微算機實驗報告

Lab # 9-2

姓名：仇健安
系級：電機系
學號：111511239
上課時間：2025/05/06

一、實驗目的：

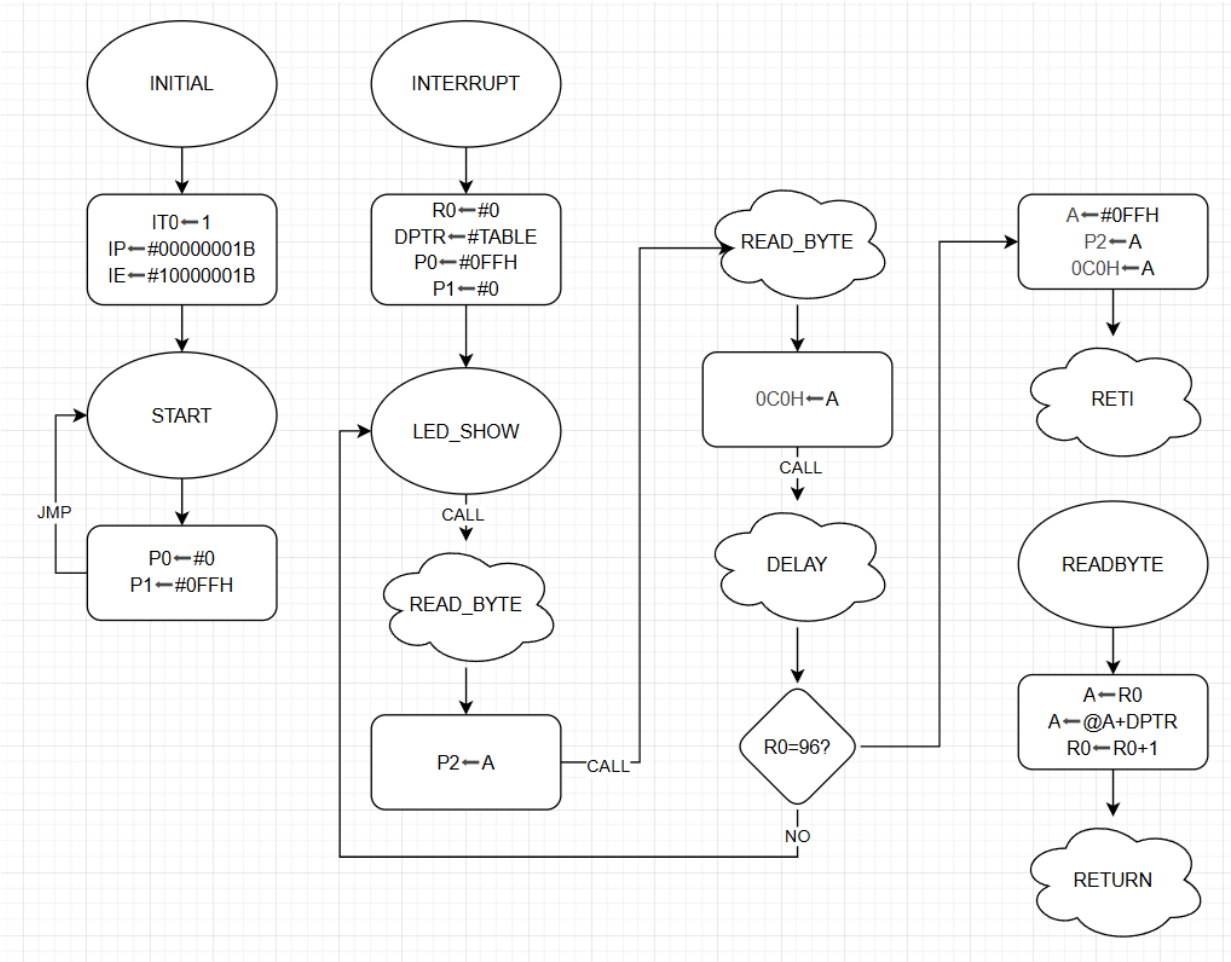
學習如何使用外部中斷控制顯示畫面，並透過讀取字型資料 (TABLE)，在旋轉 LED 上穩定顯示文字。

二、硬體架構：

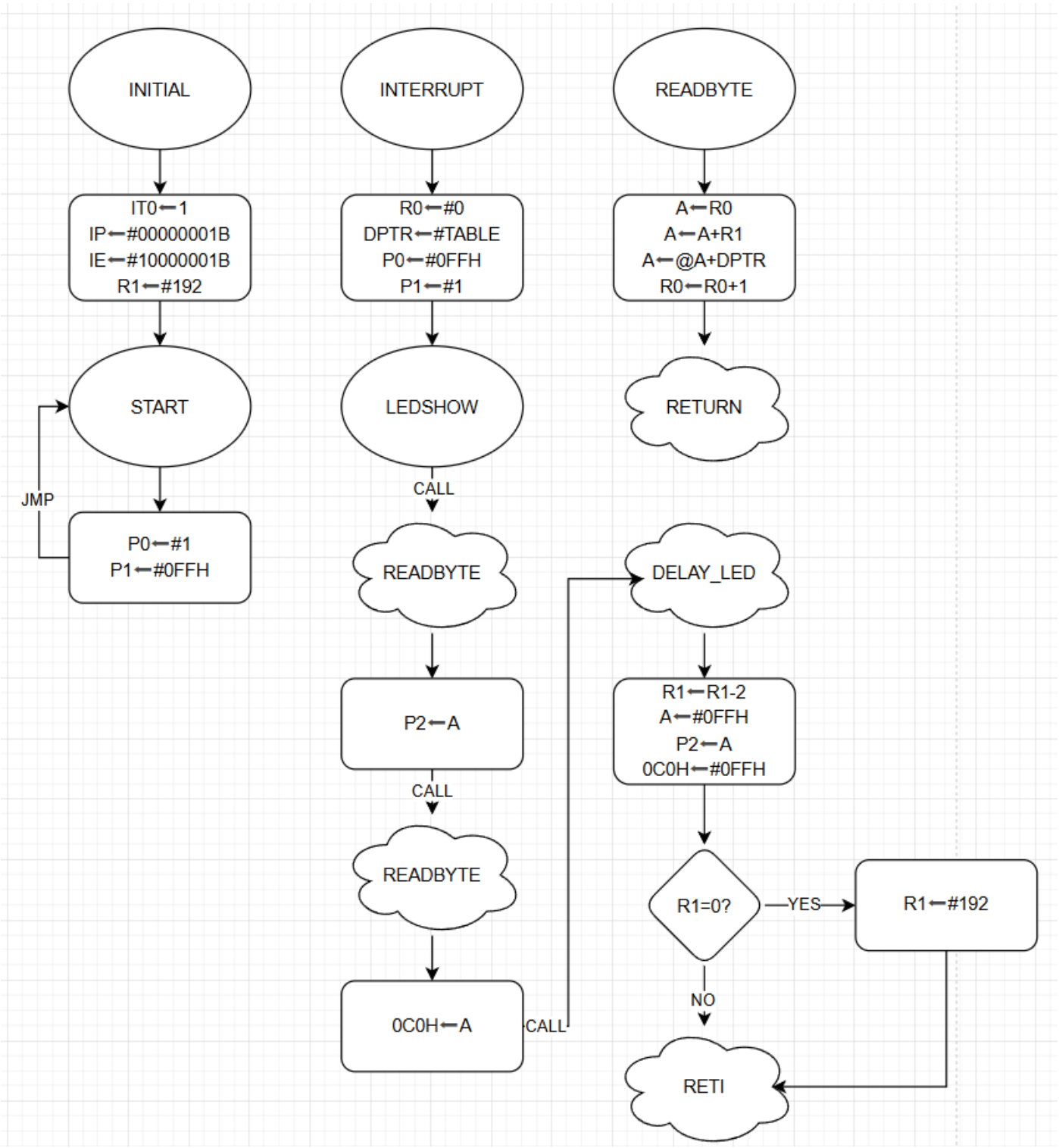
RS232 傳輸線: 黑線對 GND/白線對 RXD/綠線對 TCD/紅線對 VCC

三、程式流程圖：

基本題:



進階題:


INITIAL
IT0←1
IP←#00000001B
IE←#10000001B
R1←#192
START
JMP
P0←#1
P1←#0FFH
INTERRUPT
R0←#0
DPTR←#TABLE
P0←#0FFH
P1←#1
LEDSHOW
CALL
READBYTE
P2←A
CALL
READBYTE
0C0H←A
CALL
READBYTE
A←R0
A←A+R1
A←@A+DPTR
R0←R0+1
RETURN
DELAY_LED
R1←R1-2
A←#0FFH
P2←A
0C0H←#0FFH
R1=0?
YES
R1←#192
NO
RETI

四、問題與討論：

詳述如何使旋轉 LED 進入外部中斷。

設定外部中斷觸發方式：

將中斷 0（INT0）或中斷 1（INT1）設定為負緣觸發模式。

這可透過 TCON 暫存器進行設定，例如 IT0＝1 表示 INT0 為負緣觸發。

開啟中斷功能：

透過 IE（中斷使能暫存器）開啟對應的外部中斷位元，例如 EX0＝1 開啟 INT0。

同時需設定總中斷允許位 EA＝1。

撰寫中斷處理副程式（ISR）：

在對應的中斷向量位址處(如 INT0 是 ORG 0003H)，撰寫 AJMP 指令跳至中斷處理副程式。

在副程式中進行顯示畫面起始位置的校正或重設動作。

實體電路觸發：

當旋轉 LED 模組每轉一圈時，感測元件會輸出一個負緣訊號至 INT 腳位，進而觸發中斷。

五、程式碼與註解：

基本題:

```
ORG 0000H            ; 程式起始位址
JMP INITIAL          ; 跳至初始化程式

ORG 0003H            ; 外部中斷0（INT0）向量位址
AJMP INTERRUPT       ; 中斷觸發後跳至 INTERRUPT 程式段

ORG 0050H            ; 主程式與資料區段開始

;==============================
; TABLE: 顯示「別當我」的字型資料（每字為16x16）
; 每筆字元資料共32 bytes（兩個 byte 組成一列，共16列）
;==============================
TABLE:
;「別」
DB 0FFH,0FFH, 0FFH,07FH, 001H,0BFH, 0BBH,0C7H
DB 03BH,0F8H, 0BBH,0BDH, 0BBH,03DH, 0BBH,0BDH
DB 001H,0C1H, 0FBH,0FDH, 0FFH,0FFH, 007H,0D8H
DB 0F7H,0BFH, 0FFH,03FH, 001H,080H, 0FDH,0FFH

;「當」
DB 0FFH,0FFH, 0BFH,0FFH, 0C7H,0FFH, 0EDH,003H
DB 0EBH,0ABH, 027H,0AAH, 0AFH,0AAH, 0AFH,0AAH
DB 0A1H,082H, 0ADH,0AAH, 0AFH,0AAH, 027H,0AAH
DB 0E9H,0ABH, 0ADH,003H, 0C7H,0FFH, 0EFH,0FFH

;「我」
```

```
DB 0FFH,0FFH, 0BFH,0F7H, 0B7H,0A7H, 0B7H,0B7H
DB 0B7H,037H, 003H,080H, 0BBH,0FBH, 0BBH,0FBH
DB 0BFH,07DH, 0BFH,0BFH, 001H,0D8H, 0BDH,0E7H
DB 0BFH,0DBH, 0BBH,0BCH, 0B7H,07EH, 0A7H,00FH

;=============================
; 程式初始化段
;=============================
INITIAL:
    SETB IT0              ; 設定 INT0 為負緣觸發中斷
    MOV IP, #00000001B    ; 設定 INT0 為高優先權中斷
    MOV IE, #10000001B    ; 開啟全域中斷 EA=1，並啟用 INT0 (EX0=1)

;=============================
; 主程式迴圈（等待中斷觸發）
;=============================
START:
    MOV P0, #0            ; 水平 LED（P0）全亮（共陽極，低電位亮）
    MOV P1, #0FFH         ; 水平 LED（P1）全滅（P1未顯示資料）
    JMP START             ; 無限迴圈等待中斷

;=============================
; 外部中斷觸發後進入的中斷服務程式
;=============================
INTERRUPT:
    MOV R0, #0            ; R0 作為 TABLE 索引，初始化為 0
    MOV DPTR, #TABLE      ; 將 DPTR 指向 TABLE 起始位址
    MOV P0, #0FFH         ; 水平 LED（P0）關閉
    MOV P1, #0            ; 水平 LED（P1）全亮（亮畫面）

; 顯示文字資料（從 TABLE 依序取出並顯示）
LEDSHOW:
    CALL READBYTE         ; 取出一個 byte（輸出至 P2）
    MOV P2, A
    CALL READBYTE         ; 取出下一個 byte（輸出至 P4）
    MOV 0C0H, A           ; P4 無法直接使用，需用 MOV 0C0H 存取
    CALL DELAY            ; 保持顯示一段時間
    CJNE R0, #96, LEDSHOW ; 共有 96 byte（3 字 × 32 byte），未顯示完就重複

; 清除畫面
    MOV A, #0FFH
```

```
    MOV P2, A
    MOV 0C0H, #0FFH
    RETI                    ; 中斷服務結束，返回主程式

;============================
; 從 DPTR + R0 取出資料
;============================
READBYTE:
    MOV A, R0               ; 將索引值 R0 複製到 A
    MOVC A, @A+DPTR         ; 使用 A + DPTR 從 CODE 區讀取對應的 TABLE 資料
    INC R0                  ; 索引加 1，指向下一個資料
    RET

;============================
; 顯示延遲（讓每列文字閃爍可見）
;============================
DELAY:
    MOV R7, #10             ; 外層迴圈次數
DELAY1:
    MOV R6, #40             ; 內層迴圈次數
DELAY2:
    DJNZ R6, DELAY2         ; 內層延遲
    DJNZ R7, DELAY1         ; 外層延遲
    RET

END                         ; 程式結束標記
```

進階題:

```
;============ 程式起始位址 ============
ORG 0000H
JMP INITIAL              ; 開機後跳至 INITIAL 初始化

;============ 外部中斷入口點 (INT0) ============
ORG 0003H
JMP INTERRUPT            ; 進入中斷後執行 INTERRUPT 程式碼

;============ 主程式區起始 ============
ORG 0050H

;============ 字型資料表 (TABLE) ============

TABLE:
; 前面插入了多行 0xFF 作為空白，讓字畫面可以「隱沒」和「捲動」
; 共 96 Bytes（48 列，每列 2 Bytes）
; 可視為字體前後的 padding
DB 0FFH,0FFH   ; 重複數列
; ... (共 96 Bytes padding) ...

; 實際字資料：別、當、我（每字 16 列 × 2 Bytes，共 96 Bytes）
; 詳細略同前面已註解版本

; 後面再插入空白，達到類似捲動到空白再重來的效果
DB 0FFH,0FFH
; ... (共 96 Bytes padding) ...

;============ 初始化設定 ============
INITIAL:
    SETB IT0                ; 設定 INT0 為負緣觸發（falling edge）
    MOV IP, #00000001B      ; 設定 INT0 為高優先權
    MOV IE, #10000001B      ; 啟用全域中斷（EA=1），開啟 EX0 中斷
    MOV R1, #192            ; 設定滑動視窗的起始偏移（192 Bytes 對應畫面最左）

;============ 主迴圈：等待中斷 ============
START:
    MOV P0, #0              ; 平面 LED（P0）全亮
    MOV P1, #0FFH           ; P1 全滅
    JMP START               ; 無限迴圈等待中斷
```

```
;============ 中斷處理程式：顯示一輪文字 ============
INTERRUPT:
    MOV R0, #0              ; R0 作為 index 計數器，初始化為 0
    MOV DPTR, #TABLE        ; 將 DPTR 指向 TABLE 開頭
    MOV P0, #0FFH           ; P0 關閉 LED
    MOV P1, #0              ; P1 全亮

;========== 文字掃描與顯示 ==========
LEDSHOW:
    CALL READBYTE           ; 讀取一個 Byte，放到 A -> P2
    MOV P2, A
    CALL READBYTE           ; 讀取第二個 Byte，放到 A -> P4
    MOV 0C0H, A
    CALL DELAY              ; 留下一段時間
    CJNE R0, #96, LEDSHOW   ; 顯示 96 Bytes（48列）後跳出

;========== 移動畫面邏輯 ==========
    DEC R1                  ; 向左滑動一列（兩 Bytes）
    DEC R1
    MOV A, #0FFH            ; 清除 LED
    MOV P2, A
    MOV 0C0H, #0FFH
    CJNE R1, #0, RETURN     ; 若還沒滑完，回去等下次中斷
    MOV R1, #192            ; 若滑完則重設為起始位置（重新從最右邊開始）

RETURN:
    RETI                    ; 返回主程式等待下一圈觸發

;============ 從 TABLE 讀一筆資料（含位移量 R1） ============
READBYTE:
    MOV A, R0               ; A = 當前 index
    ADD A, R1               ; 加上滑動偏移
    MOVC A, @A+DPTR         ; 取出對應資料
    INC R0                  ; index +1，下一筆資料
    RET

;============ 簡單延遲，讓每列停留時間可見 ============
DELAY:
    MOV R7, #10
DELAY1:
    MOV R6, #40
```

```
DELAY2:
    DJNZ R6, DELAY2
    DJNZ R7, DELAY1
    RET

END
```

六、心得：

（一）上課內容心得

本次課程老師講解了”外部中斷（External Interrupt）”的原理與使用方式，特別是 8051 微控制器中 INT0、INT1 的應用。在課堂中老師詳細介紹了如何設定中斷觸發模式（例如負緣觸發）、中斷使能（IE 暫存器）與優先權（IP 暫存器），並透過實例講解中斷服務程式（ISR）的撰寫與跳轉。這讓我更清楚中斷的功能是讓微處理器可以在非主動監控的情況下，及時響應外部事件，是嵌入式系統中不可或缺的重要機制。過去對中斷的理解僅停留在理論層面，這次的課堂內容讓我對實際的設定步驟與執行流程有了具體的認識。

（二）實驗內容心得

在本次實驗中，我實際應用所學，將外部中斷用於控制旋轉 LED 模組的顯示起始點。當感測器輸出訊號觸發 INT0 時，中斷程式會依據大型 TABLE 中的資料，依序點亮 P2 與 P4 的 LED，使畫面穩定呈現「別當我」的文字。實驗中，老師強調了中斷程式設計時要避免耗時過長，以及如何配合延遲與資料指標移動實現動畫效果，這對我的實作思維有很大的幫助。特別是在進階題實作中，運用了中斷與滑動位移結合，使文字能從右往左平滑捲動，這讓我體會到中斷程式結合硬體設計可以實現非常直觀且有趣的效果。

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