

Led 显示模块

4. 4. 1 设计目的及任务

- 1、设计任务：设计一个 8 段 6 位 LED 数码管显示器以及相应的驱动电路。
- 2、功能指标：可以显示 0~9 数字以及 A~F 英文字符，并带小数点。电源为 +5V 供电
- 3、设计要求：所设计的显示器应满足 EDP 实验仪系统设计要求，并能与整个系统有效结合。

以下是一个 LED 显示器的原理、设计范例及其相应电路的讲解，仅供参考。

4. 4. 2 LED 显示器的基本工作原理

1、LED 的结构

LED 显示器采用发光二极管显示字段。单片机系统中经常采用的是八段显示器，即 LED 显示器中有 8 个发光二极管，每段 LED 的笔画分别称为 a、b、c、d、e、f、g，代表“a. b. c. d. e. f. g.”七个字段和一个小数点“dp”。它有共阴和共阳两种结构。七段 LED 的阳极连在一起称为共阳极接法，而阴极接在一起的称为共阴极接法。如图 4-4-1 所示。

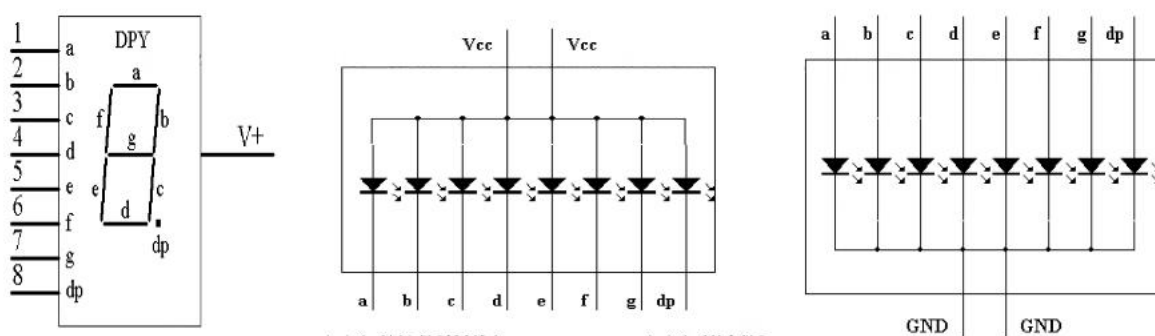


图 4-4-1: LED 数码管原理图

2、LED 的工作原理

共阴极的 LED，只要在某该段二极管加上高电平，该段即点亮，反之则暗。共阳极的与之相反。对共阴极 LED 显示器的控制采用“接地方式”即通过控制 LED 的“GND”引脚

的电平高低来达到选通的目的，该引脚即通常所说的位选线。共阳极 LED 显示器控制方式则相反。两种控制方式中，共阴极 LED 控制方式受系统器件功耗限制，只能用于小尺寸的 LED 显示器中。对于大尺寸 LED 显示器的控制（如大屏幕计时器）一般使用共阳极方式。使用 LED 显示器时，工作电流一般为 2-10mA/段，这样当 LED 处于全亮状态时，工作电流约 15-80 mA 左右。LED 显示器的亮度除与工作电流有关外，还与 LED 的型号有关。根据显示亮度的不同划分为普通亮度和高亮度 LED，高亮度 LED 显示器的发光强度远大于普通亮度的 LED，正常情况下的发光强度越是普通 LED 的 10 倍，即在 1-2 mA/段时便可点亮。一个单片机应用系统中，通常将控制 LED 显示字符的 8 位数据称之为段选码，七段 LED 的段选码如表 4-1 所示，共阴极与共阳极段选码互为补码，即两数值相加等于 FFH。

表 4-1: 七段 LED 的段代码

字符	共阴	共阳	字符	共阴	共阳
0	3FH	COH	8	7FH	80H
1	06H	F9H	9	6FH	90H
2	5BH	A4H	A	77H	88H
3	4FH	BOH	B	7CH	83H
4	06H	99H	C	39H	C6H
5	6DH	92H	D	5EH	A1H
6	7DH	82H	E	79H	86H
7	07H	F8H	F	71H	8EH

段选码的数值大小根据 LED 结构很容易得出，假设一个共阴 LED 要显示数字“0”，根据图 4-1 则有： $a=b=c=d=e=f=1, g=0$ $dp=0$ 即二进制的“00111111B”，也就是 16 进制的“3FH”。

4. 4. 3 设计内容

1. 试验仪 LED 显示器接口定义

试验仪 LED 显示器接口定义如图 4-4-2。

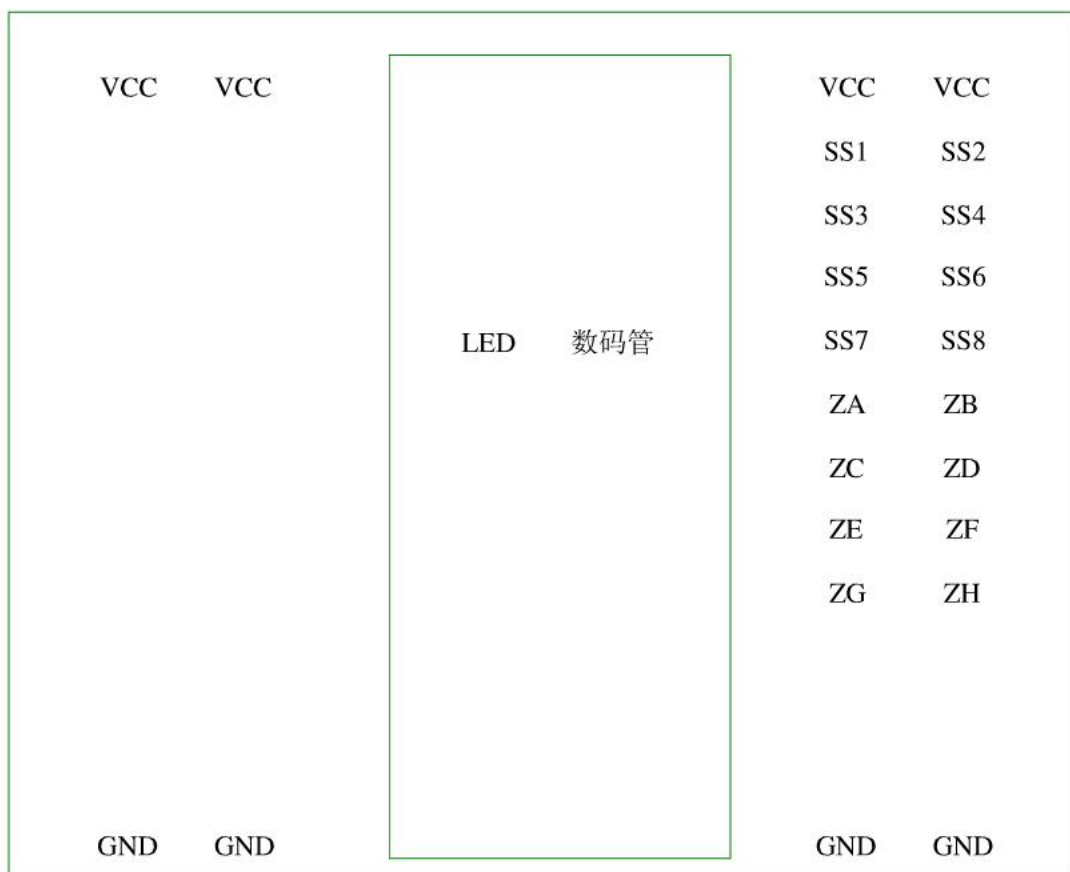


图 4-4-2 LED 数码管模块的双 26 针插脚位置及定义

2、原理图及其说明

试验仪 LED 显示器的原理图可参照图 4-4-3 和 4-4-4，由于单片机的 I/O 口有限，因此通过一片 I/O 口扩展芯片 8155 的 PA 和 PB 口作为 LED 的“段选线”和“位选线”。试验仪上的 LED 采用共阴极模块。为了驱动各段 LED，需要外加驱动电路。本模块采用两片正向驱动器 7407 与 LED 段选线连接，通过一片反向驱动器 7406 与 LED 的位选线连接。

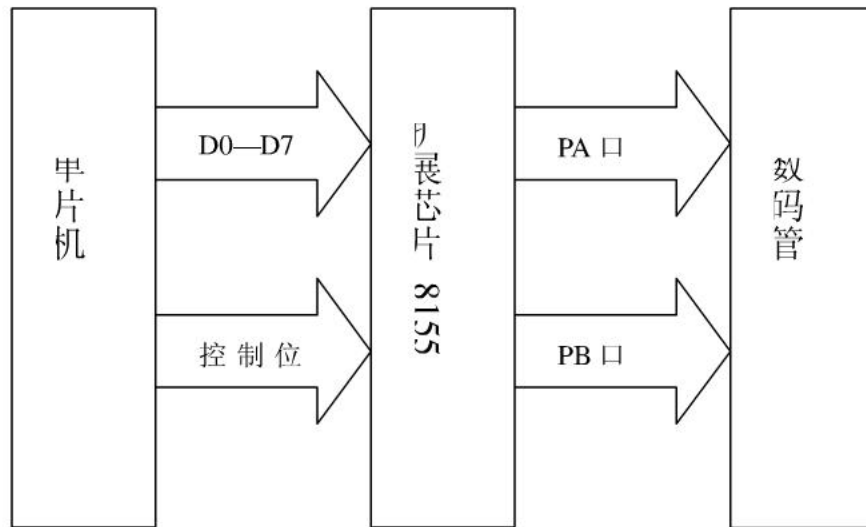


图 4-4-3 LED 数码管原理框图

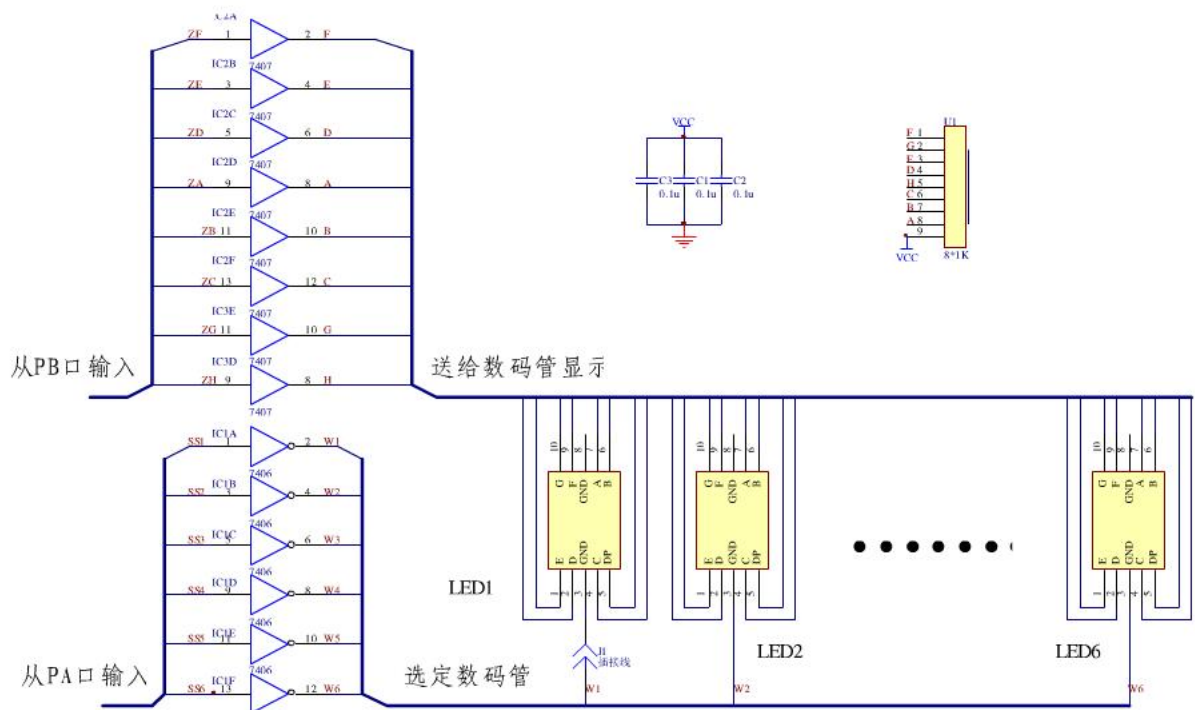


图 4-4-4 LED 数码管模块电原理图

3.用动态显示方式实现在六个 LED 显示器显示不同字符。

EDP 实验型通过 8155 I/O 扩展芯片控制 LED 显示。8155 的 PB 口和 PA 口分别作为字

形口和字位口，它们的地址分别为 0fd02H 和 0fd01H，8155 控制口为 0fd00H。设 8155 的工作方式命令字为 03H，即 8155 的 PA、PB 口作为输出口且工作于方式 0，PC 口不用。

在程序编写中，为了显示 6 个不同的字符，必须在存储器中开辟一个显示缓冲区，缓冲区里放置欲显示字符的字形码。程序从显示缓冲区中逐个取出要显示的字形码，依次把字符显示在规定的字位上。在每点亮一个 LED 显示器后，延时一段时间，使之发光稳定，然后顺序点亮其它的显示器，这样循环扫描显示。由于延时时间短，因此我们看到的效果是所有字符同时被显示出来，这种显示方式就称为动态显示。通过对延时时间的控制，可以使显示达到不同的效果。

4、软件设计方法

软件设计中，首先定义 8155 的命令寄存器的外部地址是 0x0fd00，定义 PA 和 PB 口的外部地址为 0x0fd01 和 0x0fd02，然后定义一个数据缓存数组 Led_number[6]，存储显示数据。其流程见图 4-4-5，根据软件流程得到的 LED 动态显示与 51 单片机的 C 语言接口函数 LED.c 见例 4-4-1。

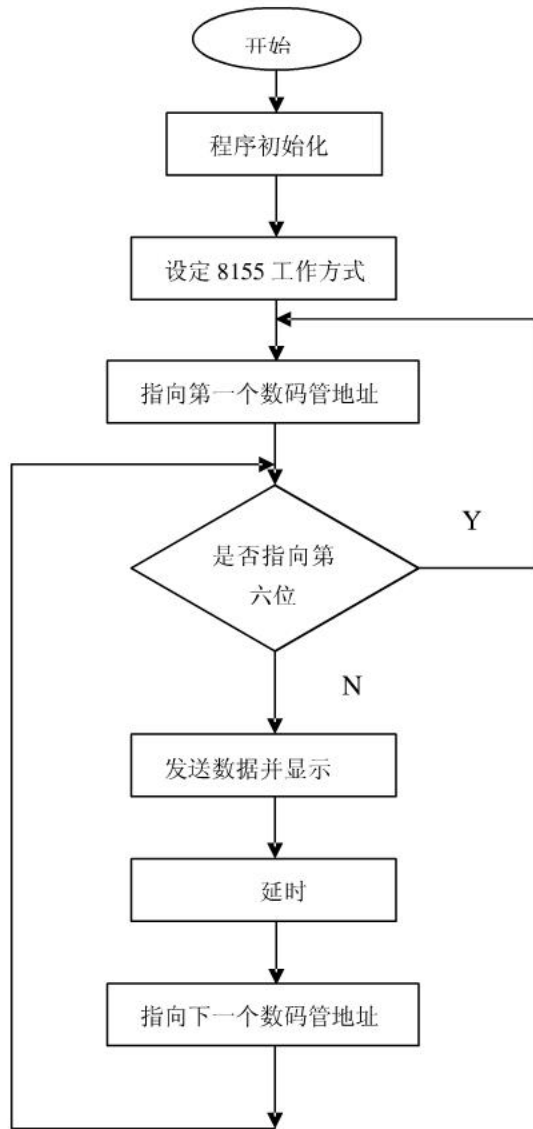


图 4-4-5 LED 软件流程

例 4-4-1: LED.C 动态显示接口程序

```

/*****

FileName:      LED.c

Description:    6 位数码管上显示函数

*****/

#include <IO.h> // 定义各种宏，作为总的头文件包含各需要的头文件

//数码管显示 0~9 是的数码段的数值
  
```

```
int S_Data[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
```

```
//数码管 1—6 位码段的数值
```

```
int W_Data[6]={0x01,0x02,0x04,0x08,0x10,0x20};
```

```
/**
```

```
函数名称：    delay()
```

```
函数功能：    实现延时功能
```

```
函数参数：    x
```

```
返回：        无
```

```
*/
```

```
void delay(uchar x)
```

```
{  
    while(x--);  
}
```

```
/**
```

```
函数名称：    LED()
```

```
函数功能：    实现数码管的显示功能
```

```
函数参数：    Data[6]
```

```
返回：        无
```

```
*/
```

```
void LED(unsigned char Data[6])
```

```
{  
    int i=0,j=0;  
    for(i=0;i<6;i++)  
    {  
        for(j=0;j<5;j++)  
        {  
            W_IO=W_Data[5-i];  
            S_IO=S_Data[Data[i]];  
        }  
    }  
}
```

```

        delay(100);
    }
}

```

主函数调用显示函数的程序如下：

```

/*****
FileName:      main.c
Description:   实现 6 位数码管的显示控制
*****/

#include <IO.h>

void LED(int Data[6]);

unsigned char Led_number[6]={0,1,2,3,4,5}; //数码管要显示的 6 位数字

void main()
{
    PORT=0x03; //8155 的 A,B 口作为输出端口，C 口为输入端口

    while(1)
    {
        LED(Led_number);
    }
}

```

5、设计调试步骤

- 1) 把电源模块、CPU 模块、8155 扩展模块、LED 显示器模块、串口通信模块插在正确的位置上，并把跳帽进行正确短接。
- 2) 建立 Keil 工程，编辑、编译源代码并生成可下载的 HEX 文件
- 3) 连接串口线，通过串口通信模块使用 STC-ISP 软件下载程序
- 4) 观察 LED 显示数据，如果以上步骤正确的话，在 LED 上应能显示 0、1、2、3、4、5 六个数字。

6、思考和发挥部分

- 1) 如何修改程序，使显示的数字类似跑马灯一样不停地闪烁？
- 2) 如何修改程序，使显示循环？
- 3) 如何修改程序，使显示左循环？

4) 如何修改程序，使显示两位移动？

4. 4. 4 电子设计 DIY

设计要求：请参考上述 LED 电原理图在 EDP 试验仪的面包板上自行完成一个 6 位 8 段数码管显示电路，画出电原理图并完成相应的软硬件设计。

设计提示：可以采用共阴或公阳 LED 数码管，与实验仪单片机的连接可以采用串行或并行方式。试验仪面包板的引脚定义可参照图 4-4-6，与单片机的连接请参考图 4-4-2 和图 4-4-4。

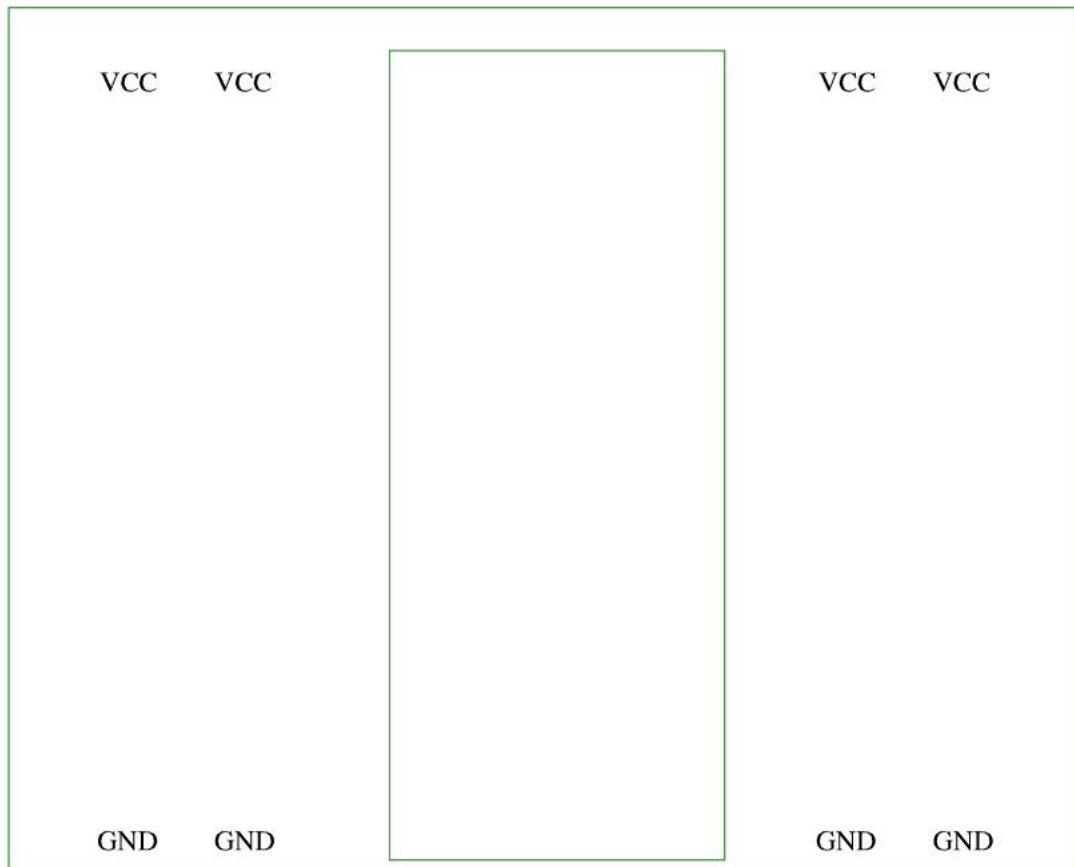


图 4-4-6 试验仪面包板的引脚定义