بسمه تعالی

گزارش کار پروژه پایانی درس مبانی اینترنت اشیاء

**سالن مطالعه هوشمند**

اعضای تیم:

مهسا توسلی

زهرا توکلی

دانشگاه صنعتی اصفهان

بهار 1404

**چکیده طرح**

در این پروژه با بهره گیری از اینترنت اشیاء و بدون نیاز به ناظر انسانی و به صورت خودکار، مواردی مانند سنجش حضور واقعی افراد و عدم اشغال میز مطالعه بدون استفاده از آن، سنجش میزان سکوت سالن مطالعه و هشدار به افراد در صورت رعایت نکردن سکوت، پایش شرایط محیطی و انجام اقدامات لازم انجام می شود و دانشجویان با روش های تعاملی به رعایت نظم و استفاده موثر از فضای مطالعه ترغیب می شوند.

**اهداف و شرح دقیق سناریو و تمام قوانین برنامه**

**.......**

**وسایل الکترونیکی مورد استفاده**

* سه عدد میکروکنترلر ESP8266
* دو عدد سنسور حرکت HC-SR501
* یک عدد سنسور کنترل کیفیت هوا MQ-135
* یک عدد سنسور دما و رطوبت DHT22
* یک عدد سنسور صدا FC-04
* دو عدد بازر
* تعدادی لامپ LED
* تعدادی مقاومت
* سه عدد برد بورد و سیم جامپر

**نحوه بستن مدار‌ها**

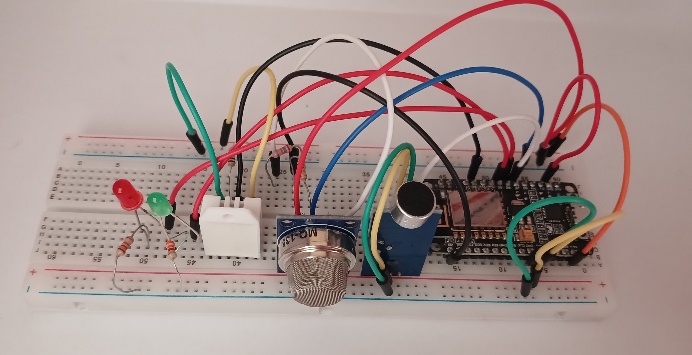
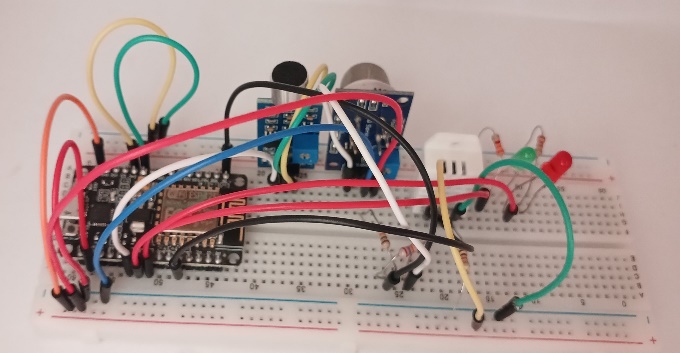
در این پروژه ما برای هر میز مطالعه از یک سنسور حرکت، یک بازر و LED برای اخطارها و یک میکروکنترلر برای ارسال و دریافت داده ها به سرور استفاده کرده ایم.

هم چنین برای هر چند میز که نزدیک به هم قرار دارند یک سنسور صدا برای سنجش میزان صدا در آن ناحیه، یک سنسور دما و یک سنسور کنترل کیفت هوا و یک میکروکنترلر برای ارتباط با سرور داریم.

سنسورها با استفاده از سیم جامپر به میکروکنترلرها متصل شده اند ولی در پیاده سازی در محیط واقعی بهتر است در صورتی که به صرفه باشد از سنسورهایی با قابلیت برقراری ارتباط بدون سیم استفاده کنیم.

میکروکنترلرها از طریق ماژول Wi-Fi با سرور ارتباط برقرار کرده و داده رد و بدل می کنند.

برای کالیبره کردن سنسورهای حرکت و صدا از پتانسیومتر تعبیه شده روی سنسور استفاده شده است. به این صورت که برای کالیبره کردن سنسور صدا پتانسیومتر به گونه ای تنظیم می شود که در سکوت LED روی سنسور خاموش و در صورت ایجاد صدا روشن شود. برای کالیبره کردن سنسور حرکت، پتانسیومترها به گونه ای تنظیم می شوند که دامنه تشخیص حرکت (برای مثال تشخیص حرکت به شعاع 3 متر از سنسور)، و مدت زمان high بود پین مربوطه در صورت تشخیص حرکت تنظیم شود. در این پروژه شعاع تشخیص و مدت زمان high بودن پایه سنسور، هر کدام روی کم ترین حد آنها یعنی 3 متر و 2.5 ثانیه تنظیم شده اند. هم چنین برای کالیبره کردن سنسور کنترل کیفیت هوا با توجه به اینکه از پین خروجی آنالوگ سنسور استفاده کرده ایم، از کد مناسب استفاده شده است. (کالیبره کردن با استفاده از پتانسیومتر روی خروجی دیجیتال سنسور اثر دارد و نه خروجی آنالوگ آن.)

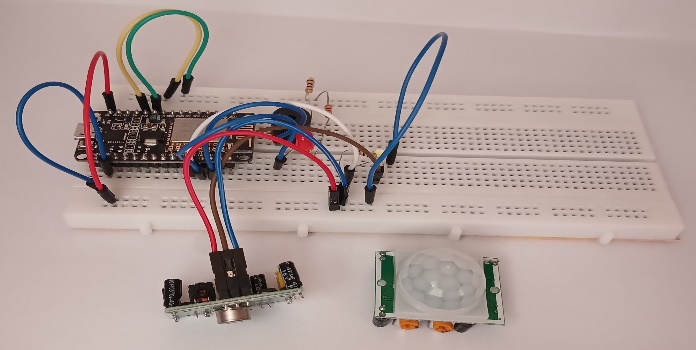
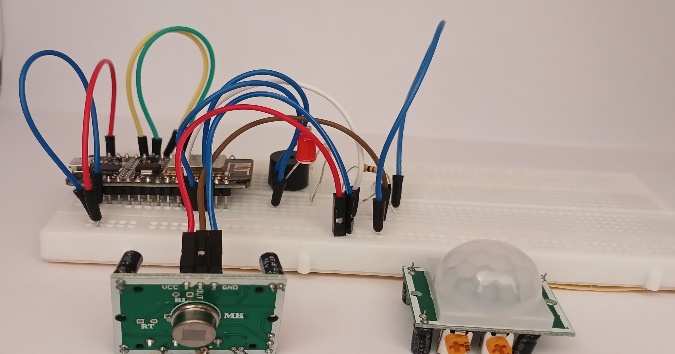
 ****

**MQ-135**

**DHT22**

**FC-04**

تصویر مدار بسته شده برای هر ناحیه از میزها جهت پایش متغیر‌های محیطی

**Buzzer**

**HC-SR501**

تصویر مدار بسته شده برای هر میز مطالعه

توجه: پوشش روی سنسور حرکت در مدار بسته شده برای میز‌ها به برای تشخیص بهتر حرکت برداشته شده است.

در مدارهای بالا برای تغذیه LED ها و بازرها از پین 3.3 ولت میکروکنترلر و برای تغذیه سنسورها از پین Vin استفاده شده است تا ولتاژ مناسب جهت راه اندازی سنسورها تأمین شود.

**توضیح کد‌های میکروکنترلر‌های مربوط به میزهای مطالعه**

کدهای میکروکنترلرها در این پروژه به دو شیوه‌ی استفاده از پروتکل HTTP و استفاده از پروتکل MQTT برای برقراری ارتباط با سرور نوشته شده اند. متاسفانه به دلیل اینکه در برقراری ارتباط با MQTT Server یا MQTT Broker پلتفرم ThingsBoard با مشکل مواجه شدیم، ادامه کار را با روش HTTP انجام دادیم ولی در اینجا کدهای MQTT نیز توضیح داده می شوند. همچنین کد‌ها در محیط آردوینو نوشته شده اند.

منطق برنامه این میکروکنترلر به این صورت است که وقتی میزی توسط کاربر رزرو می شود، میکروکنترلر وظیفه دارد در صورتی که هیچ گونه حرکتی به مدت 20 دقیقه از کاربر تشخیص داده نشد، بازر را برای هشدار به صدا در آورد و LED میز را روشن کند و این وضعیت را به سرور گزارش کند تا پیام مناسبی جهت تحویل میز و برداشتن وسایل برای کاربر در سایت نمایش دهد. سطح صدای بازر برای رعایت سکوت سالن کم بوده و در مدت زمان کوتاهی به صدا در می آید ولی با این حال بهتر است از روشهای مناسب تری برای اخطار استفاده شود. هم چنین در صورتی که در ناحیه ای سر و صدا وجود داشته باشد، میکروکنترلر دستوری مبنی بر به صدا درآوردن بازر و روشن کردن LED برای اخطار به کاربر از سرور دریافت می کند.

**روش HTTP:** ابتدا برای جلوگیری از stackoverflow سعی شده است متغیرها به صورت گلوبال تعریف شوند. متغیر deskOccupied مشخص می کند که میز توسط فردی در سایت رزرو شده است یا خیر. متغیر lastMotionTime زمان آخرین حرکت کاربر را ذخیره میکند تا در صورت گذشت 20 دقیقه از آن وضعیت به سرور گزارش شود. متغیر TIMEOUT همان 20 دقیقه را برحسب میلی ثانیه ذخیره می کند. که البته در مرحله تست برنامه این زمان 1 دقیقه در نظر گرفته شده است. برای تایمر، از تایمر نرم افزاری کتابخانه Ticker.h استفاده شده است که هر یک دقیقه مدت زمانی که از آخرین حرکت گذشته است را بررسی می کند و در صورت عدم حرکت متغیر منطقی ای را برابر true می کند تا در تابع loop اقدامات لازم انجام شود. هم چنین سنسور حرکت در صورت تشخیص حرکت وقفه ای ایجاد میکند و در تابع هندلر وقفه متغیر منطقی ای true می شود تا در تابع loop اقدامات لازم مانند تغییر متغیر lastMotionTime انجام شود. با توجه به سناریو ذکر شده در بالا میکروکنترلر موقع ارسال وضعیت عدم حرکت به سرور در نقش کلاینت و موقع دریافت دستورات سرور مبنی بر به صدا در آوردن بازر و اخطار در نقش سروری عمل می کند که سرور اصلی به آن متصل شده و دستورات را می فرستد. برای ارسال وضعیت عدم حرکت میکروکنترلر یک درخواست HTTP POST به URL مورد نظر یعنی <http://192.168.75.224:5000/sensor-data> می زند که آدرس IP با IP سرور یا نام دامنه آن جایگزین می شود. هم چنین سرور در صورتی که فردی در سایت میزی را رزرو یا آزاد کند برنامه سرور به آدرس <http://microcontroller> IP address/reserve و <http://microcontroller> IP address/release درخواست HTTP POST می زند تا میکروکنترلر مقدار متغیر deskOccupied را تغییر دهد. برای ارسال دستور اخطار به میکرو کنترلر، سرور به آدرس <http://microcontroller> IP address/sound\_alert درخواست می دهد.

در تابع setup میکروکنترلر هم کارهایی مانند راه اندازی ارتباط سریال با کامپیوتر، اتصال به Wi-Fi، راه اندازی سرور، پیکربندی پین ها و وقفه ها انجام شده است.

**روش MQTT:** در این روش ابتدا در سایت ThingsBoard بخش Cloud نظیر هر میکروکنترلر و سرور یک Device ساخته می شود و از token آن در کد برای برقراری ارتباط با MQTT Server استفاده می شود. در کد‌های این بخش هم تمام اجزا مانند تابع setup و loop، تقریبا مانند بخش قبل هستند با این تفاوت که در تابع setup میکروکنترلر سعی می کند به MQTT Server با استفاده از تابع client.connect و token ای که از سایت ThingsBoard گرفته است متصل شود و در تاپیک هایی به صورت v1/devices/me/rpc/request/+ با استفاده از تابع client.subscribe مشترک شود و subscribe کند. در تاپیک ذکر شده و تاپیک هایی که در ادامه ذکر می شوند، بخش me با token دستگاه جایگزین می شود. مشترک شدن میکروکنترلر در این تاپیک برای دریافت دستورات سرور است و در واقع سرور در این تاپیک ها دستورات را publish می کند. هر زمان سرور چیزی در این تاپیک ها منتشر کند تابع callback میکروکنترلر اجرا شده و اقدامات مناسب انجام می شود. هم چنین میکروکنترلر عدم حرکت را با استفاده از تابع client.publish در تاپیک v1/devices/me/telemetry به صورت یک document جیسون به صورت {\"timer\":\"timeout\"} منتشر میکند و سرور در این تاپیک ها قبلا subscribe کرده است و این انتقال داده ها بین میکروکنترلرها و سرور توسط MQTT Server انجام می شود.

**توضیح کد‌های میکروکنترلر‌های مربوط به پایش متغیر‌های محیطی در یک ناحیه**

منطق برنامه این میکروکنترلر به این صورت است که متغیرهای محیطی مانند دما و رطوبت و یا کیفیت هوا را با استفاده از سنسورها پایش می کند و اقدامات لازم مانند روشن کردن cooler در صورت گرم بودن محیط سالن مطالعه و یا روشن کردن heater در صورت سرد بودن محیط را انجام داده و در صورت نیاز وضعیت را به سرور گزارش می کند. در این پروژه ما از یک LED سبز رنگ به عنوان cooler و یک LED قرمز رنگ به عنوان heater استفاده کرده ایم که به ترتیب در صورتی که دما بیش از 25 درجه و کمتر از 15 درجه سانتی گراد باشد روشن می شوند. یک وظیفه دیگر میکروکنترلر این است که در صورت تشخیص سر و صدا توسط سنسور صدا آن را به سرور گزارش کند تا سرور به کاربران اخطار دهد.

**روش HTTP:** در ابتدا سعی شده است اکثر متغیرهای برنامه برای جلوگیری از stackoverflow به صورت گلوبال تعریف شوند. برای استفاده از سنسورهای دما و رطوبت و کیفیت کنترل هوا به ترتیب از کتابخانه های DHT.h و MQ135.h و اشیای کلاسهای DHT و MQ135 استفاده شده است. کتابخانه MQ135 شامل کدهای کالیبره کردن سنسور کنترل کیفیت هوا می باشد. سپس در تابع setup کارهایی مانند راه اندازی ارتباط سریال با کامپیوتر، اتصال به Wi-Fi، پیکربندی پین ها و سنسورها و وقفه ها انجام شده است. در تابع loop مقادیر سنسورهای دما و کنترل کیفیت هوا بررسی شده و اقدامات لازم مانند روشن کردن cooler و heater انجام می شود. علاوه بر آن در صورتی که سنسور صدا، صدایی را تشخیص دهد یک پالس با لبه پایین رونده تولید کرده و وقفه ای ایجاد میکند. در تابع هندلر وقفه متغیر soundDetected برابر true شده و در صورت true بودن این متغیر در تابع loop گزارش وضعیت به سرور با ارسال درخواست HTTP POST به <http://192.168.75.224:5000/sensor-data> انجام می شود.

**روش MQTT:** این روش هم از نظر انتشار یا اشتراک در تاپیک ها دقیقا مشابه روش MQTT در میکروکنترلرهای میزها است. سایر موارد مانند کدهای HTTP است که در بالا توضیح داده شده است.