

深入浅出VC 串口之第三方类（上）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022__E6_B7_B1_E5_85_A5_E6_B5_85_E5_c100_644766.htm 串口类 从本系列文章连载三、四可以看出，与通过WIN32 API进行串口访问相比，通过MScomm这个Activex控件进行串口访问要来的方便许多，它基本上可以向用户屏蔽多线程的细节，以事件（发出OnComm消息）方式实现串口的异步访问。尽管如此，MScomm控件的使用仍有诸多不便，譬如其发送和接收数据都要进行VARIANT类型对象与字符串的转化等。因此，国内外许多优秀的程序员自己编写了一些串口类，使用这些类，我们将可以更方便的操作串口。在笔者的《深入浅出Win32多线程程序设计之综合实例》（网址

：<http://dev.yesky.com>）一文中，曾向读者展示了由Remon Spekreijse编写的CSerialPort串口类，而本文将向您展示由程序员llbird编写的cnComm(中国串口?)串口类。llbird是一位优秀的程序员，他的代码风格简洁而紧凑，类的声明和实现都被定义在一个头文件中，使用这个类的朋友只需要在工程中包含这一头文件即可：/*Comm Base Library(WIN98/NT/2000) ver 1.1Compile by：BC 5. C BUILDER 4, 5, 6, X. VC 5, 6.

VC.NET. GCC.copyright(c) 2004.5 - 2005.8 llbird
wushaojian@21cn.com*/ #ifndef _CN_COMM_H_#define
_CN_COMM_H_#pragma warning(disable: 4530)#pragma
warning(disable: 4786)#pragma warning(disable: 4800)#include
#include #include //送到窗口的消息 WPARAM 端口号#define
ON_COM_RECEIVE WM_USER 618#define ON_COM_CTS

```

WM_USER 619 //LPARAM 1 valid#define ON_COM_DSR
WM_USER 621 //LPARAM 1 valid#define ON_COM_RING
WM_USER 623#define ON_COM_RLSD WM_USER 624#define
ON_COM_BREAK WM_USER 625#define
ON_COM_TXEMPTY WM_USER 626#define
ON_COM_ERROR WM_USER 627 //LPARAM save Error
ID#define DEFAULT_COM_MASK_EVENT EV_RXCHAR |
EV_ERR | EV_CTS | EV_DSR | EV_BREAK | EV_TXEMPTY |
EV_RING | EV_RLSDclass cnComm{ public:
//-----Construction-----
----- //第1个参数为是否在打开串口时启动监视线程，
第2个参数为IO方式 阻塞方式（0）/ 异步重叠方式（默认）
cnComm(bool fAutoBeginThread = true, DWORD dwIOMode
=FILE_FLAG_OVERLAPPED): _dwIOMode(dwIOMode),
_fAutoBeginThread(fAutoBeginThread) { Init(). } virtual
~cnComm() { Close(). UnInit(). }
//-----Attributes-----
----- //判断串口是否打开 inline bool IsOpen() { return
_hCommHandle != INVALID_HANDLE_VALUE. } //判断串口
是否打开 operator bool() { return _hCommHandle !=
INVALID_HANDLE_VALUE. } //获得串口句柄 inline
HANDLE GetHandle() { return _hCommHandle. } //获得串口句
柄 operator HANDLE() { return _hCommHandle. } //获得串口参
数 DCB DCB *GetState() { return IsOpen() 100Test 下载频道开通
，各类考试题目直接下载。详细请访问 www.100test.com

```