

单片机通信技术应用实例

——基于STC15W系列单片机

周长锁 王 旭 ■ 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

单片机通信技术应用实例 ——基于 STC15W 系列单片机

周长锁 王 旭 编著



電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书通过实例展现单片机多种通信模式的实现方法，以单片机的串口和 SPI 接口为主线，详细讲解 STC15W 系列单片机在串口通信和网络通信中的应用，以及在蓝牙、WiFi、GPRS 等通信技术中的应用。参考本书的实例，能比较容易地入手单片机通信类产品的设计和开发。

本书由工控开发人员编写，每章的实例都有详细的电路原理图和 C 程序源代码，实例来源于实际应用项目和技术储备测试，其中与上位机通信的实例提供了 VB6 程序源代码，与手机通信的实例提供了 Android 程序源代码，使读者能系统理解单片机的通信过程。

本书适合具有一定单片机技术基础的电子爱好者和电子产品开发人员阅读参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

单片机通信技术应用实例: 基于 STC15W 系列单片机/周长锁, 王旭编著. —北京: 电子工业出版社, 2018.6
ISBN 978-7-121-34343-8

I. ①单… II. ①周… ②王… III. ①单片微型计算机 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2018) 第 115591 号

策划编辑: 陈韦凯

责任编辑: 康 霞

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 14 字数: 358 千字

版 次: 2018 年 6 月第 1 版

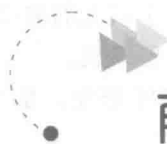
印 次: 2018 年 6 月第 1 次印刷

印 数: 2 000 册 定价: 65.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: chenwk@phei.com.cn, (010) 88254441。



前 言

随着信息化的发展和物联网概念的提出，单片机由原来主要解决产品的智能化问题，转变为现在还要解决产品的网络化问题，而要实现网络化，就要根据具体情况采用合适的通信技术。

掌握单片机通信技术是电子产品设计开发人员的基本要求，要想设计出好的产品，还需要对应用该产品的行业有较深入的理解，清楚产品的现状与不足、发展趋势和功能需求等信息，通过行业的专业知识与单片机技术的结合，才能更好地应用单片机的通信技术。

STC15W 系列单片机是抗干扰性能优良的国产单片机，其中 STC15W4K 系列有 4 个串口，适合工控产品使用，尤其适合通信协议转换方面的产品使用。同时，STC15W4K 系列单片机有容量为 4KB 的 RAM，远超同类 8 位单片机几百字节 RAM 的配置，达到了 ARM 处理器的水平，能替代 ARM 处理器在网络通信中的应用，性价比高，开发出的产品也具有竞争力。

本书共分 9 章，其中第 1~6 章介绍单片机串口通信编程技巧和串口扩展蓝牙、WiFi、GPRS 通信接口中的应用；第 7~8 章介绍单片机 SPI 接口及其扩展为网络接口和 CAN 接口的应用；第 9 章介绍单片机模拟 I²C 总线的应用。各章节的具体内容安排如下。

第 1 章介绍单片机串口通信基本知识和编程技巧。实例内容为简易读卡开锁电路设计，示范了射频读卡模块（RFID 技术）的应用。

第 2 章介绍电话线路来电显示识别技术，包括 FSK 和 DTMF 两种来电显示解码电路与单片机的接口技术。实例内容为电话来电显示装置设计，通过该实例学习单片机自编通信协议与上位机的通信数据处理过程，实例提供上位机 VB6 程序源码。

第 3 章介绍蓝牙模块的应用。实例内容为手机蓝牙接口示波器设计，通过该实例学习单片机通过蓝牙与手机的通信数据处理过程，实例提供手机 Android 程序源代码。

第 4 章介绍单片机串口转 RS485 总线通信的应用。实例内容为 RS485 接口温度传感器设计，讲解了 Modbus-RTU 通信规约的实现方法。

第 5 章介绍 GPRS 模块的应用。实例 1 是用单片机控制 GTM900B 收发短信，学习用短信传输数据和实现远程控制。实例 2 是用车辆 GPS 定位及微信远程控制装置设计，讲解如何通过“贝壳物联”实现手机微信对车辆定位信息的读取及对车辆电路的控制。

第 6 章介绍 WiFi 模块的应用。实例内容是 WiFi 遥控小车电路设计，实例提供手机 Android 程序源码。

第 7 章介绍单片机 SPI 接口及其扩展为网络接口和 CAN 接口的应用。实例 1 是基于 W5500 的串口服务器设计；实例 2 是 USB 转 CAN 调试工具设计。

第 8 章介绍以太网 Modbus-TCP 协议实现方法。实例通过电度表集中抄表装置设计，讲解了电度表 DL/T645 通信协议和一种集中抄表解决方案。

第 9 章介绍单片机模拟 I²C 总线和模拟韦根协议的应用，讲解红外温度传感器、实时时

钟、OLED 显示屏的 I²C 总线通信过程。

为方便读者测试学习，本书提供实例 C 程序和上位机、手机上的测试程序下载，读者可以登录 www.hxedu.com.cn（华信教育资源网）查找本书后免费下载。

由于编著者理论知识有限，书中的错误和不妥之处在所难免，殷切期望广大读者给予指正。

编著者

读者调查及征稿

1. 您觉得这本书怎么样？有什么不足？还能有什么改进？

2. 您在什么行业？从事什么工作？需要哪些方面的图书？

3. 您有无写作意向？愿意编写哪方面的图书？

4. 其他：

说明：

针对以上调查项目，可通过电子邮件直接联系：bjcwk@163.com 联系人：陈编辑

欢迎您的反馈和投稿！

电子工业出版社

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为，歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

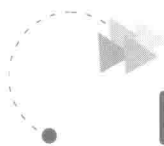
传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市海淀区万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036



目 录

第 1 章	STC15W 系列单片机串口通信	(1)
1.1	单片机串口相关寄存器的设置	(1)
1.1.1	主要串口通信参数	(1)
1.1.2	STC15W 系列单片机串口特点	(2)
1.1.3	单片机 STC15W201S 串口相关寄存器设置	(3)
1.2	串口中断发送与接收的 C 程序	(4)
1.2.1	不用串口中断发送字符串的 C 程序	(4)
1.2.2	不正确的中断发送字符串 C 程序	(4)
1.2.3	正确的串口中断发送与接收 C 程序	(5)
1.3	上位机串口调试软件使用	(9)
1.3.1	常见 USB 转串口集成电路简介	(9)
1.3.2	串口调试软件使用方法	(9)
1.4	简易读卡开锁电路设计	(10)
1.4.1	两种射频读卡模块测试	(11)
1.4.2	简易读卡开锁电路原理图	(12)
1.4.3	读卡开锁的 C 程序	(13)
第 2 章	电话来电显示装置设计	(17)
2.1	来电显示解码集成电路	(17)
2.1.1	DTMF 解码集成电路 HM9270D	(17)
2.1.2	FSK 解码集成电路 HT9032D	(18)
2.2	电话来电显示装置电路原理	(20)
2.2.1	信号隔离	(20)
2.2.2	电话线路状态判断	(20)
2.2.3	DTMF 解码	(22)
2.2.4	FSK 解码	(22)
2.2.5	USB 转串口	(23)
2.3	串口通信协议设定	(23)
2.3.1	通信协议实现功能	(23)
2.3.2	电话来电显示装置的 C 程序	(24)
2.3.3	上位机 VB 示例程序	(30)

第 3 章 手机蓝牙接口示波器	(39)
3.1 单片机串口转蓝牙通信	(39)
3.1.1 常用蓝牙模块介绍	(39)
3.1.2 蓝牙模块参数设置	(39)
3.1.3 蓝牙模块与手机、台式机连接方法	(43)
3.2 手机蓝牙接口示波器电路原理	(43)
3.2.1 信号采集	(45)
3.2.2 数据转发	(45)
3.3 手机蓝牙接口示波器软件编程	(45)
3.3.1 通信协议	(45)
3.3.2 单片机 C 程序	(46)
3.3.3 手机 Android 程序	(51)
第 4 章 RS485 接口温度传感器	(60)
4.1 温度传感器电路原理	(60)
4.1.1 常用数字测温元件	(60)
4.1.2 电路原理说明	(61)
4.2 单片机串口转 RS485	(63)
4.2.1 RS485 总线通信特点及现场敷设注意事项	(63)
4.2.2 RS485 集成电路与单片机接口	(63)
4.3 单片机实现 Modbus 协议	(65)
4.3.1 Modbus/RTU 协议格式	(65)
4.3.2 CRC 校验	(66)
4.4 温度传感器源代码及说明	(66)
4.4.1 温度传感器 C 程序	(66)
4.4.2 C 程序中关键点说明	(74)
4.4.3 用 Modbus 协议软件测试温度传感器	(76)
4.4.4 用触屏连接温度传感器	(78)
第 5 章 GPRS 模块远程数据传输	(80)
5.1 基于 GTM900B 的短信数据传输	(80)
5.1.1 华为 GTM900B 模块硬件接口	(80)
5.1.2 常用 AT 指令	(82)
5.1.3 用单片机控制 GTM900B 收发短信	(83)
5.2 基于 SIM900A 的 GPRS 数据传输	(93)
5.2.1 SIM900A 模块硬件接口	(93)
5.2.2 贝壳物联通信协议简介	(94)
5.2.3 车辆 GPS 定位及微信远程控制装置设计	(96)

第 6 章	WiFi 模块遥控与远程控制应用	(109)
6.1	WiFi 模块 ESP8266 简介	(109)
6.1.1	ESP8266 硬件接口	(109)
6.1.2	ESP8266 常用 AT 指令集	(110)
6.1.3	ESP8266 模块和 Android 手机通信测试	(111)
6.2	WiFi 模块 USR-C210 简介	(114)
6.2.1	USR-C210 模块硬件接口	(114)
6.2.2	USR-C210 模块参数设置	(114)
6.2.3	USR-C210 模块通信测试	(116)
6.3	WiFi 遥控小车电路设计	(117)
6.3.1	WiFi 遥控小车电路原理	(117)
6.3.2	WiFi 遥控小车的 C 程序	(118)
6.3.3	WiFi 遥控小车的 Android 程序	(124)
第 7 章	STC15W 单片机 SPI 通信	(129)
7.1	SPI 通信应用	(129)
7.1.1	与 SPI 功能有关的寄存器设置	(129)
7.1.2	SPI 发送数据测试	(130)
7.1.3	SPI 接口 LED 驱动器 MC14489	(132)
7.2	SPI 接口转以太网接口芯片 W5500 的应用	(136)
7.2.1	W5500 简介	(136)
7.2.2	基于 W5500 的串口服务器设计	(139)
7.3	SPI 接口转 CAN 总线应用	(160)
7.3.1	CAN 总线简介	(160)
7.3.2	CAN 控制器 MCP2515	(161)
7.3.3	USB 转 CAN 调试工具设计	(164)
第 8 章	电度表集中抄表装置设计	(181)
8.1	电度表集中抄表装置电路原理	(181)
8.2	电度表 DL/T645 通信规约实现方法	(183)
8.2.1	DL/T645 通信规约简介	(183)
8.2.2	单片机与电度表通信的 C 程序说明	(186)
8.3	以太网 Modbus/TCP 协议实现方法	(189)
8.3.1	以太网控制器 ENC28J60 数据传输	(189)
8.3.2	TCP/IP 软件协议栈	(191)
8.3.3	Modbus/TCP 协议	(193)
8.3.4	电度表集中抄表装置测试	(195)

第 9 章 单片机模拟其他通信接口	(197)
9.1 单片机模拟 I²C 总线	(197)
9.1.1 I ² C 总线简介	(197)
9.1.2 单片机模拟 I ² C 总线的 C 程序	(198)
9.2 几种 I²C 接口器件的通信测试	(201)
9.2.1 红外温度传感器 MLX90614	(201)
9.2.2 实时时钟 DS3231	(203)
9.2.3 OLED 显示屏	(203)
9.3 韦根协议	(209)
9.3.1 韦根协议简介	(209)
9.3.2 单片机模拟韦根 26 协议	(210)
参考文献	(214)

第 1 章 STC15W 系列单片机串口通信

本章讲解如何设置单片机的串口通信参数，如何用串口中断功能提高单片机的运行效率；然后介绍串口调试软件的使用及辅助串口程序的开发调试；最后通过一个小制作，带领读者熟悉串口通信电路的开发过程。

1.1 单片机串口相关寄存器的设置

1.1.1 主要串口通信参数

串口只有在参数一致的情况下才能正常通信，主要的参数有波特率、停止位、校验位，可选的波特率有 1200、2400、4800、9600、19 200、38 400、115 200 等，停止位可选 1 位或 2 位，校验位可选无校验 (n)、偶校验 (e) 和奇校验 (o)。例如，常见的串口通信参数可表示为：9600,n,8,1，其中 9600 代表波特率 9600bps，n 代表无校验，8 代表 8 个数据位，1 代表 1 个停止位。

图 1-1 是不同波特率串口时序对比图，输出同样字符“12”的时序波形一致，区别就是每个数据位的间隔不同，9600bps 每个数据位间隔约 0.104ms，8 个数据位加上前面 1 个起始位和后面 1 个停止位，总周期约 1ms，也就是说，当波特率为 9600 时，每发送 1 个字节约需 1ms；同理，当波特率为 1200 时，每发送 1 个字节约需 8.3ms。

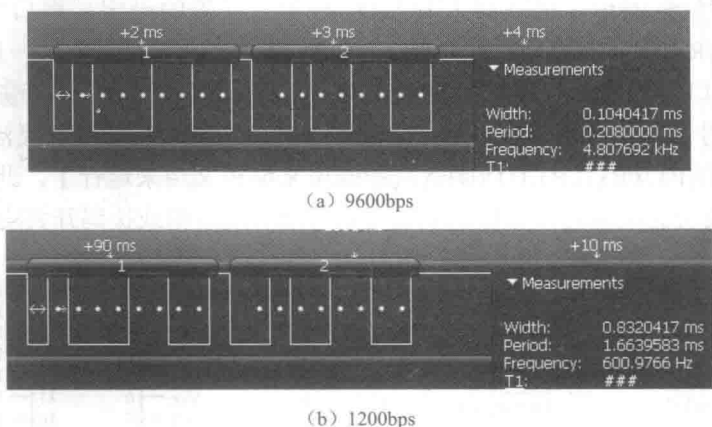
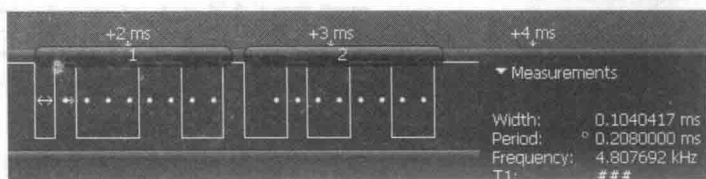


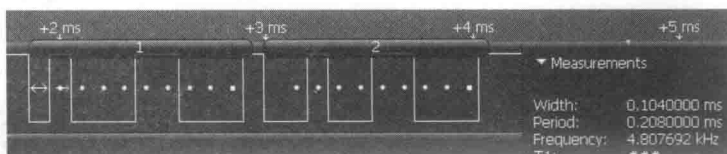
图 1-1 不同波特率串口时序对比图

当使用校验位时，每个字节会含 11 个数据位，不同校验设置时的串口输出波形对比见图 1-2。测试时发出的数据字符是“12”，十六进制数据为 0x31、0x32，先发送 1 位起始位

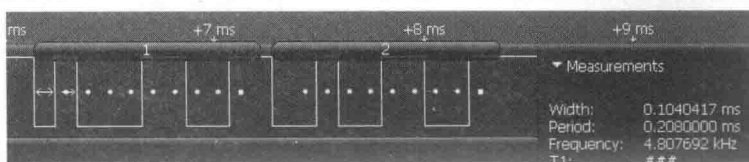
0, 然后是 8 位数字位, 低位在前, 有校验时再发送 1 位校验位, 校验位是根据前面 8 位数据位算出来的, 最后发送停止位 1。



(a) 无校验



(b) 偶校验



(c) 奇校验

图 1-2 不同校验设置时的串口输出波形对比图

1.1.2 STC15W 系列单片机串口特点

1. 串口可在不同引脚间切换

此功能限于引脚数为 16 及以上的单片机。图 1-3 是单片机 STC15W201S 不同封装引脚示意图, SOP16 封装的单片机 STC15W201S 根据寄存器的设定, 串口可以在[P3.0/RxD, P3.1/TxD]和[P3.6/RxD_2, P3.7/TxD_2]间切换, 这样的设计优点有两个: 一是[P3.0/RxD, P3.1/TxD]只作为编程口, 程序运行时切换到[P3.6/RxD_2, P3.7/TxD_2], 这样编程和运行互不影响; 二是可以分时切换, 当两个串口用, 提高了单片机引脚利用效率和灵活性。SOP8 封装的单片机串口引脚[P3.0/RxD, P3.1/TxD]就只能既用来编程又用来运行了, 此时从电路就要考虑外部接线是否会影响编程, 如影响编程可以考虑加跳线帽或拨码开关实现切换。

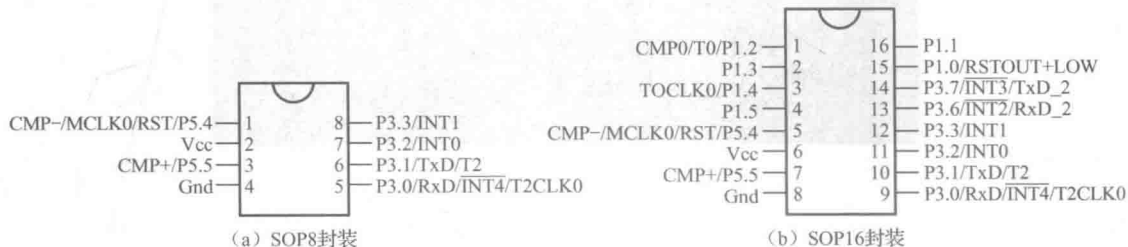


图 1-3 单片机 STC15W201S 不同封装引脚示意图

2. 串口1可设置为中继广播方式

串口1的RxD端收到数据能同步输出到TxD端，这种功能可以用于一对多通信；但是由于主机只能发送数据，所以这个功能不常用。

3. STC15W4K系列单片机有4个串口

单片机外围模块大多都是通过串口与单片机进行通信的。当某个项目需要多个串口时，普通单片机只能通过扩展电路的方式来增加串口，如CH432系列单片机就是利用SPI接口实现串口扩展的。而STC15W4K系列单片机本身支持4个串口，与扩展串口相比，在满足单片机与多外围模块之间通信要求的同时，也简化了程序代码。

4. 奇偶校验位不能自动添加

这是STC15W系列单片机的一个缺点。一般单片机使能奇偶校验后，校验位是自动生成、自动添加的。STC15W系列单片机的奇偶校验位能自动生成到寄存器PSW，发送数据时需要编程添加到SCON寄存器，具体使用方法见以后的代码。实际使用中尽量避免使用奇偶校验，一般多在通信协议中设置校验字节来保证数据的正确性。

1.1.3 单片机STC15W201S串口相关寄存器设置

STC单片机的程序下载软件的附加功能非常实用，有定时器、波特率计算功能，还有封装引脚位和范例程序等功能，方便使用。图1-4就是计算串口波特率的界面截图，选好参数后，单击“生成C代码”按钮即可自动生成串口初始化代码。

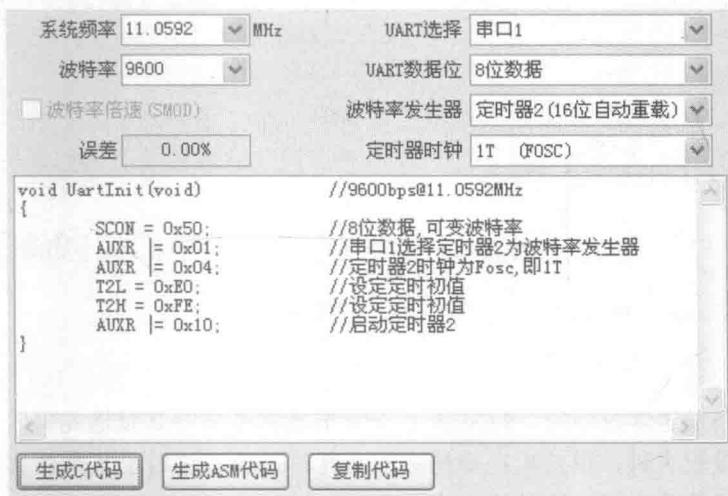


图 1-4 计算串口波特率的界面截图

使用串口通信时，单片机的系统频率最好使用 11.0592MHz 或其整数倍值，如 22.1184MHz，此时波特率最为准确，误差理论上为零。当选为 12MHz 时误差为 0.16%，感觉上不是很好，实际上也不影响通信准确性，一般认为误差在 4.5%以内是允许的。

STC15W 系列单片机的串口波特率发生器一般都占用定时器 2；对于 STC15W4K 系列 4 串口单片机，4 个串口可以都使用定时器 2，但此时波特率必须一致，如果波特率不一致就必须使用不同的定时器，串口 1 可以用定时器 1，串口 3 可以用定时器 3，串口 4 可以用定时器 4。

定时器初值计算公式如下：

$$T2H = (\text{unsigned char})((65536UL - (\text{MAIN_Fosc} / 4) / \text{Baudratel}) / 256);$$

$$T2L = (\text{unsigned char})((65536UL - (\text{MAIN_Fosc} / 4) / \text{Baudratel}) \% 256);$$

当系统频率 MAIN_Fosc=11 059 200Hz，波特率 Baudratel=9600 时，T2H=0xFE，T2L=0xE0。

1.2 串口中断发送与接收的 C 程序

在一些单片机资料的示范程序中，有用串口中断和不用串口中断的，但多数串口中断功能使用得不对，没有充分利用中断的作用。

1.2.1 不用串口中断发送字符串的 C 程序

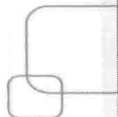
程序源代码如下：

```
//发送串口数据
void SendData(BYTE dat)
{
    SBUF = dat;                //写数据到 UART 数据寄存器
    while(!TI);              //等发送完成
    TI=0;                      //清除完成标志
}
//发送字符串
void SendString(char *s)
{
    while (*s)                //检测字符串结束标志
    {
        SendData(*s++);      //发送当前字符
    }
}
```

不用串口中断发送字符串时，每发送 1 个字节数据都要等其发送完再发送下一个字节数据，当发送数据量较大时，单片机在等待时间上耗时过长，这对于复杂的程序是不允许的，单片机还有很多任务要执行，这时就要使用串口中断功能。

1.2.2 不正确的中断发送字符串 C 程序

程序源代码如下：



```

//串口中断程序
void Uart() interrupt 4           //中断号 4
{
    if (RI)
    {
        RI = 0;                 //清除 RI 位
                                //接收数据处理
    }
    if (TI)
    {
        TI = 0;                 //清除 TI 位
        busy = 0;               //清除完成标志
    }
}
//发送串口数据
void SendData(BYTE dat)
{
    while (busy);               //等待前面的数据发送完成
    busy = 1;
    SBUF = dat;                 //写数据到 UART 数据寄存器
}
//发送字符串
void SendString(char *s)
{
    while (*s)                   //检测字符串结束标志
    {
        SendData(*s++);         //发送当前字符
    }
}

```

以上代码虽然使用了串口中断功能，但只是用中断功能清除了标志位。总体来看，无论程序是在发送字符串时，还是阻塞在发送和等待上，CPU 都无法执行别的任务。

1.2.3 正确的串口中断发送与接收 C 程序

程序源代码如下：

```

//串口测试程序
#include "STC15Wxx.h"           //头文件
#include <string.h>              //函数库
#define MAIN_Fosc 11059200L     //定义主时钟
#define T1MS (65536-MAIN_Fosc/1000) //1T 模式
//通信
unsigned char tbuf1[30];        //UART1 数据缓冲区
unsigned char rbuf1[30];        //UART1 数据缓冲区
bit rnew1;                      //接收新数据完成标志
bit ring1;                      //正在接收新数据标志

```



```

unsigned char rnl;           //接收数据字节数
unsigned char snl;          //发送数据总字节数
unsigned char spl;          //已发送数据字节数
unsigned char tl;           //通信计时
unsigned int t0;            //计时
//=====
//函数: GPIO_Init()
//说明: 初始化端口
//PxM1.n,PxM0.n =00--->Standard, 01--->push-pull
//              =10--->pure input, 11--->open drain
//=====
void GPIO_Init (void)
{
    P3M1 = 0x00;   P3M0 = 0x00;           //设置为准双向口
    P5M1 = 0x00;   P5M0 = 0x00;           //
}
//=====
//函数: Timer_Uart_Init()
//说明: 设置 Timer2 做波特率发生器,Timer0 做 1ms 定时器
//=====
void Timer_Uart_Init(void)
{
    //定时器 0 定时中断
    AUXR = 0xC5;           //定时器 0 为 1T 模式
    TMOD = 0x00;           //设置定时器为模式 0 (16 位自动重载)
    TLO = T1MS;           //初始化计时值
    TH0 = T1MS >> 8;
    TR0 = 1;               //定时器 0 开始计时
    ET0 = 1;               //使能定时器 0 中断
    //定时器 2 产生波特率 9600
    SCON = 0x50;           //8 位数据, 可变波特率
    T2L = 0xE0;            //设定定时初值
    T2H = 0xFE;            //设定定时初值
    ES = 1;                //允许中断
    REN = 1;               //允许接收
    P_SW1 &= 0x3f;
    P_SW1 |= 0x00;         //0x00: P3.0 P3.1; 0x40: P3.6 P3.7
    AUXR |= 0x10;         //启动定时器 2
}
//=====
//函数: void SendString(char *s)
//说明: 发送字符串子程序
//=====
void SendString(char *s)
{
    unsigned char char_length,j;

```