



# Informe de la Práctica 1: Programación de la Raspberry Pi 2

Grupo: 01

Nombre y apellidos: Daniel Ruskov Vangelov

Nombre y apellidos: Sergio Hurtado Solorzano

Fecha: 06/10/2020

## 1. ¿Qué pretende la práctica?

Programar la Raspberry Pi 2 desde distintas plataformas, centrándose en los pines GPIO para realizar funciones de E/S de forma usual, mediante threads e interrupciones.

## 2. ¿Qué es y para qué sirve Wiring Pi?

Es una librería que contiene funciones para usar el puerto GPIO, entre otras, a la hora de programar la Raspberry Pi en C.

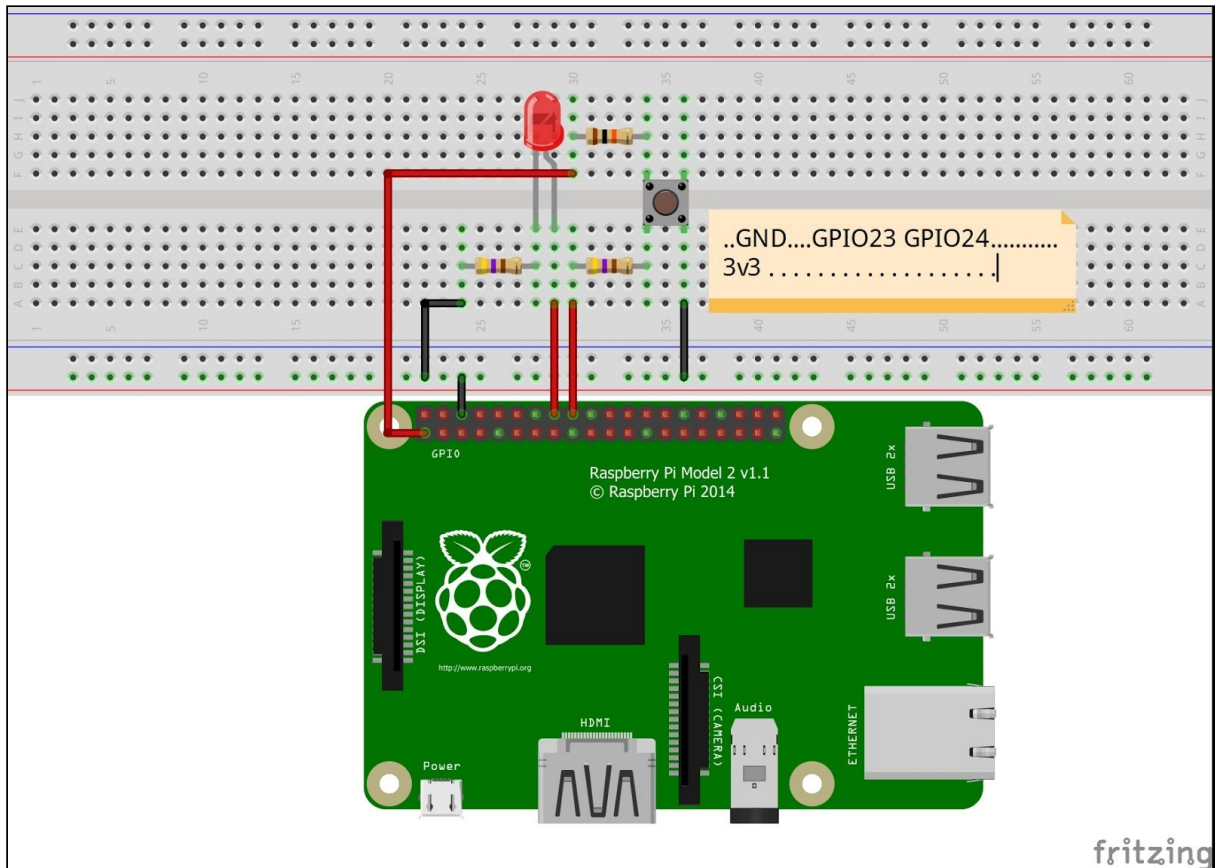
## 3. ¿Qué significa BCM, wPi, Name, Mode, V, Physical en la siguiente tabla?

```
root@robopi01:~# gpio readall
```

-----Pi 2-----											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	IN	1	3	4		5v			
3	9	SCL.1	IN	1	5	6		0v			
4	7	GPIO. 7	IN	1	7	8	1	ALT0	TxD	15	14
		0v			9	10	1	ALT0	RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v			
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23
		3.3v			17	18	1	IN	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v			
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8
		0v			25	26	1	IN	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v			
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v			
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20
		0v			39	40	0	IN	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
-----Pi 2-----											

- BCM - es el código asignado por la placa BCM GPIO
- wPi - código asignado por WiringPi
- Name - función del pin
- Mode - cómo está programado actualmente (IN/OUT)
- V - su estado lógico (1: high/true/, 0, low/false)
- Physical - posición en el conector (impares a la izquierda y pares a la derecha)

4. Haced un esquema del circuito que habéis montado (no una foto), indicando claramente los pines usados.



5. Incluir aquí o en ficheros aparte el código fuente de todos los programas, comentados [poniendo vuestro nombre en la cabecera de todos los programas]

- P1.1 - Saludo

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    char name[20];
    printf("Como te llamas?\n");
    scanf("%s", name);
    printf("Hola %s\n", name);
    exit(0);
}
```

```
root@robopi01:~# gcc -Wall P1.1.c -o P1.1
root@robopi01:~# ./P1.1
Como te llamas?
Root
Hola Root
root@robopi01:~#
```

- P1.2 - Saludo vía NetBeans  
Enlace de descarga proyecto NetBeans exportado con formato .zip:  
[https://drive.google.com/file/d/1\\_hA-Cpe2CEkhPFvWVlxHmjzpY5OES4n5/view?usp=sharing](https://drive.google.com/file/d/1_hA-Cpe2CEkhPFvWVlxHmjzpY5OES4n5/view?usp=sharing)
- P1.3.a - LED intermitente  
Enlace de descarga proyecto NetBeans exportado con formato .zip:  
<https://drive.google.com/file/d/1wgNRxmf23yF7D9dbiJsQ1etTJT3FOMFe/view?usp=sharing>
- P1.3.b - LED controlado mediante pulsador  
Enlace de descarga proyecto NetBeans exportado con formato .zip:  
<https://drive.google.com/file/d/1wS7-ZRDWEtpdFMEUv9chB509lffcSuZ/view?usp=sharing>
- P1.3.c - LED con luminosidad variable controlado mediante pulsador  
Enlace de descarga proyecto NetBeans exportado con formato .zip:  
<https://drive.google.com/file/d/1lcEmMDeuslwBlmS7WrADYxGYc570Zi76/view?usp=sharing>
- P1.4 - Programa con threads  
Enlace de descarga proyecto NetBeans exportado con formato .zip:  
<https://drive.google.com/file/d/1oHmasYsKUD54-lf6KGY3nl31xAAvAF46/view?usp=sharing>

6. ¿Resulta útil usar threads en una aplicación como ésta?  
¿Por qué?

Si, ya que se puede aprovechar su empleo para que, mientras un thread espera un suceso/evento por encuesta activa, otro puede seguir ejecutando otro código.

## 7. ¿Habéis implementado la exclusión mutua? ¿Cómo?

Si, con las funciones de la librería wiringPi relacionadas con multithreading *piLock(int)* y *piUnlock(int)*, de forma que cuando un thread accede al contador para modificar su valor, bloquea el acceso al resto de threads teniendo acceso exclusivamente el. Una vez realizada la operación, vuelve a desbloquear el acceso al resto de threads.

## 8. ¿Qué habéis observado al cambiar la prioridad de los threads?

En caso de acceder en el mismo instante ambos para modificar el contador, primero realiza la operación (y siempre primero el) el de mayor prioridad.

## 9. ¿Qué habría que hacer para cambiar la interrupción que habéis usado a encuesta?

Constantemente verificar si existe un cambio de tensión en el pulsador, guardando el valor actual y el anterior para detectar el cambio. Se puede usar una variable auxiliar como flag y que se restaure su valor una vez realizada la operación tras la interrupción por encuesta.

## 10. Comentarios, decisiones de diseño tomadas, problemas encontrados, posibles mejoras, etc.

- Problemas - despiste en la nomenclatura de la tabla (>gpio readall) mostrada en el apartado 3 de este informe y el cable de extensión de pines GPIO de la Raspberry Pi 2, generando un retraso relevante en el avance de la práctica.
- Mejoras - existen pequeñas mejoras respecto a la estructura del código, pero al tratarse de pocas líneas de código, no se han realizado cambios. No se ha considerado necesario.