微算機實驗報告

Lab # 8

姓名：仇健安
系級：電機系
學號：111511239
上課時間：2025/04/15

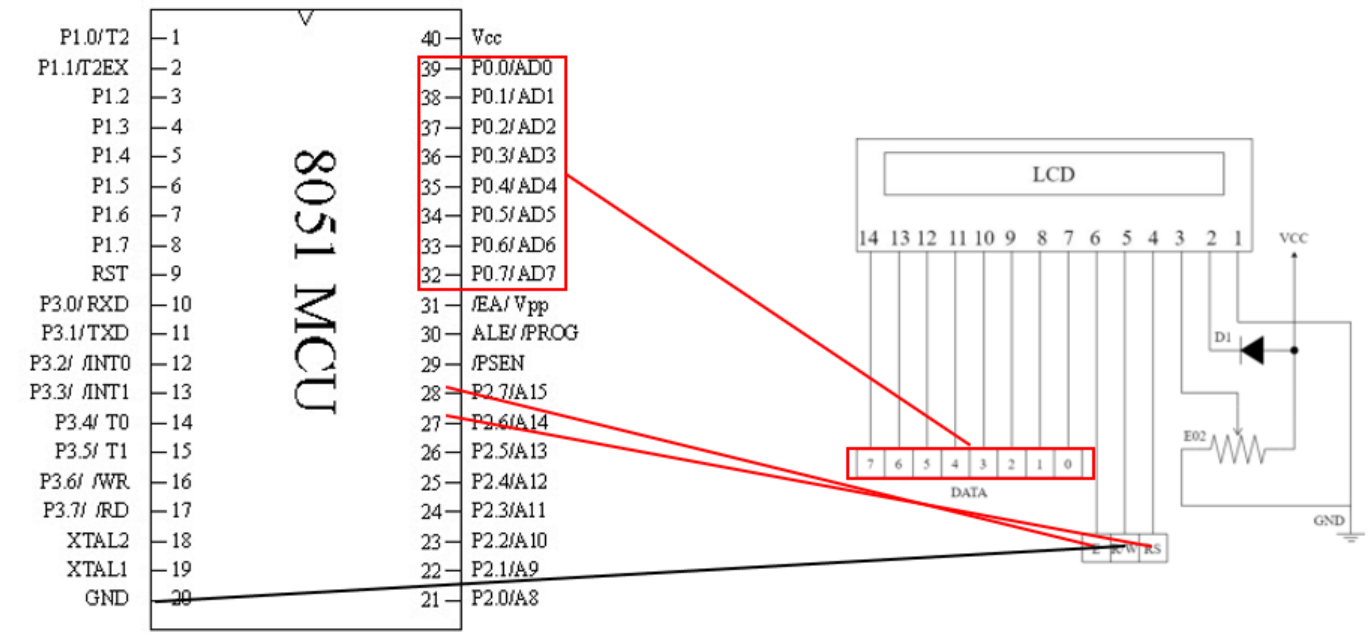
一、實驗目的：

本實驗旨在了解 LCM（液晶顯示模組）的基本工作原理與控制方式，並透過 8051 微控制器實作指令與資料的顯示流程。學生將學習如何撰寫程式控制 LCM，進一步掌握其應用技巧，包含文字顯示與自定義圖形動畫的製作。

二、硬體架構：

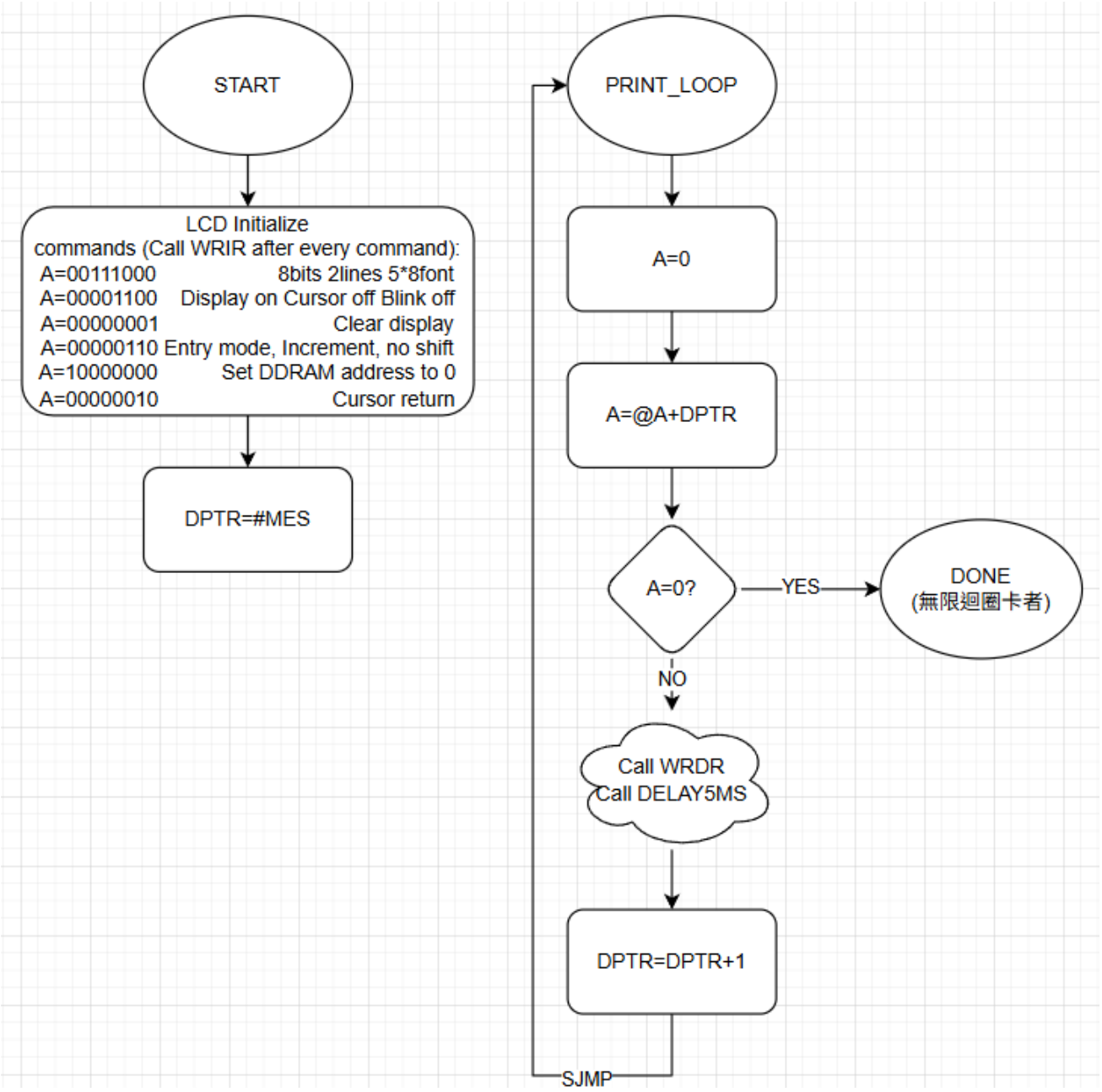
基本題:

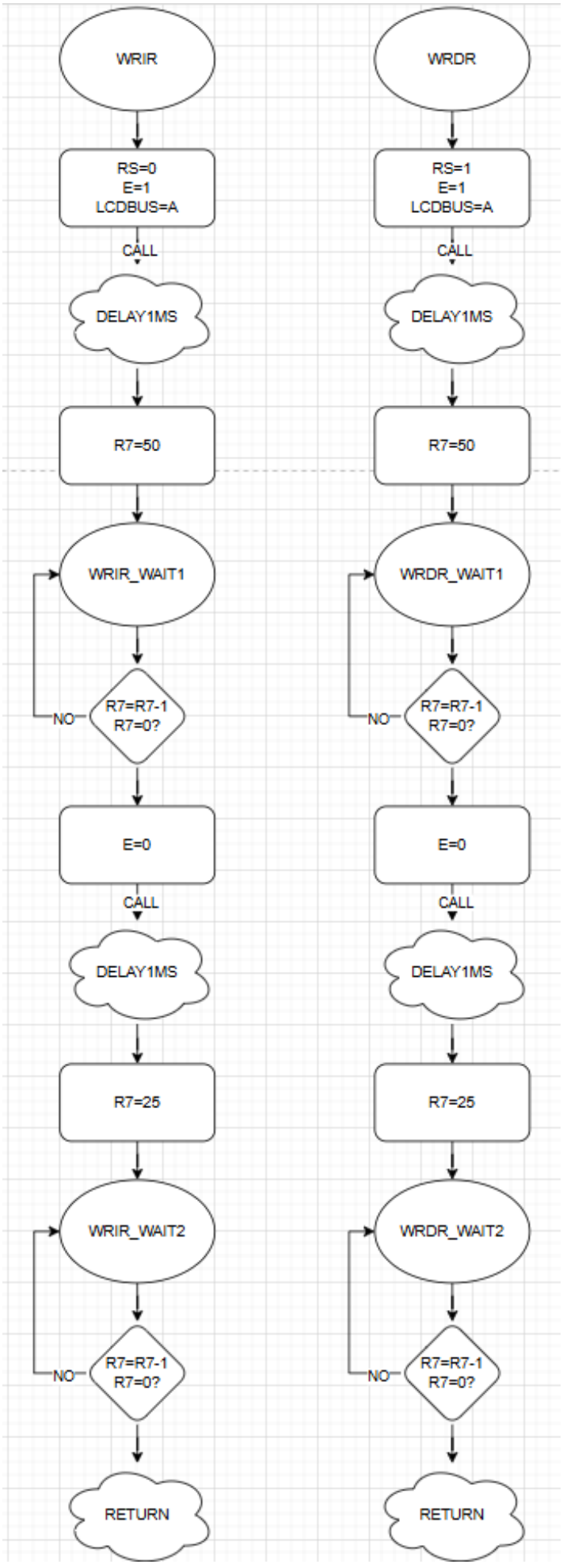
LCD 腳位名稱	**功能說明**	**8051 連接腳位**	**說明**
RS	資料/指令選擇	P2.6	RS=1：資料；RS=0：指令
E (Enable)	資料鎖存觸發	P2.7	E 由 1 到 0 觸發資料鎖存
D0–D7	8-bit 資料匯流排	P0.0–P0.7	使用 P0 作為 8-bit 資料輸入
RW	本實驗恆為零	GND	預設為寫入（RW 接 GND）
VSS	接地	GND	
VDD	電源（+5V）	VCC	
V0	對比調整	(未使用)	控制 LCD 亮度
A / K	背光 LED 電源與地線	(未使用)	視模組而定



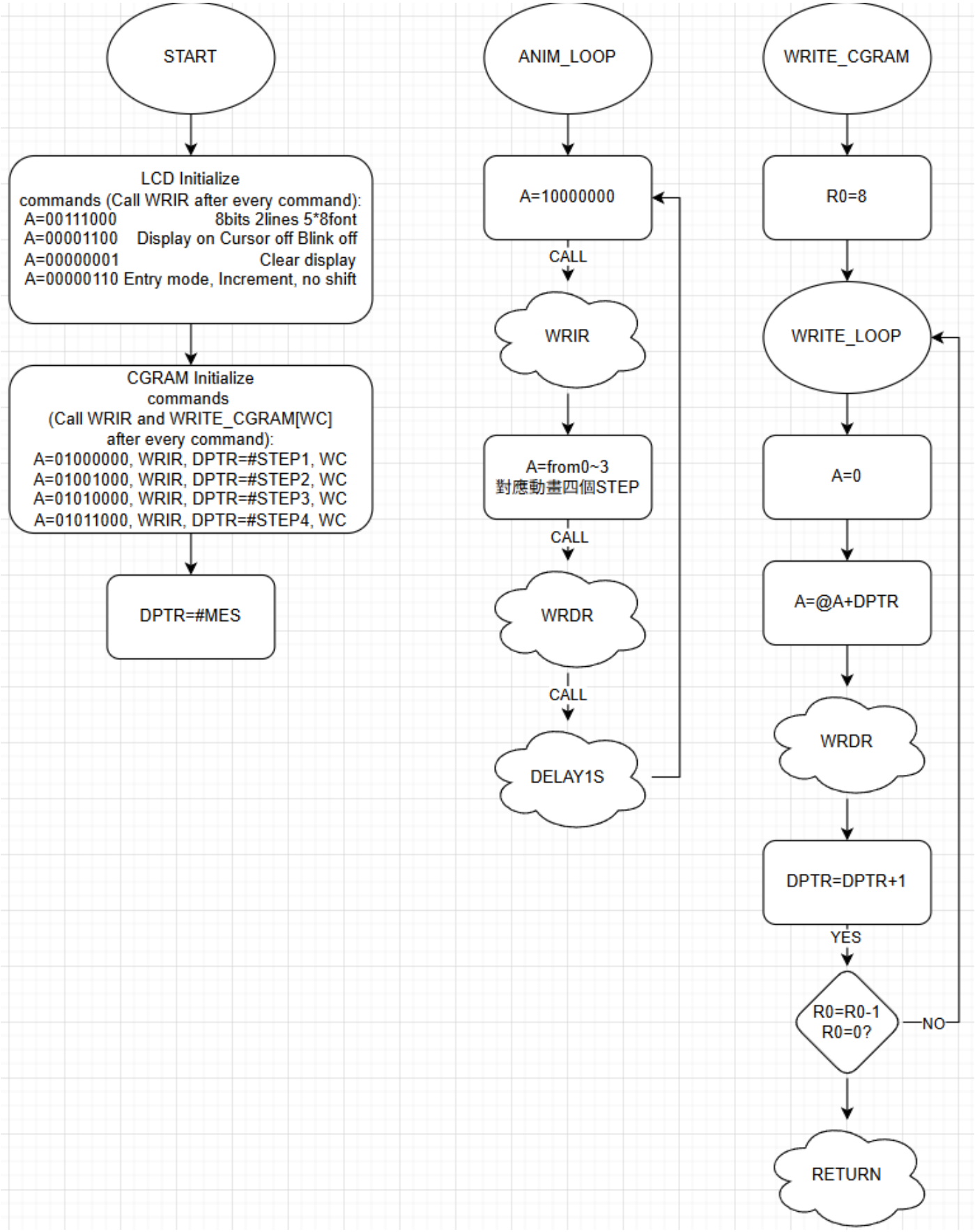
三、程式流程圖：

基本題:


START
LCD Initialize
commands (Call WRIR after every command):
A=00111000 8bits 2lines 5*8font
A=00001100 Display on Cursor off Blink off
A=00000001 Clear display
A=00000110 Entry mode, Increment, no shift
A=10000000 Set DDRAM address to 0
A=00000010 Cursor return
DPTR=#MES
PRINT_LOOP
A=0
A=@A+DPTR
A=0?
YES
DONE
(無限迴圈卡者)
NO
Call WRDR
Call DELAY5MS
DPTR=DPTR+1
SJMP


WRIR
RS=0
E=1
LCDBUS=A
CALL
DELAY1MS
R7=50
WRIR_WAIT1
R7=R7-1
R7=0?
NO
E=0
CALL
DELAY1MS
R7=25
WRIR_WAIT2
R7=R7-1
R7=0?
NO
RETURN
WRDR
RS=1
E=1
LCDBUS=A
CALL
DELAY1MS
R7=50
WRDR_WAIT1
R7=R7-1
R7=0?
NO
E=0
CALL
DELAY1MS
R7=25
WRDR_WAIT2
R7=R7-1
R7=0?
NO
RETURN

進階題:


START
LCD Initialize
commands (Call WRIR after every command):
A=00111000 8bits 2lines 5*8font
A=00001100 Display on Cursor off Blink off
A=00000001 Clear display
A=00000110 Entry mode, Increment, no shift
CGRAM Initialize
commands
(Call WRIR and WRITE_CGRAM[WC]
after every command):
A=01000000, WRIR, DPTR=#STEP1, WC
A=01001000, WRIR, DPTR=#STEP2, WC
A=01010000, WRIR, DPTR=#STEP3, WC
A=01011000, WRIR, DPTR=#STEP4, WC
DPTR=#MES
ANIM_LOOP
A=10000000
CALL
WRIR
A=from0~3
對應動畫四個STEP
CALL
WRDR
CALL
DELAY1S
WRITE_CGRAM
R0=8
WRITE_LOOP
A=0
A=@A+DPTR
WRDR
DPTR=DPTR+1
YES
R0=R0-1
R0=0?
NO
RETURN

四、問題與討論：

(1) 在講義中提到，在進行模組間通訊的資料傳輸時，在讀寫資料或指令需要一定的延遲時間。假設:今天一個系統以 1ms 的取樣週期由 LCM 讀取資料，如果在讀取資料時，延遲時間過短，會發生什麼問題?

1. LCM 尚未準備好資料，導致讀到舊資料或未定資料
2. 產生資料錯誤（bit 錯誤或殘留資料）
3. LCM 狀態機混亂，後續操作出現非預期結果（例如畫面閃爍、亂碼）

(2) 延續上一題，如果在寫指令時，延遲時間過短，會發生什麼問題?(原本這題跟上一題一樣，但我覺得助教應該是想問寫指令和讀資料這兩種狀況)

1. 指令尚未被 LCM 正確接收或執行完畢，就發送下一個指令 → 指令遺失或執行錯亂
2. LCM 控制邏輯未完成內部狀態切換 → 導致操作失效，例如畫面未清除、游標未移動
3. 顯示流程錯亂，畫面顯示異常、跳行、游標位置錯誤

五、程式碼與註解：

基本題:

```
; === 定義 LCD 接腳與資料匯流排 ===
LCDBUS   EQU     P0            ; LCD 資料匯流排接至 P0（8-bit 資料線 D0~D7）
RS       EQU     P2.6          ; RS 腳位接 P2.6，用來選擇資料（1）或指令（0）
ENABLE   EQU     P2.7          ; ENABLE 腳位接 P2.7，用於觸發讀寫動作

; === 程式起始點 ===
ORG 0000H
AJMP START                     ; 一開始跳轉至 START 主程式位置

ORG 0050H
START:
    ; === LCD 初始化流程 ===
    MOV     A,#00111000B       ; 設定 Function Set: 8-bit 傳輸、2 行顯示、5x8 字型
    CALL    WRIR               ; 呼叫寫入指令副程式
    CALL    DELAY5MS           ; 延遲 5ms 等待 LCD 處理

    MOV     A,#00001100B       ; 顯示控制: 顯示開啟、游標關閉、閃爍關閉
    CALL    WRIR
    CALL    DELAY5MS

    MOV     A,#00000001B       ; 清除顯示畫面（清除 DDRAM）
    CALL    WRIR
    CALL    DELAY5MS

    MOV     A,#00000110B       ; 設定輸入模式: 寫入資料後游標右移、不畫面移動
    CALL    WRIR
    CALL    DELAY5MS
```

```
    MOV     A,#10000000B      ; 設定 DDRAM 起始位址為 0（第一列最左邊）
    CALL    WRIR
    CALL    DELAY5MS

    MOV     A,#00000010B      ; 游標回到原點（Home）
    CALL    WRIR
    CALL    DELAY5MS

    ; === 顯示字串資料 ===
    MOV     DPTR, #MES        ; 將資料指標設為 MES 字串起始位址

PRINT_LOOP:
    CLR     A                 ; 將 A 清為 0，以便用來做 @A+DPTR 的位移讀取
    MOVC    A, @A+DPTR        ; 從 MES 中讀出目前字元（使用 A 當 offset）
    JZ      DONE              ; 若讀到 0（NULL），代表結尾 → 跳至 DONE
    CALL    WRDR              ; 顯示該字元至 LCD（寫入資料）
    CALL    DELAY5MS          ; 等待 LCD 顯示處理完成
    INC     DPTR              ; 指標移至下一個字元
    SJMP    PRINT_LOOP        ; 跳回迴圈顯示下一個字

DONE:
    SJMP $                    ; 無限迴圈（程式結束後停在這裡）

; === 顯示字元：傳送資料到 LCD ===
WRDR:
    SETB    RS                ; RS = 1 → 資料模式
    SETB    ENABLE            ; ENABLE = 1，準備資料傳輸
    MOV     LCDBUS, A         ; 將 A 中資料送至 P0（資料匯流排）
    CALL    DELAY1MS          ; 等待資料穩定
    MOV     R7, #50           ; 延遲 50 次迴圈
WRDR_WAIT1:
    DJNZ    R7, WRDR_WAIT1
    CLR     ENABLE            ; ENABLE 由 1 → 0，負緣觸發資料寫入 LCD
    CALL    DELAY1MS
    MOV     R7, #25           ; LCD 執行資料處理延遲
WRDR_WAIT2:
    DJNZ    R7, WRDR_WAIT2
    RET                       ; 回到呼叫處

; === 寫指令：傳送命令到 LCD ===
```

```
WRIR:
    CLR     RS                  ; RS = 0 → 指令模式
    SETB    ENABLE
    MOV     LCDBUS, A
    CALL    DELAY1MS
    MOV     R7, #50
WRIR_WAIT1:
    DJNZ    R7, WRIR_WAIT1
    CLR     ENABLE
    CALL    DELAY1MS
    MOV     R7, #25
WRIR_WAIT2:
    DJNZ    R7, WRIR_WAIT2
    RET

; === 顯示資料區：學號字串 ===
MES:
    DB "111511239", 0           ; 要顯示的字串，最後的 0 為結束判斷用

; === 延遲子程式：大約 1 毫秒 ===
DELAY1MS:
    MOV R6, #109                ; 外層迴圈計數
D1MS_LOOP1:
    MOV R7, #26                 ; 內層迴圈計數
D1MS_LOOP2:
    DJNZ R7, D1MS_LOOP2         ; 內層迴圈跑完
    DJNZ R6, D1MS_LOOP1         ; 外層再跑，構成較長延遲
    RET

; === 延遲約 5ms，呼叫 5 次 1ms 延遲 ===
DELAY5MS:
    ACALL DELAY1MS
    ACALL DELAY1MS
    ACALL DELAY1MS
    ACALL DELAY1MS
    ACALL DELAY1MS
    RET

; === 延遲約 40 微秒（較短延遲，暫未使用） ===
DELAY4OUS:
    MOV R7, #20
```

```
D40US_LOOP:
    DJNZ R7, D40US_LOOP
    RET

END                                 ; 程式結束
```

進階題:

```
; === 腳位設定 ===
LCDBUS   EQU     P0          ; LCD 資料腳位接 P0（D0~D7 對應 LCD 8-bit 資料匯流排）
RS       EQU     P2.6        ; RS 控制腳：RS=1寫資料，RS=0寫指令
ENABLE   EQU     P2.7        ; E 啟用腳：由高轉低觸發資料寫入

ORG 0000H
AJMP START                  ; 從重置向量跳轉至主程式 START

ORG 0050H
START:
    ; === LCD 初始化（參考 HD44780 規格書）===
    MOV     A,#00111000B    ; Function Set：8-bit 模式、2 行、5x8 字型
    CALL    WRIR
    CALL    DELAY5MS

    MOV     A,#00001100B    ; 顯示開啟、游標關、閃爍關
    CALL    WRIR
    CALL    DELAY5MS

    MOV     A,#00000001B    ; 清除顯示（clear display）
    CALL    WRIR
    CALL    DELAY5MS

    MOV     A,#00000110B    ; 輸入模式設為遞增、游標不移動
    CALL    WRIR
    CALL    DELAY5MS

    ; === 設定 CGRAM 為動畫圖案 ===
    MOV A, #01000000B       ; 設定 CGRAM 起始位址為 0x40（STEP1 的位置）
    CALL WRIR
    MOV DPTR, #STEP1        ; 將 DPTR 指向 STEP1 圖案資料
    CALL WRITE_CGRAM        ; 寫入 8-byte 資料

    MOV A, #01001000B       ; 設定 CGRAM 起始位址為 0x48（STEP2）
    CALL WRIR
```

```
    MOV DPTR, #STEP2
    CALL WRITE_CGRAM

    MOV A, #01010000B           ; STEP3
    CALL WRIR
    MOV DPTR, #STEP3
    CALL WRITE_CGRAM

    MOV A, #01011000B           ; STEP4
    CALL WRIR
    MOV DPTR, #STEP4
    CALL WRITE_CGRAM

    ; === 顯示動畫循環 ===
ANIM_LOOP:
    ; 顯示 STEP1（CGRAM 字元 0）
    MOV A, #10000000B           ; 設定 DDRAM 位址為 0（第一列第一格）
    CALL WRIR
    MOV A, #00                  ; 顯示 CGRAM 字元編號 0
    CALL WRDR
    CALL DELAY1S

    ; 顯示 STEP2（字元 1）
    MOV A, #10000000B
    CALL WRIR
    MOV A, #01
    CALL WRDR
    CALL DELAY1S

    ; 顯示 STEP3（字元 2）
    MOV A, #10000000B
    CALL WRIR
    MOV A, #02
    CALL WRDR
    CALL DELAY1S

    ; 顯示 STEP4（字元 3）
    MOV A, #10000000B
    CALL WRIR
    MOV A, #03
    CALL WRDR
```

```
    CALL DELAY1S

    SJMP ANIM_LOOP              ; 無限重複動畫顯示

; ===== 傳送資料到 LCD（顯示字元） =====
WRDR:
    SETB    RS                  ; RS = 1：資料模式
    SETB    ENABLE              ; E = 1，準備觸發
    MOV     LCDBUS, A           ; 傳送資料到 P0
    CALL    DELAY1MS            ; 確保穩定
    MOV     R7, #50
WRDR_WAIT1:
    DJNZ    R7, WRDR_WAIT1
    CLR     ENABLE              ; E = 0 → 負緣觸發資料寫入
    CALL    DELAY1MS
    MOV     R7, #25
WRDR_WAIT2:
    DJNZ    R7, WRDR_WAIT2
    RET

; ===== 傳送指令到 LCD（例如寫位址、清除畫面等）=====
WRIR:
    CLR     RS                  ; RS = 0：指令模式
    SETB    ENABLE
    MOV     LCDBUS, A
    CALL    DELAY1MS
    MOV     R7, #50
WRIR_WAIT1:
    DJNZ    R7, WRIR_WAIT1
    CLR     ENABLE
    CALL    DELAY1MS
    MOV     R7, #25
WRIR_WAIT2:
    DJNZ    R7, WRIR_WAIT2
    RET

; ===== 將 CGRAM 寫入 8 個位元組（每字元 8 列） =====
WRITE_CGRAM:
    MOV R0, #8                  ; 準備寫入 8 列圖形
WRITE_LOOP:
    CLR A
```

```
    MOVC A, @A+DPTR          ; 取出圖案資料
    CALL WRDR                ; 寫入 LCD CGRAM
    INC DPTR                 ; 前往下一列資料
    DJNZ R0, WRITE_LOOP
    RET

; ===== 延遲子程式 =====
DELAY1MS:
    MOV R6, #109
D1MS_LOOP1:
    MOV R7, #26
D1MS_LOOP2:
    DJNZ R7, D1MS_LOOP2
    DJNZ R6, D1MS_LOOP1
    RET                      ; 大約 1ms 延遲

DELAY5MS:
    ACALL DELAY1MS
    ACALL DELAY1MS
    ACALL DELAY1MS
    ACALL DELAY1MS
    ACALL DELAY1MS
    RET

DELAY10MS:
    ACALL DELAY5MS
    ACALL DELAY5MS
    RET

DELAY100MS:
    REPT 10
    ACALL DELAY10MS
    ENDM
    RET

DELAY1S:
    REPT 10
    ACALL DELAY100MS
    ENDM
    RET
```

```
; ===== 自訂小人動畫圖案資料 =====
; 每組 8 位元組對應 LCD 每列圖形的開關（點亮位元）
; 格式為：頭、身體、胯下、手臂、腰、腿1、腿2、空白

STEP1: DB 0EH,04H,0EH,15H,04H,0AH,11H,00H   ; 靜止站立（預備姿勢）
STEP2: DB 0EH,04H,0EH,15H,04H,12H,09H,00H   ; 左腳向前（右腳後）
STEP3: DB 0EH,04H,0EH,15H,04H,0AH,0AH,00H   ; 雙腳張開
STEP4: DB 0EH,04H,0EH,15H,04H,09H,12H,00H   ; 右腳向前（左腳後）

END                                          ; 程式結束
```

六、心得：

上課心得：

這次課程介紹了 LCD 的顯示原理與指令操作，對於第一次接觸 LCD 的我來說，指令的格式與各種功能設定（例如 Function Set、Entry Mode、CGRAM 設定等）一開始有些複雜，不太容易理解它們的功能與組合方式。但透過課本與老師講解逐步學習後，慢慢能掌握每一條指令的用途，也對 LCD 背後的控制邏輯有了初步的認識。

實驗心得：

實驗過程中雖然遇到了不少困難，一開始就拿到好幾個壞掉的 LCD 模組，花了不少時間在測試與更換元件上。後來在接線時也不小心接錯，導致 LCD 沒有正確顯示，讓我一度懷疑程式有錯。不過最後檢查線路後順利修正，加上程式邏輯也沒有問題，總算讓動畫成功在螢幕上顯示出來，雖然過程波折，但看到畫面動起來時還是覺得很有成就感！

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