

投稿類別：資訊類

篇名：
測速照相

作者：
尤紆豐。公立海青工商。高三善班
王宜平。公立海青工商。高三善班
許詠傑。公立海青工商。高三善班
張致嘉。公立海青工商。高三善班

指導老師：
蔡啟榮

臺、前言

一、研究動機

研究動機:根據統計在台灣 2022 年 1~9 月車禍的事故總件數就高達 271,660 件，30 天內死亡人數:2,293、受傷人數 360,698，由於目前的測速照相只有單純的拍照所以我們希望可以設計一個能夠拍照並且還能發出蜂鳴聲來達到警告效果的測速照相。

二、研究目的

- (一)利用兩個超音波來偵測物體移動的時間差。
- (二)利用蜂鳴器發出蜂鳴聲來達到警惕的效果。
- (三)利用 ESP32Cam 拍下超速的物體。
- (四)利用 ESP32Cam 搭配 LINE Notify 將照片資料上傳到手機。
- (五)利用 LCD 顯示 cm/s。
- (六)利用 ESP32 進行速度的運算。

貳、文獻探討

在我們製作的過程中我們用到了 Arduino 和一些零件去實現作品功能，以下是相關資料。

一、Arduino

Arduino 是一個硬體和軟體的開源電子平台，它提供了易學易用的整合開發環境(Interactive Development Environment, IDE)，其特色為開發簡單，參考資料多。要開發 Arduino 專題需要準備 Arduino 硬體及 Arduino 軟體。(第一章 Arduino，2022)

二、HC-SR04

超音波模組 HC-SR04，算是常見的模組，特別是用在智慧車等專案中，用來測量前方障礙物的距離。它的運作原理很簡單，模組會送出 8 個 40khz 的方波，如果前方有障礙物，信號就會返回，模組收到信號後，再利用返回的時間，去計算該障礙的距離。(傑森創工 2019)

三、蜂鳴器

蜂鳴器（英語：Buzzer）是產生聲音的信號裝置[1]，有機械型、機電型及壓電型。蜂鳴器在電路中用字母「H」或「HA」（舊標準用「FM」、「ZZG」、「LB」、「JD」等）表示。蜂鳴器的典型應用包括警笛、報警裝置、火災警報器、防空警報器、防盜器、定時器等。（維基百科，自由的百科全書，2022）

四、ESP32 cam

ESP32-CAM 是一款基於 ESP32 的低成本開發板，配有板載攝像頭，體積小巧。它是物聯網應用，原型構建和 DIY 項目的理想解決方案。該板集成了 WiFi，傳統藍牙和低功耗 BLE，以及 2 個高性能 32 位 LX6 CPU。它採用 7 級流水線架構，片上傳感器，霍爾傳感器，溫度傳感器等，主頻調節範圍為 80MHz 至 240MHz。（台灣物聯科技，2022）

五、LCD

液晶顯示器（Liquid Crystal Display，簡稱 LCD）為平面薄型的顯示裝置，由一定數量的彩色或黑白畫素組成，放置於光源或者反射環境光源。（維基百科，自由的百科全書，2022）

參、研究方法

一、文獻分析:

透過網路、文章、影片，去蒐集和測速照相有關的資料做分析與統整，再根據獲得的資料去尋找符合功能的零件。

二、研究過程

(一) 主題討論:

透過網路上現有的範例去作參考及修改，再搭配實事議題去做規劃。

(二)零件採購:

我們按照主題所需實現的功能，去尋找符和相關功能的零件。

(三)規劃實體電路圖:

我們依據 esp32 單晶片微控制器所需要搭配零件的電路去搜尋相關的資料並整合設計出實體電路。

(四)程式設計:

1.我們透過 Arduino IDE 撰寫並編譯出所要的程式。

圖一:ESP32 測速程式圖

```
1 // Santi&be Youtube
2 // https://www.youtube.com/watch?v=I-Hb0RWwMVY
3
4 #include <WiFi.h>
5 #include "esp_camera.h"
6 #include "esp_system.h"
7
8 hw_timer_t *timer = NULL;
9 void IRAM_ATTR resetModule() {
10   ets_printf("reset\n");
11   esp_restart();
12 }
13
14 #include <TridentTD_LineNotify.h>
15 #define SSID "Cwanghome" //WiFi name
16 #define PASSWORD "#Asdfaxcv1234" //PASSWORD
17 #define LINE_TOKEN "Zxxn12NR9aGVywsIHkEK4ddJe4TRRAWanGaxF3pJ2Ym"
18
19 // Pin definition for CAMERA_MODEL_AI_THINKER
20 #define PWDN_GPIO_NUM 32
21 #define RESET_GPIO_NUM -1
22 #define XCLK_GPIO_NUM 0
23 #define SIOD_GPIO_NUM 26
24 #define SIOC_GPIO_NUM 27
25
26 #define Y9_GPIO_NUM 35
27 #define Y8_GPIO_NUM 34
28 #define Y7_GPIO_NUM 39
29 #define Y6_GPIO_NUM 36
30 #define Y5_GPIO_NUM 23
31 #define Y4_GPIO_NUM 19
32 #define Y3_GPIO_NUM 18
33 #define Y2_GPIO_NUM 5
34 #define VSYNC_GPIO_NUM 25
35 #define HREF_GPIO_NUM 23
36 #define PCLK_GPIO_NUM 22
37
38 const int Led_Flash = 4;
39 const int Led_run = 13;
40 int PIR_Sensor = 12;
41 boolean startTimer = false;
42 unsigned long time_now=0;
43 int time_capture=0;
44
45 void setup() {
46   Serial.begin(115200);
47   while (!Serial) { ; }
48   pinMode(Led_Flash, OUTPUT);
49   pinMode(Led_run, OUTPUT);
50   WiFi.begin(SSID, PASSWORD);
51   Serial.printf("WiFi connecting to %s\n", SSID);
52   while(WiFi.status() != WL_CONNECTED) { Serial.print("."); delay(400); }
53   Serial.printf("\nWiFi connected\nIP : ");
54   Serial.println(WiFi.localIP());
55   LINE.setToken(LINE_TOKEN);
56
57   timer = TimerBegin(0, 80, true); //timer 0, div 80MHz
58   timerAttachInterrupt(timer, resetModule, true);
59   timerAlarmWrite(timer, 20000000, false); //set time in us 15s
60   timerAlarmEnable(timer); //enable interrupt
61
62   camera_config_t config;
63   config.ledc_channel = LEDC_CHANNEL_0;
64   config.ledc_timer = LEDC_TIMER_0;
65   config.pin_d0 = Y2_GPIO_NUM;
66   config.pin_d1 = Y3_GPIO_NUM;
67   config.pin_d2 = Y4_GPIO_NUM;
68   config.pin_d3 = Y5_GPIO_NUM;
69   config.pin_d4 = Y6_GPIO_NUM;
70   config.pin_d5 = Y7_GPIO_NUM;
71   config.pin_d6 = Y8_GPIO_NUM;
72   config.pin_d7 = Y9_GPIO_NUM;
```

2.參考網路上相關的 ESP32can 拍照程式修改。

圖二:EPS32CAM 照相程式圖

```

// Santi&be Youtube
// https://www.youtube.com/watch?v=I-HbdRWwMUY

#include <WiFi.h>
#include "esp_camera.h"
#include "esp_system.h"

hw_timer_t *timer = NULL;
void IRAM_ATTR resetModule() {
    ets_printf("reboot\n");
    esp_restart();
}

#include <TridentTD_LineNotify.h>
#define SSID "Cwanghome" //WiFi name
#define PASSWORD "88Andfzxcv1234" //PASSWORD
#define LINE_TOKEN "2xrni2NR9a0VysIHkEK4dJc4TRBAWmGsrFJpJ2Ym"

// Pin definition for CAMERA_MODEL_AI_THINKER
#define FWEN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define STOD_GPIO_NUM 26
#define STOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

const int Led_Flash = 4;
const int Led_run = 13;
int PIR_Sensor = 12;
boolean startTimer = false;
unsigned long time_now=0;
int time_capture=0;

void setup() {

    Serial.begin(115200);
    while (!Serial) { ; }
    pinMode(Led_Flash, OUTPUT);
    pinMode(Led_run, OUTPUT);
    WiFi.begin(SSID, PASSWORD);
    Serial.printf("WiFi connecting to %s\n", SSID);
    while(WiFi.status() != WL_CONNECTED) { Serial.print("."); delay(400); }
    Serial.printf("\nWiFi connected\nIP : ");
    Serial.println(WiFi.localIP());
    LINE.setToken(LINE_TOKEN);

    timer = timerBegin(0, 80, true); //timer 0, div 80Mhz
    timerAttachInterrupt(timer, &resetModule, true);
    timerAlarmWrite(timer, 2000000, false); //set time in us 15s
    timerAlarmEnable(timer); //enable interrupt

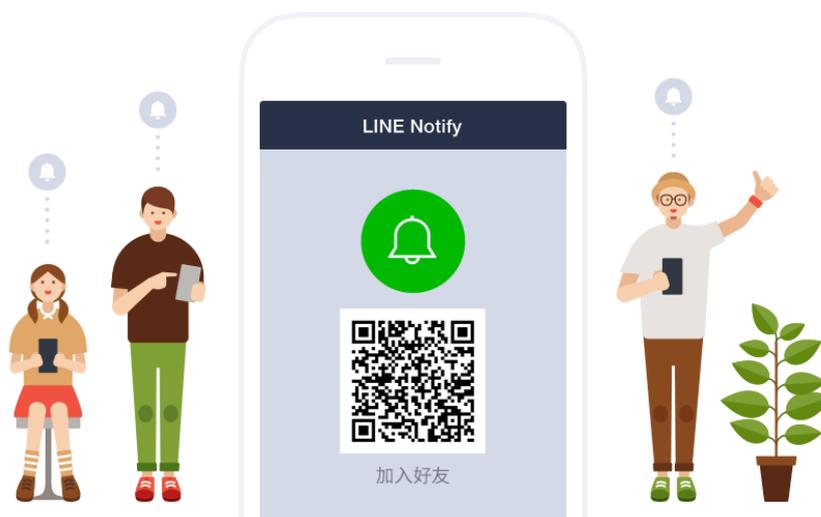
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;

```

3.搭配 LINE notify 去做資料傳遞的功能。

透過LINE接收其他網站服務通知

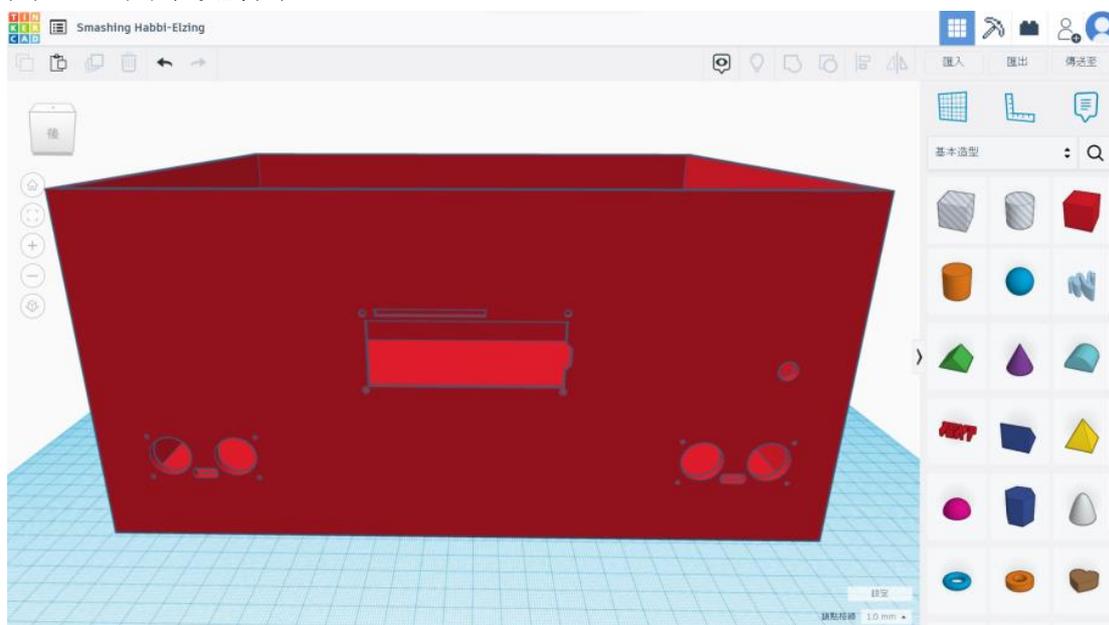
與網站服務連動完成後，LINE所提供的官方帳號「LINE Notify」將會傳送通知。
不僅可與多個服務連動，也可透過LINE群組接收通知。



(五)外殼設計;

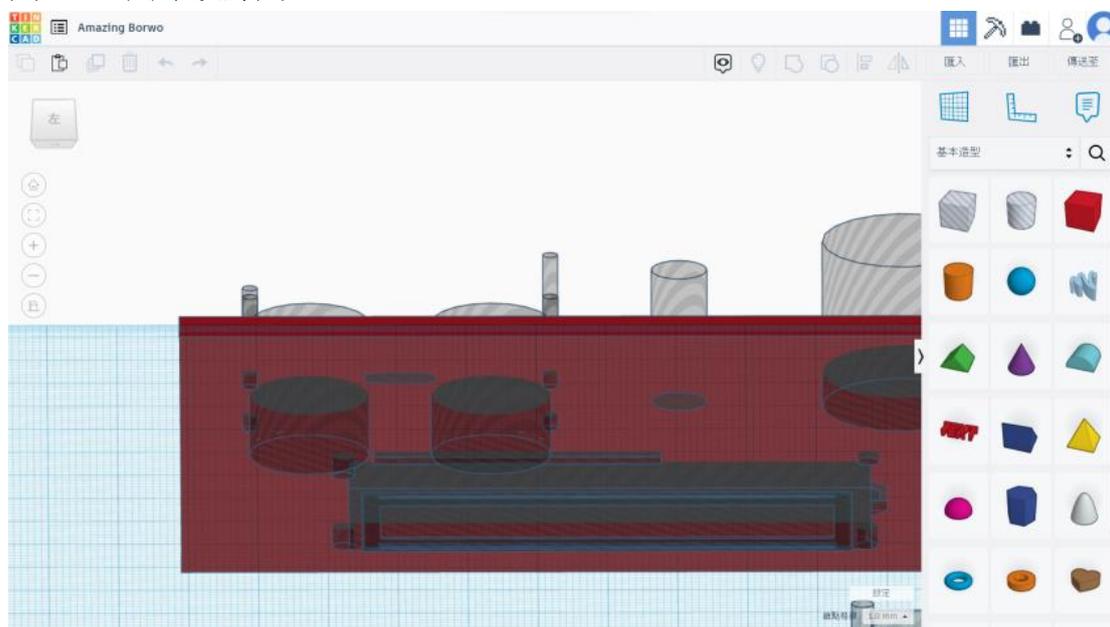
1.根據內部所需的零件大小利用 3D 繪圖軟體設計出外殼的大小。

圖四:3D 列印設計圖-1



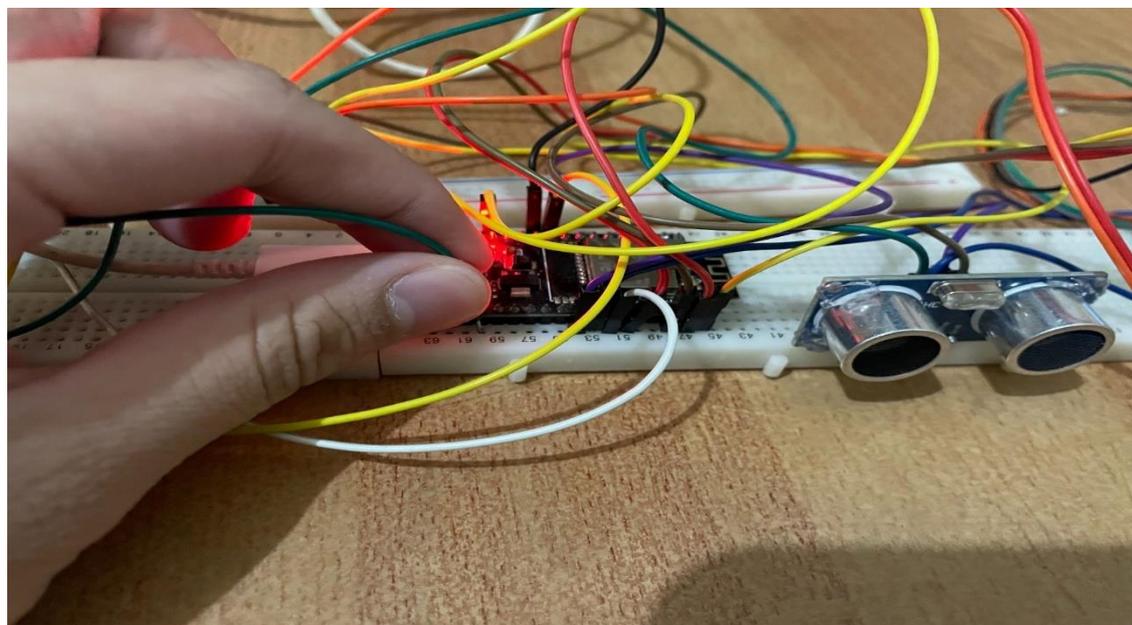
2.依照零件的偵測面去做位置的規劃以及符合零件大小的孔洞。

圖五:3D 列印設計圖-2



(六)組裝實體:

圖六:成品組裝圖



三、研究架構

超速照相

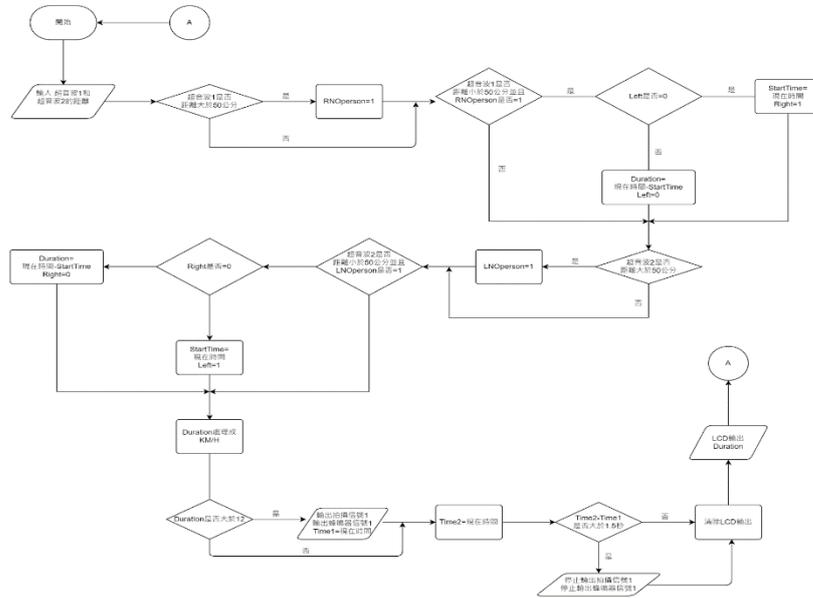
(一)專題架構圖:

圖七:實體架構圖



(二)程式流程圖:

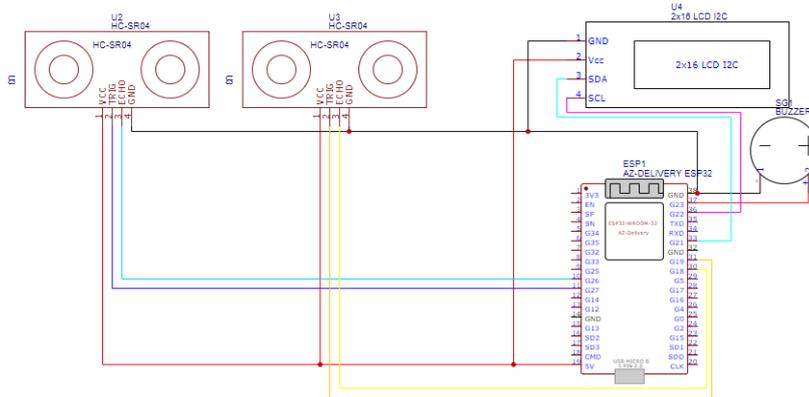
圖八:程式流程圖



(三)電路圖

圖九:電路圖

超速照相



肆、研究分析與結果

一、實驗結果

透過實驗我們得知了下列的結果

1. 單一物體速度高於每秒 300 公分可以成功發出警示聲以及拍下物體並透過 Line notify 回傳資料到手機，而低於的話只會單純顯示每秒幾公分而已。
2. 此次作品在短時內多物體移動會造成測試數值錯誤
3. 如果有兩個物體從兩邊方向經過會造成數值錯誤

二、分析

由實驗結果可以分析出下列事項

1. 單一物體偵測:
由偵測結果得知物體為單一方向。
2. 多數物體偵測
由偵測結果得知多數物體移動速度無法得知物體的實際速度。

伍、研究結論與建議

一、研究結論

此次作品我們是利用 Arduino 來撰寫程式，以及用 ESP32 和其他零件來實現超音波測速，並且可以在超速時發出聲音警示，然後拍下照片傳回手機，以記錄超速的資料。經過了此次的研究我們學習到了如何用 C++ 寫 Arduino 程式以及 ESP32-CAM、LCD、HC-SR04 相關零件如何用程式去控制等等的知識，雖然有些功能我們只能參考網路上的程式去做修改，不過我們大部分的功能還是靠自己去實現的，而在這製作的當中，我們也遭遇了許多的困難，其中最大的困難就是如何將 ESP-32CAM、ESP32、LINE notify 一起搭配來作使用並且製作出超速拍照的效果、不過幸運的是，我們經過不斷的摸索與嘗試還是成功地製作了出來。

二、研究困難

在我們製作研究時，我們碰到了一些問題並且尋找一些解決問題的方法。

(一)超音波有時偵測距離會到 1000 公分

解決方法:判斷條件增加小於 1000 才執行

(二)LCD 顯示的速度數值變成 inf

解決方法:判斷條件增加不等於 0 才執行

(三)ESP32-CAM 沒有連到手機網路

解決方法:IPhone 手機個人熱點開啟最大化相容

陸、參考文獻

一、iT 邦幫忙:一起幫忙解決難題，拯救 IT 人的一天。

<https://ithelp.ithome.com.tw/articles/10260120>

二、【IOT 物聯網應用-ESP32】 米羅科技文創學院。

<https://shop.mirotek.com.tw/iot/esp32-start-9/>

三、【IOT 物聯網應用-ESP32】 超音波測距模組 HC-SR04 | 米羅科技文創學院。

<https://shop.mirotek.com.tw/iot/esp32-start-10/>

四、ESP32(Arduino) - 夜市小霸王- 痞客邦。

<https://youyouyou.pixnet.net/blog/category/3441000>

五、照相後傳 LINE 通知- ESP32-CAM。

https://sites.google.com/a/gapp.hcc.edu.tw/telegram_esp32-cam/line