**高彈性智慧物聯網系統**

**AIOT System**

國立彰化師範大學資訊工程學系

113學年度大學部專題製作競賽

組員：

S0954016 戴育琪

指導老師：易昶沛 教授

編號：

**目錄**

# **一、摘要：**

我們將所要闡述的內容分為下面幾個部分。首先是簡介，對我們的專題高彈性智慧物聯網系統做一個簡要的概述，簡介之下還細分為我們專題的動機目的以及專題的特色。本專題旨在解決現代社會中存在的一個普遍問題，即老舊裝置無法輕松融入智慧物聯網系統，因難以整合，且需要購買專門支援的產品。我們的目標是開發一個高度可配置和易於使用的智慧物聯網系統，使任何人都能將其應用於現有裝置，無需購買專門的智慧家電，提供更靈活且經濟實惠的智慧家居解決方案。之後在第四章中，展現了我們的研究成果，我們用一些圖表的方式來輔助說明，最後兩章則是我們關於這個專題的結論以及相關的參考文獻。

# **二、簡介：**

## **2.1** **專題動機**

由於我們身處於物聯網急速發展，卻還有大量較為老舊，只能利用紅外線操作，甚至只能使用物理開關的老舊裝置，若要將其作系統化管理，就必須以高昂的價格去設計線路，甚至購買專門支援的產品，並且舊時代的房屋更新較為複雜，改造有一定的難度。雖然網路上有許許多多的搜尋系統，譬如 Amazon、google等等的智慧物聯網系統。但是對於智慧物聯網系統的整體環境來說，沒有一個可以使用最常見硬體零件的整合式資源，並提供一般使用者簡易的操作環境，必須去購買該公司的系統。也因此，我們希望能夠做出一個人人都能使用，不須購買專門智慧家電的高彈性智慧物聯網系統。

## **2.2** **專題目的**

本專題的主要目的是設計和開發一個智慧物聯網系統，旨在為老舊裝置提供簡便的整合方式，無需昂貴的硬體更改或專門支援的產品。具體目標包括：

1.創建一個易於使用的軟件平台，允許使用者將其現有的老舊裝置整合到智慧物聯網生態系統中，使用簡單的界面控制它們。

2.提供支援多種通信協議，包括紅外線、Wi-Fi、藍牙等，以確保適用於各種老舊裝置。

3.開發一個開放的硬體標準，允許裝置製造商輕鬆將其設備與我們的系統集成，促進更廣泛的應用。

透過實現這些目標，我們希望為人們提供一個更簡單、更具成本效益的方式，將他們的老舊裝置轉化為智慧物聯網系統的一部分，從而實現更智慧、更便捷的家居體驗。

## **2.3** **專題特色**

由於我們希望硬體端能減輕程式運作的負擔，避免發出之訊號不穩定，所以我們以micropython為開發語言，並用mqtt作為通訊協議去發送及訂閱消息。

為了讓各個使用者可以隨時管理各項硬體設備，我們以flutter開發了APP管理系統，並以Firebase資料庫來統合使用者、房間、各家電資訊，讓使用者們可以隨時遠端控制任何家電。

為了讓各個使用者有一個線上服務系統可以方便管理各項硬體設備，我們架設了一個以Laravel與Vue框架的管理系統，負責接收硬體端所發送的資料，也可以修改硬體端的狀態。另外這個系統有用到資料庫（database）來儲存使用者資料、房間資料與各家電狀態資料，做到即使使用者不在家中也能遠端控制房間內任何家電。

# **三、研究方法：**

## **3.1** **系統架構圖**

**3.1.1** **硬體**

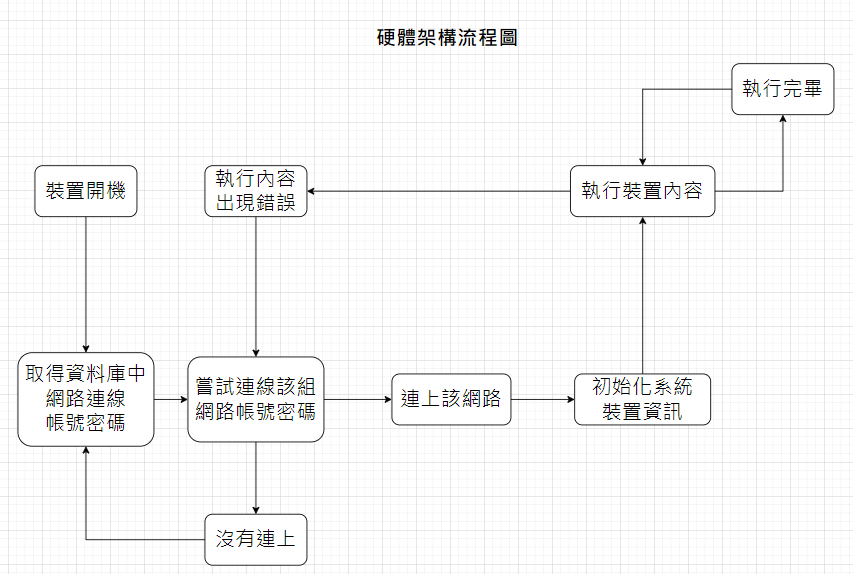


圖 XXX 硬體系統架構圖

**3.1.2** **APP**

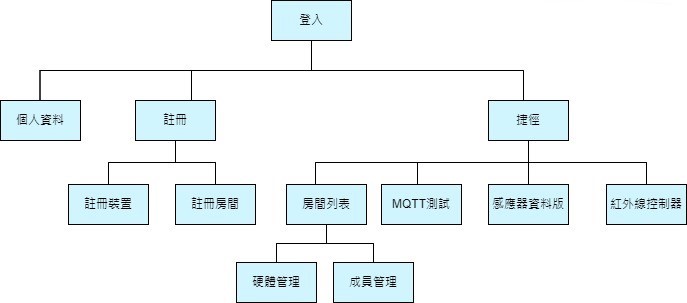


圖 XXX APP系統架構圖

**3.1.3** **網頁**

**3.2** **技術難題**

**3.2.1 硬體**

3.2.1.1 紅外線增幅

3.2.1.2 網路太差的地方，MQTT的功能會被限制

3.2.1.3 為APP增添藍芽功能時，引用 flutter\_blue\_plus模組，其僅支援低功率藍芽，而一般 藍芽無法被徵測到

**3.2.2** **APP**

**3.2.3網頁**

3.2.2.1 利用frp打穿內網，讓外網能直接連接伺服器

3.2.2.2 接硬體時，資料對接、網路連接

**3.3 專題優勢**

**3.3.1** **硬體**

3.3.1.1成本較市面上而言較低

**3.3.2** **APP**

**3.3.3** **網頁**

# **四、研究成果：**

## **4.1** **開發環境**

## **4.1.1. 硬體部分**

硬體部分: esp32 NodeMCU 32-s

使用語言：python

整合開發環境：Thonny、Vscode(Pymakr)

資料庫管理工具：自製 File System Library

溝通協定：MQTT

## **4.1.2. APP部分**

使用語言：dart(flutter 架構)

整合開發環境：Android Studio、Vscode

資料庫管理工具：flutter firebase api

資料庫系統：Firebase

伺服器管理工具：Firebase

## **4.1.3. 網頁部分**

使用語言：Vue、Laravel、PHP

整合開發環境：Docker

資料庫系統：MySQL

伺服器管理工具：dnmp

溝通協定：HTTP API server

運用技術： FRP、Docker

## **4.2** **程式模組**

**4.2.1** **資料庫：**

資料庫主要是用來儲存各式使用者資料、房間權限、硬體設備狀態，以及各使用者控制硬體的歷史紀錄。

**4.2.2** **網頁後端：**

此主程式負責處理各種存入資料庫之資料，並作為硬體端與網頁端的溝通橋梁，將資料互相傳輸，存取與更新各使用者使用硬體資料之記錄與各硬體的狀態。除此之外，身為一個管理系統，其管理包含硬體設備註冊與管理、會員註冊與管理、房間權限管理，另外附設通知功能提醒相關使用者，該設備已被更新。

## **4.3** **系統介面**

使用者介面主要參考人因介面設計，以及小組成員齊力美化。

**4.3.1 網頁介面**

當使用者登入後可選擇房間進入並操控開關，操作結果會有成功與否的通知使者，若成功並且通知其他共同控制房間的人，這項異動。