

IOS 應用程式開發

IoT 自動化設備監控

組員：

110590020 黃智勛

110590031 林暉恩

一、主題決定與功能介紹

1. APP 名稱:

IoT 自動化設備監控

2. 動機:

市面上有提供 Google Home 或是 Apple Homekit，這二者較為大宗的物聯網設備控制服務，但對於第三方設備支持不太友好(尤其是 Homekit)，所以萌生了結合 iOS 應用設計與物聯網的專題

3. 功能介紹:

- 即時設備監控: 軟體可以時時監控各種傳感器與設備狀態
- 設備可控性: 對於可控制設備軟體提供了簡單的啟用與關閉的控制
- 自動場景設備控制: 提供自動化的設備控制可以依據傳感器當前狀態對設備做自動化的控制
- 支援所有蘋果行動裝置: 基本上這個應用程式都是使用 xcode 內建元件與功能可以做到無痛上的設備移植
- 我們的應用程式允許自訂設備: 只要是 Raspberry Pi 可以支持的元件都可以加入我們的應用裡面

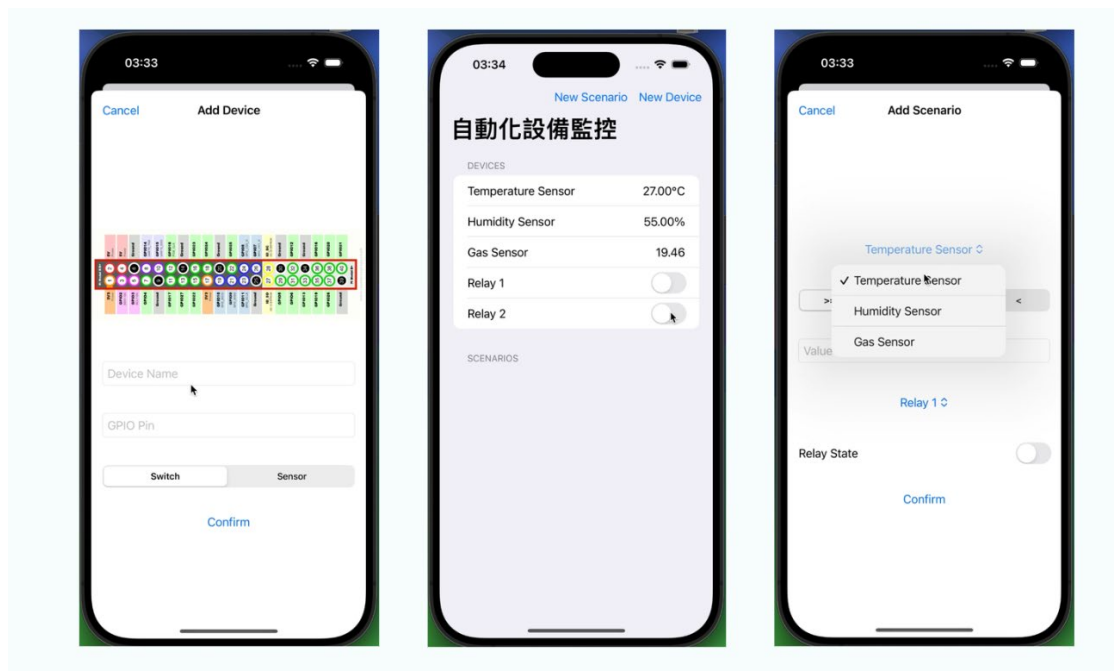
二、相近 App 介紹/比較

1. 相似軟體: Blynk IoT

2. 比較:

我們軟體有的功能，Blynk 幾乎都有而且 UI 設計真的很好看提供 widget 方便快速操作最大的不同之處，燒錄 Blynk 的開發板都需要與官方伺服器連線但是我們的軟體可以做到直接的端對端連線，無須仲介，安全

三、View Introduction



輸入裝置名稱

Cancel Add Device

Device Name

GPIO Pin

Switch Sensor

Confirm

輸入pin

傳感器與設備名稱

03:34 New Scenario New Device

自動化設備監控

DEVICES

Temperature Sensor	27.00°C
Humidity Sensor	55.00%
Gas Sensor	19.46
Relay 1	<input type="checkbox"/>
Relay 2	<input type="checkbox"/>

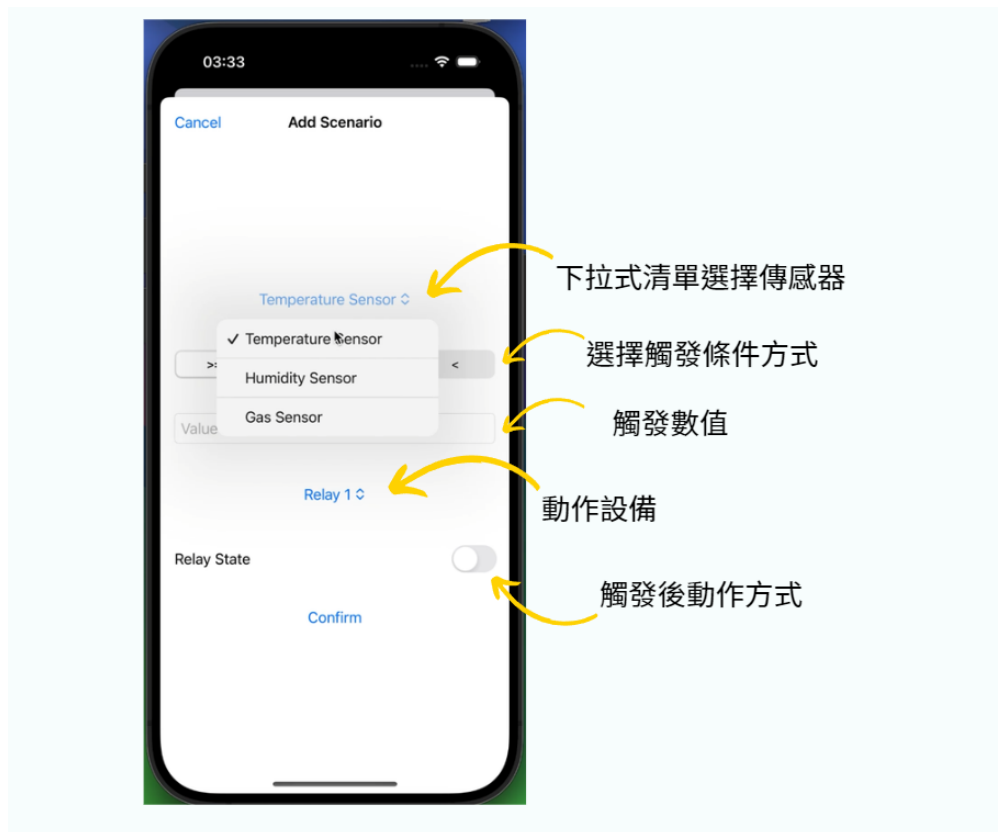
SCENARIOS

新增裝置

新增智慧場景

傳感器狀態

設備控制開關



四、 What parts of iOS will it use?

1. Device 結構：定義裝置的屬性。

屬性包含：id, name, type, gpioPin, switchStatus, sensorStatus。

2. Scenario 結構：定義情境的屬性。

屬性包含：id, sensorName, condition, value, relayName, relayState。

3. ContentView：使用 SwiftUI 的 View 協議來構建主視圖。

包含以下狀態變數：

devices: 儲存裝置列表。

scenarios: 儲存情境列表。

showAddDeviceView: 控制顯示添加裝置視圖。

showAddScenarioView: 控制顯示添加情境視圖。

timer: 定時器變數

4. 裝置列表：使用 List 和 Section 顯示裝置。

顯示裝置名稱，若為開關裝置則顯示切換按鈕以控制狀態。

5. 情境列表：使用 List 和 Section 顯示情境。

顯示情境的詳細信息，包括感測器名稱、條件、數值、繼電器名稱和狀態。

6. 導航按鈕：

Add Device: 顯示添加裝置的模式視圖。

Add Scenario: 顯示添加情境的模式視圖。

7. AddDeviceView：添加新裝置的視圖。

包含輸入欄位和選擇器來設定裝置屬性。

按下添加按鈕後將新裝置添加至裝置列表。

8. AddScenarioView：添加新情境的視圖。

包含選擇器和輸入欄位來設定情境屬性。

按下確認按鈕後將新情境添加至情境列表。

五、 Discussion

1. 困難點:

- Raspberry Pi 不支持類比輸入: 透過 ESP32 之類支持 ADC 或是 DAC 的開發板轉換
- iOS 應用程式獲取 Raspberry Pi 資料的方式: 透過 python 搭配 flask 架設簡易網站提供資料的獲取
- 一切的 Swift 語法以及更新導致對舊語法的拋棄: GPT YYDS 以及搜索網路上過來人的指點

2. 限制

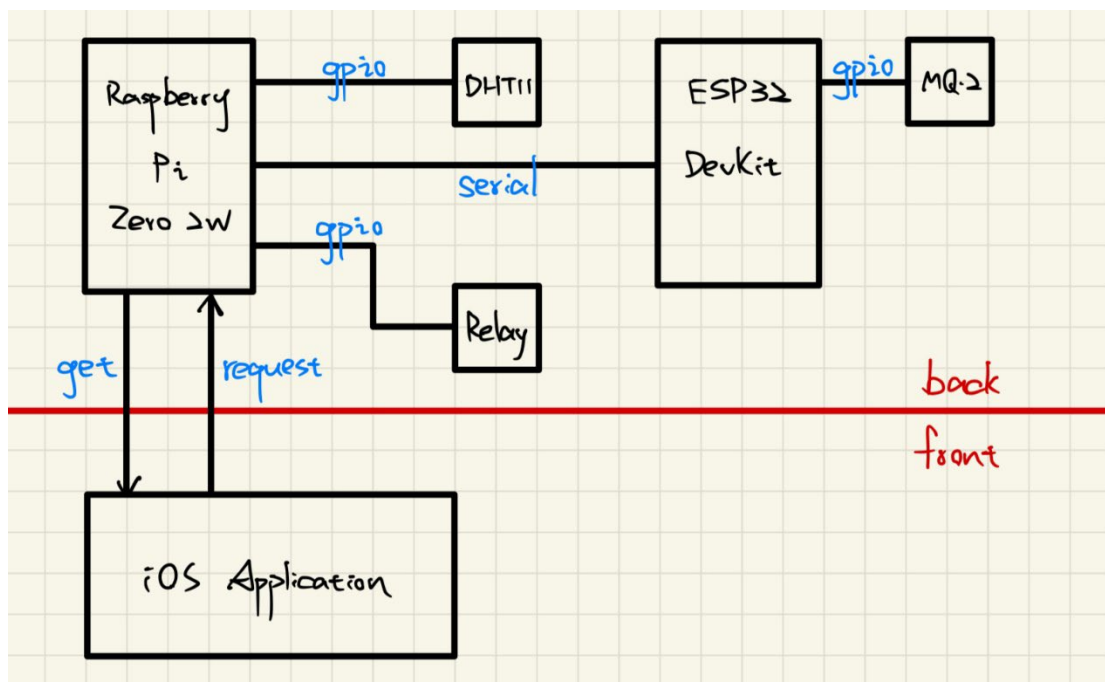
- 對於網際網路需要點了解: 外網連線的操作需要用到 Port Forwarding
- 效率低迷: Python 慢吞吞
- 設備連接上限: 一個 Raspberry Pi 撐死就 4 個電源接腳 (2*5v, 2*3v3)

3. 未來展望

- 提供更簡單便宜的上手成本
- 伺服器改用 C 或是 Cpp 撰寫
- 提供設備的 GPS 定位座標

六、 How to deployment and use our project

- For iOS Applications. It only needs to insure the Xcode version is latest.
- For rpi_server.py. It requires some dependencies by pip install, also some dependencies are only available on Raspberry Pi. Therefore, it hard to deploy on non-raspi platforms.
- For esp32_mq2.ino. It's a simple program that convert analog signal into digital signal.
- It's possible to only run iOS Applications without server running. But some features may be lost.



▲ Entire Architecture