

پاییز 1400

دوربین مداربسته

آزمایشگاه سخت افزار

گروه 1

اعضای گروه:

پرهام صارمی

سارا خسروی

فاطمه خاشعی

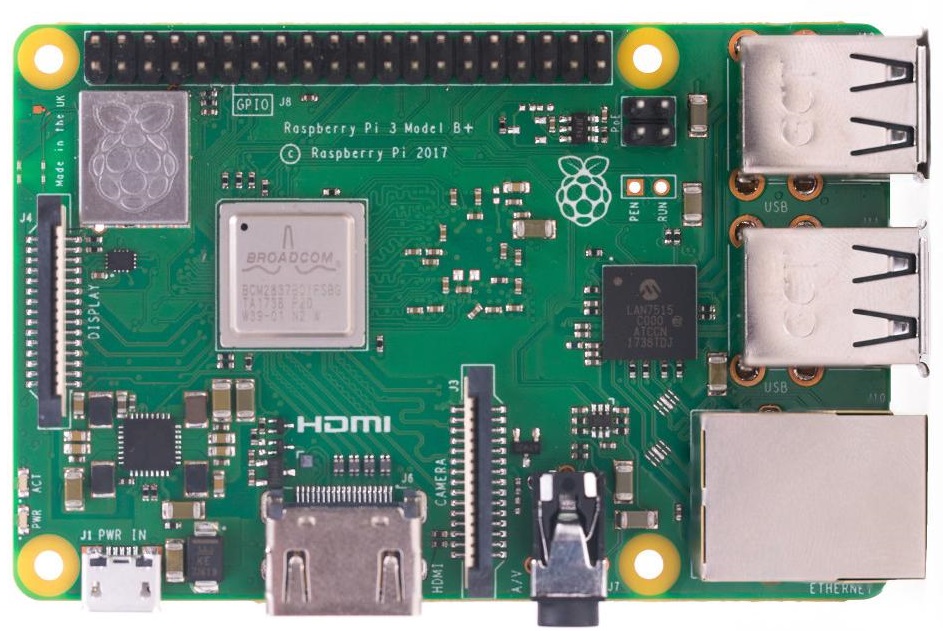
**شرح پروژه:**

در این پروژه قصد داریم تا با استفاده از قطعاتی برای تشخیص صدا و داشتن تصویر محیط، یک دوربین مداربسته طراحی کنیم. برای انجام پردازش‌های این پروژه و در وافع داشتن یک کامپیوتر ککوچک برای کنترل دستورات، می‌توان از آردویینو یا رزپبری‌پای استفاده نمود. از آنجا که انجام کار در حالت دوم راحت‌تر و باکیفیت‌تر است، از برد رزپبری‌پای استفاده می‌کنیم. این برد توان محایباتی بالاتری داشته و همچنین کارهای پیچیده‌تر از پروژه را نیز می‌توان با آن به انجام رساند. برای ضبط ویدئو یا گرفتن تصویر از محیط و کنترل اوضاع، می‌توان از یک سنسور دوربین یا تلفن همراه هوشمند استفاده کرد. در این‌جا از یک ماژول دوربین استفاده شده است. همچنین به جهت تشخیص صدای محیط، می‌توان از سنسور تشخیص صدا استفاده نمود و یا با استفاده از یک عدد هندزفری، صدا را تشخیص داد و آن را به مدار اعلام کرد. در این پروژه راه اول در پیش گرفته شده است. برای استفاده‌ی راحتتر کاربر نیز می‌توان از mobile app یا PWA استفاده کرد که ما یک اپلیکیشن موبایل ساخته‌ایم. همچنین به تعدادی سیم ماده-ماده نیز به جهت اتصالات مدار (از هر سنسور به برد رزپ‌بری) نیاز داریم. در این جا لیست قطعات اصلی مورد نیازمان را به همراه نحوه کارکرد آن‌ها شرح می‌دهیم:

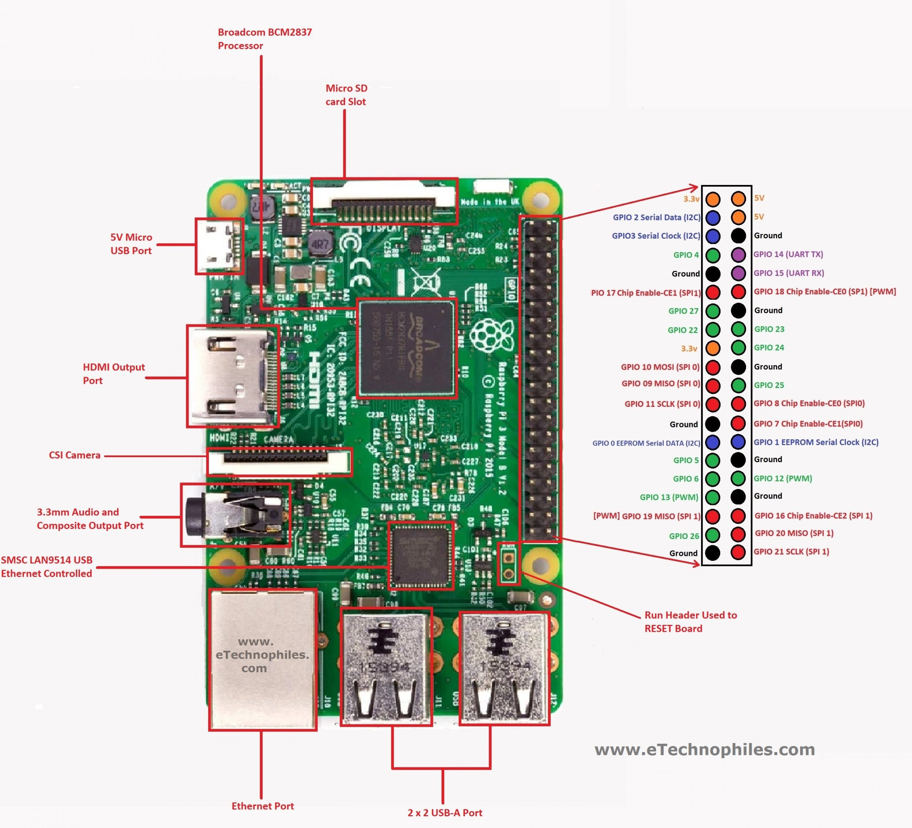
**لیست قطعات مورد نیاز:**

1. برد رزپبری‌پای

اصلی‌ترین قطعه بکار رفته در این پروژه که مانند یک کامپیوتر کوچک عمل می‌کند، برد رزپ‌بری‌پای است. در ادامه نحوه‌ اتصال هر یک از ماژول‌ها برای تشخیص صدا و تصویر به این برد توضیح داده می‌شود. مدل رزپ‌بری‌پای ما 3B+ است. برای آن که بدانیم چطور از آن استفاده کنیم و هر سنسور را به کدام پین متصل کنیم، دیتاشیت برد را مطالعه می‌کنیم.

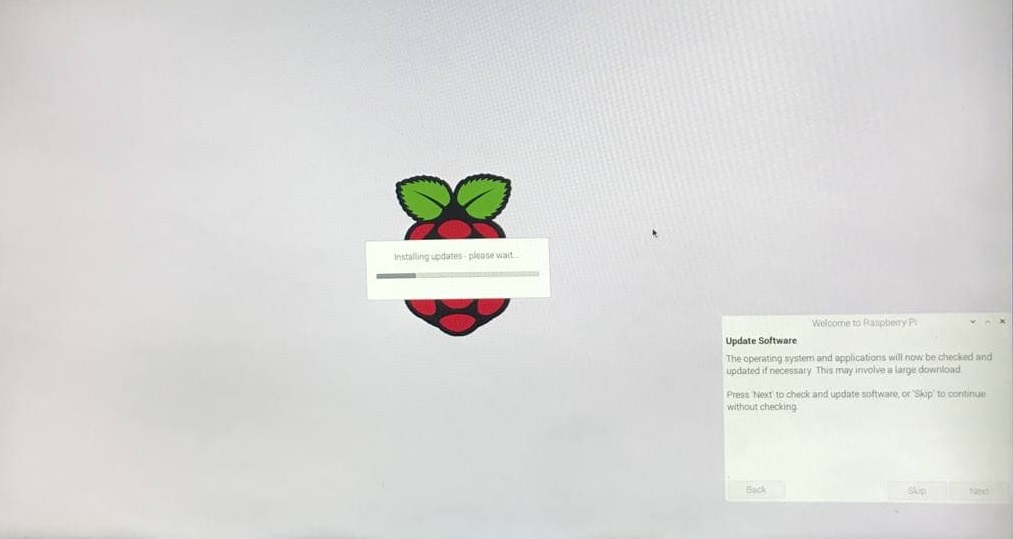


تصویر 1: برد رزپبری‌پای



تصویر2: دیتاشیت برد رزپبری‌پای

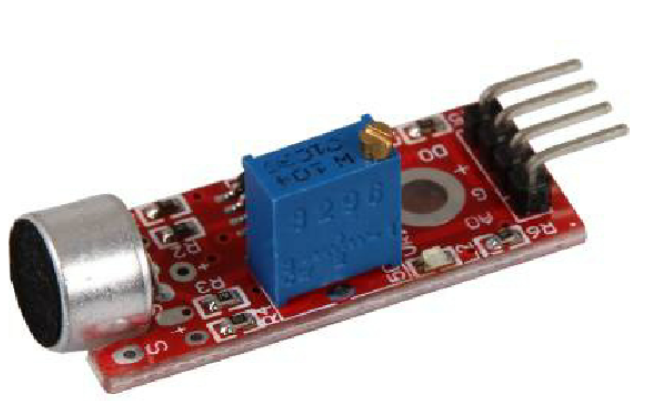
برای استفاده از برد رزپبری‌پای، نخست باید سیستم‌عامل مورد نیاز آن را نیز نصب کنیم تا سیستم بالا بیاید.



تصویر 3: نصب os برای رزپبری پای

1. سنسور ky037:

این سنسور برای تشخیص صدا به کار میرود و ابعاد آن32mm\*17mm\*8mm است. همچنین با ولتاژ ورودی 3.3 تا 5 ولت کار می‌کند.



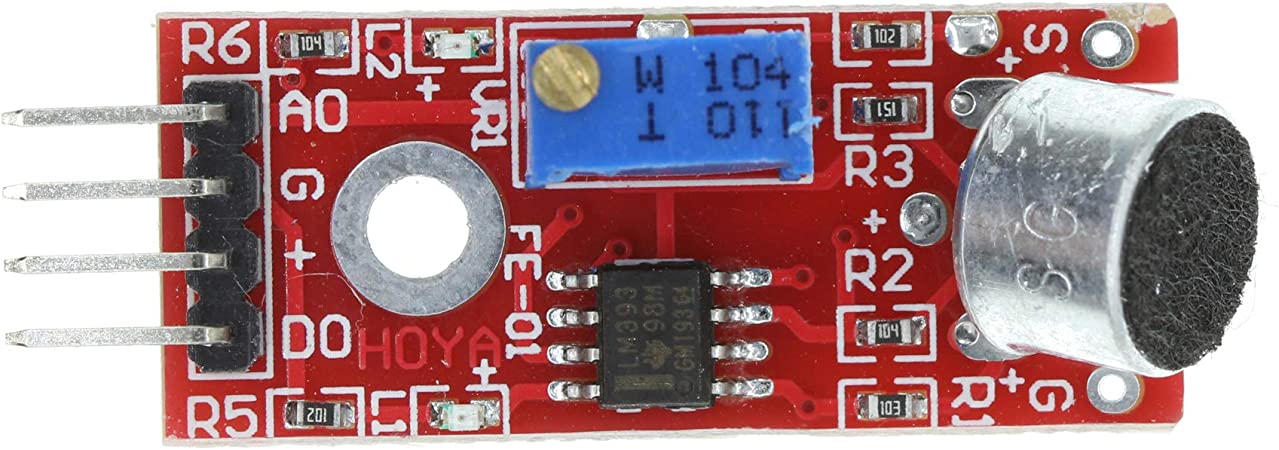
تصویر4: سنسورky-037

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 2 تا پورت خروجی دارد: AO , DO (میکروفون با حساسیت

بالا و میکروفون با حساسیت بالا و نیز پخش نور)



تصویر5:پین‌های سنسورky-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 3 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 thresholdی بگذرد.

این ماژول 2 عدد چراغ نیز دارد:

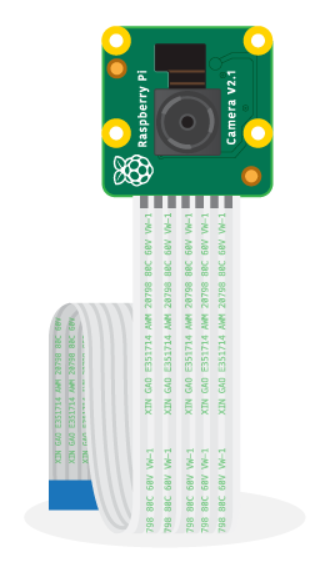
* یکی از آنها مشخص میکند که آیا توان لازم برای روشن شدن و کار کردن در اختیار سنسور قرار گرفته است یا خیر. در صورت تامین منبع انرژی لازم برای این سنسور، این چراغ (قرمز) روشن می‌شود.
* دیگری مشخص میکند که آیا صدایی از محیط توسط سنسور تشخیص داده شده است یا خیر. در صورت تشخیص صدا توسط سنسور از محیط اطراف، رنگ این نیز قرمز می‌شود.

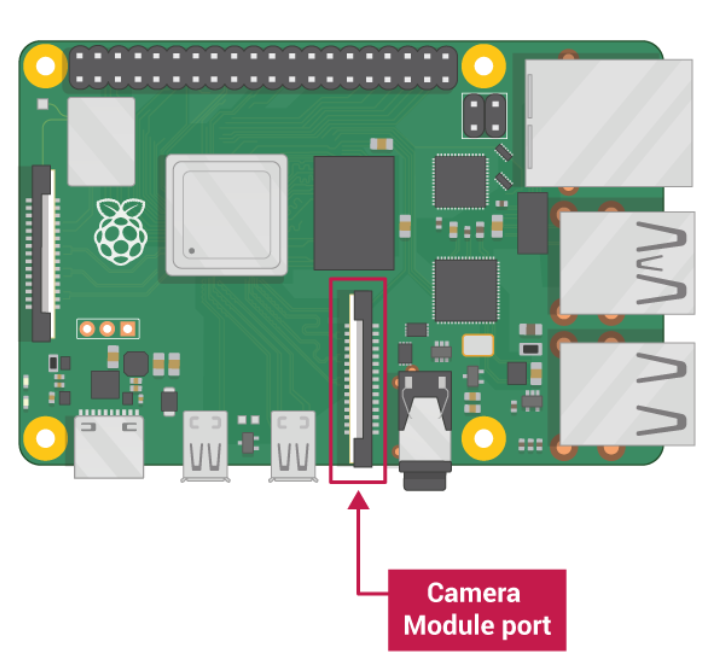
لازم به ذکر است این سنسور به هر دو صورت آنالوگ و دیجیتال می‌تواند خروجی دهد. بر خلاف آردوئینو، بر روی بورد رزپبری‌پای نه دریافت کننده‌ای برای صدای آنالوگ وجود دارد و نه هیچ مبدلی برای تبدیل ورودی آنالوگ به دیجیتال. پس چون در رزبری‌پای ما ADC - Analog Digital Converter نداریم پس در حالت عادی تنها می‌توانیم از خروجی دیجیتال سنسور استفاده کنیم. لذا دیگر به مبدل آنالوگ به دیجیتال نیاز نداریم. همچنین ما تنها پین مربوط به حالت دیجیتال را به یکی از پین‌های مربوط به ورودی یا خروجی در رزپ‌بری‌پای وصل می‌کنیم (برای مثال pin شماره 17) و الزامی به اتصال پین آنالوگ به برد نیست. اما در حالت سخت‌تر، 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 پین GND رزبری‌پای وصل می‌کنیم.
* پین VCC سنسور را به PIN دوم که 5 ولت است وصل می‌کنیم.
* پین DO سنسور را که ورودی و خروجی دیجیتال آن است به یکی از پین‌های ورودی خروجی برد رزپ‌بری پای (برای مثال پین 17) وصل می‌کنیم.

1. 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 زبان برنامه‌نویسی پایتون برای برنامه‌ دادن به آن استفاده می‌شود.





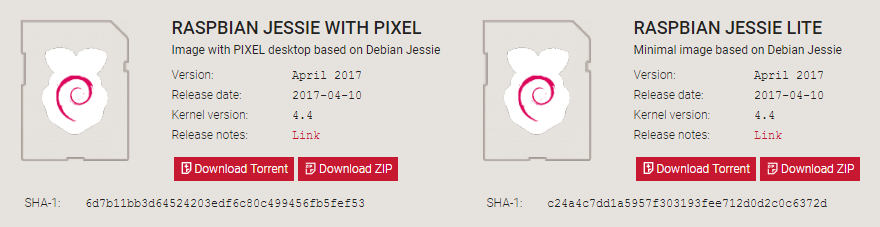
تصویر 6: ماژول تشخیص تصویر و نحوه اتصال آن به برد رزپبری پای

تصویر7: لوگوری رزپبری پای

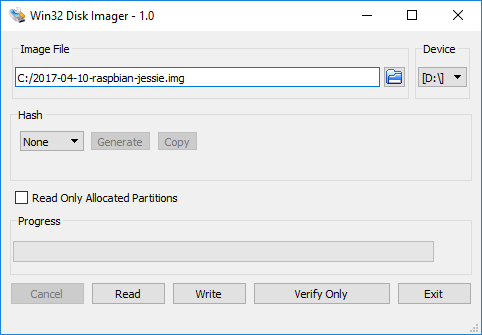
### **نصب Raspbian**

برای شروع به استفاده از برد Raspberry Pi نیاز به نصب سیستم‌عامل روی آن داریم. در این قسمت توضیح می‌دهیم که چگونه سیستم‌عامل Raspbian را روی برد Raspberry Pi نصب کنیم. برای این کار به یک کارت حافظه‌ی microSD، موس، کیبورد، مانیتور و منبع برق نیاز داریم. کابل USBموس و کیبورد را به طور مستقیم به پورت USB برد رزپ‌بری نصب می‌کنیم. همچنین با استفاده از یک کابل HDMI یک مانیتور یا تلویزیون را به برد متصل می‌کنیم. با یک کابل شارژ گوشی نیز برد را به برق شهر وصل کردیم تا توان مصرفی آن را تامین کند.

مرحله ۱: دانلود Raspbian

باید به [این](https://www.raspberrypi.com/software/) سایت برویم و Disk Image سیستم عامل Raspbian را دانلود کنیم.

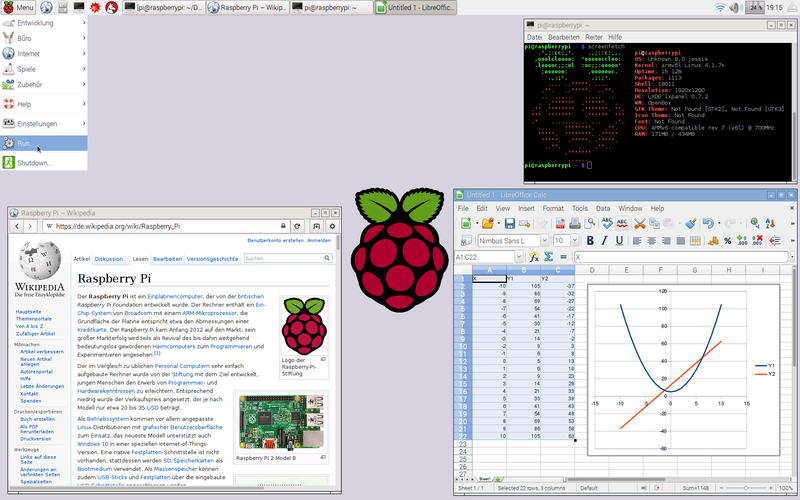
تصویر 8: لینکهای دانلود raspbien

مرحله ۲: قرار دادن فایل Disk Image در microSD

تصویر : 9نمایی از برنامه‌ی Win32 Disk Imager

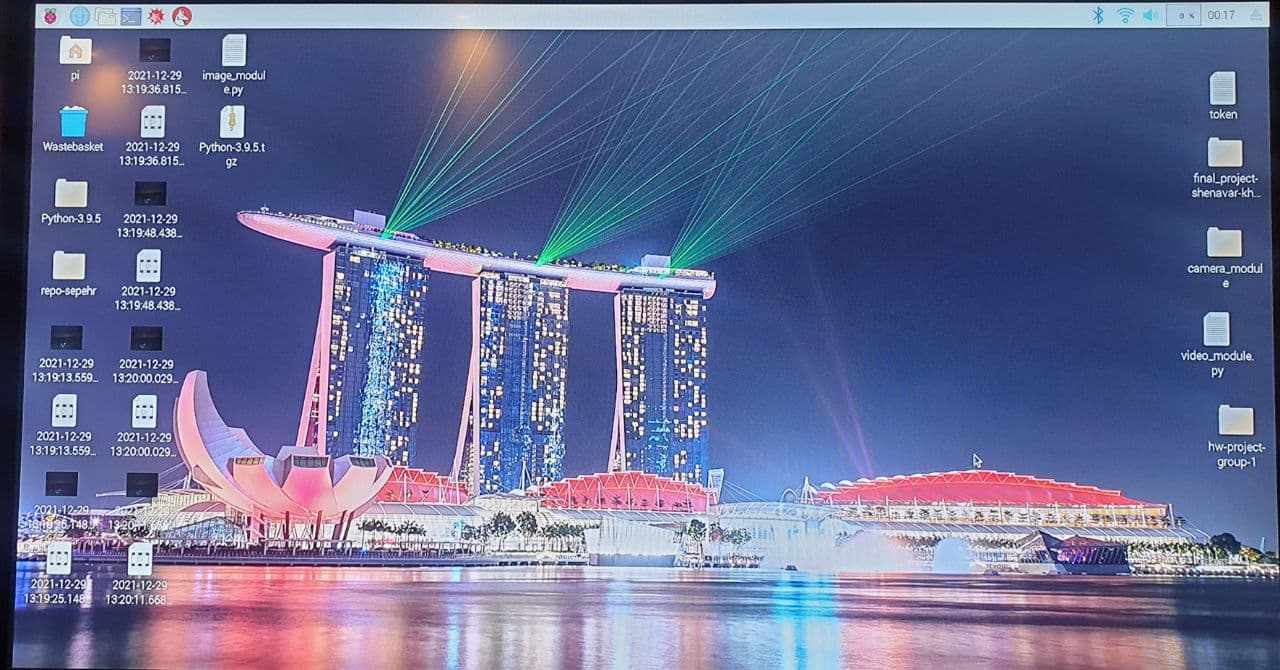
در ابتدا باید فایل دانلود شده را unzip کنیم. سپس فایل با پسوند .img را روی microSD بنویسیم. برای این کار در سیستم عامل ویندوز از برنامه‌ی [Win32 Disk Imager](https://sourceforge.net/projects/win32diskimager/) و در مک از برنامه‌ی disk utility استفاده می‌کنیم.

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 پیش‌فرض pi و password پیش‌فرض raspberry می‌باشد. برای استفاده راحت‌تر، پسورد را به 1234 تغییر دادیم.

تصویر 10نمایی از Raspbian

در نهایت باید برد را به برق متصل کرده و نیز برای راحتی نمایش اطلاعات می‌توان برد را به یک نمایشگر همانند تلویزیون متصل کرد. این کار با استفاده از یک کابل HDMIکه به پورت HDMI از برد وصل می‌شود انجام می‌گردد.



تصویر 11: نمایی از Raspbian متصل شده به تلویزیون

تصویر مدار کلی ما را که در آن نحوه‌ی اتصال ماژول‌های تشخیص صدا و ضبط تصویر مشخص است می‌توانید در زیر ملاحظه کنید. لازم به ذکر است که دو کابلی که در سمت راست به برد متصل هستند، کابل‌های USB مربوط به ماوس و کیبورد هستند که بتوان با این برد و سیستم‌عامل کار کرد.



تصویر 12: شکل کامل مدار تشخیص و ضبط صدا و تصویر

## **ماژول ها**

1. ماژول تشخیص صدا:

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

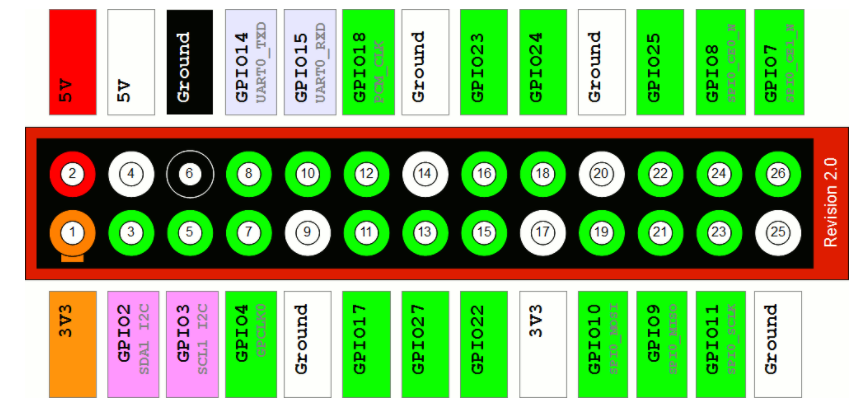
ابتدا کتابخانه‌های مورد نیاز را در پایتون نصب می‌کنیم. این کار مانند نصب هر کتابخانه دیگر با استفاده از مثلا pip در پایتون صورت می‌گیرد.

**import** RPi.GPIO **as** GPIO  
**import** time

سپس پورت‌ها 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 17 که یکی از پورت‌های ورودی رزپبری پای است متصل می‌شویم.

*#GPIO SETUP*sound = 17  
*#set pin modes*GPIO.setmode(GPIO.BCM)

لازم به ذکر است برای حالت برد 2 نوع وجود دارد. حالت اول GPIO.BOARD است که در این حالت به پین‌های روی برد بر حسب شماره‌ای که روی برد درج شده است اشاره می‌شود. حالت دوم که ما نیز از آن استفاده کرده‌ایم حالت GPIO.BCM است. در این حالت به هر پین بر اساس Broadcom SOC channel آن اشاره می‌شود که این‌ها شماره‌هایی هستند که در شکل زیر در خانه‌های سبز رنگ پس از عبارت GPIO مشخص شده‌اند. این نوع شماره‌گذاری بسته به ورژن هر برد عوض می‌شود.



تصویر 13: شماره‌گذاری بر حسب BCM

در ادامه مشخص می‌کنیم هر سیگنال و pin آن درboard ورودی باشد یا خروجی. در اینجا ماژول ما تشخیص صدا است. پس سیگنال صدا را ورودی 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)

حال در تابع callback مقدار سیگنال را چک می‌کنیم. اگر مقدار سیگنال دچار تغییر شد (در واقع این ماژول لبه را تشخیص میدهد) صدا تشخیص داده میشود و HIGH یا TRUE که وضعیت pin ورودی ماست چاپ می‌شود.

**def** callback(self):  
 **if** GPIO.input(self.sound):  
 print(**"Sound Detected!"**)  
 **else**:  
 print(**"Sound Detected!"**)  
 print(GPIO.input(self.sound))

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 thread این بخش از برنامه مجزا از thread برنامه اصلی است که در ادامه در کلاس AlarmThread که از Thread ارث‌بری می‌کند آورده شده است و با فراخوانی تابع run آن، اجرا می‌شود.

*# let us know when the pin goes HIGH or LOW*GPIO.add\_event\_detect(self.sound, GPIO.BOTH, self.callback, bouncetime=300)  
*# assign function to GPIO PIN, Run function on change*GPIO.add\_event\_callback(self.sound, self.callback)  
*##detect events*GPIO.event\_detected(self.sound)

**while True**:  
 print(GPIO.event\_detected(self.sound))  
 time.sleep(1)  
 **if** self.send **and** GPIO.event\_detected(self.sound):  
 print(**"Detected"**)  
 send\_email(\*get\_mail\_values())  
 self.send = **False**

1. ماژول تشخیص تصویر:

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

در مرحله اول برای راه‌اندازی دوربین از قطعه کد زیر استفاده می‌کنیم.

camera = PiCamera()

camera.resolution = (2592, 1944)

camera.framerate = 15

camera.brightness = 70

دقت شود که resolution برای فیلم‌برداری باید برابر 1080\*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

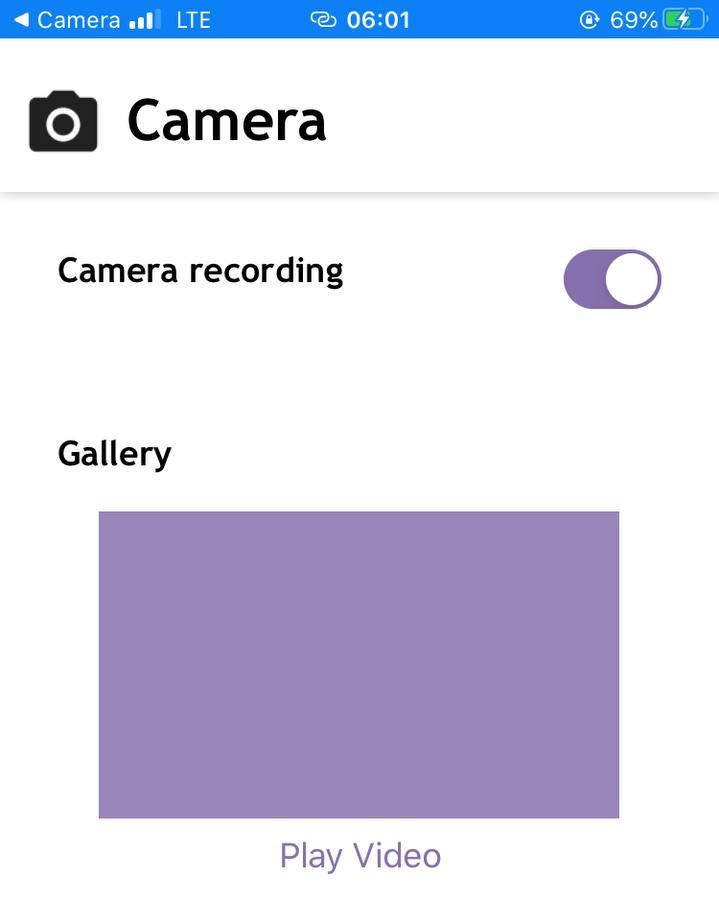
sleep(0.1)

sleep(5)

camera.brightness = 50

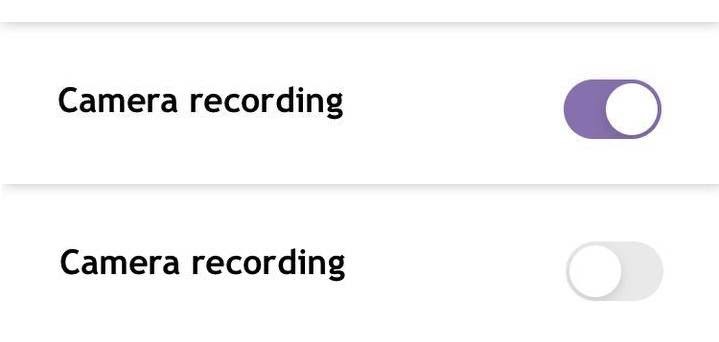
**توضیحات کد:**

### **اپ موبایل:**



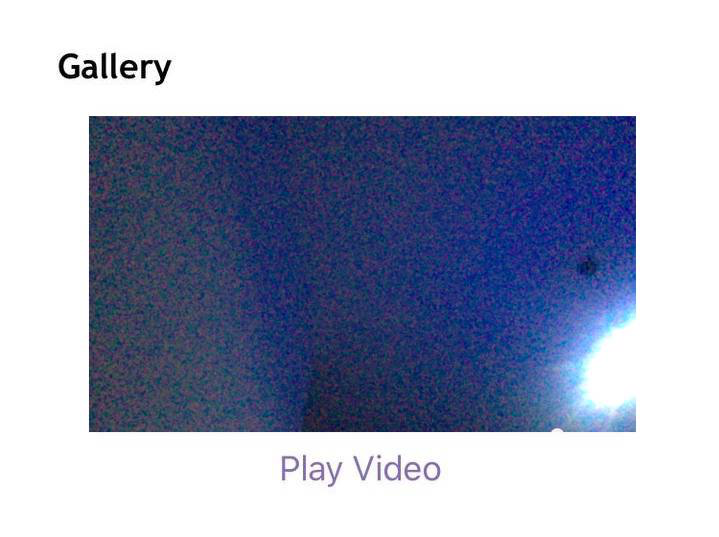
تصویر14: نمایی از اپ موبایل

به طور کلی این اپ با استفاده از react native پیاده شده است که میتوان از آن 2 تا خروجی برای اندروید و ios گرفت. در اپ موبایل عملکرد‌های پیاده‌سازی شده در سرور توسط کاربر قابل انجام می‌شوند. برای روشن و خاموش کردن دوربین از یک Switch استفاده کرده‌ایم که کاربر با استفاده از آن می‌تواند روی فرآیند ضبط مدیریت داشته باشد. در صورت فشردن دکمه recording در این اپ موبایل، برنامه شروع به ضبط می‌کند و خروجی را در گالری دخیره می‌نماید.



تصویر 15: قابلیت روشن و خاموش کردن دوربین در اپ موبایل

همچنین در صورت خاموش کردن دکمه alarm، دیگر در صورت کشف صدا و تصویر برای فرد ایمیل نمی‌رود. این پروژه به این صورت است که در بازه‌‌های زمانی مشخصی مثلا هر ۳۰ ثانیه یک ویدیو گرفته و ذخیره می‌کند و با کلیک روی show video آن را به ما نمایش می‌دهد. همچنین هر 5 ثانیه یک‌بار نیز یک عکس گرفته و در گالری ذخیره می‌شود. کاربر می‌تواند با عقب و جلو رفتن در این گالری عکس‌های گرفته شده در زمان‌های مختلف را مشاهده کند و اگر مایل بود ویدئو مربوط به عکس‌ها را مشاهده کند. لازم به ذکر است که زیر عکس‌ها یک دکمه‌ی مشاهده ویدئو وجود دارد که این امکان را برای او فراهم می‌آورد.



تصویر 16: گالری و مشاهده‌ی عکس‌های گرفته شده

# **کد سرور:**

به طور کلی این بخش با استفاده از flask پیاده‌سازی شده است. کد سرور دارای چند بخش کلی (apiها، کد مربوط به ماژول‌ها، کد ارسال ایمیل) می‌باشد که در ادامه توضیحات هر بخش آمده است. پروژه‌ی ما شامل یک token می‌باشد که در هنگام ساخت یک قطعه به صورت منحصر به فرد برای قطعه قرار داده می‌شود. و در اپلیکیشن مختص به کاربر این توکن قرار می‌گیرد به این صورت فقط کاربر می‌تواند با سیستم خود ارتباط برقرار کند و امنیت دستگاه تامین می‌شود.

## **بخش apiها:**

در این بخش به توضیح ویژگی‌های مختلف سرور می‌پردازیم:

شروع ضبط: با استفاده از api این بخش (در مسیر start\_recording) در سرور ضبط فیلم شروع می‌شود که توضیحات آن در بخش ماژول‌ها آمده است. این کار در سرور با استفاده از threadها انجام می‌شود. به این صورت که یک thread ساخته می‌شود که وظیفه آن ضبط فیلم است. این thread قابلیت توقف و ادامه ضبط را دارد که این کار با استفاده از lockها انجام می‌شود.

توقف ضبط: با استفاده از این api می‌توان در روند ضبط وقفه ایجاد کرد. نحوه پیاده‌سازی این ویژگی در قسمت قبل گفته شده.

حذف فایل: در این api با دادن نام یکی از ویدیو‌ها می‌توان آن را از مجموعه‌ی فایل‌های سرور حذف کرد. (مسیر delete\_file/name)

لیست فایل‌ها: در این api با مسیر (get\_files) لیست تمامی ویدیو‌های ضبط شده را می‌توان دریافت کرد. این لیست شامل نام فایل‌ها و عکس وسط ویدیو می‌باشد که کاربر با استفاده از آن‌ها می‌تواند تصمیم بگیرد کدام ویدیو را تماشا کند.

تماشای ویدیو: در این api با مسیر (show\_file/name) می‌توان با دادن نام یک فایل آن را به صورت stream تماشا کرد.

خاموش کردن alarm: در این مسیر می‌توان قابلیت ارسال ایمیل اطلاع‌رسانی به کاربر را غیر فعال کرد.

روشن کردن alarm: در این مسیر می‌توان قابلیت ارسال ایمیل اطلاع‌رسانی به کاربر را غیرفعال کرد.

## **بخش ماژول‌ها:**

## **بخش ایمیل:**

محصول ما قابلیت ارسال ایمیل اطلاع‌رسانی به کاربر در صورت روشن بودن alarm را دارد. در این حالت اگر صدایی توسط ماژول شناسایی شد یک ایمیل هشدار به کاربر ارسال می‌شود تا در جریان صدا ناخواسته باشد. برای ارسال ایمیل از کتابخانه‌ی smtplib استفاده شده است.

**اتصال ماژول‌ها به بک‌اند:**

برای آنکه ماژول‌های تشخیص صدا و تصویر و ضبط ویدئو و گرفتن یک عکس از محیط به طور پیوسته در حال کار باشند، لازم است که در یک Thread به طور دائمی فراخوانی شوند. بدین منظور برای هر یک از threadهای تشخیص صدا و ضبط تصویر یک کلاس ساخته شده است. در init هر کلاس تنظیمات اولیه اعم از اینکه پین ورودی رزپبری‌پای چه باشد، مقدار اولیه متغیرها مثل sound یا send چه باشد و ... تنظیم شده‌اند. سپس مطابق کدهای هفته سوم که در پوشه Week 3از مخزن قرار دارند (البته آپدیت شده‌اند ولی باز هم آنجا قرار گرفته‌اند) ماژول‌های تشخیص صدا و تشخیص تصویر را در threadها قرار می‌دهیم. نحوه کار این ماژول‌ها همانند قبل است و می‌توانید به توضیحات قبلی رجوع کنید. تنها کافیست تا هر بخشی که در ماژول‌های قبلی داخل حلقه قرار داشته اینجا نیز داخل حلقه بگذاریم، هر بخشی هم که در خارج آن بوده است را پیش از حلقه while بگذاریم.

سپس برای اجرای این ماژول‌ها، باید در مرورگر پس از آدرس ip داده شده، دستور مدنظر را بزنیم و token مدنظر را به آن بدهیم تا در نهایت تصویر به فرمت .mp4 ذخیره شود. تشخیص صدا نیز تنها به صورت یک پرینت به نمایش در میاید. البته طوری تنظیم شده است که اگر در ابتدا send درست مقداردهی شود (True)، آنگاه در صورت تشخیص صدا یک ایمیل به ایمیل آدرس داده شده ارسال می‌گردد.