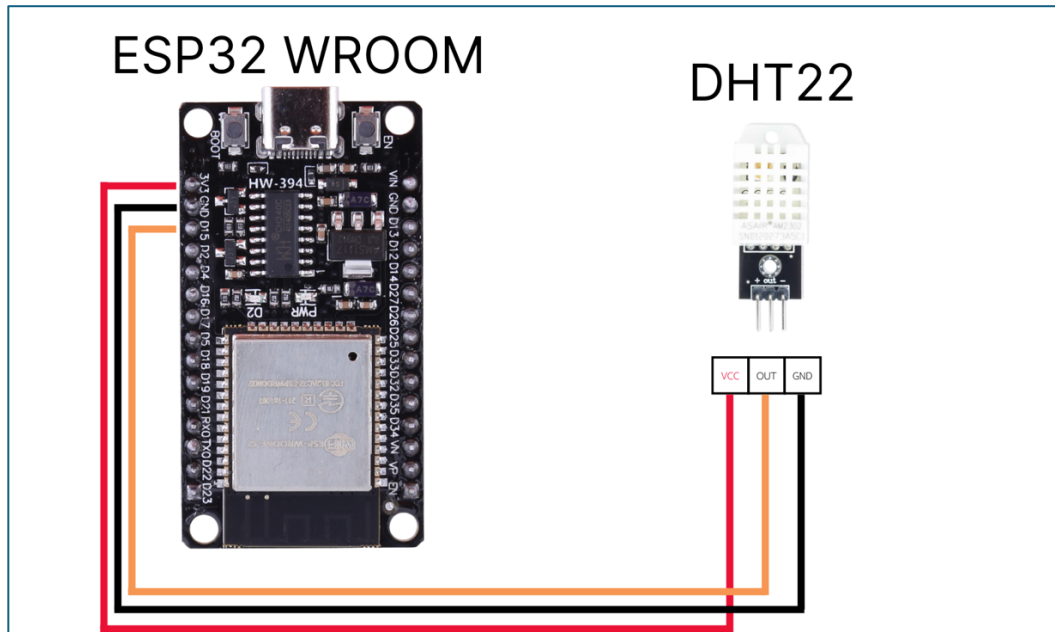
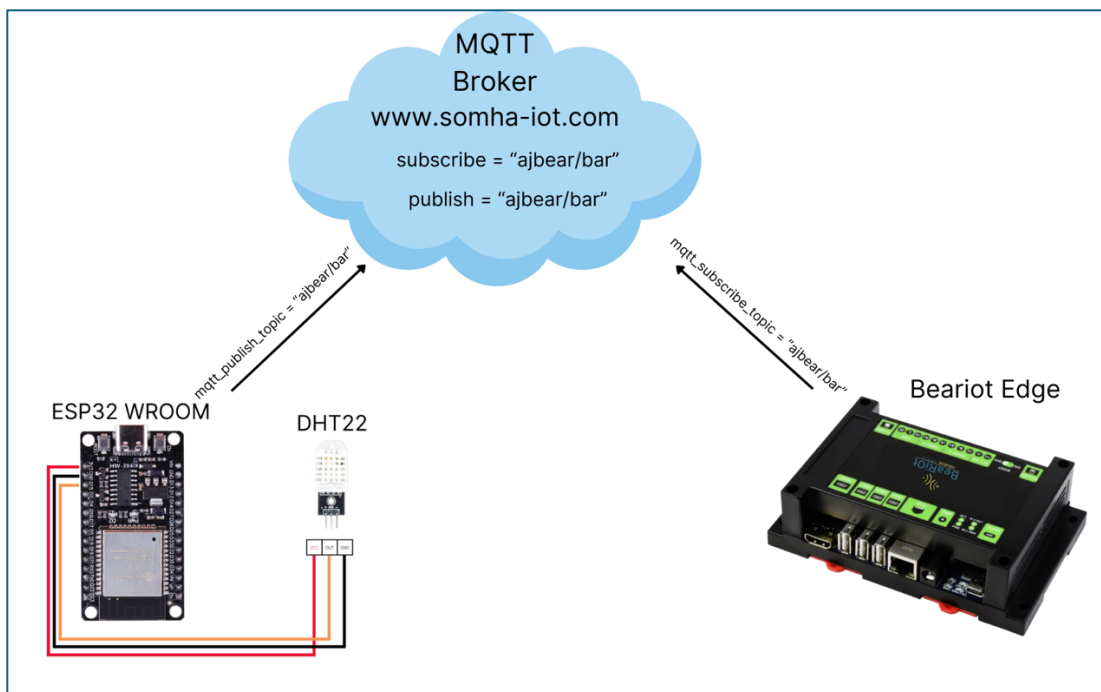


Smart Sensor ESP32 (MQTT)

การต่อวงจร ESP32 with DHT22



หลักการของ Smart Sensor ESP32 (MQTT)



โปรแกรม

สร้าง Payload เตรียมข้อมูลในการส่ง

```
# สร้าง Payload เตรียมส่งข้อมูล
def generate_payload(value):
    rtc = RTC()
    current_time = rtc.datetime()
    iso_format = '{:}-{:02d}-{:02d}T{:02d}:{:02d}:{:02d}.{:06d}+00:00'.format(
        current_time[0], current_time[1], current_time[2],
        current_time[4], current_time[5], current_time[6], current_time[7])
    print(iso_format)

    return {
        "siteID": SITE_ID,
        "deviceID": DEVICE_ID,
        "date": iso_format,
        "offset": -420,
        "connection": CONNECTION,
        "tagObj": [{
            "status": True,
            "label": LABEL,
            "value": value,
        }, {
            "status": True,
            "label": 'test',
            "value": 0,
        }]
    }
```

เชื่อมต่อ Wifi

```
# เชื่อมต่อ WIFI
def connect_wifi(ssid, password):
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)

    while not wlan.isconnected():
        time.sleep(1)
        print("Connecting to Wi-Fi...")
    print("Connected to Wi-Fi:", wlan.ifconfig())
```

เชื่อมต่อกับ MQTT Broker

```
# เชื่อมต่อ MQTT
def connect_mqtt():
    mqtt_client = MQTTClient(
        client_id=mqtt_client_id,
        server=mqtt_host,
        user=mqtt_username,
        password=mqtt_password
    )
    mqtt_client.connect()
    return mqtt_client
```

อ่านอุณหภูมิ DHT22

```
#อ่านค่าอุณหภูมิ
def read_temperature():
    sensor = dht.DHT22(Pin(15))
    sensor.measure()
    temp_value = sensor.temperature()
    return temp_value
```

Publish ข้อมูลไปยัง MQTT Broker

```
# ส่งข้อมูลไปยัง MQTT Broker
def publish_data(mqtt_client, topic, payload):
    payload_str = ujson.dumps(payload)
    print(payload_str)
    mqtt_client.publish(topic, payload_str.encode('utf-8'))
```

Function หลักในการเรียกใช้งาน

```
# ฟังก์ชันหลัก
def main():
    print("Starting BeaRiOt MQTT Test")
    print(f"Sending data to: {mqtt_host}")

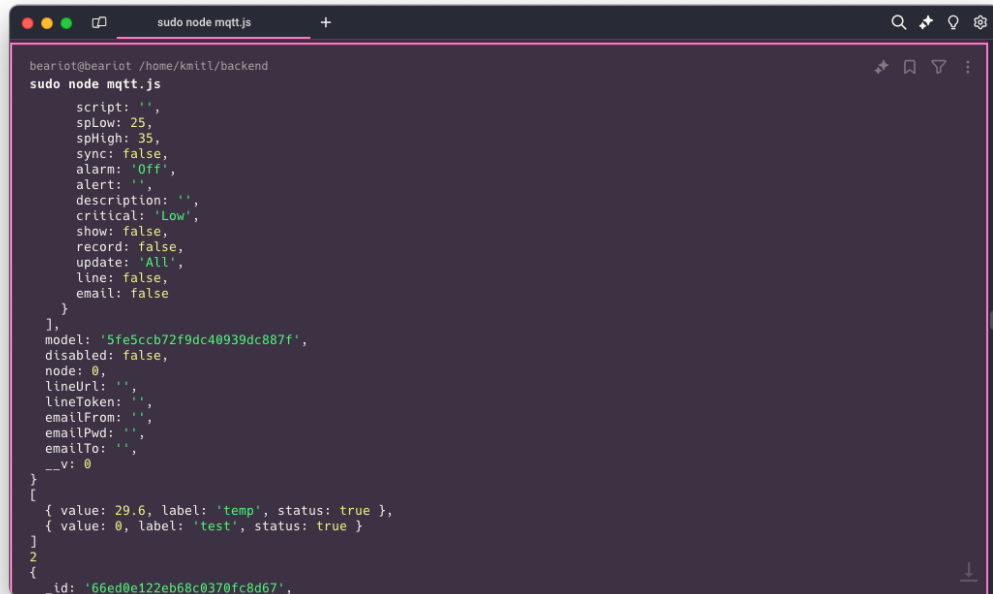
    connect_wifi(SSID, PASSWORD)
    mqtt_client = connect_mqtt()

    try:
        while True:
            temp_value = read_temperature()
            payload = generate_payload(temp_value)
            publish_data(mqtt_client, mqtt_publish_topic, payload)
            time.sleep(3)
    except KeyboardInterrupt:
        print("Test stopped by user")

if __name__ == "__main__":
    main()
```

วิธีการทดสอบ

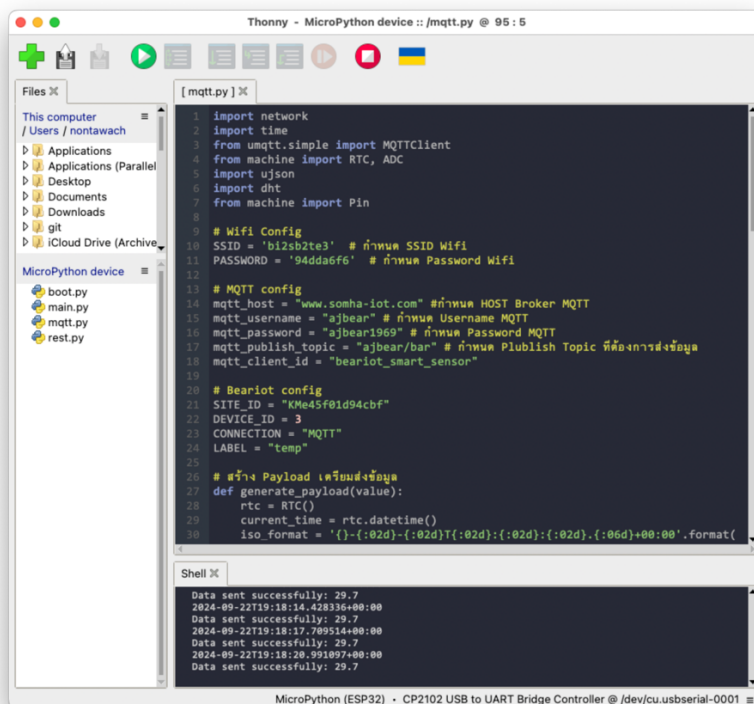
รันไฟล์ที่ Subscribe MQTT กับ MQTT Broker รันบน Beariot Edge ด้วย Node JS



```
sudo node mqtt.js

beariot@beariot /home/kmitl/backend
sudo node mqtt.js
script: '',
  spLow: 25,
  spHigh: 35,
  sync: false,
  alarm: 'Off',
  alert: '',
  description: '',
  critical: 'Low',
  show: false,
  record: false,
  update: 'All',
  line: false,
  email: false
},
model: '5fe5ccb72f9dc40939dc887f',
disabled: false,
node: 0,
lineUrl: '',
lineToken: '',
emailFrom: '',
emailPwd: '',
emailTo: '',
__v: 0
}
[
  { value: 29.6, label: 'temp', status: true },
  { value: 0, label: 'test', status: true }
]
2
{
  _id: '66ed0e122eb68c0370fc8d67',
```

รันโปรแกรม Smart Sensor MQTT บน ESP32

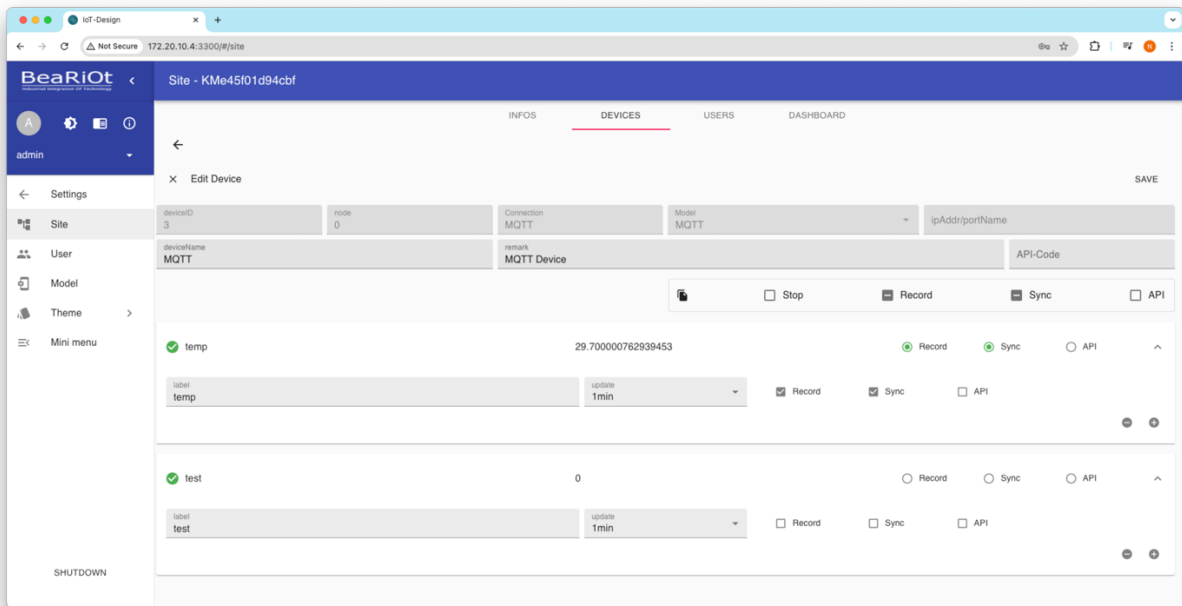


```
Thonny - MicroPython device :: /mqtt.py @ 95:5

1 import network
2 import time
3 from umqtt.simple import MQTTClient
4 from machine import RTC, ADC
5 import ujson
6 import dht
7 from machine import Pin
8
9 # Wifi Config
10 SSID = 'b12sb2te3' # กำหนด SSID Wifi
11 PASSWORD = '94dda6f6' # กำหนด Password Wifi
12
13 # MQTT config
14 mqtt_host = "www.somha-iot.com" #กำหนด HOST Broker MQTT
15 mqtt_username = "ajbear" # กำหนด Username MQTT
16 mqtt_password = "ajbear1969" # กำหนด Password MQTT
17 mqtt_publish_topic = "ajbear/bar" # กำหนด Publish Topic ที่ต้องการส่งข้อมูล
18 mqtt_client_id = "beariot_smart_sensor"
19
20 # Beariot config
21 SITE_ID = "KMe45f01d94cbf"
22 DEVICE_ID = 3
23 CONNECTION = "MQTT"
24 LABEL = "temp"
25
26 # สร้าง Payload เพื่อส่งข้อมูล
27 def generate_payload(value):
28     rtc = RTC()
29     current_time = rtc.datetime()
30     iso_format = '{:}-{:02d}-{:02d}T{:02d}:{:02d}:{:02d}.{:06d}+00:00'.format(
31         current_time[0], current_time[1], current_time[2],
32         current_time[3], current_time[4], current_time[5], value)
33     return iso_format

Data sent successfully: 29.7
2024-09-22T19:18:14.428336+00:00
Data sent successfully: 29.7
2024-09-22T19:18:17.789514+00:00
Data sent successfully: 29.7
2024-09-22T19:18:20.991097+00:00
Data sent successfully: 29.7
```

สร้าง Model Connection MQTT ขึ้นมารับข้อมูลที่ส่งมายัง Beariot Edge และตั้งชื่อ Label ให้ตรงกันกับบนโปรแกรม



ผลลัพธ์

