

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

پروژه‌ درس آزمایشگاه سخت‌افزار

**مجموعه مستندات آینه هوشمند**

نگارندگان :

آریا جلالی، محمد هجری، امیرحسین باقری

استادان:

جناب آقای دکتر اجلالی – جناب آقای دکتر فصحتی

تابستان 1401



فهرست مطالب

[مقدمه 4](#_Toc112182628)

[دیتاشیت محصول 5](#_Toc112182629)

[معماری سیستم 7](#_Toc112182630)

[طراحی و پیاده سازی سخت افزار 8](#_Toc112182631)

[LCD 8](#_Toc112182632)

[دوربین 9](#_Toc112182633)

[نکاتی درباره دوربین 9](#_Toc112182634)

[پیاده سازی کد سرور در رزبری‌پای 10](#_Toc112182635)

[مشکلات ضبط و استریم همزمان 11](#_Toc112182636)

[راهکار و پیاده سازی نهایی 12](#_Toc112182637)

[طراحی و پیاده سازی اپلیکیشن موبایل 12](#_Toc112182638)

[اجرای سرور در رزبری پای: 15](#_Toc112182639)

[قیمت: 16](#_Toc112182640)

[نماهایی از محصول: 17](#_Toc112182641)

[جمع‌بندی: 22](#_Toc112182642)

فصل 1

# مقدمه

هدف از این پروژه، ارائه محصولی جهت حفظ امنیت خودرو و اشتراک گذاری محتوا در شبکه‌های اجتماعی می‌باشد. پیاده‌سازی این پروژه به صورت کلی و توانایی استفاده از آن در انواع مختلفی از ماشین‌ها وجود دارد.

محصول توانایی پخش تصویر پشت ماشین به منظور کمک به راننده در حین رانندگی واشتراک گذاری ویدئوهای گرفته شده از پشت ماشین در هنگام سفر یا مقاصد دیگری که نیاز به اشتراک گذاری ویدئو در شبکه‌های اجتماعی دارند را دارد، همیچنین در پیشگیری از مخاطرات ناشی از عدم دید کافی راننده به پشت خود ‌می‌تواند موثر باشد.

ویدئو‌های ضبط شده توسط این محصول در یک سرور جانبی ذخیره می‌گردند، و کاربر می‌تواند به مقاصد مختلفی از آن‌ها استفاده کند.

فصل 2

# دیتاشیت محصول

|  |  |
| --- | --- |
| کیفیت دوربین | 5 مگاپیکسل |
| فریم بر ثانیه | 30 فریم در کیفیت Full HD ((1080p  60 فریم در کیفیت HD (720p)  90 فریم در کیفیت 480p |
| بهترین کیفیت عکس | 1944 × 2592 پیکسل |
| جرم دوربین به گرم | 9.97 |
| ابعاد دوربین به اینچ | 0.79×1.18×1.57 |
| فرکانس پردازنده به مگاهرتز | 300 تا 400 |
| ابعاد پردازنده به میلیمتر | 17 ×56 ×85 |
| جریان ورودی | 1.5 آمپر |
| ولتاژ ورودی | 5 ولت |
| جرم پردازنده به گرم | 50 |
| تعداد پورت USB | 4 |
| تعداد Ethernet port | 1 |
| WIFI | ✅ |
| Bluetooth | ✅ |
| کابل HDMI | ✅ |
| سیستم عامل | Raspbian OS |
| LCD | ❌ |

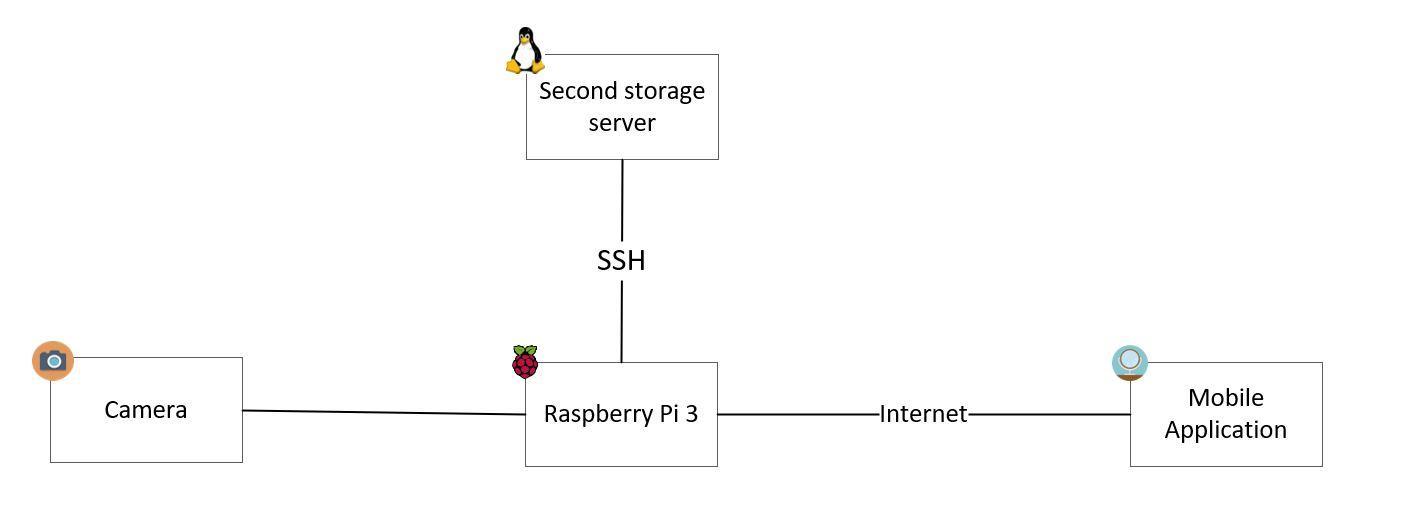
* وزن محصول بسته به شرایط و موارد مورد استفاده در تولید جعبه‌ی آن متغیر است.

فصل 3

# معماری سیستم

سیستم طراحی شده توسط ما از دو قسمت سخت افزاری و نرم افزاری تشکیل شده است. قسمت سخت افزاری شامل یک رزپری‌پای و دوربین جهت دریافت داده ها و قسمت نرم افزاری ما شامل اپلیکیشن موبایل جهت استفاده از دوربین و یک استریم سرور در رزپری‌پای است.

معماری سیستم در شکل زیر قابل مشاهده است.



## طراحی و پیاده سازی سخت افزار

پیاده سازی قسمت سخت افزاری با استفاده از قطعات زیر

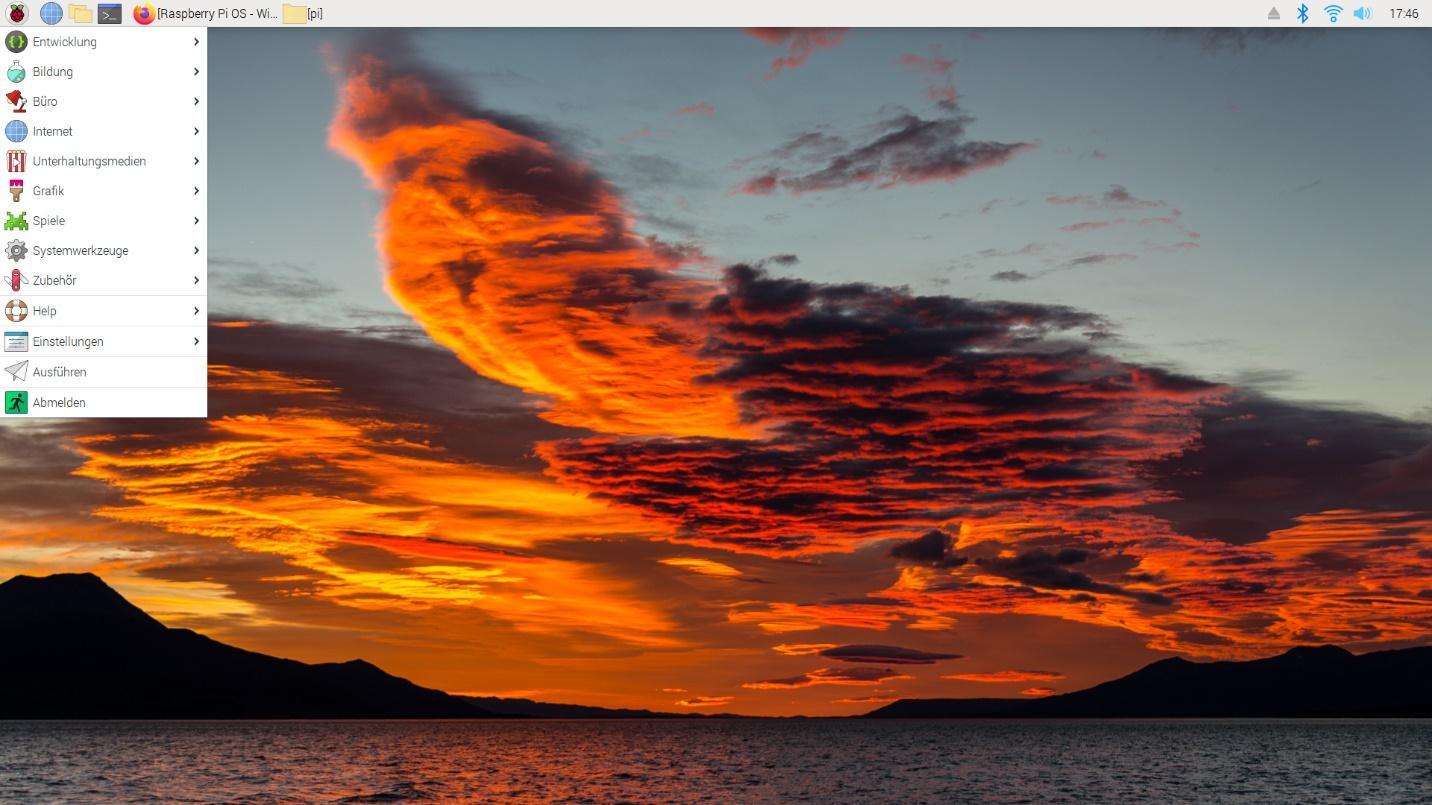
* برد رزپری‌پای
* دوربین 5 مگاپیکسلی رزپری‌پای
* LCD

### LCD

برای استفاده از LCD مشکلات زیر پیش آمد و بنابراین از استفاده از آن صرف‌نظر کردیم.

* ماژول LCD نیاز به نصب درایور دارد که توضیحات آن در این [لینک](http://www.lcdwiki.com/3.5inch_RPi_Display) آمده است.

با نصب این درایور و اتصال LCD به رزپری‌پای باید صفحه سیستم عامل رزپری‌پای را در LCD-3.5 inch مشاهده کنید.

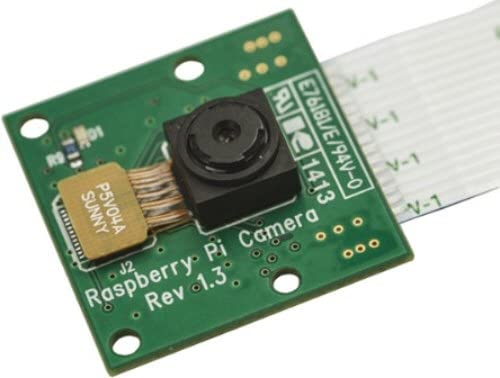


اما اگر LCD مورد استفاده چینی باشد ممکن است با مشکلی مواجه شوید که صفحه آن سفید می‌ماند و یا پس از صبر طولانی صفحه می‌آید اما کار تصویر نویز دارد و حتی کارکرد آن نیز رندم است و ممکن است دفعه بعدی حتی با صبر بسیار نیز صفحه بالا نیاید.

پیشنهاد ما به شما این است که از خیر آن بگذرید و زمان بر روی آن نگذارید. تقریبا ۳ روز تلاش بری درست کردن مشکل بی ثمر بوده و در نهایت ما نیز به پاسخ هایی برخوردیم که اگر مدل شما چینی است و بار اول درست کار نکرد دیگر درست نخواهد شد.

### دوربین

ماژول دوربین را می‌توانید مطابق [لینک](https://projects.raspberrypi.org/en/projects/getting-started-with-picamera) متصل کنید.



### نکاتی درباره دوربین

* دوربین را می‌توانید با دستور **raspivid** کنترل کنید. استفاده از لایبری **cameraPi** برای کنترل دوربین در کد پایتون نیز مهیا شده است.

# پیاده سازی کد سرور در رزبری‌پای

برای پیاده سازی کد سرور راهکار های متفاوتی وجود دارد که به راهکار های تست شده می‌پردازیم و مزایا و معایب آن‌ها را بررسی می‌کنیم. راهکار های پیش رو شامل دو دسته استفاده از برنامه های موجود در PATH سیستم عامل و استفاده از کد پایتون خواهند بود و بررسی هر کدام به شرح زیر است.

* استفاده از **netcat** , **raspivid**

|  |
| --- |
| raspivid -t {time} -w {w} -h {h} -hf -fps 20 -o - | nc {ip} {port} |

برای اجرای دستور بالا ابتدا در کامپیوتر مقصد دستور زیر را اجرا کنید.

|  |
| --- |
| nc -l {port} | mplayer -fps 200 -demuxer h264es - |

سپس با اجرای دستور اول می‌توان استریم را در کامپیوتر اول مشاهده کرد.

از مزایای این روش می‌توان به ساده بودن و دیلی فوق العاده کم در استریم آن نام برد.

* استفاده از **libcamera** , **vlc**

برای استفاده از این روش تنها نیاز است که vlc بر روی کامپیوتر مقصد نصب شده باشد.

با استفاده از دو دستور زیر می‌توانید استریم را مشاهده کنید.

|  |
| --- |
| libcamera-vid -t 0 --inline -o - | cvlc stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554/stream1}' :demux=h264 |

و در کامپیوتر مقصد دستور

|  |
| --- |
| vlc rtsp://<ip-addr-of-server>:8554/stream1 |

می‌توانید استریم را مشاهده کنید.

دقت کنید که روش بالا برای پروتکل های مختلفی پیاده سازی دارد.

(**http tcp, udp, rtsp,**)

از مشکلات این روش به دیلی بالا آن می‌توان اشاره کرد اما توانایی رکورد آن توسط **vlc** وجود دارد.

### مشکلات ضبط و استریم همزمان

استفاده از روش های بالا در عین سادگی امکان ضبط همزمان و استریم کردن را نمی‌دهد مگر اینکه با استفاده از ۲ پردازه ضبط و استریم را انجام دهیم برای ضبط کردن می‌توان از کد زیر در پایتون استفاده کرد و به عنوان یک پردازه جدا آن را کرد. اما مشکلات آن را در زیر برمی‌شماریم.

|  |
| --- |
| def save\_recording(data,name,conn):  print("create Picamera")  camera = PiCamera()  fileName = name  camera.start\_recording(fileName)  camera.wait\_recording(int(data["time"]))  camera.stop\_recording()  camera = None  n = gc.collect()   print("video-recorded")  try:  f = open(fileName, "rb")  except:  print("Error: file not found")  return  print("start sending vide")  conn.send(fileName.encode())  print(conn.recv(1024))  video = f.read(1024)  while (video):  conn.send(str(len(video)).encode())  conn.recv(1024)  conn.send(video)  video = f.read(1024)  conn.recv(1024)  conn.send("0".encode())  print(conn.recv(1024))  print("video sent") |

* چون تنها یک پردازه توانایی برداشت اطلاعات از بافر دوربین را دارد این کار سبب می‌شود که هم استریم هم ضبط دچار وقفه های طولانی شود و عملا فانکشنالیتی برنامه از بین می‌رود.
* ماژول رزبری‌پای بر اثر افزایش دما ریبوت می‌شود و ران کردن دو پردازه که دیتای سنگین فیلم و ویدئو را استریم و ذخیره می‌کند موجب داغ شدن و ریبوت شدن سیستم می‌شود‌ (‌ این فرایند تس-ت شده است)

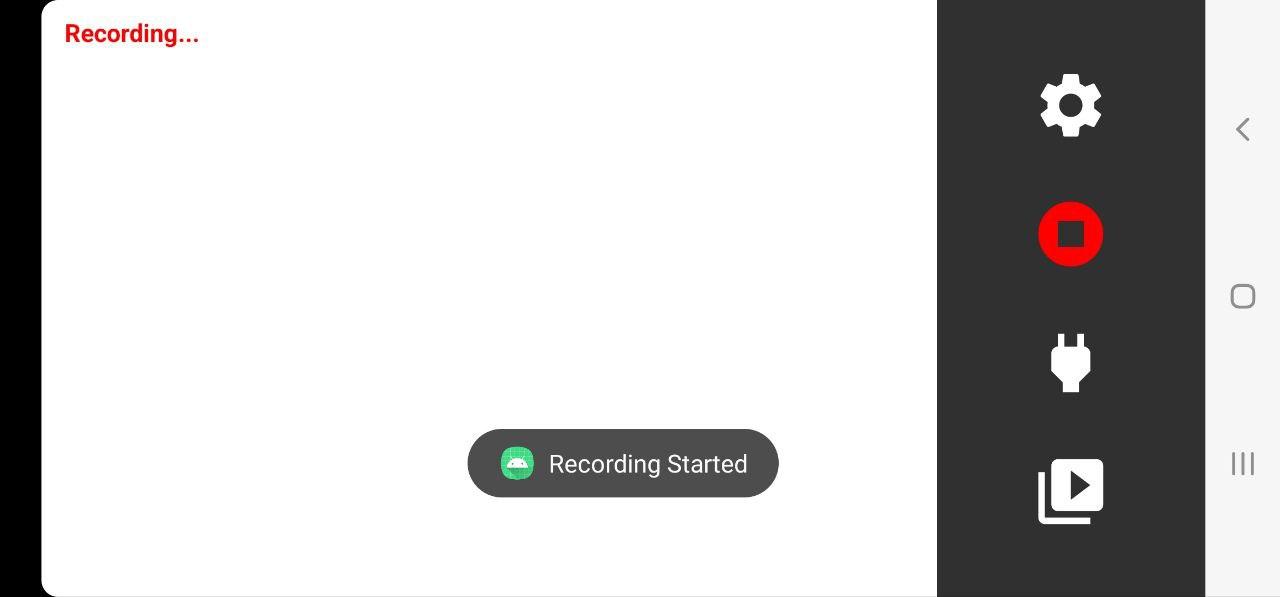
### راهکار و پیاده سازی نهایی

به همین منظور که مشکلات ذکر شده پیش نیاید تصمیم بر آن شد که ریکورد استریم توسط اپلیکیشن موبایل انجام شود. بنابراین رزبری‌پای یک استریم سرور است که بر روی پورت ۸۰۰۰ آن بالا می‌آید.

پیاده سازی سرور با استفاده از **socketserver** , **webserver** , **streamserver** پایتون پیاده شده اند.

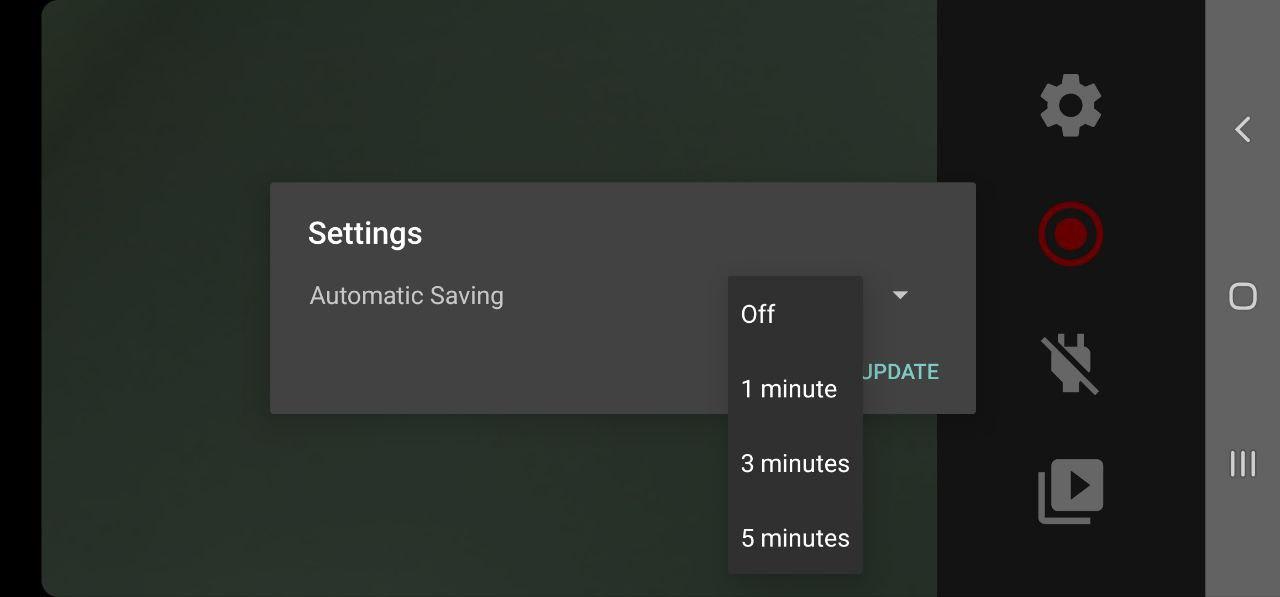
## طراحی و پیاده سازی اپلیکیشن موبایل

نمای کلی برنامه به صورت مقابل است:



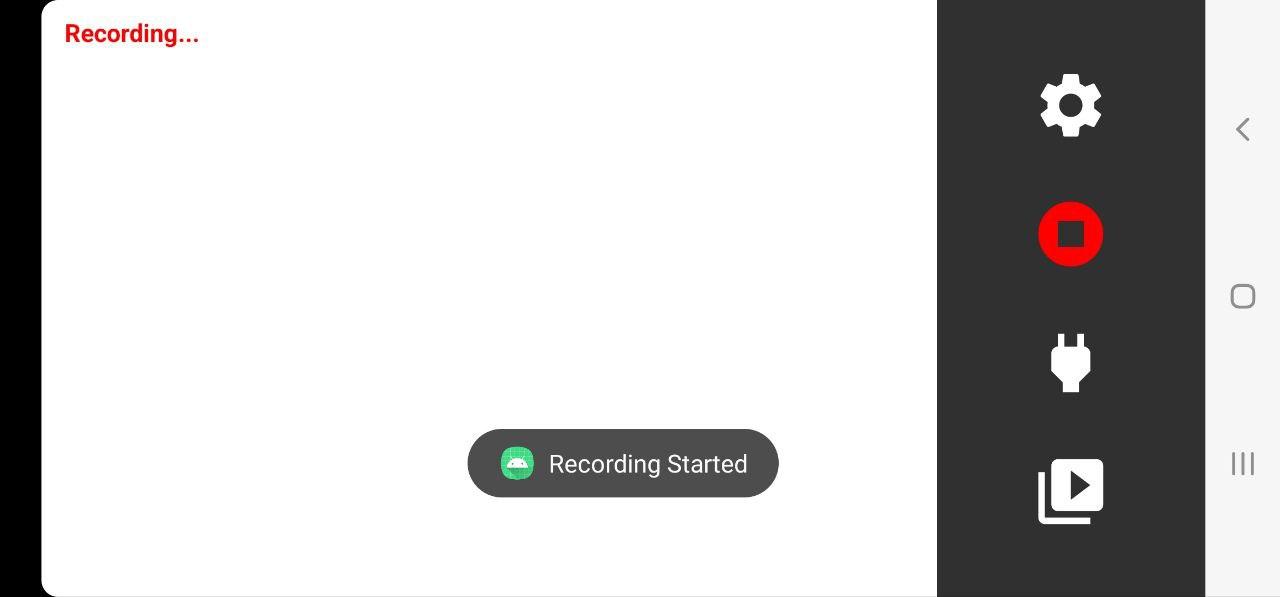
برنامه دارای 4 دکمه‌ی اصلی است که در سمت راست تصویر بالا مشخص است. در ادامه به شرح وظایف هرکدام از این دکمه‌ها می‌پردازیم:

* دکمه‌ی تنظیمات :



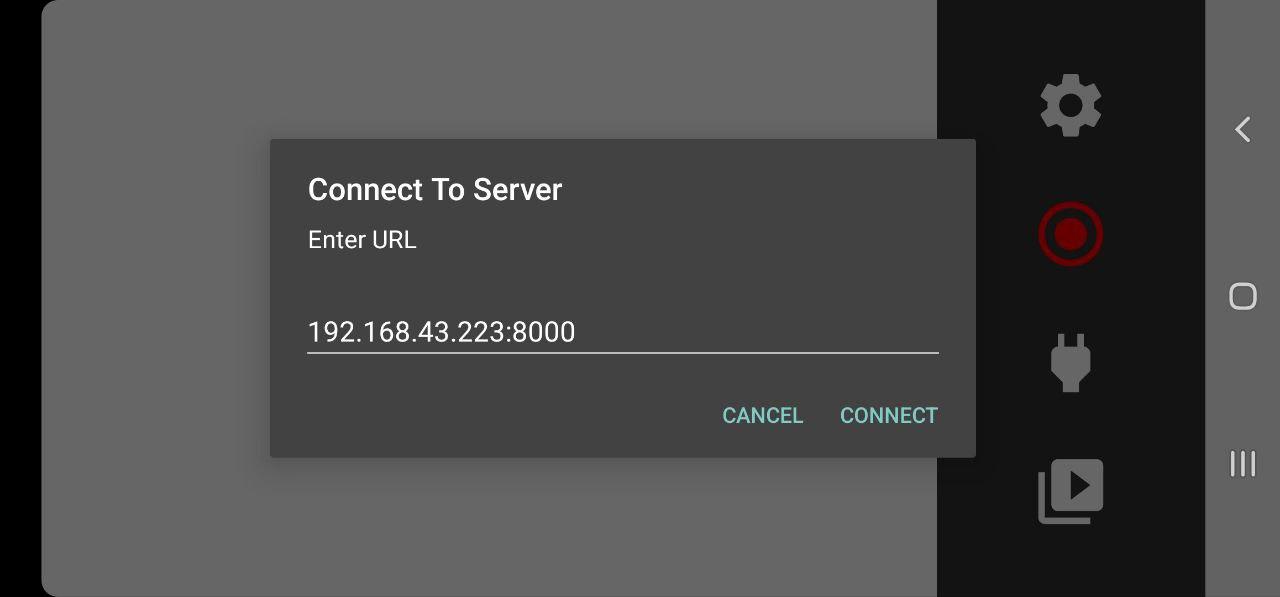
کاربر پس از فشار دادن این دکمه می‌تواند زمانی که می‌خواهیم رکورد کنیم را از بین گزینه‌های 1، 3 و 5 دقیقه انتخاب کند، و اگر کاربر بخواهد می‌تواند با انتخاب گزینه‌ی Off به مقدار دلخواه صفحه را رکورد کند.

* دکمه‌ی رکورد:



پس از فشار دادن این دکمه یک Toast مبنی بر شروع شدن ضبط نشان داده می‌شود و تکست "...Recording" در بالای صفحه به نمایش در می‌آید. اگر کاربر زمان خاصی را انتخاب کرده باشد، رکوردینگ پس از زمان مشخص شده پایان می‌یابد و فایل در حافظه‌ی گوشی ذخیره میشود. دقت کنید در هر دو حالت (انتخاب زمان پیشفرض و عدم انتخاب زمان خاص) کاربر می‌تواند در صورت نیاز دوباره دکمه‌ی رکورد را بزند و رکوردینگ را به پایان برساند.

* دکمه‌ی Connect:

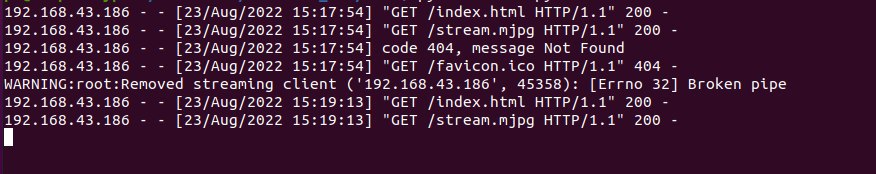


هنگام فشرده شدن دکمه‌ی Connect، یک Popup برای وارد کردن آدرس سرور به کاربر نمایش داده می‌شود. این IP در هر وصل شدن عوض می‌شود، دقت کنید در صورت عرضه‌ی این پروژه این حقیقت مشکل چندانی برای ما ایجاد نمی‌کند، زیرا در صورت عرضه شدن این پروژه سرور به ندرت خاموش می‌شود و IP برای مدت‌های بسیار طولانی ثابت می‌ماند، و یا حتی می‌توانیم یک IP ثابت برای هر کاربر با دریافت مبلغی اضافه تهیه کنیم.

* دکمه‌ی Files:

هنگام فشرده شدن دکمه‌ی Files، فولدری که فیلم‌های ضبط شده در آن‌جا ذخیره شده‌اند برای کاربر باز می‌شود تا بتواند فایل مورد نظر خود را تماشا کند.

# اجرای سرور در رزبری پای:



سرور اجرا شده در رزبری پای اطلاعات گرفته شده از دوربین رزبری پای را در بافر کرنل قرار می‌دهد و در ادامه یک کلاس Streamer این اطلاعات را از بافر می‌خواند و در قالب یک فایل HTML که دارای یک img tag (که source آن بافر ذکر شده است) قرار می‌دهد.

# قیمت:

|  |  |  |
| --- | --- | --- |
| ردیف | قطعه | قیمت |
| 1 | رزبری پای 3 | 4,150,000 تومان |
| 2 | ماژول دوربین 5 مگاپیکسلی | 159،500 تومان |
| 3 | جمع | 4,309,500 تومان |

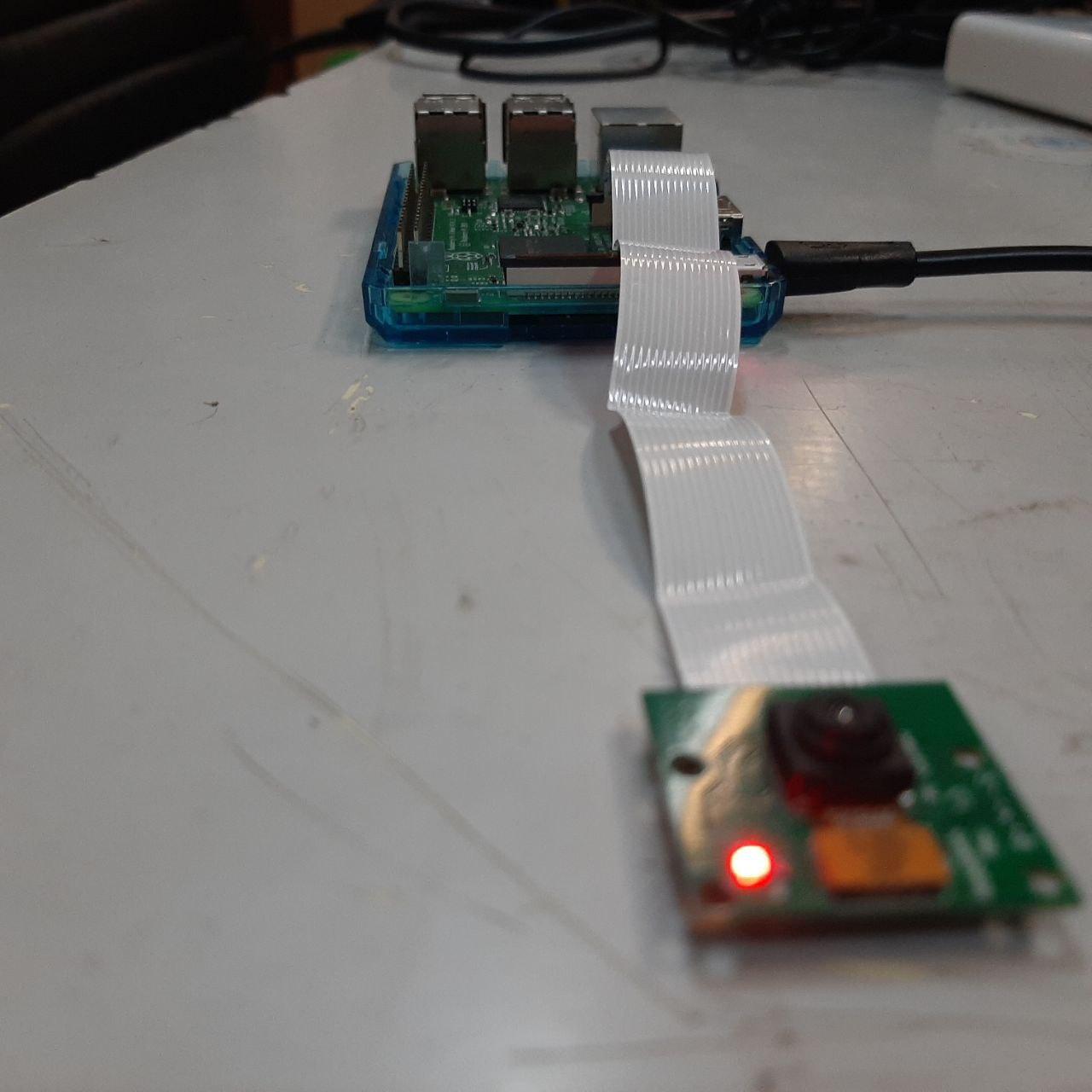
# 

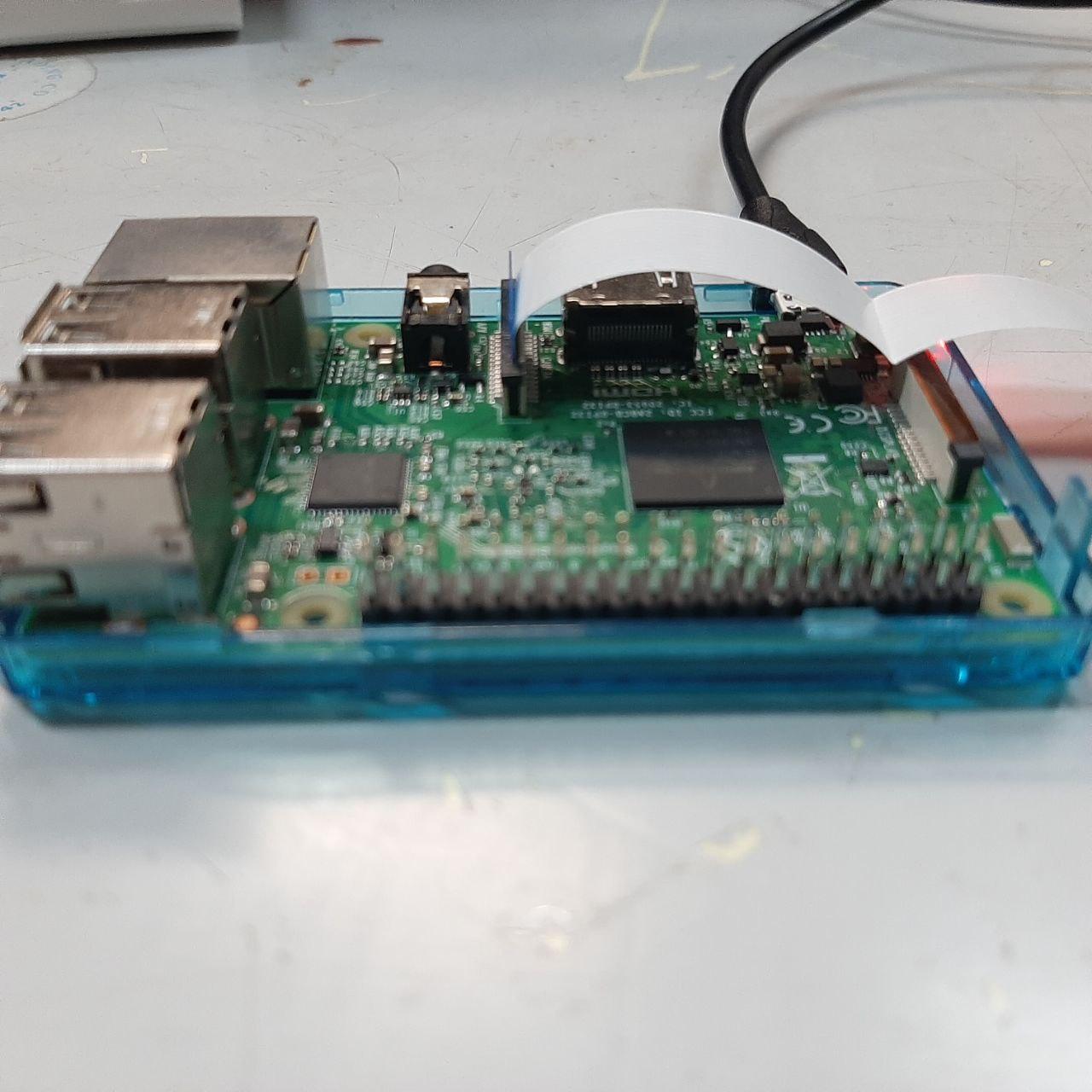
# 

# 

# نماهایی از محصول:







فصل 4:

# جمع‌بندی:

در این پروژه به پیاده‌سازی یک سیستم هوشمند آینه‌ی پشت ماشین پرداختیم. این سیستم با استفاده از یک دوربین و یک سرور در رزبری پای و یک اپلیکیشن موبایل می‌تواند به راننده در حین رانندگی کمک کند تا در هنگام وقایع غیر مترقبه مانند مسدود شدن دید پشت توسط سرنشینان پشت یا صندوق عقب ماشین به پشت خود دید داشته باشد.

در کنار این محصول، یک اپلیکیشن موبایل فراهم گشته تا با استفاده از آن بتوانید از استریم فیلم تهیه کرده و آن را در شبکه‌های اجتماعی برای مخاطبان خود به اشتراک بذارید.

می‌توان با استفاده از ابزار‌های یادگیری ماشین بر روی رزبری پای یک مدل تشخیص اشیاء تا بتواند به راننده هشدا‌ر‌های لازم جهت جلوگیری از وقوع حادثه را دهد.