

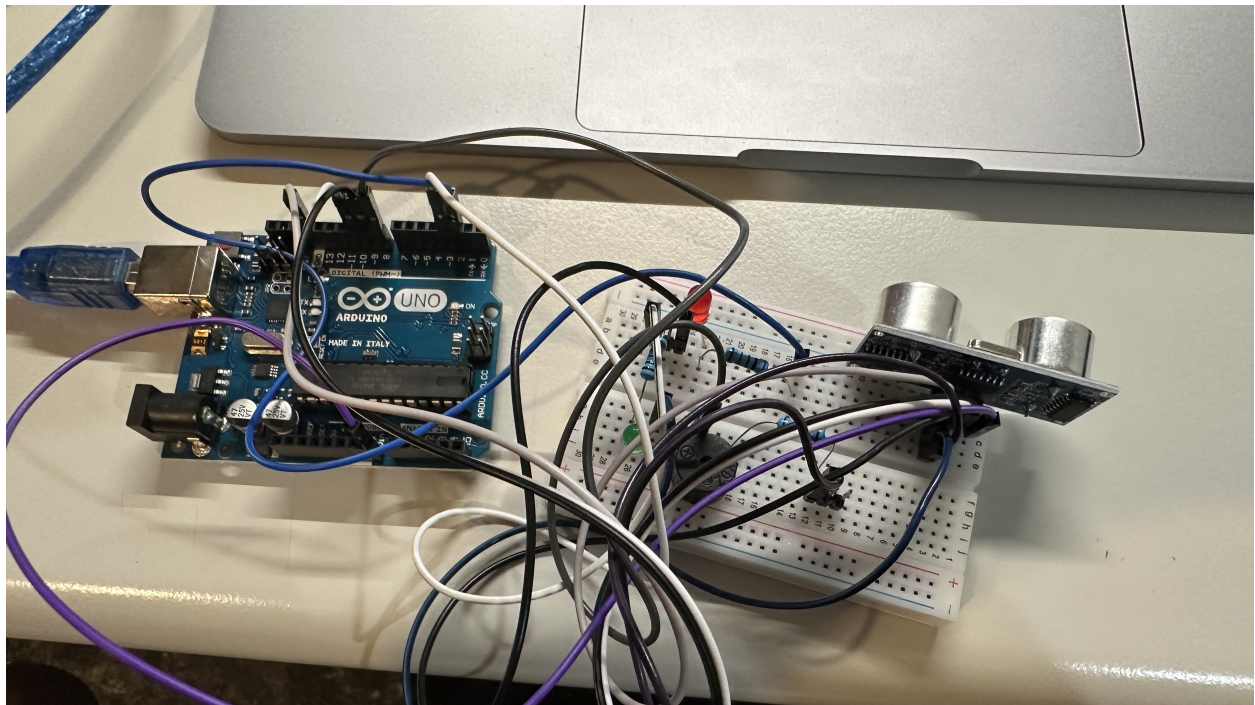
Introduction to Next-Generation Wireless Network - HW1

實作過程

程式碼

- B10915033 王新翌

實作過程



- 電路：整體電路架構就是將超音波感測器的四個腳接好，並將紅和綠LED及蜂鳴器從Digital Pin作為正極，透過元件和電阻接地即可。
- 程式：我直接使用Arduino官方提供的超聲波感測器程式碼計算距離，分別處理LED和蜂鳴器的控制。
 - LED：依照作業要求（20~50 cm）設定Digital Pin為High或Low，控制亮起或熄滅。


- 蜂鳴器：設定了一個變數buzzerFrequency，透過map(distance, 20, 50, 1000, 500)將距離和蜂鳴器頻率mapping，當距離越近頻率越高，就如同真實的倒車雷達設計。

程式碼

1. 設定Pin腳、輸入輸出。

```
1 // Pin definitions
2 const int trigPin = 9;    // HC-SR04 trigger pin
3 const int echoPin = 10;   // HC-SR04 echo pin
4 const int redLEDPin = 2;  // Red LED pin
5 const int greenLEDPin = 3; // Green LED pin
6 const int buzzerPin = 8;  // Buzzer pin
7
8 float duration, distance;
9
10 void setup() {
11   pinMode(trigPin, OUTPUT); // Set trigger pin as output
12   pinMode(echoPin, INPUT);  // Set echo pin as input
13   pinMode(redLEDPin, OUTPUT); // Set red LED pin as output
14   pinMode(greenLEDPin, OUTPUT); // Set green LED pin as output
15   pinMode(buzzerPin, OUTPUT); // Set buzzer pin as output
16
17   Serial.begin(9600); // Start serial communication at 9600 baud
18 }
```

2. 控制超音波感測器，送出訊號，計算距離，最後將結果print於Serial Monitor上。



```
1  digitalWrite(trigPin, LOW);
2  delayMicroseconds(2);
3  digitalWrite(trigPin, HIGH);
4  delayMicroseconds(10);
5  digitalWrite(trigPin, LOW);
6
7  // Read echo pulse duration and calculate distance in cm
8  duration = pulseIn(echoPin, HIGH);
9  distance = (duration * 0.0343) / 2;
10 Serial.print("Distance: ");
11 Serial.println(distance);
```

3. 透過距離設定LED紅綠燈和蜂鳴器

- a. 距離少於20紅燈亮起，蜂鳴器頻率2000，持續發出聲音。
- b. 距離介在20到50間綠燈亮起，蜂鳴器頻率介於1000~500間，間歇性發出聲音。
- c. 其他則不亮燈，蜂鳴器不發出聲音。



```
1 // Control LEDs and buzzer based on distance
2 if (distance < 20) {
3     // Red LED on, green LED off
4     digitalWrite(redLEDPin, HIGH);
5     digitalWrite(greenLEDPin, LOW);
6
7     // Buzzer continuously on, higher frequency (closer = more frequent)
8     tone(buzzerPin, 2000); // Higher frequency when very close
9
10 } else if (distance >= 20 && distance <= 50) {
11     // Green LED on, red LED off
12     digitalWrite(redLEDPin, LOW);
13     digitalWrite(greenLEDPin, HIGH);
14
15     // Buzzer beeps intermittently, frequency based on distance
16     int buzzerFrequency = map(distance, 20, 50, 1000, 500); // Adjust frequency based on distance
17     tone(buzzerPin, buzzerFrequency);
18     delay(500); // Pause for a while to create intermittent sound
19     noTone(buzzerPin); // Turn buzzer off for a short period
20     delay(500);
21
22 } else {
23     // Both LEDs off
24     digitalWrite(redLEDPin, LOW);
25     digitalWrite(greenLEDPin, LOW);
26
27     // Buzzer off
28     noTone(buzzerPin);
29 }
30
31 delay(100); // Short delay before repeating the loop
```