# 程式人 月刊 雜誌

# Programmer



讀書做善事、寫書做公益 – 歡迎程式人認養專欄或捐出您的網誌

參考價： NT 50 元，如果您喜歡本雜誌，請將書款捐贈公益團體

羅慧夫顱顏基金會 彰化銀行 (009) 帳號： 5234-01-41778-800

*012378*

愛心條碼

# 程式人雜誌

開放公益出版品

2013 年 2 月號

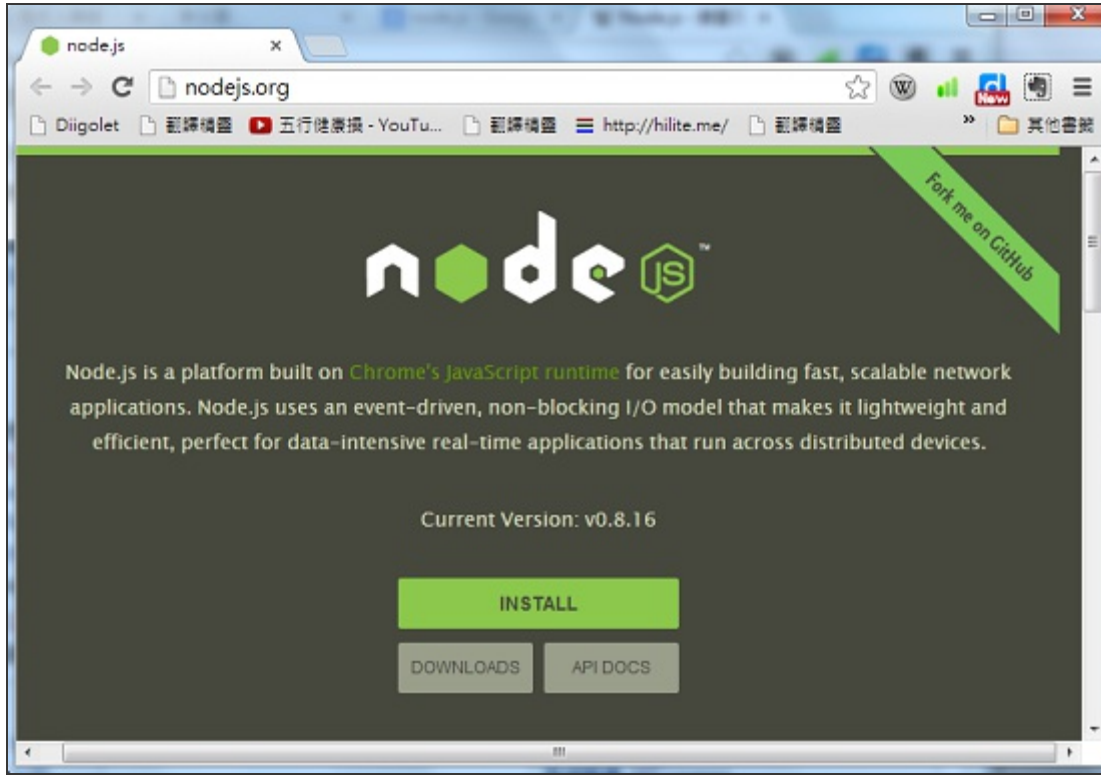
# 本期內容

- 參與編輯
- 公益資訊

# 程式人短訊

## 軟體短訊-Node.js

對許多程式人來說一直都是個大障礙，除了沒有好的開發環境之外，無法用 JavaScript 撰寫 Desktop 或 Server 端的程式也是個大問題，還好這些問題現在都有了解法。 Node.js 可以讓你在 Server 端開發 JavaScript 程式，就像 Ruby on Rail 或 PHP 一樣，但是卻有著全然不同的設計邏輯與想法。




node.js 官網畫面

透過 Node.js , 您可以很容易的測試 JavaScript 程式，並且在 Web 開發的前後端同時使用 node.js，而不需要在前後端分別使用不同語言，這點讓 node.js 特別有吸引力。 以下是一個 node.js 的 「HelloWorld !」 範例

程式，您可以看到這種寫法與 PHP 那類將程式嵌入在 HTML 當中的寫法很不相同，反而比較像用 Socket 撰寫網路程式的樣子，但又有些不同。

```
var http = require('http');

http.createServer(function (request, response) {
    response.writeHead(200, {'Content-Type': 'text/plain'});
    response.end('Hello World\n');
}).listen(8000);

console.log('Server running at http://127.0.0.1:8000/');
```

若您想學習 node.js 程式設計，網路上已經有一群熱心的網友寫出了「node.js 教學手冊」這本中文書，您可以從 http://book.nodejs.tw/ 這個網址中下載到該電子書。【本文由陳鍾誠取材並修改自維基百科】

## 硬體短訊-Verilog與icarus

從進入超大型積體電路 VLSI 的時代之後，由於硬體 IC 的線路設計的複雜性已經很難用手工設計，因此 IC 電路的設計通常會採用硬體描述語言 (Hardware Description Language, HDL) 進行設計，而 HDL 的兩大語言 – VHDL 與 Verilog 則成爲數位 IC 設計時最常使用的「程式語言」，雖然這種程式語言與一般的程式

語言有相當程度的不同點，但是只要適應了這種語言的特性，程式人就能進入硬體設計的領域了。

Icarus 是一組編譯與執行 Verilog 程式的工具軟體，您可以於其官網1上下載此一軟體的 Linux 版，或者從 bleyer網站2上下載 Windows 版本，然後可以用 iverilog 這個指令編譯程式，接著用 vvp 這個指令執行 verilog 的測試主程式，以便觀察整個電路模擬結果。

Icarus 官網畫面

Icarus 的速度很快，比 Altera Quartus II 等工具快很多，原因應該是 Icarus 只模擬但不需要產生可燒入 FPGA 的碼，因此也不太需要進行電路層級的最佳化，筆者很喜歡這個特性，因爲這可以讓我省去很多等待編譯的時間。【本文由陳鍾誠撰寫】

## 動畫短訊-3D動畫軟體 Blender

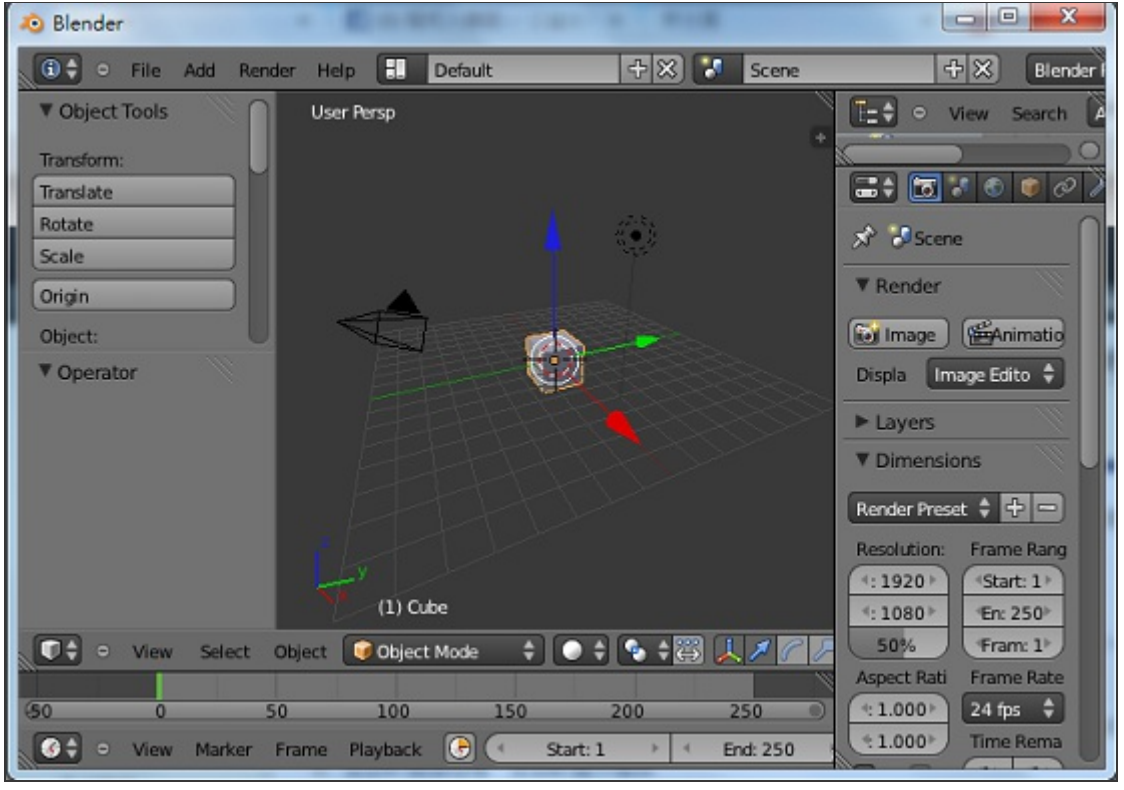
很多人都知道 3D-Max 與 Maya 這兩套 3D 動畫軟體，但是知道 Blender 這套 3D 動畫軟體的人就少了許多。但事實上，Blender 這套軟體的能力已經非常強大，並不比 3D-Max 或 Maya遜色了，而且、Blender 是一套您可以合法免費使用的開放原始碼軟體。

您可以從 Blender 的官網3下載此一軟體，2.65a 版 windows 安裝檔目前大小爲 34MB 。


Blender 下載畫面

Blender 支援 Windows, Linux, MAC OS X, FreeBSD 等環境，而且自從 2.5 版開始，功能就變得很強大，足以與 3D-Max , Maya 競爭了，如果您像筆者一樣，想要瞭解 3D 動畫的領域，從 Blender 開始學習會是一個很不錯的方式。

筆者已經用 Blender 教授動畫四年了，您可以從筆者的網站上取得 Blender 的入門資訊，學習 Blender 最好是觀看教學影片，然後跟著一步一步操作。 下圖是Blender 安裝開啓後的畫面，有興趣的讀者現在就可以立刻下載安裝。



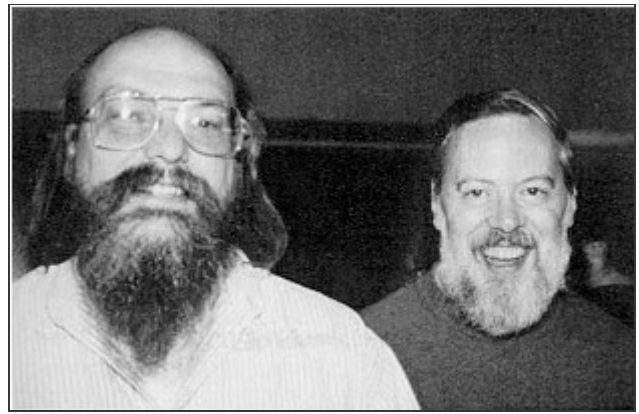

Blender 軟體畫面

歡迎與筆者一同進入 Blender 的 3D 動畫世界。【本文由陳鍾誠撰寫】

# 程式人介紹

## UNIX與C的創造者－Ken Thompson與Dennis Ritchie

如果說程式領域有偉人的話，那這兩位仁兄肯定會入列－湯普遜 (Ken Thompson) & 里奇 (Dennis Ritchie)，他們兩位在 貝爾實驗室帶領一個小組發展出了 UNIX 和 C 語言，影響了千千萬萬的程式人，我們幾乎都活在他們的陰影之下 :)。



Ken Thompson&Dennis Ritchie

最初的Unix是用組合語言編寫的，一些應用是由叫做B語言的解釋型語言和組合語言混合編寫的。B語言在進行系統編程時不夠強大，所以湯普遜和里奇對其進行了改造，並於1971年共同發明了C語言。1973年湯普遜和里奇用C語言重寫了Unix。在當時，爲了實作最高效率，系統程式都是由組合語言編寫，所以湯普遜和里奇此舉是極具大膽創新和革命意義的。用C語言編寫的Unix代碼簡潔緊湊、易移植、易讀、易修改，爲此後Unix的發展奠定了堅實基礎。

1974年，湯普遜和里奇合作在ACM通訊上發表了一篇關於UNIX的文章，這是UNIX第一次出現在貝爾實驗室以外。此後UNIX被政府機關，研究機構，企業和大學注意到，並逐漸流行開來。

1975年，UNIX發行了4、5、6三個版本。1978年，已經有大約600台電腦在執行UNIX。1979年，版本7發行，這是最後一個廣泛發行的研究型UNIX版本。20世紀80年代相繼發行的8、9、10版本只授權給了少數大學。此後這個方向上的研究導致了九號計畫的出現，這是一個新的分布式作業系統。

1982年，AT&T基於版本7開發了UNIX System Ⅲ的第一個版本，這是一個商業版本僅供出售。爲了解決混亂的UNIX版本情況，AT&T綜合了其他大學和公司開發的各種UNIX，開發了UNIX System V Release 1。

這個新的UNIX商業發行版本不再包含原始碼，所以加州大學柏克萊分校繼續開發 BSD UNIX，作爲UNIX System III和V的替代選擇。BSD對UNIX最重要的貢獻之一是TCP/IP。BSD有8個主要的發行版中包含了TCP/IP：4.1c、4.2、4.3、4.3-Tahoe、4.3-Reno、Net2、4.4以及4.4-lite。這些發行版中的TCP/IP代碼幾乎是現在所有系統中TCP/IP實作的前輩，包括AT&T System V UNIX和Microsoft Windows。

作業系統的設計通常是一個很大的工程，幾乎沒有辦法在大學當中傳授實際的設計技巧，以至於所有作業系統的課程幾乎都是以理論爲主的方式，這讓作業系統成爲程式設計師心中的一個未解之謎。

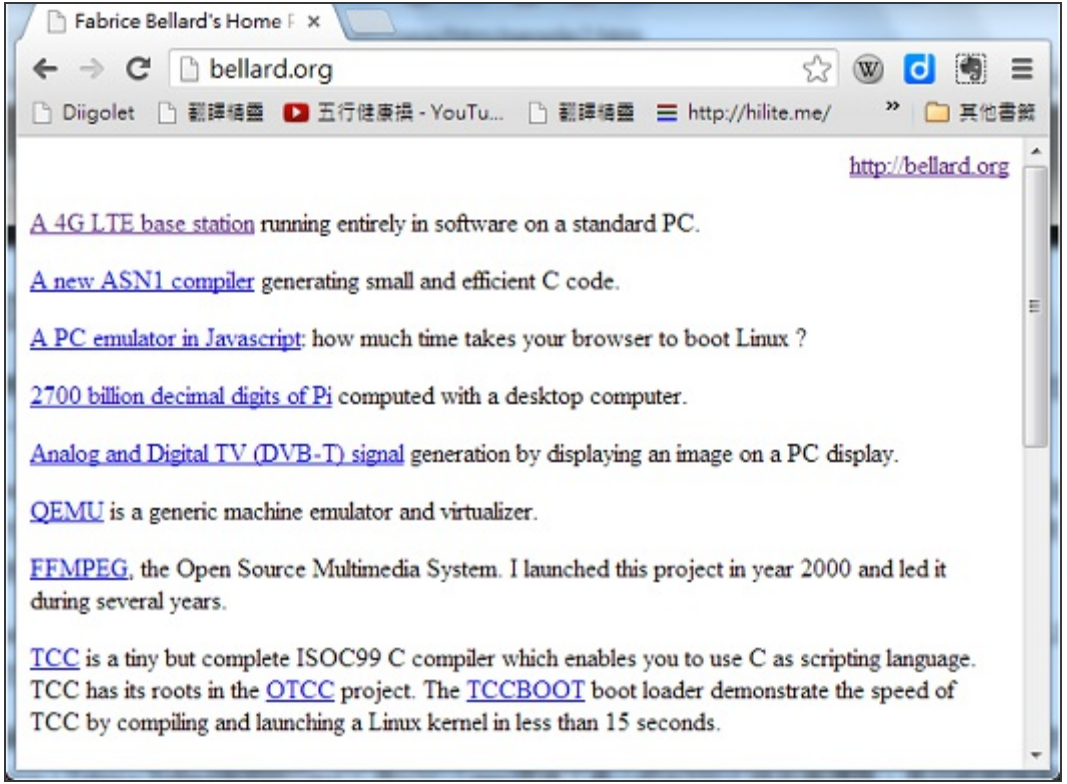
UNIXv6 是一個相當重要且小型的作業系統，幾乎是所有現代 UNIX 的始祖，雖然 UNIXv6 是在 PDP11 機器上寫的，但在網路上卻有相當完整的資料可以參考，UNIX v6 的作業系統核心大約爲 10000 行，因此相當適合想要撰寫作業系統的人學習研究，像是 John Lion 所寫的 Commentary on the Sixth Edition UNIX Operating System 就是一個很有用的參考文獻。 。

在 MIT 的課程當中，有一門編號 6.8283 的 Operating System Engineering，其教師 Frans Kaashoek 將 UNIX v6 的原始碼，修改之後放在 x86 電腦上執行，形成一個只有八千行的 UNIX v6 現代版，稱爲 UNIX xv6，Frans Kaashoek 用這個版本作爲作業系統課程的起點，這讓我們有機會一睹作業系統的核心原理。【本文由陳鍾誠取材並修改自維基百科】

## QEMU的創造者－Fabrice Bellard

如果說 43 歲的我還可以像青少年一樣有偶像的話，那Fabrice Bellard1 2 絕對是我永遠的偶像，這位法國籍的程式設計師，單槍匹馬創造出了一系列讓人驚訝的軟體。

Fabrice Bellard 創造過甚麼軟體呢？您可以從搭他的網站3看到他的作品，其中最知名的是 QEMU 這個虛擬機，另外還有 TinyCC 這個微小又強大的 C 語言編譯器，另外他還能讓 Linux 直接在瀏覽器中用 JavaScript 把整個作業系統 Boot 起來，這樣的技術根本就像神乎其技一般，讓我們嘆爲觀止，您可以從 http://bellard.org/ 中看到他的作品。




Fabrice Bellard 的網站

Fabrice Bellard 出生於1972年法國的Grenoble地區。在高中就讀期間開發了著名的可執行壓縮程序LZEXE，這是當年DOS上第一個廣泛使用的文件壓縮程序。

大學就讀於巴黎高等綜合理工學院，後在國立巴黎高等電信學院攻讀。 1996年,他編寫了一個簡潔但是完整的C編譯器和一個Java虛擬機Harissa。Fabrice Bellard發明的TinyCC是GNU/Linux環境下最小的ANSI C語言編譯器，是目前號稱編譯速度最快的C編譯器。

1997年他提出了最快速的計算圓周率的演算法，是Bailey-Borwein-Plouffe 公式的變體。

在計算圓周率的過程中，Fabrice Bellard使用改良後的查德諾夫斯基方程演算法來進行圓周率的計算，並使用Bailey-Borwein-Plouffe 演算法來驗證計算的結果。爲了紀念他對圓周率演算法所作出的傑出貢獻，Fabrice Bellard所使用的改良型演算法被命名爲Fabrice Bellard演算法，這種演算法是目前所有圓周率演算法中最快的一種，這個計算N位PI的公式比傳統的BBQ演算法要快47%。

1998年編寫了一個簡潔的OpenGL實現TinyGL. 2000年,他化名Gerard Lantau，創建了FFmpeg項目。FFmpeg單詞中的FF 指的是Fast Forward，FFmpeg這個2000年發起著名的開源多媒體播放器項目， 是MPlayer的姊妹項目。這是一個如此重要的成就，沒有這個項目，就沒有我們今天廣爲使用的暴風影音等播放器。這個多平台、多功能的多媒體編碼解碼器由Fabrice Bellard發起並管理，現在是由Michael Niedermayer在進行維護。

2003年,開發了Emacs克隆版QEmacs。

2004年,他編寫了一個只有138KB的啓動載入程序TCCBOOT，可以在15秒內從原始碼編譯並啓動Linux系統。

2005年,用普通PC和VGA卡設計了一個數字電視系統。

2009年12月31日，他聲稱打破了圓周率計算的世界紀錄，算出小數點後2.7萬億位，僅用一台普通PC機。他使用的個人PC價格不到2000歐元，僅用了116天，就計算出了PI的小數點後第2.7萬億位，超過了由目前排名世界第47位的T2K Open 超級計算機於2009年8月17日創造的世界紀錄。新紀錄比原紀錄多出1200億位，然而，他使用的這台桌面電腦的配置僅爲：2.93GHz Core i7 CPU，6GB內存，7.5TB硬碟!

2011年，他單用JavaScript寫了一個PC虛擬機 Jslinux。這個虛擬機模擬了一個32位的x86兼容處理器，一個8259可程式中斷控制器，一個8254可程式中斷計時器，和一個16450 UART。

2012年, 在PC上用軟體實現4G LTE基站.

Fabrice Bellard 的作品通常有個特色，程式碼採用極簡風格，完全單人手工打造，通常又小又精美，而且通常速度奇快無比。筆者真的對他是景仰萬分，這讓我不禁開始想像，哪一天我可以創造出像他那樣的軟體 … (筆者開始作白日夢中…)【本文由陳鍾誠取材並修改自維基百科】。

# 程式人頻道

## 看影片學 Arduino

最近這幾年，我逐漸養成了一個習慣，如果我想學習一個新的領域，就會先上 Youtube 找找相關影片看看。前一陣子我開始對單晶片與 Arduino 發生興趣，於是除了買 一組Arduino套件 來玩玩看之外，也到 Youtube 找了一些 Arduino 的相關影片來看。

我發現用影片學 Arduino 是很棒的方法，像是 Jeremy Blum 這位年輕人，就拍了一系列的 Arduino 教學影片，並且在 Jeremy Blum 網站的 ARDUINO TUTORIALS 中附上完整的程式碼3，以下是Jeremy 的一系列 Arduino 教學影片整理。

| 主題 | 影片網址 | 說明 |
| --- | --- | --- |
| Getting Acquainted with Arduino | http://youtu.be/fCxzA9_kg6s | 熟悉 Arduino |
| Buttons, PWM, and Functions | http://youtu.be/_LCCGFSMOr4 | 按鈕、函數、振幅調整 |
| Electrical Engineering Basics | http://youtu.be/abWCy_aOSwY | 電子學基礎 |

| | | |
|---|---|---|
| Analog Inputs | http://youtu.be/js4TK0U848I | 類比輸入 |
| Motors and Transistors | http://youtu.be/5bHPKU4ybHY | 馬達與電晶體 |
| Serial Communication and Processing | http://youtu.be/g0pSfyXOXj8 | 序列通訊 |
| I2C Communication and Processing | http://youtu.be/GJX0BRUagCg | I2C 通訊 |
| SPI Interfaces | http://youtu.be/1nO2SSExEnQ | SPI 界面 |
| Wireless Communication | http://youtu.be/vKVNmA8C6m8 | 無線通訊 |
| Interrupts and Hardware Debouncing | http://youtu.be/CRJUdf5TTQQ | 中斷與去除跳動 |
| SD Cards and Datalogging | http://youtu.be/5v5A3j7Rrco | 用SD Card記錄資料 |
| RFID Card Reading | http://youtu.be/gIlSLwcbeTU | 讀取 RFID 卡 |
| Liquid Crystal Displays (LCDs) | http://youtu.be/oIiDseJO4dM | LCD 液晶顯示 |
| Holiday Lights and Sounds Spectacular! | http://youtu.be/CoG_Czyr7z0 | 光線與聲音 |
| GPS Tracking | http://youtu.be/TtZEZYQG0xk | GPS衛星定位 |

看完了 Jeremy 的示範，覺得真的相當過癮。但是獨學而無友，則孤陋而寡聞，還好網友 Cooper Maa 又在網誌中介紹了 Stefan Hermann 的教學影片，這一系列的影片則是又短又精彩，Copper Maa 以經幫我們整理成網誌了，您可以連到以下網址去看看Stefan Hermann 的教學影片與 Cooper Maa 的摘要，保證不虛此行。

- http://coopermaa2nd.blogspot.tw/2011/12/arduino-video-workshop.html

# 程式人文集

## Arduino入門教學(2) - 準備開發環境（作者：Copper Maa）

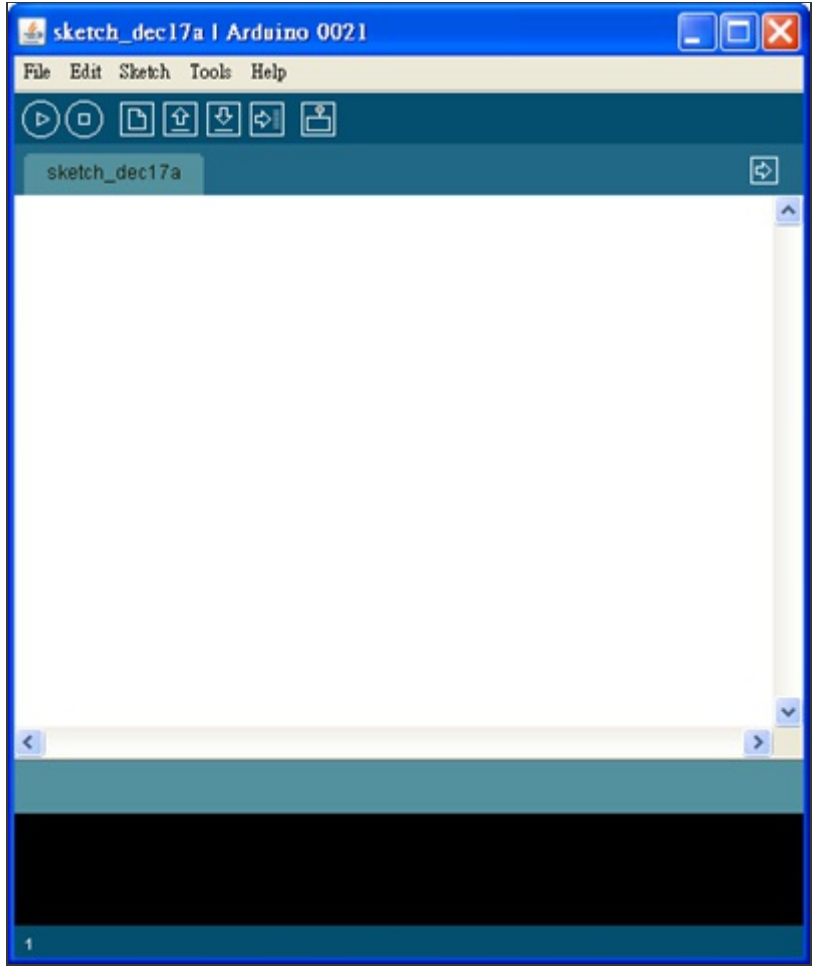
### 取得 Arduino 板子與 USB 傳輸線

首先，你必須先準備一張 Arduino 板子與一條 USB 傳輸線:

要購買 Arduino 板子，可以到國外網站 http://www.sparkfun.com 購買，或是到 Arduino.TW 樂園、藝科資訊商城 或者是露天拍賣這幾個台灣網站購買。Arduino 的板子有很多種版本，用 Arduino Uno, Arduino Duemilanove 或是 Diecimila 都可以，一張板子大約 30 塊美金左右。

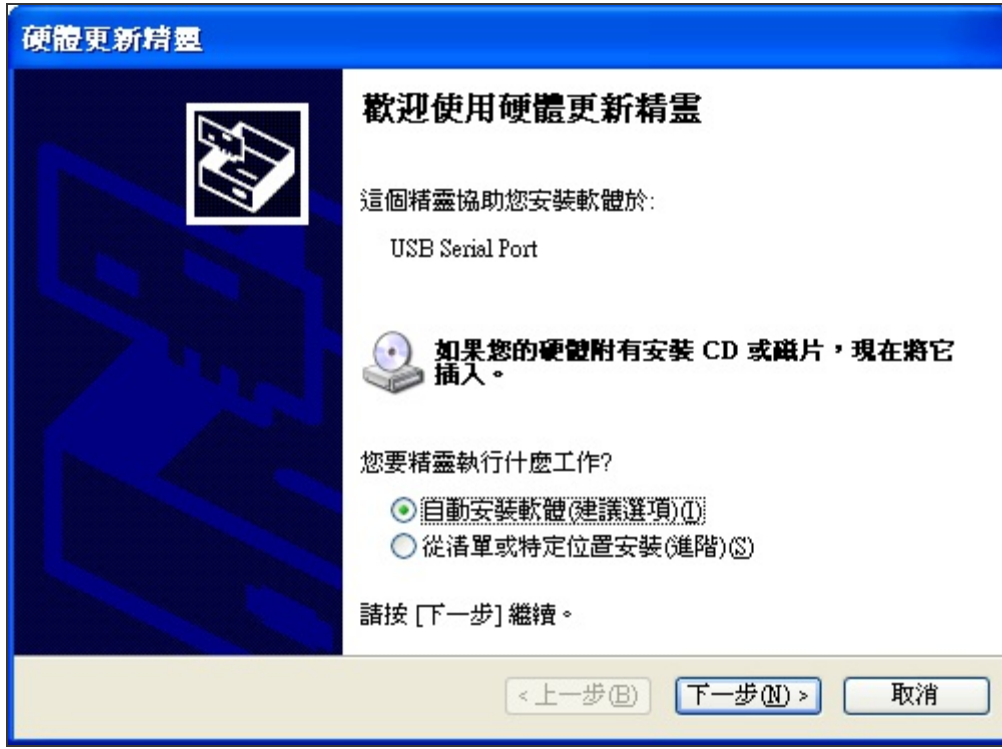
### 下載 Arduino 開發環境

到 Arduino 官方網站 arduino.cc 下載軟體。Arduino 軟體不需要安裝，下載後解開壓縮檔即可使用，解壓縮之後，雙擊 arduino.exe 就可以啓動主程式。Arduino 的軟體介面如下：

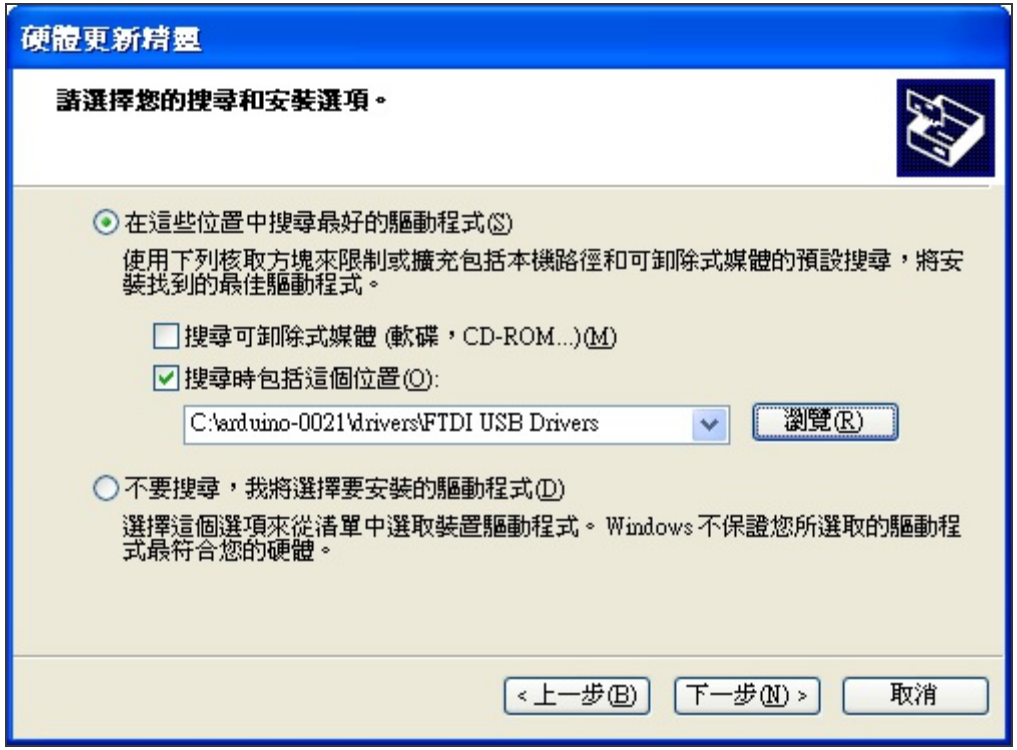

sketch_dec17a | Arduino 0021
File Edit Sketch Tools Help
sketch_dec17a
1

## 連接控制板與安裝驅動程式

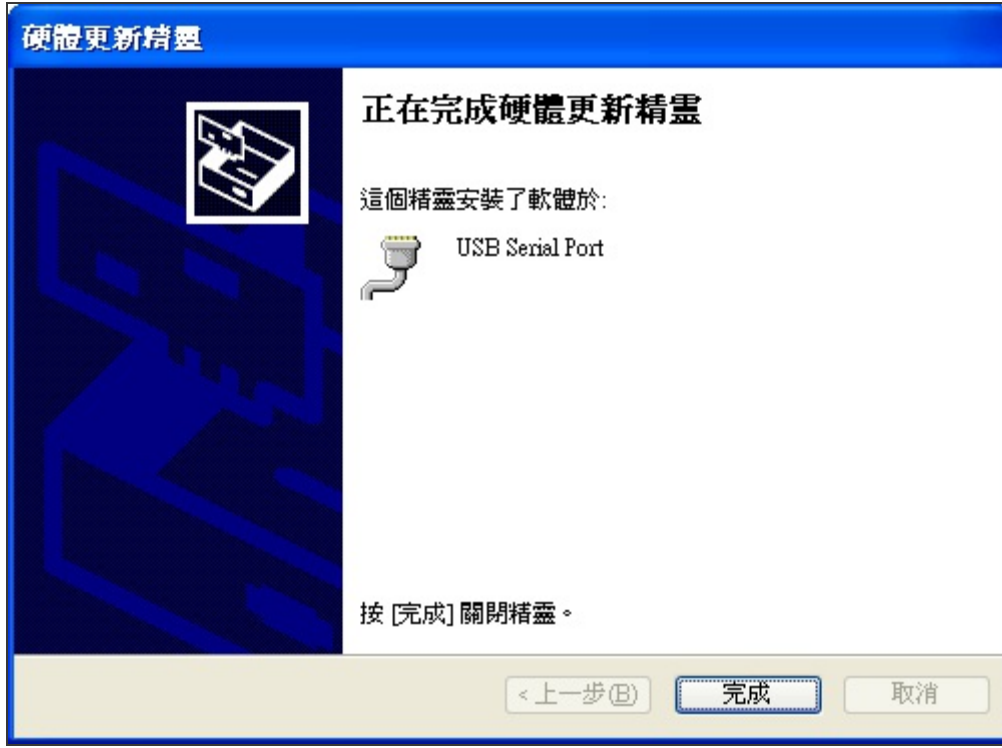
Arduino 板子可以透過 USB 供電，不需要另外接電源。將 USB 傳輸線 一端接到電腦，一端接到 Arduino 板子後，會出現 FTDI 驅動程式安裝畫面：



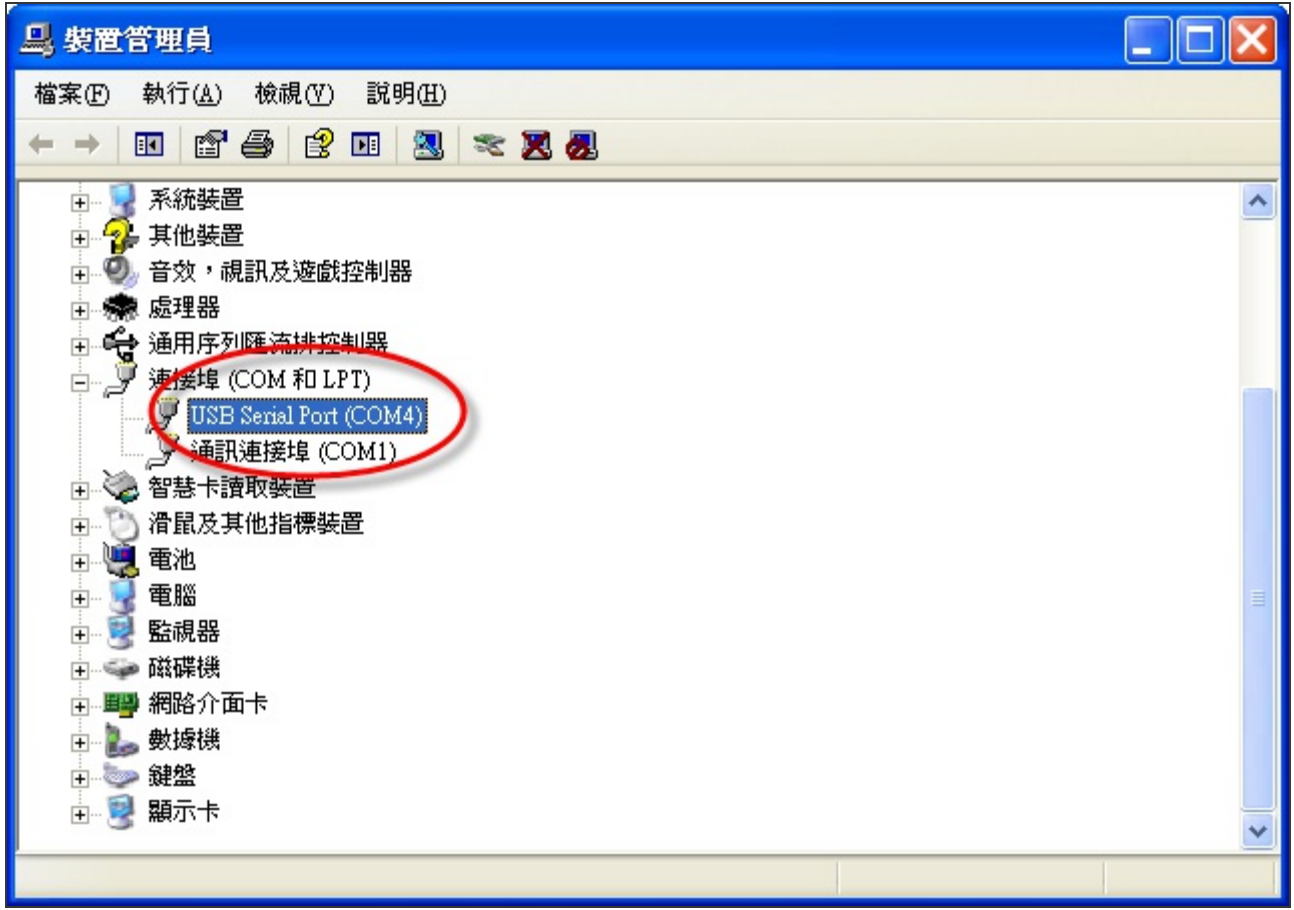
按下「下一步」，接著要指定驅動程式的位置。Arduino 軟體裏已經內附 FTDI 驅動程式，直接把位置指定到 “Arduino-00xxUSB Drivers” 後開始進行安裝，例如我用的版本是 Arduino-0021，所以我指定 “C:-0021-driversUSB Drivers”：




安裝完成的畫面：

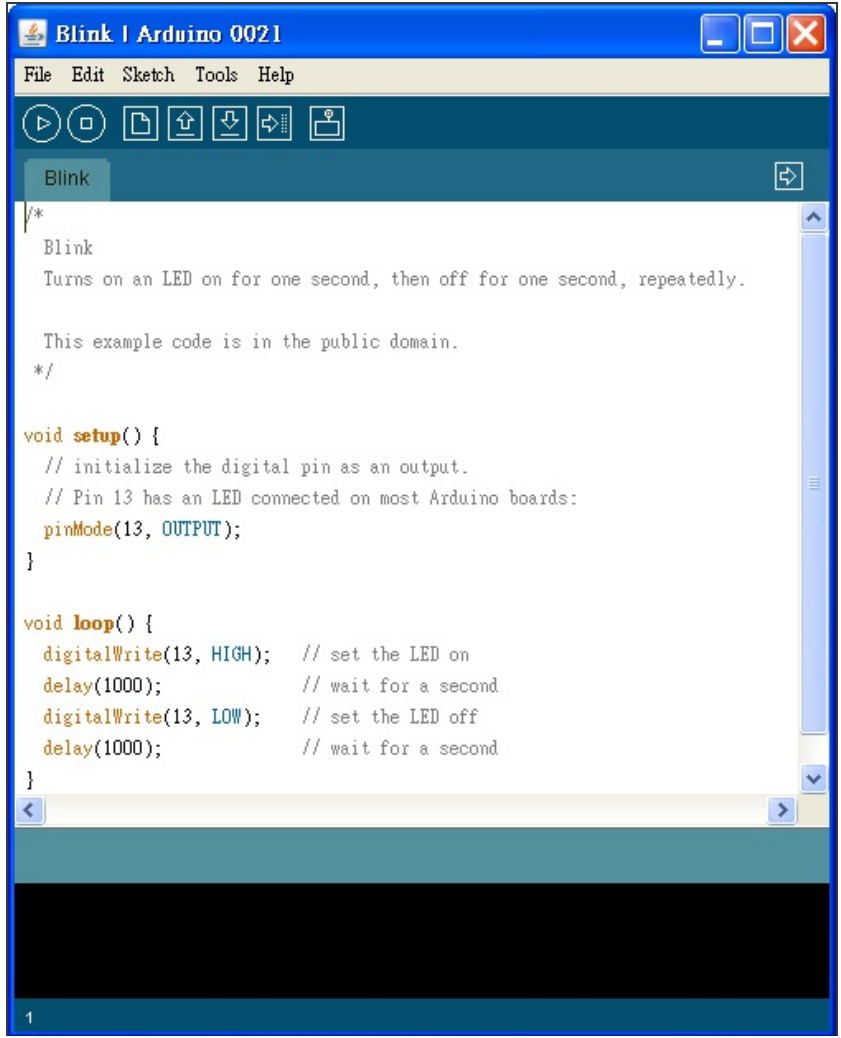



打開「裝置管理員」，檢查 Arduino 連接在哪個 COM Port。以我的電腦爲例，我的 Arduino 板子接在 COM4：

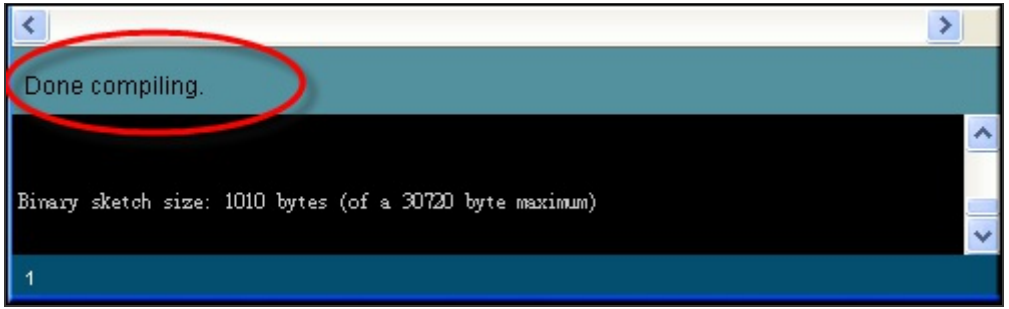



開啓並編譯 blink 範例程式

雙擊 Arduino.exe 啓動 Arduino 主程式，點選 File > Examples > 1.Basics > Blink. 打開 Blink 範例程式：

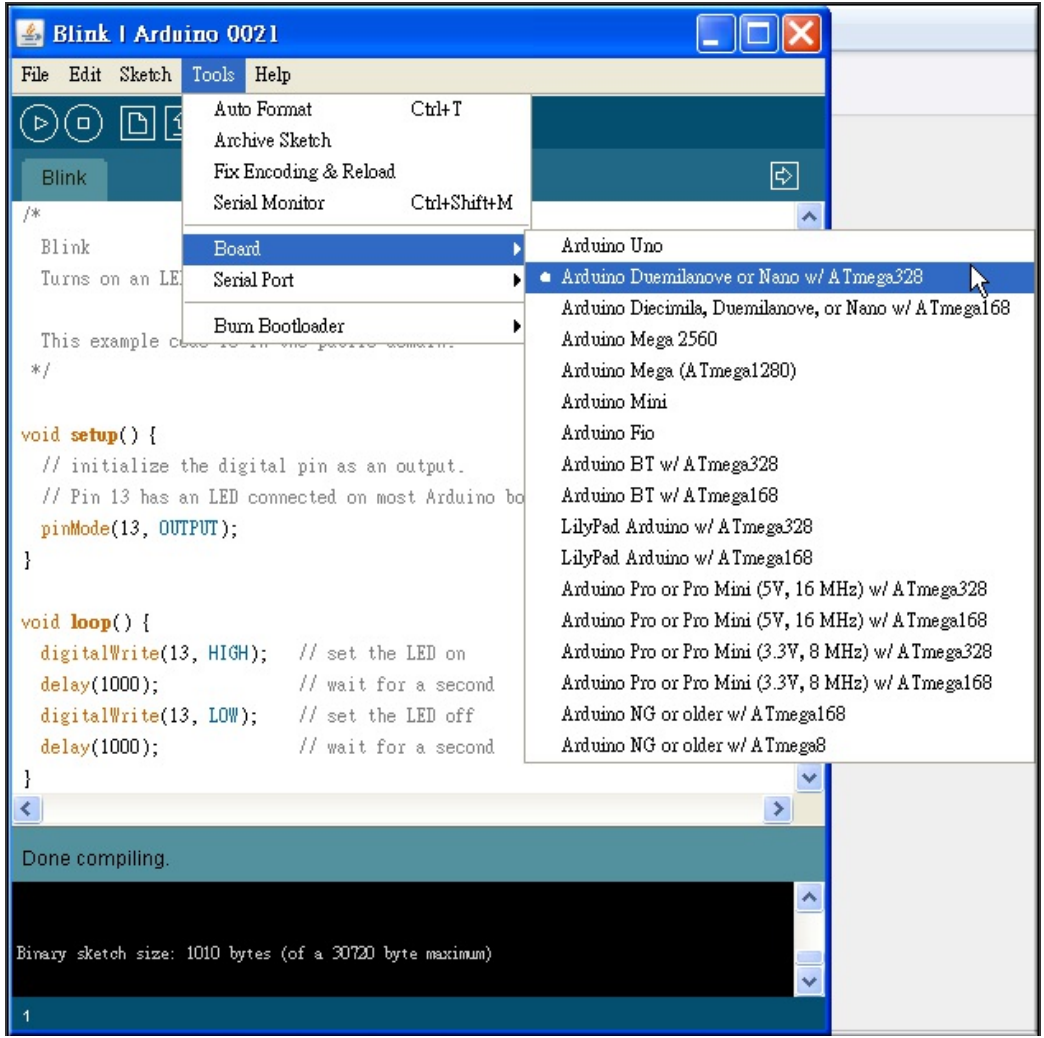

Blink | Arduino 0021
File Edit Sketch Tools Help
Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino boards:
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
1

接著按下 Verify 這個按鈕編譯程式，假如程式語法沒有錯誤，畫面下方的狀態列會出現 “Done Compiling.” 的訊息:



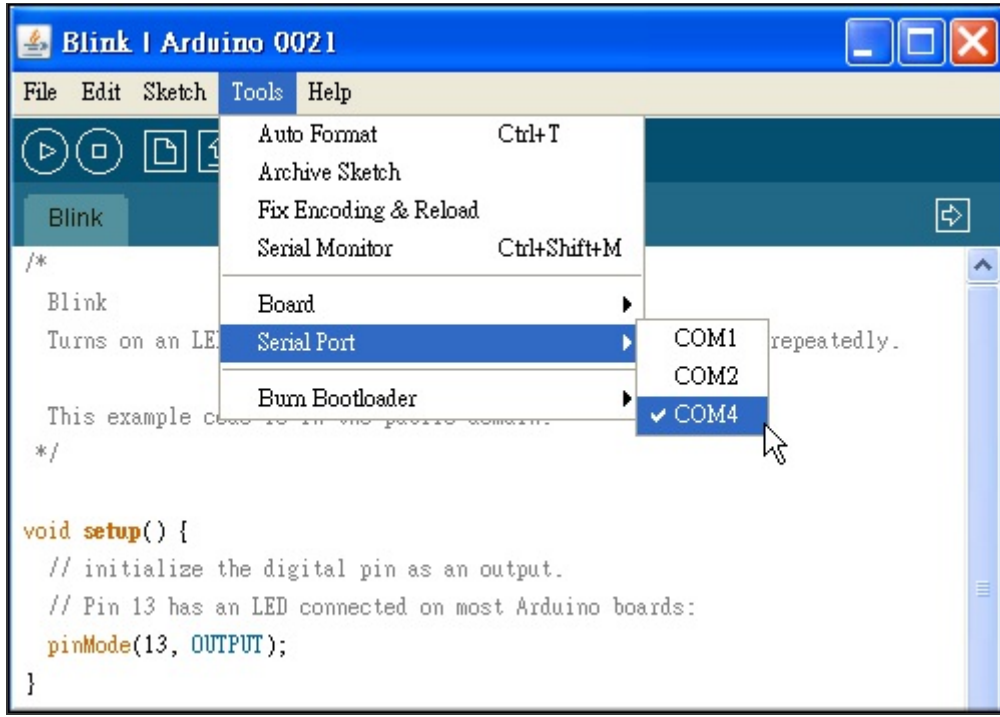

## 選擇 Arduino 控制板

點 Tools > Board 選擇跟你所用的 Arduino 對應的板子。我用的是 Arduino Duemilanove，所以我選擇 “Arduino Duemilanove or Nano w/ ATmega328” 這個選項：


Blink | Arduino 0021
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Burn Bootloader
Arduino Uno
Arduino Duemilanove or Nano w/ ATmega328
Arduino Diecimila, Duemilanove, or Nano w/ ATmega168
Arduino Mega 2560
Arduino Mega (ATmega1280)
Arduino Mini
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
Blink
/*
Blink
Turns on an LE
This example
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino bo
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
Done compiling.
Binary sketch size: 1010 bytes (of a 30720 byte maximum)
1

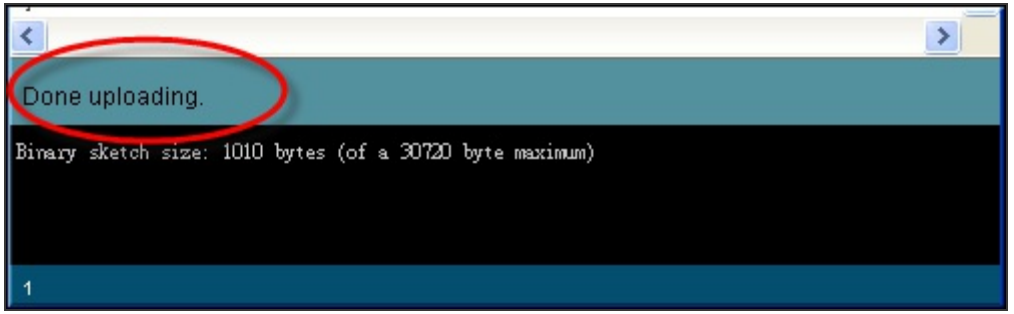
## 選擇 Serial port

點 Tools > Serial Port 選擇 COM Port，以我的電腦爲例，我的 Arduino 板子接在 COM4 上，所以我選擇 COM4：

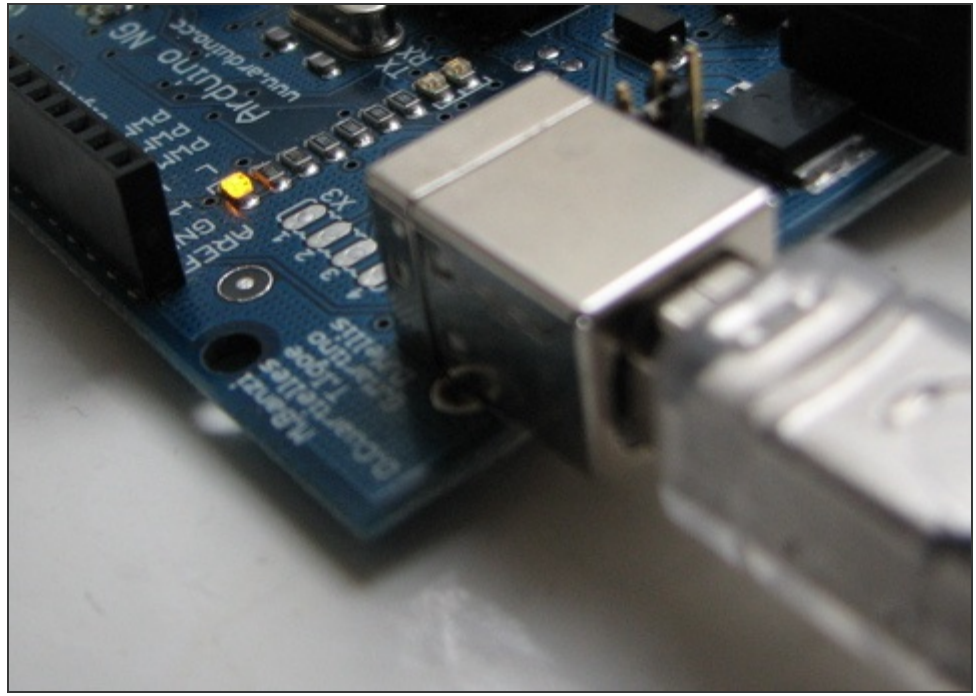


## 上傳程式

最後按下 Upload 這個按鈕，等候幾秒鐘，應該會看到板子上的 RX 和 TX 兩個燈號會快速地閃爍，如果上傳成功，狀態列會出現 “Done uploading.” 的訊息：




上傳完畢後，經過幾秒鐘後，板子上 pin 13 (L) 的燈號(黃色的燈號)就會開始閃爍，如果是的話，代表你的 Arduino 開發環境已準備好，可以開始學習寫 Arduino 程式了：




(本文作者爲馬萬圳，原爲網誌上的文章，經作者授權給程式人雜誌後由陳鍾誠編輯爲此文。原文連結：http://coopermaa2nd.blogspot.tw/2010/12/arduino.html)

## JavaScript（2） - 以原型為主的物件導向（作者：陳鍾誠）

雖然說，JavaScript 也能實作很好的物件導向功能，但是如果我們說 JavaScript 是一種物件導向語言，那麼會很容易造成混淆。因爲 JavaScript 的物件導向非常非常的特別，與 Java, C++, C# 等語言的物件導向實作方法有很大的不同。

嚴格的說，JavaScript 是一種原型導向語言，原型導向是一種特別簡單的物件導向實作機制，以下是幾個 JavaScript 的物件範例：

```
obj = new Object()
obj.x = 3;       // 爲 obj 新增一個欄位 x，其值設定爲 3
obj.y = 5;       // 爲 obj 新增一個欄位 y，其值設定爲 5
obj.z = obj.x + obj.y;    // 爲 obj 新增一個欄位 z，其值設定爲 x+y
```

對於曾經使用像 Java 這種傳統物件導向語言的人而言，會感覺到上述的程式很奇特，因爲 obj 一開始只是一個空物件，並沒有包含任何的欄位，但是我們透過指定的方式，動態的爲物件增添了 x, y, z 等欄位。

物件導向基本上有三大特性，1.封裝 2. 繼承 3. 多型，瞭解這三個特性的實作方式，通常就可以學會一種語言的物件導向語法了，以下我們將分別針對 JavaScript 中的這三大特性進行介紹。

## JavaScript 物件的封裝

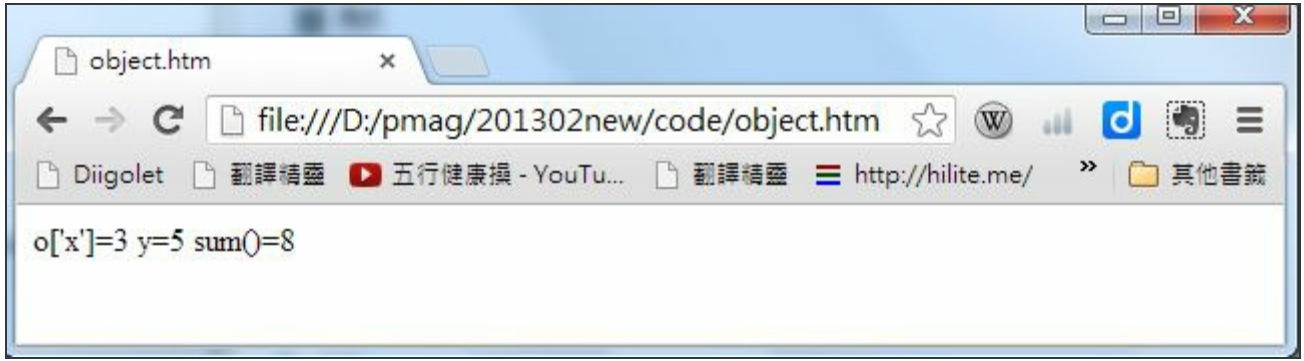
物件導向中的封裝特性，是指將「資料」與「函數」封入一種稱爲「物件」的結構當中，以下是 JavaScript 的一個物件範例，其中 x, y 是資料，而 sum 則是函數，這些成員都被封裝在 obj 這樣一個建構函

數當中，因此我們呼叫 var o = new obj() 這個指令時，就會建立一個新的物件，並傳回給 o 變數。

程式範例: object.htm

```
<html>
<body>

<script type="text/javascript">
function obj() {
  this.x = 3;       // 第一種寫法，爲 obj 新增一個欄位 x，其值設定爲 3
  this["y"] = 5;    // 第二種寫法，爲 obj 新增一個欄位 y，其值設定爲 5
  this.sum = function() { return this.x + this.y; } // 爲 obj 新增一個欄位 add，其值爲一個匿名函數
}

var o = new obj();

document.write("o['x']="+o['x']+" y="+o.y+" sum()="+o.sum());
</script>

</body>
</html>
```

執行結果




在 JavaScript 當中，並沒有像 Java 或 C# 這樣的 class 關鍵字，沒有類別這樣的明顯概念，不會外顯的定義出物件的樣板，而是直接在建構函數中建立 this 物件的成員，然後在用 new 建立物件，這種方式很動態也很彈性，但卻不是一般的物件導向製作方式，與 Java 和 C# 有很明顯的差異。

## JavaScript 物件的繼承

物件導向中的繼承特性，是指「子物件」可以繼承「父物件」的資料與函數，並且進行修改，這樣我們就不需要重複的實作父物件已經有的函數，或者定義父物件已經有的資料，以達成程式碼重用的目的。

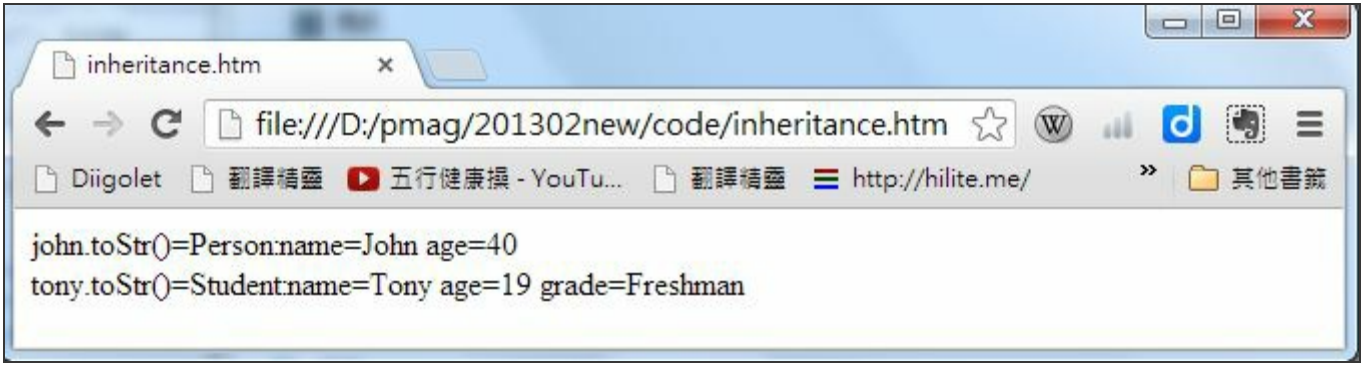
以下範例中的 Student 物件就繼承了 Person 物件，其方法是在 Student 物件的建構函數中，指定 this.prototype = Person 以達到繼承的目的，然後再呼叫 this.prototype(name, age) 以呼叫父物件的建構函數，建立出 name, age, toStr() 物件內容。

程式範例: inheritance.htm

```
<html>
<body>

<script type="text/javascript">
function Person(name, age) {
  this.name = name;
  this.age = age;
  this.toStr = function() {
    return "Person:name="+this.name+" age="+this.age;
  }
}
var john = new Person("John", 40);

document.write("john.toStr()="+john.toStr()+"<BR/>");

function Student(name, age, grade) {
  this.prototype = Person;
  this.prototype(name, age);
  this.grade = grade;
```

```
  this.toStr = function() {
    return "Student:name="+this.name+" age="+this.age+" grade="+this.grade;
  }
}

var tony = new Student("Tony", 19, "Freshman");

document.write("tony.toStr()="+tony.toStr()+"<BR/>");
</script>

</body>
</html>
```

執行結果




從上述的範例中您可以看到，JavaScript 的繼承是直接在子物件的建構函數當中呼叫父物件的建構函數所完成的，範例中的 prototype 這個屬性實作了繼承的特性，並且透過 this.prototype(name, age) 這樣的呼叫完成了父物件初始化的動作，這種方式繼動態又彈性，但仍然與 Java, C# 等語言有很大的不同，因此我們稱這種方式爲基與原型的物件導向實作方式。

## JavaScript 物件的多型

物件導向中的多型特性，是指當不同的「子物件」繼承同一個「父物件」時，我們可以透過宣告父物件容器，卻將內容指向子物件的方式，以便呼叫到不同子物件中的對應函數。

以下範例中的 var array = [ john, tony ] 這一行，其中的 john 是 Person 類型的物件，tony 是 Student 類型的物件，但是由於 Student 繼承了 Person，而且兩者都有 toStr() 函數，因此當我們用 for (i in array) … array[i].toStr() … 這樣的方式呼叫 array[i].toStr() 時，對 john 物件會呼叫到 Person 的 toStr(), 而對 tony 物件

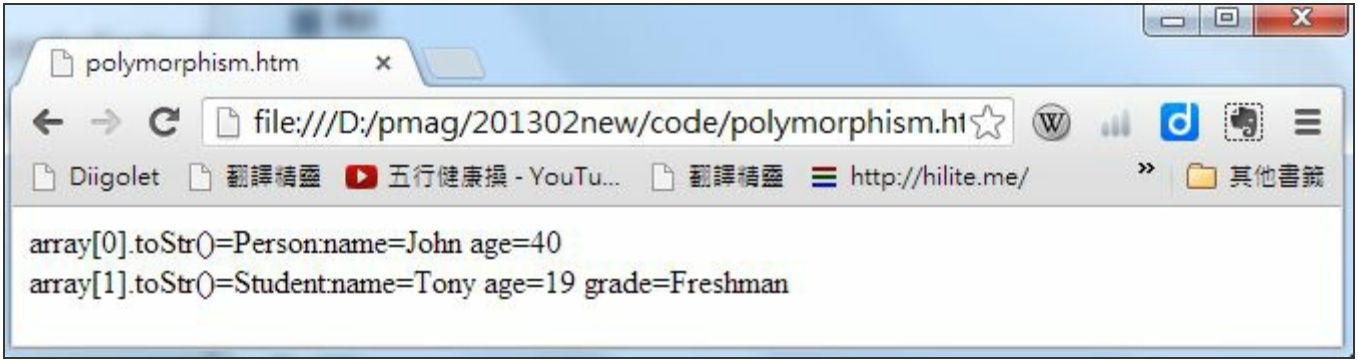
會呼叫到 Student 的 toStr(), 這樣就達到了多型的結果。

程式範例: polymorphism.htm

```
<html>
<body>

<script type="text/javascript">
function Person(name, age) {
  this.name = name;
  this.age = age;
  this.toStr = function() {
    return "Person:name="+this.name+" age="+this.age;
  }
}
var john = new Person("John", 40);

function Student(name, age, grade) {
  this.prototype = Person;
  this.prototype(name, age);
  this.grade = grade;
```

```
  this.toStr = function() {
    return "Student:name="+this.name+" age="+this.age+" grade="+this.grade;
  }
}

var tony = new Student("Tony", 19, "Freshman");

var array = [ john, tony ];

for (i in array)
  document.write("array["+i+"].toStr()="+array[i].toStr()+"<BR/>");
</script>

</body>
</html>
```

執行結果




## JavaScript 物件的精簡表示法 - JSON 物件資料交換格式

如果我們想直接在程式中宣告一個複雜的物件，可以使用 JavaScript 中的 {…} 與 […] 的語法組合，用簡單的語法建構出整個物件。這種格式也常被用在網頁程式的資料交換當中，因此有一個很特別的名稱叫 JSON （Javascript Object Notation）。

目前網路上最常使用的資料交換格式是 XML，但是 XML 文件很繁瑣且囉嗦，讓使用者撰寫不方便，而且不容易嵌入網頁中進行處理。爲了讓網頁上的共通程式語言 JavaScript 可以輕易的交換資料，網頁程式的設計者也常用 JSON 取代 XML 進行資料交換。

以下是一個採用 JSON 格式的朋友資料範例，該範例中有兩個朋友，一個是 John, 22 歲，另一個是 Mary, 28 歲。

```
{
  "friends": [
    {"name": "John", "age": 22 }
    {"name": "Mary", "age": 28 }
  ]
}
```

程式範例：json.htm

```
<html>
<body>

<script type="text/javascript">

var george = {
  "name": "George",
  "age": 25,
  "friends": [
    {"name": "John", "age": 22 },
    {"name": "Mary", "age": 28 }
  ]
```

```
};

document.write("george.age="+george.age+"<br/>");
document.write("george.friends:\n<ol>");
var friends = george.friends;
for (i in friends)
  document.write("<li>"+friends[i].name+" is "+friends[i].age+"years old!</li>");
document.write("</ol>");
</script>

</body>
</html>
```

執行結果


## 結語

以上我們簡單的介紹了 JavaScript 的物件導向功能，這種物件導向的實作方式由於是以原型爲核心的，因此筆者喜歡稱 JavaScript 爲原型導向 (prototype oriented) 的語言，雖然並不是很多人使用「原型導向」這個用法。

在這兩期當中，我們從第一期的 JavaScript 語法基礎開始，到本期的物件導向介紹，大致已經將 JavaScript 的基本語法介紹完畢。在下一期當中，我們將會討論較爲進階的 JavaScript 語言特性，像是匿名函數、閉包 (closure)、this 關鍵字的用法等等，我們下期見！

# 三角網格化 - 以Z-Scan演算法為例（作者：吳栢融）

三角網格化（Triangulation）是電腦圖學（Computer Graphics）中有關建模技術（Modeling）的一個關鍵課題，我們都知道模型檔案最基本的由點和面構成。而對於一個面所構成的點我們並無嚴格規範，換言之，三角形，矩形都是一個面。如此一來再檔案的存取上造成許多麻煩的問題，如必須動態配置每一個面到底有多少索引、規劃每個面採何種方式來絢爛（Rending）。另外非常多的圖學演算法只作用與三角網格化後的多邊形網格，故通常我們會將模型轉換成三角形網格後發布、或做其他演算。

三角網格化有非常多方法，筆者就不再一一細說，本文以Z-Scan的方式來討論如何將模型三角化。有關Z-Scan的方法提出筆者目前尚未得知爲何人。如有讀者知道還請來信告知。感謝!!

## 基礎理論

我們都知道一個模型由點和面構成，但對於三角網格而言，他們是否存在一定的關係呢？事實上，我們可以透過歸納法來驗證下述公式。

F=V-2

其中V爲構成該mesh的點的個數，F爲若將該mesh三角網格化以後會得到的面的數量。我們可以參考下圖驗證之。

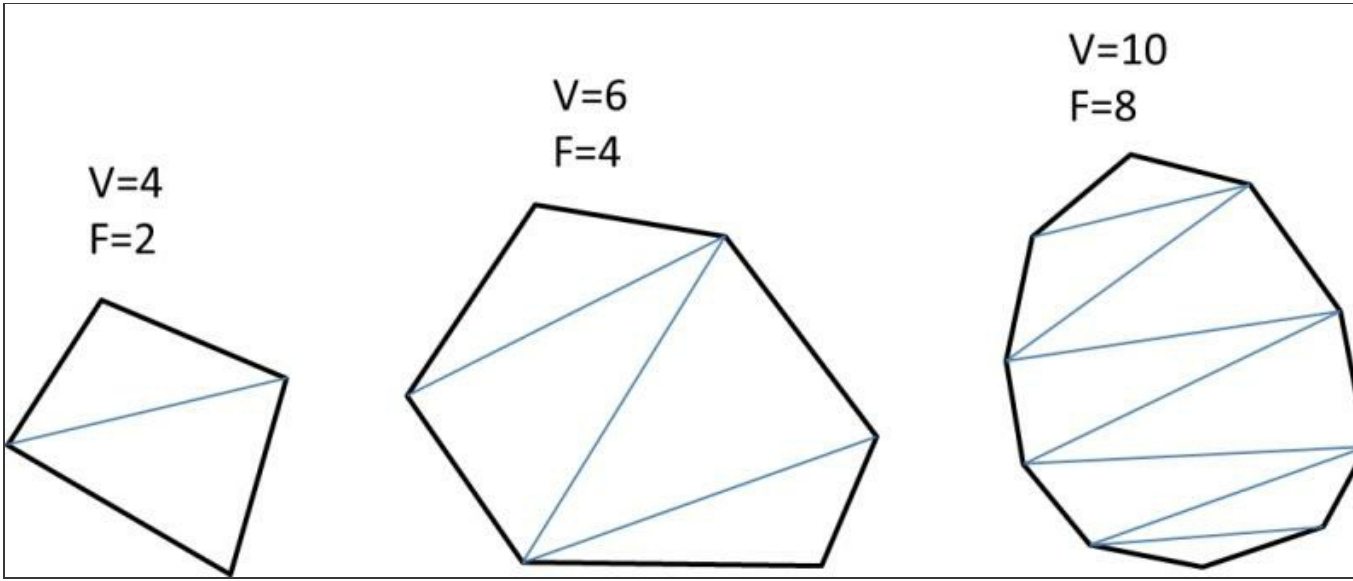



圖1:面與點的關係圖

此外，通常在OpenGL的環境中我們爲了統一normal我們預設mesh的index是依照逆時針的方式記錄。如下圖所示，由於我們統一規範mesh的排序方式，所以我們可以藉由數學公式計算出mesh的normal，而這個normal的功能，筆者就不再這說明了。

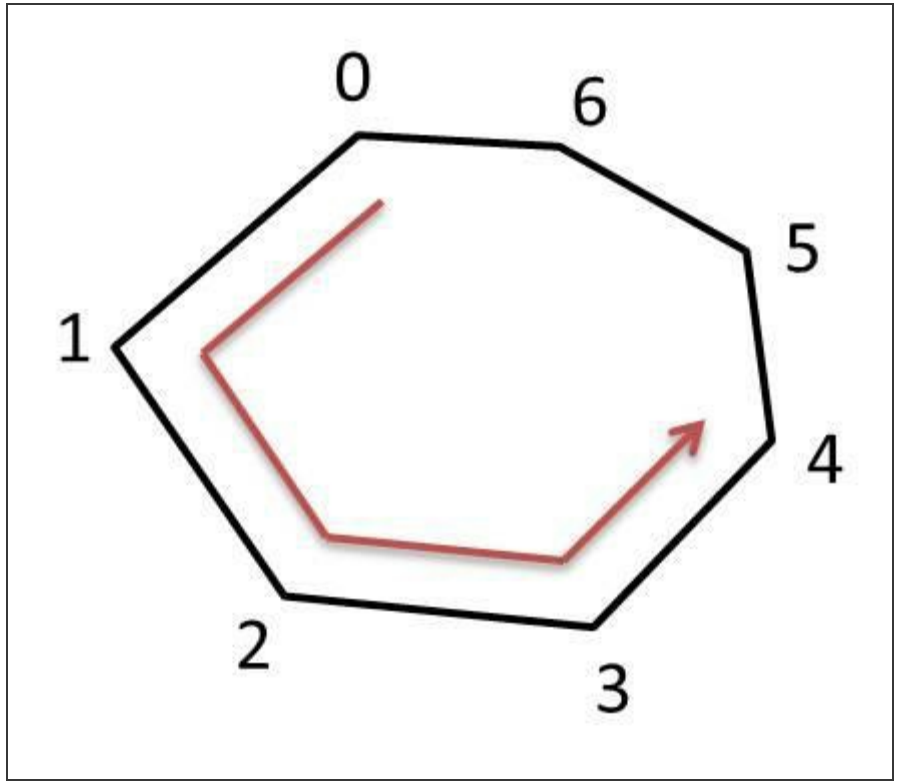



圖2:逆時針送點示意圖

## 程式分析

爲了取得一個最簡單的算法，筆者使用了歸納法來解這個問題，筆者假設圖3爲圖2三角化的結果，而該多多邊形共計7個頂點（V=7），估根據上述公式我們可以得到三角化後應該產生5個三角形面（F=V-2=5），筆者將其標記爲F0-F4，如圖3所示。並將每個三角形對應的三角形歸納入表1所示。

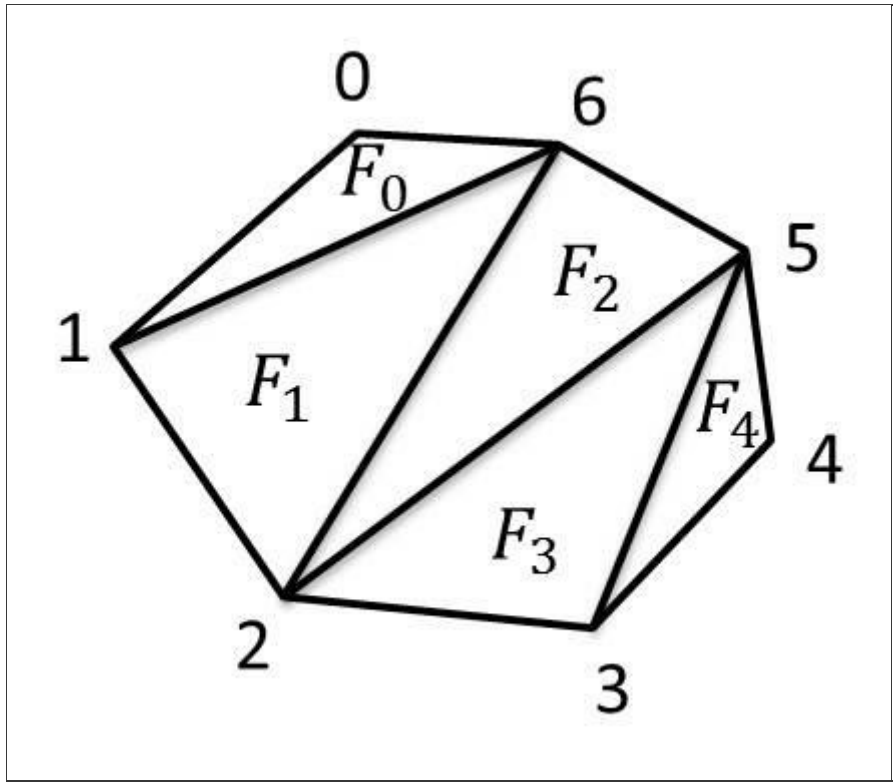



圖3:分割圖

| | Index[0] | Index[1] | Index[2] |
|---|---|---|---|
| $F_0$ | 0 | 1 | 6 |
| $F_1$ | 6 | 1 | 2 |
| $F_2$ | 6 | 2 | 5 |
| $F_3$ | 5 | 2 | 3 |
| $F_4$ | 5 | 3 | 4 |

表1:三角化後的每個三角形的對應索引

從上表我們可以得到如下關係:

- Index[0] 的次序爲 0665544332….不斷遞減
- Index[1] 的次序爲 1122334455….不斷遞增
- Index[2] 的次序爲 62534… 分成奇偶分別遞增遞減

於是我們做以下測試:

```
int V=7;
int F=V-2;
```

針對Index[0]有:

```
for(int i=0;i<F;i++){
    if(i==0)
        cout<<i<<"="<<0<<endl;
    else
        cout<<i<<"="<<V-(i+1)/2<<endl;
}
```

針對Index[1]有:

```
for(int i=0;i<F;i++){
    cout<<i<<"="<<(i+2)/2<<endl;
}
```

針對Index[2]有:

```
for(int i=0;i<F;i++){
    if(i%2==0)
        cout<<i<<"="<<V-(i+2)/2<<endl;
    else
        cout<<i<<"="<<(i+3)/2<<endl;
}
```

根據上述的分開測試。我們可以將所有程式合併在一起，如下所示：

## 完整代碼

```
for(int i=0;i<F;i++)
{
        if(i==0){
            cout<<i<<"="<<0<<"\t"<<i+1<<"\t"<<V-1<<endl;
```

```
        }else{
            if(i%2==0)
              cout<<i<<"="<<V-(i+1)/2<<"\t"<<(i+2)/2<<"\t"<<V-(i+2)/2<<endl;
            else
              cout<<i<<"="<<V-(i+1)/2<<"\t"<<(i+2)/2<<"\t"<<(i+3)/2<<endl;
        }

    }
```

## 後記

其實三角網格化還有非常多的方式可以完成，筆者本篇僅拋磚引玉，希望大家可以分享新的方法。

# 智慧型自動編譯 C 語言 VB 6.0 8X8 LED To Keil C Code For 89S51（作者：廖憲得）

## 軟體介面


## 軟體介紹與設計構想

這是一套由Visual Basic 6.0 設計的軟體，使用來與 89S51、8051 等其他單晶片結合使用的自動轉C語言工

具。它的功能可以由使用者輕輕鬆鬆使用滑鼠點一點想要的花樣，並且一鍵自動產生爲C語言的程式碼，省得晶片開發設計者需要花費許多計算時間來編譯一套屬於自我花樣的C語言，利用這套軟體只需要動動滑鼠，小小的幾個動作就能夠編譯出C語言的程式碼，甚至它可以搭配Keil C 等其他 89S51、8051 單晶片的編譯軟體。

由於身爲高職生的我，因爲單晶片C語言89S51設計這門課的專題所需，爲了有效率且快速的完成在矩陣式8x8 LED小綠人、小紅人…等等專題上，很多花樣LED燈的閃法、跑法、亮法剛開始都需要以人工的方式慢慢計算在編譯，常常因爲一個不小心算錯而導致LED燈的變化不如預期，而使用Visual Basic 6.0 開發了這一套軟體，只需要透過基本的演算法、搭配字串的資料處理，很快的就能夠編寫出這樣的一套軟體。

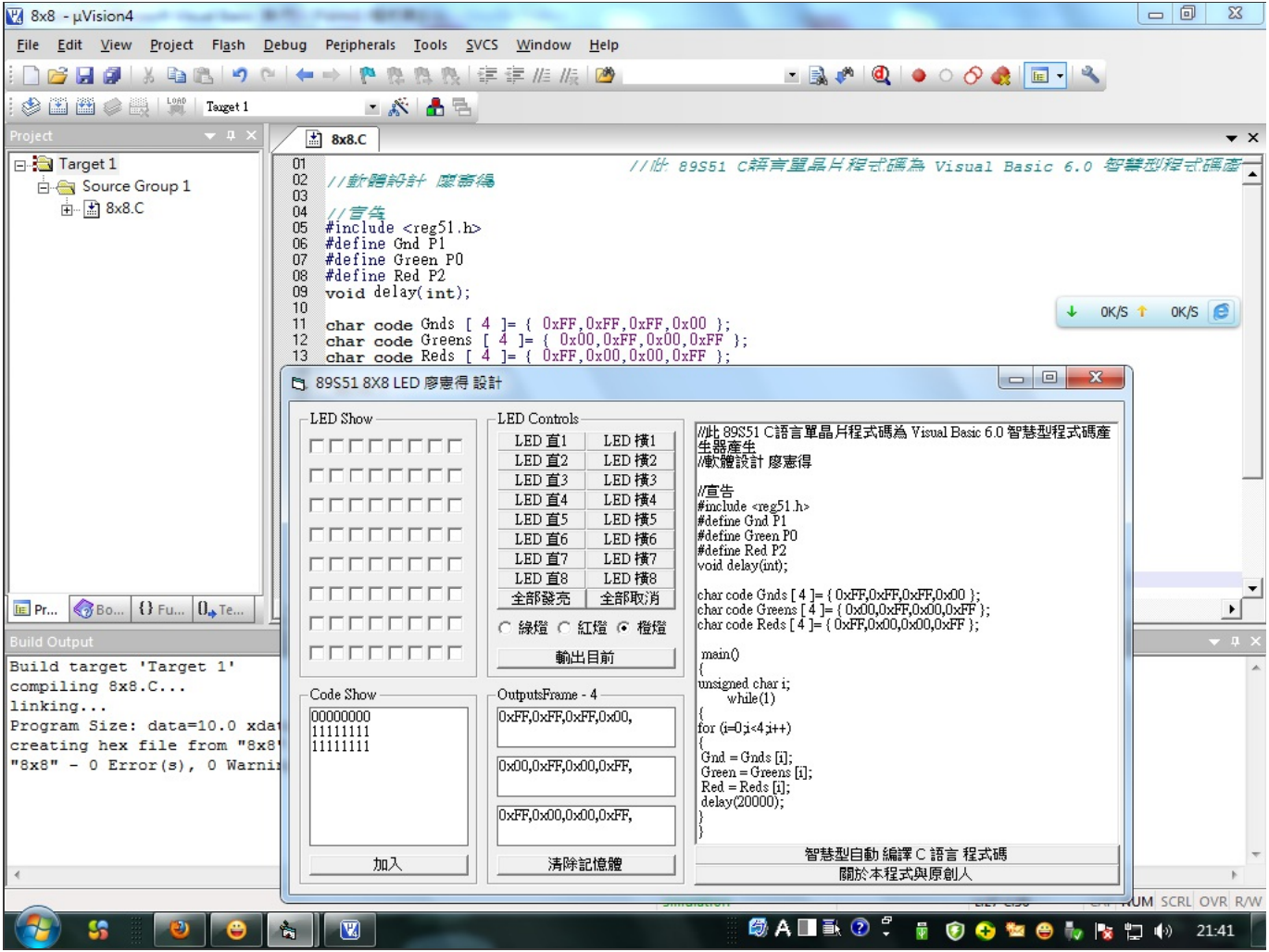
## 軟體用途與未來趨勢

其實開發這樣的軟體不只是可以運用在8X8 LED 上面，甚至我也撰寫過七段顯示器、蜂鳴器…等等的輔助軟體，相信不只是可以運用在C語言，也可以直接編譯成Java、C# 等等其他程式語言。甚至是運用在其他系統上，利用某個語言來編譯另外一個語言的概念可以衍生出更多讓開發變的更方便更便利的輔助軟體。

未來趨勢當然不只是運用在單晶片、專題上面，甚至可能連手機遊戲、APP，在日後都有相關的輔助設計軟體產生。這種〝爲了讓開發變得更便利〞的輔助軟體，在設計上也需要許多字串處理的巧思，只需要透過一些簡易的硬體演算法，像是在什麼情況中給0或給1等等，在設計此類型的輔助軟體就顯得相當容易！

## 使用效果


智慧型自動編譯 C 語言 VB 6.0 8X8 LED To Keil C Code For 89S51 軟體設計：廖憲得( De.com@msn.com )
LED Show
LED Controls
LED 直1
LED 直2
LED 直3
LED 直4
LED 直5
LED 直6
LED 直7
LED 直8
LED 橫1
LED 橫2
LED 橫3
LED 橫4
LED 橫5
LED 橫6
LED 橫7
LED 橫8
全部發亮
全部取消
綠燈
紅燈
橙燈
輸出目前
Code Show
00001111
11111111
00011111
加入
OutputsFrame - 16
0x7F,0x3F,0x2F,0xF,0xE,0xA,0x08,0x00,0x00,0x00,0x0
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x
清除記憶體
//此 89S51 C語言單晶片程式碼為 Visual Basic 6.0 智慧型程式碼產生器產生
//軟體設計 廖憲得
//宣告
#include <reg51.h>
#define Gnd P2
#define Green P1
#define Red P0
void delay(int);
char code Gnds [ 16 ]= {
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x
00,0xF };
char code Greens [ 16 ]= {
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x
FF,0x1F };
char code Reds [ 16 ]= {
0x7F,0x3F,0x2F,0xF,0xE,0xA,0x08,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,
0xFF };
main()
{
unsigned char i;
while(1)
{
for (i=0;i<16;i++)
{
Gnd = Gnds [i];
智慧型自動 編譯 C 語言 程式碼
軟體設計：大明高中 廖憲得 ( MicrosoftDe )
Programmer De (Facebook : http://www.facebook.com/MicrosoftDes Msn: De.com@msn.com )


8x8 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Target 1
Source Group 1
8x8.C
8x8.C
//此 89S51 C語言單晶片程式碼為 Visual Basic 6.0 智慧型程式碼產
//軟體設計 廖憲得
//宣告
#include <reg51.h>
#define Gnd P1
#define Green P0
#define Red P2
void delay(int);
char code Gnds [ 4 ]= { 0xFF,0xFF,0xFF,0x00 };
char code Greens [ 4 ]= { 0x00,0xFF,0x00,0xFF };
char code Reds [ 4 ]= { 0xFF,0x00,0x00,0xFF };
0K/S 0K/S
89S51 8X8 LED 廖憲得 設計
LED Show
LED Controls
LED 直1 LED 橫1
LED 直2 LED 橫2
LED 直3 LED 橫3
LED 直4 LED 橫4
LED 直5 LED 橫5
LED 直6 LED 橫6
LED 直7 LED 橫7
LED 直8 LED 橫8
全部發亮 全部取消
綠燈 紅燈 橙燈
輸出目前
//此 89S51 C語言單晶片程式碼為 Visual Basic 6.0 智慧型程式碼產生器產生
//軟體設計 廖憲得
//宣告
#include <reg51.h>
#define Gnd P1
#define Green P0
#define Red P2
void delay(int);
char code Gnds [ 4 ]= { 0xFF,0xFF,0xFF,0x00 };
char code Greens [ 4 ]= { 0x00,0xFF,0x00,0xFF };
char code Reds [ 4 ]= { 0xFF,0x00,0x00,0xFF };
main()
{
unsigned char i;
while(1)
{
for (i=0;i<4;i++)
{
Gnd = Gnds [i];
Green = Greens [i];
Red = Reds [i];
delay(20000);
}
}
智慧型自動 編譯 C 語言 程式碼
關於本程式與原創人
Code Show
00000000
11111111
11111111
加入
OutputsFrame - 4
0xFF,0xFF,0xFF,0x00,
0x00,0xFF,0x00,0xFF,
0xFF,0x00,0x00,0xFF,
清除記憶體
Build Output
Build target 'Target 1'
compiling 8x8.C...
linking...
Program Size: data=10.0 xdat
creating hex file from "8x8"
"8x8" - 0 Error(s), 0 Warni
NUM SCRL OVR R/W
21:41

## 部分程式碼與註解

```
' 以下開始皆爲編譯成 C 語言的核心資料字串處理部分
' (KeilC 爲輸出的 TextBox 文字方塊)
KeilC = KeilC & "//此 89S51 C語言單晶片程式碼爲 Visual Basic 6.0 智慧型程式碼產生器產生" & vbCrLf
KeilC = KeilC & "//軟體設計 廖憲得" & vbCrLf & vbCrLf
KeilC = KeilC & "//宣告" & vbCrLf
KeilC = KeilC & "#include <reg51.h>" & vbCrLf
KeilC = KeilC & "#define OutputP0 P0" & vbCrLf
KeilC = KeilC & "#define OutputP1 P1" & vbCrLf
KeilC = KeilC & "#define OutputP2 P2" & vbCrLf
KeilC = KeilC & "#define OutputP3 P3" & vbCrLf
KeilC = KeilC & "void delay(int);" & vbCrLf & vbCrLf
' OutputsFrame.Tag 爲 Visual Basic 中設定的 N 個動作存在 OutputsFrame.Tag 中 利用來當動態陣列數用
' 而 VBOutputs0、VBOutputs1、VBOutputs2、VBOutputs3 則是分別需要輸出的資料 (部分爲 16進制 )
KeilC = KeilC & "char code OutTempP0 [ " & OutputsFrame.Tag & " ]= { " & VBOutputs0 & " }; " & vbCrLf
KeilC = KeilC & "char code OutTempP1 [ " & OutputsFrame.Tag & " ]= { " & VBOutputs1 & " }; " & vbCrLf
KeilC = KeilC & "char code OutTempP2 [ " & OutputsFrame.Tag & " ]= { " & VBOutputs2 & " }; " & vbCrLf
KeilC = KeilC & "char code OutTempP3 [ " & OutputsFrame.Tag & " ]= { " & VBOutputs3 & " }; " & vbCrLf

KeilC = KeilC & " main()" & vbCrLf
```

```
KeilC = KeilC & "{" & vbCrLf
KeilC = KeilC & "unsigned char i;" & vbCrLf
KeilC = KeilC & "        while(1)" & vbCrLf
KeilC = KeilC & "{" & vbCrLf
KeilC = KeilC & "for (i=0;i<" & OutputsFrame.Tag & ";i++)" & vbCrLf
KeilC = KeilC & "{" & vbCrLf
' OutputP0、OutputP1、OutputP2、OutputP3 爲 89S51 單晶片的 PX 輸出變數
' OutTempP0、OutTempP1、OutTempP2、OutTempP3 則爲上面自動建立的陣列
' 原理就是把同一系列的一個所有動作自動存放到一個陣列當中，而要呼應這個動作就只需要做以下的呼
KeilC = KeilC & " OutputP0 = OutTempP0 [i];" & vbCrLf
KeilC = KeilC & " OutputP1 = OutTempP1 [i];" & vbCrLf
KeilC = KeilC & " OutputP2 = OutTempP2 [i];" & vbCrLf
KeilC = KeilC & " OutputP3 = OutTempP3 [i];" & vbCrLf

' delay 爲 C 語言中的延遲副程式 裡面的數字 20000 是拿來控制執行速度的
KeilC = KeilC & " delay(20000);" & vbCrLf

KeilC = KeilC & "}" & vbCrLf
KeilC = KeilC & "}" & vbCrLf
KeilC = KeilC & "}" & vbCrLf & vbCrLf
KeilC = KeilC & "//副程式" & vbCrLf
```

```
KeilC = KeilC & "void delay(int x)" & vbCrLf
KeilC = KeilC & "{" & vbCrLf
KeilC = KeilC & "    int i;" & vbCrLf
KeilC = KeilC & "    for (i=0;i<x;i++);" & vbCrLf
KeilC = KeilC & "}" & vbCrLf

' 哪一些要接地  設 P0 (OutTempP0) 要爲控制是否接地的資料輸出位置
For i = 7 To 0 Step -1
        Nows = i
        OpenOff = False
    For J = 7 To 0 Step -1
        If LED(Nows).Value = 1 Then OpenOff = True
        Nows = Nows + 8
    Next J
        If OpenOff Then
            P0 = P0 & "1" ' 輸出 1 = 接地
        Else
            P0 = P0 & "0" ' 輸出 0 = 導通
        End If
Next i
' 以上皆爲演算過程，這部分可以跳過，只要依照您的條件設定輸出 0 或者 1 即可
```

```
    OutTempP0 = OutTempP0 & P0 & ","
' OutTempP0 爲 P0 的資料輸出暫存變數 也是從這裡把資料送到 C 語言的陣列當中
```

(本文作者爲廖憲得, 聯絡方式爲 Facebook: http://www.facebook.com/MicrosoftDes, MSN: De.com@msn.com)

# 雜誌訊息

## 授權聲明

本雜誌採用 創作共用的「姓名標示、非商業性、相同方式分享」授權，其中許多內容來自於維基百科與開放原始碼社群，若您想要修改本書產生衍生著作時，至少應該遵守下列授權條件：

1. 標示原作者姓名
2. 不可以作爲商業用途。
3. 採用「姓名標示-非商業性-相同方式分享」的方式公開衍生著作。
4. 不得去除公益捐贈的相關描述。

## 讀者訂閱

程式人雜誌是一個結合「開放原始碼與公益捐款活動」的雜誌，簡稱「開放公益雜誌」。開放公益雜誌本著「讀書做善事、寫書做公益」的精神，我們非常歡迎程式人認養專欄、或者捐出您的網誌，如果您願意成爲本雜誌的專欄作家，請加入 程式人雜誌社團 一同共襄盛舉。

我們透過發行這本雜誌，希望讓大家可以讀到想讀的書，學到想學的技術，同時也讓寫作的朋友的作品能

產生良好價值 – 那就是讓讀者根據雜誌的價值捐款給慈善團體。 讀雜誌做公益也不需要有壓力，您不需要每讀一本就急著去捐款，您可以讀了十本再捐，或者使用固定的月捐款方式，當成是雜誌訂閱費，或者是季捐款、一年捐一次等都 OK！甚至是單純當個讀者我們也都很歡迎！ 本雜誌每期參考價：NT 50 元，如果您喜歡本雜誌，請將書款捐贈公益團體。例如可捐贈給「羅慧夫顱顏基金會 彰化銀行(009) 帳號：5234-01-41778-800」。(若匯款要加註可用「程式人雜誌」五個字)

想訂閱本雜誌的讀者，請按 我想訂閱程式人雜誌 連結並填寫表單，我們會在每一期雜誌出刊時寄送通知與下載網址到您的信箱。

## 投稿須知

給專欄寫作者： 做公益不需要有壓力。如果您願意撰寫專欄，您可以輕鬆的寫，如果當月的稿件出不來，我們會安排其他稿件上場。

給網誌捐贈者： 如果您沒時間寫專欄或投稿，沒關係，只要將您的網誌以 創作共用的「姓名標示、非商業性、相同方式分享」授權 並通知我們，我們會自動從中選取需要的文章進行編輯，放入適當的雜誌當中出刊。

給文章投稿者： 程式人雜誌非常歡迎您加入作者的行列，如果您想撰寫任何文章或投稿，請用 markdown 或 LibreOffice 編輯好您的稿件，並於每個月 25 日前投稿到程式人雜誌社團 的檔案區，我們會盡可能將稿件編入隔月1號出版程式人雜誌當中，也歡迎您到社團中與我們一同討論。

如果您要投稿給程式人雜誌，請盡可能使用以下兩種方式：

1. 使用 markdown 格式：markdown 的撰寫格式請參考 markdown樣版
2. 使用 LibreOffice odt 格式：odt 格式請下載 odt樣版。

關於以上兩種格式的寫法請下載 template.zip 以便瞭解寫作的方式與細節，請下載後閱讀 submit.md 檔案。

我們目前的編輯流程是用 pandoc 軟體將 markdown 轉換成 htm 與 epub 檔，然後再用 calibre 軟體中的 ebook-conver 指令將 epub 轉換爲 pdf 與 mobi 檔案，如此就可以得到 htm, epub, pdf, mobi 等四種版本，而針對 pdf 檔我們還進一步區分爲 A4 版與 ipad 版，以方便讀者根據需求自行取用。

## 參與編輯

您也可以擔任程式人雜誌的編輯，甚至創造一個全新的公益雜誌，我們誠摯的邀請您加入「開放公益出版」的行列，如果您想擔任編輯或創造新雜誌，也歡迎到 程式人雜誌社團 來與我們討論相關事宜。

## 公益資訊

| 公益團體 | 聯絡資訊 | 服務對象 | 捐款帳號 |
| --- | --- | --- | --- |

| | | | |
|---|---|---|---|
| 財團法人羅慧夫顱顏基金會 | http://www.nncf.org/<br>lynn@nncf.org<br>02-27190408分機232 | 顱顏患者 (如唇顎裂、小耳症或其他罕見顱顏缺陷） | 銀行：009彰化銀行民生分行<br>帳號：5234-01-41778-800 |
| 社團法人台灣省兒童少年成長協會 | http://www.cyga.org/<br>cyga99@gmail.com<br>04-23058005 | 單親、隔代教養.弱勢及一般家庭之兒童青少年 | 銀行：新光銀行<br>戶名：台灣省兒童少年成長協會<br>帳號：103-0912-10-000212-0 |