

Prática 07

MQTT

Objetivo

Estudar protocolo de comunicação MQTT

Procedimento

1) Adicione o exemplo Protocols -> MQTT -> TCP seguindo o procedimento adotado nas práticas anteriores. Faça o *build* do projeto. Este exemplo conecta à internet por uma rede wifi e a um broker mqtt para envio e recebimento de mensagens. Apenas monitore o programa após realizar as configurações 02 a 04.

2) Configure wifi no menuconfig (no powershell digite: `idf.py menuconfig`) em com as seguintes credenciais:

SSD: `sistemas_embarcados`

Senha: `embarcados2025`

3) Encontre no código e configure o broker mqtt.

```
esp_mqtt_client_config_t mqtt_cfg = {  
    .uri = "mqtt://username:senha@node02.mqttthub.com:1883",  
                                     // CONFIG_BROKER_URL: username, senha  
    .credentials.client_id = "ID",    // ID  
};
```

Substituir:

username, senha e id por: `g1device`, `g2device` ou `g3device`

4) No celular, baixe o aplicativo “MQTT dash” e configure conforme abaixo:

Name: ID

Address: `node02.mqttthub.com`

Port: 1883

User name: username

User password: senha

Cliente ID: ID



Centro Federal de Educação Tecnológica de Minas Gerais
Departamento de Engenharia Elétrica
Curso: Engenharia Elétrica
Disciplina: Sistemas Embarcados
Professores: Túlio Carvalho

Substitua username, senha e id por: g1app, g2app ou g3app

Projeto

Data de avaliação 26/11/2025

Valor 9,0 pts

- 1) Seguindo o procedimento anterior, utilize o mesmo programa das práticas anteriores. Neste exercício você irá controlar o brilho dos LEDs RGB utilizando um aplicativo via MQTT.
- 2) Adicione pasta extra de biblioteca para a programação wi-fi no cMakeList.txt na pasta raiz.

```
# (Not part of the boilerplate)
# This example uses an extra component for common functions such as Wi-Fi and Ethernet connection.
set(EXTRA_COMPONENT_DIRS $ENV{IDF_PATH}/examples/common_components/protocol_examples_common)
```

```
# (Not part of the boilerplate)
# This example uses an extra component for common functions such as Wi-Fi and Ethernet connection.
set(EXTRA_COMPONENT_DIRS
$ENV{IDF_PATH}/examples/common_components/protocol_examples_common)
```

- 3) Possivelmente o programa apresentará problemas de memória, pois, apenas 2 MB (de 4MB) são liberados para gravar o programa, nestes exemplos, conforme é possível ver em todos os programas que foram executados até o momento.

Mensagem sobre uso da memória flash no início dos programas:

```
I (534) spi_flash: detected chip: generic
I (537) spi_flash: flash io: dio
W (541) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
```

Mensagem de erro referente a memória flash:

```
FAILED: esp-idf/esptool.py/CMakeFiles/app_check_size D:/dados/RealTimeEmbeddedSystems/Projeto_embarcados_RT/build/esp-idf/esptool.py/CMakeFiles/app_check_size
cmd.exe /C "cd D:/dados/RealTimeEmbeddedSystems/Projeto_embarcados_RT/build/esp-idf/esptool.py && d:\Programas\ESP32\.espressif\python_env\idf5.1_py3.11_env\Scripts\python.exe D:/Programas/ESP32/esp-idf/components/partition_table/check_sizes.py --offset 0x8000 partition --type app D:/dados/RealTimeEmbeddedSystems/Projeto_embarcados_RT/build/partition_table/partition-table.bin D:/dados/RealTimeEmbeddedSystems/Projeto_embarcados_RT/build/main.bin"
Error: app partition is too small for binary main.bin size 0x113740:
- Part 'factory' 0/0 @ 0x10000 size 0x10000 (overflow 0x13740)
ninja: build stopped: subcommand failed.
```

- a. Adicione o arquivo partitions.csv. Este arquivo pode ser escrito em um bloco de notas. A única diferença em relação a configuração default é na última linha onde é aberto 3MB para escrita do código.

```
#name,type,subtype,offset,size
nvs,data,nvs,0x9000,0x6000
phy_init,data,phy,0xf000,0x1000
factory,app,factory,0x10000,0x300000
```

b. Faça alteração no menuconfig conforme figura abaixo:

```
(Top) → Partition Table → Partition Table
Espressif IoT D

( ) Single factory app, no OTA
( ) Single factory app (large), no OTA
( ) Factory app, two OTA definitions
(X) Custom partition table CSV

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)

(Top) → Partition Table
Espressif IoT Developer

Partition Table (Custom partition table CSV) --->
(partitions.csv) Custom partition CSV file
(0x8000) Offset of partition table
[*] Generate an MD5 checksum for the partition table

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

c. Altere também o tamanho da memória flash.

```
(Top)
Espressif IoT Developer

↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑
Security features --->
Application manager --->
Serial flasher config --->
Partition Table --->
Example Connection Configuration --->
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

