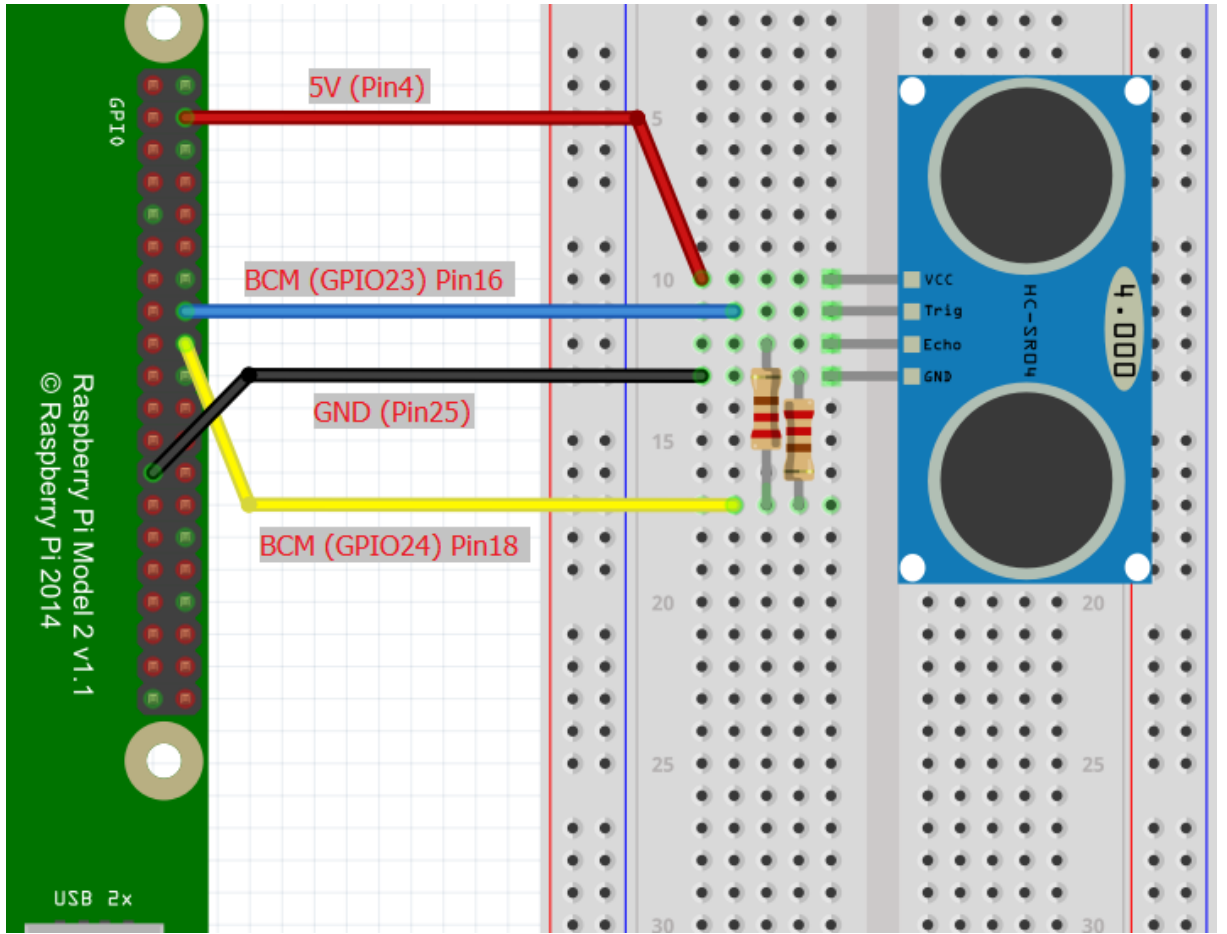
Ultrasonik ses sensörü ile mesafe ölçümü yapabilmek için önce ses üreticiden (Trigger) 10 mikro saniyelik bir ses sinyali gönderilir ve ardından zaman saymaya başlanır. Ses algılayıcısı (Echo) tarafından ses sinyali algılandıktan sonra zaman sayma işlemi sonlandırılır ve mesafe = ((tutulan zaman) x (ses sinyalinin hızı)) / 2 formülünden mesafe hesaplanır.

$$Mesafe = \frac{Tutulan\ zaman \times Ses\ sinyalinin\ hızı}{2}$$



Şekil 6.18. Ultrasonik Ses Sensörü

6.9.3. Deney Devresi



Şekil 6.19. 9. Deney Devresi

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python ultrasoniksensör.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.9.4. KOD:

```
#!/usr/bin/python
```

```
import time # Zaman kütüphanesini çağırılır
```

```
import RPi.GPIO as GPIO # Python için oluşturulmuş GPIO kütüphanesi çağırılır
```

```
def olcum(): # Mesafe ölçümü fonksiyonu oluşturuldu
```

```
    GPIO.output(GPIO_TRIGGER, True)
```

```
    time.sleep(0.00001)
```

```
    GPIO.output(GPIO_TRIGGER, False)
```

```
    basla = time.time()
```

```
    while GPIO.input(GPIO_ECHO)==0:
```

```
        basla = time.time()
```

```
    while GPIO.input(GPIO_ECHO)==1:
```

```
        dur = time.time()
```

```
    gecen_zaman = dur-basla
```

```
    mesafe = (gecen_zaman*34300)/2
```

```
    return mesafe
```

```
def ortalama_olcum(): # Ölçümdeki hatayı azaltmak için 3 ölçümün ortalaması alındı
```

```
    mesafe1=olcum()
```

```
    time.sleep(0.1)
```

```
    mesafe2=olcum()
```

```
    time.sleep(0.1)
```

```
    mesafe3=olcum()
```

```
    mesafe = mesafe1 + mesafe2 + mesafe3
```

```
    mesafe = mesafe / 3
```

```
    return mesafe
GPIO.setmode(GPIO.BCM) # GPIO isimlendirmesi işlemci (BCM) sıralaması
GPIO_TRIGGER = 23
GPIO_ECHO = 24
print "Mesafe Ölçme"

GPIO.setup(GPIO_TRIGGER,GPIO.OUT)
GPIO.setup(GPIO_ECHO,GPIO.IN)
GPIO.output(GPIO_TRIGGER, False)

try:
    while True:
        mesafe = ortalama_olcum()
        print "Mesafe : %.1f" % mesafe
        time.sleep(1)

except KeyboardInterrupt:
    GPIO.cleanup()
```

6.10. Deney 10: ISI SENSÖRÜ KULLANIMI

6.10.1. Amaç: DHT11 ısı ve nem sensörünü kullanmayı öğrenme ve ortam ısını ölçme

Kullanılacak Malzemeler:

- DHT11 ısı ve nem sensörü
- 1 adet 5K Ω direnç

6.10.2. Genel Bilgiler:

Deneyde ısı ve nem sensörü kullanılarak, ortam ısı ile nem değeri ölçülecektir. DHT11 içerisinde 8 bit mikro işlemci bulunduran ve dijital çıkış veren, nem ve sıcaklık birimi bulunan bir sensördür. Özel bir dijital sinyal elde etme tekniğine sahiptir. DHT11 sensörü içerisinde direnç tipi nem ölçüm bileşeni ve NTC Sıcaklık ölçüm bileşeni barındırmaktadır. Tek kablo üzerinden seri iletişim sistemi ile kontrol edilir.

Isı ve nem sensörünün teknik özellikleri:

- 3V – 5.5V DC besleme gerilimi
- Dijital sinyal çıkışı (8 bit)
- Çift yön seri veri iletişimi
- %20 - %90 nem, 0°C – 50°C sıcaklık ölçüm aralığı
- Nem ölçüm doğruluğu ± 5 %Rh ‘tır.
- Isı ölçüm doğruluğu ± 2 °C ‘dir.
- 4 pinden oluşur 3. Pini boştur.
- 20m’ye kadar data pini 5K Ω pull-up direnç ile bağlanır.
- 20m’den uzun mesafelerde uygun değerde bir pull-up direnç seçilir.

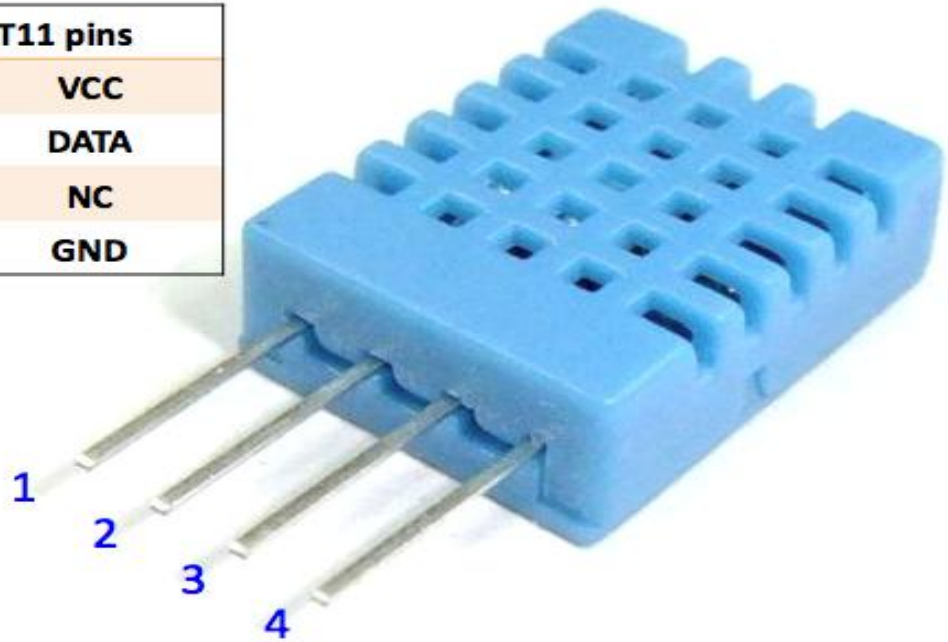
DHT11 ısı ve nem sensörü toplam 40 bitlik (5 byte) veri gönderir. İlk byte nem değerinin tam sayı kısmını, 2. byte nem değerinin ondalıklı kısmını, 3. byte sıcaklık değerinin tam sayı kısmını, 4. byte ise sıcaklık değerinin ondalıklı kısmını ifade eder. Veri iletimi doğru bir şekilde tamamlanırsa 5. byte ilk 4 byte toplamına eşit olur.

Her bit gönderiminden önce 50 μ s lik bir bekleme süresi vardır. Bu sürenin ardından gelen sinyal 26 – 28 μ s arasında sürüyor ise bu bit değeri olarak “0” ‘ı , 70 μ s sürüyor ise bit değeri olarak “1” ‘i ifade etmektedir. Raspberry pi de, yazılan programa göre gelen sinyalin uzunluğuna bakarak, gelen bitin “0” ya da “1” olduğuna karar verir. Ardından bu bitleri onluk sisteme çevirilerek ısı ve nem değerleri gösterilir.



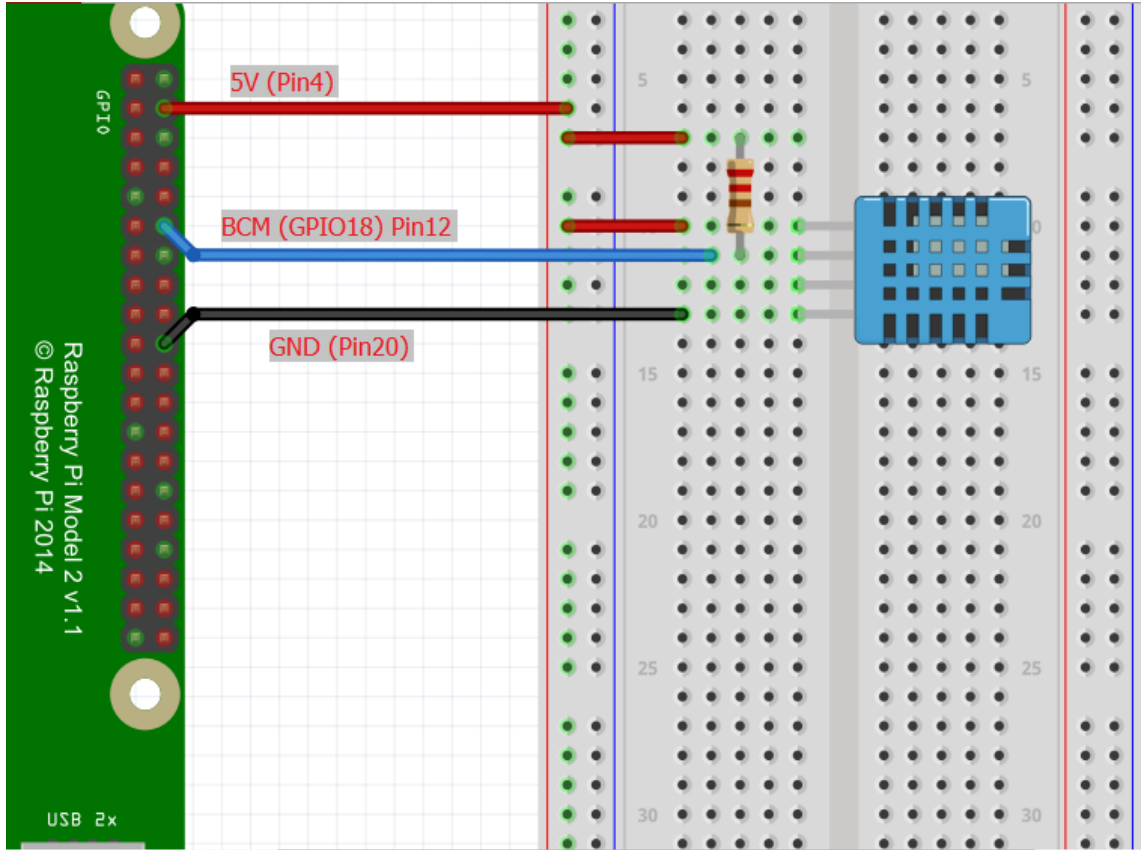
Şekil 6.20. Isı ve Nem Sensörü Data Bitleri

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Şekil 6.21. Isı ve Nem Sensörü

6.10.3. Deney Devresi



Şekil 6.22. 10. Deney Devresi

Deneyin Yapılışı

Yukarıdaki deney devresi breadboard üzerine kurulur ve terminalden “sudo python ısisensör.py” komutu girilir, açılan pencerede aşağıdaki kodlar yazılıp kaydedilir.

6.10.4. KOD:

```
import RPi.GPIO as GPIO
```

```
import time
```

```
def bin2dec(string_num):
```

```
    return str(int(string_num, 2))
```

```
data = []
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(4,GPIO.OUT)
```

```
GPIO.output(4,GPIO.HIGH)
```

```
time.sleep(0.025)
```

```
GPIO.output(4,GPIO.LOW)
```

```
time.sleep(0.02)
```

```
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

```
for i in range(0,500):# Sensörden alınan veriler data dizisine kaydedilir, nem ve ısı verileri ayrıştırılarak binary'den decimal'a çevrilir.
```

```
    data.append(GPIO.input(4))
```

```
bit_count = 0
```

```
tmp = 0
```

```
count = 0
```

```
HumidityBit = ""
```

```
TemperatureBit = ""
```

```
crc = ""
```

```
try:
    while data[count] == 1:
        tmp = 1
        count = count + 1
    for i in range(0, 32):
        bit_count = 0
        while data[count] == 0:
            tmp = 1
            count = count + 1
        while data[count] == 1:
            bit_count = bit_count + 1
            count = count + 1
        if bit_count > 3:
            if i >= 0 and i < 8:
                HumidityBit = HumidityBit + "1"
            if i >= 16 and i < 24:
                TemperatureBit = TemperatureBit + "1"
        else:
            if i >= 0 and i < 8:
                HumidityBit = HumidityBit + "0"
            if i >= 16 and i < 24:
                TemperatureBit = TemperatureBit + "0"
except:
    pass
```

```
try:
    for i in range(0, 8):
        bit_count = 0
        while data[count] == 0:
            tmp = 1
            count = count + 1
        while data[count] == 1:
            bit_count = bit_count + 1
            count = count + 1
        if bit_count > 3:
            crc = crc + "1"
        else:
            crc = crc + "0"
except:
    pass

Humidity = bin2dec(HumidityBit)
Temperature = bin2dec(TemperatureBit)

try:
    while True: # Decimal'a çevirilen nem ve ısı verileri ekrana bastırılır.

        if int(Humidity) + int(Temperature) - int(bin2dec(crc)) == 0:
            print "Humidity:" + Humidity + "%"
            print "Temperature:" + Temperature + "C"
        else:
            pass
except KeyboardInterrupt:
    GPIO.cleanup()
```


6.11. Deney 11: HAREKET SENSÖRÜ VE KAMERA KULLANIMI

6.11.1. Amaç: Passive infrared sensor (PIR) kullanımını öğrenme ve PIR ve kamerayı birlikte kullanma

Kullanılacak Malzemeler:

- HC-SR501 PIR hareket sensörü
- Raspberry pi kamera modülü

6.11.2. Genel Bilgiler:

Deneyde Passive infrared sensör (PIR sensor) ve kamera birlikte kullanılarak PIR sensörü hareket algıladığında kameranın fotoğraf çekmesi sağlanır. Uygulama MATLAB programında gerçekleştirilir.

Passive infrared sensor (PIR sensor) objelerden yayılan kızılötesi ışınları ölçer ve kızılötesi ışınlardaki değişimi algılar, kızılötesi ışınlarda belirlenen değişim önceden tanımlanan değerden büyükse bilgi sinyali üretir.

Passive infrared sensor (PIR sensor) kızılötesi ışınların değişiminden algıladığı hareket bilgisini dijital sinyal olarak verir. PIR sensör hareket algılayamadığı zaman lojik 0 bilgisi üretir, bir hareket algıladığında ise lojik 1 bilgisi verir ve bu bilgiyi yaklaşık 5 saniye boyunca tutar. PIR sensör üzerinde “Sx” ve “Tx” olmak üzere iki adet potansiyometre bulunur. “Sx” potansiyometresi sensörün görme mesafesini 3 ile 5 metre arasında değiştirir. “Tx” potansiyometresi ise sensör gördükten sonra ne kadar süre daha çıkış pininden lojik 1 çıkışını vereceğini ayarlar.

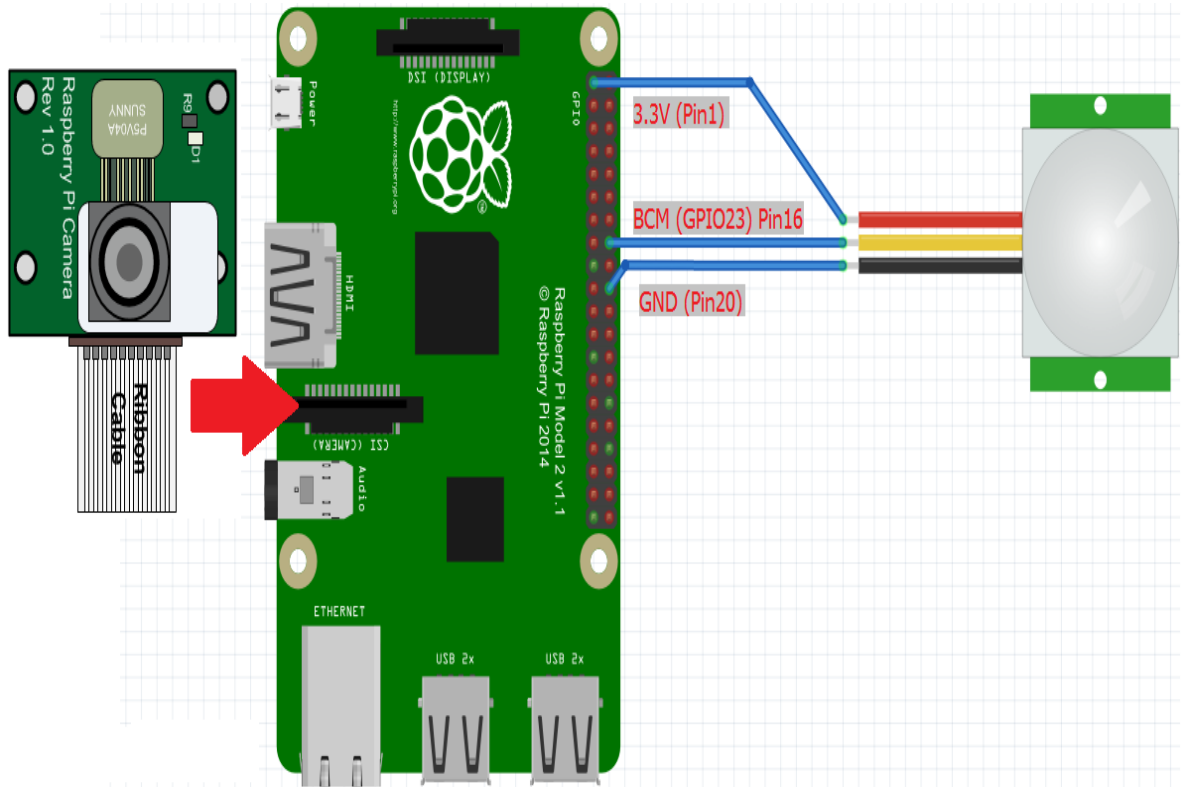
Passive infrared sensor (PIR sensor) teknik özellikleri:

- Besleme gerilimi 3.3V – 5V
- Hareket algılama mesafesi 3m – 5m
- Lojik 1 bilgisini tutma süresi 5sn – 200sn
- Algılama açısı 140°



Şekil 6.23. PIR Sensörü

6.11.3. Deney Devresi



Şekil 6.24. 11. Deney Devresi

Deneyin Yapılışı

Şekilde görüldüğü gibi kamera modülü ve Passive infrared sensör (PIR sensor) raspberry pi'ye bağlanır ve aşağıdaki matlab kodları editörde çalıştırılır.

6.11.4. KOD:

```
clear rpi  
rpi = raspi();  
cam = cameraboard(rpi);  
N = 100;  
delay = 2;  
frameNo = 0;  
  
while 1  
    motionDetected = readDigitalPin(rpi, 23);  
    if motionDetected >0  
        disp(motionDetected);  
        fprintf('Motion detected on %s\n', datestr(now));  
        snapshot(cam);  
        img = snapshot(cam);  
        imagesc(img);  
        drawnow;  
        imwrite(img, sprintf('image%d.jpg', frameNo));  
    else  
        disp(motionDetected);  
    end  
    pause(delay);  
end
```

6.12. Deney 12: ANALOG SİNYAL OKUMA

6.12.1. Amaç: Serial Peripheral Interface (SPI) seri haberleşme birimini kullanmayı ve analog dijital çevirici (analog digital converter “ADC”) kullanmayı öğrenme.

Kullanılacak Malzemeler:

- MCP3008 analog dijital çevirici
- 1 adet 10K Ω potansiyometre (ayarlanabilir direnç)

6.12.2. Genel Bilgiler:

Raspberry pi üzerinde analog sinyal girişi bulunmamaktadır, bu nedenle analog bir sinyali okuyabilmek için analog dijital çeviriciye (analog digital converter “ADC”) ihtiyaç duyulur. Deneyde MCP3008 analog dijital çevirici ile potansiyometre üzerinden gelen analog sinyalin değeri okunacaktır. MCP3008 SPI seri haberleşme birimi içeren bir analog dijital çeviricidir, raspberry pi ile SPI üzerinden haberleştirilir.

SPI haberleşme çevresel aygıtlar için kullanılan senkron veri bağlantısıdır. Ana SPI cihazı bir yada birden çok çevresel SPI cihazı ile haberleşebilir. SPI haberleşmede tüm çevre birimleri için ortak kullanılan 3 hat (Master in slave out “MISO”, Master out slave in “MOSI” ve Serial clock “SCLK”) ve hangi çevre birimle çalışacağını seçmek için CS (Chip select) hattı bulunur. Raspberry pi de 2 tane CS hattı (CE0, CE1) bulunur.

- Master in slave out (MISO): Veriler bu hatta çevre birimlerden ana SPI cihazına taşınır.
- Master out slave in (MOSI): Veriler bu hatta ana SPI cihazından çevre birimlere taşınır.
- Serial clock (SCLK): Ana SPI cihazı veri iletimi için saat sinyali oluşturur.

SPI’da her bir çevre birim aygıtı CS hattına bağlanır. SPI haberleşmesi sırasında ana SPI cihazı (Raspberry pi) her seferinde bir çevre birim aygıtını seçer.

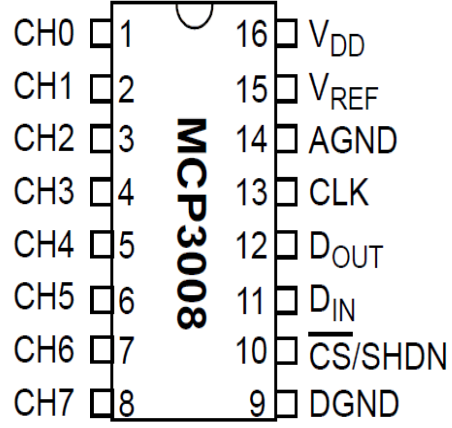
Analog Sinyal: Analog sinyallere kısaca sürekli sinyaller denir ve zamana göre değişen başka bir türden niceliği temsil edip kendisinde zamana göre değişen ve sürekli özellik gösteren sinyallerdir. Yani sinyal zaman ya da uzay gibi sürekli bir değişkenin fonksiyondur ve genellikle sürekli bir aralıkta değer alır. Mühendislikte karşılaşılan bir çok sinyal analogtur. Analog sinyaller, giriş sinyalinin bazı başka sinyaller ile toplama, çarpma veya faz farkı gibi elektriksel işlemler geçirilerek değiştirilmesi ile elde edilir. Dolayısıyla, giriş sinyali ve çıkış sinyali arasındaki geçiş bir matematiksel formül ile gösterilebilir. Analog sinyaller genlik, frekans, periyot, faz farkı bilgileriyle tanımlanırlar.

Dijital Sinyal: Sayısal veri, dijital veri veya dijital sinyal. Bir analog sinyalden belirli örnekler alınır ve analog sinyalin tam karşılığı olmayan dijital sinyal oluşturulur. Girişteki verinin saklanma veya aktarılma şeklinin değiştirilmesiyle elde edilir. Dijital sinyal ile analog sinyal arasındaki en büyük fark, analog sinyalin sürekli olan bir ölçekte, dijital sinyalin ise rakamlarla sınırlı olan, sürekli olmayan bir ölçekte var olmasıdır. Dijital sinyal rakamların arka arkaya dizilmesiyle elde edildiği için üzerinde biçim değişikliği yapılabilir. Dijital sinyal; olgu kavram yada komutların; iletişim, yorum ve işlem için elverişli biçimsel ve uzlaşımsal gösterimidir buna ek olarak bir dijital sinyale hata düzeltme kodları eklenerek veri bozulsa bile tamir edilmesi sağlanabilir. Ayrıca dijital sinyal gönderilirken başka bir dijital veri ile aynı ortama konulup gönderilebilir. Tüm bu avantajlar doğrultusunda, dijital sinyal bir çok alanda analog sinyalin yerini almaktadır.

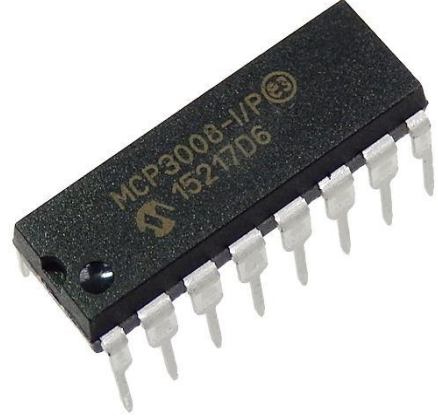
MCP3008: Microchip firmasının ürettiği ardışık yaklaşımlı (successive approximation register “SAR”) tipinde analog sinyalleri dijital sinyale çeviren bir entegredir.

- 8 analog giriş kanalı bulunur
- 10 – Bit Çözünürlüklüdür.
- SPI Serial Interface (0,0 ve 1,1)
- 100 ksps örnekleme ($V_{DD} = 5 V$)
- 50 ksps örnekleme ($V_{DD} = 2.7 V$)

- 2.7V – 5V besleme gerilimi

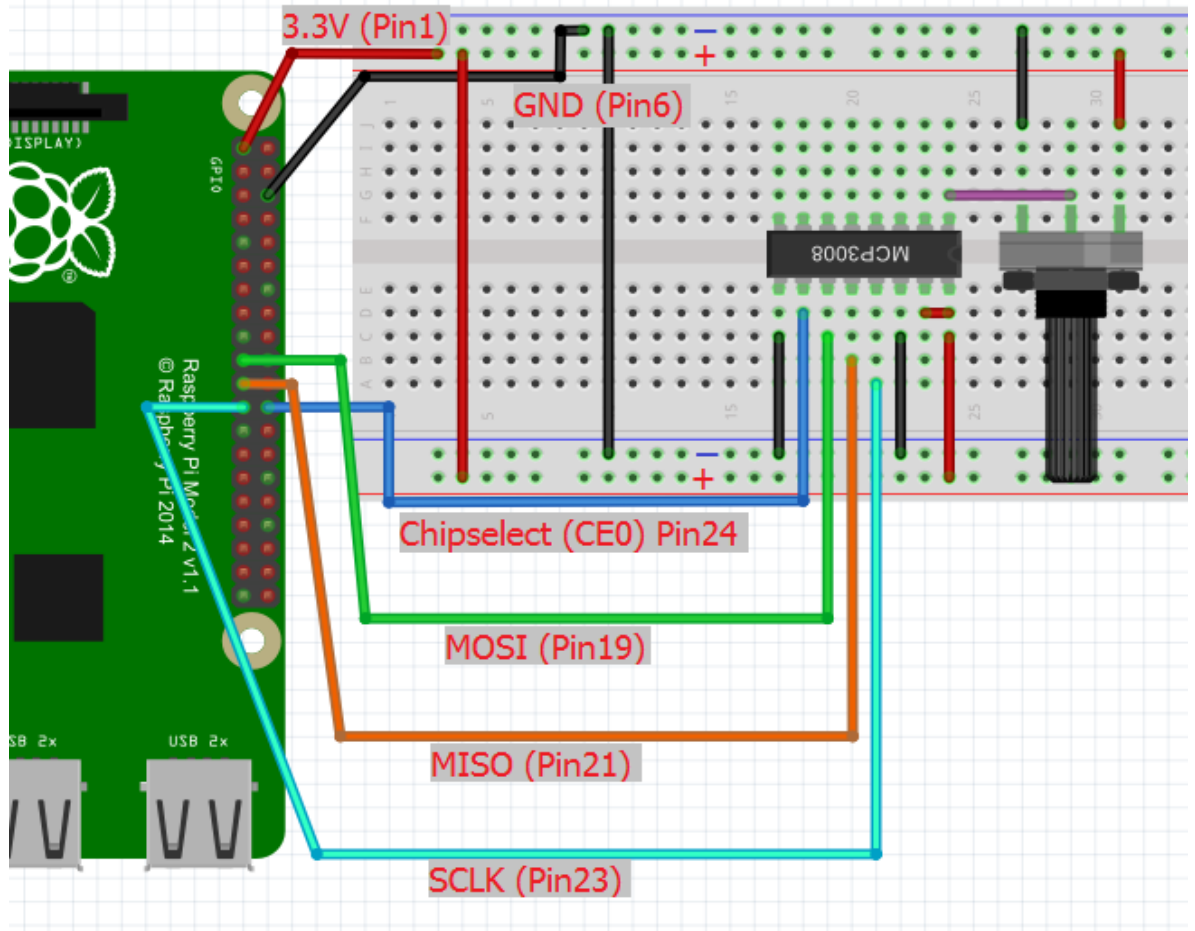


Şekil 6.25. MCP3008 Pin Şeması



Şekil 6.26. MCP3008 ADC

6.12.3. Deneysel Devre



Şekil 6.27. 12. Deneysel Devre

Deneysel Yapı

Şekilde görüldüğü gibi MCP3008 ve potansiyometre bağlantıları yapılır ve aşağıdaki matlab kodları editörde çalıştırılır.

6.12.4. KOD:

```
clear rpi
```

```
rpi = raspi();
```

```
rpi.AvailableSPICannels
```

```
enableSPI(rpi);
```

```
clear mcp3008
```

```
mcp3008 = spidev(rpi, 'CE0');  
  
data = uint16(writeRead(mcp3008, [1, bin2dec('10000000'), 0]));  
  
highbits = bitand(data(2), bin2dec('11'));  
  
voltage = double(bitor(bitshift(highbits, 8), data(3)));  
  
voltage = (3.3/1024) * voltage;  
  
while 1  
  
    voltage = readVoltage(mcp3008, 0);  
  
    fprintf('Voltage = %0.2f\n', voltage);  
  
    pause(0.2);  
  
end
```


6.13. Deney 13: I2C ile 4 Digit 7 Segment Display Kontrol Etme

6.13.1. Amaç: Inter – İntegrale Curcuit (I2C) haberleşme protokolünü öğrenme ve 4 digit 7 segment display kontrol etme.

Kullanılacak Malzemeler:

- Spurfun 4 digit 7 segment I2C ve SPI arabirimli display
- 2 adet 1.8K Ω direnç

6.13.2. Genel Bilgiler:

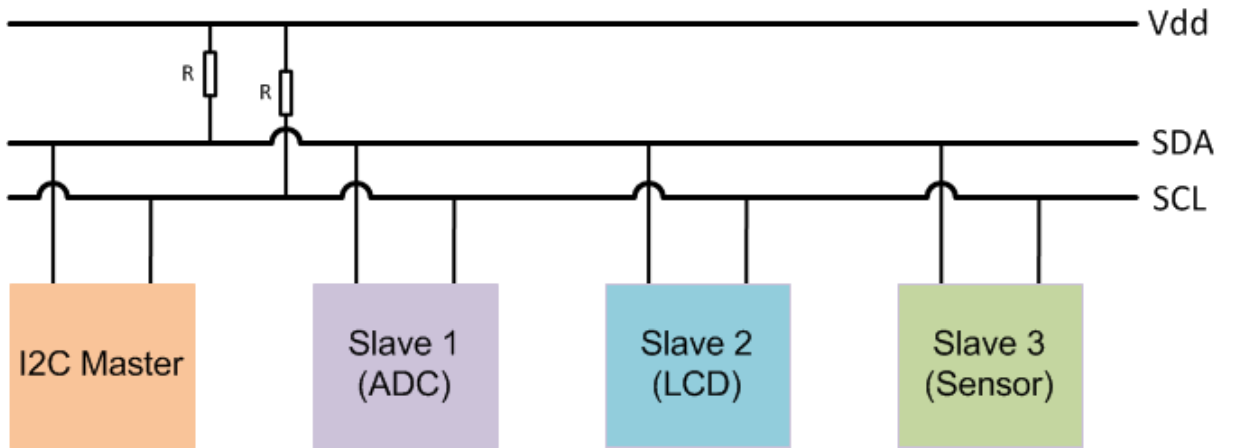
Raspberry pi üzerine I2C (Inter – İntegrale Curcuit) seri haberleşme türünden senkron haberleşme birimini bulundurur. I2C bir gömülü sistemdeki düşük hızlı çevre birimlerini bağlamak için kullanılan multi-master seri tek uçlu iletişim protokolüdür. I2C kısa mesafede ve düşük veri hızına ihtiyaç duyulan haberleşmelerde kullanılır.

I2C biriminde SDA (Serial Data) ve SCL (Serial Clock) olmak üzere iki adet çift yönlü hat bulunur.

- Serial Data Line (SDA): Ana cihaz ile çevre birim aygıtı arasında veri aktarımını yapıldığı çift yönlü hattır.
- Serial Clock (SCL): Veri iletişiminin senkronizasyonunun sağlandığı hattır. Veriler SCL de üretilen zaman sinyali ile birlikte taşınır.

Haberleşmenin tüm hat boyunca hatasız bir şekilde sağlanabilmesi için SDA ve SCL hatları, pull-up dirençlerle VCC hattına bağlanmalıdır.

I2C yolu üzerindeki her slave aygıtı bir 7-bit yada 10-bit adres atanır.10-bit adrese sahip I2C aygıtları nadirdir ve Raspberry Pi donanımı tarafından desteklenmez. 7-bit adres alanı ile I2C yolu üzerine 128 aygıtı kadar bağlantı sağlanabilir. I2C yolu hızı yavaş mod için 100 Kbit/s ve yüksek hızlı mod için 400Kbit/s arasında değişir.



Şekil 6.28. I2C Bağlantı Şeması

SparkFun 4-Digit 7-Segment Display

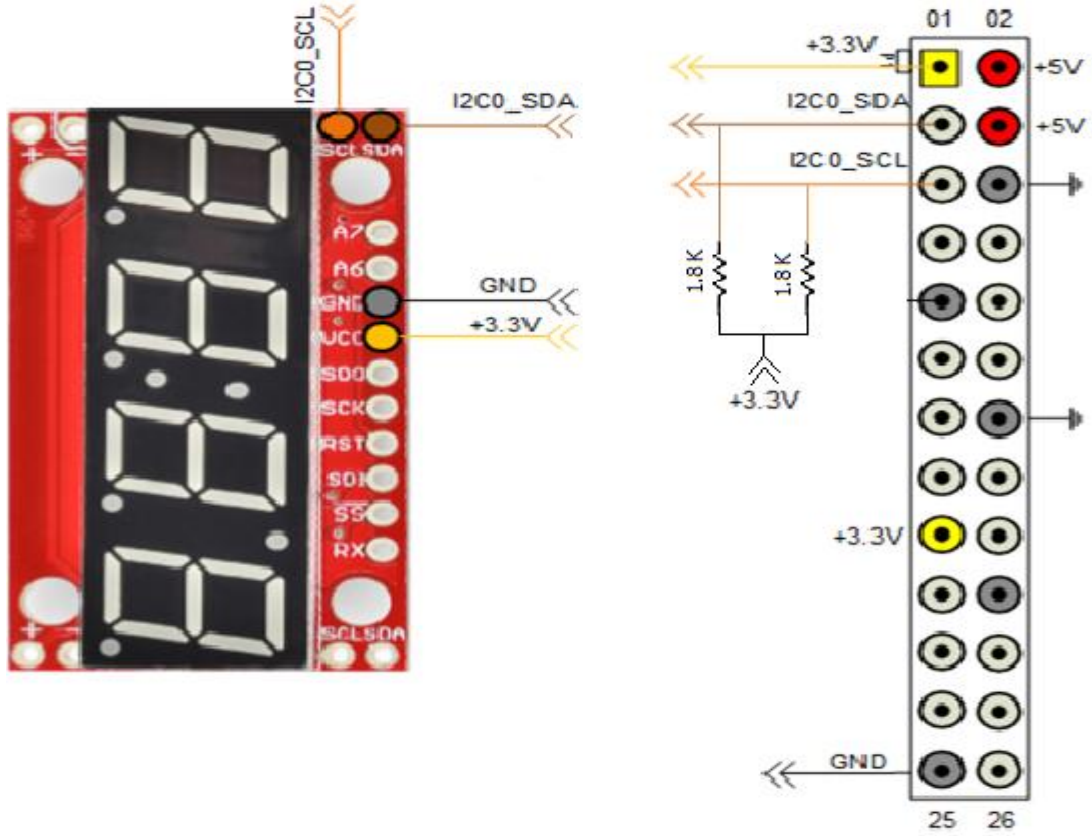
SparkFun tarafından üretilen 4 digit 7 segment display numaraları, çoğu harfleri ve ondalık noktaları, kesme işareti gibi birkaç özel karakteri görüntüleyebilir. Ekran parlaklığı ayarlanabilir. I2C ve SPI haberleşme protokollerini desteklemektedir, deneyde raspberry pi ile I2C haberleşme protokolü üzerinden kontrol edilecektir.

Displayin I2C adresi, üretilirken '0x71' olarak ayarlanmıştır. Displayi temizlemek için ana cihazdan (Raspberry pi'den) '0x76' komut byte'ı gönderilir. Belirli bir konuma imleci ayarlamak için ise '0x79' komut byte'nın ardından imlecin konumunu belirten (0 ile 3 arası) bir byte gönderilir.



Şekil 6.29. SparkFun 4 Digit 7 Segment Display

6.13.3. Deney Devresi



Şekil 6.30. 13. Deney Devresi

Deneyin Yapılışı

Deney devresinde görüldüğü gibi display raspberry pi bağlantısı yapılır ve aşağıdaki matlab kodları editörde çalıştırılır.

6.13.4. KOD:

```
clear rpi  
rpi = raspi();  
for i = 1:length(rpi.AvailableI2CBuses) % I2C aygıtının bağlantısı test edildi  
    scanI2CBus(rpi, rpi.AvailableI2CBuses{i})  
end  
segmentDisp = i2cdev(rpi, 'i2c-1', '0x71');  
write(segmentDisp, 0); % En soldaki digit'e 0 yazdırılır.  
write(segmentDisp, 7); % Bir sonraki digit'e 7 yazdırılır.  
write(segmentDisp, hex2dec('a')); % Bir sonraki digit'e a harfi yazdırılır.  
write(segmentDisp, hex2dec('b')); % Bir sonraki digit'e b harfi yazdırılır.  
write(segmentDisp, hex2dec('76'));% Ekran temizlenir.  
write(segmentDisp, [hex2dec('79'), 2]); % İmleç konumu 3. digit'e ayarlanır.  
write(segmentDisp, 9); % 3. digit'e 9 yazdırılır.
```

6.14. Deney 14: Basit Voltmetre Yapma

6.14.1. Amaç: Potansiyometre değerini display üzerinden okuma

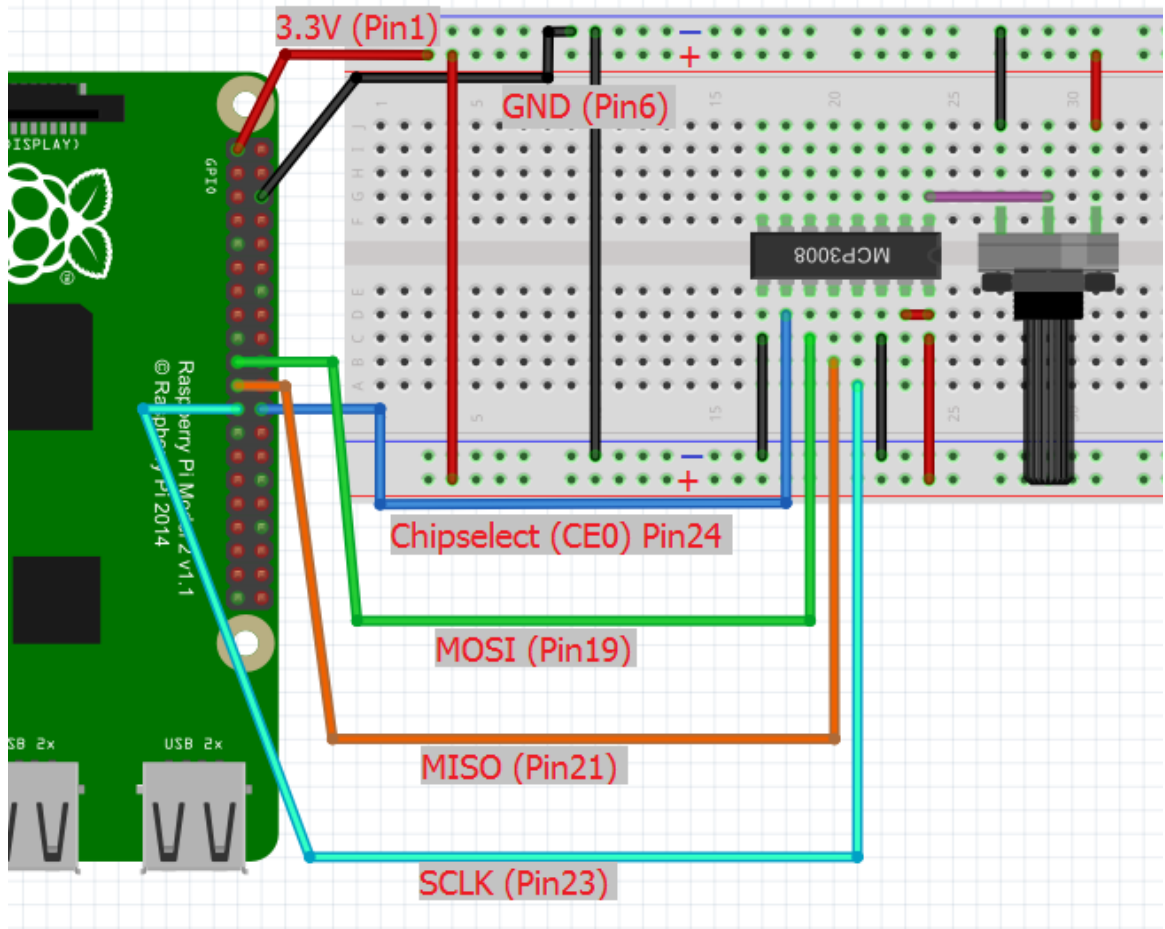
Kullanılacak Malzemeler:

- Spurfun 4 digit 7 segment I2C ve SPI arabirimli display
- 2 adet $1.8K\Omega$ direnç
- MCP3008 analog dijital çevirici
- 1 adet $10K\Omega$ potansiyometre (ayarlanabilir direnç)

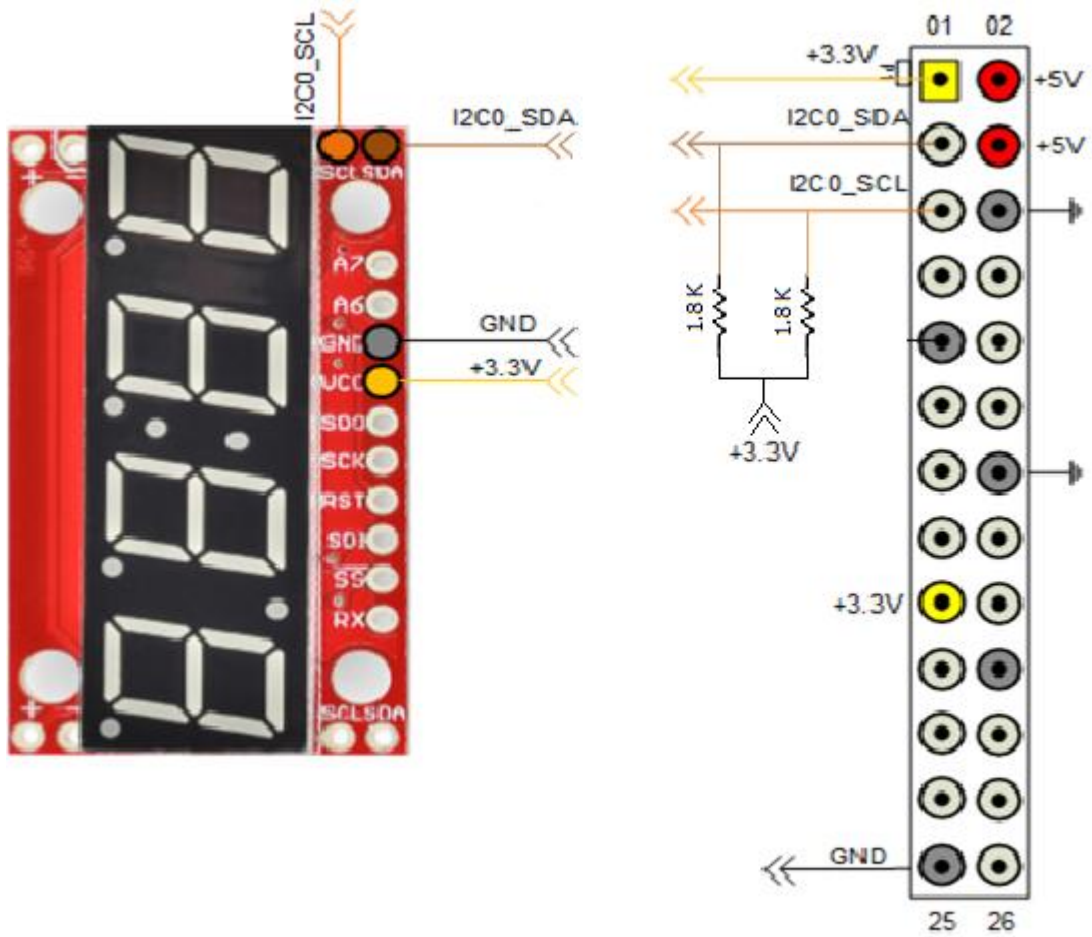
6.14.2. Genel Bilgiler:

Bu deney de analog digital konverter ve 4 digit 7 segment display beraber kullanılarak potansiyometre'nin değeri analog dijital converter yardımıyla dijitale çevirilerek displayde yazdırılacaktır. Deneyin gerçekleştirilmesinde 12. ve 13. Deney bilgilerinden faydalanılacaktır.

6.14.3. Deney Devresi



Şekil 6.31. 14. Deney Devresi 1



Şekil 6.32. 14. Deney Devresi 2

Deneyin Yapılışı

Yukarıda ki şekillerde görüldüğü gibi display ve ADC'nin bağlantıları gerçekleştirildikten sonra aşağıdaki kodlar matlab editörde çalıştırılır.

6.14.4. KOD:

```
clear rpi
rpi = raspi();
rpi.AvailableSPIChannels
enableSPI(rpi);
for i = 1:length(rpi.AvailableI2CBuses)
    scanI2CBus(rpi, rpi.AvailableI2CBuses{i})
end

clear mcp3008
mcp3008 = raspi.internal.mcp3008(rpi, 'CE0');
for i = 1:20
    voltage = readVoltage(mcp3008, 0);
    fprintf('Voltage = %0.2f\n', voltage);
    pause(0.2);
end
delay = 0.1;
adcChannel = 0;
clr(sd);
for i = 1:N
    voltage = readVoltage(mcp3008,adcChannel);
    print(sd,sprintf('%0.0f', voltage*100));
    pause(delay);
end
```

KAYNAKÇA

İnternet : <http://www.raspi.gen.tr/2015/06/06/pwm-mantigi-ve-de-ledlerde-pwm-kullanimi/>

İnternet : <http://www.burakgokalp.com/2014/07/raspberry-pi-yi-dogrudan-bilgisayara.html>

İnternet : <http://www.volkanaktas.com/2015/02/raspberry-pi-isletim-sistemi-kurulumu/>

İnternet : https://tr.wikipedia.org/wiki/Raspberry_Pi

İnternet : <http://www.serhatdundar.com/ssh-nedir-ssh-komutlari-nelerdir>

İnternet : <http://www.serdarselveli.com.tr/ssh-nedir-nasil-kullanilir.html>

İnternet : <http://www.teknokoliker.com/2011/12/fotodirenc.html>

İnternet : <http://www.datasheetspdf.com/PDF/HC-SR501/775434/1>

İnternet : <http://www.micropik.com/PDF/HCSR04.pdf>

İnternet : <http://www.micropik.com/PDF/dht11.pdf>

İnternet : <http://www.microchip.com/wwwproducts/en/MCP3008> MCP3008

İnternet : <http://akizukidenshi.com/img/goods/C/I-09485.jpg>

İnternet : <http://www.raspberrypi-spy.co.uk/wp-content/uploads/2013/10/mcp3008-e1397762478714.png> MCP3008

İnternet : <http://www.alldatasheet.com/datasheet-pdf/pdf/22440/STMICRO/ELECTRONICS/L298N.html>

İnternet : <http://www.elektrikport.com/teknik-kutuphane/pull-up-ve-pull-down-direnc-nedir/16930#ad-image-0>

İnternet : <https://sourceforge.net/p/raspberry-gpio-python/wiki/Inputs/>

İnternet : <http://raspberrypi.stackexchange.com/questions/4906/control-hardware-pwm-frequency>

İnternet : <http://www.mathworks.com/help/supportpkg/raspberrypiio/index.html>

İnternet :

http://www.mathworks.com/help/supportpkg/raspberrypiio/examples.html?s_cid=doc_footer

Kılıçdağı, A., “Raspberry Pi”, Dikey Eksen Yayınevi, Ümraniye- İstanbul, 2014.