

Utilizando o protocolo MQTT do ESP32 para acender LED de outro circuito

Elias de Almeida Sombra Neto¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - Campus Maracanaú
Av. Parque Central, 1315 - Distrito Industrial I, Maracanaú-CE, Brasil

`elias.almeida09@aluno.ifce.edu.br`

Abstract. *This work presents a practical activity developed in the Microcontrollers course at IFCE, aiming to implement communication between two ESP32 boards using the MQTT protocol. The project consisted of remotely controlling LEDs and a buzzer across two distinct circuits. Circuit assembly was performed, microcontroller code was developed in C++ with PlatformIO, and the MQTT Test Client tool was used to simulate message exchange between devices. The results confirmed the system's correct operation, validating the effectiveness of MQTT-based communication between microcontrollers.*

Resumo. *Este trabalho apresenta uma atividade prática desenvolvida na disciplina de Microcontroladores do IFCE, com o objetivo de implementar a comunicação entre dois ESP32 utilizando o protocolo MQTT. A proposta consistiu em controlar remotamente LEDs e um buzzer entre dois circuitos distintos. Para isso, foi feita a montagem dos circuitos, a programação dos microcontroladores em C++ utilizando PlatformIO, e o uso da ferramenta MQTT Test Client para simular a troca de mensagens entre os dispositivos. Os resultados demonstraram o funcionamento correto do sistema, validando a eficácia da comunicação via MQTT entre os microcontroladores.*

1. Introdução

Este relatório descreve uma atividade prática realizada no Laboratório de Eletroeletrônica e Sistema Embarcados (LAESE) durante a disciplina de Microcontroladores no Instituto Federal do Ceará (IFCE).

O objetivo da atividade é utilizar o protocolo MQTT para estabelecer comunicação entre diferentes circuitos com ESP32. Nesse sentido, ambos circuitos devem ser capazes de enviar e receber mensagens de controle de um LED. Para isso, além da montagem do circuito e do desenvolvimento do código, foi utilizada a plataforma MQTT Test Client para testar o canal de comunicação criado entre os microcontroladores.

2. Materiais utilizados

Os materiais para a construção e acionamento do circuito incluem:

- 1 ESP32 30 pinos;
- 1 cabo micro USB;
- 1 protoboard;
- 1 LED;

- 1 buzzer;
- 1 botão;
- 2 resistores;
- 3 cabos jumper macho-fêmea;
- 2 cabos jumper macho-macho.

3. Montagem do circuito

Para realizar a montagem do circuito, é preciso seguir os seguintes passos:

- Ligar o ESP32 em uma fonte de alimentação com o cabo micro USB.
- Ligar o pino GND do ESP32 na região de alimentação da protoboard através de um jumper macho-fêmea.
- Posicionar o LED na região de componentes da protoboard.
- Posicionar um pino do resistor na região de alimentação e o outro em paralelo com o pino negativo do LED.
- Ligar o GPIO13 do ESP32 em paralelo com o pino positivo do LED através de um jumper macho-fêmea.
- Posicionar o buzzer na região de componentes da protoboard.
- Ligar o pino positivo do buzzer em paralelo com o pino positivo do LED através de um jumper macho-macho.
- Ligar o pino negativo do buzzer em paralelo com a região de alimentação da protoboard através de um jumper macho-macho.
- Posicionar o botão na região de componentes da protoboard.
- Posicionar uma extremidade do resistor em paralelo com um dos pinos do botão e a outra extremidade na região de alimentação da protoboard.
- Ligar um dos pinos do botão no GPIO12 do ESP32 através de um jumper macho-fêmea.

4. Implementação do código

O código responsável por controlar o LED através de um canal de comunicação entre diferentes ESP32's foi desenvolvido com a linguagem C++ no editor de código Visual Studio Code juntamente com sua extensão PlatformIO.

```

1 #include <Arduino.h>
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4
5 #define LED 13
6 #define BUTTON 12
7
8 const char *ssid = "NOME_DA_REDE";
9 const char *password = "SENHA_DA_REDE";
10
11 const char *mqtt_broker = "broker.mqtt.cool";
12 const char *topic = "LAESE";
13 const int mqtt_port = 1883;
14
15 bool isLedOn = false;

```

```

16
17 String message;
18
19 WiFiClient espClient;
20 PubSubClient client(espClient);
21
22 void callback(char *topic, byte *payload, unsigned int length) {
23     message = "";
24     Serial.print("Message arrived in topic: ");
25     Serial.println(topic);
26     Serial.print("Message:");
27     for (int i = 0; i < length; i++) {
28         Serial.print((char) payload[i]);
29         message += (char) payload[i];
30     }
31     Serial.println();
32     Serial.println("-----");
33 }
34
35 void setup() {
36     Serial.begin(9600);
37
38     pinMode(LED, OUTPUT);
39     pinMode(BUTTON, INPUT_PULLUP);
40
41     WiFi.begin(ssid, password);
42     while (WiFi.status() != WL_CONNECTED) {
43         delay(500);
44         Serial.println("Connecting to WiFi..");
45     }
46
47     Serial.println(WiFi.localIP());
48
49     client.setServer(mqtt_broker, mqtt_port);
50     client.setCallback(callback);
51     while (!client.connected()) {
52         String client_id = "esp32-client-";
53         client_id += String(WiFi.macAddress());
54         Serial.printf("The client %s connects to the public MQTT
55             broker\n", client_id.c_str());
56         if (client.connect(client_id.c_str())) {
57             Serial.println("Public EMQX MQTT broker connected");
58         } else {
59             Serial.print("failed with state ");
60             Serial.print(client.state());
61             delay(2000);
62         }
63     }

```

```

64   client.subscribe(topic);
65 }
66
67 void loop() {
68     int buttonState = digitalRead(BUTTON);
69
70     if (buttonState == LOW) {
71         isLedOn = !isLedOn;
72         if (isLedOn) {
73             client.publish(topic, "led_amarelo_on");
74         }
75         else {
76             client.publish(topic, "led_amarelo_off");
77         }
78     }
79
80     if (message == "led_verde_on") {
81         digitalWrite(LED, HIGH);
82         delay(1000);
83     }
84     else if (message == "led_verde_off") {
85         digitalWrite(LED, LOW);
86         delay(1000);
87     }
88     client.loop();
89 }

```

5. MQTT Test Client

MQTT Test Client é uma plataforma web disponibilizada publicamente pela URL <https://testclient-cloud.mqtt.cool>. Sua função é fornecer uma interface gráfica para criação e inscrição de tópicos nos quais mensagens podem ser enviadas e recebidas. Sendo assim, o site foi usado para testar o protocolo de comunicação entre os circuitos montados em laboratório.

Para realizar o teste, foi necessário criar um tópico chamado LAESE e solicitar que um colega de laboratório se inscrevesse nele para compartilhar mensagens entre diferentes circuitos. No caso desse relatório, foi utilizado um LED da cor verde enquanto o outro circuito possui um LED amarelo. Portanto, ao pressionar o botão do circuito, o ESP32 envia uma mensagem por meio do protocolo MQTT para ligar ou desligar o LED amarelo do outro circuito. De maneira análoga, a sinalização do outro lado do canal aciona o LED verde juntamente com o buzzer.

6. Resultados

A montagem e implementação do código para estabelecer comunicação entre diferentes circuitos por meio do MQTT foi concluída de forma bem sucedida. A seguir, pode-se visualizar a forma do circuito no final da prática, assim como os testes realizados e uma demonstração do seu funcionamento.

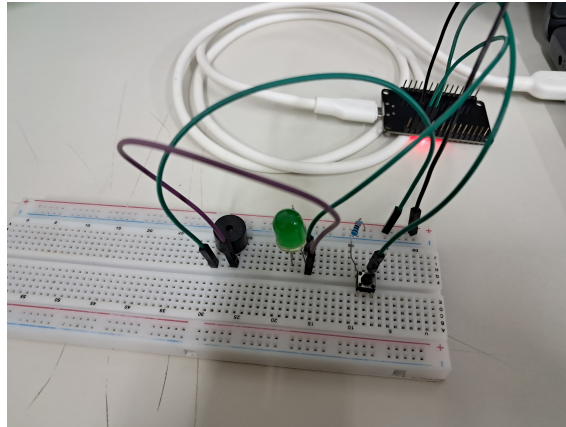


Figure 1. Circuito montado com um LED verde que é acionado e um botão que aciona outro circuito

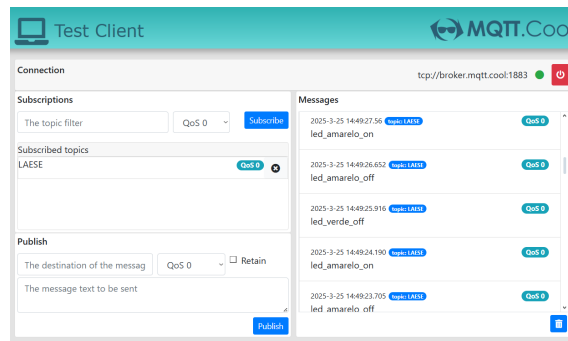


Figure 2. Captura de tela do MQTT Test Client com o tópico utilizado e o histórico de mensagens

O vídeo que demonstra o circuito em funcionamento está disponível em: [Resultado da atividade prática 09](#)

7. Conclusão

Após a finalização dessa atividade prática em laboratório, foi possível projetar o circuito e desenvolver o código que estabelece um canal de comunicação entre diferentes microcontroladores, além de controlar um LED por meio do protocolo MQTT. Isso foi possível com o uso de um microcontrolador ESP32 e alguns componentes eletrônicos como uma protoboard, um LED verde, um buzzer, um botão, dois resistores e alguns cabos jumper. Assim, o objetivo proposto foi atingido com sucesso.