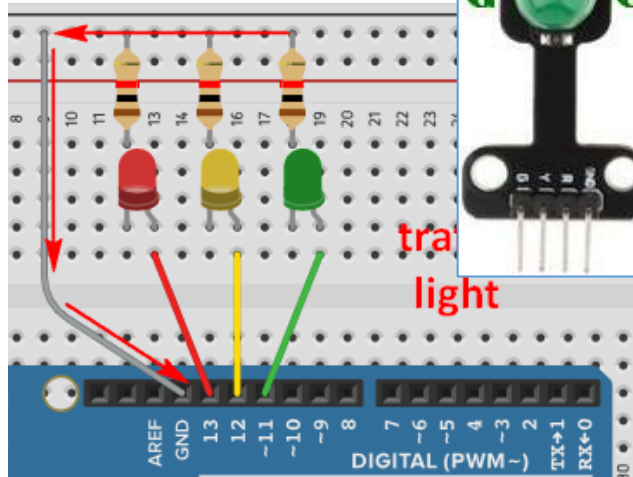


### Exercise 1

tinkercad  
& mBlock



#### 資料和指令

機器人模組

做一個變數

n

將變數 n 的值設為 0

將變數 n 的值改變 1

顯示變數 n

隱藏變數 n

做一個清單

新增積木指令

紅黃綠OFF

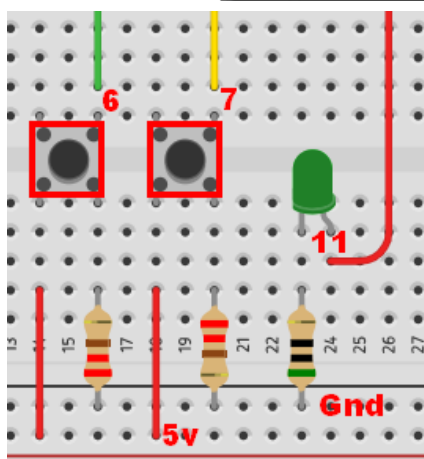
新建積木指令

紅黃綠OFF

選項

確認

取消



當 被點一下

紅黃綠off

不停重複

紅R

等待 6 秒

紅黃RY

等待 2 秒

綠G

等待 4 秒

黃Y

等待 2 秒

定義 紅黃綠off

定義 紅R

定義 紅黃RY

定義 綠G

定義 黃Y

digitalRead(7)

digitalWrite(11, \_\_\_\_\_);

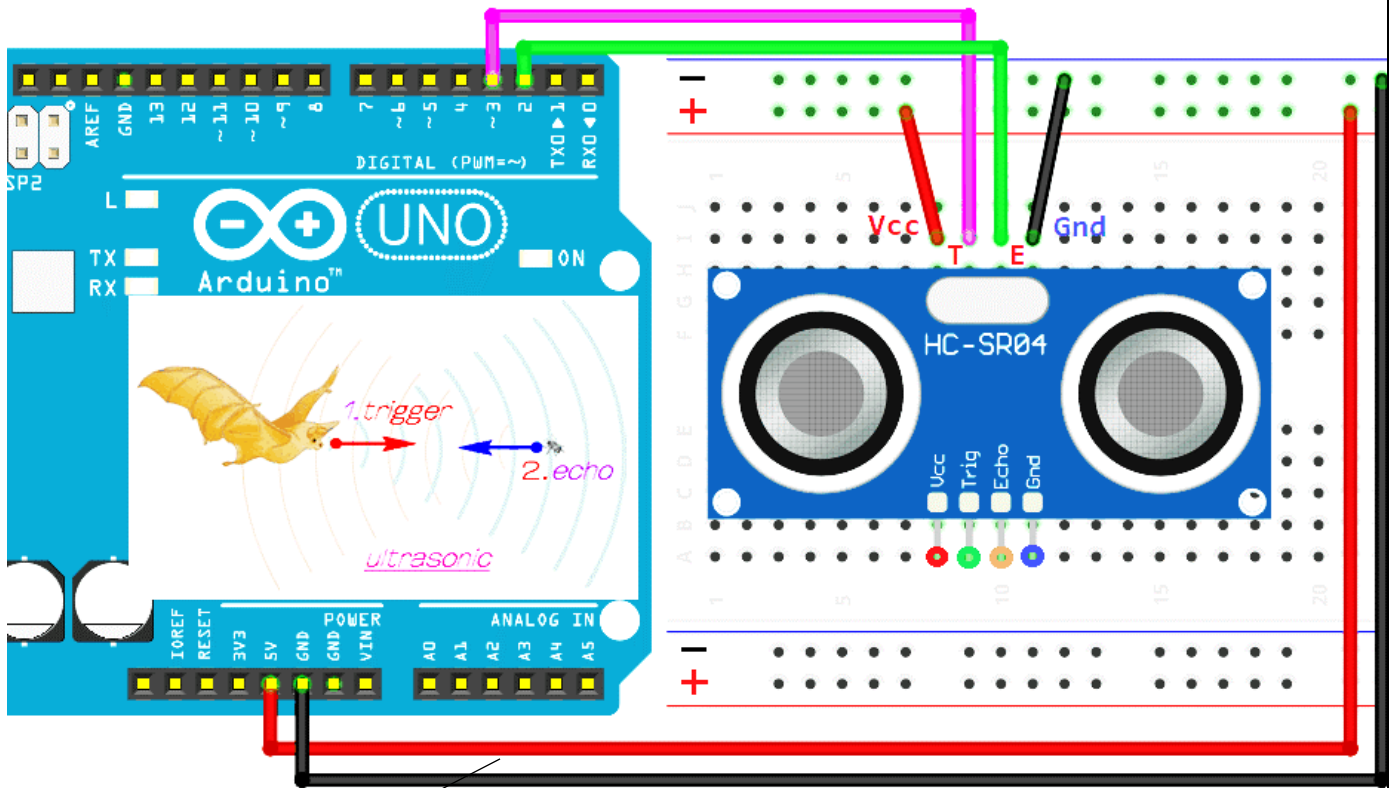
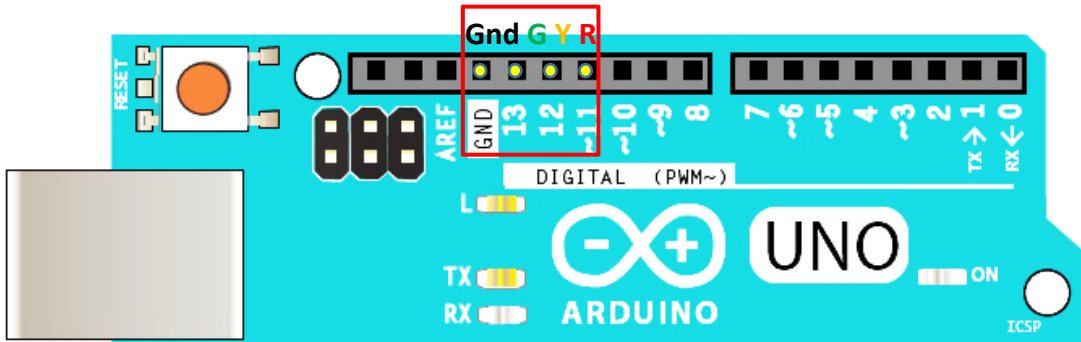
將接腳 11 設為 讀取數位接腳 7

LED 1(按下), 0(放開)

設置 數位腳位 11 輸出為 高電位

數位腳位 7

## Exercise 2



紅黑線

Vcc(5v) Trig(3) Echo(2) Gnd

音階	C(Do)	C#	D(Re)	D#	E(Mi)	F(Fa)	F#	G(So)	G#	A(La)	A#	B(Si)
低音	161	277	294	311	330	349	370	392	415	440	466	494
中音	523	554	587	622	659	698	740	784	831	880	932	988
高音	1046	1109	1175	1245	1318	1397	1480	1568	1661	1760	1865	1976

```
tone(8,330,500);
noTone(8);
```

### Exercise 3



The screenshot shows the Arduino IDE 1.8.3 interface. The 'Tools' menu is open, and the 'Board' submenu is selected, showing 'Arduino/Genuino Uno' as the selected board. The 'Serial Port' submenu is also open, showing 'COM3' as the selected port. The 'Preferences' menu item is highlighted in the main menu, and the 'Tools' menu item is highlighted in the Tools menu. A red box highlights the 'COM3' selection in the Serial Port submenu.

The screenshot shows the Arduino IDE 1.8.3 interface. The 'File' menu is open, and the 'Examples' submenu is selected, showing '01.Basics' as the selected example. The 'Blink' example is selected. The 'Serial Monitor' window is open, showing the code for the 'LED\_Run\_1' sketch. The code is as follows:

```
1. void setup() { // 設定 (只執行一次)
2.   pinMode(13, OUTPUT);
3. }
4.
5. void loop(){ // 循環 (不斷運行..)
6.   digitalWrite(13, HIGH);
7.   delay(1000);
8.   digitalWrite(13, LOW);
9. }
```

```
// (1) File → Example → Basic → Blink
```

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT); // 13  
}
```



```
void loop() {  
  digitalWrite(13, HIGH); delay(1000);  
  digitalWrite(13, LOW); delay(1000);  
}
```

```
// (2) Traffic-Light 交通燈
```

```
int state = 0; // 狀態
```

```
void setup() {  
  pinMode(13, _____);  
  pinMode(12, _____);  
  pinMode(11, _____);  
  Serial.begin(9600);  
}
```



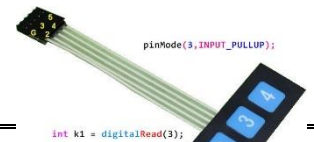
```
void loop() {  
  Serial.println(state);  
  if (state == 0) {  
    _____紅●  
  } else if (state == 1) {  
    _____紅●黃●  
  } else if (state == 2) {  
    _____綠●  
  } else if (state == 3) {  
    _____黃●  
  }  
  state++;  
  if (state > 3) state = 0;  
  delay(1000);  
}
```

```
// (3) Ultrasonic
```

```
float ultrasonic(int Trig, int Echo) {  
  pinMode(Trig, OUTPUT);  
  pinMode(Echo, INPUT);  
  digitalWrite(Trig, LOW); // 0.000002s  
  delayMicroseconds(2);  
  digitalWrite(Trig, HIGH); // 示意開始發出  
  delayMicroseconds(10);  
  digitalWrite(Trig, LOW); // 示意停止  
  float dist = pulseIn(Echo, HIGH) / 58; // 計算  
  Serial.println(dist);  
  return dist;  
}
```

```
void setup() {  
  pinMode(13, OUTPUT); // 紅燈  
  Serial.begin(9600); // 開啟監察器  
}
```

```
void loop() {  
  float d = ultrasonic(3, 2); // Trig, Echo  
  if (d > 0 && d < 10) digitalWrite(13, HIGH);  
  if (d > 10 && d < 20) _____  
  if (d > 20 && d < 30) _____  
  Serial.println(d);  
  delay(250);  
}
```



```
// (4) Button - Piano
```

```
void setup() {  
  pinMode(2, INPUT_PULLUP);  
  pinMode(3, INPUT_PULLUP);  
  pinMode(4, INPUT_PULLUP);  
  pinMode(5, INPUT_PULLUP);  
}
```

```
void loop() {  
  int k1 = digitalRead(2);  
  if (k1 == LOW) tone(8, 330, 250); // Press 1  
  else if (k2 == LOW) tone(8, 440, 250); // Press 2  
  ... ..  
  else noTone(8);  
  delay(200);  
}
```

## Exercise 4

(1) IR\_Rx.ino 接收 receive

```
#include <IRremote.h>
```

```
IRrecv irrecv(9);
```

```
decode_results results;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  irrecv.enableIRIn();
```

```
  irrecv.blink13(true);
```

```
}
```

```
void loop() {
```

```
  if (irrecv.decode(&results)) { // 如果接收到新訊息
```

```
    if (results.value != 0xFFFFFFFF) Serial.println(results.value, 16); // 顯示訊息
```

☆☆☆根據不同按鍵... 執行指令... / 作出反應...☆☆☆

```
    irrecv.resume();
```

```
    // 準備讀下一個 IR 訊號
```

```
    delay(200);
```

```
  }
```

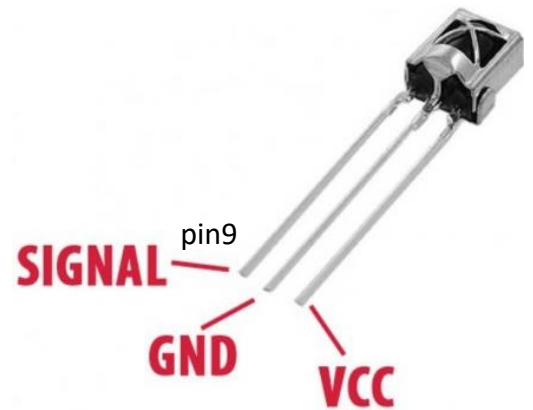
```
}
```

```
// Rx 接收 IR(Rx 接收腳=9), Gnd, Vcc
```

```
// 接收到的訊息
```

```
// Monitor 監察器
```

```
// 開始接收 IR 訊息
```



Infra-Red 紅外線

(2) results.value 接收到遙控傳來的 IR 訊號數值(16 進制)

0xFFA25D: CH-

0xFF629D: CH

0xFFE21D: CH+

0xFF22DD: prev

0xFF02FD: next

0xFFC23D: play/pause

0xFFE01F: vol-

0xFFA857: vol+

0xFF906F: EQ

0xFF9867: 100+

0xFFB04F: 200+

0xFF6897: 0

0xFF30CF: 1

0xFF18E7: 2

0xFF7A85: 3

0xFF10EF: 4

0xFF38C7: 5

0xFF5AA5: 6

0xFF42BD: 7

0xFF4AB5: 8

0xFF52AD: 9

0xFFFFFFFF: Repeat

(3) IR\_Tx.ino 發出 send

```
#include <IRremote.h>
```

```
unsigned long n=0x1234ABCD;
```

```
IRsend irsend;
```

```
// pin 3 & Gnd
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
  irsend.sendNEC(n, 32);
```

```
// 發出 IR 訊息(n)
```

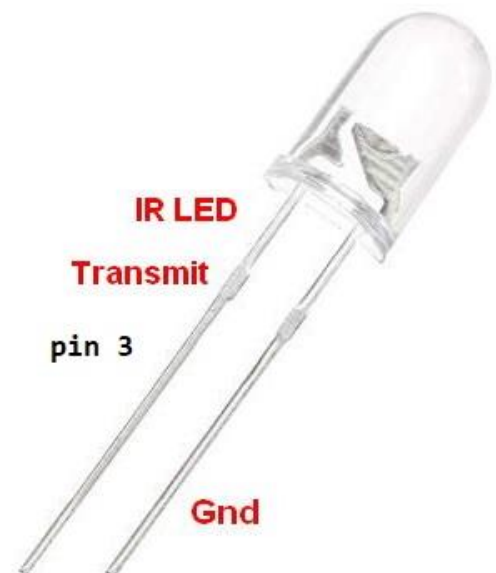
```
  Serial.println(n, 16);
```

```
// 顯示(n) 16 進制
```

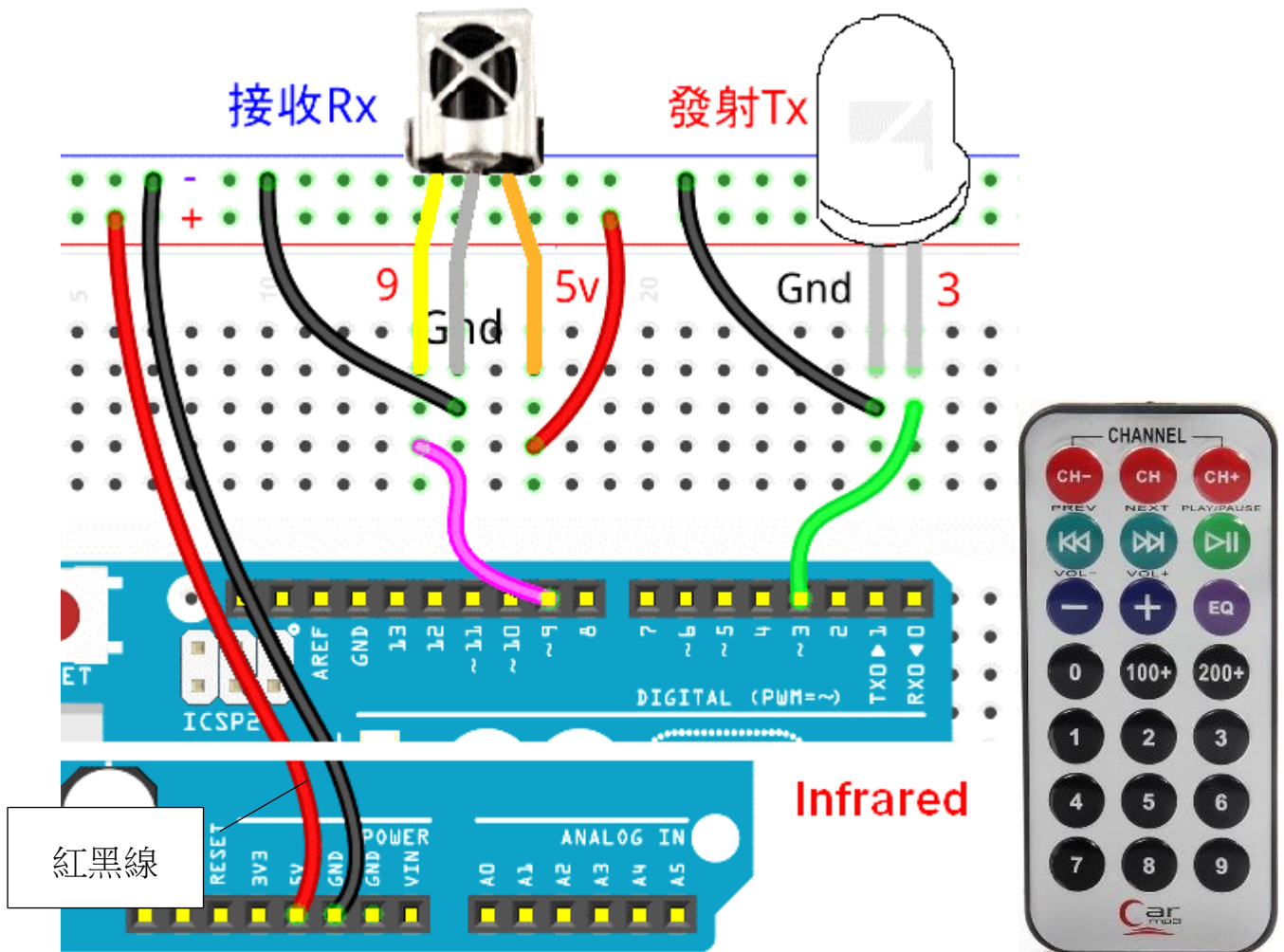
```
  delay(200);
```

```
  n++;
```

```
}
```







-----☆☆☆**任務**☆☆☆-----

1. 利用 `IR_Rx.ino` 及 Serial Monitor，查看按鍵 `0-9` 的代碼(例如：`0xFF6897`)。
2. 當使用者按 `0-3` 鍵，執行以下動作：

按 1 ●	按 2 ●	按 3 ●	按 0 ●
亮紅燈	亮黃燈	亮綠燈	關閉所有燈

3. 當使用者按 `0-3` 鍵，執行以下動作：按第 1 次(開/on)，按第 2 次(關/off)。

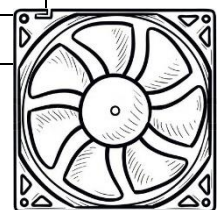
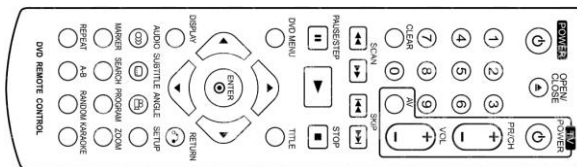
按 1 ●	按 2 ●	按 3 ●	按 0 ●
開↔關紅燈	開↔關黃燈	開↔關綠燈	開↔關風扇

```
int n=0; 按 1 : n=n+1; 或 n++;
```

if(`n%2==1`) 表示單數，

if(`n%2==0`) 表示雙數。

```
// n÷2 的餘數 == 0
```



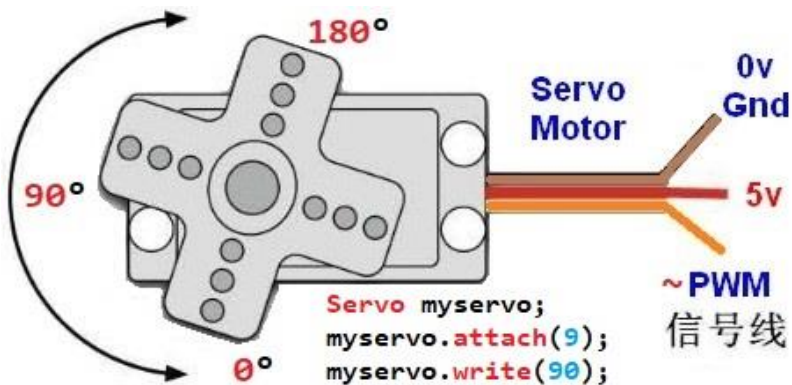
4. 利用 `IR_Rx.ino` 及 Serial Monitor，查看某電視遙控調校音量的代碼(例如：`0x123456`)。
5. 合併 `IR_Rx.ino` 及 `IR_Tx.ino`，當使用者按 `0-3` 鍵，扮電視遙控：控制音量、選台...

按 1	按 2	按 3	按 0
調高音量	轉下一台	...	開/關

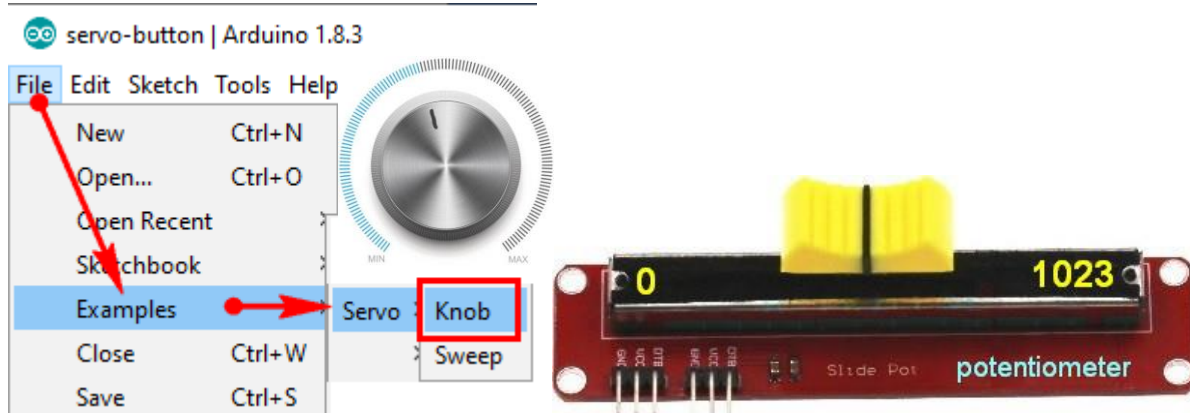
```
if(results.value==0xFF6897) // 按 1
    irsend.sendNEC(0x123456,32); // 調高音量
```

# Servo Motor 伺服馬達(舵機) (0°-90°-180°)

## Exercise 5



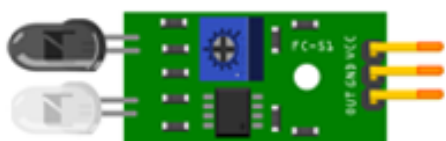
Examples: Servo → Knob



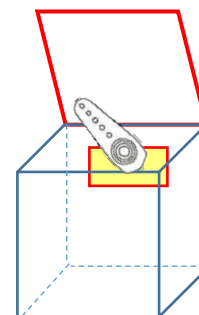
利用可變電阻(0-1023)去測試 Servo : Gnd, Vcc, Data(A0)

-----☆☆☆**任務**: 智能垃圾桶☆☆☆-----

1. 伺服馬達+紅外線近距離(紅外線壁障)傳感器 <https://jabert.world.tmall.com/>



IR Obstacle Avoidance Sensor (Out=8, Gnd, Vcc)



```
#include <Servo.h>
Servo myservo;

void setup() {
  Serial.begin(9600);
  pinMode(8, INPUT); //傳感器
  myservo.attach(9);
  myservo.write(90); // 關
}

void loop() {
  if (digitalRead(8) == LOW) {
    Serial.println("Obstacle on the way");
    myservo.write(0); // 開
  } else {
    Serial.println("All clear");
    myservo.write(90); // 關
  }
  delay(300);
}
```



## 2. 伺服馬達+超聲波

```
#include <Servo.h>
```

```
Servo myservo;
```

```
void setup() {
```

```
myservo.attach(9);
```

```
myservo.write(90); // 關
```

```
}
```

```
void loop() {
```

```
float d = ultrasonic(3, 2); // Trig 發動,  
// Echo 收回
```

```
if (d<10) myservo.write(0); // 開
```

```
delay(1000);
```

```
myservo.write(90); // 關
```

```
}
```



```
float ultrasonic (int Trig, int Echo) {
```

```
pinMode(Trig, OUTPUT);
```

```
pinMode(Echo, INPUT);
```

```
digitalWrite(Trig, LOW); delayMicroseconds(2);
```

```
digitalWrite(Trig, HIGH); delayMicroseconds(10);
```

```
digitalWrite(Trig, LOW);
```

```
float dist = pulseIn(Echo, HIGH) / 58; // cm
```

```
// 計算由 HIGH 轉 LOW 所需時間
```

```
return dist;
```

```
}
```

## 3. 伺服馬達+IRremote

```
// IR-Rx01.ino
```

```
#include <IRremote.h>
```

```
IRrecv irrecv(9);
```

// Rx 接收 IR(Rx 接收腳=9), Gnd, Vcc

```
decode_results results; // 接收到的訊息
```

```
void setup() {
```

```
irrecv.enableIRIn();
```

// 開始接收 IR 訊息

```
irrecv.blink13(true);
```

// 接收 IR 訊息會閃燈

```
}
```

```
void loop() {
```

```
if (irrecv.decode(&results)){ // 如果接收到新訊息
```

```
if(results.value==0xFF30CF) myservo.write(0); // 按 1 : 開
```

```
else if(results.value==0xFF6897) myservo.write(60); // 按 0 : 半開
```

```
delay(1000);
```

```
myservo.write(90);
```

// 關

```
irrecv.resume();
```

// 準備讀下一個 IR 訊號

```
}
```

```
}
```



<http://www.youtube.com/watch?v=mMARAsXOT2o?t=240>