

پاییز 1400

دوربین مداربسته

آزمایشگاه سخت افزار

گروه 1

اعضای گروه:

پرهام صارمی

سارا خسروی

فاطمه خاشعی

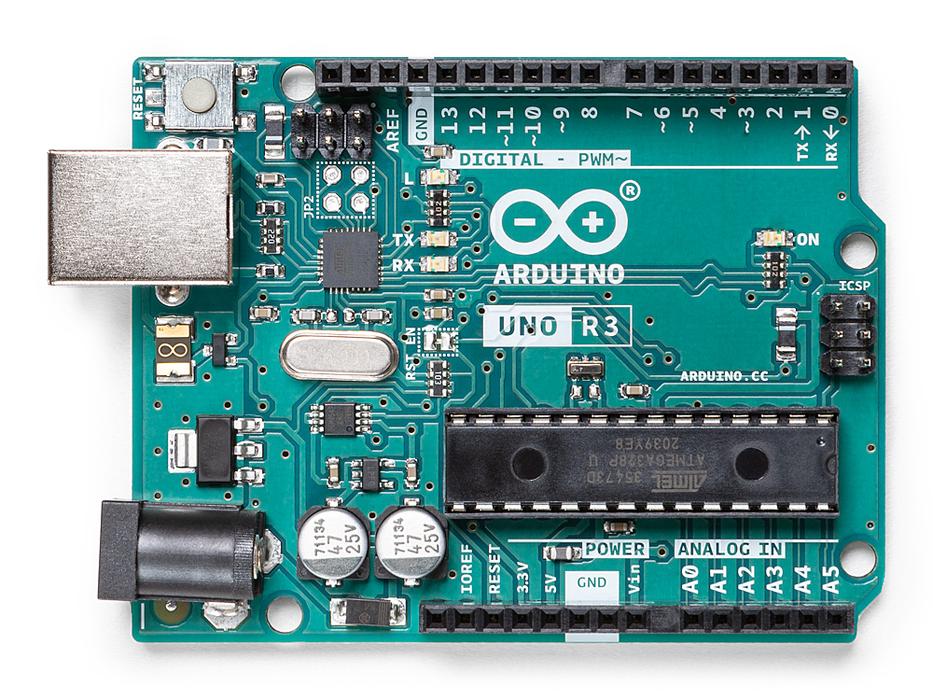
**شرح پروژه:**

در این پروژه قصد داریم تا با استفاده از قطعاتی برای تشخیص صدا و داشتن تصویر محیط، یک دوربین مدار بسته طراحی کنیم. این پروژه با استفاده از یک برد رزپبری 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:

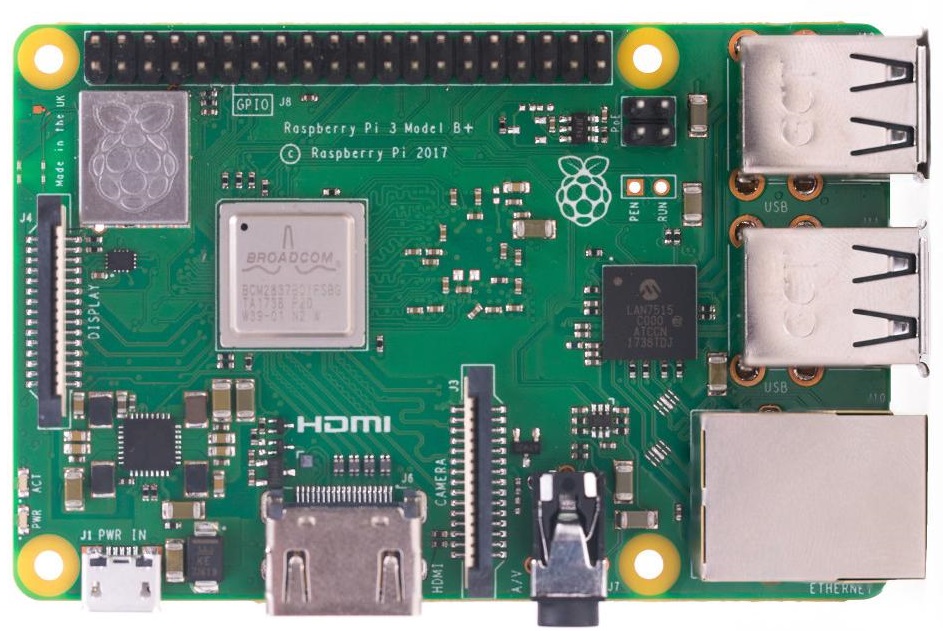
**لیست قطعات مورد نیاز:**

1. برد رزپبری پای

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

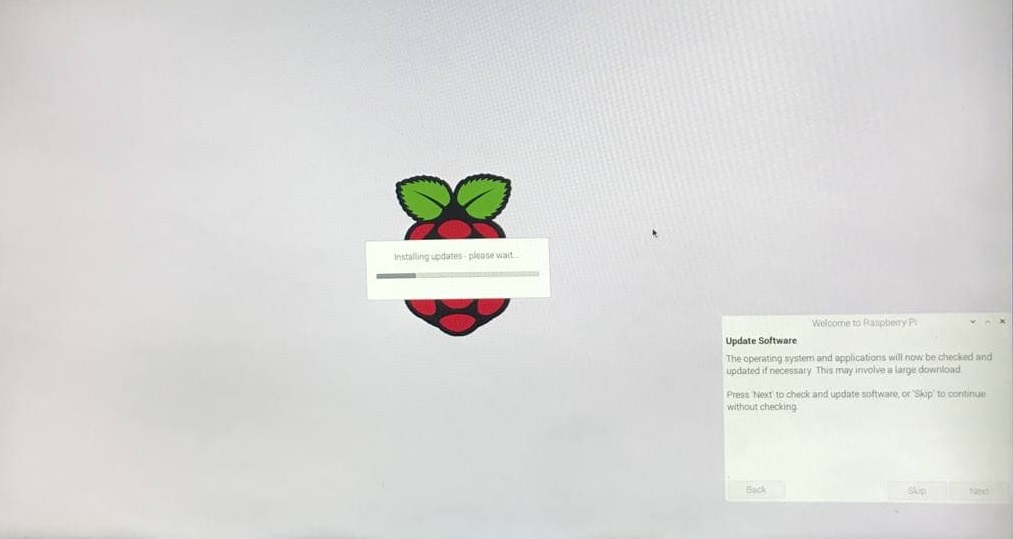


تصویر 1: برد آردویینو



تصویر 2برد رزپبری پای

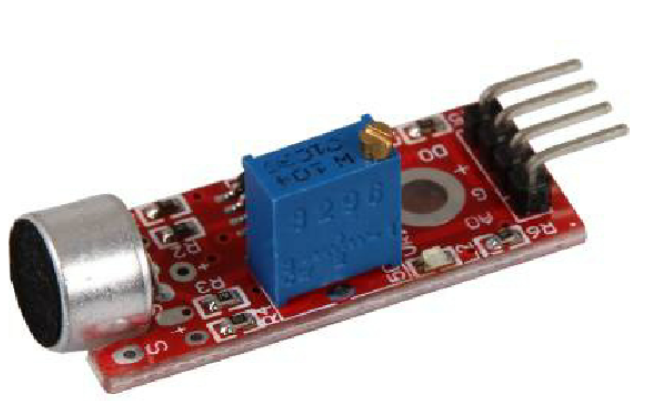
برای استفاده از برد رزپبری پای، سیستم عامل مورد نیاز آن را نیز نصب نمودیم.



تصویر 3نصب os برای رزپبری پای

1. سنسور ky-037:

این سنسور برای تشخیص صدا به کار میرود و ابعاد آن 32mm\*17mm\*8mm است. همچنین با ولتاژ ورودی 3.3 تا 5 ولت کار میکند.



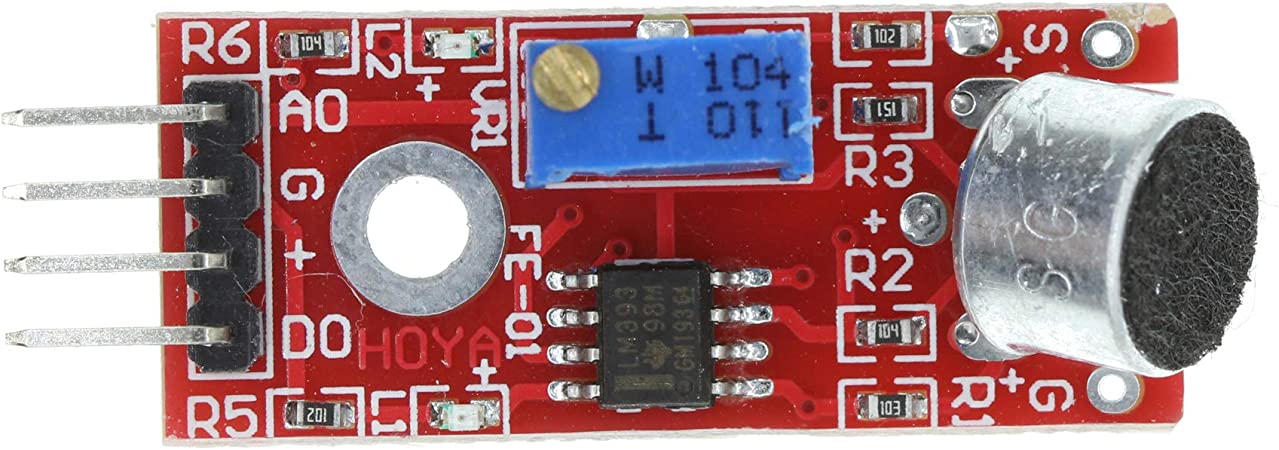
تصویر 4:سنسورky-037

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)



تصویر 5:پین های 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.

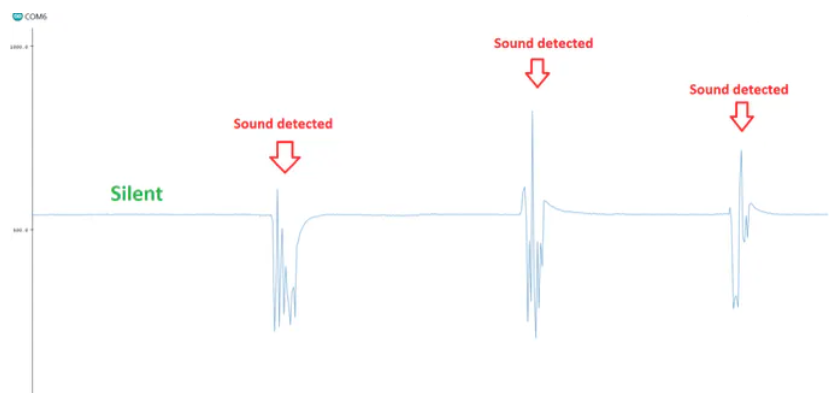
* Comparator

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. 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 2 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.



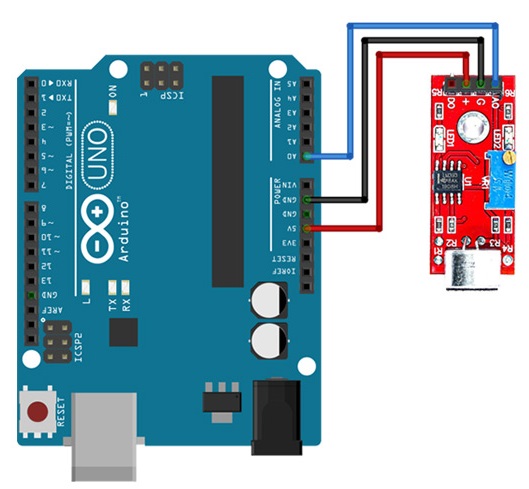
تصویر 6نمونه خروجی آنالوگ سنسور ky-037

بر خلاف آردوئینو، بر روی بورد رزپبری پای نه دریافت کننده ای برای صدای آنالوگ وجود دارد و نه هیچ مبدلی 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.

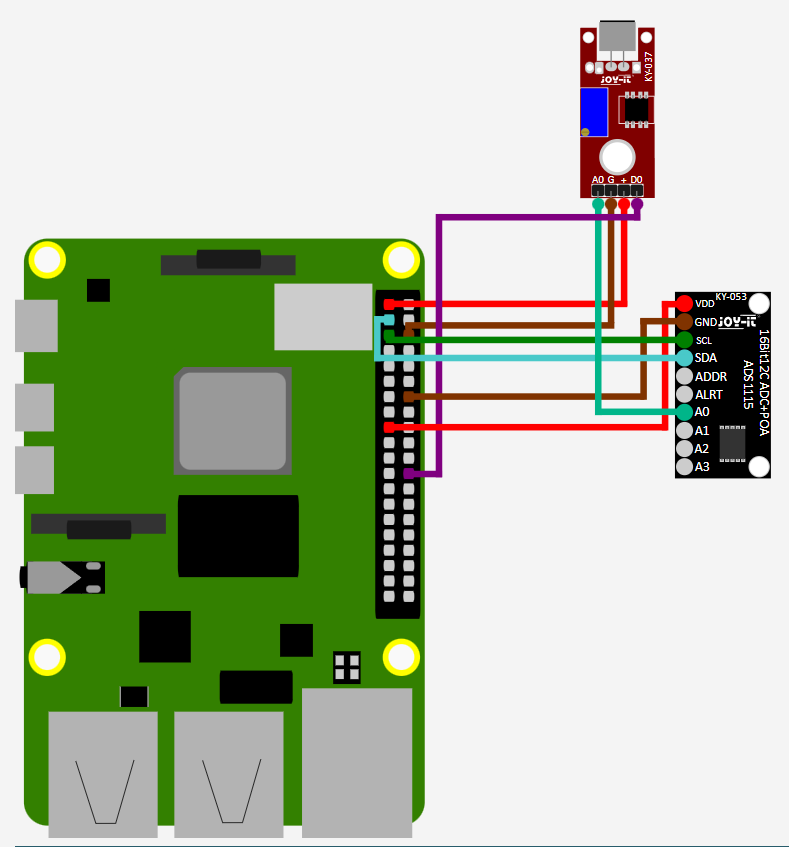
* پین GND سنسور را به GND رزبری پای وصل می‌کنیم.
* پین VCC سنسور را به PIN2 رزبری پای وصل می‌کنیم.
* پین DO سنسور را به PIN 33 - GPIO23 وصل می‌کنیم.

نحوه اتصال به برد آردویینو:



تصویر 7نحوه اتصال سنسور ky-037 به برد آردوئینو

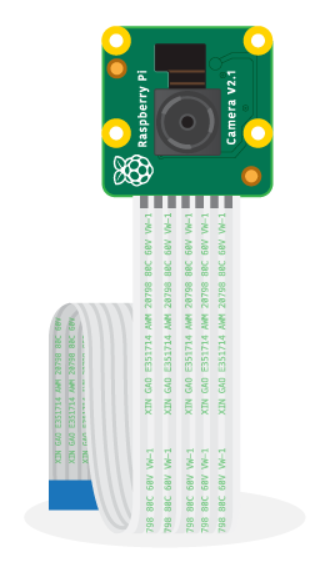
نحوه اتصال به برد رزپبری پای:

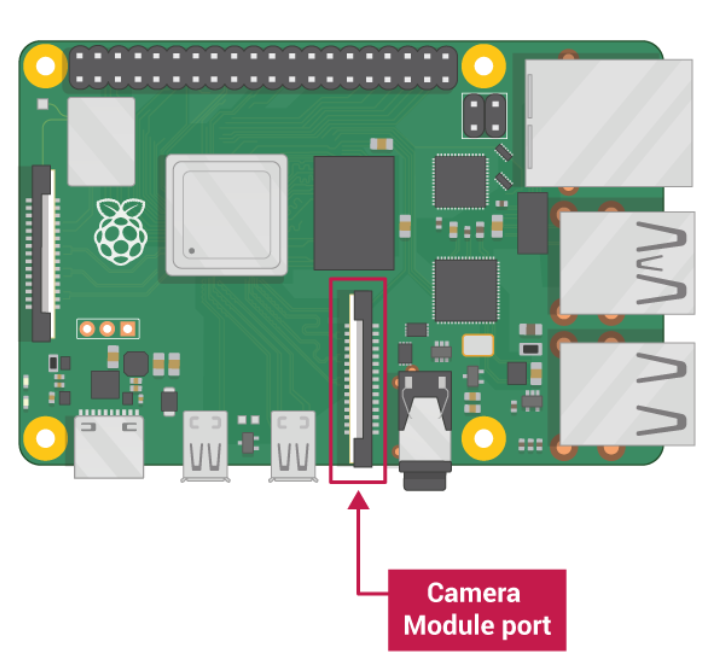


تصویر 8نحوه اتصال سنسور ky-037 به برد رزپبری پای

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 واسط پایتون استفاده میشود. همچنین میتوان آن را روی تلفن همراه پیاده کرد.





تصویر 9ماژول تشخیص تصویر و نحوه اتصال آن به برد رزپبری پای

## 

## 

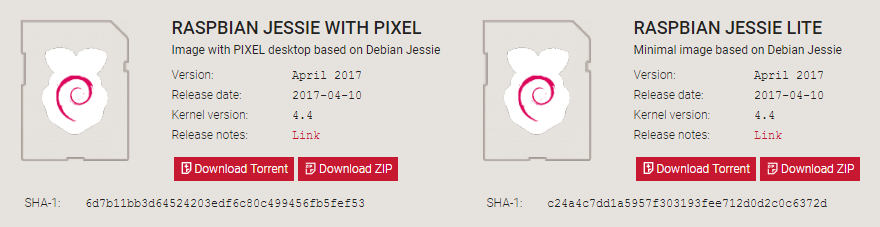
تصویر :لوگوی Raspberry Pi

### **نصب Raspbian**

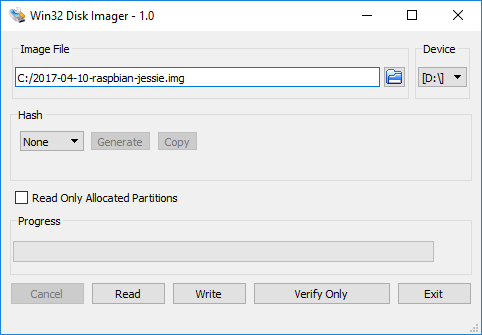
برای شروع به استفاده از برد Raspberry Pi نیاز به نصب سیستم عامل روی آن داریم. در این قسمت توضیح می‌دهیم که چگونه سیستم عامل Raspbian را روی برد Raspberry Pi نصب کنیم.

برای این کار به یک کارت microSD، موس، کیبورد، مانیتور و منبع برق نیاز داریم.

مرحله ۱: دانلود Raspbian

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

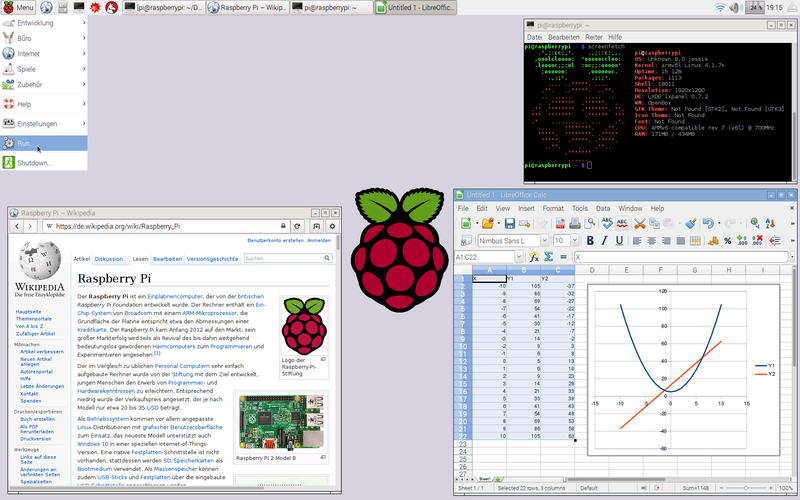
تصویر : لینک‌های دانلود Raspbian

مرحله ۲: قرار دادن فایل Disk Image در microSD

تصویر نمایی 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/)، در مک از برنامه‌ی disk utility که در مک قرار دارد و در ویندوز از [Etcher](https://etcher.io) 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.

تصویر نمایی از Raspbian

## **ماژول ها**

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 پورتها را تنظیم میکنیم. برای اینکار از حالت BCMاستفاده 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)

سپس مشخص میکنیم هر سیگنال و پین آن در بورد 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 پین 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 ترد این بخش از برنامه مجزا از ترد 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(**"HERE"**)  
 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 رزولوشن 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

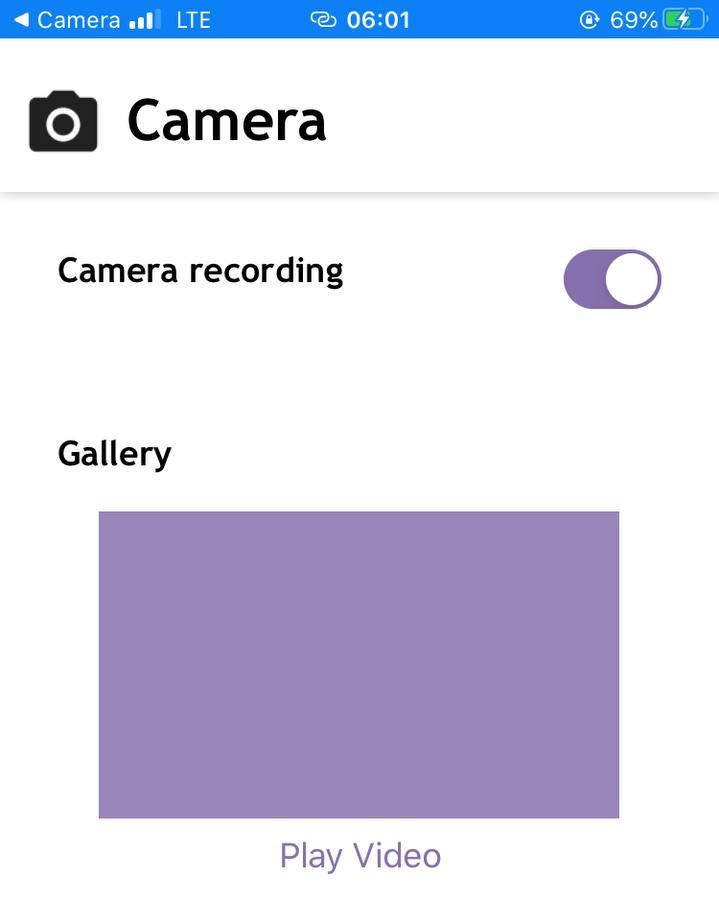
sleep(0.1)

sleep(5)

camera.brightness = 50

**توضیحات کد:**

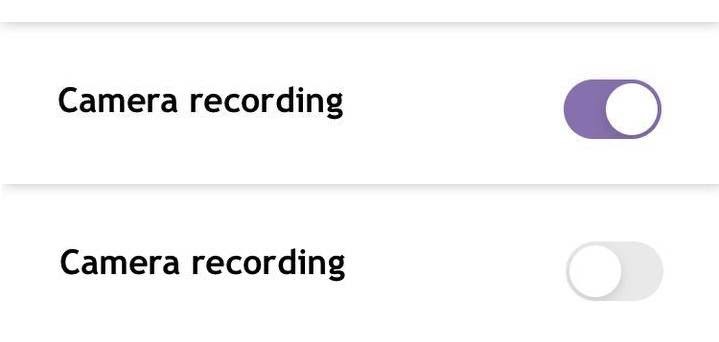
### **اپ موبایل:**



تصویر 14نمایی از اپ موبایل

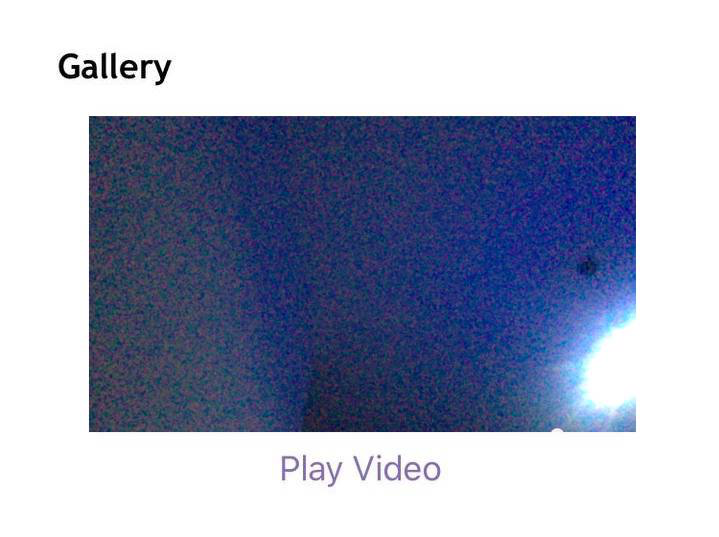
در اپ موبایل عملکرد‌های پیاده‌سازی شده در سرور توسط کاربر قابل انجام می‌شوند.

برای روشن و خاموش کردن دوربین از یک Switch استفاده کرده‌ایم که کاربر با استفاده از آن می‌تواند روی فرآیند ضبط مدیریت داشته باشد.



تصویر 15روشن و خاموش کردن دوربین

پروژه به این صورت است که در بازه‌‌های زمانی مشخصی مثلا هر ۳۰ ثانیه یک ویدیو گرفته و ذخیره می‌کند و در وسط این بازه ( ثانیه ۱۵) یک عکس گرفته می‌شود. این عکس‌ها در قالب یک گالری به کاربر نشان داده می‌شوند که کاربر با عقب و جلو رفتن در این گالری می‌تواند عکس‌های گرفته شده در زمان‌های مختلف را مشاهده کند و اگر کاربر مایل بود ویدیو مربوط به عکس‌ها را مشاهده کند، زیر عکس‌ها یک دکمه‌ی مشاهده ویدیو وجود دارد که این امکان را برای او فراهم می آورد.



تصویر 16گالری و مشاهده ویدیو عکس

# **کد سرور:**

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 لایبرری 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 ترد به طور دائمی فراخوانی شوند. بدین منظور برای هر یک از تردهای تشخیص صدا و ضبط تصویر یک کلاس ساخته شده است. در init هر کلاس تنظیمات اولیه اعم از اینکه پین ورودی رزپبری پای چه باشد، مقدار اولیه متغیرها مثل sound یا send چه باشد و ... تنظیم شده‌اند. سپس مطابق کدهای هفته سوم که در پوشه Week 3 از ریپازیتوری قرار دارند (البته آپدیت شده‌اند ولی باز هم آنجا قرار گرفته‌اند) ماژول‌های تشخیص صدا و تشخیص تصویر را در تردها 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 آیپی داده شده، دستور مدنظر را بزنیم و توکن 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.