

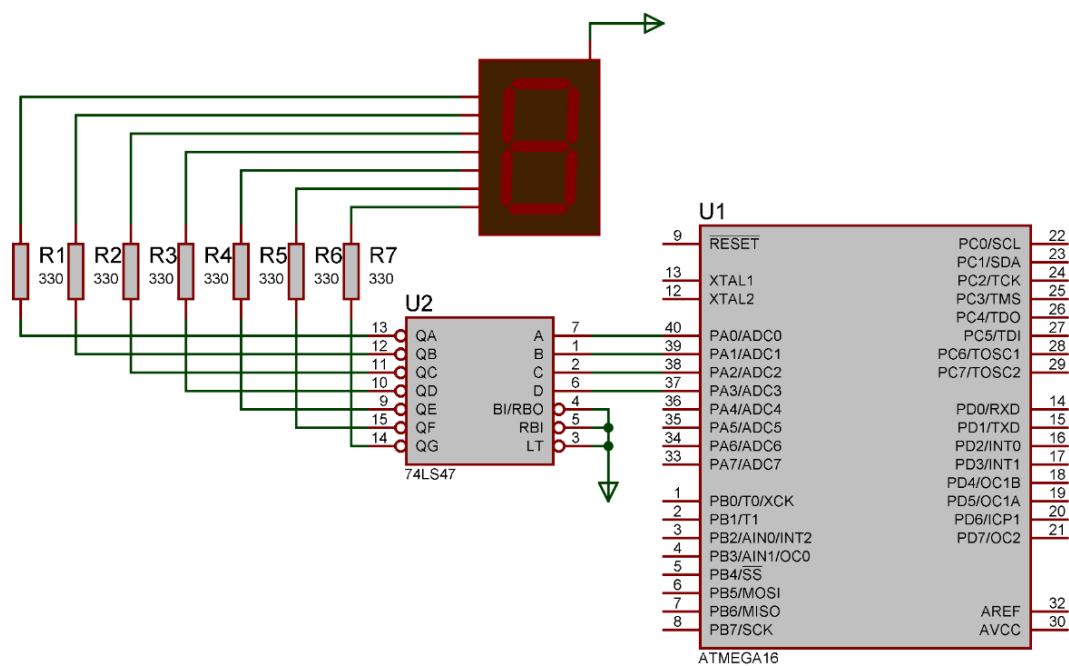
# 实验 4：静态数码管扫描

## 1. 试验描述：

74LS47 是一个 7 段数码管的驱动芯片，通过它解码，可以直接把数字转化为数码管的显示数字，从而简化了程序，节约了单片机的 I/O 开销。本实验使用单片机通过 74LS47 驱动一个 7 段数码管。

## 2.系统框图：

➤ 硬件电路



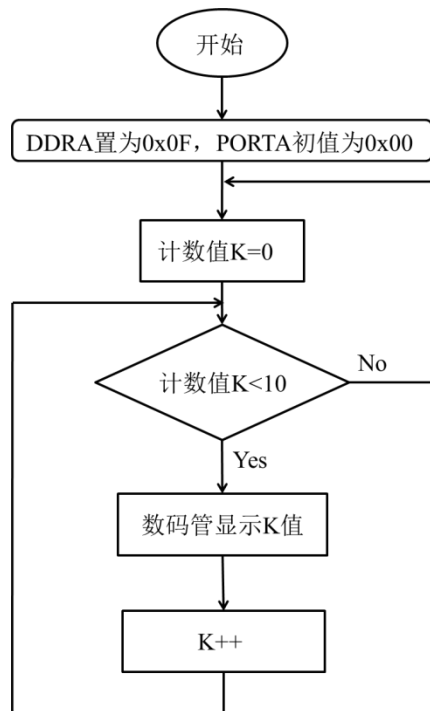
➤ 元件清单

单片机 ATmega16	电阻 330 欧姆*7	74LS47
7 段数码管		

## ➤ 74LS47 译码功能

74LS47 是 BCD-7 段数码管译码器 / 驱动器，74LS47 的功能用于将 BCD 码转化成数码块中的数字，通过它来进行解码，可以直接把数字转换为数码管的数字，从而简化了程序，节约了单片机的 I/O 开销。

## ➤ 软件流程



## 3. 程序代码:

### ➤ ICC 程序

```
#include <iom16v.h>

#define uchar unsigned char
#define uint unsigned int

void delays(uint n)
{
    uint i = 0, j = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < 100; j++)
            ;
}
```

```

int main(void)
{
    uchar k = 0;
    DDRA = 0xff; //设置 PA 口输出
    PORTA = 0xff; //PA 口高电平
    while (1)
    {
        for (k = 0; k < 10; k++)
        {
            PORTA = k;
            delayms(300);
        }
        delayms(300);
    }
    return 0;
}

```

## ➤ CAVR 程序

```

#include <mega16.h>

#define uchar unsigned char
#define uint unsigned int

void delayms(uint n)
{
    uint i = 0, j = 0;
    for (i = 0; i < n; i++)
        for (j = 0; j < 100; j++)
            ;
}

void main(void)
{
    uchar k = 0;
    DDRA = 0xff; //设置 PA 口输出
    PORTA = 0xff; //PA 口高电平

    while (1)
    {
        for (k = 0; k < 10; k++)
        {
            PORTA = k;
            delayms(300);
        }
    }
}

```

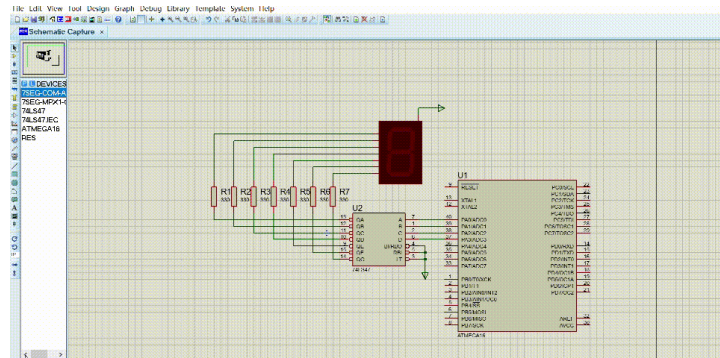
```

    }
    delayms(300);
}
}

```

## 4.仿真结果：

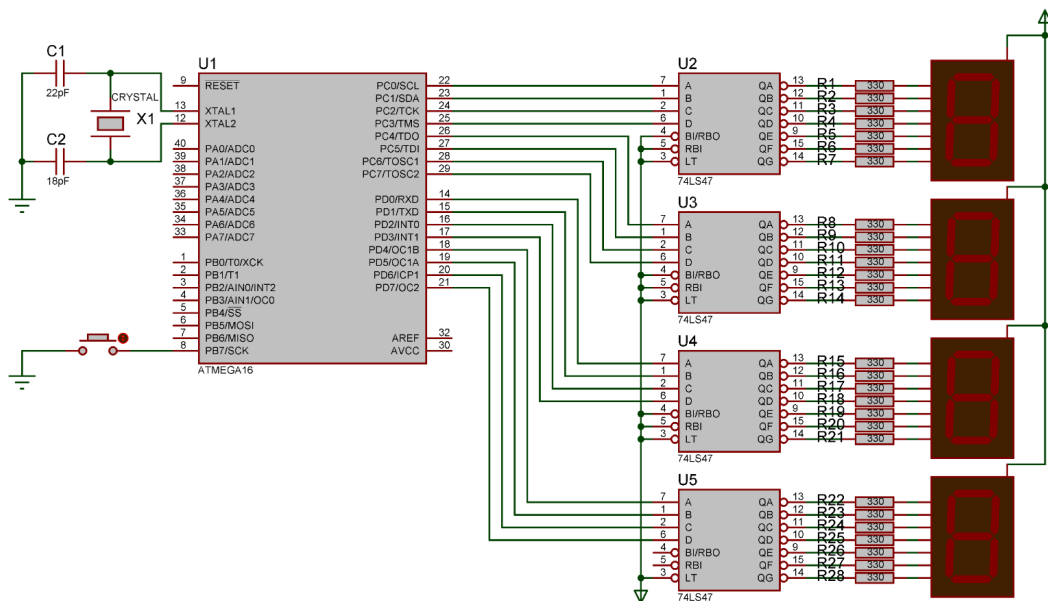
7 段数码管循环显示 0-9；



(双击图片，展示试验仿真结果)

## 5.改进：

结合之前写好的使用定时器延时的代码，实现秒表的功能，能进行 100 秒里的计时，精度为 10ms。



➤ **main.c** 程序如下:

```
#include <mega16.h>
#include <mydelay.h>

#define uchar unsigned char
#define uint unsigned int

uint time = 0;      //时间
uchar timeFlag = 0; //计时状态

void Display(void); //显示时间
void Button(void);  //判断并处理按钮事件

void main(void)
{

    DDRC = 0xff; //PC 口输出
    PORTC = 0x00;
    DDRD = 0xff; //PD 口输出
    PORTD = 0x00;
    DDRB = 0x00; //PB7 输入
    PORTB = 0x80;

    DelayInit(); //延时初始化

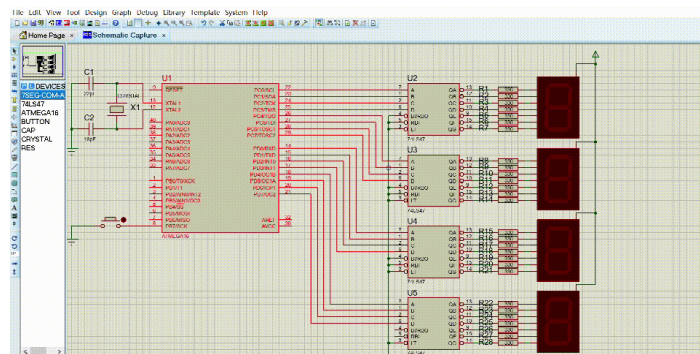
    while (1)
    {
        Button();
        if (timeFlag == 0)
            time = 0;
        if (time < 9999 && timeFlag == 1)
            time++;
        Display();
        DelayMs(10); //延时 10 毫秒
    }
}

void Display(void)
{
    PORTC = time / 1000 + (time / 100 % 10) * 16;
    PORTD = time / 10 % 10 + (time % 10) * 16;
}
```

```
void Button(void)
{
    if (!(PINB & 0x80))
    {
        while (!(PINB & 0x80))
        ;
        timeFlag++;
        if (timeFlag > 2)
            timeFlag = 0;
    }
}
```

## ➤ 仿真结果

初始状态四个数码管全零，第一次按下按键开始计时，第二次按下停止计时，再按下依次复位。



(双击图片，展示试验仿真结果)

附录:



main.c



mydelay.c



mydelay.h