

# 嵌入式系統實驗 final project

題目：Impromptu 鋼琴手套

成員：B09901008 電機三蕭千鈞

Github URL: <https://github.com/Hsiao-Chien-Ti/Impromptu>

## 一、動機

我常會在無聊的時候忍不住玩弄自己的手指，到處敲來敲去，常常會被朋友或家人嫌吵。在發想題目的時候便決定以此做題，要吵就吵得更放肆更有創意一點。

Play store 裡已經有不少鋼琴 app，但大多都有鍵盤狹窄難按的問題，因此我決定結合鋼琴功能以及喜歡亂敲的手指，直接把鋼琴結合在手套上，製作一款隨處可彈的鋼琴手套，為生活增添樂趣。

## 二、功能介紹

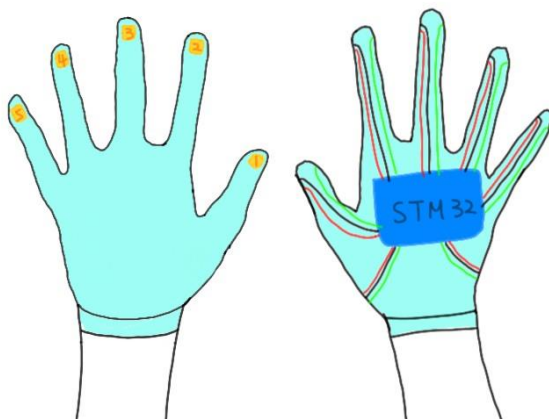
1. 隨處可彈的鋼琴手套：這份作品最大的目的即是為生活增添樂趣，隨處可彈的特性能讓使用者隨時隨地都能揮灑創意。



2. 手勢偵測切換音高：將鋼琴直接裝在手上的最大限制即是人只有五根手指，不可能直接代表整台鋼琴的音。這項功能讓使用者在彈琴的過程中，可以利用手勢調整鋼琴音高，進而彈出所有的音。
3. WiFi 連接手機：透過 WiFi 連接手套和手機，將資料傳到手機端，利用手機當作播音器。同時，因為播音器為手機，使用者也可以戴著耳機進行彈奏，在必須保持安靜的環境下也能沉浸在鋼琴世界裡。
4. 手機 APP 控制：手機端設有 APP，讓使用者可以連線及設定，同時 APP 內也附有錄音及樂譜功能，讓使用者體驗更完整。
5. 錄音功能：靈感稍縱即逝，透過錄音功能，可以記錄當前彈奏的旋律。

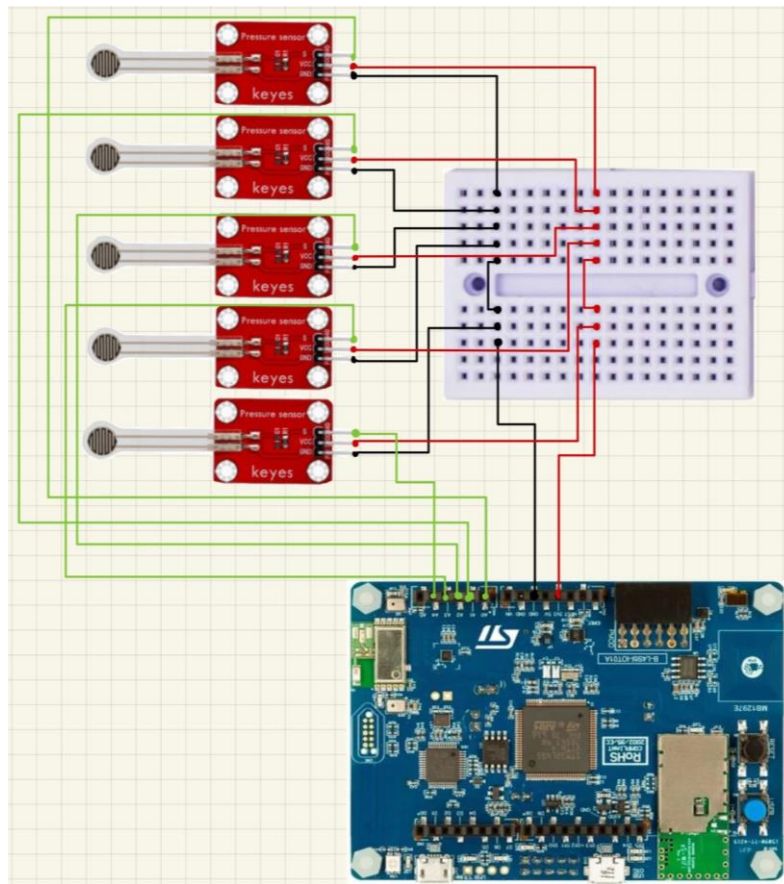
## 三、實作方式

- 設計圖：



- 硬體：

■ 電路圖：



■ STM32-L4S5I

■ Keyes 薄膜壓力感測器：在指尖裝上壓力感測器偵測使用者是否有按壓。

■ 迷你麵包板：擴充腳位

■ 棉質手套

● 軟體：

■ Mbed OS: 手套端使用 STM32 做為控制板，功能及實作方法如下：

◆ Socket Programming: 利用 WiFi Socket 連接手機和手套，由於 STM32 開發板上的網路模組不支援 bind 功能，因此無法當作 server，由手機端擔任 server，手套端擔任 client。使用手機的熱點當作網路，手機端直接連線，手套端則是連接熱點。

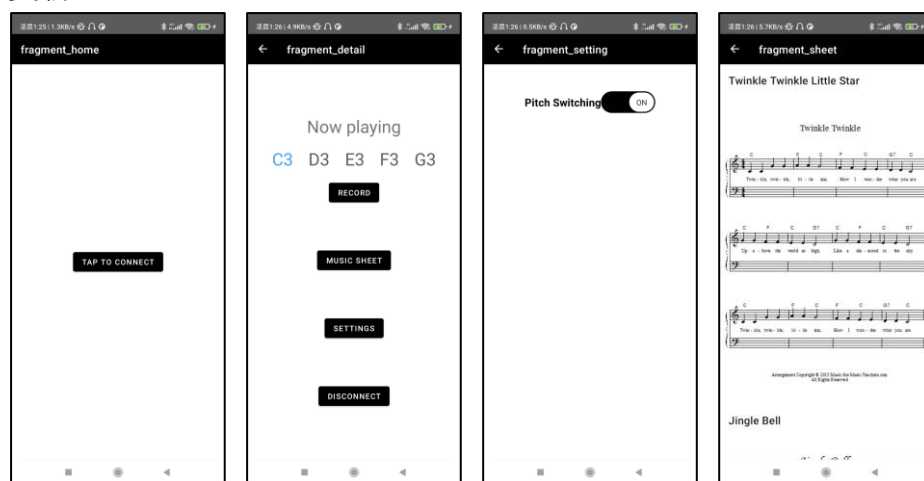
◆ 壓力感測器：利用 AnalogIn 的 read\_u16 讀取 5 個壓力感測器的值，讀進來會是一個介於 0~65535 的整數，設定閾值 14000，超過此閾值則傳送 1 給 server 端，否則傳送 0 給 server 端。測試時有發現相鄰的 AnalogIn 的讀值會互相影響，ex. 按 5 但 4 的值也變稍微變大，因此閾值要設大一點，才不會出現判斷錯誤的情況。

◆ 切換音高偵測：切換音高功能結合了板子上的加速度感測器以及 DSP 濾波處理。利用加速度判斷手套的左移右移，若出現先負再正，即可判斷是向右，反之則向左，向右代表升高音高，向左則是

降低音高。由於快速讀值，讀進來的數值可能會很不穩定，因此將一段時間內的資料儲存起來，利用 DSP 的 fir 將高頻的波濾掉，取樣比較平順，比較不易判斷錯誤。

- Android studio: 手機端使用 Android studio 製作 APP，使用的程式語言是 kotlin。為了達到想要的功能，程式碼內應用了一些上課教過的概念，如 thread 和 volatile。APP 支援的功能及實作方法如下：

- ◆ Fragments: APP 共有 home, detail, setting, sheet 四個介面，每個介面都建立成一個 fragment，並利用 navigation 功能在 fragment 間切換。此外，還建立了另一個名為 SettingVariable 的 object，裡面存了需要在各個 fragment 間共享的變數，這樣就不會在切換 fragment 時失去變數狀態。裡面的變數多存成 Volatile，才可以在 thread 間保持更新。



- ◆ WiFi 連接手套：創立一個名為 SocketHandler 的 object，裡面使用 ServerSocket 建立 server，並透過 accept 和 close 函式接受和關閉來自 client 端的連線。接收完連線之後，建立一個新的 thread，在裡面用 while loop 不斷讀取 wifi 傳過來的資料，儲存下來給播音功能做使用。SocketHandler 是一個 object，因此一次只會存在一個 instance。SocketHandler 中的變數多存成 volatile，因為可能在 main thread 和讀取資料的 thread 間更動值，更動後的值要讓兩個 thread 都看的到。
- ◆ 播放音樂：播放音樂的功能運行在一個獨立的 thread。接收到資料後，根據接收到的資料決定是否要撥放該根手指對應的音。利用 SoundPool 同時播放多個音訊，才能同時彈奏兩根手指的音。音檔來源是自己用網路上的鋼琴 APP 一個音一個音錄製的。在播放聲音的同時，也在 UI 上把正在彈奏的音顯示成藍色。更改 UI 必須在 main thread 中執行，因此利用 runOnUiThread helper function 讓在動作回到 main thread，同時不影響播音的 thread 繼續運作。
- ◆ 錄音功能：利用 MediaRecorder 錄製聲音，錄完的檔案會存在手機

的 `Android/data/com.example.eslab_final_impromptu` 資料夾內。

- ◆ 樂譜功能：利用 `RecyclerView` 建立一個卷軸式畫面，一首歌的譜為一張圖片。
- ◆ 設定：考量到可能使用者只想要彈 5 個音，不想切換音高，可以在設定處關閉此項功能。在關閉此功能後，會把 `SettingVariables` 內的 `pitchSwitching` 設為 `false`，當 `detail fragment` 裡的播音區段程式碼發現 `pitchSwitching` 為 `false` 時，會停止處理 `data` 的最後一位，不去理會音高切換。

#### 四、成果 Demo

<https://youtu.be/yZZKF2OYE3M>

#### 五、參考資料

1. Android studio: <https://developer.android.com/courses/android-basics-kotlin/course?authuser=3>
2. MbedOS: hw2, hw7
3. Previous work: <https://www.amazon.com/Electric-Piano-Gloves-Playable-Interactive/dp/B00PTMORKS>