1. 硬體準備

- 購買材料：

- 購買樹莓派 4B 作為控制中心。

- 購買感測器（亮度紅外線感測器、加速度感測器），以及必要的連接配件。

- 購買繼電器/燈泡和聲音提示器/蜂鳴器/，作為燈光控制與警報系統的執行裝置。

- 硬體安裝：

- 將亮度紅外線感測器安裝在客房的床邊或門口位置，以便偵測使用者的移動。

- 安裝加速度感測器於使用者可能佩戴的裝置上（如智能手環）或鞋底，用來偵測異常加速度（跌倒）。

- 安裝燈泡於客房牆上，確保其可以覆蓋整個行進路徑。

- 安裝聲音提示器於客房中顯眼的位置，用來發出警報提示。

2. 樹莓派設置

- 安裝樹莓派 OS：

- 將樹莓派 OS 安裝到樹莓派上，可以使用 Raspberry Pi Imager 將系統鏡像寫入 microSD 卡，並插入樹莓派進行安裝。

- 更新系統：

- 連接樹莓派到網絡並執行 `sudo apt update` 和 `sudo apt upgrade` 來確保系統為最新版本。

3. 安裝控制軟體

- 安裝Node-RED：

- 透過樹莓派安裝Node-RED，這些工具可以幫助設計和管理家電自動化邏輯。

- Node-RED 安裝命令：打開終端機，輸入 npm install -g node-red ，安裝完成後，只要繼續輸入 node-red，就會啟動 Node-RED。

- 安裝 MQTT 伺服器：

- 使用 `sudo apt install mosquitto mosquitto-clients` 安裝 MQTT，這是一個設備通訊協議，能幫助感測器與控制設備之間進行資料交換。

4. 撰寫控制腳本

- 安裝 Python 與相關庫：

- 確保樹莓派上已安裝 Python，可以使用命令 `sudo apt install python3` 進行安裝。

- 安裝必要的 Python 庫，如 GPIO 控制庫、paho-mqtt 庫等，用來控制感測器和燈光。

- 撰寫 Python 腳本：

- 人體紅外線感測器控制燈光：

- 腳本會持續監控 PIR 感測器，當偵測到動作時，透過 GPIO 控制智能燈泡自動開啟，並根據夜間條件調整亮度與色溫。

- 加速度感測器控制防跌倒系統：

- 加速度感測器的數據可以經由腳本分析，如果檢測到異常變化，觸發聲音提示器發出警報，並使用 MQTT 發送通知。

5. 設置自動化邏輯

- 設置燈光控制邏輯：

- 在Node-RED 中設置規則，當亮度紅外線感測器偵測到使用者半夜起床時，調節燈光亮度，讓燈光柔和不刺眼。

- 設置防跌倒邏輯：

- 配置加速度感測器邏輯，如果檢測到跌倒，啟動聲音提示器並透過 IFTTT向家人或醫護人員發送通知。

6. 感測器校準與測試

- 亮度紅外線感測器校準：

- 測試紅外線感測器的靈敏度，調整其靈敏度以便能夠準確偵測到夜間光線暗時，開啟壓力感測器，避免連接的燈光一直開啟

- 加速度感測器校準：

- 測試加速度感測器偵測到的數據範圍，並設置合適的跌倒判定邊界，防止誤報。

7. 系統測試與優化

- 夜間燈光控制測試：

- 模擬夜間場景，測試感測器是否在半夜起床時自動開啟燈光。

- 跌倒偵測測試：

- 模擬不同情境下的跌倒，檢測加速度感測器的準確度，確保在跌倒發生時能夠即時觸發警報並發送通知。

- 系統反應速度測試：

- 測試系統從感測器數據變化到燈光控制或警報的反應時間，確保系統即時反應，並進行必要的優化。

8. 使用者介面設置

- 設置 Web 或 App 介面：

- 使用Node-RED 設置簡單的 Web 界面，讓使用者可以通過手機或電腦查看系統狀態、手動控制燈光或警報。

- 通知功能測試：

- 測試 IFTTT的通知功能，確保當偵測到跌倒時，家人或醫護人員能夠即時收到通知。

9. 材料經費評估

|  |  |
| --- | --- |
| 材料 | 台幣 |
| 樹莓派 4B | 2,000 |
| 光線感測器\*1(LM393) | 14 |
| 加速度感測器\*1(GY-362) | 164 |
| 壓力感測器 | 135 |
| 燈泡110V/5V | 120/30 |
| 聲音提示器/蜂鳴器\*2 | 70 |
| 繼電器12V\*1 | 60 |
| 線材與其他配件(杜邦線) | 100 |
| MQTT伺服器租用 | 8.99美元/月，約3458 |
| 總計 | 約3033元 |

10. 使用的硬體

1. 樹莓派 4B：作為核心控制單元。

2. 人體紅外線感測器：偵測室內亮度，光線暗時開啟壓力感測器

3. 加速度感測器：偵測異常加速度，預防跌倒。

4. 壓力感測器：偵測下床按壓地墊，觸發電燈開起

5. 聲音提示器：發出警告聲或緊急提示。

11. 使用的軟體

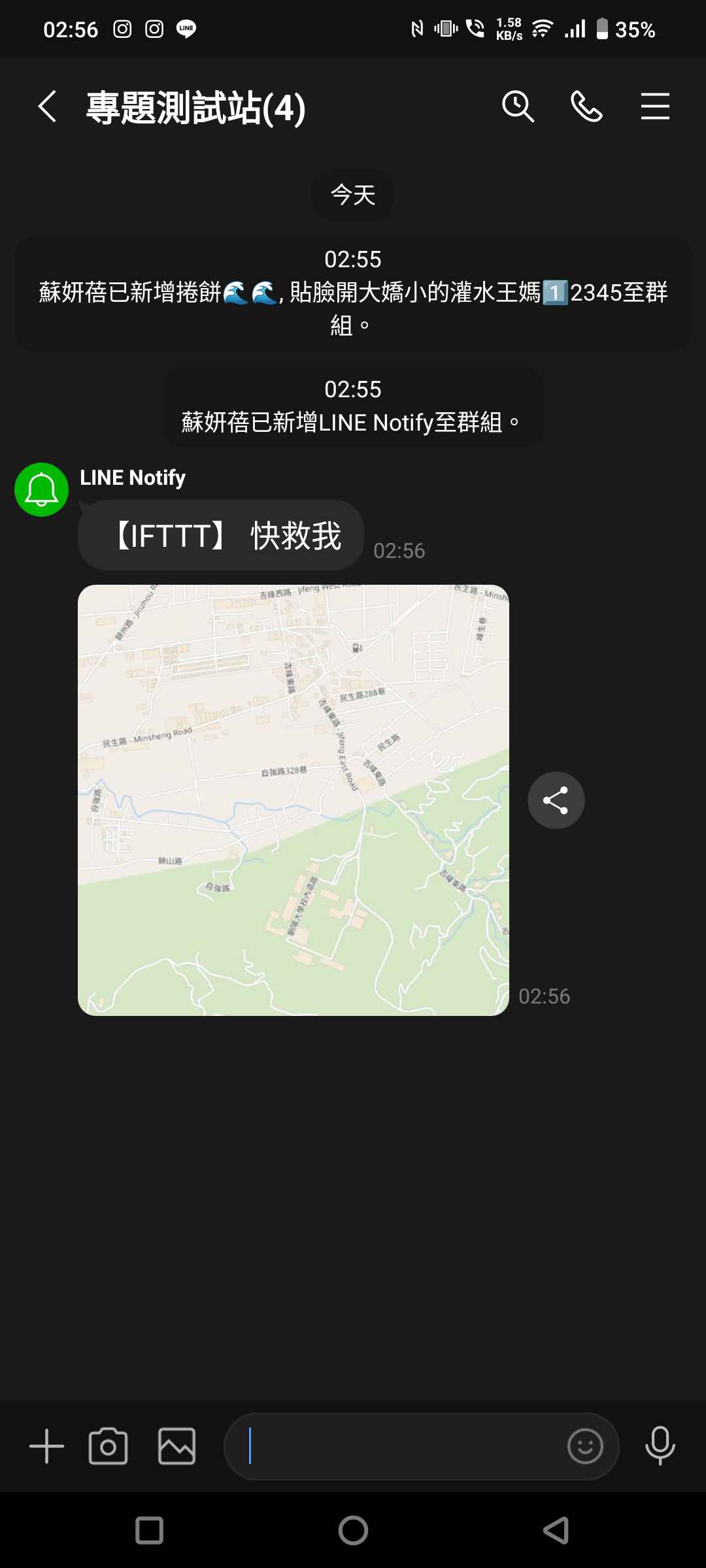
1. Raspberry Pi OS：樹莓派的作業系統。

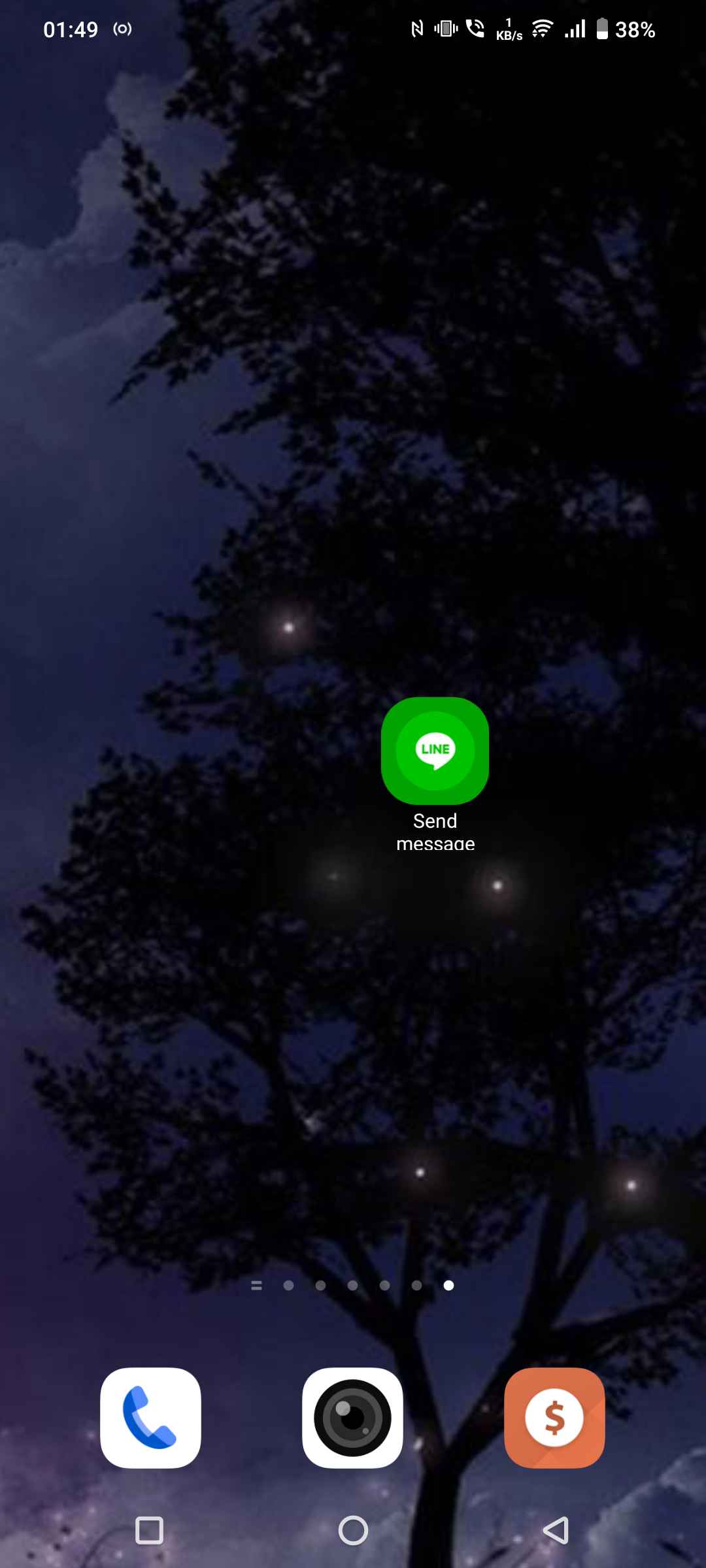
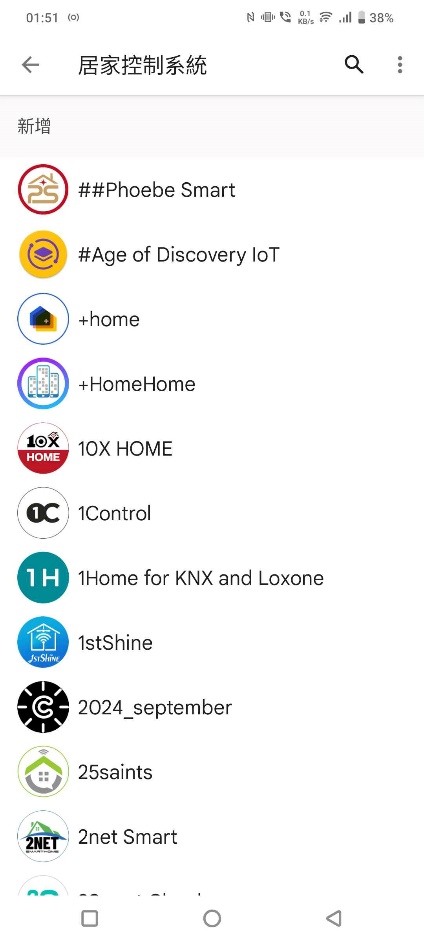
2. Python：控制感測器和燈光、自動化邏輯的編寫語言。

3. Node-RED：用來設置自動化規則和邏輯控制。

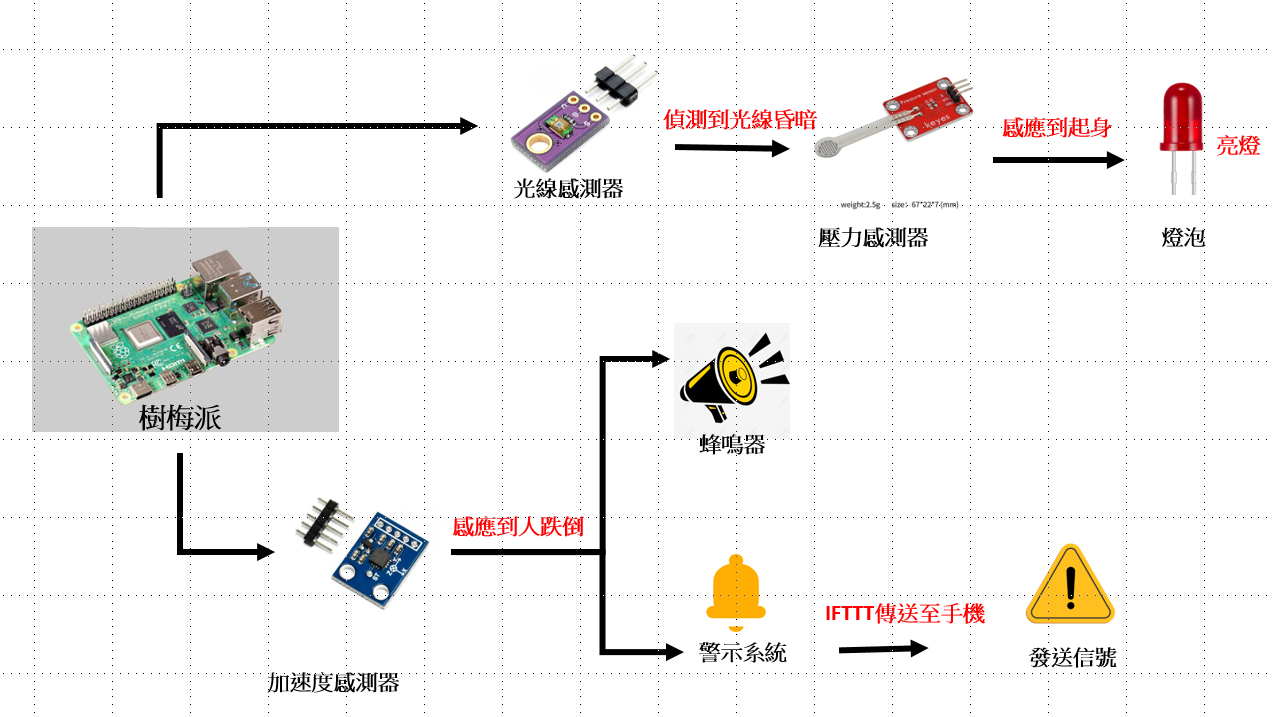
4. MQTT：用來進行設備之間的通訊。

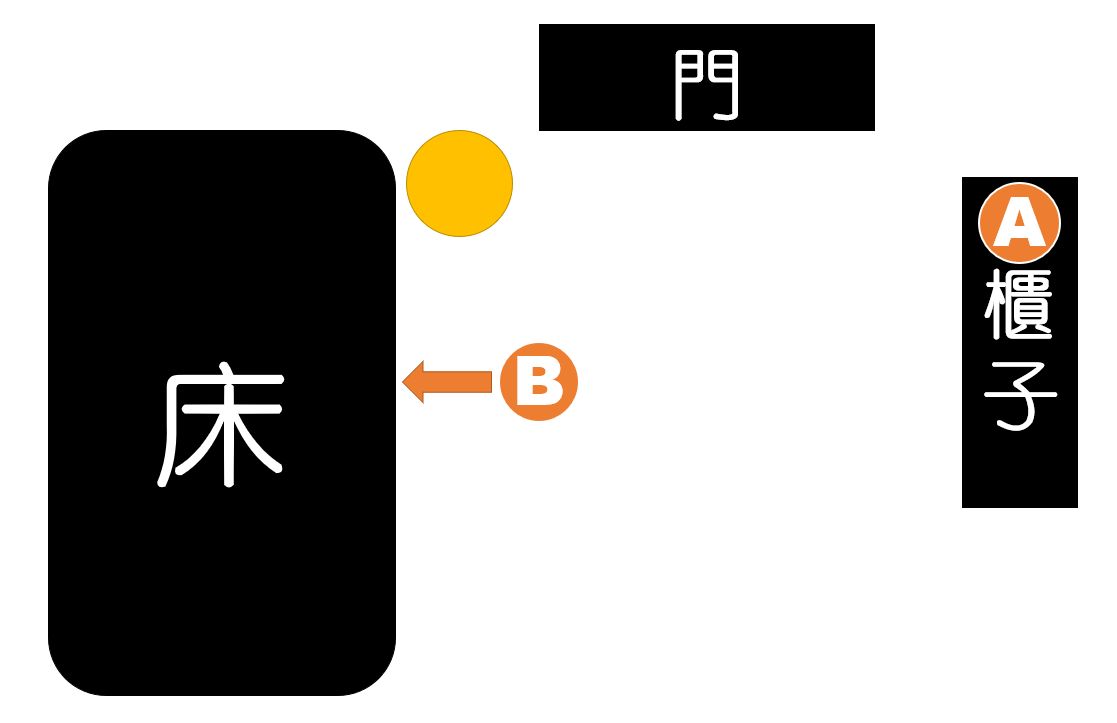
5. IFTTT：用來發送跌倒通知或求救信號

IFTTT操作圖:



整合裝置圖:





A:光線感測器、B:壓力感測器