

C++ Builder 与 RS-232 串行通信控制

范逸之 江文贤 陈立元 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书使用 C++ Builder 详细说明自动化的基础——RS-232 串行通信,涉及以下内容:将 C++ Builder 的应用扩展到硬件控制,以 C++ Builder 开发串行通信的组件,控制调制解调器,通过串行端口来控制仪器,使串行通信扩展至分布式监控系统等。本书以循序渐进的方式介绍串行通信理论,从简单的串行概念和实验到复杂的仪器控制,都提供了完整的 C++ Builder 范例。

本书面向操作,重点突出,侧重应用。书中提供了完整的设计步骤和程序代码,并给予详细的注释,既适合有程序设计基础的读者使用,也适合没有程序设计基础的读者学习。本书可供自动化控制和通信领域的开发人员和其他相关技术人员使用或参考。

本书繁体字版书名为《C++ Builder 与 RS-232 串列通讯控制》,由文魁资讯股份有限公司出版,版权属范逸之、江文贤、陈立元所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号:图字 01-2002-1744 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:C++ Builder 与 RS-232 串行通信控制

作 者:范逸之 江文贤 陈立元 编著

责任编辑:桑任松

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者:

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:27 字数:647 千字

版 次:2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

书 号:ISBN 7-900643-05-2

印 数:0001~5000

定 价:49.00 元(配 1 张光盘)

前 言

当今世界是一个高度自动化的社会，各式各样的设备都可以和计算机联机，而最简单的自动化联机方式就是使用串行通信。串行通信存在的时间已经不少了，随着科技的发展，它不仅没有被取代，其应用范围反而越来越广。这是因为串行通信设备硬件的实现比较简单，而且使用也很简便。

C++ Builder 给程序设计人员提供了相当好的开发环境，开发 VCL 组件也相当简单。作者将串行通信组件的开发过程写下来，让开发者学习使用 C++ Builder 开发串行通信的详细过程。作者根据工作经验，通过简单易懂的编排方式，将串行通信的课题分若干个章节。有关程序设计部分，尽量在书中交代清楚，希望读者可以了解设计流程中的细节，同时避免按书本操作与实际上机不符的情况发生。

本书不仅说明了串行通信的概念，也说明了在窗口环境中设计通信程序的一些技巧与注意事项。如果读者已经有程序设计的基础，看本书应该会比较轻松；如果不清楚程序设计的细节，书中也会交代。不管怎样，本书应是一本颇具实用价值的参考书。

书中各章的主要内容简述如下：

第 1 章：基本概念，主要介绍串行通信的相关硬件概念。

第 2 章：C++ Builder 简介，介绍 C++ Builder 的常用组件。如果读者已了解 C++ Builder，可以跳过此章。

第 3 章：说明如何通过 Win32 API 建立串行通信的工程，并有相关的基础实验工程。

第 4 章：将第 3 章说明的方法封装起来，以 VCL 的方式制成组件，并详述其开发的流程及方法，方便后面的程序开发。

第 5 章：串行通信的传输有字符及字节两种方式，本章详细说明它们之间的差别并以实验进行验证。

第 6 章：讨论串行数据的处理方式，重点是文字、图形及字节的显示和处理。

第 7 章：介绍调制解调器的控制，本章以 C++ Builder 工程进行详细说明，以介绍拨号调制解调器及专线调制解调器的控制方式。

第 8 章：C++ Builder 可以完成大部分的串行通信事务，还有其他通信组件可以辅助 C++ Builder 进行更好的串行通信，将 C++ Builder 的串行通信功能进一步提升。本章将说明如何运用第三方组件提升 C++ Builder 的串行通信功能。

第 9 章：使用串行通信的方式可以达到仪器控制的目的，本章使用数种市面上可见的仪器，经过串行端口实现仪器控制的自动化。

第 10 章：分布式的串行通信可以使串行端口功能向监控系统扩展，本章简单说明如何

使用 C++ Builder 控制分布式系统。

由于作者才疏学浅，所书所言均是工作及教学经验的累积整理所成，若有未臻理想之处，还请各位读者不吝指正。

范逸之 (tom@hwada.com.tw)

江文贤 (andy@hwada.com.tw)

陈立元 (0930912770@fetnet.net)

2001 年 12 月于桃园

目 录

第 1 章 基本概念.....	1	2.1.6 C++ Builder 和操作系统的 关系	37
1.1 通信	1	2.2 常用组件介绍	39
1.1.1 数据发送	1	2.2.1 Label 组件	39
1.1.2 通信的种类	2	2.2.2 Button 组件	40
1.2 串行通信	3	2.2.3 Timer 组件	41
1.2.1 RS-232 串行通信	3	2.2.4 PaintBox 组件	42
1.2.2 RS-485 串行通信	5	2.2.5 RadioButton 组件	43
1.2.3 USB 接口	7	2.2.6 GroupBox 组件	44
1.2.4 IEEE-1394	8	2.2.7 Shape 组件	44
1.3 串行通信端口	9	2.2.8 Edit 组件	45
1.3.1 信号定义	9	2.2.9 Memo 组件	46
1.3.2 ASCII 码对照表	10	2.3 程序简述	46
1.3.3 引脚意义及方向	11	2.3.1 工程的组成	46
1.3.4 通信参数	15	2.3.2 运算符	51
1.4 模式及流量	18	2.3.3 语法	53
1.4.1 工作模式	18	2.3.4 基本的信息对话框	55
1.4.2 硬件握手	19	2.3.5 字符串类型及其处理函数	57
1.4.3 软件握手	21	2.4 常见问题解答及本章习题	61
1.5 接线和错误预防	23	2.4.1 常见问题解答	61
1.5.1 接线方法	23	2.4.2 本章习题	62
1.5.2 错误预防	24	第 3 章 串行通信程序及组件建立	63
1.5.3 CRC 程序解析	25	3.1 串行通信的 Windows API 简述	63
1.6 常见问题解答及本章习题	26	3.1.1 串行通信相关函数	63
1.6.1 常见问题解答	26	3.1.2 使用流程	72
1.6.2 本章习题	27	3.1.3 检查资源设置	72
第 2 章 C++ Builder 及常用组件 简介	28	3.2 通信测试	74
2.1 窗口程序概念简述	28	3.2.1 通信步骤	74
2.1.1 对象的概念	28	3.2.2 回路测试	74
2.1.2 接口	29	3.2.3 串行端口的数字输出控制	81
2.1.3 以 C++ Builder 开发系统	31	3.2.4 串行端口的数字输入检测	89
2.1.4 C++ Builder 的环境模式	35	3.3 自动与事件	94
2.1.5 开发工程步骤	36	3.3.1 自动读取传入的字符串	95
		3.3.2 通信事件	101

3.3.3 创建多线程	110	5.3.2 本章习题	190
3.4 常见问题解答及本章习题	120	第 6 章 串行数据的处理	191
3.4.1 常见问题解答	120	6.1 命令字符串	191
3.4.2 本章习题	121	6.1.1 通信方式	191
第 4 章 串行通信组件的创建	122	6.1.2 校验和的使用	194
4.1 组件的基础	122	6.2 PSIS 仿真器简介	201
4.1.1 组件分类	122	6.2.1 PSIS 功能简介	201
4.1.2 组件组成	123	6.2.2 PSIS 上的接头定义	202
4.1.3 组件创建步骤	125	6.2.3 串行仪器控制的实习	203
4.2 通信组件的创建	126	6.2.4 通信参数的设置	204
4.2.1 组件分析	126	6.3 客户端的创建	205
4.2.2 属性创建	128	6.3.1 Memo 组件与数据显示	205
4.2.3 事件的创建	135	6.3.2 状态信号灯与数据显示	212
4.2.4 方法的考虑	139	6.3.3 PaintBox 与数据显示	219
4.2.5 集成组件	141	6.3.4 PaintBox 与数据显示—— 字节数据	230
4.3 安装及使用串行通信组件	154	6.4 常见问题解答及本章习题	237
4.3.1 TComm 的安装	154	6.4.1 常见问题解答	237
4.3.2 组件测试——数据的发送/ 接收	156	6.4.2 本章习题	237
4.3.3 组件测试——数字输入/ 输出	160	第 7 章 调制解调器的控制	238
4.3.4 组件事件测试——自动 读取数据	165	7.1 调制解调器	238
4.4 常见问题解答及本章习题	168	7.1.1 调制解调器及其种类	238
4.4.1 常见问题解答	168	7.1.2 指令集	241
4.4.2 本章习题	169	7.1.3 文件传输协议介绍	246
第 5 章 串行通信中的字符与字节	170	7.2 C++ Builder 与调制解调器	247
5.1 字符与字节	170	7.2.1 C++ Builder 与调制解调器 的通信	247
5.1.1 字符和字节的差别	170	7.2.2 与调制解调器连接	253
5.1.2 C++ Builder 中的字符串	171	7.2.3 调制解调器与专线	259
5.1.3 中英文字符串长度计算	171	7.2.4 专线模式设置与控制	260
5.1.4 字符编码	175	7.3 常见问题解答及本章习题	268
5.2 字节数据的发送与接收	179	7.3.1 常见问题解答	268
5.2.1 字节类型、声明与发送/ 接收	179	7.3.2 本章习题	268
5.2.2 动态数组	184	第 8 章 其他串行通信组件	269
5.3 常见问题解答及本章习题	189	8.1 Windows 的超级终端程序	269
5.3.1 常见问题解答	189	8.1.1 选择与使用超级终端	269
		8.1.2 与设备的联机测试	271

8.2 PComm Pro 软件.....	273	9.4.3 控制工程的创建.....	352
8.2.1 PComm Pro 的超级终端.....	274	9.5 温度记录仪.....	358
8.2.2 PComm Pro 的串行端口 性能测试.....	276	9.5.1 温度记录仪简介.....	359
8.2.3 PComm Pro 的数据监视器....	277	9.5.2 接口及命令格式.....	360
8.3 PComm Pro 通信链接库.....	280	9.5.3 通信工程的创建.....	363
8.3.1 函数库的使用及架构.....	280	9.6 转速计.....	372
8.3.2 数字输入/输出控制.....	282	9.6.1 转速计介绍.....	372
8.3.3 使用 DLL 进行串行数据的 发送与接收.....	294	9.6.2 接口及命令格式.....	373
8.3.4 事件的引发.....	300	9.6.3 工程程序的创建.....	374
8.4 常见问题解答及本章习题.....	308	9.7 条形码扫描器.....	380
8.4.1 常见问题解答.....	308	9.7.1 条形码扫描设备介绍.....	380
8.4.2 本章习题.....	308	9.7.2 接口及格式说明.....	381
第 9 章 仪器设备上的串行通信.....	309	9.7.3 工程程序的创建.....	382
9.1 噪音计.....	309	9.8 测量电表.....	385
9.1.1 仪器联机.....	309	9.8.1 Source Meter 设备介绍.....	385
9.1.2 噪音计简介.....	310	9.8.2 接口及命令格式.....	386
9.1.3 命令格式.....	311	9.8.3 工程程序的创建.....	388
9.1.4 通信工程的设计.....	312	9.9 常见问题解答及本章习题.....	395
9.1.5 噪音计读数的获取.....	318	9.9.1 常见问题解答.....	395
9.2 电功率计.....	323	9.9.2 本章习题.....	395
9.2.1 电功率计简介.....	323	第 10 章 串行通信的分布式监控.....	396
9.2.2 接口及命令格式讨论.....	324	10.1 分布式.....	396
9.2.3 测试工程的创建.....	327	10.1.1 何谓分布式监控.....	396
9.3 测量用放大器.....	338	10.1.2 多模块的网络系统.....	397
9.3.1 BK-2525 振动计简介.....	338	10.1.3 RS-232 与 RS-485 的转换.....	399
9.3.2 RS-232 接口及命令格式 说明.....	339	10.2 指令与格式.....	400
9.3.3 工程的创建.....	343	10.2.1 格式讨论.....	400
9.4 电源设备.....	348	10.2.2 发送/接收流程.....	403
9.4.1 电源设备简介.....	348	10.2.3 取得模块的配置.....	403
9.4.2 接口及命令格式.....	350	10.3 常见问题解答及本章习题.....	408
		10.3.1 常见问题解答.....	408
		10.3.2 本章习题.....	408
		附录 有关的补充资料.....	409

第 1 章 基本概念

万丈高楼平地起，学习任何一项技术总要对该技术的基础知识有一些了解，这样才能够知其然并知其所以然，也才能够真正掌握这项技术。本章将说明串行通信的一些基本概念。

1.1 通 信

不同的独立系统利用线路互相交换数据，便是通信；而构成整个通信的线路则称为网络。交换信息的系统若为计算机系统，则称为计算机网络通信。通信的目的是进行数据交换，由于数据必须经过交换才能由一端传送到另一端，发送端所使用的方法就是将数据利用一定的程序与线路发送出去，接收端则按照协商好的方式将数据收集起来并保存或显示在画面上。

1.1.1 数据发送

通信的主要目的是将数据从一端传送到另一端，实现数据的交换。从人与人之间的对话，计算机与设备间的数据交换，计算机与计算机间的数据传送，乃至广播或卫星信号的传送，都属于通信的范畴。

一个完整的通信系统包括发送端、接收端、转换数据的接口及传送数据的实际信道或媒体。一般说来，发送与接收的节点称为数据终端设备(Data Terminal Equipment, DTE)，如个人计算机、工作站或计算机系统。数据在到达正确目的地之前，可能需要经过一系列中间节点，这些中间节点负责转送数据，以将数据送达目的地，这些节点称为数据交换设备(Data Switching Equipment, DSE)，如电信局的交换机等。终端设备发送数据时，必须先将数据转换为电气信号，以便在线路上传输，而负责数据与电气信号转换的设备称为数据通信设备(Data Communication Equipment, DCE)，如调制解调器。因此，计算机等 DTE 设备进行通信时，须先经过 DCE 设备将数据转换为电气信号，才传送到电话线路；同样，信号由外界进入计算机时，也是先经过 DCE 设备将外界的电气信号进行转换，数据才会进入计算机。DTE 与 DCE 间的数据传输线路通常使用 RS-232，而 DCE 与 DSE 间的媒体则包括双绞线、同轴电缆、光纤或无线电等。

通过上述机制，设备与设备间、计算机与计算机间或是计算机与设备间便可以互相发送数据。又如图 1.1 所示，不仅可以通过线路的联系进行数据的传输，也可将两台设备进行直接连接，不需经过通信设备及交换设备的服务，就可进行数据的交换，这在传输距离较短时比较适用。

随着个人计算机的迅速发展，个人计算机成为信息传输的良好工具，数台计算机可以连接成一个速度很快的网络系统，互相交换数据。传输设备与媒体的快速发展，成本的大幅下降，也使得距离越来越不是问题；因此以计算机为基础发展出相当多的通信方法。现在以一台计算机遨游世界是多么简单。这都是因为科技的进步与通信协议的改进，提供了

一个渠道，让计算机可以通过调制解调器或其他的媒体与其他设备通信，也造就了如今高速信息流的时代。

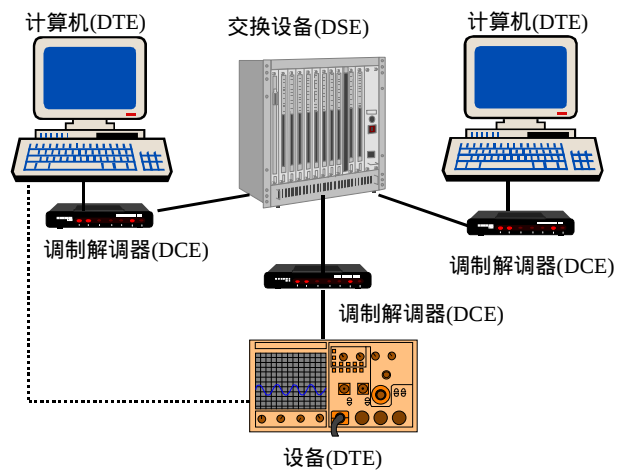


图 1.1 设备的连接

现在是一个自动化的时代，所见所闻的设备都可以与计算机相连接，而用来传输数据；通过计算机的作用，自动化在人们的生产生活中将会得到更加广泛的应用。

1.1.2 通信的种类

通常通信的类型可以区分为两种：并行通信(Parallel Communication)和串行通信(Serial Communication)。这两种不同的通信模式如图 1.2 所示。

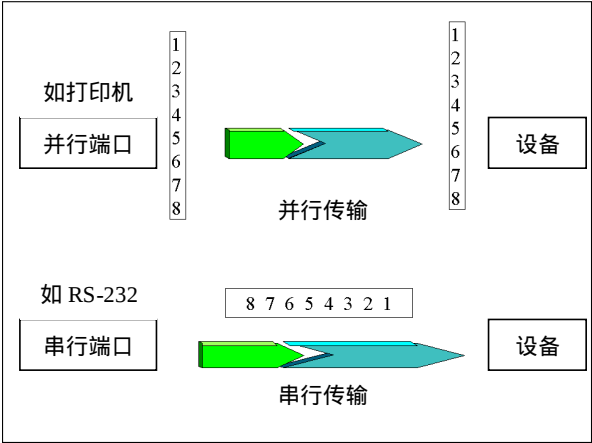


图 1.2 两种不同的传输方式

由图 1.2 中可以了解到，所谓的并行通信，即一次的传输量为 8 个位(1 个字节)；而串行通信则是一次只传输 1 个位(也就是一个标准电位的状态)。两者之间的数据传输速率相差了 7 倍之多，但并不是说串行通信就不好，它之所以存在这么长的时间，自然有它的长处。并行通信虽然可以在一次的数据传输中就传输 8 个位，但是因为数据电压传输的过程

中,容易因线路的因素而使标准电位发生变化(最常见的是电压衰减问题,以及信号间互相串音干扰(CrossTalk)),因而使传输的数据发生错误。如果传输线比较长,电压衰减效应及互相干扰问题会更加明显,数据的错误也就会比较容易发生。相比之下,串行通信一次只传 1 个位,处理的数据电压只有一个标准电位,因此比较不容易把数据漏失,再加上一些防范措施,要遗漏就更不容易了。

由最早的串行通信发展到现在,基于不同情况的需求,串行通信的样式也越来越多;而并行通信的发展其实也不曾间断过。在传输距离较短的应用场合中,使用并行通信的高传输率特性可以让数据的传输更快;在部分科学仪器、医疗诊断仪器由于需要传输的数据量都相当大,因此使用并行传输接口(如 GPIB、LPT 等)就见经常到。

1.2 串行通信

常用的串行通信有两种,本节将分别说明,并概括介绍近年来相当盛行的 USB 及 IEEE-1394,这两种也属于串行通信的扩展。

1.2.1 RS-232 串行通信

串行通信端口(Serial Communication Port)在系统控制的范畴中一直占据着极其重要的地位,它不仅没有因为时代的进步而遭淘汰,反而在规格上越来越完善,应用也越来越广泛。现在,串行通信端口(RS-232)是计算机上的标准配置,常用于连接调制解调器来传输数据。当然,它的重要性还不只如此。以下将对通信端口进行深入的介绍。

RS-232 的通信端口是每台计算机上的必要配备,通常含有 COM1 与 COM2 两个端口。一般的计算机将 COM1 以 9 引脚的接头接出,而以 25 引脚的接头将 COM2 接出。新一代的计算机均以 9 引脚的接头接出所有的 RS-232 通信端口。在计算机上的 RS-232 均是公头,即使是 25 引脚也是公头,千万不要与其他的设备弄混淆了(打印机连接端口也是 25 引脚,不过它是母头,请仔细分别)。通常与计算机连接的设备,最简单的通信接口就是 RS-232,不仅实际操作简单,而且价格上也便宜很多;在市面可见的数码相机、调制解调器等很多都提供 RS-232 作为与计算机通信的接口。

仔细检查计算机后面的接线部分,大大小小的接头一堆,有两个 9 引脚的接头区(以较新的计算机来说,以前的计算机可能有 25 引脚的情况),这就是串行通信端口。可以在计算机背后的各接头中看到如图 1.3 所示的接头,这就是 RS-232 的标准接头。

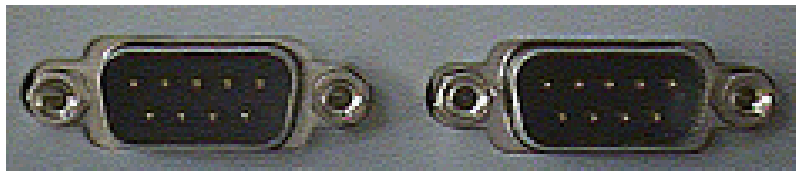


图 1.3 个人计算机上 RS-232 的外观

工业应用的场合中,工业计算机的使用比一般计算机普遍,在工业计算机上的接头位置很可能如图 1.4 所示。

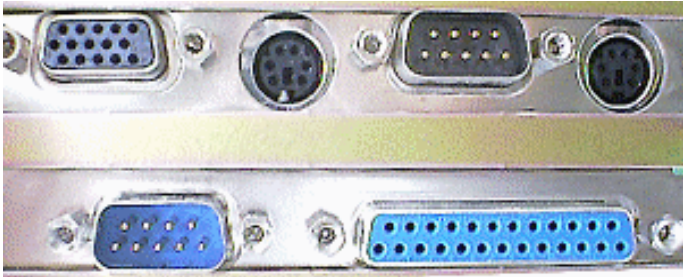


图 1.4 工业计算机中的 COM 端口及其他接头

其中 9 引脚的端口有两个，分别为上排的右数第二个以及下排的最左边一个。不管工业级还是一般的台式计算机，只要是 RS-232 就是公头 9 引脚。

计算机上的串行通信端口一定是公头，这点是不会改变的。即使所使用的计算机比较旧，上面还有 25 引脚的接头，肯定也是公头；这与打印机的母头接口是不一样的。如果从操作系统中进行查看，则可以选择【开始】 | 【设置】 | 【控制面板】命令，在【控制面板】中双击【系统】图标，在打开的【系统 属性】对话框中找到计算机的连接端口，其过程如图 1.5 所示。



图 1.5 查看计算机的连接端口

只要显示通信连接端口的名称，而且其左边没有黄色感叹号的话，就表示这个 COM

端口是可以使用的。

1.2.2 RS-485 串行通信

由于串行通信的简单易用，在工业领域大量使用串行通信作为数据交换的手段；可是工业环境通常会有噪声干扰传输线路，在用 RS-232 进行传输时经常会受到外界的电气干扰而使信号发生错误。为了解决上述问题，RS-485 的通信方式就应运而生了。

为了了解信号的干扰情况，首先来看如图 1.6 所示的 RS-232 的信号传输方式。

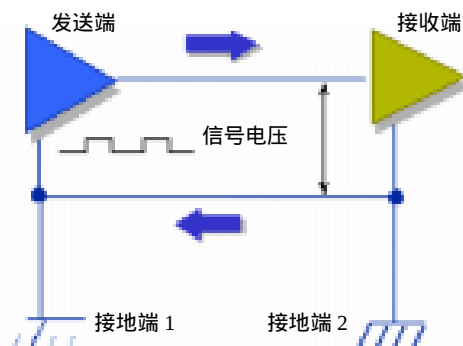


图 1.6 RS-232 的信号传输方式

由图 1.6 可知，RS-232 的信号标准电位乃是参考地线而来的。发送端参考接地端 1 来发送数据；接收端则参考接地端 2 来还原出发送端信号的标准电位；在两个接地端同电位的前提下，发送与接收端的信号标准电位是相同的。但若传输数据的过程中有噪声进到传输线路中，可能会如图 1.7 所示。

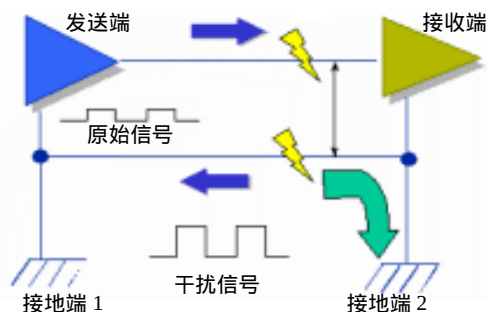


图 1.7 RS-232 信号受干扰时的情形

干扰信号在地线和信号上均会产生影响，原始信号在加上干扰信号后依然发送到接收端；而地线部分的信号则被地电压给抵消掉了，因此，信号便发生了扭曲，当然信号整个都不对了。

而 RS-485 的信号传输方式如图 1.8 所示。

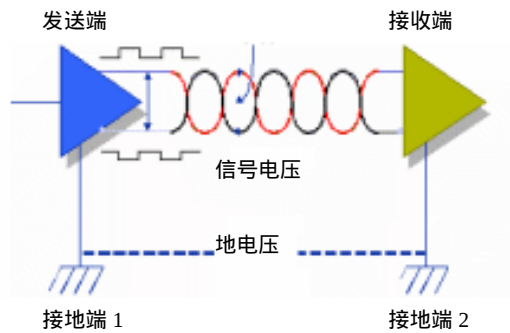


图 1.8 RS-485 的信号传输方式

RS-485 的信号在被发送出去之前会先分解成正负的两条线路，当到达接收端后，再将信号相减还原成原来的信号。如果将原始的信号标注为(DT)，而被分解后的信号分别标注为(D+)和(D-)，则原始信号与分解后的信号在由发送端发送出去时的运算关系如下：

$$(DT)=(D+) - (D-)$$

同样，接收端在接收到信号后，也按上式的关系将信号还原成原来的样子。而如果此线路受到干扰，其情况可能如图 1.9 所示。

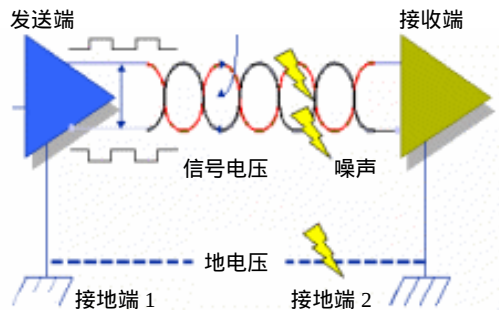


图 1.9 RS-485 传输受干扰时的情形

这时候在两条传输线上的信号会分别成为(D+) + Noise 和(D-) + Noise。如果接收端接收此信号，它必须按照一定的方式将其合成，合成的运算式如下：

$$(DT)=[(D+) + Noise] - [(D-) + Noise]=(D+) - (D-)$$

此式与前一式的结果是一样的。

所以，使用 RS-485 网络可以有效地防止噪声干扰。也正因为这种特性，工业上比较适合使用这种串行传输方式。

 **注意：** RS-485 虽可防止噪声，但也并非无懈可击，若噪声信号的电压很大(如闪电)，一样会造成错误。

1.2.3 USB 接口

由于时代的进步，串行通信也朝向高速化发展。近年来个人计算机的快速发展下，使用串行通信发展出了 USB(Universal Serial Bus，通用串行总线)，此种规格的目的在于集成一般计算机所使用的外围设备的连接方式，而且其所采用的信号传输方式也是串行通信(一次只发送一个位)。

一般的计算机，其与外围设备的联机方式如图 1.10 所示。

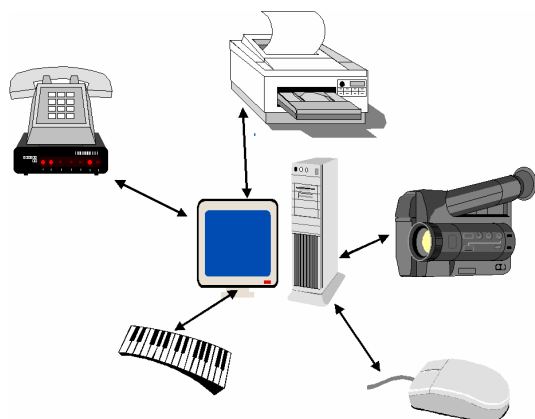


图 1.10 传统设备与计算机间的连接

每一种设备都使用一条专用的连接线与计算机后面的接头相连接，而计算机的背后则都为不同的设备事先安装了可以接到主板上的接头。由这种连接方法可以发现，计算机的背后一定是一大堆难以分辨的线路。每一次更换一个设备都得将计算机的背后给翻过来，辛苦地整理完线路，并安装好新设备后，再将计算机位置还原。这绝对不是一件经常可以做的事情，因为光是整理线路就会浪费掉不少的时间。

而如果可以采用 USB 接口的连接方式，则这些设备的接线方式将改为如图 1.11 所示的连接方式。

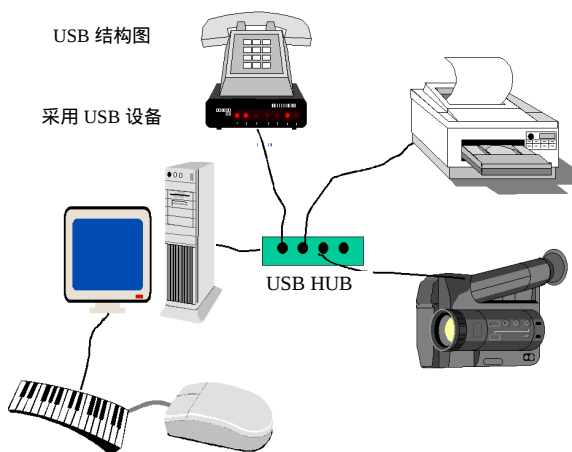


图 1.11 USB 设备和计算机间的连接

相同的外围设备，若采用 USB 接口，则连接时只需用串接的方式将各个设备连接在一起即可；而如果设备比较多，也可以采用 USB 集线器来进行集中管理(与网络的集线器管理方式是一样的道理)，设备也可以有 Windows 操作系统支持的“即插即用”功能，各个设备与集线器间的最大距离为 5 米，最多可连接为 127 个外围设备。

另一方面，使用串行通信时，速度也是一个考虑的重点。USB 集线器可以将使用 USB 接口的各个计算机外围设备连接起来，设备一多，如果众多的设备均在同时传输数据，速度必定会成为 USB 线路的一个负担。不过 USB 采用的速度高达 12 Mbps(每秒钟传输 1 200 万个位)，每个单一设备最高传输带宽高达 6 Mbps，比起一般 RS-232 的最快 115 Kbps，USB 在传输上是比较快的。其主要的特性整理如表 1.1 所示。

表 1.1 USB 特性表

特 性	描 述
低价位	USB 提供低价位的连接外围装置方案
热连接	USB 自动检测装置连接，软件会自动将其规划，以供立刻使用。此过程中，不需要用户介入
单一的连接头形式	使用单一种类的连接头连接所有的 USB 装置。可利用 USB 集线器增加额外的连接头
连接数多	最高可连接 127 个低速或高速的装置
在线供电	连接的缆线可提供 5 V 的直流电源，而不同的集线器可提供的电流范围在 100 mA ~ 500 mA 之间
不占系统资源	USB 装置不占系统的地址空间及 IRQ 线
错误检测与复原	USB 数据交换时包含了错误的检测；当错误发生时，该笔数据将被重传
节省能源	若是经过了 3 毫秒总线仍无动作，USB 装置将会自动进入省电状态；此时装置所消耗的电流不超过 500 μ A
支持 4 种传输	USB 定义了巨量、实时、中断及控制等 4 种传输模式，可依不同的情况启用不同的传输模式

USB 一经发表，在最初的两年，没有引起很大的反响，各种支持 USB 接口的设备没有流行起来。其原因之一是当时的 Windows 操作系统不支持 USB 接口。Windows 98 以后的操作系统支持 USB 接口，使用 USB 产品在安装上更容易，例如鼠标、键盘、电话、调制解调器、打印机、麦克风等，只要支持 USB 均可得到操作系统的支持，使安装与使用都非常方便。现在，已经有非常多的设备都使用 USB 作为和计算机间连接的接口了。

在 1999 年所发表的 USB 2.0 已经将其速度提升到 480 Mbps，这将使更多的设备可以使用 USB 作为数据传输的接口。

1.2.4 IEEE-1394

IEEE-1394(也称为 FireWire，火线)与 USB 同样拥有即插即用的功能，都是为了解决计算机与外围设备复杂的连接问题，同样具有容易使用与便利等优点，并且也是使用串行通信的传输方式。但不一样的是，IEEE-1394 在高速传输上占尽了优势，现在已达到的传输速度是 400 Mbps(是 USB 1.0 版的 33 倍以上)，而且速度还向 1 000 Mbps 迈进。另外 IEEE-

1394 最多可串接 63 个外围设备，还可提供异步(Asynchronous)或同步(Synchronous)两种传输模式。因此在不久的将来，人们将会发现：硬盘、打印机、扫描仪、VCD、CD-ROM、数码相机、调制解调器、音响系统、家庭电影院、电缆调制解调器(Cable Modem)、电视等，都可以使用 IEEE-1394 连接起来。而除了可进行信息技术、消费性电子产品的影像及声音传输之外，未来更可以应用在卫星传输、通信数据、工厂自动化、太空探测等用途上。IEEE-1394 应用领域之广泛，将使得家庭数字化及网络化的理想更易实现。

既然 IEEE-1394 这么强势，USB 是否就注定了没有存在的价值了呢？其实不然，两者应该是合作的关系，USB 主攻中低速的产品；而 IEEE-1394 则主攻高速传输的场合，两者的结合才能使得串行传输的功能得到最佳的支持。表 1.2 是 USB 与 IEEE-1394 这两种高速传输接口的比较。

表 1.2 USB 和 IEEE-1394 的比较

	USB	IEEE-1394
应用	低速装备	高速装备
带宽(Mbps)	1.5, 12	100, 200, 400
电缆长度	5 米	4.5 米
电缆	四线	六线
即插即用	支持	支持

1.3 串行通信端口

本节概括介绍串行通信端口的相关术语，包括引脚定义、信号、参数等。

1.3.1 信号定义

计算机并无法分辨所谓的数字或文字，在其内部所有的数据都是 0 和 1，任何数据都是这两个数字的组合。

所有电子仪器的基础均来自“开”、“关”两个状态的改变，可以将它们表示为 0、1 或是“高电位”、“低电位”，不管表示方法如何，目的在于造成“状态的改变”。将很多的 0 与 1 组合成一大串的数字序列后，就可以定义它们所代表的意义了。

既然高低电位可以形成 0、1，因此 0、1 的无数组合可以有无穷的意义。但是何谓高电位？何谓低电位呢？任何电子零件或芯片，其高低电位的定义均不相同，但是相同性质的电子零件会有一定的规定，大家遵守此规定，这样制造出来的芯片的高低电位就会一致了。当然，RS-232 也有相关的电气标准电位定义，是大家共同遵循的。

ITU(国际电信联盟)的咨询机构 CCITT(国际电报电话咨询委员会)公布了 RS-232C 标准，之后由 EIA(电子工业联合会)制定；很多国家都以此为模板，而日本也以此制定了 JIS X5101“数据回路终端装置和数据终端装置的接口”。在这个标准里，0、1 的规定如图 1.12 所示。

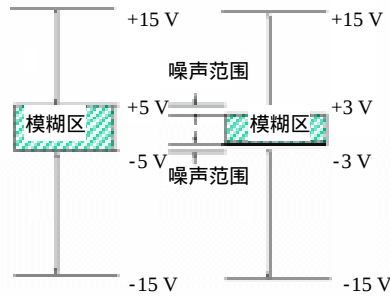


图 1.12 RS-232 电压标准电位图

原定义是+5 V~+15 V 之间为 0，-5 V~-15 V 之间为 1，考虑到噪声的干扰后，可允许的范围成为+3 V~+15 V 之间为 0，-3 V~-15 V 之间为 1。如果电压标准电位落在“模糊区”，这部分的电压将使计算机无法判断，可能得到 0，但也有可能得到 1，这样得到的结果是不可信的，因此所有信号必须明确地落于规定的范围内才行。

在 RS-232 的标准中，电压在+3 V~+15 V(一般使用+6 V)之间称为 0 或 Space，一般用途是作为 On。电压在-3 V~-15 V(一般使用-6 V)之间称为 1 或 Mark，一般用途是作为 Off。有时候以 High、Low 更能表现出其实际状态。

计算机内部的设计也是一样的，都是利用“高电位”与“低电位”的状态改变而组合成一串的数据。因此，就 RS-232 来说，信号要被传输，一定要定义所谓的 0 或 1 的状态，个人计算机上的信号采集卡等设备，其低电位是 0.7 V 以下，而高电位则是 2.1 V 以上；根据实验结果，计算机上的 RS-232 的高电位约 9 V，而低电位约为-9 V，如此正负 9 伏的改变，被记录成 0 与 1 的状态。因此，RS-232 串行通信以正负 9 V 代表 0 与 1 的状态。这个范围是符合规定的。

了解了高低电位的意义后，传输过程就是在传输在线不断地产生高低电位的变化，发送端造成传输线的电压变化，而接收端就是解读此高低电位变化而还原出原始的信息。

 **注意：** 并没有一个真正的绝对电压值称为某某电压！电压值是一种比较值的。例如 A 点是 5 V，B 点是 3 V，那么由 B 点来看 A 点，就称 A 点是+2 V；相反的，由 A 点来看 B 点就称 B 点的电压是-2 V。也由于都是找一个参考点作为电压值的读值来源，因此设备上都有一个接地点，所有的接线电压值都以这个点作为参考；想得到电压值的话，那就把参考点接在这个接地点就可以了。

1.3.2 ASCII 码对照表

由于两台计算机之间的信息传递是在一连串的高低电位之间进行，每一个标准电位可以当成一个状态，这个状态可能是 0 或 1，计算机里面将每一个 0 或 1 称为一个位，而 8 个位则组成一个字节。将 8 个位合起来后，共有 $2^8=256$ 种数值，其数值为 0~255。而在计算机中便有一个 ASCII 码对照表，用这 256 种组合分别代表 256 个字符或句柄，而通信就是交换这 256 个字符或句柄。

在键盘上大部分按键字符是可见的，这些字符集中在 ASCII 码的前 128 个字符内；而