



第11章 物聯網雲端平台： 建立ThingSpeak+Adafruit.IO儀表板

- 11-1 使用ThingSpeak物聯網平台
- 11-2 安裝和使用ThingView App
- 11-3 使用Adafruit.IO物聯網平台
- 11-4 整合應用：上傳OpenWeatherMap目前氣溫





11-1 使用ThingSpeak物聯網平台

- 11-1-1 在ThingSpeak申請帳號和新增頻道
- 11-1-2 建立MicroPython上傳資料至ThingSpeak
- 11-1-3 ThingSpeak支援的小工具





11-1-1 在ThingSpeak申請帳號和新增頻道

- ThingSpeak是免費的物聯網雲端平台，我們可以透過ThingSpeak的頻道（Channel）來即時遠端監控感測器的數據。ThingSpeak提供免費帳號，我們只需申請帳號和新增頻道，就可以將感測器數據上傳至頻道來顯示即時的統計圖表。
 - 步驟一：申請ThingSpeak帳號
 - 步驟二：新增ThingSpeak頻道和取得API金鑰





11-1-1 在ThingSpeak申請帳號和新增頻道

步驟一：申請ThingSpeak帳號

- 在ThingSpeak官方網站可以申請免費帳號，請啟動瀏覽器進入官方首頁：<https://thingspeak.com/>後，按綠色【Get Started For Free】鈕。

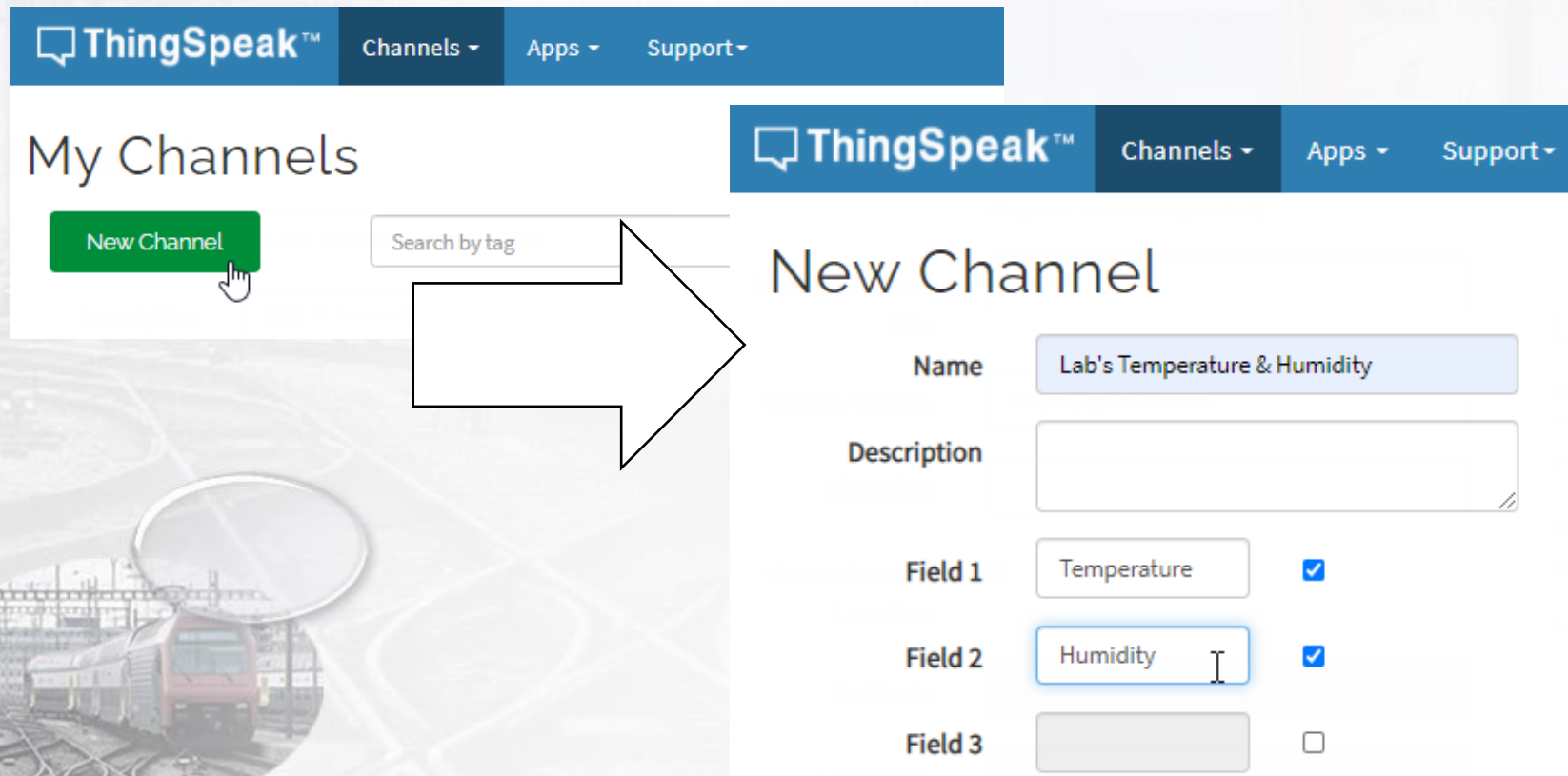




11-1-1 在ThingSpeak申請帳號和新增頻道

步驟二：新增ThingSpeak頻道和取得API金鑰

- 請登入ThingSpeak或繼續上一頁申請步驟，按【New Channel】鈕新增頻道。



The image shows two screenshots of the ThingSpeak website. The left screenshot shows the 'My Channels' page with a green 'New Channel' button. A large white arrow points from this button to the right screenshot. The right screenshot shows the 'New Channel' form with the following fields:

Field	Value	Checkbox
Name	Lab's Temperature & Humidity	
Description		
Field 1	Temperature	<input checked="" type="checkbox"/>
Field 2	Humidity	<input checked="" type="checkbox"/>
Field 3		<input type="checkbox"/>



- 可以看到【Channel ID】（請記下此頻道編號），在下方2個方框顯示2個欄位的統計圖表，請點選中間的【API Keys】標籤。在此標籤顯示Write API Key金鑰字串，請記下金鑰。



11-1-2 建立MicroPython上傳資料至ThingSpeak

- 在ThingSpeak新增頻道後，MicroPython程式是使用Channel ID和WRITE API金鑰字串來上傳感測器資料到ThingSpeak頻道。ThingSpeak上傳頻道資料的HTTP請求，如下所示：

https://api.thingspeak.com/update?api_key=<API_KEY>&field1=<V1>&field2=<V2>

- 上述<API_KEY>是WRITE API金鑰字串，<V1>和<V2>是上傳欄位值，免費帳戶允許每15秒上傳一次資料。





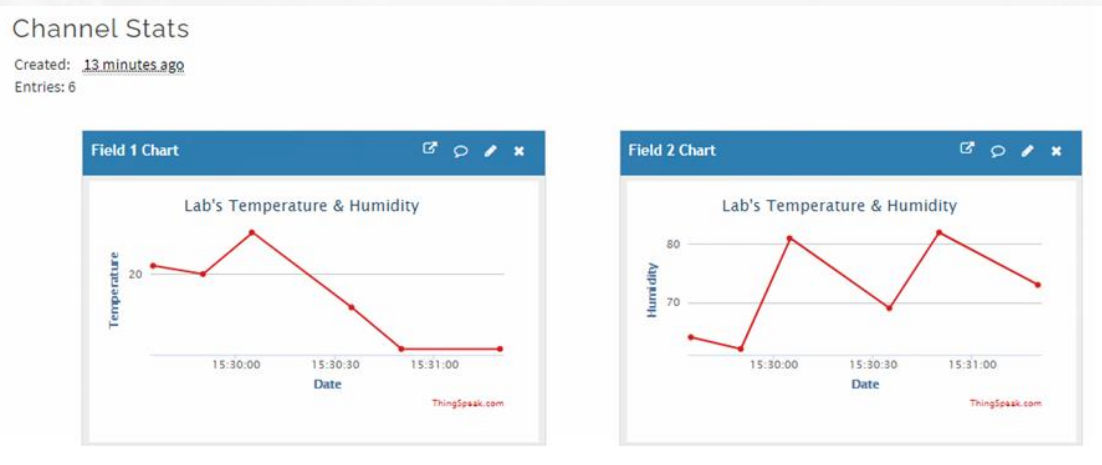
11-1-2 建立MicroPython上傳資料至ThingSpeak：ch11-1-2.py

- MicroPython程式可以使用urequests模組的get()方法送出上傳資料的HTTP請求，或直接呼叫xtools自訂模組的webhook_get()函式，首先匯入相關模組和連線WiFi，如下所示：

```
import xtools, utime
```

```
xtools.connect_wifi_led()
```

```
WRITE_API_KEY = "<WRITE API金鑰>"
```





11-1-2 建立MicroPython上傳資料至ThingSpeak：ch11-1-2.py

while True:

temp = xtools.random_in_range(10, 35)

hum = xtools.random_in_range(60, 90)

print("儲存溫度和濕度資料:", temp, hum)

url = "http://api.thingspeak.com/update?"

url += "api_key=" + WRITE_API_KEY

url += "&field1=" + str(temp)

url += "&field2=" + str(hum)

print(url)

xtools.webhook_get(url)

utime.sleep(15)





11-1-3 ThingSpeak支援的小工具

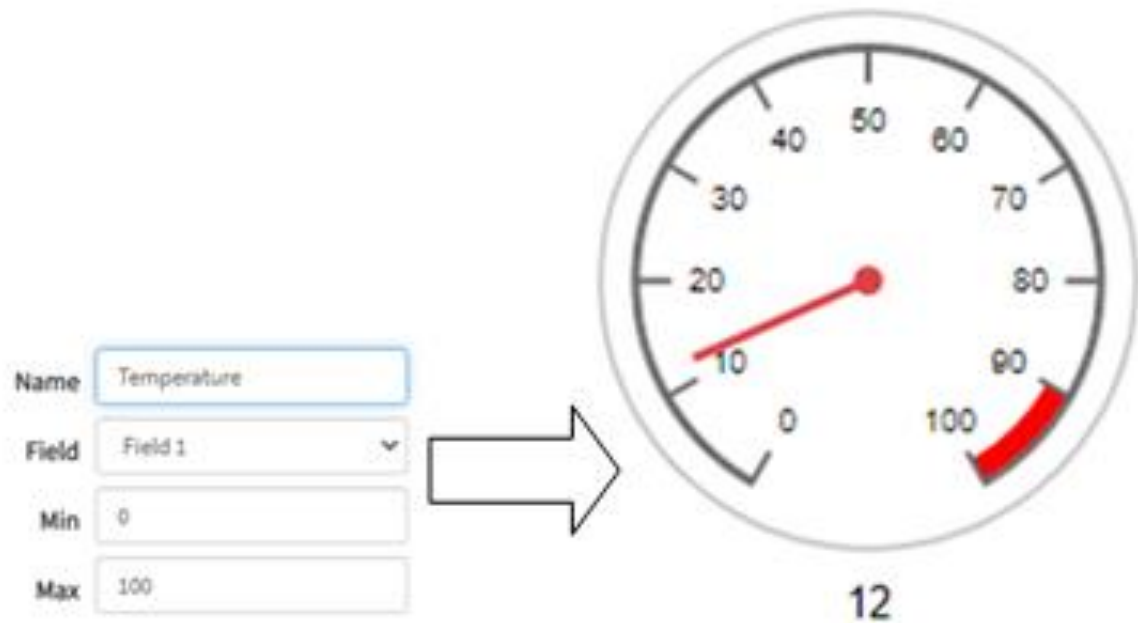
- 在ThingSpeak頻道除了支援折線圖外，還提供多種小工具（Widgets）來顯示數據。請在My Channels頁面點選頻道名稱的超連結開啟頻道後，按【Add Widgets】鈕新增小工具，如下圖所示：

The screenshot displays the ThingSpeak web interface for a channel titled "Lab's Temperature & Humidity". The channel ID is 1256814, the author is mwa0000020614554, and the access is public. Below the channel information, there are tabs for "Private View", "Public View", and "Channel Settings". A large white arrow points from the "Add Widgets" button in the "Channel Stats" section to a modal window titled "Click on a widget to add it to the Channel". The modal window shows three widget options: "Gauge" (a circular gauge with a needle), "Numeric Display" (a box showing the value 1516.12), and "Lamp Indicator" (a circular indicator). At the bottom of the modal, there are "Next" and "Cancel" buttons.



11-1-3 ThingSpeak支援的小工具 Gauge小工具

- **Gauge**小工具是使用指針顯示欄位值的計量表，在指定名稱（**Name**）和顯示欄位（**Field**）後，即可新增**Gauge**顯示欄位資料，如下圖所示：





11-1-3 ThingSpeak支援的小工具 Numeric Display小工具

- Numeric Display小工具是使用數字顯示欄位資料，在指定名稱（**Name**）和顯示欄位（**Field**）後，即可顯示此欄位的數字資料，如下圖所示：

Name: Humidity

Field: Field 2

Update Interval: Field 2 second(s)

Units: Enter Measurement Units

Preview: 67 an hour ago





11-1-3 ThingSpeak支援的小工具 Lamp Indicator小工具

- Lamp Indicator小工具是一顆指示燈，在指定名稱（**Name**）和輸入使用欄位（**Field**）且建立**If**條件後，當**If**條件成立時，就點亮指示燈（即顯示最下方指定的色彩），如下圖所示：

Name: High Temperature

Condition if

Field 1

is greater than

30 I

turn Lamp ON

Update Interval: 15 second(s)

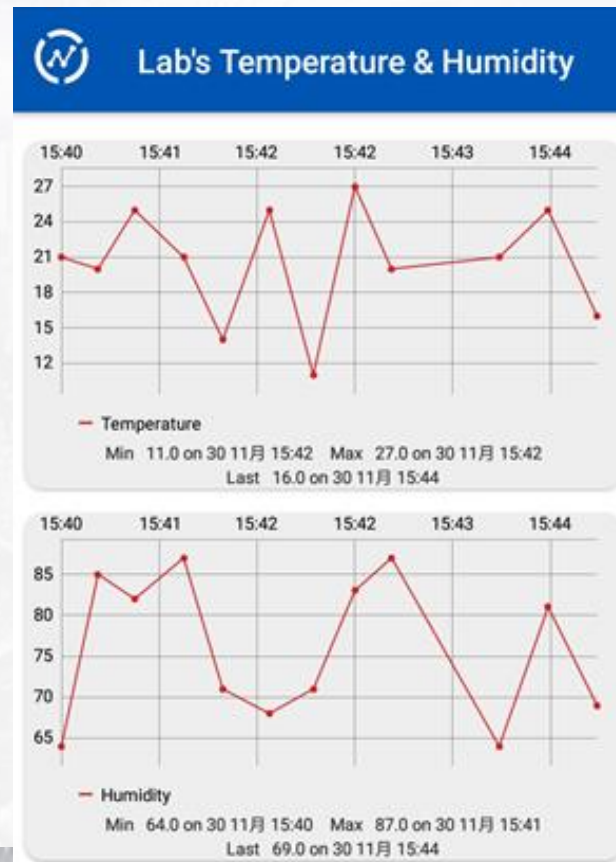
Color:

a few seconds ago



11-2 安裝和使用ThingView App

- ThingSpeak提供智慧型手機的ThingView App，可以讓我們使用手機來檢視頻道的統計圖表，如下圖所示：





11-3 使用Adafruit.IO物聯網平台

- 11-3-1 申請與使用Adafruit.IO物聯網平台
- 11-3-2 建立MicroPython上傳資料至Adafruit.IO
- 11-3-3 在Adafruit.IO儀表板新增更多區塊





11-3-1 申請與使用Adafruit.IO物聯網平台

- Adafruit.IO是另一個免費物聯網雲端平台，此平台可以自行建立儀表板來即時遠端監控上傳感測器的數據。
- Adafruit.IO提供免費帳號，我們只需申請帳號和新增**FEED**，就可以建立儀表板來新增折線圖，然後將感測器數據上傳和顯示統計圖表。
 - 步驟一：申請Adafruit.IO帳號與新增**FEED**
 - 步驟二：建立儀表板Dashboard
 - 步驟三：取得 Adafruit.IO Key (AIO KEY)





11-3-1 申請與使用Adafruit.IO物聯網平台

步驟一：申請Adafruit.IO帳號與新增FEED

FIRST NAME

Joe

LAST NAME

Chen

EMAIL

hueyanchen2014@gmail.com

USERNAME

hueyanchen2014

Username is viewable to the public on the forums, Adafruit IO, and elsewhere.

PASSWORD

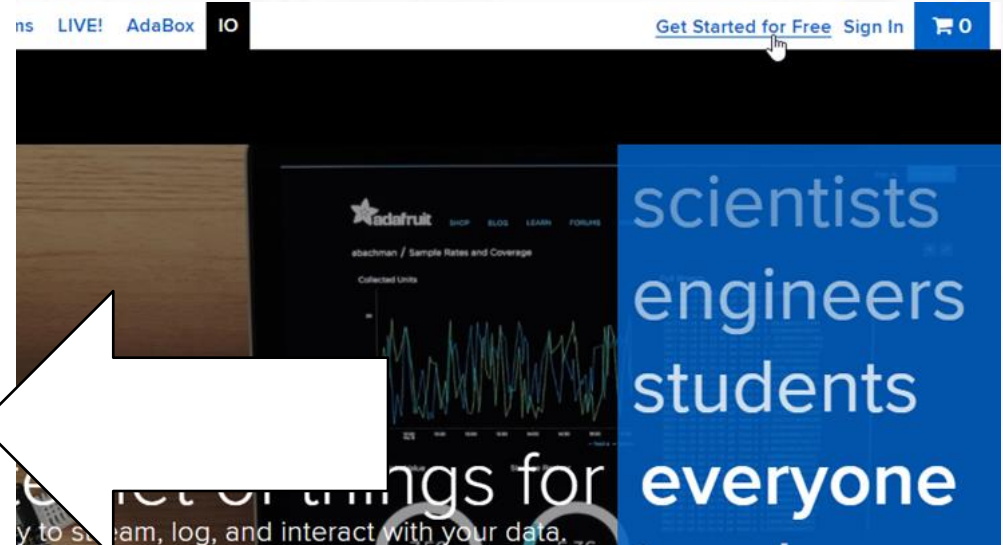
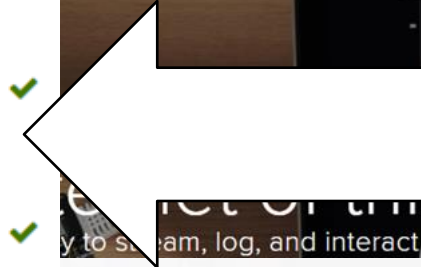
.....

CREATE ACCOUNT



HAVE AN ADAFRUIT ACCOUNT?

SIGN IN



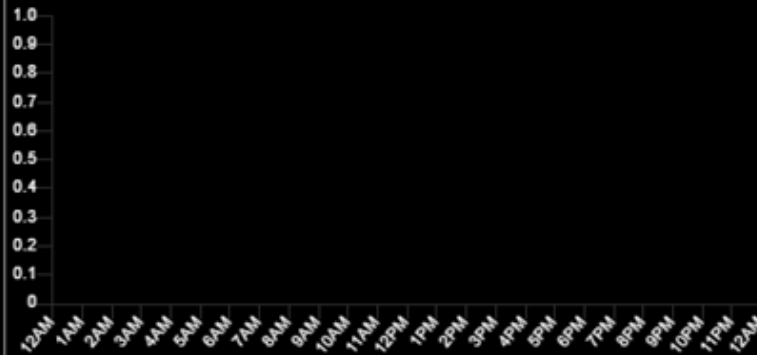


11-3-1 申請與使用Adafruit.IO物聯網平台

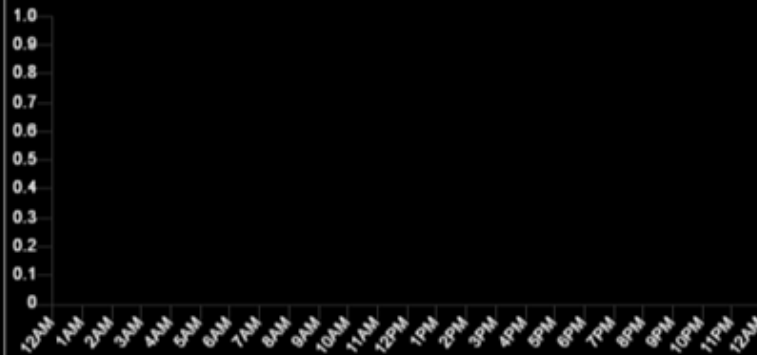
步驟二：建立儀表板Dashboard

hueyanchen2014 > Dashboards > 電腦教室的溫溼度

Temperature



Humidity





11-3-1 申請與使用Adafruit.IO物聯網平台

步驟三：取得 Adafruit.IO Key (AIO KEY)

Shop Learn Blog Forums LIVE! AdaBox IO Hi, Joe Chen | Account 0


adafruit Profile Feeds Dashboards Triggers Services My Key

The Adafruit IO SSL/TLS Certificate was updated. More details can be found here.

YOUR ADAFRUIT IO KEY

Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.



Username

Active Key

[Hide Code Samples](#)

Arduino

```
#define IO_USERNAME "hueyanchen2014"
#define IO_KEY "XXXXXXXXXXXXXXXXXXXXkvqRgB9s"
```



11-3-2 建立MicroPython上傳資料至Adafruit.IO

- 在Adafruit.IO新增FEED和儀表板的折線圖區塊後，MicroPython程式是使用Username和Active Key（AIO Key）上傳感測器資料到Adafruit.IO的FEED。上傳資料的HTTP POST請求，如下所示：
https://io.adafruit.com/api/v2/<USER>/feeds/<FEED>/data?X-AIO-Key=<AIO_KEY>
- 請注意！上述<FEED>的FEED名稱是Key欄位的小寫英文，例如：使用Key欄位的temperature；不是FEED名稱Temperature。





11-3-2 建立MicroPython上傳資料至 Adafruit.IO : ch11-3-2.py

- 因為MicroPython程式上傳的資料是Python字典，我們需要使用xrequests.py模組的post()方法送出HTTP請求，或直接呼叫xtools自訂模組的webhook_post()函式，首先匯入相關模組和連線WiFi，如下所示：

```
import xtools, utime
xtools.connect_wifi_led()
ADAFRUIT_IO_USERNAME = "hueyanchen2014"
ADAFRUIT_IO_KEY = "<AIO KEY>"
FEED1 = "temperature"
FEED2 = "humidity"
```





11-3-2 建立MicroPython上傳資料至 Adafruit.IO : ch11-3-2.py

while True:

```
temp = xtools.random_in_range(10, 35)
```

```
hum = xtools.random_in_range(60, 90)
```

```
print("儲存溫度和濕度資料: ", temp, hum)
```

```
url = "https://io.adafruit.com/api/v2/" + ADAFRUIT_IO_USERNAME
```

```
url+= "/feeds/" + FEED1 + "/data?X-AIO-Key=" + ADAFRUIT_IO_KEY
```

```
print('url1=', url)
```

```
data1 = {"value": temp}
```

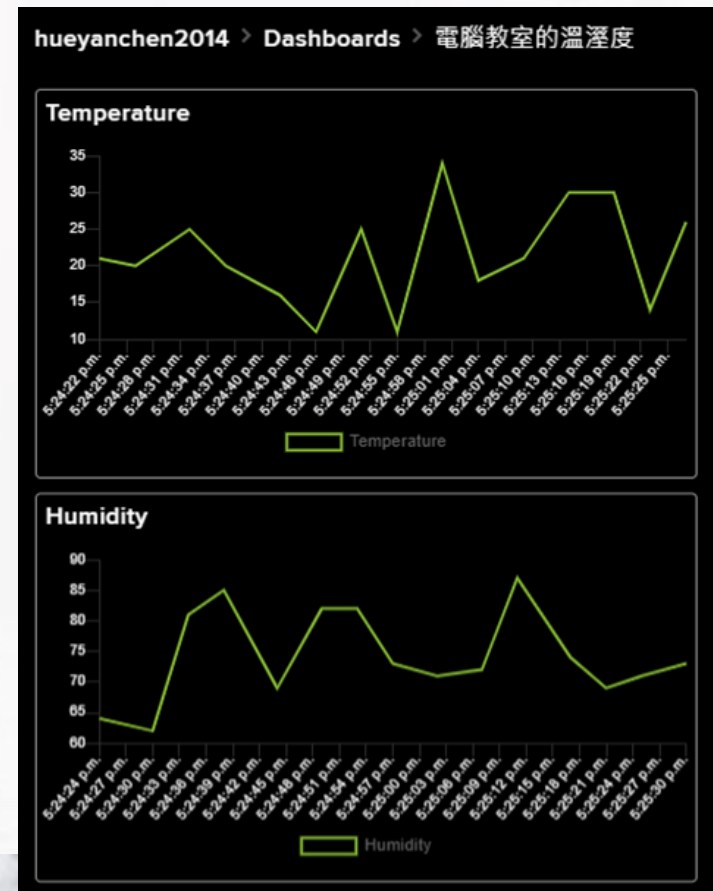
```
xtools.webhook_post(url, data1)
```





11-3-2 建立MicroPython上傳資料至 Adafruit.IO : ch11-3-2.py

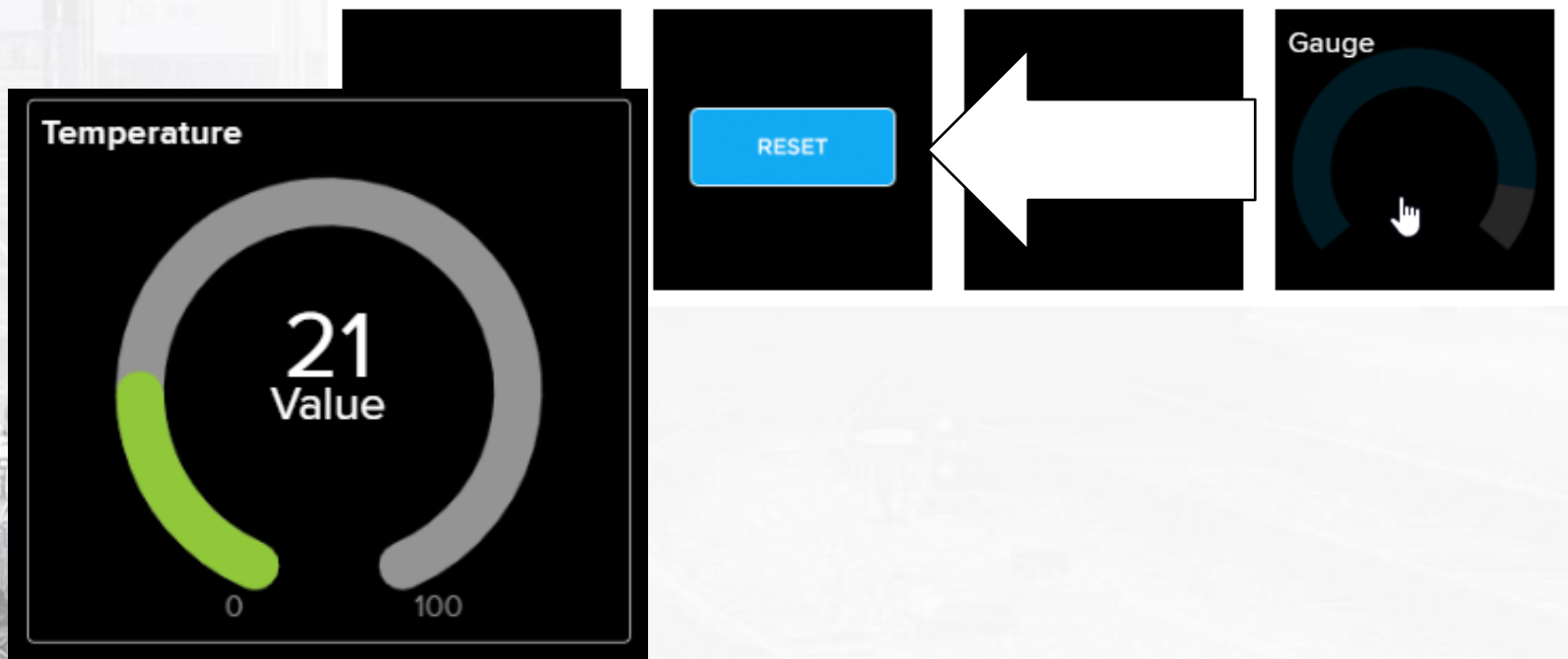
```
url = "https://io.adafruit.com/api/v2/" + ADAFRUIT_IO_USERNAME
url+= "/feeds/"+ FEED2 + "/data?X-AIO-Key=" + ADAFRUIT_IO_KEY
print('url2=', url)
data2 = {"value": hum}
xtools.webhook_post(url, data2)
utime.sleep(5)
```





11-3-3 在Adafruit.IO儀表板新增更多區塊 Gauge區塊

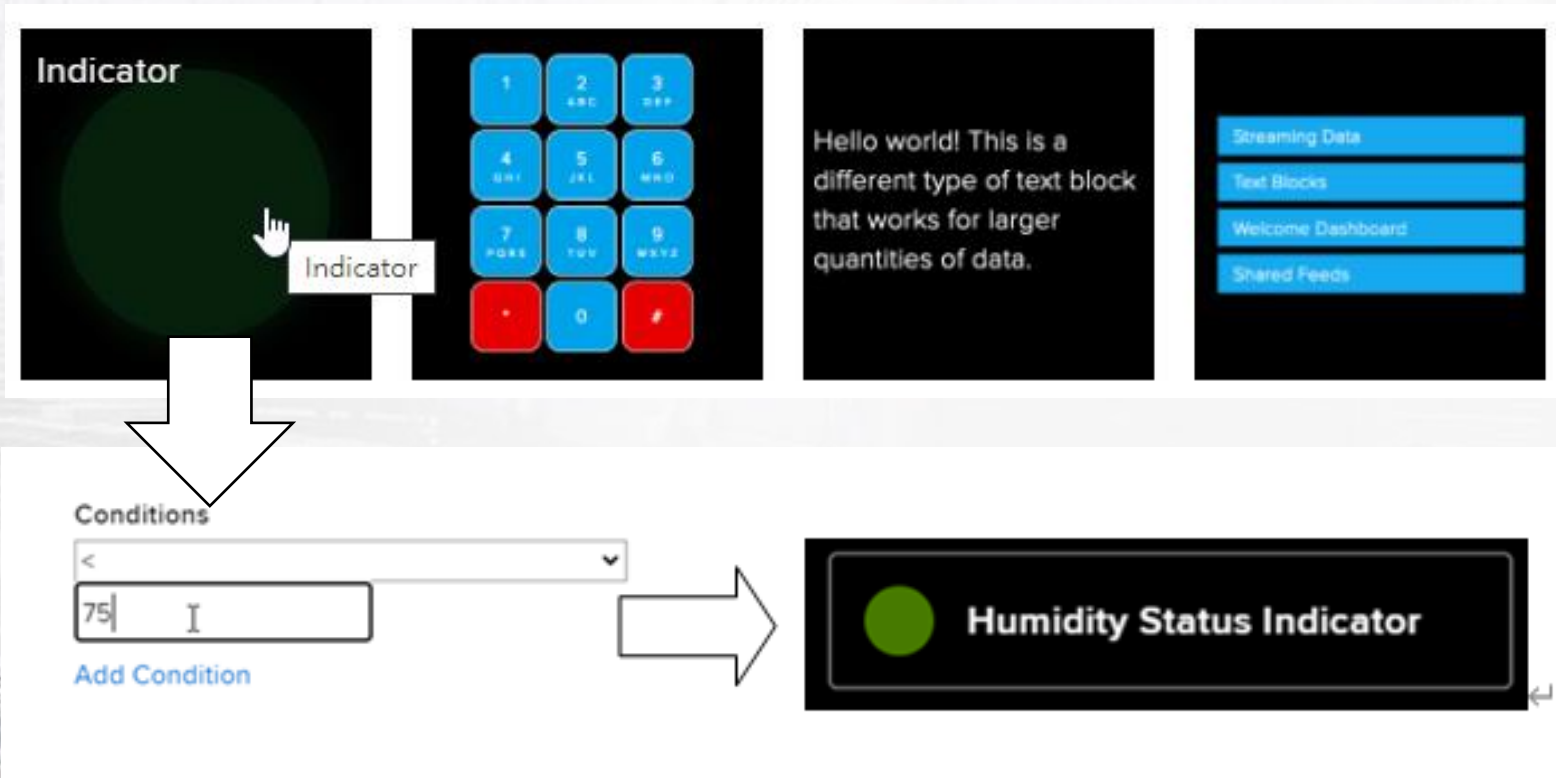
- 在Adafruit.IO儀表板可以新增多種不同區塊的元件，請先點選右方的向上箭頭圖示，再點選【Create New Block】新增其他區塊。Gauge區塊是使用指針顯示FEED值的計量表，請點選【Gauge】區塊。





11-3-3 在Adafruit.IO儀表板新增更多區塊 Indicator區塊

- Indicator區塊就是一顆指示燈，請點選【Indicator】區塊。





11-4 整合應用：上傳OpenWeatherMap目前氣溫-編輯Adafruit.IO儀表板的版面配置

- Adafruit.IO儀表板支援版面配置，可以調整區塊尺寸和位置。請先點選右方的向上箭頭圖示，再點選【Edit Layout】編輯版面配置，如下圖所示：





11-4 整合應用：上傳OpenWeatherMap目前氣溫-上傳OpenWeatherMap的目前氣溫：ch11-4.py

- MicroPython程式可以將OpenWeatherMap目前氣溫上傳至名為Temperature的FEED來更新儀表板的折線圖。MicroPython程式碼是使用get_temperature()函式來取得OpenWeatherMap的目前氣溫，如下所示：

```
12: ...
```

```
13: def get_temperature():
```

```
14:     url =
```

```
     "https://api.openweathermap.org/data/2.5/weather?"
```

```
15:     url += "q=" + city + "," + country # 城市與國別
```

```
16:     url += "&units=metric&lang=zh_tw&" # 單位
```

```
17:     url += "appid=" + API_key
```



11-4 整合應用：上傳OpenWeatherMap目前 氣溫-上傳OpenWeatherMap的目前氣溫：ch11-4.py

```
18:     try:
19:         response = urequests.get(url)
20:         data = ujson.loads(response.text)
21:         main = data["main"]
22:         return main["temp"]
23:     except:
24:         return 0
25: ...
```

