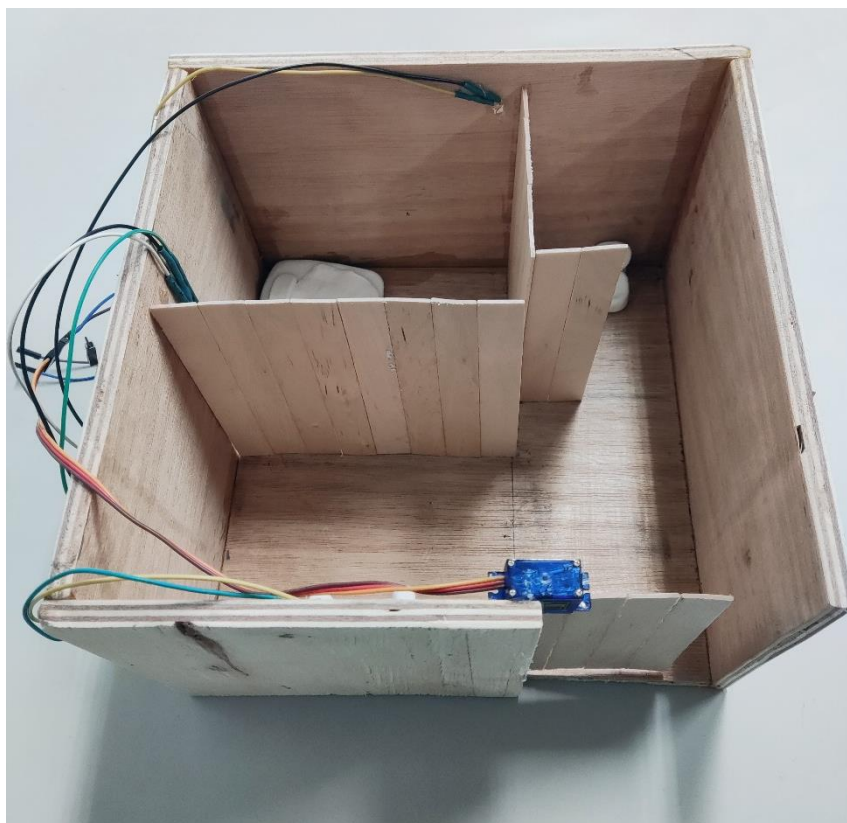


全國高級中等學校專業群科 114 年專題實作及創意競賽

「專題組」作品說明書封面



群 別：電機與電子群

作品名稱：地震斷電系統

關 鍵 詞：地震防災系統、物聯網(IOT)

目錄

壹、 摘要.....	1
貳、 研究動機	1
參、 主題與課程之相關性或教學單位之說明	2
一、基本電學、電子學.....	2
二、數位邏輯、微處理機	2
三、APP 物聯網、網頁程式設計	2
四、微電腦應用實習、行動裝置應用實習	3
肆、 研究方法、製作方法	3
一、研究方法.....	3
二、製作方法.....	3
(一)、專題架構.....	3
(二)、使用器材與軟體.....	4
(三)、製作過程.....	8
三、動作流程圖.....	12
伍、 研究結果	13
一、發生地震系統自動斷電、開啟家中房門	13
二、地震結束後使用者遠端恢復家中電源及家中房門	15

陸、 討論.....	18
柒、 結論.....	18
捌、 參考資料	19
玖、 附錄作.....	20
一、品分工表.....	20
二、競賽日誌.....	20

圖目錄

圖 1 九二一震倒塌的東星大樓	1
圖 2 花蓮大地震造成花蓮市區天王星大樓傾斜	2
圖 3 製作流程圖	3
圖 4 系統架構圖	4
圖 5 伺服馬達	4
圖 6 繼電器	5
圖 7 ANDROID STUDIO	5
圖 8 PYTHON	5
圖 9 ESP32	6
圖 10 JAVASCRIPT	6
圖 11 JAVA	7
圖 12 SQLITE	7
圖 13 MQTT	7
圖 14 WEBVIEW 關鍵程式碼	8
圖 15 WEBVIEW 執行流程圖	9
圖 16 MQTT 發送與更新控制面板	9
圖 17 SQLITE 資料庫	10
圖 18 MQTT 伺服器關鍵程式碼	10

圖 19 MQTT 處理訊息.....	11
圖 20 MQTT 伺服器程式碼執行流程圖.....	11
圖 21 動作流程圖	12

【地震斷電系統】

壹、 摘要

本專題地震斷電系統，我們使用 ESP32、手機 APP、MQTT 伺服器、繼電器與伺服馬達，提供發生地震時讓系統能夠自動切斷電源、開啟房門，提供家庭的安全性。系統以 MQTT 通訊協議作為裝置間訊息傳遞的核心，並提供使用者透過手機應用程式即時掌握家中設備狀態。我們還在手機應用程式增加開啟電源及關閉房門的功能，使在地震發生後仍可以遠端控制讓門和電力回到原本的樣子。

貳、 研究動機

由於台灣坐落在歐亞大板塊和菲律賓海板塊上，地震便是我們台灣人的日常。

發生在 1999 年 9 月 21 日凌晨 1 點 47 分地震震央位於在南投縣集集鎮，芮氏規模為 7.3 的 921 大地震所造成的慘況仍令許多人仍歷歷在目，這場地震中造就 2415 人罹難 29 人失蹤 11305 人受傷，又有多少人是因為無法及時逃生和火災而罹難的，那些都是你我身邊最親的家人。據統計那時有 51711 棟房屋全倒，53768 棟房屋半倒在地震後許多房屋坍塌變形。

在那次地震中東星大樓崩塌許須多人受困在內，在搶救過程中受火災影響而使搶救上困難重重，讓不少人錯過了最佳的黃金救援時間。在整起事件中有一件奇蹟，震後第 6 天的孫氏兄弟，在受困 130 個小時後奇蹟生還，可又有多少人像他們如此幸運呢？



圖 1 九二一震倒塌的東星大樓

在 2024 年 04 月 03 日上午 7 時 58 分 09 秒，震央位於台灣花蓮縣壽豐鄉，芮氏規模 7.2 的地震造成了許許多多的傷亡。



圖 2 花蓮大地震造成花蓮市區天王星大樓傾斜

地震劇烈的搖晃使得天王星大樓嚴重傾斜而受困在內，在這兩次地震後我們想到人們可能會為了要馬上躲到桌下和安全的位置來保護自己而無法及時關閉電源和打開房門，在地震後會有房屋變形因而無法順利逃生的情況，這成為了我們開發地震斷電系統的契機。

參、主題與課程之相關性或教學單位之說明

一、基本電學、電子學

在本專題的硬體設計和驅動方面發揮了重要作用。我們利用這些課程所學的電路設計原理，進行了 ESP8266 和 ESP32 電路的設計，以及伺服馬達和其他硬體元件的驅動控制。這些課程幫助我們理解了如何有效地配置電路，並確保整個系統的穩定性和可靠性。

二、數位邏輯、微處理機

為我們在 ESP32 與伺服馬達的程式控制提供了扎實的基礎。在這些課程中，我們學到了如何使用數位邏輯來設計和實現複雜的邏輯運算，同時掌握了微處理機的編程技術。這些知識使我們能夠編寫高效的控制程式，精確地控制硬體運作，並解決在開發過程中遇到的各種技術問題。

三、app 物聯網、網頁程式設計

雖然這些課程並未直接教授我們開發 Android 應用程式和後端伺服器的具體技術，但它們為我們提供了網頁設計和互聯網連接的基礎知識。這些基礎知識幫助我們了解了如何在網頁和互聯網環境中實現

數據傳輸和展示，並引導我們進一步學習 Kotlin、JavaScript 等新技術的興趣。

四、微電腦應用實習、行動裝置應用實習

給予了我們實踐的機會，讓我們能夠將所學的硬體與軟體技術進行整合，完成整個系統的搭建和調試。在這些實習課程中，我們學到了如何進行硬體與軟體的協同開發，並在實踐中磨練了我們的技能，提升了解決實際問題的能力

肆、 研究方法、製作方法

一、研究方法

本研究分析我國防災系統的應用案例，探討科技防災的系統運用方法，瞭解現階段我國防災技術 的運用成效及未來發展方向。首先，藉由文獻資料釐清現階段國外既有智慧科技及其技術的運用方式與現況，蒐集相關應用案例，尤其是地震防災領域，包括震前預測與預警、震時緊急應變與震後災損救災整合等階段，將各技術技運用於防災系統並整合理論與實作的各個技術運作方式。其次，透過案例分析的方式，針對智慧科技的技術，包括物聯網技術的系統運用面向、巨量資料分析的防救災機制以及多元感測器在防災的技術運用，透過多項案例瞭解目前系統運作的即時性、精準性及全面性。最後，從多項案例分析國內外防災系統技術設計與運用，使得防災作業提升效率，並提升民眾對於防災的詢求及災害應變能力。根據分析之結果，不僅統整出多項智慧防災系統的技術及核心功能，也從中延伸出對於我國實務上的應用建議，調整至我國未來在智慧防災系統方面的設計與運用安排，提出具體的結合與運用建議結果。

(一)、製作流程圖

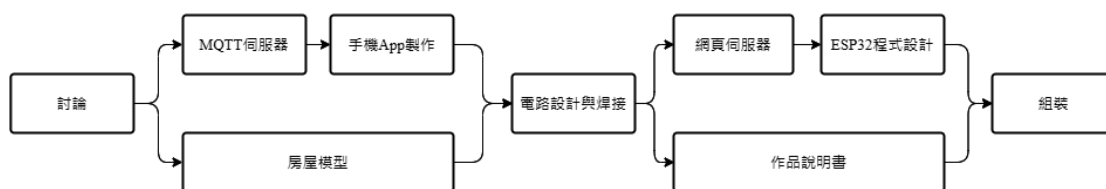


圖 3 製作流程圖

二、製作方法

(一)、專題架構

1.ESP32：作為核心控制模組，提供穩定運算能力。

- 2.伺服馬達：驅動門鎖控制。
- 3.繼電器：切斷電源。
- 4.後端伺服器：基於 Flask 架設，用於處理指令與數據。
- 5.Android 應用程式：使用 Java 開發，提供用戶友好的操作介面。
- 6.SQLite：用來儲存使用者的帳號、密碼和產品編號
- 7.MQTT 通訊：起初嘗試以 Python 開發，後改用 JavaScript 完成伺服器設置。確保通訊的安全性，加入了帳號密碼驗證功能，從而有效保護了資料傳輸的安全性。

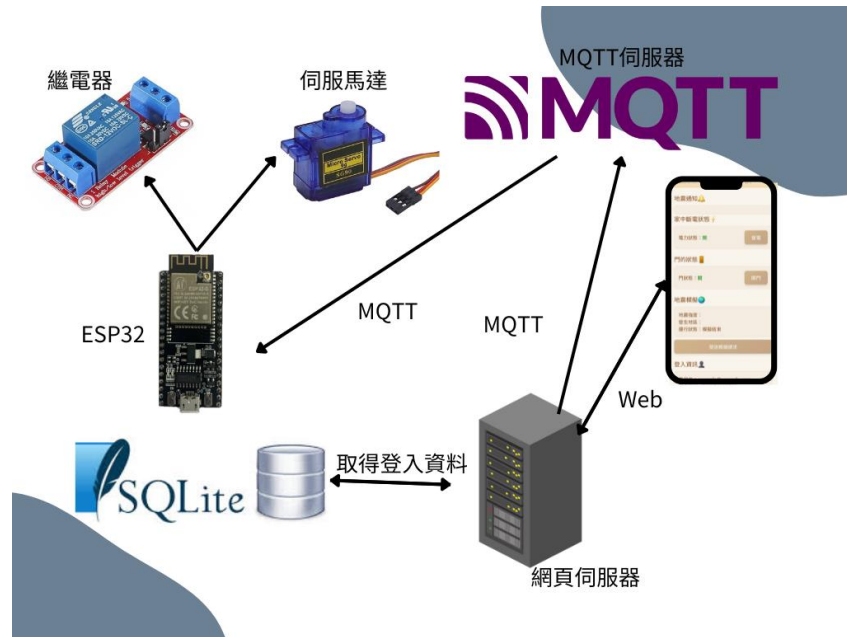


圖 4 系統架構圖

(二)、使用器材與軟體

名稱	圖片
伺服馬達	 <p>圖 5 伺服馬達</p> <p>伺服馬達負責控制房屋模型的房門開啟和關閉。通過轉向控制伺服馬達。在地震發生，伺服馬達能夠</p>

	根據系統的指示自動開啟房門。
繼電器	 <p>圖 6 繼電器</p> <p>繼電器被用來控制電路的通斷。在地震發生時，繼電器可以切斷電源。這能有效減少因電器短路引發的火災風險。</p>
Android Studio	 <p>圖 7 Android Studio</p> <p>Android Studio 是一個集成開發環境 (IDE)，適合構建 Android 應用程序。它包括用於編寫代碼、設計應用程序佈局以及模擬和測試應用程序的工具。</p>
Python	 <p>圖 8 Python</p> <p>我們使用 Python 進行伺服器端的程式控制，寫出完整精密的控制邏輯來管理與處理各個設備的數據，雖然在開發過程發生些邏輯錯誤，但透過不斷的驗證與除錯，我們最終修正了這些錯誤。</p>

<p>ESP32</p>	<div data-bbox="761 212 1077 990" data-label="Image"> </div> <p>圖 9 ESP32</p> <p>使用 ESP32 做為硬體的控制器，當 ESP32 收到 MQTT 訊息，會控制繼電器或控制伺服馬達。</p>
<p>JavaScript</p>	<div data-bbox="651 1171 1214 1720" data-label="Image"> </div> <p>圖 10 JavaScript</p> <p>在此專題中我們使用 Javascript 製作 MQTT Server 的部份，負責資料的接收、處理及分發。Javascript 非同步處理的特性讓我們能夠有效地處理多個連線的應用，在有大量的裝置同時連線時能夠維持伺服器的效能。</p>

Java	<div data-bbox="802 237 1069 719" data-label="Image"> </div> <div data-bbox="869 772 1000 806" data-label="Caption"> <p>圖 11 Java</p> </div> <div data-bbox="539 817 1340 952" data-label="Text"> <p>在本專題中，我們使用 Java 開發了 Android 應用程式，因 Java 的穩定性和豐富的庫支持讓我們能夠順利完成應用程式的開發。</p> </div>
SQLite	<div data-bbox="627 1093 1246 1290" data-label="Image"> </div> <div data-bbox="858 1444 1013 1478" data-label="Caption"> <p>圖 12 SQLite</p> </div> <div data-bbox="539 1489 1340 1624" data-label="Text"> <p>我們使用 SQLite 資料庫來管理和存儲用戶資料，包括帳號、密碼和產品編號等信息。SQLite 輕量級的設計，讓我們快速又方便的處理數據</p> </div>
MQTT	<div data-bbox="568 1662 1302 1839" data-label="Image"> </div> <div data-bbox="858 1877 1013 1910" data-label="Caption"> <p>圖 13 MQTT</p> </div> <div data-bbox="539 1921 1340 2013" data-label="Text"> <p>通過 MQTT 伺服器，網頁伺服器能夠發送訊息，讓 ESP32 接收，實現開關門與斷電復電的操作。</p> </div>

(三)、製作過程

在硬體設計階段，我們選擇了 ESP8266 作為我們的主要微控制器。這款晶片因其低功耗和豐富的功能被廣泛應用於物聯網設備中。然而，在實際設計和實施過程中，我們發現 ESP8266 缺少 5V 電源，這導致了伺服馬達運作問題。我們通過更換成 ESP32 來解決這個問題。

接下來是伺服馬達的控制。伺服馬達轉向有問題的情況使我們在測試中遇到了不少困難。我們通過重新校準和調整控制信號，最終實現了正確的轉向控制，確保了系統能夠正常運作。

Android 應用程式的開發也是一個重大挑戰。最初我們使用 Kotlin 語言開發應用程式，但遇到了許多插件版本不相容的問題，導致我們無法順利開發。後來我們改用 Java 搭配 WebView 來嵌入網頁，並成功完成了應用程式。

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); // 將布局連結到这个 Activity

    // 初始化 WebView
    webView = findViewById(R.id.webview);

    // 启用 JavaScript
    webView.getSettings().setJavaScriptEnabled(true);

    // 防止使用系统浏览器打开 URL
    webView.setWebViewClient(new WebViewClient());

    // 加载网页
    webView.loadUrl("https://ebs.mcooest.us.kg/");

    // 隐藏标题栏
    if (getSupportActionBar() != null) {
        getSupportActionBar().hide();
    }

    // 使用 onBackPressedDispatcher
    getOnBackPressedDispatcher().addCallback( owner, this, new OnBackPressedCallback( enabled, true) {
        @Override 5 個用法
        public void handleOnBackPressed() {
            if (webView.canGoBack()) {
                webView.goBack();
            } else {
                finish();
            }
        }
    });
}
```

圖 14 WebView 關鍵程式碼

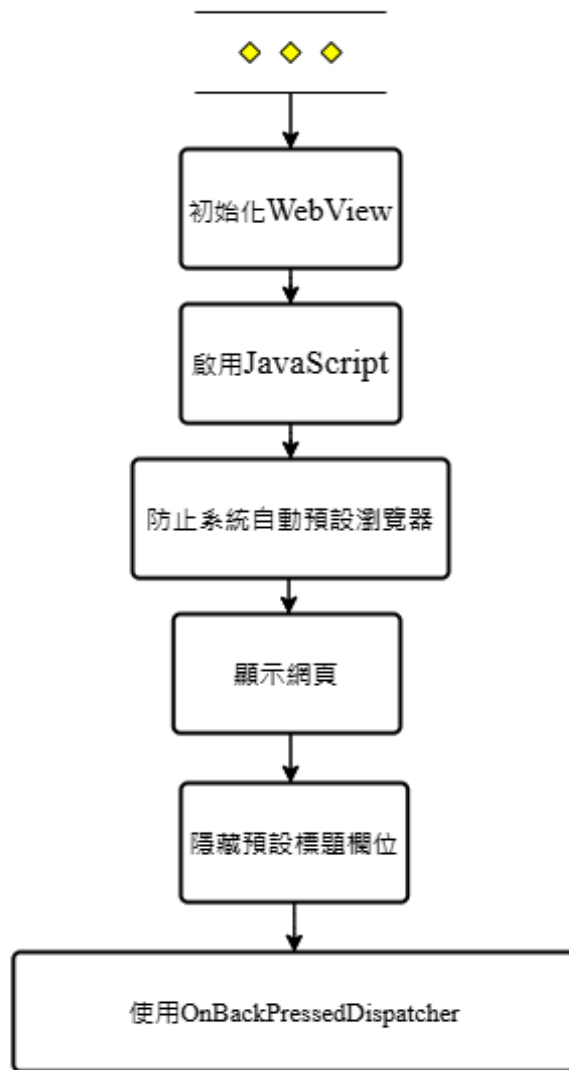


圖 15 WebView 執行流程圖

在網頁開發方面，我們主要使用 Python 進程式控制，並為其編寫了複雜的控制邏輯。然而在編寫程式時，我們遇到了邏輯錯誤，這些錯誤影響了數據處理和傳輸。我們通過反覆的測試和調試，最終修正了這些錯誤。此外，我們使用 SQLite 資料庫來儲存帳號、密碼和產品編號，這使我們能夠高效地管理和處理用戶資料，並確保數據的安全性和可靠性。

```

10.1.0.254 - - [22/Dec/2024 23:58:47] "GET /data HTTP/1.1" 200 -
10.1.0.254 - - [22/Dec/2024 23:58:49] "GET /data HTTP/1.1" 200 -
['power': '關', 'door': '動作執行中!', 'earthquake': '', 'email': 'example@example.com', 'product_id': 'E190B8805521', 'magnitude': '', 'location': '', 'state': '模擬結束', 'level': '', 'area': '']
正在連接到 mqtt_broker.mcooest.us.kg...
Door Door

正在連接到 mqtt_broker.mcooest.us.kg...

成功連接到 MQTT Broker: mqtt_broker.mcooest.us.kg
10.1.0.254 - - [22/Dec/2024 23:58:51] "GET /data HTTP/1.1" 200 -
10.1.0.254 - - [22/Dec/2024 23:58:53] "GET /data HTTP/1.1" 200 -
已成功發佈: Door
['power': '關', 'door': '關', 'earthquake': '', 'email': 'example@example.com', 'product_id': 'E190B8805521', 'magnitude': '', 'location': '', 'state': '模擬結束', 'level': '', 'area': '']
10.1.0.254 - - [22/Dec/2024 23:58:55] "GET /doorMQTT HTTP/1.1" 200 -
發佈成功, 消息ID: 71
  
```

圖 16 MQTT 發送與更新控制面板

id	email	product_id	password
1	example@example.com	E190B880S521	123
2	example2@example.com	E178B230S455	456

圖 17 SQLite 資料庫

在 MQTT 伺服器的設計上，我們起初嘗試使用 Python 開發，但始終無法成功實現所需功能。最終，我們選擇了 JavaScript 來編寫 MQTT 伺服器，並通過反覆的測試和調整，最終成功完成了伺服器的設計，使其能夠穩定地進行消息傳遞。

```

aedes.authenticate = (client, username, password, callback) => {
  const validPassword = users[username];
  if (validPassword && password.toString() === validPassword) {
    console.log(`使用者已通過身份驗證: ${username}`);
    callback(null, true); // 驗證成功
  } else {
    console.log(`使用者身份驗證失敗: ${username}`);
    callback(null, false); // 驗證失敗
  }
};

// 監聽連接和斷開事件
aedes.on('client', (client) => {
  console.log(`客戶端已連接: ${client.id}`);
});

aedes.on('clientDisconnect', (client) => {
  console.log(`客戶端已斷開連接: ${client.id}`);
});

// 接收訊息並檢查 topic
aedes.on('publish', (packet, client) => {
  const topic = packet.topic;
  const message = packet.payload.toString();

  // 過濾掉 $SYS 系統主題
  if (topic.startsWith('$SYS/')) {
    return; // 直接跳過
  }

  console.log(`主題: ${topic}, 訊息: ${message}`);
});

// 啟動 MQTT Broker
server.listen(port, () => {
  console.log(`MQTT broker 正在連接埠上運行 ${port}`);
});

```

圖 18 MQTT 伺服器關鍵程式碼


```

up to date, audited 58 packages in 8s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 10.9.1 -> 11.0.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.0.0
npm notice To update run: npm install -g npm@11.0.0
npm notice

up to date, audited 58 packages in 6s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
MQTT broker 正在連接埠上運行 23419
使用者已通過身份驗證: web1
客戶端已連接: aedes_w5R3Sro/Q/ig73pl/d7DBg/7
主題:mqtt/app, 訊息: Door

```

圖 19 MQTT 處理訊息

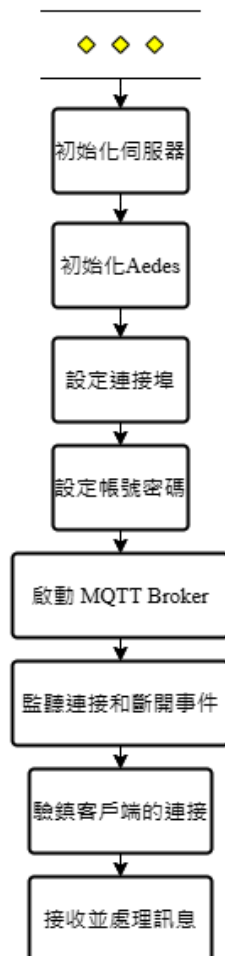


圖 20 MQTT 伺服器程式碼執行流程圖

三、動作流程圖

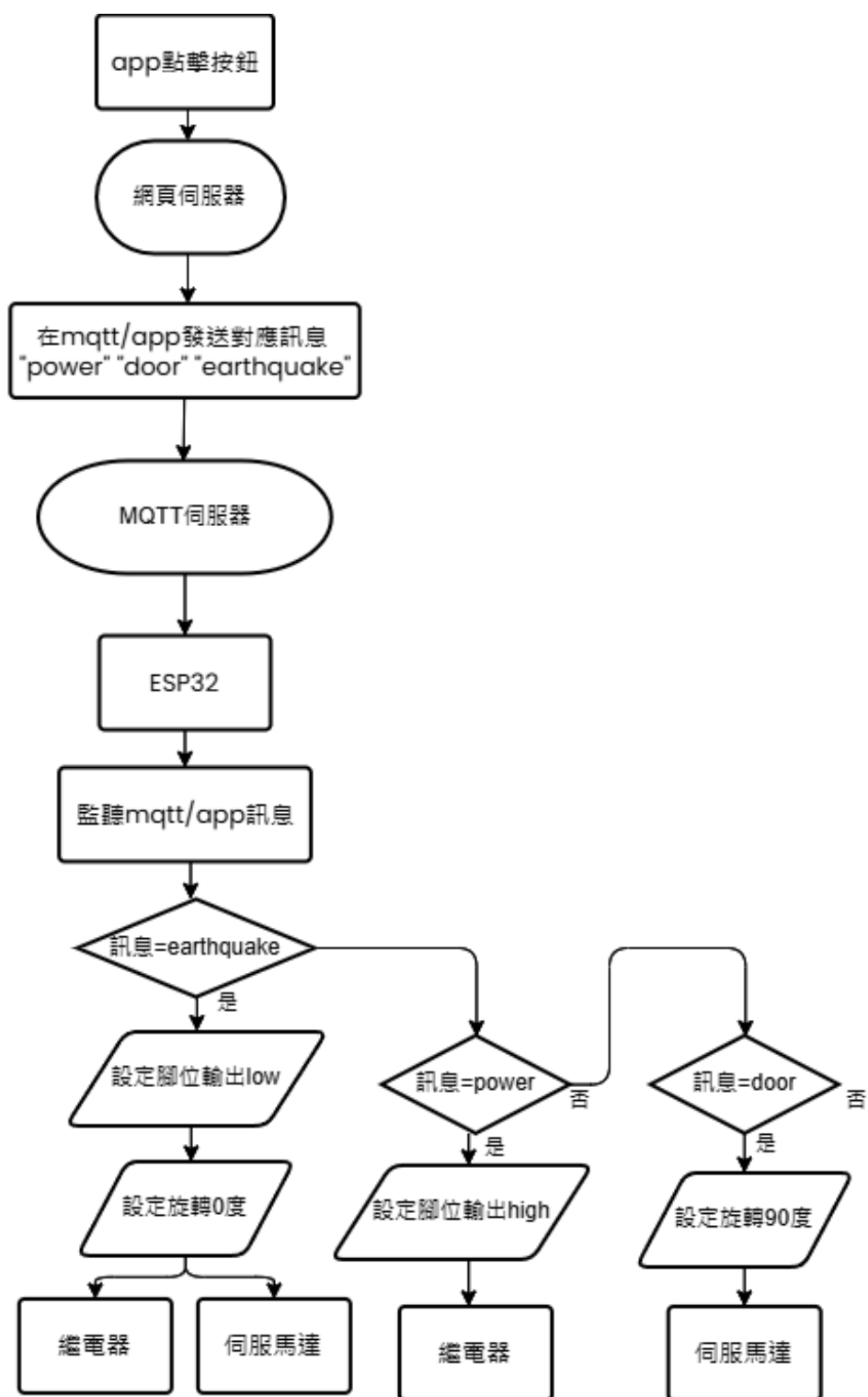


圖 21 動作流程圖

伍、 研究結果

一、發生地震系統自動斷電、開啟家中房門

1.地震模擬開始前	
	<p>控制面板供地震通知功能、家中電力狀態、門的狀態、地震模擬和登入資訊，讓使用者能遠端控制家中電源與房門。</p>
	

2.按下發送模擬請求



使用者可透過手機APP發送模擬地震請求，系統會根據操作將相應訊息傳遞至MQTT伺服器，並顯示地震強度、地區和運行狀態，進一步與硬體設備進行交互。

發送請求後可看到房屋模型電力關閉和房門開啟



二、地震結束後使用者遠端恢復家中電源及家中房門

1.地震模擬結束	
<div><div>22:52 93%</div><div>地震斷電系統控制面板</div><div>地震通知</div><div>家中斷電狀態</div><div>電力狀態：關 復電</div><div>門的狀態</div><div>門狀態：開 關門</div><div>地震模擬</div><div>地震強度： 發生地區： 運行狀態：模擬結束</div><div>發送模擬請求</div><div>登入資訊</div><div>使用者：example@example.com 產品編號：E190B880S521</div><div>登出</div></div>	<p>地震結束後可看到控制面板上家中電力與房門的開關，也可點擊復電和關門恢復原狀。</p>
<div></div>	

2. 點擊復電按鈕，恢復家中電力



點擊復電按鈕，使家中電力恢復，並在控制面板上看到電力狀態為開啟



3.點關門電按鈕，將房門關閉



The screenshot shows a mobile application interface titled "地震斷電系統控制面板" (Earthquake Power System Control Panel). The interface is divided into several sections:

- 地震通知** (Earthquake Notification): A section with a bell icon and a text input field.
- 家中斷電狀態** (Home Power Off Status): A section with a lightning bolt icon. It shows "電力狀態：開" (Power Status: On) in green text and a "復電" (Restore Power) button.
- 門的狀態** (Door Status): A section with a door icon. It shows "門狀態：關" (Door Status: Closed) in green text and a "關門" (Close Door) button.
- 地震模擬** (Earthquake Simulation): A section with a globe icon. It displays "地震強度：" (Earthquake Intensity:), "發生地區：" (Occurrence Area:), and "運行狀態：模擬結束" (Running Status: Simulation Ended). Below this is a "發送模擬請求" (Send Simulation Request) button.
- 登入資訊** (Login Information): A section with a person icon. It shows "使用者：example@example.com" (User: example@example.com) and "產品編號：E190B880S521" (Product Number: E190B880S521). Below this is a "登出" (Logout) button.

點擊關門按鈕，使家中房門關閉，並在控制面板上看到房門狀態為關閉



陸、 討論

在這次專題製作的過程中，我們面臨了許多困難，也從中學到了許多寶貴的經驗。首先是遇到的困難，包括 esp8266 缺少 5V 電源、伺服馬達轉向有問題、後端伺服器編寫程式上的邏輯錯誤，還有在開發 Android 應用程式時原本使用 Kotlin 語言但遇到各種版本不相容的問題，後來改用 WebView 仍出現一樣的問題，最終使用 Java 語言搭配 WebView 才成功完成開發。另外，MQTT 伺服器在編寫程式時，最初使用 Python 但一直無法成功，改用 JavaScript 後才順利實現功能。我們學到了許多新技能和技術，包括 WebView 的使用、新的程式語言 Kotlin、新的程式語言 JavaScript 和 Java、使用 Android Studio、MQTT broker、ESP8266 和 ESP32，以及網頁製作。透過這次的專題，我們意識到在面對自然災害，特別是地震這類突發事件時，科技的應用能夠大大提升應急反應和救援效率。我們的項目展示了如何利用物聯網技術來監測和響應地震，讓我們深刻體會到科技在保護生命和財產安全方面的巨大潛力，為社會創造更多的價值和安全保障。我們希望這個作品可以在地震中真正幫助到更多和我們一樣生活在地震帶上的生命。

柒、 結論

在未來，我們計畫將系統整合中央氣象局有感地震報告 API，這將使我們的系統能夠獲取最即時、最可靠的地震數據。這樣一來，無論是居住在地震帶的家庭還是其他對地震監測有需求的用戶，都能夠第一時間接收到準確的地震警報資訊，從而提高應對地震的準確性和效率，真正落實到家家戶戶，保障每一位用戶的安全。

此外，我們還計畫結合影像辨識技術，這樣在主震結束後，我們可以通過監控攝像頭實時監測家中人員的逃生情況。如果有陌生人闖入家中，系統也會立即通過手機 APP 發送通知，使使用者能夠及時察覺並採取相應措施。這一功能不僅能提升家居安全性，還能在緊急情況下提供重要的視覺信息，幫助使用者快速做出反應。

在設備功能方面，我們計畫加入求救信號發送功能。當房主受困時，設備可以發出有規律的警示聲，這樣可以幫助搜救人員及時發現並救援受困者。這一功能的加入將大大提高救援效率，為受困者爭取寶貴的救援時間，從而保障生命安全。

為了進一步提高地震應對效率，我們還將根據地震的強度，設計出不同的指示系統。在地震發生時，設備將根據地震的大小發出不同的指示，幫助屋主在慌亂中做出正確的反應。這一功能的設計旨在為使用者提供明確的行動指南，從而進一步保障使用者的生命安全。

最後，我們還計畫開發 iOS 系統的應用程式，讓使用 iPhone 的用戶也

能夠使用這套系統。通過擴展到 iOS 平台，我們希望能夠覆蓋更多的用戶，提供更加廣泛的服務，讓更多的人受益於這一創新的地震監測系統。這樣的拓展將使我們的系統更具普及性和實用性，進一步提升整體社會的應急反應能力

捌、 參考資料

- 1.Android Studio. (2024 , November 8). Developer.
<https://developer.android.com/studio?hl=zh-tw>
- 2.在 WebView 中建構網頁應用程式. (2024 , November 23). Developer.
<https://developer.android.com/develop/ui/views/layout/webapps/webview?hl=zh-tw>
- 3.Zheng , E. (2020 , January 17). Flash 1.0.3. PyPI.
<https://pypi.org/project/flash/>
- 4.Ghost. (2019 , November 13). 如何在 Arduino IDE 中新增 ESP8266 開發板. 傑森創工. https://blog.jmaker.com.tw/esp8266_arduino_ide
- 5.Ghost. (2022 , April 3). 如何在 Arduino IDE 中，新增 ESP32 系列開發板. 傑森創工. <https://blog.jmaker.com.tw/esp32-setup>
- 6.(, June 10). 【IOT 物聯網應用-ESP32】第三十二篇：控制 SG90 Servo 伺服馬達. 米羅科技. <https://shop.mirotek.com.tw/iot/esp32-start-32>
- 7.dekun tao. (2024 , August 2). Python 中的 MQTT 與 Paho 用戶端：初學者指南 2024. MQ. <https://www.emqx.com/en/blog/how-to-use-mqtt-in-python>
- 8.東星大樓. (2024 , April 12). 維基百科. <https://zh.wikipedia.org/zh-tw/東星大樓>
- 9.羅亦 琳. (2024 , April 4). 花蓮天王星大樓持續傾斜 社區主委認 6 年前被貼黃單. 工商時報.
<https://www.ctee.com.tw/news/20240404700724-431401>
- 10.SG90 伺服馬達 180 度(舵機). (n.d.). 今華電子有限公司. <https://jinhua.com.tw/webc/html/product/show.aspx?num=23060>
- 11.繼電器模組. (n.d.). Amazon. https://www.amazon.com/-/zh_TW/通道-5v-繼電器模組板屏蔽-帶光耦合器支援高-低級觸發相容繼電器/dp/B0D41G6T8T?th=1
- 12.Android Studio @ I/O '23: Announcing Studio Bot , an AI-Powered Coding Assistant. (2023 , May 10). Android Developers Blog.
<https://android-developers.googleblog.com/2023/05/android-studio-io-23-announcing-studio-bot.html>
- 13.Python. (2024 , December 10). 維基百科. <https://zh.wikipedia.org/zh->

- [tw/Python](#)
14. Alice musyoka. (2021, October 7). JavaScript. Webopedia.
<https://www.webopedia.com/definitions/javascript-meaning/>
15. Java. (16 C.E., September 20). 維基百科. <https://zh.wikipedia.org/zh-tw/Java>
16. Beginner's Guide to Learning SQLite: Master the Basics with Code Examples. (2024, May 15). Delta-De. <https://delta-dev-software.fr/beginners-guide-to-learning-sqlite-master-the-basics-with-code-examples>
17. 一次弄懂 MQTT ! 物聯網通訊協定全面指南. (2023, May 31). 教學資源. <https://resource.webduino.io/blog/mqtt-guide>
18. 陳冠良. (n.d.). 基本電學 (No. 04H01046-202406). 科友圖書股份有限公司.
19. 楊盛松. (n.d.). 電子學 (No. 04H02036-202405). 科友圖書股份有限公司.
20. 陳仕原. (n.d.). 微處理機 (No. 04H0403-202406). 科友圖書股份有限公司.

玖、 附錄作

一、品分工表

參賽學生	工作任務
A	模型製作、資料收集、撰寫作品說明書
B	APP 製作、MQT 通訊、資料收集、撰寫作品說明書
C	資料收集、撰寫作品說明書

二、競賽日誌

年	月	日	進度	紀錄	工作分配
113	8	17	討論製作主題	線上會議	A：主持 B：紀錄 C：提議
113	8	19	與指導老師討論 與改進原先方案	學校	A：主持 B：紀錄
113	8	20	定案、統整所需的材料	學校	A：主持 B：借場地 C：紀錄
113	8	26	學習 MQTT	學校	A：通整資料 B：操作

年	月	日	進度	紀錄	工作分配
					C：收集資料
113	9	10	MQTT 伺服器架設、python 客戶端發送與接收	組員住家	B：完成通訊功能
113	9	11	模型草圖完成	組員住家	A：完成模型草圖
113	9	26	詢問指導老師 app 前端製作	學校	B：紀錄
113	9	29	購買所需材料	商店	A：購買 B：購買 C：購買
113	9	30	App 前端改用 WebView	組員住家	B：紀錄 C：收集資料
113	10	8	房屋模型完成	組員住家	A：完成製作
113	10	12	App 前端成功	學校	B：完成製作 C：收集資料
113	10	23	撰寫做品說明書	學校	A：借場地 C：撰寫
113	10	24	撰寫做品說明書	學校	A：借場地 C：撰寫
113	11	2	控制面板製作	主員住家	B：紀錄
113	11	5	美化控制面板	學校	B：協助 C：美化
113	11	5	美化控制面板	學校	B：協助 C：美化
113	11	9	App 加入手機返回鈕的功能	線上會議	A：收集資料 B：操作
113	11	10	App 建構出現錯誤	學校	B：解決
113	11	16	App 加大能使用的 Android 版本	學校	A：收集資料 B：操作 C：收集資料
113	11	20	指導老提供建議	學校	A：紀錄 B：紀錄 C：紀錄
113	11	25	ESP8266 程式編寫	學校	B：編寫 C：收集資料

年	月	日	進度	紀錄	工作分配
113	11	30	改使用 ESP32 開發版	主員住家	B：編寫
113	12	5	控制面板新增地震模擬和狀態顏色功能	學校	A：收集資料 B：編寫
113	12	6	撰寫做品說明書	主員住家	C：撰寫
113	12	7	電路焊接	主員住家	B：焊接
113	12	8	房屋家具	主員住家	A：製作
113	12	10	電路與模型結合	學校	A：操作 B：操作 C：借場地
113	12	11	結合後的微調	主員住家	A：微調 B：微調
113	12	16	微調後的結合	學校	A：操作 B：操作 C：借場地