

導論

1-1 程式語言基本概念、功能及應用

電腦程式語言

人與人之間溝通的工具稱為語言，世界上所有民族，因其發源地不同，所以就有許多語言。例如，華語、英語及德語等。其次，人與電腦溝通的工具，則稱為電腦程式語言。那為什麼沒有電視語言、冰箱語言或冷氣語言呢？那是因為這些機器的功能較為簡單，只要幾個按鈕就能發揮其功能，但是電腦的功能非常多，多到連用整個鍵盤的所有按鍵都無法表現其功能，所以必須使用一些類似單字所組成的片語與敘述來發揮其功能，這些單字與片語的集合就稱為電腦程式語言，簡稱程式語言。就如同人類也無法用 26 個字母表達所有感受與思維，必須藉助這些字母的排列組合，先組成單字，再由單字組合成片語與句子，才能充分表達其思維。目前流行的程式語言有 Java、C、C++、Visual Basic、C#、Python 等。

程式設計

串連一些程式語言中的指令與資料，使其完成一件工作，就稱為程式設計。

程式設計功能

程式設計可將一連串重複的工作，使用程式語言，寫一段程式、儲存，然後就可無限次數重複使用，這樣可節省很多時間。例如，範例 4-3c 的解一元二次方程式為例，使用者必須不斷的計算 $d = \sqrt{b^2 - 4ac}$ ， $x_1 = (-b+d)/2a$ ， $x_2 = (-b-d)/2a$ ，才能解出方程式，那如果寫成程式，每次要求解時，只要輸入係數 a, b, c 即可求出方程式的解。

應用

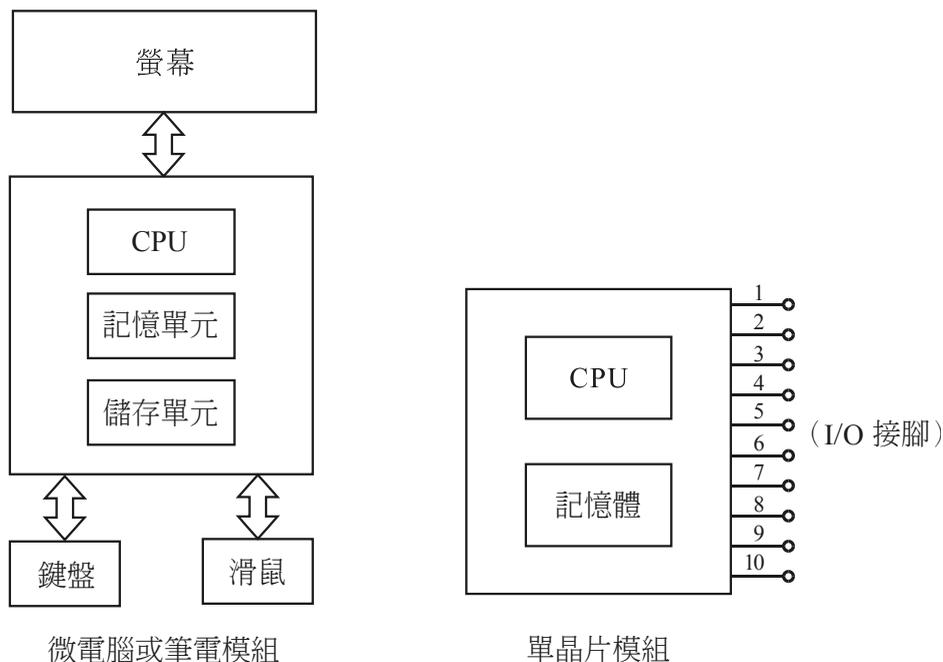
以前的程式設計的應用比較偏重商用資料處理、科學數據的計算，例如，C/C++、Visual Basic、Java、C#、JavaScript、PHP 等等，這些程式設計的輸出入僅用到鍵盤、滑鼠、螢幕與硬碟或光碟等科技設備，只有特定人士、特定功能才會使用電腦。但是以目前的生活科技，無人自駕車已經可以在街上跑、鴻海郭台銘董事長也說五年後他的工廠不用開燈、不用人、不用供餐、不用住宿，使用機器人就可 24 小時生產，所以現在的程式設計可說是融入我們的生活了。還有更可怕的事，現在人口越來越少，會凸顯人口老化問題，居家生活照顧就要仰賴機器人了，每個人眼睛一張開，就要學習與電腦機器人一起生活。所以，本書採用最新的程式語言 Arduino，這樣不但可以解決所有商用、工程與科學的所有計算，也直接配合目前生活科技的發展近況，解決當前程式設計的洪流。

1-2 單晶片 Arduino 簡介

微電腦 VS 單晶片

現在幾乎家家戶戶都有桌上型微電腦或筆電，所謂微電腦或筆電就是整合 CPU、記憶單元、儲存裝置（硬碟與光碟等）、輸出入單元（鍵盤、滑鼠、螢幕）等零件，並協調其一起工作的設備，如下圖左。但是單晶片就神奇了，它竟然可以將 CPU、記憶單元整合在一個 IC 裡面，然後預留若干 I/O 接腳如下圖右，這些 I/O 接腳可以讓使用者，以軟體程式的方式，設定低電位或高電位，進而控制所有負載的 ON 與 OFF。傳統的自動控制，都要使用硬體

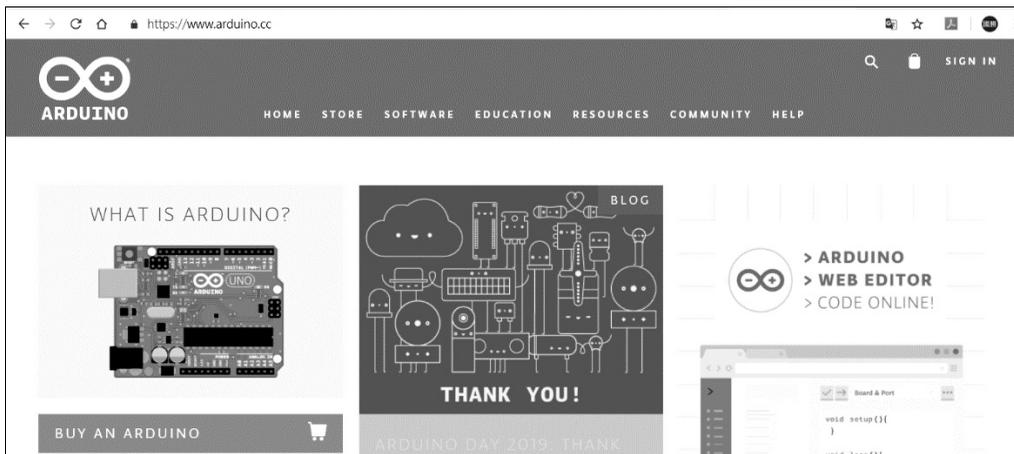
接線，所以體積龐大。但是，使用單晶片控制就神奇了，不但體積小、價格低，控制也非常彈性，所以現在幾乎所有家電都使用單晶片控制了。以傳統電鍋為例，它是利用雙金屬片當作開關，所以傳統電鍋僅有單一功能，那就是僅能煮飯；電子鍋的功能就多了，它因為使用單晶片當作控制，所以可以依照穀物的不同、或要吃白飯或稀飯而調整炊煮時間；冷氣機、洗衣機的功能也很多，冷氣機可設定恆溫、洗衣機可依衣物多寡自動調整水量與時間；汽車裡的單晶數量就更可觀了，所有安全配備與自動駕駛技術都有單晶聯合控制。機器人也是要有許多光、熱、距離、色彩等感測器，這樣才能做出決策，進行下一步動作，這也要藉助細小單晶片的協助，才能發揮機器人功效。所以，目前單晶片的普及率已經非常高了。



什麼是 Arduino

Arduino 其實就是一顆新開發的微小單晶片，那為什麼如此地快速竄紅呢？請看以下說明。請開啓 Arduino 官網 (<https://www.arduino.cc/>)，畫面如下：

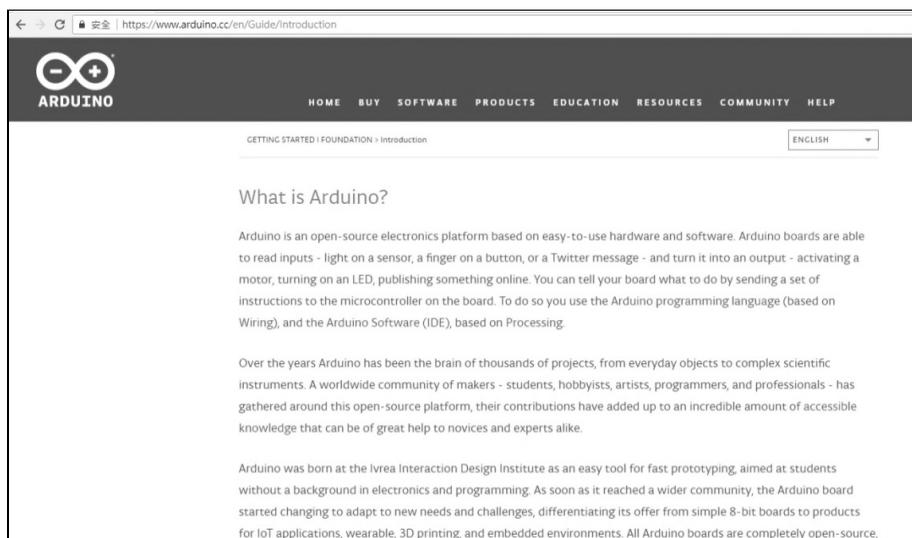
全民自造與程式設計 使用 Arduino



請點選上圖『WHAT IS ARDUINO』，畫面如下：



這個畫面有三個大的選項，分別是『WHAT IS ARDUINO』、『ARDUION BOARD』與『ARDUION SOFTWARE』，先展開第一個選項『WHAT IS ARDUION』，按一下『Learn more about Arduino』，畫面如下圖。



以上畫面說明 Arduino 產品的緣由，大致上是說它在 2003 年由義大利的 Ivrea 互動設計學院所研發，主要是希望不懂電子、不懂程式設計的人，也能非常簡單、非常快速且以非常少量的金錢開發出真實世界的科技產品。它提供一塊造價非常便宜（不超過 50 美金）且體積很小的微控板，讓您能接收一些感測器，且做出一些判斷，就能設計出很多且有互動效果的產品。例如，機器人、3D 印表機、多軸飛行器、物聯網裝置等等。

將以上畫面往下捲，畫面如下圖，此畫面說明了 Arduino 的特色如下，共有五點，以下畫面寫得非常淺顯、清楚易懂（有不懂的單字，請複製，並貼到翻譯軟體就行）。

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

筆者將以上特色簡單說明如下：

1. 非常便宜。是目前最便宜的微控板，不到 50 元美金就可買到相容品，而且也不用額外購買萬元燒錄器，就可編輯、編譯（驗證）、燒錄（上傳）與執行程式。
2. 跨平台。所有作業系統都可以使用此軟體，且操作方式都接近 Windows。
3. 整合操作環境非常簡潔。Arduino 整合操作環境的確簡潔，在同一環境就可編輯、編譯、燒錄、執行，是我目前用過最簡單的整合操作環境。

4. 軟體為開放原始碼。因為便宜，且開放原始碼，所以每天都有成千上萬的人在網站貢獻自己的成果。
5. 硬體也為開放性電路。因為硬體線路公開，所以任何人也都可生產此微控板，所以硬體成品已接近材料成本。

本人學電子、學程式、教程式迄今 30 年了，非常認同以上特色，所以選用此軟體作為中學生程式設計入門的程式語言，且本人增加一點特色如下：

6. 使用手冊非常詳盡易懂

Arduino 的使用手冊非常詳盡、淺顯易懂還有很多範例，是我目前學過所有程式語言中，內建使用手冊最詳細的語言，且其英文表達方式非常淺顯易懂，非常適合初學者查閱與學習，在此大力推薦。

晶片微控板

一個單晶片 IC，再配合電源、穩壓、過載防護、傳輸、燒錄及拉出 I/O 接腳，稱為單晶微控板，簡稱微控板。繼續按一下第二選項的『Discover the official Arduino boards』，畫面如下圖。

The screenshot shows the 'Arduino Products' page. On the left is a navigation menu with links: Summary, Entry Level, Enhanced Features, IoT, Education, Wearable, and Retired. The main content area is titled 'Arduino Products' and contains a brief description: 'Browse the full range of official Arduino products, including Boards, Modules (a smaller form-factor of classic boards), Shields (elements that can be plugged onto a board to give it extra features), and Kits. If you need more info you can compare the specs of each board here. If you are wondering if your Arduino board is authentic you can learn how to spot a counterfeit board here.'

The products are organized into three main categories, each with a list of individual product buttons:

- ENTRY LEVEL:** UNO, LEONARDO, 101, ESPORA, MICRO, NANO, MINI, MKR2UNO ADAPTER, STARTER KIT, LCD SCREEN
- ENHANCED FEATURES:** MEGA, ZERO, DUE, MEGA ADK, MO, MO PRO, MKR ZERO, MOTOR SHIELD, USB HOST SHIELD, PROTO SHIELD, MKR PROTO SHIELD, 4 RELAYS SHIELD, MEGA PROTO SHIELD, MKR RELAY PROTO SHIELD, ISP, USB2SERIAL MICRO, USB2SERIAL CONVERTER
- INTERNET OF THINGS:** YÜN, ETHERNET, TIAN, INDUSTRIAL 101, LEONARDO ETH, MKR FOX 1200

由上圖可知其微控板型號可說非常多，有入門款、加強款、物聯網款、教育總包套件款、穿戴式輕薄款、停產退役款等，這麼多款各有其用途，初學者當然眼花繚亂，請先比較其價格、I/O 腳位數與體積就好。其次，目前市面上書籍大都以 UNO 為對象，但本書選用 MEGA 2560，我的主要理由是，MEGA 2560 的 I/O 腳位數目是 UNO 的三倍多（MEGA 是 54 隻/UNO 是 14 隻），但價格卻不到 UNO 的兩倍。I/O 腳位數目多，那第九章的掃描電路就會簡潔，例如，本書的四位數七段顯示器、8×8 點陣 LED，就比別書的軟硬體都還簡潔。而且每一個實驗都有專用腳位，作完不用拆，最後要做專題時，可以讓這些腳位同時連接多種 I/O 感測器與多種設備同時動作。下圖是 MEGA 2560 微控板的總論，隨意瀏覽就好，知道它有 54 隻 I/O 接腳就好。

OVERVIEW TECH SPECS DOCUMENTATION

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila.

The Mega 2560 is an update to the Arduino Mega, which it replaces.

You can find here your board warranty informations.

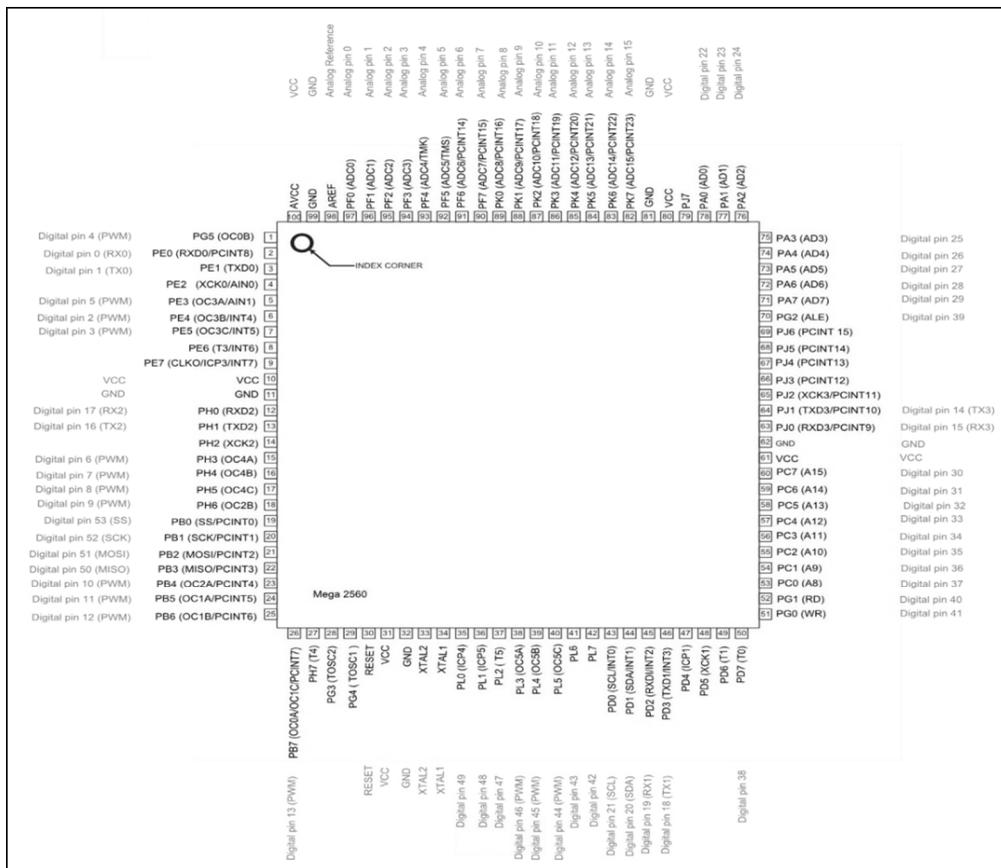
Getting Started

You can find in the Getting Started section all the information you need to configure your board, use the Arduino Software (IDE), and start tinker with coding and electronics.

下圖是其技術規格，看看瀏覽就好，反正記憶體和腳位是多到初學者用不完。

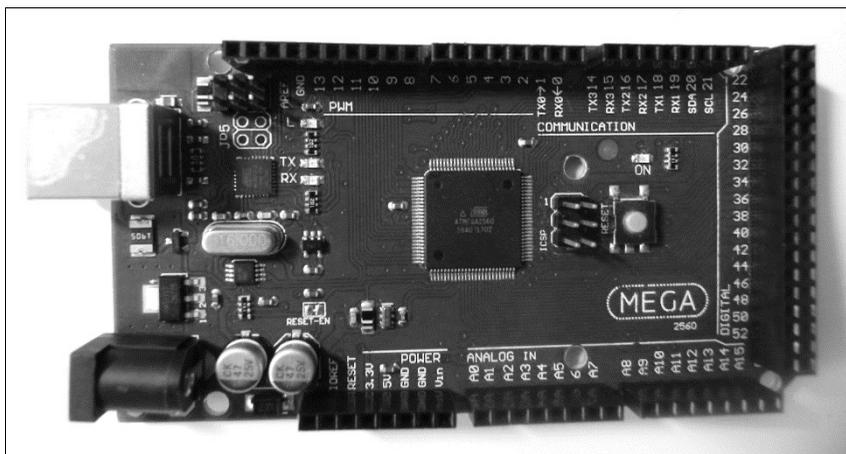
OVERVIEW	TECH SPECS	DOCUMENTATION
Microcontroller	ATmega2560	
Operating Voltage	5V	
Input Voltage (recommended)	7-12V	
Input Voltage (limit)	6-20V	
Digital I/O Pins	54 (of which 15 provide PWM output)	
Analog Input Pins	16	
DC Current per I/O Pin	20 mA	
DC Current for 3.3V Pin	50 mA	
Flash Memory	256 KB of which 8 KB used by bootloader	
SRAM	8 KB	
EEPROM	4 KB	
Clock Speed	16 MHz	
LED_BUILTIN	13	
Length	101.52 mm	

下圖是其 IC 的腳位圖（單晶片是 ATmega2560）（此文件在 DOCUMENTATION 裡面），這也不重要，因為您用不到，這留給電子科系去琢磨就好。



下圖是 MEGA 2560 微控板的實體照片，ATMAGA 2560 IC 放在中間，這顆稱為單晶片，旁邊的方形洞洞都是連接這顆 IC 的 I/O 腳位，這些腳位就可以讓使用者外接輸入或輸出元件，然後使用軟體程式，讀取、或指派這些腳位的電壓。

腳位編號請您配合下圖微控板，上面左邊編號是 0~13，上面右邊是 14~21，右邊雙排是 22~53，但 23,25,27,29 剛好缺角沒印出。以上都是數位接腳，電壓只有兩種情況，0 或 5V；下面則是 A0~A15，這些稱為類比腳位，它可以模擬輸出 0~5V 電壓，或輸入 0~5V 電壓。



下圖是微控板腳位功能對照表。這就很重要了，就如同不會種水果沒關係，但要會吃水果，也就是吃香蕉、吃鳳梨知道要削皮就好。

微控板腳位	功能一	功能二	暫存器名稱
0	數位輸出入	RX0	PE0
1	數位輸出入	TX0	PE1
2	數位輸出入	PWM(類比輸出)	PE4
3	數位輸出入	PWM(類比輸出)	PE5
4	數位輸出入	PWM(類比輸出)	PG5
5	數位輸出入	PWM(類比輸出)	PE3
6	數位輸出入	PWM(類比輸出)	PH3
7	數位輸出入	PWM(類比輸出)	PH4
8	數位輸出入	PWM(類比輸出)	PH5
9	數位輸出入	PWM(類比輸出)	PH6
10	數位輸出入	PWM(類比輸出)	PB4
11	數位輸出入	PWM(類比輸出)	PB5

微控板腳位	功能一	功能二	暫存器名稱
12	數位輸出入	PWM(類比輸出)	PB6
13	數位輸出入	PWM(類比輸出)	PB7
14	數位輸出入	TX3	PJ1
15	數位輸出入	RX3	PJ2
22	數位輸出入	AD0	PA0
23	數位輸出入	AD1	PA1
24	數位輸出入	AD2	PA2
25	數位輸出入	AD3	PA3
26	數位輸出入	AD4	PA4
27	數位輸出入	AD5	PA5
28	數位輸出入	AD6	PA6
29	數位輸出入	AD7	PA7
30	數位輸出入		PC7
31	數位輸出入		PC6
32	數位輸出入		PC5
33	數位輸出入		PC4
34	數位輸出入		PC3
35	數位輸出入		PC2
36	數位輸出入		PC1
37	數位輸出入		PC0
38	數位輸出入		PD7
39	數位輸出入		PG2
40	數位輸出入		PG1
41	數位輸出入		PG0
42	數位輸出入		PL7
43	數位輸出入		PL6
44	數位輸出入		PL5
45	數位輸出入		PL4
46	數位輸出入		PL3
47	數位輸出入		PD4
48	數位輸出入		PL1
49	數位輸出入		PL0
50	數位輸出入		PB3
51	數位輸出入		PB2
52	數位輸出入		PB1
53	數位輸出入		PB0
A0	數位輸出入	類比輸入	PF0

微控板腳位	功能一	功能二	暫存器名稱
A1	數位輸出入	類比輸入	PF1
A2	數位輸出入	類比輸入	PF2
A3	數位輸出入	類比輸入	PF3
A4	數位輸出入	類比輸入	PF4
A5	數位輸出入	類比輸入	PF5
A6	數位輸出入	類比輸入	PF6
A7	數位輸出入	類比輸入	PF7
A8	數位輸出入	類比輸入	PK0
A9	數位輸出入	類比輸入	PK1
A10	數位輸出入	類比輸入	PK2
A11	數位輸出入	類比輸入	PK3
A12	數位輸出入	類比輸入	PK4
A13	數位輸出入	類比輸入	PK5
A14	數位輸出入	類比輸入	PK6
A15	數位輸出入	類比輸入	PK7
VCC	編號 22,23 上面也是 VCC		
GND	編號 52,53 下面也是 GND		

☉ 補充說明

上表的『功能一』與『功能二』是什麼意思？就是有些腳位有兩個功能。例如，腳位 0 和 1 就有兩個功能，這兩個腳位除了可當數位輸出，還可作為單晶與電腦之間的傳輸，所以使用這兩個腳位就要特別小心，要等到資料上傳完畢，才可接上負載。所以，除非腳位不夠用，不然就不要用這兩隻腳當作數位輸出入。

下表是暫存器名稱與微控板腳位對照圖。

	7	6	5	4	3	2	1	0
PA	29	28	27	26	25	24	23	22
PB	13	12	11	10	50	51	52	53
PC	30	31	32	33	34	35	36	37
PD	38							
PE			3	2	5		1	0
PF	A7	A6	A5	A4	A3	A2	A1	A0
PG			4			39	40	41

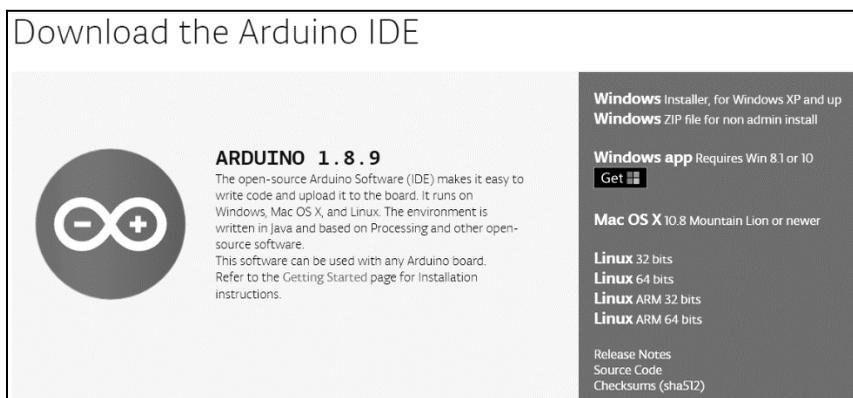
PH		9	8	7	6		16	17
PI								
PJ	未接	未接	未接	未接	未接			
PK	A15	A14	A13	A12	A11	A10	A9	A8
PL	42	43	44	45	46	47	48	49

以上暫存器的使用請看本書 2-3 節，請特別留意只有 PA、PB、PC、PF、PK 及 PL 很完整，這六個暫存器除了可以單隻腳位指派電壓，還可用暫存器名稱，8 位元一起指派電壓，也是 2-3 節有詳細介紹。

1-3 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。



➤ IDE 畫面

下圖是開啓 Arduino IDE 畫面。

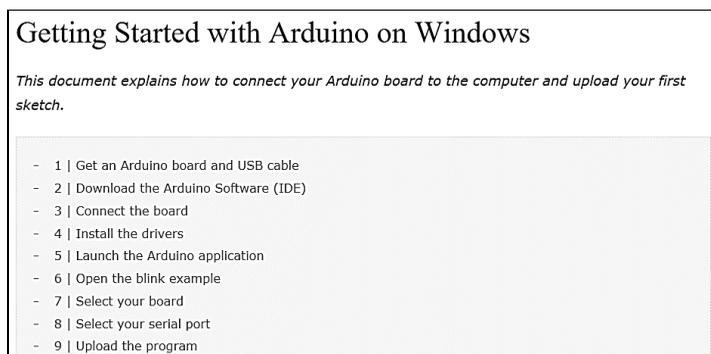


➤ 功能表的語言設定

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：



我們將上圖 Arduino 的單晶微控板使用步驟說明如下：

➤ 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。

➤ 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』）

➤ 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讓您點選）。

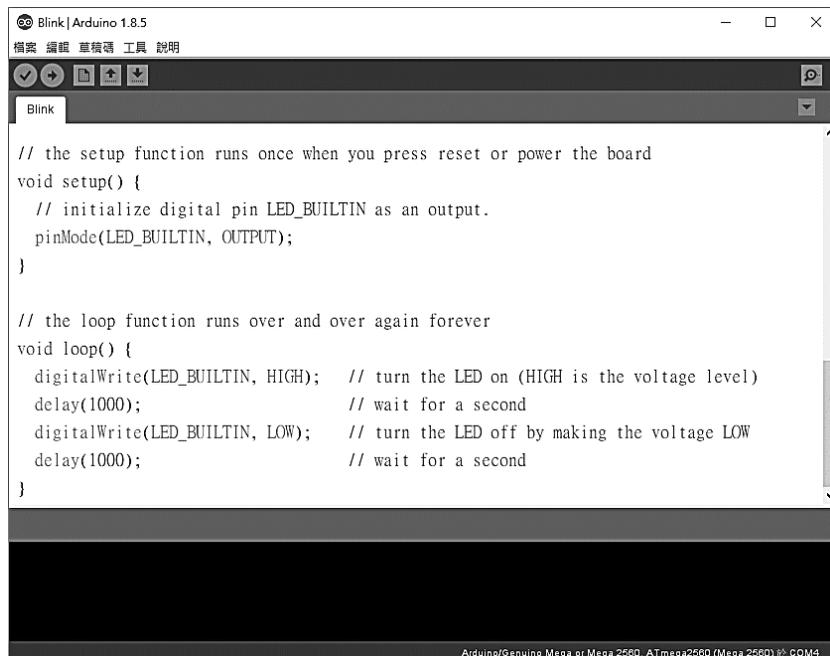
➤ 取得開發版資訊

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，才表示有抓到微控板，才能上傳程式。）



➤ 開啓測試程式

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...），都是腳位 13，以常數『LED_BUILTIN』表示），並令其閃爍明滅各一秒。



```
Arduino IDE - Blink | Arduino 1.8.5
檔案 編輯 草稿簿 工具 說明
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
}
```

➤ 驗證程式

按一下工具列的『驗證』按鈕 ，即可編譯程式。

➤ 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。

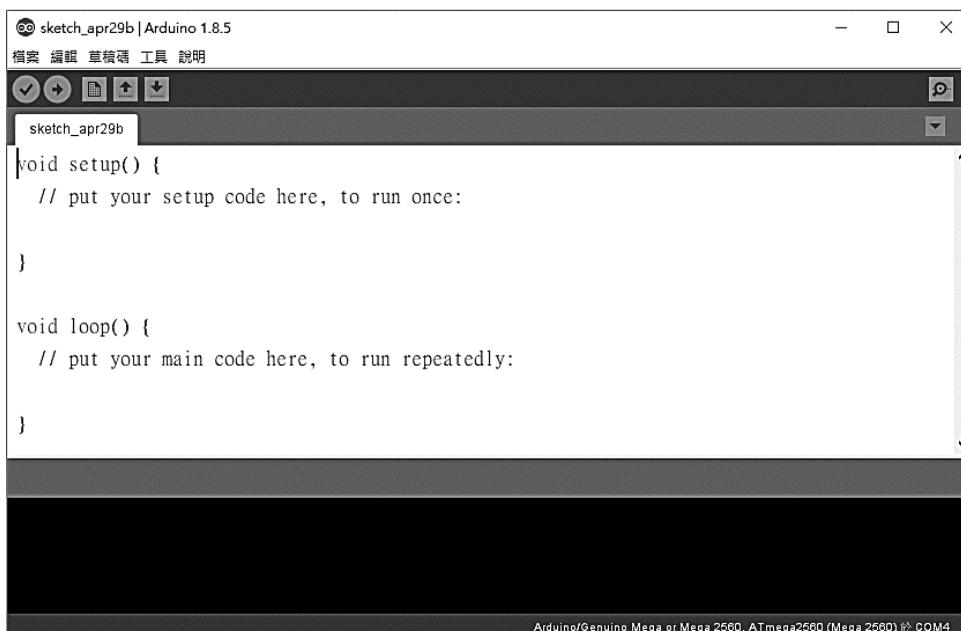
2. loop()函式則會重複不斷的被執行，所以就放置一些需要反覆執行的指令。
3. delay()函式是程式延遲程式。因為指派 LED 亮，總要停留一點時間，使用者才有機會看到。
4. 雙斜線『//』稱為註解，僅給人看，電腦不予編譯，沒有鍵入也沒關係，請看 4-2 節。

▶ 範例 1-3a

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
```

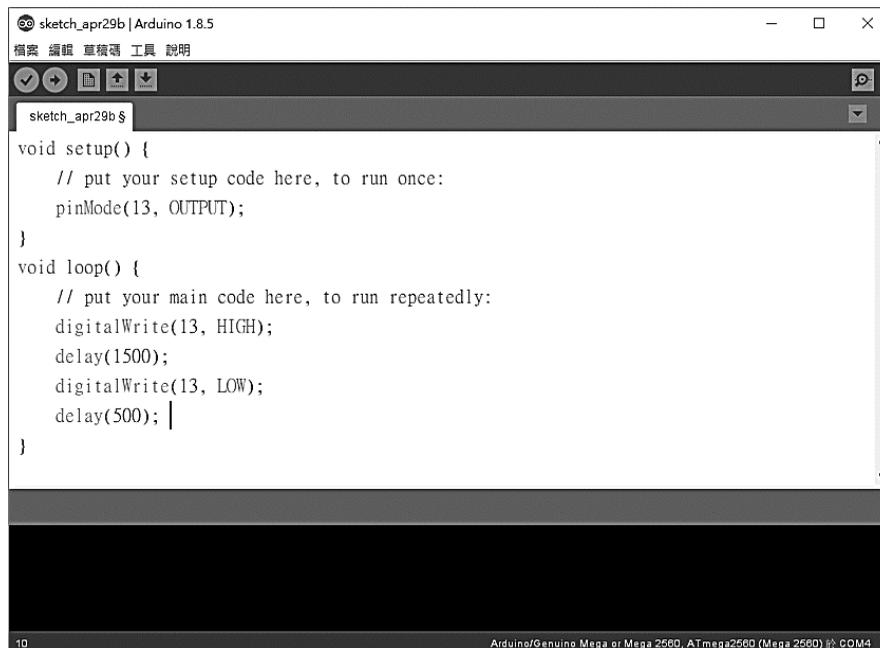
- 2 請將以下程式填入 setup(){}大括號裡面。請留意大小寫要一致，打完 pinMode，pinMode 會自動變紅色字體；打完 OUTPUT，OUTPUT 會自動變綠色字體，若沒自動變色，那表示您拚字錯誤。還有，分號 (;) 是半形的，也不能漏打。

```
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

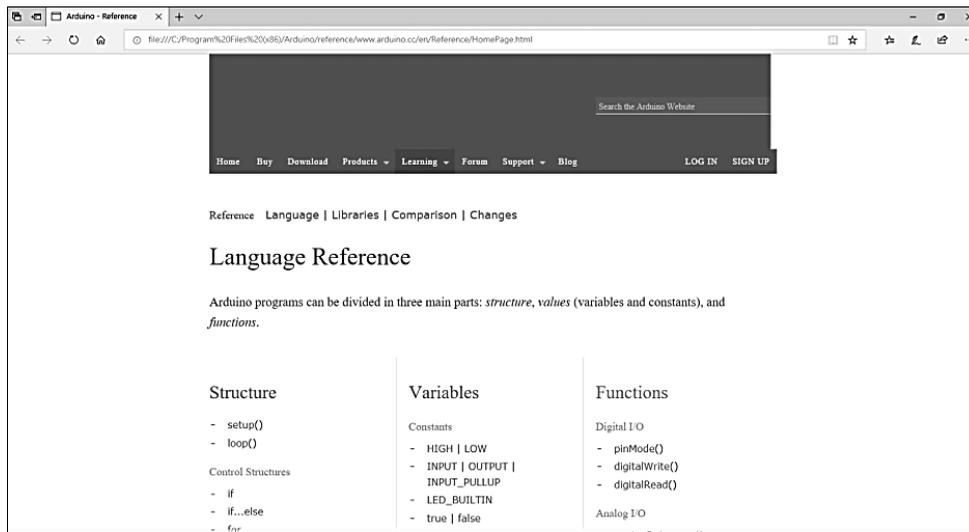
1. 請鍵入以下程式，並觀察執行結果。(請比較程式放在 setup()與 loop()的不同)

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

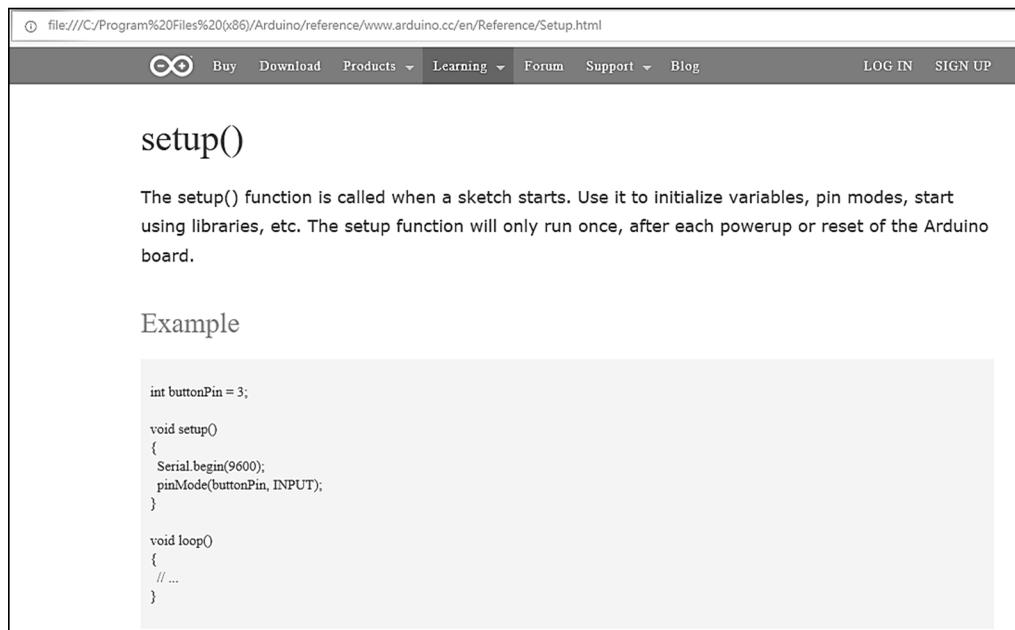
2. 假設有一個燈號亮 5 秒，接著以 0.5 秒閃爍 3 次，熄滅 3 秒，重複以上動作，請寫程式完成（請用預植編號 13 的 LED 即可）。

➤ 線上使用手冊

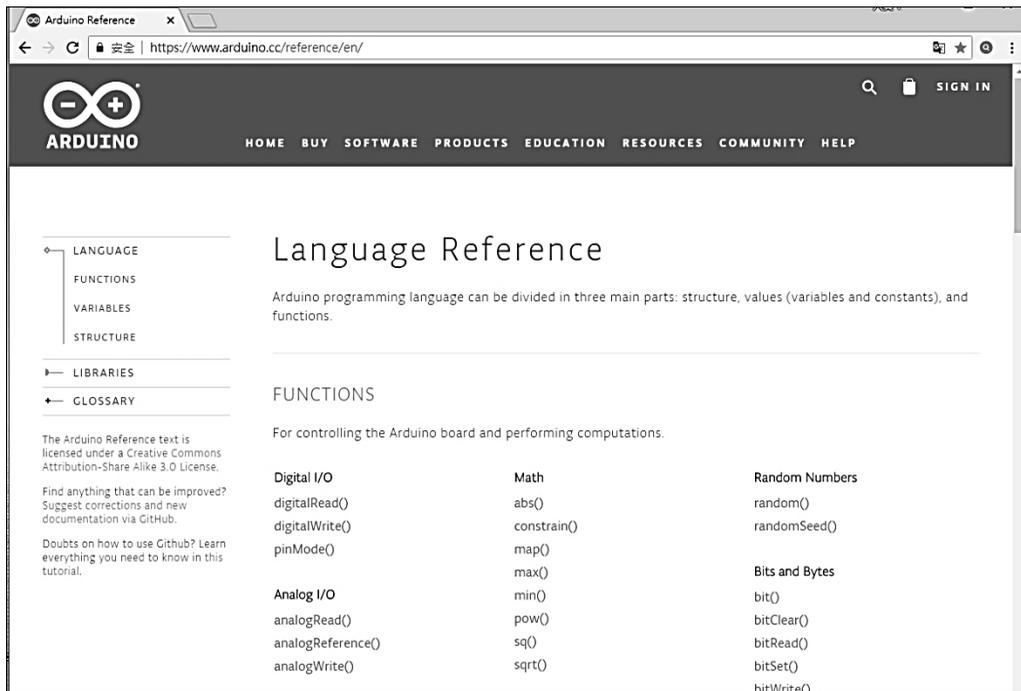
Arduino 的線上使用手冊以兩種方式存在，分別是安裝時就放到使用者硬碟的『說明/參考文件』與官網的『RESOURCE/REFERENCE』，兩者內容大致相同，但官網隨時更新，且我認為較詳細。以下是開啓硬碟使用手冊的步驟：點選 Arduino IDE 功能表的『說明/參考文件』畫面如下圖，這就是 Arduino 所提供的軟體指令手冊，不論指令分類、指令解說、範例等都很詳細，請讀者自行探索。



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。



➤ 資料搜尋

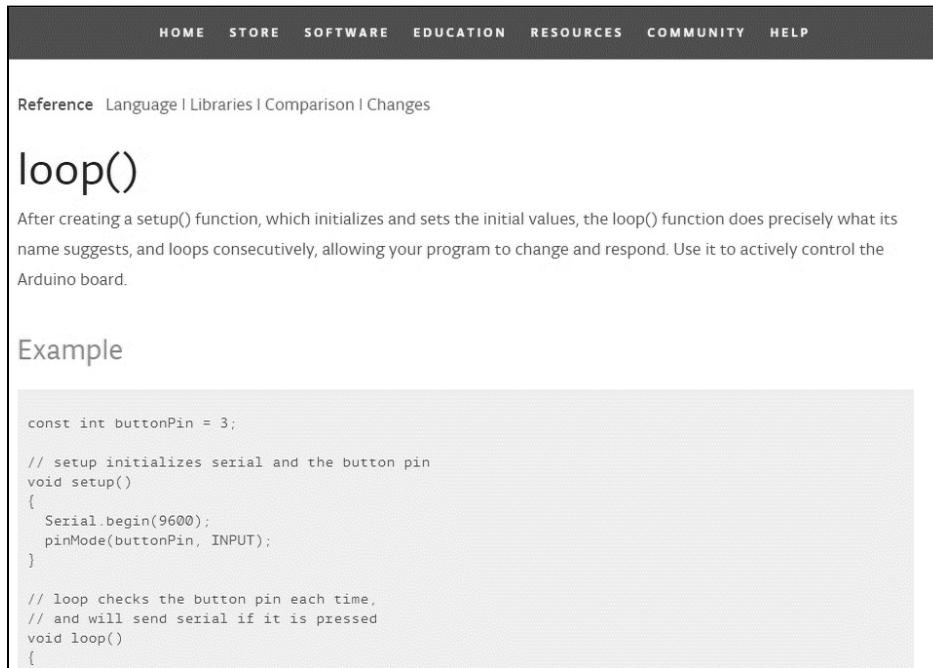
於上圖按一下『放大鏡』圖項，畫面出現如下圖，可於下圖輸入您要查詢的關鍵字。例如，下圖我已經輸入『loop』：



並按一下『Enter』，那就可以找出『loop』的說明，如下圖所示：



繼續按一下上圖的『Arduino loop』，即可得到『loop』的說明，如下圖：



The screenshot shows the Arduino IDE documentation page for the `loop()` function. At the top, there is a navigation bar with links: HOME, STORE, SOFTWARE, EDUCATION, RESOURCES, COMMUNITY, and HELP. Below the navigation bar, there is a breadcrumb trail: Reference | Language | Libraries | Comparison | Changes. The main heading is `loop()`. The text below the heading explains that after creating a `setup()` function, the `loop()` function does precisely what its name suggests, looping consecutively to allow the program to change and respond. An example code block follows, showing the initialization of a button pin and the `loop()` function.

```
const int buttonPin = 3;

// setup initializes serial and the button pin
void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

// loop checks the button pin each time,
// and will send serial if it is pressed
void loop()
{
```

