

工业相机 SDK 二次开发示例程序说明（Labview 版）

【摘要】

本文档主要介绍了使用工业相机 SDK(Software Development Kit)开发 Labview 程序方法及过程。在 SDK 开发包目录下，提供了两个 Labview 示例程序，分别为单相机示例程序 Samples 和双相机示例程序 TwoCameraSamples。示例程序分别从宏观角度展示了利用 MvCameraControlSDK 进行开发的方法，除此之外我们还对 C 语言版本的 SDK 进行了 Labview VI 封装，封装成支持 Labview 调用的子 VI。本文档就这个 Labview 示例程序的操作方法和开发流程及各个 VI 展开讨论，方便用户快速入门使用 Labview SDK。

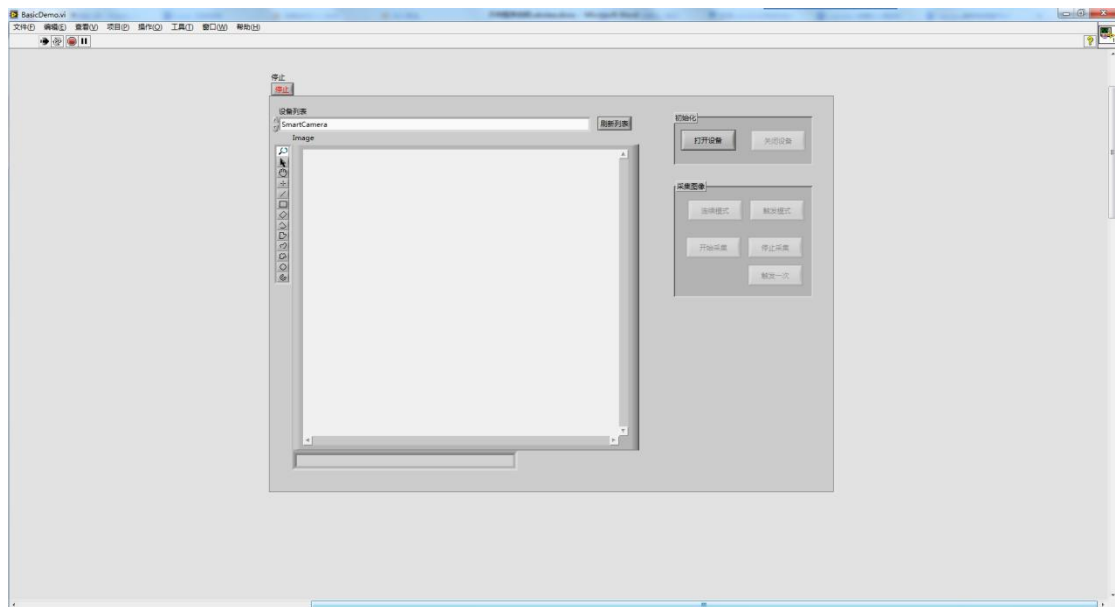
一. Samples 使用步骤及开发流程

Samples 是一个基本示例程序，包含了 SDK 使用过程中常用的一些接口调用，初次使用工业相机 SDK 进行二次开发的用户推荐首先参考 Samples，其涵盖了大多数用户对 SDK 的使用方法示例需求。

1.1 Demo 软件使用步骤

1.1.1 界面总体

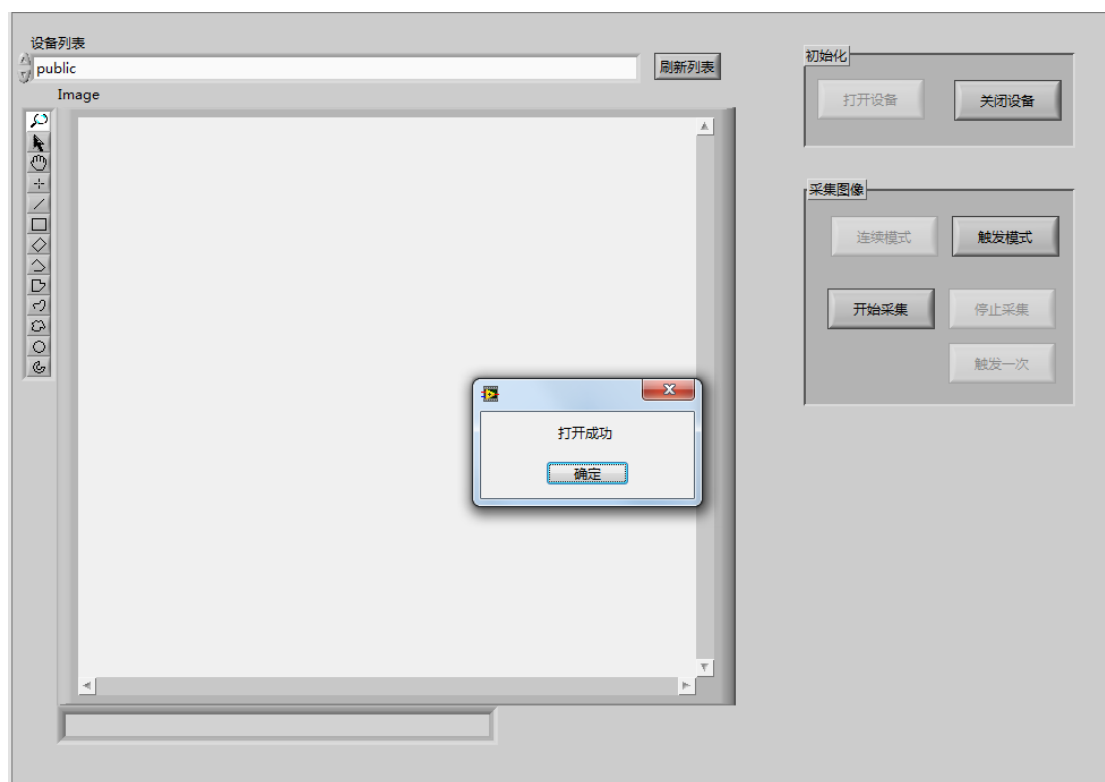
软件界面总览，一共包括四个控制模块(刷新设备，初始化，图像采集，停止运行)、一个下拉设备列表和一个图像显示区域



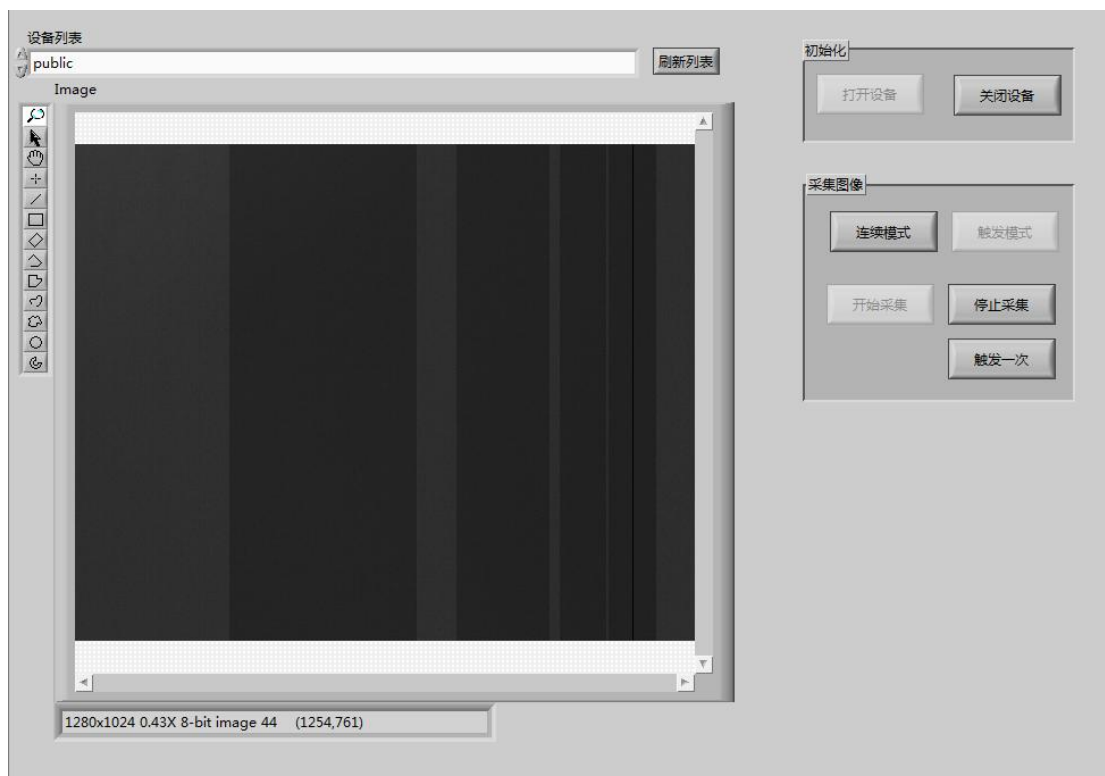
1.1.2 使用过程

点击【刷新列表】进行查找设备，这时设备列表会出现当前在线的设备列表，命名方式为用户 ID 不为空时显示设备名称，当设备名为空时，显示设备型号及序列号。

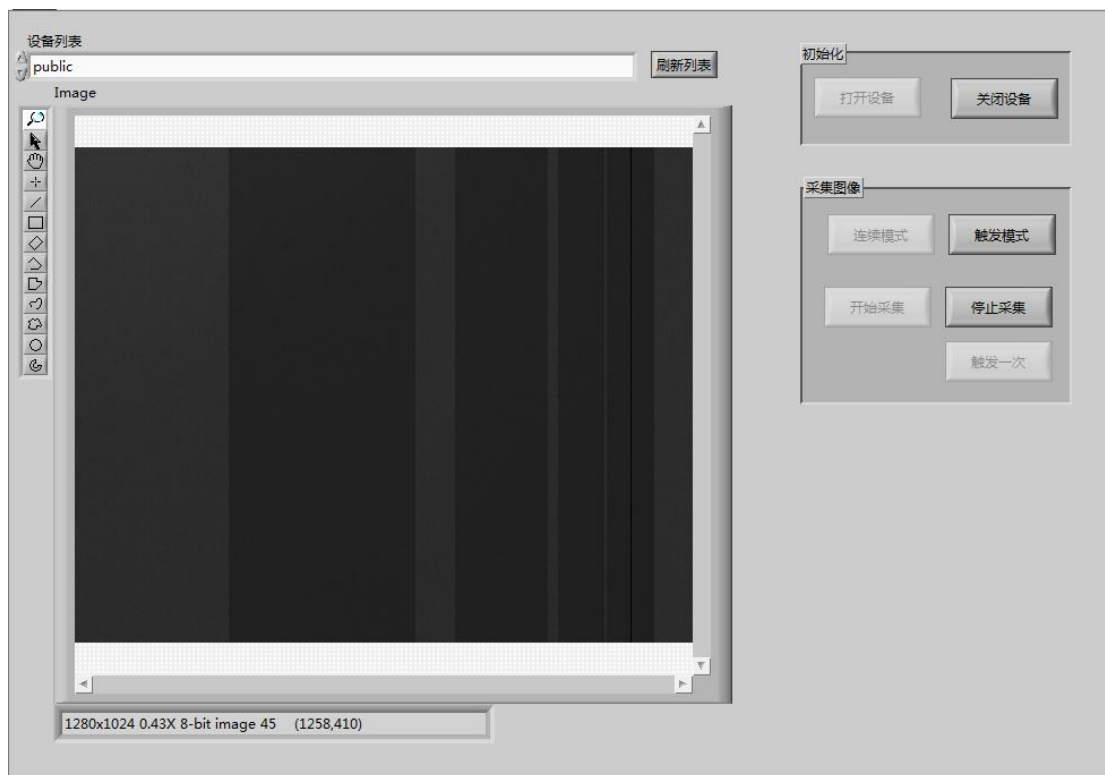
选择其中一个设备，点击【打开设备】打开当前选中的设备，默认以连续方式打开设备。
选择触发模式可以选中触发模式按钮。



在触发模式下，当点击【开始采集】后，同时【触发一次】也是可以点击从而完成触发一次功能



采用连续模式下，点击【开始采集】进行图像采集，左边的显示区域将会出现实时图像。



在使用过程中有任何异常或错误，都会以弹窗的形式出现提示，若没有任何提示，则认为一切正常地运行

1.2 Demo 软件开发步骤

1.2.1 Labview VI 介绍

为了方便用户在 Labview 平台调用 C 接口的 SDK，我们为 Labview 平台封装了子 VI。在 MvCameraLib.lvlib 中，一共有 20 个子 VI。其中 EnumDevices.vi、CreateHandle.vi、DestroyHandle.vi、OpenDevice.vi、CloseDevice.vi、StartGrabbing.vi、StopGrabbing.vi 为相机连接操作的 VI；SetValue、GetValue 等一些 VI 是设置相机参数的 VI，SaveImage.vi 是保存图片用 VI。这些 VI 是对 C 接口的 SDK 的二次封装，其内部调用了 C 接口 SDK 中的一些动态链接库。

以 EnumDevices.vi 为例。这个接口输入参数为：nLayerType 以及错误输入。nLayerType 为一个 32 位整型数据，传入枚举的设备类型，1 表示 GigE 的网口相机，4 表示 U3V 相机。输出参数分别为：function return、DeviceNum、pstGigEDevArray、pstU3VDevArray 以及错误输出。Function return 表示接口调用返回值，返回 0 表示调用成功，返回负数表示调用失败的错误码。DeviceNum 表示一共枚举到的相机数目。pstGigEDevArray、pstU3VDevArray 中分别返回 GigE 相机和 U3V 相机设备信息簇数组，设备信息簇定义如下：

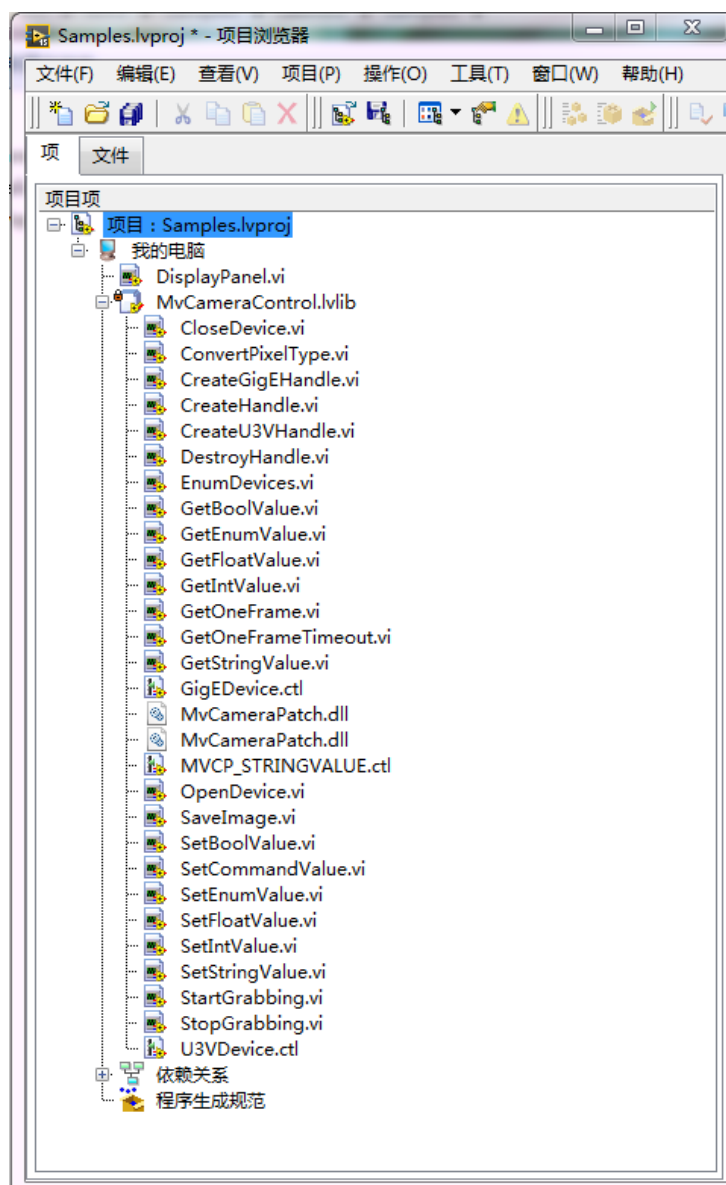
```
typedef struct _MV_CC_DEVICE_INFO_  
{  
    // common info
```

```
unsigned short    nMajorVer;
unsigned short    nMinorVer;
unsigned int      nMacAddrHigh;    // MAC 地址
unsigned int      nMacAddrLow;
unsigned int      nTLayerType;    // 设备传输层协议类型e.g. MV_GIGE_DEVICE
unsigned int      nReserved[4];
union
{
    MV_GIGE_DEVICE_INFO stGigEInfo;
    MV_USB3_DEVICE_INFO stUsb3VInfo;
    // more ...
} SpecialInfo;

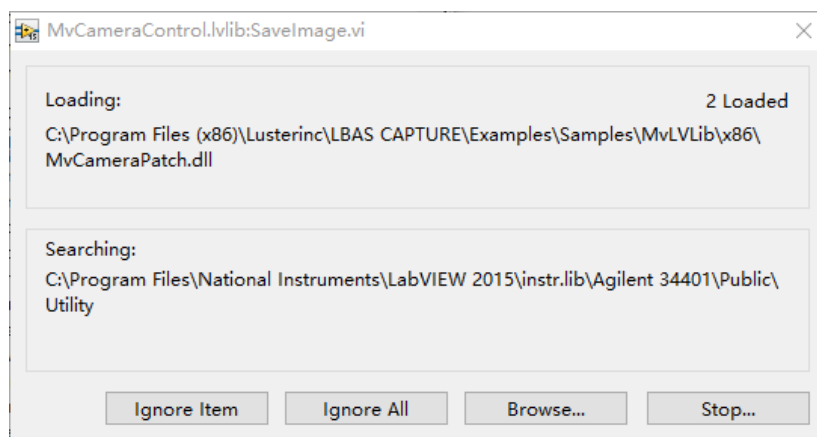
} MV_CC_DEVICE_INFO;
```

1.2.2 Labview Demo 开发

首先新建一个项目，在项目管理器中，右击我的电脑，选择添加文件，将我们的 MvCameraLib.lvlib 加入到项目中。新建一个 VI，例如命名为 DisplayPanel.vi。然后在 vi 的程序框图面板就可以添加 MvCameraLib.lvlib 中的 vi 进行相机操作。



要注意的是：Labview 在加载 vi 初始化过程中，会出现以下界面：



该界面会自动搜索 VI 调用到的 MvCameraControl.dll 和 MvCameraPatch.dll，如果无法搜索到，请用户手动点击浏览添加 SDK 中的 MvCameraControl.dll 和 MvCameraPatch.dll 路径。

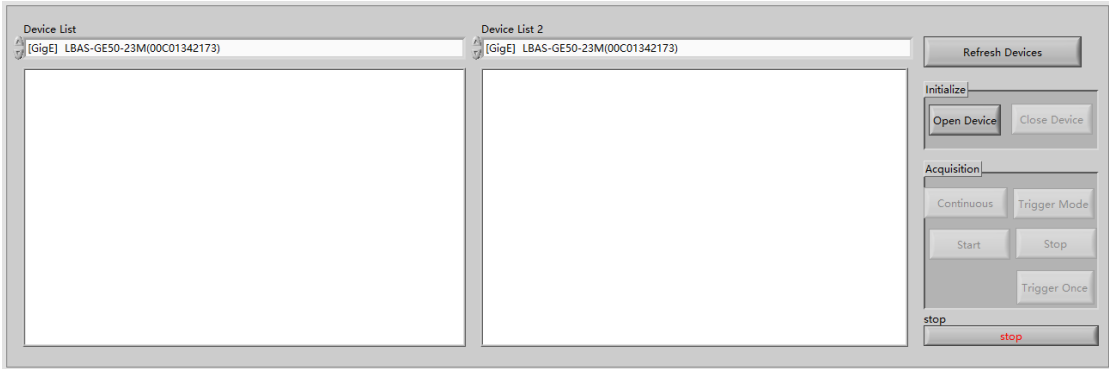
二. TwoCameraSamples 使用步骤及开发流程

TwoCameraSamples 是一个 Labview 双相机的示例程序,可以同时操作两个不同的相机。

2.1 Demo 软件使用步骤

2.1.1 界面总体

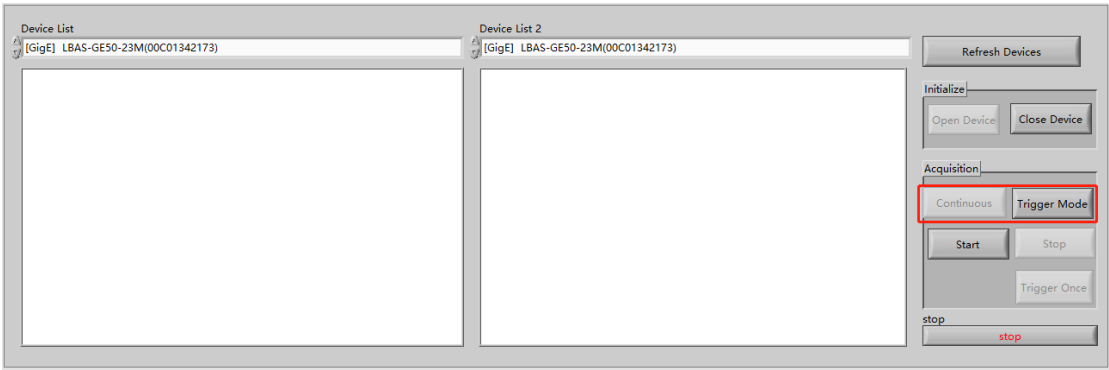
软件界面总览,一共包括四个控制模块(刷新设备,初始化,图像采集,停止运行)、两个下拉设备列表和两个图像显示区域



2.1.2 使用过程

点击【刷新列表】进行查找设备,这时设备列表会出现当前在线的设备列表,命名方式为用户 ID 不为空时显示设备名称,当设备名为空时,显示设备型号及序列号。

分别选择设备列表中两个不同的设备,点击【打开设备】打开当前选中的设备,默认以连续方式打开设备。选择触发模式可以选中触发模式按键。



在触发模式下,当点击【开始采集】后,同时【触发一次】也是可以点击从而完成触发一次功能。

采用连续模式下,点击【开始采集】进行图像采集,左边的显示区域将会出现实时图像。

在使用过程中有任何异常或错误,都会以弹窗的形式出现提示,若没有任何提示,则认为一切正常地运行

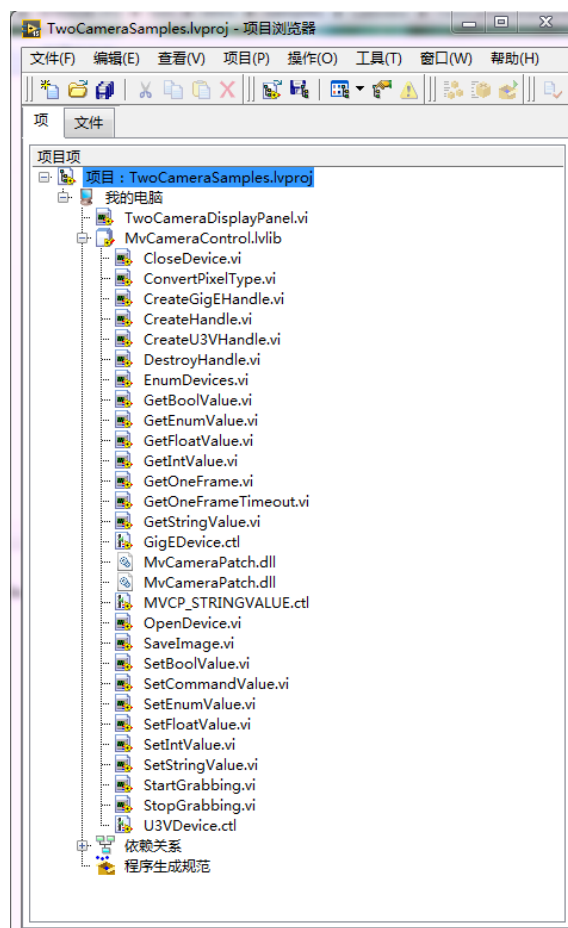
2.2 Demo 软件开发步骤

2.2.1 Labview VI 介绍

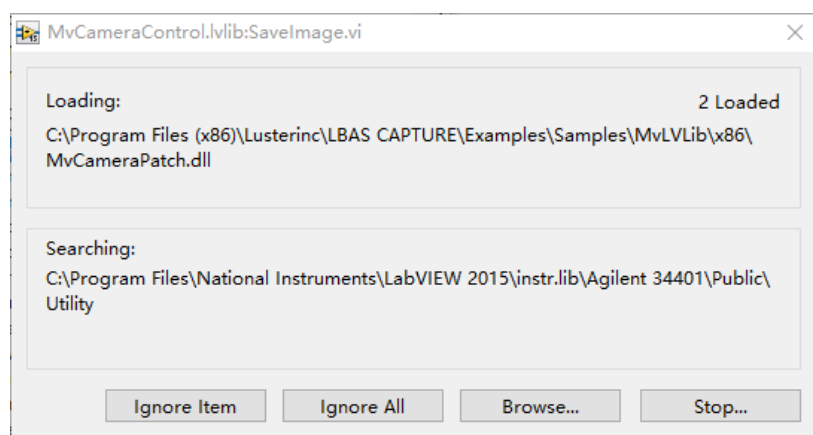
同 1.2.1。

2.2.2 Labview Demo 开发

首先新建一个项目，在项目管理器中，右击我的电脑，选择添加文件，将我们的 MvCameraLib.lvlib 加入到项目中。新建一个 VI，例如命名为 TwoCameraDisplayPanel.vi。然后在 vi 的程序框图面板就可以添加 MvCameraLib.lvlib 中的 vi 进行相机操作。



要注意的是：Labview 在加载 vi 初始化过程中，会出现以下界面：



该界面会自动搜索 VI 调用到的 MvCameraControl.dll 和 MvCameraPatch.dll，如果无法搜索到，请用户手动点击浏览添加 SDK 中的 MvCameraControl.dll 和 MvCameraPatch.dll 路径。