

Bilecik Şeyh Edebali Üniversitesi

MPi3 Player

Karakter LCDli Raspberry Mp3 Player

Efecan Altay

2016

Bilecik Şeyh Edebali Üniversitesi

Giriş

İot sistemler hayatımızda çoğalmakta ve iot sistemlere ait fonksiyonlar hızlıca gelişmekte.Raspberry Pi kolay ve sık kullanılan iot sistemdir.bize sunduğu portlar ile gömülü sistem uygulamalarında da mikrodenetleyici olarak kullanılabilir.

MPi3 ,cvlc ve lcd kullanılarak düşündüğüm raspberry pi okul projem.MPi3, Raspberry pi'ın mp3 playera dönüşümüdür.

İçindekiler

Giriş.....	1
Python Programlama Dili.....	3
Python'nun Kurulumu Derlenmesi ve Çalıştırılması	4
Python ile GPIO Kontrolü.....	7
Python ile Karakter LCD Kontrolü	8
VLC.....	10
EyeD3.....	11
Python SubProcess Kütüphanesi	11
.call methodu.....	11
.check_output methodu	11
.Popen.....	12
vlc_music.py	12
Kullanımı	12
Fonksiyonlar	12
mp3Player.py.....	13

Python Programlama Dili

Python yüksek seviyeli , kolay yazılabilen ve anlaşılabilen programlama dilidir.

Girintilere dayalı bir programlama dilidir.

Dosya uzantısı “.py” dir.(örn: toplama.py)

Python shell ile çalıştırılır.

Büyük küçük harfe duyarlıdır.

Türkçe karakterlerde içerebilir ☺ .

Genellikle kullanılan diğer programlama dillerinden farkı girintilere dayalı olup, köşeli parantez kullanılmaması olabilir.

Programlama yeteneği olan bir kişi için alışılması sadece bir gününü alacaktır, çünkü gerçekten basit bir kullanıma sahiptir.

Örnek bir kod :

```
Sayı = int(input("bir sayı giriniz: "));  
Toplam = Sayı + int(input("ikinci sayıyı giriniz: "));  
print("toplanmış sayı : ",toplam);
```

Çıktı:

bir sayı giriniz: 2

ikinci sayıyı giriniz: 3

toplanmış sayı : 5

>>>

Şeklinde basit bir kullanıma sahiptir. Veri türleri belirtilmez sadece dönüşümler vardır.



Python'un Kurulumu Derlenmesi ve Çalıştırılması

Python dilinin 2.0 versiyonu 16 ekim 2000 tarihinden beri güncellenmekle beraber buünkü tarihte 2.7 versiyonu bulunmaktadır.3 aralık 2008 tarihine beraber 3 versiyonu çıkmış olup, programdaki bazı etiketleri,fonksiyonları ve kütüphaneleri değiştirilmiştir.python2 ve python3 olarak bölümüş olan bu diller farklı çatı altında geliştirilmeleri sürdürölmektedir.

Python2 ve python3 ün aralarında fazla fark yoktur.Fakat derleyecisi, versiyona göre indirilmesi gerekir.Aralarındaki farkları araştırarak bulduğum güzel bir türkçe kaynaktan öğrenebilirsiniz.

http://www.istihza.com/wiki/doku.php?id=python3_ve_python2_arasindaki_farklar

Kurulum:

<https://www.python.org/downloads/> pythonun resmi sitesinden işletim sistemine göre python2 veya python3 indirilebilir.

Windowsta pythonun derlenmesi ve çalıştırılması için :

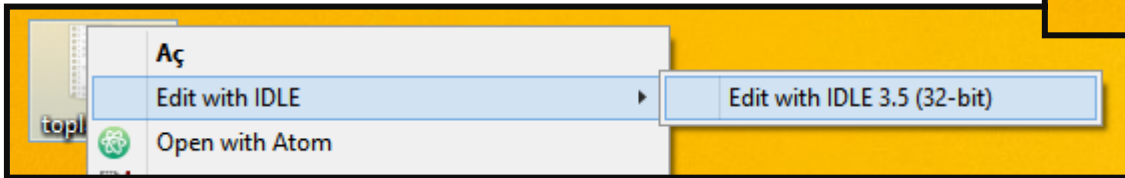
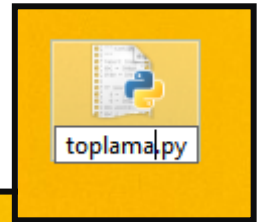
İndirme setup şeklinde olup kurma işlemleri basit bir şekilde yapılabilir.

Derlenme ve Çalıştırma:

Yazdığımız kodu derlemek ve çalıştırmak için kurulumla beraber gelen idle editorü yardımıyla python shell de gerçekleştirmektedir.

Örnek uygulama:

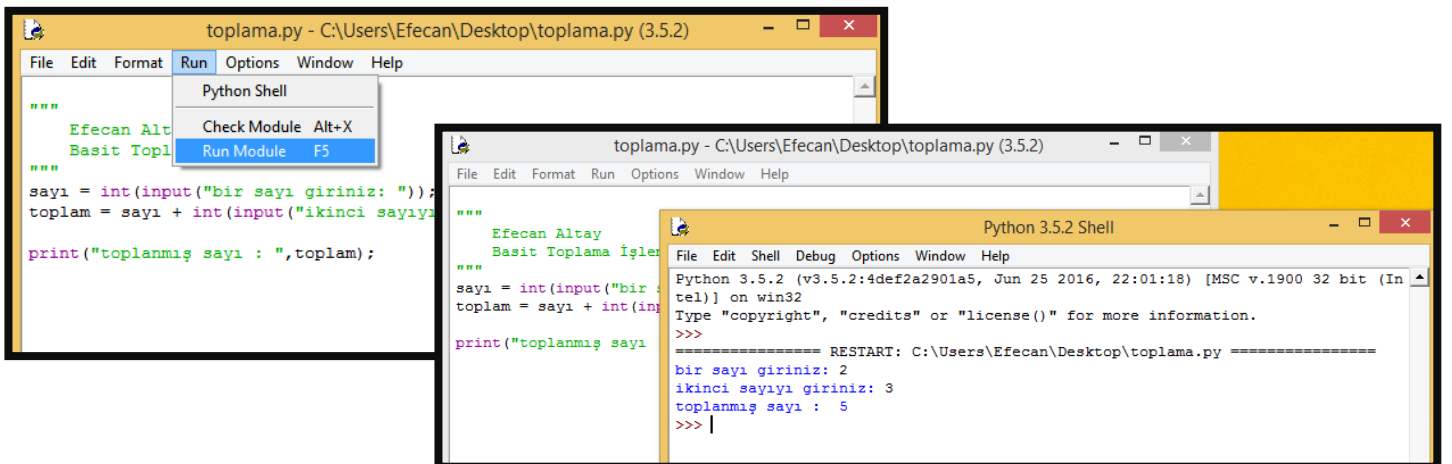
1.Bilgisayarımda masaüstüne yeni metin belgesi oluşturup “.txt” uzantısını silip “.py” olarak değitiriyorum



2.Oluşturduğum dosyama sağ tıklayıp edit with idle(idle ile düzenle) seçeneğine giriyorum.

Böylelikle idle ile kodumu yazıp derleyip çalıştırabilirim.

3.idle editörüne kodumu yazıyorum. ve f5 yada Run>Run Module seçeneinden kodumu derleyip çalıştırıyorum.python shell açılıyor ve bana hatalarımı,çıktılarımı gösteriyor veya girişlerimi talep ediyor.





Linux'ta pythonun derlenmesi ve çalıştırılması için :

Raspbian işletim sistemi linux debian tabanlı bir işletim sistemidir. ve Raspberry Pi 'mizde raspbian işletim sistemi kurulu olarak konular anlatılacaktır.Bu neden ile linux işletim sistemine pyhton derleyicisi nasıl kurul kısmını atladım.

Raspbian işletim sisteminin içinde pyhton 2 versiyonu ve Raspberry Pi ye ait kütüphaneler (örn:GPIO,USART v.b.) yüklü olarak gelmektedir.

Python 3 ile çalışmak için bu kütüphaneler hazır gelmemekle beraber python3 derleyicisine kurmak gerekir fakat aralarında çok farklar olmadığından dolayı biz python2 ile çalışmalarımızı sürdüreceğiz.

Raspberry'de örnek uygulama:

1.Raspberry pi konsol ekranı açalım ve yazacağımız kodları MyPyhton diye bir klasörde saklayalım(Bu işlem şart değil sadece düzenli çalışmak için).

\$ Mkdir MyPyhton

2.Oluşturduğumuz klasörün içine girerek nano editörle yeni bir dosya oluşturup yazalım.

\$cd MyPyhton

\$sudo nano toplama.py

(not: sudo ,super user do anlamına gelmekte dosya yazma yetkisini bu sayede alabiliriz.)

3.yazdığımız kodları kaydedip çıkış yapmak için ctrl+x tuşuna basıp,çıkan isim değiştirmede “y” yazıp enterlamamız ile kaydetmiş olacağız.(Windowsta yapılan uygulamadan farklar versiyon 2 kullanmamız)

4.Yazdığımız kodu derleyip çalıştırmak için

\$python toplama.py

Yazmamızla programımızı çalıştırmış oluyoruz.

A terminal window titled 'pi@raspberrypi: ~/MyPythonCodes' with standard window controls. The terminal shows the command 'python toplama.py' being executed. The output consists of three lines of text: 'bir sayı giriniz: 2', 'ikinci sayıyı giriniz: 3', and 'toplanmış sayı : 5'. The prompt 'pi@raspberrypi:~/MyPythonCodes \$' is followed by a green cursor.

```
pi@raspberrypi:~/MyPythonCodes $ python toplama.py
bir sayı giriniz: 2
ikinci sayıyı giriniz: 3
toplanmış sayı : 5
pi@raspberrypi:~/MyPythonCodes $
```

Windowstaki idle python editörü raspbianda da bulunmakta ama grafik arayüz kullıyorsak. ile programı aracılığıyla windowstaki gibi yazıp f5 ile derleyip çalıştırmamız mümkün.

Tek gereken şey programı çalıştırmak oda,

\$ idle toplama.py

Gibi basit bir kodla çalıştırıp çalışabiliriz.

Python ile GPIO Kontrolü

Python ile Raspberry Pi de bulunan GPIO pinlerini kontrol etmek için RPi.GPIO kütüphanesini kullanıyoruz. Tek yapmamız gereken kodumuza bu kütüphaneyi eklemek.

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

Koduyla Kütüphaneyi ekleyip G olarak yeni adını çağıracağımızı yazdırdık.

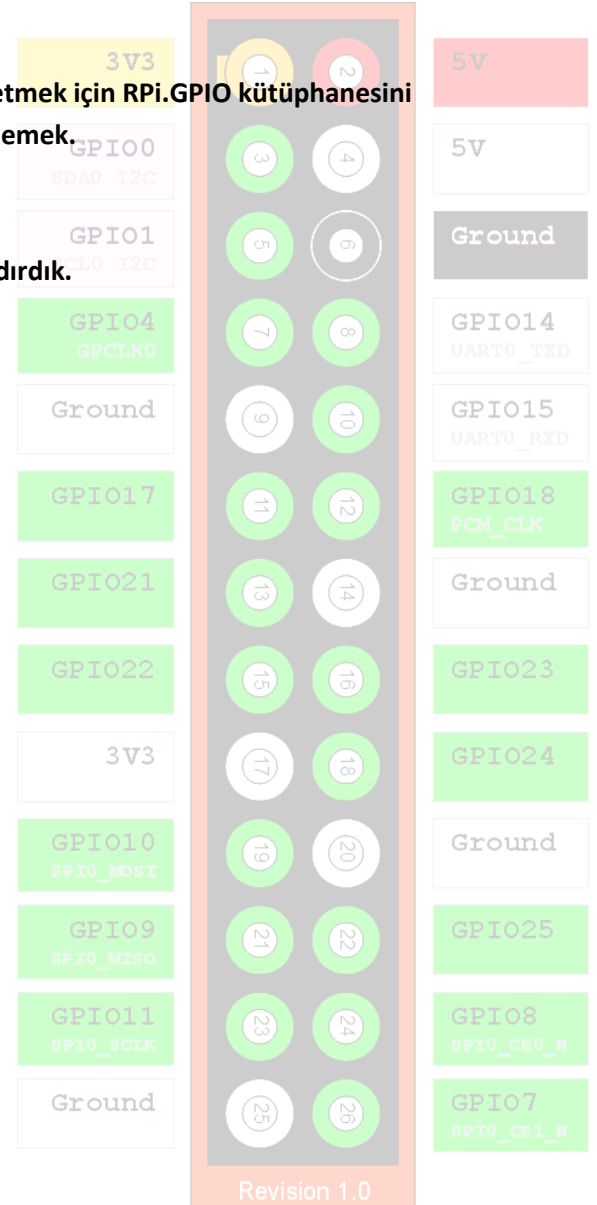
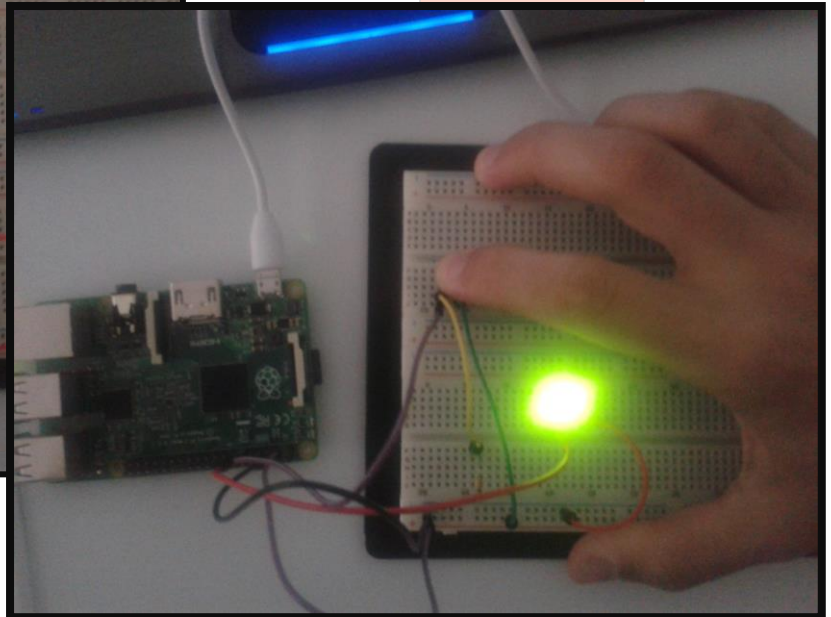
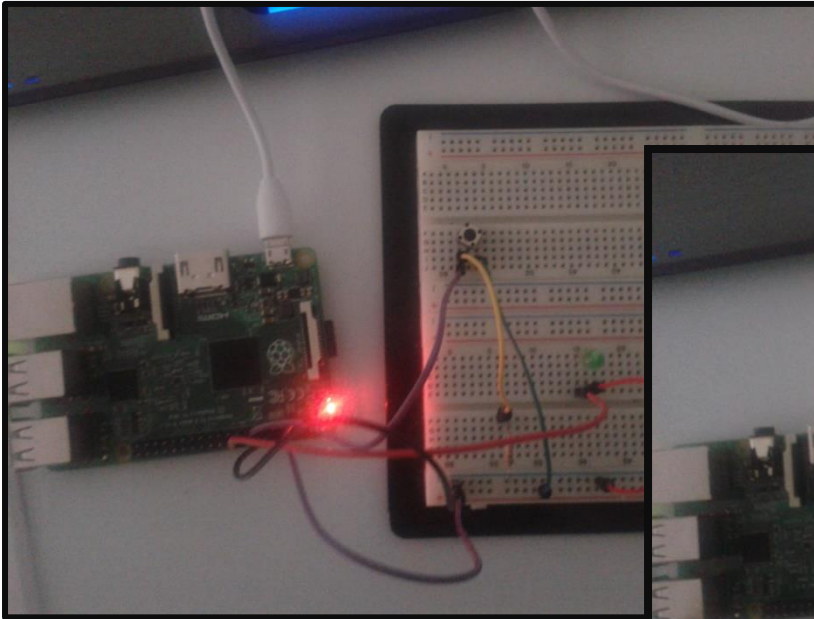
GPIO Pin numaralarını görebilmek ve ayarlamak için

```
G.setMode(G.BOARD)
```

Kodunu kullandık.(yandaki gibi GPIO pin dizilişini görebilirsiniz.)

Kod:

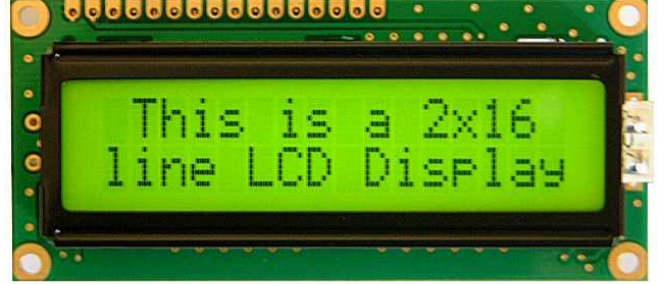
```
import RPi.GPIO as G #GPIO Kütüphanesini ekliyoruz
G.setmode(G.BOARD) #Pinlerimizi ayarlıyoruz
G.setwarnings(False) #Derleyicinin verdiği uyarıları kapatıyoruz.
G.setup(11,G.IN) #11. Pini giriş olarak alıyoruz.
G.setup(16,G.OUT) #16. Pini çıkış olarak alıyoruz.
while(1): #sonsuz döngü
    if(G.input(11) == 0): #11.Giriş 0 ise,
        G.output(16,0) #16.Pini 0 yap.
    else: #11.Giriş 0 Değilse,
        G.output(16,1) #16.Pini 1 yap.
```



Python ile Karakter LCD Kontrolü

Python ile karakter LCD kullanımında yapılacak iki şey ,

1. Karakter lcd pinlerini raspberry mize doğru şekilde bağlamak.
2. Karakter lcdnin kütüphanesinin içeriğini ,fonksiyonlarını iyi öğrenmek.



Raspberry pi için yazılmış kütüphanesine(lcd_16x2.py) ve kullanımına aşağıdaki adresten ulaşabilirsiniz.

<http://www.raspberrypi-spy.co.uk/2012/07/16x2-lcd-module-control-using-python/>

https://github.com/EfecanAltay/RaspberryPi_Character_LCD/blob/master/lcd_16x2.py Bu adresten ise lcd_16x2.py alınan kodları ben genişlettim,eklemeler yaptım. Benim kodlarımı indirerek veya kendinize en uygun kütüphaneyi bularak lcdyi kontrol edebilirsiniz.Ben kendi kodlarım üzerinden anlatacağım.

Kodlarımızı linux işletim sistemi olan raspbianımıza indirmek için;

\$ wget <indirme linki> şeklinde,

örn. (wget https://github.com/EfecanAltay/RaspberryPi_Character_LCD/blob/master/lcd_16x2.py)

indirebilirsiniz.

İndirme işlemi girdiğiniz klasöre gerçekleşecektir.Eğer farklı yere indirmişseniz mv komutunu kullanabilirsiniz.

İndirme tamamlandı;

`$sudo nano lcd_16x2.py` ile kodları görebiliriz veya düzenleyebiliriz.
(Not: kodları nano ile çalıştırırken karakter kodlama hatası verebilir. Bunun için nano ile yazılan kodların başına `# -*- coding: utf-8 -*-` ifadesi eklenmesi gerekebilir.)

`$cat lcd_16x2.py` kodu ise sadece kodu görebildiğimiz methottur.

`$idle lcd_16x2.py` ile de python ın editörünü kullanarak gui olarak kodlarımızı görebiliriz. Biz bu yöntemi(idle)kullanarak kodumuzu derleyip,çalıştıracağız.

Idle ile yazarken Türkçe klavye tanımama sorunumuz olmakta bu yüzden bi çözüm nano ile yazıp düzenleyip idle ile çalıştırmak hatayı görmek.tabiki bu hatayı almayan veya çözen idle ile yazıp güzelce derleyip çalıştırabilir.

Lcd 16x2.py kodlarındaki fonksiyonlar,

```
lcd_init():-----// LCD Kurulumu pinlerini ayarlamak.
lcd_Clear():-----// LCDnin satırlarını temizlemek. *
lcd_CursorON():-----// imleci açmak. *
lcd_CursorOFF():-----// imleci kapatmak. *
lcd_CursorBlink():-----// imleci yanar,söner yapmak. *
lcd_CursorHome():-----// imleci 1.satır ,1.sütun a getirmek. *
lcd_ShiftRight():-----// karakter ekrandaki yazıları 1 birim sağ kaydırmak. *
lcd_ShiftRights(tekrar):-----// karakter ekrandaki yazıları “tekrar birim kadar” sağ kaydırmak.*
lcd_ShiftLefts(tekrar):-----// karakter ekrandaki yazıları “tekrar birim kadar” sola kaydırmak.*
lcd_ShiftLeft():-----// karakter ekrandaki yazıları 1 birim sola kaydırmak.*
lcd_byte(bits, mode):-----// karakter ekranın bitine göre işlem yaptırmayı sağlar.
lcd_toggle_enable():-----// karakter ekranın enable pinini swicth eder.
lcd_string(message,line):-----// String bir mesajı 1. veya 2. Satıra yazdırır.
lcd_ShiftAnimLeft(tekrar,delay):-----//yazıları delay kadar sn aralıklarıyla, tekrar birim kadar sola kaydırmak. *
lcd_ShiftAnimRight(tekrar,delay): -----//yazıları delay kadar sn aralıklarıyla, tekrar birim kadar sağa kaydırmak. *
```

* olanlar benim yazdığım fonksiyonlardır.

Kütüphane kodunu indirdikten sonra kullanması çok basittir yapılması gereken şey sadece yazacağınız kodlara import etmek.

Örn:

```
import lcd_16x2 as lcd //burada lcd kodlarımızı import ettik ve lcd tagıyla kullanmak istediğimizi belirttik
```

ilk çağırılacak fonksiyon init fonksiyonudur.

```
lcd lcd_init()
```

şeklinde portlarımızı pinlerimizi ayarlayıp diğer fonksiyonları kullanabiliriz.

Örn:

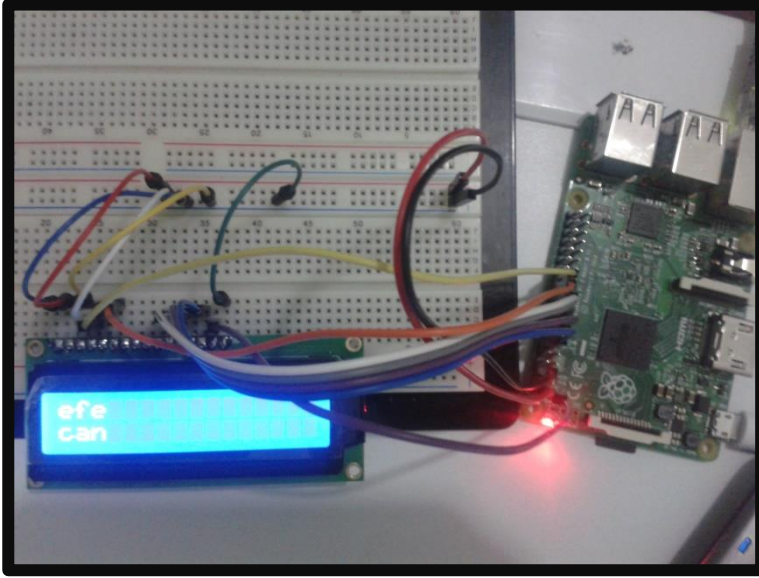
```
lcd lcd_init()
lcd lcd_Clear()
lcd lcd_string('MERHABA',lcd.LCD_LINE_1)
lcd lcd_string('MERHABA',lcd.LCD_LINE_2) şeklinde.
```

Örnek fonksiyonların kullanımı için yazdığım kodlara

https://github.com/EfecanAltay/RaspberryPi_Character_LCD/blob/master/lcd_test.py adresinden ulaşabilirsiniz.


Pinlerinin Bağlanması;

Lcd_16x2.py kodlarının içinde 16x2 pinleri ve hangi Raspberry gpio bcm dizilişine ait bağlanması gerektiğini görebilirsiniz. Yandaki resimde raspberry 2 ye ait gpio pin dizilimi bulunmakta BCM diziliş olarak **GPIOX** olarak kullanılan sıralama kullanılmakta bu dizilişe bakarak kodtaki gerekli pinlerin bağlantılarını gerçekleştirebilirsiniz. Contrast(V0) pinin toprağa çekebilirsiniz.



Raspberry Pi2 GPIO Header

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



A vertical diagram of the Raspberry Pi 2 GPIO header. It consists of 40 pins arranged in two rows of 20. The pins are color-coded: red for 3.3V power (pins 1, 17, 39), black for ground (pins 6, 9, 26, 39), and various colors for GPIO pins (blue, green, yellow, purple, orange). The colors correspond to the pin colors in the table above.

Rev. 1
26/01/2014

<http://www.element14.com>

Rev. 1
26/01/2014

<http://www.element14.com>

Pin bağlamasını tamamladıktan sonra lcd_test.py kodumuzu çalıştırabilir ve lcd'nin çalışmasını test edebilirsiniz.

VLC

VLC yaygın, hızlı ve çok fonksiyonlu multimedia oynatıcıdır. linux komut satırından istediğimiz müziği Raspberrymizde oynatmı sağlayacağız.

Kurulum için:

```
#apt repomuzu güncellemeliyiz.  
$ sudo apt-get update  
#sonra vlc yi direck olarak apt repomuzdan kurabiliriz.  
$ sudo apt-get install vlc
```

EyeD3

Müzik belgelerinin içeriklerini okuyabildiğiniz, yazabildiğiniz ve değiştirebildiğiniz python dili destekli kütüphanedir. projemde eyed3 yi kullandığım için python local reponuzda kütüphaneyi kurmanız gerekmektedir.

Kurulum aşamaları aşağıdaki gibi listelenmiştir.

Eğer easy install ve pip yüklü değilse,

```
#easy_install kurulumu
```

```
$ sudo apt-get install python-setuptools python-dev build-essential
```

```
#Pip kurulumu
```

```
$ sudo easy_install pip
```

```
#EyeD3 Kurulumu
```

```
$ pip install eyed3
```

Python SubProcess Kütüphanesi

Python programlama dili işe işletim sistemimizde alt programlar çalıştırabilmek için kullandığımız python kütüphanesidir.

`import subprocess` şeklinde projemize ekleyip kullanabiliriz.

.call methodu

Call fonksiyonun içinde kullanılan komutların çalışması halinde senkron bir şekilde komutların geri dönüşü beklenir.

Örnek kullanım :

```
>>> subprocess.call(["ls", "-l"]) // burada ls dosyaları listele l is ls nin  
parametresinden biridir.  
>>> 0
```

.check_output methodu

Check output methodu call methodu gibi çalışır fakat geri dönüş değeri byte stringdir. call methodu için geri dönüş aldığımızda

```
>>> subprocess.check_output(["echo", "Hello World!"])  
'Hello World!\n'
```

.Popen

Popen fonksiyonu yeni alt işlem(child process) başlatmaya sağlayan fonksiyondur.bu fonksiyonun geri döndürdüğü process nesnesi ile de başlattığımız alt işlemi yönetebiliriz.

```
#Aşağıda process adında cvlc programı çalışırıldı ve atandı
process = subprocess.Popen(["cvlc" , "-I" , "rc" , "--no-started-from-file","--
playlist-enqueue",muzikKonumum],stdin=run.PIPE)

process.stdin.write("stop\n"); #yukarıda oluşturulan cvlc programına stop komutu
gönderildi.
```

vlc_music.py

Eyed3 ve cvlc yi birleştirip muzikleri fonksiyonlar ile çalışabildiğin adlarını çekebildiğin ve birçok işlemi yapabildiğin python kütüphanesidir.

Kullanımı

kütüphane kullanabilmesi için yukardaki Eyed3 ve vlc yüklü olması gereklidir.

```
import vlc_music.py as vlc
```

şeklinde kütüphane yi projemize dahil edebiliriz.

Fonksiyonlar

setlocate("String Müzik konumu"):-----// Müziklerin Bulunduğu konumu ayarlamak.

updateMusicList():-----// Müzik konumundaki müzik listesini eyed3 ye güncel göstermek.müzik konumunda müzik eklediğimizde veya sildiğimizde bu komudu çalıştırmalıyız.

showMusicList():-----// Müzik konumundaki müzikleri consolda listeleyen komut.

playInList(index):-----// Listedeki giriş indexindeki müziği oynatır.

stopMusic():-----// Çalan müziği tamamen durdur.

playMusic():-----// Durdurulan müziği devam ettirir.

nextMusic():-----// Seçili müzikten sonraki müziği seçer.

lastMusic ():-----// Seçili müzikten önceki müziği seçer.

volumeUp ():-----// Sesi Açar.

volumeDown ():-----// Sesi Kısar.

getAlbum ():-----// O an secili müzikten album adını getirir.

getDocName ():-----//o an secili müziğin string olarak müzik documan adını geri döndürür.

setVolume (value):-----//ses seviyesini value parametresine göre ayarlamızı sağlar.

mp3Player.py

mp3Player.py ana proje kodlarım. Bu kodlarda interuplar ile lcd ve vlc_music kütüphanemi yürütüyorum.bu sayede mp3 player projemi yapmış oluyorum

#mp3Player Kodlarım

```
# -*- coding: utf-8 -*-  
## MPİ2-Player  
import lcd_16x2 as lcd  
import vlc_music as player  
import RPi.GPIO as GPIO  
import time  
  
global volumeValue,play,lcd  
volumeValue = 2  
play = False  
  
kesme_zamani = time.time()  
#buton kesmeleri  
def btn_kesme_next(channel):  
    global kesme_zamani,lcd  
    time_now = time.time()  
    if((time_now - kesme_zamani) >= 1):  
        if(GPIO.input(21) == True):  
            print "next Music"  
            lcd.animReset = True  
            player.nextMusic()  
            lcd.lcd_string("Secili Muzik",lcd.LCD_LINE_1)  
            if(player.getDocName()):  
                lcd.lcd_string(player.getDocName(),lcd.LCD_LINE_2)  
            else:  
                lcd.lcd_string("none",lcd.LCD_LINE_2)  
  
    kesme_zamani = time_now
```

```

def btn_kesme_last(channel):
    global kesme_zamani,lcd
    time_now = time.time()
    if((time_now - kesme_zamani) >= 1):
        if(GPIO.input(20) == True):
            print "last Music"
            lcd.animReset = True
            player.lastMusic()
            lcd.lcd_string("Secili Muzik",lcd.LCD_LINE_1)
            if(player.getDocName()):
                lcd.lcd_string(player.getDocName(),lcd.LCD_LINE_2)
            else:
                lcd.lcd_string("none",lcd.LCD_LINE_2)
        kesme_zamani = time_now

def btn_kesme_volUp(channel):
    global kesme_zamani,volumeValue,lcd
    time_now = time.time()
    if((time_now - kesme_zamani) >= 1):
        if(GPIO.input(26) == True):
            print "VolumeUp"
            lcd.animReset = True
            if(volumeValue < 6):
                volumeValue = volumeValue + 1
                player.volumeUp()
            lcd.lcd_string("Volume",lcd.LCD_LINE_1)
            lcd.lcd_string(str(volumeValue),lcd.LCD_LINE_2)
            time.sleep(1)
            lcd.lcd_string("Secili Muzik",lcd.LCD_LINE_1)
            if(player.getDocName()):
                lcd.lcd_string(player.getDocName(),lcd.LCD_LINE_2)
            else:
                lcd.lcd_string("none",lcd.LCD_LINE_2)
        kesme_zamani = time_now

```

```

def btn_kesme_volDown(channel):
    global kesme_zamani, volumeValue, lcd
    time_now = time.time()
    if((time_now - kesme_zamani) >= 1):
        if(GPIO.input(19) == True):
            print "VolumeDown"
            lcd.animReset = True
            if(volumeValue > 0 ):
                volumeValue = volumeValue - 1
                player.volumeDown()
            else:
                player.setVolume(0)
            lcd lcd_string("Volume", lcd.LCD_LINE_1)
            lcd lcd_string(str(volumeValue), lcd.LCD_LINE_2)
            time.sleep(1)
            lcd lcd_string("Secili Muzik", lcd.LCD_LINE_1)
            if(player.getDocName()):
                lcd lcd_string(player.getDocName(), lcd.LCD_LINE_2)
            else:
                lcd lcd_string("none", lcd.LCD_LINE_2)

    kesme_zamani = time_now

```

```

def btn_kesme_pause(channel):
    global kesme_zamani, volumeValue, lcd, play
    time_now = time.time()
    if((time_now - kesme_zamani) >= 1):
        if(GPIO.input(6) == True):
            print "music pausing"
            lcd.animReset = True
            player.pauseMusic()
            play = False
            lcd lcd_string("Secili Muzik", lcd.LCD_LINE_1)
            if(player.getDocName()):

```



```
    lcd lcd_string(player.getDocName(),lcd.LCD_LINE_2)
```

```
else:
```

```
    lcd lcd_string("none",lcd.LCD_LINE_2)
```

```
kesme_zamani = time_now
```

```
def btn_kesme_play(channel):
```

```
    global kesme_zamani,volumeValue,lcd,play
```

```
    time_now = time.time()
```

```
    if((time_now - kesme_zamani) >= 1):
```

```
        if(GPIO.input(13) == True):
```

```
            print "music playing"
```

```
            lcd.animReset = True
```

```
            play = True
```

```
            player.playMusic()
```

```
            lcd lcd_string("Secili Muzik",lcd.LCD_LINE_1)
```

```
            if(player.getDocName()):
```

```
                lcd lcd_string(player.getDocName(),lcd.LCD_LINE_2)
```

```
            else:
```

```
                lcd lcd_string("none",lcd.LCD_LINE_2)
```

```
kesme_zamani = time_now
```

```
#GPIO.setmode lcd kütüphanesinin içinde var
```

```
GPIO.setup(21, GPIO.IN ,pull_up_down = GPIO.PUD_DOWN)
```

```
GPIO.setup(20, GPIO.IN ,pull_up_down = GPIO.PUD_DOWN)
```

```
GPIO.setup(26, GPIO.IN ,pull_up_down = GPIO.PUD_DOWN)
```

```
GPIO.setup(19, GPIO.IN ,pull_up_down = GPIO.PUD_DOWN)
```

```
GPIO.setup(13, GPIO.IN ,pull_up_down = GPIO.PUD_DOWN)
```

```
GPIO.setup(6, GPIO.IN ,pull_up_down = GPIO.PUD_DOWN)
```

```
#Buton Kesmeleri Ekleme
```

```

#GPIO.add_event_detect(pin_no , pull_up veya pull_down,callback fonksiyonu , milisaniye)
GPIO.add_event_detect(21,GPIO.RISING,btn_kesme_next,300)
GPIO.add_event_detect(20,GPIO.RISING,btn_kesme_last,300)
GPIO.add_event_detect(26,GPIO.RISING,btn_kesme_volUp,300)
GPIO.add_event_detect(19,GPIO.RISING,btn_kesme_volDown,300)
GPIO.add_event_detect(13,GPIO.RISING,btn_kesme_play,300)
GPIO.add_event_detect(6,GPIO.RISING,btn_kesme_pause,300)

#LCD Kurulumu
lcd.lcd_init()
lcd.lcd_cursorOFF()
lcd.lcd_clear()

lcd.lcd_string("MPi3 MP3 Player",lcd.LCD_LINE_1) #Ekranın birinci satırına "MPi3 MP3
Player" yazdırdık.
lcd.lcd_string("  Hosgeldiniz",lcd.LCD_LINE_2) #Ekranın ikinci satırına "hosgeldiniz
" yazdırdık

time.sleep(1) # 1 sn beklettik.

player.updateMusicList() #vlc_music.py kütüphanesine müzikleri tanıttık.

lcd.lcd_clear() #ekranı temizledik
lcd.lcd_string("Secili Muzik :",lcd.LCD_LINE_1) #Ekranın 1. Satırına Secili Müzik
Yazdırdık.
lcd.lcd_string(player.getDocName(),lcd.LCD_LINE_2) #Ekranın ikinci satırına ilk
#gelen müzik dosya adını yazdırdık. İlk çalışırken müzik kütüphanesinde index 1
#olarak ayarlı ve ilk çekilen ilk müzik

def MusicTextAnim(): #Ana Fonksiyonumuzda Çalışan fonk.
    global lcd,play
    while(1):
        if(play):
            lcd.lcd_rowShiftAnimToEnd(lcd.LCD_LINE_2,0.2,1)
    while(1):
        MusicTextAnim()

```