

慧净电子 5 星好评资料包 赠送给各位好人，好人一生平 慧净单片机淘宝网：
<http://shop37031453.taobao.com/>

串行口通信设计

系 别： 电气工程系
班 级： 过控 1001
姓 名： 韦德 格里芬
学 号： 0402100122
 0402100106
指导老师： 王老师

目 录

| | |
|----------------------------------|----|
| 摘 要 | 3 |
| 第一章：单片机简介 | 4 |
| 1.1 单片机定义 | 4 |
| 1.2 芯片简介 | 4 |
| 1.2.1 AT89C51 单片机 | 4 |
| 1.2.2 MAX 232 简介 | 8 |
| 第二章：串行口通信原理 | 11 |
| 2.1 双向串行口通信方框图 | 11 |
| 2.1.1 通讯原理 | 11 |
| 2.1.2 串行口通讯接口 | 12 |
| 2.2 程序下载 | 14 |
| 第三章：软件部分设计 | 16 |
| 3.1 PROTEUS 介绍 | 16 |
| 3.2 程序设计 | 17 |
| 3.2.1 课程设计的程序要求 | 17 |
| 3.2.2 流程图 | 18 |
| 3.2.3 程序 | 18 |
| 第四章：调试与仿真 | 19 |
| 4.1 调试助手设置 | 19 |
| 4.1.1 串行口用于 ASCII 码字符的传输设置 | 19 |
| 4.2 串行口设置 | 20 |
| 4.2.1 通常设置 | 20 |
| 4.2.2 调试流程图 | 22 |
| 4.2.3 串行口系列选择 | 23 |
| 设计总结 | 25 |
| 参考文献 | 27 |
| 附录 | 28 |

摘 要

计算机与计算机或计算机与终端之间的数据传送可以采用串行通讯和并行通讯二种方式。由于串行通讯方式具有使用线路少、成本低，特别是在远程传输时，避免了多条线路特性的不一致而被广泛采用。在串行通讯时，要求通讯双方都采用一个标准接口，使不同的设备可以方便地连接起来进行通讯。RS-232-C 接口又称 EIA RS-232-C 是目前最常用的一种串行通讯接口。它是在 1970 年由美国电子工业协会 EIA 联合贝尔系统、调制解调器厂家及计算机终端生产厂家共同制定的用于串行通讯的标准。它的全名是“数据终端设备 DTE 和数据通讯设备 DCE 之间 串行二进制数据交换接口技术标准”该标准规定采用一个 25 个脚的 DB25 连接器，对连接器的每个引脚的信号内容加以规定，还对各种信号的电平加以规定。

随着计算机技术尤其是单片微型机技术的发展，人们已越来越多地采用单片机来对一些工业控制系统中如温度、流量和压力等参数进行检测和控制。PC 机具有强大的监控和管理功能，而单片机则具有快速及灵活的控制特点，通过 PC 机的 RS-232 串行接口与外部设备进行通信，是许多测控系统中常用的一种通信解决方案。因此如何实现 PC 机与单片机之间的通讯具有非常重要的现实意义。

第一章:单片机简介

1.1 单片机定义

单片机微型计算机是微型计算机的一个重要分支,也是颇具生命力的机种。单片机微型计算机简称单片机,特别适用于控制领域,故又称为微控制器。

通常,单片机由单块集成电路芯片构成,内部包含有计算机的基本功能部件:中央处理器、存储器和 I/O 接口电路等。因此,单片机只需要和适当的软件及外部设备相结合,便可成为一个单片机控制系统。

单片机经过 1、2、3、3 代的发展,目前单片机正朝着高性能和多品种方向发展,它们的 CPU 功能在增强,内部资源在增多,引脚的多功能化,以及低电压低功耗。

1.2 芯片简介

本章主要介绍 AT89C51 单片机和 MAX232 芯片。

1.2.1 AT89C51 单片机

单片机 AT89C51 是一种带 4K 字节闪存可编程可擦除只读存储器 (FPEROM—FlashProgrammable and Erasable Read Only Memory) 的低电压、高性能 CMOS 8 位微处理器,俗称单片机。AT89C2051 是一种带 2K 字节闪存可编程可擦除只读存储器的单片机。单片机的可擦除只读存储器可以反复擦除 1000 次。该器件采用 ATMEL 高密度非易失存储器制造技术制造,与工业标准的 MCS-51 指令集和输出管脚相兼容。由于将多功能 8 位 CPU 和闪烁存储器组合在单个芯片中,ATMEL 的 AT89C51 是一种高效微控制器,AT89C2051 是它的一

种精简版本。AT89C 单片机为很多嵌入式控制系统提供了一种灵活性

高且价廉的方案。外形及引脚排列如（图 1.1）所示。

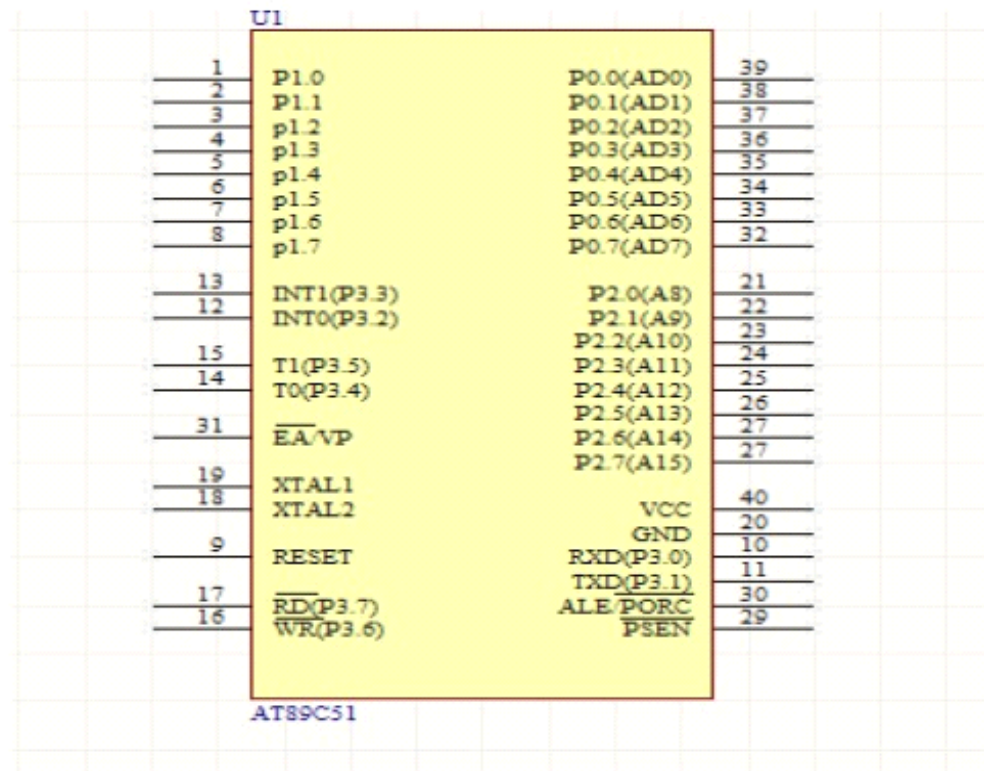


图 1.1 AT89C2

一. 主要特性：

- 与 MCS-51 兼容
- 4K 字节可编程闪烁存储器
- 寿命：1000 写/擦循环
- 数据保留时间：10 年
- 全静态工作：0Hz-24MHz
- 三级程序存储器锁定
- 128×8 位内部 RAM
- 32 可编程 I/O 线
- 两个 16 位定时器/计数器
- 5 个中断源

- 可编程串行通道
- 低功耗的闲置和掉电模式
- 片内振荡器和时钟电路

二：管脚说明

VCC：供电电压。

GND：接地。

P0 口：P0 口为一个 8 位漏级开路双向 I/O 口，每脚可吸收 8TTL 门电流。当 P1 口的管脚第一次写 1 时，被定义为高阻输入。P0 能够用于外部程序数据存储器，它可以被定义为数据/地址的第八位。在 FIASH 编程时，P0 口作为原码输入口，当 FIASH 进行校验时，P0 输出原码，此时 P0 外部必须被拉高。

P1 口：P1 口是一个内部提供上拉电阻的 8 位双向 I/O 口，口缓冲器能接收输出 4TTL 门电流。P1 口管脚写入 1 后，被内部上拉为高，可用作输入，P1 口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。在 FLASH 编程和校验时，P1 口作为第八位地址接收。

P2 口：P2 口为一个内部上拉电阻的 8 位双向 I/O 口，P2 口缓冲器可接收，输出 4 个 TTL 门电流，当 P2 口被写“1”时，其管脚被内部上拉电阻拉高，且作为输入。并因此作为输入时，P2 口的管脚被外部拉低，将输出电流。这是由于内部上拉的缘故。P2 口当用于外部程序存储器或 16 位地址外部数据存储器进行存取时，P2 口输出地址的高八位。在给出地址“1”时，它利用内部上拉优势，当对外部八位地址数据存储器进行读写时，P2 口输出其特殊功能寄存器的内容。P2 口在 FLASH 编程和校验时接收高八位地址信号和控制信号。

P3 口：P3 口管脚是 8 个带内部上拉电阻的双向 I/O 口，可接

收输出 4 个 TTL 门电流。当 P3 口写入“1”后，它们被内部上拉为高电平，并用作输入。作为输入，由于外部下拉为低电平，P3 口将输出电流（ILL）这是由于上拉的缘故。

P3 口也可作为 AT89C51 的一些特殊功能口，如下表所示：

口管脚 备选功能

P3.0 RXD（串行输入口）

P3.1 TXD（串行输出口）

P3.2 /INT0（外部中断 0）

P3.3 /INT1（外部中断 1）

P3.4 T0（记时器 0 外部输入）

P3.5 T1（记时器 1 外部输入）

P3.6 /WR（外部数据存储器写选通）

P3.7 /RD（外部数据存储器读选通）

P3 口同时为闪烁编程和编程校验接收一些控制信号。

RST：复位输入。当振荡器复位器件时，要保持 RST 脚两个机器周期的高电平时间。

ALE/PROG：当访问外部存储器时，地址锁存允许的输出电平用于锁存地址的地位字节。在 FLASH 编程期间，此引脚用于输入编程脉冲。在平时，ALE 端以不变的频率周期输出正脉冲信号，此频率为振荡器频率的 1/6。因此它可用作对外部输出的脉冲或用于定时目的。然而要注意的是：每当用作外部数据存储器时，将跳过一个 ALE 脉冲。如想禁止 ALE 的输出可在 SFR8EH 地址上置 0。此时，ALE 只有在执行 MOVX，MOVC 指令是 ALE 才起作用。另外，该引脚被略微拉高。如果微处理器在外部执行状态 ALE 禁止，置位无效。

/PSEN：外部程序存储器的选通信号。在由外部程序存储器取指期间，每个机器周期两次/PSEN 有效。但在访问外部数据存储器时，

这两次有效的/PSEN 信号将不出现。

/EA/VPP: 则在此期间外部程序存储器 当/EA 保持低电平时, (0000H-FFFFH), 不管是否有内部程序存储器。注意加密方式 1 时, /EA 将内部锁定为 RESET; 当/EA 端保持高电平时, 此间内部程序存储器。在 FLASH 编程期间, 此引脚也用于施加 12V 编程电源 (VPP)。

XTAL1: 反向振荡放大器的输入及内部时钟工作电路的输入。

XTAL2: 来自反向振荡器的输出。

振荡器特性: XTAL1 和 XTAL2 分别为反向放大器的输入和输出。该反向放大器可以配置为片内振荡器。石晶振荡和陶瓷振荡均可采用。如采用外部时钟源驱动器件, XTAL2 应不接。有余输入至内部时钟信号要通过一个二分频触发器, 因此对外部时钟信号的脉宽无任何要求, 但必须保证脉冲的高低电平要求的宽度。

1.2.2 MAX232 简介

MAX232 芯片是美信公司专门为电脑的 RS-232 标准串口设计的单电源电平转换芯片 由于标准规定 RS232 的电平采用 $\pm 12V$ 与 89C51 单片机串行口的 TTL 电平不一致。因此采用集成电平转换芯片 MAX232 进行电平转换成 5v 单电源供电, 方能实现电压一致性驱动电路。

该器件符合 TIA/EIA-232-F 标准, 每一个接收器将 TIA/EIA-232-F 电平转换成 5-V TTL/CMOS 电平。每一个发送器将 TTL/CMOS 电平转换成 TIA/EIA-232-F 电平。

如图 1.2 所示:

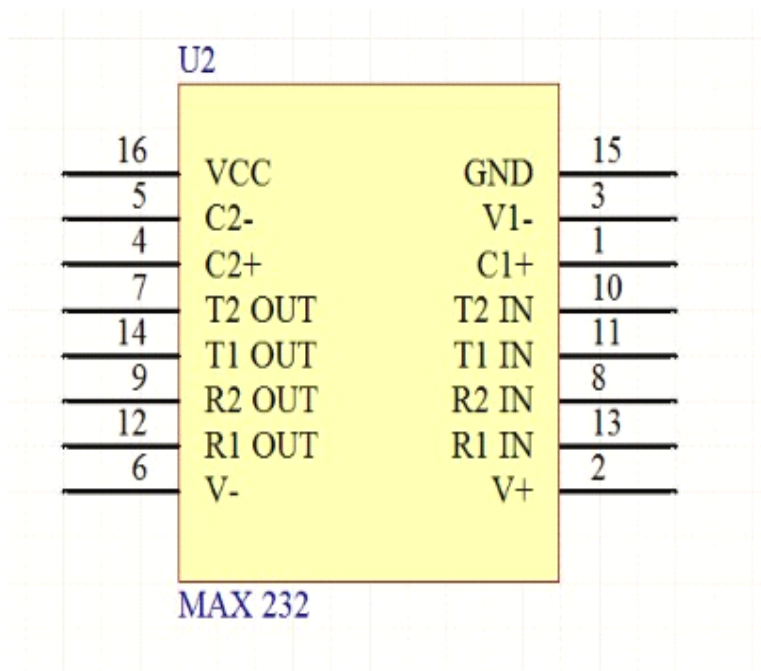


图 1.2 MAX 232 芯片

一. 引脚介绍

第一部分是电荷泵电路。由 1、2、3、4、5、6 脚和 4 只电容构成。功能是产生 12v 和 -12v 两个电源，提供给 RS-232 串口电平的需要。

第二部分是数据转换通道。由 7、8、9、10、11、12、13、14 脚构成两个数据通道。

其中 13 脚(R1IN)、12 脚(R1OUT)、11 脚(T1IN)、14 脚(T1OUT)为第一、9 数据通道。8 脚(R2IN) 脚(R2OUT) 、7、10 脚(T2IN) 脚(T2OUT)为第二数据通道。TTL/CMOS 数据从 T1IN、T2IN 输入转换成 RS-232 数据从 T1OUT、T2OUT 送到电脑 DB9 插头；DB9 插头的 RS-232 数据从 R1IN、R2IN 输入转换成 TTL/CMOS 数据后从 R1OUT、R2OUT 输出。

第三部分是供电。15 脚 GND、16 脚 VCC (+5v)。

二. 主要特点

- 1、符合所有的 RS-232C 技术标准
- 2、只需要单一 +5V 电源供电

3、片载电荷泵具有升压、电压极性反转能力，能够产生+10V 和 -10V 电压 V+、V-

4、功耗低，典型供电电流 5mA

5、内部集成 2 个 RS-232C 驱动器

6、内部集成两个 RS-232C 接收器

三.RS232 串行口标准

串行通信接口标准经过使用和发展，目前已经有多种；但都是在 RS232 标准的基础上经过改进而形成的。RS-232C 标准最初是由美国 EIA 和 BELL 公司一起开发并于 1969 年公布的通信进议，而至 1997 年，远程工业协会（TIA）又发布了最新的一个版本，命名为 TIA/EIA-232-F。而 ITF（国际电信联盟）和 CCITT（国际电话与电报顾问委员会）发布了一个类似的标准——V.28。这些标准都对串行通信接口的有关问题，如信号线功能、电器特性都作了明确规定。由于通信设备厂商都生产与 RS-232C 制式兼容的通信设备，因此，它作为一种标准，目前已在微机通信接口中广泛采用，不仅被内置于每一台计算机，同时也被内置于从微控制器到主机的多种类型的计算机及其相连接的设备；也就是常说的 RS232 端口。

PC 机的 RS232 端口与一般单片机上的 UART 接口从数据收发的时序上看，是一样的协议，不同的是两者用以表达逻辑“1”和“0”的规定是不一样的，即电平是不兼容的。RS232 端口用正负电压来表示逻辑状态，与 TTL（单片机的 UART 接口电平）以高低电平表示逻辑状态的规定是不同的。因此，为了能够同计算机接口和终端的 TTL 器件连接，必需进行电平和逻辑关系的变换。实现这种变换的方法可用分立元件，也可用集成电路芯片。MAX232 芯片就是目前应用较为广泛的电平转换器件，它可以实现 TTL 到 EIA 双向电平转换；该芯片支持两路串行异步通讯，RIN1 为第一路输入，ROUT1 为第一路输出；RIN2 为第二路输入管脚，ROUT2 为第二路输出。

第二章：串口通信原理

2.1 双向串口通信方框图

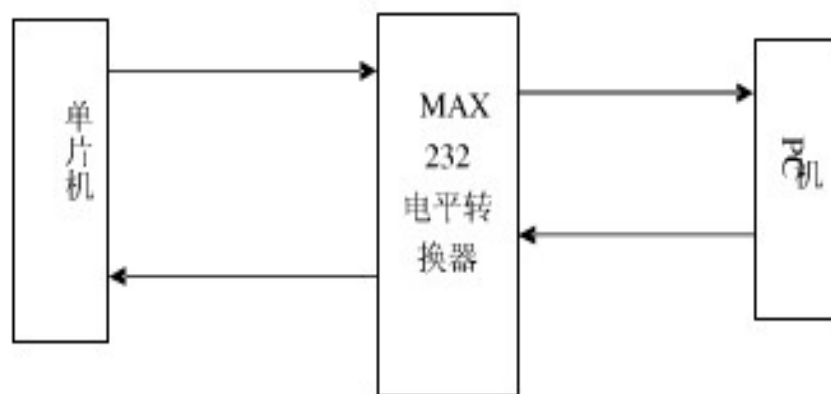


图 2.1 串行口通信方框图

2.1.1 通讯原理

串行通讯的基本概念：与外界的信息交换称为通讯。基本的通讯方式有并行通讯和串行通讯两种。

一条信息的各位数据被同时传送的通讯方式称为并行通讯。并行通讯的特点是：各数据位同时传送，传送速度快、效率高，但有多少数据位就需多少根数据线，因此传送成本高，且只适用于近距离（相距数米）的通讯。

一条信息的各位数据被逐位按顺序传送的通讯方式称为串行通讯。串行通讯的特点是：数据按位顺序传送，最少仅需一根传输线即可完成，成本低但传送速度慢。串行通讯的距离可以从几米到几千米。

根据信息的传送方向，串行通讯可以进一步分为单工、半双工和全双工三种。信息只能单向传送为单工；信息能双向传送但不能同时双向传送称为半双工；信息能够同时双向传送则称为全双工。

串行通讯又分为异步通讯和同步通讯两种方式。在单片机中，主要使用异步通讯方式。

SPCE061A 单片机有一个全双工串行口。全双工的串行通讯只需要一根输出线和一根输入线。数据的输出又称发送数据 (TXD)，数据的输入又称接收数据 (RXD)。串行通讯中主要有两个技术问题，一个是数据传送、另一个是数据转换。数据传送主要解决传送中的标准、格式及工作方式等问题。数据转换是指数据的串并行转换。具体说，在发送端，要把并行数据转换为串行数据；而在接收端，却要把接收到的串行数据转换为并行数据。

异步通讯：它用一个起始位表示字符的开始，用停止位表示字符的结束。其每帧的格式如下：

在一帧格式中，先是一个起始位 0，然后是 8 个数据位，规定低位在前，高位在后，接下来是奇偶校验位（可以省略），最后是停止位 1。用这种格式表示字符，则字符可以一个接一个地传送。

启动位 0 0 0 0 0 0 0 0 1 start stop 8 位数据 奇偶 优先 检查 位
结束位 parity

在异步通讯中，CPU 与外设之间必须有两项规定，即字符格式和波特率。字符格式的规定是双方能够对 0 和 1 的串理解成同一种意义。原则上字符格式可以由通讯的双方自由制定，但从通用、方便的角度出发，一般还是使用一些标准为好，如采用 ASCII 标准。波特率即数据传送的速率，其定义是每秒钟传送的二进制数的位数。

2.1.2 串口通讯接口

常用 PC 机串行接口有 3 种：PS / 2 接口用于连接键盘和鼠标；RS232C 串行接口一般用来实现 PC 机与较低速外部设备之间的远距离通信；USB 通用串行总线接口是现在比较流行的接口，它最大的好处在于能支持多达 127 个外设，外设可以独立供电，也可以通过 USB 接口从主板上获得 500 mA 5 V 的电流，并且支持热拔插，真正做到即插

即用。

PC 机的 3 种串行接口都可以用于与外设之间的数据通信，PS / 2 接口由于是专用于键盘和鼠标，在 PC 机的编程处理上要麻烦一些，而且在多数情况下，其他外设还不能占用。USB 接口有着功能强大、传输速度快、连接外设数量多，可向外设提供电源等特点，其应用越来越广，但是与 RS232C 串行接口比较，USB 接口的上位机即 PC 机程序的开发有着开发难度大、涉及知识面广、开发周期长等特点，同时在下位机即单片机硬件设计时必须选用带有 USB 接口的单片机或扩展专门的 USB 接口芯片，这必然会给下位机的软硬件系统设计增加难度并提高了软硬件成本。所以，USB 接口通常用于对传输速度要求高、传输功能复杂、或需上位机提供电源的外设和装置上。整体原理图如（图 2.2 串口通信原理图）。

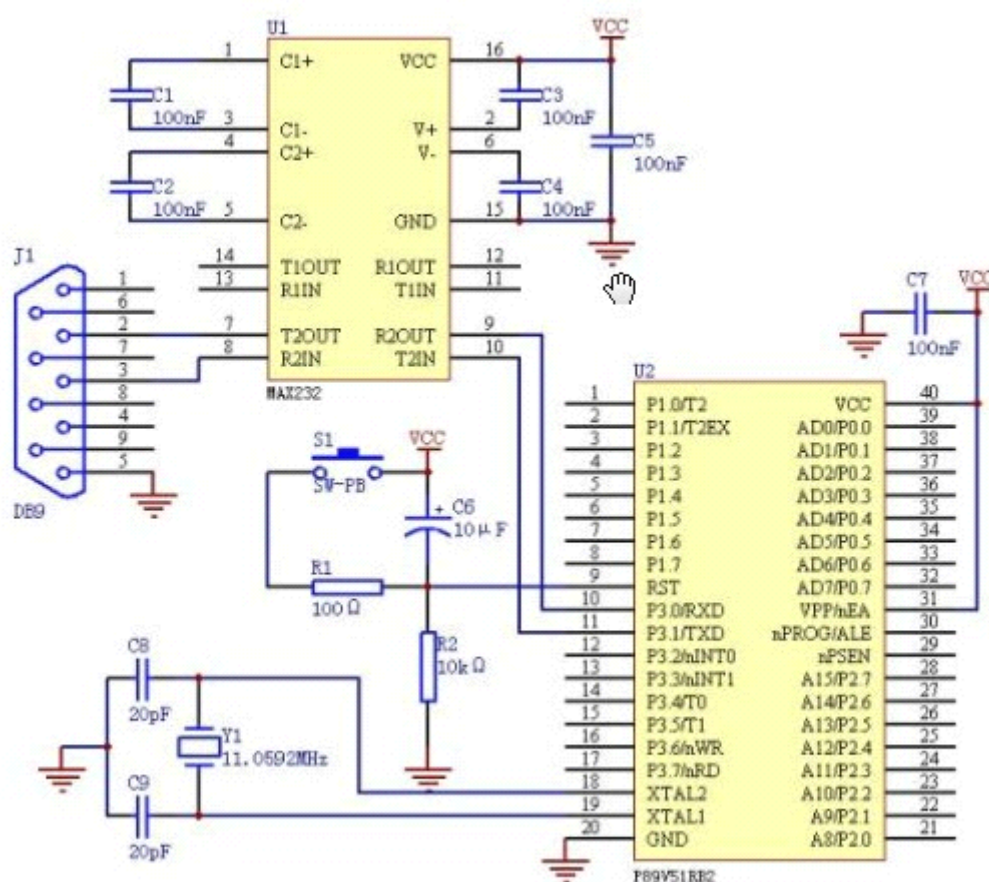


图 2.2 串行口通信原理图

在此电路中要注意：

- (1) VCC 接电源，必须保证稳定可靠；
- (2) nEA 管脚不可悬空，必须连到 VCC，或者通过上拉电阻接到 VCC；
- (3) 复位电路建议采用传统的 RC 复位，最好多带一个复位按键，以便操作；
- (4) 晶振的频率点不是任意的，推荐使用 11.0592MHZ 或者 22.1184MHZ；
- (5) nPSEN 管脚请悬空，不要接 GND；

2.2 程序下载

51 单片机有一个全双工的串行通讯口，所以单片机和计算机之间可以方便地进行串口通讯。进行串行通讯时要满足一定的条件，比如计算机的串口是 RS232 电平的，而单片机的串口是 TTL 电平的，两者之间必须有一个电平转换电路，我们采用了专用芯片 MAX232 进行转换，虽然也可以用几个三极管进行模拟转换，但是还是用专用芯片更简单可靠。我们采用了三线制连接串口，也就是说和计算机的 9 针串口只连接其中的 3 根线：第 5 脚的 GND、第 2 脚的 RXD、第 3 脚的 TXD。这是最简单的连接方法，但是对我们来说已经足够使用了，电路如下图所示，第 MAX232 的第 10 脚和单片机的 11 脚连接，9 脚和单片机的 10 脚连接，第 15 脚和单片机的 20 脚连接。如下图 3.1 图示

第三章：软件部分设计

3.1 PROTEUS 介绍

Proteus 软件是英国 Labcenter electronics 公司出版的 EDA 工具软件（该软件中国总代理为广州风标电子技术有限公司）。它不仅具有其它 EDA 工具软件的仿真功能，还能仿真单片机及外围器件。它是目前最好的仿真单片机及外围器件的工具。虽然目前国内推广刚起步，但已受到单片机爱好者、从事单片机教学的教师、致力于单片机开发应用的科技工作者的青睐。Proteus 是世界上著名的 EDA 工具仿真软件，从原理图布图、代码调试到单片机与外围电路协同仿真，一键切换到 PCB 设计，真正实现了从概念到产品的完整设计。是目前世界上唯一将电路仿真软件、PCB 设计软件和虚拟模型仿真软件三合一的设计平台，其处理器模型支持 8051、HC11、PIC10/12/16/18/24/30/DsPIC33、AVR、ARM、8086 和 MSP430 等，2010 年即将增加 Cortex 和 DSP 系列处理器，并持续增加其他系列处理器模型。在编译方面，它也支持 IAR、Keil 和 MPLAB 等多种编译器。

一. 其功能特点

Proteus 软件具有其它 EDA 工具软件（例：multisim）的功能。这些功能是：

- （1）原理布图
- （2）PCB 自动或人工布线
- （3）SPICE 电路仿真

二. 革命性的特点

- （1）互动的电路仿真 用户甚至可以实时采用诸如 RAM，ROM，键

盘，马达，LED，LCD，AD/DA，部分 SPI 器件，部分 IIC 器件。

(2) 仿真处理器及其外围电路可以仿真 51 系列、AVR、PIC、ARM、等常用主流单片机。还可以直接在基于原理图的虚拟原型。上编程，再配合显示及输出，能看到运行后输入输出的效果。配合系统配置的虚拟逻辑分析仪、示波器等，Proteus 建立了完备的电子设计开发环境。

三．具有 4 大功能模块

(1) 智能原理图设计 (ISIS)

(2) 完善的电路仿真功能 (Prospice)

(3) 独特的单片机协同仿真功能 (VSM)

(4) 实用的 PCB 设计平台

四．电路功能仿真

在 PROTEUS 绘制好原理图后，调入已编译好的目标代码文件：.HEX，可以在 PROTEUS 的原理图中看到模拟的实物运行状态和过程。PROTEUS 是单片机课堂教学的先进助手。PROTEUS 不仅可将许多单片机实例功能形象化，也可将许多单片机实例运行过程形象化。前者可在相当程度上得到实物演示实验的效果，后者则是实物演示实验难以达到的效果。目前 Proteus 的最新版为 7.7 sp2 今年将推出 8.0 版本，增加 DSP 系列及 ARM cortex 处理器。

3.2 程序设计

3.2.1 课程设计的程序要求：

单片机接受 PC 机发来的不同指令信号，完成不同的工作。

本设计程序流程：在 PC 机上使用串口调试软件，向单片机发送字符串。

(1) 当接收到指令信号为：‘go’。单片机向 PC 机发送字符串 ‘WELCOME TO CHINA!’；

(2) 当接收到指令信号为：‘who’。单片机向 PC 机发送字符串

‘MY NAME IS LIMING!’ ；

(3) 当接收到指令信号为：‘home’ 。单片机向 PC 机发送字符串
‘YINGYONGKEXUEXUEYUAN!’ ；

这时，串行口调试软件就会在窗口显示出相应的字符串，如果接受的字符串不正确，单片机不响应。

3.2.2 流程图

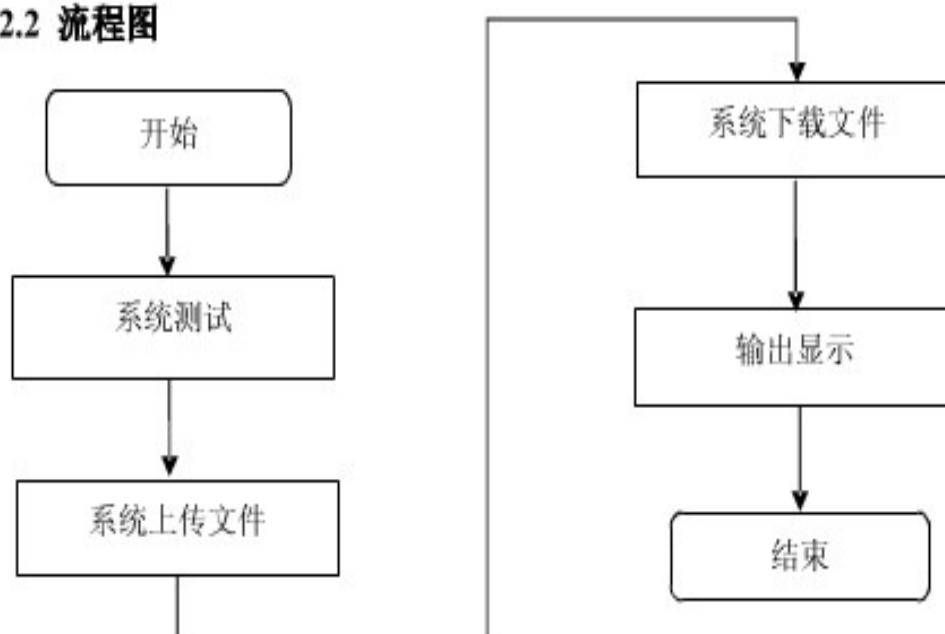


图 3.1 串行口通信流程图

3.2.3 程序

见附录

第四章：调试与仿真

4.1 调试助手设置

串口是计算机上一种非常通用设备通信的协议(小要与通用串行总线 Universal Serial Bus 或者 uSB 混淆)。大多数计算机包含两个基丁 RS232 的串口。串口同时也是仪器仪表设备通用的通信协议;很多 GPIB 兼容的设备也带有 Rs - 232 u。同时,串 u 通信协议也可以用于获取远程采集设备的数据。

串口通信的概念非常简单,串口按位(bit)发送和接收字节。尽管比按字节(byte)的并行通信慢,但是串口可以在使用一根线发送数据的同时用另一根线接收数据。它很简单并且能够实现远距离通信。比如 IEEE488 定义并行通行状态时,规定设备线总常不得超过 20 米,并且任意两个设备问的长度不得超过 2 米;而对于串口而言,长度可达 1200 米。

4.1.1 串口用于 ASCII 码字符的传输设置

通信使用 3 根线完成:(1)地线,(2)发送,(3)接收。由于串口通信是异步的,端口能够在在一根线上发送数据同时在另一根线上接收数据。其他线用于握手,但是不是必须的。串口通信最重要的参数是波特率、数据位、停止位和奇偶校验。对于两个进行通行的端口,这些参数必须匹配:

a, 波特率:这是一个衡量通信速度的参数。它表示每秒钟传送的 bit 的个数。例如:300 波特表示每秒钟发送. 300 个 bit。当我们提到时钟周期时,我们就是指波特率例如如果协议需要 4800 波特率,那么时钟是 4800HZ 串口通信在数据线上的采样率为 4800Hz。通常电话线的波特率为 14400, 28800 和 36600。波特率可以远远大于这些值,但是波特率和距离成反比。高波特率常常用于放置的很近的仪器间的通信,典型的例子就是 GPIB 设备的通信。

b, 数据位:这是衡量通信中实际数据位的参数。当计算机发送

一个信息包，实际的数据不会是 8 位的，标准的值是 5、7 和 8 位。如何设置取决于你想传送的信息。比如，标准的 AS(：II 码是 0~127(7 位)。扩展的 AS(：II 码是 0~255(8 位)。如果数据使用简单的文本(标准 AS(：II 码)，那么每个数据包使用 7 位数据。每个包是指一个字节，包括开始 / 停止位，数据位和奇偶校验位。由于实际数据位取决于通信协议的选取，术语“包”指任何通信的情况。

c，停止位：用于表示单个包的最后一位。典型的值为 1，1.5 和 2 位。由于数据是在传输线上定时的，并且每一个设备有其自己的时钟，很可能在通信中两台设备间出现了小小的不同步。因此停止位不仅仅是表示传输的结束，并且提供计算机校正时钟同步的机会。适用于停止位的位数越多，不同时钟同步的容忍程度越大，但是数据传输率同时也越慢。

d，奇偶校验位：在串口通信中一种简单的检错方式。有四种检错方式：偶、奇、高和低。当然没有校验位也是可以的。对于偶和奇校验的情况，串口会设置校验位(数据位后面的一位)，用一个值确保传输的数据有偶个或者奇个逻辑高位。例如，如果数据是 011，那么对于偶校验，校验位为 0，保证逻辑高的位数是偶数个。如果是奇校验，校验位为 1，这样就有 3 个逻辑高位。高位和低位不真正的检查数据，简单置位逻辑高或者逻辑低校验。这样使得接收设备能够知道一个位的状态，有机会判断是否有噪声干扰了通信或者是否传输和接收数据是否不同步。

4. 2 串口设置

为了使调成程序，我们首先要对串 u 进行设置。

4. 2. 1 通常设置

1. 支持 1—64 个串行端口。
2. 可设置非标准或更高的波特率，不限于列表中的特定波特率。
3. 可设置串 u 的 2 个输出线路信号 DTR / RTS 的状态，以及 TXD 信号

的 BREAK 状态。

4. 可实时显示串 u 的 4 个输入线路信号 cTs/DSR/RING/DCD 状态。
5. 可分别清空发送区或接收区的数据。
6. 可对发送的数据和接收的数据单独计数。

一. 发送数据：

1. 发送数据时，可采用手动发送和自动发送，自动发送还可以指定发送的周期。
2. 发送区的数据可以按字符方式或 16 进制方式解释、发送。
3. 按字符方式发送时，发送区可输入中英文混合字符。
4. 按 16 进制方式发送时，发送区的每个数据间只需用格或回车隔开，每个数据只有最低 2 个 16 进制字符有效。
5. 发送区的数据可以从文件中调入，读入到发送区时，可按字符方式或 16 进制方式显示。

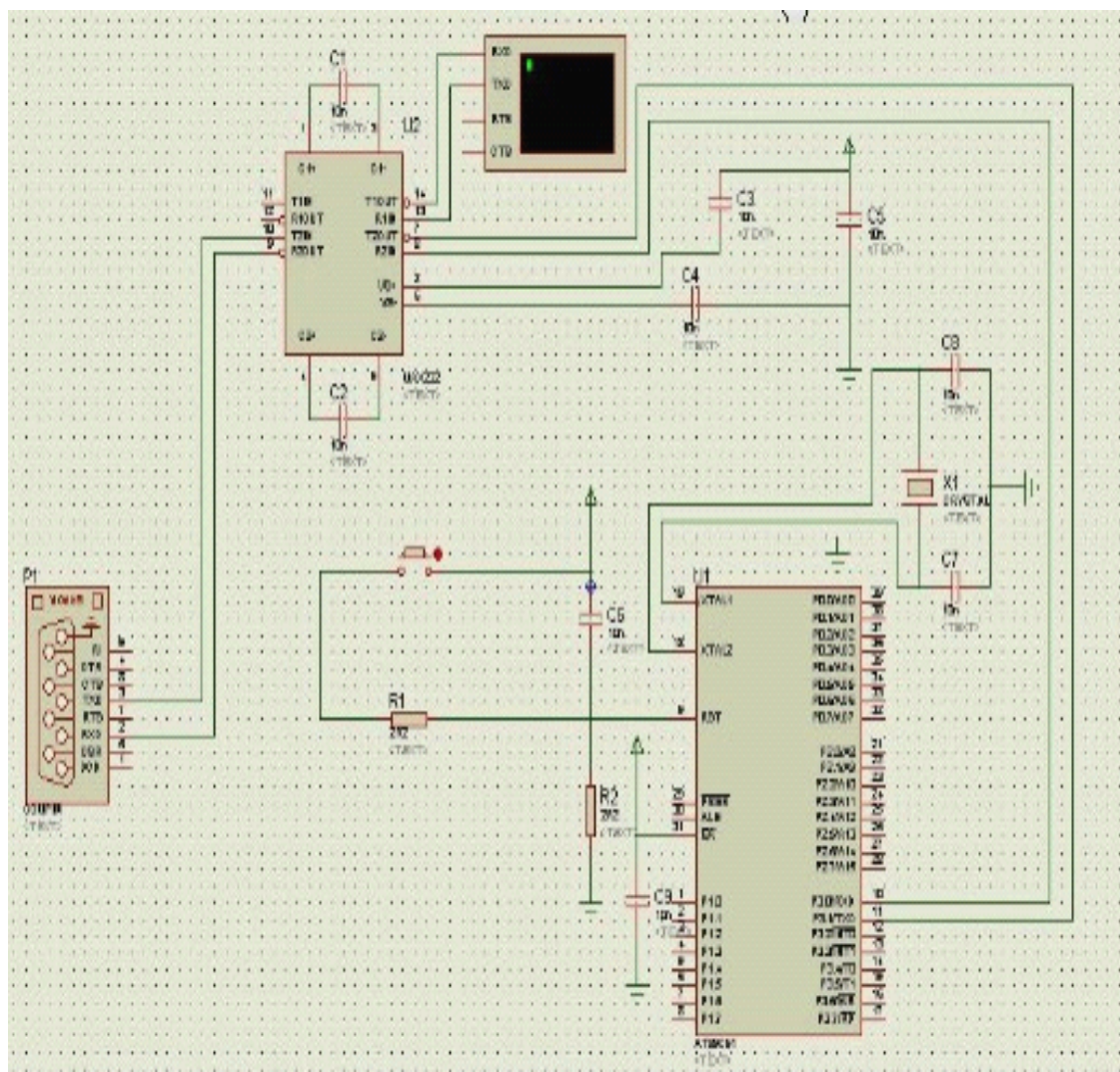


图 4. 1 调试原理图

二．接收数据：

1. 接收区的数据可以按宁符方式或 16 进制方式显示。
2. 按宁符方式接收时，在低速波特率下，可正确接收和显示中英文混合宁符。
3. 可手动或自动保存接收区内容到指定文件中。自动保存的接收内容长度没有限制。

可采用仿超级终端方式操作和显示。

4. 2. 2 调试流程图

调试内容为第四章的实验要求。

按要求完成实验内容。

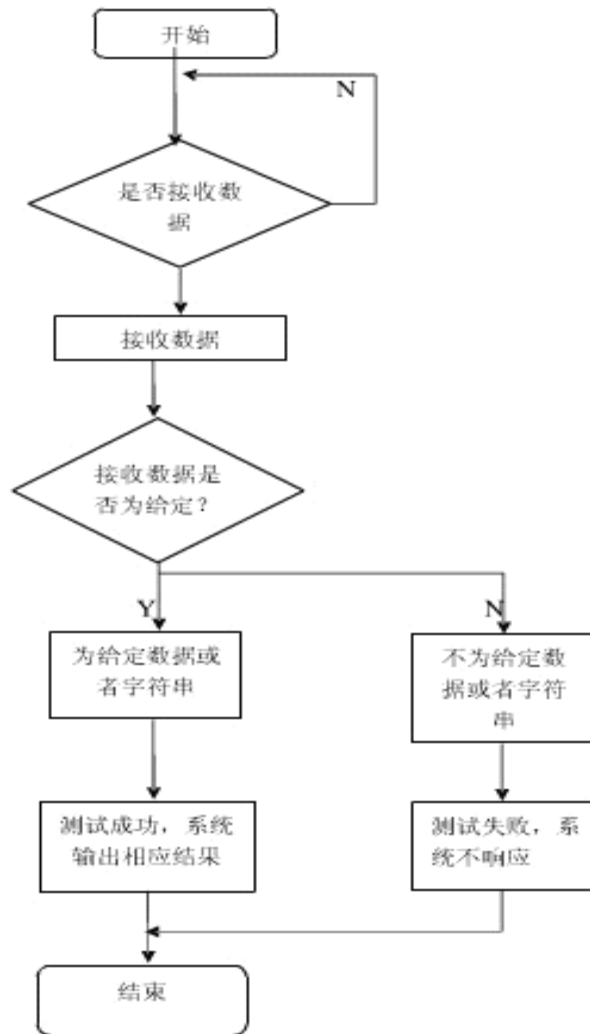


图 4.2 调试流程图

4. 2. 3 串口系列选择

V2. 00 2002 / 12 / 31 159K

1. 串行端口号增加到 16 个。

2. 增加一些选项，可选择以下功能：

(1) 可选择接收区的光标是台总是跟随存接收内容的最后。

a. 如果是，可以看到最新接收到的内容，但当内容比较多时，接收区会闪烁。

b. 如果小是，则会看小到最后接收到的内容，即屏幕小刷新，但接收速度快。

(2) 可选择存接收字符时，控制字符(0 `` 0h — 1 F1])是台显示为[xx]。

a. 如果是，则小可显示的控制字符将按 16 进制方式显示，用[]括起来。

b. 如果小是，控制字符会显示为空格或其他，特别是若接收到控制字符 0 `` 0h 后会造成之后接收到的其它内容都无法正常显示出来。

(3)可选择是台仿超级终端方式。

a. 如果是，则发送和接收的内容都同时显示在接收区中，特别方便 M. dem 或 ADAM40 `` 00 系列等字倚设备的调试。

b. 如果小是，则接收区只显示接收到的内容。

(4)可选择是台自动保存接收区内容。

a. 如果是，可设置自动保存的接收字符总数，以及保存文件名。接收之前，先按[开始保存]，之后每接收满指定的字符数，就自动保存到文件中，并清空接收区以便继续接收。接收完成后，按[停止保存]可得到完整的接收文件。接收内容的长度没有限制。

b. 如果不是，则需要自己手动保存。接收完成后，按[保存为]，再输入文件名，确认后，就可保存接收区的所有内容到指定文件中。

3. 发送/接收显示区的范围大小可自由上下左右调整。

4. 可选择是否计数。

5. 修改一些 BUG。

V1. 10 2001 / 12 / 30 153K

1. 史换通讯模块，支持在 wind. ws 9x/Me/NT/2000/xP 下正确运行。

2. 增加 TxD 信号的 BREAK 控制。

3. 增加发送数据的流量控制功能。

4. 在低速波特率下，仍能正确接收和显示中文字符。

5. 串口初始化时，根据 DTR/RTS/BREAK 的选择，自动设置输出信号的状态。

6. 接收区显示字符超过 30000 个，就自动清空。

设计总结

这次单片机课程设计历时四天，整整四天日子里，可以说是熬夜上网查书，虽然很扛不住但是学到很多的知识，而且学到了很多在书本上所没有学到过的知识。通过这次设计，进一步加深了对单片机的了解，让我对它有了更加浓厚的兴趣。特别是当每一个子模块编写调试成功时，心里特别的开心。但是在编写顶层文件的程序时，遇到了不少问题，特别是各元件之间的连接，以及定时器的设定，总是有错误，在细心的检查下，终于找出了错误和警告，排除困难后，程序编译就通过了，心里终于舒了一口气。在调试仿真时，也遇到了一点困难，想要的结果总是不能正确的显示，后来，在多次的调试之后，才发现是因为输入的定时器时间太长了。经过屡次调试，终于找到了比较合适的输入数值，仿真成功了。

其次，在连接各个模块编译程序的时候一定要注意各个变量的定义和功能，因为每个定义和功能不一样的，会影响实验结果，再者就是程序需规范化，语法不能出错，编译才能通过，试验才能成功。

通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。在设计的过程中遇到问题，可以说得是困难重重，在实验过程中难免会遇到过各种各样的问题，同时在设计的过程中发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固。

同时也确定了，单片机在目前的发展形势下，表现出几大趋势：可靠性及应用越来越水平高和互联网连接已是一种明显的走向。所集成的部件越来越多；Ns(美国国家半导体)公司的单片机已把语音、图象部件也集成到单片机中，也就是说，单片机的意义只是在于单片集

成电路，而不在于其功能了；如果从功能上讲它可以讲是万用机。原因是其内部已集成上各种应用电路。功耗越来越低和模拟电路结合越来越多。随着半导体工艺技术的发展及系统设计水平的提高，单片机还会不断产生新的变化和进步，最终人们可能发现：单片机与微机系统之间的距离越来越小，甚至难以辨认。

总的来说，这次设计的串口通信还是比较成功的，在设计中遇到了很多问题，最后在老师的辛勤的指导下，终于游逆而解，有点小小的成就感，终于觉得下时所学的知识有了实用的价值，达到了理论与实际相结合的目的，不仅学到了不少知识，而且锻炼了自己的能力，使自己对以后的路有了更加清楚的认识。最后对给过我帮助的所有同学和各位指导老师再次表示衷心的感谢！

参考文献

- 【1】官波、容玉平. AT89c51 的扩展串口与 Pc 机间的通讯[J]. 2001
- 【2】马宏忠. 多台单片机与 IBM—Pc 机间的快速通讯[J]. 电气自动化
- 【3】朱明宇, 武峰. 8031 单片机丰从式数据采集系统设计[J], 1997,
- 【4】李华. Mcs51 系列单片机实用接口技术[M]. 北京: 北京航空航天大学出版社. 1993
- 【5】张毅坤. 单片微型计算机原理及应用, 西安电子科技大学出版社. 1998
- 【6】余锡存曹国华. 单片机原理及接口技术[M]. 陕西: 西安电子科技大学出版社. 2000. 7
- 【7】雷丽文等. 微机原理与接口技术[M]. 北京: 电子工业出版社. 1997. 2
- 【8】余永权. ATMEL89 系列(MCS 51 兼容)FI_ASH 单片机原理及应用. 1997
- 【9】李建忠 单片机原理及应用 西安电子科技大学出版社. 2002(1)
- 【10】胡汉才. 单片机原理及其接口技术. 清华大学出版社. 1996 年
- 【11】戴佳戴卫恒 c 语言应用程序设计电子工业出版社. 2006(1)
- 【12】潘新民王燕芳. 微型计算机控制技术. 电子工业出版社. 2003 年
- 【13】李钟实. 实用电子报警器精选百例. 北京: 科学技术文献出版社. 2002(4)
- 【14】江思敏姚鹏翼胡荣. Protel 电路设计教程. 清华大学出版社. 2002
- 【15】阎石 数字电子技术基础. 高等教育出版社. 1998(2)
- 【16】童诗白. 模拟电子技术基础. 高等教育出版社. 1998(4)
- 【17】龚运新 单片机 c 语言开发技术 清华大学出版社. 2006(1)
- 【18】王延才 电力电子变流技术 机械工业出版社
- 【19】王文郁 晶闸管变流技术应用图集或电力电子技术应用电路 机械工业出版社
- 【20】曾芳 电力电子变流技术 西安电子科技大学出版社
- 【21】张华 电类专业课程设计指导 机械工业出版社

附录

```
#include<reg51.h>

code unsigned char r1[]={"go"};
code unsigned char r2[]={"who"};
code unsigned char r3[]={"home"};
code unsigned char s1[]={"WELCOME TO CHINA"};
code unsigned char s2[]={"MY NAME IS LIMING"};
code unsigned char s3[]={"YINGYONGKEXUEXUEYUAN"};
unsigned char send_s
unsigned char s_count,s_count2,t_count;
unsigned char temp;
unsigned char buf[20];
void service_intt0() interrupt 1 using 2 //串口中断号为1，后面的 using
为寄存器组切换 2
{
    EA=0;
    t_count++;
    if(t_count>=10)
    {
        if(s_count==2)
        {
            if((buf[0]==r1[0])&&(buf[1]==r1[1]))
            {
                send_s=1
            }
        }
    }
```

```

else if(s_count==3)
{
    if((buf[0]==r2[0])&&(buf[1]==r2[1])&&(buf[2]==r2[2]))
    {
        send_s=2;
    }
}
else if(s_count==4)
{
    if((buf[0]==r3[0])&&(buf[1]==r3[1])&&(buf[2]==r3[2]))==r3
[3]))
    {
        send_s=3;
    }
}
s_count=0;
t_count=0;
}
TH0=0x3c;
TL0=0xb0;
EA=1;
}

void service_ints() interrupt 4 using 3    //串口中断号为 4.
using 为寄存器组切换 3
{
    EA=0;
    if(RI==1)

```

```

    {
        buf[s_count]=SBUF;

        s_count++;

        t_count=0;

        TH0=0x3c;

        TL0=0xb0;

        RI=0;

    }

    else if(TI==1)

    {

        ;

    }

    EA=1;

}

void main()

{

    TMOD=0x21;           //定时器 1 工作在方式 2，定时器 0 工作在方式

1    TH0=0x3c;           //波特率为 9600

    TL0=0xb0;

    TH1=0xfd;           //实现波特率为 9600（系统时钟 11.0592MHZ）

    TL1=0xfd;

    TR0=1;               //启动定时器 0

    TR1=1;               //启动定时器 1

    SCON=0x50;           //选择串口工作方式. 打开接受允许

    PCON=0x90;           //波特率不增倍

```

```

RI=0;                //清楚中断标志位
TI=0;                //允许处理发送中断
ET0=1;
ES=1;                //允许串口中断
EA=1;                //单片机中断允许
while(1)
{
    if(send_s==1)
    {
        EA=0;
        s_count2=0;
        while(s1[s_count2] != '\0')
        {
            SBUF=s1[s_count2];
            while(TI==0) {;}          //TI=0, 则中断响应, 不做任何
处理

            TI=0;
            s_count2++;
        }
        send_s=0;
        TH0=0x3c;                //波特率设置
        TL0=0x0b;
        EA=1;                    //单片机中断允许
    }
    else if(send_s==2)
    {
        EA=0;

```

```

s_count2=0;
while(s2[s_count2=0]!='\0')
{
    SBUF=s2[s_count2];
    while(TI==0){;}
    TI=0;
    s_count2++;
}
send_s=0;
TH0=0x3c;
TL0=0x0b;

}
else if(send_s==3)
{
    EA=0;
    s_count2=0;
    while(s3[s_count2=0]!='\0')
    {
        SBUF=s3[s_count2];
        while(TI==0){;}
        TI=0;
        s_count2++;
    }
    send_s=0;
    TH0=0x3c;
    TL0=0x0b;

```



```
        EA=1;
    }
    else if(send_s==4)
    {
        EA=0;
        send_s=0;
        TH0=0x3c;
        TL0=0x0b;
        EA=1;
    }
}
}
```