

بسمه تعالی

امنیت در اینترنت اشیا

تمرین سری چهارم



نیم سال پاییز ۱۴۰۴

مقدمه

ESP32 یک میکروکنترلر محبوب مبتنی بر معماری RISC-V است که توسط شرکت چینی Espressif Systems ساخته شده است. پردازنده دو هسته‌ای، قابلیت برقرای ارتباط از طریق Wi-Fi و بلوتوث، تعداد پین‌های GPIO متعدد و همچنین قیمت مناسب این میکروکنترلر، آن را برای طیف گسترده‌ای از پروژه‌های IoT ایده‌آل می‌کند.

به جهت استفاده‌های آموزشی، ماژول‌های متعددی برای این میکروکنترلر توسعه داده شده، که از جمله آن‌ها می‌توان به ماژول‌های NodeMCU و ESP-WROOM-32 اشاره کرد.

برنامه‌نویسی ESP32 پیچیدگی زیادی ندارد و می‌توانید با جستجو در اینترنت و آموزش‌های متعددی که در این زمینه وجود دارد، کار خود را شروع کنید. به صورت کلی، دو framework مشهور برای برنامه‌نویسی ESP32 وجود دارد:

- **ESP-IDF (Espressif IoT Development Framework)**: این فریم‌ورک توسط شرکت سازنده ESP توسعه داده شده است. ESP-IDF دسترسی مستقیم به ویژگی‌ها و قابلیت‌های ESP32 را در سطوح پایین سخت‌افزاری فراهم می‌کند و به همین جهت، برای توسعه کدهای بهینه مناسب‌تر است.
- **Arduino**: فریم‌ورکی راحت و ابتدایی. 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.

صورت تمرین

کنترل و وضعیت LED از طریق بلوتوث کم مصرف (BLE) با استفاده از ESP32

هدف:

در این تمرین قصد داریم با نحوه‌ی برقراری ارتباط **Bluetooth Low Energy (BLE)** بین ماژول **ESP32** و تلفن همراه آشنا شویم.

هدف نهایی این است که بتوان از طریق موبایل با استفاده از برنامه‌ی **nRF Connect**، LED روی ماژول را روشن، خاموش یا در حالت چشمک‌زن قرار داد.

شرح تمرین:

۱. با استفاده از ماژول **NodeMCU-32 (ESP32)**، یک **BLE Server** پیاده‌سازی کنید.
۲. این سرور باید شامل یک **Service** و سه **Characteristic** باشد:
 - کاراکتر ۱: کنترل روشن شدن LED (ارسال مقدار 1)
 - کاراکتر ۲: کنترل خاموش شدن LED (ارسال مقدار 0)
 - کاراکتر ۳: فعال‌سازی حالت چشمک‌زن LED (ارسال مقدار 2)
۳. وقتی مقدار 2 دریافت شد، LED باید به صورت خودکار هر ۲ ثانیه یک‌بار چشمک بزند.
۴. وضعیت فعلی LED (روشن، خاموش، چشمک‌زن) باید در **Serial Monitor** چاپ شود.
- از طریق نرم‌افزار **nRF Connect** روی تلفن همراه، با **characteristic** ها ارتباط برقرار کنید.
۵. مقدار **RSSI** (قدرت سیگنال) در کنسول نمایش دهید.

نکات فنی:

- ارتباط باید از طریق **Characteristic** با خاصیت **writable** انجام شود.
- LED باید بلافاصله پس از تغییر **characteristic**، وضعیت جدید را بگیرد.

- در حالت چشمک‌زن، تا زمانی که دستور جدیدی (0 یا 1) نرسد، LED باید به چشمک‌زدن ادامه دهد.
- نمایش مقدار RSSI (قدرت سیگنال) در Serial Monitor

گزارش تمرین:

در گزارش خود موارد زیر را بیاورید:

۱. تصویر از اتصال موفق در کنسول و nRF Connect و ارسال مقادیر
۲. نمونه خروجی Serial Monitor برای هر حالت (روشن، خاموش، چشمک‌زن)

نکات قابل توجه:

استفاده از افزونه PlatformIO بالامانع است. البته توجه داشته باشید که به جهت محدودیت های فریم ورک Arduino ، شما مجاز به استفاده از این فریم ورک نیستید و باید از ESP-IDF استفاده کنید. فایل مربوط به کدهای ESP و همچنین گزارش تمرین را در یک فایل زیپ و با فرمت نامگذاری زیر در CW آپلود کنید:

SIOT_HW4_StudentID

گزارش شما شامل توضیحات لازم و تصاویر مربوطه و یک ویدئو که در آن عملکرد بورد خود را توضیح می دهید می باشد. توجه کنید که طول ویدئو از ۱ دقیقه و حجم آن از ۳۰ مگابایت بیشتر نشود، در غیر این صورت، تمرین شما با کسر نمره تصحیح خواهد شد.

موفق باشید