التحكم باللوحة الإعلانية عن بعد

لوحة الإعلانات اللاسلكية هو إنتقاء جيد لهذا المشروع، بدلا من أن تكون مجرد لوحة إعلانية بسيطة. أولا يجب أن نفهم الغرض من هذا المشروع، في هذا النظام يمكننا عرض رسالة أو شعار لبعض أجهزة العرض مثل شاشات الكريستال السائل (LCD)، وهذه الرسالة يمكن بسهولة وضعها و تغييرها من أي مكان في العالم، عن طريق إستخدام الانترنت لإرسال الرسالة لاسلكيا من متصفح ويب إلى شاشة الـ (LCD) المتصلة بلوحة الراسبيري باي.لذالك يمكنك إرسال الرسالة بإستخدام الحاسوب، الهاتف الذكي أو tablet .

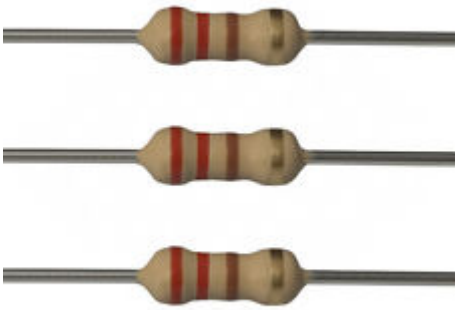
القطع المطلوبة :

في نظام التحكم بشاشات الإعلانات، سنقوم بإنشاء خادم الويب المحلي (local web server)، والذي يمكنه أن يكون (A global server) على شبكة الإنترنت. في الراسبيري باي ، سيتم إستخدام شاشة العرض LCD 16×2 لعرض الرسالة و Flask لإستقبال الرسالة عبر الشبكة. كلما يتلقى الراسبيري باي أي رسالة لاسلكية من صفحة ويب ، فإنه يعرض هذه الرسالة على شاشة الكريستال السائل (LCD).

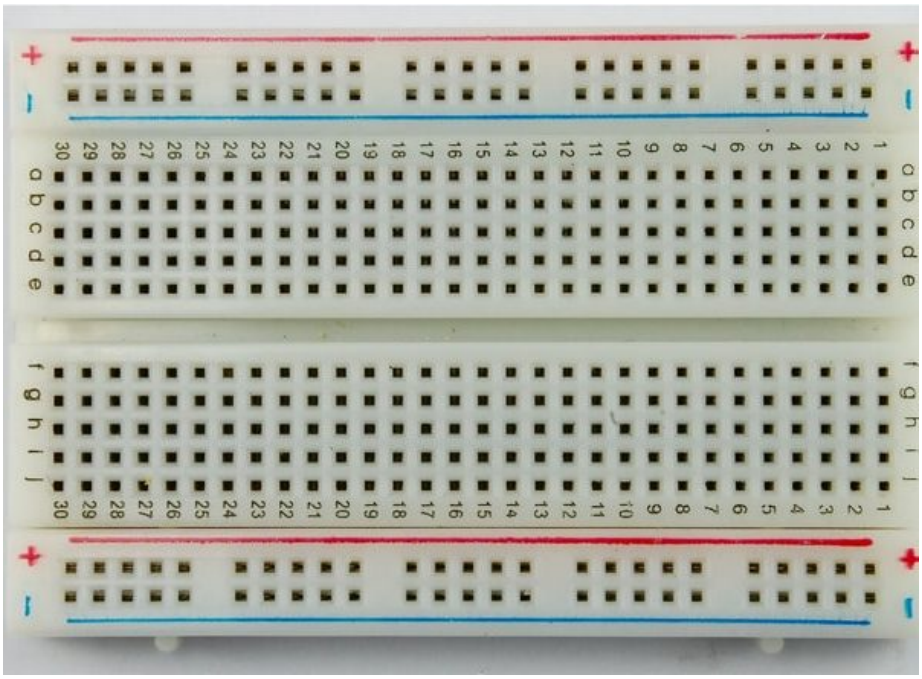
الأدوات التي تحتاجها لهذا المشروع :

شاشة LCD 16×2

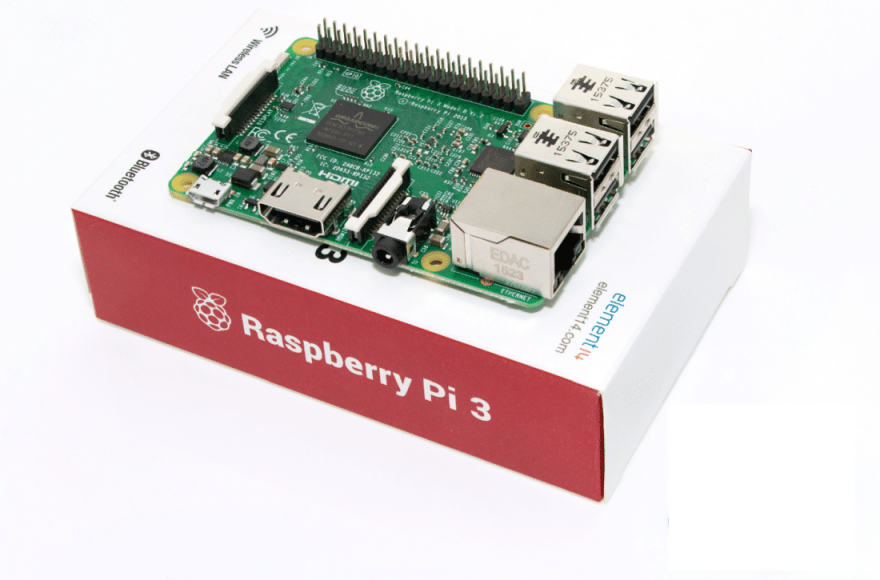
Potentiometer 10K Ω



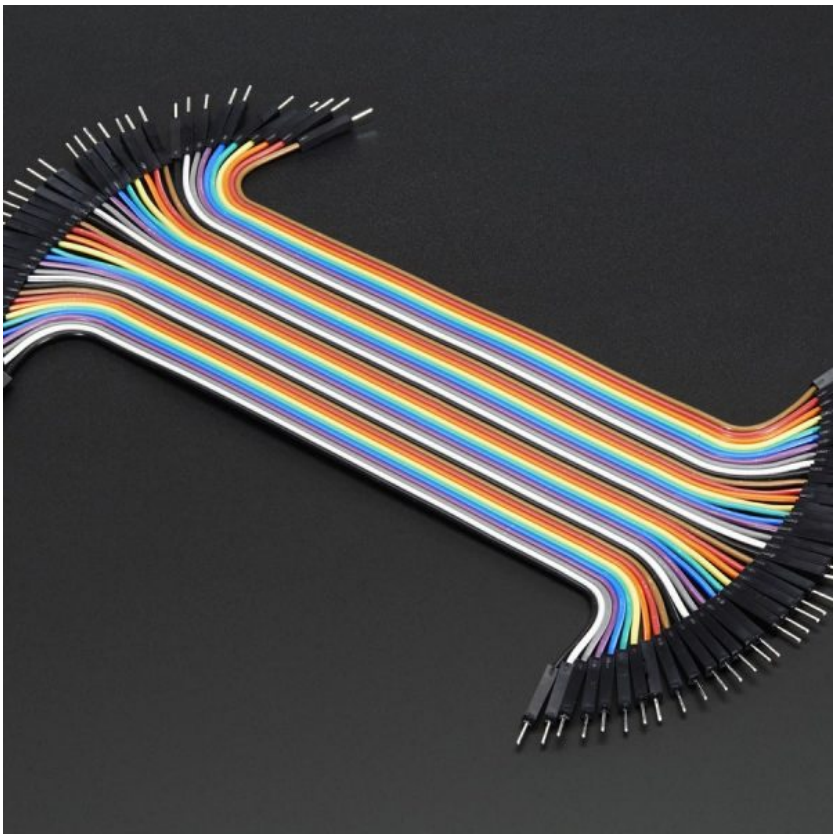
مقاومة 220 اوم



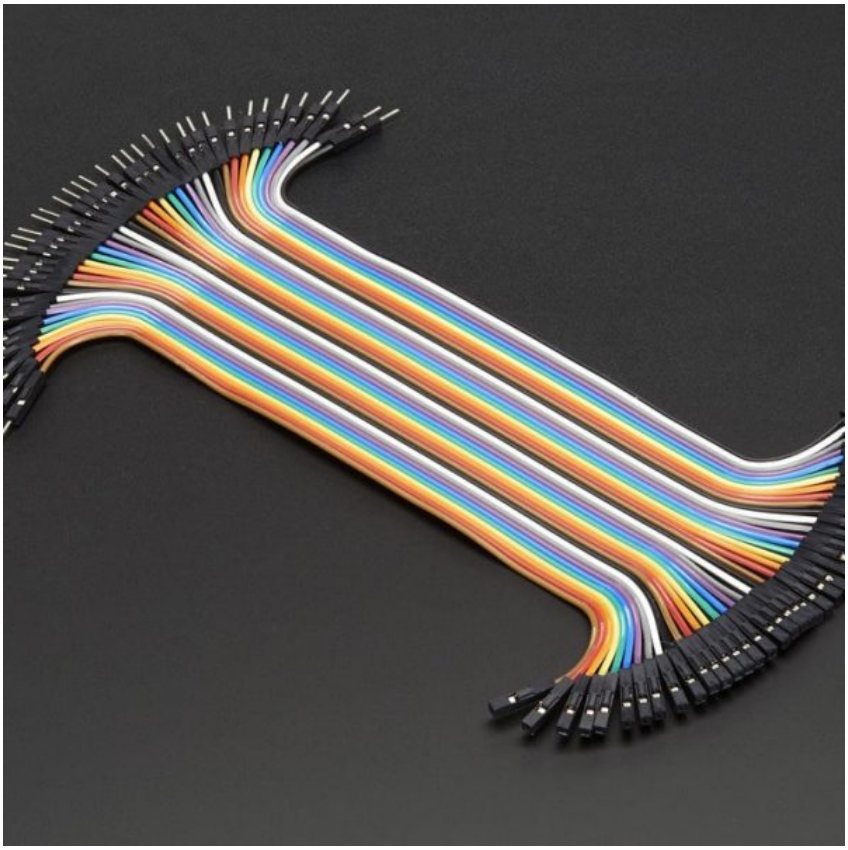
لوح تجارب حجم متوسط (Half size breadboard)



Raspberry Pi 3 Model B



اسلاك توصيل ذكر/ذكر (Jumper Wires Male Male)



اسلاك توصيل أنثى/ذكر (Jumper Wires Female/male)

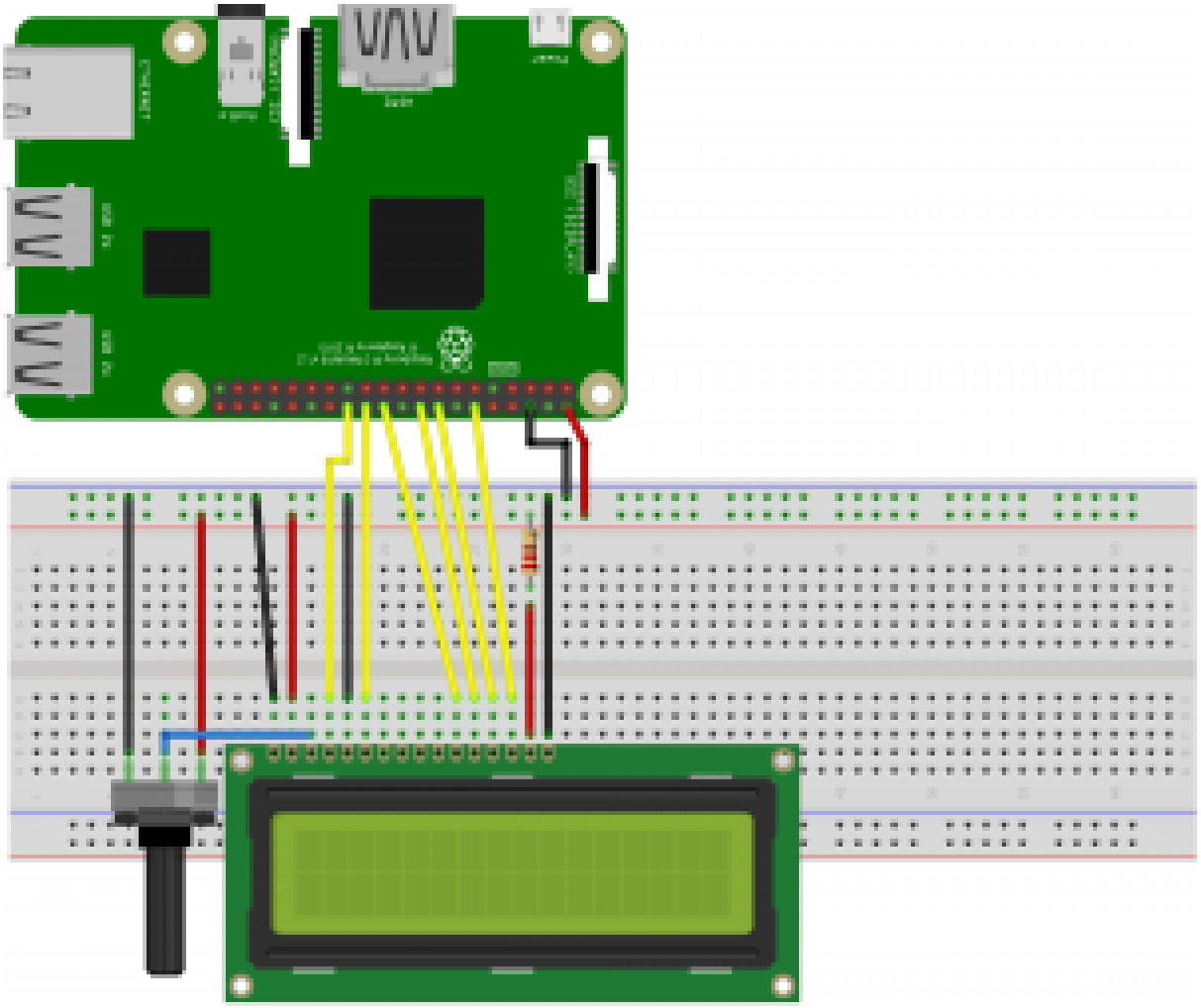
Power Supply

Samsung 8Gb Class 6 Microsd Memory Card

تصميم لوح التجارب :

نحن بحاجة فقط لتوصيل شاشة العرض (LCD) مع لوحة الراسبيري باي عن طريق إستخدام بعض الاسلاك على لوحة التجارب.

قم بتوصيل الدائرة كما هو موضح بالصورة :

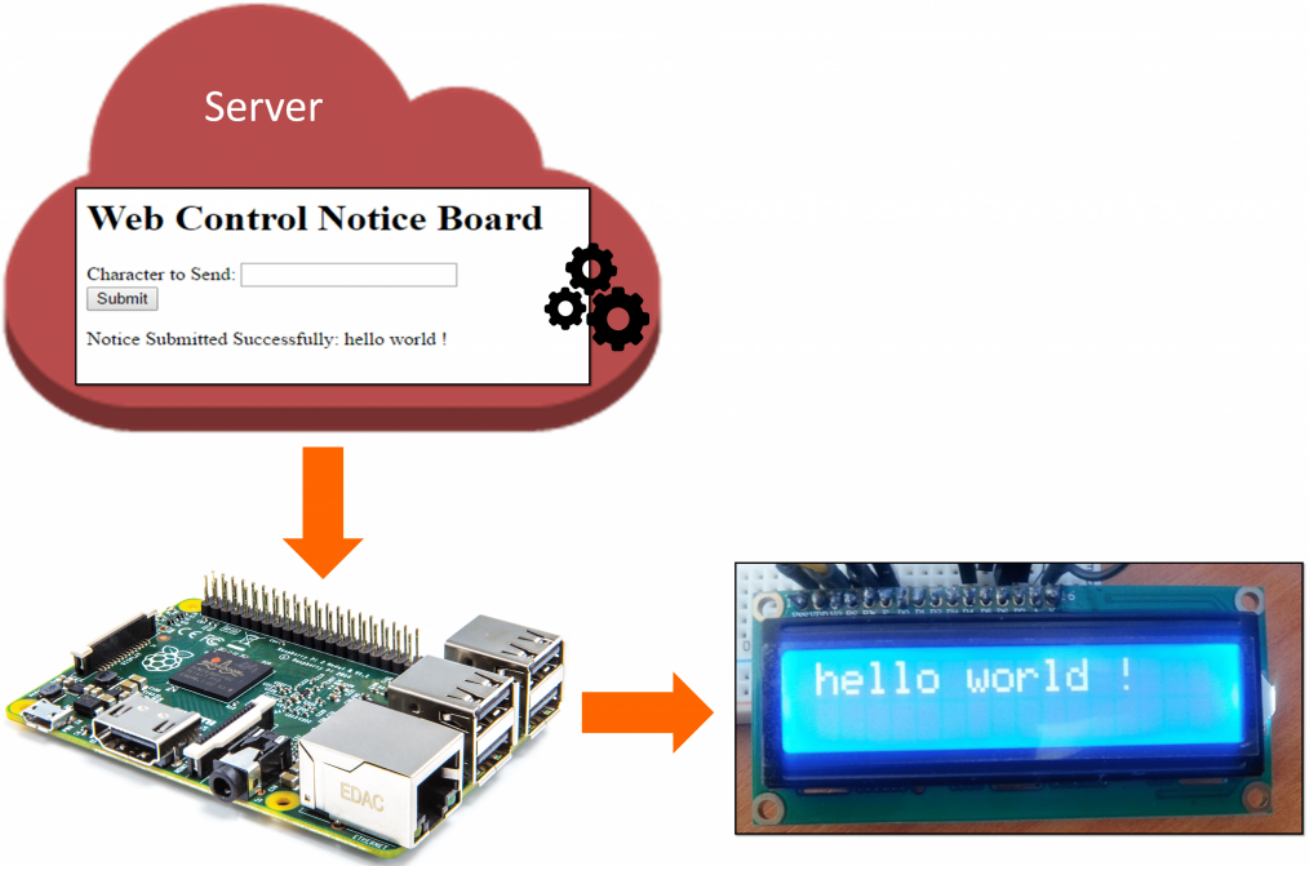


توصيل شاشة الـ LCD :

راسبيري باي	شاشة LCD
GND	Vss
5v	VDD
GND	RW
GPIO7	RS pin
GPIO8	Enable pin
GPIO25	D4 pin
GPIO24	D5 pin
GPIO23	D6 pin
GPIO18	D7 pin

إنشاء صفحة ويب :

في هذا المشروع، العنصر الرئيسي هو الراسبيري باي، و التي تمثل قلب هذا المشروع،وتستخدم للتحكم على العمليات المرتبطة بهذا المشروع. مثل : التحكم بشاشة العرض (LCD)، وتلقي الرسائل من الخادم (Server) .



سيتم إنشاء خادم ويب (Web Server)، الذي يوفر وسيلة لإرسال "الرسالة المراد إعلانها" الى الراسبيري باي بإستخدام Flask في متصفح الويب. Flask هو إطار عمل مُصغّر/صغير (Micro-Framework) وقوي فى نفس الوقت لتطوير برامج الويب عبر لغة بايثون Python، يتوفّر على عدد لا بأس به من الدوال المُساعدة، مُناسب لتطوير تطبيقات صغيرة ومُتوسّطة (مُدونة، منتدى، موقع شخصي...).

قم بإنشاء ملف webapp ليتم حفظ الملفات المتعلق بهذا المشروع داخله عن طريق كتابة الأمر التالي على نافة Terminal :

```
mkdir webapp
```

في هذا المشروع سوف نقوم بإنشاء صفحة ويب مع مربع نص (TextBox) و زر إرسال (Submit button)، حيث يمكننا أن ندخل "رسالة الاعلان" في مربع نص ثم رفعها إلى الخادم (Server) عن طريق النقر على زر إرسال. يتم تطوير هذا التطبيق على شبكة الإنترنت باستخدام لغة HTML. الكود البرمجي لصفحة الويب هذه سيتم شرحه و عرضه في الخطوات التالية.

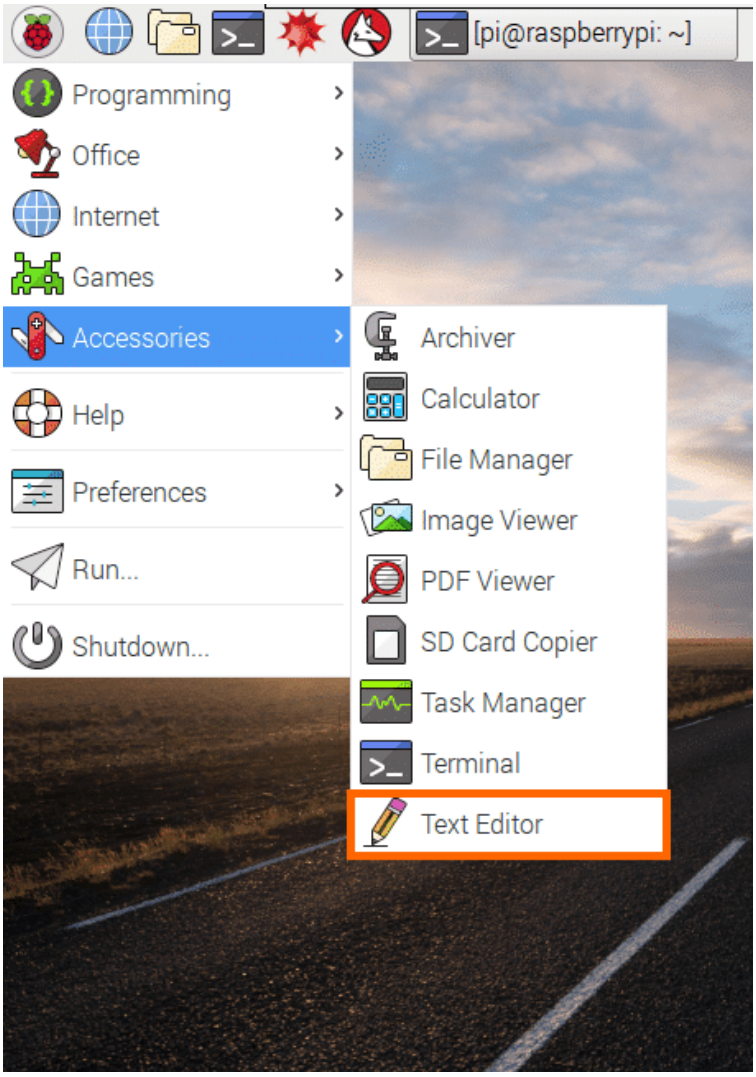
لإنشاء صفحة ويب :

أولا: قم بإنشاء ملف templates في الملف webapp عن طريق إدخال الأمر التالي في Terminal:

```
cd /home/pi/webapp
```

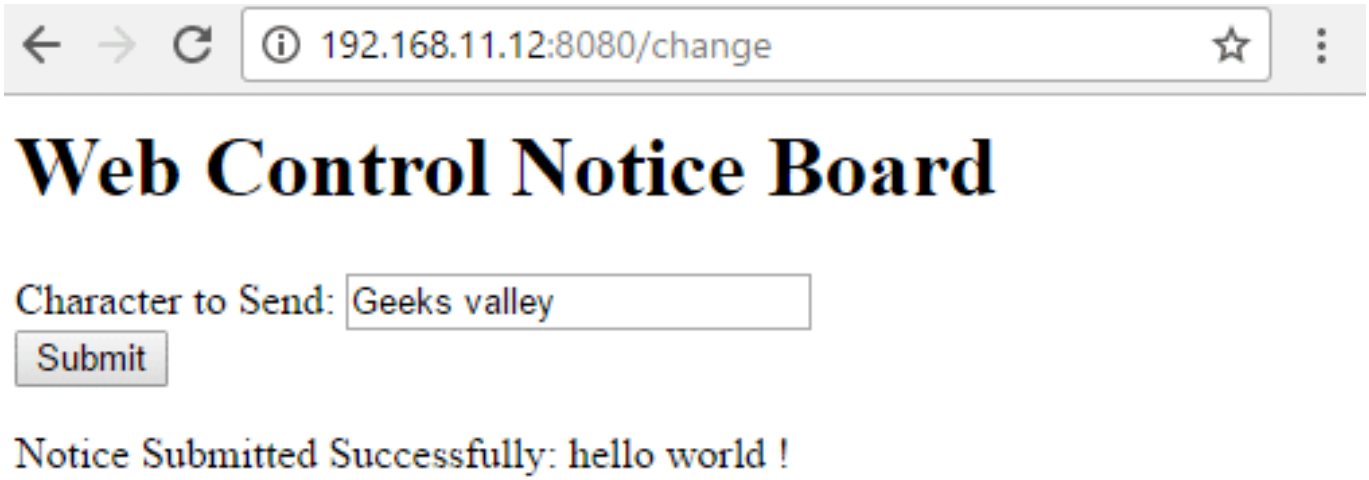
```
mkdir templates
```

تحتاج إلى كتابة كود HTML في محرر نص (text editor) و حفظ الملف مع تمديد (.HTML)
قم بفتح محرر نص (Text Editor) تحت قائمة Accessories في القائمة الرئيسية :



قم بكتابة كود HTML التالي :

```
<h1>Web Control Notice Board</h1>
</div>

<div data-role="content">

<form method="post" action="change">
<label for="slider-1">Notice Message:</label>
<input type="text" name="lcd" id="lcd" />
<br />

<input type="submit" value="Submit" />
</form>

{% if value %}
<p> Notice Submitted Successfully: {{ value }}</p>
{% endif %}

</div>
```



قم بحفظ الملف بتمديد(.HTML) مثل (index.html) في ملف templates ، الذي تم إنشاءه داخل ملف webapp.

لمحة عن الكود :

صفحة الويب يتم إنشاءها بلغة HTML ، والتي تحتوي على مربع نص :

```
<input type="text" name="lcd" id="lcd" />
```

وزر إرسال (Submit button) :

```
<input type="submit" value="Submit" />
```

مع عنوان للصفحه :

```
<h1>Web Control Notice Board</h1>
```

عند النقر على زر إرسال سيتم إستخدام POST method لتنفيذ الكود. ثم نقوم بعرض الرسالة التي تم إرسالها إلى الراسبيري باي عن طريق الخادم (Server).

```
{% if value %}
<p>Notice Submitted Successfully: {{ value }}</p>
{% endif %}
```

فهي تقوم بالتحقق ما اذا كانت هناك نص في مربع النص ثم تقوم بطباعة النص على صفحة الويب نفسها، بحيث يمكن للمستخدم أيضا رؤية الرسالة المرسلة.

أولا قم بتثبيت حزمة Flask. وتأكد من أنك متصل بالانترنت، إما عن طريق Ethernet Cable أو عن طريق WIFI قبل أن تبدأ.

قم بكتابة الأمر التالي على نافذة Terminal لتثبيت الـ Flask :

```
sudo apt-get install python3-flask
```

جزء البرمجة في هذا المشروع يلعب دورا هاما للغاية لأداء جميع العمليات. أولا وقبل كل شي نحن بحاجة إلى درج المكتبات المطلوبة

لـ Flask .ثم تهيئة المتغييرات و تحديد Pins لشاشة الكريستال السائل (LCD) .

```
from flask import Flask
from flask import render_template, request
import RPi.GPIO as GPIO
import time
app = Flask(__name__)

# Define GPIO to LCD mapping
LCD_RS = 7
LCD_E = 8
LCD_D4 = 25
LCD_D5 = 24
LCD_D6 = 23
LCD_D7 = 18
```

لبرمجة شاشة الكريستال السائل (LCD) ، يتم إنشاء الدالة ()lcd_init لتهيئة شاشة LCD.

```
def lcd_init():
# Initialise display
  lcd_byte(0x33,LCD_CMD) # 110011 Initialise
  lcd_byte(0x32,LCD_CMD) # 110010 Initialise
  lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
  lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
  lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
  lcd_byte(0x01,LCD_CMD) # 000001 Clear display
  time.sleep(E_DELAY)
```

و الداله ()lcd_byte لإرسال الأوامر إلى الشاشة، , و الداله ()lcd_string لإرسال البيانات المراد عرضها على الشاشة إلى الـ LCD .

```
def lcd_string(message,line):
# Send string to display
  message = message.ljust(LCD_WIDTH," ")
  lcd_byte(line, LCD_CMD)
  for i in range(LCD_WIDTH):
     lcd_byte(ord(message[i]),LCD_CHR)
```

و فيما يلي هذا الجزء من البرنامج يستخدم لإرسال رسالة من متصفح ويب إلى الراسبيري باي بإستخدام Flask .

```
@app.route("/")
def index():
   return render_template('index.html')
@app.route("/change", methods=['POST'])
def change():
  if request.method == 'POST':
    # Getting the value from the webpage
    data1 = request.form['lcd']
    print ("---Message is", data1)
    lcd_string(data1,LCD_LINE_1)
  return render_template('index.html', value=data1)
if __name__ == "__main__":
```

```
  app.debug = True
  app.run('192.168.1.19', port=8080,debug=True)
```

قم بكتابة الكود البرمجي كاملا كالتالي :

```
#import
from flask import Flask
from flask import render_template, request
import RPi.GPIO as GPIO
import time
app = Flask(__name__)

# Define GPIO to LCD mapping
LCD_RS = 7
LCD_E  = 8
LCD_D4 = 25
LCD_D5 = 24
LCD_D6 = 23
LCD_D7 = 18


# Define some device constants
LCD_WIDTH = 16    # Maximum characters per line
LCD_CHR = True
LCD_CMD = False

LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line

# Timing constants
E_PULSE = 0.0005
E_DELAY = 0.0005
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)       # Use BCM GPIO numbers
GPIO.setup(LCD_E, GPIO.OUT)  # E
GPIO.setup(LCD_RS, GPIO.OUT) # RS
GPIO.setup(LCD_D4, GPIO.OUT) # DB4
GPIO.setup(LCD_D5, GPIO.OUT) # DB5
GPIO.setup(LCD_D6, GPIO.OUT) # DB6
GPIO.setup(LCD_D7, GPIO.OUT) # DB7


def lcd_init():
  # Initialise display
  lcd_byte(0x33,LCD_CMD) # 110011 Initialise
  lcd_byte(0x32,LCD_CMD) # 110010 Initialise
  lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
  lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
  lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
  lcd_byte(0x01,LCD_CMD) # 000001 Clear display
  time.sleep(E_DELAY)

def lcd_byte(bits, mode):
```

```
  # Send byte to data pins
  # bits = data
  # mode = True  for character
  #        False for command

  GPIO.output(LCD_RS, mode) # RS

  # High bits
  GPIO.output(LCD_D4, False)
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x10==0x10:
    GPIO.output(LCD_D4, True)
  if bits&0x20==0x20:
    GPIO.output(LCD_D5, True)
  if bits&0x40==0x40:
    GPIO.output(LCD_D6, True)
  if bits&0x80==0x80:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  lcd_toggle_enable()

  # Low bits
  GPIO.output(LCD_D4, False)
  GPIO.output(LCD_D5, False)
  GPIO.output(LCD_D6, False)
  GPIO.output(LCD_D7, False)
  if bits&0x01==0x01:
    GPIO.output(LCD_D4, True)
  if bits&0x02==0x02:
    GPIO.output(LCD_D5, True)
  if bits&0x04==0x04:
    GPIO.output(LCD_D6, True)
  if bits&0x08==0x08:
    GPIO.output(LCD_D7, True)

  # Toggle 'Enable' pin
  lcd_toggle_enable()

def lcd_toggle_enable():
  # Toggle enable
  time.sleep(E_DELAY)
  GPIO.output(LCD_E, True)
  time.sleep(E_PULSE)
  GPIO.output(LCD_E, False)
  time.sleep(E_DELAY)

def lcd_string(message,line):
  # Send string to display
  message = message.ljust(LCD_WIDTH," ")
```

```
  lcd_byte(line, LCD_CMD)

  for i in range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)

lcd_init()
lcd_string("Rasbperry Pi",LCD_LINE_1)

@app.route("/")
def index():
    return render_template('index.html')
@app.route("/change", methods=['POST'])
def change():
 if request.method == 'POST':
    # Getting the value from the webpage
   data1 = request.form['lcd']
   print ("---Message is", data1)
   lcd_string(data1,LCD_LINE_1)

 return render_template('index.html', value=data1)
if __name__ == "__main__":
    app.debug = True
    app.run('192.168.1.19', port=8080,debug=True)
```

قم بحفظ ملف الكود app.py داخل ملف webapp الذي تم إنشاءه مسبقا. flask سوف يقوم بالبحث عن index.html في ملف templates، الموجود على نفس الملف الذي يحتوي ملف الكود app.py .
بعد ذلك يمكنك تشغيل برنامج البايثون ولكن قبل ذلك تحتاج إلى استبدال عنوان IP في برنامج مع عنوان IP للراسبيري باي الخاصه بك.يمكنك التحقق من عنوان IP عن طريق كتابة الامر التالي على نافذة terminal :

```
Ifconfig
```

بعد تشغيل كود البايثون ثم بفتح http://IP_address_of_your_Pi:8080 في متصفح الويب و ادخل الرسالة ثم انقر فوق زر الارسال .