**微算機實驗報告**

姓名：滕瑋安(中三)

學號：0510743

系級：電機09

上課日期：12/12

Lab #10

**ㄧ、實驗目的：**

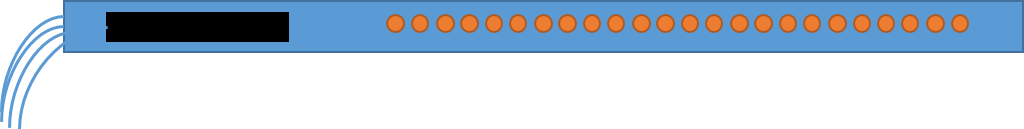
1.搖搖棒的顯示原理。

2.水銀開關的觸發原理。

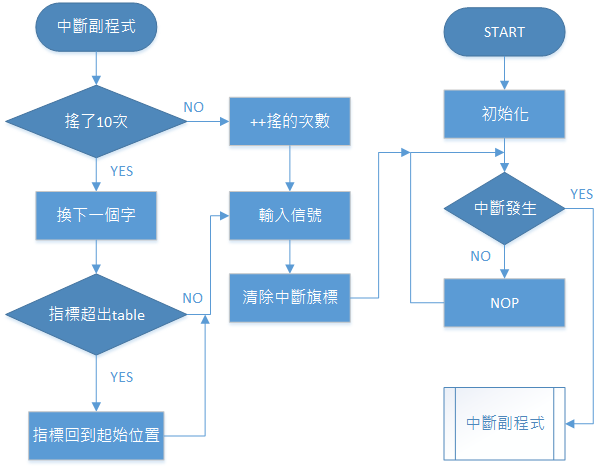
3.大型文字Table的讀取方式。

4.外部中斷的原理。

**二、硬體架構：**



**三、程式流程圖：**



**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?**

**ANS:** 外部中斷在程式架構上，比Polling的方式判定更容易設計，不必擔心因鍵入副程式而無法循環的問題，不過要考慮的是：是否允許中斷中亦有中斷發生，否則可能會出現不在預期中的結果。雖然理論是這麼說，我還是覺得搖搖棒的Polling比較簡單。

**五、程式碼與註解：**

**第一題：**

**ORG 00H**

**JMP INITIAL**

**ORG 03H ;中斷向量**

**JMP INTERRUPT**

**ORG 50H**

**INITIAL:**

**SETB IT0 ;選擇falling edge trigger**

**MOV IE,#10000001B ;開啟EA、EX0**

**MOV IP,#00000001B ;開啟PX0**

**CLR IE0 ;清除中斷旗標**

**MOV R0,#0FFH ;表一開始的位置**

**MOV R1,#0 ;搖的次數**

**MOV R2,#0 ;指標所在行數**

**MOV R3,#12 ;讀一個數字所要的行數**

**MOV DPTR,#ID ;初始化指標位址**

**MAIN:**

**JMP MAIN ;無窮迴圈**

**INTERRUPT:**

**CJNE R1,#30,ADDL ;若搖的次數還沒10次，跳到ADDL**

**MOV R1,#0 ;若達到10次，計數器歸零**

**MOV A,#36**

**ADD A,R2**

**MOV R2,A ;指標所在位址跳12行(一個數字)**

**CJNE R2,#252,NEXT\_COLUMN ;若沒超出TABLE就輸入信號**

**MOV R2,#0H ;超出TABLE就回到起始位址**

**CALL NEXT\_COLUMN ;輸入信號**

**CLR IE0 ;清除中斷旗標**

**RETI**

**ADDL:**

**INC R1 ;搖的次數加一**

**CALL NEXT\_COLUMN ;輸入信號**

**CLR IE0 ;清除中斷旗標**

**RETI**

**NEXT\_COLUMN:**

**CALL READ\_BYTE ;讀進一個BYTE**

**MOV P1,A ;把值放進P0**

**CALL READ\_BYTE ;讀進一個BYTE**

**MOV P2,A ;把值放進P2**

**CALL READ\_BYTE ;讀進一個BYTE**

**MOV P0,A ;把值放進P1**

**CALL DELAY**

**DJNZ R3,NEXT\_COLUMN ;換行**

**MOV R3,#12**

**MOV R0,#0FFH**

**RETI**

**;=====READ 1 BYTE FROM TABLE=======================**

**READ\_BYTE: ;讀一個BYTE**

**INC R0 ;R0加一**

**MOV A,R0 ;把R0放進A**

**ADD A,R2**

**MOVC A,@A+DPTR ;對應到TABLE並把值放進A**

**CPL A ;把A做反向**

**RET**

**DELAY:**

**MOV R6,#050H**

**DELAY1:**

**MOV R7,#0AFH**

**DELAY2:**

**DJNZ R7,DELAY2**

**DJNZ R6,DELAY1**

**RET**

**ID:**

**(12\*3\*7的table)**

**六、心得：**

生平第一次接觸外部中斷，覺得這東西真的是一個謎。就算讀了書，照著範例打，也不一定知道自己到底打了什麼(\*ﾉωﾉ)把自己的code改到最後變得跟同學的87%像也還是有BUG，真的不知道我這輩子跟微算機結了什麼仇。不過我秉持著駑馬十駕，一回生，二回熟的精神，硬是打了很多次，你以為我懂了嗎?我還是不懂自己到底在打什麼。但是我相信經由時間的淬鍊，可能有一天起床就會知道外部中斷的意義與發明人當初的用意所在，因此我每天睡前都會思考一次，相信明天，我就會懂外部中斷的真諦了吧？