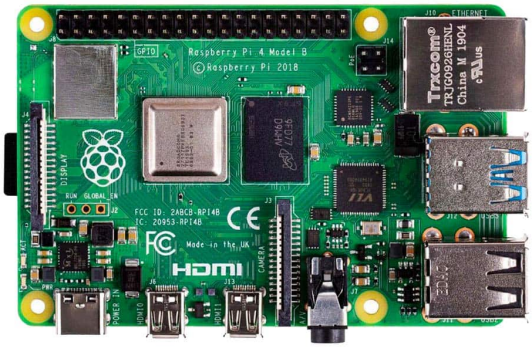
انشاء واجهة مستخدم في الراسبيري باي لتحكم في المصباح

مقدمة

في هذا الدرس سنتعرف على طريقة انشاء واجهة مستخدم رسومية في الراسبيري باي تساعدنا على التحكم في المصباح.

https://geeksvalley.com/wp-content/uploads/2022/08/FullSizeRender.mp4

المواد والأدوات

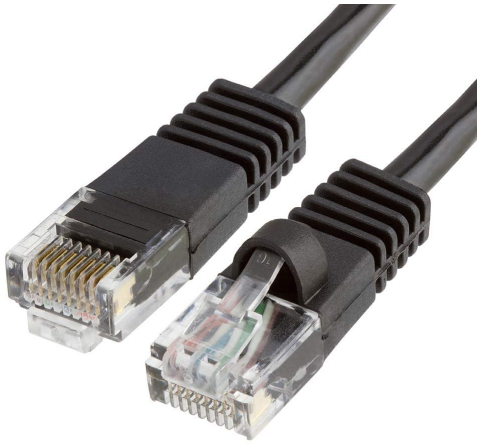


راسبيري باي (Raspberry Pi 4)

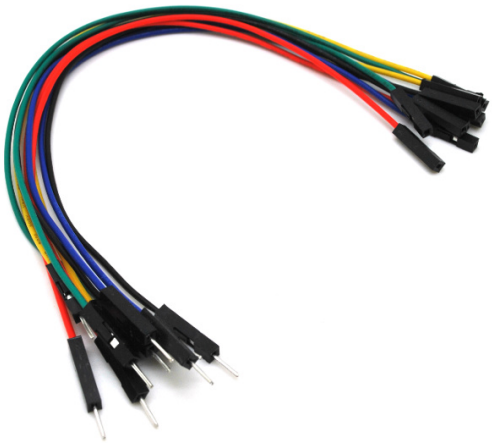
1× سلك (HDMI)

1× محول تيار

1× كرت ذاكرة



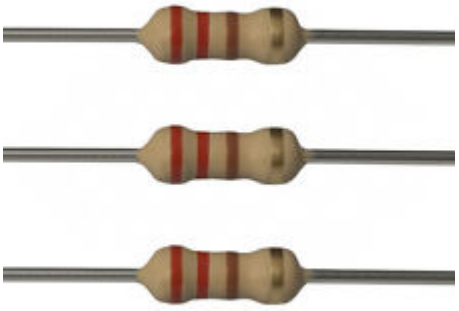
1× سلك ايثرنت



حزمة أسلاك توصيل (ذكر – أنثى)

1× لوحة تجارب حجم صغير

1× ثنائي مشع للضوء أحمر (LED)



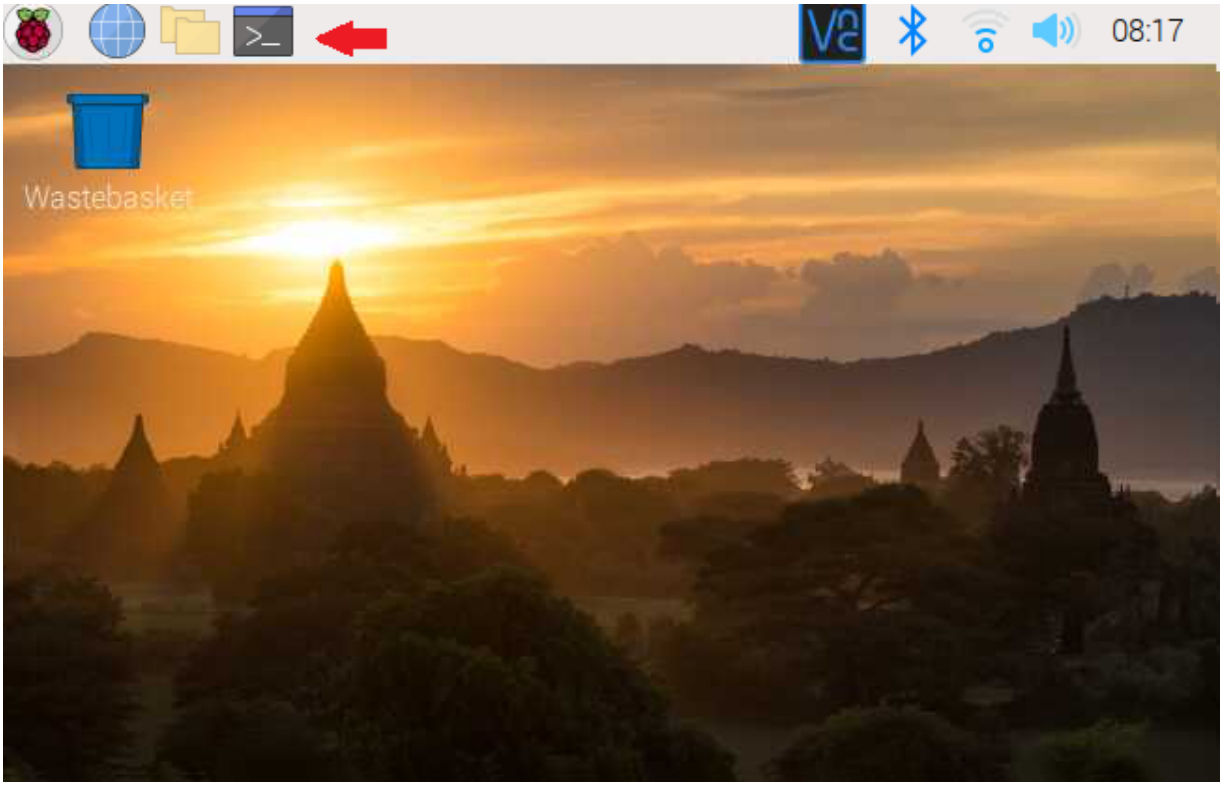
1× مقاومة 220Ω

تهيئة الراسبيري باي

أولا، سوف تحتاج إلى تثبيت نظام الراسبيان على الراسبيري باي إذا لم تكن قد فعلت ذلك قم بالإطلاع على الدرس الأول نظام تشغيل الراسبيري باي

وبعد تثبيت النظام، يمكنك تهيئة النظام من خلال الرجوع للدرس الثاني تهيئة نظام التشغيل

افتح الشاشة السوداء LXterminal للبدء باستخدام الراسبيري باي.



للبدء، قم بإستخدام الـ Termial لتحديث الراسبيري باي إلى أحدث إصدار.

```
sudo apt-get update
sudo apt-get upgrade
```

ثم أعد تشغيل الراسبيري باي.

```
reboot
```

حمّل البايثون على الراسبيري باي.

```
sudo apt install python3 idle3
```

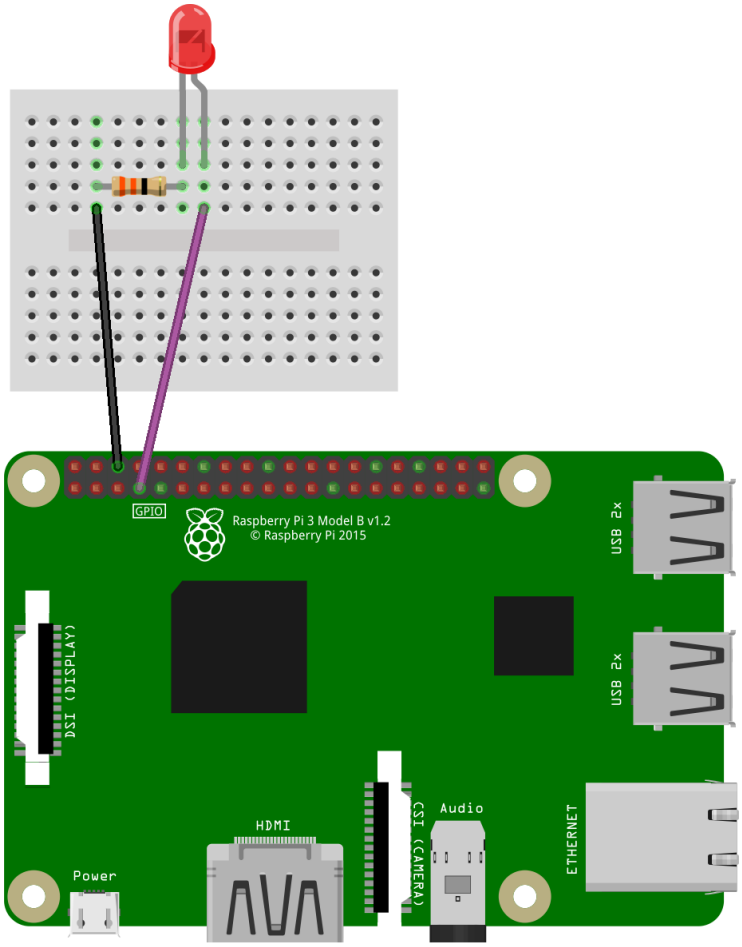
تحمل حزمة GUIZERO

سنقوم بتحميل حزمة guizero التي تساعدنا على انشاء واجهة مستخدم رسومية.

```
sudo  pip3 install guizero
```

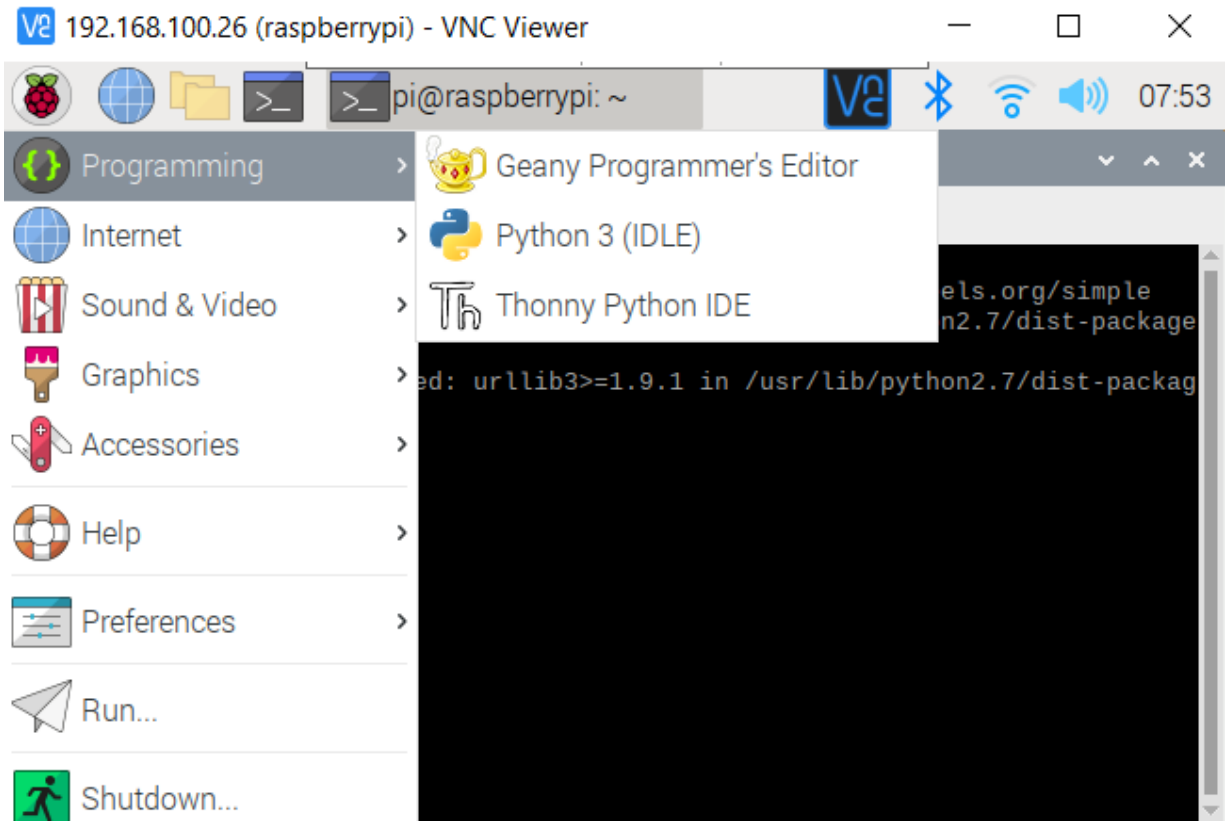
توصيل الدائرة

وصل الدائرة الكهربائية كما هو ظاهر بالشكل:



البرمجة

ستشاهد في قائمة Programming وجود برنامج جديد وهو Python 3 (IDLE) انقر عليه.



من قائمة File اختر New File.

انسخ الكود الخاص بمشروع انشاء واجهة مستخدم في الراسبيري باي لتحكم في المصباح والصقه بالملف الفارغ.

```
from gpiozero import LED
from time import sleep
from guizero import App, Text, PushButton
import sys

myled = LED(4)

def switch_on():
  print("ON")
  myled.on()

def switch_off():
  print("OFF")
  myled.off()

def close_gui():
  sys.exit()
  myled.off()

def blink_LED():
  count = 0
  while count < 5:
    myled.on()
    sleep(1)
    myled.off()
    sleep(1)
    count+=1

app = App(title="LED Control")

button1 = PushButton(app, command=switch_on, text="LED ON", width=10,height=3)
button2 = PushButton(app, command=switch_off, text="LED OFF", width=10,height=3)
button3 = PushButton(app, command=blink_LED, text = "Blink LED", width=10,height=3)
button4 = PushButton(app, command=close_gui, text="Close", width=10, height=3)

app.display()
```

شرح الكود البرمجي

في هذه الأسطر تم استدعاء المكتبات الضرورية مثل مكتبة gpiozero التي تساعدنا على انشاء الوجهات الرسومية ومكتبة Time التي تشتمل على الوحدات المتعلقة بمعالجة الوقت ومكتبة sys التي تختَص وتهتَم في الوظائف الخاصة بالنظام.

```
from gpiozero import LED
from time import sleep
from guizero import App, Text, PushButton
import sys
```

هنا يتم تعريف المنفذ المستخدم لربط المصباح LED.

```
myled = LED(4)
```

إذا كان الزر الأول مضغوط فسيعمل المصباح.

إذا كان الزر الثاني مضغوط سيتوقف المصباح عن العمل.

إذا كان الزر الثالث مضغوط فإن المصباح سيومض تارة يعمل وتارة يتوقف عن العمل بشكل مستمر.

إذا كان الزر الرابع مضغوط فسيتم إغلاق الصفحة بالكامل.

```
def switch_on():
print("ON")
myled.on()
def switch_off():
print("OFF")
myled.off()
def close_gui():
sys.exit()
myled.off()
def blink_LED():
count = 0
 while count < 5:
myled.on()
sleep(1)
myled.off()
sleep(1)
count+=1
```

هنا يتم تهيئة الواجهة الرسومية.

سيظهر العنوان في الأعلى "LED control".

ثم سيظهر الزر الأول والثاني والثالث والرابع على شكل مربعات بأبعاد (الطول=3 والعرض=10).

الزر الأول عن ضغطه سيعمل المصباح "LED ON".

الزر الثاني عن ضغطه سيتوقف المصباح عن العمل "LED OFF".

الزر الثالث عن ضغطه سيبدأ المصباح بالوميض.

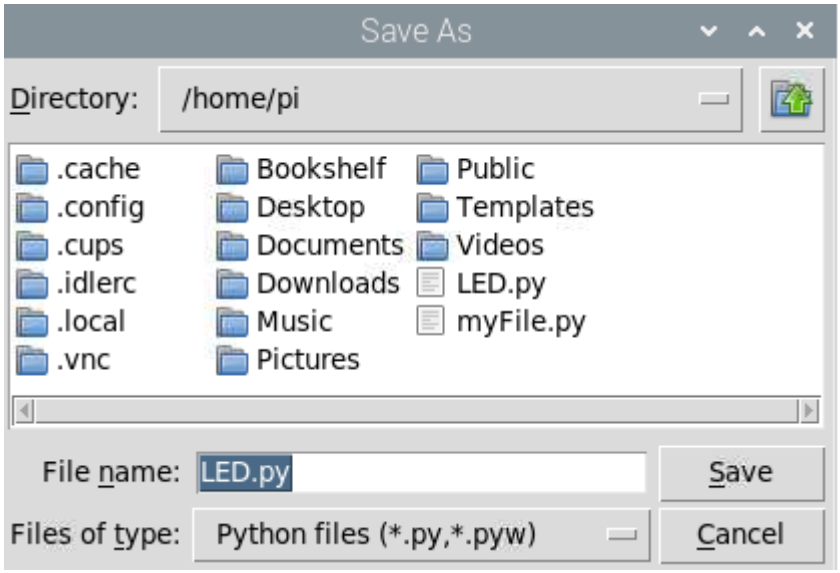
الزر الرابع عن ضغطه سيتم إغلاق الصفحة بالكامل.

```
app = App(title="LED Control")

button1 = PushButton(app, command=switch_on, text="LED ON", width=10,height=3)
button2 = PushButton(app, command=switch_off, text="LED OFF", width=10,height=3)
button3 = PushButton(app, command=blink_LED, text = "Blink LED", width=10,height=3)
button4 = PushButton(app, command=close_gui, text="Close", width=10, height=3)
```

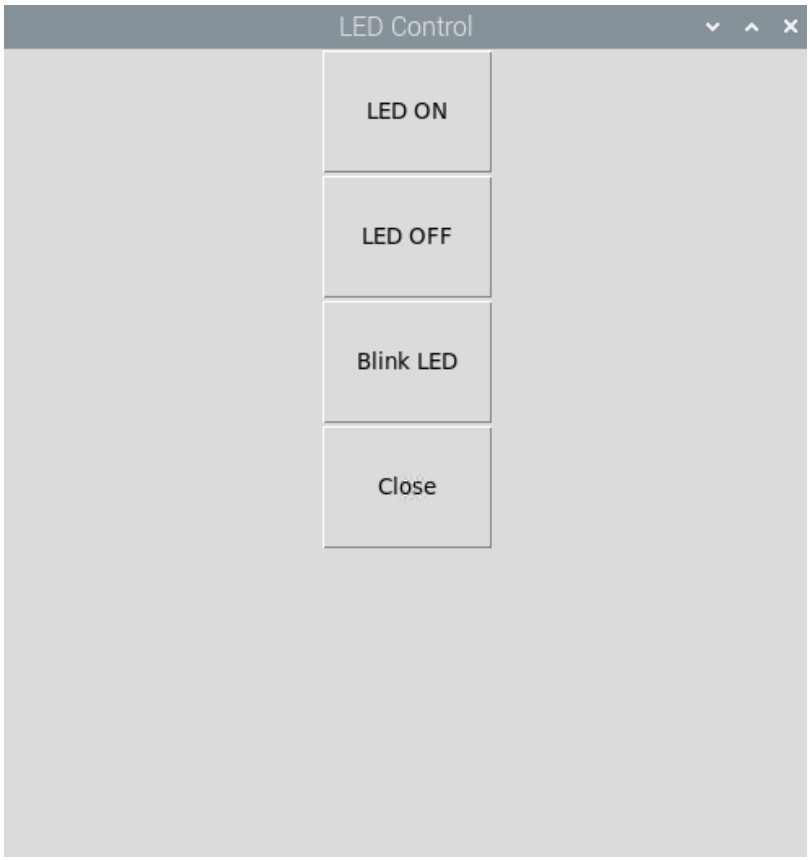
```
app.display()
```

احفظ الملف من قائمة File اختر Save اكتب اسم الملف ثم انقر على Save.



بعد ذلك افتح الشاشة السوداء LXterminal.

واكتب الأمر التالي لتنفيذ الكود البرمجي لمشروع انشاء واجهة مستخدم في الراسبيري باي لتحكم في المصباح.

```
python LED.py
```

بعد عملية الربفع ستظهر واجهة مستخدم بهذا الشكل: