



라즈베리파이 - HW1

임베디드스쿨1기

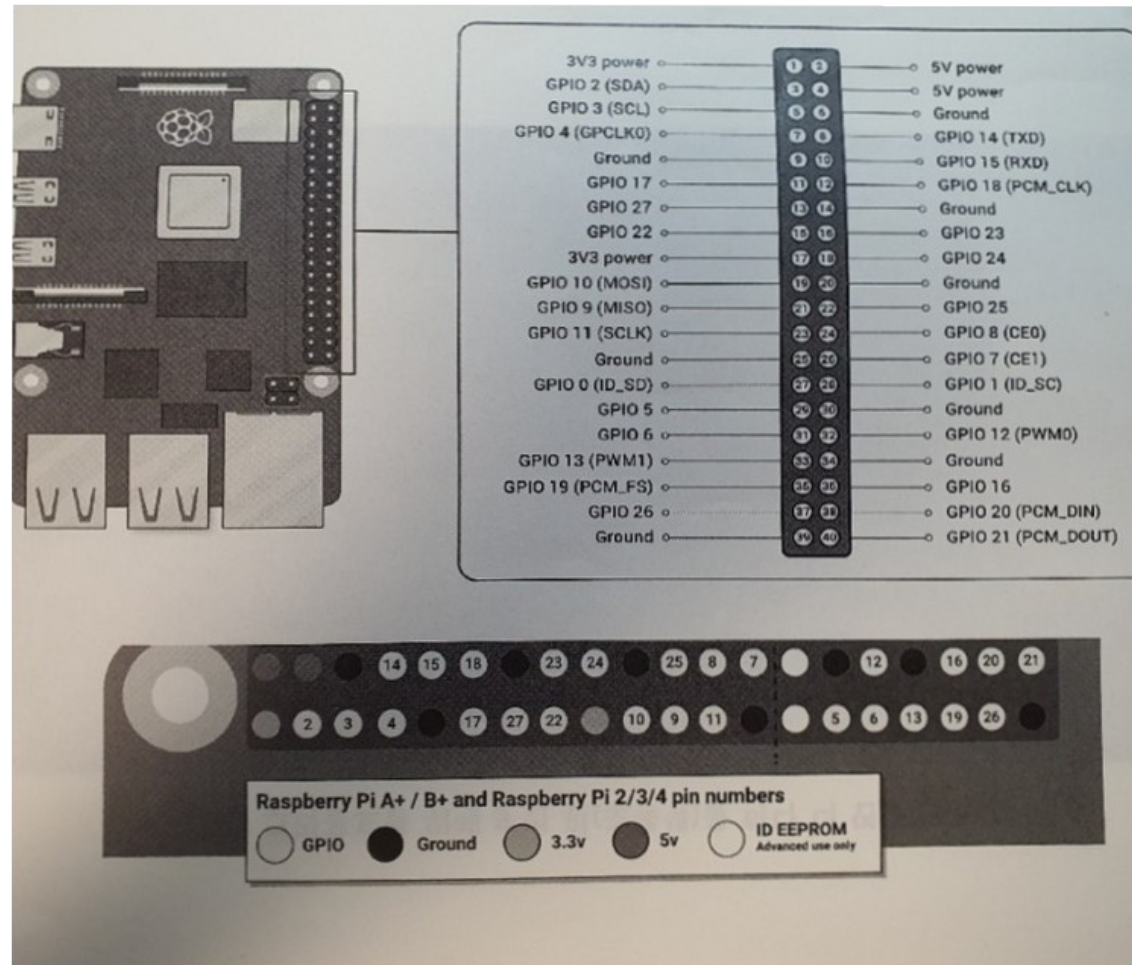
Lv1과정

2020. 09. 22

박하늘

PINMAP

1) Raspberry Pi B - PINMAP



1. LED - Button Event with OC Circuit

- (소스) call_back 함수를 사용하여 Button Rising event가 발생하면 Button Pushed!를 출력한다.
- (회로)공통 그라운드에 9번핀을 하나 더 연결한다. Open Collector를 사용한다.
- (회로)mcu_input_pin에 10번핀을 연결한다. floating을 방지하기 위해 R3에 저항을 단다.

```
import RPi.GPIO as GPIO
import time

#button callback함수
def button_callback(channel):
    print("Button pushed!")

GPIO.setmode(GPIO.BOARD)

#사용할 GPIO핀의 번호 선정
button_pin = 10
led_pin = 8

#불필요한 warning 제거
GPIO.setwarnings(False)

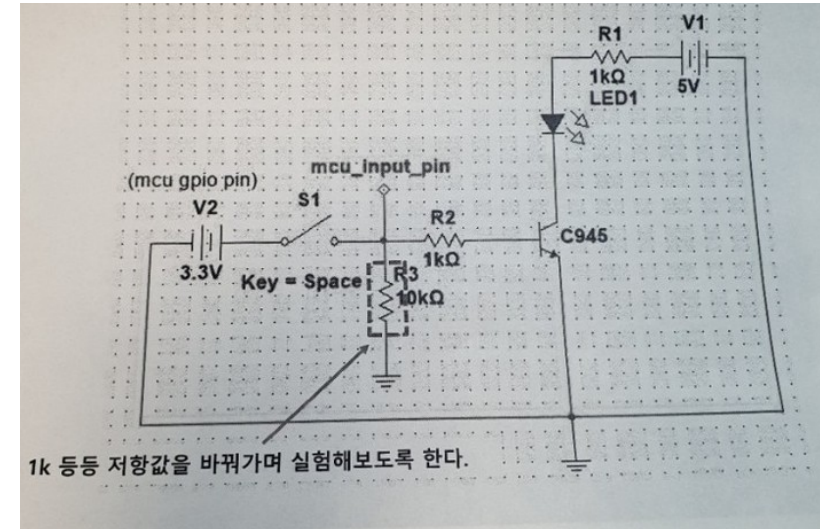
#버튼 핀의 in/out 설정, pull down 설정
GPIO.setup(button_pin, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

#LED
GPIO.setup(led_pin, GPIO.OUT, initial= GPIO.HIGH)

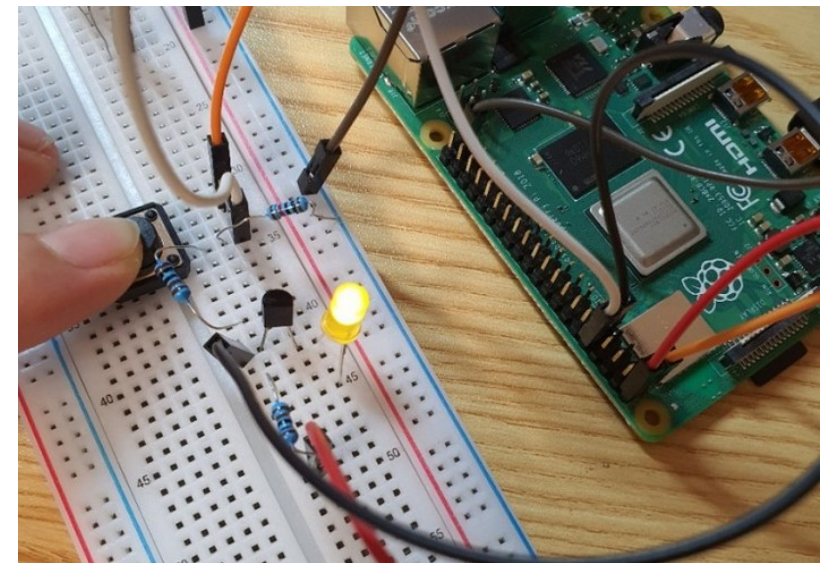
#Event 방식으로 핀의 rising신호를 감지하면 button_callback함수를 실행
GPIO.add_event_detect(button_pin, GPIO.RISING, callback=button_callback)

while 1:
    time.sleep(0.1) #0.1초 딜레이
```

Button pushed!
Button pushed!
Button pushed!
Button pushed!
Button pushed!
Button pushed!



1k 등등 저항값을 바꿔가며 실험해보도록 한다.



2. LED – Button Polling with GPIO

- (소스) 버튼 핀이 GPIO HIGH 신호가 들어오면, Button pushed!가 계속 출력된다.
- (회로) 저항 크기를 바꿔가며 LED 빛 밝기 조절
- 밝기를 더 키우고 싶다면, Open Collector 구성을 한다. (다음장 참고)

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)

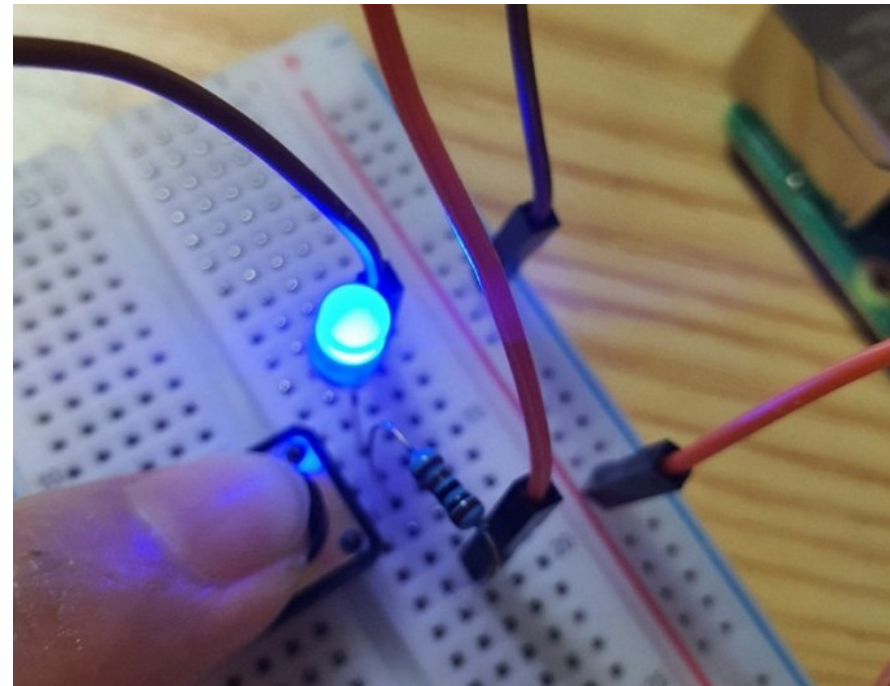
#사용할 GPIO핀의 번호 선정
button_pin = 10
led_pin = 8

#불필요한 warning 제거
GPIO.setwarnings(False)

#버튼 핀의 in/out 설정, pull down 설정
GPIO.setup(button_pin, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

#LED
GPIO.setup(led_pin, GPIO.OUT, initial= GPIO.HIGH)

while 1:
    #만약 버튼핀이 HIGH신호면, print 출력
    if GPIO.input(button_pin) == GPIO.HIGH:
        print("Button pushed!")
        time.sleep(0.1) #0.1초 딜레이
```



3-1. LED – Button Polling with OC Circuit

- (소스) 버튼 핀이 GPIO HIGH 신호가 들어오면, Button pushed!가 계속 출력된다.
- (회로) 공통 그라운드에 9번핀을 하나 더 연결한다. Open Collector를 사용한다.
- (회로) emitter에 220옴의 저항과 LED를 연결한다.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)

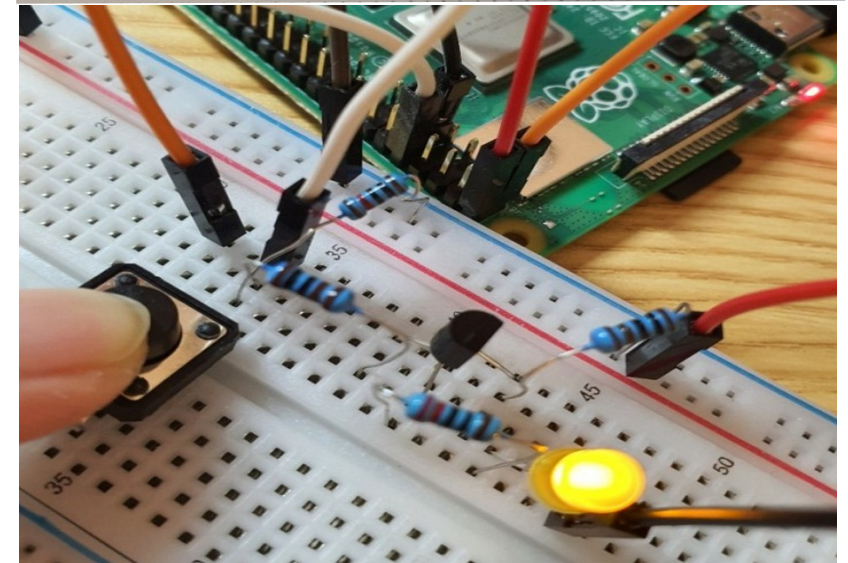
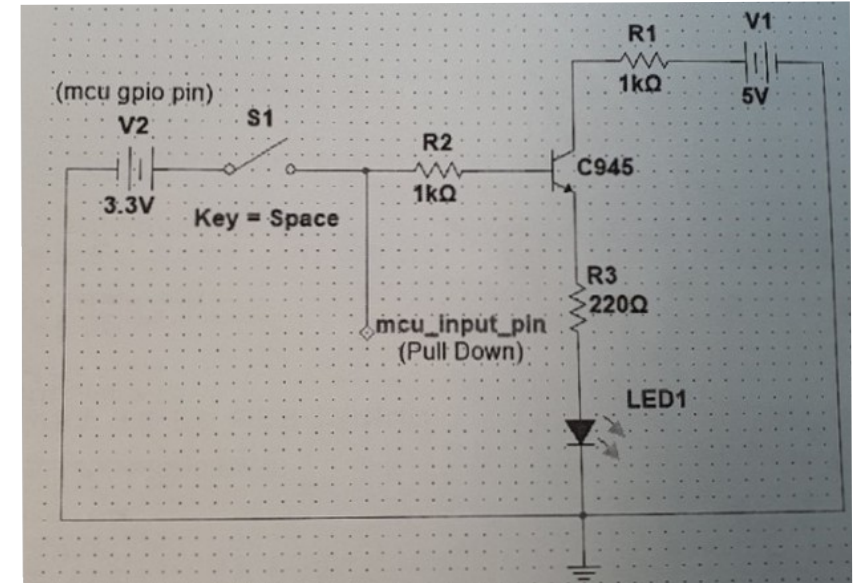
#사용할 GPIO핀의 번호 선정
button_pin = 10
led_pin = 8

#불필요한 warning 제거
GPIO.setwarnings(False)

#버튼 핀의 in/out 설정, pull down 설정
GPIO.setup(button_pin, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)

#LED
GPIO.setup(led_pin, GPIO.OUT, initial= GPIO.HIGH)

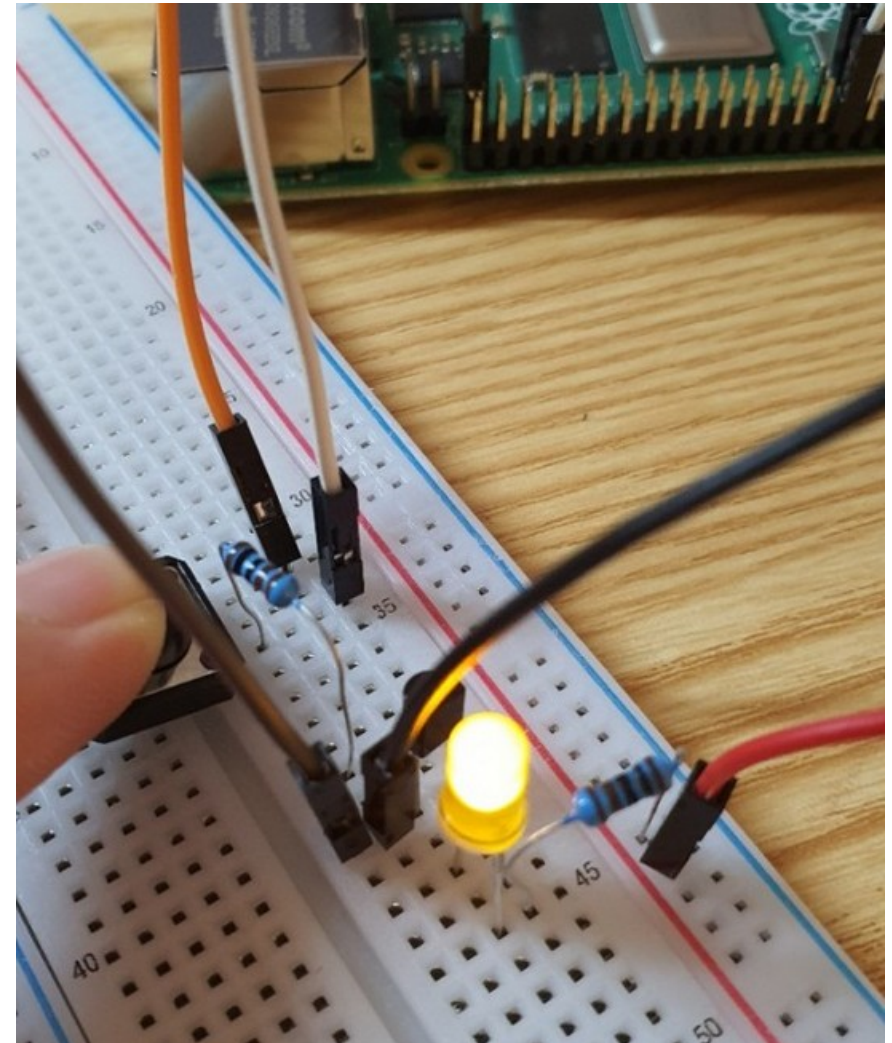
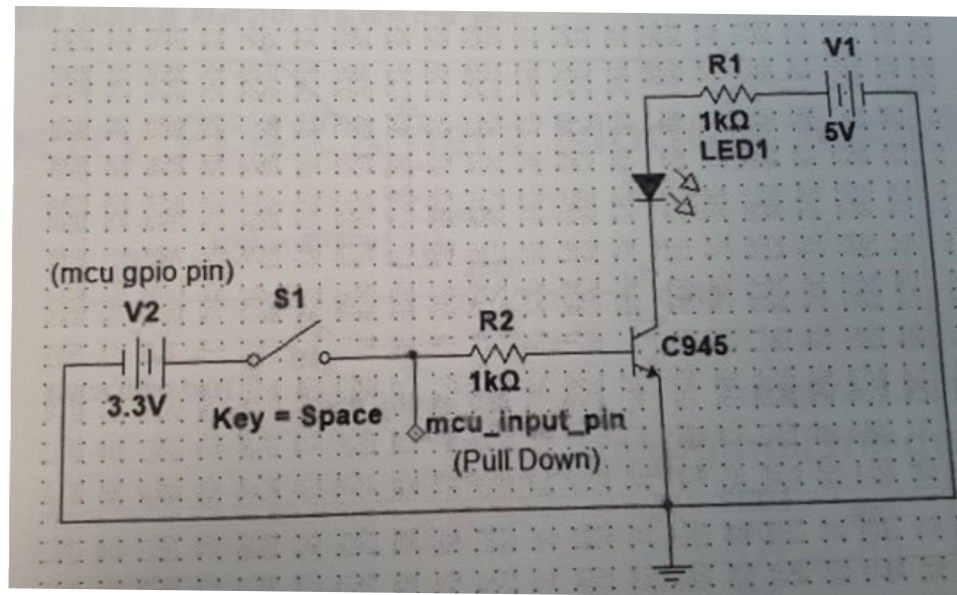
while 1:
    #만약 버튼핀이 HIGH신호면, print 출력
    if GPIO.input(button_pin) == GPIO.HIGH:
        print("Button pushed!")
        time.sleep(0.1) #0.1초 딜레이
```



3-2. LED – Button Polling with OC Circuit

(3-1 장에서 개선된 회로)

- collector 쪽에 LED 와 1K 저항을 연결한다.



3-3. LED – Button Polling with OC Circuit

(3-2장에서 Floating 문제를 개선한 회로)

- mcu_input_pin에 10K인 R3 저항을 연결하여 플로팅 문제 해결한다.

