**Aplicação com comunicação sem fio para IoT**

Nome: **José Adriano Filho**

Matrícula: **2025101109806**

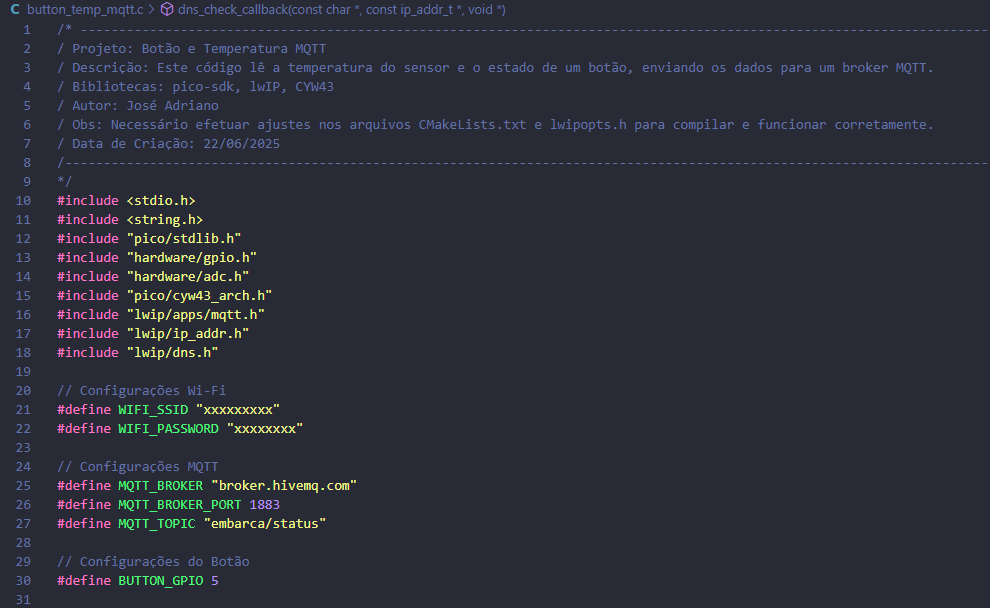
Unidade\_3 - Capítulo\_1

1. Leitura de Luminosidade com o Sensor BH1750: configurar o sensor BH1750 para medir a intensidade de luz, utilizando o RP2040. A programação do microcontrolador será feita para coletar os dados de luminosidade e exibi-los diretamente em um monitor serial, permitindo a observação em tempo real das variações na intensidade da luz. Além disso, como um desafio extra, programe o RP2040 para ajustar automaticamente a posição de um Servo Motor 9G SG90 em resposta ao nível de luz captado pelo sensor, de forma que o servo reaja dinamicamente conforme a iluminação detectada.

**R.: link do código:**

<https://github.com/EngAdriano/Residencia/tree/main/Exercicios/U2C3Tarefas/U2C3T3/button_temp_mqtt>

O código foi desenvolvido baseado em exemplos encontrados na internet já que existe uma dificuldade de documentações sobre MQTT em linguagem C. Procuramos desenvolver o monitoramento do botão e a leitura do sensor de temperatura interno, sendo que a cada um segundo enviamos o status e a medida em graus centigrados para o broker MQTT na plataforma da HiveMQ.



Inicialmente fizemos os includes necessários para a comunicação via WiFi, bem como algumas definições importantes.

2. Monitoramento de Temperatura e Umidade com o Sensor AHT10: configurar o sensor AHT10 para monitorar tanto a temperatura quanto a umidade do ambiente, utilizando o RP2040. A programação do microcontrolador deverá coletar e exibir esses dados em uma tela LCD 320x240, possibilitando uma visualização em tempo real. Como um desafio adicional, ajuste o código para que o LCD exiba um aviso caso a umidade ultrapasse 70% ou a temperatura fique abaixo de 20°C, criando uma interface que alerta o usuário sobre condições específicas do ambiente.

3. GPS e Display de Localização: configurar o módulo GY-NEO6MV2 (GPS) para coletar dados de localização e exibi-los na tela LCD 320x240, conectada ao RP2040. A programação deve mostrar, em tempo real, as coordenadas de latitude e longitude na tela, possibilitando o monitoramento da posição. Como desafio extra, programe o sistema para registrar os dados de localização em um cartão SD conectado ao sistema via SPI IDC, permitindo a criação de um histórico de coordenadas armazenado para consultas posteriores.

4. Controle e Monitoramento de Movimento com MPU6050: configurar o sensor MPU6050 para captar dados de movimento, como aceleração e rotação, e exibir as leituras de inclinação no monitor serial. Além disso, programe o RP2040 para ajustar a posição de um Servo Motor 9G SG90 em função do ângulo de inclinação detectado, promovendo um controle dinâmico de movimento em resposta à orientação do sensor. No desafio extra, você deverá adicionar um alerta visual na tela LCD 320x240 que indique quando o sistema ultrapassa um determinado ângulo de inclinação, criando um sistema de monitoramento visual das mudanças de posição.

5. (**Questão desafio**) - Comunicação LoRa e Alerta de Proximidade: configure o sensor de proximidade VI530X em conjunto com o módulo LoRa 915MHz para detectar a presença de objetos próximos e enviar alertas por meio de comunicação LoRa. A programação no RP2040 deve ativar o envio de um alerta ao detectar a presença de um objeto a menos de 10 cm, transmitindo a informação por LoRa. Como desafio adicional, simule a recepção do alerta em outro dispositivo e registre todas as ocorrências de proximidade em um cartão SD, criando um registro histórico de detecções de proximidade para análise futura.