



## Registro de temperatura

2021



## Contenido

Objetivos .....	3
Introducción .....	3
Requerimientos .....	3
Instalar Matplotlib .....	3
Graficar los datos en vivo .....	6
Ejecutar el script al reinicio .....	6
Bibliografía .....	8



## Objetivos

- Crear una secuencia de comandos de Python simple que se pueda ejecutar automáticamente al iniciar la Raspberry Pi.
- Tomar medidas del sensor de temperatura a intervalos determinados y escribirlas en archivos de registro.
- Ver los datos como un gráfico trazado de forma interactiva.

## Introducción

El sistema en un chip (SoC) de la Raspberry Pi tiene un sensor de temperatura que se puede usar para medir su temperatura desde la línea de comando.

A veces haciendo cosas que requieren de mucha potencia, aparece un termómetro en la parte superior derecha de la pantalla que indica que la temperatura de la Raspberry Pi es alta.

Si el termómetro es completamente rojo significa que el procesador está a más de 85 grados centígrados. Si ocurre eso se deben tomar medidas drásticas para reducir la temperatura y no dañar o reducir la vida útil de la Raspberry Pi.

## Requerimientos

- Una Raspberry Pi

## Instalar Matplotlib

1. Para instalar el software que se necesita, ejecutar el siguiente comando en una terminal (mantener presionado ctrl + alt + t):

```
sudo apt-get install python3-matplotlib
```



## Monitorear la temperatura

1. Abrir un nuevo documento de Python 3 dirigiéndose a Menú>Programación>mu.
2. Se puede usar el módulo GPIO Zero para encontrar la temperatura de la CPU. Copiar y pegar lo siguiente en el documento creado:

```
from gpiozero import CPUtemperature
cpu = CPUtemperature()
```

3. Guardar y ejecutar el programa, y luego en el shell en la parte inferior de la ventana de Mu, escribir `cpu.temperature`.

```
>>> cpu.temperature
75.75
```

Figura 1 Temperatura de la CPU en °C.

## Escribir los datos en un archivo CSV

Sería útil si esos datos pudieran almacenarse en algún lugar. Un archivo CSV (valores separados por comas) es ideal para esto, ya que puede ser utilizado por aplicaciones como Excel y LibreOffice.

1. Escribir el siguiente código en el archivo Python para importar las funciones y métodos necesarios:

```
from time import sleep, strftime, time
```

`sleep` permite pausar el programa, `strftime` obtiene la fecha como una cadena de caracteres y `time` obtiene la hora exacta, lo que se conoce como tiempo UNIX.

Después se debe declarar una variable para almacenar la URL de la llamada a la API y realizar la llamada.

Para escribir en el archivo csv primero debe crearse.

2. Agregar la siguiente línea:

```
with open("/home/pi/Desktop/Registro_temperatura.csv", "a") as log:
```

Esto crea un nuevo archivo llamado `Registro_temperatura.csv` y lo abre con el nombre de `log`.



Ahora, se debe iniciar un ciclo infinito que se ejecutará hasta que se finalice el programa con Ctrl + C, dentro del bucle, se obtiene la temperatura y se almacena como una variable.

```
while True:  
    temp = cpu.temperature
```

3. Escribir la siguiente línea para guardar tanto la fecha y la hora actuales, más la temperatura, en el archivo CSV:

```
log.write("{0},{1}\n".format(strftime("%Y-%m-%d %H:%M:%S"), str(temp)))
```

Esa línea es un poco complicada, así que analicémosla un poco:

- `log.write()` escribirá cualquier cadena que esté entre corchetes en el archivo CSV.
- `"{0},{1}\n"` es una cadena que contiene dos marcadores de posición separados por una coma y que terminan en una nueva línea.
- `strftime("%Y-%m-%d %H:%M:%S")` se inserta en el primer marcador de posición. Es la fecha y hora actuales como una cadena.
- `str(temp)` es la temperatura de la CPU convertida en una cadena, que se escribe en el segundo marcador de posición después de la coma.

4. Por último, se puede agregar una sola línea al final del archivo para pausar el script entre escrituras. Aquí está haciendo una pausa de un segundo, pero se puede usar cualquier intervalo que se desee:

```
sleep(1)
```

El script completo luce de esta forma:

```
from gpiozero import CPUtemperature  
from time import sleep, strftime, time  
  
cpu = CPUtemperature()  
  
with open("/home/pi/Desktop/Registro_temperatura.csv", "a") as log:  
    while True:  
        temp = cpu.temperature  
        log.write("{0},{1}\n".format(strftime("%Y-%m-%d  
%H:%M:%S"), str(temp)))  
  
        sleep(1)
```



## Graficar los datos en vivo

Se puede producir un gráfico de temperaturas de la CPU que se actualizará a medida que se registren los datos.

1. Importar la biblioteca matplotlib donde están las otras importaciones:

```
import matplotlib.pyplot as plt
```

2. Las siguientes tres líneas pueden ir después de las importaciones. Le dicen a matplotlib que va a realizar un trazado interactivo y también crean las dos listas que contendrán los datos que se trazarán:

```
plt.ion()  
x = []  
y = []
```

3. Las siguientes líneas van todas al ciclo while True, antes de que se escriba el CSV, pero después de la línea temp = cpu.temperature. En primer lugar, agregar la temperatura actual al final de la lista y, y el tiempo al final de la lista x:

```
y.append(temp)  
x.append(time())
```

4. A continuación, se debe borrar el gráfico y luego se calculan los puntos y las líneas:

```
plt.clf()  
plt.scatter(x, y)  
plt.plot(x, y)
```

5. Por último, la gráfica se puede pausar por un segundo y luego se puede dibujar la gráfica. Eliminar la función de sleep y usar plt.pause en su lugar:

```
plt.pause(1)  
plt.draw()
```

6. Ejecutar el programa y ver que el gráfico se dibuja de forma interactiva.

## Ejecutar el script al reinicio

Si se desea ejecutar el script cuando se inicia la Raspberry es mejor que este ordenado en funciones de la siguiente forma:

```
from gpiozero import CPUtemperature  
from time import sleep, strftime, time  
import matplotlib.pyplot as plt
```

```
cpu = CPUtemperature()
```

```
plt.ion()
```



```
x = []
y = []

def write_temp(temp):
    with open("/home/pi/Desktop/Registro_temperatura.csv", "a") as log:
        log.write("{0},{1}\n".format(strftime("%Y-%m-%d
%H:%M:%S"), str(temp)))

def graph(temp):
    y.append(temp)
    x.append(time())
    plt.clf()
    plt.scatter(x,y)
    plt.plot(x,y)
    plt.draw()

while True:
    temp = cpu.temperature
    write_temp(temp)
    # graph(temp)
    plt.pause(1)
```

Automatizar scripts es simple con crontab. Básicamente, se trata de un archivo donde se pueden colocar comandos que se ejecutarán en determinados momentos o después de determinados eventos.

1. Abrir una terminal y teclear el siguiente comando (en caso de que pregunte escoger nano como el editor):

**crontab -e**

2. En la parte inferior del archivo escribir la siguiente línea que contiene la ruta del script a ejecutar:

```
@reboot python3 /home/pi/Desktop/Registro_temperatura.py
```

3. Reiniciar la Raspberry.
4. Teclear en una terminal lo siguiente:

```
cat /home/pi/Desktop/Registro_temperatura.csv
```

Ahora se pueden observar los datos del archivo csv.

5. Para ver el grafico en vivo descomentar la línea graph(temp) borrando el símbolo #.

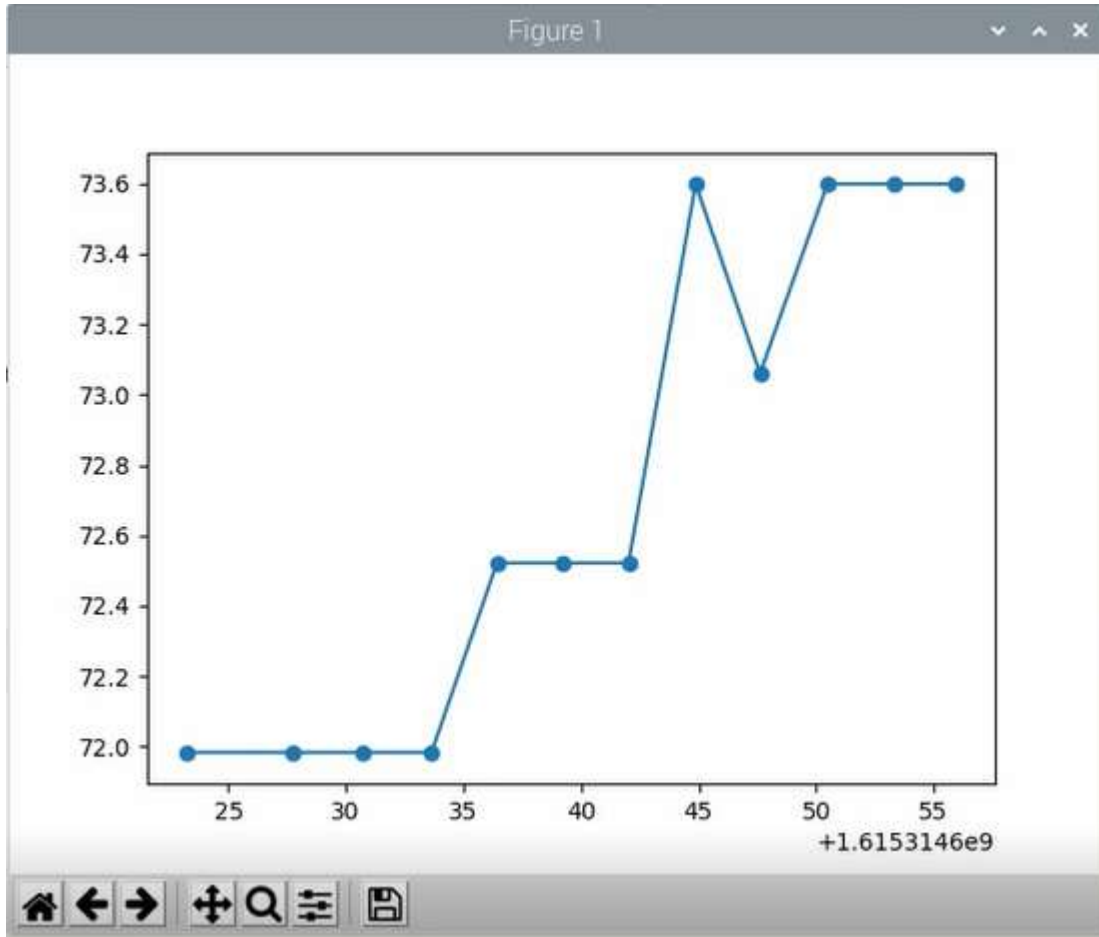


Figura 2 Gráfica de temperatura en vivo.

## Bibliografía

- Temperature log. (2021). Recuperado el 10 de marzo de 2021, de <https://projects.raspberrypi.org/en/projects/temperature-log>