

串行通信开发指南

(第二版)



万水计算机技术实用大全系列

[美] Mark Nelson 著

潇湘工作室 译

C++ 程序员
的优秀指南



- 创建能够在不同的操作系统和硬件平台之间轻松移植的串行通信程序。
- 学习新的面向对象技术，为调制解调器和其他通信系统编写软件。
- 本书配套光盘中有建立串行通信的程序所需要的 C++ 类。



中国水利水电出版社
www.waterpub.com.cn



00112737



万水计算机技术实用大全系列

串行通信开发指南

(第二版)

[美] Mark Nelson 著

潇湘工作室 译

中国水利水电出版社



JS42-25

内 容 提 要

本书全面而完整地介绍在 32 位 Windows 平台上进行串行通信编程。其主要内容有串行通信的基础知识、Windows API 的技巧、在高速调制解调器上达到最大数据交换速率的方法以及面向对象编程的最新技术。通过学习本书,读者将能够掌握以下知识:

(1) 编写可以很容易地在不同的操作系统和硬件平台之间进行移植的通信程序;(2) 跨越网络访问调制解调器和其他串行设备;(3) 使用标准和智能多端口板;(4) 实现文件传输格式,如 XMODEM、YMODEM、ZMODEM 和 Kermit;(5) 以 57.6Kbps 和更高速度与最新的高速调制解调器联系;(6) 编写在各种系统上工作的终端仿真代码;(7) 利用 Windows 95、98 和 NT 通信 API 编写通信程序;(8) 使用 TAPI 并掌握调制解调器的设置。另外,本书还提供了丰富的程序代码,在本书光盘中,有这些代码的源程序。

本书适用于利用 C++ 进行通信编程的程序员。

"Original English language edition Copyright © 1999 by IDG Books Worldwide, Inc. All rights reserved including the right of reproduction in whole or in part in any form. This translation published by arrangement with IDG Books Worldwide, Inc."

"The IDG Books Worldwide logo is a trademark or registered trademark in the United States and/or other countries under exclusive license to IDG Books Worldwide, Inc., from International Data Group, Inc. The Developer's Guide trade dress is a trademark of IDG Books Worldwide, Inc. in the United States and/or other countries. Used by permission."

北京市版权局著作权登记号: 图字 01-2000-0436 号

书 名	串行通信开发指南 (第二版)
作 者	[美]Mark Nelson 著
译 者	潇湘工作室
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: sale@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092 毫米 16 开本 39 印张 928 千字
版 次	2000 年 9 月第一版 2001 年 3 月北京第二次印刷
印 数	4001—7000 册
定 价	80.00 元 (1CD, 含配套书)

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

前 言

欢迎阅读本书第二版。本书的目的是教会程序员使用 C++ 来开发灵活的通信程序。本书的示例代码面向 IBM PC 机，但根据书中的说明，可以简单地将其移植到其他类型的计算机上。

过去，进行 RS-232 硬件编程的 C 程序员发现，他们受到具体硬件的极大限制。即使软件可以在所给的环境中正常运行，但只要几个因素中有一个有所变化，例如硬件平台、操作系统、C 编译器或者所用串口的类型，都需要大量重新编写程序。本书将说明如何利用 C++ 的优势来避免串行编程中的限制。

本书将着重介绍 RS-232 端口的编程，在此并不详细讨论 RS-232 的硬件结构、应用软件、BBS 程序等问题，这些内容在其他有关 RS-232 的书都有介绍。本书将讨论一些其他书中没有涉及到的问题，即开发软件使计算机与 RS-232 硬件进行通信的复杂过程。

本书的目的

RS-232 目前仍是广泛使用的实现计算机间相互通信的方法之一。几乎每台台式机都有一两个串行接口，用来与调制解调器、实验室设备、POS 终端等进行数据传输。

尽管 RS-232 使用广泛，但在编程方面却有一些小问题。编程接口几乎没有什么文档说明，不存在设备驱动程序，或使用驱动程序非常复杂。试图在原 IBM PC 机上正确使用串行通信的程序员称这类编程为“黑色艺术”。

更糟的是，串行通信代码与硬件平台是密切相关的。适应 Macintosh API 环境的代码不能方便地移植到 Win32 系统中。尽管 Excel 和 Word 这样的程序可以在各种平台之间移植，但串行程序通常却局限于特定的硬件。

这种问题再也不存在了！使用一些精心设计的类，只要简单地不同的驱动程序中链接，就可以使程序在不同的平台间进行移植。本书将介绍如何完成这类工作。

本版新内容

本书第一版于 1992 年完成，当时 Windows NT 刚刚出现，Windows 95 也尚未问世。事实上，Windows 3.0 也只是刚开始在台式机上流行起来。当时，Win32 通信 API 没有普遍的意义。根据当时的情况，本书第一版用一章的篇幅介绍了 Win16 API，而没有提到 32 位版本。

如今，Windows 32 位版本在全世界的台式机上得到了广泛的应用，程序员们就有必要知道如何支持这类平台了。因此，本书第二版中的所有新内容都与 32 位版本的 Windows

有关。新增内容如下：

- ◆ 在第十章中详细讨论 Win32 通信 API。
- ◆ 在第十一章中讲述有用的而且功能完备的 Win32 端口类的开发。
- ◆ 在第十二章中介绍 Win32 的综合测试程序。
- ◆ 在第十三章中讨论一系列执行 Win32 终端仿真的类。
- ◆ 整个十五章将讨论 TAPI (Windows 电话 API, Windows Telephony API)。使用 TAPI 可以更方便地在程序中加入调制解调器支持。

必备知识

本书不适于初级程序员。为了有效地使用代码，要求读者能熟练编写 C 程序，最好有编写 C++ 程序的经验。本书代码并没有完全使用 C++ 的特征，读者只要了解类的概念和虚拟函数就可以了。

本书不要求读者对 RS-232 有很多的了解。每一章的内容都是建立在前一章知识的基础上的。另外，本书假设读者都是初学者，前几章将比较详细地回顾一些基本概念。如果读者不愿阅读或者确有这方面的实际经验，则完全可以跳过前几章的内容。

必备条件

读者应能够编译本书的 C++ 代码。很多代码是 16 位代码，可以在 MS-DOS 或 16 位 Windows 下运行。这些代码可在 Microsoft Visual C++ 1.52 或 Borland C++ 4.5 及其先前版本中编译。这两种 16 位编译器都是制造商限定的。

本书中的 Win32 代码已在 Visual C++ 5.0 中编译通过。读者应有 Visual C++ 5.0 或更新的版本。

本书结构

组织内容的方法有很多种。本书不是把各个章节硬性分成几个部分，而是让每一章都建立在前一章的基础上。然而，本书将涉及如下所示的六个主题。在本书的最后还有一个附录，解释本书光盘内容，另外还有一个术语表。

第一章至第二章：基本知识

第一章详细概括有关 RS-232 的知识。从中可了解到 RS-232 的标准、RS-232 的典型实现及其使用方法。这里讲述的具体内容有调试工具、制作电缆的设备等。

第二章介绍 RS232 类，这是与本书所有硬件接口有关的基本类。开发出通用的而且易于理解的 RS232 类是非常重要的，这将直接影响到本书中所有由此派生的类。

第三章至第七章：16位DOS接口

这些章节中将讨论在 16 位 DOS 操作系统中广泛使用的、用于串行通信的软硬件。包括能直接与标准 PC 机通信端口对话的类，以及与 PC 机 BIOS、多端口板、FOSSIL 驱动程序连接的类。

第八章至第十三章：终端仿真

终端仿真器给用户提供更便捷的途径，通过 RS-232 和其他的计算机相连。在这些章节中将讨论两种不同的终端仿真例程：DOS 和 Windows。这两种程序都使用本书中创建的标准 RS-232 类。DOS 和 Windows 终端仿真程序看上去截然不同，与 PC 机硬件连接的机制也是完全不同的。然而，它们都使用共享代码的同一部分与 RS-232 端口进行对话，并执行终端仿真内部函数。

第九章至第十二章：Windows 类

第九章至第十二章将涉及 16 位和 32 位操作系统的串行编程。除了这两种操作系统的基本特点，它们的串行接口驱动程序也是完全不同的。32 位驱动程序的使用比较复杂，该部分所讨论的包装类的使用是非常有帮助的。

第九章中将讲述 Win16 API 和相应的类。Win32 API 和包装类相对而言更复杂，这些内容将在第十章、第十一章、第十二章中阐述。

第十四章和第十五章：使用调制解调器

这两章提供两种不同的方法使程序和标准的智能调制解调器连接。第十四章讲述 MS-DOS 下的硬件方法，即仔细定义调制解调器能理解的 AT 命令，并在程序中恰到好处地使用这些命令。

第十五章将讲述如何使用 TAPI 来控制调制解调器，在这种方法下，Windows 了解所有关于调制解调器连接的重要信息。

第十六章：文件传输协议

在最后一章中，将阐述如何运用本书中其余部分开发的类来编写文件传输协议。本章将给出实现流行的 ZMODEM 协议的示例。

本书光盘

本书中所有的源代码都包含在本书光盘中。这些代码可以生成本书中的所有示例。要了解更多有关本书光盘的内容，请参看附录。

其他

出版商和作者都希望得到读者的反馈。请读者阅读本书后，在网上进行注册，网址为 <http://my2cents>（详情请见本书后面的 my2cents 页）。请您忠实地进行评价。您的反馈可以让作者知道如何更好地满足您的需求。

欢迎针对本书内容提出具体的问题。作者将尽力回答这些问题，但不能保证对每一个问题都作出答复。您最好通过电子邮件同作者联系，作者的电子邮件地址为：

`markn@ieee.org`

此外，欢迎访问作者的 WWW 主页，该网页中包含有更多的信息。尽管作者尽力使本书做到准确无误，但是难免会有一些疏漏。作者的网页提供了一个错误列表。该网址为：<http://WWW.dogma.net/markn/>。

致 谢

没有一本书是只通过一个人的努力完成的，本书也不例外。作者衷心感谢 IDG 出版社的编辑人员：Greg Croy、Barbra Guerra、Susan Christophersen、Terri Varveris、Mildred Sanchez、Richard H.Adin。感谢他们为本书付出的辛勤劳动。

同时还要感谢我的家人。Denise、Kaitlin 和 Joey 一直以来理解并宽容由于本书的撰写而带来的生活中的不便。

最后，还要感谢第一版的读者。由于你们的意见、建议和鼓励，本书才得以完成。

目 录

前言

致谢

第一章 RS-232 概述.....	1
1.1 RS-232 标准.....	1
1.2 标准省略的结果.....	2
1.3 ASCII 码——另一种 RS-232-C 吗.....	2
1.4 ASCII 码的问题.....	3
1.5 DTE 和 DCE.....	4
1.6 RS-232: 物理接口.....	4
1.6.1 信号格式.....	6
1.6.2 数据线.....	7
1.6.3 标记和间隔.....	8
1.6.4 控制线.....	8
1.6.5 电线.....	9
1.7 RS-232-C 规范的限制.....	9
1.8 电缆.....	10
1.8.1 空调制解调器电缆.....	10
1.8.2 更合理的电缆设计.....	12
1.8.3 RS-232 9 到 25 针适配器.....	13
1.8.4 自制电缆.....	14
1.8.5 Macintosh 电缆.....	14
1.8.6 使用电缆的建议.....	14
1.9 UART.....	15
1.9.1 8250 系列产品.....	15
1.9.2 UART 的发展.....	15
1.9.3 8250 的功能.....	16
1.9.4 控制寄存器.....	17
1.9.5 状态寄存器.....	17
1.9.6 中断功能.....	17
1.9.7 16550 的不同点.....	17
1.9.8 16550 的子代产品.....	18
1.10 多端口板.....	18

1.11	智能多端口板	19
1.12	调制解调器	20
1.12.1	数据速率	21
1.12.2	智能	22
1.12.3	握手	24
1.13	流控制	24
1.13.1	硬件流控制	24
1.13.2	软件流控制	25
1.13.3	本地流控制与直通流控制	26
1.14	文件传输	26
1.15	软件和硬件工具	28
1.15.1	PC 机软件	28
1.15.2	BBS 软件	28
1.15.3	编程库	29
1.15.4	商业化工具	29
1.15.5	电缆制作设备	31
1.16	什么是 Internet	33
1.17	小结	34
第二章	RS232 类	35
2.1	为什么使用 C++	35
2.1.1	试用 C 语言	35
2.1.2	分析方法的弊端	36
2.1.3	用函数指针调度	36
2.1.4	将函数指针组成结构	37
2.1.5	为什么使用 C++	38
2.2	RS232 类	39
2.2.1	一些约定	41
2.2.2	默认参数	43
2.2.3	强制函数和可选函数	44
2.2.4	哪里是构造函数	44
2.3	RS232 类的各个成员	44
2.3.1	保护成员	45
2.3.2	保护成员函数	47
2.3.3	公共数据成员	47
2.3.4	强制虚拟函数	48
2.3.5	非虚拟函数	51
2.3.6	短小的例程	54

2.3.7 可选函数	55
2.4 代码	61
2.5 小结	74
第三章 PC8250 类	75
3.1 8250 UART	75
3.1.1 8250 寄存器集	76
3.1.2 8250 的兼容芯片	85
3.1.3 8250 设计的扩展	87
3.1.4 8250 的特别之处	87
3.2 标准 COM 板卡	89
3.2.1 便携机的特点	89
3.2.2 PC 机的中断方式	90
3.2.3 硬件基础	90
3.3 PC8250 驱动程序结构	91
3.3.1 中断服务例程	92
3.3.2 Isr_data 结构	95
3.3.3 ISR 程序代码	97
3.3.4 PC8250.CPP	105
3.4 支持类	132
3.4.1 队列软件包	132
3.4.2 中断管理程序软件包	135
3.5 测试程序	142
3.6 小结	144
第四章 共享中断设备	145
4.1 多端口板	145
4.1.1 中断线共享-状态寄存器	146
4.1.2 选择 IRQ 线和地址	146
4.2 Handler 类	147
4.3 Digi Classic 板中断处理程序	148
4.3.1 构造函数	149
4.3.2 析构函数	151
4.4 CHAPT04.CPP	152
4.4.1 头文件总结	152
4.4.2 变量和常量	152
4.4.3 加入标准处理程序	153
4.4.4 打开端口和窗口	153
4.4.5 建立并运行程序	154

4.5	MS-DOS 下的多任务	155
4.6	小结	156
第五章	智能多端口板	157
5.1	硬件	157
5.1.1	自由设计	158
5.1.2	程序控制	158
5.2	软件接口	159
5.3	Digi International API	160
5.3.1	函数 0: 初始化端口 (BIOS 兼容)	161
5.3.2	函数 1: 输出字符 (BIOS 兼容)	162
5.3.3	函数 2: 输入字符 (BIOS 兼容)	162
5.3.4	函数 3: 读入线和调制解调器的状态 (BIOS 兼容)	162
5.3.5	函数 4: 初始化扩展端口 (EBIOS 兼容)	163
5.3.6	函数 5: 扩展端口控制 (EBIOS 兼容)	164
5.3.7	函数 6, 子函数 0: 获得端口名	164
5.3.8	函数 6, 子函数 1: 获得驱动程序信息	164
5.3.9	函数 6, 子函数 2: 获得板卡信息	165
5.3.10	函数 6, 子函数 0xff: 获得驱动程序名	165
5.3.11	函数 7: 发送中断	166
5.3.12	函数 8: 改变状态检查	166
5.3.13	函数 9: 清除端口缓冲区	166
5.3.14	函数 0x0A: 输入缓冲区记数	167
5.3.15	函数 0x0B: 去掉端口的握手线	167
5.3.16	函数 0x0C: 获得端口参数	167
5.3.17	函数 0x0D: 获得字符就绪标志的指针	168
5.3.18	函数 0x0E: 写缓冲区	169
5.3.19	函数 0x0F: 读缓冲区	169
5.3.20	函数 0x10: 清除 RX 缓冲区	169
5.3.21	函数 0x11: 清除 TX 缓冲区	169
5.3.22	函数 0x12: 释放 TX 缓冲区的空间	170
5.3.23	函数 0x13: 提高端口的握手线	170
5.3.24	函数 0x14: 查看字符	170
5.3.25	函数 0x15: 获得 RX 缓冲区中的空间	170
5.3.26	函数 0x1B: 获得缓冲区大小和水位标志	171
5.3.27	函数 0x1C: 设置握手水标志	171
5.3.28	函数 0x1E: 设置握手	171
5.3.29	函数 0x20: 打开/关闭 BIOS 调步	172

5.3.30 函数 0xFD: 获得缓冲区的记数	172
5.4 INT14 驱动程序函数小结	172
5.5 配置智能板	173
5.6 配置端口	175
5.7 实现 DigiBoard 类	175
5.8 代码	176
5.9 CHAPT05.EXE	190
5.10 小结	191
第六章 BIOS 和 EBIOS 类	192
6.1 BIOS 细节	192
6.1.1 函数 0: 初始化端口 (BIOS)	193
6.1.2 函数 1: 输出字符 (BIOS)	193
6.1.3 函数 2: 输入字符 (BIOS)	194
6.1.4 函数 3: 读入线路和调制解调器状态 (BIOS)	194
6.1.5 函数 4: 扩充的端口初始化 (EBIOS)	194
6.1.6 函数 5, 子函数 0: 读取调制解调器控制寄存器 (EBIOS)	195
6.1.7 函数 5, 子函数 1: 写入调制解调器控制寄存器 (EBIOS)	195
6.2 问题	196
6.3 代码	196
6.4 继承性	197
6.5 测试 BIOS 类	211
6.6 小结	213
第 7 章 FOSSIL 接口	214
7.1 历史	214
7.2 FOSSIL 技术要求说明	215
7.2.1 函数 0: 初始化端口 (BIOS)	215
7.2.2 函数 1: 传送单一字节 (BIOS)	216
7.2.3 函数 2: 获取接收的字符 (BIOS)	216
7.2.4 函数 3: 读取状态寄存器 (BIOS)	216
7.2.5 函数 4: 打开串口	217
7.2.6 函数 5: 关闭端口	217
7.2.7 函数 6: 控制 DTR	217
7.2.8 函数 8: 刷新 TX 缓冲区	218
7.2.9 函数 9: 清除输入缓冲区	218
7.2.10 函数 0x0A: 清除输出缓冲区	218
7.2.11 函数 0x0B: 立即发送	218

7.2.12	函数 0x0C: 查看单字符	219
7.2.13	函数 0x0F: 选择流控制	219
7.2.14	函数 0x18: 读取缓冲区	219
7.2.15	函数 0x19: 写入缓冲区	220
7.2.16	函数 0x1A: 中断控制	220
7.2.17	函数 0x1B: 获取 FOSSIL 驱动程序信息	220
7.3	信号源	221
7.4	源代码	221
7.5	编译 CHAPT07.EXE	231
7.6	测试运行	232
7.7	小结	233
第八章	终端仿真	234
8.1	如何创建测试程序	234
8.2	为什么要仿真终端	234
8.3	转义序列	235
8.3.1	终端智能	235
8.3.2	转义序列的发展	235
8.4	ANSI.SYS	236
8.4.1	ANSI.SYS 转义序列	236
8.4.2	ANSI.SYS 键盘序列	240
8.5	终端类	241
8.6	测试程序	243
8.7	AnsiTerminal 类	244
8.8	调试挂钩	258
8.9	BaseWindow 类	260
8.10	TextWindow 类	264
8.11	生成测试程序	278
8.12	小结	279
第九章	Win16 驱动程序	280
9.1	Windows 编程	280
9.2	Microsoft Windows 16 位设备驱动程序	281
9.3	通信 API	281
9.4	综合应用	288
9.5	Win16 测试程序	304
9.6	生成 Test232W.EXE	309
9.7	小结	311
第十章	Win32 通信 API	312

10.1 Win32 编程的不同之处	312
10.1.1 作为文件对象的通信端口	312
10.1.2 产生必要的线程	312
10.2 Win32 设备驱动程序	313
10.2.1 数据结构	313
10.2.2 通信 API	319
10.3 小结	334
第十一章 Win32Port 类	335
11.1 Win32Port 类	335
11.1.1 MTTY.C——按实例进行存档	335
11.1.2 线程结构和约定	336
11.1.3 输出线程	337
11.1.4 输入线程	341
11.1.5 通知	345
11.1.6 类成员概述	346
11.1.7 辅助类——MTDeque	359
11.1.8 辅助类——Dcb	360
11.2 Win32 下的 IdleFunction	361
11.3 小结	362
第十二章 Win32Test 程序	363
12.1 CHAPT12.EXE——基于控制台的测试程序	363
12.1.1 基于对话框的控制面板	363
12.1.2 定制的派生类	368
12.2 源代码	369
12.3 小结	409
第十三章 Win32 终端仿真	410
13.1 终端窗口类	410
13.1.1 需求	410
13.1.2 C++与 Windows	411
13.1.3 Win32Term 类——基本的内部函数	413
13.2 Win32 类——定义	415
13.2.1 私有类	415
13.2.2 数据成员	415
13.2.3 公共代码成员	420
13.2.4 保护代码成员	423
13.3 集成 Win32Term 与 AnsiTerm	425
13.3.1 AnsiWinTerm 类	426

13.3.2	AnsiWinTerm 和 Win32Port 类.....	426
13.3.3	AnsiWinTerm RX 通知.....	427
13.3.4	AnsiWinTerm 和 AnsiTerm 类.....	428
13.3.5	AnsiWinTerm 清单	429
13.4	本章演示程序	436
13.4.1	菜单命令	436
13.4.2	源代码.....	438
13.5	改进建议	446
13.6	小结	464
第十四章	在 MS-DOS 下使用调制解调器	465
14.1	调制解调器标准的冲突	465
14.1.1	硬件标准	465
14.1.2	软件标准	466
14.1.3	今天的调制解调器	471
14.2	调制解调器的功能	472
14.2.1	创建功能项	474
14.2.2	Modem 类.....	474
14.2.3	公共接口	476
14.2.4	保护成员	478
14.3	测试程序	485
14.4	生成 TSTMODEM.CPP.....	488
14.5	小结	489
第十五章	在 Win32 下使用调制解调器——TAPI	490
15.1	TAPI 的历史	490
15.2	TAPI 概述	491
15.2.1	TAPI 通知方法	491
15.2.2	同步与异步函数	491
15.2.3	TAPI 对象	492
15.2.4	TAPI 的麻烦之处	492
15.3	本章使用的 TAPI 函数	492
15.3.1	TAPI 初始化和关闭	493
15.3.2	线路控制	494
15.3.3	呼叫配置	497
15.3.4	呼叫控制	499
15.3.5	回调函数	502
15.4	SimpleTapi 类.....	503
15.4.1	类管理和状态	503

15.4.2 呼叫及线路管理/配置	504
15.4.3 通知函数	506
15.4.4 重要的内部函数	507
15.4.5 调试的注意事项	508
15.4.6 改进缺点	509
15.4.7 类 Tapi32Port——最后的细节	509
15.5 CHAPT15.EXE 演示程序	509
15.5.1 SimpleTapi 集成	510
15.5.2 源代码	511
15.6 小结	550
第十六章 文件传输与 ZMODEM	552
16.1 协议的历史	552
16.1.1 进入 ZMODEM	553
16.1.2 选择 ZMODEM 的原因	553
16.2 ZMODEM 概述	554
16.2.1 ZMODEM 帧类型	554
16.2.2 头格式	558
16.2.3 数据子包格式	559
16.2.4 编码	560
16.2.5 其他事项	560
16.3 文件传输	561
16.4 FileTransfer 类	562
16.5 Zmodem 类	563
16.6 测试程序	564
16.7 CRC 类	566
16.8 源代码	568
16.9 小结	594
附录 本书光盘内容	595
词汇表	599
本书光盘安装指导	604