

臺北市立大安高級工業職業學校

電子科

專題報告

交通指揮棒

Traffic Baton

學生：組長：許睿辰

組員：范皓綸

組員：陳恩祈

組員：陳彥瑾

指導老師：林家德

摘要

如今的社會中工業科技發達，家家戶戶都有各自的汽車與摩托車，但在工業科技普及的情況下，難免有一些弊端的產生，光是汽車就有817萬輛，而摩托車則有1400多萬輛，如此龐大的車流量我們決定改造交通警察的交通指揮棒，讓交通警察能更明確的表達交通警察之指令。

英文：

In today's society, all the families have a cars and motorcycles. But, with the popularization of industrial technology has some disadvantages. Just the cars are 8.17 million vehicles and motorcycles are reach up to 14 million vehicles. In order to improve the huge traffic flow, we decide to transform our traffic baton. Let the police can express the instructions of traffic more clear.

目錄

摘要.....	
目錄.....	
圖目錄.....	
第一章 前言.....	
1-1 背景及目的.....	
1-2 預期成果.....	
第二章 理論探討.....	
2-1 ESP32.....	
2-2 Ws128B.....	
2-2 UTC7805.....	
2-2 水銀開關.....	
第三章 專題準備.....	
3-1 系統流程.....	
3-2 甘特圖.....	
第四章 專題成果.....	
4-1 電路設計.....	
4-2 外殼設計.....	
4-3 程式設計.....	
4-4 成果展示.....	
4-5 問題與解決.....	
第五章 結論與建議.....	
5-1 結論.....	
5-2 建議.....	
第六章 心得.....	
6-1 心得.....	
第七章 參考文獻.....	

圖目錄

圖1-1 揮舞之效果示意圖.....	
圖2-1 ESP32樣品圖	
圖2-2 WS128B 樣品圖	
圖2-3 TO-220示意圖.....	
圖2-4 TO-220之腳位	
圖2-5 水銀開關.....	
圖3-1 系統流程圖	
圖3-2 甘特圖	
圖4-1 繪圖軟體之 ICON	
圖4-2 電路之設計圖.....	
圖4-3 正反列印圖.....	
圖4-4 曝光設備.....	
圖4-5 蝕刻機.....	
圖4-6 電路板之成品圖	
圖4-7 熔融沉積成型示意圖.....	
圖4-8 cyberpi 手把	
圖4-9 Cyberpi 手把切片圖	
圖4-10~14 ESP32外殼固定設計-初版.....	
圖4-15改良品1 頂部	
圖4-16改良品2 頂部	
圖4-17 改良品1組裝成品圖	
圖4-18~22 ESP32外殼固定設計-改良品_底部：	
圖4-23 ESP32外殼固定設計--終版：	
圖4-24 外殼仰視圖	
圖4-25 外殼俯視圖	
圖4-26 Arduino 設定變數.....	
圖4-27 Arduino 設定接角.....	
圖4-28 Arduino LED 輸出	
圖4-29 Arduino 掃描程式.....	

圖4-30 Python 設定字型.....

圖4-31 Python 轉換進制

圖4-32 Python 藍芽傳輸.....

第一章 前言

1-1 背景及目的

交通指揮棒是對於指揮交通不可或缺的東西，如果沒有他，交通警察可能就會變得不是那麼的顯眼，再來是在高速行駛時，駕駛人可能來不及看到交通警察的手勢根本不知道前方有狀況，經過我們的思考後，發現在高速行駛時觀看圖像是能最直接了解到資訊的，於是這次的專題我們採用視覺暫留來讓交通警察在揮舞時能顯示出字幕。但它的功能可能不只適用於交通指揮，像是一些演唱會現場，當觀眾在揮舞時，也能顯現出文字。

1-2 預期成果

在完成所有的步驟之後，我們預期它可以擁有上述 1-1 之所有的功能，其中包括使用藍芽模組來實現能用手機控制想顯現之文字。

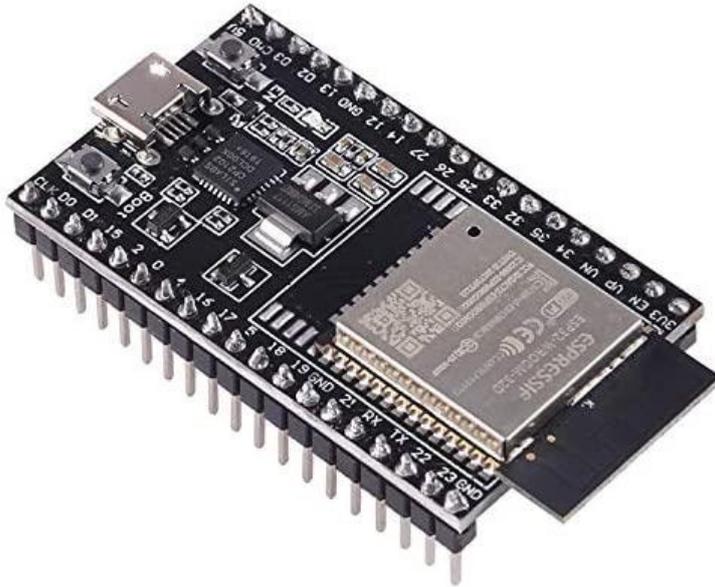
▼ 圖1-1 揮舞之效果示意圖



第二章 理論討論

2-1 ESP32

▼ 圖2-1 ESP32



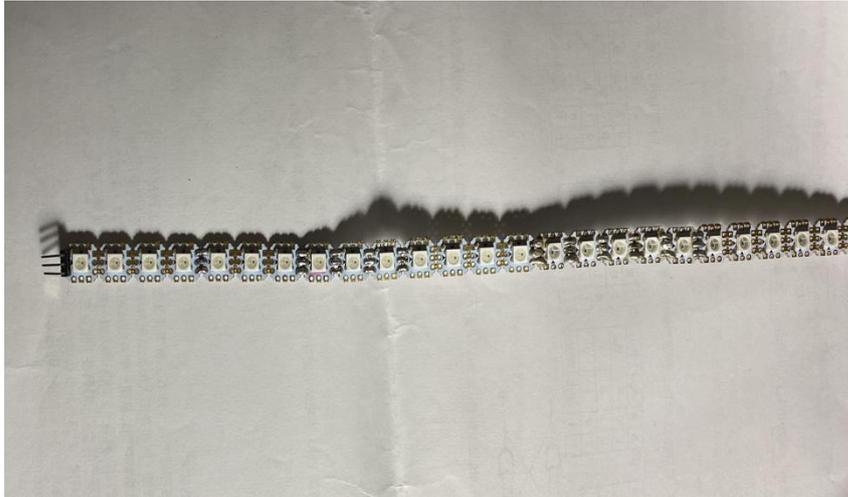
ESP32 是一系列低成本、低功耗的單晶片微控制器，整合了 Wi-Fi 和雙模藍牙。

他的 CPU 是 Xtensa 雙核心（或者單核心）32 位元 LX6 微處理器，工作時脈 160/240 MHz，記憶體是 448 KB ROM 及 520 KB SRAM 分為兩種

可使用 SPI 儲存器 對映到外部記憶體空間，部分儲存器可做為外部儲存器的快取。

▼ 圖2-2 WS128B

2-2 WS128B



有著 3 個輸入端，分別為正、負和控制線，有著單線式通訊、方便使用程式控制及

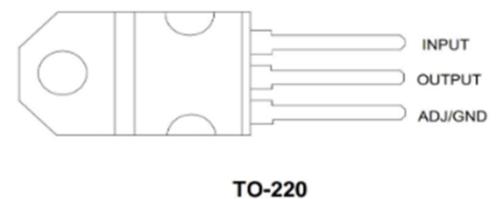
共陽極的 LED。上面也有一顆表面粘接式的電阻，用來保護 LED 不會因電流太大而燒毀。

2-3 UTC7805

▼ 圖2-3 T0-220示意圖



▼ 圖2-4 T0-220之腳位



為 TO-220 封裝，用來消除雜訊及穩壓，當今天輸入 5V 以上的電壓進去時，可以讓他穩壓成 3.3V，上面也有一個散熱版，避免當今天溫度過高時，會導致我們穩壓效果會下降。

2-4 水銀開關

▼ 圖2-5 水銀開關



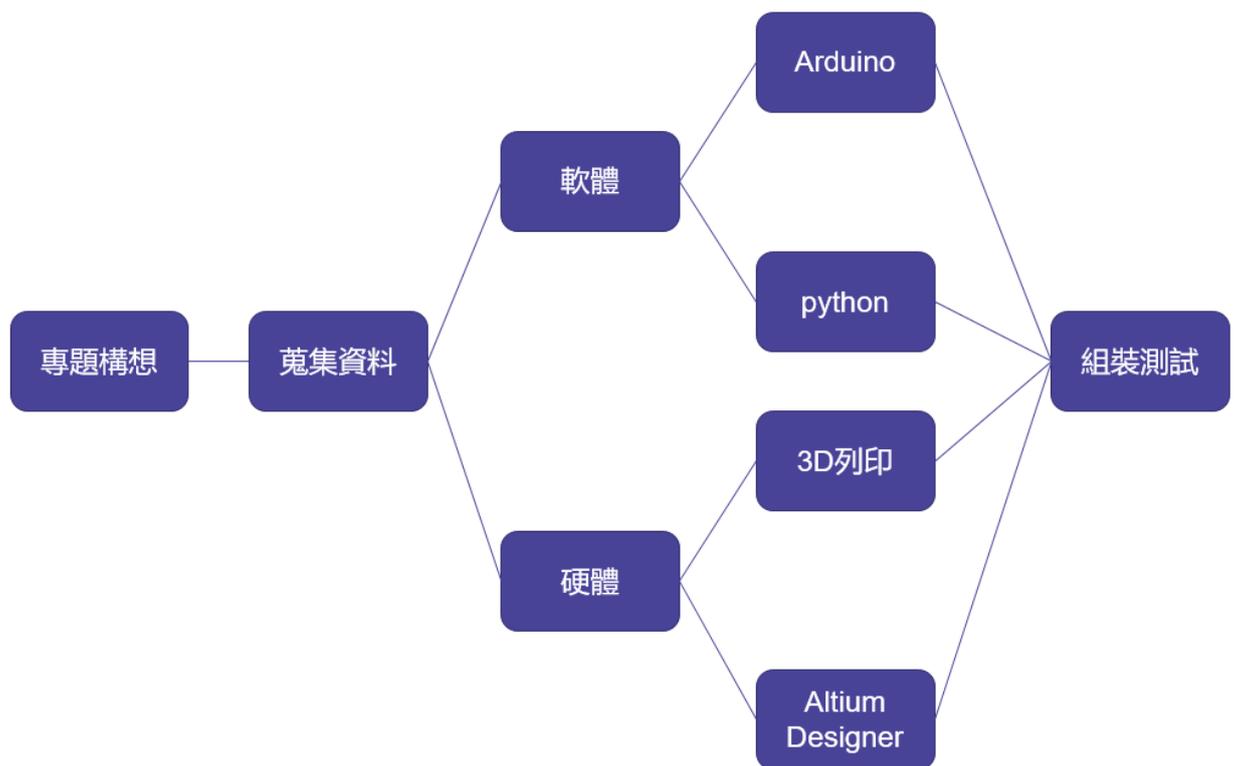
利用重力 使有接觸到電極才導通，並且體積小。

第三章 專題準備

3-1 系統流程

由專題構想選擇專題功能之大致方向，再來蒐集此次作品所需之參考資料，再來分為硬體與軟體兩部分來同時進行，最後組裝測試得出成品。

▼ 圖3-1 系統流程圖



3-2 甘特圖

▼ 圖3-2 甘特圖

工作項目	週次																		負責成員
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
討論專題分配	■	■		■	■								■	■			■		全體成員
選材購材		■	■										■	■	■				1
程式設計						■	■	■	■				■	■	■	■	■		1 19 24
建模設計	■	■	■	■								■	■	■			■	■	1
燈條焊接			■	■													■	■	19 24
底座設計	■	■															■	■	1 15
3D列印			■	■	■	■							■	■			■	■	1
文書			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	15 19
PCB 電路圖設計															■	■	■		19
電路板焊接														■	■	■	■		15 19
洗版																	■		15 19
各項目間協助	■	■	■				■	■					■	■			■	■	1 24
預定進度	5	10	15	20	25	30	40	45	50	55	60	65	70	75	80	90	95	100	累積百分比%

第四章 專題成果

4-1 電路設計

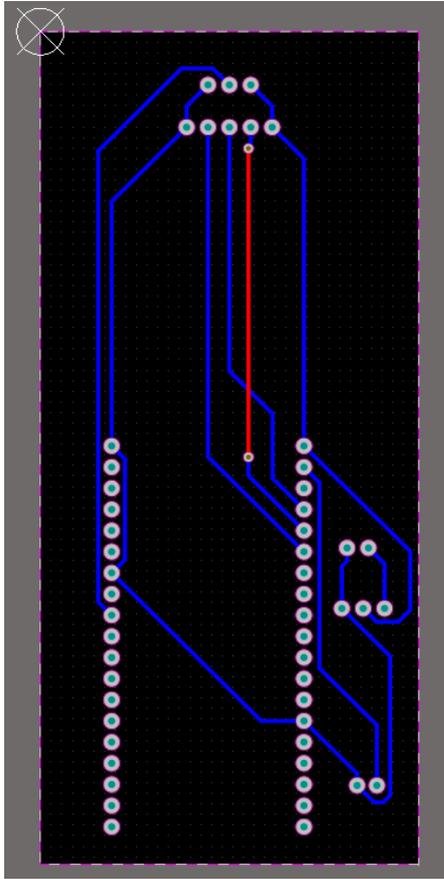
Altium designer

▼ 圖4-1 繪圖軟體之 ICON



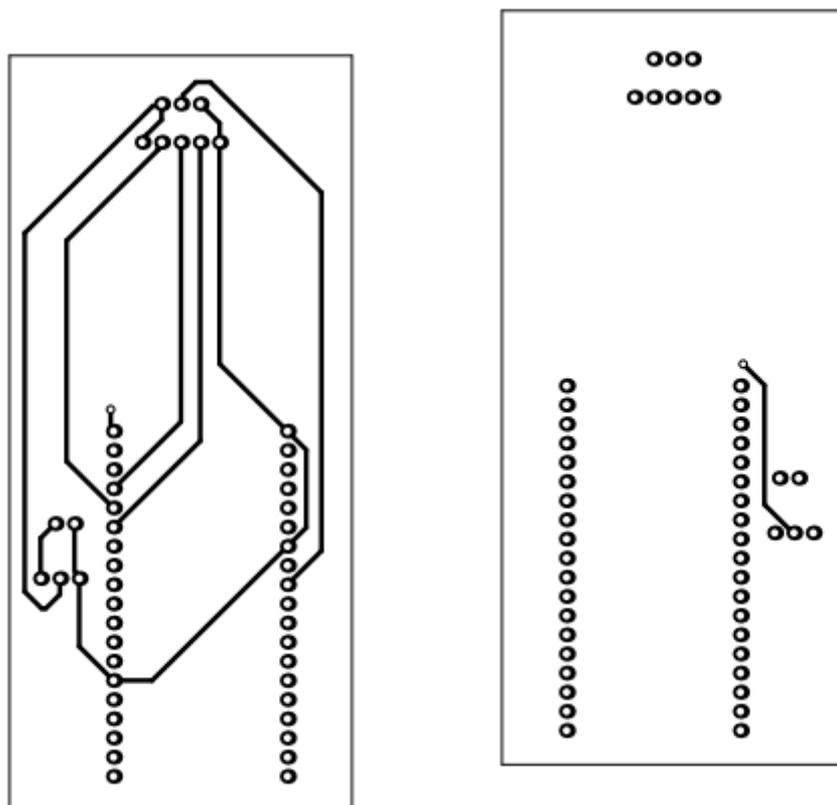
是一款電子設計自動化軟體，由在澳大利亞的軟體公司 Altium Ltd. 開發，在高速電路板設計領域，它可進行差分對布線，還具有板的交互式 3D 編輯。

▼ 圖4-2 電路之設計圖



先利用 Altium designer 設計出我們的板子樣式，並且把圓孔設為 65mm，接著利用自動佈線把所有的腳位連在一起，也盡量都佈線在背面的地方。

▼ 圖4-3 正反列印圖



再來是把他的圖檔列印在 A4 紙上，並且在一開始的時候設定為黑白和設定鑽孔的孔位。

▼ 圖4-4 曝光設備



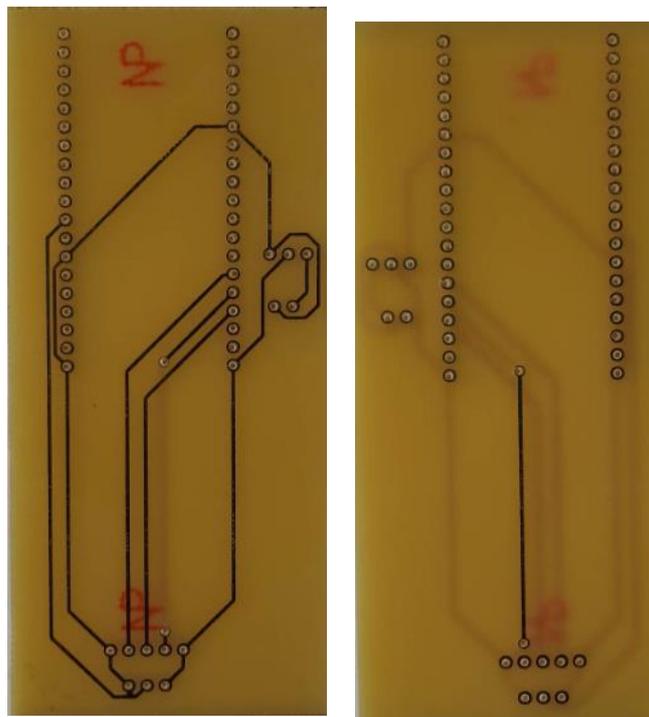
把板子放進紙內並且放進曝光機曝光 300 秒，讓線路印在 PCB 板上。

▼ 圖4-5 蝕刻機



最後把曝光完的板子放進顯影溶液裡面，把多餘的銅洗刷掉，接著再放進蝕刻機裡面。

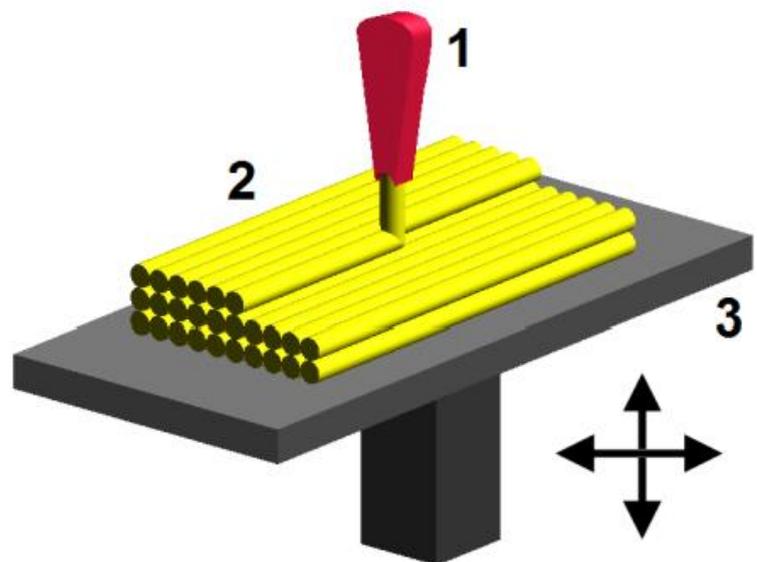
成品圖： ▼ 圖4-6 電路板之成品圖



4-2 外殼設計

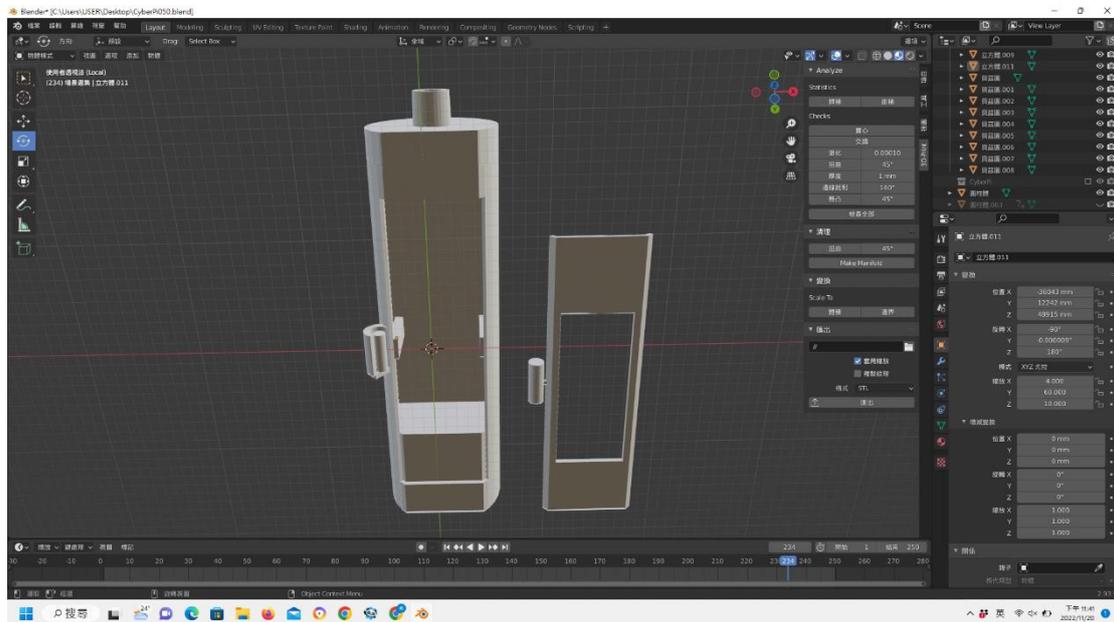
3D 列印

▼ 圖4-7 熔融沉積成型示意圖



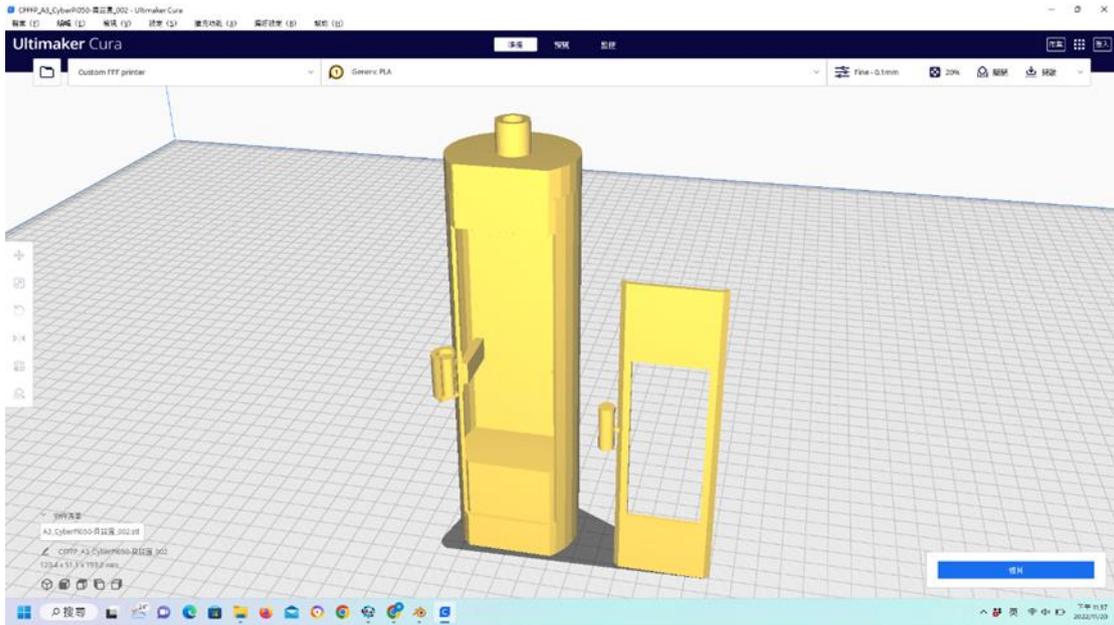
熔融沉積成型(Fused Deposition Modeling) 熔融沉積成型技術，是目前台灣市面上 3D 列印機主流。其機台與消耗性線材因為價格低廉而被廣泛使用。其技術特徵原理是將各種熱熔性的絲狀線料加熱熔解。當列印材料被加熱到半液體狀態時，列印機工作的平台面 X 軸 Y 軸 Z 軸方向前後左右上上下下移動，將列印材料擠壓出，於工作層逐漸凝固成型。列印完成的 3D 列印模型，需將支撐材拆除，以及使用砂紙將表面研磨。

▼ 圖4-8 cyberpi 手把



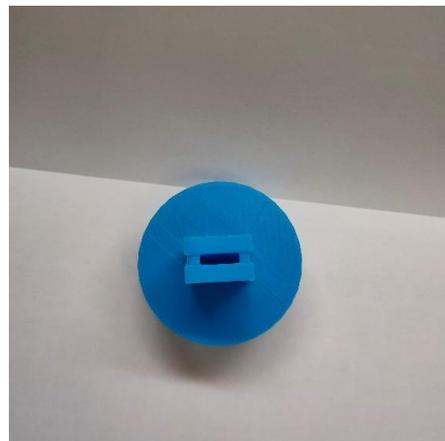
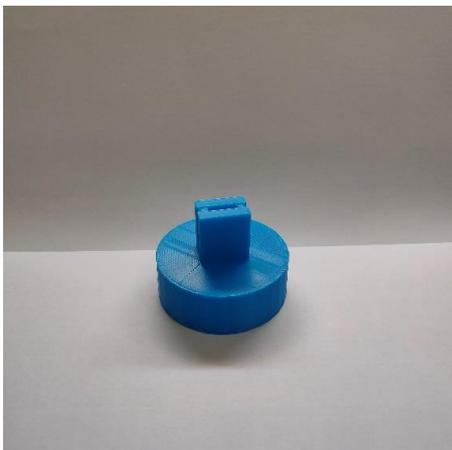
分別設計手把和手把門的部分，並利用小卡榫進行手把與手把門之間的結合與開關。手把門上另外做鏤空設計，讓 Cyberpi 在手把門關上時仍可進行操作。底部做了加高設計，讓 Cyberpi 能被架高，以至於能有足夠的高度可以進行準確偵測

▼ 圖4-9 Cyberpi 手把切片圖



Cyberpi 手把門開關模擬影片：<https://reurl.cc/MXIGAv>

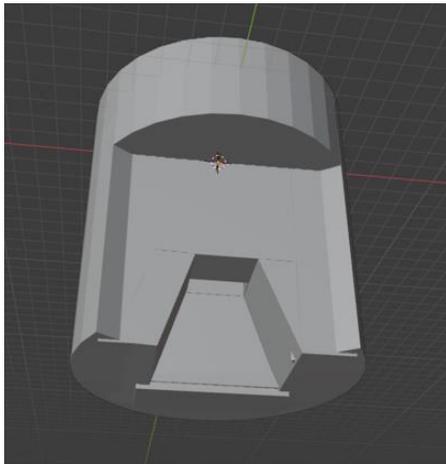
▼ 圖4-10-14 ESP32外殼固定設計-初版



簡單的固定設計，底部鏤空可固定 ESP32，設計小孔使 ESP32 與電腦間的連接線可穿過，頂部卡榫可固定燈條

ESP32 外殼固定設計-改良品_頂部 1 & 2 (對比):

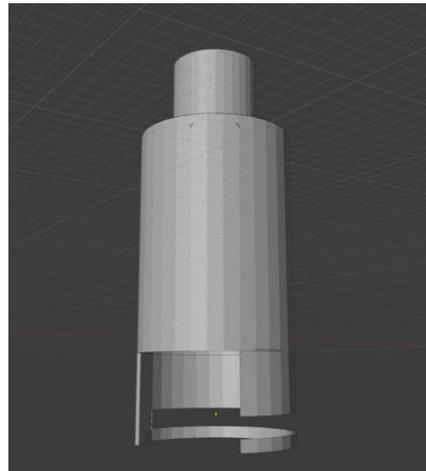
▼ 圖4-15改良品1 頂部



頂部新增坡道式設計，放入燈條的方式由插拔改為放入，使燈條放置的過程能更加順遂、不易造成燈條損傷

▼ 圖4-17 改良品1組裝成品圖

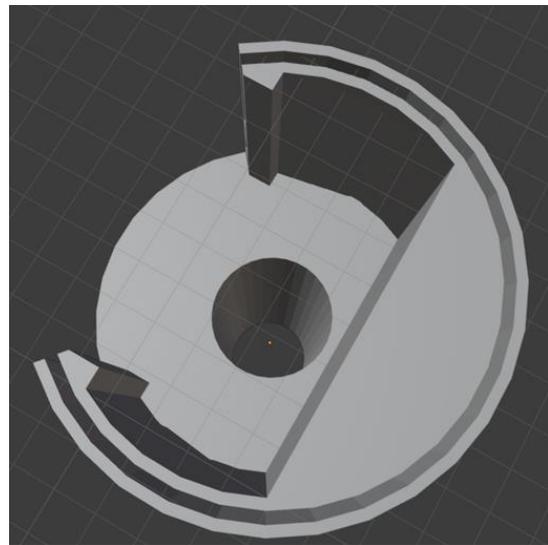
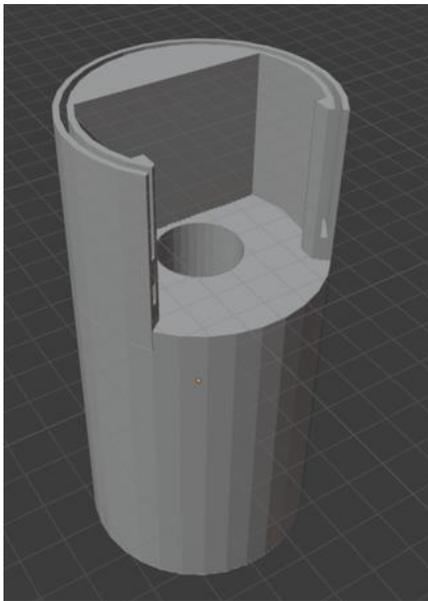
▼ 圖4-16改良品2 頂部

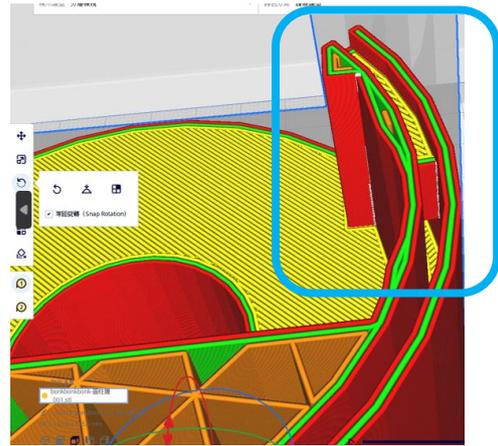
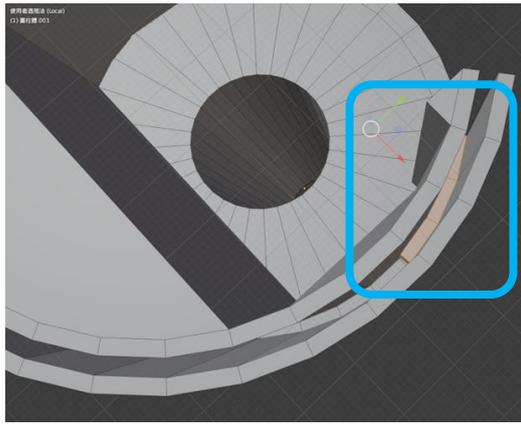


頂部改為包覆式設計，於頂部另外建構卡入透明燈管之卡榫使燈條牢牢固定於燈管中，大幅提升燈條維護性，同時固定 Cyberpi，並以旋轉方式與底部進行連接



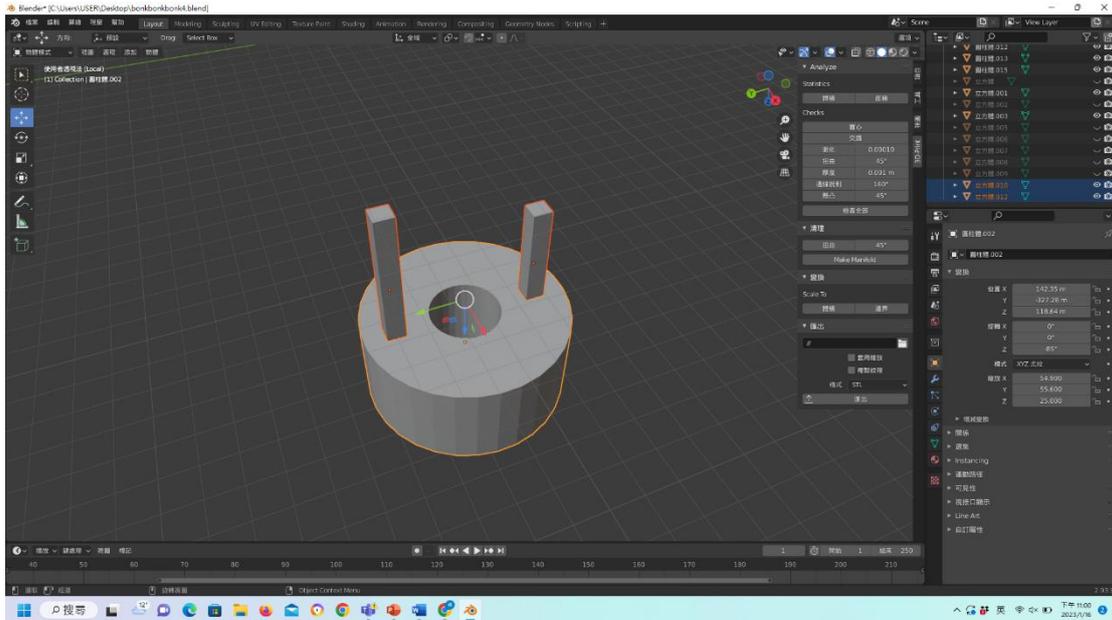
▼ 圖4-18~22 ESP32外殼固定設計-改良品_底部：





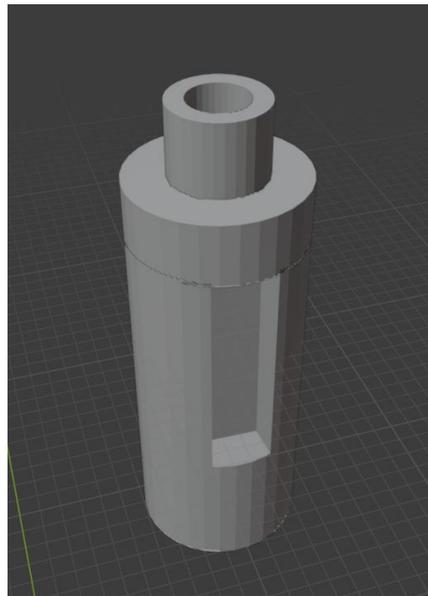
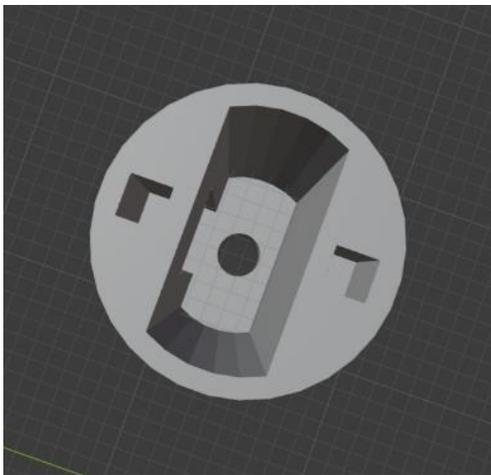
改良型外殼連接模擬影片：<https://reurl.cc/jRn0my>

▼ 圖4-23 ESP32外殼固定設計--終版：



▼ 圖4-25 外殼俯視圖

▼ 圖4-24 外殼仰視圖



將固定外殼改為一體式，減少卡榫的設計以節省列印時間，中間新增鏤空，作為放置鋰電池的位置，上方中空圓柱體用以卡住透明管

終版外殼連接模擬影片：<https://reurl.cc/deMvgD>

4-3 程式設計

首先，程式設計由兩個部分構成，分別為 Arduino 和 Python。

首先是 Arduino 的部分，一開始先宣告函式庫及指定所需用的變數：

▼ 圖4-26 Arduino 設定變數

```
Arduino:
#include <Adafruit_NeoPixel.h>
#include "BluetoothSerial.h"

BluetoothSerial SerialBT;
int R,G,B;
float X,Y,theta,oldX,oldtheta;
Adafruit_NeoPixel pixels( 32, 18 )
int t,line=0;
float p,ct,oldct,T;
int count=0;
#define pi 3.14
#define minDir 0
float maxDir,timer,draw_mode;
float deltaDir;
#define frame 32
unsigned long temp;
int i,nowline;
unsigned long draw_data[frame];
```

再來，在只會執行一次的 setup 函式裡設定要使用的接角：

▼ 圖4-27 Arduino 設定接角

```

--
void setup() {
  Serial.begin(9600);
  SerialBT.begin("ansenchens's_ESP32");
  delay(1000);
  pixels.begin();
  pinMode(34,INPUT);/*X*/
  pinMode(39,INPUT);/*Y*/
}

```

然後將接下來會頻繁使用到的輸出 LED 做成函式：

▼ 圖4-28 Arduino LED 輸出

```

void draw(int sw){↵
temp=draw_data[sw];↵
//Serial.println(temp);↵
pixels.clear();↵
for(int i=0;i < 32 ;i++){↵
//Serial.print(temp%2);↵
if((temp%2)==1){↵
↵
pixels.setPixelColor( i, pixels.Color( 225,0,0));↵
}↵
temp/=2;↵
}↵
//Serial.println();↵
pixels.setBrightness(100);↵
pixels.show();↵
}↵

```

最後，將藍芽輸入、解碼部分及搖動掃描顯示做入 loop(重複執行)的回圈之中：

▼ 圖4-29 Arduino 掃描程式

```

void loop() {
  String input;
  if (SerialBT.available()) {
    input = SerialBT.readString();
    Serial.println(input);
    Serial.println(input.length());
    int output=0,temp_length=input.length();
    for(int j=0;j<temp_length;j+=8){
      output=0;
      for(int i=0;i<8;i++){
        if(input[i+j]>60){
          output=output*16 + input[i+j]-55;
        }
        else{
          output=output*16 + input[i+j]-48;
        }
      }
      draw_data[31-(j/8)]=output;
      Serial.print(draw_data[temp_length/8]);
      Serial.print(" ");
      Serial.println(j/8);
    }
  }
}

```

這樣，Arduino 的部分就完成了

再來是 PyThon 的部分

首先，匯入需要的函式及設定讀取文字之大小，並創建相應的圖片檔

▼ 圖4-30 Python 設定字型

```

import bluetooth↵
↵
from PIL import Image, ImageDraw, ImageFont↵
from time import sleep↵
aaa = [0]*32↵
# 定義圖檔寬、高，中文的字型點陣數，中文字型檔文字的偏移量↵
(w,h)=(500,300)↵
FontSize=32↵
offset=(0,-7)↵
bd_addr = "CC:50:E3:9C:3B:B6" # server 端的 addr↵
↵
newImage=Image.new('RGB',(w,h),"Yellow")↵
drawObj=ImageDraw.Draw(newImage)↵
port = 1↵
↵
strText="停"↵
fontInfo=ImageFont.truetype('NotoSansTC-Regular.otf',FontSize)↵
drawObj.text(offset,strText,fill='Blue', font=fontInfo)↵
↵

```

再來，讀取圖片檔並分析是否有文字，並將其轉換為 16 進制：

▼ 圖4-31 Python 轉換進制

```

# 編碼的加權值↵
KEYS = [0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01]↵
# 縱向編碼，每 8 位元為 1byte，MSB↵
count=0↵
for x in range(0,FontSize*len(strText)):↵
    #if x%8==0: print()↵
    for y in range(0,FontSize,32):↵
        raw=0↵
        for yy in range(0,32):↵
            if newImage.getpixel((x,y+yy))!=(255,255,0):↵
                raw+=pow(2, 31-yy)↵
            print("0x{0:08X}".format(raw),end=',')↵
            aaa[count]="{0:08X}".format(raw)↵
            count=count+1↵
        print()↵
↵

```

最後，將其用藍芽來傳輸資料給 Arduino 端就完成了。

▼ 圖4-32 Python 藍芽傳輸

```
for i in range(0,32,1):  
    sock=bluetooth.BluetoothSocket( bluetooth.RFCOMM )  
    sock.connect((bd_addr, port))  
    sock.send(aaa[i])  
    sock.close()  
# 儲存圖檔  
newImage.save("decode.png")
```

第五章 結論與建議

5-1 結論

經過這次的專題製作訓練，我們這組從中學到了很多，像運用 ESP32 和 WS12B 的使用方式、也更熟練地運用軟體製作電路板和外殼，更重要的是我們還學到了團隊合作，俗話說：「三個臭皮匠，勝過一個諸葛亮。」一個人無法完成的事情，大家合作後，一定會有辦法的。

5-2 建議

製作指揮棒棒的時候，我們發現許多的東西是我們需要一直去改進的，像是我們程式中是有寫到藍牙模組的部分，但是我們由於可能努力不夠，現階段無法完成的。

再來是我們目前的交通指揮棒是需要用人力來揮動，且需要一定的揮動速度，我們討論出來的結果是能再設計一個能讓他自動揮舞的裝置，來讓他更加的方便。

第六章 心得

姓名	許睿辰	班級	19	
曾修習專業科目	電子學 基本電學 數位邏輯 微處理機			
參與專題工作項目	PCB 洗板、焊接、文書			
經歷簡介	<ol style="list-style-type: none"> 1. 曾擔任實習工廠材料 2. 考過工業電子丙級 3. 擔任過風紀、餐膳幹事 4. 擔任過大安電研的社長 5. 參加過簡版急救技能訓練 			
心得	<p>這次做專題讓我獲益良多，從一開始的懂一些到現在了解更多更深入，比起理論，實作讓我學習到更多知識，雖然這之中組員有些衝突，但還是最後有搞出我們的成品，讓我們能夠交出一個功能雖然不多，但效果也不錯的，經過這次的合作，我也了解到原來共同做一個專題也是不容易的，常常需要很多的溝通與協調，對於別人的缺點，也是需要多包涵的。</p>			

姓名	范皓 綸	班 級	電子三 甲	
曾修習 專業科目	程式設計實習 基本電學實習 電子學實習			
參與專題工作項 目	文書、焊接			
經歷簡介	工業電子丙級執照			
心得感想	關於這次的專題我覺得屬實得來不易，從一開始毫無頭緒到最後完成專題是非常困難的，裡面許多東西，大部分都是需要我們自己去摸索以及討論，我們這組的專題，實際上開始製作的時間相較於其他組別來說是晚了非常多，或許我在這組別中能力不是那麼的出色而且還有很多東西都涉及了我知識的盲區，但是組別中需要我的地方我也會盡力去幫忙以及支援，對於這次的專題製作，我很感謝隊友，許多我不會的事情都是靠組員來幫忙解決的，如果沒有他們，我們的專題也不會那麼的順利。			

姓名	陳恩祈	班級	24	
曾修習專業科目	電子學 基本電學 數位邏輯 微處理機			
參與專題工作項目	程式、燈條焊接、個項目協助			
經歷簡介	1. 曾參加國語朗讀比賽得第一名 2. 考過工業電子丙級 3. 當過餐膳幹事 4. 當過工科賽選手			
心得	這次專題製作適逢我要比工科賽的訓練時間，故有許多時間無法專心於專題報告的製作上，也無法與隊友正確的溝通與協調，導致我們的專題報告進度十分緩慢，直到我工科賽結束後回來認真製作專題才發現遇到的困難超乎想像，與隊友的良性溝通是一件非常困難的課題，且程式的設計上也屢屢碰壁，經常因為硬體上的限制而只能打掉全部重做，但是感謝有許多其他組同學的協助令我不至於灰心喪志，能一步一步地完成這個專題，也在這個專題中我學習了更多關於溝通和磨合的技巧，相信這些都是非常有用的能力。			

姓名	陳彥瑾	班級	1	
曾修習專業科目	電子學 基本電學 數位邏輯 微處理機			
參與專題工作項目	選購材料、建模設計、3D 列印、底座設計、各項目協助			
經歷簡介	<ol style="list-style-type: none"> 1. 曾參加作文比賽 2. 考過工業電子丙級 3. 當過輔導幹事 4. 參加字音字形比賽獲得第一名 5. 參加過英文簡報比賽 			
心得	<p>這次專題製作我學到非常多東西，從一開始的 cyberpi 的手把設計，再到 ESP32 的外殼設計，我都花了許多時間在這些上面，想說要怎麼設計的更好更便利，亦或者是在管子上，做一個頂部的蓋子，讓我們的燈條不會飛出去，最後也設計出非常多方案，也持續地在改良，也在這之中更熟悉 3D 列印，在與組員溝通上，也是經過一些波折，有時也有些火花的產生，但幸好最後還沒全部吵起來，經過此次的專題，也了解溝通的重要性，也謝謝各位老師給予我們的協助，幫助我並不了解的地方。</p>			

第七章 參考文獻

blender 的操作方法：<https://youtu.be/0-9c4RnS0gI>

3D 列印的前置設定教學：

<https://youtu.be/dstlHy39zww>

拆支撐材的技巧：<https://youtu.be/gYeQRlLwiVo>

Altium designer 圖

片：<https://www.altium.com/altium-designer>

Arduino IDE 設定：

<http://blog.s2u4o.com/education/self-study/software-settings/arduinoideesp8266esp32/>

