**Guía para Subir Datos de Temperatura y Humedad a Ubidots Usando Arduino y ESP32**

**1. Objetivo**

Configurar un sistema para monitorear la temperatura y la humedad de una huerta en tiempo real, utilizando un sensor DHT conectado a un Arduino para recolectar datos, y un ESP32 para transmitirlos a la plataforma Ubidots.

**2. Materiales Necesarios**

1. **Sensor DHT11 o DHT22**: Sensor para medir temperatura y humedad.
2. **Arduino Uno o Nano**: Recolectará los datos del sensor.
3. **ESP32**: Enviará los datos a la red Wi-Fi y luego a Ubidots.
4. **Cables Jumper**: Para realizar las conexiones entre el sensor, el Arduino, y el ESP32.
5. **Cuenta en Ubidots**: Si no tienes una, regístrate en [Ubidots](https://ubidots.com).

**3. Configuración de Ubidots**

1. **Crear un Dispositivo**:
   * Inicia sesión en Ubidots y crea un nuevo dispositivo llamado "Huerta".
   * Agrega dos variables al dispositivo: "Temperatura" y "Humedad".
2. **Obtener el Token de API**:
   * Ve a tu perfil y copia el **token de API**. Lo necesitarás para autenticar la conexión desde el ESP32.

**4. Conexión del Sensor al Arduino**

1. Conecta el **sensor DHT** al Arduino:
   * **Pin de señal (OUT)** del sensor al **Pin digital 2** del Arduino.
   * **VCC** del sensor a **5V** en el Arduino.
   * **GND** del sensor a **GND** en el Arduino.
2. Conecta el **Arduino** al **ESP32**:
   * **TX del Arduino** al **RX del ESP32**.
   * **RX del Arduino** al **TX del ESP32**.
   * Asegúrate de conectar los **GND** de ambos dispositivos.

**5. Código para el Arduino**

El Arduino lee los datos del sensor DHT y los envía al ESP32 por el puerto serie.

// ----- Configuración del Sensor DHT -----

#include <DHT.h>

#define DHTPIN 2 // Pin donde está conectado el DHT

#define DHTTYPE DHT22 // Cambia a DHT11 si usas ese modelo

DHT dht(DHTPIN, DHTTYPE); // Inicialización del sensor DHT

// ----- Configuración de la Comunicación Serial -----

void setup() {

Serial.begin(9600); // Inicia la comunicación serial con el ESP32

dht.begin(); // Inicia el sensor DHT

}

// ----- Lectura y Envío de Datos -----

void loop() {

// Lee la temperatura y la humedad

float temperatura = dht.readTemperature();

float humedad = dht.readHumidity();

// Verifica si la lectura es válida

if (isnan(temperatura) || isnan(humedad)) {

Serial.println("Error al leer el sensor DHT"); // Muestra error si falla la lectura

} else {

// Enviar datos al ESP32

Serial.print(temperatura);

Serial.print(",");

Serial.println(humedad);

}

delay(2000); // Lee cada 2 segundos

}

**6. Código para el ESP32**

El ESP32 recibe los datos desde el Arduino y los envía a Ubidots a través de la conexión Wi-Fi.

**Librerías Necesarias**

* **WiFi.h** (incluida en el IDE de Arduino para el ESP32).
* **HTTPClient.h** para manejar las solicitudes HTTP.

**Código del ESP32**

Asegúrate de reemplazar "TU\_SSID", "TU\_PASSWORD" y "TU\_TOKEN" con tus credenciales de red Wi-Fi y token de Ubidots.

// ----- Configuración de Red y Librerías -----

#include <WiFi.h>

#include <HTTPClient.h>

const char\* ssid = "TU\_SSID"; // Nombre de tu red Wi-Fi

const char\* password = "TU\_PASSWORD"; // Contraseña de tu red Wi-Fi

const char\* ubidotsToken = "TU\_TOKEN"; // Token de Ubidots

const char\* device = "Huerta"; // Nombre del dispositivo en Ubidots

const char\* variableTemp = "Temperatura"; // Nombre de la variable de temperatura

const char\* variableHum = "Humedad"; // Nombre de la variable de humedad

String serverName = "http://industrial.api.ubidots.com/api/v1.6/devices/";

// ----- Configuración de Conexión Wi-Fi -----

void setup() {

Serial.begin(9600);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Conectando a WiFi...");

}

Serial.println("Conectado a WiFi");

}

// ----- Lectura de Datos Seriales y Envío a Ubidots -----

void loop() {

// Verifica si hay datos disponibles en el puerto serie

if (Serial.available()) {

// Lee los datos del Arduino (temperatura, humedad)

String data = Serial.readStringUntil('\n');

int commaIndex = data.indexOf(',');

float temperatura = data.substring(0, commaIndex).toFloat();

float humedad = data.substring(commaIndex + 1).toFloat();

// Verifica la conexión Wi-Fi antes de enviar los datos

if (WiFi.status() == WL\_CONNECTED) {

HTTPClient http;

// ----- Envío de Temperatura -----

String urlTemp = serverName + device + "/" + variableTemp + "/values";

http.begin(urlTemp.c\_str());

http.addHeader("Content-Type", "application/json");

http.addHeader("X-Auth-Token", ubidotsToken);

String payloadTemp = "{\"value\": " + String(temperatura) + "}";

int httpResponseCode = http.POST(payloadTemp);

Serial.println("Enviando temperatura a Ubidots: " + String(httpResponseCode));

http.end(); // Cierra la conexión

// ----- Envío de Humedad -----

String urlHum = serverName + device + "/" + variableHum + "/values";

http.begin(urlHum.c\_str());

http.addHeader("Content-Type", "application/json");

http.addHeader("X-Auth-Token", ubidotsToken);

String payloadHum = "{\"value\": " + String(humedad) + "}";

httpResponseCode = http.POST(payloadHum);

Serial.println("Enviando humedad a Ubidots: " + String(httpResponseCode));

http.end(); // Cierra la conexión

} else {

Serial.println("Error de conexión Wi-Fi");

}

}

delay(2000); // Envía los datos cada 2 segundos

}

**7. Explicación del Código**

* **Arduino**:
  + Lee la temperatura y humedad del sensor DHT y las envía por el puerto serie al ESP32 en el formato: temperatura,humedad.
* **ESP32**:
  + Conecta el ESP32 a Wi-Fi y escucha los datos del puerto serie enviados por el Arduino.
  + Recibe los valores de temperatura y humedad, y luego envía cada valor a Ubidots en una solicitud POST.
  + **Ejemplo de Solicitud HTTP**: Para enviar los datos a Ubidots, el ESP32 crea una solicitud HTTP en formato JSON con el valor de la variable y el token de API.

**8. Verificación de Datos en Ubidots**

1. Ve a tu cuenta de Ubidots y accede al dispositivo "Huerta".
2. Deberías ver los datos de temperatura y humedad actualizándose en tiempo real en el dashboard.
3. Puedes crear gráficos o widgets para visualizar las variables.