

《LED 数码管显示原理》

除了键盘以外，LED 数码显示管与 LCD 液晶显示器在人机沟通方面也扮演了非常重要的角色，下面我们分别对其进行讨论。

一. LED 显示器的结构

LED(Light Emitting Diode)是发光二极管的缩写。LED 数码管里面有 8 只发光二极管，与实验板 P1 端口所接的二极管是相同的。分别记作 a、b、c、d、e、f、g、dp、其中 dp 为小数点，每一只发光二极管都有一根电极引到外部引脚上，而另外一只引脚就连接在一起同样也引到外部引脚上，记作公共端（COM），如图 5-8 所示，而图 5-9 为实物图，其中引脚的排列因不同的厂商而有所不同。

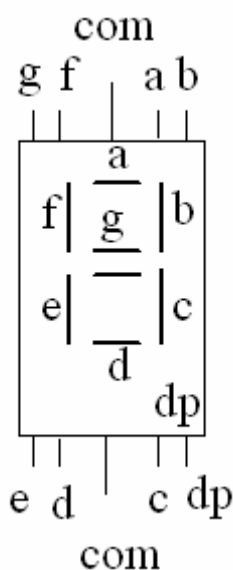


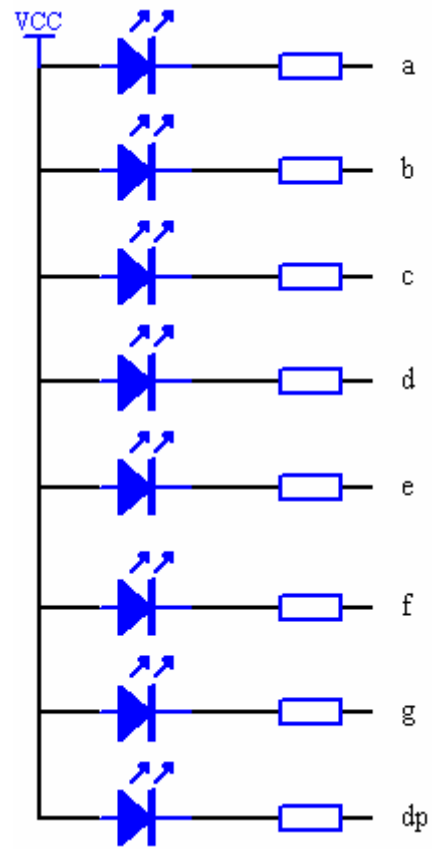
图 5-8 （数码管引脚图）



图 5-9（数码管实物图）

市面上常用的 LED 数码管有两种，分为共阳极与共阴极。共阳极：当数码管里面的发光二极管的阳极接在一起作为公共引脚，在正常使用时此引脚接电源正极。当发光二极管的阴极接低电平时，发光二极管被点亮，从而相应的数码段显示(如图 5-10 所示)。

而输入高电平的段则不能点亮。相反，共阴极：当数码管里面的发光二极管的阴极接在一起作为公共引脚，在正常使用时此引脚接电源负极。当发光二极管的阳极接高电平时，发光二极管被点亮，从而相应的数码段显示(如图 5-11 所示)，而输入低电平的段则不能点亮。



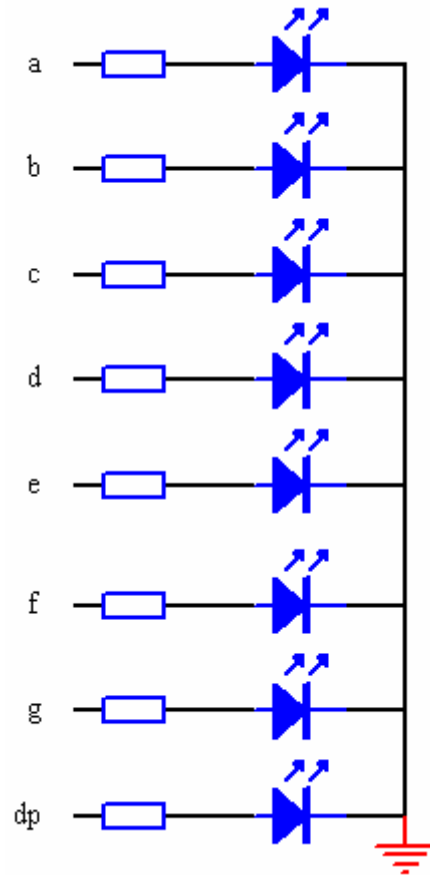


图 5-11 (共

图 5-10 (共阳极)

阴极)

二. LED 显示器的工作原理

下面图 5-12 为实验板其中一只数码管，而图 5-13 为数码管的内部接法，也就是前面所说的共阳极。当要其显示“1”时，只需置 B 与 C 为低电平，而其它的为高电平；当要显示“2”时，只需置 A、B、G、E、D 为低电平，而其它的为高电平；当要显示“8”时，就除了小数点以外全部为低电平；如此类推。

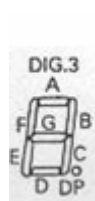


图 5-12

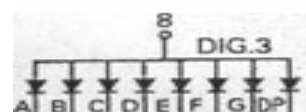


图 5-13

下面我们来系统地介绍一下在单片机应用电路中最为实用的 LED 数码管显示方法——“动态扫描”。什么是动态扫描？就是所要工作的若干个数码管轮流显示，只要轮流显示的速度足够快，每秒约 50 次以上，由于人眼的“视觉暂留”特性，看起来就像是连续显示，这样称为动态扫描。这种显示方式在数码管应用系统中应用得最为广泛。

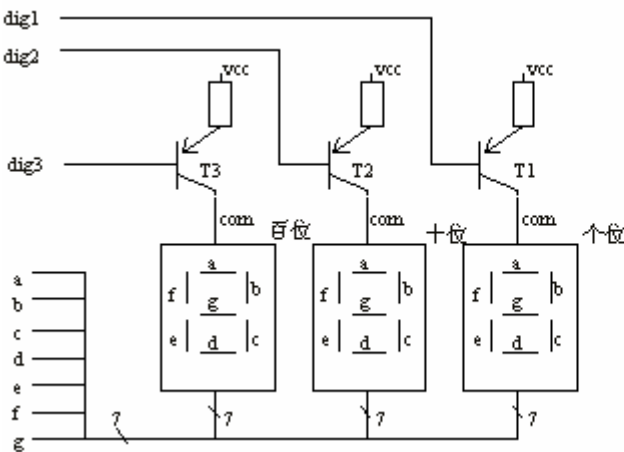


图 5-14 (动态扫描电路)

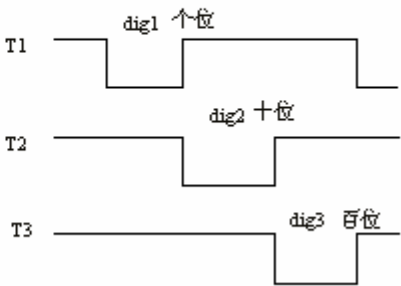


图 5-15 (动态扫描的工作时序)

图 5-14 为共阳极的动态扫描电路，而 5-15 则为电路的工作时序。下面我们来分析一下动态扫描的工作原理。在电路中 T1、T2、T3 为用作开关作用，当 B 极 dig1、dig2、dig3 为低电平时导通，为高电平时截止，三个电阻为限流电阻。下面我们从图 5-15 来理解一下动态扫描的工作原理。

第一：首先显示个位，在单片机中将 dig1 置低电平，而 dig2 与 dig3 置高电平，所以只有 T1 导通，而 T2 与 T3 则截止，同时在段码 a~g 中输出相应段码的低电平，那么在数码管中只有个位显示，而相应的十位与百位则没有显示。

第二：显示十位，在单片机中将 dig2 置低电平，而 dig1 与 dig3 置高电平，所以只有 T2 导通，而 T1 与 T3 则截止，同时在段码 a~g 中输出相应段码的低电平，那么在数码管中只有十位显示，而相应的个位与百位则没有显示。

第三：显示百位，在单片机中将 dig3 置低电平，而 dig1 与 dig2 置高电平，所以只有 T3 导通，而 T1 与 T2 则截止，同时在段码 a~g 中输出相应段码的低电平，那么在数码管中只有百位显示，而相应的个位与十位则没有显示。

这就是数码管动态扫描在单片机系统中的应用。下面我们通过实战来加深了解。

动手实验（2）

实验目的：了解数码管的工作原理。

实验内容：下面的实验程序对了解 LED 数码管的工作原理是非常显著，利用实验板的三个数码管显示“1”“2”“3”图 5-16 为实验板的基本原理图。我们这次的任务是要点亮三只数码管，如果我们要 1 秒钟点亮 3 只数码管 50 次，那么一只数码管大概要点亮 6ms。

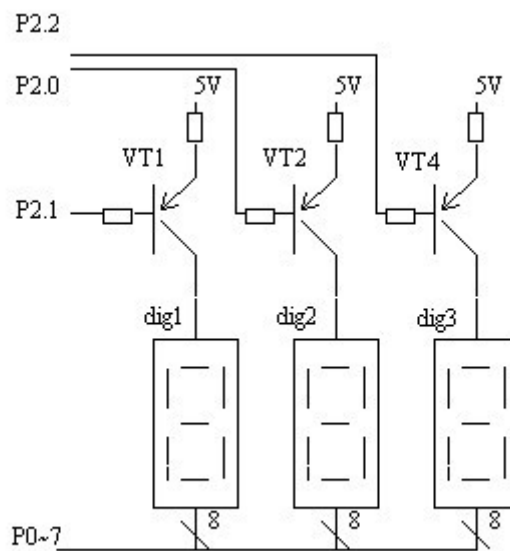


图 5-16

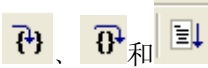
```
#include<reg52.h>//文件包含
#define uchar unsigned char
sbit led1=P2^1;//定义第一只数码管的控制线
sbit led2=P2^0;//定义第二只数码管的控制线
sbit led3=P2^2;//定义第三只数码管的控制线
#define dig1 led1=0;led2=1;led3=1;//只点亮第一只数码管
#define dig2 led1=1;led2=0;led3=1;//只点亮第二只数码管
#define dig3 led1=1;led2=1;led3=0;//只点亮第三只数码管
#define show P0//定义 P0 口为所显示段码的控制
void delay_ms(unsigned int time)//延时 1 毫秒程序，n 是形式参数
{
    unsigned int i,j;
```

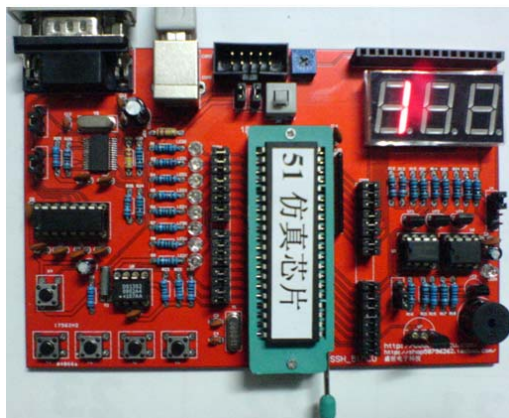
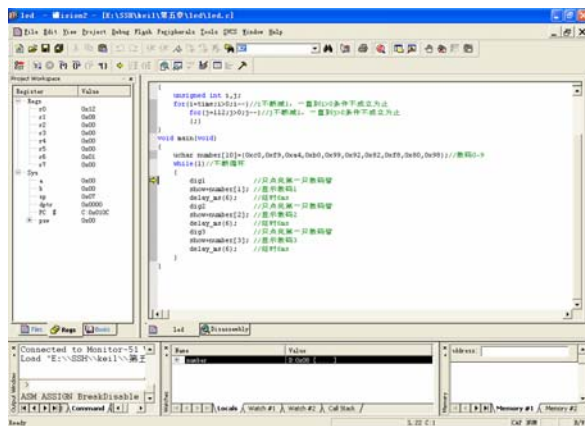
```

    for(i=time;i>0;i--)//i 不断减 1，一直到 i>0 条件不成立为止
        for(j=112;j>0;j--)//j 不断减 1，一直到 j>0 条件不成立为止
            {;}
}void main(void)
{
    uchar number[10]={0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x98};//数码
0~9
    while(1)//不断循环
    {
        dig1          //只点亮第一只数码管
        show=number[1]; //显示 1
        delay_ms(6);    //延时 6ms
        dig2          //只点亮第一只数码管
        show=number[2]; //显示 2
        delay_ms(6);    //延时 6ms
        dig3          //只点亮第一只数码管
        show=number[3]; //显示 3
        delay_ms(6);    //延时 6ms
    }
}

```

实验步骤:

1. 打开光盘第 5 章/ led / led.uv2 工程文件，对程序进行编译、链接、调试产生 led.uv2 烧写文件。
2. 将实验板的 J4 短接到 LEDP 的一边，J7 短接到 LEDE 的一边。
3. 把 SSH_51 仿真器正确装上到 SSH_51MCU 实验板上, 将工程设置为硬件仿真，同时将波特率设置为 38400。
4.  对程序进行调试，同时观察数码管显示的变化。
5. 下面（实图 3）为仿真时的 KEIL 界面，（实图 4）为实验板的同步况情。



(实图 3)

(实图 4)

实验总结：从上面的实验当中我们可以清楚地了解到数码管的工作原理，但是实际应用中是不会在主函数中用死循环来点亮数码的。大家试想一下，假如在主程序中不断用死循环来点亮数码管，若果当单片机还要处理按键扫描，数据的发送与接收等等那怎么办啊！下面我们来介绍一种非常实用的数码管扫描技术。

动手实验 (3)

实验目的：学习利用定时器来对 LED 数码管进行动态扫描。


实验内容：利用定时器的定时中断来对 LED 数码管进行扫描，实现上面同样的功能，使数码管显示“1”“2”“3”。

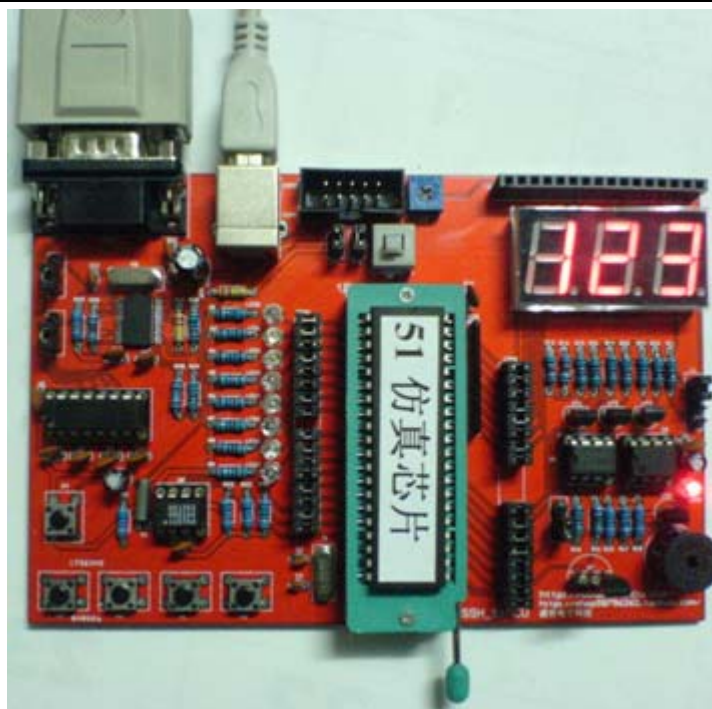
```
#include<reg52.h>//文件包含
#define uchar unsigned char
sbit led1=P2^1;//定义第一只数码管的控制线
sbit led2=P2^0;//定义第二只数码管的控制线
sbit led3=P2^2;//定义第三只数码管的控制线
#define dig1 led1=0;led2=1;led3=1;//只点亮第一只数码管
```

```
#define dig2 led1=1;led2=0;led3=1;//只点亮第二只数码管
#define dig3 led1=1;led2=1;led3=0;//只点亮第三只数码管
#define show P0//定义 P0 口为所显示段码
uchar number[10]={0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x98}; //数码 0~9
uchar one;    //个位
uchar ten;    //十位
uchar hundred;//百位
void T0_init(void)
{
    TMOD=0x01;//选择工作方式 1
    TL0=0x66;//每 6ms 中断一次
    TH0=0xea;
    TF0=0;//中断标志位清零
    ET0=1;//允许定时器 0 中断
    EA=1;//允许总中断
    TR0=1;//启动定时器
}
void main(void)
{
    T0_init();    //定时器 0 初始化
    one=number[3]; //个位显示 3
    ten=number[2];  //十位显示 2
    hundred=number[1]; //百位显示 1
    while(1);
}
void time_display(void) interrupt 1
{
    static uchar show_bit=1;
    TL0=0x66;//定时器赋初始
    TH0=0xea;
    switch(show_bit)
    {
        case 1:
        {
            dig1    //只点亮第一只数码管
            show=hundred; //百位
            show_bit=2;//下次进中断程序点亮十位
            break;
        }
        case 2:
```

```
    {  
        dig2          //只点亮第二只数码管  
        show=ten; //十位  
        show_bit=3;//下次进中断程序点亮百位  
        break;  
    }  
case 3:  
    {  
        dig3          //只点亮第三只数码管  
        show=one; //个位  
        show_bit=1;//下次进中断程序点亮个位  
        break;  
    }  
}  
}
```

实验步骤:

1. 打开光盘第 5 章/ T_led / T_led.uv2 工程文件，对程序进行编译、链接、调试产生 T_led.uv2 烧写文件。
2. 将实验板的 J4 短接到 LEDP 的一边，J7 短接到 LEDE 的一边。
3. 把 SSH_51 仿真器正确装上到 SSH_51MCU 实验板上，将工程设置为硬件仿真，同时将波特率设置为 38400，以便对程序进行调试。
4.  对程序进行全速运行，同时观察数码管显示状态的变化。实验效果如下（实图 5）



(实图 5)

实验结果：本实验中利用定时器的定时中断对 LED 管进行点亮。这样单片机可以在主函数中处理其它的事情，如按键扫描，数据的发送与接收等。当定时计数溢出时单片机才去响应中断程序，点亮一次数码管再回到主程序，这样不断周而复始地循环。

动手实验（4）

实验目的：学习 LED 数码管的实用技术。

实验内容：用 LED 数码管记录下 K0 所按下的次数，即按一下 K0 增值一次曾在数码管中显示出来。

```
#include<reg52.h> //文件包含
#define uchar unsigned char
sbit led1=P2^1; //定义第一只数码管的控制线
sbit led2=P2^0; //定义第二只数码管的控制线
sbit led3=P2^2; //定义第三只数码管的控制线
sbit K0=P3^2;
#define dig1 led1=0;led2=1;led3=1; //只点亮第一只数码管
#define dig2 led1=1;led2=0;led3=1; //只点亮第二只数码管
#define dig3 led1=1;led2=1;led3=0; //只点亮第三只数码管
#define show P0 //定义 P0 口为所显示段码
uchar number[10]={0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x98}; //数码 0~9
uchar one; //个位
```

```
uchar ten;    //十位
uchar hundred;//百位
void delay_ms(unsigned int time)//延时 1 毫秒程序，n 是形式参数
{
    unsigned int i,j;
    for(i=time;i>0;i--)//i 不断减 1，一直到 i>0 条件不成立为止
        for(j=112;j>0;j--)//j 不断减 1，一直到 j>0 条件不成立为止
            {;}
}
void T0_init(void)
{
    TMOD=0x01;//选择工作方式 1
    TL0=0x66;//每 6ms 中断一次
    TH0=0xea;
    TF0=0;//中断标志位清零
    ET0=1;//允许定时器 0 中断
    EA=1;//允许总中断
    TR0=1;//启动定时器
}
void main(void)
{
    uchar i,j,k;
    T0_init();//定时器 0 初始化
    i=0;//控制个位数码
    j=0;//控制十位数码
    k=0;//控制百位数码
    one=number[0];//上电时三位数码管都显示为 000
    ten=number[0];
    hundred=number[0];
    while(1)
    {
        if (K0==0)//判断是否有按键被按下
        {
            delay_ms(20);//延时消抖
            if (K0==0)//确认有按键被按下
            {
                while (K0==0)//等待按键松开
                {
                    i++;//个位递增
                    one=number[i];//显示个位
                    if(i==10)//假如个位为 10 立即向十位进 1
```

```

        {
            i=0;//个位再从 0 开始递增
            j++;//十位递增
            one=number[i];//显示个位
            ten=number[j]; //显示十位
            if(j==10)//假如十位为 10 立即向百位进 1
            {
                j=0;//十位再从 0 开始递增
                ten=number[j];//显示十位
                k++;//百位递增
                hundred=number[k]; //显示百位
                if(k==10)//当计数为 999 时立即转为 000 从新开始递增
                {
                    i=0;//三位数码管都显示为 000
                    j=0;
                    k=0;
                    one=number[0];
                    ten=number[0];
                    hundred=number[0];
                }
            }
        }
    }
}

void time_display(void) interrupt 1
{
    static uchar show_bit=1;
    TL0=0x66;//定时器赋初始
    TH0=0xea;
    switch(show_bit)
    {
        case 1:
        {
            dig1 //只点亮第一只数码管
            show=hundred; //百位
            show_bit=2;//下次进中断程序点亮十位
            break;
        }
    }
}

```

```

    case 2:
    {
        dig2          //只点亮第二只数码管
        show=ten; //十位
        show_bit=3; //下次进中断程序点亮百位
        break;
    }
    case 3:
    {
        dig3          //只点亮第三只数码管
        show=one; //个位
        show_bit=1; //下次进中断程序点亮个位
        break;
    }
}
}

```

实验步骤

1. 打开光盘第 5 章/ key_led / key_led.uv2 工程文件，对程序进行编译、链接、调试产生 key_led.hex 烧写文件。
2. 将实验板的 J4 短接到 LEDP 的一边，J7 短接到 LEDE 的一边。
3. 把串口线与 USB 线将实验板正确连接。把 key_led.hex 烧写文件下载到 STC89C52 单片机中去。
4. 此时会见到数码管显示 000，当每按下一次 K0 数码管的值就会增 1。实验效果如下（实图 6）



（实图 6）

自我练习：自行编写一个程序，利用实验板的三个数码管显示从 000~999 每 300 毫秒增加一次，不断周而复始地循环。工程名命名为“myled”。

程序的设计思路：要实现本程序，我们可以在动手实验（4）的主程序中稍作修改

就可以了，关于程序的其它可以完全不变。

- （1）在主函数中声明三个变量 i、j、k、并使其初始化为 0。
- （2）每 300ms 使 i 的值自加 1，当 i 的值为 10 时使 j 自加 1，当 j 为 10 时再使 k 自加 1，当计数到 999 时再把 i、j、k、三个变量赋值为 0 从新开始计数。
- （3）计数的同时将 i、j、k、三个变量赋给全局变量的个位、十位、百位。

（练习的答案附光盘中）