

# 深圳大学实验报告

课程名称： 微型计算机技术

实验项目名称： 串口的使用-单字节数据的发送与接收

学院： 医学院

专业： 生物医学工程

指导教师： 徐海华、刘昕宇

报告人： 陈焕鑫 学号： 2016222042 班级： 生工2班

实验时间： 2018-12-4

实验报告提交时间： 2018-12-18

教务处制

## 一、实验目的

- 1、RS-232 串口是单片机中最重要的功能之一，大部分通信都是依靠串口实现的；
- 2、掌握如何设置串口工作方式、波特率等内容；
- 3、掌握通过查询标志位方式、中断方式发送和接收数据的方法

## 二、实验仪器

微机原理实验板

## 三、实验内容

### 1、编写程序，实现以下功能

首先设置串口 1, 使用 Timer1 作为串口 1 的波特率发生器, 并且将波特率设置为 9600 bps

(1) 使用在主程序内**循环查询 TI + 延时**的方式, 约每 0.5 秒向上位机发送一个字节。同时用 STC 的串口助手观察单片机发上去的数据, 看看是否和程序中设置的一样。

(2) 使用在主程序内**循环查询 RI**的方式, 接收 PC 发下来的单个字节, 并且发回一个同样的数据给 PC, 通过串口助手观察 MCU 发回的数据, 是否与程序一致。

(3) 使用串口 1 **中断方式**, 在中断内接收 PC 发下来的单个字节, 并且在接收到后, 返回一个同样的数据, 并通过串口助手观察 MCU 发回的数据, 是否与程序一致。

## 四、实验原理

### 1、STC 串口使用方法

```
REN  = 1;           //允许接收控制位置 1
SM0  = 0;
SM1  = 1;           //UART 工作方式
TR1  = 1;           //启动定时器 1
TMOD = 0x20;        //定时器 1 的方式
TH1  = 0xfd;        //设置波特率
TL1  = 0xfd;
```

### 2、C51 串口中断的使用

1) 串行口的中断源有两个，一个是发送中断(TI)，一个是接收中断(RI)。但是共用一个中断入口，所以进入串行口中断之后，需要先对中断源进行判断。

2) 51 单片机没有 FIFO 缓冲区，所以发送（接收）一个字节就会产生一次中断，所以接收的时候需要注意，需要在下一个字节到来之前将前一个数据从 SBUF 中读取出来，否则就会出现丢失数据的情况。

### 3) 范例

```
void Uart1_ISR() interrupt 4
```

```
{
    if(TI==1){           // 如果 TI=1,则说明中断是由发送完成引发的
        TI=0;           // 手动清零 TI 标志位
        __Uart1_Send_ISR(); // 调用串口 1 发送完成处理函数
    }
    else if(RI==1){      // 如果 RI=1,则说明中断是由接收完成引发的
        RI=0;
        __Uart1_Recv_ISR();
    }
}
```

## 五、实验方法及步骤

首先，打开 Keil 软件，新建工程名为 Lab9Prj，在工程中添加空的 main.c 文件。编辑 main.c，在 main.c 中添加头文件 STC12C5A60S2.h，该头文件中包含了对 51 单片机引脚和寄存器的各种宏定义，只有包含了该头文件，我们才能在.c 文件中直接使用各个寄存器的简称。

(1)(2) 题都可以通过 while 循环来不断查询标志位 TI 和 RI 的内容，只要这些标志位的内容发送变化就执行相应的操作。(3) 题使用串口中断，不管是发送还是接收数据，都可以触发中断，然后可以在中断中判断标志位 TI 和 RI 的情况来判断是哪个需要发送数据还是接收数据，进行相应的操作。

具体程序代码如下：

(1) 使用在主程序内循环查询 TI + 延时的方式，约每 0.5 秒向上位机发送一个字节。

```
#include "STC12C5A60S2.h"

typedef unsigned char uint8; //定义 uint8 为无符号 8 位数据类型

void Delay_ms(int DelayTimes); //延时函数
void Uart1_Init(); //初始化串口函数

void main()
{
    Uart1_Init(); //初始化串口

    while(1)
    {
        SBUF = 0x5A; //串口发送 0x5A

        while(TI == 0) //等待发送结束时，TI 标志位自动置 1
        {
            TI = 0; //将标志位清零
        }

        Delay_ms(500); //延时 500ms
    }
}

void Delay_ms(int DelayTimes)
{
    int cnt;

    //循环直至 DelayTimes 递减为 0
    for(cnt = 1000; DelayTimes > 0; DelayTimes--)
```

```

{
    for(; cnt > 0; cnt--); //循环直至 cnt 递减为 0
}
}

void Uart1_Init()
{
    REN = 1;        //允许接收控制位置 1
    SM0 = 0;        //设置 UART 工作方式
    SM1 = 1;
    TR1 = 1;        //启动定时器
    TMOD = 0x20;    //设置定时器的方式
    TH1 = 0xfd;     //设置波特率
    TL1 = 0xfd;
}

```

(2) 使用在主程序内**循环查询 RI** 的方式，接收 PC 发下来的单个字节，并且发回一个变化后的数据给 PC。

```

#include "STC12C5A60S2.h"

typedef unsigned char uint8; //定义 uint8 为无符号 8 位数据类型

void Uart1_Init();          //初始化串口函数

void main()
{
    uint8 Uart_Tmp;          //收发缓存器
    Uart1_Init();            //初始化串口

    while(1)
    {
        while(RI == 0)      //循环等待接收到字符
        {
        }

        RI = 0;              //将标志位清零
        Uart_Tmp = SBUF;     //接收串口接收到的数据
        SBUF = Uart_Tmp + 5; //将数据加上 5 后再次发送

        while(TI == 0)      //等待发送结束
        {
        }

        TI = 0;              //将标志位清零
    }
}

```

```

}

void Uart1_Init()
{
    REN = 1;    //允许接收控制位置 1
    SM0 = 0;    //设置 UART 工作方式
    SM1 = 1;
    TR1 = 1;    //启动定时器
    TMOD = 0x20; //设置定时器方式
    TH1 = 0xfd; //设置波特率
    TL1 = 0xfd;
}

```

(3) 使用串口 1 **中断方式**，在中断内接收 PC 发下来的单个字节，并且在接收到后，返回一个变化后的数据。

```

//Uart.h
#ifndef __UART_H_
#define __UART_H_

typedef unsigned char uint8; //定义 uint8 为无符号 8 位数据类型

void Uart1_Init();           //初始化串口函数

#endif

//Uart.c
#include "STC12C5A60S2.h"
#include "Uart.h"

static void __Uart1_Send_ISR(); //串口发送中断服务函数
static void __Uart1_Recv_ISR(); //串口接收中断服务函数

void Uart1_ISR() interrupt 4
{
    if(TI == 1)                //需要发送字符
    {
        TI = 0;                //将发送标志位清零
        __Uart1_Send_ISR();    //调用发送中断服务函数
    }
    else if(RI == 1)           //接收到字符
    {
        RI = 0;                //将接收标志位清零
        __Uart1_Recv_ISR();    //调用结束中断服务函数
    }
}

```

```

    }
}

static void __Uart1_Send_ISR()
{

}

static void __Uart1_Recv_ISR()
{
    uint8 Uart_Tmp;    //收发缓冲器

    Uart_Tmp = SBUF;    //获取接收到的字符
    SBUF = Uart_Tmp + 5; //将字符加上 5 后再次发送
}

void Uart1_Init()
{
    REN = 1;    //允许接收控制位置 1
    SM0 = 0;    //设置 UART 工作方式
    SM1 = 1;
    TR1 = 1;    //启动定时器
    TMOD = 0x20; //设置定时器方式
    TH1 = 0xfd; //设置波特率
    TL1 = 0xfd;
    ES = 1;    //开启串口中断
    PS = 0;    //设置串口中断优先级
    EA = 1;    //开启 CPU 中断总开关
}

//main.c
#include "STC12C5A60S2.h"
#include "Uart.h"

void main()
{
    Uart1_Init(); //初始化串口
    while(1);    //循环等待中断
}

```

检查代码无误之后，分别编译、链接、生成 Hex 文件，将 Hex 文件通过串口烧进实验平台中，观察实验现象。

## 六、实验现象

烧进程序之后，可以观察到实验现象：

(1) 的实验现象如图 1-1 所示，串口不断地向 PC 发送 0x5A。

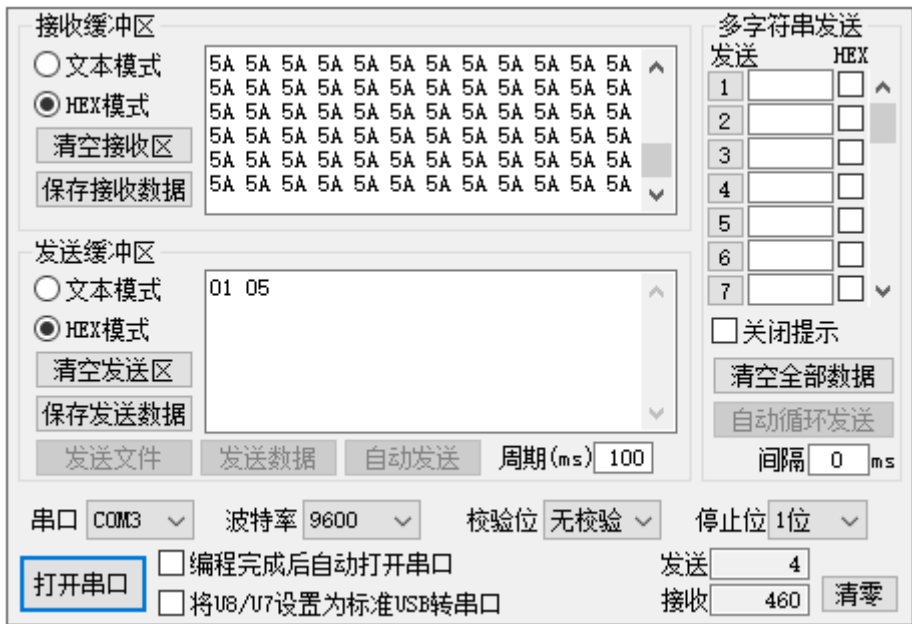


图 1-1 实验(1)的现象

(2) 和 (3) 虽然实现的方法不一样，但是实验现象是一样的。如图 1-2 所示。

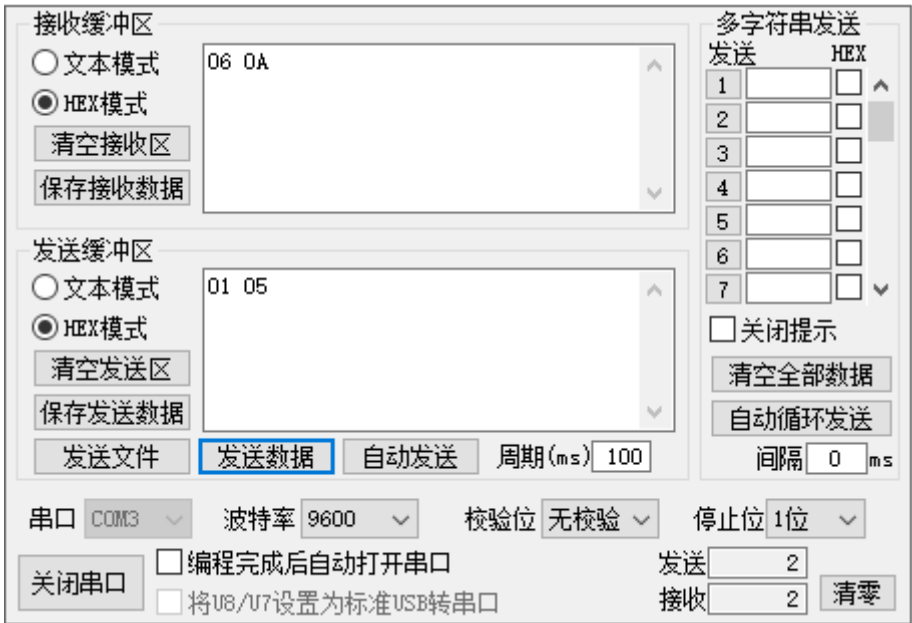


图 1-2 实验(2)和(3)的现象

## 七、实验结论

通过这次实验的学习，我了解了 RS-232 串口的概念，知道了串口的工作原理，对串口有了初步的认识；掌握了如何通过设置设置串口的工作方式、波特率等内容；能够通过查询标志位的方式、中断的方式接受和发送数据。



指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。  
2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。