



Chap 5

Arduino的類比I/O

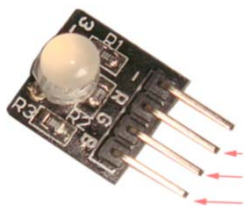
使用的 I/O 裝置



B50k 可變電阻



LED 燈

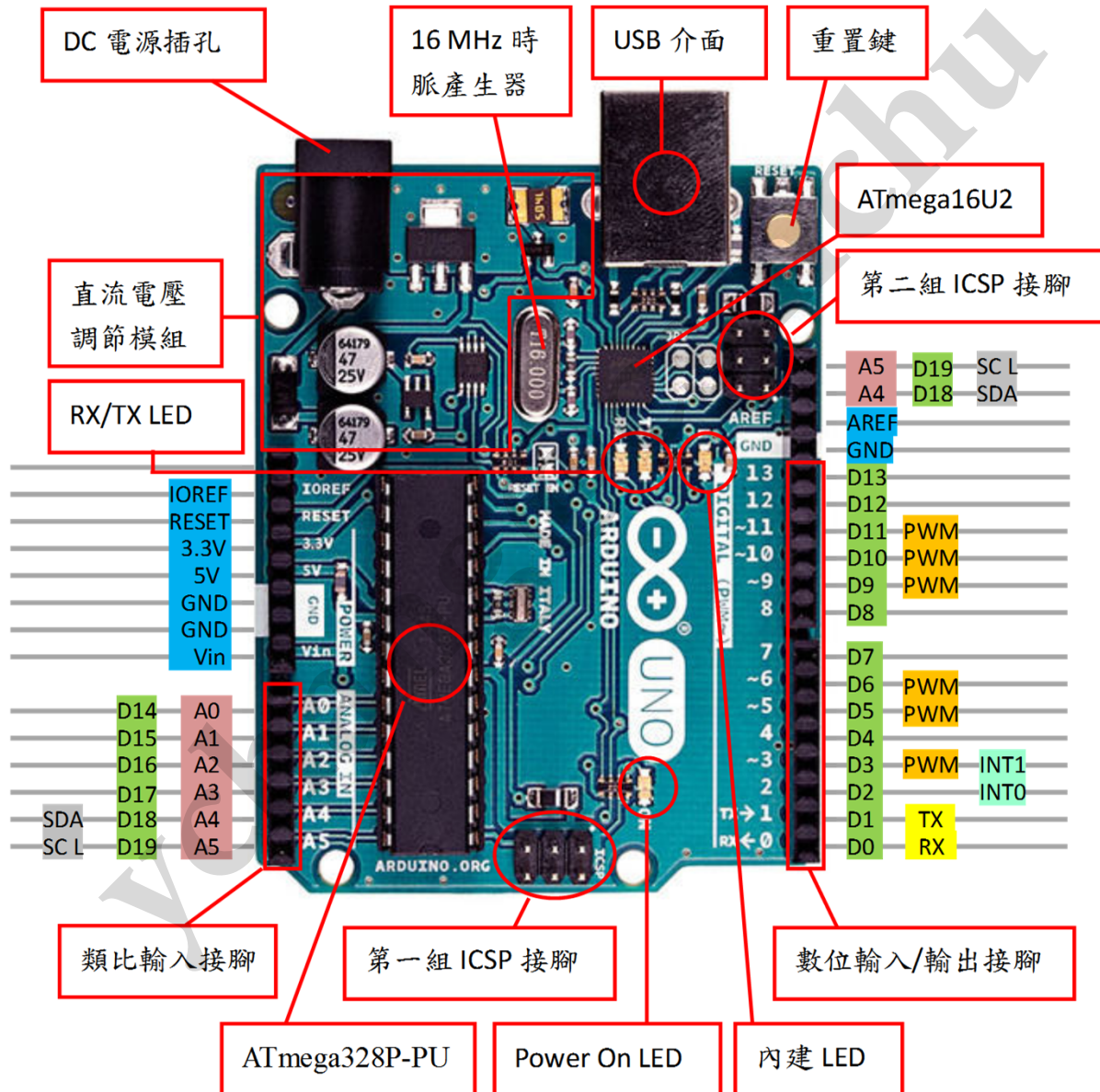
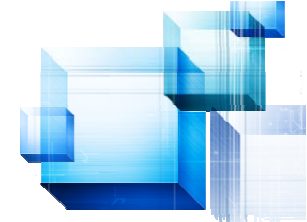


RGB LED 模組

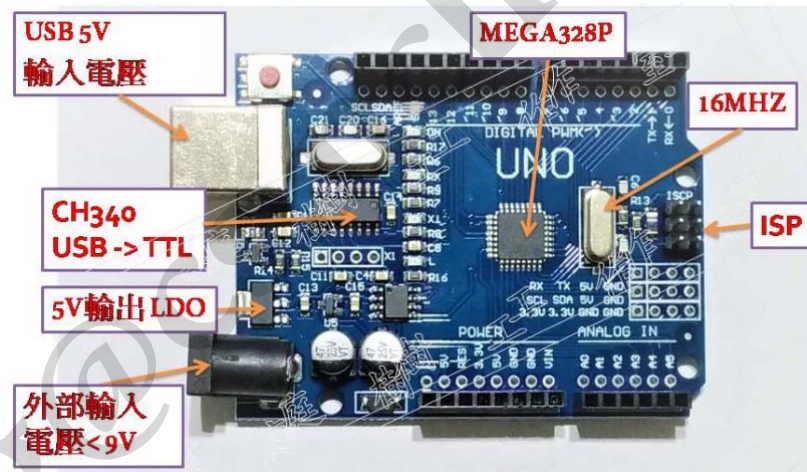
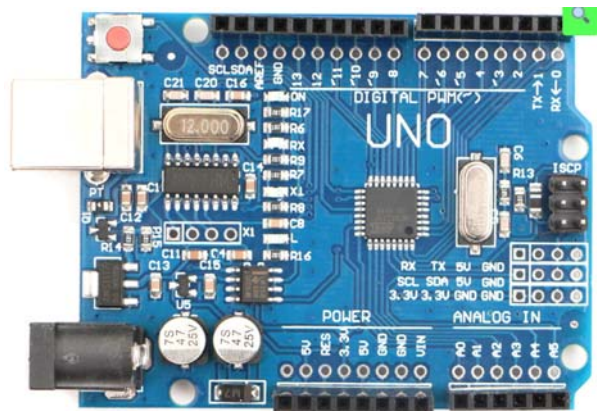


按鈕開關

Arduino UNO

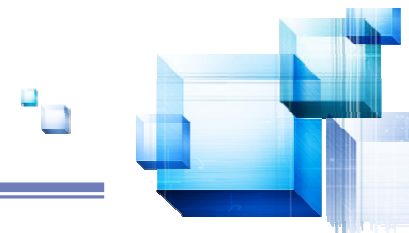


副廠Arduino Uno R3 經濟版



- 副廠Arduino Uno R3 經濟版，為了降低成本，ATMEGA328P晶片改為貼片封裝，ATmega16U2 改以 CH340G USB 晶片取代。並與原始Arduino Uno R3 100%相容。

基本 I/O 函式



● 基本 I/O 函式

- ▶ pinMode()
- ▶ digitalRead()
- ▶ digitalWrite()
- ▶ analogRead()
- ▶ analogReference()



5.1 analogRead()



【描述】讀取類比輸入接腳的訊號值

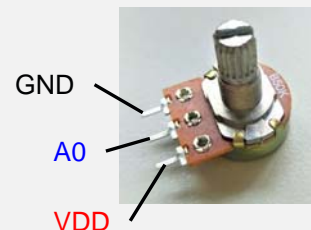
【語法】analogRead(pin)

【參數】pin：類比接腳的編號，以UNO而言，有A0～A5總共6支類比I/O接腳。

【傳回值】0~1023的整數值

【範例】

```
1 //讀取可變電阻(電位器)的輸出值
2 const int potPin=A0;
3 void setup() {
4     pinMode(potPin, INPUT);
5     Serial.begin(9600);
6 }
7
8 void loop() {
9     int val;
10    float volt;
11    val=analogRead(potPin); //讀取 A0，傳回 0-1023
12    volt=val*0.00488;      //轉換成電壓值 5v/1024=0.00488v
13    Serial.print(val); Serial.print(" => "); //印出結果
14    Serial.print(volt, 1); //小數點 1 位
15    Serial.println(' V');
16    delay(1000);          //延遲 1 秒
17 }
```

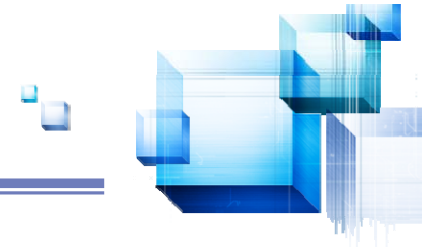


analogRead()



【說明】

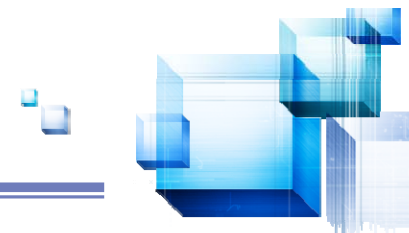
- (1) 類比輸入接腳編號A0～A5是Arduino預設的常數，其數值對應為14～19。
- (2) 因為 Arduino開發板上的類比數位轉換器（ADC）是10-bit，代表0~5V的電壓會對應到0~1023個整數值，所以解析度等於 $5V/1024$ ，約為每單位4.9mV。
- (3) 根據官網說明，Arduino讀取類比訊號的時間，一次讀取大約是100微秒（microsecond），所以最高的讀取頻率為每秒10000次。
- (4) 如果類比輸入接腳沒有連接任何裝置，則analogRead()的傳回值會是外在因素所造成的波動值，例如其他類比訊號的輸入、或是手靠近板子的距離遠近等。
- (5) 如果數位接腳不夠用，A0～A5也可以改設定為數位輸出，例如 `pinMode(A0, OUTPUT)`; 然後可以使用 `digitalWrite(A0, HIGH/LOW)`; 寫出資料，但是這樣混合使用，在讀取其他類比腳位時會產生雜訊干擾，建議還是盡量避免。



練習 5-1

ychang@cs.nchu

5.2 analogWrite()



【描述】 使用duty cycle的值，將PWM的訊號波形輸出到指定的接腳

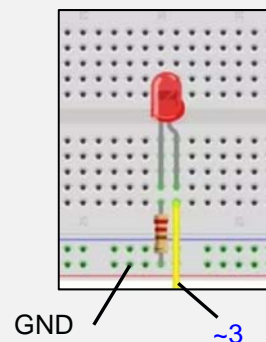
【語法】 analogWrite(pin, value)

【參數】 pin：接腳的編號，以UNO而言，有3、5、6、9、10、11。
value：0~255的整數值，可表示PWM的duty cycle。

【傳回值】 無

【範例】

```
1 //LED 燈的亮度控制
2 const int ledPin=3;    //LED 輸出接腳
3 void setup() {
4     pinMode(ledPin, OUTPUT);
5     Serial.begin(9600);
6 }
7
8 void loop() {
9     int val;
10    for (val=0; val<256; val=val+15) { //漸亮
11        analogWrite(ledPin, val);
12        delay(100); //間隔 0.2 秒
13    }
14    for (val=255; val>=0; val=val-15) { //漸暗
15        analogWrite(ledPin, val);
16        delay(100); //間隔 0.2 秒
17    }
18 }
```



analogWrite()



【說明】

- (1) digitalWrite()只可輸出0V或5V二種電壓值，相較之下analogWrite()可以輸出0~5V中的任一電壓值，達到類比輸出的效果。
- (2) Duty cycle（佔空比）表示在一個周期內，工作時間佔整個週期時間的比值，假設value=128，則duty cycle=128/255約為50%，輸出電壓=5Vx50%=2.5V。
- (3) 在大部分的Arduino開發板，只要微控晶片是ATmega168或是ATmega328P系列的，包含UNO（請參考圖2.7），analogWrite()都可以使用在接腳3、5、6、9、10、11正常工作。
- (4) 執行analogWrite()之後，指定的接腳就會持續輸出一個穩定的方波，其值由第二個參數value來決定，直到在相同的接腳執行到下一個analogWrite()，digitalRead()或是digitalWrite()時才會停止。

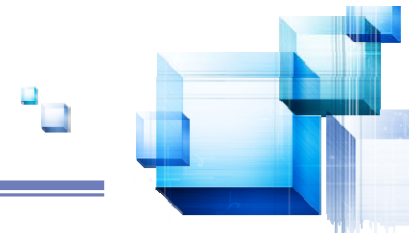
5.3 PWM 簡介



8.1. 什麼是脈衝寬度調變？

脈衝寬度調變的原文全名是 Pulse Width Modulation，縮寫為 PWM，簡單來說它是一種使用數位訊號模擬產生類比訊號的技術，舉例說明，一個數位系統只會有高電位（5V）與低電位（0V）二種訊號的輸出，在這種限制下我們如何實現 1V，2V，甚至是 3.5V 的輸出？答案當然是肯定的，其關鍵技術就是 PWM，實際上只要適當的控制，我們可以產生 0~5V 之間任一點電壓值，PWM 的技術特色，就是使數位系統具有類比電路的行為，但是卻沒有類比電路功耗大和對雜訊干擾敏感的缺點。脈衝寬度調變 PWM 的技術可產生 0%~100% 的類比電壓，常應用在燈光或螢幕亮度的控制、馬達轉速的控制、音量的大小聲/聲音高低的控制...等。

學習完這章，你不再只是會使用簡單的 `analogWrite()` 函式，你可獲得比 `analogWrite()` 函式更多的控制權，能實做出更複雜更精準的 PWM 輸出控制，成為 PWM 達人。



8.2. PWM 的工作原理

在介紹 PWM 的工作原理之前，針對週期性的訊號方波，如圖 8.2.1 所示，我們必須先了解以下基本名詞的定義。

(1) 週期 (T)

通常以符號 T 表示，其定義為一個方波從開始到結束所經過的時間，以秒為單位。

(2) 頻率 (f)

1 秒內產生方波的數量，單位為赫茲 Hz，例如：一個頻率為 10kHz 的時脈產生器，代表

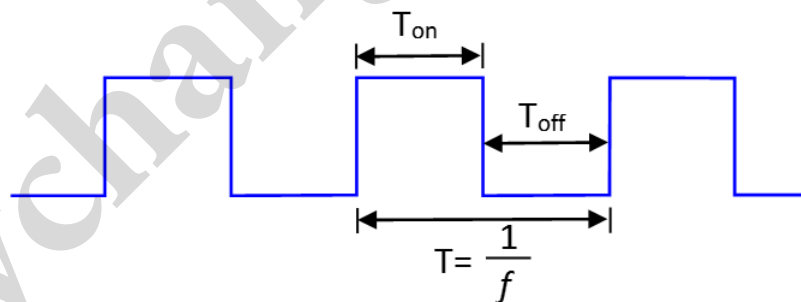


圖 8.2.1 週期性訊號方波



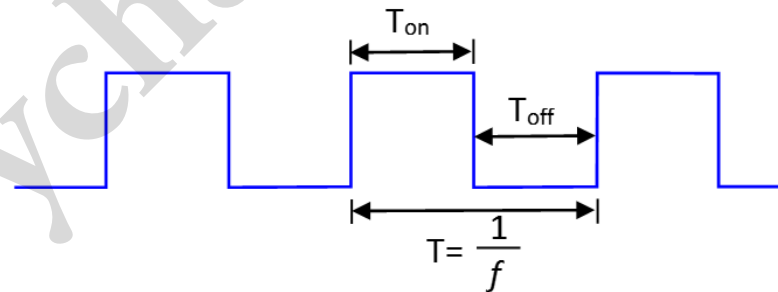
1 秒內會產生 10k 個時脈週期。根據定義我們可推得週期與頻率的關係為 $f = \frac{1}{T}$ 或是 $T = \frac{1}{f}$ ，只要知道週期就可推得頻率，反之亦然。所以頻率為 10kHz，可推得其週期為 $1/10k=0.1ms$ 。

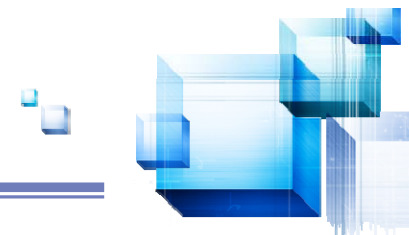
(3) T_{on}/T_{off}

T_{on} 代表一個週期內訊號為高電位的時間， T_{off} 則為低電位的時間，以正向邏輯而言，我們可以把 T_{on} 當成啟動或是工作的時間，而 T_{off} 則為休息的時間。

(4) Duty Cycle，工作占比，也有人稱為占空比或工作週期。

定義為 T_{on}/T 的比例，也就是指真正工作的時間占整個週期的百分比，例如：一個訊號方波，其週期固定為 10ms， T_{on} 為 3ms，則 $duty\ cycle=3ms/10ms=30\%$ 。



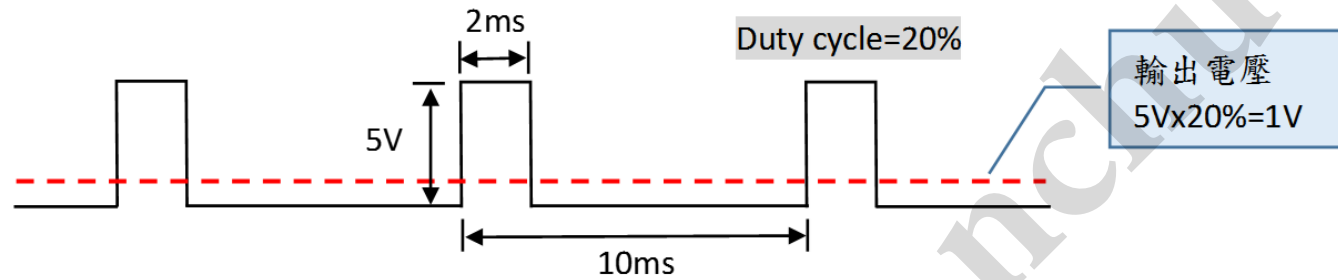


要解釋 PWM 的工作原理，以亮燈的例子來說明比較容易了解，如果我們用開關來控制燈光，(1)開 1 秒關 1 秒，燈光就會跟著亮 1 秒暗 1 秒，可是當開關的速度加快到很高速的時候，我們的眼睛就會看不到燈光一亮一暗的閃爍，取而代之的是昏暗半亮的持續燈光，(2)如果我們改變開關時間的比例，變成開 1 秒關 2 秒，燈光就會跟著亮 1 秒暗 2 秒，比例不變只要開關速度持續加快，最後就會變成 1/3 亮的持續燈光，這就是 PWM 的作用。

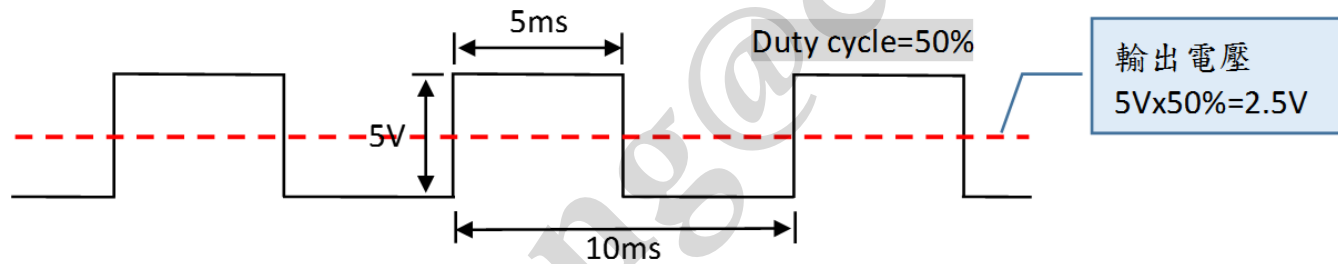
從以上例子的說明，其實脈衝寬度調變 PWM 的名稱就是源自對 T_{on} 脈衝寬度的調整，若訊號週期固定不變，只要改變 T_{on} 的寬度（也就是 T_{on} 的時間），就能改變 duty cycle 得到我們預先規劃的訊號準位，PWM 輸出電壓的公式如下：

$$\text{PWM 輸出電壓} = V_{DD} \times \text{Duty Cycle} = V_{DD} \times \frac{T_{ON}}{T}$$

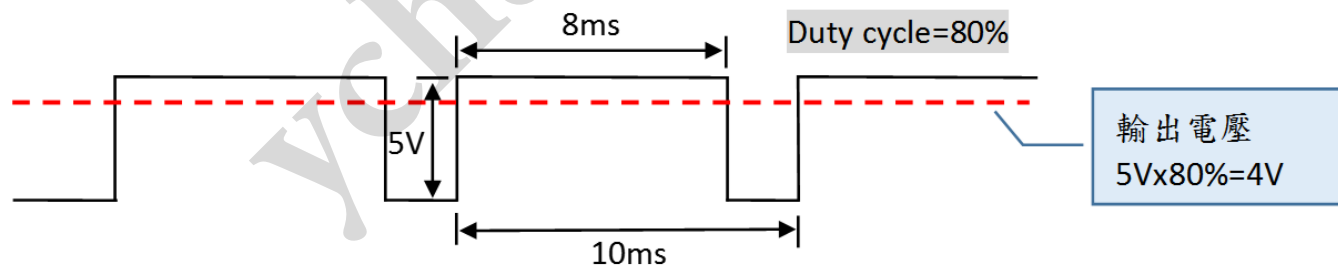
圖 8.2.2 的範例可充分說明 PWM 的工作原理，在範例(a)中，因為 duty cycle=20%，所以輸出電壓=5Vx20%=1V，同理，在範例(b)跟(c)中，我們可藉著增加 duty cycle，進而把電壓調高到 2.5V 與 4V，以符合需求。



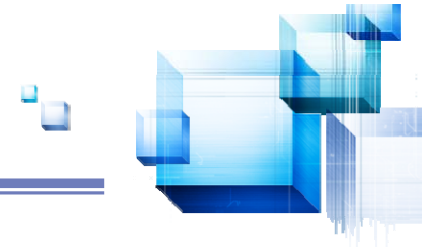
(a) Duty cycle=20%



(b) Duty cycle=50%



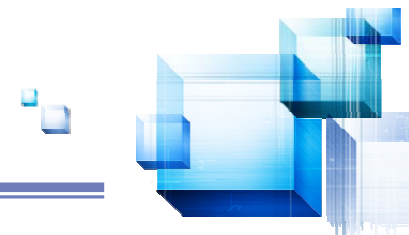
(c) Duty cycle=80%



練習 5-2

ychang@cs.nchu

5.4 全彩RGB LED 模組

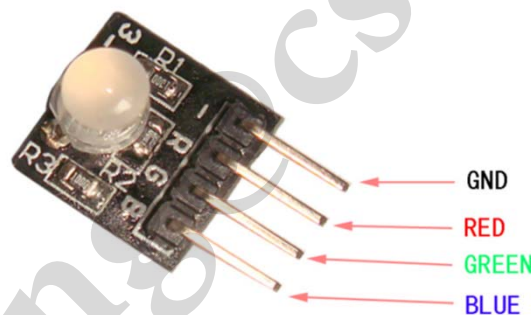


● 利用RGB三種不同顏色的輸入，混合出全彩的效果。

▶ R 紅色: 0~255

▶ G 綠色: 0~255

▶ B 藍色: 0~255



● 程式碼