

1

- 1 [什麼是 Arduino](#)
- 2 [Arduino 的資源](#)
- 3 [Arduino 開發環境](#)

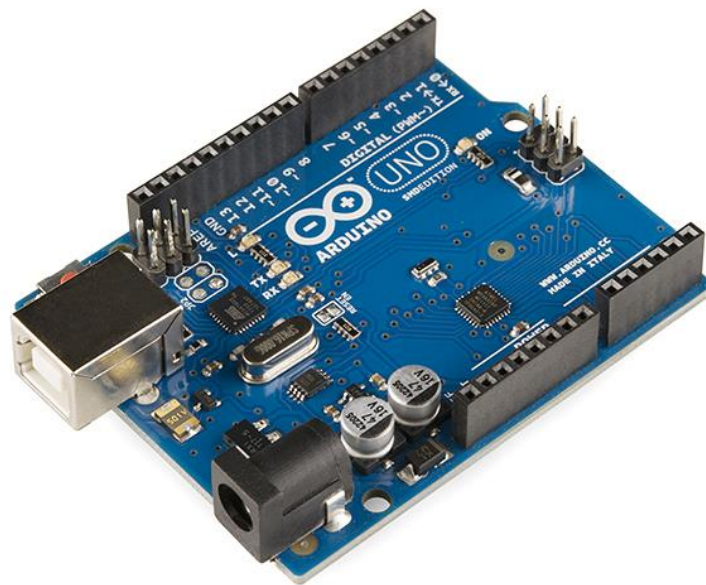
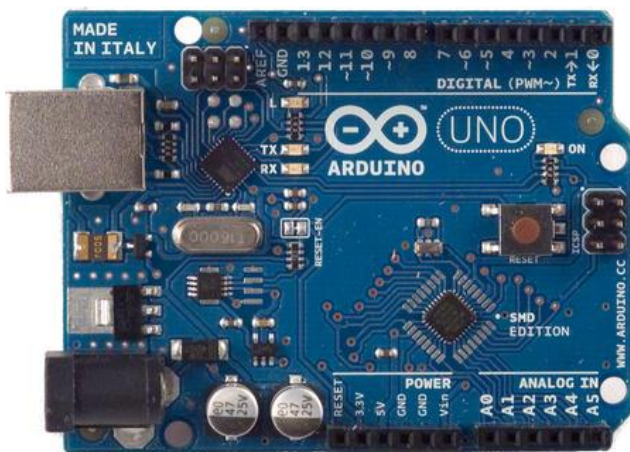
認識 Arduino

講師 張傑帆 Chang, Jie-Fan

開放式硬體的發展，尤其是 Arduino 的迅速發展，讓更多沒有電子背景的人開始進入電子學這個精彩的領域，這裡包括電子愛好者、藝術家、設計師，甚至是軟體發展人員。不需要瞭解和掌握太多硬體方面的知識，每個人都能結合自己擅長的方面製作出一些有意思的電子作品。Arduino 與普通元件最大的不同就是能夠藉由改變內部的程式實作各種不同的功能。接下來就讓我們真正進入 Arduino 的世界。

什麼是 Arduino

- Arduino 是源自義大利的一個開放原始程式的硬體專案平台
- 開發者可以用來開發互動產品，例如可以讀取大量的訊號，用來控制電源開關和傳感器設備的訊號
- 是一塊開放原始碼程式

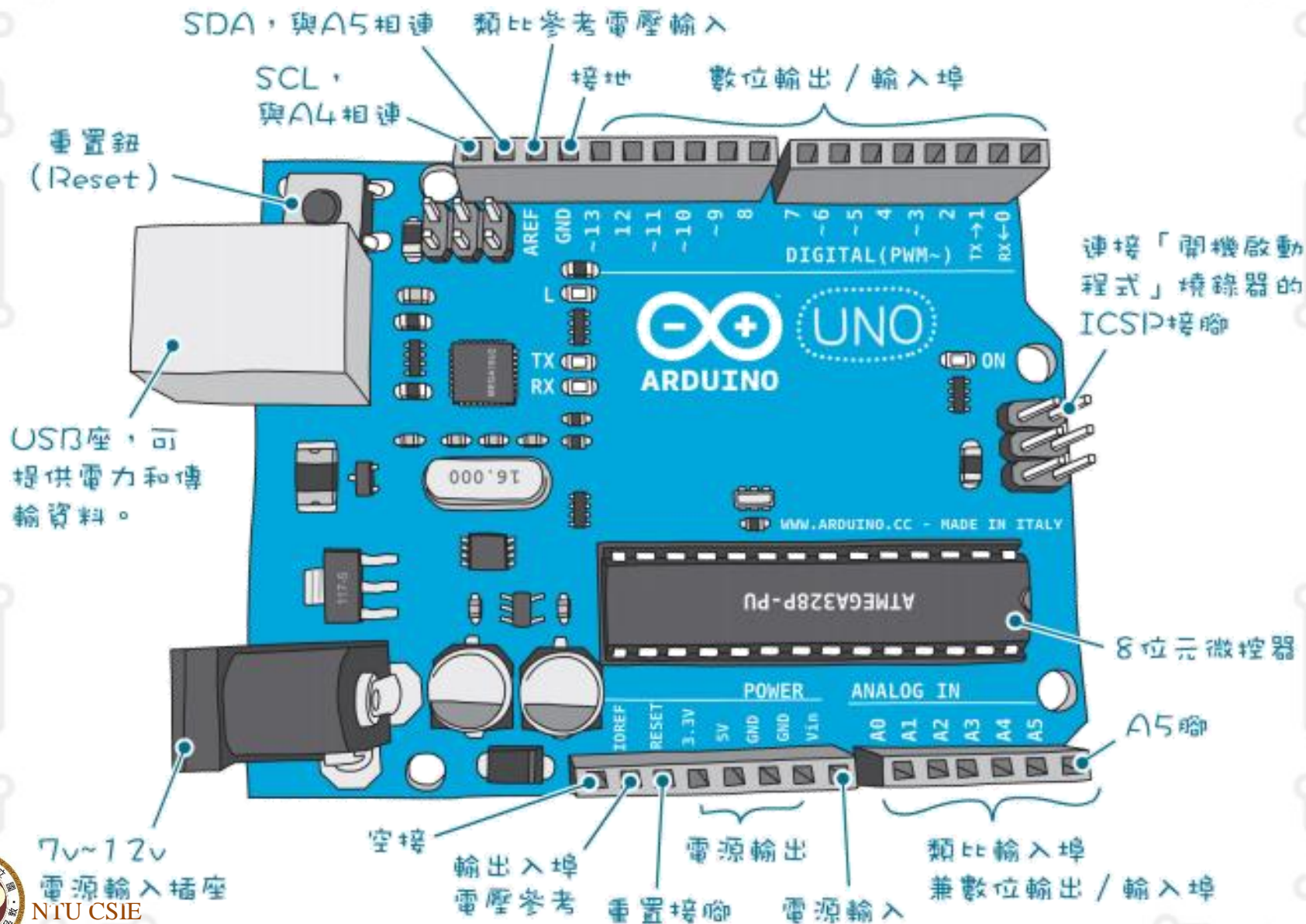


開放式硬體

- Arduino是由義大利米蘭互動設計學院 Massimo Banzi，David Cuartielles，Tom Igoe，Gianluca Martino，David Mellis及 Nicholas Zambetti 等核心開發團隊成員所創造出來，並將設計圖上傳到網路上：開放硬體設計的設計圖與程式碼
- 開放式硬體是開放原始碼 (open-source) 文化的一部分，是指在設計中公開詳細資訊的硬體設計，包括機構、電路圖、材料清單和控制程式碼等
- 應用開放原始碼思想，可以大量取得各式各樣的天才思想和設計，使軟體和硬體系統更加的完善，同時更加的開放。
- Arduino 基於知識共用的 CC 協議設計，任何人都可以複製、重設計甚至出售 Arduino 板。人們不用花錢購買版權，連申請許可權都不用，如果你加工出售 Arduino 原板，版權還是歸 Arduino 團隊所有。Arduino 設計者們唯一所有的就是 "Arduino " 這個商標



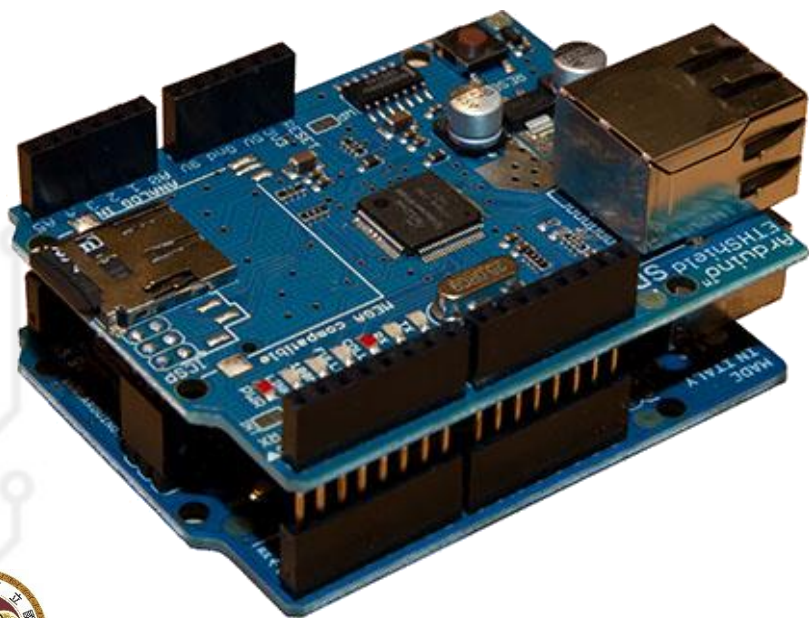
Arduino的接腳



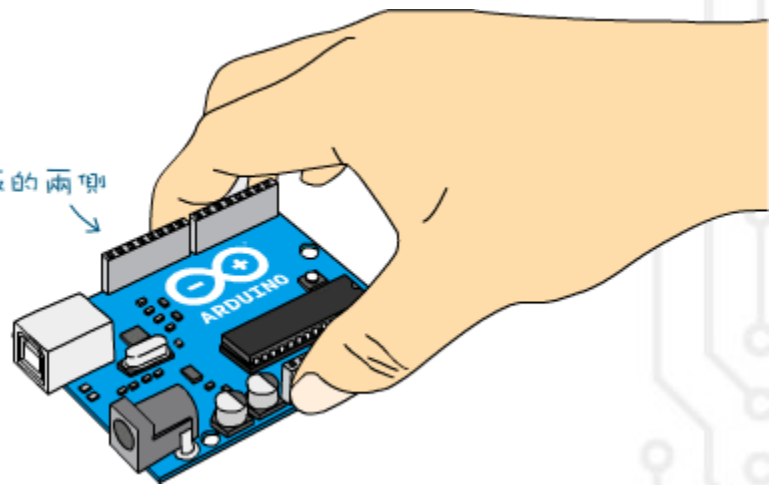
Arduino的擴充板（ shield ）

微電腦控制板就像是一個具有大腦和神經線，但是沒有感官和行動能力的物體。

微電腦板子上下兩側的黑色插槽，叫做**杜邦接頭**，是Arduino的擴充介面槽，用於連接Arduino擴充槽相容的擴充板（ Shield ）

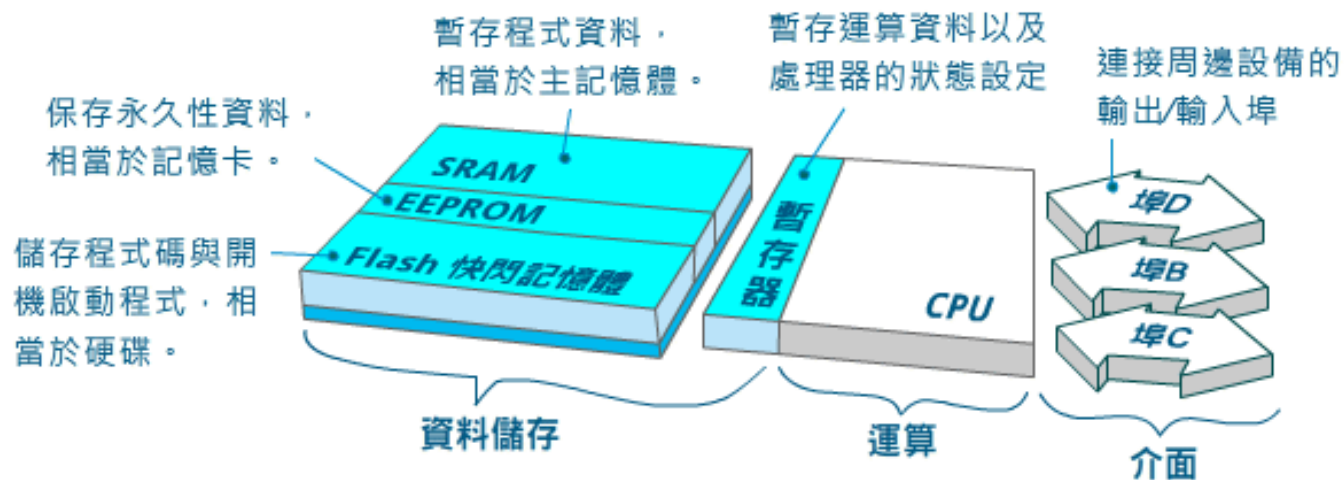


抓取Arduino板的兩側



認識ATmega328微控器

Arduino的微控器 (Microcontroller) 包含CPU、記憶體、類比/數位訊號轉換器以及周邊控制介面。



微控器內部具有三種記憶體：

名稱	類型	容量大小	用途
SRAM	揮發性 (volatile) ，代表資料在斷電後消失。	2048bytes (2KB)	資料記憶體；暫存程式運作中所需的資料。
Flash	非揮發性，代表斷電後，資料仍存在。	32768bytes (32KB)	程式記憶體；存放開機啟動程式和我們自訂的程式碼。
EEPROM	非揮發性	1024bytes (1KB)	存放程式的永久性資料。



Arduino 板的種類

Arduino 設計之初的目的是希望讓設計師和藝術家們能夠透過它很快地學習電子和感測器的知識，並應用到他們的設計當中，設計中所要表現的想法和創意才是最主要的，至於控制板如何工作，硬體的電路是如何構成的，設計師和藝術家們並不需要考慮。

1. 標準版 Arduino
2. 小型的 Arduino
3. 資源更豐富的 Arduino
4. 變形的 Arduino

Arduino 網站中還有一些**擴展板**，如 WiFi Shield、Ethernet Shield、Motor Shield 等，這些擴展板能夠將 Arduino 擴展到更廣泛的領域。



Arduino硬體介紹

表1-1 ATMEGA 系列晶片記憶體容量比較

記憶體容量	ATMEGA8	ATMEGA168	ATMEGA328	ATMEGA1280
Flash	8KB	16KB	32KB	128KB
SRAM	1KB	1KB	2KB	8KB
EEPROM	512bytes	512bytes	1KB	4KB

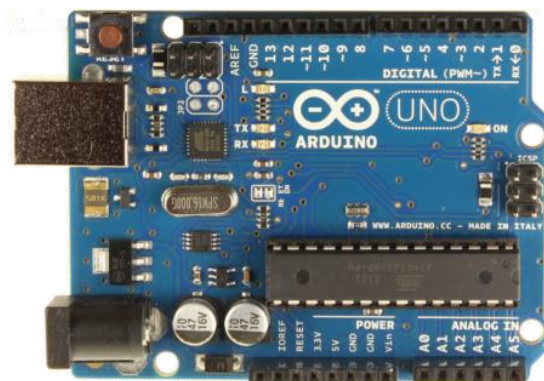


圖 1-2 Arduino Uno 板（圖片來源：arduino.cc）

Uno板

- “Uno”的義大利文是“一”的意思，用來紀念Arduino 1.0的發布，內部使用ATmega328微控制器。
- 在Arduino Uno板上有第二個微控制器ATmegaA8U2 或 Atmega16U2，取代FIDI公司的USB晶片，用來處理USB的傳輸通訊。
- 使用16 MHz 石英晶體振盪器。
- 有14支數位輸入/輸出腳（其中 6 支可作為 PWM 輸出腳）。
- 6支類比輸入腳，提供10位元的解析度。

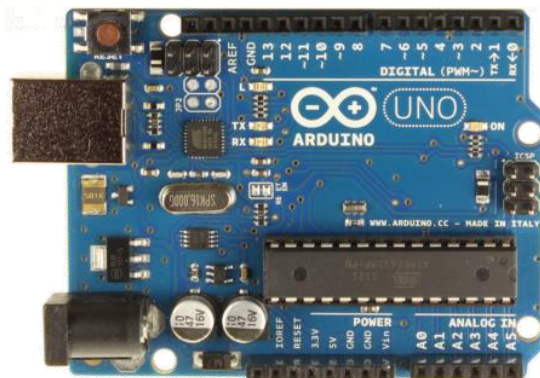


圖 1-2 Arduino Uno 板（圖片來源：arduino.cc）

Mini板

- Arduino Mini板，屬於Arduino迷你版，與郵票大小相同，使用 ATmega168或ATmega328微控制器，現今多使用 ATmega328微控制器。
- 使用16 MHz 晶體振盪器。
- 有 14 支數位輸入 / 輸出腳（其中6支可作為 PWM 輸出腳）。
- 8支類比輸入腳，每支類比輸入腳提供10位元的解析度。

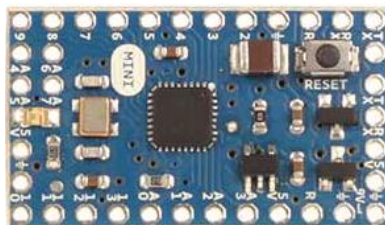


圖 1-4 Arduino Mini 板（圖片來源：arduino.cc）

Mega板

- Arduino Mega板，使用**ATmega1280** 微控制器，有更多的I/O連接埠，以及更強的微控制器。
- 使用**16 MHz** 石英晶體振盪器。
- 有**54支數位輸入 / 輸出**腳（其中**14支**可當作**PWM**輸出）。
- **16支類比輸入**腳。
- 大多數Arduino板只有一組UART串列埠，Mega板有**4組UART**串列埠。

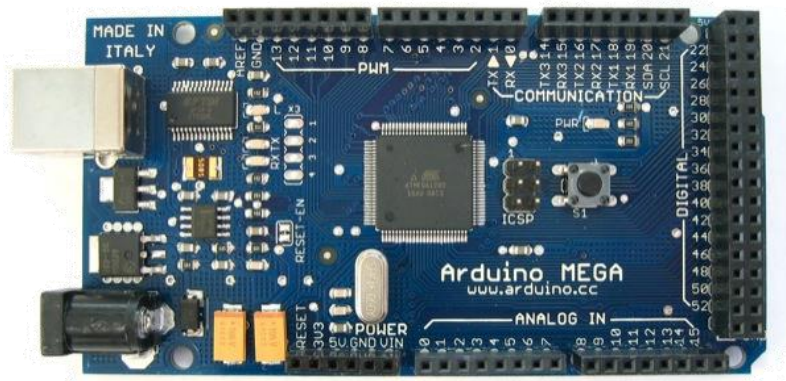


圖 1-5 Arduino Mega 板（圖片來源：arduino.cc）

LilyPad板

- Arduino LilyPad板，使用ATmega168V（ATmega168低功率版）或使用ATmega328V微控制器（ATmega328的低功率版）。
- Arduino LilyPad板主要是應用在服裝設計上，因為是圓型設計，所以可以如鈕扣一樣直接縫合到衣物上。

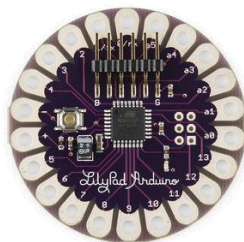


圖 1-6 Arduino LilyPad 板（圖片來源：arduino.cc）

Fio板

- Arduino Fio板，使用ATmega328P 微控制器，主要應用在無線網路上。
- Arduino Fio 板工作電壓為3.3V
- 使用8MHz振盪器。
- 有14支數位輸入 / 輸出腳（其中 6支可當作 PWM輸出）。
- 8支類比輸入腳

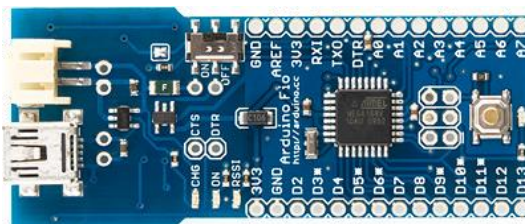
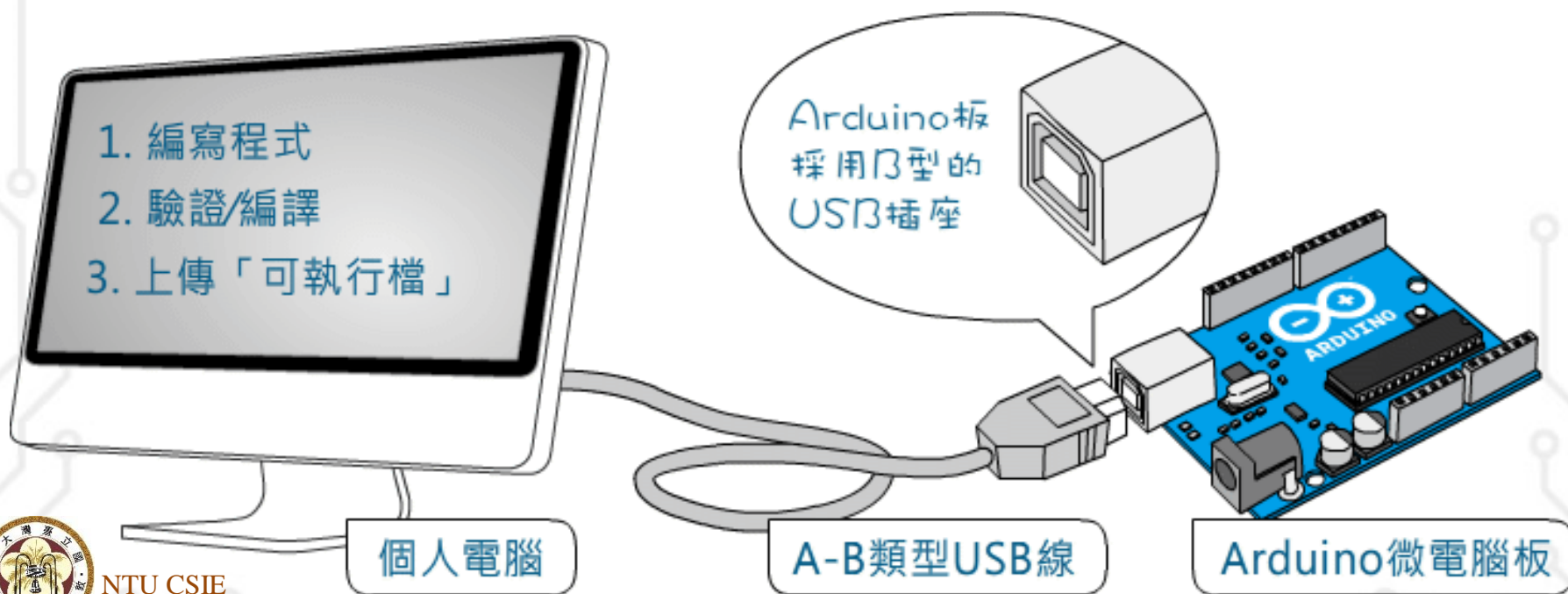


圖 1-7 Arduino Fio 板（圖片來源：arduino.cc）

Arduino程式開發步驟

1. 規劃裝置的功能和軟 硬體：裝置有什麼用途？需要哪些輸入裝置或感測器元件？有什麼輸出結果？
2. 組裝硬體：通常使用麵包板把電子零件組裝起來。
3. 編寫程式
4. 驗證和編譯：檢查程式內容是否有錯誤，並且編譯 (compile) 原始碼。
5. 上傳：也稱為燒錄，把編譯完畢的程式寫入微處理器內部的記憶體。



STEP 1

在 官 網 首 頁
arduino.cc 點選
【Download】選
項，進入下載畫
面。

Arduino 開發環境

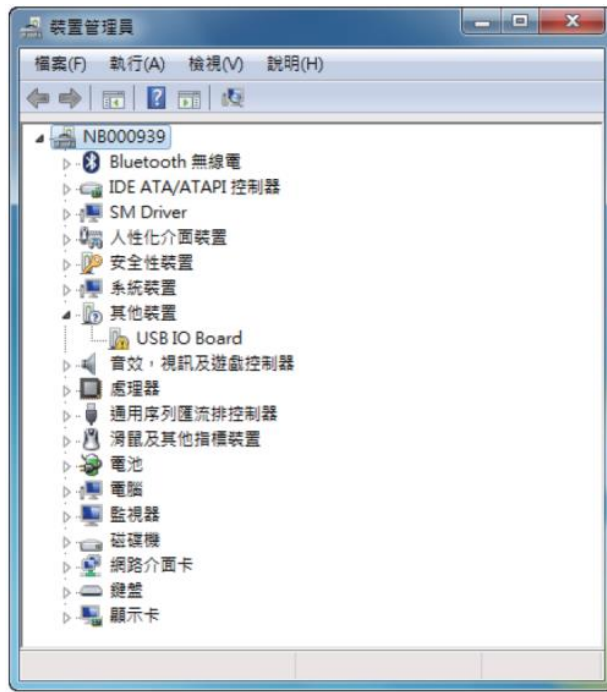
開發環境的取得

- Arduino 的開發環境（Arduino IDE）是完全免費而且是開放的，無需安裝，下載完成並解壓縮後就可以直接打開使用了，在 Arduino 的網站 <http://www.arduino.cc> 的 Download 頁面內提供壓縮檔的下載連結。
- 網站上提供了 Arduino 發展至今每個版本的開發環境，同時每個版本下針對不同操作系統提供了 Windows、Mac 和 Linux 三個壓縮檔（Linux 下又分為 32 位的和 64 位的），使用者可以根據自己的情況選擇合適的版本下載。
- 硬體使用 Leonardo 的話，必須使用 1.01 以上的版本（包括 1.01 版），本書中使用的是 Windows 環境下 1.04 版。

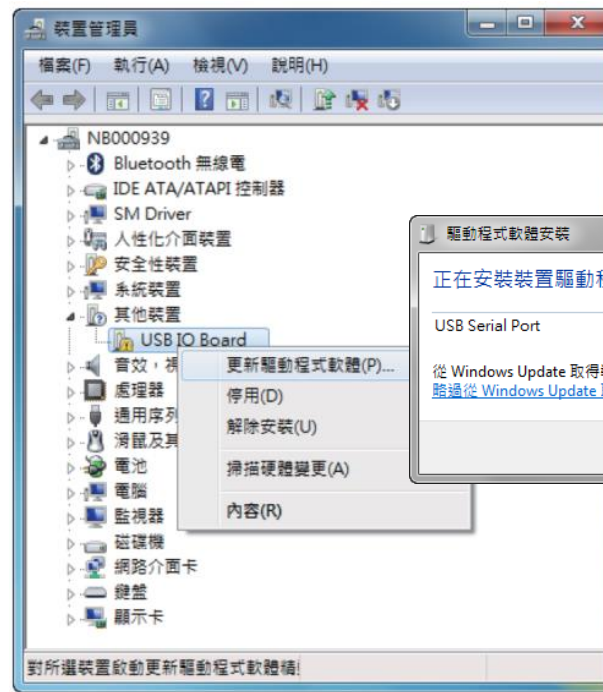


安裝新硬體

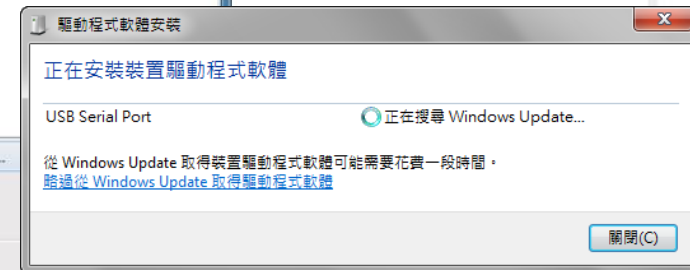
Arduino 控制板使用之前是需要安裝驅動程式的，上一小節說了，這個驅動程式就在 drivers 資料夾內。

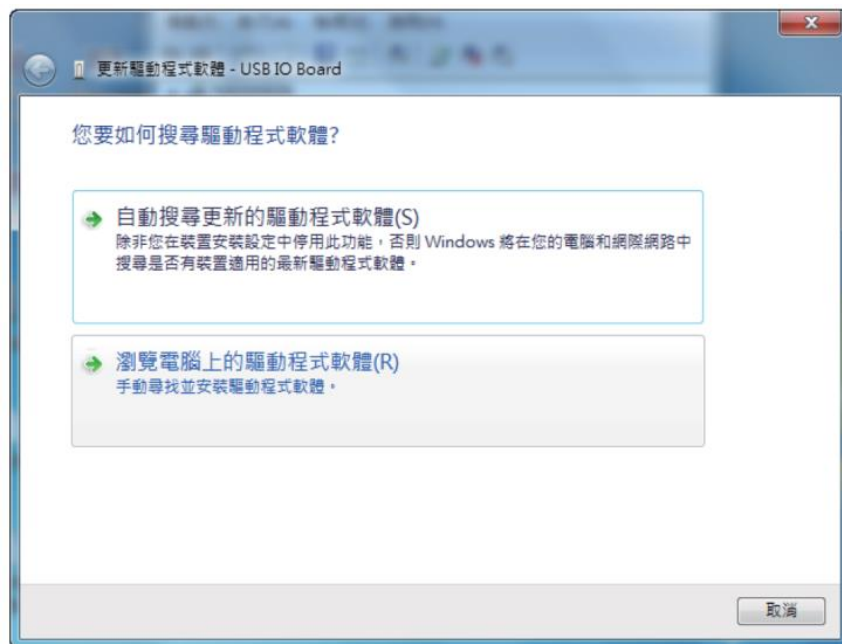


▲ 裝置管理員

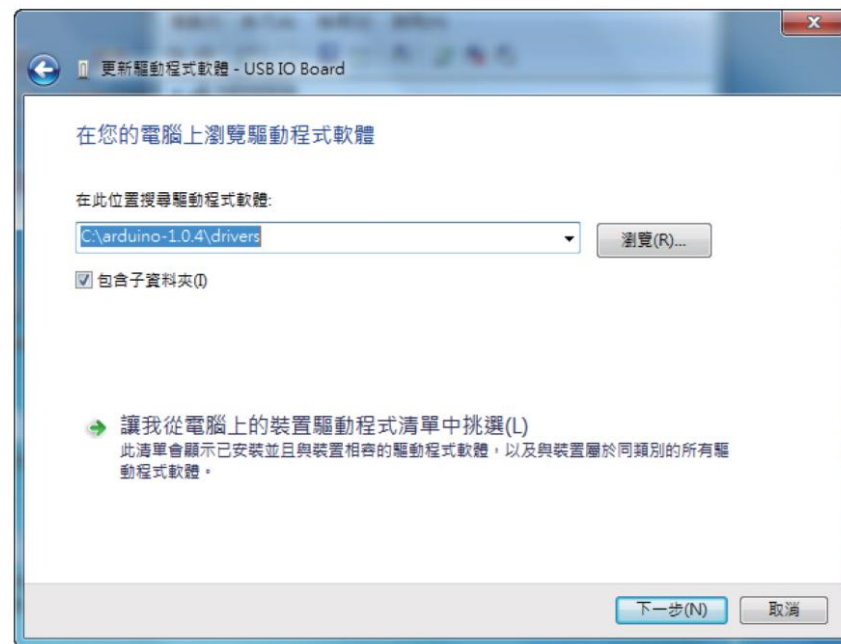


▲ 選擇 “更新驅動程式軟體”





▲ 選擇手動安裝方式



▲ 在電腦上找到驅動程式所在資料夾

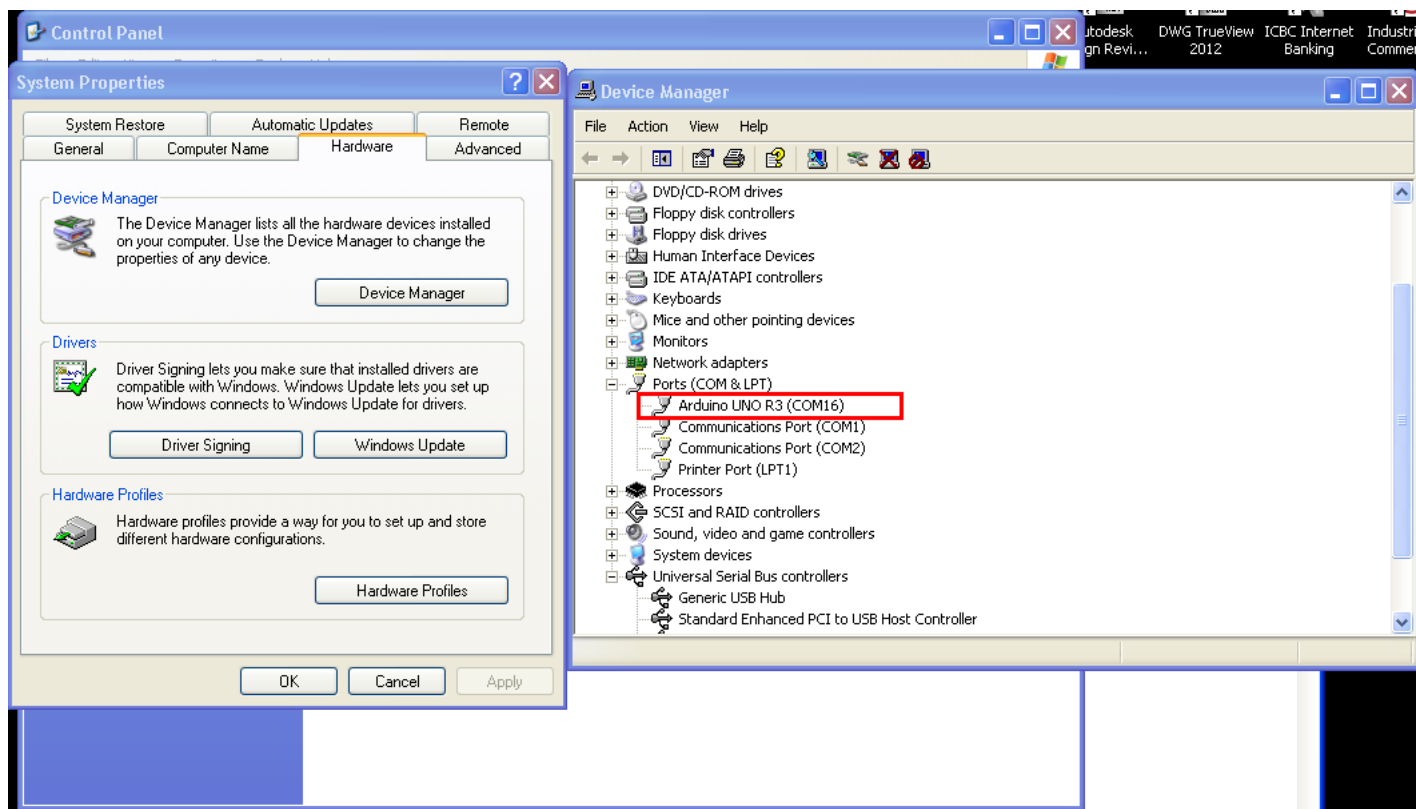


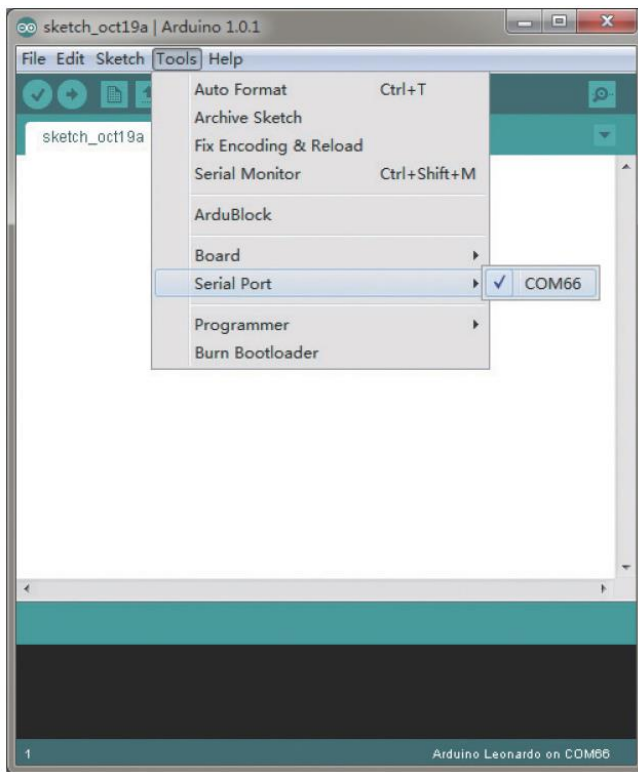
開發環境設置

- 介面裡中間的白色區域就是程序編輯區，黑色區域為資訊提示區，上方的功能表及按鈕為操作區
- 在操作區 Tools 選單下的 Serial Port 屬性中，選擇剛剛新增設備對應的串列埠號碼，這裡選擇的是 COM16(詳見下頁)，在 Board 一項中選擇 Arduino UNO
- 如果在你的 Board 一項中沒有 Arduino UNO，請檢查使用的開發環境版本是不是 1.01 或是更新的版本

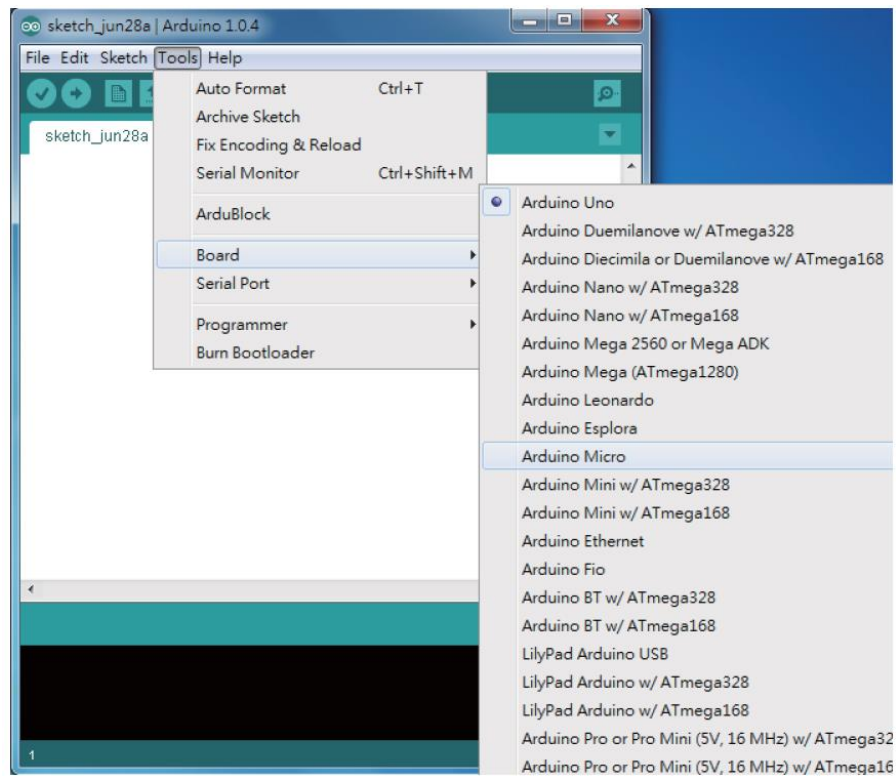


1. 請把 Windows Device Manager 裝置管理員打開。
2. 透過 Device Manager 調查 Ports





▲ 選擇串列埠









▲ 選擇 Arduino XXX (依版子而定)






Arduino IDE 介面中的快捷按鈕

快捷按鈕的介紹

- ◆  Verify (校驗)，用以完成程式的檢查與編譯。
- ◆  Upload (上傳)，將編譯後的程式檔上傳到 Arduino 板中。
- ◆  New (新建)，可新建一個程式檔。
- ◆  Open (打開)，打開一個存在的程式檔，Arduino 1.0 之後開發環境中的程式文件副檔名為 ".ino"。
- ◆  Save (保存)，保存目前的程式檔。
- ◆  Serial Monitor (串列埠監視視窗)，可監視開發環境使用的串列埠收發的資料，目前只能顯示 ASCII 字元。



快捷按鈕的使用

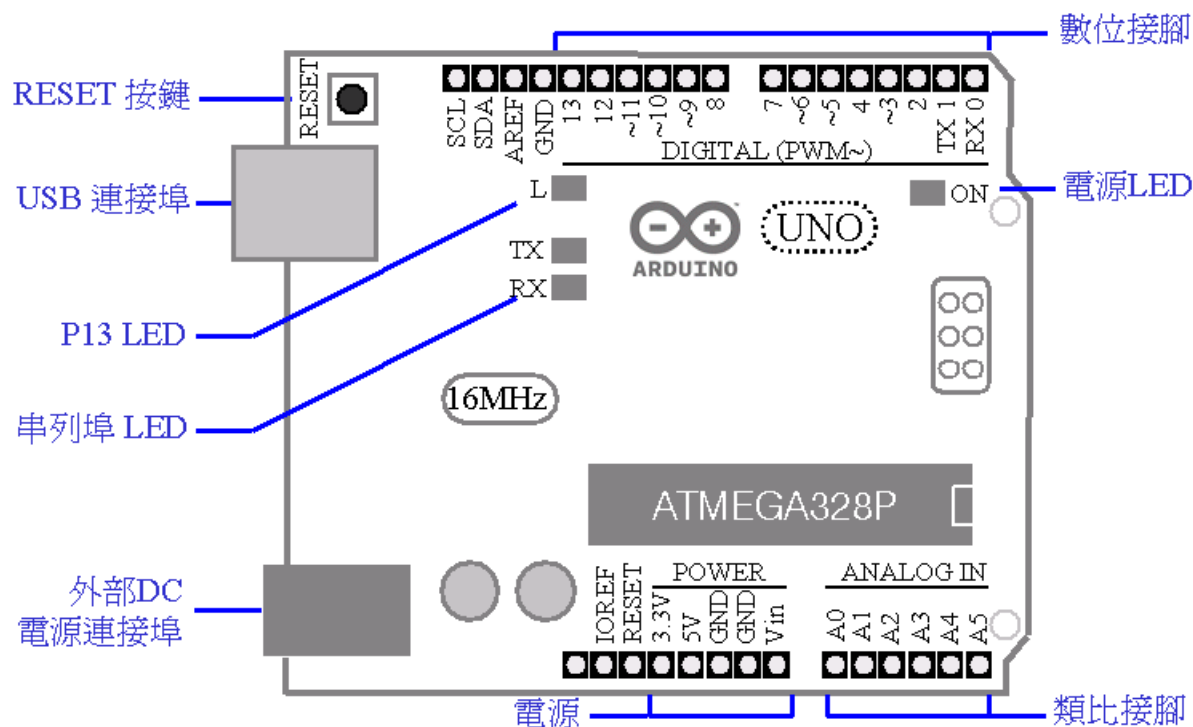
- 程式載入到程式區後，點擊  按鈕對程式進行編譯（編譯的工作可以理解為檢查程序的錯誤，編譯完成後狀態欄會提示 “Done compiling”（程式編譯完成）。
- 編譯完成後點擊  按鈕，將程式寫入 Arduino 控制板中。上傳通常要等待幾分鐘，此時控制板上的串列埠指示燈（TX 和 RX）會不停閃爍。
- 在點擊  按鈕之後，開發環境也會執行編譯的操作，如果程式存在問題，是無法寫入控制板中的。上傳完成後，狀態欄會有上傳成功的提示 “Done uploading”。



Blink 範例的程式

STEP 1

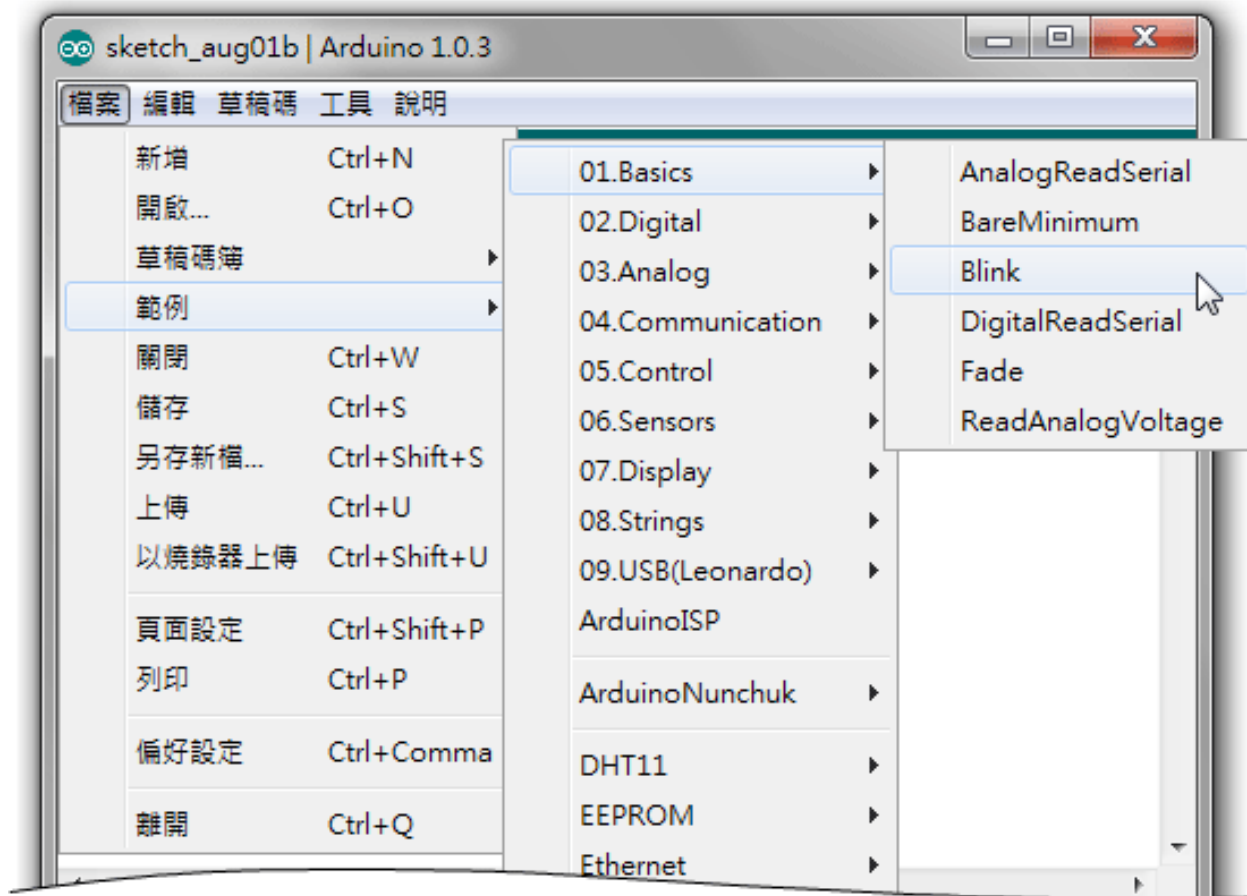
以 USB 線連接
如右圖所示
Arduino UNO 板
USB 連接埠與
電腦 USB 埠，並
檢查 電源
LED(綠色)是否
有亮。




STEP 2

開啟 Arduino 開發環境。

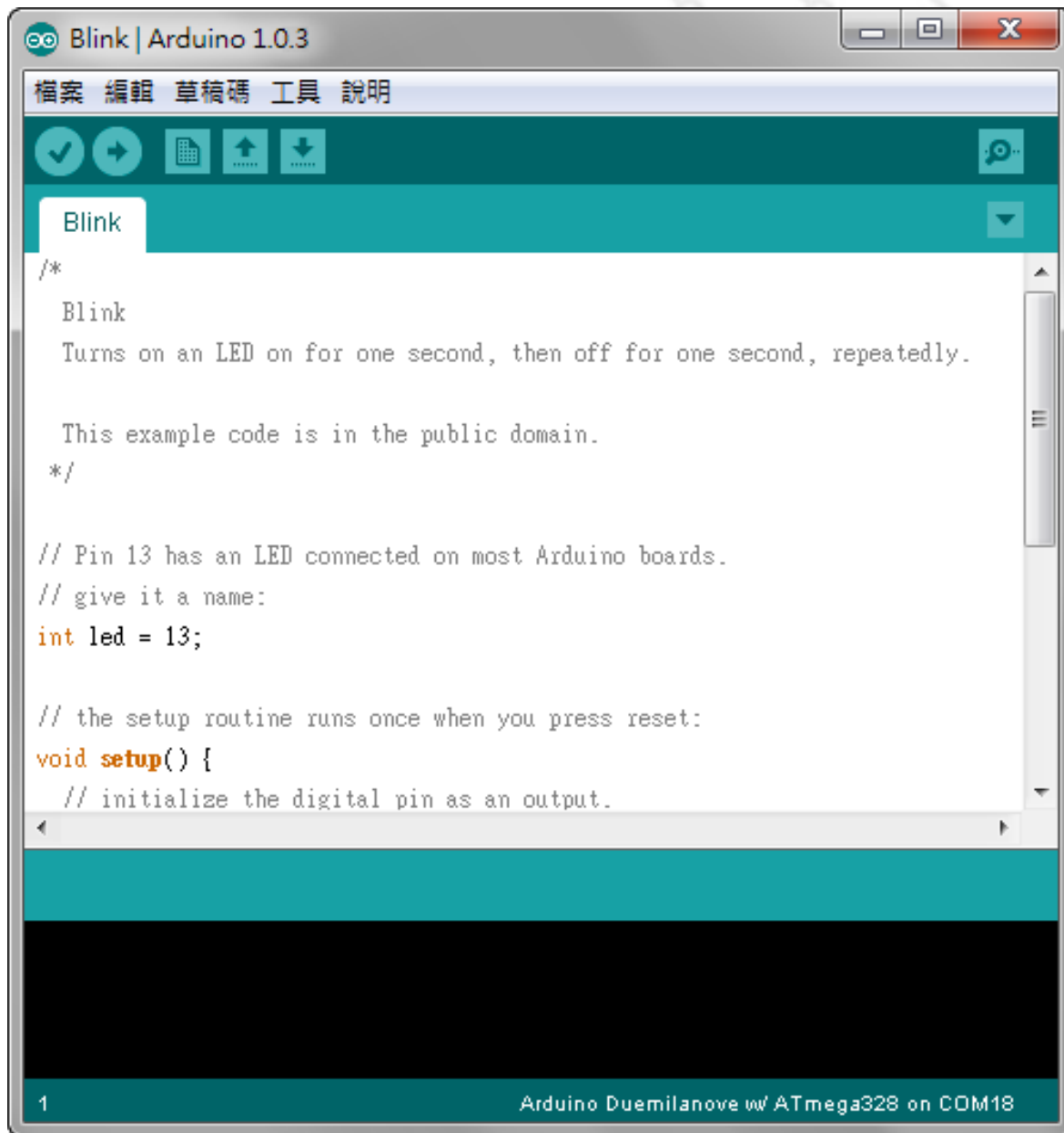
點選【檔案】【範例】【01.Basics】開啟 Blink 程式檔。Blink 程式功能為使 LED 亮 1 秒、暗 1 秒的閃爍功能。此處先不用理會程式的內容。



STEP 3

點選驗證鈕
，編譯專案檔。

編譯完成後，在下方的訊息視窗中會出現草稿碼 (sketch) 使用的 flashROM 大小及可使用的上限，因為使用 ATmega328P 微控制器，所以上限為 32KB。




The screenshot shows the Arduino IDE interface. The top menu bar includes '檔案' (File), '編輯' (Edit), '草稿碼' (Sketch), '工具' (Tools), and '說明' (Help). The toolbar contains icons for opening, saving, and uploading. The 'Blink' sketch is open, showing the following code:

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

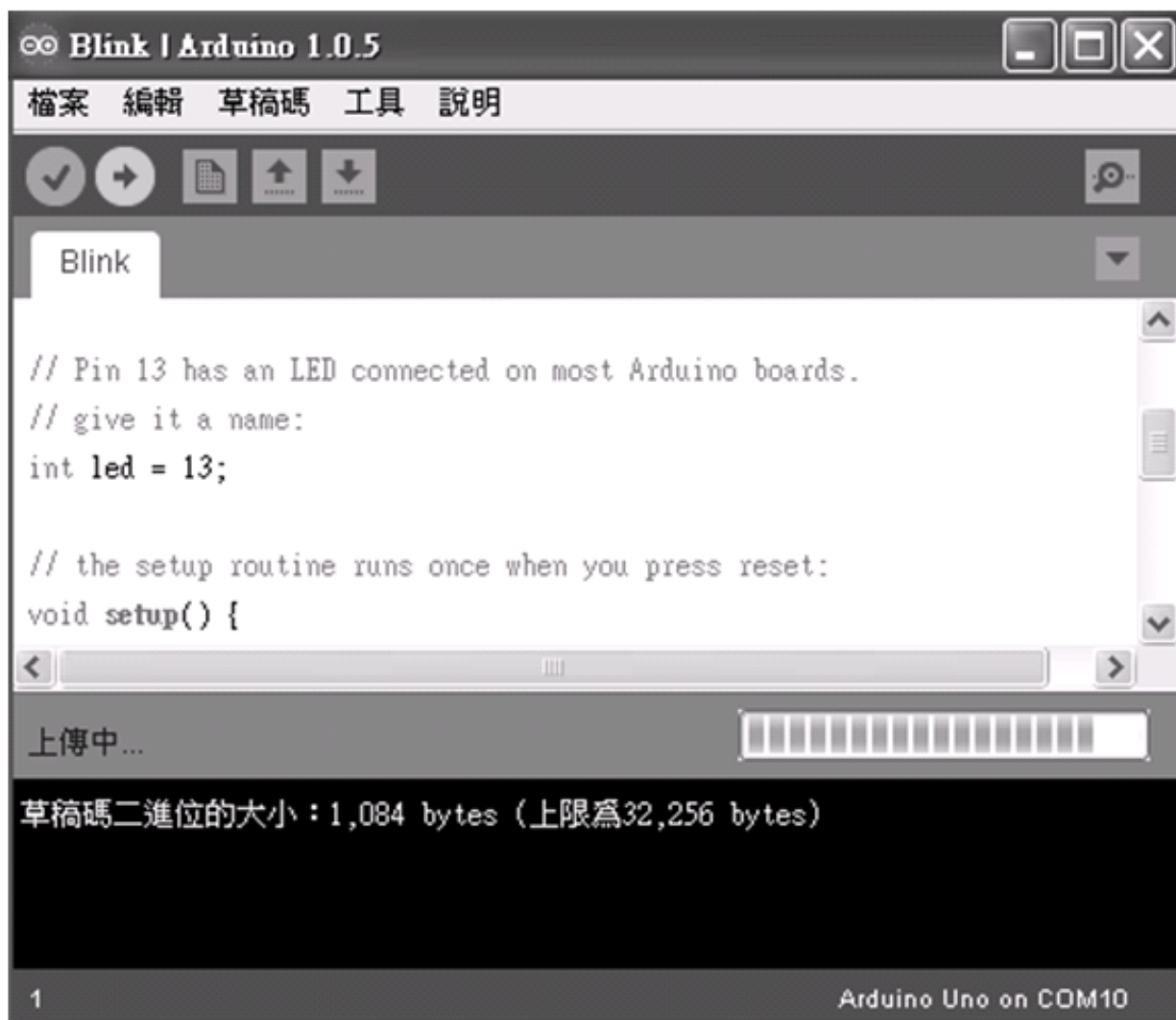
The serial monitor at the bottom shows the output: '1' and 'Arduino Duemilanove w/ ATmega328 on COM18'.



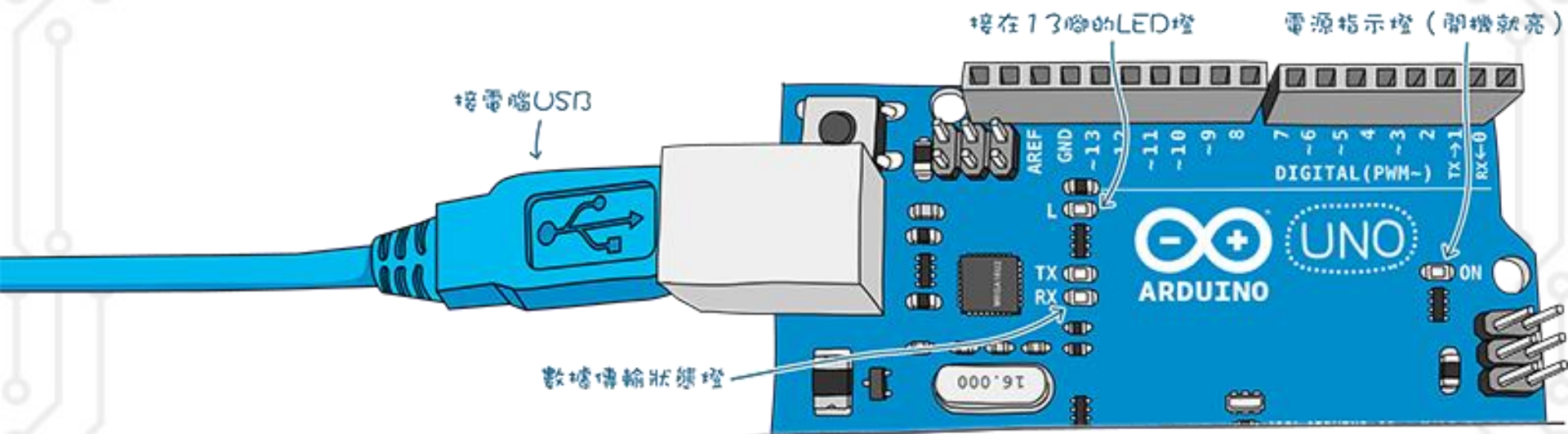
STEP 4

點選上傳鈕，將『可執行檔』上傳至 Arduino 板。上傳過程中，在訊息列會出現『上傳中...』訊息。完成後則出現『上傳完成』。

檢視 Arduino 板上 P13 LED(橙色)是否正確閃爍？



上傳「LED閃爍」程式



註解

- 在程式編輯區只有兩種類型的資訊，一種是程式，另一種是註解。程式是寫給控制板或者晶片的，而註解是寫給人看的。
- 在電腦程式設計領域中，註解就是對程式碼的解釋和說明（後面說的註解均是程式設計領域的註解）。目的是為了讓別人和自己能夠很容易地看懂程式碼的含義，讓別人一看就知道這段程式碼是做什麼用的。
- 將程式碼中的註解統統刪掉之後，程式碼依然可以正常運行。正確的程式註解通常包括序言性註解和功能性註解。
- 序言性註解的主要內容包括模組的介面、資料的描述和模組硬體的功能。程式的功能性註解的主要內容包括程式段的功能、語句的功能和資料的狀態。
- 單行註解以符號 `"//"` 開頭；多行註解以符號 `"/*"` 開始，直到後續的行中出現 `"*/"` 符號表示結束。



資料型態/類型

這個 `int` 叫資料型態，在程式設計領域是按照整數、小數、字元這樣的規則來區分資料型態的，不同的資料類型所占的記憶體空間不一樣。

▼ 表 幾種常見的資料類型

資料類型	說明
char	字元型資料
double	雙精度浮點數
float	單精度浮點數
int	整數型態
long	長整數型態
short	短整數型態
signed	有符號數，二進位資料中最高位元為符號位元
unsigned	無符號資料

程式中每一條語句必須以分號結束



函數

- 在程式設計領域中，將一段程式碼用大括弧封裝起來，在需要使用時直接透過函數名就可以調用，這就是程式中的函數。
- 函數的優勢在於它的模組化，一旦函數被定義，函數中的程式就不需要重複編寫了。
- 正是因為在 Arduino 中有很多其他工程師定義並完成的函數來實作底層硬體的操作，才能讓我們在不瞭解太多硬體知識的基礎上完成電子作品製作。

資料類型 函數名（參數）

{

實作功能的程式碼片段

}



- 其中，**函數名**相當於生產線的名字，以方便我們**調用**；函數名後面的**括弧內**是**參數**，參數相當於輸入的原始材料，參數可以是一個，也可以是多個，**沒有參數**的話括弧內就為**空**；函數同樣需要定義資料類型，這個資料類型是用來指定函數輸出的資料是什麼格式。
- 函數沒有輸出時函數類型為 `void`，這種函數叫無傳回值函數；在大括弧內的程式碼就是對輸入的資料進行處理的指令。
- **調用**一個已經完成的函數就用**函數名加上參數**，以**分號結束**就可以了，用法如下。

函數名（參數）；

- Ex:

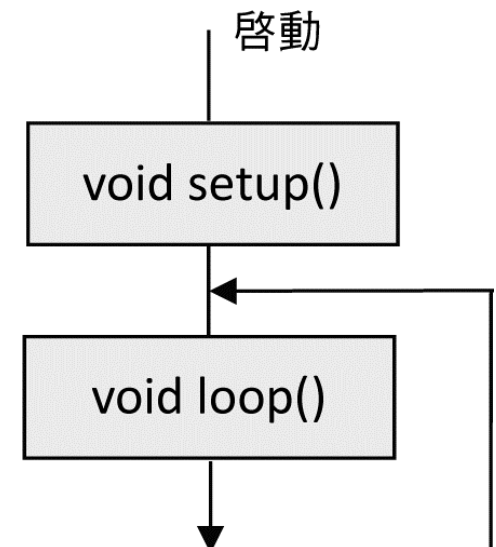
```
pinMode(13, OUTPUT);  
digitalWrite(13, HIGH);  
delay(1000);
```



程式結構

- **setup() 函數**和 **loop() 函數**是必須要寫的，這兩個函數分別負責 Arduino 程式的**初始化**部分和**迴圈執行**部分。
- 兩個函數**均無傳回值**，也**沒有參數**。
- **setup()** 函數用於控制板初始化，主要是用於設置一些接腳的**輸出 / 輸入模式**、**初始化串列埠**等，該函數只在控制板通電或重啟時執行一次
- **loop()** 函數是一個**循環**，其中的程式碼將被**迴圈執行**，來完成程式的功能，如：**讀入接腳狀態**、**設置接腳狀態**等

```
01 void setup()           // void setup()是當 Arduino 啟動時須設定的參數，做初始化的動作，只執行一次
02 {
03
04 }
05 void loop()             // 啟動後，會一直執行 loop()，loop()執行完畢後，又會再次呼叫 loop()
06 {
07
08 }
```



■ **pinMode()** 函數的功能是設置接腳的工作方式，有兩個參數

1. 第一個是所要設定的接腳，程式中的參數值寫的是 led，經由之前的內容我們已經知道 led 就是 第 13 腳位的名字；
2. 第二個參數是該接腳的工作方式，有 INPUT（輸入）和 OUTPUT（輸出）兩種狀態

```
pinMode(13, OUTPUT);
```

■ **digitalWrite()** 函數的功能是設置接腳的狀態，第一個是所要設定的接腳，同 pinMode() 函數類似，程式中的參數值寫的是 led；第二個參數是該接腳的狀態，有 HIGH（設定高電位，即輸出 +5V 電壓）和 LOW（設定低電位，即輸出 0V 電壓）兩種狀態。

■ **delay()** 不用介紹，大家一定猜到了這是一個延遲函數，該函數只有一個參數，就是等待或延遲的時間，參數的單位是毫秒，類似的還有一個 delayMicroseconds() 函數，所不同的是該函數的參數單位是微秒 (1 毫秒 = 1000 微秒)。

```
digitalWrite(13, HIGH);  
delay(1000);
```



Setup 函數

程式初始化

點亮 LED

延遲 1 秒

熄滅 LED

延遲 1 秒

Loop 函數

```
void setup() {  
    // initialize digital pin  
    pinMode(13, OUTPUT);  
}  
  
// the loop function runs c  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

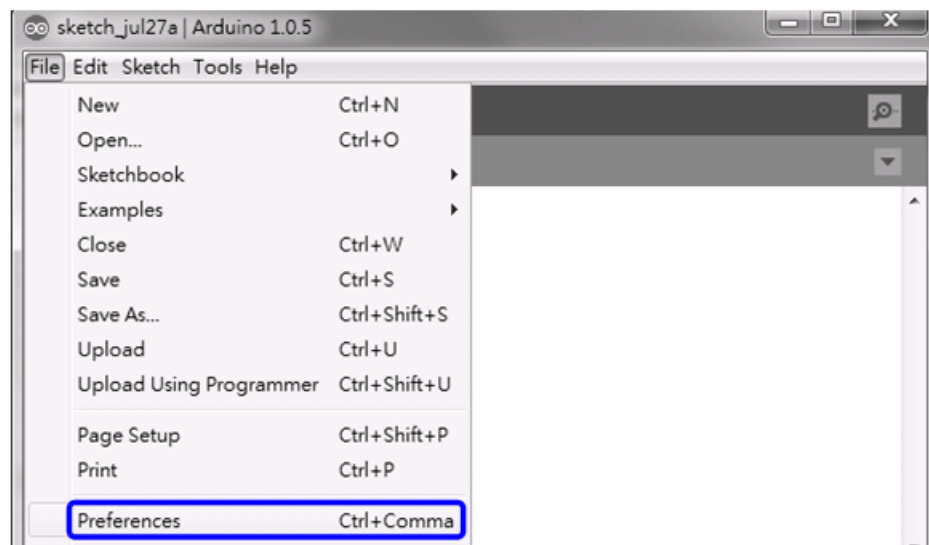
▲ Blink 程式示意圖



更改中文介面

STEP 3

開啟 Arduino IDE，點選【File】
【Preferences】
進入偏好設定畫面。



STEP 4

選擇【繁體中文】選項後，按下【OK】鈕。

要重新關閉再開啟 Arduino IDE，設定才會生效。

