

# Monitoreo de temperatura en tiempo real

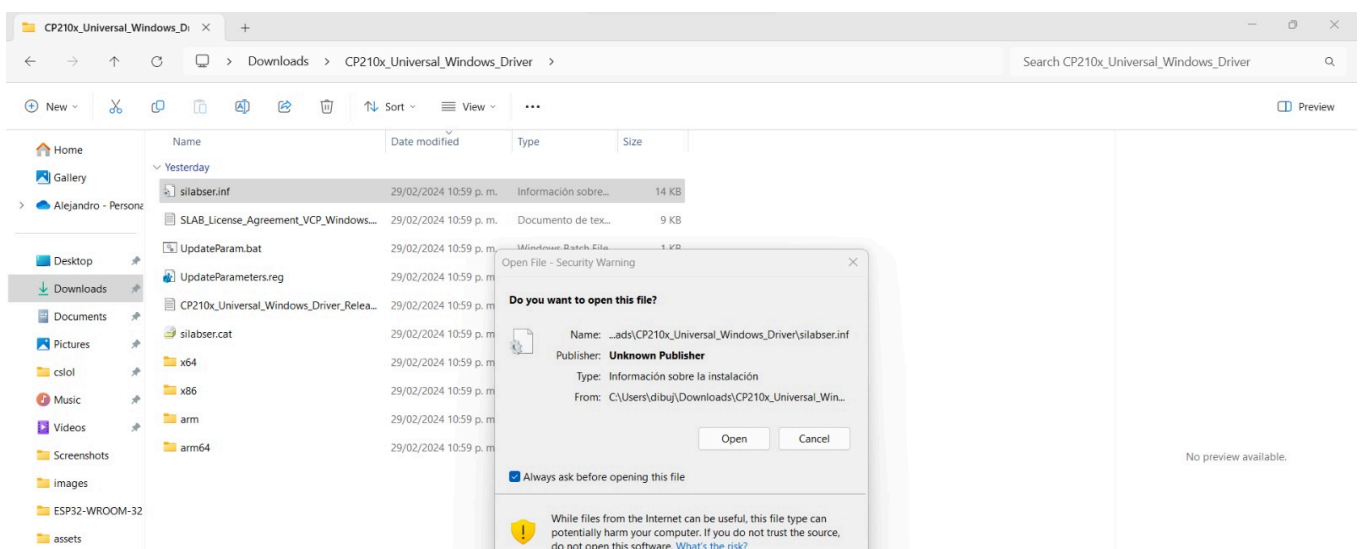
**Objetivo:** Implementar un sistema IoT simple utilizando el sensor de temperatura DHT 11 y la plataforma ThingSpeak para monitorear y visualizar datos de temperatura en tiempo real usando la placa ESP32.

## Materiales Necesarios:

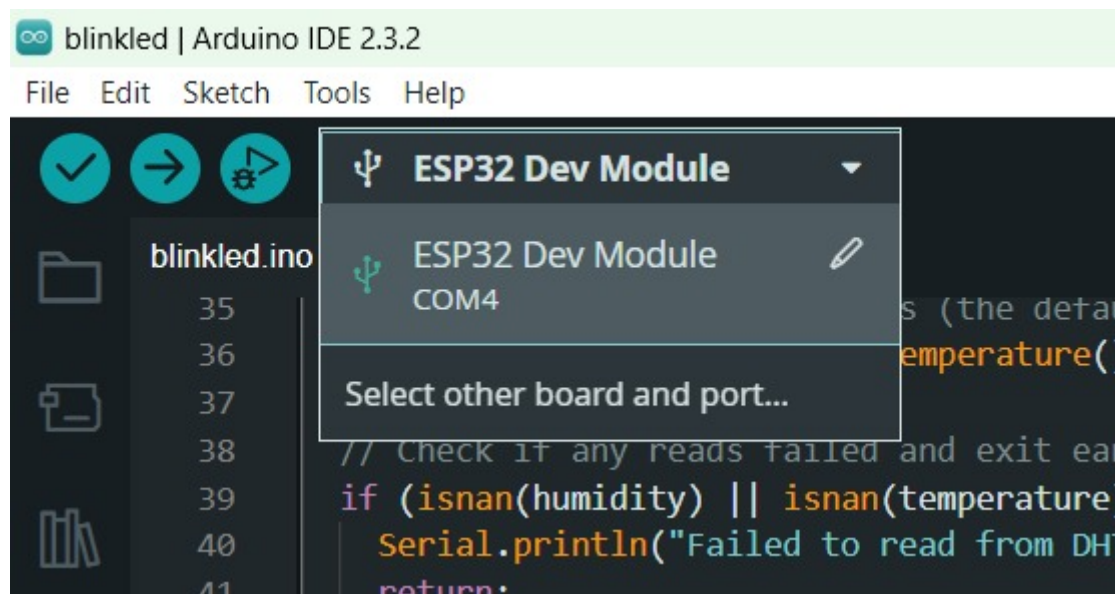
- Placa de desarrollo ESP32.
- Sensor de temperatura DHT11.
- Cables de conexión.
- Placa de prototipado.
- Conexión a Internet.

## Pasos realizados:

1. Visitamos la página Silicon Labs y descargamos los controladores CP210x USB to UART Bridge VCP

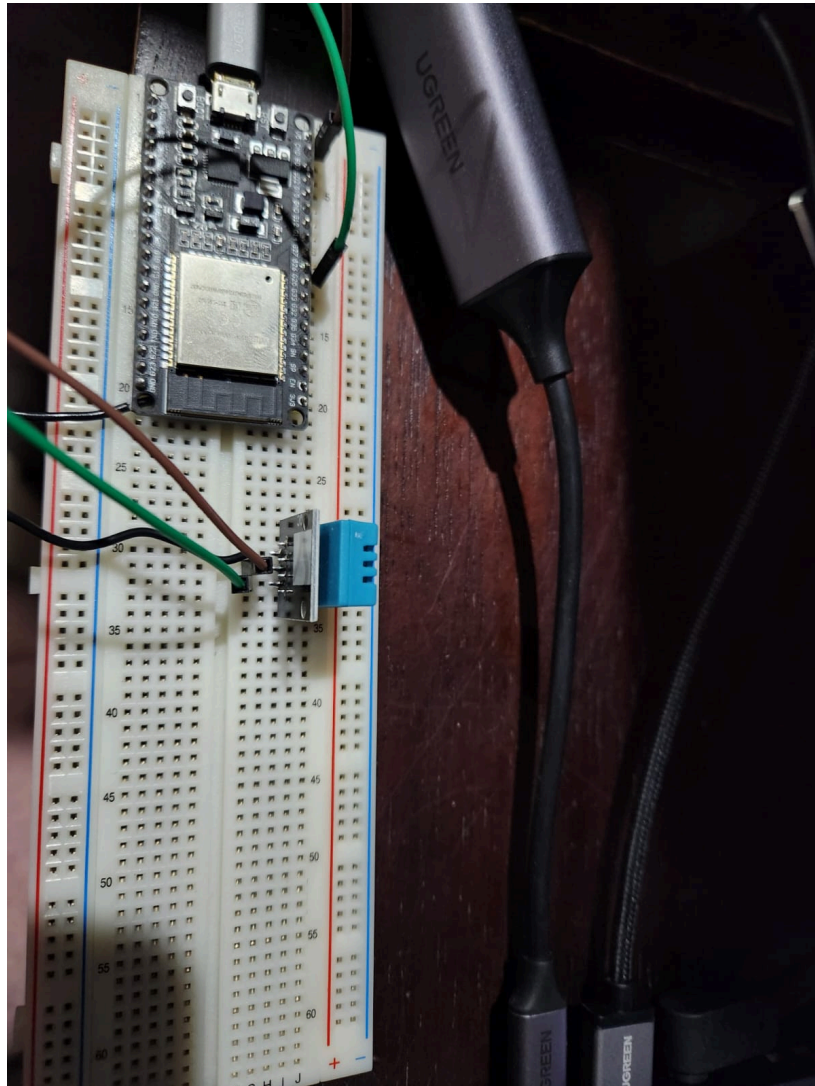


2. Seleccionamos el puerto de serie adecuado (En este caso fue el COM4)
3. Una vez seleccionado el puerto y verificar la correcta instalación de los drivers, se procede a instalar la tarjeta correspondiente dentro del IDE de Arduino.



4. Entramos a la página de MathWorks
5. Procedemos a hacer la conexión de la placa (ESP32) junto con el sensor de temperatura (Utilizamos el DHT11) en nuestra protoboard, conectamos el PIN 26 de la ESP32 al sensor, así como la entrada de

3.3v y la salida GND (tierra).



6. Una vez conectados todos los componentes anteriormente mencionados, se ingresa a la página de ThingSpeak y se crea el canal correspondiente para el monitoreo, junto con las etiquetas "Temperatura" y "Humedad".

## Practica 1

Channel ID: **2452531**

Author: **mwa0000033061781**

Access: Public

| Temperatura y Humedad

7. Cuando se crea el canal, nos proporciona la llave de la API correspondiente para realizar la conexión al canal, la cuál se ingresa en el código.
8. Una vez corroborada la funcionalidad de la placa, se coloca el siguiente código

```
#include <WiFi.h>

#include <ThingSpeak.h>

#include <DHT.h>

#define DHTPIN 26 // The pin you connected the DHT's data pin
to

#define DHTTYPE DHT11 // DHT 11

#define LED 32

// Replace with your WiFi credentials
const char* ssid = "INFINITUM9C9C_2.4";
const char* password = "9XE5GZJ8Sw";

// ThingSpeak Settings
unsigned long myChannelNumber = 2452531;
const char * myWriteAPIKey = "4G8EB5IOSQZCKCE4";

WiFiClient  client;

DHT dht(DHTPIN, DHTTYPE);
```

```
unsigned long startTime = millis();

unsigned long ledLastToggleTime = millis();

unsigned long period = 300000; // 5 minutes in milliseconds

unsigned long ledInterval = 5000; // 5 seconds in milliseconds

bool ledState = false;


void setup() {

    pinMode(LED, OUTPUT);

    Serial.begin(115200);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        delay(500);

        Serial.print(".");

    }

    ThingSpeak.begin(client); // Initialize ThingSpeak

    dht.begin();

}


void loop() {

    unsigned long currentMillis = millis();

    // Correctly blink LED every 5 seconds

    if (currentMillis - ledLastToggleTime >= ledInterval) {

        // Toggle the LED state
```

```

    ledState = !ledState;

    digitalWrite(LED, ledState ? HIGH : LOW);

    ledLastToggleTime = currentMillis;
}

// Check if 5 minutes have passed to read temperature and
humidity

if (currentMillis - startTime >= period) {

    float humidity = dht.readHumidity();

    float temperature = dht.readTemperature();

    if (isnan(humidity) || isnan(temperature)) {

        Serial.println("Failed to read from DHT sensor!");

    } else {

        Serial.print("Humidity: ");

        Serial.print(humidity);

        Serial.print(" %\t");

        Serial.print("Temperature: ");

        Serial.print(temperature);

        Serial.println(" *C ");

        ThingSpeak.setField(1, temperature);

        ThingSpeak.setField(2, humidity);

        ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    }
}

```

```

        startTime = currentMillis; // Reset the timer for the next
read

    }

}

```

Para que la placa ESP32 pueda medir datos de manera continua por 3 días sin apagarse se le agregó una función, la cual prende un LED por 5 segundos y luego lo apaga por 5 segundos. Esto evita que entre en un estado de inactividad durante los 5 minutos que tarda en tomar los datos del sensor.

Además de la librería DHT mencionada anteriormente, se utilizó la librería “ThingSpeak” que habilita la conexión con la API del canal de ThingSpeak creado, y la librería “WiFi”, la cuál habilita la conexión inalámbrica por internet de la placa utilizada.

- Una vez realizada la conexión entre la placa y la página de ThingSpeak (<https://thingspeak.com/channels/2452531>), se pueden observar los cambios de temperatura desde el sensor, y se reflejan en la página



10. Se pueden modificar los parámetros de actualización para enviar los datos desde el sensor hasta la página dentro del código.

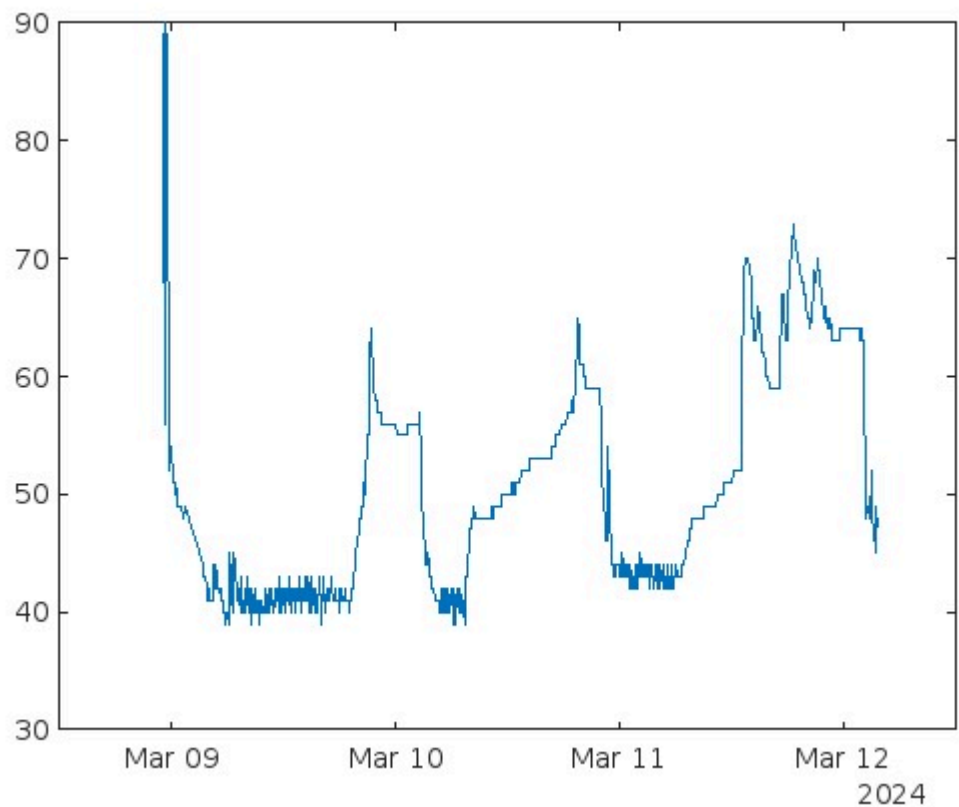
11. Luego se utilizan los templates de MATLAB para poder visualizar los datos en gráficas. Esto se hace desde la página del canal de ThingSpeak, haciendo click a MATLAB Visualization. Se utiliza el siguiente código:

```
% Template MATLAB code for visualizing data from a channel as a 2D line
% plot using PLOT function.
% Prior to running this MATLAB code template, assign the channel variables.
% Set 'readChannelID' to the channel ID of the channel to read from.
% Also, assign the read field ID to 'fieldID1'.
% TODO - Replace the [] with channel ID to read data from:
readChannelID = 2452531;
% TODO - Replace the [] with the Field ID to read data from:
fieldID2 = 2;
% Channel Read API Key
% If your channel is private, then enter the read API
% Key between the " below:
readAPIKey = '4TGG2BK6PSM0U4OJ';
%% Read Data %%
[data, time] = thingSpeakRead(readChannelID, 'Field', fieldID2, 'NumPoints', 955, 'ReadKey',
readAPIKey);
%% Visualize Data %%
plot(time, data);
```

- a. Se puede modificar el código para obtener cualquiera de los dos campos (temperatura o humedad) cambiando el parámetro “fieldID2 = 2” a “fieldID1 = 1”.

12. Se presiona el botón de Save and Run y se obtienen los resultados.





## Conclusión

El proyecto es un dispositivo IOT potente y accesible, el sensor DHT11 ha permitido obtener datos precisos y confiables sobre las condiciones ambientales más específicas en la temperatura y humedad de nuestro ambiente cotidiano, esto tomando solo 3 días.

La capacidad de ThingSpeak para generar gráficos y estadísticas en tiempo real nos ha ayudado significativamente a la comprensión de las variaciones en la humedad y temperatura a lo largo del período de tres días pero necesitamos el uso de otra herramienta para analizar mejor los datos.

La integración con MATLAB ha proporcionado una herramienta poderosa para el análisis de los datos recopilados. El uso de las herramientas de MATLAB dentro de

ThingSpeak ha permitido realizar un gráfico para visualizar los datos y ver cómo se comportaron durante los tres días que estuvo obteniendo datos la placa.