



微算機實驗報告

Lab #10

姓名：楊哲睿

系級：電機 10

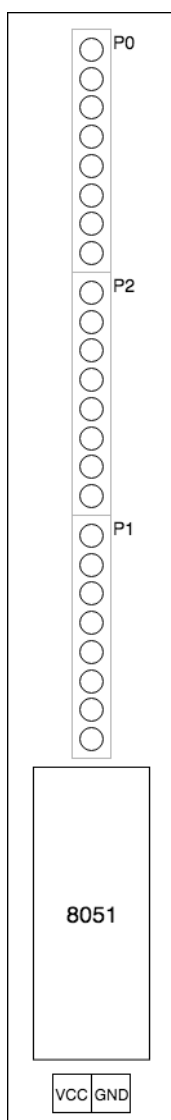
學號：0610780

上課時間：4EF、4IJ

一、實驗目的：

- 搖搖棒的顯示原理。
- 水銀開關的觸發原理。
- 大型文字 Table 的讀取方式。
- 外部中斷的原理。

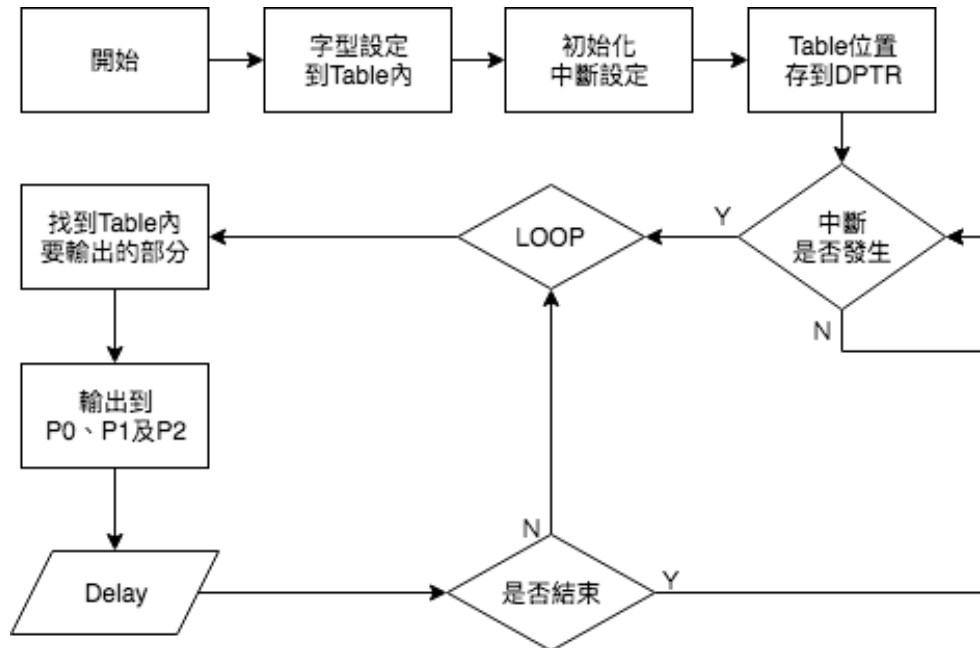
二、硬體架構：



三、程式流程圖：

- 基本題

用搖搖棒來顯示自己的學號，一次顯示一個數字，此外水銀開關的判斷機制必須以外部中斷的方法來實現。



四、問題與討論：

- 這兩次搖搖棒實驗，分別使用外部中斷以及 Polling 的方式來判別水銀開關的狀態，這兩種方法各有何優缺點？

外部中斷的優勢在於，當使用外部中斷時，CPU 不用浪費大量時間在不斷檢查某旗標的狀態是否改變，只要中斷發生，可立即取得服務，故執行上較有效率。Lab6 中的 Polling 是透過不斷檢查的方式來查看某旗標狀態是否改變，需要耗費大量的時間在這些檢查上。

至於 Polling 的優點，對我來說應該是程式較為直觀且容易理解，外部中斷的初始化設定相當複雜，必須要先設置好許多旗標來開啟中斷的功能，且支援中斷的 CPU 也需要有額外的硬體設計。

五、程式碼與註解：

- 基本題

用搖搖棒來顯示自己的學號，一次顯示一個數字，此外水銀開關的判斷機制必須以外部中斷的方法來實現。

```
ORG 0000H
JMP INITIAL
ORG 0003H
```

```

JMP INTERRUPT
ORG 0050H

INITIAL:
    SETB IT0          ;edge trigger
    SETB PX0          ;INT0 interrupt high priority
    SETB EX0          ;將外部中斷致能旗標設為 1
    CLR IE0           ;將 IE0 旗標設為 0
    SETB EA           ;開啟所有中斷功能
    MOV R4, #0         ;R4 為來回甩動的次數(來回一次會加 2)
    MOV DPTR, #TABLE0 ;將 DPTR 設為第一個字 0

START:
    JMP START          ;如果中斷沒有發生就在這裡無限迴圈

INTERRUPT:
NUM1:
    CJNE R4, #8, NUM2 ;R4 不等於 8 就跳至 NUM2
    MOV DPTR, #TABLE6 ;將 DPTR 設為下一個字 6
    INC R4             ;搖一次(左右來回)會加二
    JMP LED_SHOW       ;顯示
NUM2:
    CJNE R4, #16, NUM3;R4 不等於 16 就跳至 NUM3
    MOV DPTR, #TABLE1 ;將 DPTR 設為下一個字 1
    INC R4             ;搖一次(左右來回)會加二
    JMP LED_SHOW       ;顯示
NUM3:
    CJNE R4, #24, NUM4;R4 不等於 24 就跳至 NUM4
    MOV DPTR, #TABLE0 ;將 DPTR 設為下一個字 0
    INC R4             ;搖一次(左右來回)會加二
    JMP LED_SHOW       ;顯示
NUM4:
    CJNE R4, #32, NUM5;R4 不等於 32 就跳至 NUM5
    MOV DPTR, #TABLE7 ;將 DPTR 設為下一個字 7
    INC R4             ;搖一次(左右來回)會加二
    JMP LED_SHOW       ;顯示
NUM5:
    CJNE R4, #40, NUM6;R4 不等於 40 就跳至 NUM6
    MOV DPTR, #TABLE8 ;將 DPTR 設為下一個字 8

```

```

        INC R4                ;搖一次(左右來回)會加二
        JMP LED_SHOW         ;顯示
NUM6:
        CJNE R4, #48, FIN    ;R4 不等於 48 就跳至 FIN
        MOV DPTR, #TABLE0    ;將 DPTR 設為下一個字 0
        MOV R4, #0           ;搖一次(左右來回)會加二
        JMP LED_SHOW         ;顯示
FIN:
        INC R4
LED_SHOW:
        MOV R0, #72
        MOV R1, #24
NEXT_COLUMN:
        CALL READ_BYTE
        MOV P0, A
        CALL READ_BYTE
        MOV P2, A
        CALL READ_BYTE
        MOV P1, A
        CALL DELAY
        DJNZ R1, NEXT_COLUMN
        RETI

;====READ 1 BYTE FROM TABLE====
READ_BYTE:
        DEC R0
        MOV A, R0
        MOVC A, @A+DPTR
        CPL A
        RET

DELAY:
        MOV R5, #20H
DELAY1:
        MOV R6, #25H
DELAY2:
        MOV R7, #05H
DELAY3:

```

```
DJNZ R7, DELAY3
DJNZ R6, DELAY2
DJNZ R5, DELAY1
RET
```

TABLE0:
(略)

TABLE6:
(略)

TABLE1:
(略)

TABLE7:
(略)

TABLE8:
(略)

END

六、心得：

這次的實驗其實跟 Lab6 很像，只是判斷有沒有搖的機制從 Polling 變成外部中斷而已。外部中斷的初始化設定有點複雜，各種旗標跟 Enable，讓我花了一些時間才搞懂，但理解了之後發現，其實善用中斷這個功能，能讓程式更有效率也更精簡，很多判斷都不用自己寫得落落長，微控器會自己幫你完成。而且聽同學說，中斷是期末考的重點之一，看來真的是要好好學習呢。