

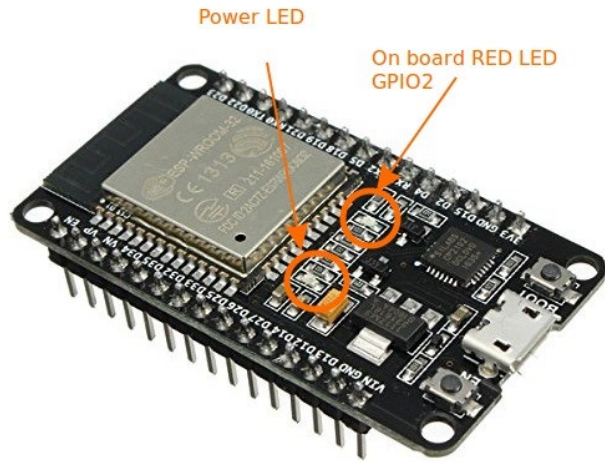
PSI3542 – 2023

SISTEMAS EMBARCADOS DISTRIBUIDOS

AULA 09 – ATIVIDADE 9.1 DISPOSITIVO IOT THINGSPEAK SUBSCRIBER

SERGIO TAKEO KOFUJI

KOFUJI@USP.BR



MQQT-SUB



MQQT-PUB

MQQT-PUB

API REST



ATIVIDADE 9.1

DISPOSITIVO IOT ATUADOR MQTT

Tarefas

- Implementar um canal no ThingSpeed com 1 campo e dois dispositivos MQTT:
 - Field3 – para controle do led da placa ESP32/ESP8266
 - MQTT01 – será o dispositivo ESP32/ESP8266
 - MQTT02 – será o cliente MQTTX, para controle de liga/desliga do led da placa ESP32/ESP8266

Programa main.py

Módulos

```
import network
```

```
import time
```

```
from machine import Pin
```

```
#from umqtt.simple import MQTTClient
```

```
#utilizaremos a versao “robusta” do cliente umqtt
```

```
from umqtt.robust import MQTTClient
```

Parâmetros de conexão

MQTT Server Parameters

MQTT_CLIENT_ID = "xxxxxx"

MQTT_BROKER = "mqtt3.thingspeak.com"

MQTT_USER = "xxxxxx"

MQTT_PASSWORD = "yyyyyy"

#<https://www.mathworks.com/help/thingspeak/subscribetoachannelfieldfeed.html>

#substituir zzzzzz pelo channel ID ThingSpeed

MQTT_SUB_TOPIC = "channels/zzzzz/subscribe/fields/field3"

#WIFI parameters

WIFI_SSID = "qqqqqq"

WIFI_PASSWD = "rrrrrr"

Subscribe Callback function

```
def callback(topic, msg):  
    print((topic, msg))  
    if msg == b'0':  
        led1.on()  
    else:  
        led1.off(  
            )
```


Configurar o pino de controle do LED

```
#vamos utilizar o led da placa ESP32/8266 (GPIO02)
```

```
led1=Pin(2, Pin.Out)
```

```
led1.on()
```

```
led1.off()
```

Conexão WiFi

#Create a WLAN network interface object. STA_IF = station aka client, connects to upstream WiFi access points

```
print("Connecting to WiFi", end="")
```

```
nic = network.WLAN(network.STA_IF)
```

```
nic.active(True)
```

```
nic.connect(WIFI_SSID, WIFI_PASSWD)
```

```
while not nic.isconnected():
```

```
    print(".", end="")
```

```
    time.sleep(0.1)
```

```
print(" Connected!")
```

Conexão com o broker mqtt thingspeak

```
print("Connecting to MQTT server... ", end="")
client = MQTTClient (MQTT_CLIENT_ID, MQTT_BROKER, user = MQTT_USER, password = MQTT_PASSWORD,
ssl=False)
# callback to handle data when MQTT channel updates
client.set_callback(callback)

try:
    client.connect()
except Exception as e:
    print('could not connect to MQTT server {}'.format(type(e).__name__, e))
    sys.exit()
print("Connected!")

client.subscribe(MQTT_SUB_TOPIC)
```

Loop de espera de mensagem mqtt sub

```
while True:
```

```
    try:
```

```
        client.wait_msg()
```

```
    except KeyboardInterrupt:
```

```
        print('Ctrl-C pressed...exiting')
```

```
        client.disconnect()
```

```
        sys.exit()
```

TESTE COM API REST

CURL E POSTMAN

CURL

- INSTALE A FERRAMENTA CURL NO COMPUTADOR
- PELO TERMINAL DE COMANDOS DO WINDOWS ENVIE COMANDOS “GET” AO THINGSPEAK PARA ACENDER E APAGAR O LED DA PLACA ESP32/8266

POSTMAN

- PELO POSTMAN ENVIE COMANDOS À PLACA PARA ACENDER E APAGAR O LED

MQTTX

MQTTX

- Configure o MQTTX – conecte ao dispositivo mqtt02 do ThingSpeed
 - Dispositivo **publisher** do **field3**
 - Tópico channels\channelid\publisher\Fields\field3
- Verifique o envio de dados ao ThingSpeed para acender e apagar o LED da placa ESP32/8266

Bom Trabalho

kofuji@usp.br