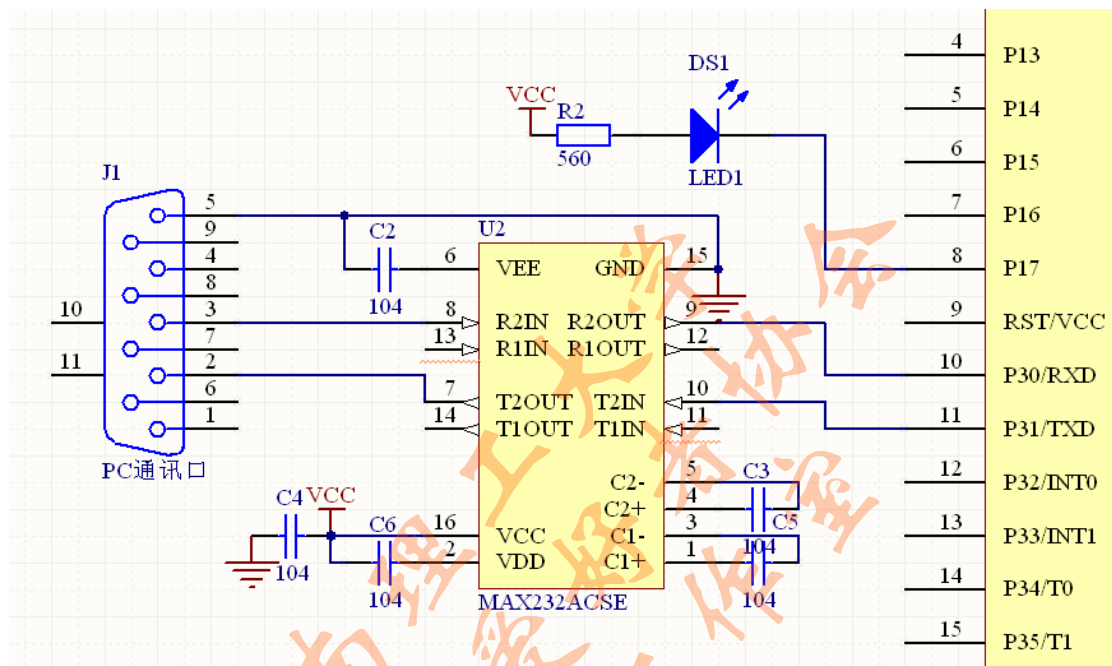


实验十：串口通讯演示程序

一.实验目的:

- 1,了解串口通讯的原理
- 2,通过串口实现与电脑的通讯

二,实验原理:



串口通讯原理图

RS-232 是美国电子工业协会正式公布的串行总线标准,也是目前最常用的串行接口标准,用来实现计算机与计算机之间、计算机与外设之间的数据通讯。

RS-232 串行接口总线适用于:设备之间的通讯距离不大于15m,传输速率最大为20kbps。RS-232 协议以 $-5V-15V$ 表示逻辑1;以 $+5V-15V$ 表示逻辑0。我们是用MAX232 芯片将RS232 电平转换为TTL 电平的。一个完整的RS-232 接口有22根线,采用标准的25 芯插头座。我们在这里使用的是简化的9 芯插头座。

注意我们在这里使用的晶振是11.0592M 的,而不是12M。因为波特率的设置的要求需要11.0592M 的。

单片机串口通信原理:

51 系列单片机内部集成了两个同名不同地址的串口缓冲区 SBUF,一个是发送缓冲区,一个是接收缓冲区,虽然二者名字相同但不相互干扰;发送数据时 MCU 自动把数据写到发送 SBUF,接收到的数据自动放到接收 SBUF,无需程序指定。串口发送和接收事件发生时,由硬件标志来通知处理器,RI 为接收事件发生标志,TI 为发送完成标志,‘1’为事件发生。在串口中断打开的条件下,二者任意一种情况发生都会引起中断,单片机程序可以根据 RI=1 还是 TI=1 进行相应的处理,

串口通信中还有重要的概念就是“波特率”。“波特率”就是单位时间(1S)内传送的二进制数据的位数,单位是Bit/S,例如:波特率为9600Bit/S是指1秒钟传送9600个二

进制位。波特率可以根据通信的两个系统自己设定，只要二者相同就可以。单片机中波特率的控制是使用定时器，让定时器产生定时中断去控制发送或接收的频率，根据波特率可以计算初定时器的初值，计算方法如下：

串口工作于方式0的波特率固定为晶振频率的1/12；

串口工作于方式2的波特率计算公式为： $B = f \times 2^{SMOD} / 64$ ；

串行口工作于方式1和方式3的波特率计算公式为：

A,T1工作于方式0和方式1时：

$$B = \frac{f * 2^{SMOD}}{(2^N - Z + 9) * 12 * 32}$$

T1方式0: N = 13

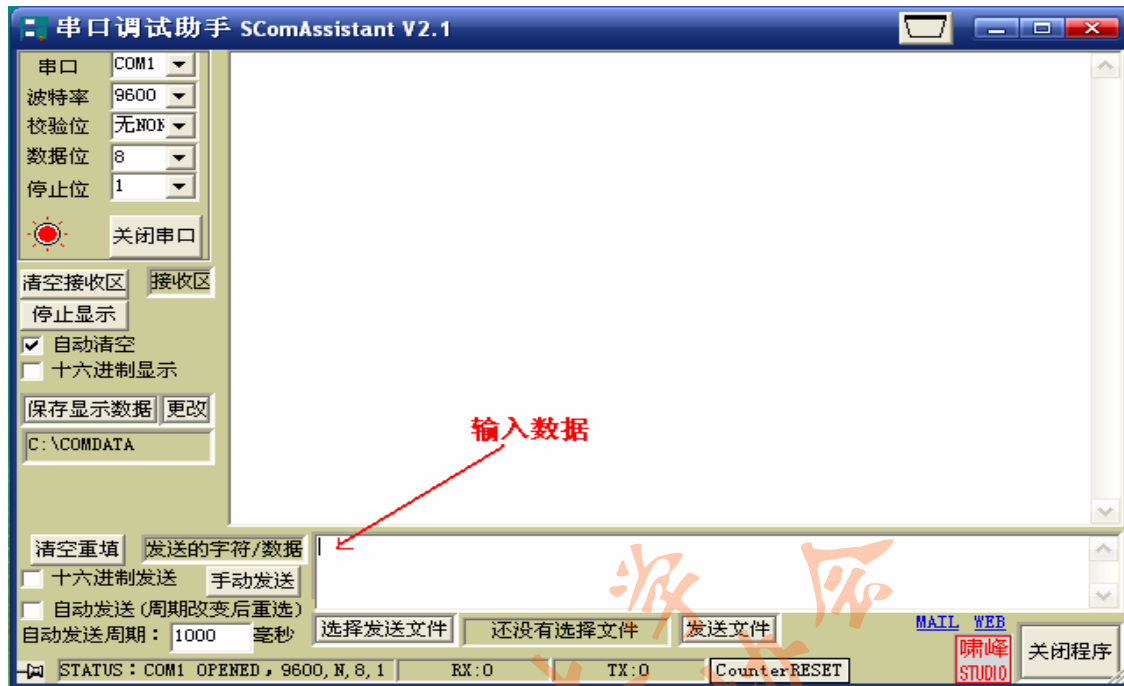
T1方式1: N = 16

B,T1工作于方式2时：

$$B = \frac{f * 2^{SMOD}}{(2^8 - Z) * 12 * 32}$$

Z为要计算的定时器初值，把它转换成十六进制后高位给TH，低位给TL。在这里我们利用定时器1作为波特率发生器，工作于方式2，波特率设定为9600bit，有上面的分析可知，只要我们取Z=253(0XFD)，在11.0592M的晶振下就可以产生需要的波特率。关于详细的串口使用资料读者可以参考光盘里的相关资料。

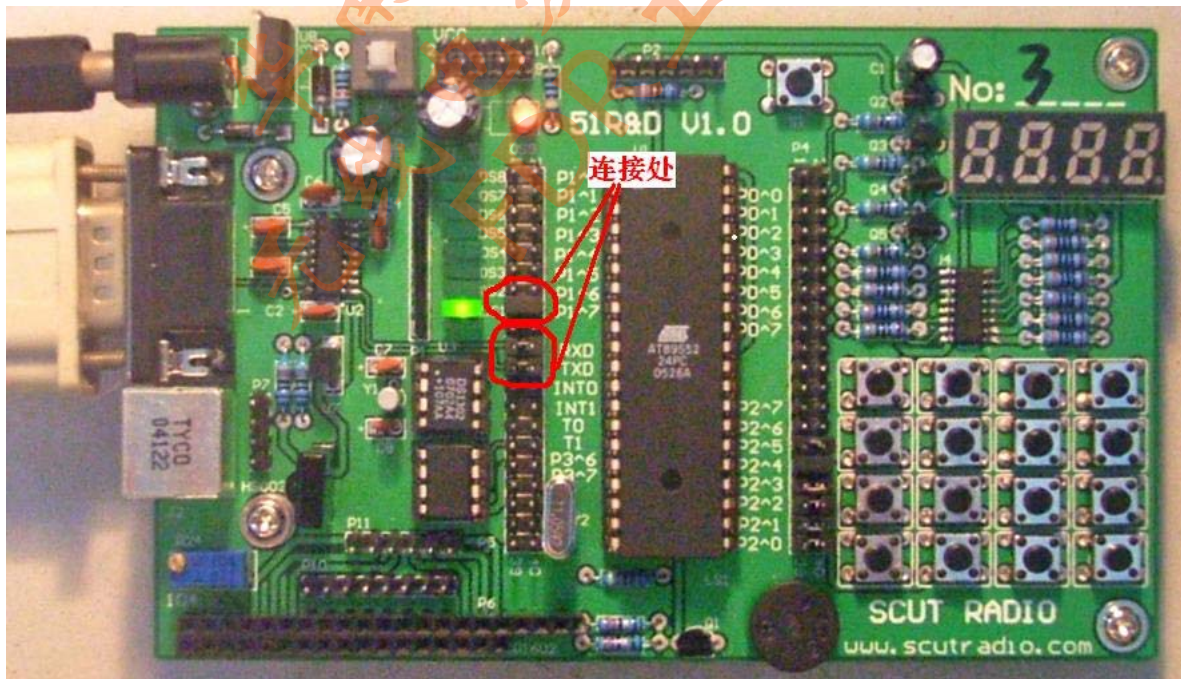
设定好波特率之后我们就可以跟电脑通讯了，只要电脑的串口波特率和单片机的波特率一致。与电脑的通讯我们通过“串口调试助手V2.1.exe”软件来实现，该软件的使用很简单，只要将串口选择‘COM1’波特率设置为‘9600’数据位为8位。打开串口（如果关闭）。然后在发送区里输入要发送的数据，单击手动发送就将数据发送出去了。注意，如果选中‘十六进制发送’那么发送的数据是十六进制的，必须输入两位数据。如果没有选中，则发送的是ASCLL码，那么单片机控制的数码管将显示ASCLL码值。此次实验串口调试助手的设置如下图：



本次实验我们通过串口调试助手发送两个数据到单片机，单片机得到数据之后对这两个数据进行求和，把结果再返回给串口调试助手，在电脑上显示出来，同时每完成一次计算单片机的P1.7脚控制LED一次状态翻转。用来指示单片机运行与否，方便观察。

注意：发送时每个数据必须以回车键结束。

三,硬件连接:



硬件连线图

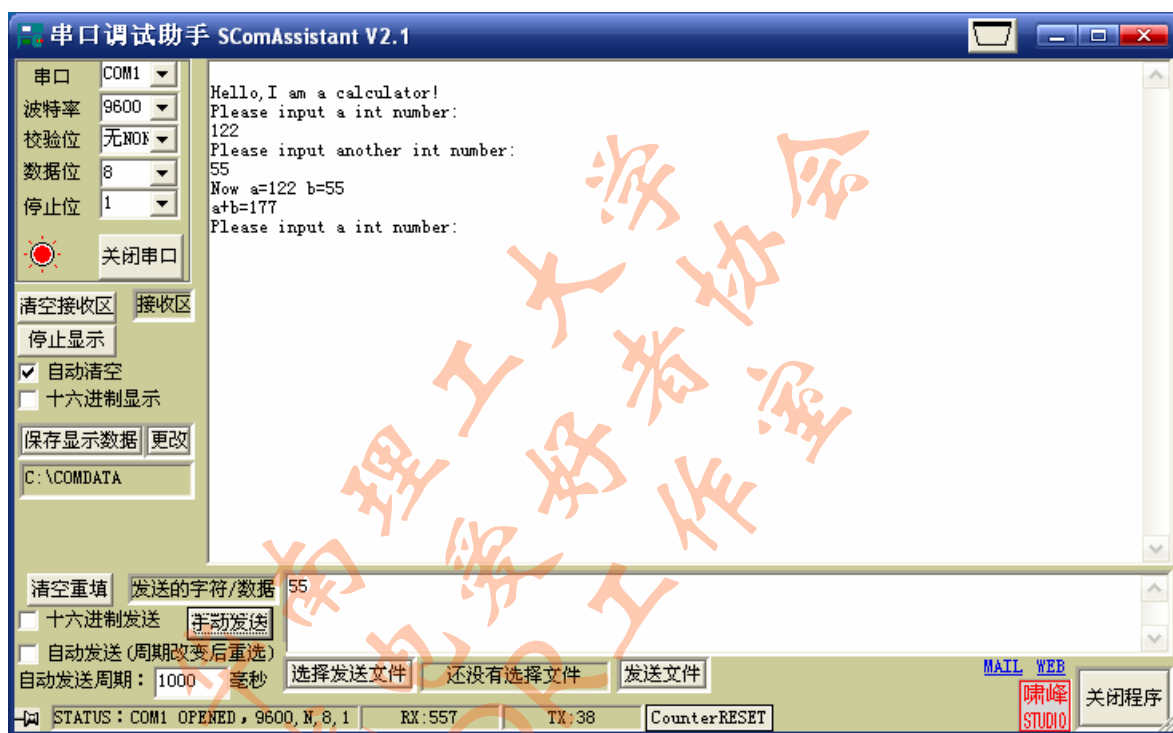
四,实验代码:

```
#include <reg51.h>
```

```
#include<stdio.h>
#define uchar unsigned char
#define uint unsigned int
#define LEDPORT P0
sbit LED=P1^7; //指示灯
uchar mark=0;
/*-----time ms延时函数-----*/
void delay_ms(uint time)
{
    uchar tres;
    for(;time>0;time--)
    {
        tres=150;
        while(tres--);
    }
}
void Inituart()
{
    SCON=0X50; //串口:方式1,接受允许
    TMOD=0X20; //定时器1:模式2,8位自动重装模式,用于产生波特率
    TH1=0XFD; //初值设定 9600bit
    TL1=TH1;
    TR1=1; //开启定时器1
    TI=1; //特别注意:强行使上次数据发送结束,才开始printf (不信可以试试去掉本句试试)
}
void main()
{
    uint a=12,b=22;
    Inituart();
    printf ("%s\n","Hello,I am a calculator!"); //发送
    for(a=0;a<20;a++)//LED闪烁10次,指示单片机已经准备好了
    {
        LED=!LED;
        delay_ms(100);
    }
    LED=1; //关掉指示灯
    while(1)
    {
        printf ("%s\n","Please input a int number:"); //发送提示语
        scanf("%d",&a); //接收第一个数
        delay_ms(500);
        printf ("%s","Please input another int number:"); //发送提示语
        scanf("%d",&b); //接收第二个数
        delay_ms(500);
    }
}
```

```
printf ("Now a=%d",a); //将得到的数在电脑上显示出来
printf (" b=%d\n",b);
delay_ms(500);
printf ("a+b=%d\n",a+b); //将结果返回给电脑
delay_ms(500);
LED=!LED; //每运算完一次LED状态就会改变一次
}
}
```

按硬件连接所示,把硬件连接好(红色框内部分),把程序编译之后,下载到单片机后,就可以通过串口调试助手让单片机执行两个无符号整数的加法计算了,如下图所示:



扩展:

通过串口调试助手,写一个基于单片机的无符号整形数据计算器,熟悉计算器的基本工作原理,同时巩固对串口的理解.