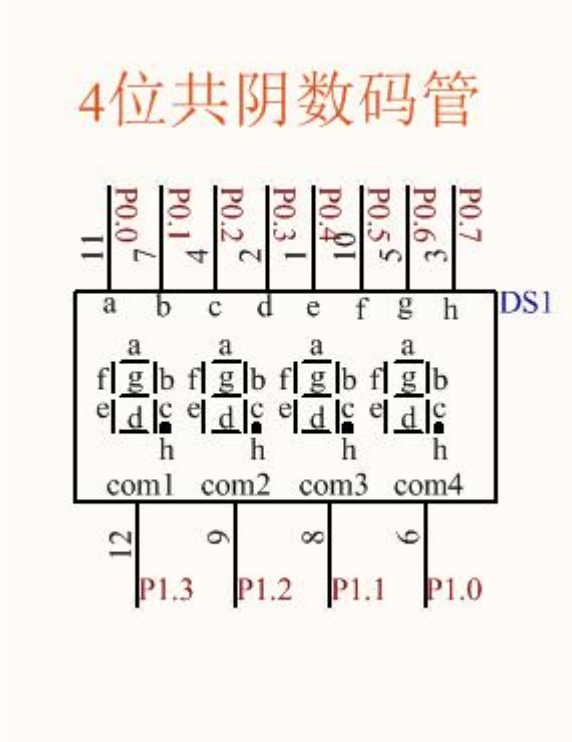
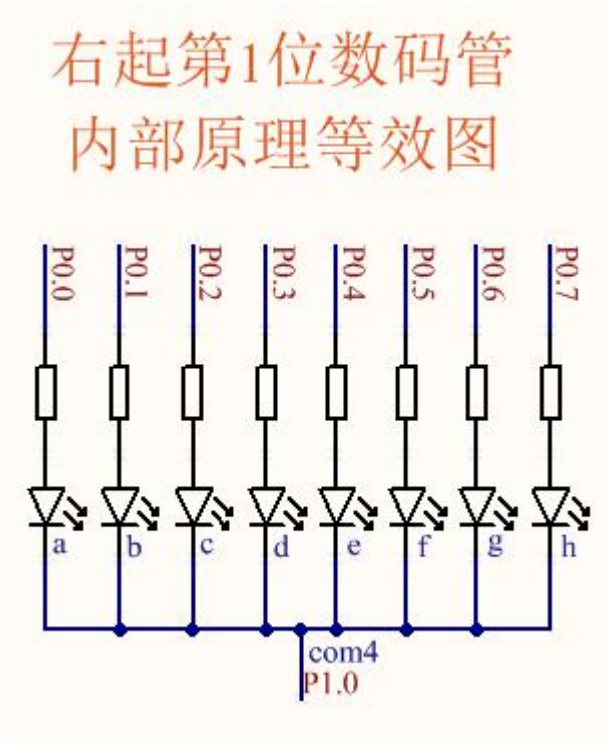
第一百一十二节： 数码管显示的基础知识。

【112.1　数码管显示的基础知识。】



上图 112.1.1　数码管



上图 112.1.2　等效图

如上图 112.1.1，一个数码管内部有 8 个段码，每个段码内部对应一颗能发光的 LED 灯，把相关位置的

段码点亮或熄灭就可以显示出不同的数字或者小数点。比如，要显示一个数字“1”，只需要点亮 b 和 c 这两个段码 LED 即可，其它 6 个 a,d,e,f,g,h 段码 LED 熄灭，就可以显示一个数字“1”。再进一步深入分析数码管内部的等效图（上图 112.1.2），com4 是右起第 1 位数码管内部 8 个段码 LED 的公共端，要点亮任何一个段码 LED 的前提必须是公共端 com4 为低电平(P1.0 输出 0 信号)。如果公共端 com4 为高电平(P1.0 输出 1 信号)，则不管段码端 P0 口的 8 个 IO 口输出什么信号，8 个段码 LED 都是熄灭的（无正压差，则无电流无回路）。因此，公共端（比如 com4,com3,com2,com1）就是某个数码管的“总开关”。比如，右起第 1 位数码管要显示数字“1”，要点亮 b 和 c，则 P0.1 和 P0.2 必须输出“1”高电平，其它 P0.0，P0.3，P0.4，P0.5，P0.6，P0.7 必须输出“0”低电平，把这 8 个 IO 口二进制的信号转换成十六进制，则整个 P0 口总线只需输出一个十六进制的 0x06，最后，“总开关”打开，公共端 com4 输出“0”，即可显示一个数字“1”。如果需要显示其它的不同数字，只需要改变段码端 P0 口的十六进制输出数值即可，如果提前把要显示的数字放在一个数组里，这个数组就是编码转换表，类似于一个字库表。现在编写一个程序例子，右起第 1 个和第 3 个数码管循环显示从 0 到 9 的数字，另外右起第 2 个和第 4 个数码管则关闭不显示，程序代码如下：

```
#include "REG52.H"

#define CHANGE_TIME  1000    //数码管切换显示数字的时间

void T0_time();
void SystemInitial(void) ;
void Delay(unsigned long u32DelayTime) ;
void PeripheralInitial(void) ;

void DisplayTask(void);    //数码管显示的任务函数

sbit P1_0=P1^0;  //右起第 1 位数码管的公共端 com4
sbit P1_1=P1^1;  //右起第 2 位数码管的公共端 com3
sbit P1_2=P1^2;  //右起第 3 位数码管的公共端 com2
sbit P1_3=P1^3;  //右起第 4 位数码管的公共端 com1

//根据原理图得出的共阴数码管编码转换表，类似于一个字库表
code unsigned char Cu8DigTable[]=
{
0x3f,  //0       序号 0
0x06,  //1       序号 1
0x5b,  //2       序号 2
0x4f,  //3       序号 3
0x66,  //4       序号 4
0x6d,  //5       序号 5
0x7d,  //6       序号 6
0x07,  //7       序号 7
0x7f,  //8       序号 8
0x6f,  //9       序号 9
0x00,  //不显示  序号 10
};
```

```
volatile unsigned char vGu8ChangeTimerFlag=0;  //控制切换数字的时间的定时器
volatile unsigned int vGu16ChangeTimerCnt=0;

unsigned char Gu8Number=0; //从 0 到 9 依次循环显示的数字

void main()
{
    SystemInitial();
    Delay(10000);
    PeripheralInitial();
    while(1)
    {
        DisplayTask();    //数码管显示的任务函数
    }
}

void DisplayTask(void)    //数码管显示的任务函数
{
    static unsigned char Su8GetCode;  //从编码转换表中提取出来的编码。

    if(0==vGu16ChangeTimerCnt)  //定时的时间到，更新显示下一个数字，依次循环显示
    {
        Su8GetCode=Cu8DigTable[Gu8Number]; //从编码转换表中提取出来的编码。

        P0=Su8GetCode; //段码端输出需要显示的编码
        P1_0=0;  //右起第 1 位数码管的公共端 com4，“总开关”打开，输出低电平 0
        P1_1=1;  //右起第 2 位数码管的公共端 com3，“总开关”关闭，输出高电平 1
        P1_2=0;  //右起第 3 位数码管的公共端 com2，“总开关”打开，输出低电平 0
        P1_3=1;  //右起第 4 位数码管的公共端 com1，“总开关”关闭，输出高电平 1

        Gu8Number++;  //显示的数字不断从 0 到 9 累加
        if(Gu8Number>9)
        {
            Gu8Number=0;
        }

        vGu8ChangeTimerFlag=0;
        vGu16ChangeTimerCnt=CHANGE_TIME;
        vGu8ChangeTimerFlag=1;  //启动新一轮的定时器
    }
}

void T0_time() interrupt 1
{
```

```
    if(1==vGu8ChangeTimerFlag&&vGu16ChangeTimerCnt>0)  //数码管显示切换时间的定时器
    {
        vGu16ChangeTimerCnt--;
    }

    TH0=0xfc;
    TL0=0x66;
}


void SystemInitial(void)
{
    //初始化上电瞬间数码管的状态
    P1_0=1;  //右起第 1 位数码管的公共端 com4，“总开关”关闭，输出低电平 1
    P1_1=1;  //右起第 2 位数码管的公共端 com3，“总开关”关闭，输出高电平 1
    P1_2=1;  //右起第 3 位数码管的公共端 com2，“总开关”关闭，输出低电平 1
    P1_3=1;  //右起第 4 位数码管的公共端 com1，“总开关”关闭，输出高电平 1

    TMOD=0x01;
    TH0=0xfc;
    TL0=0x66;
    EA=1;
    ET0=1;
    TR0=1;
}

void Delay(unsigned long u32DelayTime)
{
    for(;u32DelayTime>0;u32DelayTime--);
}

void PeripheralInitial(void)
{

}
```