

# EVIDENCIAS PRACTICA 1

## PRIMER MÓDULO MQTT

### (PARTE 2)

Grupo:

Integrantes: David Flores y Diego García.

Fecha: 01/10/2024

#### Actividad 4: ESP32 como publicador

The screenshot displays the Arduino IDE environment with a sketch titled 'MQTT\_ESP32\_Publicador.ino'. The code is written in C++ and implements an MQTT publisher for an ESP32. It includes the necessary libraries, sets the server and port, and defines a loop that reads a temperature sensor (analogRead) and publishes the data to the 'test\_IoT/' topic. The serial monitor shows the device successfully connecting to the MQTT broker and publishing temperature readings of 8.00.

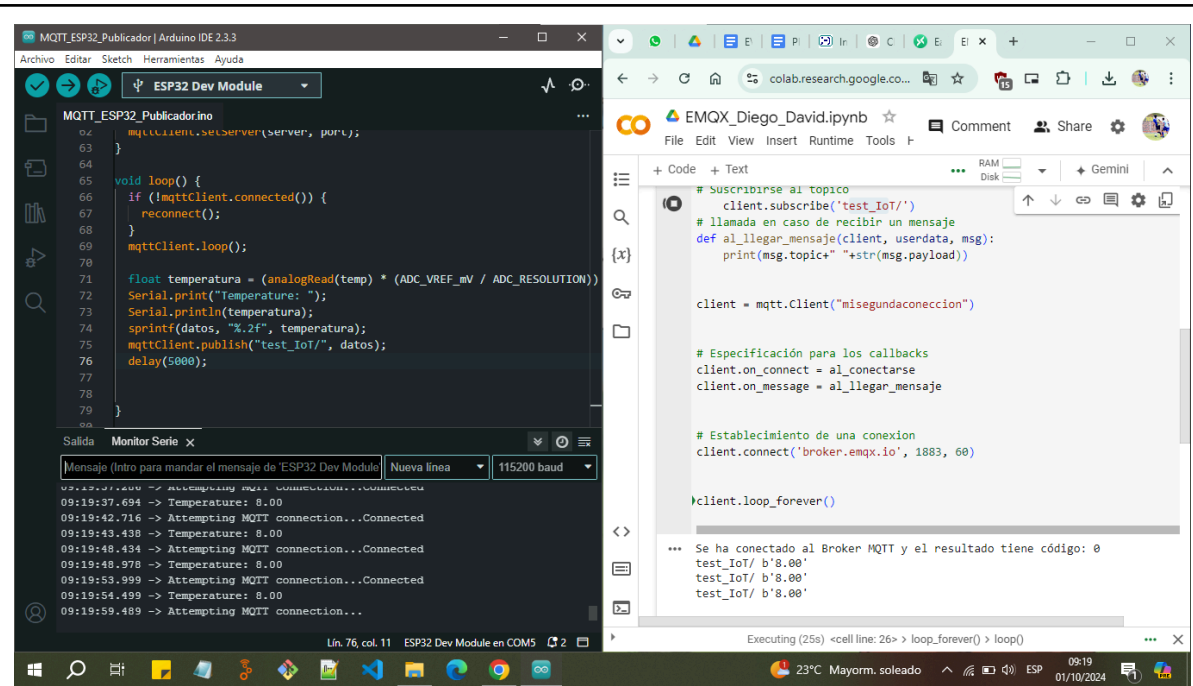
```
MQTT_ESP32_Publicador.ino
62 mqttClient.setServer(server, port);
63 }
64
65 void loop() {
66   if (!mqttClient.connected()) {
67     reconnect();
68   }
69   mqttClient.loop();
70
71   float temperatura = (analogRead(temp) * (ADC_VREF_mV / ADC_RESOLUTION));
72   Serial.print("Temperature: ");
73   Serial.println(temperatura);
74   sprintf(datos, "%.2f", temperatura);
75   mqttClient.publish("test_IoT/", datos);
76   delay(5000);
77 }
78
79 }
```

The serial monitor output shows the following sequence of events:

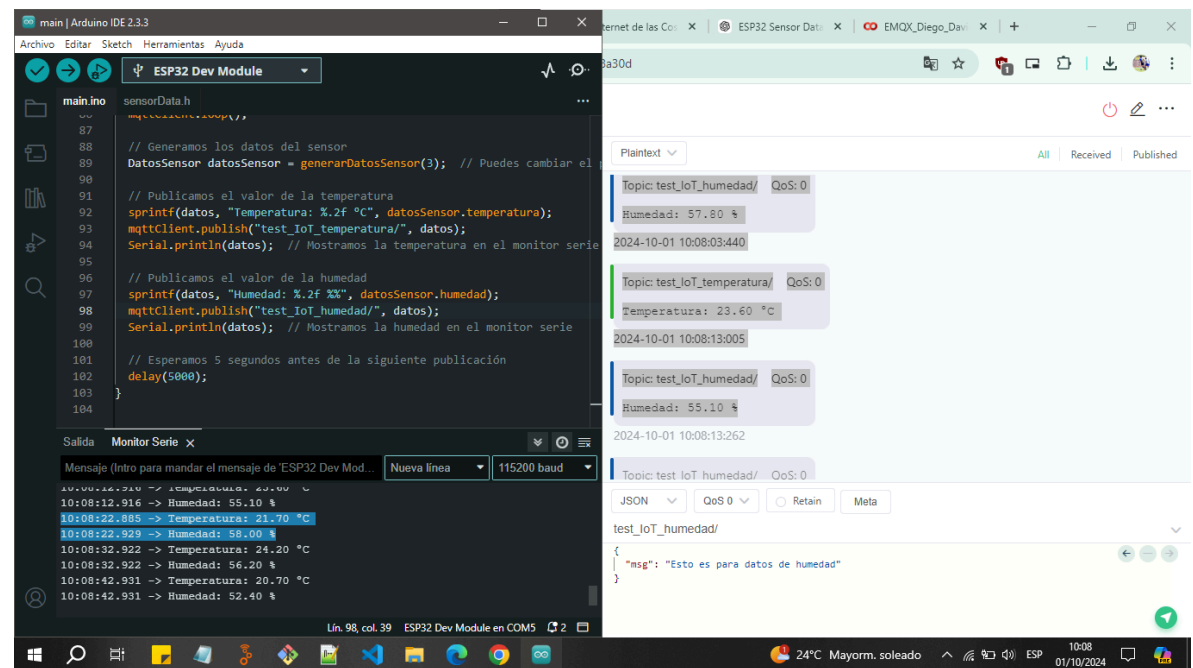
- 09:17:06.612 -> Attempting MQTT connection...Connected
- 09:17:11.541 -> Temperature: 8.00
- 09:17:12.332 -> Temperature: 8.00
- 09:17:17.342 -> Attempting MQTT connection...Connected
- 09:17:17.819 -> Attempting MQTT connection...Connected
- 09:17:18.228 -> Temperature: 8.00
- 09:17:23.210 -> Attempting MQTT connection...Connected
- 09:17:23.764 -> Temperature: 8.00

On the right, a MQTT client interface (likely Mosquitto) shows the received messages for the 'test\_IoT/' topic. The messages are in JSON format and contain the temperature data: {"msg": "8.00"}.

#### Actividad 4: ESP32 y Python



## Actividad de cierre: Evidencias etapa 1



## Actividad de cierre: Evidencias etapa 2

EMQX\_Diego\_David.ipynb - Co

colab.research.google.co...

EMQX\_Diego\_David.ipynb

File Edit View Insert Runtime Tools

+ Code + Text

```
client.on_connect = al_conectaranse
client.on_message = al_llegar_mensaje

# Establecimiento de una conexion
client.connect('broker.emqx.io', 1883, 60)

client.loop_forever()

...
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
test_IoT_temperatura/ b'Temperatura: 23.60 \xc2\xbc'
test_IoT_humedad/ b'Humedad: 50.40 %'
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
test_IoT_temperatura/ b'Temperatura: 24.40 \xc2\xbc'
test_IoT_humedad/ b'Humedad: 51.50 %'
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
Se ha conectado al Broker MQTT y el resultado tiene código: 0
test_IoT_temperatura/ b'Temperatura: 23.70 \xc2\xbc'
Se ha conectado al Broker MQTT y el resultado tiene código: 0
```

Executing (38s) <cell line: 27> > loop\_forever() > \_reconnect\_wait()

Internet de las Cosas: EV

ESP32 Sensor Data Inte

ba30d

Plaintext

All Received Published

2024-10-01 10:10:33443

Topic: test\_IoT\_humedad/ QoS: 0

Humedad: 51.50 %

2024-10-01 10:10:13464

Topic: test\_IoT\_humedad/ QoS: 0

Humedad: 52.70 %

2024-10-01 10:10:23333

Topic: test\_IoT\_humedad/ QoS: 0

Humedad: 52.70 %

2024-10-01 10:10:33362

JSON QoS 0 Retain Meta

test\_IoT\_humedad/

{ "msg": "Esto es para datos de humedad"

### Actividad de cierre: Evidencias etapa 3