



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر
پروژه درس ساختار و زبان کامپیوتر

عنوان:

پروژه سیستم امنیتی

Security System Project

اعضای گروه

محمد مهدی فراهانی - ۴۰۰۱۷۰۳۴۱

امیر حسین نقد علی - ۴۰۰۱۰۵۲۹۶

سید علی جعفری - ۴۰۰۱۰۴۸۸۹

دستیار مسئول

آقای روزبه پیراعیادی

استاد

دکتر حسین اسدی

بهمن ۱۴۰۱



۱ توضیحات قسمت رابط کاربری و دوربین

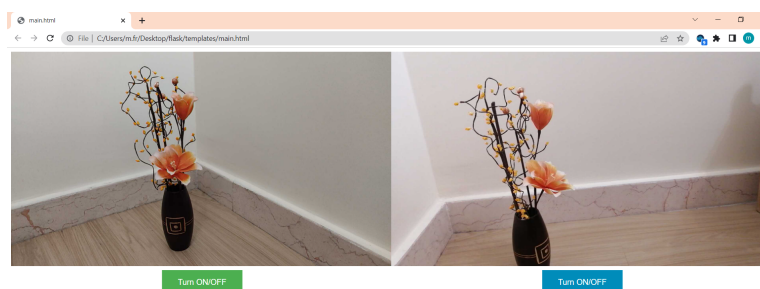
۱-۱ دوربین

برای پیاده سازی دوربین از نرم افزار IP Webcam استفاده کردیم. نحوه کار این برنامه به این صورت است که پس از باز کردن برنامه، گزینه start server را انتخاب میکنیم و دوربین فعال میشود و اکنون میتوانیم از همه دستگاه هایی که با تلفن همراه در یک شبکه هستند ویدئو را به صورت زنده ببینیم. برای اینکار میتوانیم در browser به آدرس `http://(camera ip):8080/video` مراجعه کنیم.

۲-۱ رابط کاربری

۱-۲-۱ front end

برای پیاده سازی این قسمت از html و css استفاده شده است. ظاهر صفحه به این صورت است که تصویر دو دوربین کنار یکدیگر در بالای صفحه قرار دارد و زیر تصویر هر دوربین دکمه ای برای فعال کردن و یا غیرفعال کردن آن قسمت از سیستم قرار دارد. برای نشان دادن تصاویر دوربین ها از تگ image استفاده میکنیم و src attribute آن را برابر url دوربین قرار میدهیم. با فشردن دکمه های زیرتصاویر، html یک request به سرور میفرستد که اگر مربوط به قسمت یک (دوربین سمت چپ) باشد، این get request یک کلید با نام cam1 و اگر قسمت دو باشد، یک کلید با نام cam2 خواهد داشت. سرور بر اساس این کلید ها متوجه میشود که کدام قسمت را در حالت غیر فعال قرار دهد.



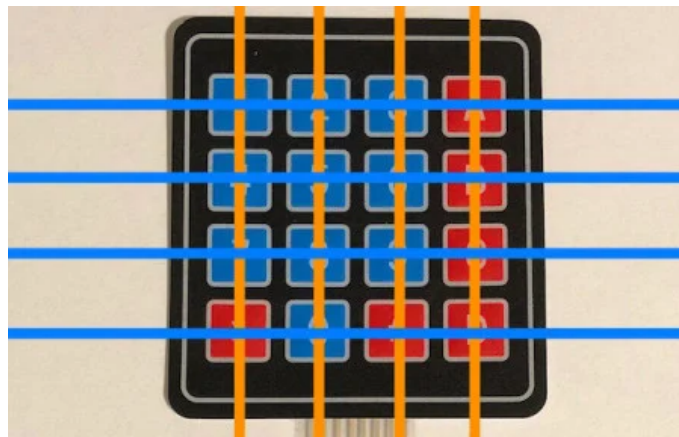
شکل ۱: تصویری از رابط کاربری

برای پیاده سازی این قسمت از flask framework استفاده شده است. در این برنامه localhost:5000 به تابع home مپ شده است و کد html مرحله قبل را برمیگرداند. و در تابع activate که به localhost:5000/activate مپ شده است پارامترهای درخواست get را بررسی میکند و سیستمی که نیاز به تغییر حالت دارد را شناسایی میکند و محتوای فایل activation.txt (که نشان دهنده فعال یا غیر فعال بودن سیستم است) را تغییر میدهد.

۲ توضیحات قسمت keypad

۱-۲ ساختار و نحوه کارکرد وسیله

از این وسیله به منظور ثبت عدد یا علائم به شکل فیزیکی استفاده می‌شود. انواع مختلف این وسیله شامل تعداد مختلفی کلید و نتیجتاً سیم هستند. نوع keypad استفاده شده در این پروژه ۴*۳ می‌باشد. به شکلی که به ازای هر سطر و ستون یک سیم وجود دارد و به هنگام فشردن یک کلید خروجی سیم‌های متناظر با سطر و ستون آن high می‌شود و از این طریق مشخص می‌شود که کدام کلید فشرده شده است. برای نمونه، شکل ؟؟ را ببینید.



شکل ۲: یک نوع keypad 4*4 و اتصالات آن [منبع]

۲-۲ توضیحات کد

۱-۲-۲ ذخیره‌سازی و تغییر رمز

در این پروژه عمدتاً از این وسیله برای احراز درستی رمز یا تغییر آن استفاده می‌شود. رمز سیستم در صورت تغییر در یک فایل (به نام password.txt) نوشته می‌شود و در نتیجه در صورت راه اندازی مجدد رمز بازیابی می‌شود. رمز پیش فرض در نخستین راه اندازی برنامه "1234" در نظر گرفته شده است.

فرایند تغییر رمز به این صورت می‌باشد که پس از فشردن کلیدی خاص (جلوتر درباره نحوه انتصاب و کارکرد کلیدها صحبت می‌شود) ابتدا با دریافت و بررسی رمز صلاحیت فرد برای تغییر آن را بررسی می‌کند و سپس رمز جدید و تکرار آن (به منظور جلوگیری از خطا) دریافت می‌شود.

همچنین همانطور که توضیح داده شد در صورت موفقیت آمیز بودن این روند رمز جدید در فایل ذخیره می شود.

۲-۲-۲ تنظیمات ابتدایی GPIO

همانطور که در؟؟ ذکر شد این وسیله به تعداد مجموع سطرها و ستون ها اتصالات دارد و در نتیجه به بر کدام از آنها باید یک پین^۱ اختصاص یابد و نحوه آدرس دهی آنها (BCM یا BOARD) مشخص شود. همچنین باید وضعیت ورودی/خروجی بودن آنها و انتصاب حالت پیش فرض (در صورت نیاز) صورت گیرد.

۳-۲-۲ تخصیص ویژگی به کلیدها

ابتدا باید یک character (یا حتی یک رشته^۲ از آنها) به هر کلید منتصب کرد. در واقع محدود به علائم روی وسیله نمی باشیم و می توان هر علامت یا کارکردی را برای هر کلید تعریف کرد. در این برنامه از همان علامت از قبل منتصب به آن استفاده می کنیم. همچنین از کاراکتر * به عنوان نشانه پایان رمز و عملگر تغییر حالت دوربین استفاده می کنیم. کاراکتر # به عنوان عملگری برای تغییر رمز استفاده می شود. دیگر کلیدها صرفا به عنوان عدد (درون رمز) به کار می روند و کارایی ویژه دیگری ندارند. همچنین شایان ذکر است یک رمز تنها می تواند شامل اعداد باشد و علائم * و # در آن وجود ندارند.

۴-۲-۲ منطق برنامه

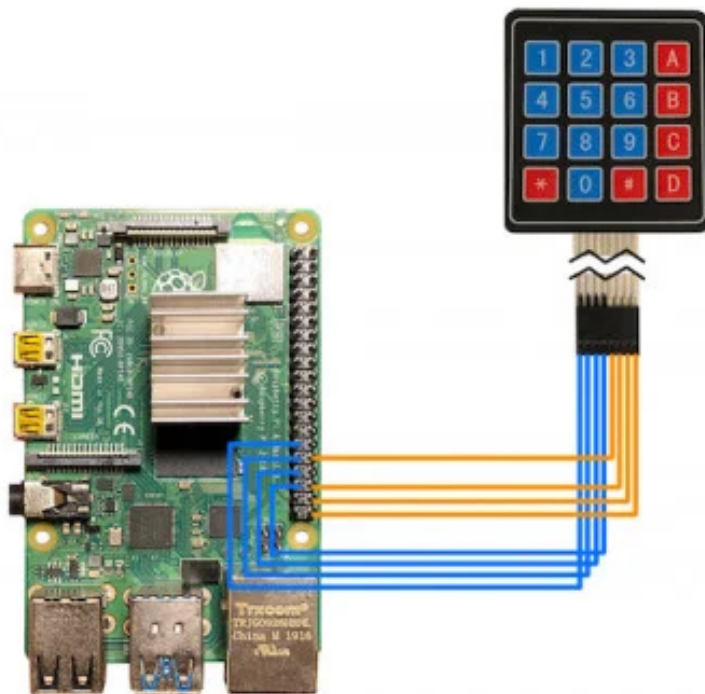
اصل برنامه درون یک حلقه قرار دارد و منتظر فشردن کلیدی می ماند. در صورتی که کلیدی که فشرده می شود معادل انجام عملکرد خاصی در نظر گرفته شده باشد به انجام آن می پردازد و در غیر این صورت آن را نادیده می گیرد. (توضیحات تابع read_input) همچنین از این تابع برای دریافت رمز نیز استفاده می شود که همانطور که توضیح داده شد تا زمانی که * فشرده نشود به دریافت رمز می پردازد.

¹pin

²string

۳-۲ یستن مدار مربوطه

در شکل ؟؟ یک نوع keypad ۴*۴ و اتصال آن به برد Raspberry Pi را مشاهده می‌کنید.

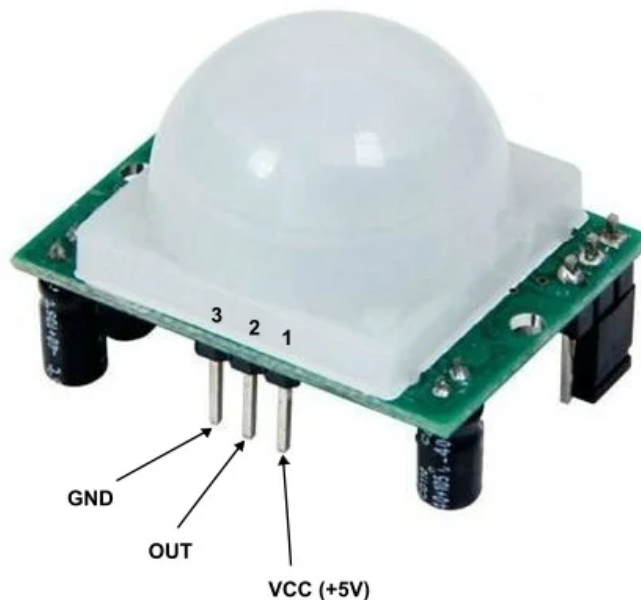


شکل ۳: یک نوع keypad 4*4 و نحوه اتصال آن به برد [منبع]

۳ توضیحات قسمت تشخیص حرکت

۱-۳ ساختار و نحوه کارکرد وسیله

برای اینکه بتوانیم حرکت را تشخیص دهیم، از PIR Motion Sensor استفاده کردیم. کاری که این سنسور انجام می دهد، تشخیص تشعشعات حرارتی مادون قرمز می باشد؛ پس می توان از آن برای شناسایی موجودات زنده ای که از خود اشعه مادون قرمز ساطع می کنند استفاده کرد. زمانی که این سنسور حرکتی را حس کند مقدار ولتاژ خروجی آن زیاد می گردد. اما اگر حرکتی حس نشود، خروجی ولتاژ آن کم است.



شکل ۴: PIR Motion Sensor

۲-۳ توضیحات کد

۱-۲-۳ اتصال و مقداردهی اولیه

در ابتدای کد این قسمت از برنامه مقداردهی پین های متصل به سنسور تشخیص حرکت و همچنین Buzzer انجام شده است. در ادامه نیاز است به مدت ۱۵ ثانیه وقفه ایجاد گردد تا سنسور تشخیص حرکت بتواند مقداردهی اولیه گردد. حال سنسور آماده استفاده است.

۲-۲-۳ پخش صدا از طریق Buzzer

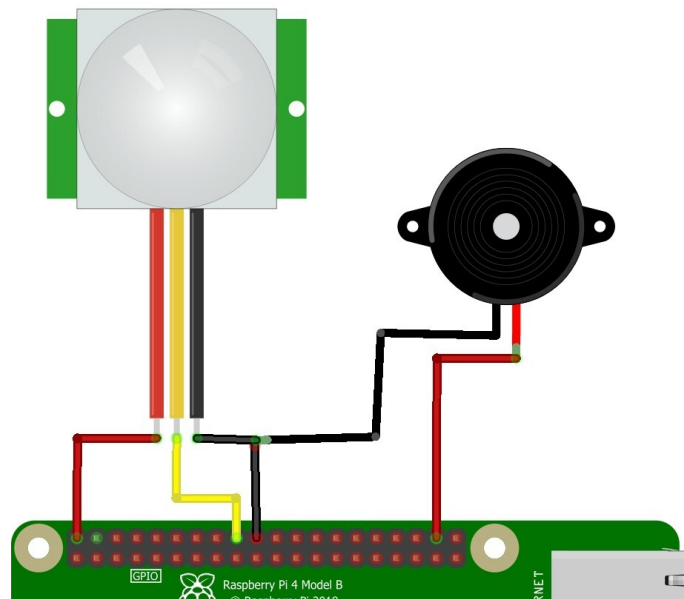
در ادامه تابعی داریم که حلقه ای در آن می باشد و به این صورت عمل می کند که زمانی که سنسور حرکت تشخیص داد و همچنین سیستم امنیتی ما فعال بود، وارد بلوک شرطی شده و صدای Buzzer فعال می گردد. بعد از مدت 0.2 ثانیه صدا قطع شده و این عمل دوباره تکرار می گردد.

۳-۲-۳ ارسال ایمیل در صورت تشخیص حرکت

بعد از اینکه حرکت تشخیص داده شد و صدا ایجاد شد، در این قسمت از کتابخانه ای در پایتون استفاده کردیم تا بتوانیم هشدار را برای کاربر ایمیل کنیم؛ تا کاربر بتواند متوجه شود که فردی قصد ورود غیر مجاز داشته است.

۳-۳ بستن مدار مربوطه

شکل کلی بستن مدار سنسور تشخیص حرکت و تولید کننده صدا به صورت زیر است:

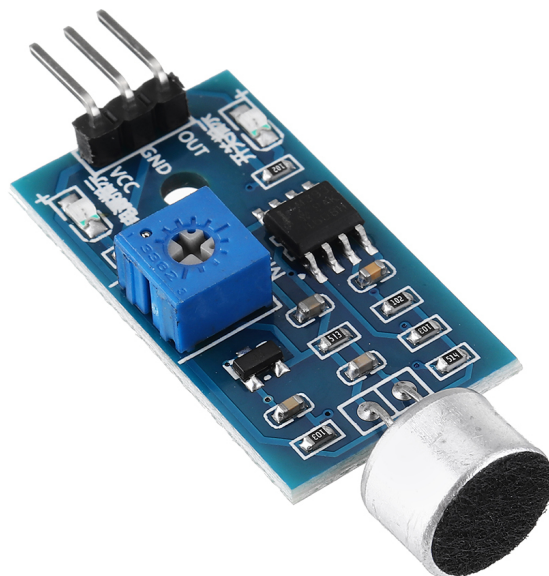


شکل ۵: PIR Motion Sensor and Buzzer circuit

۴ * قسمت امتیازی: توضیحات قسمت تشخیص صدا

۱-۴ ساختار و نحوه کارکرد وسیله

این سنسور یک برای تشخیص صدا استفاده می گردد. نحوه کار این سنسور بدین صورت است که این سنسور دارای یک میکروفون می باشد که موج صوتی صدا را تشخیص داده و سپس آن را برای برای تراشه LM393 خود می فرستد. این تراشه نیز آن را به صورت سیگنال دیجیتال خروجی می دهد. این سنسور مجهز به یک پتانسیومتر هم می باشد که می توان با آن حساسیت تشخیص صدا را تغییر داد.



شکل ۶: Sound Detection Sensor

۲-۴ توضیحات کد

۱-۲-۴ اتصال و مقداردهی اولیه

در ابتدای کد این قسمت از برنامه مقدار دهی پین های متصل به سنسور تشخیص صدا انجام شده است.

۲-۲-۴ تشخیص صدا

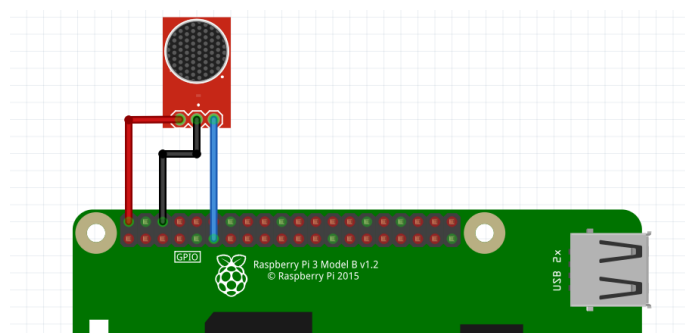
در ادامه تابعی وجود دارد که در صورتی که صدا تشخیص داده شود، توسط صدا ساز، صدایی ایجاد می کند. البته شرطی که اجرای این قسمت دارد، فعال بودن سیستم نیز می باشد و در صورت غیرفعال بودن آن صدایی ایجاد نشده و ایمیلی ارسال نمی گردد.

۳-۲-۴ ارسال ایمیل در صورت تشخیص صدا

بعد از اینکه صدا تشخیص داده شد و صدا ایجاد شد، در این قسمت از کتابخانه ای در پایتون استفاده کردیم تا بتوانیم هشدار را برای کاربر ایمیل کنیم؛ تا کاربر بتواند متوجه شود که فردی قصد ورود غیر مجاز داشته است.

۳-۴ بستن مدار مربوطه

شکل کلی بستن مدار سنسور تشخیص صدا به صورت زیر است:



شکل ۷: Sound Detection circuit