

گزارش نهایی پروژه سیستم‌های نهفته

طراحی سامانه رصد وضعیت هوا

گروه سه:

کوشا جافریان ۹۵۱۰۵۴۵۴

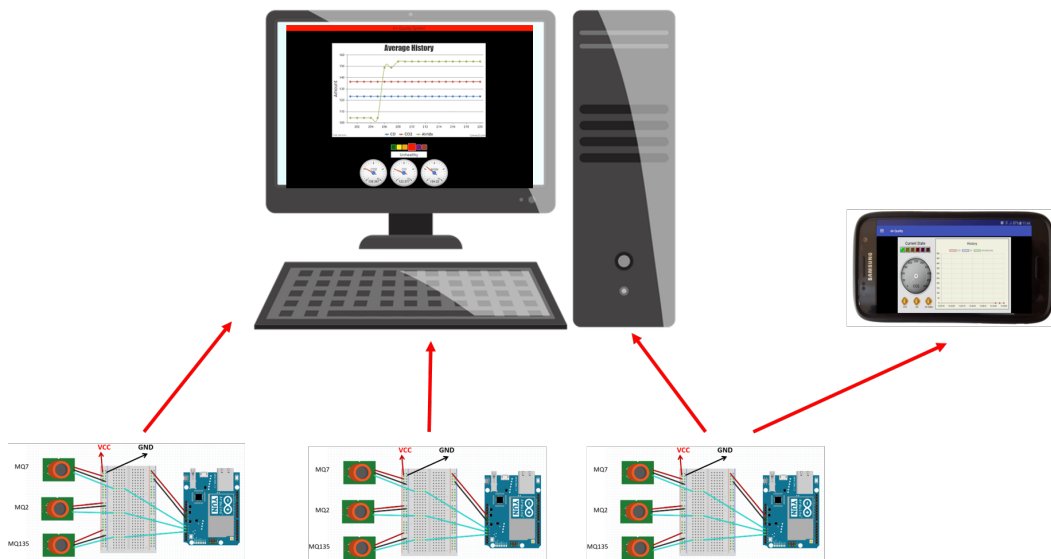
سروش باسلی‌زاده ۹۵۱۰۵۴۰۸

یاسمین طباطبایی ۹۵۱۰۴۸۶۶

۱۰ مرداد ۱۳۹۹

۱ شرح پروژه

در این پروژه ما یک سامانه‌ی *iot* رصد وضعیت هوا طراحی کردیم که وضعیت ۳ نوع گاز سمی در هوا را اندازه‌گیری می‌کند و با توجه به مقادیر این گازها اخطارهای لازم را به کاربران می‌دهد. این سامانه اطلاعات وضعیت هوا را از تعدادی دستگاه که هر یک سه سنسور مخصوص به گاز دارند می‌گیرد و میانگین وضعیت آن‌ها را نشان می‌دهد و بر اساس وضعیت گاز با بیشترین غلظت وضعیت آلودگی کلی را نشان می‌دهد. این سامانه به صورت یک اپلیکیشن گوشی پیاده‌سازی شده است که با یک لوکیشن خاص کار می‌کند و همچنین یک وب‌سرور که اطلاعات میانگین چند دستگاه را نمایش می‌دهد. شمای کلی طراحی را در شکل زیر مشاهده می‌کنید.



شکل ۱: شمای کلی سیستم

۱.۱ تغییرات نسبت به پروپوزال اولیه

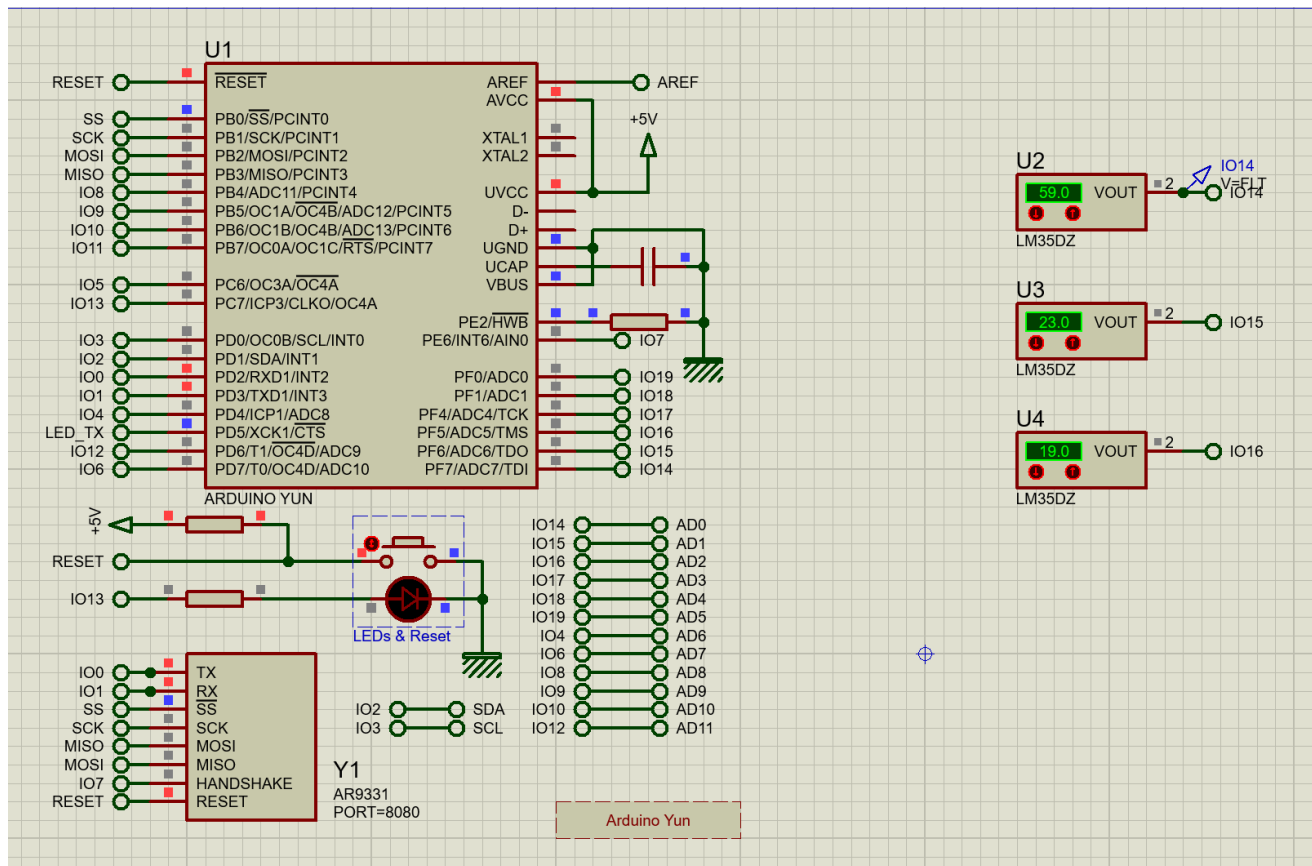
در پروپوزال پروژه ما سیستم را به صورت یک دستگاه در نظر گرفتیم اما در ارائه‌ی اولیه تصمیم بر آن شد که قابلیت داشتن دستگاه در چند لوکیشن مختلف را نیز پیاده‌سازی کنیم. بنابراین ما یک وب‌سرور طراحی کردیم که بتواند مقادیر گازها را از چند برد مختلف در لوکیشن‌های مختلف بگیرد و میانگین آن‌ها را گزارش دهد که مشابه کاری است که دستگاه‌های رصد هوای مزارک مناطق مختلف شهر انجام می‌دهند. همچنین اپلیکیشن گوشی نیز طراحی کردیم که مطابق آنچه در پروپوزال گفته شده بود از یک دستگاه آمار را می‌گیرد و مشابه یک سامانه رصد هوای خانگی برای ساختمان‌های هوشمند عمل می‌کند.

۲ طراحی و پیاده‌سازی

در این پروژه ما با استفاده از *iot-builder* مدارها را طراحی کردیم. این ابزار برای سیمولیشن بردهایی که برای کاربردهای *iot* استفاده می‌شوند و نیاز به استفاده از *wifi* دارند به کار می‌رود و قابلیت‌های گوناگونی دارد. در ادامه بخش‌های مختلف طراحی سیستم را شرح می‌دهیم.

۱.۲ شماتیک مدار

شماتیک مدار را در شکل زیر مشاهده می‌کنید. در این مدار یک برد Arduino Yun که برای کاربردهای IoT مناسب است و سه سنسور LM35DZ تعبیه شده است که سنسورهای دما هستند اما ما از آن‌ها به عنوان سنسور تشخیص گاز استفاده می‌کنیم. در پروپوزال ما سنسورهای واقعی اندازه‌گیری گاز MQ-7 و MQ-2 و MQ-135 را استفاده کرده بودیم اما چون این سنسورها در Proteus نبودند ما از سنسورهای جایگزین اندازه‌گیری دما استفاده کردیم.



شکل ۲: شماتیک مدار

۲.۲ توضیح کلی کد

کد اصلی هر دستگاه به زبان arduino زده شده است و در فایل main.ino قرار دارد. شیوه‌ی کلی کار کد هم این است که ابتدا اعداد سنسورها از پورت‌هایشان خوانده می‌شود و سپس با تبدیل زیر به مقدار مناسب در بازه‌ی سنسور گازهای هوا در می‌آیند.

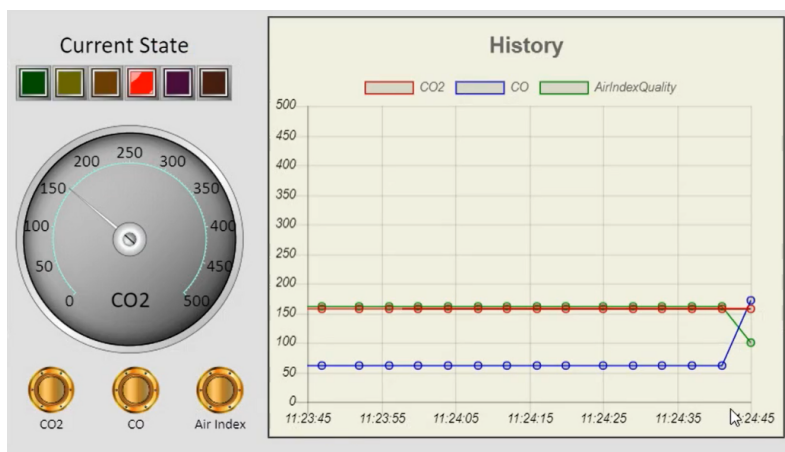
```
s = (s * 500) / 308;
```

سپس عدد هر گاز روی speedometer معادل آن نمایش داده می‌شود و تغییرات لحظه‌ای گازها نیز روی نمودار با سه خط رنگی ثبت می‌شود. همچنین ماکزیمم مقدار گازهای به عنوان معیار اختار در نظر گرفته می‌شود و یکی از ۶ مربع رنگی که در صفحه موجود است بر رنگ می‌شود.

۳.۲ ایلکیشن گوشتی

و دیگر اجزای پنل کاربری استفاده کردیم. سپس آپدیت مقادیر این اجزا را در کد انجام دادیم. قابلیت خوب `iotbuilder` این

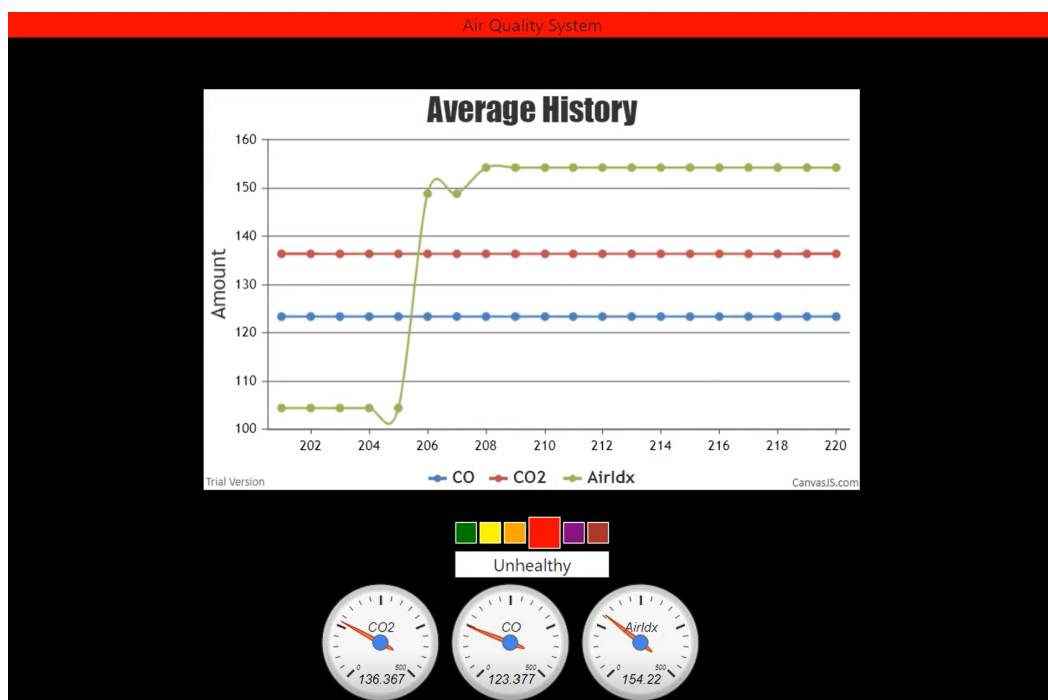
بود که پس از قرار دادن این کنترلرها در پنل گرافیکی به طور اتوماتیک المنتی معادل آن در کد ساخته می‌شد و بنابراین روند کد زدن را خیلی راحت می‌کرد. در زیر شمای پنل کاربری را مشاهده می‌کنید.



شکل ۳: پنل گرافیکی

۴.۲ وب‌سرور و میانگین‌گیری از نتایج چند دستگاه

برای پیاده‌سازی کار با چند لوکیشن، ما وب‌سروری طراحی کردیم که اطلاعات گازها را از دستگاه‌های مختلف می‌گیرد و میانگین آن‌ها را در نمودار نمایش می‌دهد. این وب‌سرور با یک ایندکس به نام `iot` که در `elasticsearch` ساخته شده کار می‌کند و هر یک از دستگاه‌ها مقادیرشان را در ایندکس پوش می‌کنند و وب‌سرور به صورت دوره‌ای این مقادیر را می‌خواند و از روی میانگین آن‌ها نمودار می‌کشد. همچنین مشابه قبل اظهارهای مربوط به وضعیت هوا را با استفاده از مقدار ماکزیمم یک گاز نمایش می‌دهیم. در زیر شمای بخش میانگین‌گیری وب‌سرور را مشاهده می‌کنید.



شکل ۴: وب‌سرور

۳ فایل‌ها و شیوه‌ی اجرای برنامه

برای اجرای برنامه مراحل زیر را طی می‌کنیم:

- **راه‌اندازی سرور الاستیک:** برای راه‌اندازی وب سرور باید elasticsearch روی سیستم نصب باشد. ورژن‌های مختلف نرم‌افزار در اینجا موجود است. پس از دانلود و unzip کردن وارد پوشه می‌شویم فایل elasticsearch.yml را با فایلی که پوشه‌ی config از پروژه قرار دارد جایگزین می‌کنیم. این فایل قابلیت cors را به سرور اضافه می‌کند.

سپس دستور

```
elasticsearch-7.8.0> .\bin\elasticsearch
```

را می‌زنیم تا سرور راه بیفتد. در نهایت با دستور زیر فایل index.py را ران می‌کنیم.

```
python elasticsearch\index.py
```

تا یک ایندکس خالی به نام iot در سرور الاستیک ساخته شود (روی یک سیستم فقط یک بار نیاز است این دستور اجرا شود).

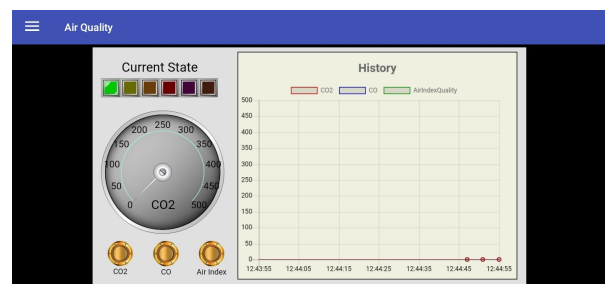
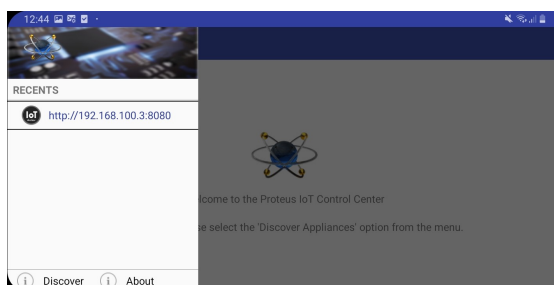
- **فایل‌های Arduino Yun:** پس از باز کردن پروژه در پروتئوس پوشه‌ای با نامی مشابه زیر در آدرسی مشابه زیر ساخته می‌شود:

```
Users\USERNAME\AppData\Local\Temp\VSM Studio\RANDOM_NUMBER
```

در این پوشه یک پوشه‌ی دیگر به نام ARDUINO YUN قرار دارد که محتوای آن باید با محتوای این پوشه که ما در فایل‌های پروژه قرار دادیم جایگزین شود.

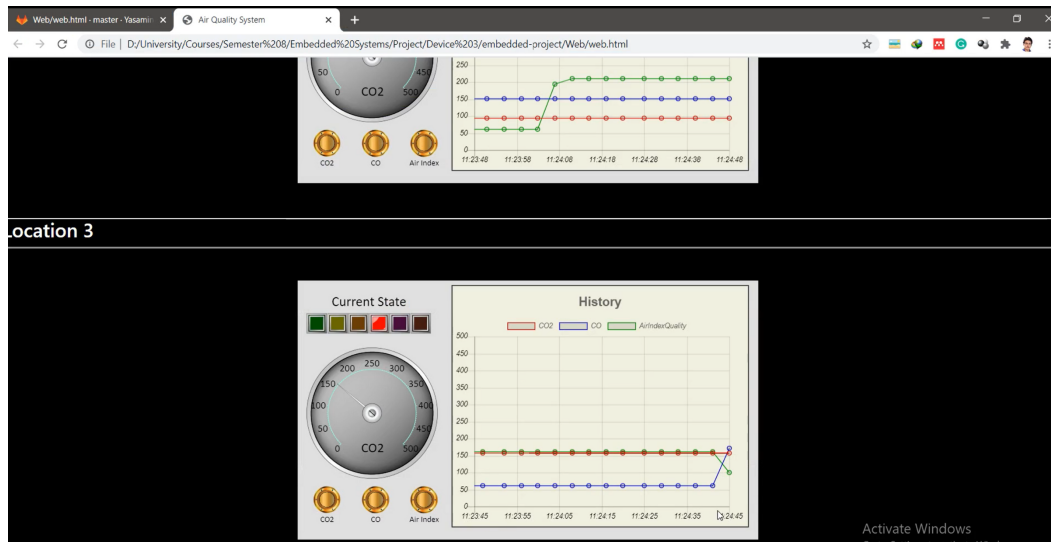
در این پیاده‌سازی هر یک از device ها روی یک پورت از localhost اجرا می‌شوند.

- **راه‌اندازی کلاینت گوشی همراه:** برای راه‌اندازی کلاینت گوشی همراه باید نرم‌افزار IoT Controller را از اینجا نصب کنیم. سپس به صورت زیر در بخش Discover آی پی و پورتی که برد آردینو روی آن در حال اجرا است را وارد کنیم تا گوشی به برد وصل شود.



شکل ۵: شیوه‌ی اتصال گوشی به برد

- **مشاهده نتایج در وب سرور:** برای مشاهده‌ی نتایج در وب سرور فایل web.html در پوشه‌ی Web را اجرا می‌کنیم. ابتدا میانگین وضعیت کلی دستگاه‌های متصل به سرور و سپس وضعیت هر لوکیشن (هر یک از دستگاه‌ها) به ترتیب مشاهده می‌شود.



شکل ۶: لوکیشن‌های مختلف به صورت مجزا

۴ سیمولیشن و نتایج

ویدیوی کاملی از تست سیستم با چند لوکیشن در فایل‌های پروژه به نام test-video.mp4 آورده شده است.

۵ چالش‌ها

مشکل اصلی‌ای که در ابتدای پیاده‌سازی با آن مواجه شدیم این بود که محیط شبیه‌سازی مناسب برای کاربردهای IoT که با بردهای آردوینو کار کند بسیار کم بود و خود IoTBuilder هم با وجود امکانات زیادش بسته به سیستم ممکن است مشکلاتی را در اجرا ایجاد کند. پیاده‌سازی با برد واقعی بسیاری از این مشکلات را حل می‌کرد.

۶ منابع

لیستی از منابعی که در این پروژه از آن‌ها استفاده کردیم در زیر آورده شده است.

- https://create.arduino.cc/projecthub/vishnutheerth_e_p/iot-based-simple-air-pollution-monitoring-system-ce61ad
- <https://labcenter.s3.amazonaws.com/downloads/IoTHelp.pdf>
- <https://create.arduino.cc/projecthub/east-west-university/indoor-air-quality-monitoring-system-5b5244>
- https://create.arduino.cc/projecthub/vishnutheerth_e_p/iot-based-simple-air-pollution-monitoring-system-ce61ad
- <https://github.com/Circuito-io/Red-Balloon-Air-Pollution-Detector>