

# MV-E EM 系列

## 千兆以太网工业相机

VC 开发手册 v1.0.1801

维视数字图像技术有限公司

[产品部]

## 目录

1 开发流程 .....	2
2 实例说明.....	3
第 1 步    新建工程.....	4
第 2 步    窗口布局.....	5
第 3 步    环境配置.....	6
第 4 步    代码开发.....	9
第 5 步    工程运行.....	21
第 6 步    运行结果.....	22

# 1 开发流程

MV-E EM 系列相机,驱动分为 32bit 和 64bit,根据计算机系统安装相应驱动。如果要在 64bit 计算机系统中开发 32bit 相机应用程序,需要用 32bit 相机驱动开发文件。

驱动安装目录:( 以下为安装默认目录 )

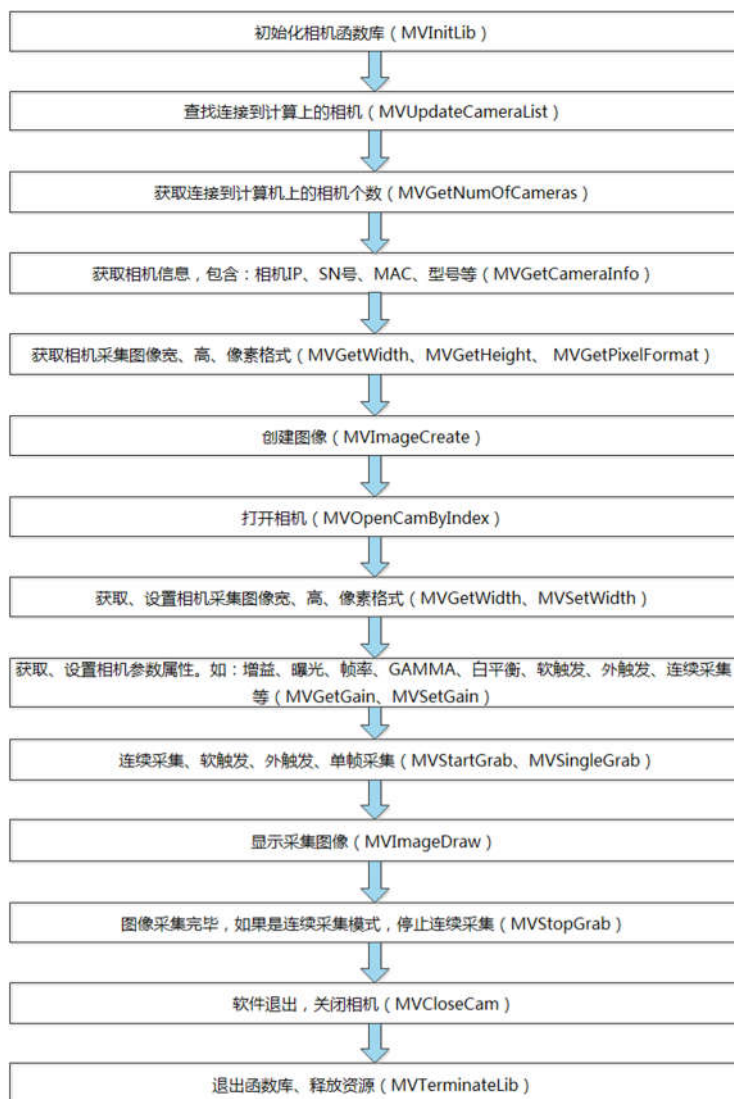
C:\Program Files\ Microvision\E\_EM\SDK\include

C:\Program Files\ Microvision\E\_EM\SDK\lib

C:\Program Files\Microvision\E\_EM\SDK\bin

分别包含相机开发的.h\lib\.dll 文件。

MV-E EM 系列相机,进行二次开发步骤如下( 开发接口参考《MV-EM E 系列千兆以太网工业相机开发说明 v2.0.1711》):



注：在相机采集过程中可以设置增益、曝光、帧率、GAMMA、白平衡等。

## 2 实例说明

如何用 Microsoft Visual C++ 6.0 工具开发一个 Dialog 相机程序。

## 第1步 新建工程

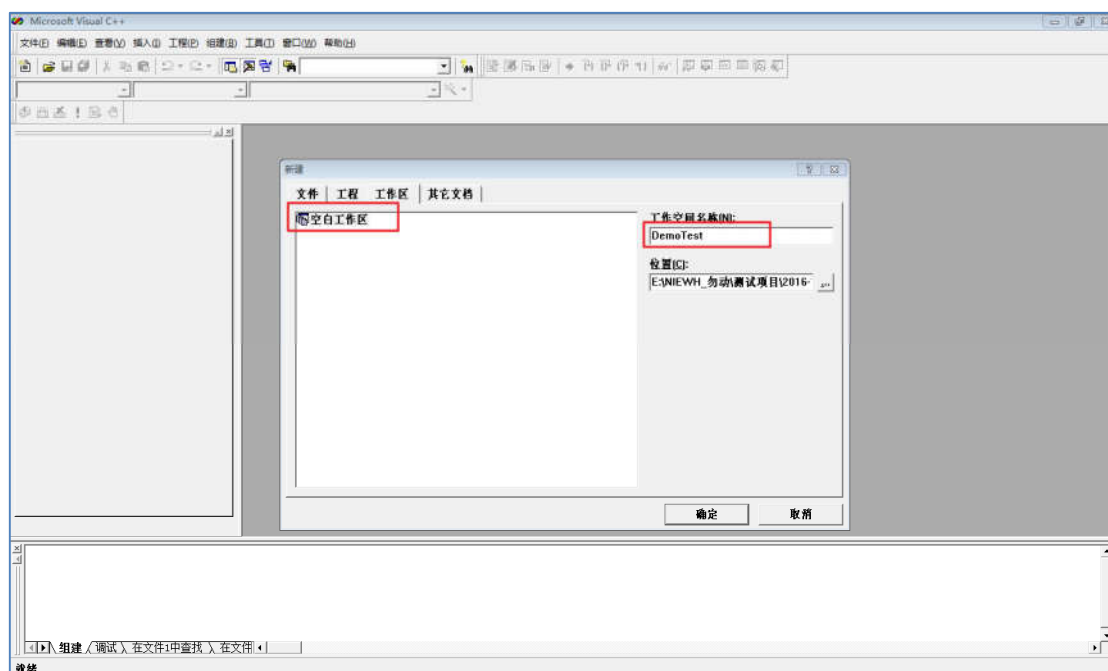


图 1-1-1 Microsoft Visual C++ 6.0 新建工程

在 DemoTest 工程中添加一个 MFC AppWizard ( exe ) 项目,命名 demo。

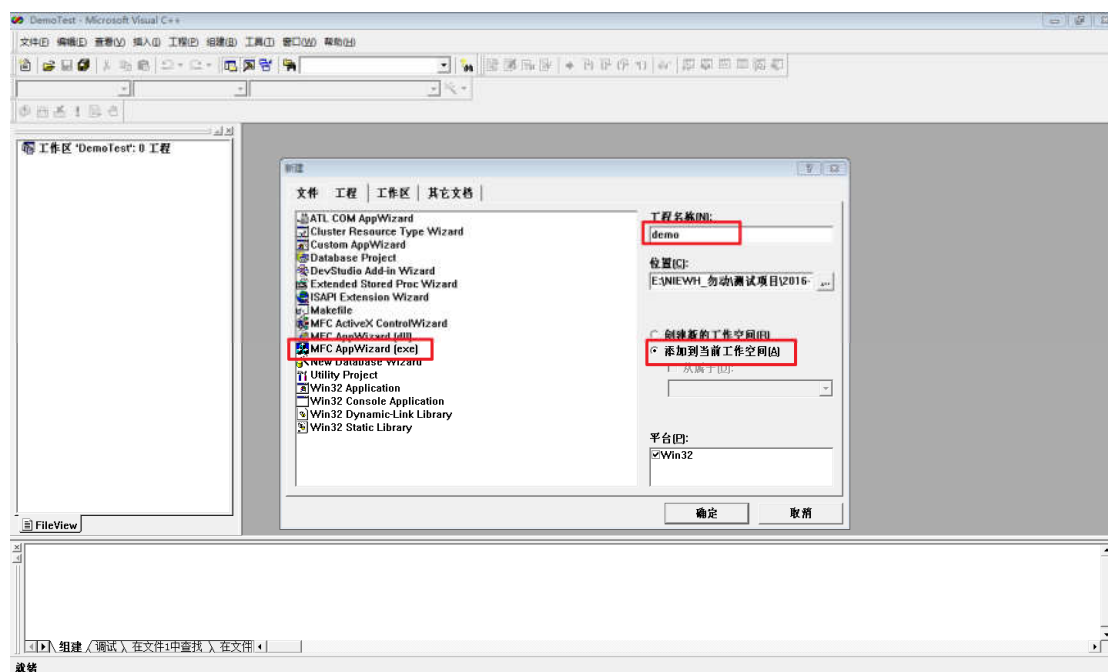


图 1-1-2 Microsoft Visual C++ 6.0 工程添加项目

MFC AppWizard ( exe ) 项目选择“基本对话框”应用程序，点击【完成】。

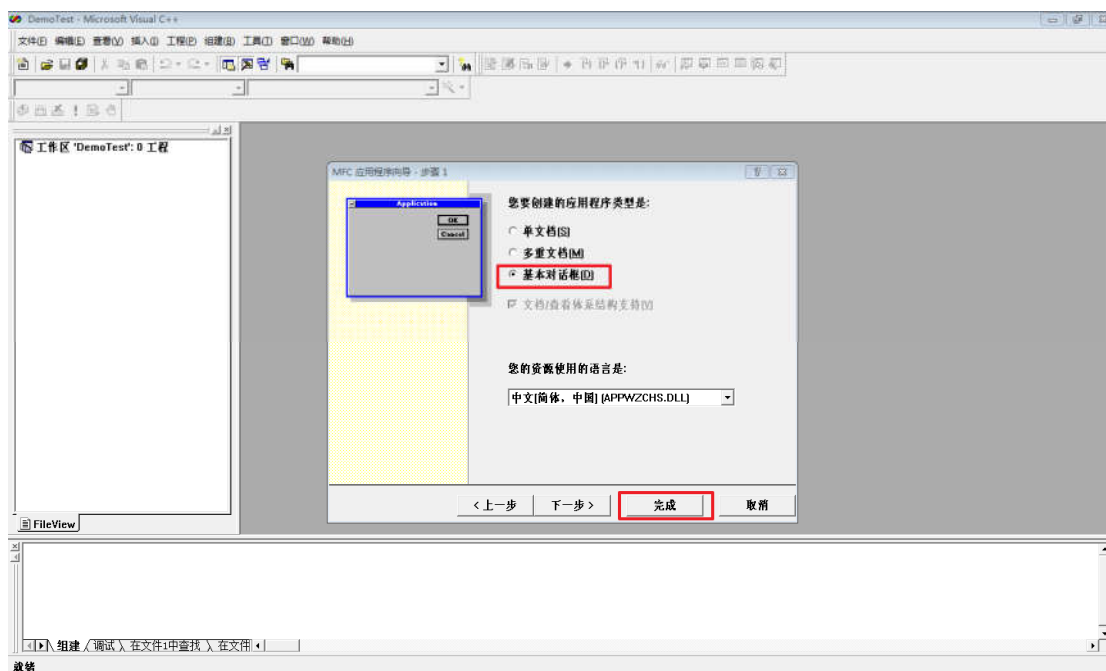


图 1-1-3 Microsoft Visual C++ 6.0 MFC AppWizard ( exe ) 选择应用程序

## 第2步 窗口布局

在 Dialog 窗口上做三个按键 Button ( 命名：打开相机、开始采集、关闭相机，ID：

IDC\_OpenCam、IDC\_StartGrab、IDC\_CloseCam )，一个画布 Picture ( ID：pic )。

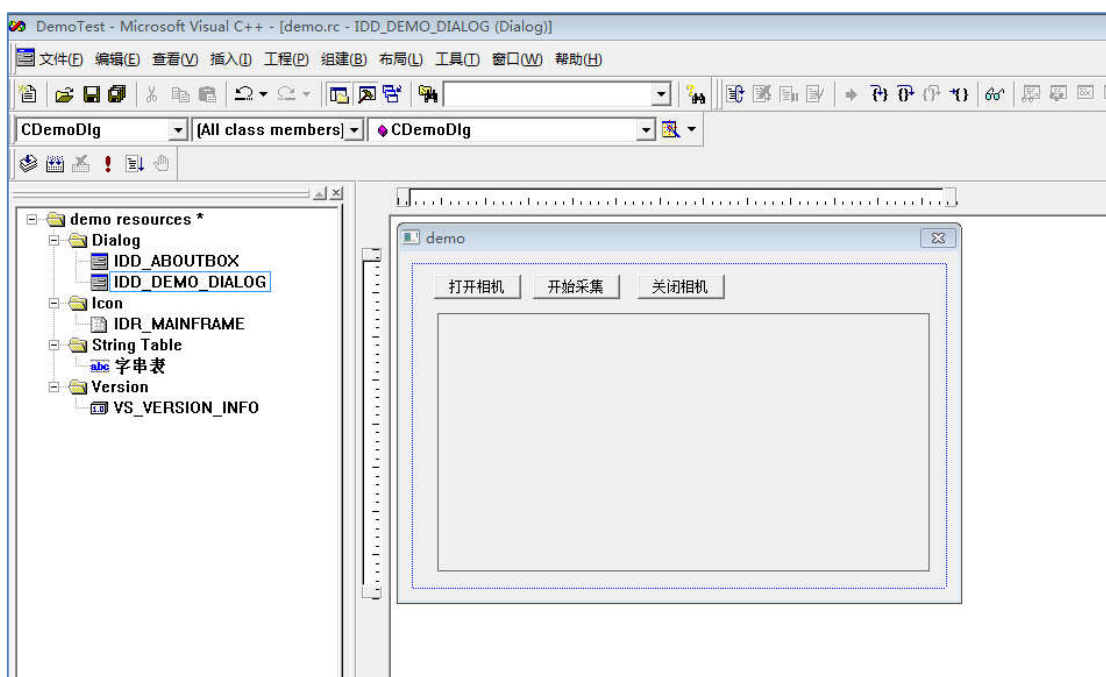


图 1-2-1 Microsoft Visual C++ 6.0 Dialog 窗口化程序界面

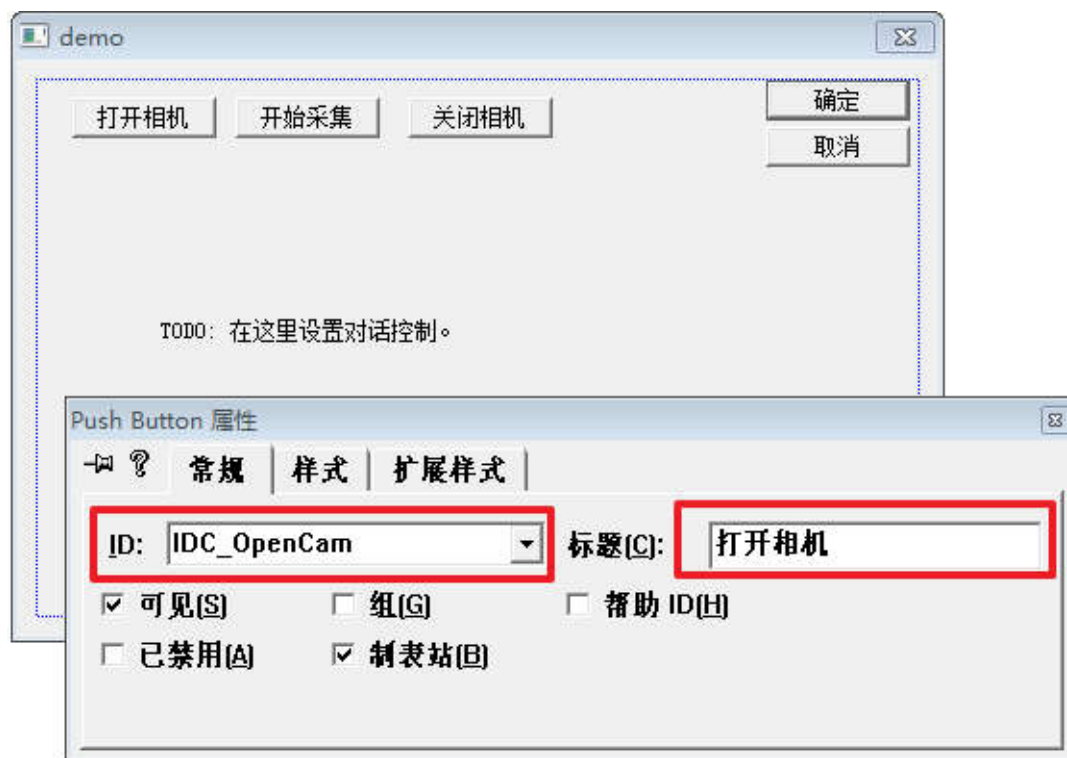


图 1-2-2 Microsoft Visual C++ 6.0 Dialog 修改命名和 ID 界面

### 第3步 环境配置

Demo 项目配置 MV\_E EM 相机开发环境，依次点选 Tools->Options->Directories, Show directories for : Include files 增加 MV\_EM E 相机 SDK .h 文件引用目录。如：

D:\Program Files\Microvision\E\_EM\SDK\include

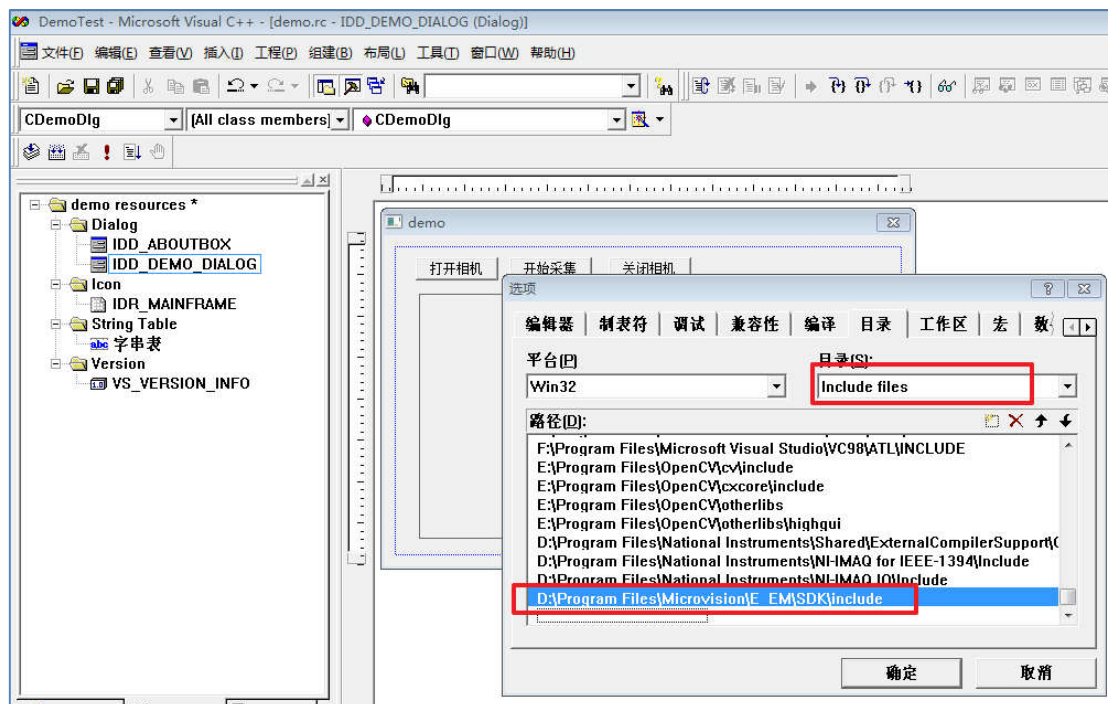


图 1-3-1 Microsoft Visual C++ 6.0 添加头文件引用目录

Tools->Options->Directories, Show directories for : Library files 增加 MV\_EM E 相机 SDK .lib .dll 文件引用目录。如：

D:\Program Files\Microvision\E\_EM\SDK\lib

D:\Program Files\Microvision\E\_EM\SDK\bin



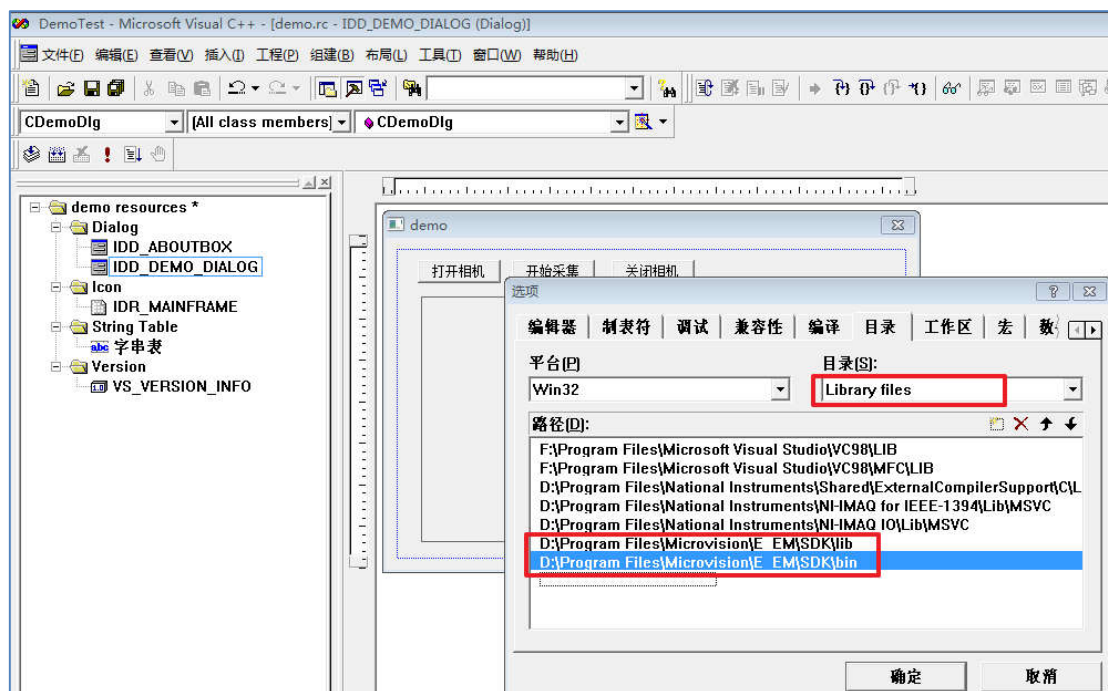


图 1-3-2 VC6.0 添加静态库引用目录

Project->Settings->Link , Object/library modules。增加.lib 库的连接。如:

MVGigE.lib MVCamProptySheet.lib MVTickDlg.lib

多个 lib 库中间用空格隔开。

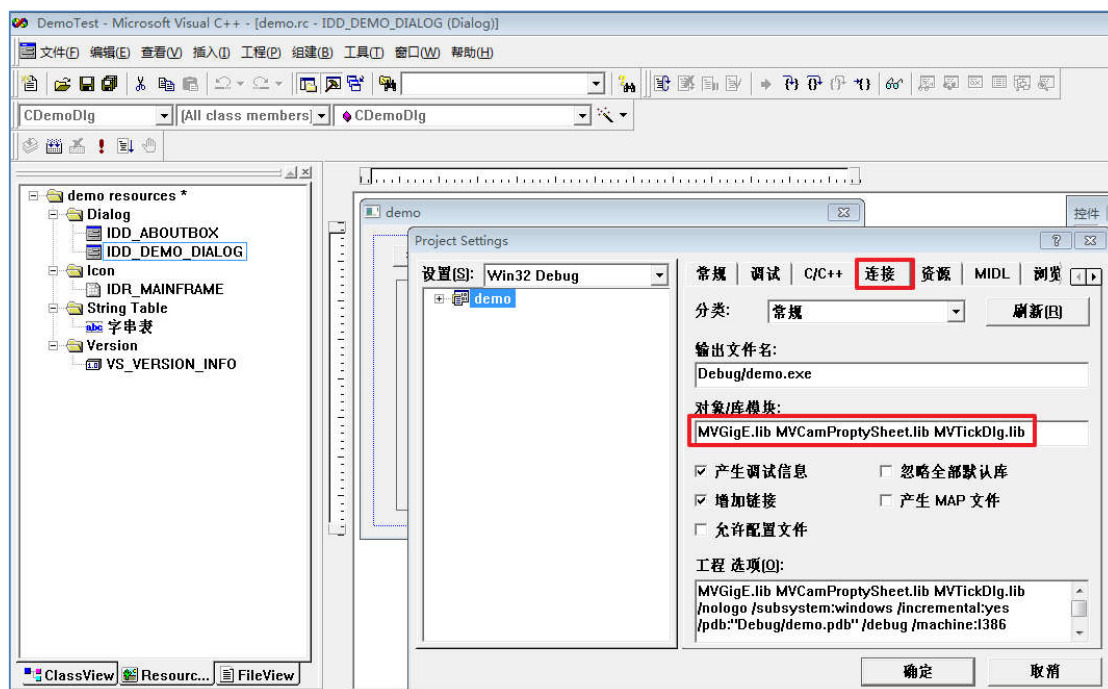


图 1-3-3 Microsoft Visual C++ 6.0 添加静态库链接

## 第4步 代码开发

在【demoDlg.h】文件中添加头文件：

```
#include "MVGigE.h"
```

```
#include "MVImage.h"
```

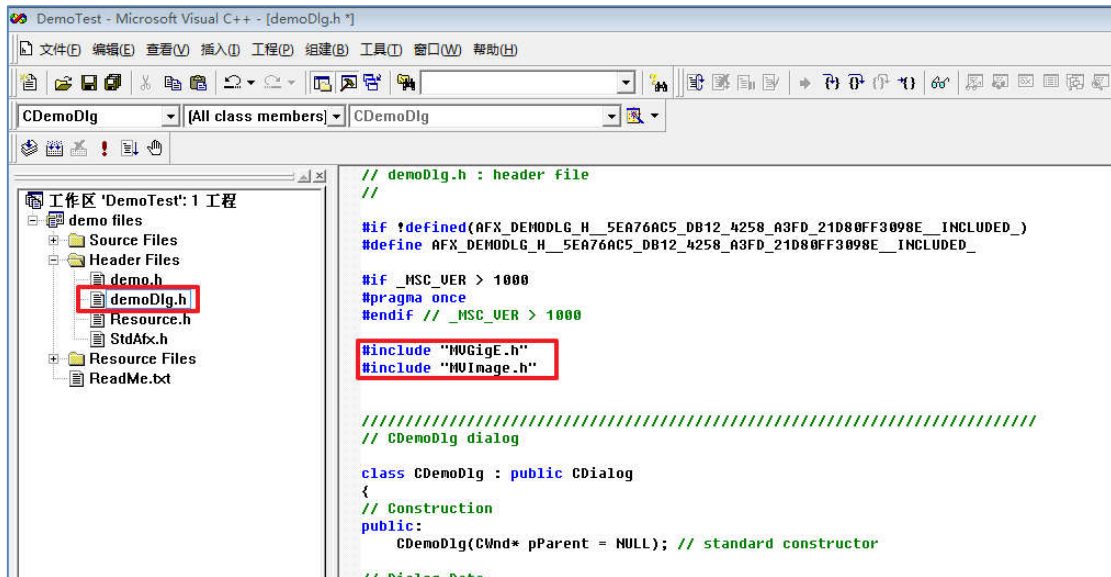


图 1-4-1 Microsoft Visual C++ 6.0 添加头文件

在【demoDlg.h】文件中的 CDemoDlg 类中添加：

```
public:
```

```
//定义相机句柄对象
```

```
HANDLE m_hCam;
```

```
//定义图像处理对象
```

```
MVImage m_image;
```

```
//定义像素格式对象
```

```
MV_PixelFormatEnums m_PixelFormat;
```

```
//采集标识
```

```
bool m_bRun;
```

//通过回调函数获取图像数据信息

```
int OnStreamCB( MV_IMAGE_INFO *pInfo);
```

//画图

```
void DrawImage();
```

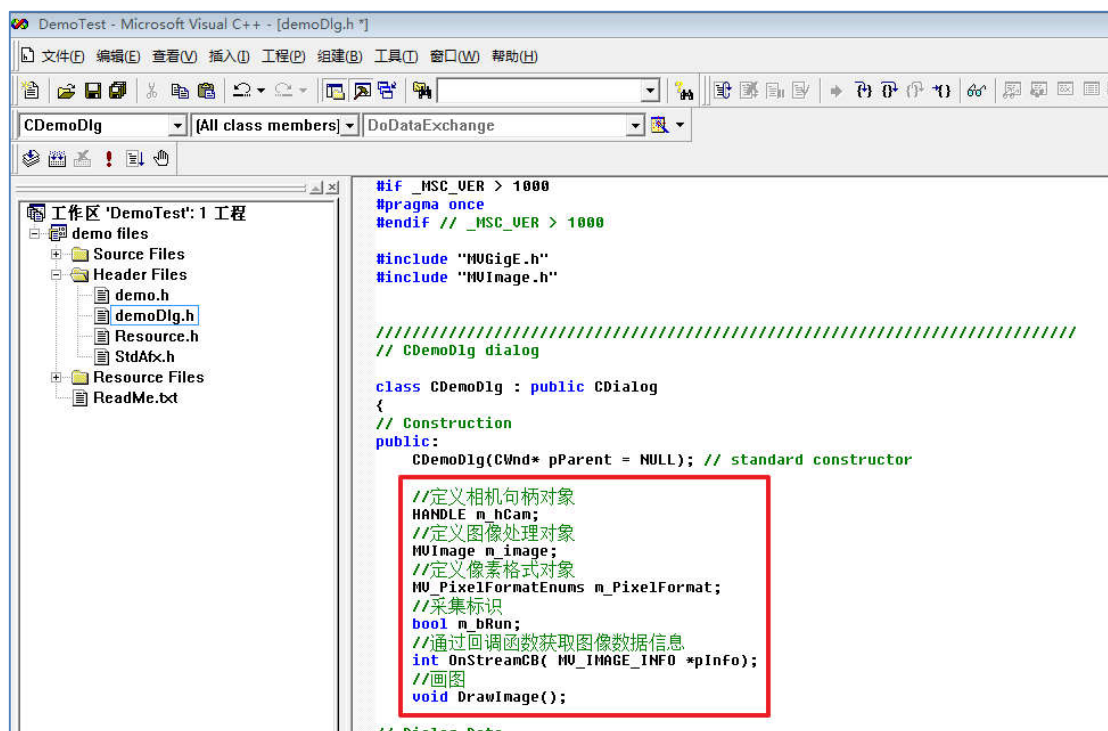


图 1-4-2 Microsoft Visual C++ 6.0 添加对象，方法

在【demoDlg.cpp】文件中的 CDemoDlg 类构造函数中添加：

```
,m_hCam(NULL)
```

```
,m_bRun(false)
```

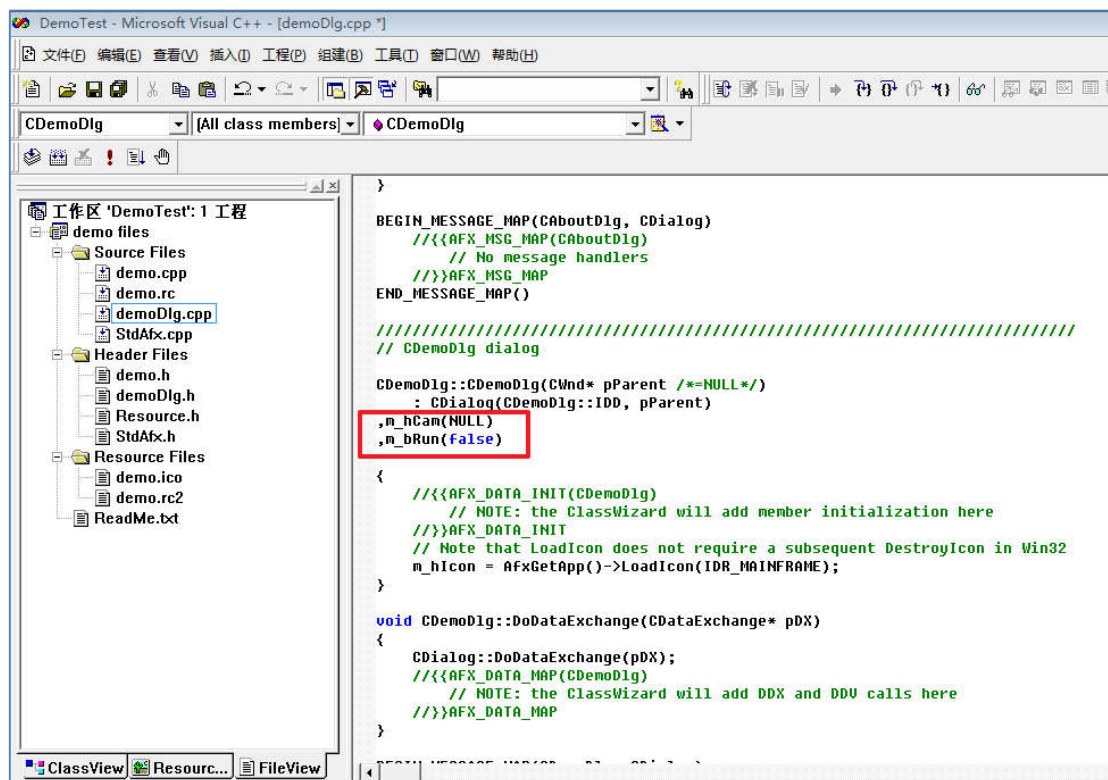


图 1-4-3 Microsoft Visual C++ 6.0 初始化对象

在【demoDlg.cpp】文件中的 OnInitDialog 函数中添加：

```

MVSTATUS_CODES r;

//初始化函数库

r = MVInitLib();

if (r != MVST_SUCCESS)

{

    MessageBox("函数库初始化失败！", "提示", MB_ICONWARNING);

    return TRUE;

}

//产找连接计算机的相机

r = MVUpdateCameraList();
    
```

```
if (r != MVST_SUCCESS)

{

    MessageBox("查找连接计算机的相机失败！", "提示", MB_ICONWARNING);

    return TRUE;

}
```

GetDlgItem(IDC\_OpenCam)->EnableWindow(true);

GetDlgItem(IDC\_StartGrab)->EnableWindow(false);

GetDlgItem(IDC\_CloseCam)->EnableWindow(false);

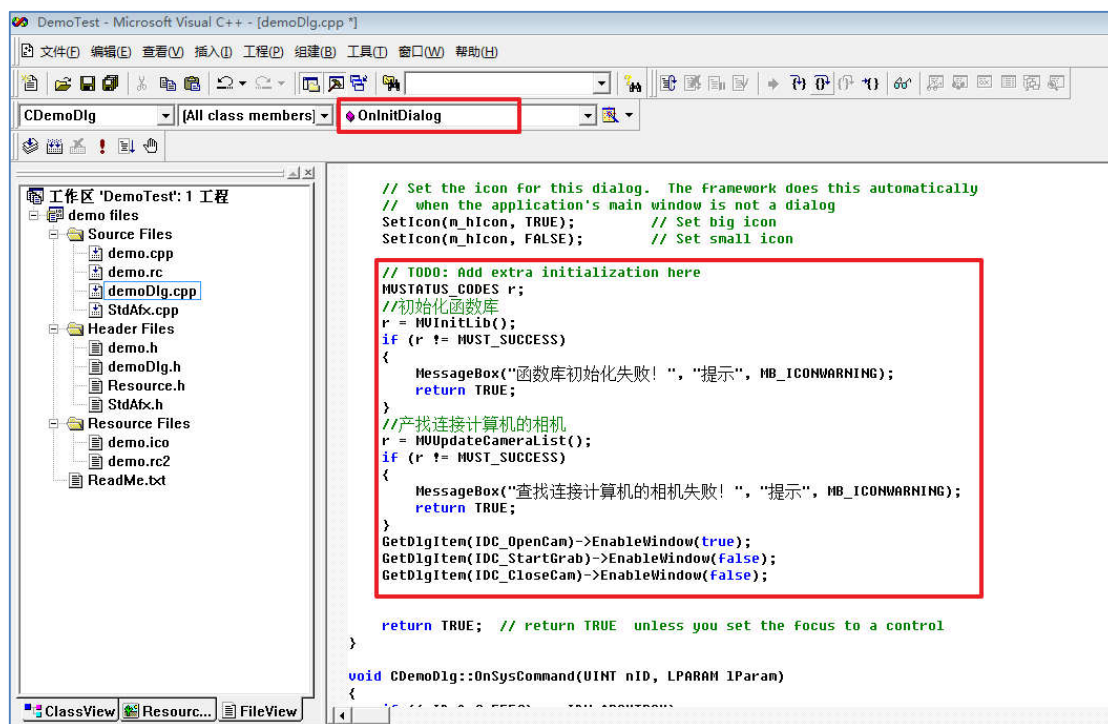


图 1-4-4 Microsoft Visual C++ 6.0 初始化函数库、查找连接计算机的相机

在【打开相机】按钮，点击事件 BN\_CLICKED 中添加：

```
void CDemoDlg::OnOpenCam()

{

    int nCams = 0;
```

```
//获取连接计算机的相机个数

    MVGetNumOfCameras(&nCams);

    if( nCams == 0 )

    {

        MessageBox(" 没有找到相机,请确认连接和相机 IP 设置","提示",MB_ICONWARNING);

        return ;

    }

//打开第 0 个相机

    MVSTATUS_CODES r = MVOpenCamByIndex(0, &m_hCam);

    if (m_hCam == NULL)

    {

        if (r == MVST_ACCESS_DENIED)

            MessageBox("无法打开相机,可能正被别的软件控制","提示",MB_ICONWARNING);

        else

            MessageBox("无法打开相机","提示",MB_ICONWARNING);

        return;

    }

    int w,h;

//获取图像宽
```

```
MVGetWidth(m_hCam,&w);

//获取图像高

MVGetHeight(m_hCam,&h);

//获取图像像素格式

MVGetPixelFormat(m_hCam,&m_PixelFormat);

//创建图像

m_image.CreateByPixelFormat(w,h,m_PixelFormat);


GetDlgItem(IDC_OpenCam)->EnableWindow(false);

GetDlgItem(IDC_StartGrab)->EnableWindow(true);

GetDlgItem(IDC_CloseCam)->EnableWindow(false);

}
```



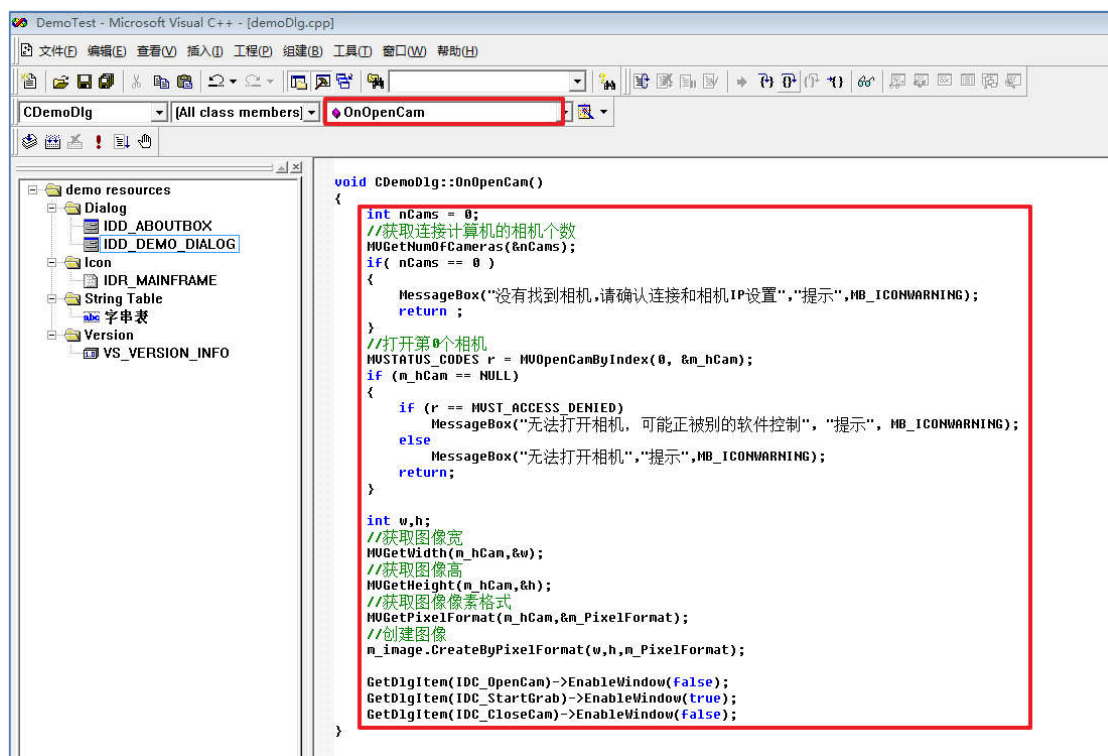


图 1-4-5 Microsoft Visual C++ 6.0 打开相机、创建图像信息

在【开始采集】按钮，点击事件 BN\_CLICKED 中添加：

```

void CDemoDlg::OnStartGrab()
{
    TriggerModeEnums enumMode;

    //获取触发模式

    MVGetTriggerMode(m_hCam, &enumMode);

    //如果相机不是连续采集模式

    if (enumMode != TriggerMode_Off)
    {
        //设置相机连续采集模式

        MVSetTriggerMode(m_hCam, TriggerMode_Off);
    }
}
    
```



```

        Sleep(100);

    }

    //开始采集

    MVStartGrab(m_hCam,StreamCB,( ULONG_PTR)this);

    m_bRun = true;

    GetDlgItem(IDC_OpenCam)->EnableWindow(false);

    GetDlgItem(IDC_StartGrab)->EnableWindow(false);

    GetDlgItem(IDC_CloseCam)->EnableWindow(true);

}

```

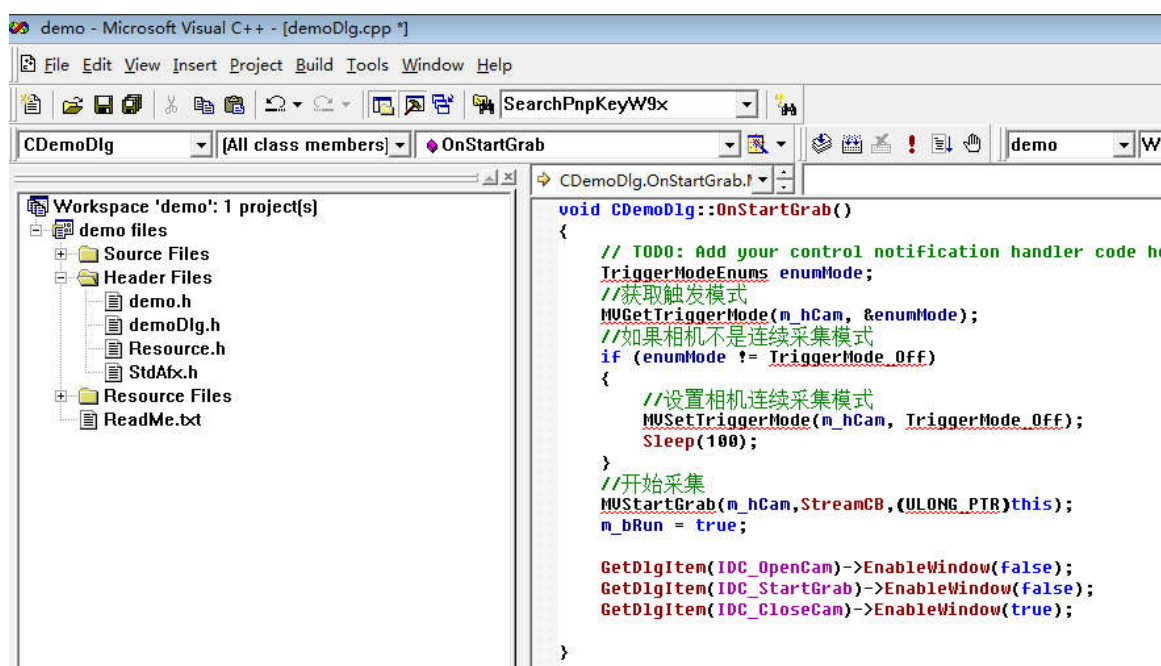


图 1-4-6 Microsoft Visual C++ 6.0 开始采集，连续采集获取图像数据

获取原始图像数据，转成图像格式进行显示：

```

void CDemoDlg::DrawImage()

{

```

```
CRect rct;

GetDlgItem(pic)->GetClientRect(&rct);

int dstW = rct.Width();

int dstH = rct.Height();


CDC *pDC = GetDlgItem(pic)->GetDC();

{

    pDC->SetStretchBltMode(COLORONCOLOR);

    //将 m_image 图像画到 pic 控件上

    m_image.Draw(pDC->GetSafeHdc(),0,0,dstW,dstH);

}

ReleaseDC(pDC);

}


int CDemoDlg::OnStreamCB( MV_IMAGE_INFO *pInfo)

{

    //将采集到的原始帧转化为 m_image 图像格式

    MVInfo2Image(m_hCam,pInfo,&m_image);

    //画图

    DrawImage();

    return 0;

}
```

```
int __stdcall StreamCB(MV_IMAGE_INFO *pInfo, ULONG_PTR nUserVal)
{
    CDemoDlg *pDlg = (CDemoDlg *)nUserVal;

    return ( pDlg->OnStreamCB(pInfo) );
}
```

注：int \_\_stdcall StreamCB 该函数必须放在调用它的函数上面

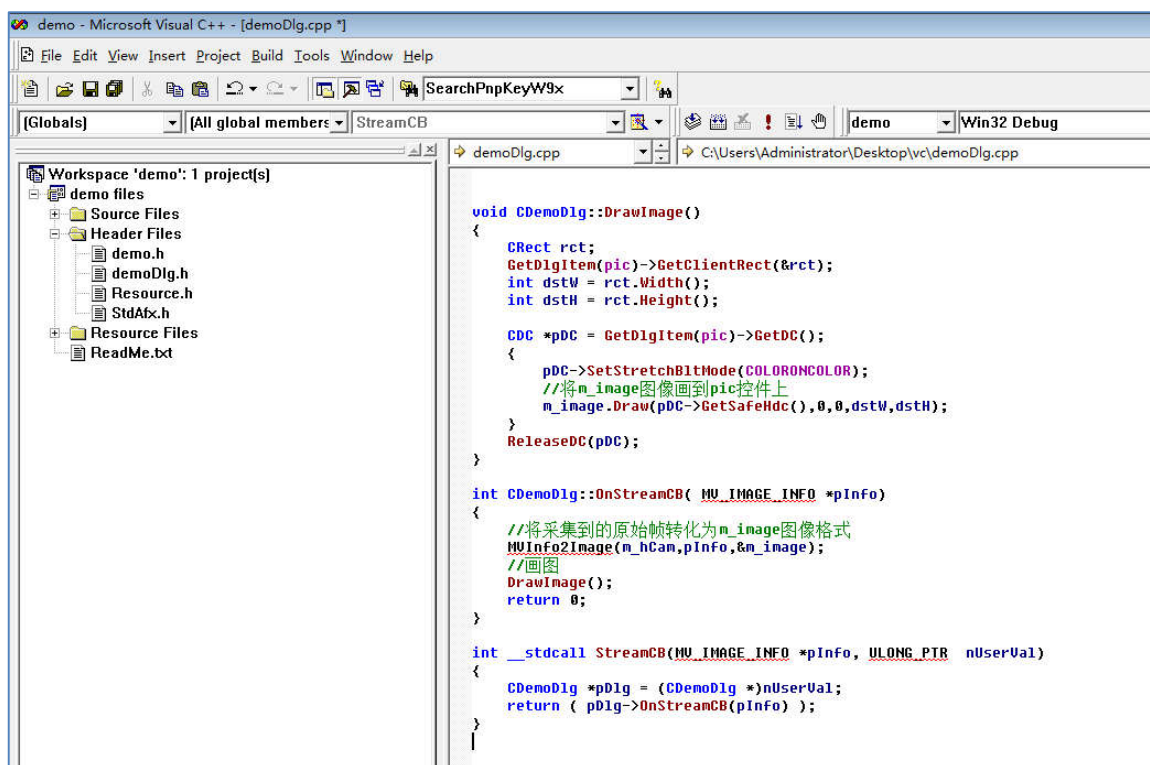


图 1-4-7 Microsoft Visual C++ 6.0 图像采集回调、画图函数实现

在【关闭相机】按钮，点击事件 BN\_CLICKED 中添加：

```
void CDemoDlg::OnCloseCam()
{

```

```

//停止采集

MVStopGrab(m_hCam);

//关闭相机

MVCloseCam(m_hCam);

m_bRun = false;

GetDlgItem(IDC_OpenCam)->EnableWindow(true);

GetDlgItem(IDC_StartGrab)->EnableWindow(false);

GetDlgItem(IDC_CloseCam)->EnableWindow(false);

}

```

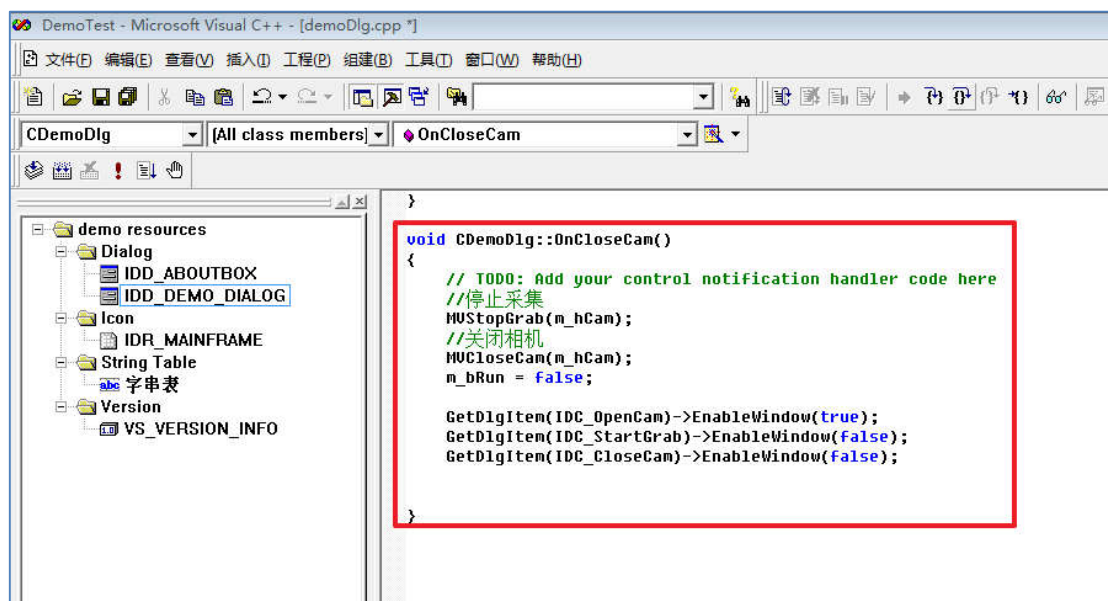


图 1-4-8 Microsoft Visual C++ 6.0 停止采集、关闭相机

在 IDD\_DEMO\_DIALOG 窗口关闭事件 WM\_CLOSE 中添加：

```
void CDemoDlg::OnClose()
```

```
{  
  
    if(m_bRun != false)  
  
    {  
  
        //停止采集  
  
        MVStopGrab(m_hCam);  
  
    }  
  
    if(!GetDlgItem(IDC_OpenCam)->IsWindowEnabled())  
  
    {  
  
        //关闭相机  
  
        MVCloseCam(m_hCam);  
  
    }  
  
    //释放相机资源  
  
    MVTerminateLib();  
  
    CDialog::OnClose();  
  
}
```

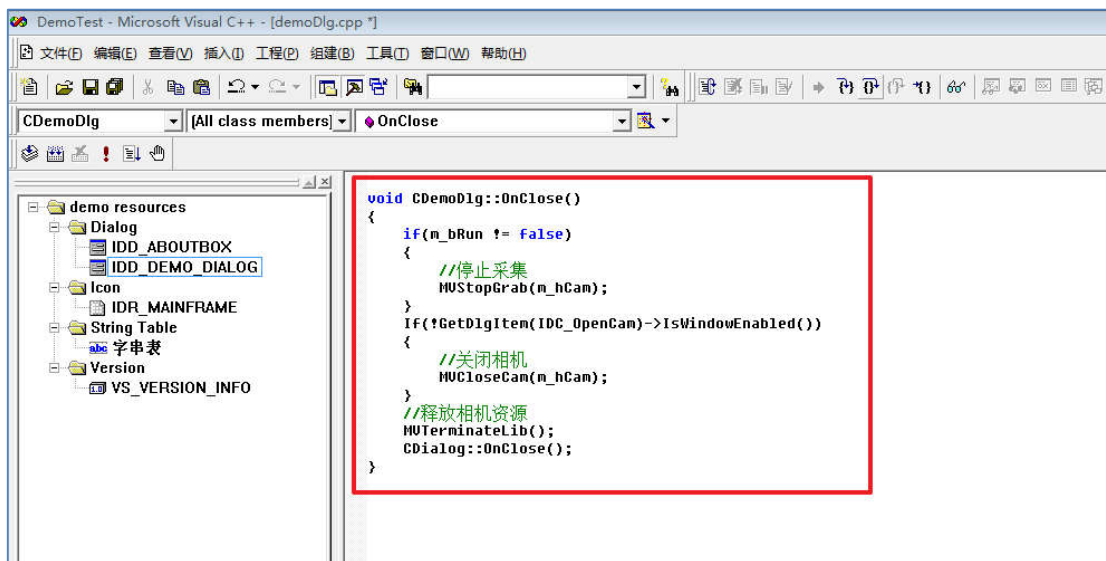


图 1-4-9 Microsoft Visual C++ 6.0 停止采集、关闭相机、释放资源

## 第5步 工程运行

保证以上操作步骤正确无误后，开始运行程序，弹出如下界面，则证明运行成功

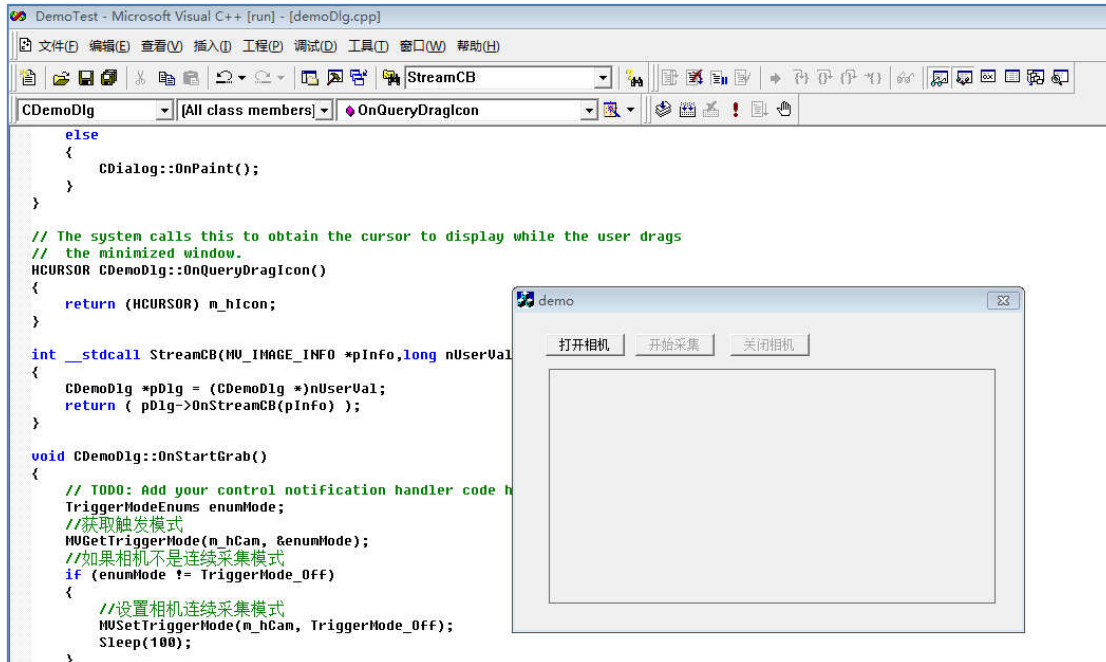


图 1-5-1 demo 程序编译成功界面

## 第6步 运行结果

依次点选红色方框按钮可进入连续采集图像界面，如下图所示。



图 1-6-1 连续采集图像界面

### 西安（总部）

电话：4000-400-860

传真：4000-400-860 转 3

Email: sales@xamv.com

### 深圳

电话：0755-33034411

Email: sz.sale@xamv.com

### 北京

电话：010-58246500

Email: tuxiangmv@126.com

### 上海

电话：4000-400-860

Email: xamv10@126.com

©Microvision 2017

[www.microvision.com.cn](http://www.microvision.com.cn)

**Microvision**  
维视图像



维视图像营销 QQ  
4000400860



维视图像微信  
microvision400