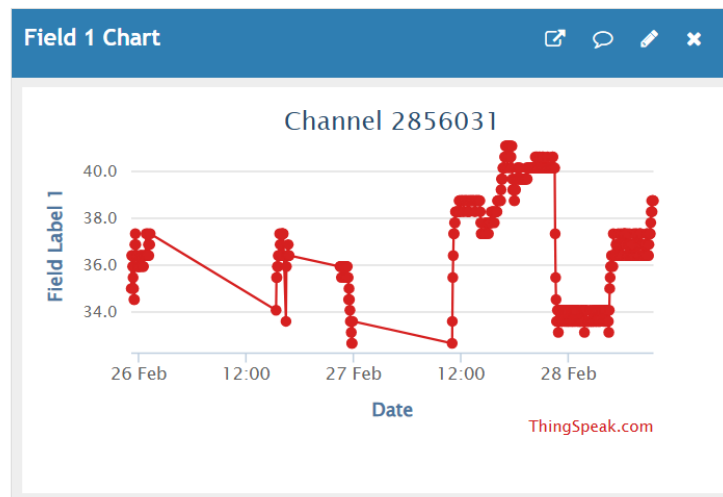


Channel Stats

Created: 2 days ago

Last entry: less than a minute ago

Entries: 547



ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy BS

Channel ID: 2856031
Author: mwa000036990936
Access: Public

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Add Visualizations Add Widgets Export recent data

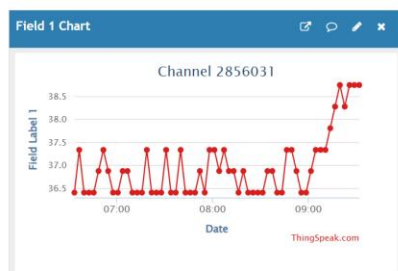
MATLAB Analysis MATLAB Visualization

Channel Stats

Created: 2 days ago

Last entry: about a minute ago

Entries: 547



ThingSpeak™

Channels ▾ Apps ▾ Devices ▾ Support ▾

Commercial Use How to Buy

BS

Apps / MATLAB Analysis / Promedio / Edit

Name

Promedio

MATLAB Code

```
1
2 % Configuración
3 readChannelID = 2856031; % ID del canal de ThingSpeak
4 fieldID = 1; % Field1 (temperatura)
5 numPoints = 10; % Últimos 10 datos
6
7 % Leer los últimos 10 valores
8 temperatureData = thingSpeakRead(readChannelID, 'Fields', fieldID, 'NumPoints', numPoints);
9
10 % Calcular el promedio
11 avgTemperature = mean(temperatureData);
12
13 % Mostrar el resultado
14 fprintf('Temperatura promedio: %.2f °C\n', avgTemperature);
```

Save and Run

Save

Output

Temperatura promedio: 38.14 °C

Help

My Channels

Documentation

New Channel

Most recent channels

Name: Channel 2856031

Channel ID: 2856031

Access: Public

Read API Key: 3038G0R79524ZL5C

Write API Key: 92EGPAMN2LQRDQVZ

Fields:

1: Field Label 1

Analyze Data

thingSpeakRead

thingSpeakWrite

urlfilter

ThingSpeak™

Channels ▾ Apps ▾ Devices ▾ Support ▾

Commercial Use How to Buy

BS

My Channels

Documentation

New Channel

Most recent channels

Name: Channel 2856031

Channel ID: 2856031

Access: Public

Read API Key: 3038G0R79524ZL5C

Write API Key: 92EGPAMN2LQRDQVZ

Fields:

1: Field Label 1

Analyze Data

thingSpeakRead

thingSpeakWrite

urlfilter

Name

Alert

MATLAB Code

```
1 |
2 % Configuración del canal
3 readChannelID = 2856031; % Reemplaza con tu Channel ID real
4 readAPIKey = '3038G0R79524ZL5C'; % ??? Agrega comillas
5 writeAPIKey = '92EGPAMN2LQRDQVZ'; % ??? Agrega comillas
6 fieldID = 1; % Field1 (temperatura)
7
8 % Leer el último dato de temperatura
9 lastTemperature = thingSpeakRead(readChannelID, 'Fields', fieldID, 'NumPoints', 1, 'ReadKey', readAPIKey);
10
11 % Verificar si la temperatura es mayor a 35°C
12 if lastTemperature > 35
13     % Guardar alerta en Field2
14     thingSpeakWrite(readChannelID, 'Fields', 2, 'Values', 1, 'WriteKey', writeAPIKey);
15     fprintf('⚠ Alerta: Temperatura alta (%.2f °C)\n', lastTemperature);
16 else
17     thingSpeakWrite(readChannelID, 'Fields', 2, 'Values', 0, 'WriteKey', writeAPIKey);
18     fprintf('✅ Temperatura normal: %.2f °C\n', lastTemperature);
19 end
```

Save and Run

Save

Output

⚠ Alerta: Temperatura alta (38.75 °C)

Thonny - C:\Users\grill\Downloads\Codigo.py @ 11:51

Fichero Editar Visualizar Ejecutar Herramientas Ayuda



Codigo.py

```
1 import network # Librería para manejar la conexión Wi-Fi
2 import urequests # Librería para realizar solicitudes HTTP
3 import utime # Librería para manejar el tiempo y pausas en la ejecución
4 import machine # Librería para controlar hardware del microcontrolador
5
6 # Configuración Wi-Fi
7 SSID = "CARMONA" # Nombre de la red Wi-Fi a la que se conectará
8 PASSWORD = "AbabL2019" # Contraseña de la red Wi-Fi
9
10 # Configuración de ThingSpeak
11 API_KEY = "92EGPAWN2LQRDQVZ" # Clave API de ThingSpeak para enviar datos
12 THINGSPEAK_URL = "https://api.thingspeak.com/update" # URL de la API de ThingSpeak
13
14 # Función para conectarse a la red Wi-Fi
15 def conectar_wifi():
16     wlan = network.WLAN(network.STA_IF) # Se crea un objeto WLAN en modo estación (STA)
17     wlan.active(True) # Se activa la interfaz Wi-Fi
18     wlan.connect(SSID, PASSWORD) # Se intenta conectar a la red con las credenciales
19
20     print("Conectando a Wi-Fi...")
21     while not wlan.isconnected(): # Se espera hasta que la conexión se establezca
22         utime.sleep(1) # Espera de 1 segundo antes de volver a comprobar
23
24     print("Conectado a Wi-Fi:", wlan.ifconfig()) # Se imprime la configuración de la conexión
25
26 # Función para leer la temperatura interna del RP2040
27 def leer_temperatura():
28     sensor_temp = machine.ADC(4) # Se inicializa el ADC en el canal 4 (sensor interno de temperatura)
29     conversion_factor = 3.3 / 65535 # Factor de conversión para obtener el voltaje real
30     lectura = sensor_temp.read_u16() * conversion_factor # Se obtiene la lectura en voltios
31     temperatura = 27 - (lectura - 0.706) / 0.001721 # Conversión a temperatura en grados Celsius según la fórmula del RP2040
```

4

Consola

>>> %Run -c \$EDITOR_CONTENT

```
MPY: soft reboot
Conectando a Wi-Fi...
Conectado a Wi-Fi: ('192.168.3.106', '255.255.255.0', '192.168.3.1', '192.168.3.1')
Temperatura actual: 37.81 °C
Enviado a ThingSpeak: 621
```