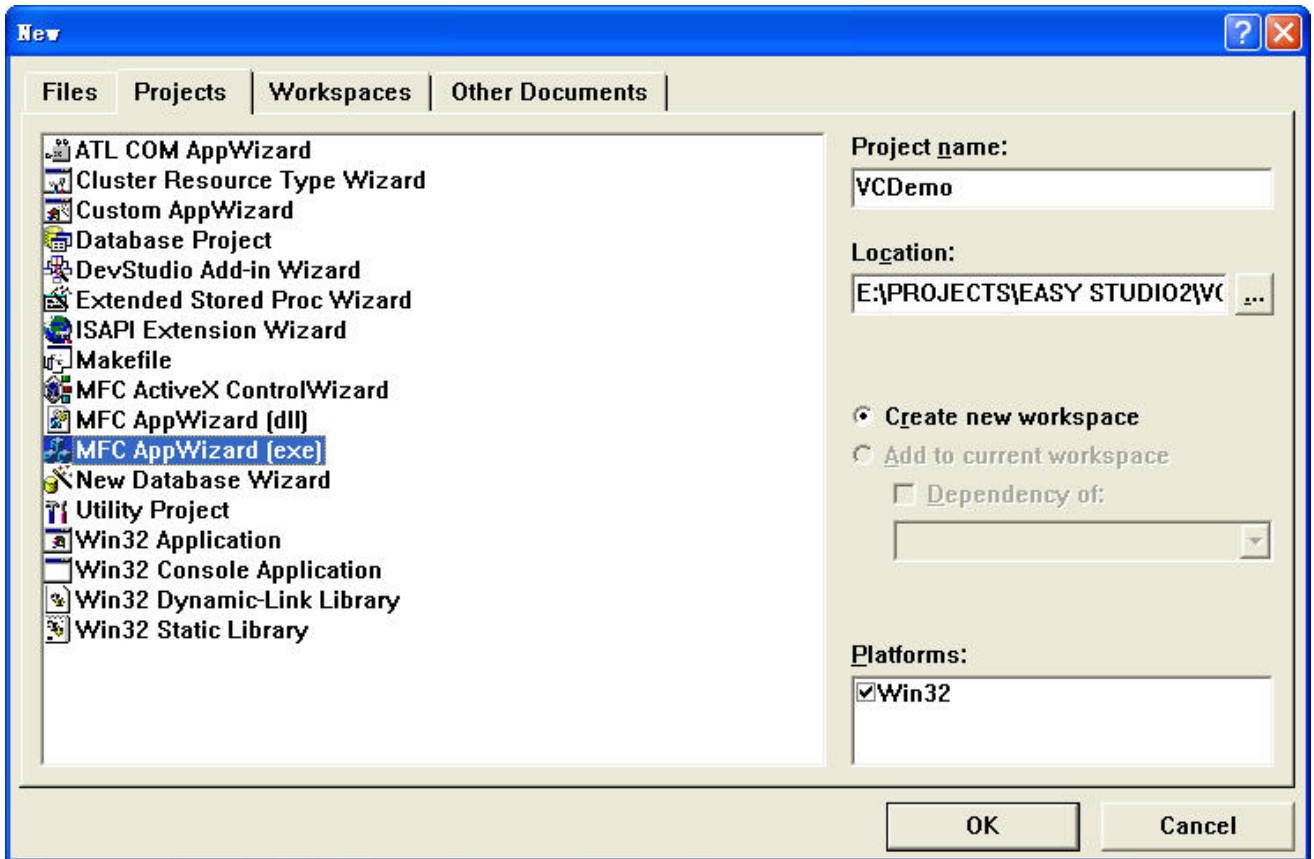


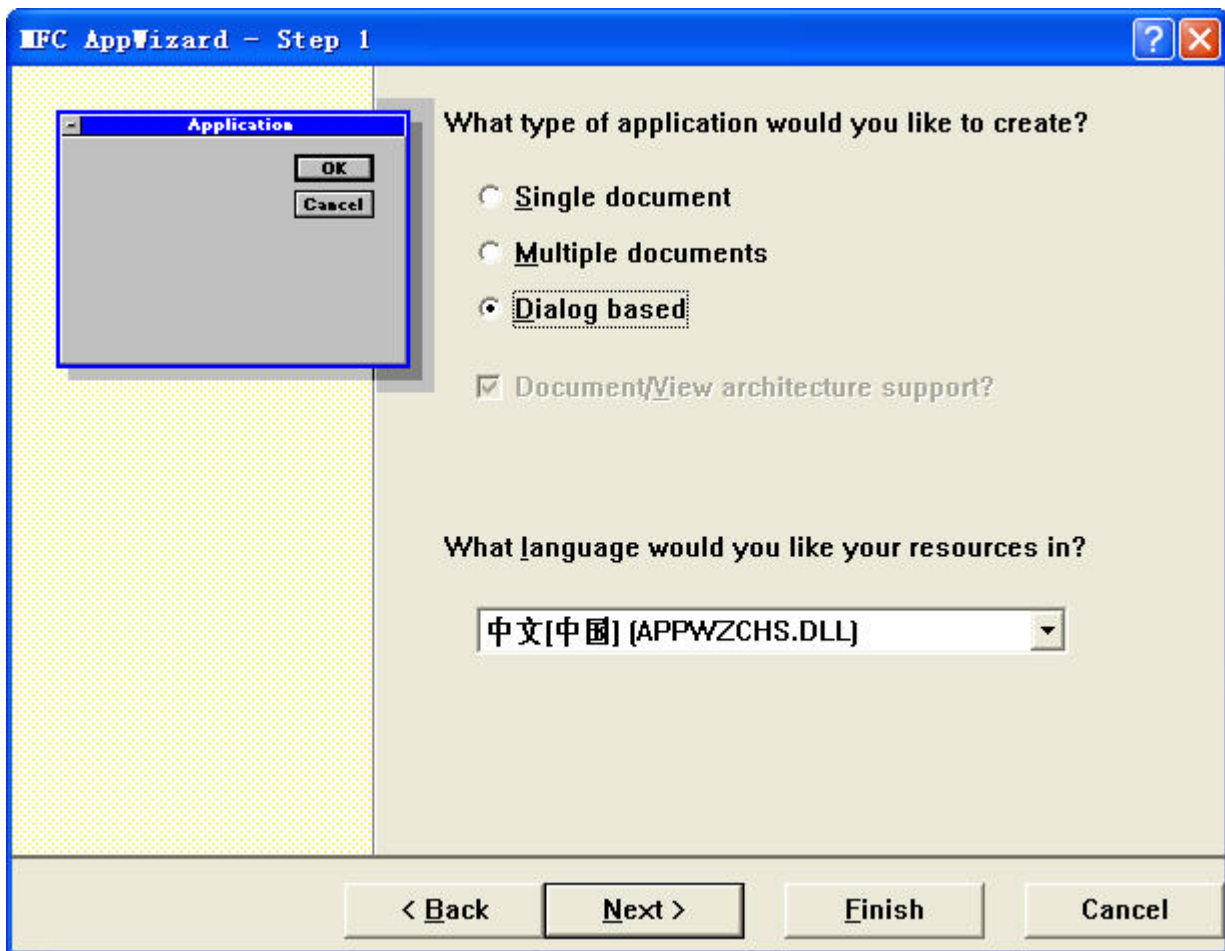
VC6 运动控制入门指南

1 新建 MFC 工程



工程名称位置根据需要编辑和选择

2 选择工程类型



本例中以基于对话框的程序为例，其它类型程序类似

3 复制 VC 开发需要的文件到工程目录



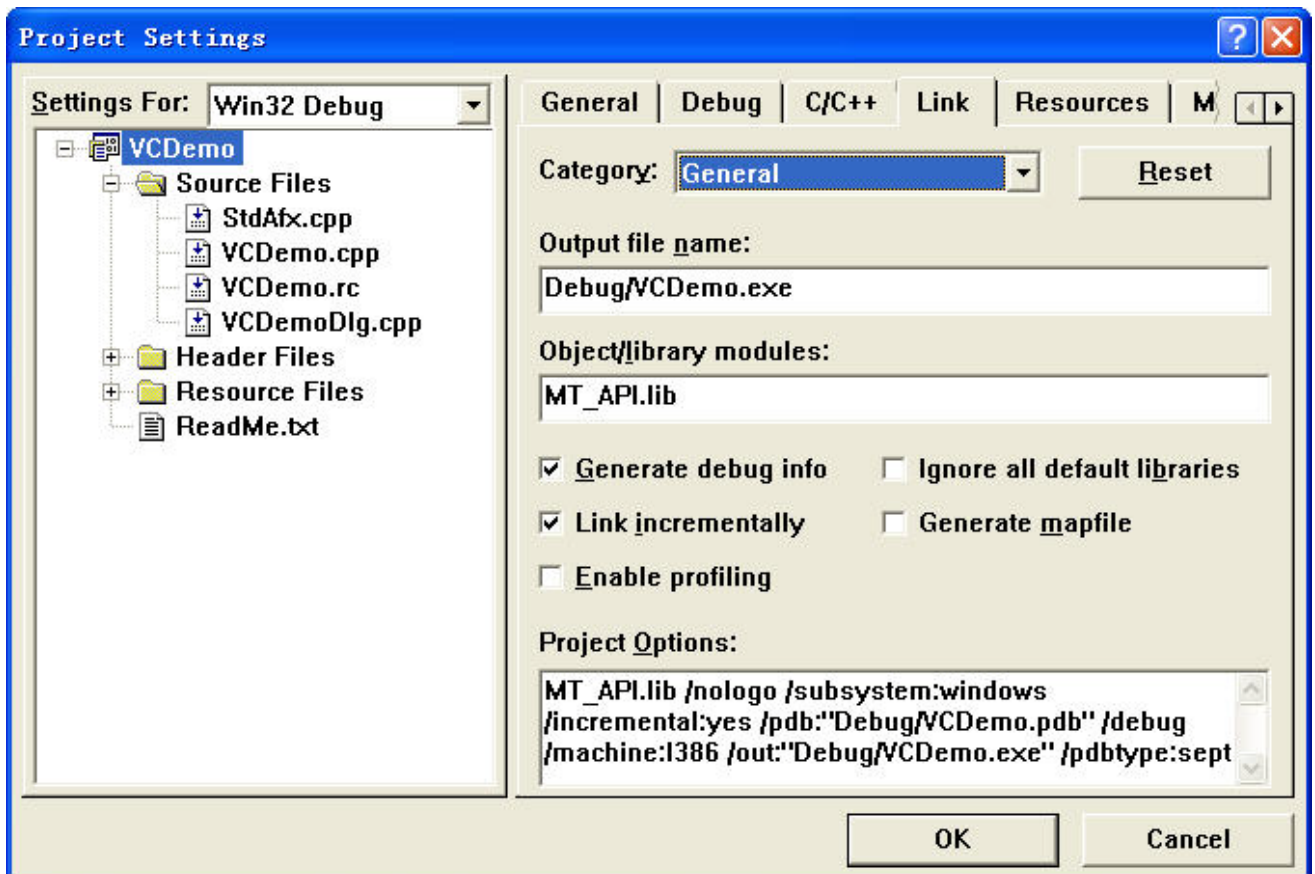
一般【MT_API.h】和【MT_API.lib】放在和工程文件名【xxx.dsw】（本例中为 VCDemo.dsw）一个目录，【MT_API.dll】放到 Debug 或者 Release 目录，保持和最后生成的 exe 在一个目录。

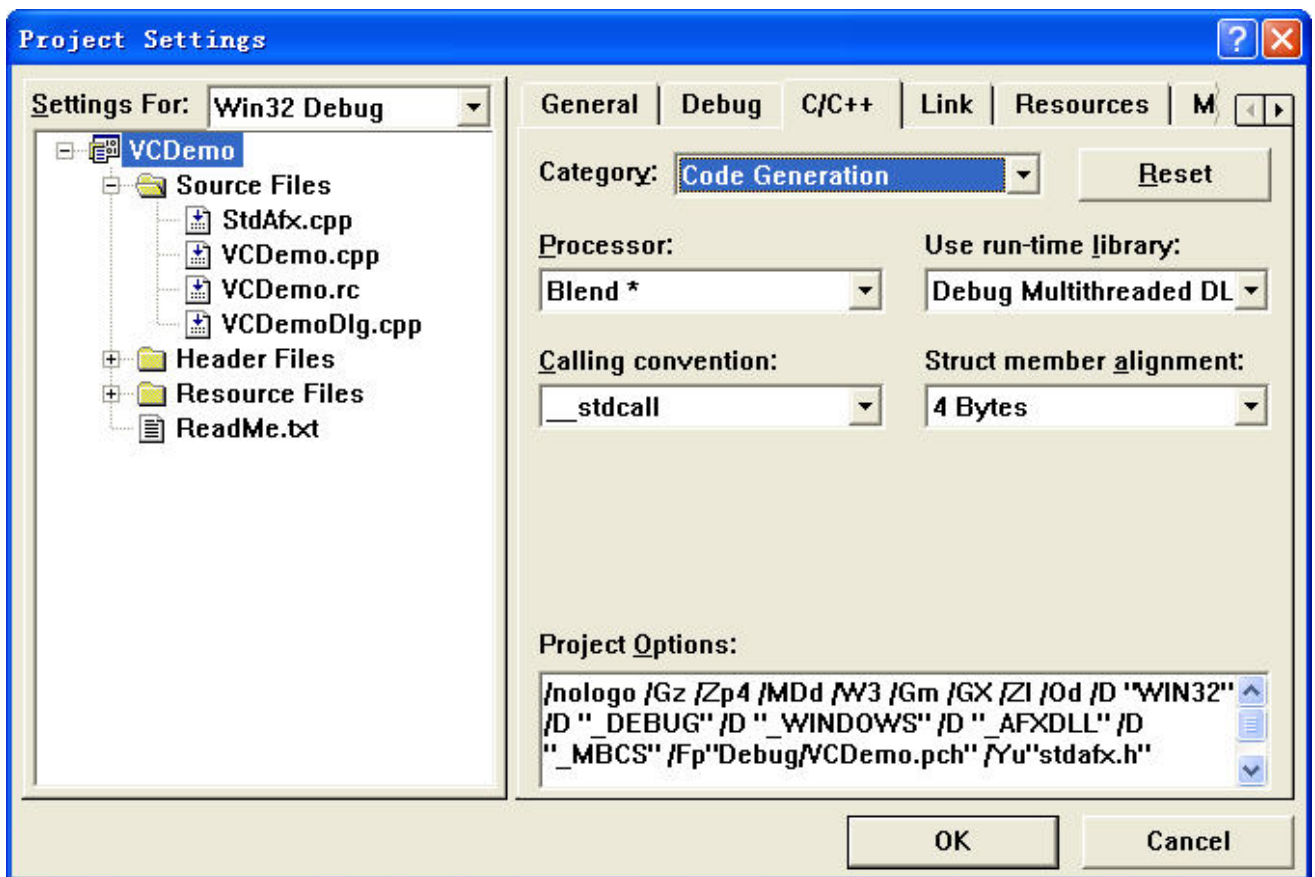
按照上述目录放置默认路径已支持，如果放到别的目录，请保证工程的搜索目录包含这些文件所在的目录

4 配置工程

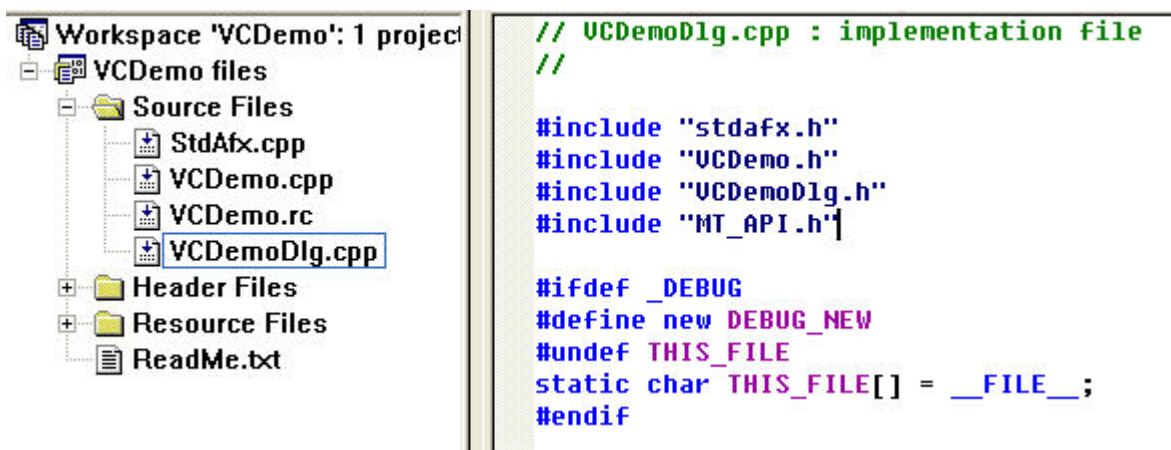
通过 Project/Settings 菜单激活工程属性设置

共两处需要设置为下图所示





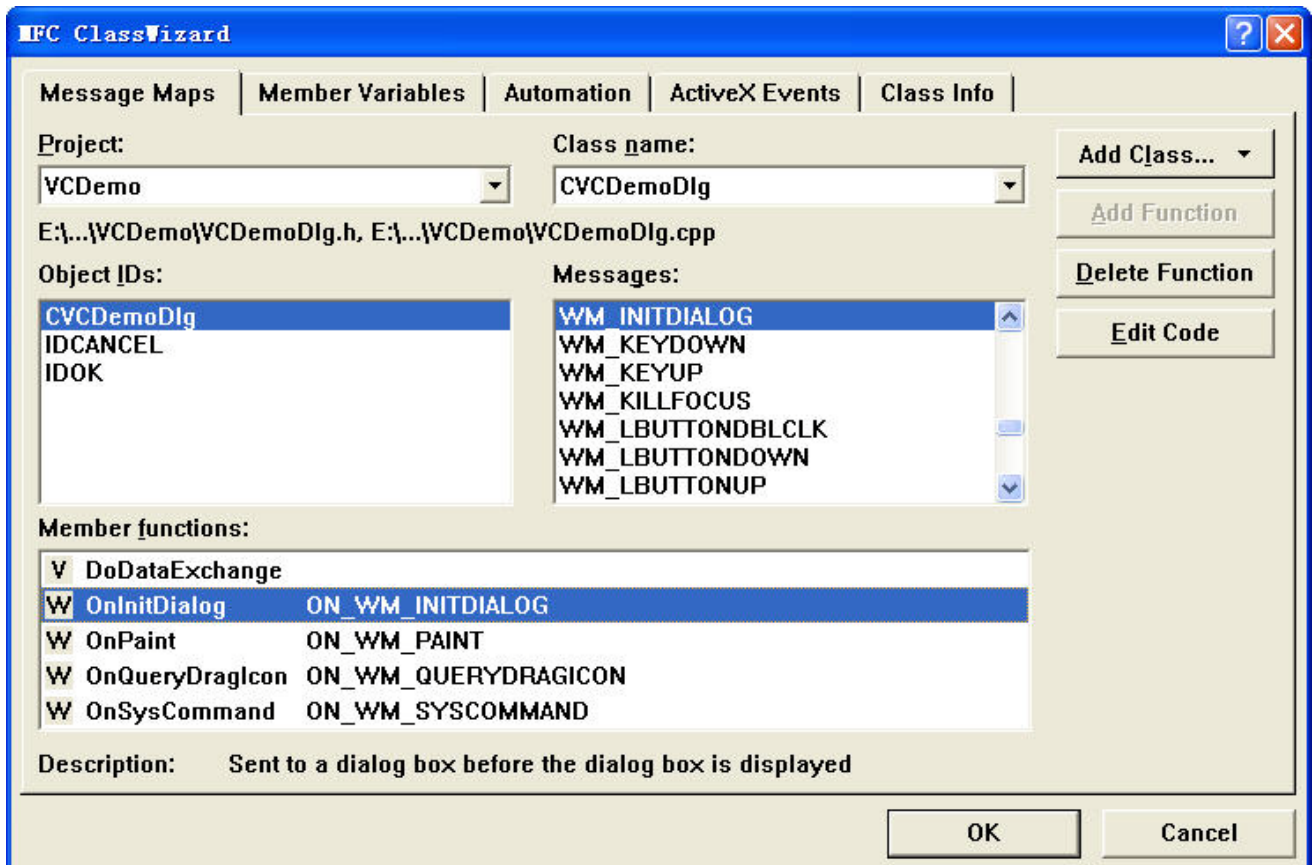
5 引用开发包内容



在需要的文件中引用【MT_API.h】文件

6 初始化

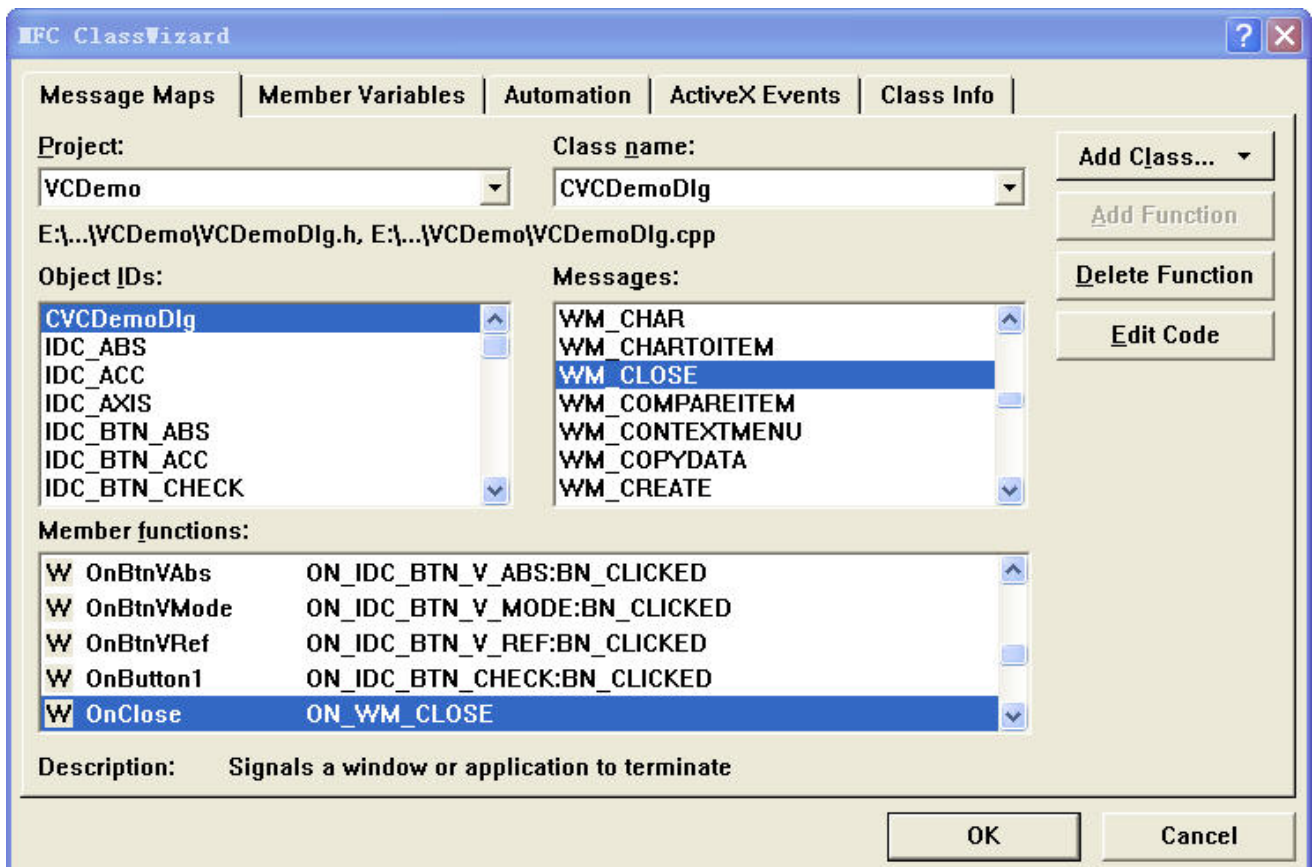
通过类向导，添加对话框初始化函数，非对话框类型工程请选择对应的初始化函数，保证在调用其它 MT 开头的函数前调用【MT_Init】，本函数用来申请控制运动卡的资源，必须首先被调用



```
// TODO: Add extra initialization here  
  
//初始化申请资源  
MT_Init();  
  
return TRUE; // return TRUE unless you set the focus to a control
```

7 资源释放

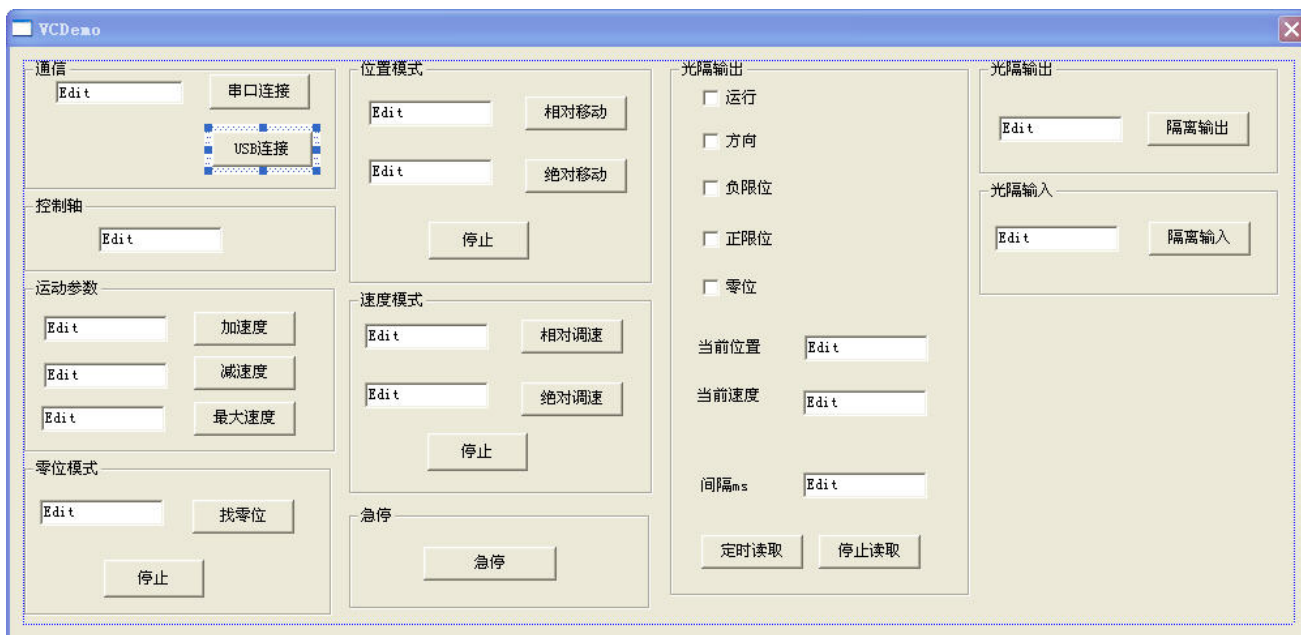
通过类向导在窗口关闭的事件中释放资源，非对话框类型在类似析构函数中做本动作。



```
void CVCDemoDlg::OnClose()
{
    // TODO: Add your message handler code here and/or call default
    MT_DeInit();
    CDialog::OnClose();
}
```

8 制作界面

通过对话框编辑进行需要的界面的设计



9 建立通信

通过【MT_Open_UART】【MT_Open_USB】【MT_Open_Net】函数打开通信口。通过判断返回值来确认是否通信口打开成功；打开成功后通过【MT_Check】函数进行检测真实的通信。

10 添加实用功能

可以通过参数设置相关的函数、不同工作模式控制的函数、状态读取函数来进行编程，具体请参照例程代码。详细的函数说明请查看【MT_SDK_Manual】手册