

LabVIEW 接口开发说明书

版本：V1.0.1

发布日期：2020-05-19

本手册中所提及的其它软硬件产品的商标与名称，都属于相应公司所有。

本手册的版权属于中国大恒（集团）有限公司北京图像视觉技术分公司所有。未得到本公司的正式许可，任何组织或个人均不得以任何手段和形式对本手册内容进行复制或传播。

本手册的内容若有任何修改，恕不另行通知。

© 2020 中国大恒（集团）有限公司北京图像视觉技术分公司版权所有

网 站：<http://www.daheng-imaging.com>

销售信箱：sales@daheng-imaging.com

销售热线：010-82828878 转 8068

支持信箱：support@daheng-imaging.com

支持热线：400-999-7595

目录

1. 相机工作流程	1
1.1. 整体工作流程	1
1.2. 功能控制流程	2
2. 编程指引	3
2.1. 搭建编程环境	3
2.1.1. LabVIEW VI 介绍	3
2.1.2. LabVIEW 程序开发	3
2.2. 快速上手	4
2.2.1. 初始化和反初始化	4
2.2.2. 枚举设备	4
2.2.3. 打开关闭设备	5
2.2.4. 采集控制	5
2.2.5. 相机控制	6
2.2.5.1. 属性控制器种类	6
2.2.5.2. 属性数据类型	7
2.2.6. 错误处理	9
3. 模块接口定义	11
3.1. Init.vi	11
3.2. UnInit.vi	11
3.3. UpdateDeviceList.vi	11
3.4. OpenDeviceBySN.vi	11
3.5. CloseDevice.vi	12
3.6. OpenStream.vi	12
3.7. CloseStream.vi	12
3.8. IsImplemented.vi	12
3.9. GetBoolValue.vi	13
3.10. SetBoolValue.vi	13
3.11. GetIntValue.vi	13
3.12. SetIntValue.vi	14
3.13. GetFloatValue.vi	14
3.14. SetFloatValue.vi	14
3.15. GetEnumValue.vi	14

3.16. SetEnumValue.vi	15
3.17. GetStringValue.vi	15
3.18. SetStringValue.vi	15
3.19. SetCommandValue.vi	16
3.20. StartGrab.vi	16
3.21. StopGrab.vi	16
3.22. GetImage.vi	16
3.23. ImageGrabEvent.vi	17
3.24. GrabImageDisplay.vi	17
3.25. SaveImage.vi	17
4. 常见问题解答	18
5. 版本说明	19

1. 相机工作流程

1.1. 整体工作流程

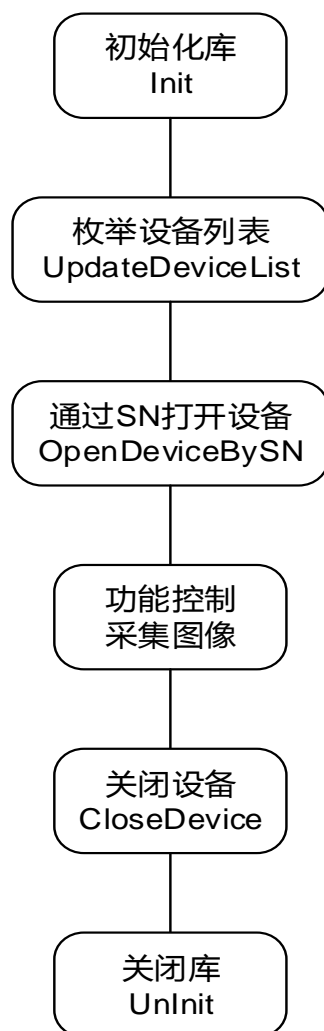


图 1-1 整体工作流程

1.2. 功能控制流程

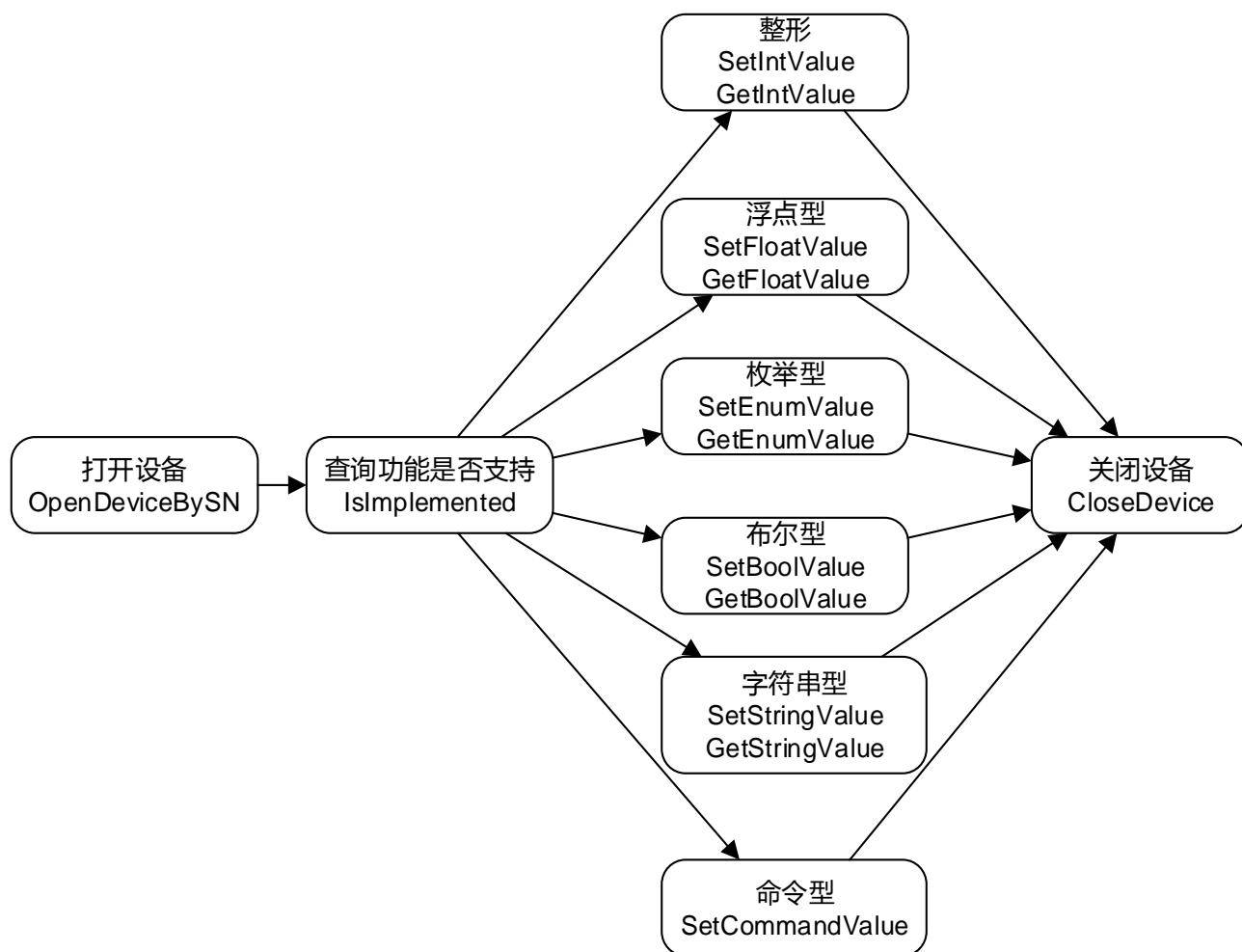


图 1-2 功能控制流程

2. 编程指引

2.1. 搭建编程环境

支持 LabVIEW2013 及以上版本，使用时除安装 LabVIEW 程序外还需安装对应版本的 VISION 视觉库。

2.1.1. LabVIEW VI 介绍

为了方便用户在 LabVIEW 平台调用.NET 接口的 SDK，我们在 LabVIEW 平台封装了子 VI。在 GxLVLib.lvlib 中，一共有 25 个子 VI。这些 VI 是对部分.NET 接口的 SDK 的二次封装，其内部调用了.NET 接口 SDK 中的一些动态链接库，用户也可以在 LabVIEW 中直接调用.NET 接口。

2.1.2. LabVIEW 程序开发

首先新建一个项目，在项目管理器中，右击**我的电脑（My Computer）**，选择**添加»文件**，将我们的 GxLVLib.lvlib 添加到项目中。新建一个 VI，命名为 Demo.vi。然后打开 Demo.vi，在其程序框图面板就可以添加 GxLVLib.lvlib 中的 VI 进行相机操作。

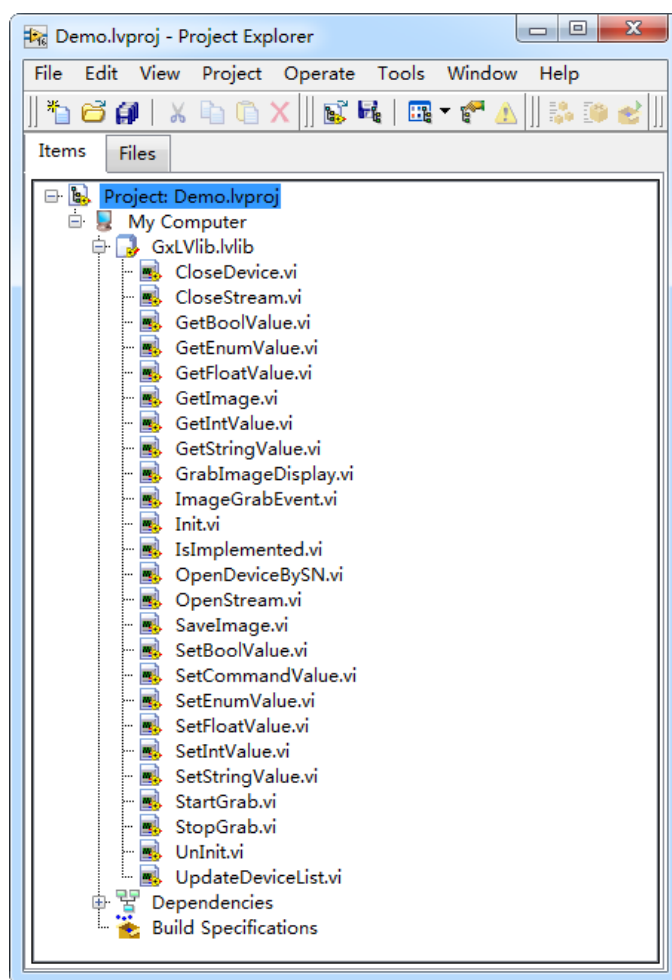


图 2-1 VI 接口列表

注意：LabVIEW 在加载 VI 初始化过程中，可能会出现图 2-2 界面：

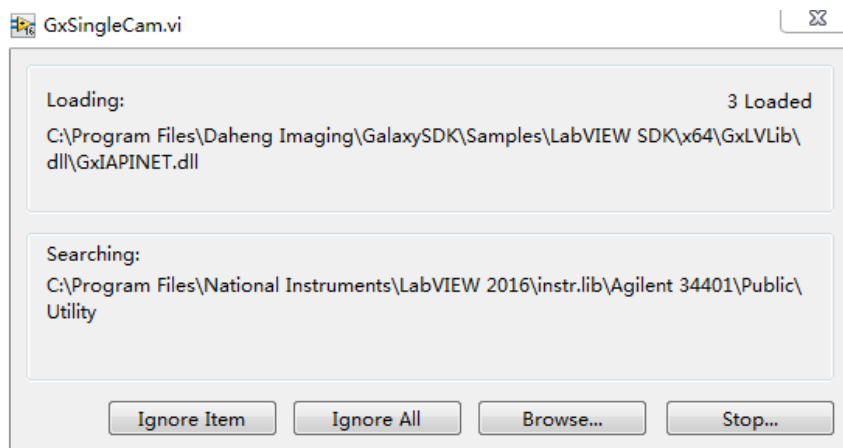


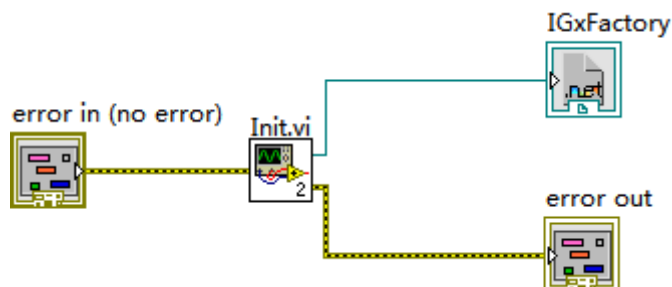
图 2-2 加载 VI 界面

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

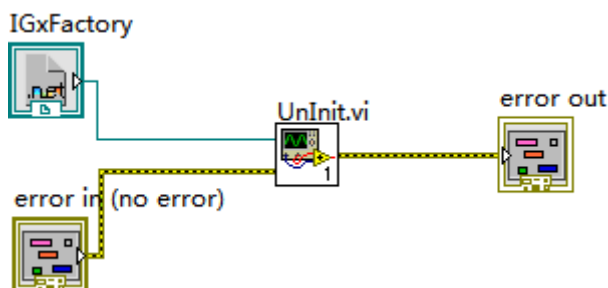
2.2. 快速上手

2.2.1. 初始化和反初始化

GxLVLib.lvlib 为.NET 接口库的二次封装，而 GxIAPINET 在使用时必须执行初始化和反初始化。在调用其他接口之前必须调用 Init.vi 执行初始化。

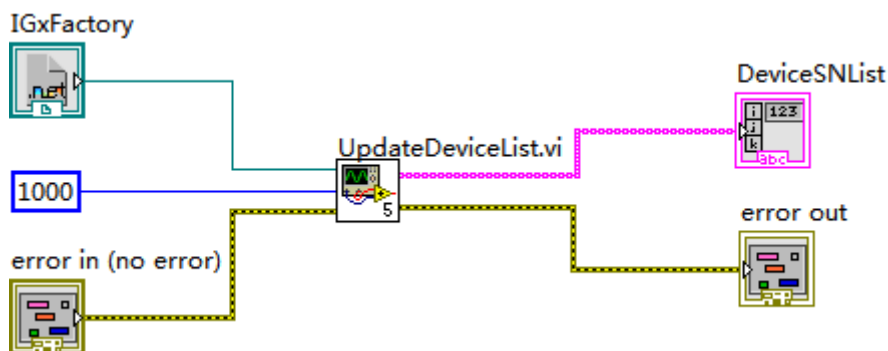


在进程退出之前必须调用 UnInit.vi 释放 GxIAPINET 申请的所有资源。



2.2.2. 枚举设备

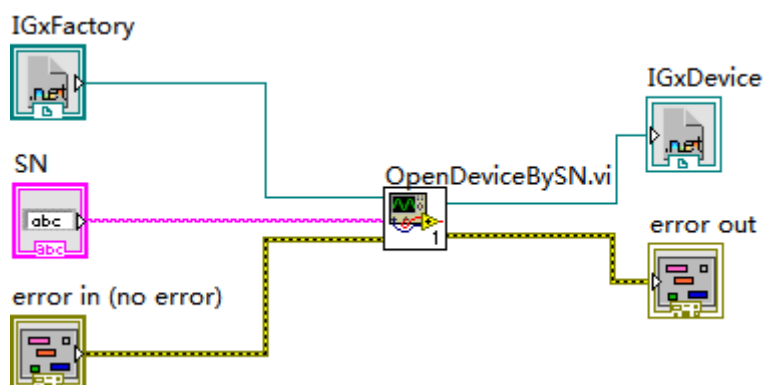
用户通过调用 UpdateDeviceList.vi 枚举当前所有可用设备，函数返回值为设备序列号列表 DeviceSNList。设备信息列表的元素个数为枚举到的设备个数，列表中元素的数据类型为 string。



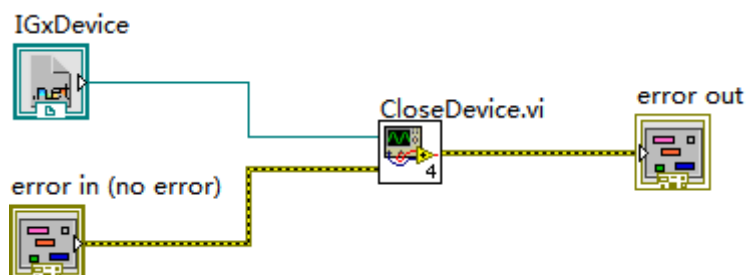
2.2.3. 打开关闭设备

用户可通过 OpenDeviceBySN.vi 打开设备，函数的输入参数为设备序列号，类型为 string。

用户在打开设备前应先调用枚举设备接口，更新 GxIAPINET 库内部的设备列表，否则可能打开设备失败。



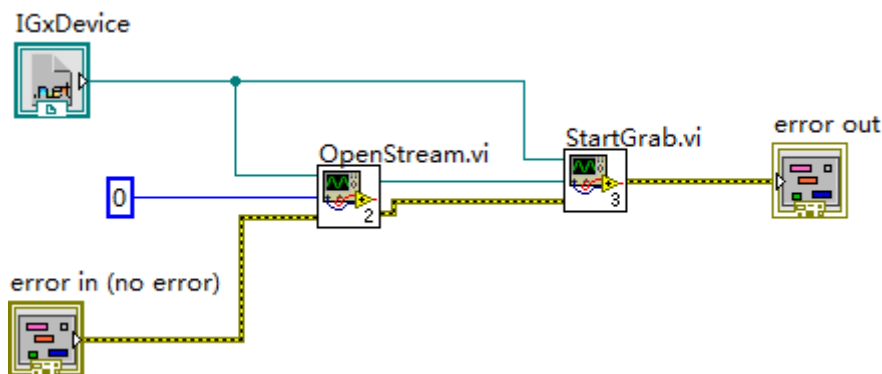
用户可以调用 CloseDevice.vi 来关闭设备，释放所有设备资源。



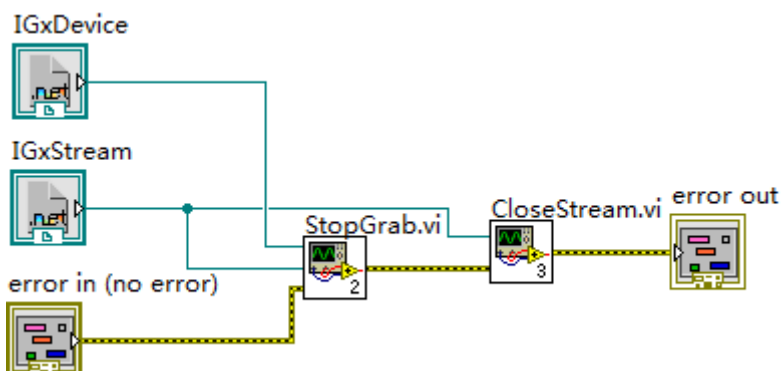
2.2.4. 采集控制

在设备正常开启并设置好相机采集参数后，用户可调用 StartGrab.vi 和 StopGrab.vi 执行开停采。

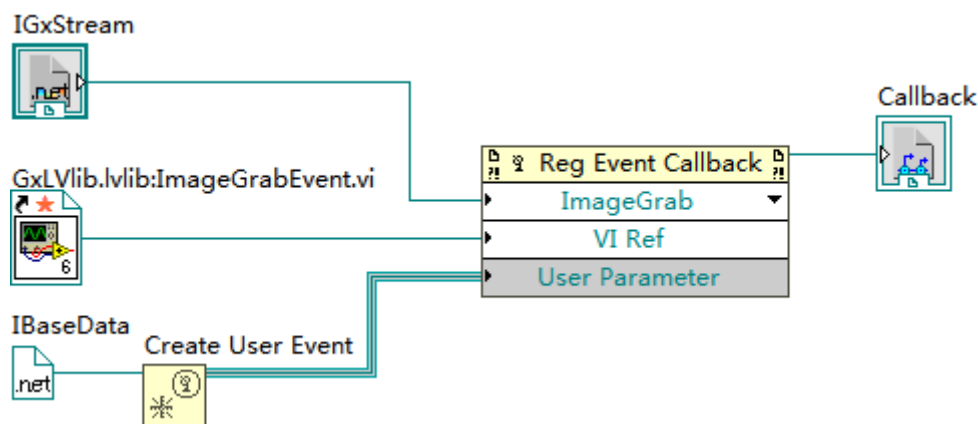
执行开采时，需要先打开流通道数据流，目前仅支持第 0 通道数据流，然后执行 StartGrab.vi 即可。



执行停采时，需要先执行 StopGrab.vi，然后关闭流通道。



用户可通过 GetImage 方式获取图像，也可通过回调方式获取图像。若采用回调方式，在开采前通过 Register Event Callback 注册回调事件即可。



2.2.5. 相机控制

2.2.5.1. 属性控制器种类

分三种属性控制器，如下：

远端设备属性控制器，比如宽高、曝光、增益等。我们提供的 VI 可实现远端设备属性控制器的访问和控制。

本地属性控制器，不同类型的设备具备的功能也不一样。用户可直接使用 IGxAPINET 库的 IGXDevice::GetFeatureControl 函数进行控制。

流对象属性控制器，关于采集控制和采集数据统计的属性访问控制器。用户可直接使用 IGxAPINET 库的 IGXStream::GetFeatureControl 函数进行控制。

2.2.5.2. 属性数据类型

VI 接口中共实现了 6 种数据类型，如下：

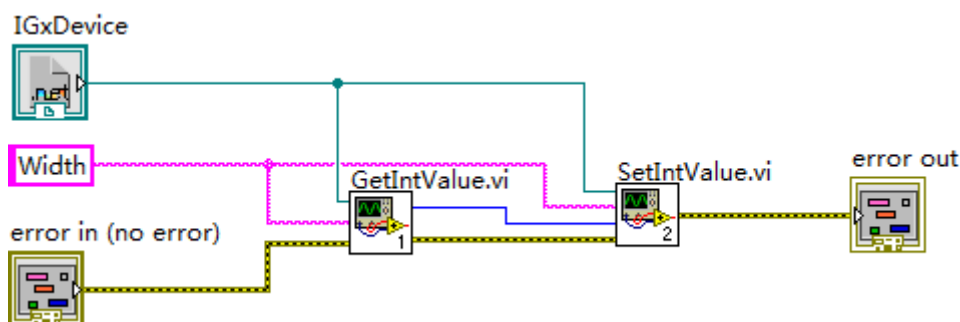
整型：

相关接口：

SetIntValue.vi //设置

GetIntValue.vi //读取

代码样例：



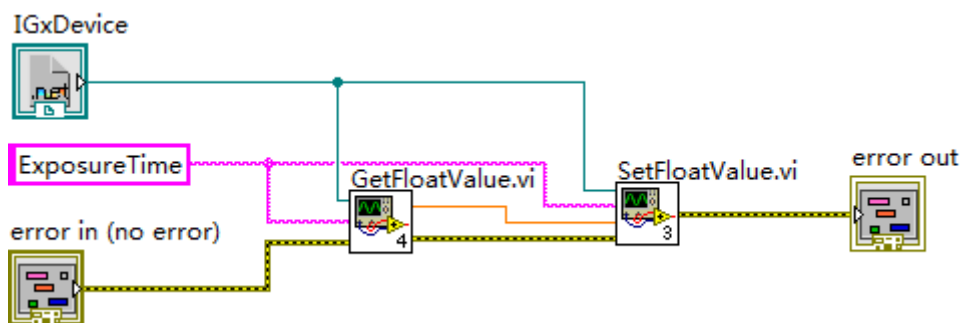
浮点型：

相关接口：

SetFloatValue.vi //设置

GetFloatValue.vi //读取

代码样例：



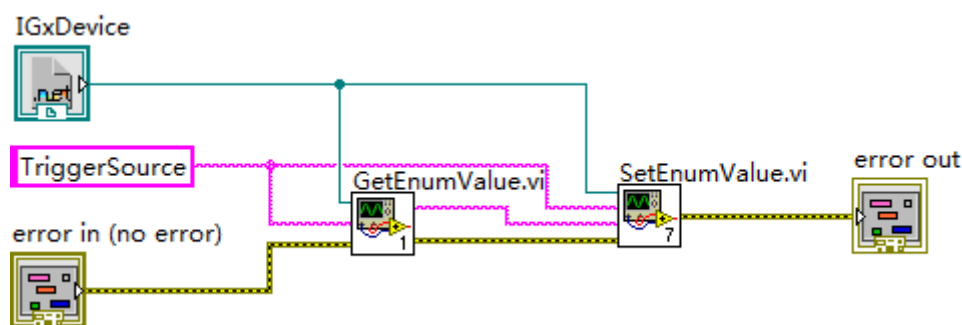
枚举型：

相关接口：

SetEnumValue.vi //设置

GetEnumValue.vi //读取

代码样例：



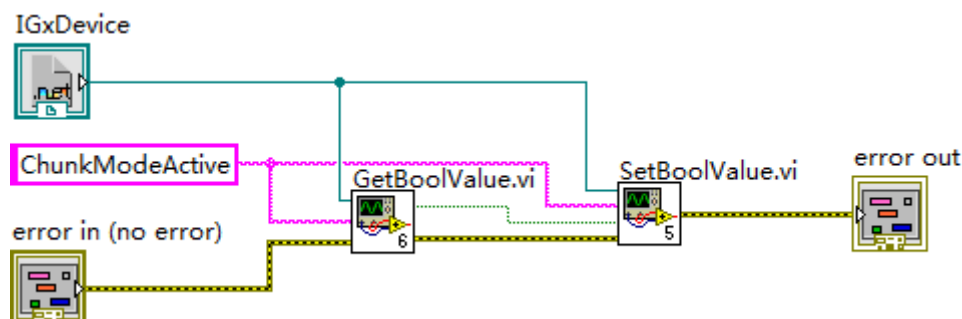
布尔型：

相关接口：

SetBoolValue.vi //设置

GetBoolValue.vi //读取

代码样例：



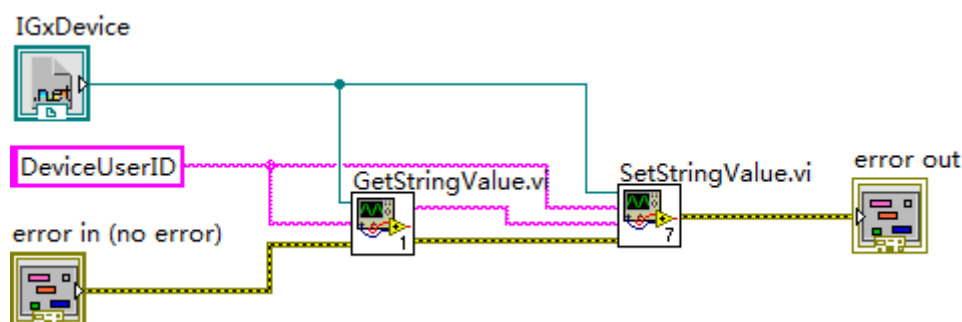
字符串型：

相关接口：

SetStringValue.vi //设置

GetStringValue.vi //读取

代码样例：

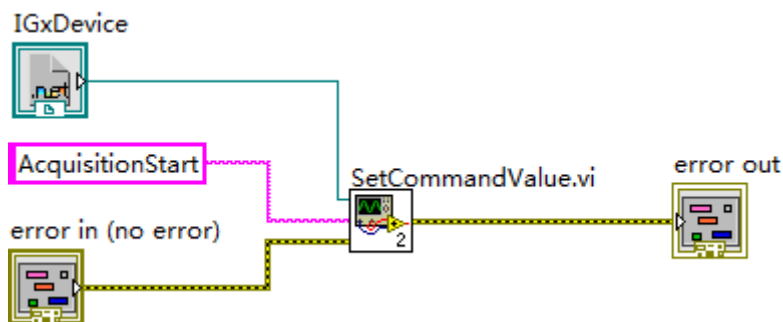


命令类型：

相关接口：

SetCommandValue.vi //设置

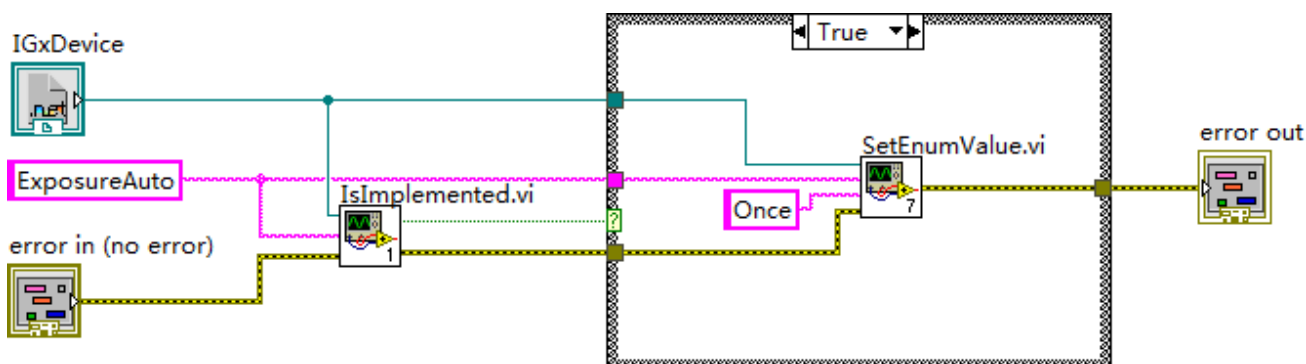
代码样例：



建议用户在操作属性参数前，先查询属性的功能是否实现：

查询属性功能是否实现的相关接口：

IsImplemented.vi //是否支持



2.2.6. 错误处理

LabVIEW 的错误信息由错误输入(error in)以及错误输出(error out)两个错误簇控件来表示，该簇是由一个表示状态的布尔变量(status)、一个表示错误代码的 32 位整型变量(code)以及一个说明错误原因的字符串变量(source)组成的，错误簇用于传递 LabVIEW 代码运行期间的出错信息。

用户可以按照如下步骤使用**错误代码编辑器**自定义错误代码。

- 1) 选择**工具»高级»编辑错误代码**，打开**错误代码编辑器**。
- 2) 在出现的提示中，单击**新建**按钮，创建新错误代码文件或单击**已有**按钮，找到错误代码文件。
- 3) 在**本文件注释**文本栏中输入关于错误代码文件的注释。
- 4) 单击错误代码进行编辑，选择文本栏下的空白为错误添加文本说明。单击**添加错误代码**按钮，向错误代码文件添加另一个错误代码。
- 5) 若要编辑错误代码和说明，选择要修改的错误代码或文本，然后输入修改内容。选择**错误代码排序**按钮，以升序排列错误代码。如要删除一个错误代码，选择该错误代码并按**删除所选错误代码**按钮。
- 6) 错误代码文件编辑完成后，单击**保存**按钮，将错误代码文件保存至 labview\user.lib\errors 目录。
错误代码文件必须保持为 xxx-errors.txt，其中，xxx 是用户自定义的名称。

我们提供的错误代码文件为 GxLVLib-errors.txt，放置在\GxLVLib\res 目录下，用户使用时需拷贝至 labview\user.lib\errors 目录。

目前提供了 2 个错误类型，用户可以自行添加和修改：

错误代码	意义
5000	Invalid param
5001	Not found device

表 2-1 错误代码列表

GxIAPINET 库报错时以抛异常的形式表示，LabVIEW 会直接弹框提示异常信息，如图 2-3 所示：

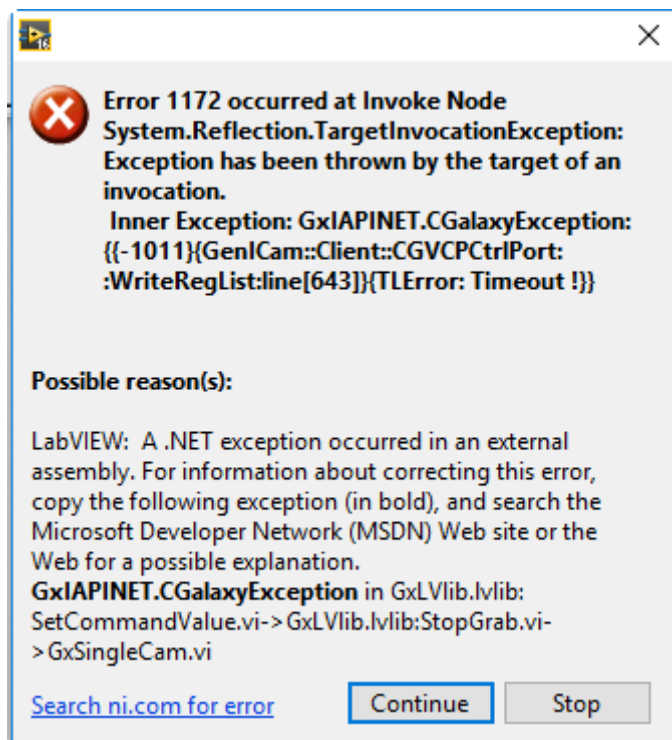


图 2-3 错误提示

3. 模块接口定义

3.1. Init.vi

意义：

初始化库资源

参数：

error in：错误输入

IGxFactory：IGXFactory 对象实例

error out：错误输出

3.2. UnInit.vi

意义：

释放库资源

参数：

IGxFactory：IGXFactory 对象实例

error in：错误输入

error out：错误输出

3.3. UpdateDeviceList.vi

意义：

枚举设备（对于千兆网设备是子网枚举）

参数：

IGxFactory：IGXFactory 对象实例

Time out：枚举超时时间，单位 ms

error in：错误输入

DeviceSNList：设备 SN 列表

error out：错误输出

3.4. OpenDeviceBySN.vi

意义：

通过序列号打开设备

参数：

IGxFactory：IGXFactory 对象实例

SN：设备序列号

error in : 错误输入

IGxDevice : IGxDevice 对象实例

error out : 错误输出

3.5. CloseDevice.vi

意义 :

关闭设备

参数 :

IGxDevice : IGxDevice 对象实例

error in : 错误输入

error out : 错误输出

3.6. OpenStream.vi

意义 :

用户指定流通道序号打开某个流, 获取流通道对象

参数 :

IGxDevice : IGxDevice 对象实例

StreamID : 流通道序号

error in : 错误输入

IGxStream : IGxStream 对象实例

error out : 错误输出

3.7. CloseStream.vi

意义 :

关闭流通道

参数 :

IGxStream : IGxStream 对象实例

error in : 错误输入

error out : 错误输出

3.8. IsImplemented.vi

意义 :

查询当前属性控制是否支持该功能

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

error in : 错误输入

IsImplement : 是否支持

error out : 错误输出

3.9. GetBoolValue.vi

意义 :

获取布尔类型值

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

error in : 错误输入

Bool Value : Bool 类型值

error out : 错误输出

3.10. SetBoolValue.vi

意义 :

设置布尔类型值

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

Bool Value : Bool 类型值

error in : 错误输入

error out : 错误输出

3.11. GetIntValue.vi

意义 :

获取整型类型值

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

error in : 错误输入

Int Value : Int 类型值

error out : 错误输出

3.12. SetIntValue.vi

意义：

设置整型类型值

参数：

IGxDevice：IGxDevice 对象实例

Feature Name：属性名称

Int Value：Int 类型值

error in：错误输入

error out：错误输出

3.13. GetFloatValue.vi

意义：

获取浮点型类型值

参数：

IGxDevice：IGxDevice 对象实例

Feature Name：属性名称

error in：错误输入

Float Value：Float 类型值

error out：错误输出

3.14. SetFloatValue.vi

意义：

设置浮点型类型值

参数：

IGxDevice：IGxDevice 对象实例

Feature Name：属性名称

Float Value：Float 类型值

error in：错误输入

error out：错误输出

3.15. GetEnumValue.vi

意义：

获取枚举型类型值

参数：

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

error in : 错误输入

Enum Value : Enum 类型值

error out : 错误输出

3.16. SetEnumValue.vi

意义 :

设置枚举型类型值

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

Enum Value : Enum 类型值

error in : 错误输入

error out : 错误输出

3.17. GetStringValue.vi

意义 :

获取字符串型类型值

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

error in : 错误输入

String Value : String 类型值

error out : 错误输出

3.18. SetStringValue.vi

意义 :

设置字符串型类型值

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

String Value : String 类型值

error in : 错误输入

error out : 错误输出

3.19. SetCommandValue.vi

意义 :

执行命令

参数 :

IGxDevice : IGxDevice 对象实例

Feature Name : 属性名称

error in : 错误输入

error out : 错误输出

3.20. StartGrab.vi

意义 :

开始采集

参数 :

IGxDevice : IGxDevice 对象实例

IGxStream : IGxStream 对象实例

error in : 错误输入

error out : 错误输出

3.21. StopGrab.vi

意义 :

停止采集

参数 :

IGxDevice : IGxDevice 对象实例

IGxStream : IGxStream 对象实例

error in : 错误输入

error out : 错误输出

3.22. GetImage.vi

意义 :

在开始采集之后, 通过此接口可以直接获取图像, 注意此接口不能与回调采集方式混用

参数 :

IGxStream : IGxStream 对象实例

Time out : 超时时间, 单位 ms

error in : 错误输入

ImageData : 获取到的图像数据对象

error out : 错误输出

3.23. ImageGrabEvent.vi

意义 :

在回调采集方式下, 通过此接口触发回调事件

参数 :

Event Common Data : 事件通用参数

Callback Param : GxIAPINET 库输出的图像数据等信息

.Control ref : 系统对象

Callback Event : 触发的 LabVIEW 回调事件

error out : 错误输出

3.24. GrabImageDisplay.vi

意义 :

显示图像

参数 :

IBaseData : IBaseData 图像数据对象实例

Image Src : 为存储图像数据开辟空间

error in : 错误输入

Image out : 输出图像为 IMAQImage 格式

error out : 错误输出

3.25. SaveImage.vi

意义 :

保存图像到指定路径

参数 :

Image Src : 要保存的 IMAQImage 图像数据

Path : 保存路径

error in : 错误输入

error out : 错误输出

4. 常见问题解答

问题 1：LabVIEW 加载 VI 过程中，会弹出搜索 dll 界面。

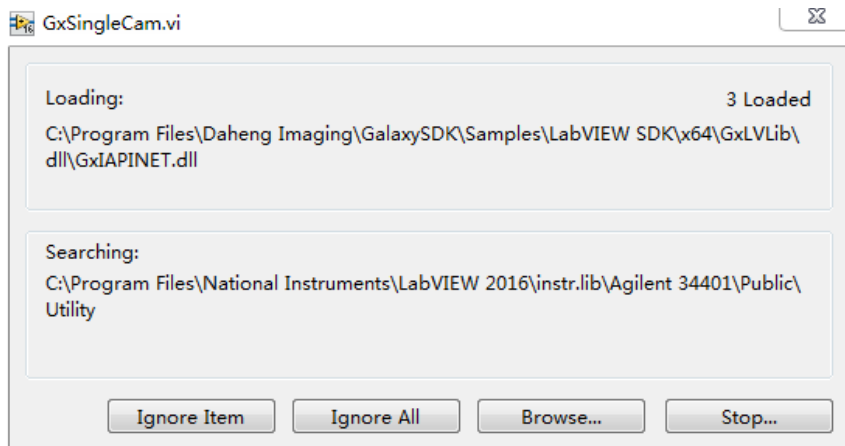


图 4-1 加载 VI 界面

解决方法：

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

问题 2：在浏览选择 GxIAPINET.dll 路径之后，提示加载程序集失败。

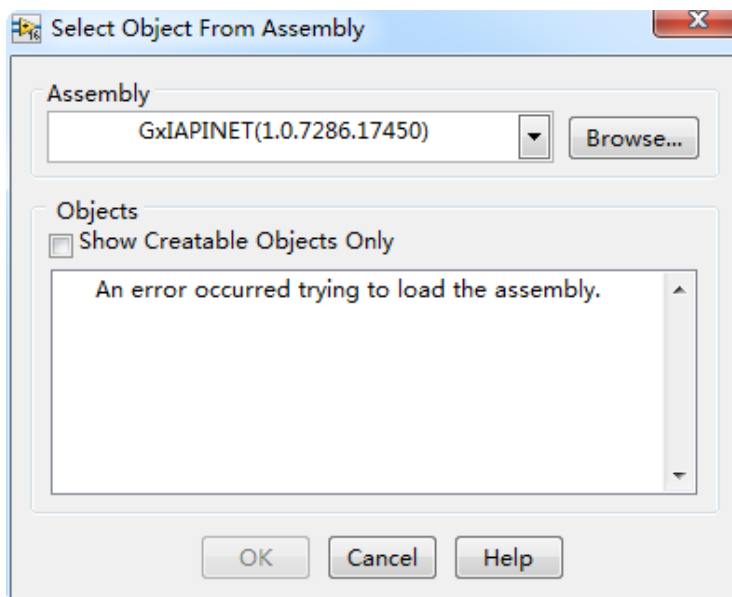


图 4-2 加载程序集失败界面

解决方法：

用户可能在使用 64 位 LabVIEW 时选择了 32 位 GxIAPINET 库，或在使用 32 位 LabVIEW 时选择了 64 位 GxIAPINET 库，选择相应版本的库即可。

5. 版本说明

序号	修订版本号	所做改动	发布日期
1	V1.0.0	初始发布	2020-01-19
2	V1.0.1	修改 4. 常见问题解答	2020-05-19