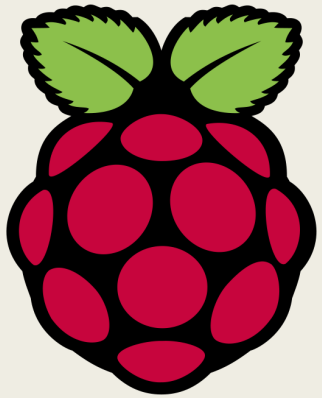
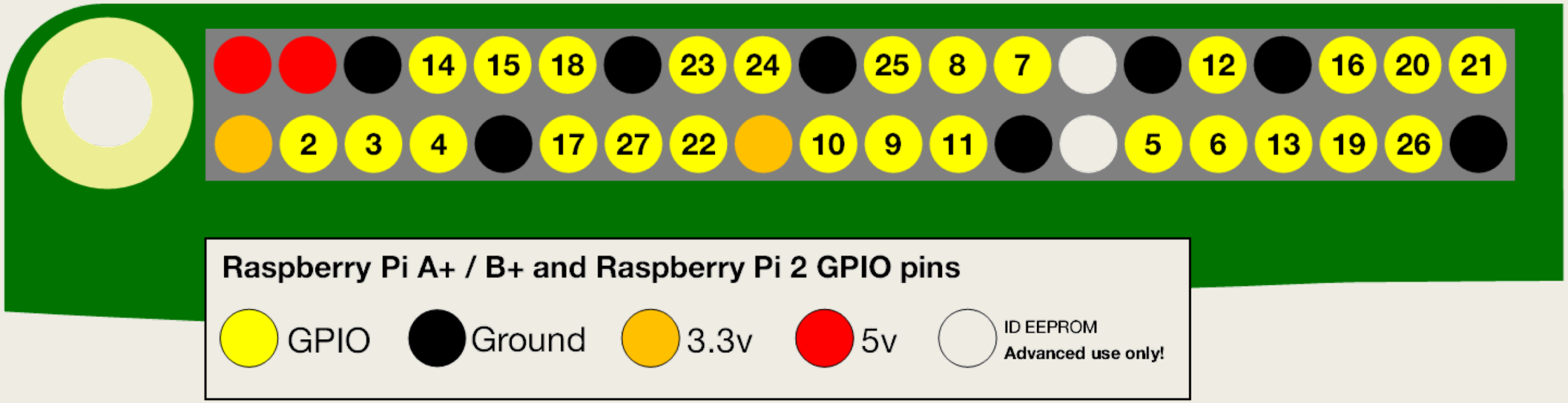
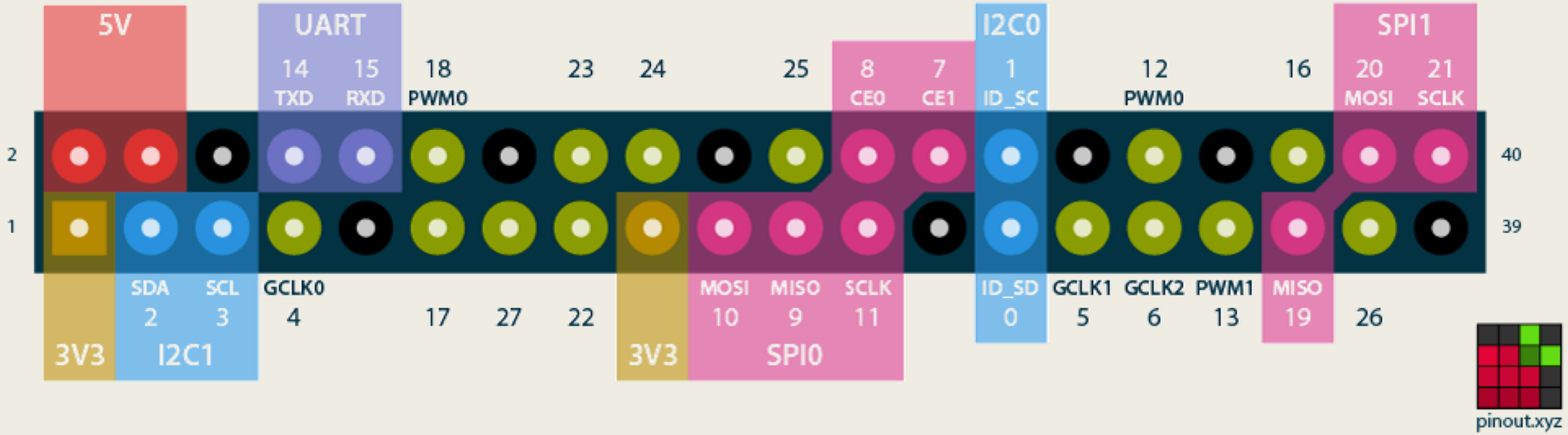
GPIO

朱克剛




14 15 18 23 24 25 8 7 12 16 20 21
2 3 4 17 27 22 10 9 11 5 6 13 19 26
Raspberry Pi A+ / B+ and Raspberry Pi 2 GPIO pins
GPIO
Ground
3.3v
5v
ID EEPROM
Advanced use only!

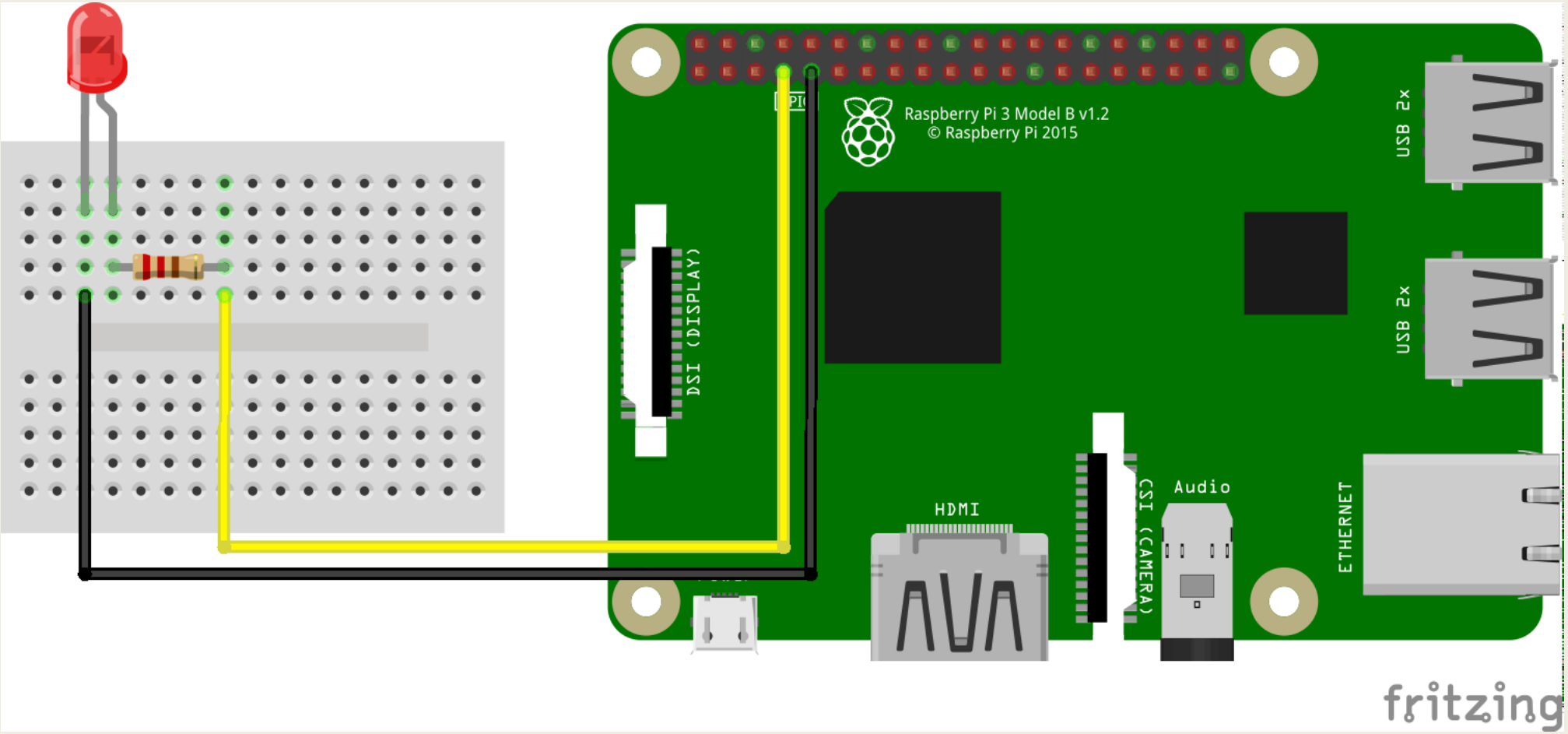
Raspberry Pi GPIO BCM numbering



ckk — pi@humpback: ~/cv — ssh -Y pi@humpback.local — 65...

```
(cv) pi@humpback:~/cv $ pinout
,--------------------------------.
| oo ooo oo oooo o ooo J8     +====
| 1ooo ooooooo ooooooo       P | USB
|  Wi                       ooo +====
|  Fi  Pi Model 3B+ V1.3    ooE    |
|                                  +====
| |D|    ,----.                    | USB
| |S|    |SoC |                    +====
| |I|    `----'                       |
|                       |C|      +======
|                       |S|      |   Net
|  pwr        |HDMI|    |I||A|   +======
`-| |---------|    |----|V|-------'

Revision           : a020d3
SoC                : BCM2837
RAM                : 1024Mb
Storage            : MicroSD
USB ports          : 4 (excluding power)
Ethernet ports     : 1
Wi-fi              : True
Bluetooth          : True
Camera ports (CSI) : 1
Display ports (DSI): 1

J8:
   3V3  (1) (2)  5V
 GPIO2  (3) (4)  5V
 GPIO3  (5) (6)  GND
 GPIO4  (7) (8)  GPIO14
   GND  (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
   3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
 GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO8
   GND (25) (26) GPIO7
 GPIO0 (27) (28) GPIO1
 GPIO5 (29) (30) GND
 GPIO6 (31) (32) GPIO12
GPIO13 (33) (34) GND
GPIO19 (35) (36) GPIO16
GPIO26 (37) (38) GPIO20
   GND (39) (40) GPIO21

For further information, please refer to https://pinout.xyz/
(cv) pi@humpback:~/cv $
```

LED


Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
fritzing

輸出

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(4, GPIO.OUT)    # 設定 GPIO4 為 output
GPIO.output(4, 1)          # 設定 GPIO4 高電位
```

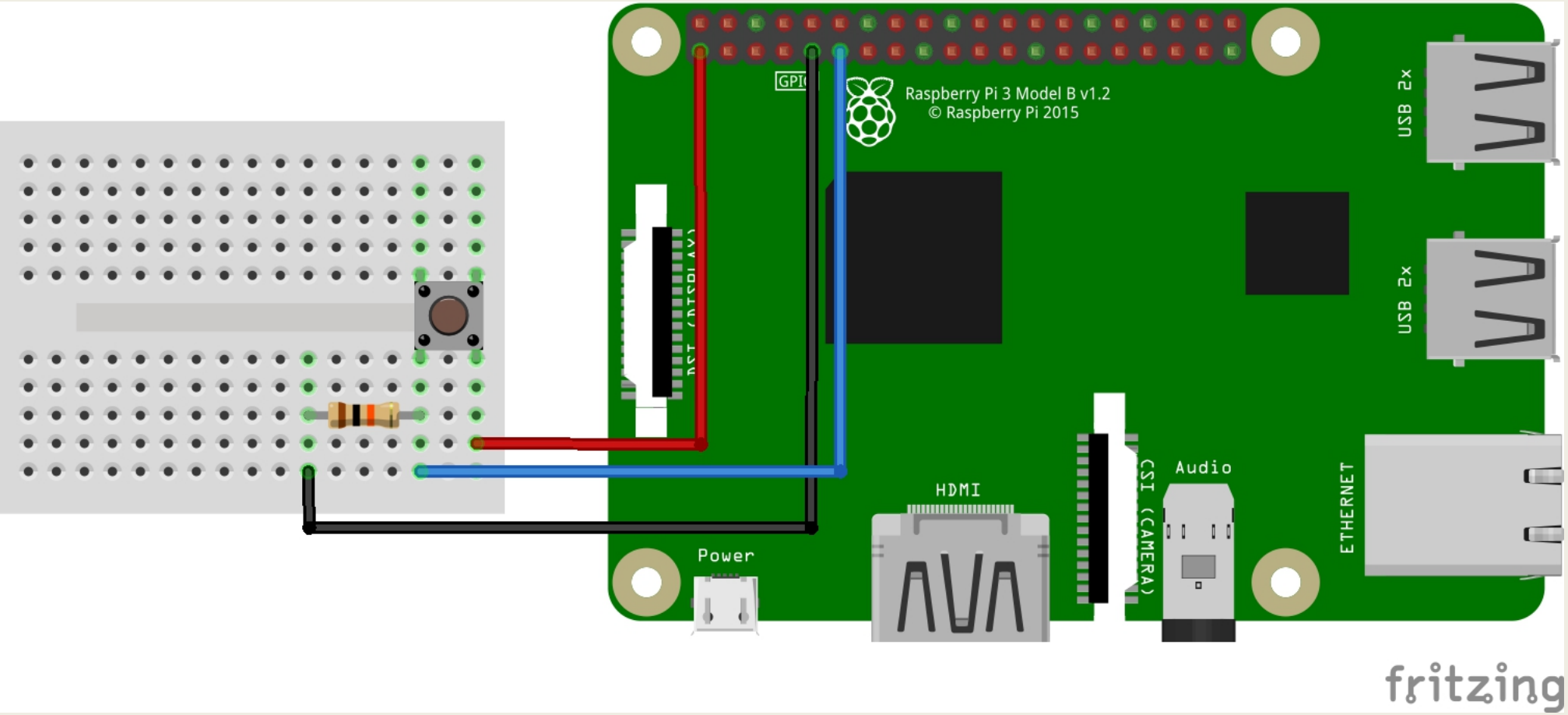
LED閃爍

```
import RPi.GPIO as GPIO
import time

pinLED = 4

GPIO.setmode(GPIO.BCM)
GPIO.setup(pinLED, GPIO.OUT)

try:
   while True:
       GPIO.output(pinLED, 1)
       time.sleep(1)
       GPIO.output(pinLED, 0)
       time.sleep(1)
except KeyboardInterrupt:
   pass

GPIO.cleanup()
```

輸入

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(17, GPIO.IN)      # 設定 GPIO17 為 input
input = GPIO.input(17)       # 讀取 GPIO17 電位

if (input):
    print ("高電位")
```

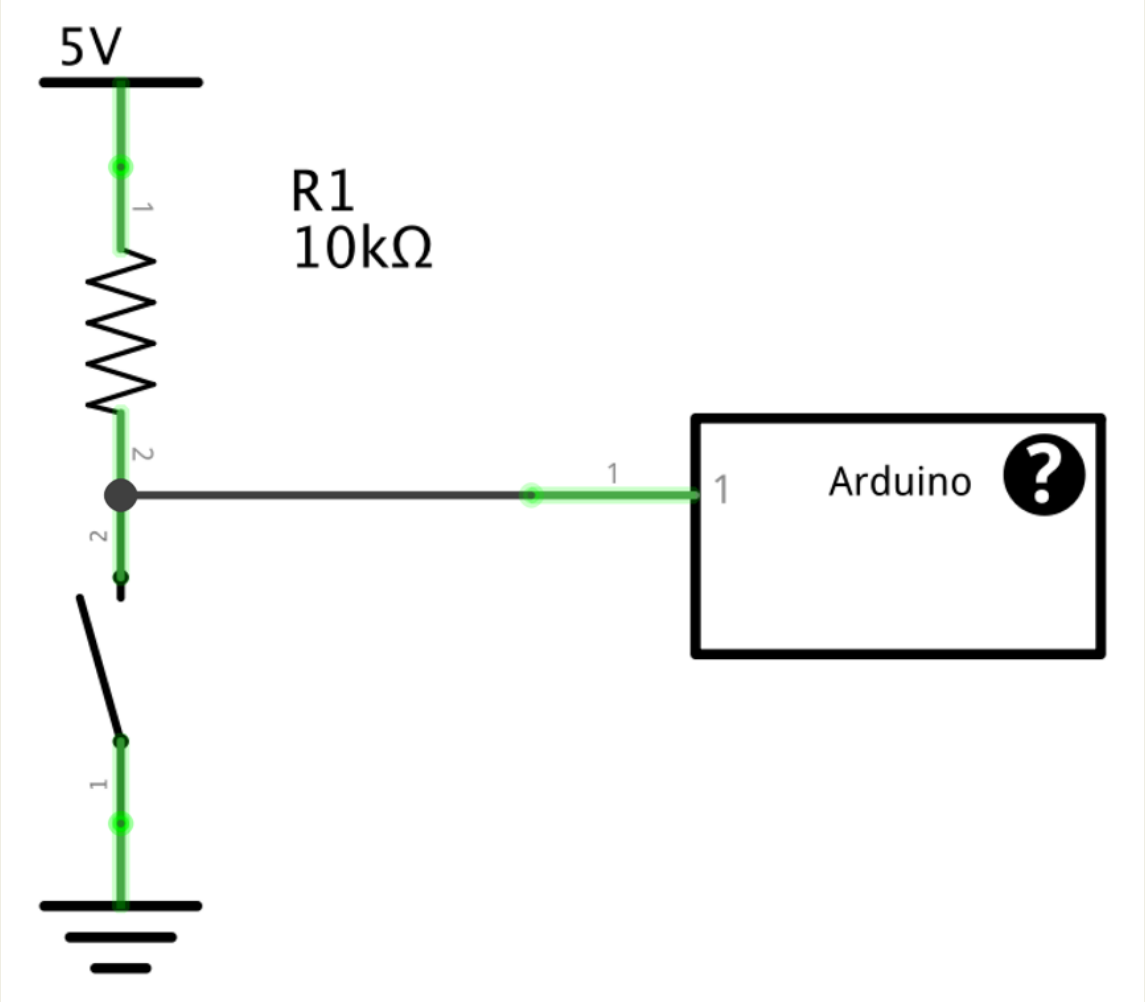
按鈕


Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
HDMI
Audio
CSI (CAMERA)
ETHERNET
Power
fritzing

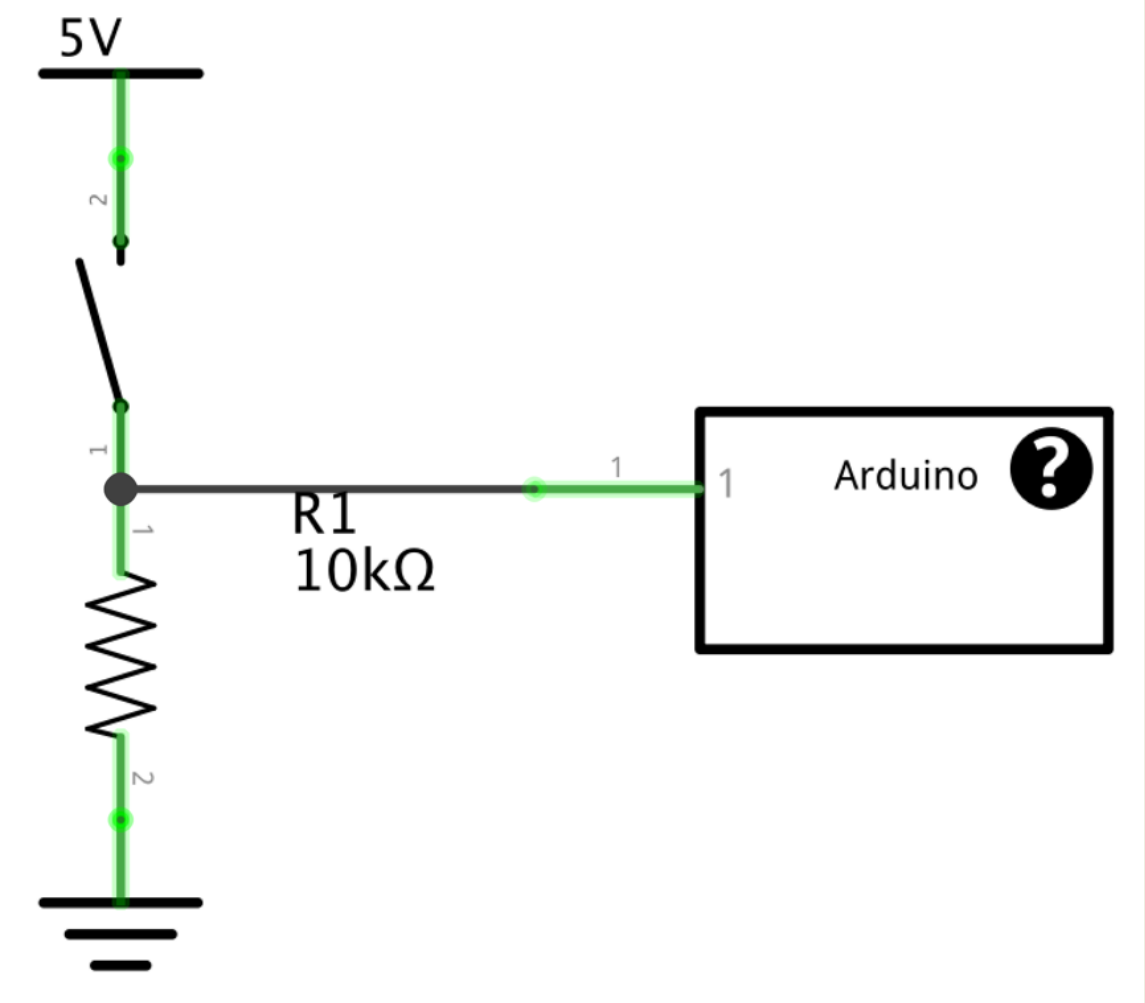
按鈕

```
import RPi.GPIO as GPIO
import time

pinBN = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(pinBN, GPIO.IN)

try:
   while True:
       if GPIO.input(pinBN):
           print ("button down")
       time.sleep(0.1)
except KeyboardInterrupt:
   pass

GPIO.cleanup()
```

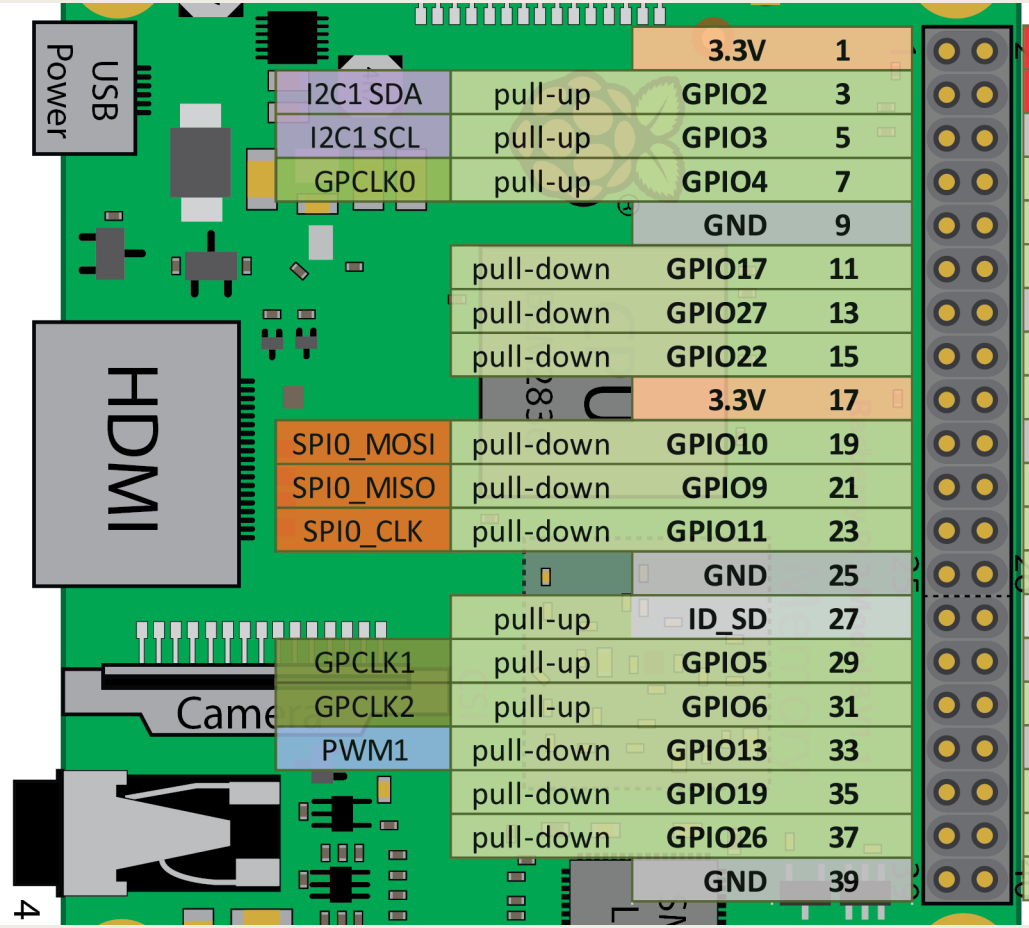
上拉電阻



下拉電阻



GPIO與上、下拉電阻



		3.3V	1		2	5V		
I2C1 SDA	pull-up	GPIO2	3		4	5V		
I2C1 SCL	pull-up	GPIO3	5		6	GND		
GPCLK0	pull-up	GPIO4	7		8	GPIO14	pull-down	TXD0
		GND	9		10	GPIO15	pull-down	RXD0
	pull-down	GPIO17	11		12	GPIO18	pull-down	PWM0
	pull-down	GPIO27	13		14	GND		
	pull-down	GPIO22	15		16	GPIO23	pull-down	
		3.3V	17		18	GPIO24	pull-down	
SPI0_MOSI	pull-down	GPIO10	19		20	GND		
SPI0_MISO	pull-down	GPIO9	21		22	GPIO25	pull-down	
SPI0_CLK	pull-down	GPIO11	23		24	GPIO8	pull-up	SPI_CE0_N
		GND	25		26	GPIO7	pull-up	SPI_CE1_N
	pull-up	ID_SD	27		28	ID_SC	pull-up	
GPCLK1	pull-up	GPIO5	29		30	GND		
GPCLK2	pull-up	GPIO6	31		32	GPIO12	pull-down	PWM0
PWM1	pull-down	GPIO13	33		34	GND		
	pull-down	GPIO19	35		36	GPIO16	pull-down	
	pull-down	GPIO26	37		38	GPIO20	pull-down	GPCLK0
		GND	39		40	GPIO21	pull-down	GPCLK1

啟動內建上拉或下拉電阻

- 啟動 GPIO4 的上拉電阻

```
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```

- 啟動 GPIO25 的下拉電阻

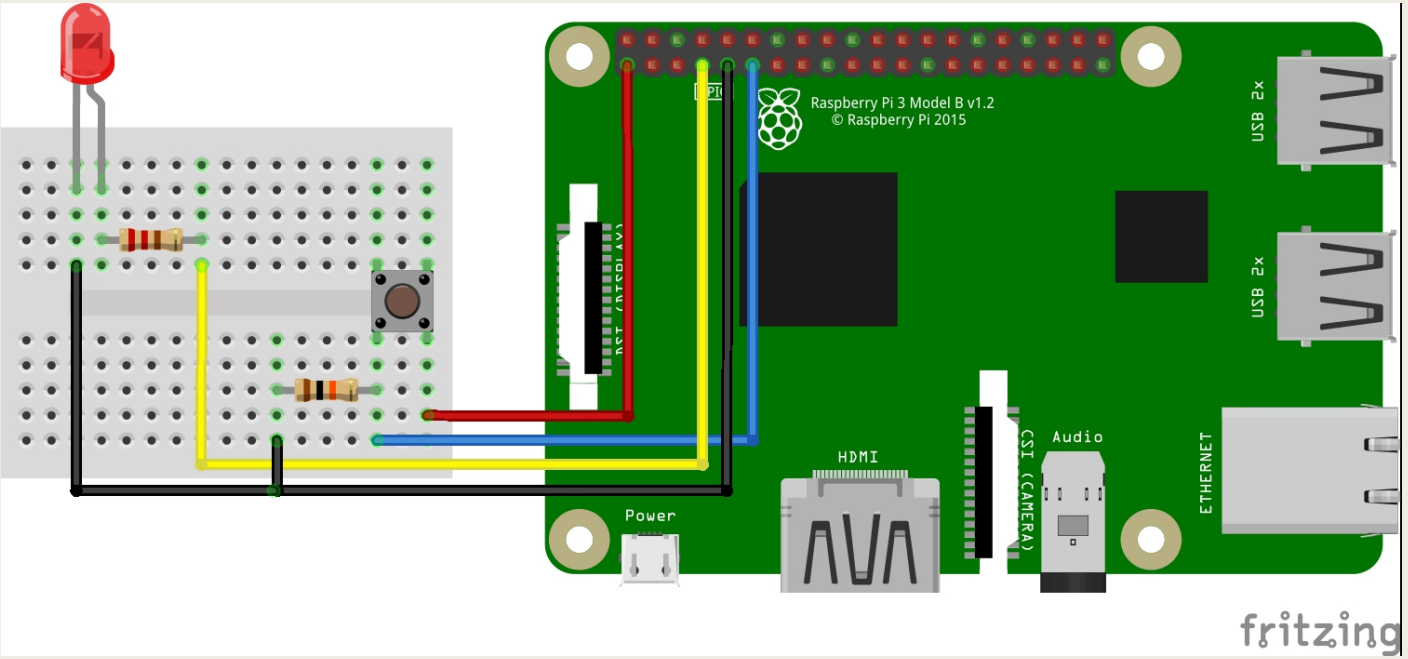
```
GPIO.setup(25, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

練習一

- 偵測按鈕的「click」事件

練習二

- 使用按鈕來控制LED亮滅



中斷

- 如何偵測開關或開關型感測器作動
  - *Polling（輪詢）*
  - *Interrupt（中斷）*
- Polling 方式為使用迴圈，不斷偵測 GPIO input 是否變化
  - *大量消耗 CPU 時間*
  - *加上 sleep 減緩 CPU 負荷*
    - sleep 中無法偵測按鈕變化

wait_for_edge()

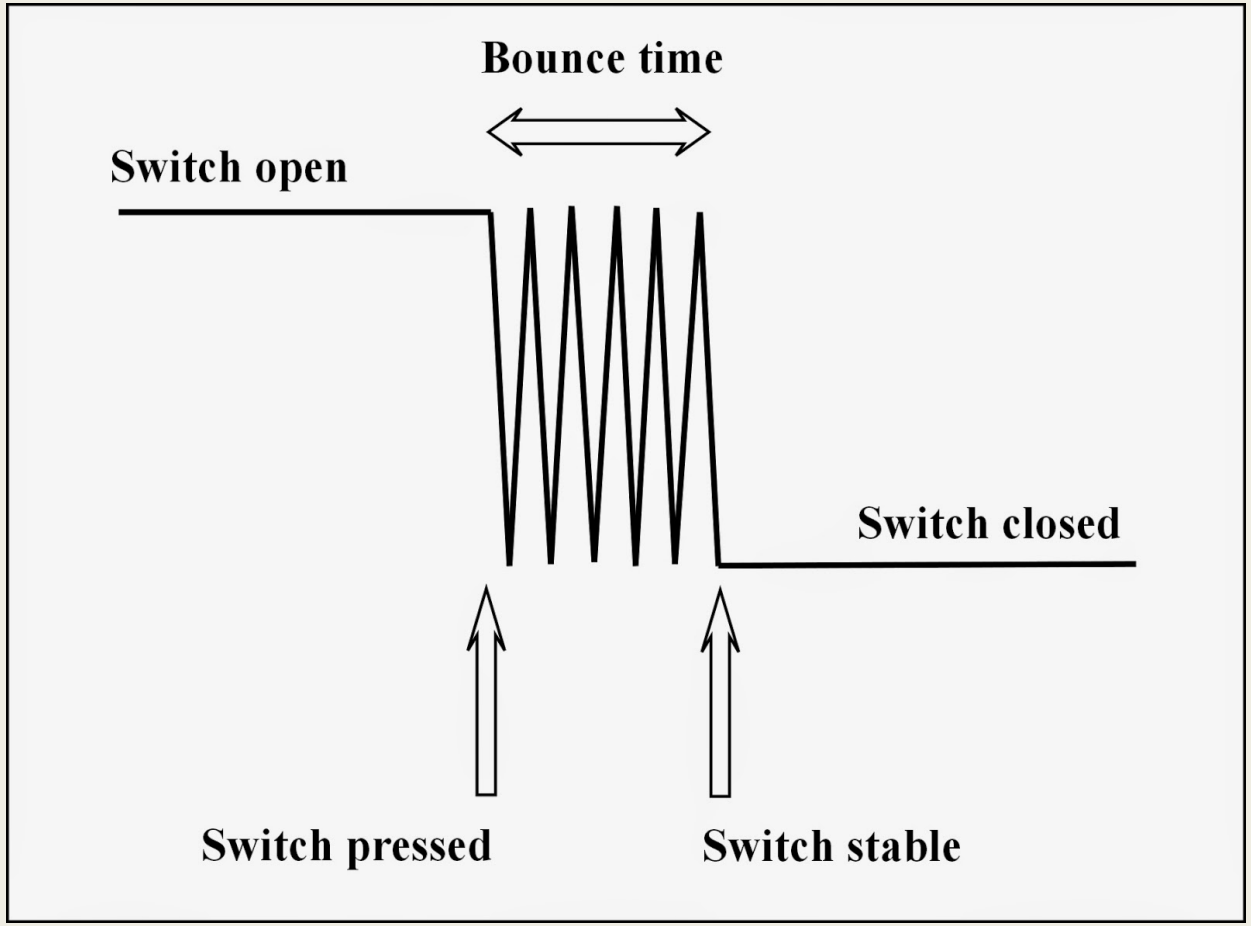
```
channel = GPIO.wait_for_edge(channel, GPIO.RISING, timeout=5000)

if channel is None:
    print ('逾時')
else:
    print ('邊緣觸發，位於 channel', channel)
```

- GPIO.RISING
- GPIO.FALLING
- GPIO.BOTH

event_detected()

```
def my_callback(channel):
    print ('中斷發生')
    pass

GPIO.add_event_detect(channel, GPIO.RISING, callback=my_callback)

while True:
    # do something
    time.sleep(10000000)
```

接點抖動（Switch Bounce）



解除抖動（Switch Debounce）

- 在開關旁接一顆0.1uF電容
- 軟體處理

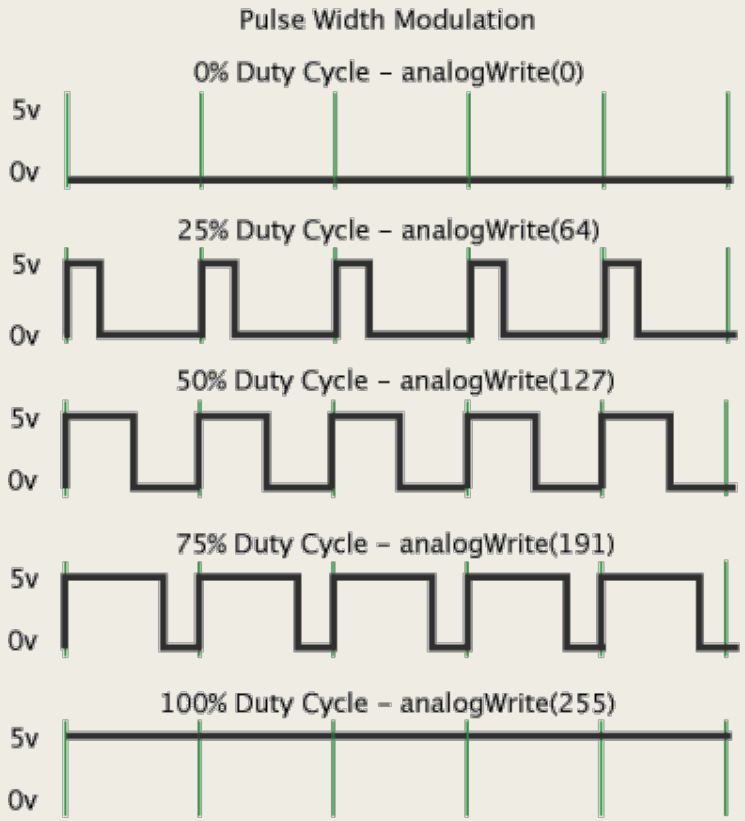
```
GPIO.add_event_detect(channel, GPIO.RISING, callback=my_callback, bouncetime=200)
```

移除事件偵測

```
GPIO.remove_event_detect(channel)
```

脈衝寬度調變PWM

- 透過在一個訊號週期內調整高電位訊號的持續時間來模擬類比訊號輸出的技術
- Duty Cycle（佔空比）
  - *0 ~ 100*
- 讓LED只有50%亮度的方式：
  - *用電阻*
    - 耗電、發熱
  - *用PWM*
    - 省電



硬體與軟體PWM

- 硬體PWM
  - *PWM0：GPIO12、GPIO18*
  - *PWM1：GPIO13*
  - *優點：準確*
- 軟體模擬
  - *任何GPIO OUTPUT接腳均可*
  - *Python 的GPIO lib為軟體模擬*
  - *缺點：慢，準度稍差*

LED每一秒亮滅一次

```
import RPi.GPIO as GPIO

pinPWM = 4
freq = 1
dc = 50

GPIO.setmode(GPIO.BCM)
GPIO.setup(pinPWM, GPIO.OUT)

p = GPIO.PWM(pinPWM, freq)
p.start(dc)

raw_input("Press return to stop:")
p.stop()
GPIO.cleanup()
```

修改頻率與佔空比

- 修改頻率
  - *p.ChangeFrequency(freq)*
- 修改佔空比
  - *p.ChangeDutyCycle(dc)*