

Arduino 開放硬體介紹與 韌體安裝

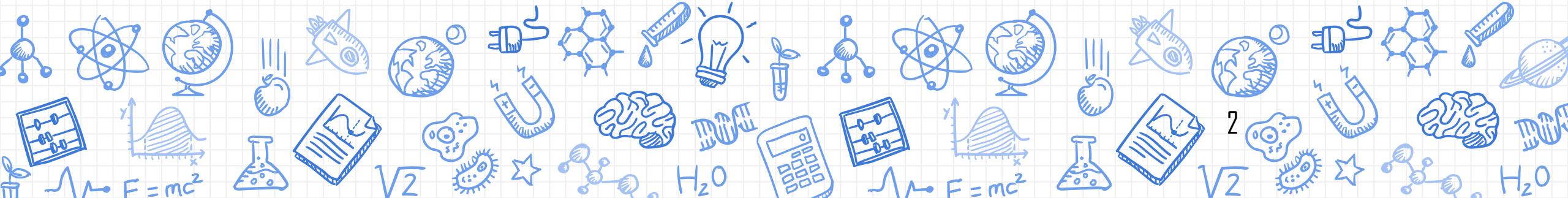


Outline

- 開放硬體介紹
- Arduino 環境架設
- Scratch For Arduino (S4A)
- 自由發揮小專題

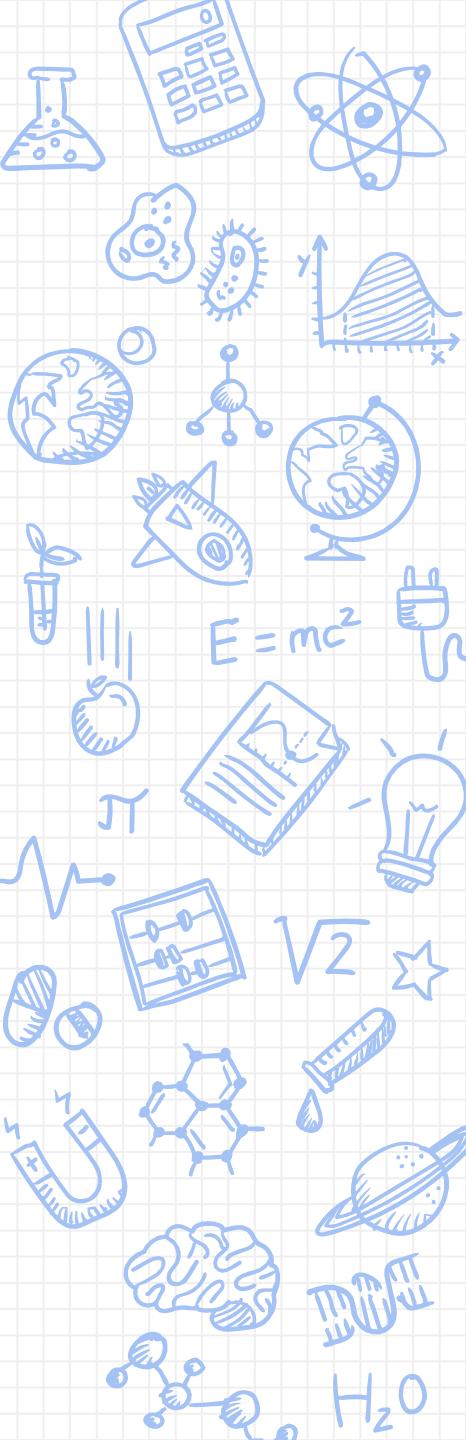
Outline

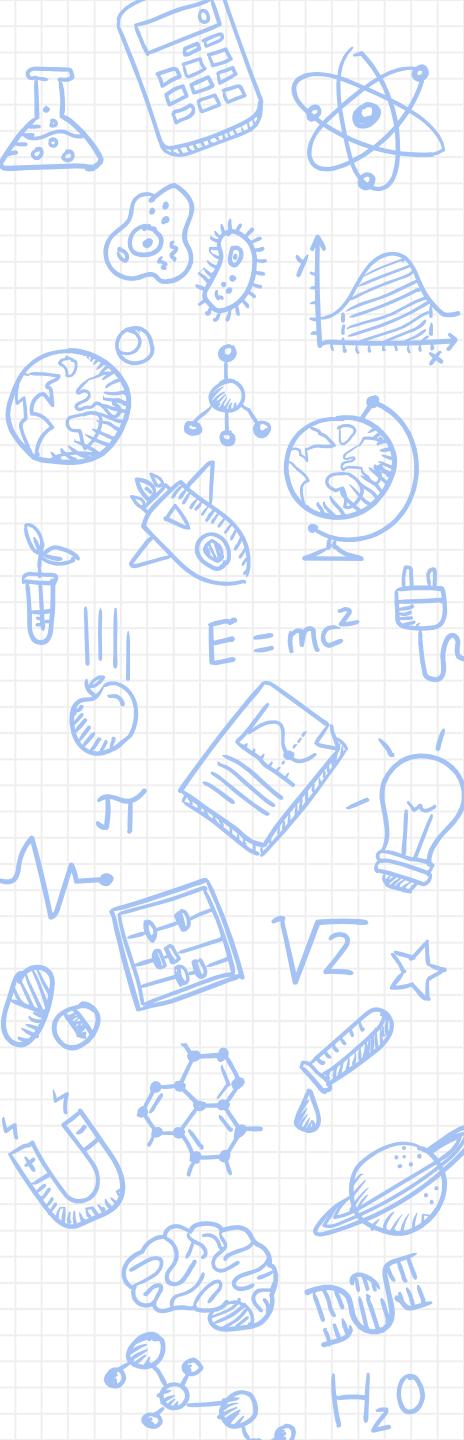
- 開放硬體介紹
- Arduino 環境架設
- Scratch For Arduino (S4A)
- 自由發揮小專題



開放硬體介紹 (1/2)

- 開放硬體(open hardware)，意即設計者將設計圖、零件、線路、布局……等技術開放給每個人使用。
- 雖然名為開放，但使用者仍必須遵守主要規則 GPL (general public license 通用公共授權條款)。
- 特色為容易上手、售價低、網路說明文件及社群討論廣泛，能夠輕易在他人範例程式中找尋自己需要的元素。





開放硬體介紹 (2/2)

- 從泛科學(PanSci網路雜誌)的分類中，開放硬體分成四個層次：

第一層，IC 層：最底層，即為將 IC 的設計圖開放，這部份還不成熟。

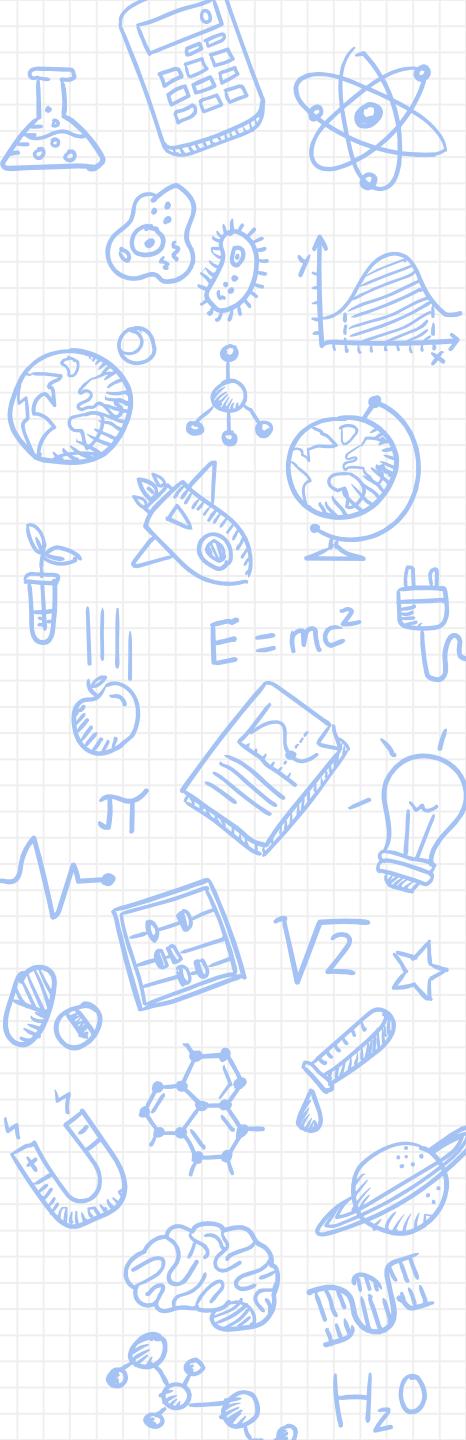
第二層，board level：目前市場上流行的 *arduino*、*raspberry pi*、……等等。

第三層，system level：這一層開放裝置，需要高度整合軟硬體資源，例如 3D 列表機需要整合 3D 繪圖軟體以及 3D 輸出裝置。

第四層，跨領域整合層：這一層開放裝置，除了整合嵌入軟硬體之外，還整合了生物醫學等不同領域的專業知識，例如個人DNA檢測機。

Arduino 介紹 (1/2)

- Arduino，採用開放原始碼的微控制器，擁有簡易的GPIO，他的出現最主要是為了取代不直觀的單晶片控制器(例如5081)。
- Arduino為微控制器，能透過I/O連接多種感測器，透過程式設計者的創意發想，就是當下討論最熱門的物聯網(IoT)。



Arduino 介紹 (2/2)

ENTRY LEVEL	ARDUINO UNO	ARDUINO 101	ARDUINO PRO	ARDUINO PRO MINI	ARDUINO MICRO
	ARDUINO NANO	ARDUINO STARTER KIT	ARDUINO BASIC KIT	ARDUINO MOTOR SHIELD	
ENHANCED FEATURES	ARDUINO MEGA	ARDUINO ZERO	ARDUINO DUE	ARDUINO PROTO SHIELD	
INTERNET OF THINGS	ARDUINO YÚN	ARDUINO MKR1000	ARDUINO ETHERNET SHIELD	ARDUINO GSM SHIELD	
	ARDUINO WIFI SHIELD 101				
WEARABLE	ARDUINO GEMMA	LILYPAD ARDUINO USB	LILYPAD ARDUINO MAIN BOARD		
	LILYPAD ARDUINO SIMPLE	LILYPAD ARDUINO SIMPLE SNAP			
3D PRINTING	MATERIA 101				

BOARDS

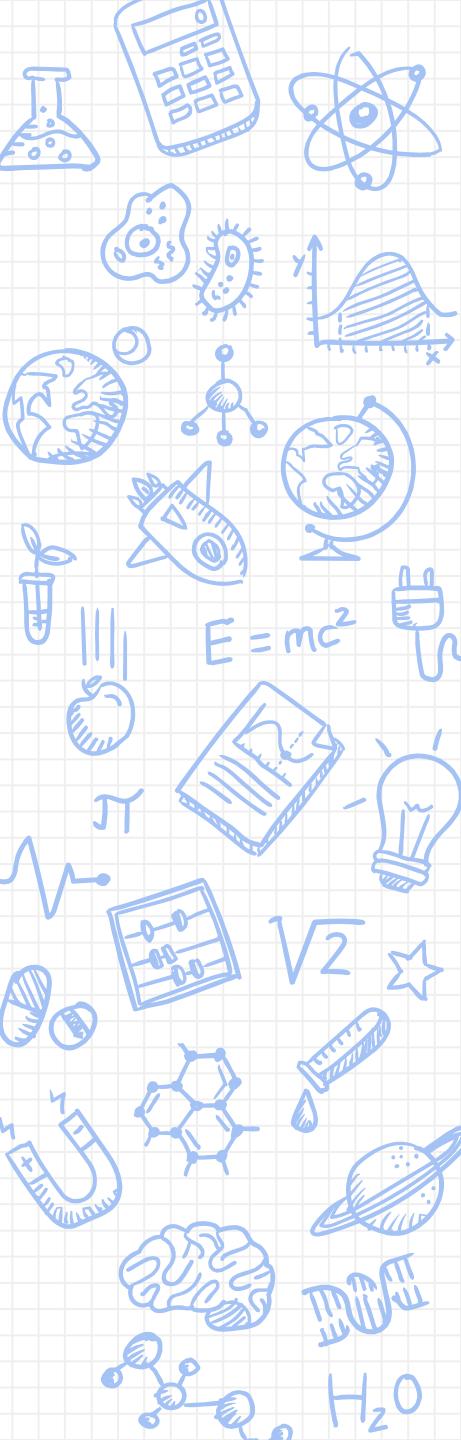
MODULES

SHIELDS

KITS

ACCESSORIES

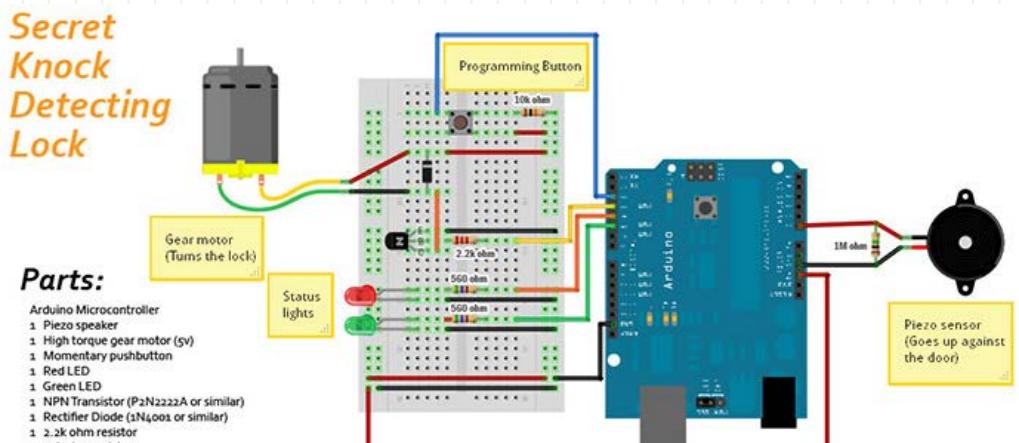
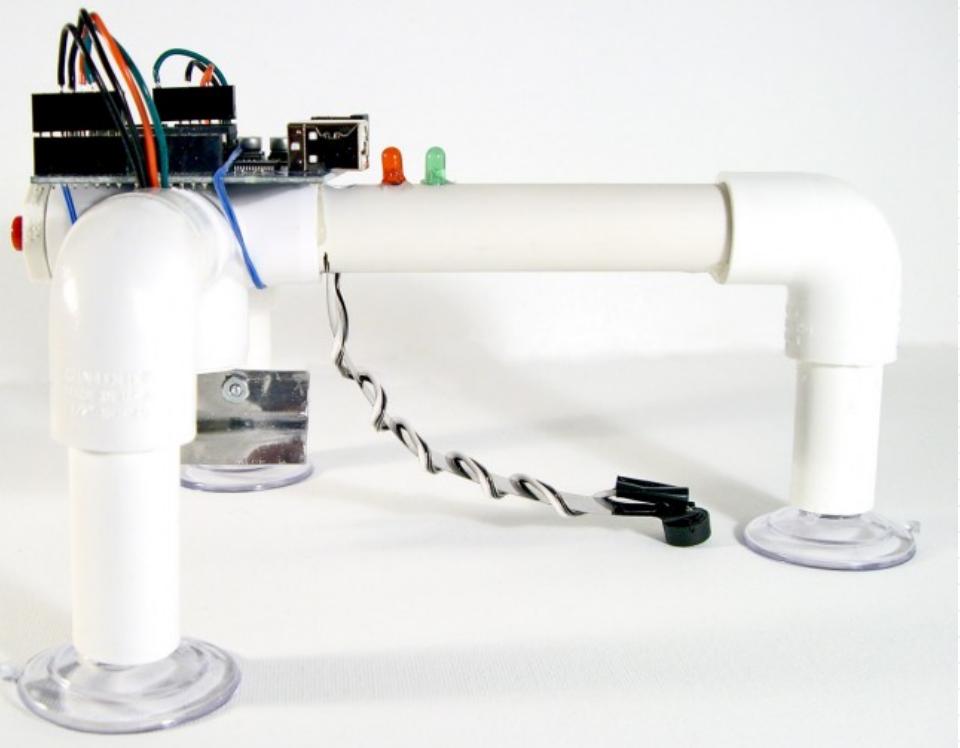
COMING NEXT



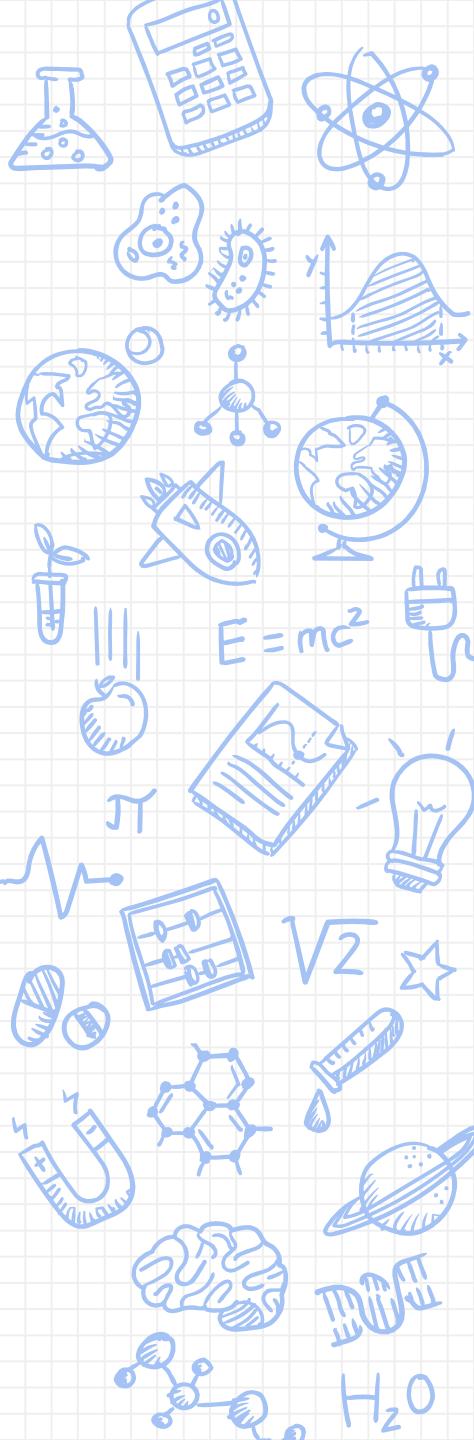
Arduino 作品 (1/2)

- Secret Knock Detecting Door Lock

http://grathio.com/2009/11/secret_knock_detecting_door_lock/

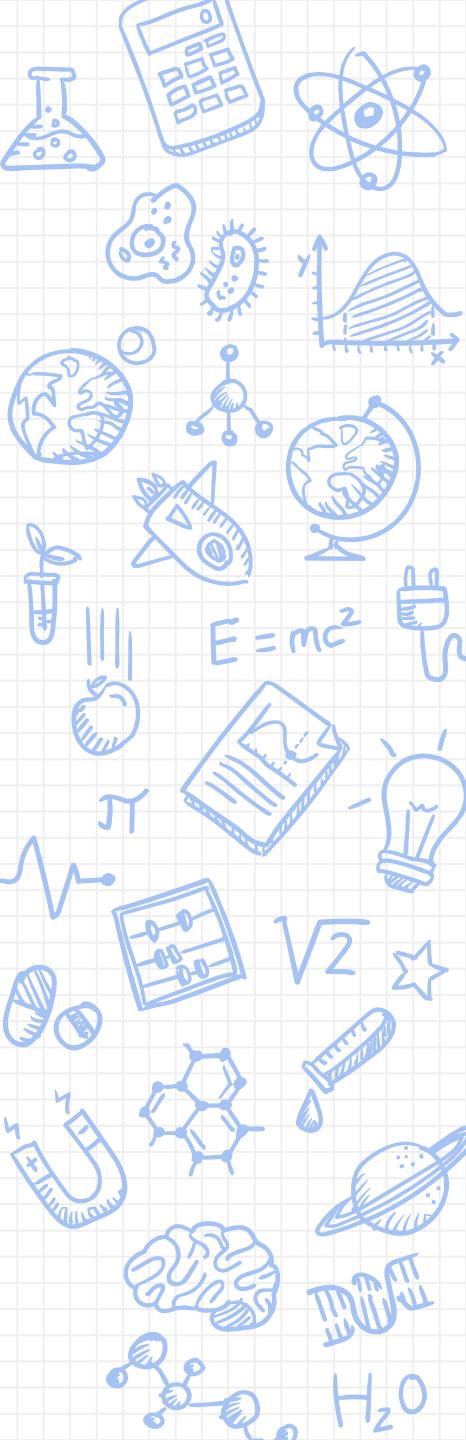


Made with Fritzing <http://fritzing.org>



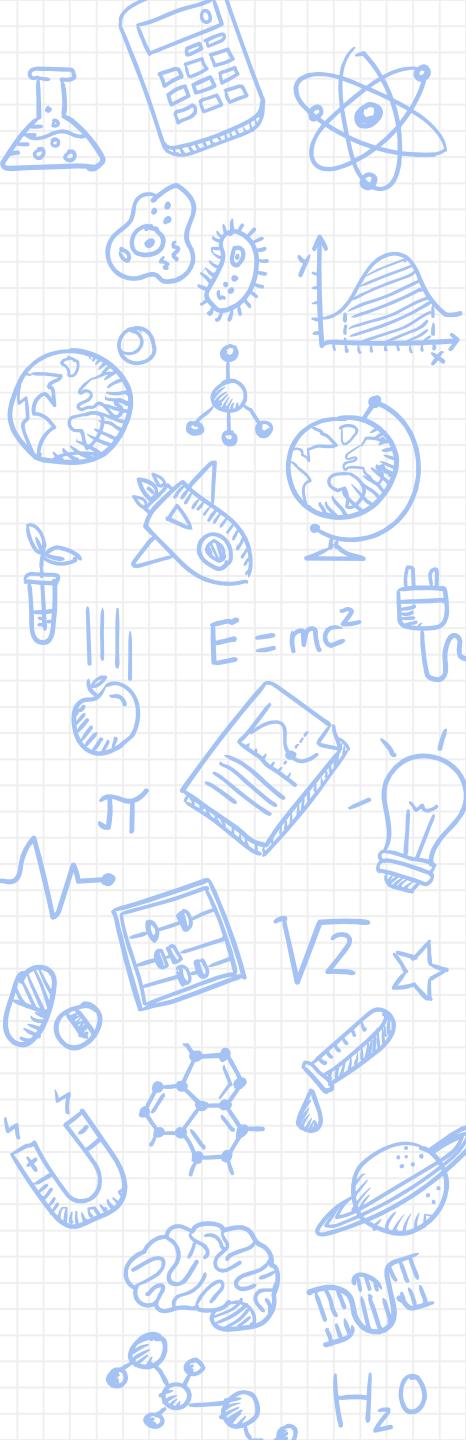
Arduino 作品 (2/2)

- 寵物餵食器



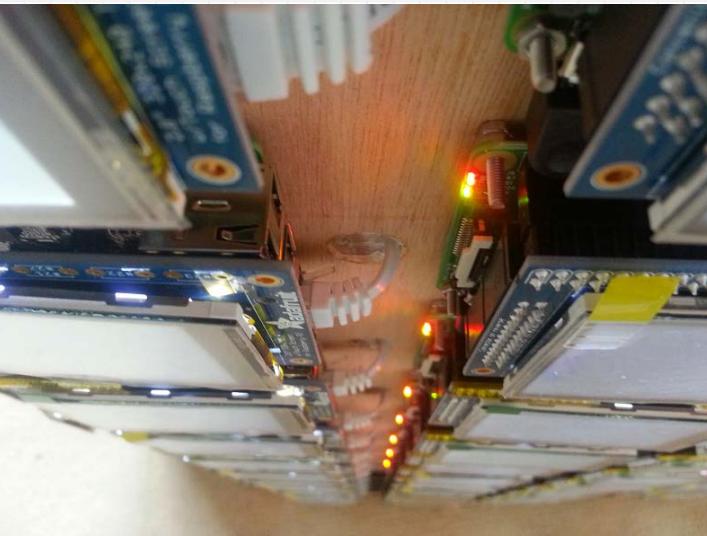
Raspberry Pi 介紹

- Raspberry Pi，是以Linux作為底層作業系統的微型電腦，與Arduino最大不同的地方在於，Arduino僅為單晶片控制器。
- 最初Raspberry Pi是設計給小學生接觸程式設計的第一步，以省電與價格低廉作為最主要優勢而聞名。爾後才有人將大量的Raspberry Pi串聯，做為高密度叢集電腦執行多核心運算。

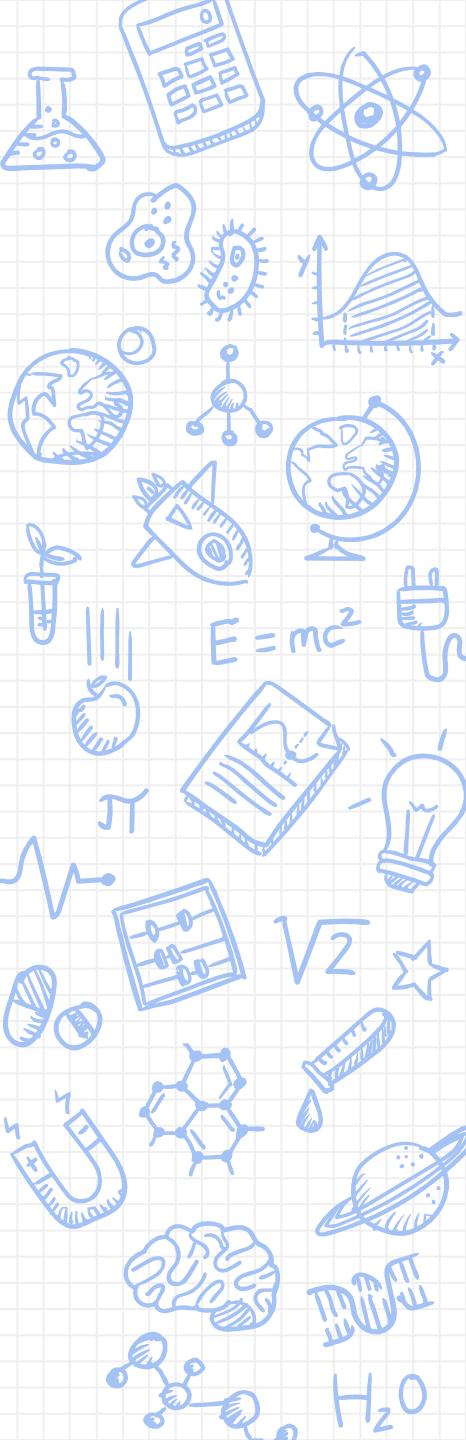


Raspberry Pi 作品 (1/2)

- 120-Raspberry Pi Cluster



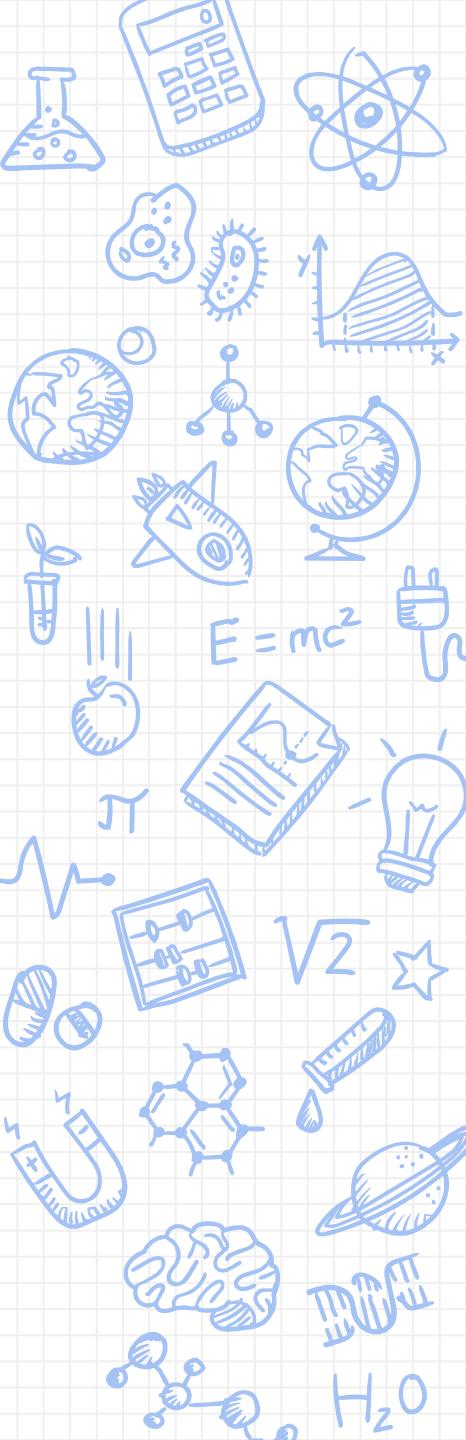
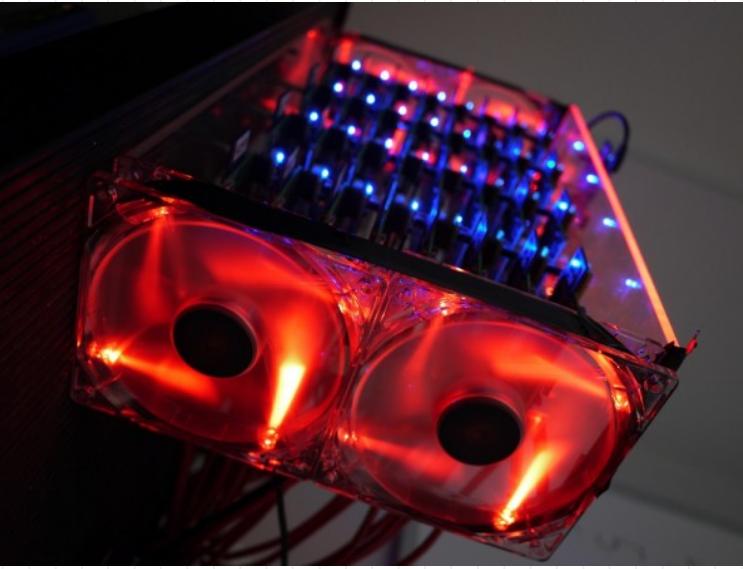
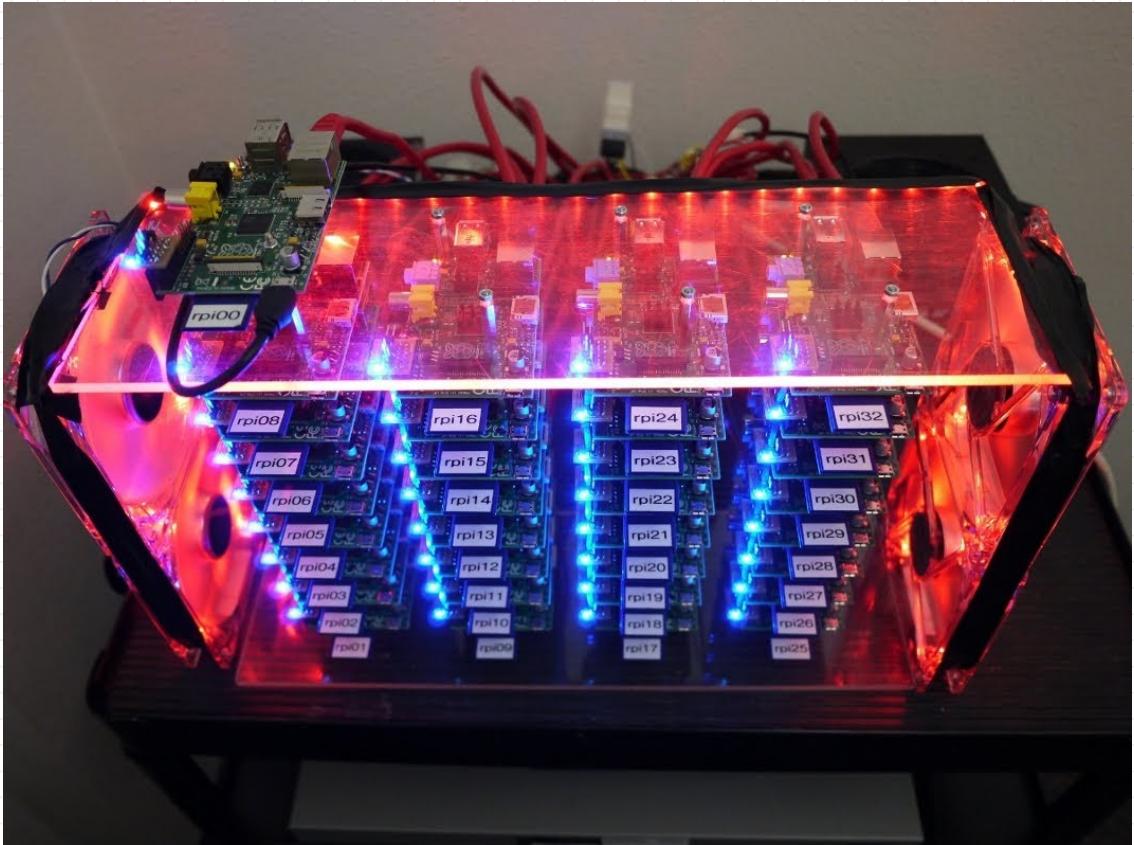
10



Raspberry Pi 作品 (2/2)

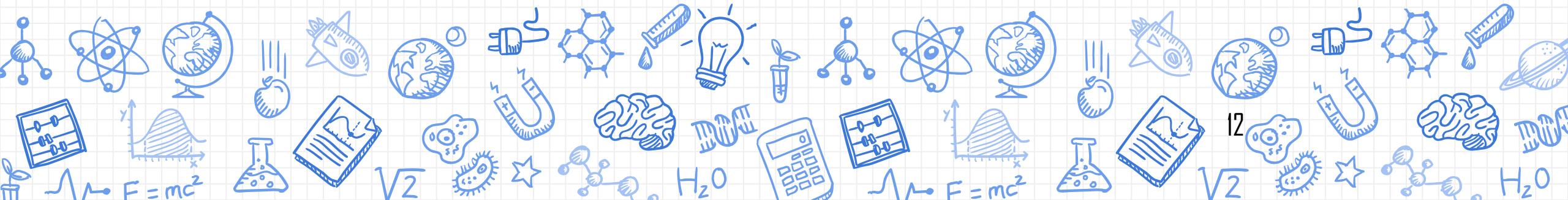
- Raspberry Pi 挖彼特幣

https://www.youtube.com/watch?v=i_r3z1jYHAc



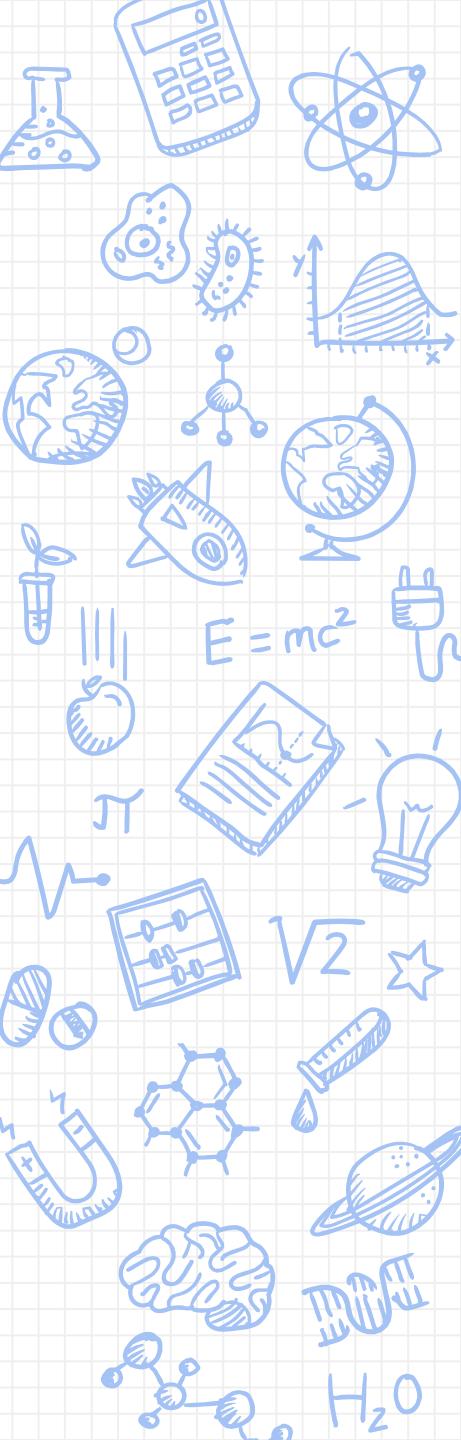
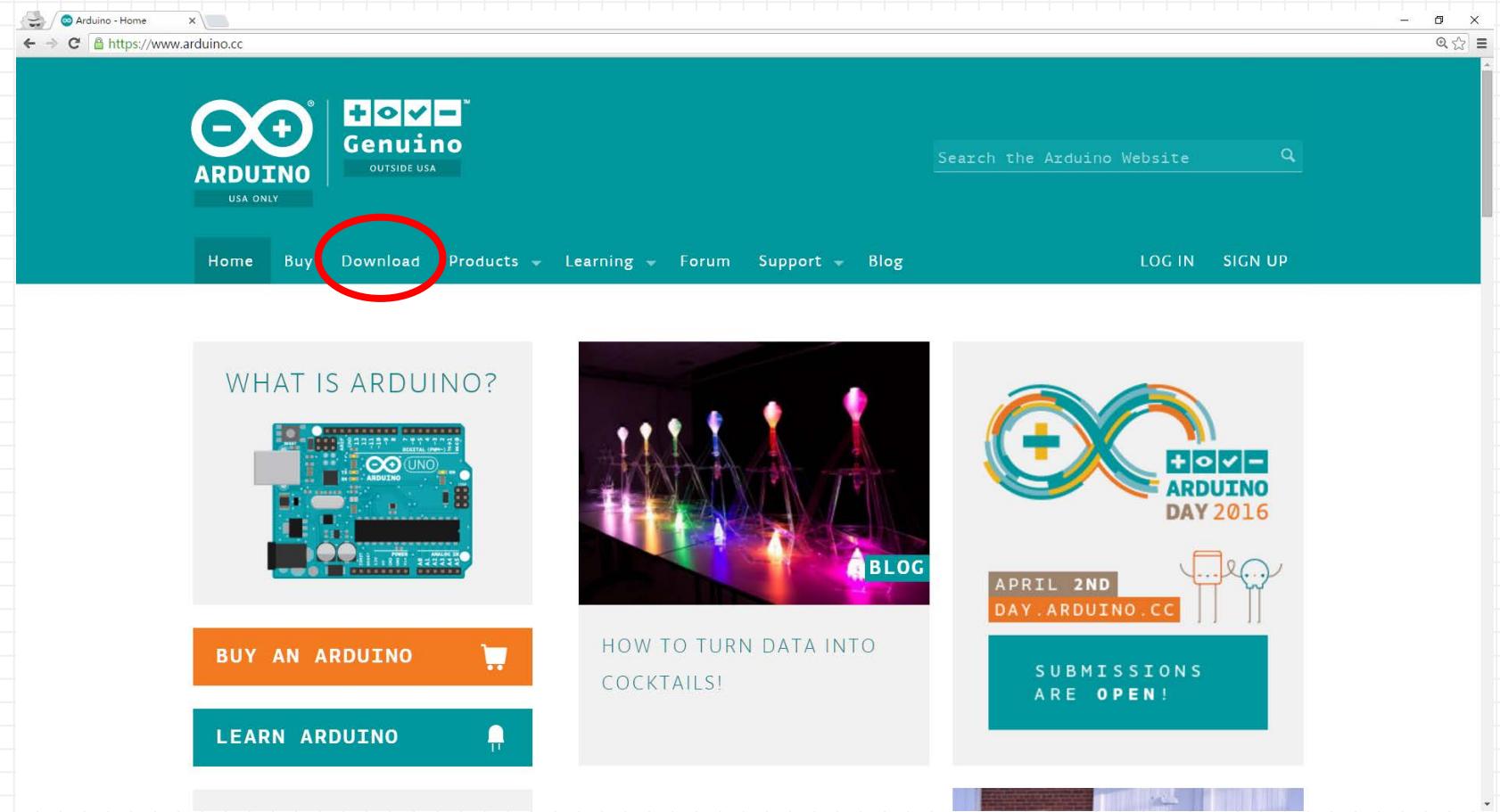
Outline

- 開放硬體介紹
- Arduino 環境架設
- Scratch For Arduino (S4A)
- 自由發揮小專題



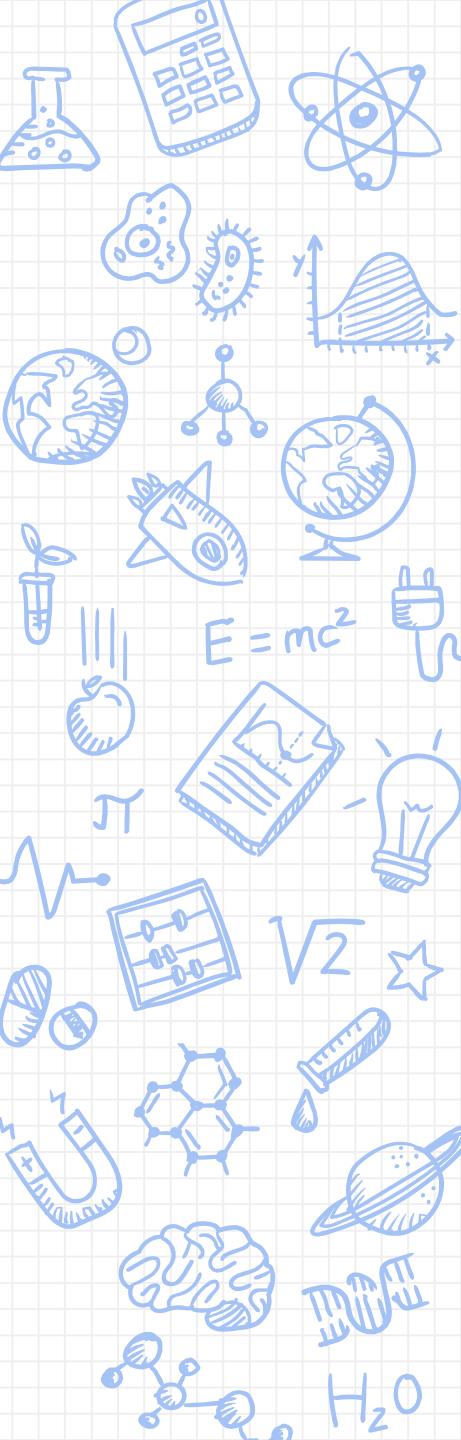
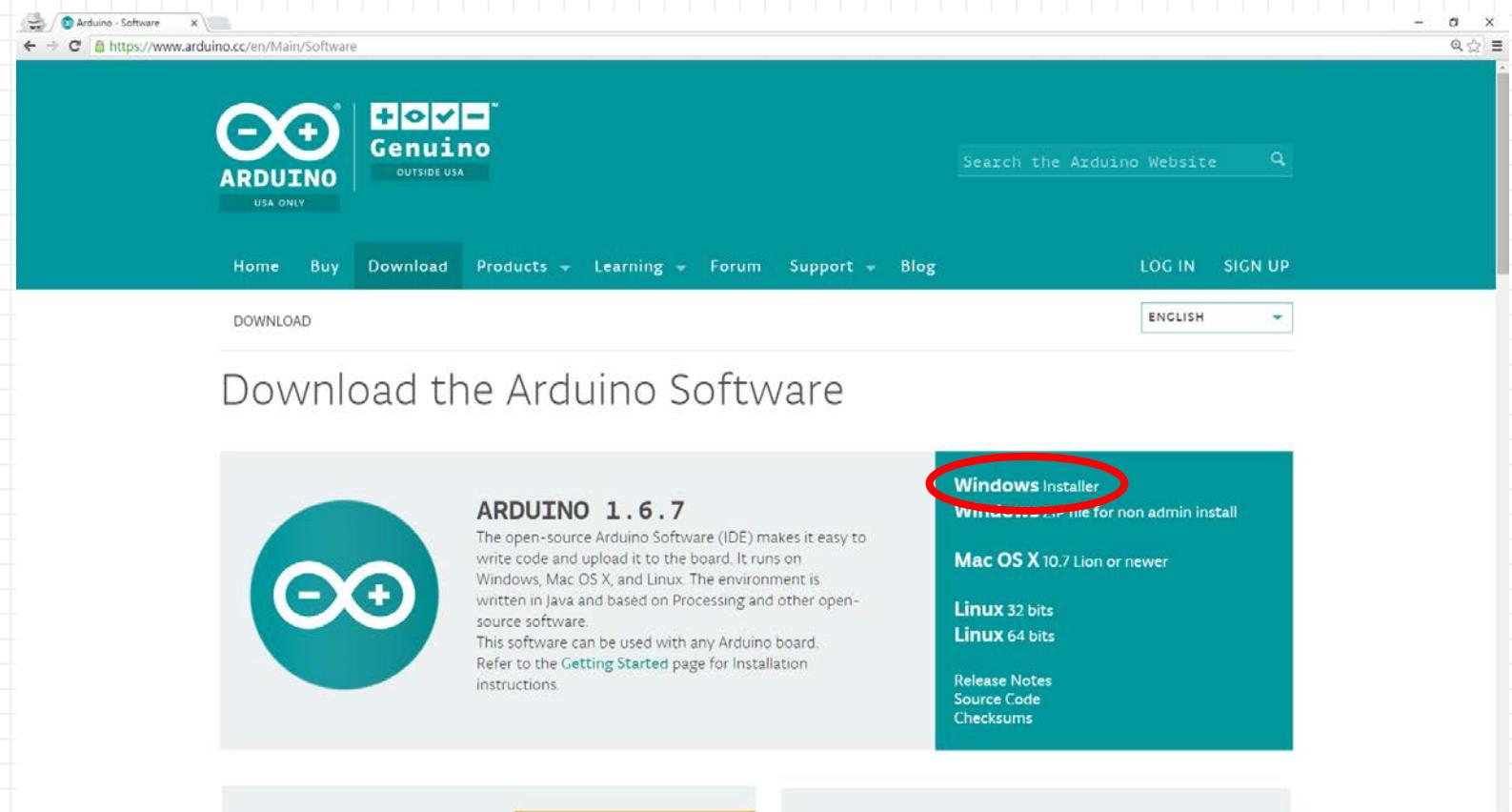
Arduino 環境架設(1/10)

- 從 <https://www.arduino.cc/> 下載 IDE



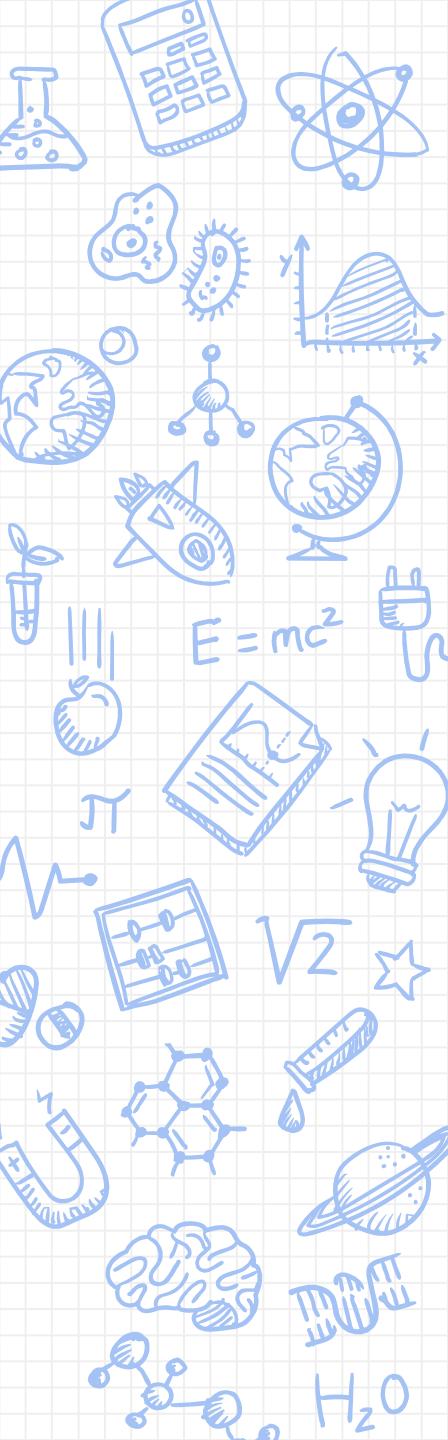
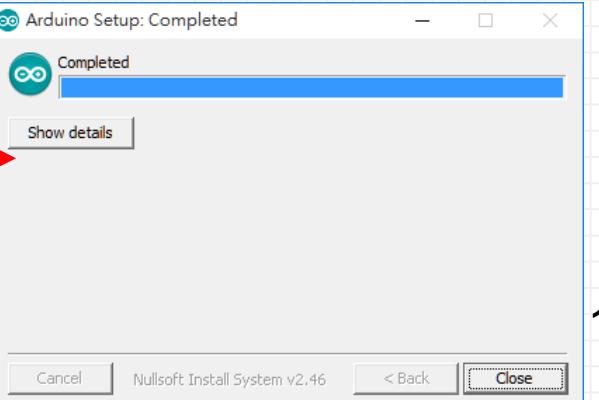
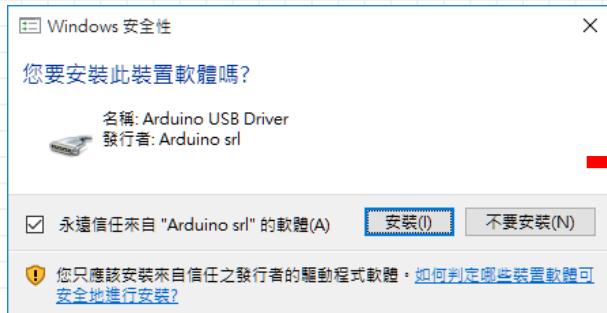
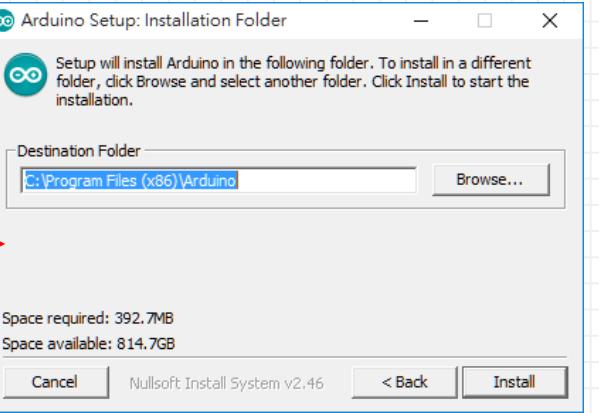
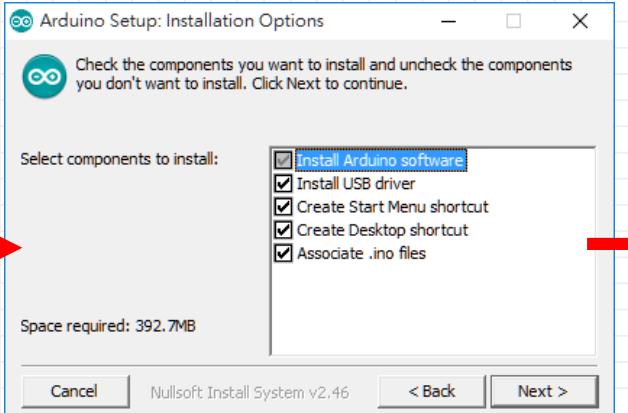
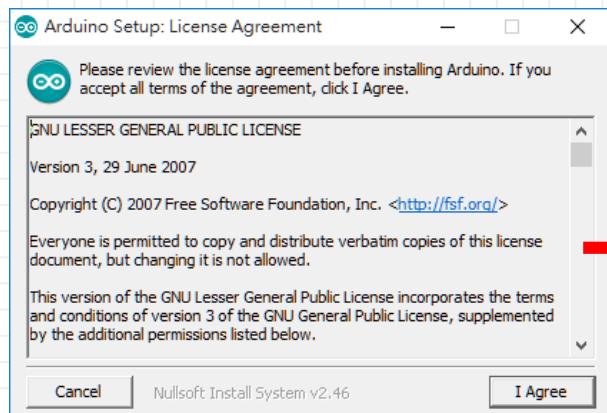
Arduino 環境架設(2/10)

- 此投影片下載版本為Arduino 1.6.7，於Windows作業系統執行。



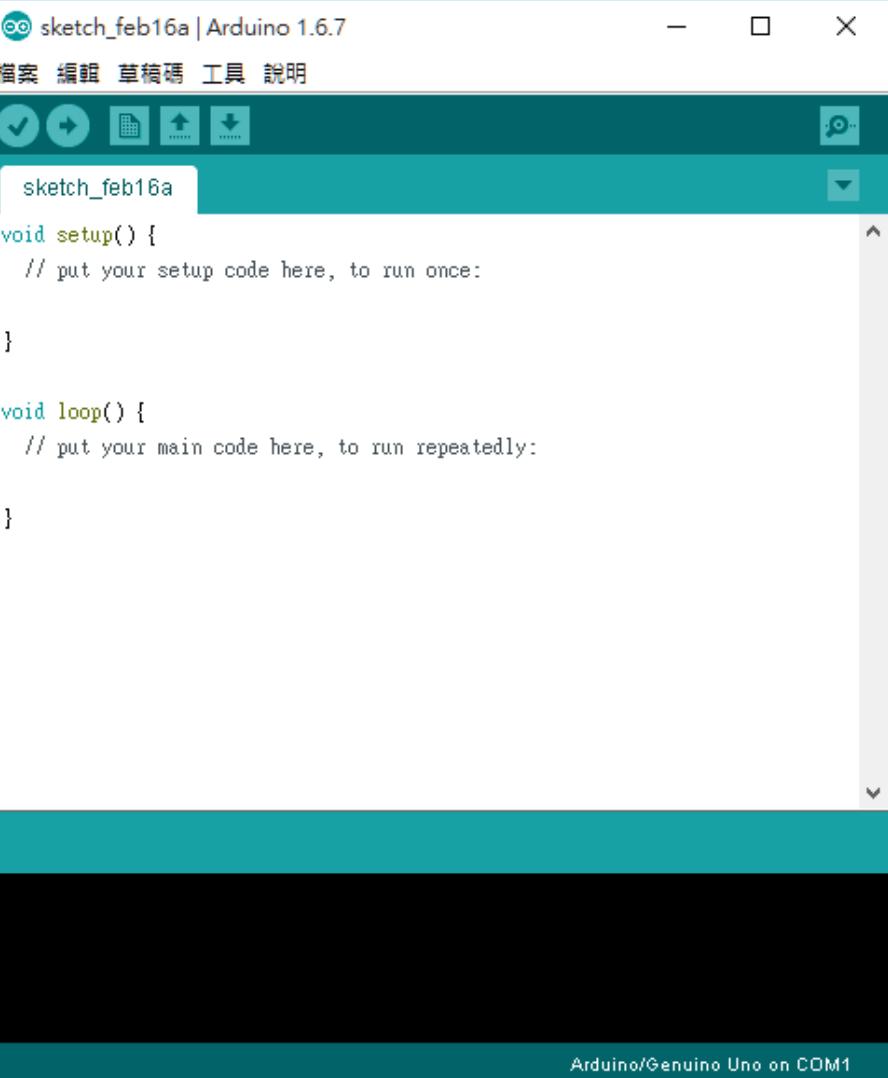
Arduino 環境架設(3/10)

- 執行下載程式，並順勢安裝USB驅動程式



Arduino 環境架設(4/10)

- 開啟Arduino IDE。
- `setup()`是當程式被啟動時，第一個執行的函式，即為初始化的概念。例如：初始化I/O。
- `loop()`是一個會反覆被執行的函式，當程式初始化完成之後，將會進入`loop`迴圈無限輪迴，執行使用者寫入的程式，大多是以FSM的方式去實現。
- 透過下頁例子可以更加瞭解運作原理。

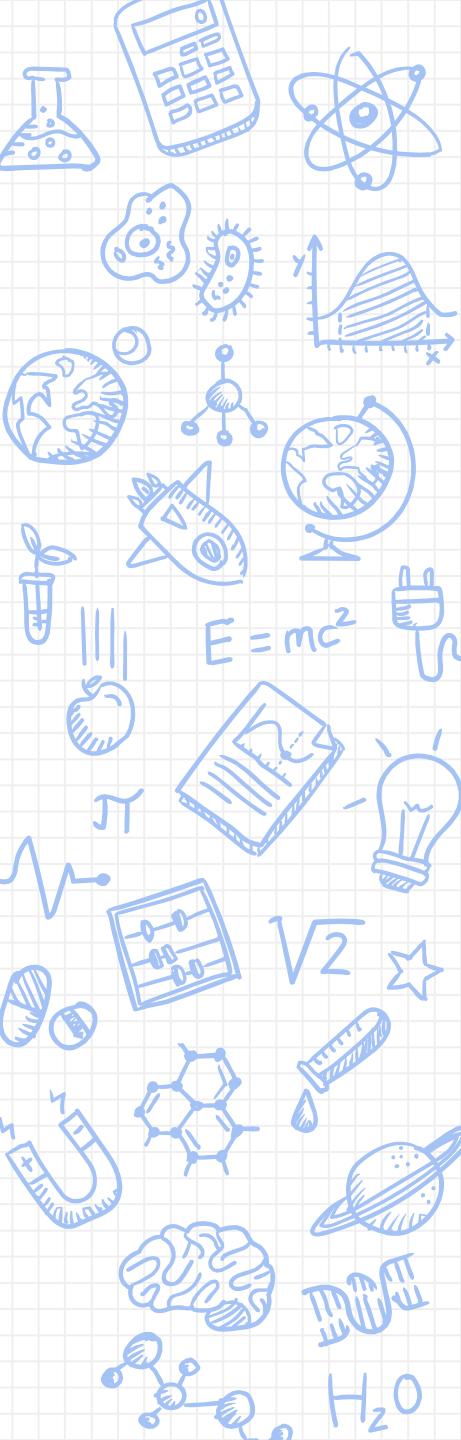


The screenshot shows the Arduino IDE interface with the title bar "sketch_feb16a | Arduino 1.6.7". The menu bar includes "檔案", "編輯", "草稿碼", "工具", and "說明". The main code area contains the following:

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

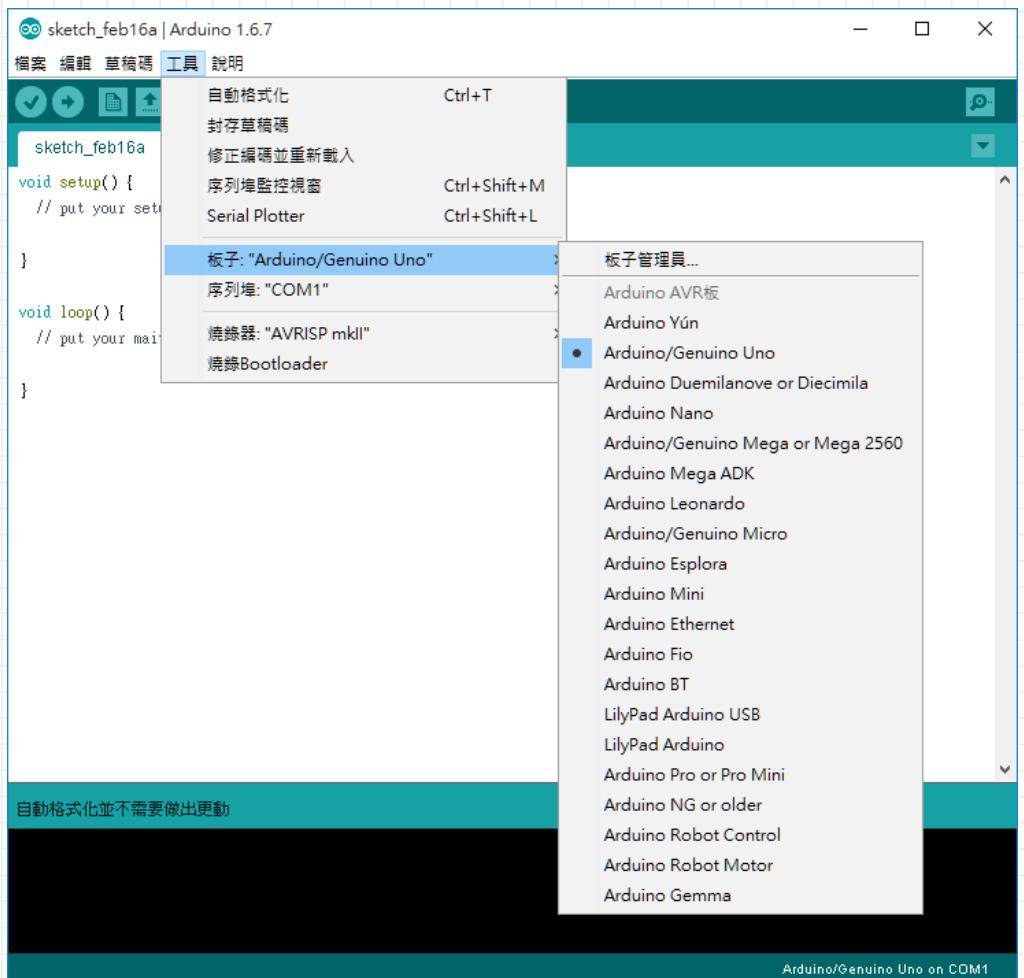
void loop() {
  // put your main code here, to run repeatedly:
}
```

At the bottom of the screen, it says "Arduino/Genuino Uno on COM1".



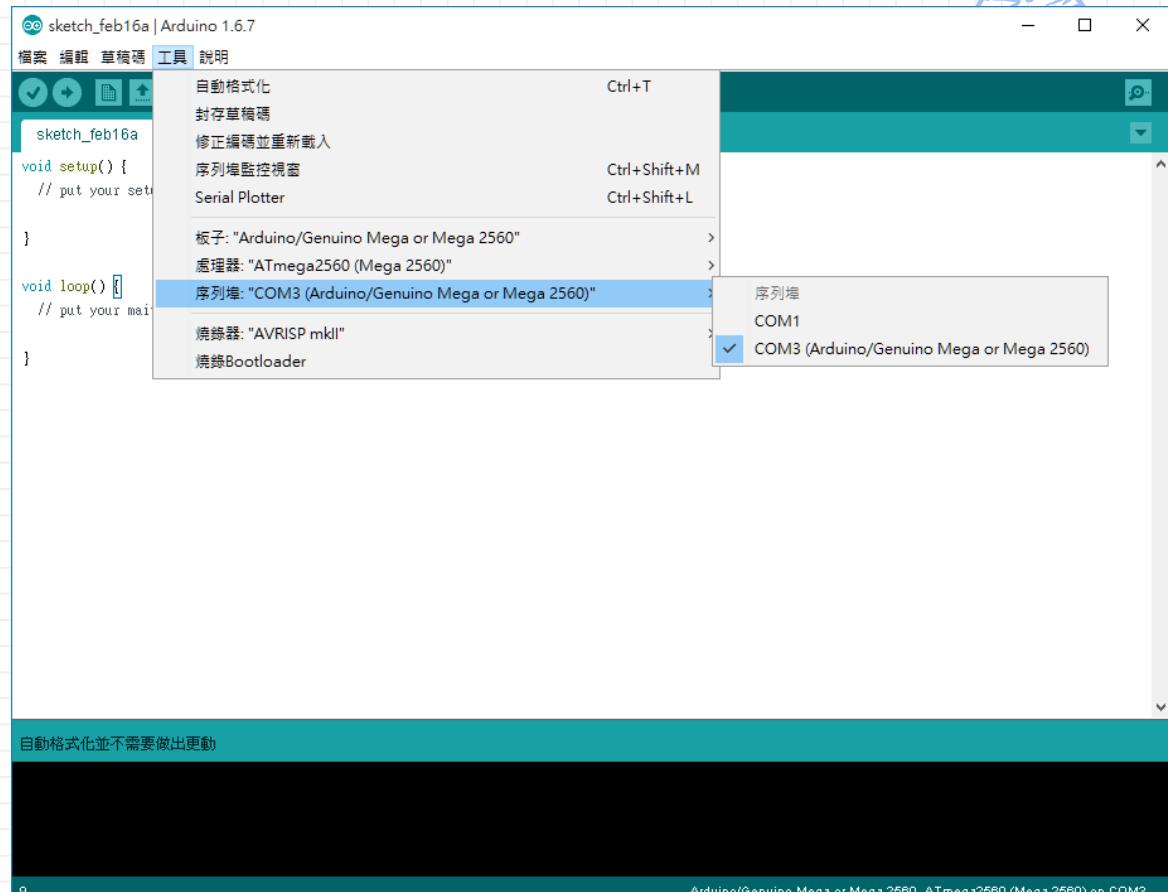
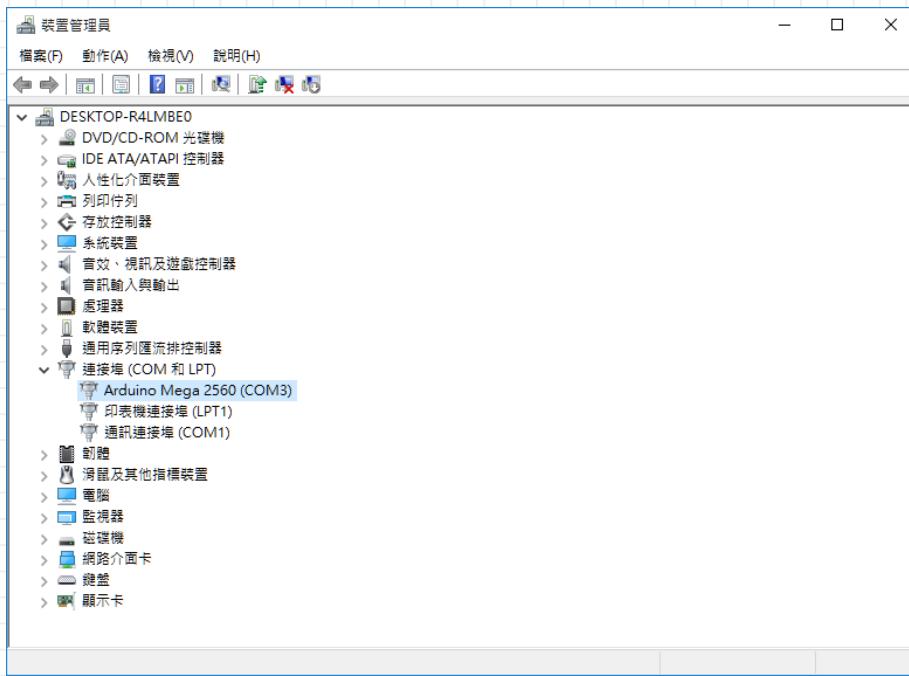
Arduino 環境架設(5/10)

- 燒錄第一個範例程式
 - 選取正確的Arduino型號



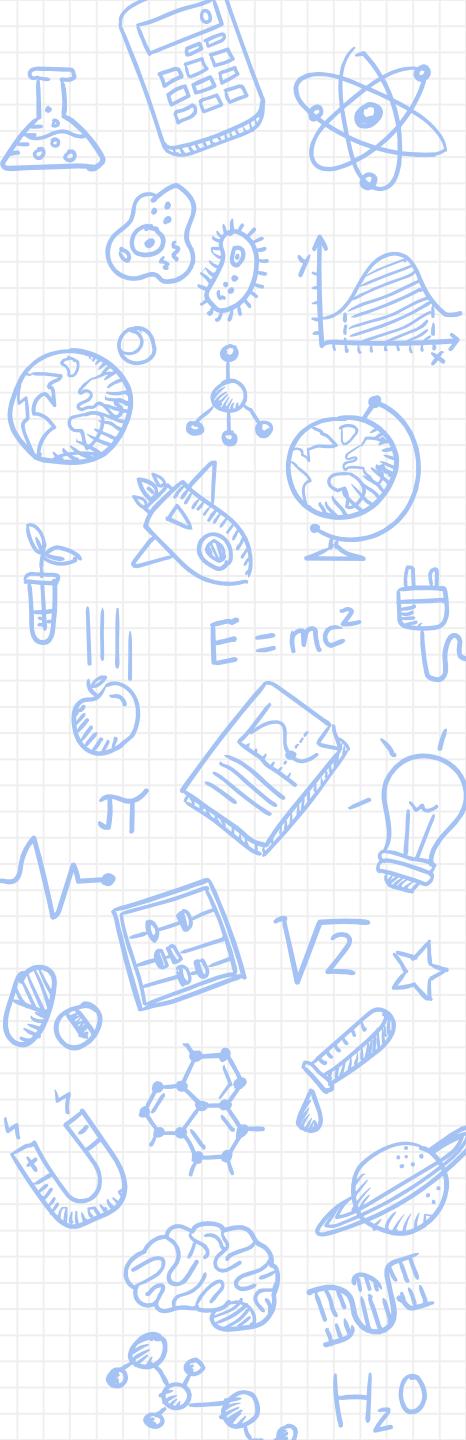
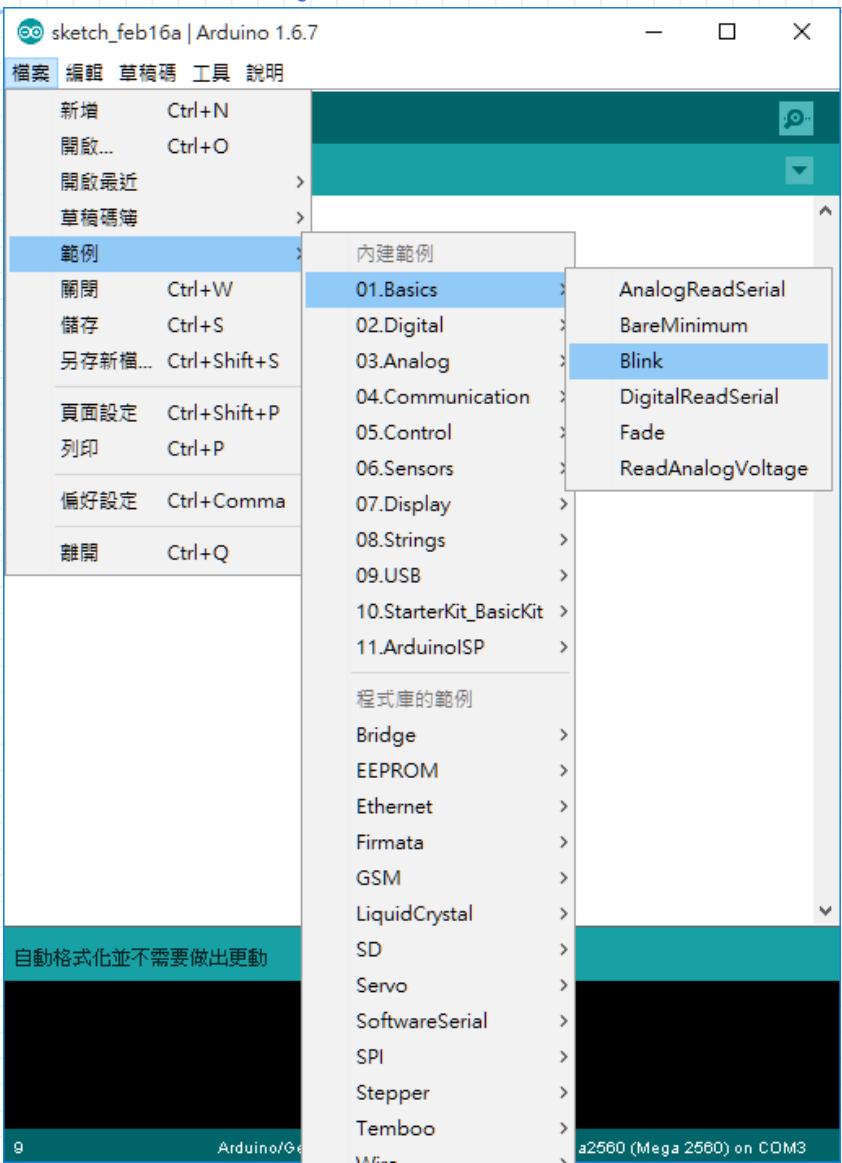
Arduino 環境架設(6/10)

- 選擇正確的COM port。
- 若電腦接上多種器材，則必須要從裝置管理員中確定正確的COM port。



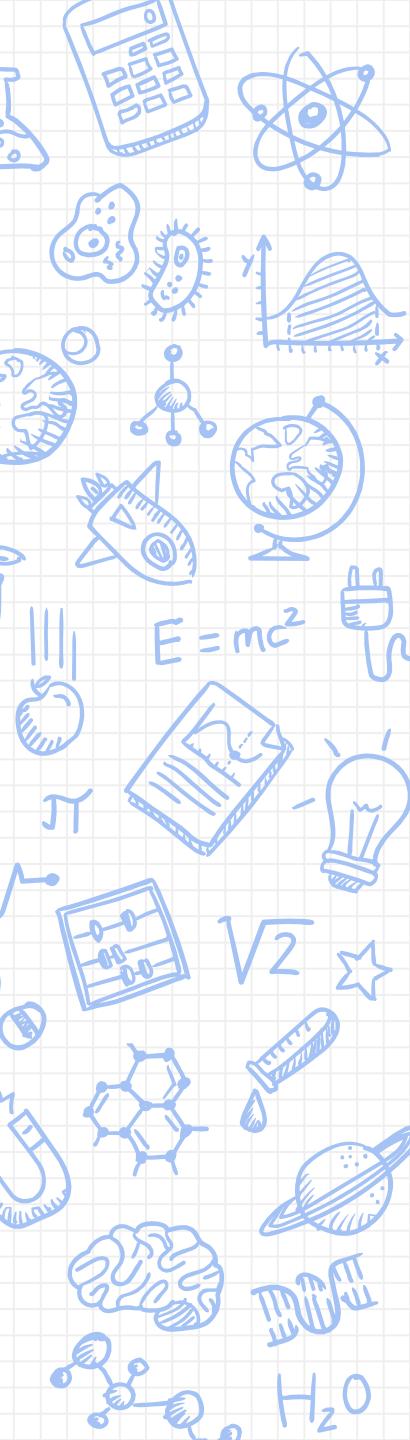
Arduino 環境架設(7/10)

- 開啟範例程式。
- 此投影片以Blink作為教學範例。



Arduino 環境架設

- `setup()`中初始化I/O。
- `pinMode(13, OUTPUT)`，意即將編號13的I/O設定為輸出使用。
- `loop()`反覆執行四行程式。
- `digitalWrite(13, HIGH)`，寫入編號13的輸出為高準位電壓。
- `Delay(1000)`，程式暫停1000 ms。



A screenshot of the Arduino IDE showing the 'Blink' example sketch. The interface includes a toolbar with file, edit, and upload buttons, and a menu bar with Chinese labels: 檔案 (File), 編輯 (Edit), 草稿碼 (Sketch), 工具 (Tools), and 說明 (Help). The code editor displays the 'Blink' sketch with its comments and logic.

```
/*
 * Blink
 *
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://www.arduino.cc
 *
 * This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

Arduino 環境架設

- 確認設定無誤後，點選上傳按鈕，此程式將自動編譯後燒到板子上。
- 若想測試程式是否有Bug，可以點選上傳鈕的左邊一個按鍵，驗證程式是否正確被編譯。
- 板子上LED會間隔一秒閃爍，是因為此LED已經預設接在第13個IO上，故無需另外接LED驗證。

Blink | Arduino 1.6.7

檔案 編輯 草稿碼 工具 說明

上傳

Blink

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://www.arduino.cc

This example code is in the public domain.

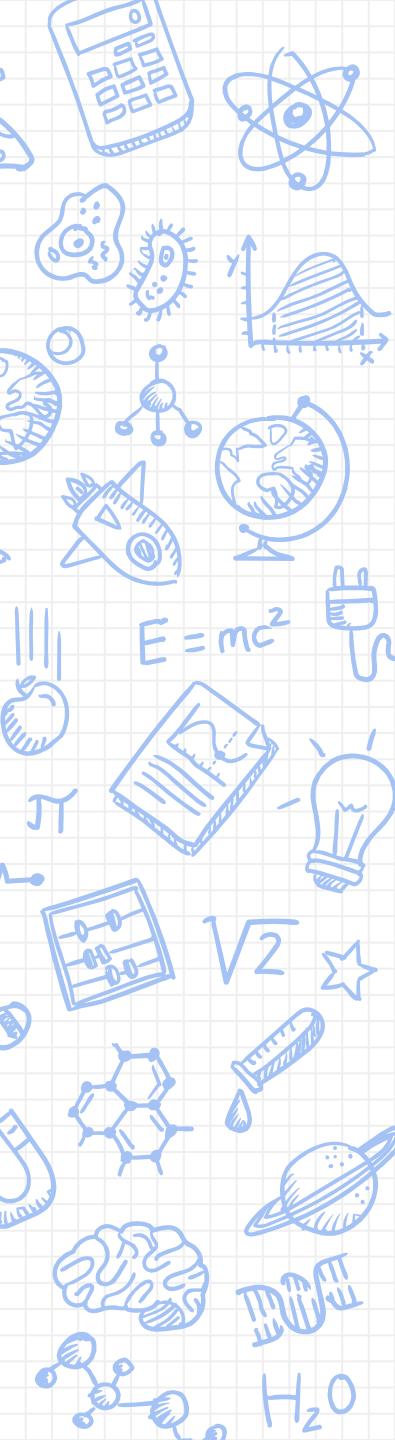
modified 8 May 2014
by Scott Fitzgerald
*/
```

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);              // wait for a second
    digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);              // wait for a second
}
```

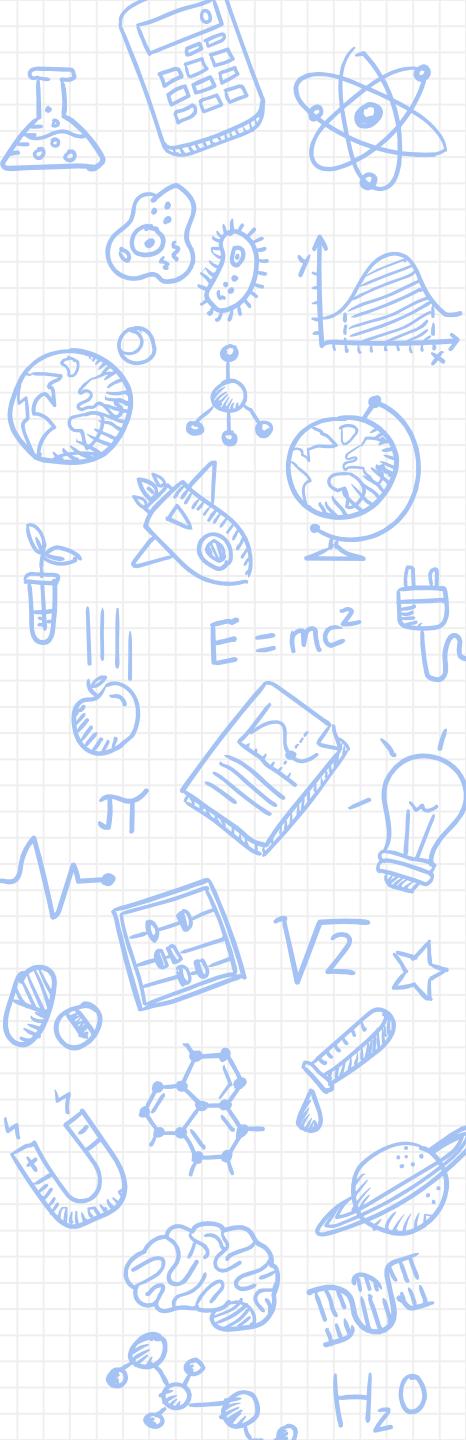
上傳完畢。

草稿碼使用了 1,518 bytes (0%) 的程式存儲空間。最大值為 253,952 bytes。
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 8,183 bytes 供局部變數。最大值為 8,192 bytes。



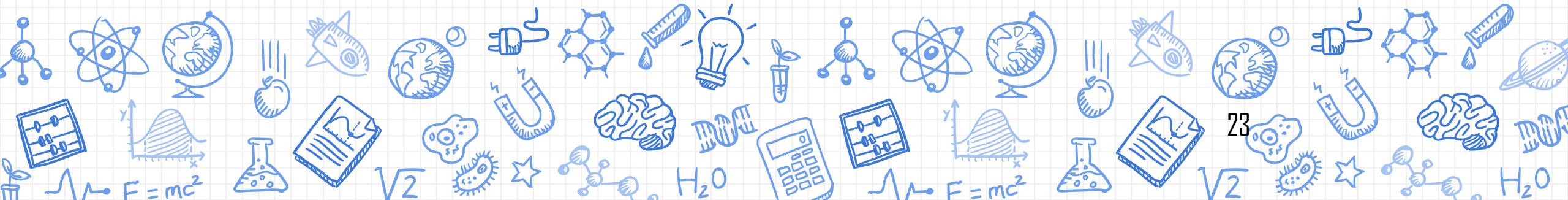
Arduino 環境架設 (10/10)

- Demo 1：使用按鈕(Button)，控制LED的明暗。
Button按下，LED亮。
Button放開，LED滅。
- 進階題：
Button按下，LED馬上亮起。
Button放開，LED快速閃爍5秒後才熄滅。



Outline

- 開放硬體介紹
- Arduino 環境架設
- Scratch For Arduino (S4A)
- 自由發揮小專題



Scratch For Arduino (1/10)

- 搜尋 <http://s4a.cat/> 並下載S4A。

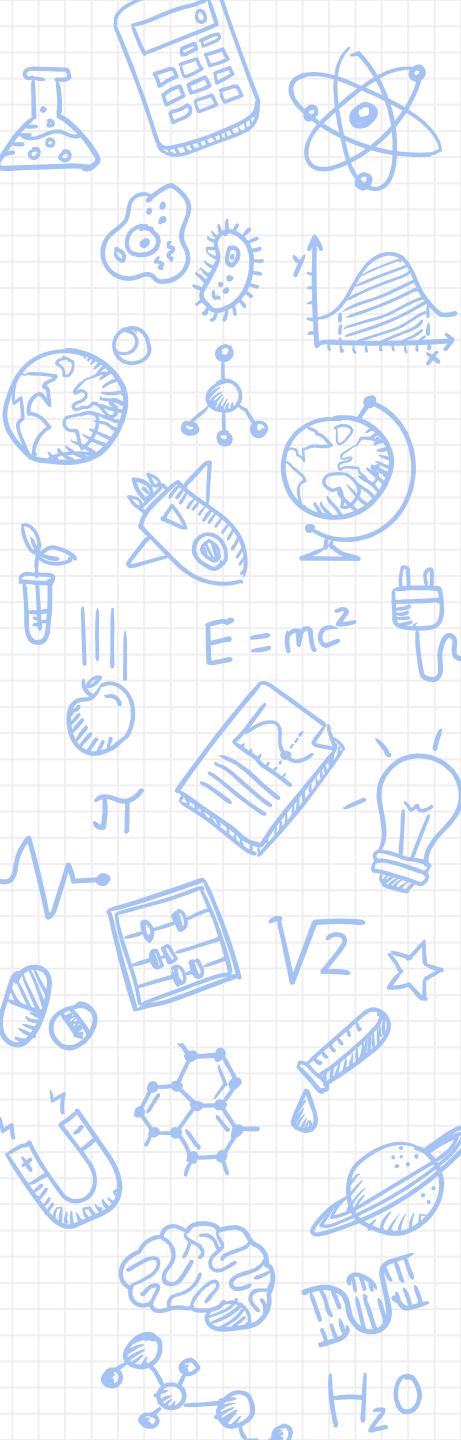
The screenshot shows the official website for S4A (Scratch For Arduino). The header features the S4A logo (two overlapping circles, orange and blue) and navigation links: About, Docs, Android, Changelog, [Downloads](#) (which is circled in red), Kit, FAQ, Team, Contact, Snap!, and Blog. A sidebar on the right contains a "Donate to S4A" section with a button to "donate any amount". The main content area includes sections for "About S4A" and "The interface".

About S4A

S4A is a [Scratch](#) modification that allows for simple programming of the [Arduino](#) open source hardware platform. It provides new blocks for managing sensors and actuators connected to [Arduino](#). There is also a sensors report board similar to the PicoBoard one.

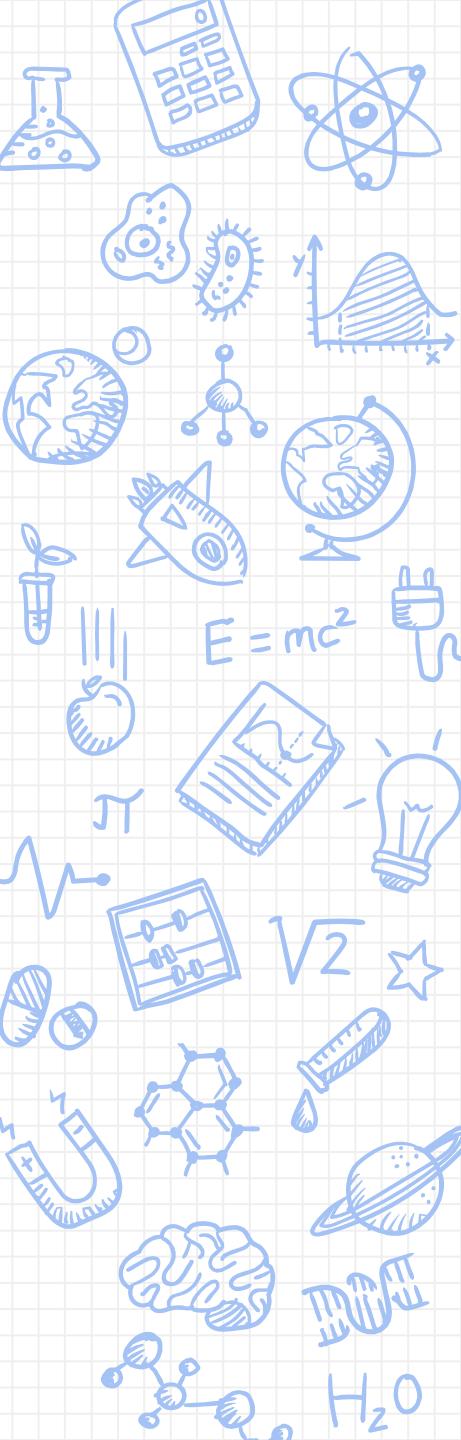
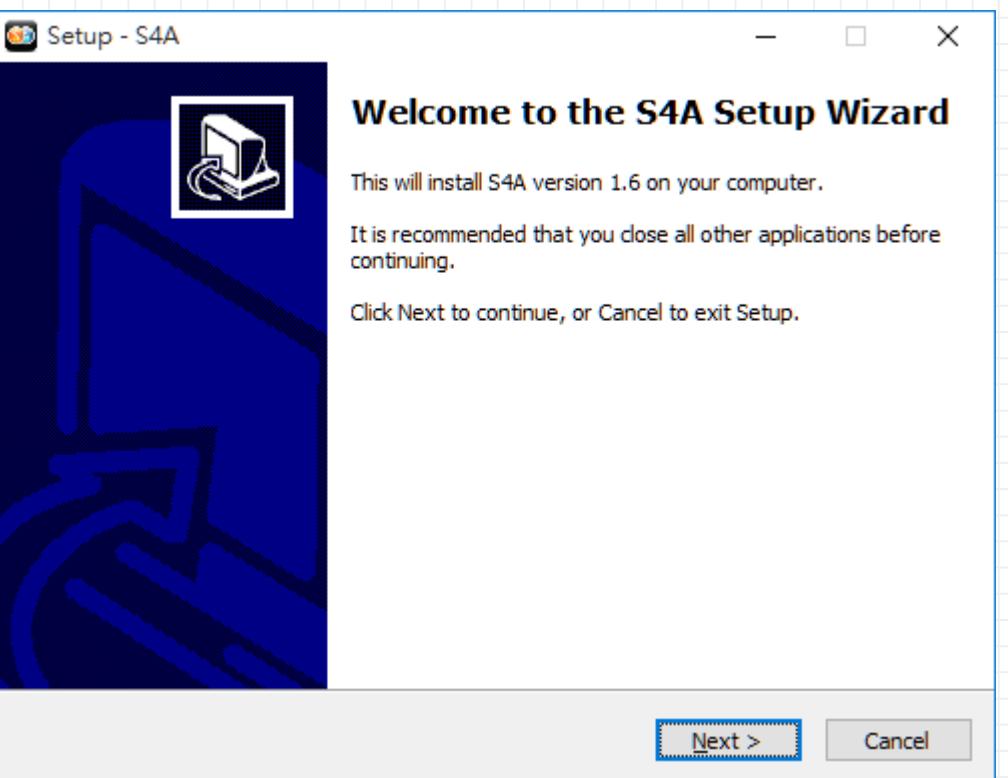
The main aim of the project is attracting people to the programming world. The goal is also to provide a high level interface to [Arduino](#) programmers with functionalities such as interacting with a set of boards through user events.

The interface



Scratch For Arduino (2/10)

- 進行安裝S4A。



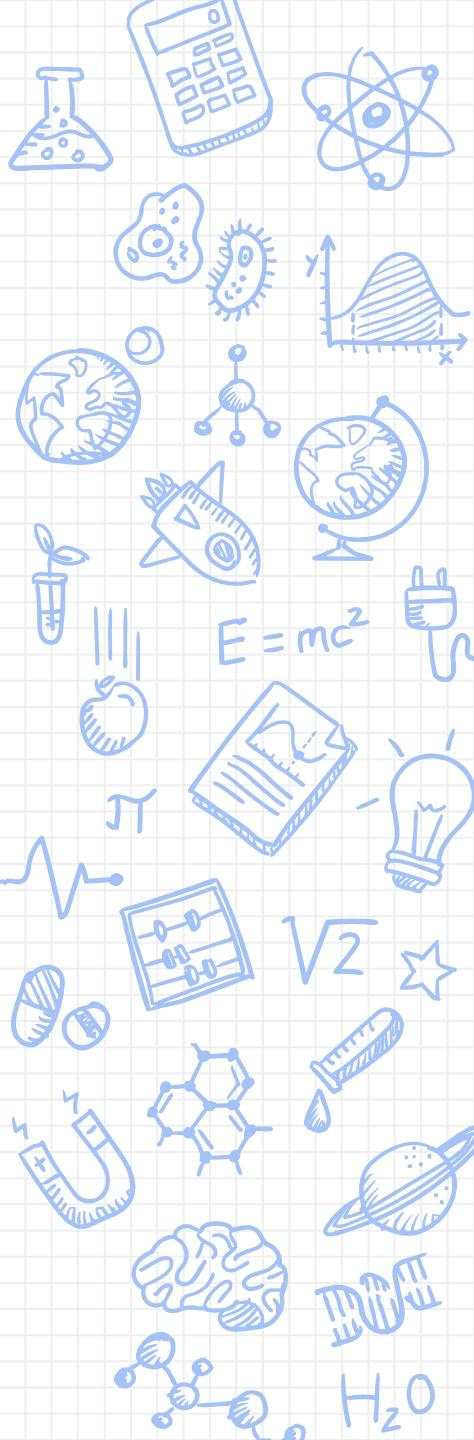
Scratch For Arduino (3/10)

- 在前頁下載頁面的下方，使用Arduino IDE把此段code燒進板子。

The screenshot shows the Arduino website's download section. It lists operating system options: Windows, Mac, Linux (Debian), Linux (Fedora) (version 1.5), and Raspbian (Debian for RaspberryPi) (version 1.5). Below this, under the heading "Installing the Firmware into your Arduino", it says: "This firmware is a piece of software you need to install into your Arduino board to be able to communicate with it from S4A." A bulleted list of steps follows:

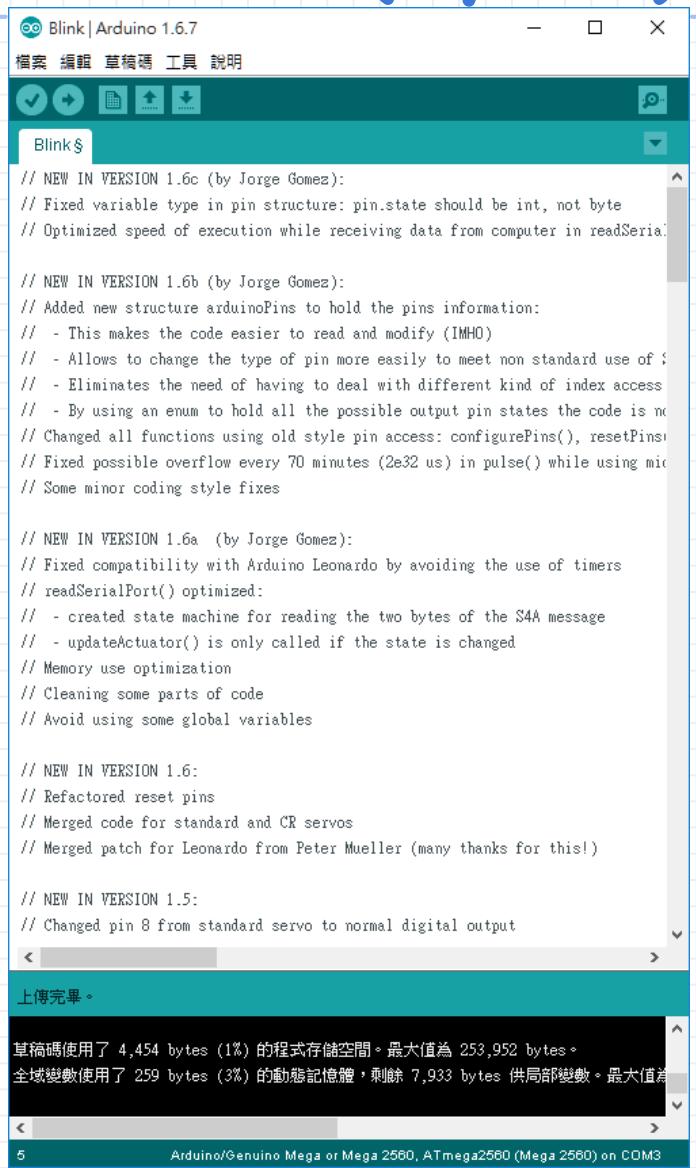
- Download and install the Arduino environment by following the instructions on <http://arduino.cc/en/Main/Software>. Take in account Arduino Uno requires at least version 0022.
- Download our firmware from [here](#) (this link is circled in red).
- Connect your Arduino board to a USB port in your computer.
- Open the firmware file (S4AFirmware16.ino) from the Arduino environment.
- In the Tools menu, select the board version and the serial port where the board is connected.
- Load the firmware into your board through File > Upload

The screenshot shows the Arduino IDE interface with the "Blink" sketch open. The code is displayed in the central editor area. At the bottom, a progress bar indicates the upload process: "上传完毕。" (Upload completed). Below the progress bar, status messages in Chinese are shown: "草稿碼使用了 4,454 bytes (1%) 的程式存儲空間。最大值為 253,952 bytes。" and "全局變數使用了 259 bytes (3%) 的動態記憶體，剩餘 7,933 bytes 供局部變數。最大值為 1,048,576 bytes。" The status bar at the bottom right shows "Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3".



Scratch For Arduino (4/10)

- 這一段code是S4A的韌體，也就是說Arduino要透過S4A來開發程式前，必須要透過燒錄此韌體才能使用。



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.7". The menu bar includes "檔案", "編輯", "草稿碼", "工具", and "說明". The toolbar has icons for file operations like Open, Save, and Print. The main code area displays the "Blink" sketch:

```
// NEW IN VERSION 1.6c (by Jorge Gomez):
// Fixed variable type in pin structure: pin.state should be int, not byte
// Optimized speed of execution while receiving data from computer in readSerial()

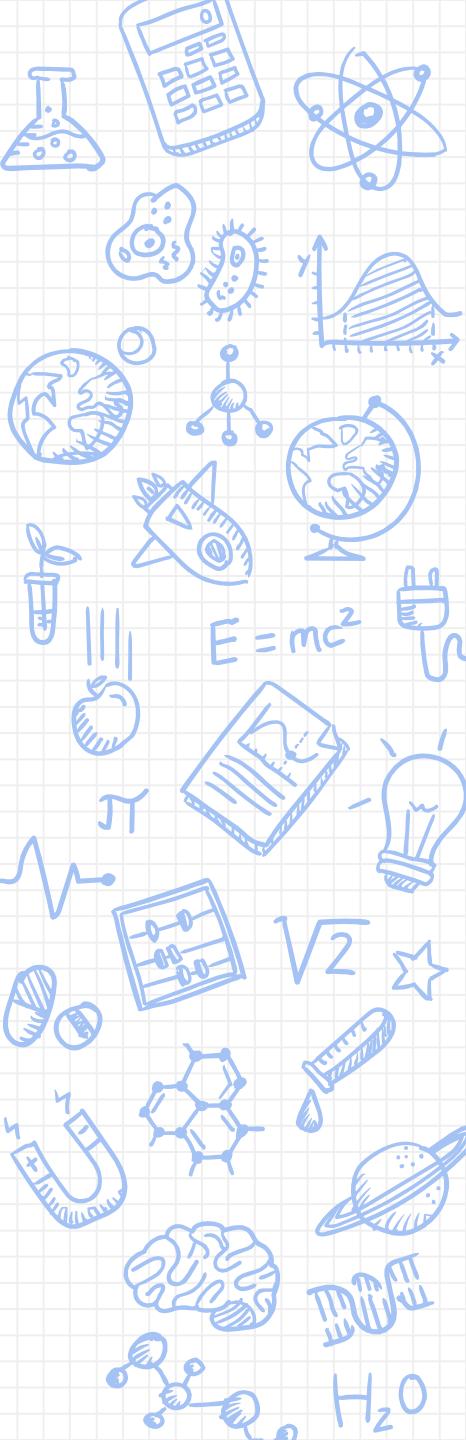
// NEW IN VERSION 1.6b (by Jorge Gomez):
// Added new structure arduinoPins to hold the pins information:
// - This makes the code easier to read and modify (IMHO)
// - Allows to change the type of pin more easily to meet non standard use of :
// - Eliminates the need of having to deal with different kind of index access
// - By using an enum to hold all the possible output pin states the code is more readable
// Changed all functions using old style pin access: configurePins(), resetPins()
// Fixed possible overflow every 70 minutes (2e32 us) in pulse() while using microsecond
// Some minor coding style fixes

// NEW IN VERSION 1.6a (by Jorge Gomez):
// Fixed compatibility with Arduino Leonardo by avoiding the use of timers
// readSerialPort() optimized:
// - created state machine for reading the two bytes of the S4A message
// - updateActuator() is only called if the state is changed
// Memory use optimization
// Cleaning some parts of code
// Avoid using some global variables

// NEW IN VERSION 1.6:
// Refactored reset pins
// Merged code for standard and CR servos
// Merged patch for Leonardo from Peter Mueller (many thanks for this!)

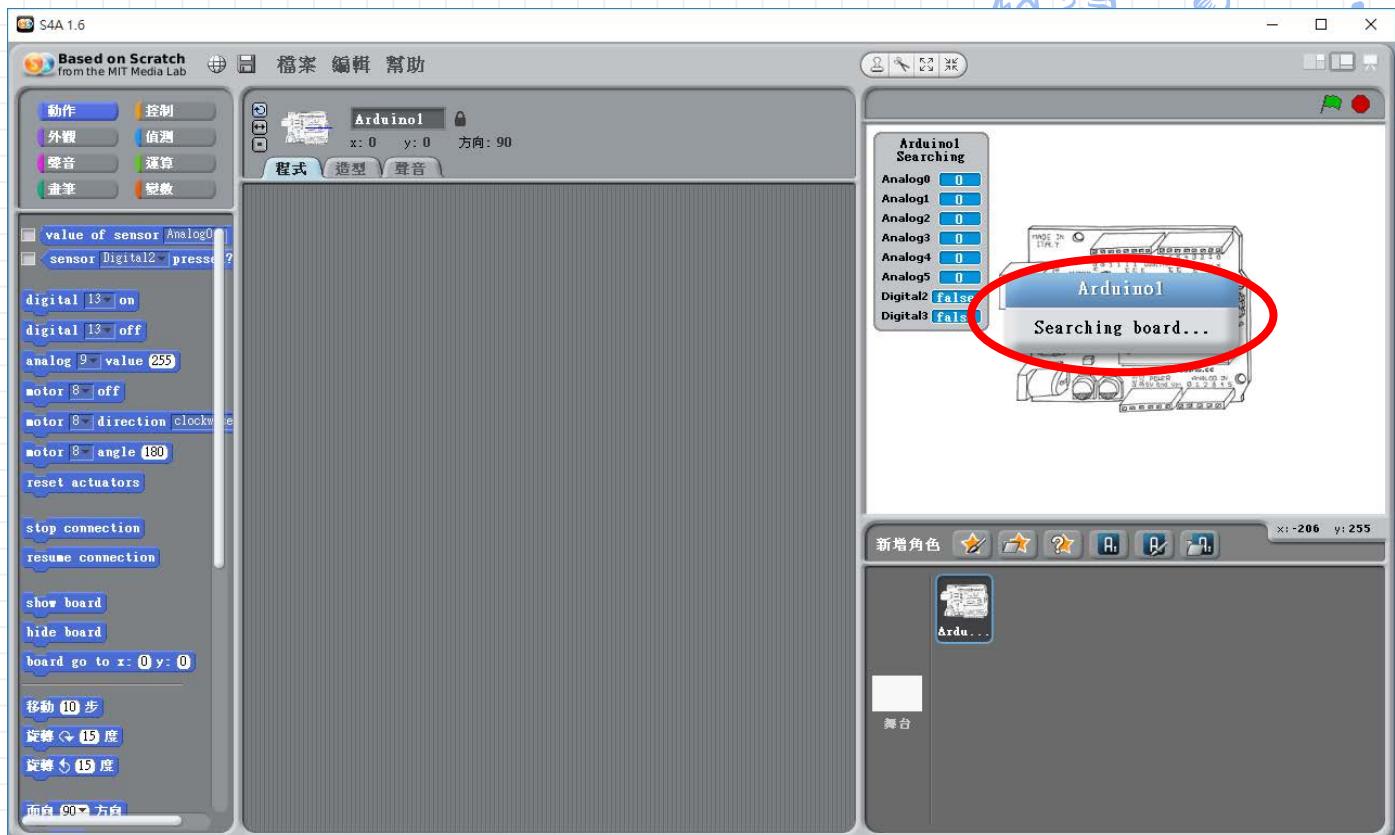
// NEW IN VERSION 1.5:
// Changed pin 8 from standard servo to normal digital output
```

At the bottom of the code area, it says "上傳完畢。" (Upload completed). The status bar at the bottom right indicates "Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3".



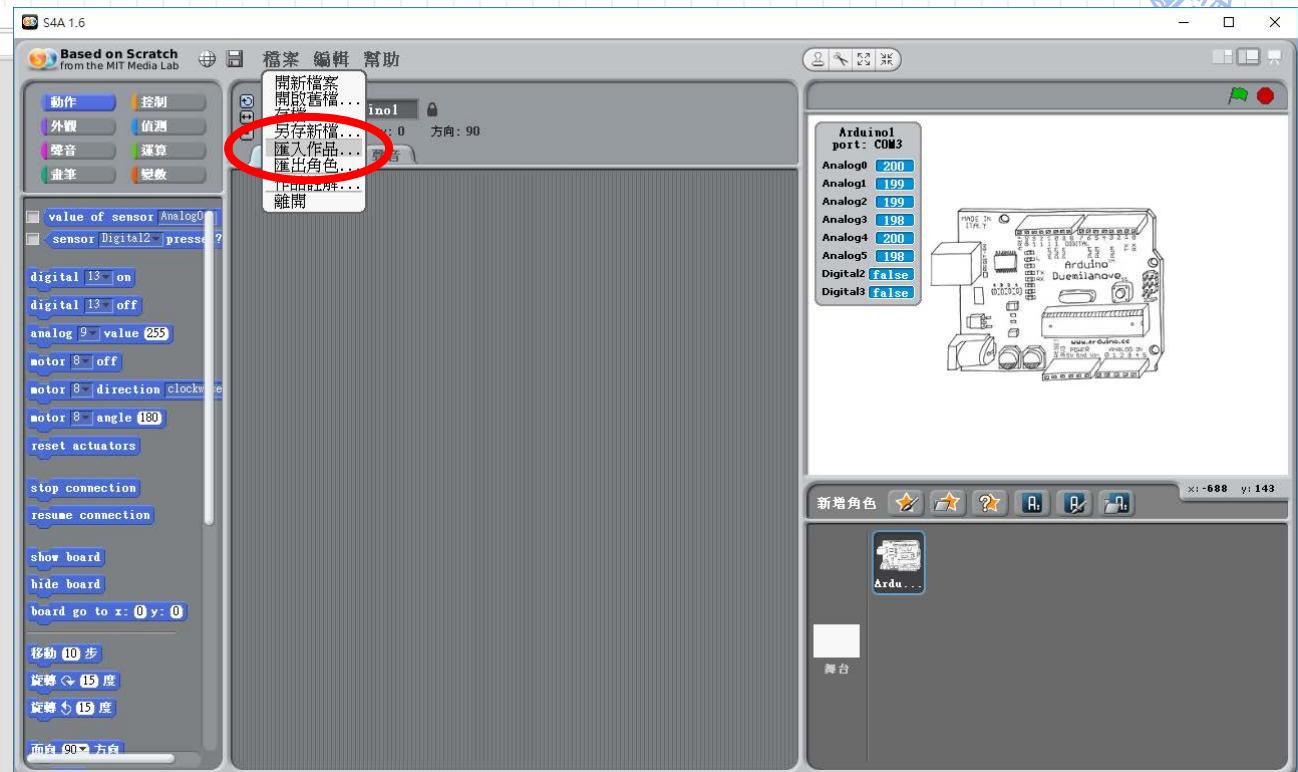
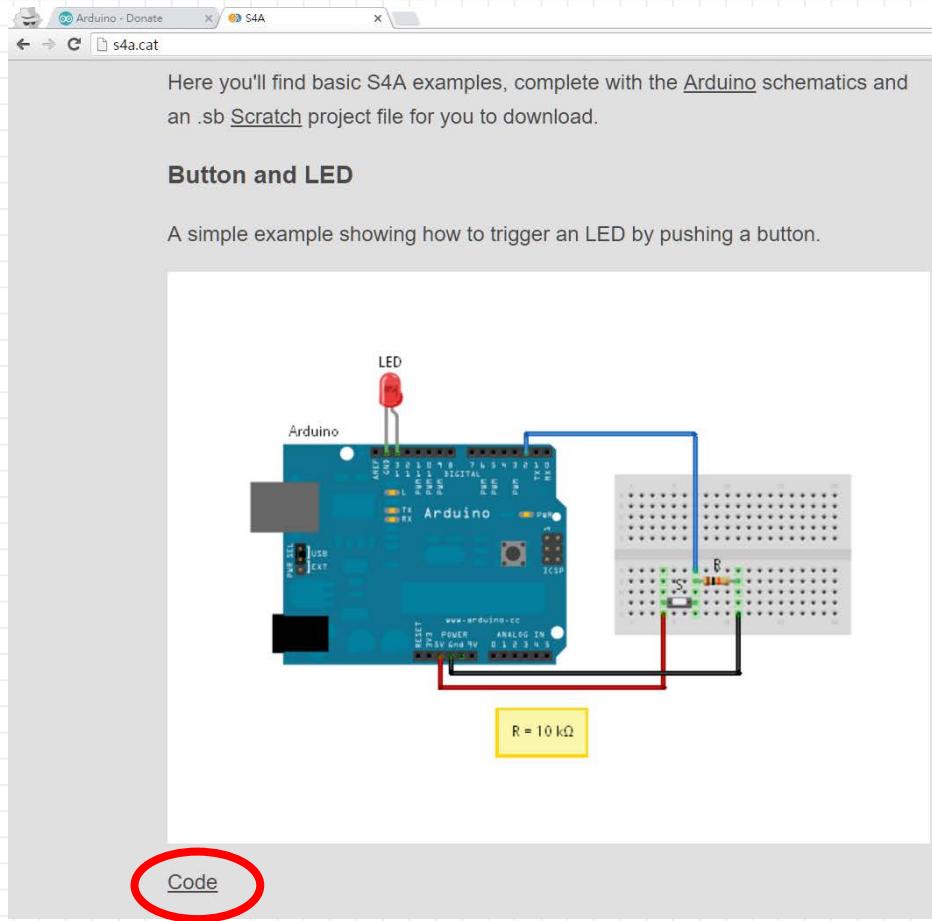
Scratch For Arduino (5/10)

- 燒錄完成後，開啟S4A。
注意，若Searching Board還在螢幕中未消失，代表正在抓取Arduino板子，長時間未消失的話，代表燒前一個步驟的韌體尚未燒錄成功。

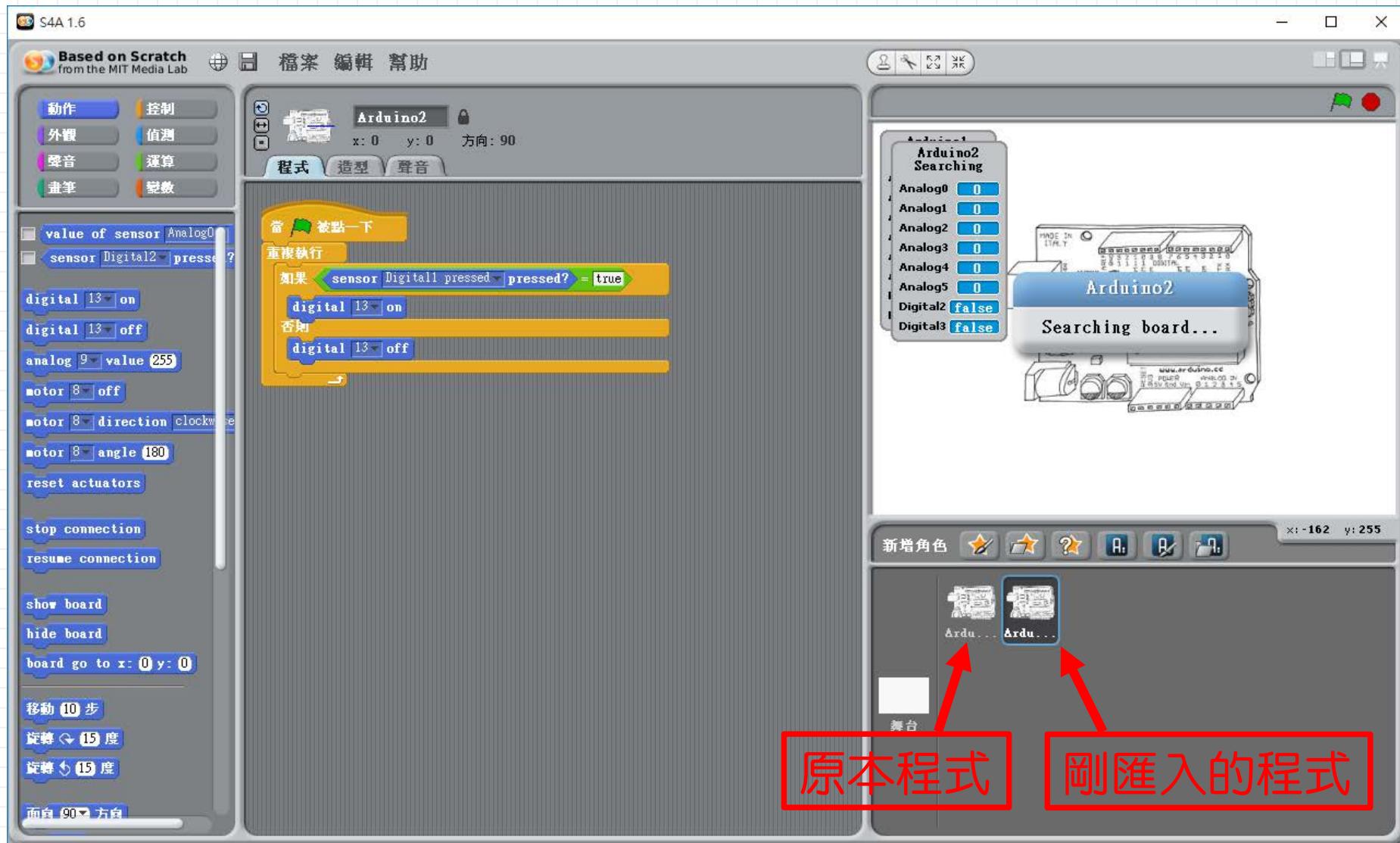


Scratch For Arduino (6/10)

- 下載官方網站的範例程式，並匯入S4A。



Scratch For Arduino (7/10)



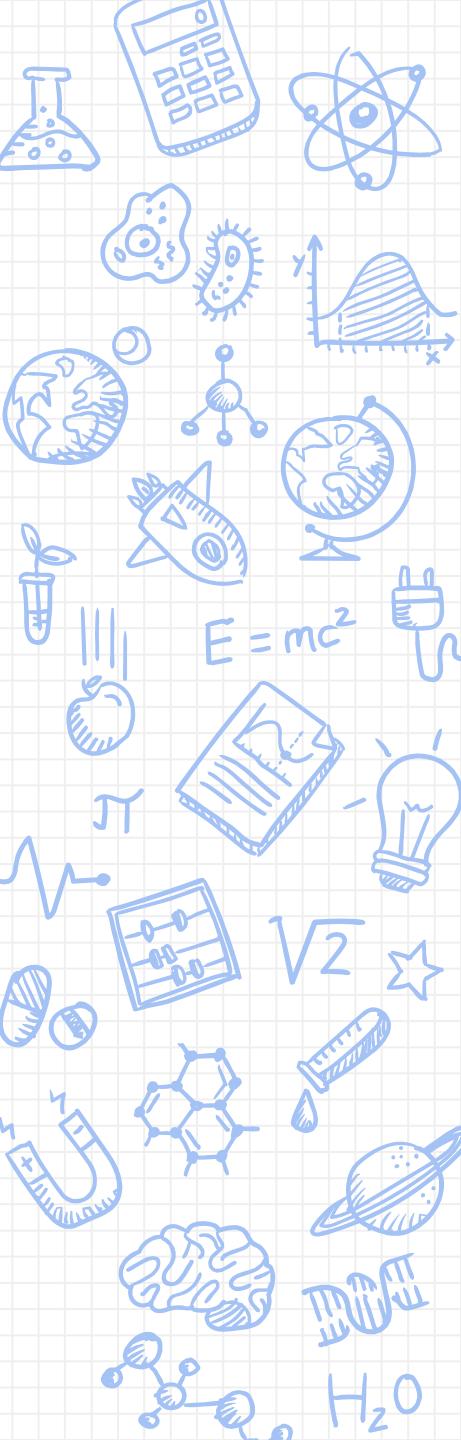
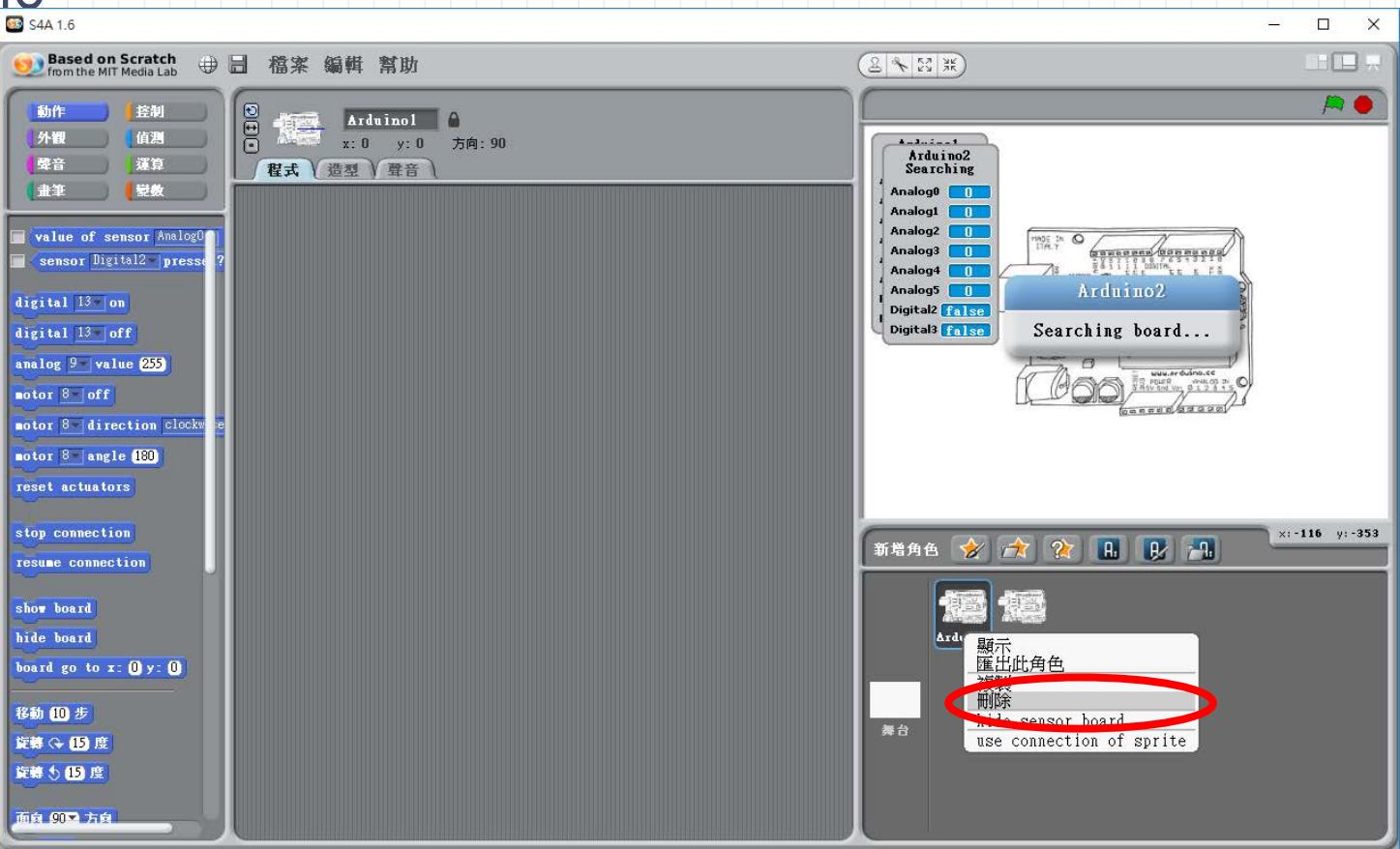
原本程式

剛匯入的程式

30

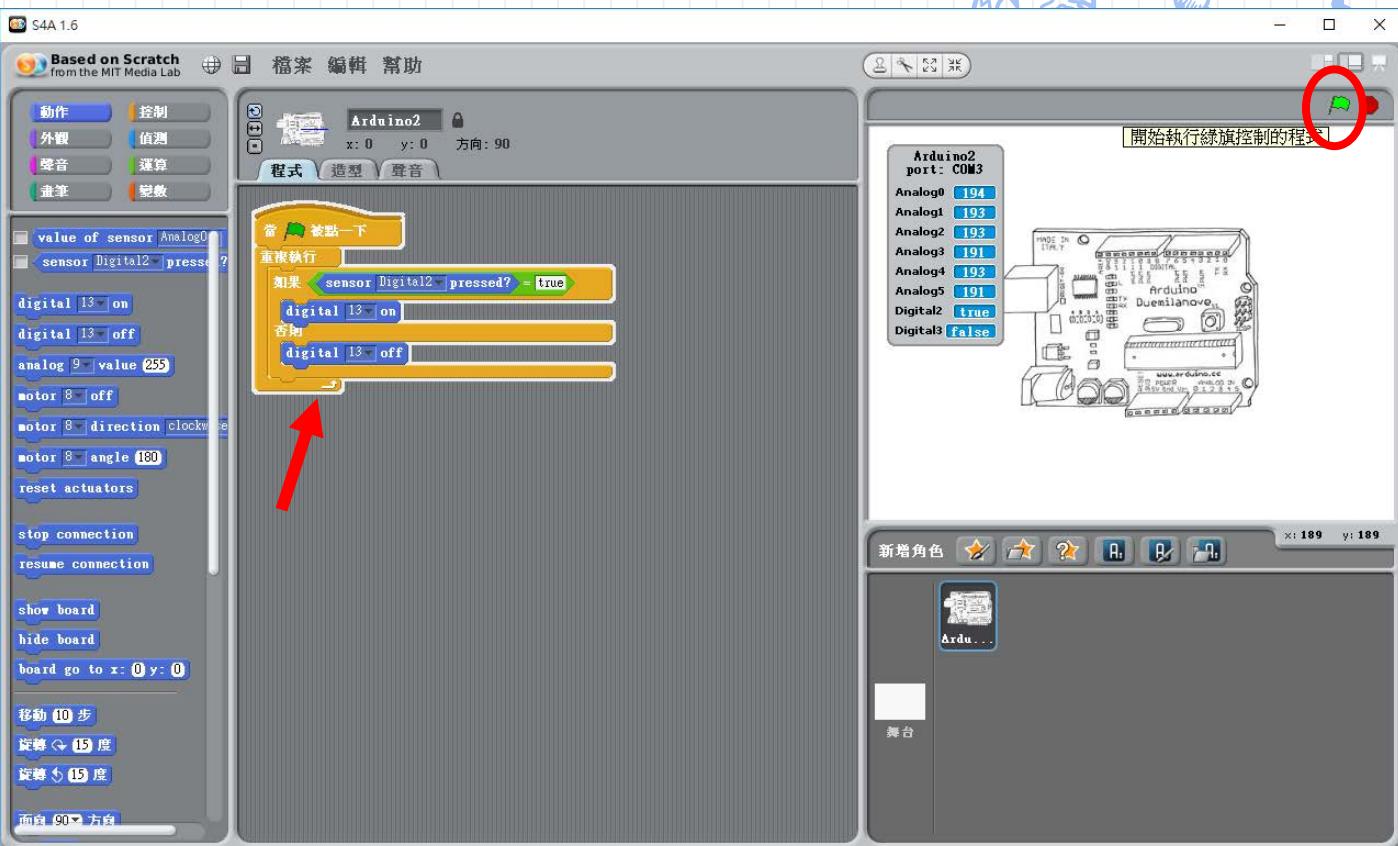
Scratch For Arduino (8/10)

- 必須把原本的專案刪除，才能讓匯入的專案抓到 Arduino。



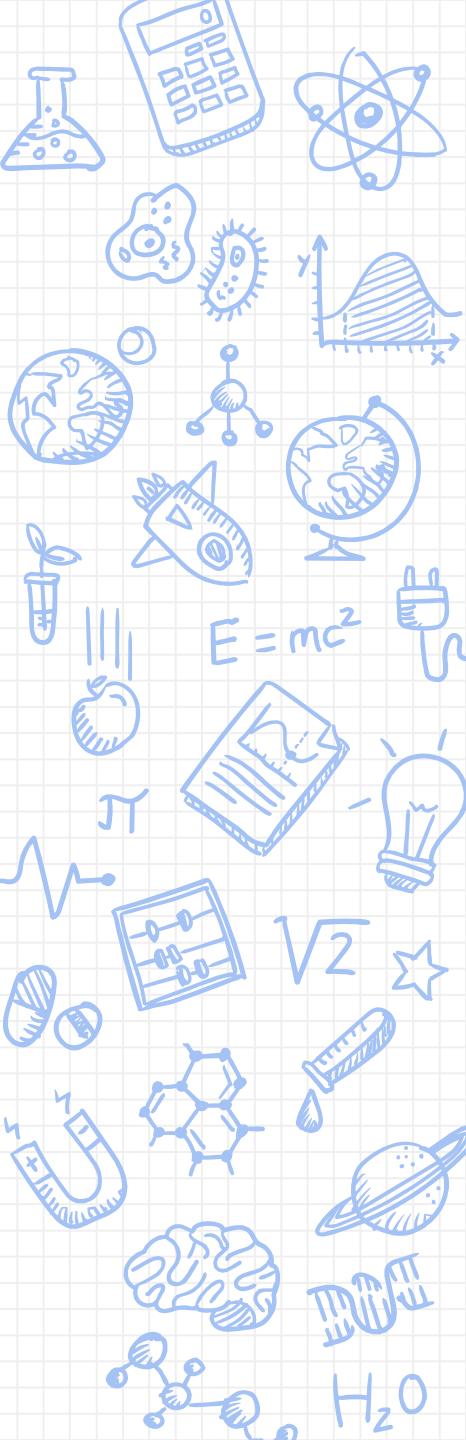
Scratch For Arduino (9/10)

- 修改程式碼後，按下路綠色旗標，代表程式已開始執行，程式碼外圍也會被米白色線條框住，代表正在執行中。



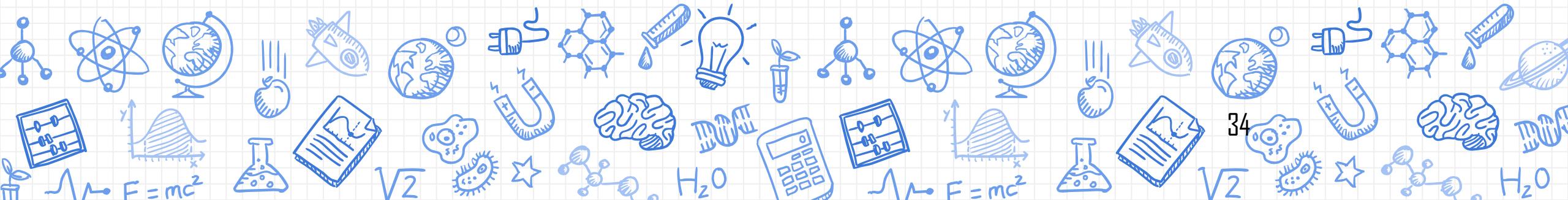
Scratch For Arduino (10/10)

- Demo 2: 嘗試修改範例程式，同第一題，以 Button控制LED。
按下Button ，LED燈亮。
放開Button ，LED燈滅。



Outline

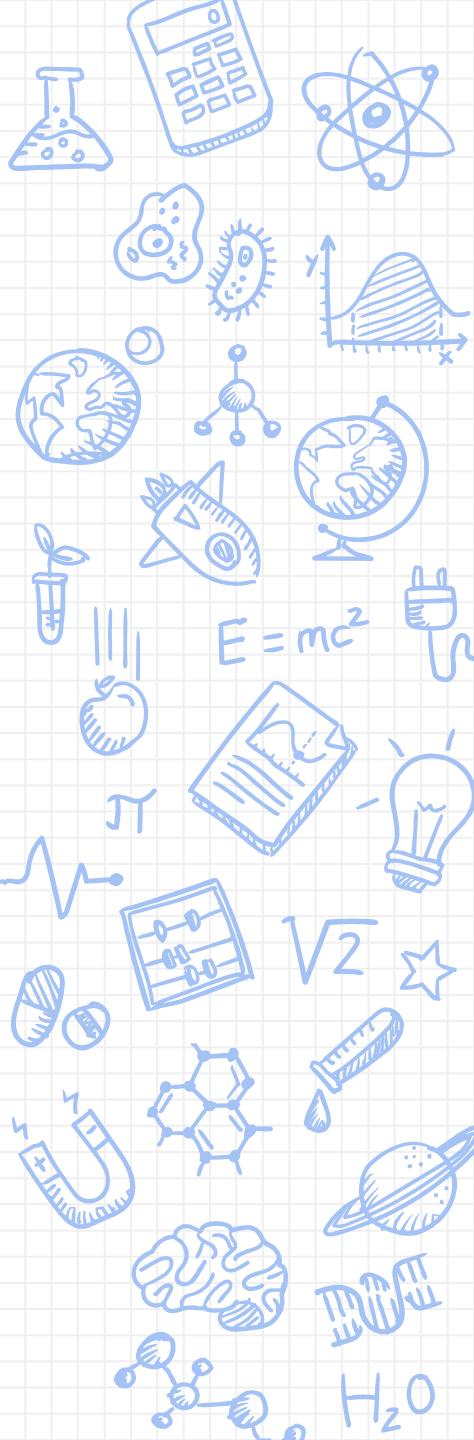
- 開放硬體介紹
- Arduino 環境架設
- Scratch For Arduino (S4A)
- 自由發揮小專題

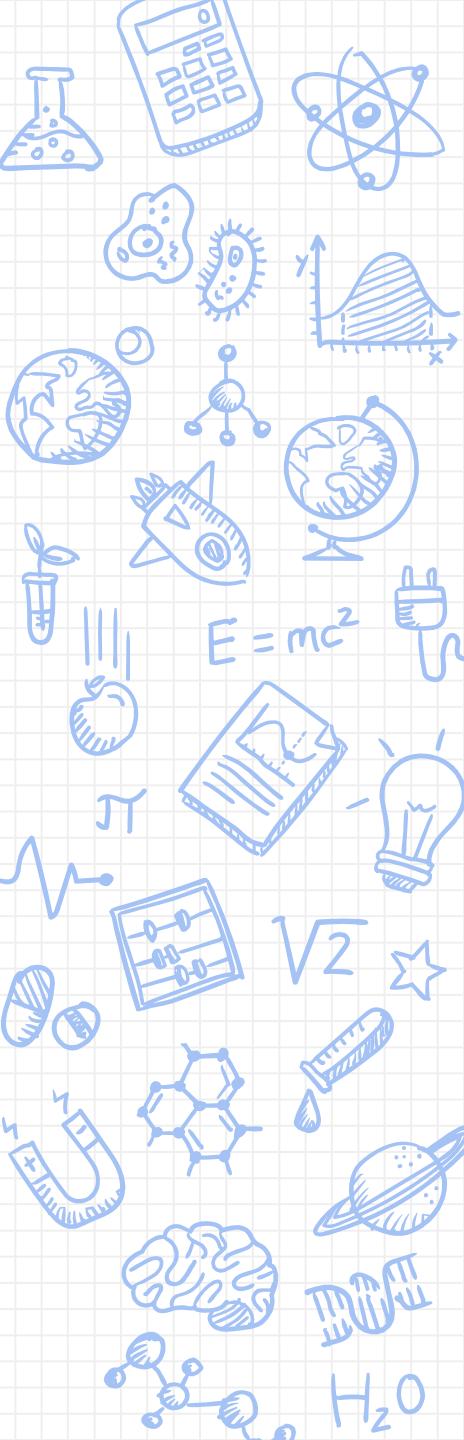


自由發揮小專題

- 使用兩種以上的Sensors，功能自由發揮，使用IDE或S4A皆可。
 - 提醒1：最好使用IDE，因為未來的程式幾乎都是用文字來實現。
 - 提醒2：IO的編號請重新命名，對未來Debug比較方便。
例如：

```
int LedPin = 13;  
pinMode(LedPin, OUTPUT);
```
 - 提醒3：在IDE介面中，ctrl + T 可以自動格式化，讓程式看起來更舒服。





小遊戲設計-讀秒計時器

- 遊戲設定：使用者藉由外接sensors(例如button)，設定時間範圍10~50秒，遊戲開始後，使用者心裡默數設定的秒數，時間到之後再按下button，由系統判斷是否猜到原先設定的秒數。
 - 遊戲細節請自行開發，展示時說明清楚即可。
 - 不限使用I/O，可由多個button輔助，亦可加LED/七段顯示器增進使用者體驗。
 - Hint：Arduino沒有提供精準的clock可以細算，故在讀秒時可利用計算void loop()這個函式跑一次需多少時間，設定一個counter來統計總時間。