جامعة حلب
كلية الهندسة الكهربائية والالكترونية
مخبر النظم الالكترونية المتقدمة

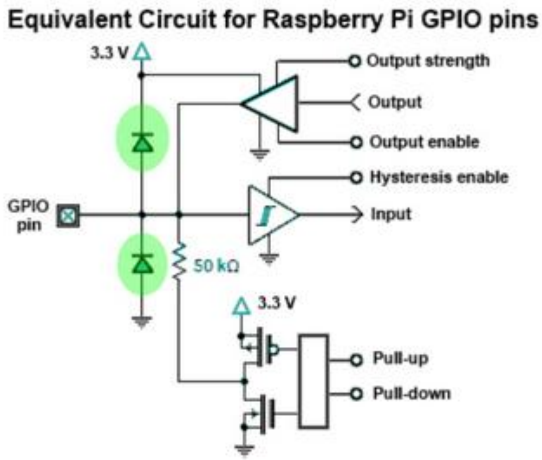
التجربة الخامسة

تطبيقات حول استخدام المقاطعات ضمن RPi.GPIO

الهدف من التجربة:

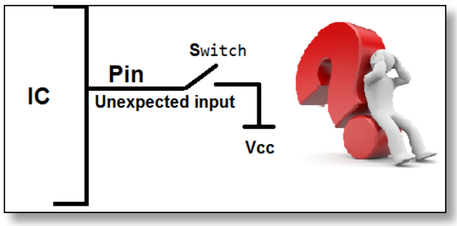
- التعرف على كيفية تفعيل المقاطعات باستخدام لغة بايثون ومكتبة أقطاب التحكم RPi.GPIO.
- التعرف على طرق استخدام توابع Events لكشف الجبهة الصاعدة أو الهابطة.

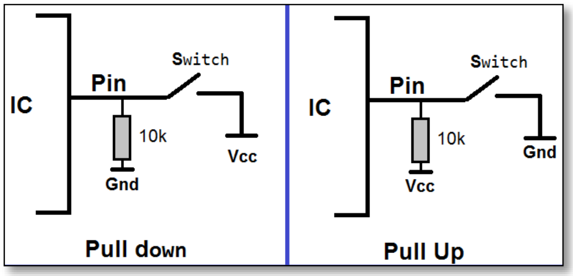
لمحة عامة:



تعرفنا سابقا على بنية العامة لأقطاب اللوحة وتهيئة الأقطاب كمداخل أو مخارج واستخدمنا الطريقة التالية للتصريح عنها:

```
GPIO.setup(12, GPIO.OUT)
GPIO.setup(13, GPIO.IN)
GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)
```



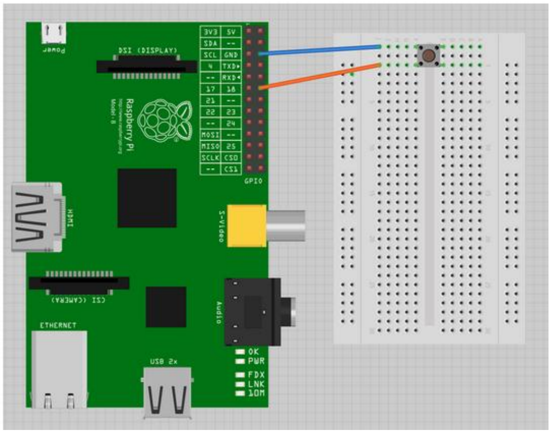


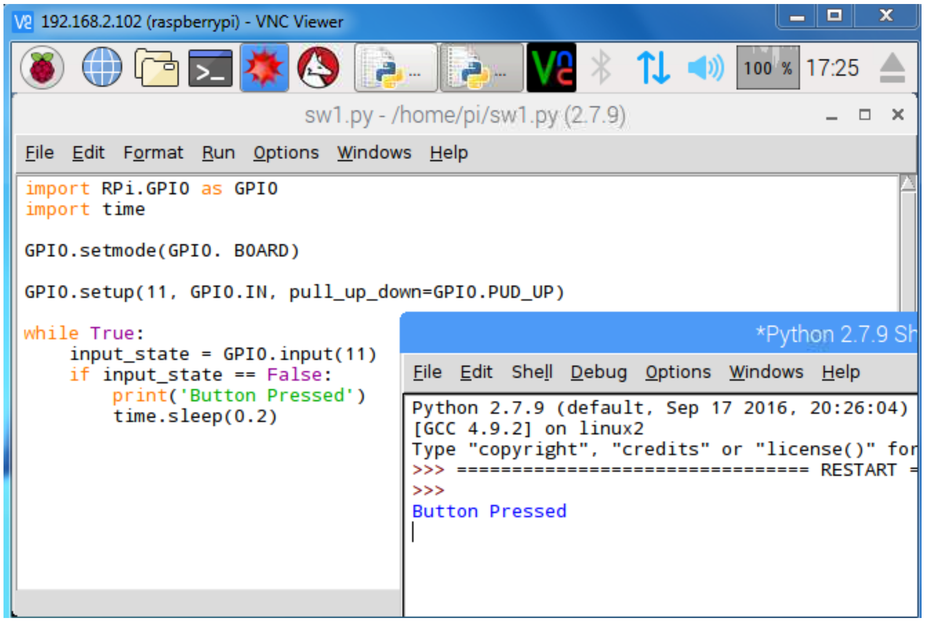
ملاحظة:

نفذ الأمثلة التالية لفحص حالة دخل محدد، واكتب النتائج التي استخلصتها من كل مثال.

مثال 1:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO. BOARD)

GPIO.setup(11, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
    input_state = GPIO.input(11)
    if input_state == False:
        print('Button Pressed')
        time.sleep(0.2)
```



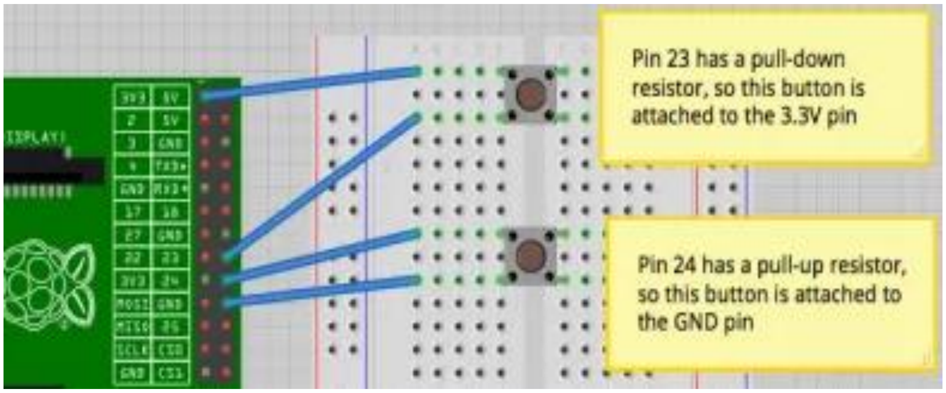

192.168.2.102 (raspberrypi) - VNC Viewer
100 %
17:25
sw1.py - /home/pi/sw1.py (2.7.9)
File Edit Format Run Options Windows Help
*Python 2.7.9 Sh
File Edit Shell Debug Options Windows Help

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO. BOARD)

GPIO.setup(11, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
    input_state = GPIO.input(11)
    if input_state == False:
        print('Button Pressed')
        time.sleep(0.2)
```

```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for
>>> ================================ RESTART =
>>>
Button Pressed
```

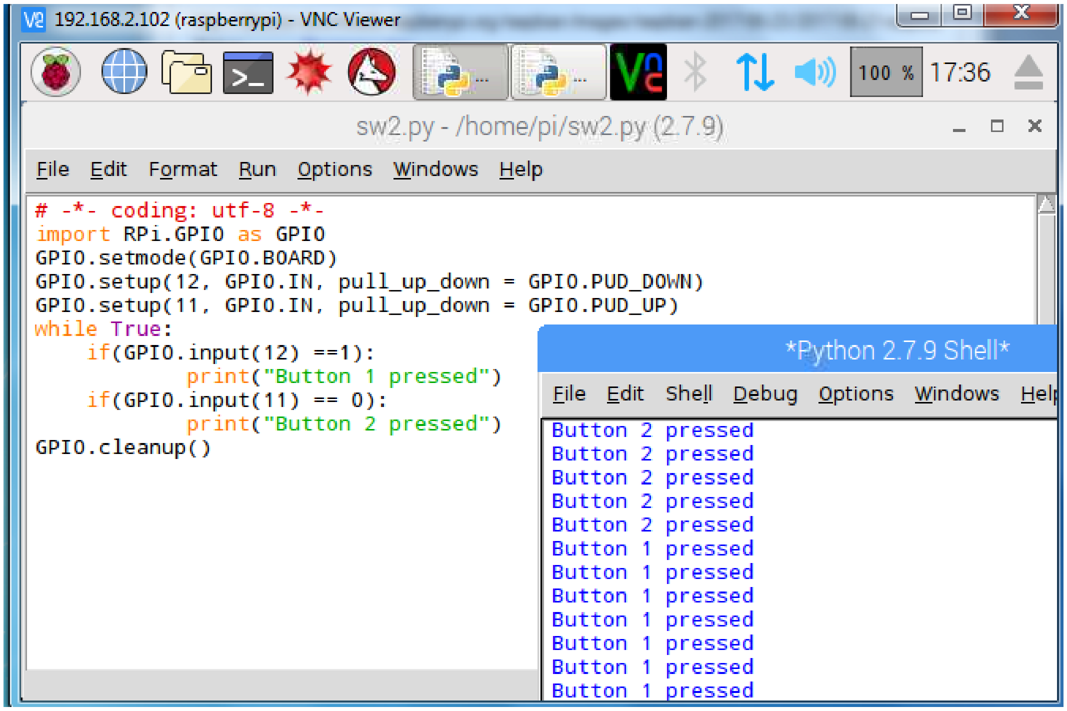
مثال 2:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)
while True:
 if(GPIO.input(23) ==1):
   print(“Button 1 pressed”)
 if(GPIO.input(24) == 0):
   print(“Button 2 pressed”)
GPIO.cleanup()
```



Pin 23 has a pull-down resistor, so this button is attached to the 3.3V pin

Pin 24 has a pull-up resistor, so this button is attached to the GND pin


192.168.2.102 (raspberrypi) - VNC Viewer
100 %
17:36
sw2.py - /home/pi/sw2.py (2.7.9)
File Edit Format Run Options Windows Help
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help

```
# -*- coding: utf-8 -*-
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(11, GPIO.IN, pull_up_down = GPIO.PUD_UP)
while True:
    if(GPIO.input(12) ==1):
            print("Button 1 pressed")
    if(GPIO.input(11) == 0):
            print("Button 2 pressed")
GPIO.cleanup()
```

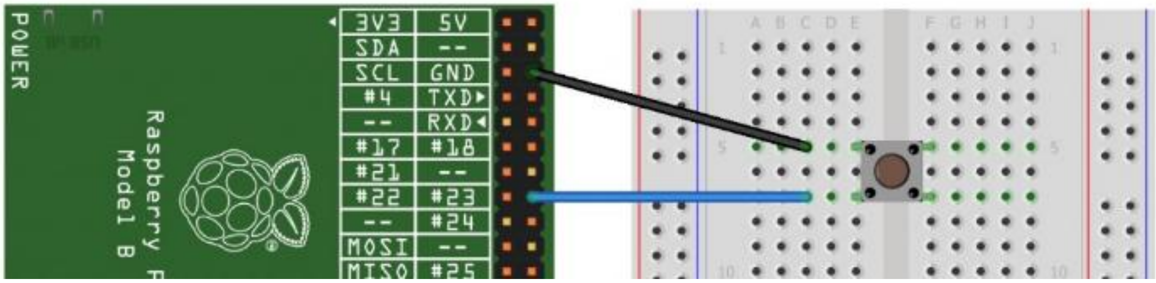
```
Button 2 pressed
Button 2 pressed
Button 2 pressed
Button 2 pressed
Button 2 pressed
Button 1 pressed
Button 1 pressed
Button 1 pressed
Button 1 pressed
Button 1 pressed
Button 1 pressed
Button 1 pressed
```

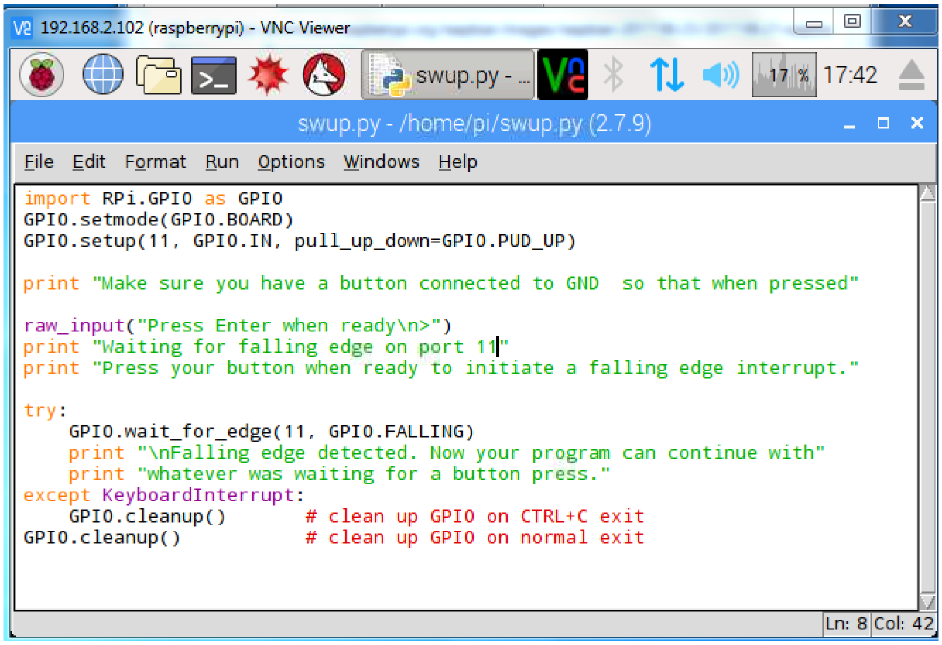
التعرف على طرق استخدام توابع كشف الجبهة الصاعدة أو الهابطة:

1- الطريقة الأولى باستخدام wait_for_edge :

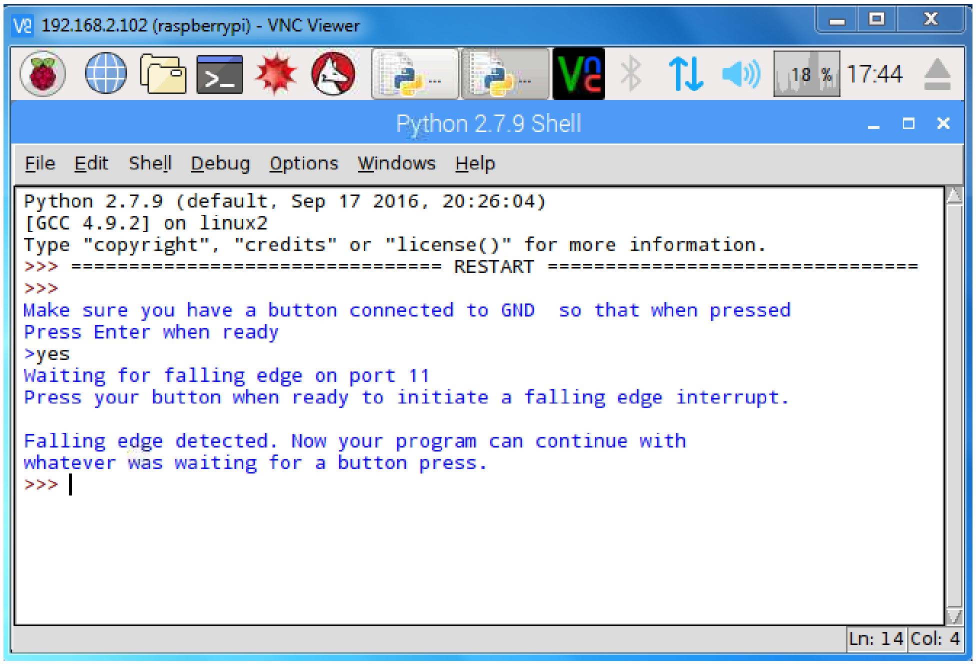
مثال 3:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)
print ("Make sure you have a button connected
to GND  so that when pressed")
input("Press Enter when ready\n>")
print ("Waiting for falling edge on port 23")
print ("Press your button when ready to initiate a falling edge interrupt.")

try:
    GPIO.wait_for_edge(23, GPIO.FALLING)
    print ("\nFalling edge detected. Now your program can continue with")
    print ("whatever was waiting for a button press.")
except KeyboardInterrupt:
    GPIO.cleanup()       # clean up GPIO on CTRL+C exit
GPIO.cleanup()           # clean up GPIO on normal exit
```




192.168.2.102 (raspberrypi) - VNC Viewer
swup.py - ...
17 %
17:42
swup.py - /home/pi/swup.py (2.7.9)
File Edit Format Run Options Windows Help
Ln: 8 Col: 42

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.IN, pull_up_down=GPIO.PUD_UP)

print "Make sure you have a button connected to GND  so that when pressed"

raw_input("Press Enter when ready\n>")
print "Waiting for falling edge on port 11"
print "Press your button when ready to initiate a falling edge interrupt."

try:
    GPIO.wait_for_edge(11, GPIO.FALLING)
    print "\nFalling edge detected. Now your program can continue with"
    print "whatever was waiting for a button press."
except KeyboardInterrupt:
    GPIO.cleanup()       # clean up GPIO on CTRL+C exit
GPIO.cleanup()           # clean up GPIO on normal exit
```

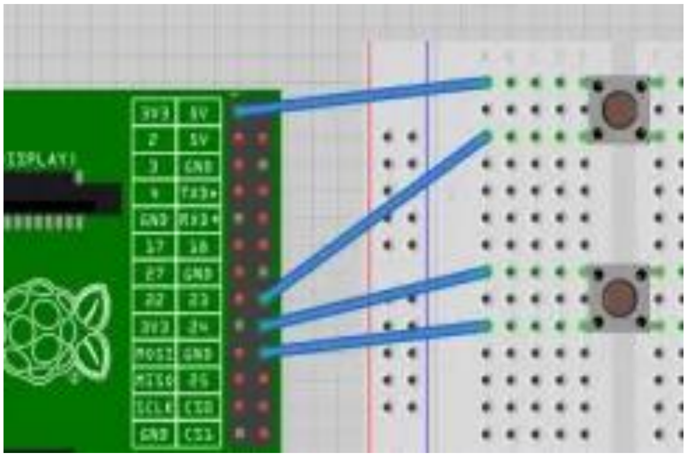

192.168.2.102 (raspberrypi) - VNC Viewer
18 %
17:44
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Ln: 14 Col: 4

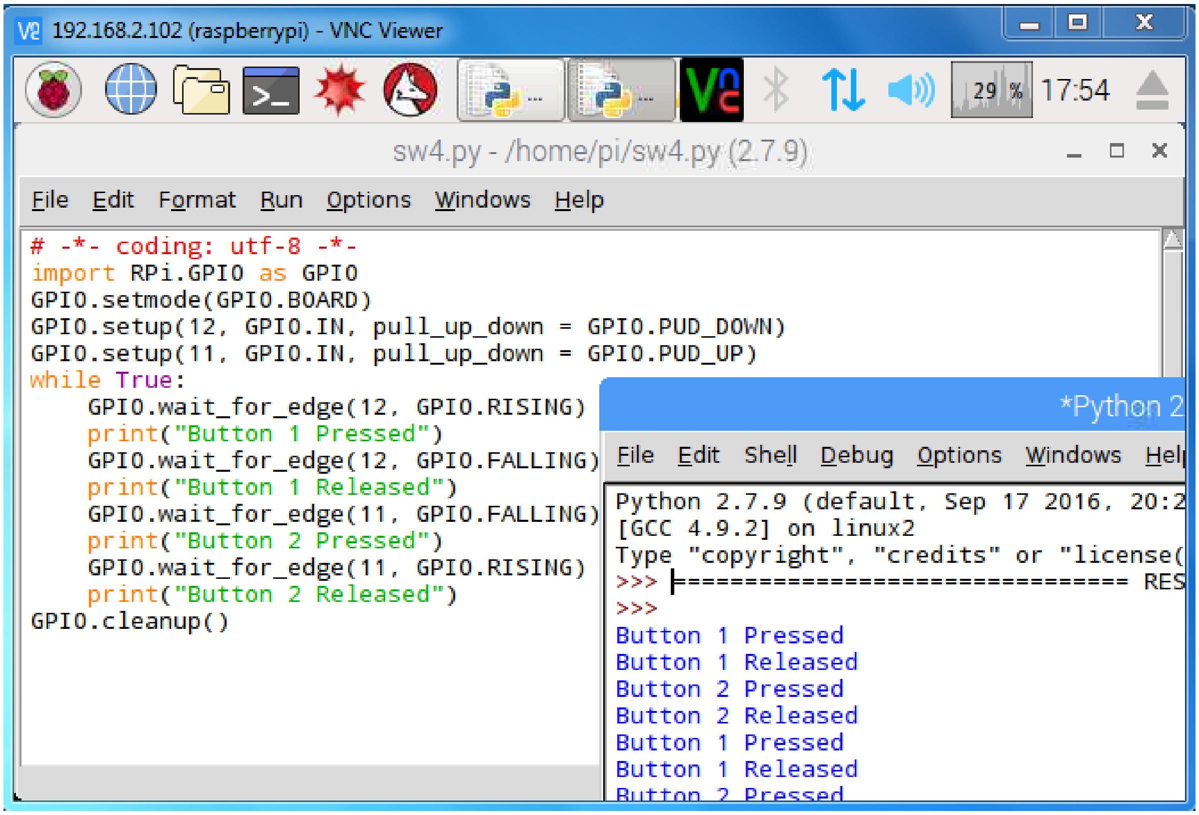
```
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>> 
Make sure you have a button connected to GND  so that when pressed
Press Enter when ready
>yes
Waiting for falling edge on port 11
Press your button when ready to initiate a falling edge interrupt.

Falling edge detected. Now your program can continue with
whatever was waiting for a button press.
>>> 
```

مثال 4:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)
while True:
GPIO.wait_for_edge(23, GPIO.RISING)
print(“Button 1 Pressed”)
```

```
GPIO.wait_for_edge(23, GPIO.FALLING)
print("Button 1 Released")
GPIO.wait_for_edge(24, GPIO.FALLING)
print("Button 2 Pressed")
GPIO.wait_for_edge(24, GPIO.RISING)
print("Button 2 Released")
GPIO.cleanup()
```

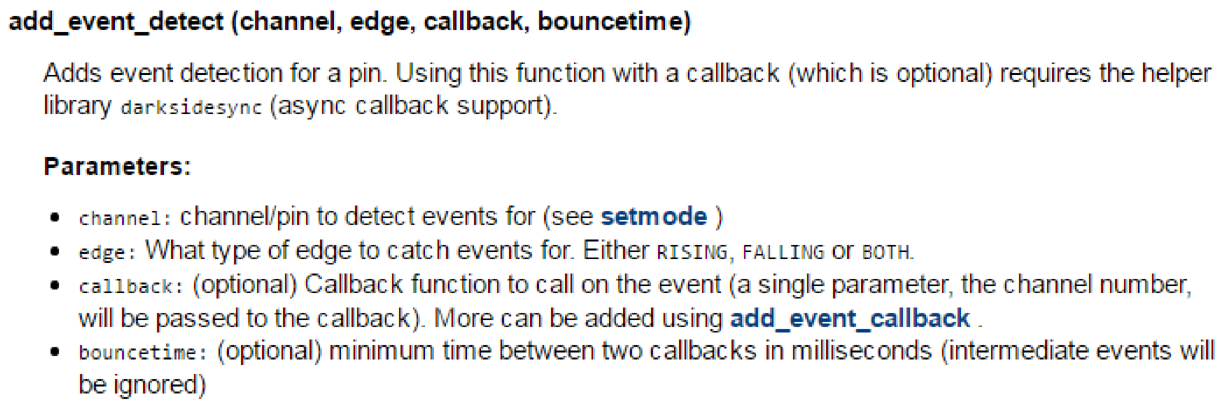



192.168.2.102 (raspberrypi) - VNC Viewer
29 %
17:54
sw4.py - /home/pi/sw4.py (2.7.9)
File Edit Format Run Options Windows Help
*Python 2
File Edit Shell Debug Options Windows Help

```
# -*- coding: utf-8 -*-
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(11, GPIO.IN, pull_up_down = GPIO.PUD_UP)
while True:
    GPIO.wait_for_edge(12, GPIO.RISING)
    print("Button 1 Pressed")
    GPIO.wait_for_edge(12, GPIO.FALLING)
    print("Button 1 Released")
    GPIO.wait_for_edge(11, GPIO.FALLING)
    print("Button 2 Pressed")
    GPIO.wait_for_edge(11, GPIO.RISING)
    print("Button 2 Released")
GPIO.cleanup()
```

```
Python 2.7.9 (default, Sep 17 2016, 20:2
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license(
>>> ================================ RES
>>>
Button 1 Pressed
Button 1 Released
Button 2 Pressed
Button 2 Released
Button 1 Pressed
Button 1 Released
Button 2 Pressed
```

2- الطريقة الثانية باستخدام add_event_detect:



add_event_detect (channel, edge, callback, bouncetime)

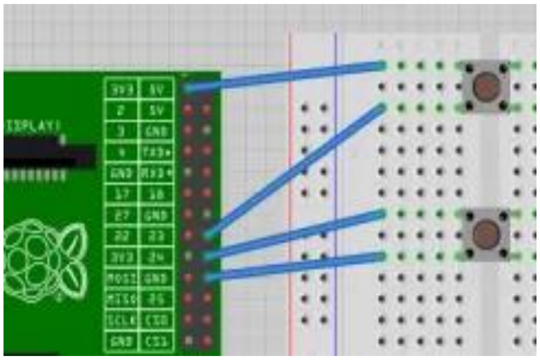
Adds event detection for a pin. Using this function with a callback (which is optional) requires the helper library `darksidesync` (async callback support).

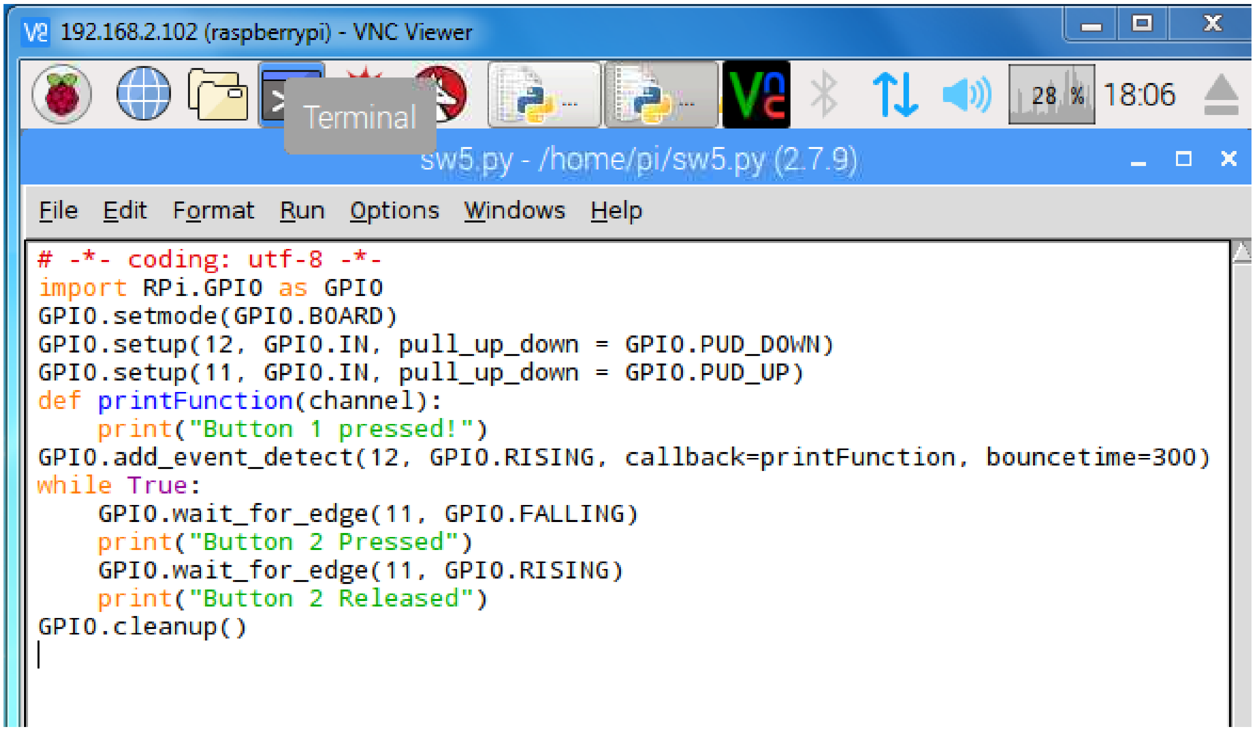
Parameters:

- `channel:` channel/pin to detect events for (see **setmode**)
- `edge:` What type of edge to catch events for. Either `RISING`, `FALLING` or `BOTH`.
- `callback:` (optional) Callback function to call on the event (a single parameter, the channel number, will be passed to the callback). More can be added using **add_event_callback** .
- `bouncetime:` (optional) minimum time between two callbacks in milliseconds (intermediate events will be ignored)

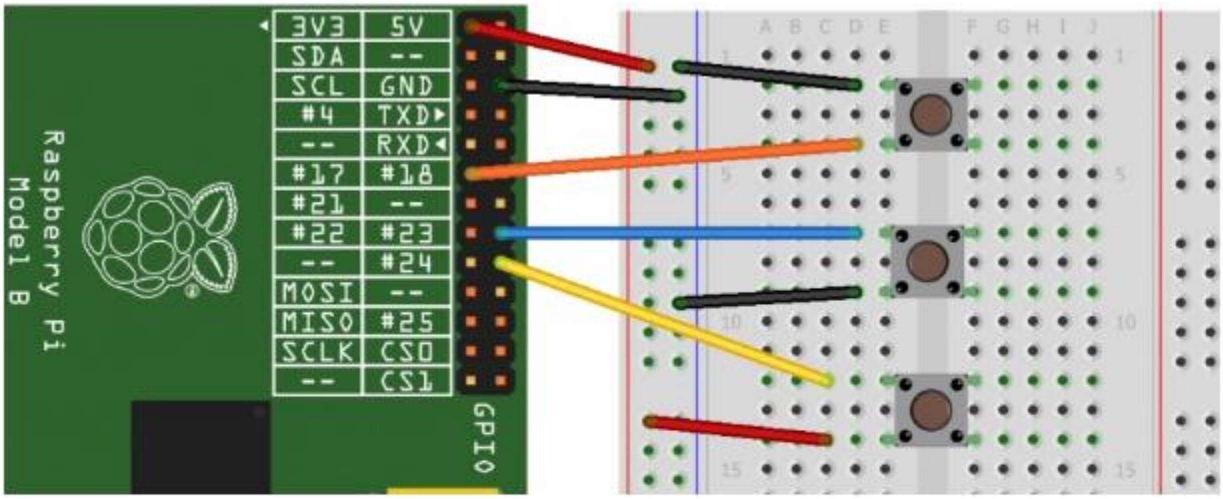
مثال 5:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(11, GPIO.IN, pull_up_down = GPIO.PUD_UP)
def printFunction(channel):
    print("Button 1 pressed"!)

GPIO.add_event_detect(12, GPIO.RISING, callback=printFunction,
bouncetime=300)

while True:
    GPIO.wait_for_edge(11, GPIO.FALLING)
    print("Button 2 Pressed")
    GPIO.wait_for_edge(11, GPIO.RISING)
    print("Button 2 Released")
GPIO.cleanup()
```




192.168.2.102 (raspberrypi) - VNC Viewer
Terminal
28 %
18:06
sw5.py - /home/pi/sw5.py (2.7.9)
File
Edit
Format
Run
Options
Windows
Help

```
# -*- coding: utf-8 -*-
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(11, GPIO.IN, pull_up_down = GPIO.PUD_UP)
def printFunction(channel):
    print("Button 1 pressed!")
GPIO.add_event_detect(12, GPIO.RISING, callback=printFunction, bouncetime=300)
while True:
    GPIO.wait_for_edge(11, GPIO.FALLING)
    print("Button 2 Pressed")
    GPIO.wait_for_edge(11, GPIO.RISING)
    print("Button 2 Released")
GPIO.cleanup()
```

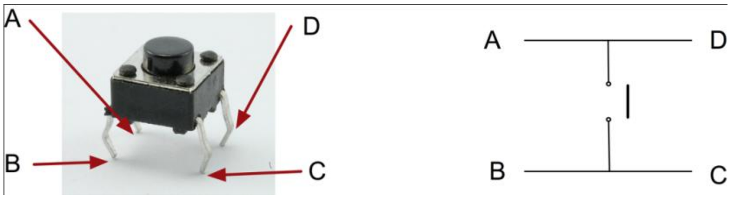


تطبيق حول حالة (multiple threaded call back) لمراقبة مداخل متعددة:

<u>مثال 6:</u>

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)

GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)

GPIO.setup(11, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def my_callback(channel):

    print ("falling edge detected on 11")

def my_callback2(channel):

    print "falling edge detected on 23"

input("Press Enter when ready\n>")

GPIO.add_event_detect(11, GPIO.FALLING, callback=my_callback, bouncetime=300)

GPIO.add_event_detect(23,GPIO.FALLING, callback=my_callback2, bouncetime=300)

try:

    GPIO.wait_for_edge(12, GPIO.RISING)

    print ("Rising edge detected on port 12")

except KeyboardInterrupt:

    GPIO.cleanup()        # clean up GPIO on CTRL+C exit

GPIO.cleanup()            # clean up GPIO on normal exit
```

ملاحظات:



Functions

cleanup ()	Cleans up the modules' running operations.
gpio_function (channel)	Gets the configuration of a pin.
input (channel)	Reads the pin value.
output (channel, value)	Sets the output of a pin.
setmode (mode)	Sets the pin numbering scheme to be used.
setup_channel (channel, direction, pull_up_down, initial)	Sets a channel up on the GPIO interface.
setwarnings (mode)	Turns warnings on or off.