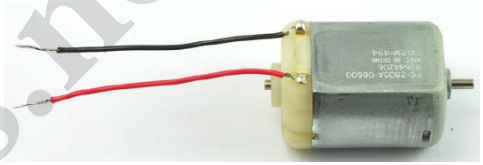
Chap 9
脈波寬度調變
PWM

使用的 I/O 裝置

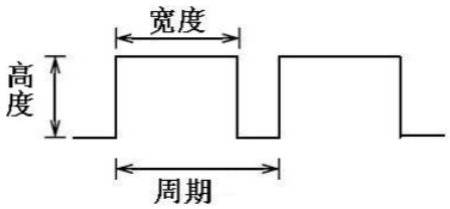
伺服馬達

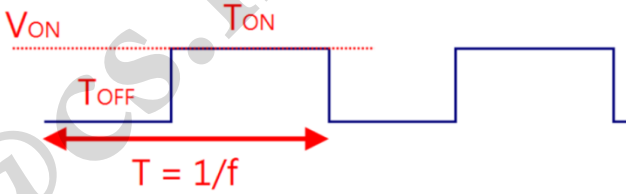


直流馬達

9-1 PWM?

- **脈波寬度調變 Pulse Width Modulation (PWM)**，簡單的說，是一種利用數位訊號模擬類比訊號的方式。通常我們可以用來調整燈光的亮度、馬達的轉速、RGB LED 的配色、螢幕亮度控制、喇叭的大小聲/聲音頻率等...。

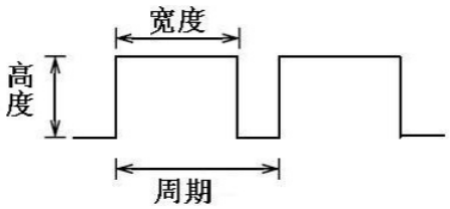


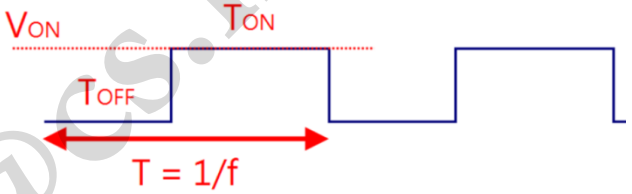


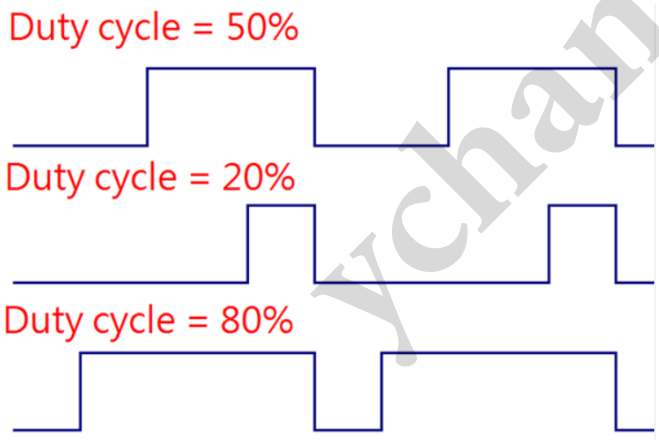
- 切換週期 T，通常我們會用**頻率 f (Hz)** 來表達，在 Arduino pin 上基本的切換頻率大約是 490Hz，部分的 pin 是 980 Hz (Arduino Uno pin 5&6)。
- **Duty Cycle = Ton/T (%)**，為 ON 的時間與 周期時間T相除的百分比
- **模擬出的電壓 V = Von x duty cycle (%)**，可以知道 duty cycle 越高模擬出的電壓越高，當完全沒有 OFF 的時候，duty cycle = 100%，**V = Von 為最大可輸出的電壓，這時候電燈會最亮**

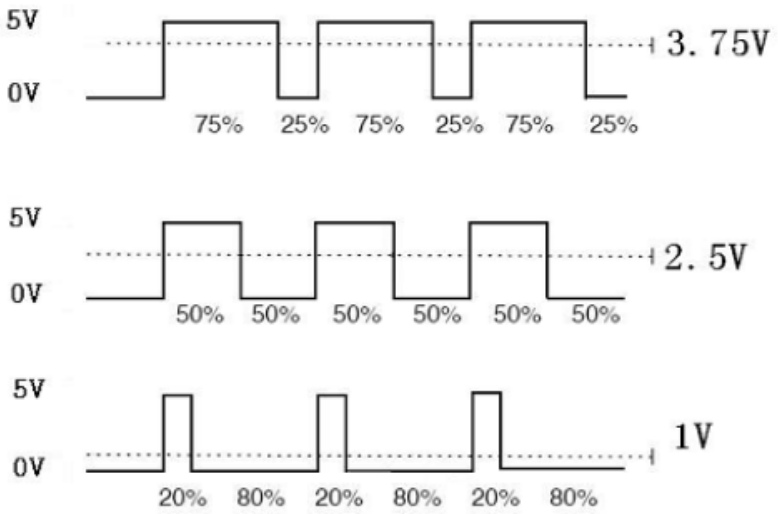
9-1 PWM?

- **脈波寬度調變 Pulse Width Modulation (PWM)**，簡單的說，是一種利用數位訊號模擬類比訊號的方式。通常我們可以用來調整燈光的亮度、馬達的轉速、RGB LED 的配色、螢幕亮度控制、喇叭的大小聲/聲音頻率等...。





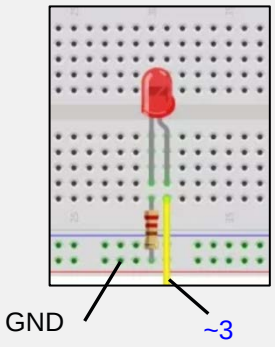




analogWrite( )

【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。

以下是官網上的模擬範例:

```
int pin = 13;
void setup() {
  pinMode(pin, OUTPUT);
}
void loop(){
  digitalWrite(pin, HIGH);
  delayMicroseconds(100); // Approximately 10% duty cycle @ 1KHz
  digitalWrite(pin, LOW);
  delayMicroseconds(1000 - 100);
}
```

這個範例中, **一個循環是 1000 us = 1ms, 所以一秒循環 1000次, 因此 Frequency 是 1 KHz,**
每個循環中, **有電的比率是 100/1000 * 100% = 10%, 所以 duty cycle (佔空比)為 10%;**
這樣就可以模擬出 5Volt x 10% = **0.5 Volt 的電壓!**

如果真的這樣做, 有好處也有壞處, 官網上已經說了:

好處是任一支 pin 都可這樣用, 包括 Pint 0 到 Pin 13, 以及 Pin A0 到 A5 共 20支 pin 都可以!
壞處卻更多, 首先就是頻率(Frequency)和佔空比(duty cycle)可能受中斷(Interrup)的影響變成不是很準確 !!
最大的壞處是, 在某支 pin 做 PWM 輸出期間都沒辦法做別的事情 !!

既然說這只是示範可以這樣做, **在 Arduino 當然不可能是這麼做,**
那 Arduino 是怎麼做的呢?
就是透過 **Timer** 計時器直接控制 pin 做 PWM 輸出, Arduino UNO 的 MCU 有三個 timer,
其中 **timer0** 控制 **pin 5, pin 6; timer1** 控制 **pin 9, pin 10; timer 2 控制 pin 11, pin 3;**
所以, 我們可以對這些 **pin** 用 **analogWrite(pin, val);** 輸出 0 到 255 的 **val** 值到 pin ;
如果輸出 val 是 0, 它會偷偷直接改用 digitalWrite(pin, 0); 輸出,
如果 val 是 255, 也是會偷偷直接改用 digitalWrite(pin, 1); 輸出!
如果 val 是 1 到 254, 則會下命令請 pin 腳對應的 timer 計時器(定時器)幫忙!!

- Timer0: pin 5, 6, PWM 頻率 976.5625Hz, duty cycle可以 2/256 ~ 255/256 (對應 1 到254);
- Timer1: pin 9, 10, PWM 頻率 490.196Hz, duty cycle 可以 1/255 ~ 254/255(對應 1 到254);
- Timer2: pin 11, 3, 同Timer1

- Timer0: pin 5, 6, Fast PWM mode, 16MHz/64/256=976.5625Hz;
- Timer1: pin 9, 10, 8-bit phase correct PWM mode, 16MHz/64/255/2=490.196Hz;
- Timer2: pin 11, 3, 8-bit phase correct PWM mode, 16MHz/64/255/2=490.196Hz;

- **那 PWM 的 Frequency 可不可以更改?**
- 可以, 偷改 timer 的 Prescaler 就可以達到更改 Frequency 的目的！ 但是, 千萬不要更改 timer0 的 Prescaler, 否則 millis( ) 和 micros( ) 以及 delay() 都會受到影響 !!!

Timer1

TCCR1B = TCCR1B & 0xF8 | ?;
其中？與對應頻率如下:

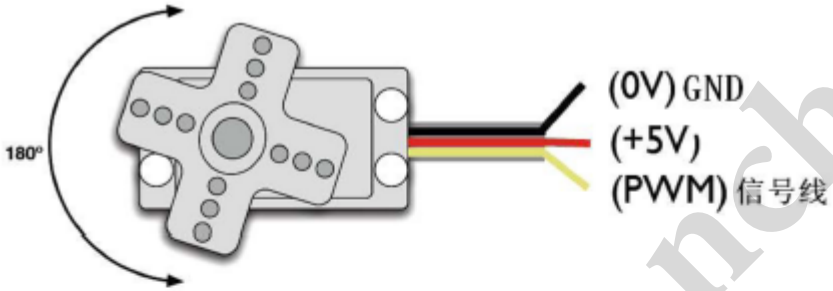
?	Prescaler	Frequency
1	1	31372.549 Hz
2	8	3921.569
3	64	490.196
4	256	122.549
5	1024	30.637 Hz

Timer2

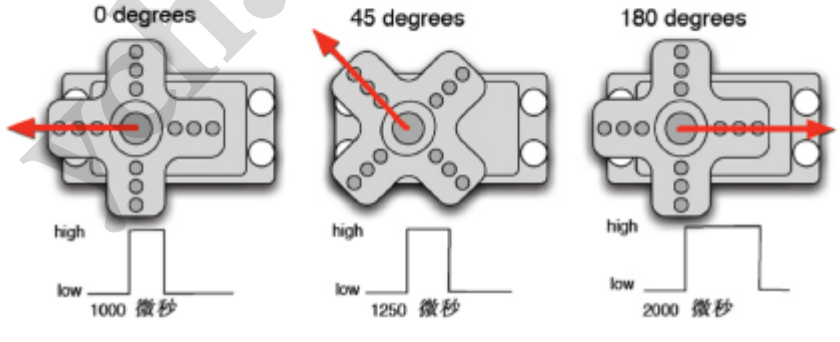
TCCR2B = TCCR2B & 0xF8 | ?;
其中？與對應頻率如下:

?	Prescaler	Frequency
1	1	31372.549 Hz
2	8	3921.569
3	32	980.392
4	64	490.196
5	128	245.098
6	256	122.549
7	1024	30.637 Hz

範例 9-1



舵机的转动的角度是通过调节PWM（脉冲宽度调制）信号的占空比来实现的，标准PWM（脉冲宽度调制）信号的周期固定为20ms（50Hz），理论上脉宽分布应在1ms到2ms 之间，但是，事实上脉宽可由0.5ms 到2.5ms 之间，脉宽和舵机的转角0°～180°相对应。有一点值得注意的地方，由于舵机牌子不同，对于同一信号，不同牌子的舵机旋转的角度也会有所不同。



範例 9-1

500-2500us的PWM高电平部分对应控制180度舵机的0-180度

以180度角度伺服为例，那么对应的控制关系是这样的：

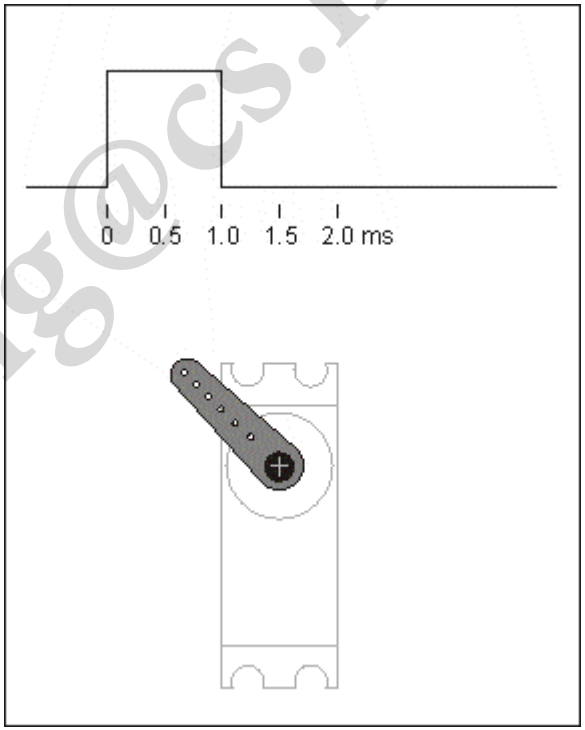
0.5ms----------0度；

1.0ms----------45度；

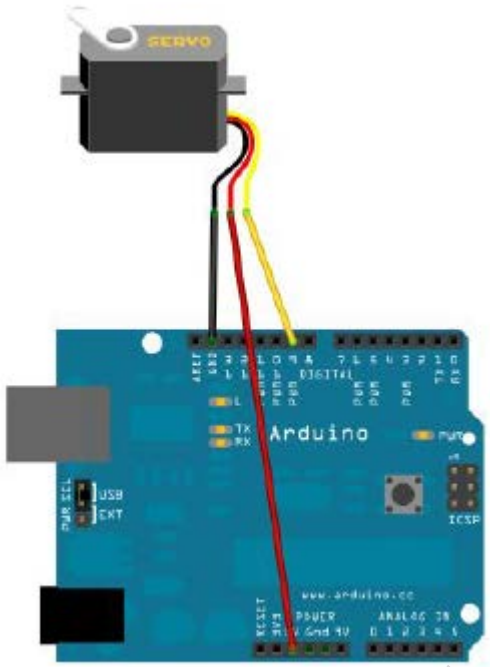
1.5ms----------90度；

2.0ms----------135度；

2.5ms----------180度；



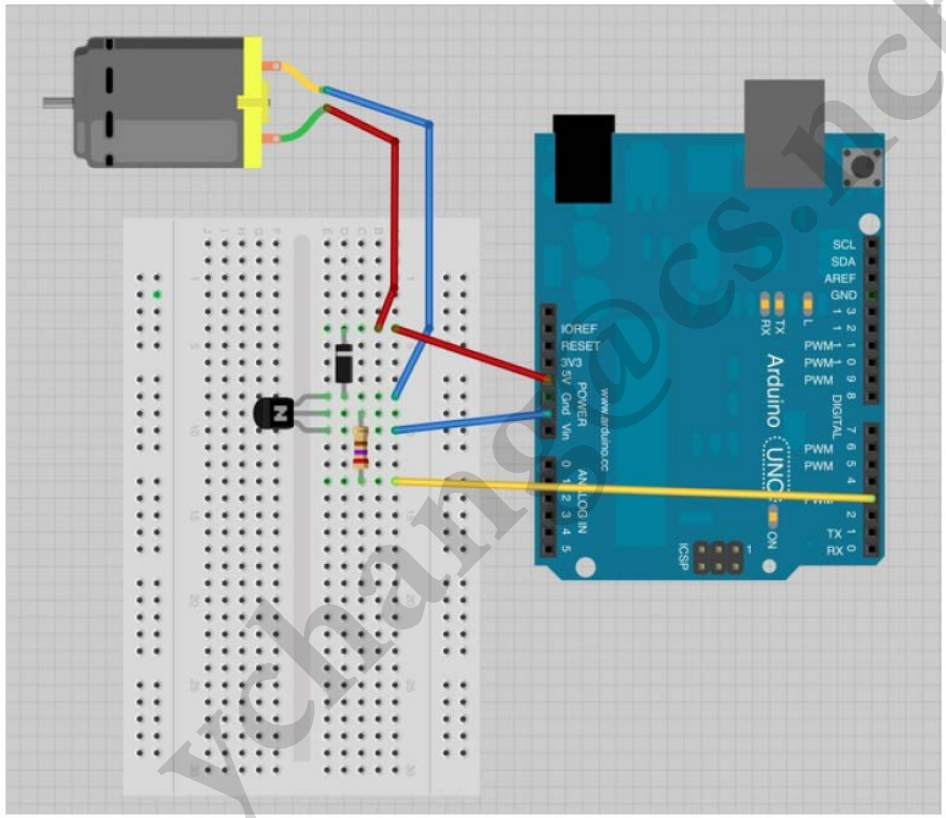
範例 9-1




```
void servopulse(int servopin,int myangle)//定义一个脉冲函数
{
pulsewidth=(myangle*11)+500;//将角度转化为500-2480  的脉宽值
digitalWrite(servopin,HIGH);//将舵机接口电平至高
delayMicroseconds(pulsewidth);//延时脉宽值的微秒数
digitalWrite(servopin,LOW);//将舵机接口电平至低
delay(20-pulsewidth/1000);
}
```

- 程式碼

範例 9-2



- 程式碼