

## Laboratorio 4 – Configuración de la RaspberryPi

**Entrega:** Individual en ALUD, Domingo 18 de Octubre 2020

### Objetivos

El objetivo de esta divertida e interesante práctica es configurar el sistema embebido RaspberryPi “headless” (sin conectarle monitor o teclado) para poder programarlo desde un ordenador portátil. En esta práctica aprenderás de los conceptos de SSH, VNC, y SFTP. Además, podrás practicar la creación de un servicio en Linux que se lanzará en tu Raspberry Pi.

En el resto del documento nos referiremos a las Raspberry Pi como RPi

### Importante:

Por favor completa las actividades en orden, pues en general cada una requiere haber completado las anteriores.

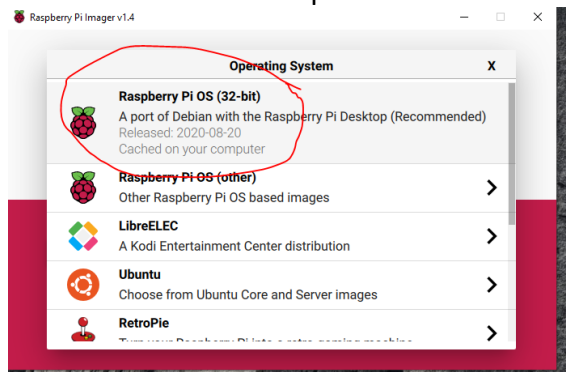
## Parte 1: Configuración de la RPi

### Actividad 1: Instalación de RaspberryPi OS.

1. Raspberry Pi es el sistema operativo oficial de la RPi. Usa el *Raspberry Pi imager* para instalar fácilmente el sistema operativo en la tarjeta SD. Haz clic en el siguiente link para descargar:

[Raspberry Pi Imager for Windows](#)

2. Una vez descargado, ejecuta el archivo para instalar el programa. Se abrirá automáticamente. Selecciona el sistema operativo como se muestra debajo



3. Una vez seleccionado, haz clic en “Write”

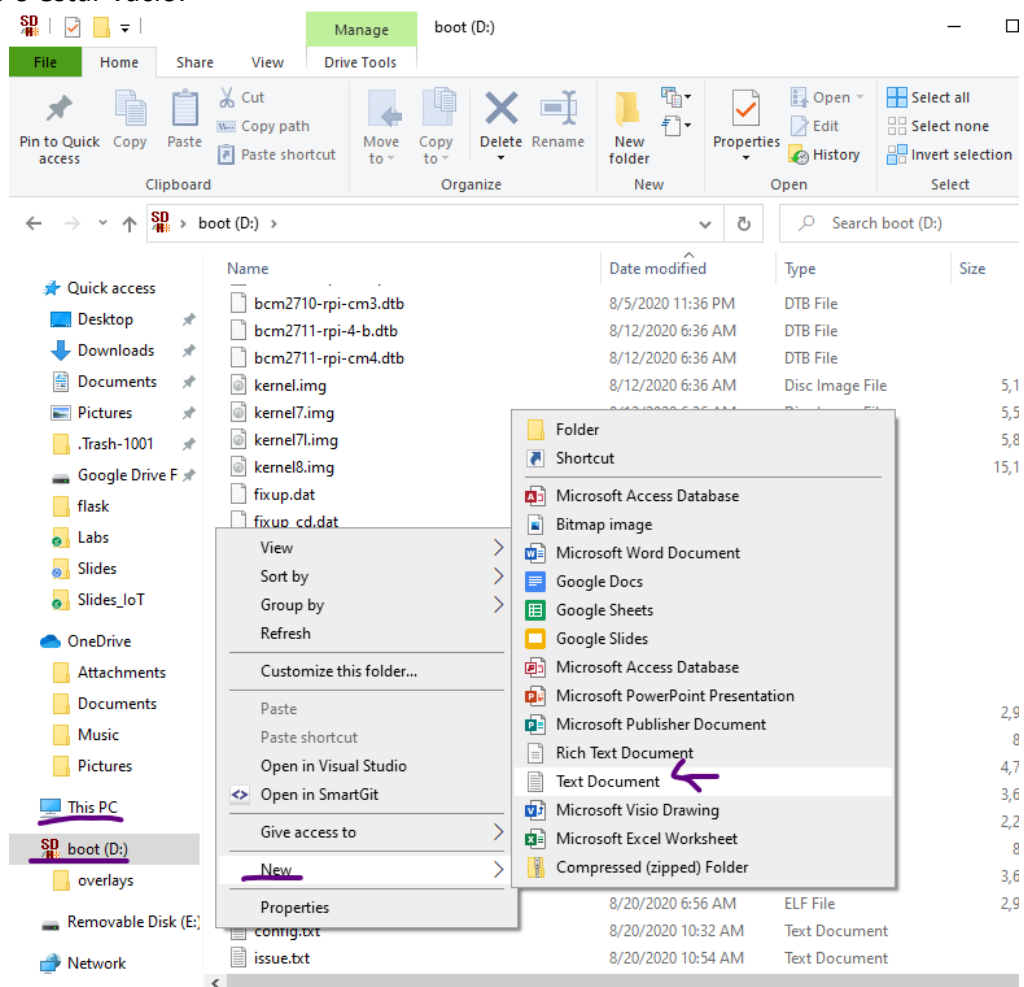
## Actividad 2: Conexión por ssh

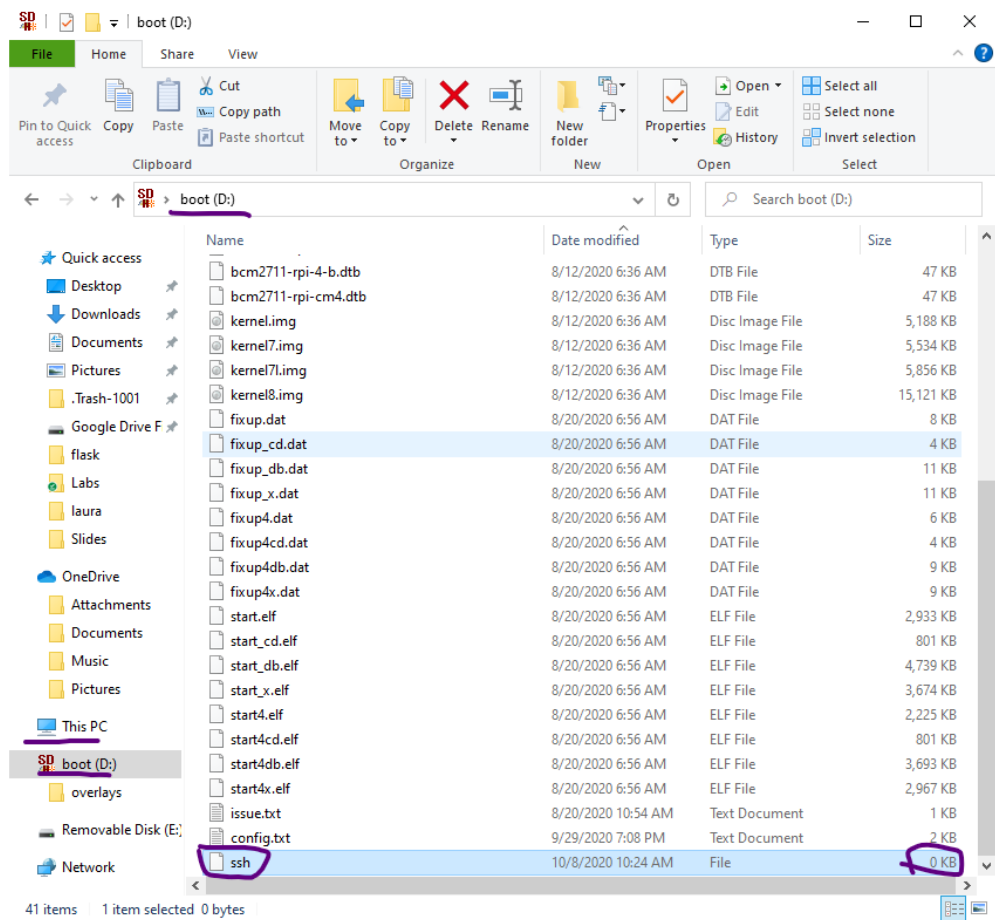
SSH (Secure Shell) es un protocolo de red criptográfico conectarse a servicios de red de forma segura a través de una red que no es segura. SSH proporciona un canal seguro de comunicación usando la arquitectura cliente-servidor. En nuestro caso, el cliente será el ordenador portátil y el servidor será la RPi.

Para poder acceder a nuestra RPi a través de ssh por primera vez, sin tener monitor ni teclado, debemos activar la interfaz de ssh.

1. Abre el contenido de la tarjeta SD desde Windows y crea un archivo llamado **ssh** en la partición de arranque, **sin ninguna extensión** ni contenido.

Lo que ocurre es que cuando la RPi arranca, busca el archivo ssh. Si lo encuentra, ssh se activa y el fichero se borra. El contenido del fichero no importa, puede contener cualquier texto o estar vacío.





Siguientes pasos:

1. Conecta tu RPi a la misma red local que tu portátil mediante un cable de Ethernet, y conecta la alimentación. Ten cuidado de no poner la RPi encima de una superficie metálica, así como de no cubrirla por encima, para favorecer su ventilación. Puedes usar un switch, o conectar la RPi y tu portátil directamente mediante un cable de Ethernet.
2. El siguiente paso es encontrar la dirección IP de tu RPi. Para ello, tienes varias opciones, como ya se ha comentado en clase:
  - Por defecto, la RPi tiene el hostname "raspberrypi". Por tanto, puedes usar el comando ping para conocer su IP  

```
$ ping raspberrypi
```
  - Usar el comando nmap como vimos en clase  
Primero buscar la dirección IP de la subred desde Windows, con el programa cmd, y el comando ipconfig. A continuación, ejecutar  

```
$ nmap -sn xxx.xxx.x.0/24,
```

  
donde los valores con x son los que has encontrado de la subred.  
Si no tienes instalado nmap en la venta de Ubuntu en Windows, usa  

```
$ sudo apt install nmap
```

3. Una vez hayas encontrado la dirección IP de RPi, ya puedes conectarte a ella mediante ssh. En la ventana de Ubuntu de tu portátil, ejecuta  
`$ ssh pi@direcciónIP`  
Te pedirá la contraseña, que por defecto es **raspberry**

Para apagar tu RPi desde la conexión con ssh, usa

```
$ sudo shutdown now
```

### Actividad 3: Cambiar el hostname a “tunombrepi”

Por ejemplo, en mi caso sería laurapi

Es importante que cada uno tengáis vuestra RPi con un hostname diferente. Por el contrario, podéis conectaros por error a la RPi de un compañero.

Para ello:

```
$ raspi-config
```

(este comando lo usaremos mucho para otros temas de configuración)

Buscad la interfaz para cambiar el hostname (opción 2: Network Options). También podéis cambiar la contraseña si queréis.

Puedes comprobar que se ha actualizado mediante

```
$ hostname
```

También puedes encontrar la dirección IP de la RPi con

```
$ hostname -I
```

Una vez tengas el hostname cambiado, puedes conectarte a tu RPi a través del hostname en lugar de la dirección IP

```
$ ssh pi@raspberrypi
```

Nota: pi es el “username”, mientras que raspberrypi es el “hostname”

Te pedirá una contraseña. Por defecto, la contraseña es: **raspberry**

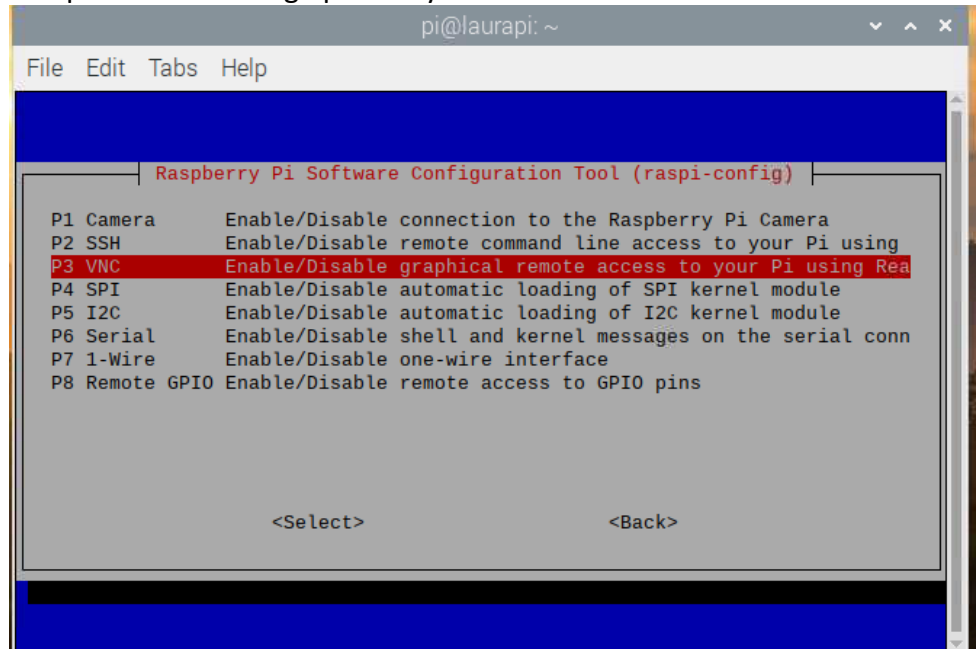
## Actividad 4: Conexión mediante VNC

Cómo habéis podido comprobar, trabajar mediante una conexión ssh funciona muy bien, pero no es tan intuitivo como tener una interfaz de usuario.

En primer lugar, activa VNC en la RPI.

`$raspi-config`

Selecciona la opción "Interfacing options" y selecciona VNC

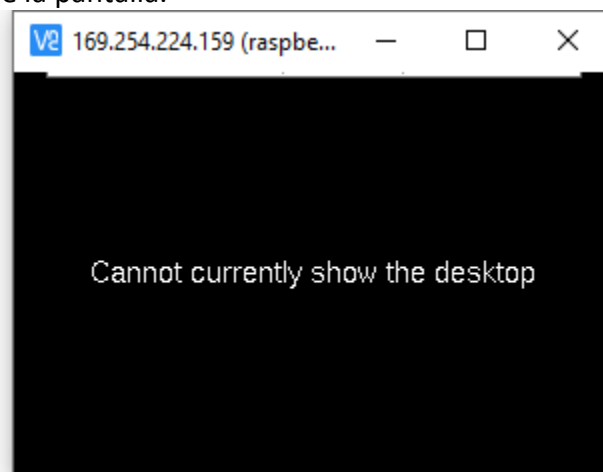


Descarga el siguiente programa de cliente VNC para Windows

<https://www.realvnc.com/en/connect/download/viewer/windows/>

Nota:

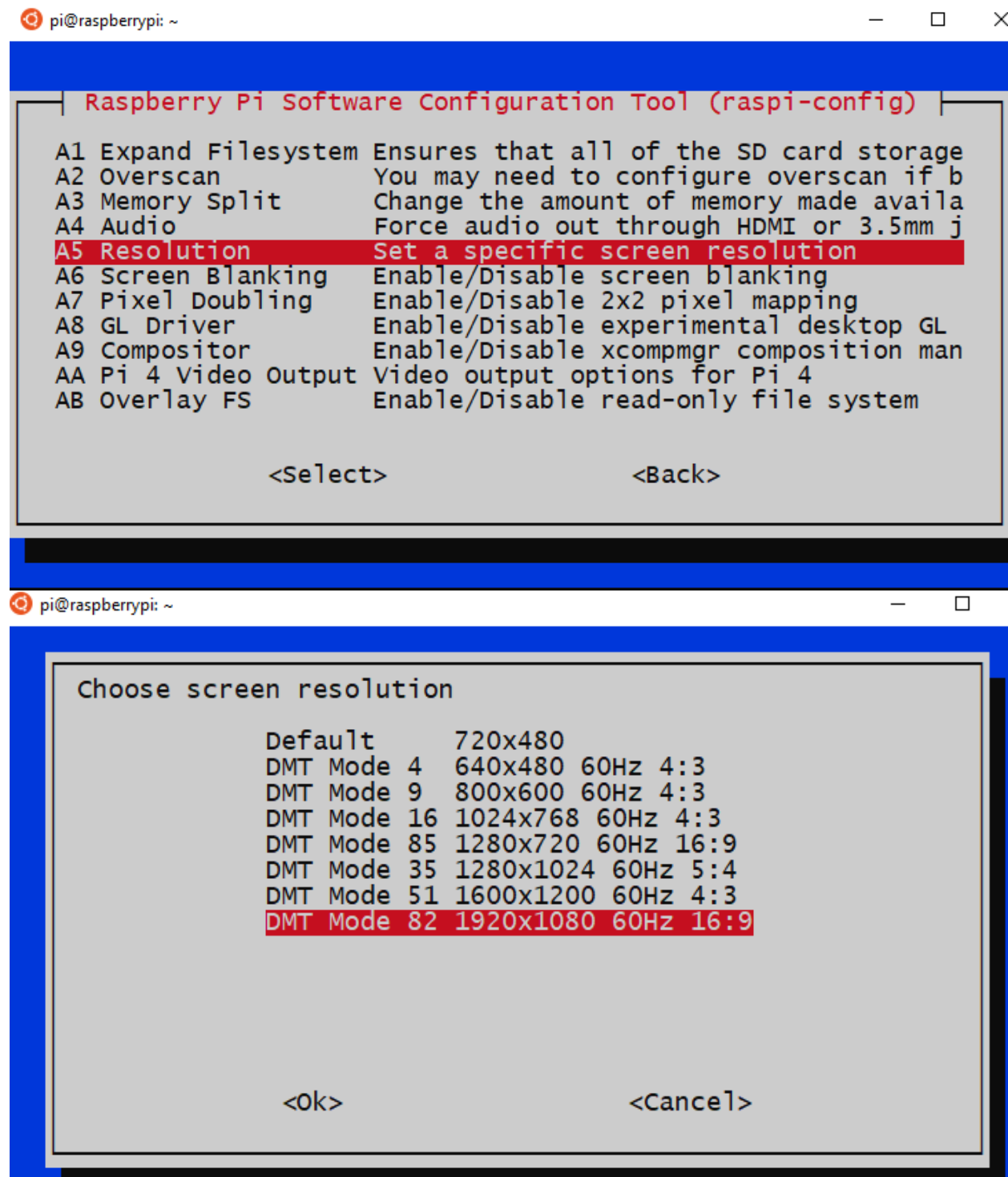
Si la primera vez que te conectas por VNC te da el siguiente error, es probable que tengas que cambiar la resolución de la pantalla.



Para solucionarlo, conéctate a la Rpi por SSH y ejecuta

`$sudo raspi-config`

Selecciona la opción "Advanced Options" -> Resolution -> DMT Mode 82  
Dile que Sí a reiniciar (reboot) ahora la RPi



### Actividad 5: Transferencia de ficheros por SFTP

El protocolo SFPT (SSH File Transfer Protocol) es un protocolo de red que proporciona acceso, transferencia, y gestión de ficheros sobre SSH. Con SFPT puedes fácilmente intercambiar, navegar, y editar ficheros en tu RPi. Necesitas que la RPi tenga SSH activado (ya lo deberías tener activado de los ejercicios anteriores)

Descárgate e instala el cliente de Windows para SFTP [WinSCP SFTP client](#).

Para conectarte a tu RPi:

Hostname: tunombrepi (lo debiste cambiar en la Actividad 3)

User name: pi

Password: raspberry (o la nueva si la has cambiado)

### Actividad 6: Conexión a Internet (esta parte dejarla para casa)

- a) Mediante Ethernet: conecta tu RPi mediante un cable de Ethernet directamente a un switch que a su vez esté conectado al router con otro cable. Conecta además tu portátil a ese mismo switch, para tener ambos dispositivos en la misma LAN.
- b) Mediante WiFi
  - I. Sin monitor (*headless* RPi): conecta tu RPi a tu portátil mediante ssh (detalles en Actividad 1), y configura la red WiFi siguiendo este tutorial:  
<https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>
  - II. Accede a tu RPi mediante un escritorio remoto usando VNC (ver Actividad 3) y configura la red WiFi con la interfaz gráfica.

## Parte 2: Actividades a entregar

---

### Actividad 1 (5 puntos): Entregar el archivo `rpiTemp.py`

Vamos a realizar un script en Python (`rpiTemp.py`) que **continuamente** obtenga la temperatura del microprocesador y que si esta pasa de un umbral definido (p.ej. 40 grados), que se comience a parpadear el LED Verde de la Raspberry Pi. Cuando la temperatura baja del umbral se vuelve a apagar el LED.

A continuación se explica el código de Python que necesitáis para obtener el valor de la temperatura, y controlar el LED

#### Función que obtiene el valor de la temperatura en python

```
import subprocess

def medir_temperatura():
    cpu = subprocess.check_output('sudo cat /sys/class/thermal/thermal_zone0/temp', shell=True)
    cpu_c = int(cpu)/1000
    print(cpu_c)
    return cpu_c
```

#### Para apagar el LED

```
import os
os.system('echo 0 >/sys/class/leds/led0/brightness') #Valor inicial: apagamos el LED
```

#### Para encender el LED (instantáneamente)

```
import os
os.system('echo 1 >/sys/class/leds/led0/brightness') #Valor inicial: apagos le LED
```

#### Para hacer que parpadee el led

```
os.system('modprobe ledtrig_heartbeat')
os.system('echo heartbeat >/sys/class/leds/led0/trigger')
```

Adicionalmente, echad un vistazo al Apéndice. Aunque no lo necesites para esta actividad, es importante y útil que lo conozcáis.



## Actividad 2 (2 puntos):

Entregar:

- fichero `_servicioTem.service`
- fichero `servicioTem.timer`
- pequeño vídeo donde se muestra donde arranquéis el servicio, mostréis que está activo con el comando `$systemctl status servicioTemp.service`, y donde se muestre el correcto funcionamiento, tal y como se ha explicado en clase. Para entregar el video, podéis compartir el link de donde lo tenéis guardado (github, Google drive, youtube, o cualquier otra plataforma que queráis)

Modificar el fichero anterior *rpiTemp.py* para que solo haga la comprobación 1 única vez, cuando se ejecuta el archivo (pista: eliminar el bucle que tengáis).

A continuación, tenéis que crear un servicio con `systemctl`, que arranque el script de la Actividad 1 (*rpiTemp.py*) al arrancar la RPi (cuando se enciende), y que se repita de lunes a viernes, cada 15 segundos.

Para realizar esta tarea, tenéis en ALUD, en la sección de esta tarea de Laboratorio 4, un servicio con temporizador como ejemplo. En el documento `README.txt` se dan detalles de como utilizarlo.

## Actividad 3 (3 puntos):

Repetir la actividad 1, pero en vez de parpadear el LED interno de la RPi, se debe encender LED que se os ha entregado, conectado al GPIO 18.

Llamad al fichero *rpiTemp\_gpio.py*

No es necesario que parpadee. Cuando la temperatura del microprocesador esta por encima del umbral, el LED se enciende. Cuando la temperatura esta por debajo del umbral, se apaga. Esta comprobación se debe repetir continuamente. No es necesario hacer un servicio para esta parte.

## Apéndice A: Comandos en Bash para ver la temperatura interna de la RPI

### Código en Bash

mostrar por pantalla la temperatura interna de la RPI (de la CPU)

```
$ cat /sys/class/thermal/thermal_zone0/temp
```

Para encender el LED (durante un instante):

```
$ echo 1 >/sys/class/leds/led0/brightness
```

Para apagar el LED:

```
$ echo 0 >/sys/class/leds/led0/brightness
```

*Para parpadear el LED*

```
$ modprobe ledtrig_heartbeat
```

```
$ echo heartbeat >/sys/class/leds/led0/trigger
```

Comando para mostrar por pantalla la temperatura interna de la RPI (de la CPU)

```
$cat /sys/class/thermal/thermal_zone0/temp
```

Nota: dividir por 1000 para tener la temperatura en grados centígrados

Comando para mostrar por pantalla la temperatura de la GPU de la RPI

```
$cat /opt/vc/bin/vcgencmd measure_temp)
```

Por último, podemos realizar la monitorización de la memoria en GNU/Linux

```
$cat /proc/meminfo
```

Este pequeño script (temperatura.sh) muestra de una forma “user-friendly” la temperatura de la CPU y de la GPU de la RPi.

```
cpu=$(cat /sys/class/thermal/thermal_zone0/temp)
echo "$(date) @ $(hostname)"
echo "-----"
echo "GPU => $(/opt/vc/bin/vcgencmd measure_temp)"
echo "CPU => $((cpu/1000))'C"
```

También podemos hacer esto en Python (temperatura.py). Para ejecutarlo, simplemente con \$python3 temperatura.py

```
import os
cpu =os.system('$(</sys/class/thermal/thermal_zone0/temp) ')
os.system('cpu=$(cat /sys/class/thermal/thermal_zone0/temp)')
os.system('echo "$(date) @ $(hostname)"')
os.system('echo "-----"')
os.system('echo "GPU => $(/opt/vc/bin/vcgencmd measure_temp)"')
os.system('echo "CPU => $((cpu/1000))"')
```

## Apéndice B: Comandos para controlar los “onboard” LEDs

Nota importante: Para ejecutar los siguientes comandos de configuración, debes estar logeado como superusuario

```
$sudo su
```

La RPi tienes dos LEDs “on-board” que podéis ver a simple vista

4. Color rojo (led1): indicador de que tiene la alimentación conectada
5. Color verde (led0): se puede programar como un GPIO. Este LED está conectado al pin GPIO 16, y es el indicador de lectura de la tarjeta SD.

El driver del LED del kernel, que controla el led0 y led1, tiene “triggers” que permiten a otra parte del kernel controlar el LED. El valor por defecto de ambos es ‘mmc0’

El LED se puede encender y apagar usando el archivo “brightness”. La luminosidad mínima es 0, y la máxima es 255. Sin embargo, dado que no hay soporte hardware para la variable de luminosidad, cualquier valor mayor que cero encenderá el LED.

Para encender el LED (durante un instante): `$ echo 1 >/sys/class/leds/led0/brightness`

Para apagar el LED: `$ echo 0 >/sys/class/leds/led0/brightness`

Existen además algunos módulos del kernel que puedes cargar para configurar el LED. Por ejemplo, el módulo ledtrig\_heartbeat hace que parpadee el LED (para terminar el efecto de parpadeo, usa el comando anterior para apagar el LED)

```
$ modprobe ledtrig_heartbeat
```

```
$ echo heartbeat >/sys/class/leds/led0/trigger
```

Finalmente, para revertir los cambios y dejar el LED a su funcionamiento por defecto hay 2 opciones:

- a) Reiniciar la RPI
- b) `$ echo mmc0 >/sys/class/leds/led0/trigger`