

Vídeo de Kick-off de Gemelos
digitales con Wokwi para ESP32

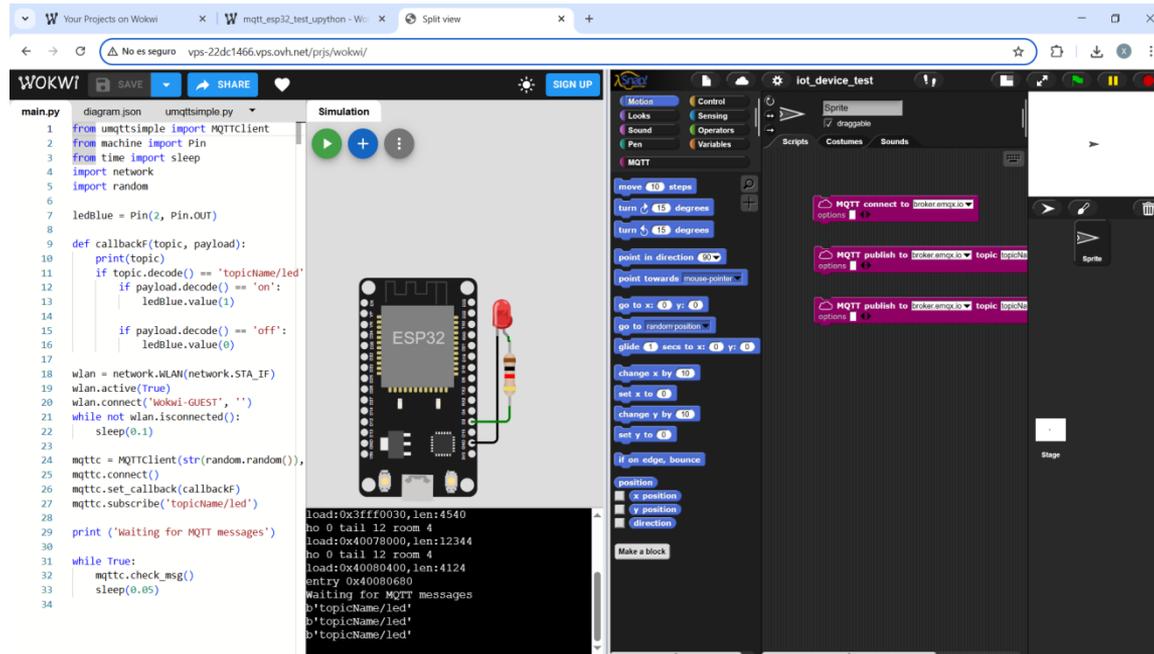
Acceso a Wokwi

www.wokwi.com



Ejemplo

<http://vps-22dc1466.vps.ovh.net/prjs/wokwi>



The screenshot displays the Wokwi web IDE interface. On the left, a Python script named `main.py` is shown, which configures an ESP32 to connect to a Wi-Fi network and publish MQTT messages to a topic named `led`. The script includes a `callbackF` function that toggles an LED based on the received MQTT payload. The central panel shows a 3D simulation of the ESP32 board with a red LED connected to pin 2. The right panel features a Scratch-style block-based code editor for a sprite, with a sequence of blocks including MQTT connection and publishing actions, movement blocks, and a loop.

```
1 from umqttsimple import MQTTClient
2 from machine import Pin
3 from time import sleep
4 import network
5 import random
6
7 ledBlue = Pin(2, Pin.OUT)
8
9 def callbackF(topic, payload):
10     print(topic)
11     if topic.decode() == 'topicName/led':
12         if payload.decode() == 'on':
13             ledBlue.value(1)
14
15         if payload.decode() == 'off':
16             ledBlue.value(0)
17
18 wlan = network.WLAN(network.STA_IF)
19 wlan.active(True)
20 wlan.connect('wokwi-GUEST', '')
21 while not wlan.isconnected():
22     sleep(0.1)
23
24 mqttc = MQTTClient(str(random.random()),
25 mqttc.connect()
26 mqttc.set_callback(callbackF)
27 mqttc.subscribe('topicName/led')
28
29 print('waiting for MQTT messages')
30
31 while True:
32     mqttc.check_msg()
33     sleep(0.05)
```

load:0x3fff0030, len:4540
no 0 tail 12 room 4
load:0x40078000, len:12344
no 0 tail 12 room 4
load:0x40080400, len:4124
entry 0x40080680
waiting for MQTT messages
b'topicName/led'
b'topicName/led'
b'topicName/led'