

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

https://www.100test.com/kao_ti2020/140/2021_2022__E6_B7_B1_E5_85_A5_E6_B5_85_E5_c100_140210.htm 2.实例 程序的功能和界面（如下图）都与本文连载三、四中《基于WIN32 API的串口编程》和《基于控件的串口编程》相同，不同的只是本节的串口通信要以IIBird定义的cnComm类来实现。我们需要为串口的接收事件定义一个用户消息ON_COM_RECEIVE，因此对话框的消息映射为

```
: BEGIN_MESSAGE_MAP(CSerialPortClassDlg,
CDialog)//{{AFX_MSG_MAP(CSerialPortClassDlg)
ON_WM_SYSCOMMAND() ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_CLEAR_BUTTON, OnClearButton)
ON_BN_CLICKED(IDC_SEND_BUTTON, OnSendButton)
ON_MESSAGE(ON_COM_RECEIVE, OnCommRecv)//}}AFX_
MSG_MAPEND_MESSAGE_MAP() 同时，我们需要在对话框
类的头文件中定义cnComm类的成员变量com和接收数据消息
处理函数OnCommRecv：cnComm com.afx_msg void
OnCommRecv(WPARAM wParam, LPARAM lParam). 在对话框
初始化时调用打开串口：BOOL
CSerialPortClassDlg::OnInitDialog(){ CDialog::OnInitDialog(). //
Add "About..." menu item to system menu. // IDM_ABOUTBOX
must be in the system command range.
ASSERT((IDM_ABOUTBOX & 0xFFFF) ==
IDM_ABOUTBOX). ASSERT(IDM_ABOUTBOX < CMenu
```

```
*pSysMenu = GetSystemMenu(FALSE). if (pSysMenu != NULL) {
CString strAboutMenu.
strAboutMenu.LoadString(IDS_ABOUTBOX). if
(!strAboutMenu.IsEmpty()) {
pSysMenu->AppendMenu(MF_SEPARATOR).
pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu). } } // Set the icon for this dialog. The framework
does this automatically // when the applications main window is not
a dialog SetIcon(m_hIcon, TRUE). // Set big icon
SetIcon(m_hIcon, FALSE). // Set small icon // TODO: Add extra
initialization here com.Open(1). //打开串口1并使用默认设置
com.SetWnd(AfxGetMainWnd()->m_hWnd). //设置消息处理窗
口 return TRUE. // return TRUE unless you set the focus to a
control} 发送字符串的过程很简单，只需要调用cnComm类
的Write函数：// "发送"按钮函数（完成数据的发送功能）void
CSerialPortClassDlg::OnSendButton(){ // TODO: Add your
control notification handler code here UpdateData(true).
com.Write(m_send). //发送字符串} 接收字符串的过程也很简
单，只需要调用cnComm类的ReadString函数： void
CSerialPortClassDlg::OnCommRecv(WPARAM wParam,
LPARAM lParam){ UpdateData(true). //读取串口上的字符 char
str[100]. com.ReadString(str, 100). m_recv = str.
UpdateData(false).} 读者朋友们这时一定会发出感慨：使
用cnComm类后，进行串口数据收发的程序是多么简单啊！
的确，串口的初始化、读写几乎都是用1~2条语句搞定的！
这就是我们要特别用一次连载来讲使用第三方类来进行串
```

口通信的原因。实际上，笔者在进行网络通信程序编程时，也不认为MS提供的CSocket类是最方便的选择，照样习惯使用第三方的网络通信类。它们的确有非常简洁明快的接口，这一点也是值得MS哥哥们学习的。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