國立北港農工

資訊科專題製作

智慧偵測安全帽

組員：吳O承、吳O濂

目錄

組員與工作分配 0
研究動機 1
材料列表 2
材料圖片 3
電路圖 5
啟動流程圖 6
ARDUINO 程式碼 7
ESP32 程式碼 9
成品展示 11
成果演示 12
心得與反思 15
參考文獻 16

組員與工作分配

吳宗承:

擬定主題、撰寫研究動機及目的、蒐集資料、採購材料、程式設計、完成投片簡報、口頭發表

吳松濂:

擬定主題、蒐集資料、採購材料、程式設計、實際製作實習、完成投影片簡報、口頭發表

研究動機

近年來，國人出車禍，次數頻繁，設想安全帽可用在一般場景、霧天、火災、工地 現場等，可使受傷率和死亡率下降。

設想打造從物理層至 IOT 物聯網中裝置都能即時反應出當前狀況。

我們利用超聲波、紅外線測距兩路分別為發光二極體、蜂鳴器展現出距離遠近，及時在吵雜的及煙霧中也能聽見依靠光源及聲音判斷。

同時我們也使用 Line Notify 製作跌倒通知功能，只要一人戴著安全帽跌倒了，另一人就能即時收到通知。

材料列表

- 1.Arduino Uno R3 開發板
- 2.USB 線
- 3.杜邦線、麵包線
- 4.ESP32
- 5.麵包版
- 6.HC-SR04 超聲波測距模組
- 7.紅外線測距 感測器 GP2Y0A41SK0F
- 8.精密電阻
- 9.GY-61 ADXL335 三軸加速度模組
- 10.發光二極體
- 11.有源蜂鳴器

材料圖片

HC-SR04 超音波測距模組

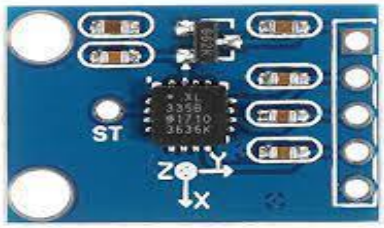
有源蜂鳴器

ESP32

GP2Y0A41SK0F 紅外線距離感測器

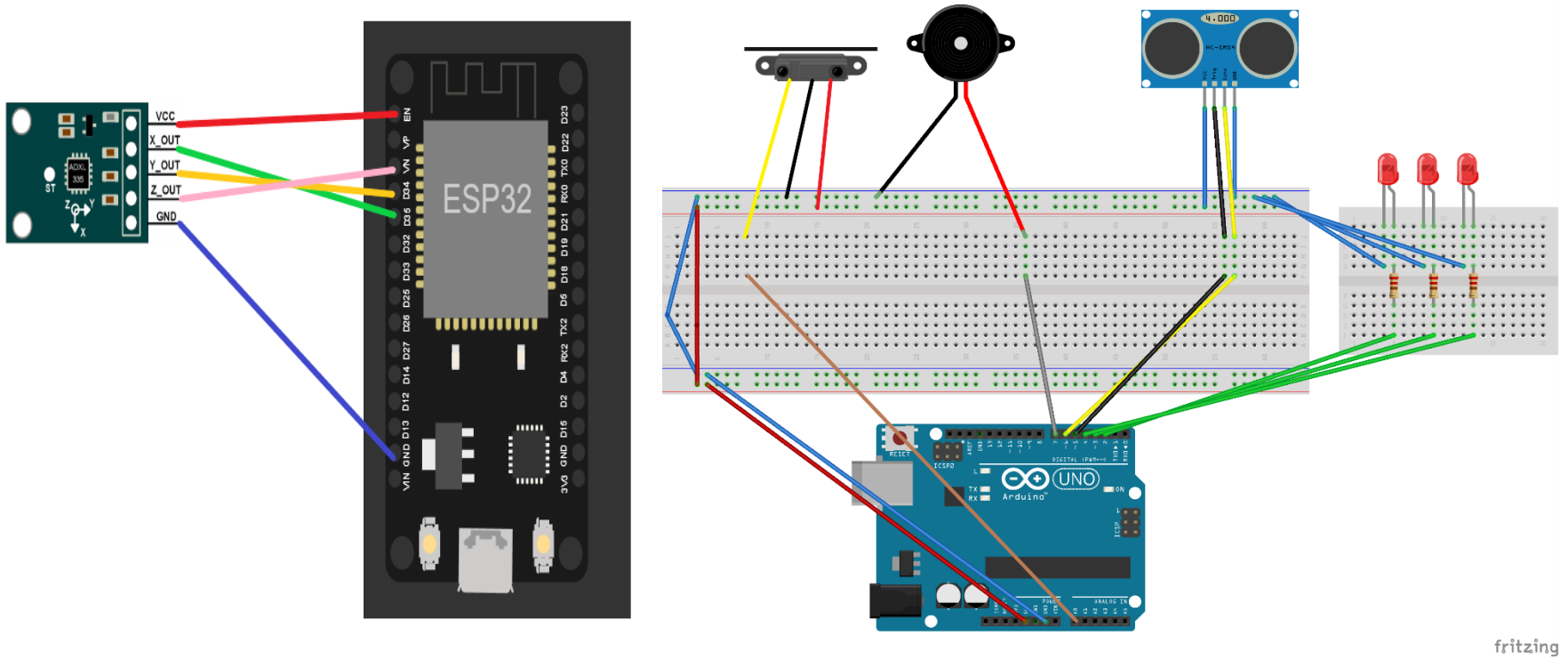
Arduino Uno R3

發光二極體

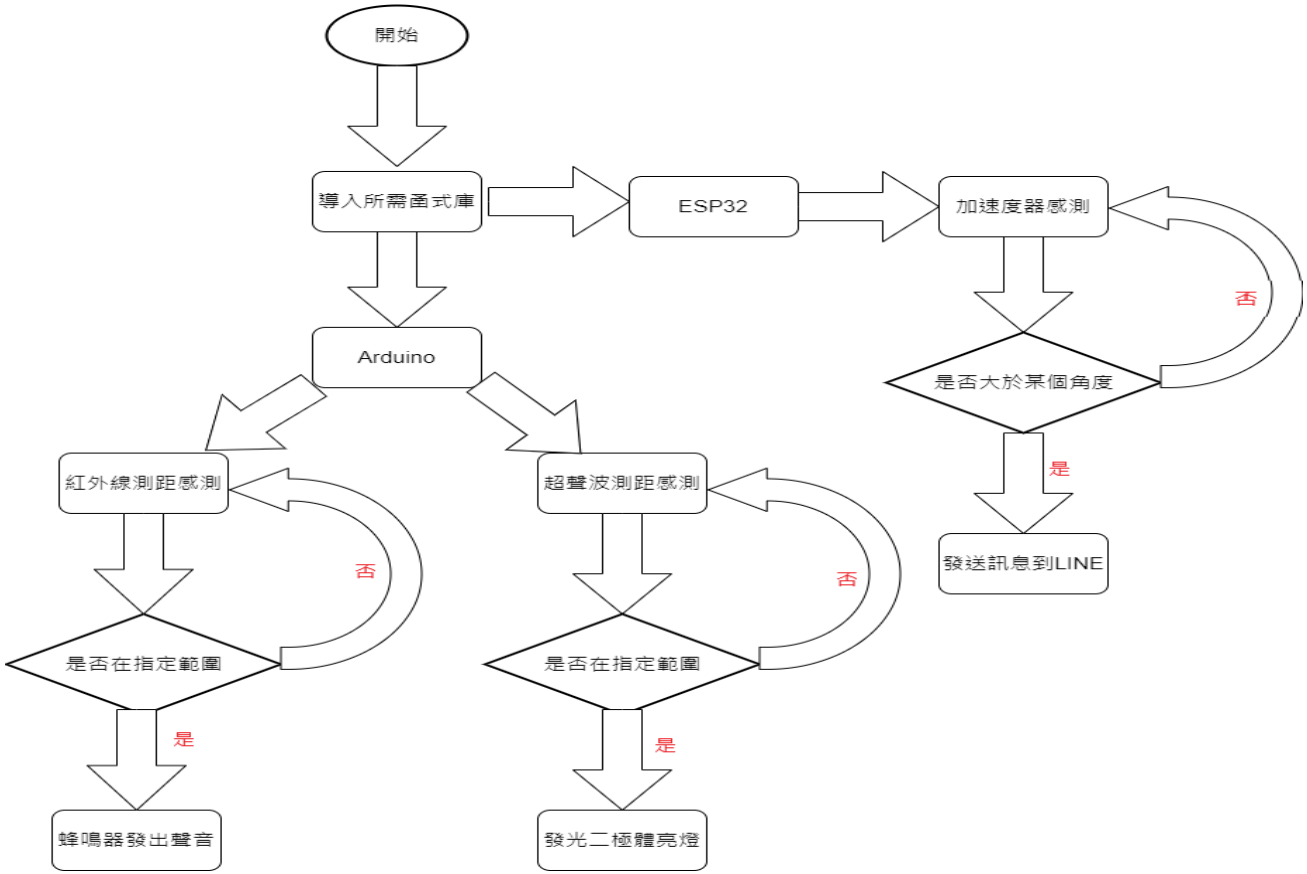


GY-61ADXL335 三軸重力加速度角度感測器

電路圖



啟動流程圖


開始
導入所需函式庫
ESP32
加速度器感測
否
是否大於某個角度
是
發送訊息到LINE
Arduino
紅外線測距感測
否
是否在指定範圍
是
蜂鳴器發出聲音
超聲波測距感測
否
是否在指定範圍
是
發光二極體亮燈

ARDUINO 程式碼

```
int trigPin = 5;    // 超音波感測器 Trig腳接 Arduino pin 5
int echoPin = 6;    //超音波感測器 Echo 腳接 Arduino pin 6
int LED2= 2;          // LED pin 2
int LED3= 3;         // LED pin 3
int LED4= 4;         // LED pin 4
long duration, cm ;  //定義單位

#define sensor A0
int speakerpin = 7;  //蜂鳴器 + 腳接 Arduino pin 7

void setup() {                                    //LED
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(speakerpin, OUTPUT);
}
void loop(){
  digitalWrite(trigPin, LOW);                        //LED
  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 LED
  Serial.println(" cm");


  if (cm > 1 && cm <= 20){              //LED
 digitalWrite(LED2,HIGH);              //距離介於1到20公分，LED2、3、4亮
  digitalWrite(LED3,HIGH);
   digitalWrite(LED4,HIGH);
 delay(10);
  }

  if (cm > 20 && cm <= 40){            //距離介於20到40公分，LED3、4亮
 digitalWrite(LED3,HIGH);
  digitalWrite(LED4,HIGH);
```

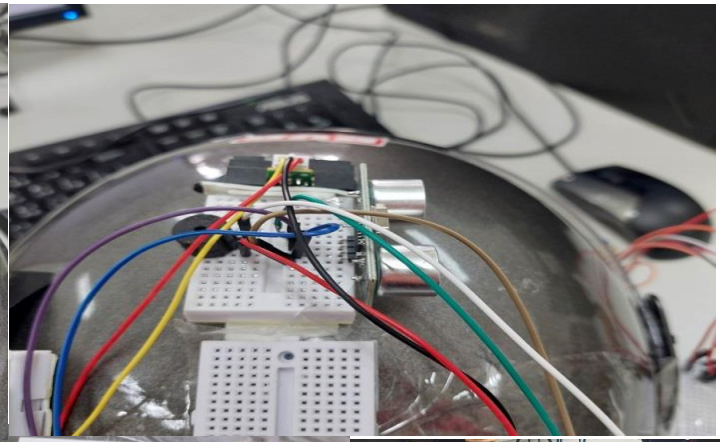
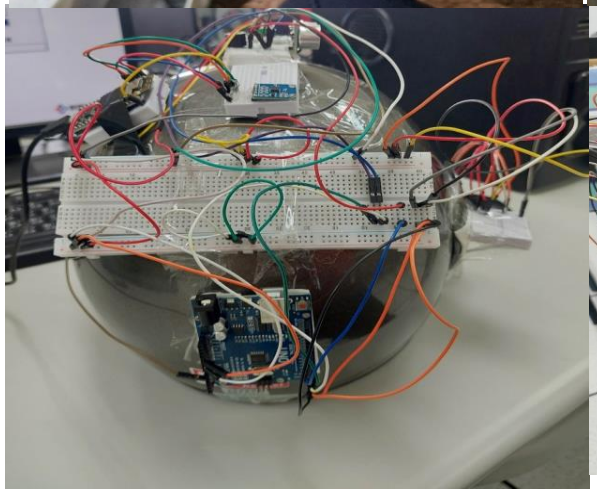
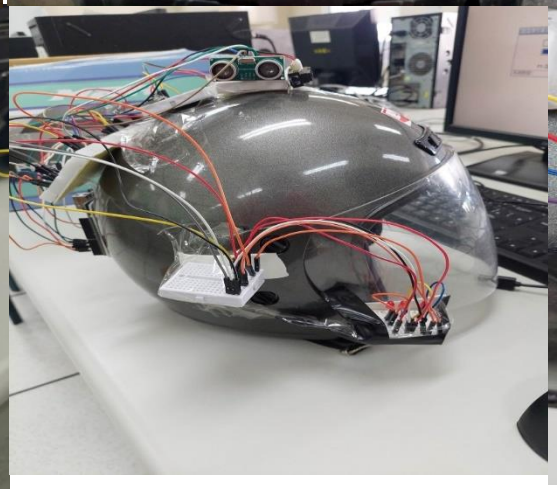
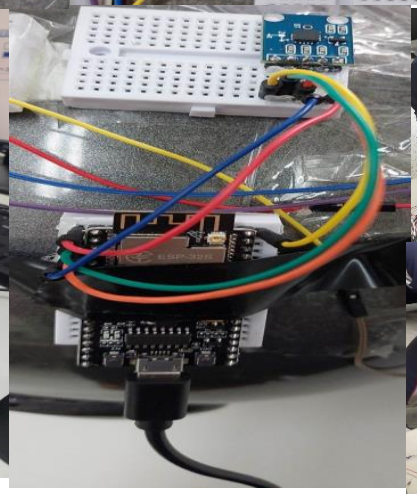
```
delay(10);
 }
  if (cm > 40 && cm <= 60){           //距離介於40到60公分，LED4亮
    digitalWrite(LED2,LOW);
digitalWrite(LED3,LOW);
 digitalWrite(LED4,HIGH);
delay(10);
  }
   if (cm > 60){
digitalWrite(LED2,LOW);               //距離大於60公分 全滅 RESET點
digitalWrite(LED3,LOW);
 digitalWrite(LED4,LOW);
delay(10);
}

   // 5v
 float volts = analogRead(sensor)*0.0048828125;  // value from sensor * (5/1024)    //以下皆為紅外線測距
 int distance2 = 13*pow(volts, -1); // worked out from datasheet graph

 Serial.println(distance2);   // print the distance


  if (distance2> 1 && distance2 <= 10) {                      //距離介於1到15公分 ，每次0.1秒
   digitalWrite(speakerpin, HIGH);
   delay (100);
     }

 if (distance2> 10 && distance2 <= 15) {             //距離介於15到30公分，蜂鳴器斷斷續續叫，每次0.3秒
   digitalWrite(speakerpin, HIGH);
   delay (300);
   digitalWrite(speakerpin, LOW);
   delay (300);
  }
      if (distance2>15){
   digitalWrite(speakerpin, LOW);                            //RESET
 delay (500);
     }

 delay(100);
   }
```

ESP32 程式碼

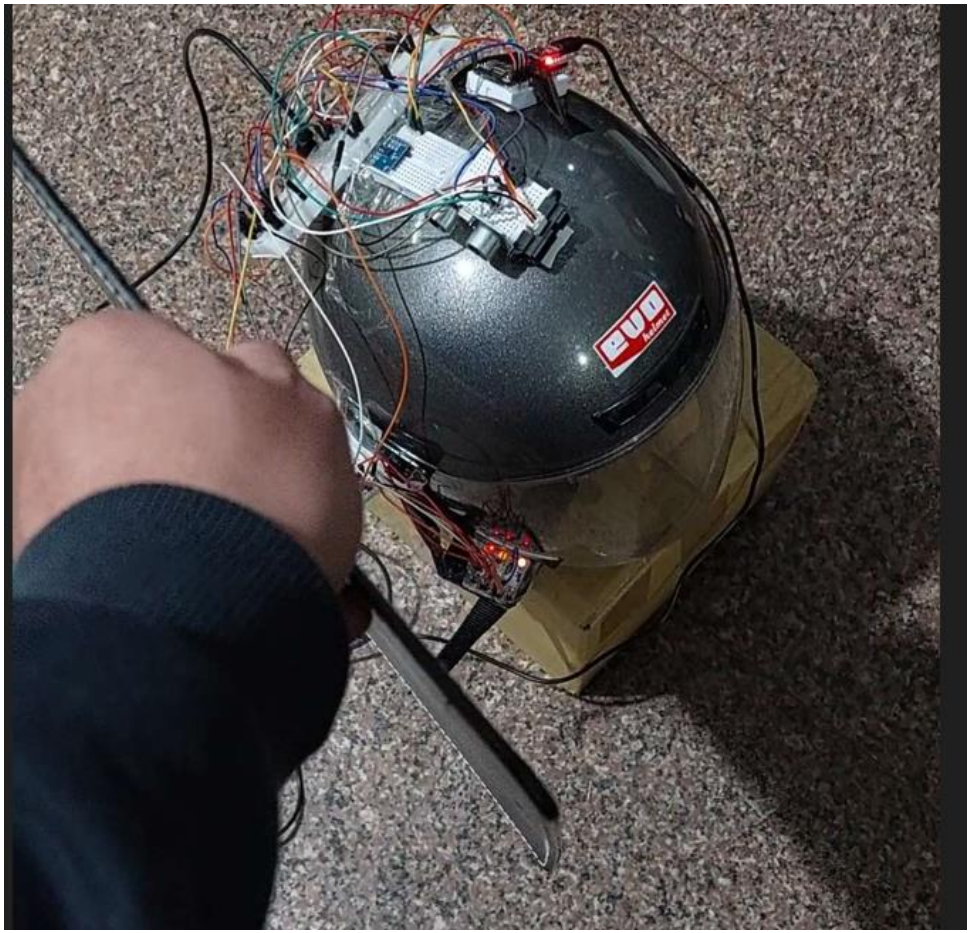
```
#include <WiFi.h>  //ESP32為WiFi.h
#include <WiFiClient.h>
#include <TridentTD_LineNotify.h>
#define LINE_TOKEN "1yNVyE45uEqMAXJL1wOOFiQETa6ekRHCbd5yLxVXt5z"


// Set WiFi credentials
#define WIFI_SSID "Galaxy A52 5G"
#define WIFI_PASS "asql6800"
WiFiClient client; //網路連線物件

const byte xpin = 34;                  // x-axis of the accelerometer
const byte ypin = 39;                  // y-axis
const byte zpin = 36;                  // z-axis (only on 3-axis models)

void setup() {
  // initialize the serial communications:
    Serial.begin(9600);  // 設定速率
  analogReadResolution(9);  // Default resolution is 12 bits. Choose between 9-12.
                            // 9-bit resolution will produce readings from 0 to 512


// 進行無線網路連接
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASS);

//等待無線網路連接
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(100);
    Serial.print(".");
  }
  LINE.setToken(LINE_TOKEN); //設定Line Token權杖

  LINE.notify("已開機"); //發送訊息到Line上面
}
```

```
37
38  void loop() {
39    // print the sensor values:
40    Serial.print(analogRead(xpin));
41    // print a tab between values:
42    Serial.print("\t");
43    Serial.print(analogRead(ypin));
44    // print a tab between values:
45    Serial.print("\t");
46    Serial.print(analogRead(zpin));
47
48    if (analogRead(zpin) < 230)
49    {
50      LINE.notify("跌倒警報");
51      delay(50);
52    }
53
54    if (analogRead(xpin) <= 190 )
55    {
56      LINE.notify("跌倒警報");
57      delay(50);
58    }
59
60    if (analogRead(ypin) <= 190)
61    {
62      LINE.notify("跌倒警報");
63      delay(50);
64    }
65
66    Serial.println();
67    // delay before next reading:
68    delay(100);
69  }
70
```

成品展示

成果演示

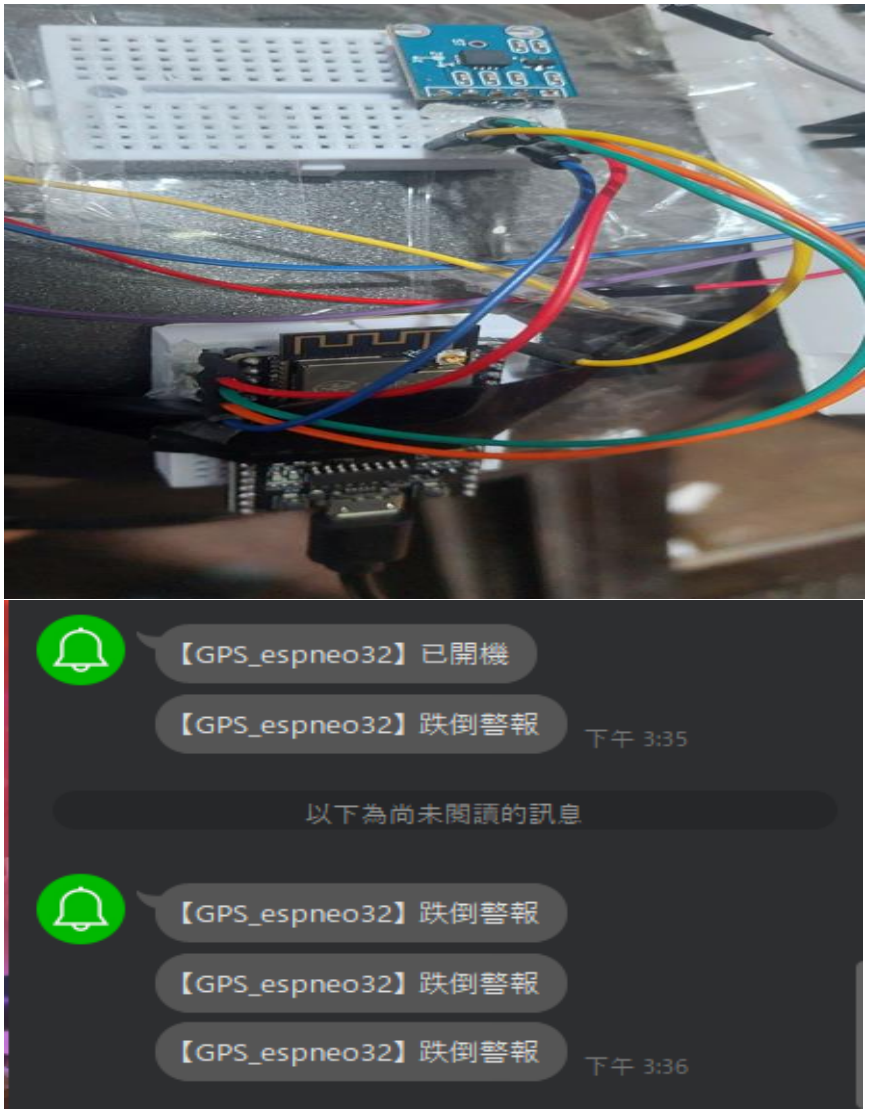


安全帽右路使用 HC-SR04 搭配 LED，距離越近，亮的燈數越多。如超過最大距離設定為距離，則 LED 全暗。

影片展示：HC SR04+LED 測距

安全帽前方使用紅外測距搭配有源蜂鳴器，根據識別不同的距離，所發出的聲音頻率不同。

影片展示：紅外線測距+蜂鳴器



以 ESP32 搭配 ADXL-335，在裝置開機後，搭配無線網路實現跌倒警報，當安全帽有著大角度的傾斜，便會透過 Line Notify 傳送給手機。

心得與反思

這是我們第一次做專題，題目想了很久，我說不然做智慧安全帽，我們覺得可行就做了。一開始做的其實比較沒有方向性，連帶的也衍生出許多問題：討論出現摩擦、明確的分工等。但我們盡力解決、克服了這些。

關於這次的專題，我們學到最重要的是如何分工合作、討論問題，這些看似簡單的事情實則只有自己實作了之後才知道有多不容易。

還有許多可以完善的點 ，像是一開始，我們不知道從哪開始下手，
耽誤了很多時間，差點來不及， 而這個專題題目乏人問津， 但我們做了。所以這裡引用愛因斯坦的一句話:

「唯有那些異想天開的人，才能完成不可能的事。」

這個安全帽目前而言是概念品，有很多功能還不是那麼完善 ，日後可增加更多路測距及更多 IOT 應用等。

參考文獻

《Arduino 入門》第一篇：認識 Arduino

超音波距離感測器 HC-SR04

三軸加速度角度感測器 Accelerometer

NodeMCU-32S Windows Arduino 環境設置與驅動程式安裝

Esp32+Line

使用超音波距離感測器和 Arduino + LED 進行距離測量

How to use the Sharp IR Sensor

《Arduino 入門》第六篇：有源蜂鳴器、無源蜂鳴器