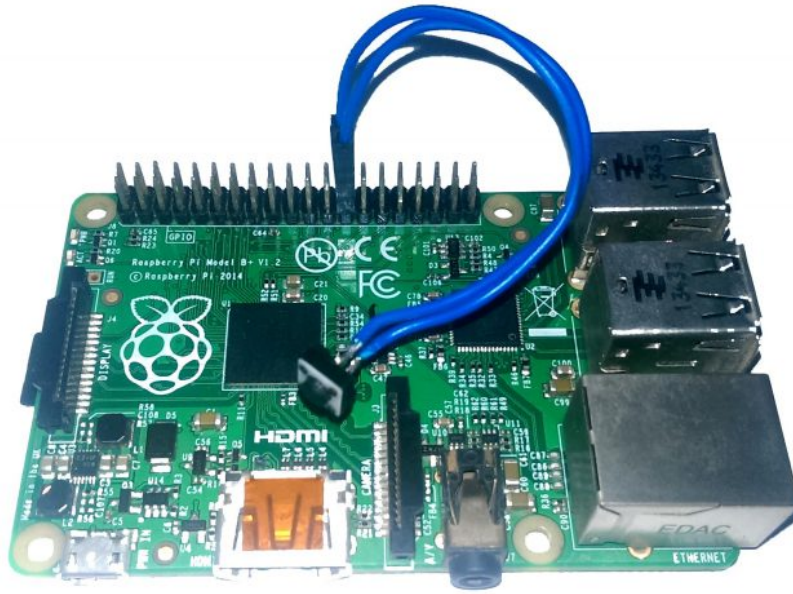


زر إغلاق الـ Raspberry Pi

في هذا الدرس سنتعلم كيفية عمل زر لإغلاق الـ Raspberry Pi بأمان بدون الإضرار بإغلاقها من خلال أمر الإغلاق ولا فصل مصدر الطاقة عنها.



المكونات المطلوبة



راسپری بای



سلك توصيل female \ female



زر ذو طرفين press button



مكواة لحام



قصدير لحام

توصيل الدائرة

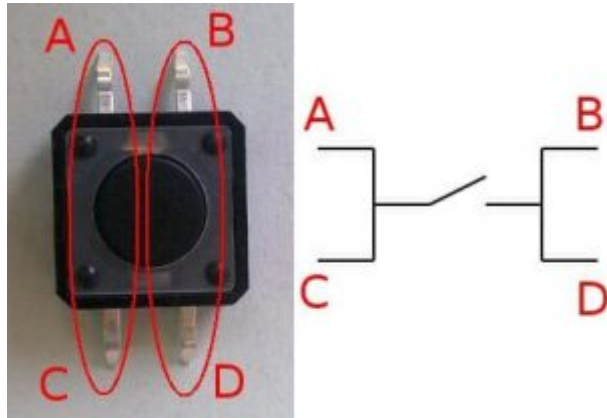
نقوم بقص سلك التوصيل لنصفين متساويين في الطول ثم نقوم بقطع جزء من الغلاف البلاستيكي الحامي للسلك حتى يظهر جزء مناسب من السلك فنستطيع لحامه في زر الضغط.



نقوم بلحام طرفي السلكين في طرفي زر الضغط كالتالي.



إذا كان زر الضغط من ذوى الأربعة أطراف فعليك ملاحظة الصورة التالية وأن تقوم بلحام طرفى الأسلاك فى النقطتين A,B أو C,D.



كل ما عليك فعله الآن هو إختيار أحد أرجل التوصيل GPIO المناسبة والتي ستختلف طبقاً لنوع الراسبرى باى المستخدمة، فى حالتى هذه أستخدم راسبرى باى موديل +B فقمتم بإختيار أقرب رجل توصيل لأحد رجول الأرضي GND وكانت هى الرجل GPIO7 لنا للا وبإمكانك إختيار ما تريد.

3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
GPIO 10 MOSI	19	20	GROUND
GPIO 9 MISO	21	22	GPIO 25
GPIO 11 SCLK	23	24	GPIO 8
GROUND	25	26	GPIO 7
DNC	27	28	DNC
GPIO 5	29	30	GROUND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GROUND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GROUND	39	40	GPIO 21

نقوم بتوصيل الزر السابق الإعداد في الطرفين رقم 26 و رقم 25 من أرجل التوصيل GPIO كالتالي

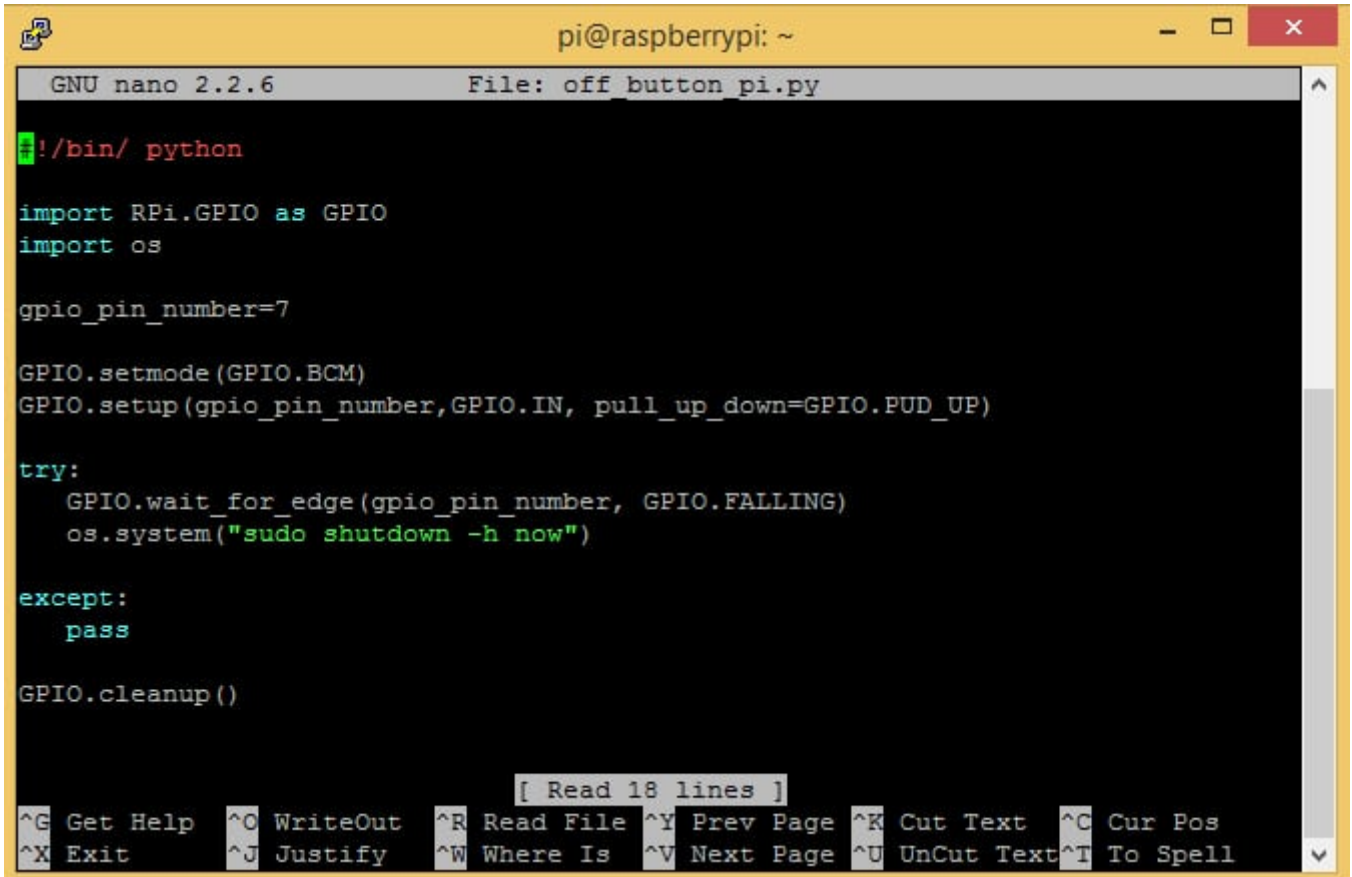


الكود البرمجي

في البداية نقوم بفتح ال Terminal الخاص ب الـ Raspberry باي وكتابة الأوامر التالية أو نقوم بفتح نافذة الـ SSH الخاصة بها من جهاز آخر مربوط معها على نفس الشبكة كما تم شرحه في الدرس الخامس.
نقوم بكتابة الأمر التالي الذي يقوم بفتح محرر النصوص الشهير Nano ويقوم بإنشاء ملف اسمه off_button_pi.py لكتابة الكود بداخله بلغة البايثون python.

```
nano off_button_pi.py
```

نقوم بكتابة الكود التالي بداخل الملف السابق



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: off_button_pi.py

#!/bin/ python

import RPi.GPIO as GPIO
import os

gpio_pin_number=7

GPIO.setmode(GPIO.BCM)
GPIO.setup(gpio_pin_number,GPIO.IN, pull_up_down=GPIO.PUD_UP)

try:
    GPIO.wait_for_edge(gpio_pin_number, GPIO.FALLING)
    os.system("sudo shutdown -h now")

except:
    pass

GPIO.cleanup()
```

ثم نقوم بالضغط على Ctrl+O ثم enter ثم ctrl+X وذلك لحفظ الملف السابق.

وكما نلاحظ الكود بسيط جداً يعتمد على إستدعاء مكتبتين رئيسيتين وهما GPIO وهي للتعامل مع أرجل التوصيل وجعل الرجل الموصلة على زر الضغط تعمل كطرف إدخال، وكذلك لتشغيل الـ Pull Up Resistor الداخلية في الـ راسبيري باى فيمكننا بذلك جعل رجل التوصيل موصلة دائماً على فولت High وعند الضغط فقط على الزر تصبح موصلة بـ Low ومكتبة OS وهي الأهم هنا حيث تمكنا من التعامل مع نظام التشغيل وإرسال أوامر له عن طريق كود البايثون فنستخدمها داخل أمر try الذى ينتظر أن يتم الضغط على الزر عن طريق أمر GPIO.wait_for_edge ليقوم بعد ذلك بإرسال الأمر sudo shutdown -h now ليغلق الـ راسبيري باى بطريقة سليمة.

إذا كنت تريد تغيير رجل التوصيل GPIO كل ما عليك هو تغيير رقمها فى الملف السابق gpio_pin_number لتصبح تساوى رقم الـ GPIO الأخرى التى تريد إستخدامها.

يبقى لنا خطوة واحدة وهي أن نجعل هذا البرنامج يعمل تلقائياً عند فتح الـ راسبيري باى وذلك بفتح الملف rc.local بمحرر النصوص Nano مع كتابة sudo قبل الأمر وذلك ليعطى للمستخدم صلاحيات الـ super user فيمكنك تعديل هذا الملف.

```
sudo nano /etc/rc.local
```

نقوم بإضافة مسار الملف الذى قمنا بإنشائه فى آخر rc.local قبل exit 0 مع كتابة اسم اللغة المستخدمة فى كتابة البرنامج قبل المسار وهي python كالتالى.



```
pi@raspberrypi: ~  
GNU nano 2.2.6 File: /etc/rc.local Modified  
  
# By default this script does nothing.  
  
# Print the IP address  
_IP=$(hostname -I) || true  
if [ "$_IP" ]; then  
    printf "My IP address is %s\n" "$_IP"  
fi  
python /home/pi/off_button_pi.py  
  
exit 0  
  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

ثم نقوم بالضغط على Ctrl+O ثم enter ثم Ctrl+X وذلك لحفظ الملف السابق.

كل ما تبقى عليك هو عمل reboot للراسبيري باى هكذا.

```
sudo reboot
```

تستطيع الآن الضغط على زر الإغلاق وهو سيقوم بطريقة سليمة وصحيحة بإغلاق الراسبيري باى.