

Visual C++

串口通信开发入门 与编程实践



周韧研 商 斌 编著
飞思科技产品研发中心 监制

C/C++
开发专家

与初学者探讨Visual C++串口通信开发技术，每一步都有详尽说明
案例式教学，丰富的项目实例+详尽的代码+完整的注解
倡导轻松、快捷编程的程序员必备手册
Visual C++串口通信开发入门、进阶的“一指神功”

本书实例源代码请到
<http://www.fecit.com.cn>的“下载专区”中下载



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

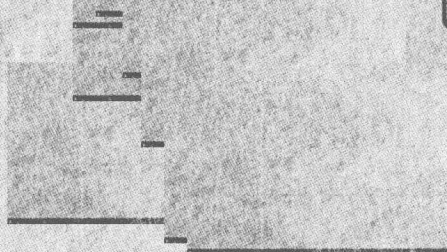
Visual C++

串口通信开发入门 与编程实践



周韧研 商 斌 编著
飞思科技产品研发中心 监制

C/C++
开发专家



电子工业出版社

Publishing House of Electronics Industry

北京·100013

内容简介

本书着重介绍计算机串口通信的硬件原理、软件开发和工程实例。

本书从通信原理和工程实现出发,结合示例讲述使用串口进行通信的原理和开发技术。全书共分12章,第1、2章从电路和通信电子设备角度出发,详细讲解串行通信的理论基础和物理实现;第3~6章介绍在通用计算机(PC)的Windows环境下如何进行串口通信软件的开发,包括使用Windows API、C++串口类以及Qt串口类等;第7~12章以工程中的6个实例为蓝本,介绍Windows平台下串行接口开发与嵌入式设备串口通信开发的相关技术。

书中源代码和项目设计代码请到 www.fecit.com.cn 的“下载专区”中下载。

本书的编写者具有计算机接口设备开发及软件开发的大量经验。本书适合于对电子电路和计算机体系有初步了解,但对软件开发掌握不多的读者作为自学用书,也可以作为自动化、电气、工控、机械等领域有相关需求的工程人员的参考书。

未经许可,不得以任何方式复制或抄袭本书的部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual C++串口通信开发入门与编程实践 / 周韧研, 商斌编著. —北京: 电子工业出版社, 2009.4

(C/C++开发专家)

ISBN 978-7-121-08269-6

I. V… II. ①周…②商… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2009)第017570号

责任编辑: 王树伟 李利健

印刷: 北京智力达印刷有限公司

装订: 北京中新伟业印刷有限公司

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: 787×1092 1/16 印张: 31.25 字数: 800千字

印次: 2009年4月第1次印刷

印数: 4000册 定价: 55.00元

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。

出版说明

“聪明的人使用 Delphi，真正的程序员使用 C++。”

时至今日，这句曾经在业内广为流行的话语又增添了更丰富的内涵。

脚本语言、Java、.NET 等正在争夺更大的天地。

然而，C/C++ 仍不失为最好、最纯粹的编程语言。

——“C/C++ 开发专家”引导你成为真正的程序员

C/C++ 的发展

作为一种结构化的中高级编程语言，C 语言具有功能齐全、适用范围广的优势，一直为很多程序员所钟爱，并被视为最佳的编程入门语言，拥有着庞大的使用和学习人群。C++ 是在 C 语言基础上开发的一种集面向对象编程、通用编程和传统的过程化编程于一体的编程语言，是目前业界广泛使用的一种编程语言。然而，软件产业的规模和环境发展到今天，已经发生了深刻的变化。如今企业级应用整合与开发的任务主要由 Java、基于 .NET 平台的 C# 及各种新型动态语言来承担。C++ 的应用场合有所收缩，不再像之前那样从上到下包打天下，呈现出鲜明的行业应用特色。未来 C++ 主要在系统级复杂应用程序，高性能、实时中间件和嵌入式领域发挥所长。随着多核 CPU 的普及和网络安全重要性的空前提升，在并发程序设计、系统安全及视频处理、嵌入式开发方面，C++ 将获得新的应用空间。在大规模、高性能计算，游戏开发、嵌入式实时应用开发方面，以及一些传统的客户端软件和构件开发中，C++ 也将继续保持其稳定的地位。

C/C++ 的图书现状

C++ 的教学和使用具有其复杂性，而传统图书和学习方法的各种弊端更加剧了这一现象，使 C++ 成为不少人望而生畏的难学、难用的“专家语言”。虽然国内的 C/C++ 图书并不缺乏，但大多只适合有一定经验的程序员提升功力之用，而内容全面准确、讲解循序渐进、学习简明易懂的原创图书并不多见。近期 C/C++ 图书市场存在如下特点：

1. 国外经典图书全面翻新。近年来国外一些书商根据 C++ 所发生的变化，不断地进行版本升级或全面改写书稿，推出新的力作。
2. 国内原创图书缺乏力作。近年来国内虽然有一批令人耳目一新的 C++ 好书面世，但在技术层面上对实践的关注略显不足，难解读读者之渴。
3. 关键性图书存在空白。基于组件的软件开发、复杂网络应用，以及热度尚在的 COM 开发等方面的图书有待开发。

基于上述现状，我们组织 C/C++ 各应用领域的作者，推出本丛书“C/C++ 开发专家”，力求从新的、实用的、全面的角度介绍 C/C++，使其紧密地跟踪当前国内最实用、最热门的编程技术。我们期望通过这套丛书，能够提高各位读者的 C/C++ 开发水平及编程的实践能力，为我国计算机产业奉献一份微薄之力。

“C/C++ 开发专家”助你成为真正的程序员

“C/C++ 开发专家”的读者定位是：C/C++ 初学者，需要提升应用开发能力的程序员，具有实际开发经验的中高级程序员。对阅读本丛书的读者建议如下：

➤ 面向 C/C++ 初学者

本丛书通俗易懂，并自成体系。丛书全面介绍 C/C++ 及 Visual C++ 的编程技术和实践操作。通过学习，初学者可快速地掌握涉及 OOP、STL、泛型编程等标准 C/C++ 的内容，对 C/C++ 技术应用有更深刻的理解。

➤ 面向需要提升应用开发能力的程序员

对于那些急需提升应用开发能力的程序员来说，本丛书是再好不过的专家向导。丛书除全面介绍标准 C/C++ 的内容外，还涉及数字图像处理、流媒体、网络通信和嵌入式开发等多个领域，可以为从事相关领域开发的程序员提供有益的帮助和参考。

➤ 面向具有实际开发经验的中高级程序员

本书同样适合于具有实际开发经验的中高级程序员。书中列举的大部分实例具体翔实，非常值得广大高级程序员学习和借鉴。

“C/C++开发专家”为程序员量身打造

本套丛书通过不同种类的图书来满足读者的需求。

➤ 语言入门

C/C++是一门优秀的高级语言。它绝不像一些传统图书所述是一门晦涩难懂、高深莫测的“专家语言”。本丛书的语言入门分支面向初学者，以通俗易懂的语言，介绍标准的C/C++语言知识，以及Visual C++编程技术；在保证知识体系的完整性的同时，在语言、体例上更贴近程序员的学习心理需求。

➤ 应用实践

如果脱离了具体的应用背景，任何一门计算机语言的学习都是“纸上谈兵”。如果程序员没有真正掌握面向应用的实践开发技能，那么很有可能面临来自就业的压力。本丛书的应用实践分支面向数字图像处理、流媒体、网络通信、嵌入式开发等不同的行业应用方向，介绍C/C++应用技术。目标是努力将读者培养成具有实际开发能力的从业人员。

➤ 开发详解

只让人阅读一遍的书很难说是一本好书。任何一本书在读者的眼中总会经历“厚→薄→厚”的过程。同样，C/C++语言会耐人寻味，但真正理解C/C++一般性内容需要花时间，而要做到融会贯通则更要下工夫。本丛书的开发详解分支针对C/C++语言及Visual C++中的高级特性，进行深入的剖析和讲解。C/C++程序员一旦掌握更高级的编程技巧，且对C/C++的语言内涵及开发技术有更为深入的理解，就能得心应手地运用这门语言。

➤ 技巧集锦

从大规模的并行计算到嵌入式系统开发，C/C++的应用领域非常广泛。即便是世界上最厚的一本书，也无法介绍所有的C/C++技术。针对这一特点，本丛书的技巧集锦分支对程序员经常遇到的问题进行解答和分析，并注重举一反三，启发读者思考。通过对这一话题的讨论，给正在从事或即将从事C/C++开发的程序员以最大的启迪。

“C/C++开发专家”丛书特色

本书具有如下特色。

➤ 由浅入深，通俗易懂

实际上，根本就不存在只面向纯粹的初学者的C/C++书籍。原因很简单：C/C++就不是初级的语言。初学者选择C/C++的时候，除了有足够的兴趣之外，还要有足够的耐心和恒心。为此，本书在保持完整性的同时注重语言的通俗性和知识的趣味性，避免了较为复杂的理论概念，取而代之的是常见的编程技巧和实际例子，力求由浅入深，通俗易懂，充分调动读者的阅读兴趣。

➤ 案例为主，内容生动

如果没有“案例”，C/C++的学习可能非常枯燥无趣；如果没有合适、有趣的“案例”，C/C++的学习仍会枯燥无趣。与以往的风格不同，本书强调编程实践，提供了大量的实例及源代码。这些案例均由作者从实际开发工作中设计的原型案例精简加工而成，形式丰富多样，具有很好的实用价值。

➤ 倡导正确的编程思想

“授之以鱼，不如授之以渔。”本书并非按部就班地完成知识传授，而是在介绍知识的同时倡导正确的学习思想和方法。如：倡导OOP思想、泛型编程、流行的设计模式、不断的重构理念和开源精神等。读者在阅读本书的同时，会接触到这些新的理念和方法。在某些开放性话题上，本书一反以往一些图书的“专家”面孔，更加贴近读者，从各个角度与读者展开交流和探讨。

飞思科技产品研发中心

计算机的体系接口是计算机体系中的重要组成部分,体系接口的发展也是计算机技术发展的一个重要标志。计算机体系接口复杂多样,在它的发展过程中,总有一些接口在慢慢消失,又有一些接口在不断出现。在当代,主流的体系接口有 PCI/PCI-E 接口、USB 1.0/2.0、千兆以太网等,新兴接口还有蓝牙、USB 3.0、吉位以太网、光纤等。这些体系接口都能够提供通用计算机与包括嵌入式系统在内的非计算机电子设备之间进行通信的功能。

无论是主流的体系接口还是新兴的体系接口,其传输速度和传输可靠性都使得传统体系接口(并口、串口)不能望其项背。但与此同时,这些接口的复杂性也是传统体系接口所不能比拟的。这体现在两个方面:一个是下位设备,也就是与通用计算机通信的设备的复杂性。通常,为了支持诸如以太网或者 USB 的接口,要么需要专业的硬件或者嵌入式软件开发者投入大量的精力进行开发、验证和维护,要么花费资金购买相应的硬件授权(IP 核)或者接口芯片,硬件(嵌入式软件)的复杂性必然带来各种硬件开销(硬件量、功耗等)的增加。另一个是通用计算机上软件的复杂性。对支持以太网或者 USB 通信的软件开发,由于体系本身的复杂性,程序员需要拥有相当的知识储备和过硬的调试能力。

考察在科研和生产中所使用的电子电气设备和与通用计算机的交互状态,我们发现,有相当一部分设备并不需要主流或者新兴计算机体系接口所提供的大吞吐量,而是往往对通信的实时性有特别的要求。显然,对于这样的情况,使用主流或者新兴计算机体系接口就不合适。嵌入式设备,特别是嵌入式计算机系统,都希望能够通过一种简单、可靠、高实时性的接口与通用计算机通信,而且,在通用计算机上运行的软件也要易于开发和调试。作者认为,正是由于这个需求的存在,是通用计算机上的异步串行通信接口(UART)能够保留到现在的原因。

针对通用计算机上的异步串行通信接口开发的特点,结合目前较为新颖的开发思路,我们组织编写了本书。归结起来,本书具有如下特色:

- 理论和实践的结合。针对目前接口开发教程对通信理论、硬件技术介绍不足的缺点,本书着重介绍了一般的通信理论和软硬件相连接部分的技术细节。例如,当我们回答为什么要在软件层面也实现某种协议的时候,在理论和硬件实现的原理上就能够找到答案。同样,在软件中启动一个操作,在硬件层面上会发生什么变化,本书也有详细的说明。
- 软件技术的新颖。在本书中,我们抛弃了 Windows 下串口通信中经常使用,但又被不恰当使用的 MSCOMM 控件,而是从基本的 Windows API 函数讲起,然后介绍它的类封装。在图形界面开发的部分,我们不仅介绍主流的 MFC 技术,而且对目前非常流行的 Qt 库中如何使用串口通信进行了详细说明。
- 工程实例的新颖。在工程实例介绍的环节,我们介绍了 DSP 系统、串口到以太网通信等目前嵌入式领域发展迅猛的技术,供有相似工程需求的读者进行参考。

全书共分 12 章，分别介绍如下：

第 1 章——串行通信的基本概念。介绍与串口通信密切相关的理论和技术概念，首先从通信问题的数学模型切入，在此基础上，解释 EIA RS-232 的基本原则、规范，以及高层次协议设计的必要性。

第 2 章——异步串行通信接口电路简介。介绍在通用计算机和嵌入式设备中兼容串口电路的实现，以及软硬件之间的体系分界。

第 3 章——在 Windows NT 中搭建开发环境。本章中介绍平台和开发工具，首先说明进行串口程序开发所需要的硬件和软件环境，然后结合第一个串口程序 Hello World 介绍 Visual C++ 的使用。我们把重点放在开发流程上，因为有一个良好的流程是非常重要的。

第 4 章——使用 Windows API 串口编程。Windows 以 SDK 串口通信函数的方式提供了应用程序对设备的操作接口。Windows API 是串口通信开发的基本。本章将结合示例详细介绍与串口通信相关的 Windows API 的使用方法。

第 5 章——使用 CSerial 类。使用非面向对象的 SDK 编程不利于程序的维护和代码复用。本章将介绍一种基于 C++ 的通信 API 封装——CSerial 类。CSerial 类适用于文本程序界面和图形程序界面，代码量少，使用方便。我们以 CSerial 类为基础，介绍它的使用方法和在图形界面程序设计中的应用。

第 6 章——使用 Qt 进行串口编程。当今，跨平台的快速应用开发已经成为主流。Qt 是一款高性能、跨平台的 C++ 应用程序开发框架，包含一个类库和一系列的工具。在官方的 Qt 库中没有提供串口编程的类，本章以开源项目 QextSerialPort 为例，介绍如何使用 Qt 开发串口应用程序。

第 7 章——Windows 下双机点到点串行通信系统设计与开发。Windows 下双机的串行通信系统是一个典型的通信系统，它是我们为了实现计算机底层的工作，以及为了用户更好地和系统能够直接相连而提出来的，它不但可以广泛地应用于各个领域中，还可以在比较艰苦或者不方便的情况下如没有网线的时候进行双机通信。

第 8 章——16 位高速 DSP 增强型同步串口的设计。本章主要针对 DSP 芯片外设电路中的增强型同步串口及其他一些外设模块开展工作。外设电路对于提高整个芯片的性能起着非常重要的作用，它是内核和片外电路的接口，负责外部电路的数据交换，外设电路性能的好坏直接影响整个芯片的工作。

第 9 章——串口与以太网数据传输实现。串口和以太网口数据转换模块可以应用在串口设备需要远程传输文件的场合，如数控机床控制文件的远程传输等。本章首先介绍该类模块在国内外的现状；然后经过分析比较，选取了 Rabbit 公司的 RCM2200 模块来实现以太网口和串口的数据传输；接下来介绍 RCM2200 微控制器核心模块以及 Dynamic C 软件开发环境，在此基础上选择并实现了串口和以太网传输协议，即 XMODEM 和 TFTP 协议；最后介绍实现文件传输的整体方案，通过计算机的实验演示和数控机床的实际应用，都达到了比较好的效果。

第 10 章——基于串口的 DNC 信息采集系统的开发。数据采集是分布式数字控制 (DNC)

系统的一个重要功能。在自动化制造中,无论是信息检测、测试与监控、物流和设备管理,还是设备诊断与维修,均是以数据采集技术作为支撑的。可以说,先进合理的数据采集技术是实现 DNC、MES、MRPII、ERP 和制造自动化的重要基础。DNC 信息采集系统作为 DNC 的子系统,是 DNC 获得底层信息的主要来源,在 DNC 系统中占有重要地位,是 DNC 的核心内容之一。

第 11 章——Windows XP 下 USB 转 RS-232 桥接器驱动程序开发。本章利用 Windows XP

DDK、Visual C++6.0、DbgView、SoftICE 等开发和调试工具开发出了以 Windows XP 为平台的桥接器驱动程序,创建了和真实串口功能基本相同的虚拟串口,为桥接器的使用提供了软件保证。首先对桥接器硬件设计进行了分析,对 WDM (Windows Driver Mode) 驱动模式进行了说明;然后设计了桥接器驱动模型,确立了开发方案,明确并设计了所需的各种机制,设计了重点例程的实现策略,用驱动程序中的典型用例来说明具体实现;最后对驱动测试、安装过程进行了说明,对整个设计和应用做了总结,并提出了进一步的完善思路。

第 12 章——串口通信在机器人实时控制中的应用开发。主要以 MOTOMAN UP6 机器人

为载体,介绍 Visual C++环境下机器人软件控制系统的开发过程。本项目主要是通过 PC 来实现机器人各种运动的控制。PC 和下位机之间通过一条 RS-232C 串口线进行串口通信。本项目采用了多线程编程技术,并将各种不同的运动模块封装起来,使其具有可移植性,便于以后开发工作的进一步深入。本控制系统界面友好,在测试过程中实现了对 UP6 机器人的运动控制,并且能够对机器人的动作进行实时监控与图像记录,还能够对机器人进行单步控制,实现了对 UP6 功能的扩展。

全书讲解由浅入深、通俗易懂、注重实践,是一本不可多得的程序员手册。本书实例源代码和项目设计代码请到 www.fecit.com.cn 的“下载专区”中下载。

本书适合于对电子电路和计算机体系有初步了解,但对软件开发掌握不多的读者作为自学用书,也可作为自动化、电气、工控、机械等领域有相关需求的工程人员的参考书。

由于作者水平有限,书中疏漏和不当之处在所难免,欢迎读者指正。联系信箱: DevEmb@gmail.com, 详情请垂询 <http://books.vcer.net/vcserial>。

编 著 者

联系方式

咨询电话: (010) 68134545 88254160

电子邮件: support@fecit.com.cn

服务网址: <http://www.fecit.com.cn> <http://www.fecit.net>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

目 录

第 1 章 串行通信的基本概念	1
1.1 从电路到通信系统	2
1.1.1 应用信息论简要	2
1.1.2 串行通信的信源特性	8
1.1.3 串行通信的信道特性	9
1.2 计算机的数据接口	12
1.2.1 使用数据接口	13
1.2.2 计算机数据接口的发展	14
1.2.3 LPC 总线与串口适配器	17
1.3 串行通信协议	19
1.3.1 为什么制定协议	19
1.3.2 OSI 协议参考模型	20
1.3.3 一般模型与协议栈	20
1.3.4 串行通信协议	23
1.4 EIA RS-232 串行接口标准介绍	28
1.4.1 EIA RS-232 标准概述	28
1.4.2 电气特性	30
1.4.3 机械特性	32
1.4.4 信号线定义	35
1.4.5 串口近距离通信	39
1.4.6 串口通信的流控制	41
1.5 RS-422 与 RS-485 串行接口标准	45
1.5.1 RS-422 电气规定	46
1.5.2 RS-485 电气规定	47
1.5.3 RS-422 与 RS-485 的网络安装注意要点	48
1.5.4 RS-422 与 RS-485 传输线上匹配的一些说明	49
1.5.5 RS-422 与 RS-485 的接地问题	50
1.5.6 RS-422 与 RS-485 的网络失效保护	51
1.5.7 RS-422 与 RS-485 的瞬态保护	52
1.6 本章小结	53
第 2 章 异步串行通信接口电路简介	55
2.1 8250 兼容接口电路	56
2.1.1 8250 兼容接口电路概述	56
2.1.2 8250 的结构	57
2.1.3 8250 的编程方法	58
2.2 8251A 接口电路	76
2.2.1 8251A 的外部特性	76
2.2.2 8251A 的内部结构和编程模型	77
2.2.3 8251A 的状态字	79
2.2.4 8251A 的方式命令和工作命令的使用	79
2.2.5 8251A 应用举例	79
2.3 本章小结	81
第 3 章 在 Windows NT 中搭建开发环境	83
3.1 准备工作	84
3.1.1 使用串口调试助手	84
3.1.2 双端口互联方案	84
3.1.3 单端口自联方案	85
3.1.4 使用 USB-UART 转换器	85
3.1.5 使用虚拟串口	86
3.2 使用 Microsoft Visual C++	88
3.2.1 开发平台的选择	89
3.2.2 工程类型和开发流程	89
3.2.3 Hello World——第一个串口通信程序	90
3.3 本章小结	95
第 4 章 使用 Windows API 串口编程	97
4.1 Windows API 串口编程概述	98
4.1.1 不使用 Windows API	98
4.1.2 Windows API 初探	100
4.1.3 使用 Windows API 进行串口开发	102
4.2 同步和异步 I/O——基本的读写问题	104
4.2.1 CreateFile 函数——开启串口	104
4.2.2 CreateEvent 函数——创建事件	105
4.2.3 Overlapped 结构——异步模式信息的表达	106
4.2.4 WriteFile 函数——发送数据	107
4.2.5 ReadFile 函数——接收数据	107

4.2.6	WaitForSingleObject——等待事件信号	108
4.2.7	一个同步和异步 I/O 例子	108
4.3	Windows 通信 API	114
4.3.1	DCB 概述	114
4.3.2	流控制	117
4.3.3	传输超时	118
4.3.4	串口状态	121
4.4	本章小结	128
第 5 章	使用 CSerial 类	129
5.1	封装串口通信 API	130
5.1.1	串口编程回顾	130
5.1.2	封装串口通信相关的 API	131
5.1.3	封装方案	133
5.2	CSerial 简介	134
5.2.1	概述	134
5.2.2	修改“Hello World”程序	134
5.3	CSerial 的串口事件	143
5.3.1	监听器: 串口事件的响应	143
5.3.2	异步串口事件	149
5.4	Windows GUI 编程初探	154
5.4.1	函数指针与 Windows GUI	154
5.4.2	使用 SDK 开发 Windows GUI 程序	156
5.4.3	资源、对话框和对话框模板	162
5.4.4	使用 MFC	169
5.5	使用 CSerial 和 MFC 编写串口通信程序	174
5.5.1	向框架添加 CSerial	174
5.5.2	设计应用程序界面	176
5.5.3	串口配置	178
5.5.4	数据的发送和接收	184
5.5.5	有关 CSerialMFC 的特别说明	187
5.6	本章小结	187
第 6 章	使用 Qt 进行串口编程	189
6.1	Qt 简介	190
6.1.1	Qt 的组成和特点	190
6.1.2	Qt 的安装	190
6.1.3	Qt 应用程序开发流程	194
6.1.4	获取帮助	213
6.2	使用 Qt 开发串口通信程序	213
6.2.1	Qt 中的多线程编程	213

6.2.2	使用 QextSerialPort——简单的实验	221
6.2.3	完善的串口通信程序	228
6.2.4	若干问题	240
6.3	本章小结	241
第 7 章	Windows 下双机点到点串行通信系统设计与开发	243
7.1	Windows 下双机的串行通信系统简介	244
7.2	Windows 下双机点到点的串行通信系统的用户需求	244
7.3	Windows 下双机的串行通信系统的分析	244
7.4	利用 UML 为本工程实例建模	245
7.5	Windows 下双机的串行通信系统设计概述	247
7.5.1	系统模块图与面向对象方法介绍	248
7.5.2	封面设计	249
7.5.3	主界面设计	252
7.6	Windows 下双机的串行通信系统的调试与实现	262
7.6.1	系统调试出现的问题	262
7.6.2	系统的实现	263
7.7	本章小结	263
第 8 章	16 位高速 DSP 增强型同步串口的设计	265
8.1	概述	266
8.2	F206 DSP 处理器体系结构分析	266
8.2.1	F206 DSP 处理器概述	266
8.2.2	总线结构	268
8.2.3	中央处理单元概述	269
8.2.4	存储器和 I/O 空间	272
8.3	增强型同步串口的系统及设计	273
8.3.1	同步串口的基本原理	274
8.3.2	同步串口基本操作结构图	275
8.3.3	各种信号	275
8.3.4	缓存器与寄存器	276
8.3.5	中断	277
8.3.6	查错	277

8.4	接收电路的设计	280
8.4.1	突发模式的接收	280
8.4.2	连续模式的接收	281
8.5	发送电路的设计	282
8.5.1	利用内部帧同步的突发模式传送	283
8.5.2	利用外部帧同步的突发模式传送	284
8.5.3	利用内部帧同步的连续模式传送	285
8.5.4	利用外部帧同步的连续模式传送	286
8.6	接收与发送电路的实现与研究	286
8.7	同步串口中帧同步的设计	287
8.7.1	并行同步设计思想	288
8.7.2	多路并行帧同步系统	288
8.8	FIFO 缓存器电路的设计	289
8.8.1	FIFO 基本原理	289
8.8.2	通用 FIFO 的设计	290
8.8.3	同步串口中 FIFO 的设计	291
8.9	同步串口中特殊功能的设计	293
8.9.1	内部时钟和帧同步电路的设计	293
8.9.2	多通道选择电路的设计	294
8.10	同步串口中的状态寄存器	296
8.11	本章小结	297
第 9 章	串口与以太网数据传输实现	299
9.1	概述	300
9.2	设计芯片的软硬件选择	301
9.2.1	嵌入式网络模块 DSLC-SOM-01	301
9.2.2	CT-xweb2000 系统	302
9.2.3	ZNE-100T 增强型嵌入式以太网转串口模块	303
9.2.4	RCM2200 模块	305
9.2.5	各开发包的优缺点及其最终选择	306
9.3	Rabbit 开发包和编程工具 Dynamic C 简介	306
9.3.1	Rabbit 微处理器结构图与 Rabbit2000 特点介绍	307

9.3.2	RCM2200 系统结构与详细技术参数	309
9.3.3	Dynamic C 的特点	310
9.4	传输协议概述	311
9.5	串口传输协议的选择	312
9.5.1	XMODEM 协议概述	313
9.5.2	XMODEM 传输协议的实现函数	313
9.5.3	CHECKSUM 校验方法	314
9.6	以太网口传输协议的选择与实现	316
9.6.1	网络传输协议的选择和 UDP 协议	316
9.6.2	TFTP 服务器和客户端概述	317
9.6.3	TFTP 传输的初始连接	318
9.6.4	TFTP 包	318
9.6.5	TFTP 传输的正常终止	319
9.6.6	TFTP 协议的实现	320
9.7	串口到以太网口传输文件的实现	321
9.7.1	程序设计思想	321
9.7.2	程序实现	322
9.7.3	串口到以太网口文件传输源程序	323
9.8	以太网口到串口传输文件的实现	331
9.8.1	程序设计思想	331
9.8.2	程序实现	332
9.8.3	以太网口到串口文件传输源程序	332
9.9	工程应用概述	340
9.10	本章小结	341
第 10 章	基于串口的 DNC 信息采集系统的开发	343
10.1	DNC 技术概述	344
10.1.1	DNC 技术的产生与发展	344
10.1.2	DNC 数据采集的重要性	345
10.1.3	DNC 数据采集的现状	346
10.2	DNC 信息采集系统的功能	346
10.3	DNC 信息采集系统的底层设备接口	347
10.4	DNC 数据采集系统通信技术	348

10.4.1	DNC 数据采集系统的 内部通信技术概述	349	11.2.1	USB 系统拓扑结构	386
10.4.2	DNC 数据采集系统的 外部通信技术概述	351	11.2.2	USB 总线逻辑结构	388
10.4.3	DNC 数据采集的方法	352	11.2.3	传输协议	388
10.5	基于串口的 DNC 信息采集 系统的总体设计	354	11.2.4	传输类型	394
10.5.1	系统总体结构	355	11.2.5	设备框架	399
10.5.2	系统硬件组成	355	11.2.6	USB 主机协议	402
10.5.3	系统软件组成及 主要功能	356	11.3	USB 转 RS-232 桥接器硬件 设计	404
10.6	DNC 信息采集系统相关 技术	358	11.3.1	系统整体结构	404
10.6.1	串口通信技术	358	11.3.2	USB 接口设计	406
10.6.2	宏指令采集	359	11.3.3	UART 设计	407
10.6.3	特殊程序上报采集	364	11.3.4	Buffer 设计	409
10.6.4	采集数据处理与信息 发布	365	11.3.5	FIFO 设计	409
10.6.5	串口传输速度匹配	366	11.4	桥接器驱动模型分析与 实现机制	411
10.7	基于串口的 DNC 信息采集 系统设计和开发	367	11.4.1	Windows 驱动模型的 发展	412
10.7.1	系统概述	367	11.4.2	WDM 驱动模型简介	412
10.7.2	数据库设计	367	11.4.3	基于 WDM 模型的桥接器 驱动框架设计	414
10.7.3	串口通信实现	369	11.4.4	确立开发方案	418
10.7.4	多线程的实现	369	11.4.5	I/O 请求包 (IRP)	418
10.7.5	数据采集的实现	370	11.4.6	USB 数据处理	420
10.7.6	信息发布实现	372	11.4.7	内存分配策略	421
10.7.7	系统测试	374	11.4.8	同步问题	422
10.7.8	系统的测试效果	377	11.4.9	使用推迟过程调用	424
10.8	本章小结	378	11.4.10	使用完成例程	425
第 11 章	Windows XP 下 USB 转 RS- 232 桥接器驱动程序开发	379	11.5	桥接器驱动程序重点例程 设计	426
11.1	USB 转 RS-232 桥接器概述	380	11.5.1	驱动程序入口例程	426
11.1.1	设备驱动的概念	380	11.5.2	即插即用例程实现策略	428
11.1.2	USB 技术特点	380	11.5.3	分发例程实现策略	432
11.1.3	USB 的广泛应用	381	11.5.4	电源管理例程	438
11.1.4	USB 在嵌入式设备中的 应用	383	11.5.5	卸载例程	438
11.1.5	计算机常用外部总线 比较	383	11.6	驱动测试与安装	438
11.1.6	USB 转 RS-232 桥接器 发展现状	385	11.6.1	驱动测试	438
11.1.7	USB 转 RS-232 桥接器 驱动发展现状	385	11.6.2	驱动的安装	440
11.2	USB 总线技术介绍	386	11.7	本章小结	441
			第 12 章	串口通信在机器人实时 控制中的应用开发	443
			12.1	工业机器人概述	444
			12.1.1	工业机器人的发展	444
			12.1.2	工业机器人的应用	445
			12.1.3	工业机器人技术概述	446
			12.1.4	机器人控制技术概述	447

12.1.5	串行通信	449	12.4	实时控制系统的实现	459
12.1.6	实时控制系统	450	12.4.1	程序设计思路	459
12.2	MOTOMAN UP6 工业		12.4.2	编程语言及方法选择	460
	机器人系统介绍	450	12.4.3	Visual C++ 环境配置	462
12.2.1	MOTOMAN UP6 的		12.4.4	通信参数设置与建立	
	结构与性能	451		连接	463
12.2.2	MOTOMAN UP6 机器人		12.4.5	机器人的主要状态	463
	控制系统	451	12.4.6	机器人通信功能模块及其	
12.2.3	MOTOMAN UP6 机器人			具体实现	464
	运动参数	452	12.4.7	机器人可执行文件格式	480
12.2.4	机器人虚拟样机技术	453	12.4.8	编程注意事项	481
12.2.5	机器人运动仿真	454	12.4.9	系统软件发布	482
12.3	实时控制系统的总体分析	455	12.5	控制系统中图像监控的	
12.3.1	系统总体结构和组成	455		实现	482
12.3.2	机器人功能分析	456	12.5.1	视频捕获	482
12.3.3	通信协议分析	456	12.5.2	多线程编程技术在图像	
12.3.4	RS-232C 的不足之处及			监控中的运用	483
	修正方案	457	12.6	本章小结	485
12.3.5	机器人控制系统的				
	图像监控	458			

第 1 章

串行通信的基本概念

作为计算机接口软件的开发人员，非常有必要对硬件相关的原理和协议有一定的了解。本章将介绍与串行通信密切相关的理论和技术概念。首先从通信问题的数学模型切入，在此基础之上，解释 EIA RS-232 的基本原则、规范，以及高层次协议设计的必要性。我们还将介绍串口通信接口——一种在计算机体系当中几十年来一直保持不变动的通信构架。

串口通信是串行通信实现的一种方式，也是最古老的方式之一。然而，时至今日，尽管速度快、稳定性高的串行通信体系层出不穷，但其底层在实现上与串口通信并无实质性的差别。

1.1 从电路到通信系统

电子电路与电气电路在目的上首先就有很大的不同。电气电路强调所传递的能量，而电子电路则关注在一定的能量下，能够传递多少信息。

通信系统是复杂的电子电路。如果将通信系统狭义地用到计算机数据接口通信，则它仍然是一个复杂的电子电路。那么，在电子电路到通信系统的演化过程中，复杂程度的增加难道仅仅意味着电子器件数量的变化，或者电路拓扑结构的复杂化？

一个系统一旦涉及到信息的传递，则经典的电路分析理论就显得不足了，这时需要更高层的数学工具对系统的性态和行为进行分析。在这些工具中，首先用到的就是信息论。

1.1.1 应用信息论简要

在信息论中，首先被定义的是信息。

信息是与事件的随机性密切相关的。简单地说，如果某个事物或者过程有若干不确定的状态，那么我们就说这个事物或者过程含有一定量的信息。也就是说，事物对于不了解它的人来说，含有信息。对于事物进行了解的过程也就是消除一些不确定性的过程，是信息的传递。

信息论就是这样一个讨论信息本身以及信息传递过程的理论。为了表述信息以及信息的传递过程，信息论使用熵和互信息这两个概念。

熵是事物所含信息的度量，其实质是概率分布到正实数的一个映射关系。但并非所有的映射关系都能够称做熵。信息论给出了一系列的条件，这些条件在今天被看做是一组公理。

使用概率负对数的期望作为熵，是可以被证实满足熵公理的。

假设这样一种情况：我们取一枚均匀的硬币投向空中，硬币在空中翻滚的过程中，我们不能够预测落地后的硬币哪一面朝上。此时我们说这个事件含有信息。

虽然我们不知道硬币最终哪一面朝上，但是我们知道的是，硬币最终的状态只可能有三种，一种是正面朝上，一种是反面朝上，一种是恰好竖立在地面上。经验表明，第三种情况的可能性微乎其微，我们称之为小概率事件，或者称为不可能事件。前两种情况则各占一半的可能性。

我们知道，在求期望的过程中，小概率事件的成分可以被忽略（这个结论不是显然的，而是因为无穷小与无穷小的对数的乘积趋于 0）。因此，可以认为，前两种情况各占 50% 的概率。那么这个分布就是 (50%, 50%)，我们求这个分布的概率负对数的期望。其求法是，对于每一个概率，先求对数，然后求相反数，再与概率本身相乘，最后把每个概率算得的积相加。

很容易得知，如果以 2 为底，计算上面所示分布的负对数期望，其结果应该是 1。这也是硬币翻滚时，以熵作为度量的所含有的信息。当所选对数的底为 2 时，我们把熵的单位称为比特 (bit)。换句话说，翻滚中的硬币含有 1 比特的信息。

这里的比特和计算机系统中度量信息的比特的含义完全相同。这是因为，任何存储系统均以两种对立的状态存储信息。在未知处于何种状态的情况时，可以认为存储的数据和翻滚中的硬币没有区别。

已经确定状态的事件含有多少信息？由于确定事件可以看做（100%,0,0,0,...）这样的分布，那么无论以何为底求对数，其期望结果都为0。也就是说，确定的事件不含有任何信息。

我们看到，信息论对于事件本身处于某种状态，以及事件产生某个结果，并不关心。信息论分析的重点在于事件所处的若干种状态各自的可能性，也就是所谓的概率分布。概率分布的确定决定了信息量的确定，而事件状态的确定过程意味着信息量坍缩为0。

信息的传递如何描述？刚才说过，消除不确定性就是传递了信息，但是，消除不确定性是个后验的，如何通过统计规律确定信息的传递？

信息论认为，如果一个事件X不同的状态会导致另外一个事件Y发生概率分布的变化，就认为有信息进行了传递，反之亦然。事件X称为信源，事件Y称为信宿。

传递的信息量是通过互信息进行表述的。说到互信息，就不得不涉及条件熵。什么是条件熵？先取X的一个状态，观察Y的概率分布。由于X的状态已经确定，所以此时Y的概率分布也是确定的。我们来计算这个熵，这个熵因为和X的某个特定状态有关联，我们不妨称为在X状态下的状态关联熵（不同的论著对这个熵的命名各有不同）。X的不同状态有不同的概率，我们把状态关联熵以其所关联的状态概率进行加权求和，这个和就是已知X分布的条件下，Y的条件熵。状态关联熵所表述的模型称为信道。

总的来看，事件Y含有多少信息？我们可以想像这样一种场景：一个小孩放学回家。他有75%的可能性会回自己家，有25%的可能性去同学家。不碰巧的是，如果他去了同学家，妈妈很有可能打电话叫他立即回家，这个事情发生的可能性是80%。问题是，总的来说，这个小孩最终回到自己家的可能性有多大？不难计算，应该是 $75\%+25\%\times 80\%$ 。这个算法称为贝叶斯公式，其计算的依据是排列与组合中的加法和乘法原理。那么小孩回家这件事含有多少信息？实际上相当于计算一个 $(75\%+25\%\times 80\%, 1-75\%+25\%\times 80\%)$ 分布的熵。信宿Y的熵也这样计算。

互信息就定义为信宿的熵与已知信源分布的条件下信宿的条件熵之差。想像一种极端情况：无论X处于何种状态，Y的概率分布均相同，此时所计算的条件熵和Y本身的信息量相同，互信息为0，这就是说，在此过程中传递了量为0的信息，即，一个事件X不同的状态没有导致事件Y发生概率分布的变化，就会传递量为0的信息。同时可以严格证明，互信息总是正数。

信源、信宿与信道是通信中的基本要素。通常，信源和信宿具有一致性，即由信源的特性决定信宿的特性，也就是我们常说的收发机协议匹配。信道常常被独立考虑。

以经典的概率与统计理论为基础，信息论通过严格的数学证明，发展了一系列结论，其中最著名的就是编码理论，这是现代通信技术的理论基础。

编码理论主要讨论在点对点的信息传输中，应该如何对信息进行表达，以获得特定评估方法范围内的最优化。

如图1-1所示为一个点对点的通信系统。

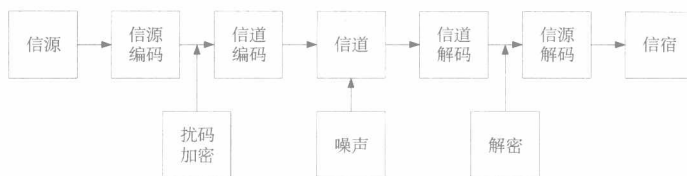


图 1-1 点对点的通信系统

编码理论分为三个方向，即无损信源编码、有损信源编码和信道编码。信息论的创立者建立了一整套严格的数学体系对这三种编码进行了描述。

信源就是产生事件的个体，事件的不同结果称做信源的不同状态，每一个状态称为一个符号（Symbol）。如果把信源看做概率分布含有时间参数的随机事件，即随机过程，则无损信源编码就需要解决以下若干问题：

- 如何用一定长的码（Code）序列表达信源的符号（编码）。
- 如何评价编码的有效性。
- 接收端恢复信源符号的复杂性如何。

我们以投骰子为例，投骰子结果的可能性有 6 种。如果将投骰子看做信源，则 6 种结果就是这个信源的 6 个符号。假设这样一个情景，一个人 A 在不停地投骰子（随机过程），另外一个人 B 通过电报机把每一次投掷的结果告诉相距甚远的朋友 C。

如果按照最一般的方法使用电报机，那么电报机每一次只能发长和短两个码。不妨称长码为 1，短码为 0。用一段 0 和 1 的序列表示投骰子的结果，这就是二进制信源编码。“二进制”的意思是每一个码有两种取值。

采用多少进制的码要取决于信道。如果信道只允许传送两个状态，就必须用二进制的码。由于一个码最多代表两个符号，所以要使用多个码表示更多的符号。由于要这些码依次穿过信道，所以就有了码串（Word，也叫码字）的概念。编码就是用不同的码串代表不同的符号，码串之间的互斥保证接收端准确地还原符号。有了码串之后，最关键的是信源的符号如何与码串建立起一一对应的关系。

在骰子问题中如何编码？首先想到的是采取十进制转二进制编码的方法。码串 0 对应点数 1，1 对应点数 2，10 对应点数 3，11 对应点数 4，100 对应点数 5，101 对应点数 6。信息论采用了平均码长描述编码方案的效率。平均码长就是每一个信源符号对应的码串长度与信源符号概率的加权和。按照平均码长的定义，这种编码的平均码长是 $1/6 \times 1 + 1/6 \times 1 + 1/6 \times 2 + 1/6 \times 2 + 1/6 \times 3 + 1/6 \times 3 = 1.5$ （bit）。

使用这种编码的前提是，每一个编码之间的时间间隔要足够长，否则接收端将不能够区分不同的码串。例如，一个 4bit 的串 0100，接收端既可以解释为 0、10、0，也可以解释为 0、100。所谓“间隔”，就是指除了 0 和 1 之外的“第三态”。在电报通信中，可以利用无消息表示间隔，然而在数字通信中，二进制非 0 即 1，不定态是不允许的。

一个简单的解决方案是等长编码。在不足 3bit 的编码前补 0 使之达到 3bit，令接收端每三个比特截取一次码串并译码。这样就避免了码间模糊的问题，但是这无形之中增加了码长。3bit 等长编码的平均码长是 3bit。

等长编码在大规模的符号量面前会带来码串长度的指数级暴涨。比如，16bit 的等长码可以对 65 536 个符号进行编码，但是若要对 65 537 个符号编码，就必须使用 17bit 的码长，17bit