

# Acionamento de LED RGB via Telegram

Elias de Almeida Sombra Neto<sup>1</sup>

<sup>1</sup>Instituto 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 the control of an RGB LED through commands sent by a Telegram bot, using the ESP32 platform connected to the internet. The activity included circuit assembly, bot creation via BotFather, and C++ code development. The system responded efficiently, demonstrating the potential of integrating microcontrollers with IoT applications.*

**Resumo.** *Este trabalho apresenta o acionamento de um LED RGB via comandos enviados por um bot do Telegram, utilizando a plataforma ESP32 conectada à internet. A atividade envolveu a montagem de um circuito simples, a criação do bot com o BotFather e o desenvolvimento de código em C++. O sistema respondeu de forma eficiente aos comandos recebidos, demonstrando o potencial da integração entre microcontroladores e aplicações de IoT.*

## 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).

A atividade consiste em ligar e desligar um LED RGB com comandos digitados no chat de um bot do Telegram. Portanto, além de montar e codificar o circuito, foi necessário conectar o ESP32 em uma rede WiFi para que o LED pudesse ser acionado via chat. Em seguida, foi preciso criar o bot com o recurso "BotFather" do Telegram e codificar os comandos que serão gravados e reconhecidos pelo ESP32.

## 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 RGB;
- 1 resistor;
- 4 cabos jumper macho-fêmea.

## 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 RGB na região de componentes da protoboard.
- Posicionar um pino do resistor na região de alimentação e a outra em paralelo com o pino negativo do LED RGB.
- Ligar o pino 13 do ESP32 em paralelo com o pino correspondente a cor vermelha do LED RGB através de um jumper macho-fêmea.
- Ligar o pino 12 do ESP32 em paralelo com o pino correspondente a cor azul do LED RGB através de um jumper macho-fêmea.
- Ligar o pino 14 do ESP32 em paralelo com o pino correspondente a cor verde do LED RGB através de um jumper macho-fêmea.

#### 4. Criação do bot

Pra criar o bot, é preciso seguir os seguintes passos:

- Acessar o Telegram e buscar por "BotFather" para iniciar uma conversa;
- Digitar o comando "/start" para ter acesso ao BotFather;
- Criar um novo bot através do comando "/newbot";
- O BotFather solicitará um nome, mas é importante que tenha a terminação em "\_bot";
- Salvar o token para acesso da API do BotFather após receber a resposta de bot criado com sucesso;
- Pesquisar por "IdBot" no Telegram e iniciar uma nova conversa;
- Digitar o comando "/start" e salvar o ID que o IdBot fornecerá.

#### 5. Implementação do código

O código responsável por acionar o LED via comando no chat bot do Telegram foi desenvolvido com a linguagem C++ no editor de código Visual Studio Code juntamente com sua extensão PlatformIO.

```

1 #include <WiFiClientSecure.h>
2 #include <UniversalTelegramBot.h>
3 #include <Arduino.h>
4
5 const char* ssid = "NOME_DA_REDE";
6 const char* password = "SENHA_DA_REDE";
7
8 #define BOTtoken "TOKEN"
9 #define CHAT_ID "ID"
10
11 WiFiClientSecure client;
12 UniversalTelegramBot bot(BOTtoken, client);
13
14 int botRequestDelay = 1000;
15 unsigned long lastTimeBotRan;
16
17 #define LED_VERMELHO 13
18 #define LED_AZUL 12

```

```

19 #define LED_VERDE 14
20
21 void handleNewMessages(int numNewMessages) {
22     Serial.println("handleNewMessages");
23     Serial.println(String(numNewMessages));
24
25     for (int i=0; i<numNewMessages; i++) {
26         String chat_id = String(bot.messages[i].chat_id);
27         if (chat_id != CHAT_ID){
28             bot.sendMessage(chat_id, "Usuario nao autorizado", "");
29             continue;
30         }
31
32         String text = bot.messages[i].text;
33         Serial.println(text);
34
35         String from_name = bot.messages[i].from_name;
36
37         if (text == "/start") {
38             String welcome = "Bem vindo, " + from_name + ".\n";
39             welcome += "Use os seguintes comandos para controlar o\n";
40             welcome += "circuito.\n\n";
41             welcome += "/led_vermelho_on \n";
42             welcome += "/led_vermelho_off\n";
43             welcome += "/led_azul_on \n";
44             welcome += "/led_azul_off\n";
45             welcome += "/led_verde_on \n";
46             welcome += "/led_verde_off\n";
47             bot.sendMessage(chat_id, welcome, "");
48         }
49
50         if (text == "/led_vermelho_on") {
51             bot.sendMessage(chat_id, "LED vermelho foi ligado", "");
52             digitalWrite(LED_AZUL, LOW);
53             digitalWrite(LED_VERDE, LOW);
54             digitalWrite(LED_VERMELHO, HIGH);
55         }
56
57         if (text == "/led_vermelho_off") {
58             bot.sendMessage(chat_id, "LED vermelho foi desligado", "");
59             ;
60             digitalWrite(LED_VERMELHO, LOW);
61         }
62
63         if (text == "/led_azul_on") {
64             bot.sendMessage(chat_id, "LED azul foi ligado", "");
65             digitalWrite(LED_VERMELHO, LOW);
66             digitalWrite(LED_VERDE, LOW);
67             digitalWrite(LED_AZUL, HIGH);

```

```

66     }
67
68     if (text == "/led_azul_off") {
69         bot.sendMessage(chat_id, "LED azul foi desligado", "");
70         digitalWrite(LED_AZUL, LOW);
71     }
72
73     if (text == "/led_verde_on") {
74         bot.sendMessage(chat_id, "LED verde foi ligado", "");
75         digitalWrite(LED_AZUL, LOW);
76         digitalWrite(LED_VERMELHO, LOW);
77         digitalWrite(LED_VERDE, HIGH);
78     }
79
80     if (text == "/led_verde_off") {
81         bot.sendMessage(chat_id, "LED verde foi desligado", "");
82         digitalWrite(LED_VERDE, LOW);
83     }
84 }
85 }
86
87 void setup() {
88     pinMode(LED_VERMELHO, OUTPUT);
89     pinMode(LED_AZUL, OUTPUT);
90     pinMode(LED_VERDE, OUTPUT);
91
92     Serial.begin(9600);
93
94     WiFi.mode(WIFI_STA);
95     WiFi.begin(ssid, password);
96
97     #ifndef ESP32
98         client.setCACert(TELEGRAM_CERTIFICATE_ROOT);
99     #endif
100
101     while (WiFi.status() != WL_CONNECTED) {
102         delay(1000);
103         Serial.println("Conectando com o WiFi...");
104     }
105
106     Serial.println(WiFi.localIP());
107 }
108
109 void loop() {
110     if (millis() > lastTimeBotRan + botRequestDelay) {
111         int numNewMessages = bot.getUpdates(bot.
            last_message_received + 1);
112
113         while (numNewMessages) {

```

```
114     Serial.println("Resposta adquirida");
115     handleNewMessages(numNewMessages);
116     numNewMessages = bot.getUpdates(bot.last_message_received
        + 1);
117 }
118 lastTimeBotRan = millis();
119 }
120 }
```

## 6. Resultados

A montagem e implementação do código para acionar o LED uso de comandos via bot do Telegram foi concluída de forma bem sucedida. A seguir, pode-se visualizar a forma do circuito no final da prática, assim como uma demonstração do seu funcionamento.

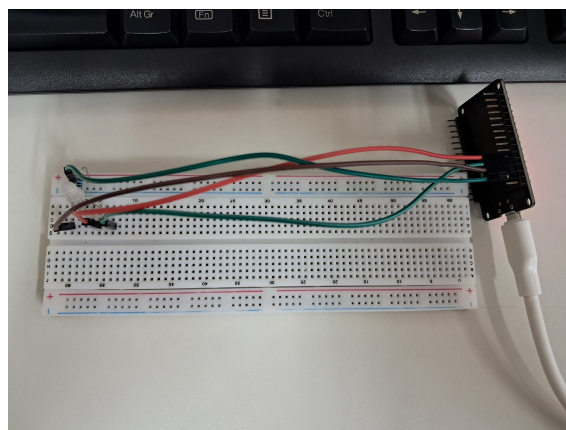
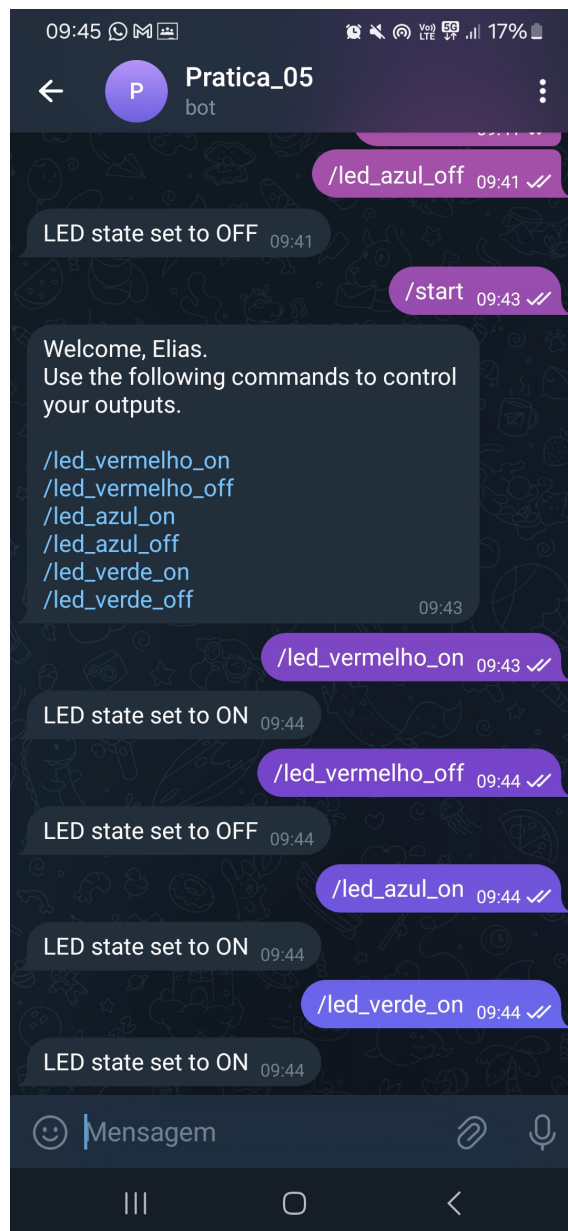


Figure 1. Circuito que liga e desliga o LED por meio do Telegram



**Figure 2. Comandos digitados no chat do bot que foi criado para acionar o circuito**

O vídeo que demonstra o circuito em funcionamento está disponível em: [Resultado da atividade prática 05](#)

## 7. Conclusão

Após a finalização dessa atividade prática em laboratório, foi possível montar um circuito, criar um chat bot no Telegram e desenvolver um código que liga e desliga o LED RGB via WiFi. Dessa maneira, o circuito pode ser acionado a longas distâncias devido a capacidade do ESP32 de se conectar em redes sem fio. Assim, com a criação de um bot e com o uso do LED RGB, um resistor, o ESP32, uma protoboard e alguns cabos jumper foi possível concluir essa atividade. Por isso, o objetivo proposto foi atingido com sucesso.