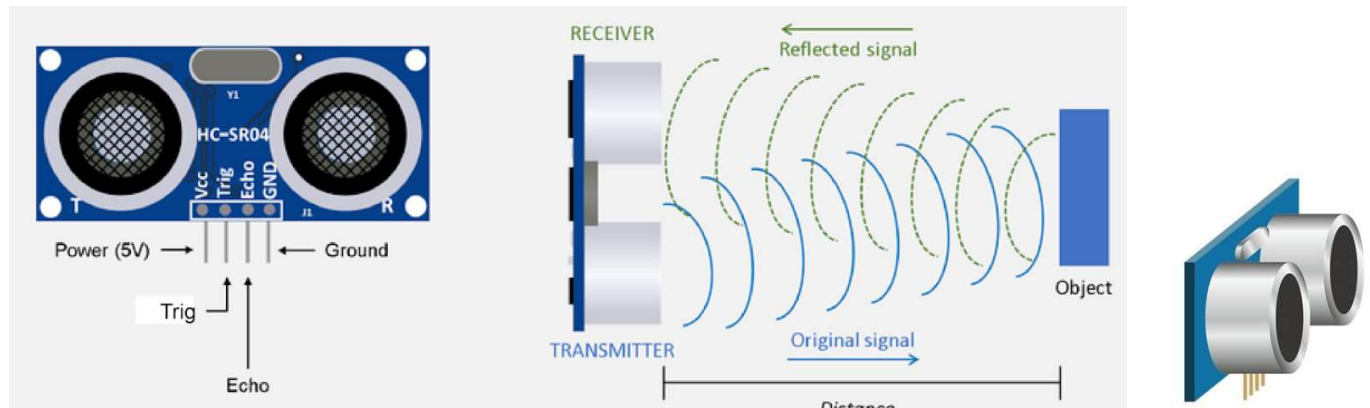


超音波感測器介紹

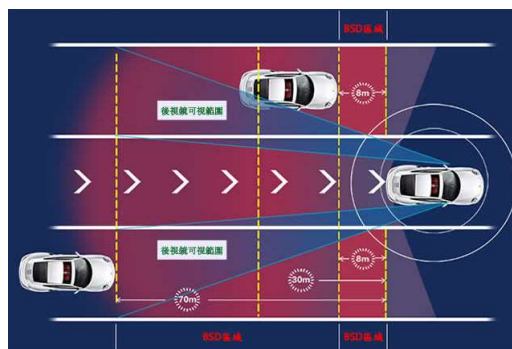
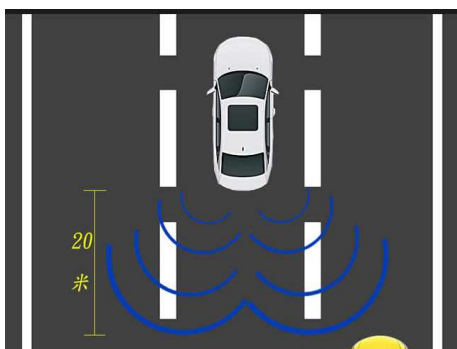
超音波距離感測器 HC-SR04，是利用超音波射出和反射的時間差，來測定感測器和障礙物之間的距離。如下圖所示：感測器有兩個筒狀物，其中一個為超音波**發射器** (Transmitter)，另一個為**接收器** (Receiver)。當發射的超音波遇到感測器前方的障礙物後，會被反射。接收器接到反射的超音波後，會計算出從發射到接收的時間差。在氣溫不變下，超音波在空氣中傳播速度是固定的。時間差乘以傳播速度，就可以得到感測器和障礙物之間的距離。



一般聲音速度為 340m/s，因此只要從我們發射超音波開始計時，直到接收到反射的超音波，取得這段時間，就可以開始計算距離了。只要將聲音速度*時間，就可以得到距離，但因為我們是發射出去跟反射回來，所以還要再除以 2，得到以下公式：

$$\text{Distance} = 340 * \text{duration} / 2 \text{ (m)}$$

超音波感測器應用

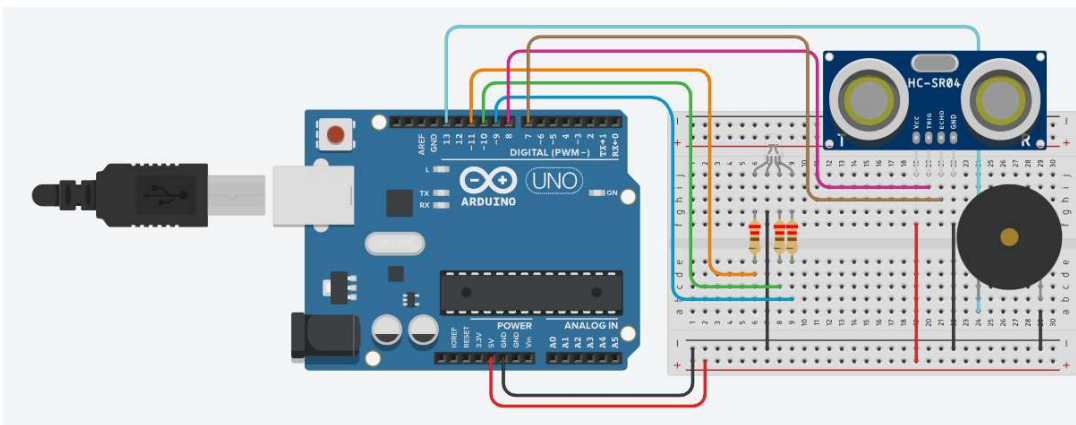


程式碼：

```
int trigPin = 11;    // 超音波感測器 Trig 腳接 Arduino pin 11
int echoPin = 12;   // 超音波感測器 Echo 腳接 Arduino pin 12
int speakerpin = 7; // 蜂鳴器 + 腳接 Arduino pin 7
long duration, cm;  // 宣告計算距離時，需要用到的兩個實數

void setup() {
  Serial.begin(9600);           // 設定序列埠監控視窗 和 Arduino 資料傳輸速率為 9600 bps (Bits Per Second)
  pinMode(trigPin, OUTPUT);     // Arduino 對外啟動距離感測器 Trig 腳，射出超音波
  pinMode(echoPin, INPUT);      // 超音波被障礙物反射後，Arduino 讀取感測器 Echo 腳的時間差
  pinMode(speakerpin, OUTPUT);  // Arduino 對蜂鳴器送出電壓，使其鳴叫
}

void loop()
{
  // 程式計算出距離值 cm
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  cm = (duration/2) / 29.1;
  Serial.print(cm);           // 印出距離值 cm 在序列埠監控顯示器 單位公分
  Serial.println(" cm");
  if (cm <= 30) {             // 距離小於 5 公分，蜂鳴器一直叫
    digitalWrite(speakerpin, HIGH);
    delay (20);
  }
  if (cm > 30 && cm <= 100) { // 距離介於 5 到 15 公分，蜂鳴器斷斷續續叫，每次 0.1 秒
    digitalWrite(speakerpin, HIGH);
    delay (100);
    digitalWrite(speakerpin, LOW);
    delay (100);
  }
  if (cm > 100){             // 距離大於 15 公分，蜂鳴器斷斷續續叫，每次 0.5 秒
    digitalWrite(speakerpin, HIGH);
    delay (500);
    digitalWrite(speakerpin, LOW);
    delay (500);
    } delay(10);
}
```

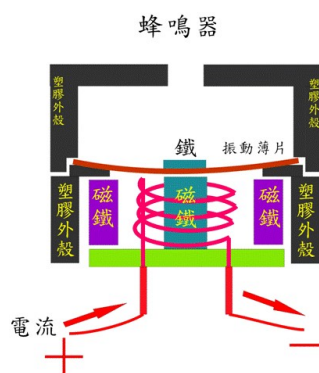


```

將 x 設定為 讀取單元 公分 上層發器接腳 8 回聲
如果 x < 20 則
  將接腳 11 10 9 中的 RGB LED 設為顏色 紅
  將接腳 13 設為 高
  等待 0.1 秒
  將接腳 13 設為 低
  等待 0.1 秒
如果 x ≥ 20 且 x ≤ 50 則
  將接腳 11 10 9 中的 RGB LED 設為顏色 藍
  將接腳 13 設為 高
  等待 0.5 秒
  將接腳 13 設為 低
  等待 0.5 秒
如果 x > 50 則
  將接腳 11 10 9 中的 RGB LED 設為顏色 綠
  將接腳 13 設為 高
  等待 2 秒
  將接腳 13 設為 低
  等待 2 秒

```

蜂鳴器介紹



有源蜂鳴器與無源蜂鳴器之差異: 這裡的"源"不是指電源。而是指震盪源。有源蜂鳴器往往比無源的貴，就是因為裡面多個震盪電路。

1. 有源蜂鳴器只能發出單頻率的聲音。

2. 而無源內部不帶震盪源，所以如果用直流信號無法令其鳴叫。必須用 **2K~5K 的方波** 去驅動它。無源蜂鳴器可以發出不同頻率的聲音。無源蜂鳴器的優點是：

(1) 便宜 (2) 聲音頻率可控，可以做出 “Do Re Mi Fa So La Si” 的效果。

有源蜂鳴器的優點是：程式控制簡單方便

程式碼（有源蜂鳴器）：

```
int buzzPin = 13;
void setup()
{
  pinMode(buzzPin, OUTPUT);
}
void loop()
{
  digitalWrite(buzzPin, HIGH); //有源蜂鳴器響起
  delay(2000);
  digitalWrite(buzzPin, LOW); //有源蜂鳴器關閉
  delay(2000);
}
```

程式碼（無源蜂鳴器）：

```
int buzzer = 3; // 用 Pin3 輸出方波至蜂鳴器(手機鈴聲)
void setup()
{
  pinMode(buzzer, OUTPUT);
}
void loop()
{
  for ( int i=0; i<10; i++ ) {
    tone(buzzer, 1000);
    delay(50);
    tone(buzzer, 500);
    delay(50);
  }
  noTone(buzzer);
  delay(2000);
}
```

《避免蜂鳴器發熱使用須知》：

蜂鳴器本身硬體原廠設計是「低電位 Low 觸發」，

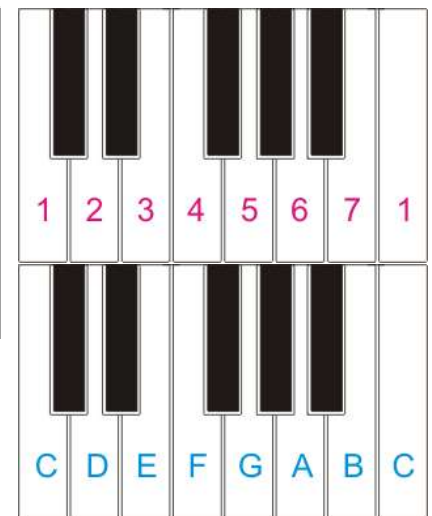
即使沒在跑蜂鳴器程式時，平時是處於低電位 Low，即會一直送電觸發蜂鳴器，因為是 DC 直流，所以沒聲音，但觸發蜂鳴器即會消耗電功率，物理能量守恆原理，電功率沒轉換成聲能就轉換成熱能，蜂鳴器就會發熱發燙。

因此使用蜂鳴器積木之前或之後，要把蜂鳴器腳位的電位設在高電位 High。

要讓蜂鳴器唱歌基本上就是讓蜂鳴器可以產生不同頻率的聲音，頻率的單位是赫茲(Hz)，也就是每秒振動多少次。標準被國際標準化組織認可，並於 1975 年確定下來。換句話說~我們只要讓一個東西能以每秒振動 **440** 下，聽到的聲音就是 A。

音階--頻率對照表(單位：Hz)

低音	C1 (Do)	D1 (Re)	E1 (Mi)	F1 (Fa)	G1 (Sol)	A1 (La)	B1 (Si)
頻率	262	294	330	349	392	440	494
中音	C2	D2	E2	F2	G2	A2	B2
頻率	523	587	659	698	784	880	988
高音	C3	D3	E3	F3	G3	A3	B3
頻率	1046	1175	1318	1397	1568	1760	1976



1. 利用 PWM 模擬不同的電壓輸出，讓蜂鳴器發出不同的音量。
2. 發出 Do(C) Re(D) Mi(E) Fa(F) Sol(G) La(A) Si(B) 7 個音調

音樂程式碼

每一秒控制無源蜂鳴器發出**兩隻老虎**音調的聲音。

tone()函數說明： tone(pin, frequency, duration) pin=訊號接口，frequency=頻率，duration=間距

```
int buzzer=3; //設定蜂鳴器接腳為第 3 孔
int duration = 500;
int aSo = 392; //低音 Sol
int bDo = 523; //中音 Do
int bRe = 587;
int bMi = 659;
int bFa = 698;
int bSo = 784;
int bLa = 880;
int bSi = 988;
int bDDo = 1046; //高音 Do
void setup()
{
  pinMode(buzzer,OUTPUT); //設定蜂鳴器為輸出
}
void loop()
{
```

```

tone(3,bDo,duration); //簡譜中音 1 //中音 Do
delay(600);
tone(buzzer,bRe,duration); //簡譜中音 2 //中音 Re
delay(600);
tone(buzzer,bMi,duration); //簡譜中音 2 //中音 Mi
delay(600);
tone(buzzer,bDo,duration);
delay(800);
tone(buzzer,bDo,duration);
delay(600);
tone(buzzer,bRe,duration);
delay(600);
tone(buzzer,bMi,duration);
delay(600);
tone(buzzer,bDo,duration);
delay(800);
tone(buzzer 3,bMi,duration);
delay(600);
tone(buzzer,bFa,duration);
delay(600);
tone(buzzer,bSo,duration);
delay(800);
tone(buzzer,bMi,duration);
delay(600);
tone(buzzer,bFa,duration);
delay(600);
tone(buzzer,bSo,duration);
delay(800);
tone(buzzer,bSo,duration);
delay(600);
tone(buzzer,bLa,duration);
delay(600);
tone(buzzer,bSo,duration);
delay(600);
tone(buzzer,bFa,duration);
delay(600);
tone(buzzer,bMi,duration);
delay(700);
tone(buzzer,bDo,duration);
delay(800);
tone(buzzer,bSo,duration);
delay(600);
tone(buzzer,bLa,duration);
delay(600);
tone(buzzer,bSo,duration);
delay(600);
tone(buzzer,bFa,duration);
delay(600);
tone(buzzer,bMi,duration);
delay(700);

```

1=C 4/4													
1	2	3	1	1	2	3	1	3	4	5-	3	4	5-
两只老虎				两只老虎				跑得快		跑得快			
<u>56</u>	<u>54</u>	3	1	<u>56</u>	<u>54</u>	3	1	2	5	1-	2	5	1-
一只没有耳朵				一只没有尾巴				真奇怪		真奇怪			

```

tone(buzzer,bDo,duration);
delay(800);
tone(buzzer,bDo,duration);
delay(700);
tone(buzzer,aSo,duration);
delay(700);
tone(buzzer,bDo,duration);
delay(800);
tone(buzzer,bDo,duration);
delay(700);
tone(buzzer,aSo,duration);
delay(700);
tone(buzzer,bDo,duration);
delay(800);
}

```

音樂程式碼

```

int buzzer = 3;
//音階頻率
int freq[] = {523,587,659,698,784}; //中音 Do Re Mi Fa Sol
//輸入歌曲音調
int melody[] = {5,3,3, 4,2,2, 1,2,3,4, 5,5,5, 5,3,3, 4,2,2, 1,3,5,5, 3, 2,2,2,2, 2,3,4, 3,3,3,3, 3,4,5, 5,3,3, 4,2,2, 1,3,5,5, 1};
//輸入節拍 4/4 , 1 拍 250ms , 2 拍 500ms , 4 拍 1000ms
int beat[] = {1,1,2, 1,1,2, 1,1,1,1, 1,1,2, 1,1,2, 1,1,2, 1,1,1,1, 4, 1,1,1,1, 1,1,2, 1,1,1,1, 1,1,2, 1,1,2, 1,1,1,1, 4};
void setup() {
  pinMode(buzzer,OUTPUT);
}

void loop() {
  for(int i = 0 ; i < 49 ; i++){
    tone(buzzer, freq[melody[i]-1]);
    delay(beat[i]*250);
    noTone(buzzer);
    delay(0);
  }
  delay(1000);
}

```

小蜜蜂

德國民謠
記譜：Sunghsi

1=C $\frac{4}{4}$ ♩=120

(1) | 5 3 3 - | 4 2 2 - | 1 2 3 4 | 5 5 5 - |

(5) | 5 3 3 - | 4 2 2 - | 1 3 5 5 | 3 - 0 0 |

(9) | 2 2 2 2 | 2 3 4 - | 3 3 3 3 | 3 4 5 - |

(13) | 5 3 3 - | 4 2 2 - | 1 3 5 5 | 1 - 0 0 ||

C調鍵盤