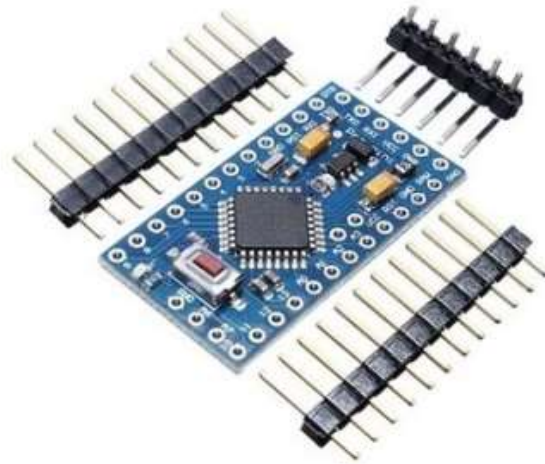


電光火石之路

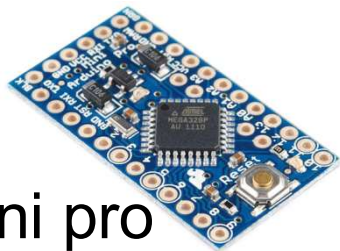
Arduino入門



Arduino 硬體端



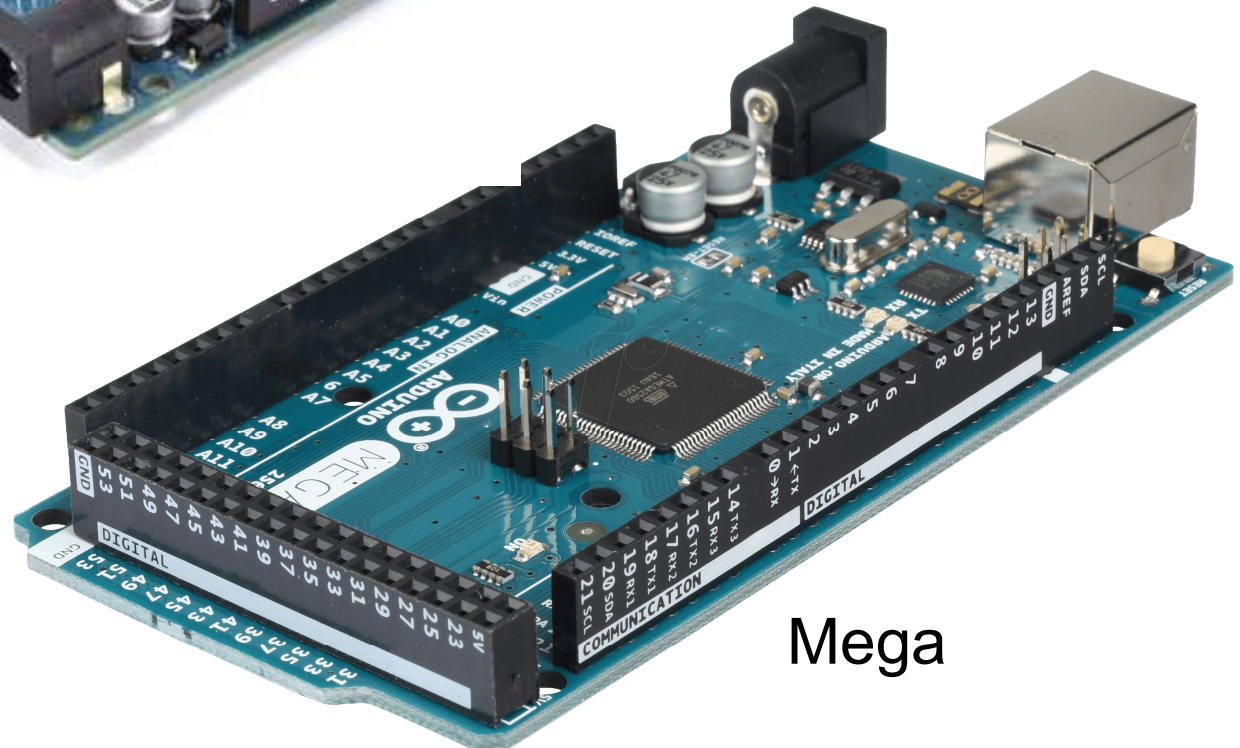
Uno R3



Mini pro

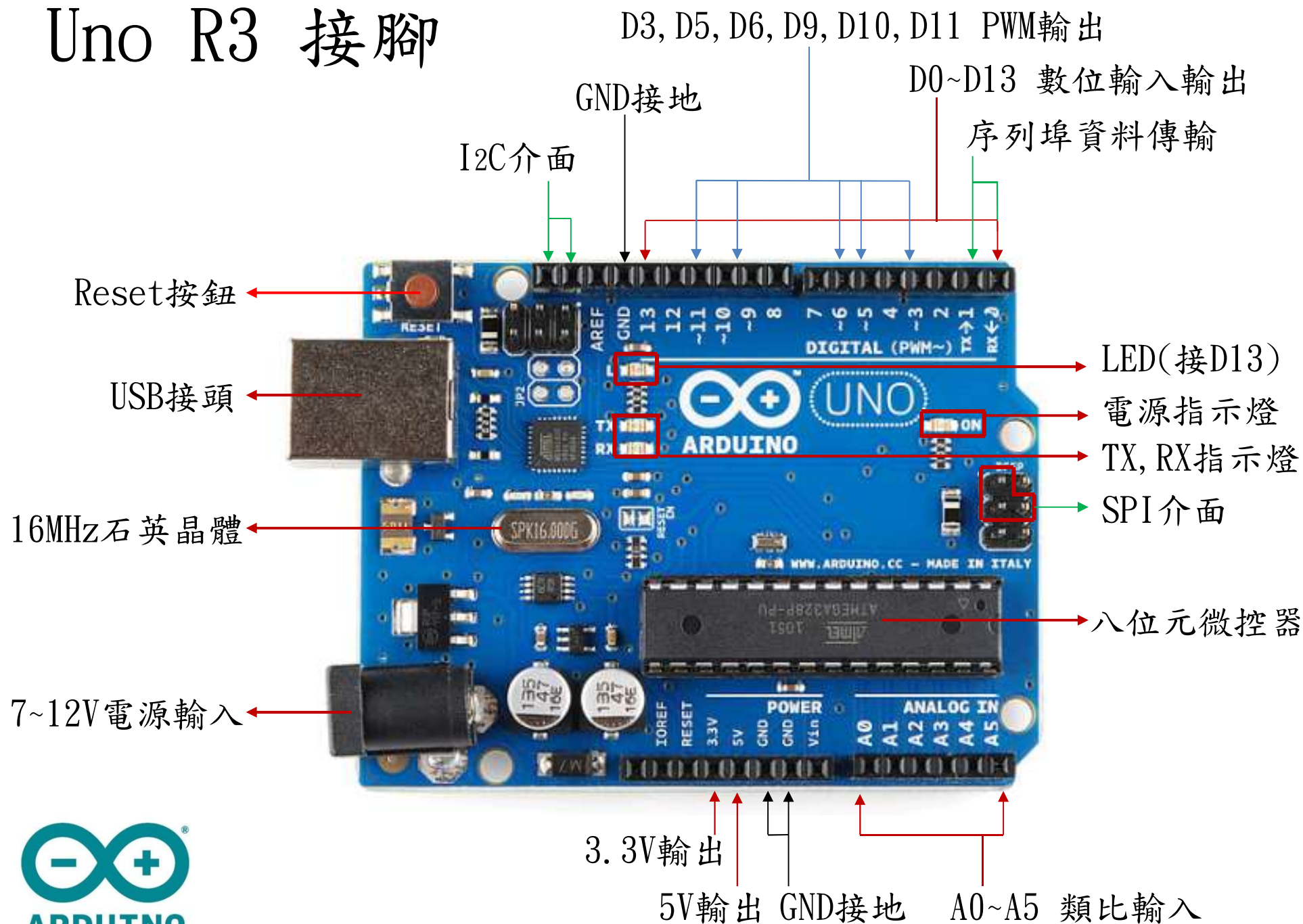


Nano



Mega

Uno R3 接腳



Arduino 軟體端



Download the Arduino IDE



ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

請選擇您的搜尋和安裝選項。



☒ 在這些位置中搜尋最好的驅動程式(S)

使用下列核取方塊來限制或擴充包括本機路徑和可卸除式媒體的預設搜尋，將安裝找到的最佳驅動程式。

☐ 搜尋可卸除式媒體 (軟碟，CD-ROM...)(M)

☒ 搜尋時包括這個位置(O):

C:\arduino-0021\drivers\FTDI USB Drivers

瀏覽(R)

☐ 不要搜尋，我將選擇要安裝的驅動程式(D)

選擇這個選項來從清單中選取裝置驅動程式。Windows 不保證您所選取的驅動程式最符合您的硬體。

< 上一步(B)

下一步(N) >

取消

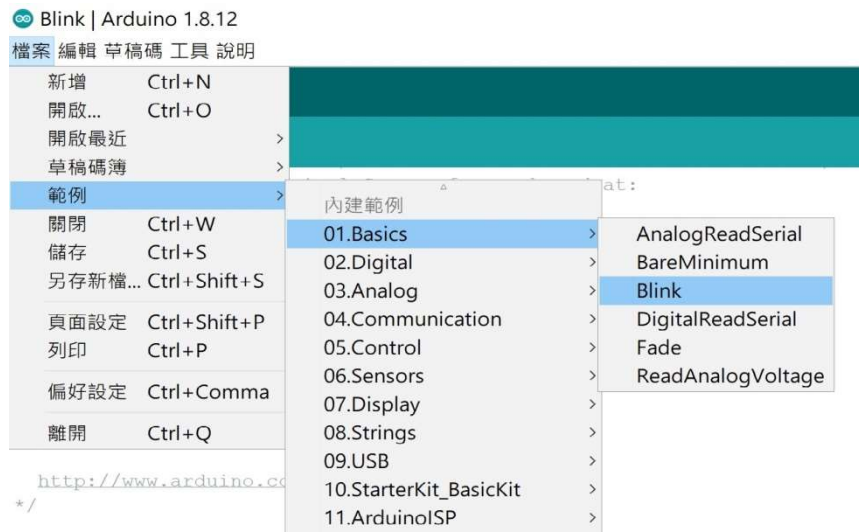
到官方網站arduino.cc下載軟體 Arduino IDE。最新版本為1.8.13。軟體不需要安裝，下載後解開壓縮檔即可使用，解壓縮之後滑鼠雙擊arduino.exe就可以啟動主程式。

Arduino板子可以透過USB供電，不需要另外接電源。將USB傳輸線一端接到電腦，一端接到Arduino板子後，Windows10版本會自動安裝，其他版本需搜尋USB驅動程式安裝。

Arduino 軟體設定



打開「裝置管理員」，點選連接埠，檢查Arduino接在哪個COM Port。如左圖，為Arduino板連接電腦成功，顯示目前Arduino接在COM4的位置。



軟體上傳測試，於桌面上點選 Arduino.exe 啟動 Arduino 程式，點選 **檔案** > **範例** > **1.Basics** > **Blink** 打開 Blink 範例程式。

Arduino 軟體設定

Blink | Arduino 1.8.2

檔案 編輯 草稿碼 工具 說明

✓ ↻ 📄 ⬆ ⬇ 驗證

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 (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  
}
```

編譯完畢。

草稿碼使用了 928 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。

Blink | Arduino 1.8.2

檔案 編輯 草稿碼 工具 說明

✓ ↻ 📄 ⬆ ⬇

Blink

by Scott Fitz

modified 2 Se

by Arturo Gua

modified 8 Se

by Colby Newm

*/

void setup() {

// initialize digital pin LED_BUILTIN as an output.

pinMode(LED_BUILTIN, OUTPUT);

}

// the setup function runs once when you press reset or power t

void setup() {

// initialize digital pin LED_BUILTIN as an output.

pinMode(LED_BUILTIN, OUTPUT);

}

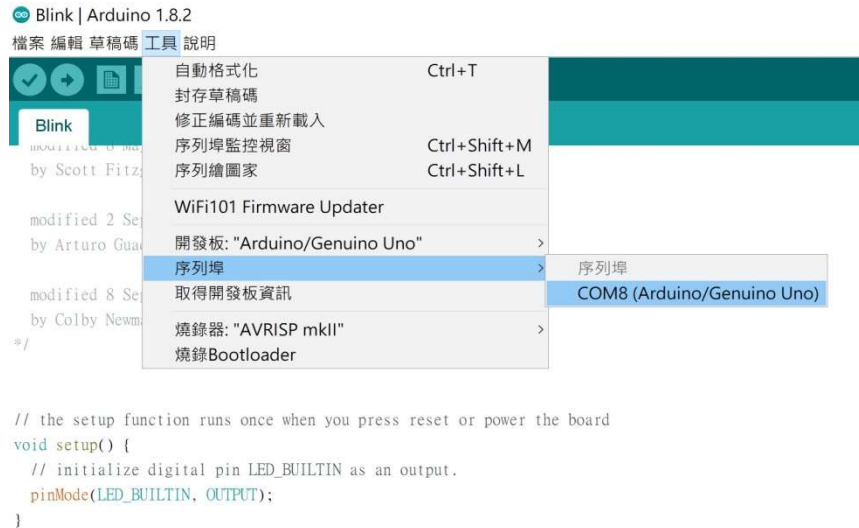
自動格式化 Ctrl+T
封存草稿碼
修正編碼並重新載入
序列埠監控視窗 Ctrl+Shift+M
序列繪圖家 Ctrl+Shift+L
WiFi101 Firmware Updater
開發板: "Arduino/Genuino Uno" >
序列埠 >
取得開發板資訊
燒錄器: "AVRISP mkII" >
燒錄Bootloader

開發板管理員...
Arduino AVR 板
Arduino Yún
• Arduino/Genuino Uno
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

接著按下  「驗證/編譯」這個按鈕編譯程式，假如程式語法沒有錯誤，畫面下方的狀態列會出現「編譯完畢」的訊息。

選擇跟你所用的 Arduino 對應的板子。點選 工具 > 開發板 > Arduino/Genuino Uno

Arduino 軟體設定

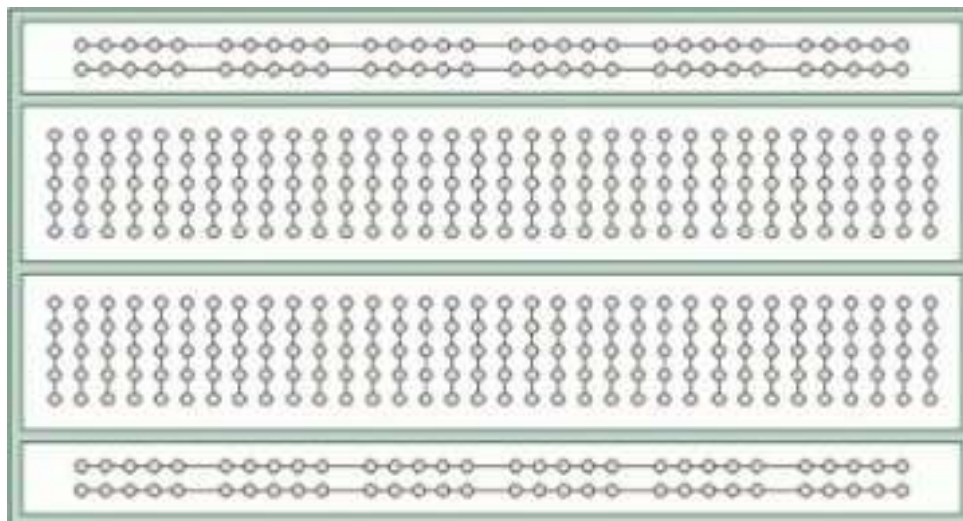
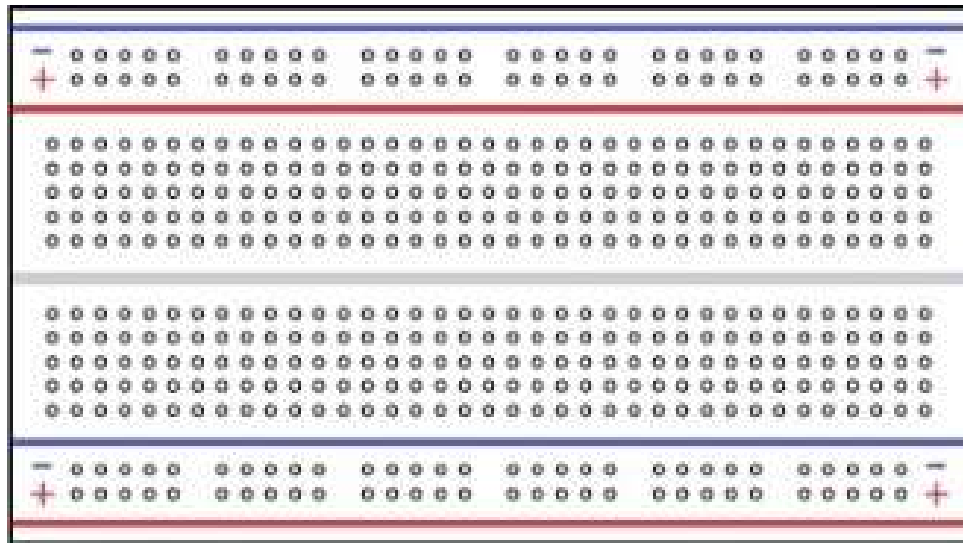


選擇Arduino所在的序列埠。以此圖例，點選 **工具** > **序列埠** > **COM8 (Arduino/Genuino Uno)**，此步驟很重要！若沒點選，會導致程式上傳失敗。



接著按下 **上傳** 這個按鈕，等候幾秒鐘，會看到板子上的 RX 和 TX 兩個燈號會快速地閃爍，如果上傳成功，狀態列會出現「**上傳完畢**」的訊息，此時板子LED的燈號(黃色)就會開始閃爍。

電路工具-麵包板



杜邦線



跳線



跳線盒



Arduino 程式架構

```
byte i = 13;
```

顏色非黑色者為程式保留字

全域常數、全域變數宣告區

```
void setup() {
```

```
  pinMode(i, OUTPUT);
```

```
}
```

指令注意大小寫

初始設定區(必要)

```
void loop() {
```

```
  digitalWrite(i, HIGH);
```

```
  delay(1000); // 延遲1秒
```

```
  digitalWrite(i, LOW);
```

```
  delay(1000);
```

```
}
```

前後有括號

程式末端必加分號

// 後方可加註解

主程式區(必要)

Arduino 常、變數資料類型

資料型態	說明	記憶體長度	數值範圍
boolean	布林	8 bits	true (1, HIGH), false(0, LOW)
byte	位元組	8 bits	0~255
char	字元	8 bits	-128~127
short	短整數	16 bits	-32768~32767
int	整數	16 bits	-32768~32767
word	字	16 bits	0~65535
long	長整數	32 bits	-2147483648~2147483647
float	浮點數	32 bits	+/-3.4028235e+38
double	倍精度浮點數	32 bits	+/-3.4028235e+38

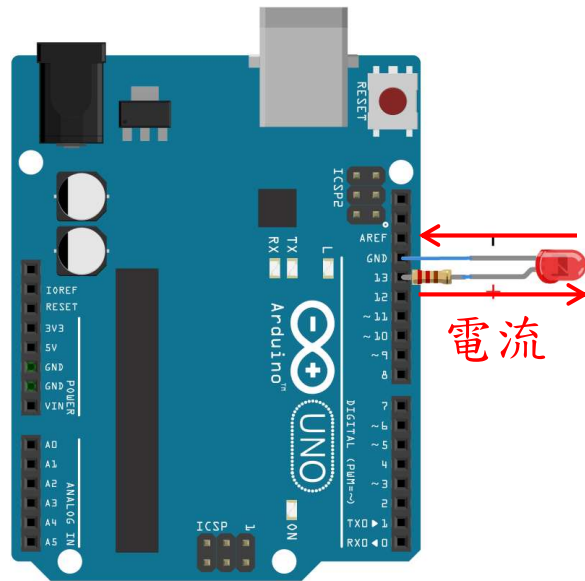
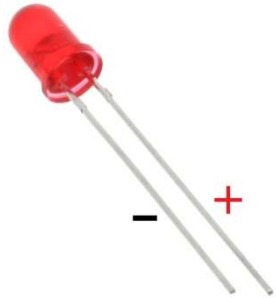
宣告變數一定要在前面加上資料類型，例如：

`boolean va2 = HIGH;` // 宣告布林變數 va2 的初始值為HIGH(高電壓)

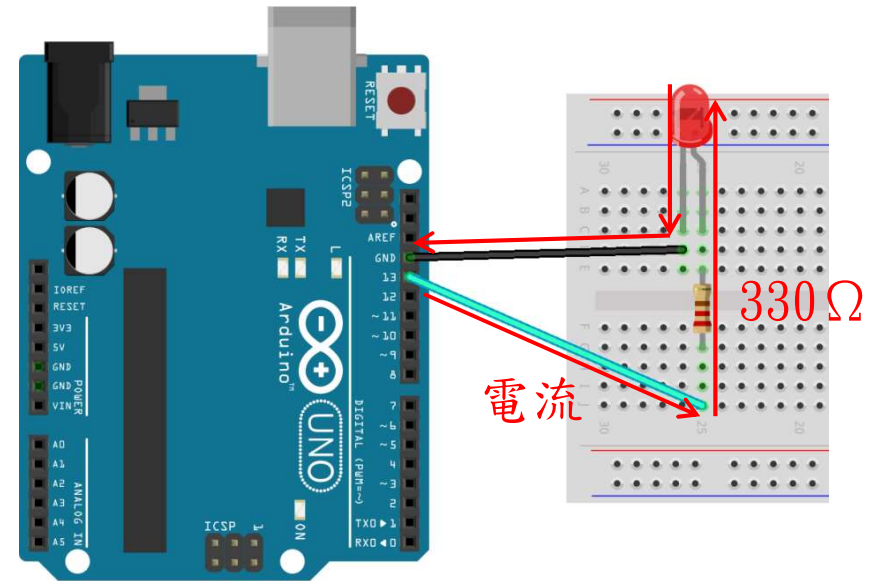
`const int n = 30000;` // 宣告整數常數 n 的值為30000

Arduino 數位輸出-閃光紅燈

紅光LED



連接方式1-焊接



連接方式2-麵包板

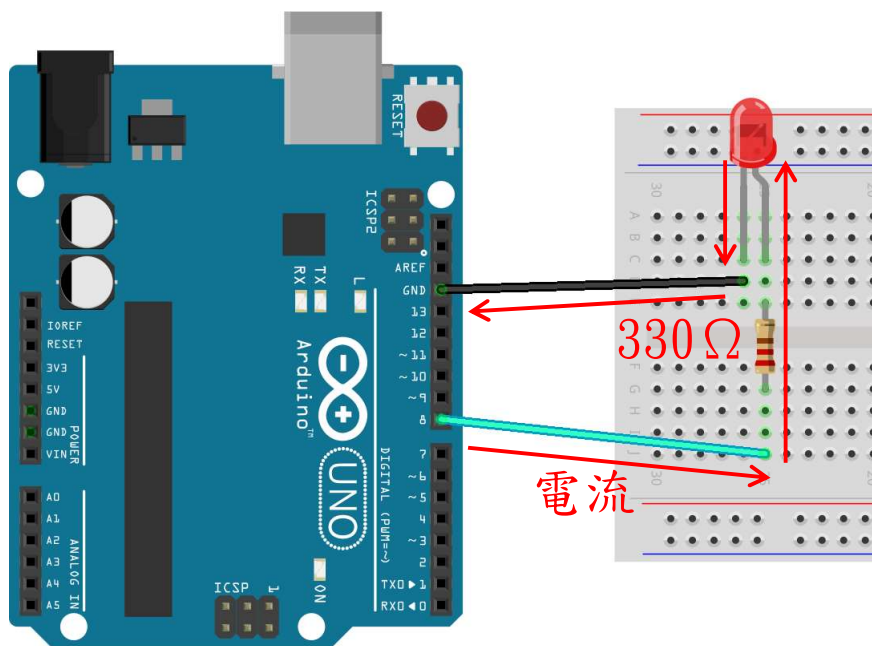
Arduino程式：

```
void setup() {  
    pinMode(腳位, 輸出OUTPUT或輸入INPUT)  
    pinMode(13, OUTPUT); // 以 PinMode(腳位,OUTPUT) 指令，設定LED (D13) 輸出  
}  
  
void loop() {  
    digitalWrite(腳位, 電壓高HIGH或低LOW)  
    digitalWrite(13, HIGH); // 以 digitalWrite(腳位,HIGH或LOW輸出) 指令，指定LED (D13) 高電位輸出  
    delay(1000); // 以 delay(幾毫秒) 指令，持續1000毫秒  
    digitalWrite(13, LOW); // 指定LED (D13) 低電位輸出  
    delay(1000); // 持續1000毫秒  
    delay(延遲幾毫秒)  
}
```

Arduino 數位輸出練習

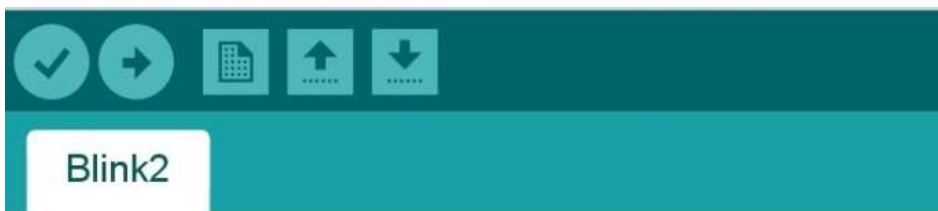
將數位輸出接腳由D13改成D8，
讓LED燈改成由D8控制，閃爍時
間間隔改為500毫秒。

D0~D13皆可數位輸出，可視為提供LED電壓及電流



Blink2 | Arduino 1.8.2

檔案 編輯 草稿碼 工具 說明



```
void setup() {  
  pinMode(8, OUTPUT); // 設定D8腳位輸出  
}
```

腳位由13改為8

```
void loop() {  
  digitalWrite(8, HIGH); // 指定D8腳位高電位輸出  
  delay(500); // 持續500毫秒  
  digitalWrite(8, LOW); // 指定D8腳位低電位輸出  
  delay(500); // 持續500毫秒  
}
```

延遲改為500毫秒

Arduino 程式常犯錯誤

```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(13, HIGH); 程式末端沒加分號  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

expected ';' before 'delay'

複製錯誤訊息

```
exit status 1  
expected ';' before 'delay'
```

```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(13, HIGH); 使用全形字體  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

stray '\357' in program

複製錯誤訊息

```
exit status 1  
stray '\357' in program
```

```
void setup() {  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

} 使用括號包住程式碼

expected '}' at end of input

複製錯誤訊息

```
exit status 1  
expected '}' at end of input
```

```
void setup() {  
  pinMode(button, OUTPUT);  
}
```

```
void loop() {  
  button = 13; 未宣告常、變數之資料型態  
  digitalWrite(button, HIGH);  
  delay(1000);  
  digitalWrite(button, LOW);  
  delay(1000);  
}
```

'button' was not declared in this scope

複製錯誤訊息

```
exit status 1  
'button' was not declared in this scope
```


Arduino

大家來找碴

```
const int buttonPin = 2;  
    ledPin = 13;  
int buttonState = 0;
```

- 1. 使用全形
- 2. 未宣告常數之資料型態

```
void setup() {
```

```
    pinMode(ledPin, OUTPUT);
```

```
    pinMode(buttonPin, INPUT)
```

- 3. 程式末端沒有分號

```
}
```

```
void loop() {
```

```
    buttonState = digitalRead(buttonPin);
```

- 4. 使用全形

```
    if (buttonState == HIGH) {
```

```
        digitalWrite(ledPin, HIGH);
```

- 5. 沒有括號 }

```
    else {
```

```
        digitalWrite(ledPin, LOW;
```

- 6. 沒有括號)

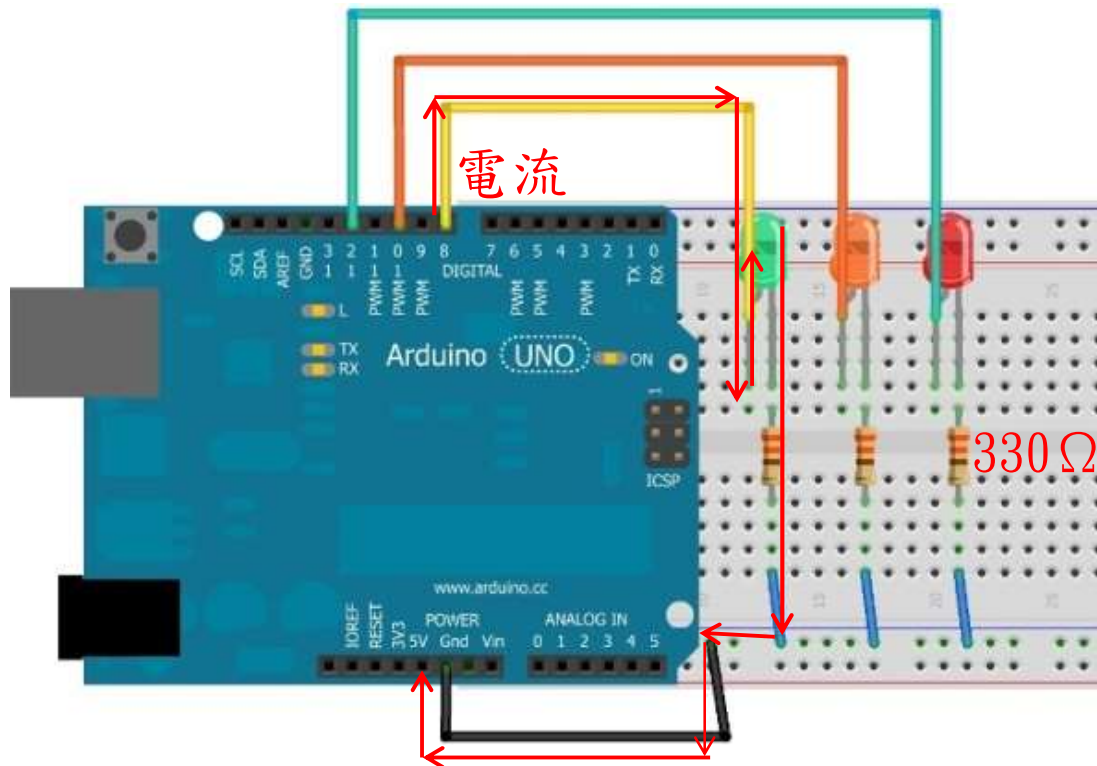
```
    }
```

```
}
```

Arduino 數位輸出

設計一個紅綠燈號誌，綠燈在熄滅後，黃燈會閃爍四次，最後亮紅燈。

D8、D10、D12分別輸出至綠光、黃光、紅光三種LED



```
void setup() {  
  pinMode(8, OUTPUT); // 設定D8腳位(綠燈)輸出  
  pinMode(10, OUTPUT); // 設定D10腳位(黃燈)輸出  
  pinMode(12, OUTPUT); // 設定D12腳位(紅燈)輸出  
}  
  
void loop() {  
  digitalWrite(8, HIGH); // 指定D8腳位(綠燈)亮5秒  
  delay(5000);  
  digitalWrite(8, LOW);  
  delay(100);
```

for (變數類型 i=起數;i<尾數;i++)
{ 要重複的程式 }

// 以for迴圈，指定D10腳位(黃燈)閃爍4次

```
for (byte i = 1 ; i < 5 ; i++){  
  digitalWrite(10, HIGH);  
  delay(300);  
  digitalWrite(10, LOW);  
  delay(300);  
}
```

```
digitalWrite(12, HIGH); // 指定D12腳位(紅燈)亮8秒  
delay(8000);  
digitalWrite(12, LOW);  
delay(100);  
}
```

```
int freq[] = {523, 587, 659, 698, 784, 880, 988, 1047}; // 將音階頻率以整數資料存入freq[]陣列
void setup() {
  pinMode(8, OUTPUT); // 設定D8腳位輸出
}
void loop() {
```

陣列：

資料類型 名稱[] = {資料1, 資料2, 資料3, ...}

編號0 編號1 編號2



無源蜂鳴器

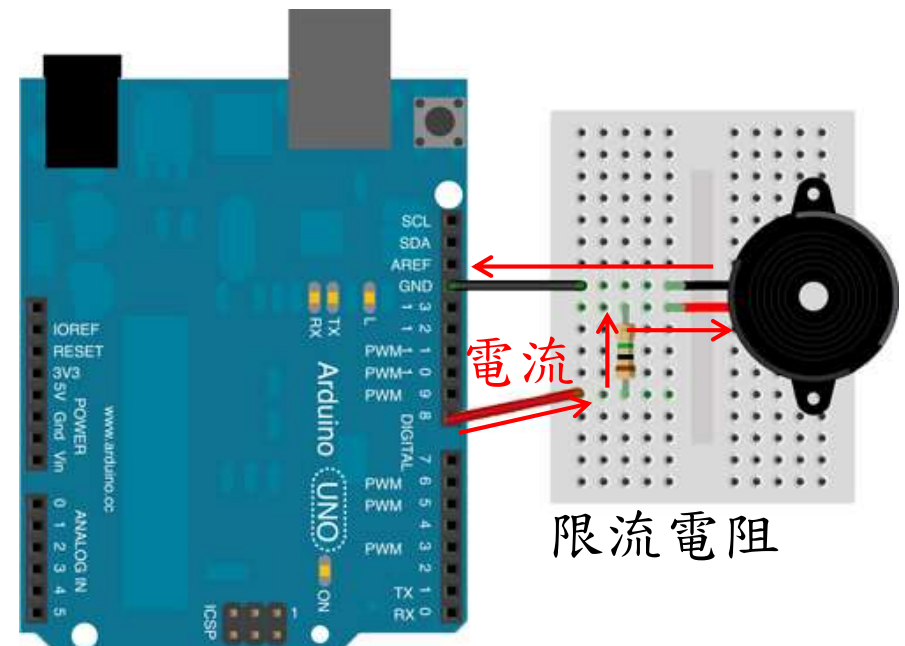
```
for (byte i=0; i<8; i++) { // 設定位元變數i，以for迴圈由i=0遞增至i=7指定陣列中的頻率
  tone(8, freq[i]); // 以 tone(腳位,freq()陣列對應頻率) 指令，發聲
  delay(500); // 發聲持續500毫秒
  noTone(8); // 以 noTone(腳位) 指令，停止發聲
}
```

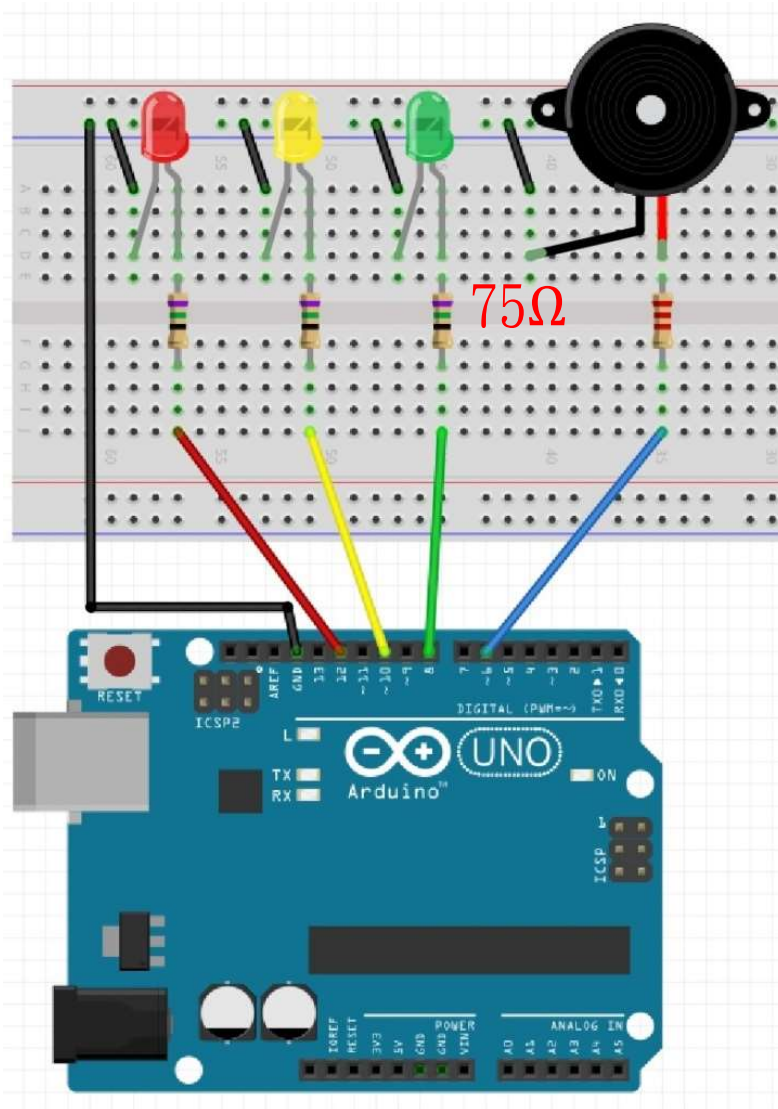
陣列結合迴圈，

由迴圈i=0, 1, 2, 3, 4, 5, 6, 7

帶入陣列中8個音階頻率並發聲

音階程式





Arduino 練習 紅綠燈加蜂鳴器

```
void setup() {
  pinMode(8, OUTPUT); // 設定D8腳位(綠燈)輸出
  pinMode(10, OUTPUT); // 設定D10腳位(黃燈)輸出
  pinMode(12, OUTPUT); // 設定D12腳位(紅燈)輸出
  pinMode(6, OUTPUT); // 設定D6腳位輸出至蜂鳴器
}

void loop() {

  digitalWrite(8, HIGH); // 指定D8腳位(綠燈)亮9秒
  for (byte i = 1; i < 4; i++) { // 以for迴圈，指定D6腳位發聲3次
    tone(6, 1047); // 發出高音Do 0.4秒
    delay(400);
    noTone(6);
    delay(100);
    tone(6, 784); // 發出Sol 0.5秒
    delay(500);
    noTone(6);
    delay(2000);
  }
  digitalWrite(8, LOW); // 指定D8腳位(綠燈)熄滅

  for (byte i = 1; i < 5; i++) { // 以for迴圈，指定D10腳位(黃燈)閃爍4次
    digitalWrite(10, HIGH);
    tone(6, 1047); // 發出高音Do 0.3秒
    delay(300);
    noTone(6);
    digitalWrite(10, LOW);
    delay(300);
  }

  digitalWrite(12, HIGH); // 指定D12腳位(紅燈)亮4秒
  tone(6, 1047); // 發出高音Do
  delay(4000);
  noTone(6);
  digitalWrite(12, LOW);
  delay(100);

}
```

Arduino

蜂鳴器練習

小蜜蜂

1=C 5 3 3 4 2 2 1 2 3 4 5 5 5

小 蜜 蜂 嗡 嗡 嗡， 大 家 一 起 来 做 工。

5 3 3 4 2 2 1 3 5 5 3 —

来 匆 匆， 去 匆 匆， 做 工 兴 味 浓。

2 2 2 2 2 3 4 3 3 3 3 3 4 5

天 暖 花 开 不 做 工， 将 来 哪 里 能 过 冬？

5 3 3 4 2 2 1 3 5 5 1 —

快 做 工， 快 做 工， 别 学 懒 惰 虫。

小蜜蜂樂譜

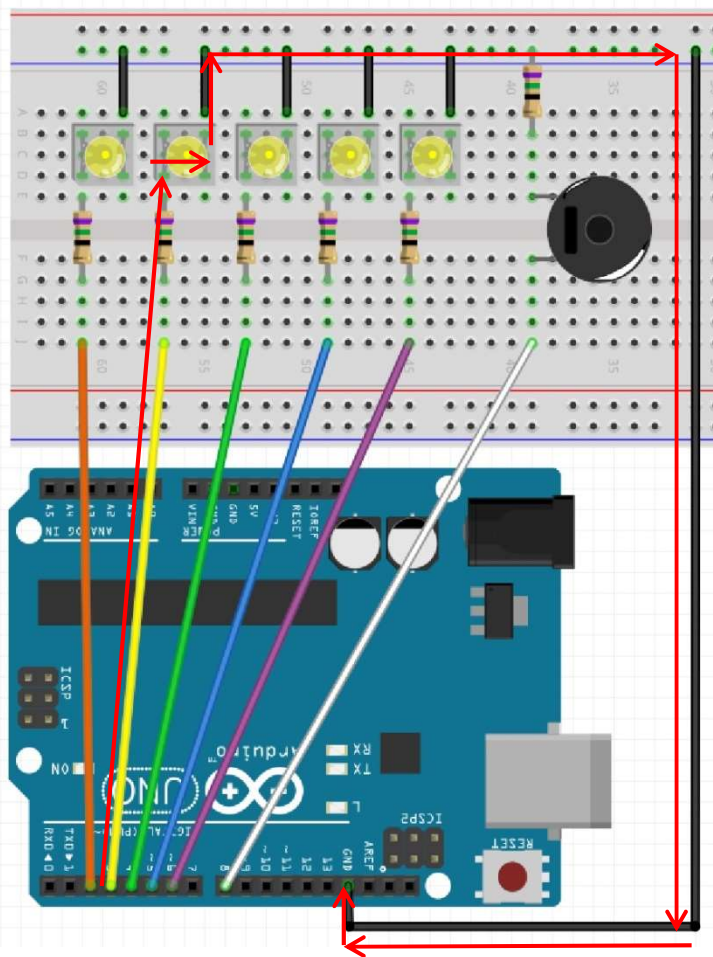
```
int freq[] = {523,587,659,698,784,880,988,1047}; // 將頻率存入陣列
```

```
int melody[] = // 將曲子音調存入melody[]陣列  
{5,3,3,4,2,2,1,2,3,4,5,5,5,  
5,3,3,4,2,2,1,3,5,5,3,  
2,2,2,2,2,3,4,  
3,3,3,3,3,4,5,  
5,3,3,4,2,2,1,3,5,5,1};
```

```
int beat[] = // 將曲子拍子存入beat[]陣列  
{1,1,2,1,1,2,1,1,1,1,1,1,2,  
1,1,2,1,1,2,1,1,1,1,4,  
1,1,1,1,1,1,2,  
1,1,1,1,1,1,2,  
1,1,2,1,1,2,1,1,1,1,4};
```

```
void setup() {  
  pinMode(8,OUTPUT); // D8輸出至蜂鳴器  
}
```

```
void loop() {  
  for(int i = 0 ; i < 49 ; i++){ // 以for迴圈，依49個音符之音調及拍子發聲  
    tone(8, freq[melody[i]-1]);  
    delay(beat[i]*500); // 1個拍子500毫秒  
    noTone(8);  
  }  
  delay(2000); // 曲終，延遲2000毫秒  
}
```



Arduino 練習 蜂鳴器加LED

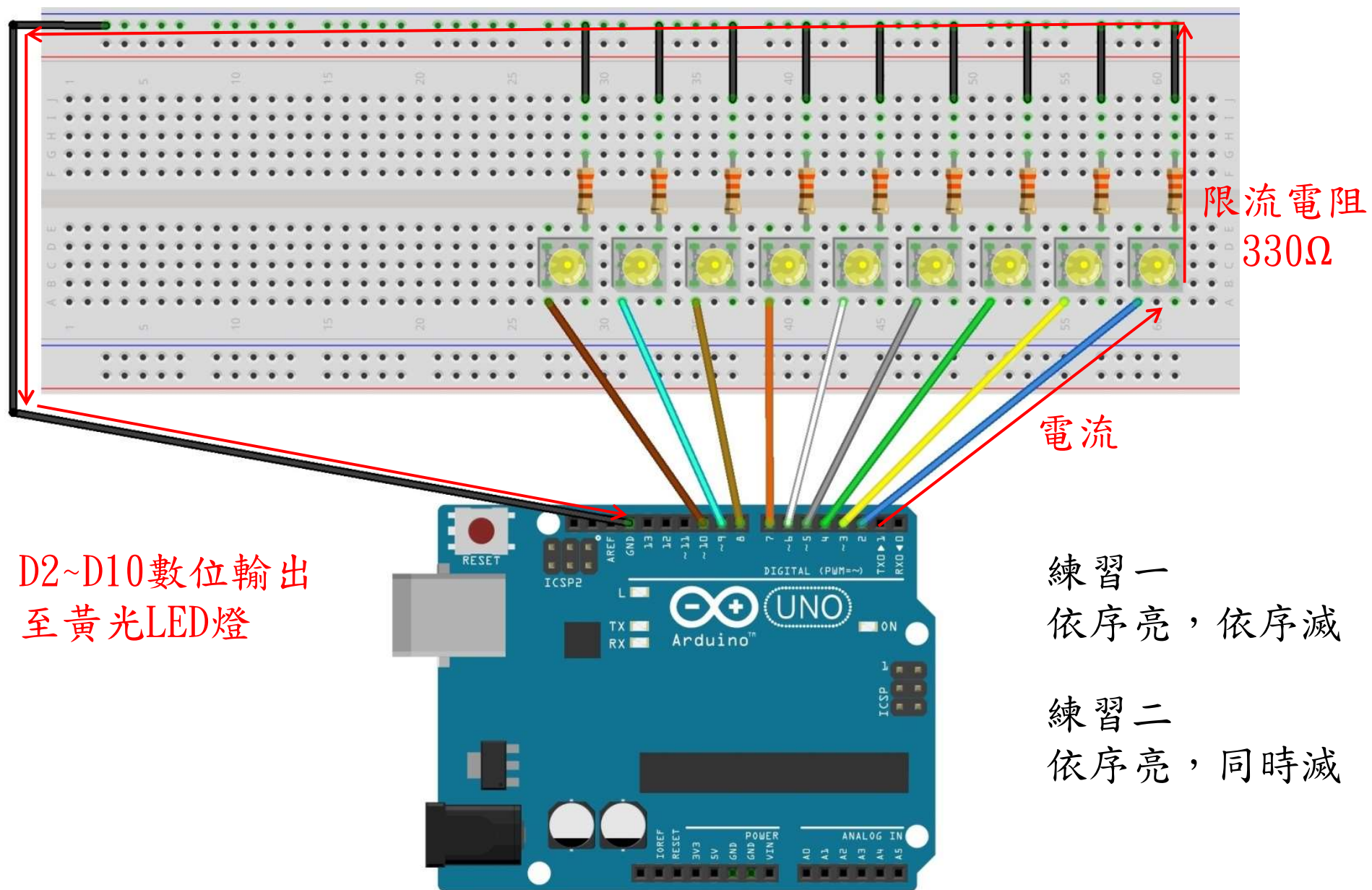
```
int freq[] = {523,587,659,698,784,880,988,1047}; // 將音階頻率存入freq[]陣列

int melody[] = // 將曲子49個音符之音調存入melody[]陣列
{5,3,3,4,2,2,1,2,3,4,5,5,5,
 5,3,3,4,2,2,1,3,5,5,3,
 2,2,2,2,2,3,4,3,3,3,3,3,4,5,
 5,3,3,4,2,2,1,3,5,5,1};

int beat[] = // 將曲子49個音符之拍子長度存入beat[]陣列
{1,1,2,1,1,2,1,1,1,1,1,1,2,
 1,1,2,1,1,2,1,1,1,1,4,
 1,1,1,1,1,1,2,1,1,1,1,1,2,
 1,1,2,1,1,2,1,1,1,1,4};

void setup() {
  pinMode(8,OUTPUT); // D8輸出至蜂鳴器
  pinMode(2,OUTPUT); // D2輸出至LED1=Do
  pinMode(3,OUTPUT); // D3輸出至LED2=Re
  pinMode(4,OUTPUT); // D4輸出至LED3=Mi
  pinMode(5,OUTPUT); // D5輸出至LED4=Fa
  pinMode(6,OUTPUT); // D6輸出至LED5=Sol
}

void loop() {
  for(int i = 0 ; i < 49 ; i++){ // 以for迴圈，依49個音符之音調及拍子發聲
    tone(8, freq[ melody[i]-1 ]); // 使用雙陣列freq[melody[ ]]讓蜂鳴器發出指定頻率
    digitalWrite(melody[i]+1, HIGH); // 音符+1=指定腳位點亮LED
    delay(beat[i]*500); // 用beat[]陣列讓蜂鳴器發出指定拍子長度，1拍500毫秒
    noTone(8);
    digitalWrite(melody[i]+1, LOW);
    delay(20); // 避免同音調音符聽起來沒間段
  }
  delay(2000); // 曲終，延遲2000毫秒
}
```

Arduino 練習-多LED控制

```
const byte t=40; // 設定常數t=40ms
void setup() {
```

```
// 以for迴圈依序設定D2~D10腳位輸出
for (byte i = 2 ; i < 11 ; i++){
  pinMode(i, OUTPUT);
}
```

```
}
void loop() {
```

```
// 以for迴圈依序點亮D2~D10，並帶入常數t=40ms
for (byte i = 2 ; i < 11 ; i++){
  digitalWrite(i, HIGH);
  delay( t);
}
```

```
// 以for迴圈依序熄滅D2~D10，並帶入常數t=40ms
for (byte i = 2 ; i < 11 ; i++){
  digitalWrite(i, LOW);
  delay( t);
}
```

程式一
依序亮，依序滅

```
const byte t=70; // 設定常數t=70ms
void setup() {
```

```
// 以for迴圈依序設定D2~D10腳位輸出
for (byte i = 2 ; i < 11 ; i++){
  pinMode(i, OUTPUT);
}
```

```
}
void loop() {
```

```
// 以for迴圈依序點亮D2~D10，並帶入常數t=70ms
for (byte i = 2 ; i < 11 ; i++){
  digitalWrite(i, HIGH);
  delay( t);
}
```

```
// 以for迴圈熄滅全部D2~D10
for (byte i = 2 ; i < 11 ; i++){
  digitalWrite(i, LOW);
}
```

```
delay(t); // 延遲70ms
}
```

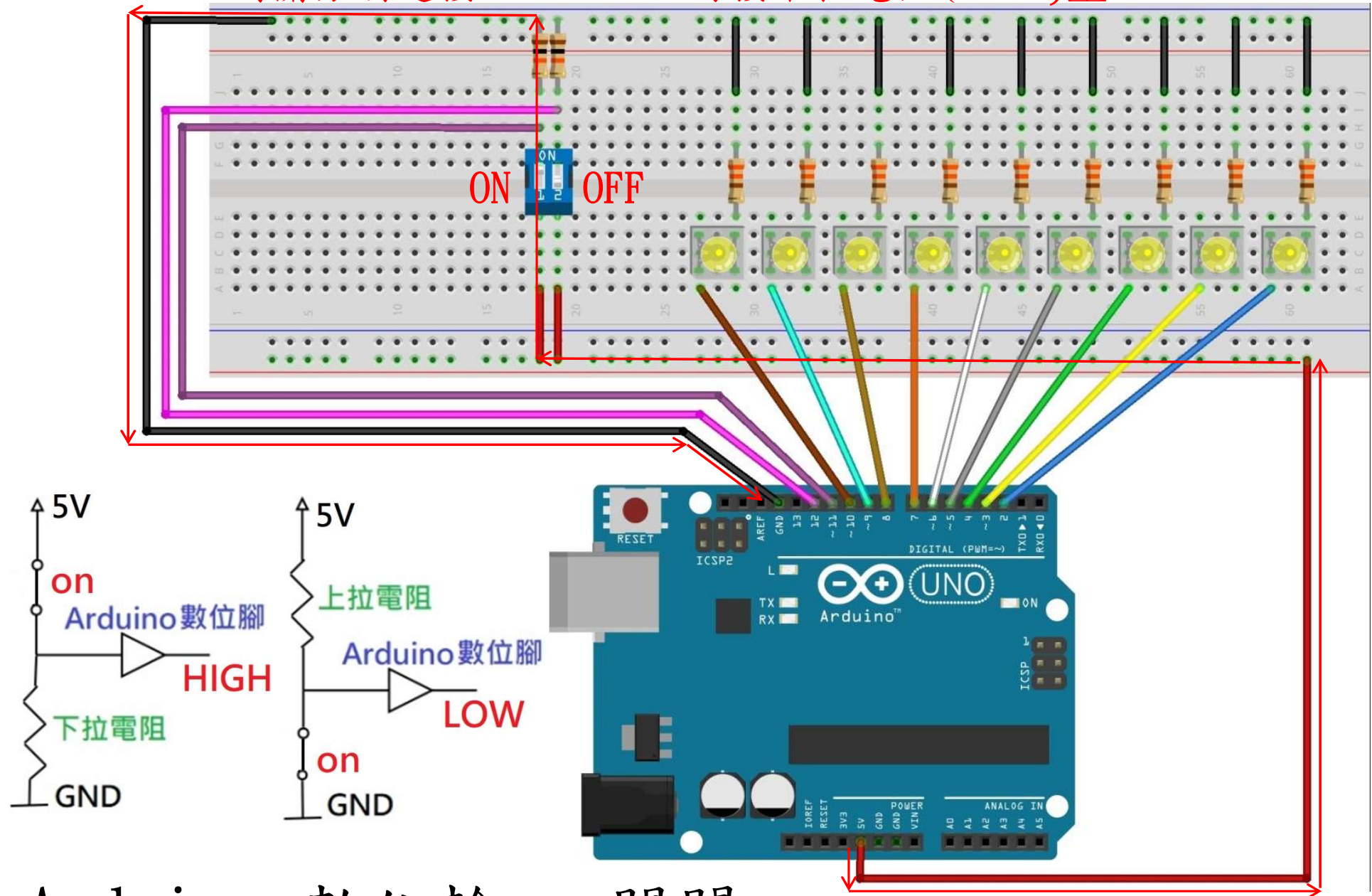
Arduino練習

多LED控制

兩種程式都想要！
可否設計切換？

程式二
依序亮，同時滅

開關分別連接D11、D12，再接下拉電阻(10kΩ)至GND



Arduino 數位輸入-開關

Arduino 應用 開關切換方向燈

```
void setup() {  
  for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈依序設定D2~D10腳位輸出  
    pinMode(i, OUTPUT);  
  }  
  for (byte i=11; i < 13 ; i++) { // 以for迴圈設定D11~D12腳位輸入  
    pinMode(i, INPUT);  
  }  
}
```

```
void loop() {  
  boolean val1 = digitalRead(11); // 讀取D11~D12,將結果存入變數val1,val2 } 讀取D11,D12信號  
  boolean val2 = digitalRead(12);  
  if (val1==HIGH && val2==LOW) { // 如果D11接通且D12斷開,執行第一種模式 } 判斷是否模式1  
    const byte t=40; // 設定常數t=40ms  
    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈依序點亮D2~D10,並帶入常數t=40ms  
      digitalWrite(i, HIGH);  
      delay( t);  
    }  
    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈依序熄滅D2~D10,並帶入常數t=40ms  
      digitalWrite(i, LOW);  
      delay( t);  
    }  
  }  
  else if (val1==LOW && val2==HIGH) { // 如果D11斷開且D12接通,執行第二種模式 } 判斷是否模式2  
    const byte t=70; // 設定常數t=70ms  
    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈依序點亮D2~D10,並帶入常數t=70ms  
      digitalWrite(i, HIGH);  
      delay( t);  
    }  
    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈熄滅全部D2~D10  
      digitalWrite(i, LOW);  
    }  
    delay(t); // 延遲70ms  
  }  
}
```

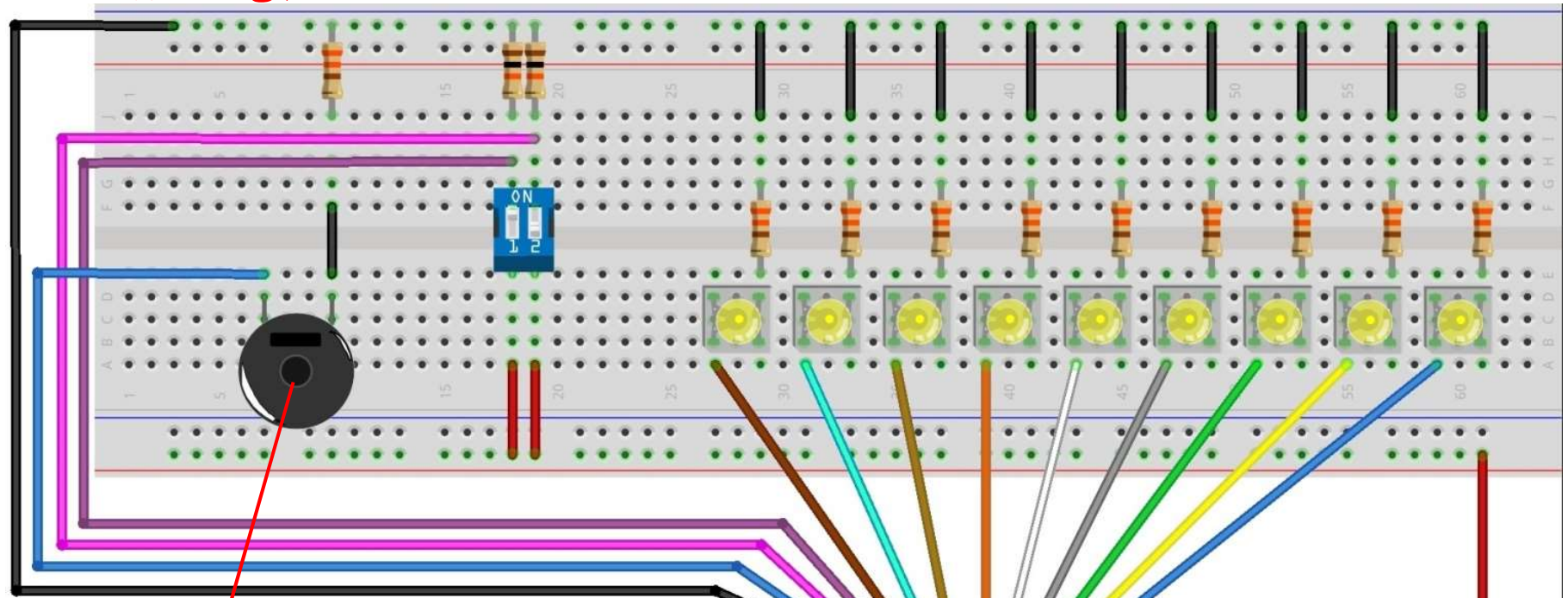
if (條件) {若符合,執行程式}

else if (條件) {若符合,執行程式}

模式1

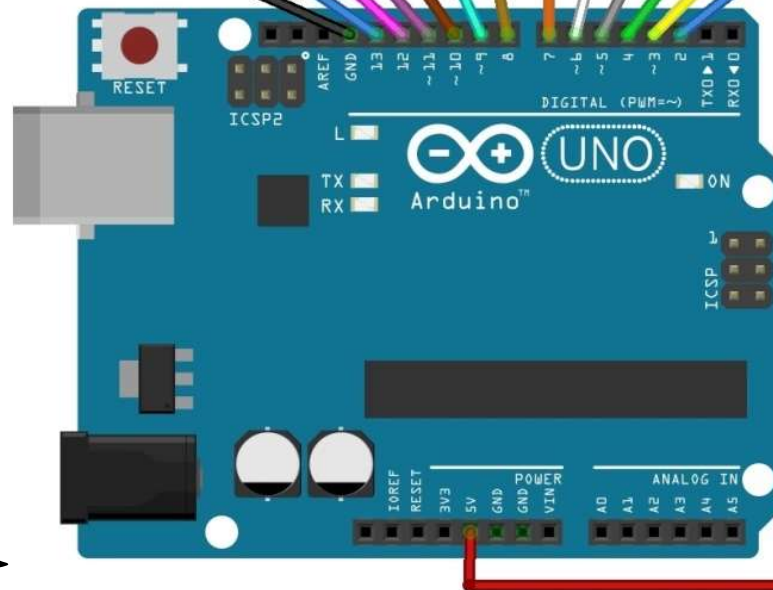
模式2

限流電阻330Ω



D13輸出至蜂鳴器，並
接上限流電阻(330Ω)

Arduino 應用
方向燈加音效



Arduino 應用

方向燈加音效

```
void setup() {
  for (byte i = 2 ; i < 11 ; i++) { pinMode(i, OUTPUT); } // 以for迴圈依序設定D2~D10腳位輸出
  for (byte i=11; i < 13; i++) { pinMode(i, INPUT); } // 以for迴圈設定D11~D12腳位輸入
  pinMode(13, OUTPUT); // 設定D13腳位輸出至蜂鳴器
}

void loop() {
  boolean val1 = digitalRead(11); // 讀取D11~D12,將結果存入變數val1,val2
  boolean val2 = digitalRead(12);
  if (val1==HIGH && val2==LOW) { // 如果D11接通且D12斷開,執行第一種模式
    const byte t=40; // 設定常數t=40ms
    tone(13, 100); // D2亮燈前,蜂鳴器100Hz低音輸出,持續10毫秒
    delay(10);
    noTone(13);

    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈依序點亮D2~D10,並帶入常數t=40ms
      if (i==9) { tone(13, 361); // D9亮燈前,蜂鳴器361Hz高音輸出,持續10毫秒
        delay(10);
        noTone(13);}

      digitalWrite(i, HIGH);
      delay( t);
    }

    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈依序熄滅D2~D10,並帶入常數t=40ms
      digitalWrite(i, LOW);
      delay( t);
    }
  }
  else if (val1==LOW && val2==HIGH) { // 如果D11斷開且D12接通,執行第二種模式
    const byte t=70; // 設定常數t=70ms
    tone(13, 100); // D2亮燈前,蜂鳴器100Hz低音輸出,持續10毫秒
    delay(10);
    noTone(13);

    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈依序點亮D2~D10,並帶入常數t=70ms
      if (i==6) { tone(13, 361); // 亮到D6時,蜂鳴器361Hz高音輸出,持續10毫秒
        delay(10);
        noTone(13);}

      digitalWrite(i, HIGH);
      delay( t);
    }

    for (byte i = 2 ; i < 11 ; i++) { // 以for迴圈熄滅全部D2~D10
      digitalWrite(i, LOW);
    }
    delay(t); // 延遲70ms
  }
}
```

100Hz低音

361Hz高音

100Hz低音

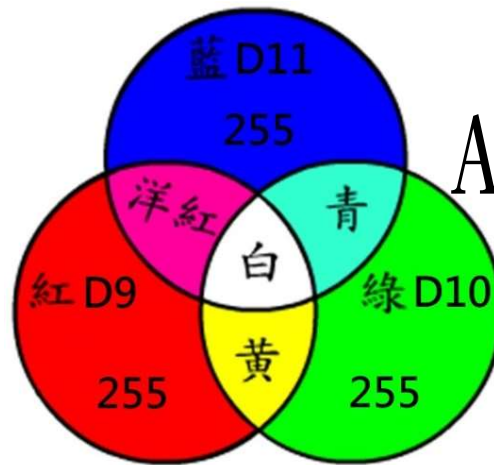
361Hz高音

增加音效程式


```

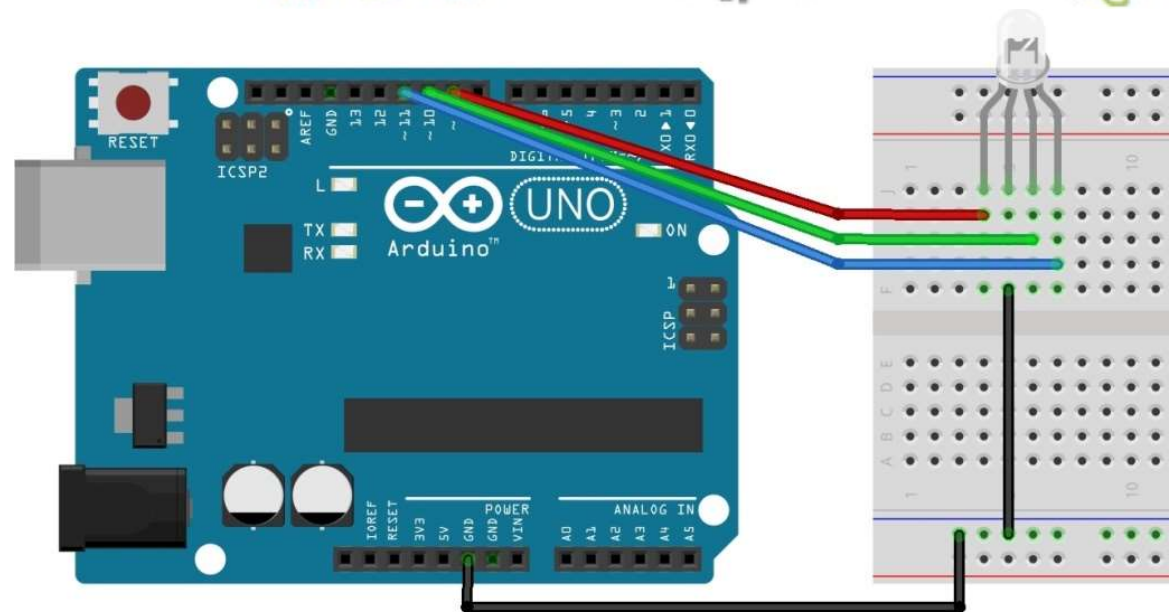
void setup() {
  pinMode( 9, OUTPUT); // 設定D9腳位紅光輸出
  pinMode(10, OUTPUT); // 設定D10腳位綠光輸出
  pinMode(11, OUTPUT); // 設定D11腳位藍光輸出
}
void loop() {
  analogWrite( 9, 255); // 紅光輸出
  analogWrite(10,  0);
  analogWrite(11,  0);
  delay(1000); // 持續1000毫秒
  analogWrite( 9,  0); // 綠光輸出
  analogWrite(10, 255);
  analogWrite(11,  0);
  delay(1000);
  analogWrite( 9,  0); // 藍光輸出
  analogWrite(10,  0);
  analogWrite(11, 255);
  delay(1000);
  analogWrite( 9, 255); // 紅+綠=黃光輸出
  analogWrite(10, 255);
  analogWrite(11,  0);
  delay(1000);
  analogWrite( 9,  0); // 綠+藍=青光輸出
  analogWrite(10, 255);
  analogWrite(11, 255);
  delay(1000);
  analogWrite( 9, 255); // 紅+藍=洋紅光輸出
  analogWrite(10,  0);
  analogWrite(11, 255);
  delay(1000);
  analogWrite( 9, 255); // 紅+綠+藍=白光輸出
  analogWrite(10, 255);
  analogWrite(11, 255);
  delay(1000);
}

```



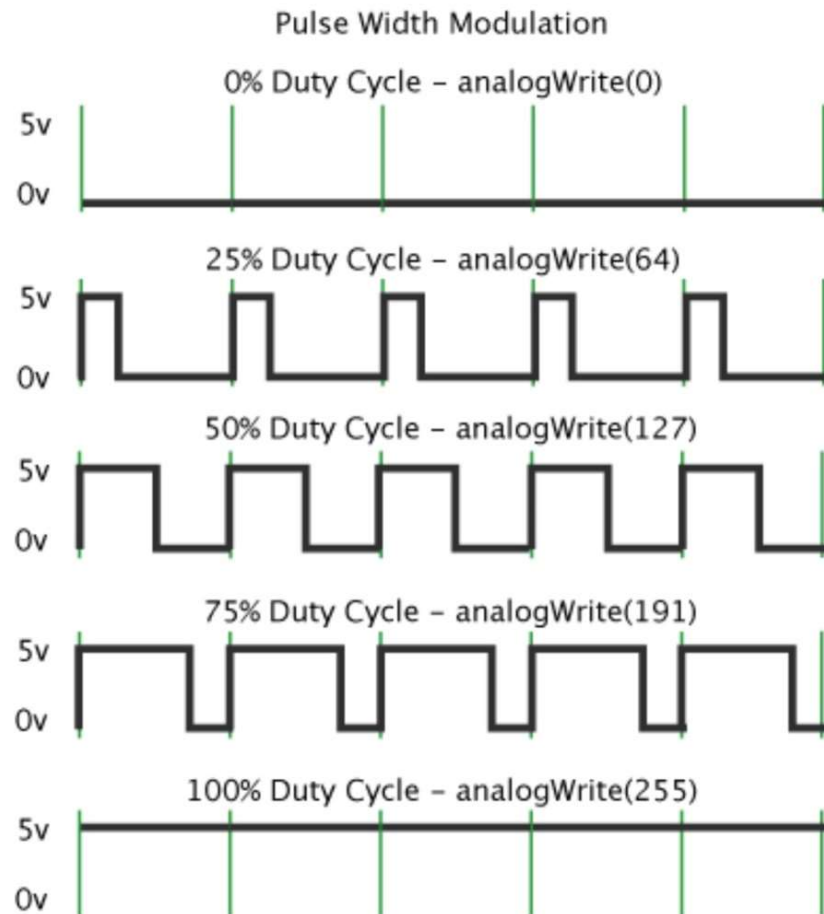
Arduino PWM輸出 三色LED

`analogWrite(腳位, 0~255)`



D9, D10, D11分別連接R, G, B

Arduino PWM(脈衝寬度調變)輸出



輸出電壓=5V×0%=0V

輸出電壓=5V×25%=1.25V

輸出電壓=5V×50%=2.5V

輸出電壓=5V×75%=3.75V

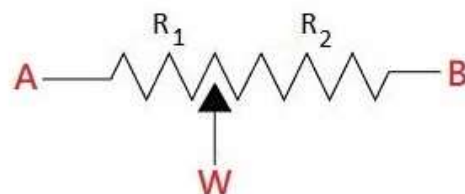
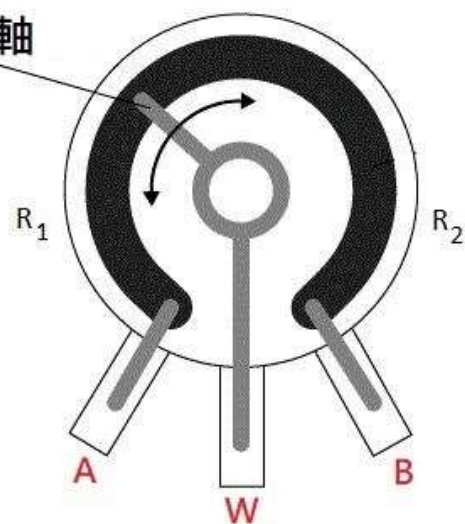
輸出電壓=5V×100%=5V

Arduino D3, D5, D6, D9, D10, D11等接腳，可提供PWM輸出

可變電阻

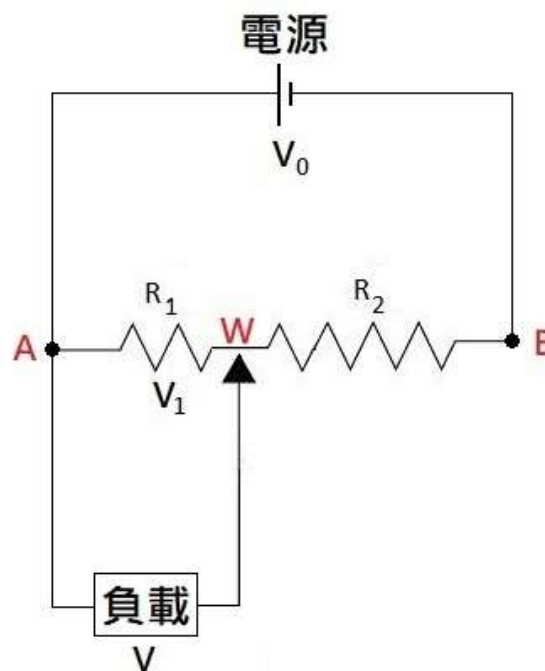


銅轉軸



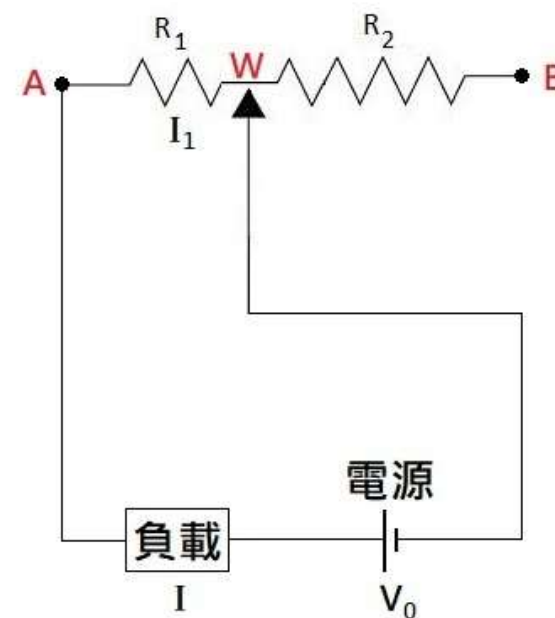
電路符號

電壓控制



負載電壓 $V=V_1$

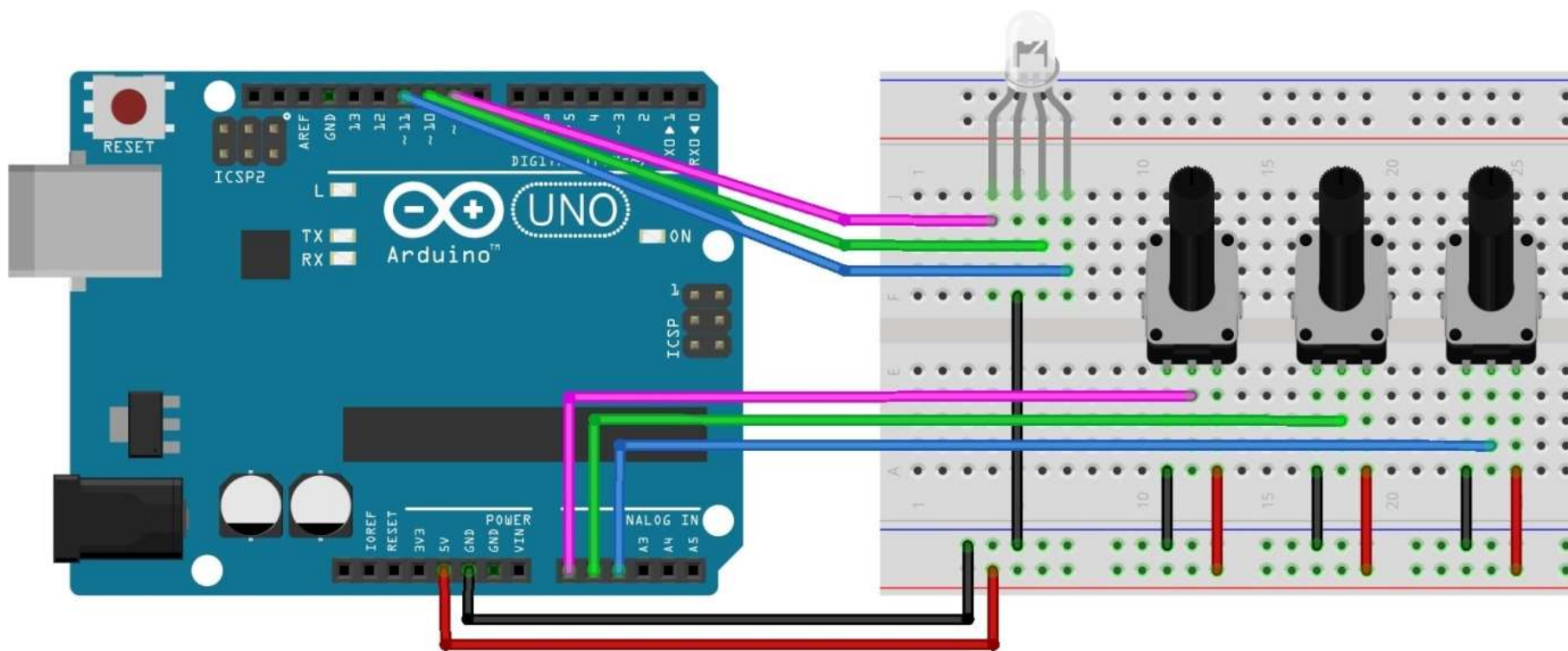
電流控制



負載電流 $I=I_1$

Arduino 類比輸入+PWM輸出

電壓值轉換為0至255用D9, D10, D11以PWM輸出至RGB LED



A0, A1, A2類比讀取可變電阻VR之電壓值(0至1024)

Arduino 類比輸入+PWM輸出

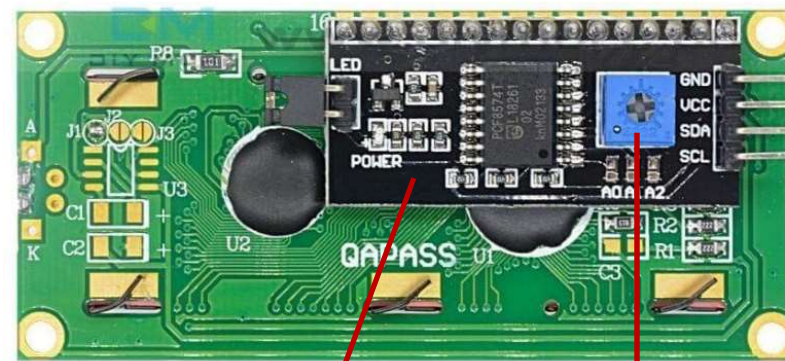
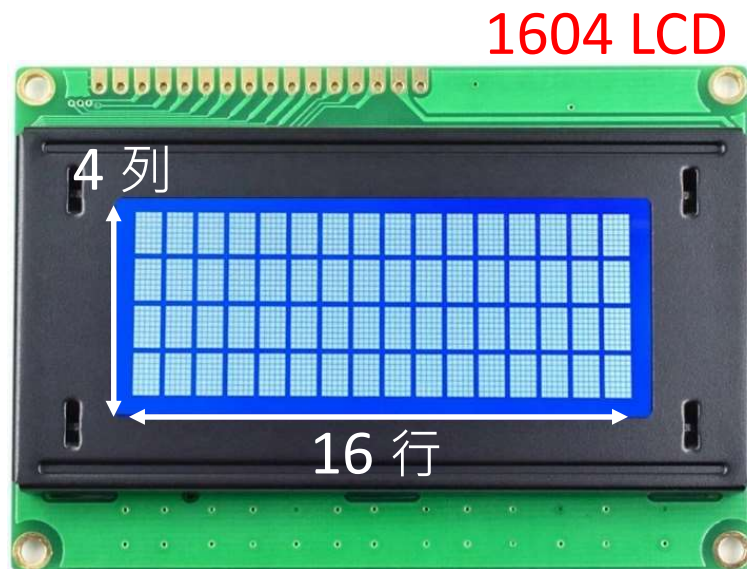
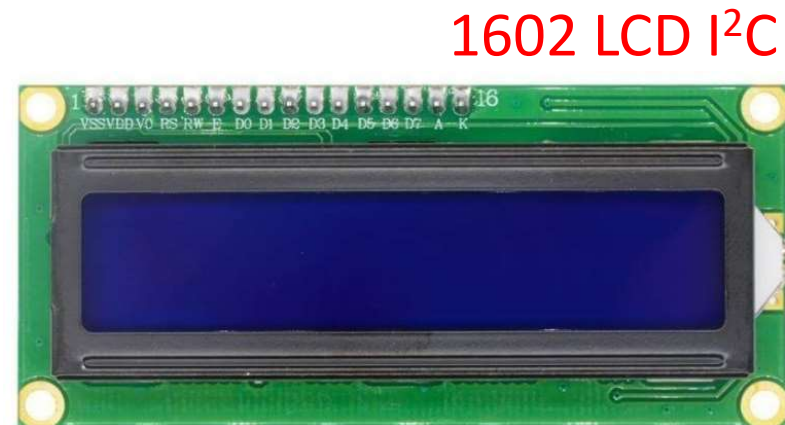
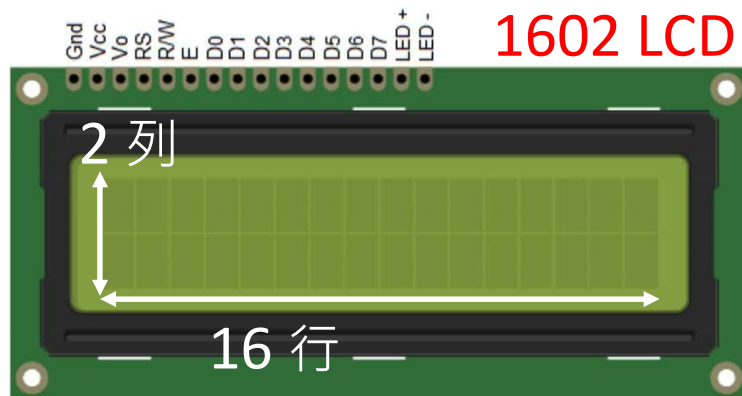
```
int rval = 0; // 宣告rval,gval,bval三個變數
int gval = 0;
int bval = 0;
byte rcolor = 0; // 宣告rcolor,gcolor,bcolor三個變數
byte gcolor = 0;
byte bcolor = 0;
void setup() {
  Serial.begin(9600); // 設定序列埠連線速率
  pinMode(9, OUTPUT); // 設定D9腳位紅光輸出
  pinMode(10, OUTPUT); // 設定D10腳位綠光輸出
  pinMode(11, OUTPUT); // 設定D11腳位藍光輸出
}
void loop() {
  rval = analogRead(A0); // 讀取類比A0,A1,A2可變電阻值，將值存入rval,gval,bval
  gval = analogRead(A1);
  bval = analogRead(A2);
  rcolor = map(rval,0,1023,0,255); // 以map函式，將類比讀取之0~1023，轉成0~255，存入rcolor,gcolor,bcolor
  gcolor = map(gval,0,1023,0,255);
  bcolor = map(bval,0,1023,0,255);
  analogWrite(9, rcolor); // 將rcolor,gcolor,bcolor三個變數值，PWM輸出至D9,D10,D11三個腳位
  analogWrite(10, gcolor);
  analogWrite(11, bcolor);
  Serial.print("R:"); // 螢幕顯示R:
  Serial.print(rcolor); // 螢幕顯示PWM之rcolor值
  Serial.print(" G:"); // 螢幕顯示G:
  Serial.print(gcolor); // 螢幕顯示PWM之gcolor值
  Serial.print(" B:"); // 螢幕顯示B:
  Serial.println(bcolor); // 螢幕顯示PWM之bcolor值，並換行
  delay(200); // 每隔200毫秒更新一次
}
```

增加序列埠顯示PWM數值

→ Serial.begin(連線速率);

→ Serial.print("欲顯示文字");
Serial.print(變數);

液晶螢幕(Liquid Crystal Display)

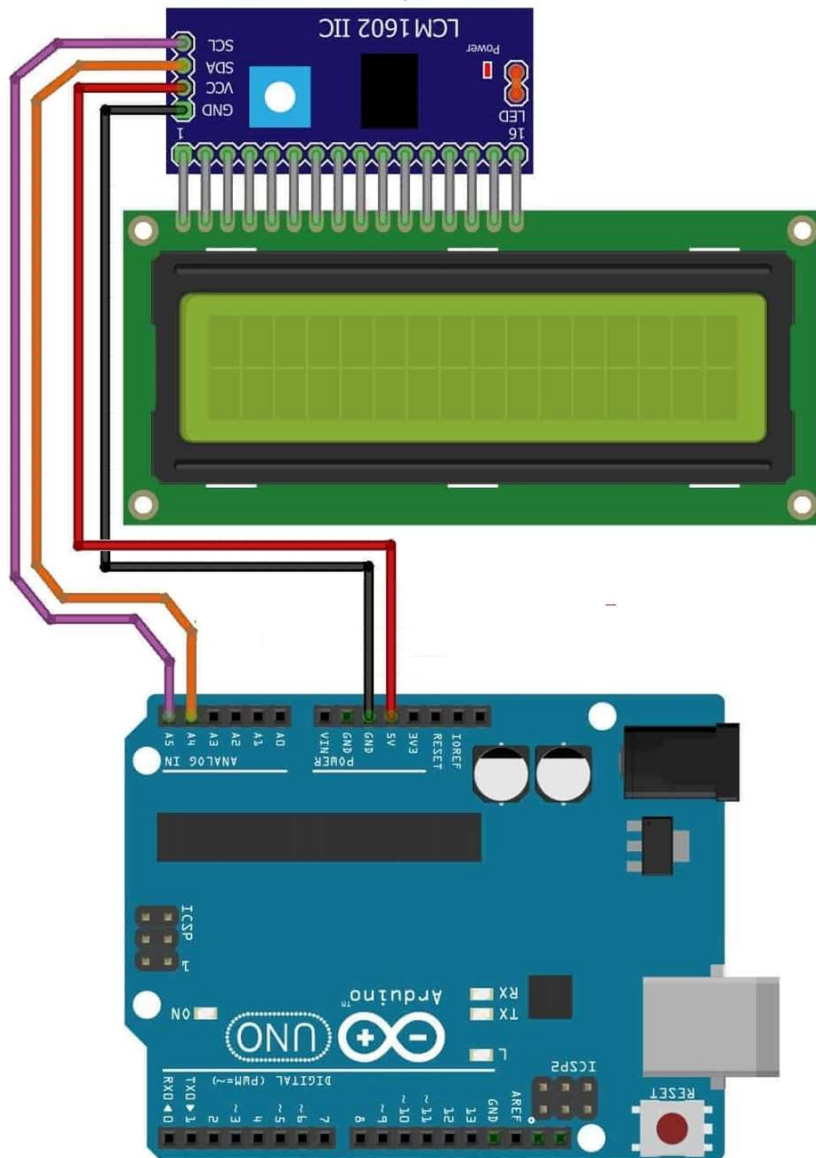


負極
正極
接 A4
接 A5

I²C 模組

亮度調整鈕

Arduino 控制 1602 I²C LCD

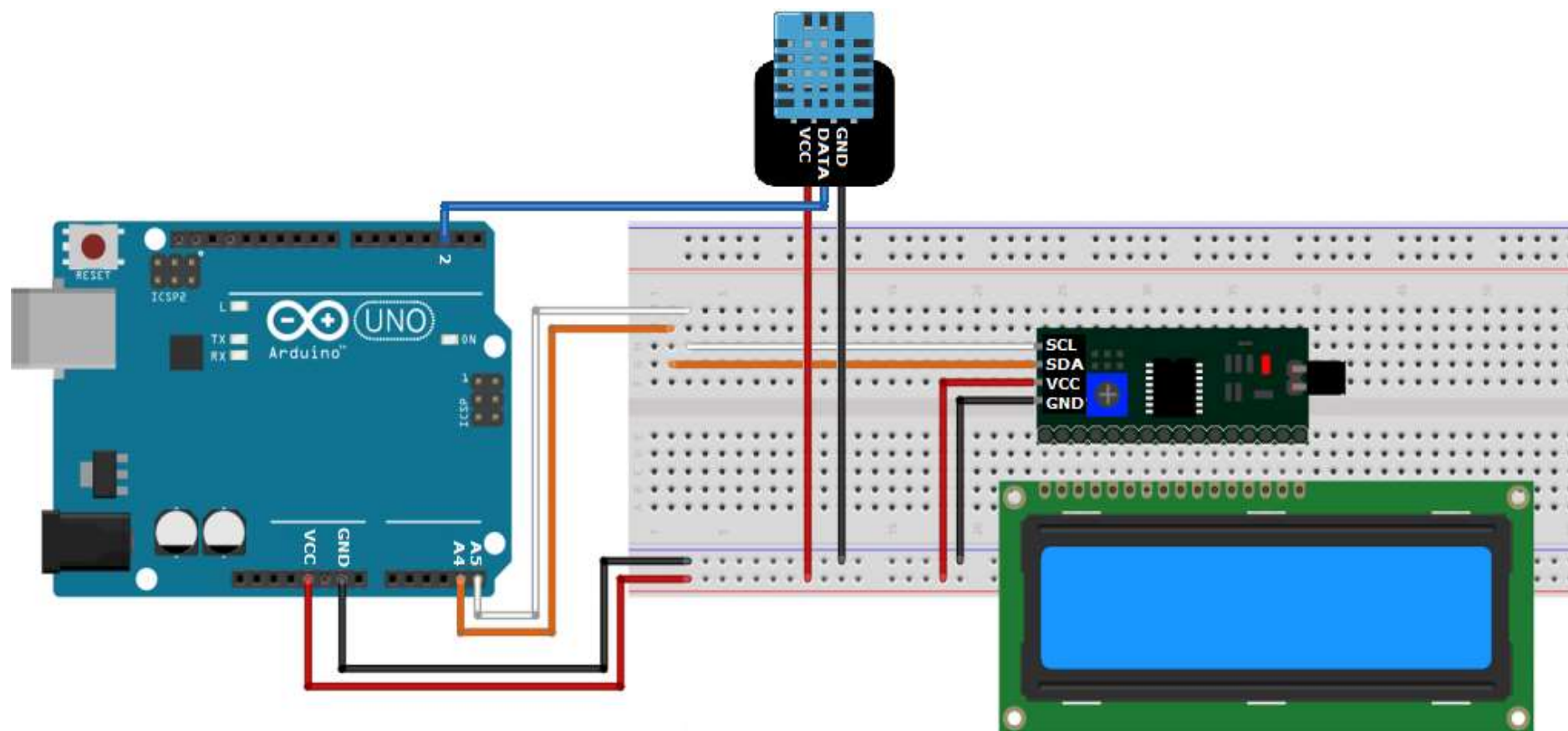


```
#include <Wire.h> // 引用I2C函式庫
#include <LiquidCrystal_I2C.h> // 引用LCD函式庫
LiquidCrystal_I2C lcd(0x27, 16, 2); // 設定 LCD I2C 位址
                                     , SDA接A4, SCL接A5

void setup() {
  lcd.init(); // lcd初始化
  for(int i = 0; i < 3; i++) { // 閃爍3次
    lcd.backlight(); // 開啟背光
    delay(250);
    lcd.noBacklight(); // 關閉背光
    delay(250);
  }
  lcd.backlight();
  lcd.setCursor(0, 0); // 設定游標位置在第一行行首
  lcd.print("HMSH");
  delay(1000);
  lcd.setCursor(0, 1); // 設定游標位置在第二行行首
  lcd.print("Hello, Everybody!");
  delay(3000);
  lcd.clear(); // 清除所有文字
}

void loop() {
  lcd.setCursor(0, 0); // 設定游標位置在第一行行首
  lcd.print("HMSH NO.1");
  lcd.setCursor(0, 1); // 設定游標位置在第二行行首
  lcd.print(millis()/1000); // 顯示啟動至今秒數
}
```

DHT11溫濕度模組 + 1602 LCD 顯示



DHT11 溫濕度模組 + 1602 LCD 顯示

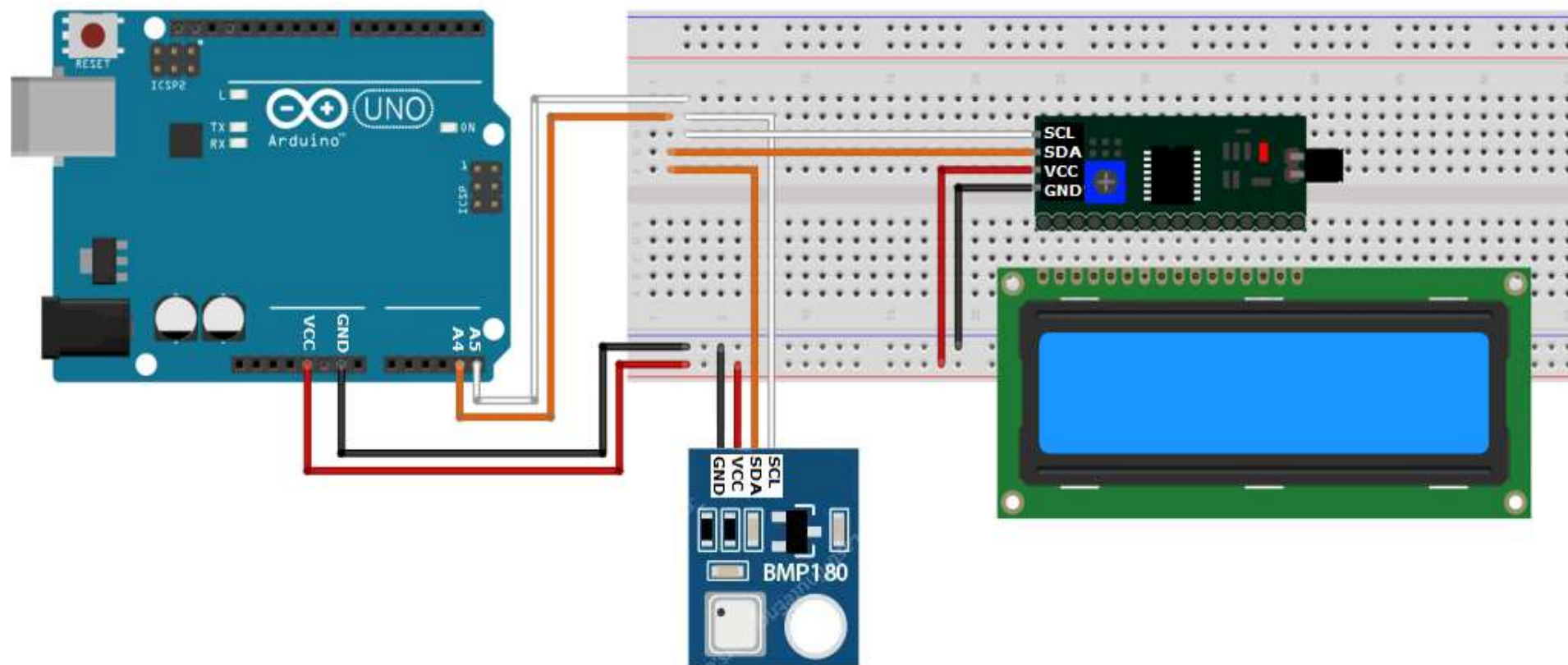
```
#include <Wire.h> // 引用I2C函式庫
#include <LiquidCrystal_I2C.h> // 引用LCD函式庫
#include <DHT.h> // 引用DHT溫溼度函式庫
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // 設定LCD I2C 位址
#define DHTPIN 2 // 設定DHT11於D2接腳輸出
#define DHTTYPE DHT11 // 溫濕度模組型號為DHT 11
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {
  lcd.init(); // lcd初始化
  dht.begin(); // DHT11初始化
  for(int i = 0; i < 3; i++) { // 閃爍3次
    lcd.backlight(); // 開啟背光
    delay(250);
    lcd.noBacklight(); // 關閉背光
    delay(250);
  }
  lcd.backlight();
  lcd.setCursor(0, 0); // 設定游標位置在第一行行首
  lcd.print("DHT 11");
  delay(2000);
  lcd.clear(); // 清除所有文字
}
```

```
void loop() {
  int h = dht.readHumidity(); // 將讀取濕度存入h變數
  int t = dht.readTemperature(); // 將讀取溫度存入t變數
  lcd.setCursor(0, 0); // 設定游標位置在第一行行首
  lcd.print("Temperature");
  lcd.setCursor(12, 0);
  lcd.print(t); // 顯示溫度
  lcd.setCursor(14, 0);
  lcd.print((char)223); // 顯示度
  lcd.print("C");
  lcd.setCursor(0, 1); // 設定游標位置在第二行行首
  lcd.print("Humidity");
  lcd.setCursor(12, 1);
  lcd.print(h); // 顯示濕度
  lcd.setCursor(15, 1);
  lcd.print("%");
}
```


BMP180 氣壓模組 + 1602 LCD 顯示



BMP180 氣壓模組 + 1602 LCD 顯示

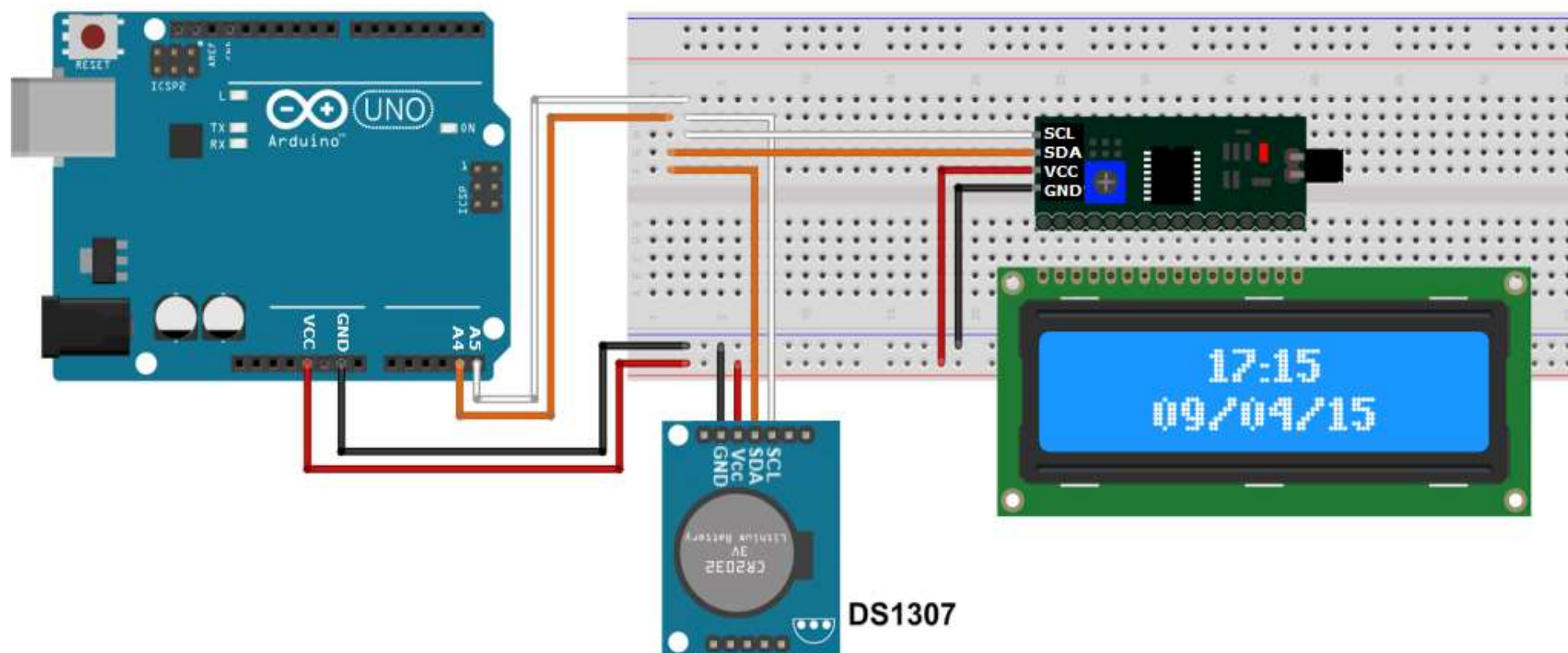
```
#include <Wire.h> // 引用I2C函式庫
#include <LiquidCrystal_I2C.h> // 引用LCD函式庫
#include <BMP180.h> // 引用BMP180函式庫

LiquidCrystal_I2C lcd(0x27, 16, 2); // 設定LCD I2C位址
BMP180 bmp180; // 將讀取結果存入bmp180變數

void setup() {
  lcd.init(); // lcd初始化
  bmp180.init(); // BMP180初始化
  for(int i = 0; i < 3; i++) { // 閃爍3次
    lcd.backlight(); // 開啟背光
    delay(250);
    lcd.noBacklight(); // 關閉背光
    delay(250);
  }
  lcd.backlight();
  lcd.setCursor(0, 0); // 設定游標位置在第一行行首
  lcd.print("BMP180");
  delay(2000);
  lcd.clear(); // 清除所有文字
}
```

```
void loop() {
  float t = bmp180.getTemperature(); // 將讀取溫度存入t變數
  float p = bmp180.getPressure(); // 將讀取壓力存入p變數
  lcd.setCursor(0, 0); // 設定游標位置在第一行行首
  lcd.print("T:");
  lcd.setCursor(5, 0);
  lcd.print(t); // 顯示溫度
  lcd.setCursor(11, 0);
  lcd.print((char)223); // 顯示度
  lcd.print("C");
  lcd.setCursor(0, 1); // 設定游標位置在第二行行首
  lcd.print("p:");
  lcd.setCursor(3, 1);
  lcd.print(p); // 顯示壓力
  lcd.setCursor(11, 1);
  lcd.print("hPa");
  delay(1000);
}
```

DS1307時間模組 + 1602 LCD 顯示




```

#include <Wire.h> // 引用I2C函式庫
#include <LiquidCrystal_I2C.h> // 引用LCD函式庫
LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD I2C 位址，SDA接A4，SCL接A5
const byte DS1307_I2C_ADDRESS = 0x68; // DS1307 I2C 地址
const byte NubberOfFields = 7; // 設定 DS1307 I2C 資料範圍
int y; // 年
byte m, d, w, h, mi, s; // 月/日/週幾/時/分/秒
void setup() {
    lcd.init(); // lcd初始化
    // setTime(23,8,1,4,2,5,0); // 設定時間：2023年8月1日星期四2點5分0秒
    for(int i = 0; i < 3; i++) { // 閃爍3次
        lcd.backlight(); // 開啟背光
        delay(250);
        lcd.noBacklight(); // 關閉背光
        delay(250);
    }
    lcd.backlight();
    lcd.setCursor(0, 0); // 設定游標位置在第一行行首
    lcd.print("DSC1307");
    delay(2000);
    lcd.clear(); // 清除所有文字
}
void loop() {
    getTime(); // 取得時間
    lcd.setCursor(0, 0); // 設定游標位置在第一行行首
    lcd.print("Day ");
    lcd.print(y);
    lcd.print("/");
    lcd.print(m);
    lcd.print("/");
    lcd.print(d);
    lcd.setCursor(0, 1); // 設定游標位置在第二行行首
    lcd.print("Time ");
    lcd.print(h);
    lcd.print(":");
    lcd.print(mi);
    lcd.print(":");
    lcd.print(s);
    delay(1000);
}

```

```

// BCD(二進位) 轉 DEC(十進位)
byte bcdToDec(byte val) {return ((val / 16 * 10) + (val % 16));}
// DEC(十進位) 轉 BCD(二進位)
byte decToBcd(byte val) {return ((val / 10 * 16) + (val % 10));}

```

```

// 設定時間
void setTime(byte y, byte m, byte d, byte w, byte h, byte mi, byte s){
    Wire.beginTransmission(DS1307_I2C_ADDRESS);
    Wire.write(0);
    Wire.write(decToBcd(s));
    Wire.write(decToBcd(mi));
    Wire.write(decToBcd(h));
    Wire.write(decToBcd(w));
    Wire.write(decToBcd(d));
    Wire.write(decToBcd(m));
    Wire.write(decToBcd(y));
    Wire.endTransmission();
}

```

```

// 取得時間
void getTime(){
    Wire.beginTransmission(DS1307_I2C_ADDRESS);
    Wire.write(0);
    Wire.endTransmission();
    Wire.requestFrom(DS1307_I2C_ADDRESS, NubberOfFields);
    s = bcdToDec(Wire.read() & 0x7f);
    mi = bcdToDec(Wire.read());
    h = bcdToDec(Wire.read() & 0x7f);
    w = bcdToDec(Wire.read());
    d = bcdToDec(Wire.read());
    m = bcdToDec(Wire.read());
    y = bcdToDec(Wire.read()) + 2000;
}

```

DS1307時間模組 + 1602 LCD 顯示