دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

پروژه درس آزمایشگاه سخت‌افزار

مجموعه مستندات آینه هوشمند

نگارندگان

آریا جلالی، محمد هجری، امیرحسین باقری

استاد گرامی

جناب آقای دکتر اجلالی – جناب آقای دکتر فصحتی

تابستان ۱۴۰۱


بسم الله الرحمن الرحيم

فهرست مطالب

فصل ۱

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 و اشتراک گذاری ویدیوهای گرفته شده از پشت ماشین در هنگام سفر یا مقاصد دیگری که نیاز به اشتراک گذاری ویدئو در شبکه‌های اجتماعی دارند را دارد، همچنین 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 ۲

دیتاشیت محصول

کیفیت دوربین	۵ مگاپیکسل
فریم بر ثانیه	۳۰ فریم در کیفیت Full HD ۶۰ فریم در کیفیت HD ۹۰ فریم در کیفیت SD
بهترین کیفیت عکس	۱۹۴۴ × ۲۵۹۲ پیکسل
جرم دوربین به گرم	۹.۹۷
ابعاد دوربین به اینچ	۰.۷۹×۱.۱۸×۱.۵۷
فرکانس پردازنده به مگاهرتز	۳۰۰ تا ۴۰۰
ابعاد پردازنده به میلیمتر	۸۵× ۵۶× ۱۷
جریان ورودی	۱/۵ آمپر
ولتاژ ورودی	۵ ولت
جرم پردازنده به گرم	۵۰
تعداد پورت USB	۴

تعداد Ethernet port	۱
WIFI	☑
Bluetooth	☑
کابل HDMI	☑
سیستم عامل	Raspbian OS
LCD	✗

جدول ۱ - دیتاشیت سیستم

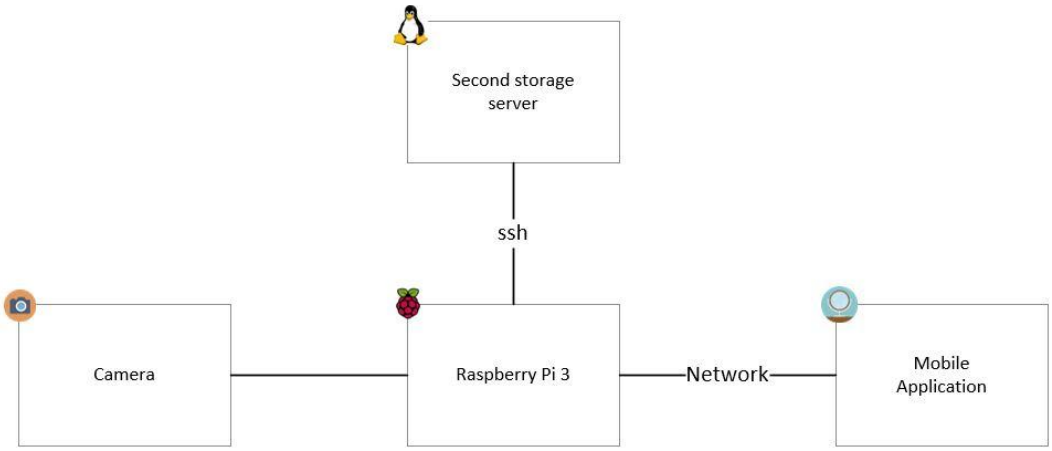
* وزن محصول بسته به شرایط و موارد مورد استفاده در تولید جعبه‌ی آن متغیر است.

فصل ۳

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.



شکل ۱ - معماری کلی سیستم طراحی شده

طراحی و پیاده سازی سخت افزار

پیاده سازی قسمت سخت افزاری با استفاده از قطعات زیر انجام شده است.

- برد رزپری‌پای
- دوربین ۵ 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 آمده است.

با نصب این درایور و اتصال LCD به رزپری‌پای باید صفحه سیستم عامل رزپری‌پای را در LCD-۳.۵ inch مشاهده کنید.

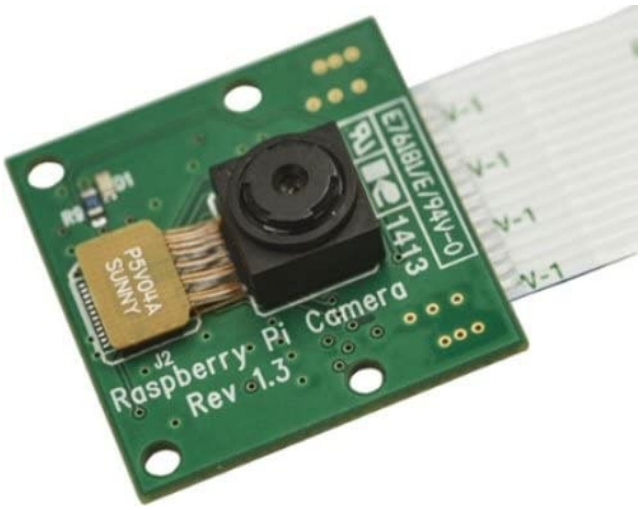
شکل ۲ صفحه‌ی سیستم عامل رزبری پای

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. تقریباً سه روز تلاش برای درست کردن این مشکل بی ثمر بوده و در نهایت ما نیز به پاسخ‌هایی برخوردیم که اگر مدل شما چینی است و بار اول درست کار نکرد دیگر درست نخواهد شد.

دوربین

ماژول دوربین را می‌توانید مطابق لینک متصل کنید.



شکل ۳ - ماژول دوربین ۵ مگاپیکسلی رزبری پای

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 **raspivid** , **netcat**

```
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 **vlc** , **libcamera**

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.

به دلیل محدود بودن قدرت پردازشی رزبری پای و کیفیت کم ماژول دوربین، لازم بود برای فرستادن فیلم به صورت زنده و استریم کردن آن پروتکل مناسبی را انتخاب کنیم که هم بتواند استریمی با دیلی و تاخیر قابل قبول بفرستد و هم فشار زیادی به پردازنده وارد نکند.

پروتکل‌هایی همانند TCP، UDP، HTTP و RTST تست شدند و در نهایت به نتایج مقابل رسیدیم:

TCP

این پروتکل به دلیل بررسی هر بایت خروجی یا ورودی Output بسیار بدی داشت و باعث دیلی بسیار شدید (در بعضی از شرایط تا حد ده ثانیه و بیشتر) می‌شد، ولی خوبی این پروتکل راحتی کار با آن بود و می‌توانستیم به راحتی استریم فرستاده شده از این پروتکل را در اپلیکیشن موبایل دریافت و ضبط کنیم.

UDP

این پروتکل بسیار سریعتر از همتا خود فیلم را استریم می‌کرد و تنها دیلی بسیار جزئی‌ای در آن دیده می‌شد. دلیل عدم استفاده از این پروتکل فشار زیادی بود که بر پردازنده وارد می‌شد و از طرفی دریافت استریم توسط اپلیکیشن با این پروتکل بسیار پیچیده‌تر از پروتکل‌های دیگر بود. این دلایل باعث شد که از استفاده از این پروتکل صرف نظر کنیم.

RTST

این پروتکل نیز همانند UDP دارای سرعت موردنیاز بود، ولی دریافت آن از طریق اپلیکیشن دشوار بود و از آن استفاده نشد.

HTTP

این پروتکل که پروتکل استفاده شده در محصول نهایی است، بهترین قابلیت پروتکل‌های قبلی بدون مشکلات آن‌ها را درون خود داراست. با استفاده از این پروتکل توانستیم استریمی با دیلی بسیار کمتر از TCP و اندکی بیشتر از UDP که غیر قابل تشخیص است با بار سبکی بر پردازنده نمایش دهیم. نحوه‌ی عملکرد این پروتکل به این صورت است که استریم به یک وب سرور فرستاده می‌شود و در ادامه این استریم با استفاده از یک WebView دریافت می‌شود.

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 تست 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 **stream server**، **socket server** و **web server** پایتون پیاده شده اند.

دلایل عدم استفاده از پروتکل‌های بالا به صورت کلی

به دلیل درخواست استاد گرامی مبنا بر نوشتن اپلیکیشن موبایل بر بستر پلتفرم اندروید، نمی‌توانستیم از بسیاری از پروتکل‌های ذکر شده برای نمایش استریم دوربین استفاده کنیم. همانطور که بالاتر نیز به آن اشاره شده است netcat بهترین ابزار برای تماشای استریم را برای ما فراهم می‌کرد، ولی استفاده از آن روی سیستم‌عامل اندروید غیرممکن بود و مجبور به انتخاب پروتکل‌های کندتر و ساده‌تر شدیم تا بتوانیم نمایش استریم بر روی گوشی را ممکن سازیم.

می‌توان به جای استفاده از موبایل، از یک کامپیوتر شخصی برای نمایش و ضبط استریم استفاده کرد. این کار یک استریم بدون لگ را برای ما مهیا می‌کند ولی کاربردی نخواهد بود، زیرا استفاده از کامپیوتر شخصی برای راننده‌ی یک ماشین امری بسیار دشوار یا غیر ممکن خواهد بود.

مقایسه راهکار های گفته شده

روش	میانگین فریم بر ثانیه استریم (تاخیر)	توانایی ضبط و پخش
raspivid + netcat	۶۰ fps	stream only
raspivid + Network protocol (http,udp,tcp,rtsp)	۱۲ fps	stream only
libcamera + VLC (http,udp,tcp,rtsp)	۲۵ fps	stream only recording with vlc player
raw server on raspberry using raspivid	۶۰ fps	stream and record (but needs second server)
web server on raspberry using PiCamera	۶۰ fps	stream and record

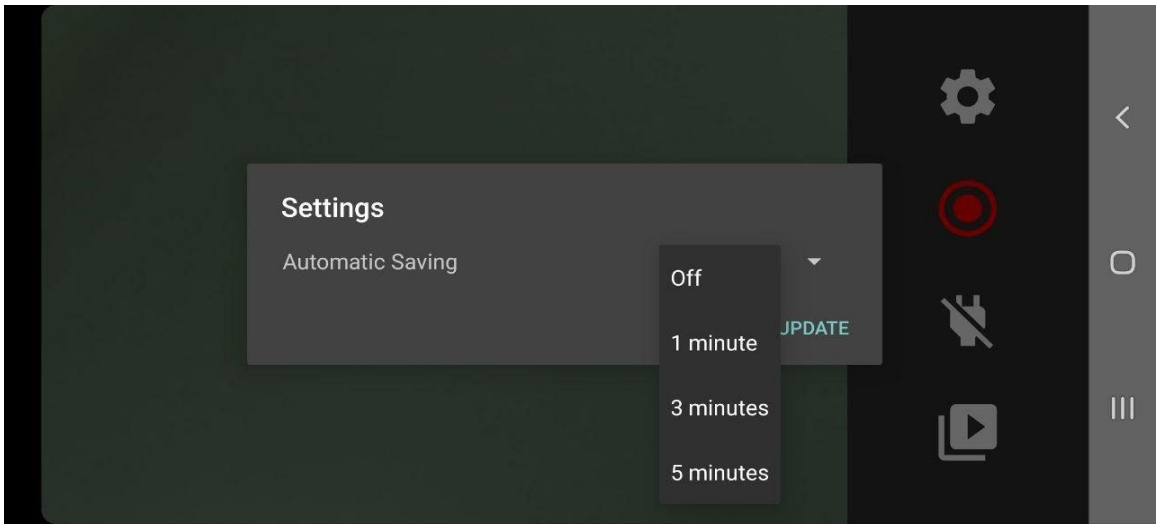
تاخیر گفته شده در جدول بالا به این صورت است که در حالت عادی استریم با ۳۰ فریم بر ثانیه باید انجام شود. هنگامی که لگ در استریم پیش می‌آید به این معناست که در فریم بر ثانیه پخش تصاویر تاخیر آمده است و برای محاسبه تاخیر ما میانگین fps سمت کلاینت را در طول یک دقیقه در نظر گرفته ایم. و این بدان معنا نیست که تصویر به عنوان مثال ۱۲ فریم بر ثانیه پخش میشود بلکه بدان معناست که تعداد فریم های دریافتی در یک دقیقه که باید معادل ۳۶۰۰ فریم باشد به عدد میانگین رسیده است.

نمای کلی برنامه به صورت مقابل است:

شکل ۴ - نمای کلی اپلیکیشن

برنامه دارای ۴ دکمه‌ی اصلی است که در سمت راست تصویر بالا مشخص است. در ادامه به شرح وظایف هرکدام از این دکمه‌ها می‌پردازیم:

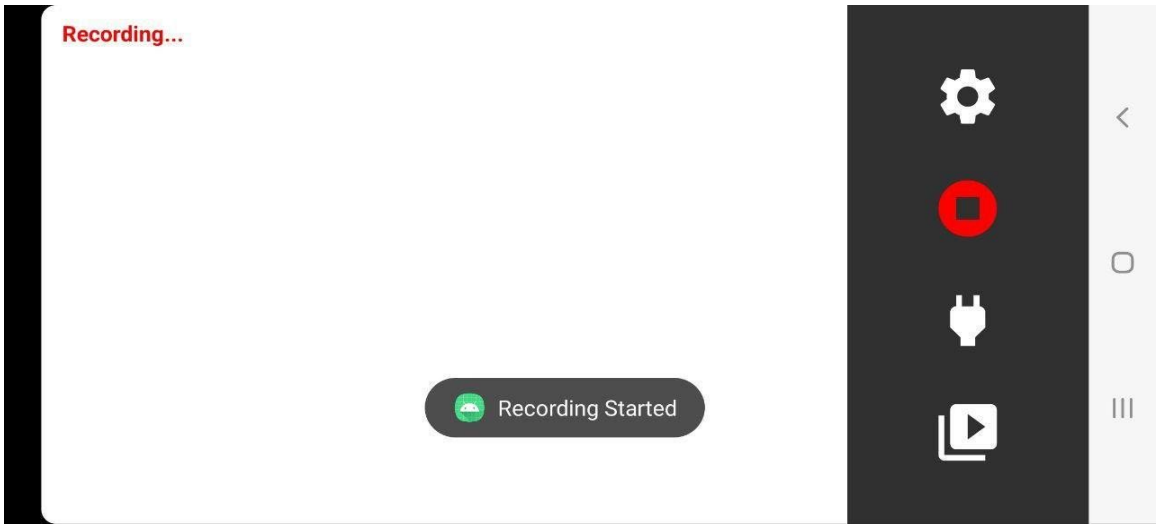
- دکمه‌ی تنظیمات



شکل ۵ - انتخاب زمان ضبط از منوی تنظیمات

کاربر پس از فشار دادن این دکمه می‌تواند زمانی که می‌خواهیم رکورد کنیم را از بین گزینه‌های ۱، ۳ و ۵ دقیقه انتخاب کند، و اگر کاربر بخواهد می‌تواند با انتخاب گزینه‌ی Off به مقدار دلخواه صفحه را رکورد کند.

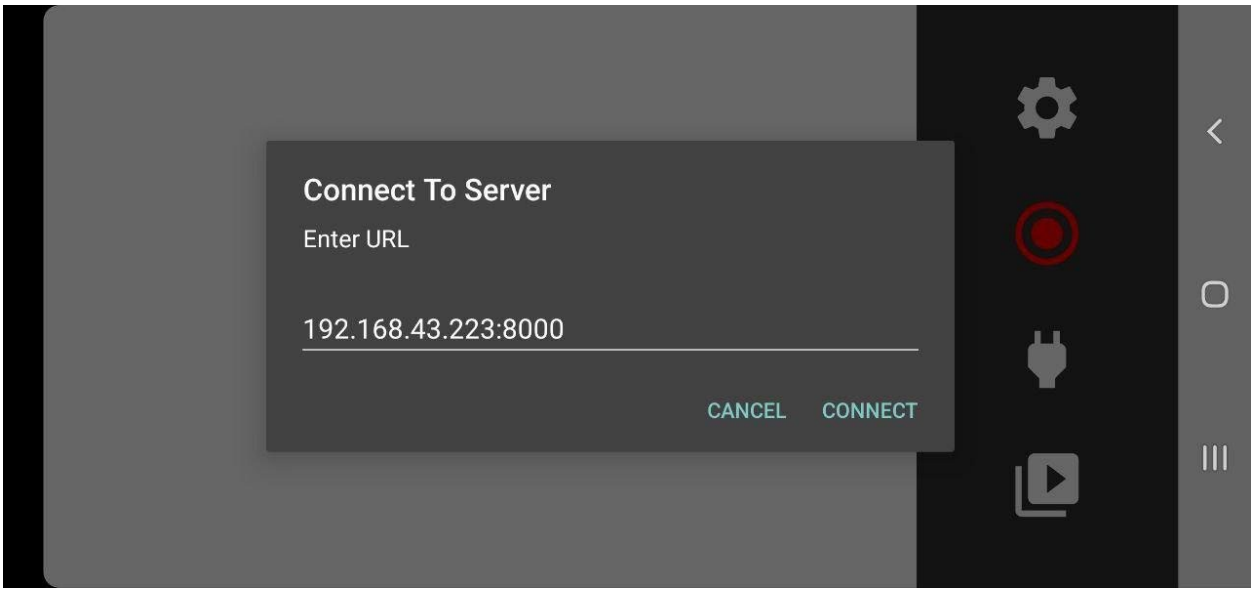
- دکمه‌ی رکورد



شکل ۶ - شروع ضبط صفحه در اپلیکیشن

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



شکل ۷ - وارد کردن **IP** سرور رزبری در بخش اتصال

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.

امکانات اپلیکیشن نوشته شده

ذخیره‌ی اتوماتیک

اگر در هنگام ذخیره‌ی استریم، کاربر به هر دلیلی (زنگ زدن گوشی، قفل شدن اشتباهی) مجبور به خروج از برنامه شود، ضبط متوقف می‌شود و پس از ورود کاربر ادامه می‌یابد.

ذخیره‌ی استریم به مدت زمان‌های پیشفرض

همانطور که در بخش توضیحات بخش‌های مختلف اپلیکیشن به آن پرداختیم، کاربر می‌تواند زمان‌های ضبط را برابر با ۱، ۳ و ۵ دقیقه قرار دهد و یا می‌تواند کلا این گزینه را نادیده بگیرد و ضبط را به صورت دستی قطع کرده و نتیجه‌ی آن را ذخیره کند.

این قابلیت در بسیاری از موارد می‌تواند کمک حال راننده باشد; فرض کنید راننده در حال رانندگی قصد ضبط کردن را دارد و نمی‌تواند به مدت زیادی با گوشی تعامل داشته باشد. او می‌تواند با انتخاب یکی از زمان‌های پیشفرض دکمه‌ی ضبط را بزند و با خیال راحت به رانندگی خود ادامه دهد و مطمئن باشد که ویدئو ضبط شده پس از گذر مدت مشخص شده در گوشی او ذخیره می‌شود.

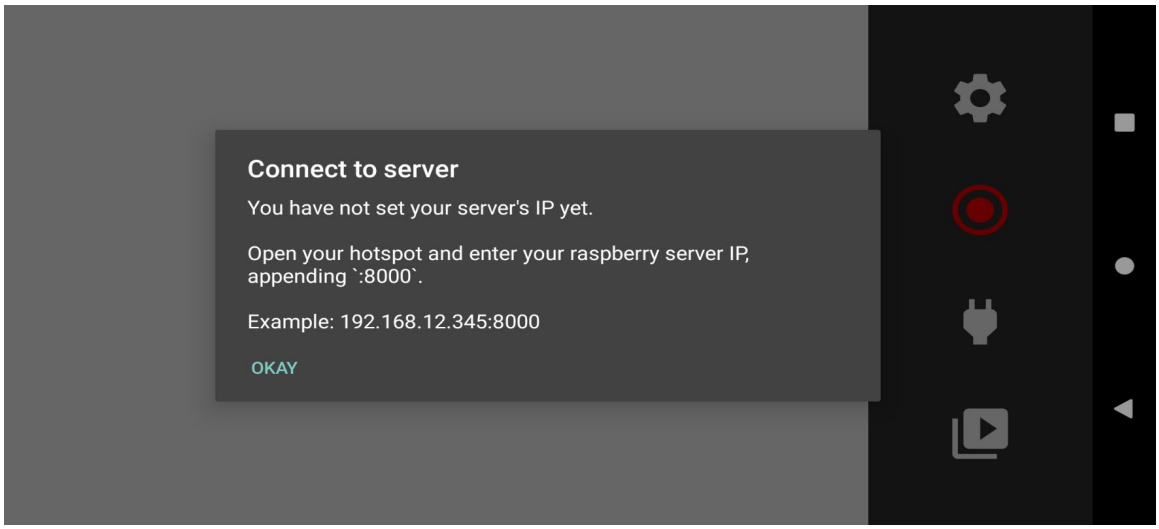
عدم وجود دیلی و لگ در استریم و ضبط

یکی از مشکلاتی که در پیاده‌سازی پروژه به آن برخورد کردیم، وجود لگ و دیلی در استریم به دلیل ویژگی‌های پروتکل‌های مختلف بود; به عنوان مثال پروتکل tcp به دلیل بررسی هر packet فرستاده شده به آن باعث بروز دیلی بسیار زیاد در استریم می‌شد. پس از بررسی پروتکل‌های مختلف به این نتیجه رسیدیم که استفاده از پروتکل HTTP بهترین نتیجه را دارد و می‌توان آن را بر روی برنامه‌های اندروید به راحتی نمایش داد.

مشکل بزرگ دیگری که زمان زیادی را صرف حل آن کردیم، عدم توانایی ضبط و استریم همزمان در رزبری پای بود. سیستم رزبری پای به دلیل داشتن یک بافر برای بایت‌های فرستاده شده توسط دوربین و داشتن یک Thread توانایی استریم و ضبط همزمان را به صورت lag free نداشت و هنگامی که می‌خواستیم استریم را ضبط کنیم، با دیلی بسیار زیادی مواجه می‌شدیم. در نهایت توانستیم با ضبط استریم فرستاده شده توسط گوشی سربار ضبط و استریم همزمان را از دوش رزبری پای برداریم و بین پردازنده و گوشی کاربر تقسیم کنیم، که نتیجه‌ی آن یک استریم روان و بدون لگ است که می‌توان آن را به مدت زمان دلخواه ضبط کرد.

نمایش اخطار در صورت عدم ست کردن IP

در صورتی که کاربر برای اولین بار وارد برنامه شود، و یا Cache برنامه را پاک کرده باشد، برنامه به او هشداری مبنا بر عدم وجود IP می‌دهد که نمای آن در تصویر مقابل آمده است.



شکل ۸ - اخطار مبنی بر عدم وارد کردن **IP**

اجرای سرور در رزبری پای

شکل **۱۰** شروع سرور از طریق اجرای کد پایتون

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.

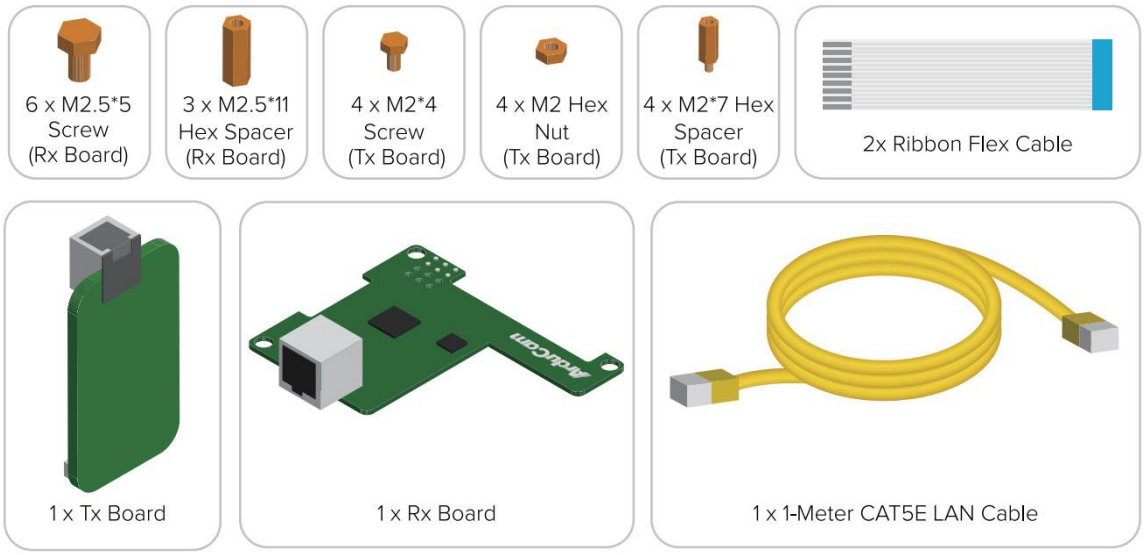
قیمت

ردیف	قطعه	قیمت
۱	رزبری پای ۳	۴,۱۵۰,۰۰۰ تومان
۲	ماژول دوربین ۵ مگاپیکسلی	۱۵۹٬۵۰۰ تومان
۳	کیت Cable Extension	۱٬۳۷۵٬۰۰۰ تومان
۴	جمع	۵٬۶۸۴٬۵۰۰ تومان

جدول ۲ - قیمت قطعات و محصول نهایی

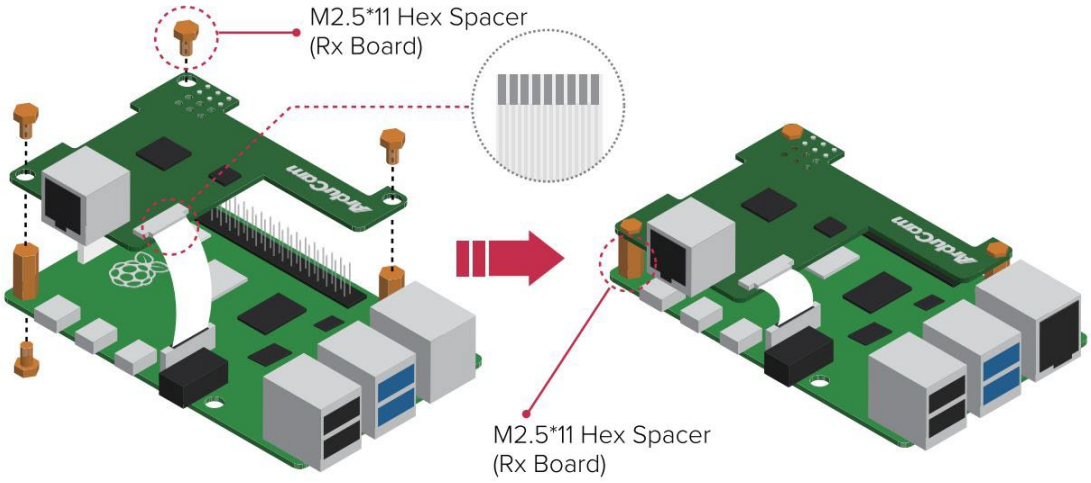
بسته‌بندی محصول

قطعات مورد نیاز



شکل ۱۰ - قطعات لازم محصول نهایی

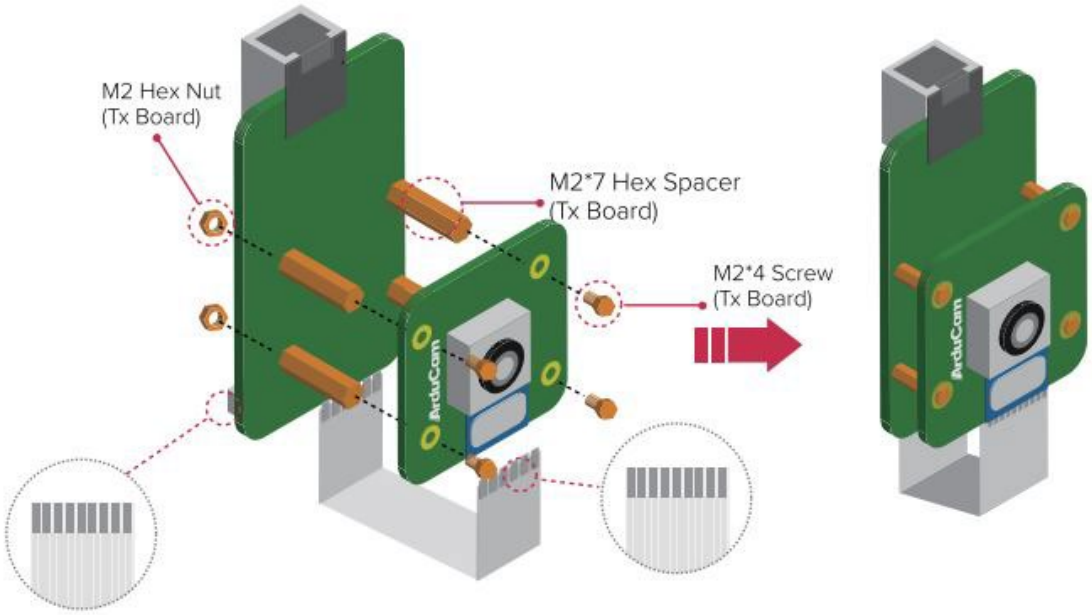
مرحله‌ی اول



شکل ۱۱ - وصل کردن برد TX به رزبری

ابتدا بورد RX را با استفاده از کابل نواری و پیچ‌های مربوط به رزبری‌پای ۳ همانند شکل بالا متصل می‌کنیم.

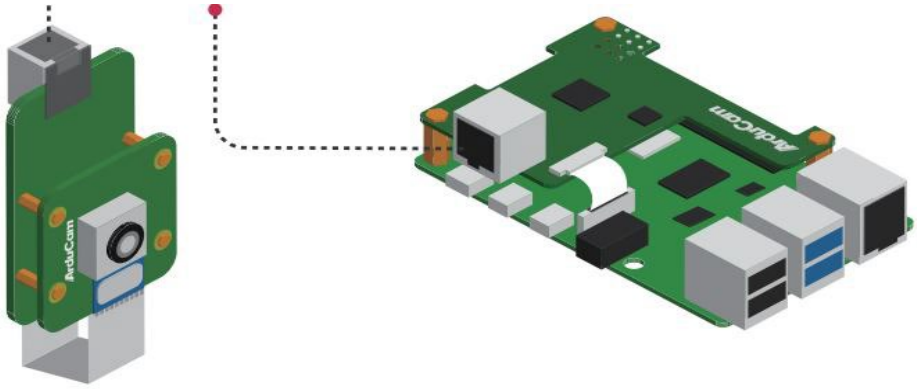
مرحله‌ی دوم



شکل ۱۲ - وصل کردن ماژول دوربین به برد RX

در ادامه ماژول دوربین را با استفاده از کابل نواری دوم و پیچ‌های مربوط به برد TX متصل می‌کنیم.

مرحله‌ی سوم

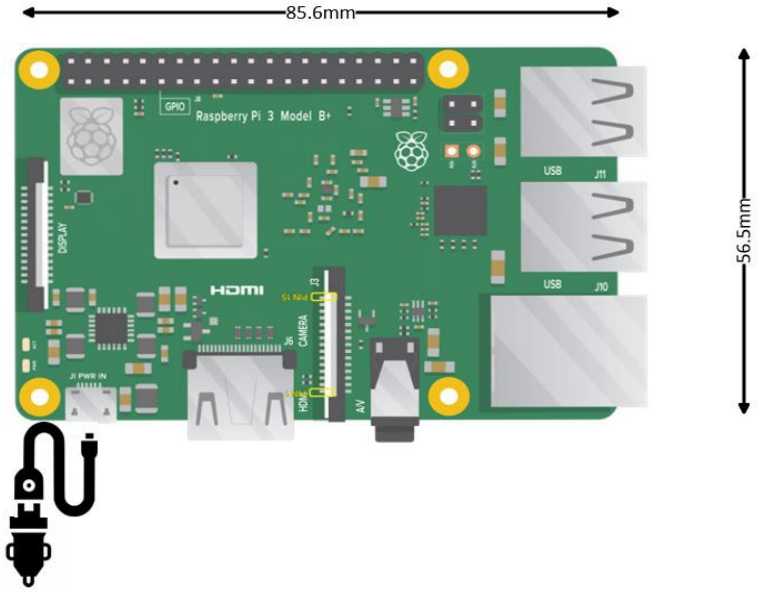


شکل ۱۳ - وصل کردن برد RX به رزبری

در نهایت ۲ برد TX و RX را با استفاده از یک کابل LAN به هم متصل می‌کنیم.

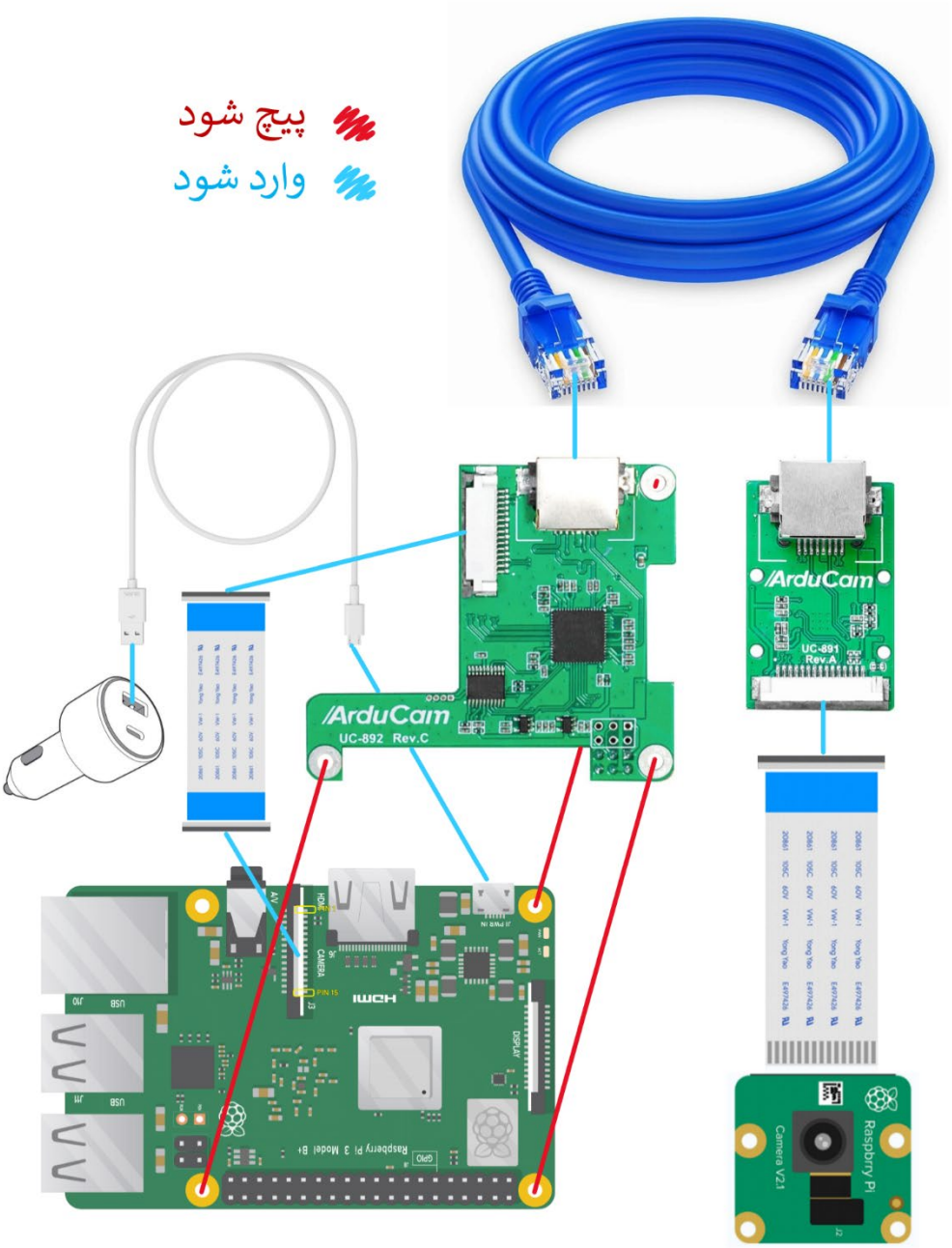
* برای کارکرد بدون لگ و دیلی، توصیه می‌شود طول کابل LAN بیشتر از ۱۵ متر نباشد.

مرحله‌ی چهارم



شکل ۱۴ - وصل کردن رزبری به شارژر فندکی ماشین

در نهایت برای کارایی مدار نیاز داریم با استفاده از کابل نوع C، آن را به شارژر فندکی وسیله‌ی نقلیه متصل کنیم. در تصویر بعدی، شکل کلی اتصال قطعات را در یک نمای واقعی‌تر مشاهده می‌کنید.

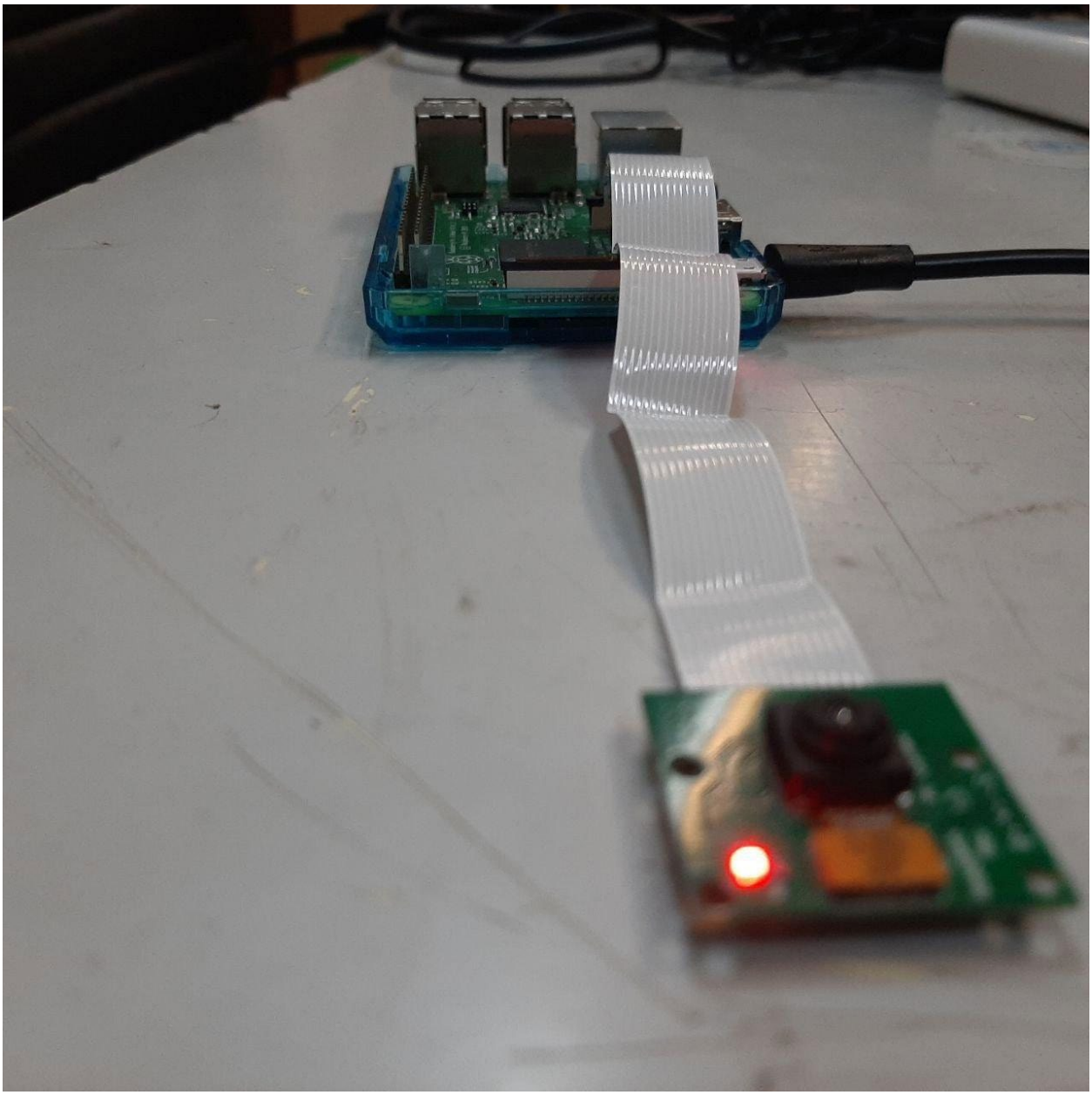


شکل ۱۵ - نمای کلی بسته بندی محصول

نماهایی از محصول

در ادامه می‌توانید نماهایی از محصول را مشاهده کنید.

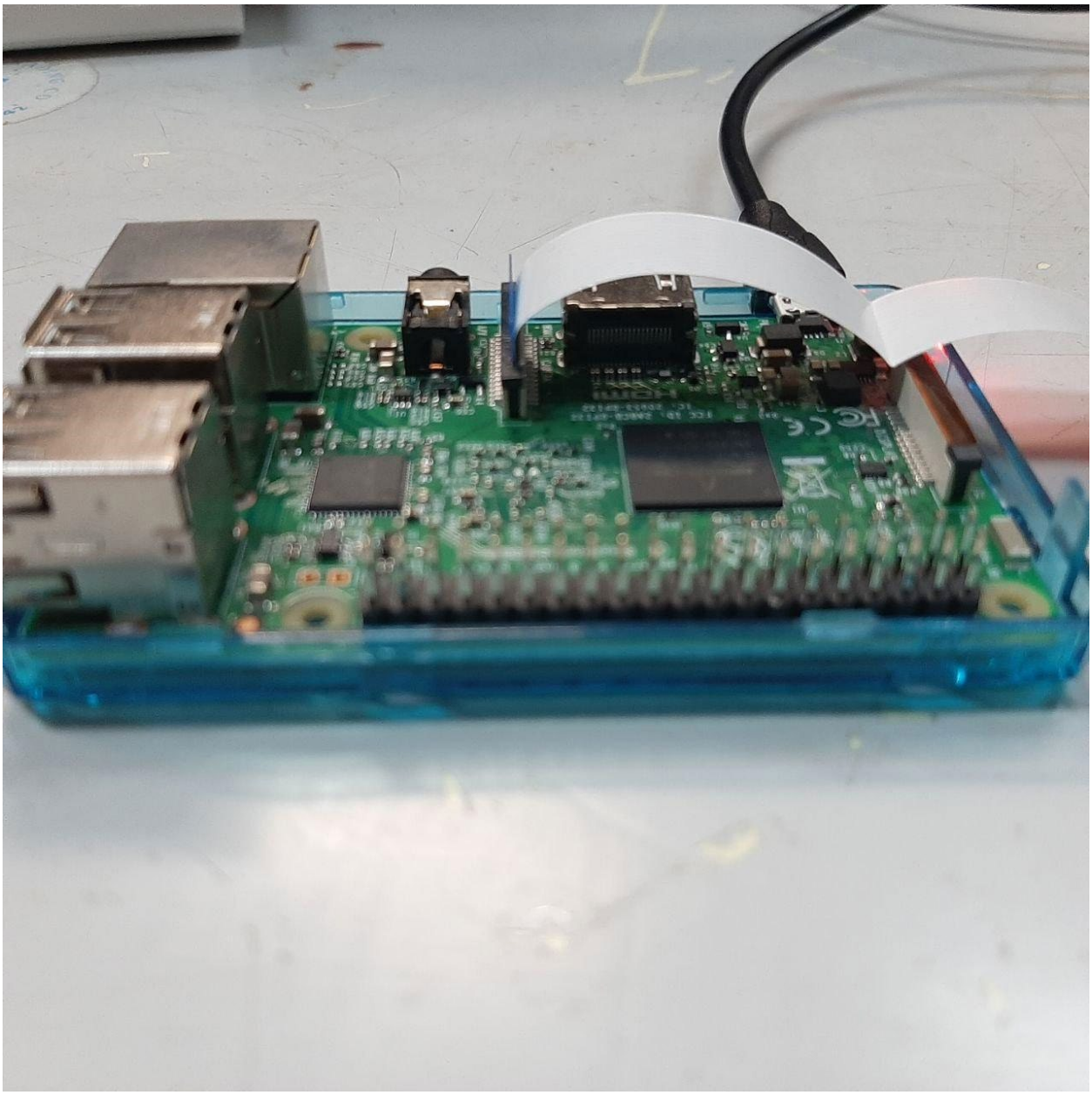
شکل ۱۶ - نمایی از محصول



شکل ۱۷ - نمایی از محصول

شکل ۱۸ - نمایی از محصول

شکل ۱۹ - نمایی از محصول



شکل ۲۰ - نمایی از محصول

فصل ۵

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.