

Para mais informações sobre o NodeMCU ESP-12E e a instalação dos módulos, reveja o material da aula prática zero.

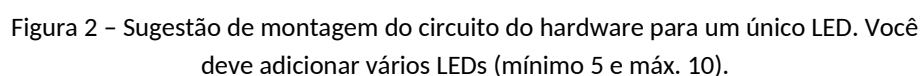
Este material orienta para a prática de como utilizar o MQTT para publicar mensagens em tópicos usando o serviço *online* CloudMQTT.

```

graph LR
    PRODUCER[PRODUCER] -- "Publish (topic, data)" --> BROKER[BROKER]
    BROKER -- "Publish (topic, data)" --> CONSUMER[CONSUMER]
    CONSUMER -. "Subscribe (topic)" .-> BROKER
  
```

The diagram illustrates the basic messaging pattern. It consists of three main components: a **PRODUCER** (yellow box), a **BROKER** (light blue box), and a **CONSUMER** (green box). The PRODUCER sends a message to the BROKER via a solid arrow labeled "Publish (topic, data)". The BROKER then delivers the message to the CONSUMER via a solid arrow labeled "Publish (topic, data)". The CONSUMER subscribes to the message via a dashed arrow labeled "Subscribe (topic)".

Abaixo está a sugestão de montagem de um circuito para um único led, mas você deve adicionar de 3 a 8 leds no seu circuito (adaptando no código depois).



ACESSO AO CloudMQTT

Inicialmente, crie a sua conta (caso não possua) no site <https://www.cloudmqtt.com/> . Crie uma nova instância e preencha os dados solicitados. Na opção “Details” você terá acesso aos dados protegidos (senha e usuário).

CÓDIGO INICIAL

Uma vez que o circuito esteja corretamente montado e que você já tenha criado sua conta no site CloudMQTT, atualize os campos senha e outros dados no início do código que está disponível no site da disciplina (baixe o arquivo **Pratica 02 – CloudMQTT.ino**). O mesmo código também está no final deste documento, como anexo.

EXECUÇÃO E PÓS-PROCESSAMENTO

Após compilar e executar o código no NodeMCU, vá ao site CloudMQTT, mais especificamente em **Websocket UI**, e envie mensagens com o tópico LED e a mensagem que desejar, conforme exemplos a seguir:

Comando	Função / ação
L1	Liga led 1
L2	Liga led 2
D1	Desliga led 1
LT	Liga todos os leds
DT	Desliga todos os leds

Você pode incluir outras mensagens a serem interpretadas, basta adicionar a funcionalidade no seu código-fonte.

O QUE ENVIAR PELO SOLAR?

- Enviar um arquivo .ZIP, no ambiente SOLAR (seção PORTFOLIO), contendo:
 - Código-fonte (geralmente um ou vários arquivos .ino)
 - Link para um vídeo de até 3 minutos mostrando o funcionamento da prática (hospedar em um serviço da nuvem, enviar apenas o link)
 - Relatório em PDF de até 3 pág. contendo os nomes dos membros da equipe, o que cada um fez, como a prática foi desenvolvida (equipamentos, versão do software, sistema operacional utilizado), resultados gerais, observações e link para o vídeo.

LITERATURA CONSULTADA

Documentation | CloudMQTT. Disponível em: < <https://www.cloudmqtt.com/docs/index.html> >. Acesso em: 04 mai. 2019.

GitHub - ckwiek/ESP8266_CloudMQTT_: Como conectar um ESP8266 no website Cloudmqtt.
Disponível em: < https://github.com/ckwiek/ESP8266_CloudMQTT >. Acesso em: 17 mai.
2018.

ANEXO

Código-fonte

```
/*//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
 *
 * Este programa mostra como fazer uma aplicacao usando linguagem mqtt para ligar e desligar leds
 /
 * Usando um ESP8266-12E
 * Foi utilizado o site IoT https://www.cloudmqtt.com/
 * O tutorial deste e de outras aplicações didaticas estão disponiveis no meu website
 * ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
 *
 * ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
 * Prática baseada no código disponível em www.carloskwiek.com.br
 * ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*/

#include <ESP8266WiFi.h>
#include <PubSubClient.h> // Biblioteca usada, baixe e instale se não a tiver, link abaixo

//https://github.com/knolleary/pubsubclient/blob/master/examples/mqtt_esp8266/
mqtt_esp8266.ino

//Define a pinagem do ESP8266

#define D0 16
#define D1 5
#define D2 4
#define D3 0
#define D4 2
#define D5 14
#define D6 12
#define D7 13
#define D8 15
#define D9 3
#define D10 1

const char* ssid = "nomeRede"; //Aqui o nome da sua rede local wi fi
const char* password = "senha"; // Aqui a senha da sua rede local wi fi
const char* mqttServer = "xxxx.cloudmqtt.com"; // Aqui o endereço do seu servidor fornecido pelo
site
const int mqttPort =00000; // Aqui mude para sua porta fornecida pelo site
const char* mqttUser = "nome"; // Aqui o nome de usuario fornecido pelo site
const char* mqttPassword = "senha"; // Aqui sua senha fornecida pelo site
//char EstadoSaida = '0';
WiFiClient espClient;
PubSubClient client(espClient);

void mqtt_callback(char* topic, byte* payload, unsigned int length);

void setup() {

    pinMode(D1, OUTPUT);
    digitalWrite(D1, LOW);
    pinMode(D2, OUTPUT);
    digitalWrite(D2, LOW);
```

```

pinMode(D3, OUTPUT);
digitalWrite(D3, LOW);
pinMode(D4, OUTPUT);
digitalWrite(D4, LOW);
pinMode(D5, OUTPUT);
digitalWrite(D5, LOW);
pinMode(D6, OUTPUT);
digitalWrite(D6, LOW);
pinMode(D7, OUTPUT);
digitalWrite(D7, LOW);
pinMode(D0, OUTPUT);
digitalWrite(D0, LOW);
pinMode(D8, OUTPUT);
digitalWrite(D8, LOW);

Serial.begin(115200);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.println("Connecting to WiFi..");
}
Serial.println("Connected to the WiFi network");

client.setServer(mqttServer, mqttPort);
client.setCallback(callback);

while (!client.connected()) {
  Serial.println("Connecting to MQTT...");

  if (client.connect("ESP8266Client", mqttUser, mqttPassword )) {

    Serial.println("connected");

  } else {

    Serial.print("failed with state ");
    Serial.print(client.state());
    delay(2000);

  }
}

client.publish("liga", "desliga");
client.subscribe("LED");
client.subscribe("PP");
}

void callback(char* topic, byte* payload, unsigned int length) {

  Serial.print("Mensagem recebida no topico: ");
  Serial.println(topic);

  Serial.print("Mensagem:");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
}

```

```

String msg;

//obtem a string do payload recebido
for(int i = 0; i < length; i++)
{
    char c = (char)payload[i];
    msg += c;
}

//toma ação dependendo da string recebida:
//verifica se deve colocar nível alto de tensão na saída.
//IMPORTANTE: o Led já contido na placa é acionado com lógica invertida (ou seja,
//enviar HIGH para o output faz o Led apagar / enviar LOW faz o Led acender)

//verifica se deve colocar nível alto de tensão na saída se enviar L e dígito, ou nível baixo se
enviar D e dígito no topico LED

    if (msg.equals("L1"))
    {
        digitalWrite(D1, HIGH);
        // EstadoSaida = '0';
    }
    if (msg.equals("D1"))
    {
        digitalWrite(D1, LOW);
        // EstadoSaida = '1';
    }
    if (msg.equals("L2"))
    {
        digitalWrite(D2, HIGH);
        // EstadoSaida = '0';
    }
    if (msg.equals("D2"))
    {
        digitalWrite(D2, LOW);
        // EstadoSaida = '1';
    }

    if (msg.equals("L3"))
    {
        digitalWrite(D3, HIGH);
        // EstadoSaida = '0';
    }
    if (msg.equals("D3"))
    {
        digitalWrite(D3, LOW);
        // EstadoSaida = '1';
    }

    if (msg.equals("L4"))
    {
        digitalWrite(D4, HIGH);
        // EstadoSaida = '0';
    }

```

```

if (msg.equals("D4"))
{
    digitalWrite(D4, LOW);
    // EstadoSaida = '1';
}

    if (msg.equals("L5"))
    {
        digitalWrite(D5, HIGH);
        // EstadoSaida = '0';
    }
if (msg.equals("D5"))
{
    digitalWrite(D5, LOW);
    // EstadoSaida = '1';
}
    if (msg.equals("L6"))
    {
        digitalWrite(D6, HIGH);
        // EstadoSaida = '0';
    }
if (msg.equals("D6"))
{
    digitalWrite(D6, LOW);
    // EstadoSaida = '1';
}
    if (msg.equals("L7"))
    {
        digitalWrite(D7, HIGH);
        // EstadoSaida = '0';
    }
if (msg.equals("D7"))
{
    digitalWrite(D7, LOW);
    // EstadoSaida = '1';
}

    if (msg.equals("L8"))
    {
        digitalWrite(D0, HIGH);
        // EstadoSaida = '0';
    }
if (msg.equals("D8"))
{
    digitalWrite(D0, LOW);
    // EstadoSaida = '1';
}

}

// Liga todos os leds se enviar LT no topico LED

if (msg.equals("LT"))
{
    digitalWrite(D0,HIGH);
    digitalWrite(D1,HIGH);
    digitalWrite(D2,HIGH);
    digitalWrite(D3,HIGH);
    digitalWrite(D4,HIGH);

```

```

    digitalWrite(D5,HIGH);
    digitalWrite(D6,HIGH);
    digitalWrite(D7,HIGH);
    digitalWrite(D8,HIGH);

}
// Desliga todos os leds se enviar DT no topico LED

if (msg.equals("DT"))
{
    digitalWrite(D0,LOW);
    digitalWrite(D1,LOW);
    digitalWrite(D2,LOW);
    digitalWrite(D3,LOW);
    digitalWrite(D4,LOW);
    digitalWrite(D5,LOW);
    digitalWrite(D6,LOW);
    digitalWrite(D7,LOW);
    digitalWrite(D8,LOW);
    delay(1000);

}
}

Serial.println();
Serial.println("-----");
}

void loop() {
    client.loop();
}

```