

## Programa Sensor de Presión

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>
#include <HX711.h>
#include <Wire.h>
#include <SH1106Wire.h> // Librería para la pantalla OLED SH1106

// **Configuración de la red WiFi**
const char *ssid = "Familia Baretta* 2.4GHz"; // Reemplaza con el
nombre de tu red WiFi
const char *password = "jesusesvida"; // Reemplaza con la
contraseña de tu red WiFi

// **Configuración de Adafruit IO**
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883 // Puerto MQTT para
Adafruit IO
#define AIO_USERNAME "Charly76" // Reemplaza con tu
nombre de usuario de Adafruit IO
#define AIO_KEY "aio_peET13z39trG2TzxrBPVYNzC0qm9" //
Reemplaza con tu llave de Adafruit IO

WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);

// **Configuración de los feeds de Adafruit IO**
Adafruit_MQTT_Publish pressureFeed = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/presion");

// **Configuración de ThingSpeak**
const char* THINGSPEAK_API_KEY = "XHHZDQFO5NNW6ZN4"; // Reemplaza con
tu Write API Key de ThingSpeak
const char* THINGSPEAK_SERVER = "api.thingspeak.com";
const int THINGSPEAK_PORT = 80;

// **Pines del HX711**
#define DT_PIN D2 // OUT del módulo HX711 (pin de datos)
#define SCK_PIN D1 // SCK del módulo HX711 (pin de reloj)
HX711 scale;
```

```

// **Factor de calibración**
float calibration_factor = 0.001; // Ajusta este valor según tus
pruebas
float pressure_MCA = 0.0;          // Variable para almacenar la presión
en MCA

// **Configuración de la pantalla OLED SH1106**
SH1106Wire display(0x3C, D3, D5); // Dirección I2C 0x3C, pines SDA y
SCL

// **Función para conectarse a la red WiFi**
void connectWiFi() {
    Serial.print("Conectando a ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("\nConectado a WiFi");
    Serial.print("Dirección IP: ");
    Serial.println(WiFi.localIP());
}

// **Función para conectar al servidor de Adafruit IO**
void MQTT_connect() {
    int8_t ret;

    if (mqtt.connected()) {
        return;
    }

    Serial.print("Conectando al servidor de Adafruit IO...");

    while ((ret = mqtt.connect()) != 0) {
        Serial.println(mqtt.connectErrorString(ret));
        Serial.println("Reintentando conexión en 5 segundos...");
        delay(5000);
    }
}

```

```

    Serial.println("Conectado a Adafruit IO!");
}

// **Función para enviar datos a ThingSpeak**
void sendToThingSpeak(float pressure) {
    if (WiFi.status() == WL_CONNECTED) {
        WiFiClient client;
        HTTPClient http;

        String url = "http://" + String(THINGSPEAK_SERVER) +
"/update?api_key=" + THINGSPEAK_API_KEY +
        "&field4=" + String(pressure);

        http.begin(client, url);
        int httpCode = http.GET();

        if (httpCode > 0) {
            Serial.print("Datos enviados a ThingSpeak. Código de respuesta:
");
            Serial.println(httpCode);
        } else {
            Serial.print("Error al enviar datos a ThingSpeak: ");
            Serial.println(http.errorToString(httpCode).c_str());
        }

        http.end();
    } else {
        Serial.println("Error: No conectado a WiFi");
    }
}

// **Función para leer la presión del sensor**
void readPressure() {
    float raw_units = scale.get_units(); // Obtiene las unidades
ajustadas

    // Convertir las unidades a presión en BAR
    pressure_MCA = raw_units * calibration_factor * 0.00010197 - 0.08;

    Serial.print("Presión medida: ");
    Serial.print(pressure_MCA, 2); // Mostrar con dos decimales
    Serial.println(" BAR");
}

```

```

// **Mostrar el valor en la pantalla OLED**
display.clear();
display.drawRect(0, 0, 128, 60);
display.setFont(ArialMT_Plain_24);
display.setTextAlignment(TEXT_ALIGN_LEFT);
display.drawString(10, 0, "Presión:");
display.drawString(10, 20, String(pressure_MCA, 2) + " BAR");
display.display();
}

void setup() {
  Serial.begin(115200);
  scale.begin(DT_PIN, SCK_PIN); // Inicializar el HX711
  connectWiFi();
  MQTT_connect();

  // **Inicializar la pantalla OLED**
  display.init();
  display.flipScreenVertically();
  display.clear();
  display.display();
  display.setFont(ArialMT_Plain_10);
  display.drawString(0, 0, "Inicializando...");
  display.display();
}

void loop() {
  // **Asegurarse de que la conexión a Adafruit IO esté activa**
  MQTT_connect();

  // **Leer presión**
  readPressure();

  // **Publicar la presión en Adafruit IO**
  if (!pressureFeed.publish(pressure_MCA)) {
    Serial.println("Error publicando la presión en Adafruit IO");
  } else {
    Serial.print("Presión publicada en Adafruit IO: ");
    Serial.print(pressure_MCA);
    Serial.println(" BAR");
  }

  // **Enviar datos a ThingSpeak**

```

```
sendToThingSpeak(pressure_MCA);

// **Esperar 10 segundos antes de la próxima lectura**
delay(10000);
}
```