

# 推薦序 I

preface

## 點燃心中的創作魂

傳統的 Arduino 要做到裝置互聯並非易事，尤其是要走出區域網路與多人共享資料時，選項只剩下 Wi-Fi。因此，這兩三年來搭配 Wi-Fi 的開發板不斷推陳出新，包含 Intel、Samsung 與台灣聯發科技等公司都推出了具備 Wi-Fi 功能的開發板，就更不用說以 Raspberry Pi 為首的 Linux 單板電腦大軍了。有了板子以後，就要思考：「連上網之後要做什麼呢？」，也許是控制家電，也許是資料收集，或許作一台可愛的小機器人也不錯，本書都幫您準備好了喔！

本書架構清楚易懂，並兼顧初學者與進階玩家的需求。將 Webduino 開發板連上無線網路之後，即可整合各式常見的周邊電子元件來製作各式專題。另一方面，Webduino 有圖形化開發環境，對於初學者而言，可以先從理解系統執行流程開始，很容易就能完成各式簡易的小專案，也適合學校作為相關教材。如果您對於文字式程式語言有一定基礎的話，Webduino 的線上編輯環境也提供您使用 JavaScript 自行開發程式指令。對於有進階運算需求的人來說，真的是非常棒的功能。

書中的每個專題都提供了圖形化指令與 JavaScript 語法的對照，相關領域的老師們可以將其視為入門與進階的兩階段教學，讓有興趣的學生可以順利銜接互動式網頁或手機 app 開發。

在物聯網的世界裡，有人是從硬體連上網路，也有人是從網路擁抱硬體。誠摯推薦本書給各位讀者，期待您從本書中找到點燃心中創作魂的那把火。

祝 展卷愉快

曾吉弘  
CAVEDU 教育團隊



# 推薦序 II

preface

## 重拾數位創作的感動！

智慧家庭、IoT 物聯網、工業 4.0 之類的話題有多夯，未來潛力有多大等等的話題我就不贅述了。因為讀者只要稍微注意一下報章雜誌或數位媒體，隨便都可以找到一堆。

然後，市面上什麼 Arduino-like、Linked It、Raspberry Pi，還是其他技術解決方案的，只要稍微翻找一下資訊相關書籍或網站，一樣也是琳瑯滿目地要獨佔你的目光。

但是，你是否曾經想過，甚至實際做過 .....

用自己寫的程式來點亮一顆 LED 燈嗎？

沒有！？ ...

因為 .....

光是電子電路和軟體程式技術的艱澀，就足以把一個原本充滿創作熱情的人，活生生地給搞到完全心灰意冷！

以我本身為電子科班出身，卻一直從事軟體開發、網站和資訊教育的技術阿宅來說。雖然常常會有各種創新構想，而且也具備這些技術的相關基礎能力，但之前卻都一直遲遲沒有動手去進行任何嘗試 ....

..... 即便只是去點亮一顆 LED 燈 ....

然而，就在 2015 年三月的某一天，Webduino 開發團隊的大隊長 Marty（執行長 許益祥 先生）跟我展示出 Webduino（網頁技術）開發方案之後。

我相當震驚！！.....

原來一切可以這麼容易！

要知道，當你發現自己可以直接使用早已熟悉的能力，去做一件原本覺得很繁瑣或困難的事情時，內心會有多大的震撼！？

我想，大概就好比原本只會在地上走的人，突然發現自己可以自由自在地在天上飛那樣的震撼！.. XD

是的，感謝 Webduino 的出現。讓我覺得自己能夠再次「重拾數位創作的感動」！～

現在，Webduino 變得更容易了！甚至原本完全不會程式的人，都可以透過 Blockly 這種積木圖像開發方式來實現自己的構想和原型。

這對於「非」資訊或電子相關本科系的人來說，真的是一大鼓舞和希望啊！～

這次 Webduino 專書的推出，相信會有更多跟我一樣，對於數位藝術創作、或是開發虛實互動整合有強烈熱情和創意的人，都能夠在學習過程中，發現更多的可能性，並且更自由自在地去實現你心中的那個創意或構想！～

最後，預祝大家學習順利，並用您的創作一起來創造出更多的感動！

羅友志（DOFI）

KIMU 高雄獨立遊戲開發者聚會創辦人之一

Webduino 傳教士暨特約講師

30 餘本專業電腦書籍譯者暨資訊講師

## 2.1

## 什麼是 Webduino ?

## ◎ 緣起

Webduino 名稱的由來，就是 Web 和 Arduino 這兩個單字的組合，雖然 Webduino 這個詞早在 2012 年就已經有人提出，但仍然停留在「能夠讓 Arduino 上網」的階段，而本書所提及之 Webduino，除了可以讓 Arduino 上網，更可以「雲端更新韌體程式」、「用各種程式語言開發」以及「使用 Wifi 控制」，真正實現了 Web 與 Arduino 的完美結合，不再只需要 C/C++ 才能進行開發，完美的實現了物聯網的創意與想法。

## ◎ Webduino 與 Arduino 的差異

許多人都會誤會 Webduino 是要搶佔 Arduino 的市場，但實際上 Webduino 賦予了 Arduino 原本沒有的功能：「用 Web 開發」及「Wifi 控制」簡化了 Arduino 的開發過程，為原本 Arduino 的開發者縮短開發時程、降低開發門檻，創造一個全新的體驗和開發模式，更讓許多原本不懂該領域的開發者、設計師或學生們，可以更輕鬆愜意的跨入物聯網的領域，更方便快速的學習研究。

	Arduino	Webduino
開發語言	C / C++	HTML / JavaScript
開發環境	Arduino IDE	瀏覽器 / ...
連接方式	USB	WIFI
更新程式	連接燒錄	立即更新

## 3.1

## HTML 簡介

透過瀏覽器瀏覽網頁時看到的是文字，網頁上大部分的文字是格式化過的文字，而非純文字。現今的網頁設計師能使用數百種不同的字體、大小、顏色甚至各種非拉丁字元，瀏覽器大多能正確呈現。網頁還包含圖片、影片或背景音樂。有時會有下拉選單、搜索框或連結，讓你能進入其他頁面。有些網站甚至能讓用戶按照自己的偏好，調整頁面的顯示方式，以符合個人所需（例如克服視覺障礙、聽障或是色弱等）。網頁通常包含一個可隨畫面捲動的內容區塊，而頁面的其餘部分則保持靜態。



↑ Webduino 網站 <https://webduino.io/tutorials.html>

HTML 是 Hyper Text Markup Language（超文字標記語言）的縮寫，一種標記語言。所謂的標記語言，是一種使用特定語法的語言，告訴瀏覽器該如何顯示一個網頁。HTML 使用一組預先定義好的元素來定義內容的類型，元素由一或多個「標籤」（HTML tag）組成，包含或表示某種內容。標籤通常由「小於符號」起始，而以「大於符號」作結束，並會成對出現，例如：`<p></p>`、`<b></b>`。

## 4.1

## CSS 簡介

CSS 是層疊樣式表（Cascading Style Sheets）的縮寫，是一種用來為結構化文件（如 HTML）添加樣式（字型、間距和顏色等）的電腦語言，由 W3C 定義和維護。目前已經到了 CSS3，CSS3 現在也已被大部分瀏覽器所支援，CSS 定義了如何顯示 HTML 元素，就像 HTML 直接利用字體標籤和顏色屬性所發揮的作用那樣（建議使用 CSS 而不要由 HTML 去做設定）。樣式除了可以寫在 HTML 的 style 內，也會保存在外部的 .css 檔中，透過外部的 CSS 文件引入，就可以去改變網站中的佈局和外觀樣貌。

由於允許同時控制網頁內多個元素，也可讓多個網頁共用同一個 CSS，我們能夠為每個 HTML 元素定義樣式，並將之應用於多的頁面中。如需進行全域的更新，只需簡單地改變樣式，然後網站中的所有元素均會自動地更新。

## 4.2

## CSS 語法介紹

CSS 的寫法由選擇器與屬性組成，選擇器後方會接上大括號「{}」，大括號內是屬性名稱與對應的值，各個屬性名稱由分號「;」區隔，整體結構如下圖：



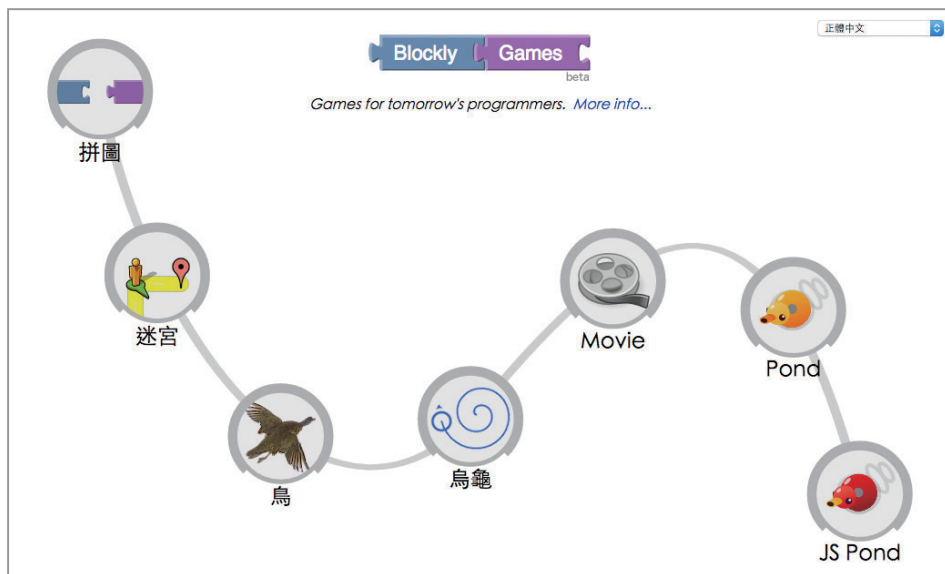
以下面這個例子而言，<p></p> 標籤內的文字顏色是黑色，<h1></h1> 內的文字是紅色，<span></span> 內的文字是綠色並且加粗體。



## HTML 程式碼

```
<h1>我是 h1</h1>
```

```
<p>我是 p <span>我是 p 裡面的 span</span></p>
```



## 5.3

## 用 Blockly 玩轉 JavaScript

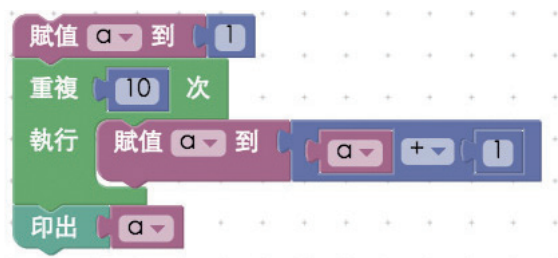
為了加速學習 JavaScript 的過程，我們使用 Blockly 來快速產生 JavaScript 的程式碼，以下將使用 Webduino Blockly 線上編輯工具，透過程式積木的組裝，降低學習程式設計的門檻（Webduino Blockly 學習工具：<http://blockly.webduino.io/>）。

### ◎ 變量（變數）

「變量」（Variables）的定義，是在程式裡最基本的語法，我們可以賦予變量一個名稱，並用變量來儲存文字、數字、陣列、函式或物件 ... 等內容。由於 JavaScript 的語言特型，我們可以隨時將變量的型別進行轉換，例如儲存文字的變量型別為「字串」，我們如果今天儲存數字進去，或強迫讓其轉換為數字格式，該變量的型別就會變成「數字」格式。

在 Blockly 的積木方塊歸類在「變量」（或 Variables），點選後就會出現變數的積木方塊，將第一個後方有缺口的方塊拖拉到中間的畫面裡（第二個方塊象徵我們可以使用這個變量）。

接著，設定一個變數  $a$  的值為 1，然後讓每執行一次迴圈，就會自動把  $a$  加 1，然後迴圈完成之後就會自動印出  $a$  的數值。



由於 JavaScript 程式語言的特性，程式執行一定會按照順序執行，因此「印出」的指令一定會在「迴圈結束」之後才會執行，所以最後的結果是 11。



如果我們看程式碼，就會發現迴圈是放在一個 `for` 裡頭，這也是常見的「for 迴圈」，`for` 的後面接了一個小括號，裏頭根據分號分成三個數學式，代表每次執行 `count` 都會加 1，從 0 加到 9 共加了十次為止。

```
var a;

a = 1;
for (var count = 0; count < 10; count++) {
    a = a + 1;
}
window.alert(a);
```



實際照片：



編輯方式先把開發板放到畫面中，把超音波放到開發板內，變數預設命名為 ultrasonic，Trig 接在 11、Echo 接在 10。



接著放入「擷取距離」的積木，設定每 500 毫秒（0.5 秒）擷取一次，並將截取到的數值顯示在畫面右邊的網頁中。



確認 Webduino 開發板上線之後，點選執行按鈕，用手或平整的遮蔽物放在超音波傳感器前方，就可以看到數值的變化。

## 11.1

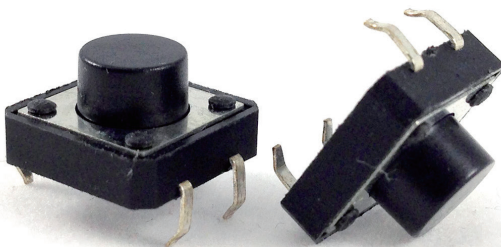
## 認識按鈕開關



練習網址

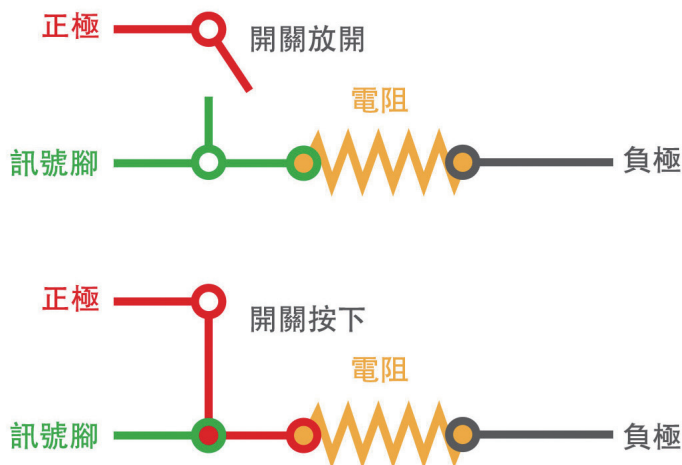
<http://blockly.webduino.io/?page=tutorials/button-1>

按鈕開關就是具備了「點壓」功能的開關，壓下去的時候「開」，放開的時候「關」，按鈕開關有四支腳，兩兩成對，成對的一邊互不通電，當按壓按鈕時就會通電。



因為是開關，所以我們先來認識一下「開關的原理」，開關的目的是讓正極與負極中間可成「通路」或成「斷路」（如圖所示）。當通路的時候，也就是開關打開，與負極在同一側的訊號腳會收到不同電位差的訊號，可知開關打開；當斷路也就是開關關閉時，訊號腳會恢復原本的電位差，也就知道開關關閉。

開關原理圖：



```
});
button.on("longPress",function(){
  console.log("longPress");
  document.getElementById("show").innerHTML = "長按";

});
});
```

## 11.2

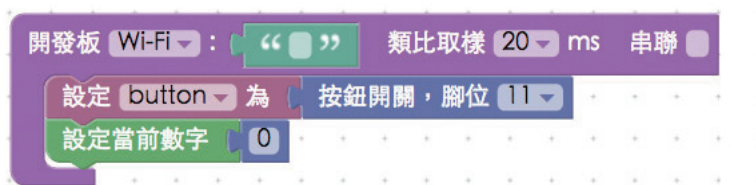
## 點擊按鈕開關增加數字



練習網址

<http://blockly.webduino.io/?page=tutorials/button-2>

能夠改變文字還不夠，接下來要利用點擊按鈕開關，改成增加數字或讓數字歸零，實作的方式很簡單，只要我們在剛才「按下」的動作發生時，把數字加 1，在「長按」的動作發生時，把數字歸零。一開始先把開發板積木放到畫面中，擺上名為 button 的按鈕積木，腳位設定為 11，並且把現在右邊網頁的數字設為 0，為什麼要先把當前數字設為 0 呢？因為我們要把數字往上加，總得有個最初的數字基準，不然就會變成  $? + 1$  等於？的情形了。



再來就是放入按鈕的行為，這裡只要擺入「按下」和「長按」即可，在按下的時候，讓顯示的數字等於當前的數字加 1，長按的時候，讓當前的數字歸零。

## 18.1

## 認識 Webduino 自走車

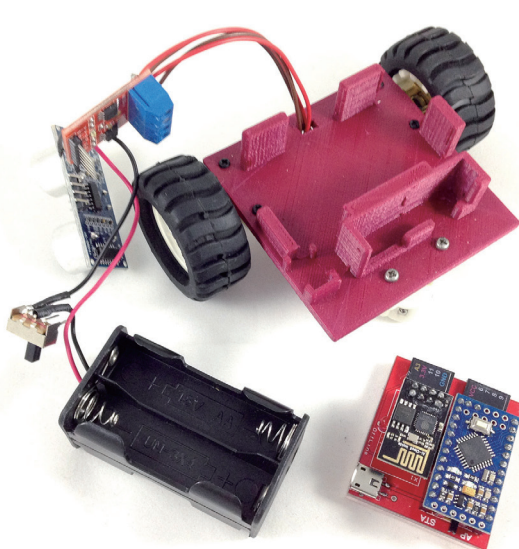


練習網址

直接使用 Webduino Blockly 線上編輯工具練習：

<http://blockly.webduino.io/>

首先要組裝自走車，自走車包含的組件如下：自走車底版（3D 列印）、左右各一顆輪子與馬達、開關、電池盒、馬達驅動板、超音波傳感器和 Webduino 開發板，由於主要的結構都已經焊接完成，組裝上變得相當簡單。



首先把電池放到電池盒內（需要四顆三號電池），然後將電池盒放到車子上。

