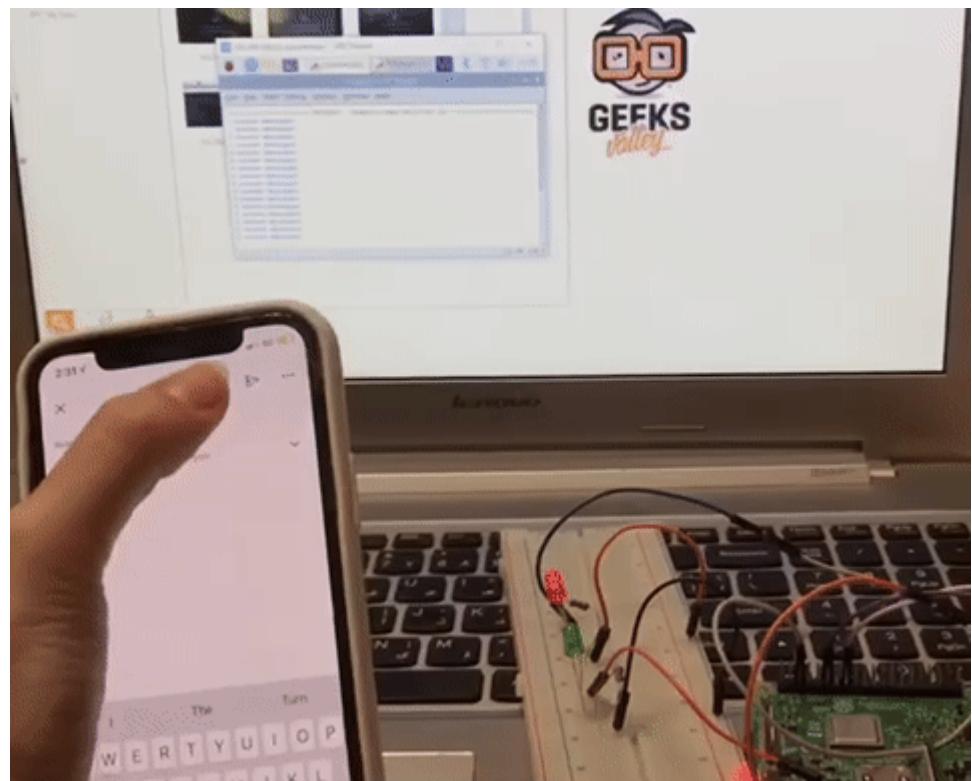


اصنع منبه لرسائل البريد الإلكتروني باستخدام الراسبيري باي

مقدمة

يعتبر المنبه من الوسائل التي يستخدمها الأفراد لتنذيرهم بمواعيدهم المهمة في السابق اعتاد الناس على وضع الساعة الرملية وفي وقتنا الحاضر أصبحت الأجهزة الذكية محلّ لها، في هذا الدرس ستتعلم كيفية صناعة منبه لرسائل البريد الإلكتروني باستخدام الراسبيري باي وال الثنائي المشع للضوء.



المواد والأدوات



راسبيري باي ×1



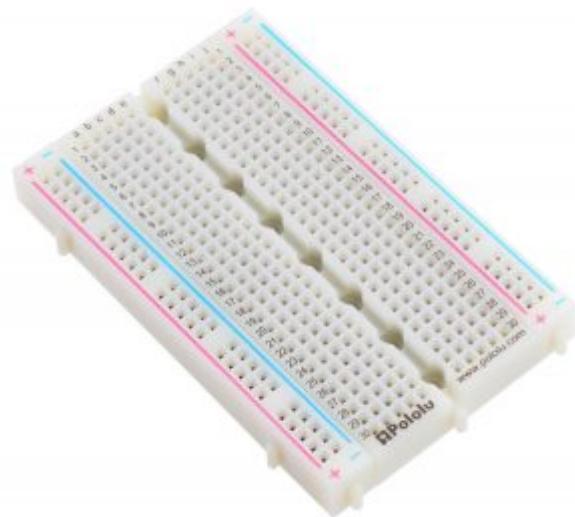
(HDMI) سلك ×1



(5V-2A) محول تيار ×1



١ × ذاكرة



١ × لوحة تجارب - حجم كبير



حزمة أسلاك توصيل (ذكر - أنثى)



١ × ثنايٍ مشع للضوء أحمر (LED)



١ × ثنايٍ مشع للضوء أخضر (LED)

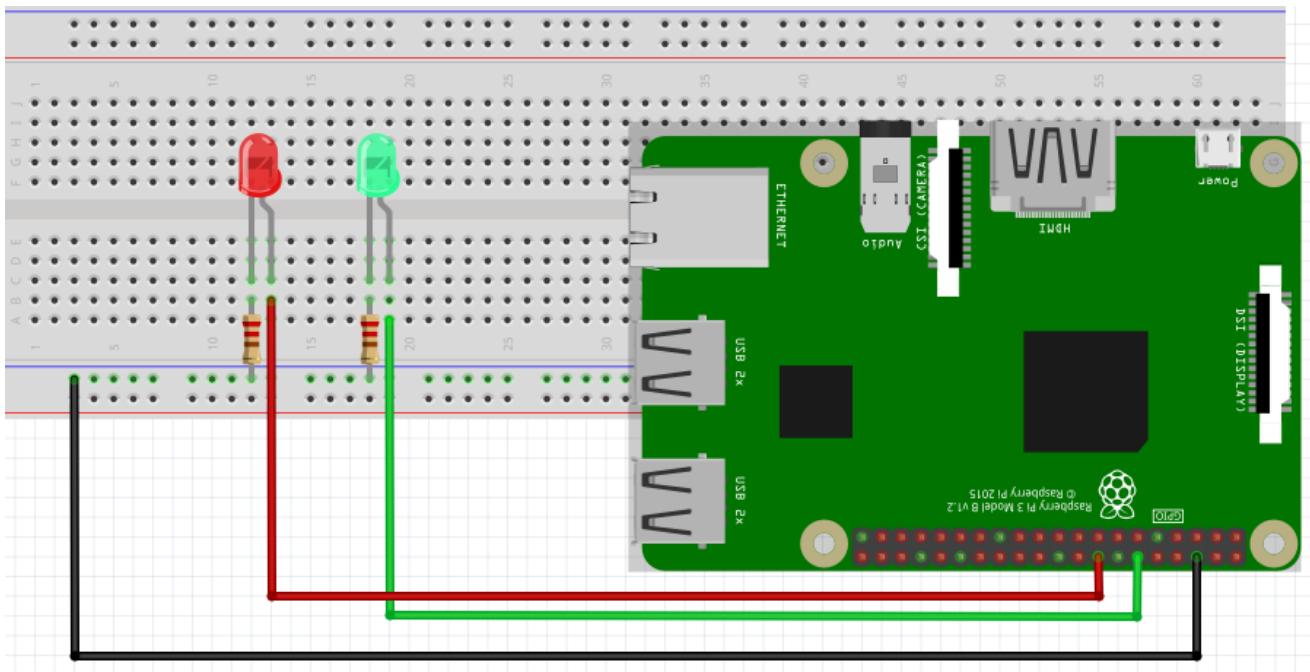


٢ × مقاومة 220Ω

توصيل الدائرة

وصل الدائرة الكهربائية كما في الشكل حتى تتمكن من صنع منه لرسائل البريد الإلكتروني.

للمزيد حول توصيل الراسبيري باي مع الثنائي المشع للضوء يمكنك الرجوع للدرس نظام التحكم ومراقبة الإضاءه عن بعد

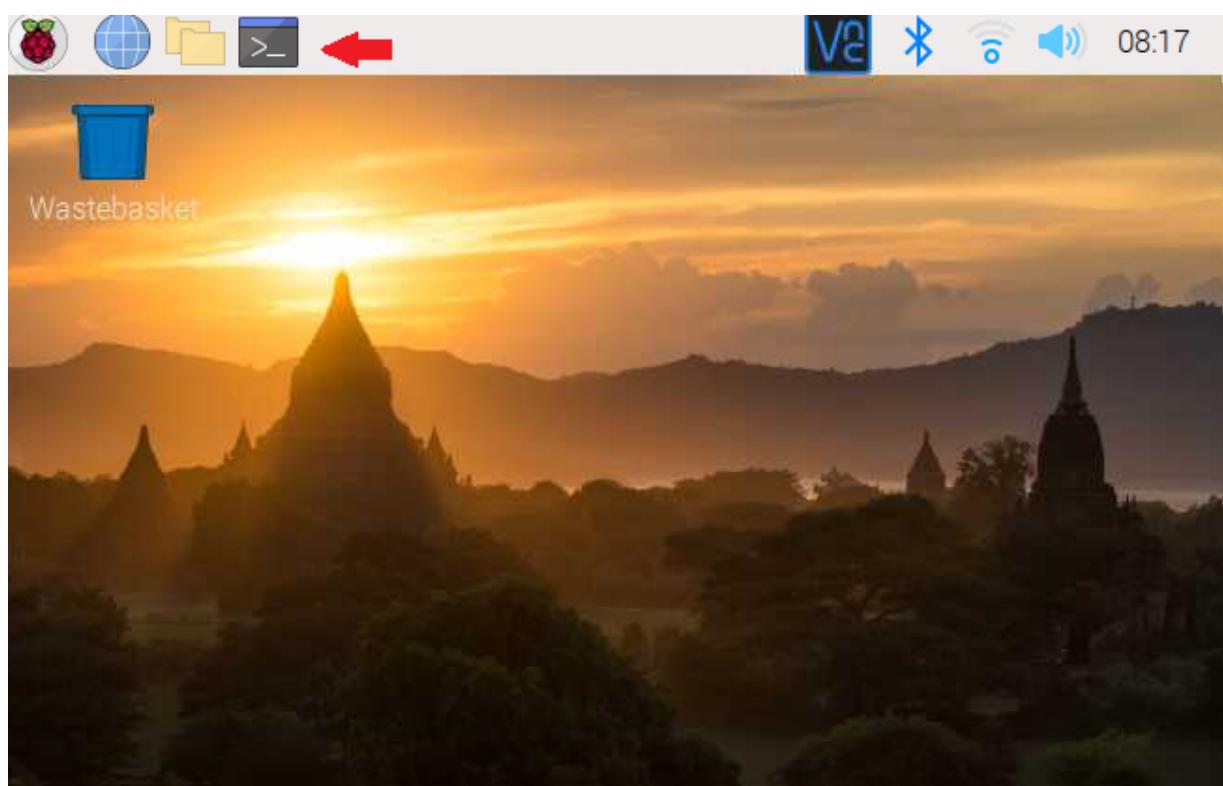


تهيئة الراسبيري باي

أولاً، سوف تحتاج إلى تثبيت نظام الراسبيان على الراسبيري باي إذا لم تكون 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 Terminal لتحديث الراسبيري باي إلى أحدث إصدار.

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

ثم أعد تشغيل الراسبيري باي.

```
reboot
```

حمل البايثون على الراسبيري باي.

```
sudo apt install python3 idle3
```

بعد ذلك حمل مكتبة .adafruit-blinka

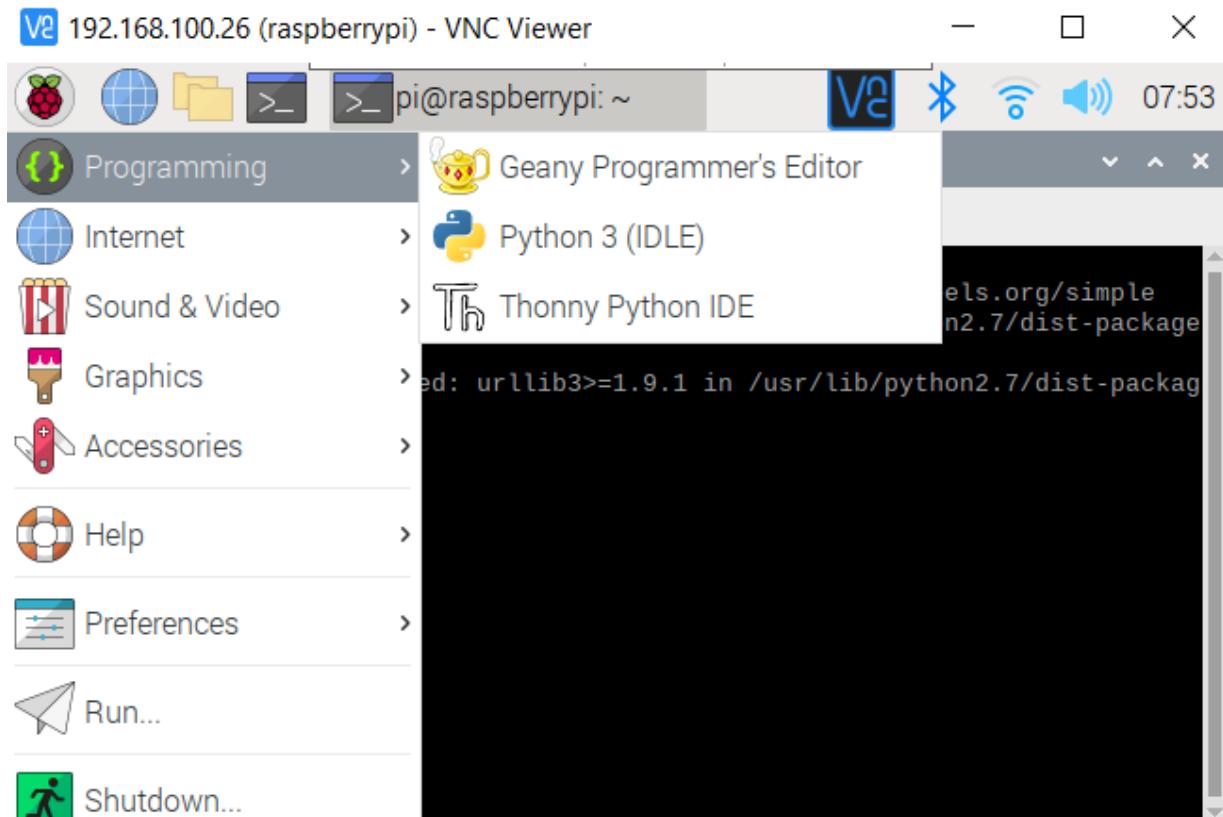
```
sudo pip3 install adafruit-blinka
```

حمل آخر مكتبة .imapclient

```
sudo pip3 install imapclient
```

ال코드 البرمجي

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



من قائمة File اختر New File

انسخ الكود لنظام منبه لرسائل البريد الإلكتروني والصقه بالملف الفارغ لكن هناك أمور مهمة عليك العمل بها قبل رفع الكود البرنامجي

```
import time
import board
from imapclient import IMAPClient
from digitalio import DigitalInOut, Direction
HOSTNAME = 'imap.gmail.com'
MAILBOX = 'Inbox'
MAIL_CHECK_FREQ = 10 # check mail every 10 seconds
# The following three variables must be customized for this
# script to work
USERNAME = 'Your Email'
PASSWORD = 'Your Password'
NEWMAIL_OFFSET = 0
# setup Pi pins as output for LEDs
green_led = DigitalInOut(board.D18)
red_led = DigitalInOut(board.D23)
green_led.direction = Direction.OUTPUT
red_led.direction = Direction.OUTPUT

def mail_check():
# login to mailserver
    server = IMAPClient(HOSTNAME, use_uid=True, ssl=True)
    server.login(USERNAME, PASSWORD)

# select our MAILBOX and looked for unread messages
    unseen = server.folder_status(MAILBOX, ['UNSEEN'])

# number of unread messages
# print to console to determine NEWMAIL_OFFSET
    newmail_count = (unseen[b'UNSEEN'])
    print('%d unseen messages' % newmail_count)

if newmail_count > NEWMAIL_OFFSET:
    green_led.value = True
    red_led.value = False
else:
    green_led.value = False
    red_led.value = True

time.sleep(MAIL_CHECK_FREQ)

while True:
    mail_check()
```

شرح الكود البرمجي

هنا نستعدي المكتبات الضرورية لنظام منبه لرسائل البريد الإلكتروني مثل `time` و `board` و `imapclient` مثل `DigitalInOut`, `Direction`.

```
import time
import board
from imapclient import IMAPClient
from digitalio import DigitalInOut, Direction
```

في HOSTNAME يتم تعريف اسم المستضيف Gmail . وفي MAILBOX يتم توضيح أي المجلدات التي سيتم فحصها هنا سنستخدم المجلد Inbox .

```
HOSTNAME = 'imap.gmail.com'
MAILBOX = 'Inbox'
```

المتغير NEWMAIL_OFFSET يحمل القيمة 0 وستستخدم هذه القيمة فيما بعد لمقارنة عدد رسائل البريد الإلكتروني .

```
NEWMAIL_OFFSET = 0
```

في المتغير MAIL_CHECK_FREQ يتم تعين المدة الزمنية التي سيتم بها فحص مجلد Inbox هنا سيتم فحصه كل ثانية للتأكد من وجود رسائل جديدة أم لا .

```
MAIL_CHECK_FREQ = 1 # check mail every 1 second
```

في المتغير USERNAME ضع الايميل الخاص بك .

وفي المتغير PASSWORD ضع كلمة السر .

```
USERNAME = 'Your Email'
PASSWORD = 'Your Password'
```

في المتغير green_led يتم تعريف المدخل المستخدم لربط الثنائي المشع للضوء الأخضر وهو GPIO 18 .

وفي المتغير red_led يتم تعريف المدخل المستخدم لربط الثنائي المشع للضوء الأحمر وهو GPIO 23 .

```
green_led = DigitalInOut(board.D18)
red_led = DigitalInOut(board.D23)
```

في الدالة def mail_check سيتم أخذ الصلاحيه لتسجيل الدخول للبريد الإلكتروني بناء على البيانات الموجودة في المتغيرين USERNAME و PASSWORD .

```
def mail_check():
# login to mailserver
server = IMAPClient(HOSTNAME, use_uid=True, ssl=True)
server.login(USERNAME, PASSWORD)
```

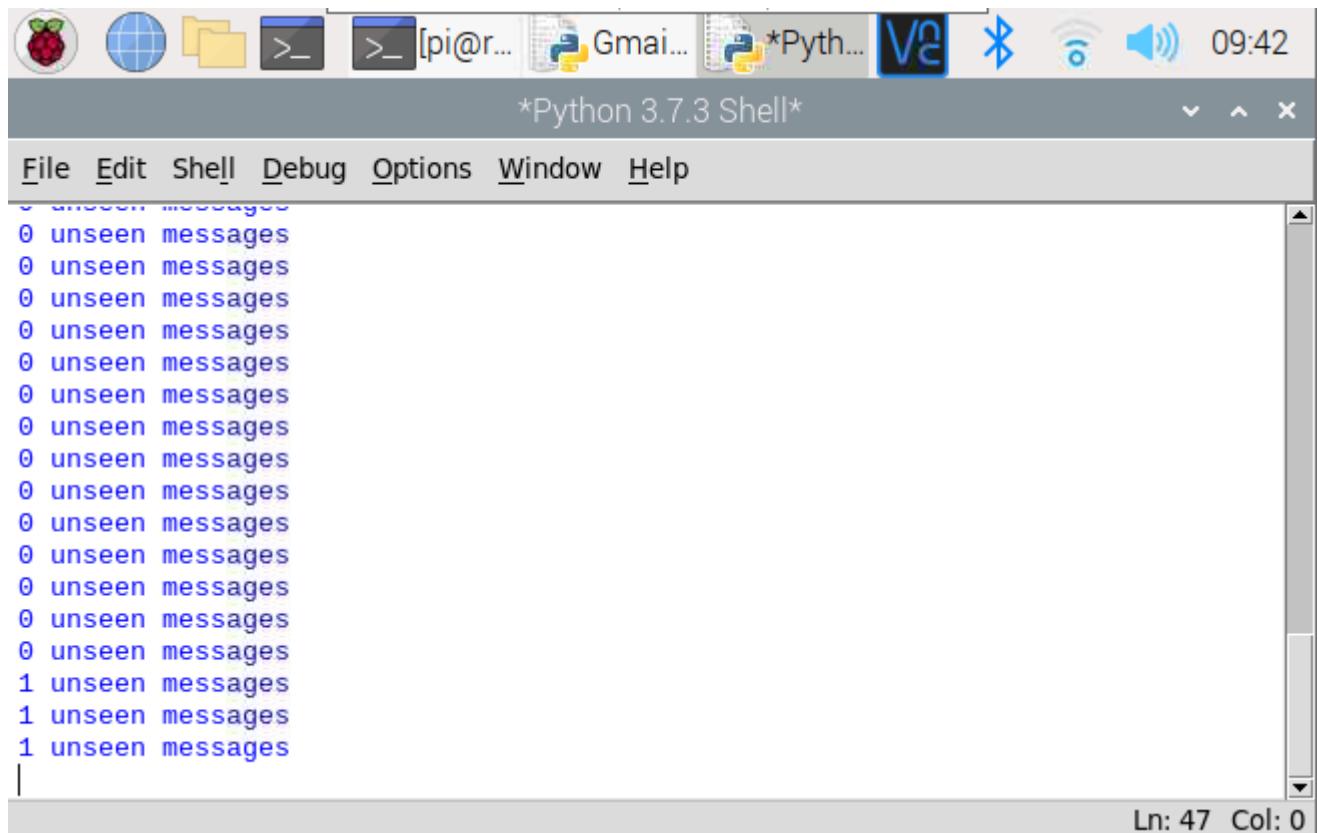
في المتغير unseen يتم فحص مجلد Inbox .

```
unseen = server.folder_status(MAILBOX, [ 'UNSEEN' ])
```

في المتغير newmail_count يتم حساب عدد الرسائل التي لم تقرأ بعد .

سينطبع على الشاشة unseen messages و بجانبها عدد الرسائل التي لم تقرأ وسيتم تحديث المتغير كل ثانية.

```
newmail_count = (unseen[b'UNSEEN'])
print('%d unseen messages' % newmail_count)
```



```
*Python 3.7.3 Shell*
```

```
File Edit Shell Debug Options Window Help
0 unseen messages
1 unseen messages
1 unseen messages
1 unseen messages
```

Ln: 47 Col: 0

إذا كان عدد الرسائل المرسلة المخزنة في NEWMAIL_OFFSET أكبر من القيمة المعينة في المتغير NEWMAIL_OFFSET سيضيء الثنائي المشع للضوء الأخضر ليفيد الأفراد بوجود رسائل جديدة.

وإذا كان أقل سينضيء الثنائي المشع للضوء الأحمر وهكذا يستمر النظام.

```
if newmail_count > NEWMAIL_OFFSET:
    green_led.value = True
    red_led.value = False
else:
    green_led.value = False
    red_led.value = True

time.sleep(MAIL_CHECK_FREQ)
```

يمكنك رفع الكود البرمجي واختبار صحة الخطوات.

لا تنس فصل وحدة مزود الطاقة بعد الانتهاء من استخدام نظام منبه لرسائل البريد الإلكتروني.