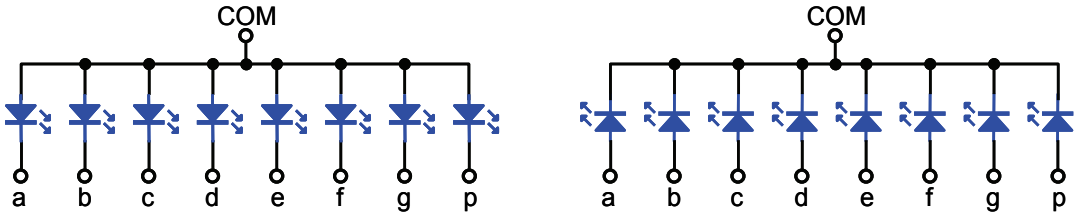
8-1 認識七段顯示器

如圖 8-1 所示七段顯示器，是以 8 個 LED 排列組合而成，由順時針方向依序命名爲 a、b、c、d、e、f、g 及小數點 p，因爲七段顯示器是由 8 個 LED 所組成，所以電氣特性與 LED 完全相同。另外在上、下各有一支 COM 腳，以方便電路板佈線。

(a) 元件　　(b) 正面接腳圖

圖 8-1　七段顯示器

如圖 8-2 所示爲七段顯示器的內部結構，可分成兩種：一爲共陽極（common anode，簡記 CA）七段顯示器，各段 LED 的陽極互相連通；另一爲共陰極（common cathode，簡記 CC）七段顯示器，各段 LED 的陰極互相連通。



(a)共陽極結構　　(b)共陰極結構

圖 8-2　七段顯示器的內部結構

□ 驅動共陽極七段顯示器

驅動共陽極七段顯示器的方法是將 COM 腳連接至+5V 電源，各段連接一 220Ω 限流電阻接地即會發亮。如表 8-1 所示爲共陽極七段顯示器數字 0~9 字型碼，如果 8051 的埠腳位元 7~0 依序連接 p、g、f、e、d、c、b、a 等腳，則 16 進顯示碼依序爲 0xc0、0xf9、0xa4、0xb0、0x99、0x92、0x82、0xf8、0x80、0x90。

碁峰 www.gotop.com.tw

表8-1　共陽極七段顯示器數字 0~9 字型碼

字型	p	g	f	e	d	c	b	a	字型	p	g	f	e	d	c	b	a
0	1	1	0	0	0	0	0	0	5	1	0	0	1	0	0	1	0
1	1	1	1	1	1	0	0	1	6	1	0	0	0	0	0	1	0
2	1	0	1	0	0	1	0	0	7	1	1	1	1	1	0	0	0
3	1	0	1	1	0	0	0	0	8	1	0	0	0	0	0	0	0
4	1	0	0	1	1	0	0	1	9	1	0	0	1	0	0	0	0

驅動共陰極七段顯示器

驅動共陰極七段顯示器的方法是將 COM 腳接地，各段連接一 220Ω限流電阻接 +5V 電源即會發亮。表 8-2 所示爲共陰極七段顯示器數字 0~9 字型碼，如果 8051 的埠腳位元 7~0 依序連接 p、g、f、e、d、c、b、a 等腳，則 16 進顯示碼依序爲 0x3f、0x06、0x5b、0x4f、0x66、0x6d、0x7d、0x07、0x7f、0x6f。

表8-2　共陰極七段顯示器 0~9 字型碼

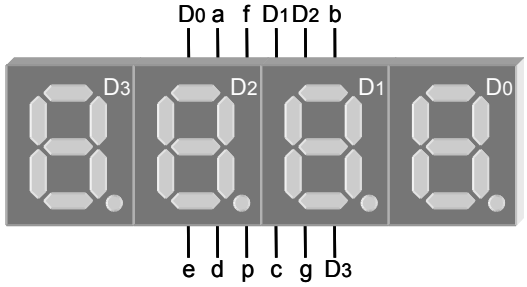
字型	p	g	f	e	d	c	b	a	字型	p	g	f	e	d	c	b	a
0	0	0	1	1	1	1	1	1	5	0	1	1	0	1	1	0	1
1	0	0	0	0	0	1	1	0	6	0	1	1	1	1	1	0	1
2	0	1	0	1	1	0	1	1	7	0	0	0	0	0	1	1	1
3	0	1	0	0	1	1	1	1	8	0	1	1	1	1	1	1	1
4	0	1	1	0	0	1	1	0	9	0	1	1	0	1	1	1	1

碁峰 www.gotop.com.tw

有時候爲了減少電路板佈線的複雜度，常如圖 8-3 所示將四個七段顯示器包裝在一起，並且將各段相同名稱的接腳連接在一起，而每一個七段顯示器都有一個驅動腳，由左而右依序爲 D_3、D_2、D_1、D_0。

如果是共陽極四連七段顯示器，各段連接一 220Ω限流電阻接地，而 D_3、D_2、D_1、D_0 任一腳接+5V，相對位數即會發亮。如果是共陰極四連七段顯示器，各段連接一 220Ω限流電阻接+5V，而 D_3、D_2、D_1、D_0 任一腳接地，相對位數即會發亮。

(a) 元件



(b) 正面接腳圖

圖 8-3　四連七段顯示器

8-2　函式說明

8-2-1　display()函式

display( )函式的功能是顯示四位數十進計數值，函式有一個 unsigned int 資料型態的引數必須設定，其值爲 0~9999，沒有傳回值。

函式原型：`void display(void)`

```
範　例：unsigned int num=1234;          //設定數值 num 爲 1234。
        display(num);                   //顯示數值。
```


碁峰 www.gotop.com.tw

8-3 實作練習

8-3-1 一位七段顯示器計數 0~9 實習

功能說明：

利用 8051 控制一位共陽七段顯示器上數計數並顯示數字 0~9。因爲是使用共陽極七段顯示器，所以 COM 腳必須連接至+5V 電源，再依表 8-1 所示將數字 0~9 字型碼由 P1 埠輸出至顯示器。

電路圖：

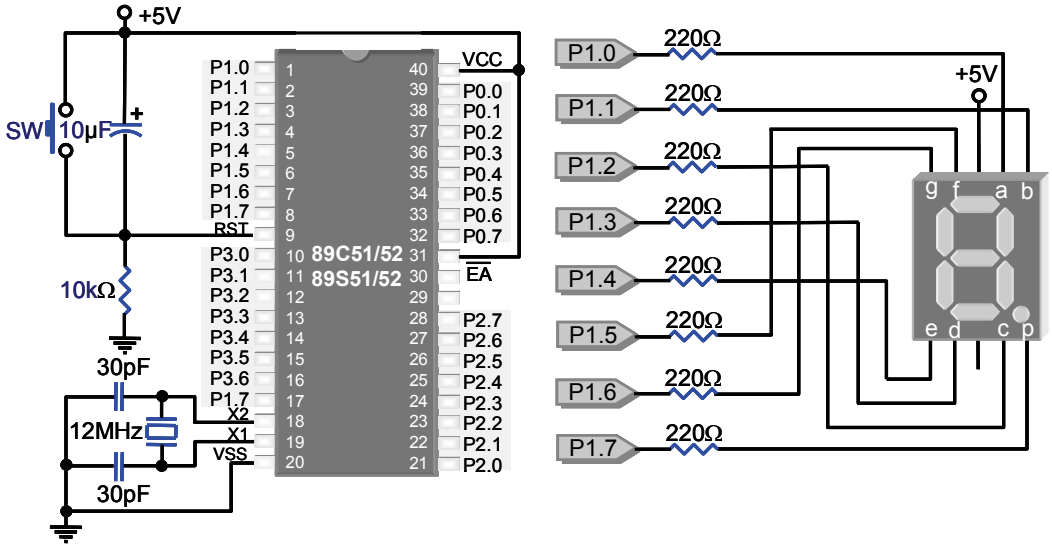


圖 8-4 一位七段顯示器計數 0~9 實習電路圖

程式：ch8-1.c

```
#include "reg51.h"                    //8051 接腳定義。
main()
{
   void delay(unsigned int);          //delay 函式原型宣告。
   char num;                          //顯示數字。
   unsigned char seg[10]=
   {0xc0,0xf9,0xa4,0xb0,0x99,         //0~4 顯示碼。
    0x92,0x82,0xf8,0x80,0x90};        //5~9 顯示碼。
   while(1)
```


碁峰 www.gotop.com.tw

```
    {
        for(num=0;num<10;num++)          //顯示數字 0~9。
        {
            P1=seg[num];                 //解碼輸出。
            delay(50000);                //延遲 0.5 秒。
        }
    }
}
void delay(unsigned int count)           //延遲函式。
{
   unsigned int i;                       //無號整數 i。
   for(i=0;i<count;i++)                  //延遲迴圈。
       ;
}
```

練習

1. 設計 8051 程式，控制一位數七段顯示器下數計數並顯示數字 9~0。
2. 設計 8051 程式，控制一位數七段顯示器下數計數並閃爍顯示 9~0。

8-3-2 兩位七段顯示器計數 00~99 實習

功能說明：

利用 8051 控制兩位共陽七段顯示器上數計數並顯示 00~99。兩位數以上七段顯示器常會將相同名稱的 LED 段連接在一起，並且使用多工掃描來驅動，不但省電而且也節省 I/O 腳的使用。

如圖 8-4 所示四位數多工掃描電路工作原理，假設四位數七段顯示器要顯示數值 1234，第一次埠腳輸出低電位驅動 Q3 電晶體導通，供電 D3 七段顯示器，再將 1 的數字碼輸出至 a~p 等腳。第二次埠腳輸出低電位驅動 Q2 電晶體導通，供電 D2 七段顯示器，再將 2 的數字碼輸出至 a~p 等腳。第三次埠腳輸出低電位驅動 Q1 電晶體導通，供電 D1 七段顯示器，再將 3 的數字碼輸出至 a~p 等腳。第四次埠腳輸出低電位驅動 Q0 電晶體導通，供電 D0 七段顯示器，再將 4 的數字碼輸出至 a~p 等腳。雖然在相同時間內，只會有一個七段顯示器通電工作，但

碁峰 www.gotop.com.tw

是因爲視覺暫留現象，只要掃描速度夠快，則人眼所看到的影像，如同是四位七段顯示器同時顯示，此即所謂的分時多工掃描。

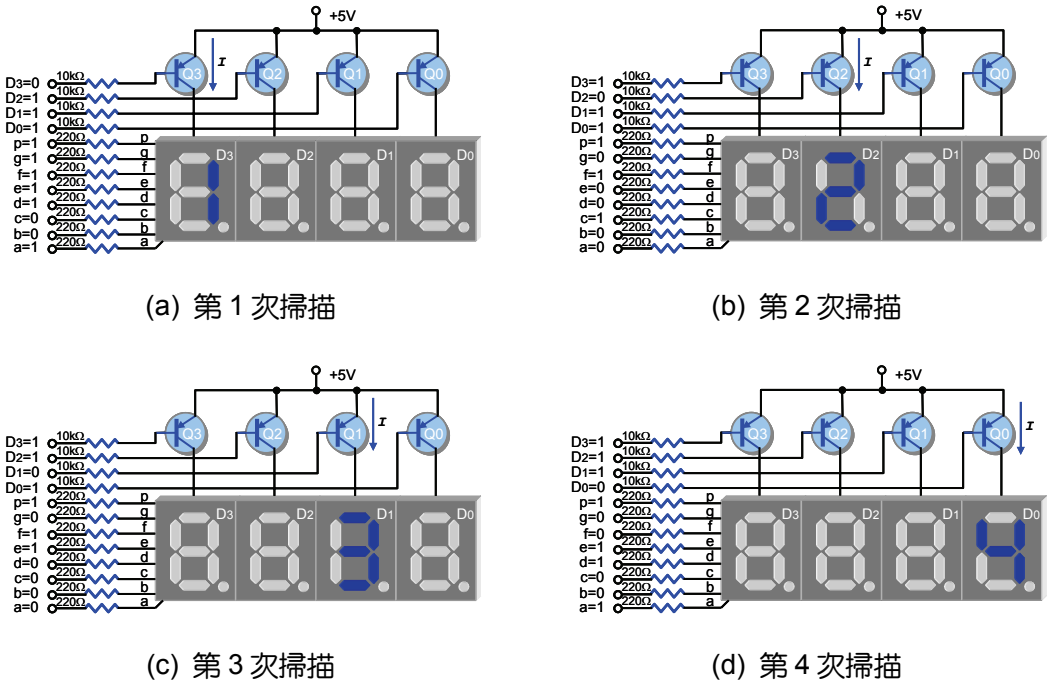


圖 8-5　多工掃描電路工作原理

人眼的視覺暫留平均時間約爲 1/16 秒，因此每位七段顯示器的顯示時間必須小於 1/ (16n) 秒，其中 n 值等於七段顯示器的總位數。如表 8-3 所示爲掃描參考時間，可視實際情形調整，如果掃描時間太短則顯示亮度不足，反之如果掃描時間太長，則會有閃爍的現象。

表8-3　掃描時間參考值

掃描總行數	工作週期	掃描總時間	每行掃描最大時間
2	$\frac{1}{2}$	$\frac{1}{64}$ 秒	$\frac{1}{64}\times\frac{1}{2}=\frac{1}{128}\cong 8\text{ms}$
4	$\frac{1}{4}$	$\frac{1}{64}$ 秒	$\frac{1}{64}\times\frac{1}{4}=\frac{1}{256}\cong 4\text{ms}$
8	$\frac{1}{8}$	$\frac{1}{64}$ 秒	$\frac{1}{64}\times\frac{1}{8}=\frac{1}{512}\cong 2\text{ms}$

碁峰 www.gotop.com.tw

電路圖：

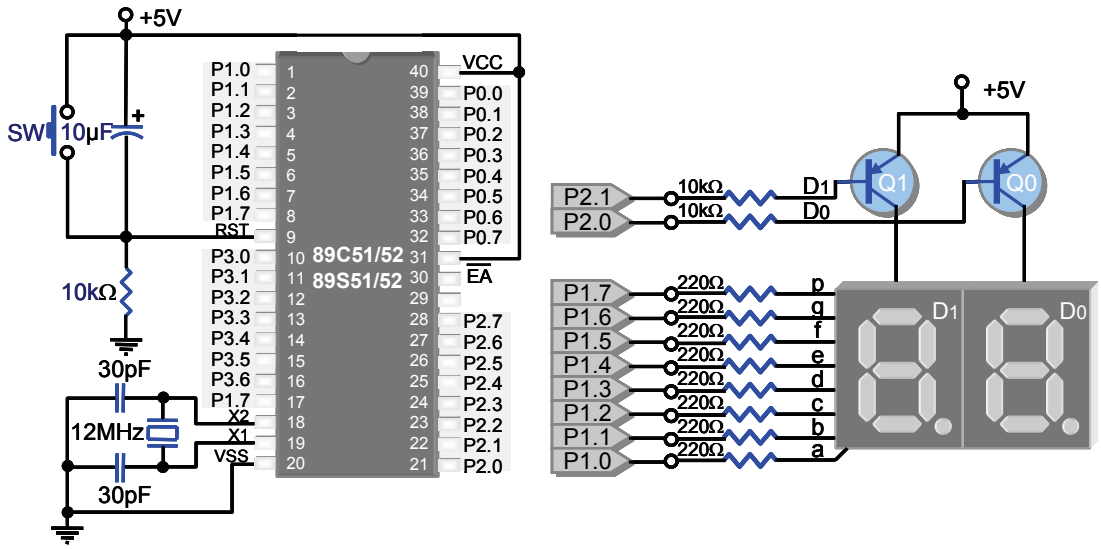


圖 8-6　兩位七段顯示器計數 00~99 實習電路圖

程式：ch8-2.c

```
#include "reg51.h"                          //8051 接腳定義。
main()
{
   void delay(unsigned int);                //延遲函式原型宣告。
   unsigned char i,j;
   char seg[10]={0xc0,0xf9,0xa4,0xb0,0x99, //七段顯示字型碼。
                 0x92,0x82,0xf8,0x80,0x90};
   while(1)
   {    for(i=0;i<99;i++)                  //計數值 00~99。
        {
            for(j=0;j<50;j++)              //計數時間控制。
            {
                P2=0x0d;                   //十位數掃描信號。
                P1=seg[i/10];              //顯示十位數字。
                delay(200);                //多工掃描時間約 2ms。
                P2=0x0e;                   //個位數掃描信號。
                P1=seg[i%10];              //顯示個位數字。
                delay(200);                //多工掃描時間約 2ms。
            }
```


碁峰 www.gotop.com.tw

```
        }
    }
}
void delay(unsigned int count)                    //延遲函式。
{
    unsigned int i;
    for(i=1;i<=count;i++)                         //延遲時間 countx10μs。
        ;
}
```

練習

1. 設計 8051 程式，控制兩位數七段顯示器下數計數並顯示數示 99~00。
2. 設計 8051 程式，控制兩位數七段顯示器下數計數並閃爍顯示 99~00。

8-3-3 四位七段顯示器計數 0000~9999 實習

功能說明

利用 8051 控制四位共陽七段顯示器上數計數並顯示數字 0000~9999。參考表 8-3 所示選用 4ms 的掃描時間。

電路圖

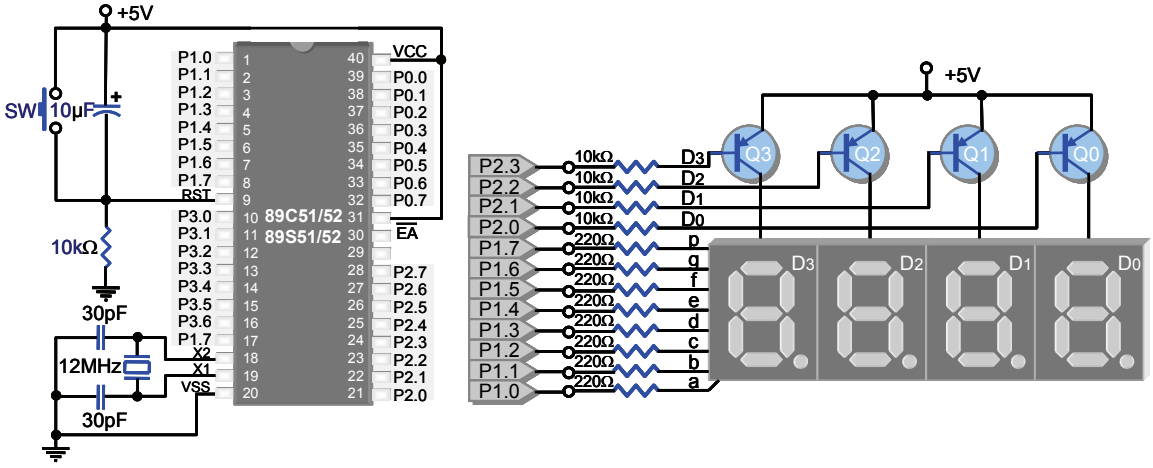


圖 8-7 四位數七段顯示器計數 0000~9999 實習電路圖

碁峰 www.gotop.com.tw

程式：ch8-3.c

```
#include "reg51.h"                              //8051 接腳定義。
main()
{
   void delay(unsigned int);                    //延遲函式原型宣告。
   int i;
   unsigned char j;
   char seg[10]={0xc0,0xf9,0xa4,0xb0,0x99,      //七段顯示字型碼。
                 0x92,0x82,0xf8,0x80,0x90};
   while(1)
   {
       for(i=0;i<9999;i++)                      //計數 0000~9999。
       {
           for(j=0;j<50;j++)                    //計數時間控制。
           {
               P2=0x07;                         //千位數掃描信號。
               P1=seg[(i/100)/10];              //顯示千位數字。
               delay(200);                      //多工掃描時間約 2ms。
               P2=0x0b;                         //百位數掃描信號。
               P1=seg[(i/100)%10];              //顯示百位數字。
               delay(200);                      //多工掃描時間約 2ms。
               P2=0x0d;                         //十位數掃描信號。
               P1=seg[(i%100)/10];              //顯示十位數字。
               delay(200);                      //多工掃描時間約 2ms。
               P2=0x0e;                         //個位數掃描信號。
               P1=seg[(i%100)%10];              //顯示個位數字。
               delay(200);                      //多工掃描時間約 2ms。
           }
       }
   }
}
void delay(unsigned int count)                  //延遲函式。
{
   unsigned int i;
   for(i=0;i<count;i++)                         //延遲時間 count×10μs。
       ;
}
```


碁峰 www.gotop.com.tw

練習

1. 設計 8051 程式，控制四位數七段顯示器下數計數並顯示 9999~0000。
2. 設計 8051 程式，控制四位數七段顯示器下數計數並閃爍顯示 9999~0000。

8-3-4 一個按鍵開關控制一位七段顯示器實習

功能說明

利用一個按鍵開關 S 控制一位共陽七段顯示器向上計數並顯示 0~9。按鍵開關可以切換顯示狀態，系統重置時的顯示值爲 0，按一下按鍵開關，顯示值由 0~9 上數計數，延遲時間由迴圈控制；再按一下按鍵開關，則停止計數。

電路圖

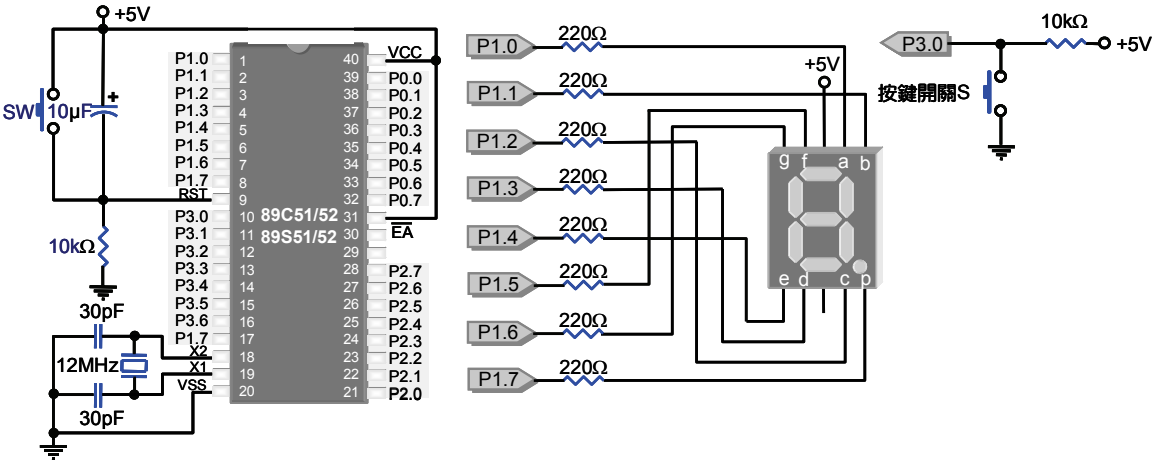


圖 8-8 一個按鍵開關控制一位七段顯示器計數實習電路圖

程式：ch8-4.c

```
#include "reg51.h"                    //8051 接腳定義。
sbit sw=0xb0;                         //按鍵開關連接至 P3.0。
char one=0;                           //彈跳期間鍵值為 1 的次數。
char zero=0;                          //彈跳期間鍵值為 0 的次數。
bit KeyData=1;                        //鍵值。
unsigned char key=0;                  //按鍵次數。
```


碁峰 www.gotop.com.tw

```
unsigned int i;                                  //延遲變數。
unsigned char num;                               //計數值。
bit timeout=0;                                   //延遲時間控制位元。
char seg[10]={  0xc0,0xf9,0xa4,0xb0,0x99,        //0~4 顯示碼。
                0x92,0x82,0xf8,0x80,0x90};       //5~9 顯示碼。
main()
{
   void oneKeyScan(void);                        //單鍵掃描函式原型宣告。
   P1=seg[0];                                    //顯示初值為 0。
   while(1)
   {
       for(i=0;i<20000;i++)                      //延遲時間控制。
       {
           oneKeyScan();                         //按鍵掃描。
           if(KeyData==0)                        //按鍵被按下?
           {
               key++;                            //按鍵次數加 1。
               if(key>2)                         //key>2?
                   key=0;                        //清除 key=0。
               KeyData=1;                        //清除鍵值。
           }
           if(timeout==1)                        //已到所設定的延遲時間?
           {
               timeout=0;                        //清除 timeout=0。
               if(key==1)                        //按第 n+1 次按鍵，n=0、2、4、…?
               {
                   num++;                        //計數值上數加 1。
                   if(num>9)                     //計數值大於 9?
                       num=0;                    //重設計數值為 0。
               }
               P1=seg[num];                      //顯示計數值。
           }
       }
       timeout=1;                                //已到所設定的延遲時間。
   }
} /* main */
void oneKeyScan(void)                            //單鍵掃描函式。
```


碁峰 www.gotop.com.tw