



微算機實驗報告

Lab # 4

姓名：仇健安

系級：電機系

學號：111511239

上課時間：2025/03/11

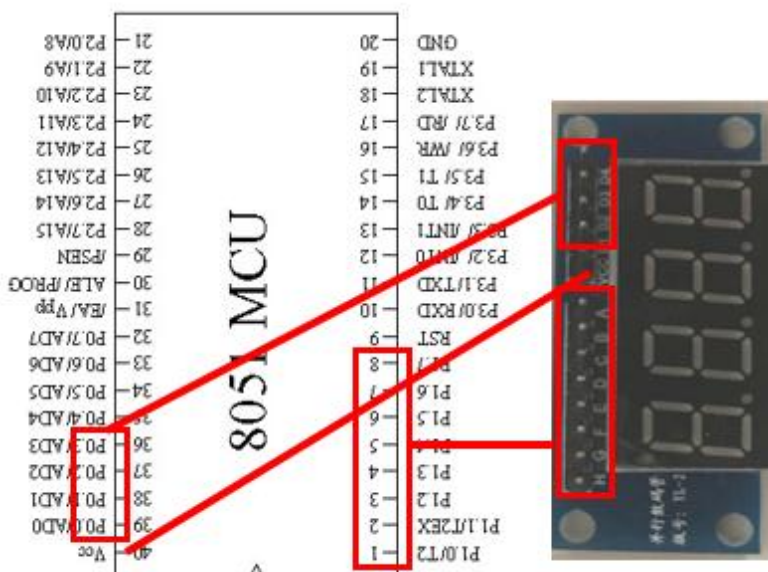
一、實驗目的：

本次實驗的目的是讓我們深入了解「四合一型七段顯示器」的硬體結構與其控制邏輯，並透過實作來掌握如何利用微控制器實現多位數字的顯示。特別是，這組七段顯示器採用「共陽極」與「共用控制訊號」的設計，因此我們需要透過切換陽極控制腳（D1~D4）並配合陰極控制（A~H），使用「視覺暫留」的原理來達成多顯示器的輪流更新，產生「同步顯示」的效果。此外，我們也學習如何設計倒數計時器與碼錶，並掌握如何讓顯示亮度一致與控制時間間隔，這些都是實際系統設計中非常關鍵的技巧。

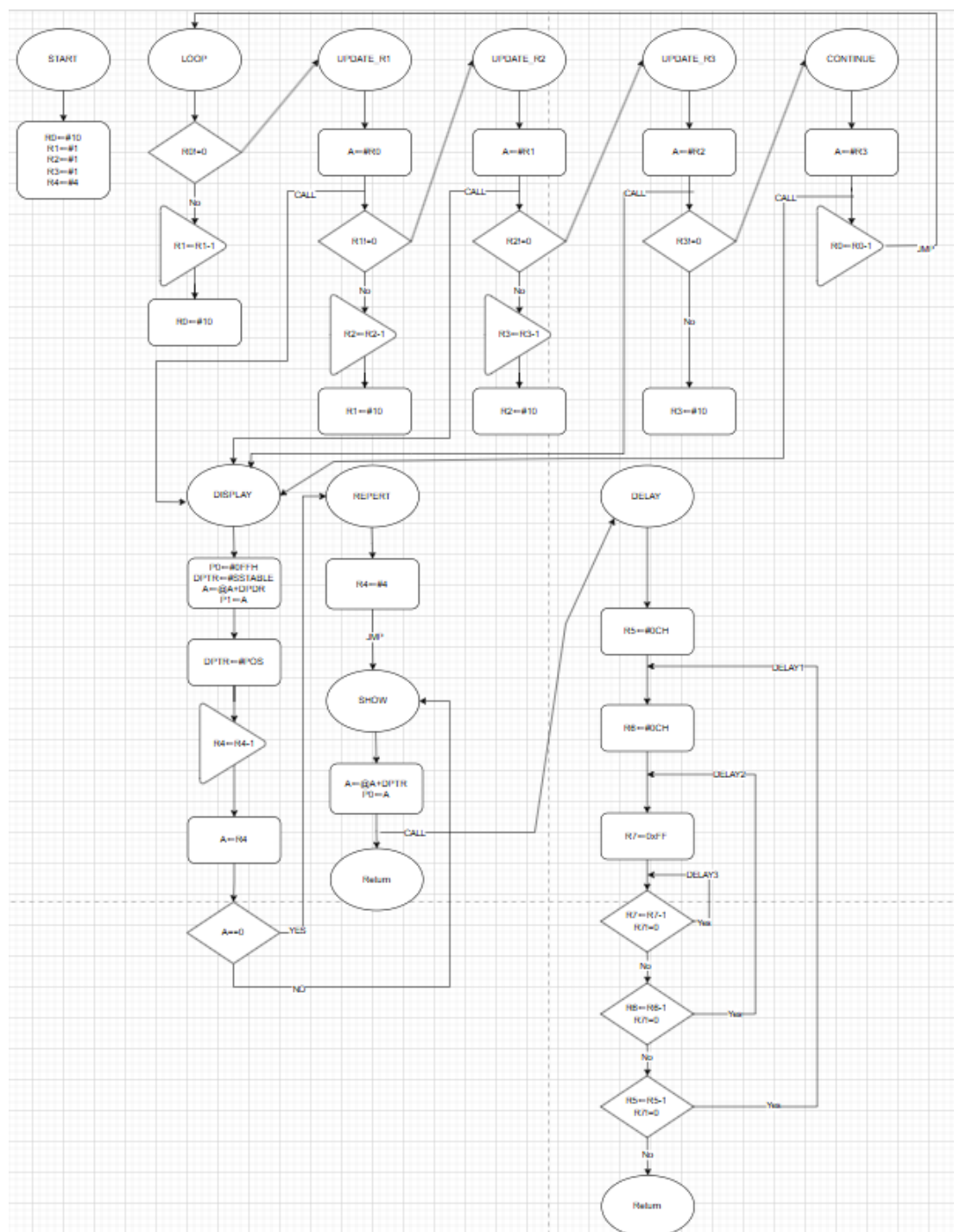
二、硬體架構：

本實驗所使用的七段顯示器為「四合一型共陽極七段顯示器模組」，也就是四個七段顯示器整合在同一塊電路板上。其最大特點為：

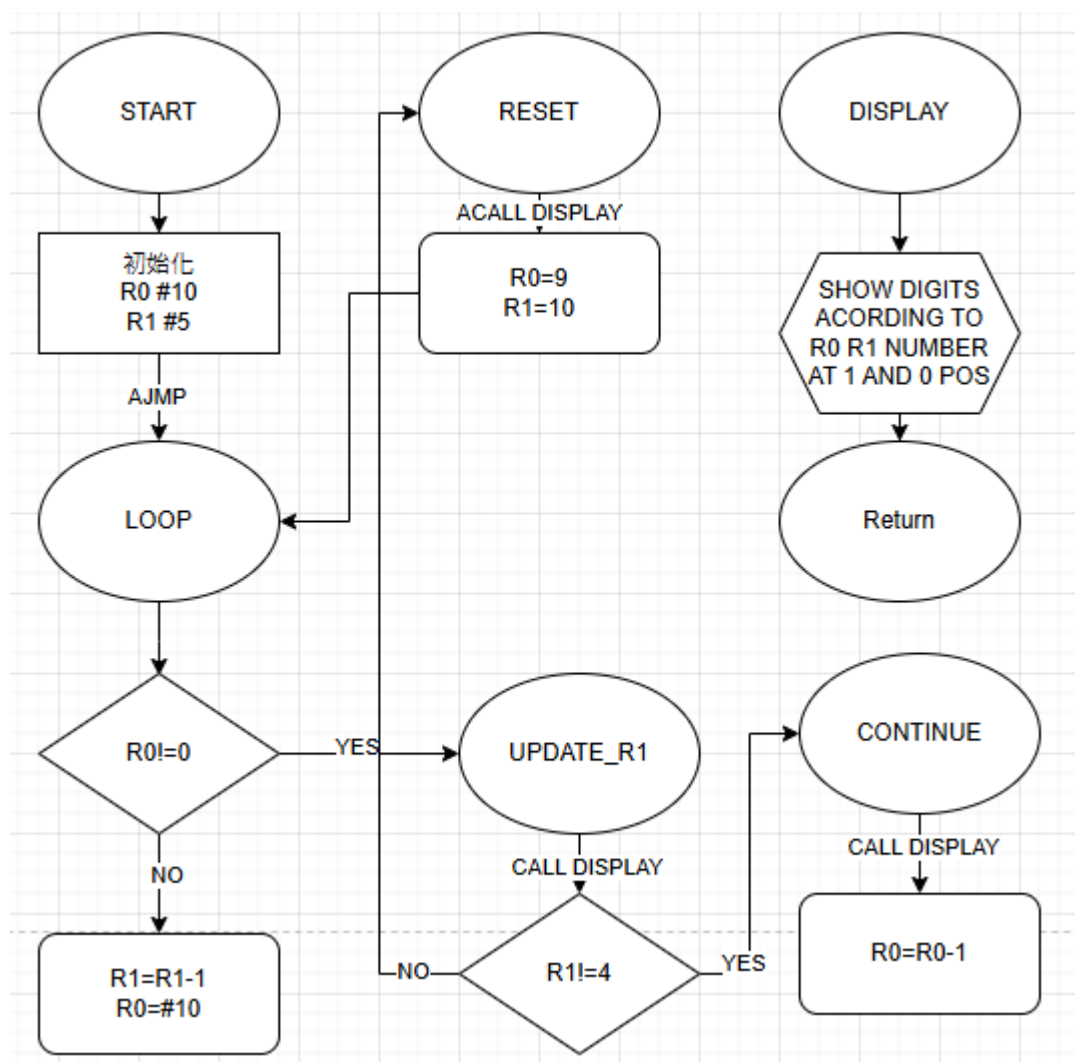
- 共陽極設計：每顆七段顯示器的陽極腳集中接至正電源（VCC），而各段的 LED 陰極則分別接到控制腳位（A~G 與 dot），使其能透過 LOW 電位來點亮特定段數。
- 共用控制訊號：四個顯示器共用同一組段數控制腳（A~H），這代表同一時間只能有一顆顯示器在顯示正確內容，因此需透過快速輪流啟用各顆顯示器的方式，搭配視覺暫留效果，來讓人眼感受到四顯示器同時亮著不同的數字。
- 顯示器選擇腳（D1~D4）：這四個腳位分別控制哪一顆七段顯示器要「被啟動」，也就是是否讓 VCC 通過 PNP 電晶體流入顯示器的共陽極。當 D1~D4 某一腳為 LOW 時，對應的顯示器會被選中；其他顯示器則會保持關閉狀態。
- PNP 電晶體作為開關：這些電晶體用來控制顯示器是否導通。只有在該電晶體的控制腳為 LOW，電晶體才會導通，VCC 才能進入該顯示器，進而配合陰極的 LOW 信號讓 LED 段數亮起。
- 點亮邏輯：只有在「顯示器被選中（D 腳為 0）」且「該段的控制腳為 0（A~H）」的情況下，對應的 LED 才會亮起。否則，不管段數腳怎麼設，都不會亮。



基本題：



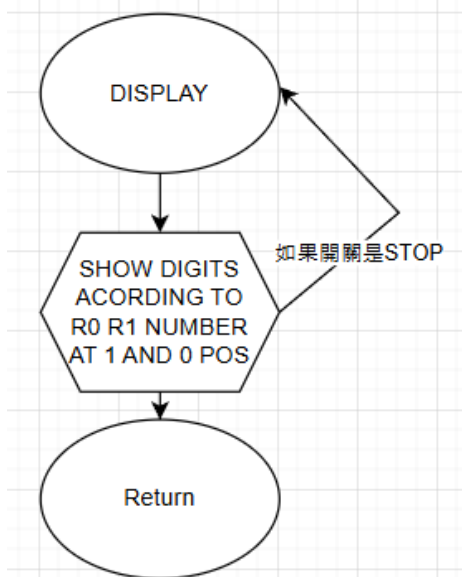
進階題:



四、問題與討論：

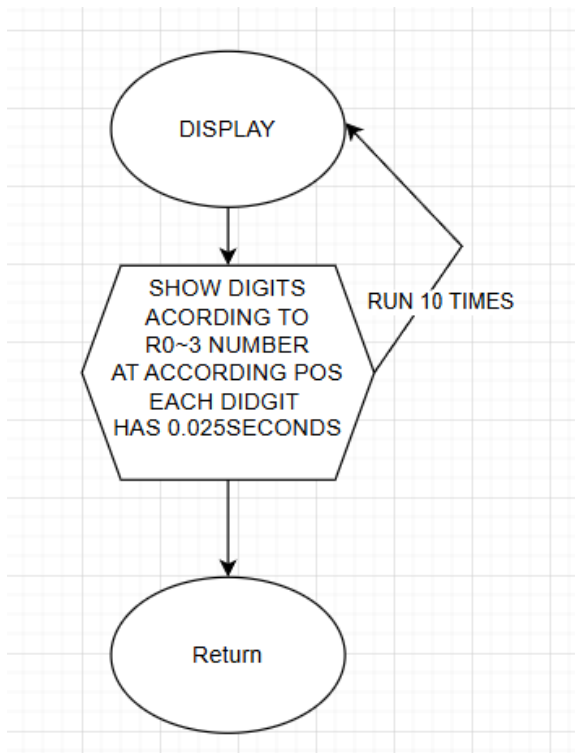
(1) 若要使用一個開關來控制計數器 run 或 stop，要如何設計？(用文字及系統流程圖說明，不用附程式)

在 DISPLAY 函式中新增判斷，如果開關是 stop(0)則利用 CJNE 跳回 DISPLAY 重複顯示，顯示當前數字，反之則繼續執行



(2) 延續本實驗的基本題，若想在一秒中進行最少次數的燈號變換在確保視覺暫留的效果的同時令每顆七段顯示器的亮度一致，要如何設計？(用文字及系統流程圖說明，不用附程式)

要讓人眼感覺「穩定亮著」或「同步顯示」不同內容時，每顆七段顯示器至少要以 10 FPS 被刷新，這代表每一個七段顯示器一秒都要顯示 10 次 所以總共 1 秒內四個七段顯示器要總共閃 40 次，因此大圈的 loop 要設定 10 對應 fps，然後裡面的每個顯示器會分到 $0.1/4=0.025$ 秒的維持時間，這樣就可以完成最低限度的視覺暫留。



五、程式碼與註解：

基本題：

```
ORG 0000H          ; 程式起始位置
JMP START           ; 開機後跳到 START 執行主程式
```

```
ORG 0050H          ; 主程式開始位置
```

START:

```
MOV R0, #10        ; R0：個位數 (從 10 開始，會顯示 9)
MOV R1, #1          ; R1：十位數
MOV R2, #1          ; R2：百位數
MOV R3, #1          ; R3：千位數
MOV R4, #4          ; R4：顯示第幾位 (位置控制用，初始 4)
```

;===== 主倒數迴圈開始 =====

LOOP:

```
CJNE R0, #0, UPDATE_R1 ; 若 R0≠0，跳到 UPDATE_R1；否則個位數減完了
DEC R1                  ; 十位數 -1
MOV R0, #10             ; 個位數歸 10
```

UPDATE_R1:

```
MOV A, R0               ; 把個位數放進累加器
CALL DISPLAY            ; 顯示個位數
CJNE R1, #0, UPDATE_R2 ; 若 R1≠0，跳到 UPDATE_R2；否則十位數減完了
DEC R2                  ; 百位數 -1
MOV R1, #10             ; 十位數歸 10
```

UPDATE_R2:

```
MOV A, R1               ; 把十位數放進累加器
CALL DISPLAY            ; 顯示十位數
CJNE R2, #0, UPDATE_R3 ; 若 R2≠0，跳到 UPDATE_R3；否則百位數減完了
DEC R3                  ; 千位數 -1
MOV R2, #10             ; 百位數歸 10
```

UPDATE_R3:

```
MOV A, R2               ; 把百位數放進累加器
CALL DISPLAY            ; 顯示百位數
CJNE R3, #0, CONTINUE   ; 若 R3≠0，繼續；否則千位數已歸 0
MOV R3, #10             ; 千位數歸 10
```

CONTINUE:

```

MOV A,R3          ; 把千位數放進累加器
CALL DISPLAY      ; 顯示千位數
DEC R0            ; 個位數 -1
JMP LOOP          ; 回到 LOOP 開始下一輪

```

;===== 顯示子程序 =====

DISPLAY:

```

MOV P0, #0FFH      ; 關閉所有位選 (位置選擇)
MOV DPTR, #SSTABLE ; 指向七段碼表
MOVC A, @A+DPTR     ; 取得對應數字的七段碼
MOV P1, A           ; 輸出到七段顯示器的段選腳

MOV DPTR, #POS      ; 指向位置控制表 (D1~D4)

DEC R4              ; R4 -1, 用來決定目前顯示的位置
MOV A, R4
JZ REPEAT           ; 若 R4=0, 跳回 REPEAT (重設 R4 為 4)

```

SHOW:

```

MOVC A, @A+DPTR     ; 取出對應位置的控制碼
MOV P0, A           ; 啟用該位置的七段顯示器
CALL DELAY          ; 顯示時間延遲
RET                 ; 回主程式

```

REPEAT:

```

MOV R4, #4          ; 顯示完一輪, 重設為顯示第 4 位
JMP SHOW            ; 繼續顯示目前這一位 (補上原先沒顯的)

```

;===== 七段碼表: 0~9 對應的段碼 =====

SSTABLE:

```

DB 0H               ; 無用位元, 可略過
DB 0C0H, 0F9H, 0A4H, 0B0H, 099H
DB 092H, 082H, 0F8H, 080H, 090H ; 依序為 0~9 的七段碼

```

;===== 顯示位置控制碼表 (D1~D4 對應 P0 輸出) =====

POS:

```

DB 0FEH, 0FDH, 0FBH, 0F7H, 0FEH ; 輪流選通不同位置

```

;===== 延遲子程式 (控制單位顯示時間) =====

DELAY:

```

MOV R5, #0CH

```

DELAY1:

MOV R6, #0CH

DELAY2:

MOV R7, #0FFH

DELAY3:

DJNZ R7, DELAY3

DJNZ R6, DELAY2

DJNZ R5, DELAY1

RET

END

進階題:

ORG 0000H

JMP START ; 程式起始後跳轉到 START

ORG 0050H ; 主程式開始位址

START:

MOV R0, #10 ; R0 = 個位數，起始設為 10（會先減變成 9）

MOV R1, #5 ; R1 = 十位數，起始從 5 開始（代表 50）

AJMP LOOP ; 跳到主迴圈

; ===== 重設計數器到 49 =====

RESET:

ACALL DISPLAY ; 顯示目前數值 (04)

MOV R0, #9 ; 個位數歸 9

MOV R1, #10 ; 十位數歸 10（會減成 9 表示 49）

; ===== 主倒數計數邏輯 =====

LOOP:

CJNE R0, #0, UPDATE_R1 ; 若 R0 ≠ 0，跳到 UPDATE_R1（還沒減完）

DEC R1 ; 否則，十位數減 1

MOV R0, #10 ; 個位數歸 10

UPDATE_R1:

ACALL DISPLAY ; 顯示目前數值

CJNE R1, #4, CONTINUE ; 若十位數 ≠ 4，跳到 CONTINUE

AJMP RESET ; 若十位數 == 4（下一步是變成 03），就歸 49

CONTINUE:

ACALL DISPLAY ; 顯示目前數值

```
DEC R0                ; 個位數 -1
JMP LOOP              ; 進入下一輪
```

;===== 顯示子程序，會重複刷新畫面達到穩定亮度效果 =====

DISPLAY:

```
MOV R4, #22D          ; R4 為顯示次數計數器（調整刷新時間）
```

DISPLAY_LOOP:

```
MOV DPTR, #SSTABLE
MOV A, R1              ; 先顯示十位數
MOVC A, @A+DPTR        ; 查七段碼表
MOV P1, A              ; 顯示於段選 (P1)
```

```
MOV DPTR, #POS
MOV A, #1              ; 位置 1：十位數顯示器
MOVC A, @A+DPTR
MOV P0, A              ; 顯示於位選 (P0)
CALL DELAY             ; 延遲一段時間
```

```
MOV DPTR, #SSTABLE
MOV A, R0              ; 再顯示個位數
MOVC A, @A+DPTR
MOV P1, A
```

```
MOV DPTR, #POS
MOV A, #0              ; 位置 0：個位數顯示器
MOVC A, @A+DPTR
MOV P0, A
CALL DELAY             ; 延遲一段時間
```

```
DJNZ R4, DISPLAY_LOOP ; 顯示足夠次數（達到視覺暫留），否則結束
RET
```

;===== 七段碼表（0~9，編碼順序與原本顛倒） =====

SSTABLE:

```
DB 090H, 090H, 080H, 0F8H, 082H, 092H, 099H, 0B0H, 0A4H, 0F9H, 0C0H
; 實際對應為：9, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
```

;===== 顯示位置資料（依據 P0 控制 D1~D2） =====

POS:

```
DB 0F7H, 0FBH, 0FDH, 0FEH ; 只用了第 0,1 位（個位、十位）
```


;===== 延遲子程式（控制閃爍間隔）=====

DELAY:

MOV R5, #0CH

DELAY1:

MOV R6, #0BH

DELAY2:

MOV R7, #0FFH

DELAY3:

DJNZ R7, DELAY3

DJNZ R6, DELAY2

DJNZ R5, DELAY1

RET

END

六、心得：

上課心得:

老師在課堂上剛開始講解四合一七段顯示器的時候，我還在想為甚麼只需要 1 對控制腳位就夠了，後來老師提到視覺暫留就完全懂了，除此之外老師更深入的探討指令記憶體和 CPU 與記憶體的合作讓我對電腦架構更清楚。

實驗心得:

自己做出計時器的微電腦程式蠻有成就感的。

Notes:

1. 內容字體大小為 12，間距為單行間距
2. 中文字字體為標楷體
3. 英文字和阿拉伯數字為 Times New Roman
4. 嚴禁抄襲，抄襲者以 0 分計算
5. 請於報告左上角附上照片
6. 每次實驗課繳交上次實驗結報