

# 嵌入式系統實驗 Final Project Report - 防盜系統

電機四 陳玟瑋 B10901044

電機四 高鉅茗 B10901159

## 一、 專題動機

隨著科技進步，提升居家安全並有效防止陌生人或竊賊入侵住家已成為現代生活的重要需求。為了強化門戶的防護能力，我們設計了一套具備智慧偵測、警報通知與遠端管理功能的防盜系統，旨在針對出入口進行精準監控。

本專題利用 STM32L475 開發板作為核心，結合超音波感測器與伺服馬達轉動模組，建構一套專門部署於大門正前方的監控裝置。透過自動掃描與偵測機制，系統能涵蓋正前方 180 度的防禦範圍，並整合了警鈴（蜂鳴器）與藍牙低功耗（BLE）無線通訊技術。

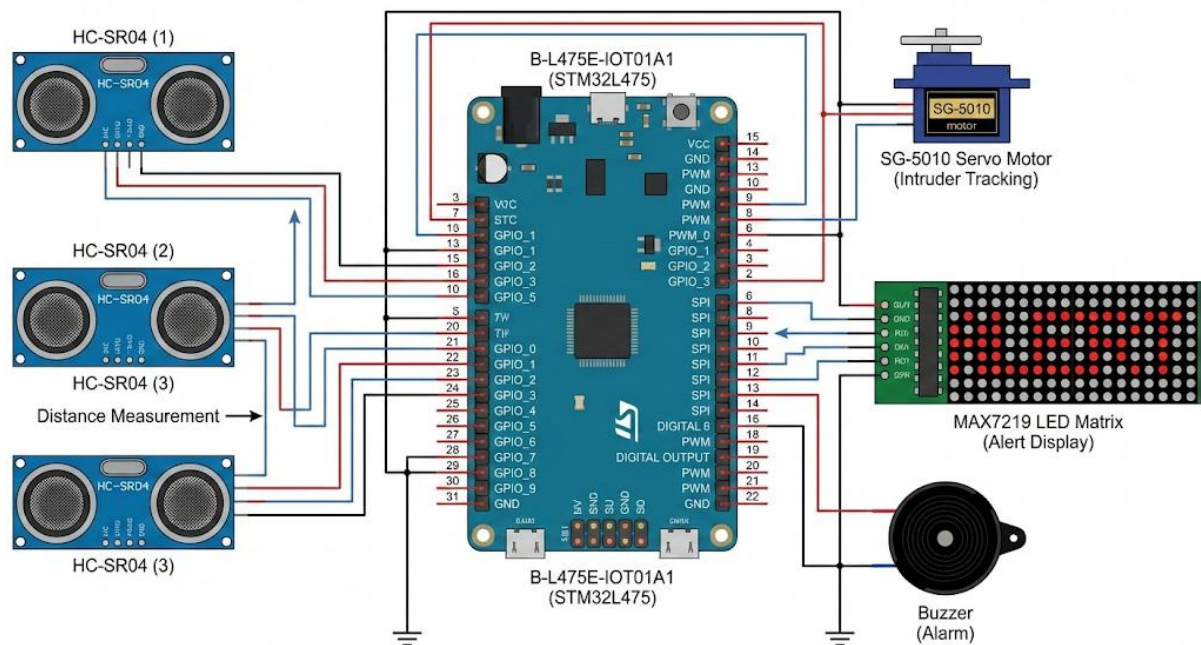
這套系統讓使用者能透過行動裝置遠端切換警戒狀態，一旦在監控扇形區域內偵測到異狀，系統將即時發送通知，確保使用者能第一時間掌握門前異常狀況，達成智慧化的居家防衛。

## 二、 系統做法與架構

### 1. 硬體架構

系統核心基於 B-L475E-IOT01A1 (STM32L475) 開發板，並整合以下周邊元件：

- 感測元件：HC-SR04 超音波模組（3組），用於量測距離。
- 致動元件：SG-5010 伺服馬達（追蹤入侵者位置）與 MAX7219 LED 矩陣（警示顯示）。
- 報警元件：蜂鳴器。



## 2. 軟體設計與工作排程 (RTOS)

系統採用 RTOS 進行多工管理，主要有4項任務：

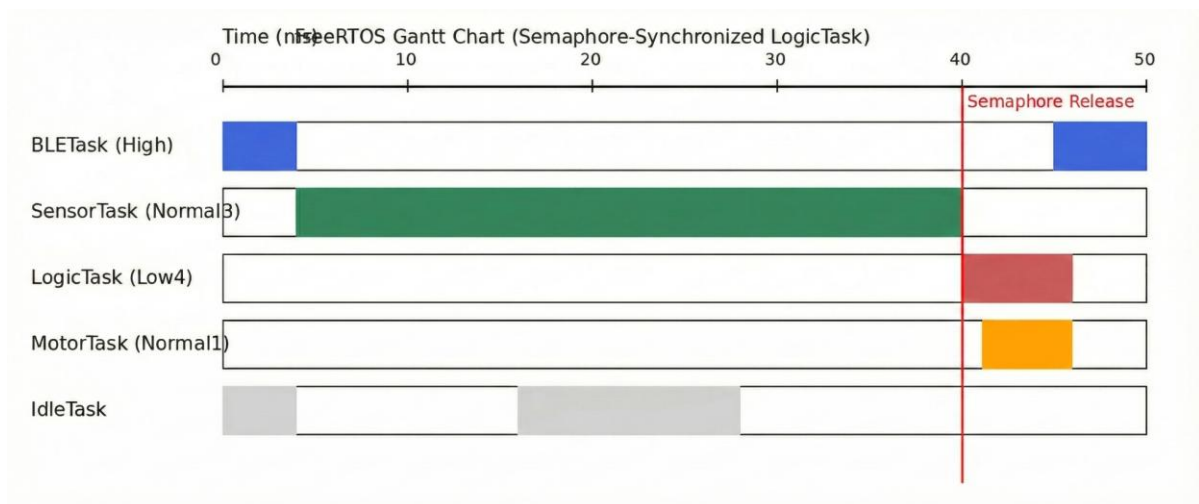
1. BLE Task
2. Sensor Task
3. Motor Task
4. Logic Task

考慮連線品質最優先，因此BLE priority最高，而邏輯判斷(是否入侵、偵測位置)則需要超音波的資料回傳才能進行，於是Sensor 排在BLE之後、Logic Task前，又馬達轉動是為了追蹤人的位置，因此在判斷完當下就應立即轉向，因此排在Sensor之後、Logic之前，得各任務依優先權排序及功能如下：

1. **BLE Task (High)**：負責與網頁端連線，接收控制指令與傳送 Notify 通知。建立一個Service儲存兩個Characterisitics，一個負責紀錄警戒是否開啟(0關/1開)，另一個紀錄當下是否有偵測到人(0無人/1有人)，每次輪到該任務進行時根據兩項Characteristics進而決定當下是否Notify(警戒開啟且有人)，之後空出50ms的delay讓其他任務可以執行。
2. **SensorTask (Normal3)**：負責超音波量測，與Logic Task之間共用Semaphore確保最新資料的使用，ECHO使用EXTI Callback的方式更新回傳的數值。由於 RTOS 最小延遲為 1ms，系統改用 System Clock 實作1 us精度的延遲以產生 Trig

訊號。為了避免超音波模組間相互干擾，每個Trig間隔12us(考慮偵測範圍約2公尺即可)，並在最後加入14ms的delay，以空出時間讓Logic Task可以執行

3. **MotorTask (Normal1)**：驅動伺服馬達，Motor Task會等待Queue裡面的指令，執行時機為當Logic Task發出轉動角度的訊息到Queue中，馬達便會因高優先權而中斷Logic進行轉動。主要使用 Timer 5 產生 50Hz (20ms 週期) 的 PWM 訊號，透過 500 us至 2500 us 的脈波寬度控制馬達於0至180度間轉動。
4. **LogicTask (Low4)**：邏輯核心，包含判斷：1. 警戒是否開啟、2. 是否有人(<50cm，並設置滯留區間50cm~65cm避免頻繁切換)及位置在哪(左、中、右)、3. 根據是否有人決定是否執行亮燈及蜂鳴器。為了確保判斷時使用的資料是最新的，會等待 SensorTask 釋放 Semaphore 後才執行，之後若判斷入侵狀態則透過 **MotorQueue** 發送指令給馬達(左:45°、中:90°、右:135°)並使MAX7219全亮，反之全暗。



### 3. 網頁端

使用HTML方便手機或電腦透過BLE連線，控制是否開啟警戒(Write Request)，以及接收是否入侵的通知(Notify)，當警戒關閉時呈上鎖圖案，開啟時根據是否有人顯示監控中(綠色)或入侵警報(紅色)，如下圖：

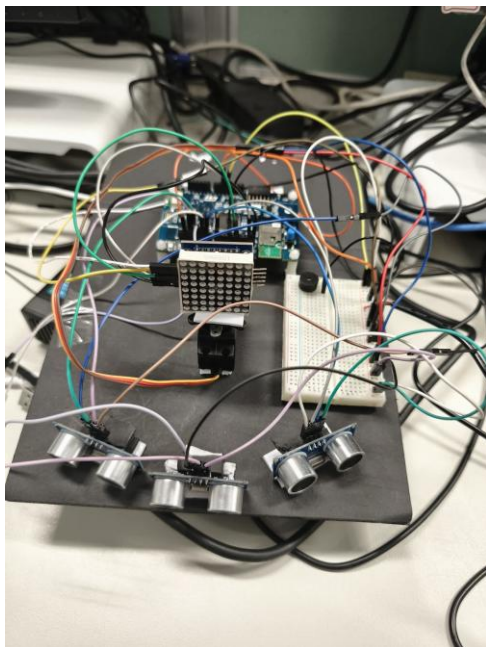


警戒關閉

警戒開啟:安全

警戒開啟:入侵

### 三、實作成果



#### 1. 遠端監控網頁介面

本系統開發了專屬網頁介面作為 Server 端：

- **狀態控制：**使用者可透過 Write 指令控制 STM32 進入 Active（警戒）或 Inactive（待機）模式。
- **即時回饋：**網頁會顯示「連線狀態」、「系統模式」及「感測器狀態」。
- **入侵警報：**當偵測到物體時，介面會即時顯示「偵測到物體」與「入侵警報」字樣。

## 2. 系統演示

### 2-1. 模式切換與觸發機制

系統具備明確的狀態管理，區分為「待機」、「安全」與「入侵」三種模式：

- **佈防前狀態：**在 BLE 裝置連線前，或連線後但尚未透過網頁開啟 Active 指令時，系統不進行距離判定，即使物體進入 50 cm 內也不會觸發警報。
- **入侵判定：**開啟「警戒模式」後，系統整合左、中、右三個超音波感測器資訊。若任一感測器測得之最短距離在 50 cm 以內，系統即刻切換至「入侵模式」。
- **警報執行：**進入入侵模式時，MAX7219 LED 矩陣將全亮，且蜂鳴器持續響起，直到狀態解除。

### 2-2. 精準追蹤與動態控制

系統透過伺服馬達實現「視線」追蹤功能，確保監控核心始終對準入侵者：

- **最小距離優先：**LogicTask 會比較三個感測器的數值，以距離最低（即最靠近人體）的方向作為馬達指向目標。
- **角度對應：**
  - **正前方：**若中央感測器數值最低，SG-5010 伺服馬達指向90度。
  - **左前方/右前方：**當人移動至側邊，對應感測器數值變為最低，馬達會帶動 LED 燈轉動 45度 進行指向鎖定。
- **反應時效：**系統確保每次轉動的處理延遲不超過 2 秒，維持追蹤的即時性。

### 2-3. 滯後處理與通訊策略

為了避免入侵者在臨界點徘徊導致警報反覆開關，系統設計了緩衝機制：

- **緩衝距離機制：**系統進入入侵模式後，LED 與蜂鳴器會維持開啟，直到入侵者退至 65 cm（緩衝距離）以外，系統才會切回安全模式。
- **BLE 資訊傳輸：**
  - **狀態變更通知：**在「安全模式」與「入侵模式」相互切換的瞬間，STM32 會主動向連線裝置傳送資訊。
  - **持續監控通知：**若系統持續維持在入侵模式，會穩定地向連線裝置發送 Notify 訊息，確保遠端使用者能掌握即時現況。

影片連結(第一人稱視角):

[NTU EE 114-1 ES-Lab Final Project 防盜系統 陳玟瑋 高鉅茗](#)

## 3. 遇到的問題

如影片所示，人靠近時的確會偵測到，但當移動後馬達的轉動無法及時跟上人的移動速度，且亮燈的部分會再馬達轉動時停止，當停止時才會再次亮起。這裡我們思考原因可能一部分是藍牙處理耗時較久，又因BLE優先級最高，於是每次都會被耗時較久的BLE打斷。而BLE耗時較久的原因可能是Service的建立，由於我們的Service除了額外建立用來記錄狀態的之外，還保留了原本其他的Service，因此多了一部份的時間要處理用不到地方。

原本推測馬達啟動但燈不亮，可能是因為亮燈功能在 Logic Task 執行時被馬達（優先序較高）強制中斷。但按理來說，MAX7219 在 SPI 沒有訊號輸入時，應該會保持之前的顯示狀態，所以當馬達運轉時，LED 理論上不會自動全部熄滅。較合理的推測是電源供應瞬間不足。進一步分析原因，SG-5010 伺服馬達在啟動瞬間（start-up / inrush）會產生顯著的瞬間電流需求，其電流大小通常接近堵轉電流（stall current）等級，約為 0.5~0.7 A（500~700 mA，@5~6 V），此電流遠高於一般運轉時的 100~300 mA。因此，在馬達啟動或方向切換的瞬間，會瞬間拉走大量電流，造成系統 5 V 電源電壓下跌（voltage drop），MAX7219 無法獲得足夠的工作電流，導致顯示異常或全暗。

綜合以上兩點問題，BLE考慮的調整方向首先是刪掉不必要的Service減少處理量，再來可以調低優先序，也許放在最後，因為整套流程應不會耗費太久時間，不至於讓BLE斷線。至於亮燈的狀況，可能要再給馬達獨立的電源供給，讓整個系統的供電能更為穩定。

此外，在開發過程中，原計畫結合 Wi-Fi 傳送通知，但考量到 BLE 與 Wi-Fi 共用 SPI3 介面，且兩者資料位元長度不同（BLE 為 8-bit，Wi-Fi 為 16-bit），切換頻繁易導致系統不穩定，故最終聚焦於穩定度較高的 BLE 通訊解決方案。

### 參考資料

HC-SR04超音波測距模組：[https://blog.csdn.net/weixin\\_35815766/article/details/152659222](https://blog.csdn.net/weixin_35815766/article/details/152659222)

SG-5010伺服馬達：<https://docs.cirkitdesigner.com/component/44530394-fe5b-47d0-b423-bb55a0499d69/sg5010-servo>

GitHub連結：<https://github.com/AWeil01602/NTU-EE-114-1-Embedded-System-Lab/tree/main/Final%20Project>