



微算機實驗報告

Lab # 7

姓名：仇健安
系級：電機系
學號：111511239
上課時間：2025/04/08

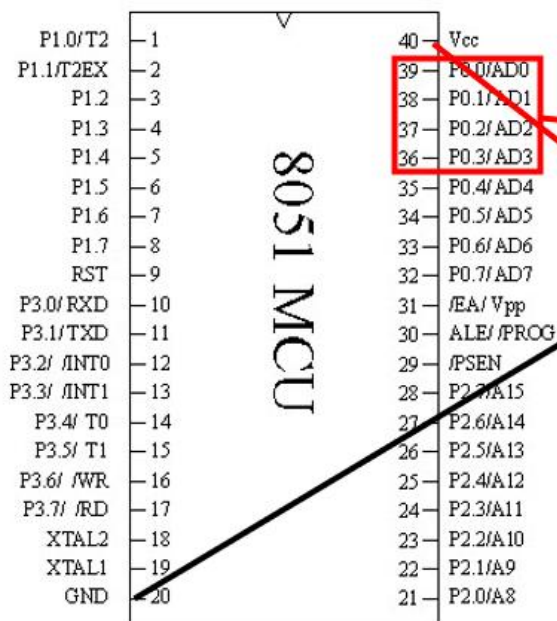
一、實驗目的：

本實驗旨在了解步進馬達之基本運作原理，學習其驅動電路架構及控制方法。透過實作 1 相激磁、2 相激磁與 1-2 相激磁等控制模式，觀察其在轉矩、穩定性與耗電間的差異，並透過鍵盤輸入實現多角度控制，培養學生設計控制程式與解析驅動邏輯的能力

二、硬體架構：

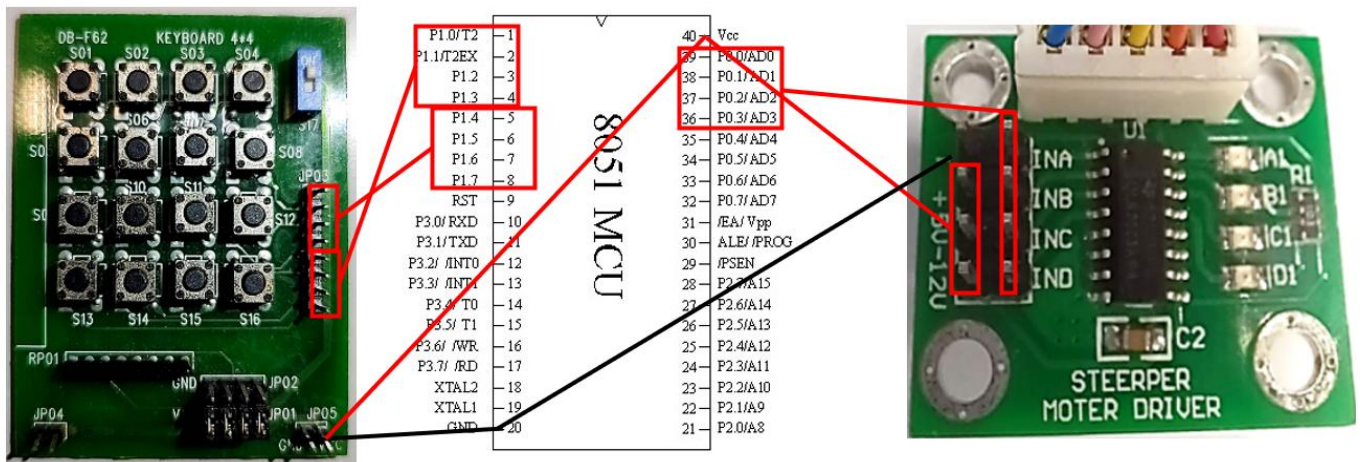
基本題：

功能說明	步進馬達 相位	ULN2003A 輸入端	輸 出	8051 腳 位	控制程式碼 對應值	備註說明
第 1 相控制	A 相	IN1		P0.0	#04H	二進位 00000001，只啟動 P0.0
第 2 相控制	B 相	IN2		P0.1	#02H	二進位 00000010，只啟動 P0.1
第 3 相控制	C 相	IN3		P0.2	#01H	二進位 00000100，只啟動 P0.2
第 4 相控制	D 相	IN4		P0.3	#08H	二進位 00001000，只啟動 P0.3
電 源 供 應 (VCC)	馬達電源	VCC		VCC	-	接至開發板 VCC
接地 (GND)	馬達地線	GND		GND	-	接至開發板 GND



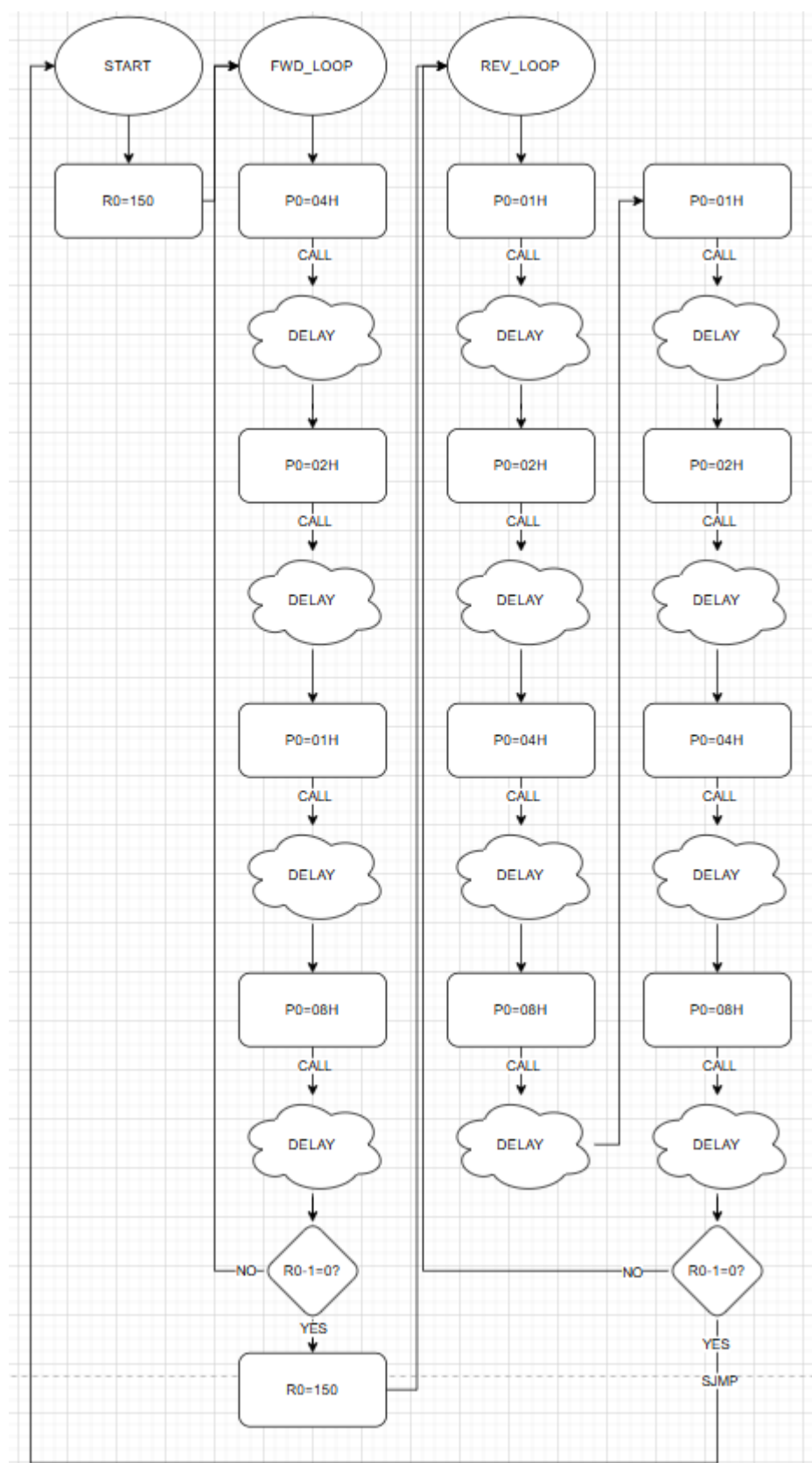
進階題:

功能說明	元件	對應腳位	8051 腳位	說明
小鍵盤列掃描	小鍵盤 Row1~4	R1~R4 (JP03.0~3)	P1.7、P1.6、P1.5、 P1.4	利用 MOV P1, #7FH 等輸出位元控制列電壓
小鍵盤行讀取	小鍵盤 Col1~4	C1~C4 (JP03.4~7)	P1.3、P1.2、P1.1、 P1.0	透過讀取 P1 的低四位元來判斷按鍵
馬達控制輸出	ULN2003 IN1~IN4	IN1~IN4	P0.0 ~ P0.3	1 相、2 相、1-2 相激磁皆由此輸出控制
馬達電源	步進馬達 VCC	電源輸入	-	接開發板 VCC
馬達接地	步進馬達 GND	接地	-	接開發板 GND

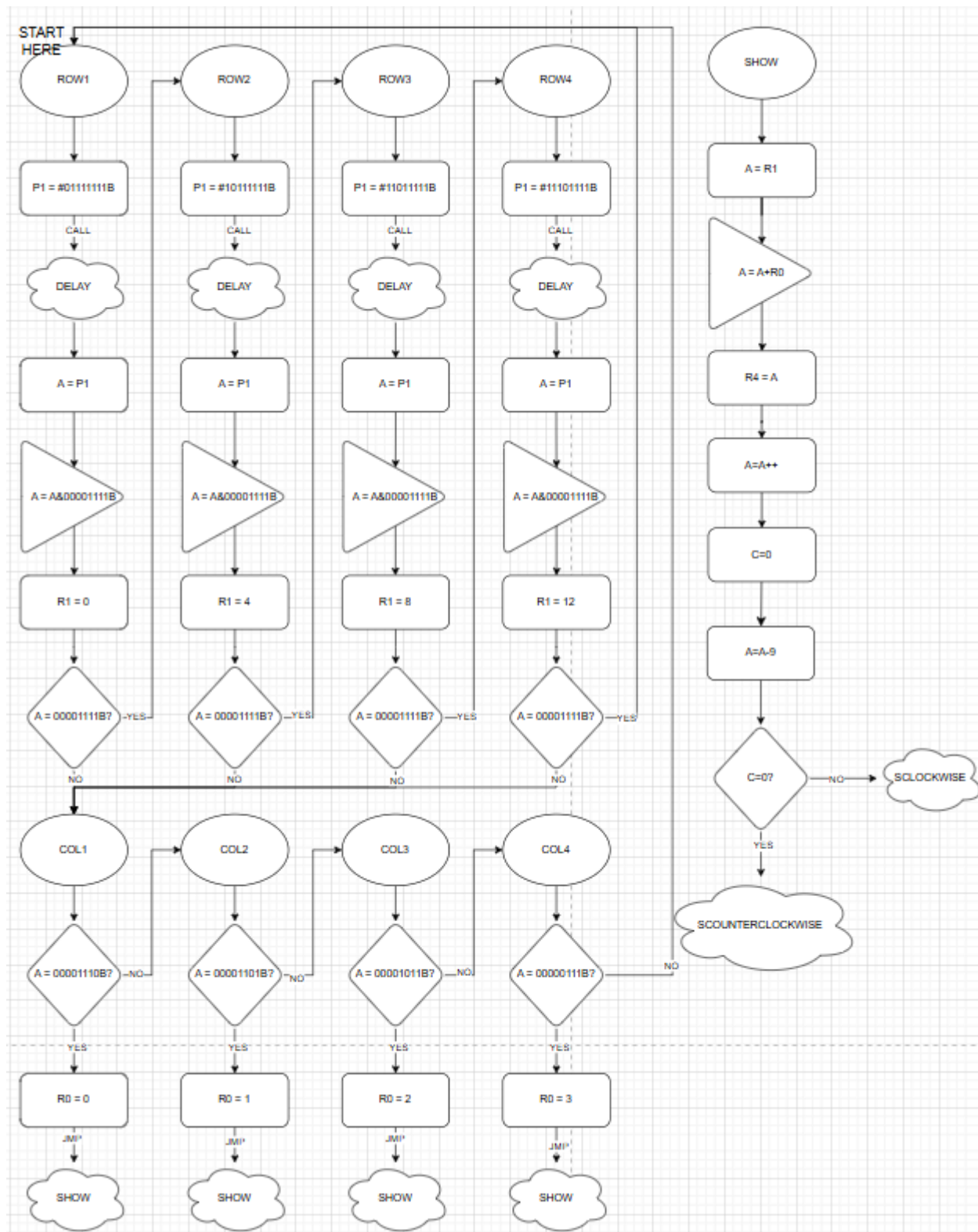


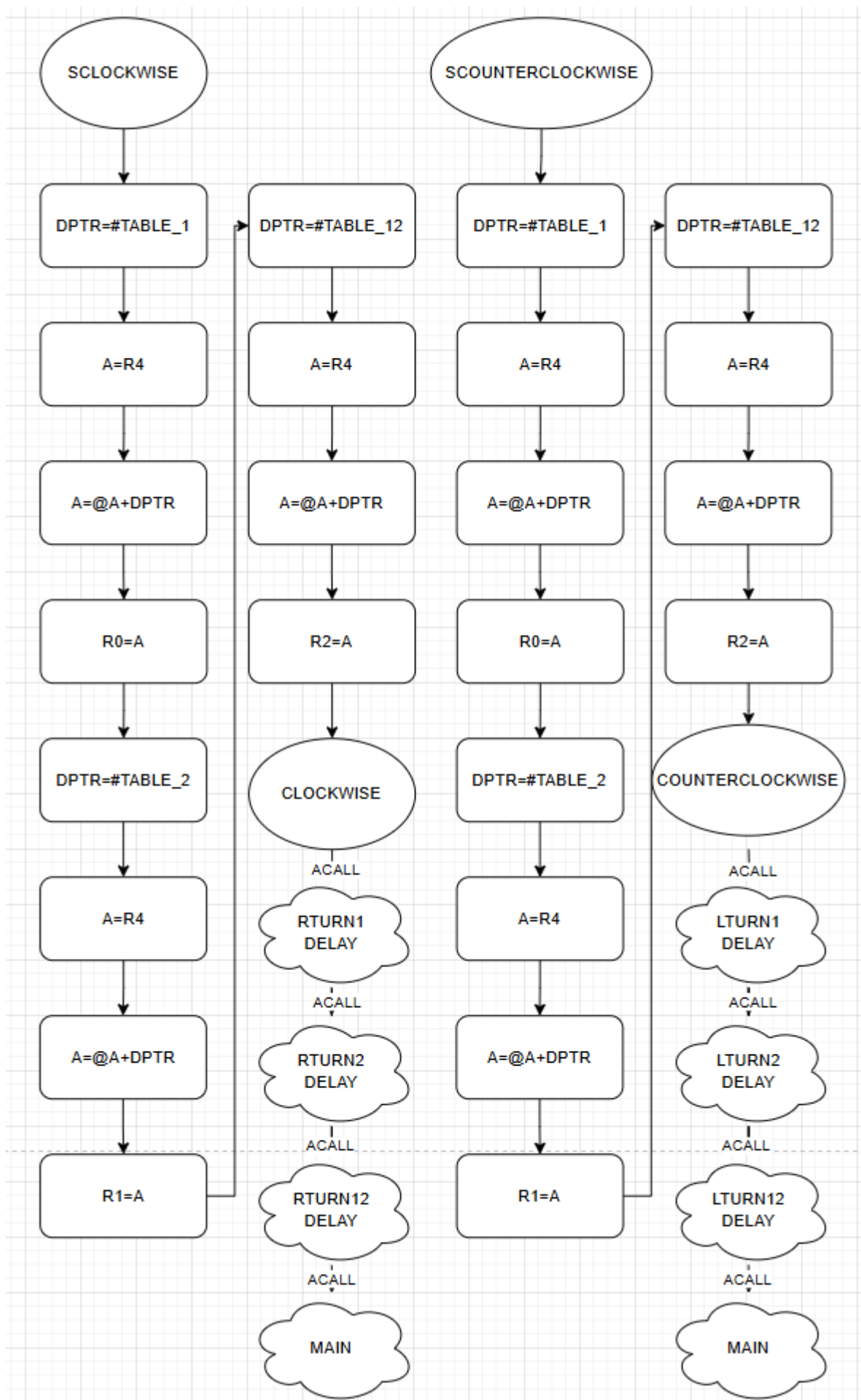
三、程式流程圖：

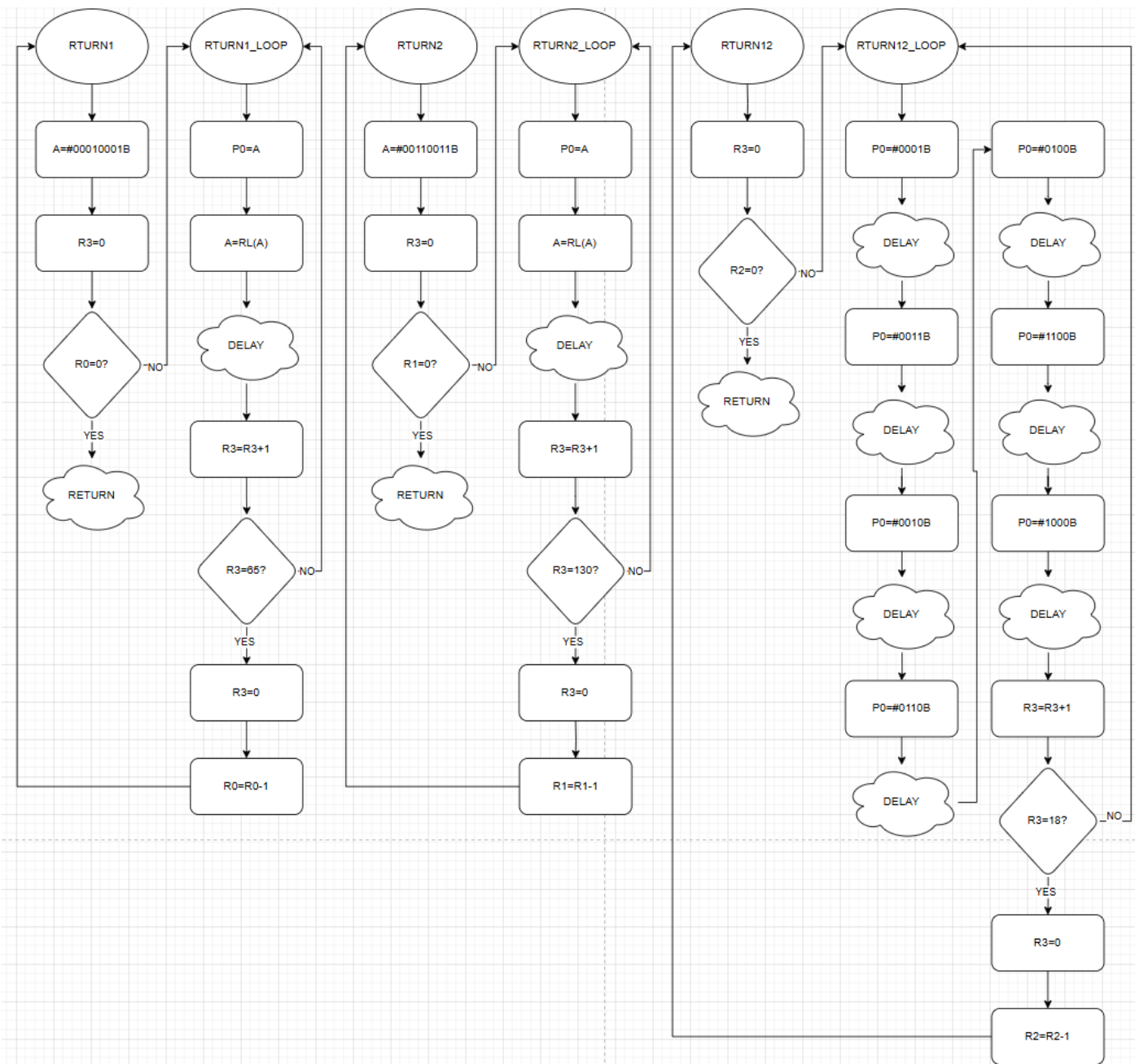
基本題：

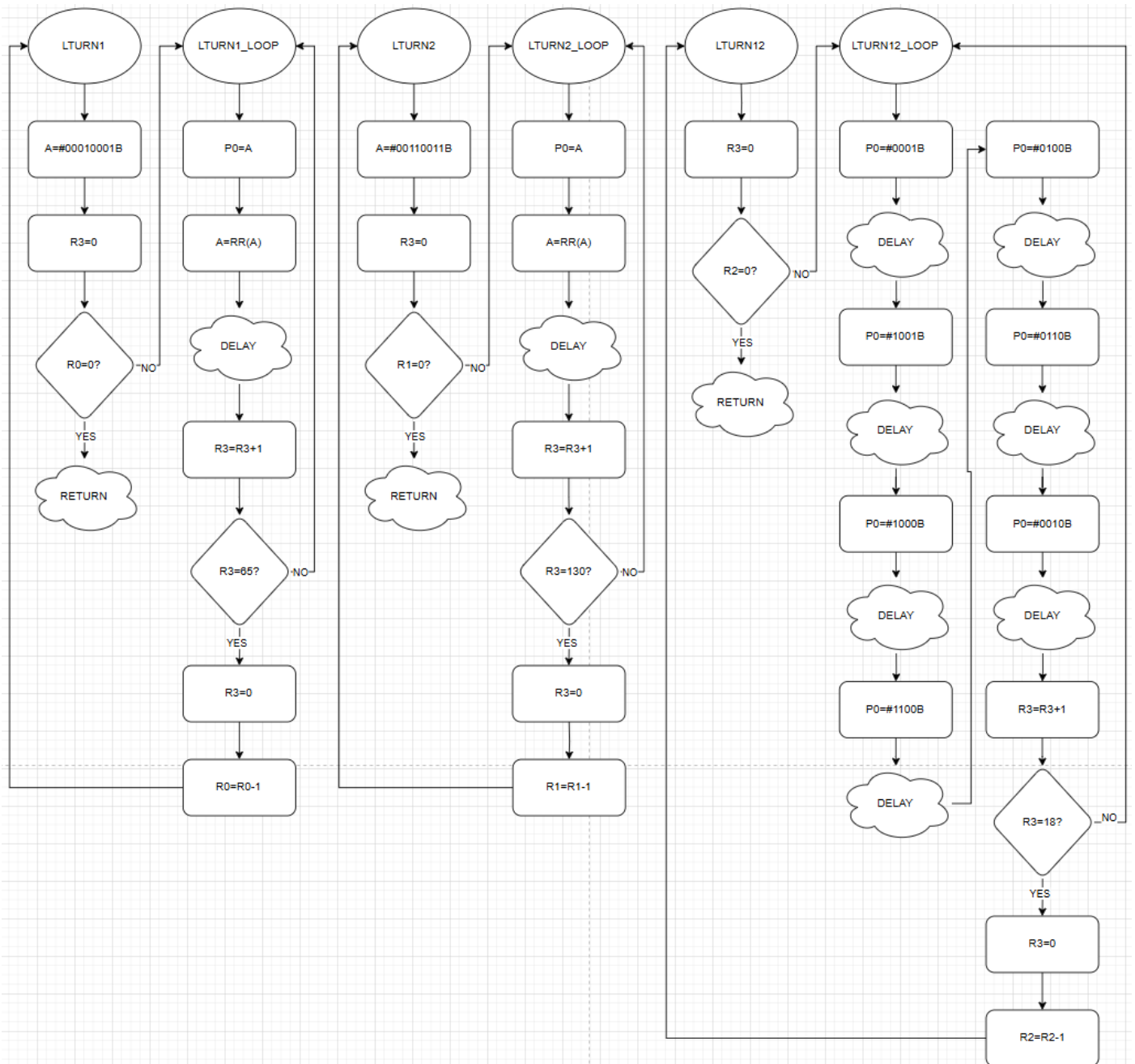


進階題:









四、問題與討論：

(1) 當步進馬達轉動時，馬達驅動板上的四個 LED 燈會點亮，請參考電路圖解釋其意義。

驅動板上的四個 LED 分別對應 ULN2003A 晶片的輸出腳位 (Out1 ~ Out4)，這些輸出端連接至步進馬達的四個相位線圈 (A、B、C、D)。當 MCU 傳送控制訊號 (例如 MOV P0, #04H) 時，對應的輸入腳 (INx) 會被拉低，ULN2003A 內部 Darlington Pair 開啟，讓電流從馬達線圈流經晶片至地，形成迴路，此時對應的 LED 被導通點亮。

因此，LED 的亮滅情形反映了馬達各相線圈的通電狀態。

(2) 若馬達驅動板不接電，但用手轉動馬達，LED 燈亦會亮，請解釋這個現象。

這是因為步進馬達在本質上是多組線圈組成的電磁結構，當手動旋轉馬達軸時，內部的磁場相對於繞線移動，產生感應電動勢。這個感應電壓會在對應的線圈中產生微小電流，而該電流會經由 ULN2003A 的導通路徑流經 LED，使 LED 短暫發光。

此現象本質上是法拉第電磁感應的應用。

(3) 請問在每次馬達的相位變化間加入 delay 與否會造成馬達轉動情形如何？

步進馬達仰賴各相位依序正確切換來完成轉動。若相位切換過快（即無 delay），馬達來不及對電磁變化產生機械反應，可能導致：

1. 馬達不轉動或抖動
2. 漏步（失步）現象，造成轉動角度錯誤
3. 轉矩不足或無法啟動

加入適當 delay 可確保馬達有足夠時間完成每一步動作，運轉更穩定。

因此，delay 是確保準確轉動與穩定驅動的關鍵因素。

五、程式碼與註解：

基本題：

ORG 0000H

JMP START ; 開機時跳轉到主程式開始執行

ORG 0050H ; 主程式起始位址

START:

MOV R0, #150 ; 設定 R0 為 150，代表順時鐘轉動的步數

FWD_LOOP:

MOV P0, #04H ; 啟動 P0.2（第1相 A）→ 控制馬達前進第一步

CALL DELAY ; 呼叫延遲，讓馬達有時間轉動

MOV P0, #02H ; 啟動 P0.1（第2相 B）

CALL DELAY

MOV P0, #01H ; 啟動 P0.0（第3相 C）

CALL DELAY

MOV P0, #08H ; 啟動 P0.3（第4相 D）

CALL DELAY

DJNZ R0, FWD_LOOP ; R0 減一，若未到 0 則重複 FWD_LOOP，順時鐘轉動

MOV R0, #150 ; 設定 R0 為 150，準備逆時鐘轉動

REV_LOOP:

MOV P0, #01H ; 啟動 P0.0（第3相 C）→ 反方向轉第一步

CALL DELAY

MOV P0, #02H ; 啟動 P0.1（第2相 B）

CALL DELAY

MOV P0, #04H ; 啟動 P0.2（第1相 A）

CALL DELAY

MOV P0, #08H ; 啟動 P0.3（第4相 D）

CALL DELAY

MOV P0, #01H ; 再次重複反方向相位順序


```

CALL DELAY
MOV P0, #02H
CALL DELAY
MOV P0, #04H
CALL DELAY
MOV P0, #08H
CALL DELAY
DJNZ R0, REV_LOOP ; R0 減一，若未到 0 則繼續逆時鐘轉動

```

STOP:

```

SJMP START          ; 回到 START，持續重複順時鐘→逆時鐘動作

```

;----- 延遲副程式 -----

DELAY:

```

MOV R6, #20D        ; 外層迴圈計數器

```

DELAY1:

```

MOV R7, #20D        ; 中層迴圈計數器

```

DELAY2:

```

MOV R5, #25D        ; 內層迴圈計數器

```

DELAY3:

```

DJNZ R5, DELAY3     ; 最內層延遲
DJNZ R7, DELAY2     ; 回到中層
DJNZ R6, DELAY1     ; 回到外層
RET                 ; 延遲結束，返回主程式

```

END ; 程式結束

進階題:

```

ORG 00H
JMP MAIN          ; 程式啟動入口，跳轉至主程式 MAIN

```

```

ORG 50H          ; 主程式的實際儲存位置為 0050H 開始

```

MAIN:

;===== 小鍵盤掃描階段：列(row)輸出低電位，行(col)讀入 =====

ROW1:

```

MOV P1, #7FH      ; 將第1列 (P1.7) 設為 0，其餘為1
CALL DELAY
MOV A, P1
ANL A, #0FH       ; 只保留低4位（行輸入）
MOV R1, #0        ; 記錄目前掃描的是第1列，基底為0

```

CJNE A,#0FH,COL1 ; 若有按鍵被按下，跳至 COL1 判斷是哪一行

ROW2:

MOV P1,#0BFH ; 第2列 (P1.6) 為 0，其餘為1
CALL DELAY
MOV A,P1
ANL A,#0FH
MOV R1,#4 ; 記錄目前列的基底為4（每列+4）
CJNE A,#0FH,COL1

ROW3:

MOV P1,#0DFH ; 第3列 (P1.5) 為 0
CALL DELAY
MOV A,P1
ANL A,#0FH
MOV R1,#8
CJNE A,#0FH,COL1

ROW4:

MOV P1,#0EFH ; 第4列 (P1.4) 為 0
CALL DELAY
MOV A,P1
ANL A,#0FH
MOV R1,#12
CJNE A,#0FH,COL1
JMP ROW1 ; 若沒偵測到按鍵，回到ROW1繼續掃描

; ===== 行(column)判斷，依據按鍵位置加總出鍵值 =====

COL1:

CJNE A,#0EH,COL2 ; 若 P1.3 為低電位，代表第1行
MOV R0,#0
JMP SHOW

COL2:

CJNE A,#0DH,COL3 ; 第2行
MOV R0,#1
JMP SHOW

COL3:

CJNE A,#0BH,COL4 ; 第3行
MOV R0,#2
JMP SHOW

COL4:

```

CJNE A,#07H,ROW1    ; 第4行；無效按鍵則重新掃描
MOV R0,#3
JMP SHOW

```

;===== 計算按鍵對應編號，並決定旋轉方向 =====

SHOW:

```

MOV A, R1
ADD A, R0            ; 鍵值 = 列基底 + 行值
MOV R4, A            ; R4 存鍵值
INC A                ; 增加1做減法比較
CLR C                ; 清除進位標誌
SUBB A, #9           ; 若結果 < 9 則 CY=1 (鍵值小於9)
JNC SCOUNTERCLOCKWISE ; 若鍵值 >=9 (CY=0)，跳轉逆時針方向

```

;===== 順時針方向 =====

SCLOCKWISE:

```

MOV DPTR,#TABLE_1    ; 根據鍵值查表，取得各激磁模式次數
MOV A,R4
MOVC A,@A+DPTR
MOV R0,A              ; R0: 1相激磁次數

```

```

MOV DPTR,#TABLE_2
MOV A,R4
MOVC A,@A+DPTR
MOV R1,A              ; R1: 2相激磁次數

```

```

MOV DPTR,#TABLE_12
MOV A,R4
MOVC A,@A+DPTR
MOV R2,A              ; R2: 1-2相激磁次數

```

CLOCKWISE:

```

ACALL RTURN1          ; 執行 1 相激磁 (順時針)
ACALL DELAY
ACALL RTURN2          ; 執行 2 相激磁 (順時針)
ACALL DELAY
ACALL RTURN12         ; 執行 1-2 相激磁 (順時針)
ACALL DELAY
JMP MAIN              ; 回主迴圈重新等待按鍵

```

;===== 逆時針方向 =====

SCOUNTERCLOCKWISE:

```
MOV DPTR,#TABLE_1
MOV A,R4
MOVC A,@A+DPTR
MOV R0,A
```

```
MOV DPTR,#TABLE_2
MOV A,R4
MOVC A,@A+DPTR
MOV R1,A
```

```
MOV DPTR,#TABLE_12
MOV A,R4
MOVC A,@A+DPTR
MOV R2,A
```

COUNTERCLOCKWISE:

```
ACALL LTURN1          ; 執行 1 相激磁 (逆時針)
ACALL DELAY
ACALL LTURN2          ; 執行 2 相激磁 (逆時針)
ACALL DELAY
ACALL LTURN12         ; 執行 1-2 相激磁 (逆時針)
ACALL DELAY
JMP MAIN
```

;===== 各種轉動子程序 =====

;--- 1 相激磁 (順時針) ---

RTURN1:

```
MOV A,#00010001B
MOV R3,#0
CJNE R0,#0,RTURN1_LOOP
RET
```

RTURN1_LOOP:

```
MOV P0,A
RL A
CALL DELAY
INC R3
CJNE R3,#65,RTURN1_LOOP
MOV R3,#0
DEC R0
JMP RTURN1
```

; --- 2 相激磁（順時針） ---

RTURN2:

MOV A,#00110011B

MOV R3,#0

CJNE R1,#0,RTURN2_LOOP

RET

RTURN2_LOOP:

MOV P0,A

RL A

CALL DELAY

INC R3

CJNE R3,#130,RTURN2_LOOP

MOV R3,#0

DEC R1

JMP RTURN2

; --- 1-2 相激磁（順時針） ---

RTURN12:

MOV R3,#0

CJNE R2,#0,RTURN12_LOOP

RET

RTURN12_LOOP:

MOV P0,#0001B

CALL DELAY

MOV P0,#0011B

CALL DELAY

MOV P0,#0010B

CALL DELAY

MOV P0,#0110B

CALL DELAY

MOV P0,#0100B

CALL DELAY

MOV P0,#1100B

CALL DELAY

MOV P0,#1000B

CALL DELAY

INC R3

CJNE R3,#18,RTURN12_LOOP

MOV R3,#0

DEC R2

JMP RTURN12

; --- 1 相激磁（逆時針） ---

LTURN1:

MOV A,#00010001B

MOV R3,#0

CJNE R0,#0,LTURN1_LOOP

RET

LTURN1_LOOP:

MOV P0,A

RR A

CALL DELAY

INC R3

CJNE R3,#65,LTURN1_LOOP

MOV R3,#0

DEC R0

JMP LTURN1

; --- 2 相激磁（逆時針） ---

LTURN2:

MOV A,#00110011B

MOV R3,#0

CJNE R1,#0,LTURN2_LOOP

RET

LTURN2_LOOP:

MOV P0,A

RR A

CALL DELAY

INC R3

CJNE R3,#130,LTURN2_LOOP

MOV R3,#0

DEC R1

JMP LTURN2

; --- 1-2 相激磁（逆時針） ---

LTURN12:

MOV R3,#0

CJNE R2,#0,LTURN12_LOOP

RET

LTURN12_LOOP:

MOV P0,#0001B


```

CALL DELAY
MOV P0,#1001B
CALL DELAY
MOV P0,#1000B
CALL DELAY
MOV P0,#1100B
CALL DELAY
MOV P0,#0100B
CALL DELAY
MOV P0,#0110B
CALL DELAY
MOV P0,#0010B
CALL DELAY
MOV P0,#0011B
CALL DELAY
INC R3
CJNE R3,#18,LTURN12_LOOP
MOV R3,#0
DEC R2
JMP LTURN12

```

; --- 延遲副程式 ---

DELAY:

```
MOV R6,#255
```

DELAY1:

```
MOV R7,#250
```

DELAY2:

```
DJNZ R7,DELAY2
```

```
DJNZ R6,DELAY1
```

```
RET
```

; --- 馬達轉動次數查表 ---

TABLE_1:

```
DB 4,0,0,15
```

```
DB 18,0,0,0
```

```
DB 4,0,0,15
```

```
DB 18,0,0,0
```

TABLE_2:

```
DB 0,4,0,0
```

```
DB 0,12,0,0
```

```
DB 0,4,0,0
```

```
DB 0,12,0,0
TABLE_12:
DB 0,0,12,0
DB 0,0,26,30
DB 0,0,12,0
DB 0,0,26,30
```

END

六、心得：

1. 上課心得:

透過課堂講解，我對步進馬達的控制理論、激磁方式，以及其在實務應用中的重要性有了完整認識。老師對 1 相、2 相與 1-2 相激磁特性的比較，讓我清楚理解不同控制方式在轉矩、功耗與精準度上的差異。

2. 實驗心得:

實驗中實作小鍵盤掃描與步進馬達控制，讓我實際應用了課堂上學到的內容。從鍵盤行列掃描邏輯設計，到根據按鍵編號選擇旋轉方向與角度，整個過程訓練了我對程式結構與硬體控制的綜合能力。雖然一開始在給足馬達電壓上頭疼了很久，最後換了 ic 板才得以解決。

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