

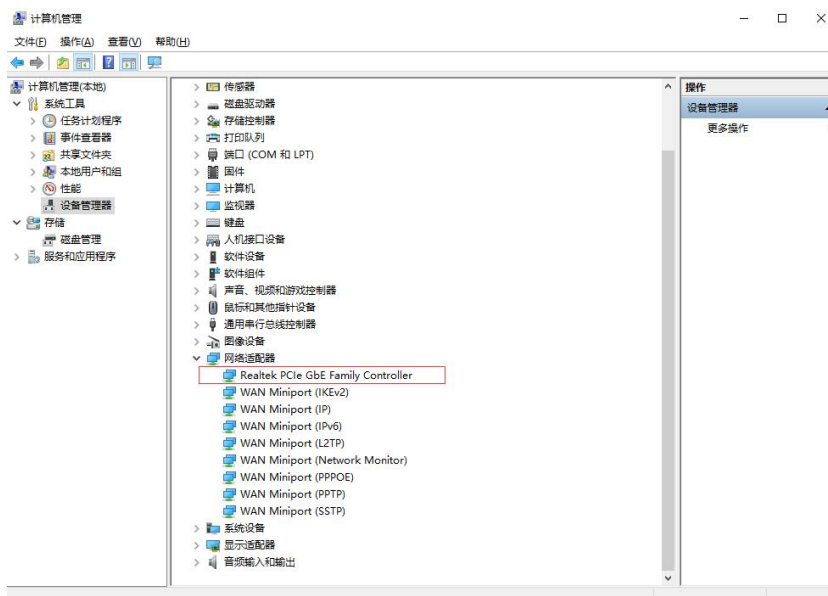
工业相机和电脑的连接

本手册描述工业相机和电脑的连接的具体方法。

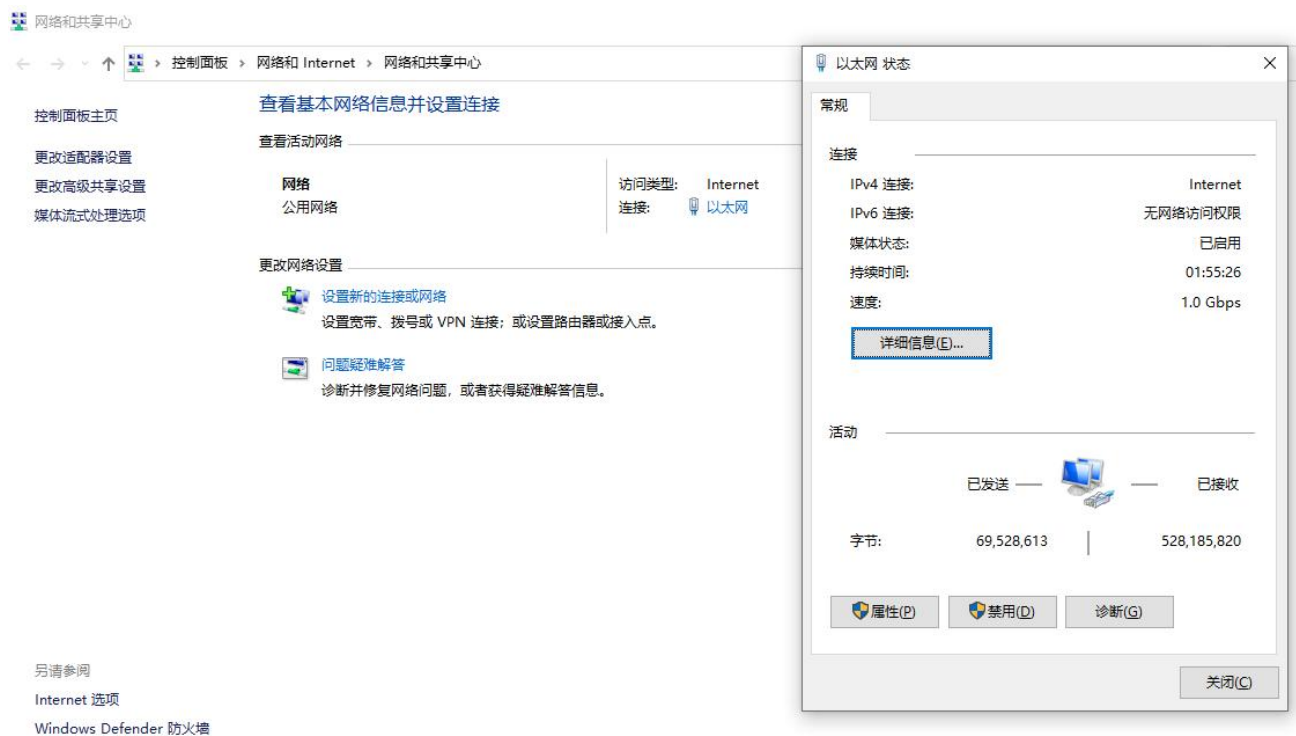
1 千兆网卡

千兆网卡即：1000Mbit/s 网卡，是根据网速从 10M/100M/1000M 自适应的网卡，也称为千兆以太网网卡。其传输速率可达 1000Mbps，根据端口区分有 1 口，2 口，4 口网卡，可以提供高速的连接，提高网络主干系统的响应速度。

1) 在电脑管理网络适配器中查看，安装专有 Realtek PCIe GbE Family Controller 驱动。



2) 打开控制面板->网络和 Internet->网络共享中心，查看其属性，选择设置巨帧提高性能，还能兼容百兆网，其中速度约位 70Mbps。



网卡如图所示:

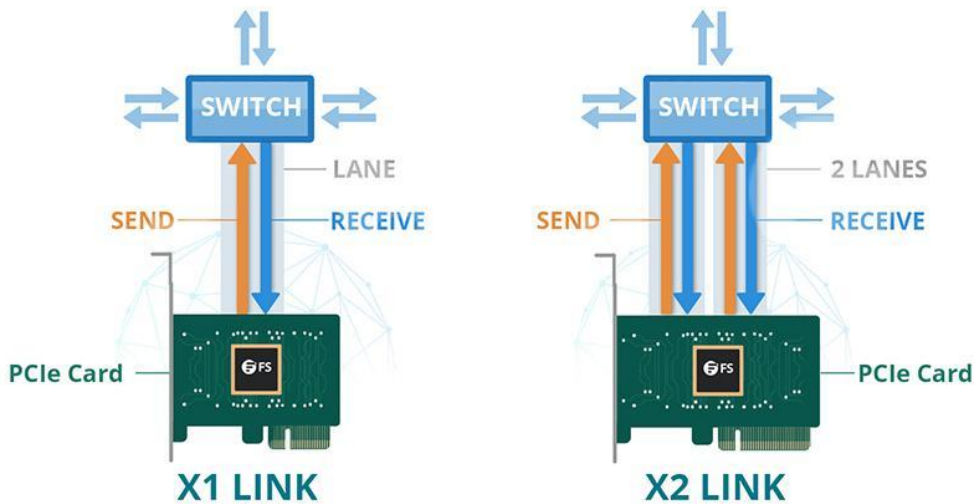
2 PCIE 网卡

PCIE 卡是一种具有 PCIE 接口的网卡,在主板级连接中用作扩展端口。PCIE 卡又分为 POE 卡和非 POE 卡。POE 指的是在现有的以太网的布线基础架构不作任何改动的情况下,在为一一些基于 IP 的终端的传输数据信号的同时,还能为此类设备提供直流供电的技术。非 POE 卡即是普通的网卡,不具备供电技术。

在实际中,常使用的 PCIE 卡的规格有 x1、x2、x4。其中 x1PCIE 网卡插好之后,插槽和网卡之间将形成逻辑连接,以便互相通信。这种逻辑连接称为互连或链路,它支持两个 PCIE 端口之间的点对点通信通道,并允许它们发送和接收普通的 PCI 请求或中断。如下图所示。在

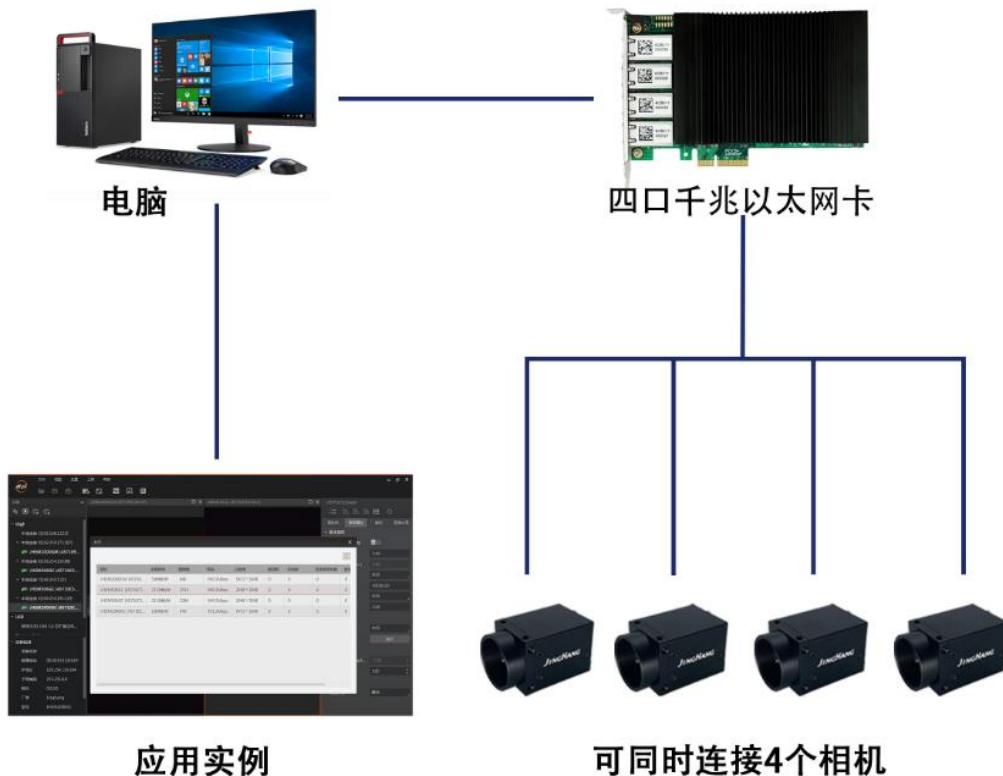
x2 链路中，每条通道包含两对不同的数据传输组，一对用于传输数据，另一对用于接收数据。因此，每条通道包含了四根电线或信号线路组成，x4 以此类推。

PCI EXPRESS LINKS AND LANES



<https://blog.csdn.net/FS>

多网口相机连接实物图如下图所示：



Tel: 0755-21505130 Website: <https://www.jinghangtech.cn>

PCIE 卡版本都是向后兼容的,也就是说任何版本的 PCIE 卡和主板都可以以最低版本的模式工作。下表展示了 PCIE 版本之间的传输速率比较。

版本	传输速率 (吞吐量, x1)	线路码
PCIE 1.0	2.5 GT/s (250 MB/s)	8b
PCIE 2.0	5.0 GT/s (500 MB/s)	8b
PCIE 4.0	16.0 GT/s (1969 MB/s)	128b

3 千兆交换机

千兆交换机是一种端口可以支持 1000Mbps 或 10/100/1000Mbps 速率的交换机。千兆交换机具有组网灵活的特点,可以提供全千兆接入以及增强的万兆上行端口扩展能力,此外,还能够实现机架内、机架间以及跨区域超长距的高性能堆叠。

千兆网交换机分为 POE 和非 POE 交换机。“POE 交换机”指的是在不改动任何现有的以太网布线基础的架构下为一些基于 IP 终端的工业相机传输数据的同时还能实现供电功能的设备,也可以理解为支持以太网供电的交换机。它的作用主要是数据传输和供电。非 POE 交换机就是普通的交换机,不具备供电的功能。千兆交换机具体连接多网口相机如下图所示:



4 万兆交换机

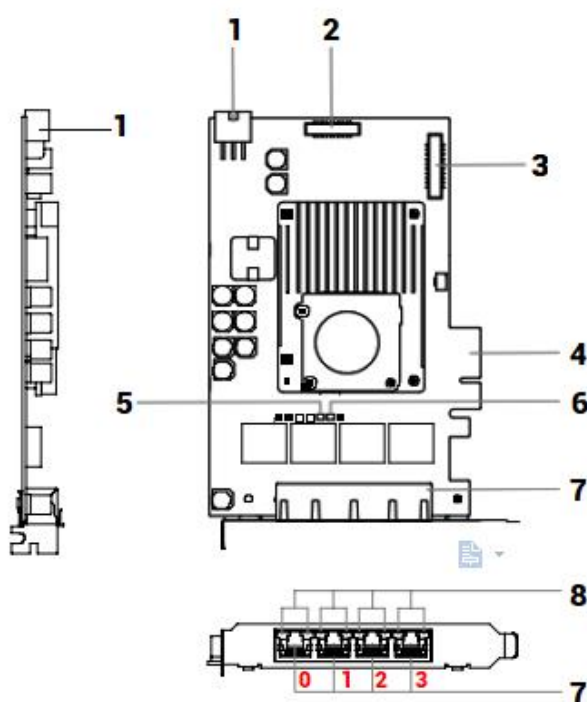
万兆交换机是为满足高性能、高可靠性要求而设计的新一代智能型路由交换机，主要面向企业网等大型网络，可帮助用户建设高性价比、高可靠性的优化网络。万兆以太网交换机也不仅仅是在已有的千兆以太网交换机上支持万兆的光模块，它从交换机体系结构、二/三层技术的更新和有效的带宽管理方面都做出了巨大改变，而且万兆交换机可支持最多 10 台全带宽工作。

5 采集卡

5.1 采集卡简介

GigE 采集卡可提供 4 路 RJ45 接口, 最多可接入 4 台相机。具备板载缓存, 可在卡端进行图像处理, 大幅度减轻 CPU 的负载。主机端采用 PCIE Gen2 × 4 总线接口, 可通过 DMA 直接映射于主机内存, 快速读取采集到的图像数据。

采集外观如下图所示, 具体接口介绍请见下表。



序号	接口名称	功能说明
1	ATX 供电接口	用于 POE 链路供电。当 RJ45 接口使用 POE 功能时, 需确保该接口接入主机的 6-pin ATX 12V 供电线缆。
2	I/O 接口(排母 1)	用于 I/O 信号连接, 可提供 4 路光耦隔离输入和 4 路光耦隔离输出、4 路 TTL 输入和 4 路 TTL 输出。

3	I/O 接口(排母 2)	用于 I/O 信号连接, 可提供 4 路 422 信号输入和 4 路 422 信号输出。
4	PCIE2.0 接口	支持 x4 模式的 PCIE2.0 接口, 与主机连接。
5	POE 输入指示灯	用于判断 POE 48V 电源是否正常。 <ul style="list-style-type: none"> ● 当指示灯常亮时, 表示 48V 电源正常; ● 当指示灯常灭时, 表示 48V 电源异常或未外接 6-pin ATX 12V 供电线缆。
6	总线指示灯	用于判断主机 PCIE 是否识别到采集卡。 <ul style="list-style-type: none"> ● 当指示灯常亮时, 表示 PCIE 识别到采集卡; ● 当指示灯常灭时, 表示 PCIE 未识别到采集卡。
7	千兆网口	采集卡共有 4 个 RJ45 接口, 即 0~3, 用于 4 路千兆网线链路连接, 支持 POE。 靠近 PCIE2.0 接口的 RJ45 接口为 3, 远离 PCIE2.0 接口的 RJ45 接口为 0。
8	通道指示灯	用于显示 4 个网口通路状态, 每个千兆网口含 2 个通道指示灯。 <ul style="list-style-type: none"> ● 绿灯常亮表示网络连接成功; ● 黄灯闪烁表示正在传输数据。

5.2 采集卡的使用

1) MVS 客户端安装

MVS 客户端支持安装在 Windows XP/7/10 32/64bit 操作系统上, MVS 客户端 V3.4.1 220324 及以上版本支持接入采集卡。

2) MVS 客户端操作

通过 MVS 客户端设备列表的 GenTL 可枚举采集卡和采集卡连接的相机, 可对采集卡

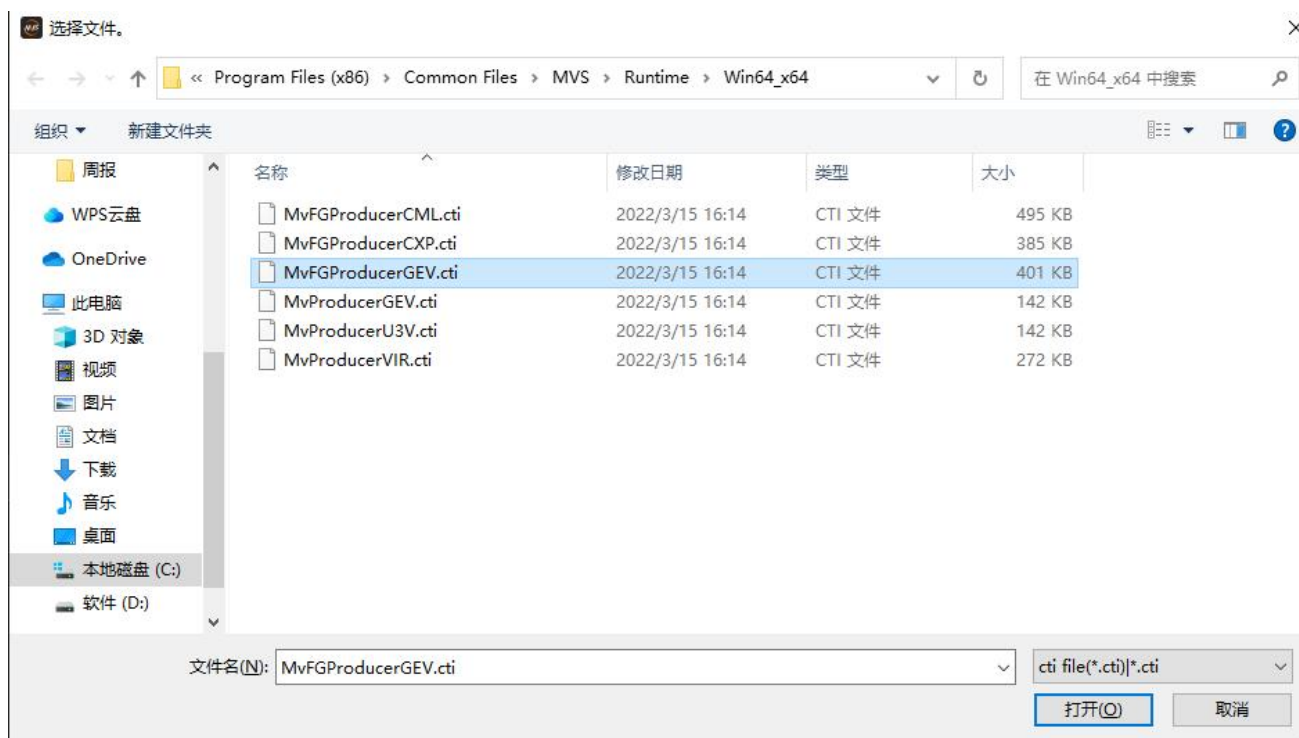
进行参数设置或固件升级等操作。

操作步骤

1. 设备列表选中 GenTL，右键单击选择 cti 文件，如下图所示。



2. 在弹出的界面中选择采集卡的 MvFGProducerGEV.cti 文件，如下图所示，单击打开。

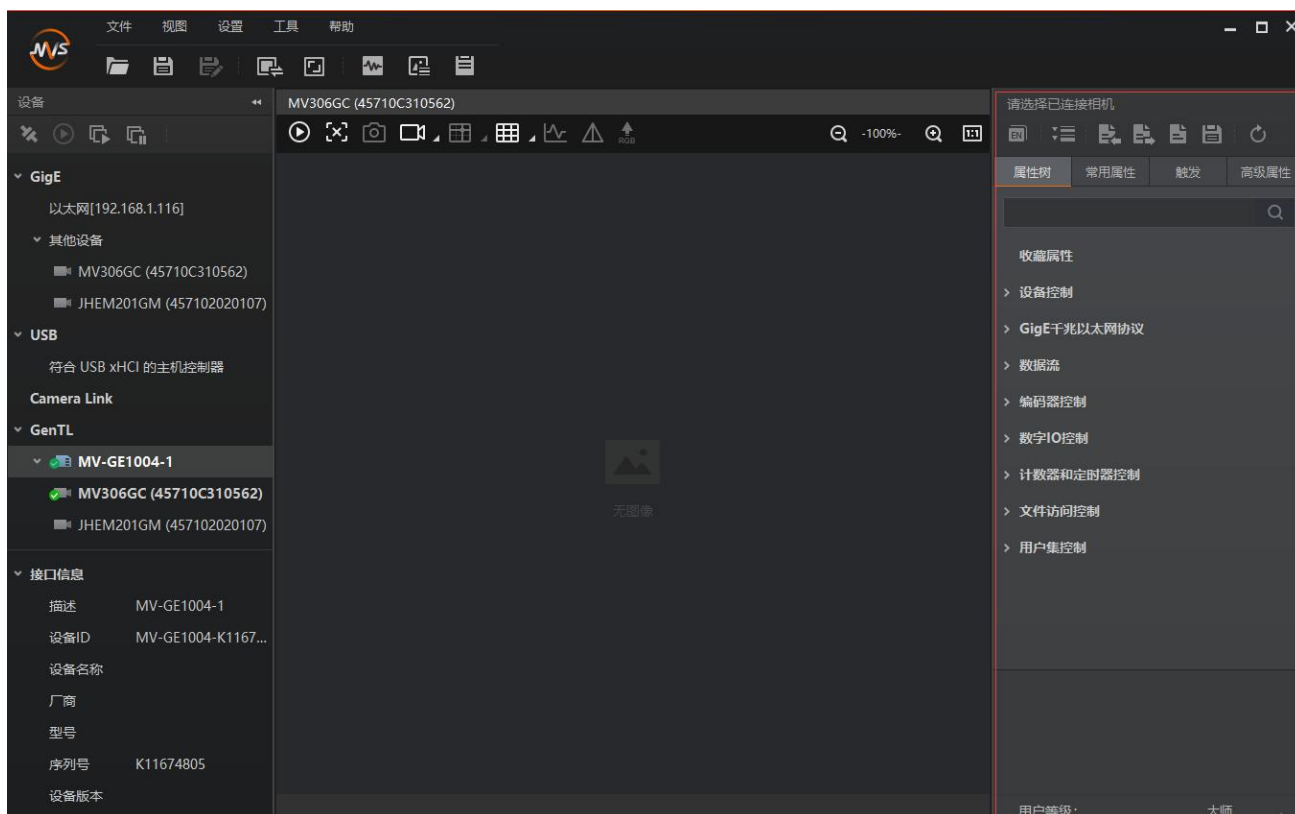


- GenTL 标准库的 SDK 文件路径为 C:\Program Files (x86)\Common Files\MVS\Runtime。

- cti 文件分为 32 位和 64 位，请根据实际需求进行选择。32 位 cti 文件路径为 C:\Program Files (x86)\Common Files\MVS\Runtime\Win32_i86；64 位 cti 文件

路径为 C:\Program Files (x86)\Common Files\MVS\Runtime\Win64_x64

3. 点击 连接采集卡，右侧显示采集卡属性树，MVS 客户端主界面如下图所示，可查看或设置采集卡相关参数。



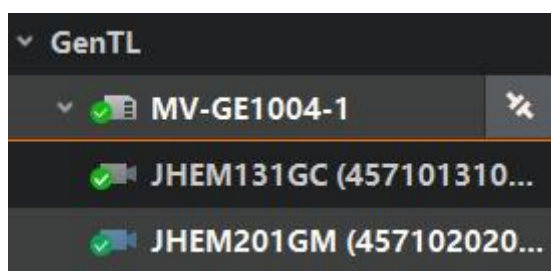
4. 通过设备属性区域可设置相关参数。单击属性名称前的">", 可展开采集卡的具体属性。

各属性分类的介绍请见下表。

属性	名称	功能概述
Device Control	设备控制	该属性可用于查看采集卡基本信息
GigE	GigE 传输层控制	该属性可对采集卡的 PoE 功能和动作控制进行设置
Stream	流控制	该属性可对帧信号处理和图像调试功能进行设置
Encoder Control	轴编码器控制	该属性可设置轴编码器相关信号源及其他相关参数

Digital IO Control	数字 I/O 控制	该属性可对 I/O 信号进行设置和管理
Counter And Timer Control	计数器和定时器控制	该属性可设置定时器相关信号源及其他相关参数
File Access Control	文件存取控制	该属性可查看支持文件存取功能采集卡参数组的信息。
User Set Control	用户参数设置	该属性用于保存、加载采集卡的参数组，也可设置默认启动的参数组

5. 加载 GenTL 库后，客户端 GenTL 下自动枚举采集卡以及连接采集卡的相机，如下图所示，可对相机进行参数设置或采集图像等操作。



6. 基于采集卡 SDK 进行二次开发

以下示例为枚举采集卡:

```

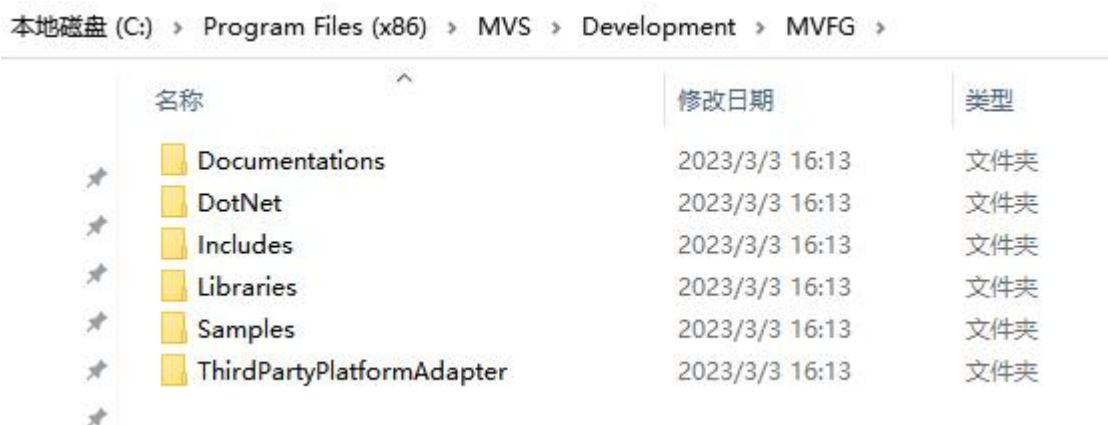
1     MV_GENTL_IF_INFO_LIST m_stIFInfoList;
2     memset(&m_stIFInfoList, 0, sizeof(MV_GENTL_IF_INFO_LIST));
3     MV_GENTL_DEV_INFO_LIST m_stDeviceList;
4     memset(&m_stDeviceList, 0, sizeof(MV_GENTL_DEV_INFO_LIST));
5     // ch:基于GenTL枚举Interface | en:Enum Interface Based On GenTL
6     nRet = MV_CC_EnumInterfacesByGenTL(&m_stIFInfoList, "C:/Program Files/KAYA Instruments/Common/bin/KYFGLibGer
7     if (MV_OK != nRet)
8     {
9         printf("Enum InterfacesByGenTL fail! nRet [0x%x]\n", nRet);
10    }
11    if (m_stIFInfoList.nInterfaceNum == 0)
12    {
13        printf("nInterfaceNum 0\n");
14    }
15    for (unsigned int i = 0; i < m_stIFInfoList.nInterfaceNum; i++)//打印枚举到的采集卡
16    {
17        char strUserName[256];
18        MV_GENTL_IF_INFO* pstIFInfo = m_stIFInfoList.pIFInfo[i];
19        sprintf_s(strUserName, "Interface[%d]:%s %s (%s) (%d)", i, pstIFInfo->chTLType, pstIFInfo->chInterfaceID
20        printf("strUserName:%s\n", strUserName);
21    }
22    // ch:枚举板卡
23    nRet = MV_CC_EnumDevicesByGenTL(m_stIFInfoList.pIFInfo[0], &m_stDeviceList);
24    if (MV_OK != nRet)
25    {
26        printf("Enum DevicesByGenTL fail! nRet [0x%x]\n", nRet);
27    }
28    if (m_stDeviceList.nDeviceNum == 0)
29    {
30        printf("nDeviceNum 0\n");
31    }
32    // ch:选择设备并创建句柄 | en:Select device and create handle
33    int index = 0;
34    nRet = MV_CC_CreateHandleByGenTL(&handle, m_stDeviceList.pDeviceInfo[index]);
35    if (MV_OK != nRet)
36    {
37        printf("Create Handle fail! nRet [0x%x]\n", nRet);
38    }
39    // ch:打开设备 | en:Open device
40    nRet = MV_CC_OpenDevice(handle);
41    if (MV_OK != nRet)
42    {
43        printf("Open Device fail! nRet [0x%x]\n", nRet);
44    }
    
```

基于工业相机 SDK 枚举相机:

```

1 // ch:枚举设备 | Enum device
2 MV_CC_DEVICE_INFO_LIST stDeviceList;
3 memset(&stDeviceList, 0, sizeof(MV_CC_DEVICE_INFO_LIST));
4 nRet = MV_CC_EnumDevices(MV_GIGE_DEVICE | MV_USB_DEVICE, &stDeviceList);
5 if (MV_OK != nRet)
6 {
7     printf("Enum Devices fail! nRet [0x%x]\n", nRet);
8 }
9 if (stDeviceList.nDeviceNum > 0)
10 {
11     for (unsigned int i = 0; i < stDeviceList.nDeviceNum; i++)
12     {
13         printf("[device %d]:\n", i);
14         MV_CC_DEVICE_INFO* pDeviceInfo = stDeviceList.pDeviceInfo[i];
15         if (NULL == pDeviceInfo)
16         {
17             break;
18         }
19         PrintDeviceInfo(pDeviceInfo);
20     }
21 }
22 else
23 {
24     printf("Find No Devices!\n");
25     break;
26 }
27
28 printf("Please Input camera index(0-%d):", stDeviceList.nDeviceNum-1);
29 unsigned int nIndex = 0;
30 scanf_s("%d", &nIndex);
31 if (nIndex >= stDeviceList.nDeviceNum)
32 {
33     printf("Input error!\n");
34 }
35 // ch:选择设备并创建句柄 | Select device and create handle
36 nRet = MV_CC_CreateHandle(&handle, stDeviceList.pDeviceInfo[nIndex]);
37 if (MV_OK != nRet)
38 {
39     printf("Create Handle fail! nRet [0x%x]\n", nRet);
40 }
41 // ch:打开设备 | Open device
42 nRet = MV_CC_OpenDevice(handle);
43 if (MV_OK != nRet)
44 {
45     printf("Open Device fail! nRet [0x%x]\n", nRet);
46 }
    
```

更多示例程序路径: C:\Program Files (x86)\MVS\Development\MVFG, 如下图所示:



4) 采集卡常见问题

问题描述	可能的原因	解决方法
无法搜索到采集卡	<ul style="list-style-type: none"> ● 驱动安装异常 ● MVS 客户端或其他 软件正在使用采集卡 ● 主板供电不足 ● MVS 软件未完整的安装 	<ul style="list-style-type: none"> ● 重新安装驱动, 安装成功后, 设备管理器中能够正常显示采集卡 ● 关闭 MVS 客户端或正在使用采集卡的软件 ● 使用机箱电源, 对采集卡单独供电 ● 卸载原有的 MVS, 重新安装
采集卡可搜索, 相机无法搜索	<ul style="list-style-type: none"> ● 相机未正常工作 ● 数据线异常 	<ul style="list-style-type: none"> ● 确认相机是否正常工作 ● 数据线是否正常连接
相机帧率不稳定	<ul style="list-style-type: none"> ● PCIE 带宽不足 ● 系统 CPU 性能低 	<ul style="list-style-type: none"> ● 更换 PCIE 卡槽 ● 使用高性能 CPU 再进行测试