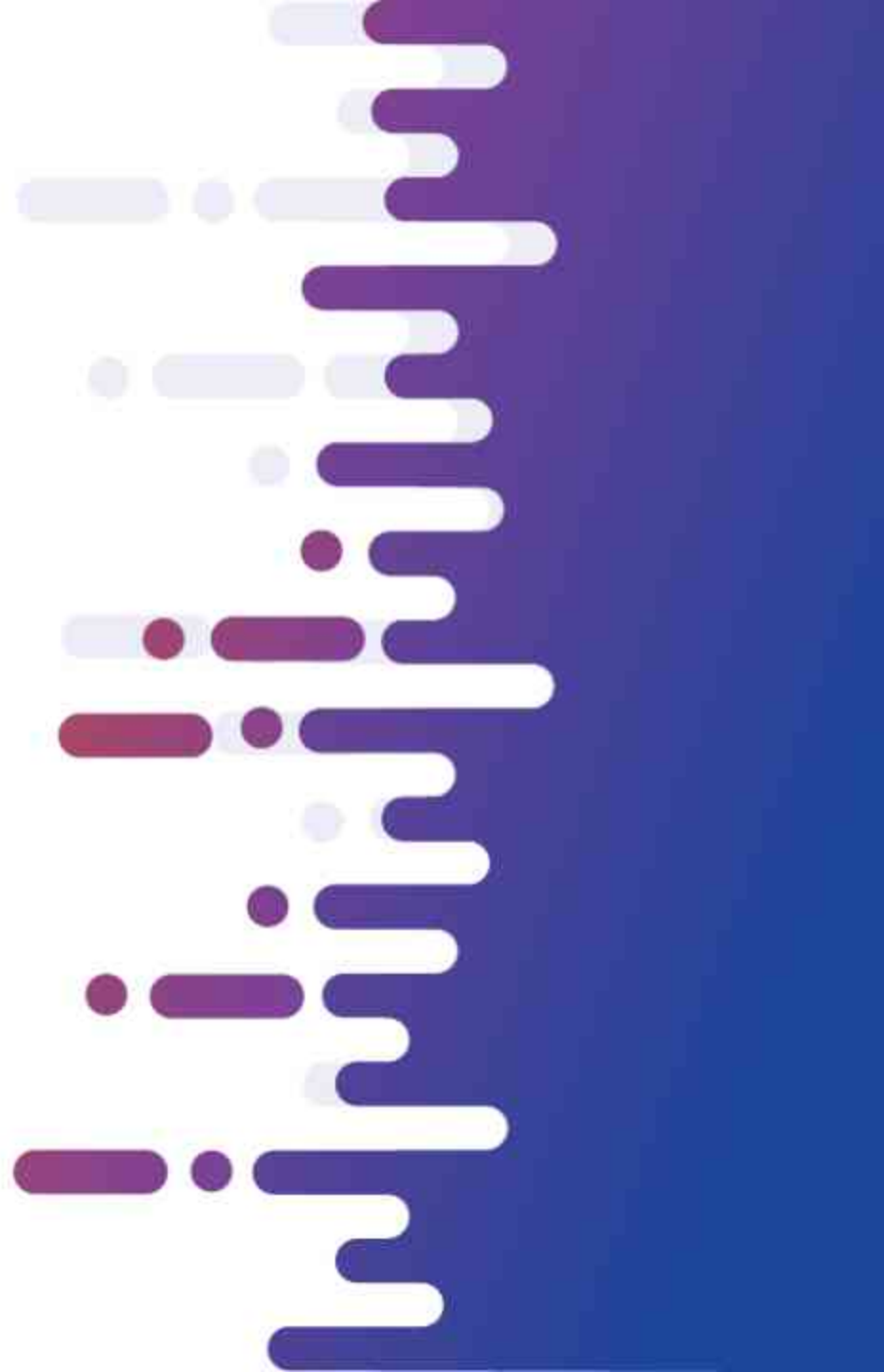


# 交通指揮棒

第八組

組員:陳彥瑾、范皓綸、許睿辰、陳恩祈

指導老師:林家德老師



# 大綱

一 製作動機

二 理論探討

三 成果展示

四 未來展望

五 分工表與流程圖

六 Q&A



製作動機

# 製作動機





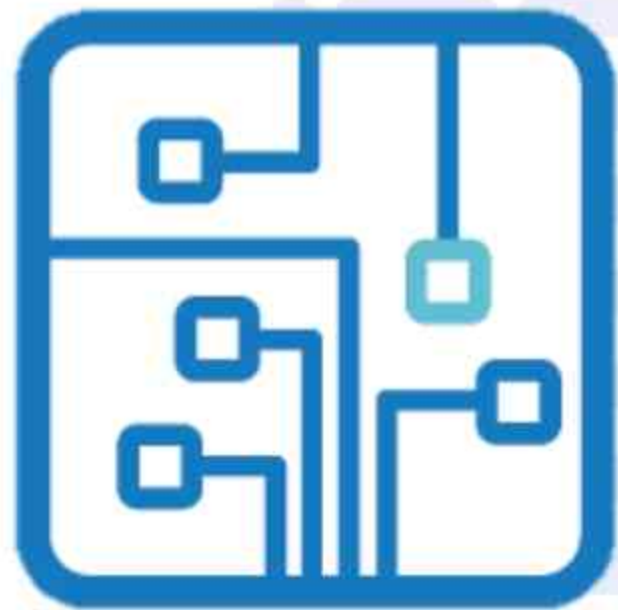
# 理論探討

## 2-1 硬體



分為4部份

- NodeMCU-32S
- 水銀開關
- WS2812
- UTC7805



## 2-1-1 硬體



### NodeMCU-32S

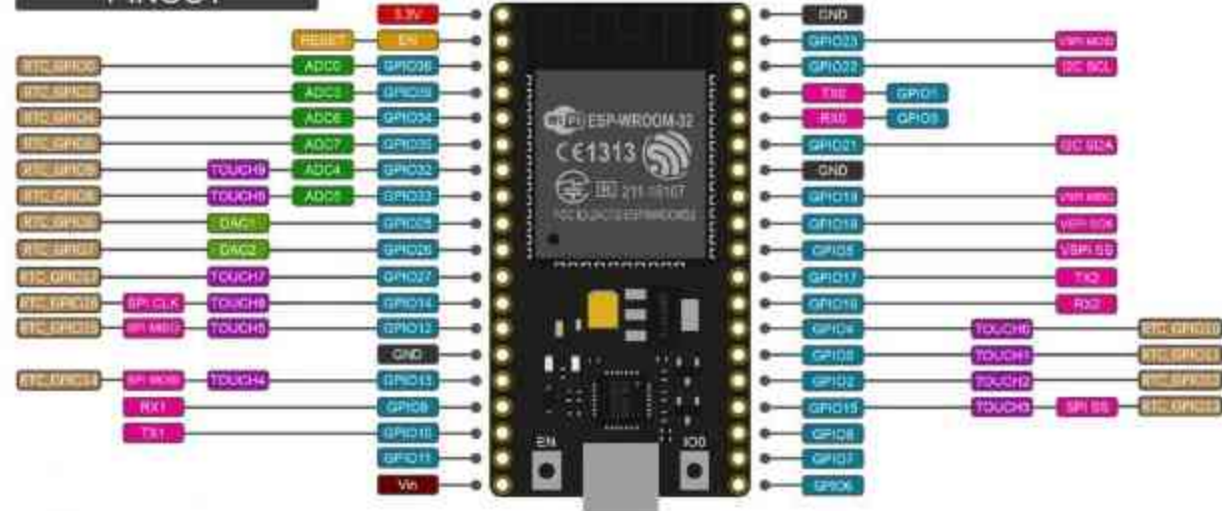
- 使用Arduino IDE函式庫
- 支援I<sup>2</sup>C協定
- 內建WIFI及藍芽模組



### 水銀開關

- 利用重力 使有接觸到電極才導通
- 體積小

NodeMCU-32S  
PINOUT



## 2-1-2 硬體



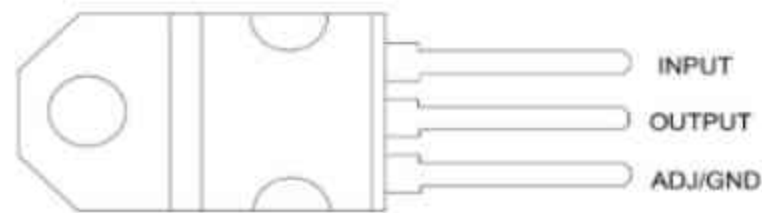
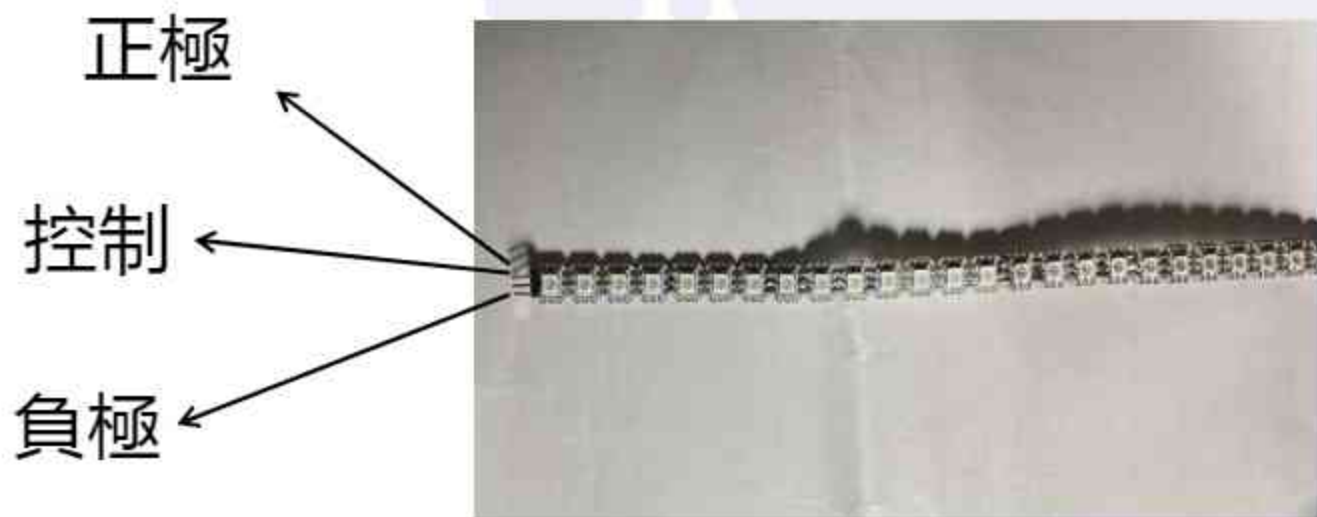
### WS2812

- 單線式通訊
- 方便使用程式控制
- 共陽極的 LED



### UTC7805

- 採用TO-220封裝
- 用來消除雜訊及穩壓



TO-220





## 2-2 程式

 使用Arduino IDE製作

 分為2部份

- 指揮棒主程式
- 文字轉點陣



## 2-2-1 程式

### 指揮棒主程式

- 宣告函式庫
- 設定所需腳位

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];
```

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



## 2-2-2 程式

- LED顯示函式
- 延遲函式

```
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();  
}
```

```
int delayus(int us){  
for(int i=0;i<us;i++;  
}
```



## 2-2-3 程式

- 藍牙資料傳輸
- 搖動顯示

```
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);  
    }  
  }  
}
```

```
X=digitalRead(34);  
if(X==1 && oldX==0){  
  count=(count+1)%4;  
  if(count==0){  
    delay(5);  
    for(int j=0;j<32;j++){  
      draw(j);  
      delay(4);  
    }  
  }  
  oldX=X;  
}  
  
Serial.println(X);
```



## 2-2-4 程式

### 文字轉點陣

- 引入函式
- 製作圖檔

```
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:86" # 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)
+
```



## 2-2-5 程式

- 讀取圖檔

```
# 若是背景色輸出空格，否則輸出*  
for y in range(0,FontSize):  
    for x in range(0,FontSize*len(strText)):  
        if newImage.getpixel((x,y))==(255,255,0):  
            print(end=' ')  
        else:  
            print(end='*')  
    print()  
  
sleep(2)
```




## 2-2-6 程式

- 資料編碼
- 藍牙輸出

```
# 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()↵
↵
```

```
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")↵
```

## 2-3 電路圖

 使用Altium Designer設計

 分為上下兩層

- 使用者介面
- 主控版、感測器與其他元件



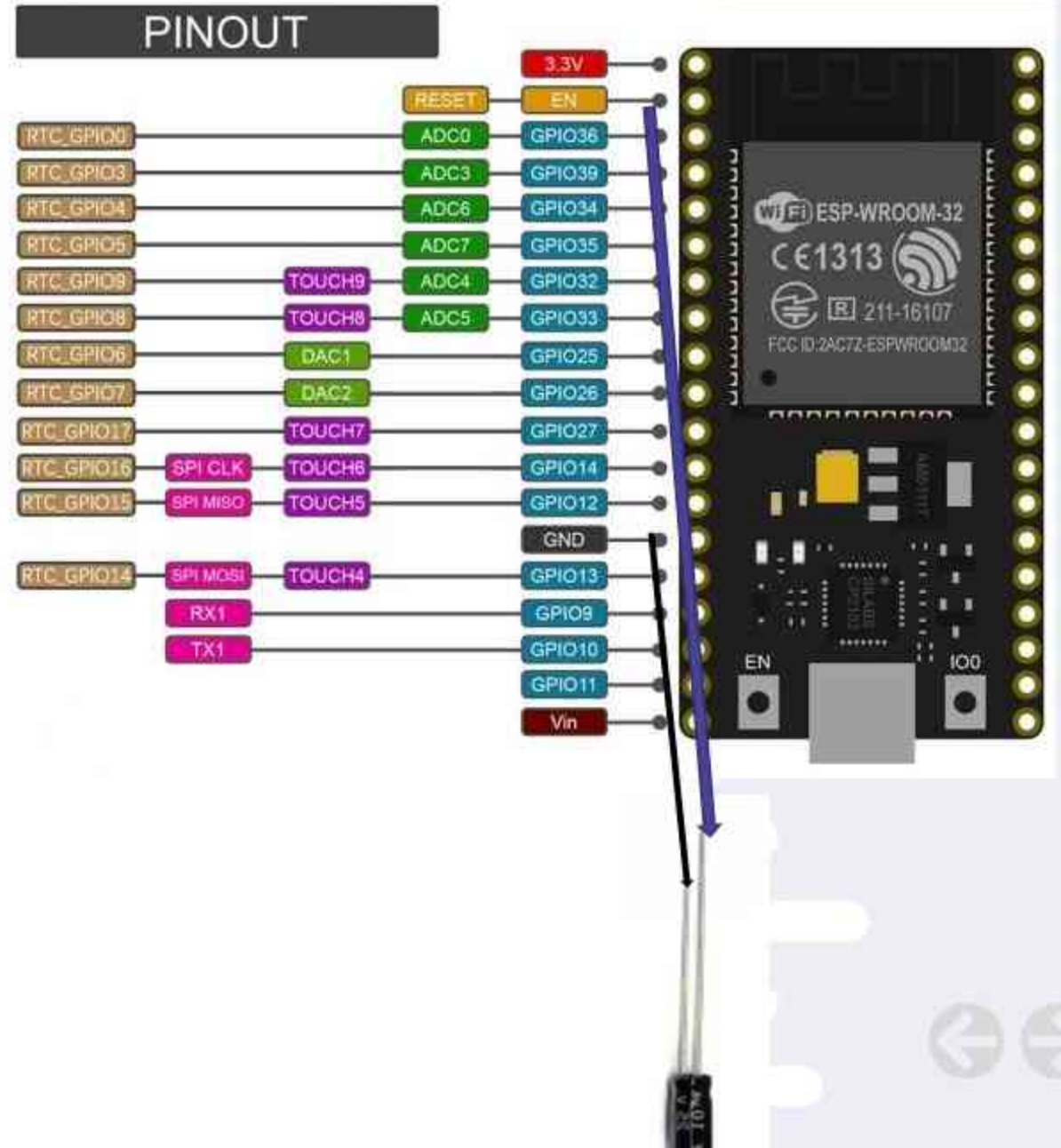
ALTIUM  
DESIGNER





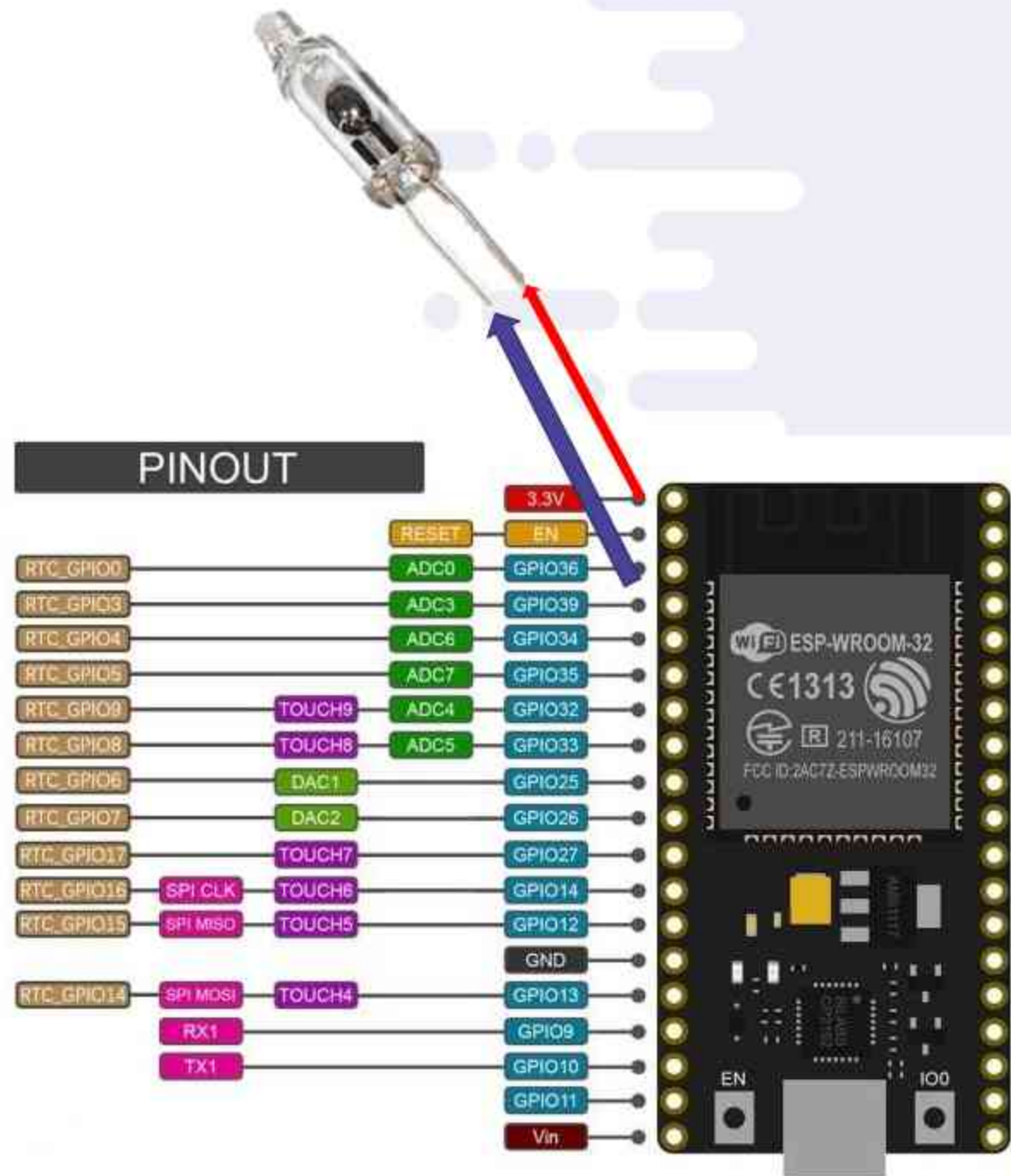
## 2-3-1 電路圖

- 電容所使用之腳位



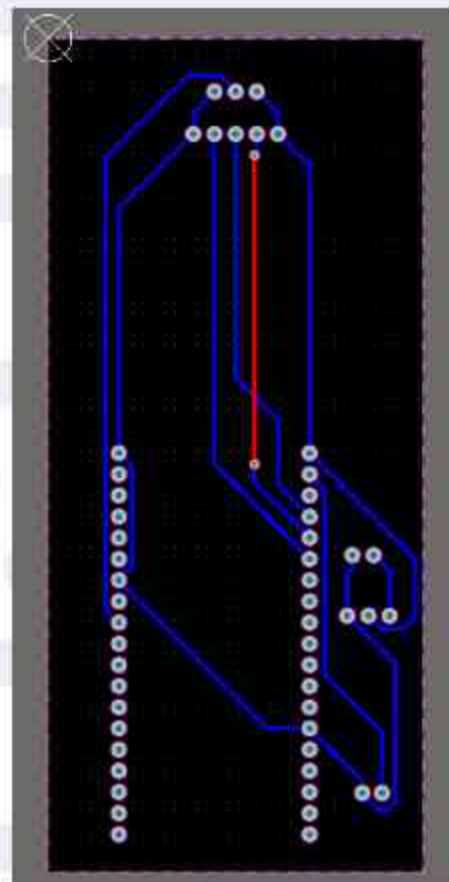
## 2-3-2 電路圖

- 水銀開關所使用之腳位



## 2-3-3 電路圖

- 所設計之電路圖



## 2-4 外殼



使用軟體:

- blender
- cura

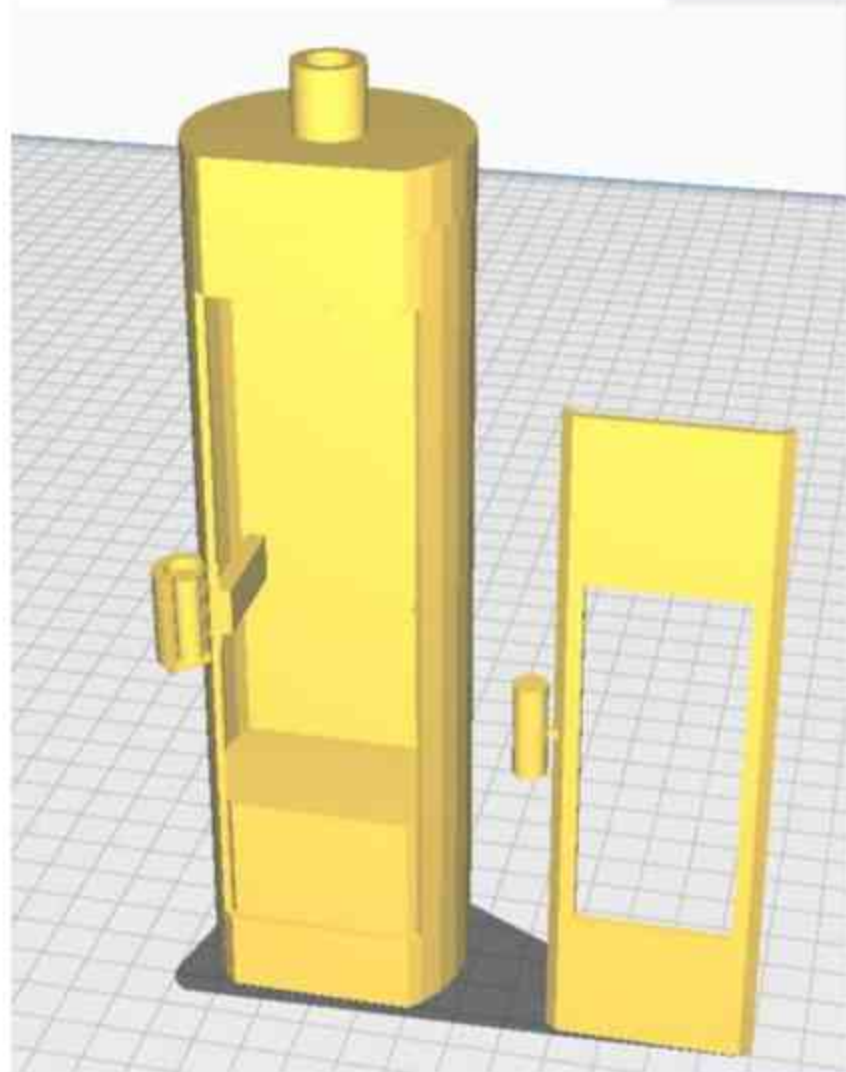
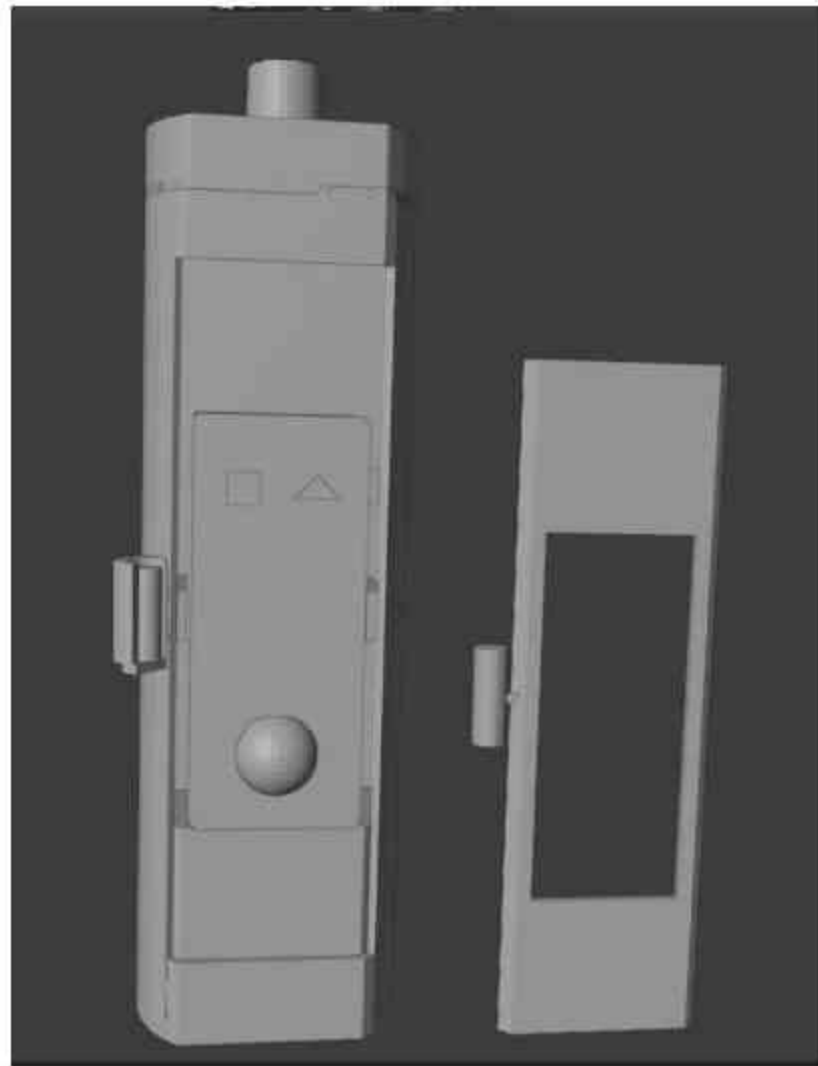


分為3部份

- 原有設計
- 主體設計



## 2-4-1 原有設計



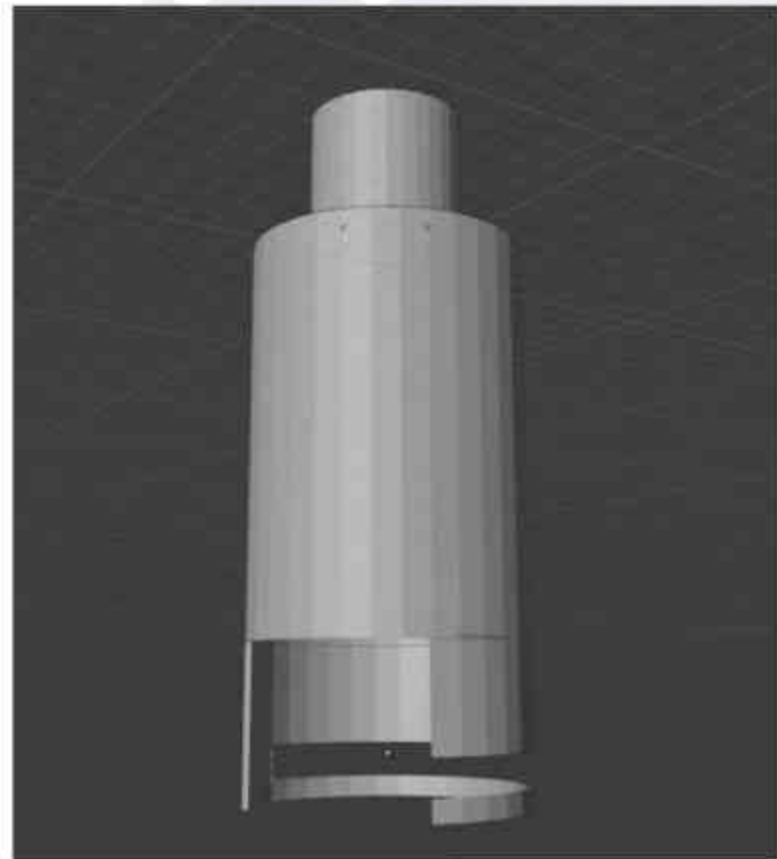
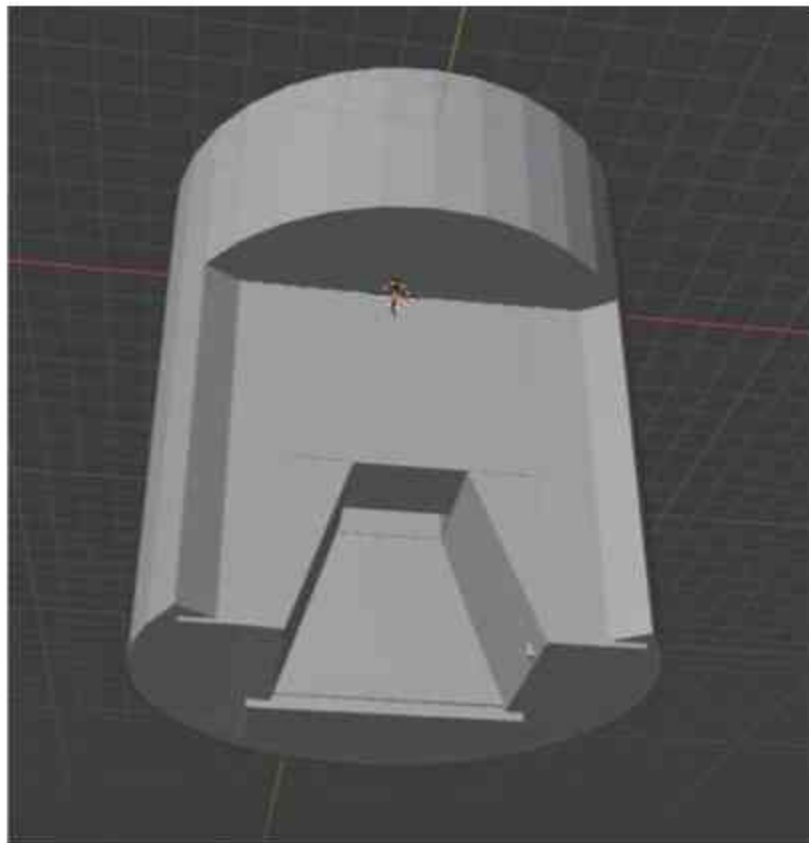
## 2-4-2原有設計



## 2-4-3主體設計-初版

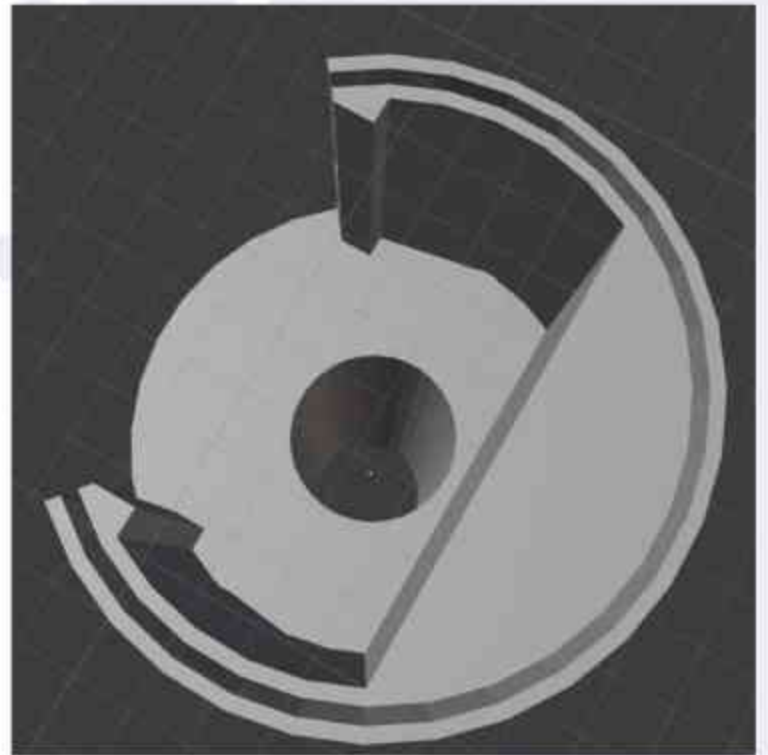
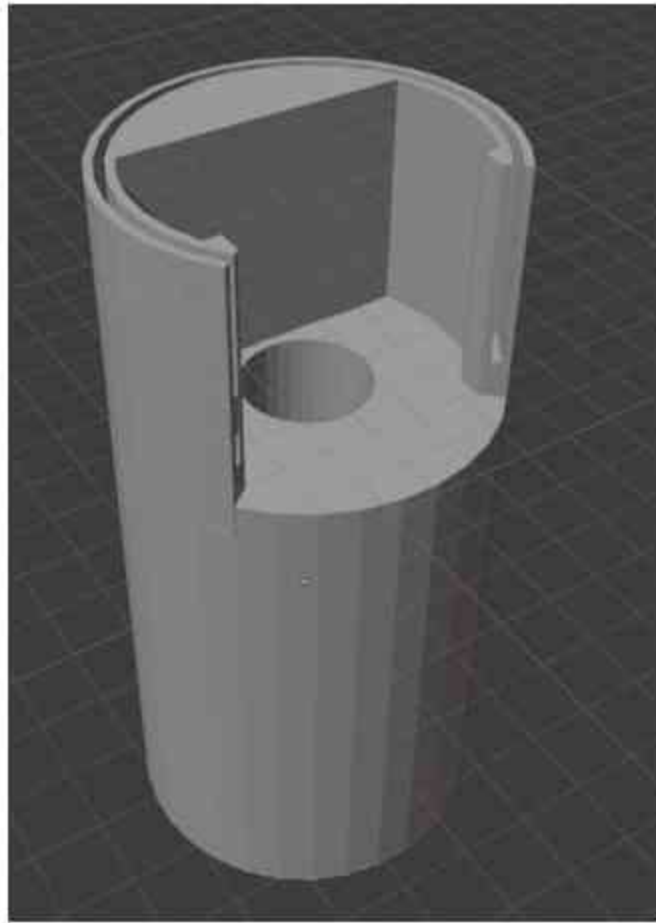


## 2-4-3主體設計-改良



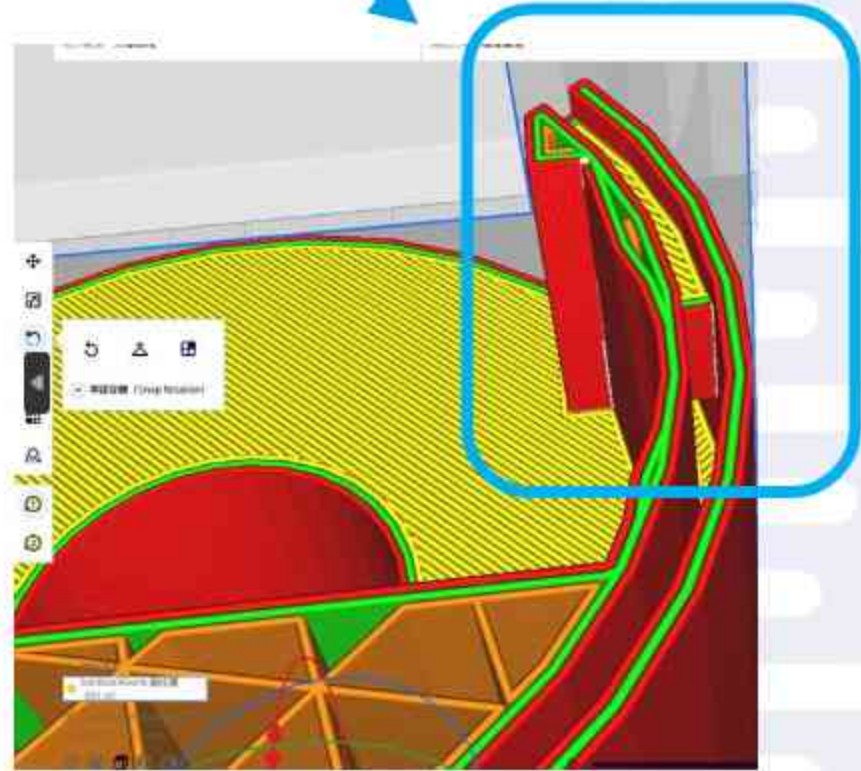
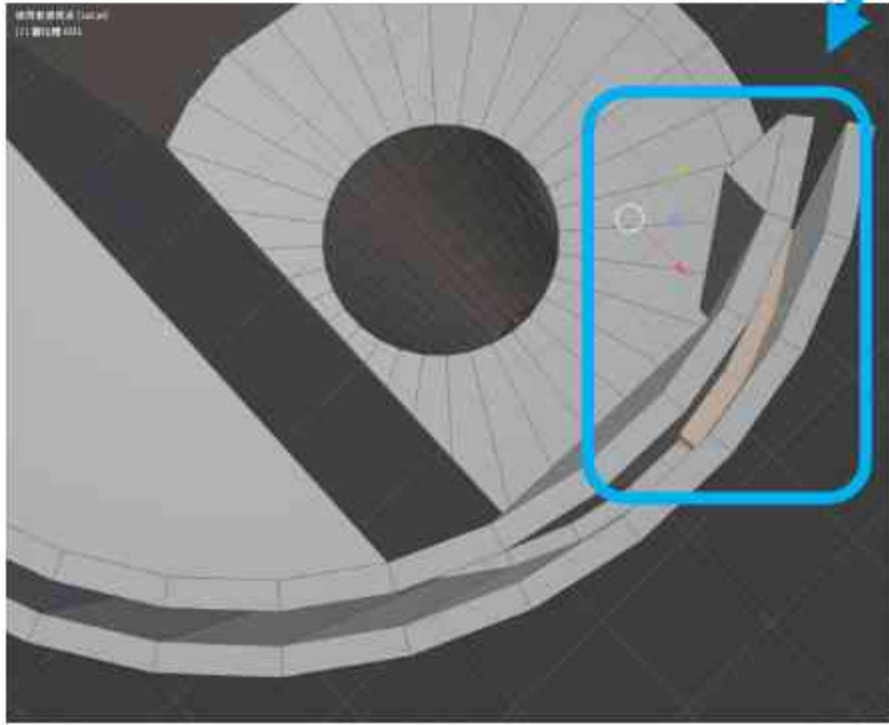


## 2-4-4主體設計-改良

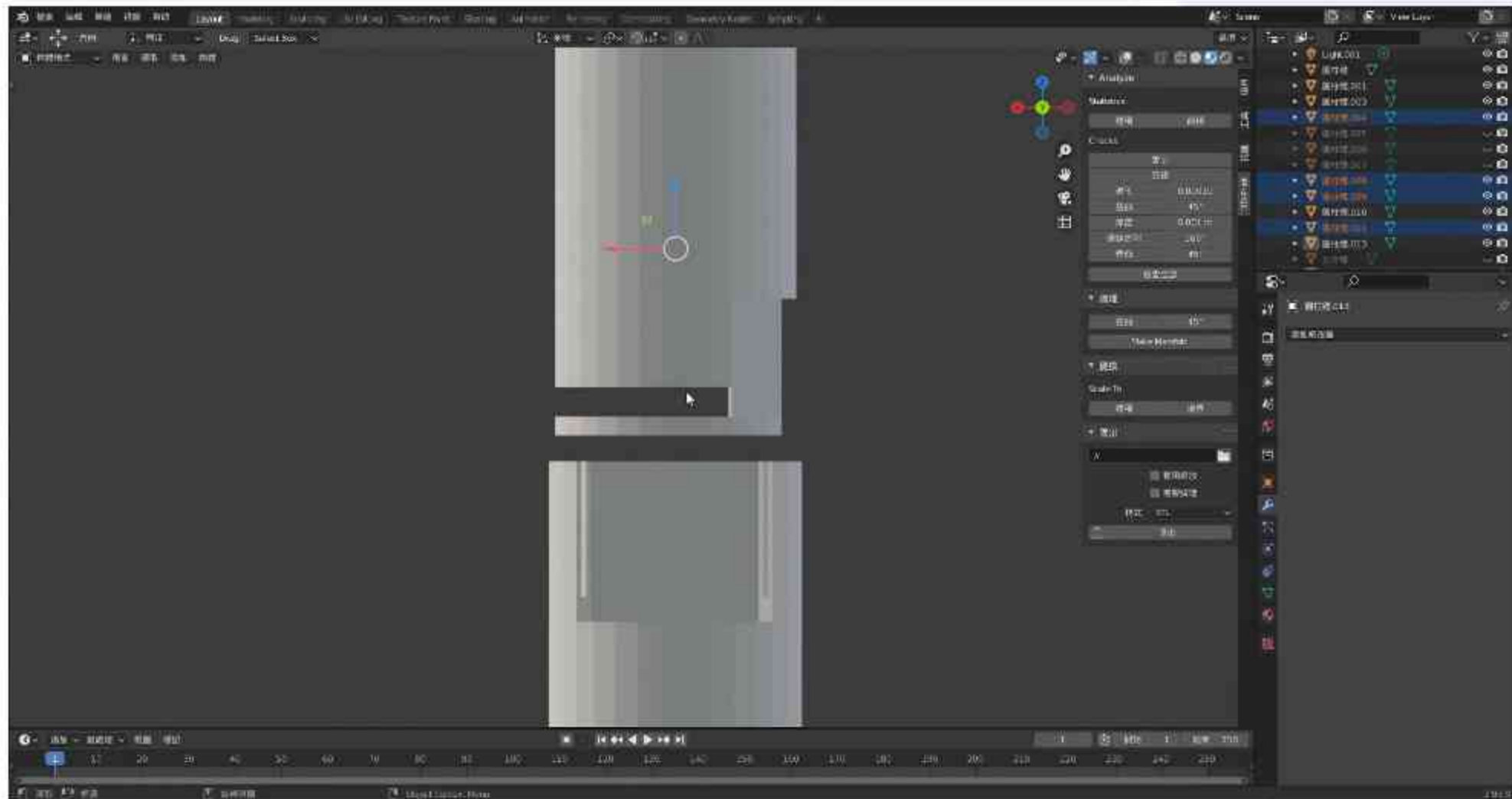


## 2-4-5主體設計-改良

卡榫



## 2-4-6主體設計-改良





成果展示

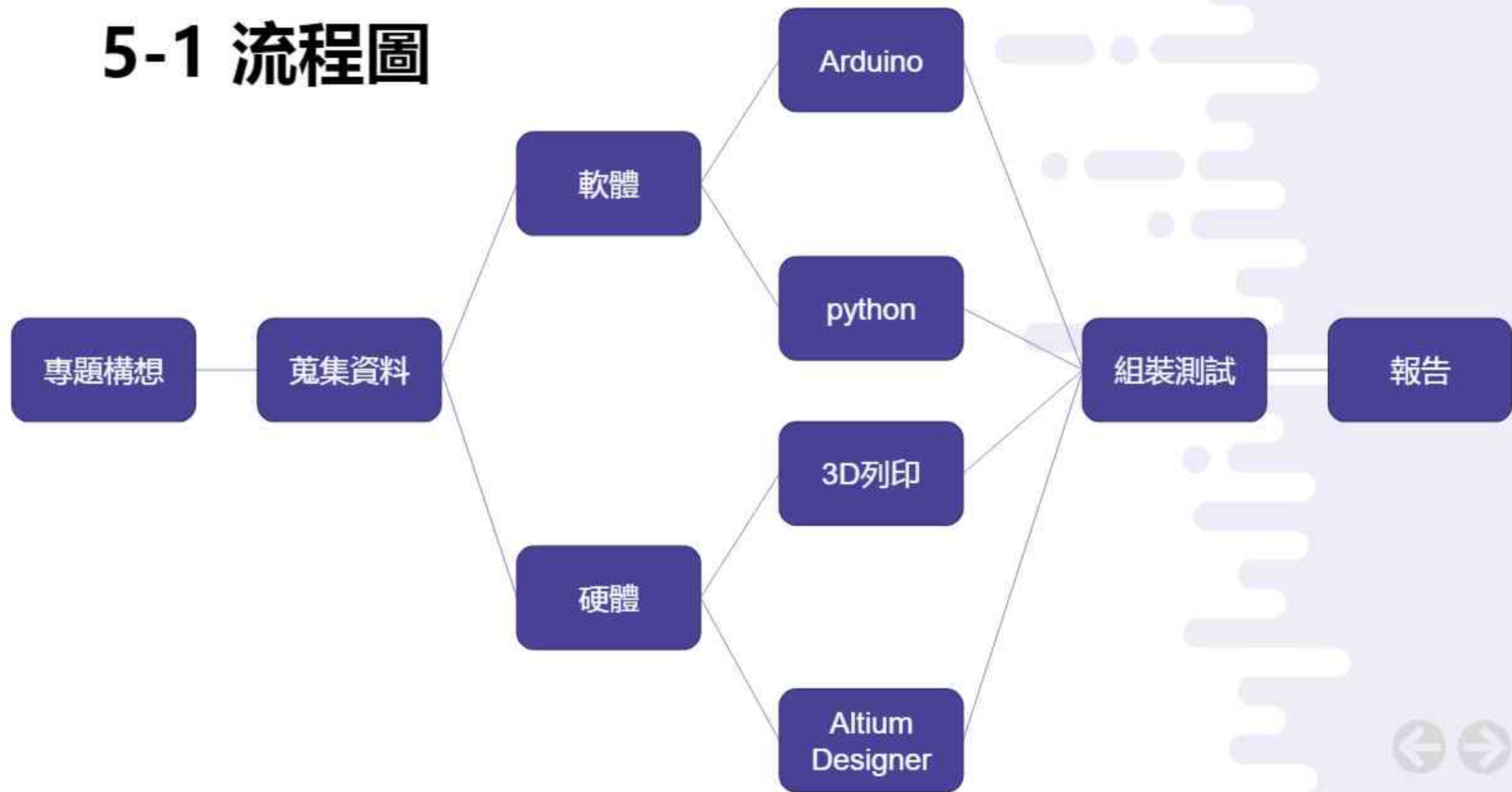
## 4-1 成果展示





# 流程圖與分工表

## 5-1 流程圖



## 5-2 分工表

| 組員名稱    | 個人工作項目與貢獻說明               |
|---------|---------------------------|
| 組長 許睿辰  | 負責文書、焊接板子及燈條和PCB板蝕刻、設計電路圖 |
| 組員1 陳彥瑾 | 負責採購、外殼設計、繪製PCB電路圖        |
| 組員2 范皓綸 | 負責文書、焊接、PCB板蝕刻            |
| 組員3 陳恩祈 | 負責程式設計、燈條焊接、硬體測試與偵錯       |





The image features a background with a gradient from light purple at the top to dark blue at the bottom. A series of white, rounded, wavy shapes resembling a stylized skyline or soundwave runs across the middle. Overlaid on this are several vertical purple bars of varying heights and widths, some topped with small circles. In the upper right, there are faint, semi-transparent horizontal lines and circles, suggesting a grid or data visualization.

未來展望

## 5-1 未來展望

1. 加強燈條固定及外殼結構
2. 精度不足 時常無法準確偵測
3. 設計一個底座能自動揮動
4. 用藍牙模組實現無線傳送字元





**Q&A**



**報告到此結束**  
**謝謝大家**



# 參考資料

blender的操作方法: <https://youtu.be/O-9c4RnS0gl>

3D列印的前置設定教學: <https://youtu.be/dstlHy39zww>

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

Altium designer圖片:<https://www.altium.com/altium-designer>

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

