Roteiro Nº 08	**Protocolo MQTT**	
Curso Engenharia Elétrica	**Disciplina** ELT 1119 – Redes e Aplicações IoT	**Professor** Carlos Alberto Vasconcelos Bezerra
Nome do Estudante		**Data**

1. Objetivos da aula

- Compreender o funcionamento do protocolo MQTT no modelo publish/subscribe;
- Configurar o ESP32 para conectar-se a uma rede Wi-Fi e a um broker MQTT;
- Programar o ESP32 para acionar um LED com base em mensagens recebidas via MQTT;
- Utilizar um client MQTT no computador para publicar mensagens de controle;
- Realizar a integração prática entre hardware (ESP32 + LED) e software (broker + client).

2. Materiais utilizados

- 1 placa ESP32
- 1 LED externo
- 1 resistor de 220 Ω
- Jumpers
- 1 protoboard
- Cabo USB para comunicação e alimentação
- Computador com:
  - Python instalado
  - Thonny IDE
  - Acesso à internet
- Broker MQTT (público - test.mosquitto.org)

3. Resumo teórico

O protocolo MQTT (Message Queuing Telemetry Transport) é um protocolo de comunicação leve, baseado no modelo publish/subscribe (publicador/assinante). Ele é ideal para aplicações de Internet das Coisas (IoT), onde dispositivos precisam trocar informações de forma rápida e com baixo uso de banda.
No modelo MQTT:

- O publicador envia mensagens a um broker (servidor MQTT) em tópicos específicos.
- O assinante (subscriber) inscreve-se nesses tópicos para receber as mensagens.

No experimento, o ESP32 atua como assinante, recebendo comandos (ON ou OFF) enviados por um client MQTT no computador. Ao receber ON, o ESP32 acende o LED; ao receber OFF, o ESP32 apaga o LED.

4. Desenvolvimento da aula

Passo 1: montagem do circuito

- Conecte o cátodo (lado curto) do LED ao GND do ESP32.
- Conecte o ânodo (lado longo) ao resistor de 220Ω, e o outro lado do resistor no pino GPIO5 do ESP32.

Passo 2: configuração do ESP32

1. Grave o firmware MicroPython no ESP32 (caso ainda não tenha).
2. Abra o Thonny IDE e configure o interpretador para **MicroPython (ESP32)**.
3. Salve o código do ESP32 como main.py na memória da placa (código em anexo).

- Esse código conecta o ESP32 ao Wi-Fi, ao broker MQTT e inscreve-se no tópico casa/sala/led.

Passo 3: instalação das bibliotecas no computador

- No Prompt de Comando (CMD), instale a biblioteca paho-mqtt:

pip install paho-mqtt

Exemplo:

"C:\Program Files (x86)\Thonny\python.exe" -m pip install paho-mqtt
(Caminho onde está instalado o Thonny)

Essa biblioteca permite que o PC envie mensagens MQTT via script Python.

Passo 4: execução

1. Desconecte o Thonny da placa (o ESP32 continuará rodando main.py).
2. Rode o **client MQTT no computador** (código em anexo) para enviar ON ou OFF.
3. Observe o LED acender e apagar conforme os comandos enviados.

- O ESP32 imprimirá no console as mensagens recebidas.

5. Códigos utilizados (em anexo)

Código no ESP32 (main.py):

```
from machine import Pin
import network
import time
from umqtt.simple import MQTTClient

# Configuração da Wi-Fi
ssid = 'NOME_DA_REDE_WIFI'
password = 'SENHA_DA_REDE_WIFI'

# Conectando ao Wi-Fi
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.connect(ssid, password)

print('Conectando-se ao Wi-Fi...')
while not sta_if.isconnected():
    time.sleep(1)
print('Wi-Fi conectado:', sta_if.ifconfig())

# Configuração do LED (GPIO5)
led = Pin(5, Pin.OUT)

# Configuração do MQTT
broker = 'test.mosquitto.org'  # ou outro broker público/local
client_id = 'esp32_led_controller'
topic_sub = b'casa/sala/led'
```

```
def callback_mqtt(topic, msg):
    print('Mensagem recebida:', topic, msg)
    if msg == b'ON':
        led.on()
    elif msg == b'OFF':
        led.off()

client = MQTTClient(client_id, broker)
client.set_callback(callback_mqtt)
client.connect()
client.subscribe(topic_sub)

print('Aguardando mensagens no tópico', topic_sub)

try:
    while True:
        client.check_msg()
        time.sleep(1)
except KeyboardInterrupt:
    client.disconnect()
    print('Desconectado do MQTT')
```

Código no computador (client_mqtt.py):

```
import paho.mqtt.client as mqtt

broker = 'test.mosquitto.org'
port = 1883
topic = 'casa/sala/led'

client = mqtt.Client('cliente_pc')

client.connect(broker, port)

mensagem = 'ON'  # ou 'OFF' para apagar
client.publish(topic, mensagem)

print(f'Mensagem \"{mensagem}\" enviada ao tópico \"{topic}\"')

client.disconnect()
```

Bibliografia

ASCHER, D.; LUTZ, M. Aprendendo Python. Porto Alegre: Bookman, 2007.

MENEZES, N. N. C. Introdução à programação com Python. São Paulo: Ed. Novatec, 2014.

OLIVEIRA, S. Internet das coisas com ESP8266, Arduino e Raspberry Pi. São Paulo: Ed. Novatec, 2017.

OLIVEIRA, C. L. V. IoT com Micropython e NodeMCU. São Paulo: Ed. Novatec, 2022.