



UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL
INTERNET DAS COISAS – 2018.2 – PROF. WENDLEY S. SILVA

www.ec.ufc.br/professores/wendley/
wendley@ufc.br

AULA PRÁTICA – ACESSO AO NODEMCU ESP-12E ATRAVÉS DO
APLICATIVO TELEGRAM

MONITORA: KAMILA AMÉLIA SOUSA GOMES

APLICATIVO TELEGRAM

O Telegram é um aplicativo de troca de mensagens concorrente do Whatsapp. O aplicativo pode ser usado no smartphone (Android e IOS) e também no seu PC (Windows, MAC ou Linux). O TELEGRAM tem um BOT que permite fazer a conexão IOT .

Definição de BOT – Wikipedia:

“Bot, diminutivo de robot, também conhecido como Internet bot ou web robot, é uma aplicação de software concebida para simular ações humanas repetidas vezes de maneira padrão, da mesma forma como faria um robô.” Com o BOT do TELEGRAM, você poderá receber mensagens dos dispositivos (nesse caso, o NodeMCU) e enviar comandos, através do seu Smartphone ou PC.

INSTALANDO O TELEGRAM

Se você usa um Smartphone Android ou um IPHONE, instale o aplicativo através da Google Play ou da Apple Store. Após a instalação e configuração do TELEGRAM , vai descobrir que muitos amigos e conhecidos seus já são usuários deste aplicativo. Da forma que você pode conversar com os seus amigos, você poderá “conversar” também com o BOT, através de comandos.

CONFIGURANDO O BOT DO TELEGRAM

Para poder criar um novo usuário BOT, é necessário acessar o usuário BOT Pai (BotFather), como observado na Figura 1.

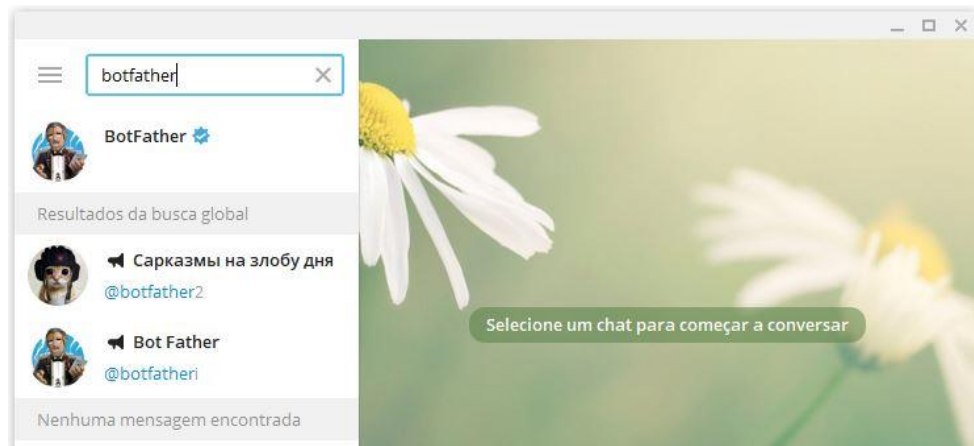


Figura 1 – Acessando BotFather no Telegram

Na janela do BotFather, na linha de comandos (parte inferior) clique em Começar. Irá aparecer uma lista de comandos do BotFather. (Digite /help se precisar de ajuda).

Todos os comandos do BOT devem começar com uma barra /.

Para criar o seu BOT , digite o comando /newbot . Após a primeira pergunta, insira o nome do seu BOT. Se o seu nome já existir, use um outro nome , como por exemplo (no meu caso):

GustESP8266

Na segunda pergunta, digite o nome do seu novo usuário. O nome do usuário deve terminar com bot. Se o seu nome já existir, use um outro nome , como por exemplo :

GustESP8266bot

Se o usuário foi criado com sucesso, aparecerá a mensagem " Done!" . Veja que uma chave TOKEN foi criada. Essa chave deverá ser copiada, pois será inserida no Sketch do NodeMCU ESP12 !

Use this token to access the HTTP API (exemplo):

123456789:ABCDEFGHIJKLMOPQRST-ABCDEFGHIJKLMNO

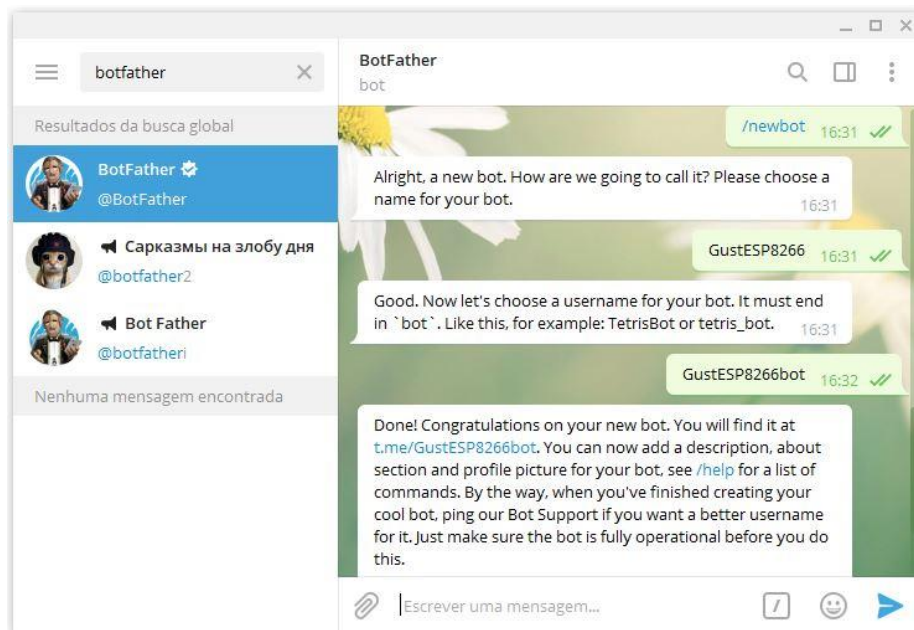


Figura 2 – Configurando BotFather no Telegram

MONTAGEM PROPOSTA PARA O HARDWARE

Inicialmente, faremos a montagem do hardware conforme a figura a abaixo:

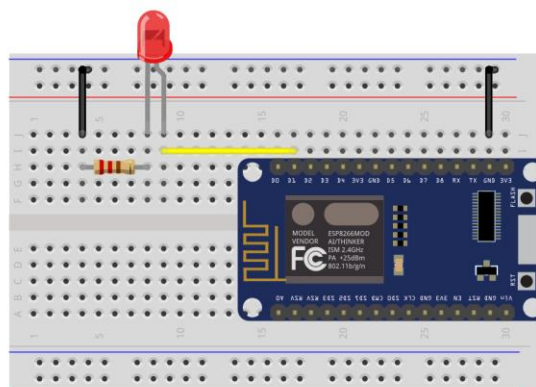


Figura 3 – Montagem proposta para o hardware

Nesse ponto, é importante mencionar que os pinos do NodeMCU e Arduino possuem o seguinte mapeamento:

- D0 = 16;
- D1 = 5;
- D2 = 4;
- D3 = 0;
- D4 = 2;
- D5 = 14;
- D6 = 12;

D7 = 13;
D8 = 15;
D9 = 3;
D10 = 1;

Ou seja, como no esquema da Figura 3 o LED está associado à porta D2, no código-fonte, se necessário tratar manualmente, usaremos pin de valor 4.

INSTALANDO BIBLIOTECAS ARDUINO PARA O TELEGRAM

Para usar o NodeMCU-ESP12 com o TELEGRAM, será necessário a instalação de duas Bibliotecas na Arduino IDE:

- *ArduinoJson*
- *Universal Arduino Telegram Bot*

Vamos instalar essas duas Bibliotecas, usando o **Gerenciador de Bibliotecas**. Abra a Arduino IDE, já configurada para o NodeMCU-ESP12. Para instalar a nova Biblioteca, clique em **Sketch > Incluir Biblioteca > Gerenciar Bibliotecas**. Na janela do Gerenciador de Bibliotecas, refine a busca digitando **arduinojson**. Clique em **more info** e depois em **instalar**.



Figura 4 – Instalando Bibliotecas Arduino para Telegram

Vamos instalar agora a segunda biblioteca UniversalTelegramBot com o mesmo procedimento acima. Refine a busca digitando telegram. Clique em more info e depois em instalar. Após a instalação das duas Bibliotecas, é necessário que feche e abra novamente o programa Arduino IDE, para efetivar as Bibliotecas.

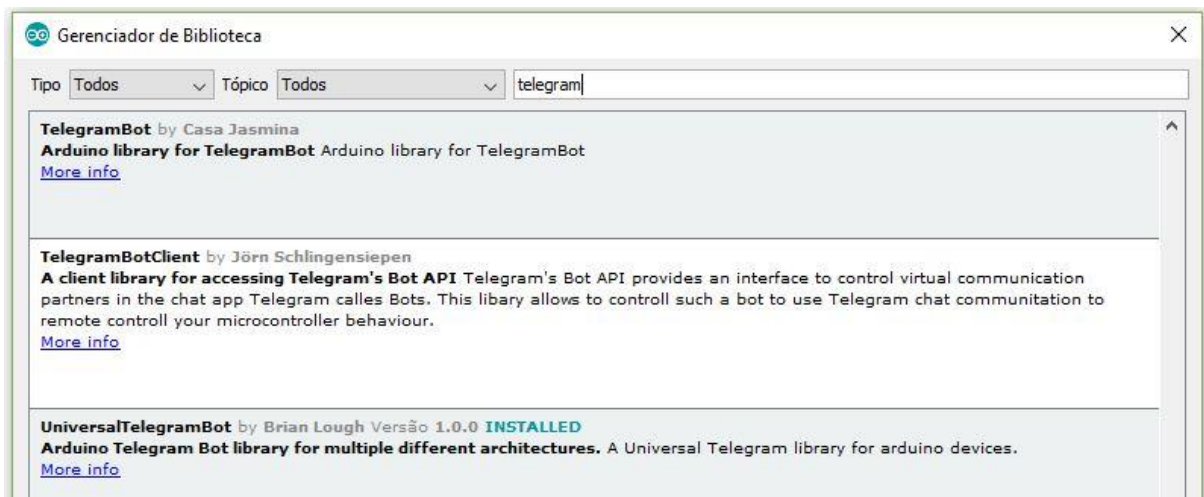


Figura 5 – Instalando Bibliotecas Arduino para Telegram

TESTANDO NODEMCU COM TELEGRAM

Para testarmos a comunicação do Aplicativo **TELEGRAM** com o **NodeMCU-ESP12**, usaremos um Sketch baseado em um exemplo da Biblioteca **Universal Arduino Telegram Bot**. Sabemos que o **Led azul** da placa NodeMCU está conectado no pino **GPIO_4**. Um pulso LOW(0V) acionará esse led. Portanto, através de comandos no TELEGRAM, faremos o Led azul acender e apagar.

No código, você deverá alterar o nome do seu roteador WIFI (**ssid**) e a senha do roteador (**password**) também. Na linha do programa **#define BOTtoken**, insira a **chave Token** (entre aspas) que foi copiada quando gerou o seu usuário **BOT** através do **BotFather**.

Exemplo: **"123456789:ABCDEFGHIJKLMOPQRST-ABCDEFGHIJKLMNO"**

```
#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

// Inicializando a conexao WIFI com o Roteador

char ssid[] = "wifi";           // nome do seu roteador WIFI (SSID)
```

```
char password[] = "a1b2c3d4e5"; // senha do roteador WIFI

// Inicializa o BOT Telegram - copie aqui a chave Token quando configurou o
// seu BOT - entre aspas

#define          BOTtoken          "528420337:AAHLwwVuklQ2P8mbUh5-
7CdIlgq7mSzy5uYI" // sua chave Token Bot

WiFiClientSecure client;

UniversalTelegramBot bot(BOTtoken, client);

int Bot_mtbs = 1000; // tempo entre a leitura das mensagens

long Bot_lasttime; // ultima mensagem lida

bool Start = false;

const int ledPin = 4; // GPIO_4do LED azul

int ledStatus = 0;

void handleNewMessages(int numNewMessages){

    Serial.print("Mensagem recebida = ");

    Serial.println(String(numNewMessages));

    for (int i = 0; i < numNewMessages; i++){

        String chat_id = String(bot.messages[i].chat_id);

        String text = bot.messages[i].text;

        String from_name = bot.messages[i].from_name;
```

```
if (from_name == "") from_name = "Guest";

if (text == "/ledon")                // comando Ledon

    {digitalWrite(ledPin, HIGH);      // acende LED azul

    ledStatus = 1;

    bot.sendMessage(chat_id, "LED esta aceso", ""); // envia mensagem

}

If (text == "/ledoff")                // comando Ledoff

    { ledStatus = 0;

    digitalWrite(ledPin,LOW );        // apaga LED azul

    bot.sendMessage(chat_id, "LED esta apagado", ""); // envia mensagem

}

if (text == "/status")    {                // comando estado do LED

    if (ledStatus) {

        bot.sendMessage(chat_id, "LED esta aceso", ""); }

    else {

        bot.sendMessage(chat_id, "LED esta apagado", "");

    }

}
```

```

if (text == "/start") // comando começa

{String welcome = "Bem-vindo a Biblioteca Universal Arduino Telegram Bot, "
+ from_name + ".\n";

welcome += "Esse é um exemplo de controle do Led.\n\n";

welcome += "/ledon : para acender o LED\n";

welcome += "/ledoff : para apagar o LED\n";

welcome += "/status : mostra o estado do LED\n";

bot.sendMessage(chat_id, welcome, "Markdown");

}

}

}

void setup() {

    Serial.begin(115200);

    WiFi.mode(WIFI_STA); // Configura o WIFI do NodeMCU para modo
estação

    WiFi.disconnect(); // desconecta o WIFI

    delay(100); // atraso de 100 milisegundos

    Serial.print("Conectando Wifi: ");

    Serial.println(ssid);

```



```
WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) // aguardando a conexão
WEB {Serial.print(".");

        delay(500); // atraso de 0,5 segundos

    }

Serial.println("");

Serial.println("WiFi conectado"); // WIFI conectado

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

pinMode(ledPin, OUTPUT); // configura o pino do LED como saída

delay(10);

digitalWrite(ledPin, HIGH); // inicializa o LED como apagado

}

void loop(){

    if (millis() > Bot_lasttime + Bot_mtbs) // controlando as mensagens {

        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

        while (numNewMessages) // numero de novas mensagens{

            Serial.println("Resposta recebida do Telegram");
```

```
        handleNewMessages(numNewMessages);

        numNewMessages=bot.getUpdates(bot.last_message_received + 1);

    }

    Bot_lasttime = millis();}

}
```

Copie o código acima na Arduino IDE. Conecte a placa NodeMCU no seu PC. Certifique-se que ela foi reconhecida. Após compilar e carregar o programa no NodeMCU-ESP12 (clique no botão Carregar), abra a janela da console (Serial Monitor) da IDE. Clique no botão Monitor serial e altere a velocidade para 115200 Bps na barra inferior da janela.

No aplicativo TELEGRAM (poderá testar no seu PC ou no seu Smartphone) , procure pelo nome do seu usuário BOT . No meu caso **GustESP8266**. Clique no nome para abrir a janela de comunicação. Clique em **Começar** para iniciar os testes. Sempre que executar algum comando, aguarde alguns segundos para obter a resposta.

Comandos para teste (pode clicar em cima dos comandos já digitados) :

/ledon – para acender o LED

/ledoff – para apagar o LED

/status – para mostrar o estado do LED

/start – para inciar a comunicação

•

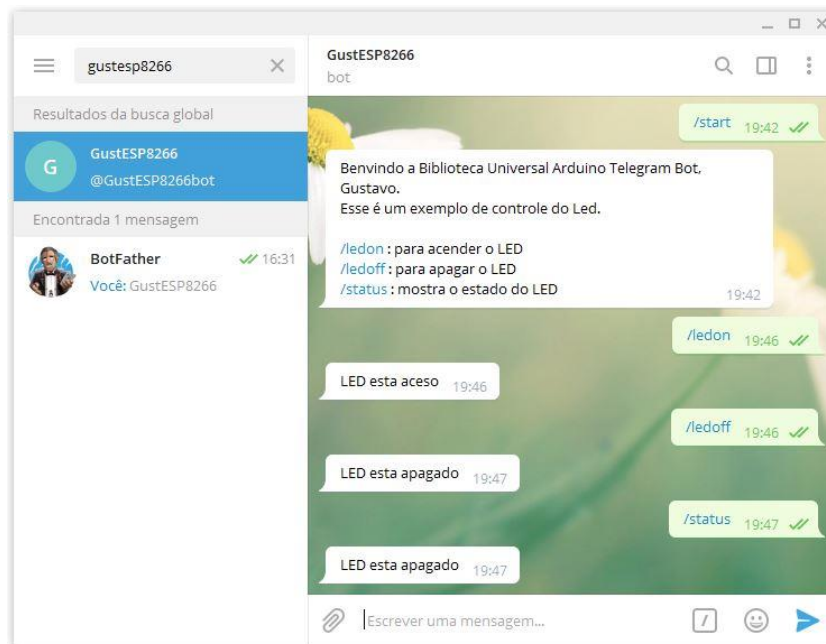


Figura 6: Exemplo de Comandos

TAREFA PARA DESENVOLVIMENTO NO LABORATÓRIO

Adicione um botão, ou sensor de temperatura, na montagem do seu hardware, e faça a comunicação com o aplicativo Telegram.

LITERATURA CONSULTADA

MURTA, Gustavo. . Guia Completo do NodeMCU - ESP12 Disponível em: <<http://blog.eletragate.com/nodemcu-esp12-alarme-residencial-iot-3/>>. Acesso em: 13 set. 2018.