

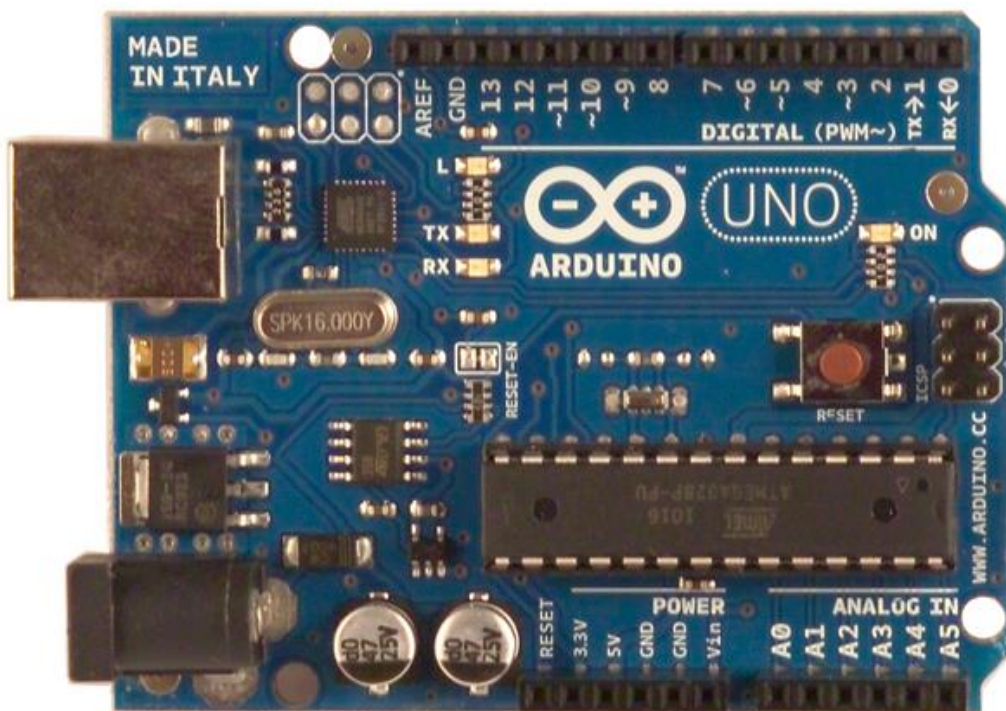


Arduino 編程

## 1. Arduino 簡介及安裝方法

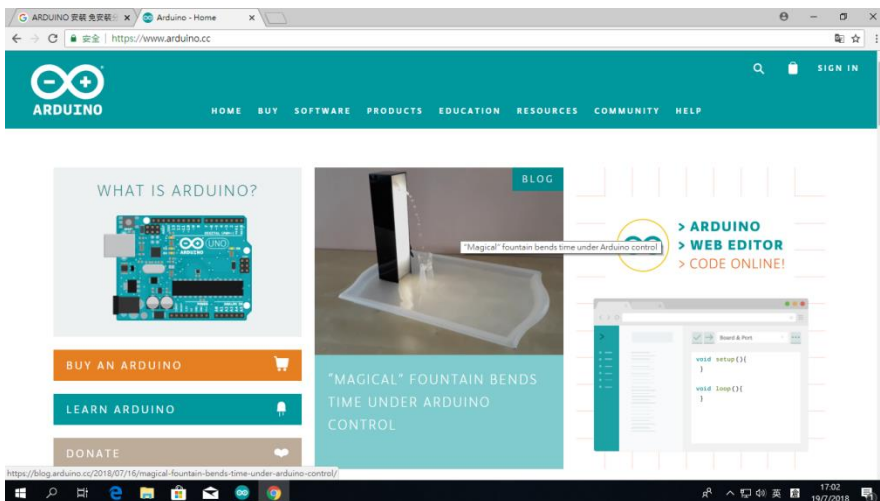
Arduino 是一張細小的微控制器板子。我們可以在 Arduino 板子上接上各種電子裝置，例如 LED 燈、光敏電阻、溫濕度感測器等，再配合編寫一些控制程式，就能做出各式各樣的產品，例如使用光敏電阻制作智能光控檯燈。

Arduino 微控制器板子有很多不同的種類，以下的 Arduino UNO 只是其中一種。

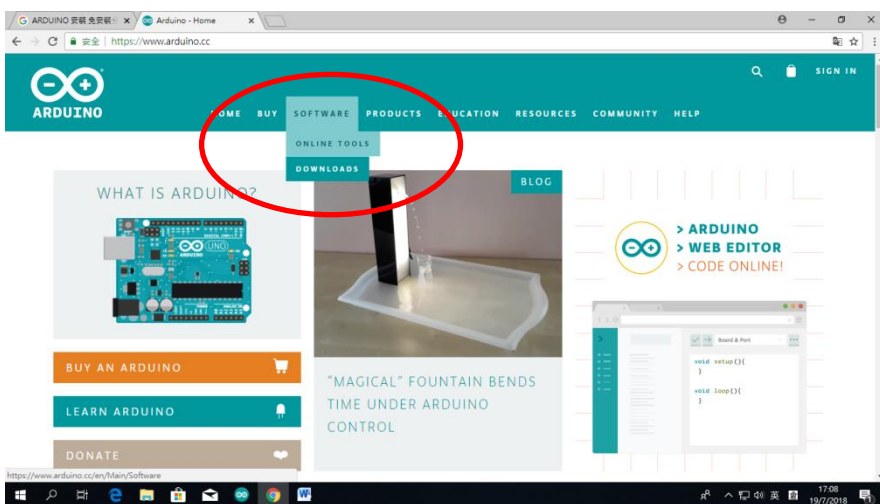


在學習如何編程前，我們需要先下載編寫程式的軟件。

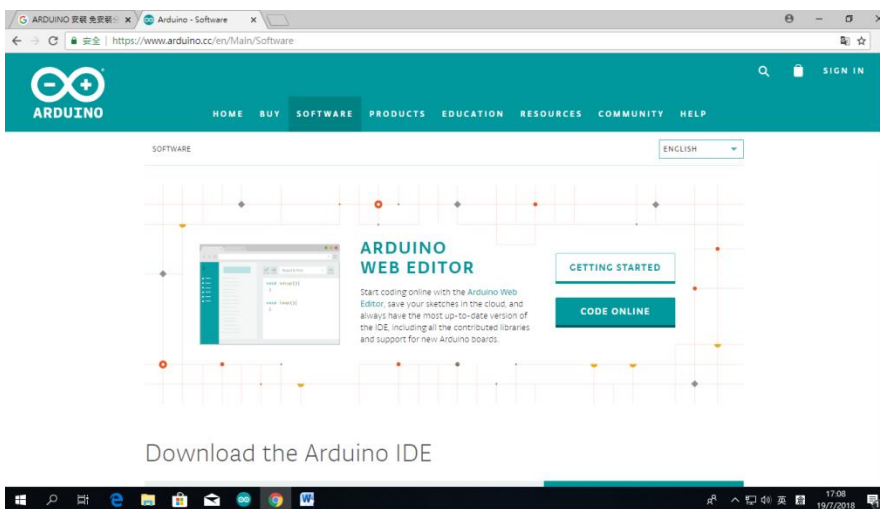
## 1.1 瀏覽 Arduino 官方網頁 <https://www.arduino.cc/>



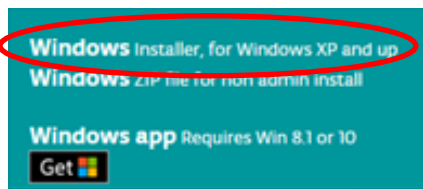
## 1.2 按“SOFTWARE”，然後按“DOWNLOAD”進入新的頁面。



## 1.3 當進入新頁面後，利用滾動軸把頁面往下拉。

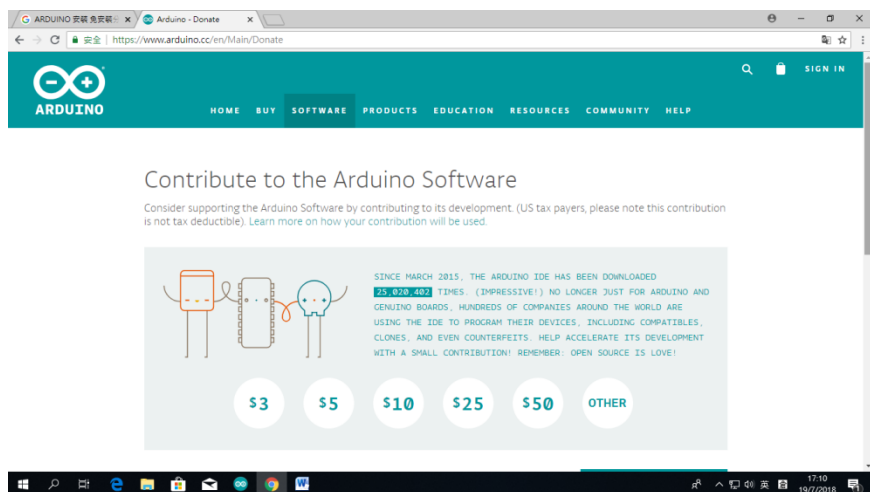


#### 1.4 拉至圓圈中的位置，按 “Windows Installer, for Windows XP and up”



#### 1.5 按 “Windows Installer, for Windows XP and up” 後，我們便會進入新的頁面。

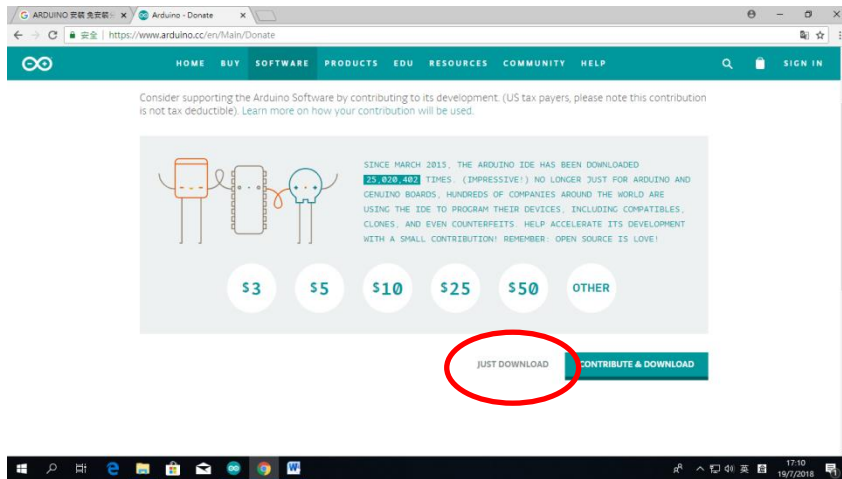
同樣地，利用滾動軸把頁面往下拉。



1.6 直至出現 “JUST DOWNLOAD” (只下載)和 “CONTRIBUTE & DOWNLOAD”

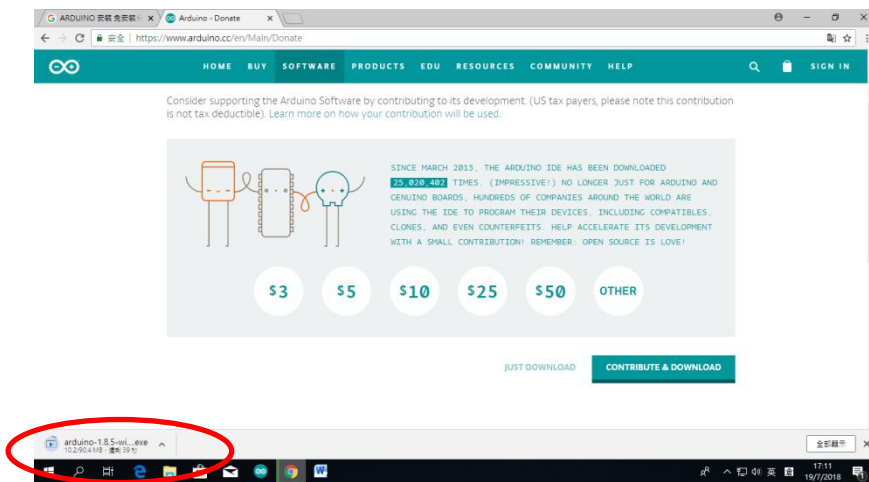
(捐獻和下載)方塊為止，如圖中圓圈所示。

按 “JUST DOWNLOAD” 以下載安裝軟件。

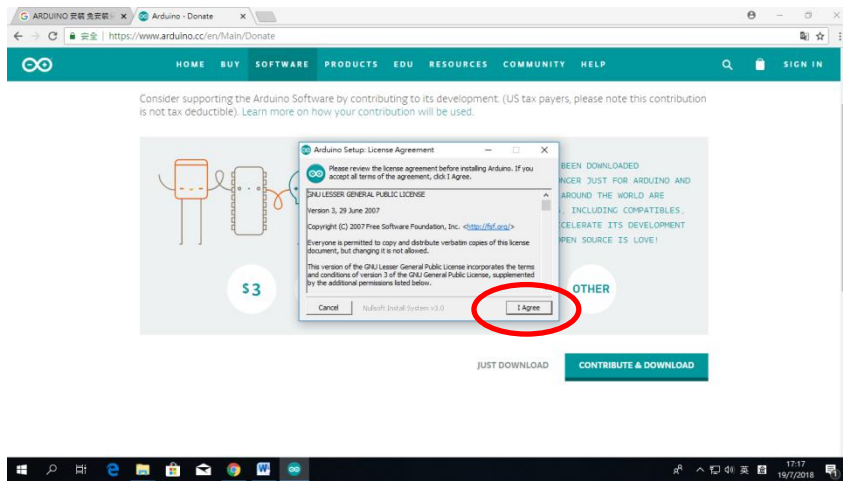


1.7 按 “JUST DOWNLOAD” 後，我們可以在網頁的左下方看見正在下載的安裝軟件。

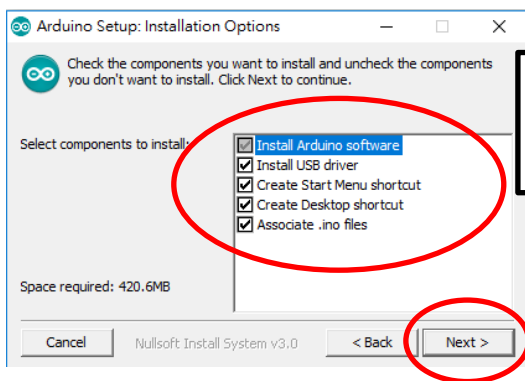
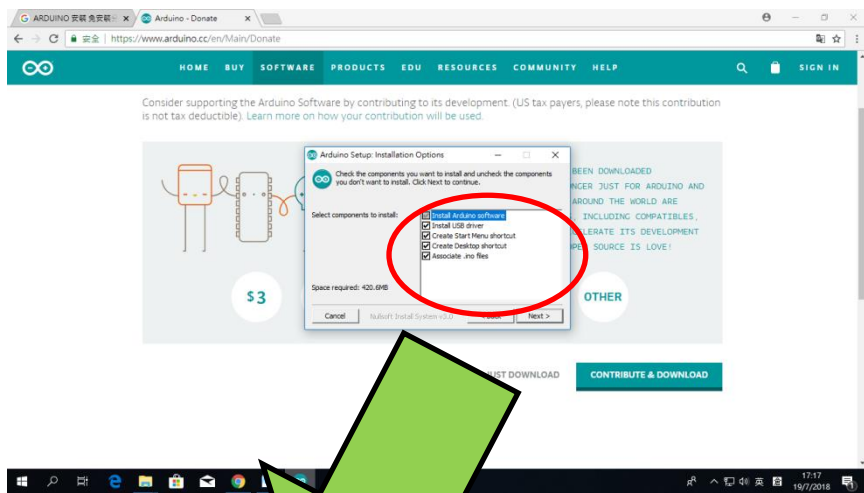
下載完成後，按一下該位置進行安裝。



1.8 當安裝介面彈出後，按 “I Agree” (我同意) 進行安裝。



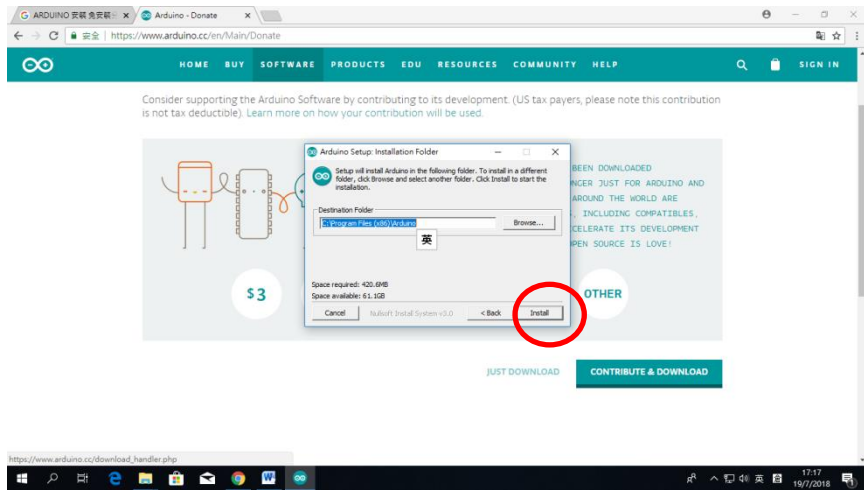
1.9 勾選所有可供選擇的選項，然後按 “NEXT” (下一步)。



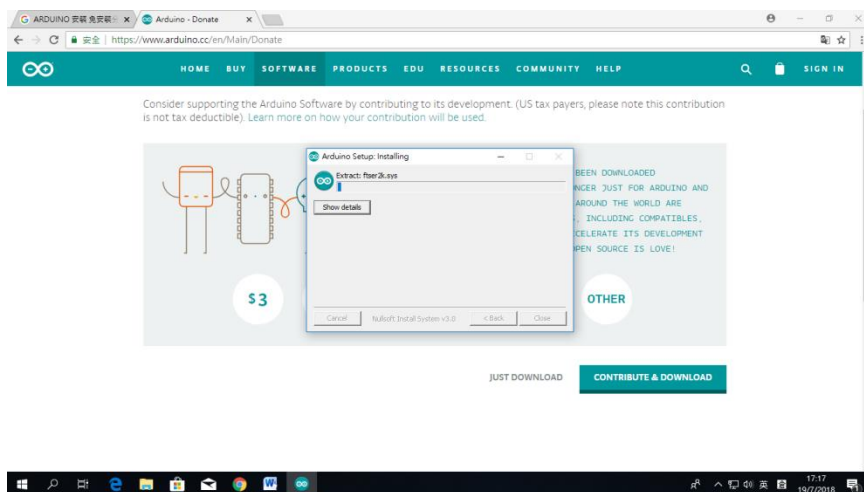
1. 勾選所有可供選擇的選項

2. 按 NEXT

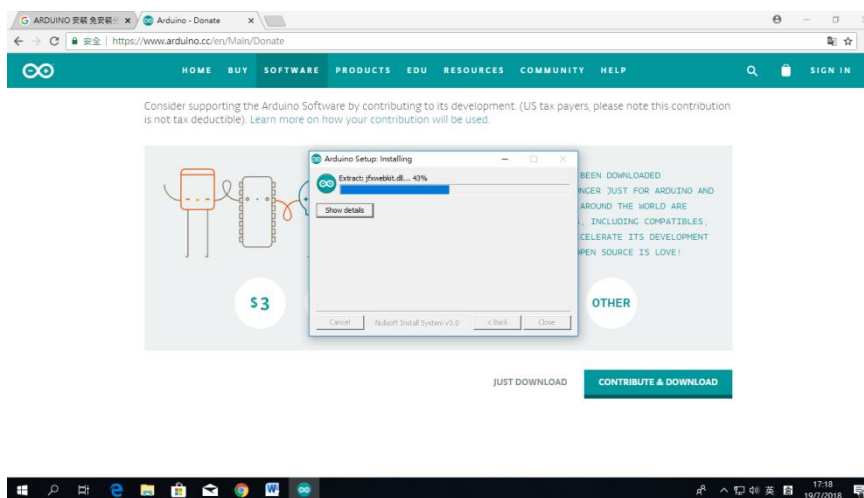
## 1.10 按 “Install” (安裝)



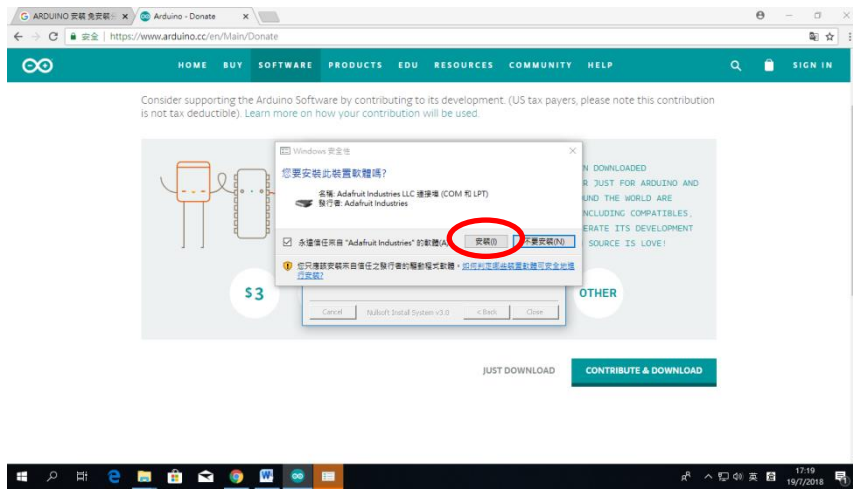
## 1.11 開始安裝軟件。



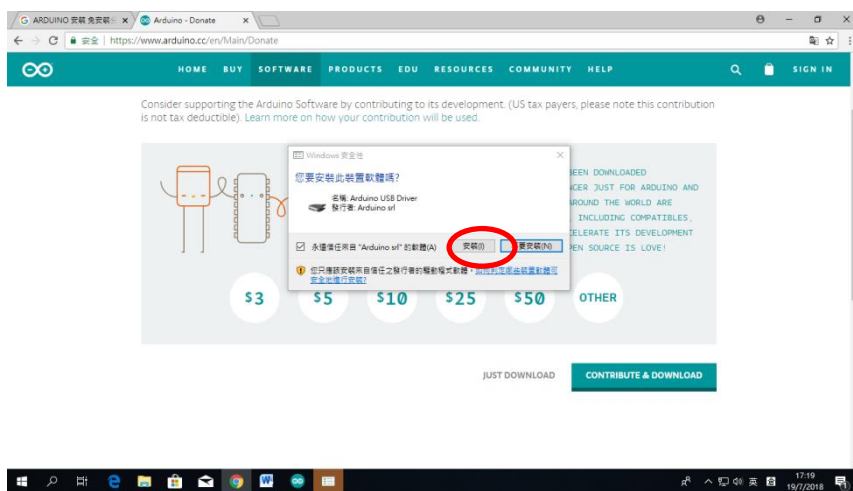
## 1.12 軟件正在安裝中，藍色棒顯示了安裝的進度。



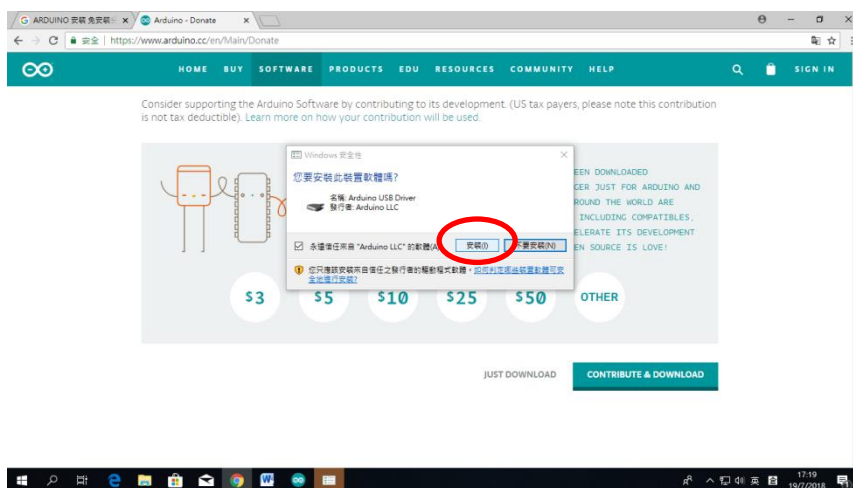
1.13 在安裝的過程中，安裝程式會彈出不同的視窗詢問使用者是否安裝該軟件，我們只需要按“安裝(I)”便可。



1.14 和上一個步驟一樣按“安裝(I)”。

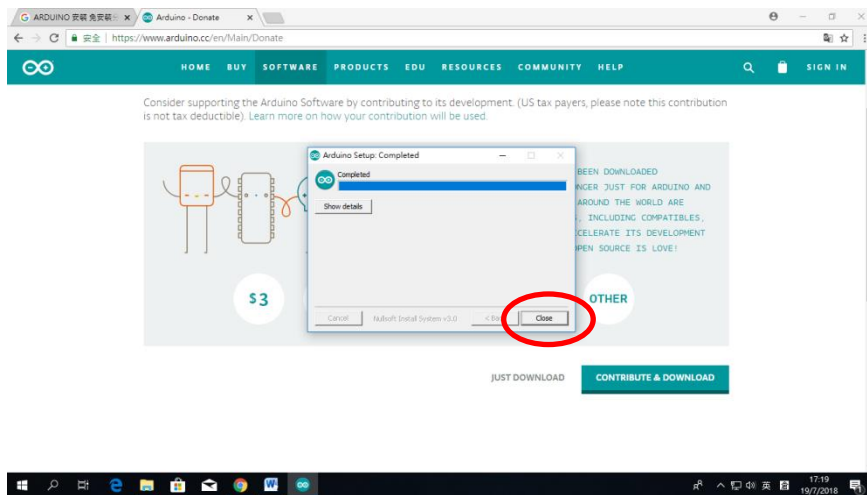



1.15 按“安裝(I)”





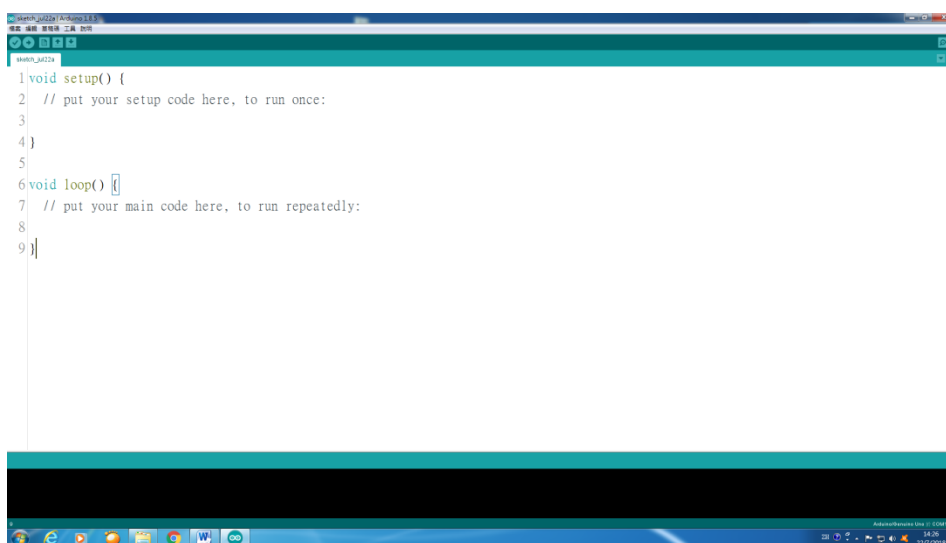
1.16 藍色棒顯示安裝完成，按“Close”以完成安裝程序。



1.17 安裝完成後，桌面出現 Arduino 軟件的圖示 ，連按兩下以開啟 Arduino 軟件。




1.18 開啟 Arduino 後的介面:

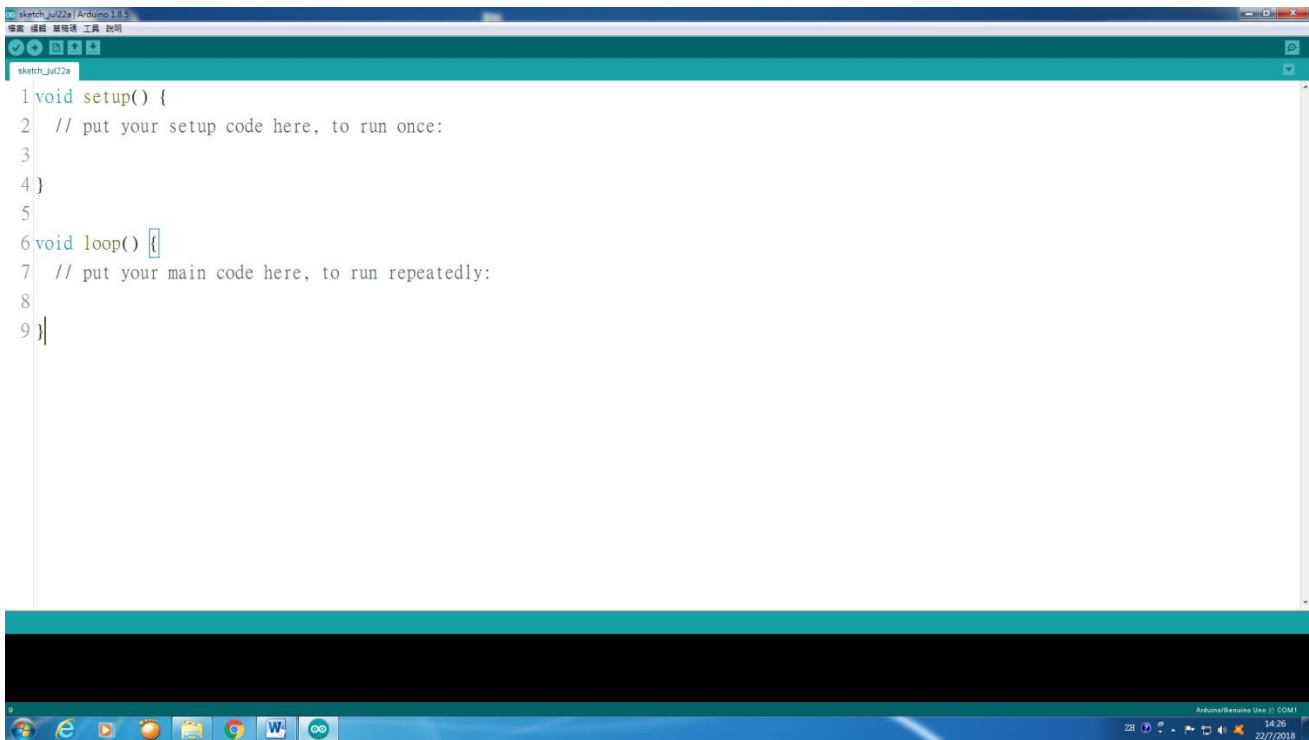


## 2. Arduino 使用介紹

現在先讓我們了解一下 Arduino 的基本操作，例如：新增、儲存和開啟等。

### 2.1 開啟 Arduino

在桌面上連接兩下 Arduino 軟件的圖示 。開啟軟件後的介面如下：



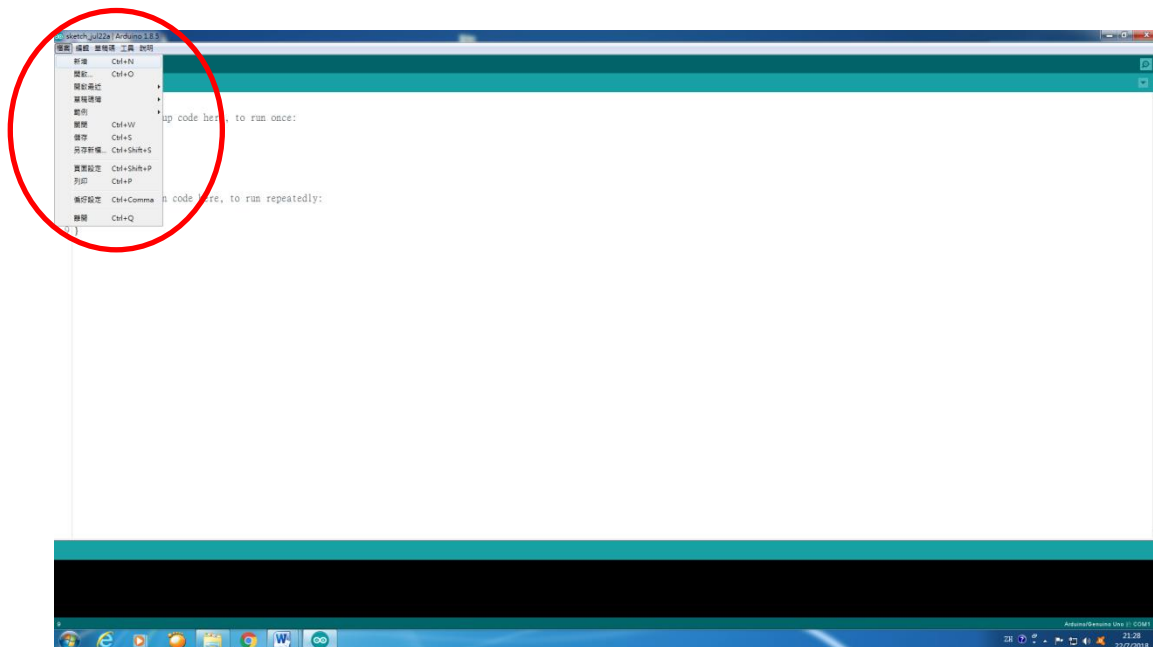
```
sketch Jul 22, 2018 | Arduino 1.8.5
檔案 編輯 匯編譯 工具 說明
sketch Jul 22, 2018
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino@Gensho Uko || COM1  
14:26  
22/7/2018

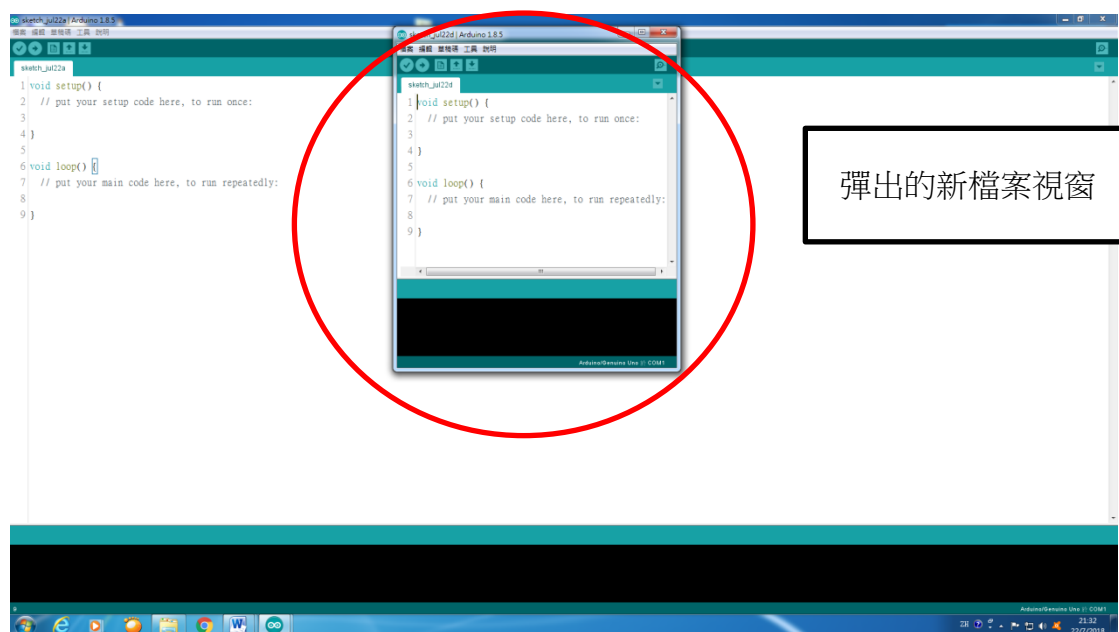
## 2.2 新增

新增是指在現有的工作視窗開啟全新的檔案。我們可以用以下三種方法來開啟全新的檔案。

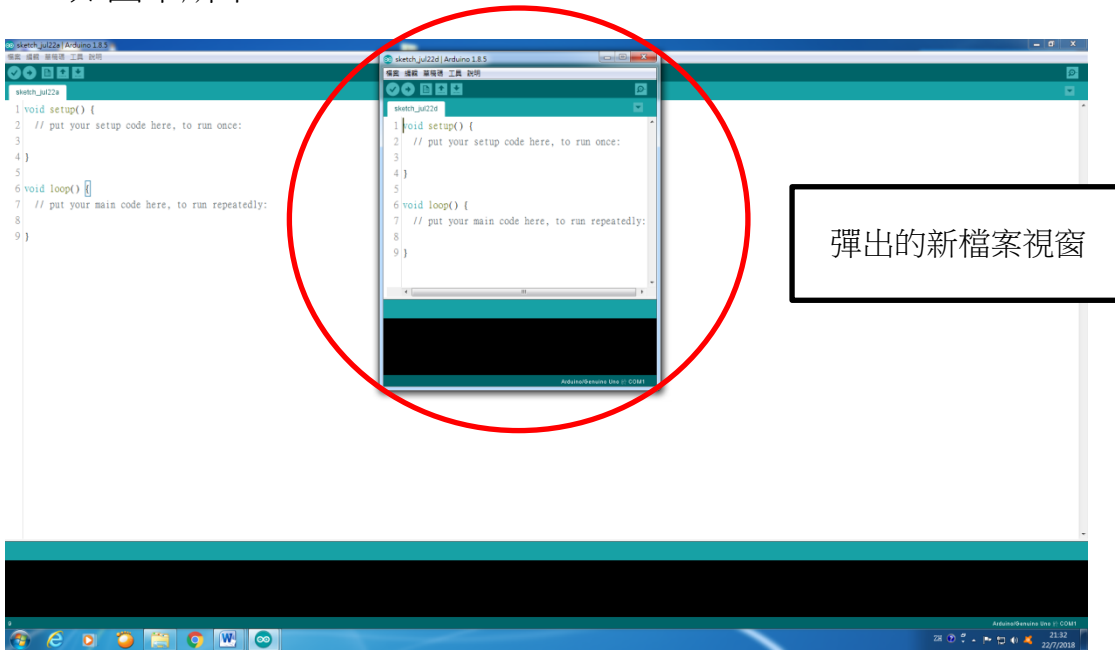
- 方法 1：按“檔案”，然後按“新增”。



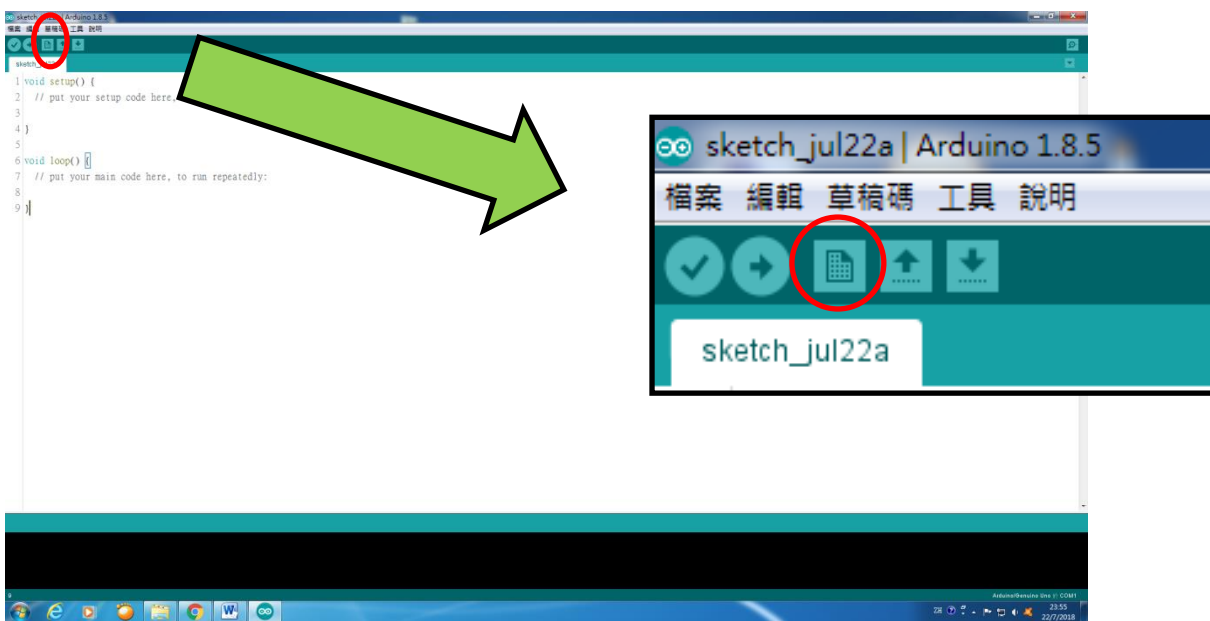
按“新增”後，新檔案的視窗將會彈出，如圖中所示。



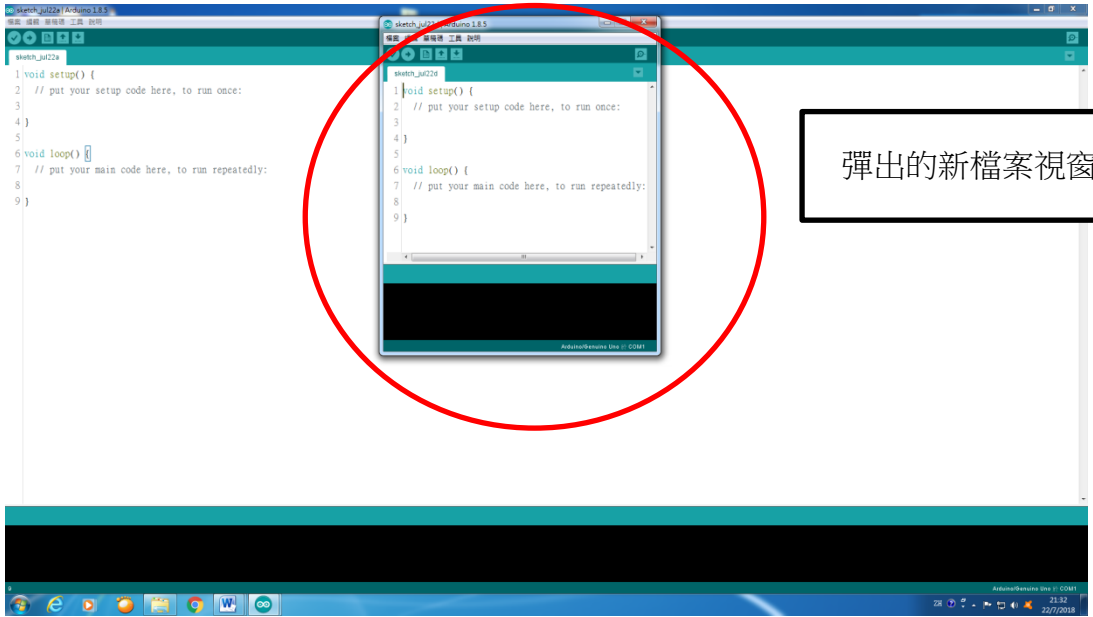
- 方法 2：使用快捷鍵，按“Ctrl + N”（即同時按 Ctrl 和 N 鍵）便會出現新的檔案，如圖中所示。



- 方法 3：使用“新增”快捷按鈕 。按圓圈中的快捷按鈕，新增新的檔案。




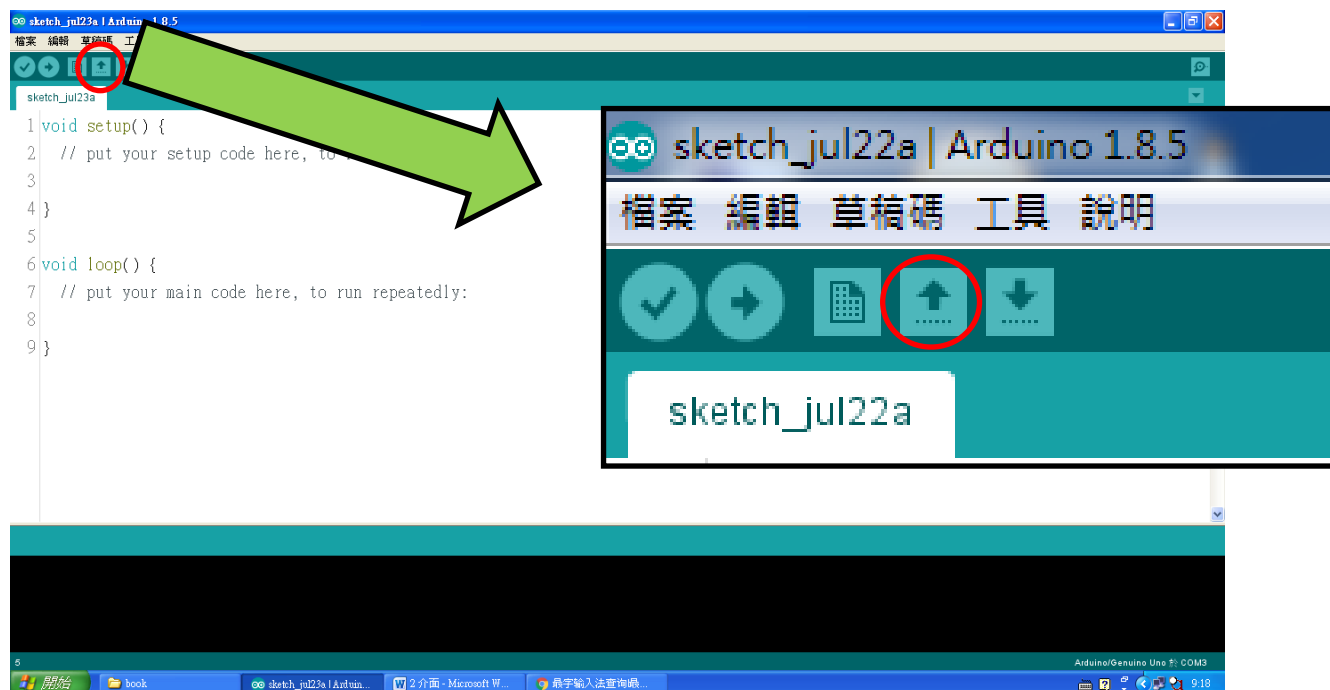
按“新增”快捷按鈕後，新檔案的視窗便會彈出。




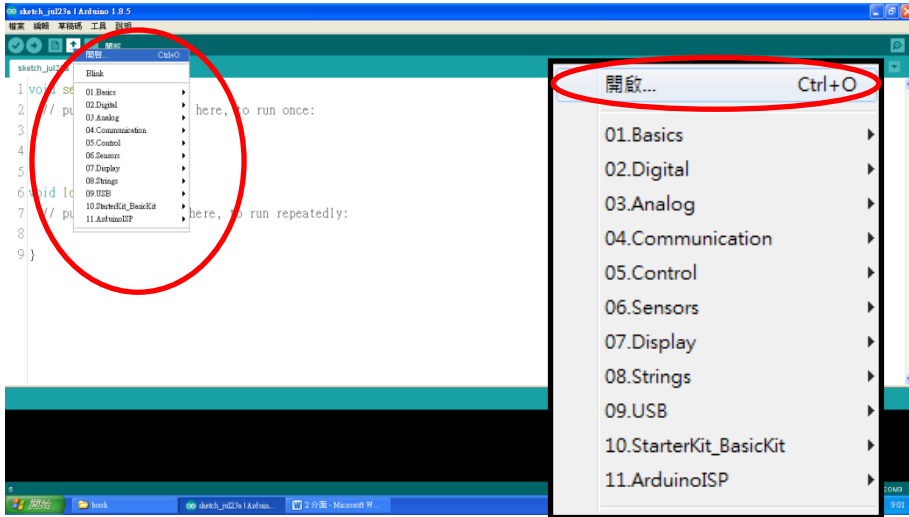
## 2.3 開啟

開啟是指在現有的工作視窗開啟已儲存的檔案。和新增檔案一樣，我們有數種方法來開啟已儲存的檔案。但是，我們只介紹其中較快捷和常用的方法。

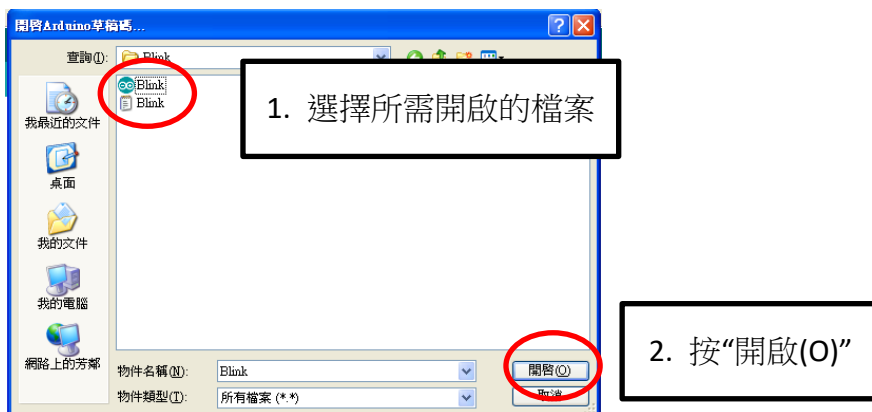
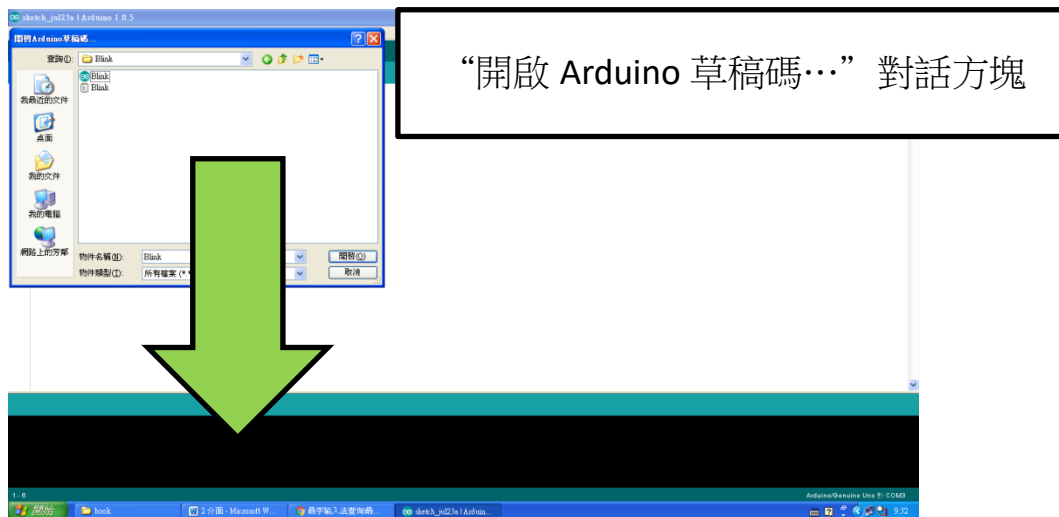
- 使用“開啟”快捷按鈕 。按圓圈中的快捷按鈕，開啟已儲存的檔案。



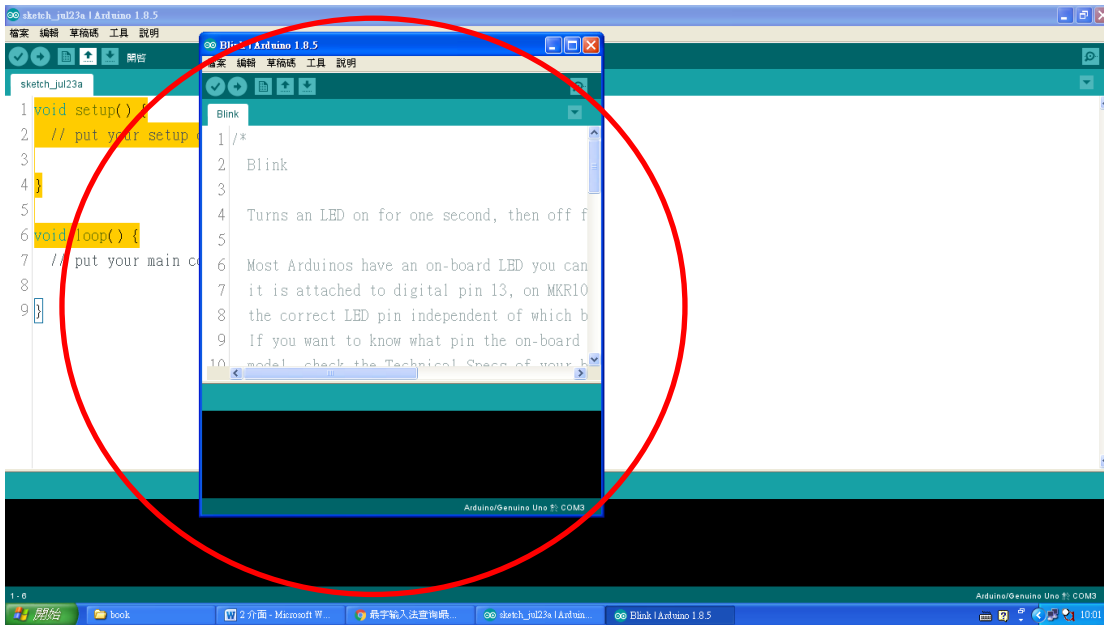
按“開啟”快捷按鈕後，將會出現一個選單，然後按“開啟”。



按“開啟”後，會出現一個對話方塊：“開啟 Arduino 草稿碼...”。選擇所需開啟的檔案，然後按“開啟(O)”。




按“開啟(O)”後，已儲存檔案的視窗便會彈出，我們便可以繼續編輯或使用該檔案。

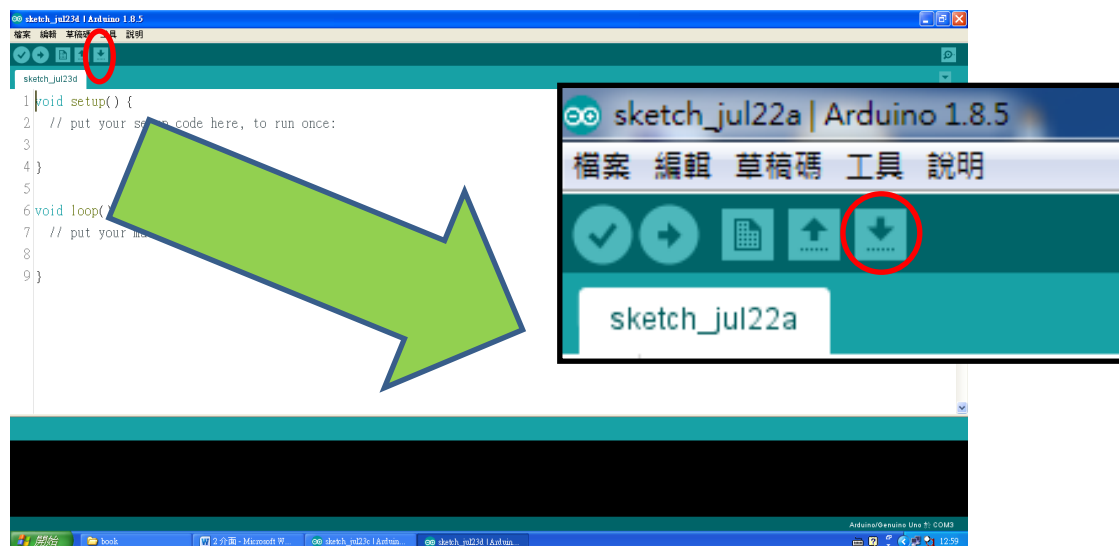




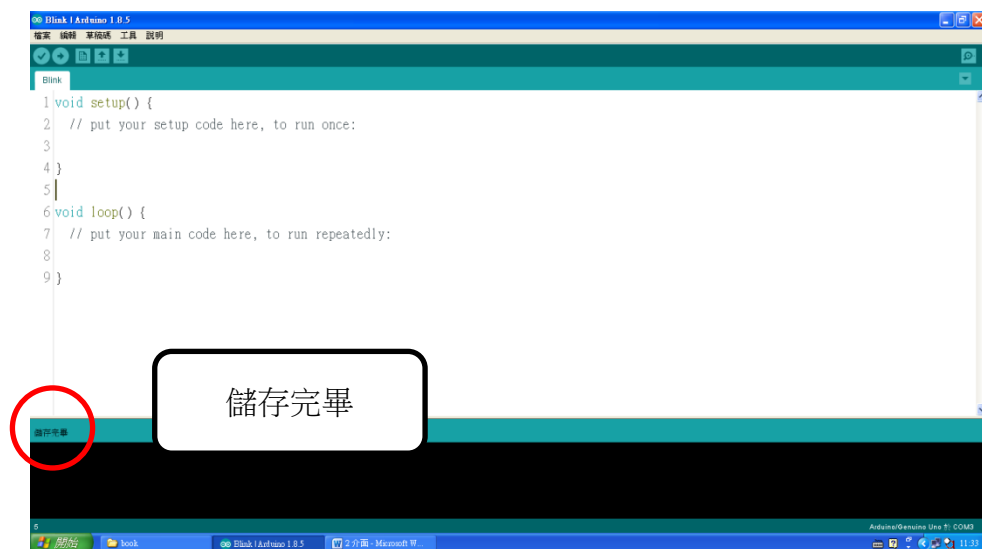
## 2.4 儲存

儲存是指把正在編輯的檔案儲存，以方便下次使用。以下，我們介紹其中一種儲存檔案的方法。

- 按“儲存”快捷按鈕。



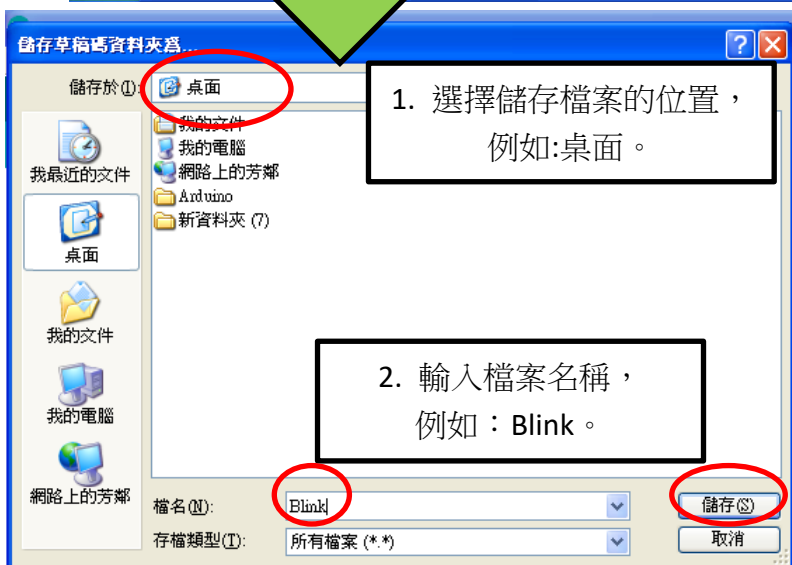
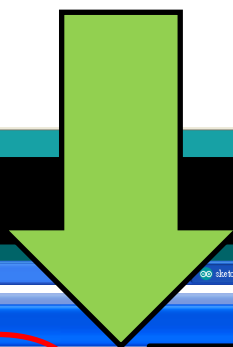
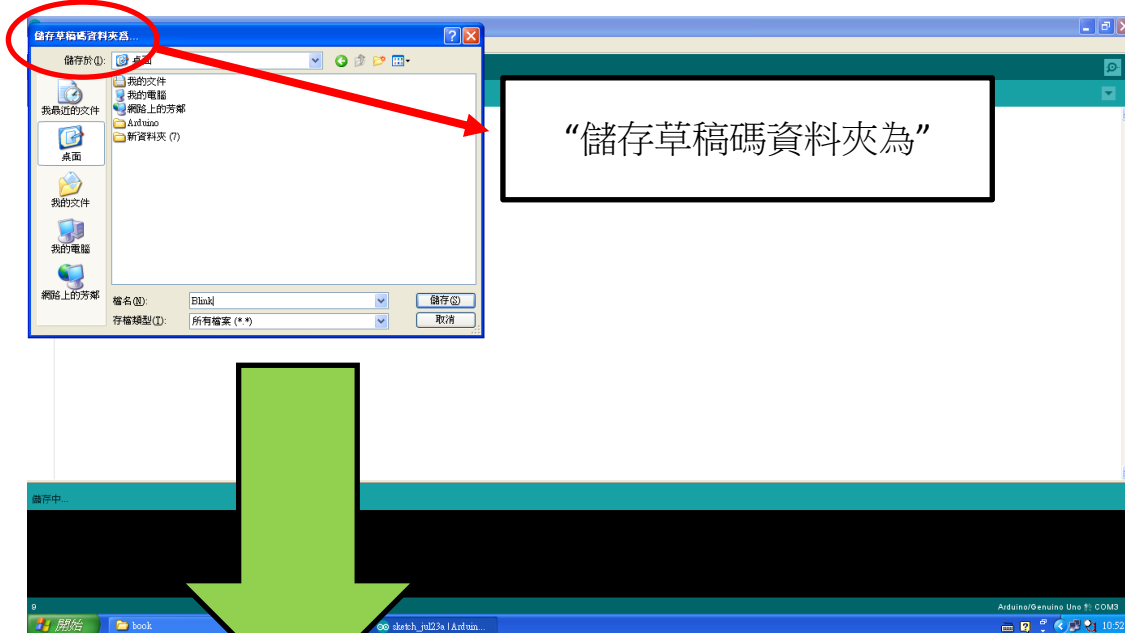
成功儲存後，在軟件介面的左下方會顯示“儲存完畢”。



- 如果儲存的檔案是新的檔案即從沒有儲存過的檔案，當我們按下“儲存”快捷按鈕



時，便會看見以下視窗。然後選擇儲存的位置，並輸入檔案名稱，再按“儲存(S)”。



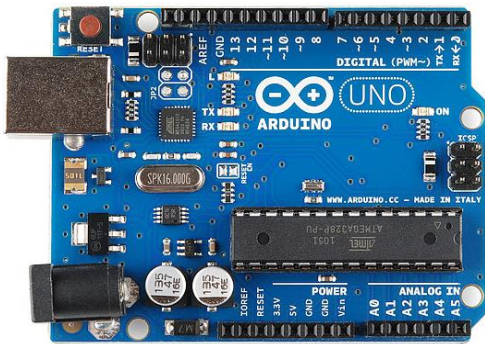
### 3. 初始設定區塊和重複執行區塊的分別

#### 3.1 學習目的:

在這一課節裏，我們會學習如何利用 Arduino 小板子控制 LED 燈每隔一秒閃爍一次，從而學會初始設定區塊和重複執行區塊的分別。

#### 3.2 材料介紹:


- Arduino 主板 x 1 塊



- LED x 1 顆

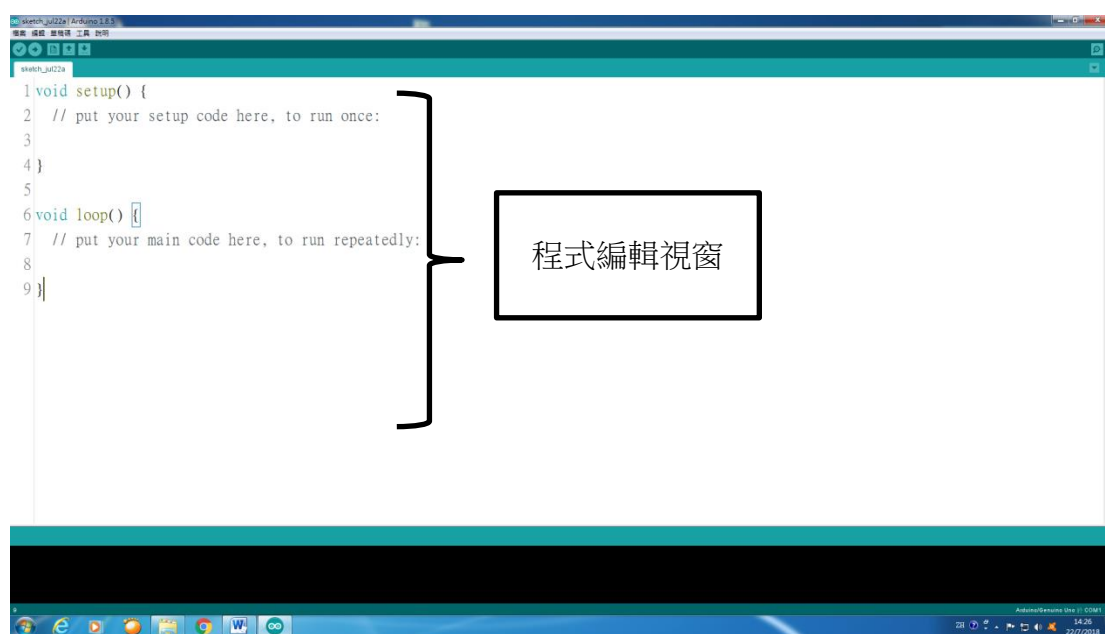


### 3.3 編程步驟

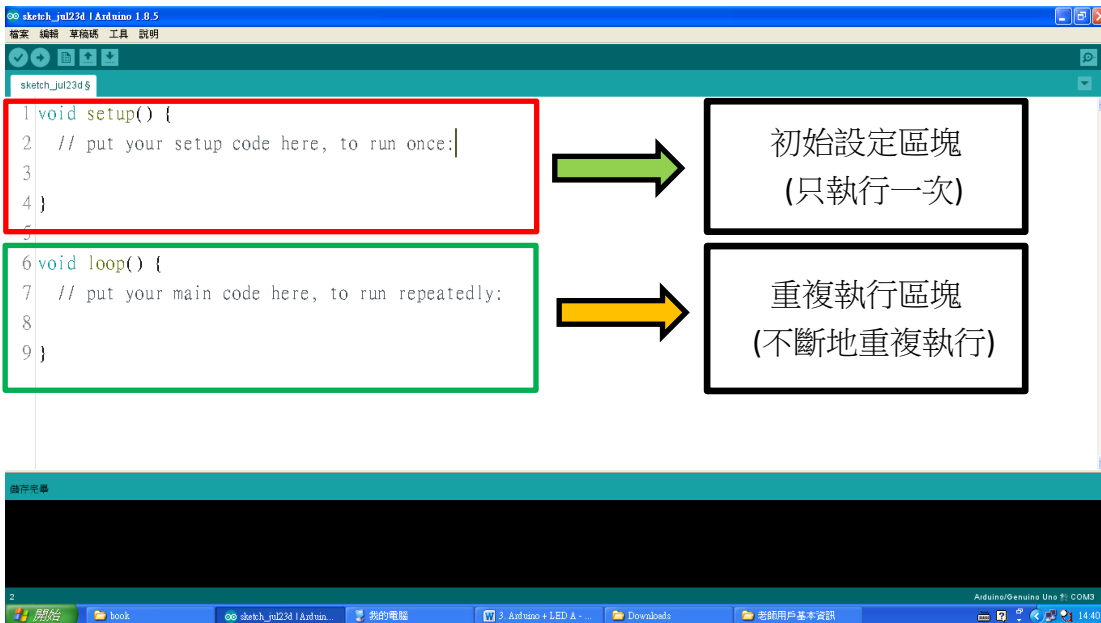
- 按兩下桌面上的 Arduino 軟件圖示 ，以開啟 Arduino 軟件。



- 開啟 Arduino 軟件後的介面如下，白色部分為程式編輯視窗。在該位置，我們可以編寫不同的程式。

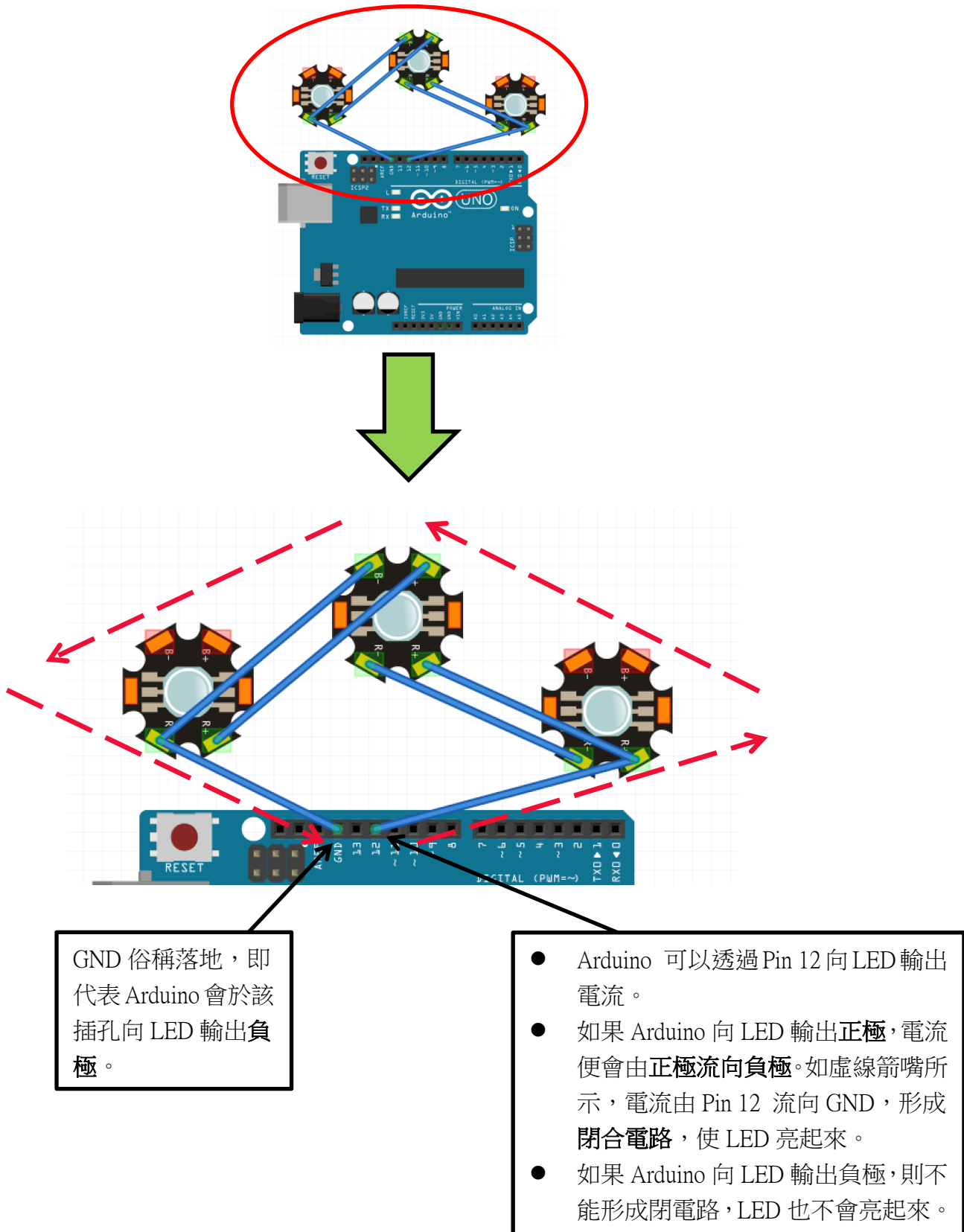


- 在程式編輯視窗裏，我們可以把它分為兩部分。第一部分為初始設定區塊(即紅色方格所示位置)。在這個區塊裏的指令只會被執行一次。第二部分是重複執行區塊(即綠色方格所示位置)。在這個區塊裏的指令會被不斷地重複執行。我們會透過後面的練習中向大家示範兩個區塊的分別。



- 控制 LED 燈閃爍一次

- 電路圖



## - 編寫 Arduino 程式碼

由於我們只需令 LED 燈閃爍一次，我們可以在初始設定區塊裏輸入以下指令。首先，輸入 “pinMode(ledPin, OUTPUT);” 告訴 Arduino ledPin 為輸出模式，在上圖中 ledPin 便是 pin 12。然後輸入 “digitalWrite(ledPin, HIGH);” 令 Arduino 在 ledPin 輸出正極，形成閉合電路，使 LED 亮起來。接著輸入 “ delay(1000);” 以維持亮着 1000 毫秒，即是一秒鐘。最後，輸入 “digitalWrite(ledPin, LOW);” 令 Arduino 輸出負極使 LED 燈熄滅，因它不能形成閉合電路。

換個方式來表達便是這樣:

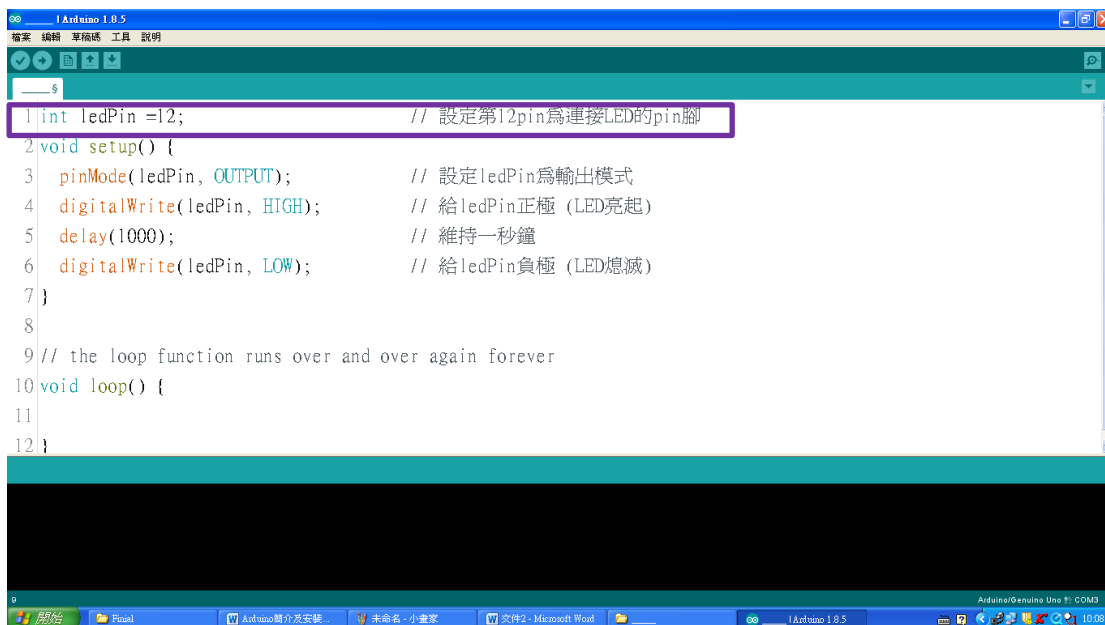
步驟	程式碼
1	pinMode(ledPin, OUTPUT); // 設定 ledPin 為輸出模式
2	digitalWrite(ledPin, HIGH); //給 ledPin 正極 (LED 亮起)
3	delay(1000); //維持一秒鐘
4	digitalWrite(ledPin, LOW); //給 ledPin 負極 (LED 熄滅)

注意:

1. 英文字母的大小寫是有差別的，會令程式碼出現錯誤。
2. ‘//’ 後的文字是老師用來講解程式碼的功能，同學不用輸入。

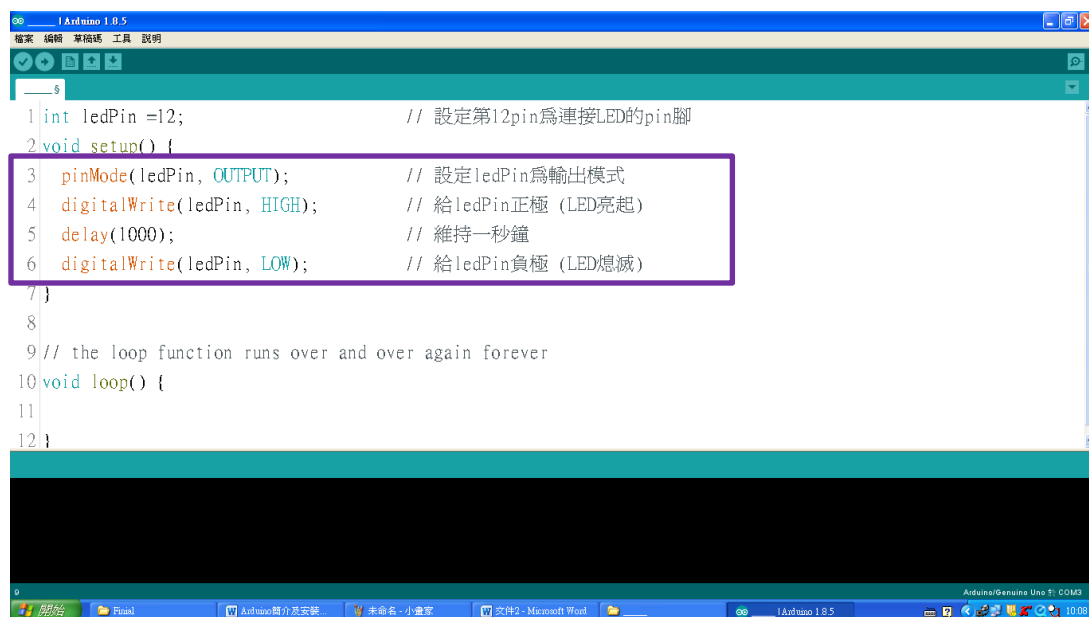
現在我們可以開始如何編寫 Arduino 程式了。首先開啟 Arduino 軟件，在程式編輯視窗

輸入 “int ledPin = 12;” 用以告訴 Arduino pin 12 連接着 LED，如圖中所示。



```
1 int ledPin = 12;           // 設定第12pin為連接LED的pin腳
2 void setup() {
3   pinMode(ledPin, OUTPUT); // 設定ledPin為輸出模式
4   digitalWrite(ledPin, HIGH); // 給ledPin正極 (LED亮起)
5   delay(1000);             // 維持一秒鐘
6   digitalWrite(ledPin, LOW); // 給ledPin負極 (LED熄滅)
7 }
8
9 // the loop function runs over and over again forever
10 void loop() {
11
12 }
```

然後，我們便可以在初始設定區塊裏輸入之前學的 4 個指令，使 LED 閃爍一次，如圖中所示。



```
1 int ledPin = 12;           // 設定第12pin為連接LED的pin腳
2 void setup() {
3   pinMode(ledPin, OUTPUT); // 設定ledPin為輸出模式
4   digitalWrite(ledPin, HIGH); // 給ledPin正極 (LED亮起)
5   delay(1000);             // 維持一秒鐘
6   digitalWrite(ledPin, LOW); // 給ledPin負極 (LED熄滅)
7 }
8
9 // the loop function runs over and over again forever
10 void loop() {
11
12 }
```



放大圖如下:

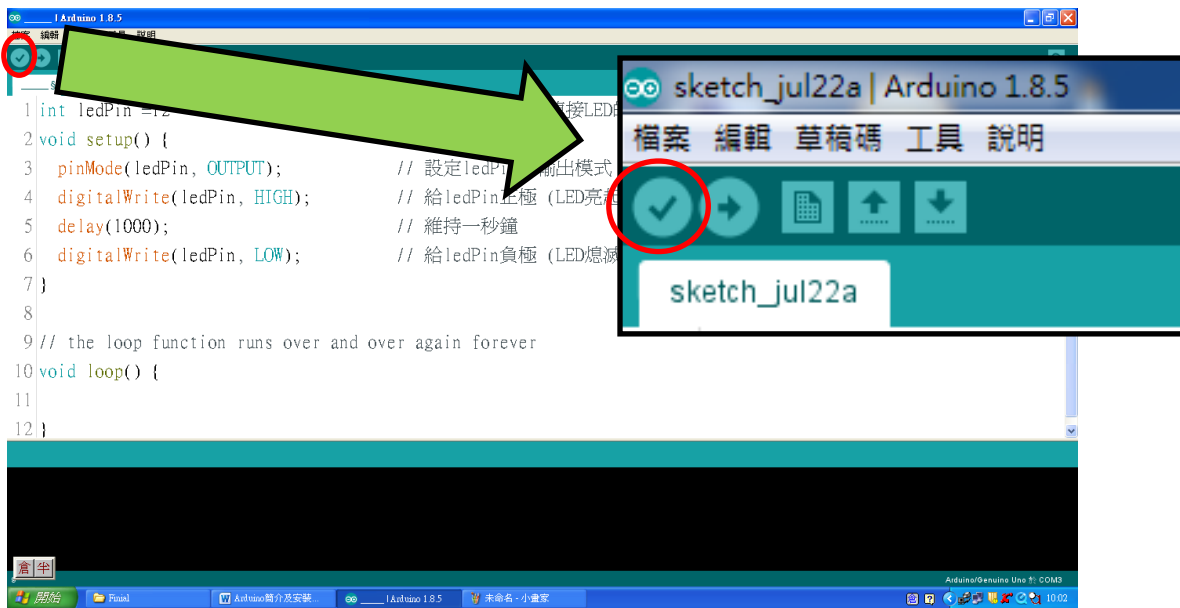
```
1 int ledPin =12; // 設定第12pin為連接LED的pin腳
2 void setup() {
3   pinMode(ledPin, OUTPUT); // 設定ledPin為輸出模式
4   digitalWrite(ledPin, HIGH); // 給ledPin正極 (LED亮起)
5   delay(1000); // 維持一秒鐘
6   digitalWrite(ledPin, LOW); // 給ledPin負極 (LED熄滅)
7 }
8
9 // the loop function runs over and over again forever
10 void loop() {
11
12 }
```

雖然我們在這次編程中並沒有使用這部分，但是它是必需的結構之一，不要刪除。

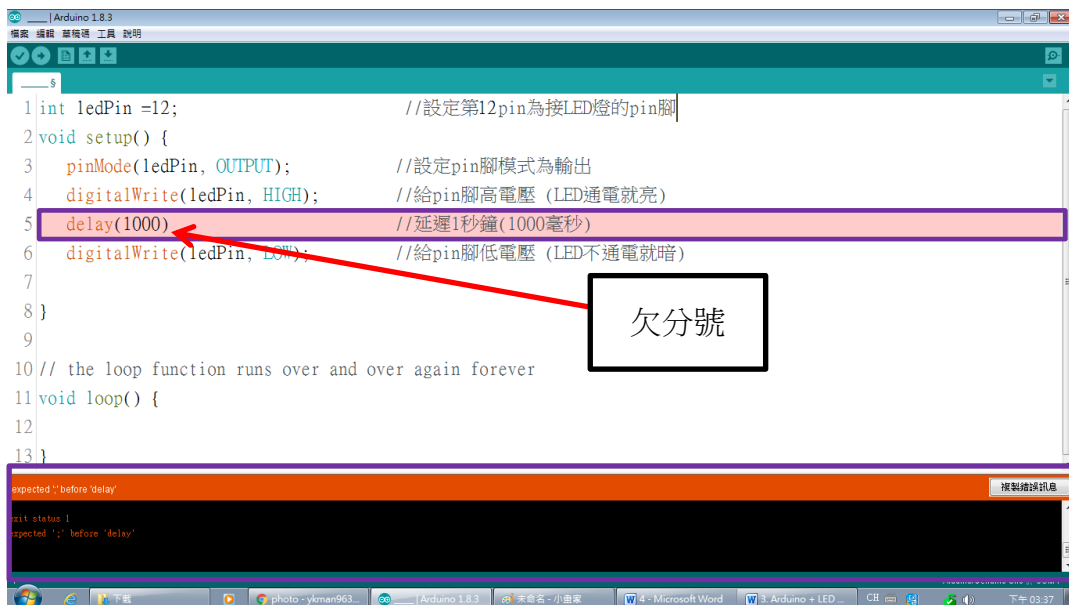
## - 驗證

當我們完成編寫程式後，我們需要進行一次驗證，看看編寫的程式是否正確。驗證的方法如下：

按驗證快捷按鈕 ，進行驗證。



如果在驗證過程中發錯誤的地方，電腦會把錯誤的地方顯示為紅色，並在視窗的下方顯示錯誤訊息。那麼我們便需要作出修改。



如果驗證後沒有發現任何錯誤，在視窗的左下角會顯示“編譯完畢”。

```
1 int ledPin =12;           // 設定第12pin為連接LED的pin腳
2 void setup() {
3   pinMode(ledPin, OUTPUT); // 設定ledPin為輸出模式
4   digitalWrite(ledPin, HIGH); // 給ledPin正極 (LED亮起)
5   delay(1000);           // 維持一秒鐘
6   digitalWrite(ledPin, LOW); // 給ledPin負極 (LED熄滅)
7 }
8
9 // the loop function runs over and over again forever
10 void loop() {
11
12 }
```

編譯完畢

草稿碼使用... 全域變數使用了 9 bytes

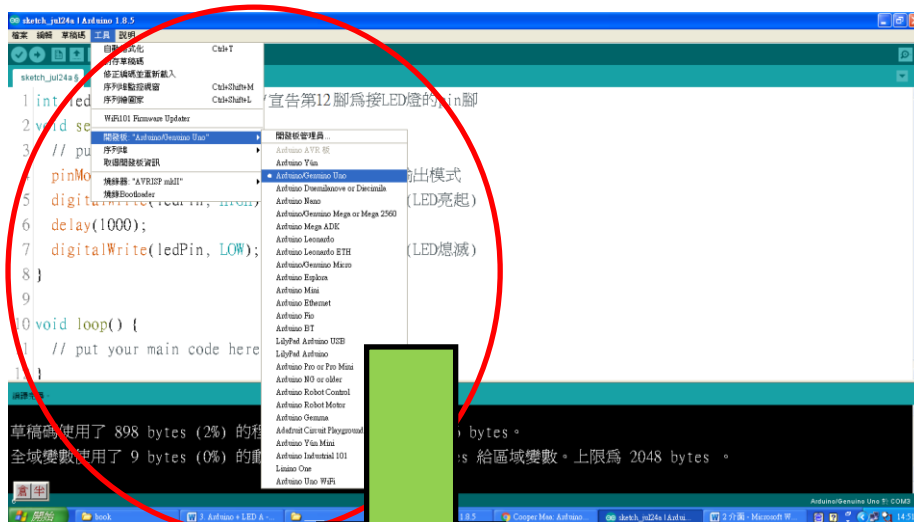
現在，我們便可以準備把程式碼燒錄到 Arduino 底板上。在燒錄先前，我們需要設定開發板和序列埠以確保 Arduino 底板連接在電腦上。

## - 設定開發板

由於 Arduino 底板有很多不同的種類，我們需要告訴電腦我們正在使用的是哪一種。

首先按“工具”，然後在彈出的選單中選

開發板：“Arduino/Genuino Uno”。接著，選擇“Arduino/Genuino Uno”，如圖中所示。



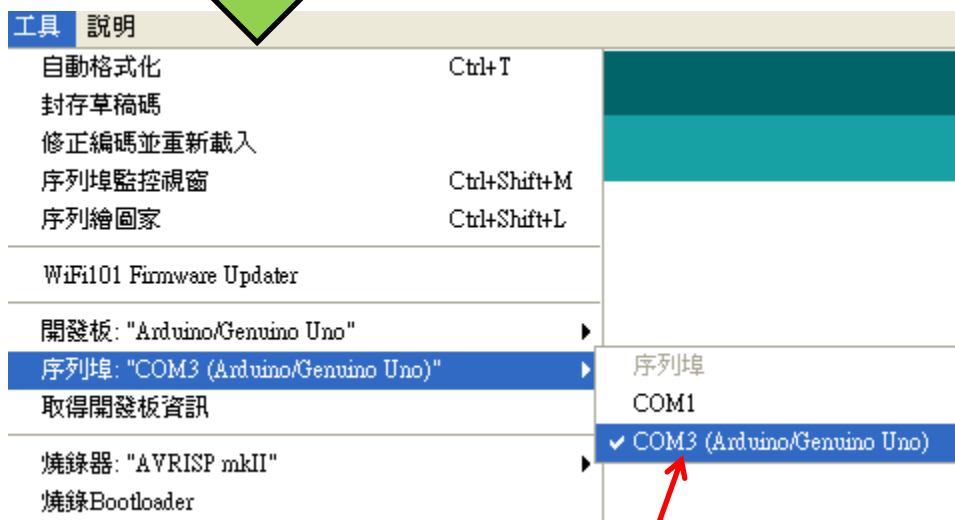
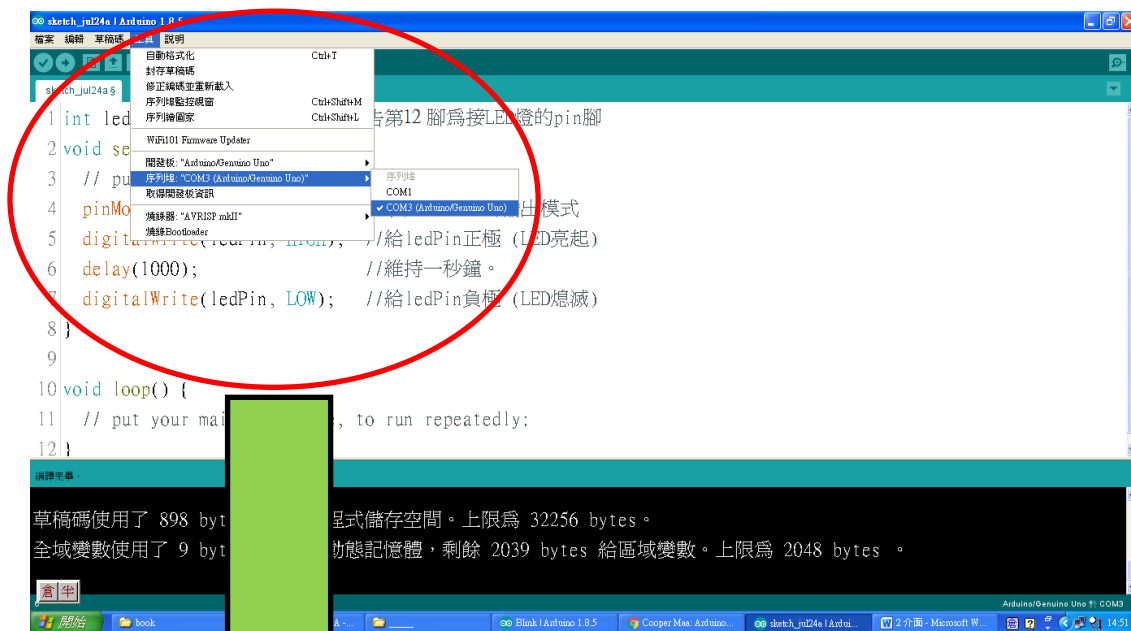
工具	說明
自動格式化	Ctrl+T
封存草稿碼	
修正編碼並重新載入	
序列埠監控視窗	Ctrl+Shift+M
序列埠圖表	Ctrl+Shift+L
WiFi101 Firmware Updater	
開發板: "Arduino/Genuino Uno"	開發板管理員...
序列埠	Arduino AVR 板
取得開發板資訊	Arduino Yún
燒錄器: "AVRISP mkII"	• Arduino/Genuino Uno
燒錄Bootloader	Arduino Duemilanove or Diecimila
	Arduino Nano
	Arduino/Genuino Mega or Mega 2560
	Arduino Mega ADK
	Arduino Leonardo
	Arduino Leonardo ETH
	Arduino/Genuino Micro
	Arduino Esplora
	Arduino Mini
	Arduino Ethernet
	Arduino Fio
	Arduino BT
	LilyPad Arduino USB
	LilyPad Arduino
	Arduino Pro or Pro Mini
	Arduino NG or older
	Arduino Robot Control
	Arduino Robot Motor
	Arduino Gemma
	Adafruit Circuit Playground
	Arduino Yún Mini
	Arduino Industrial 101
	Limino One
	Arduino Uno WiFi

## - 設定序列埠

首先按“工具”，然後在彈出的選單中選

序列埠: “COM3 (Arduino/Genuino Uno)”。接著，選擇 COM3 ( Arduino/Genuino Uno) ，

如圖中所示。

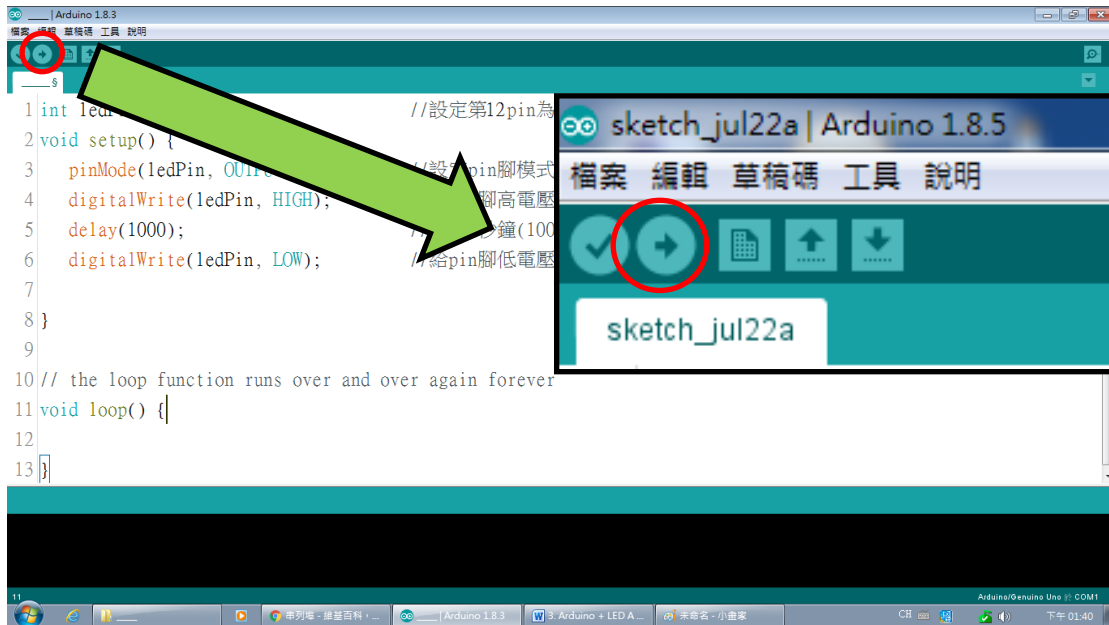


並不是每部電腦都出現 COM3，可以是 COM4，COM5..... 最重要的是必須出現 (Aduino/Genuino Uno)這三個字

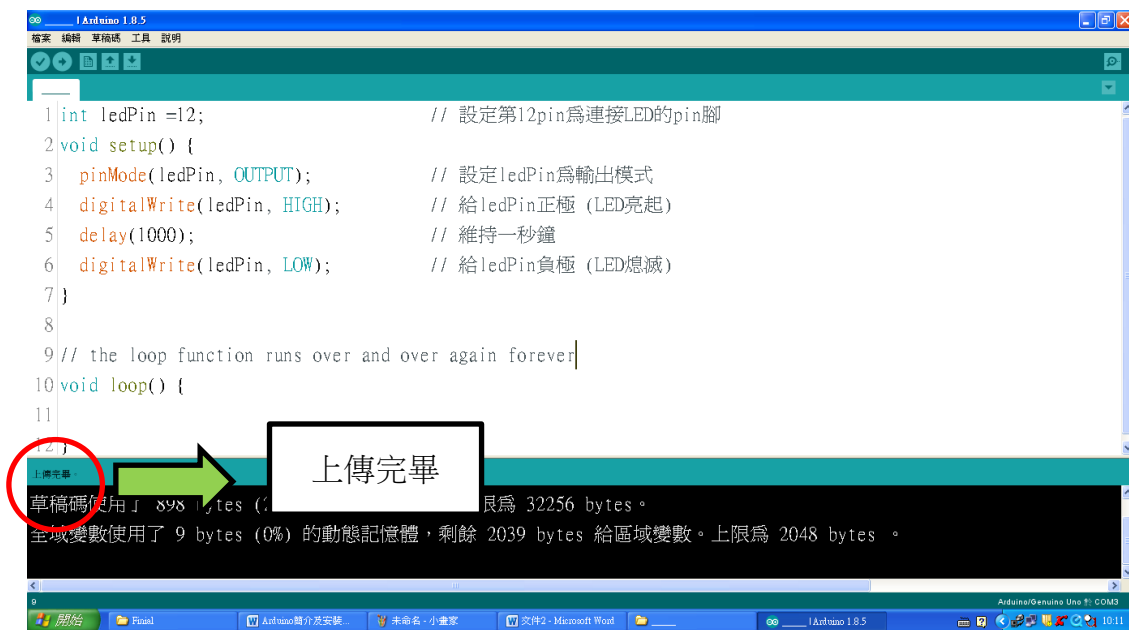
## - 上傳

完成以上步驟後，我們便可以利用上傳功能把編寫好的程式燒錄到 Arduino 底板上。

按上傳快捷按鈕，進行燒錄。



燒錄完畢後，會於左下角顯示“上傳完畢”。

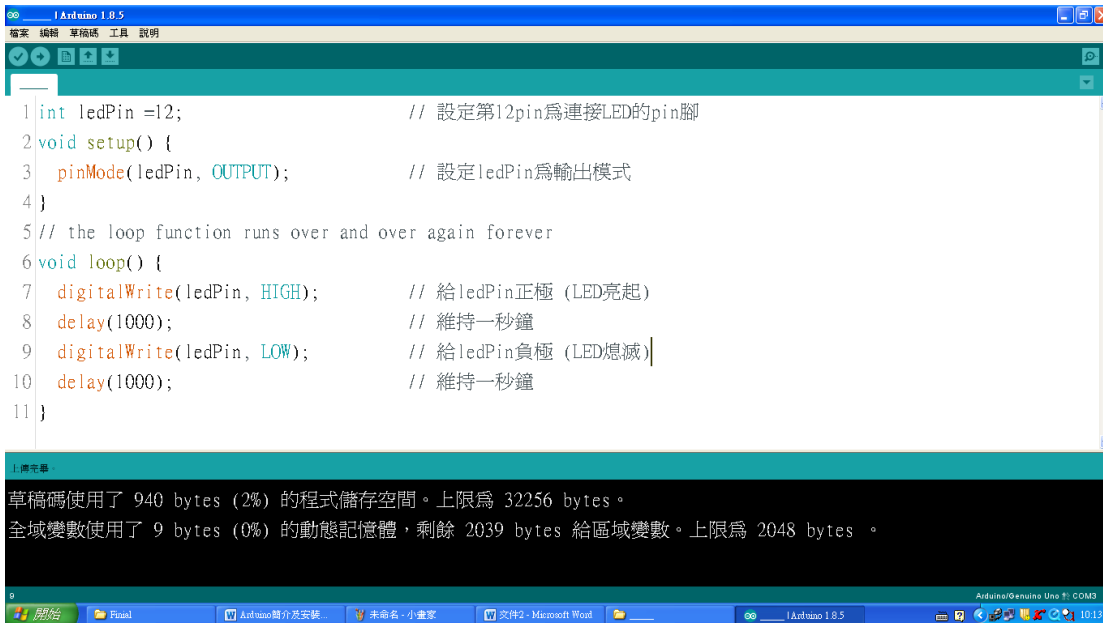


完成後，我們會發現 LED 只閃爍一次，便不會再次閃爍。

- 控制 LED 燈每隔一秒閃爍一次

如果想要 LED 在 Arduino 開機後不停地閃爍，我們可以怎樣修改之前的程式碼呢？

首先我們在初始設定區塊中輸入 “pinMode(ledPin, OUTPUT);” 。然後，在重複執行區塊中輸入 “digitalWrite(ledPin, HIGH);” “ delay(1000);” 和 “ digitalWrite(ledPin, LOW);” 最後，再加上 “delay(1000);” 讓 LED 保持熄滅一秒鐘。



```
1 int ledPin =12;           // 設定第12pin為連接LED的pin腳
2 void setup() {
3   pinMode(ledPin, OUTPUT); // 設定ledPin為輸出模式
4 }
5 // the loop function runs over and over again forever
6 void loop() {
7   digitalWrite(ledPin, HIGH); // 給ledPin正極 (LED亮起)
8   delay(1000);                // 維持一秒鐘
9   digitalWrite(ledPin, LOW);  // 給ledPin負極 (LED熄滅)
10  delay(1000);                // 維持一秒鐘
11 }
```

草稿碼使用了 940 bytes (2%) 的程式儲存空間。上限為 32256 bytes。  
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。

最後，我們把程式碼進行驗證、設定開發板、設定序列埠和燒錄到 Arduino 底板上。這樣 LED 燈便能每隔一秒閃爍一次。

## 4 智能光控枱燈

### 4.1 學習目的:

在這一課節裏，我們會學習如何利用光敏電阻制作一個智能光控枱燈，在光線不足時，枱燈會自動亮起來。相反，在光線充足時，枱燈會自動關掉。

### 4.2 材料介紹:

- Arduino 底板 x 1 塊



- 光敏電阻 x 1 顆



- 10K 電阻 x 1 顆



- LED X 1 顆

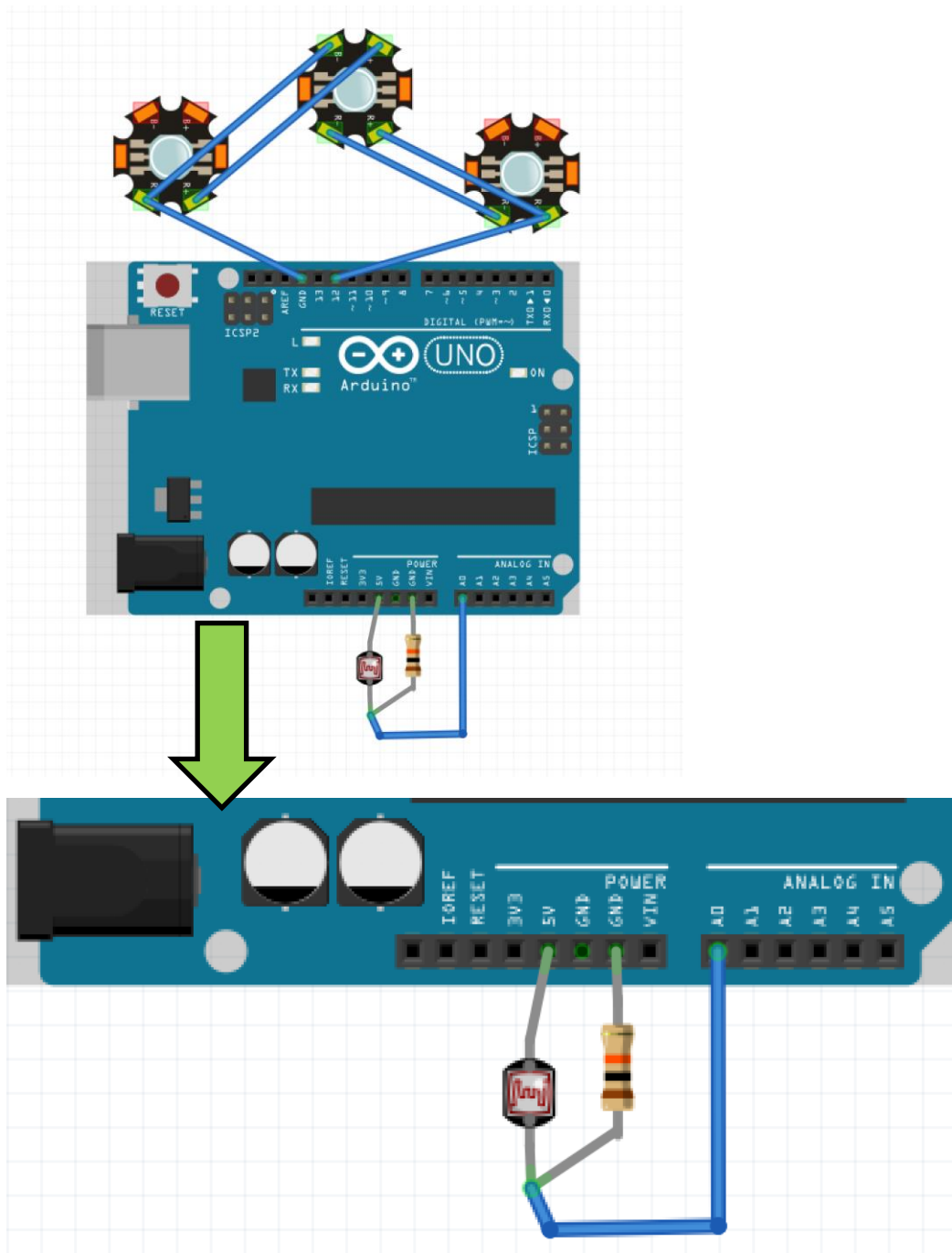




## 4.3 編程步驟

- 電路圖

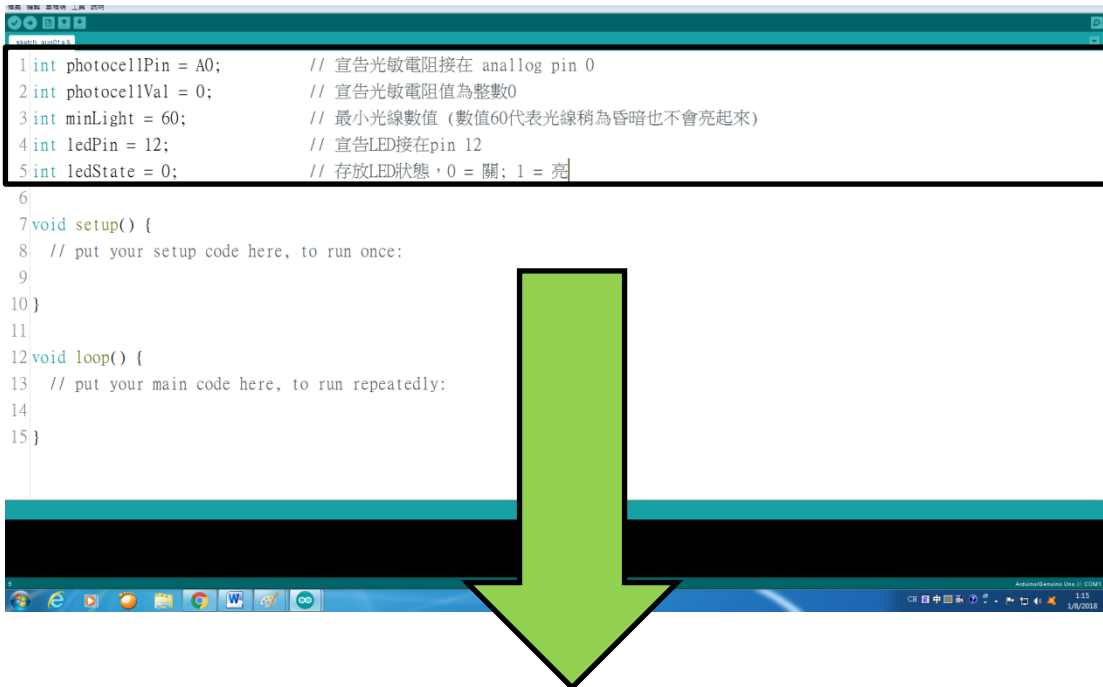
制作智能光控檯燈的方法很簡單，我們只需在上一次的練習中加上一個光敏電阻和 10K 電阻便可以了。連接的方法可參考下面的電路圖：



## ● 編寫 Arduino 程式碼

首先，在初始設定區塊前輸入以下第一至第五行的程式碼，

以告知 Arduino 一些基本資料。



```
1 int photocellPin = A0;           // 宣告光敏電阻接在 analog pin 0
2 int photocellVal = 0;           // 宣告光敏電阻值為整數0
3 int minLight = 60;             // 最小光線數值 (數值60代表光線稍為昏暗也不會亮起來)
4 int ledPin = 12;              // 宣告LED接在pin 12
5 int ledState = 0;              // 存放LED狀態，0 = 關; 1 = 亮
6
7 void setup() {
8   // put your setup code here, to run once:
9
10 }
11
12 void loop() {
13   // put your main code here, to run repeatedly:
14
15 }
```

```
1 int photocellPin = A0;           // 宣告光敏電阻接在 analog pin 0
2 int photocellVal = 0;           // 宣告光敏電阻值為整數0
3 int minLight = 60;             // 最小光線數值 (數值60代表光線稍為昏暗也不會亮起來)
4 int ledPin = 12;              // 宣告LED接在pin 12
5 int ledState = 0;              // 存放LED狀態，0 = 關; 1 = 亮
```

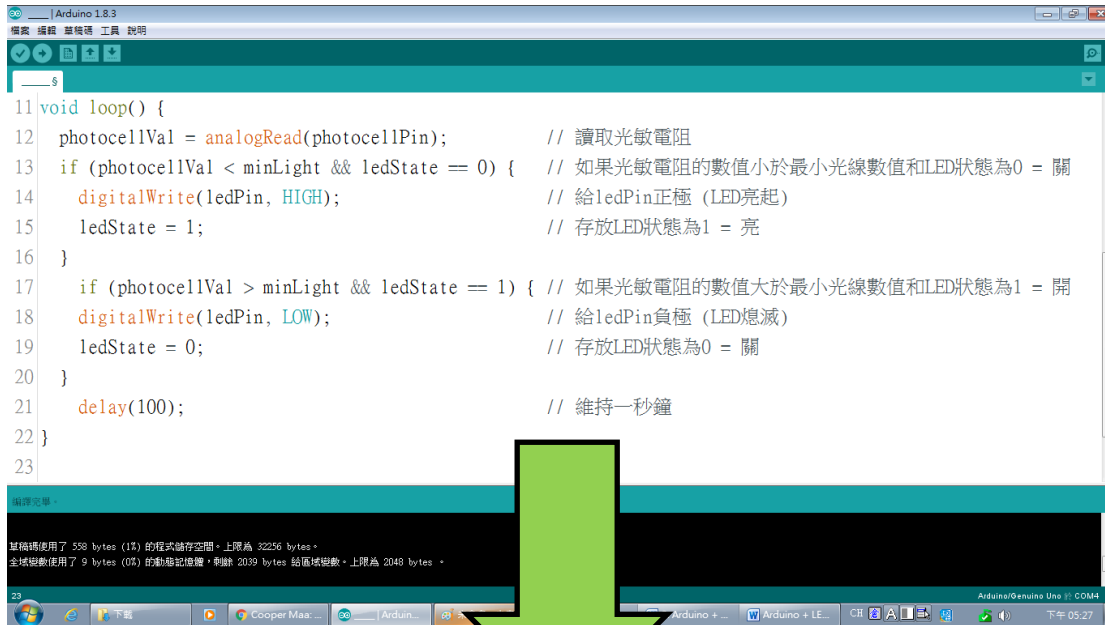
然後在初始設定區塊中輸入以下程式碼：

```
1 int photocellPin = A0;           // 宣告光敏電阻接在 analog pin 0
2 int photocellVal = 0;           // 宣告光敏電阻值為整數0
3 int minLight = 60;              // 最小光線數值 (數值60代表光線稍為昏暗也不會亮起來)
4 int ledPin = 12;                // 宣告LED接在pin 12
5 int ledState = 0;               // 存放LED狀態, 0 = 關; 1 = 亮
6
7 void setup() {
8   pinMode(ledPin, OUTPUT);      // 設定pin腳模式為輸出
9 }
10
11
12 void loop() {
13   // put your main code here, to run repeatedly:
14
15 }
```



```
7 void setup() {
8   pinMode(ledPin, OUTPUT);      // 設定pin腳模式為輸出
9 }
```

接著，在重複執行區塊輸入以下程式碼：



```
11 void loop() {
12   photocellVal = analogRead(photocellPin);           // 讀取光敏電阻
13   if (photocellVal < minLight && ledState == 0) {    // 如果光敏電阻的數值小於最小光線數值和LED狀態為0 = 關
14     digitalWrite(ledPin, HIGH);                    // 給ledPin正極 (LED亮起)
15     ledState = 1;                                   // 存放LED狀態為1 = 亮
16   }
17   if (photocellVal > minLight && ledState == 1) {    // 如果光敏電阻的數值大於最小光線數值和LED狀態為1 = 開
18     digitalWrite(ledPin, LOW);                     // 給ledPin負極 (LED熄滅)
19     ledState = 0;                                   // 存放LED狀態為0 = 關
20   }
21   delay(100);                                       // 維持一秒鐘
22 }
23
```

```
11 void loop() {
12   photocellVal = analogRead(photocellPin);           // 讀取光敏電阻
13   if (photocellVal < minLight && ledState == 0) {    // 如果光敏電阻的數值小於最小光線數值和LED狀態為0 = 關
14     digitalWrite(ledPin, HIGH);                    // 給ledPin正極 (LED亮起)
15     ledState = 1;                                   // 存放LED狀態為1 = 亮
16   }
17   if (photocellVal > minLight && ledState == 1) {    // 如果光敏電阻的數值大於最小光線數值和LED狀態為1 = 開
18     digitalWrite(ledPin, LOW);                     // 給ledPin負極 (LED熄滅)
19     ledState = 0;                                   // 存放LED狀態為0 = 關
20   }
21   delay(100);                                       // 維持一秒鐘
22 }
```

最後，我們把程式碼進行驗證、設定開發板、設定序列埠和燒錄到 Arduino 底板上。

#### 4.4 思考題

如果你想把智能檯燈修改到在完全漆黑的環境下才亮起來，

你可以怎樣修改你的程式碼？