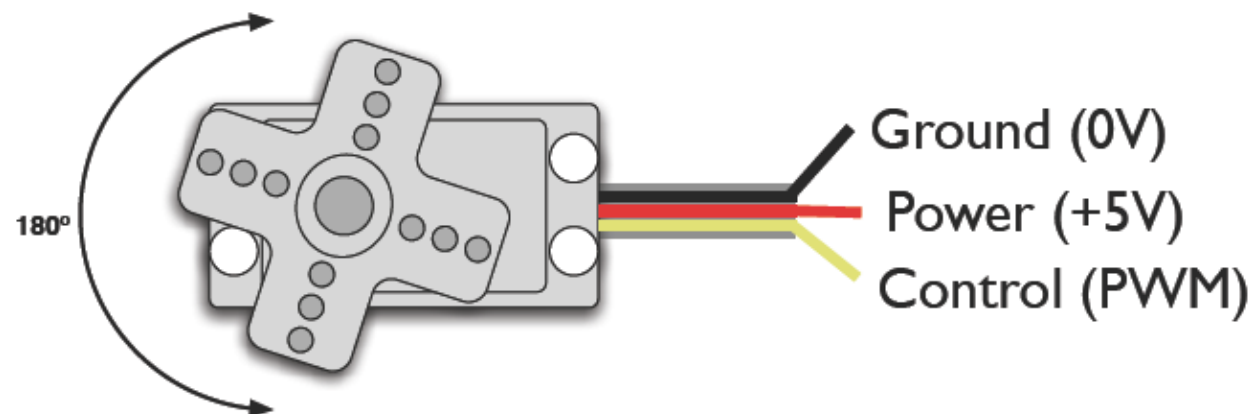
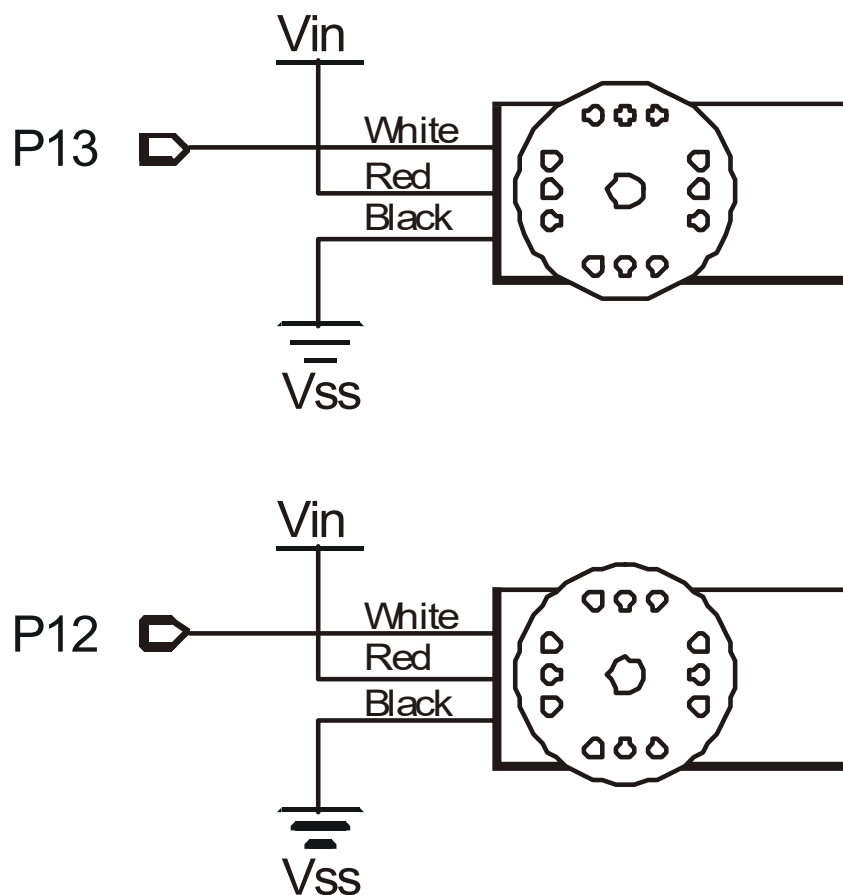


# 5.ABB-Car運動



## 連續旋轉伺服機介紹:





轉向	PWM輸出 HIGH的時間( $\mu$ s)
順時鐘轉	1300
停止轉動	1500
逆時鐘轉	1700

# 藉由LED來觀察伺服機的控制訊號

從Arduino發送到伺服機HIGH或LOW訊號必須維持一段時間，且要比HIGH和LOW指令更準確。

這是因為伺服機需要測量這段時間來視為轉動的指示，接著用這些傳送進來的訊號來指示連續旋轉伺服機上面的轉盤要轉到什麼位置。為了讓伺服機可以精準的控制，這一段極短的時間所需要的準確度並不是Delay指令所能夠達成的。

當你修正Delay指令後面的Duration參數時，時間常數每多一或少一就和原來的時間長度差1ms(Delay)指令的Duration參數單位是千分之一秒。

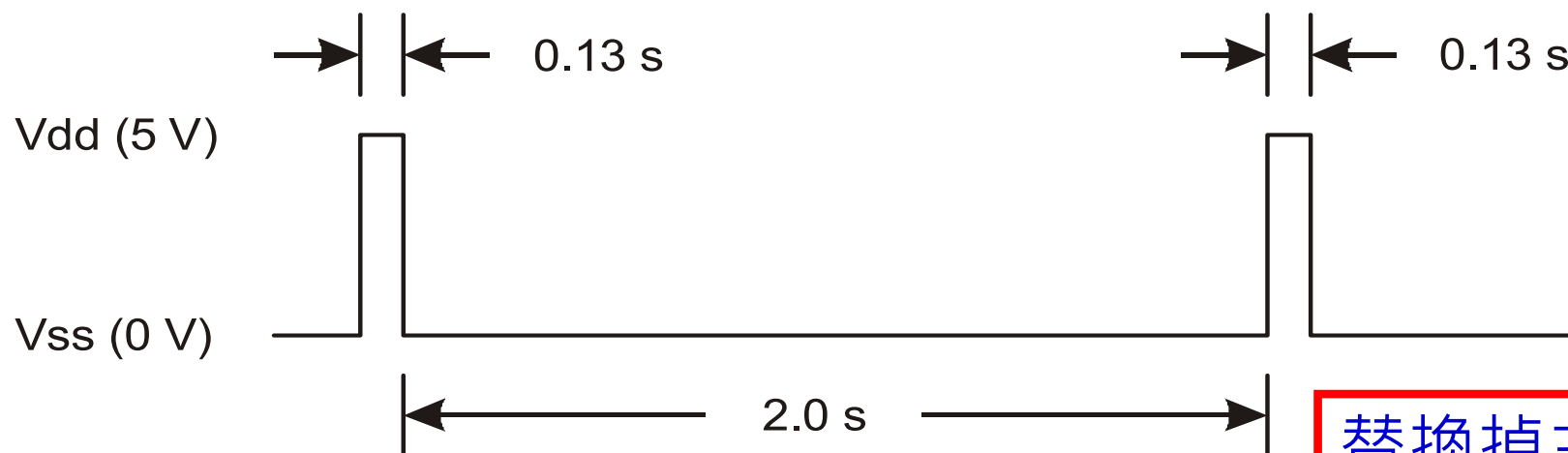
這也表示了，我們無法靠Delay指令來達到比1ms小的時間長度；因此我們必須要依靠另一個指令來達到更小的時間長度，配合伺服機量測訊號高低時所需要的時間準確度。這個指令就是DelayMicroseconds指令，而這個DelayMicroseconds指令Duration參數的單位是百萬分之一秒！



一微秒是百萬分之一秒。它的簡寫是 $\mu s$ 。當你要寫下這個單位時，要千萬小心它不是英文字母中的u，而是希臘字母中的 $\mu$ 。  
舉例：8微秒就會被簡寫成 $8\mu s$ 。

# 範例程式：LED模擬伺服機訊號

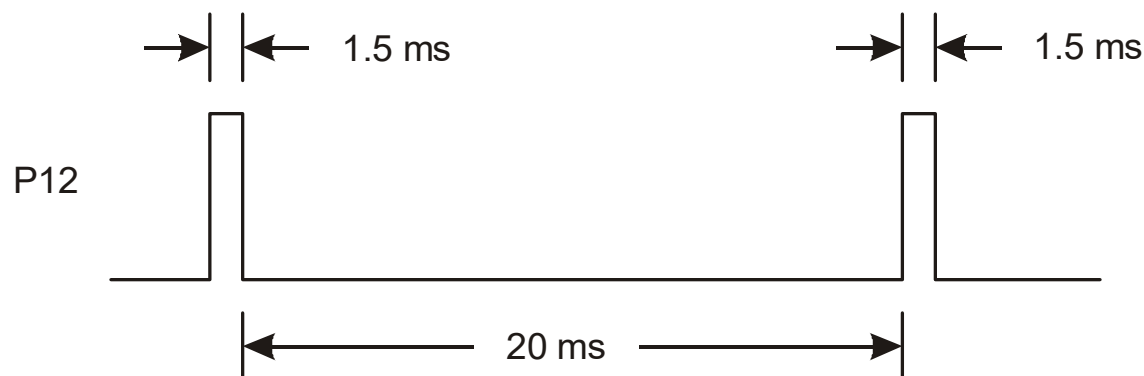
HIGH訊號持續0.13秒，LOW訊號持續2秒。這種訊號切換速度是**伺服機**用來控制運動的速度的**一百倍**慢。



替換掉主  
程式內容

```
void loop(){  
  digitalWrite(ledPin1,HIGH);  
  delayMicroseconds(130000);  
  digitalWrite(ledPin1,LOW);  
  delay(2000);  
}
```

## 傳送訊號



如圖所示，對伺服機輸出1.5毫秒的脈衝告訴伺服機要 保持停止。

## 範例程式：

```
//伺服馬達歸零  
int servoPin=12;  
//設定servoPin由12腳位控制  
  
void setup(){  
  pinMode(servoPin,OUTPUT);  
  //設定12腳位為輸出  
}
```

```
void loop(){  
  digitalWrite(servoPin,HIGH);  
  delayMicroseconds(1500);  
  //維持1.5ms  
  digitalWrite(servoPin,LOW);  
  delay(20);  
}
```

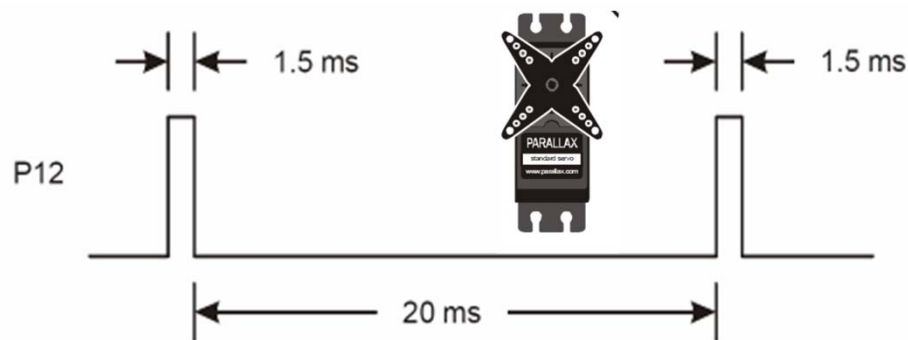
如果旋轉盤會開始旋轉，這時你可以聽到伺服機發出的聲音。  
可以拿起螺絲起子，輕輕的調整它裡面的電位計。  
如圖所示:調整電位計直到伺服機停止旋轉。



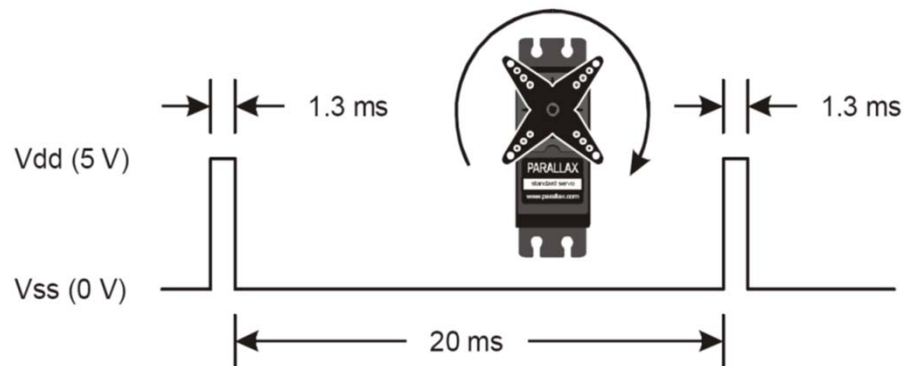
注意：使用螺絲起子時，千萬不要壓的太用力。在伺服機裡面的電位計相當精密，不能承受太大的力量，因此調整時千萬要小心控制力道。



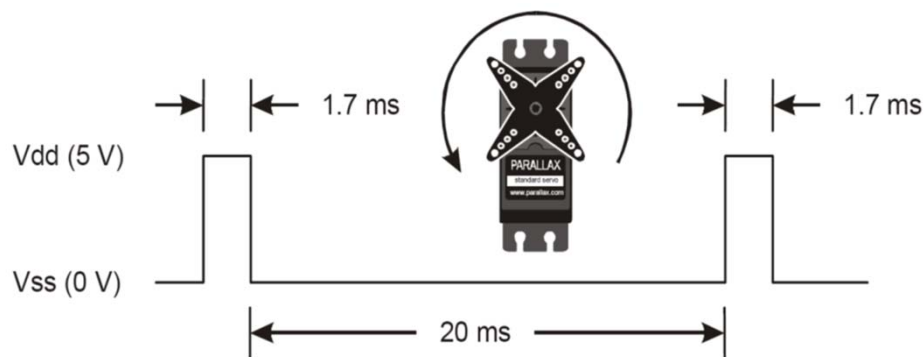
# 連續旋轉伺服馬達測試 (PWM訊號)



```
void loop()  
{  
  digitalWrite(servoPin,HIGH);  
  delayMicroseconds(1500);  
  digitalWrite(servoPin,LOW);  
  delay(20); }  
}
```



```
void loop()  
{  
  digitalWrite(servoPin,HIGH);  
  delayMicroseconds(1300);  
  digitalWrite(servoPin,LOW);  
  delay(20); }  
}
```

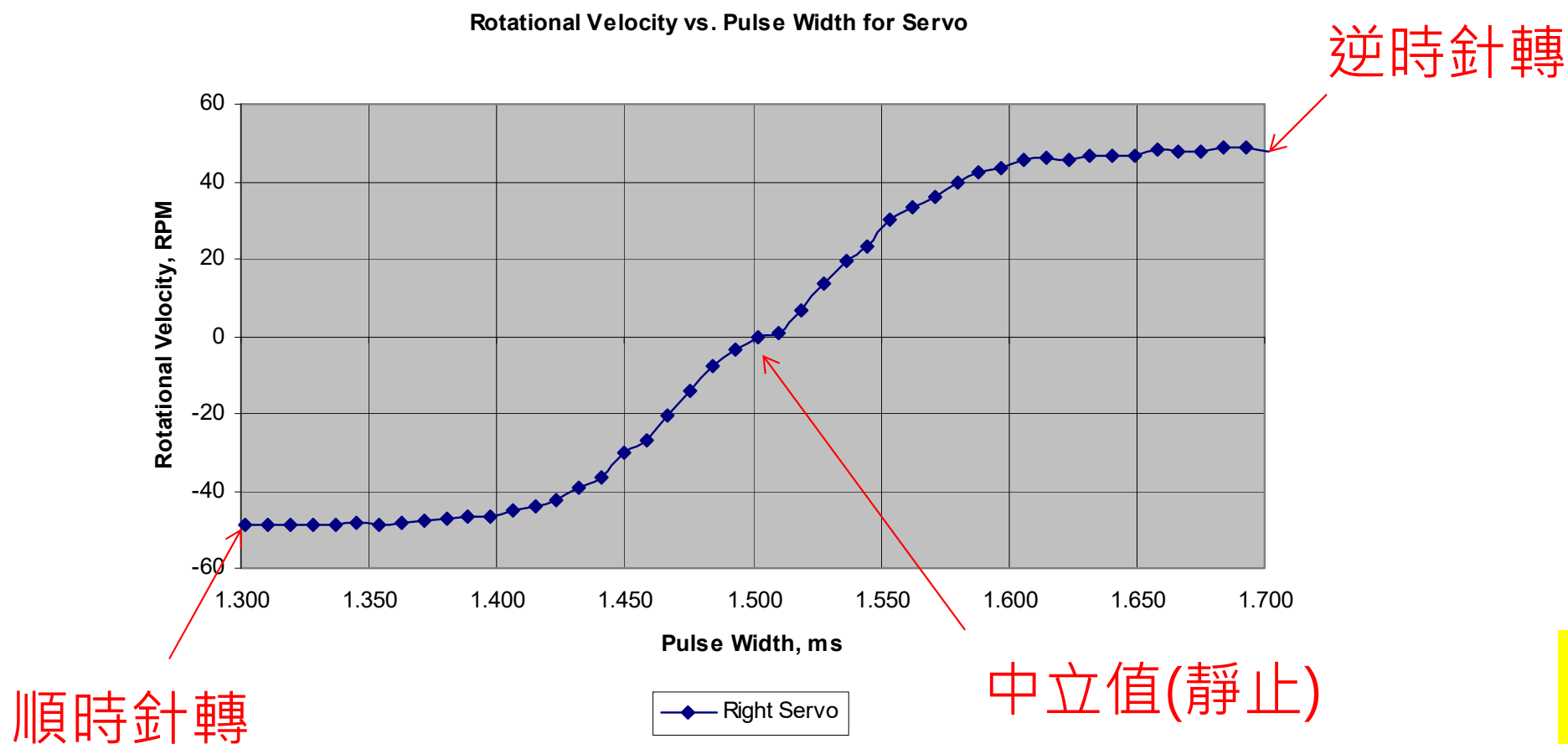


```
void loop()  
{  
  digitalWrite(servoPin,HIGH);  
  delayMicroseconds(1700);  
  digitalWrite(servoPin,LOW);  
  delay(20); }  
}
```



# 脈波訊號長短與伺服馬達轉速關係圖

下圖為輸入各種不同脈波寬度的脈訊號，對伺服馬達轉速所造成的影響。





# 機器人行為

Description	Behavior
全速 P13 逆時針, P12 順時針	前進
全速 P13 順時針, P12 逆時針	後退
全速 P13 逆時針, P12 逆時針	右轉
全速 P13 順時針, P12 順時針	左轉
P13 停止 P12 全速逆時針轉	以左輪為中心向後左轉
P13 全速順時針轉 P12 停止	以右輪為中心向後右轉
P13 停止 P12 停止	停止
P13 慢速度逆時針轉 P12 慢速度順時針轉	慢慢前進
P13 中等速度逆時針轉 P12 中等速度順時針轉	中等速度前進
P13 全速逆時針轉 P12 中等速度順時針轉	逐漸向右轉
P13 中等速度逆時針轉 P12 全速順時針轉	逐漸向左轉

# 測試ABB-Car程式

int servoPin1=12; //設定servoMotor1由12腳位控制

```
void setup(){  
  pinMode(servoPin1,OUTPUT); //12腳位為輸出  
}
```

\_6\_motortest\_01

```
void loop(){  
  for (int i=0;i<=120;i++){ //正向旋轉約三秒
```

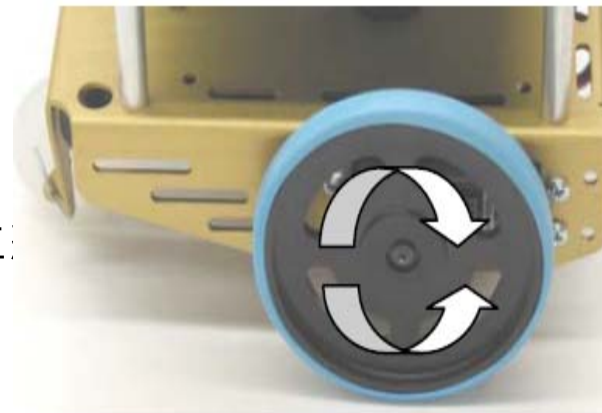
```
    digitalWrite(servoPin1,HIGH);  
    delayMicroseconds(1300);  
    digitalWrite(servoPin1,LOW);  
    delay(20);  }
```

```
  for (int i=0;i<=40;i++){ //靜止
```

```
    digitalWrite(servoPin1,HIGH);  
    delayMicroseconds(1500);  
    digitalWrite(servoPin1,LOW);  
    delay(20);  }
```

```
  for (int i=0;i<=120;i++){ //反向旋轉約三秒
```

```
    digitalWrite(servoPin1,HIGH);  
    delayMicroseconds(1700);  
    digitalWrite(servoPin1,LOW);  
    delay(20);  } }
```

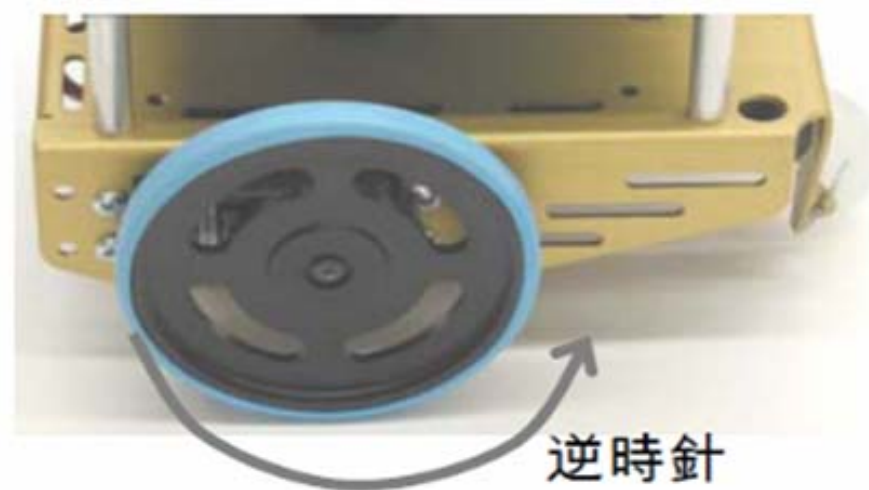
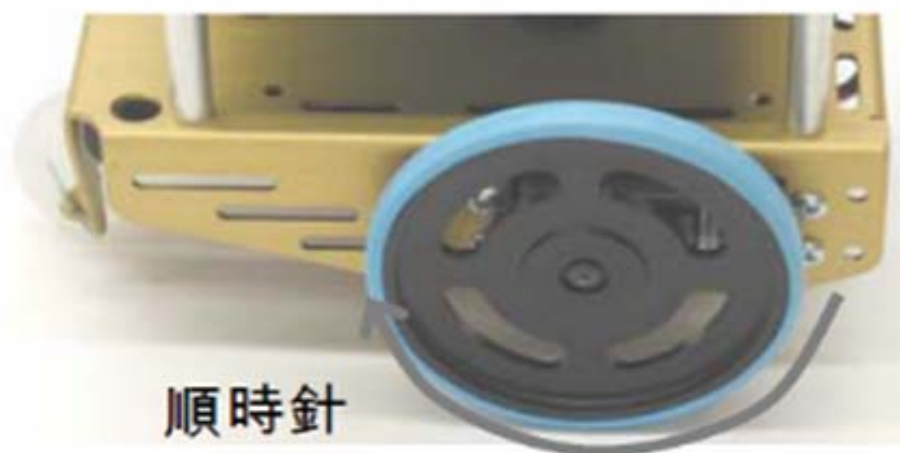


Clockwise 3 seconds

Stop 1 second

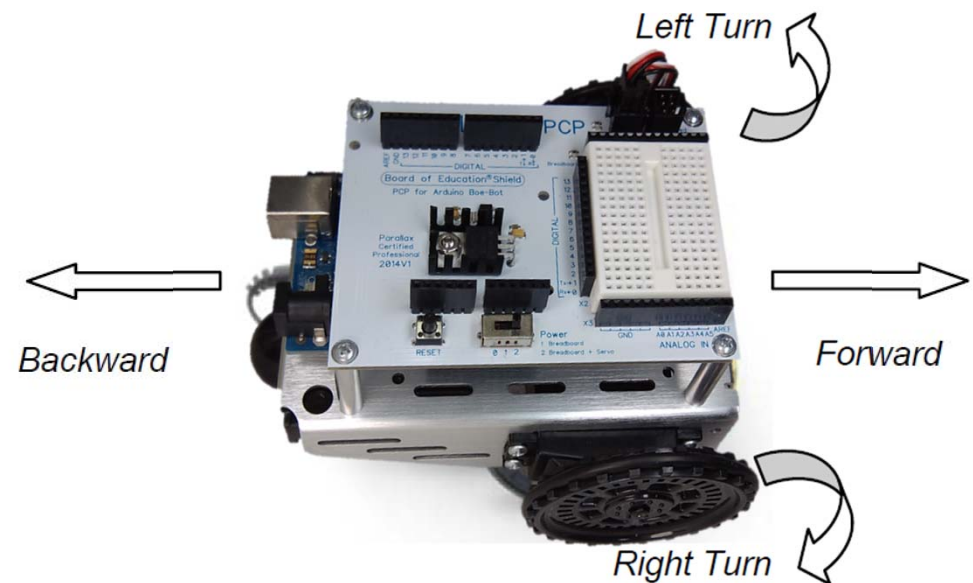
Counterclockwise 3 seconds

# ABB-Car基本運動練習



## ◆使ABB-Car開始向前走

```
int servoPin1=12;           //設定servoMotor1由12腳位控制
int servoPin2=13;           //設定servoMotor2由13腳位控制
void setup(){
  pinMode(servoPin1,OUTPUT); //設定12腳位為輸出
  pinMode(servoPin2,OUTPUT); //設定13腳位為輸出
}
void loop(){
  digitalWrite(servoPin1,HIGH);
  delayMicroseconds(1300);
  digitalWrite(servoPin1,LOW);
  delay(20);
  digitalWrite(servoPin2,HIGH);
  delayMicroseconds(1700);
  digitalWrite(servoPin2,LOW);
  delay(20);
}
```



# 範例程式:

## ◆讓ABB-Car前進、左轉、右轉接著後退

```
int servoPin1=12;           //servoMotor1由12腳位控制
int servoPin2=13;           //servoMotor2由13腳位控制
void setup(){
  pinMode(servoPin1,OUTPUT); //12腳位為輸出
  pinMode(servoPin2,OUTPUT); //13腳位為輸出
}
void loop(){
  for (int i=0;i<=40;i++){ //前進
    digitalWrite(servoPin1,HIGH);
    delayMicroseconds(1300);
    digitalWrite(servoPin1,LOW);
    delay(20);
    digitalWrite(servoPin2,HIGH);
    delayMicroseconds(1700);
    digitalWrite(servoPin2,LOW);
    delay(20);
  }
```

```
for (int i=0;i<=20;i++){           //左轉
    digitalWrite(servoPin1,HIGH);
    delayMicroseconds(1300);
    digitalWrite(servoPin1,LOW);
    delay(20);
    digitalWrite(servoPin2,HIGH);
    delayMicroseconds(1300);
    digitalWrite(servoPin2,LOW);
    delay(20);
}
for (int i=0;i<=20;i++){           //右轉
    digitalWrite(servoPin1,HIGH);
    delayMicroseconds(1700);
    digitalWrite(servoPin1,LOW);
    delay(20);
    digitalWrite(servoPin2,HIGH);
    delayMicroseconds(1700);
    digitalWrite(servoPin2,LOW);
    delay(20);
}
```

```
for (int i=0;i<=40;i++){           //後退
    digitalWrite(servoPin1,HIGH);
    delayMicroseconds(1700);
    digitalWrite(servoPin1,LOW);
    delay(20);
    digitalWrite(servoPin2,HIGH);
    delayMicroseconds(1300);
    digitalWrite(servoPin2,LOW);
    delay(20);
}
}
```



# 以<Servo.h>內的函式來控制馬達

注意:使用Servo.XXX() 的函數，一開始必須要打上#include <Servo.h>

```
//使用<Servo.h>內函數控制馬達  _7_motortest_02
#include <Servo.h>           // 引用 Servo Library
Servo servoLeft;            // [Servo 物件名]建立一個 Servo 物件
void setup(){
    servoLeft.attach(13); } // Servo物件接 pin13
void loop(){                //馬達不停的正反轉變換
    servoLeft.writeMicroseconds(1300); //正向旋轉三秒
    delay(3000);
    servoLeft.writeMicroseconds(1500); //靜止一秒
    delay(1000);
    servoLeft.writeMicroseconds(1700); //反向旋轉三秒
    delay(3000);
    servoLeft.detach();      // 停止傳送訊號給伺服馬達
}
```

Table 4-1: us parameters in Servo.writeMicroseconds(us)

Top speed clockwise	Linear speed zone starts	Full stop	Linear speed zone ends	Top speed counterclockwise
1300	1400	1500	1600	1700



# 使用副程式方式控制ABB Car

```
//使用副程式控制ABB Car <_8_ABBmove>
#include <Servo.h>    // 引用 Servo Library
Servo servoRight;     // 建立Servo 物件
Servo servoLeft;
void setup(){
    servoRight.attach(12); // servoRight 接在 pin 12
    servoLeft.attach(13); // servoLeft 接在 pin 13
}
void loop(){
    forward();          //呼叫前進的副程式
    trunLeft();         //呼叫左轉的副程式
    trunRight();        //呼叫右轉的副程式
    backward();         //呼叫後退的副程式
}
```

```
int forward(){           //前進的副程式
    servoRight.writeMicroseconds(1300);
    servoLeft.writeMicroseconds(1700);
    delay(1000); }
int trunLeft(){          //左轉副程式
    servoRight.writeMicroseconds(1700);
    servoLeft.writeMicroseconds(1700);
    delay(300); }
int trunRight(){         //右轉的副程式
    servoRight.writeMicroseconds(1300);
    servoLeft.writeMicroseconds(1300);
    delay(300); }
int backward(){          //後退的副程式
    servoRight.writeMicroseconds(1700);
    servoLeft.writeMicroseconds(1300);
    delay(1000); }
```

試看看：調整delay()的時間來使車子轉向90°

換你試看看練習 \_ 練習下面的路徑

