

嵌入式系統概論 Final Report

102034020 王子洋

系統目標：

將二階段認證技術應用在 Arduino 上，實作出高安全性的嵌入式系統

功能：

本系統分為兩個部分：伺服器端、客戶端

客戶端由 LCD 顯示，共提供三個功能：登入、修改密碼、顯示時間

1. 登入: 使用者依序輸入“帳號” -> “密碼” -> “二階段驗證碼” 來核對使用者身分
2. 修改密碼: 使用者依序輸入“帳號” -> “密碼” -> “新密碼” -> “二階段驗證碼” 來修改密碼
3. 顯示時間: LCD 將顯示當前時間提供給使用者

伺服器端主要提供建立帳號與顯示 LOG 功能

1. 建立帳號: 提供使用者建立新的帳號，如下圖

Register

Username

Password

Password again

Register

Two Factor Authentication Setup

Please use a google authenticator



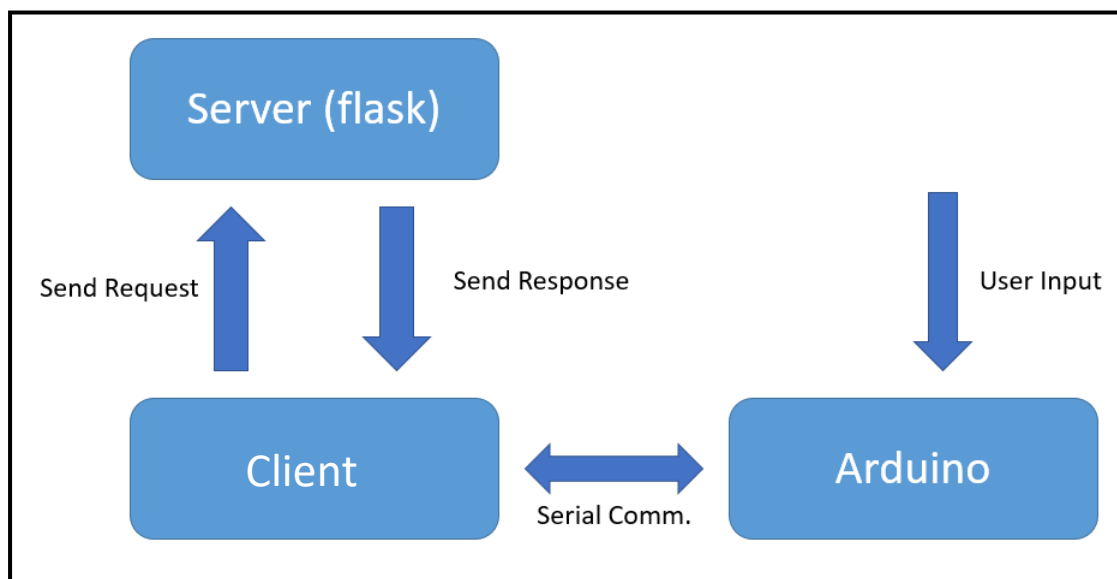
2. 顯示 LOG: 顯示登入狀況，供使用者查詢是否有異常，如下圖

Log

id	time	user	log
11	2016-12-29 22:11:40.542493	B	B login
10	2016-12-29 22:11:08.117166	B	B updated password
9	2016-12-29 22:10:36.814960	B	B login
8	2016-12-29 22:10:00.289769	B	B failed to login
7	2016-12-29 22:09:27.413520	B	B failed to update password
6	2016-12-29 22:08:36.409977	-	Add user: B
5	2016-12-29 22:08:10.637925	A	A updated password
4	2016-12-29 22:07:18.437895	A	A failed to login
3	2016-12-29 22:06:16.908608	A	A login
2	2016-12-29 21:55:24.374616	A	A login
1	2016-12-29 21:54:51.218461	-	Add user: A

實作細節：

主要架構圖



Arduino 部分

Arduino 主要是接受外在的資訊，我用了以下工具:

1. Timer: 用於計時
2. DAC:
 - A. 蜂鳴器:輸入密碼後，給予反饋
 - B. LED: 登入成功 給反饋
3. LCD: 顯示使用者介面
4. Sensor

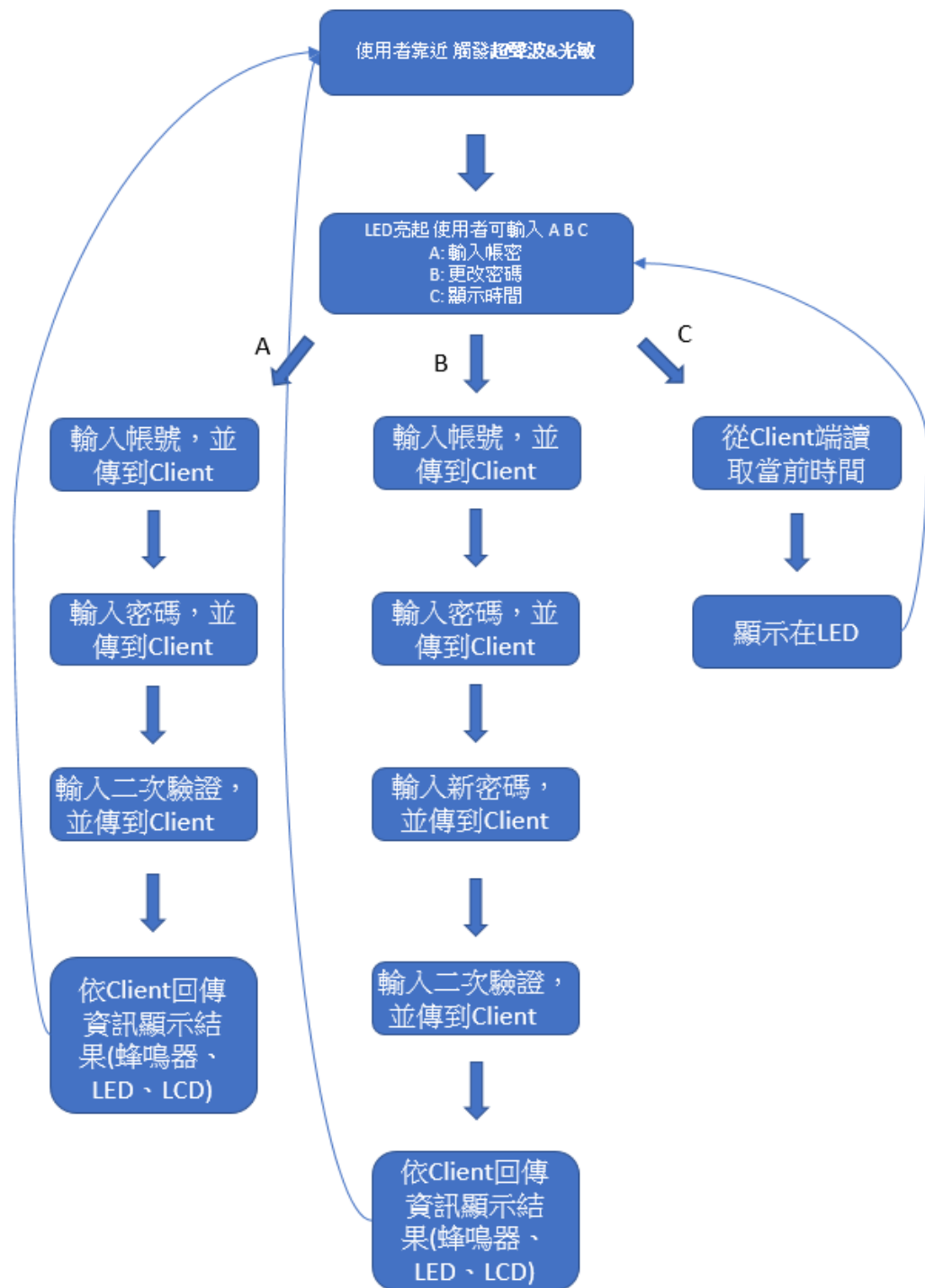
- A. Gyroscopic: 光敏 感測使用者
- B. Acceleration: 超聲波 感測使用者距離
- C. Touch: Keypad 使用者可用 keypad 操作整個介面

5. FreeRTOS

流程圖

本系統共有三大功能，因此我使用 FreeRTOS，建立了三個 TASK，分別為：

- Normal: 監控是否有人靠近，並顯示主選單，主選單部分會顯示各按鍵的功用(A:輸入帳密、B:更改密碼、C:顯示時間)，這邊用 timer 讓 LED 輪流顯示。除外如果使用者按了“顯示時間”，也是在這個 TASK 操作。
- Enter: 主要管理登入的功能
- ChPassword: 主要管理修改密碼的功能



Client 部分

Client 是用 Python 寫的主要是作為 Arduino 和 Server 溝通的橋樑，運用到兩 lib：pyserial & requests

Server 部分

Server 是用 Python 和 Flask 所寫，與 Client 是用 http 協定溝通，傳輸的格式為 json。

主要提供以下功能

- / 根目錄: 註冊帳號畫面
- /register: 輸入註冊資訊
- /twofactor: 註冊後顯示 QR code 供使用者掃描
- /show: 顯示登入 log
- /login: 登入驗證資料用頁面(使用者看不到，只供 client 呼叫用)
 - 回傳 0 表示登入失敗
 - 回傳 1 表示登入成功
- /update: 修改密碼
 - 回傳 0 表示修改失敗
 - 回傳 1 表示修改成功

遇到困難

1. Stack: 因為 stack 有限然後我有三個 TASK，每個 task 要做的事很多，所分配的 stack 大小不能太大也不能太小，在實作時，太大或太小的 bug 並不會顯示錯誤，因此花了很多時間才知道是這個問題。
2. Priority 不足:FreeRTOS 預設只有四種 priority(0, 1, 2, 3)，因為我每個 task 都不能同時做，在實作時需要不停調整自己與其他的 task 的 priority，調整順序錯的話，就會出現難以發現的 bug。