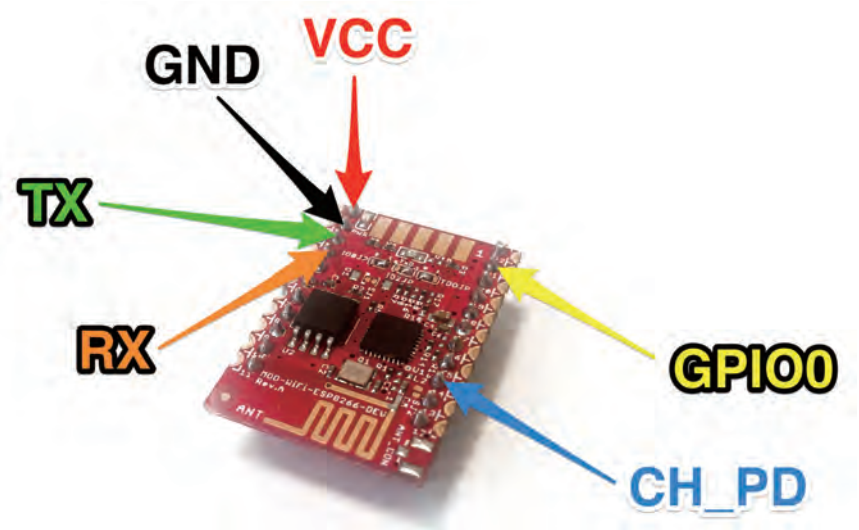
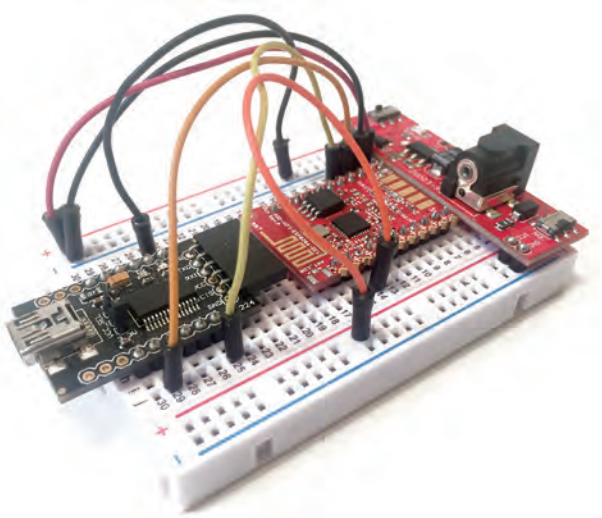
這是 Olimex 板：



這是 Olimex 板最後的樣子：

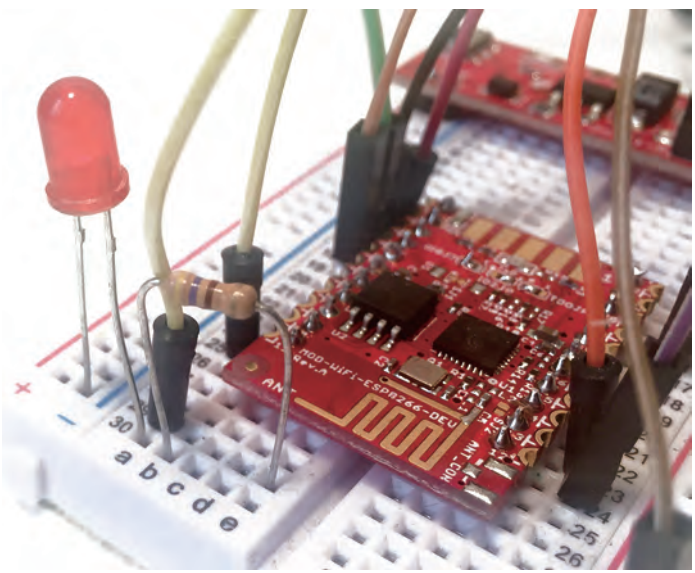


請確認您確實根據了示意圖把所有東西都接好了，否則後面什麼事都無法進行。

控制 LED

首先，讓我們來看看如何控制一個簡單的 LED。ESP8266 的 GPIO 腳位能夠理解許多不同的功能：輸入、輸出、PWM 輸出以及 SPI 或是 I^2C 通訊。第一個專案將教導您如何使用晶片上的 GPIO 腳位來輸出：

1. 第一步就是在專案中加進一顆 LED。以下是在這個專案中會用到的額外元件：
   - 5mm LED（*https://www.sparkfun.com/products/9590*）
   - 330 歐姆電阻，用來限制流 LED 的電流（*https://www.sparkfun.com/products/8377*）
2. 我們要藉由電組將 LED 連上 ESP8266 板。請先將電阻裝上麵包板。
3. 將 LED 裝上麵包板並將 LED 上最長的腳位 (陽極) 接上電阻的其中一個腳位。
4. 將電阻的另一個腳位接上 ESP98266 上的 GPIO 5 號腳位，再將 LED 的另一個腳位接地。完成圖如下：

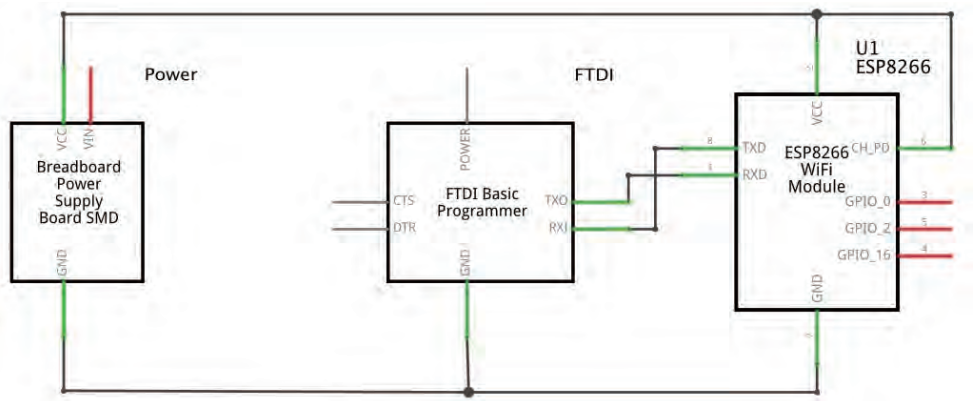


請依照以下步驟安裝 Arduino 函式庫：

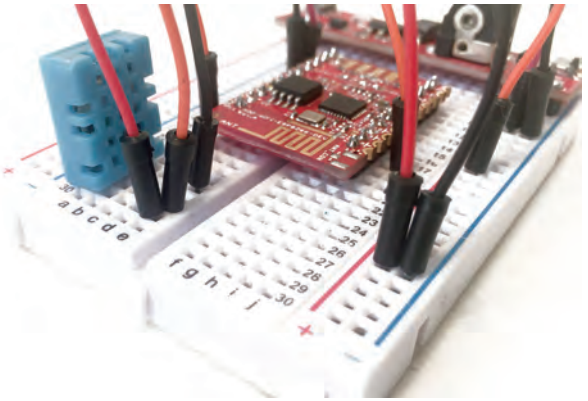
1. 首先，請由 GitHub 下載所要的函式庫。
2. 接著，進入 Arduino IDE，然後請由 **Sketch |Include Library | Add .ZIP Library** 來加入這個函式庫。
3. 最後，點選您剛才下載好的檔案。

硬體設定

首先，我們要來了解如何讓硬體能夠使用 ESP8266 開發板。以下會告訴您如何連結各種元件：



1. 基本上，您得將麵包板供電模組的 VCC 與 GND 腳位分別接到 ESP8266 的 VCC 與 GND。並且將 FTDI 轉接器的 GND 腳位和接上 ESP8266GND。
2. 接下來，將 FTDI 板的 TX 接在 ESP8266 板的 RX 上，然後 RX 接在 TX 上。
3. 最後，將 ESP8266 板的 CH_PD (或者 CHIP_EN) 腳位接上 VCC。
4. 完成之後，請將 DHT11 感測器插上麵包板。
5. 接著，將左邊的腳位接在 VCC (紅色電源軌) 上，右邊的腳位接在 GND (藍色電源軌) 上。請將 VCC 旁的腳位接在 ESP8266 晶片的 GPIO 5 號腳位上。完成圖如下，但未顯示 USB-to-Serial 線：



請您確定已經依照示意圖完成所有接線，不然會沒有辦法繼續。也要確定在元件 (FTDI 模組和電源) 的開關都設定為 3.3V，否則會害您的晶片受損。

6. 另外，請在 ESP8266 GPIO 的 0 號腳位多接一條線出來，另一端現在先不接任何東西，等會兒我們會需要它將晶片設成燒錄模式。

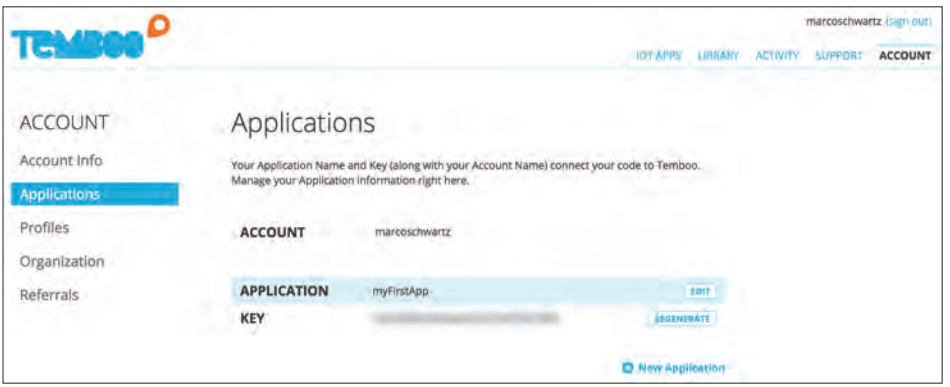
測試感測器

現在要使用感測器了。再次提醒您，我們用的是 Arduino IDE，所以可以像在 Arduino 開發板上一樣寫程式。在此只會將溫度顯示在 Arduino IDE 的序列監視器視窗上。若您尚未從 Arduino IDE 函式庫安裝 Adafruit DHT 感測器函式庫，請您立即安裝。

以下是本段的完整程式碼：

```
// 匯入函式庫
#include "DHT.h"
// 定義感測器腳位
#define DHTPIN 5
// 選用 DHT11 感測器
#define DHTTYPE DHT11
// 初始化 DHT 感測器
```

完成後，前往 Account 頁面並建立您的第一個應用程式。之後在本章您會需要 APPLICATION 和 KEY：



您也會用到我為了讓它能用於 ESP8266 而修改的 `Temboo` 自訂函式庫。您可以在本章的程式碼中找到，就在本書的 GitHub 文件庫中：

https://github.com/openhomeautomation/iot-esp8266-packt

接著，請您前往 **Arduino** 函式庫資料夾（Windows 用戶位於 `C:\Program Files (x86)\Arduino\libraries`; OS X 用戶請在 Arduino 應用程式中點擊 **Show package content**）。請記得儲存原有的 `Temboo` 函式庫以便您之後要用在 ESP8266 以外的專案上。完成後，刪除原有的 `Temboo` 函式庫，並將您從本章程式碼中取得的函式庫取代進去。

從 Yahoo 取得天氣資料

在本章的第一個專案中，我們將學習如何從 Yahoo 天氣取得資料。多虧 Temboo，這件事變得非常簡單：

1. 首先，前往以下 URL：

   https://temboo.com/library/Library/Yahoo/Weather/ GetWeatherByAddress/

恭喜！現在您能夠在 Twitter 帳號上發佈來自 ESP8266 的資料了！

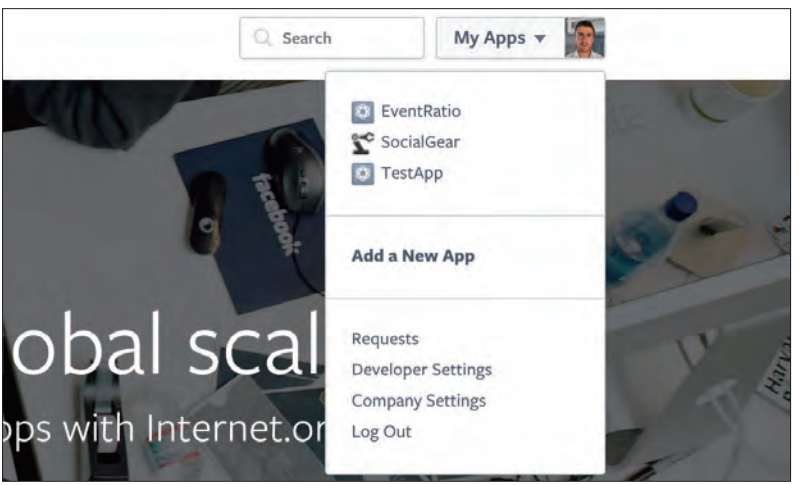
從 ESP8266 發佈 Facebook 狀態

在本章的最後一個專案中，我們要來學習如何藉由 Temboo 用 ESP8266 晶片和 Facebook 互動。在此的範例只是單純地發佈一個狀態，但是您也可以用同樣的方法發佈在朋友的動態牆上或是在某個頁面上。

1. 首先，您必須要建立一個 Facebook 應用程式。請前往以下網址：

   https://developers.facebook.com/

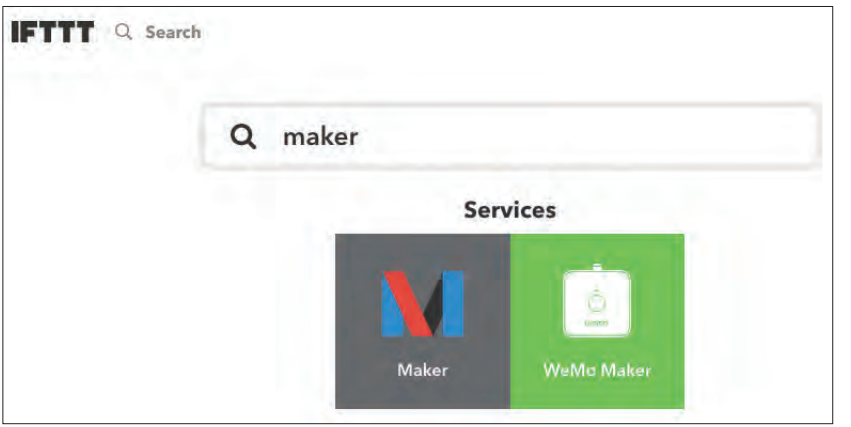
2. 點擊 **Add a New App**：



寄發電子郵件通知

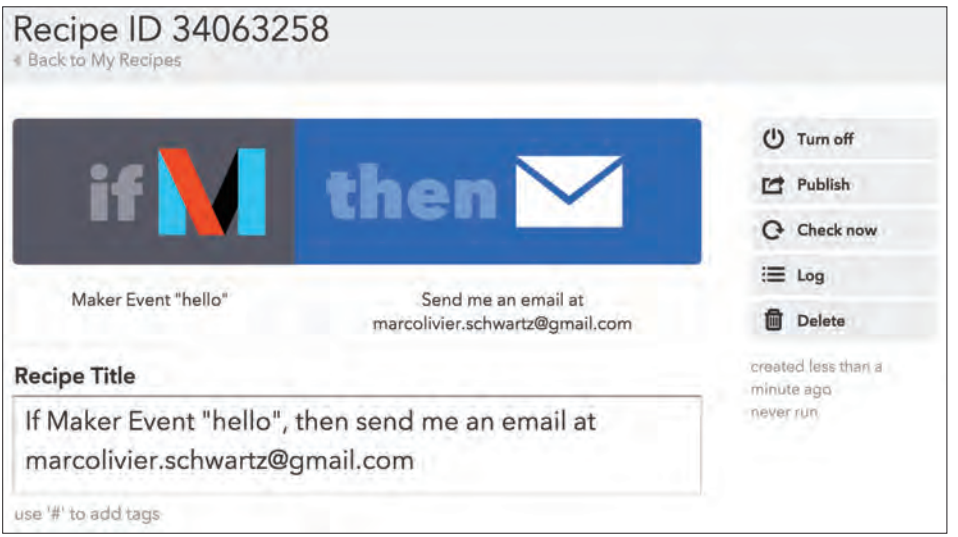
現在是時候來製作第一個專題了：發送 e-mail 通知！作為第一個通知的做法，我們會定期對您所指定的 e-mail 收件人發送訊息。

要把 ESP8266 連上 IFTTT 需要一個稱為 **Maker** 的服務，在 IFTTT 上就可找到。請點選頁面左上角的 **Search** 後搜尋這個服務：



開啟之後，請點選 Connect 將它與您的 IFTTT 帳號連接起來：

按下 Create action 之後把這個 applet 存檔，接著就可以在 IFTTT 儀表板中看到它了：



現在這個 applet 已經啟動，可以來編寫 Arduino 草稿碼來實際觸發它了。按照慣例，在此我只會介紹草稿碼一些最重要的地方：

1. 草稿碼首先匯入了 ESP8266 Wi-Fi 函式庫：

```
#include <ESP8266WiFi.h>
```

2. 輸入要連接的 Wi-Fi 網路帳號密碼：

```
const char* ssid =     "wifi-name";
const char* password = "wifi-pass";
```

3. 接著要輸入 IFTTT 的相關資訊，例如事件名稱以及您的 IFTTT Maker 服務金鑰：

```
const char* host = "maker.ifttt.com";
const char* eventName = "hello";
const char* key = "your-key";
```

硬體與軟體需求

本專題仍然需要用到 ESP8266 晶片。我在本書大部分的專題都是使用 Adafruit 的 Huzzah ESP8266 模組，但任何一款 ESP8266 模組都沒問題。

我選用的是 12V DC 直流磁簧門鎖，市面上任何一款相同規格的鎖頭當然也可以。您還需要一些元件才能操作這組門鎖：一個 NPN 電晶體、一個保護用的二極體還有一個 1K 歐姆電阻。我把所有元件的相關文件都整理在本段中。最後還需要一個可接上麵包板的 12V 直流電源。

以下是我用於本專題的門鎖：

您還需要一個 3.3V/5V FTDI USB 模組將程式燒錄到 ESP8266 晶片。最後需要的是一些跳線與一片麵包板。

以下是本章中所用到的元件清單：

- Adafruit ESP8266 模組（*https://www.adafruit.com/products/2471*）
- FTDI USB 模組（*https://www.adafruit.com/products/284*）
- 門鎖式電磁閥（*https://www.adafruit.com/products/1512*）
- NPN 電晶體（*https://www.sparkfun.com/products/13689*）

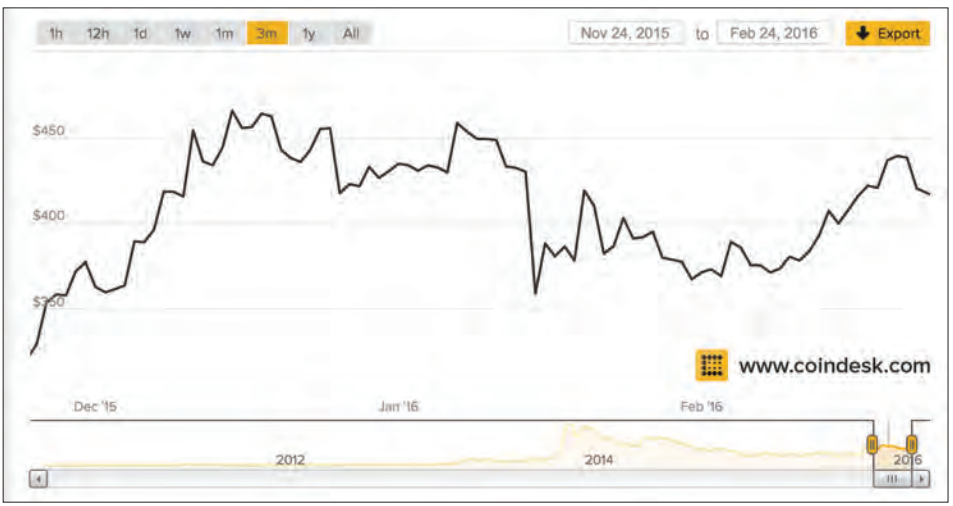
線上比特幣服務

開始使用線上比特幣服務之前，我們得先知道如何取得比特幣的目前價格（美金為單位）。

在網路上很容易就能取得比特幣的目前價格或是歷史價格走勢。例如 Coindesk 就是個不錯的網站：

http://www.coindesk.com/price/

您可以在網站上取得比特幣的目前價格，還有相當漂亮的圖表來顯示比特幣的歷史價格走勢：

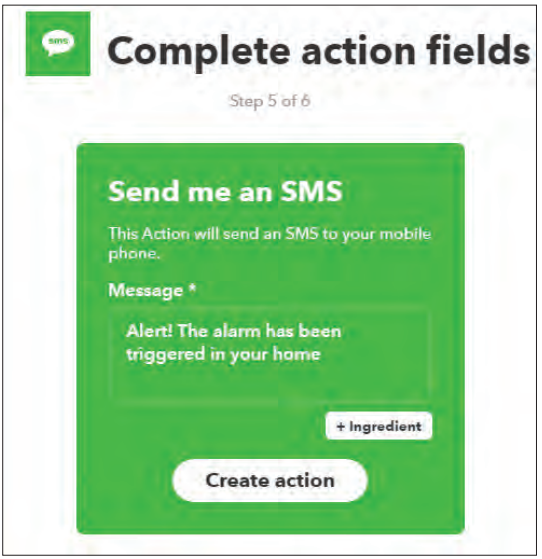


這樣很好，但只能在網站上操作，這對於咱們的 ESP8266 小板子來說太複雜了。

有個更好的做法就是透過稱為 bitcoin ticker 的服務，它可以顯示比特幣的即時價格。以下這個網站還蠻好用的：

http://preev.com/btc/usd

訊息內容要寫什麼都可以，例如：



現在請建立這個 applet 並啟動它，您會發現當把手放在感測器前方時，您的手機馬上就會收到一則警示訊息。如果要關閉這個警報系統的話，只要從 IFTTT 上關閉這個 applet 就好，非常簡單。

這個系統可以加裝更多感測器，各自都能在 IFTTT 上發布同樣的警示訊息。因此只要有任一個感測器偵測到動態，您的手機都會收到警示。

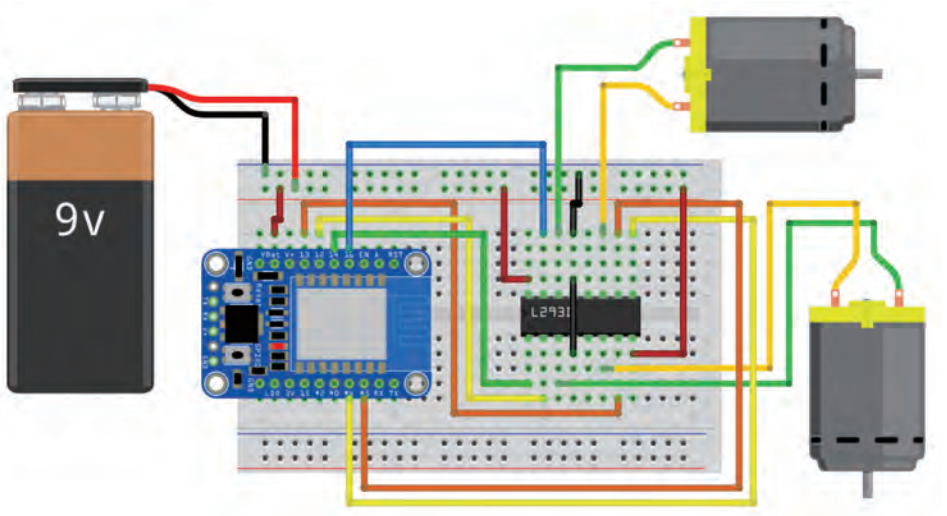
自動化您的家

到了本章最後一段，我們要使用 IFTTT 多變一點把戲。這次不使用動態感測器，我們要看看如何使用不同的 IFTTT 觸發器來建立一個可控制 LED 模組的自動化流程。當然啦，您可以把 LED 換成任何您想控制的家電，例如檯燈。

IFTTT 的 **Maker** 服務也可作為動作服務，這次我們就是要這麼做。當指定狀況發生時，我們會用它來呼叫 aREST API。

硬體設定

現在我們要來組裝本專案的硬體了。因為有點複雜，所以我繪製了詳細圖解，讓您了解各部分不同的接線：



我也畫了一個詳細的電路圖來幫助您：