**Aplicação com comunicação sem fio para IoT**

Nome: **José Adriano Filho**

Matrícula: **2025101109806**

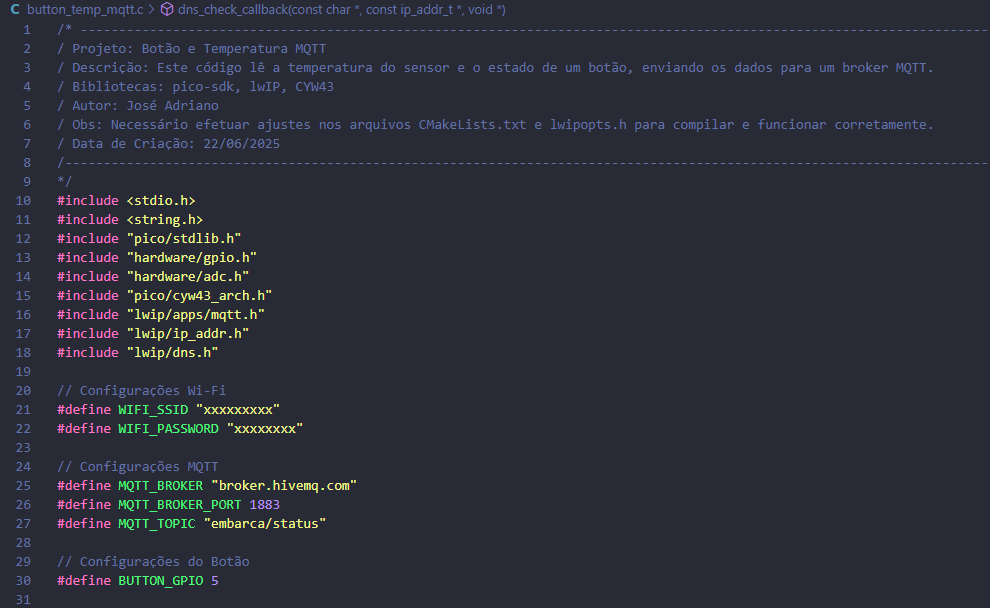
Unidade2-Capítulo3

1. Utilizando a placa Bitdoglab, crie um programa para monitorar o status de um pino, que possa ser definido como entrada, da placa Bitdoglab, e enviar, a cada 1 segundo o status atual para o servidor HiveMQ, utilizando o protocolo MQTT. Além disso, como desafio extra, acrescente algum sensor e envie a informação desse sensor para o servidor.

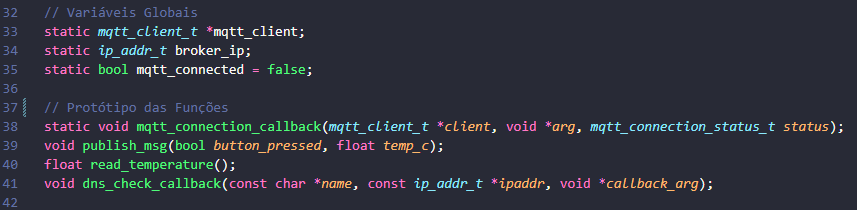
**R.: link do código:**

<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.

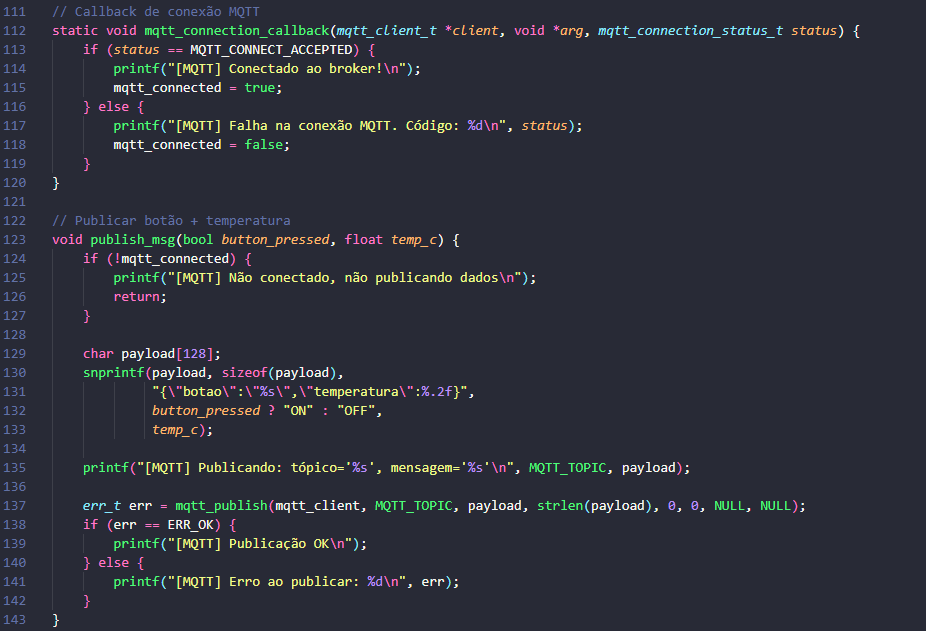


Inicialmente fizemos os includes necessários para a comunicação via WiFi, bem como algumas definições importantes.

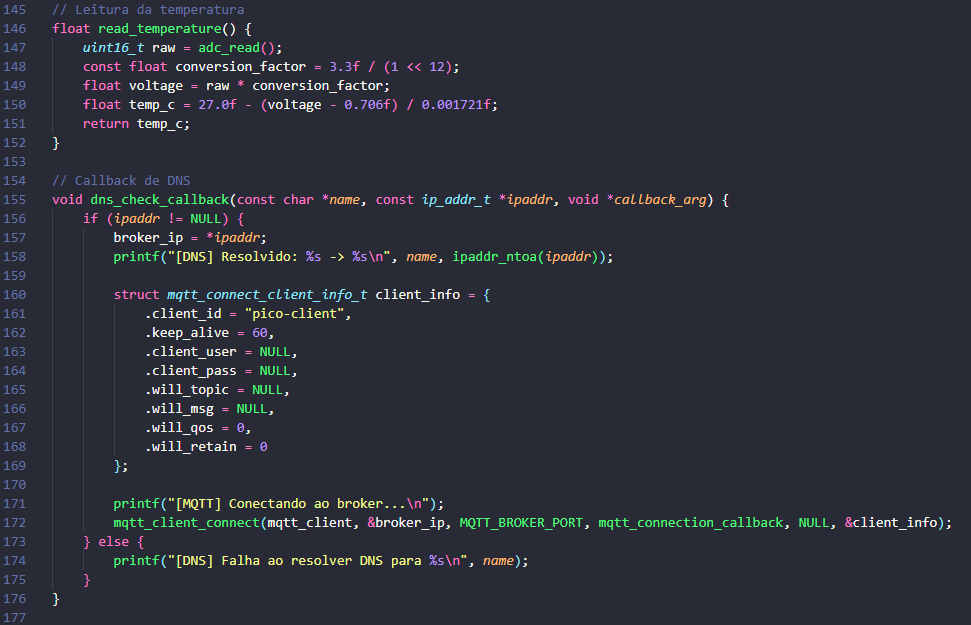


Acima algumas variáveis importantes e os protótipos das funções necessários para executar a leitura do sensor, a resolução do DNS, ou seja, converter o nome para acesso ao broker em um ip fixo, o callback para conexão MQTT e a função de publicar nossa mensagem no tópico.

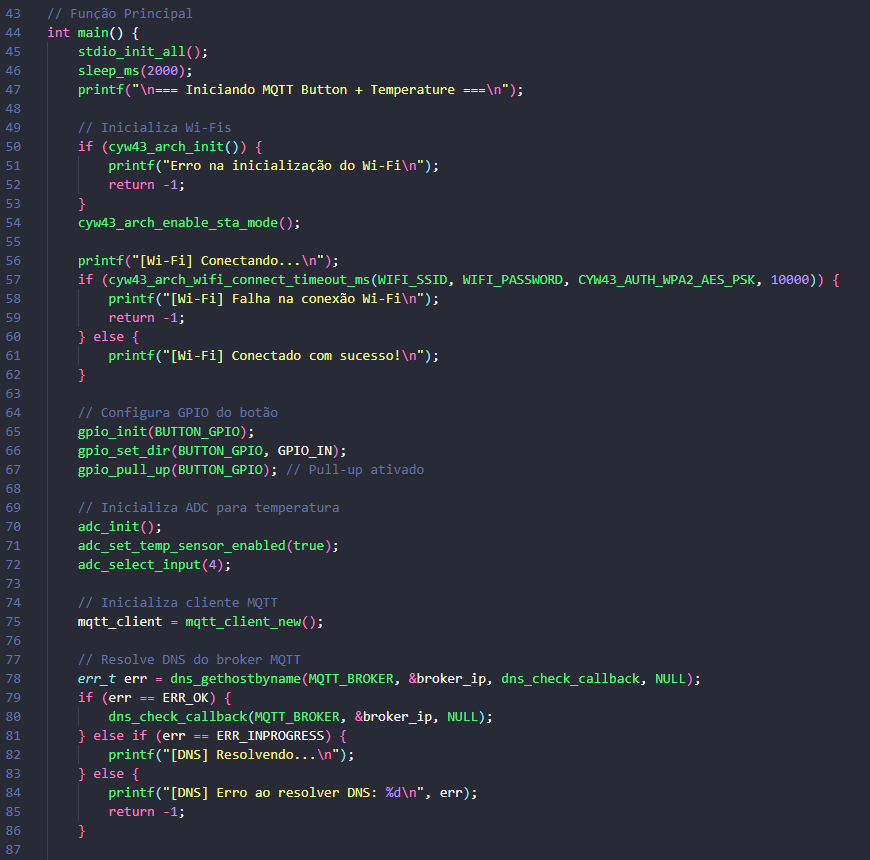
A seguir, de acordo com as assinaturas das funções temos os códigoss das funções para conexão do MQTT e publicar o staus do botão e a temperatura interna.

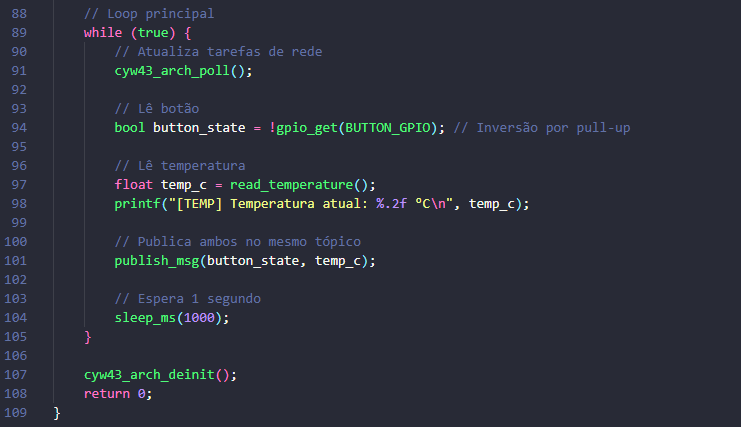


A seguir as outras duas funções, responsáveis pela leitura da temperatura e a resolução do DNS.



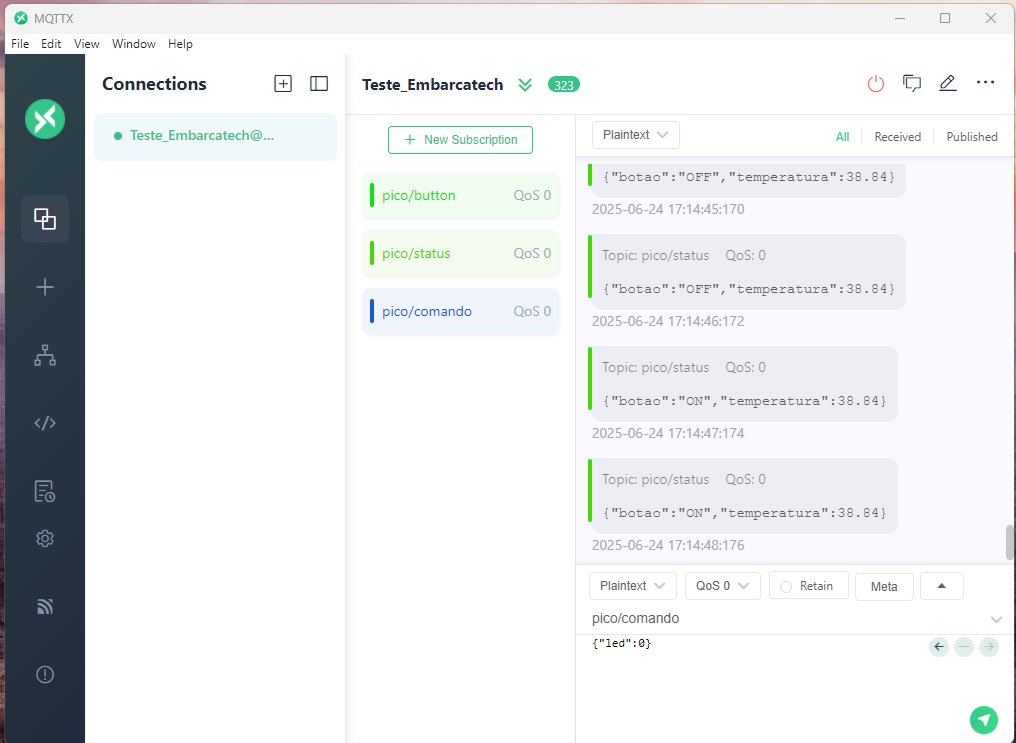
Abaixo o código desenvolvido para a função principal main.



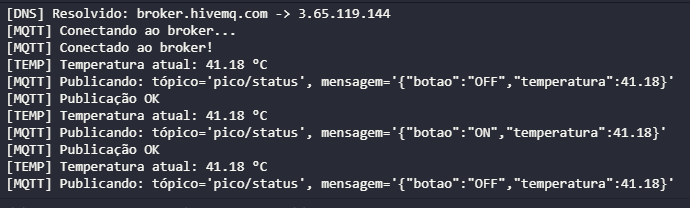


A cada 1 segundo executamos a atualização do status do botão e do valor do sensor de temperatura.

Os testes foram executados com o cliente MQTT “MQTTX” rodando em um pc e inscrito no tópico “pico/status.



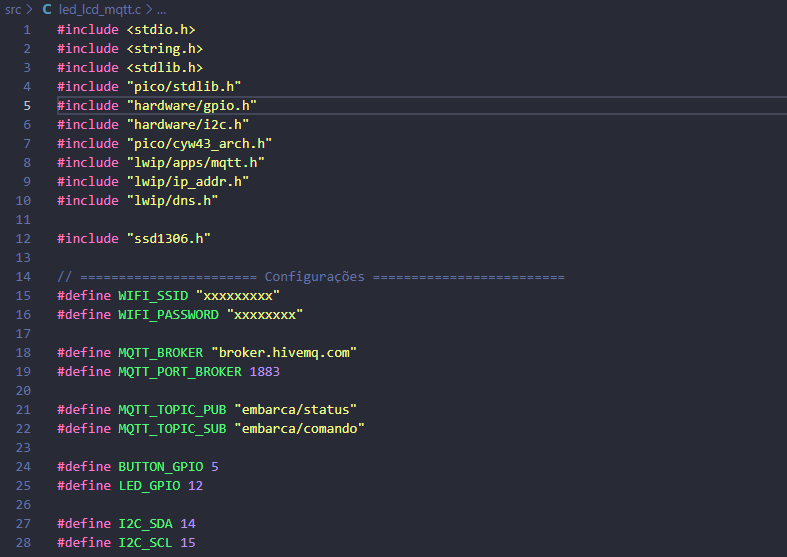
Na saída serial temos alguns “prints” para monitoramento e debug da aplicação.



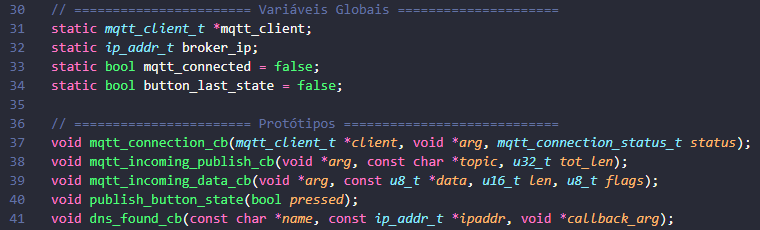
2. Utilizando a placa Bitdoglab, crie um programa para receber dados do servidor HiveMQ, utilizando o protocolo MQTT. Os dados recebidos devem ligar ou desligar um LED, que deve ser ligado a placa. Além disso, como desafio extra, implemente um LCD ou outro tipo de display para monitorar o recebimento de mensagens.

**R.: link do código:** <https://github.com/EngAdriano/Residencia/tree/main/Exercicios/U2C3Tarefas/U2C3T3/led_lcd_mqtt>

Nossa proposta agora é comandar um led por comando enviado pelo protocolo MQTT, bem como sinalizar no display da BigDogLab o status dos acontecimentos durante a execução.

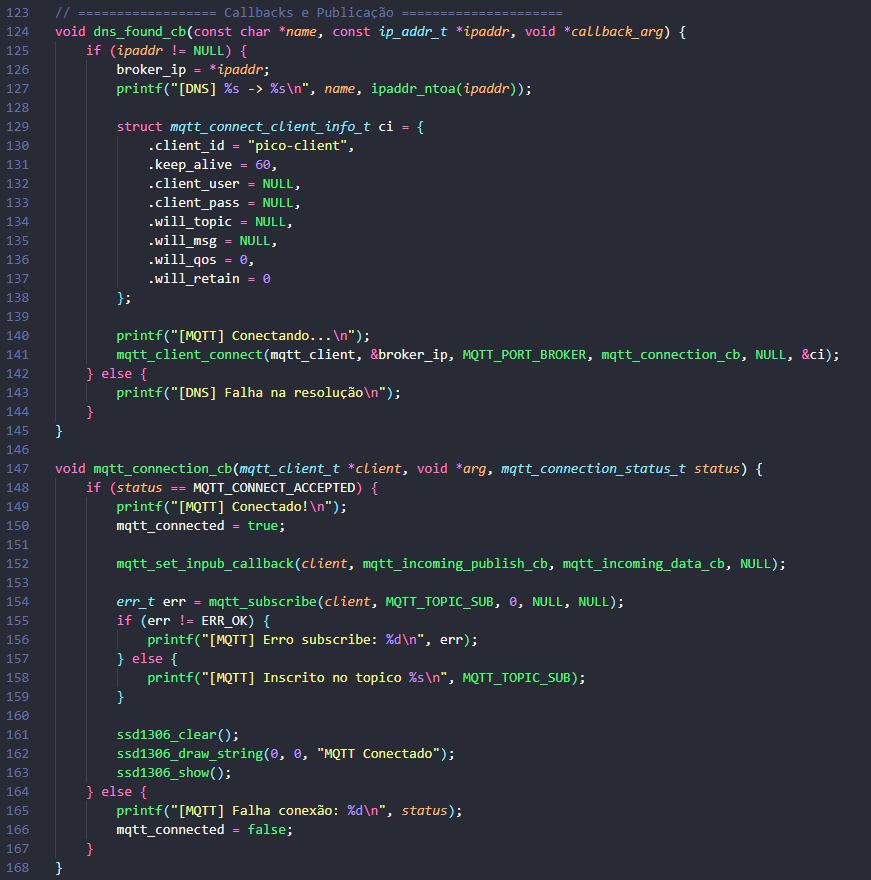


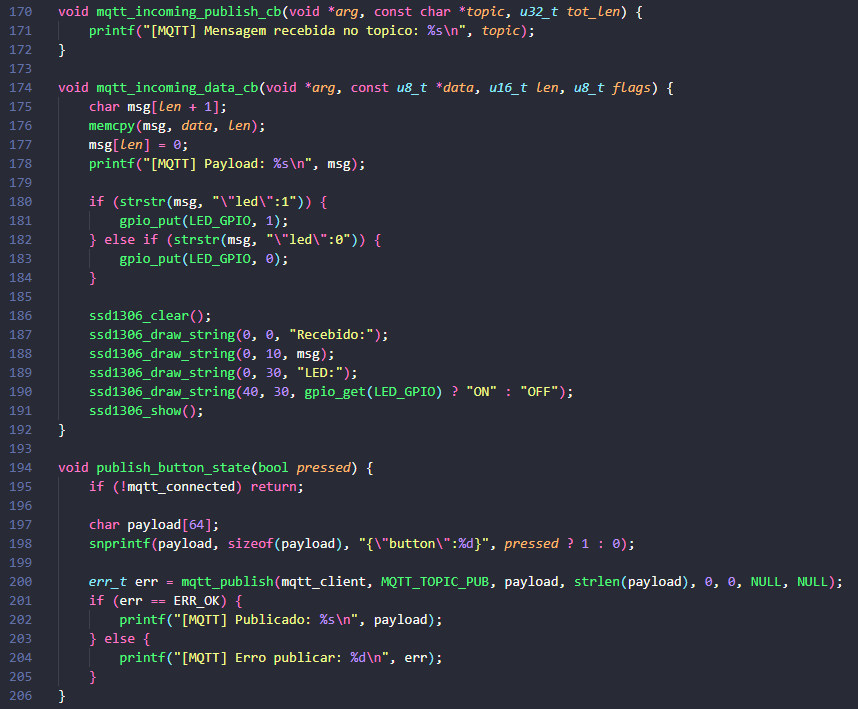
Aqui temos os includes iniciais e as definições necessárias para o desenvolvimento do programa.



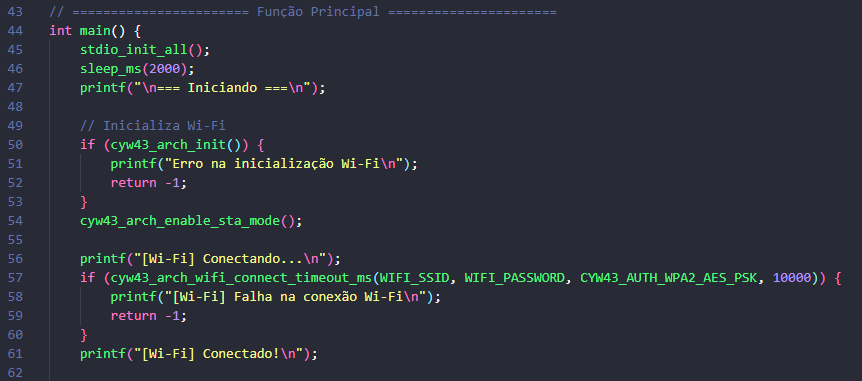
Na imagem acima temos as variáveis necessárias, já que nos baseamos em nosso programa anterior, bem como os protótipos das funções desenvolvidas.

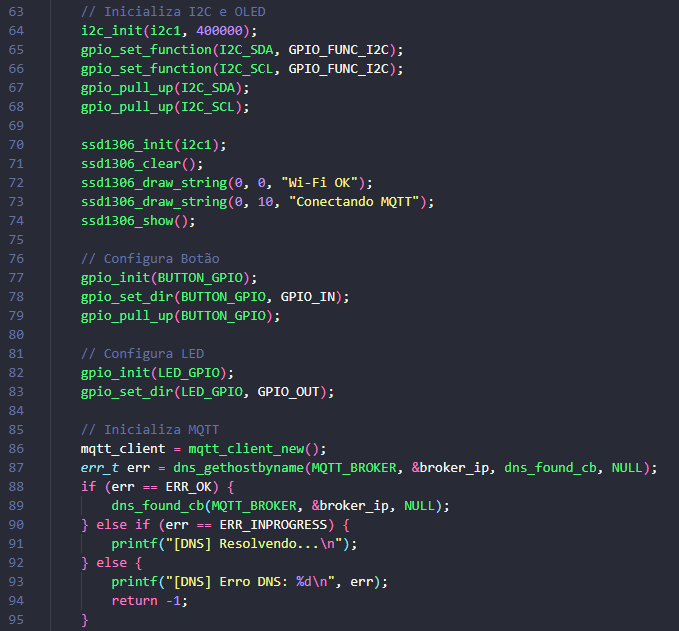
Na página seguinte temos a codificação das duas funções.

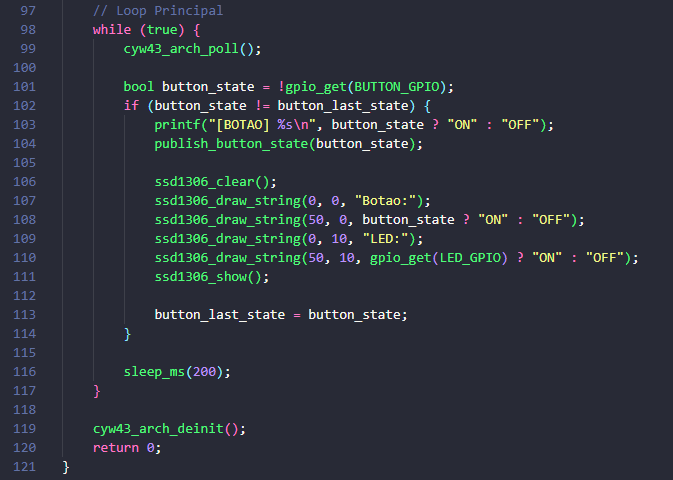




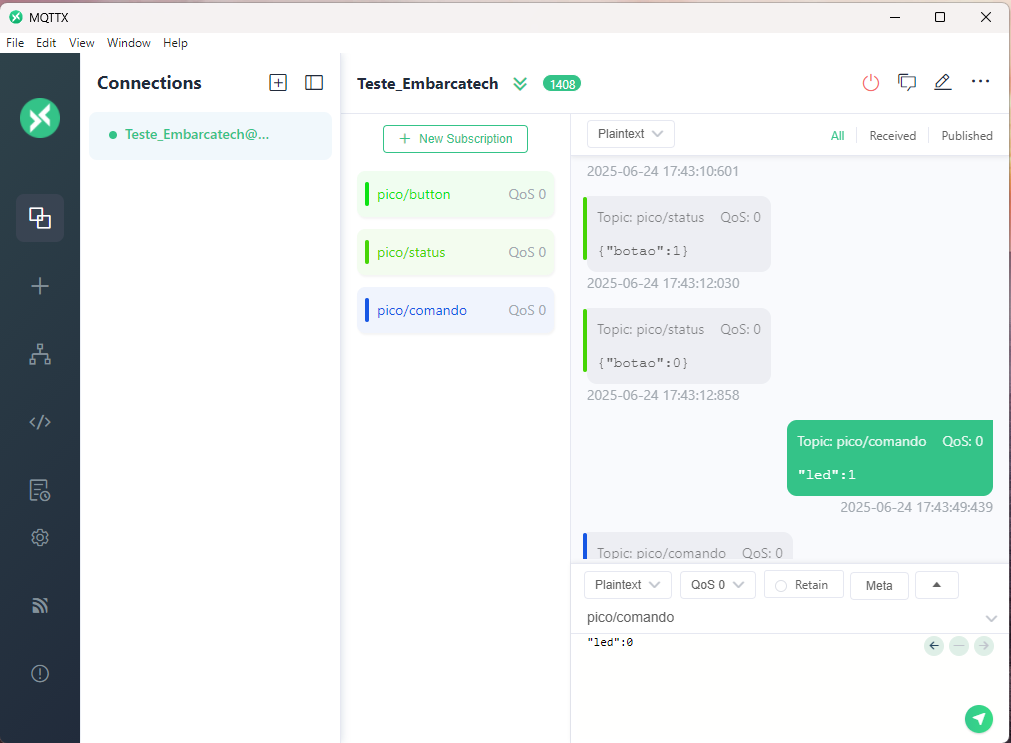
Após o desenvolvimento das funções, montamos nosso algoritmo para a função principal, conforme mostram as imagens abaixo.

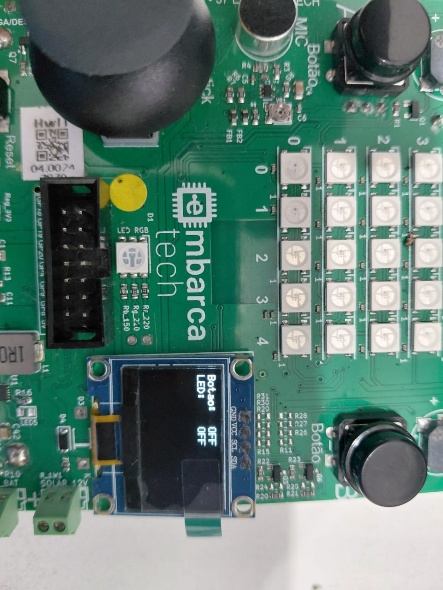


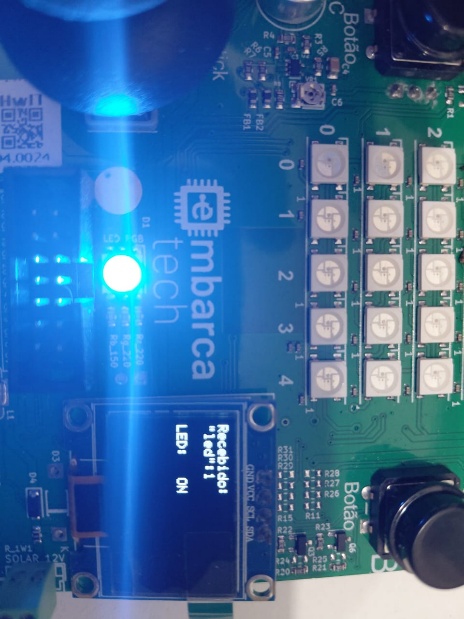


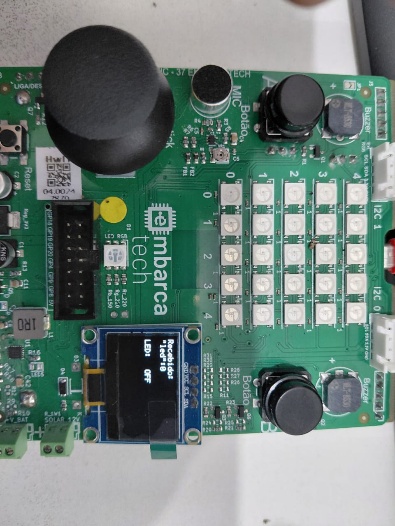


Como na versão anterior, utilizamos o “MQTTX” como cliente para enviar e receber os comandos MQTT do broker da HiveMQ.



Fotos da placa BigDogLab executando o firmware:





**Desafios:**

**Servidor em nuvem:** Refaça as tarefas anteriores, utilizando outro servidor MQTT, como por exemplo: AWS, Google, MQTTX e entre outros.

Desafio 1: Utilizando a placa Bitdoglab, crie um programa para monitorar o status de um pino.

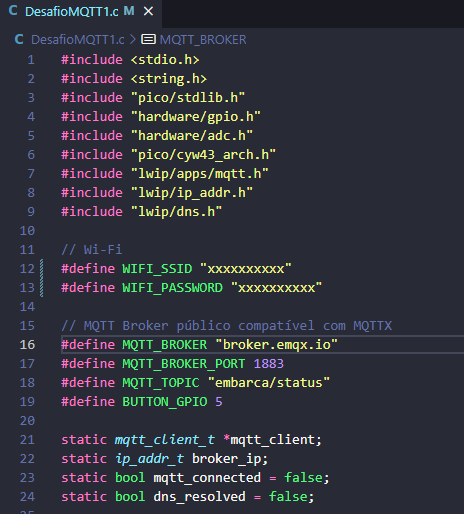
“*A Amazon Web Services (AWS), assim como outras BigTechs oferece um nível gratuito para novos usuários, permitindo o acesso a diversos serviços por um período limitado ou com uso limitado. Este nível gratuito não tem prazo de validade, mas é importante verificar os limites de uso de cada serviço, pois após ultrapassar esses limites, pode haver cobranças.*”

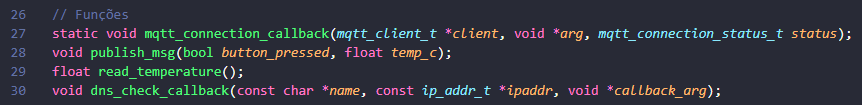
Depois de ver essa informação e que é obrigatório o cadastro de um cartão de crédito, decidi fazer meu teste com nuvem no broker MQTTX, que tem um acesso grátis sem o uso do cartão.

Abaixo o código utilizado, para acesso a este novo broker:

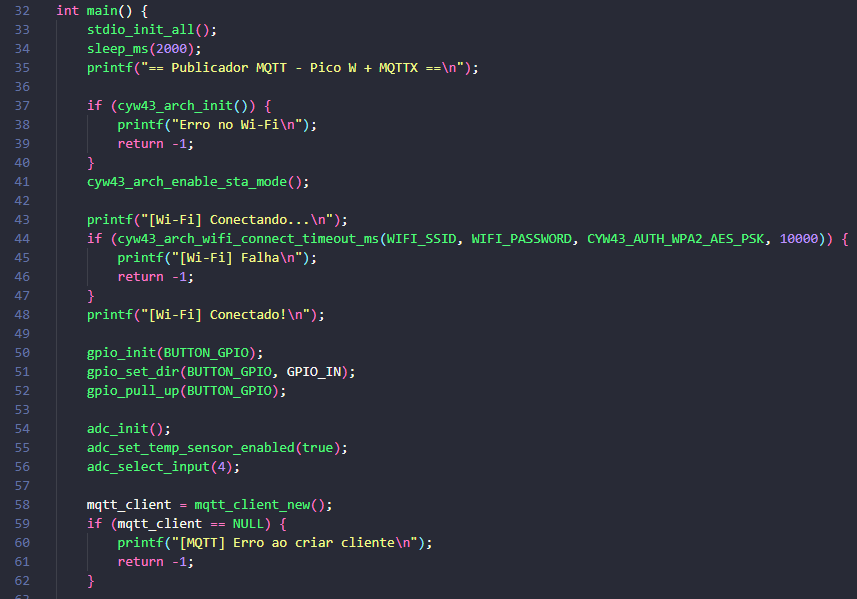
Conforme podemos perceber, não temos muitas mudanças em relação ao acesso anterior, até porque, os broker utilizados até o momento são todos em nuvem, apenas utilizo um cliente em meu computador para fazer a inscrição nos tópicos utilizados e receber os dados.

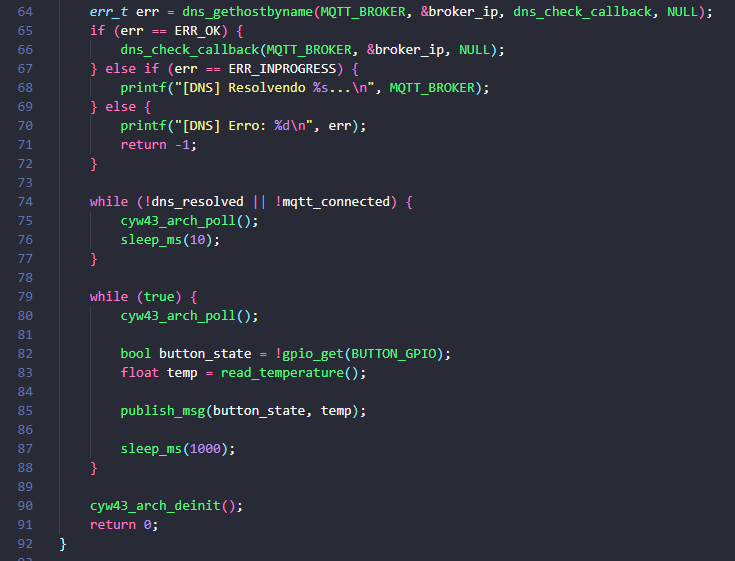
Iniciamos com os includes necessários, definições e protótipo das funções:



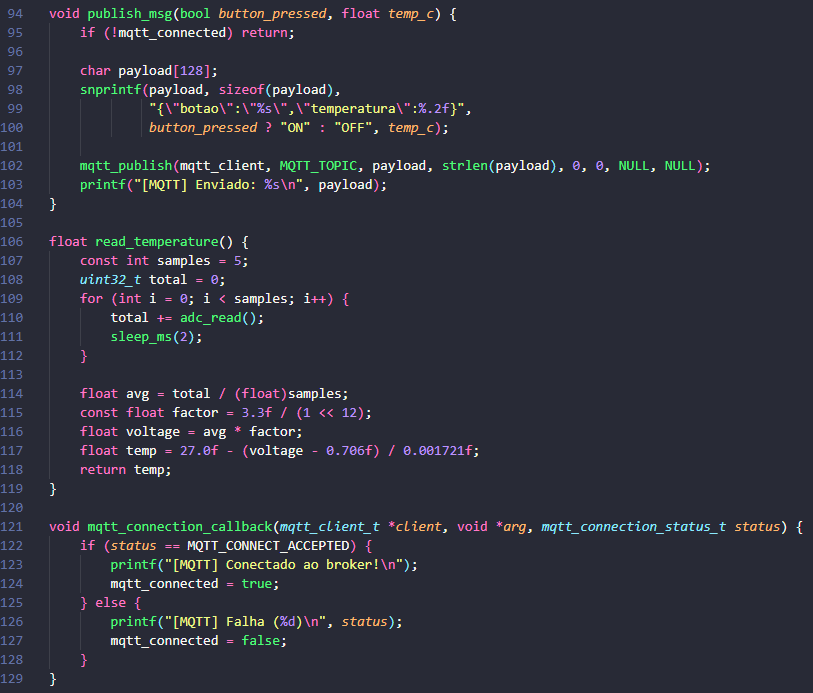


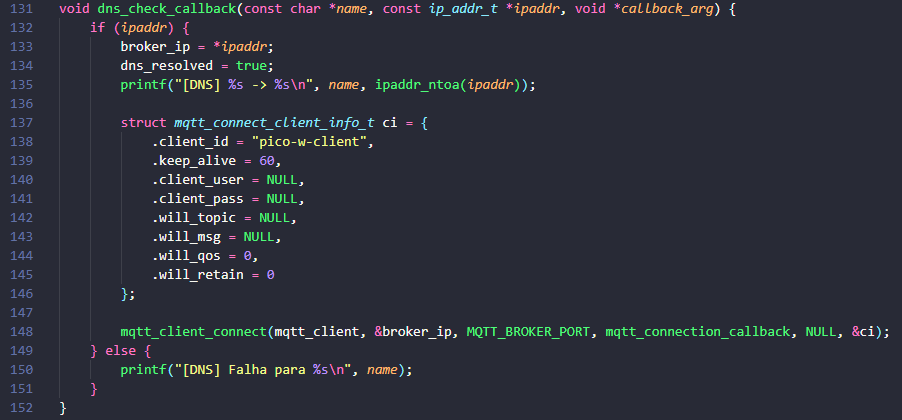
Nosso *main*:



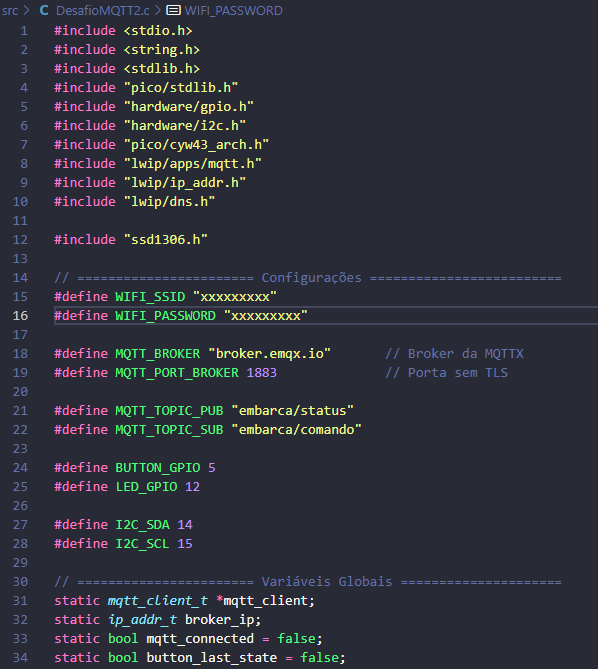


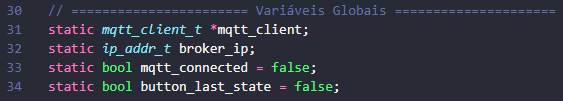
Nossas funções:



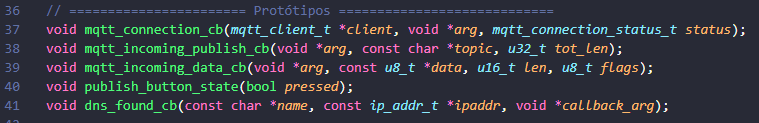


Desafio 2: Reescrever o programa 2 com outro broker em nuvem:

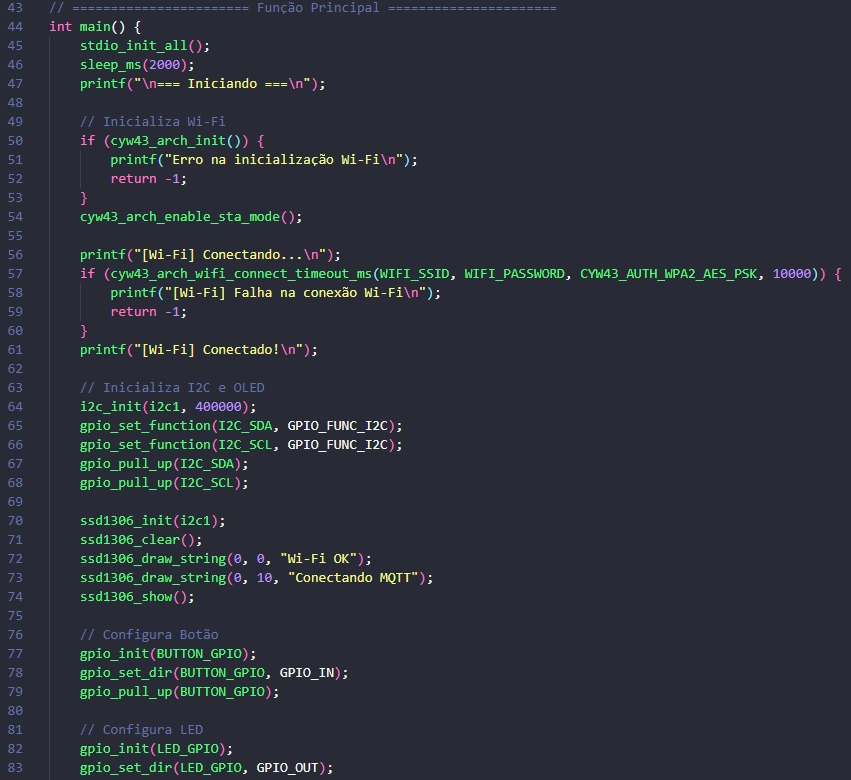


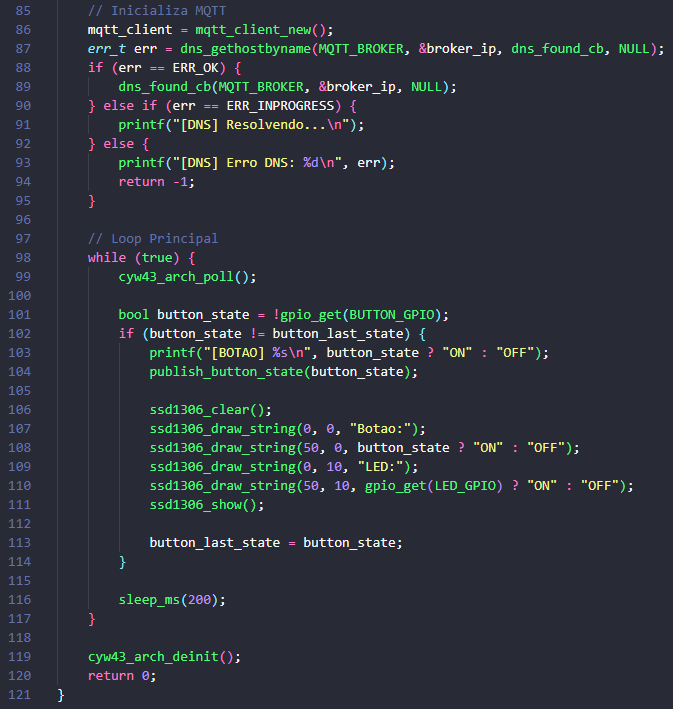


Nossos protótipos de funções:

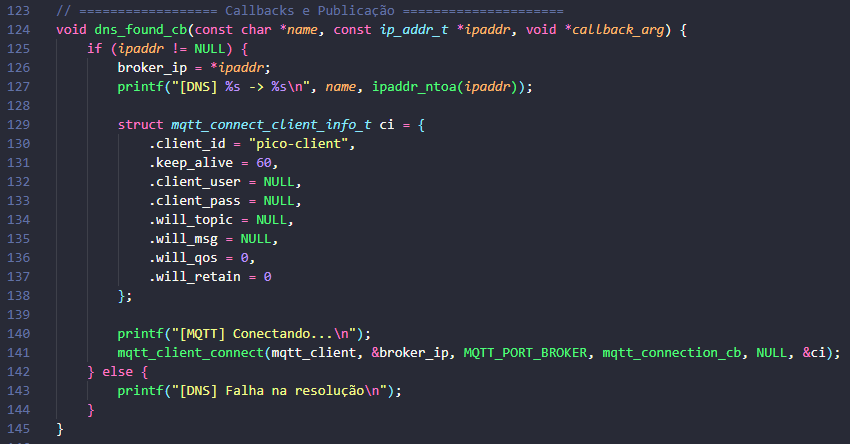


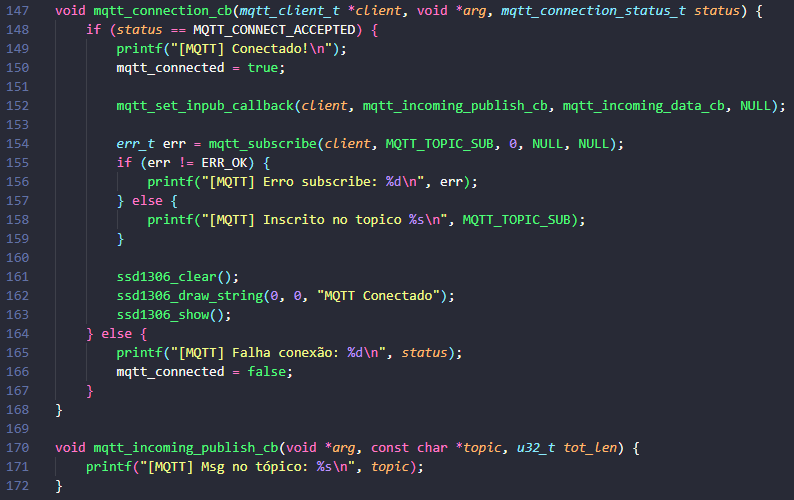
Nosso “*main*”:

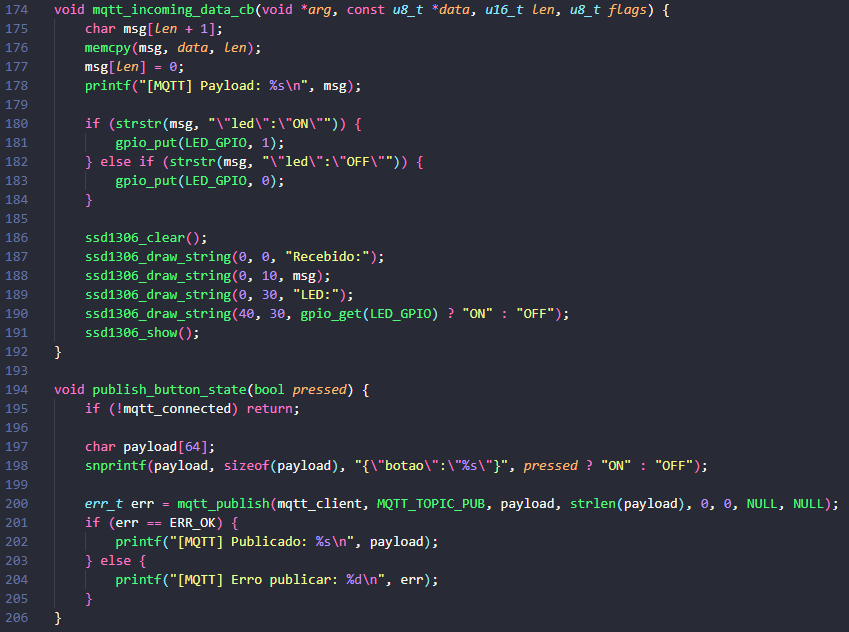




Nossas funções:







Todos os nossos códigos trabalham em nuvem, por isso utilizamos a função para acesso a um servidor DNS para localizar o IP fixo do broker na nuvem.

Os teste são similares as versões iniciais.