Chapter
03

單晶片微處理機開發流程

學習大綱

- 3-1　高階程式開發流程
- 3-2　程式編輯、編譯及連結
- 3-3　程式的模擬、除錯與燒錄
- 3-4　I/O 腳位探索

學習目標

- 1. 能瞭解高階程式開發流程。
- 2. 能夠完成程式編輯、編譯及連結。
- 3. 能完成程式的除錯與燒錄。
- 4. 能認識微處理機的 I/O 腳位。

3-1 高階程式開發流程

以往開發單晶片微處理機程式僅能使用低階組合語言，要先學習微處理機的暫存器、記憶體與組合語言，現在 Arduino 則使用 C 語言語法搭配 I/O 函式開發產品，所以學習單晶片微處理機開發流程如圖 3-1：

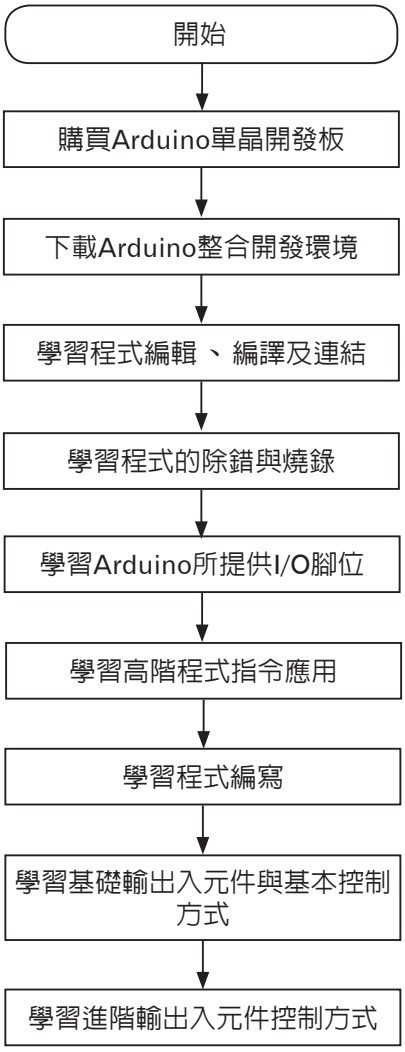

圖 3-1 單晶片微處理機開發流程圖

1. 購買 Arduino 單晶開發板，請看本書第 2 章。
2. 下載 Arduino 整合開發環境，請看本書 3-2 節。
3. 學習程式編輯、編譯及連結，請看本書 3-2 節。
4. 學習程式的除錯與燒錄，請看本書 3-3 節。
5. 學習 Arduino 所提供 I/O 腳位，請看本書 3-4 節。
6. 學習高階程式指令應用，請看本書 4-1 節。
7. 學習程式編寫，請看本書 4-2 節。
8. 學習基礎輸出入元件與基本控制方式，請看本書第 5 章。
9. 學習進階輸出入元件控制方式，請看本書第 6 章。

以上是微處理機高階程式開發流程，微處理機應用實例開發流程將在範例 5-1a 介紹。

3-2 程式編輯、編譯及連結

Arduino軟體下載與安裝

以往所有單晶片的程式開發，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，然後連結若干目的程式成爲一個完整程式。最後還要購買萬元燒錄器，將程式燒到單晶片。但是 Arduino 就方便了，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，此稱爲整合編輯程式（IDE, Integrated Development Environment 的縮寫）。請於 Arduino 官網點選『SOFTWARE/DOWNLOADS』，畫面如圖 3-2，請點選『Windows win10 and newer, 64bits』，即可下載最新安裝執行檔(*.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。

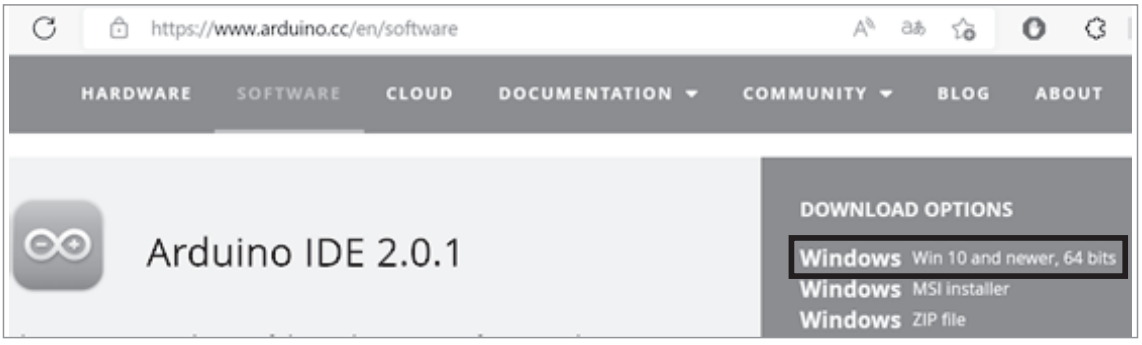

圖 3-2 Arduino 下載點

IDE畫面

圖 3-3 是完成安裝且開啓 Arduino IDE 畫面。

圖 3-3　Arduino IDE 畫面

單晶微控板使用步驟

Arduino 的單晶微控板使用步驟如下：1. 插入微控板、2. 點選開發板型號、3. 點選通訊埠編號、4. 取得開發板資訊，以上動作分別說明如下：

1. **插入微控板**。
   請依照指示，將微控板 USB 插頭插入電腦或筆電 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』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，本書使用 Arduino Mega2560，所以是點選『Arduino/Genuino Mega or Mega 2560』，若您是 UNO 板，也是在此點選，因爲不同版本腳位數量不一樣，選錯了就無法正確編譯程式。

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 com3 或 com4,5,6 等等，讓您點選，有時候會同時出現很多個編號 com3,com4,com5…，請按照順序嘗試點選，直到可上傳 (或稱燒錄) 爲止。其次，有些電腦不會自動抓到通訊埠 (com3、com4…)，請到網路搜尋與下載『CH34x-install-Windows』，並安裝，直到出現通訊埠。）

4. **取得開發板資訊**

請點選功能表的『工具 / 取得開發板資訊』即可取得您的微控板資訊。若出現下圖，才表示以上安裝與設定就緒，才能上傳程式。（實驗的中途，若改插入別人的微控板，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。其次，圖 3-4a 是原廠的微控板，若不是原廠，那可能就像圖 3-4b，顯示『未知的開發板』。）

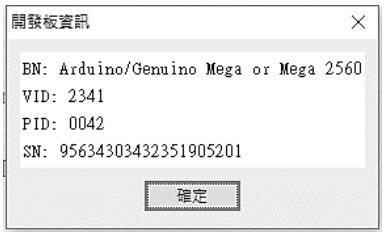

圖 3-4a　原廠開發板資訊

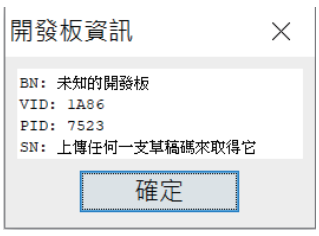

圖 3-4b　非原廠開發板資訊

自我測試

整合開發環境安裝後，內部也含一個自我測試程式，這樣就可以測試此微控板是否已經安裝完成。此自我測試程式使用微控板內植 LED（腳位編號是 13，別名是 LED_BUILTIN）使其明亮 1 秒，熄滅 1 秒。其次，進行單晶片控制都要這樣一步一步來，當問題發生時，才能一步一步除錯，逐漸縮小錯誤範圍，並排除故障。以下說明如何自我測試：

1. **開啓自我測試程式**。

圖 3-5 是開啓測試程式 Blink（請點選功能表的『檔案 / 範例 /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），並令其亮滅各一秒。

```
/*
  Blink
  Turns an LED on for one second, then off for one second, repeatedly.
  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/
// 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
}

```

圖 3-5　Arduino 自我測試程式

以上程式，「/*　*/」之間的文字稱爲註解，此註解僅給人看，微處理機不會處理，剩下的程式如下：

```
1.  // the setup function runs once when you press reset or power the board
2.  void setup() {
3.    // initialize digital pin LED_BUILTIN as an output.
4.    pinMode(LED_BUILTIN, OUTPUT);
5.  }
6.  // the loop function runs over and over again forever
7.  void loop() {
8.    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is
9.                                        the voltage level)
10.   delay(1000);                     // wait for a second
11.   digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by
12.                                       making the voltage LOW
13.   delay(1000);                     // wait for a second
14. }
```

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（上傳又稱爲燒錄）。上傳結束將會自動執行程式，請觀察微控板上所預植 LED 是否亮滅（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. pinMode(LED_BUILTIN, OUTPUT); 是指派編號 13 這隻腳的功能是輸出。
3. 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。
4. digitalWrite(LED_BUILTIN, HIGH); 是指派腳位 13 爲高電位，請用三用電表量其電壓，將會有 5V，這樣就可以讓 LED 發光。
5. delay() 函式是程式延遲程式，單位是毫秒 ms，千分之一秒。因爲指派 LED 亮，總要停留一點時間，使用者才有機會看到。請自行調整時間，並觀察執行結果。
6. digitalWrite(LED_BUILTIN, LOW); 是指派腳位 13 爲低電位，請用三用電表量其電壓，將會有 0V，這樣就可以讓 LED 熄滅。
7. 雙斜線『//』稱爲註解，此爲單列註解，僅給人看，微處理機不予編譯，沒有鍵入也沒關係。其次『/*』與『*/』之間的文字，也都是註解，此稱爲多列註解，這些註解都是給人看的，微處理機不會編譯。

自我練習

1. 請自行修改程式，使得 LED 亮 2 秒，且熄滅 0.5 秒。
2. 請自行修改程式，使得 LED 亮 3 秒、熄滅 1 秒、亮 2 秒、熄滅 0.5 秒、亮 1 秒、熄滅 0.2 秒。

線上使用手冊

點選 Arduino IDE 功能表的『Help/Reference』畫面如圖 3-6（電腦請先連線），此爲 Arduino 所提供的軟體指令參考手冊，不論指令分類、指令解說、範例、函式等解說都很詳細。

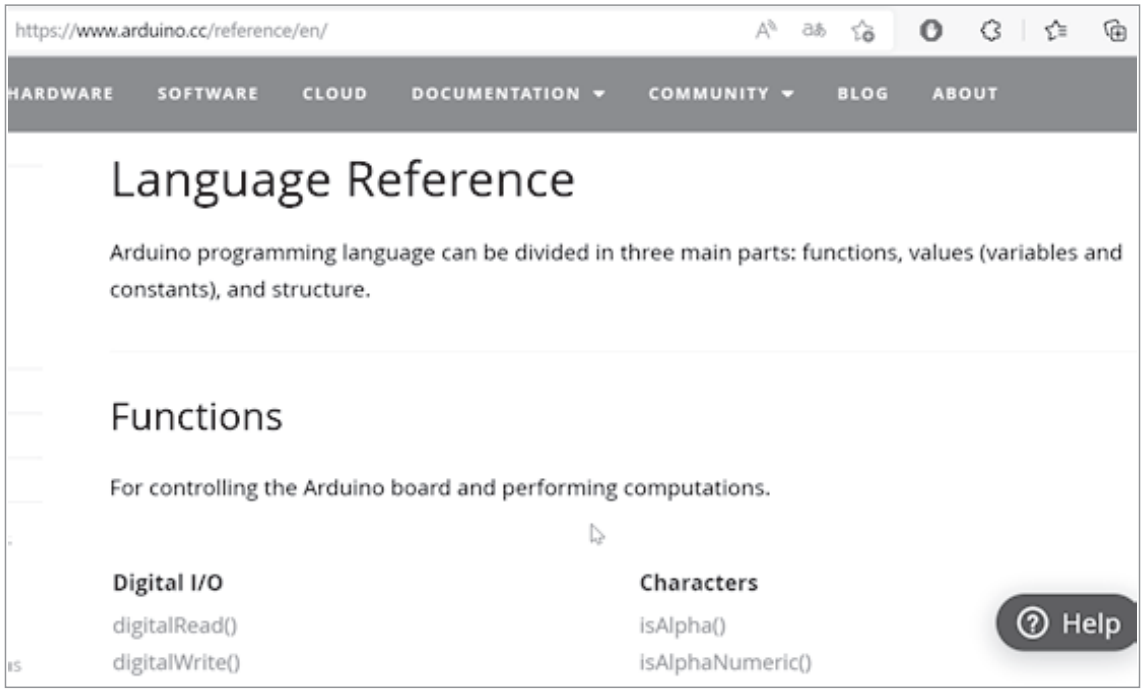

圖 3-6　Arduino 指令、函式參考文件

點選圖 3-6 的 digitalWrite()，畫面出現如圖 3-7。

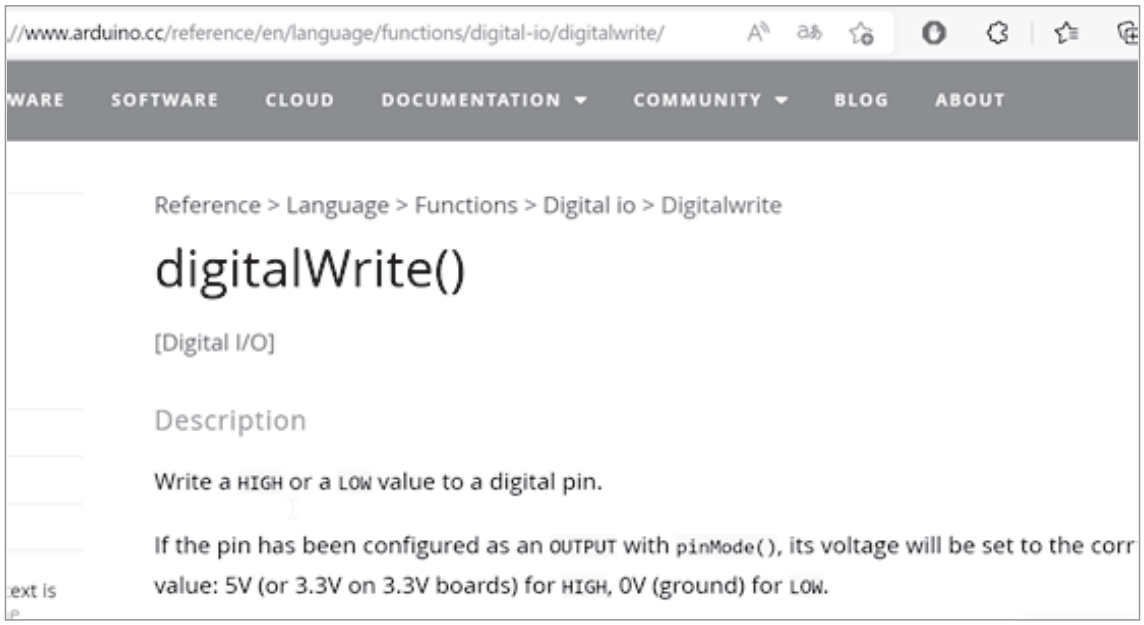

圖 3-7　digitalWrite() 說明畫面

3-3 程式的模擬、除錯與燒錄

程式模擬

以往微處理機使用**可程式唯讀記憶體**（Programmable ROM，PROM）燒錄程式，每個 PROM 僅能燒錄 1 次，所以微處理機廠商都有提供模擬程式，先模擬結果，等到程式都如預期再行燒錄。但是現在 PROM 已經進化爲可重複使用，先是進化使用紫外線抹除的 EPROM，目前則是進化到電子式無限次數抹除的 EEPROM，所以 PROM 成本降低。又因爲 Arduino 提供的序列埠視窗可以先使用電腦的螢幕直接觀察微處理機的執行過程，此一工具不僅可以學習微處理機程式，也可以直接觀察執行結果，比以住的模擬程式完善，所以模擬程式已經沒有存在的必要，Arduino 已經不提供模擬程式。

程式除錯

程式除錯是所有程式設計者的惡夢。程式設計常見的錯誤是指令拚字錯誤、演算法錯誤等，分別說明如下：

▮拚字錯誤

拚字錯誤是初學者最容易發生的錯誤，但是現在很多整合開發環境已經將程式解析，給予提示，能將資料型態、常數、函式、指令解析給予不同的顏色。下圖左是 Arduino 的程式指令解析畫面，共有四種顏色，常數與資料型態解析爲綠色，函式解析爲紅色，指令解析爲灰色，剩下黑色則爲使用者自訂識別字如圖 3-8。所以當鍵入程式並打完資料型態、函式、或指令時，若沒有變色，此時就要馬上更正，直到指令變色爲止。

Blink

```
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 (HI
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by
  delay(1000);                       // wait for a second
}
```

圖 3-8　編譯器指令解析圖

▮演算法錯誤

若程式編譯過程沒有錯誤訊息，結果卻錯誤，此時就要一步一步執行與觀察變數的變化。Arduino 提供序列埠監控視窗用來觀察程式執行過程，是學習程式設計、與除錯程式的最佳工具，請看以下介紹。

序列埠監控視窗

Arduino 的序列埠監控視窗如圖 3-9 所示:(點選『工具』/『序列埠視窗』)

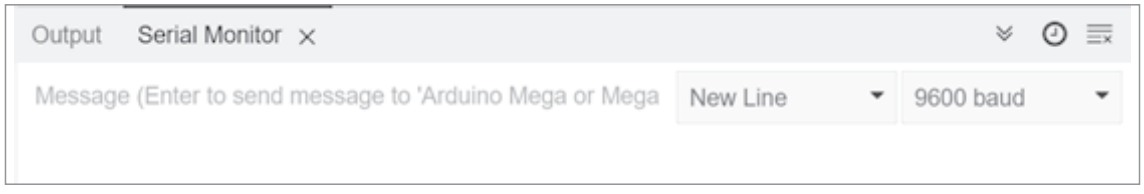

圖 3-9　序列埠視窗

它可以讓我們在程式開發階段，用電腦的鍵盤與螢幕和微處理機雙向溝通。此一功能是以往其他微處理機所沒有的功能，因爲透過此序列埠監控視窗，不僅可以學習 Arduino 語言，也可直接看到程式的執行結果，或在程式執行過程中，將變數內容先行輸出，達到協助除錯的功能，這樣比模擬效果好，可完全取代程式模擬，所以 Arduino 就不提供程式模擬的功能。例如，以下程式可以學習 Arduino 的運算子與數學函式，因爲程式執行後，可在序列埠視窗出現執行結果，如圖 3-10。

```
void setup() {
    Serial.begin(9600);
    Serial.println(4+2);
    Serial.println(4>=2);
    Serial.println(pow(2,3));
} void loop() {}//不可省略
```

```
6
1
8.00
```

圖 3-10　序列埠輸出結果

其次，於開發 Arduino 程式中，從感測器所讀到的值、或運算結果要從輸出元件輸出前，都可將結果先行輸出，先行測試此段程式邏輯與硬體接線是否正確。例如，以下程式，可以將指撥開關的輸入值，先行輸出於電腦螢幕的序列埠視窗，因此可以確認此部分硬體接線是否正確。

```
a=digitalRead(37);//讀取腳位37的電壓值
Serial.println(a);//於序列埠視窗輸出電壓值
```

以下程式，可以先將運算結果先行輸出，這樣可以檢查此軟體運算思維是否正確，然後再輸出於 PORTF。

```
b=a+3//處理
Serial.println(b);//於序列埠視窗輸出電壓值
PORTF=b;
```

Serial物件

電腦或筆電用來編輯微處理機程式，而將程式燒錄微處理機之後，電腦也可以被微處理機控制，電腦的鍵盤、螢幕也可作爲微處理機的 I/O 介面，此功能是以往微處理機所沒有的。要讓電腦的鍵盤與螢幕成爲微處理的輸出入介面，可使用微處理的 Serial 物件。請開啓 Arduino 參考文件 (點選『Help』/『Reference』/『Serial』)。Serial 的 Function 如圖 3-11 所示，這些方法可以讓我們啓用序列埠，然後進行輸出文數字、輸入字元、輸入字串與數值，分別說明如下。(補充說明：以下這些函式，在舊式函式導向程式設計稱爲『函式』，但在目前物件導向的程式設計領域，函式已經改稱爲『方法』)

Functions

If (Serial)
available()
availableForWrite()
begin()
end()
find()
findUntil()
flush()
parseFloat()
parseInt()
peek()
print()
println()
read()
readBytes()
readBytesUntil()
setTimeout()
write()
serialEvent()

圖 3-11　Serial 物件的方法

啓用序列埠

要使用序列埠，首先要使用 begin() 方法啓用序列埠，先規定雙方傳輸速度，本例規定 9600，程式如下：

```
Serial.begin(9600);
```

然後也於序列埠視窗點選相同的速率，如圖 3-12：

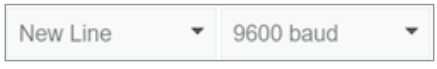

圖 3-12　序列埠傳輸速率

輸出

要在序列埠輸出文數字，要用 print() 或 println() 方法，兩者的差別是 println() 輸出後換行歸位，print() 則沒有。例如，以下程式，可輸出『GwoshengGwosheng』，如下右圖。

`Serial.print("Gwosheng");` `Serial.print("Gwosheng");`	`GwoshengGwosheng`

以下程式可輸出兩列 Gwosheng，如下圖右。

`Serial.println("Gwosheng");` `Serial.println("Gwosheng");`	`Gwosheng` `Gwosheng`

若是 print() 或 println() 方法內接變數，則會轉爲印出此變數的內容。例如，以下程式，可將變數所代表的內容輸出至使用者電腦螢幕。

```
int a=3;//規定資料的儲存空間，宣告a爲整數變數
String b="Gwosheng";//規定資料的儲存空間，宣告b爲字串變數
Serial.println(a);//3
Serial.println(b);//Gwosheng
```

以下程式可輸出『a=3』。有加雙引號『""』的是字串，沒加雙引號的是變數，遇到字串就直接輸出，遇到變數就往前找此變數所儲存的值，並輸出。

```
int a=3;//規定資料的儲存空間，宣告a爲整數變數
Serial.print("a=");//字串就直接輸出a=
Serial.print(a);//變數，輸出變數的值3
```

自我練習

1. 請鍵入以下程式，並寫出執行結果。

```
1.  void setup() {
2.      Serial.begin(9600);
3.      Serial.print("Gwosheng");
4.      Serial.print("Gwosheng");
5.      Serial.println();//only linefeed
6.      Serial.println("Gwosheng");
7.      Serial.println("Gwosheng");
8.      Serial.println();//only linefeed
9.      int a=3;//規定資料的儲存空間，請看4-1節
10.     String b="Gwosheng";//規定資料的儲存空間，請看4-1節
11.     Serial.println(a);
```

```
12.     Serial.println(b);
13.     Serial.print("a=");
14.     Serial.print(a);
15. }void loop() {}//此函式雖然沒有放程式，但也不能刪除
```

2. 請於序列埠監控視窗輸出自己的座號與名字，本例輸出座號後換行歸位，再輸出姓名。
3. 同上題，請在同一列輸出自己座號與名字。

程式的燒錄

以前的單晶燒錄，是要將單晶 IC 從電路板拆下來，先用紫外線抹除資料，然後放到燒錄器燒錄，最後再重新插入電路板。Arduino 則方便了，在整合編輯環境與 Arduino 微控板裡，此單晶 IC 不用一再插拔，使用者的電路透過微控板與單晶片連接，一個按鍵「Upload」  就可直接燒錄。燒錄結束，USB 連接線不用插拔，USB 連接線繼續扮演提供電源角色，微控板內的單晶片轉爲自己當家，掌控使用者所連接的電路（也含電腦的鍵盤與螢幕），自動執行此程式。

3-4 I/O腳位探索

Arduino Mega 2560 如圖 3-13，有 70 個 I/O 接腳，編號爲 0 到 53 與 A0 ～ A15。

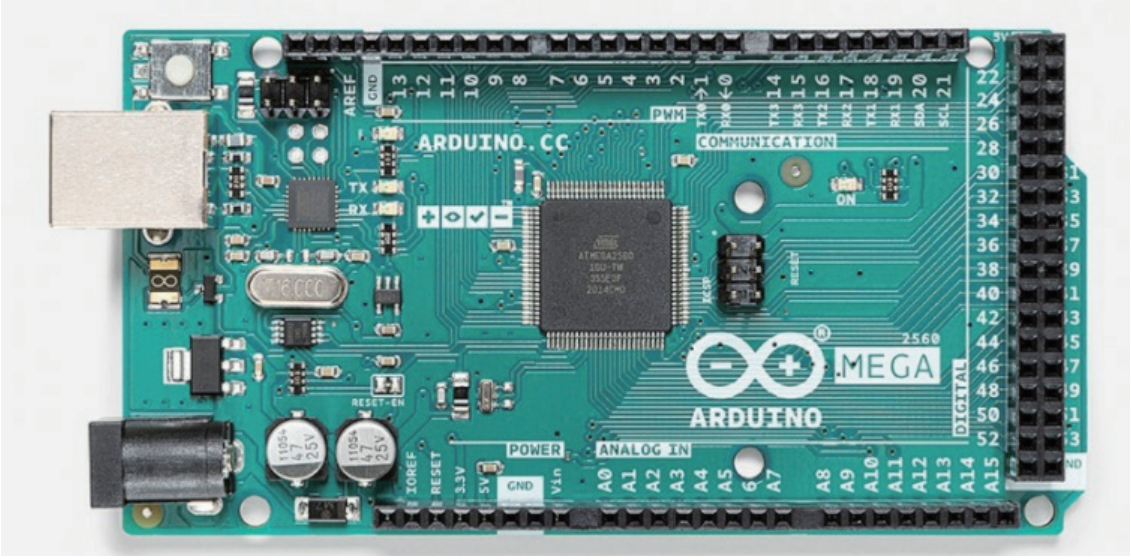

圖 3-13　Arduino Mega 2560 實體圖

這些接腳都可使用腳位編號（例如，0, 1, 2,…A1, A2…）單獨使用，也可使用暫存器名稱指派工作。例如，22, 23, 24, 25, 26, 27, 28, 29 這 8 隻腳暫存器名稱爲 PORTA，亦可以使用暫存器名稱 PORTA，同時使用 22~29 等腳位。Arduino Mega 2560 所有腳位與暫存器名稱，如表 3-1 所示：

▶ 表 3-1　Arduino 暫存器名稱與腳位編號對照表

暫存器\位元	7	6	5	4	3	2	1	0
PORTA	29	28	27	26	25	24	23	22
PORTB	13	12	11	10	50	51	52	53
PORTC	30	31	32	33	34	35	36	37
PORTD	38				18	19	20	21
PORTE			3	2	5		1	0
PORTF	A7	A6	A5	A4	A3	A2	A1	A0
PORTG			4			39	40	41
PORTH		9	8	7	6		16	17

PORTI								
PORTJ						15	14	
PORTK	A15	A14	A13	A12	A11	A10	A9	A8
PORTL	42	43	44	45	46	47	48	49

以上這些接腳都可以使用軟體的設定，指派腳位功能。其次，腳位功能有三種，分別是數位輸出、數位輸入、或有上拉電阻 INPUT_PULL UP 的輸入等 3 種功能，分別說明如下：

指派腳位功能

Arduino 指派腳位功能有兩種方式，分別是單隻腳位的 pinMode 指派與 8 隻腳一起指派的 DDRA 指令（Data Direction of Port A 的縮寫）。例如，以下程式可使用 pinMode 指派腳位 13 作爲數位輸出。

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

以下程式可指派腳位 29,28,27,26,25,24,23,22 作爲數位輸出。

```
pinMode(29, OUTPUT);
pinMode(28, OUTPUT);
pinMode(27, OUTPUT);
pinMode(26, OUTPUT);
pinMode(25, OUTPUT);
pinMode(24, OUTPUT);
pinMode(23, OUTPUT);
pinMode(22, OUTPUT);
```

以上腳位 29 ～ 22 剛好是 PORTA 暫存器，所以以上程式亦可簡化爲

```
DDRA=B11111111;// 1是輸出，指派PORTA爲輸出
```

B 表示後續數字代表二進位，1 表示輸出，0 表示輸入，也可寫成

```
DDRA=0xff;//0x開頭代表後續是16進位數字
```

同理，若是

```
DDRA=B00000000;// 0是輸入，指派PORTA爲輸入
```

則指派 PORTA 爲輸入。也就是 pinMode 是配合腳位名稱，一次指派 1 個腳位的功能；DDRA 是配合暫存器名稱，一次指派 8 個腳位的功能，同理 PORTB、PORTC 就用 DDRB、DDRC 等指派其功能。

數位輸出

Arduino 腳位若當作數位輸出，使用手冊的說明如下：

> **Properties of Pins Configured as OUTPUT**
>
> Pins configured as OUTPUT with pinMode() are said to be in a low-impedance state. This means that they can provide a substantial amount of current to other circuits. Atmega pins can source (provide positive current) or sink (provide negative current) up to 40 mA (milliamps) of current to other devices/circuits. This is enough current to brightly light up an LED (don't forget the series resistor), or run many sensors, for example, but not enough current to run most relays, solenoids, or motors.
>
> Short circuits on Arduino pins, or attempting to run high current devices from them, can damage or destroy the output transistors in the pin, or damage the entire Atmega chip. Often this will result in a "dead" pin in the microcontroller but the remaining chip will still function adequately. For this reason it is a good idea to connect OUTPUT pins to other devices with 470Ω or 1k resistors, unless maximum current draw from the pins is required for a particular application.

大意是說，當輸出時，若設定爲 HIGH，則電壓有 5V，且每隻腳位高電位的最大輸出電流是 20mA；其次，設定爲 LOW 時，可承受或稱流入 40mA 的電流。

指派電位

單晶片的優點是您可直接下指令，指派任何接腳爲 HIGH 或 LOW。指派的方式有兩種，分別是單隻腳位的 digitalWrite 指派與八位元的暫存器名稱 (如 PORTA) 整體指派。例如，以下程式，您可指派接腳 22 爲 HIGH。

```
digitalWrite(22, HIGH);
```

以下程式，您可指派其爲 LOW。

```
digitalWrite(22, LOW);
```

所有腳位都有一個暫存器名稱，還可使用暫存器名稱一起指派 8 隻腳的電位，例如，以下程式可快速指派 PORTA 的 8 個位元全爲 HIGH，PORTA 是暫存器名稱。

```
PORTA=B11111111;
```

以下程式可快速指派 PORTF 的 8 個位元輸出全爲 LOW。

```
PORTF=0;//0就0，當然不用再指派任何進位。
```

範例 3-4a

PORTA 腳位探索實習。

1. 請鍵入以下程式，驗證、上傳。

```
void setup() {
  // put your setup code here, to run once:
    DDRA=B11111111;// 指派PA爲輸出
    PORTA=B11111111;//指派PA爲高電位
}void loop() {}//此loop()函式雖然沒用到，但不能刪除
```

2. 請用三用電表，檢驗微控板的腳位 22, 23, 24, 25, 26, 27, 28, 29 是否爲高電位 5V。
3. 鍵入以下程式，重新驗證、上傳，再檢驗微控板的腳位 22, 23, 24, 25, 26, 27, 28, 29 是否爲低電位 0V。

```
void setup() {
    DDRA=B11111111;// 指派PORTA爲輸出
    PORTA=0;//指派PORTA輸出低電位
}void loop() {}
```

自我練習

1. 請鍵入以下程式，並驗證 PORTA、PORTB、PORTC 對應腳位是否爲高電位。

```
void setup() {
  // put your setup code here, to run once:
    DDRA=B11111111;//指派PA爲輸出
    PORTA=B11111111;//設定PA0～7均爲高電位
    DDRB=B11111111;//指派PB爲輸出
    PORTB=B11111111;//設定PB0～7均爲高電位
    DDRC=B11111111;
    PORTC=B11111111;
} void loop() {}
```

2. 請探索 PORTF、PORTK、PORTL 腳位在哪裡？並寫程式檢驗這些腳位電壓是否能隨著您的程式而變化。

學後測驗

1. 請問 Arduino 安裝完成後，可開啓哪一自我測試程式，此測試程式可以讓內植 LED 連續亮滅？_____
2. 請問 Arduino 程式由哪兩個函式組成？_____，_____。那一個函式僅執行一次，通常當作基本設定？_____，那一個函式無限重複循環執行，通常放主程式？_____
3. 寫出可以讓內植 LED 亮 1 秒的程式？_____
4. 寫出可以讓內植 LED 滅 1 秒的程式？_____
5. 寫出於序列埠視窗輸出「AA」的程式？_____
6. 寫出 PORTA 暫存器由哪些腳位組成？_____
7. 寫出 PORTB 暫存器由哪些腳位組成？_____
8. 寫出指派 PORTC 為輸出的程式？_____
9. 寫出指派 PORTC 為輸入的程式？_____
10. 寫出指派 PORTF 全為高電位的程式？_____
11. 寫出指派 PORTF 全為低電位的程式？_____

MEMO