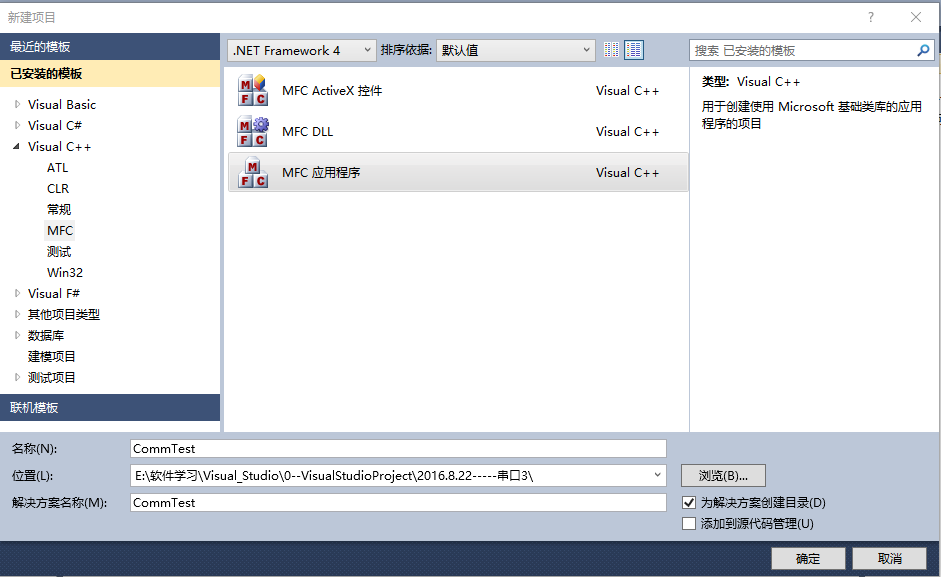
# MSComm控件使用-VisualStudio2010

## 1.准备

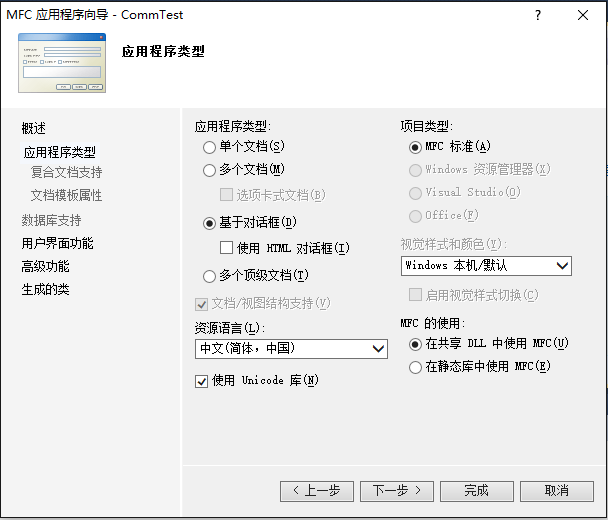
一般VS2010 自带MSComm控件,没有的话可以在网上下载。

## 2. 建立工程

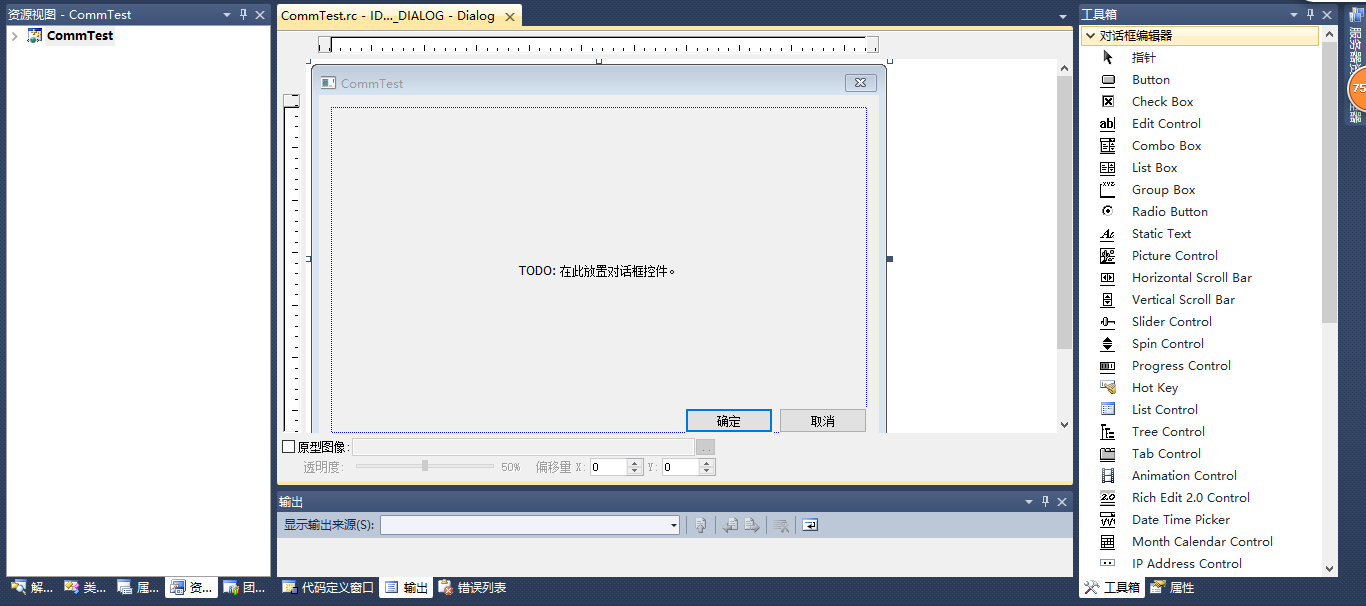
1.建立“基于对话框”的MFC 工程，命名为CommTest。



2.应用程序类型选择“基于对话框”。



3.建立好的工程如下图所示：



## 3. 编辑界面

1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”。

2添加如下控件：

(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”（ID会在后面的编程中涉及到）。

(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)“十六进制”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“十六进制”，修改“ID”为“IDC\_BUTTON\_SIXTEEN”。

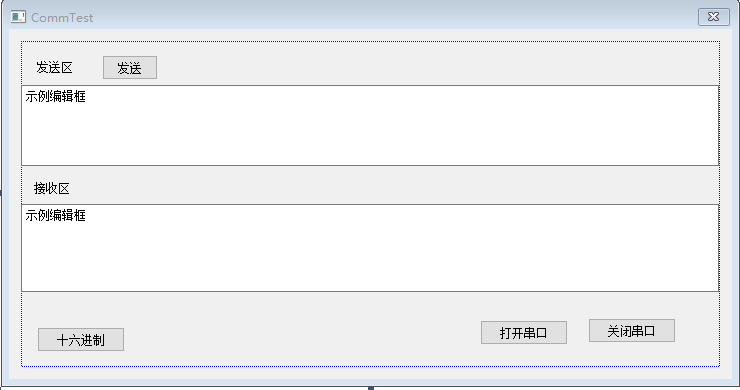
(5)“发送编辑框”，添加方法为从右侧“工具箱”拖放一个“Edit Control”到对话框。

(6)“接收编辑框”，添加方法为从右侧“工具箱”拖放一个“Edit Control”到对话框。

(7)“发送静态文本框”，添加方法为从右侧“工具箱”拖放一个“Static Text”到对话框，并在右侧“属性”卡中修改“Caption”为“发送区”。

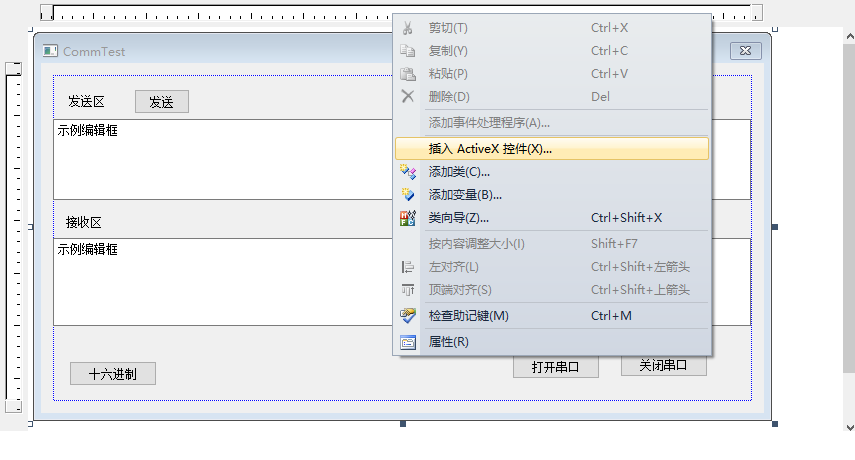
(8)“接收静态文本框”，添加方法为从右侧“工具箱”拖放一个“Static Text”到对话框，并在右侧“属性”卡中修改“Caption”为“接收区”。

编辑好的界面如下图所示：

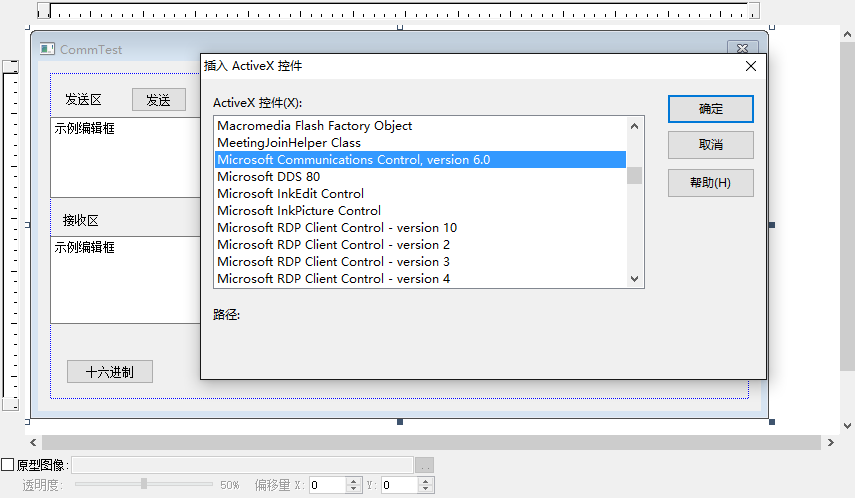


## 4. 添加串口通信控件

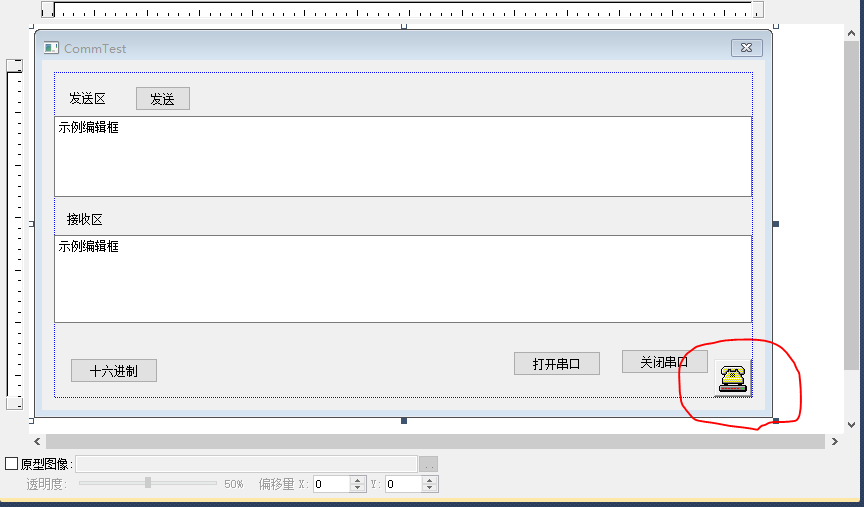
1在对话框上“右键”——>插入ActiveX控件



2选择Microsoft Communications Control,version6.0。(如果安装的Visual Studio2010没有自带需要从网上下载)

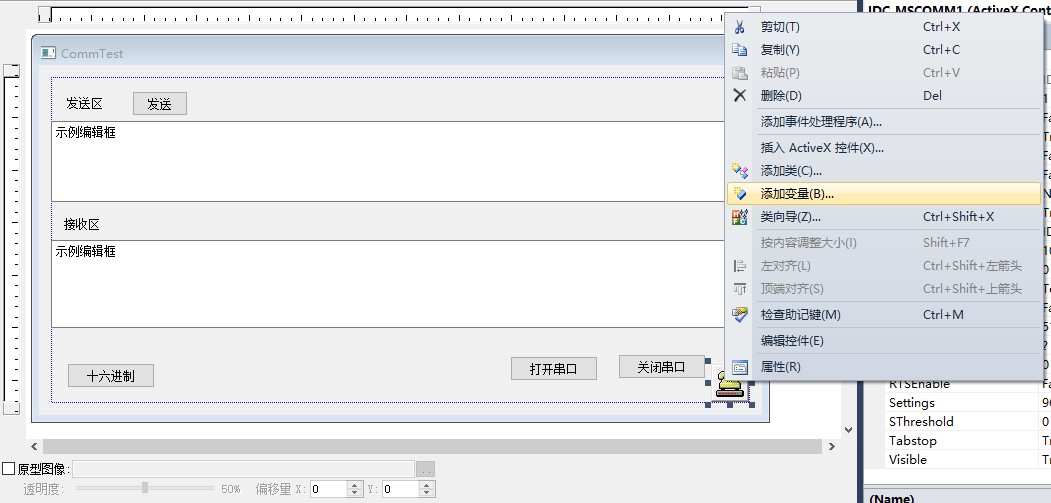


3对话框右下角（默认在左上角）的电话图标就是串口控件。

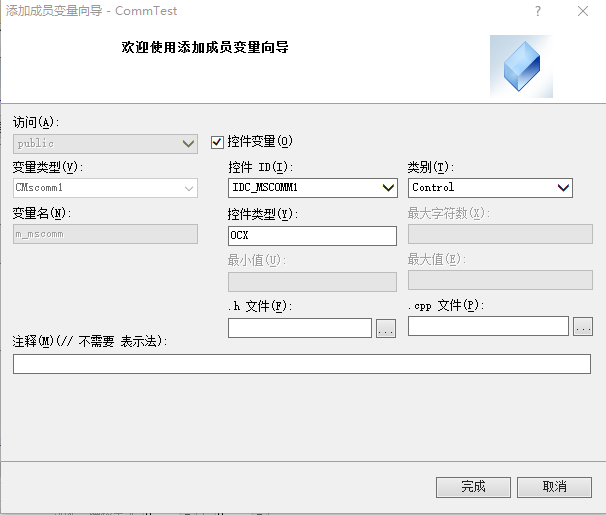


## 5. 给控件添加成员变量

1在控件(电话图标)上“右键”——>添加变量

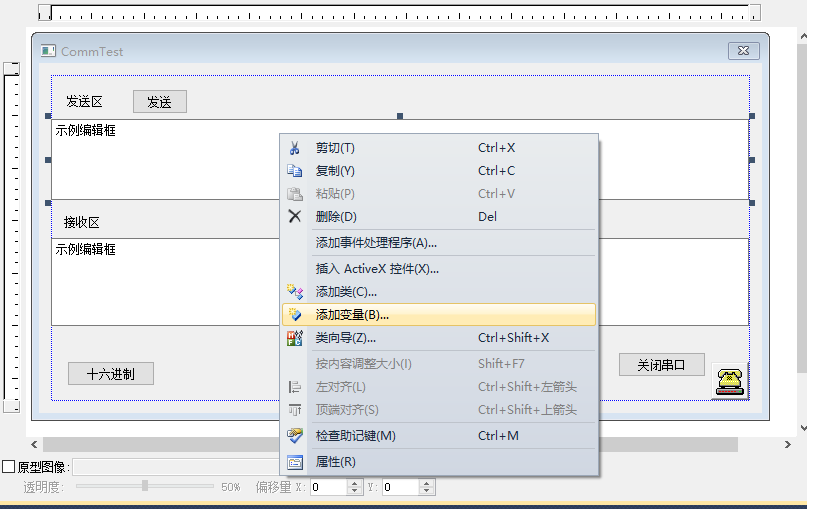


2 命名为m\_mscomm，点击完成，工程中会自动添加“mscomm1.h”和“mscomm1.cpp”两个文件。



## 6. 给编辑框添加成员变量

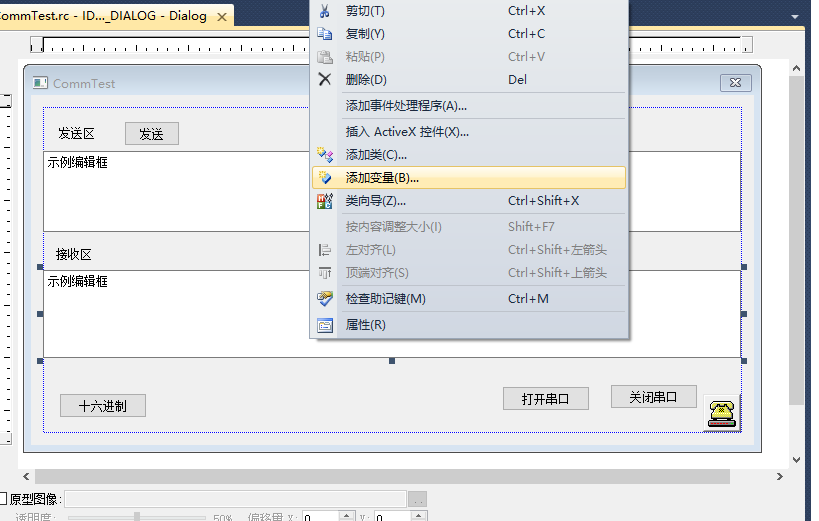
1在上面的编辑框上“右键”——>添加变量



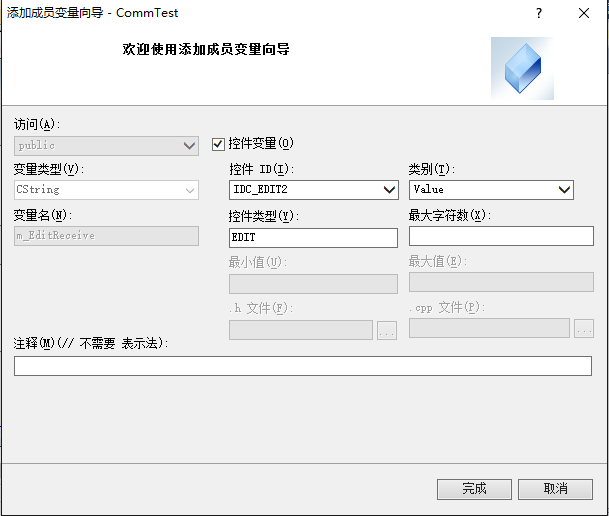
2类别选Value，变量类型选CString(选错了就编译不鸟了)，变量名为m\_EditSend。



3在下面的编辑框上“右键”——>添加变量



4类别选Value，变量类型选CString(选错了就编译不鸟了)，变量名为m\_EditReceive。



## 7. 添加三个按钮的事件

直接“双击”四个按钮，在CommTestDlg.cpp 文件中会增加如下几个函数：

void CCommTestDlg::OnBnClickedButtonSend()

{

// TODO: 在此添加控件通知处理程序代码

}

void CCommTestDlg::OnBnClickedButtonOpen()

{

// TODO: 在此添加控件通知处理程序代码

}

void CCommTestDlg::OnBnClickedButtonClose()

{

// TODO: 在此添加控件通知处理程序代码

}

void CCommTestDlg::OnBnClickedButtonSixteen()

{

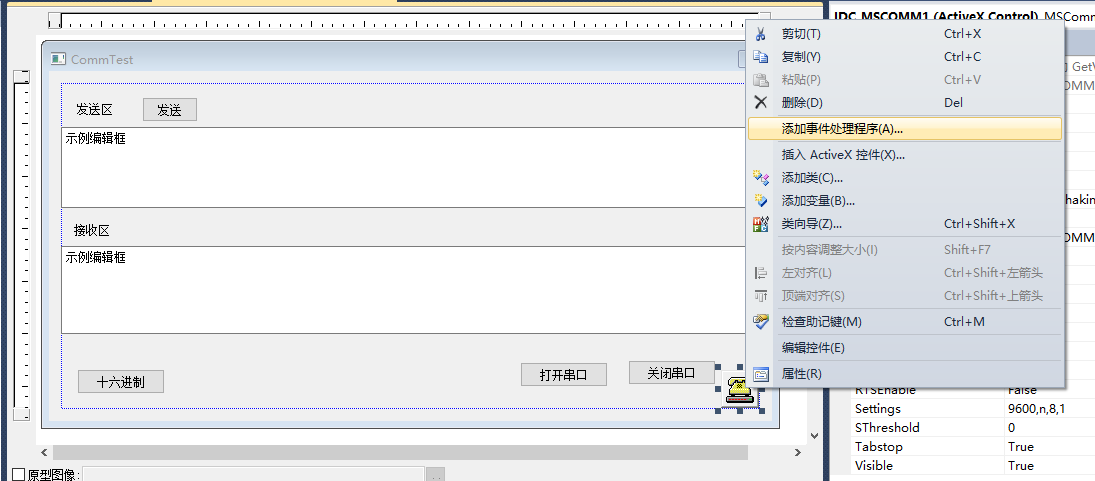
// TODO: 在此添加控件通知处理程序代码

}

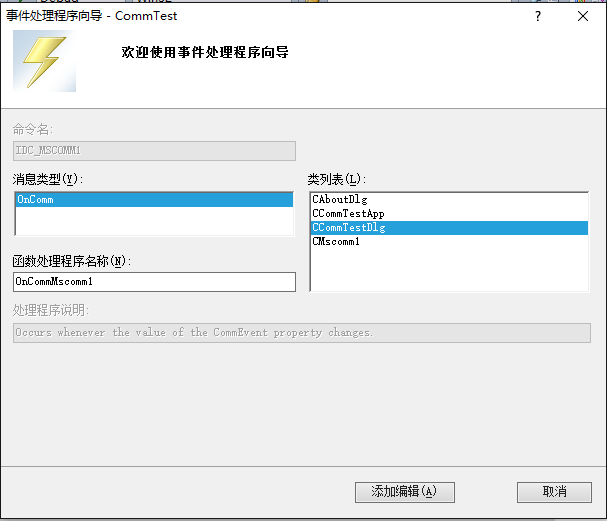
消息映射已经自动关联。

## 8. 添加串口控件的事件处理程序

1在控件(电话图标)上“右键”——>添加事件处理程序



2点击“添加编辑”



在CommTestDlg.cpp 文件中会增加函数：

BEGIN\_EVENTSINK\_MAP(CCommTestDlg, CDialogEx)

ON\_EVENT (CCommTestDlg, IDC\_MSCOMM1, 1, CCommTestDlg:: OnCommMscomm1, VTS\_NONE)

END\_EVENTSINK\_MAP()

void CCommTestDlg::OnCommMscomm1()

{

// TODO: 在此处添加消息处理程序代码

}

数据接收将在此函数中进行。

## 9. 添加函数

### 1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"));

}

}

### 2. 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"));

}

### 3. 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

}

### 4. 添加十六进制按钮消息响应函数

void CCommTestDlg::OnBnClickedButtonSixteen()

{

// TODO: 在此添加控件通知处理程序代码

CByteArray hexdata;//发送的数据

hexdata.Add(0xC1);

hexdata.Add(0x03);

hexdata.Add(0x01);

hexdata.Add(0x00);

hexdata.Add(0x00);

hexdata.Add(0x30);

m\_mscomm.put\_OutBufferCount(0);//清空发送缓冲区

m\_mscomm.put\_Output(COleVariant(hexdata));//发送十六进制数据

}

### 5. 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，待进行进一步处理，此处只是以16 进制显示出来。

## 10串口测试

1进行测试之前需要两个辅助软件，串口模拟器，串口调试助手

