

第 3 章



驱动数码管(1)

数码管是单片机应用系统中常用的显示器件,能以 7 段或 8 段的形式显示数字、部分英文字母和小数点(Data Point, DP),有一位、两位一体、三位一体和四位一体等多种封装形式。

本章首先介绍数码管的内部结构与显示原理,然后重点讲述应用单片机驱动一位数码管和四位一体数码管的基本方法。

3.1 数码管的结构与显示原理

一位数码管,如图 3-1 所示,内部由 7 个或 8 个发光二极管组成,引脚排列和字段定义如图 3-2(a)所示。根据内部发光二极管的连接方式不同,数码管分为共阳极型(Common Anode, CA)和共阴极(Common Cathode, CC)两种类型。一位数码管的内部结构分别如图 3-2(b)和图 3-2(c)所示,其中 COM 为公共端。

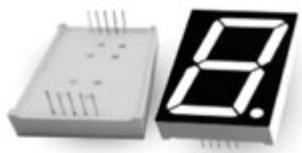


图 3-1 一位数码管



第 6 集
微课视频

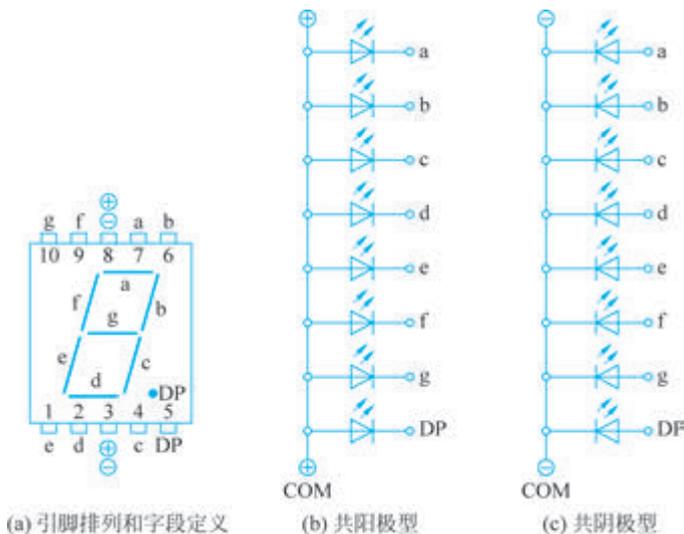


图 3-2 一位数码管的内部结构

对于共阳极型数码管,应用时应将 COM 端接到单机电源(V_{CC})上,字段驱动信号 a~g 和 DP 低电平有效;对于共阴极数码管,应用时应将 COM 端接地(GND),字段驱动信号 a~g 和 DP 高电平有效。

由于发光二极管导通时相应的字段被点亮,因此可以通过点亮不同的字段显示不同的数字或字符信息。十六进制数码的显示字段定义如图 3-3 所示。



图 3-3 十六进制数码的显示字段定义

对于 8051 单片机应用系统,若定义驱动数码管字段的字节数据格式如表 3-1 所示,那么十六进制数码和部分英文字符的驱动段码(Segment Code)如表 3-2 所示。

表 3-1 驱动数码管字段的字节数据格式定义

DP	g	f	e	d	c	b	a
----	---	---	---	---	---	---	---

表 3-2 十六进制数码和部分英文字符的驱动段码表

显示数字/ 字符	共阳极型数码管字段电平								十六进制段码	
	DP	g	f	e	d	c	b	a	共阳极型	共阴极型
0	1	1	0	0	0	0	0	0	c0	3f
1	1	1	1	1	1	0	0	1	f9	06
2	1	0	1	0	0	1	0	0	a4	5b
3	1	0	1	1	0	0	0	0	b0	4f
4	1	0	0	1	1	0	0	1	99	66
5	1	0	0	1	0	0	1	0	92	6d
6	1	0	0	0	0	0	1	0	82	7d
7	1	1	1	1	1	0	0	0	f8	07
8	1	0	0	0	0	0	0	0	80	7f
9	1	0	0	1	0	0	0	0	90	6f
A	1	0	0	0	1	0	0	0	88	77
b	1	0	0	0	0	0	1	1	83	7c
C	1	1	0	0	0	1	1	0	c6	39
d	1	0	1	0	0	0	0	0	a1	5e
E	1	0	0	0	0	1	1	0	86	79
F	1	0	0	0	1	1	1	0	8e	71
-	1	0	1	1	1	1	1	1	bf	40
灭	1	1	1	1	1	1	1	1	ff	00
P	1	0	0	0	1	1	0	0	8c	73
H	1	0	0	0	1	0	0	1	89	76

续表

显示数字/ 字符	共阳极型数码管字段电平								十六进制段码	
	DP	g	f	e	d	c	b	a	共阳极型	共阴极型
L	1	1	0	0	0	1	1	1	c7	38
U	1	1	0	0	0	0	0	1	c1	3e
n	1	1	0	0	1	0	0	0	c8	37

应用 C51 编写数码管驱动程序时,首先需要根据所使用共阳极型/共阴极型数码管定义相应的段码数组。

共阳极型数码管的十六进制段码数组定义的 C51 代码参考如下。

```
unsigned char code SEGcode[16] = //共阳极型十六进制段码:0123456789 AbCdEf
    { 0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e };
```

其中,关键词“code”表示随定义的段码数组将应用程序存储到 ROM 中。

在单片机应用系统中,经常使用多位一体数码管以节约单片机的 I/O 资源。四位一体数码管的封装形式及其内部结构如图 3-4 所示。其中每个数码管的段驱动(A~G 和 DP)是公用的,而数码管的 COM 端(DIG. 1~DIG. 4)是独立的,通过位码控制 COM 端选择当前显示的数码管。

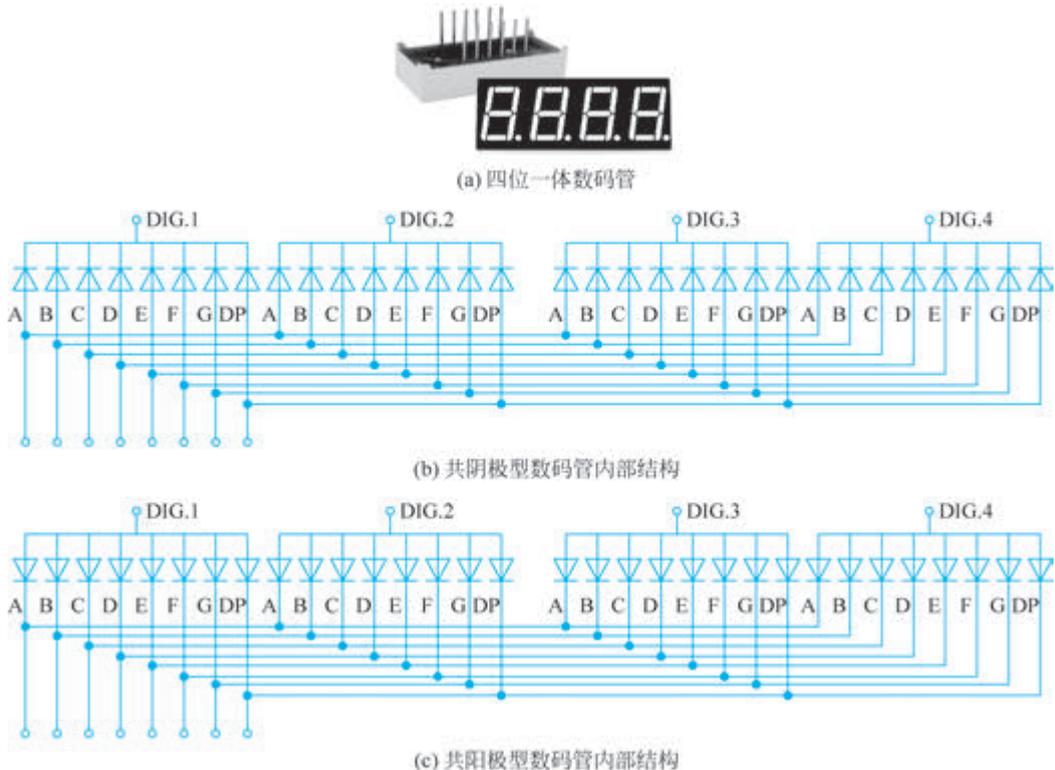


图 3-4 四位一体数码管

驱动四位一体数码管需要占用单片机 12 个 I/O 接口,其中段驱动占 8 个 I/O 接口、位驱动 4 个 I/O 接口。应用多位数码管显示不同数字或字符信息时,需要采用动态扫描的方式进行驱动。

思考与练习

3-1 根据 7 段数码管的字段结构,分析 7 段数码管除了能够显示数字 0~9 外,还能够显示哪些英文字符(设显示的字符不区别大小写)。

3.2 数码管的驱动方法

数码管有静态和动态两种驱动方法。

3.2.1 数码管的静态驱动

静态驱动指多个数码管中每个数码管的字段均由独立的 I/O 接口持续驱动,所以驱动一个数码管需要占用 8 个 I/O 接口,驱动 4 个数码管需要占用 32 个 I/O 接口。

静态驱动的优点是亮度高,但驱动电路复杂,功耗大。由于图 2-39 所示人机接口电路中的两个 4 位一体数码管字段对应相连,应用静态驱动时只能选用其中一个数码管,或者将 8 个数码管当作一个数码管使用,显示相同的数字或字符信息。对于共阳极型数码管,某个数码管的 COM 端为高电平时,将在该数码管上显示信息。

在 Keil μ Vision 开发环境中新建工程 DTdrv_demo.uvproj,选用 AT89C52 单片机,完成如下所示 8 位数码管的静态驱动程序 CP03_DTdrv_static.c 的编辑、编译和目标设置,以生成编程文件 DTdrv_demo.hex。

```
//文件名 CP03_DTdrv_static.c
#include <reg52.h>
//12T 单片机延时子程序(1ms,@12MHz)
void Delay1ms(unsigned int xms)
{
    unsigned char i;
    while ( xms -- )
        for( i = 0; i < 123; i++);
}
//共阳极型数码管段码表 0123456789AbCdEF
unsigned char code SEGcode[16] =
    { 0xc0,0xf9,0xa4,0xb0,0x99, 0x92,0x82, 0xf8,
      0x80,0x90,0x88,0x83, 0xc6,0xc0,0x86,0x8e };
//共阳极型数码管位码表
unsigned char code BITcode[8] = { 0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80 };
//数码管静态驱动子程序
void DTdisp_static()
```

```

{
    unsigned char i,j;                //定义循环变量
    //显示模式 1:驱动单个数码管从左向右依次循环显示 0~9
    for( i = 0; i < 8; i++)           //对应数码管序号 0~7
    {
        P2 = BITcode[i];             //送位码
        for( j = 0; j < 10; j++)     //循环显示 0~9
        {
            P0 = SEGcode[j];        //送段码
            Delay1ms(100);           //延时 100ms
        }
    }
    //显示模式 2:驱动 8 个数码管同步显示 0~F
    P2 = 0xff;                        //送位码
    for ( j = 0; j < 16; j++)
    {
        P0 = SEGcode[j];            //送段码
        Delay1ms(200);              //延时 200ms
    }
}
// 主程序
void main()
{
    while(1) DTdisp_static();        //循环调用静态显示子程序
}

```

将编程文件 DTdrv_demo.hex 加载到图 2-39 所示的人机接口电路中,完成数码管的静态驱动软硬件联合仿真。仿真结果如图 3-5 所示。

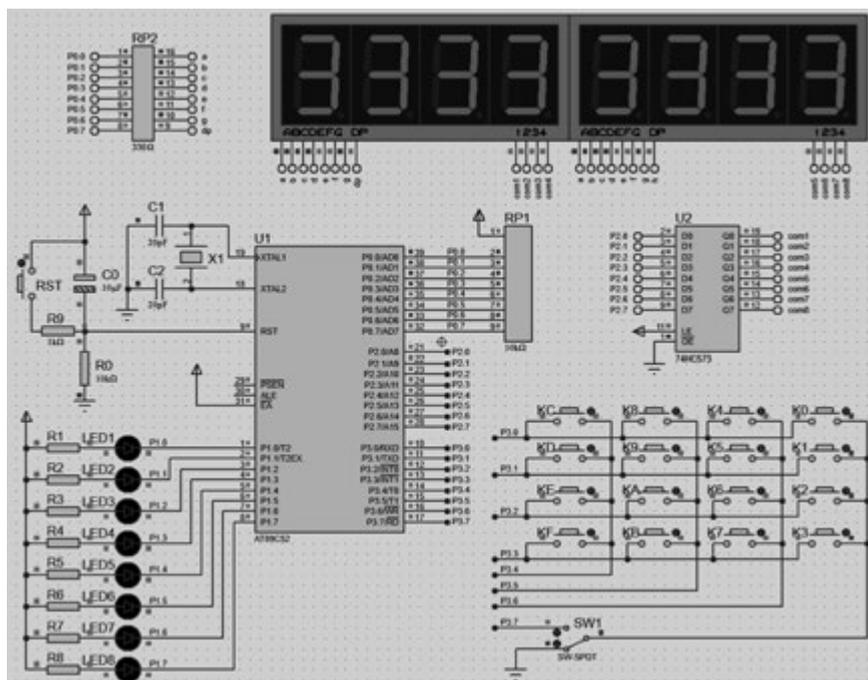


图 3-5 软硬件联合仿真结果

思考与练习

3-2 应用 Proteus 和 Keil μ Vision, 完成数码管静态驱动程序 CP03_DTdrv_static.c 的功能仿真。

3.2.2 数码管的动态驱动

动态驱动指多个数码管的字段对应相连, 由一组 I/O 接口扫描驱动。具体的方法是, 用位码轮流选通每个数码管的 COM 端, 依次循环点亮每个数码管, 当显示数码管的帧刷新率大于或等于 25 帧/s 时, 由于人眼的视觉暂留现象(Visual Staying Phenomenon), 就可以看到多位一体数码管“同时”显示信息。

动态驱动的优点是驱动电路简单, 节约单片机的 I/O 接口资源, 因为驱动四位一体数码管只需要占用 12 个 I/O 接口。由于动态驱动瞬时只点亮一个数码管, 因此功耗低, 合理控制扫描频率也能够保证数码显示的亮度。

对于 8 位数码管, 如果设计每个数码管的显示时间为 5ms, 那么数码管显示的帧刷新率为 $1000\text{ms}/(5\text{ms}\times 8)=25$ 帧/s。

编写应用程序 CP03_DTdrv_dynamic.c, 驱动图 2-39 所示人机接口电路中的两个四位一体数码管稳定显示“18931226”的程序代码参考如下。

```
//文件名 CP03_DTdrv_dynamic.c
#include <reg52.h>
//共阳极型数码管段码表:0~9
unsigned char code SEGcode[10] = { 0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90 };
//12T 单片机延时子程序(约 1ms, @12MHz)
void Delay1ms(unsigned int xms)
{
    unsigned char i;
    while( xms -- )
        for (i = 0; i < 123; i++);
}
//数码显示数据缓存数组及位码定义。预设显示"18931226"
unsigned char DTbuf[8] = { 1, 8, 9, 3, 1, 2, 2, 6 };
void DTdisp_dynamic(void)
{
    unsigned char i; //定义循环变量
    for(i = 0; i < 8; i++) //对应 4 个数码管
    {
        P0 = SEGcode[DTbuf[i]]; //送显示数据段码
        P2 = 0x01 << i; //送位码
        Delay1ms(2); //延时 2ms
        P2 = 0x00; //留给读者分析
    }
}
//主函数
void main()
{
    while(1) DTdisp_dynamic(); //循环调用动态扫描子程序
}
```

将工程 DTdrv_demo 中的静态驱动应用程序 CP03_DTdrv_static.c 替换为上述动态驱动应用程序,重新编译以生成编程文件 DT4drv.hex。将编程文件加载到人机接口电路中,完成四位一体数码管驱动电路的软硬件联合仿真。仿真结果如图 3-6 所示。

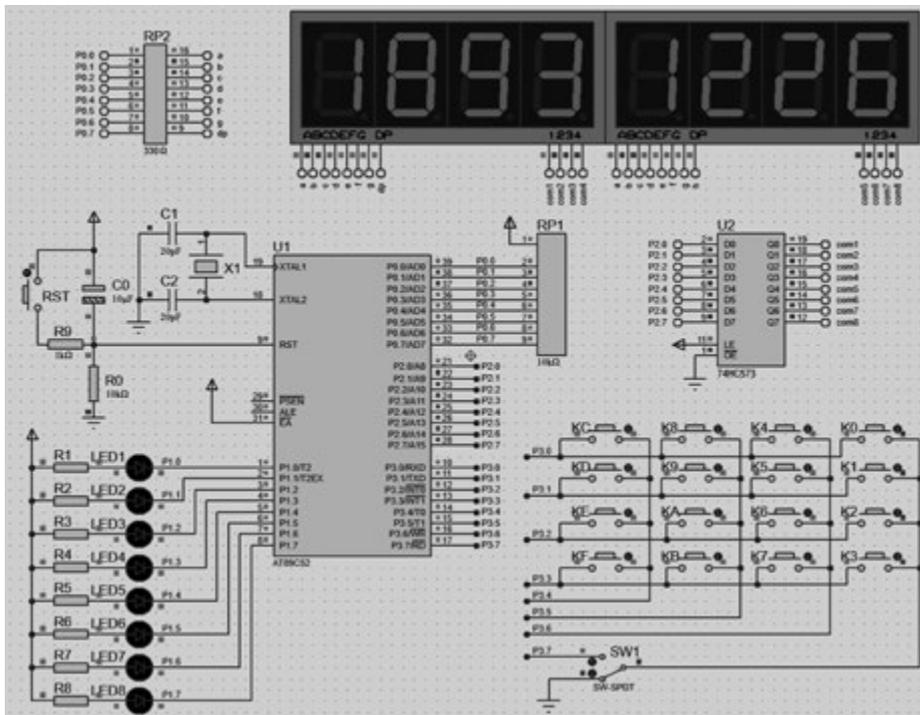


图 3-6 软硬件联合仿真结果

思考与练习

3-3 应用 Proteus 和 Keil μ Vision, 完成 8 位数码管动态驱动程序 CP03_DTdrv_dynamic.c 的功能仿真。

3-4 分析数码管动态驱动程序 CP03_DTdrv_dynamic.c 中语句“P2=0x00”的作用。如果将语句“P2=0x00”注销掉,会出现什么问题? 应用 Proteus 和 Keil μ Vision 进行仿真验证。

拓展训练题

对于图 2-39 所示的人机接口电路,编写应用程序,驱动 8 个数码管稳定显示数字“19491001”的同时,依次循环点亮发光二极管 LED~LED8。设每个 LED 的显示时间约为 500ms。应用 Proteus 和 Keil μ Vision 完成系统软硬件的功能仿真。