

Designing Series-communication

用 VC 开发 Windows 下的串口异步通信程序

Program with Visual C++6 in Windows

摘要: 本文讨论串行异步通信有关的技术及 Windows 环境下利用 VC(Visual C++)

进行串行异步通信程序的设计方法。

关键词: 串行异步通信 Windows VC

刘春立 (河北沧州师范专科学校 061001)

在工业过程检测和控制系统中, 作为上位机的微机一般都要通过串口与作为下位机的单片机系统、STD 系统或其他微机系统之间传递数据和命令信息。在以往的 DOS 环境下, 该功能可以通过编写串口中断程序来实现。而在 Windows 环境下, 系统完全接管了各种硬件资源, 不允许用户直接控制串口的中断 (以便实现资源共享), 程序员可以通过各种 API (应用程序接口) 函数与串口打交道。但数目繁多的 API 函数和复杂的使用方法往往使得许多程序员望而却步。Visual C++ 提供的 MSComm 控件 (Microsoft Comm Control) 帮助我们解决了这一难题。本文就相关的内容作些介绍。

1 串行口的控制方法

MSComm 控件提供两种控制串口进行串行异步通信的方法, 类似于 DOS 中的查询和中断。对于较简单的通信任务, 可以通过查询串口的 CommEvent 属性来了解最近发生的时间或错误并进行相应处理。对于较复杂的通信任务, MSComm 提供了一种事件驱动的方法, 通过 SetCommEvent() 函数设置串口要响应的事件, 则当响应事件或串口错误发生时, 串口会激发一个 OnComm() 事件。在 OnComm() 函数中添加相应的处理代码, 则可以实现类似于 DOS 中断的串口处理。

2 程序设计

本文将结合一个具体的实例说明如何在程序中使用 MSComm 控件, 如何利用它来设置串口参数, 以及如何控制串口的输入输出。为简单起见, 本程序采用查询方式, 只向下位机发送 5 个字节的召唤信息: "Hello", 然后等待下位机回送 8 个字节的应答信息并显示。对于复杂的通信协议, 其实现原理是一样的。

2.1 在程序中嵌入 MSComm 控件

启动 Visual C++6.0 利用 MFC App Wizard 新建一个项目文件 (Project Workspace) 并命名为 Commport, 在 AppWizard 第一步选择应用程序类型为基于对话框的 (dialog based), 在第二步将 ActiveX Controls 复选框选中, 表示本程序要求支持 ActiveX 控件。其他各步均可接受缺省设置, MFC AppWizard 将自动生成以下主要源文件: Commport.cpp, Commport.rc, CommportDlg.cpp 和 StdAfx.cpp。编译、连接自动生成的文件, 即可得到一个能运行的简单程序。程序以一个对话框为其主窗口, 该主窗口是 MFC AppWizard 自动生成的一个缺省窗口。下面我们将改造此窗口, 并在其中插入 MSComm 控件。

首先, 将 MFC App Wizard 生成的对话框中的默认控件全部删除, 然后添加两个按钮、一个编辑框和一个 MSComm 控件。添加 MSComm 控件的步骤如下:

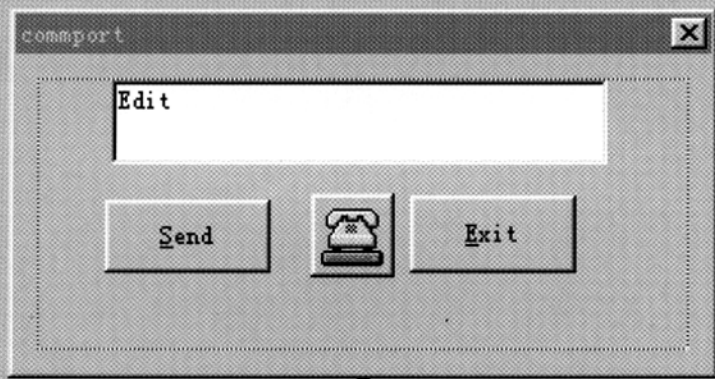


图 1

表 1 添加控件表

控件名称	控件 ID	Caption
按钮 (btnSend)	IDC_BUTTON1	&Send
按钮 (btnExit)	IDC_BUTTON2	E&xit
编辑框 (Edit)	IDC_EDIT1	
串口 (MSComm)	IDC_MSCOMM1	

选择Project 菜单下Add To Project 子菜单下的Components and Controls...选项, 在出现的对话框中选择 Registered ActiveX Controls 项双击, 在该列表框中列出的所有 ActiveX 控件中, 选择 Microsoft Communications Control version 6.0 (即 MSComm), 单击 Insert 按钮, 关闭该列表框。单击控件工具栏中的MSComm控件, 将其放入到对话框中, 控件的名称、ID和Caption 如表 1 所示。对话框程序的界面如图 1 所示。

2.2 与 MSComm 控件相关的属性及访问

在程序中嵌入 MSComm 控件后, 还要对其属性进行设置, 表 2 列出了一些经常用到的属性: (表 2 在下页上部)

2.3 在程序中使用 MSComm 控件

通过 Class Wizard, 为编辑框 IDC_EDIT1 添加变量 m_EDIT1 (类型为 CString), 用以记录编辑框内容; 为串口控件 IDC_MSCOMM1 添加变量 m_Com1 (类型为 CMSComm), 后面对串口的各种操作都可通过操作 m_Com1 来实现。

选择按钮控件 IDC_DBUTTON1, 再选择 BN_CLICK 消息, 为 Send 按钮添加处理单击事件的函数 OnButton1(), 在其中编写代码如下 (黑体部分为新增的代码, 其余代码由 Class Wizard 自动生成)。

```
Void CcommuDlg::OnSendbutton()
{
//TODO:Add your control notification handler code here

    BOOL bResponse=FALSE;
```

```
    if(!m_Com1.GetPortOpen())
        m_Com1.SetPortOpen(TRUE); // 打开串口
    m_Com1.SetSettings("1200,E,8,1"); // 设置参数
    m_Com1.SetOutput(COleVariant("Hello")); // 发送 "Hello"
    CTime StartCTime=CTime::GetCurrentTime();
    CTimeSpan ElapsedTime;

    Do
    {
        if(m_Com1.GetInBufferCount()==6) // 收到 6 个字节?
        {
            m_Com1.SetInputLen(0); // 读取全部数据
            m_Edit1=m_Com1.GetInput().bstrVal; //
            UpdateData(FALSE); // 更新编辑框内容
            bResponse=TRUE;
        }

        ElapsedTime=CTime::GetCurrentTime()-StartTime;
    }while(ElapsedTime.GetTotalSeconds()<3); // 等待 3 秒钟

    if(!bResponse) // 超时无响应
        MessageBox("Time Over!");
}
```

现在, 我们已完成了上位机的编程任务。读者可以自己编写下位机中的应答程序, 并与上面的通信程序进行通信实验。

2.4 其他应注意的问题

程序中为了控制串口, 在对话框中插入了一个 MSComm 控件并定义了一个变量 m_Com1。若要同时控制多个串口, 需要插入多个 MSComm 控件并分别为其定义变量。对不同串口的操作就通过操作不同的串口变量来实现。当然也可以为每一个控件编写 OnComm() 函数 (本程序使用查询方式, 没有添加此函数), 每当某个串口事件发生时, 系统会调用相应的 OnComm() 函数, 实现完全独立的串口控制。

利用 Visual C++ 提供的 MSComm 控件, 可以非常简单地编写 Windows 下的串行通信程序, 这为大量基于 C 语言的微机监控系统软件从 DOS 转向 Windows 提供了一条方便的途径。■

参考文献

- 1 林俊杰, 《Visual C++6 程序设计经典》, 科学出版社, 1999.11.
- 2 李朝青, 《PC 机及单片机数据通信技术》, 北京航空航天大学出版社, 2000.12.