

数码管显示

1. 实验目的:

学会用数码管显示数字。

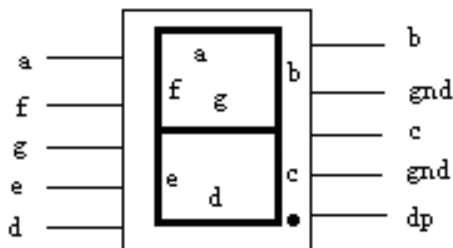
2. 知识准备:

一、数码显示原理

LED 数码显示器是由若干个发光二极管组成的，当发光二极管导通时，相应的点或线段发光，将这些二极管排成一定图形，控制不同组合的二极管导通，就可以显示出不同的字形。

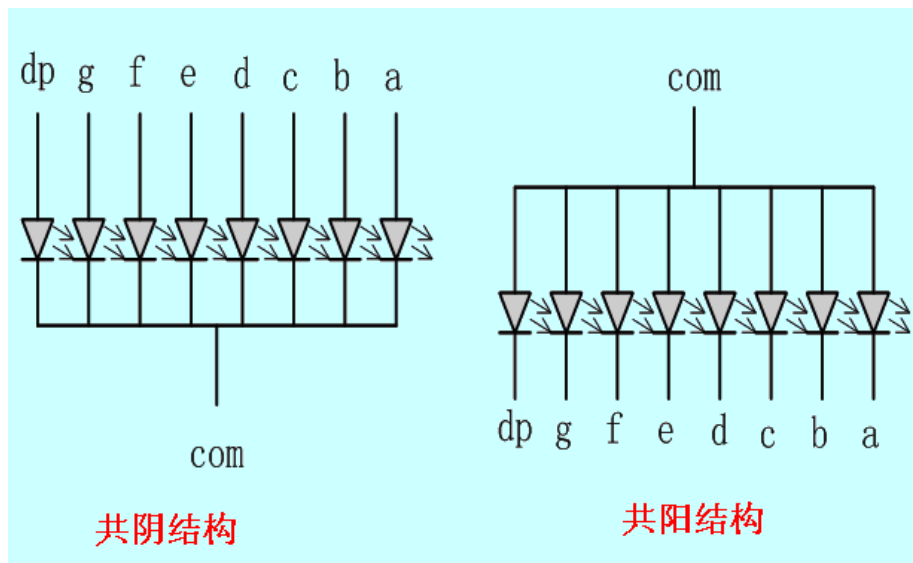
二、八段显示器

图是本次实验使用的数码外围引脚。



图

数码管分共阳极和共阴极，内部原理结构如图所示。本次实验使用的是共阴极数码管。



图

三、数码管段码的编码:

8 段正好是一个字节，通常采用如下编码方式。这种编码需单片机数据总线的 D0-D7 分别和数码管的 a、b、c、...dp 对应相连。

其编码如下:

共阳极数码管 a b c d e f g h a-h h-a
0 0 0 0 0 0 1 1 03 C0

1	1 0 0 1 1 1 1 1	9F F9
2	0 0 1 0 0 1 0 1	25 A4
3	0 0 0 0 1 1 0 1	0D B0
4	1 0 0 1 1 0 0 1	99 99
5	0 1 0 0 1 0 0 1	49 92
6	0 1 0 0 0 0 0 1	41 82
7	0 0 0 1 1 1 1 1	1F F8
8	0 0 0 0 0 0 0 1	01 80
9	0 0 0 0 1 0 0 1	09 90
A	0 0 0 1 0 0 0 1	11 88
B	1 1 0 0 0 0 0 1	C1 83
C	0 1 1 0 0 0 1 1	63 C6
D	1 0 0 0 0 1 0 1	85 A1
E	0 1 1 0 0 0 0 1	61 86
F	0 1 1 1 0 0 0 1	71 8E

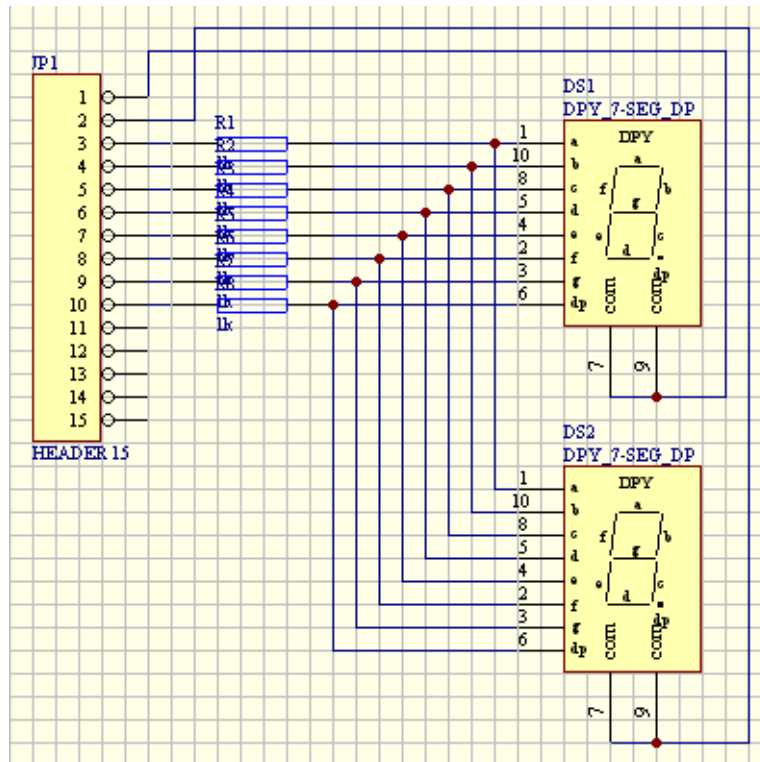
共阴极数码管 a b c d e f g h a-h h-a

0	1 1 1 1 1 1 0 0	FC 3F
1	0 1 1 0 0 0 0 0	60 06
2	1 1 0 1 1 0 1 0	DA 5B
3	1 1 1 1 0 0 1 0	F2 4F
4	0 1 1 0 0 1 1 0	66 66
5	1 0 1 1 0 1 1 0	B6 6D
6	1 0 1 1 1 1 1 0	BE 7D
7	1 1 1 0 0 0 0 0	E0 07
8	1 1 1 1 1 1 1 0	FE 7F
9	1 1 1 1 0 1 1 0	F6 6F
A	1 1 1 0 1 1 1 0	EE 77
B	0 0 1 1 1 1 1 0	3E 7C
C	1 0 0 1 1 1 0 0	9C 39
D	0 1 1 1 1 0 1 0	7A 5E
E	1 0 0 1 1 1 1 0	9E 79
F	1 0 0 0 1 1 1 0	8E 71

本实验采用共阴极数码管的编码。

3. 实验原理：

图是本实验的数码管的原理图，实验采用动态显示，这样可以节省单片机的 I/O 口。两片数码的引脚 1、2、3、4、5、8、10、6 分别并联接端口 PTA0~PTA7，数码管一引脚 7、9 接 PTD6、数码管二引脚 7、9 接 PTA7，用着片选。例如让左边数码管显示，将 PTA7 置高电平，PTA 置 0x84，同理显示其他数字。



图

4. 实验步骤:

- 一、打开电脑
- 二、打开实验板和编程软件
- 三、编写程序
- 四、下载调试程序

```
void main() {
PTA=0x00;
PTD_PTD6=0;
PTD_PTD7=0;
DDRA=0xff;
DDRD_DDRD6=1;
DDRD_DDRD7=1;
while(1) {
int i0=0,i1=0,i2=0,i3=0,i4=0,i5=0,i6=0,i7=0,i8=0,i9=0;
while(i0<100)
{
PTA=0xbd;//0
PTD_PTD6=1;
PTD_PTD7=0;
wait(5);
PTA=0xbd;//0
}
```

```
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i0++;
}
while(i1<100)
{
    PTA=0x84;//1
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0x84;//1
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i1++;
}
while(i2<100)
{
    PTA=0xd9;//2
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0xd9;//2
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i2++;
}
while(i3<100)
{
    PTA=0xcd;//3
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0xcd;//3
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i3++;
}
while(i4<100)
{
```

```
    PTA=0xe4;//4
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0xe4;//4
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i4++;
}
while(i5<100)
{
    PTA=0x6d;//5
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0x6d;//5
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i5++;
}
while(i6<100)
{
    PTA=0x7d;//6
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0x7d;//6
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i6++;
}
while(i7<100)
{
    PTA=0x85;//7
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0x85;//7
    PTD_PTD6=0;
    PTD_PTD7=1;
```

```
    wait(5);
    i7++;
}
while(i8<100)
{
    PTA=0xfd; //8
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0xfd; //8
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i8++;
}
while(i9<100)
{
    PTA=0xed; //9
    PTD_PTD6=1;
    PTD_PTD7=0;
    wait(5);
    PTA=0xed; //9
    PTD_PTD6=0;
    PTD_PTD7=1;
    wait(5);
    i9++;
}
}
}
```