UNIVERSITY OF MISSOURI KANSAS CITY

Mini-project #3: Data Exchange Using Bluetooth Low Energy

Objectives

- Set up a BLE device using ESP32 to transmit temperature and humidity data.
- Broadcast BLE services and allow client connections for real-time updates.
- Ensure proper connection and disconnection handling.
- Implement a battery level simulation.
- **(Optional)** Enable users to set a threshold for filtering notifications for extra credit.

Project Requirements (1/3)

1. Device Naming:
   - The ESP32 device must be named in the format "Team X", where X represents the team number.

2. BLE UUID Setup:
   - Implement a BLE service using the Environmental Sensing Service UUID (0x181A).
   - Set up two characteristics: temperature (UUID: 0x2A6E) and humidity (UUID: 0x2A6F).
   - Both characteristics must support the Notification property, allowing connected clients to receive updates.

Project Requirements (2/3)

3. Handling Connections:
   - Properly manage device connections and disconnections, ensuring that the device is connectable and disconnects correctly.
   - Use BLE debugging apps (e.g., LightBlue or nRF Connect) to verify the BLE services and characteristics.

4. Battery Level Characteristic:
   - Implement a Battery Service (UUID: 0x180F) with a Battery Level Characteristic (UUID: 0x2A19). Simulate a battery draining from 100% to 0% at a rate of 1% per minute, notifying connected clients of the changes.

Project Requirements (3/3)

5. (Optional for extra credit: 3 points) Humidity Threshold Filtering:

- Implement a feature allowing users to set a specific humidity threshold (e.g., greater than 40%) within the Bluetooth apps. Once configured, these settings should prompt the apps to relay only those data packets to the ESP32 that contain humidity values exceeding the established threshold.

Debug Guide

- Use a BLE debugging app to find the ESP32 device and verify that the services and characteristics (temperature and humidity) are correctly implemented.

- Recommended apps:
  - For iOS: LightBlue or nRF Connect
  - For Android: nRF Connect or BLE Scanner

- Ensure that notifications are properly handled, with updates delivered at reasonable intervals (1 to 10 seconds).

Submissions

- Source Code: Provide well-documented code (Arduino code *.ino) with comments explaining each part of the code. **(Deadline: 11: 59 PM on March 16, 2025)**

- Project Report: A concise report describing the project's implementation and any challenges faced. **(Deadline: 11: 59 PM on March 16, 2025)**

- In-class Demo: Showcase the working project in real-time. **(Deadline: 5:30 PM – 6:45 PM on March 17, 2025)**

Evaluation Criteria

- The device must broadcast the correct name, service, characteristics (temperature and humidity), and battery level simulation with proper notification intervals.

- **(Optional for extra credit)** Implement functionality that allows users to set a threshold for filtering notifications.

- Code quality (readability, comments, and structure).

- Clarity and thoroughness of the project report and demonstration.