

# 輸出入埠之進階應用: Part I 四個七段顯示器掃描式輸出

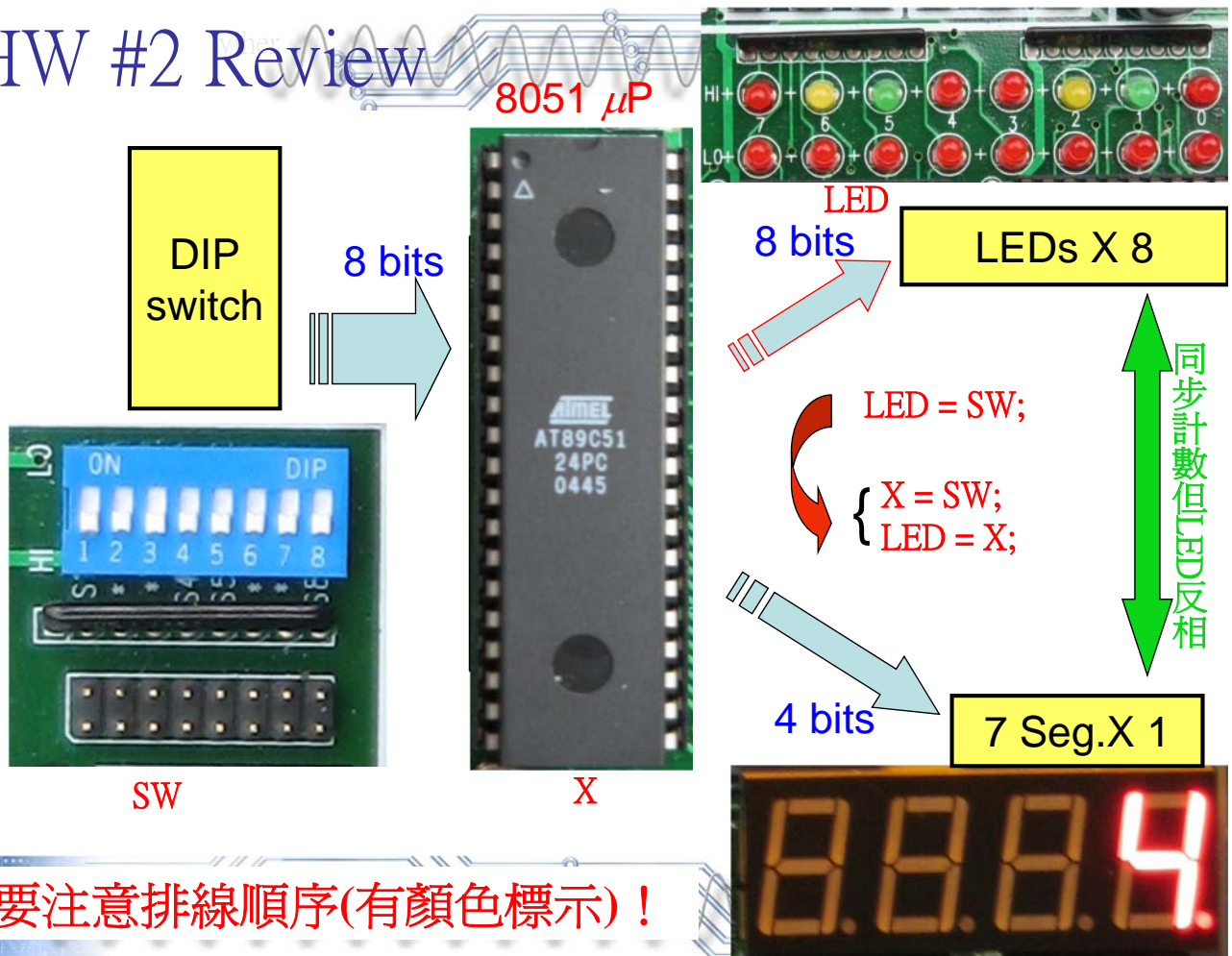
## 第五章(Part I)

2012/10/30

例說89S51-C語言

5-1

### HW #2 Review



# HW #2 Review (cont.)

- 十進位: 254
- 八進位: 0376
- 16進位: 0xfe
  - 指的是二進位數字1111 1110 (C51無2進位寫法)

## • 判斷某單一開關或按鍵的方法:

- if (P1 == 0x01)      ←不恰當
- if ((P1 & 0x01) == 1)      →更好的寫法if ((P1 & 0x01) != 0)
- 例如第2個開關: if ((P1 & 0x02) != 0)

2012/10/30

例說89S51-C語言

5-3

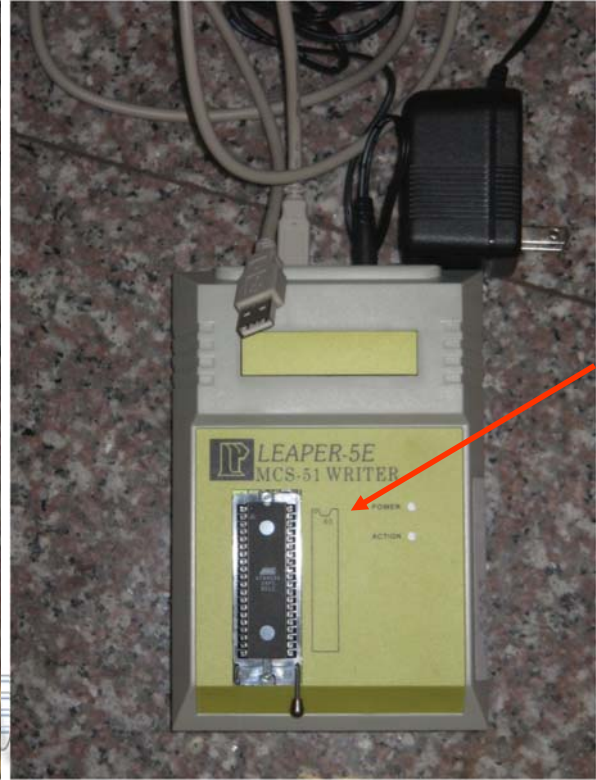
## 2-5 運算子-7

### • 運算子 優先順序

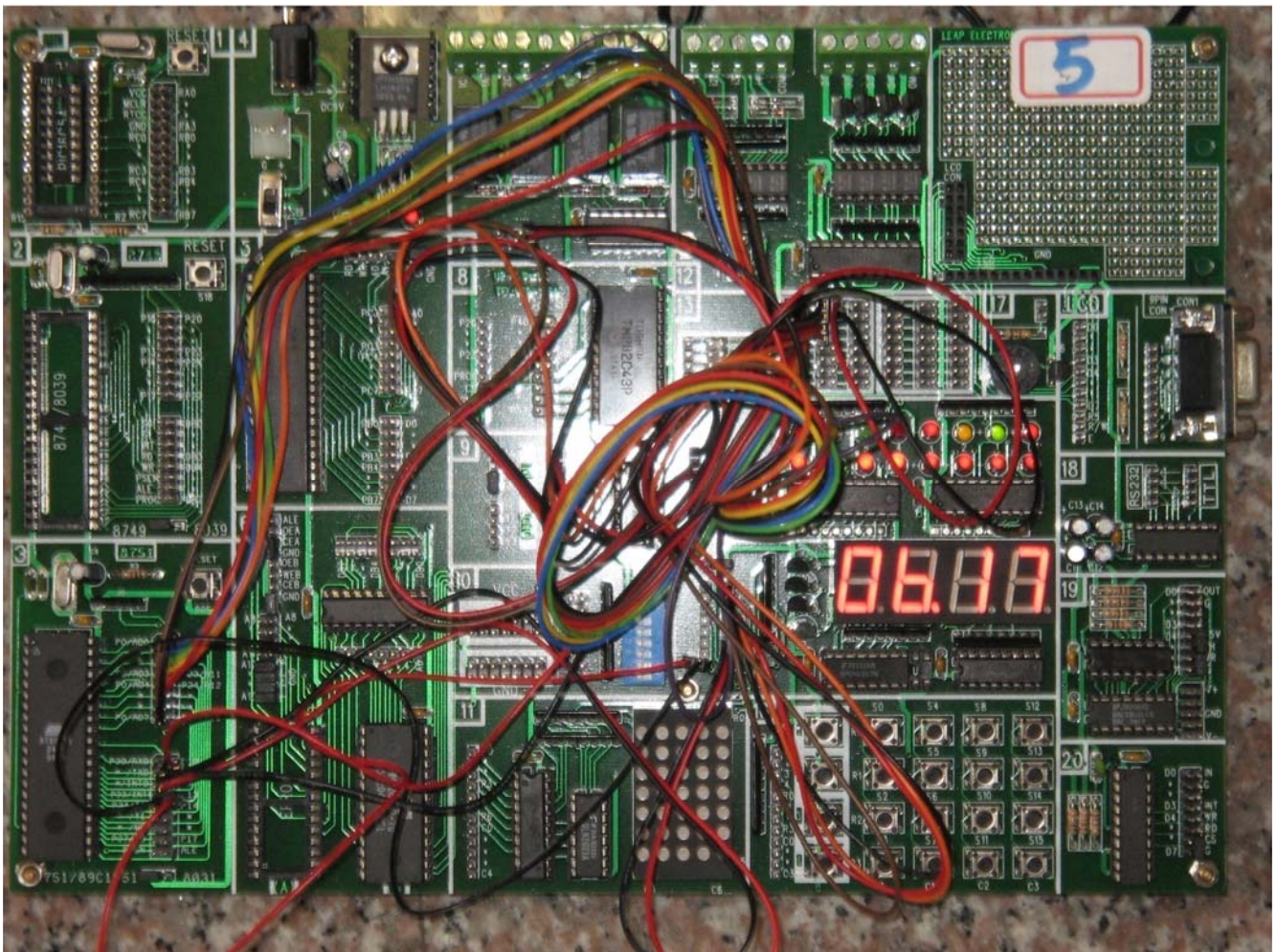
優先順序	運算子或操作符號	說明
1	(、)	小括號
2	~、!	補數、反相運算
3	++、--	遞增、遞減
4	*、/、%	乘、除、取餘數
5	+、-	加、減
6	<<、>>	左移、右移
7	<、>、<=、>=、==、!=	關係運算子
8	&	布林運算—AND
9	^	布林運算—XOR
10		布林運算—OR
11	&&	邏輯運算—AND
12		邏輯運算—OR
13	=、*=、/=、%=、+=、--、<<=、>>=、&=、^=、 =	指定運算子

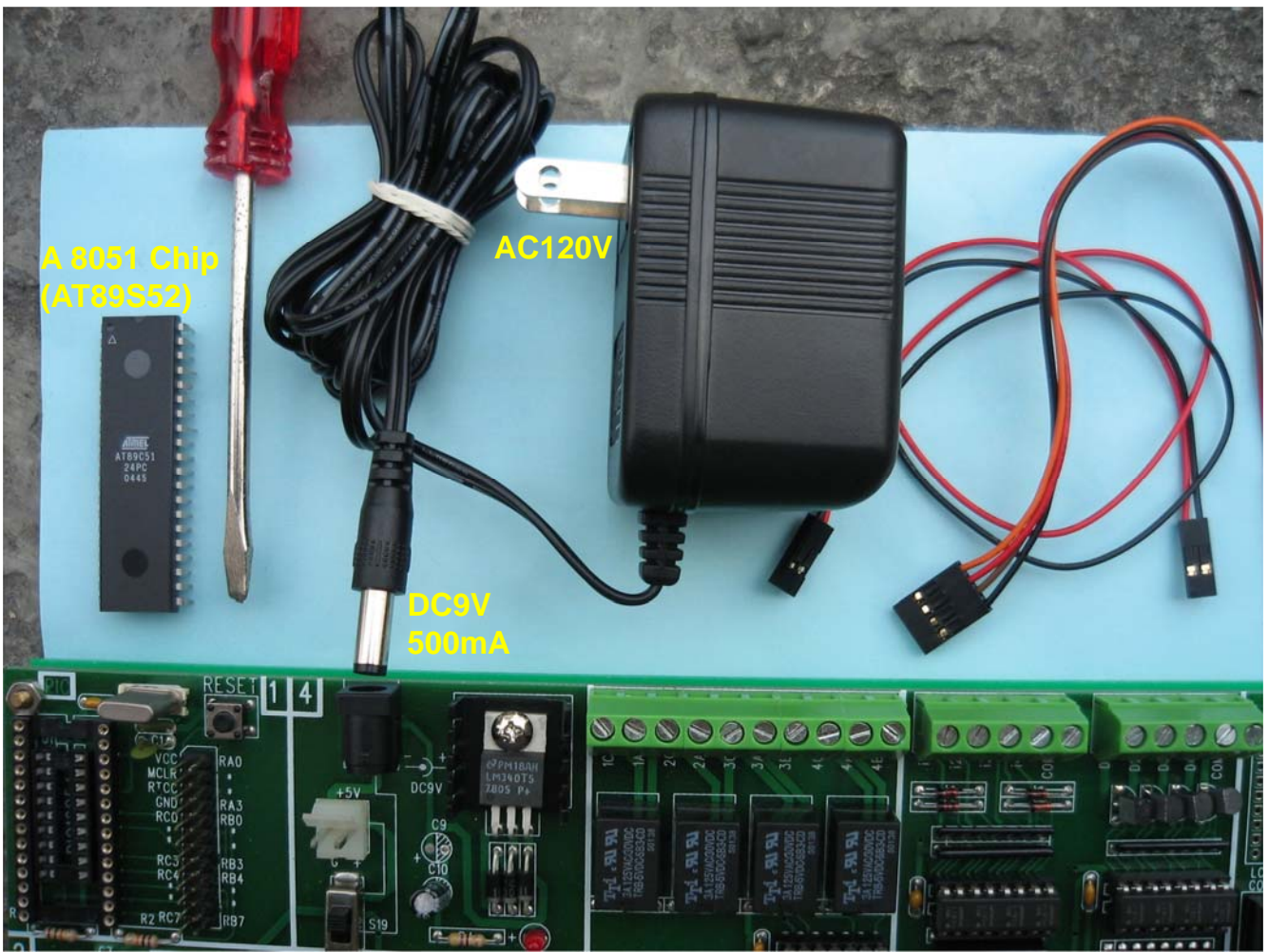
2012/10/30

# LEAPER-5E MCS-51 Writer

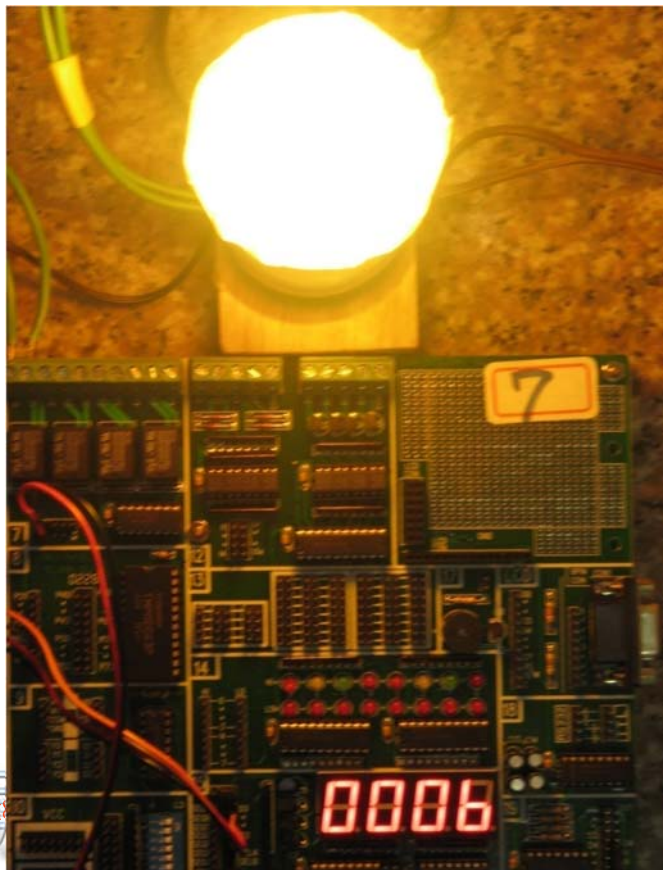


注意IC凹槽方向





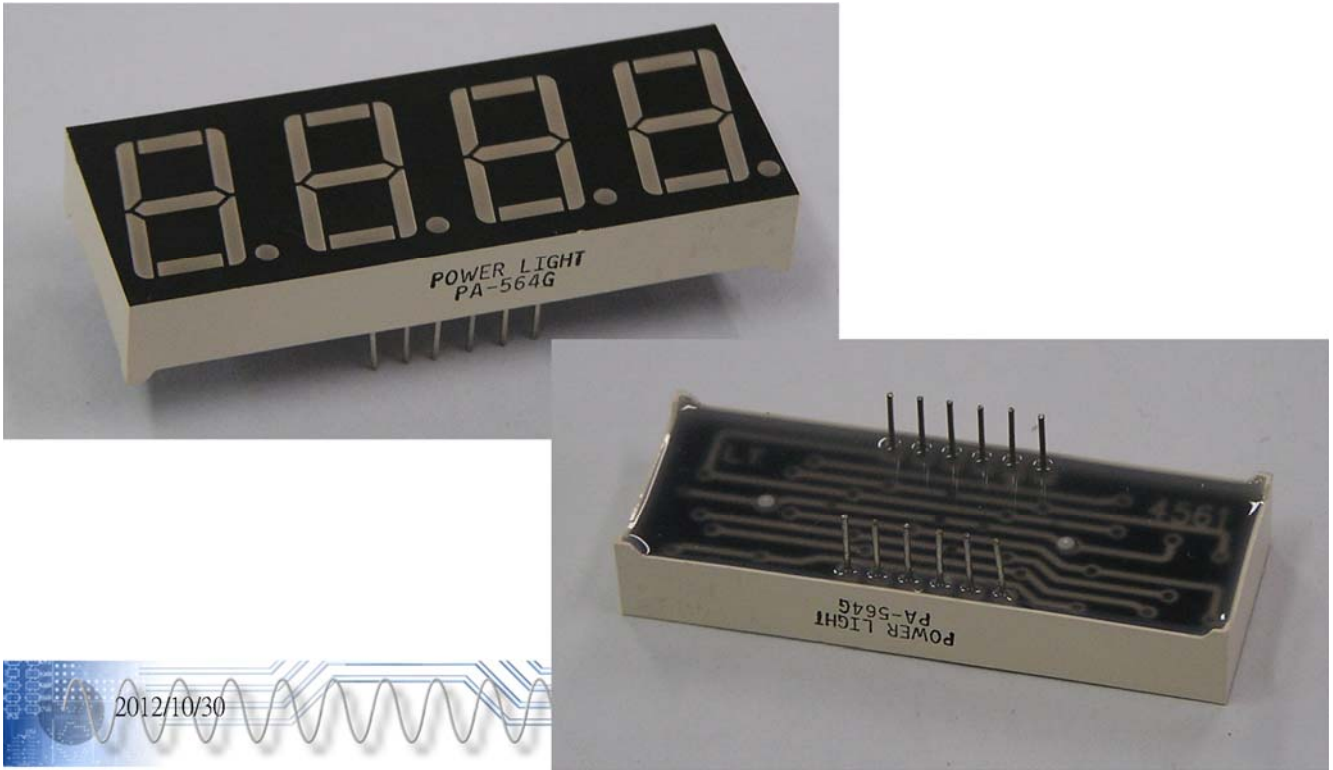
## 定時裝置(4位數計時器)



例說

yiher

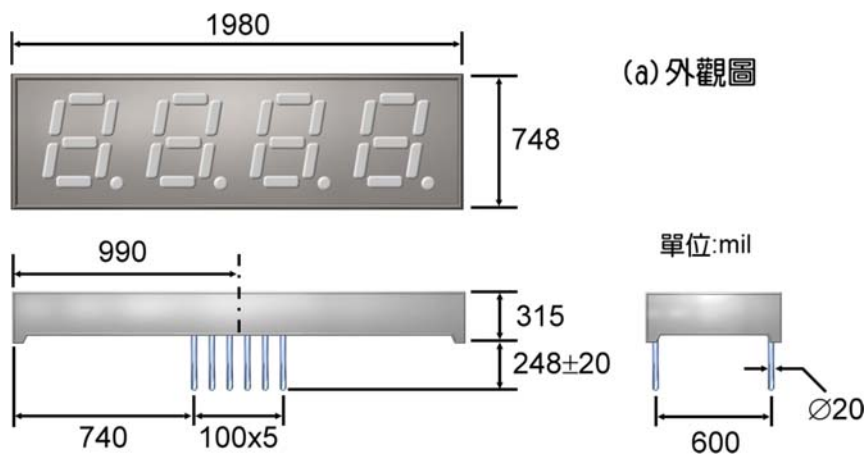
# 四位數七節顯示器模組 (左為正面圖、右為背面圖)



2012/10/30

yiher

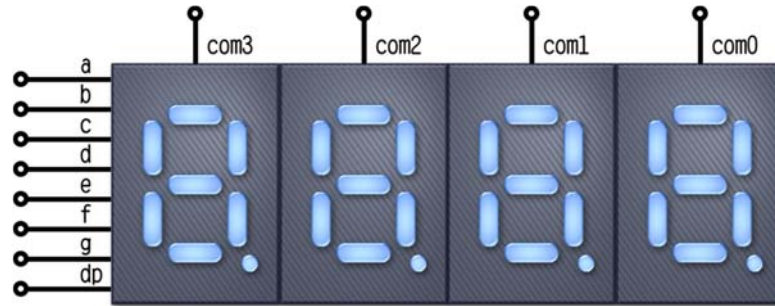
# 市售四位數七節顯示器模組



2012/10/30

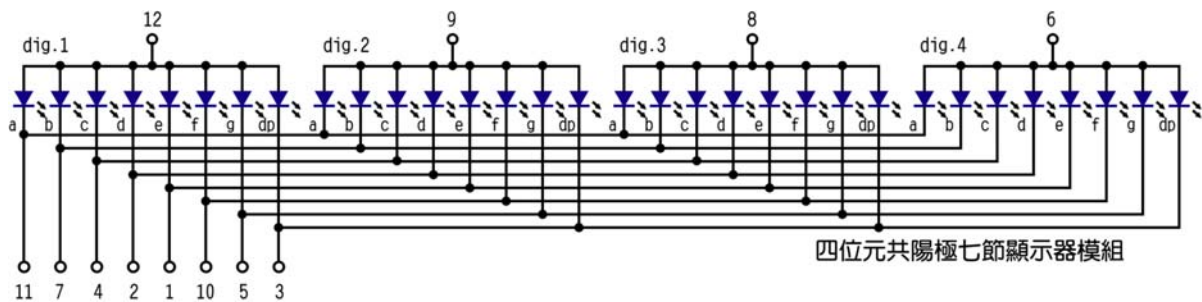
例說89S51-C語言

# 七節顯示器模組

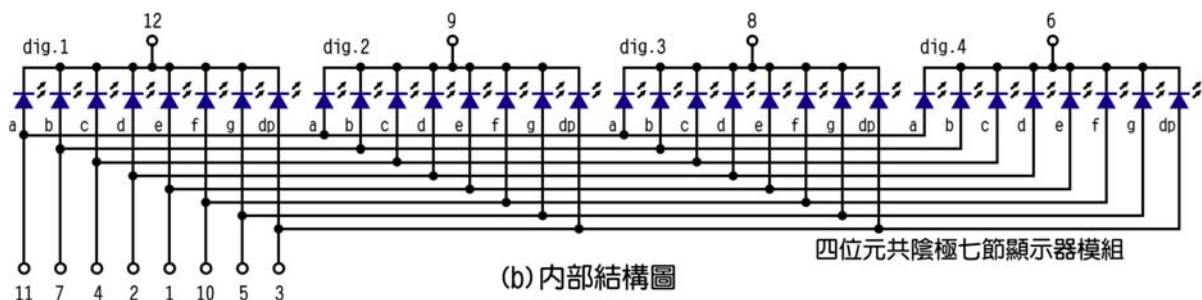


(a) 七節顯示器模組

# 市售四位數七節顯示器模組



四位元共陽極七節顯示器模組



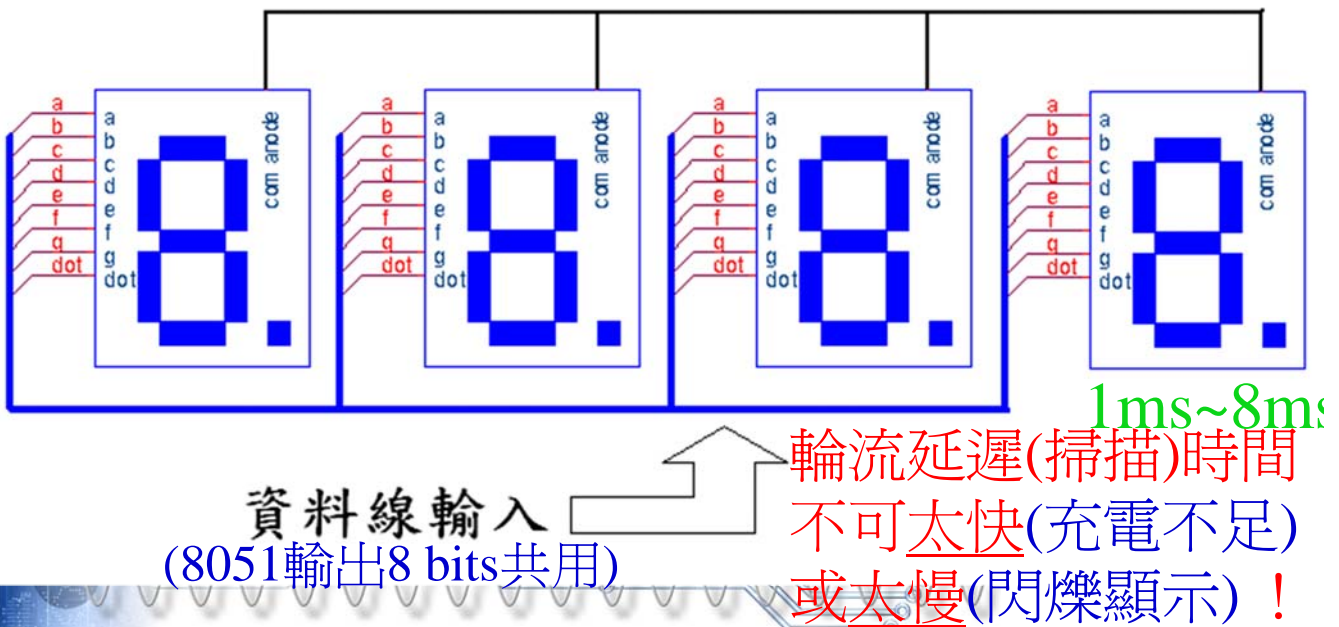
四位元共陰極七節顯示器模組

(b) 內部結構圖

# 掃描式輸出原理

選擇掃描線  
(四選一)

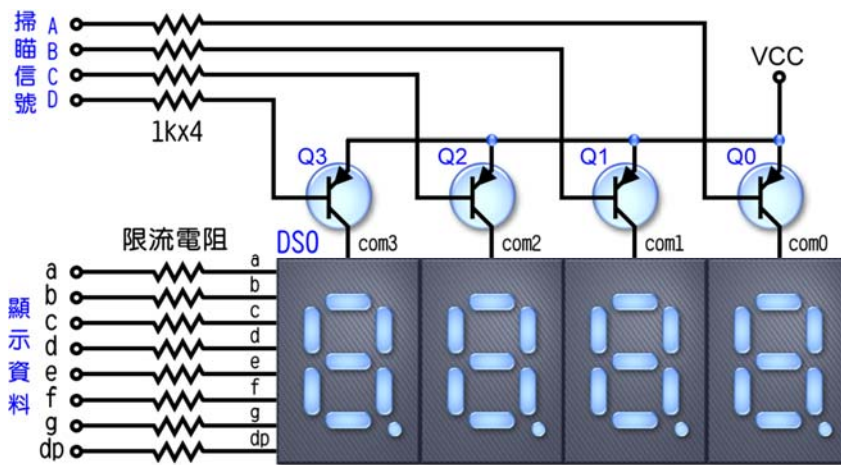
輪流！



## 四個七段顯示器-掃描式輸出原理

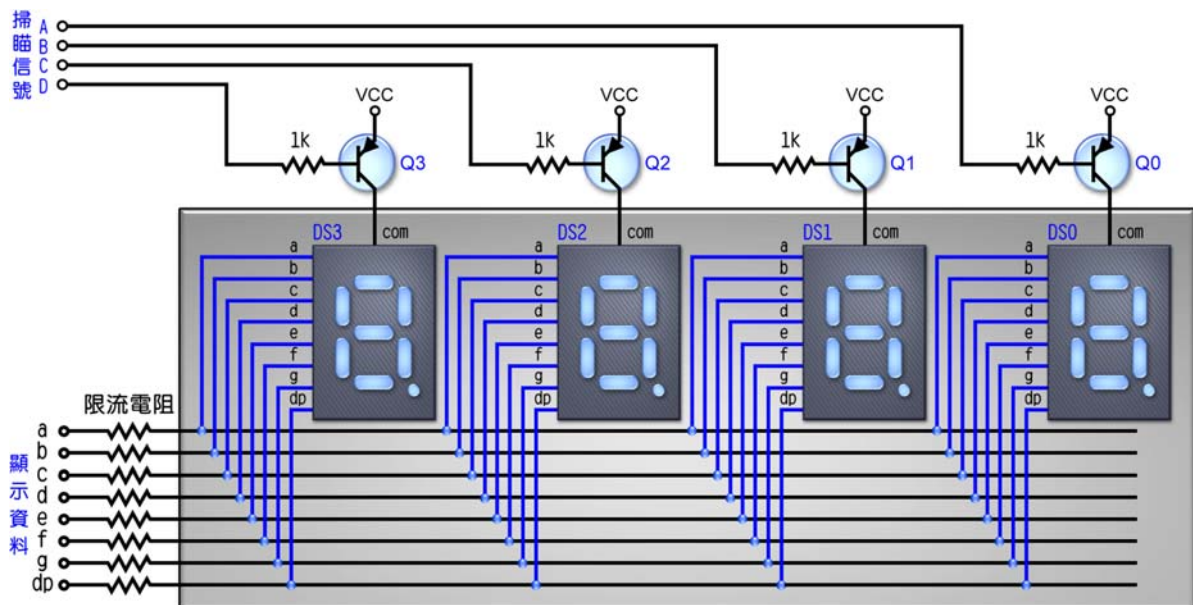
- 掃描式輸出的基本原理是，將輸出埠區分為輸出資料線與選擇掃描線。每個七段顯示器的共用輸出資料線(8位元)，每個七段顯示器再單獨使用一個位元的選擇掃描線，做為選擇輸入的開關，在一特定時間內，只有其中一個掃描選擇線會開啓，讓資料指定輸出到其中一個七段顯示器。選擇掃描線會以掃描方式，輪流循環的輸出到每一個七段顯示器上，在人的視覺暫留時間範圍內，感覺可以同時看到多個七段顯示器顯示的數字資料，達到以最少的輸出埠，同時輸出多個七段顯示器的目的。)

# 七節顯示器模組



(b) 七節顯示器模組之應用

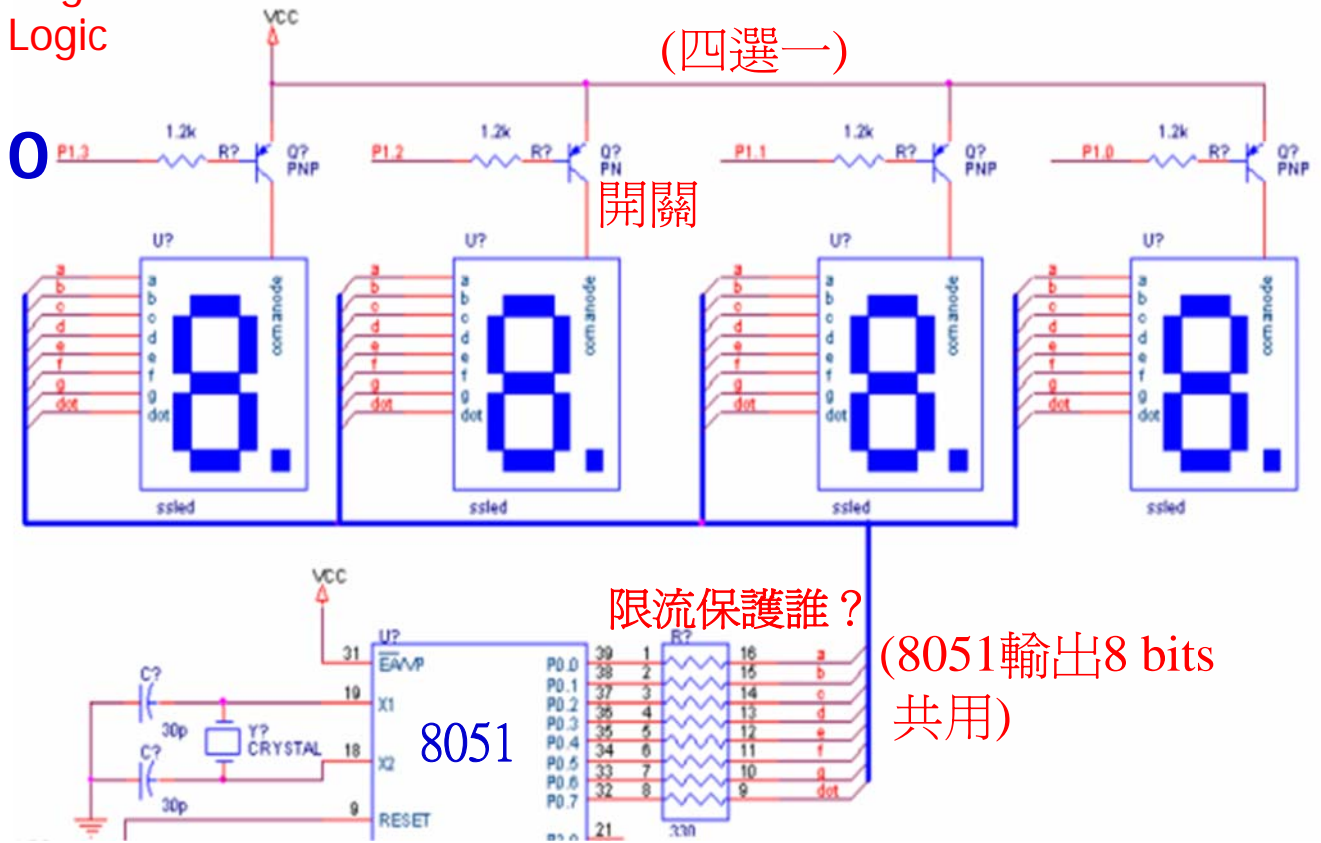
# 四個共陽極七節顯示器



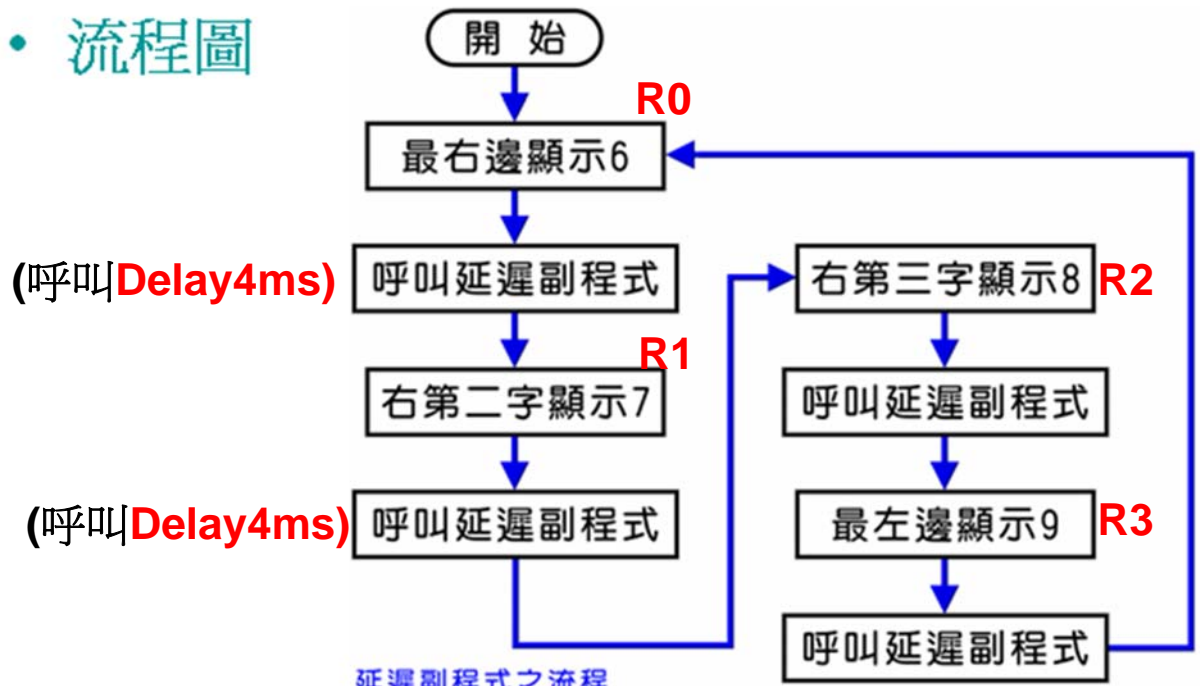


# 四個七段顯示器的電路

Negative Logic



## • 流程圖



例說 8051

```
//檔名:ch7C_3.C
```

```
#include <reg51.h>
```

```
unsigned char code seg7[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
```

```
unsigned char code scanline[]={0xf7,0xfb,0xfd,0xfe};
```

```
void delay1ms(int count);
```

```
main()
```

```
{
```

```
char i;
```

```
while(1){
```

```
for(i=0;i<4;i++){
```

```
    P1=0xff;
```

```
    P0=seg7[i];
```

```
    P1=scanline[i];
```

```
    delay1ms(1);
```

```
}
```

```
}
```

```
}
```

```
void delay1ms(int count)
```

```
{
```

```
int i,j;
```

```
for(i=0;i<count;i++) //透過引數 count 控制外迴圈數
```

```
    for(j=0;j<122;j++);
```

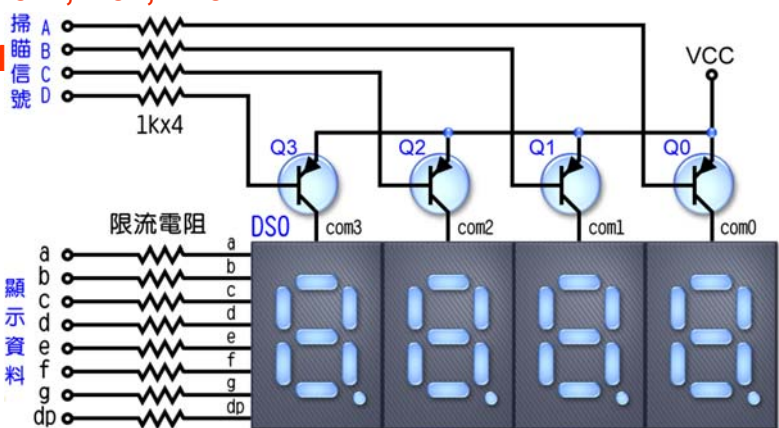
```
}
```

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

0111,1011,1101,1110

P1  
掃描信號  
A  
B  
C  
D

P0  
顯示資料  
a  
b  
c  
d  
e  
f  
g  
dp

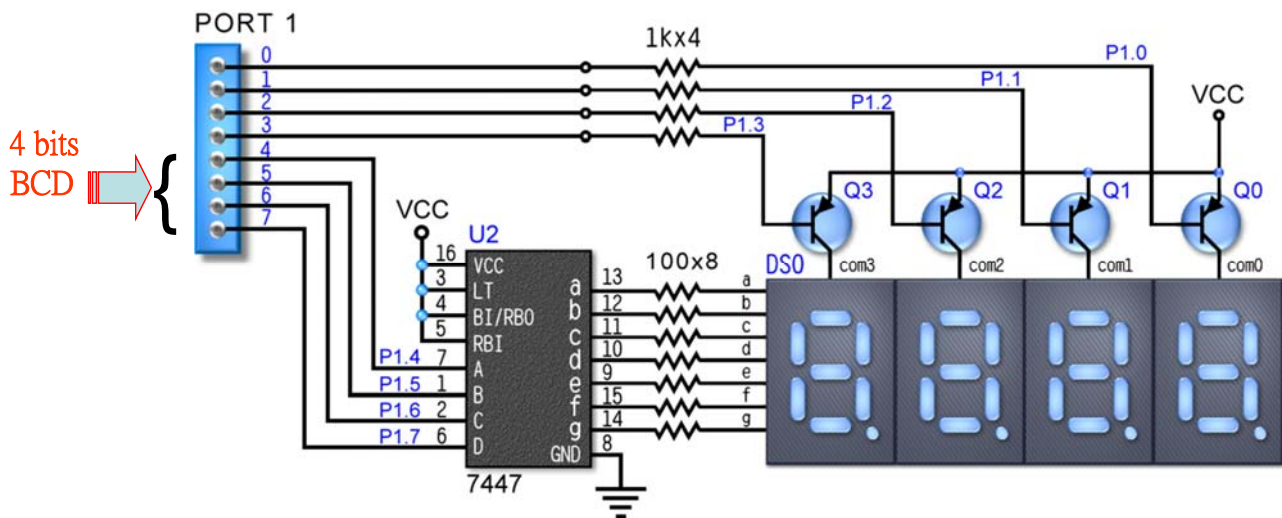


(b) 七節顯示器模組之應用

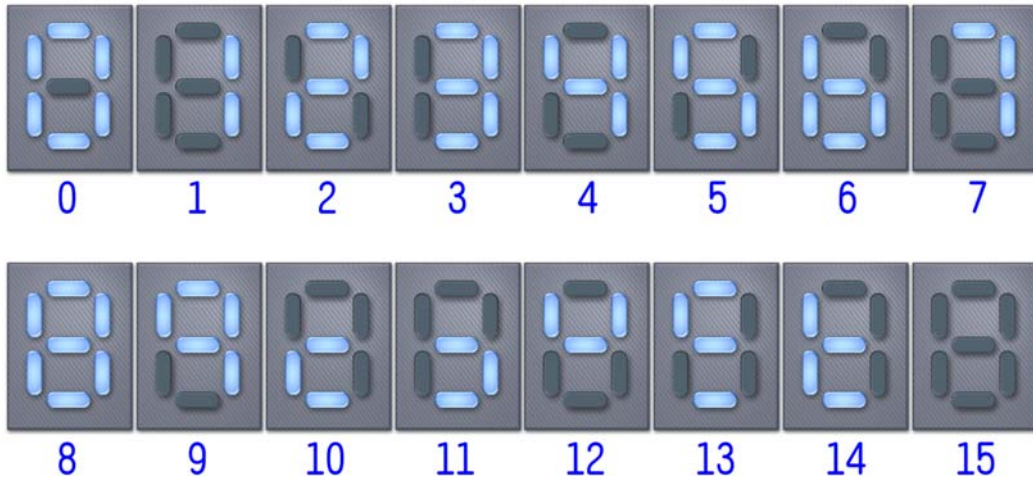
## 七段顯示器

## 顯示1234四位數字

# 使用7447解碼器之七節顯示器模組 驅動電路圖之一



# 七節顯示器

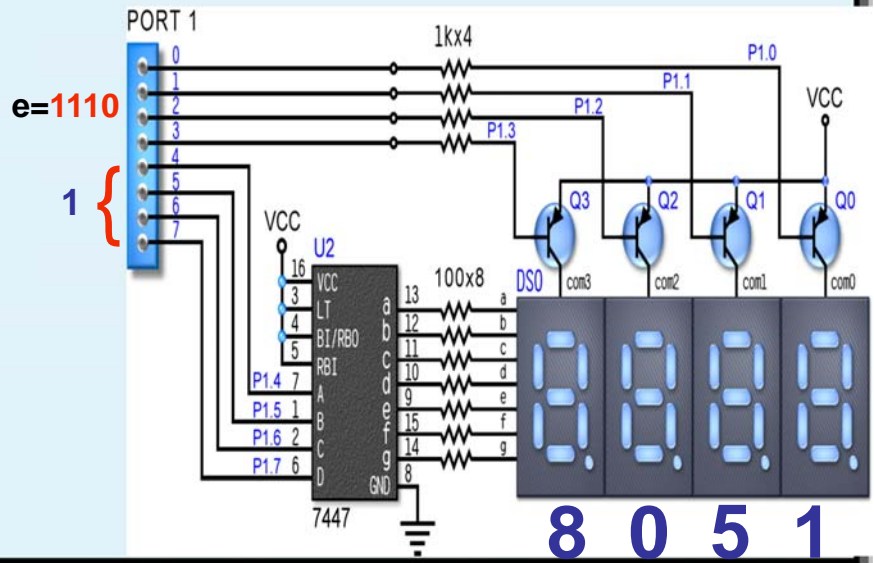


# 程式

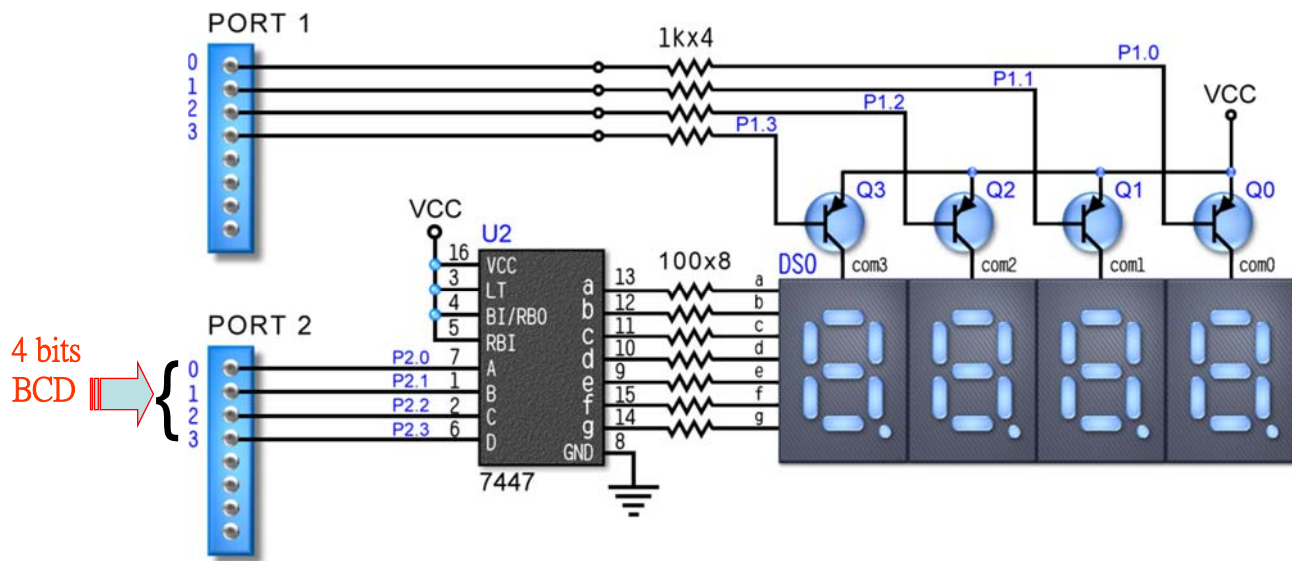
```

main()
{
  while(1)
  {
    P1=0x1e;
    delay1ms(4);
    P1=0x5d;
    delay1ms(4);
    P1=0x0b;
    delay1ms(4);
    P1=0x87;
    delay1ms(4);
  }
}

```



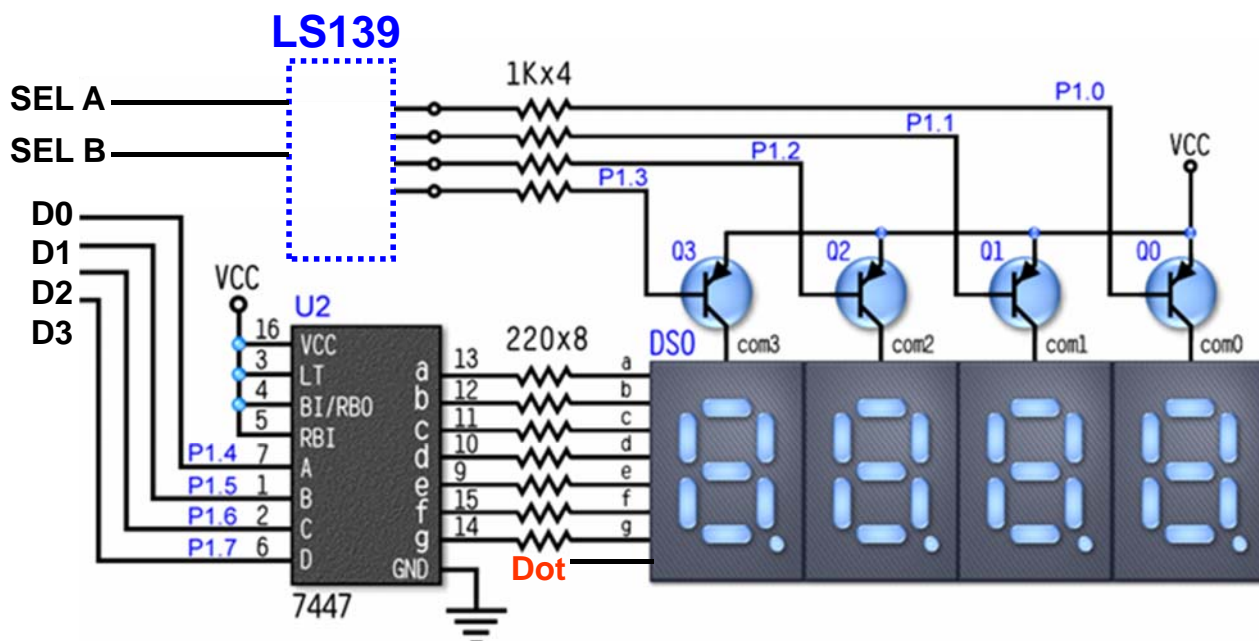
# 使用7447解碼器之七節顯示器模組 驅動電路圖之二



2012/10/30

例說89S51-C語言

5-31

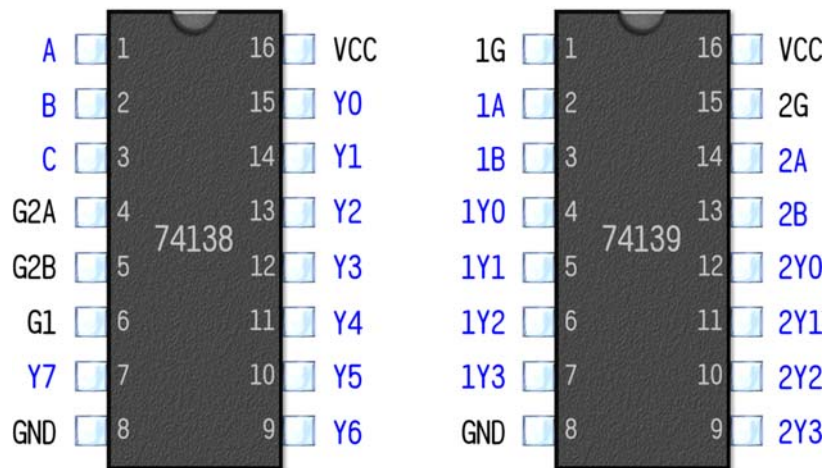


2012/10/30

例說89S51-C語言

5-34

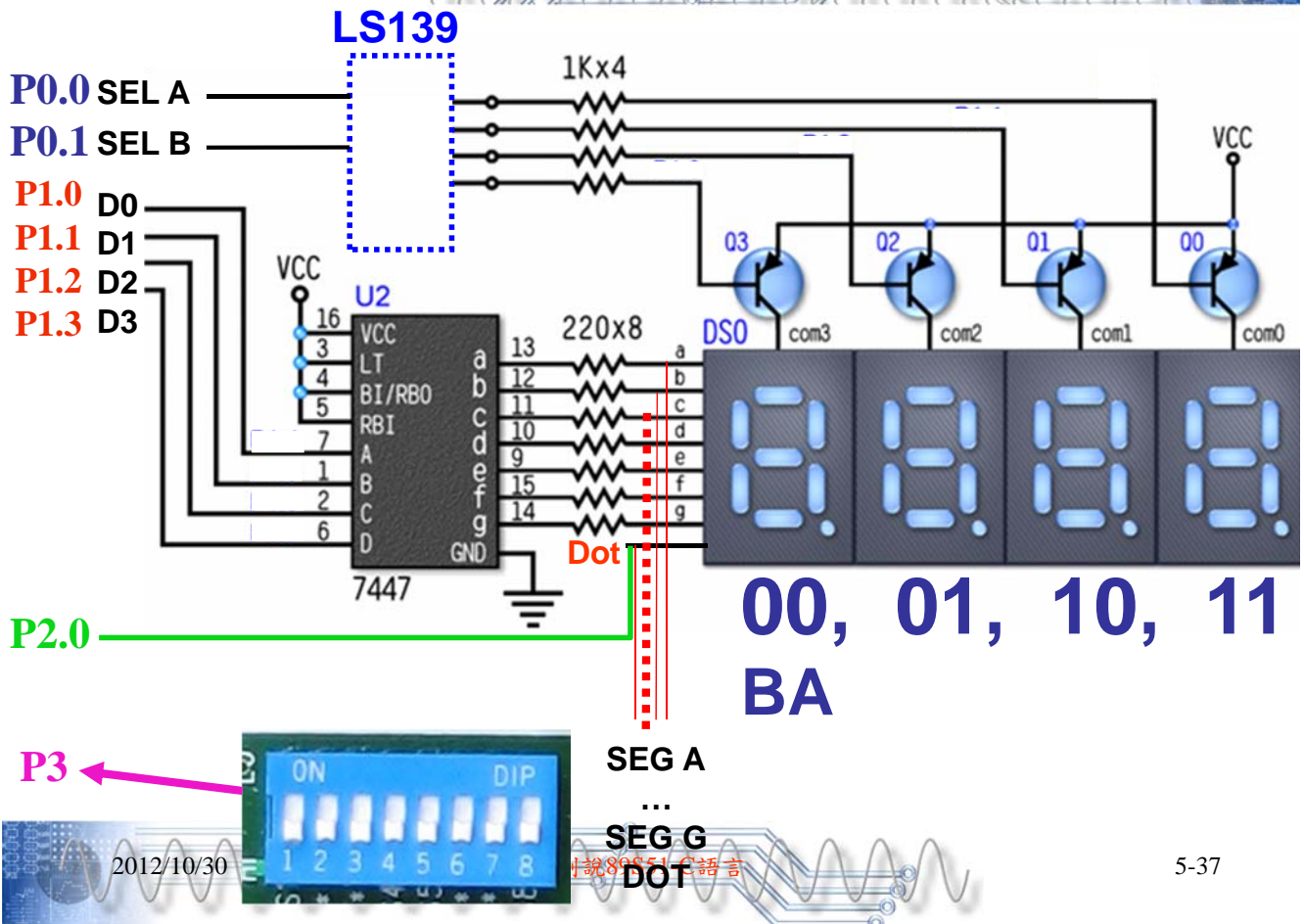
# 74139、74138之接腳



# 74139真值表

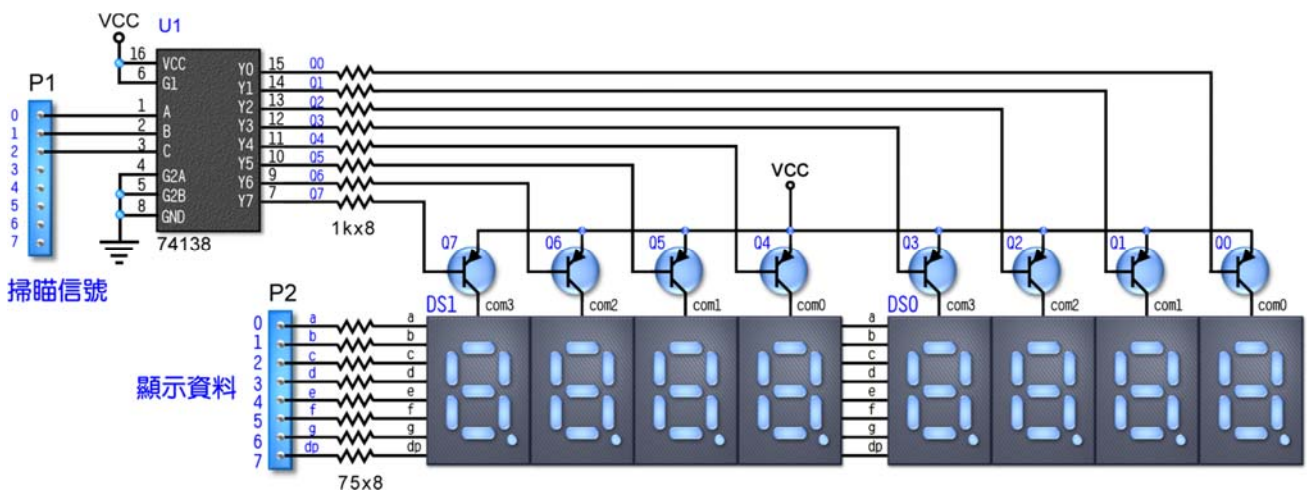
輸 入			輸 出			
$\overline{1G}$	SEL B	SEL A	1Y0	1Y1	1Y2	1Y3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

x：代表可為 0 或 1



5-37

## 使用掃描解碼器



2012/10/30

例說89S51-C語言

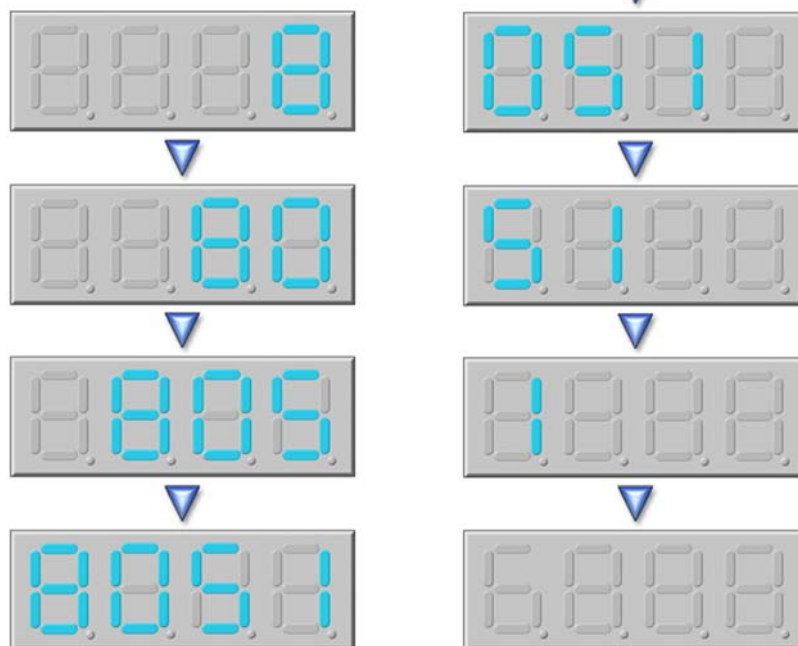
5-42

# 實例演練5-4-3

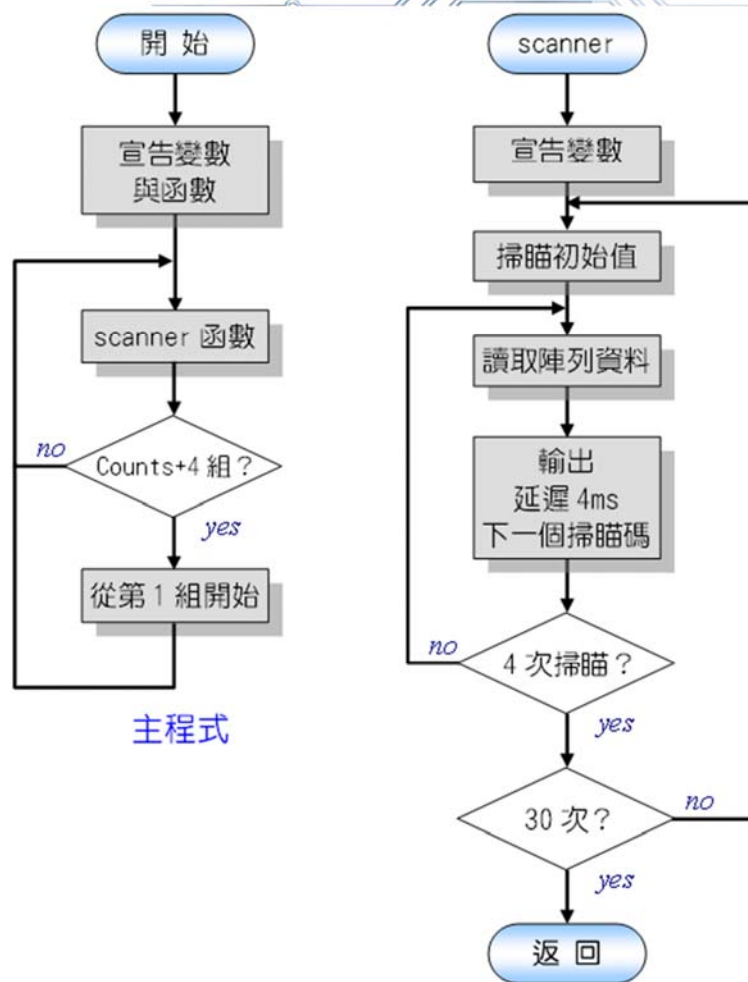
## 動態跑馬燈實驗



## 跑馬燈之分解動作



# 流程圖



2012/10/30

5-47

## 跑馬燈實驗程式(ch05-4-3.c) -1

```

/* ch05-4-3.c - 4 個七節顯示器跑馬燈實驗,P1.0~3 為掃描信號 P2 接七節顯示器 */
//==宣告區=====
#include <reg51.h> // 定義 8051 暫存器之標頭檔,P2-17~19
#define SCANP P1 // 定義掃描碼由 Port 1 輸出
#define SEG7P P2 // 定義七節顯示碼由 Port 2 輸出
char code TAB[11]={ 0xc0, 0xf9, 0xa4, 0xb0, 0x99, // 數字 0-4
                   0x92, 0x83, 0xf8, 0x80, 0x98, 0xbf }; // 數字 5-9,-
#define counts 6 // 宣告字組數量
char disp[counts+7]={10,10,10,10,1,2,3,4,5,6,10,10,10};//----123456---
void delay1ms(int); // 宣告延遲函數
void scanner(char); // 掃描函數
//==主程式=====
main() // 主程式開始
{
    char i; // 宣告變數 i
    while(1) // 無窮迴圈,程式一直跑
        for(i=0;i<counts+4;i++) // 顯示 count 列字組,for 迴圈(字組 i)開始
            scanner(i); // 掃描第 i 列字組
} // 主程式結束
  
```

2012/10/30

例說89S51-C語言

5-48



# 跑馬燈實驗程式(ch05-4-3.c) -2

```

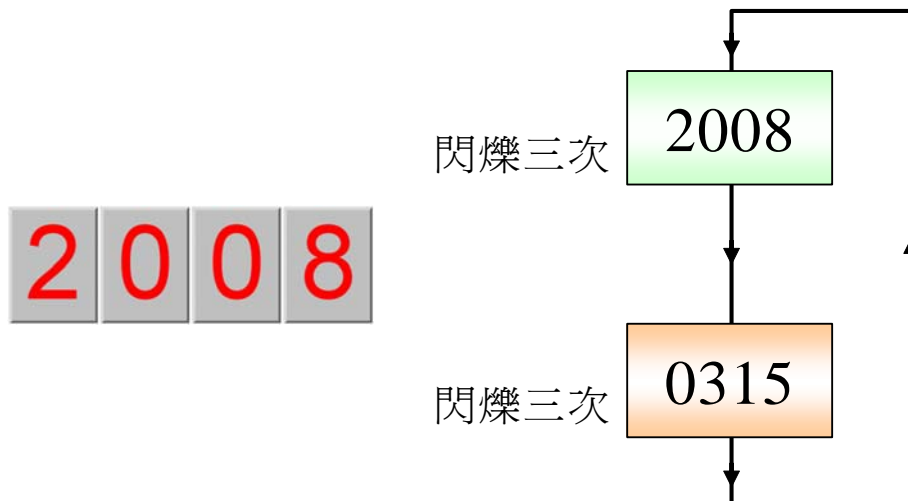
//==副程式=====
/* 延遲函數,延遲約 x*1ms */
void delay1ms(int x)
{
    int i,j;
    for (i=0;i<x;i++)
        for (j=0;j<120;j++);
}
/* 掃描字組函數,顯示第 x 組數字 */
void scanner(char x)
{
    char i,j,BCD,scan;
    for (i=0;i<30;i++)
    {
        scan=0xf7;
        for (j=0;j<4;j++)
        {
            SEG7P=0xff;
            SCANP=scan;
            BCD=disp[x+j];
            SEG7P=TAB[BCD];
            delay1ms(4);
            scan>>=1;
        }
    }
}

```

// 延遲函數開始  
 // 宣告整數變數 i,j  
 // 計數 x 次,延遲 x\*1ms  
 // 計數 120 次,延遲 1ms  
 // 延遲函數結束  
 // 掃描字組函數開始  
 // 宣告變數  
 // 掃描 30 次 i 迴圈  
 // 掃描信號初值 1111 0111  
 // 掃描 4 個數字 j 迴圈  
 // 關閉七段顯示器(防止閃動)  
 // 輸出掃描信號(低 4 位元)  
 // 讀取第 x 組第 j 個數字之 BCD 碼  
 // 輸出至七節顯示器  
 // 延遲 4ms  
 // 產生下個掃描信號  
 // 結束掃描 4 個數字 j 迴圈  
 // 結束掃描 30 次 i 迴圈  
 // scanner 函數結束

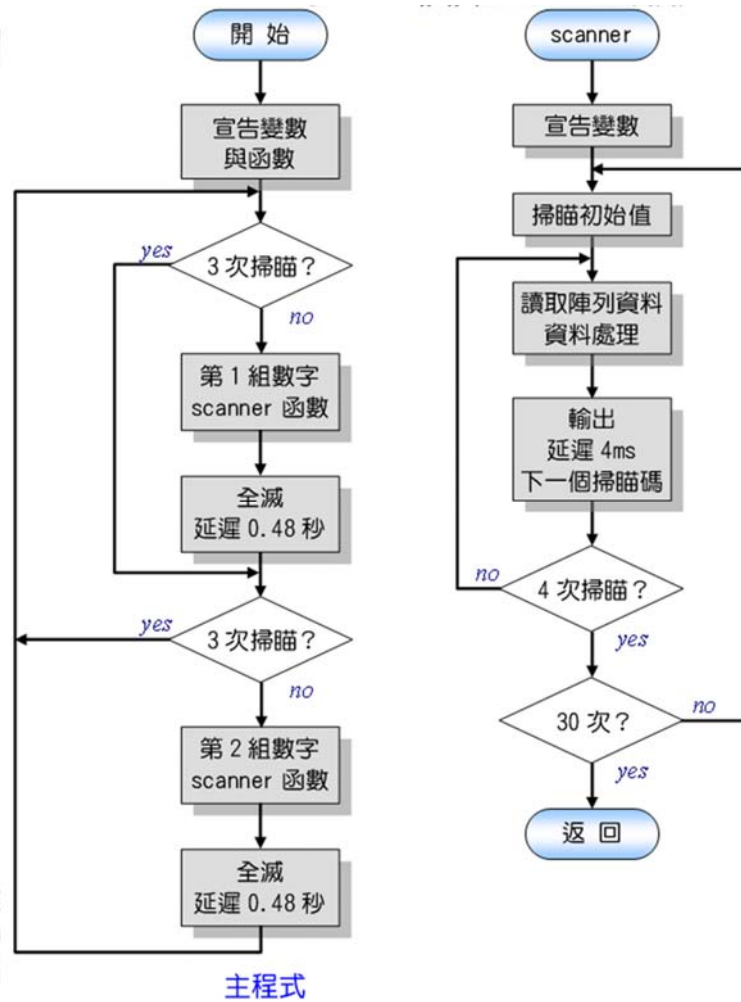
## 實例演練5-4-1

### • 目標



# 流程圖

yiher



2012/10/30

主程式

5-51

## 直接驅動七節顯示器實驗 (ch05-4-2.c) - 1

yiher

```
/* ch05-4-2.c - 直接驅動 4 位七節顯示器實驗,P1.0~3 為掃描信號 P2 接七節顯示器 */  
//==宣告區=====  
#include <reg51.h> // 定義 8051 暫存器之標頭檔,P2-17~19  
#define SCANP P1 // 定義掃描碼由 Port 1 輸出  
#define SEG7P P2 // 定義七節顯示碼由 Port 2 輸出  
char code TAB[10]={ 0xc0, 0xf9, 0xa4, 0xb0, 0x99, //數字 0-4  
                   0x92, 0x83, 0xf8, 0x80, 0x98 };//數字 5-9  
char code disp[2][4]={ {2,0,0,8}, // 顯示資料(第 0 組)  
                       {0,3,1,5}}; // 顯示資料(第 1 組)  
void delay1ms(int); // 宣告延遲函數  
void scanner(char); // 掃描函數
```



2012/10/30

例說89S51-C語言

5-52

# 直接驅動七節顯示器實驗 (ch05-4-2.c) -2

```
//==主程式=====
main()                // 主程式開始
{  char i,j;         // 宣告變數 i,j
  while(1)           // 無窮迴圈,程式一直跑
  {  for(i=0;i<2;i++) // 顯示第 0,1 列字組,for 迴圈(字組 i)開始
    {  for(j=0;j<3;j++) // 閃爍三次
      {  scanner(i);   // 掃描第 i 列字組
        SCANP=0xff;   // 全滅
        delay1ms(480); // 延遲 480×1m=0.48s
      }
    }
  }
}                    // while 迴圈結束
                    // 主程式結束
//==副程式=====
/* 延遲函數,延遲約 x×1ms */
void delay1ms(int x) // 延遲函數開始
{  int i,j;         // 宣告整數變數 i,j
  for (i=0;i<x;i++) // 計數 x 次,延遲 x×1ms
    for (j=0;j<120;j++); // 計數 120 次,延遲 1ms
}                    // 延遲函數結束
```

2012/10/30

例說89S51-C語言

5-53

# 直接驅動七節顯示器實驗 (ch05-4-2.c) -3

```
/* 掃描字組函數,顯示第 x 組數字 */
void scanner(char x) // 掃描字組函數開始
{  char i,j,BCD,scan; // 宣告變數
  for (i=0;i<30;i++) // 掃描 30 次 i 迴圈
  {  scan=0xf7;      // 掃描信號初值 1111 0111
    for (j=0;j<4;j++) // 掃描 4 個數字 j 迴圈
    {  SEG7P=0xff;   // 關閉七段顯示器(防止閃動)
      SCANP=scan;   // 輸出掃描信號(低 4 位元)
      BCD=disp[x][j]; // 讀取第 x 組第 j 個數字之 BCD 碼
      SEG7P=TAB[BCD]; // 輸出至七節顯示器
      delay1ms(4);  // 延遲 4ms
      scan>>=1;    // 產生下個掃描信號
    }
  }
}                    // 結束掃描 4 個數字 j 迴圈
                    // 結束掃描 30 次 i 迴圈
                    // scanner 函數結束
```

2012/10/30

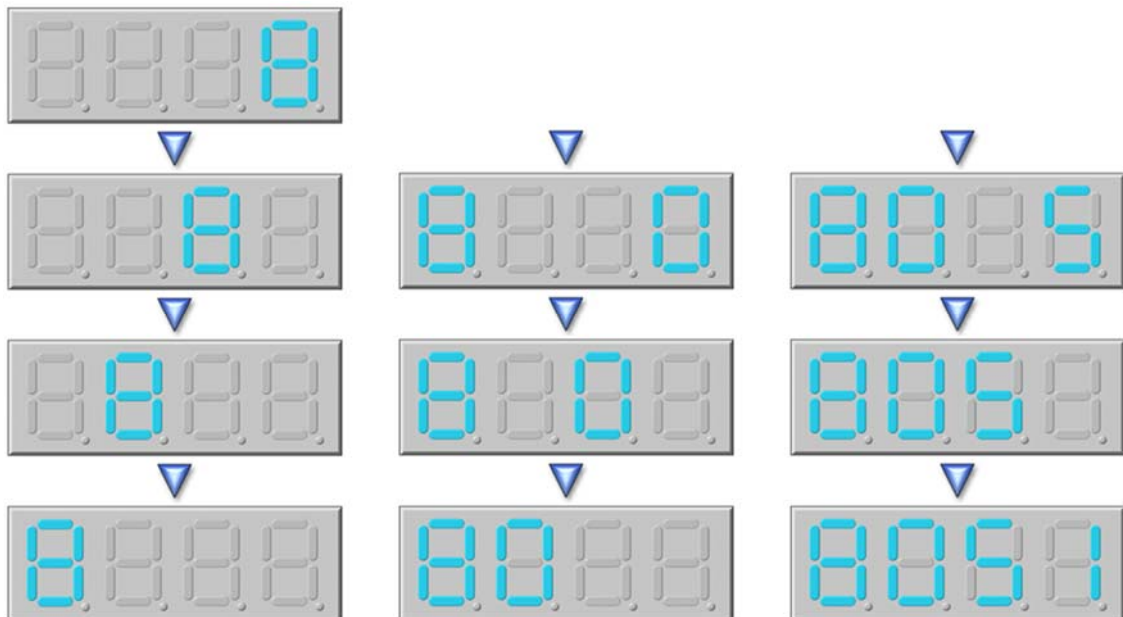
例說89S51-C語言

5-54

# 動態飛入



# 由右邊「飛入」之分解動作



# 程式-1

```

/*宣告驅動信號陣列*/
char code TAB[11]={ 0xc0, 0xf9, 0xa4, 0xb0, 0x99,
                   0x92, 0x83, 0xf8, 0x80, 0x98, 0xff };

/*宣告顯示資料陣列*/
char code disp[10][4]={ {10,10,10,8},{10,10,8,10},
                        {10,8,10,10},{8,10,10,10},
                        {8,10,10,0},{8,10,0,10},
                        {8,0,10,10},
                        {8,0,10,5},{8,0,5,10},
                        {8,0,5,1}  };

/*宣告掃描函數*/
void scanner(char);

main() // 主程式開始
{ char i; // 宣告 i 變數
  while(1) // while 迴圈開始
  { for(i=0;i<10;i++) // for 迴圈開始
    scanner(i); } // 掃描
} // 主程式結束

//=====

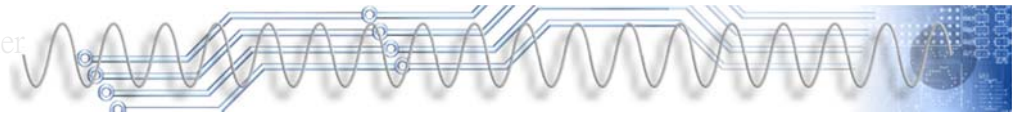
```

# 程式-2

```

void scanner(char x)
{ char i,j,k; // 宣告變數 i,j,k
  char scan; // 宣告掃描信號
  for(k=0;k<30;k++) // 設定 30 次掃描
  { scan=1; // 設定掃描信號初始值
    for(i=0;i<4;i++) // for 敘述開始
    { j=disp[x][3-i]; // 取出第一組顯示數字
      P2=TAB[j]; // 轉換成驅動信號，並輸出到 P2
      P1=~scan; // 輸出掃描信號
      delay1ms(4); // 延遲 4ms
      scan<<=1; // 下一個掃描信號
    } // 結束 i 迴圈
  } // 結束 k 迴圈
} // 結束 scan 函數

```



# Q&A

Thank you!

