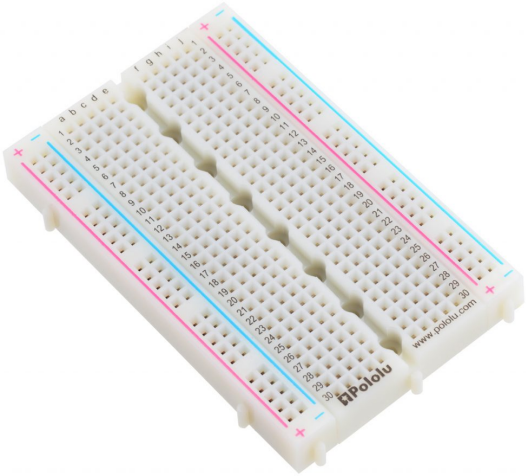
مؤشر أسعار العملة الرقمية بتكوين باستخدام الراسبيري باي

يعد بتكوين (Bitcoin) الشكل الأكثر شهرة من العملات الرقمية، في هذا الدرس سنتعلم صناعة جهاز مؤشر أسعار العملة الرقمية (Bitcoin)، باستخدام راسبيري باي (RaspberryPi) وشاشة عرض كرستالية (LCD)

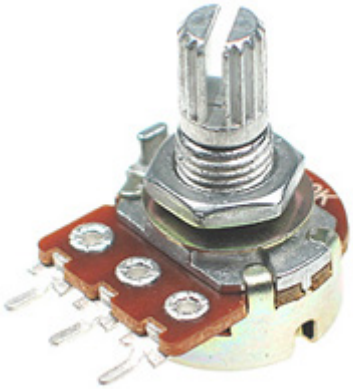
المواد و الأدوات

1X راسبيري باي

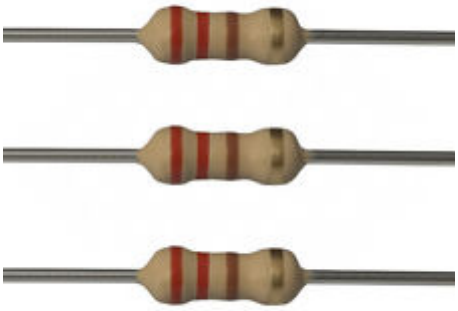
1X شاشة كرستالية (LCD)



1X لوحة تجارب



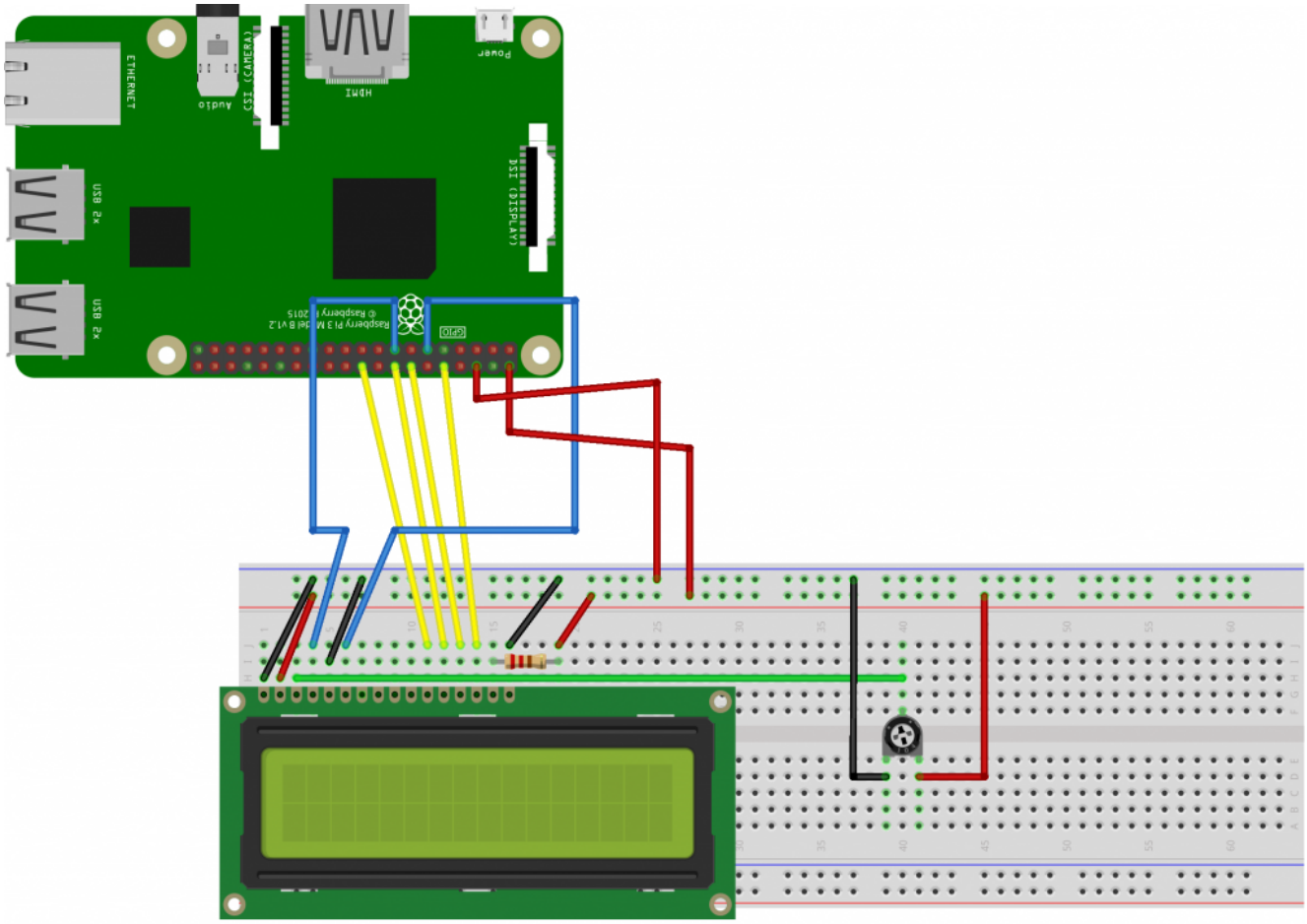
1Xمقاومة متغيرة



1X مقاومة 220 أوم

مجموعة أسلاك توصيل

توصيل الدائرة



راسبيري باي بتكوين

الكود البرمجي

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

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

افتح الشاشة السوداء LXterminal للبدء باستخدام الراسبيري باي.

قم بتحديث النظام بالبداية

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

تثبيت البايثون و مدير المكتبات في البايثون (PIP3)

يتم تثبيت البايثون و(pip3) عند تثبيت نظام التشغيل Raspbian على راسبيري باي كامل ، لكن اذا ثبتت نظام Raspbian Lite فإنه لا يتضمن على البايثون و مدير المكتبات pip3 لذا يجب تثبيتهما بالأمر التالي

```
sudo apt install python3 idle3
```

```
sudo apt-get install python3-pip
```

تحميل مكتبة (Install adafruit-blinka)

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

تثبيت مكتبة (Install adafruit-circuitpython-charlcd)و مكتبة (Install adafruit-circuitpython-charlcd) التي ستساعدنا في تعريف شاشة LCD و الطباعة عليها

```
sudo pip3 install adafruit-circuitpython-charlcd
```

تثبيت مكتبة (json) التي تساعد في ارسال البيانات إلى راسبيري باي بالأمر التالي

```
sudo apt-get install -y php-json
```

نفتح ملف بايثون لكتابة الكود البرمجي من خلال قائمة (Programming) نختر (Python 3) و ننشئ ملف جديد

نكتب الكود البرمجي لتعقب راسبيري باي سعر بتكوين وعرضه على شاشة (LCD)

```
from subprocess import Popen, PIPE
from time import sleep
from datetime import datetime
import json
import requests
import board
import digitalio
import adafruit_character_lcd.character_lcd as characterlcd

# Modify this if you have a different sized character LCD
lcd_columns = 16
lcd_rows = 2

# compatible with all versions of RPI as of Jan. 2019
# v1 - v3B+
lcd_rs = digitalio.DigitalInOut(board.D22)
lcd_en = digitalio.DigitalInOut(board.D17)
lcd_d4 = digitalio.DigitalInOut(board.D25)
lcd_d5 = digitalio.DigitalInOut(board.D24)
lcd_d6 = digitalio.DigitalInOut(board.D23)
lcd_d7 = digitalio.DigitalInOut(board.D18)


# Initialise the lcd class
lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
                                      lcd_d7, lcd_columns, lcd_rows)



# set variables for last price of currency
lastPrice = 0
```

```
# Preload btc price before beginning while loop
def btcPrice():
    try:
        b =
requests.get('https://min-api.cryptocompare.com/data/price?fsym=BTC&tsyms=USD')
        priceFloat = float(json.loads(b.text)['USD'])
        return priceFloat
    except requests.ConnectionError:
        print ("Error querying Crytocompare API")
    for line in btcPrice.splitlines():
        if "state UP" in line:
            dev_name = line.split(':')[1]
    return dev_name


# wipe LCD screen before we start
lcd.clear()

# before we start the main loop
sleep(10)


while True:
    sleep(0.4)
    lastPrice = btcPrice()



    lcd_line_2 = "$ " + str(btcPrice())  + " /BTC"

    # date and time
    lcd_line_1 = datetime.now().strftime('%b %d  %H:%M:%S\n')


    lcd.message = lcd_line_1 + lcd_line_2
    lastPrice = btcPrice()
```

قم بحفظ الكود البرمجي

اعطي أمر التشغيل للكود البرمجي من خلال النقر على قائمة (RUN) ثم (RUN Module)

شرح الكود البرمجي

استدعاء المكتبات لتشغيل المشروع

```
from subprocess import Popen, PIPE
from time import sleep
from datetime import datetime
import json
import requests
import board
import digitalio
```

```
import adafruit_character_lcd.character_lcd as characterlcd
```

تحديد حجم شاشة (LCD)

```
# Modify this if you have a different sized character LCD
lcd_columns = 16
lcd_rows = 2
```

تحديد منافذ شاشة LCD التي تم توصيلها مع راسبيري باي

```
# compatible with all versions of RPI as of Jan. 2019
# v1 - v3B+
lcd_rs = digitalio.DigitalInOut(board.D22)
lcd_en = digitalio.DigitalInOut(board.D17)
lcd_d4 = digitalio.DigitalInOut(board.D25)
lcd_d5 = digitalio.DigitalInOut(board.D24)
lcd_d6 = digitalio.DigitalInOut(board.D23)
lcd_d7 = digitalio.DigitalInOut(board.D18)


# Initialise the lcd class
lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
lcd_d7, lcd_columns, lcd_rows)
```

ننشئ متغير لقيمة سعر البتكوين

```
# set variables for last price of currency
lastPrice = 0
```

ننشئ دالة باسم (btcPrice) وضيفتها أخذ القيم رابط الموقع و حفظها في متغير سعر البتكوين

```
 # Preload btc price before beginning while loop
def btcPrice():
try:
b = requests.get('https://min-api.cryptocompare.com/data/price?fsym=BTC&tsyms=USD')
priceFloat = float(json.loads(b.text)['USD'])
return priceFloat
```

في حال انقطع الاتصال بالانترنت يتم طباعة التص التالي "Error querying Crytocompare API"

```
except requests.ConnectionError:
print ("Error querying Crytocompare API")
```

نحدد السطر الذي سيتم طباعة السعر عليه

```
for line in btcPrice.splitlines():
if "state UP" in line:
dev_name = line.split(':')[1]
return dev_name
```

مسح الشاشة (LCD) بالبداية

```
# wipe LCD screen before we start
lcd.clear()
```

نحدد وقت 10 ثواني لتحديث طباعة القيم

```
# before we start the main loop
sleep(10)


while True:

sleep(0.4)
lastPrice = btcPrice()
```

طباعة سعر بتكوين في السطر الثاني

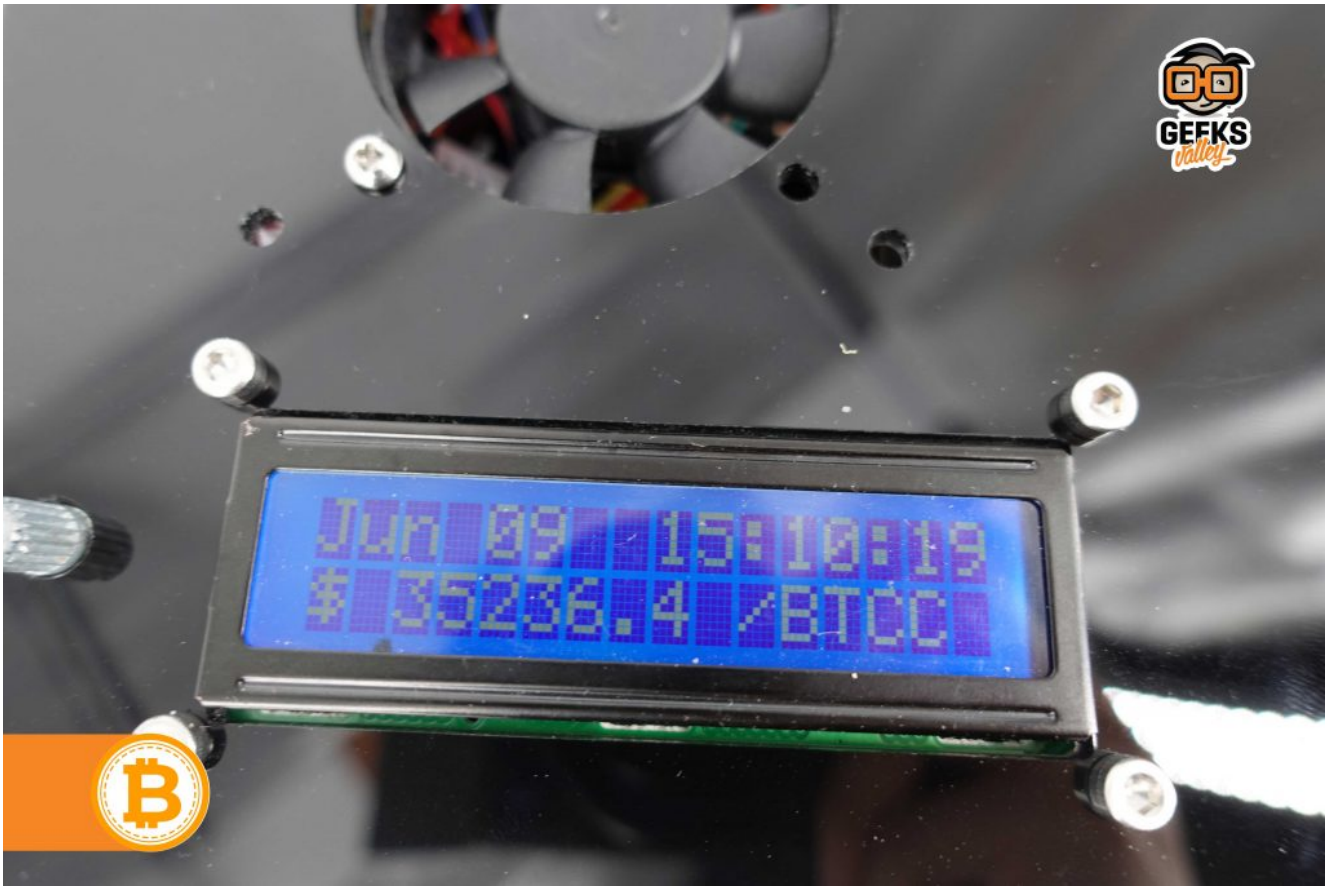
```
lcd_line_2 = "$ " + str(btcPrice()) + " /BTC"
```

طباعة الوقت و التاريخ

```
lcd_line_1 = datetime.now().strftime('%b %d %H:%M:%S\n')

lcd.message = lcd_line_1 + lcd_line_2
lastPrice = btcPrice()
```

وسيت ارسال سعر بتكوين استمرار إلى الراسبيري باي و يقوم البرنامج بعرض السعر على شاشة (LCD)


Jun 09 15:10:19
$ 35236.4 /BTCC