

微處理機 Lab6 Report

一、 實驗名稱：STM32 Keypad Scanning

二、 實驗目的：

- 了解 STM32 使用原理。
- 了解如何使用 C code 控制 STM32。
- 設計 7-Seg LED 和 keypad 程式。

三、 實驗步驟：

1. Max7219 displayer。
2. Keypad Scanning。
3. 處理單或多按鍵。
4. 設計簡易計算機。

四、 實驗結果與分析：

1. Max7219 displayer：

這題只要把上一次的初始化跟 MAX7219 send 轉成可以讓 C 去呼叫該函數即可，一開始遇到的問題是 C 檔的 S 檔的檔名不能一樣，這個問題困擾我們很久，其他方面都還蠻順利的。

2. Keypad Scanning：

這題一開始遇到的問題是當我按下某些按鍵時，MAX7219 並沒有顯示正確的值，經過 debug 發現可能是掃描的時候出了問題，原本會造成錯誤的寫法是當要掃描某一條 column 時，我們才把上一個 column 的輸出清為 0，接著把該 column 的輸

出設為 1，然後才開始掃描；當把寫法換成先把該 column 的輸出設為 1，拿到 input 的值後直接把該 column 的輸出改為 0，這樣的寫法就不會造成 MAX7219 顯示錯誤數值的問題了，推測可能錯誤的原因是更改 output 的時間間隔不夠久，導至拿到錯誤的 input 而使 MAX7219 沒有顯示正確。

3. 處理單或多按鍵：

這題是我們覺得最難做的一題，原因就在於題目看的不是很懂，經過助教的解釋才豁然開朗原來並不難，這題處理多按鍵的方法就是用有限狀態機，我們分別給每個按鈕三個狀態：沒按下，剛按下，按下很久仍沒有放開，以無窮迴圈的方式去掃描 keypad 並更新其狀態，當掃完一輪後，用目前 state 的資料去更新要加多少值進去並呼叫函式使其顯示在 MAX7219 上。

4. 設計簡易計算機：

計算機這題的程式碼其實並不難，難的部分在於這個計算機該如何運作，靠的是兩個 stack 的幫忙來運作的，分別存放運算元和運算子，當遇到新的運算子時，先檢查運算子 stack top 的運算子的優先級是否小於等於目前新的運算子，如果是，則從運算元 stack pop 兩個元素出來做運算後存回運算元 stack，接著再重複一樣的運算子檢查直到運算子 stack 為空或是沒有比新的運算子優先級低的運算子，這是再將運算子 push 到 stack 中；而當遇到等號時，則依序上述動作直到運算子 stack 為空，此時運算元 stack 也只會剩一個元素，該元素即是輸入算式的結果。

五、心得討論與應用聯想：

這次的 LAB，最大的問題是在要如何使用 C 語言來開啟 GPIO，因為需要 include 許多神奇的 structure，在做完第一題能夠正確的找到 KEY 值後，第二跟第三題就很好解決，第二題只要寫狀態機來確認誰要累加，第三題就是把大一寫過的計算機再拿出來用而已。