**微算機實驗報告**

**姓名：楊哲睿**

**系級：電機10**

**學號：0610780**

**上課時間：4EF、4IJ**

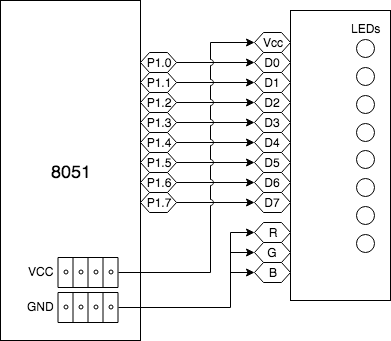


Lab #1

**ㄧ、實驗目的：**

* 瞭解8051的開發環境及開發的流程，包含參數的設置，組合語言的使用，以及燒錄到晶片的流程等。
* 暸解I/O Port的使用方式，並透過控制輸出Port之high-low，讓LED燈號作出跑馬燈和霹靂燈的效果。

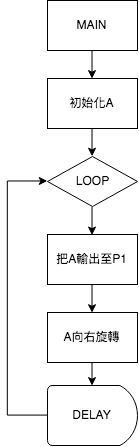
**二、硬體架構：**



**三、程式流程圖：**

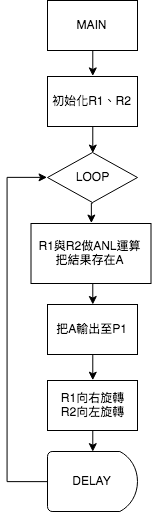
* 基本題

用Port 1控制8個LED燈，使LED由左向右做出跑馬燈效果。



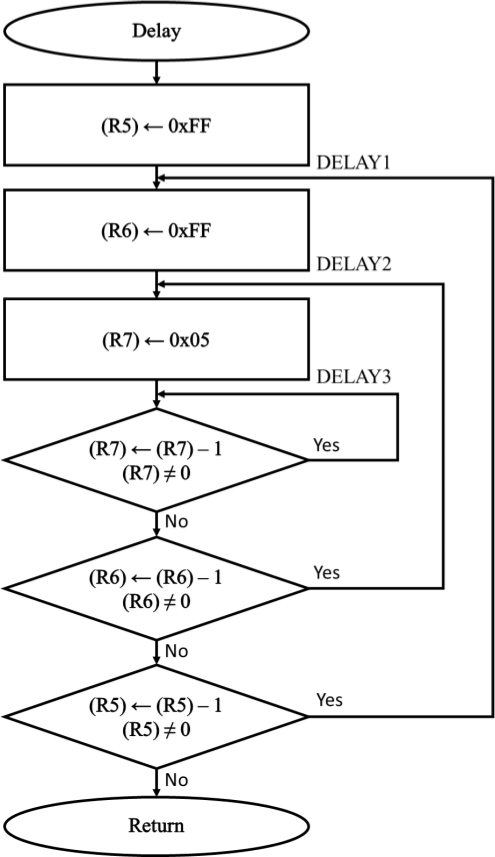
* 進階題

請做出霹靂燈的效果，效果類似有兩組LED同時由兩端開始，交錯後繼續向左以及向右跑。



**四、問題與討論：**

* 若時間隔設定為 0.5 sec，則時間延遲的副程式為何？



根據Delay副程式的流程圖、每個指令所需要的時間（用幾個clock計算），以及處理器的時脈（12MHz）我們可以知道，總延遲時間可以用以下式子表示：

根據計算，共有九種不同的解能讓Delay時間就剛好是0.5秒，計算的程式如下：

#include <stdio.h>

int main() {

for (int i = 0; i < 256; i++) {

for (int j = 0; j < 256; j++) {

for (int k = 0; k < 256; k++) {

int clock\_num = 2+(2+(2+(4\*k)+4)\*j+4)\*i+4;

if(clock\_num == 6000000) {

printf("(R5, R6, R7) = (%d, %d, %d)\n", i, j, k);

printf("(R5, R6, R7) = (%#x, %#x, %#x)\n\n", i, j, k);

break;

}

}

}

}

return 0;

}

輸出如下：

(R5, R6, R7) = (33, 218, 207)

(R5, R6, R7) = (0x21, 0xda, 0xcf)

(R5, R6, R7) = (77, 86, 225)

(R5, R6, R7) = (0x4d, 0x56, 0xe1)

(R5, R6, R7) = (91, 164, 99)

(R5, R6, R7) = (0x5b, 0xa4, 0x63)

(R5, R6, R7) = (99, 60, 251)

(R5, R6, R7) = (0x63, 0x3c, 0xfb)

(R5, R6, R7) = (99, 100, 150)

(R5, R6, R7) = (0x63, 0x64, 0x96)

(R5, R6, R7) = (143, 48, 217)

(R5, R6, R7) = (0x8f, 0x30, 0xd9)

(R5, R6, R7) = (189, 46, 171)

(R5, R6, R7) = (0xbd, 0x2e, 0xab)

(R5, R6, R7) = (189, 138, 56)

(R5, R6, R7) = (0xbd, 0x8a, 0x38)

(R5, R6, R7) = (189, 230, 33)

(R5, R6, R7) = (0xbd, 0xe6, 0x21)

取其中一個解把R7設為0x21（33）、R6設為0xE6（230）、R5設為0xBD（189）。完整Delay副程式如下：

DELAY:

MOV R5, #21H

DELAY1:

MOV R6, #0E6H

DELAY2:

MOV R7, #0BDH

DELAY3:

DJNZ R7, DELAY3

DJNZ R6, DELAY2

DJNZ R5, DELAY1

RET

* JMP以及CALL這兩種指令都能跳躍到指定的程式記憶體位址，其差別在哪？

JMP是單向的，一旦離開了原來的位址就回不來了；CALL是雙向的，透過在副程式裡加入RET指令，就能回到原來的位址。

**五、程式碼與註解：**

* 基本題

用Port 1控制8個LED燈，使LED由左向右做出跑馬燈效果。

ORG 0000H

AJMP MAIN

ORG 0050H

MAIN :

MOV A, #7FH ;顯示最左邊的燈

LOOP :

MOV P1, A ;輸出A的值到P1

RR A ;放到累加器做右旋

ACALL DELAY ;避免視覺暫留讓效果不好，這邊delay 0.141秒

JMP LOOP

DELAY:

MOV R5, #FFH

DELAY1:

MOV R6, #FFH

DELAY2:

MOV R7, #05H ;R5、R6、R7的值可以控制delay的時間長度

DELAY3:

DJNZ R7, DELAY3

DJNZ R6, DELAY2

DJNZ R5, DELAY1

RET

* 進階題

請做出霹靂燈的效果，效果類似有兩組LED同時由兩端開始，交錯後繼續向左以及向右跑。

ORG 0000H

AJMP MAIN

ORG 0050H

MAIN :

MOV R1, #7FH ;顯示最左邊的燈

MOV R2, #0FEH ;顯示最右邊的燈

LOOP :

MOV A, R1

ANL A, R2

MOV P1, A ;對前R1和R2裡的兩個值做AND運算並輸出至Port1

MOV A, R1

RR A ;放到累加器做右旋

MOV R1, A

MOV A, R2

RL A ;放到累加器做左旋

MOV R2, A

ACALL DELAY ;避免視覺暫留讓效果不好

JMP LOOP

DELAY:

MOV R5, #0FFH

DELAY1:

MOV R6, #0FFH

DELAY2:

MOV R7, #05H ;R5、R6、R7的值可以控制delay的時間長度

DELAY3:

DJNZ R7, DELAY3

DJNZ R6, DELAY2

DJNZ R5, DELAY1

RET

**六、心得：**

這次的實驗是我第一次使用8051這種單晶片的微控器，也是我第一次接觸到組合語言，所以從一開始就遇到了不少問題。對開發軟體的不熟悉，光是把該設定的東西設定完就花了我不少時間；設定完成後又遇到虛機無法讀取USB，無法燒錄程式的情況；在所有東西都就緒後，程式燒錄時卻不斷發生錯誤且找不到原因，最後卻只是因為線材接觸不良。種種問題都讓我花費了許多時間與心力去摸索。

另外，組合語言的編寫方式一開始也讓我頗為困惑，因為組合語言的形式與之前學過的C或C++差異甚大，組合語言相較於其他高階語言，有許多小細節要注意，像是我就遇到一開始不知道RR、RL指令只有A累加器才能使用，我在其他暫存器上使用這個指令卻一直遇到我看不懂的錯誤，是後來靠著上網搜尋才找到問題所在。雖然第一次實驗的過程十分混亂，不但弄到很晚加分題也做不出來，但這次實驗課結束後，算是有稍微建立起對微算機的基本認知，頗有斬獲。