

## Manual de uso del código de ESP32

A continuación, encontrará el código que conecta un ESP32 a AWS IoT Core y publica datos de presión, altitud, BPM, temperatura y humedad por medio de MQTT. También se puede suscribir a un tema para recibir mensajes de AWS IoT Core si se desea programar un actuador. Para usar el código, siga estos pasos:

1. Instale las bibliotecas necesarias:

- WiFiClientSecure
- PubSubClient
- ArduinoJson
- DHT sensor library
- Adafruit BMP085 Library
- PulseSensorPlayground

Es recomendable instalar todas sus dependencias para evitar inconvenientes de compatibilidad.

2. Cree un archivo llamado Secrets.h y agregue las siguientes credenciales

- Desde su acceso a internet:
  - i. WIFI\_SSID
  - ii. WIFI\_PASSWORD
- Desde su portal de AWS IoT:
  - i. AWS\_IOT\_ENDPOINT
  - ii. AWS\_CERT\_CRT
  - iii. AWS\_CERT\_PRIVATE
  - iv. THINGNAME

3. Abra el código y modifique las siguientes variables según sea necesario:

- a. Pin donde se conecta el sensor DHT11
  - #define DHTPIN 15

4. Cargue el código en el ESP32.

5. Abra el AWS IoT Core Console y navegue hasta el menú Things.

6. En la sección MQTT Test Client, haga clic en Connect.

7. En el campo Topic, escriba el tema al que se suscribirá el ESP32:

- esp32/pub

8. Haga clic en Subscribe.

9. En el monitor serial, verá un mensaje que indica que se ha conectado a AWS IoT Core.
10. Cada segundo, el ESP32 publicará un mensaje con los datos de temperatura y humedad en el tema.
11. En el AWS IoT Core Console, verá los mensajes publicados por el ESP32 en la sección Incoming.

## Uso del ESP32 para suscribirse a mensajes de AWS IoT Core

Para suscribirse a mensajes de AWS IoT Core, el ESP32 debe estar conectado a AWS IoT Core y suscribirse al tema del que desea recibir mensajes.

Para suscribirse a un tema, use el siguiente código:

```
client.subscribe("<TOPIC>");
```

Donde **<TOPIC>** es el tema del que desea recibir mensajes, en este caso AWS\_IOT\_SUBSCRIBE\_TOPIC.

Cuando el ESP32 reciba un mensaje en el tema suscrito, llamará a la función **messageHandler()**.

El siguiente código muestra un ejemplo de una función **messageHandler()** que imprime el mensaje recibido en la consola serial:

```
Serial.print("incoming: ");  
Serial.println(topic);  
StaticJsonDocument doc;  
deserializeJson(doc, payload);  
const char* message = doc["message"];  
Serial.println(message);
```

## Publicación de datos AWS IoT Core

Para publicar datos de presión, altitud, BPM, temperatura y humedad en AWS IoT Core, el ESP32 debe estar conectado a AWS IoT Core y publicar los datos en un topic, en este caso es el tema **esp32/pub**.

Para publicar datos se utiliza una función denominada **publishMessage()**, la cual contiene el siguiente código:

```
StaticJsonDocument<400> doc;

doc["humidity"] = h;
doc["temperature"] = t;
doc["altitude"] = a;
doc["pressure"] = p;
doc["BPM"] = myBPM;

char jsonBuffer[512];
serializeJson(doc, jsonBuffer); // print to client

client.publish(AWS_IOT_PUBLISH_TOPIC, jsonBuffer);
```

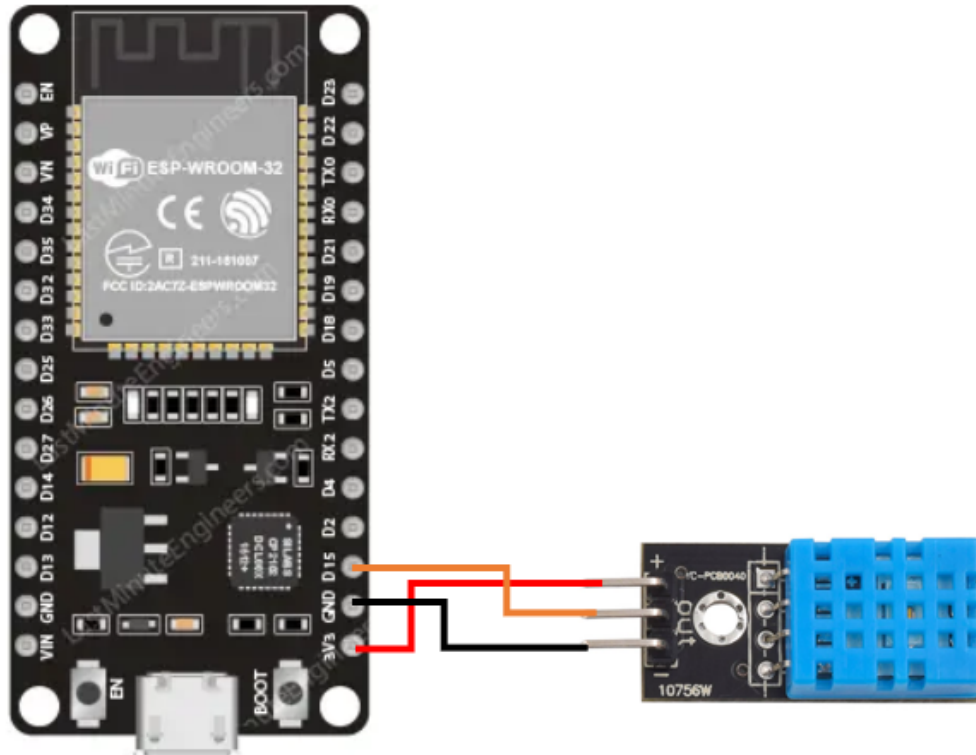
## Recolección de Datos

Para llevar a cabo este proceso, en la función **loop()**, se llaman las funciones propias de cada sensor y se asignan a las variables globales declaradas en un principio del código. Posteriormente, se determina si todas las variables fueron recolectadas de forma correcta por medio de varias condicionales e imprime el resultado de estas ya sea positivo o negativo. Por último, se llama la función **publishMessage()** para llevar a cabo el envío de los datos y así visualizar estos en la sección de MQTT en AWS IoT Core. A continuación, encontrará las funciones utilizadas para la recolección de datos:

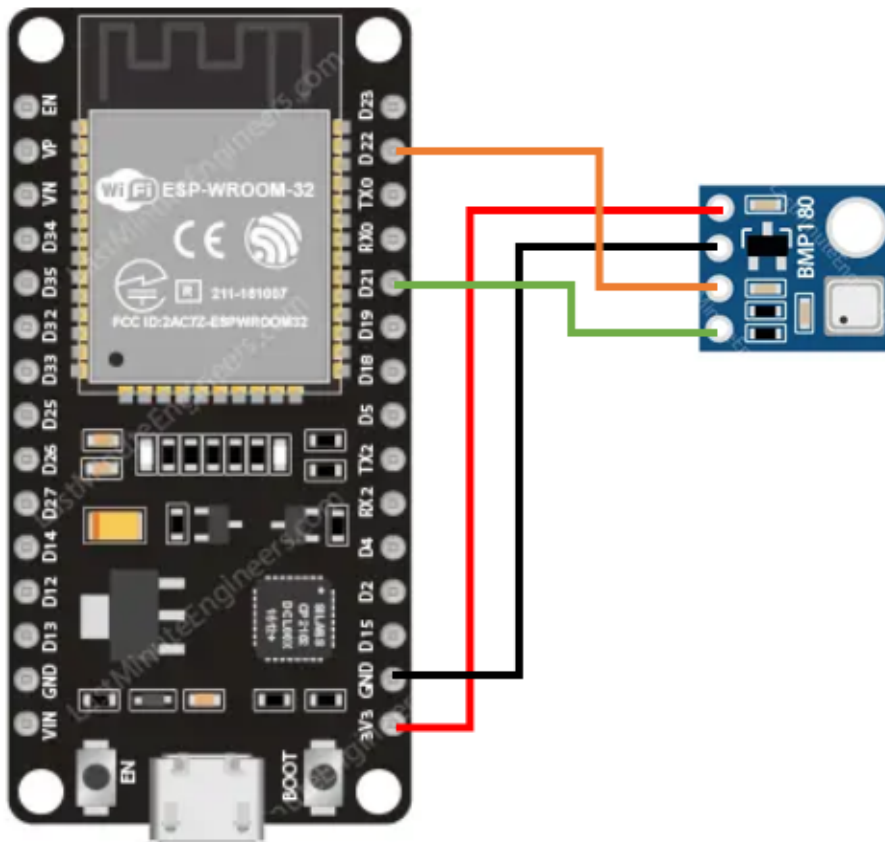
1. Sensor DHT11
  - a. Humedad: `dht.readHumidity()`
  - b. Temperatura: `dht.readTemperature()`
2. Sensor BMP
  - a. Temperatura: `bmp.readTemperature()`
  - b. Presión: `bmp.readPressure()`
  - c. Altitud: `bmp.readAltitude()`
3. Sensor de Pulso Cardíaco
  - a. Pulsos Por Minuto: `pulseSesor.getBeatsPerMinute()`

## CONEXIONES REALIZADAS

SENSOR DHT11



SENSOR BMP180



## SENSOR RITMO CARDIACO

