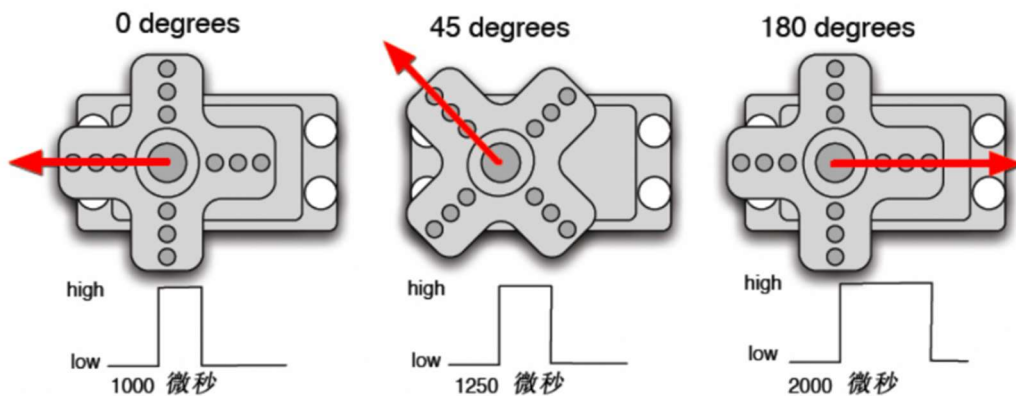


## 伺服馬達

- 伺服馬達的轉動的角度是通過調節 **PWM (脈衝寬度調製)** 信號的實現來實現的，標準 PWM (脈衝寬度調製) 信號的周期固定為 20ms (50Hz)，理論上脈寬分佈應在 1ms 到 2ms 之間，但是，實際上脈寬預期 0.5ms 到 2.5ms 之間，脈寬和舵機的轉角  $0^{\circ} \sim 180^{\circ}$  相對應。有一點不會的地方，由於舵機牌子不同，對於同一信號，不同牌子的舵機旋轉的角度也會有所不同。
- Arduino 的驅動能力有限，所以當需要控制 1 個以上的舵機時建議加上馬達擴展板。



- 另外各位在選購伺服馬達舵機時務必注意，同種型號會有 2 種規格
  1. 180 度：標準舵機 (可控制角度)
  2. 360 度：連續旋轉舵機 (不可控制角度)
  3. 舵機通過脈波寬度的時間來控制(PWM)，頻率是 50Hz 的(即一個周期是 20ms)  
SG90 舵機控制參數的範圍是： $0^{\circ}$ : 0.5ms,  $180^{\circ}$ : 2.5ms。然後對於每一度，將脈波寬度除以  $180^{\circ}$  即得。
  4. `servo.attach(GIO, pulse_width_min, pulse_width_max);` 這裡就可以設置上面的脈波寬度 min 和 max(出於安全考慮，Servo 的默認值是 1000,2000，導致實際運轉時，並不是完全是  $180^{\circ}$ )

## 伺服馬達種類



### SG-90 伺服馬達

- 重量：9g.
- 尺寸：23\*12.2\*29mm.
- 轉矩：1.8kg-cm，電壓為 4.8V 時
- 

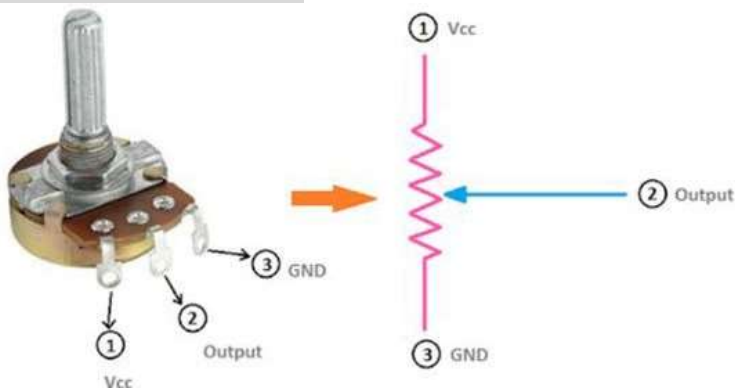
### MG-90S 伺服馬達

- 重量：13.6g.
- 尺寸：22.8\*12.2\*28.5mm.
- 轉矩：2.0kg-cm，電壓為 4.8V 時

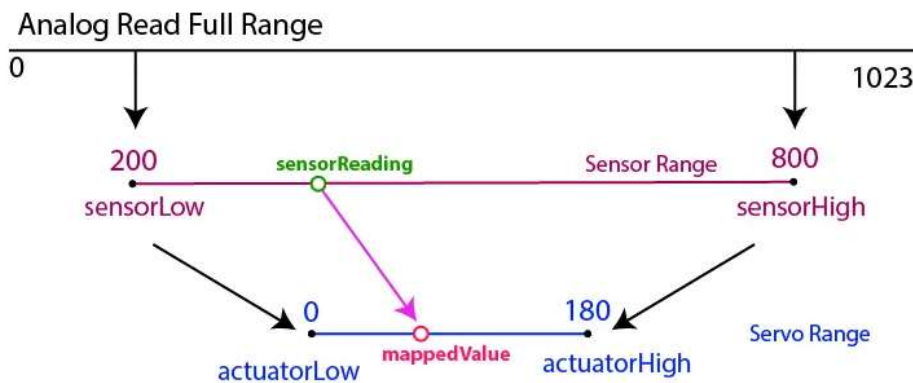
### MG-996R 伺服馬達

- 重量：55g.
- 尺寸：54\*38\*20mm.
- 轉矩：9kg-cm，電壓為 4.8V 時

## 電位計(可變電阻)

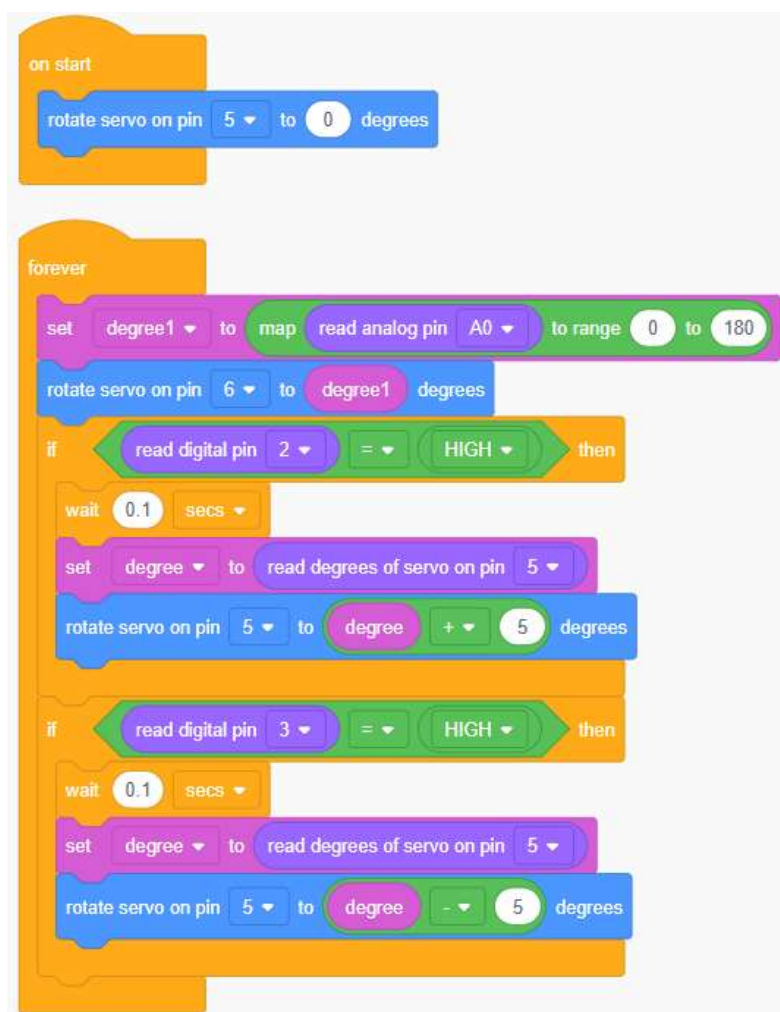
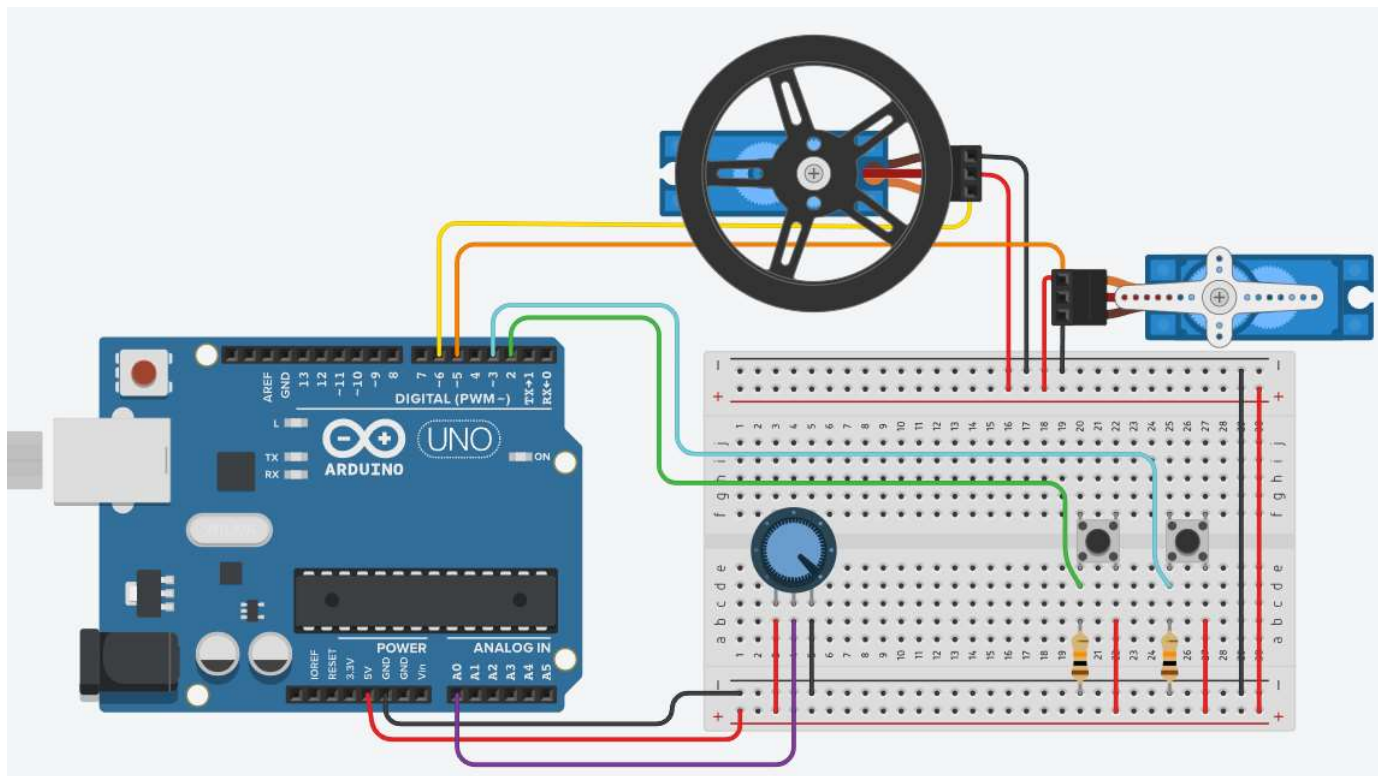


## Map 函數



```
int mappedValue = map(sensorReading, sensorLow, sensorHigh, actuatorLow, actuatorHigh);
```

# 伺服馬達實作應用



```

#include <Servo.h>

int degree = 0;
int degree1 = 0;
Servo servo_5;
Servo servo_6;

void setup() {
    servo_5.attach(5, 500, 2500) ;
    pinMode(A0, INPUT);
    servo_6.attach(6, 500, 2500);
    pinMode(2, INPUT);
    pinMode(3, INPUT);
    servo_5.write(0);
}

void loop() {
    degree1 = map(analogRead(A0), 0, 1023, 0, 180);
    servo_6.write(degree1);
    if (digitalRead(2) == HIGH) {
        delay(100);
        degree = servo_5.read();
        servo_5.write((degree + 5));
    }
    if (digitalRead(3) == HIGH) {
        delay(100);
        degree = servo_5.read();
        servo_5.write((degree - 5));
    }
}

```

```
#include <Servo.h> //載入函式庫，這是內建的，不用安裝
Servo myservo; // 建立 SERVO 物件

void setup() {
    myservo.attach(9); // 設定要將伺服馬達接到哪一個 PIN 腳
}

void loop() {
    myservo.write(0); //旋轉到 0 度，就是一般所說的歸零 //順時針旋轉最高速
    delay(1000);
    myservo.write(90); //旋轉到 90 度 //停止旋轉
    delay(1000);
    myservo.write(180); //旋轉到 180 度 //逆時針旋轉最高速
    delay(1000);
    myservo.write(90);
    delay(1000);
}
```