

Código para verificar si el led del raspberry funciona

```
hunny - <sin nombre> @ 4:9
ero  Editar  Visualizar  Ejecutar  Herramientas  Ayuda
hivos x
: computador
Users \ torda
) anaconda3
) ansel
) Contacts
) cypress
) demo
) django-app-modelk
) Documents
) Downloads
) env
) Favorites
) name

1 import machine
2 led = machine.Pin("LED", machine.Pin.OUT)
3 led.off()
4 led.on()
```

Código para mandar los datos cada 3 minutos.

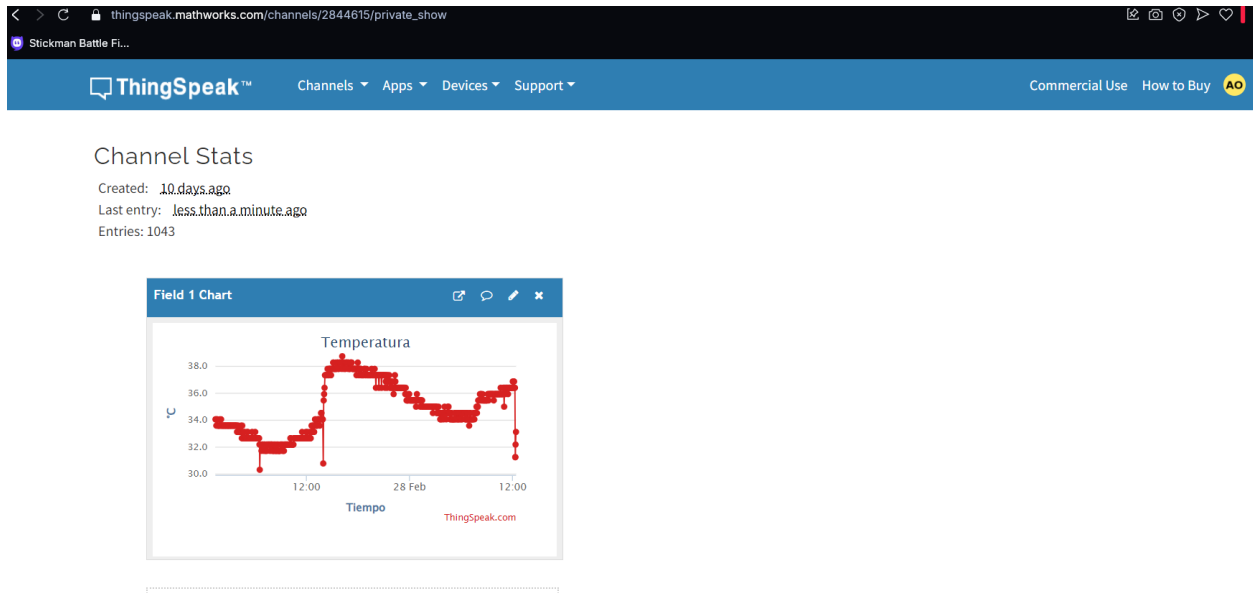
```
<sin nombre> * * [ main.py ] <
1  import network
2  import urequests
3  import utime
4  from machine import ADC
5
6  # Configuración de Wi-Fi
7  SSID = "INFINITUM7E34"
8  PASSWORD = "Yu2Yu2Fd3r"
9
10 # Configuración de ThingSpeak
11 THINGSPEAK_API_KEY = "I15QW8COVA2DECM7"
12 THINGSPEAK_URL = "https://api.thingSpeak.com/update"
13
14 # Sensor de temperatura interno (Canal ADC 4)
15 sensor_temp = ADC(4)
16 VOLTAGE_REFERENCE = 3.3 # Voltaje de referencia del ADC
17 CONVERSION_FACTOR = VOLTAGE_REFERENCE / 65535 # Para convertir la lectura de 16 bits
18
19 # Conexión a Wi-Fi
20 def conectar_wifi():
21     wlan = network.WLAN(network.STA_IF)
22     wlan.active(True)
23     wlan.connect(SSID, PASSWORD)
24
25     print("Conectando a Wi-Fi...", end="")
26     while not wlan.isconnected():
27         print(".", end="")
28         utime.sleep(1)
29     print("\nConectado:", wlan.ifconfig())
30
31 # Lectura de temperatura interna en °C
32 def leer_temperatura():
33     valor_adc = sensor_temp.read_u16() # Leer ADC
34     voltaje = valor_adc * CONVERSION_FACTOR # Convertir a voltaje
35     temperatura = 27 - (voltaje - 0.706) / 0.001721 # Fórmula del RP2040
36     return temperatura
37
38 # Envío de datos a ThingSpeak
39 def enviar_a_thingSpeak(temperatura):
40     try:
41         url = f"{THINGSPEAK_URL}?api_key={THINGSPEAK_API_KEY}&field1={temperatura}"
42         respuesta = urequests.get(url)
43         print("Datos enviados:", respuesta.text)
44         respuesta.close()
45     except Exception as e:
46         print("Error al enviar datos:", e)
47
48 # Flujo principal
49 conectar_wifi()
50 while True:
51     temp = leer_temperatura()
52     print(f"Temperatura interna: {temp:.2f}°C")
53     enviar_a_thingSpeak(temp)
54     utime.sleep(180) # Esperar 180 segundos
55
```

30

Consola

```
MPY: soft reboot
Conectando a Wi-Fi...
Conectado: ('192.168.1.91', '255.255.255.0', '192.168.1.254', '192.168.1.254')
Temperatura interna: 32.19°C
Datos enviados: 1042
```

Por otra parte, se creó un canal en la plataforma de Thingspeak para poder visualizar los datos mediante una gráfica, al código se le puso el id del canal creado para que este pudiera recibir los datos y así se fuera formando la gráfica.



Después se hicieron las partes de alerta de temperatura y el promedio de temperatura.

La de promedio toma los últimos 10 datos generados y los promedia para obtener un valor general de temperatura.

Name

Promedio temperatura

MATLAB Code

```
1 readChannelID =2844615;
2 fieldID = 1;
3 numPoints = 10; % Últimos 10 datos
4
5 % Leer los últimos 10 valores
6 temperatureData = thingSpeakRead(readChannelID, 'Fields', fieldID, 'NumPoints', numPoints);
7
8 % Calcular el promedio
9 avgTemperature = mean(temperatureData);
10
11 % Mostrar el resultado
12 fprintf('Temperatura promedio: %.2f °C\n', avgTemperature);
13
```

Save and Run Save

Output

Temperatura promedio: 34.77 °C

my channels

New Channel

Most recent channels

Name: Tarea

Channel ID: 2844615

Access: Public

Read API Key: CLU0SBY48XEJKS10

Write API Key: I15QW8COVA2DECM7

Fields:

1: Field Label 1

Analyze Data

thingSpeakRead

thingSpeakWrite

urlfilter

Y el código de alerta lo que hace es que toma el dato más reciente y verifica si la temperatura es mayor a 35 grados, en caso de ser así manda una alerta como de precaución.

Temperatura normal

MATLAB Code

```
1 readChannelID = 2844615;
2 readAPIKey = 'CLU0SBY48XEJKS10';
3 writeAPIKey = 'I15QW8COVA2DECM7';
4 fieldID = 1;
5
6 % Leer el último dato de temperatura
7 lastTemperature = thingSpeakRead(readChannelID, 'Fields', fieldID, 'NumPoints', 1, 'ReadKey', readAPIKey);
8
9 % Verificar si la temperatura es mayor a 35°C
10 if lastTemperature > 35
11     % Guardar alerta en Field2
12     thingSpeakWrite(readChannelID, 'Fields', 2, 'Values', 1, 'WriteKey', writeAPIKey);
13     fprintf('⚠ Alerta: Temperatura alta (%.2f °C)\n', lastTemperature);
14 else
15     thingSpeakWrite(readChannelID, 'Fields', 2, 'Values', 0, 'WriteKey', writeAPIKey);
16     fprintf('✅ Temperatura normal: %.2f °C\n', lastTemperature);
17 end
18
19
20
21
22
```

Save and Run

Save

Output

✅ Temperatura normal: 33.60 °C

Most recent channels

Name: Tarea

Channel ID: 2844615

Access: Public

Read API Key: CLU0SBY48XEJK

Write API Key: I15QW8COVA2DE

Fields:

1: Field Label 1

➤ Analyze Data

➤ thingSpeakRead

➤ thingSpeakWrite

➤ urlfilter

Temperatura arriba de 35 grados

MATLAB Code

```
1 readChannelID = 2844615;
2 readAPIKey = 'CLU0SBY48XEJKS10';
3 writeAPIKey = 'I15QW8COVA2DECM7';
4 fieldID = 1;
5
6 % Leer el último dato de temperatura
7 lastTemperature = thingSpeakRead(readChannelID, 'Fields', fieldID, 'NumPoints', 1, 'ReadKey', readAPIKey);
8
9 % Verificar si la temperatura es mayor a 35°C
10 if lastTemperature > 35
11     % Guardar alerta en Field2
12     thingSpeakWrite(readChannelID, 'Fields', 2, 'Values', 1, 'WriteKey', writeAPIKey);
13     fprintf('⚠ Alerta: Temperatura alta (%.2f °C)\n', lastTemperature);
14 else
15     thingSpeakWrite(readChannelID, 'Fields', 2, 'Values', 0, 'WriteKey', writeAPIKey);
16     fprintf('✅ Temperatura normal: %.2f °C\n', lastTemperature);
17 end
18
19
20
21
22
```

Save and Run

Save

Output

⚠ Alerta: Temperatura alta (43.43 °C)

Name: Tarea

Channel ID: 2844615

Access: Public

Read API Key: CLU0SBY48XEJK

Write API Key: I15QW8COVA2DE

Fields:

1: Field Label 1

➤ Analyze Data

➤ thingSpeakRead

➤ thingSpeakWrite

➤ urlfilter

