

第2單元

開發環境的認識與操作

► 軟體下載與安裝

所有傳統的單晶片，都要自備類似記事本的編輯器，編寫程式、存檔、離開，然後使用其所提供的編譯程式，若有錯誤則要繼續開啓記事本修改，再編譯，直到完全正確。然後還要購買燒錄器，將程式燒到單晶片。但是 Arduino 就神了，竟然有免費整合編輯視窗，整合以上編輯、編譯、燒錄（上傳）於單一視窗，此稱為整合編輯程式（IDE, Integrated Development Environment 的縮寫）。請於 Arduino 官網點選『SOFTWARE/DOWNLOADS』，畫面如下，請點選『Windows installer for windows xp and up』，即可下載最新執行檔（*.exe）。接著，請開啓檔案總管，至下載區按兩下該執行檔，即可安裝。



The screenshot shows the Arduino IDE 1.8.13 download page. The page has a dark teal header with navigation links: HARDWARE, SOFTWARE (selected), DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. The main content area is white with a teal sidebar on the right. The main content includes the Arduino logo, the title 'Arduino IDE 1.8.13', a description of the IDE, and a link to the 'Getting Started' page. The sidebar contains 'DOWNLOAD OPTIONS' for Windows, Linux, and Mac OS X, with a 'Get' button for the Windows app. At the bottom, there are links for 'SOURCE CODE' and 'Release Notes'.

arduino.cc/en/software

HARDWARE SOFTWARE DOCUMENTATION COMMUNITY BLOG ABOUT

 Arduino IDE 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 [Get](#)

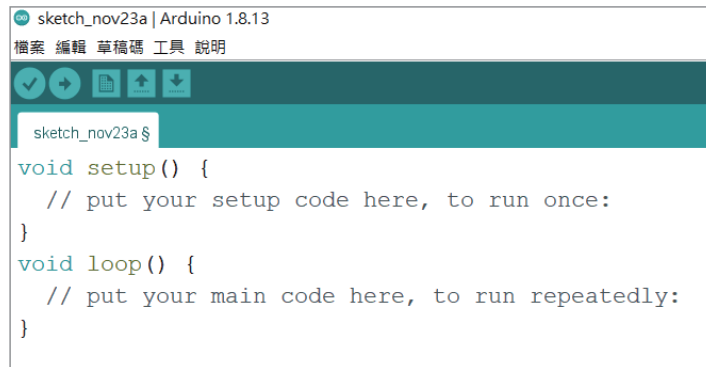
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

Release Notes
Checksums (sha512)

► IDE畫面

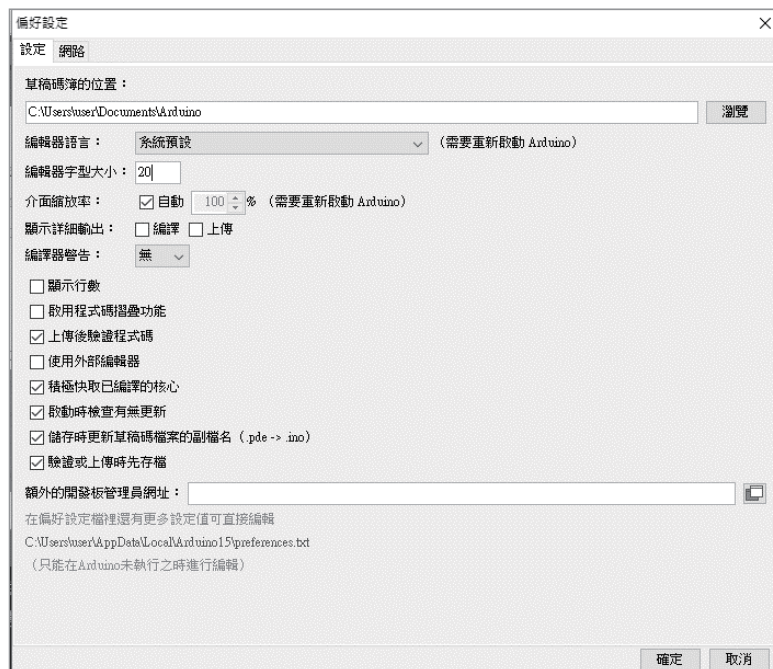
下圖是開啓 Arduino IDE 畫面。



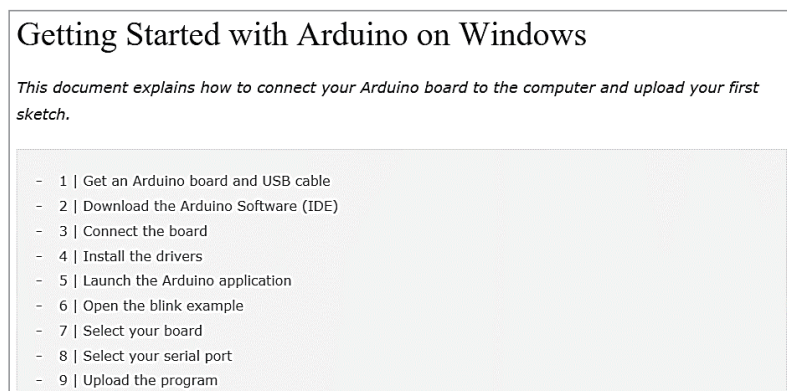
```
sketch_nov23a | Arduino 1.8.13
檔案 編輯 草稿碼 工具 說明
sketch_nov23a §
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

► 功能表的語言設定

理論上安裝後，功能表的語言會依據作業系統的語言自動完成設定，但若不是您所要的語言，請於功能表點選『檔案 / 偏好設定』，如下圖，即可點選您要的語言。



下圖是點選功能表的『說明 / 入門手冊』，畫面如下圖：



► 單晶微控板使用步驟

Arduino 的單晶微控板使用步驟分別是插入微控板、點選開發板型號、點選通訊埠編號、取得開發版資訊，分別說明如下：

1. 插入微控板

請依照指示，將微控板 USB 插頭插入電腦 USB 插座。請留意微控板右上角電源指示燈是否亮起。

2. 點選開發板型號

請點選功能表的『工具 / 開發板資訊』即可點選您所使用的微控板型號。(請依照您的微控板型號點選，筆者是點選『Arduino/Genuino Mega or Mega 2560』)

3. 點選通訊埠編號

請點選功能表的『工具 / 序列埠』即可點選您所使用的序列埠編號(備註：系統會出現可用編號 com1 或 com2,3,4,5 等讓您點選)。

4. 取得開發版資訊

請點選功能表的『工具 / 開發版資訊』即可取得您的微控板資訊。若出現下圖，才表示以上設定就緒，才能上傳程式。（實驗的中途，若改插入別人的微控板，那也要重覆以上步驟，直到看到下圖，才表示有抓到此微控板，才能上傳程式。）

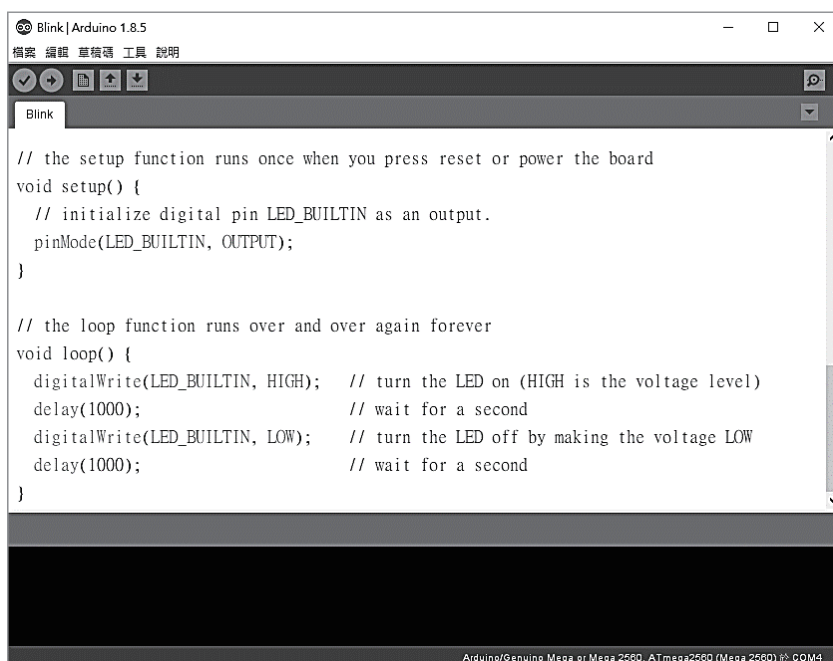


▶ 自我測試

Arduino 有內植一顆 LED，腳位編號是 13，整合開發環境安裝後，內部也含一個自我測試程式，以下說明如何自我測試，自我測試成功，那就表示此塊微控板是好的，且電腦與微控板可以一起工作。進行單晶片控制都要這樣一步一步來，這樣當問題發生時，才能一步一步除錯，逐漸縮小錯誤範圍。

1. 開啓自我測試程式

下圖是開啓測試程式 Blink（請點選功能表的『檔案 / 開啓』，預設路徑是 C:/ProgramFiles(x86)/Arduino/examples/01. Basics/Blink），其功能是直接使用微控板預植的 LED（不管什麼板 UNO、MEGA…等），都是腳位 13，以常數『LED_BUILTIN』表示），並令其明滅閃爍各一秒。

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.8.5". The menu bar includes "檔案", "編輯", "上傳", "工具", and "說明". The toolbar shows icons for "驗證" (Verify) and "上傳" (Upload). The main text area contains the following C++ code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}


// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

The status bar at the bottom indicates "Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) @ COM4".

2. 驗證程式

按一下工具列的『驗證』按鈕 ，即可編譯程式。

3. 上傳程式

按一下工具列的『上傳』按鈕 ，即可上傳程式到微控板（上傳又稱為燒錄）。上傳結束將會自動執行程式，請觀察微控板上所預植 LED 是否明滅（此顆 LED 就在腳位 13 旁）。

補充說明

1. `setup()` 函式僅在程式一開始時執行一次。所以通常放置執行程式的初始設定、或程式執行後只執行一次的指令。
2. `loop()` 函式則會重複不斷的被執行，所以就放置一些需要反覆執行的指令。
3. `delay()` 函式是程式延遲程式。因為指派 LED 亮，總要停留一點時間，使用者才有機會看到。

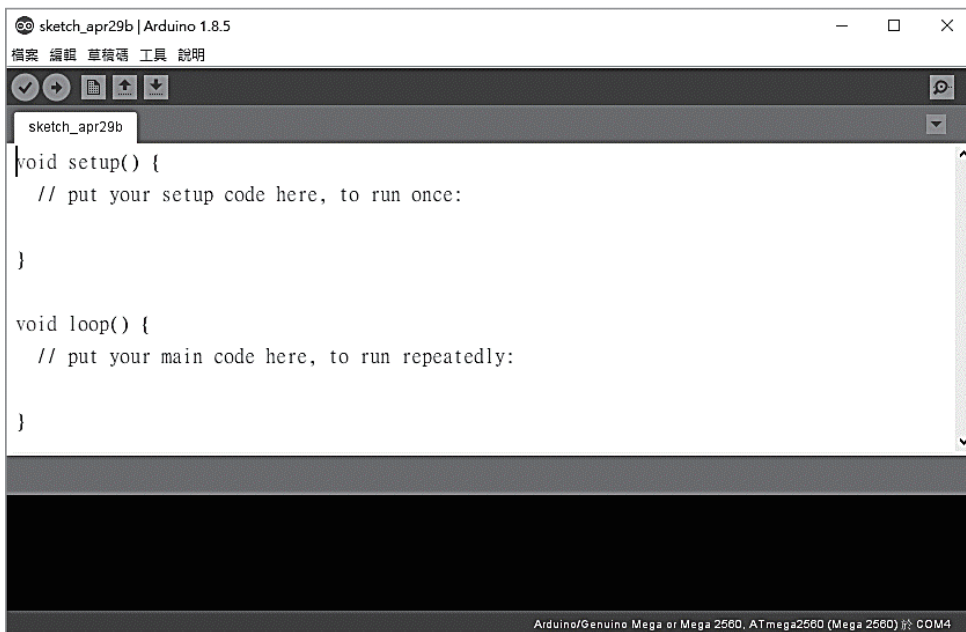
4. 雙斜線『//』稱為註解，僅給人看，電腦不予編譯，沒有鍵入也沒關係。

範例 2a

請設計一程式，讓內建 LED 明亮 1.5 秒，滅掉 0.5 秒。

操作步驟

1. 開新檔案。開新檔案，出現程式樣版如下：(點選功能表的『檔案 / 新增』或工具列的『新增』圖項。



```
sketch_apr29b | Arduino 1.8.5
檔案 編輯 草稿碼 工具 說明
sketch_apr29b
void setup() {
  // put your setup code here, to run once:
}

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

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) 於 COM4
```

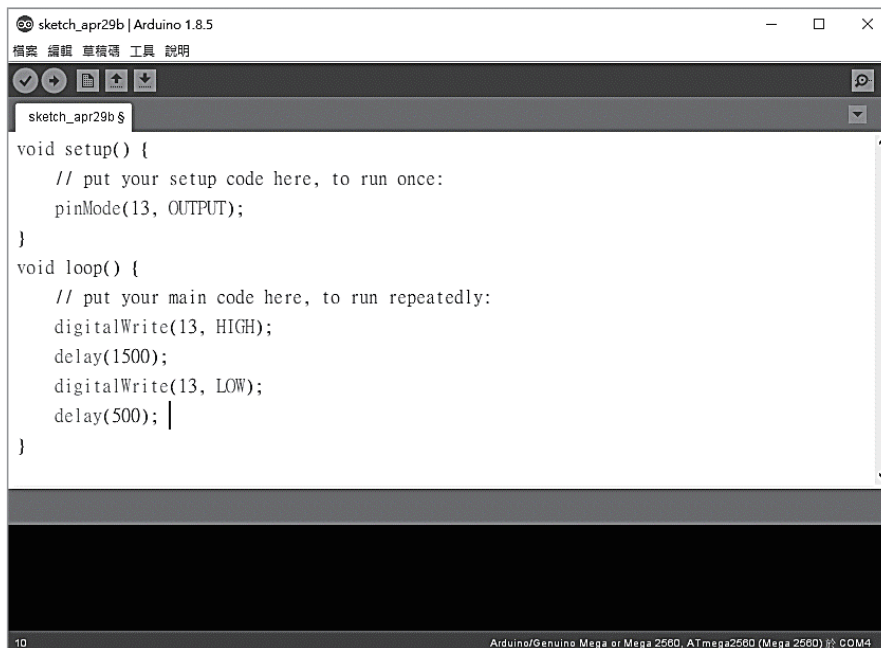
2. 請將以下程式填入 `setup(){}` 大括號裡面。請留意大小寫要一致，打完 `pinMode`，`pinMode` 會自動變紅色字體；打完 `OUTPUT`，`OUTPUT` 會自動變綠色字體，若沒自動變色，那表示您拚字錯誤。還有，分號(;)是半形的，也不能漏打。

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

3. 請將以下程式填入 `loop(){} 大括號` 裡面。

```
digitalWrite(13, HIGH);  
delay(1500);  
digitalWrite(13, LOW);  
delay(500);
```

4. 完成後，如下圖。



5. 存檔。
- (1) 點選功能表『檔案 / 另存新檔』，並自行設定路徑。本書預設路徑是 `d:\ad`，且再以各章名稱分類，例如，`ch01,ch02` 等。
 - (2) 檔名僅輸入主檔名就好，副檔名預設為『`ino`』，請不要輸入。

- (3) 輸入檔名後，系統會自動以檔名建立資料夾。例如，檔名若為 `blink`，那會先自動建立 `blink` 資料夾，再以 `blink.ino` 為檔名，儲存在 `blink` 資料夾內。



6. 驗證。(若有錯誤，請自行修改)
7. 上傳。請觀察執行結果。

自我練習

1. 請鍵入以下程式，並觀察執行結果。(請比較程式放在 `setup()` 與 `loop()` 的不同)

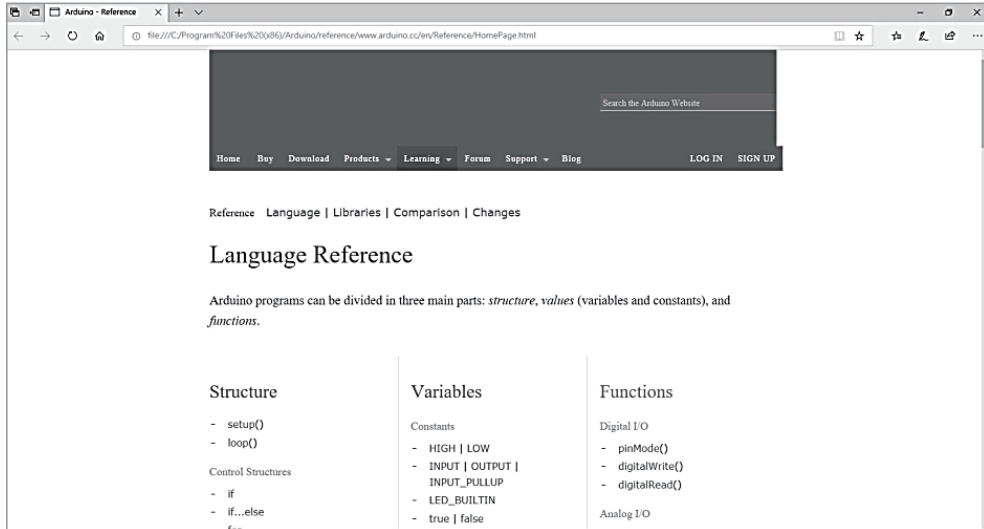
```
void setup() {  
    pinMode(13, OUTPUT);  
    digitalWrite(13, HIGH);  
    delay(1500);  
    digitalWrite(13, LOW);  
    delay(500);  
}  
void loop() {}//絕不能省略
```

2. 假設有一個燈號亮 3 秒，接著熄滅 0.5 秒，亮 0.5 秒閃爍 3 次，最後熄滅 1 秒，重複以上動作，請寫程式完成（請用預植編號 13 的 LED 即可）。

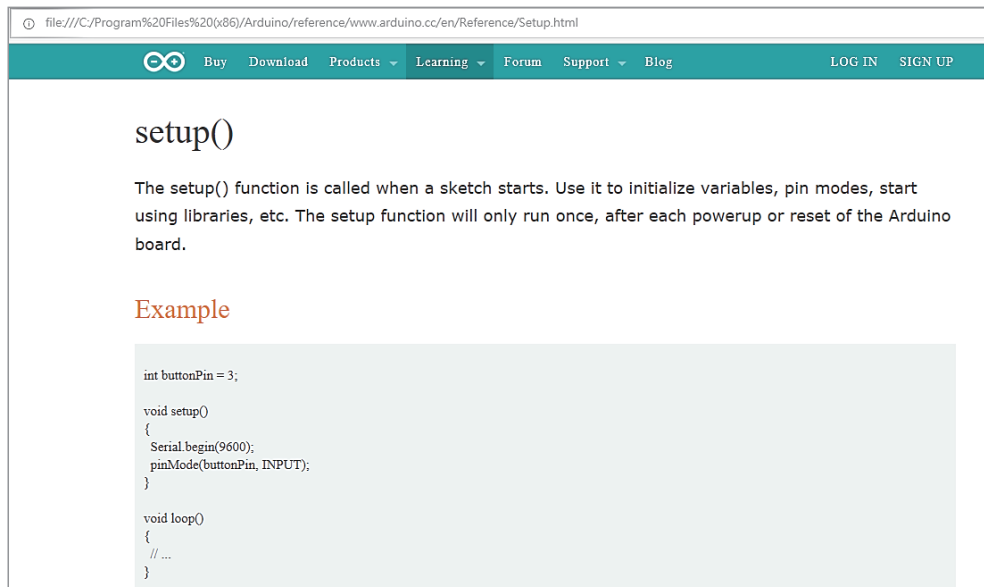
線上使用手冊

Arduino 的線上使用手冊以兩種方式存在，分別是安裝時就放到使用者硬碟的『說明/參考文件』與官網的『RESOURCE/REFERENCE』，兩者內容大致相同，但官網隨時更新，且我認為較詳細。以下是開啓硬碟使用手冊的步驟：點選 Arduino IDE 功

能表的『說明 / 參考文件』畫面如下圖，這就是 Arduino 所提供的軟體指令手冊，不論指令分類、指令解說、範例等都很詳細，請讀者自行探索。



於上圖點選 `setup()`，畫面如下圖，出現 `setup()` 的解說。



下圖是點選 Arduino 官網的『RESOURCE/REFERENCE』的參考文件，此參考文件隨時更新，請讀者自行探索。

