

实例详解丛书

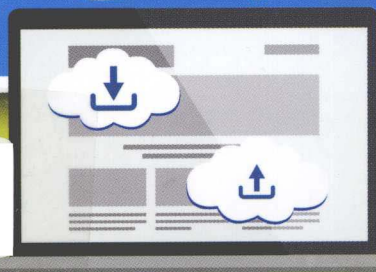
· 完整实例代码 · 程序运行录屏
· 系统测试录像 · 软、硬件资源



Visual C++

串口通信及测控应用实例详解

主 编 刘长征
副主编 张荣华 党 媚 李江全



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

实例详解丛书

Visual C++串口通信及 测控应用实例详解

主 编 刘长征
副主编 张荣华 党 媚 李江全

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从应用角度出发, 首先介绍与串口通信程序设计有关的基础知识, 包括串行通信概述、Visual C++ 串口通信基础、计数制与编码; 然后通过 PC、远程 I/O 模块、三菱/西门子 PLC、单片机、智能仪器等典型串口设备, 详细讲解 Visual C++ 开发串口通信及测控程序的设计方法。每个设计实例由设计任务、线路连接、串口通信调试、程序界面设计、程序设计详解、系统运行测试等部分构成。

本书内容丰富, 论述深入浅出, 有较强的实用性和可操作性, 可供自动化、计算机应用、电子信息、机电一体化、测控仪器等专业的大学生、研究生以及计算机测控系统研发的工程技术人员学习和参考。

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

Visual C++ 串口通信及测控应用实例详解 / 刘长征主编. —北京: 电子工业出版社, 2014.1
(实例详解丛书)

ISBN 978-7-121-21916-0

I. ①V… II. ①刘… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 276127 号

责任编辑: 田宏峰 特约编辑: 牛雪峰

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: 24.25 字数: 610 千字

印 次: 2014 年 1 月第 1 次印刷

印 数: 4 000 册 定价: 68.00 元 (含 DVD 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 zltts@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

目前,许多检测仪表、在线分析仪、工控机等均采用 RS-232 或 RS-485 协议通过串口与上位 PC 进行通信,尤其在钢铁、有色、热工等领域,用户通过串行通信可以实时访问远程设备的工作状况,对设备运行参数进行优化,通过指令反馈,控制仪器设备的正常运行,从而实现生产的节能与质量控制。此外,串行通信在实验教学、科学研究中也发挥重要的作用,科技工作者采用串行通信接口能够方便快捷地对温度、浓度、流量、压力等过程参数进行实时监测与分析,大大减少工业试验次数,在低成本、高质量前提下完成设备的调试,因此,掌握串行通信技术是广大科研人员的迫切需求。

鉴于此,我们编写本书,采用高效率的 Visual C++6.0(以下简称 VC++)作为上位机开发语言设计串口通信程序,并对串行通信编程技术进行详细讲解,使读者能够快速掌握该项技术并应用到科研生产环节。

就串口开发而言,VC++提供了串口通信控件 MSComm,通过对此控件的属性和事件进行相应的编程操作,就可以轻松地实现串口通信;此外,Windows 还提供了丰富的 API 函数,VC++调用与串口有关的 API 函数也可实现串口通信。在实践中,使用 MSComm 控件实现通信的方法比调用 API 函数的方法更加方便、快捷,而且用较少的代码可以实现相同的功能,从而使编程效率大大提高,也减少因编程不当而导致的系统不稳定。

本书从应用角度出发,首先介绍与串口通信程序设计有关的基础知识,包括串行通信概述、VC++串口通信基础、计数制与编码;然后通过 PC、远程 I/O 模块、三菱/西门子 PLC、单片机、智能仪器等典型串口设备,详细讲解 VC++开发串口通信及测控程序的设计方法。每个设计实例由设计任务、线路连接、串口通信调试、程序界面设计、程序设计详解、系统运行测试等部分构成。

本书内容丰富,论述深入浅出,有较强的实用性和可操作性,可供自动化、计算机应用、电子信息、机电一体化、测控仪器等专业的大学生、研究生以及计算机测控系统研发的工程技术人员学习和参考。

本书由石河子大学张荣华编写第 1、5 章,王磊编写第 3、4 章,刘长征编写第 2、6 章,欧阳异能编写第 7 章,李江全编写第 8 章,西安航空职业技术学院党媚编写第 9、10 章,全书由刘长征担任主编,张荣华、党媚、李江全担任副主编。参与本书编写工作的还有王建平、张有强、李传峰、王丽、万畅、廖结安、张洪洲、李向阳、刘新英等老师,编者借此机会对他们致以深深的谢意。

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

2013 年 10 月



目 录

第 1 章 串行通信概述	(1)
1.1 串行通信的基本概念	(1)
1.1.1 通信与通信方式	(1)
1.1.2 串行通信的数据传输方式	(3)
1.1.3 串行通信的基本参数	(4)
1.1.4 串行通信的基本方式	(4)
1.2 串行通信协议	(5)
1.2.1 异步串行通信协议	(5)
1.2.2 同步串行通信协议	(8)
1.3 串行通信的接口标准	(10)
1.3.1 RS-232C 串口通信标准	(11)
1.3.2 RS-422/485 串口通信标准	(13)
1.4 串行通信线路连接	(15)
1.4.1 短距离线路连接	(15)
1.4.2 长距离线路连接	(15)
1.5 串口调试工具	(17)
1.5.1 超级终端程序	(17)
1.5.2 串口调试助手	(18)
第 2 章 Visual C++与串口通信	(19)
2.1 串口通信控件 MSComm	(19)
2.1.1 MSComm 控件处理通信的方式	(19)
2.1.2 MSComm 控件的添加	(20)
2.1.3 MSComm 控件的常用属性	(21)
2.1.4 MSComm 控件的 OnComm 事件	(27)
2.2.5 MSComm 控件通信步骤	(29)
2.2 串口通信 API 函数	(30)
2.2.1 动态链接库与 API 函数	(30)
2.2.2 VC++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	(35)
2.2.5 利用 API 函数实现串口通信的查询方法	(37)
2.2.6 使用 API 函数实现 PC 与 PC 串口通信实例	(38)
第 3 章 计数制与编码	(44)
3.1 计数制	(44)

3.1.1	计数制概述	(44)
3.1.2	十进制与二进制	(45)
3.1.3	八进制与十六进制	(46)
3.2	计数制转换及其程序设计	(47)
3.2.1	二进制与十进制的转换	(47)
3.2.2	二进制与八进制的转换	(51)
3.2.3	二进制与十六进制的转换	(53)
3.2.4	八进制与十进制的转换	(55)
3.2.5	十六进制与十进制的转换	(57)
3.2.6	八进制与十六进制的转换	(59)
3.3	字符编码	(62)
3.3.1	非数值数据	(62)
3.3.2	BCD 码	(62)
3.3.3	ASCII 编码	(63)
3.3.4	ASCII 编码与十六进制的转换	(64)
第 4 章	PC 与 PC 串口通信及应用	(66)
4.1	PC 中的串行端口	(66)
4.1.1	查看串行端口信息	(66)
4.1.2	虚拟串口的使用	(67)
4.2	PC 与 PC 串口通信程序设计	(69)
4.2.1	设计任务	(69)
4.2.2	线路连接	(69)
4.2.3	串口通信调试	(70)
4.2.4	程序界面设计	(71)
4.2.5	程序设计详解	(72)
4.2.6	系统运行测试	(75)
4.3	PC 双串口互通信程序设计	(75)
4.3.1	设计任务	(75)
4.3.2	线路连接	(76)
4.3.3	串口通信调试	(76)
4.3.4	程序界面设计	(77)
4.3.5	程序代码设计详解	(79)
4.3.6	系统运行测试	(82)
第 5 章	远程 I/O 模块串口通信及测控应用	(83)
5.1	典型分布式 I/O 模块简介	(83)
5.1.1	集散控制系统的结构与特点	(83)
5.1.2	ADAM4000 远程数据采集控制系统	(85)
5.1.3	ADAM4000 系列模块简介	(87)
5.1.4	ADAM4000 系列模块的软件安装	(93)

5.2	模拟电压采集程序设计	(96)
5.2.1	设计任务	(96)
5.2.2	线路连接	(97)
5.2.3	串口通信调试	(97)
5.2.4	程序界面设计	(98)
5.2.5	PC 端程序设计详解	(99)
5.2.6	系统运行测试	(101)
5.3	模拟电压输出程序设计	(102)
5.3.1	设计任务	(102)
5.3.2	线路连接	(102)
5.3.3	串口通信调试	(103)
5.3.4	程序界面设计	(103)
5.3.5	PC 端程序设计详解	(104)
5.3.6	系统运行测试	(105)
5.4	开关信号输入程序设计	(106)
5.4.1	设计任务	(106)
5.4.2	线路连接	(106)
5.4.3	串口通信调试	(106)
5.4.4	程序界面设计	(107)
5.4.5	PC 端程序设计详解	(108)
5.4.6	系统运行测试	(112)
5.5	开关信号输出程序设计	(113)
5.5.1	设计任务	(113)
5.5.2	线路连接	(113)
5.5.3	串口通信调试	(114)
5.5.4	程序界面设计	(114)
5.5.5	PC 端程序设计详解	(115)
5.5.6	系统运行测试	(118)
5.6	温度测控程序设计	(118)
5.6.1	设计任务	(118)
5.6.2	线路连接	(119)
5.6.3	串口通信调试	(119)
5.6.4	程序界面设计	(120)
5.6.5	PC 端程序设计详解	(122)
5.6.6	系统运行测试	(127)
第 6 章	三菱 PLC 串口通信及测控应用	(129)
6.1	三菱 PLC 特殊功能模块与通信协议	(129)
6.1.1	FX _{2N} 系列 PLC 的特殊功能模块	(129)
6.1.2	三菱 PLC 编程口通信协议	(139)

6.2	模拟电压采集程序设计	(146)
6.2.1	设计任务	(146)
6.2.2	线路连接	(146)
6.2.3	PLC 端电压输入程序	(147)
6.2.4	串口通信调试	(149)
6.2.5	PC 端程序界面设计	(150)
6.2.6	PC 端程序设计详解	(151)
6.2.7	系统运行测试	(156)
6.3	模拟电压输出程序设计	(156)
6.3.1	设计任务	(157)
6.3.2	线路连接	(157)
6.3.3	PLC 端电压输出程序	(158)
6.3.4	串口通信调试	(160)
6.3.5	PC 端程序界面设计	(161)
6.3.6	PC 端程序设计详解	(162)
6.3.7	系统运行测试	(165)
6.4	开关信号输入程序设计	(165)
6.4.1	设计任务	(165)
6.4.2	线路连接	(166)
6.4.3	串口通信调试	(166)
6.4.4	PC 端程序界面设计	(167)
6.4.5	PC 端程序设计详解	(168)
6.4.6	系统运行测试	(173)
6.5	开关信号输出程序设计	(174)
6.5.1	设计任务	(174)
6.5.2	线路连接	(174)
6.5.3	串口通信调试	(175)
6.5.4	PC 端程序界面设计	(176)
6.5.5	PC 端程序设计详解	(177)
6.5.6	系统运行测试	(179)
6.6	温度测控程序设计	(180)
6.6.1	设计任务	(180)
6.6.2	线路连接	(180)
6.6.3	PLC 端温度测控程序	(181)
6.6.4	串口通信调试	(184)
6.6.5	PC 端程序界面设计	(185)
6.6.6	PC 端程序设计详解	(186)
6.6.7	系统运行测试	(190)

第 7 章 西门子 PLC 串口通信及测控应用	(192)
7.1 西门子 PLC 模拟量扩展模块与通信协议	(192)
7.1.1 西门子 S7-200 PLC 的模拟量扩展模块	(192)
7.1.2 西门子 S7-200 PLC PPI 通信协议	(197)
7.2 模拟电压采集程序设计	(200)
7.2.1 设计任务	(201)
7.2.2 线路连接	(201)
7.2.3 PLC 端电压输入程序	(202)
7.2.4 串口通信调试	(203)
7.2.5 PC 端程序界面设计	(204)
7.2.6 PC 端程序设计详解	(205)
7.2.7 系统运行测试	(209)
7.3 模拟电压输出程序设计	(210)
7.3.1 设计任务	(210)
7.3.2 线路连接	(210)
7.3.3 PLC 端电压输出程序	(211)
7.3.4 串口通信调试	(212)
7.3.5 PC 端程序界面设计	(213)
7.3.6 PC 端程序设计详解	(214)
7.3.7 系统运行测试	(218)
7.4 开关信号输入程序设计	(218)
7.4.1 设计任务	(218)
7.4.2 线路连接	(219)
7.4.3 串口通信调试	(219)
7.4.4 PC 端程序界面设计	(220)
7.4.5 PC 端程序设计详解	(221)
7.4.6 系统运行测试	(227)
7.5 开关信号输出程序设计	(227)
7.5.1 设计任务	(227)
7.5.2 线路连接	(228)
7.5.3 串口通信调试	(228)
7.5.4 PC 端程序界面设计	(229)
7.5.5 PC 端程序设计详解	(230)
7.5.6 系统运行测试	(234)
7.6 温度测控程序设计	(235)
7.6.1 设计任务	(235)
7.6.2 线路连接	(235)
7.6.3 PLC 端温度测控程序	(236)
7.6.4 串口通信调试	(239)

7.6.5	PC 端程序界面设计	(240)
7.6.6	PC 端程序设计详解	(241)
7.6.7	系统运行测试	(247)
第 8 章	单片机串口通信及测控应用	(248)
8.1	典型单片机开发板简介	(248)
8.1.1	单片机测控系统的组成	(248)
8.1.2	单片机开发板 B 的功能	(251)
8.1.3	单片机开发板 B 的主要电路	(252)
8.2	模拟电压采集程序设计	(255)
8.2.1	设计任务	(255)
8.2.2	线路连接	(255)
8.2.3	单片机端 C51 程序设计详解	(257)
8.2.4	串口通信调试	(261)
8.2.5	PC 端程序界面设计	(261)
8.2.6	PC 端程序设计详解	(263)
8.2.7	系统运行测试	(266)
8.3	模拟电压输出程序设计	(266)
8.3.1	设计任务	(266)
8.3.2	线路连接	(267)
8.3.3	单片机端 C51 程序设计详解	(268)
8.3.4	串口通信调试	(271)
8.3.5	PC 端程序界面设计	(272)
8.3.6	PC 端程序设计详解	(273)
8.3.7	系统运行测试	(274)
8.4	开关信号输入程序设计	(274)
8.4.1	设计任务	(274)
8.4.2	线路连接	(274)
8.4.3	单片机端 C51 程序设计详解	(275)
8.4.4	串口通信调试	(278)
8.4.5	PC 端程序界面设计	(278)
8.4.6	PC 端程序设计详解	(280)
8.4.7	系统运行测试	(283)
8.5	开关信号输出程序设计	(283)
8.5.1	设计任务	(284)
8.5.2	线路连接	(284)
8.5.3	单片机端 C51 程序设计详解	(285)
8.5.4	串口通信调试	(286)
8.5.5	PC 端程序界面设计	(287)
8.5.6	PC 端程序设计详解	(289)

8.5.7 系统运行测试	(291)
8.6 温度测控程序设计	(291)
8.6.1 设计任务	(291)
8.6.2 线路连接	(292)
8.6.3 单片机端 Pt100 温度检测 C51 程序设计详解	(293)
8.6.4 单片机端 DS18B20 温度检测 C51 程序设计详解	(297)
8.6.5 串口通信调试	(302)
8.6.6 PC 端程序界面设计	(303)
8.6.7 PC 端程序设计详解	(305)
8.6.8 系统运行测试	(311)
第 9 章 智能仪器串口通信及测控应用	(312)
9.1 典型智能仪器简介	(312)
9.1.1 智能仪器的结构与特点	(312)
9.1.2 XMT-3000A 型智能仪器的通信协议	(313)
9.2 单台智能仪器温度测量程序设计	(316)
9.2.1 设计任务	(316)
9.2.2 线路连接	(316)
9.2.3 串口通信调试	(317)
9.2.4 程序界面设计	(318)
9.2.5 PC 端程序设计详解	(319)
9.2.6 系统运行测试	(323)
9.3 多台智能仪器温度测量程序设计	(323)
9.3.1 设计任务	(324)
9.3.2 线路连接	(324)
9.3.3 串口通信调试	(325)
9.3.4 程序界面设计	(325)
9.3.5 PC 端程序设计详解	(327)
9.3.6 系统运行测试	(333)
第 10 章 GSM 短信模块串口通信及测控应用	(335)
10.1 GSM 网络短信测控技术	(335)
10.1.1 GSM 短信测控系统的特点与组成	(335)
10.1.2 AT 指令介绍	(339)
10.1.3 超级终端的使用	(343)
10.2 短信发送与接收程序设计	(348)
10.2.1 设计任务	(348)
10.2.2 线路连接	(348)
10.2.3 单片机端 C51 短信发送程序设计详解	(349)
10.2.4 单片机端 C51 短信接收程序设计详解	(357)
10.2.5 PC 端短信收发程序设计	(363)
参考文献	(373)

第 1 章

串行通信概述

目前计算机的串行通信应用十分广泛，串口已成为计算机的必需部件和接口之一。串行接口技术简单成熟、性能可靠、价格低廉，所要求的软/硬件环境或条件都很低，广泛应用于计算机控制相关领域，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。

本章对串行通信的基本概念、串行通信的接口标准及串行通信线路连接进行简要介绍。

1.1 串行通信的基本概念

1.1.1 通信与通信方式

什么是通信？简单地说，通信就是两个人之间的沟通，也可以说是两个设备之间的数据交换。人们之间的通信使用了诸如电话、书信等工具进行；而设备之间的通信则是使用电信号。最常见的信号传递就是使用电压的改变来达到表示不同状态的目的。以计算机为例，高电位代表了一种状态，而低电位代表了另一种状态，在组合了很多电位状态后就形成了两种设备之间的数据交换。

在计算机内部，所有的数据都是使用位来存储的，每一位都是电位的一个状态（计算机中以 0、1 表示）；计算机内部使用组合在一起的 8 位代表一般所使用的字符、数字及一些符号，如 01000001 就表示一个字符。一般来说，必须传递这些字符、数字或符号才能算是数据交换。

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

如果一组数据的各数据位在多条线上同时被传输，这种传输被称为并行通信，如图 1-1 所示，使用了 8 条信号线一次将一个字符 11001101 全部传输完毕。

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。例如，打印机端口就是一个典型的例子。

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，通常小于 30 m。

2. 串行通信

串行通信是将数据的各个位一位一位地通过单条 1 位宽的传输线按顺序分时传输，即通信双方一次传输一个二进制位，以每次一个二进制数 0、1 为最小单位逐位进行传输，如图 1-2 所示。

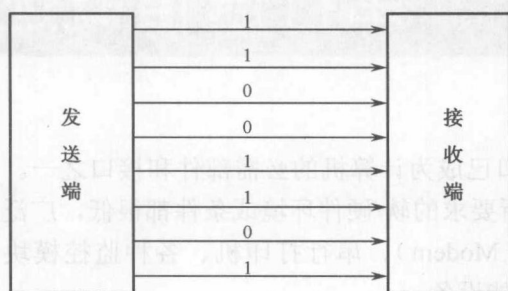


图 1-1 并行通信

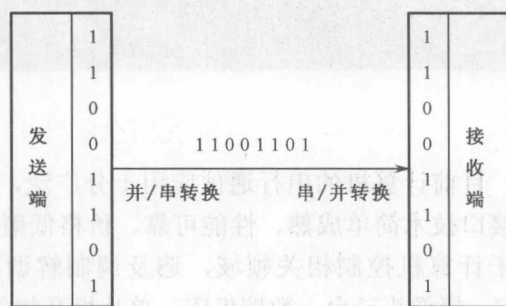


图 1-2 串行通信

串行数据传输的特点是：数据传输按位顺序进行，只需要一根传输线即可完成，可节省传输线。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；在长距离内串行数据传输速率会比并行数据传输速率快；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。

正是由于串行通信的接线少、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，产品也多种多样。计算机和单片机间都采用串行通信方式。

3. 串行通信与并行通信比较

在实际应用中，串行通信比并行通信要多，串行通信不仅广泛应用于主机与键盘和鼠标等低速外部设备之间，而且越来越多地用于中低速、甚至高速外部设备与主机的通信，计算机与计算机之间的通信绝大多数都使用串行通信。

这是因为随着通信速率的提高和通信距离的延长，并行通信中出现的信号变形与抖动及互相干扰等问题制约了并行通信的应用范围。在进行并行通信时，虽然 8 位或 16 位的数据同时从发送端发送，但在它们到达接收端的时候，传播延时将导致某些信号比别的信号提前到达。随着通信距离的延长，最先和最后到达的数据位之间的时间差也会迅速增加。

此外，在时钟速率很高时，并行信号之间可能会互相干扰，这些都使得并行通信的通信时钟速率受到限制。而在串行通信中，数据是按时间逐位发送的，由于每个位到达的时间没有规定，因此信号时钟频率可以大幅度提高。

并行通信与串行通信各有其应用场合。

(1) 从通信距离上看：并行通信适宜于近距离的数据传输，通常小于 30 m，而串行通信适宜于远距离传输，可以从几米到数千千米。

(2) 从通信速率上看：一般应用中，在短距离内，并行接口的数据传输速率显然比串行接口的传输速率高得多，但长距离内串行数据传输速率会比并行数据传输速率快。由于串行通信的通信时钟频率较并行通信容易提高，因此许多高速外部设备，如数字摄像机与计算

机之间的通信也往往使用串行通信方式。

(3) 从抗干扰性能上看：串行通信由于只有一两根信号线，信号间的互相干扰完全可以忽略。

(4) 从设备和费用上看：随着大规模和超大规模集成电路的发展，逻辑器件价格趋低，而通信线路费用趋高，因此对远距离通信而言，串行通信的费用显然会低得多；另一方面串行通信还可利用现有的电话网络来实现远程通信，降低了通信费用。

串行通信与并行通信相比，虽然有许多优点，但也随之带来了数据的串/并转换及并/串转换、数据格式的要求以及位计数等问题，使之实现起来比并行通信更为复杂。

1.1.2 串行通信的数据传输方式

通过单线传输信息是串行数据通信的基础。数据通常是在二个站（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. 单工形式

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

单工形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形式，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. 半双工形式

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，因此半双工形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。

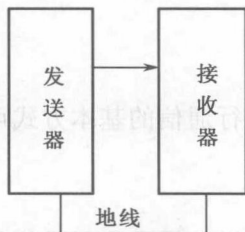


图 1-3 单工形式

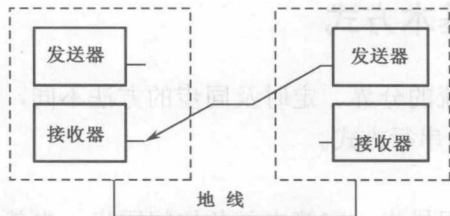


图 1-4 半双工形式

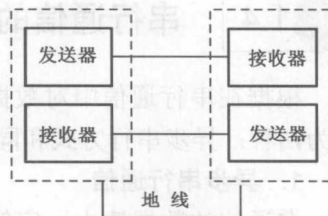


图 1-5 全双工形式

在全双工方式中，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。

1.1.3 串行通信的基本参数

串行端口的通信方式是将字节拆分成一个接着一个的位再传送出去。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，在数据传输时，应在通信端口的初始化时设置几个通信参数。

1. 波特率

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左右，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一秒所传输的数据位数，单位是 bps。我们经常可以看到仪器或 Modem 的规格书上都写着 19 200 bps、38 400 bps、…，所指的就是传输速率。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 bps 进行传输，则每秒可以传输 600 个字符。

2. 数据位

当接收设备收到起始位后，紧接着就会收到数据位，数据位的个数可以是 5、6、7 或 8 位数据。在字符数据传输的过程中，数据位从最低有效位开始传输。

3. 起始位

在通信线上，没有数据传输时处于逻辑“1”状态。当发送设备要发送一个字符数据时，首先发出一个逻辑“0”信号，这个逻辑低电平就是起始位。起始位通过通信线传向接收设备，当接收设备检测到这个逻辑低电平后，就开始准备接收数据位信号。因此，起始位所起的作用就是表示字符传输的开始。

4. 停止位

在奇偶校验位或者数据位（无奇偶校验位时）之后是停止位。它可以是 1 位、1.5 位或 2 位，停止位是一个字符数据的结束标志。

5. 奇偶校验位

数据位发送完之后，就可以发送奇偶校验位。奇偶校验用于有限差错检验，通信双方在通信时约定一致的奇偶校验方式。就数据传输而言，奇偶校验位是冗余位，但它表示数据的一种性质，这种性质虽然只用于检错但很容易实现。

1.1.4 串行通信的基本方式

根据在串行通信中对数据流的分界、定时及同步的方法不同，串行通信的基本方式可分为两种：异步串行方式和同步串行方式。

1. 异步串行通信

在通信的数据流中，字符间异步，字符内部各位间同步。也就是说，异步串行通信是以字符为信息单位传输的，每个字符作为一个独立的信息单位（1 帧数据），可以随机出现在数据流中，即发送端发出的每个字符在数据流中出现的时间是任意的，接收端预先并不知道。这就是说，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。然而，一旦传输开始，收发双方则以预先约定的传输速率，在时钟的作用下，传输这个

字符中的每一位,即要求位与位之间有严格而精确的定时,也就是说,异步通信在传输同一个字符的每一位时,是同步的。因此,所谓异步通信,主要指字符与字符之间的传输是异步的,而字符内部位与位之间还是基本同步传输的。

2. 同步串行通信

数据流中的字符与字符之间和字符内部的位与位之间都同步。同步串行通信是以数据块(字符块)为信息单位传输,而每帧信息包括成百上千个字符,因此传输一旦开始,要求每帧信息内部的每一位都要同步。也就是说,同步通信不仅字符内部的位传输是同步的,字符与字符之间的传输也应该是同步的,这样才能保证收发双方对每一位都同步。显然,这种通信方式对时钟同步要求非常严格,为此,收发两端必须使用同一时钟来控制数据块传输中字符与字符和字符内部位与位之间的定时。

无论是异步串行通信还是同步串行通信,为了保证通信的正确,发送方和接收方事先必须有一个双方共同遵守的协定,如数据传输格式、起始标志、结束标志、校验方式和通信波特率等。异步串行通信一般用在数据发送时间不能确知、发送数据不连续、数据量较少和数据速率较慢的场合。而同步串行通信则适于用在要求快速、连续传输大量数据的场合。

1.2 串行通信协议

数据通信中,在收/发器之间传输的是一组二进制的“0”和“1”位串。但它们在不同的位置可能有不同的含义,有的只是用于同步,有的代表了通信双方的地址,有的是一些控制信息,有的则是通信中真正要传输的数据,还有的是为了差错控制而附加上去的冗余位。这些都需要在通信协议中事先约定好,以形成一种收发双方共同遵守的格式。

通信协议又称通信规程,是指通信双方对数据传输控制的一种约定。约定中包括对数据格式、同步方式、传输速率、传输步骤、检纠错方式以及控制字符定义等问题作出统一规定,通信双方必须共同遵守,它也叫做链路控制规程。

1.2.1 异步串行通信协议

异步串行通信的“异步”主要体现在字符与字符之间,而同一字符内部各位是同步的。可见,为了确保异步通信的正确性,必须找到一种方法,使收发双方在随机传输的字符与字符间实现同步。这种方法就是在字符格式中设置起始位和停止位,即在一个字符正式发送之前先发一个起始位,该字符结束时再发送一个停止位。接收器检测到起始位便知道字符到达,开始接收字符,检测到停止位则知道字符已结束。由于这种通信协议是靠起始位和停止位来进行字符同步的,因此有时也称之为起止式协议。

异步通信采用电报通信中的电传打字机(TTY)规程,每帧信息格式如图 1-6 所示。异步通信信息由以下几部分组成:

(1) 1 位起始位。逻辑“0”信号,该位及该帧各位持续时间均为波特率的倒数。

(2) 5~8 位数据位。紧接着起始位之后,数据位个数可以是 5、6、7、8,构成一个字符。通常用 ASCII 码,也可采用 EBCD 码和电报码等。从最低位开始传输,靠时钟定位。

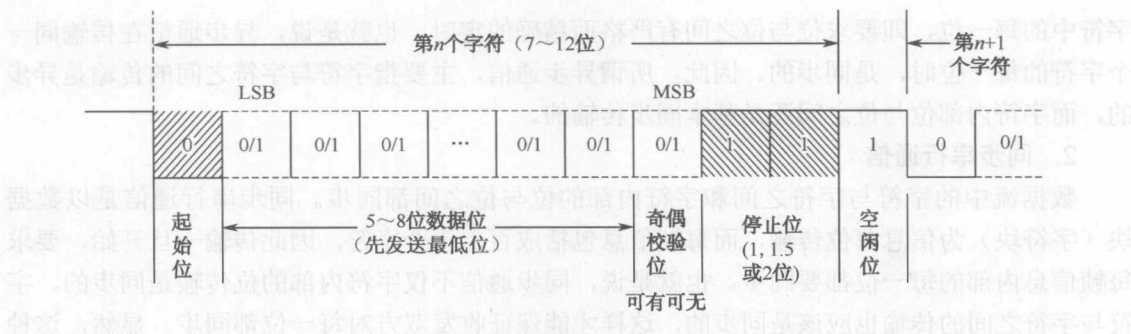


图 1-6 异步串行通信数据格式

(3) 0 或 1 位奇偶校验位。数据位加上这一位后，使得“1”的位数应为偶数（偶校验）或奇数（奇校验），以此来校验数据传输的正确性。当数据传输距离较近或数据传输速率较低时，通信双方可约定不用添加奇偶校验位。

(4) 停止位。它是一个字符数据的结束标志。可以是 1 位、1.5 位和 2 位的逻辑“1”。接收设备收到停止位之后，通信线便又恢复逻辑“1”状态，直至下一个字符数据的起始位到来。

(5) 空闲位。当线路上没有数据传输时，处于逻辑“1”状态。

异步通信要求在发送每一个字符时都要在数据位的前面加上 1 位起始位，在数据位后要有 1 位或 1.5 位或 2 位的停止位。在数据位和停止位之间可以有 1 位奇偶校验位，数据位可以为 5~8 位长，起始位（又叫空号）为“0”，停止位（又叫传号）为“1”。字符之间允许有不定长度的空闲位，空闲位均称为传号。这样在串行位流中以起始位和停止位将一个字符区分开来。

传输开始后，接收设备不断地检测传输线，当在检测到一系列的“1”之后检测到一个“0”时就确认一个字符开始，于是以位时间（1/波特率）为间隔移位接收规定的数据位和奇偶校验位，拼成一个字符的并行字节。这之后应接收所规定位长的停止位“1”，若没有收到即为“帧出错”。只有既无帧出错又无奇偶错才算正确地接收到一个字符。一个字符接收完毕，接收设备又继续测试传输线，监视“0”电平的到来和下一字符的开始，直到全部数据传输完毕。

由于异步通信系统中接收器和发送器使用的是各自独立的控制时钟，尽管它们的频率要求选得相同，但实际上总不可能真正严格相同，两者的上下边沿不可避免地会出现一定的时间偏移。为了保证数据的正确传输，不致因收/发双方时钟的相对误差而导致接收端的采样错误，除了如上所述，采用相反极性的起始位和停止位/空闲位提供准确的时间基准外，通常还采取以下两项措施。

(1) 接收器在每位的中心采样，以获得最大的收发时钟频率偏差容限。这样在 7~12 位的整个字符传输期间，收发双方时钟的偏差最多可允许有正、负半个位周期，只要不超过它，就不会产生采样错误。也就是说，要求收发时钟的误差容限不超过 4.17%（按每个字符最多 12 个位算）即可。显然，这个要求是很容易实现的。为了保证在每位的中心位置采样，在准确知道起始位前沿的前提下，接收器在起始位前沿到来后，先等半个位周期采样一次，然后每过一个位周期采样一次，直到收到停止位。