

1

Web 工程師的 物聯網

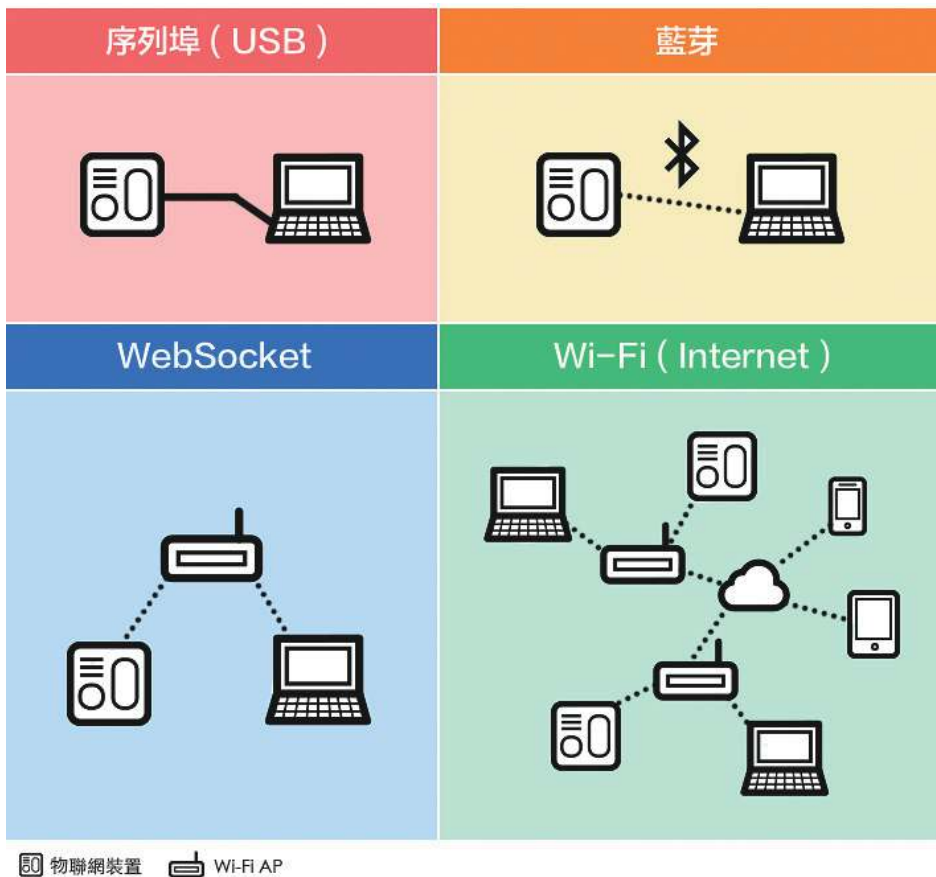
第一次聽到「物聯網」這三個字，是在好幾年前的研討會，當時還覺得是哪個地方的科技用語，沒想到在幾年後的今天，物聯網竟然是如此的蓬勃發展，成為大眾耳熟能詳的詞彙。

隨著網路技術的蓬勃發展，各種框架和語言不斷推陳出新，網頁工程師也搖身一變成為最夯的職業，也因此物聯網慢慢的不再只是傳統「Maker」的專業，透過各式各樣的網頁技術和雲端平台的輔助，慢慢地將物聯網推向了一個充滿想像與創意的藍海空間。



↑ (圖片修改自：<http://www.arreverie.com/iot.html>)

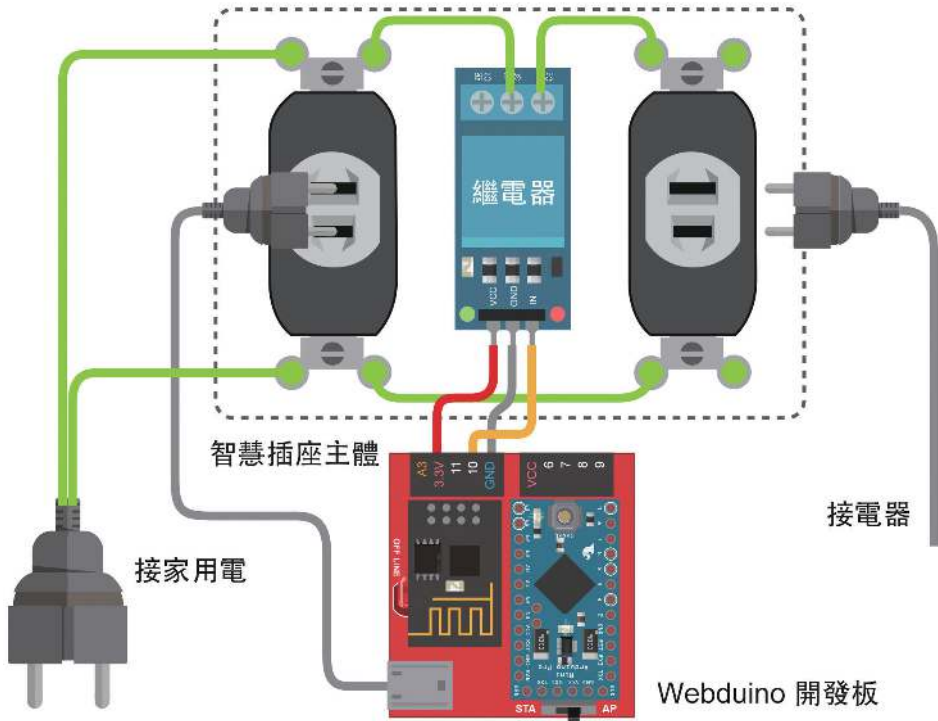
這四種方式各有好處，例如要在台灣控制美國的裝置，就必須使用 Wi-Fi 控制，如果想要控制「手邊的」多台裝置，就可以考慮 WebSocket 和 Bluetooth 連線，如果只想單純接 USB 線控制，更可以採用 Serial Port 的做法。



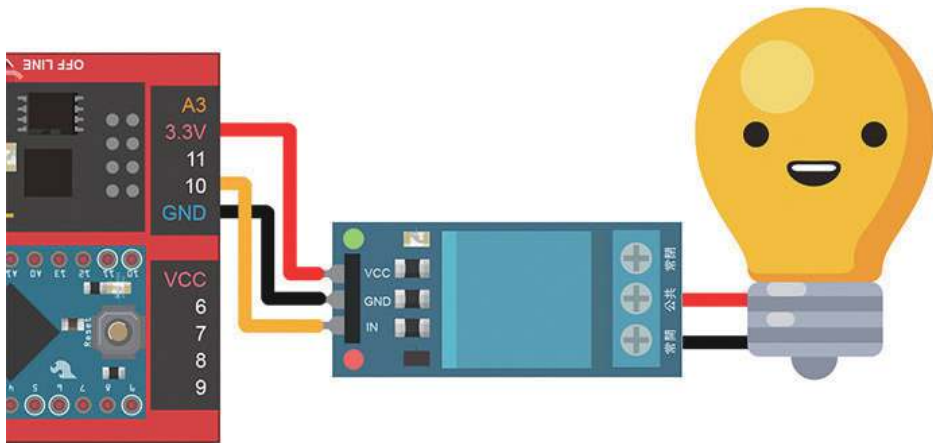
使用 Web 開發的優缺點

❖ 優點

- ◆ 跨平台（連 APP 都可以用網頁開發）
- ◆ 跨地區（只要有網路，台灣也可以控制美國的東西）
- ◆ 一個網頁搞定（不用額外多學其它程式語言）



如果覺得接線圖太複雜，原理其實就像下面這張圖，只是因為要接家用電，所以必須使用插座與插頭。



← → ↻ 🏠 🔒 安全 https://webduino.io/device/

Device Status

██████: OK

出現 OK 表示成功上線

如果想在開發階段即時知道裝置狀態，在編輯工具的右上角有個「閃電符號」的按鈕，點選後會展開輸入欄位，輸入開發板的 ID，如果開發板成功連上 Wi-Fi，就會變成綠色的閃電，如果開發板斷線或是沒有上線，則會是紅色閃電。



Webduino Blockly 操作

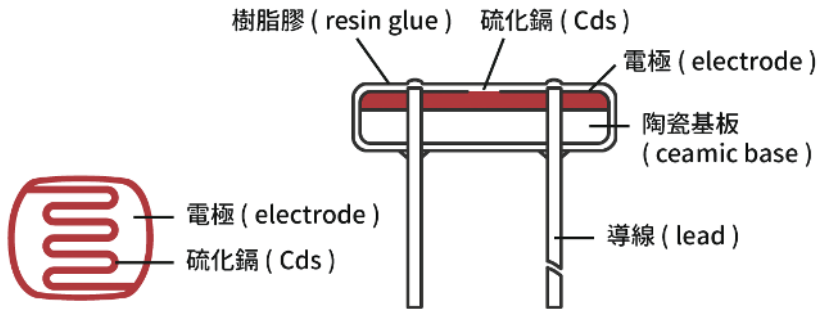
確定裝置都上線之後，從編輯工具的右上角可以打開範例，快速的選擇第一個範例「LED」，就可以帶入第一組範例程式。



◆ 光敏電阻的長相

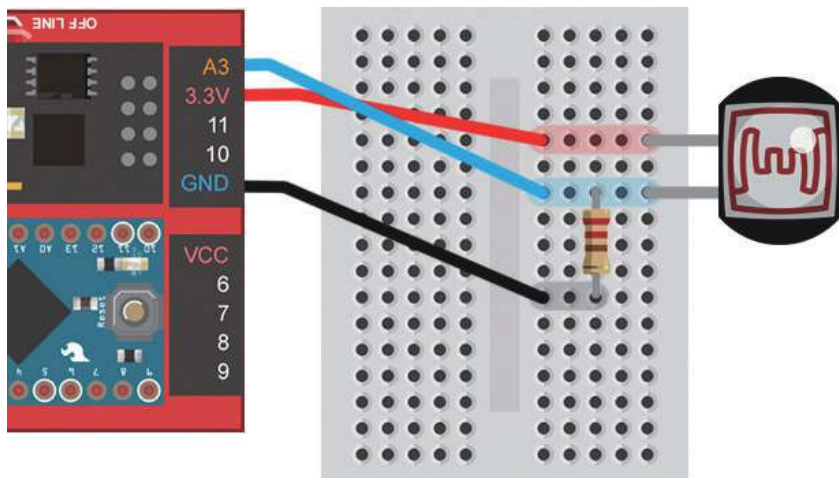


◆ 光敏電阻的結構



Webduino Blockly 操作

由於每個光敏的電阻值範圍不同，有些甚至會趨近於 0，因此在接線上必須要用到一個「普通電阻」在線路裡頭，作為避免短路的「保護」作用，而光敏電阻有兩支腳，不管哪一支都可以作為訊號腳，訊號腳會接在類比的位置（A 開頭的腳位），而另外一支就是 VCC 或 GND，這裡我們把另外一支腳接在 3.3V（VCC），訊號腳另外接一個電阻連到 GND，因為要用到同一個 GND 腳位，所以必須得用麵包板來輔助（VCC 和 GND 顛倒過來也可以，但數值也會跟著顛倒過來，如果要看專有名詞，可以查詢「上拉電阻」或「下拉電阻」）。

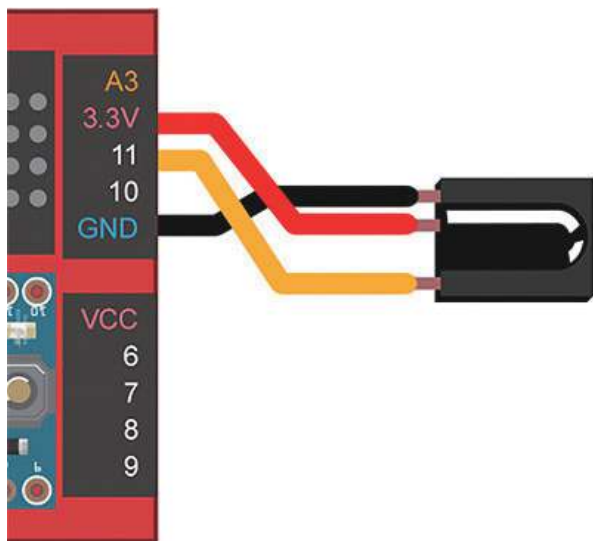


接線完成之後，先透過 Webduino Blockly 編輯器 (<https://blockly.webduino.io>) 來測試，因為會用網頁「顯示文字」來顯示光敏電阻數值，所以要先點選右上方「網頁互動測試」的按鈕，打開內嵌測試的網頁，用下拉選單選擇「顯示文字」。



把開發板放到編輯畫面裡，填入對應的 Webduino 開發板名稱，開發板內放入光敏電阻積木，名稱設定為 photocell，腳位設定為 A3。





接線完成後，先透過 Webduino Blockly (<https://blockly.webduino.io>) 來測試一下，打開內嵌的網頁，選擇「顯示文字」，放入開發板的積木，輸入裝置的 device ID，接著放入紅外線接收的積木，腳位設定為 10 號腳，再放入「開始偵測」，如果偵測到訊號，就透過「顯示文字」顯示出代碼（其實和人體紅外線偵測幾乎一模一樣）。



點選「執行」，用電視機遙控器對著接收器點按，就會看到按鍵的訊號顯示出來了，這裡先記下幾組訊號，作為待會要控制燈泡使用，這邊我記下來的號碼是按鍵 1「20df8877」，按鍵 2「20df48b7」，按鍵 3「20dfc837」（因有時候紅外線會因為距離被干擾，建議對著接收器多點幾次，才能確定是真正的訊號喔）。



坦白說，這其實還蠻多可以玩的，例如，用電視機遙控器操控 Youtube、用電視機遙控器切換網頁、用電視機遙控器玩 Facebook？總而言之，當我們可以接收訊號之後，就可以在網頁裡頭盡情的應用這些訊號！



Webduino 程式碼解析

在 JavaScript 的使用方法裡，紅外線接收的宣告使用 `getIRRecv(board, pin)`，`pin` 是數位腳，例如接在 11 號腳就改為 11。

```
var irrecv = getIRRecv(board, 11);
```

接收訊號和「光敏電阻」的用法很像，使用 `.on` 以及 `callback` 實作，`callback` 帶一個參數，這個參數就是接收到的紅外線訊號。

```
irrecv.on(function(val){  
  });
```

同樣的也可以透過 `.off()` 來關閉。

```
irrecv.off();
```

如果是紅外線發射，就是 `irled = getIRLed(board)`；為什麼沒有 `pin` 腳呢？因為 Arduino 的發射振盪器只有 9 號腳有，因此，一定要接在這支腳才能發射紅外線訊號。

```
irled = getIRLed(board);
```

發射訊號的方式：

```
irled.send("紅外線代碼");
```


11

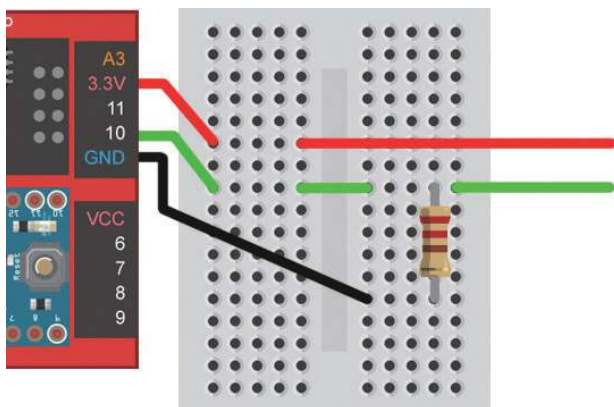
打造電流急急棒，碰三次就開燈！（開關應用）

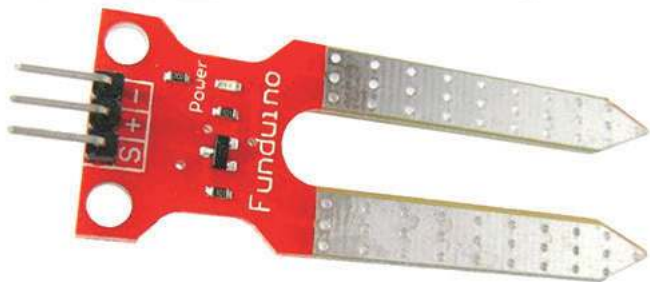
上一章我們認識了「開關」，並透過按壓開關來關燈，但這樣子的做法和日常生活中的開關實在沒兩樣，所以我們可以透過同樣的原理，實際做出一個簡單的電流急急棒，或是直接把現成的電流急急棒玩具拿來改造，變成和智慧插座結合的智慧玩具囉！



認識電流急急棒

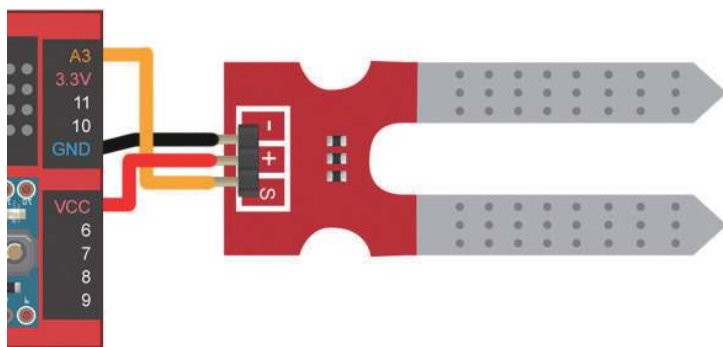
首先看一下電流急急棒的原理，其實就是把「訊號腳」和「正極」（VCC 或 3.3V）的延伸出來，當「兩者接觸的時候，就等同於按下開關」。





Webduino Blockly 操作

因為土壤濕度偵測計回傳的是類比訊號，所以接線的方法要把 S 的腳位接在類比腳 (A 開頭)，+ 接 VCC，- 接 GND。



這裡先打開 Webduino Blockly 編輯工具測試 (<https://blockly.webduino.io>)，放入開發板，輸入手邊裝置的 ID，放入土壤濕度偵測的積木，設定開始偵測並顯示偵測數值。



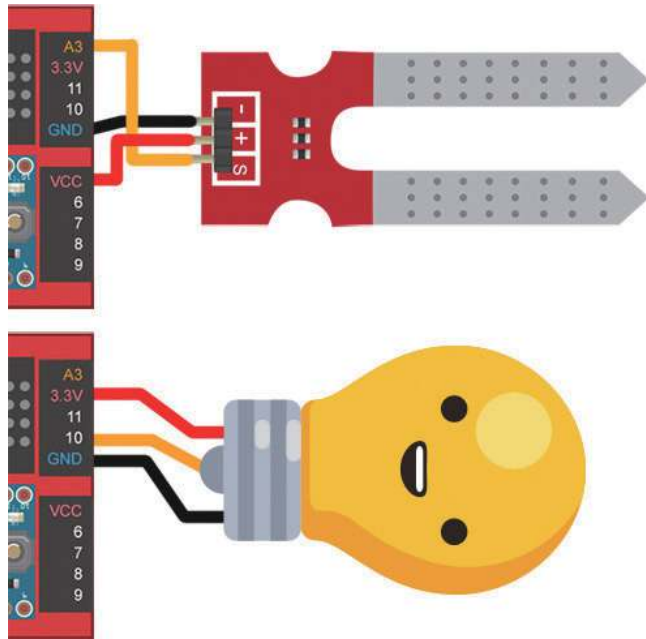
如果要停止偵測，使用「.off()」即可。

```
soil.off();
```



網頁操控

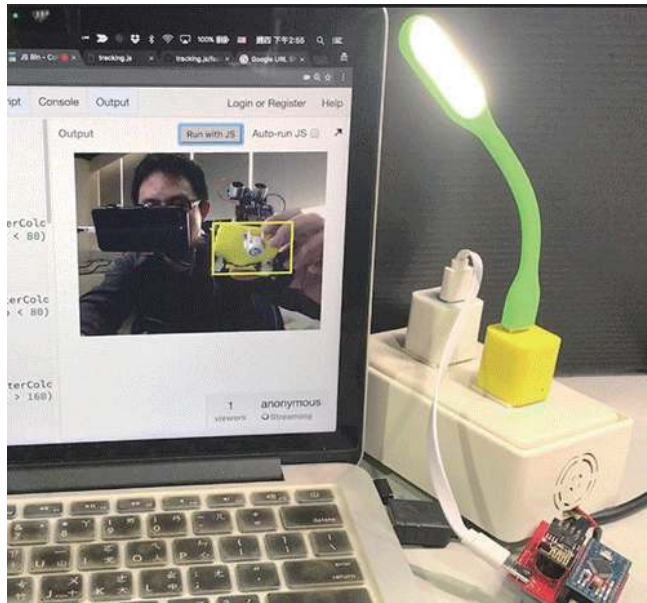
開始控制之前要先接線，一塊裝置開發板接土壤濕度偵測計（腳位 A3），另一塊裝置接智慧插座（腳位 10），其實不一定要這樣做，用同一塊開發板也可以完成，只是用兩塊更能凸顯「Wi-Fi」操控的能力，對於土壤偵測來說也比較合理，畢竟花盆擺在陽台上，燈泡在室內是比較正常的狀況。



HTML 的部分先在自己的網頁內引入「webduino-min.js」還有「webduino-blockly.js」這兩個 JavaScript，這樣我們才有操控開發板以及土壤偵測的 API 可以使用，在 body 的區域放入一個 h2 來顯示一些文字訊息，然後放入兩張燈泡一明一暗的圖片，當偵測到訊號的時候，網頁上的燈泡也會發生反應。

範例程式：<https://goo.gl/LBFJR2>

實際完成的畫面（動畫圖片網址：<https://goo.gl/opiZ3F>）



透過人臉追蹤點亮燈泡

至於人臉追蹤，這邊使用人臉的 x 數值來追蹤，預設追蹤到的 x, y 數值是對應到 video 的左上角，這也是為什麼我們可以拿來作為 CSS 位置的主要原因，因為用的 video 寬度為 360，所以讓 x 數值大於 $(180 - \text{追蹤框寬度} / 2)$ 的時候就亮燈，也就是臉往右邊移動超過中線的時候會開燈。

```
$(function(){
  var track = {};
  var $demo = $('#demo');
  var led;

  track.myTracker = new tracking.ObjectTracker("face");
  track.myTracker.setInitialScale(4);
  track.myTracker.setStepSize(0.5);
  track.myTracker.setEdgesDensity(0.1);
```