

C/C++

串口通信

典型应用实例编程实践

曹卫彬 等编著



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

C/C++串口通信

典型应用实例编程实践

曹卫彬 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书从工程应用的角度出发,对目前流行的三种不同类型的 C/C++ 语言(包括 C++ Builder、Visual C++ 和 LabWindows/CVI)开发串口通信(也称为串行通信)程序进行了阐述。基础篇简单介绍了 C/C++ 语言实现串口通信需要的基本知识;应用篇通过 PC 与 PC、PC 与单片机、PC 与智能仪器、PC 与 PLC、PC 与 GSM 短信模块、PC 与远程 I/O 模块等典型应用实例,详细介绍了 C/C++ 语言利用 MSComm 控件、API 函数实现串口通信的程序设计方法。每个实例的设计任务同时采用 C++ Builder、Visual C++ 和 LabWindows/CVI 实现。

本书内容丰富,结构清晰,文字叙述深入浅出,通俗易懂,以实践应用为主,突出程序设计,重在功能实现,有较强的实用性和可操作性。

本书可作为高等院校计算机应用、测控技术、机电一体化、自动化等专业的研究生、本科生以及计算机测控系统研发的工程技术人员等学习 C/C++ 串口通信编程技术的参考书。

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

C/C++ 串口通信典型应用实例编程实践 / 曹卫彬等编著. —北京:电子工业出版社, 2009.7

ISBN 978-7-121-08980-0

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

中国版本图书馆 CIP 数据核字(2009)第 086167 号

责任编辑:高买花 田宏峰 特约编辑:刘 涛

印 刷:北京天宇星印刷厂

装 订:三河市皇庄路通装订厂

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:20.25 字数:518 千字

印 次:2009 年 7 月第 1 次印刷

印 数:4 000 册 定价:48.00 元(含光盘 1 张)

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。

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。

服务热线:(010) 88258888。

前言

RS-232/485 串口通信总线是一种久远但目前仍常用的通信方式,早期的仪器、单片机、PLC 等均使用串口与计算机进行通信,最初多用于数据通信中,随着工业测控行业的发展,许多测量仪器都带有 RS-232 串口总线接口。

因此,在计算机测控领域,串口通信编程是很多技术人员要面临的问题;在高校,有众多的理工科学生在学习串口通信编程技术,他们都迫切需要相关的书籍来帮助他们学习。

开发串口通信的编程语言很多,但应用最普遍、功能最强大的还是 C 语言,早期一般使用 Turbo C,现在已很少使用,目前市面上流行的 C 语言是 C++ Builder、Visual C++ 和 LabWindows/CVI。

C++ Builder 是 Borland 公司推出的一种基于面向对象程序设计语言的可视化集成编程工具,它具有专业 C++ 开发环境所能提供的全部功能,实现了可视化的编程环境和功能强大的编程语言(C++)的完美结合。它的核心来自 Delphi,使用的 VCL 类库和 Delphi 是完全一样的,其中 VCL 功能丰富,易于使用,并且完美地体现了面向对象的设计思想。C++ Builder 具有编译速度快、IDE 反应速度快、完全开放的 VCL 源代码、领先的数据库技术等优点,程序设计者可以利用 C++ Builder 完成许多高级应用开发,扩展空间几乎是无限的。

Visual C++ 是微软公司推出的一种面向对象的可视化集成编程工具,使用 Visual C++ 可以胜任从最简单的用户界面程序到高级、复杂的 Windows 应用程序的编程工作。使用 Visual C++ 编写的程序具有代码尺寸小、运行速度高和移植能力强等特点。Visual C++ 成为目前最受欢迎的开发工具之一,其中一个重要的原因是它有一个庞大而且功能齐全的 MFC 类库,该类库实现了对大多数 Windows API 的封装。在 Visual C++ 6.0 中既可以使用 MFC 完成大多数的工作,也可以调用 Windows API 函数完成更深层次的开发。

LabWindows/CVI 是 NI 公司开发的 Measurement Studio 软件组中的一员,它是 32 位的面向计算机测控领域的虚拟仪器软件开发平台,可以在多操作系统下运行。LabWindows/CVI 是以 ANSI C 为核心的交互式虚拟仪器开发环境,它将功能强大的 C 语言与测控技术有机地结合起来,具有灵活的交互式编程方法和丰富的库函数,为开发人员建立检测系统、自动测试环境、数据采集系统、过程监控系统等提供了理想的软件开发环境。

本书同时采用 C++ Builder、Visual C++ 和 LabWindows/CVI 三种语言,通过 PC 与 PC、PC 与单片机、PC 与智能仪器、PC 与 PLC、PC 与 GSM 短信模块、PC 与远程 I/O 模块等串口通信典型应用实例,使用多种方法(控件、库函数等)编写串口通信程序,使读者对 C/C++ 语言编写串口通信程序有一个全面的认识。

本书内容丰富,结构清晰,文字叙述深入浅出,通俗易懂,以应用实践为主,突出程序设计,重在功能实现,有较强的实用性和可操作性。

书中提供的程序具有实际参考价值,全部在 Windows XP 环境下运行通过,并经过硬件系统测试,读者可以直接拿来使用或者稍加修改便可用于自己的项目设计中。

为了方便读者学习,本书提供配套光盘,内容包括源代码、教学视频以及软硬件资源等。

本书可作为高等院校计算机应用、测控技术、机电一体化、自动化等专业的研究生、本科生以及计算机测控系统研发的工程技术人员等学习 C/C++ 串口通信编程技术的参考书。

本书由石河子大学曹卫彬、朱东芹、田敏、郑瑶、刘恩博、李江全编著。其中郑瑶负责基础篇(第1~2章)编写,朱东芹、田敏、刘恩博分别负责应用篇(第3~10章)中C++ Builder、Visual C++、LabWindows/CVI串口通信程序设计及操作步骤编写,曹卫彬教授负责统稿,李江全负责硬件系统搭建与测试。南京朝阳仪表、电子开发网等公司为本书提供了大量的技术支持,编者借此机会对他们致以深深的谢意。

编者要特别感谢石河子大学教材出版计划、新疆建设兵团高新技术研究发展计划对本书出版所做的支持和帮助。

由于编者水平有限,书中难免存在不妥或错误之处,恳请广大读者批评指正。

编者

2009年7月

目 录

基础篇

第 1 章 串口通信基础	(2)
1.1 串口通信的基本概念	(2)
1.1.1 并行通信与串行通信	(2)
1.1.2 串行通信的工作模式	(3)
1.1.3 异步传输与同步传输	(4)
1.1.4 硬件握手与软件握手	(4)
1.1.5 串口通信的基本参数	(5)
1.2 串口通信及其标准	(7)
1.2.1 RS-232C 串口通信标准	(7)
1.2.2 RS-485 串口通信标准	(9)
1.2.3 串口通信线路连接	(11)
1.2.4 个人计算机中的串口	(12)
1.3 串口通信调试	(15)
1.3.1 调试软件	(15)
1.3.2 应用实例	(17)
1.3.3 虚拟串口	(18)
第 2 章 C/C++编程语言与串口通信	(20)
2.1 串行通信控件 MSComm	(20)
2.1.1 MSComm 控件处理通信的方式	(20)
2.1.2 MSComm 控件的引用	(21)
2.1.3 MSComm 控件的常用属性	(23)
2.1.4 MSComm 控件的事件	(27)
2.1.5 MSComm 控件通信步骤	(29)
2.2 API 函数	(30)
2.2.1 动态链接库与 API 函数	(30)
2.2.2 C++与动态链接库	(31)
2.2.3 与串口通信有关的 API 函数	(32)
2.2.4 利用 API 函数操作串口	(36)
2.2.5 利用 API 函数实现串口通信的查询方法	(38)
2.3 线程	(39)
2.3.1 进程与线程的基本概念	(40)
2.3.2 线程的同步与优先级	(41)

2.3.3 在串口通信编程中使用线程	(42)
2.4 LabWindows/CVI 串口通信函数	(42)
2.4.1 串口打开/关闭函数	(43)
2.4.2 串口输入/输出函数	(43)
2.4.3 串口控制函数	(44)
2.4.4 串口状态查询函数	(45)
2.4.5 串口事件处理函数	(46)
2.4.6 调制解调文件传输函数	(46)

应用篇

第3章 PC 与 PC 串口通信程序设计	(48)
3.1 设计说明	(48)
3.1.1 设计目的	(48)
3.1.2 设计用软、硬件	(48)
3.1.3 硬件线路	(48)
3.1.4 设计任务	(49)
3.2 任务实现	(50)
3.2.1 利用 C++ Builder 实现 PC 与 PC 串口通信	(50)
3.2.2 利用 Visual C++ 实现 PC 与 PC 串口通信	(64)
3.2.3 利用 LabWindows/CVI 实现 PC 与 PC 串口通信	(80)
第4章 PC 与单片机串口通信程序设计	(88)
4.1 设计说明	(88)
4.1.1 实训目的	(88)
4.1.2 设计用软、硬件	(88)
4.1.3 硬件线路	(89)
4.1.4 设计任务	(89)
4.2 任务实现	(90)
4.2.1 单片机端 C51 程序	(90)
4.2.2 利用 C++ Builder 实现 PC 与单片机串口通信程	(103)
4.2.3 利用 Visual C++ 实现 PC 与单片机串口通信程	(116)
4.2.4 利用 LabWindows/CVI 实现 PC 与单片机串口通信程	(136)
第5章 PC 与智能仪器串口通信程序设计	(153)
5.1 设计说明	(153)
5.1.1 设计目的	(153)
5.1.2 设计用软、硬件	(153)
5.1.3 硬件线路	(154)
5.1.4 设计任务	(158)
5.2 任务实现	(158)
5.2.1 利用 C++ Builder 实现 PC 与智能仪器串口通信	(158)
5.2.2 利用 Visual C++ 实现 PC 与智能仪器串口通信	(163)

5.2.3	利用 LabWindows/CVI 实现 PC 与智能仪器串口通信	(169)
第 6 章	PC 与 PLC 串口通信程序设计	(175)
6.1	设计说明	(175)
6.1.1	设计目的	(175)
6.1.2	设计用软、硬件	(175)
6.1.3	硬件线路	(176)
6.1.4	设计任务	(176)
6.2	任务实现	(177)
6.2.1	PLC 端程序	(177)
6.2.2	利用 C++ Builder 实现 PC 与 PLC 串口通信	(180)
6.2.3	利用 Visual C++ 实现 PC 与 PLC 串口通信	(196)
6.2.4	利用 LabWindows/CVI 实现 PC 与 PLC 串口通信	(210)
第 7 章	PC 与 GSM 短信模块串口通信程序设计	(226)
7.1	设计说明	(226)
7.1.1	设计目的	(226)
7.1.1	设计用软、硬件	(226)
7.1.2	硬件线路	(227)
7.1.3	设计任务	(227)
7.2	任务实现	(228)
7.2.1	利用 C++ Builder 实现 PC 与 GSM 短信模块串口通信	(228)
7.2.2	利用 Visual C++ 实现 PC 与 GSM 短信模块串口通信	(243)
7.2.3	利用 LabWindows/CVI 实现 PC 与 GSM 短信模块串口通信	(257)
第 8 章	PC 与读卡器串口通信程序设计	(263)
8.1	设计说明	(263)
8.1.1	设计目的	(263)
8.1.2	设计用软、硬件	(263)
8.1.3	设计线路	(264)
8.1.4	设计任务	(264)
8.2	任务实现	(264)
8.2.1	利用 C++ Builder 实现 PC 与读卡器串口通信	(264)
8.2.2	利用 Visual C++ 实现 PC 与读卡器串口通信	(268)
第 9 章	PC 与智能仪器构成的 DCS 程序设计	(274)
9.1	设计说明	(274)
9.1.1	设计目的	(274)
9.1.2	设计用软、硬件	(274)
9.1.3	硬件线路	(275)
9.1.4	设计任务	(276)
9.2	任务实现	(276)
9.2.1	利用 C++ Builder 实现 PC 与多个智能仪器串口通信	(276)
9.2.2	利用 Visual C++ 实现 PC 与多个智能仪器串口通信	(282)

9.2.3 利用 LabWindows/CVI 实现 PC 与多个智能仪器串口通信	(289)
第 10 章 PC 与远程 I/O 模块构成的 DCS 程序设计	(295)
10.1 设计说明	(295)
10.1.1 设计目的	(295)
10.1.2 设计用软、硬件	(295)
10.1.3 硬件线路	(296)
10.1.4 设计任务	(299)
10.2 任务实现	(299)
10.2.1 利用 C++ Builder 实现 PC 与远程 I/O 模块	(299)
10.2.2 利用 Visual C++ 实现 PC 与远程 I/O 模块	(304)
10.2.3 利用 LabWindows/CVI 实现 PC 与远程 I/O 模块	(309)
参考文献	(315)

基础篇

◇ 串口通信基础

◇ C/C++编程语言与串口通信

串口通信入门

串口通信入门

串口通信入门

串口通信入门

串口通信入门

串口通信入门

串口通信入门

第1章

串口通信基础

串行总线是一种久远但目前仍常用的通信方式，早期的仪器、单片机、PLC 等均使用串口与计算机进行通信，最初多用于数据通信上，但随着工业测控行业的发展，许多测量仪器都带有串口总线接口，因此了解和掌握串口通信技术及其编程是非常有必要的。

串行接口一般包括 RS-232/422/485，其技术简单成熟，性能可靠，价格低廉，所要求的软硬件环境或条件都很低，广泛应用于计算机及相关领域，遍及调制解调器（Modem）、串行打印机、各种监控模块、PLC、摄像头云台、数控机床、单片机及相关智能设备，甚至路由器也不例外（通过串口设置参数）。



1.1 串口通信的基本概念

1.1.1 并行通信与串行通信

终端与其他设备（例如其他终端、计算机和外部设备）通过数据传输进行通信。数据传输可以通过两种方式进行：并行通信和串行通信。

1. 并行通信

在计算机和终端之间的数据传输通常是靠电缆或信道上的电流或电压变化实现的。如果一组数据的各数据位在多条线上同时被传输，这种传输方式称为并行通信，如图 1-1 所示。

并行数据传输的特点是：各数据位同时传输，传输速度快、效率高，多用在实时、快速的场合。

并行传输的数据宽度可以是 1~128 位，甚至更宽，但是有多少数据位就需要多少根数据线，因此传输的成本较高。在集成电路芯片的内部、同一插件板上各部件之间、同一机箱内各插件板之间的数据传输都是并行的。

并行数据传输只适用于近距离的通信，通常传输距离小于 30 m。

2. 串行通信

串行通信是指通信的发送端和接收端之间数据信息的传输是在单根数据线上，以每次一个二进制的 0、1 为最小单位逐位进行传输，如图 1-2 所示。

串行数据传输的特点是：数据传输按位顺序进行，仅需要一根传输线即可完成，节省传输线。与并行通信相比，串行通信还有较为显著的优点：传输距离长，可以从几米到几千米；串行通信的通信时钟频率容易提高；串行通信的抗干扰能力十分强，其信号间的互相干扰完

全可以忽略。但是串行通信传送速度比并行通信慢得多。

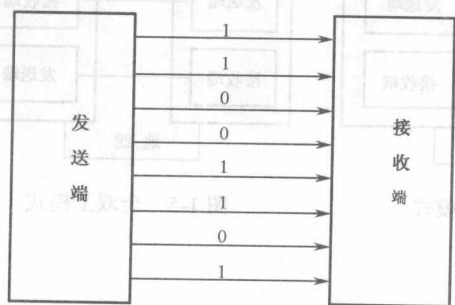


图 1-1 并行通信



图 1-2 串行通信

正是由于串行通信的连线少、成本低，因此它在数据采集和控制系统中得到了广泛的应用，产品也是多种多样的。

1.1.2 串行通信的工作模式

通过单线传输信息是串行数据通信的基础。数据通常是在两个站（点对点）之间进行传输，按照数据流的方向可分成三种传输模式：单工、半双工、全双工。

1. 单工模式

单工模式的数据传输是单向的。通信双方中，一方固定为发送端，另一方则固定为接收端。信息只能沿一个方向传输，使用一根传输线，如图 1-3 所示。

单工模式一般用在只向一个方向传输数据的场合。例如计算机与打印机之间的通信是单工模式，因为只有计算机向打印机传输数据，而没有相反方向的数据传输。还有在某些通信信道中，如单工无线发送等。

2. 半双工模式

半双工通信使用同一根传输线，既可发送数据又可接收数据，但不能同时进行发送和接收。

在任何时刻只能由其中的一方发送数据，另一方接收数据。因此半双工模式既可以使用一条数据线，也可以使用两条数据线，如图 1-4 所示。

半双工通信中每端需有一个收发切换电子开关，通过切换来决定数据向哪个方向传输。因为有切换，所以会产生时间延迟。信息传输效率低些。

3. 全双工形式

全双工数据通信分别由两根可以在两个不同的站点同时发送和接收的传输线进行传输，通信双方都能在同一时刻进行发送和接收操作，如图 1-5 所示。

在全双工模式中，每一端都有发送器和接收器，有两条传输线，可在交互式应用和远程监控系统中使用，信息传输效率较高。

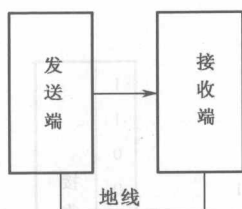


图 1-3 单工模式

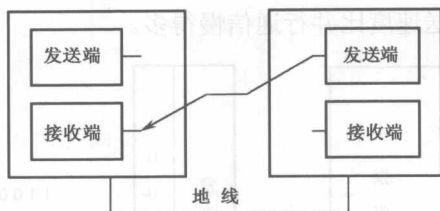


图 1-4 半双工模式

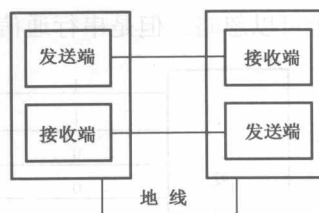


图 1-5 全双工模式

1.1.3 异步传输与同步传输

在串行传输中，数据是一位一位地按照到达的顺序依次进行传输的，每位数据的发送和接收都需要时钟来控制。发送端通过发送时钟确定数据位的开始和结束，接收端需要在适当的时间间隔对数据流进行采样来正确地识别数据。接收端和发送端必须保持步调一致，否则就会在数据传输中出现差错。为了解决以上问题，串行传输可采用以下两种方式：异步传输和同步传输。

1. 异步传输

在异步传输方式中，字符是数据传输单位。在通信的数据流中，字符之间异步，字符内部各位间同步。异步通信方式的“异步”主要体现在字符与字符之间通信没有严格的定时要求。在异步传输中，字符可以是连续地、一个个地发送，也可以是不连续地、随机地单独发送。在一个字符格式的停止位之后，立即发送下一个字符的起始位，开始一个新的字符的传输，这叫做连续的串行数据发送，即帧与帧之间是连续的。断续的串行数据传输是指在一帧结束之后维持数据线的“空闲”状态，新的起始位可在任何时刻开始。一旦传输开始，组成这个字符的各个数据位将被连续发送，并且每个数据位持续的时间是相等的。接收端根据这个特点与数据发送端保持同步，从而正确地恢复数据。收发双方则以预先约定的传输速度，在时钟的作用下，传输这个字符中的每一位。

2. 同步传输

在同步传输方式中，比特块以稳定的比特流的形式传输，数据被封装成更大的传输单位，称为帧。每个帧中含有多个字符代码，而且字符代码与字符代码之间没有间隙以及起始位和停止位。和异步传输相比，数据传输单位的加长容易引起时钟漂移。为了保证接收端能够正确地区分数据流中的每个数据位，收发双方必须通过某种方法建立起同步的时钟。例如，可以在发送器和接收器之间提供一条独立的时钟线路，由线路的一端（发送器或者接收器）定期地在每个比特时间中向线路发送一个短脉冲信号，另一端则将这些有规律的脉冲作为时钟。这种方法在短距离传输时表现良好，但在长距离传输中，定时脉冲可能会和信息信号一样受到破坏，从而出现定时误差。另一种方法是通过采用嵌有时钟信息的数据编码位向接收端提供同步信息。

1.1.4 硬件握手与软件握手

握手信号实际上是控制信号，用来控制数据的传输。通过握手信号，发送端可以得知接收端是否有数据要发送。接收端通过握手信号通知发送端它是否已经准备好了接收信号。握

手信号遵循某种协议。

当发送端和接收端处理数据的速度不一样时，可能会造成数据丢失。在传输中，如果发送端的发送速度大于接收端的接收速度，同时若接收端处理数据的速度不够快的话，那么接收端的缓冲区必定在一定时间后溢出，从而造成以后发送过来的数据不能进入缓冲区而丢失。发送端何时可以继续发送数据，何时必须暂停发送，从而让接收端有时间处理数据，称为流量控制，必须靠握手信号来解决这个问题。例如，打印机和计算机进行通信时，一些打印机打印速度可能跟不上计算机发送数据的速度，就必须要通过握手信号通知计算机暂停发送数据。

1. 硬件握手

在硬件握手中，发送端通过将某一个导线拉到高电平或者低电平，来表示发送端可以发送数据。接收端已经准备好接收数据之后，也把某一个导线拉到高电平或者是低电平，来通知发送端，发送端一直在检测这个信号。接收端可以在任何时候把这个信号变为无效，甚至是在接收一个数据块过程中。当发送端检测到这个信号变为无效之后，就必须停止本次发送，直到这个信号变为有效为止。

2. 软件握手

在软件握手中，以数据线上的数据信号来代替实际的硬件电路。这种方法用在直接连接或者通过调制解调器连接的两台计算机之间进行双向通信的场合。

对于软件握手现在已经建立了一些标准协议，其中最常用的是通信协议。通信协议是指通信双方的一种约定，约定包括对数据格式、同步方式、传输速度、传输步骤、检/纠错方式以及控制字符定义等问题进行统一规定，通信双方必须共同遵守，也叫做通信控制规程或称传输控制规程，它属于 OSI 七层参考模型中的数据链路层。

3. 硬件和软件结合的握手方法

如果使用个人计算机或终端通过调制解调器和主计算机进行通信，那么调制解调器使用硬件握手方法和个人计算机联系，主计算机和调制解调器之间也使用硬件握手方法，主计算机和个人计算机之间将使用软件握手方法进行联系。因此，个人计算机必须要编程，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。

一些计算机自动监测硬件握手信号，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。

1.1.5 串口通信的基本参数

串口的通信方式是将字节拆分成一个接着一个的位后再进行传输。接到此电位信号的一方再将此一个一个的位组合成原来的字节，如此形成一个字节的完整传输。

在传输进行的过程中，双方明确传输信息的具体方式，否则双方就会没有一套共同的译码方式，从而无法了解对方所传输过来的信息意义。因此双方为了进行通信，必须遵守一定的通信规则，这个共同的规则就是串口的初始化。

串口的初始化必须对以下几项参数进行设置。

1. 数据的传输速度

串行通信的传输受到通信双方配备性能及通信线路的特性所控制，收发双方必须按照同样的速率进行串行通信，即收发双方采用同样的波特率。我们通常将传输速度称为波特率，指的是串行通信中每秒所传输的数据位数，单位是 b/s。我们经常可以看到仪器或 Modem 的规格书上都写着 19 200 b/s、38 400 b/s，…，所指的就是传输速度。

就仪器或工业场合来说，4 800 b/s、9 600 b/s 是最常见的传输速度，现在的个人计算机所提供的串口的传输速度都可达到 15 200 b/s（甚至达到 921 600 b/s），若传输距离较近而设备也提供时，使用最高的传输速度也可以。

例如在某异步串行通信中，每传输一个字符需要 8 位，如果采用波特率 4 800 b/s 进行传输，则每秒可以传输 600 个字符。

2. 数据的传输单位

一般串口所传输的数据是字符型，若用来传输文件，则会使用二进制的数据形式。当使用字符型时，工业界使用到的有 ASCII 字符码及 JIS 字符码；ASCII 码使用了 8 位形成一个字符，而 JIS 码则以 7 位形成一个字符。我们可以发现，欧美的设备多使用 8 位的字节，而日本的设备多使用 7 位为一个字节。以实际的 RS-232 传输上看来，由于工业界常使用的 PLC 大多只是传输文字码，因此只要 7 位就可以将 ASCII 码的 0~127 码表达出来（ $2^7=128$ ，共有 128 种组合方式），所有的可见字符也在此范围内，所以只要 7 个数据位就够了。不同的情形下（根据所使用的协议），会使用到不同的传输单位。使用多少位合成一个字节必须先行确定。

3. 起始位与停止位

由于异步串行传输中并没有使用同步脉冲作为基准，故接收端完全不知道发送端何时将进行数据的传输。发送端准备要开始传输数据时，发送端会在所送出的字符前后分别加上高电位的起始位（逻辑 0）及低电位的停止位（逻辑 1），它们分别是所谓的起始位和停止位，也就是说，当发送端要开始传输数据时，便将传输线上的电位由低电位提升至高电位，而当传输结束后，再将电位降至低电位。接收端会因起始位的触发（因电压由低电位升至高电位）而开始接收数据；并因停止位的通知（因电压维持在低电位）而明确数据的字符信号已经结束；加入了起始位及停止位才能比较容易达到多字符的接收能力。起始位固定为 1 位，而停止位则有 1、1.5、2 位等多种选择，如何选择呢？只要通信双方协议通过即可，没有强制规定。

4. 校验位

为了预防错误的产生，使用校验位作为检查的机制。校验位是用来检查所传输数据的正确性的一种核对码，又可分成奇校验与偶校验两种，分别是检查字符码中 1 的数目是奇数或偶数。以偶校验为例，“A”的 ASCII 码是 41H（十六进制），将它以二进制表示时，是 01000001，其中 1 的数目是 2，因此校验位便是 0，使 1 的数目保持偶数；同样地，校验位是奇校验时，“A”的校验位便是 1，使 1 的数目保持在奇数。接收端重新计算奇偶校验位，如果新的计算值正确，那么表示正常。如果新的计算值错误，那么接收端就会发出一些指示，表示此次接收的数据有误。

1.2 串口通信及其标准

1.2.1 RS-232C 串口通信标准

RS-232C 是美国电子工业协会 (Electronic Industry Association, EIA) 于 1962 年公布并于 1969 年修订的串行接口标准, 它已经成为了国际上通用的标准。1987 年 1 月, RS-232C 经修改后, 正式改名为 EIA-232D。由于标准修改并不多, 因此现在很多厂商仍用旧的名称。

RS-232C 标准 (协议) 的全称是 EIA-RS-232C 标准, 其中 RS (Recommended Standard) 代表推荐标准, 232 是标识号, C 代表 RS-232 的最新一次修改, 它适合于数据传输速度在 0~20 000b/s 范围内的通信。这个标准对串行通信接口的有关问题, 如信号电平、信号线功能、电气特性、机械特性等都作了明确的规定。

目前 RS-232C 已成为数据终端设备 (Data Terminal Equipment, DTE, 如计算机) 和数据通信设备 (Data Communication Equipment, DCE, 如 Modem) 的接口标准。RS-232C 是 PC 与通信工业中应用最广泛的一种串行接口, 在 IBM PC 上的 COM1 和 COM2, 就是 RS-232C 接口。

利用 RS-232C 串行通信接口可实现两台个人计算机点对点的通信; 通过 RS-232C 接口可与其他外设 (如打印机、逻辑分析仪、智能调节仪、PLC 等) 近距离串行连接; 通过 RS-232C 接口连接调制解调器可远距离地与其他计算机通信; 将 RS-232C 接口转换为 RS-422 或 RS-485 接口, 可实现一台个人计算机与多台现场设备之间的通信。

1. 接口连接器

由于 RS-232C 并未定义连接器的物理特性, 因此, 出现了 DB-25 和 DB-9 各种类型的连接器, 其引脚的定义也各不相同。现在计算机上一般只提供 DB-9 连接器, 都为公头。相应的连接线上的串口连接器也有公头和母头之分, 见图 1-6。

作为多功能 I/O 卡或主板上提供的 COM1 和 COM2 两个串行接口的 DB-9 连接器, 它只提供异步通信的 9 个信号引脚, 见图 1-7, 各引脚的信号功能描述见表 1-1。

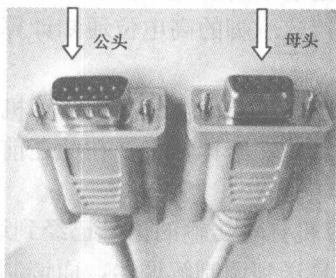


图 1-6 公头与母头串口连接

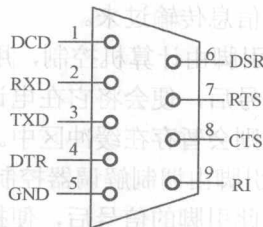


图 1-7 DB9 串口连接器

RS-232 的每一引脚都有它的作用, 也有它信号流动的方向。原来的 RS-232 是设计用来连接调制解调器作为传输用的, 因此它的引脚位意义通常也和调制解调器传输有关。

从功能来看, 全部信号线分为三类, 即数据线 (TXD、RXD)、地线 (GND) 和联络控制线 (DSR、DTR、RI、DCD、RTS、CTS)。

表 1-1 9 针串口的引脚功能

引 脚	符 号	通信方向	功 能
1	DCD	计算机 → 调制解调器	载波信号检测
2	RXD	计算机 ← 调制解调器	接收数据
3	TXD	计算机 → 调制解调器	发送数据
4	DTR	计算机 → 调制解调器	数据终端准备好
5	GND	计算机 = 调制解调器	信号地线
6	DSR	计算机 ← 调制解调器	数据装置准备好
7	RTS	计算机 → 调制解调器	请求发送
8	CTS	计算机 ← 调制解调器	清除发送
9	RI	计算机 ← 调制解调器	振铃信号指示

以下是这 9 个引脚的相关说明：

DCD：此引脚是由调制解调器（或其他 DCE，以下同）控制的，当电话接通之后，传输的信号被加载在载波信号上面，调制解调器利用此引脚通知计算机检测到载波，而当载波被检测到时才可保证此时是处于连接的状态。

一般情况下，如果计算机没有收到此信号，仍会有所响应，并将调制解调器挂起。

RXD：此引脚负责将传输过来的远程信息进行接收。在接收的过程中，由于信息是以数字形式传输的，用户可以在调制解调器的 RXD 指示灯上看到明灭交错，这是由于 0、1 交替导致的结果，也就是电位高低所产生的现象。

TXD：此引脚负责将计算机即将传输出去的信息传输出去。在传输的过程中，由于信息是以数字形式传输的，读者可以在调制解调器的 TXD 指示灯上看到明灭交错，这是由于 0、1 交替导致的结果，也就是电位高低所产生的现象。

DTR：此引脚由计算机（或其他 DTE，以下同）控制，用以通知调制解调器可以进行传输。高电位时表示计算机已经准备就绪，随时可以接收信息。

GND：此引脚为地线，作为计算机与调制解调器之间的参考基准。两端设备的地线准位必须一样，否则会产生地回路，使得信号因参考基准的不同而产生偏移，也会导致结果失常。RS-232 信息在传输上是采用单向式的信号传输方式，其特点是信号的电压基准由参考地线提供，因此传输双方的地线必须连接在一起，以避免基准不同而造成信息的错误。

DSR：此引脚由调制解调器控制，调制解调器用该引脚的高电位通知计算机一切均准备就绪，可以把信息传输过来。

RTS：此引脚由计算机控制，用以通知调制解调器马上传输信息到计算机。而当调制解调器收到此信号后，便会将它在电话线上收到的信息传输给计算机，在此之前若有信息传输到调制解调器则会暂存在缓冲区中。

CTS：此引脚由调制解调器控制，用以通知计算机打算传输的信息已经到达调制解调器。当计算机收到此引脚的信号后，便把准备传输的信息送到调制解调器，而调制解调器则将计算机传输过来的信息通过电话线路送出。

RI：调制解调器通知计算机有电话进来，是否接听电话则由计算机决定。如果计算机设置调制解调器为自动应答模式，则调制解调器在听到铃响后便会自动接听电话。

上述控制信号线何时有效，何时无效的顺序表示了接口信号的传输过程。例如，只有当 DSR 和 DTR 都处于有效（ON）状态时，才能在 DTE 和 DCE 之间进行传输操作。若 DTE