

# 專題 - 地震警報器



負責主題構想,硬體設計以及程式撰寫

# 專題背景與動機

當下地震很多。本專題希望透過 ESP32 + MPU6050 感測地震，結合 Python 進行連接，並透過 Discord 即時發送警報，以達到防災警示的作用。

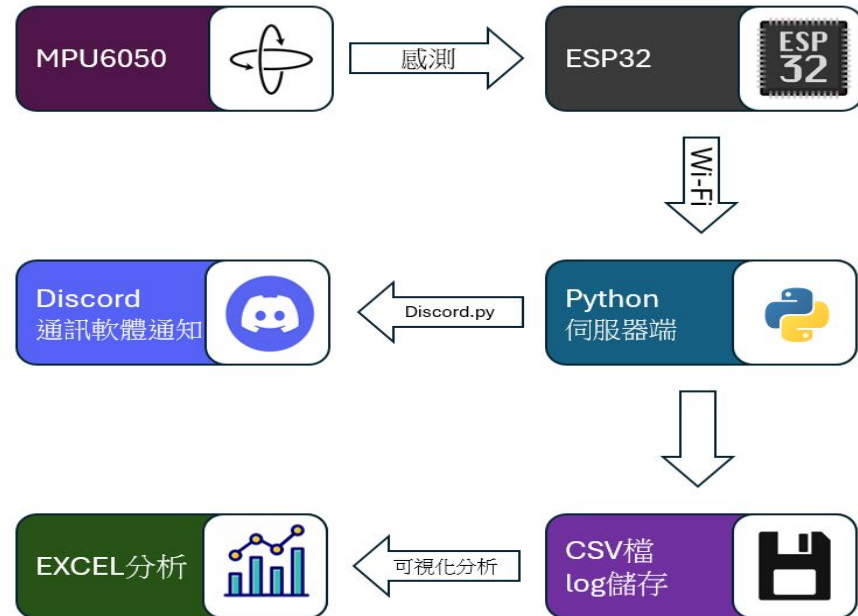
# 專題目標

- 1. 透過 MPU6050 偵測加速度變化,測算震度
- 2. 連接 Python 進行數據儲存與整合
- 3. 整合 Discord Bot 發送警報通知
- 4. 進行視覺化與分析

# 技術與工具

- 硬體: ESP32, MPU6050, server用電腦
- 語言: Arduino C, Python
- 鏈接: HTTP / socket
- 通知系統: discord.py

構想圖.



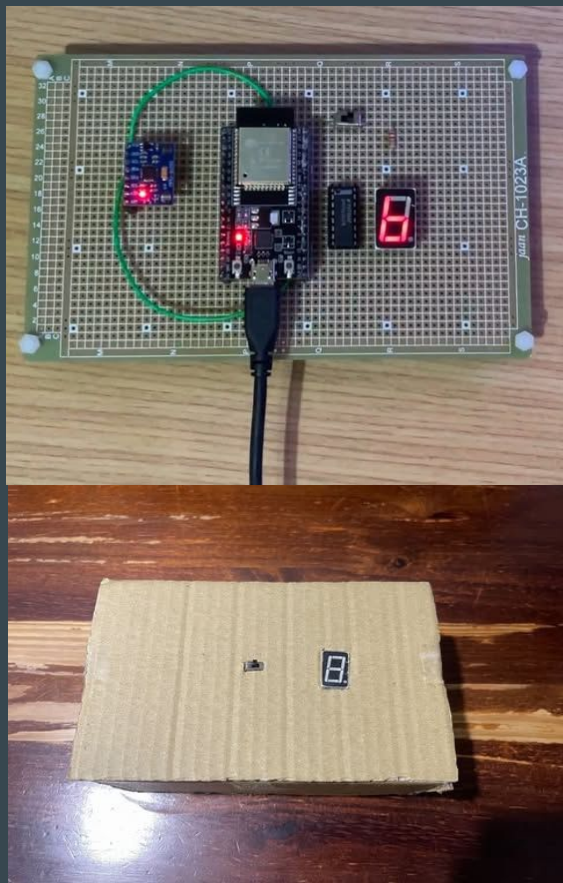
# 專題實作流程

- 1. ESP32 讀取 MPU6050 感測數據並傳至 Python Server
- 2. Python Server 進行數據儲存與地震判定
- 3. 透過 Discord Bot 發送警報
- 4. 儲存歷史數據並進行 Excel 分析

# 挑戰與解決方案

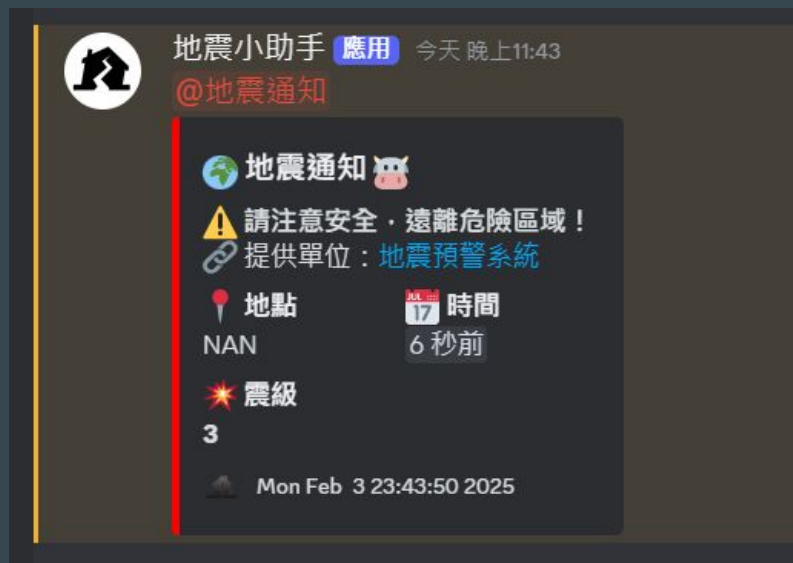
- MPU 6050 精度問題
  - 抵銷了重力及個體數值差值
- 感測數據波動過大
  - 使用移動平均法過濾雜訊
- Python 連接 discord bot RPC 429 問題
  - 設置條件式輸出

圖1.硬體與即時輸出  
圖2.為加上外殼後



# 成果

圖3.Disocrd應用通知

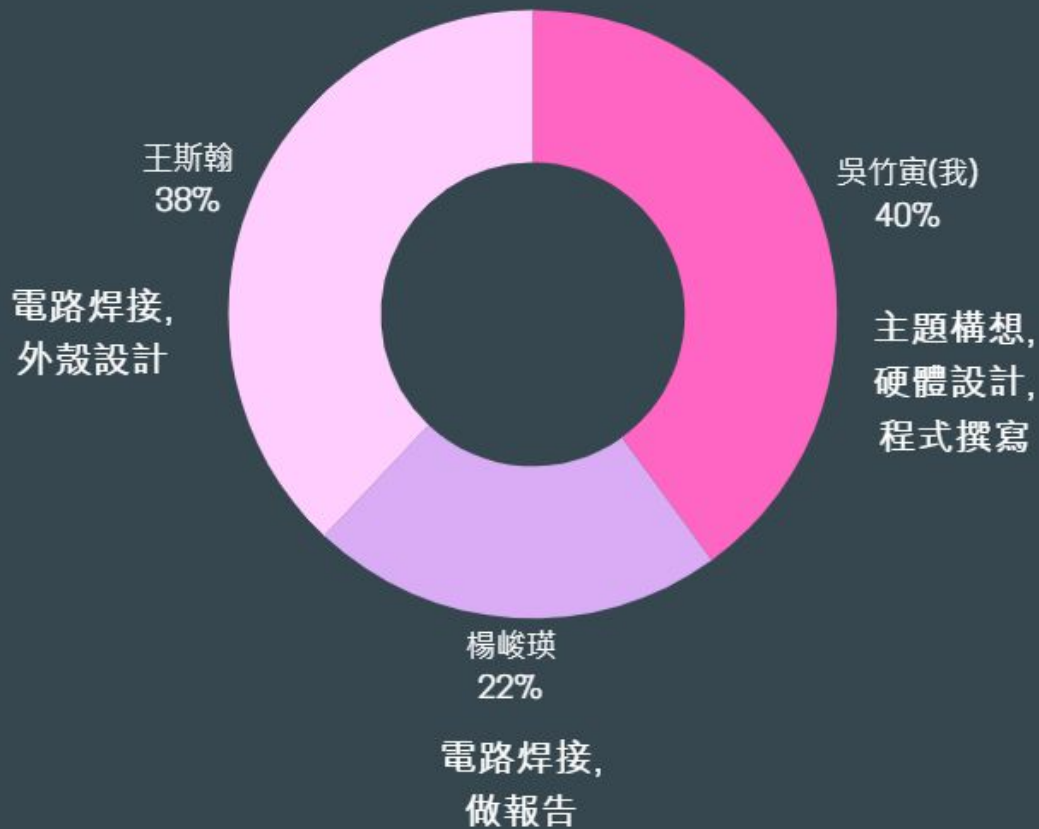


# 心得與結論

從硬體設計再到軟體設計，要產出一個成果需要很多的時間。一開始在焊完電路板後，發現了原本的接腳不能使用I2C，導致MPU6050連接不上，後來做完硬體後又遇到了區域網以及公網ip配置的問題。我感受到了計畫趕不上變化的問題，調整MPU6050的數值又調到心態很差，硬體要考慮的因素太多了每個元件的規格，配置太駁雜，我學到了需要多人協作原因，以及其中要多加幾次測試，例如先在麵包版上接線測試傳輸的功能，封裝好後感測器的數值又要調整。下次要多規劃意外發生後的處理措施。



# 分工表



# 資料來源

Python 內建函式庫 : [Link](#)

Discord API : [Link](#)

MPU 6050 使用方法參考 : [Link 1.](#) [Link 2.](#)

地震分析方法 : [Link 1.](#) [Link 2.](#)

傳輸連接方法 : [Link 1.](#) [Link 2.](#)