



求是科技

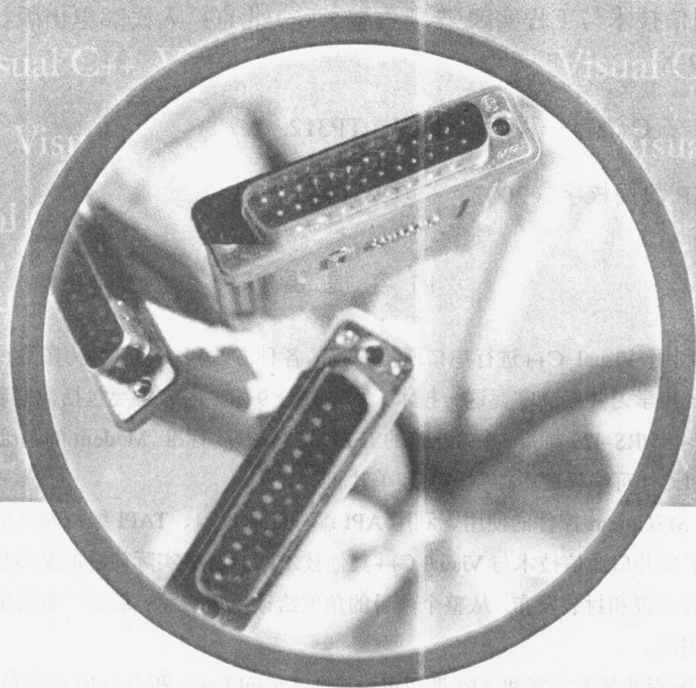


Visual C++ 串口通信技术与 工程实践

■ 求是科技
李现勇 编著

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS





Visual C++ 串口通信技术与 工程实践

■ 求是科技

李现勇 编著

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

Visual C++ 串口通信技术与工程实践/李现勇编著. —北京: 人民邮电出版社, 2002.5
ISBN 7-115-10249-X

I.V... II.李... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 020514 号

内 容 提 要

本书详细介绍了利用 Visual C++ 进行串口通信编程的各种方法和技巧, 并力图通过生动的讲解和丰富的应用实例让读者进一步学习并提高掌握这一技术。本书共分 9 章, 内容主要包括数据通信的基本概念、串口通信的 RS-232C/RS-485/RS-422 标准、并口通信的 IEEE 1284 1994 标准、Modem 的基础知识和 AT 命令集、传真机的传真协议、PLC (可编程逻辑控制器) 和单片机等。

本书详细介绍了 MSComm 控件的使用、利用 API 编程控制串口、TAPI 编程及其应用, 并通过一系列应用实例, 详细阐明了串并口通信技术与 Visual C++ 其他技术结合解决实际问题的基本思路和方法; 通过串口编程调试精灵的设计实现和封装发布, 从整个项目的角度给读者以启示; 最后一章给出串并口编程在工程实践中的诸多方案与用法。

本书中诸多串口编程的技术方案和 API 调用可供广大 Visual Basic 和 Delphi 程序员阅读参考。

本书专业性和实用性较强, 对于利用 Visual C++ 进行通信项目开发具有较高的参考价值。本书适合中高级程序员、软件开发人员和系统分析人员阅读和参考。

Visual C++ 串口通信技术与工程实践

◆ 编 著 求是科技 李现勇
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@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16
印张: 28.25
字数: 690 千字 2002 年 5 月第 1 版
印数: 1-5 000 册 2002 年 5 月北京第 1 次印刷

ISBN 7-115-10249-X/TP · 2842

定价: 49.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

在很多情况下, 远程监控和工业自动化领域系统常见的通信编程多为串并口通信编程。计算机串并口编程在通信软件中有着十分广泛的应用, 如电话、传真、视频和控制等。在 Visual C++、Delphi、Visual Basis 开发工具之间, Visual C++ 由于功能强大和应用灵活, 同时也得到 Microsoft 系统的最好支持, 因此涉及硬件操作的通信编程, 笔者一般采用 Visual C++ 作为开发工具。

编写本书的目的就是将笔者和其他通信软件设计开发人员的相关经验加以总结归纳, 本着将理论知识和应用相结合的思路, 系统详尽地介绍串并口编程中常见的、而又不容易解决的问题, 希望本书能够给广大的通信编程人员以帮助。

本书共分 9 章, 下面简单概述各章的内容。

第 1 章: 概述

本章首先介绍了计算机接口技术的基本内容, 然后重点介绍了串口通信协议——RS-232C、RS-485/422 标准 (包括引脚定义、分类、标准、安装和使用), 并口通信的 IEEE1284 1994 标准以及这些协议的硬件接口等, 然后介绍了计算机端口结构和端口资源、典型外设, 最后还分析了端口技术的发展应用前景。熟悉和掌握这些内容有助于从总体上了解串并口编程技术及其应用。

第 2 章: 端口通信编程基础

本章结合串并口编程的基本要求, 对串口通信的基本概念 (同步通信、异步通信、全双工、半双工、单工、通信协议等)、模数数模转换的原理和应用、端口参数和流控制方式等进行了简单的介绍, 最后介绍了串并口编程的简单控制流程和方法。

第 3 章: MSComm 控件编程控制串口

本章详细介绍了利用 MSComm 控件进行通信编程的基本方法、属性和事件。MSComm 控件将通信的大部分底层操作都封装在控件内部, 应用程序只需要获取和设置相应的 MSComm 控件属性就可以, 从而大大简化了编程工作。

本章结合一个字符传输的实例, 进一步阐述了 MSComm 控件通信的基本方法。最后, 还对利用 MSComm 控件通信的高级应用——SAFEARRAY 和帧结构进行了分析。

第 4 章: Windows API 编程控制串口

为了让读者对 Windows API 串口通信编程有一个全面的了解, 本章首先介绍了 Windows 开放服务结构和 Windows 操作系统下串口通信 API 的演变, 接着分类详细介绍了串口通信的主要 API 函数, 最后通过一个实例——TTY 终端仿真的实际构建和代码分析, 阐述了利用 Windows API 进行串口通信编程的基本流程和注意事项。

第 5 章: TAPI 编程

本章首先介绍了 TAPI 的基本体系结构, 之后介绍了辅助 TAPI 函数, 并通过一个实例描述了辅助 TAPI 函数的应用。然后详细介绍了 Windows 完全 TAPI, 并通过详细构建和分析一个基于 TAPI 电话拨号的程序, 使读者对 TAPI 主要函数和通信程序的基本步骤, 能有更深刻

的理解。

第6章：与典型外设通信

本章对串接口与典型外设通信进行了详细的阐述。首先详细介绍了PC机与Modem、传真机通信的基本内容，包括Modem的命令状态和在线状态以及两者之间的相互转换，AT命令集，S寄存器等；传真机的T.3和T.4协议、数据帧、DIS/DCS位影响、传真编码和传真过程的实例描述。然后分别介绍了串口和PLC、射频卡读写单元、单片机等的通信，并给出了相应的流程图和源代码。最后简单介绍了并口外设通信的基本情况。这一章能够大大拓展读者对串并口通信应用的基本认识。

第7章：Modem拨号系统

本章主要通过几个实例从不同的方面说明串口和Modem拨号系统应用的基本情况。其内容主要有串口对Modem的简单控制技术（两台计算机通过Modem和电话网互联）、实现电子邮件自动寻呼系统（串口和Modem拨号系统和MAPI技术结合）、串口编程与ODBC/DAO技术相结合（串口和Modem系统与数据库结束结合）、串行通信协议的软件侦测方法、一个免费的可以获得Modem信息的Visual C++类的应用和串口数据实时采集的解决方案和例程。

第8章：串口编程调试精灵

本章通过一个综合性的实例——串口编程调试精灵（具有串口监视、调试程序，可适用于Windows 9X/2000/NT平台）的设计实现和封装发布，让读者对一个项目从需求分析、设计实现到最终的终端程序的发布过程有一个初步的了解。

第9章：工程实践

本章给出若干典型的、串并口编程技术在工程实际中的应用。使读者更进一步掌握具体技术在工程实践中的使用方法。

在此，感谢人民邮电出版社的信任和支持，同时在中国工控网 www.chinakong.com 和啸峰工作室 roaringwind.best.163.com 网友发表的文章中得到了很多启迪，在此一并特别表示真诚的谢意。

本书范例全部采用Visual C++6.0实现，安装程序采用Installshield Professional完成，部分图形采用Smart Draw5.0实现。本书所有范例均可以在随书发售的CD-ROM光盘中找到。

由于作者水平有限，书中难免有不足和疏忽之处，恳请读者朋友和各位同仁批评指正。

作者
2002年3月
于中科院自动化所

目 录

第 1 章 概述	1
1.1 计算机接口技术简介	1
1.1.1 什么是接口	1
1.1.2 接口的基本功能	1
1.1.3 常见接口简介	2
1.1.4 接口的控制方式	4
1.2 RS-232-C 串口标准	5
1.2.1 RS-232-C 概述	5
1.2.2 数据终端设备与数据通信设备	6
1.2.3 RS-232 协议标准	7
1.2.4 RS-232 串口通信基本接线方法	11
1.3 RS-485/422 串口标准	12
1.3.1 RS-422 与 RS-485 的由来	12
1.3.2 RS-422 与 RS-485 接口标准	13
1.3.3 RS-422 与 RS-485 网络安装注意要点	15
1.3.4 RS-422 与 RS-485 的接地问题	16
1.4 IEEE1284 1994 并口标准	17
1.4.1 并行打印机端口概述	17
1.4.2 IEEE1284 1994 标准	18
1.4.3 增强型并行端口 (EPP)	19
1.4.4 扩展型并行端口	21
1.5 端口结构与端口资源	22
1.5.1 异步串口硬件结构	22
1.5.2 串口端口资源	24
1.5.3 串行接口输入输出过程描述	26
1.5.4 并行端口结构	26
1.5.5 并行端口资源	27
1.5.6 并行接口输入输出过程描述	28
1.5.7 并行接口与串行接口通信比较	28
1.6 典型外设介绍	29
1.6.1 打印机	29
1.6.2 CRT 终端设备	30
1.6.3 Modem	31

1.6.4 传真机	32
1.7 端口技术应用展望	33
1.8 本章小结	34
第2章 端口通信编程基础	35
2.1 基本概念	35
2.1.1 端口通信——同步传输和异步传输	35
2.1.2 单工、双工和全双工通信	36
2.1.3 端口通信——通信协议	37
2.1.4 数据编码	41
2.1.5 数据安全和完整	42
2.2 模数/数模转换	43
2.2.1 数模转换器 (DAC)	43
2.2.2 模数转换器 (ADC)	44
2.3 端口参数	46
2.3.1 比特率	46
2.3.2 流量控制方式	47
2.3.3 缓冲存储器	51
2.3.4 串行适配器终端和 I/O 地址	52
2.4 端口编程概述	53
2.5 本章小结	55
第3章 用 MSComm 控件进行串口编程	56
3.1 MSComm 控件介绍	56
3.1.1 MSComm 控件的操作模式	56
3.1.2 MSComm 控件的属性	58
3.1.3 MSComm 控件的事件	63
3.2 MSComm 控件串口编程——基本流程	64
3.2.1 在当前 Project 中插入 MSComm 控件	64
3.2.2 初始化并打开串口	64
3.2.3 捕捉串口事项	65
3.2.4 串口读写	65
3.2.5 关闭串口	66
3.3 MSComm 控件串口数据通信——字符传输	66
3.3.1 ASCII 控制字符	66
3.3.2 串口数据通信字符传输实例——字符仿真终端	69
3.4 MSComm 控件串口数据通信——高级应用	78
3.4.1 SAFEARRAY 数据结构	79
3.4.2 自定义数据帧格式	81
3.5 本章小结	88

第 4 章 Windows API 编程控制串口	89
4.1 Windows API 串口通信编程概述	89
4.2 Windows 串口通信相关 API 函数	90
4.2.1 打开和关闭串口	90
4.2.2 串口配置和串口属性	93
4.2.3 读写串口	103
4.2.4 通信事件	113
4.2.5 设备控制命令	115
4.3 Windows API 串口编程实例	116
4.3.1 功能目标	116
4.3.2 主要技术/算法	117
4.3.3 具体实现	118
4.4 本章小结	145
第 5 章 TAPI 编程	146
5.1 TAPI 简介	146
5.1.1 TAPI 的含义	146
5.1.2 TAPI 的体系结构	146
5.2 辅助 TAPI 及其应用	147
5.2.1 辅助 TAPI	147
5.2.2 辅助 TAPI 的应用——简单拨号程序	148
5.3 Windows 完整 TAPI	152
5.3.1 基本概念	152
5.3.2 Windows TAPI 函数集	154
5.4 编程实例——利用 TAPI 拨打电话	185
5.4.1 功能目标	185
5.4.2 主要技术/算法	186
5.4.3 具体实现	186
5.4.4 补充说明	226
5.5 本章小结	227
第 6 章 与典型外设通信	228
6.1 串口与 Modem	228
6.1.1 Modem 的基本工作原理	228
6.1.2 Modem 的分类、安装与使用	229
6.1.3 Modem 通信协议体系	230
6.1.4 Modem 通信的规则和标准	240
6.1.5 通用异步接收发送器 UART	259
6.1.6 Modem 技术的新发展	261
6.2 串口与传真机	262
6.2.1 传真机串口编程的基本知识	263

6.2.2	T.4 传真图像协议	270
6.2.3	DISC/DCS 位映像	276
6.2.4	传真编码	284
6.2.5	传真过程的实例描述	295
6.3	串口与 PLC 通信	302
6.3.1	PLC 的基本结构	302
6.3.2	PLC 编程口操作命令类型与通信端口初始化	304
6.3.3	PLC 编程口操作	305
6.4	串口与射频卡读写单元	311
6.4.1	射频卡读写单元的基本结构	311
6.4.2	实际的射频卡读写单元产品的简单介绍	312
6.5	串口与单片机通信	314
6.5.1	串行通信硬件电路图	314
6.5.2	异步串行通信软件实现	315
6.6	并口外设通信简介	318
6.6.1	并行打印机适配器	318
6.6.2	并行端口示警盒	319
6.6.3	并行端口的电源开关电路	319
6.6.4	并口 IO 控制卡	320
6.7	本章小结	322
第 7 章	串口和 Modem 拨号应用实例	323
7.1	对 Modem 的简单控制技术	323
7.1.1	功能目标	323
7.1.2	主要技术/算法	323
7.1.3	具体实现	323
7.1.4	补充说明	336
7.2	电子邮件自动寻呼系统	338
7.2.1	功能目标	338
7.2.2	主要技术/算法	339
7.2.3	具体实现	343
7.2.4	补充说明	355
7.3	串口编程与 ODBC/DAO 技术相结合	355
7.3.1	功能目标	356
7.3.2	主要技术/算法	356
7.3.3	具体实现	361
7.3.4	补充说明	374
7.4	串行通信协议的软件侦测方法	375
7.4.1	功能目标	376
7.4.2	主要技术/算法	376

7.4.3	具体实现	376
7.4.4	补充说明	392
7.5	获取 Modem 信息的 Visual C++ 类	393
7.5.1	功能目标	393
7.5.2	主要技术/算法	393
7.5.3	具体实现	394
7.5.4	补充说明	396
7.6	串口数据实时采集的解决方案和例程	397
7.6.1	功能目标	397
7.6.2	主要技术/算法	398
7.6.3	具体实现	398
7.7	本章小结	402
第 8 章	串口编程调试精灵	403
8.1	串口编程调试精灵的设计实现	403
8.1.1	功能目标	403
8.1.2	主要技术/算法	403
8.1.3	具体实现	404
8.1.4	补充说明	422
8.2	串口编程调试精灵的发布	423
8.2.1	串口编程调试精灵安装项目的基本设置	423
8.2.2	串口编程调试精灵安装项目文件的设置	425
8.3	本章小结	427
第 9 章	工程实践案例	428
9.1	云台控制系统	428
9.1.1	系统描述	428
9.1.2	设备布局与线路连接	428
9.1.3	功能分析与设计	429
9.1.4	关键技术与算法	431
9.1.5	特别强调与补充	433
9.2	智能安防报警系统	433
9.2.1	系统描述	433
9.2.2	设备布局与线路连接	434
9.2.3	功能分析与工作流程	435
9.2.4	关键技术与算法	436
9.3	本章小结	440

第 1 章 概 述

1.1 计算机接口技术简介

1.1.1 什么是接口

通常情况下，人们是通过外部设备使用计算机的，但是由于种种原因，外设往往不能与 CPU 直接相连，它们之间的信息交换需要一个中间环节（或界面），我们称之为接口电路。

接口是 CPU 与外界的连接部件（电路），是 CPU 与外界进行信息交换的中转站。从广义上讲接口技术，是研究 CPU 如何与外界进行最佳结合与匹配，将计算机系统中的各种功能部件连接起来构成一个完整的、实用的计算机系统，并能实现与外界高效、可靠地交换信息的一门硬、软件相结合的技术，是软硬件结合之体现，是微型计算机应用的关键。例如，CPU 与系统总线连接所需的总线驱动器、数据收发器、总线控制器及总线裁决器等，称为处理器接口；存储器的组织及与系统总线的连接逻辑、存储管理部件、DMA 控制器等，称为存储器接口；各种输入输出设备与主机进行信息交换所需要的硬件逻辑和软件设计，称之为外部设备接口技术或输入输出接口技术（I/O 接口）。I/O 设备品种繁多，其相应的接口电路也各不相同，因此，习惯上说到接口仅指 I/O 接口。

1.1.2 接口的基本功能

由于计算机的外围设备品种繁多，它们几乎都采用了机电传动设备，因此，CPU 在与 I/O 设备进行数据交换时存在以下问题：

（1）速度不匹配

I/O 设备的工作速度通常要比 CPU 慢许多，而且由于种类的不同，I/O 设备之间的速度差异也很大，例如硬盘的传输速度就要比打印机快。

（2）时序不匹配

各个 I/O 设备都有自己的定时控制电路，它们均以自己的速度传输数据，无法与 CPU 的时序取得统一。

（3）信息格式不匹配

不同的 I/O 设备存储和处理信息的格式不同，例如可以分为串行和并行两种，也可以分为二进制格式、ASCII 编码和 BCD 编码等。

(4) 信息类型不匹配

不同 I/O 设备采用的信号类型不同,有些是数字信号,而有些是模拟信号,因此所采用的处理方式也不同。

基于以上原因,CPU 与外设之间的数据交换必须通过接口来完成,通常情况下,接口应具备以下一些功能:

(1) 数据缓冲功能

接口中设置数据寄存器或锁存器,以解决主机高速与外设低速的矛盾,避免因速度不一致而造成数据丢失。接口通常由一些寄存器或 RAM 芯片组成,如果芯片存储容量足够大还可以实现批量数据的传输。

(2) 信号转换功能

由于外设所提供的状态信号和它所需要的控制信号往往同微机的总线信号不兼容,因此,信号转换就成为接口设计中关键的一环。

(3) 设备选择功能

系统中一般带有多种外设,而 CPU 在同一时间里只能与微机同一外设交换信息,因此就需要借助接口的地址码对外设进行寻址,被选定的设备才能与 CPU 进行数据交换或通信。

(4) 接受、解释并执行微处理器的命令功能

CPU 对外设的各种命令都是以代码形式发送到接口电路中,然后再经过接口电路读取后,形成一系列控制信号去控制被控对象。为了联络,接口电路还要提供寄存器“空”、“满”、“准备好”、“忙”、“闲”等状态信号,并向 CPU 报告寄存器工作情况。

(5) 中断管理功能

当外设迫切需要 CPU 的服务时,特别在出现故障需要 CPU 即刻处理时,就会要求在接口中设置中断控制器,为 CPU 处理有关中断事务,这样既增加了微机系统对外界的响应速度,又使 CPU 与外设并行工作,大大提高了 CPU 的工作效率。

(6) 数据宽度变换的功能

CPU 所处理的是并行数据,而有的外设只能处理串行数据,因此,接口就应该具有“数并→串和串→并”的变换功能。为此,在接口中设置移位寄存器。

(7) 可编程功能

现在的接口芯片绝大多数是可编程的,这样可以在不改动硬件的情况下,只修改驱动程序即可改变接口的工作方式,该功能增强了接口的灵活性和可扩充性,使接口逐步向智能化发展。

上述各种功能并非每种芯片都具备,不同配置和不同用途的系统,其接口功能不尽相同,接口电路的复杂程度也大不一样。

1.1.3 常见接口简介

1. 并行接口

主机与接口、接口与外设之间都是以并行方式传送数据。即每次传送一个字或字节的全

部代码。其特点是：

- 接口中具有与系统总线直接相连的输入/输出数据缓冲器和一组数据线。
- 接口中具有与外部设备进行联络的应答信号。
- 接口中有地址译码电路，产生片选信号和内部寄存器地址选择。
- 接口可使用程序传送方式或程序中断传送方式实现外设与主机之间的信息交换。
- 简单并行接口只有一个 I/O 通路驱动外设，对于复杂的、可编程并行接口可有几个 I/O 通路驱动不同外设的输入/输出。

并行接口的数据通路宽度是按字或字节设置，其数据传送速率高。当外部设备的工作方式是并行传送数据，并且与主机系统距离较近时，则选用并行接口。例如 Intel 公司的 Intel 8255、Motorla 公司的 MC6820、ZILOG 公司的 Z80-PIO 等组件都属于这一类。由于并行接口可以直接和各种 I/O 设备的数据线相连，所以用它来连接外设组成系统非常方便，故在慢速设备中使用很普遍。

2. 串行接口

计算机的另一种标准接口是串行口，现在的 PC 机一般至少有两个串行口 COM1 和 COM2。串行口不同于并行口之处在于它的数据和控制信息是一位接一位串行地传送下去。这样，虽然速度会慢一些，但传送距离较并行口更长，因此长距离的通信应使用串行口。通常 COM1 使用的是 9 针 D 形连接器，而 COM2 使用的是老式的 DB25 针连接器。常用的串行外设用电打字机、CRT 终端设备、传真机以及远程数据采集设备、通信设备等。

3. 磁盘接口

(1) IDE 接口

IDE 的全称是 Integrated Driver Electronics，一般也叫做 ATA 端口（AT-Attachment，AT 嵌入式接口），即集成驱动器电子部件。它是由 COMPAQ 公司开发并由 Western Digital 公司生产的控制器接口。

IDE 是在原有的 ST506 的基础上改进而成的，它的最大特点是把控制器集成到驱动器内。因此在硬盘适配卡中，不再有控制器这一部分了。这样做的最大好处是由于把控制器和驱动电路集成到一起，可以消除驱动器和控制器之间的数据丢失问题，使数据传输十分可靠。这就可以将每一磁道的扇区数提高到 30 以上，从而增大可访问容量。由于控制器电路并入驱动器内，因而从驱动器中引出的信号线已不是控制器和驱动器之间的接口信号线，而是通过简单处理后可以与主系统连接的接口信号线，这种接口方式是与 ST506 接口不同的。它只可以接两个容量不超过 528MB 的硬盘驱动器，接口的成本很低，因此在 386、486 时期非常流行。但大多数 IDE 接口不支持 DMA 数据传送，只能使用标准的 PCI/O 端口指令来传送所有的命令、状态、数据。

(2) EIDE 接口

EIDE 接口较 IDE 接口有了很大改进，是目前最流行的接口。首先，它所支持的外设不再是 2 个而是 4 个了，所支持的设备除了硬盘，还包括 CD-ROM 驱动器磁盘备份设备等。EIDE 通常提供两个插座，称为主插座和辅插座，每个插座又可连接主、从两个设备。主插座通常与高速的局部总线相连，供硬盘使用。而辅插座则与 ISA 总线相连，供磁带机或

CD-ROM 驱动器相连。其次, EIDE 标准取消了 528MB 的限制, 对每个硬盘支持的最高容量可达 8.4G。第三, EIDE 有更高的数据传送速率, 支持 PIO 模式 3 和模式 4 标准, 支持的数据传输率可达 6Mbit/s 以上。EIDE 支持的 3 种硬盘工作模式的技术参数如表 1-1 所示。

表 1-1 EIDE 支持的三种硬盘工作模式的技术参数表

技术参数 硬盘参数	最大柱面数	最大磁头数	最大扇区数	每扇区数字节	硬盘最大容量
普通模式	1024	16	63	512	528MB
寻址模式	1024	255	63	512	8.4GB
大硬盘模式	512	32	63	512	528MB

4. SCSI 接口

SCSI (Small Computer System Interface) 小计算机系统接口, 在做图形处理和网络服务的计算机中被广泛采用 SCSI 接口的硬盘。除了硬盘以外, SCSI 接口还可以连接 CD-ROM 驱动器、扫描仪和打印机等, 它具有以下特点:

- SCSI 是一个多任务接口, 具有总线仲裁功能。因此, SCSI 总线上的适配器和控制器可以并行工作, 在同一个 SCSI 控制器控制下的多台外设也可以并行工作。SCSI 可以按同步方式和异步方式传输数据。SCSI 总线上的设备没有主从之分, 双方平等。
- SCSI 可分为单端传送方式和差分传送方式。单端 SCSI 的电缆不能超过 6m, 如果数据传送距离超过 6m, 应采用差分 SCSI 传送方式。
- SCSI 接口是智能化的, 可以彼此通信而不增加 CPU 的负担。在 IDE 和 EIDE 设备之间传输数据时, CPU 必须介入, 而 SCSI 设备在数据传输过程中起主动作用, 并能在 SCSI 总线内部具体执行, 直至完成再通知 CPU。

但是 SCSI 接口的成本较 IDE 和 EIDE 接口高很多, SCSI 接口硬盘必须和 SCSI 接口卡配合使用, SCSI 接口卡也比 IDE 和 EIDE 接口贵很多。总的来说 SCSI 对 PC 机应是一种很好的选择, 它不仅是一个接口, 也是一条总线。随着技术的发展, SCSI 也会像 IDE 一样广泛应用在微机系统和外设中。

5. USB 接口

最新的 USB 串行接口标准是由 Microsoft、Intel、COMPAQ、IBM 等大公司共同推出的, 这是新一代接口标准, 也是一种连接外围设备的机外总线。它提供机箱外的热即插即用连接, 用户在连接外设时不用再打开机箱、关闭电源, 而是采用“级联”方式, 每个 USB 设备用一个 USB 插头连接到一个外设的 USB 插座上, 而其本身又提供一个 USB 插座给下一个 USB 设备使用, 通过这种方式的连接, 一个 USB 控制器可以连接多达 127 个外设, 而每个外设间的距离可达 5m。USB 统一的 4 针圆形插头将取代机箱后的众多的串 / 并口 (如鼠标、Modem 键盘等) 插头。USB 能智能识别 USB 链上外围设备的插入或拆卸, 除了能够连接键盘、鼠标, 还可以连接 ISDN、电话系统、数字 音响、打印机以及扫描仪等外设。

1.1.4 接口的控制方式

CPU 通过接口对外设进行控制的方式有以下几种:

(1) 程序查询方式

在这种方式下, CPU 通过 I/O 指令询问指定外设当前的状态, 如果外设准备就绪, 则进行数据的输入或输出, 否则 CPU 等待, 循环查询。

这种方式的优点是结构简单, 只需要少量的硬件电路即可, 缺点是由于 CPU 的速度远远高于外设, 因此通常处于等待状态, 工作效率低。

(2) 中断处理方式

在这种方式下, CPU 不再被动等待, 而是可以执行其他程序, 一旦外设为数据交换准备就绪, 就可以向 CPU 提出服务请求, CPU 如果响应该请求, 便暂时停止当前程序的执行, 转去执行与该请求对应的服务程序, 完成后, 再继续执行原来被中断的程序。

中断处理方式的优点是显而易见的, 它不但为 CPU 省去了查询外设状态和等待外设就绪所花费的时间, 提高了 CPU 的工作效率, 还满足了外设的实时要求。但需要为每个 I/O 设备分配一个中断请求号和相应的中断服务程序, 此外还需要一个中断控制器 (I/O 接口芯片) 来管理 I/O 设备提出的中断请求, 例如设置中断屏蔽、中断请求优先级等。

中断处理方式的缺点是每传送一个字符都要进行中断, 启动中断控制器, 还保留和恢复现场以便能继续执行原程序, 花费的工作量很大, 这样如果需要大量数据交换, 系统的性能会很低。

(3) DMA (直接存储器存取) 传送方式

DMA 最明显的一个特点是它不是用软件而是采用一个专门的硬件电路——DMA 控制器来控制内存与外设之间的数据交流。该方式无须 CPU 介入, 大大提高 CPU 的工作效率。

在进行 DMA 数据传送之前, DMA 控制器会向 CPU 申请总线控制权, CPU 如果允许, 则将控制权交出, 因此, 在数据交换时, 总线控制权由 DMA 控制器掌握, 在传输结束后, DMA 控制器将总线控制权交还给 CPU。

1.2 RS-232-C 串口标准

1.2.1 RS-232-C 概述

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 RS-232 标准的基础上经过改进而形成的。RS-232C 标准是美国 EIA (电子工业联合会) 与 BELL 等公司一起开发并于 1969 年公布的通信协议。它适合于数据传输速率在 $0 \sim 20000 \text{ bit/s}$ 范围内的通信。远程工业协会 (TIA) 1997 年发布了最新的一个版本, 命名为 TIA/EIA-232-F。ITU (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。这个标准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, 它不仅已被内置于每一台计算机, 同时也已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。RS-232 常用于连接到一个 Modem, 其他拥有 RS-232 接口的设备包括打印机、数据采集模块、测试装置和控制回路。另外, RS-232 也可以直接应用于任何

类型的计算机之间的简单连接上。

RS-232 有几个优点：

- RS-232 应用广泛。每一台 PC 都有一个或多个 RS-232 端口。更新的计算机现在支持其他诸如 USB 这样的串行接口，但是 RS-232 可以做很多 USB 无法进行的工作。
- 在微控制器中，接口芯片使得一个 5V 串口转换成 RS-232 变得非常容易。
- 连接距离可以达到 50~100 英尺。USB 连接最长可以达到 16 英尺，PC 机的并行打印机接口与主机的距离可以达到 10~15 英尺，或者利用 IEEE-1284B 型驱动器可以达到 30 英尺。如果 RS-232 端口与 Modem 相连，则可以在世界范围内接受和传送数据。
- 对于双向连接，只需要 3 条导线。而并行连接一般需要 8 条数据线，两条或者更多的控制线信号线和几条接地线，这使得连接成本也比较高。

注意：最近，出现了很多更快和更复杂的接口技术，但是因为 RS-232 的硬件编程要求比较简单，价格便宜，而且现有设备中很多已经内置了这种嵌入式接口。另外 RS-232 也产生了许多派生产品，尽管他们更快或者更便宜，但是在很多方面始终保持与 RS-232 的兼容。因此，本节将详细介绍 RS-232 标准。

1.2.2 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，统称为数据通信。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：

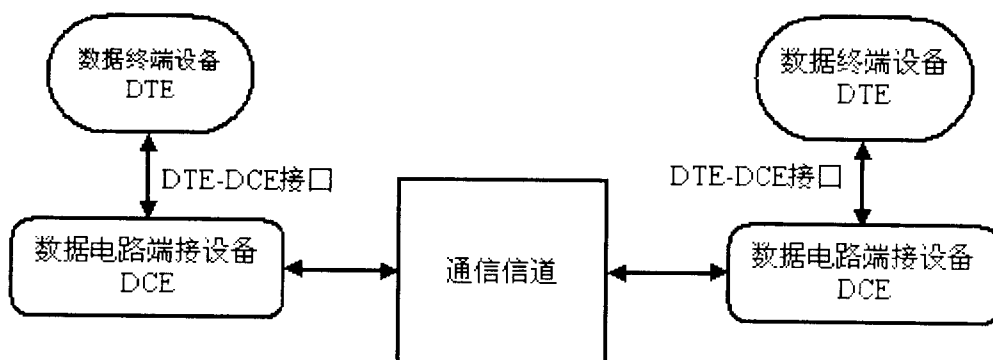


图 1-1 数据通信系统的基本构成

数据通信发送方为信源，接受方为信宿。链接信源和信宿的通道称为通信通道（简称信道）或通信线路，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，因此必需通过相同的传输规则才可以使得通信双方协调、稳定、可靠地工作。数据电路加上传输控制规则称为数据链路。

在数据通信系统中，用于接受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，也可以是一台只接受数据的打印机。用来连接与数据通信网络的设备称为数据通信设备 DCE（Data communications Equipment）或称为数据电路终结设备。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，它们必须经过一个中间连接设备——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。

在讨论 RS-232C 接口标准的内容之前，应说明两点：

首先，RS-232C 标准最初是远程通信连接数据终端设备 DTE (Data Terminal Equipment) 与数据通信设备 DCE (Data Communication Equipment) 而制定的。因此标准的制定，并未考虑计算机系统的应用要求。但目前它又广泛地被用于计算机（更准确说，是计算机接口）与终端或外设之间的近端连接标准。显然，这个标准的有些规定和计算机系统是不一致的，甚至是相矛盾的。有了对这种背景的了解，我们对 RS-232C 标准与计算机不兼容的地方就不难理解了。

其次，RS-232C 标准中所提到的“发送”和“接收”都是站在 DTE 立场上，而不是站在 DCE 的立场来定义的。由于在计算机系统中，往往是 CPU 和 I/O 设备之间传送信息，两者都是 DTE，因此双方都能发送和接收。

1.2.3 RS-232 协议标准

电气特性

EIA-RS-232C 对电器特性、逻辑电平和各种信号线功能都作了规定。

在 TxD 和 RxD 上：

- 逻辑 1 (MARK) = -3V ~ -15V。
- 逻辑 0 (SPACE) = +3 ~ +15V。

在 RTS、CTS、DSR、DTR 和 DCD 等控制线上：

- 信号有效（接通，ON 状态，正电压）= +3V ~ +15V。
- 信号无效（断开，OFF 状态，负电压）= -3V ~ -15V。

以上规定说明了 RS-232C 标准对逻辑电平的定义。对于数据（信息码）：逻辑“1”（传号）的电平低于-3V，逻辑“0”（空号）的电平高于+3V；对于控制信号：接通状态（ON）即信号有效的电平高于+3V、低于+15V，断开状态（OFF）即信号无效的电平高于-15V 低于-3V，也就是当传输电平的绝对值大于 3V 时，电路可以有效地检查出来，介于-3~+3V 之间的电压无意义，低于-15V 或高于+15V 的电压也认为无意义，因此，实际工作时，应保证电平在±(3~15)V 之间。

EIA-RS-232C 与 TTL 转换：EIA-RS-232C 用正负电压来表示逻辑状态，与 TTL 以高低电平表示逻辑状态的规定不同。因此，为了能够同计算机接口或终端的 TTL 器件连接，必须在 EIA-RS-232C 与 TTL 电路之间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。目前较为广泛地使用集成电路转换器件，如 MC1488、SN75150 芯片可完成 TTL 电平到 EIA 电平的转换，而 MC1489、SN75154 可实现 EIA 电平到 TTL 电平的转换。MAX232 芯片可完成 TTL ↔ EIA 双向电平转换，图 1-2 显示了 1488 和 1489 的内部结构和引脚。

MC1488 的引脚 2、(4, 5)、(9, 10) 和 (12, 13) 接 TTL 输入。引脚 3、6、8、11 输出端接 EIA-RS-232C。MC1498 的 1、4、10、13 脚接 EIA 输入，而 3、6、8、11 脚接 TTL 输出。具体连接方法如图 1-3 所示。图 1-3 中的左边是微机串行接口电路中的主芯片 UART，