

# ARDUINO 基礎教學(一)



瑞司創科技教學專用-智光電子科

# Arduino 是什麼” 碗糕 “

**Restrong**

瑞司創科技有限公司

---

- Arduino是什麼？

是一個開放源始程式碼的單晶片微控制系統，它使用了Atmel AVR單晶片，採用了基於開放源始程式碼的軟硬體平台，構建於開放源始程式碼simple I/O 接口板，並且具有使用類似Java，C 語言的Processing / Wiring 開發環境

open-source electronics prototyping platform是什麼”  
東東”

open source 開放源始碼-包含硬軟體  
electronics prototyping platform  
電子電路原型平台

- [www.arduino.cc](http://www.arduino.cc) - 發源地
- too many to name here.族繁不及備載-將跨越x86  
目前已進化到ARM Cortex(32Bit)

<http://wiesel.ece.utah.edu/redmine/projects/wesec/wiki>  
參考資料 <http://coopermaa2nd.blogspot.tw/2011/05/arduino.html>

本投影片內容與資料來源皆有引用到原廠網站與各大代理商資料

# Arduino 開發平台與Atmel認證

**Restrong**

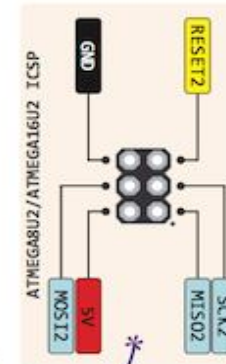
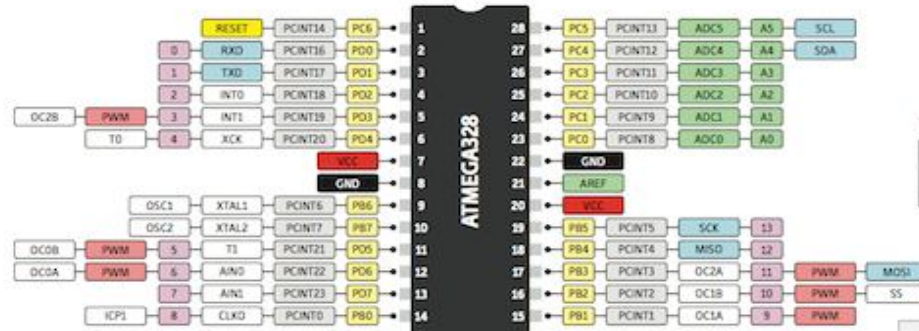
瑞司創科技有限公司

- Arduino UNO – 8 Bit AVR MCU CHIP
- 上課用板

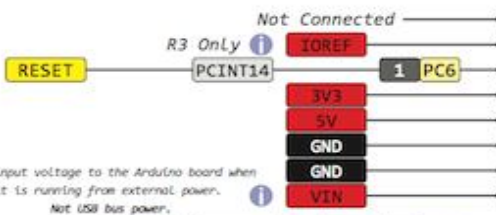


# THE DEFINITIVE ARDUINO UNO PINOUT DIAGRAM

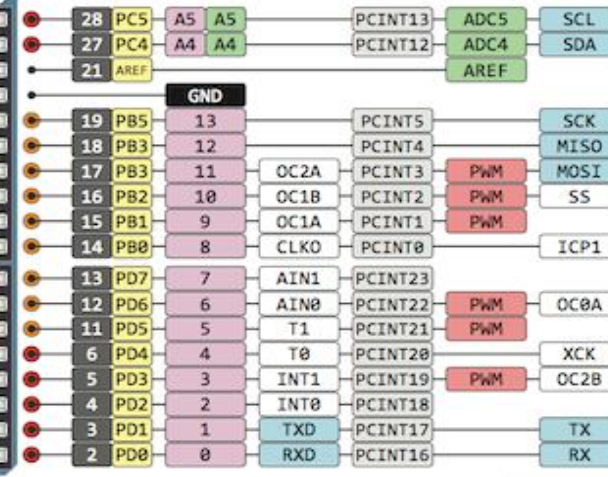
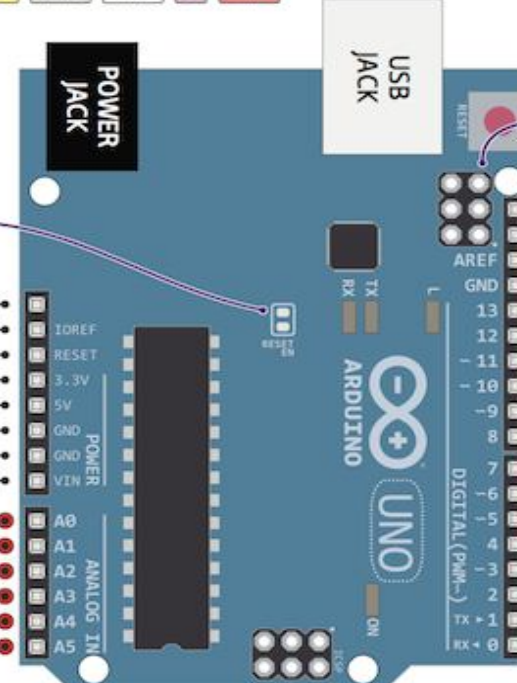
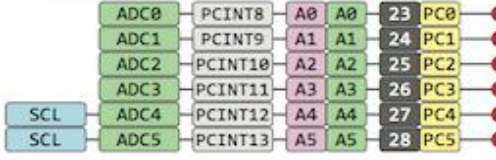
⚠ Absolute max per pin 48mA recommended 20mA  
⚡ Absolute max 200mA for entire package



Cut to disable the auto-reset



The input voltage to the Arduino board when it is running from external power. Not USB bus power.



R3 Only

Connected to the ATmega and used for USB program and communicating with it

- GND
- Power
- Control
- Physical Pin
- Port Pin
- Pin Function
- Digital Pin
- Analog Related Pin
- PWM Pin
- Serial Pin
- IDE
- Source Total 150mA

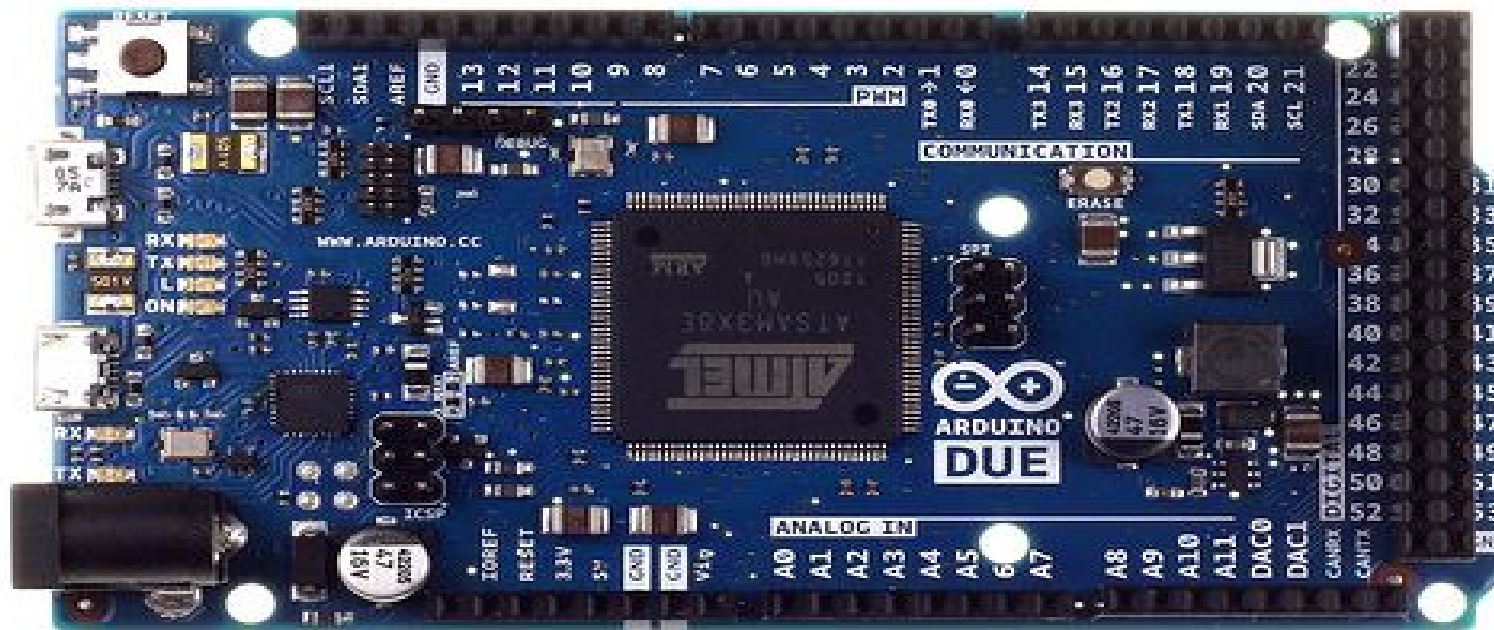
- MCU:ATmega328-16 MHz
- 32KBytes Flash,SRAM 2KBytes,EEPROM 1K
- 操作電壓5V,輸入電壓(建議) 7V-12V
- 數位 I/O 接腳: 14支
- 類比 I/O 接腳: 6支
- 每一支接腳的直流電流: 40mA
- 在3.3V時每一支接腳的直流電流: 50mA
- SPI,serial,PWM 界面



- Arduino ZERO – 32 Bit Cortex M0+ MCU
- Atmel SAMD21 MCU, Clock Speed 48MHz



- Arduino Due – 32 Bit Cortex M3 MCU
- Atmel SAM3X8E Cortex-M3 MCU, Clock Speed 84MHz, Flash 512kbytes.





Intel Galileo 採用了 Intel® Quark SoC X1000 的32 位元、單核心/單執行緒、Pentium® 指令集架構 (ISA) 相容的處理器，運作時脈最高達 400MHz。

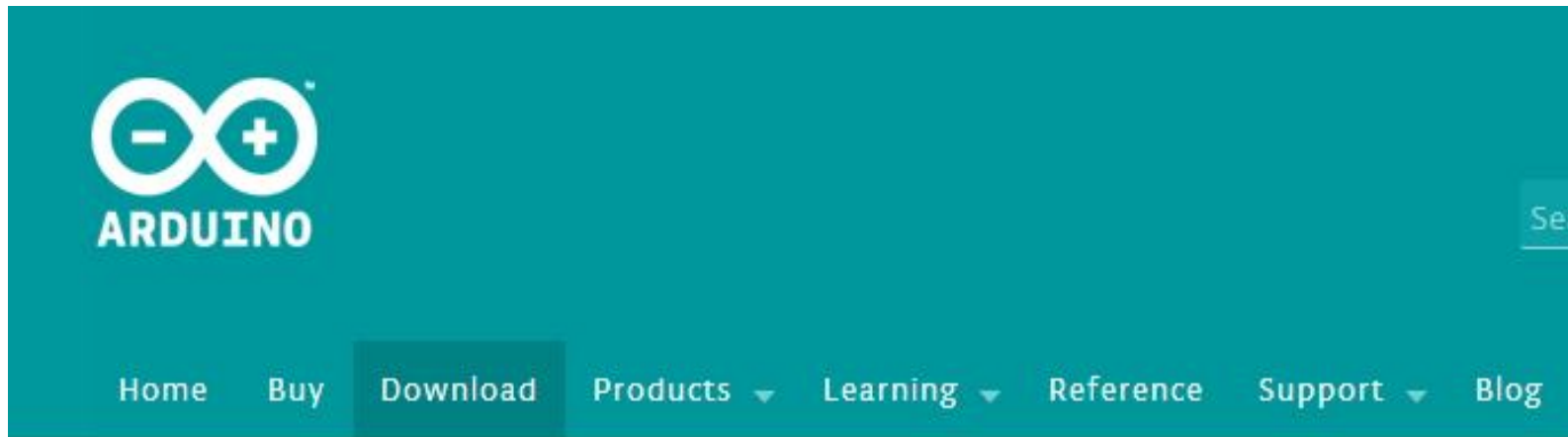


# Arduinio 入門

**Restrong**

瑞司創科技有限公司

- 請至[www.arduino.cc](http://www.arduino.cc)下載



- 點選Download後

- 下載 Arduino 1.0.5的windows版本

## Arduino IDE

### Arduino 1.0.5

#### Download

Arduino 1.0.5 ([release notes](#)), hosted by [Google Code](#):

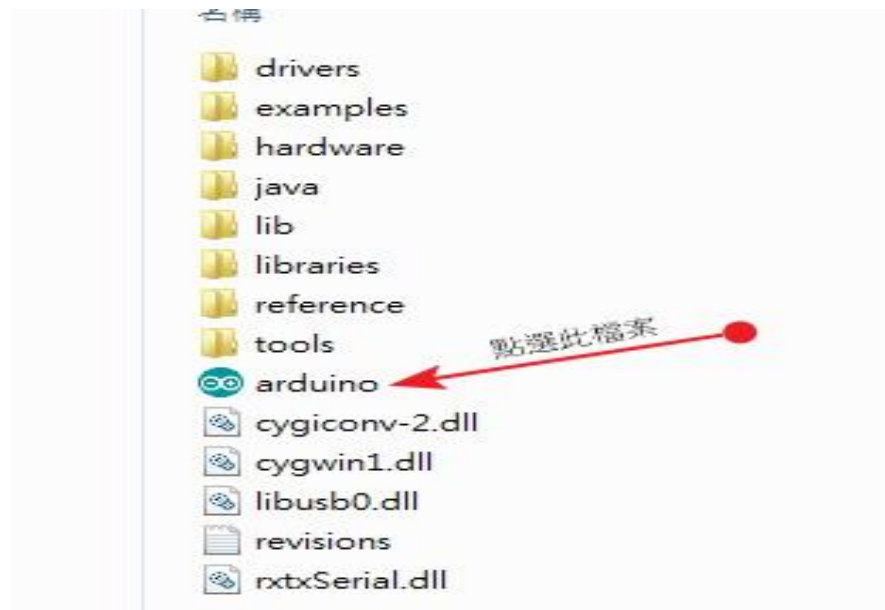
NOTICE: Arduino Drivers have been updated to add support for Windows 8.1, you can download the updated IDE (version 1.0.5-r2 for Windows) from the download links below.

- [Windows Installer](#), [Windows ZIP file](#) (for non-administrator install)
- [Mac OS X](#)
- [Linux: 32 bit](#), [64 bit](#)
- [source](#)

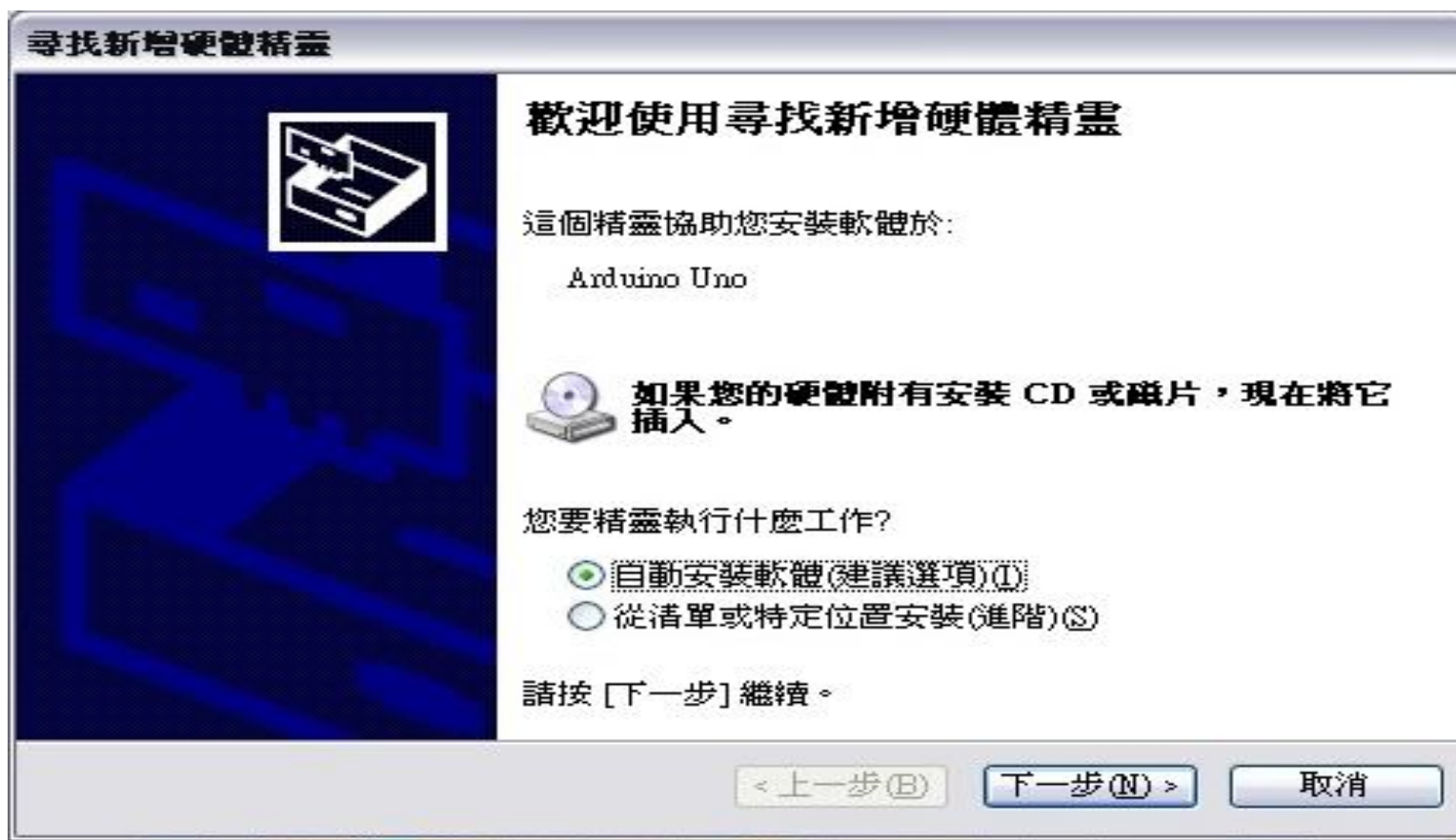
#### Next steps

[Getting Started](#)  
[Reference](#)  
[Environment](#)  
[Examples](#)  
[Foundations](#)  
[FAQ](#)

- 將壓縮檔案解壓縮至D:\
- 執行D:\arduino-1.0.5-r2\arduino
- 即可安裝軟體



- 安裝完畢後插入usb cable連接電腦會出現





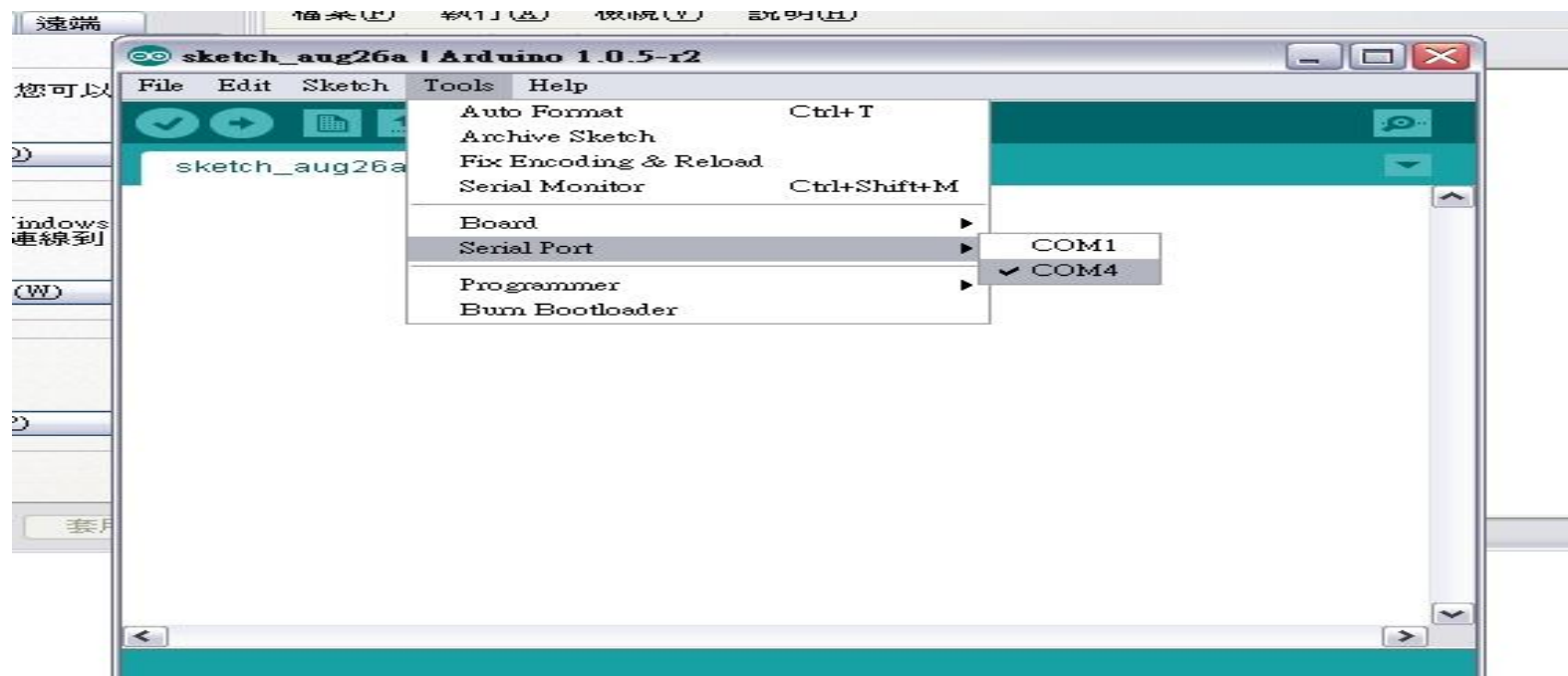
搜尋相關的驅動程式:



- 安裝完畢後將會顯示com埠在那個位置: 本機為COM4

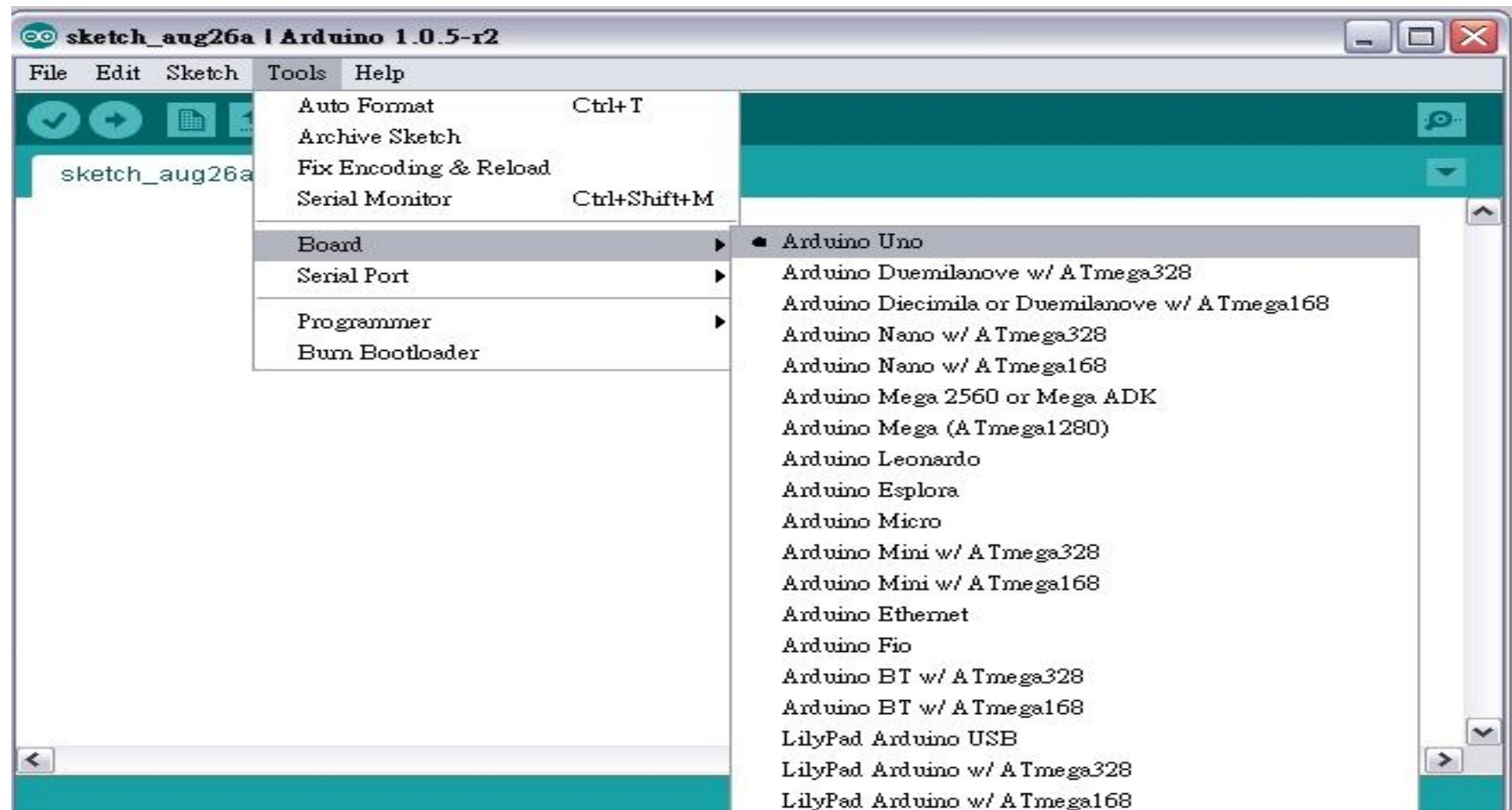


- 調整IDE環境內的開始板使用那一片與COM埠的設定值後就進入開發程式的階段(Tools – Board選擇UNO,與Serial選擇COM4



# Restrong

瑞司創科技有限公司



## 課程內容

- 
- (一). Arduino I/O 輸出
- (二). Arduino I/O 輸入
- (三). Arduino serial PORT
- (四). Arduino A/D
- (五). 綜合範例



# Arduinio 程式基本架構

**Restrong**

瑞司創科技有限公司

- 點選功能表File > Examples > 01.Basics > BareMinimum，載入BareMinimum範例程式
- 仔細觀察程式中的內容會發現，**Arduino**程式主要由**setup()**及**loop()**兩個函數(**function**)區塊所組成，茲分述如下：
- **Setup()**函數語法：  
  

```
void setup()  
{  
  
    setup()程式段中用來進行pin 接腳類型的設定、變數的初始化等，這個函數區塊中的程式只有在程式開始執行時以及程式重新執行(reset)時會被呼叫執行一次  
  
}
```

- **Loop()**函數語法：

```
void loop()
```

```
{
```

執行完**setup()**函數中的所有設定及程式後，系統便會將控制權交給**loop()**函數，自此之後將不斷的重覆執行在**loop()**函數區塊中的程式，除非Arduino 板的電力耗盡，否則撰寫在**loop()**函數區塊中的程式碼將會反覆不斷的被執行，永不停止.

```
}
```

- Arduino I/O 輸出
- LED,7段顯示器的應用

# Ardunio 第一支程式

**Restrong**

瑞司創科技有限公司

- 發光二極體 ( Light-Emitting Diode , 縮寫 : LED )
- 此種電子元件早在1962年出現，早期只能夠發出低光度的紅光，被惠普買下專利後當作指示燈利用
- 發光二極體只能夠往一個方向導通（通電），叫作順向偏壓（順向偏壓），當電流流過時，電子與電洞在其內重合而發出單色光，這叫電致發光效應
- 使用壽命長 - 且不因連續閃爍而影響其壽命
- 有機發光二極體OLED。其發光原理跟發光二極體一樣，不同之處是其發光物半導體是有機化合物（有機半導體），例如有機聚合物等。OLED製程簡單，成本也較低，可以用印刷等廉價生產方法製造

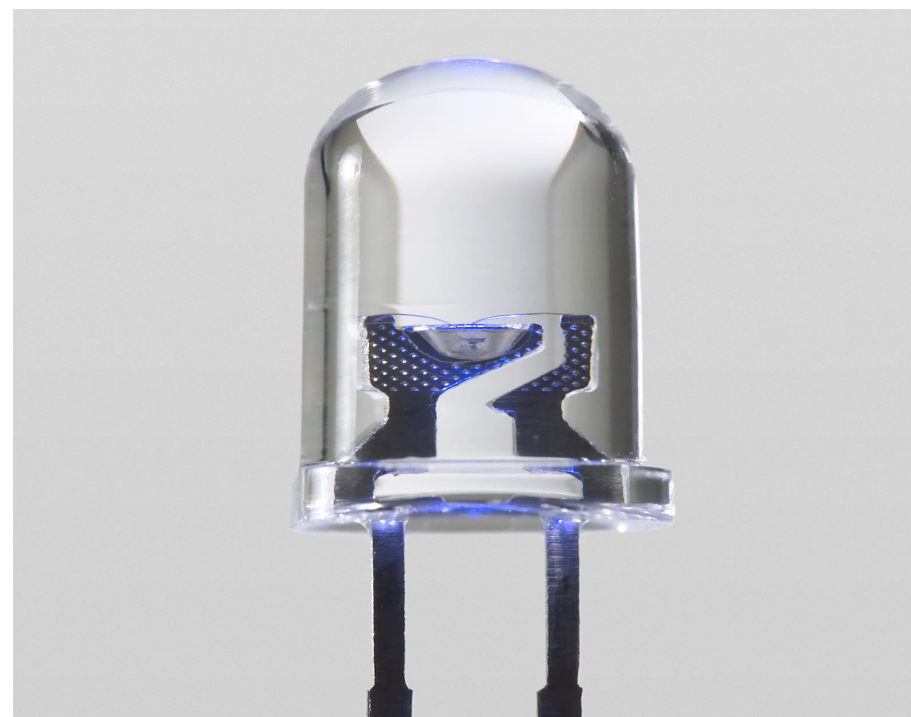
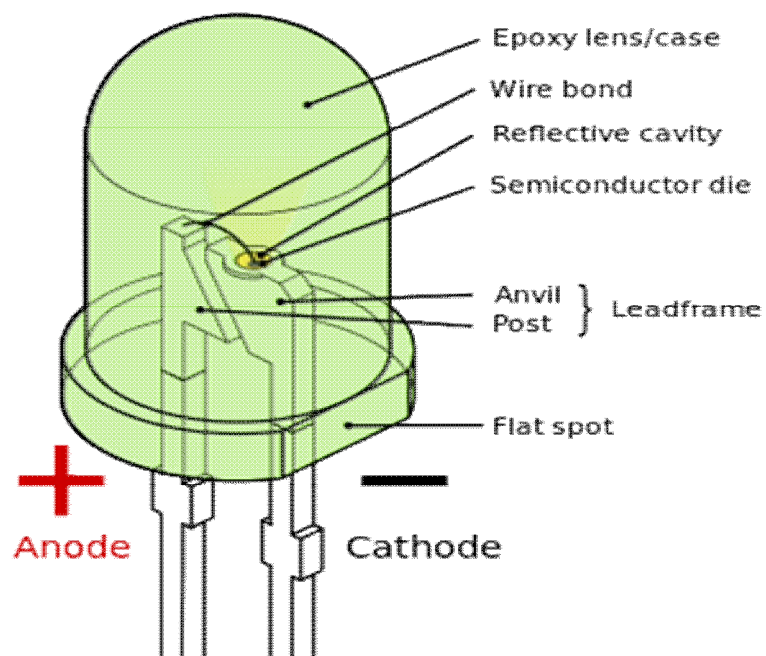
- LED為PN二極體的一種,LED長腳為P型(Positive,正)亦稱為陽極(anode,簡稱A),短腳為N型,亦稱為陰極(cathode,簡稱k).
- LED應用上在5V電壓下順向電流大約10-40mA之間,須要串接一顆限流電阻約47-120 $\Omega$ 左右,若電阻值太小流過LED的電流太大將會燒毀LED,反之若阻值太大,LED的亮度將會變暗.



# Restrong

瑞司創科技有限公司

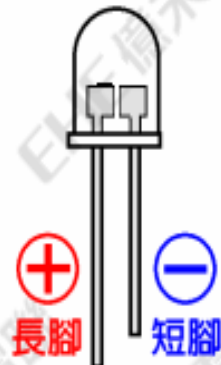
●



- \* 圖片引用億禾電聯

## 接腳圖

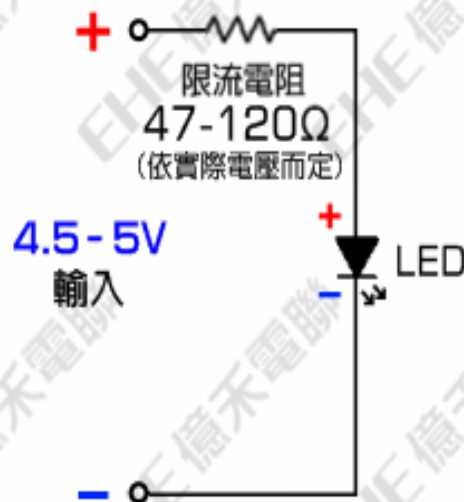
LED之長腳為正



※LED極性  
請以長短腳為準

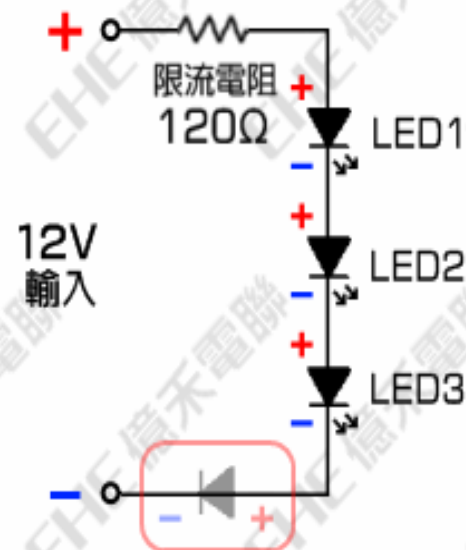
## LED接法 (基本上與一般白、藍光接法相同)

1顆LED之接法

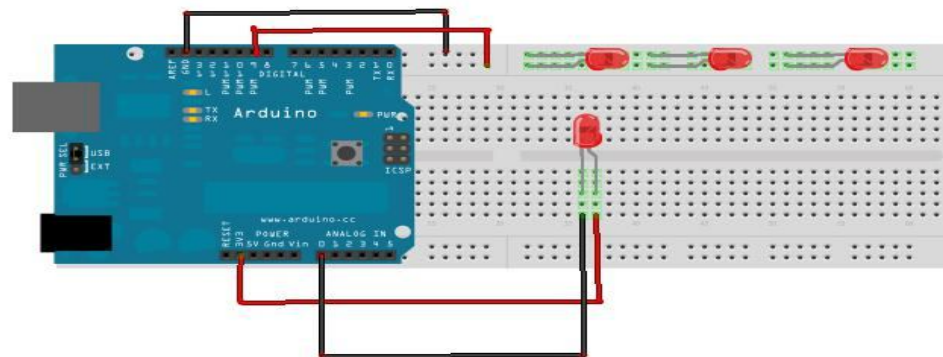


3顆LED為一組之接法

※如用於車用 需串接防逆壓二極體



- A0接線到LED的正極(+),再將LED的負極接到UNO的GND



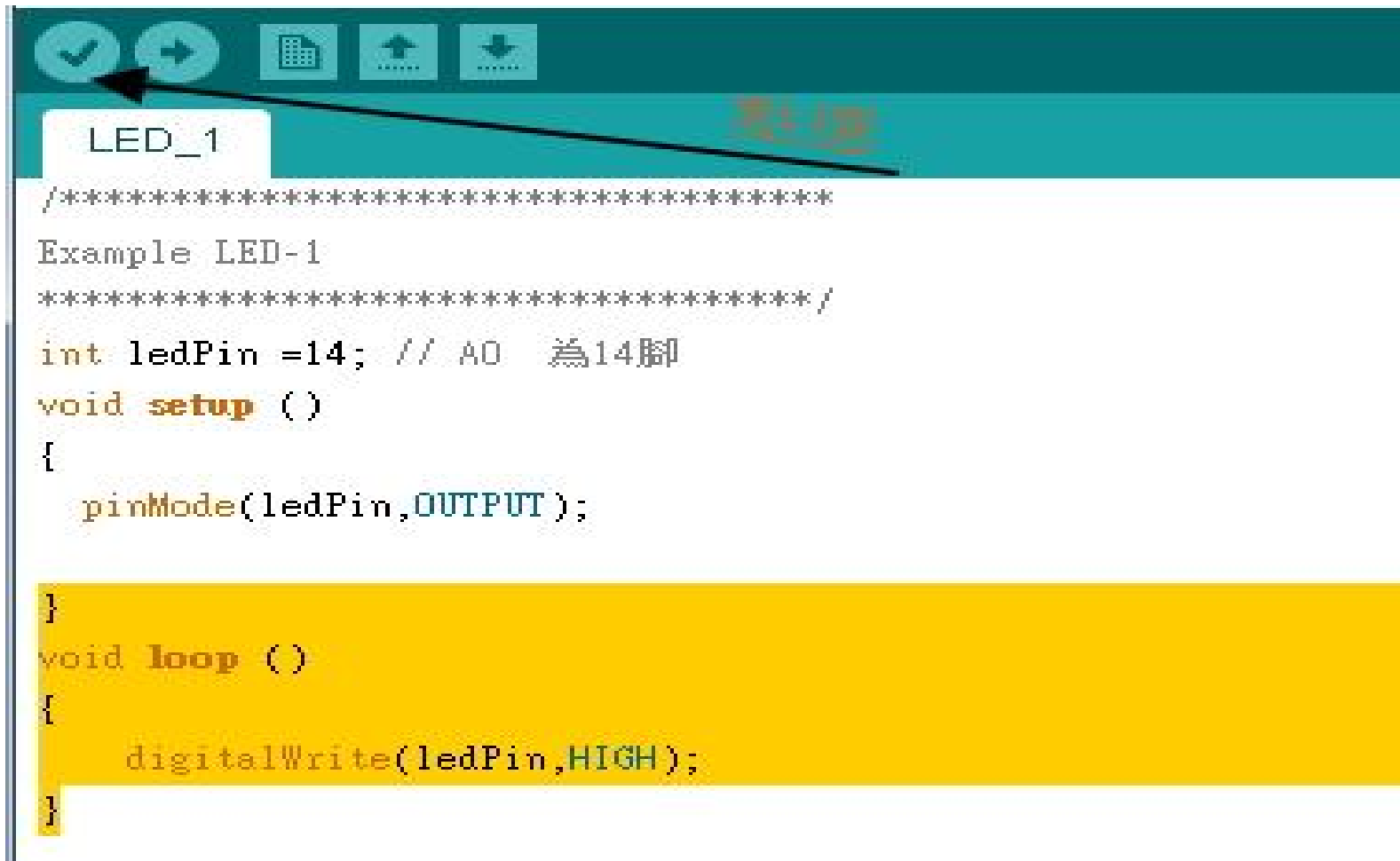
- 步驟一:點選功能表File > Examples > 01.Basics > BareMinimum , 載入BareMinimum範例程式
- 第一行請打 `int ledPin =14; // A0 為14腳`
- 步驟二:請在**`void setup()`**內加入
- `void setup ()`  
{  
    `pinMode(ledPin,OUTPUT);`  
  
}

- 步驟三:請在**void loop()**內加入

```
void loop ()  
{  
  digitalWrite(ledPin,HIGH);  
}
```

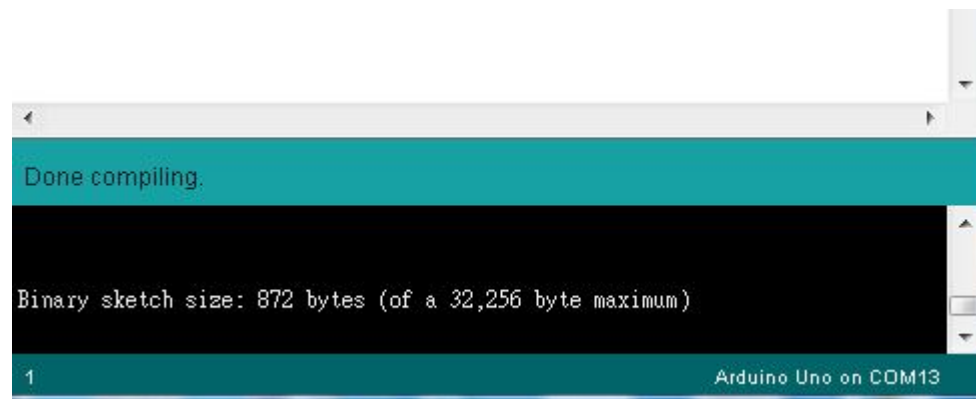
- 步驟四: 執行編譯,觀看是否出現錯誤訊息





```
/*  
Example LED-1  
*/  
int ledPin = 14; // A0 為14腳  
void setup ()  
{  
  pinMode(ledPin,OUTPUT);  
  
}  
void loop ()  
{  
  digitalWrite(ledPin,HIGH);  
}
```

- IDE環境中Verif進行程式的編譯,check是否語法錯誤與總共使用多少的容量空間



- 再將程式碼下載到UNO的板子MCU Flash中



- 步驟五: 下載到UNO板子觀看LED狀態



# Arduinio 變數與常數-資料結構

**Restrong**

瑞司創科技有限公司

- `int ledPin = 14; // A0`
- 利用變數與常數的資料型態來取代記憶體的真实位址,至於安排在那一個位址則由編譯器(IDE)來做分配  
`int` – 資料型態 ; `ledPin`- 變數  
也就是宣告資料型態為整數名字為`ledPin`的變數,其值為14

在Arduino中資料型態包含布林,整數與浮點數三種.  
整數資料型態包含: `char`(字元),`int`(整數),`long`(長整數)  
再搭配上`signed`(有號數),`unsigned`(無號數),改變資料的範圍

- 浮點數資料型態包含float ,double兩種
- 布林(boolean)資料型態有兩種true(1)與false(0)

資料型態	位元數	範圍
int	16	-32768 ~ +32767
Unsigned int	16	0 ~ 65535
char	8	-128 ~ +127
Unsigned char	8	0 ~ 255

資料型態	位元數	範圍
byte	8	0 ~ 255
word	16	0 ~ 65535
float	32	-3.4028235E+38 ~ +3.4028235E+38
double	32	-3.4028235E+38 ~ +3.4028235E+38

資料型態	位元數	範圍
boolean	8	True(1),false(0)

- void 是指「無效」或「無意義」的意思，您無法使用 void 型態來宣告任何變數(癥結在於無法確知將要使用的記憶體大小為何，編譯器無法計算這個變數到底佔多大空間)。
- ```
void setup ()  
{  
  pinMode(ledPin,OUTPUT);  
}
```



- pinMode()函數:用來設定接腳模式(pin mode),指令格式為下:

pinMode(接腳的編號,模式);

其中接腳的編號即為mcu的接腳(pin),有類比接腳,數位接腳  
模式則為: 輸出模式或輸入模式,只能二選一.

輸出模式: OUTPUT 全部大寫,輸入模式: INPUT 全部大寫

注意: pinMode中的M為大寫,結束時請多加上" ; "

```
void loop ()  
{  
  digitalWrite(ledPin,HIGH);  
}
```

digitalWrite()函數:用來輸出腳位電位的指令,包含2種的電位,  
高電位(HIGH)或者 1, 低電位(LOW)或者  
為0,指令格式為下:

digitalWrite(接腳的編號,輸出訊號高或低);

```
int ledPin =14; // A0 為14腳
void setup ()
{
  pinMode(ledPin,OUTPUT); //14腳設定為輸出模式
}

void loop ()
{
  digitalWrite(ledPin,HIGH); // 14腳給它一個高電壓位準
}
```

## 範例2-填入相關指令

**Restrong**

瑞司創科技有限公司

- `/******`
- Example LED-2 填入相關指令
- `*****/`
- `int ledPinA0=14; // A0為14腳`
- `int ledPinA1=15; // A1 為15腳`
- `void setup ()`
- `{`
- `pinMode( ,OUTPUT); //將LED SET OUTPUT`
- `pinMode( , OUTPUT);`
- `}`
- `void loop ()`
- `{`
- `digitalWrite(ledPinA0, ); // 14腳給HIGH`
- `digitalWrite(ledPinA1, ); // 15腳給HIGH`
- `}`

# 範例3-加入延遲時間

**Restrong**

瑞司創科技有限公司

```
• /******  
• Example LED-3  
• *****/  
• int ledPinA0=14; // A0為14腳  
• int ledPinA1=15; // A1為14腳  
  
• void setup ()  
• {  
•   pinMode(ledPinA0,OUTPUT); //將LED SET OUTPUT  
•   pinMode(ledPinA1,OUTPUT);  
•  
• }  
• void loop ()  
• {  
•   digitalWrite(ledPinA0,HIGH); // 14腳給HIGH  
•   digitalWrite(ledPinA1,LOW); // 15腳給LOW  
•   delay(1000);  
•   digitalWrite(ledPinA0,LOW); // 14腳給LOW  
•   digitalWrite(ledPinA1,HIGH); // 15腳給HIGH  
• }
```

- `delay()`函數-維持之前的動作不改變或者延遲接下面的指令動作.
- `delay()` ; 即延遲多少毫秒(ms)
- `delayMicroseconds()`;即延遲多少微秒( $\mu$ s)

`delay(1000);` 即為1000ms,也就是1s(秒)

問題:上述程式如何讓2顆led持續閃爍

- 上述問題如何將程式碼縮短

## 範例4-霹靂燈

**Restrong**

瑞司創科技有限公司

- `/******`
- `Example LED-4`
- `*****/`

```
int ledPin[4] = {10,11,12,13};
int i;
int j;
void setup ()
{
  for(i=0;i<4;i++)
  {
    pinMode(ledPin[i],OUTPUT);
  }
}
```

- void loop ()
- {
- for (i=0;i<4;i++)
- {
- digitalWrite(ledPin[i],LOW);
- }
- for(j=0;j<4;j++)
- {
- digitalWrite(ledPin[j],HIGH);
- delay(1000);
- }
- if(j=4)
- j=0;
- }



- 關係運算子

|    |     |   |    |    |      |
|----|-----|---|----|----|------|
| == | 等於  | < | 小於 | <= | 小於等於 |
| != | 不等於 | > | 大於 | >= | 大於等於 |

# 範例5-霹靂燈改良程式

**Restrong**

瑞司創科技有限公司

- 一維陣列的應用
- 宣告格式:  
資料型態 陣列名稱[陣列大小]= { 初始0,初始1,...初始n-1};

- 簡單範例:

```
Void setup()
{
}
Void loop()
{
    int n;
    int a[4]={0,2,4,6};
}
```

- 執行結果:
- $a[0]=0, a[1]=2, a[2]=4, a[3]=6$
- 判斷式 for 迴圈的應用
- For(開始的數值;迴圈停止的條件;每次迴圈結束後的動作)
  - {
  - // 執行的內容
  - }

- 算數運算子

|    |    |          |                   |
|----|----|----------|-------------------|
| +  | 加法 | $a + b$  | a的內容值與b的內容值相加     |
| -  | 減法 | $a - b$  | a的內容值與b的內容值相減     |
| *  | 乘法 | $a * b$  | a的內容值與b的內容值相乘     |
| /  | 除法 | $a / b$  | a的內容值除以b的內容值取商數   |
| %  | 餘數 | $a \% b$ | a的內容值除以b的內容值後與餘數  |
| ++ | 遞增 | $a++$    | a的內容值加1,即 $a=a+1$ |
| -- | 遞減 | $a--$    | a的內容值減1,即 $a=a-1$ |

- 判斷式 if迴圈的應用
- if(判斷條件)  
    {  
        // 若為判斷的條件,執行程式  
    }

# 7段顯示器

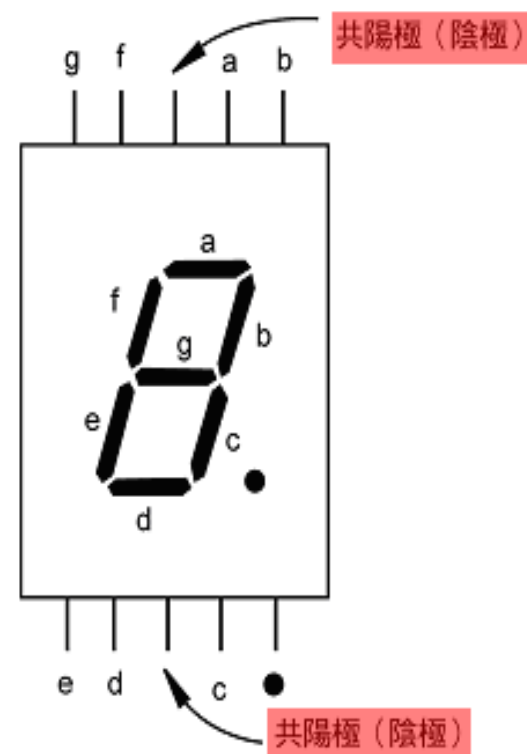
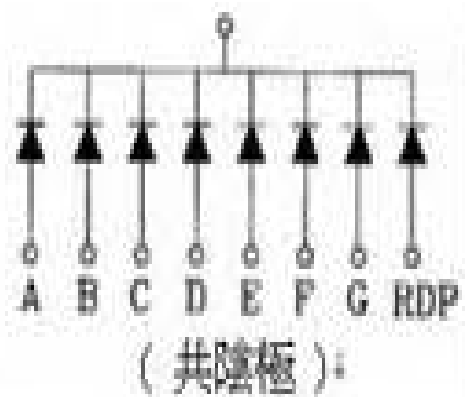
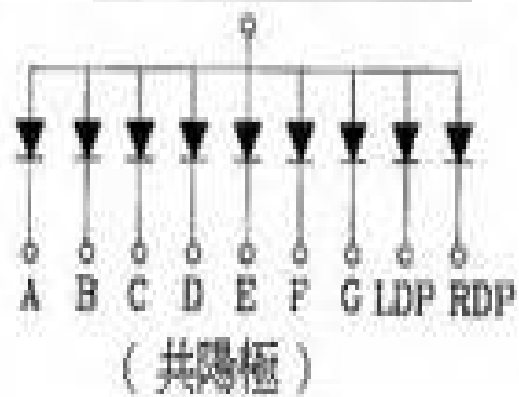
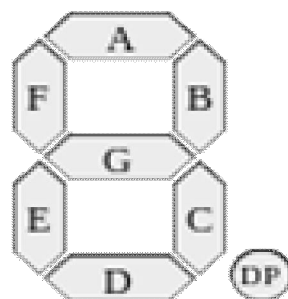
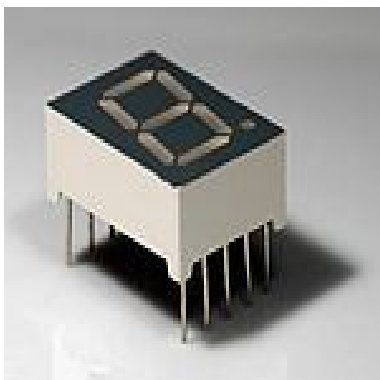
**Restrong**

瑞司創科技有限公司

- 7段顯示器（英語：Seven-segment display）為常用顯示數字的電子元件。因為藉由七個發光二極體以不同組合來顯示數字，所以稱為7段顯示器。
- 一般的七段顯示器擁有八個發光二極體用以顯示十進位0至9的數字，也可以顯示英文字母，包括十六進位和二十進位中的英文 A 至 F（b，d，i 為小寫，其他為大寫）
- 七段顯示器分為共陽極及共陰極，共陽極的七段顯示器的正極（或陽極）為八個發光二極體的共有正極，其他接點為獨立發光二極體的負極（或陰極），使用者只需把正極接電

# Restrong

瑞司創科技有限公司



- 7段顯示器的COM接到Digital 2接腳
- 7段顯示器的A接到Digital 3接腳
- 7段顯示器的B接到Digital 4接腳
- 7段顯示器的C接到Digital 5接腳
- 7段顯示器的D接到Digital 6接腳
- 7段顯示器的E接到Digital 7接腳
- 7段顯示器的F接到Digital 8接腳
- 7段顯示器的G接到Digital 9接腳



- `/******`
- Example 7 SEG-1
- Restrong Technology – Jeff Chen
- `*****/`
- `void setup ()`
- `{`
- `pinMode(2,OUTPUT); //將LED SET OUTPUT`
- `pinMode(3,OUTPUT); //將LED A SET OUTPUT`
- `pinMode(4,OUTPUT); //將LED B SET OUTPUT`
- `pinMode(5,OUTPUT); //將LED C SET OUTPUT`
- `pinMode(6,OUTPUT); //將LED D SET OUTPUT`
- `pinMode(7,OUTPUT); //將LED E SET OUTPUT`
- `pinMode(8,OUTPUT); //將LED F SET OUTPUT`
- `pinMode(9,OUTPUT); //將LED G SET OUTPUT`
- `}`

- void loop ()
- {
- digitalWrite(2,LOW); // COOM GND
- digitalWrite(3,HIGH); //
- digitalWrite(4,HIGH); //
- digitalWrite(5,HIGH); //
- digitalWrite(6,HIGH); //
- digitalWrite(7,HIGH); //
- digitalWrite(8,HIGH); //
- digitalWrite(9,HIGH); //
- }

- 上述的程式請編譯後執行觀看結果
- 若將digitalWrite(2,LOW); 改為HIGH後會有何狀態
- 設計一個由0,1,2---9的顯示值,由7段顯示器輸出

- `/******`
- Example 7 SEG-2
- Restrong Technology – Jeff Chen
- `*****/`
- `void setup ()`
- `{`
- `pinMode(2,OUTPUT); //將LED SET OUTPUT`
- `pinMode(3,OUTPUT); //將LED A SET OUTPUT`
- `pinMode(4,OUTPUT); //將LED B SET OUTPUT`
- `pinMode(5,OUTPUT); //將LED C SET OUTPUT`
- `pinMode(6,OUTPUT); //將LED D SET OUTPUT`
- `pinMode(7,OUTPUT); //將LED E SET OUTPUT`
- `pinMode(8,OUTPUT); //將LED F SET OUTPUT`
- `pinMode(9,OUTPUT); //將LED G SET OUTPUT`
- `}`

- void loop ()
- {
- digitalWrite(3,LOW); //
- digitalWrite(4,LOW); //
- digitalWrite(5,LOW); //
- digitalWrite(6,LOW); //
- digitalWrite(9,LOW); //
- digitalWrite(2,LOW); // COOM GND
- digitalWrite(4,HIGH); //
- digitalWrite(5,HIGH); //
- delay(2000);
- digitalWrite(4,LOW); //
- digitalWrite(5,LOW); //
- digitalWrite(2,LOW); // COOM GND
- digitalWrite(3,HIGH); //

- digitalWrite(4,HIGH); //
- digitalWrite(6,HIGH); //
- digitalWrite(7,HIGH); //
- digitalWrite(9,HIGH); //
- delay(2000);
- digitalWrite(2,LOW); // COOM GND
- digitalWrite(3,LOW); //
- digitalWrite(4,LOW); //
- digitalWrite(6,LOW); //
- digitalWrite(7,LOW); //
- digitalWrite(9,LOW); //
- digitalWrite(3,HIGH); //
- digitalWrite(4,HIGH); //
- digitalWrite(5,HIGH); //
- digitalWrite(6,HIGH); //
- digitalWrite(9,HIGH); //
- delay(2000);
- }

- 有沒有發現程式真的很長,打的很累
- 是否有其他方法將程式碼簡化
- 利用陣列方式是否可達成
- 何為陣列?下列範例可說明一切

- `/******`
- Example 7 SEG-3
- Restrong Technology – Jeff Chen
- `*****/`
- `int i=0;`
- `int j=0;`
- `void OFF()`
- `{`
- `digitalWrite(3,LOW);`
- `digitalWrite(4,LOW);`
- `digitalWrite(5,LOW);`
- `digitalWrite(6,LOW);`
- `digitalWrite(7,LOW);`
- `digitalWrite(8,LOW);`
- `digitalWrite(9,LOW);`
- `}`



- void setup ()
- {
- pinMode(2,OUTPUT); //將LED SET OUTPUT
- pinMode(3,OUTPUT); //將LED A SET OUTPUT
- pinMode(4,OUTPUT); //將LED B SET OUTPUT
- pinMode(5,OUTPUT); //將LED C SET OUTPUT
- pinMode(6,OUTPUT); //將LED D SET OUTPUT
- pinMode(7,OUTPUT); //將LED E SET OUTPUT
- pinMode(8,OUTPUT); //將LED F SET OUTPUT
- pinMode(9,OUTPUT); //將LED G SET OUTPUT
- digitalWrite(2,LOW);
- }

```
• void loop ()
• {
•
•   int number[4][6] =
•       { { 3,4,5,6,7,8},
•         { 4,5},
•         { 3,4,6,7,9},
•         { 3,4,5,6,9}
•       };
•   for(i=0;i<4;i++)
•   {
•       for(j=0;j<6;j++)
•
•       digitalWrite(number[i][j],HIGH);
•       delay(500);
•       OFF();
•
•   }
• }
```

- 二維陣列
- 宣告格式:
- 資料型態 陣列名稱[m][n] =
- {
- { 初始0,初始1,...初始n-1}, // 第0列
- { 初始0,初始1,...初始n-1}, // 第1列
- :
- { 初始0,初始1,...初始n-1}, // 第m-1列
- }

- 簡單範例:

```
Void setup()
```

```
{ }
```

```
Void loop()
```

```
{
```

```
    int m,n;
```

```
    int a[3],[2]=
```

```
{ (0,1), //第0列
```

```
    (2,3), //第1列
```

```
    (4,5) } ; //第2列
```

- 執行結果:
- $a[0][0]=0, a[0][1]=1, a[1][0]=2, a[1][1]=3$
- $a[2][0]=4, a[2][1]=5$
- 有沒有發現這樣的做法將使用多支I/O腳位,如何減少腳位  
請各位思考一下