

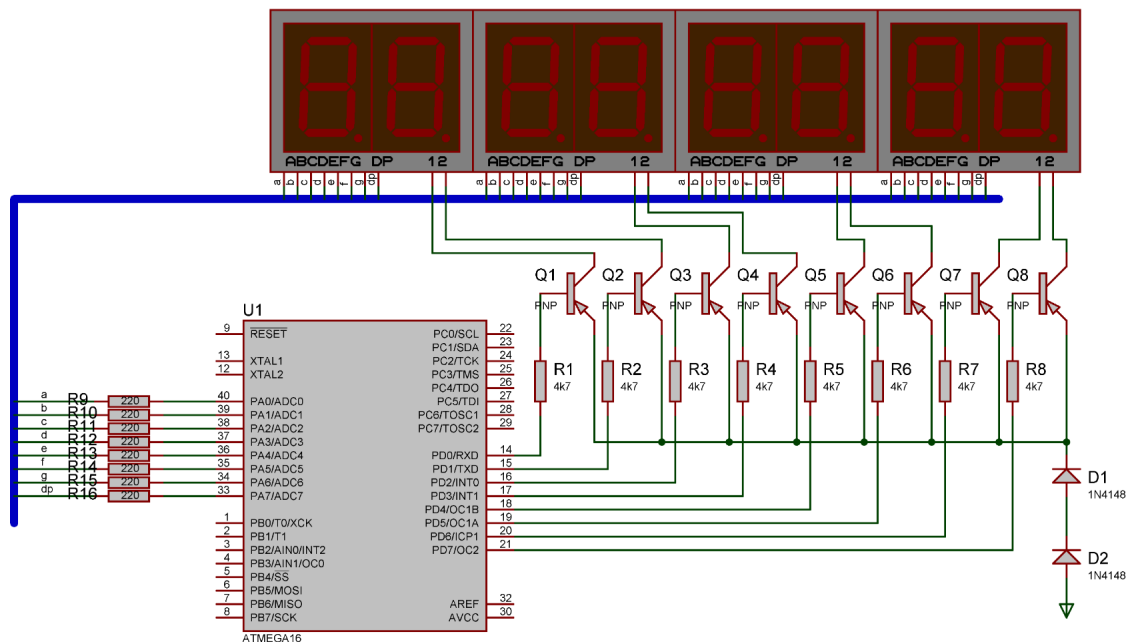
实验 5：数码管动态扫描显示

1. 试验描述：

实验使用 4 个两位的数码管，PD 口和 PA 口分别控制数码管段显示和位选，并且使用 PNP 三极管提高驱动能力。

2.系统框图：

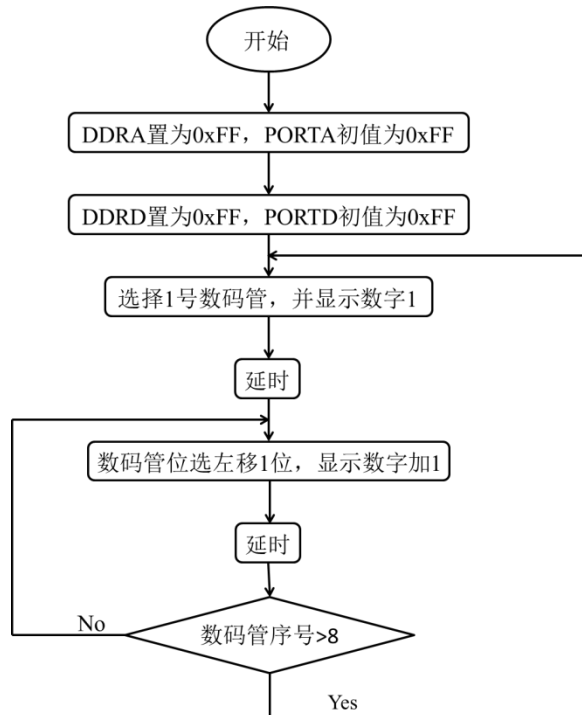
➤ 硬件电路



➤ 元件清单

单片机 ATmega16	电阻 220 欧姆*8 4k7 欧姆*8	两位带小数点 7 段数码管*4
PNP 三极管*8		

➤ 软件流程



3. 程序代码:

➤ ICC 程序

```
//PA 口接数码管数据口, PD 口接数码管位选口
#include <iom16v.h>
#define uchar unsigned char
#define uint unsigned int
const uchar dis_code[10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90};
uchar dis_buf[8]; //显存缓冲区基地址
uchar dis_index; //显示索引, 用于标识当前显示的数码管和缓冲区的偏移量
uchar dis_digit; //位选通值, 用于选通当前数码管的数值

//毫秒级延时函数
void delayms(uint n)
{
    uint i = 0, j = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < 250; j++)
            ;
}
```

```

//主函数
void main(void)
{
    uchar i;
    DDRA = 0xff;
    PORTA = 0xff;
    DDRD = 0xff;
    PORTD = 0xff;

    for (i = 0; i < 8; i++)
    {
        dis_buf[i] = dis_code[i + 1];
    }
    dis_digit = 0xfe;
    dis_index = 0;

    while (1)
    {
        PORTA = 0xff;
        PORTD = dis_digit;
        PORTA = dis_buf[dis_index];

        delays(1);
        dis_digit = (dis_digit << 1) | 0x01;
        dis_index++;
        if (dis_index == 8)
        {
            dis_digit = 0xfe;
            dis_index = 0;
            PORTD = 0xff;
        }
    }
}

```

➤ CAVR 程序

```

//PA 口接数码管数据口，PD 口接数码管位选口
#include <mega16.h>

#define uchar unsigned char
#define uint unsigned int

flash uchar dis_code[10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0
xf8, 0x80, 0x90};

```

```

uchar dis_buf[8]; //显存缓冲区基地址
uchar dis_index; //显示索引，用于标识当前显示的数码管和缓冲区的偏移量
uchar dis_digit; //位选通值，用于选通当前数码管的数值

//毫秒级延时函数
void delayms(uint n)
{
    uint i = 0, j = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < 250; j++)
            ;
}

//主函数
void main(void)
{
    uchar i;
    DDRA = 0xff;
    PORTA = 0xff;
    DDRD = 0xff;
    PORTD = 0xff;

    for (i = 0; i < 8; i++)
    {
        dis_buf[i] = dis_code[i + 1];
    }

    dis_digit = 0xfe;
    dis_index = 0;

    while (1)
    {
        PORTA = 0xff;
        PORTD = dis_digit;
        PORTA = dis_buf[dis_index];

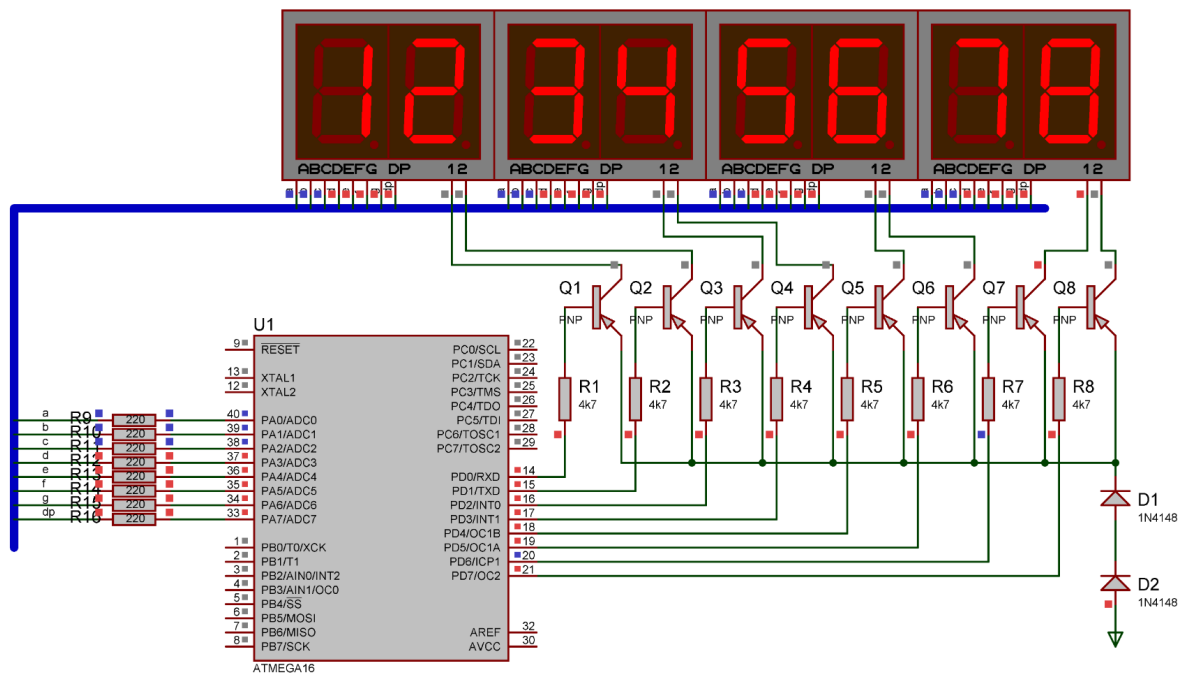
        delayms(1);
        dis_digit = (dis_digit << 1) | 0x01;
        dis_index++;
        if (dis_index == 8)
        {
            dis_digit = 0xfe;
            dis_index = 0;
            PORTD = 0xff;

```

```
}  
}  
}
```

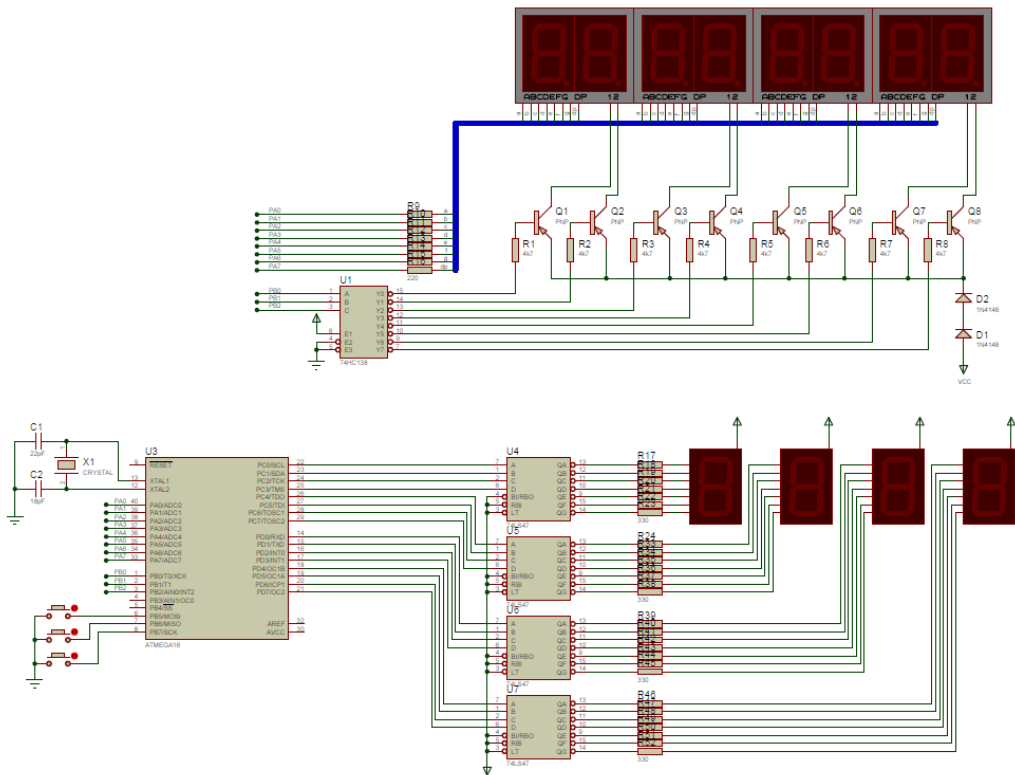
4.仿真结果:

数码管从左至右依次显示数字 1-8。



5.改进:

将上次实验的静态数码管稍加改进，再结合此次实验，实现数字时钟。



➤ main.c 改写如下:

```
#include <mega16.h>
#include "digitalclock.h"
```

```
void main(void)
{
    DigitalclockInit();

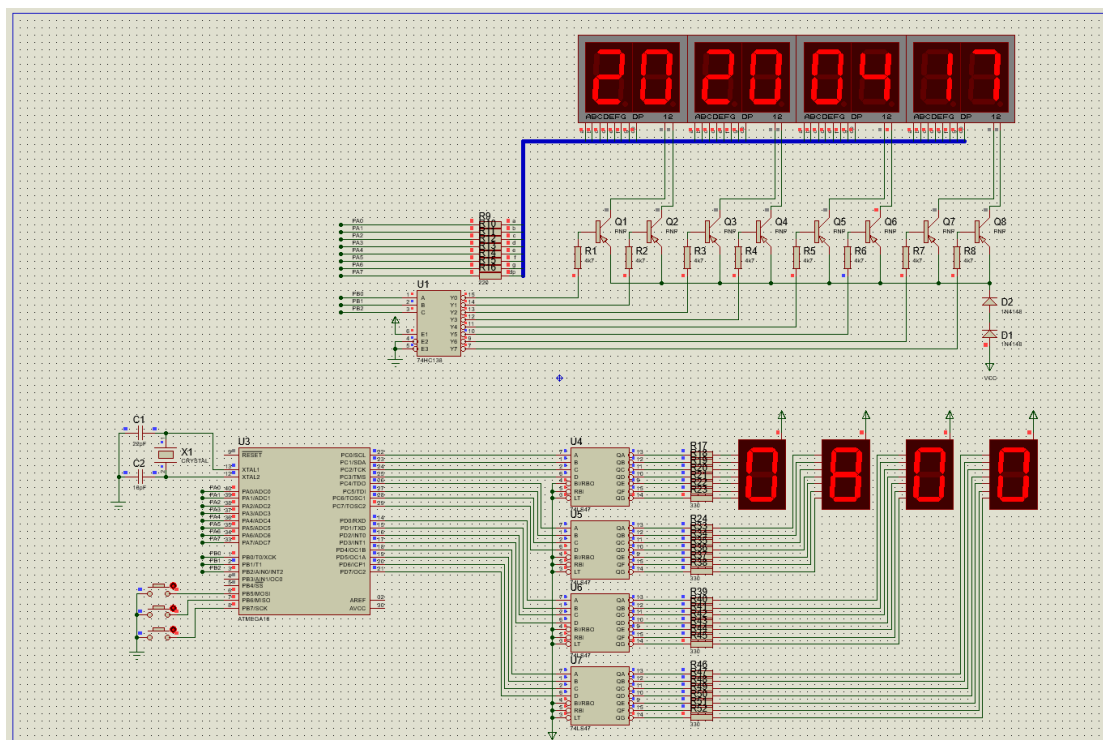
    DigitalclockWork();

    while (1)
    {
    }
}
```

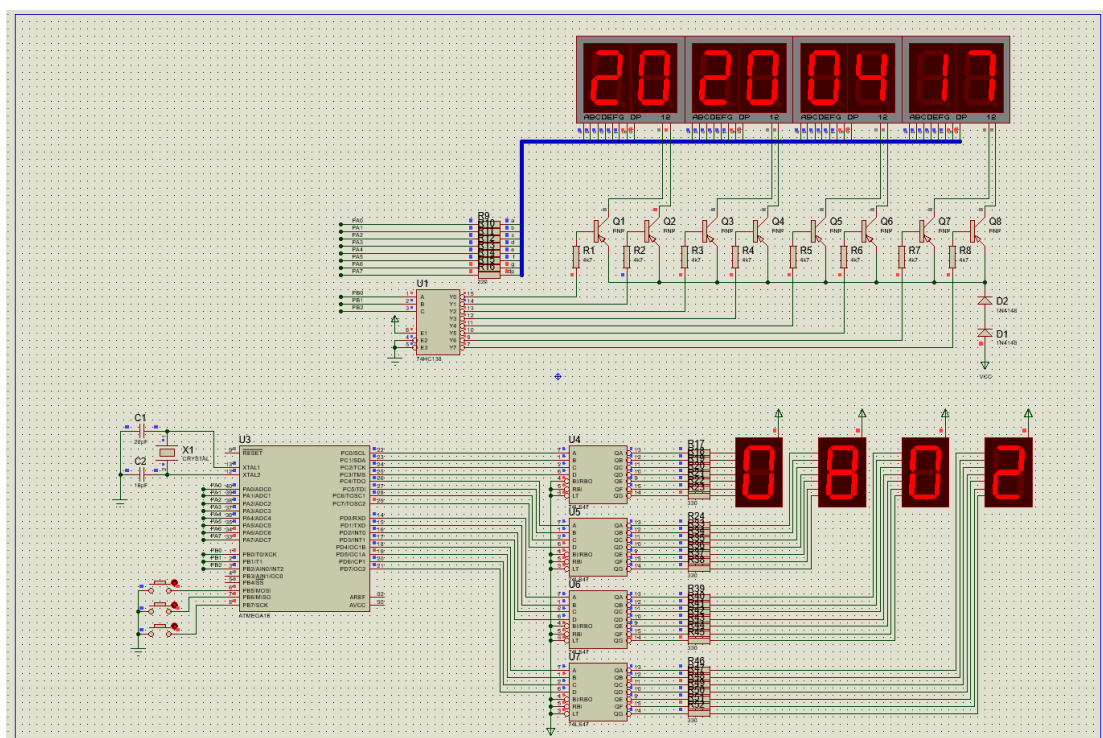
➤ mydisplay.c、mydisplay.h、digital.c、digital.h 文件请见附录

➤ 仿真结果如下

初始状态



数码管分别显示年月日、时分



下方三个按钮分别进入设置模式，调整设置的位，设置值。

附录:



main.c



mydelay.c



mydelay.h



digitalclock.c



digitalclock.h



mydisplay.c



mydisplay.h