



آزمایشگاه سخت افزار

بررسی پروژه ی کلید دوپل هوشمند

مجید گروسی ۹۶۱۰۹۸۵۵ - الناز معصومی ۹۶۱۰۶۱۰۶

۱ مقدمه

هدف از این پروژه طراحی و ساخت یک کلید دو پل هوشمند است. منظور از هوشمند این است که این کلید باید علاوه بر کارکرد عادی ش، توسط یک ساز و کار بی سیم (مانند وایرلس، بلوتوث یا...) قابل کنترل باشد. به این منظور در این سند روش های انجام بخش های مختلف پروژه بررسی می شود. در نهایت این روش ها با یکدیگر مقایسه شده و ترکیبی از آن ها به منظور انجام پروژه ی نهایی انتخاب می شود.

۲ بررسی

۱.۲ پروتکل ارتباطی

به منظور برقراری ارتباط بی سیم، به طور کلی دو روش WiFi و Bluetooth قابل توجه هستند.

۱.۱.۲ Bluetooth

در این روش ارتباط به این شکل خواهد بود که کاربر با ابزاری مانند موبایل به قطعه متصل می شود و توسط یک برنامه ی کاربردی تحت ساز و کار بلوتوث کلیدها را خاموش/روشن می نماید.

۲.۱.۲ WiFi(IEEE 802.11)

در این روش برای اتصال به کلید و تغییر حالت آن، لازم است کاربر به یک شبکه ی WiFi متصل شود و در آن تنظیمات دلخواه را انجام دهد. مزیتی که این روش نسبت به بلوتوث دارد این است که این شبکه ی WiFi می تواند شبکه ی خود خانه باشد که دستگاه های کاربر (موبایل، کامپیوتر و...) در حالت عادی به آن متصل هستند.

اگر قرار بر استفاده از WiFi باشد می توان از بردهای NodeMCU(ESP8266) استفاده نمود. اما برای استفاده از بلوتوث علاوه بر یک برد پردازنده (مانند آردوینو) نیاز به یک ماژول اضافی بلوتوث هم هست؛ این امر کمی باعث افزایش پیچیدگی خواهد شد.

۲.۲ رابط کاربری مجازی

در این بخش به این پرداخته می شود که یک کاربر پس از اتصال به شبکه ی ارتباطی کلید، با چه رابطی حالت آن را تغییر می دهد.

۱.۲.۲ برنامه‌ی کاربردی (Application)

در این حالت یک برنامه‌ی کاربردی (مثلا برنامه‌ی اندروید) برای ارتباط با کلید ساخته می‌شود و وظیفه‌ی ارسال پیام‌ها را این برنامه بر عهده می‌گیرد.

۲.۲.۲ وب (Web)

در این حالت یک رابط کاربری تحت وب در اختیار کاربر قرار می‌گیرد. در این رابط کاربری یک پنل قرار دارد که کاربر توسط آن وضعیت کلیدها را مشخص می‌کند. البته خود این روش نیز ۲ حالت می‌تواند داشته باشد.

یک روش این است که پیام‌های Http به خود کلیدها ارسال شوند؛ اما یک روش دیگر این است که یک قطعه به شبکه متصل شود (مثلا یک رزبری پای) و این رابط کاربری روی آن اجرا شود. در این صورت پیچیدگی پیاده‌سازی کلیدها کاهش می‌یابد و می‌تواند با یک ساختار ساده‌تر (مانند سوکت عادی) با رزبری پای در ارتباط باشند.

از مزایای روش نهایی می‌توان به این اشاره کرد که در ادامه امکانات دیگر (زمان‌بندی روشنایی، ثبت و پردازش گزارش از تمام کلیدهای داخل خانه و...) با سرعت و راحتی بیشتری می‌توانند روی رزبری پای پیاده‌سازی شوند.

۳.۲ رابط کاربری فیزیکی

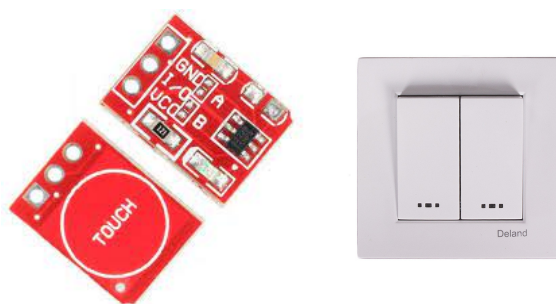
بحث دیگری که در این قسمت مطرح می‌شود این است که آن کلید از نظر ظاهری چه ساختاری داشته باشد.

۱.۳.۲ کلید دوپل عادی

در این حالت رابط فیزیکی مانند یک کلید دوپل عادی خواهد بود؛ مانند سمت راست تصویر زیر.

۲.۳.۲ کلیدهای لمسی

در این حالت به جای کلیدهای مکانیکی که در کلیدهای دوپل عادی استفاده می‌شوند از کلیدهای لمسی استفاده می‌شود. مانند سنسور خازنی TTP۲۲۳ که در سمت چپ تصویر زیر قابل مشاهده است.



۴.۲ ارتباط کلید و مصرف‌کننده

۱.۴.۲ ارتباط باسیم

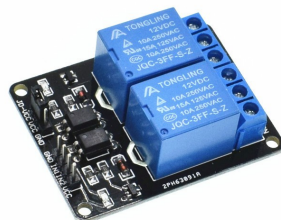
این روش شباهت بیشتری به روش معمول سیم‌کشی خانه‌ها دارد. به این صورت که در کلید دوپل یک رله وجود دارد که توسط برد کنترلرش قطع/وصل می‌شود و خروجی این رله مستقیماً به دستگاه (مثلا لامپ) متصل است.

۲.۴.۲ ارتباط بی سیم

در این روش می توان یک کنترلر برای کلید و یک کنترلر برای مصرف کننده قرار داد. در این صورت با قطع/وصل شدن کلید، کنترلر کلید به کنترلر مصرف کننده فرمان می دهد که تغذیه ی دستگاه را قطع/وصل کند. گرچه این روش نمونه هایی در بازار دارد، اما در این پروژه به دلیل افزودن پیچیدگی از آن استفاده نخواهیم کرد.

۵.۲ قطع و وصل کردن اتصال

برای قطع و وصل کردن جریان از داخل کلید، معمول ترین روش استفاده از رله است. از آنجایی که ما قصد پیاده سازی کلید دویپل داریم، استفاده از یک ماژول رله ی ۲ کاناله با ولتاژ کنترل ی ۵ ولت منطقی به نظر می رسد. نمونه ای از این ماژول در تصویر زیر قابل مشاهده است.



۶.۲ تنظیم کلید

بدیهی است که کلیدها تنظیماتی دارند. از جمله ی این تنظیمات می توان به آدرس IP در شبکه، نام شبکه، رمز عبور و... اشاره نمود. یک راه ایجاد این تنظیمات اصطلاحاً Hardcode کردن آن ها در کد منبع برنامه است.

اما روش دیگری که منطقی تر است این است که این تنظیمات تحت یک رابط سریال که معمولاً روی تمام بردها وجود دارد انجام شوند. مثلاً فرد تنظیم کننده لپ تاپش را با یک کابل USB به برد NodeMCU متصل می کند و یک پوسته (shell) در اختیارش قرار می گیرد. سپس وی در این پوسته دستورات مورد نظرش را وارد می کند تا تغییرات مورد نظرش اعمال شوند.

۳ جمع بندی

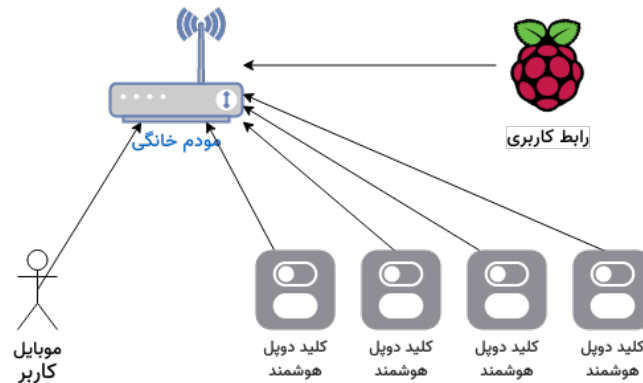
با توجه به موارد ذکر شده در فصل پیشین می توان این طور جمع بندی نمود که برای کنترلر کلیدها از برد NodeMCU استفاده خواهد شد و ارتباط بی سیم تحت شبکه ی WiFi که به احتمال زیاد در محیط وجود دارد صورت می گیرد.

همچنین یک گره ی مرکزی در شبکه وجود خواهد داشت که رابط کاربری تحت وب در اختیار کاربران قرار می دهد، پیام های کنترلی را به کلیدها ارسال می کند و همچنین وضعیت کلیدها را به کاربر نمایش می دهد. این گره می تواند یک برد رزبری پای باشد و یا هر سیستم لینوکسی که در تمام طول شبانه روز روشن است.

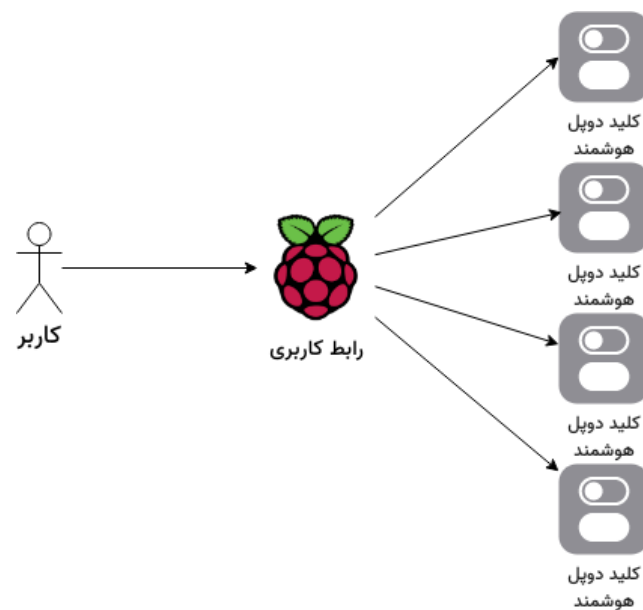
همچنین ظاهر فیزیکی این شبکه نیز شامل ۲ سنسور تماس خواهد بود که با لمس آن ها وضعیت هر پل کلید تغییر می کند. برای جلوگیری از ابهام، می توان به ازای هر پل ۲ عدد LED سبز/قرمز نیز در سامانه قرار داد تا وضعیت فعلی آن کلید را مشخص کنند.

۱.۳ معماری

طبق توضیحات مذکور، ساختار اتصالات اجزا در شبکه از این قرار خواهد بود:



همچنین اجزایی از شبکه که با یکدیگر ارتباط خواهند داشت در تصویر زیر قابل مشاهده است. یعنی کاربر با رابط مرکزی در ارتباط خواهد بود و آن رابط دستورات کاربر را به کلیدهای مختلف ارسال خواهد نمود.



۲.۳ زیرسیستم‌ها

کدهای این پروژه به طور عمده در زیرسیستم‌های زیر قرار می‌گیرند. این دسته‌بندی با این هدف انجام شده است که هر فرد بتواند با حداقل درگیری با قسمت‌های دیگر پروژه، یک قابلیت خاص از سیستم را پیاده‌سازی نماید.

۱.۲.۳ مدیریت رله‌ها

این قسمت از پروژه وظیفه‌ی پایه‌ای مدیریت کلید را بر عهده دارد؛ یعنی وضعیت هر کلید را نگهداری کند، با توجه به این وضعیت رله‌ها را کنترل کند و همچنین با توجه به اتفاقاتی که در سامانه رخ می‌دهد (لمس سنسورهای فیزیکی/دریافت دستور خاموش و روشن از شبکه) این وضعیت را تغییر دهد.

۲.۲.۳ ارتباطات شبکه‌ای کلید

این قسمت وظیفه دارد تا رابطی در اختیار رزبری پای قرار دهد و دستورات را از آن دریافت کند. همچنین چنان چه نیاز به گزارش‌گیری از کلیدها باشد، زیرسیستم مدیریت رله‌ها تابعی از این زیرسیستم را فراخوانی می‌کند تا اطلاعات به رزبری پای ارسال شوند.

۳.۲.۳ درایورهای کلید

برای این که برد رزبری پای بتواند دستوراتی به کلیدها ارسال کند، نیاز به پیروی از یک ساز و کار (Protocol) خاص دارد. در این بخش این ساز و کار که قرار است روی رزبری پای اجرا شود پیاده‌سازی می‌شود و به عنوان یک واسطه بین سایت و کلیدها عمل می‌کند؛ دقیقاً مانند یک درایور سیستم‌عامل که رابطی است بین سیستم‌عامل و سخت‌افزار مربوطه.

۴.۲.۳ رابط کاربری (سایت)

این قسمت در واقع یک پروژه‌ی وب است که با Python/Django پیاده‌سازی می‌شود و روی رزبری پای قرار می‌گیرد. کاربران سامانه این وب‌سایت را روی دستگاه‌هایشان باز خواهند کرد و دستورات روشن/خاموش کلید دلخواهشان را به سامانه ارسال می‌کنند. لیست کلیدها و آدرس شبکه‌ی آن‌ها نیز روی پایگاه داده‌ی همین وب‌سایت ذخیره خواهد شد.

۳.۳ لیست قطعات و تخمین هزینه

لیست قطعات لازم برای کلید دو پل و همچنین رابط کاربری مرکزی به همراه تخمین هزینه‌های لازم در دو جدول زیر موجود است.

ردیف	قطعه	هزینه (هزار تومان)
۱	برد NodeMCU	۱۳۰
۲	ماژول رله ۵ ولت ۲ کانال	۴۵
۳	سنسور تماس خازنی TTP۲۲۳ (۲ عدد)	۲۰
۴	دیود نوری LED (۵ عدد)	۱
۵	کابل رابط USB	۱۰
۶	مخارج متفرقه (سیم و...)	۱۰
	مجموع	۲۱۶

ردیف	قطعه	هزینه (هزار تومان)
۱	برد رزبری پای ۳B	۱۶۸۰
۲	کابل شبکه	۲۰
	مجموع	۱۷۰۰

۴.۳ جمع‌بندی

جمع‌بندی نهایی این است که این پروژه با هزینه‌ی ۲ میلیون تومان (با احتساب حاشیه‌ی امن) قابل انجام است. همچنین در صورت نیاز به استفاده در محیط واقعی، مقداری هزینه‌ی پکیجینگ به این پروژه افزوده می‌شود. اضافه کردن هر کلید دابل جدید به سامانه طبق توضیحات مذکور حدوداً ۲۵۰ هزار تومان هزینه خواهد داشت.