

Arduino 輕鬆上手18堂課程目錄

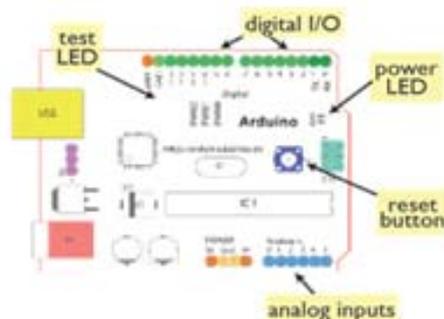


Class 1 :	認識Arduino	Page- 2
Class 2 :	Arduino程式語言	Page- 17
Class 3 :	Arduino程式控制元件	Page- 34
Class 4 :	LED元件的運用	Page- 49
Class 5 :	數位輸入-按鍵的運用	Page- 65
Class 6 :	數位與類比	Page- 79
Class 7 :	可變電阻的運用	Page- 85
Class 8 :	七段顯示器的運用	Page- 91
Class 9 :	蜂鳴器的運用	Page- 97
Class 10 :	綜合練習一：使用按鍵玩音樂	Page-106
Class 11 :	光敏電阻的運用	Page-108
Class 12 :	伺服馬達的運用	Page-116
Class 13 :	電晶體的運用	Page-125
Class 14 :	綜合練習二：使用光敏電阻使馬達追光	Page-133
Class 15 :	紅外線距離感測器	Page-134
Class 16 :	薄膜電位計	Page-139
Class 17 :	超音波感測器	Page-143
Class 18 :	加速度計	Page-149

What's Arduino ?

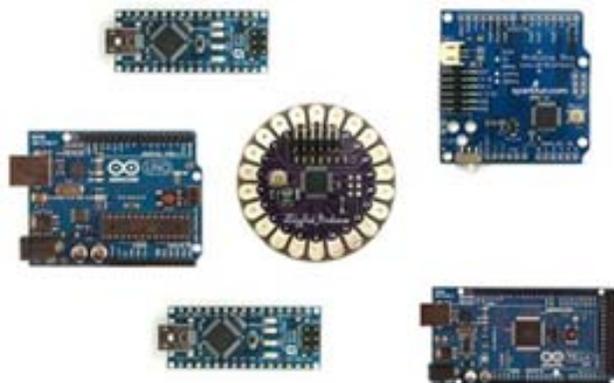


Arduino UNO 模組



Arduino UNO 硬體說明圖

Arduino 其他各種
微處理器



Open Source Physical Computing Platform (硬體)

可程式化單晶片

提供包括類比、數位、SPI、串列介面與PWM輸出

可以透過USB與PC/Mac電腦通訊，或是可以獨立運作。

Sensor elements



Ultrasonic



Temperature/Humidity



GPS Receiver



Tri-Axis Accelerometer



Color

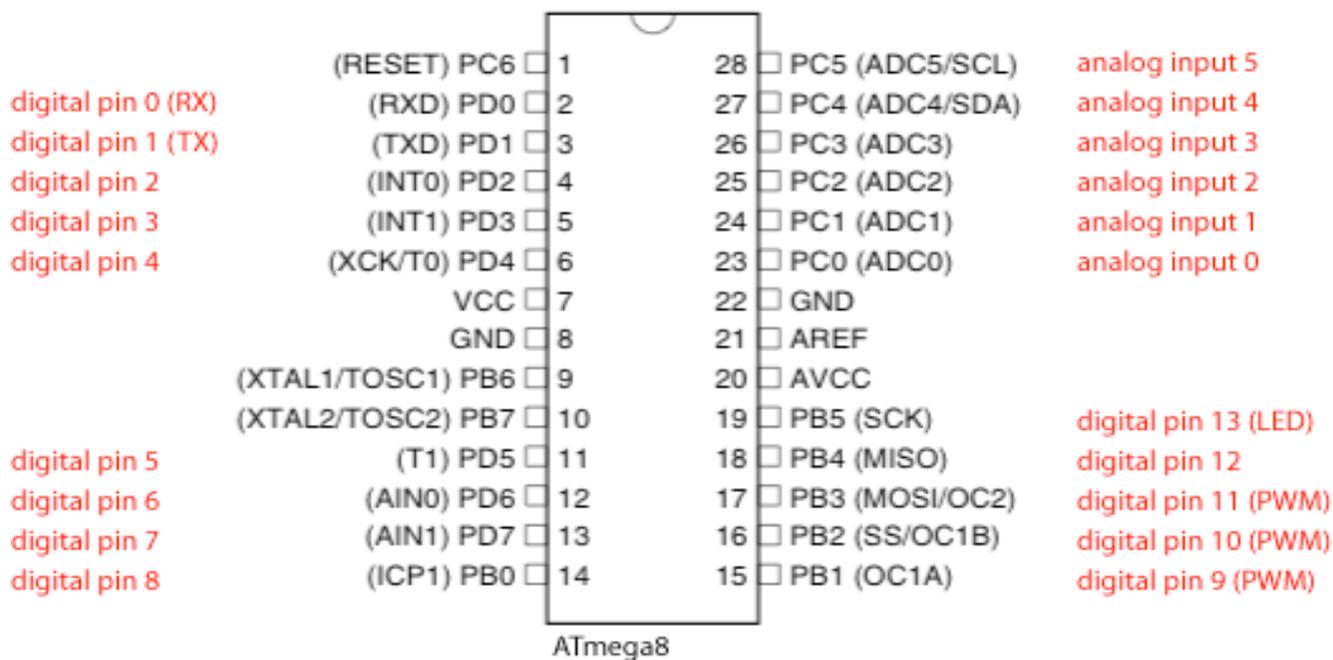


PIR

Arduino的IC

Arduino Pin Mapping

www.arduino.cc



規格

Arduino UNO 規格:

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

腳位 (I/O)	輸入 (Input)	輸出 (Output)
D0(RX)	Digital	Digital
D1(TX)	Digital	Digital
D2	Digital	Digital
D3	Digital	Digital ,Analog
D4	Digital	Digital
D5	Digital	Digital ,Analog
D6	Digital	Digital ,Analog
D7	Digital	Digital
D8	Digital	Digital
D9	Digital	Digital ,Analog
D10	Digital	Digital ,Analog
D11	Digital	Digital ,Analog
D12	Digital	Digital
D13	Digital	Digital
A0	Analog	X
A1	Analog	X
A2	Analog	X
A3	Analog	X
A4	Analog	X
A5	Analog	X

如何安裝軟體

步驟一:至官方網站下載軟體” Arduino 0022 “(www.arduino.cc/en/Main/Software)

Download

Arduino 0022 (release notes), hosted by Google Code:

- + Windows
- + Mac OS X
- + Linux: 32 bit, 64 bit
- + source

依照作業系統選擇適合的版本

Next steps

- Getting Started
- Reference
- Environment
- Examples
- Foundations
- FAQ

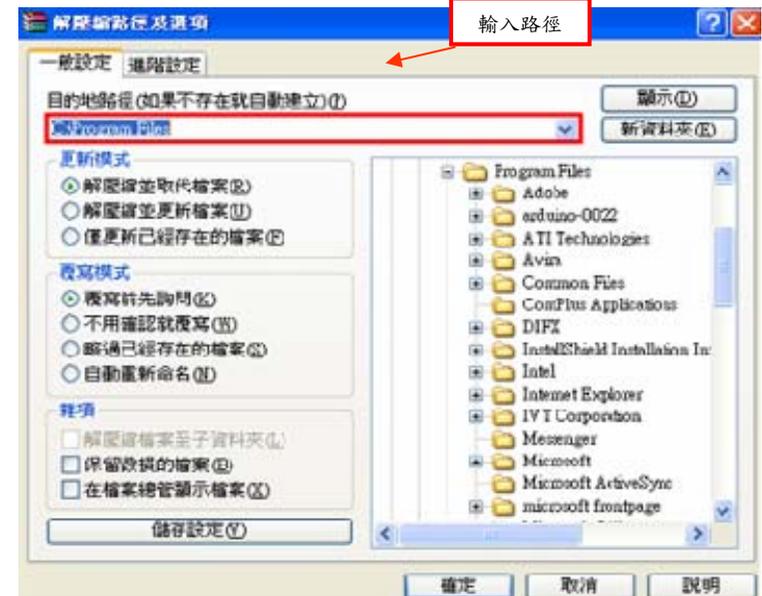
Also available from Arduino.cc: [Windows](#), [Mac OS X](#), [Linux \(32bit\)](#), [Linux \(64bit\)](#), [Source](#)

選擇適合自己的作業系統下載

步驟二:將軟體解壓縮到指定資料夾(以下將用WindowsXP示範)

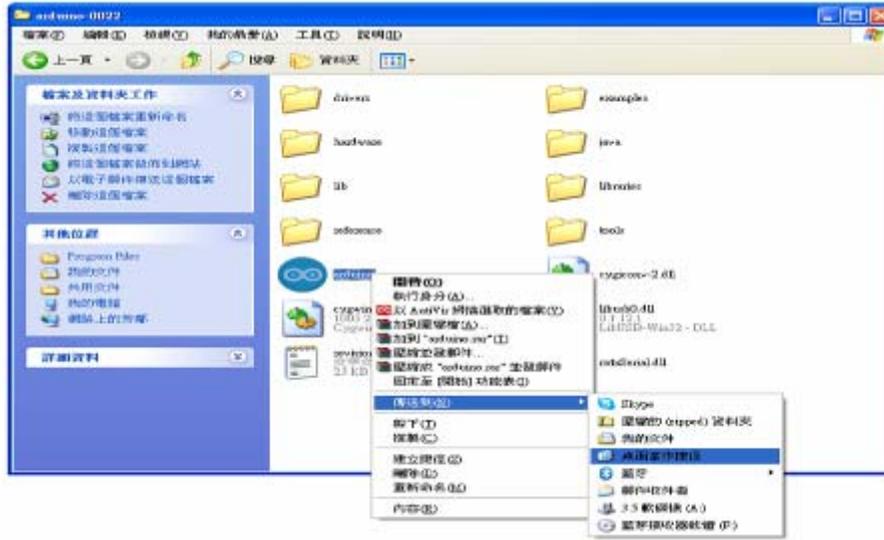


將Arduino 0022 解壓縮到指定資料夾



指定路徑(C:\program files\)

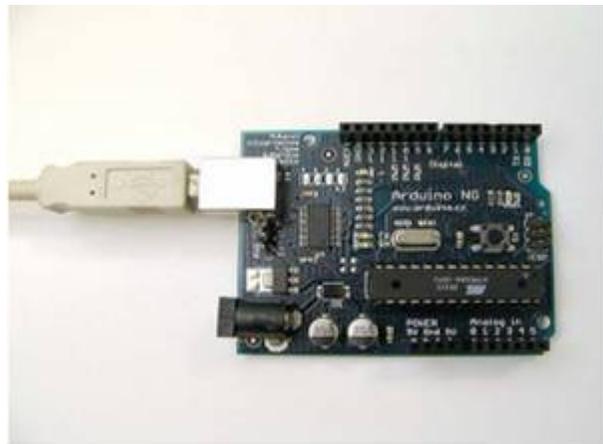
步驟三:在桌面建立Arduino捷徑



打開
C:\Program Files\arduino-0022
↓
對Arduino.exe 按下滑鼠右鍵
↓
傳送到桌面當捷徑

指定傳送到桌面當捷徑

步驟四:將Arduino UNO 連結上電腦



將Arduino 與電腦連結

使用USB傳輸線連結Arduino UNO與電腦

(僅須接上USB線不需要在額外接上電源)

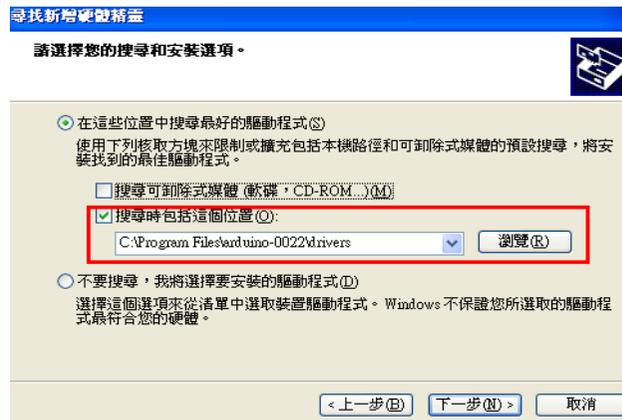
灌入驅動程式

步驟五：驅動Arduino

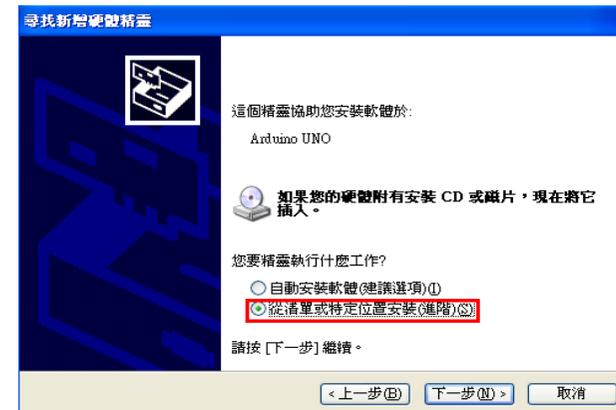
當正確連結上電腦後，我們還要灌驅動程式讓電腦能夠正確的控制 Arduino，當電腦偵測到Arduion時，請依照下面步驟灌入驅動程式



在新增硬體精靈選擇不要連線到網際網路



搜尋位置選擇Arduino-0022資料夾中的Drivers 資料夾，或輸入路徑
C:\Program Files\arduino-0022\drivers



勾選從清單或特定位置安裝



打開裝置管理員，檢查是否正確連結上Arduino UNO

測試Arduino第一個程式

當電腦可以正確與Arduino連結，我們就可以嘗試在Arduino上撰寫程式了！



```
//Robotics with the ABB-Car ABB_to_PC.pde
//Arduino sends a text message to your PC

void setup(){ //初始設定
  Serial.begin(9600); //設定串列埠傳輸的速率
}
void loop(){ //不斷重複執行
  Serial.println("HELLO,this is a message from your Arduino."); //在螢幕上輸出
  delay(500); //延遲500ms
}

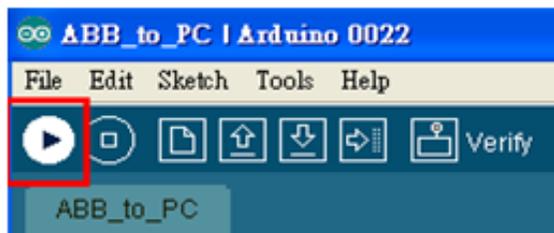
|

Done Saving.
Binary sketch size: 2118 bytes (of a 32256 byte maximum)
12
```

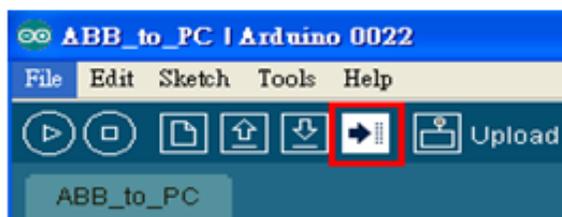
1. 開啟桌面上Arduino 0022 進入程式畫面
2. 依照圖1-9輸入程式 (請務必注意大小寫)

範例程式(在電腦螢幕顯示輸入字串)

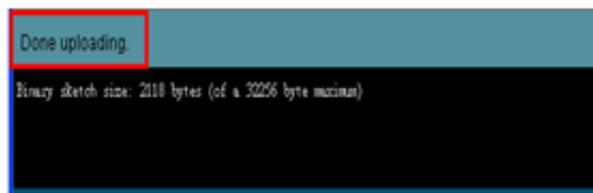
將程式燒錄進Arduino 中



按下Verify 檢查程式能否運行

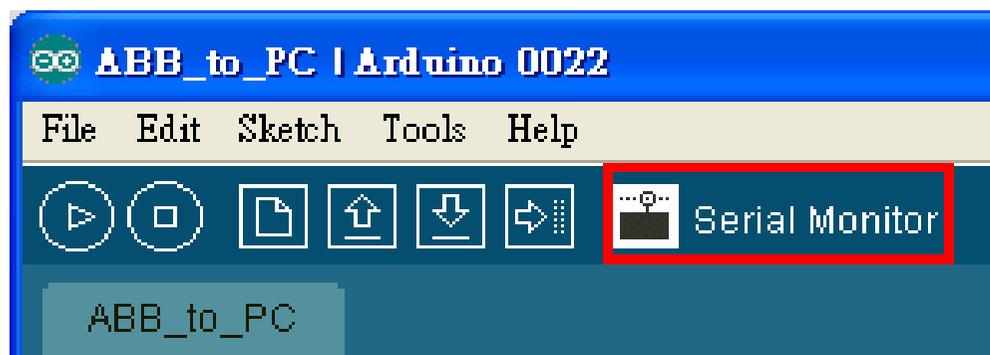


按下Upload將程式燒錄進Arduino

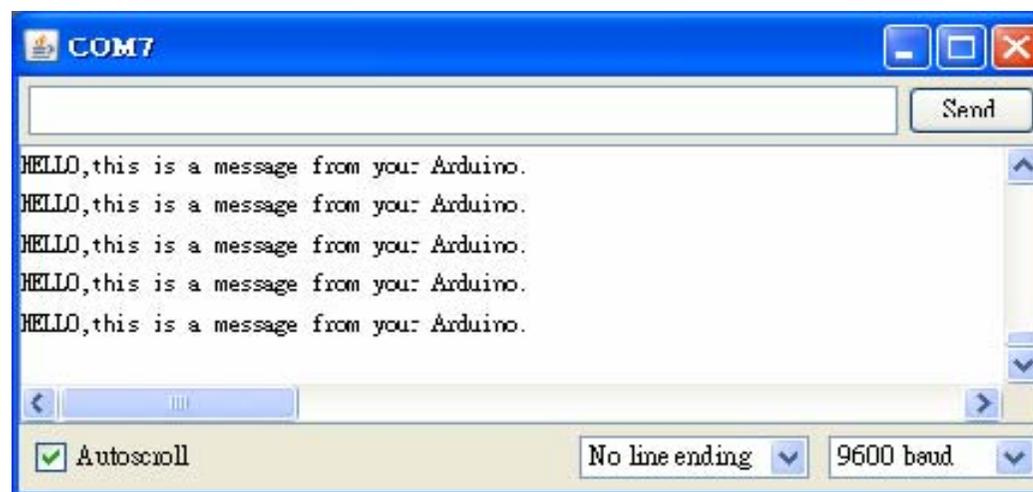


燒錄完成後視窗下方會顯示Done uploading

依照圖1-10.1~1-10.5的步驟
將可以將程式燒錄進Arduino
中並且在螢幕中顯示輸入字
串



燒錄完成後點下此鈕即可在螢幕上看到回傳值



在電腦中會顯示出輸入的字串

程式範例:HelloArduino.pde



```
ABB_to_PC | Arduino 0022
File Edit Sketch Tools Help
ABB_to_PC $
//Robotics with the ABB-Car ABB_to_PC.pde
//Arduino sends a text message to your PC

void setup(){
    //初始設定
    Serial.begin(9600); //設定串列埠傳輸的速率
}

void loop(){
    //不斷重複執行
    Serial.println("HELLO,this is a message from your Arduino.");
    Serial.println("What's 7x11?");
    Serial.println("7x11=");
    Serial.println(7*11,DEC);
    Serial.println("-----");
    delay(1000); //延遲1秒
}

Done uploading.
Binary sketch size: 2272 bytes (of a 32256 byte maximum)
14
```

請依照圖1-11所示修改範例程式，觀察有何變化

C語言的由來：

C語言最早是由丹尼斯·里奇為了在PDP-11電腦上運行的UNIX系統所設計出來的程式語言，第一次發展在1969年到1973年之間。之所以被稱為「C」是因為C語言的很多特性是由一種更早的被稱為B語言的程式語言中發展而來。

Arduino 的語法：

為義大利籍工程師David Mellis所設計，用法與C語言相似，但在使用上更具有便利性，十分容易上手。

程式架構

在我們開始編寫程式之前，首先我們必須要了解到Arduino語法中最重要也最基本的兩個控制結構Setup()、Loop()以下是兩個控制結構的介紹。

初始化Setup()：

在這個函數範圍內放置初始化Arduino的程式 - 在主程式開始撰寫前，使Arduino板子裝置妥當的指令。

例：`int a=1;`

```
Setup(){  
pinMode(a,INPUT);  
}
```

→在開始的時候執行一次,設定pin1腳為輸入腳

執行Loop()：

在此放置你的Arduino腳本。這部份的程式會一直重複的被執行，直到Arduino板子被關閉。

例：`Loop(){`

```
.....  
}
```

→程式開始後會一直執行程式內的內容

符號

;(分號):

Arduino 語言每一行程序都是以分號為結尾。這樣的語法讓你可以自由地安排代碼，你可以將兩個指令放置在同一行，只要中間用分號隔開。
(但這樣做可能降低程式的可讀性。)

例：`delay(500);`

{}(大括號):

大括號用來將程式代碼分成一個又一個的區塊，如以下範例所示，在 `loop()` 函數的前、後，必須用大括號括起來。

例：

```
void loop(){
  Serial.println("cial");
}
```

註釋:

程式的註釋就是對代碼的解釋和說明，編寫註釋有助於程式設計師(或其他人)了解代碼的功能。Arduino 處理器在對程式碼進行編譯時會忽略註釋的部份。而 Arduino 語言中的編寫註釋有下列範例的兩種方式

例：`//單行註釋：這整行的文字會被處理器忽略`

```
/*多行註釋：
在這個範圍內你可以寫一整首詩
*/
```

資料型態

變數用來指定Arduino 記憶體來儲存資料，使我們可以透過程式代碼去不限次數的操作變數的值，其中變數還區分為全域變數和區域變數兩種，前者表示此變數對應到整個程式，只要程式中使用到該變數都會對應到同一個記憶體位只，而後者則僅在某區段中使用。

Arduino 的語法可以讓我們十分容易去使用變數來儲存資料，但你要宣告一個變數時必須先定義他的資料型態，好讓微處理器知道準備多大的空間以儲存這個變數值，下面我們將為大家介紹各種不同的資料型態。

boolean (布林變數) :

布林變數的值只能為真(true)或是假(false)

char (字元) :

單一字元例如 A，Arduino和一般的電腦做法一樣會自動將字元儲存成一個數字，即使你看到的明明就是一個文字。

bit (位元) :

資料型態中最小的單位，僅僅只占一個記憶體空間，儲存的數字範圍僅有0和1兩種。

byte (位元組) :

儲存的數值範圍為0到255。如同字元一樣位元組型態的變數只需要用一個位元組(8位元)的記憶體空間儲存。