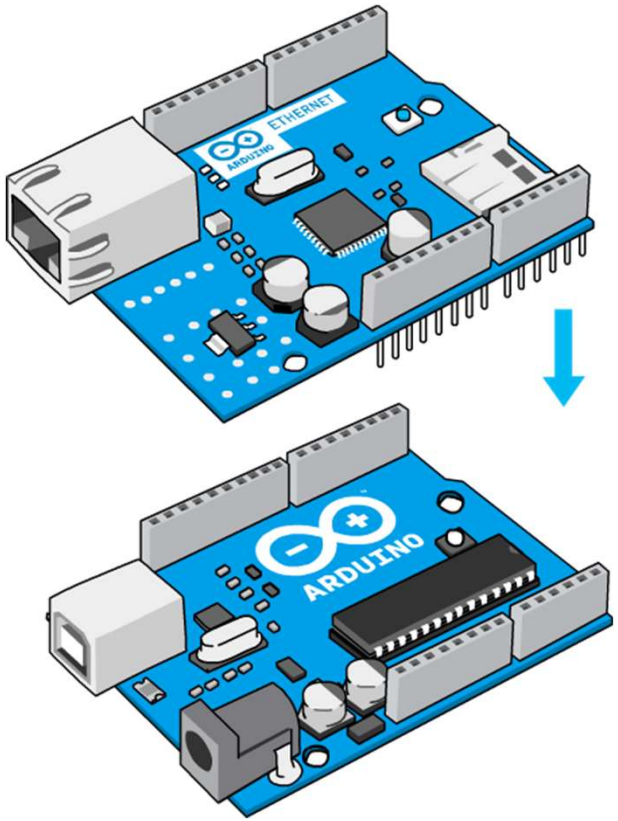
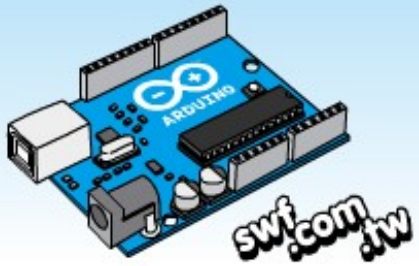
第四章 開關電路與LED跑馬燈

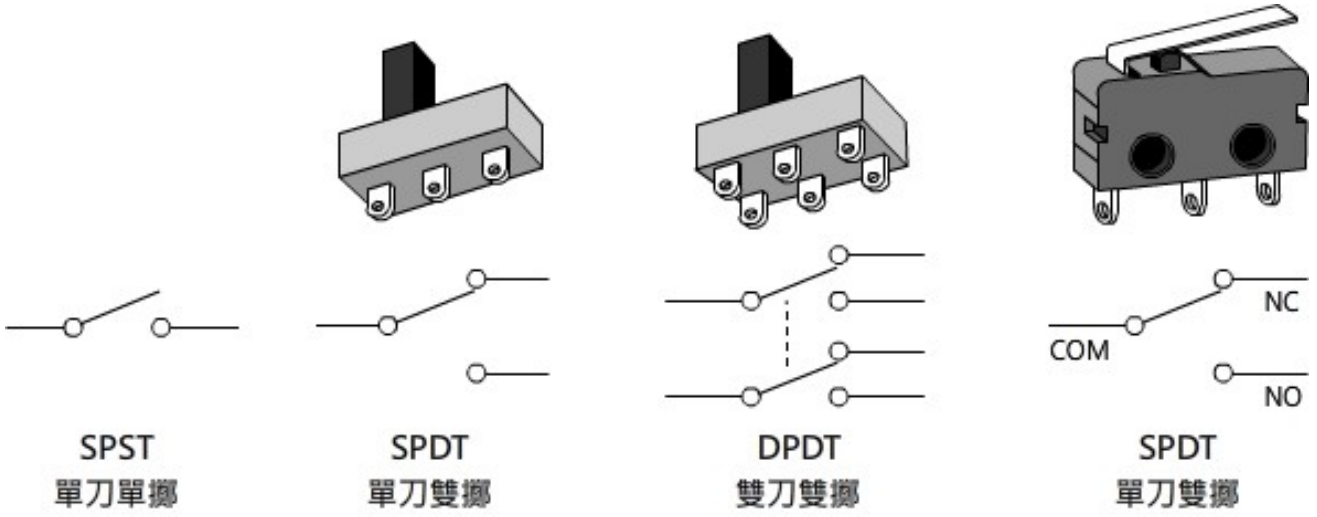


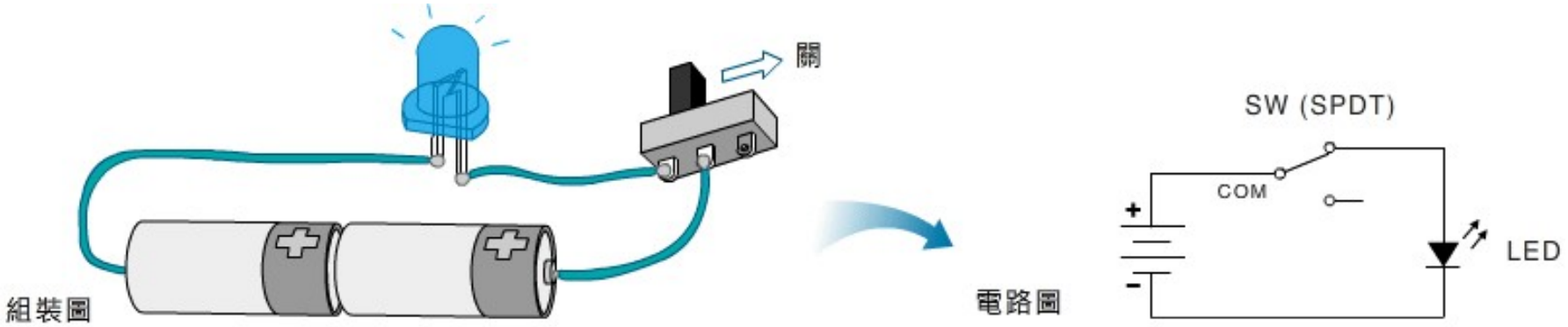
- 認識開關
- 讀取數位輸入值
- if條件式
- 撰寫迴圈程式
- 認識陣列
- 認識埠口與位移操作

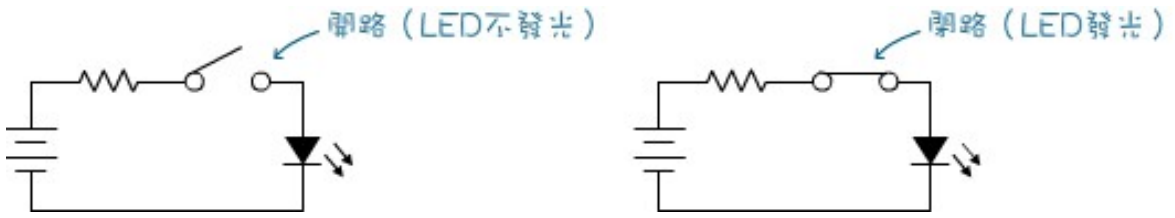


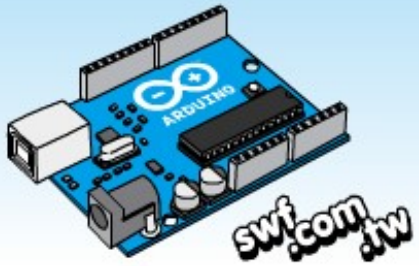
認識開關

開關的外觀與電路符號



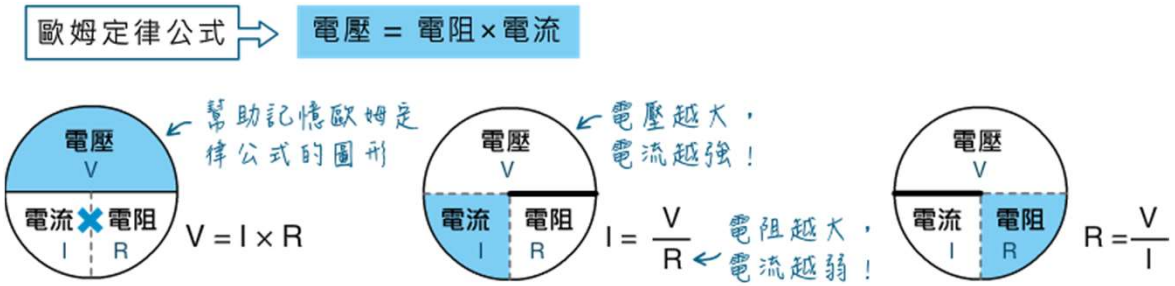




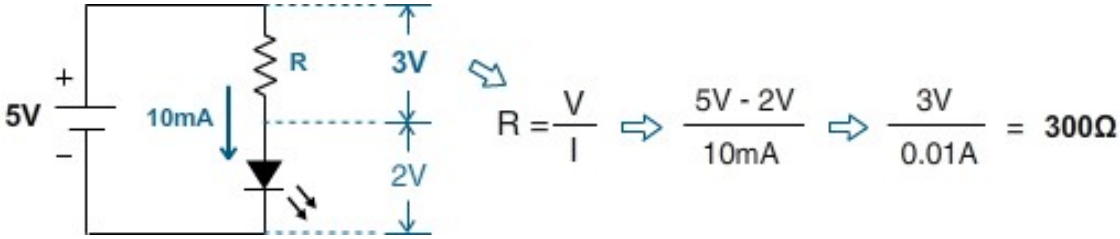


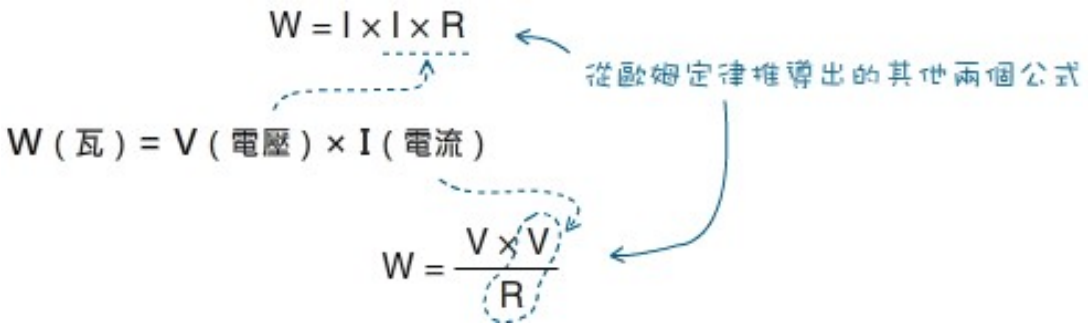
歐姆定律

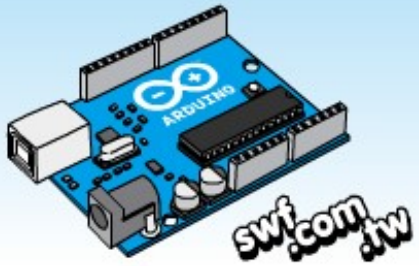
電路中的電壓、電流和電阻之間的關係，可以用歐姆定理表示：**電流和電壓成正比，和電阻成反比。**



透過歐姆定律，可求出電路所需的限流電阻值，以及消耗功率。





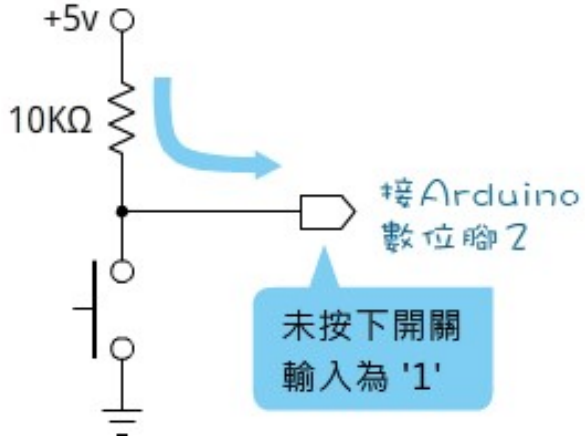


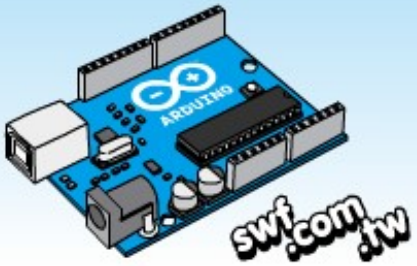
讀取數位輸入值

- 讀取數位輸入值的語法：

```
boolean 變數名稱 = digitalRead(接腳編號);
```

- 電路的接法：



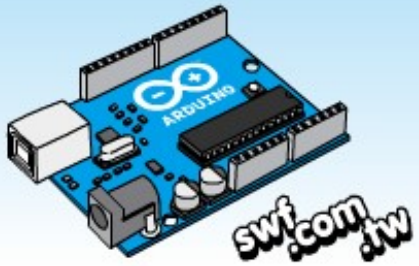


比較運算子

比較運算子	說明
==	如果兩者相等則成立
!=	如果不相等則成立
<	如果左邊小於右邊則成立
>	如果左邊大於右邊則成立
<=	如果左邊小於或等於右邊則成立
>=	如果左邊大於或等於右邊則成立

條件式當中的且、或和反相測試：

名稱	運算符號	運算式	說明
且（AND）	&&	A && B	只有A和B兩個值都成立時，整個條件才算成立。
或（OR）	||	A || B	只要A或B任何一方成立，整個條件就算成立。
反相（NOT）	!	!A	把成立的變為不成立；不成立的變為成立。

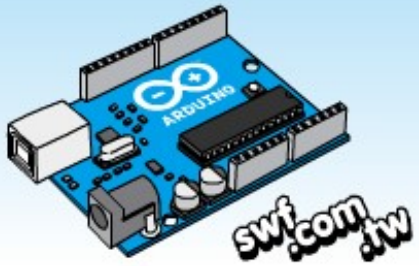


指定運算子

運算子	意義	說明
++	遞增	變數加1
--	遞減	變數減1
+=	指定增加	
-=	指定減少	
*=	指定相乘	
/=	指定相除	

以下三種敘述，意義相同：

```
a = a+1;
a ++;
a += 1;
```

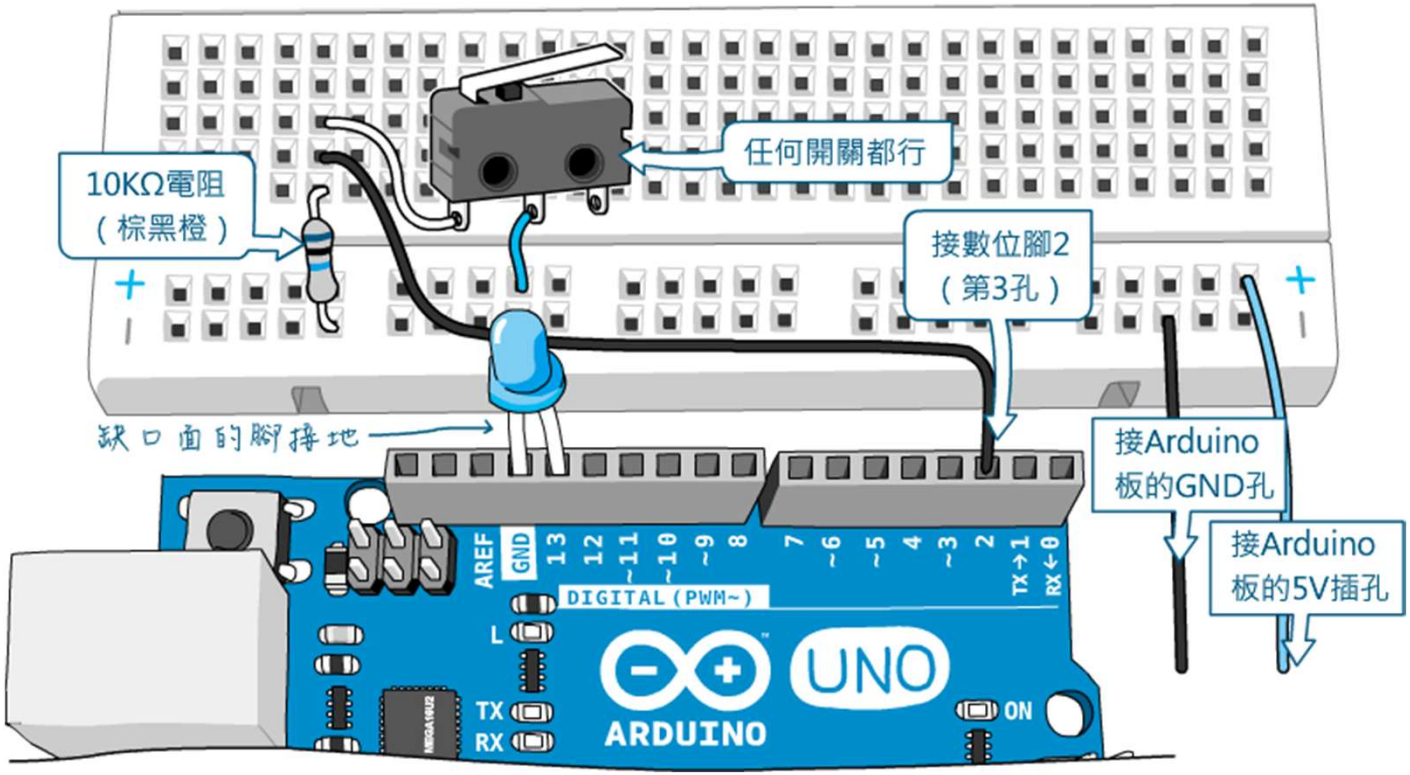


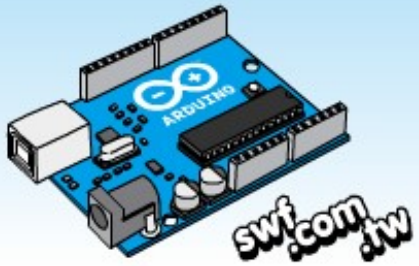
數學運算子

運算子	意義	說明
+	加	
-	減	
*	相乘	
/	相除	得到商值
%	餘數	得到餘數

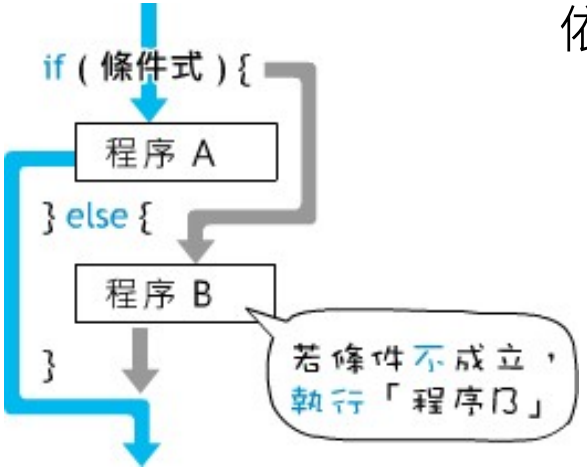
7 / 5　>>>>得到 1，取整數
7 % 5　>>>>得到 2

開關實驗麵包板電路

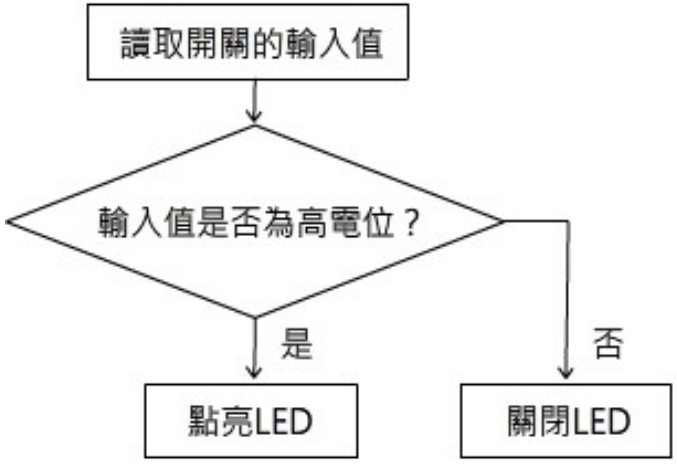

10KΩ電阻
（棕黑橙）
任何開關都行
接數位腳2
（第3孔）
缺口面的腳接地
接Arduino
板的GND孔
接Arduino
板的5V插孔
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON



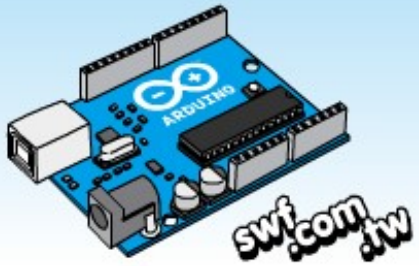
條件判斷式敘述



依據開關的狀態，點亮或關閉LED。

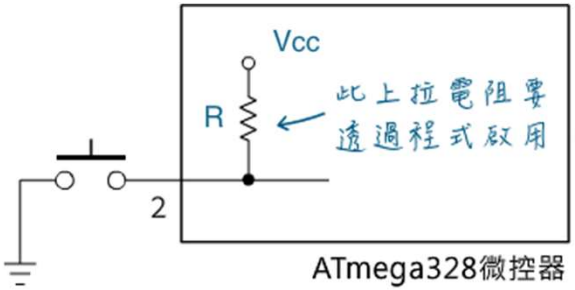



```
const byte LED = 13;
const byte SW = 2;

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(SW, INPUT);
}

void loop(){

   boolean val = digitalRead(2);

   if (val){
     digitalWrite(LED, HIGH);
   } else {
     digitalWrite(LED, LOW);
   }
}
```

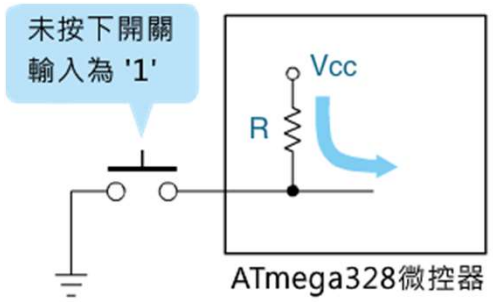


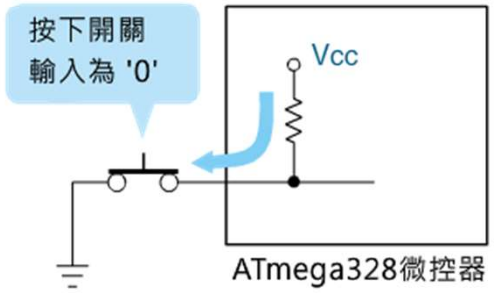
啟用上拉電阻

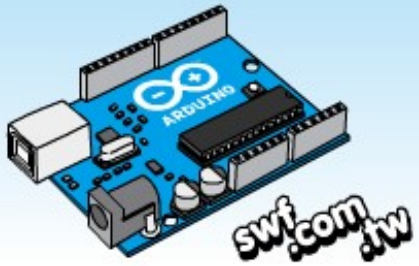
ATmeg328微控器的數位接腳有內建上拉電阻，可透過底下的敘述啟用，啟用內建的上拉電阻後，開關電路就能省略外接電阻。




```
/* 啟用第2腳內部的「上拉電阻」 */
pinMode(2, INPUT_PULLUP);
```

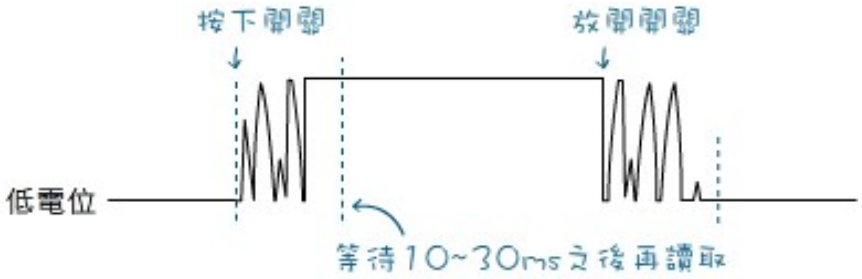




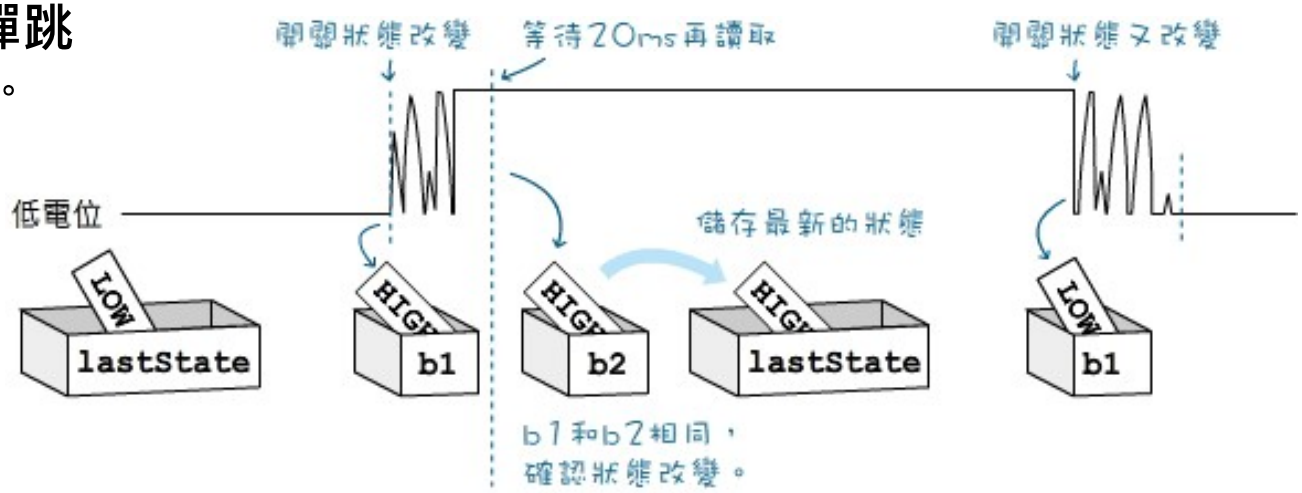


消除開關的彈跳訊號

機械式開關在切換的過程中，電子信號並非立即從0變成1（或從1變成0），而會經過短暫的，忽高忽低變化的彈跳現象。

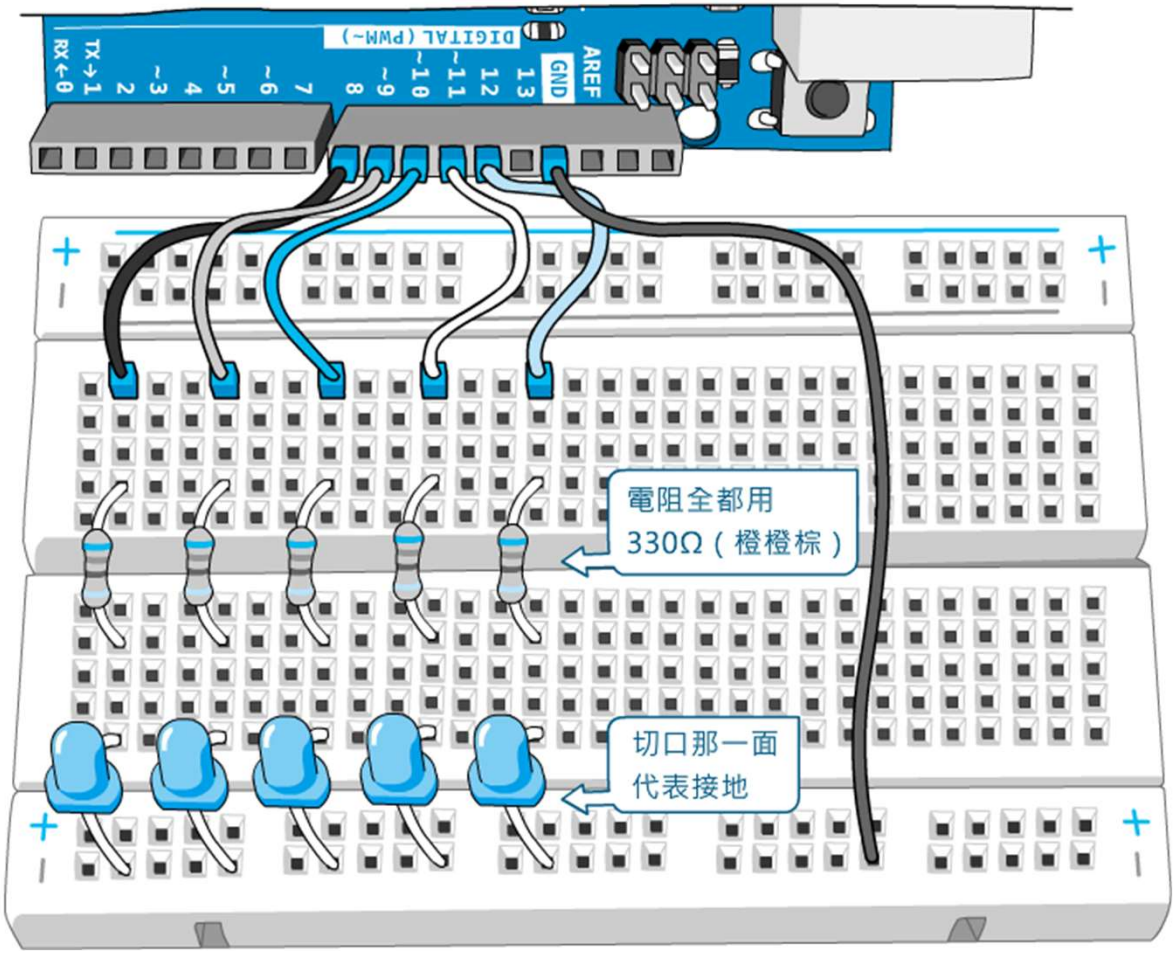


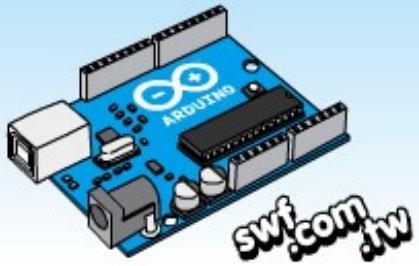
為了避免上述狀況，程式（或者硬體）需要加入所謂的**消除彈跳（de-bouncing）**處理機制。



LED跑馬燈電路

輪流點亮一個LED



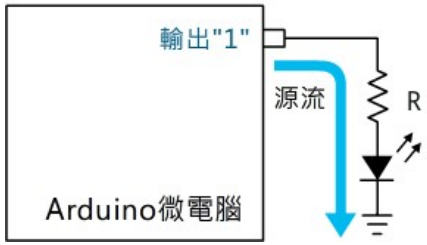


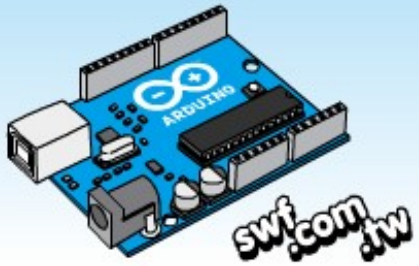
LED跑馬燈程式一

- 輪流點亮一個LED的程式碼

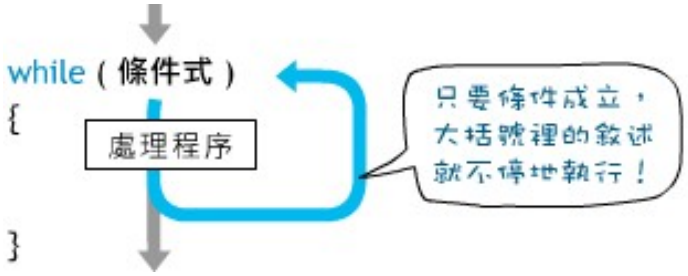
```
void loop() {
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  delay(100);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, LOW);
  delay(100);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, HIGH);
  delay(100);
}
```

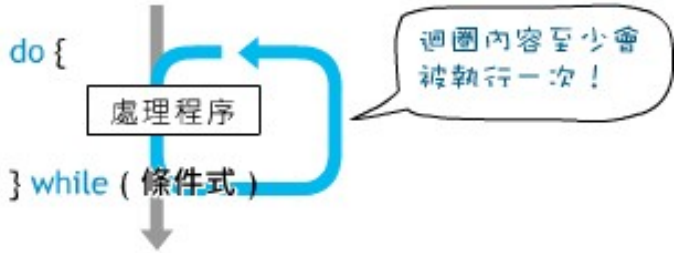
點亮第一個LED



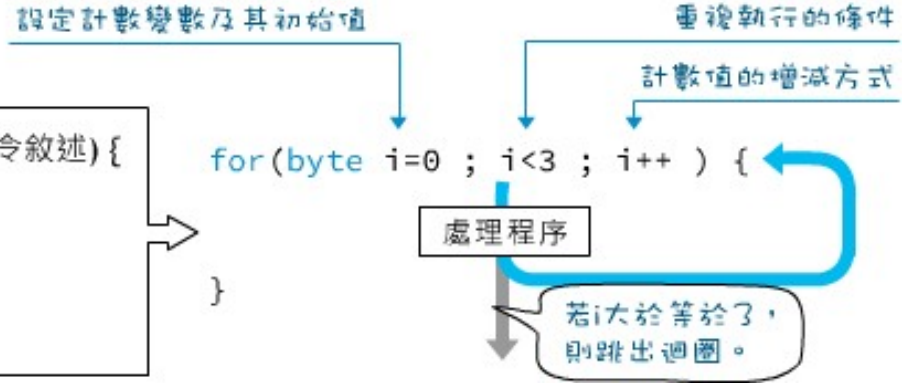


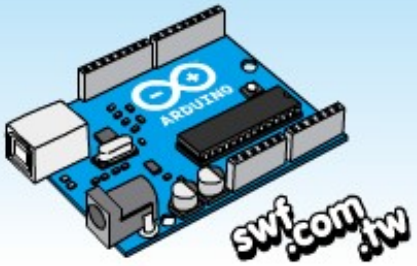
迴圈控制指令






```
for (初始變數；條件式；指令敘述){
  指令敘述1
  指令敘述2
    :
}
```



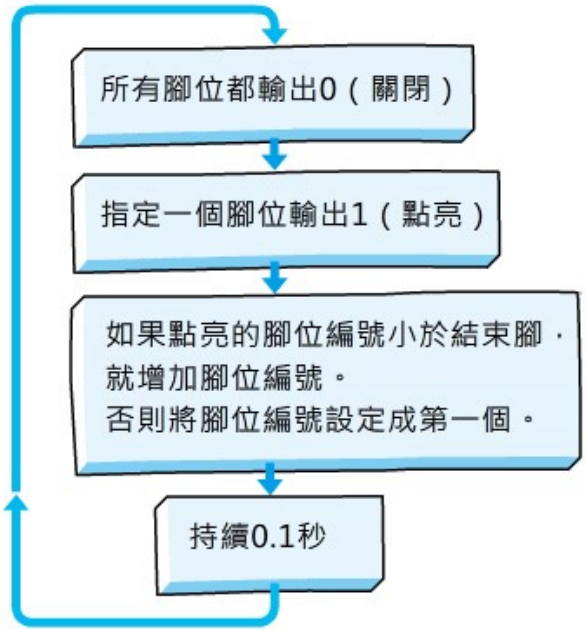


使用for迴圈的跑馬燈程式

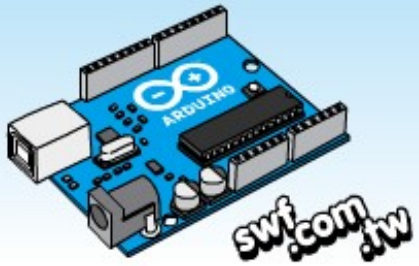
```
void setup() {
  for (byte i = startPin; i <= endPin; i++) {
    pinMode(i, OUTPUT);
  }
}
```

起始腳位 ↓ startPin　　結束腳位 ↓ endPin

主程式迴圈




```
void loop() {

  for (byte i=startPin; i<=endPin; i++) {
    digitalWrite(i, LOW);
  }

  digitalWrite(lightPin, HIGH);

  if (lightPin < endPin) {
    lightPin ++;
  } else {
    lightPin = startPin;
  }

  delay(100);
}
```

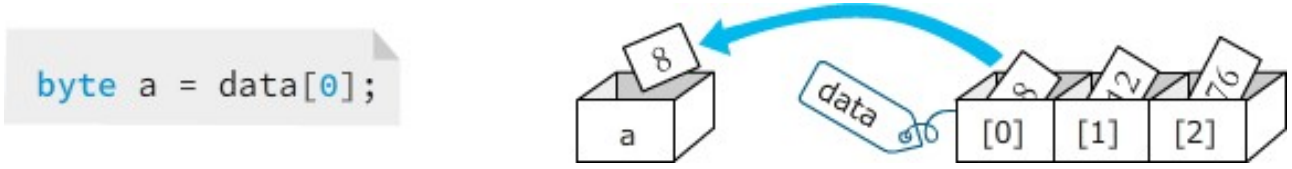


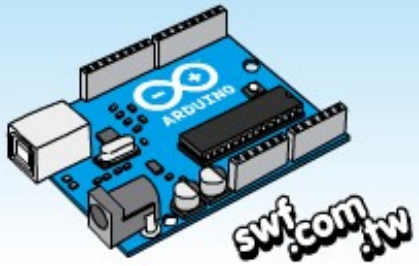
陣列變數

陣列（array）變數可以存放很多不同值，就像具有不同分隔空間的盒子一樣。

宣告陣列的同時可一併設定其值

讀取陣列元素



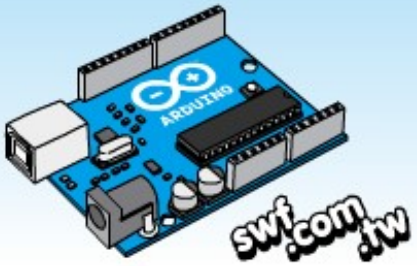


使用陣列的跑馬燈程式

```
const byte LEDs[] = {8,9,10,11,12};
const byte total = sizeof(LEDs);
byte index = 0;

void setup() {
  for (byte i=0; i<total; i++) {
    pinMode(LEDs[i], OUTPUT);
  }
}

void loop() {
  for (byte i=0; i<total; i++) {
    digitalWrite(LEDs[i], LOW);
  }
  digitalWrite(LEDs[index], HIGH);
  if (index < total) {
    index ++;
  } else {
    index = 0;
  }
  delay(100);
}
```

i值將是0~4，依序讀取出8, 9, 10, 11和12元素值。

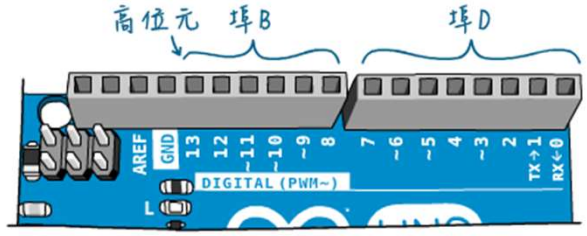
在指定腳位輸出「高電位」



埠口操作與位移指令

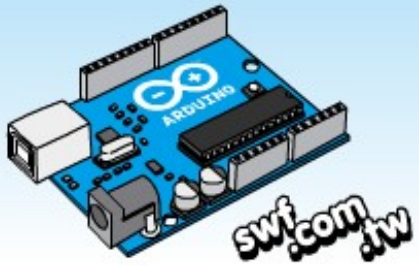
ATMega328處理器，具有D, B兩個數位輸出/輸入埠，以及一個類比埠C。

埠口指令可用一行敘述，將多個腳位設定成輸出或輸入。



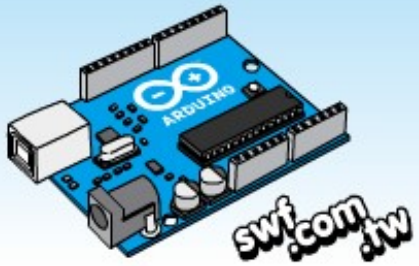
位移（shift）運算子可以將資料裡的所有位元向右或向左移動，空缺的部分補上0。

位移運算	位移結果（二進位）	結果（十進位）	
1	00000001	1	
1 << 1	00000010	2	相當於乘2
1 << 2	00000100	4	相當於乘4
12 >> 0	00001100	12	
12 >> 1	00000110	6	相當於除2
12 >> 2	00000011	3	相當於除4
12 >> 3	00000001	1	



使用埠口與位移製作跑馬燈

```
byte data = B00001;
byte shift = 0;
byte max = 5;

void setup(){
 DDRB = B011111;   // 8~12腳設成輸出。
}

void loop() {
  PORTB = data << shift;   ← 資料往左移之後，輸出到埠口。
  shift ++;
  if (shift == max) {      ← 如果位移等於最大值，就重設為0。
    shift = 0;
  }
  delay(100);
}
```



- 一分鐘回饋:
- https://goo.gl/forms/0C6jWOW5MTX9paos1