課程：DCP3117 Microprocessor System Lab 授課教師：曹孝櫟教授 2019

*NCTU CS* 國立交通大學資訊工程學系

**Lab4 7-Seg LED**

**1. Lab objectives 實驗目的**

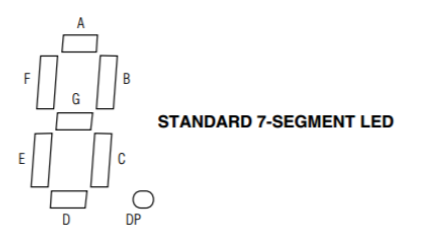
● Understand the principle of using MAX7219.

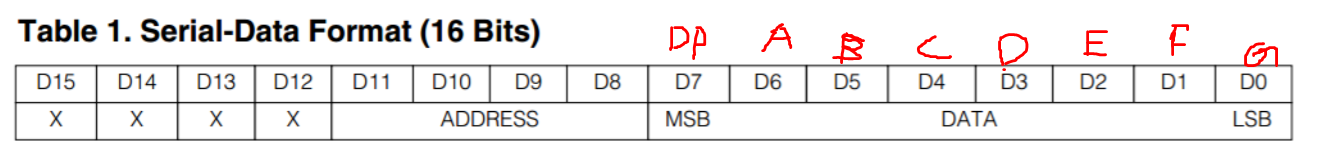
● Design the program of 7-Seg LED.

● 了解 MAX7219 使用原理

● 設計 7-Seg LED 程式

**2. Lab theory 實驗原理**





Str進去Max7219( pa5=Din, pa6=LS, pa7=CLK )的data格式如上：

D7~D0是塞進去的data，D11~D8是拿來定址的，例如：

* 0x01,…,0x09是第1,…,第8顆7-seg
* 0x09: Decode，D<x>是1就代表第<x+1>顆7-seg是Decode Mode。
* 在Decode Mode下只看D3~D0的值，只有16種顯示可用(0~9,-,e,h,l,p,<blank>)，

不在Decode Mode時比較自由，你可以透過定址 + data，來決定哪一顆的哪一段LED亮。

* 例如Decode Mode時把0x037b (=0000 0011 0111 1011)透過send function Str進去pa5，可以讓第4顆7-seg亮字母E。
* 不在Decode Mode時把0x037b (=0000 0011 0111 1011)透過send function Str進去pa5，可以讓第4顆7-seg亮數字9。
* 0x0A: Intensity，亮度0~15(所以只有D3~D0的值有用)
* 0x0B: Scan-Limit顯示幾個7-seg (1~8，所以只有D2~D0的值有用)
* 0x0C: Shutdown，只有D0值有用。1為正常，0為Shutdown mode 所有7-seg LED會關掉，是一種省電模式
* 0x0F: Display Test，只有D0值有用，0為正常，1為Test mode，會讓所有燈全亮
* Send function 要模擬clk，把data一步步的傳進去Max7219

見lab4\_note 課程講義。

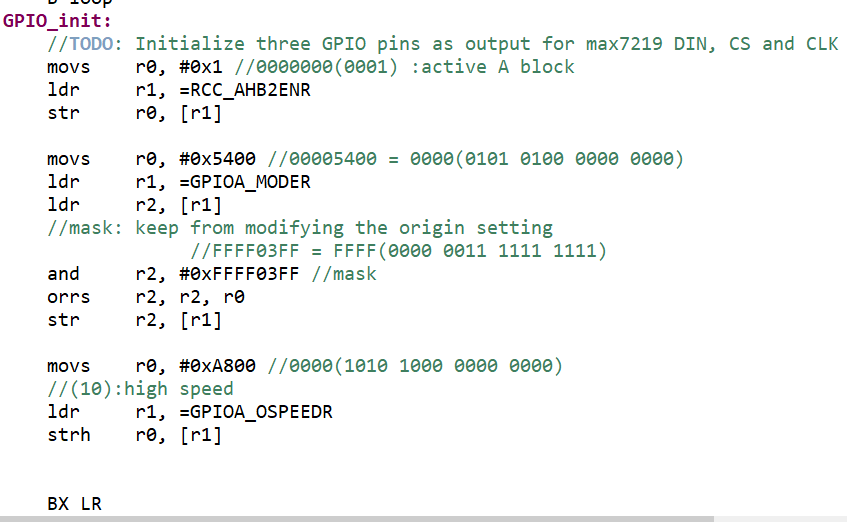
**3. Steps 實驗步驟**

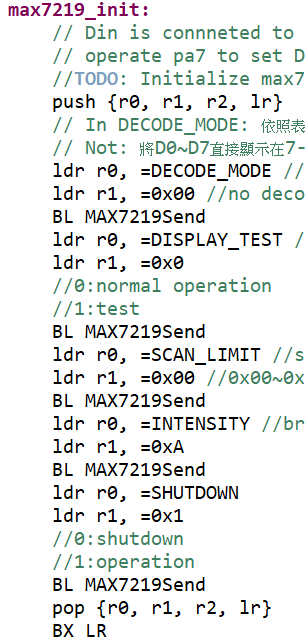
**3.1. Lab4.1: Max7219 與 7-Seg LED練習 without decode mode**

將 stm32 的 3.3V 接到 7-Seg LED 板的 VCC，GND 接到 GND，並選擇三GPIO 接腳分別接到 DIN、CS 和 CLK。並利用 GPIO 控制 Max7219 並在 7-Seg LED 上的第一位依序顯示 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, b, C, d, E, F (時間間隔1秒)，。

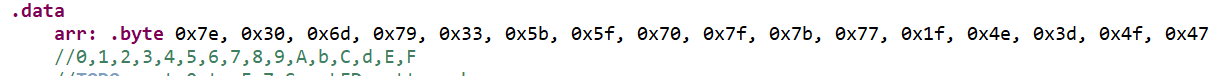
了解上面如何運作後就簡單了，只要設定都對就沒問題了。

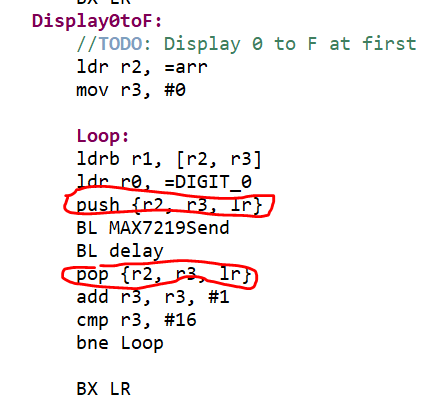
* 把GPIO開好(這次用pa5, pa6, pa7)





* 把Max7219設定好
  + DECODE：00，全部不decode
  + DISPLAY：0，不test
  + SCAN\_LIMIT：0，第一題只要1位，顯示不必要的會因為殘留的data導致顯示亂碼
  + INTENSITY：10，隨便一個值
  + SHUNDOWN：0，不shutdown



* 基本上就是把已經寫好的值一個個塞好
* 注意在BL過的函式內跳別的BL時，一定要先把以後需要的值(特別是lr) push 進stack，之後回來再pop。

**3.2 Lab4.2: Max7219 與 7-Seg LED 練習**

(**Practice of Max7219 and 7-Seg LED) — use code B decode mode (30%)**

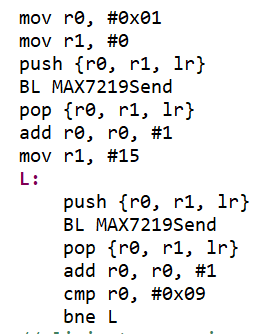
* 要以decode B mode顯示學號，得將每個bit開啟decode B mode，做法是在send decode mode的address時，data設為FF(0b11111111)，除此之外，為了要同時印出7個數字，要將scan limit設為0x6。
* 將學號以連續的方式存在data區段，當讀入後直接以其值當作data，配合要印的address送至Max7219，執行七次後表示七個數字皆完成。

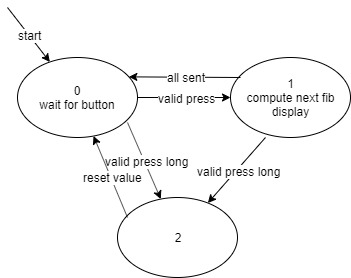
**3-3設計一程式偵測實驗板上的 User button，當 User button 按 N 次時，7-Seg LED 上會顯示 fib(N) 的值。User button 長按 1 秒則將數值歸零。**

**fib(0) = 0、fib(1) = 1、fib(2) = 1 、…**

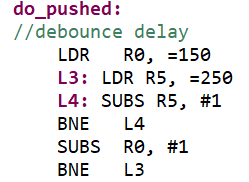
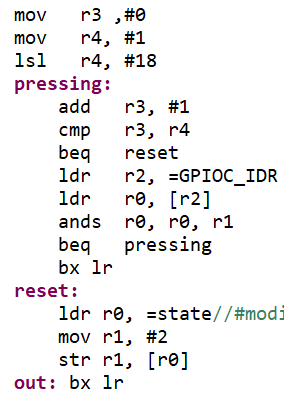
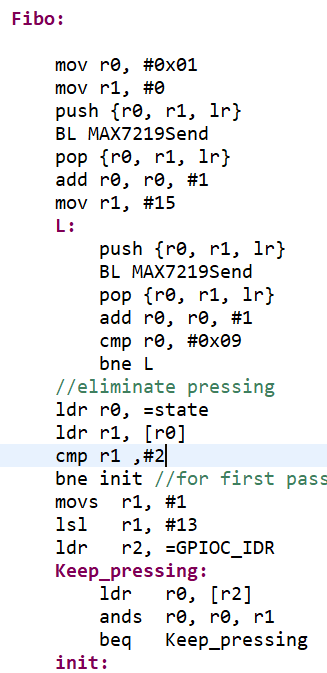
**若 fib(N) ≥ 100000000 則顯示-1。(超過8位)**

* State:0:等按鈕；1:算下一個數字：2:重置





* 一開始先開好SCAN\_LIMIT=7，並顯示空白跟初始值。如果一開始SCAN\_LIMIT=0還要一直回去調，很麻煩。
* Debounce長度用下圖設定感覺效果挺不錯，不會有按鈕遲鈍的問題。



長按的功能：

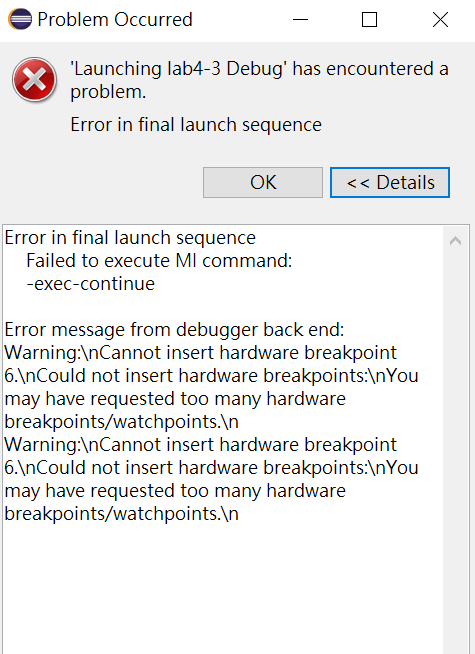
* 把之前過濾後續按按鈕的程式稍微修改，只要一直按的話就會一直累加r3，當r3超過一定值之後就會跳離，並修改state狀態。
* 因為長按視為重置，所以把後續過濾長按的功能移到最前面，如果state=2代表是從init跳來的，所以要執行過濾長按的程式部分。

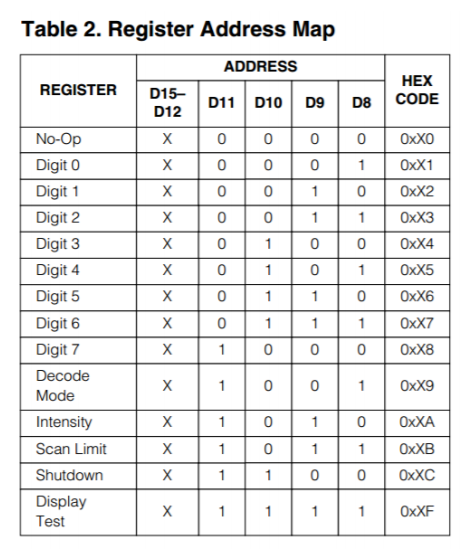
其他要注意的：

要用loop把要改的digit一個個給Max7219。

**4. Feedback 實驗心得或建議**

* Too many hardware breakpoints

這次漸漸發現設斷點很好用，即使是斷點的狀態，按按鈕也是有反應的。不過要注意斷點數有上限。



* 給出的位址超出01~08範圍，發生全亮，猜測可能是不小心塞給了test mode(D11~D8=0x0F) 的D0 = 1，因為當時寫的是loop address 0~15去塞，要注意address 0x00是non op，address 0x01~ 0x09才是7-seg的位址，
* 以及address offset和decode mode不要搞混，一個是address，一個是data。

