

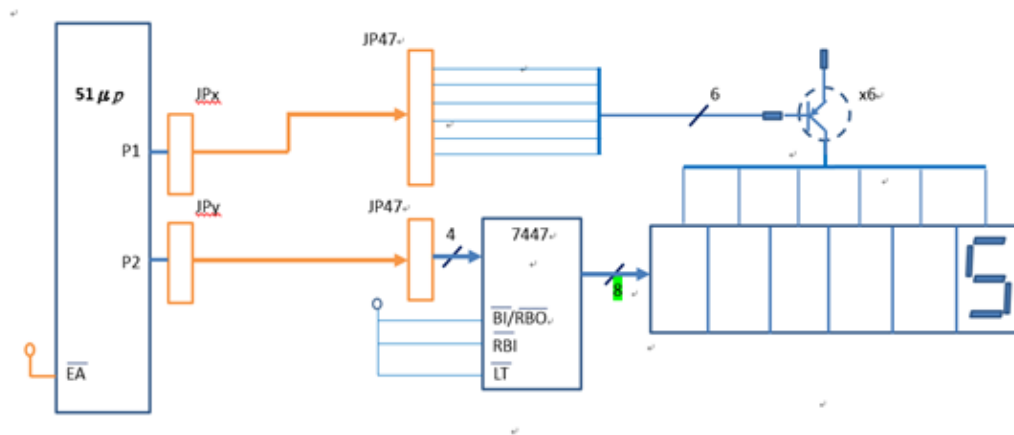
微處理機 Lab4 結報

0413335 郭逸琳 0416039 李佳燕

實驗主題與訓練目的

此次實驗主要是先透過接線了解七段式 LED 的運作方式，再進一步實驗能不能只用線路不用程式就能接出想要的結果。

實驗電路



實驗過程

先在 uvision 裡貼上下列的 code：

```
org      0
        mov     SP, #50H
start:   mov     R7, #6
next1:
        mov     R5, #250
next11:
        mov     R6, #6
        mov     R1, #0FEH
        mov     R2, #0
next12:
        mov     A, R1
        mov     P1, A
        RL      A
        mov     R1, A
```

```

        mov     A, R2
        inc     R2
        mov     P2, A
        call    delay1      ; ===KKK===
        djnz    R6, next12
        djnz    R5, next11
        mov     P1, #0FFH
        call    delay2      ; ===LLL===
        djnz    R7, next1
        mov     R7, #6
next2:
        mov     P1, #0F8H
        mov     P2, #5
        call    delay2      ; ===III===
        mov     P1, #0C7H
        mov     P2, #1
        call    delay2      ; ===JJJ===
        djnz    R7, next2
        jmp     start
delay1:
        push    1
        mov     R1, #200
        djnz    R1, $
        pop     1
        ret
delay2:
        ; appx. 0.5sec delay, why?
        push    1
        push    2
        push    3
        mov     R1, #100
dd22:   mov     R2, #250
dd21:   mov     R3, #10
        djnz    R3, $
        djnz    R2, dd21
        djnz    R1, dd22
        pop     3
        pop     2
        pop     1

```

```
ret  
end
```

然後再利用以下方式接線：

P1 接 8bit 的線(我們那組板子沒有只有 7bit 的線，但其他組好像有)，線的另外一頭接到七段式 LED 的下排，P2 接 4bit 的線，線的另外一頭接到七段式 LED 的上排。

最後將程式碼燒到板子裡再執行就可看到結果。

解釋程式碼

P1 會是 LED 亮的位置, P2 則是要亮的數字

先給 P1 0FEh(第一個 LED 亮其他暗)，P2 給 0(亮數字 0)，P1 做 left rotate 的同時 P2+1(亮的數字+1 且改成隔壁的 LED 亮)，然後每次做完後要 call 一次 delay，重複此行為六次後把 P1 改為 FFH(全暗)並 call delay(這次 delay 的時間比較長)。這一大段的行為總共會重複 6 次。然後進入第二階段，先給 P1 0F8H(亮右半邊)，P2 給 5(亮數字 5)，亮一次並 call delay 後將 P1 改成 0C7H(亮左半邊)，將 P2 改成 1(亮數字 1)，亮完後 call delay 並重複此行為 6 次。

實驗結果

這次作業有兩個 pattern：

第一個 pattern 是一次只亮一個跑 6*250 次(從第一個 LED 到最後一個 LED 再重頭)再全暗，整個行為跑六次。但因為跑很快且全暗的 delay 比較久，所以整體看起來不會是一次亮一個的，會是全部都亮的，而且會閃爍。

第二個 pattern 是右半邊先亮數字 5，接下來左半邊亮數字 1，一直輪流交替。

實驗延伸

Manually wire-up the circuit so that all 7-seg digits display "0", without code-driving. Rewire the circuit so that "0" appears on digit-0, and again without code-driving. Explain the difference in the display intensity perceived in the two cases.

不更改 Code，僅僅用接線的方式，達到上述的要求

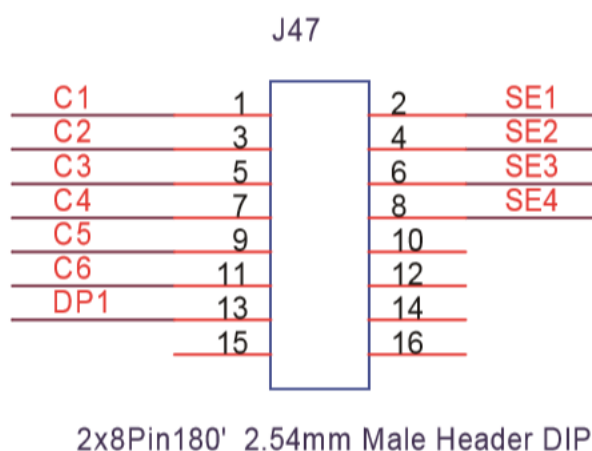
7 段顯示器上排用了 4 個 bit(控制顯示的數字)，下排用 7 個 bit(6 個小塊+1 個點)(控制顯示的位置)，所以下排要接在控制開關的旁邊，就是 DIP SW1~8，這樣就可以透過開關來選擇要讓哪個亮，而上排就要接 0000 給它(可以透過 P1 或

P2 中一直是 0 或是其他 pin 預設是 0 的位置(J28 的 LOW)接給它)。

透過實驗我們發現只有在開關那排右邊的其中一排的 pin(上面有寫 L 的)預設是 0 其他所有 pin 預設都是 1，包括七段式 LED 上的 pin，所以如果 LED 中的四個 bit 中要設為 1 的 bit 可以不用接線，只要把要設為 0 的 bit 接線就可以了。例如要顯示 3(00000011)的話，因為只取後面 4 個位數，所以只看 0011，這時候 1 的訊號不用接，因為 LED 的預設就是 1，但 0 一定要接，不然會出錯。

實驗模組

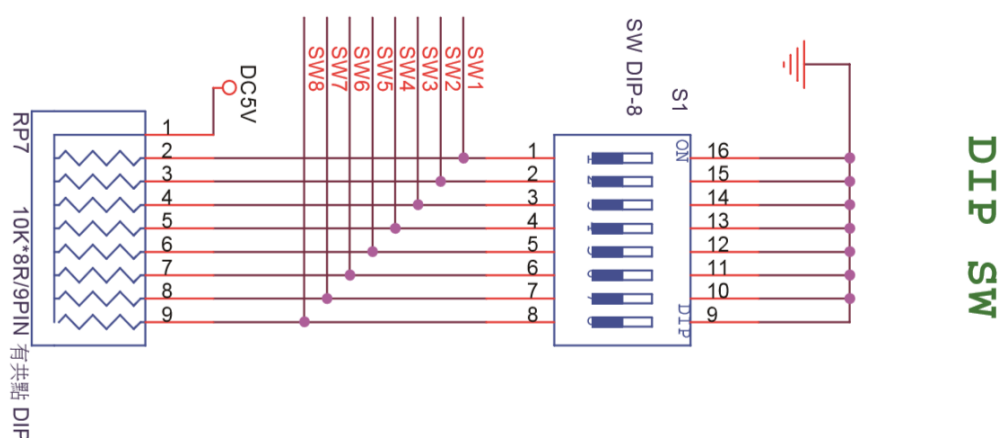
J47：



左邊的就是板子上 下排的 7 個 bit (C1~6 控制要亮的區塊; DP1 控制要不要亮小數點)，bit 為 0 表示要亮，bit 為 1 表示不要亮。

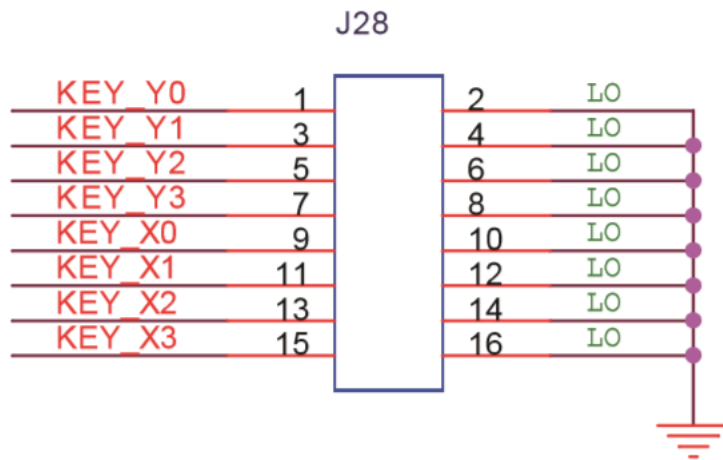
右邊的就是板子上 上排的 4 個 bit，控制要顯示的數字。

DIP SW 1~8：



透過開關可以控制要亮哪一個(6 個區塊或是 1 個點，第八個開關不會用到)

J28 的 LOW :



透過名字可以看出來，他的預設值是 0，可用來控制要顯示的數字。

實驗心得

原來單純靠用板子接線，就可以讓這個 7-segment LED 顯示出想要的數字，以及要亮哪個(共有 6 個小塊，加上一個點)。

然後我們還發現，因為 7 段 LED 的下排接在開關那邊(DIP SW1~8)，我們發現如果讓越多個開關打開，平均的亮度就會減低，因為同樣的電流會分散出去，同理，只有一個開關打開的時候是最亮的。

透過這次實驗，我們比之前更加了解板子及各個 pin 的運作方式，更能體會到微處理機的樂趣所在。