

单片机（AT89C51）的串行通信实验

原创

叶玉恒 于 2020-04-28 13:31:20 发布 13907 收藏 324

分类专栏：[单片机](#) 文章标签：[单片机](#) [串口通信](#)

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_45629315/article/details/105810753

版权



[单片机](#) 专栏收录该内容

8 篇文章 23 订阅

订阅专栏

串口通信的重要性和基础就不用我多讲了吧？基础不懂直接点链

接：https://blog.csdn.net/weixin_45629315/article/details/105717414

这一次的重点是讲解几个串口的实验来加深自己对串口通信的理解。

目录

实验一（将一个简单的字符串输出到单片机中）：

实验一-----电路图：

实验一-----代码段：

实验一-----结果：

实验二（通过串口通信的方式实现开关控制灯的亮灭）

实验二-----电路图

实验二-----代码块

发送机：

接收机：

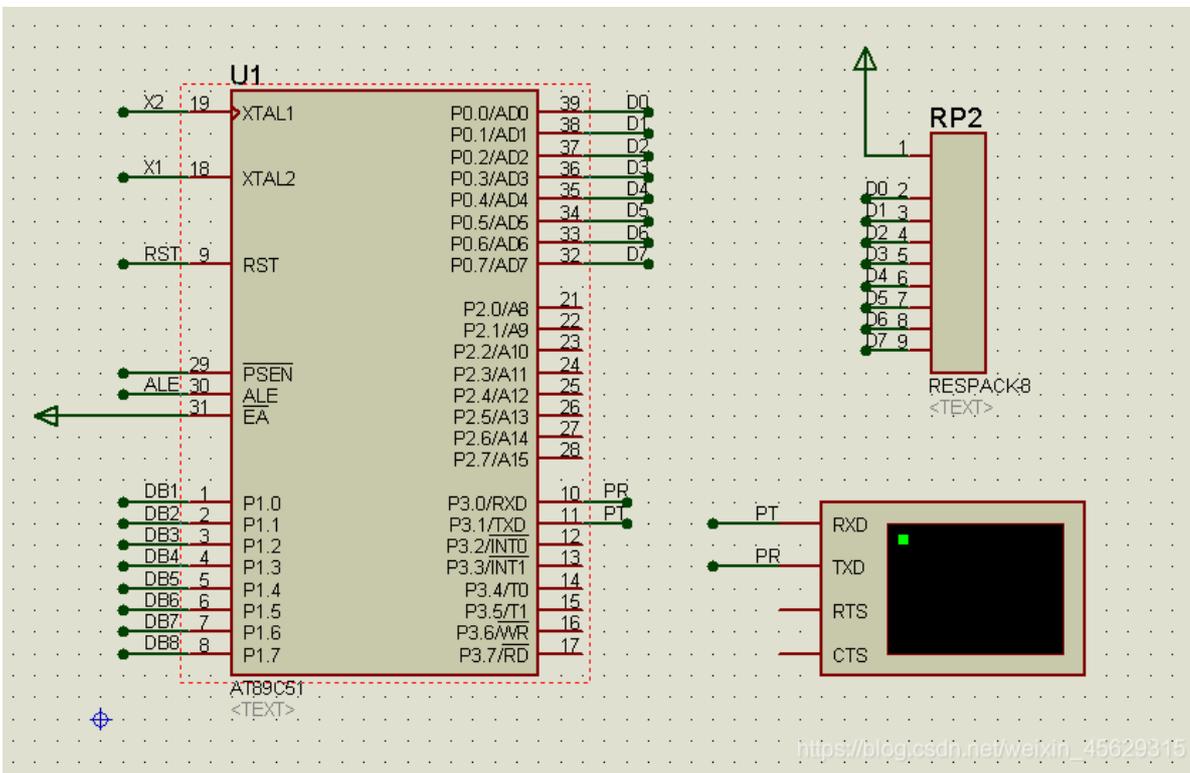
实验二-----结果

实验一（将一个简单的字符串输出到单片机中）：

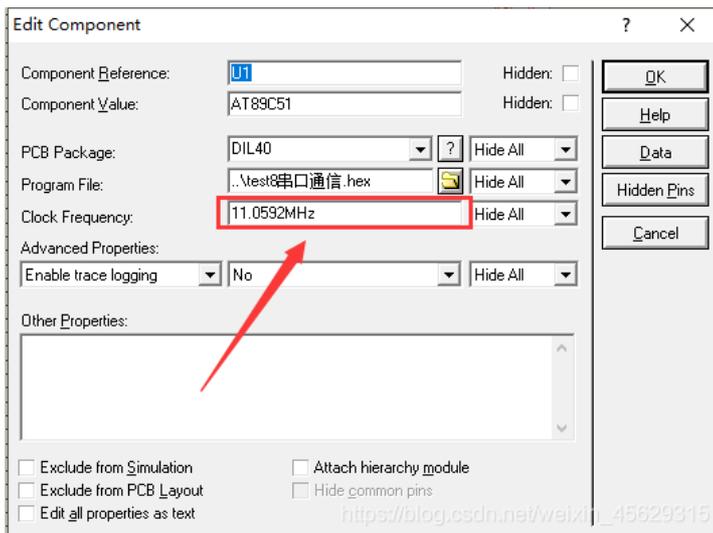
实验目的：

1. 实现输出“201805021123 YEYUEHNG ”
2. 用VIRTUAL TERMINAL显示出来

实验一-----电路图：



实验图比较的简单明了，主要要注意的就是单片机的晶振要设置为11.0592MHz，不然不会出现预想的结果，如下图设置：



实验一-----代码段：

```

#include<reg51.h>
unsigned char code str[]="201805021123 YEYUEHNG "; //定义一个发送的数据

//函数的功能是：向PC端发送一个字节数据

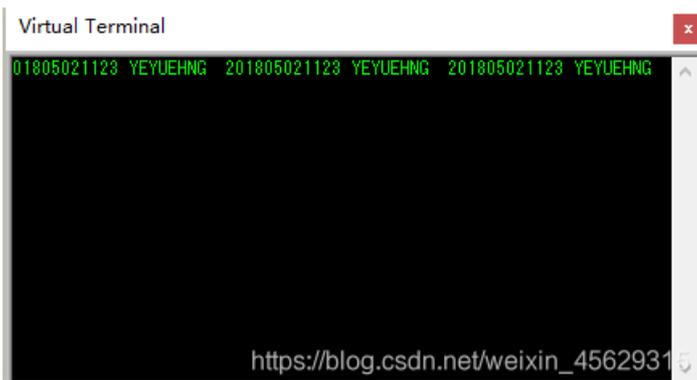
void send(unsigned char dat)
{
    SBUF=dat; //发送数据
    while(TI==0); //检查发送完成中断标志，发送完成为1，未完成为0
    TI=0; //复位发送标志位
}

void delay(unsigned int n) //设置延时函数
{
    unsigned int i,j;
    for(i=0;i<n;i++)
        for(j=249;j>0;j--)
            ;
}

void main(void)
{
    unsigned int i;
    TMOD=0x20; //TMOD=0010 0000,设置定时器T1工作于方式二
    SCON=0x60; //SCON=0100 0000,设置串口的工作方式为方式1
    PCON=0x00; //PCON=0000 0000,波特率为9600 晶振为11.0592
    TH1=0xfd; //设置定时器T1的初值
    TL1=0xfd; //定时器T1自动填充的值
    TR1=1; //启动定时器T1
    while(1)
    {
        i=0;
        while(str[i]!='\0') //循环发送字节数组中的数据
        {
            send(str[i]);
            i++;
            delay(1);
        }
        delay(1000);
    }
}

```

实验一-----结果：



如果实验中得到的是你所设置的字符串的话就是实验成

功了

实验一比较的简单，我们可以通过实验一思考可不可以通过开关来控制灯的亮灭（实验二）

实验二比较的贴合实际中用到的情况相比之下实验二肯定会比较有难度，不过有了实验一的基础也就可以顺藤摸瓜的得出实验的代码了。

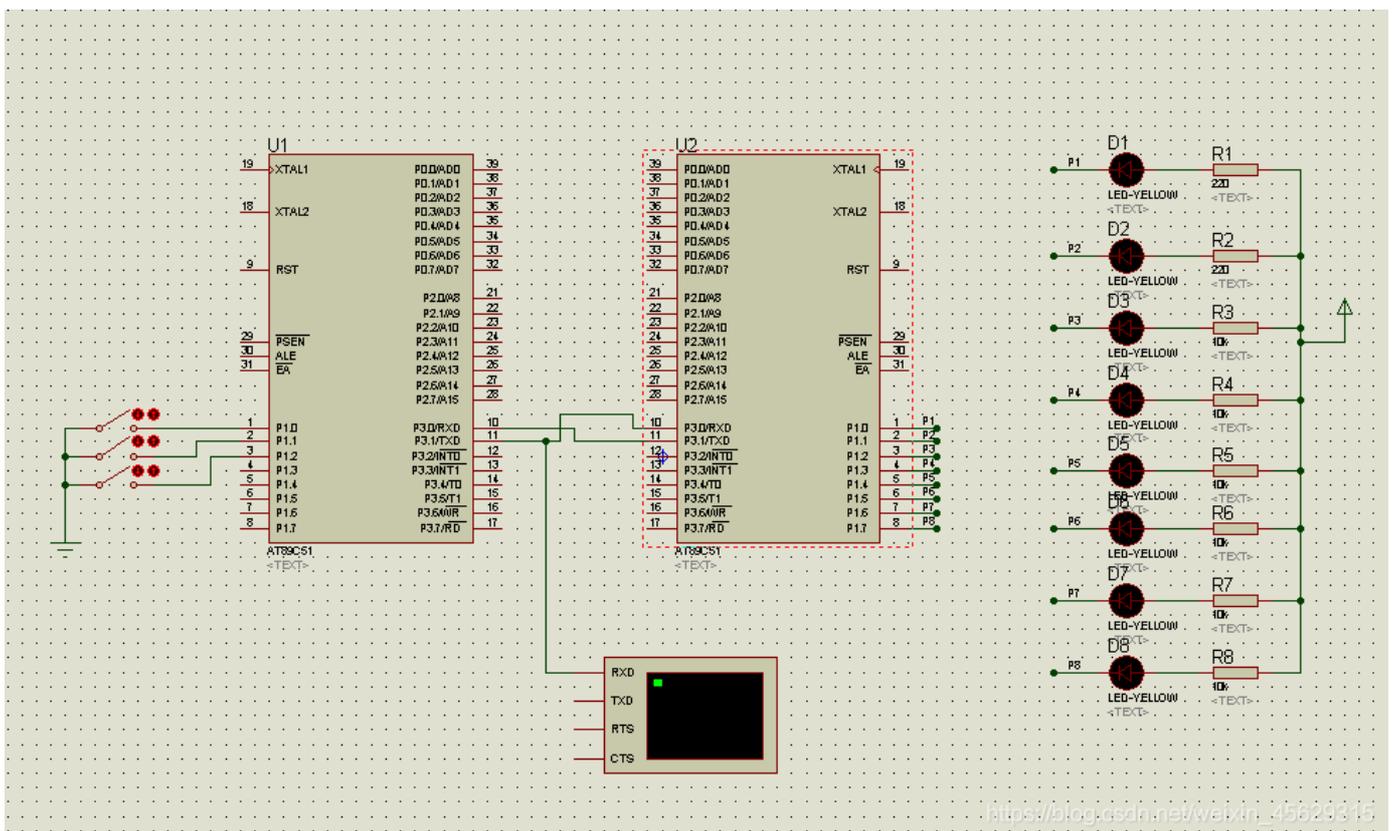
实验二（通过串口通信的方式实现开关控制灯的亮灭）

实验的目的和要求：

- (1) 设计2个单片机相互通信，第一个单片机的按键发送指令控制第二个单片机上的led灯。
- (2) 第一个单片机三个按键，控制发送三个指令，分别为"15000101"，"15000102"，"15000103"。
- (3) 第二个单片机接收指令，如果接收到第一个指令，则P1.0口灯亮，如果接收到第二个指令，则P1.1口灯亮，如果接收到第三个指令，则P1.1和P1.0口灯同时亮。

实验的目的中只提到了两个灯的操作，其实八个灯也就没什么问题了，可以以此类推的实现多灯的控制。

实验二-----电路图



注：需要设置两个单片机的晶振为11.0592MHz

led灯旁边的电阻值不要太大，不然led无法点亮

实验二-----代码块

实验涉及两个单片机所以需要两个代码块，分别为接收机和发送机的。

发送机：

```
#include<reg51.h>
unsigned char code str1[]="15000101";      //设置需要发送的协议代码
unsigned char code str2[]="15000102";
unsigned char code str3[]="15000103";
sbit P1_0=P1^0;                            //给相应端口一个编号
sbit P1_1=P1^1;
sbit P1_2=P1^2;

//函数的功能是：向PC端发送一个字节数据
void send(unsigned char dat)
{
    SBUF=dat;
    while(TI==0);
    TI=0;
}

void delay(unsigned int n)                 //延时函数的设置
{
    unsigned int i,j;
    for(i=0;i<n;i++)
        for(j=249;j>0;j--)
            ;
}

void main(void)
{
    unsigned int i;
    TMOD=0x20;                            //TMOD=0010 0000,设置定时器T1工作于方式二
    SCON=0x60;                            //SCON=0100 0000,设置串口的工作方式为方式1
    PCON=0x00;                            //PCON=0000 0000,波特率为9600 晶振为11.0592
    TH1=0xfd;                             //设置定时器T1的初值
    TL1=0xfd;                             //定时器T1自动填充的值
    TR1=1;                                 //启动定时器T1
    while(1)
    {
        if(P1_0==0)                      //判断P1_0口的开关状态
        {
            i=0;
            while(str1[i]!='\0')        //循环发送相应数组中的数据
            {
                send(str1[i]);
                i++;
                delay(1);
            }
            send('\0');                  //最后一位补上\0作为接收机判断结束的标准
            delay(1000);
        }
        else if(P1_1==0)                 //判断P1_1口的开关状态
        {
            i=0;
            while(str2[i]!='\0')        //循环发送相应数组中的数据
```

```

    {
        send(str2[i]);
        i++;
        delay(1);
    }
    send('\0'); //最后一位补上\0作为接收机判断结束的标准
    delay(1000);
}
else if(P1_2==0) //判断P1_2口的开关状态
{
    i=0;
    while(str3[i]!='\0') //循环发送相应数组中的数据
    {
        send(str3[i]);
        i++;
        delay(1);
    }
    send('\0'); //最后一位补上\0作为接收机判断结束的标准
    delay(1000);
}
}
}

```

接收机:

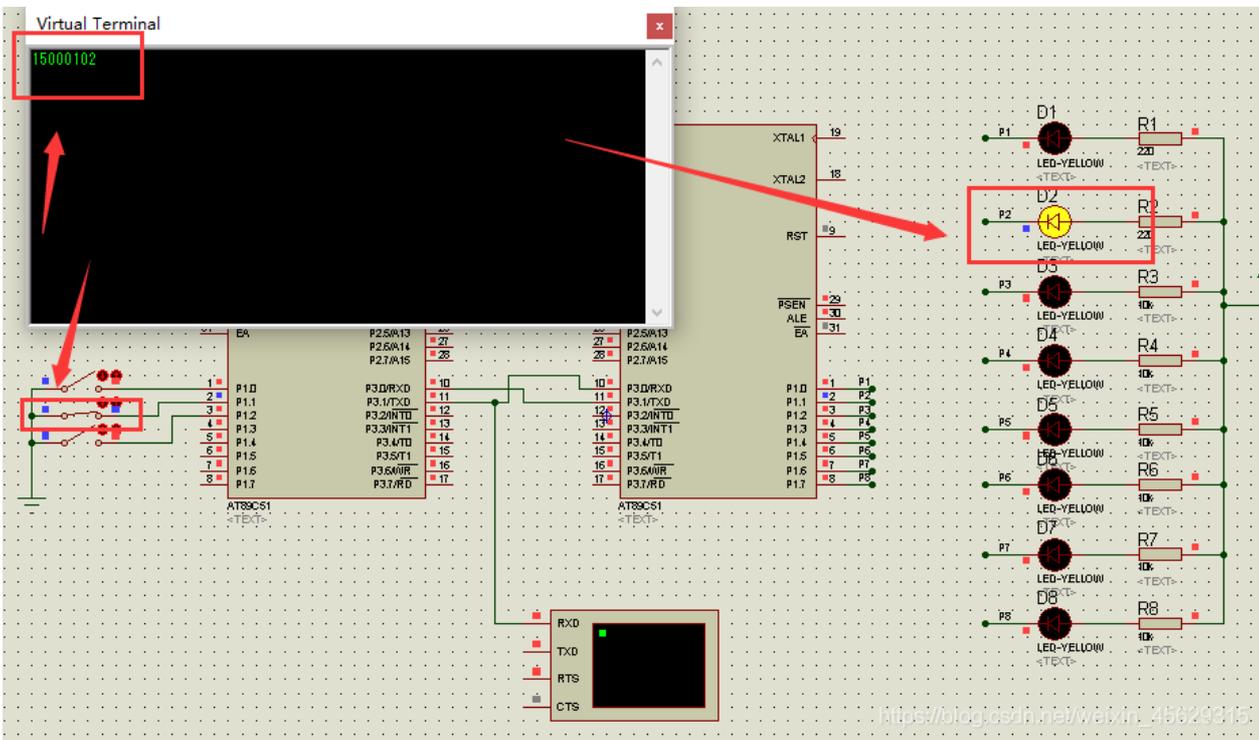
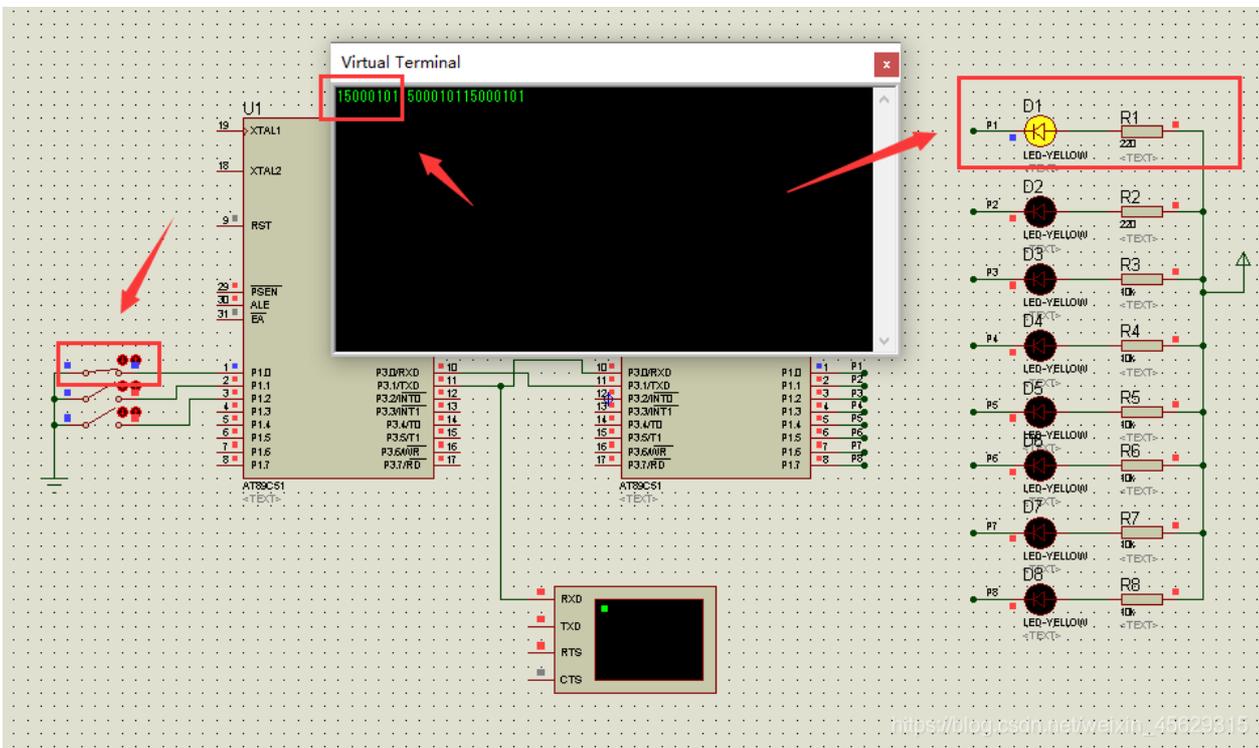
```

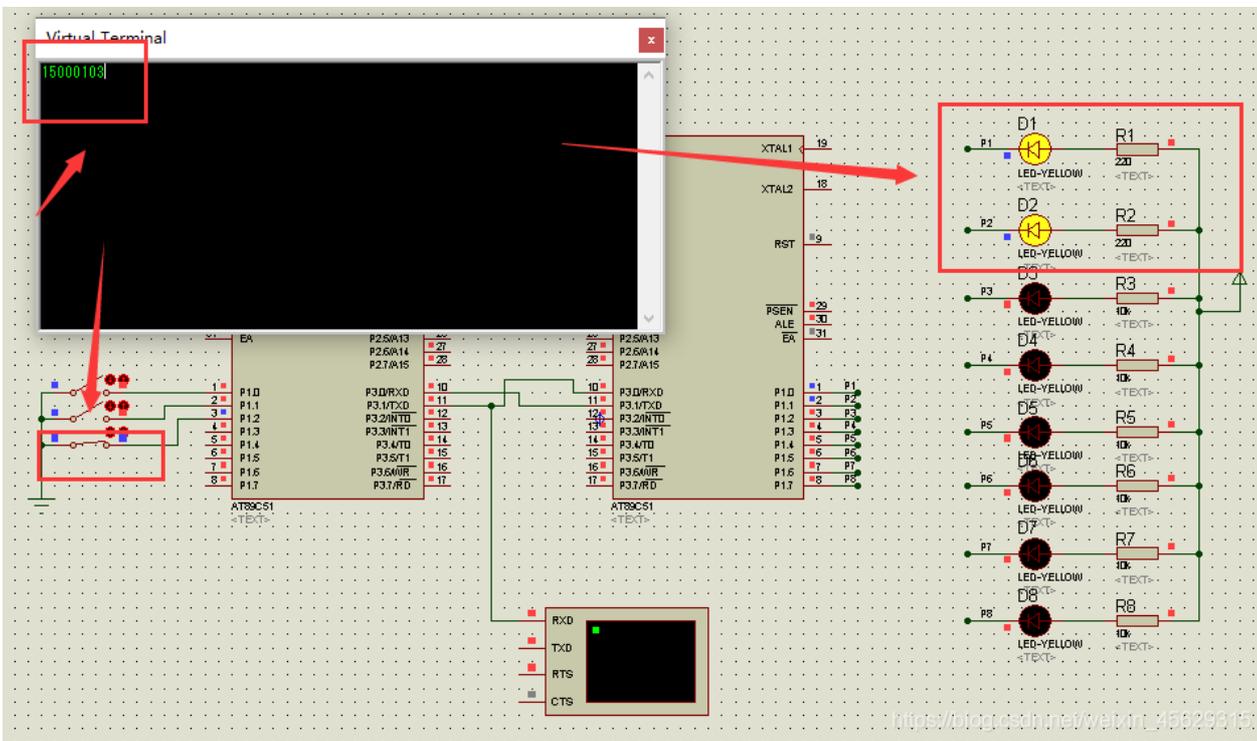
#include<reg51.h>
#define uchar unsigned char
uchar temp,flag;
uchar a[10];
uchar count=0;
void main()
{
    TMOD=0x20;           //定时器T1工作于方式2
    SCON=0x40;           //SCON=0100 0000B, 串口工作方式1
    PCON=0x00;           //PCON=0000 0000B, 波特率9600
    TH1=0xfd;           //根据规定给定时器T1赋初值
    TL1=0xfd;           //根据规定给定时器T1赋初值
    TR1=1;              //启动定时器T1
    REN=1;              //允许接收
    EA=1;               //开起总中断开关
    ES=1;               //需要用到串口的中断所以ES=1就是把串口的中断打开了
    while(1)            //进入循环
    {
        if(flag==1)     //等待中断的来临, 然后flag就会被赋值为1
        {
            if(a[7]==0x31) //如果发送的数据最后一位为1的话 (“15000101”最后一位1
                P1=0xFE;   //对应的就是十六进制的0x31) 0xfe对应1111 1110 就第
                            //一盏灯亮
            else if(a[7]==0x32) //以此类推
                P1=0xFD;
            else if(a[7]==0x33) //以此类推
                P1=0xFC;
            else
                P1=0x00;     //发送为其他数据时灯全亮
            flag=0;
        }
    }
}

void serial() interrupt 4 //设置中断函数
{
    temp=SBUF;           //当有中断时把传过来的数据暂存到temp中
    RI=0;                //把设置中断的标志初始化, 等待下一中断
    if(temp==0x00)      //当到达最后一位
    {
        a[count]=temp;
        count=0;        //计数器初始化为下一次中断准备
        flag=1;         //标志置1进入主函数, 进行端口的赋值
    }
    else
    {
        a[count]=temp;
        count=count+1;
    }
}

```

实验二----结果





有错请指出，不胜感激！！！！



最后的最后给我《点赞+收藏+关注》吧！
感谢各位！

https://blog.csdn.net/weixin_45629315

No pains No results