

```
1 import RPi.GPIO as GPIO # GPIOを利用する
2 import time # sleepを利用する
3
4 # ポート番号の定義
5 SWITCH = 18
6 LED = 21
7 led_value = GPIO.LOW
8 time_now = 0
9 time_start = 0
10
11 # GPIOの初期化
12 GPIO.setmode(GPIO.BCM)
13 GPIO.setup(SWITCH, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
14 GPIO.setup(LED, GPIO.OUT)
15
16
17 # スイッチの切り替えイベント関数の定義
18 def callback_change_switch(ch):
19     global led_value #globalは関数内でグローバル変数を操作可能にする
20     global time_now = time.time()#ボタンを押した時間を記録
21     print("callback", ch)
22     if ch != SWITCH:
23         print("incorrect switch is pushed!")
24         return#呼び出したボタンが違うときに処理をしないためのプログラム
25     time_spent = #穴埋め①：かかった時間を計算
26     print("time", time_spent)#呼び出しの時間（秒）を表示
27     #LEDの状態の切り替え
28     if led_value == GPIO.LOW:
29         led_value = GPIO.HIGH
30     else:
31         led_value = GPIO.LOW
32
33
34
35 GPIO.output(LED, GPIO.LOW)# LEDを消灯しておく
36 time_start = time.time()#処理の開始時間を取得
37
38 # イベント（割り込み）の開始
39 GPIO.add_event_detect(
40     SWITCH, # ポート番号
41     GPIO.RISING, # イベントの種類
42     callback=callback_change_switch, # 関数の指定
43     bouncetime=200) # 連続イベントを制限
44
45
46 try:
47     #メインループ
48     while True:
49         if led_value == GPIO.HIGH:
50             #穴埋め②led_valueがHIGHの状態のとき
51         else:
52             #穴埋め③led_valueがLOWの状態のとき
```

```
53         time.sleep(0.1)
54
55     except KeyboardInterrupt:
56         GPIO.cleanup()
57
58
```