

# 儀表電子能力本位訓練教材 PIC 語法應用

編號：PEN-IMT1015

編著者：董勝源

審稿者：吳經文、杜日富

主辦單位：行政院勞工委員會職業訓練局

研製單位：中華民國職業訓練研究發展中心

印製日期：九十年十二月

## 單元 PEN-IMT1015 學習指引

當你在學習本單元之前，你必須熟悉下列之知識：認識 PIC 單晶片微電腦，PIC 指令介紹、如何組譯 PIC 程式及認識與操作 PIC 模擬器的工作。

- 1、如果你都能夠勝任，請按下列之指示進行學習：
- 2、如果你不會認識 PIC 單晶片微電腦，請由編號： PEN-IMT1011 教材開始學習，或請教你的老師。
- 3、如果你不會 PIC 指令介紹，請由編號： PEN-IMT1012 教材開始學習，或請教你的老師。
- 4、如果你不會如何組譯 PIC 程式，請由編號： PEN-IMT1013 教材開始學習，或請教你的老師。
- 5、如果你不會認識與操作 PIC 模擬器的工作，請由編號： PEN-IMT1014 教材開始學習，或請教你的老師。

## 引言

本章將應用「類 8051」的指令來撰寫程式。此「類 8051」的指令用法和 MCS-51 指令大致雷同，因此有關 MCS-51 的應用程式，只要配合 RISC 的單晶片的特性，給予少許的修改，即可應用在 RISC 的單晶片上。在撰寫程式時，甚致較 RISC 及 MCS-51 的寫法還簡易，不過缺點是須要較多的記憶體才能完成同樣的動作。

## 定義

各名詞的定義在本單元中將會介紹。

## 學習目標

- 一、學員能在具備指令表之下，寫出基本的資料的搬移及輸出動作程式。
- 二、學員能在具備指令表之下，寫出算術動作程式。
- 三、學員能在具備指令表之下，寫出邏輯動作輸出程式。
- 四、學員能在具備指令表之下，寫出列表法輸出程式。
- 五、學員能在具備指令表之下，寫出輸入動作程式。
- 六、學員能在具備指令表之下，寫出執行時間計算程式。
- 七、學員能在具備指令表之下，寫出大容量程式及 PIC16C84 與 PIC16C5X 程式的轉換。

## 學習活動

本單元之學習活動在學習 PIC 指令之前，必須先學習認識 PIC 單晶片微電腦的內部硬體電路的結構路及認識與操作 PIC 電腦模擬器的其相關知識，你可以由下列途徑去學習。

- (一)儀表電子教材，單元編號：PEN-IMT1011、PEN-IMT1012、PEN-IMT1013 及 PEN-IMT1014 的內容。
- (二)如果你已經瞭解上述的相關知識，請你由本教材之第 11 頁開始進行實際操作。

本單元的第一個學習目標是：

學員能在具備指令表之下，寫出基本的資料的搬移及輸出動作程式。

## 一、資料的搬移及輸出動作實習

組合語言中的搬移指令只是從事拷貝的動作而已，原有的資料不變。如指令“MOV W,fr”是將 fr 的資料多拷貝一份到 W。

如果對 I/O 暫存器讀寫或搬移，即是進行輸出入的動作。常用的指令如下表。

搬移及輸出動作指令

指令	字元數	影響旗標	隱含 W	說明
搬移指令				
MOV W,#literal	1			W←常數
MOV W,fr	1	Z		W←fr
MOV fr,W	1			fr←W
MOV fr,#literal	2	W	隱含 W	fr←常數
MOV fr1,fr2	2	Z	隱含 W	fr1←fr2
MOV W,<>fr	1			W←fr0-3 和 fr4-7 交換
SWAP fr	1			fr←fr0-3 和 fr4-7 交換
清除指令				
CLR W	1	Z		W=0
CLR fr	1	Z		fr=0

本節實習請讀者先進行軟體模擬，其中請將“CALL DELAY”指令前面加上“；”號使該指令失效，以免進入空回轉，會浪費很多時間。實習項目如下：

- ◎ 資料搬移動作實習
- ◎ 輸出動作實習
- ◎ 間接資料搬移動作實習

## (一)、資料搬移動作實習

1、動作：將資料經 W 送入一般暫存器，再令資料的前後四位元互換。

2、程式：2-1-1.ASM

```

;***** 2-1-1.ASM *****
;
;
;
; #05H → [W: 05H] → [10H: 50H]
;
;
;*****
                DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;*****
                ORG      00H          ;啓始位址
                MOV     W,#05        ;W <-- 資料 05
                MOV     10H,W        ;暫存器 <--W
LOOP
                SWAP    10H          ;暫存器的前後四位元互換
                JMP     LOOP         ;跳到 LOOP 處
;*****
                END

```

## (二)、輸出動作實習

在進行輸出入時，必須先設定其內部的三態電路，令其接腳為輸出或輸入。

1、動作：將資料直接送入 I/O 暫存器輸出。

2、程式：2-1-2.ASM

```

;*****2-1-2.ASM *****
;
;
;
; #data  → RA
;
;
; #data  → RB
;
;
;*****
                DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;*****
                ORG      00H      ;啓始位址
START
                MOV     !RA,#00   ;設定 RA 接腳為輸出
                MOV     !RB,#00   ;設定 RB 接腳為輸出
LOOP
                CLR     RA        ;清除 RA=0
                CLR     RB        ;清除 RB=0
                MOV     RA,#01H   ;令 RA 輸出資料 01H
                MOV     RB,#23H   ;令 RB 輸出資料 23H

                MOV     RA,#04H   ;令 RA 輸出資料 04H
                MOV     RB,#56H   ;令 RB 輸出資料 56H

                MOV     RA,#07H   ;令 RA 輸出資料 07H
                MOV     RB,#89H   ;令 RB 輸出資料 89H

                JMP     LOOP      ;跳到 LOOP 處
;*****
                END

```

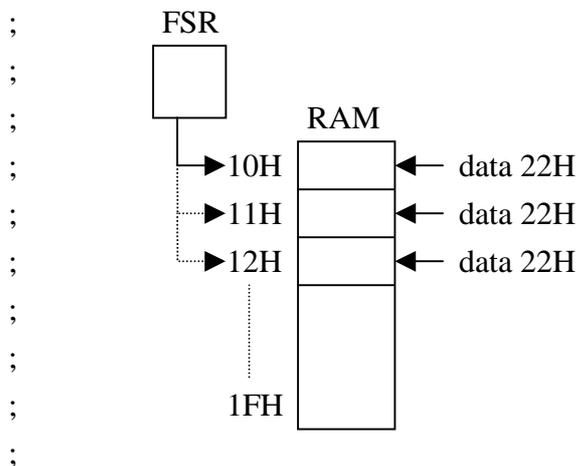
## (三)、間接資料搬移動作實習

在 PIC16C84 中的間接資料的搬移則必須由『間接資料指標暫存器[INDF]』及『檔案選擇[INDF]』為一個 8 位元的間接資料指標暫存器，本身並非可存取的暫存器，它是被用來代表指向另一個暫存器。只要將指定暫存器的位址放入 [FSR] 中，[INDF] 就會變成被指定的暫存器。

例如設定 [FSR]=20H，則對 [INDF] 來作存取，也就是對通用暫存器 20H 來作存取。這就是「間接暫存器」的資料存取。

- 1、動作：將指定的資料以「間接」方式，存入 RAM 10H~1FH 內。
- 2、程式：2-1-3.ASM

```
***** 2-1-3.ASM *****
```



```
DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
```

```
*****
```

```
ORG 00H ;啓始位址
```

```
START
```

```
MOV FSR,#10H ;指定檔案選擇為暫存器 10H
```

```
LOOP MOV INDF,#22H ;資料以間接方式存入指定的暫存器
```

```
INC FSR ;選擇下一個暫存器
```

```
CJNE FSR,#20H,LOOP ;暫存器選擇未完則跳到 LOOP 處
```

```
JMP $ ;自轉，停止執行
```

```
*****
```

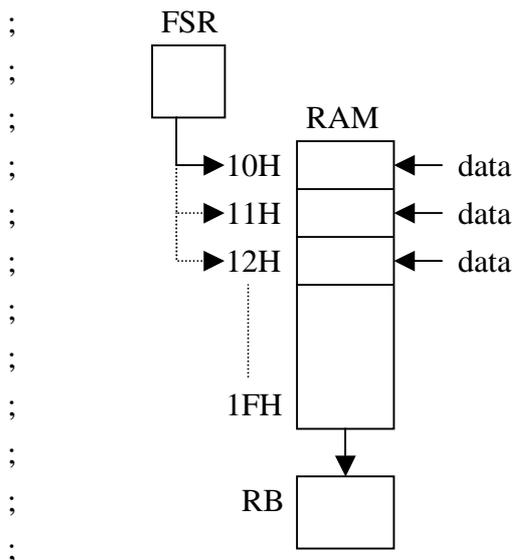
```
END
```

## (四)、間接資料搬移及輸出動作實習

1、動作：將不同的資料以「間接」方式，存入 RAM 10H~1FH 內，再分別以「間接」方式取出由 RB 輸出。

2、程式：2-1-4.ASM

```
;***** 2-1-4.ASM *****
```



```
DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
```

```
;*****
```

```
ORG 0CH ;一般暫存器啓始位址
```

```
COUNT DS 1 ;定義計數暫存器
```

```
;*****
```

```
ORG 00H ;程式啓始位址
```

```
MOV !RB,#0 ;設定 RB 爲輸出
```

```
START
```

```
MOV COUNT,#0 ;計數資料=0
```

```
MOV FSR,#10H ;指定檔案選擇爲暫存器 10H
```

```
LOOP MOV INDF,COUNT ;資料以間接方式存入指定的暫存器
```

```
INC COUNT ;計數資料加一
```

```
INC FSR ;選擇下一個暫存器
```

```
CJNE FSR,#20H,LOOP ;暫存器選擇未完則跳到 LOOP 處，換下一筆存入
```

```
;-----
```

```
READ_DATA
```

```
MOV FSR,#10H ;指定檔案選擇爲暫存器 10H
```

```
LOOP1
```

```
MOV RB,INDF ;指定暫存器資料以間接方式讀出，由 RB 輸出
```

[請翻至下一頁](#)

```

CALL    DELAY           ;呼叫延時副程式
INC     FSR             ;選擇下一個暫存器
CJNE    FSR,#20H,LOOP1 ;暫存器選擇未完則跳到 LOOP1 處，換下一筆讀取
JMP     READ_DATA      ;若暫存器選完則跳到 READ_DATA 處，重覆讀取
;*****
DELAY                                       ;延時 0.2 秒副程式
        MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL     MOV     DELY2,#0FFH
        DJNZ   DELY2,$
        DJNZ   DELY1,DEL
        RET                                ;回主程式
;*****
END

```

## 學習評量一：

1、如下列程式，最後 RAM 20H 的內容為何？

```

;*****
;
    DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;*****
    ORG     00H
    MOV     W,#23H
    MOV     20H,W
    SWAP    20H
LOOP
    JMP     LOOP
;*****
    END

```

2、如下列程式，最後 RA 和 RB 的輸出內容為何？

```

;*****
;
    DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;*****
    ORG     00H
START
    MOV     !RA,#00
    MOV     !RB,#00
LOOP
    CLR     RA
    CLR     RB
    MOV     RA,#01H
    MOV     RB,#23H

    MOV     RA,#04H
    MOV     RB,#56H

    MOV     RA,#33H
    MOV     RB,#55H
    JMP     LOOP
;*****
    END

```

3、如下列程式，最後 RAM 20H-2FH 的內容為何?

```
;*****  
;          DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF  
;*****  
          ORG      00H  
START  
          MOV      FSR,#20H  
LOOP     MOV      INDF,#45H  
          INC      FSR  
          CJNE    FSR,#30H,LOOP  
          JMP      $  
;*****  
          END
```

## 學習評量一答案：

1、如下列程式，最後 RAM 20H 的內容為 32H。

```

;*****
;
    DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;*****
;
    ORG      00H
    MOV      W,#23H
    MOV      20H,W
    SWAP     20H

LOOP
    JMP      LOOP
;*****
;
    END

```

2、如下列程式，最後的輸出內容 RA=33H 和 RB=56H。

```

;*****
;
    DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;*****
;
    ORG      00H

START
    MOV      !RA,#00
    MOV      !RB,#00

LOOP
    CLR      RA
    CLR      RB
    MOV      RA,#01H
    MOV      RB,#23H

    MOV      RA,#04H
    MOV      RB,#56H

    MOV      RA,#33H
    MOV      RB,#55H
    JMP      LOOP
;*****
;
    END

```

3、如下列程式，最後 RAM 20H-2FH 的內容均為 45H。

```
*****  
;  
    DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF  
*****  
;  
    ORG    00H  
START  
    MOV    FSR,#20H  
LOOP  MOV    INDF,#45H  
    INC    FSR  
    CJNE   FSR,#30H,LOOP  
    JMP    $  
*****  
;  
    END
```

---

假如你的答案與上述之重點相似，請翻下一頁，假如你的答案不與上述之重點相似，請翻至第 4 頁重新閱讀以便發現你的錯誤之處，並將錯誤改正。

本單元的第二個學習目標是：

學員能在具備指令表之下，寫出算術動作程式。

## 二、 算術動作實習

PIC16C84 所有的暫存器及 RAM (位址 00~2F 及 80~AF)均可直接和 ALU 作算術及邏輯運算。

在進行算術動作時，必須特別注意旗標位元的變化，它會影響執行的結果，我們可由這些旗標位元的變化，用位元分支指令來控制程式。如下表。

常用算術動作及位元分支的指令

指令	字元數	影響旗標	隱含 W	說明
遞加與遞減指令				
INC fr	1	Z		$fr \leftarrow fr+1$
DEC fr	1	Z		$fr \leftarrow fr-1$
MOV W,++fr	1	Z		$W \leftarrow fr+1$
MOV W,--fr	1	Z		$W \leftarrow fr-1$
加法指令				
ADD W,#literal	1	C,DC,Z		$W \leftarrow W+常數$
ADD W,fr	1	C,DC,Z		$W \leftarrow fr+常數$
ADD fr,W	1	C,DC,Z		$fr \leftarrow W+常數$
ADD fr,#literal	2	C,DC,Z	隱含 W	$fr \leftarrow fr+常數$
ADD fr1,fr2	2	C,DC,Z	隱含 W	$fr1 \leftarrow fr1+fr2$
減法指令				
SUB fr,W	1	C,DC,Z		$fr \leftarrow fr-W$
SUB fr,#literal	2	C,DC,Z	隱含 W	$fr \leftarrow fr-常數$
SUB fr1,fr2	2	C,DC,Z	隱含 W	$fr1 \leftarrow fr1-fr2$
MOV W,#literal-W	1	C,DC,Z		$W \leftarrow 常數-W$
MOV W,fr-W	1	C,DC,Z		$W \leftarrow fr-W$
位元分支指令				
SB bit	1			若 bit=1，則跳過下一行
SC	1			若 C=1，則跳過下一行
SZ	1			若 Z=1，則跳過下一行
SNB bit	1			若 bit=0，則跳過下一行
SNC	1			若 C=0，則跳過下一行
SNZ	1			若 Z=0，則跳過下一行
JB bit,addr9	2			若 bit=1，則跳到 addr9
JC addr9	2			若 C=1，則跳到 addr9
JZ addr9	2			若 Z=1，則跳到 addr9
JNB bit_addr,addr9	2			若 bit=0，則跳到 addr9
JNC addr9	2			若 C=0，則跳到 addr9
JNZ addr9	2			若 Z=0，則跳到 addr9

另外和關算術動作有關的是「比較指令」，它是將兩者做相減運算，再由所影響的旗標 Z、C 及 DC 來判斷是否大於(>)、小於(<) 或等於(=)，但不改變原有暫存器的內容。如下表。

比較動作的指令

指 令	字元數	影響旗標	隱含 W	說 明
CSA fr,#literal	3	C,DC,Z	隱含 W	若 fr > 常數，則跳過下一行
CSA fr1,fr2	3	C,DC,Z	隱含 W	若 fr1 > fr2，則跳過下一行
CSAE fr,#literal	3	C,DC,Z	隱含 W	若 fr >= 常數，則跳過下一行
CSAE fr1,fr2	3	C,DC,Z	隱含 W	若 fr1 >= fr2，則跳過下一行
CSB fr,#literal	3	C,DC,Z	隱含 W	若 fr < 常數，則跳過下一行
CSB fr1,fr2	3	C,DC,Z	隱含 W	若 fr1 < fr2，則跳過下一行
CSBE fr,#literal	3	C,DC,Z	隱含 W	若 fr < 常數，則跳過下一行
CSBE fr1,fr2	3	C,DC,Z	隱含 W	若 fr1 <= fr2，則跳過下一行
CSE fr,#literal	3	C,DC,Z	隱含 W	若 fr = 常數，則跳過下一行
CSE fr1,fr2	3	C,DC,Z	隱含 W	若 fr1 = fr2，則跳過下一行
CSNE fr,#literal	3	C,DC,Z	隱含 W	若 fr <> 常數，則跳過下一行
CSNE fr1,fr2	3	C,DC,Z	隱含 W	若 fr1 <> fr2，則跳過下一行
CJA fr,#literal,addr9	4	C,DC,Z	隱含 W	若 fr > 常數，則跳到 addr9
CJA fr1,fr2,addr9	4	C,DC,Z	隱含 W	若 fr1 > fr2，則跳到 addr9
CJAE fr,#literal,addr9	4	C,DC,Z	隱含 W	若 fr >= 常數，則跳到 addr9
JAE fr1,fr2,addr9	4	C,DC,Z	隱含 W	若 fr1 >= fr2，則跳到 addr9
CJB fr,#literal,addr9	4	C,DC,Z	隱含 W	若 fr < 常數，則跳到 addr9
JB fr1,fr2,addr9	4	C,DC,Z	隱含 W	若 fr1 < fr2，則跳到 addr9
CJBE fr,#literal,addr9	4	C,DC,Z	隱含 W	若 fr <= 常數，則跳到 addr9
CJBE fr1,fr2,addr9	4	C,DC,Z	隱含 W	若 fr1 <= fr2，則跳到 addr9
CJE fr,#literal,addr9	4	C,DC,Z	隱含 W	若 fr = 常數，則跳到 addr9
CJE fr1,fr2,addr9	4	C,DC,Z	隱含 W	若 fr1 = fr2，則跳到 addr9
CJNE fr,#literal,addr9	4	C,DC,Z	隱含 W	若 fr <> 常數，則跳到 addr9
CJNE fr1,fr2,addr9	4	C,DC,Z	隱含 W	若 fr1 <> fr2，則跳到 addr9

本節進行的實習項目如下：

- ◎ 遞加及遞減動作實習：8~16 位元遞加/遞減計數
- ◎ 加法動作實習：BCD 碼由 0000~1999 遞加計數。
- ◎ 減法動作實習：BCD 碼由 1999 ~0000 遞減計數。

## (一)、遞加及遞減動作實習

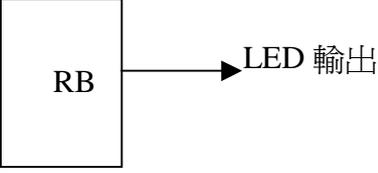
在進行遞加及遞減指令時，結果若為“0”，會令  $Z = 0$ ，其餘不受影響。

## 1、8 位元遞減計數實習

(1)、動作：令 RB 計數由 FF~00。

(2)、程式：2-2-1.ASM

```

;*****2-2-1.ASM *****
;
;   PIC16C84
;
;   
;
;
;
;
;
;
;
;
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG     0CH ;使用者 RAM 啓始位址
DELY1  DS    1 ;延時暫存器
DELY2  DS    1 ;延時暫存器
;*****
;
;   ORG     00H ;程式啓始位址
START
;   MOV     !RB,#00 ;設 RB 爲輸出 ---(1)
;   CLR     RB ;清除埠 B 輸出 ---(2)
LOOP   CALL  DELAY ;呼叫延時副程式-(3)
;   DEC     RB ;埠 B 減 1 ---(4)
;   JMP     LOOP ;跳到 LOOP 處重覆輸出(5)
;*****
;
;   DELAY ;延時 0.2 秒副程式
;   MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL     MOV     DELY2,#0FFH
;   DJNZ    DELY2,$
;   DJNZ    DELY1,DEL
;   RET ;回主程式
;*****
;
;   END

```





## (二)、加法動作實習

在進行加法時，所影響的旗標如下：

◎若加法的結果為“0”，會令  $Z = 1$ 。

◎若加法的結果，有中間進位時，會令  $DC = 1$ 。

◎若加法的結果，有進位時，會令  $C = 1$ 。

如：

$$\begin{array}{r} 00000111 \\ + 00000001 \\ \hline 00001000 \end{array}$$

旗標位元	動作	原因
Z	0	結果不為0
DC	0	無中間進位
C	0	無進位

如：

$$\begin{array}{r} 00001111 \\ + 00000001 \\ \hline 00010000 \\ \quad \uparrow \text{中間進位} \\ \quad \square \end{array}$$

旗標位元	動作	原因
Z	0	結果不為0
DC	1	有中間進位
C	0	無進位

如：

$$\begin{array}{r} 11111111 \\ + 00000001 \\ \hline 00000000 \\ \text{進位} \quad \uparrow \quad \uparrow \quad \text{中間進位} \\ \quad \square \quad \square \end{array}$$

旗標位元	動作	原因
Z	1	結果為0
DC	1	有中間進位
C	1	有進位



```

ADD    TEMP,#60H    ;十位數加 6
JNC    Q2           ;若未超過 9，C=0，跳到 Q2
MOV    RB,#0        ;若有超過 9，C=1，RB=0
ADD    RA,#01       ;若 RB 超過 99，RA+1    ;---(7)
MOV    TEMP,RA
ADD    TEMP,#06     ;百位數加 6            ;---(8)
JNB    DC,Q2        ;若未超過 9，DC=0，跳到 Q2
ADD    RA,#06       ;若有超過 9，DC=1，RA+6

```

Q2

```

RET                ;回主程式

```

```

;*****

```

```

DELAY                ;延時 0.2 秒副程式

```

```

MOV    DELY1,#0FFH  ;自轉 255*255 圈

```

```

DEL    MOV    DELY2,#0FFH

```

```

DJNZ   DELY2,$

```

```

DJNZ   DELY1,DEL

```

```

RET                ;回主程式

```

```

;*****

```

```

END

```

## (三)、減法動作實習

在進行減法時，所影響的旗標如下：

◎結果若為“0”，會令  $Z = 1$ 。

◎結果若有中間借位時，會令  $DC = 0$ 。

◎結果若有借位時，會令  $C = 0$ 。

如：

```

  1 1 1 1 1 1 1 1
-
  0 0 0 0 0 0 0 1
-----
  1 1 1 1 1 1 1 0

```

旗標位元	動作	原因
Z	0	結果不為0
DC	1	無中間借位
C	1	無借位

如：

```

  1 1 1 1 0 0 0 0
-
  0 0 0 0 0 0 0 1
-----
  1 1 1 0 1 1 1 1
      | ↑ 中間借位
      □

```

旗標位元	動作	原因
Z	0	結果不為0
DC	0	有中間借位
C	1	無借位

如：

```

  1 1 1 1 0 0 0 0
-
  1 1 1 1 0 0 0 0
-----
  0 0 0 0 0 0 0 0

```

旗標位元	動作	原因
Z	1	結果為0
DC	1	無中間借位
C	1	無借位

如：

```

  1 1 1 1 0 0 0 0
-
  1 1 1 1 0 0 0 1
-----
  1 1 1 1 1 1 1 1
借位 | ↑      | ↑ 中間借位
      □      □

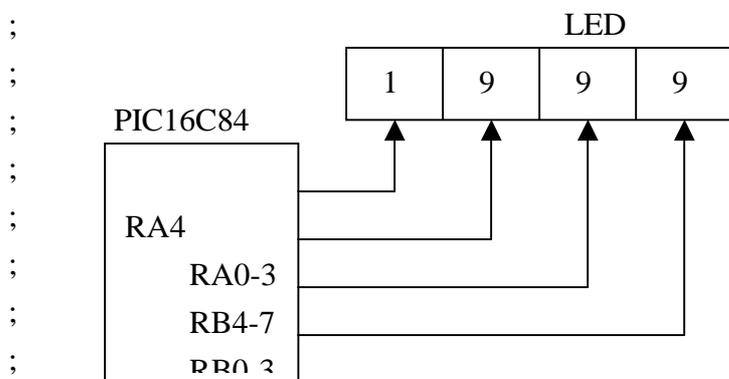
```

旗標位元	動作	原因
Z	0	結果不為0
DC	0	有中間借位
C	0	有借位

動作：令 RA 及 RB 計數以 BCD 碼由 1999 ~0000 進行遞減工作。

1、程式：2-2-5.ASM

```
*****2-2-5.ASM *****
```



```
*****
```

```
DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
```

```
ORG 0CH ;使用者 RAM 啓始位址
```

```
DELY1 DS 1 ;延時暫存器
```

```
DELY2 DS 1 ;延時暫存器
```

```
TEMP DS 1 ;暫時暫存器
```

```
*****
```

```
ORG 00H ;程式啓始位址
```

```
START
```

```
MOV !RA,#00 ;設 RA 爲輸出
```

```
MOV !RB,#00 ;設 RB 爲輸出
```

```
MOV RA,#19H ;RA=19
```

```
MOV RB,#99H ;RB=99
```

```
LOOP
```

```
CLRB DC ;DC=0
```

```
CLRB C ;C=0
```

```
SUB RB,#01 ;RB=RB-1
```

```
CALL CONV10 ;呼叫轉換 10 進制的副程式
```

```
CALL DELAY ;呼叫延時副程式
```

```
JMP LOOP ;跳到 LOOP 處重覆輸出
```

```
*****
```

```
CONV10 ;轉換 10 進制的副程式
```

```
JB DC,Q2 ;相減若無中間借位，DC=1，跳到 Q2
```

```
JNC Q1 ;若有中間借位，再檢查是否有借位，若有 C=0，跳到 Q1
```

```
SUB RB,#6 ;有中間借位，而無借位，則 RB-6
```

```
RET ;回主程式
```

Q1

```

MOV    RB,#99H    ;有中間借位及有借位，則 RB=99
SUB    RA,#01     ;RA=RA-1
JB     DC,Q2      ;相減若無中間借位，DC=1，跳到 Q2
SUB    RA,#06     ;若有中間借位，則 RA-6

```

Q2

```

RET                    ;回主程式

```

```

;*****

```

```

DELAY                    ;延時 0.2 秒副程式

```

```

MOV    DELY1,#0FFH    ;自轉 255*255 圈

```

```

DEL    MOV    DELY2,#0FFH

```

```

DJNZ   DELY2,$

```

```

DJNZ   DELY1,DEL

```

```

RET                    ;回主程式

```

```

;*****

```

```

END

```

## 學習評量二：

1、如下列程式 RB 會有何動作?

```

;*****
        DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
        ORG     0CH
DELY1   DS     1
DELY2   DS     1
;*****
        ORG     00H
START
        MOV     !RB,#00
        CLR     RB
LOOP    CALL    DELAY
        INC     RB
        JMP     LOOP
;*****
DELAY
        MOV     DELY1,#0FFH
DEL     MOV     DELY2,#0FFH
        DJNZ   DELY2,$
        DJNZ   DELY1,DEL
        RET
;*****
        END

```

2、如下列程式 RB 會有何動作?

```

;*****
        DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
        ORG     0CH
DELY1   DS     1
DELY2   DS     1
;*****

        ORG     00H
START
        MOV     !RB,#00
        CLR     RB
LOOP    CALL    DELAY
        DEC     RB
        JZ      $
        JMP     LOOP
;*****

DELAY
        MOV     DELY1,#0FFH
DEL     MOV     DELY2,#0FFH
        DJNZ    DELY2,$
        DJNZ    DELY1,DEL
        RET
;*****

        END

```

3、如下列程式 RA 及 RB 會有何動作?

```

;*****
DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
    ORG    0CH
DELY1  DS    1
DELY2  DS    1
TEMP   DS    1
    ORG    00H
START
    MOV    !RA,#00
    MOV    !RB,#00
    CLR    RA
    CLR    RB
LOOP
    ADD    RB,#01
    CALL   CONVT10
    JMP    LOOP
;*****
CONVT10
    MOV    TEMP,RB
    ADD    TEMP,#06
    JNB    DC,Q2
    ADD    RB,#06
Q1
    MOV    TEMP,RB
    ADD    TEMP,#60H
    JNC    Q2
    MOV    RB,#0
    ADD    RA,#01
    MOV    TEMP,RA
    ADD    TEMP,#06
    JNB    DC,Q2
    ADD    RA,#06
Q2
    RET
;*****
END

```

4、如下列程式 RA 及 RB 會有何動作？

```

;*****
DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
    ORG    0CH
DELY1  DS    1
DELY2  DS    1
TEMP   DS    1
;*****

    ORG    00H
START
    MOV    !RA,#00
    MOV    !RB,#00
    MOV    RA,#19H
    MOV    RB,#99H
LOOP
    CLRB   DC
    CLRB   C
    SUB    RB,#01
    CALL   CONVT 10
    JMP    LOOP
;*****
CONVT10
    JB     DC,Q2
    JNC    Q1
    SUB    RB,#6
    RET
Q1
    MOV    RB,#99H
    SUB    RA,#01
    JB     DC,Q2
    SUB    RA,#06
Q2
    RET
;*****
END

```

## 學習評量二答案：

1、如下列程式 RB 為由 00 至 FF 不斷的遞加動作。

```

;*****
        DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
        ORG     0CH
DELY1   DS     1
DELY2   DS     1
;*****
        ORG     00H
START
        MOV     !RB,#00
        CLR     RB
LOOP    CALL    DELAY
        INC     RB
        JMP     LOOP
;*****
DELAY
        MOV     DELY1,#0FFH
DEL     MOV     DELY2,#0FFH
        DJNZ    DELY2,$
        DJNZ    DELY1,DEL
        RET
;*****
        END

```

2、如下列程式會令 RB 由 FF 遞減至 00 即停止的動作。

```

;*****
        DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
        ORG     0CH
DELY1  DS     1
DELY2  DS     1
;*****

        ORG     00H
START
        MOV     !RB,#00
        CLR     RB
LOOP   CALL    DELAY
        DEC     RB
        JZ      $
        JMP     LOOP
;*****

DELAY
        MOV     DELY1,#0FFH
DEL    MOV     DELY2,#0FFH
        DJNZ    DELY2,$
        DJNZ    DELY1,DEL
        RET
;*****

        END

```

3、如下列程式 RA 及 RB 十進位的遞加動作。

```

;*****
DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF

        ORG    0CH
DELY1   DS    1
DELY2   DS    1
TEMP    DS    1
        ORG    00H
START
        MOV    !RA,#00
        MOV    !RB,#00
        CLR    RA
        CLR    RB
LOOP
        ADD    RB,#01
        CALL   CONV10
        JMP    LOOP
;*****
CONV10
        MOV    TEMP,RB
        ADD    TEMP,#06
        JNB   DC,Q2
        ADD    RB,#06
Q1
        MOV    TEMP,RB
        ADD    TEMP,#60H
        JNC   Q2
        MOV    RB,#0
        ADD    RA,#01
        MOV    TEMP,RA
        ADD    TEMP,#06
        JNB   DC,Q2
        ADD    RA,#06
Q2
        RET
;*****
END

```

4、如下列程式 RA 及 RB 十進位的遞減動作。

```

;*****
DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
        ORG      0CH
DELY1   DS      1
DELY2   DS      1
TEMP    DS      1
;*****
        ORG      00H
START
        MOV      !RA,#00
        MOV      !RB,#00
        MOV      RA,#19H
        MOV      RB,#99H
LOOP
        CLRB     DC
        CLRB     C
        SUB      RB,#01
        CALL     CONVT 10
        JMP      LOOP
;*****
CONVT10
        JB       DC,Q2
        JNC      Q1
        SUB      RB,#6
        RET
Q1
        MOV      RB,#99H
        SUB      RA,#01
        JB       DC,Q2
        SUB      RA,#06
Q2
        RET
;*****
END

```

假如你的答案與上述之重點相似，請翻下一頁，假如你的答案不與上述之重點相似，請翻至第 16 頁重新閱讀以便發現你的錯誤之處，並將錯誤改正。

本單元的第三個學習目標是：

學員能在具備指令表之下，寫出邏輯動作輸出程式。

### 三、 邏輯動作輸出實習

邏輯動作指令可針對位元或位元組來進行，此時資料產生特殊的變化，且它僅對 Z 旗標有影響。介紹如下：

◎NOT 指令：令位元組資料反相，範例如下：

```

      1 1 0 0 1 0 1 0   : 原始資料
NOT   ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
      0 0 1 1 0 1 0 1   : 結果
  
```

◎AND 指令：和 “0” 做 AND 動作時，會令某位元為 “0”，其餘保留，範例如下：

```

      1 1 0 0 1 0 1 0   : 原始資料
AND   ↓
      0 1 1 1 1 1 1 1   : 邏輯
-----
      0 1 0 0 1 0 1 0   : 結果
      ↑ | | | | | | | | 其餘保留
      | | | | | | | | 令原始資料位元為 “0”
  
```

◎OR 指令：和 “1” 做 OR 動作時，會令某位元為 “1”，其餘保留，範例如下：

```

      0 0 0 1 1 0 1 0   : 原始資料
OR   ↓
      1 0 0 0 0 0 0 0   : 邏輯
-----
      1 0 0 1 1 0 1 0   : 結果
      ↑ | | | | | | | | 其餘保留
      | | | | | | | | 令原始資料位元為 “1”
  
```

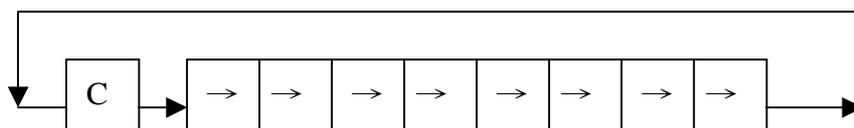
◎XOR 指令：和“1”做 XOR 動作時，會令某位元反相，其餘保留，範例如下：

```

      1 0 0 1 1 0 1 0    : 原始資料
XOR  ↓
      1 0 0 0 0 0 0 0    : 邏輯
-----
結果：0 0 0 1 1 0 1 0    : 結果
      ↑ | | | | | | |
      令原始資料位元為反相 其餘保留
  
```

◎旋轉指令：有右旋轉(RR)和左旋轉(RL)兩種指令，旋轉時，暫存器內的資料會連帶進位旗標(C)一起旋轉。

右旋轉(RR)



左旋轉(RL)



和邏輯動作有關的指令如下表：

### 邏輯指令

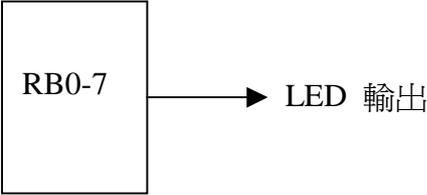
指令	字元數	影響旗標	隱含 W	說明
反相指令				
NOT W	1	Z		W←W 反相
MOV W,/fr	1	Z		W←fr 反相
NOT fr	1	Z		fr←fr 反相
及指令				
AND W,#literal	1	Z		W←W AND 常數
AND W,fr	1	Z		W←W AND fr
AND fr,W	1	Z		fr←fr AND W
AND fr,#literal	2	Z	隱含 W	fr←fr AND 常數
AND fr1,fr2	2	Z	隱含 W	fr1←fr1 AND fr2
或指令				
OR W,#literal	1	Z		W←W OR 常數
OR W,fr	1	Z		W←W OR fr
OR fr,W	1	Z		fr←fr OR W
OR fr,#literal	2	Z	隱含 W	fr←fr OR 常數
OR fr1,fr2	2	Z	隱含 W	fr1←fr1 OR fr2
互斥或指令				
XOR W,#literal	1	Z		W←W XOR 常數
XOR W,fr	1	Z		W←W XOR fr
XOR fr,W	1	Z		fr←fr XOR W
指令				
XOR fr,#literal	2	Z	隱含 W	r←fr XOR 常數
XOR fr1,fr2	2	Z	隱含 W	fr1←fr1 XOR fr2
旋轉指令				
MOV W,>>fr	1	C		W←fr 右旋轉
RR fr	1	C		fr←fr 右旋轉
MOV W,<<fr	1	C		W←fr 左旋轉
RL fr	1	C		fr←fr 左旋轉
位元指令				
CLRB bit	1			設定 bit=0
CLC	1			設定 C=0
CLZ	1			設定 Z=0
SETB bit	1			設定 bit=1
STC	1			設定 C=1
STZ	1			設定 Z=1
MOVB bit1,bit2	4			bit1←bit2
MOVB bit1,/bit2	4			bit1←bit2 反相



## (二)、OR 及 NOT 指令練習實習

- 1、動作：先設定令 RB，再令其閃爍輸出。
- 2、程式：2-3-2.ASM。

```

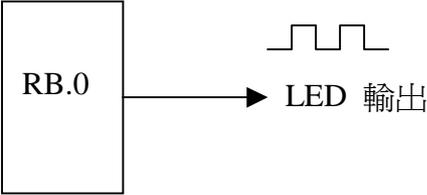
;*****2-3-2.ASM *****
;
;   PIC16C84
;
;   
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG     0CH ;使用者 RAM 啓始位址
DELY1 DS   1 ;延時暫存器
DELY2 DS   1 ;延時暫存器
;*****
;
;   ORG     00H ;程式啓始位址
START
;   MOV     !RB,#00 ;設 RB 爲輸出 ----(1)
;   CLR     RB ;令 RB=0
;   OR      RB,#11110000B ;預先設定 RB 的內容--(2)
LOOP  CALL  DELAY ;呼叫延時副程式 --(3)
;   NOT     RB ;RB 反相輸出 --(4)
;   JMP     LOOP ;跳到 LOOP 處重覆輸出(5)
;*****
;
;   DELAY ;延時 0.2 秒副程式
;   MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL  MOV     DELY2,#0FFH
;   DJNZ    DELY2,$
;   DJNZ    DELY1,DEL
;   RET     ;回主程式
;*****
;
;   END

```

## (三)、CLRB 及 SETB 指令練習實習

- 1、動作：令 RA.0 輸出方波。
- 2、程式：2-3-3.ASM。

```

;*****2-3-3.ASM *****
;
;   PIC16C84
;
;   
;
;
;
;
;
;
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG     0CH ;使用者 RAM 啓始位址
DELY1 DS   1   ;延時暫存器
DELY2 DS   1   ;延時暫存器
;*****
;
;   ORG     00H           ;程式啓始位址
START
;   MOV     !RB,#00      ;設 RB 爲輸出---(1)
LOOP
;   CLRB   RB.0         ;令 RB.0=0----(2)
;   CALL   DELAY        ;呼叫延時副程式
;   SETB   RB.0         ;令 RB.0=1  --(3)
;   CALL   DELAY        ;呼叫延時副程式
;   JMP    LOOP         ;跳到 LOOP 處重覆輸出(4)
;*****
;
;   DELAY           ;延時 0.2 秒副程式
;   MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL    MOV     DELY2,#0FFH
;   DJNZ   DELY2,$
;   DJNZ   DELY1,DEL
;   RET                    ;回主程式
;*****
;
;   END

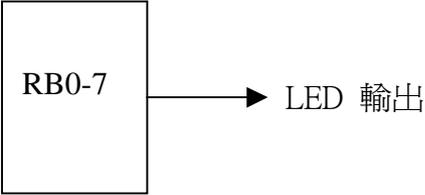
```



## (五)、8 位元左右移動作實習

- 1、動作：令 RB 左右來回移位。
- 2、程式：2-3-5.ASM。

```

;***** 2-3-5.ASM *****
;
;   PIC16C84
;
;   
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG    0CH    ;使用者 RAM 啓始位址
DELY1 DS    1    ;延時暫存器
DELY2 DS    1    ;延時暫存器
COUNT DS   1    ;旋轉次數暫存器
;*****
;
;   ORG    00H    ;程式啓始位址
;   MOV    !RB,# 00    ;設 RB 爲輸出 -----(1)
START
;   MOV    COUNT,#7    ;設定左旋轉次數=7 ----(2)
;   CLRB   C          ;令 C=0 -----(3)
;   MOV    RB,#0FFH    ;預先設定 RB 的內容
:LOOP CALL    DELAY    ;呼叫延時副程式-----(4)
;   RL     RB          ;令 RB 左旋轉輸出-----(5)
;   DJNZ  COUNT,:LOOP ;令左旋轉次數-1,若≠0(6)
;           ;則跳到:LOOP
;   MOV    COUNT,#7    ;設右旋轉次數)爲 7 ----(7)
:LOOP CALL    DELAY    ;呼叫延時副程式 -----(8)
;   RR     RB          ;令 RB 右旋轉輸出 -----(9)
;   DJNZ  COUNT,:LOOP ;令右旋轉次數-1,若≠0-(10)
;           ;則跳到:LOOP
;   JMP   START      ;若結束則跳 START 從頭開始(11)
;*****
;
;   DELAY                                ;延時 0.2 秒副程式
;           MOV    DELY1,#0FFH ;自轉 255*255 圈

```

```
DEL    MOV    DELY2,#0FFH
        DJNZ   DELY2,$
        DJNZ   DELY1,DEL
        RET                                ;回主程式
;*****
        END
```

**學習評量三：**

- 1、動作：令 RB.0 及 RA.0 均輸出方波，其餘保留。
- 2、動作：先設定令 RA 及 RB，再令其閃爍輸出。
- 3、動作：令 RB.2 輸出方波。



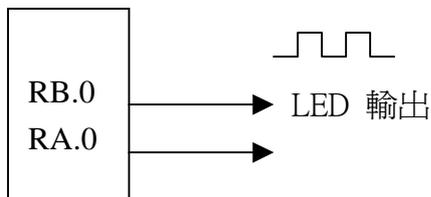
## 學習評量三答案：

1、令 RB.0 及 RA.0 均輸出方波，其餘保留的程式如下：

```

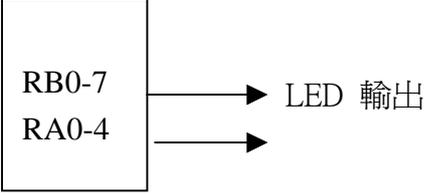
;*****
;
;   PIC16C84
;
;
;
;
;
;
;
;
;
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG     0CH ;使用者 RAM 啓始位址
DELY1 DS    1 ;延時暫存器
DELY2 DS    1 ;延時暫存器
;*****
;
;   ORG     00H ;程式啓始位址
START
;   MOV     !RB,#00 ;設 RB 爲輸出-----(1)
;   MOV     !RA,#00 ;設 RA 爲輸出
;   AND     RB,#1111110B ;令 RB.0=0-----(2)
;   AND     RA,#1111110B ;令 RA.0=0
LOOP  CALL  DELAY ;呼叫延時副程式-----(3)
;   XOR     RB,#00000001B ;RB.0 反相輸出-----(4)
;   XOR     RA,#00000001B ;RA.0 反相輸出
;   JMP     LOOP ;跳到 LOOP 處重覆輸出-(5)
;*****
;
;   DELAY ;延時 0.2 秒副程式
;   MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL  MOV     DELY2,#0FFH
;   DJNZ    DELY2,$
;   DJNZ    DELY1,DEL
;   RET ;回主程式
;*****
;
;   END

```



2、先設定令 RA 及 RB，再令其閃爍輸出的動作程式如下：。

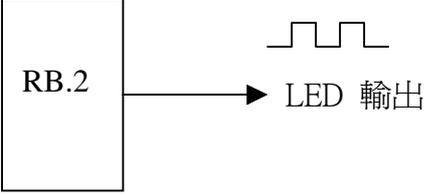
```

;*****
;
;   PIC16C84
;
;   
;
;   RB0-7
;   RA0-4
;
;   LED 輸出
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG    0CH ;使用者 RAM 啓始位址
DELY1 DS    1 ;延時暫存器
DELY2 DS    1 ;延時暫存器
;*****
;
;   ORG    00H ;程式啓始位址
START
;   MOV    !RB,#00 ;設 RB 爲輸出 ----(1)
;   MOV    !RA,#00 ;設 RA 爲輸出
;   MOV    RB,#11110000B ;預先設定 RB 的內容--(2)
;   MOV    RA,#00001111B ;預先設定 RA 的內容
LOOP CALL    DELAY ;呼叫延時副程式 --(3)
;   NOT    RB ;RB 反相輸出 --(4)
;   NOT    RA ;RA 反相輸出
;   JMP    LOOP ;跳到 LOOP 處重覆輸出(5)
;*****
;
;   DELAY ;延時 0.2 秒副程式
;   MOV    DELY1,#0FFH ;自轉 255*255 圈
DEL MOV    DELY2,#0FFH
;   DJNZ   DELY2,$
;   DJNZ   DELY1,DEL
;   RET    ;回主程式
;*****
;
;   END

```

2、令 RB.2 輸出方波動作程式如下：。

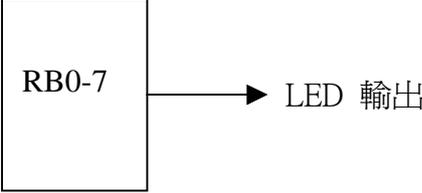
```

;*****
;
;   PIC16C84
;
;   
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG    0CH ;使用者 RAM 啓始位址
DELY1 DS    1 ;延時暫存器
DELY2 DS    1 ;延時暫存器
;*****
;
;   ORG    00H ;程式啓始位址
START
;   MOV    !RB,#00 ;設 RB 爲輸出---(1)
LOOP
;   CLRB   RB.2 ;令 RB.2=0----(2)
;   CALL  DELAY ;呼叫延時副程式
;   SETB   RB.2 ;令 RB.2=1 --(3)
;   CALL  DELAY ;呼叫延時副程式
;   JMP    LOOP ;跳到 LOOP 處重覆輸出(4)
;*****
;
;   DELAY ;延時 0.2 秒副程式
;   MOV    ELY1,#0FFH ;自轉 255*255 圈
DEL    MOV    DELY2,#0FFH
;   DJNZ   DELY2,$
;   DJNZ   DELY1,DEL
;   RET    ;回主程式
;*****
;
;   END

```

2、令 RB 右旋轉動作程式如下：。

```

;*****
;
;   PIC16C84
;
;   
;
;
;
;
;
;
;
;*****
        DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
        ORG     0CH      ;使用者 RAM 啓始位址
DELY1   DS     1        ;延時暫存器
DELY2   DS     1        ;延時暫存器
;*****
        ORG     00H      ;程式啓始位址
START
        MOV     !RB,#00   ;設 RB 爲輸出-----(1)
        CLRB    C        ;令 C=0 -----(2)
        MOV     RB,#0FFH  ;預先設定 RB 的內容--(3)
LOOP     CALL   DELAY     ;呼叫延時副程式
        RR     RB        ;令 RB 右旋轉輸出 ---(4)
        JMP    LOOP      ;跳到 LOOP 處重覆輸出(5)
;*****
DELAY                                         ;延時 0.2 秒副程式
        MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL     MOV     DELY2,#0FFH
        DJNZ   DELY2,$
        DJNZ   DELY1,DEL
        RET                                           ;回主程式
;*****
        END

```

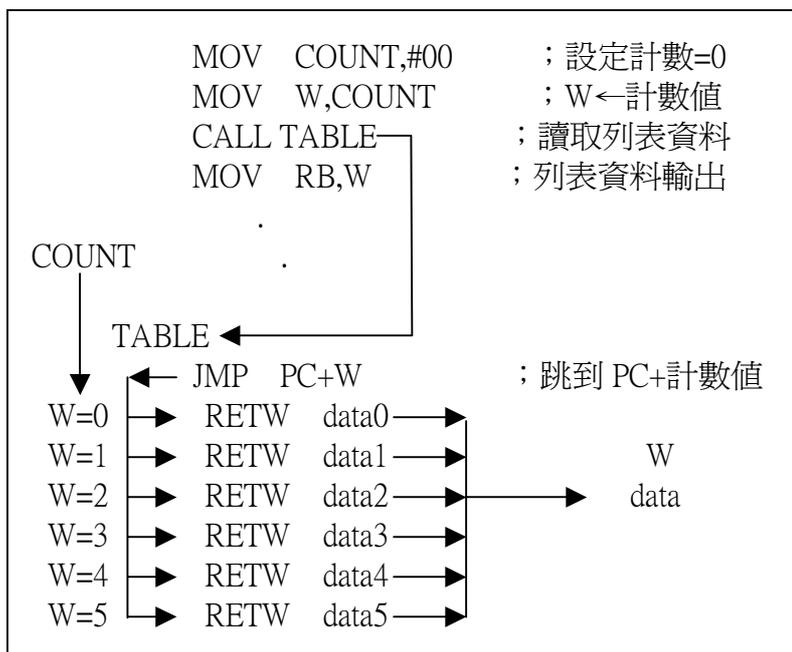
假如你的答案與上述之重點相似，請翻下一頁，假如你的答案不與上述之重點相似，請翻至第 36 頁重新閱讀以便發現你的錯誤之處，並將錯誤改正。

### 本單元的第四個學習目標是：

學員能在具備指令表之下，寫出列表法輸出程式。

所謂列表法就是在 ROM 內建立一個表(TABLE)，事先將資料燒錄在列表內，然後依序取出資料來工作。使得動作更具有多樣化。

PIC 系列的單晶片微處理機本身並無列表功能的指令，所以它必須應用呼叫的方式。當回歸主程式時，由 RETW 指令將 ROM 的資料存入工作暫存器(W)內，來達到列表的功能。動作如下圖。



它的動作是將計數值(COUNT) 送入 W 暫存器，然後由 W 的內容決定讀取那一個資料，讀取到的資料會回存入 W，再將 W 的資料輸出。如 COUNT=0 則讀到的資料為 data0，依此類推。

列表資料的存放位址有四個限制要特別注意：

◎列表資料的型態可分為：

二進制：如 RETW 10010101B,00010001B,10111101B,

十進制：如 RETW 82,05,40

十六進制：如 RETW 8AH,05H,0FFH

ASCII 碼：如 RETW 'abcdefg'

混合資料：如 RETW 'abc',0A5H,32,00000111B

◎ 列表資料的位址匯流排僅能改變低 8 位元，故總容量不得超過 256 Byte (00H~FFH)。

假如你認為能夠勝任以上學習目標的能力，請翻至第 62 頁做測驗。

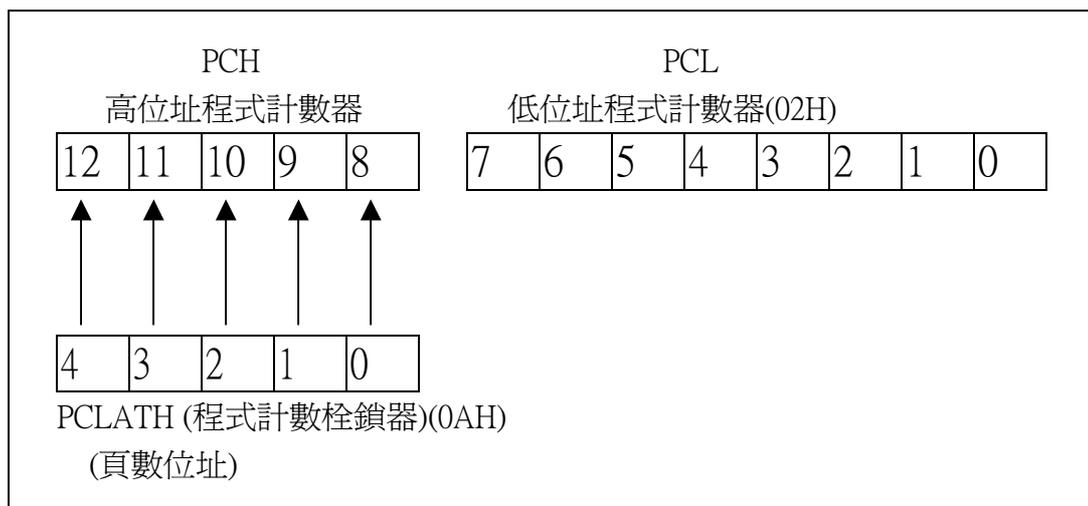
假如你需要更多學習的話，請翻至第 51 頁。

- ◎列表資料的位址不得有跨頁的情形，如 2FH~350H。
  - ◎列表資料的位址若不在第 0 頁，必須將高位址(也就是頁數)存入 PCLATH 暫存器，再存入 PCH 暫存器，配合 PCL 暫存器，才能形成 13 位元的位址匯流排。
- 習慣上我們將記憶體以頁數來區分，如下表。

記憶體各頁數位址

頁數	位址
0	000~0FFH
1	100~1FFH
2	200~2FFH
3	300~3FFH

其中 PCL 在特殊暫存器位址 02H，而 PCLATH 在特殊暫存器位址 0AH。如下圖。



列表工作時程式記憶體的位址

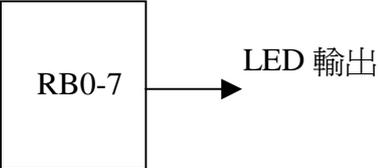
本節進行的實習項目如下：

- ◎ 列表法基本應用實習
- ◎ 紅黃綠燈順序控制實習
- ◎ 大容量列表資料程式實習(1)
- ◎ 大容量列表資料程式實習(2)

### (一)、列表法基本應用實習

- 1、動作：由 RB 輸出列表內的動作。
- 2、程式：2-4-1.ASM

```

;***** 2-4-1.ASM *****
;
;   PIC16C84
;
;   
;
;
;
;
;
;
;
;*****
          DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
          ORG     0CH ;使用者 RAM 啓始位址
COUNT   DS     1   ;列表的計數位址暫存器
DELY1    DS     1   ;延時暫存器
DELY2    DS     1   ;延時暫存器
DELY3    DS     1   ;延時暫存器
;*****
          ORG     00H           ;程式啓始位址
START
          MOV     !RB,#00       ;設 RB 爲輸出 -----(1)
          CLR     COUNT         ;清除列表的計數位址 -----(2)
LOOP
          MOV     W,COUNT       ;計數位址→ W -----(3)
          CALL    TABLE        ;讀取列表資料→W -----(4)
          MOV     RB,W          ;列表的資料由埠 B 輸出-----(5)
          CALL    DELAY         ;呼叫延時副程式
          INC     COUNT         ;列表的計數位址加一 -----(6)
          CJNE   COUNT,#16,LOOP ;計數位址-1，若未結束則跳 --(7)

```

```

;到 LOOP 處重覆下一個輸出
        JMP      START      ;若已結束則跳到 START 從頭開始(8)
;*****
DELAY
        MOV      DELY1,#0FFH ;延時 0.2 秒副程式
        ;自轉 255*255 圈
DEL      MOV      DELY2,#0FFH
        DJNZ     DELY2,$
        DJNZ     DELY1,DEL
        RET
;回主程式
;*****
TABLE
        JMP      PC+W
        RETW     1000000B
        RETW     0100000B
        RETW     0010000B
        RETW     0001000B
        RETW     0000100B
        RETW     0000010B
        RETW     0000001B
        RETW     0000001B
        RETW     0000010B
        RETW     0000100B
        RETW     0000100B
        RETW     0001000B
        RETW     0010000B
        RETW     0100000B
        RETW     1000000B
        RETW     1111111B
;-----
        END

```



```

MOV     W,COUNT      ;計數位址→ W -----(7)
CALL    TABLE       ;讀取列表延時資料→W
MOV     DELY0,W       ;將延時資料→DELY0 ----(8)
CALL    DELAY        ;呼叫延時副程式
INC     COUNT        ;列表的計數位址加一 -----(9)
JMP     LOOP         ;跳到 LOOP 重覆下一個資料輸出-(10)
;*****
DELAY                                       ;延時 0.2*DELY0 秒副程式
MOV     DELY1,#0FFH ;自轉 255*255*DELY0 圈
DEL     MOV     DELY2,#0FFH
        DJNZ    DELY2,$
        DJNZ    DELY1,DEL
        DJNZ    DELY0,DELAY
        RET     ;回主程式
;*****
TABLE                                       ;列表資料
        JMP     PC+W
;      紅黃綠燈 ,延時
;      RYGRYG
;      222111 ,DELAY
RETW    00100001B,30
RETW    00100000B,1
RETW    00100001B,1
RETW    00100000B,1
RETW    00100001B,1
RETW    00100000B,1
RETW    00100001B,1
RETW    00100010B,4
RETW    00001100B,30
RETW    00000100B,1
RETW    00001100B,1
RETW    00000100B,1
RETW    00001100B,1
RETW    00000100B,1
RETW    00001100B,1
RETW    00000100B,1
RETW    00010100B,4
RETW    0          ;從頭開始
END

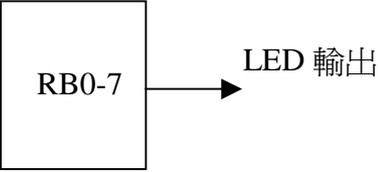
```

## (三)、大容量列表資料程式實習(1)

1、動作：列表資料佔用 3 頁，由 RB 輸出列表資料。

2、程式：2-4-3.ASM

```

;***** 2-4-3.ASM *****
;
;   PIC16C84
;
;   
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG     0CH           ;使用者 RAM 啓始位址
COUNT    DS     1       ;列表的計數位址暫存器
DELY1     DS     1       ;延時暫存器
DELY2     DS     1       ;延時暫存器
;*****
;
;   ORG     00H           ;程式啓始位址
;   MOV     !RB,#00      ;設 RB 爲輸出
START
;   MOV     PCLATH,#1    ;設定第一頁位址
;   CLR     COUNT        ;清除列表的計數位址
LOOP1
;   MOV     W,COUNT      ;計數位址→ W
;   CALL    TABLE1      ;讀取第一頁列表資料→W
;   TEST    W             ;測試資料是否爲 0(結束)
;   JZ     PAGE2         ;若是，表示已結束，則跳到 PAGE2 換第 2 頁
;   MOV     RB,W          ;列表的資料反相由埠 B 輸出
;   CALL    DELAY         ;呼叫延時副程式
;   INC     COUNT         ;列表的計數位址加一
;   JMP     LOOP1        ;跳到 LOOP1 重覆下一個資料輸出
;-----
PAGE2
;   MOV     PCLATH,#2    ;設定第二頁位址
;   CLR     COUNT        ;列表的計數位址=0
LOOP2
;   MOV     W,COUNT      ;計數位址→ W
;   CALL    TABLE2      ;讀取第二頁列表資料→W

```

```

TEST    W                ;測試資料是否為 0(結束)
JZ      PAGE3            ;若是，表示已結束，則跳到 PAGE3 換第 3 頁
MOV     RB,W             ;列表的資料反相由埠 B 輸出
CALL    DELAY            ;呼叫延時副程式
INC     COUNT            ;列表的計數位址加一
JMP     LOOP2            ;跳到 LOOP2 重覆下一個資料輸出
;-----
PAGE3                                ;讀取第三頁資料
MOV     PCLATH,#3        ;設定第三頁位址
CLR     COUNT            ;列表的計數位址=0
LOOP3
MOV     W,COUNT          ;計數位址→ W
CALL    TABLE3         ;讀取第三頁列資料→W
TEST    W                ;測試資料是否為 0(結束)
JZ      START           ;若是，表示已結束，則跳到 START 從頭開始
MOV     RB,W             ;列表的資料反相由埠 B 輸出
CALL    DELAY            ;呼叫延時副程式
INC     COUNT            ;列表的計數位址加一
JMP     LOOP3            ;跳到 LOOP3 重覆下一個資料輸出
;*****
DELAY                                ;延時 0.2 秒副程式
MOV     DELY1,#0FFH     ;自轉 255*255 圈
DEL     MOV     DELY2,#0FFH
        DJNZ   DELY2,$
        DJNZ   DELY1,DEL
RET                                ;回主程式
;*****
ORG     100H              ;第一頁列表資料
;
TABLE1  ; 76543210
        JMP   PC+W
        RETW  1000000B
        RETW  0100000B
        RETW  0010000B
        RETW  0001000B
        RETW  0000100B
        RETW  00000100B
        RETW  00000010B

```

```
        RETW    0000001B
        RETW    0000000B
;*****
;
        ORG     200H           ;第二頁列表資料
;
        DATA
TABLE2      ; 76543210
        JMP     PC+W
        RETW    1100000B
        RETW    0110000B
        RETW    0011000B
        RETW    0001100B
        RETW    0000110B
        RETW    0000011B
        RETW    0000011B
        RETW    0000000B
;*****
;
        ORG     300H           ;第三頁列表資料
;
        DATA
TABLE3      ; 76543210
        JMP     PC+W
        RETW    1110000B
        RETW    0111000B
        RETW    0011100B
        RETW    0001110B
        RETW    0000111B
        RETW    0000011B
        RETW    0001110B
        RETW    0001110B
        RETW    0000000B
;*****
;
        END
```

## (四)、大容量列表資料程式實習(2)

上述 2-4-3.ASM 程式中有許多重覆的地方，太佔用記憶體，可精簡如下：

1、動作：列表資料佔用 3 頁，由 RB 輸出列表資料，並由 RA 輸出頁數。

2、程式：2-4-4.ASM

```

;***** 2-4-4.ASM *****
;
;   PIC16C84
;
;   RA0-4 → 頁輸出
;
;   RB0-7 → 資料輸出
;
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG    0CH                ;使用者 RAM 啓始位址
COUNT   DS    1             ;列表的計數位址暫存器
PAGE     DS    1             ;頁數暫存器
DELY1    DS    1             ;延時暫存器
DELY2    DS    1             ;延時暫存器
;*****
;
;   ORG    00H                ;程式啓始位址
;   MOV    !RA,#00            ;設 RA 爲輸出
;   MOV    !RB,#00            ;設 RB 爲輸出
START
;   MOV    PAGE,#01           ;設定第一頁
LOOP_PAGE
;   MOV    RA,PAGE            ;頁數由 RA 輸出
;   MOV    PCLATH,PAGE        ;設定讀取頁數
;   CLR    COUNT              ;列表的計數位址=0
LOOP
;   MOV    W,COUNT            ;計數位址→ W
;   CALL   TABLE              ;讀取列表資料→W
;   TEST   W                    ;測試資料是否爲 0(結束)
;   JZ     DATA_END           ;若是，跳到 DATA_END，換下一頁
;   MOV    RB,W                ;列表的資料反相由埠 B 輸出
;   CALL   DELAY                ;呼叫延時副程式

```

```

        INC     COUNT           ;列表的計數位址加一
        JMP     LOOP           ;跳到 LOOP 重覆下一個資料輸出
DATA_END
        INC     PAGE           ;頁數加一
        CJNE   PAGE,#04,LOOP_PAGE ;頁數未結束，則到 LOOP_PAGE，換下一頁
        JMP     START         ;若頁數結束，跳到 START 從頭開始
;*****
DELAY                                     ;延時 0.2 秒副程式
        MOV     DELY1,#0FFH    ;自轉 255*255 圈
DEL     MOV     DELY2,#0FFH
        DJNZ   DELY2,$
        DJNZ   DELY1,DEL
        RET                                     ;回主程式
;*****
        ORG     100H           ;第一頁列表資料
;
        DATA
TABLE   ; 76543210
        JMP     PC+W
        RETW   10000000B
        RETW   01000000B
        RETW   00100000B
        RETW   00010000B
        RETW   00001000B
        RETW   00000100B
        RETW   00000010B
        RETW   00000001B
        RETW   00000000B
;*****
        ORG     200H           ;第二頁列表資料
        JMP     PC+W
        RETW   11000000B
        RETW   01100000B
        RETW   00110000B
        RETW   00011000B
        RETW   00001100B
        RETW   00000110B
        RETW   00000011B
        RETW   00000110B

```

```
    RETW    00000000B
;*****
;
    ORG     300H           ;第三頁列表資料
    JMP     PC+W
    RETW    11100000B
    RETW    01110000B
    RETW    00111000B
    RETW    00011100B
    RETW    00001110B
    RETW    00000111B
    RETW    00001110B
    RETW    00011100B
    RETW    00000000B
;*****
;
    END
```

學習評量四：

動作：由 RB 輸出如下表的順序動作。

11000000B

01100000B

00110000B

00011000B

00001100B

00000110B

00000011B

00000110B

00001100B

00011000B

00110000B

01100000B

11000000B

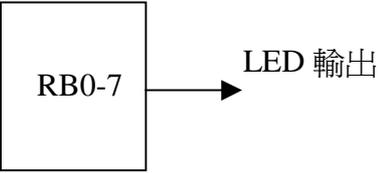
11111111B



## 學習評量四答案：

由 RB 輸出指定的順序動作程式如下。

```

;*****
;
;   PIC16C84
;
;   
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG     0CH      ;使用者 RAM 啓始位址
COUNT    DS     1      ;列表的計數位址暫存器
DELY1     DS     1      ;延時暫存器
DELY2     DS     1      ;延時暫存器
DELY3     DS     1      ;延時暫存器
;*****
;
;   ORG     00H      ;程式啓始位址
START
;   MOV     !RB,#00      ;設 RB 爲輸出 -----(1)
;   CLR     COUNT      ;清除列表的計數位址 -----(2)
LOOP
;   MOV     W,COUNT     ;計數位址→ W -----(3)
;   CALL    TABLE     ;讀取列表資料→W -----(4)
;   MOV     RB,W        ;列表的資料由埠 B 輸出-----(5)
;   CALL    DELAY      ;呼叫延時副程式
;   INC     COUNT      ;列表的計數位址加一 -----(6)
;   CJNE   COUNT,#14,LOOP ;計數位址-1，若未結束則跳 --(7)
;                                     ;到 LOOP 處重覆下一個輸出
;   JMP     START      ;若已結束則跳到 START 從頭開始(8)
;*****
;
;   DELAY      ;延時 0.2 秒副程式
;   MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL      MOV     DELY2,#0FFH
;   DJNZ   DELY2,$
;   DJNZ   DELY1,DEL
;   RET      ;回主程式

```

```
*****
;
```

```
TABLE ;列表資料
```

```

    JMP    PC+W
    RETW   11000000B
    RETW   01100000B
    RETW   00110000B
    RETW   00011000B
    RETW   00001100B
    RETW   00000110B
    RETW   00000011B
    RETW   00000110B
    RETW   00001100B
    RETW   00011000B
    RETW   00110000B
    RETW   01100000B
    RETW   11000000B
    RETW   11111111B
```

```
-----;
```

```
END
```

---

假如你的答案與上述之重點相似，請翻下一頁，假如你的答案不與上述之重點相似，請翻至第 50 頁重新閱讀以便發現你的錯誤之處，並將錯誤改正。

本單元的第五個學習目標是：

學員能在具備指令表之下，寫出輸入動作程式。

### 五、 輸入動作實習

PIC16C84 可以處理位元組及位元的輸入動作，一般是配合搬移及位元分指令來進行輸入資料及判斷，常用的指令如下表：

位元條件分支跳躍指令

指令	字元數	影響旗標	隱含 W	說明
SB bit	1			若 bit=1，則跳過下一行
SNB bit	1			若 bit=0，則跳過下一行
JB bit,addr9	2			若 bit=1，則跳到 addr9
JNB bit_addr,addr9	2			若 bit=0，則跳到 addr9

- 1、動作：PA0-4 輸入的資料，由 PB0-4 反相閃爍輸出。
- 2、程式：如 2-5-1.ASM。

```

;*****2-5-1.ASM *****
;
;          PIC16C84
;
;
;  SW  ────┐
;          │
;          │ RA0-4
;          │ RB0-4
;          │
;          └───┬─── LED 輸出
;
;
;*****
;          DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;          ORG      0CH ;使用者 RAM 啓始位址
DELY1  DS      1 ;延時暫存器
DELY2  DS      1 ;延時暫存器
;*****
;          ORG      00H ;程式啓始位址
START
MOV     !RB,#00 ;設 RB 爲輸出-----(1)

```

假如你認為能夠勝任以上學習目標的能力，請翻至第 68 頁做測驗。  
假如你需要更多學習的話，請翻至第 65 頁。

```

        MOV    !RA,#0FFH    ;設 RA 為輸入

        MOV    RB,RA        ;由埠 A 輸入至埠 B 輸--(2)
LOOP    CALL   DELAY        ;呼叫延時副程式----(3)
        NOT    RB           ;埠 B 反相輸出-----(4)
        JMP    LOOP        ;跳到 LOOP 處重覆輸出(5)
;*****
DELAY                                ;延時 0.2 秒副程式
        MOV    DELY1,#0FFH  ;自轉 255*255 圈
DEL     MOV    DELY2,#0FFH
        DJNZ   DELY2,$
        DJNZ   DELY1,DEL
        RET                                ;回主程式
;*****
        END

```

## (二)、某一位元輸入控制

- 1、動作：由 RA.0 控制 RB 輸出上數或下數計數。
- 2、程式：如 2-5-2.ASM。

```

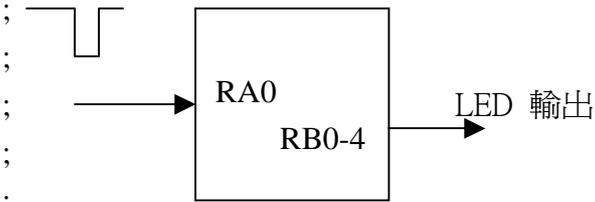
;*****2-5-2.ASM *****
;
;          PIC16C84
;
;
;  SW  →  [ RA0
;         RB0-4 ]  →  LED 輸出
;
;
;*****
;          DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;          ORG     0CH ;使用者 RAM 啓始位址
DELY1  DS      1 ;延時暫存器
DELY2  DS      1 ;延時暫存器
;*****
;          ORG     00H ;程式啓始位址
START
;          MOV     !RB,#00 ;設 RB 爲輸出 -----(1)
;          MOV     !RA,#01H ;設 RA 爲輸入
;          CLR     RB ;清除埠 B 輸出 -----(2)
LOOP
;          CALL    DELAY ;呼叫延時副程式 -----(3)
;          JB     RA.0,QQQ ;判斷若 RA.0=1 則跳到 QQQ-(4)
;          INC     RB ;若 RA.0=0，則 RB+1 輸出--(5)
;          JMP     LOOP ;跳到 LOOP 處重覆動作
QQQ
;          DEC     RB ;若 RA.0=1，則 RB-1 輸出 -(6)
;          JMP     LOOP ;跳到 LOOP 處重覆動作
;*****
;          DELAY ;延時 0.2 秒副程式
;          MOV     DELY1,#0FFH ;自轉 255*255 圈
DEL     MOV     DELY2,#0FFH
;          DJNZ   DELY2,$
;          DJNZ   DELY1,DEL
;          RET ;回主程式
;          END

```

## (三)、某一位元輸入脈波控制

- 1、動作：由 RA.0 手動輸入脈波，控制 RB 輸出上數計數。
- 2、程式：如 2-5-3.ASM。

```

;*****2-5-3.ASM *****
;
;          PIC16C84
;
; 
;
; *****
;
;          DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;          ORG      0CH ;使用者 RAM 啓始位址
DELY1 DS      1 ;延時暫存器
DELY2 DS      1 ;延時暫存器
; *****
;          ORG      00H ;程式啓始位址
START
MOV      !RB,#00 ;設 RB 爲輸出 -----(1)
MOV      !RA,#01H ;設 RA 爲輸入
CLR      RB ;清除埠 B 輸出 -----(2)
LOOP
JB       RA.0,$ ;若 RA.0=1 自轉,等待輸入 0 ---(3)
CALL    DELAY ;呼叫延時副程式,避免機械跳動
JNB     RA.0,$ ;若 RA.0=0 自轉,等待輸入 1 ----(4)
CALL    DELAY ;呼叫延時副程式,避免機械跳動
INC     RB ;若 RA.0 輸入一個脈波,RB+1 輸出-(5)
JMP     LOOP ;跳到 LOOP 處重覆動作
; *****
;          DELAY ;延時 0.768ms
;          MOV     DELY2,#0FFH ;自轉 255 圈
;          DJNZ    DELY2,$
;          RET
; *****
;          END

```

**學習評量五：**

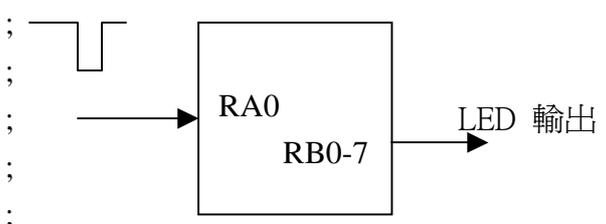
動作：由 RA.0 每輸入一個脈波，即令 RB 由列表輸出一位元組的資料。



## 學習評量五答案：

令由 RA.0 每輸入一個脈波，即令 RB 由列表輸出一位元組的資料，的程式如下：

```

;*****
;
;          PIC16C84
;
; 
;
;          RA0
;          RB0-7
;          LED 輸出
;
;*****
          DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
          ORG      0CH      ;使用者 RAM 啓始位址
COUNT DS      1          ;列表的計數位址暫存器
DELY1  DS      1          ;延時暫存器
DELY2  DS      1          ;延時暫存器
DELY3  DS      1          ;延時暫存器
;*****
          ORG      00H          ;程式啓始位址
START
          MOV      !RB,#00      ;設 RB 爲輸出 -----(1)
          MOV      !RA,#01H     ;設 RA0 爲輸入
          CLR      COUNT        ;清除列表的計數位址 -----(2)
LOOP
          JB       RA.0,$        ;若 RA.0=1 自轉,等待輸入 0 ---(3)
          CALL    DELAY1         ;呼叫延時 1 副程式,避免機械跳動
          JNB     RA.0,$        ;若 RA.0=0 自轉,等待輸入 1
          CALL    DELAY1         ;呼叫延時 1 副程式,避免機械跳動
          MOV     W,COUNT        ;計數位址→ W -----(4)
          CALL    TABLE        ;讀取列表資料→W
          MOV     RB,W          ;列表的資料由埠 B 輸出-----(5)
          CALL    DELAY         ;呼叫延時副程式
          INC     COUNT         ;列表的計數位址加一 -----(6)
          CJNE   COUNT,#14,LOOP;計數位址-1，若未結束則跳 --(7)
                                ;到 LOOP 處重覆下一個輸出
          JMP     START         ;若已結束則跳到 START 從頭開始(8)
;*****

```

```

DELAY                                ;延時 0.2 秒副程式
    MOV    DELY1,#0FFH                ;自轉 255*255 圈
DEL    MOV    DELY2,#0FFH
    DJNZ   DELY2,$
    DJNZ   DELY1,DEL
    RET                                ;回主程式
;*****
DELAY1                                ;延時 0.768ms
    MOV    DELY2,#0FFH                ;自轉 255 圈
    DJNZ   DELY2,$
    RET
;*****
TABLE                                ;列表資料
    JMP    PC+W
    RETW   11000000B
    RETW   01100000B
    RETW   00110000B
    RETW   00011000B
    RETW   00001100B
    RETW   00000110B
    RETW   00000011B
    RETW   00000110B
    RETW   00001100B
    RETW   00011000B
    RETW   00110000B
    RETW   01100000B
    RETW   11000000B
    RETW   11111111B
;-----
    END

```

假如你的答案與上述之重點相似，請翻下一頁，假如你的答案不與上述之重點相似，請翻至第 64 頁重新閱讀以便發現你的錯誤之處，並將錯誤改正。

本單元的第六個學習目標是：

學員能在具備指令表之下，寫出執行時間計算的程式。

#### 六、執行時間計算實習

微處理機執行的時間是由石英晶體或輸入的時脈頻率來決定的，如下表。

狀態週期時間

石英晶體	狀態週期時間
4MHz	$8\text{MHz}/4=1\text{MHz}=1\mu\text{s}$
10KHz	$40\text{KHz}/4=10\text{KHz}=0.1\text{ms}$

一般都是應用指令的迴轉來達到延時的功能，所以必須知道每個指令執行時所須耗費的狀態週期時間，如下表。

如此我們祇要將程式中每個指令執行的狀態週期數總和，再乘以狀態週期時間，即可求得執行的時間。

常用的指令狀態週期數

指 令	字元數	狀態週期	分 支 指 令	字元數	狀態週期
MOV W,#lit	1	1	CALL addr	1	2
MOV W,fr	1	1	RET	1	2
MOV fr,#lit	2	2	RETI	1	2
MOV fr,#fr1	2	2	JMP addr	1	2
AND W,#lit	1	1	JNB bit,addr	2	(2)3
XOR W,fr	1	1	JB bit,addr	2	(2)3
XOR fr,W	1	1	JNC addr	2	(2)3
XOR fr,W	2	2	JC addr	2	(2)3
XOR fr,#lit	2	2	DJNZ fr,addr	2	(2)3
XOR fr,fr1	2	2	CJNE fr,#lit,addr	4	(4)5
NOT W	1	1			
NOT W,fr	1	1			
SETB bit	1	1			
CLRB bit	1	1			

由下表知，每個指令的狀態週期和字元數是相關的，其中分支指令若有產生跳躍動作時，將字元數加一即是它的狀態週期。但若不產生分支跳躍動作時則狀態週期不加一。

以延時副程式為例：

```

DELAY
    MOV    DELY2,#100 ;2
    DJNZ
DELY2,$    ;3*99+2
    RET                ;2

```

其中指令“DJNZ DELY2,\$”會自轉 99 圈，最後一次因條件符合會往下執行，故指令的狀態週期為“3\*99+2”。如果時間不須太精確可以用“3\*100”來計算。

執行時間計算實習範例如下：

- (一)、動作：石英晶體使用 4MHz，令 RB0-7 輸出方波。
- (二)、程式：2-6-1.ASM。

```

;*****2-6-1.ASM *****

```

```

; PIC16C84

```

```

;

```

```

;

```

```

;

```

```

;

```

```

;

```

```

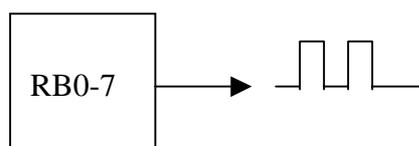
;

```

```

;

```



```

;*****
        DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
        ORG     0CH ;使用者 RAM 啓始位址
DELY1  DS     1 ;延時暫存器
DELY2  DS     1 ;延時暫存器
;*****
        ORG     00H ;程式啓始位址
START
        MOV     !RB,#00 ;設 RB 爲輸出
        CLR     RB ;清除埠 B 輸出
LOOP
        CALL    DELAY1 ;3+(2+3*99+2+2)=306--(1)
; CALL    DELAY2 ;3+{2+(2+3*100+3)*200+2}=61007--(3)
        NOT     RB ;1-----(2)
        JMP     LOOP ;2
;*****
DELAY1
        MOV     DELY2,#100 ;2
        DJNZ    DELY2,$ ;3*99+2
        RET     ;2
;*****
DELAY2
        MOV     DELY1,#200 ;2
DEL     MOV     DELY2,#100 ;2
        DJNZ    DELY2,$ ;3*100
        DJNZ    DELY1,DEL ;3 *200
        RET     ;2
;*****
        END

```

### (三)、程式說明：

1、呼叫副程式 DELAY1 的延時狀態週期計算公式：

狀態週期=3+(2+3\*99+2+2)=306 個。

2、再加上第(2)項的兩個指令的 1+2 個狀態週期，即是 RB 反相輸出的時間。

$T = \{3 + (2 + 3 \times 99 + 2 + 2) + 1 + 2\} \times 1 \mu s = 309 \mu s$
--

- 3、若要有較長的延時間改為呼叫副程式 DELAY2，使用雙迴圈延時副程式，並以較不精確的計算如下：

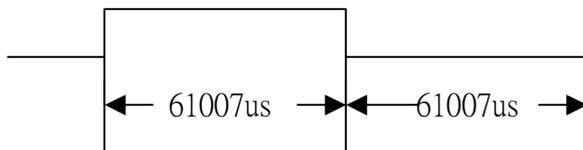
```

DELAY2
    LD    DELY1,#200    ;2
DEL    LD    DELY2,#100 ;2 ←
    DJNZ  DELY2,$      ;3*100
    DJNZ  DELY1,DEL     ;3   *200
    RET                ;2
  
```

此雙迴圈所形成的延時時間公式為：

$$T = \{2 + (2 + 3 * 100 + 3) * 200 + 2\} * 1\mu s = 61004\mu s = 61.004ms$$

由 RB0-7 輸出方波的反相時間 =  $61004\mu s + 1\mu s + 2\mu s = 61007\mu s$ ，如下：



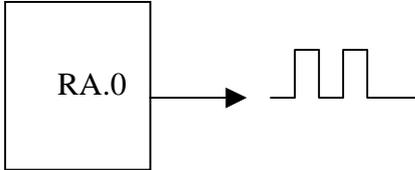
**學習評量六：**

寫出以自己學號的最後兩碼數字，作為由 RA.0 輸出的 XX KHz 的頻率程式，並以示波器驗證。



## 學習評量六答案：

```

;*****
;
;   PIC16C84
;
;   
;
;
;
;
;
;
;*****
;
;   DEVICE   PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG      0CH   ;使用者 RAM 啓始位址
DELY1 DS    1     ;延時暫存器
DELY2 DS    1     ;延時暫存器
;*****
;
;   ORG      00H           ;程式啓始位址
START
;   MOV      !RA,#00      ;設 RA 爲輸出
LOOP
;   CALL     DELAY
;   XOR      RA,#0000001B ;令 RA.0 反相
;   JMP      LOOP        ;2
;*****
;
;   DELAY1                                ;延時時間隨各人的工作頻率而不同
;   MOV      DELY2,#100   ;2
;   DJNZ    DELY2,$      ;3*99+2
;   RET                                           ;2
;*****
;
;   END

```

假如你的答案與上述之重點相似，請翻下一頁，假如你的答案不與上述之重點相似，請翻至第 71 頁重新閱讀以便發現你的錯誤之處，並將錯誤改正。

本單元的第七個學習目標是：

學員能在具備指令表之下，寫出大容量程式及 PIC16C84 與 PIC16C5X 程式的轉換。

---

假如你認為能夠勝任以上學習目標的能力，請翻至第 89 頁做測驗。  
假如你需要更多學習的話，請翻至第 78 頁。

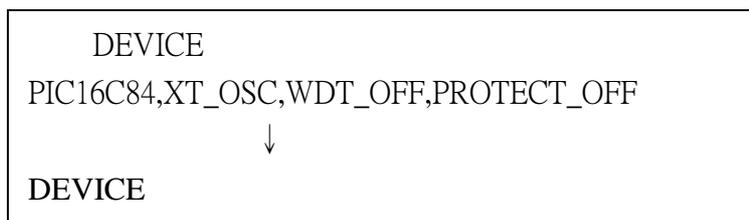
## 七、大容量程式及 PIC16C84 與 PIC16C5X 程式的轉換實習

PIC16C84 是 PIC16C54/56 的改良品，除了 RA4 外，接腳是完全相同的。差別是 PIC16C54/56 的功能較少，及本身記憶體管理受限較多，且只能燒錄一次，不過價格較為便宜。

如果要將 PIC16C84 的程式轉換為 PIC16C54/56 來使用時，有些地方要特別留意。

## (一)、使用晶片名稱的修改

組譯器的晶片選擇必須加以修改，如下：



## (二)、啓始位址及一般暫存器位址的修改

PIC16C84 和 PIC16C54/56 在記憶體管理上差異很大，如下：

1、PIC16C84 和 PIC16C54/56 在記憶體管理比較，如下表。

**PIC16C84 和 PIC16C54/56 的記憶體管理比較表**

晶片編號	最大記憶體	啓始位址	一般暫存器	EEPROM	記憶體管理限制
PIC16C84	1K *14 bit	00H	0CH~2FH	64BYTE	列表預定在第 0 頁，須設定 PCLATH 才能換頁。 副程式不限位址。
PIC16C54	512*14 bit	1FFH	07H~1FH	無	副程式及列表在第 0 頁內
PIC16C56	1K *14 bit	3FFH	07H~1FH	無	列表在第 0 頁內副程式 在第 0 頁及 2 頁內

(1)、PIC16C84 的程式啓始位址在 00H，而 PIC16C5X 程式啓始位址在最後面。

(2)、PIC16C84 的一般暫存器或 RAM 啓始位址由 0CH 開始，而 PIC16C5X 由 07H 開始。

(3)、PIC16C84 的 CALL 列表指令，預定在第 0 頁，可用 PCLATH 暫存器來選頁。

(4)、PIC16C84 的 CALL 指令不受位址限制。

(5)、PIC16C5X 本身沒有 PCLATH 暫存器。

(6)、PIC16C5X 的 CALL 指令會令 PC8=0，故 CALL 位址限於第 0 頁(00~FFH)及第 2 頁 (200H~2FFH)。其中 PIC16C54 可僅有第 0 頁，PIC16C56 可選第 0、2 頁。

## 2、PIC16C54/56 的記憶體設定

PIC16C84 程式的啓始位址在 000H，而 PIC16C54/56 在最後面，如 PIC16C54 在 01FFH 及 PIC16C56 在 3FFH。此時可用指令“RESET START”由啓始位址跳到 START 處開始執行。

PIC16C84 一般暫存器的啓始位址在 0CH，而 PIC16C54/56 由 07H 開始。程式設定必須加以修改，如下：



## (三)、大容量程式的修改

PIC16C54 的所有副程式及列表資料均必須在第 0 頁內(位址 00~FFH)。如下：

```

        DEVICE PIC16C54,XT_OSC,WDT_OFF,PROTECT_OFF
                ORG    07H    ;暫存器的位址由 07H 開始
DELY1    DS    1
DELY2    DS    1
;*****
        RESET    START    ;由啓始位址跳到 START
;*****
①        ORG    100H    ;主程式由 100H 開始
        → START
        主程式
        CALL    DELAY
        CALL    TABLE ←
;*****
        ORG    00H    ;副程式由 00H 開始
        → DELAY
        副程式
        RET ← ③
;*****
;列表資料第 0 頁內
TABLE    JMP    PC+W
        RETW    '      '
        RETW    'ABCDWFGHIJKLMNQP'
        RETW    '      '

```

PIC16C56 的列表資料均必須在第 0 頁內(位址 00~FFH)及副程式可在第 0 及 2 頁，若程式太大，必須加以修改，如下：

```

        DEVICE PIC16C56,XT_OSC,WDT_OFF,PROTECT_OFF
                ORG    07H    ;暫存器的位址由 07H 開始
DELY1    DS    1
DELY2    DS    1
;*****
        RESET    START    ;由啓始位址跳到 START
;*****
①      ORG    100H    ;主程式由 100H 開始
        START
                主程式
        LCALL   DELAY    ;長程呼叫副程式
        CALL   TABLE
;*****
        ORG    200H    ;副程式由 200H 開始
        DELAY
                副程式
        RET
;*****
        ORG    00H    ;列表資料第 0 頁內
TABLE1   JMP    PC+W
        RETW    '          '
        RETW    'ABCDFGHIJKLMNOP'
        RETW    '          '

```

The diagram illustrates the program flow with three numbered callouts:

- ①: Points to the `START` label, which is the entry point after the `RESET` instruction.
- ②: Points to the `CALL TABLE` instruction, which is reached after the `LCALL DELAY` instruction.
- ③: Points to the `RET` instruction, which returns control from the `DELAY` subroutine back to the `CALL TABLE` instruction.

## (三)、大容量副程式及列表資料的修改

若所有副程式無法一起容納在第 0 頁或第 2 頁內，可跳到其它頁數執行後，再跳回第 0 頁內，經 RET 回主程式，如下：

```

                DEVICE PIC16C56,XT_OSC,WDT_OFF,PROTECT_OFF
                ORG    07H      ;暫存器的位址由 07H 開始
DELY1          DS    1
DELY2          DS    1
;*****
RESET         START      ;由啓始位址跳到 START
;*****
①            ORG    100H     ;主程式由 100H 開始
START
LOOP          ;主程式
            LCALL   DELAY    ;長程呼叫副式
            LJMP   LOOP     ;須用長程跳躍
②            ;*****
            ORG    00H     ;副程式由 00H 開始
            DELAY
            LJMP   DELAY1   ;長程跳到副程式
③            ;*****
            ORG    300H    ;副程式在第 3 頁
            DELAY1
            ;直接回主程式
            執行副程式
            RET

```

## (四)、大容量副程式及列表資料的修改範例

PIC16C84 主程式及副程式各佔用 1 頁，它的副程式位址不受限制，如程式 2-7-1.ASM。

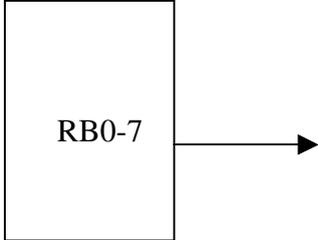
```

;***** 2-7-1.ASM *****
;
;   PIC16C84
;
;   ┌───────────┐
;   │           │
;   │   RB0-7   │ ────>
;   │           │
;   └───────────┘
;
;*****
;
;   DEVICE PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG    0CH ;暫存器的位址由 0CH 開始
DELY1  DS    1 ;延時暫存器
DELY2  DS    1 ;延時暫存器
;*****8*****
;
;   ORG    00H ;主程式由 00 開始
START
;   MOV    !RB,#00 ;設定 RB 為輸出
;   CLR    RB ;RB 輸出 0
LOOP
;   CALL   DELAY ;呼叫延時副程式
;   INC    RB ;RB+1
;   JMP    LOOP ;跳到 LOOP 繼續
;*****
;
;   ORG    300H ;副程式由 300H 開始
DELAY
;   MOV    DELY1,#00
DEL     MOV    DELY2,#00
;   DJNZ   DELY2,$
;   DJNZ   DELY1,DEL
;   RET
;*****
;
;   END

```

PIC16C54 主程式及副程式各佔用 1 頁，它的副程式位址受限於第 0 頁，如程式 2-7-2.ASM。

```

;***** 2-7-2.ASM *****
;
;   PIC16C84
;
;   
;
;   *****
;
;   DEVICE PIC16C54,XT_OSC,WDT_OFF,PROTECT_OFF
;   ORG     07H ;暫存器的位址由 07H 開始
DELY1 DS   1 ;延時暫存器
DELY2 DS   1 ;延時暫存器
;
;   *****
;
;   RESET  START ;由啓始位址跳到 START
;   ORG    100H ;主程式由 100H 開始
START
;   MOV    !RB,#00 ;設定 RB 爲輸出
;   CLR    RB ;RB 輸出 0
LOOP
;   CALL   DELAY ;呼叫延時副程式
;   INC    RB ;RB+1
;   JMP    LOOP ;跳到 LOOP 繼續
;
;   *****
;
;   ORG    00H ;副程式由 00H 開始
DELAY
;   MOV    DELY1,#00
DEL     MOV    DELY2,#00
;   DJNZ   DELY2,$
;   DJNZ   DELY1,DEL
;   RET
;
;   *****
;
;   END

```



```
        DJNZ     DELY1,DEL
        RET
;*****
        ORG     300H           ;列表資料第 3 頁內
;
        TABLE ; 76543210
        JMP     PC+W
        RETW   1000000B
        RETW   0100000B
        RETW   0010000B
        RETW   0001000B
        RETW   0000100B
        RETW   0000010B
        RETW   0000001B
        RETW   0000000B
;*****
        END
```



```
JMP    PC+W
RETW   10000000B
RETW   01000000B
RETW   00100000B
RETW   00010000B
RETW   00001000B
RETW   00000100B
RETW   00000010B
RETW   00000001B
RETW   00000000B
;*****
;
    ORG    300H           ;真正副程式由 300H 開始
DELAY1
    MOV    DELY1,#00
DEL    MOV    DELY2,#00
    DJNZ   DELY2,$
    DJNZ   DELY1,DEL
    RET
;*****
    END
```

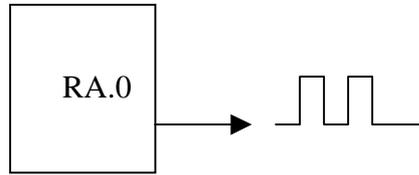
## 學習評量七：

將下列 PIC16C84 的程式，改為 PIC16C54 的程式。

```

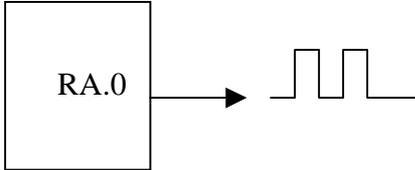
;*****
;
;      PIC16C84
;
;
;
;
;
;
;
;
;
;
;*****
          DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
          ORG     0CH ;使用者 RAM 啓始位址
DELY1 DS    1 ;延時暫存器
DELY2 DS    1 ;延時暫存器
;*****
          ORG     00H ;程式啓始位址
START
          MOV     !RA,#00 ;設 RA 爲輸出
LOOP
          CALL    DELAY
          XOR     RA,#00000001B ;令 RA.0 反相
          JMP     LOOP ;2
;*****
DELY1 ;延時時間隨各人的工作頻率而不同
          MOV     DELY2,#100 ;2
          DJNZ    DELY2,$ ;3*99+2
          RET     ;2
;*****
          END

```



## 學習評量七答案：

```

;*****
;
;   PIC16C54
;
;   
;
;
;
;
;
;
;*****
DEVICE PIC16C54,XT_OSC,WDT_OFF,PROTECT_OFF
      ORG    07H      ;暫存器的位址由 07H 開始
DELY1  DS    1
DELY2  DS    1
;*****
;
      RESET  START ;由啓始位址跳到 START
      ORG    00H      ;程式啓始位址
START
      MOV    !RA,#00      ;設 RA 爲輸出
LOOP
      CALL   DELAY
      XOR    RA,#0000001B ;令 RA.0 反相
      JMP    LOOP        ;2
;*****
;
DELY1                                     ;延時時間隨各人的工作頻率而不同
      MOV    DELY2,#100    ;2
      DJNZ   DELY2,$        ;3*99+2
      RET                                     ;2
;*****
;
      END

```

假如你的答案與上述之重點相似，請翻下一頁，假如你的答案不與上述之重點相似，請翻至第 77 頁重新閱讀以便發現你的錯誤之處，並將錯誤改正。

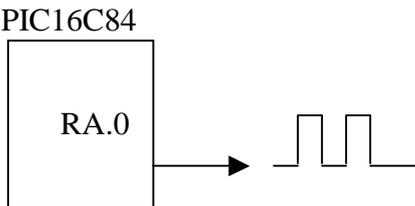
## 學後評量

將下列 PIC16C84 的程式，改為 PIC16C54 的程式。

```

;*****
;
;      PIC16C84
;
;
;
;
;
;
;
;
;*****
      DEVICE  PIC16C84,XT_OSC,WDT_OFF,PROTECT_OFF
      ORG     0CH ;使用者 RAM 啓始位址
DELY1  DS    1   ;延時暫存器
DELY2  DS    1   ;延時暫存器
;*****
      ORG     00H           ;程式啓始位址
START
      MOV     !RA,#00       ;設 RA 爲輸出
LOOP
      CALL    DELAY
      XOR     RA,#0000001B ;令 RA.0 反相
      JMP     LOOP         ;2
;*****
DELY1                                     ;延時時間隨各人的工作頻率而不同
      MOV     DELY2,#100   ;2
      DJNZ   DELY2,$       ;3*99+2
      RET                                     ;2
;*****
      END

```



**我的工作計畫**

作業名稱：\_\_\_\_\_

工作開始日期：\_\_\_\_\_ 完成日期：\_\_\_\_\_

工作時間：\_\_\_\_\_小時 教師認可：\_\_\_\_\_

我製作上列工作時所需用之工具及機器：

1 \_\_\_\_\_ 5 \_\_\_\_\_ 9 \_\_\_\_\_

2 \_\_\_\_\_ 6 \_\_\_\_\_ 10 \_\_\_\_\_

3 \_\_\_\_\_ 7 \_\_\_\_\_ 11 \_\_\_\_\_

4 \_\_\_\_\_ 8 \_\_\_\_\_ 12 \_\_\_\_\_

**我所需要的材料及消耗品：**

名 稱	說 明	規 格	數 量	估 價

**我計畫如何做我的作業：**

工作步驟	安全注意事項	工作時注意要項

注意：(1)現在你已完成你的作業計畫，請不要馬上工作，你先檢討一下，有沒有其他更好的方法呢？有沒有遺漏呢？將你的計畫送給你的老師認可；然後再開始工作，工作時間為二小時。

(2)當你做好了作業，請對你的成品做自我評價（Self-evaluation），然後送交老師評分。

## 一、我對我作業之評分

### (一) 正確度共 70%

部份	滿分	動作	評 分 標 準
A	25%	寫出搬移及輸出程式	動作正確
B	15%	算術動作寫出程式	動作正確
C	15%	寫出邏輯動作輸出程式	動作正確
T	10%	寫出列表法輸出程式	動作正確
60°	5%	寫出輸入動作程式	動作正確

### (二) 跨頁程式的撰寫 30%

部位	處數	滿分	扣分標準	得分
		5		

我的作業評分 = 正確度 + 程式的撰寫 = \_\_\_\_分，屬於\_\_\_\_等

A=95 分以上 B=85 分以上 C=75 分以上

D=65 分以上 E=64 分以上

二、我的工作計畫得分\_\_\_\_分，屬於\_\_\_\_等。

三、安全習慣得分\_\_\_\_分，屬於\_\_\_\_等。

四、工作精神與學習態度得分\_\_\_\_分，屬於\_\_\_\_等。

### 五、教師評分

(一) 作業得分 \_\_\_\_\_ 3.安全習慣\_\_\_\_\_

(二) 工作計畫\_\_\_\_\_ 4. 工作精神與學習態度得分\_\_\_\_\_

總得分\_\_\_\_\_屬於\_\_\_\_等

### 六、時間
