ECE 558 IoT MQTT Project
ESP32 Version

Atharva Lele - PSU ID 909998277

Project Description:

- I've used the Adafruit HUZZAH32 board, and HiveMQ as the MQTT Broker
- The board's MQTT client was written in C++
    - Used PlatformIO to create the project for it
    - Used the MQTT Library linked in the Project Assignment (https://github.com/plapointe6/EspMQTTClient)
- Changes from the original circuit:
    - Used an RGB LED instead of just a single colored LED
    - Used PWM to control LED brightness, controllable from the Android app
    - Configured the button GPIO pin as an input with internal pulldown (instead of using an actual resistor on the breadboard)

Theory of Operation:

- **Android:**
    - Connect to the internet and open the app
    - The app will quit if it detects that internet access is unavailable. A toast message will be displayed to inform the user.
    - Once the app connects to the MQTT server, the app subscribes to various topics and then waits for messages.
    - On receiving a message, it will update data on the screen depending on which message is received
    - The app also has sliders to control RGB values of the LED, as well as a slider to control the sensor's read interval
        - On change of the slider value, an MQTT message is published to notify the ESP32 of the updated values

- **ESP32:**
    - Once powered, the ESP32 will try to connect to the wifi network that was specified in the code (RGB LED blinking purple)
    - Once connected to WiFi (LED turning blue), it will attempt to connect to the HiveMQ MQTT Broker. LED will turn green when it connects successfully
    - Once connected, it subscribes to MQTT topics for the sensor read interval and the RGB brightness values. The device also sends a "I'm connected" status message.
    - Every sensor read interval, the ESP32 samples and publishes temperature and humidity to their respective topics
    - The onboard LED blinks every time a message is published to the MQTT broker
    - Every time the button is pressed / released, the ESP32 publishes this to the button topic
    - The code has been split into modules of button, sensor, and MQTT so that everything is independent and modular.

Github Repo (Github classroom): https://github.com/ECE558-winter2022/proj2-atharvalele
Github Repo (personal): https://github.com/atharvalele/ece558_proj2_mqtt
(I've mirrored my commits to the Github classroom repo, keeping this link here in case of any issues)