

《电脑编程技巧与维护》杂志十周年庆典暨真情回馈读者活动
《电脑编程技巧与维护》杂志社策划

编程技巧典型案例集锦系列

《电脑编程技巧与维护》杂志社 编著

Visual C++

Visual C++ Programming

编程技巧典型案例解析

Visual C++ Programming Visual C++ Programming

——网络与通信及计算机安全与维护篇

- 掌握 DLL 编程，在动态库中建立串口监视线程的实时串口通信
- 如何通过内存映射文件和复杂的 DLL 注入技术在应用程序之间共享代码和数据
- 管理多线程，包括网络文件传输的设计、局域网络会议的开发以及线程与主线程间通信的安全性问题
- 探讨 Socket 底层编程，分析 Socket 网络程序中参数对传输性能的影响
- 一阶贝塞尔曲线的信息掩藏算法和 IDEA 算法的数据加密技术

赠送超值 1CD，内含实例源代码，
编程高手经验汇集，现学现用



中国电力出版社
www.infopower.com.cn

《电脑编程技巧与维护》杂志十周年庆典暨真情回馈读者活动
《电脑编程技巧与维护》杂志社策划

编程技巧典型案例集锦系列

Visual C++

编程技巧典型案例解析

——网络与通信及计算机安全与维护篇

《电脑编程技巧与维护》杂志社 编著



中国电力出版社

www.infopower.com.cn

内 容 简 介

本书以实例的形式讲解了利用Visual C++语言实现网络与通信编程及计算机安全与维护编程的典型技巧和方法。全书共56个实例,分别对32位串口通信程序的编程方法、Winsoc编程、网络监控、MFC多线程技术、WinDriver在基于机器指纹的软件加密中的应用、MIRACL大数运算库实现对共享软件的加密、基于IDEA算法的加密工具的使用、贝塞尔曲线的信息掩藏算法、软件加密的反拷贝技术等进行了详细讲解,并附带了相关源代码。

本书是编程人员的经验之谈,适合于软件开发、网络管理、数据加密解密人员及其他编程爱好者、高等学校的学生等相关人员使用。

图书在版编目(CIP)数据

Visual C++编程技巧典型案例解析.网络与通信及计算机安全与维护篇/《电脑编程技巧与维护》杂志社编著. —北京:中国电力出版社,2005

(编程技巧典型案例集锦系列)

ISBN 7-5083-3146-X

I.V... II.电... III.①计算机网络—C语言—程序设计—案例—分析②计算机通信—C语言—程序设计—案例—分析③电子计算机—安全技术—C语言—程序设计—案例—分析 IV.TP312

中国版本图书馆CIP数据核字(2005)第008012号

版权声明

本书由中国电力出版社独家出版。未经出版者书面许可,任何单位和个人不得以任何形式复制或传播本书的部分或全部内容。

本书内容所提及的公司及个人名称、产品名称、优秀作品及其名称,均为所属公司或者个人所有,本书引用仅为宣传之用,绝无侵权之意,特此声明。

策 划:裴红义

姚贵胜

责任编辑:夏华香

责任校对:崔燕菊

责任印制:李志强

丛 书 名:编程技巧典型案例集锦系列

书 名:Visual C++编程技巧典型案例解析——网络与通信及计算机安全与维护篇

编 著:《电脑编程技巧与维护》杂志社

出版发行:中国电力出版社

地址:北京市三里河路6号 邮政编码:100044

电话:(010)88515918 传真:(010)88518169

印 刷:北京市铁成印刷厂

开本尺寸:185×260

印 张:21.5

书 号:ISBN 7-5083-3146-X

版 次:2005年7月北京第1版

印 次:2005年7月第1次印刷

印 数:1~5000

定 价:39.00元(含1CD)

丛书序

在《电脑编程技巧与维护》杂志创刊 10 周年之际，为了真诚回报多年来一直关爱和支持本刊的广大读者，《电脑编程技巧与维护》杂志社和中国电力出版社共同策划出版了《编程技巧典型案例集锦系列》丛书。《电脑编程技巧与维护》杂志是为从事电脑编程、系统应用开发人员创办的专业性和实用性都很强的技术刊物，它从 1994 年创刊，十多年来始终遵循着“实用第一，智慧密集”的办刊宗旨，紧跟计算机软硬件技术发展和应用趋势，不断求变创新，针对软件开发过程中许多关键技术问题，着重提供各类解决方案。对电脑编程人员来说，程序开发能力的提高，除了对语言和算法的学习外，还要集思广益，充分借鉴参考别人的长处，深入透彻地理解其中的精髓，然后融入到自己的设计方案中去，这样无论是对于自身还是整体都有莫大的提高，这也正是我们编写这套系列丛书的初衷。

本丛书包括《Visual C++ 编程技巧典型案例解析——基础与应用篇（上）》、《Visual C++ 编程技巧典型案例解析——基础与应用篇（下）》、《Visual C++ 编程技巧典型案例解析——图形图像处理与数据库篇》、《Visual C++ 编程技巧典型案例解析——网络与通信及计算机安全与维护篇》、《Visual Basic 编程技巧典型案例解析》、《Delphi 编程技巧典型案例解析》、《C# 编程技巧典型案例解析》、《Java 编程技巧典型案例解析》、《PowerBuilder 管理信息系统编程技巧典型案例解析》9 册共 545 个典型案例。每册书的编程案例，均依不同的编程应用分成若干章，条目清晰可查，使用极为方便。

本丛书选编了《电脑编程技巧与维护》杂志近一两年发表的和一部分尚未发表而又极为实用、精彩的典型编程实例，特点是：其各册内容均来自编程高手的智慧，凝结了 500 余位编程高手与名家的心血，关键技术专家点评；其案例是从实际项目提炼出的开发范例，超过 800 个技术要点的经典解决方案。案例讲解部分先给出设计目标，然后介绍实现目标的基本思想和方法，最后详细给出其核心程序的源代码，对程序的关键部分进行讲解并给出程序的运行效果；其编程技巧新颖实用，构思巧妙，汇集了众多顶级程序员和业界知名专家的成功经验，告诉读者最好的创意和最实用的方法。全套书既讲究内容的深入性、专业性和权威性，同时兼顾轻松、通俗易懂、时效性强的特点，带给读者的是一份清

新、纯粹的体验感受。

本丛书是《电脑编程技巧与维护》杂志资源的二次开发，浓缩了当前主流编程语言 Visual C++、Visual Basic、Delphi、Java、C#、PowerBuilder 等程序设计的精华，其目的是力求为读者建造一个真正的知识整合，是编程思想、编程技术、技巧交流的平台，让读者从中学习到编程高手的诀窍，丰富读者的编程技巧，拓宽读者的编程思路，迅速提升读者的程序开发能力。该丛书可作为高等院校学生进行课程项目开发、毕业项目设计的参考教材，软件从业人员及编程爱好者的珍藏宝典，也可作为高等培训学校的案例教程。

实例导航学编程，自学成才成高手，思想、智慧、理念、经验、技巧无处不在……

《电脑编程技巧与维护》杂志社

2005年1月

前 言

Visual C++ 作为功能强大的面向对象与可视化应用程序开发工具，是业界公认的优秀应用开发工具。Microsoft 的基本类库 MFC 将类之间的关系紧密地联系在一起，而 Visual C++ 支持 MFC 的程序开发，提高了 MFC Application Wizard 的功能，帮助程序员构建一套基础程序，并从中开发应用程序，适合做各种系统软件、应用软件、网络软件、游戏软件等开发平台。

在《编程技巧典型案例集锦系列》丛书中，Visual C++ 编程技巧典型案例精选了《电脑编程技巧与维护》杂志近两年半共 30 期已发表的精彩编程实例 238 例。根据 Visual C++ 的不同应用对象，将其分为《Visual C++ 编程技巧典型案例解析——基础与应用篇（上）》、《Visual C++ 编程技巧典型案例解析——基础与应用篇（下）》、《Visual C++ 编程技巧典型案例解析——图形图像处理与数据库篇》、《Visual C++ 编程技巧典型案例解析——网络与通信及计算机安全与维护篇》4 册出版。每一册书都始终遵循着“实用第一，智慧密集”的宗旨，介绍 Visual C++ 开发各类应用程序关键技术的解决案例，每一个案例都给出了开发过程、技术难点及其解决的方法和技巧，涉及到 Visual C++ 应用程序设计的新思路和新方法。这些典型案例所涵盖的编程技巧是经验的总结，具有一定的代表性，很值得借鉴。本着实用的原则，紧紧围绕着一个主题展开，循序渐进、由浅入深地介绍了使用 Visual C++ 进行应用程序开发的思想方法与编程技巧，在此我们愿与更多的程序员分享这些经验。

本书为《Visual C++ 编程技巧典型案例解析——网络与通信及计算机安全与维护篇》，全书分两章共计 56 个案例。第 1 章介绍网络与通信编程实例，精选了 28 个 Visual C++ 在网络与通信应用中的典型而实用的编程案例；第 2 章介绍计算机安全与维护编程实例，也精选了 28 个 Visual C++ 在计算机安全与维护应用中的典型而实用的编程案例。

本书的特色体现在如下几点：第一，每一章都是通过一个个的实例来介绍 Visual C++ 应用编程方法和技巧，避免枯燥、空洞的理论讲解，并且每一个实例都具有很强的

实用性和代表性。在实例的讲解上一般都是先给出设计目标，接着介绍实现该目标的基本思想和方法，然后详细给出其核心程序的源代码，对程序的关键部分进行讲解并给出程序的运行效果。第二，所选的每一个实例都是从事 Visual C++ 应用编程人员的经验总结，具有很强的实用性，其中的编程技巧可供借鉴。第三，每一个实例的程序源代码都是经过上机调试通过的，给程序开发人员移植源代码带来了方便，加快了应用编程的步伐。第四，对个别版本和开发环境稍微低一些的经典实例进行点评和分析，其目的是对读者能够起到触类旁通的作用。

本书是《电脑编程技巧与维护》杂志资源的二次开发，浓缩了 Visual C++ 网络与通信及计算机安全与维护应用程序设计的精华，其目的是提升读者 Visual C++ 程序开发的能力，把应用 Visual C++ 进行编程的心得体会、经验与大家共享。本书定位于有 Visual C++ 应用基础的编程人员和应用开发人员，对初学 Visual C++ 编程的新手也有一定的参考价值。本书内容深入、概念清晰、层次分明、例题典型而实用，但不足甚至疏漏之处在所难免，恳请广大读者批评指正。

《电脑编程技巧与维护》杂志社

2005 年 1 月

目 录

丛书序

前 言

第 1 章 网络与通信编程实例

实例 1	用 Visual C++ 编写 32 位串行通信程序	3
实例 2	Visual C++ 编制串行通信库	6
实例 3	动态库(DLL)中建立串口监视线程的实时串口通信	10
实例 4	Visual C++ 平台下实现 Foxmail 邮件转化器	14
实例 5	利用 Winsock 编程实现局域网上所有 IP 包的捕获和分析	18
实例 6	实现 Windows 与 Unix 系统数据通信	21
实例 7	设计实现 Windows 2000 下网络监控平台软件	24
实例 8	WinInet 类集开发 Internet 客户应用程序探索	28
实例 9	用 C++ 创建 ASP 服务器组件	32
实例 10	利用 Visual C++ 实现 DDE 客户机/服务器应用	37
实例 11	Visual C++ 编程实现网络嗅探器	42
实例 12	NetMeeting SDK 的网络视频监视系统	47
实例 13	用 Visual C++ 制作 ASP 组件	52
实例 14	实现 DOS 和 Windows 应用程序之间 IPX 实时数据通信	57
实例 15	Linux XWindows 环境下用 Kylix 3(C++)实现串口通信	63
实例 16	在 Visual C++ 6.0 下利用共享内存、消息实现内部进程通信	71
实例 17	利用 MFC 多线程技术开发基于 UDP 数据广播的局域网络会议程序	78
实例 18	一种基于非对称多播网络的传输控件设计	84
实例 19	Socket 网络程序中参数对流传输性能的影响分析	88
实例 20	多线程网络文件传输的设计与实现	97
实例 21	Windows 编程实现 MC35 数据传输	107
实例 22	基于 Web 的免费手机短信	119
实例 23	Pocket PC Phone Edition 中的短信息编程技术	130
实例 24	Linux 下的套接口底层协议编程及其应用实例	134
实例 25	多串口通信的软件实现方法	138
实例 26	Visual C++ 下使用 SMAPI 为应用程序添加信报收发功能	141

实例 27	Visual C++ 与 DHTML 混合编程并实现与 JSP 语言的参数传递	146
实例 28	利用 Visual C++ 6.0 编写多功能数据处理显示程序	150

第 2 章 计算机安全与维护编程实例

实例 29	WinDriver 在基于机器指纹的软件加密中的应用	171
实例 30	在 Win32 应用程序中直接读写软盘物理扇区	177
实例 31	Visual C++ 编程实现在 Windows 2000 下对磁盘扇区数据的直接读写	186
实例 32	在 Windows XP 平台上直接读写硬盘扇区	190
实例 33	用 Windows API 配置打印机属性	193
实例 34	AutoCAD 中动态拖动及交互调整功能的实现	196
实例 35	用 Visual C++ 实现程序自删除的一种方法	202
实例 36	网络环境下,通过遍历注册表获取“添加/删除程序”对话框中的信息	205
实例 37	Visual C++ 下利用 C++ Test 进行软件单元测试	210
实例 38	一种简单而实用的 Windows 应用程序反动态跟踪方法	217
实例 39	端口扫描方法的原理、实现和防御	220
实例 40	借助 MIRACL 大数运算库实现对共享软件的安全注册	227
实例 41	基于一阶 Bézier 曲线的信息隐藏算法	232
实例 42	利用 ANTLR 生成 C++ 描述的分析程序	238
实例 43	在 Windows NT/2000/XP 下登录认证方式的替换和控制编程方法	247
实例 44	实现基于 IDEA 算法的加密工具	255
实例 45	基于文件时间信息和 IDEA 算法的数据加密	265
实例 46	使用 Visual C++ 实现工作者线程与主线程间的可靠、安全通信	269
实例 47	在 Visual C++ 6.0 下利用 Rijndael 算法进行任意文件的加密	274
实例 48	基于 Linux 环境下包过滤防火墙的设计与实现	279
实例 49	使用 VxD 实现 Windows 98 环境下的文件和目录隐藏	295
实例 50	利用 Windows Hook 技术监控计算机的使用	307
实例 51	用 Visual C++ 实现数字图像迭代混沌加密方法	311
实例 52	软件加密解密之动态跟踪	314
实例 53	远程计算机重启原理及实现	318
实例 54	基于 IDEA 加密的局域网 OTP 身份认证模型的研究与设计	323
实例 55	MIDAS 数据安全	328
实例 56	软件加密之反拷贝技术	330

第 1 章

网络与通信编程实例



■ 实例 1

用 Visual C++ 编写 32 位串行通信程序

在 Windows 应用程序的开发中，大家经常会遇到工业控制计算机与外围设备（如：调制解调器、通信电源、智能空调、智能油机或其他一些智能设备）进行通信的问题。通信接口有串口、CAN 总线口、网口等多种类型，现在仅就串口通信进行一下讨论。

随着 32 位应用程序的普及，在 Windows 环境下编写串行通信程序的方法也有很大的变化。因此，如何在 32 位应用程序中对计算机的串行通信资源进行操作成为一个问题。

本文详细介绍了编写 32 位串行通信程序的方法及实例，仅供大家参考。

一、编写串口通信程序的基本操作步骤

基本操作步骤如图 1-1 所示。

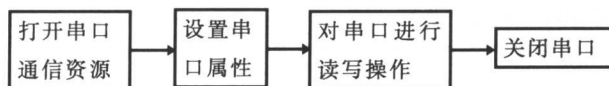


图 1-1

二、具体实现方法

```

#include <winbase.h>
HANDLE m_hCom; //定义串口句柄
// =====
// 功能:打开资源
// 参数:ComNo:串口号,Baut:波特率,Data:数据位,Stop:停止位
//Parity:校验位,cbInBuf:输入队列大小,cbOutBuf:输出队列大小
// =====
BOOL OpenCom(int ComNo, DWORD Baut, int Data, int Stop, nt Parity, DWORD cbInBuf,
DWORD cbOutBuf)
{
    DCB Dcb;
    //Dcb是指向DCB结构的指针,该结构包含串口的各种属性,如:波特率、数据位、停止位、校验
    //位等
    COMMTIMEOUTS TimeOut;
    if(INVALID_HANDLE_VALUE! = m_hCom)
        CloseHandle(m_hCom);
    char p[30];
    sprintf(p, "COM%d", ComNo);
    // 用 CreateFile 函数打开串口通信资源
  
```

```

m_hCom = CreateFile(p, GENERIC_READ|GENERIC_WRITE, 0, NULL, OPEN_EXISTING,
NULL, NULL);
if (m_hCom == INVALID_HANDLE_VALUE) return FALSE;
GetCommState(m_hCom, & Dcb);
// 读取串口参数
Dcb. BaudRate = Baut;
Dcb. ByteSize = Data;
Dcb. StopBits = Stop; // 0, 1, 2 = 1, 1.5, 2
Dcb. Parity = Parity; // 0 - 4 = no, odd, even, mark, space
if (! SetCommState(m_hCom, & Dcb))
// 设置串口参数
{
    CloseHandle(m_hCom);
    m_hCom = INVALID_HANDLE_VALUE;
    return FALSE;
}
memset(& TimeOut, 0, sizeof(TimeOut));
TimeOut. ReadIntervalTimeout = MAXDWORD;
// 串口读写操作后立即返回
SetCommTimeouts(m_hCom, & TimeOut);
SetupComm(m_hCom, cbInBuf, cbOutBuf);
// 设置串口的接收、发送缓冲区大小
return TRUE;
}

// =====
// 功能: 获取串行口缓冲区的字节数
// =====
int GetDataLen()
{
    if (m_hCom == INVALID_HANDLE_VALUE)
        return 0;
    COMSTAT comstat;
    DWORD dwError;
    if (ClearCommError(m_hCom, & dwError, & comstat))
        return comstat. cbInQue;
    dwError = GetLastError();
    return 0;
}

// =====
// 功能: 读串行口
// =====
DWORD ReadCom(BYTE * pBuff, int nCount)
{
    if (m_hCom == INVALID_HANDLE_VALUE)
        return 0;
    DWORD read = 0;
    ReadFile(m_hCom, pBuff, nCount, & read, NULL);
    // pBuff: 读取数据的存放缓冲区, nCount: 要读取数据的字节数, read: 实际从串口读取数据
    // 的字节数
    return read;
}

```

```

// =====
// 功能:写串口
// 参数:pBuff:数据缓冲区,nCount:要写入数据的字节数
// =====
BOOL WriteCom(BYTE * pBuff, int nCount)
{
    if(m_hCom == INVALID_HANDLE_VALUE)
        return FALSE;
    DWORD written = 0;
    BOOL ret = WriteFile(m_hCom, pBuff, nCount, & written, NULL);
    //pBuff:写数据的存放缓冲区,nCount:要写入数据的字节数,written:实际写入串口数据的字
    //节数
    return ret;
}
// =====
// 功能:关闭串口
// =====
BOOL CloseCom()
{
    if(m_hCom != INVALID_HANDLE_VALUE)
    {
        if(CloseHandle(m_hCom))
            //结束通信应用程序前必须关闭通信资源 m_hCom
            m_hCom = INVALID_HANDLE_VALUE;
    }
    return (m_hCom == INVALID_HANDLE_VALUE);
}

```

上述是串口操作的具体功能函数，编写串行通信程序时，先调用 `OpenCom` 函数打开串口资源，并对串口属性进行设置（如：串口号、波特率、数据位、停止位、校验位、串口接收/发送缓冲区的大小等属性），然后调用函数 `ReadCom`、`WriteCom` 对串口进行读写操作，最后在程序退出之前，不要忘记调用关闭串口函数 `CloseCom()`，释放资源。

(姜 丽)

■实例 2

Visual C++ + 编制串行通信库

串行通信是计算机之间有效的通信手段，因其高效可靠、标准统一而得到广泛应用。但由于 Windows 操作系统禁止应用程序直接和硬件打交道，因此程序员只能使用 Windows 提供的标准通信函数与硬件接口。而且直接在应用程序中调用单个的 Win32 API 通信函数进行串口操作相当烦琐，因此先用 Visual C++ 生成好的串行通信 DLL 全面封装有关的 API 通信函数，再由 Visual Basic、PowerBuilder 等各种高级应用程序调用，即可获得简洁、方便的串行通信功能。

一、函数和结构

程序中所用 API 通信函数简介：

CreateFile()：打开串口

ReadFile()：从串口读取数据

WriteFile()：发送数据

GetLastError()：获取错误信息

PurgeComm()：终止任何未完的后台读或写操作，并清除 I/O 缓冲区

EscapeCommFunction()：控制硬件通信信号，撤销连接

CloseHandle()：关闭串口

GetCommProperties()：获取串口属性信息

SetCommTimeouts()：实现超时设置

ClearCommError()：用于确定有关端口的状态信息

程序中所用通信结构简介如下：

LPCOMMPROP：存储端口属性信息

COMMTIMEOUTS：存储超时设置信息

COMSTAT：存储端口状态信息

二、程序功能

本程序通过生成的 EX22C.dll 共导出 5 个函数，即

```
bool open(int nport);
```

```
bool close();
```

```
int read(LPSTR buffer);
```

```
bool settime(int time);
```

```
int write(LPSTR buffer, int wlen);
```

分别实现打开、关闭串口、读、写和超时设置这 5 种功能。另外，因串口的通信参数可由 Win-

Windows 操作系统的设备管理器对话框简便地设置, 所以此功能未被包括。用户还可根据需要对以上函数进行扩充。

三、设计步骤

(1) 在 Visual C++ 中新建 MFC AppWizard (DLL) 工程 EX22C, 并在 AppWizard 选项卡中选择 Regular DLL Using Shared MFC DLL。

(2) 在 EX22C.cpp 文件中增加如下内容:

```
HANDLE hidcomdev; //打开串口的句柄,全局变量 bool opened = false;
//串口状态标志,全局变量
// =====
extern "C" _declspec(dllexport) bool open(int nport)
{
    if(opened) return(true);
    char sname[8];
    DWORD dwerror;
    wsprintf(sname, "COM%d", nport);
    hidcomdev = :: CreateFile(sname, GENERIC_READ|GENERIC_WRITE, 0, NULL,
    OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL|FILE_FLAG_OVERLAPPED, NULL);
    if(hidcomdev == INVALID_HANDLE_VALUE)
    {
        dwerror = :: GetLastError();
        AfxMessageBox("打开通信串口的操作失败,请检查参数设置后重试!");
        return(false);
    }
    opened = true;
    AfxMessageBox("通信串口已打开!");
    return(opened);
}
// =====
extern "C" _declspec(dllexport) bool close()
{
    if(!opened || hidcomdev == NULL) return(true);
    :: PurgeComm(hidcomdev, PURGE_TXABORT|PURGE_RXABORT|PURGE_TXCLEAR|
    PURGE_RXCLEAR);
    :: EscapeCommFunction(hidcomdev, CLRDTR);
    :: CloseHandle(hidcomdev);
    AfxMessageBox("通信串口已关闭!");
    opened = false;
    hidcomdev = NULL;
    return(true);
}
// =====
extern "C" _declspec(dllexport) int read(LPSTR buffer)
{
    int rlen;
    if(!opened || hidcomdev == NULL) return(0);
    DWORD dwbytesread, dwerrorflags;
    COMSTAT comstat;
    static OVERLAPPED ovrread;
```



```

:: ClearCommError(hidcomdev, & dwerrorflags, & comstat);
dwbytesread = (DWORD) comstat. cbInQue;
//获取接收缓冲区中等待被读取的字节数
rlen = (int) dwbytesread;
if(rlen > 300)
{
    dwbytesread = (DWORD) 300;
//限定最大被读取的字节数
if(!:: ReadFile(hidcomdev, buffer, dwbytesread,
& dwbytesread, & overlapped))
{
    :: GetLastError();
    return(0);
}
else{ rlen = (int) dwbytesread;
    return(rlen);
}
// = = = = =
extern "C" __declspec(dllexport) int write(LPSTR buffer, int wlen)
{
if(! opened || hidcomdev == NULL) return(0);
DWORD dwbyteswritten = 0;
static OVERLAPPED overlapped;
dwbyteswritten = (DWORD) wlen;
if(!:: WriteFile(hidcomdev, buffer, dwbyteswritten,
& dwbyteswritten, & overlapped))
{
    :: GetLastError();
    return(0);
}
else {
    return((int) dwbyteswritten);
}
}
// = = = = =
extern "C" __declspec(dllexport) bool settime(int time)
{
LPCOMMPROP cp;
COMMTIMEOUTS to;
cp = (LPCOMMPROP) malloc(1000);
//为 COMMPROP 结构外加任何附加数据分配内存
if(!:: GetCommProperties(hidcomdev, cp))
{
    return(false);
}
if(!(cp-> dwProvCapabilities & PCF_TOTALTIMEOUTS))
{ //确认超时设置被支持
AfxMessageBox("超时设置失败,退出!");
return(false);
}
}

```