

به نام خدا



دانشگاه صنعتی شریف

دانشکده مهندسی برق

سیستم‌های نهفته بی‌درنگ

پروژه پایانی

دستیاران آموزشی : نیما صمدی، علی قنبریان

استاد درس : دکتر ایمان غلامپور

نیم‌سال ۲-۱۴۰۳

برای پروژه درس سیستم‌های نهفته بی‌درنگ ۲ موضوع زیر در نظر گرفته شده است که می‌توانید به دلخواه یک مورد را انتخاب کرده و پیاده‌سازی کنید:

۱. کار با اعداد ممیز ثابت

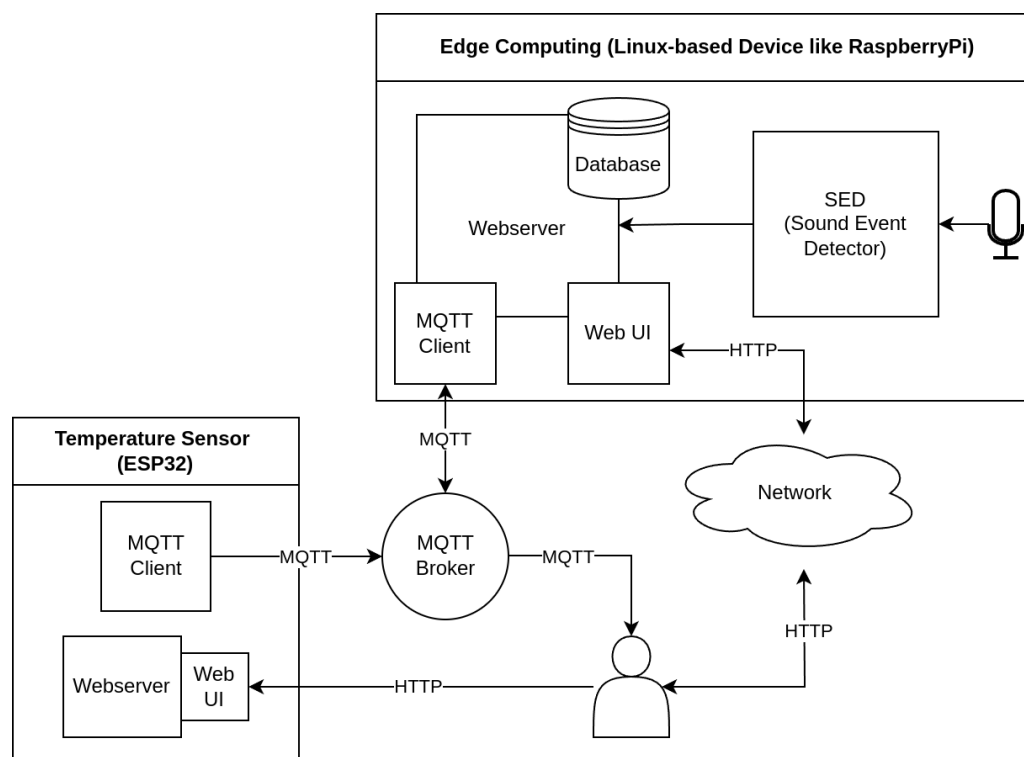
۲. تشخیص رویداد صوتی و کار با ESP32

## ۱ کار با اعداد ممیز ثابت

این پروژه توسط استاد درس طراحی شده است و دانشجویان علاقه‌مند برای انجام آن به ایشان پیام داده و جزئیات و نیازمندی‌های لازم را جویا شوند.

## ۲ تشخیص رویدادهای صوتی و کار با ESP32

در این پروژه، هدفمان تشخیص رویدادهای صوتی با استفاده از مدل‌های آماده شبکه عصبی در یک دیوایس لینوکسی و ایجاد رابط کاربری مناسب با کاربر است. همچنین کار با دیوایس ESP32 و ارتباط با پروتکل MQTT هدف دیگر این پروژه است. در حقیقت در این پروژه، قصد داریم یک پلتفرم اینترنت اشیاء را با تمرکز ویژه بر سیستم‌های نهفته آن بالا بیاوریم. ساختار ماژولار پروژه به صورت زیر می‌باشد:



شکل ۱: ساختار پروژه

در این پلتفرم، دو دستگاه embedded وجود دارد:

۱. دستگاه مبتنی بر سیستم عامل لینوکس. این دستگاه، قرار است به صورت دائم و از طریق میکروفونی که به آن متصل است، صداهای محیط را پردازش کند و در صورت شناسایی رخداد ویژه‌ای در محیط، آن را گزارش کند. در همین دستگاه، یک WebServer وجود دارد که به ماژول تشخیص رخداد صوتی متصل است و رخدادهای صوتی را در یک Database ذخیره

می‌کند. همچنین اطلاعات کلی وضعیت این دیوایس از جمله مصرف رم و لود پردازنده در پایگاه داده ذخیره می‌شود.

برای ارتباط وب سرور دستگاه با کاربر، دو ماژول وجود دارد. ماژول اول، یک رابط کاربری تحت وب می‌باشد که کاربر از طریق پروتکل HTTP می‌تواند محتوای آن را مشاهده کند. در این ماژول، باید رخدادهای صوتی تشخیص داده شده به کاربر نشان داده شوند. ماژول دوم، یک کلاینت MQTT است که در صورت شناسایی یک رویداد، به کاربر از طریق پروتکل MQTT هشدار می‌دهد. همچنین این کلاینت، داده دمایی از سمت سنسور دما را دریافت می‌کند. در صورتیکه این دما از حد مشخصی بیشتر شود، از طریق پروتکل MQTT به کاربر هشدار می‌دهد.

۲. سنسور دما، که قرار است دمای محیط را به دستگاه لینوکسی از طریق پروتکل MQTT ارسال کند. در این بخش، از میکروکنترلر ESP استفاده می‌کنیم. دوره تناوب ارسال داده دمایی، باید توسط کاربر تعیین شود. به این منظور یک وب سرور ساده روی ESP باید بالا بیاید تا کاربر با اتصال به آن از طریق پروتکل HTTP، دوره تناوب را مشخص کند.

در ادامه جزئیات هر ماژول برای پیاده‌سازی را توضیح می‌دهیم.

## ۱.۲ تشخیص صدا

این ماژول باید به میکروفون متصل شود و دائماً در حال ضبط صدا و پردازش آن باشد. در صورتی که رویدادی در صدای ضبط شده وجود داشت، ۱۰ ثانیه از آن صدا ذخیره شده و از طریق وب سرور، به همراه زمان ثبت آن رویداد در دیتابیس ذخیره می‌شود. برای پیاده‌سازی این ماژول از زبان پایتون استفاده کنید. شما می‌توانید از شبکه‌های موجود و از قبل آموزش داده‌شده استفاده کنید. پیشنهاد ما استفاده از Yamnet است. برای اطلاعات بیشتر در مورد رویدادهای صوتی و انواع آنها، به [این لینک](#) مراجعه کنید. فرایند تشخیص رویداد صوتی نباید زمان زیادی بگیرد. زمان مطلوب برای اجرای مدل تشخیص رویداد حداکثر ۱ ثانیه است. همچنین زمان سر به سر (end-to-end) از لحظه‌ای که رویداد صوتی اتفاق می‌افتد تا زمانی که کاربر تشخیص آن را در رابط کاربری مشاهده می‌کند باید حداکثر ۲ ثانیه باشد. این زمان را اندازه‌گیری کنید و در گزارش خود ذکر کنید. سعی کنید اندازه‌گیری تا حد ممکن دقیق باشد.

## ۲.۲ وب سرور

وب سرور وظیفه‌ی مدیریت دسترسی کاربر از طریق اینترنت و نمایش اطلاعات به او را بر عهده دارد. این ماژول پس از دریافت تشخیص یک رویداد صوتی توسط ماژول تشخیص صدا، رویداد تشخیص داده شده و همچنین صدای ضبط شده را در پایگاه داده ذخیره می‌کند. وب سرور با ارسال دستورات SQL به پایگاه داده، از آن خوانده و یا در آن می‌نویسد.

این ماژول همچنین اطلاعات لازم را از پایگاه داده گرفته و از طریق رابط کاربری و اینترنت به کاربر نشان می‌دهد. وب سرور همچنین باید اطلاعات وضعیت دستگاه لینوکسی را در دیتابیس ذخیره کند تا به درخواست کاربر برای مشاهده‌ی آنها پاسخ دهد. پیشنهاد می‌شود اطلاعات وضعیت سرور در جدول جداگانه‌ای در پایگاه داده ذخیره شود. بار پردازنده، دمای پردازنده و میزان مصرف رم چند مورد از متریک‌هایی هستند که باید در پایگاه داده ذخیره شوند.

پیاده‌سازی این ماژول باید با استفاده از زبان C یا C++ صورت پذیرد. برای پیاده‌سازی وب سرور از کتابخانه‌ی Beast در Boost استفاده کنید. در کنار پروژه آموزش لازم برای استفاده از این کتابخانه در اختیار شما قرار خواهد گرفت. توجه کنید که برای بیلد پروژه از CMake استفاده کنید و CMakeLists خود را مطابق ساختار استاندارد بنویسید. توجه کنید وب سرور پیاده‌سازی شده باید امکان ارسال فایل و مدیریت کردن ریکوئست‌های GET را داشته باشد.

پس از پیاده‌سازی کد مربوط به وب سرور، آنرا به صورت یک سرویس در لینوکس اجرا کنید که با بالا آمدن سیستم اجرا شده و در صورت kill شدن از نو اجرا می‌شود.

## ۳.۲ پایگاه داده

پایگاه داده وظیفه‌ی جمع‌آوری و ذخیره‌ی داده‌هایی را که از طرف وب‌سرور ارسال می‌شود، بر عهده دارد. با دریافت درخواست از طرف وب‌سرور، جدول مورد نظر و سطرهای آن خوانده شده و برای وب‌سرور ارسال می‌شود تا به کاربر نمایش داده شوند.

برای پایگاه داده می‌توانید از MariaDB که به پیاده‌سازی مبتنی بر MySQL است، استفاده کنید. پیشنهاد می‌شود دو جدول در پایگاه داده ایجاد کنید که یکی برای ذخیره رویدادهای مربوط به صدا و زمان تشخیص آنها و دیگری برای ذخیره اطلاعات کلی وضعیت سرور استفاده شود. همچنین یک جدول برای ذخیره‌ی دمای اندازه‌گیری شده توسط سنسور ESP32 ایجاد کنید.

## ۴.۲ رابط کاربری وب

برای تسهیل ارتباط کاربر با دیوایس‌ها، برای وب‌سرورها یک رابط کاربری ساده طراحی کنید. رابط کاربری وب سرور دیوایس لینوکسی باید امکان نمایش تاریخچه‌ی رویدادهای ثبت شده توسط ماژول تشخیص رویداد صوتی را داشته باشد. همچنین با درخواست کاربر، اطلاعات کلی وضعیت سرور که پیشتر در پایگاه داده ذخیره شده است به او نمایش داده شود.

برای طراحی رابط کاربری نیازی به استفاده از فریمورک‌های پیشرفته‌ی Frontend نیست اما پیشنهاد می‌شود از Bootstrap برای زیبایی رابط کاربری استفاده کنید.

## ۵.۲ بروکر MQTT

MQTT یکی از پرکاربردترین پروتکل‌های مورد استفاده در اینترنت اشیا است. ساختار این پروتکل مبتنی بر معماری Publish-Subscribe است؛ به این صورت که یک واسط تحت عنوان broker وجود دارد که پیام‌ها را بین کاربرانی که به آن متصل شده‌اند جابجا می‌کند. هرکس بخواهد پیامی را دریافت کند، در یک topic مشخص subscribe می‌کند و هر کس بخواهد پیامی ارسال کند، در همان topic پیام خود را publish می‌کند.

در ساختار ماژولار ذکر شده در تصویر ۱، بروکر MQTT در یک دیوایس امبدد دیگر قرار دارد تا بتواند ارتباط به دیوایس‌های مختلف را مدیریت کند. اگر دیوایس دیگری در اختیار ندارید، می‌توانید بروکر را روی سیستم لینوکس (مانند رزبری پای یا لپتاپ خود) بالا بیاورید. بروکر را روی پورت ۱۸۸۳ بالا بیاورید و یک سرویس بنویسید که در هنگام بالا آمدن سیستم، آن را راه‌اندازی کند. صحت عملکرد بروکر و بالا بودن آن را به صورت محلی و روی همان سیستم لینوکسی تست کنید. در مرحله بعد، از طریق یک دستگاه خارجی مثل گوشی موبایل، به همان شبکه وای‌فای که سیستم لینوکس‌تان وصل است، متصل شوید. برای اطمینان از صحت اتصال، روی تاپیک project/test توسط سیستم لینوکسی خود subscribe کنید. سپس روی همین تاپیک، پیامی را توسط دستگاه خارجی publish کنید و روی سیستم لینوکسی خود آن را دریافت کنید.

## ۶.۲ کار با ESP32

### ۱.۶.۲ معرفی

ESP32 یک میکروکنترلر کاربردی با قابلیت اتصال به WiFi و بلوتوث است که توسط شرکت Espressif تولید می‌شود. این میکروکنترلر مبتنی بر معماری RISC-V و دارای دو هسته‌ی پردازشی با فرکانس ۲۴۰ مگاهرتز می‌باشد. به جهت استفاده‌های آموزشی، ماژول‌های متعددی برای این میکروکنترلر توسعه داده شده که از آن جمله می‌توان به ماژول NodeMCU اشاره کرد. برای برنامه‌نویسی ESP32 دو framework وجود دارد:

۱. ESP-IDF (Espressif IoT Development Framework): این فریم‌ورک توسط شرکت سازنده ESP توسعه

داده شده است و دسترسی مستقیم به ویژگی‌ها و قابلیت‌های ESP32 را در سطوح پایین سخت‌افزاری فراهم می‌کند و به

همین جهت برای توسعه کدهای حرفه‌ای و بهینه مناسب است.

۲. Arduino: فریم‌ورکی راحت و کاربردی. این فریم‌ورک به جهت سادگی و سازگاری با دستورات Arduino استفاده زیادی دارد؛ هرچند که نسبت به ESP-IDF دسترسی‌های سخت‌افزاری کمتری را به توسعه دهنده می‌دهد.

از جمله مهم‌ترین محیط‌هایی که برای برنامه‌نویسی ESP32 استفاده می‌شوند، می‌توان به Arduino-IDE و افزونه PlatformIO در محیط VSCode اشاره کرد؛ که PlatformIO به جهت مدیریت فایل‌های جداگانه و کتابخانه‌های مختلف، محیطی حرفه‌ای‌تر به حساب می‌آید. توصیه اکید ما این است که در این پروژه از PlatformIO استفاده کنید. برای انتخاب framework محدودیتی وجود ندارد.

## ۲.۶.۲ استفاده از ESP32 به عنوان سنسور

در این بخش می‌خواهیم کاربرد ESP32 را به عنوان یک دستگاه در پلتفرم اینترنت اشیا تجربه کنیم.

۱. فرض کنید سنسور دمایی ساختیم که قرار است به صورت متناوب، در بازه‌های زمانی معینی دمای محیط را اندازه بگیرد و به سرور بفرستد. باتوجه به عدم دسترسی اکثر دانشجویان به سنسور دما، دمای درونی تراشه را مبنا قرار می‌دهیم. یک برنامه ESP32 بنویسید که در بازه‌های زمانی ۲۰ ثانیه‌ای، دمای تراشه را بخواند. سپس با اتصال به شبکه مودم Wi-Fi محلی و با داشتن IP سیستم لینوکسی که بروکر روی آن فعال است، به بروکر متصل شده و دمای تراشه را روی تاپیک device/temperature publish کند.

نرخ گزارش دما باید قابل تنظیم باشد. بدین منظور، مطابق شکل ۱، یک وب سرور سبک با استفاده از فریم‌ورک‌های برنامه‌نویسی ESP32 پیاده‌سازی کنید که امکان دریافت نرخ جدید را از کاربر فراهم سازد. همچنین یک رابط کاربری ساده که از طریق اینترنت قابل دسترسی باشد و همین وب سرور آنرا به کاربر نمایش دهد بنویسید.

۲. روی دیوایس لینوکسی که ماژول تشخیص صدا و وب سرور اصلی را پیاده‌سازی کرده‌اید، یک سرویس بنویسید که به بروکر متصل شده، روی تاپیک device/temperature سابسکرایب کرده و پس از خواندن دمای ارسال شده از سمت دستگاه، آن را در جدول temperature در دیتابیس ذخیره کند.

## نکات تحویل پروژه

- در فایل تحویلی باید کد، گزارش و نتایج موجود باشند.
- از برخی از دانشجویان تحویل آنلاین صورت خواهد گرفت و دانشجو باید بتواند کد و نتایج خود را باز تولید و توضیح دهد.
- فایل گزارش، کد پروژه و توضیحات موردنیاز را فشرده کرده و به فرمت emb-proj-#STUDENTID.zip تحویل دهید.