

一、【實驗目的】：

What was your design? What were the concepts you have used for your design?

第一題

首先我改善之前判斷按鈕連點的邏輯，用 goto 來改善按鈕點下去瞬間的閃爍，接下來我將 + - * / 這四個 define 成字串，方便我可以用 for 遍歷出來並且存入 result 陣列中。

第二題

為了讓七段顯示器不閃爍，已經利用 loop_count 去解決，而蜂鳴器也是同理，只要 mod 想要間隔的 ms 就可以。

二、【遭遇的問題】：

What problems you faced during design and implementation?

七段顯示器的閃爍問題

Lcd 更新問題

三、【解決方法】：

How did you solve the problems?

七段顯示器的閃爍問題

如果每個七段顯示器更新時，每次迴圈都小於 5 ms 那麼就不會有閃爍的感覺，經過一些測試，我一個 loop_count 去當作計數器，並把迴圈每次 delay 1 ms。

當 loop_count % 5000 就將七段顯示器更新，為了知道現在更新到第幾顆還需要 seven_segment_index 來記錄現在要更新第幾顆，不過有時候現在剛好要更新的七段顯示器是不用顯示的，例如 16，這時候我會用一個 while 去判斷是否為不需要顯示的，如果不需要顯示就把 seven_segment_index++，來跳到下個更新。

如此一來就不會閃爍了。

Lcd 更新問題

使用 lcd_x lcd_y 將迴圈限制每次只更新一個字元，並且用兩個 buffer 去記錄當下 lcd 狀態跟下次 lcd 的狀態，兩個陣列去比較，如果不相同再更新即可，這樣可以大幅減少重複更新的消耗資源，不過主要是 printLine, printS 實在太卡，只要用這兩個 function 就會遇到七段顯示器的閃爍，也就是單次迴圈超過了 5 ms，所以才使用逐步更新的作法，且肉眼看不太出來。

四、【未能解決的問題】：

Was there any problem that you were unable to solve? Why was it unsolvable?

沒有

五、【投影片的問題】：

1. 為甚麼 MCU_init.h 要加入 Define MCU Interfaces?

```
#define MCU_INTERFACE_SPI3
#define SPI3_CLOCK_SOURCE_HCLK // HCLK, PLL
#define PIN_SPI3_SS0_PD8
#define PIN_SPI3_SCLK_PD9
#define PIN_SPI3_MISO0_PD10
#define PIN_SPI3_MOSI0_PD11
```

Trace code 後發現，在 SYS_init.c 裡面用 ifdef ，去判斷是否定義腳位來決定 lcd 是否工作。

PD8, PD9, PD10, PD11 是 SPI 訊號，

SPI3_CLOCK_SOURCE_HCLK 是用來選擇 CLOCK 模式
CLK_SetModuleClock(SPI3_MODULE, CLK_CLKSEL1_SPI3_S_HCLK, 0);

2. 為甚麼以下 code Text 這個陣列使用 Print_Line 印出時在第 10 格的 Hello 沒有印出來？

```
sprintf(Text, "%d&%d = %d", 1, 0, 0); // print scankey input to
sprintf(Text+10, "Hello"); // print scankey input to string
print_Line(1, Text); // display string on LCD
```

因為第一行 sprint 只有 8 個字元，所以除了 0~7 有字元外剩下後面都填入 \0 ，而當第二行直接從第 10 位開始接上 Hello 那麼就會出現兩個字串中間有 \0 ，但是 print_line 遇到 \0 會以為字串已經結束了，所以不會印出後面 Hello。

所以如果將中間用 Text[8]=' ', Text[9]=' ', 填入空白就可以印出了。

如以下

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	&	&	0		=		0	\0	\0	\0	\0	\0	\0	\0	\0
	1	&	&	0		=		0	\0	\0	H	e	l	l	o	\0