**UNIVERSIDAD MODELO**

“Escuela de Ingeniería”

**"Monitoreo de Temperatura con Raspberry Pi Pico W y ThingSpeak"**

**ASIGNATURA**

Internet de las Cosas

**PROFESOR**

Freddy Antonio Ix Andrade

**ESTUDIANTE**

Luis Manuel Pacho Ayora [*15222403*]

**ENTREGA**

24/02/2025

**Introducción**

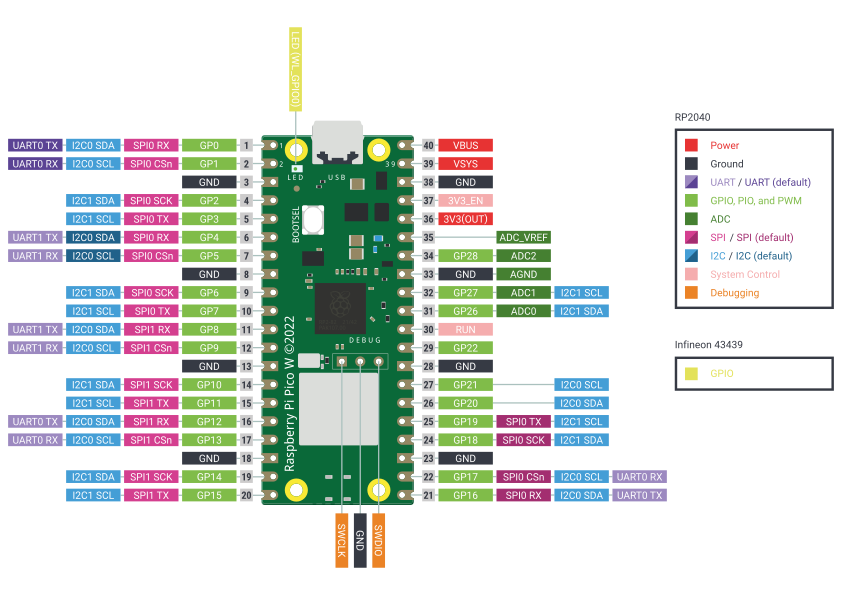
En la actualidad, el Internet de las Cosas (IoT) juega un papel crucial en la monitorización y control de diversos parámetros en múltiples sectores, incluyendo el industrial, doméstico y de investigación. Este proyecto tiene como objetivo desarrollar un sistema de monitoreo de temperatura basado en IoT, utilizando la Raspberry Pi Pico W y el sensor LM35 para capturar datos y enviarlos a la plataforma ThingSpeak para su almacenamiento y visualización.

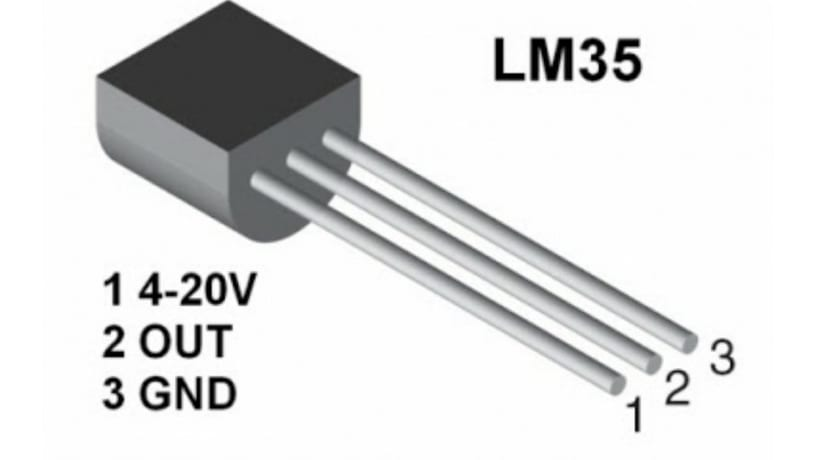
Para lograrlo, se emplean las siguientes herramientas:

* **Raspberry Pi Pico W**: una microcontroladora con conectividad Wi-Fi.
* **Sensor LM35**: que permite medir la temperatura con una salida analógica proporcional a la temperatura en grados Celsius.
* **ThingSpeak**: una plataforma en la nube utilizada para almacenar y visualizar los datos en tiempo real.
* **MathWorks**: para analizar los datos recopilados y generar alertas en caso de valores fuera de rango.
* **GitHub**: como sistema de control de versiones para documentar el código y compartir el proyecto.

Desarrollo del Proyecto

Diagrama de conexión del **LM35 con la Raspberry Pi Pico W**.



Este es el diagrama de la Raspberry Pi Pico W donde nos dice en que pines debemos poner el sensor LM35 en este caso este es el datashet del sensor para tenerlo mas claro:  


Pasos para conectar el sensor LM35 a la Raspberry.

1. Agarrar en PIN 40 (**Raspberry**) y conectarlo al PIN 1(VCC) del LM35.
2. Conectar el PIN 31 (GP 26) (**Raspberry**) al PIN 2 (Out) del LM35.
3. Conectar el PIN 38 (GND) (**Raspberry**) al PIN 3 (GND) del LM35.

Explicación del **funcionamiento del LM35** y su salida analógica.

El **sensor LM35** es un sensor de temperatura de precisión cuya salida es una señal analógica proporcional a la temperatura en grados Celsius. Su principal ventaja es que proporciona una lectura lineal con una escala de **10 mV por grado Celsius** (10 mV/°C), lo que facilita su conversión a temperatura sin necesidad de calibración adicional.

**Funcionamiento del LM35**

El LM35 tiene tres pines:

* **VCC**: Alimentación (4V a 30V, en este caso se usa 5V de la Raspberry Pi Pico W).
* **GND**: Tierra.
* **Vout**: Salida analógica que varía en función de la temperatura medida.

La salida del LM35 se puede leer con un **convertidor analógico-digital (ADC)**, como el incorporado en la Raspberry Pi Pico W.

Donde:

* VoutV\_{out}Vout​ es el voltaje en voltios medido en la salida del sensor.
* El factor de conversión 100100100 proviene de la escala de 10 mV/°C.

Circuito Ensamblado



***Explicación del código fuente***

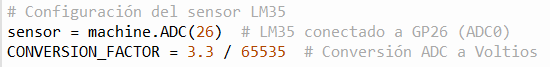
Para establecer la conexión a Internet, se utiliza la librería **network**.

1. Se activa la interfaz Wi-Fi con network.WLAN(network.STA\_IF).
2. Si no hay conexión, intenta conectarse con wlan.connect(SSID, PASSWORD).
3. Se espera un máximo de 15 segundos para conectarse, verificando el estado con wlan.isconnected().
4. Si la conexión es exitosa, imprime la dirección IP obtenida; de lo contrario, muestra un mensaje de error.

***Lectura del sensor LM35 usando el ADC***

El sensor LM35 genera una señal analógica proporcional a la temperatura.

1. Se configura el **ADC (Convertidor Analógico-Digital)** en el pin **GP26** con machine.ADC(26).
2. Se lee la señal analógica con sensor.read\_u16(), que devuelve un valor de 16 bits (0-65535).
3. Se convierte el valor en voltios usando CONVERSION\_FACTOR = 3.3 / 65535.



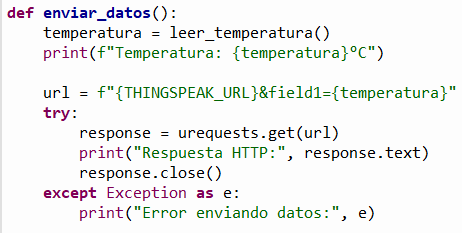
***Conversión de la señal analógica a temperatura en °C***

1. Se multiplica la lectura del ADC por el factor de conversión para obtener el voltaje.
2. Se convierte el voltaje en temperatura con la fórmula:

Dado que el **LM35 tiene una sensibilidad de 10mV/°C**, se divide el resultado entre 10 para obtener la temperatura en grados Celsius.

***Envío de datos a ThingSpeak***

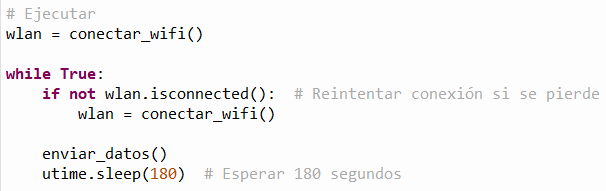
1. Se obtiene la temperatura llamando a leer\_temperatura().
2. Se genera la URL de la API de ThingSpeak con el valor de temperatura en field1.
3. Se envía la solicitud HTTP con urequests.get(url).
4. Si la solicitud es exitosa, imprime la respuesta del servidor; si falla, se captura la excepción y se muestra un mensaje de error.



***Ejecución del programa principal***

1. Se establece la conexión Wi-Fi con conectar\_wifi().
2. Se inicia un bucle infinito donde:

* Se verifica que la conexión a Wi-Fi sigue activa.
* Se envía la temperatura a **ThingSpeak**.
* Se espera **180 segundos** antes de la siguiente lectura.



GitHub

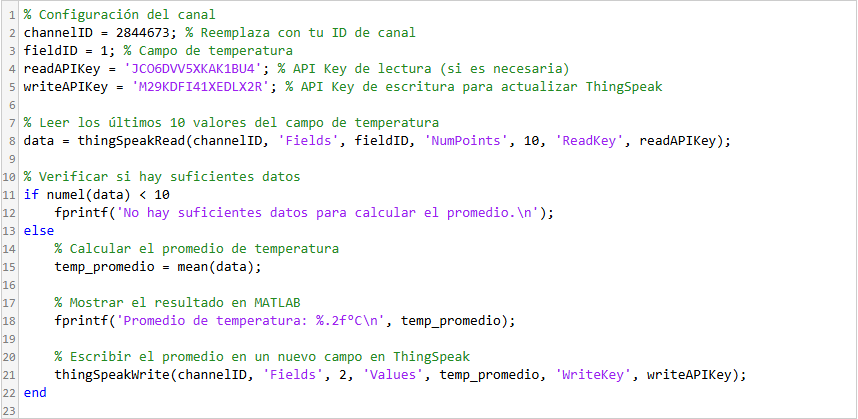
*README*: <https://github.com/xLuisPacho/IoT-Temperature-Monitor/blob/main/README.md>







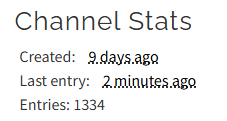
***Promedio y alerta de Temperatura***



Este código en .m nos nos saca el promedio de los últimos 10 datos que sacamos y lo promedia, y también nos manda una alerta si la temperatura pasa de los 35 grados todo esto mostrado en este código.

***Análisis de los resultados***

Hoy 27 de febrero del 2025 a las 5:30pm se han registrado **1334,** el sensor de temperatura fue conectado el lunes 24 de febrero a las 8:30pm y eso han sido los datos que se han reportado.

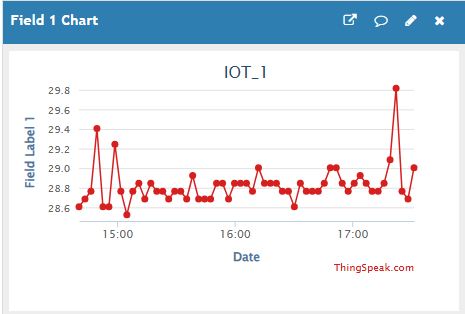


Hay 1334 datos registrado que a captado en sensor conectado.

***Preguntas***

1. **¿Qué tendencias o patrones se observaron en la temperatura?**

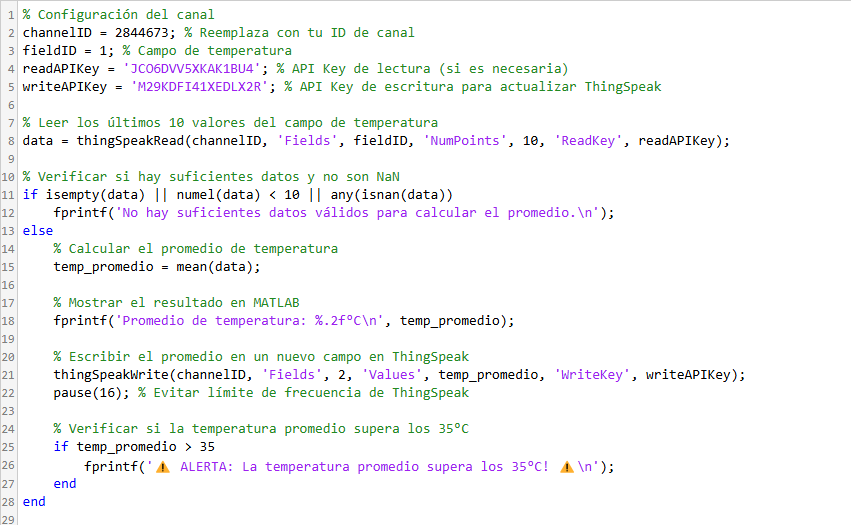
Lo que notaba en los resultados es que tuve un promedio de temperatura de 28.88 grados y si notaba picos de temperatura.



Entonces por las mañanas notaba una temperatura promedio, pero en la noche era cuando bajaba mas y puede ser por la temperatura que en la noche hay frio, o prendo el aire acondicionado puede variar por muchas cosas

1. **¿Se activaron alertas en MathWorks? ¿En qué condiciones?**

El código que tengo esta puesto que si la temperatura llega a 35 grados se active una alerta con el sensor, este es la parte del código:



1. **¿Se cumplieron los objetivos del proyecto?**

Por mi parte como estudiante y participe del proyecto yo creo que si cumpli con todos los objetivos del proyecto porque:

* Logre conectar el LM35 con la Raspberry
* Logre hacer una conexión con la Raspberry y el Thingspeak
* Logre hacer el github de mi proyecto

**Conclusiones y Reflexión**

¿**Qué aprendí sobre IoT y ThingSpeak?**

* Aprendí a conectar un **sensor de temperatura LM35** a una **Raspberry Pi Pico W** y leer datos utilizando **MicroPython**.
* Descubrí cómo enviar estos datos a **ThingSpeak**, almacenarlos en la nube y visualizarlos en tiempo real mediante gráficos.
* Usé **MATLAB Analysis** en ThingSpeak para calcular el promedio de temperatura y generar alertas cuando se superan ciertos umbrales.
* Comprendí los **límites de frecuencia de ThingSpeak** y la importancia de manejar correctamente las solicitudes para evitar errores.

¿**Cómo podría mejorar el proyecto en el futuro?**

* **Optimizar el código**, mejorando el manejo de errores y añadiendo una reconexión automática en caso de fallos de conexión WiFi.
* **Implementar notificaciones en tiempo real**, como alertas por **correo electrónico o SMS** cuando la temperatura supere un umbral crítico.
* **Desarrollar una interfaz gráfica más avanzada**, integrando los datos con **Power BI** o **Grafana** para mejorar la visualización.
* **Aplicar Machine Learning** en MATLAB para analizar tendencias y predecir cambios de temperatura antes de que ocurran problemas.

**Posibles aplicaciones prácticas en la industria:**

* **Monitoreo de temperatura en almacenes y laboratorios**, asegurando condiciones óptimas para productos sensibles.
* **Control de sistemas HVAC (calefacción, ventilación y aire acondicionado)** en edificios inteligentes para mejorar la eficiencia energética.
* **Supervisión de maquinaria industrial**, detectando sobrecalentamientos y previniendo fallos antes de que ocurran.
* **Agricultura inteligente**, optimizando el riego y la producción mediante el monitoreo de temperatura y humedad.