



پاییز 1400

# دوربین مدار بسته

آزمایشگاه سخت افزار

گروه ۱

اعضای گروه:  
پرهام صارمی  
فاطمه خاشعی  
سارا خسروی

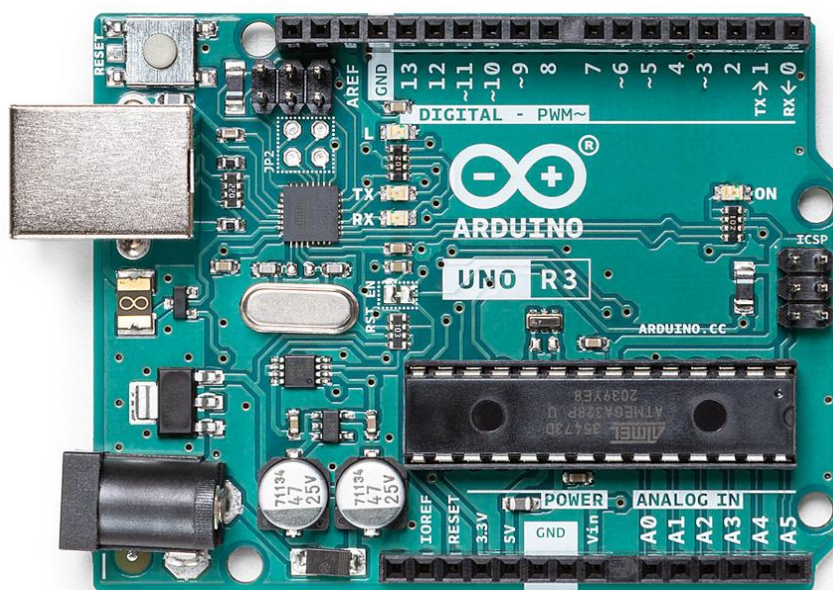
شرح پروژه:

در این پروژه قصد داریم تا با استفاده از قطعاتی برای تشخیص صدا و داشتن تصویر محیط، یک دوربین مدار بسته طراحی کنیم. این پروژه با استفاده از یک برد رزپیری پای انجام میشود. به جهت دریافت تصویر از محیط و کنترل آن، میتوان از سنسور دوربین و یا از تلفن همراه هوشمند استفاده کرد. همچنین به جهت تشخیص صدا، میتوان از قطعه ای که در ادامه برای تشخیص صدا استفاده میشود استفاده نمود و یا با استفاده از یک عدد هندزفری، صدا را تشخیص داد و آن را به مدار اعلام کرد. برای استفاده کاربر نیز میتوان از **mobile app** یا **PWA** استفاده کرد. همچنین به تعدادی سیم نیز به جهت اتصالات مدار نیاز داریم. در این جا لیست قطعات اصلی مورد نیازمان را شرح میدهیم:

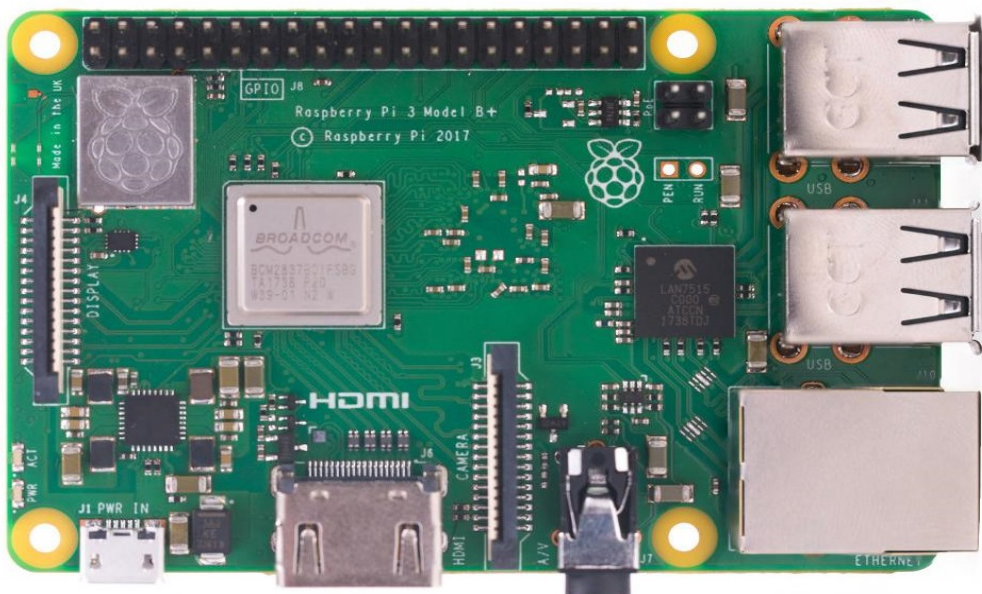
لیست قطعات مورد نیاز:

### (1) برد رزپیری پای

برای انجام این پروژه می توان از آردوینو و یا رزپیری پای استفاده نمود. از آنجا که انجام کار در حالت دوم راحت تر و باکیفیت تر است، از برد رزپیری پای استفاده میکنیم. در ادامه نحوه اتصال ماژولها برای تشخیص صدا و تصویر به این برد توضیح داده میشود.

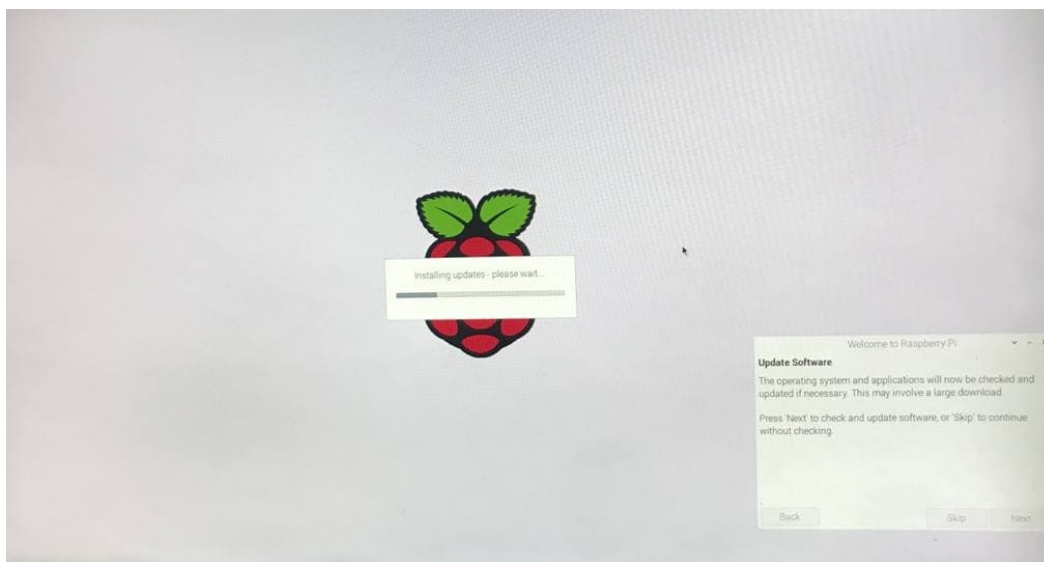


تصویر ۱: برد آردوینو



تصویر ۲ برد رزپبری پای

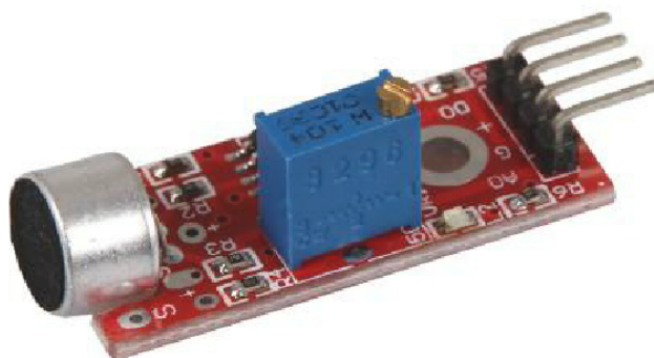
برای استفاده از برد رزپبری پای، سیستم عامل مورد نیاز آن را نیز نصب نمودیم.



تصویر ۳ نصب OS برای رزپبری پای

## (2) سنسور ky-۰۳۷:

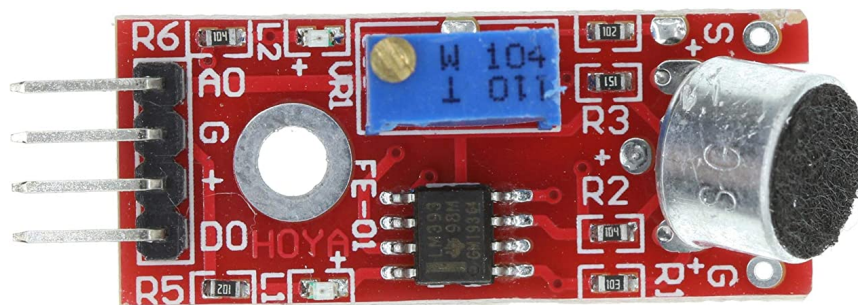
این سنسور برای تشخیص صدا به کار میرود و ابعاد آن 32mm\*17mm\*8mm است. همچنین با ولتاژ ورودی ۳.۳ تا ۵ ولت کار میکند.



تصویر ۴: سنسور ky-037

این سنسور دارای چهار پین AO, DO, GND و VCC است و سیگنال خروجی آن به دو صورت دیجیتال و آنالوگ می‌باشد:

- AO - Analog Output: مقدار این خروجی بر اساس شدت ورودی تغییر می‌کند.
  - DO - Digital Output: این خروجی مثل یک کلید عمل می‌کند و با استفاده از پتانسیومتری که روی سنسور قرار دارد می‌توان مقداری تنظیم کرد که هنگامی که خروجی از آن مقدار بیشتر شود، یک سیگنال توسط این خروجی فرستاده شود.
- برای تشخیص صدا ۲ تا پورت خروجی دارد: DO , AO (میکروفون با حساسیت بالا و میکروفون با حساسیت بالا و نیز پخش نور)



تصویر ۵: پین‌های سنسور ky-037

این سنسور از یک تشخیص دهنده صدا که می‌تواند به صورت آنالوگ و دیجیتال صدا و نیز شدت آن را از محیط تشخیص دهد، تشکیل شده است.

برد این سنسور صدا از ۳ بخش اصلی تشکیل شده است:

- Electret condenser microphone (ECM)

به عنوان سنسور تشخیص صدا استفاده شده و ناحیه سازنده صوت را تشخیص میدهد و یک سیگنال آنالوگ تولید میکند.

- Audio amplifier

سیگنال آنالوگ را از ECM گرفته و آن را بسته به میزان مقاومت پتانسیومتر تقویت میکند و سیگنال را به خروجی آنالوگ این ماژول میفرستد.

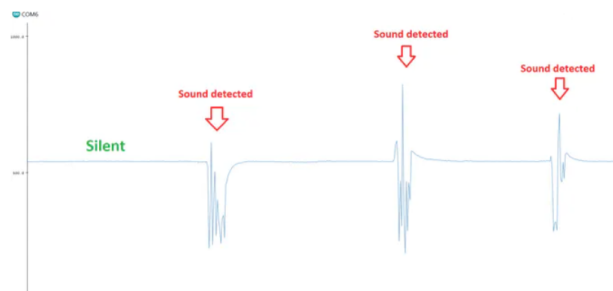
- Comparator

سیگنال تقویت شده صدا را گرفته و با مرجع مقایسه میکند و میزان خروجی را وابسته به آن عوض میکند. در واقع این سنسور زمانی خروجی را به high تغییر میدهد که شدت صدا از یک threshold بگذرد. میزان این threshold با تنظیم کردن پتانسیومتر قابل تعیین است. همچنین ممکن است سیگنال معکوس شود. به این معنا که با گرفتن یک سیگنال با مقدار بیشتر از میزان آستانه تعیین شده، ولتاژ نشان داده شده سطح پایینی داشته باشد.

این ماژول ۲ عدد LED نیز دارد:

- یکی از آنها مشخص میکند که این سنسور powered شده است یا خیر
- دومی مشخص میکند که آیا صدایی از محیط توسط سنسور تشخیص داده شده است یا خیر.

این سنسور به هر دو صورت آنالوگ و دیجیتال میتواند خروجی دهد. ما نیز از خروجی آنالوگ این سنسور استفاده میکنیم و به همین دلیل به یک مبدل آنالوگ به دیجیتال نیاز داریم.



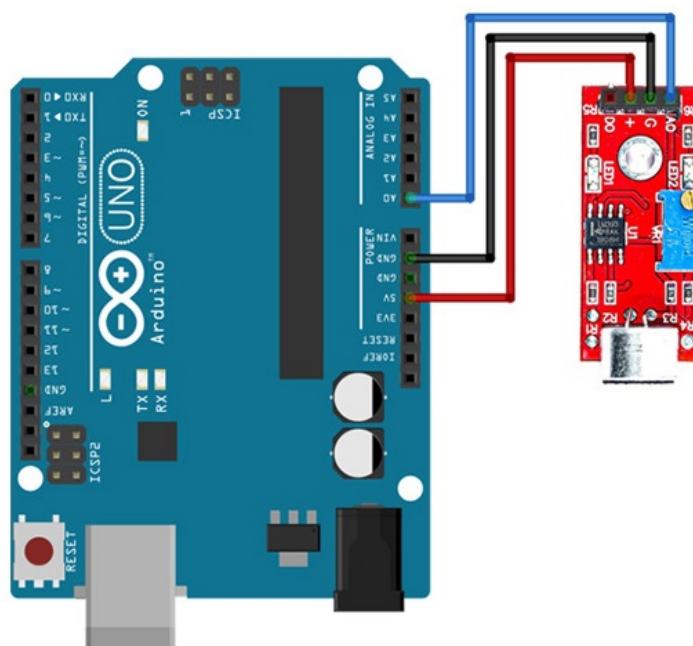
تصویر نمونه خروجی آنالوگ سنسور ky-037

بر خلاف آردوینو، بر روی برد رزبری پای نه دریافت کننده ای برای صدای آنالوگ وجود دارد و نه هیچ تبدیلی برای تبدیل ورودی آنالوگ به دیجیتال. پس چون در رزبری پای ما ADC - Analog Digital Converter نداریم پس در حالت عادی تنها می‌توانیم از خروجی دیجیتال سنسور استفاده کنیم.

اگر بخواهیم از خروجی آنالوگ استفاده کنیم باید با استفاده از یک تبدیل کننده سیگنال‌های آنالوگ را به دیجیتال تبدیل کنیم و سپس به رزبری پای وصل کنیم.

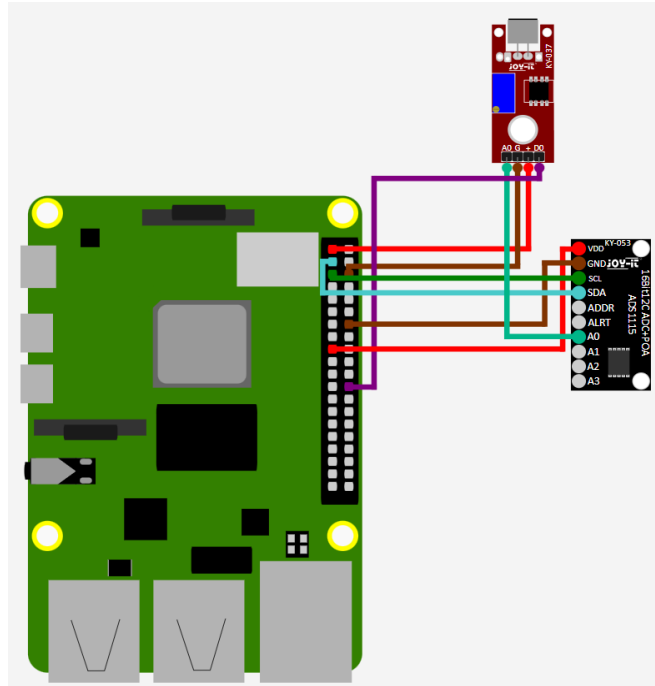
- پین GND سنسور را به GND رزبری پای وصل می‌کنیم.
- پین VCC سنسور را به PIN۲ رزبری پای وصل می‌کنیم.
- پین DO سنسور را به GPIO۲۳ - PIN 33 وصل می‌کنیم.

نحوه اتصال به برد آردوینو:



تصویر ۷ نحوه اتصال سنسور ky-037 به برد آردوینو

نحوه اتصال به برد رزبری پای:

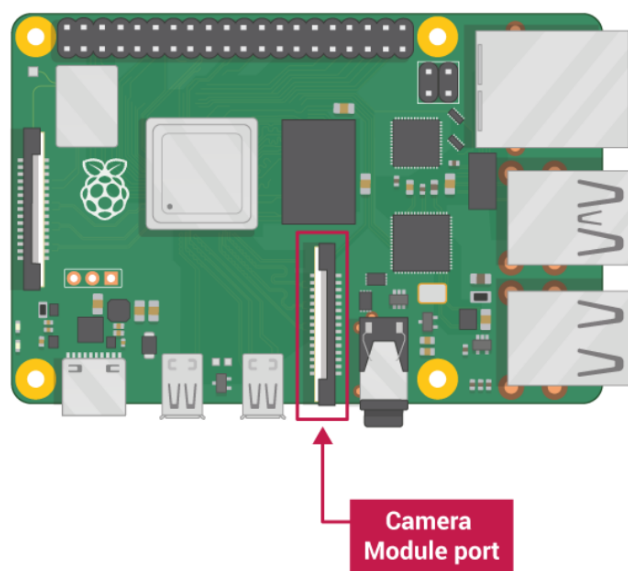
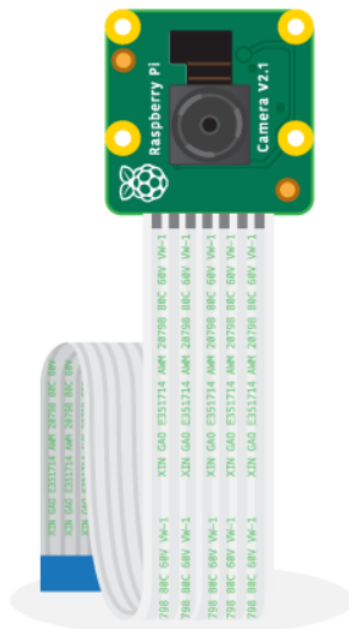
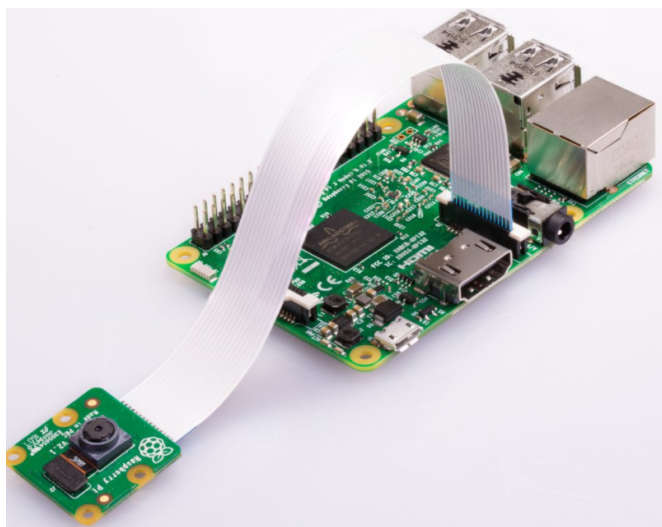


تصویر ۸ نحوه اتصال سنسور *ky-037* به برد رزپبری پای

### (3) دوربین

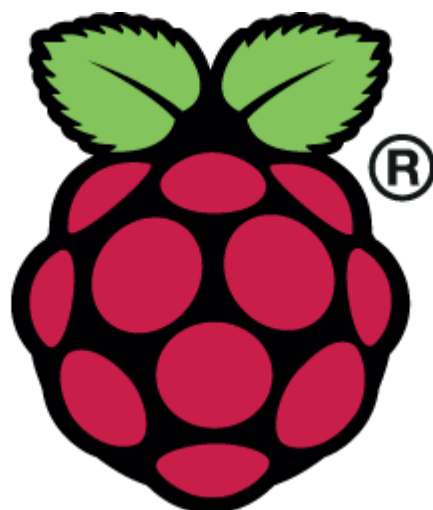
برای تشخیص و ضبط تصویر محیط، همانطور که پیشتر گفته شد دو راه وجود دارد: استفاده از دوربین تلفن همراه و نیز استفاده از *picamera*. در صورت استفاده از این ماژول از واسط پایتون استفاده میشود. همچنین میتوان آن را روی تلفن همراه پیاده کرد.





تصویر ۹: مازول تشخیص تصویر و نحوه اتصال آن به برد رزپیری پای







## نصب Raspbian

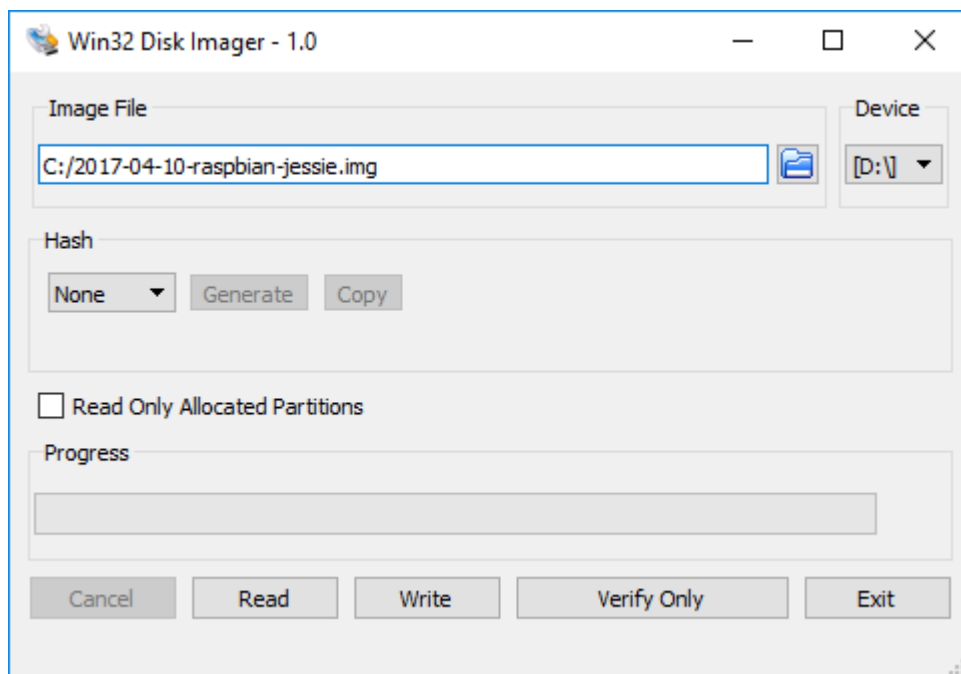
برای شروع به استفاده از برد Raspberry Pi نیاز به نصب سیستم عامل روی آن داریم. در این قسمت توضیح می‌دهیم که چگونه سیستم عامل Raspbian را روی برد Raspberry Pi نصب کنیم. برای این کار به یک کارت microSD، موس، کیبورد، مانیتور و منبع برق نیاز داریم.

مرحله ۱: دانلود Raspbian

RASPBIAN JESSIE WITH PIXEL		RASPBIAN JESSIE LITE	
	Image with PIXEL desktop based on Debian Jessie		Minimal image based on Debian Jessie
Version:	April 2017	Version:	April 2017
Release date:	2017-04-10	Release date:	2017-04-10
Kernel version:	4.4	Kernel version:	4.4
Release notes:	<a href="#">Link</a>	Release notes:	<a href="#">Link</a>
<a href="#">Download Torrent</a> <a href="#">Download ZIP</a>		<a href="#">Download Torrent</a> <a href="#">Download ZIP</a>	
SHA-1:	6d7b11bb3d64524203edf6c80c499456fb5fef53	SHA-1:	c24a4c7dd1a5957f303193fee712d0d2c0c6372d

باید به این سایت برویم و Disk Image سیستم عامل Raspbian را دانلود کنیم.

مرحله ۲: قرار دادن فایل Disk Image در microSD



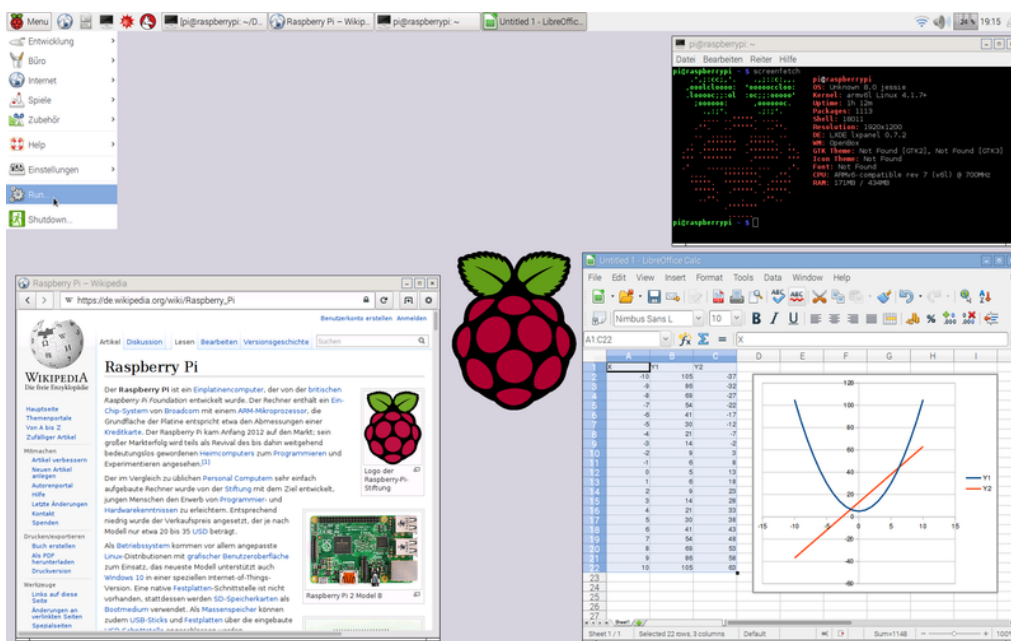
در ابتدا باید فایل دانلود شده را unzip کنیم. سپس فایل با پسوند .img را روی microSD بنویسیم برای این کار در سیستم عامل ویندوز از برنامه‌ی Win32 Disk Imager، در مک از برنامه‌ی disk utility که در مک قرار دارد و در ویندوز از Etcher استفاده می‌کنیم.

مرحله ۳: قرار دادن microSD در Raspberry Pi و روشن کردن آن  
پس از روشن کردن مراحل نصب Raspbian به صورت اتومات انجام می‌شود. username دیفالت pi و password دیفالت raspberry می‌باشد.

ماژول‌ها

(1) ماژول تشخیص صدا:

کد این بخش در ریپازیتوری موجود است. اما اینجا نیز کد ماژول صدا را آورده و توضیح می‌دهیم:



ابتدا کتابخانه‌های مورد نیاز را در پایتون نصب میکنیم. این کار مانند نصب هر کتابخانه دیگر با استفاده از مثلا pip در پایتون صورت میگیرد.

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

سپس پورتهای را تنظیم میکنیم. برای اینکار از حالت BCM استفاده میکنیم و مود برد را روی این حالت میگذاریم. سپس برای هر سیگنال یک پورت تعیین میکنیم. برای صدا به پورت ۱۷ و برای روشن شدن LED به پورت ۲۷ وصل میشویم.

```
#GPIO SETUP
sound = 17
led = 27
#set pin modes
GPIO.setmode(GPIO.BCM)
```

سپس مشخص میکنیم هر سیگنال و پین آن در مورد ورودی باشد یا خروجی. در اینجا ماژول ما تشخیص صدا است و میخواهیم در صورتی که صدایی تشخیص داده شد، چراغ LED روشن شود. پس سیگنال صدا را ورودی و سیگنال نور را خروجی تعریف میکنیم. به هریک از این ورودی و خروجی‌ها در پایتون کانال گفته میشود.

```
#set channels as input and output
GPIO.setup(sound, GPIO.IN)
GPIO.setup(led,GPIO.OUT)
```

حال در تابع callback مقدار سیگنال را چک میکنیم. اگر مقدار سیگنال بالا یا زیاد بود چراغ را روشن میکنیم. برای این حالت به مقدار HIGH یا روشن یا بالا را میدهیم. اگر کم بود به مقدار LED مقدار پایین یا LOW یا خاموش را میدهیم.

```
def callback(sound):
    if GPIO.input(sound):
        print("sound detected")
        GPIO.output(led,HIGH)
    else:
        GPIO.output(led,LOW)
```

یک event\_detect تعریف میکنیم تا بالا رفتن یا پایین آمدن یا به طور کلی هر تغییری در سیگنال صدا را تشخیص دهد. در واقع event\_detect مقدار سیگنال را گرفته و به تابع callback میفرستد. لازم به ذکر است که ترد این بخش از برنامه مجزا از ترد برنامه اصلی است.

```
#detect when the pin goes HIGH or LOW
GPIO.add_event_detect(sound, GPIO.BOTH, bouncetime=300)
# assign function to GPIO PIN, Run function on change
GPIO.add_event_callback(sound, callback)

# infinite loop
while True:
    time.sleep(1)
```

## (2) مازول تشخیص تصویر:

کد این بخش در ریپازیتوری موجود است. اما اینجا نیز کد مازول تصویر را آورده و توضیح میدهیم: در مرحله اول برای راهاندازی دوربین از قطعه کد زیر استفاده می کنیم.

```
()camera = PiCamera
camera.resolution = (2592, 1944)
camera.framerate = 15
```

```
camera.brightness = 70
```

دقت شود که رزولوشن برای فیلمبرداری باید برابر  $1080 \times 1920$  قرار داده شود. برای فیلمبرداری در این رزولوشن باید framerate برابر با ۱۵ قرار داده شود. روشنایی تصویر در حالت عادی برابر ۵۰ است و می‌تواند عددی بین ۰ تا ۱۰۰ را دریافت کند.

```
camera.start_preview(alpha = 200)
```

قطعه کد بالا تصویر دوربین را به ما نشان می‌دهد. با افزایش مقدار آلفا می‌توان شفافیت را افزایش داد.

```
sleep(5)
```

```
camera.capture('/home/pi/Desktop/image_normal.jpg')
```

با استفاده از قطعه کد بالا می‌توان یک عکس را در مکان دلخواه ذخیره کرد.

```
camera.rotation = 90
```

```
sleep(5)
```

```
camera.capture('/home/pi/Desktop/image_90.jpg')
```

```
camera.rotation = 180
```

```
sleep(5)
```

```
camera.capture('/home/pi/Desktop/image_180.jpg')
```

```
camera.rotation = 270
```

```
sleep(5)
```

```
camera.capture('/home/pi/Desktop/image_270.jpg')
```

```
camera.rotation = 0
```

```
sleep(5)
```

با استفاده از قطعه کد بالا عکس را هر دفعه ۹۰ درجه می‌چرخانیم و یک عکس را ذخیره می‌کنیم. دوربین قابلیت این را دارد که در چهار حالت ۰، ۹۰، ۱۸۰ و ۲۷۰ درجه تصویر را نمایش دهد. چرخش با مقداردهی rotation انجام می‌شود.

```
camera.start_recording('/home/pi/Desktop/video.h264')
```

```
sleep(25)
```

```
()camera.stop_recording
```

با استفاده از قطعه کد بالا می‌توان ویدیو را به مدت دلخواه ضبط کرد و در مکان خواسته شده ذخیره کرد. کد بالا این کار را به مدت ۲۵ ثانیه انجام می‌دهد.

همانطور که گفته شد می‌توان مقدار روشنایی را با مقداردهی brightness تغییر داد. کد زیر مقدار روشنایی را به آرامی از ۰ تا ۱۰۰ تغییر می‌دهد و در نهایت پس از ۵ ثانیه به مقدار نرمال برمی‌گرداند.

```
:for i in range(100)
```

```
    camera.brightness = i
```

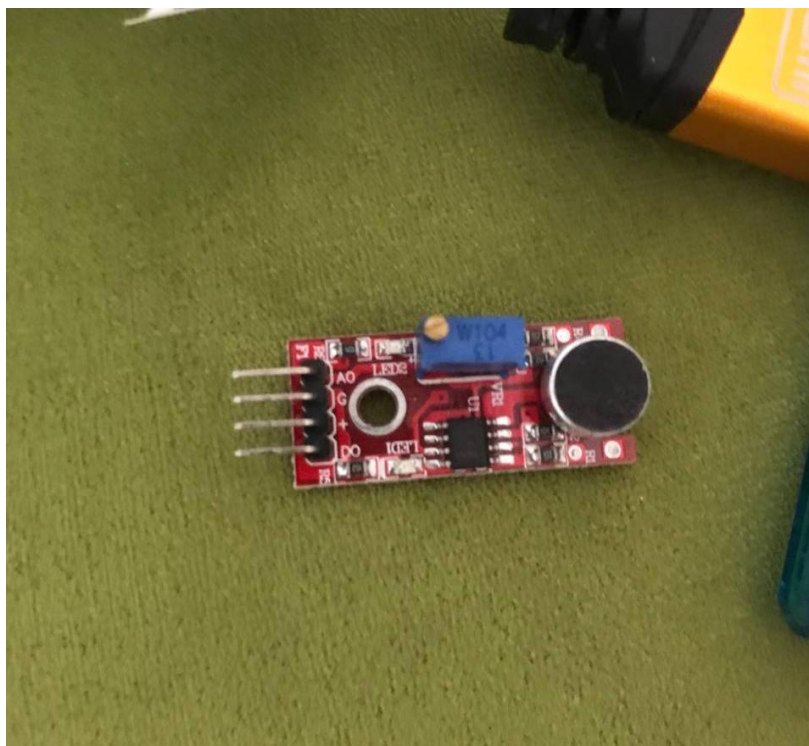
```
    sleep(0.1)
```

```
sleep(5)  
camera.brightness = 50
```

## ۴ تست سخت‌افزاری

در ابتدا برد را به مانیتور و موس و کیبورد وصل کرده و آنرا روشن می‌کنیم.

### ۴.۱ تست میکروفن



سنسور تشخیص صدا

برای این کار همانطور که در قسمت کدهای میکروفن گفته شده، پین ۱۷ را به عنوان پین ورودی صدا در نظر می‌گیریم. پین GND سنسور را به GND رزبری پای وصل می‌کنیم. پین VCC سنسور را به PIN۲ رزبری پای وصل می‌کنیم. پین DO سنسور را به GPIO۱۷ یا همان PIN۱۱ وصل می‌کنیم.

ابتدا با استفاده از pip کتابخانه‌های مورد نیاز را در پایتون نصب می‌کنیم.

سپس کد را اجرا می‌کنیم.

مشاهده می‌شود که پس از اجرا با یک Threshold مشخص این میکروفن قابلیت تشخیص صدا را دارد.



وصل کردن سنسور به raspberryPi

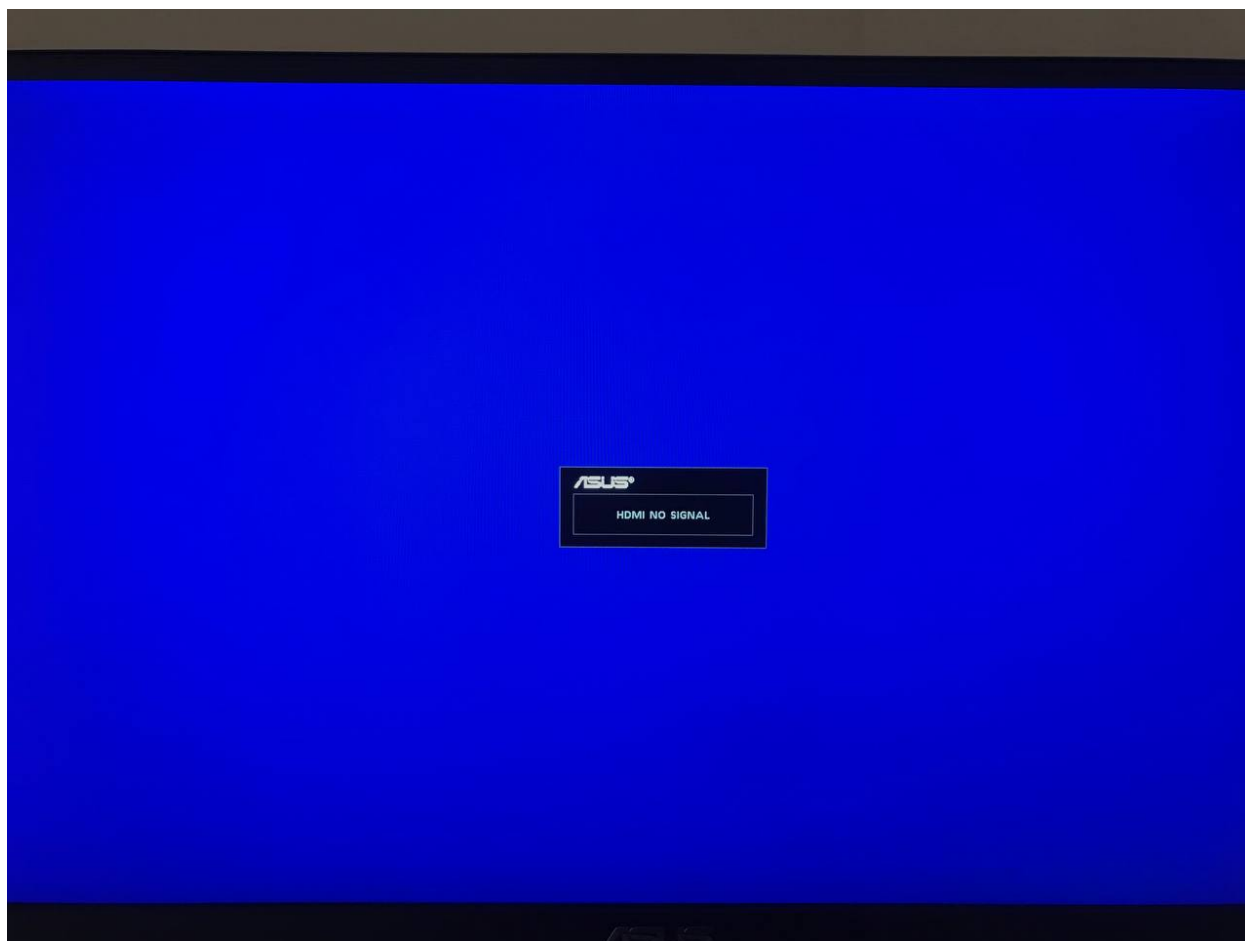
۴.۱ تست دوربین



سنسور دوربین



برای تست دوربین، دوربین را به Raspberry Pi وصل کردیم اما پس از آن OS دچار اختلال شد و پس از خاموش کردن Raspberry Pi دیگر روشن نشد.  
مانیتور خطای No HDMI Signal Found می‌داد.



خطای مانیتور

برد هم با وصل کردن به برق فقط چراغ قرمز آن روشن می‌شد که نشان‌دهنده‌ی آن است که به منبع برق وصل است و چراغ سبز روشن نمی‌شود به این معنا که برد نمی‌تواند SD Card موجود را بخواند، ابتدا SD Card را درآوردیم و تمیز کردیم اما جواب نداد، سپس تصمیم به فرمت SD Card و نصب دوباره‌ی OS گرفتیم که در این مرحله تمام partition های SD Card را دوباره به هم متصل کردیم و آنرا فرمت کردیم

سپس image سیستم عامل Raspbian را روی آن ریختم و دوباره SD Card را در برد قرار دادم و ایندفعه هم کار نکرد.



روشن شدن چراغ قرمز برد

۵ تست ماژول ها

۵.۱ میکروفن

۵.۲ دوربین

۵.۲.۱ ماژول دوربین

```
class Camera:
    def __init__(self, resolution, framerate, brightness = 70): ...
    def calc_video_capacity(self): ...
    def check_video_capacity(self): ...
    def start(self): ...
    def stop(self): ...
    def capture_picture(self): ...
    def start_recording(self): ...
    def stop_recording(self): ...
```

ماژول دوربین

یک **class** به نام **camera** داریم که امکان کانفیگ کردن ویژگی‌های دوربین مانند رزولوشن، گرفتن عکس، گرفتن فیلم و ... را برای ما فراهم می‌کند. عکس‌ها و فیلم‌ها در فولدر **module\_camera** در دسکتاپ ذخیره خواهند شد. فرآیند ذخیره به این صورت است که هر ۳۰ ثانیه یک فیلم ذخیره می‌شود و پس از رسیدن به محدودیت حافظه، اولین ویدیو از نظر زمانی حذف خواهد شد. کارایی هر کدام از تابع‌ها در زیر به تفکیک گفته شده:

۱ **\_\_init\_\_**

این تابع همان **constructor** بوده و کانفیگ مورد نظر را ورودی گرفته و محاسباتی از قبیل محدودیت حافظه را انجام می‌دهد.

**capacity\_video\_calc ۲**

این تابع محدودیت حافظه را محاسبه می‌کند، در واقع به ما می‌گوید که ما چقدر تا فیلم می‌توانیم در raspberry pi ذخیره کنیم.

**capacity\_video\_check ۳**

این تابع وضعیت حافظه را چک می‌کند و اگر به محدودیت محاسبه‌شده رسیده باشیم اولین ویدیو گرفته شده از نظر زمانی را حذف می‌کند.

**start ۴**

دوربین آماده شروع به کار می‌شود.

**stop ۵**

دوربین خاموش می‌شود.

**picutre\_capture ۶**

در صورتی که دوربین روشن باشد، یک عکس گرفته و ذخیره می‌کند.

**recording\_start ۷**

در صورتی که دوربین روشن باشد، ویدیو گرفتن شروع می‌شود.

**recording\_stop ۸**

در صورتی که دوربین روشن باشد، ویدیو گرفتن به اتمام می‌رسد.

۵.۲.۲ تست مازول دوربین