

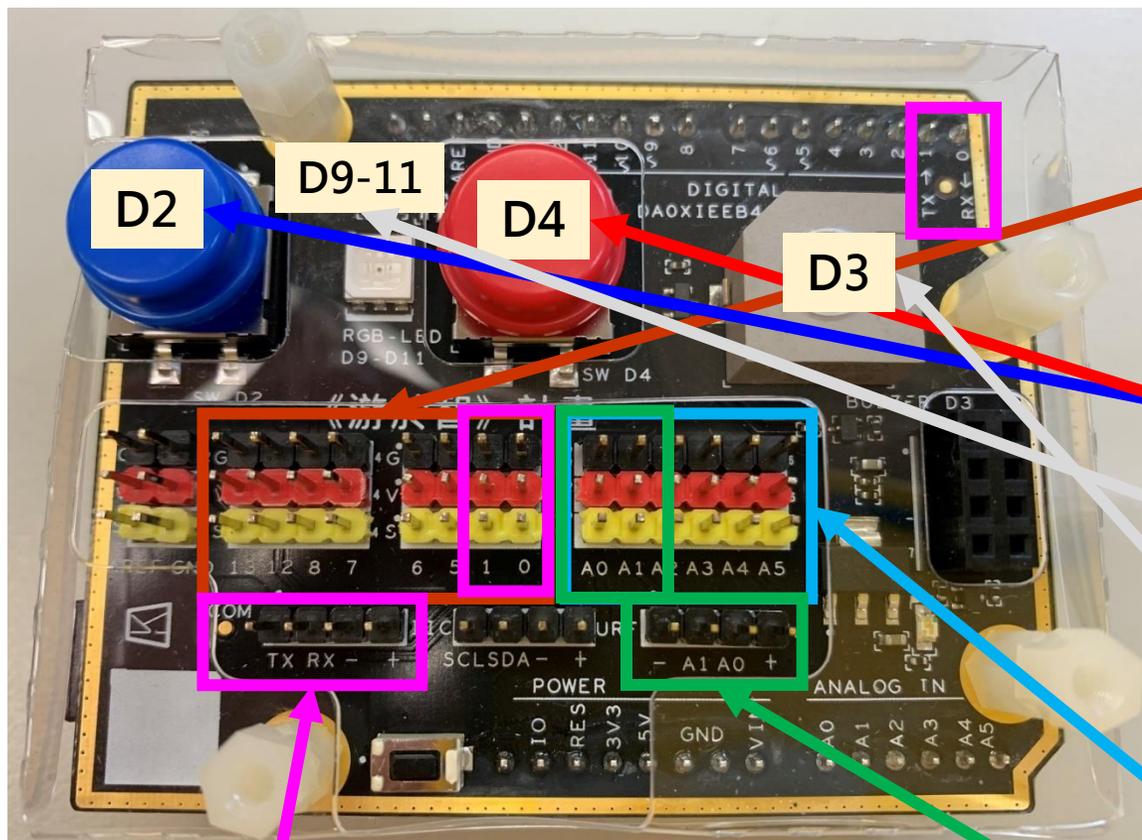


# 按鈕-按壓感測器



# 一、認識Quno、擴充板及Qblock

## 認識Quno擴充板\_腳位



### 1.數位腳位(Digital) :

D0、D1、D5、D6、D7、D8、D12、D13，其中D0、D1預設為序列埠，在連接USB線狀態時無法使用

按鈕使用D2、D4

RGB燈使用D9：綠、D10：紅、D11：藍

蜂鳴器使用D3

### 2.類比腳位(Analog) :

A0、A1、A2、A3、A4、A5

D0與D1通常當作資料的傳輸腳位，多在安裝藍芽感測器時才會用到。

超音波腳位

## 二、感測器實作 按鈕

- 按鈕分別使用D2、D4腳位。
- 先讀取按鈕的數值。

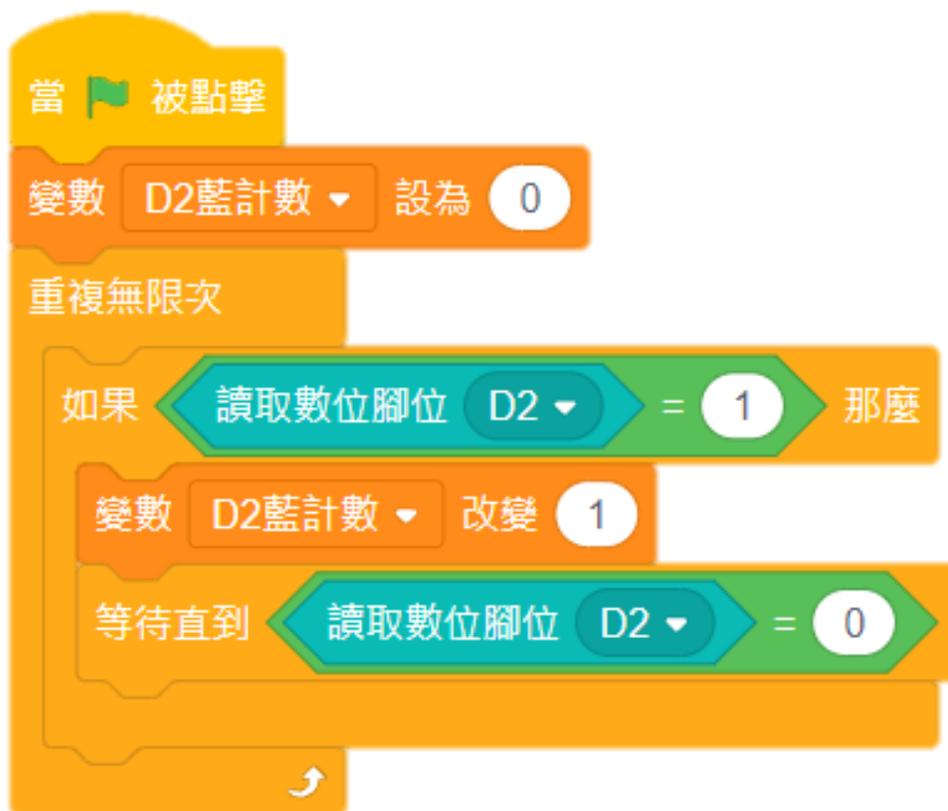


- **請問**：按鈕是輸入還是輸出裝置？

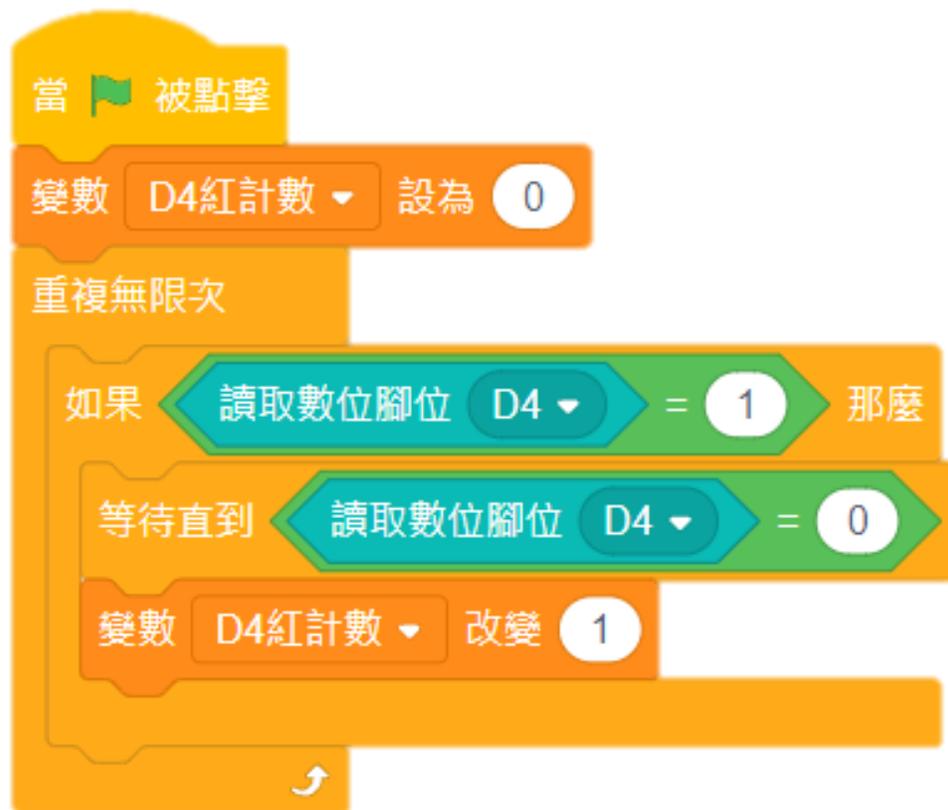
# 三、感測器交互運用

## 1. 按鈕計數器

- 判斷：請問這二個程式的差別？



## 2. 連接





# 按按內含光



# 三、感測器交互運用

## 按鈕 + RGB燈 [檔名：04.按鈕]

- 按鈕分別使用D2、D4腳位。
- **挑戰1**：按D2鈕，開藍燈。  
按D4鈕，關藍燈。
- 提示：

### 1. 應用讀取的數值



### 2. 一個條件判斷積木



# 三、感測器交互運用

## 按鈕 + RGB燈

- 按鈕分別使用D2、D4腳位。
- **挑戰2**：按下或放開D2鈕，開關藍燈。  
按下或放開D4鈕，開關紅燈。
- 提示：

### 1. 應用讀取的數值



### 2. 二個條件判斷積木



# 三、感測器交互運用

## 按鈕 + RGB燈 + 移動Q桑

- 按鈕分別使用D2、D4腳位。
- **挑戰3**：按下或放開D2鈕，開關藍燈。

Q桑往左移動10步

按下或放開D4鈕，開關紅燈。

Q桑往右移動10步

- 提示：

1. 應用讀取的數值

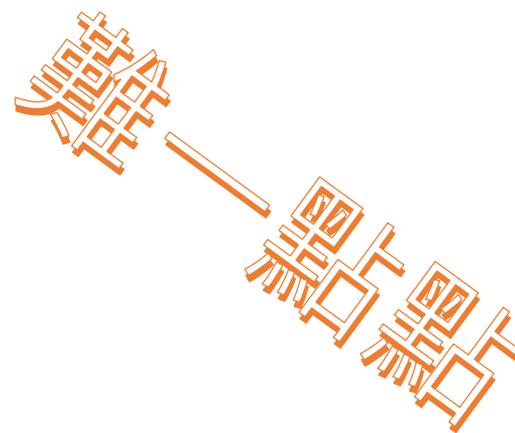


2. 二個條件判斷積木



# 三、感測器交互運用

## 按鈕+RGB燈



- 按鈕分別使用D2、D4腳位。
- **挑戰4**：按下D2鈕，開藍燈；放開D2鈕，關藍燈。

按下D4鈕，開紅燈；放開D4鈕，關紅燈。

同時按下D2與D4鈕，開綠燈；放開D2與D4鈕，關綠燈。

- 提示：

腳位 燈色	D2	且或	D4
藍	1	且	0
紅	0	且	1
綠	1	且	1



# 三、感測器交互運用 按鈕+RGB燈

- 按鈕分別使用D2、D4腳位。
- 挑戰4：參考程式。

當 被點擊

在即時模式下\_亮燈反應會比較慢

開始Arduino程式碼

以開始Arduino程式碼燒錄\_燈會即時反應喔!

當 被點擊

重複無限次

如果 讀取數位腳位 D2 = 1 且 讀取數位腳位 D4 = 0 那麼

設定PWM腳位 D11 (PWM:藍) 輸出為 15

否則

設定數位腳位 D11 (PWM:藍) 輸出為 低電位

如果 讀取數位腳位 D2 = 0 且 讀取數位腳位 D4 = 1 那麼

設定PWM腳位 D10 (PWM:紅) 輸出為 15

否則

設定數位腳位 D10 (PWM:紅) 輸出為 低電位

如果 讀取數位腳位 D2 = 1 且 讀取數位腳位 D4 = 1 那麼

設定PWM腳位 D9 (PWM:綠) 輸出為 15

否則

設定數位腳位 D9 (PWM:綠) 輸出為 低電位



# 蜂鳴器



# 二、感測器實作

## 蜂鳴器



兒歌簡譜

<https://moodle.tcavs.tc.edu.tw/moodle/mod/page/view.php?id=6971>

低音	Do	Re	Mi	Fa	So	La	Si
頻率	262	294	330	349	392	440	494
中音	Do	Re	Mi	Fa	So	La	Si
頻率	523	587	659	698	784	880	988
高音	Do	Re	Mi	Fa	So	La	Si
頻率	1046	1175	1318	1397	1568	1760	1976

音階--頻率對照表(單位：Hz)

## 二、感測器運作 蜂鳴器

設定蜂鳴器在腳位 **D3 (PWM)** 並播放聲音,頻率為 **Do, 262** 時間長度為 **500** ms

設定蜂鳴器在腳位 **D3 (PWM)** 並播放聲音,頻率為 **Do, 262**

設定蜂鳴器在腳位 **D3 (PWM)** 並停止播放

**Sol#, 415**

**La, 440**

**La#, 466**

**Si, 494**

**Do, 523**

**Do#, 554**

**Re, 587**

**Re#, 622**

**Mi, 659**

- 請您編一首歌和大家分享您的創作。
- **延伸**：自製電子琴  
請學生以電腦鍵盤製作電子琴

## 二、感測器實作 蜂鳴器



- 1.無源蜂鳴器是沒有正負之分的，類似於喇叭，只要在兩個針腳上加載不同的頻率的電信號就可以實現發聲，根據不同的頻率所發出的聲音也是不一樣的。
- 2.有源蜂鳴器是有正負之分的，只需要在兩個針腳上加上電壓信號就會發聲，發出的聲音音調單一、頻率固定。

低音	Do	Re	Mi	Fa	So	La	Si
頻率	262	294	330	349	392	440	494
中音	Do	Re	Mi	Fa	So	La	Si
頻率	523	587	659	698	784	880	988
高音	Do	Re	Mi	Fa	So	La	Si
頻率	1046	1175	1318	1397	1568	1760	1976

音階--頻率對照表(單位：Hz)

## 二、感測器實作 蜂鳴器

- 自製電子琴：  
電腦鍵盤製作電子琴  
參考程式

```
當 空白 ▾ 鍵被按下  
設定蜂鳴器在腳位 D3 (PWM) ▾ 並停止播放
```

```
當 a ▾ 鍵被按下  
設定蜂鳴器在腳位 D3 (PWM) ▾ 並播放聲音,頻率為 Do, 262 ▾ 時間長度為 500 ms
```

```
當 被點擊  
重複無限次  
  如果 1 ▾ 鍵被按下? 那麼  
    設定蜂鳴器在腳位 D3 (PWM) ▾ 並播放聲音,頻率為 Do, 262 ▾ 時間長度為 200 ms  
  否則  
    如果 2 ▾ 鍵被按下? 那麼  
      設定蜂鳴器在腳位 D3 (PWM) ▾ 並播放聲音,頻率為 Re, 294 ▾ 時間長度為 200 ms  
    否則  
      如果 3 ▾ 鍵被按下? 那麼  
        設定蜂鳴器在腳位 D3 (PWM) ▾ 並播放聲音,頻率為 Mi, 330 ▾ 時間長度為 200 ms  
      否則  
        [ ]
```

## 二、感測器實作 蜂鳴器

編寫一首歌和大家  
分享您的創作  
參考程式

當  被點擊

重複無限次

如果

讀取數位腳位

D2

= 1

那麼

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Mi, 330 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Do, 262 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Sol, 196 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Do, 262 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Re, 294 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Sol, 392 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Re, 294 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Mi, 330 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Re, 294 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Sol, 196 時間長度為 500 ms

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音,頻率為 Do, 262 時間長度為 500 ms



# 超音波感測器



## 二、感測器實作

### 超音波感測器

認識超音波（Ultrasound）：

- 你的耳朵能夠聽到聲音，是因為空氣震動，層層傳到耳蝸轉成聽覺訊號。
- 而人的耳朵能夠聽到的震動範圍大概是15~20,000 Hz這個區間，Hz是赫茲，也就是每秒震動幾次的意思。
- 超過這個範圍就叫超音波，人的耳朵是聽不到的。
- 某些動物，如狗、海豚、以及蝙蝠等等都有著超乎人類的耳朵，也因此可以聽到超音波。

## 二、感測器實作

### 超音波感測器

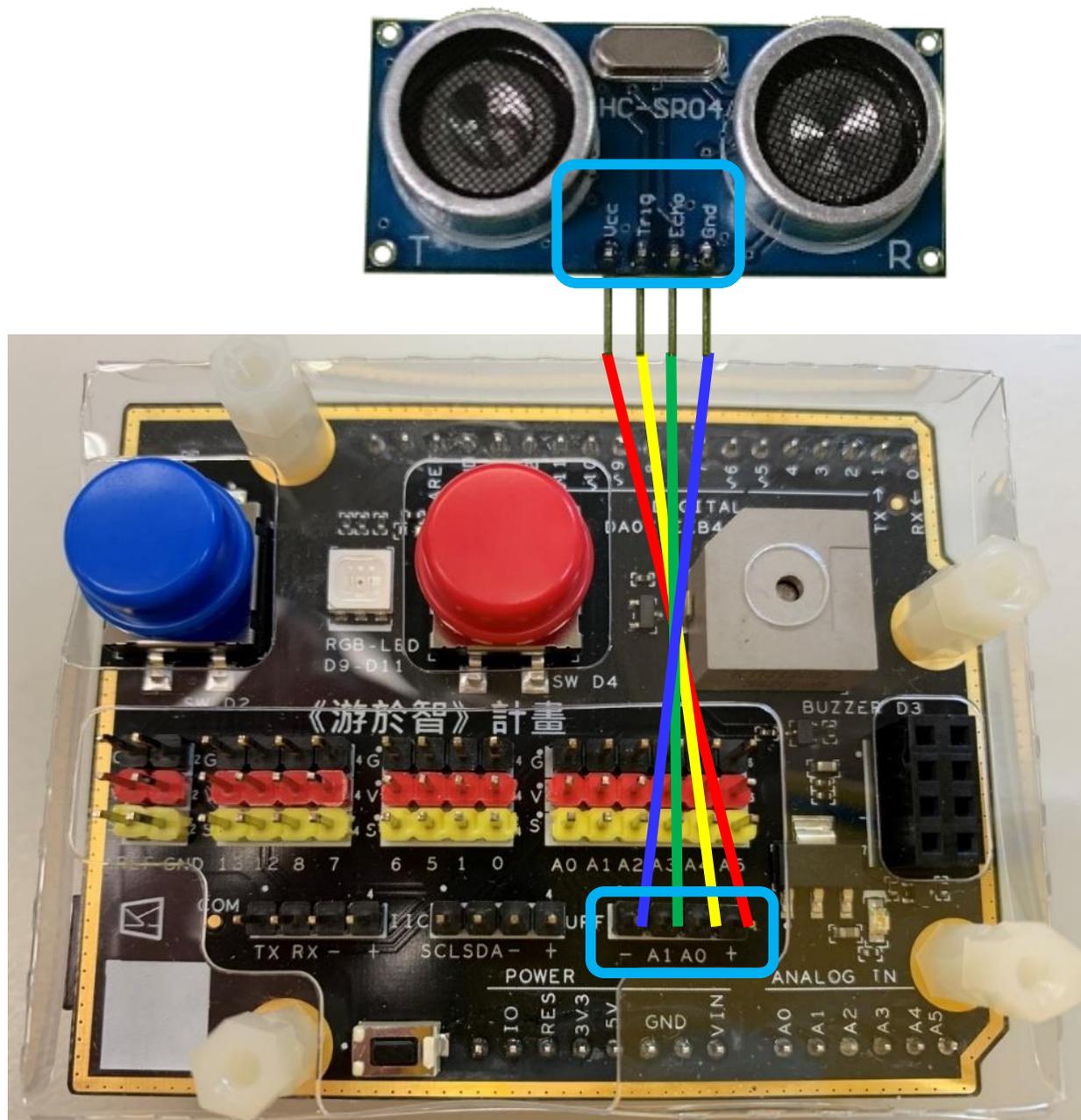


1. 超音波感測器(PING Sensor)是用來測量距離，利用聲波反射原理達到測距的功能。
  2. 最長的測量距離為**450cm**，最短**2cm**，超過這些距離就很可能測不到。
  3. 超音波也有角度的限制為**15度**，若角度大於**15度**，會收不到反射的聲波而測量不到，所以盡量以小角度去偵測障礙物，準確度會較高。
  4. 主要有一個超音波發射器和一個接收器組成。左邊**T**是發射器，會發出**40kHz**的超音波；右邊**R**是接收器，接收反射回來的超音波。
- 感測器有四個接腳：接線方式為**Vcc**接**5V**，**Gnd**接地，**Trig**（發送端）和**Echo**（接收端）。

## 二、感測器實作 超音波感測器

- 接線方式：
- Vcc接 +
- Trig接A0
- Echo接A1
- Gnd接 -

在安裝各式感測器時  
Qblock請先“斷開連接” 喔！



## 二、感測器實作 超音波感測器

- 首先先讀取超音波感測器數值。

提問：

- 數值變化太快，要如何修正？
- 如何將數值變成**整數**加上**長度單位**？



當 旗幟 被點擊

重複無限次

等待 1 秒

說出 字串組合 四捨五入數值 請取超音波距離感測器: trig腳位 A0 , echo腳位 A1 cm

## 二、感測器實作

### 超音波感測器

- 利用超音波感測器的數值。
- **挑戰1**：將偵測到的距使得角色由變小變大。
- 提示：



**挑戰2**：偵測距離越近角色愈大，距離越遠角色越小...



當  被點擊

尺寸設為 100 %

重複無限次

尺寸設為 300 - 請取超音波距離感測器: trig腳位 A0 ▾ , echo腳位 A1 ▾ \* 10 %



# 跨域-自動照明燈



# 三、感測器交互運用

## 超音波感測器 + RGB = 自動照明燈

- 自動照明：當有人靠近(ex:20cm)時，啟動照明。
- 距離小於20cm時，啟動照明。
- 距離大於20cm時，自動關燈。
- 提示：



# 三、感測器交互運用

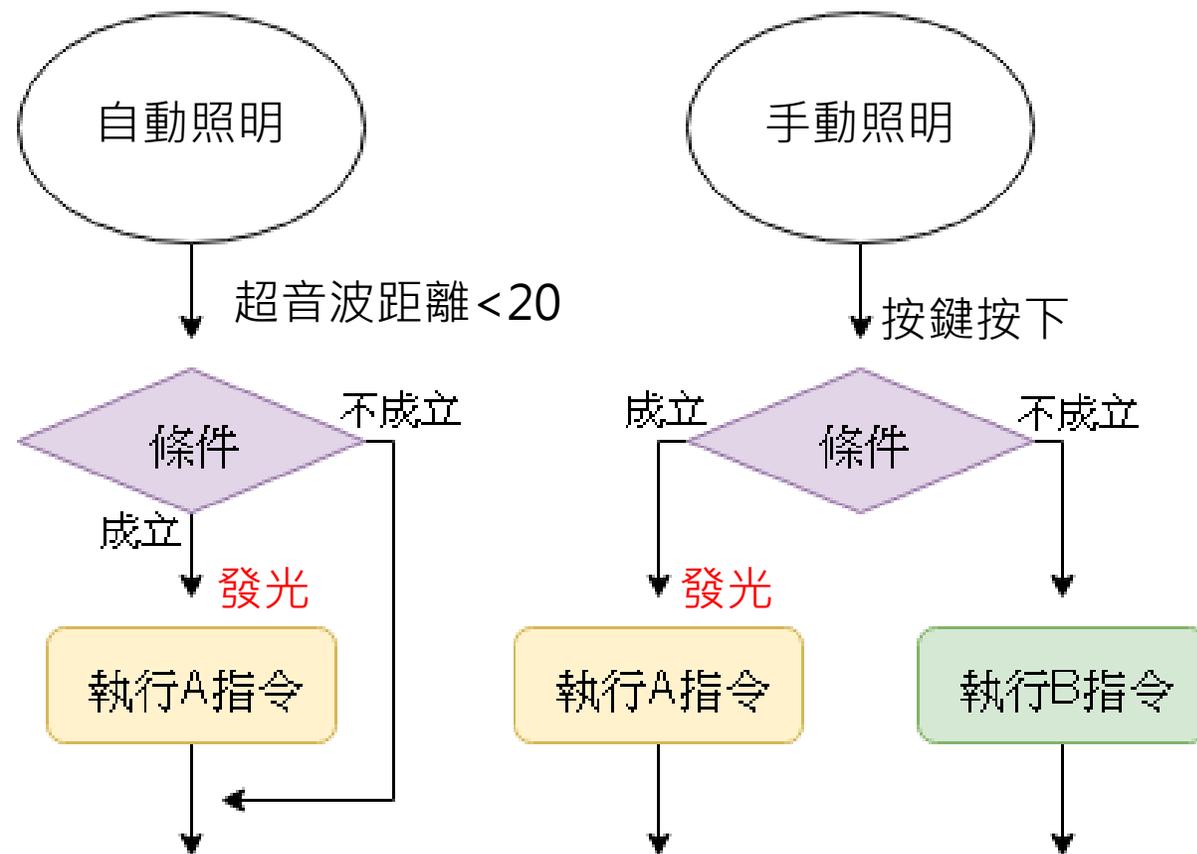
超音波感測器 + RGB + 按鍵 = 手動 + 自動感應燈

手動照明 + 自動照明：按鍵模組  
+ 超音波感測器 + RGB LED

整合--多功能照明燈：超音波  
+ RGB + 按鍵

功能：按下按鍵，開燈 + 距離  
20公分以內會自動開燈

提示：



# 三、感測器交互運用

超音波感測器 + RGB + 按鍵 = 手動 + 自動感應燈

手動照明 + 自動照明：按鍵模組 + 超音波感測器 + RGB LED

參考程式：

The image shows a Scratch script for controlling an RGB LED. The script starts with a '當被點擊' (When clicked) event block. This is followed by a '重複無限次' (Repeat forever) loop. Inside the loop, there is an '如果' (If) conditional block. The condition is '讀取數位腳位 D2 = 1 或 讀取超音波距離感測器: trig腳位 A0, echo腳位 A1 < 20' (Digital pin D2 is 1 or ultrasonic sensor distance is less than 20). If true, the script sets '設定數位腳位 D9 (PWM:綠) 輸出為 高電位' (Digital pin D9 to high). If false, it sets '設定數位腳位 D9 (PWM:綠) 輸出為 低電位' (Digital pin D9 to low). A red annotation '可利用變數來簡化程式...' (Can use variables to simplify the program...) points to the sensor reading block. A separate block above shows '變數 距離 設為 讀取超音波距離感測器: trig腳位 A0, echo腳位 A1' (Set variable 'distance' to sensor reading) and a '距離 < 20' comparison block.



# 跨域-倒車雷達



# 三、感測器交互運用

## 蜂鳴器+超音波感測器

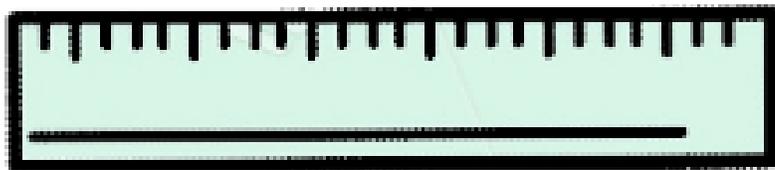


# 三、感測器交互運用

## 蜂鳴器+超音波感測器

模擬倒車雷達：

- 情境說明：當手掌(障礙物)靠近超音波模組(汽車)時，
- 蜂鳴器會發出警訊；距離越近，聲音越急促。
- 蜂鳴器模組(D3)。
- 距離35-50，警訊聲緩慢。距離20-35，警訊聲稍快。距離20-0，警訊聲急促。
- 提示：



# 三、感測器交互運用 蜂鳴器+超音波感測器

模擬倒車雷達參考程式：

當 被點擊

重複無限次

變數 距離 設為 四捨五入數值 讀取超音波距離感測器: trig腳位 A0 , echo腳位 A1

如果  $50 > \text{距離}$  且  $\text{距離} > 35$  那麼

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音, 頻率為 Do, 262 時間長度為 800 ms

如果  $35 > \text{距離}$  且  $\text{距離} > 20$  那麼

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音, 頻率為 Do, 523 時間長度為 500 ms

如果  $20 > \text{距離}$  且  $\text{距離} > 0$  那麼

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音, 頻率為 Do, 1046 時間長度為 50 ms

當 被點擊

重複無限次

變數 距離 設為 四捨五入數值 讀取超音波距離感測器: trig腳位 A0 , echo腳位 A1

如果  $50 > \text{距離}$  且  $\text{距離} > 35$  那麼

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音, 頻率為 Do, 262 時間長度為 800 ms

否則

如果  $35 > \text{距離}$  且  $\text{距離} > 20$  那麼

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音, 頻率為 Do, 523 時間長度為 500 ms

否則

如果  $20 > \text{距離}$  且  $\text{距離} > 0$  那麼

設定蜂鳴器在腳位 D3 (PWM) 並播放聲音, 頻率為 Do, 1046 時間長度為 50 ms

否則

請問有沒有第三種寫法...



# 跨域-空氣電子琴



# 三、感測器交互運用

## 蜂鳴器+超音波感測器

**挑戰3**：空氣電子琴：超音波偵測到不同距離會發出不同音頻。

如：

- 距離：0-3，發出音頻Do。
- 距離：3-6，發出音頻Re。
- 距離：6-9，發出音階Mi。
- ...依此類推，...-...，發出音頻...



# 三、感測器交互運用

## 蜂鳴器 + 超音波感測器 [檔名：02-1空氣電子琴]

### 空氣電子琴



# 三、感測器交互運用

## 蜂鳴器+超音波感測器

**挑戰3**：空氣電子琴：超音波偵測到不同距離會發出不同音頻。

如：

- 距離：0-3，發出音頻Do。
- 距離：3-6，發出音頻Re。
- 距離：6-9，發出音階Mi。
- ...依此類推，...-...，發出音頻...



# 三、感測器交互運用

## 蜂鳴器 + 超音波感測器 [檔名：02-1空氣電子琴]

### 空氣電子琴嚴謹的寫法

```
beginArduinoCode
  loop {
    waitUntil { digitalRead(D4) == 1 or digitalRead(D2) == 1 }
    repeatForever {
      setVariable(S, readUltrasonic(trig: A0, echo: A1))
      if (0 < S and S < 3) {
        setBuzzer(D3, PWM, Do, 262)
      } else {
        if (3 < S and S < 6) {
          setBuzzer(D3, PWM, Re, 294)
        } else {
          if (6 < S and S < 9) {
            setBuzzer(D3, PWM, Mi, 330)
          } else {
            if (9 < S and S < 12) {
              setBuzzer(D3, PWM, Fa, 349)
            } else {
              if (12 < S and S < 15) {
                setBuzzer(D3, PWM, Sol, 392)
              } else {
                if (15 < S and S < 18) {
                  setBuzzer(D3, PWM, La, 440)
                } else {
                  if (18 < S and S < 21) {
                    setBuzzer(D3, PWM, Si, 494)
                  } else {
                    setBuzzer(D3, PWM, stop)
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

The code is written in a block-based programming style. It starts with a 'begin Arduino code' block, followed by a 'loop forever' block. Inside the loop, it reads the distance from an ultrasonic sensor into a variable 'S'. Then, it uses a series of 'if-else' conditions to determine the frequency of the buzzer based on the distance 'S'. The conditions are: 0 < S < 3 (Do, 262), 3 < S < 6 (Re, 294), 6 < S < 9 (Mi, 330), 9 < S < 12 (Fa, 349), 12 < S < 15 (Sol, 392), 15 < S < 18 (La, 440), and 18 < S < 21 (Si, 494). If the distance is greater than 21, the buzzer is stopped.