

第四章：人機介面行動技術- Arduino

1

南臺科技大學資訊工程系
李育強 助理教授

Outline

2

- 4 - 1 **Arduino**簡介
- 4 - 2 建立**Arduino**開發環境
- 4 - 3 利用**Arduino** 開發第一個程式
- 4 - 4 **Arduino**裝置人機互動設計
- 4 - 5 **Arduino**與電腦端互動設計
- 4 - 6 **Arduino**與行動裝置互動設計

4 - 1 Arduino簡介

3

- 什麼是**Arduino**?
- **Arduino**是一個開放源始碼的電子原型設計平台，在硬體及軟體上的使用都相當的方便。它適用的範圍相當的廣泛，不管是藝術家、設計師或是有興趣對於不同物體或環境間互相連結的其他業餘愛好者，都非常適使用**Arduino**。



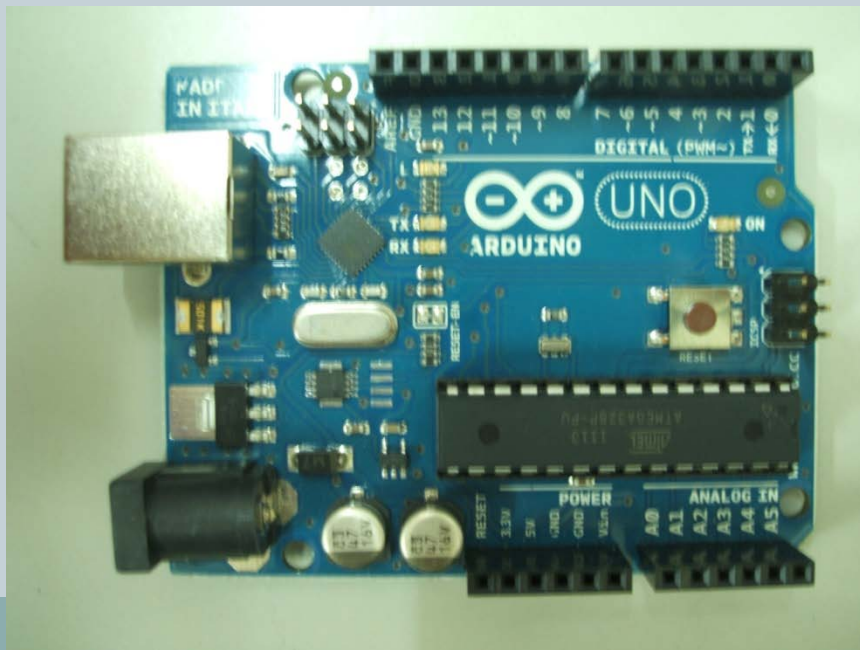
4 - 1 Arduino簡介

- **Arduino**特色：
- **Arduino**是一種基於開放原始碼的軟硬體開發平台，**Arduino**使用類似**Java**的程式語言，開發者只要學具備**C**語言或是**Java**的基礎就可以進行開發，其**IDE**支援的作業系統有**Windows**、**Linux**及**Mac OS X**提供開發者自行選擇。開放原始碼讓**Arduino**的學習資源豐富，除了學習資源豐富之外，**Arduino**也支援大多數的感測元件，只要能夠取得感測元件的規格書都可以使用**Arduino**開發出驅動該感測元件的應用，再搭配上其專用的擴充板就能夠輕易的達到連接網際網路或者是以通訊模組作為傳輸的媒介。硬體售價平易近人也是一個特點，一般大眾皆可購買得到，實現人人皆可自己動手做的理念。

4 - 1 Arduino簡介

5

- 本範例所使用的**Arduino**基本設備：
- **Arduino**控制板、**USB轉Mini USB**的公對公傳輸線。
- 此範例所使用的版本為：**Arduino UNO**。
- 下面左方圖片為：**Arduino**控制板。下面右方圖片為：**USB轉Mini USB**的公對公傳輸線

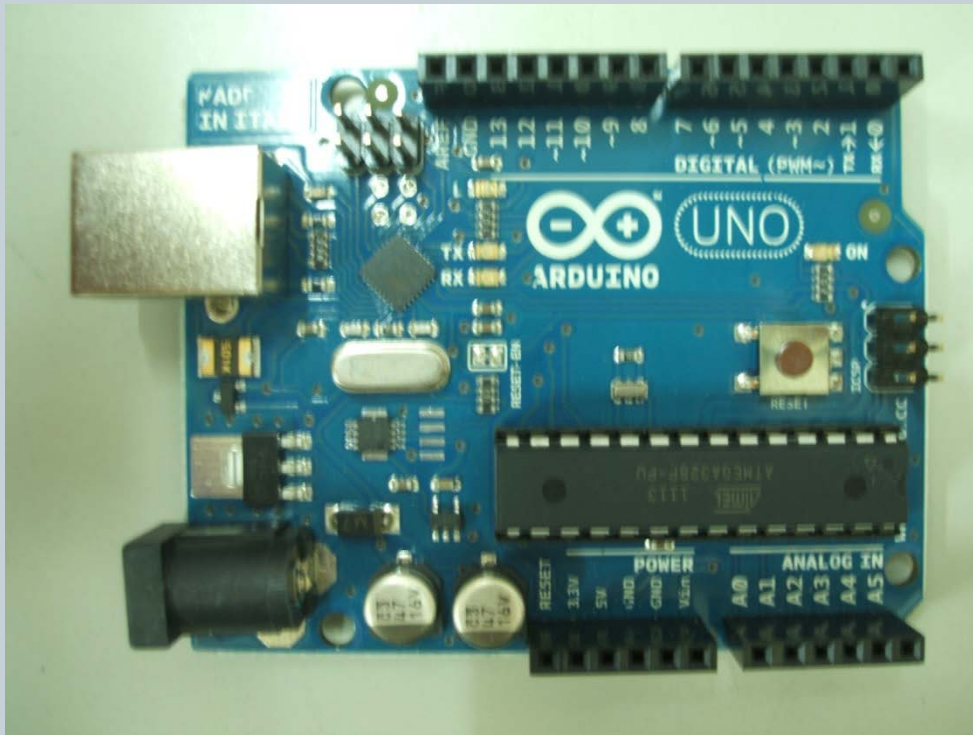


4 - 1 Arduino簡介

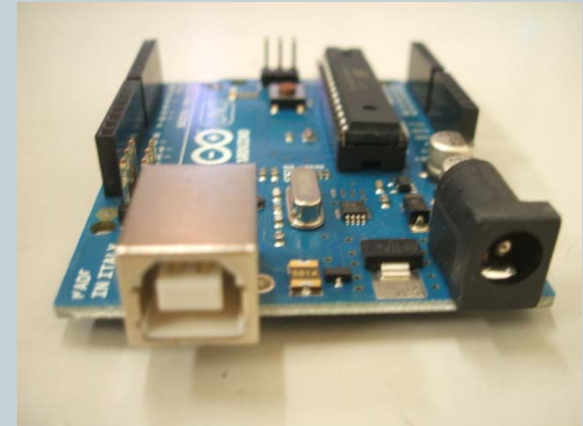
6

- **Arduino**控制板外觀：

俯視圖：使用版本為**Arduino UNO**



側面圖-左：USB與電源插孔



側面圖-右：ICSP針腳



4 - 1 Arduino簡介

7

- **USB轉Mini USB傳輸線外觀：**

俯視圖：USB接頭轉Mini USB接頭



側面圖-左：USB接頭（公）__



側面圖-右：Mini USB接頭（公）



4 - 1 Arduino簡介

8

- 硬體規格：

微控制器	ATmega328
工作電壓	5V
輸入電壓(建議)	7-12V
輸入電壓(限制)	6-20V
數位 I/O Pins	14 支(其中有 6 支腳位可提供 PWM 輸出)
類入 Input Pins	6 支
I/O pin 直流電流	40mA
3.3V pin 直流電流	50mA
Flash 記憶體	32KB, 其中 0.5KB 拿去給 bootloader 使用
SRAM	2KB
EEPROM	1KB
時脈	16MHz

4 - 1 Arduino簡介

9

- 類比輸入 Pins:
- **Arduino Uno** 有 **6** 支類比輸入腳，標記為 **A0** 到 **A5**，每支腳都可提供 **10** 位元的解析 (即 **1024** 種不同的數值)。這些腳位所用的參考電壓預設為 **0** 到 **5V**，不過參考電壓也是可以更改的，方法是透過 **AREF** 腳和 [analogReference\(\)](#) 函式。
- 另外，有幾支腳也有特殊功能:

I2C	4 (SDA) 和 5 (SCL) 這兩支腳。透過 Wire library 可以提供 I2C 通訊。
------------	--

- 其它:

AREF	類比輸入的參考電壓，搭配 analogReference() 函式一起使用。
Reset	當 Reset 腳為 LOW 時，微控制器會重置。

4 - 1 Arduino簡介

10

- 數位 I/O Pins:
- 14 支數位 I/O Pins 可以當作 **input** 使用，也可以當作 **output** 使用，使用方法是透過 [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) 這幾個函式。這 14 支數位 I/O Pins，其中幾支腳有特殊的功能：

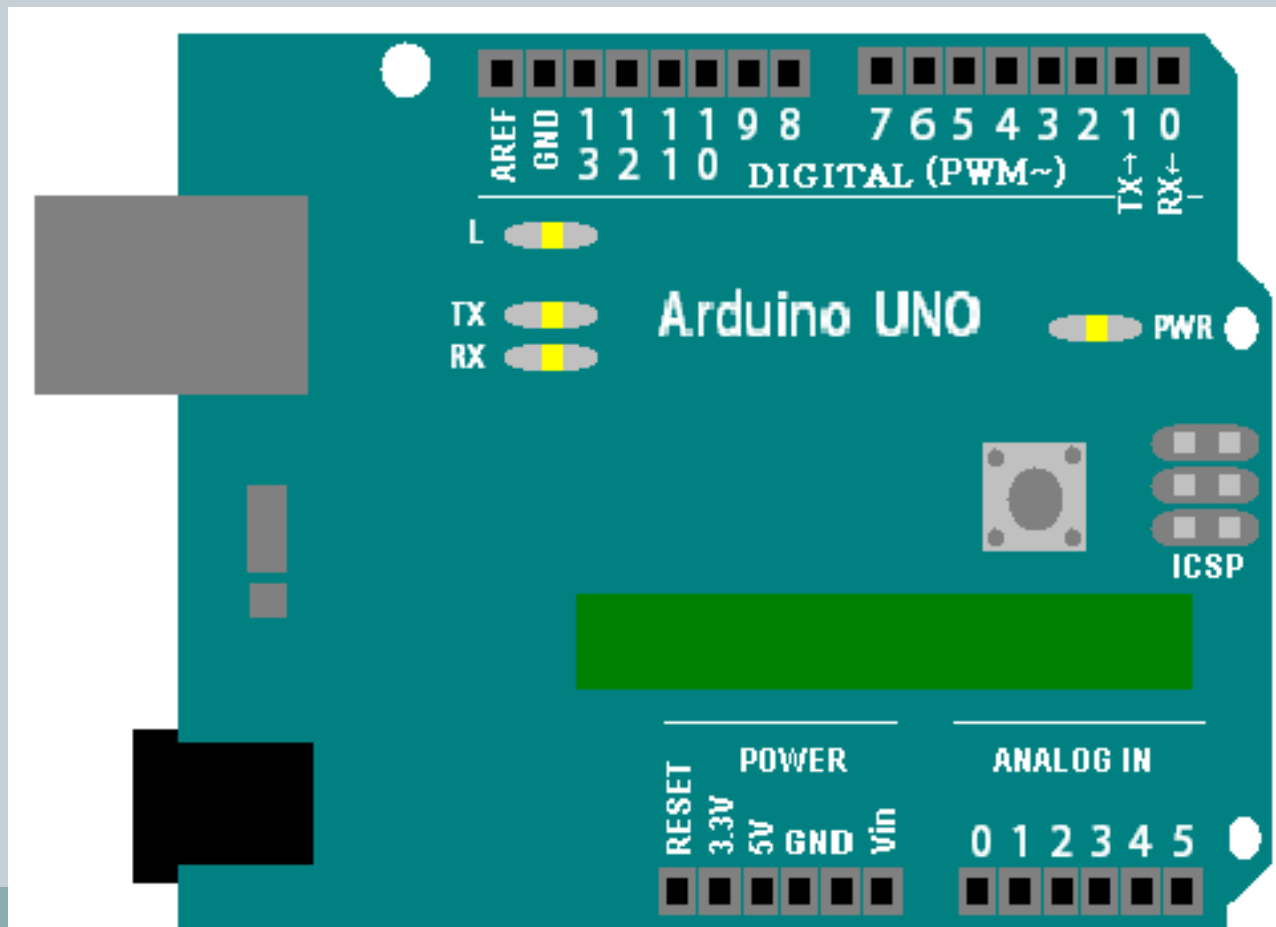
Serial 通訊	0(RX) 和 1 (TX) 這兩支腳。用來接收(RX)與傳輸(TX) TTL 訊號的序列資料。這兩支腳也連接到 USB Converter 晶片中。
外部中斷	2 和 3 這兩支腳。這兩支腳可以利用外部事件觸發中斷。詳細內容請參考 attachInterrupt() 函式。
PWM	3, 5, 6, 9, 10 和 11 共六支腳。透過 analogWrite() 函式可以提供 8-bit 的 PWM 輸出。
SPI	10 (SS), 11 (MOSI), 12 (MISO) 和 13 (SCK) 這四支腳。這四支腳搭配 SPI Library 可提供 SPI 序列通訊。
LED	13。內建一顆 LED，當 pin 腳為 HIGH 時，LED 打開，當 pin 腳為 LOW 時，LED 關閉。

引用至：<http://coopermaa2nd.blogspot.tw/2010/12/arduino-arduino.html>

4 - 1 Arduino簡介

11

- Arduino控制板針腳平面圖



4 - 2 建立Arduino開發環境

12

- 我們可以由此網址「<http://arduino.cc>」連結到Arduino的官方網站。首頁畫面，如下面圖片所示。
- 在首頁上的分頁選單中包含以下七個分頁連結點：「**Buy**」（購買）、「**Download**」（下載）、「**Getting**」（入門）、「**Started Learning**」（開始學習）、「**Reference**」（參考）、「**Hardware**」（硬體）、「**FAQ**」（常見問題），我們選擇的「**Download**」頁面，即可進入軟體的下載頁面。



4 - 2 建立Arduino開發環境

13

- 點選「**Download**」分頁面後，網頁畫面如下圖所示。
- 若讀者想查閱有關**Arduino**發行版本的相關說明可在點擊畫面中的**Arduino 1.0.1 (release notes)**的超連結。
- **Arduino**可支援的作業系統有**Windows**、**Mac OS X**、**Linux: 32 bit, 64 bit**。
- 在此分頁的中間部分，可以找到**Arduino**編輯軟體 (此時最高版本為**Arduino 1.0.1**)的下載點，並依照讀者的作業系統之規格，選擇適合的版本。
- 此範例的作業系統為X P，故我們選擇「**Windows**」版本後，即可下載**Arduino**編輯軟體。

Download

Arduino 1.0.1 ([release notes](#)), hosted by [Google Code](#):

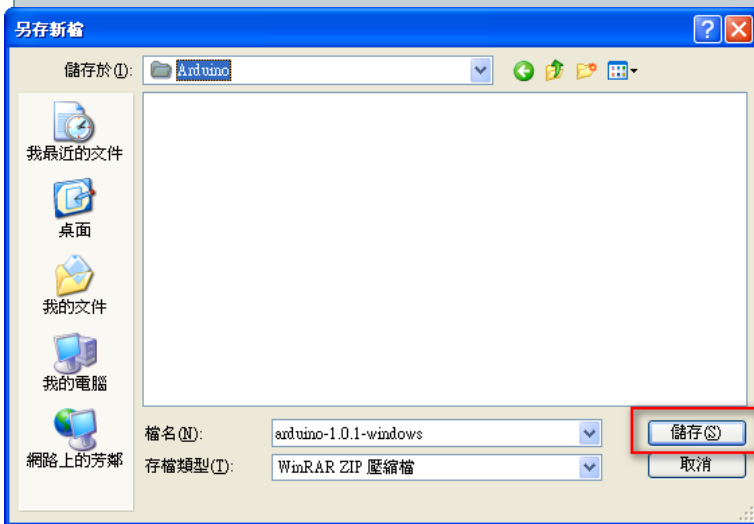
- + [Windows](#)
- + [Mac OS X](#)
- + [Linux: 32 bit, 64 bit](#)
- + [source](#)

Downloads will be available from [arduino.cc](#) soon.

4 - 2 建立Arduino開發環境

14

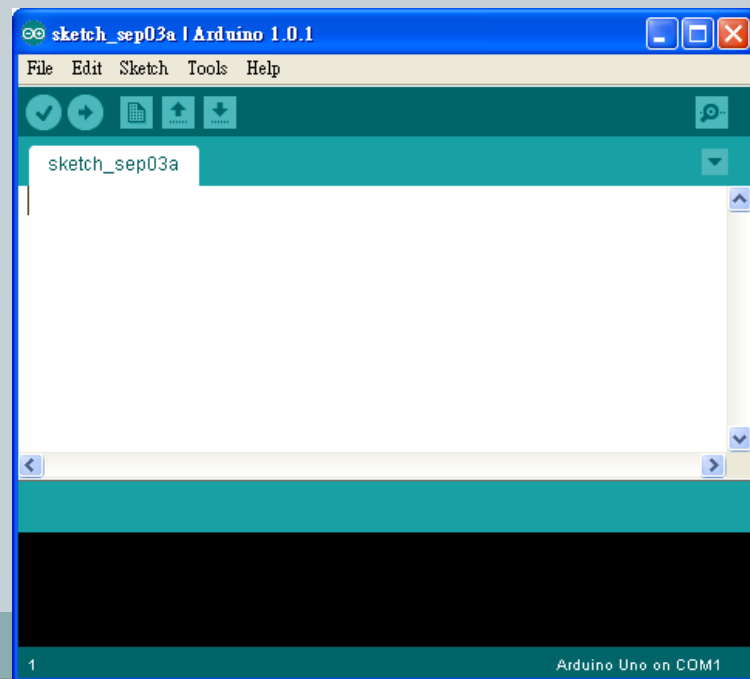
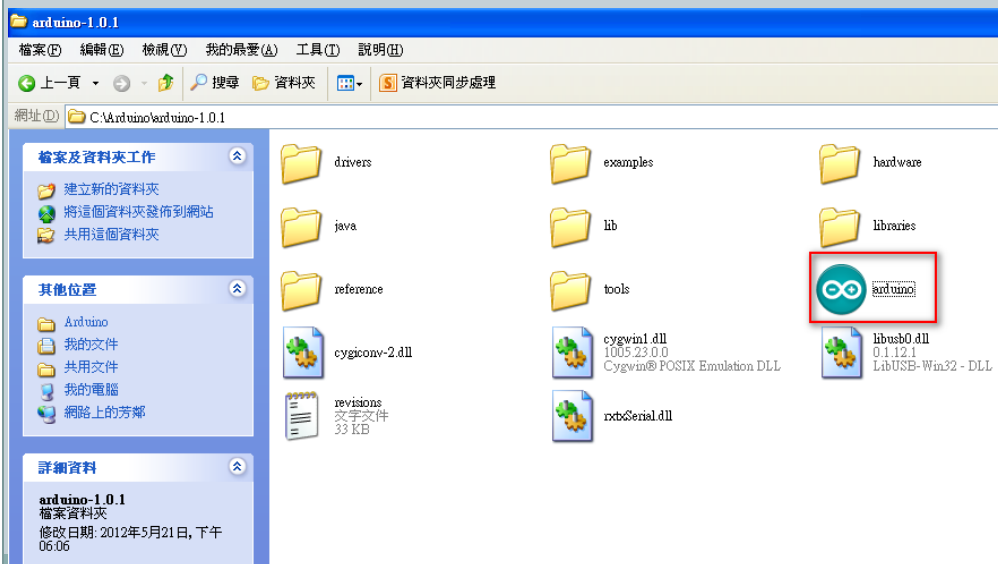
- 點選下載版本並選擇存檔的路徑後，即可下載**Arduino**編輯軟體，其畫面如下圖所示。此範例下載路徑為：**C:\Arduino**。
- 下載檔案完成後，將「**Arduino 1.0.1.zip**」解壓縮即可。



4 - 2 建立Arduino開發環境

15

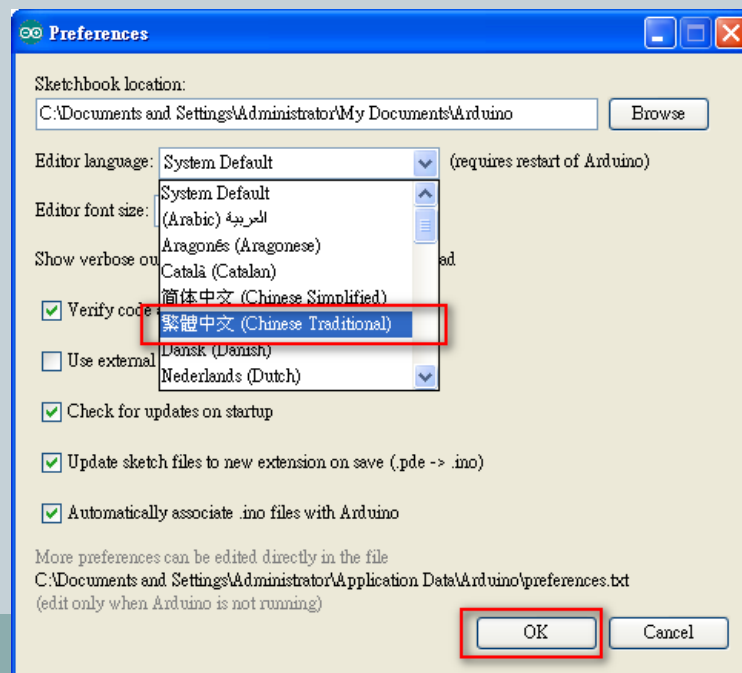
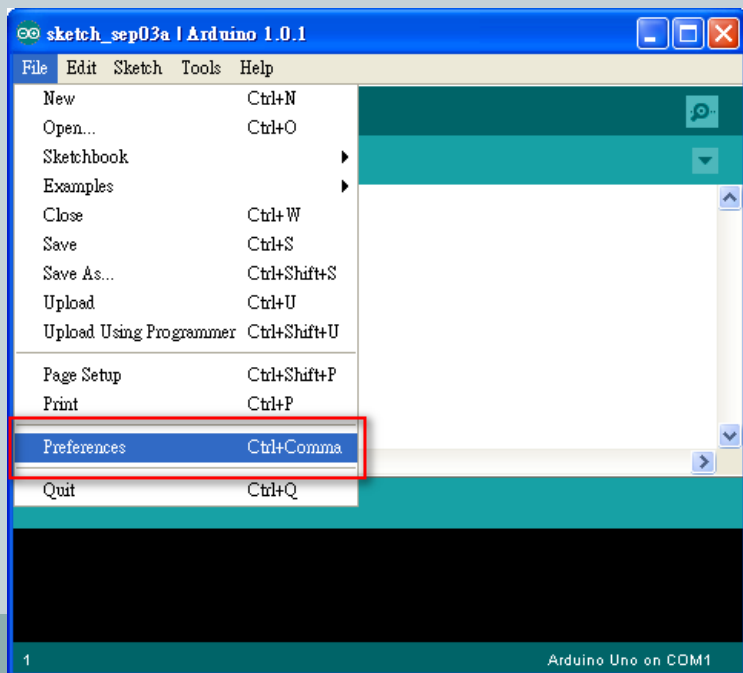
- 進入剛剛所解壓縮後的「**Arduino 1.0.1**」資料夾中後，其畫面如左下圖所示。
- 在資料夾中找到**Arduino**編輯軟體的執行檔「**Arduino.exe**」後，即可點擊左鍵兩下執行，執行後的畫面如右下圖所示。



4 - 2 建立Arduino開發環境

16

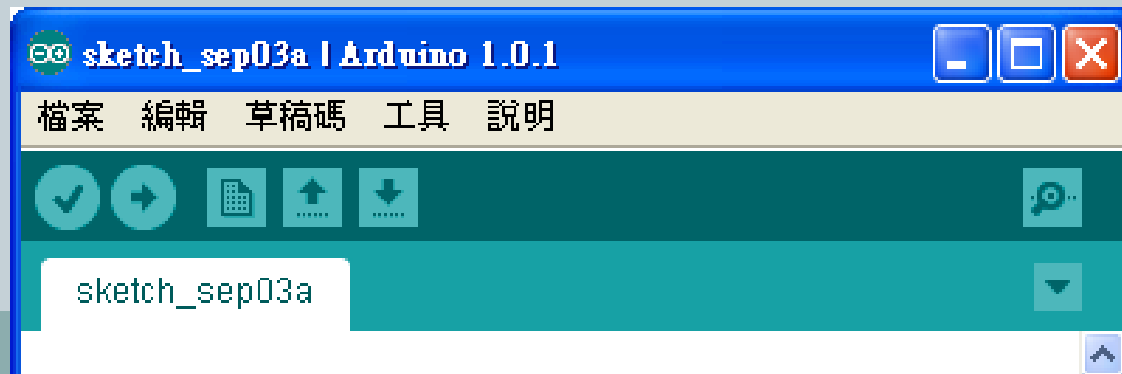
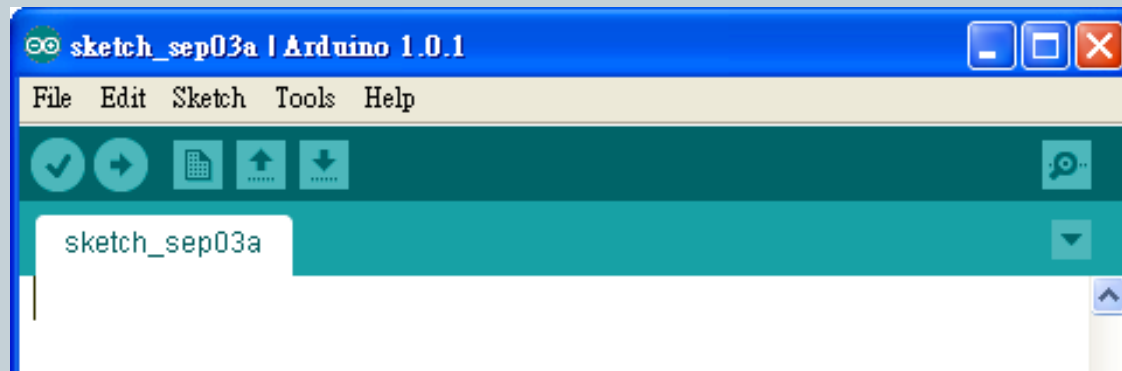
- **Arduino**編輯軟體的介面語言預設值為英文，若讀者想跟改語言可以至「**File**」→「**Preferenecs**」→「**Editor language**」下更改自己所以慣用的語言模式後，重新啟動編輯軟體即可。
- 此範例為：「**繁體中文 (Chinese Traditional)**」。



4 - 2 建立Arduino開發環境

17

- 將語言模式更改為「繁體中文 (Chinese Traditional)」後，並重新啟動編輯軟體的畫面如下圖所示。
- 在完成編輯軟體的設定後，可以接著安裝Arduino 的驅動程式。



4 - 2 建立Arduino開發環境

18

- 在安裝**Arduino** 的驅動程式前，必須先準備「**Arduino**的控制板」及「**USB**傳輸線」。如下圖所示。



4 - 2 建立Arduino開發環境

19

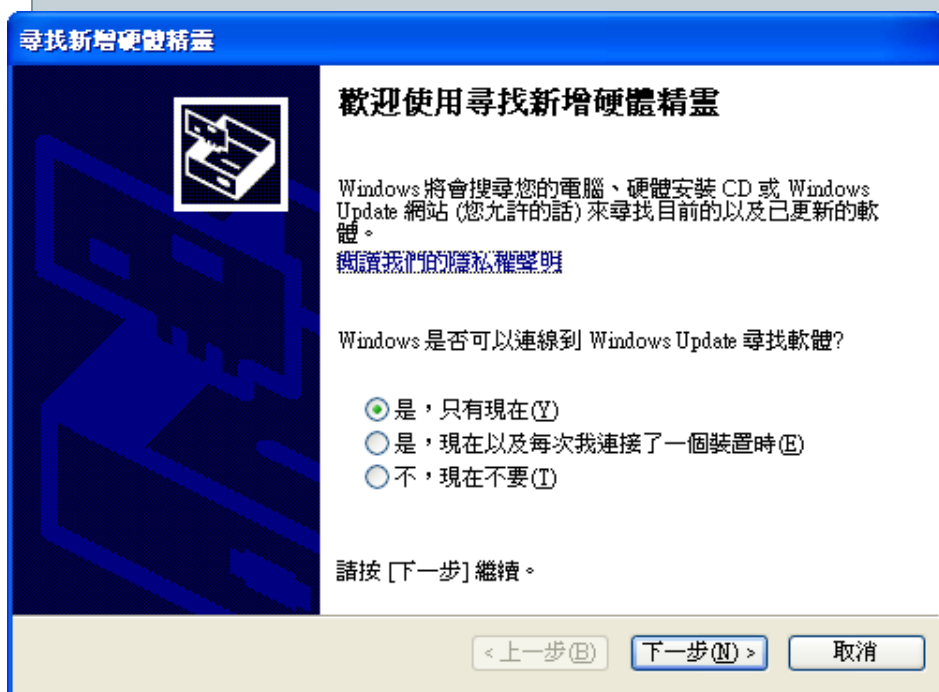
- 在利用**USB**傳輸線，將「**Arduino**控制板」與電腦的「**USB Port**」作連接。如下圖所示。



4 - 2 建立Arduino開發環境

20

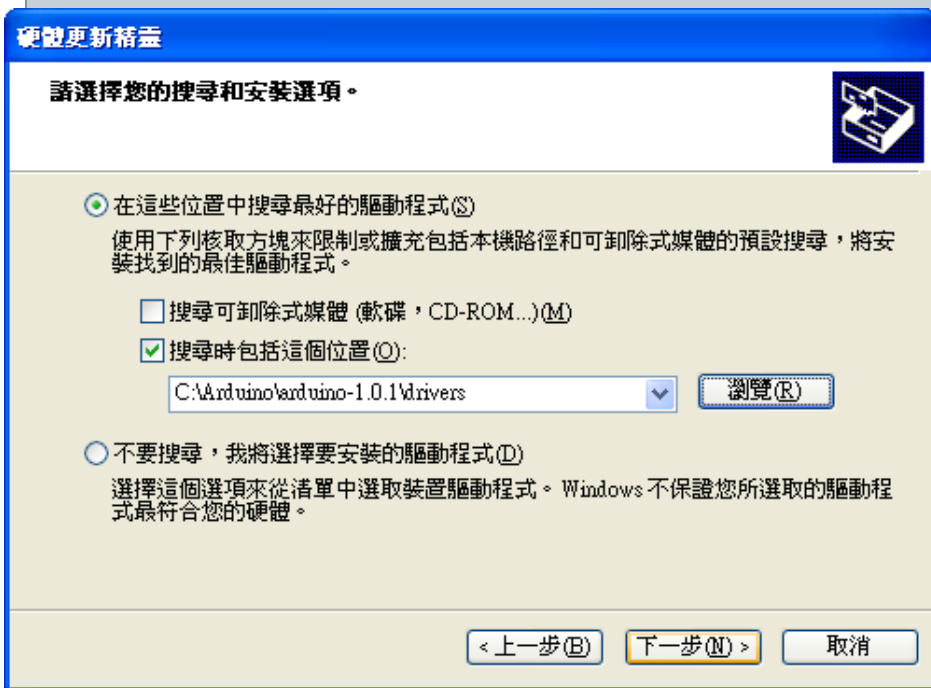
- 將「Arduino控制板」與電腦的「USB Port」作連接後，會自動出現「搜尋新增硬體精靈」，如左下圖所示。
- 接著依序選擇「是，只有現在(Y)」→「下一步(N)」→「從清單或特定位位置安裝(進階)(S)」→「下一步(N)」。



4 - 2 建立Arduino開發環境

21

- 選擇「在這些位置中搜尋最好的驅動程式」(S) → 勾選「搜尋時包括這個位置(O)」 → 「預覽(R)」 → 選擇Arduino 1.0.1 資料夾下的「drivers」 → 「下一步(N)」，即可看到「Communications Port」的安裝畫面。
- 此範例「drivers」的路徑為：C:\Arduino\arduino-1.0.1\drivers。



4 - 2 建立Arduino開發環境

22

- 若安裝「**Communications Port**」的過程出現警告時，則選擇「繼續安裝(C)」，當出現「正在完成硬體更新精靈」時，即可點選「完成」，已結束Arduino的驅動安裝。

硬體安裝



您正要為這個硬體安裝的軟體：

Communications Port

尚未通過 Windows 標誌測試以確認它與 Windows XP 的相容性。[\(告訴我這項測試的重要性。\)](#)

繼續安裝這個軟體會在現在或將來，使您的系統操作不穩定或受損。Microsoft 強烈建議您立即停止這項安裝，並連絡硬體廠商索取已通過 Windows 標誌測試的軟體。

繼續安裝(C)

停止安裝(S)

硬體更新精靈



正在完成硬體更新精靈

這個精靈安裝了軟體於：



Communications Port

按 [完成] 關閉精靈。

< 上一步(B)

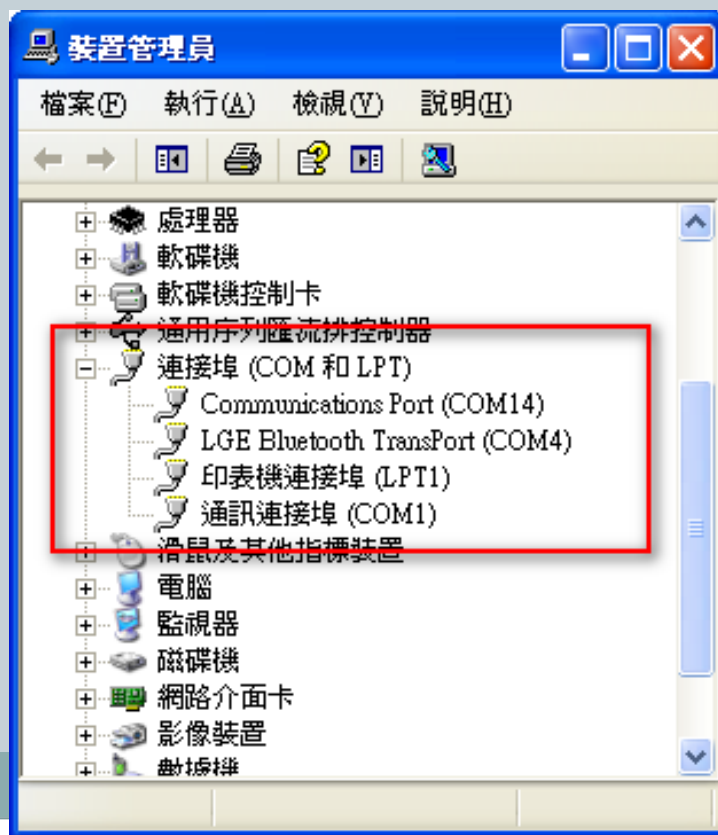
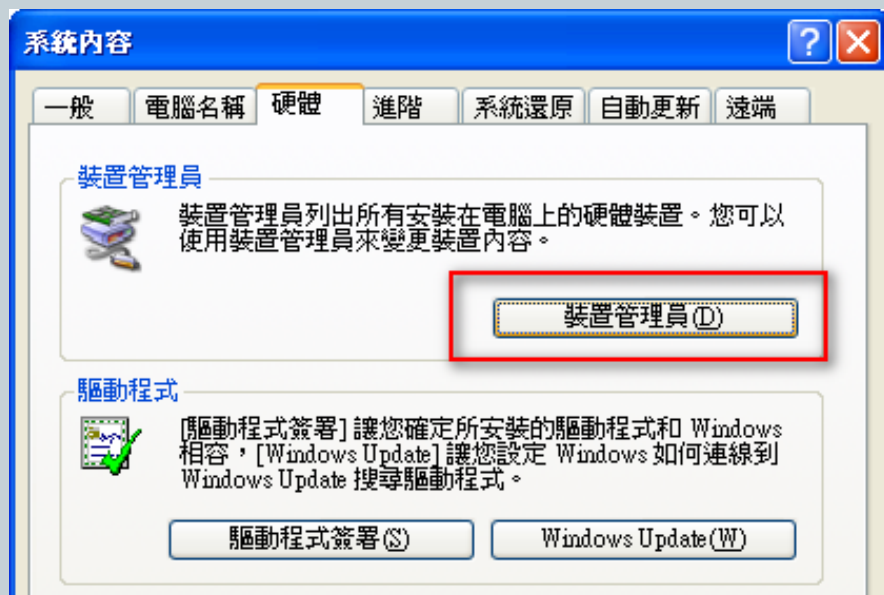
完成

取消

4 - 2 建立Arduino開發環境

23

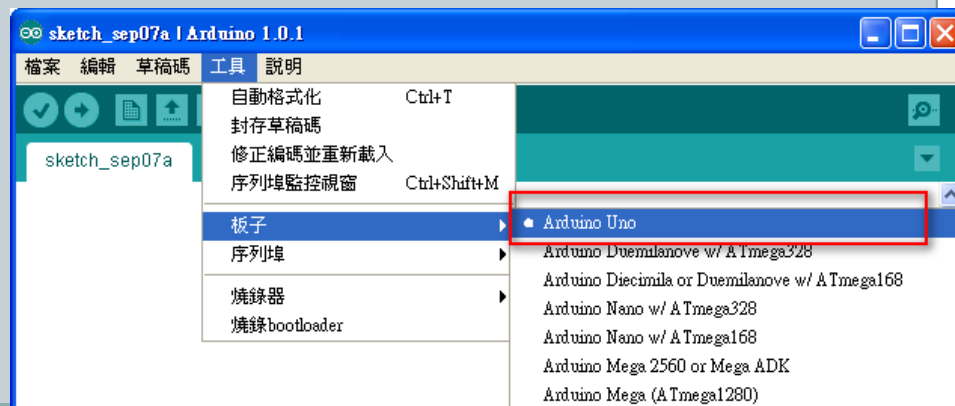
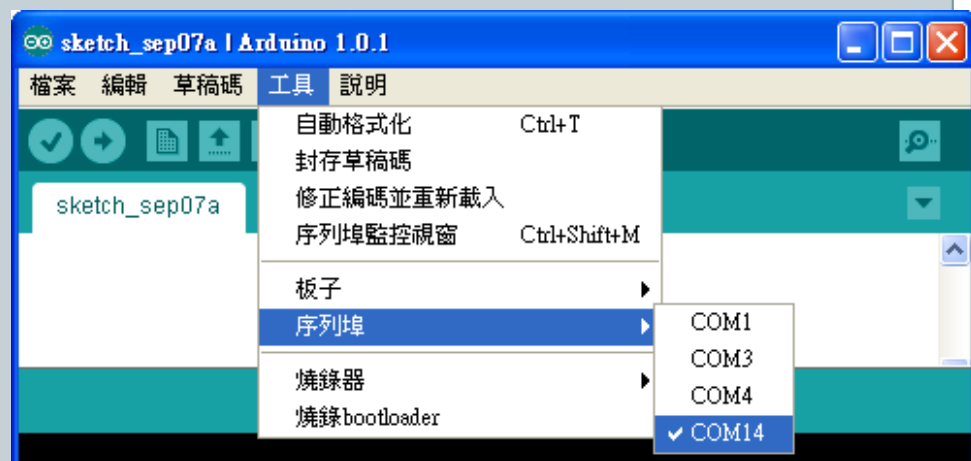
- 已結束Arduino的驅動安裝後，可以至「我的電腦」點擊滑鼠右鍵→「內容」→「硬體」→「裝置管理員(D)」→「連接埠(COM和LPT)」的裝置中，確認是否安裝成功。
- 若有出現「Communications Port (COM XX)」，則表示已完成arduino開發環境的建置。
- COM XX(Port)會隨著使用者所連接的USB連接埠而有所不同，此範例所連接的位置為COM 1 4 的位置。



4 - 2 建立Arduino開發環境

24

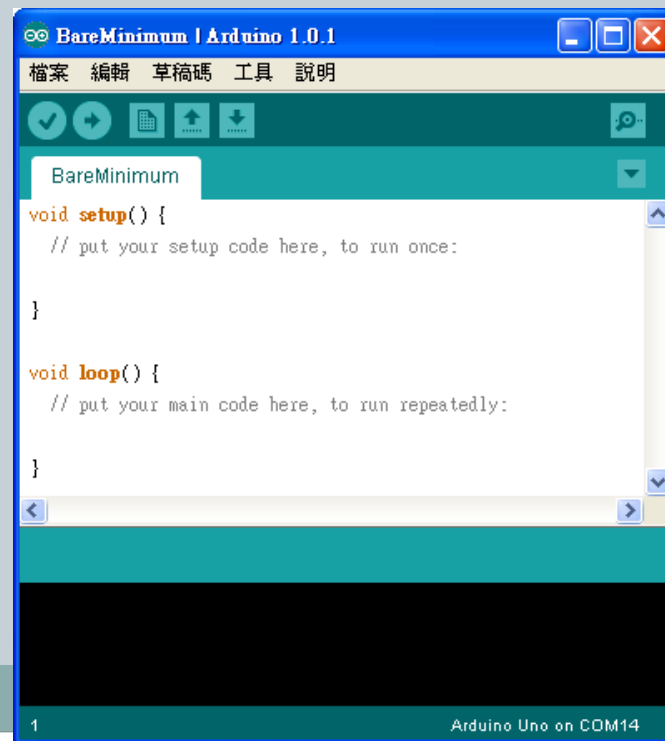
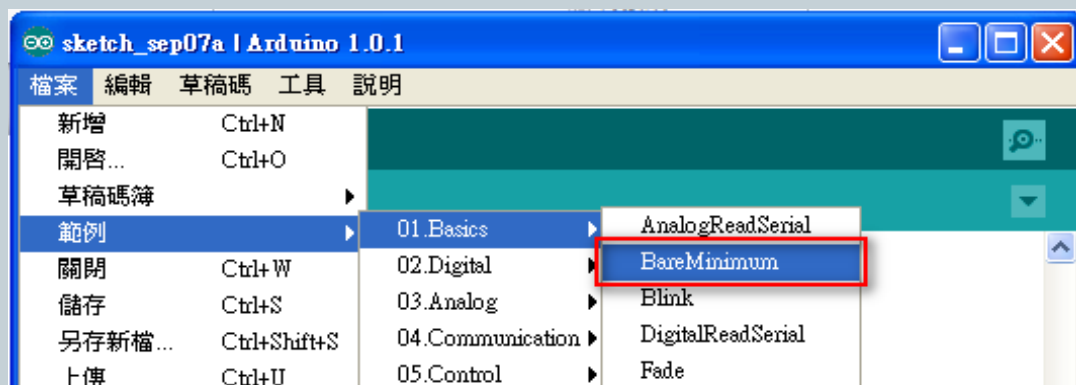
- 完成Arduino的驅動安裝，並設定「**Communications Port**」與Arduino控制板之規格的對應。
- 首先確認「**Communications Port**」。選擇Arduino編輯軟體的上列工具選單的「工具」→「序列埠」，選擇所對應的序列埠，即可進行控制板的對應設定。
- 此範例對應位置為：**COM 1 4**。
- 然後確認我們所使用的Arduino控制板的規格。選擇Arduino編輯軟體的上列工具選單的「工具」→「板子」，即可進行控制板的對應設定。
- 此範使用的控制板為：**Arduino UNO**。



4 - 3 利用Arduino 開發第一個程式

25

- 在完成Arduino的「**Communications Port**」與Arduino控制板之規格的對應後，即可進行Arduino程式的開發。
- **Arduino**編輯軟體內建了許多的範例檔案，可以提供開發者進行開發，增加開發者們的便利性，在此先利用範例檔來了說明**Arduino**程式的主要架構。
- 選擇**Arduino**編輯軟體的上列工具選單的「檔案」→「範例」→「01·Basice」→「**BareMinimum**」的範例檔案。



4 - 3 利用Arduino 開發第一個程式

26

- Arduino程式的主要架構分為「**void setup() { // put your setup code here, to run once: }**」與「**void loop() { // put your main code here, to run repeatedly: }**」兩個，在「**void setup() { }**」的執行次數為：1次，主要用來初始化程式內相關的變數。「**void loop() { }**」的執行次數為：無窮迴圈，所以程式主要執行方法都幾乎都是寫在「**void loop() { }**」裡面。

BareMinimum程式碼

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

4 - 3 利用Arduino 開發第一個程式

27

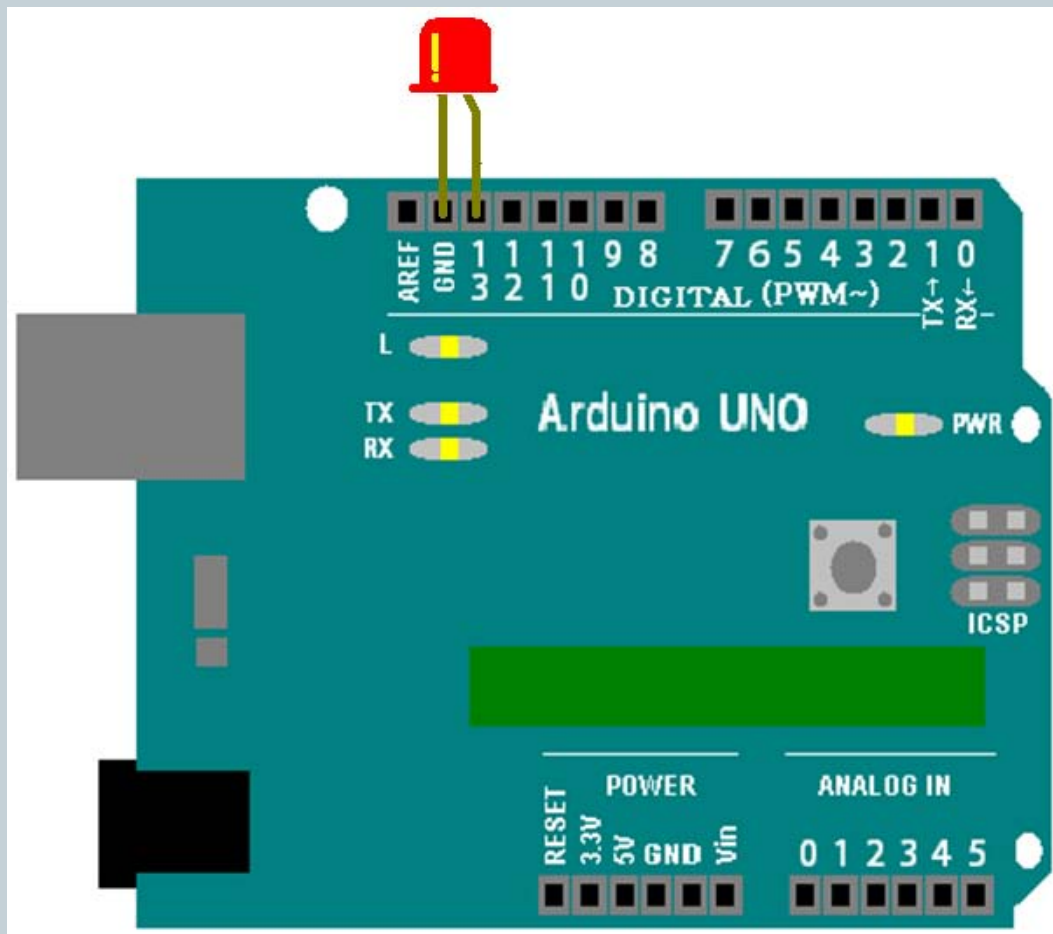
- 了解了**Arduino**程式的主要架構後，就可以執行第二個範例程式。
- 程式名稱：**Blink**。
- 實驗目的：利用程式控制，使L E D燈擁有閃爍的效果。
- 所需材料：

圖示	數量	名稱
	1	L E D燈（任何顏色皆可）
	1	Arduino 控制板

4 - 3 利用Arduino 開發第一個程式

28

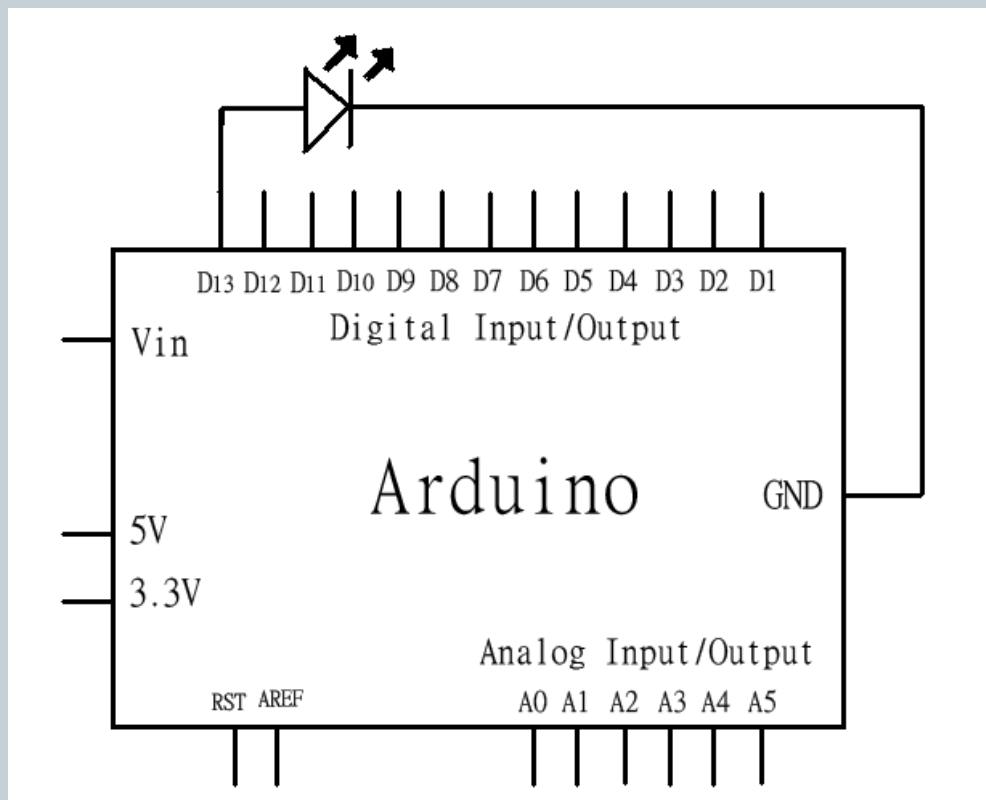
- 接線配置方式：
- 將**LED**較短的針腳(陰極)接於**DIGITAL (PWM~)的Pin GND**，另外一隻較長的針腳(陽極)接於**DIGITAL (PWM~) 的Pin 13**即可。
- 如右圖所示：



4 - 3 利用Arduino 開發第一個程式

29

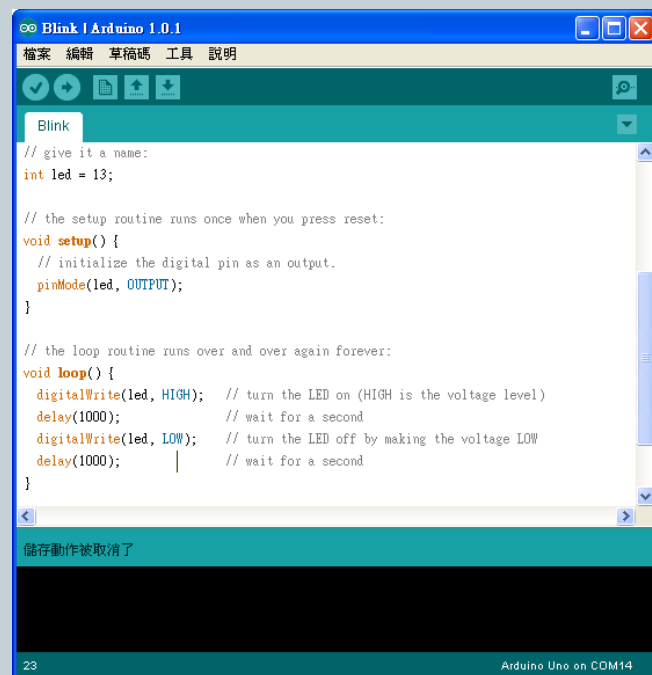
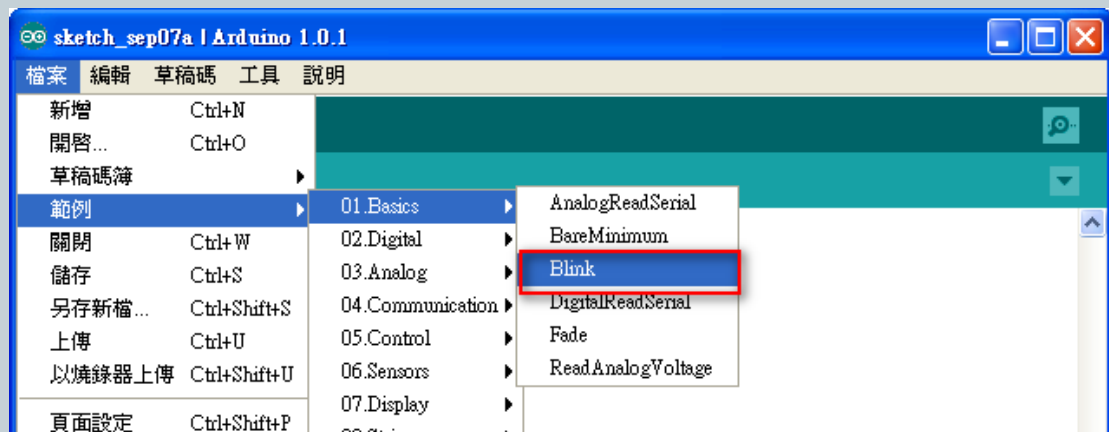
- 電路圖：
- 將**LED**較短的針腳(陰極)接於**DIGITAL (PWM~)**的**Pin GND**，另外一隻較長的針腳(陽極)接於**DIGITAL (PWM~)**的**Pin 13**即可。
- 如右圖所示：



4 - 3 利用Arduino 開發第一個程式

30

- 當接好LED燈並將Arduino控制板連接至電腦後，後選擇Arduino編輯軟體上列工具選單的「檔案」→「範例」→「01 · Basice」→「Blink」的範例檔案。



4 - 3 利用Arduino 開發第一個程式

31

- 在開發**Arduino**之前必須先了解**Arduino**的基本語法語法。

語法	說明
HIGH	高電位。 當模式為 OUTPUT 時，電位為：5 伏特。 當模式為 INPUT 時，輸入電位必須：3 伏特以上。
LOW	低電位。 當模式為 OUTPUT 時，電位為：0 伏特。 當模式為 INPUT 時，輸入電位必需：2 伏特以下。
delay()	延遲時間，參數所設定的單位為毫秒， delay(ms) 。
pinMode()	設定 Pin 的模式為 OUTPUT 或 INPUT ， pinMode(pin, mode) 。
digitalWrite()	設定 Pin 值為高電位或低電位。 模式為 OUTPUT 時： 設定輸出值為 HIGH ，表示則輸出電壓為 5V 或 3.3V 。 設定輸出值為 LOW ，表示則輸出電壓為 0V 或接地。

詳細說明可參照教材後面的：**Arduino**變數與方法說明

4 - 3 利用Arduino 開發第一個程式

32

- 13行：宣告整數變數 **led** 值為 **13**。
- 15行：**pinMode(led, OUTPUT);**設定**pin 13**的模式為**output(輸出)**。
- 20行：提供**pin 13** **HIGH(5V)**的電壓，以推動**LED燈**。
- 21行、23行：設定延遲時間為**1000毫秒(1秒)**。
- 22行：提供**pin 13** **LOW(0V)**的電壓，以關閉**LED燈**。

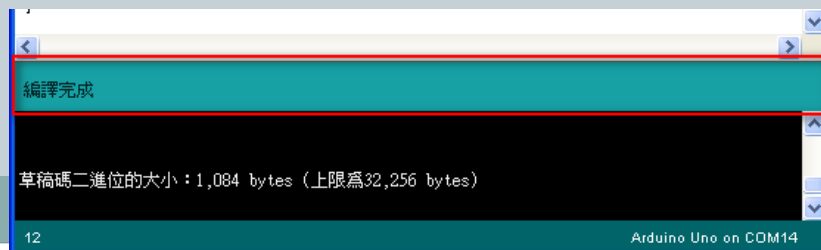
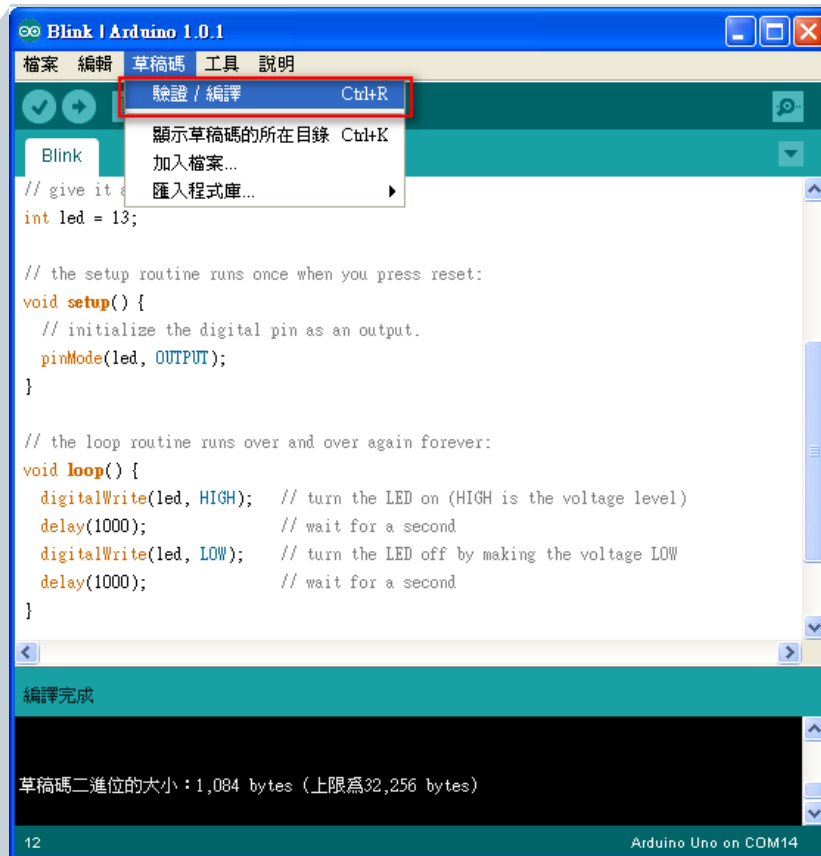
Blink程式碼

```
1  /*
2   Blink
3   Turns on an LED on for one second, then off for one second, repeatedly.
4
5   This example code is in the public domain.
6   */
7
8   // Pin 13 has an LED connected on most Arduino boards.
9   // give it a name:
10  int led = 13;
11
12  // the setup routine runs once when you press reset:
13  void setup() {
14    // initialize the digital pin as an output.
15    pinMode(led, OUTPUT);
16  }
17
18  // the loop routine runs over and over again forever:
19  void loop() {
20    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
21    delay(1000);             // wait for a second
22    digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
23    delay(1000);             // wait for a second
24  }
```

4 - 3 利用Arduino 開發第一個程式

33

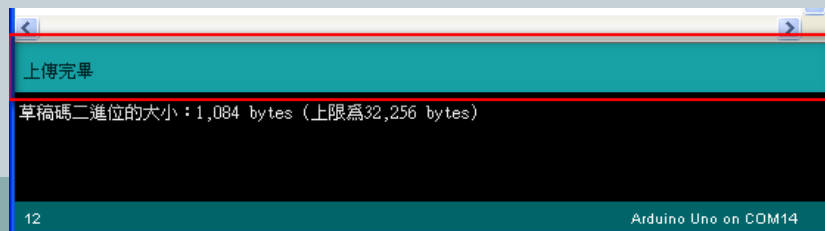
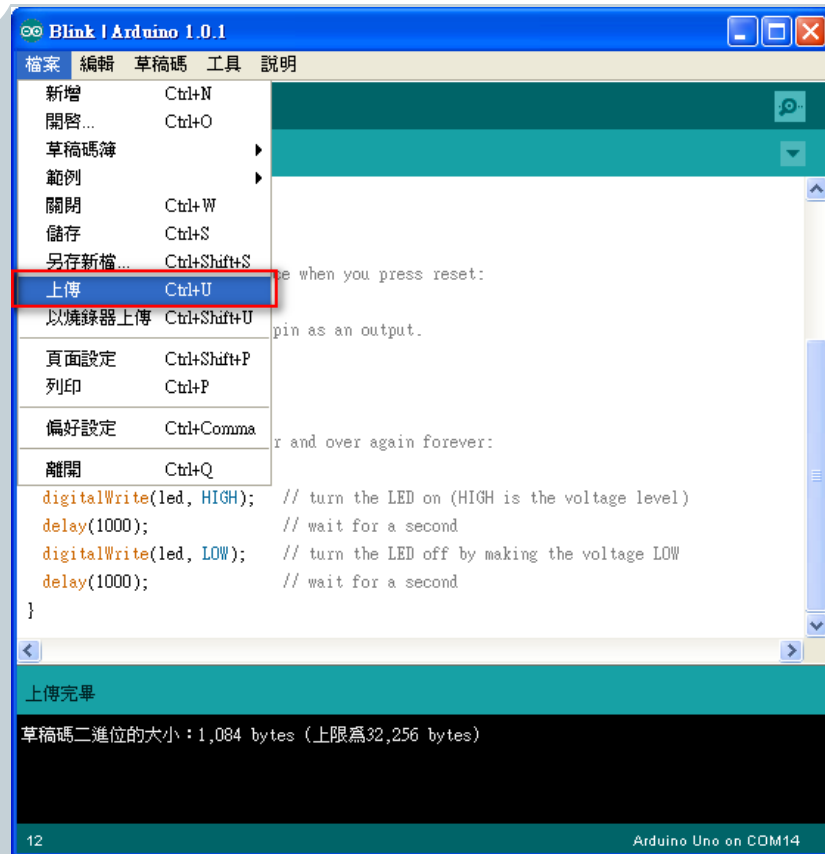
- 程式撰寫完畢後，可以利用Arduino編輯軟體上列工具選單的「草稿碼」→「驗證/編譯 Ctrl+R」，進行程式碼的編譯。
- 如編譯成功，下列的提示字串會顯示「編譯完成」。
- 若編譯失敗，下列的提示字串則會顯示編譯失敗的提示訊息。



4 - 3 利用Arduino 開發第一個程式

34

- 程式碼編譯完畢後，就可以利用**Arduino**編輯軟體上列工具選單的「檔案」→「上傳 **Ctrl+U**」，將程式的寫入至 **Arduino** 的控制板上。
- 如上傳成功，下列的提示字串會顯示「上傳完畢」。
- 若上傳失敗，下列的提示字串則會顯示上傳失敗的提示訊息。



4 - 3 利用Arduino 開發第一個程式

35

- 成果及說明如下：
- 利用**digitalWrite()**的方法，將**pin 13**的腳位設定為**HIGH (5V)**，以提供**LED HIGH (5V)**的電壓使**LED**亮起，如左圖所示，
- 利用**digitalWrite()**的方法，將**pin 13**的腳位設定為**LOW (0V)**，以提供**LED LOW (0V)**的電壓使**LED**熄滅，如右圖所示。
- 再利用與**delay()**的方法，將延遲時間設為**1000毫秒(1秒)**，讓**LED**亮滅之間有**1000毫秒(1秒)**延遲時間，產生出**LED**閃爍的效果。



Arduino變數與方法說明-Structure

36

Structure setup() loop() Control Structures if if...else for switch case while do... while break continue return goto	Further Syntax ; (semicolon) { } (curly braces) // (single line comment) /* */ (multi-line comment) #define #include Arithmetic Operators = (assignment operator) + (addition) - (subtraction) * (multiplication) / (division) % (modulo)	Comparison Operators == (equal to) != (not equal to) < (less than) > (greater than) <= (less than or equal to) >= (greater than or equal to) Boolean Operators && (and) (or) ! (not) Pointer Access Operators * dereference operator & reference operator	Bitwise Operators & (bitwise and) (bitwise or) ^ (bitwise xor) ~ (bitwise not) << (bitshift left) >> (bitshift right) Compound Operators ++ (increment) -- (decrement) += (compound addition) -= (compound subtraction) *= (compound multiplication) /= (compound division) &= (compound bitwise and) = (compound bitwise or)
--	--	--	---

Arduino變數與方法說明-Variables

37

Constants

HIGH | LOW

INPUT |

OUTPUT|INPUT_PULLUP

true | false

integer constants

floating point constants

Data Types

void

boolean

char

unsigned char

byte

int

unsigned int

word

long

unsigned long

float

double

string - char array

String - object

array

Variable Scope & Qualifiers

variable scope

static

volatile

const

Utilities

sizeof()

Conversion

char()

byte()

int()

word()

long()

float()

Arduino變數與方法說明-Functions

38

Functions

Digital I/O
pinMode()
digitalWrite()
digitalRead()

Analog I/O

analogReference()
analogRead()
analogWrite() - PWM

Advanced I/O

tone()
noTone()
shiftOut()
shiftIn()
pulseIn()

Random Numbers

randomSeed()
random()

Time

millis()
micros()
delay()
delayMicroseconds()

Math

min()
max()
abs()
constrain()
map()
pow()
sqrt()

Trigonometry

sin()
cos()
tan()

Bits and Bytes

lowByte()
highByte()
bitRead()
bitWrite()
bitSet()
bitClear()
bit()

External Interrupts

attachInterrupt()
detachInterrupt()

Interrupts

interrupts()
noInterrupts()

Communication

Serial
Stream

Leonardo Specific

Keyboard
Mouse

Looking for something else?

See the libraries page for interfacing with particular types of hardware. Try the list of community-contributed code. The Arduino language is based on C/C++. It links against AVR Libc and allows the use of any of its functions; see its user manual for details.

Arduino變數與方法說明

39

Defining Pin Levels, HIGH and LOW

When reading or writing to a digital pin there are only two possible values a pin can take/be-set-to: HIGH and LOW.

LOW

The meaning of LOW also has a different meaning depending on whether a pin is set to INPUT or OUTPUT. When a pin is configured as an INPUT with `pinMode`, and read with `digitalRead`, the microcontroller will report LOW if a voltage of 2 volts or less is present at the pin.

When a pin is configured to OUTPUT with `pinMode`, and set to LOW with `digitalWrite`, the pin is at 0 volts. In this state it can sink current, e.g. light an LED that is connected through a series resistor to, 5 volts, or to another pin configured as an output, and set to HIGH.

HIGH

The meaning of HIGH (in reference to a pin) is somewhat different depending on whether a pin is set to an INPUT or OUTPUT. When a pin is configured as an INPUT with `pinMode`, and read with `digitalRead`, the microcontroller will report HIGH if a voltage of 3 volts or more is present at the pin.

A pin may also be configured as an INPUT with `pinMode`, and subsequently made HIGH with `digitalWrite`, this will set the internal 20K pullup resistors, which will steer the input pin to a HIGH reading unless it is pulled LOW by external circuitry. This is how `INPUT_PULLUP` works as well

When a pin is configured to OUTPUT with `pinMode`, and set to HIGH with `digitalWrite`, the pin is at 5 volts. In this state it can source current, e.g. light an LED that is connected through a series resistor to ground, or to another pin configured as an output, and set to LOW.

Arduino變數與方法說明

(40)

delay()

Description

Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)

Syntax

delay(ms)

Parameters

ms: the number of milliseconds to pause (*unsigned long*)

Returns

nothing

Example

```
int ledPin = 13;           // LED connected to digital pin 13
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Arduino變數與方法說明

41

pinMode()

Description

Configures the specified pin to behave either as an input or an output. See the description of [digital pins](#) for details on the functionality of the pins. As of Arduino 1.0.1, it is possible to enable the internal pullup resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pullups.

Syntax	Parameters	Returns
pinMode(pin, mode)	pin: the number of the pin whose mode you wish to set mode: INPUT , OUTPUT , or INPUT_PULLUP . (see the digital pins page for a more complete description of the functionality.)	None

Arduino變數與方法說明

42

Example

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Note :

The analog input pins can be used as digital pins, referred to as A0, A1, etc.

Arduino變數與方法說明

43

digitalWrite()

Description

Write a [HIGH](#) or a [LOW](#) value to a digital pin.

If the pin has been configured as an OUTPUT with [pinMode\(\)](#), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

If the pin is configured as an INPUT, writing a HIGH value with digitalWrite() will enable an internal 20K pullup resistor (see the [tutorial on digital pins](#)). Writing LOW will disable the pullup. The pullup resistor is enough to light an LED dimly, so if LEDs appear to work, but very dimly, this is a likely cause. The remedy is to set the pin to an output with the pinMode() function.

NOTE: Digital pin 13 is harder to use as a digital input than the other digital pins because it has an LED and resistor attached to it that's soldered to the board on most boards. If you enable its internal 20k pull-up resistor, it will hang at around 1.7 V instead of the expected 5V because the onboard LED and series resistor pull the voltage level down, meaning it always returns LOW. If you must use pin 13 as a digital input, use an external pull down resistor.

Arduino變數與方法說明

44

Syntax	Parameters	Returns
<code>digitalWrite(pin, value)</code>	pin: the pin number value: HIGH or LOW	none

Example

```
int ledPin = 13;           // LED connected to digital pin 13
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Note :

Sets pin 13 to HIGH, makes a one-second-long delay, and sets the pin back to LOW.

引用自：<http://arduino.cc/en/Reference/HomePage>

參考資料

45

- <http://www.arduino.cc/>
- Arduino臺灣使用者社群。 <http://arduino.tw/>
- <http://coopermaa2nd.blogspot.tw/>
- 孫駿榮、吳明展、盧聰勇。最簡單的互動設計 Arduino一試就上手。碁峰。2010。