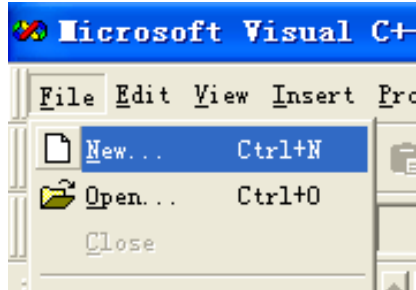


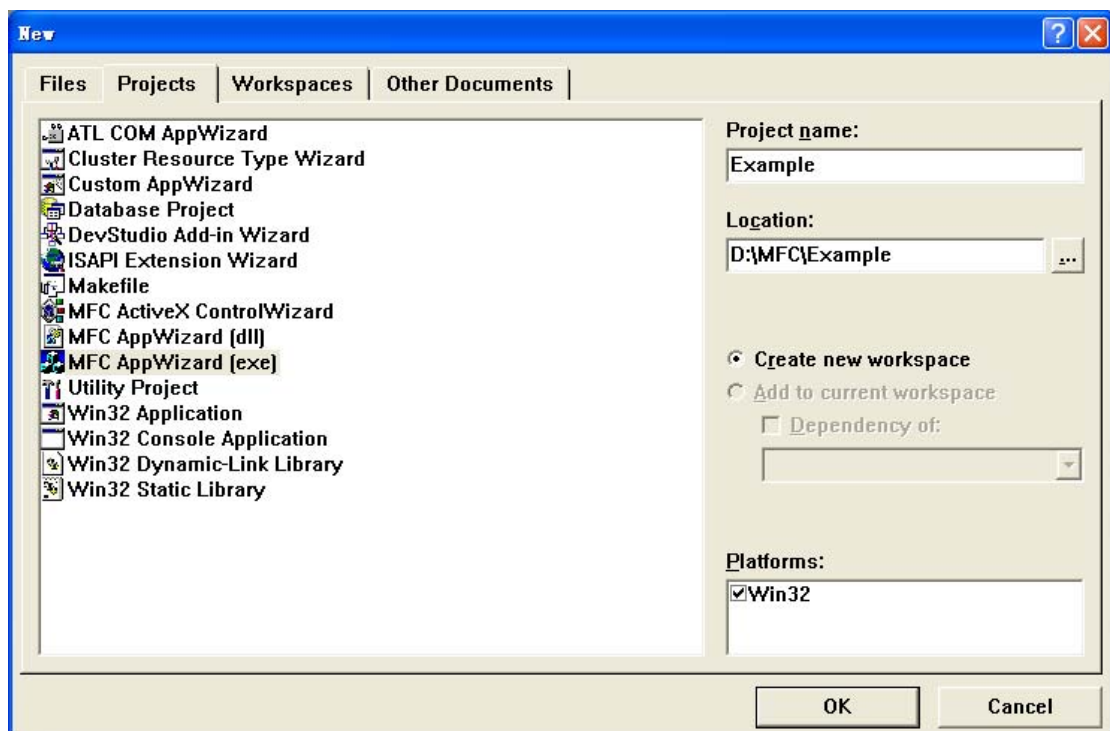
VC 串口通信步骤

创建对话框: File→New→MFC AppWizard(exe)→(在 Project name: 下写工程名(Example), 然后在 Location: 下设置保存路径)→点击 OK→选择 Dialog based→点击 Finish→OK

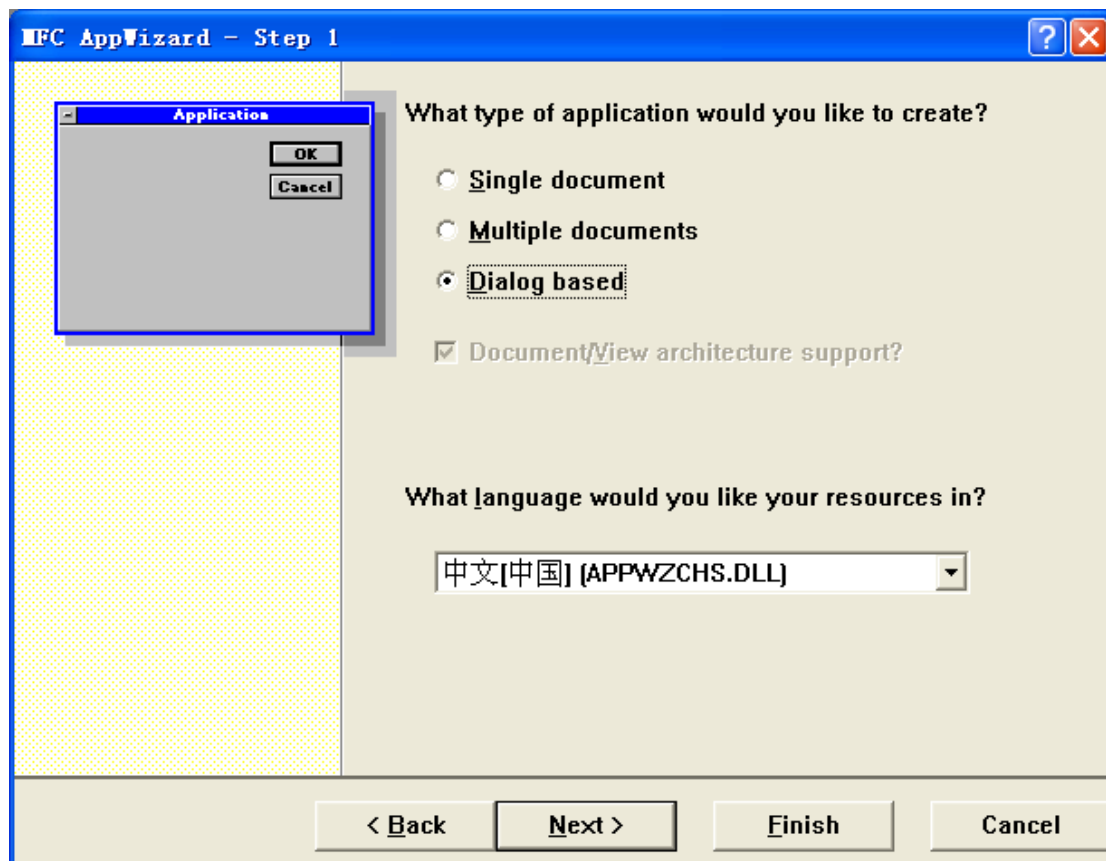
File→New



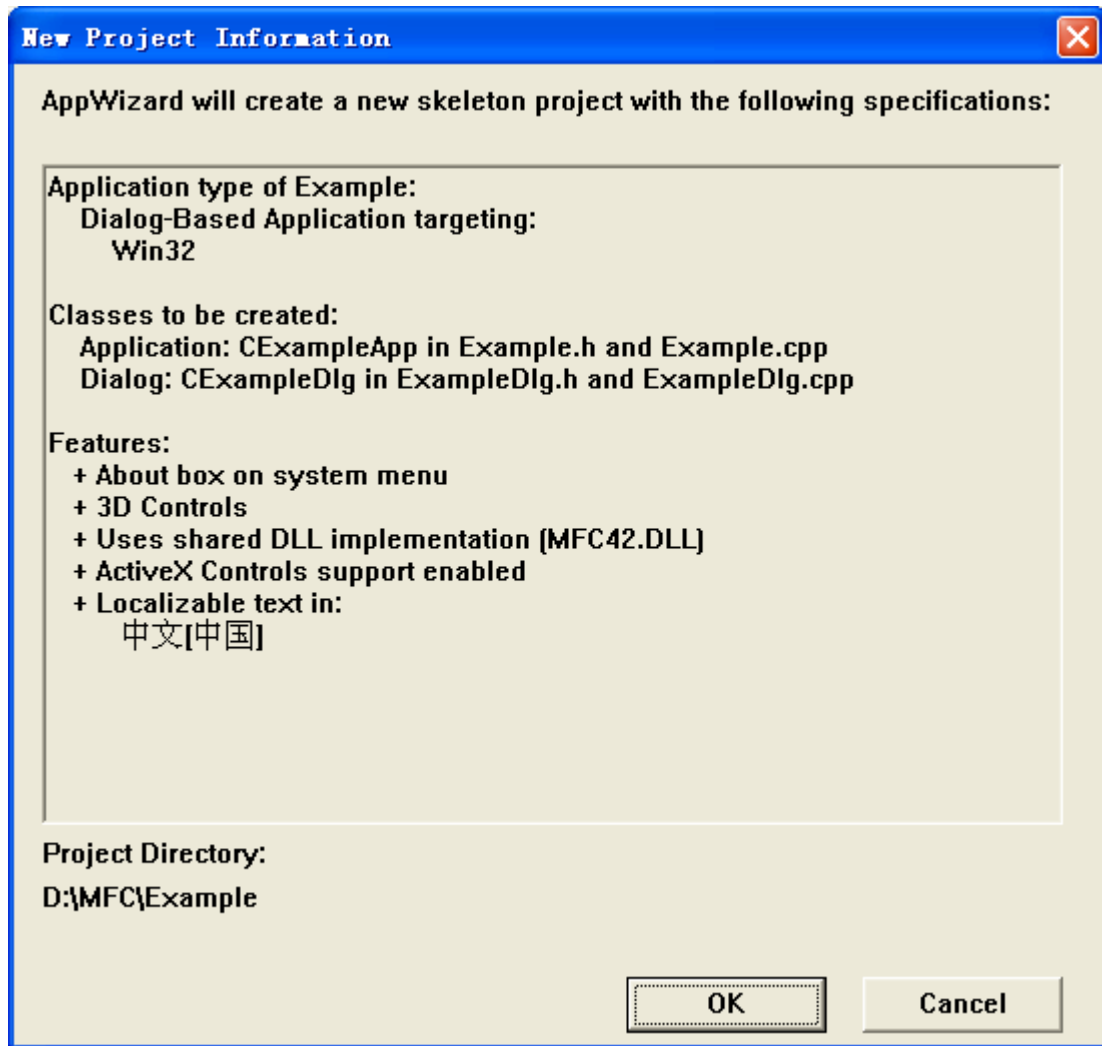
在 Project name: 下写工程名(Example), 然后在 Location: 下设置保存路径→点击 OK



选择 Dialog based→点击 Finish

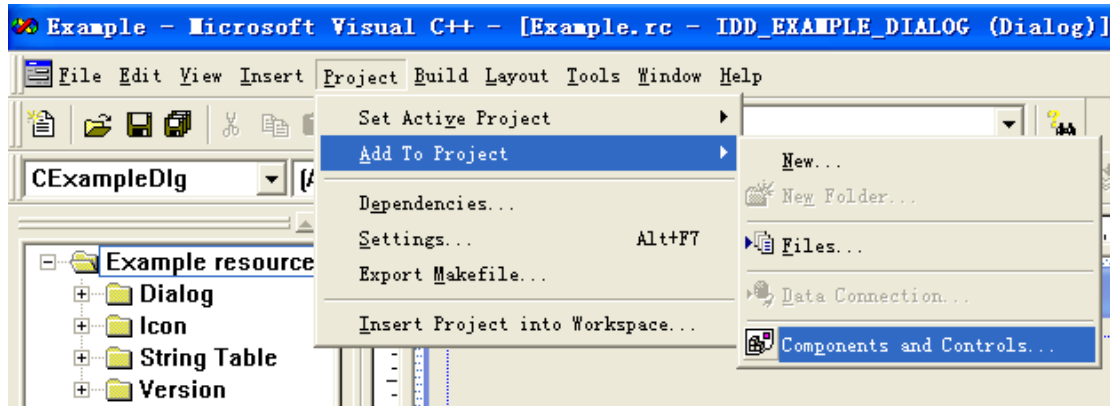


点击 OK

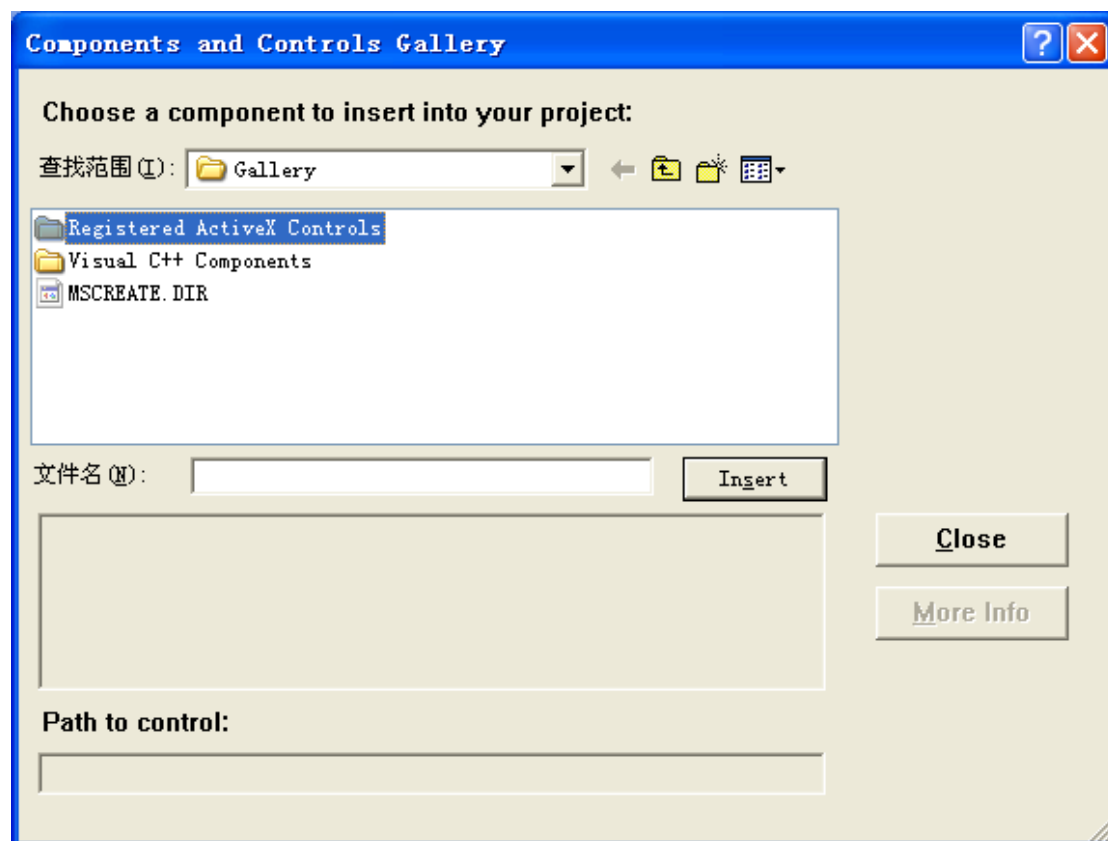


加载 MSComm 控件: Project→Add To Project→Components And Controls→单击 Registered ActiveX Controls→Insert→找到 Microsoft Communications Control,version 6.0 并单击选中它→Insert→确定→OK→Close

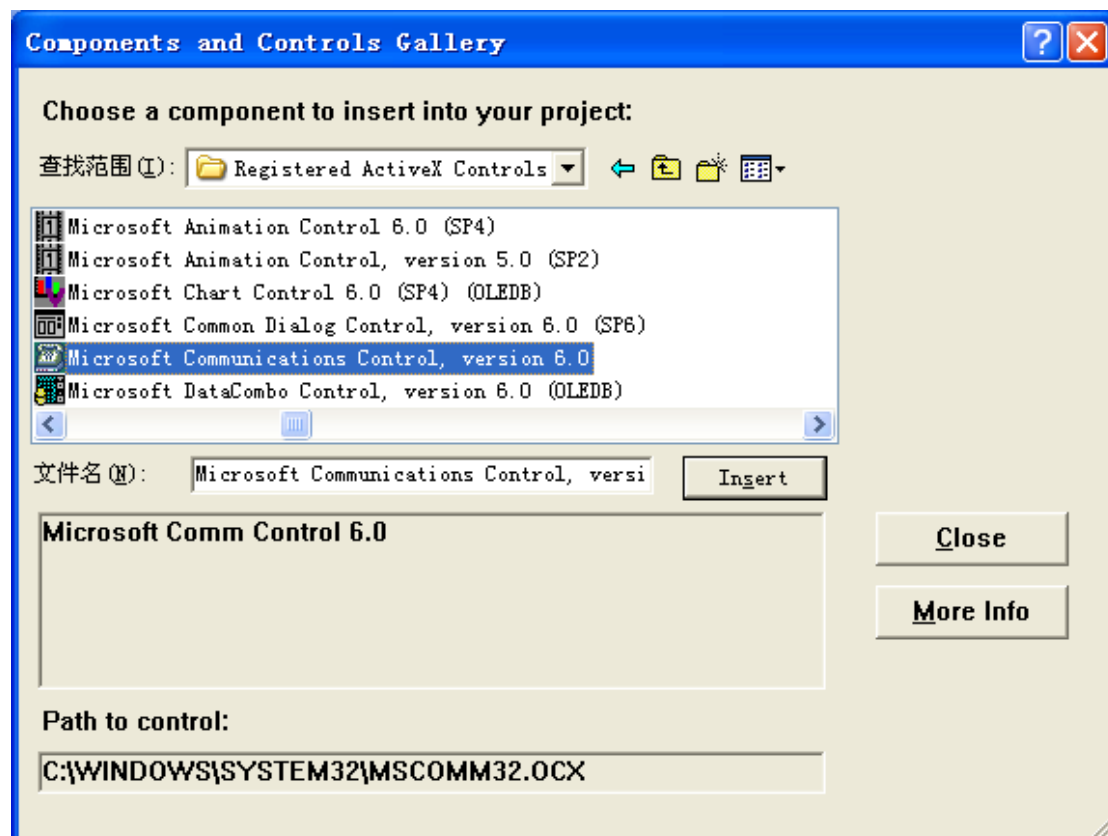
Project→Add To Project→Components And Controls



单击 Registered ActiveX Controls→Insert



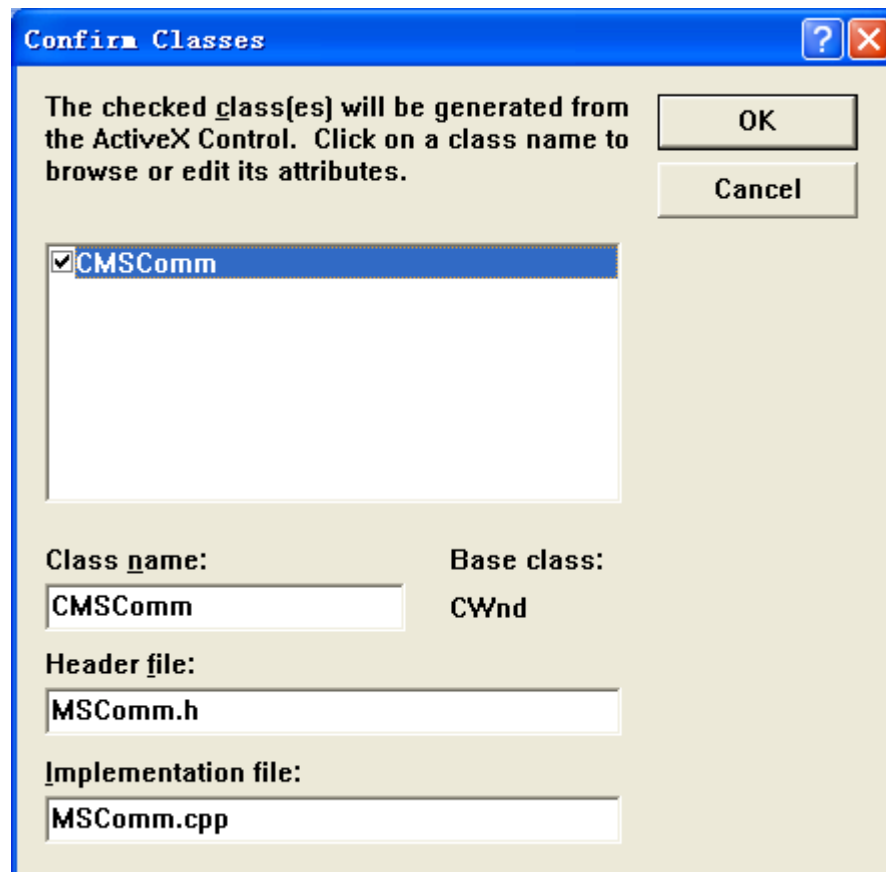
找到 Microsoft Communications Control,version 6.0 并单击选中它→Insert



选择确定



选择 OK

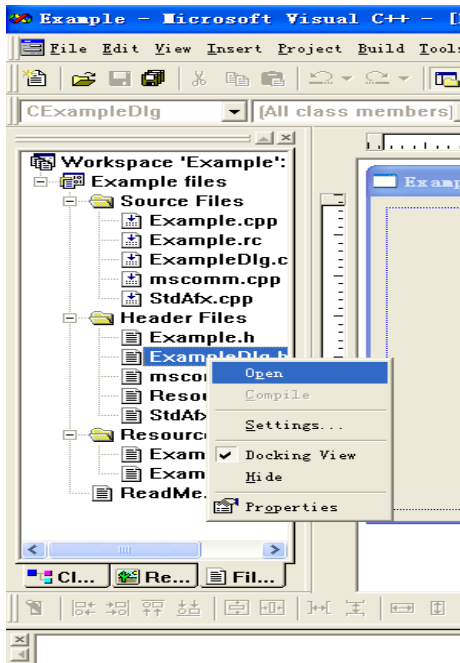


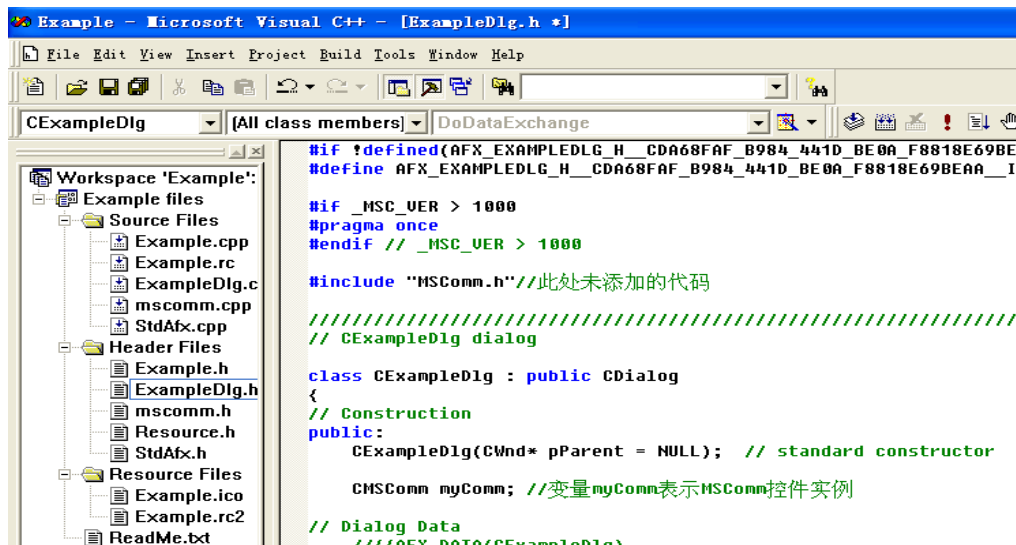


控件栏出现串口控件

包含头文件、声明 MSComm 类变量：打开(双击)Header Files 中的 ExampleDlg.h→在 class CJDlg : public CDialog 上面一行包含串口通信的头文件：`#include "MSComm.h"`
在 `CJDlg(CWnd* pParent = NULL); // standard constructor` 的下一行写(声明 MSComm 类变量)：

`MSComm myComm; //变量 myComm 表示 MSComm 控件实例`





创建 MSCOMM 实例：View→Class Wizard→Message Maps→在 Object Ids:中单击 CJDlg→在 Messages:下单击 WM_CREATE→Add Function→在 Member Functions:下双击 W OnCreate→在// TODO: Add your specialized creation code here 下面一行添加如下代码：

```
DWORD style=WS_VISIBLE|WS_CHILD;
```

```

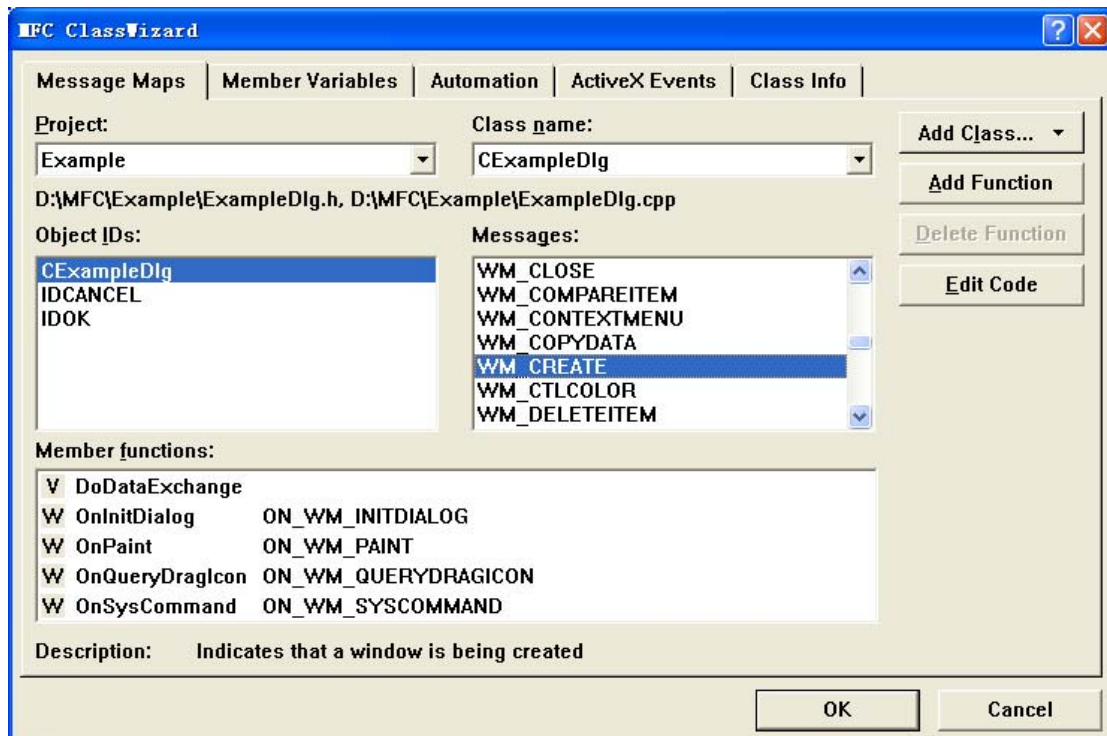
    if(!myComm.Create(NULL,style,CRect(0,0,0,0),this,IDC_MSCOMM1))
    {
        TRACE0("Failed to create OLE Communication Control!\n");
        return -1;
    }

```

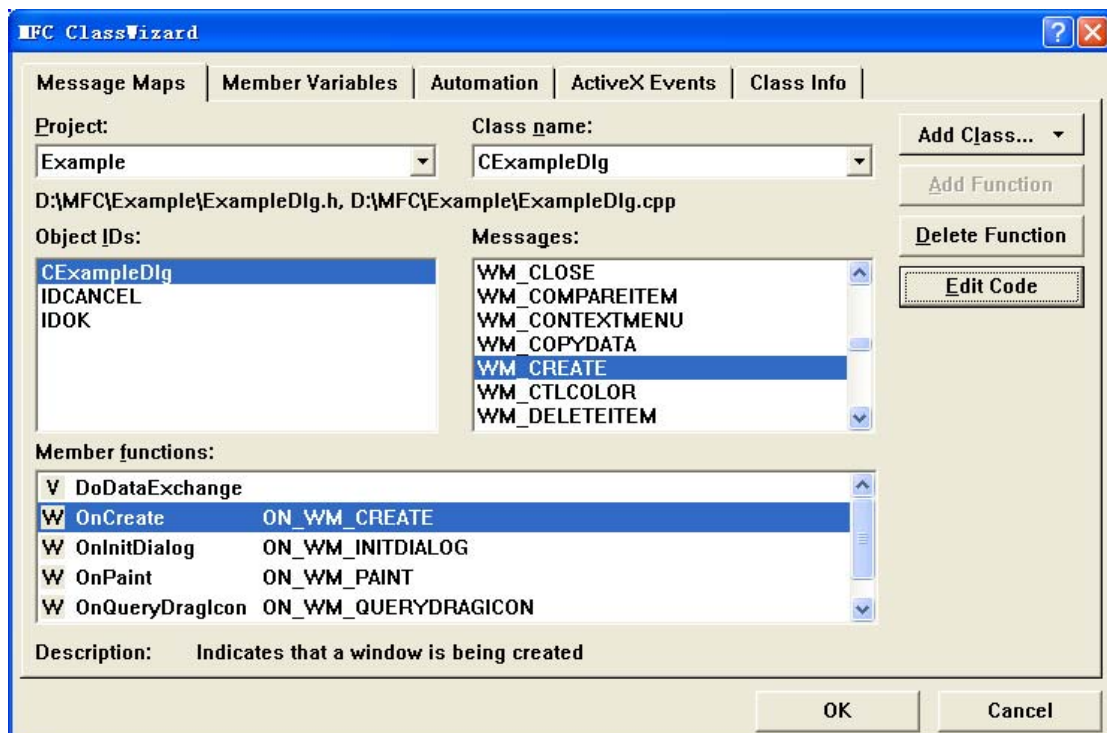
View→Class Wizard



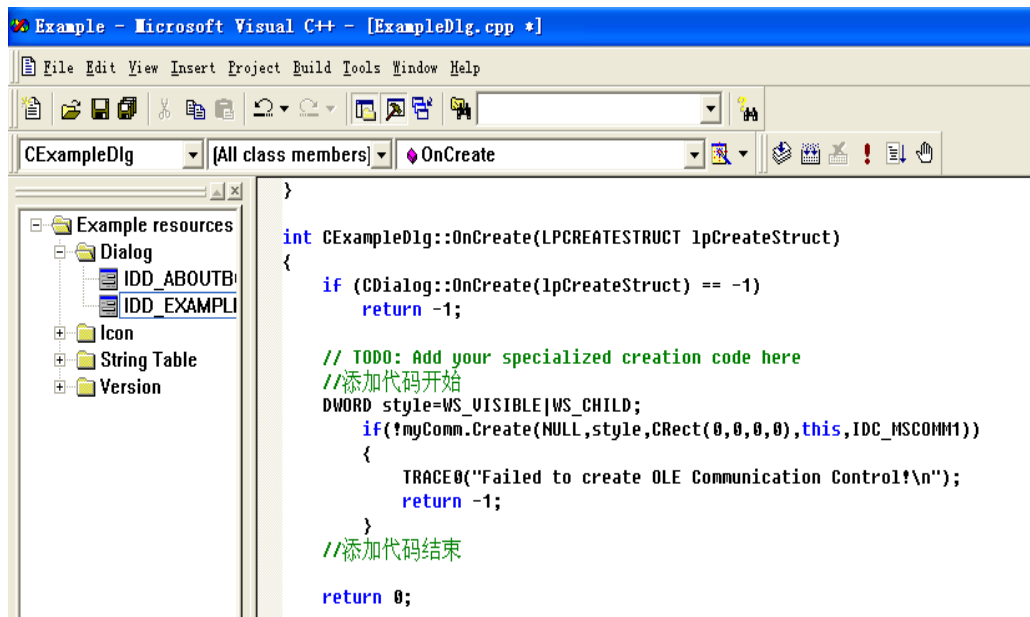
Message Maps→在 Object Ids:中单击 CJDlg→在 Messages:下单击 WM_CREATE→Add Function



在 Member Functions:下双击 W OnCreate



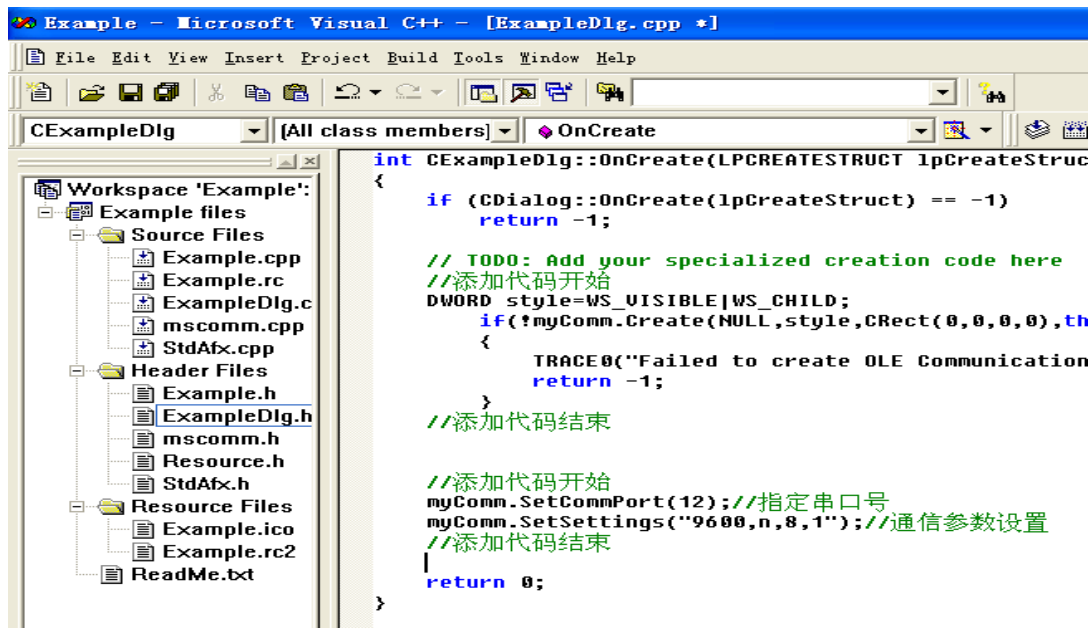
在// TODO: Add your specialized creation code here 下面一行添加代码



初始化串口：接着在上一步骤代码的后面添加如下代码：

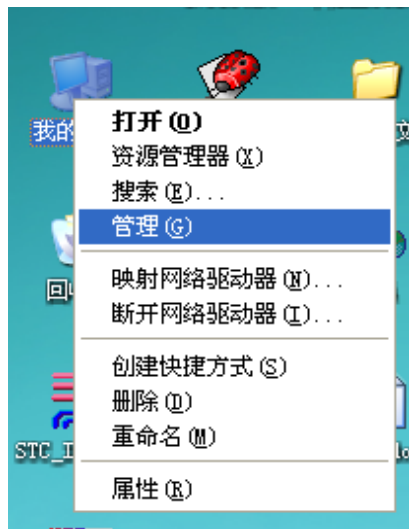
myComm.SetCommPort(12);//指定串口号（注意：根据具体情况写串口号）

myComm.SetSettings("9600,n,8,1");//通信参数设置

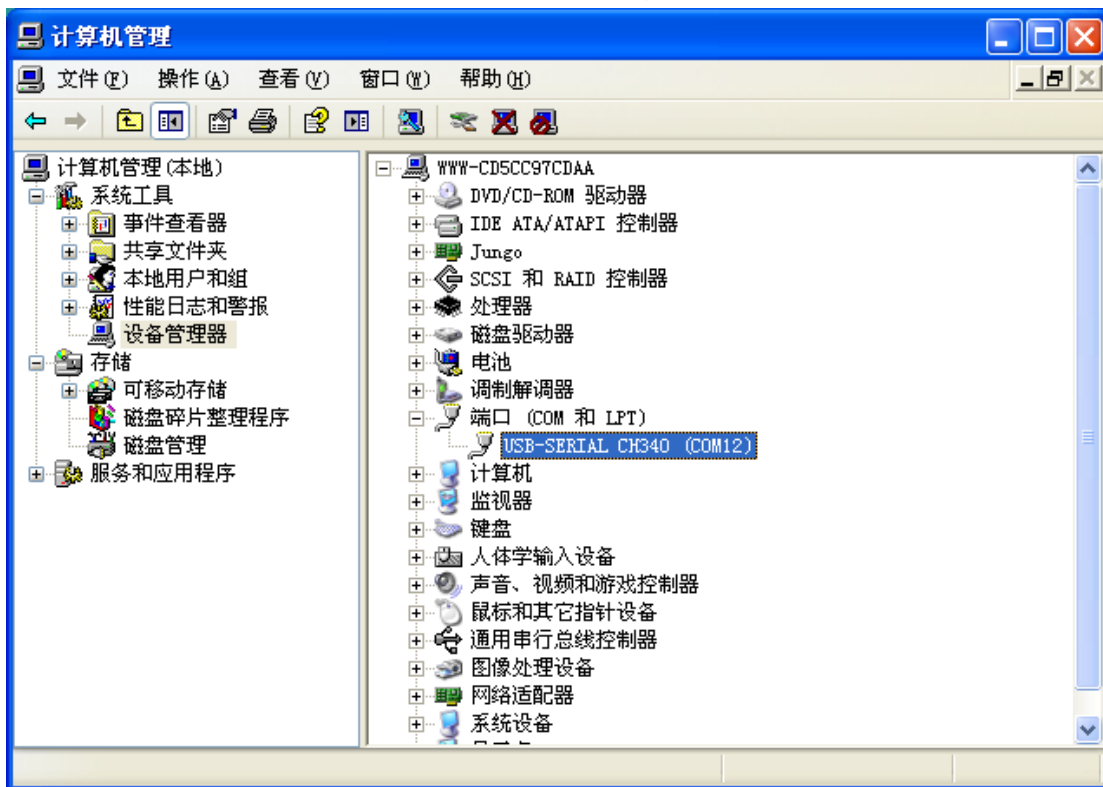


由于刚学串口通信，尚未实验动态检测配置串口，先将就一下（获取串口号）：

插上串口（USB 转串口也可的）→右键 我的电脑→管理



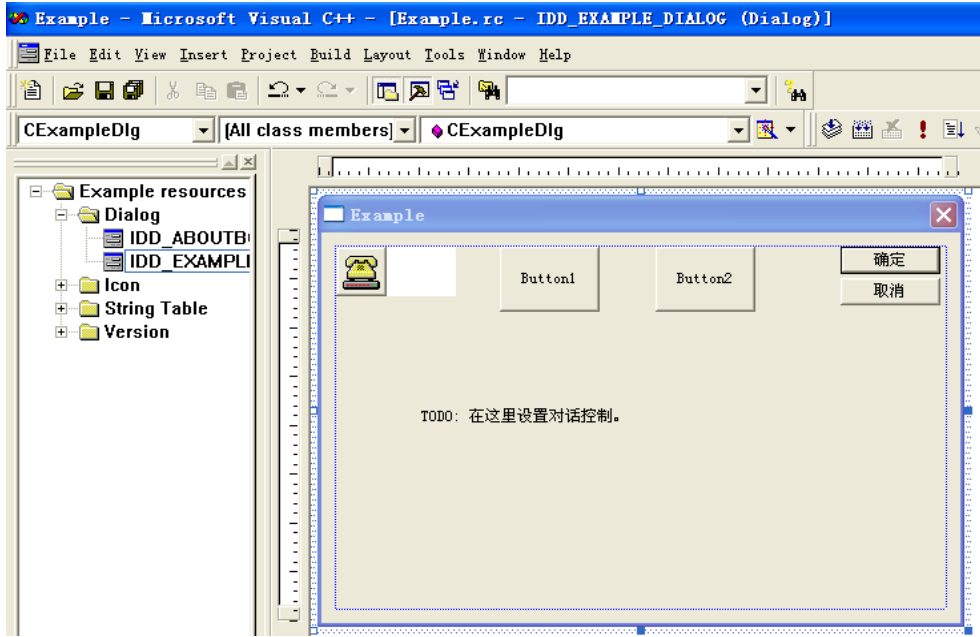
设备管理→端口（我的是 12），所以程序中写的是 myComm.SetCommPort(12);



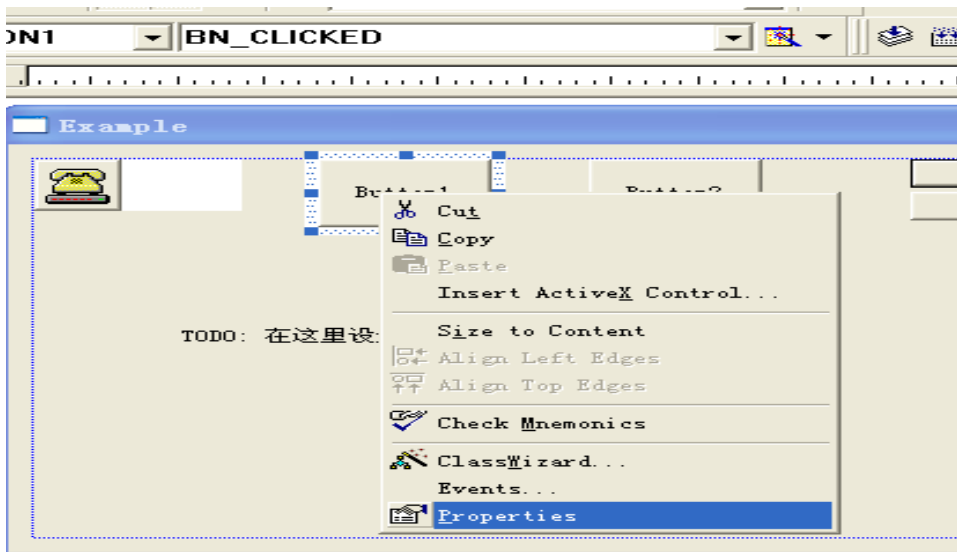
在对话框中放置两个按钮控件和一个串口通信控件，在第一个按钮上右键→Properties→在 General 中 Caption: 填写“打开串口”→关闭

在第二个按钮上右键→Properties→在 General 中 Caption: 填写“关闭串口”→关闭

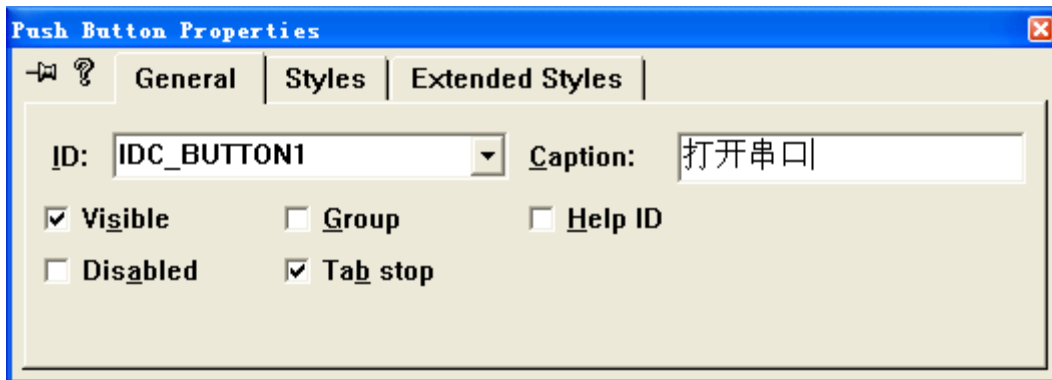
在对话框中放置两个按钮控件和一个串口通信控件



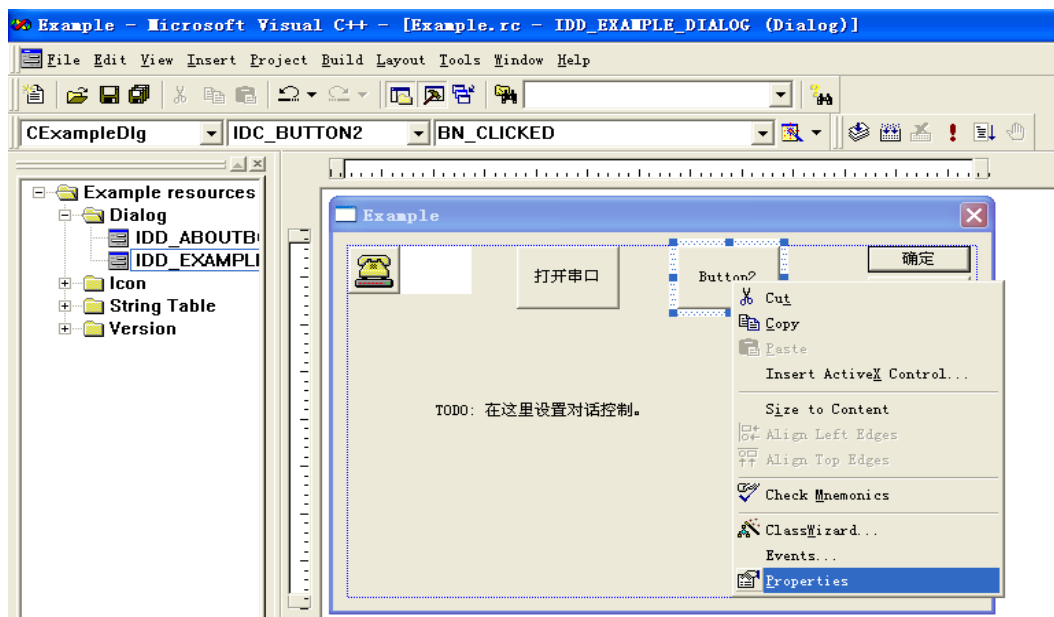
右键→Properties



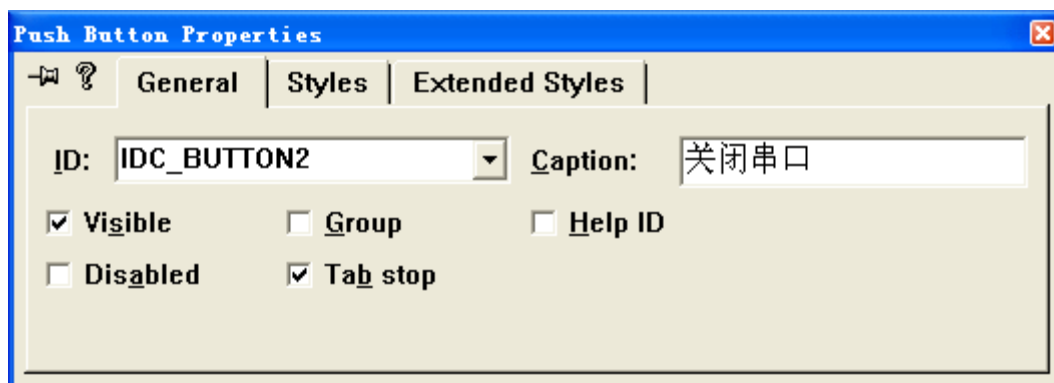
General 中 Caption: 填写“打开串口” → 关闭



右键→Properties



在 General 中 Caption: 填写“关闭串口” → 关闭



双击“打开串口”按钮→对话框中改为 OnBtnOpen→OK→接着在// TODO: Add your control notification handler code here 下面添加如下代码:

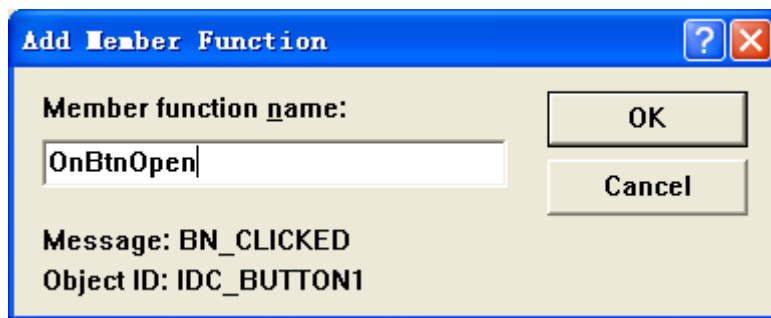
```
if(!myComm.GetPortOpen())
```

```
{  
    myComm.SetPortOpen(true);  
    AfxMessageBox("Serial Port 12 is Opened!");  
}
```

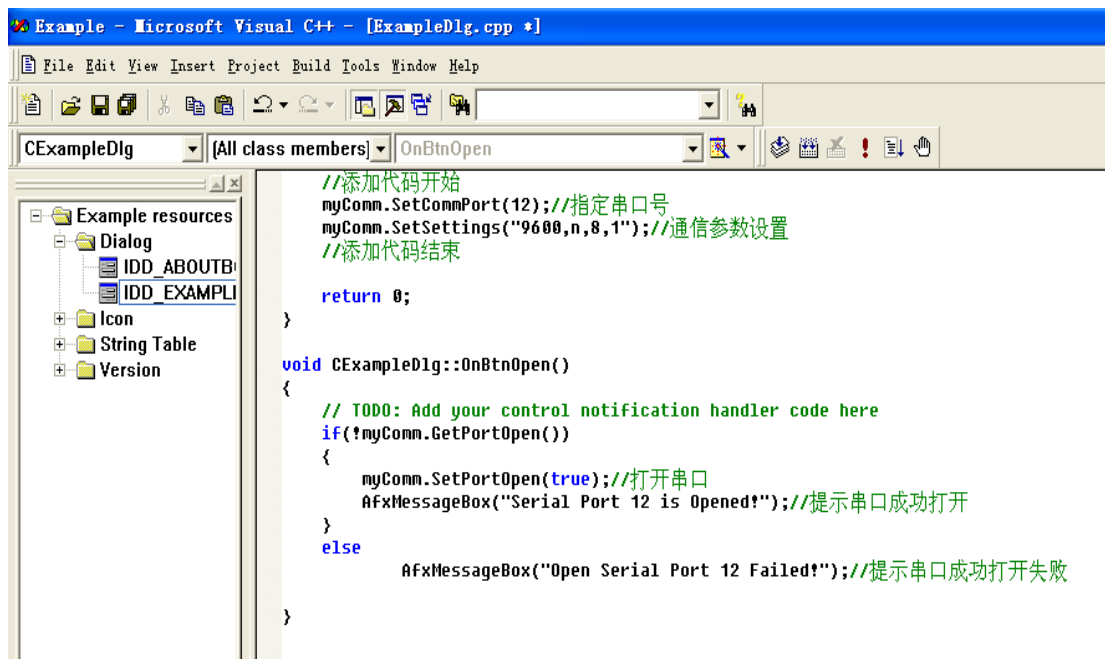
```
else
```

```
    AfxMessageBox("Open Serial Port 12 Failed!");
```

双击“打开串口”按钮→对话框中改为 OnBtnOpen



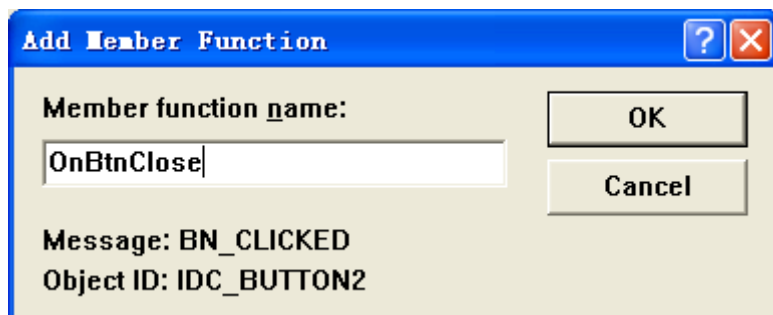
接着在// TODO: Add your control notification handler code here 下面添加代码



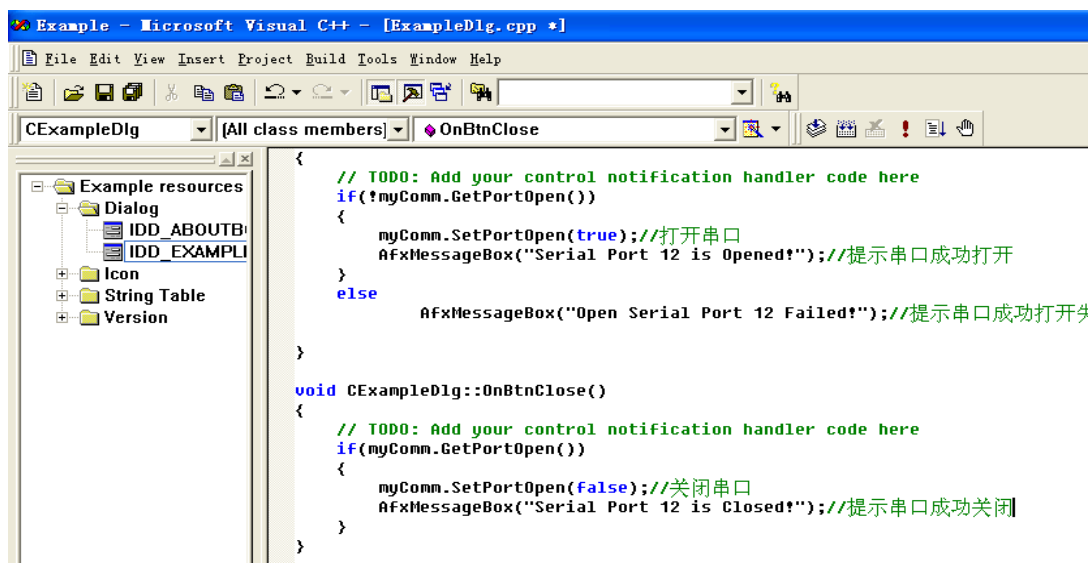
双击“关闭串口”按钮→对话框中改为 OnBtnClose→OK→接着在// TODO: Add your control notification handler code here 下面添加如下代码:

```
if(myComm.GetPortOpen())
{
    myComm.SetPortOpen(false);
    AfxMessageBox("Serial Port 12 is Closed!");
}
```

双击“关闭串口”按钮→对话框中改为 OnBtnClose



接着在// TODO: Add your control notification handler code here 下面添加代码



编译(忽略错误)连接运行，查看结果。

单击打开串口



单击关闭串口

