



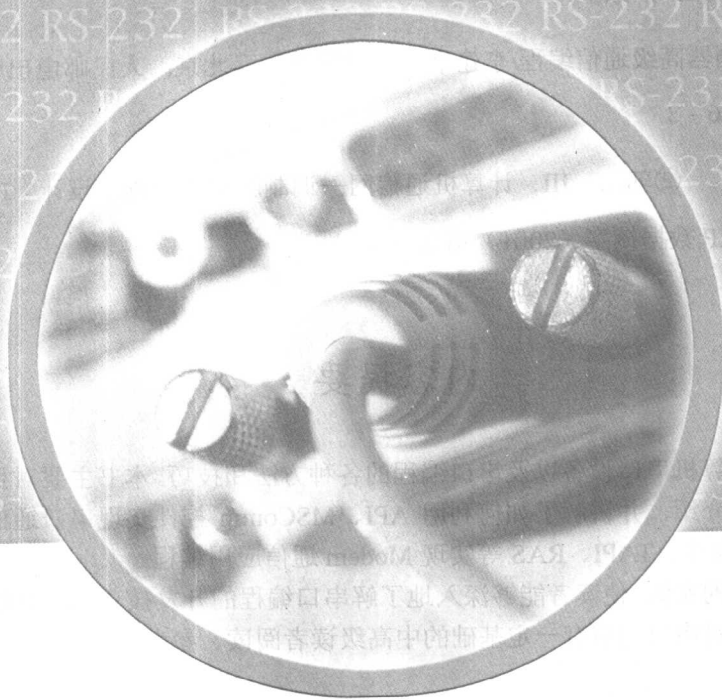
# RS-232 和 调制解调器 高级通信编程

李建华 郭明 编著

人民邮电出版社  
[www.pptph.com.cn](http://www.pptph.com.cn)



附光盘  
CD-ROM



# RS-232 和 调制解调器 高级通信编程

李建华 郭明 编著

人民邮电出版社

## 图书在版编目 (CIP) 数据

RS-232 和调制解调器高级通信编程/李建华, 郭明编著. —北京: 人民邮电出版社, 2001.7

ISBN 7-115-09446-2

I. R... II. ①李... ②郭... III. 计算机通信网—调制解调器—程序设计 IV. TN915.05

中国版本图书馆 CIP 数据核字 (2001) 第 039621 号

## 内容提要

本书介绍了计算机串口设备以及串口编程的各种方法和技巧。本书主要对目前常用的各种串口设备进行了介绍, 并讲解了如何利用 API、MSComm 控件实现串口通信应用编程, 以及如何利用 AT 命令、TAPI、RAS 等实现 Modem 通信应用编程。

书中附有大量的实例, 使读者能够深入地了解串口编程的开发技巧。本书技术内容较深, 实用性强, 适合于对串口通信有一定基础的中高级读者阅读、参考。

MS185/08

### RS-232 和调制解调器高级通信编程

- ◆ 编 著 李建华 郭 明  
责任编辑 张立科
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@pptph.com.cn  
网址 <http://www.pptph.com.cn>  
读者热线: 010-67129212 010-67129211(传真)  
北京汉魂图文设计有限公司制作  
北京顺义向阳胶印厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 787 × 1092 1/16  
印张: 22.5  
字数: 552 千字 2001 年 7 月第 1 版  
印数: 1-5000 册 2001 年 7 月北京第 1 次印刷

ISBN 7-115-09446-2/TP·2326

定价: 48.00 元(附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010)67129223

计算机串口编程在通信软件中有着广泛的应用，如电话、传真、视频和控制等。尽管串口通信所包含的内容纷繁复杂，每一位程序员还是不可避免地要和串行端口打交道。

本书的侧重点是将理论知识与应用相结合，系统地介绍串口编程中常见的，而一般程序员比较头痛的知识点，包括一些 VB 编程中非常有用的知识。本书面向对串口通信有一定基础的读者。本书还配有大量简洁的、有针对性的源代码，以供读者参考。

本书章节概述如下：

## 第 1 章： 串口通信基础知识

本章介绍有关数据通信的基本概念和有关串口的硬件知识，这些内容对于一个串口通信程序开发者是必要的。

## 第 2 章： 用汇编语言开发串口通信程序

在有些场合适当地应用汇编嵌入式语言编写计算机的串口通信程序，可以使程序更加灵活、高效。作为一个串口通信的程序员对低级语言的了解和对底层硬件的掌握是必不可少的，本章介绍了如何用汇编语言开发串口通信程序。

## 第 3 章： Windows API 串口编程

Windows 环境下的串口编程是与设备无关的。通过 Windows 通信 API 函数编写串口通信应用程序，因而不需要对硬件直接进行操作。本章介绍了 Windows 通信 API 函数，并给出了利用 Windows 通信 API 函数编写串口通信程序的实例源程序。

## 第 4 章： 使用 MSComm 控件编程

利用 MSComm 控件，可以使编写串口程序变得十分简单。本章介绍了 MSComm 控件的属性和事件，介绍了串口通信中的字符传输等知识，并给出了利用 MSComm 控件编写字符仿真和编写串口通信程序的实例源程序。

## 第 5 章： Modem 通信技术

本章主要介绍了 Modem 的各种通信规则、AT 命令以及 Modem 的返回信息码，还详细地介绍了异步接收发送器的内部结构以及实际应用。

## 第 6 章： 开发 Modem 应用程序

本章主要介绍了如何利用 MSComm 控件向 Modem 送出控制命令，并提供了 Modem 连接和文件传输的实例源程序。

## 第 7 章： Windows TAPI 应用

TAPI 服务为通信应用程序提供了一种与设备无关的方法，避免了使用 AT 命令等直接访问硬件可能造成的冲突，并且可以让应用程序与各种通信网络进行通信。本章主要介绍了 TAPI 的拨号理论，并提供了利用 TAPI 拨号的实例源程序。

## 第 8 章： RAS 高级编程

本章主要介绍如何编写拨号上网程序、编辑管理电话簿以及连接管理等，同时介绍

了相关理论。

#### 第 9 章：传真机的串口编程

本章介绍传真机的串口编程的基础知识、EIA 传真调制解调器简介、传真编码以及应用实例等。

#### 第 10 章： 串口通信的工业应用

串口通信在工业中的应用很广泛，计算机可以通过串口得到设备的某种数据，而后利用计算机强大的功能进行处理，发送出另一种数据来控制设备。本章介绍了两个利用串口通信的工业应用实例。

本书的实例都是经过作者精心调试的，代码都放在随书附赠的 CD-ROM 光盘中。由于作者水平有限，书中错误之处在所难免，欢迎读者朋友批评指正。

编者

2001 年 6 月

# 目 录

|                                     |           |
|-------------------------------------|-----------|
| <b>第 1 章 串口通信基础知识 .....</b>         | <b>1</b>  |
| 1.1 数据通信的基础知识 .....                 | 1         |
| 1.1.1 数据终端设备与数据通信设备 .....           | 1         |
| 1.1.2 字符传输 .....                    | 2         |
| 1.1.3 握手和缓冲存储器 .....                | 6         |
| 1.2 硬件基础知识 .....                    | 10        |
| <b>第 2 章 用汇编语言开发串口通信应用程序 .....</b>  | <b>14</b> |
| 2.1 用嵌入式汇编语言编写计算机的串口通信程序 .....      | 14        |
| 2.1.1 PC 系统及体系结构 .....              | 14        |
| 2.1.2 汇编语言的基本操作 .....               | 18        |
| 2.1.3 DOS 串行通信口功能 .....             | 19        |
| 2.1.4 BIOS 串行通信口功能 .....            | 20        |
| 2.1.5 高级语言中嵌入汇编语言指令 .....           | 23        |
| 2.2 汇编语言编写串口通信应用实例 .....            | 23        |
| <b>第 3 章 Windows API 串口编程 .....</b> | <b>28</b> |
| 3.1 Windows 串口通信编程 .....            | 28        |
| 3.1.1 Windows 通信 API 和串口通信 .....    | 28        |
| 3.1.2 打开和关闭串口 .....                 | 28        |
| 3.1.3 串口配置和串口属性 .....               | 31        |
| 3.1.4 读写串口 .....                    | 41        |
| 3.1.5 通信事件 .....                    | 48        |
| 3.1.6 设备控制命令 .....                  | 50        |
| 3.2 利用 Windows API 串口通信高级实例 .....   | 52        |
| 3.2.1 VB 中调用 Windows API .....      | 52        |
| 3.2.2 建立工程项目 .....                  | 53        |
| 3.2.3 代码分析 .....                    | 54        |
| <b>第 4 章 使用 MSComm 控件编程 .....</b>   | <b>79</b> |
| 4.1 MSComm 控件 .....                 | 79        |
| 4.1.1 引用 MSComm 控件 .....            | 79        |
| 4.1.2 MSComm 控件信息处理 .....           | 80        |
| 4.1.3 MSComm 控件的属性 .....            | 80        |
| 4.1.4 MSComm 控件的事件 .....            | 83        |

|              |                              |            |
|--------------|------------------------------|------------|
| 4.1.5        | 利用 MSComm 控件的通信步骤 .....      | 83         |
| 4.2          | 串口通信中字符传输 .....              | 84         |
| 4.2.1        | ASCII 控制字符 .....             | 84         |
| 4.2.2        | 通信中的字符和字节 .....              | 88         |
| 4.3          | VB 使用 MSComm 控件高级编程实例 .....  | 90         |
| 4.3.1        | 建立工程项目 .....                 | 90         |
| 4.3.2        | 分析代码 .....                   | 90         |
| <b>第 5 章</b> | <b>Modem 通信技术 .....</b>      | <b>106</b> |
| 5.1          | 基础知识 .....                   | 106        |
| 5.1.1        | Modem 简介 .....               | 106        |
| 5.1.2        | RS-232C 标准 .....             | 108        |
| 5.1.3        | 文件传输协议 .....                 | 116        |
| 5.2          | Modem 通信的规则和标准 .....         | 124        |
| 5.2.1        | Modem 状态 .....               | 124        |
| 5.2.2        | AT 命令 .....                  | 125        |
| 5.2.3        | S 寄存器 .....                  | 139        |
| 5.2.4        | Modem 返回信息码 .....            | 142        |
| 5.3          | 通用异步接收发送器 UART .....         | 143        |
| 5.3.1        | 异步串口硬件结构 .....               | 143        |
| 5.3.2        | UART 内部结构 .....              | 145        |
| 5.3.3        | 实际的 UART (NS16550) .....     | 156        |
| <b>第 6 章</b> | <b>开发 Modem 应用程序 .....</b>   | <b>160</b> |
| 6.1          | 两台计算机通过 Modem 建立连接 .....     | 160        |
| 6.1.1        | 建立项目 .....                   | 160        |
| 6.1.2        | 代码分析 .....                   | 160        |
| 6.2          | 两台计算机利用 Modem 进行文件传输 .....   | 164        |
| 6.2.1        | 建立项目 .....                   | 164        |
| 6.2.2        | 代码分析 .....                   | 165        |
| <b>第 7 章</b> | <b>Windows TAPI 应用 .....</b> | <b>169</b> |
| 7.1          | 简单的拨号 .....                  | 169        |
| 7.1.1        | 简单的拨号理论 .....                | 169        |
| 7.1.2        | 简单的拨号程序——SimpleDial 程序 ..... | 170        |
| 7.2          | Windows TAPI 通信编程 .....      | 173        |
| 7.2.1        | TAPI 版本介绍 .....              | 174        |
| 7.2.2        | TAPI 初始化 .....               | 174        |
| 7.2.3        | TAPI 版本协商 .....              | 177        |
| 7.2.4        | TAPI 线路控制 .....              | 178        |
| 7.2.5        | TAPI 地址信息 .....              | 184        |

|              |                          |            |
|--------------|--------------------------|------------|
| 7.2.6        | TAPI 属性配置 .....          | 195        |
| 7.2.7        | TAPI 呼叫控制 .....          | 198        |
| 7.3          | 利用 TAPI 拨打电话编程高级实例 ..... | 202        |
| 7.3.1        | 建立工程项目 .....             | 202        |
| 7.3.2        | 代码分析 .....               | 203        |
| <b>第 8 章</b> | <b>RAS 高级编程 .....</b>    | <b>234</b> |
| 8.1          | RAS 客户机 .....            | 234        |
| 8.2          | 建立拨号连接 .....             | 235        |
| 8.3          | RAS 简单拨号程序 .....         | 239        |
| 8.4          | RAS 重要函数 .....           | 240        |
| 8.4.1        | 连接函数 .....               | 240        |
| 8.4.2        | 连接管理函数 .....             | 245        |
| 8.4.3        | 电话簿和用户凭证管理 .....         | 248        |
| 8.4.4        | 拨号方式 .....               | 249        |
| 8.5          | RAS 高级程序开发实例 .....       | 251        |
| 8.5.1        | 建立工程项目 .....             | 251        |
| 8.5.2        | 程序运行结果图 .....            | 251        |
| 8.5.3        | 关键代码分析 .....             | 256        |
| 8.5.4        | RAS 编程小结 .....           | 291        |
| 8.6          | RAS 应用实例——远程文件共享 .....   | 292        |
| <b>第 9 章</b> | <b>传真机的串口编程 .....</b>    | <b>297</b> |
| 9.1          | 传真机的串口编程的基础知识 .....      | 298        |
| 9.1.1        | T.3 传真通信协议 .....         | 298        |
| 9.1.2        | HDLC 信息包 .....           | 299        |
| 9.1.3        | 传真字段 .....               | 300        |
| 9.1.4        | 成串信息包 .....              | 301        |
| 9.1.5        | 同步线路控制 .....             | 302        |
| 9.1.6        | 传真的 5 个阶段介绍 .....        | 303        |
| 9.2          | EIA 传真调制解调器 .....        | 305        |
| 9.2.1        | EIA 传真调制解调器 .....        | 305        |
| 9.2.2        | EIA 578(Class 1) .....   | 307        |
| 9.3          | 传真过程的实例描述 .....          | 311        |
| 9.3.1        | 单页传真 .....               | 311        |
| 9.3.2        | 轮询方式单页传真 .....           | 311        |
| 9.3.3        | G3 训练期降低速度 .....         | 316        |
| 9.3.4        | 在 G3 训练期间提高速度 .....      | 318        |
| 9.3.5        | 规格相同的多页传真 .....          | 318        |
| 9.3.6        | 不同规格的多页传真 .....          | 319        |

|               |                          |            |
|---------------|--------------------------|------------|
| 9.4           | DIS/DCS 位映像 .....        | 320        |
| 9.4.1         | 向后兼容性和可扩展性 .....         | 322        |
| 9.4.2         | 新的 FCF .....             | 323        |
| 9.4.3         | 最小性能集合 .....             | 323        |
| 9.4.4         | DIS / DCS 信息包的逐位解释 ..... | 323        |
| 9.5           | T.4 传真图像协议 .....         | 327        |
| 9.5.1         | 分辨率 .....                | 327        |
| 9.5.2         | 文件尺寸 .....               | 329        |
| 9.6           | 传真编码 .....               | 333        |
| 9.6.1         | 一维编码 (改进型哈夫曼编码) .....    | 333        |
| 9.6.2         | 二维编码 (READ 编码) .....     | 337        |
| 9.6.3         | 编码方式综述 .....             | 339        |
| 9.6.4         | 行终码 .....                | 339        |
| 9.6.5         | 页编码 .....                | 341        |
| 9.6.6         | 差错 .....                 | 342        |
| <b>第 10 章</b> | <b>串口通信的工业应用 .....</b>   | <b>344</b> |
| 10.1          | 串口通信在远程监控中的应用 .....      | 344        |
| 10.1.1        | 背景介绍 .....               | 344        |
| 10.1.2        | 远程监控系统结构 .....           | 344        |
| 10.1.3        | 程序实现 .....               | 345        |
| 10.2          | 串口通信在自动控制中的应用 .....      | 347        |
| 10.2.1        | 射频卡读写模块简介 .....          | 347        |
| 10.2.2        | 程序实现 .....               | 350        |

## 第1章 串口通信基础知识

计算机串口通信技术在实际中应用广泛，但是因为其内容十分复杂，往往让读者感到无从下手。在介绍高级的串口编程技术之前介绍一下串口通信的基础知识，将有助于读者对以后章节更加深入理解。本章介绍有关数据通信的基本概念和串口有关的硬件知识，这些内容对于开发串口通信程序是很有用的。

### 1.1 数据通信的基础知识

#### 1.1.1 数据终端设备与数据通信设备

终端与计算机之间的通信或计算机与计算机之间的通信，通常称为数据通信。数据通信技术综合了计算机技术和通信技术，一个数据通信系统的基本构成如图 1-1 所示。

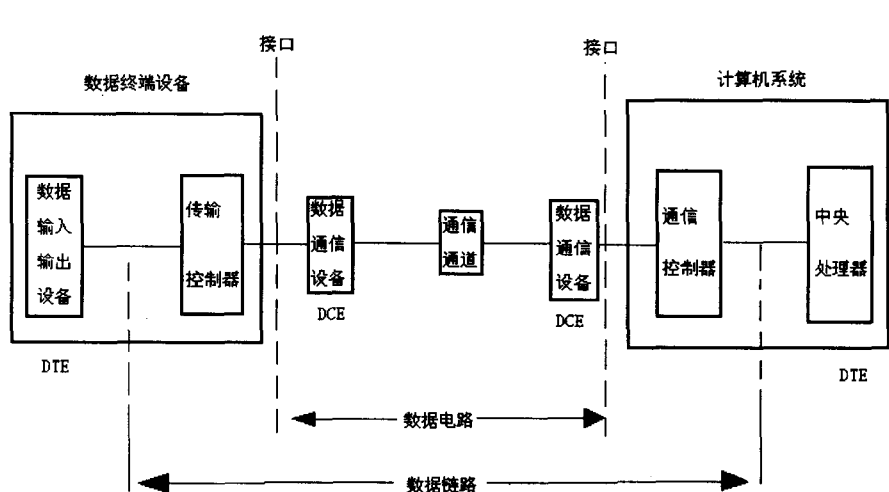


图 1-1 数据通信系统的基本构成

数据通信发送方为信源，接收方为信宿。连接信源和信宿的通道称为通信通道（简称信道）或称通信线路，它可以是电缆、电话线等。

数据通信与电话通信不同，由于在通信过程中没有人的直接参与，因而必须通过遵守相同的传输规程才可以使通信双方协调、可靠地工作。数据电路加上传输控制规程就称为数据链路。

在数据通信系统中，用于接收和发送数据的设备称为数据终端设备 DTE（Data Terminal Equipment）。DTE 既可以是一台计算机，也可以是一台只接收数据的打印机。用来连接 DTE

与数据通信网络的设备称为数据通信设备 DCE (Data Communications Equipment) 或称数据电路终接设备。DCE 可以是一个调制解调器 (Modem), 也可以是一个简单的线路驱动器。

目前最广泛使用的数据传输信道就是模拟电话线路。计算机所能处理的数字信号不能直接进入这样的信道, 而必须通过一个中间设备——Modem, Modem 是最重要的 DCE 设备之一, 用它来实现数字信号到模拟信号、模拟信号到数字信号的相互转换。

### 1.1.2 字符传输

为了掌握数据传输的知识, 首先必须清楚数据在计算机中是如何存储的。即数据在计算机中的存放格式。

#### 1. 位和字节

在十进制计数制中有 10 个数码 0~9, 在二进制计数制中只有两个数字 0、1。二进制计数制中的每一位 0 和 1, 被叫做一个位。每 8 个位构成一个字节, 因此一个字节值的范围从 00000000~11111111, 也就是十进制的 0~255。

在一个字节中最右面的位被称为位 0 位, 最左面的位被称为位 7 位。位 0 位是最低有效位, 而位 7 位是最高有效位。

#### 2. 文本编码

差不多所有的计算机都工作在二进制方式下, 因为二进制的代码 0 和 1 易用正电压或负电压表示。在大多数计算机中, 存储器存储的最小单位是字节, 信息在计算机中进行存储和操作, 都要转换成字节序列。

当文本 (字母字符、标点符号等) 在计算机中存储时, 每个不同的字符都用不同的数值来表示。这些数值的范围通常为 0~127 或者为 0~255。由于一个字节能表示一个 0~255 之间的确定的值, 因此在文本数据中每一个字母或标点符号都自然地分配了一个字节的空间来存储。

一般计算机上使用的都是 ASCII 码, 它利用数值 32~126 代表数字、字母、标点符号和其他通用字符 (数值 0~31 以及 127 都有特殊含义)。因为 ASCII 码在计算机里只使用了一个字节的 7 个位, 一共有 127 种数值组合, 所以在计算机里所有字符都是用 0~127 之间的数值来表示的, 另外每个字节里留了一个备用位。

很多计算机的字母编码使用了字节中的全部 8 个位, 这样共给出了 256 种不同的组合。其中前 128 个遵循 ASCII 规则, 其余的 128 个用来做扩展字符、数字符号、图形字符等编码。但是, 这些扩展的字符目前还没有统一的标准 (例如中文编码), 在不同的计算机中有不同的含义。

#### 3. 非文本编码

当然, 并不是所有的信息都以文本形式在计算机中存储, 某些可执行指令文件和图形图像文件就是以二进制形式而不是 ASCII 码形式存储的。这些数据通常是使用一个字节所具有的 256 种可能组合来编码的。一个数值可用二进制形式存储, 可以占多个字节。在通信领域, 常常把这种类型的资料叫做二进制数据。

因为非文本数据可以使用一个字节的任何位组合值, 所以非文本数据中的某些值可能与 ASCII 编码表中有特殊意义的码值是一致的。例如在传输数据的过程中, 接收设备正好接收到一个非文本字节数据, 而将其解释为信息结束的码值, 那可能会引起混乱, 在这种情况下

这种数据不能以它们的原型传送。

当然，针对这个问题，成熟的协议已被设计出来了，它们将在后面的章节中进行讨论。

#### 4. 并行传送和串行传送

并行传送意味着当数据（例如一个字节）从一台计算机传送到另外一台计算机时，不是依次一个比特位一个比特位地传送，而是用并行方式沿多线同时传送多个位，传送的并行的位数可能是不同的，但一般是8位或8位的倍数。

串行传送是指数据被依次一位一位地传送，因此通信接口必须能够分别将所有位依次一个地发送，也应能够将数据一位一位地接收下来。

#### 5. 同步和异步通信

将数据转换成串行通信有两种方式：同步方式或异步方式。

在串口通信中，异步通信方式以字符为传送单位，传送的字符之间有无规律的间隔，这样可能使接收设备不能正确接收数据，因为每接收完一个字符之后都不能确切地知道下一个将被接收的字符将从何时开始。这就需要在每个字符的头、尾各附加一个比特位，用来指示一个字符的开始和结束。这些额外的位被称为开始比特位和停止比特位，有时也经常加上一个称为奇校验的错误校验位。

当被发送字符以块为单位时，则它们能被有规则的分割开来。它不再需要在每个字符的首尾附加开始和停止比特位，因为一旦收到首字符时，接收设备能确切地预计接收后续字符的位置。也就是说，接收设备能自动与发送计算机同步。这种通信方法被叫做同步通信。

因为异步通信需要在每个字符的首尾附加开始和停止比特位，和同步通信相比，异步通信每传送一个文件要多花费时间。

#### 6. 帧

在串行异步通信情况下，构成一个字节的那些数据单元被称为数据位，在数据位的前、后要加上起始位、停止位和奇偶校验位。

一个字符所使用的位数根据不同协议而不同。称这些位数为字符位数据长或字长。通常不是8位就是7位。每发送一个字符，就发送一个组，这个组包括一个起始位、数据位、可选的奇偶校验位、一个或更多个停止位。为了清楚起见，将构成每个字符及伴随的位作为一个帧。两个帧传输的例子如图1-2所示。

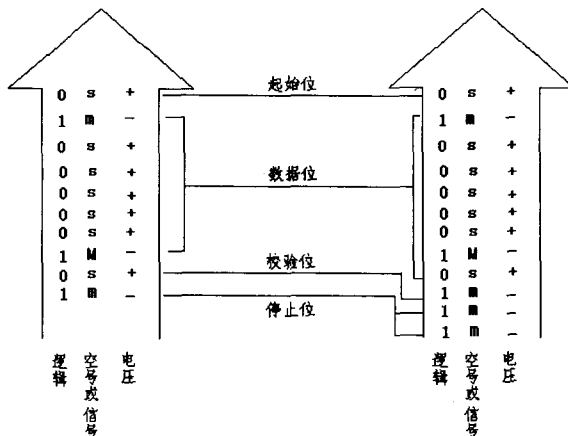


图 1-2 两个帧传输的示例

## 7. 起始位

起始位总是加在帧开始，为的是提醒接收设备接收数据，起到同步作用，在接收数据位过程中将被分离出去。1 个起始位是一个 SPACE（或二进制 0）。在直接连接情况下，一个 SPACE（或二进制 0）被当作一个正电压传输，两帧之间是负电压，因此，在每一帧开始，电压从正电压向负电压转变。

## 8. 数据位

串行通信标准，也叫串行通信协议，允许传输字符或字的长度不同。如果通信软件要求选择字长度时，要么选择 7 位字符，要么选择 8 位字符（有时也选择其他字长，但非常少见）。如果所有的数据都用 ASCII 方式传输，7 位字符是最充分的。不要忘记 ASCII 码表只分配 0~127 个数，所有这些字符能够用 7 位的字节来表示。

如果要传输非 ASCII 数据（假如使用扩展字符设置的文本或二进制数据），需要用到一个字节的的所有 8 位。除非将该数据转换成 7 位形式，否则不能使用 7 位协议。

数据位被传输时从一个字符的最低位数据开始，最高位数据在最后。例如字母 C 在 ASCII 表中是十进制的 67，二进制的 01000011，那么传输的将是 11000010（或者是 7 比特编码的 1100001）。

## 9. 奇偶校验位

奇偶校验是检验正在传输的数据是否被正确接收的一种方法。发送设备将根据数据位的内容附加一个 0 或 1。接收设备检查此奇偶校验位是否还保持与其他位的正确关系。如果不保持正确关系，那么在传输过程中可能产生了错误。奇偶校验有如下几种方法：

偶校验，意指数据位中 1 的个数加上奇偶校验位（1 或 0），使 1 的总数为偶数。例如字母 A 的二进制是 01000001，其中 1 的个数为 2 是偶数，则奇偶校验位应为 0，使总的 1 的个数保持偶数。如果收到字母 A 的二进制序列中 1 的个数不是偶数个，那么就意味着在传输中产生了错误。

奇校验，意指数据位中 1 的个数加上奇偶校验位，使其 1 的个数总数为奇数。例如字母 A，为保持 1 的个数为奇数，则奇偶校验位应为 1，即 1 的总数为 3，奇数。

无校验，是指没有奇偶校验位。一个奇偶校验位并不总是被使用的，甚至即使已经使用奇偶校验位，校验接收设备也并不去检测奇偶校验位，这个完全依赖通信编程来控制。

SPACE 校验，（或零校验，有时称为位平衡）有一个奇偶校验检验位，但它总是 0。SPACE 校验提供一些错误类型检验方法，如果严格筛选出的奇偶位在传输过程中变成 1，这就提示出了某类的错误。

SPACE 校验能用在要求传输 8 位字符的接收设备上发送使用 7 位字符。接收数据将数据字的最后一个比特位解释为奇偶校验位。如果所有的字符以标准的 ASCII 码形式发送，则第 8 位永远不使用，所以在那个位置上的 0 就是无意义的。

MARK 校验的工作方法和 SPACE 校验的方法相同，指示校验位总是 1。因为在那个位置上的 1 可以被解释为一个附加校验位，设备或接收计算机必须依靠编程来忽视它。

## 10. 停止位

在每个帧的末尾总要发送一个停止位。这个停止位可以是 1 位、1.5 位或 2 位。最常用的是 1 位。这就保证了在每一帧之后都是负电位，而下一帧可以通过开始位的正电位来识别。超过一位的停止位通常用于这样的场合：在处理下一个即将发送来的字符之前接收设备要求

附加时间。

1 个半位的停止位在时间上要比常规长许多，这样在两机之间停止位增加了间隔。它们被当作二进制的 1 来传送，在直接连接的情况下，等于一个负电位。

2 个停止位通常使用在 110 波特低速传输系统中，这是一个要求连续的老式的电报终端，它使用低波特率并且要求额外字符处理时间。

### 11. 硬件中断 (Break) 信号

硬件中断信号是当一种特殊的帧出错（一般是通信线路自身的故障）时，接收方发送的一个信号；硬件中断信号可以用于系统的复位、振铃或者中断一个进程，该信号提示系统忽略通信线上传过来的代码数据。它是一种主要的通信信号，而不仅是一种故障标记。

标准情况下，数据线在传输时通常处在正电压状态，数据线上任何一个 SPACE 状态都要比一个字符和它的帧传递持续更长的时间。最长 SPACE 状态应当为 1/15s，即 66.67ms；因而可以使用一个持续较长的时间间隔（通常是 100 到 600ms）的状态信号用来作为硬件中断 (Break) 信号。

硬件中断信号也可以用在主机上，等效与 PC 机中的“Ctrl+C”。它可以中断正在运行的程序并使用户回到操作系统或者进入程序某段的上一级中去。硬件中断 (Break) 信号和“Ctrl+C”一样，常用来从正在运行的死循环程序中退出来。

### 12. 波特率

波特率是每秒传送的离散信号数，它是用法国通信学科开创者波特的名字命名的。对于二进制通信来说，就是指每秒传输的二进制位数。波特和位有很大的区别，但是常常被弄错。

在两个 RS-232 直接连接时，一个信号在任何时间里只使用两种状态的一种，所以这时的波特率和位率是相同的概念。无论如何，当一个信号在两个调制解调器之间直接传送时，可以用多个状态中的一个。信号的长度可以是 1/600s（600 波特），但是由于一个被传送的状态可能携带超过 2 位信息，则位率将比波特率高。要注意无论是波特率还是涉及到在一个信号帧内传送信息的位速率，因为字符是使用不同速度输入的，两帧之间的间隔是可以变化的，所以，所提到的无论波特率还是位速率都是指真正被传送的信息。

### 13. 故障排除

当两个设备互相通信时，即使使用相同的速率、字长、停止位和奇偶校验位等，也一定要对他们进行设置。如果接收设备什么也没有收到，那有可能是物理连接的错误（数据线断了或者接错了）。如果没有正确接收到握手信号，错误可能由下面一些原因引起的。

#### (1) 波特率失配

如果两个设备的波特率设置不同，当接收设备试图接收数据时，程序将报告校验错和帧错。接收字符数和发送字符数不相符是判断此类错误的最简单的方法。

#### (2) 校验错误

一个奇偶错误，严格地说是指数据在传输中被破坏，至少可以说明设备在奇偶校验位类型设置不同或者字长不同。

#### (3) 字长不匹配

如果发送的是 8 位字长，接收采用 7 位字长，这个区别在文本传输时可能不会引起注意，因为前 7 位在任何情况下都是有效的。这主要是因为位 0 位首先被发送，而位 7 位在以 ASCII 码形式的传输中本来就忽略不用，所以就显得不那么重要。但接收设备试图将附加位作为校

验位来解释，并且要报告一个错误。所以发生一个校验错，如果不是数据在传输过程中被破坏了，就是指示字长不匹配。

如果发送的是 7 位字长，而希望接收的是 8 位字长，那么可能校验位被处理成缺少的第 8 个位。通常奇偶校验位不是 1，就是 0，这时将发现接收设备显示的是扩展编码字符。

#### (4) 停止位错

如果接收端要求一个停止位，而发送端发送了两个停止位，这可能不会有什么问题。这个多余附加位，可能会被简单地合并成两个字符之间允许的分割字符。但如果要求两个停止位，而只发送一个停止位，这可能会出现什么问题，这是由接收设备决定的。如果使用调制解调器，这可能不被认为是个问题。

#### (5) 帧错

一个帧的错误，是指位数不匹配。这个类型的错误，通常是在没有接收到要求的停止位时出现。

### 1.1.3 握手和缓冲存储器

握手信号提供了一种控制数据流的方法，即接收设备可以控制发送设备的数据发送。有的打印机打印速度可能跟不上从计算机那里接收数据的速度，这时打印机必须使用握手信号去通知计算机暂停发送。握手信号也用在打印机没有打印纸和一台计算机向另一台计算机发送数据而另一台计算机的字符处理速度赶不上发送速度的时候。

当接收设备字符处理速度比字符发送速度快时，握手信号可以忽略。

#### 1. 硬件握手

图 1-3 显示了一个使用硬件握手的计算机向打印机传送数据的典型流程图。

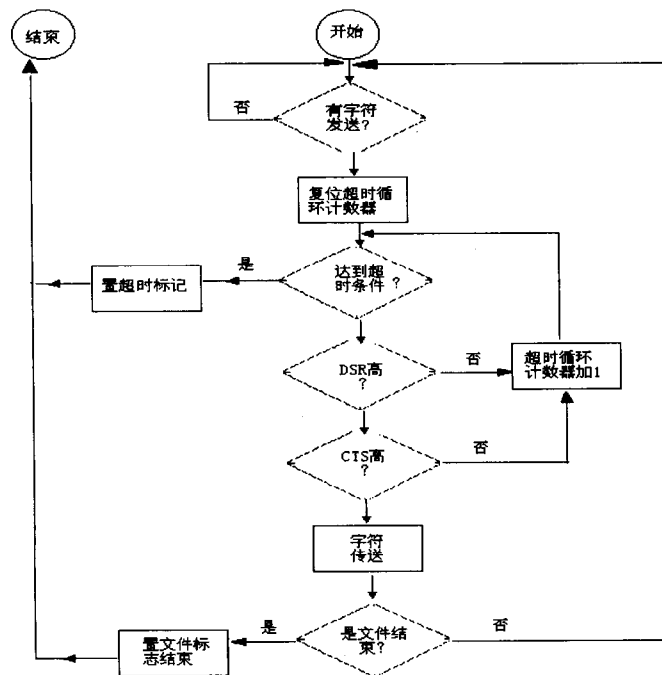


图 1-3 发送设备的硬件握手流程图

硬件握手是使用专门的握手电路去控制数据的传输的。DCE 通常使用 DSR（数据设备就绪信号线）作为一个主握手线去通知 DTE 它已具备接收数据的能力并且已准备好。它们也可以使用 CTS（清除发送信号线）作为辅助握手线。另一方面，DTE 设备使用 DTR（数据终端就绪信号线）作为主握手线去通知 DCE 它已经做好接收数据的准备，也可以将 RTS（请求发送信号线）作为一个辅助握手线。

根据惯例，传输线被使用时这些握手线上具有正电压，当需要发送挂起时这些握手线变为负电压。

例如，一个作为 DTE 设备的串行打印机，当其准备好接收字符时，将设置 DTR 线为高电位；当它想要挂起传输时，将设置 DTR 线为低电位。它也可以使用 RTS 作为一个辅助握手线。许多计算机可以用编程来控制其停止发送数据，直到 DSR 和 CTS 变为高电位。

## 2. 软件握手

软件握手信号是在数据线（TxD 和 RxD）上进行传送的，用来代替硬件握手中在专门握手线路上传输的信号。这种方法用在直接连接或通过调制解调器连接的两台计算机相互进行双向通信的场合。

对软件握手人们已经建立了一些标准协议，最通常的是 XON/XOFF 协议和 ETX/ACK。

### （1）XON/XOFF 协议

在 XON/XOFF 协议下，要想使发送设备停止发送，接收设备只需发送一个 ASCII 字符 DC3（十进制是 19，十六进制是 13）给发送设备。

如果想要发送设备继续发送数据，则接收设备需发送一个 ASCII 字符 DC1（十进制 17，十六进制 11）给发送设备。

在一般实际应用中，使用一个缓冲存储器。当缓冲存储器装满时，字符 DC3 发送给发送方；当缓冲存储器快要空时，则向发送方发送一个 DC1 字符（缓冲存储器将在后面介绍）。

图 1-4 示出使用 XON/XOFF 协议的从计算机向打印机传送数据的典型流程图。

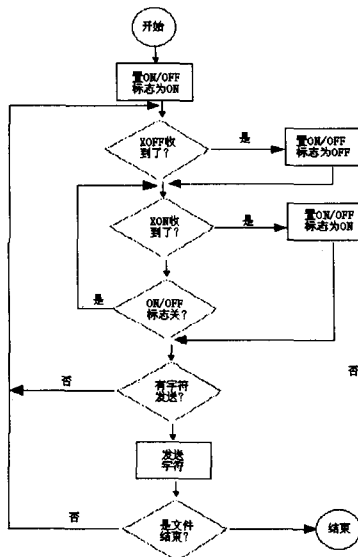


图 1-4 使用 XON/XOFF 协议的从计算机向打印机传送数据的典型流程图

### （2）ETX/ACK（结束/响应）

在 ETX/ACK 方式中, 数据被分批发送, 每批长度固定不变。发送完之后发送设备接着发送一个 EXT 字符 ASCII 3。接收端要对发送端做出相应的响应, 如果正确地接收了字符就回答一个 ACK 信号 (ASCII 6); 如果接收端检测出错误, 则回答一个 NAK 信号 (ASCII 21)。

### 3. 硬件和软件结合的握手方法

假设使用 PC 机或终端通过调制解调器和主计算机进行通信。调制解调器使用硬件握手方法和 PC 机联系, 同样主计算机和调制解调器之间也使用硬件握手方法。但主计算机和 PC 机之间将使用软件握手方法进行联系。

因此, PC 机必须要编程, 仅当调制解调器的 DSR (有时可能使用 CD) 线电位变高和软件停止信号没有收到时, 软件才送出一个通知信号。

无论如何, 一些计算机总是自动地监视硬件握手信号。这些计算机等待硬件握手信号变高以后送出一个字符, 所以程序只需要处理软件握手信号即可。

### 4. 缓冲存储器

缓冲存储器是用来放置接收字符和发送字符的区域存储器。使用缓冲存储器能够减少必须发送握手信号的数目, 因为分成较大数据传输所需的握手信号的数目要少于一个一个字符传输所需的信号数目。

#### (1) 输入缓冲存储器

当接收设备接收字符的速度比处理字符的速度快时, 应使用输入缓冲存储器。例如, 一台打印机能以 1200 波特率接收数据, 而它只能等效 300 波特率打印字符。与其打印机在每个字符后都送给计算机一个握手信号直到打印字符处理完毕, 还不如设计者在打印机里设置一个缓冲存储器, 使它具有有一定存储容量来存放一定数量的字符。这样的区域存储器叫做输入缓冲存储器。可以想象这个缓冲存储器是一个水箱, 水箱可以在顶口将其填满水, 在底端可以将其放空。在这个缓冲存储器没有完全满之前, 一个停止信号将被送出。当这个缓冲存储器未完全空之前, 一个重新开始发送的信号将被送出。如果打印机等待到缓冲存储器完全填满以后才发出停止信号, 以及到缓冲存储器只有空时才发出重新开始的信号, 那么当第一次填满缓冲存储器以后缓冲存储器将很快就不起作用了。从那时以后, 每收到一个字符都要发出停止信号, 而每处理一个字符以后都要发出重新开始的信号, 全然和没有缓冲存储器一样。

在缓冲存储器完全填满以前发送停止信号的另一个原因是防止字符丢失, 因为最后传送的字符会同停止信号同时接收。

硬件握手信号实际上是一个停止信号, 通常它能立即引起发送设备的传输挂起。使用软件握手, 无论如何, 在停止命令生效以前会产生一个延时, 因为停止命令还需要发送到计算机处理, 这时一些字符可以被送出。

图 1-5 到图 1-7 说明打印机缓冲存储器填充和到空的不同阶段。图 1-5 说明缓冲存储器只填充了一半, 这个阶段一边从计算机接收字符, 一边处理打印字符。图 1-6 表示缓冲存储器几乎被填满 (还未填满), 这时打印机要求计算机停止发送数据, 这是由打印机释放握手信号或者送出一个 XOFF 完成的。图 1-7 表示缓冲存储器几乎空 (还没有完全空) 时, 打印机要求计算机恢复字符发送, 这是由打印机抬起握手线或者送出一个 XON 实现的。