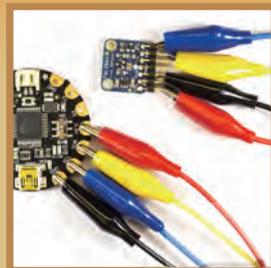
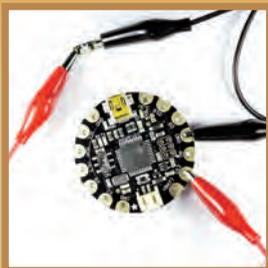
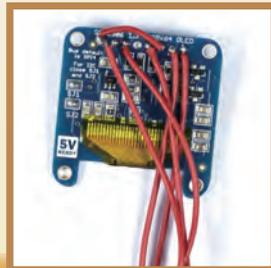
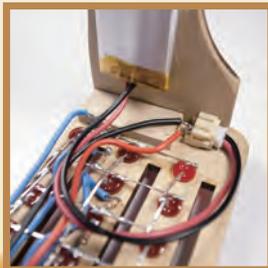
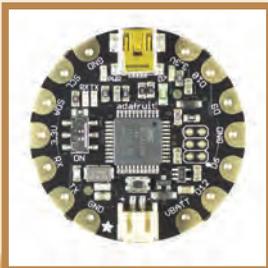
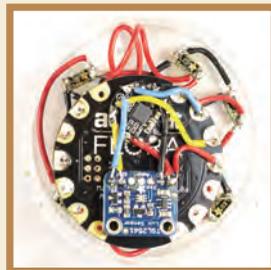
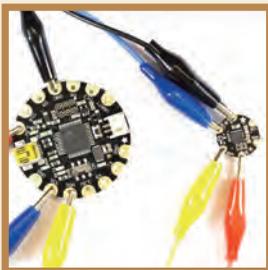


CHAPTER

1

初探 Arduino 與 燈光閃爍



碁峯

www.gotop.com.tw

在 IDE 畫面的最右邊，可看到一個長得像放大鏡的按鈕，作用是開啟**序列埠監控視窗 (Serial monitor)**，按下後會開啟另一個新視窗，讓電腦與開發板兩者之間能進行傳輸溝通，其用途非常廣泛，我們將於第 2 章「感測器」深入介紹。

在 IDE 或螢幕的最上方，可看到基本的應用程式選單，根據你所使用的作業系統，其模樣或許稍有不同，但結構都是一樣的。在檔案 (**File**) 選單裡，可找到各種選項，包括開啟先前儲存的草稿碼，以及 IDE 內建附帶的範例草稿碼，如圖 1.5 所示。在編輯 (**Edit**) 選單裡，可找到各種不同的選項，以及與編輯程式相關的快速指令。在草稿碼 (**Sketch**) 選單裡，可找到與 IDE 視窗按鈕具備相同功能的選項。

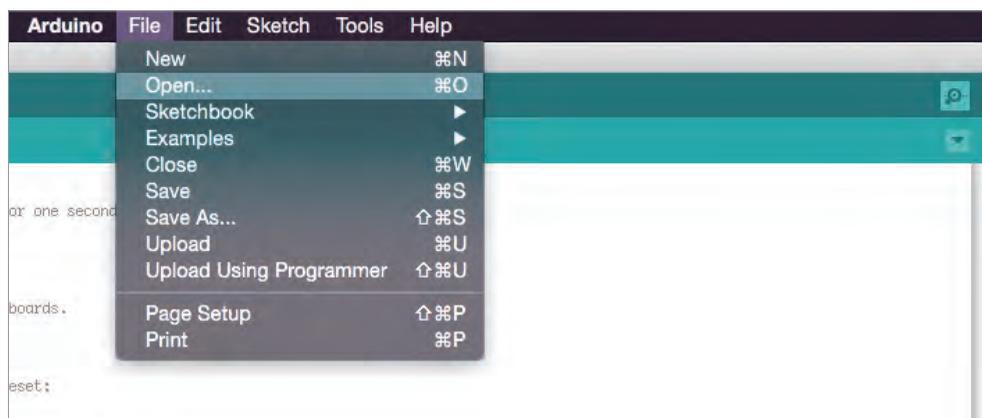


圖 1.5「檔案」選單

在工具 (**Tools**) 選單裡，裡頭有兩個非常重要的選項，當要上傳草稿碼到開發板之前，必須先確認設定是否正確。在「工具 > 板子 (Board)」裡，會列出各種不同的 Arduino 開發板，你必須根據開發板類型選擇正確的設定。在「工具 > 序列埠 (Serial port)」裡，必須根據開發板連接到電腦的哪個 USB 埠來設定，依照不同的作業系統，此傳輸埠的名稱也會有所不同。若是 Windows，其名會是 **COM***，若是 OS X，則會被命名為 **/dev/tty.******。

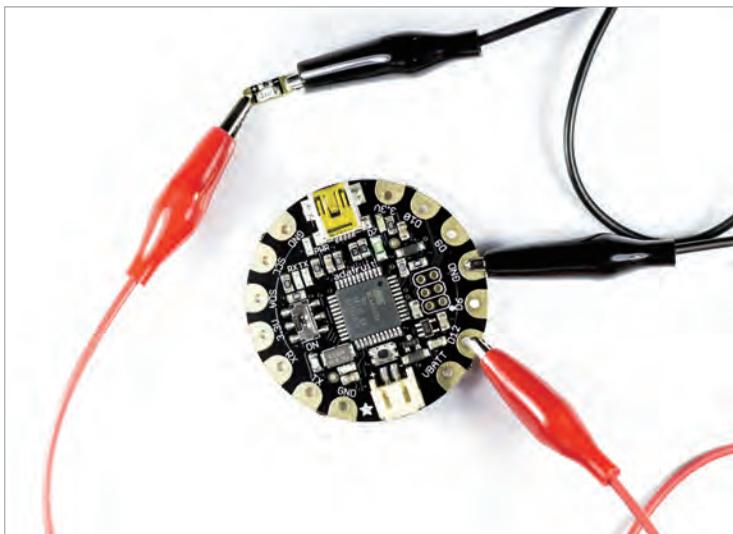


圖 1.10 使用鱷魚夾連接 LED 和板子

此處使用鱷魚夾連接 LED 與 FLORA 開發板，鱷魚夾其實就是在一般電線的兩端加上金屬夾子，在建構專案原型時是非常便利的工具，而且特別適用於穿戴式 Arduino 開發板，例如 FLORA。LED 的腳位分為正極與負極，以圖 1.10 裡的 LED 為例，印刷電路板上標示為「+」的那端是正極，「-」那端則為負極，請把正極接到 FLORA 板子腳位 **D12**，接著為了形成完整的電路，把負極接到板子的 **GND**。

不同閃爍速率

底下草稿碼示範如何使用 for 迴圈，讓 LED 以不同速率閃爍：

```
int led = 12; // 宣告變數，命名為 led，接到腳位 12

void setup() {
  pinMode(led, OUTPUT); // 設定 led 為輸出腳位
}

void loop() {
  // for 迴圈，直到 i 大於 5 才停止
```

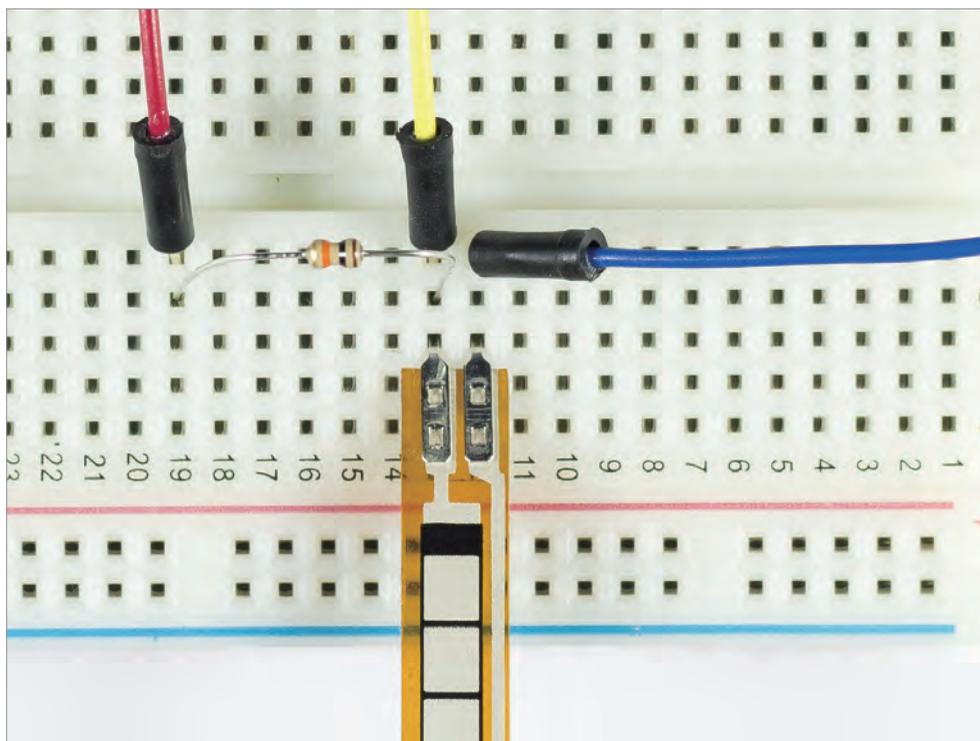


圖 2.3 近看麵包板上的線路

在電阻器的左端，線路連接到 FLORA 的 **3.3V**，電阻器不區分極性，所以其方向無關緊要，正插、反插都可以。

在電阻器的右端，線路之一接到彎曲感測器，第二條線路則接到 FLORA 板子的類比腳位；彎曲感測器另一端線路接地。在這份電路裡，電源將穿過電阻器、但會受到限制，其中部分將回到類比腳位，當我們動手折彎彎曲感測器時，電子穿過感測器的難易程度，會根據彎曲程度而改變，無法穿過感測器的過多電子將回到類比腳位，我們便能以 FLORA 板子讀取得知。

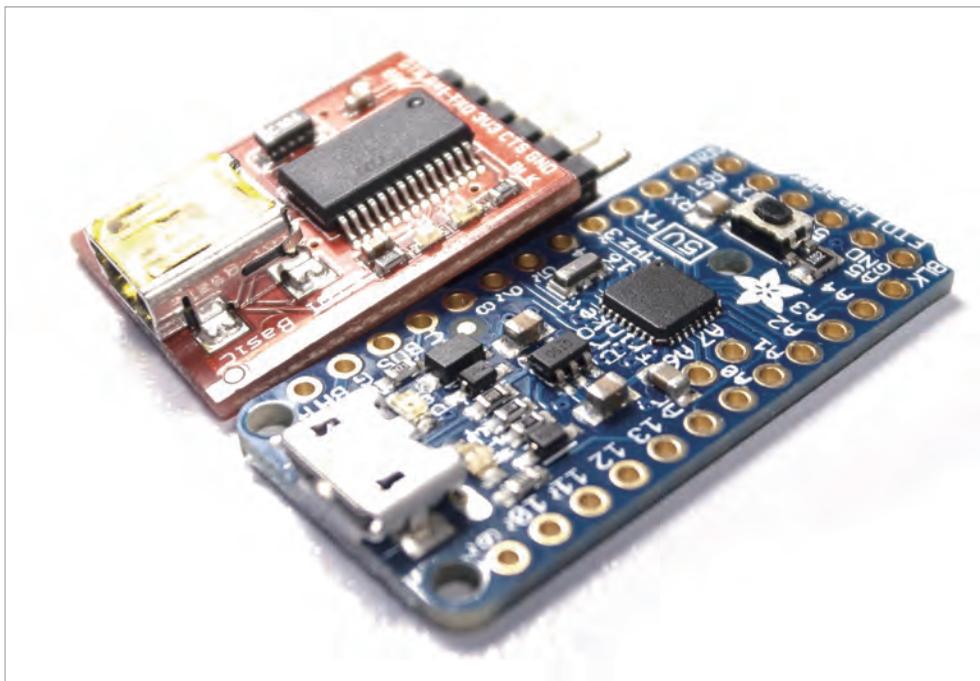


圖 4.8 FTDI 的序列介面轉 USB 轉換器與 Trinket 開發板

一般使用 Trinket 板子時，會焊接連接 FTDI 轉換器的排針，但我們想要盡量讓 Trinket 板保持平坦，也不想眼鏡內部放入尖銳的針腳，一不小心就可能發生意外戳傷眼睛。此處採用的作法是僅在 FTDI 端接上排針，上傳 Trinket 程式時就暫時拿著 FTDI 分線板，然而該條連線時不時可能接觸不良，致使上傳失敗，所幸這個問題不大，你只要確認針腳接觸良好，然後再次啟動上傳程序即可。

現在讓我們寫支草稿碼，檢查所有 LED 是否能正常亮起，為此，將以迴圈逐一迭代 LED，看看能否按預期運作。上傳 Trinket 程式的程序，就跟一般 Arduino Uno 相同，所以記得到選單裡選擇這塊板子，上傳底下的程式，檢查眼鏡正面的 LED。

到目前為止所介紹的專案，各位讀者是否還喜歡？現在該是時候認真一點、製作較為嚴肅的電子專案。如果說我們能夠讓電子專案連接上繞著地球轉的衛星，是否會挑起你的創意神經呢？對我來說的確會，而那正是本章將要介紹的內容，目標是建構獨一無二的手錶，除了能顯示時間，還能告訴你確切的 GPS 位置，另有額外資訊，告知你正以多快的速度移動，以及所在地的海拔高度。**GPS** 是 **Global Positioning System**（**全球定位系統**）的縮寫，於 1970 年代開始發展，但直到 1992 年才完成，現今共有 27 個衛星繞著地球轉，任何人若擁有一台 GPS 接收器，就有能力根據衛星發出的資訊、算出身處位置的經緯度座標。現代人周遭的許多電子裝置，早已裝上 GPS，包括汽車、飛機、船隻。很快地，你的手錶也會擁有 GPS 定位能力。

為了能夠顯示文字資訊，需要使用 LED 螢幕，LED 螢幕的形狀與尺寸形形色色，種類繁多，本章將會選用最小型的那種，學習如何從 GPS 接收器接收資訊，然後顯示在螢幕上頭，另外還會看看如何製作手錶的外殼，本章專案製作出來的手錶，其體積比起一般的手錶，稍微大上一點，但既然這是你自己動手製作的 GPS 手錶，我想讀者應該會想要讓它引人注目才對。圖 5.1 秀出本章專案需要的零件模組。

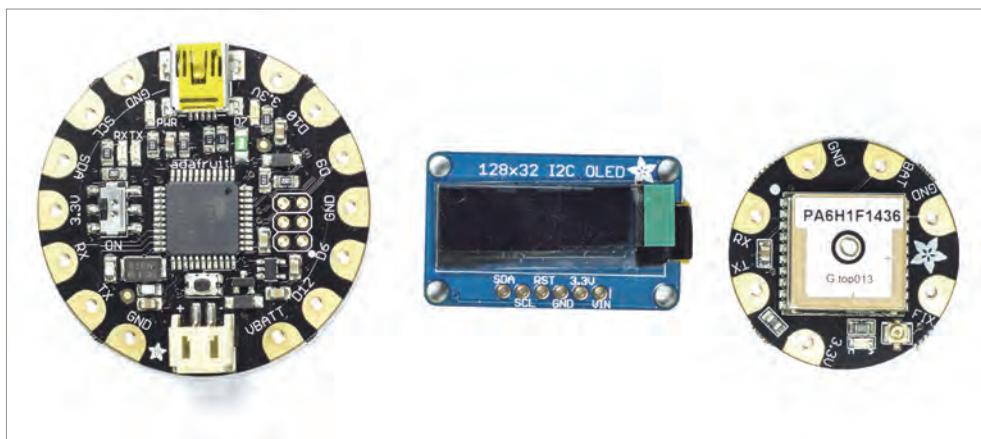


圖 5.1 FLORA 開發板、OLED、GPS 模組

連接馬達

為了能夠開啟真正的門，我們需要在門內部提供某種機制。伺服馬達（**servo motor**）非常普遍，常見於機器人專案，操控方式非常簡單。伺服馬達可分為兩種，標準型轉動範圍限於 180 度，連續轉動型伺服馬達則可朝兩個方向隨意轉動，後者的缺點是較不精準。此章專案將使用標準型伺服馬達，因為只需要旋轉 45 度即可，而且標準型伺服馬達通常較為精準。同樣的，該使用哪一種取決於你想操控的門，可沒有通用之門這回事，讀者大概需要根據眼前實際的門，稍稍修改本章專案的設計。下一節，將會詳細檢視我用於本章專案的門，但首先讓我們趕緊來看看如何操控伺服馬達。

因為在 Arduino 社群之中，伺服馬達非常熱門且其運用相當廣泛，所以 Arduino IDE 早已內建操控伺服馬達的程式庫。首先要連接馬達的線路，伺服馬達一定有 3 條線，線路顏色可能是紅、黑、白，也可能是紅、黑、黃。紅線應連接到 **5V**，黑線應接到 **GND**，黃線或白線則是訊號線路，應連接到 Arduino 的數位腳位，本章專案連接到腳位 **9**。圖 6.4 秀出伺服馬達連接 Arduino Uno 開發板後的模樣。

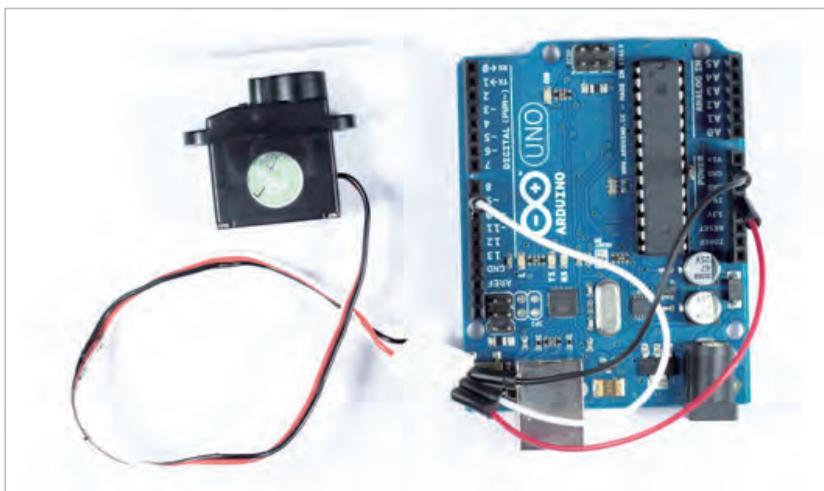


圖 6.4 伺服馬達連接 Arduino Uno 板

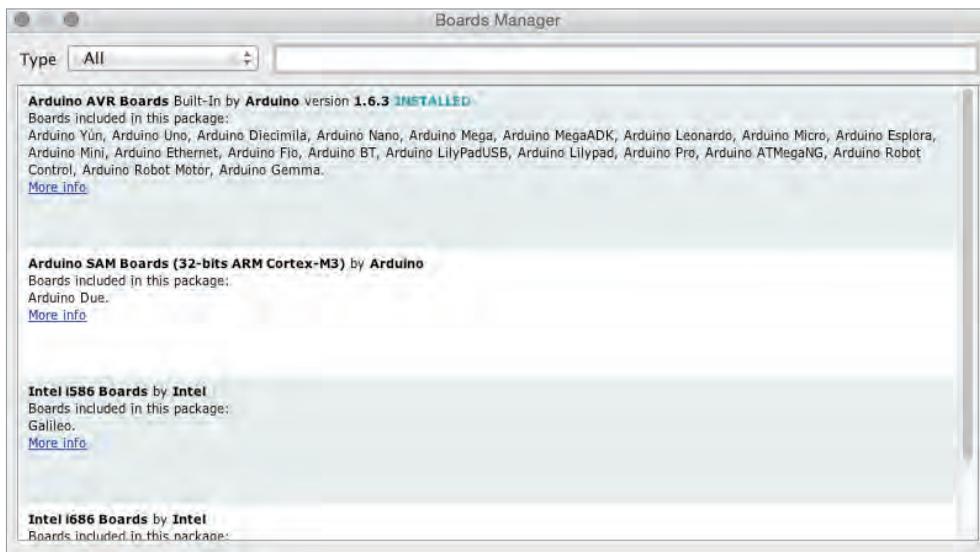


圖 7.1 板子管理員

鍵入「Blend Micro」搜尋，然後按下「安裝」按鈕即可，支援 Blend Micro 板子的所需檔案，就會自動開始安裝。若你使用舊版的 Arduino IDE，則請依循底下步驟，手動修改 IDE 的設定：

1. 首先，請到 Red Bear 的 GitHub 頁面，從底下網址下載 ZIP 壓縮檔：<https://github.com/RedBearLab/Blend>。
2. 解壓縮後，找出 hardware 資料夾，位於 Blend-**** 的 Arduino 裡。
3. 複製該資料夾，放到 Arduino IDE 安裝路徑的文件資料夾的 Arduino 資料夾裡，把 hardware 資料夾複製到 Documents 的 Arduino 子資料夾裡，應該和你的程式庫資料夾位於同一層。

接下來，需要修改位於 Arduino IDE 應用軟體資料夾裡的 main.cpp 檔，才能讓 Blend Micro 板出現在 IDE 裡頭，請注意，與剛剛的 Documents 的 Arduino 子資料夾不同，此處我們想要的資料夾位於 Arduino 應用軟體的資料夾裡。

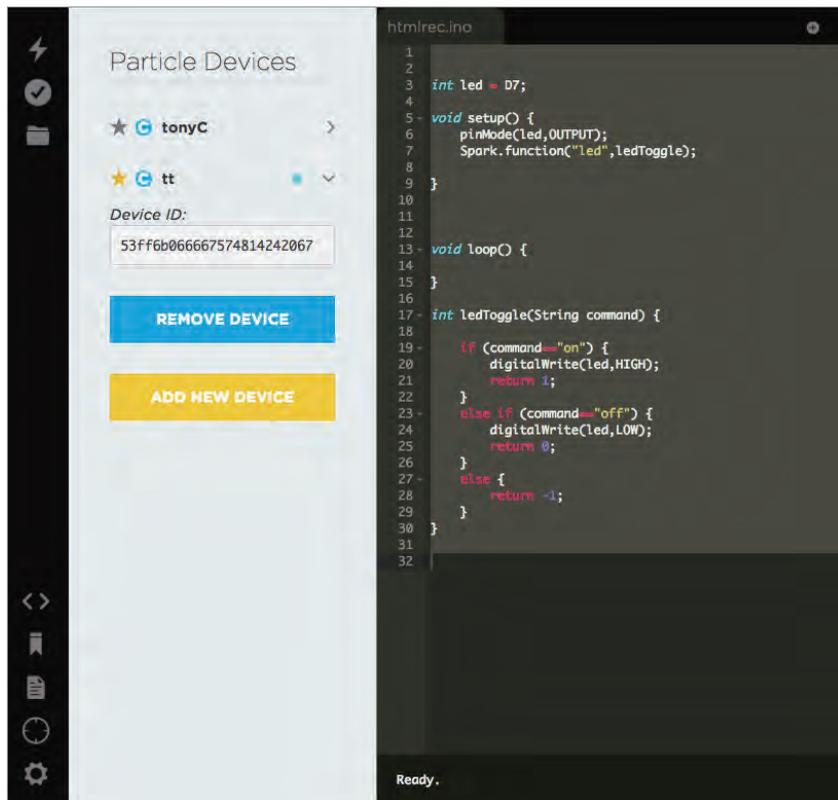


圖 8.7 裝置選單

之後將需要這個裝置 ID，請記下。第二部分是**存取標記 (Access Token)**，如果不知道的話，他人幾乎不可能連接到你的板子。標記是經由網路溝通、進行不同事務時，諸如登入工作階段、使用者認證、或是賦予權限，所需傳輸資料的一部分，可以把它想像成存取 Core 板、傳輸時需要的存取憑證；首先需要裝置 ID 找到板子、然後需要存取標記才能連接該裝置。可到網頁介面 IDE 的設定選單裡，找到**存取標記**，範例請見圖 8.8。

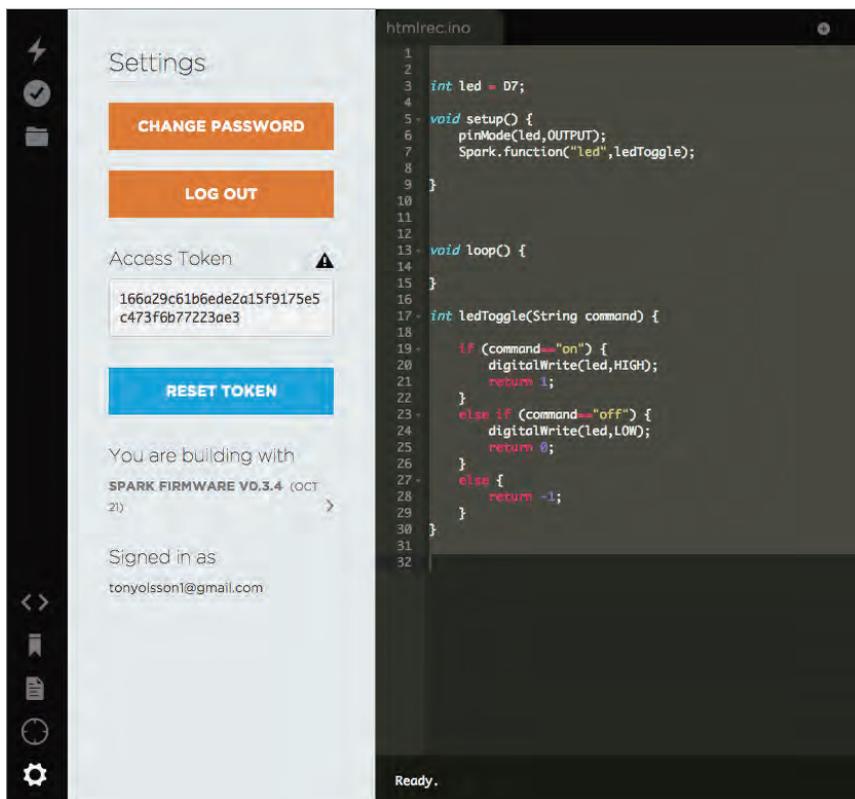


圖 8.8 設定選單，在此可找到存取標記

若有安全考量，可隨時重置存取標記，但在重置之前，請確認沒有任何重要的東西連接到你的 Core 板子，因為一旦產生新的存取標記，就必須全面更新。

現在，讓我們來看看雲端函式，底下這份程式範例實作了此種函式，當雲端函式被呼叫時，就會去觸發內部函式。

```
int led = D7;

void setup() {
  pinMode(led,OUTPUT);
  // 宣告雲端函式 led，它會觸發 ledOnOff 函式
```



圖 8.13 為配方建立觸發器

有些讀者應已明白，你可以讓 Core 韌體含有多個監看函式，就會出現在變數清單裡，然後便可選擇作為 IFTTT 的動作。在我們的配方設定中，監看 Core 板子的類比數值，若超過 500，就做點事情。現在請按下 **Create trigger (建立觸發器)**，讓我們開始設定動作的部分。

也會出現同樣的選項，但這次是為了設定想執行的動作，當類比值超過 500 時就執行。請按照底下步驟：

1. 尋找臉書 (Facebook) 通道並選擇。
2. 第一次使用該通道時，需要填入該服務的帳號密碼。我假定各位讀者大都擁有臉書帳號 (或許是我太過樂觀)，若你沒有的話，可選擇任何其他服務之一，因為底下的步驟大都相同。
3. 連接後，將會顯示你的帳號與可用的功能，有 **Create a status message (建立狀態訊息)**、**Create a link post (建立連結發文)**、**Upload a photo from URL (以網址上傳照片)**，請選擇 **Create a status message (建立狀態訊息)**。

沒有一種方式可稱得上最佳作法，所以我會建議讀者觀看影片學習，越多越好，方能掌握其中的技巧與概念，然後到周圍翻找，搜出不要的電子垃圾，開始練習解焊。底下詳述兩種作法：

- ✧ 使用烙鐵加熱固定零件的焊錫，把整個電路板拿起來、狠狠地摔到桌上，若焊錫已被加熱，當零件撞擊桌子時，焊錫應會掉落到桌上。使用此技巧時必須非常小心，你可不想讓熱焊錫飛濺到手上或其他會燃燒的物體表面。絕對不要使用非工作用的桌子，因為熱焊錫會在桌面留下痕跡。
- ✧ 另一種方式是加熱焊錫，以鑷子或斜嘴鉗夾住零件並拔起，同樣的，拿烙鐵加熱固定零件的焊錫，然後以另一隻手拿鑷子夾起零件。注意，加熱時間不可過長，否則零件可能損壞，僅該加熱數秒鐘，然後就試著拔起零件，若拔不起來，再試一次之前，請讓它冷卻數秒。注意鑷子的加熱狀況，可能會變得非常燙。

當你覺得掌握解焊的技巧後，便可開始移除電池充電模組的 JST 端子，此處假定你選購與我類似的充電器，以及從 Particle Core 板子移除排針。若一切工作按照計畫進行，結果應如圖 9.7 所示。

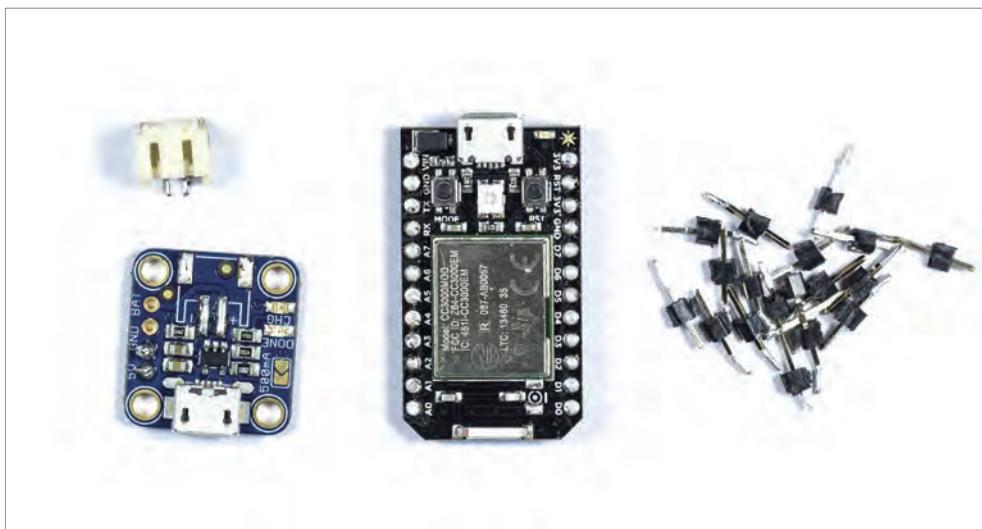


圖 9.7 解焊後的 JST 端子和公排針