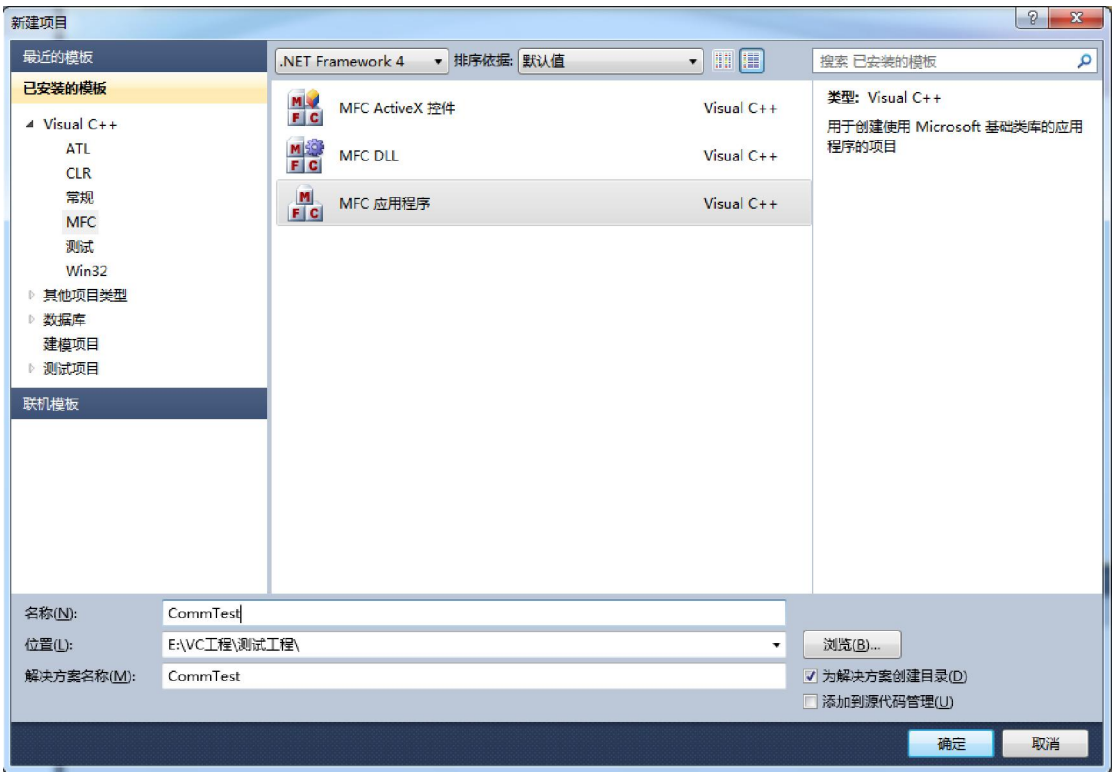
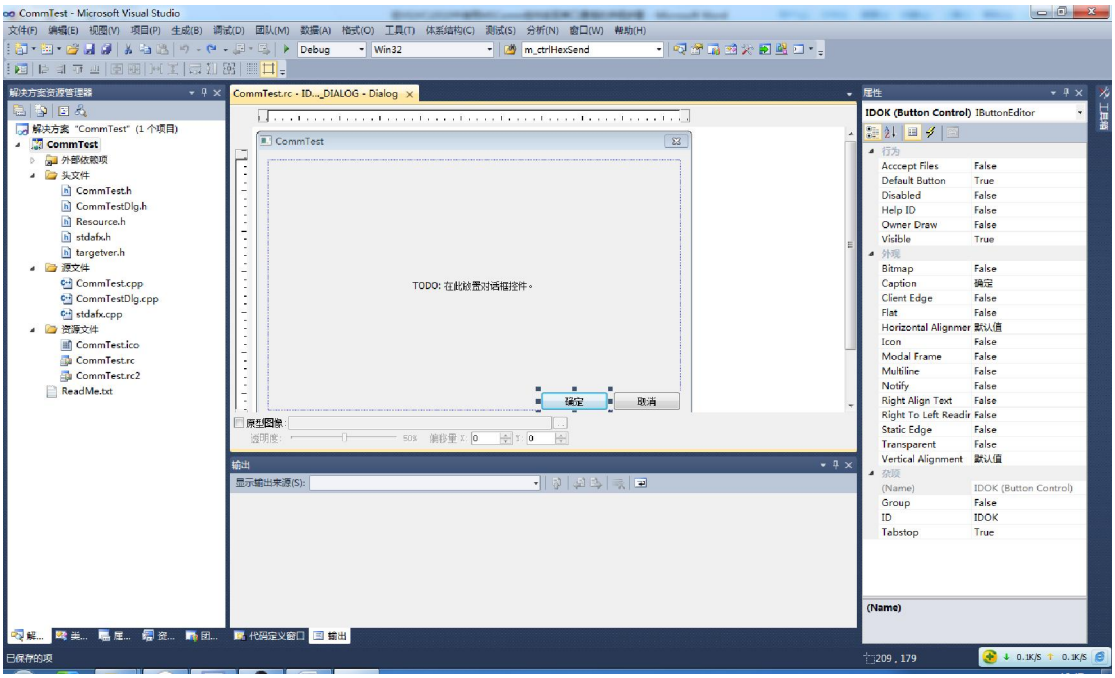
1. 安装好 VS2010，网上很多人说使用 VC6.0 的 mscomm32.ocx 控件，下载并注册，注册过程看上去还很复杂。我是使用 VS2010 自带的控件，因此没有这些过程，只需要安装好 VS2010 就行了。
2. 建立“基于对话框”的 MFC 工程，命名为 CommTest，应用程序类型选择“基于对话框”




建立好的工程如下图所示。




3. 删除默认的“确定”，“取消”按钮和静态文本框“TODO:在此放置对话框控件”，添加如下对话框控件：

① “打开串口”按钮，添加方法为从右侧“工具箱”拖放一个“Button”到对话框，

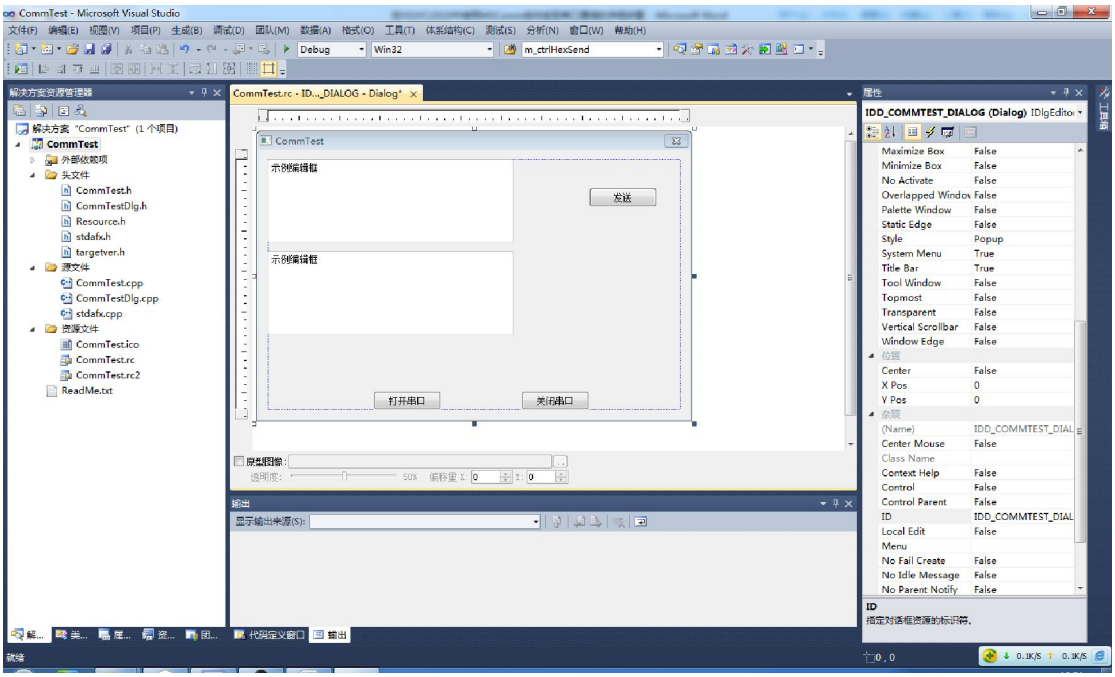
并在右侧“属性”卡中修改“Caption”为“打开串口”，修改“ID”为“IDC_BUTTON_OPEN”。

② “关闭串口”按钮，添加方法为从右侧“工具箱”拖放一个“Button”到对话框，并在右侧“属性”卡中修改“Caption”为“关闭串口”，修改“ID”为“IDC_BUTTON_CLOSE”。

③ “发送”按钮，添加方法为从右侧“工具箱”拖放一个“Button”到对话框，并在右侧“属性”卡中修改“Caption”为“发送”，修改“ID”为“IDC_BUTTON_SEND”。

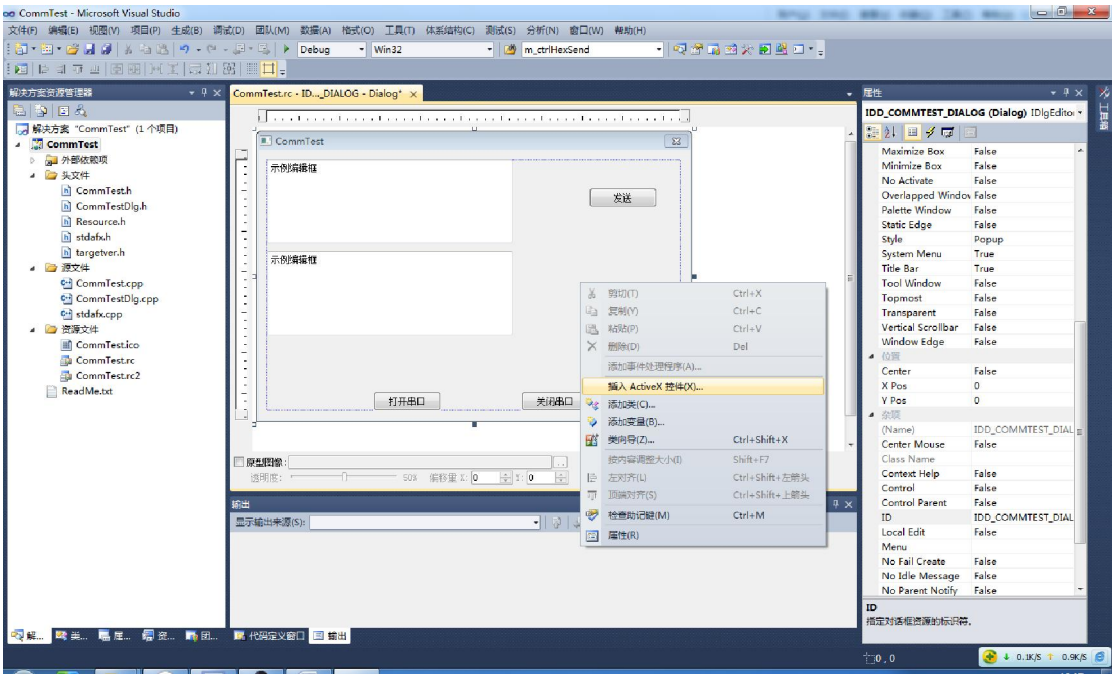
④ “发送编辑框”。

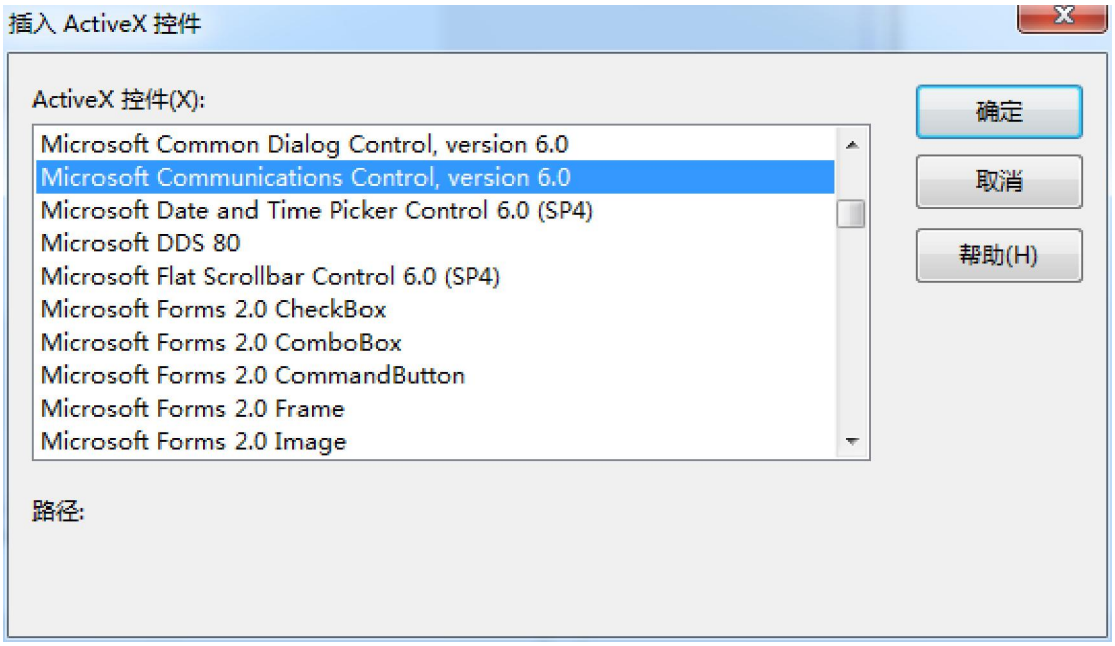
⑤ “接受编辑框”

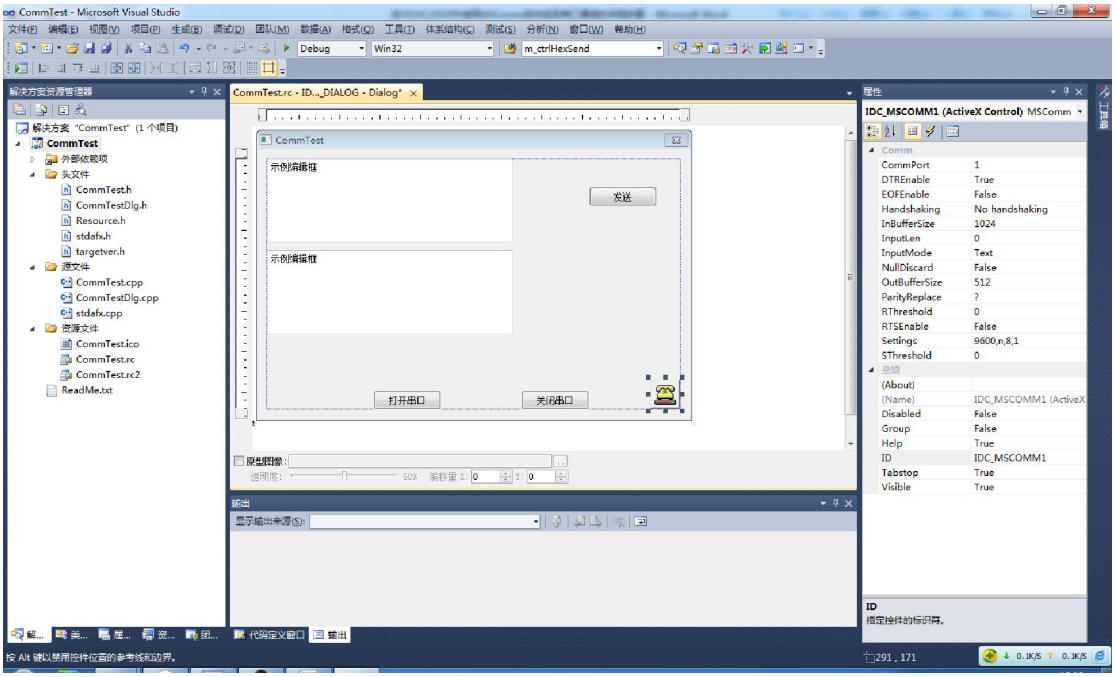



4. 添加串口通信控件。

在对话框上“右键”



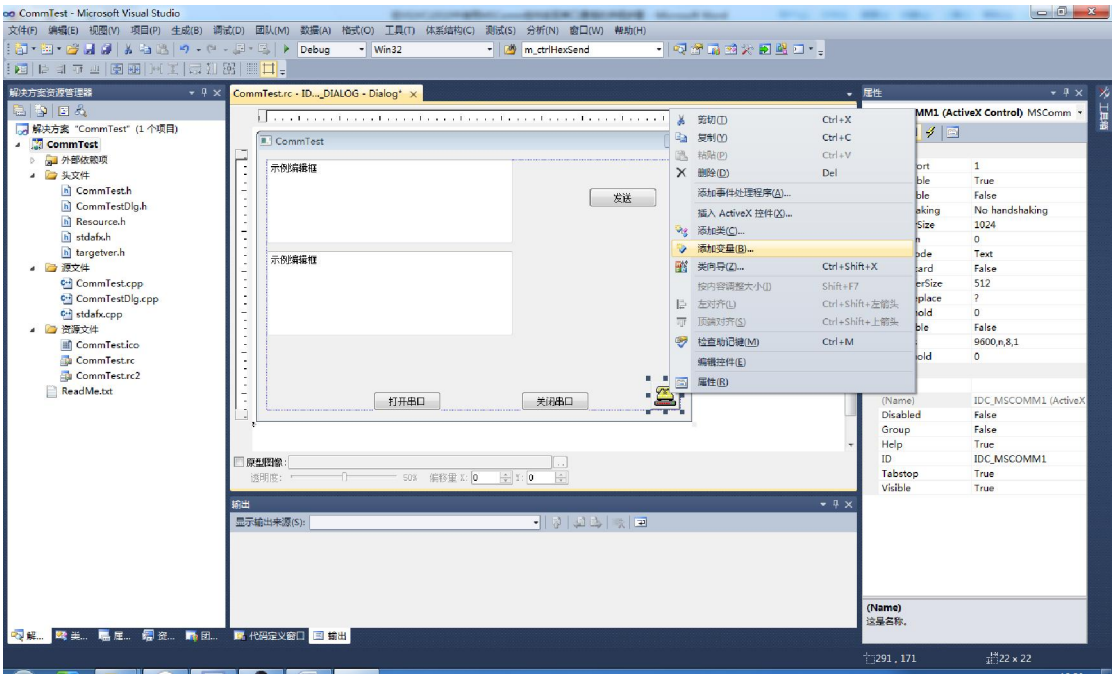


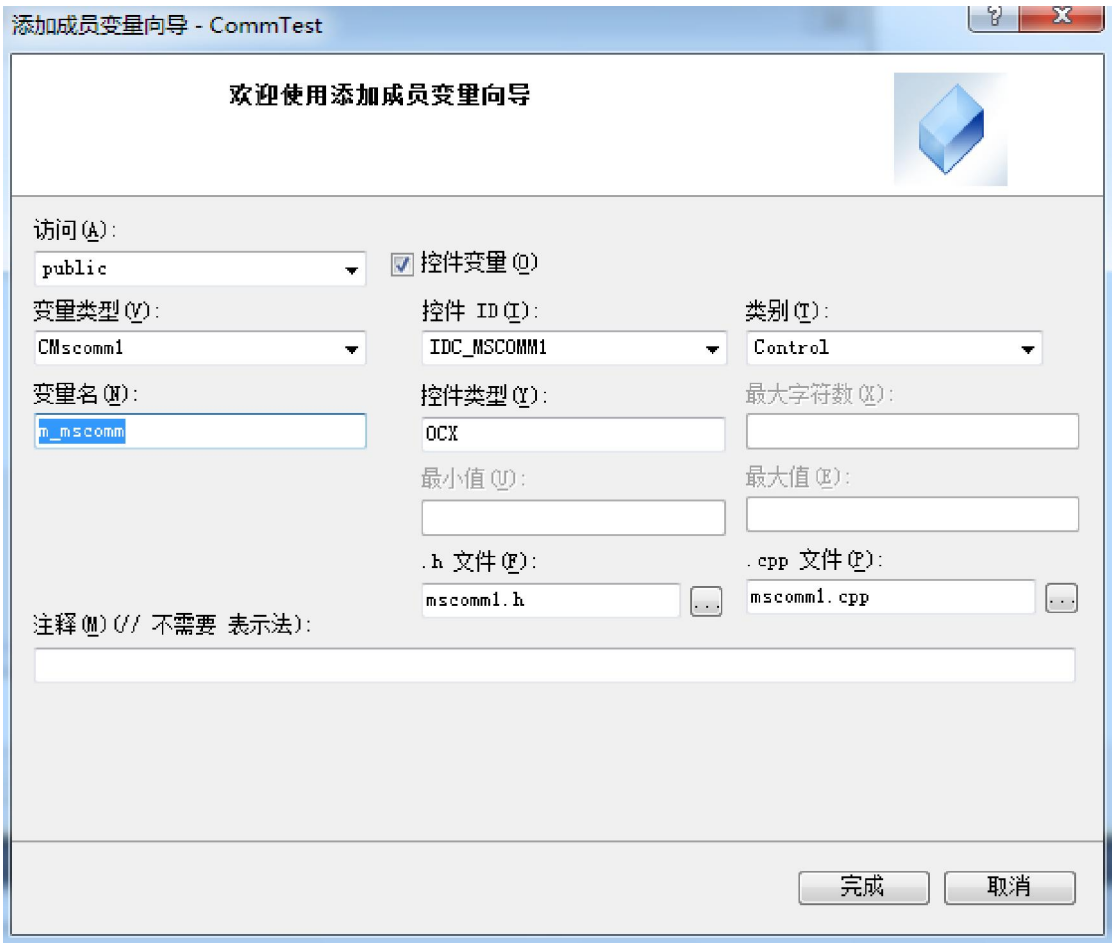




对话框右下角（默认在左上角）的电话图标就是串口控件。

5. 给控件添加变量

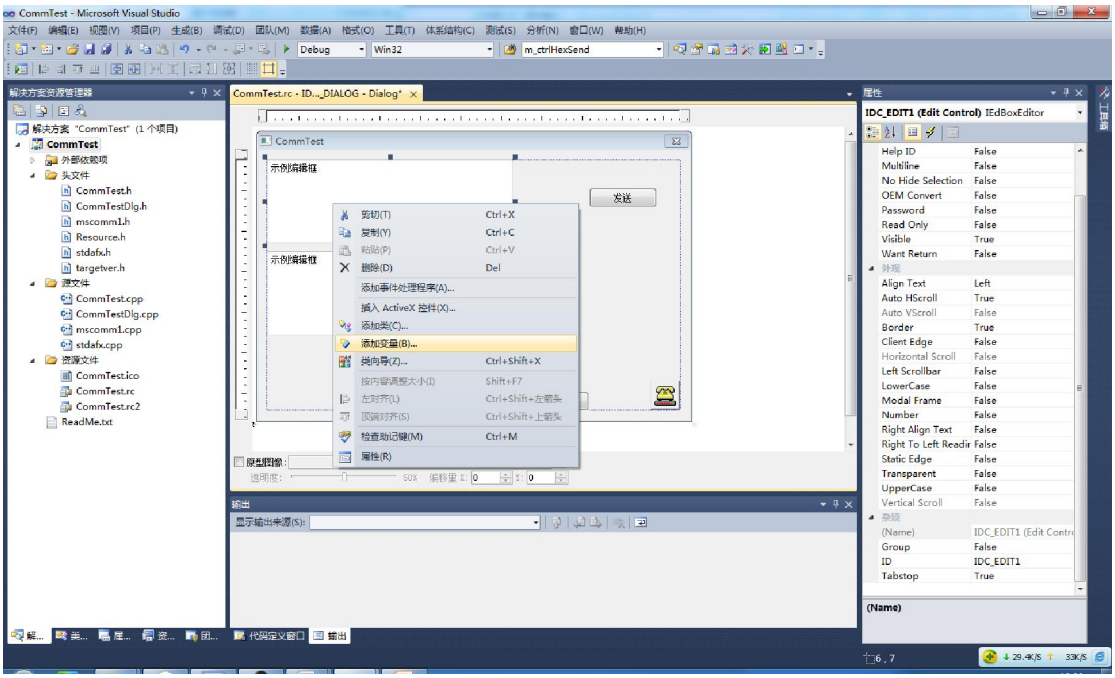
在控件(电话图标)上“右键”



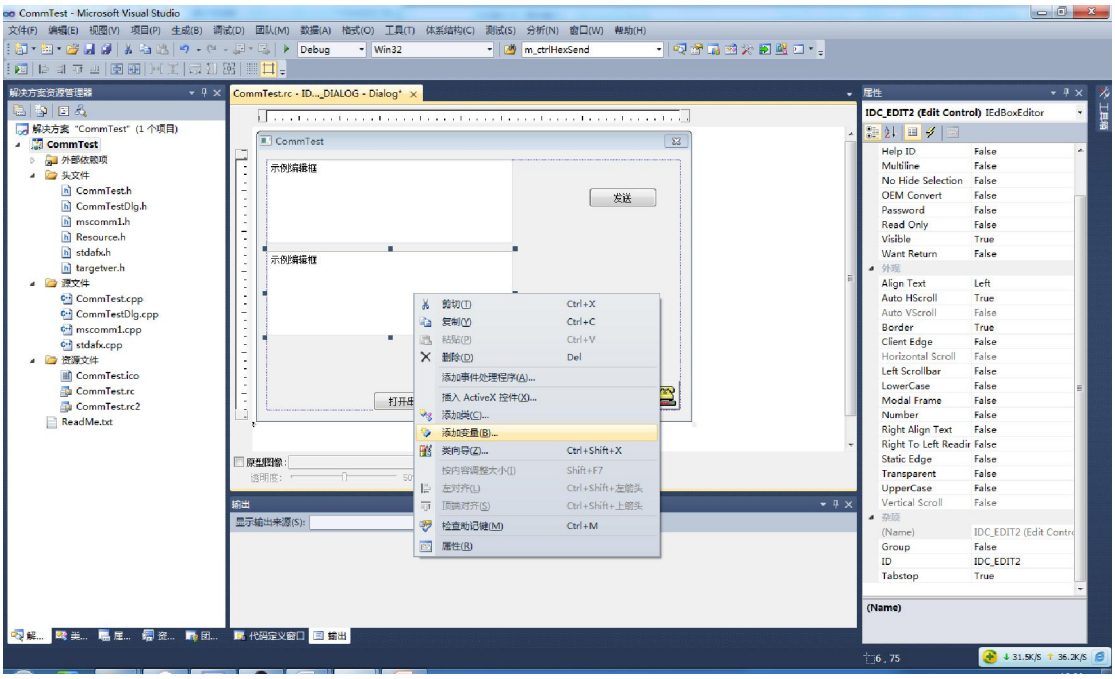




命名为 m_mscomm，点击完成，工程中会自动添加“mscomm.h”和“mscomm.cpp”两个文件。

6. 给两个编辑框添加成员变量


CommTest - Microsoft Visual Studio
CommTest.rc - ID..._DIALOG - Dialog*
CommTest
示例编辑框
发送
属性
IDC_EDIT1 (Edit Control) IEdBoxEditor


添加成员变量向导 - CommTest
欢迎使用添加成员变量向导
访问(A):
public
控件变量(O)
变量类型(V):
CString
控件 ID(I):
IDC_EDIT1
类别(T):
Value
变量名(N):
m_EditSend
控件类型(Y):
EDIT
最大字符数(X):
最小值(U):
最大值(E):
.h 文件(F):
.cpp 文件(P):
注释(M)(// 不需要 表示法):
完成
取消





7. 添加三个按钮的事件

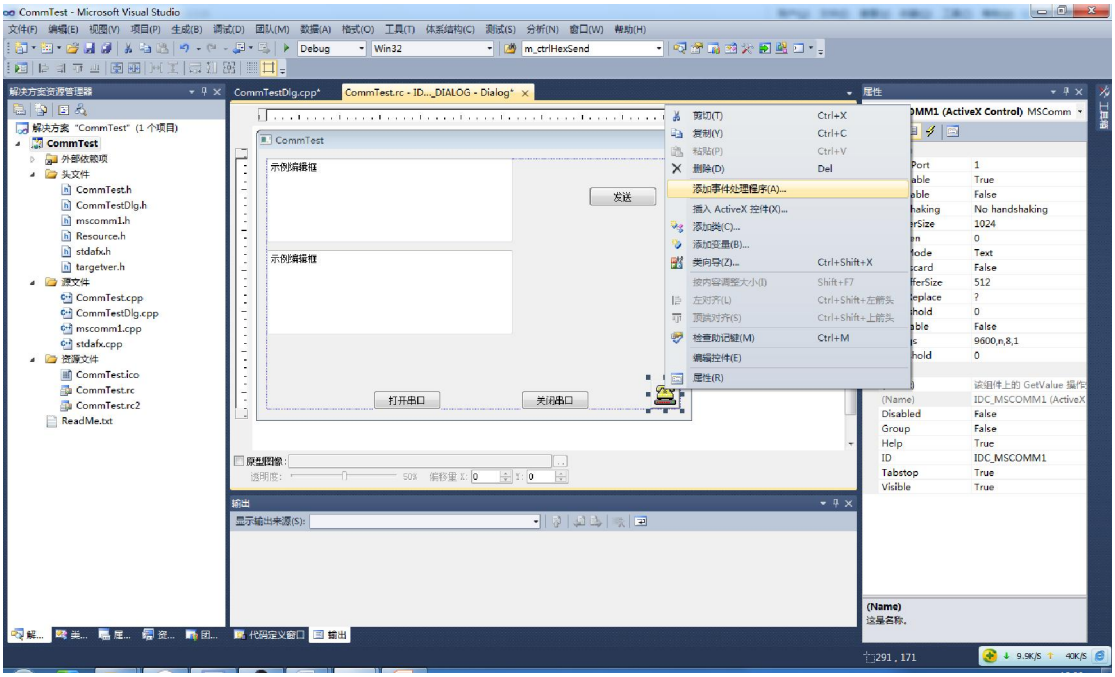
直接“双击”三个按钮，在 CommTestDlg.cpp 文件中会增加如下几个函数：

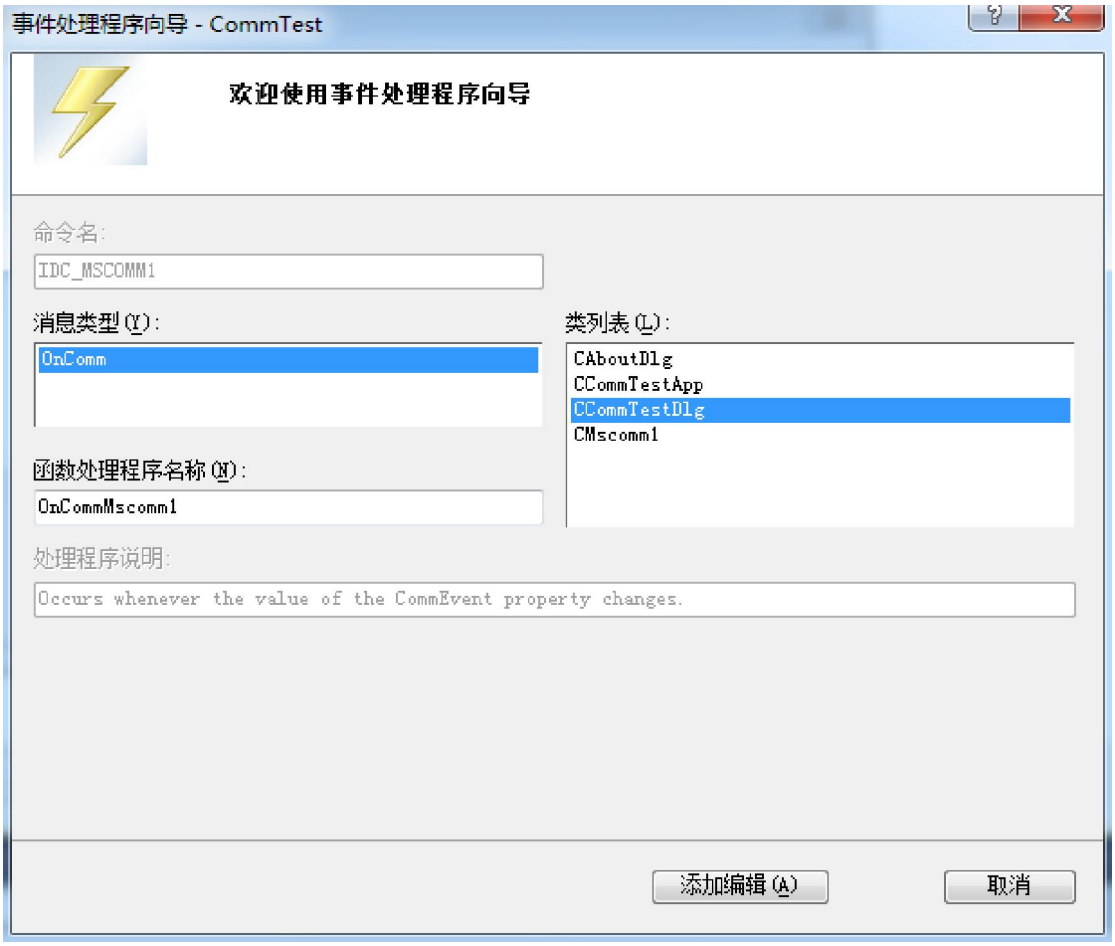
```
void CCommTestDlg::OnBnClickedButtonSend()
{
```

```
    // TODO:  在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?
}

void CCommTestDlg::OnBnClickedButtonOpen()
{
    // TODO:  在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?
}

void CCommTestDlg::OnBnClickedButtonClose()
{
    // TODO:  在¨²此ä?添¬¨ª加¨®控?件t通ª¡§知a处ä|理¤¨ª程¨¬序¨°代ä¨²码?
}
```

消息映射已经自动关联。

8. 添加串口控件的事件处理程序。






点击“添加编辑”即可，在 CommTestDlg.cpp 文件中会增加函数：

```
void CCommTestDlg::OnCommMscomm1()
{
    // TODO: 在¨²此ä?处ä|添¬¨ª加¨®消?息¡é处ä|理¤¨ª程¨¬序¨°代ä¨²码?
}
```

数据接收将在此函数中进行。

9. 打开串口及串口设置。

```
void CCommTestDlg::OnBnClickedButtonOpen()
{
// TODO: 在此添加控件通知处理程序代码
if(m_mscomm.get_PortOpen()) //如果串口是打开的，则行关闭串口
{
    m_mscomm.put_PortOpen(FALSE);
}
m_mscomm.put_CommPort(1); //选择COM1
m_mscomm.put_InBufferSize(1024); //接收缓冲区
m_mscomm.put_OutBufferSize(1024);//发送缓冲区
m_mscomm.put_InputLen(0);//设置当前接收区数据长度为0,表示全部读取
m_mscomm.put_InputMode(1);//以二进制方式读写数据
m_mscomm.put_RThreshold(1);//接收缓冲区有1个及1个以上字符时，将引发接收数据的OnComm
```

事件

```
m_mscomm.put_Settings(_T("9600,n,8,1"));//波特率9600无检验位，8个数据位，1个停止位
if(!m_mscomm.get_PortOpen())//如果串口没有打开则打开
{
    m_mscomm.put_PortOpen(TRUE);//打开串口
    AfxMessageBox(_T("串口1打开成功"));
}
else
{
    m_mscomm.put_OutBufferCount(0);
    AfxMessageBox(_T("串口1打开失败"));
}
}
```

10. 添加关闭串口按钮的消息响应函数

```
void CCommTestDlg::OnBnClickedButtonClose()
{
 // TODO: 在此添加控件通知处理程序代码
 m_mscomm.put_PortOpen(FALSE);//关闭串口
 AfxMessageBox(_T("串口 1 已关闭"));
}
```

11. 添加发送按钮消息响应函数

```
void CCommTestDlg::OnBnClickedButtonSend()
{
 // TODO: 在此添加控件通知处理程序代码
 UpdateData(true); //读取编辑框内容
 m_mscomm.put_Output(COleVariant(m_EditSend));//发送数据
 m_EditSend.Empty(); //发送后清空输入框
 UpdateData(false); //更新编辑框内容
}
```

12. 接收数据

```
void CCommTestDlg::OnCommMscomm1()
{
 // TODO: 在此处添加消息处理程序代码
 static unsigned int cnt=0;
 VARIANT variant_inp;
 COleSafeArray safearray_inp;
 long len,k;
 unsigned int data[1024]={0};
 byte rxdata[1024]; //设置 BYTE 数组
 CString strtemp;
 if(m_mscomm.get_CommEvent()==2) //值为 2 表示接收缓冲区内有字符
 {
```

```
        cnt++;
        variant_inp=m_mscomm.get_Input(); //读缓冲区消息
        safearray_inp=variant_inp; ///变量转换
        len=safearray_inp.GetOneDimSize(); //得到有效的数据长度
        for(k=0;k<len;k++)
        {
            safearray_inp.GetElement(&k,rxdata+k);
        }
        for(k=0;k<len;k++) //将数组转换为 CString 型变量
        {
            strtemp.Format(_T("%x"),*(rxdata+k));
            m_EditReceive+=strtemp;
            CString temp=_T("\r\n"); //换行
            m_EditReceive+=temp;

        }
    }
    UpdateData(FALSE); //更新编辑框内容
}
```

接收到的数据存放在 byte 型数组 rxdata[1024]中，再进行进一步处理，此处只是以 16 进制显示出来。

结束。

说明(自己的理解)：上面是基于事件驱动型的串口通信方式，感觉事件驱动与中断类似，但绝对不是中断，每一次响应事件的时候接收缓冲区的数据的长度不是固定的。例如：我的下位机发送了 10 个 BYTE 的数据，第一次上位机响应的时候只接收到 8 个，第二次响应时接收到另外两个。并不是每收到一个数据就响应一次。

下图是测试结果：
接收到的是字母‘a’，‘b’,……‘j’10 个字母，显示出来的是他们的 16 进制数。