

工业相机参数设置获取

*文档适用范围：工业相机 SDK

关键字：SDK、参数设置

目录

1.1 SDK 通用接口说明.....	1
1.2 相机参数类型查询.....	2
1.3 常用参数设置获取示例.....	2

1.1 SDK 通用接口说明

相机参数类型可分为六类,除 command 参数外,每一类都有其对应的设置与获取函数接口,

表 1 参数类型及对应函数接口介绍

类型	描述	对应函数接口
Integer	整数型参数	MV_CC_GetIntValueEx()
		MV_CC_SetIntValueEx()
Enumeration	枚举型参数	MV_CC_GetEnumValue()
		MV_CC_SetEnumValue()
Boolean	布尔型参数	MV_CC_GetBoolValue()
		MV_CC_SetBoolValue()
Float	浮点型参数	MV_CC_GetFloatValue()
		MV_CC_SetFloatValue()
String	字符串型参数	MV_CC_GetStringValue()
		MV_CC_SetStringValue()
Command	命令型参数	MV_CC_SetCommandValue()

*详细函数接口可参考 SDK 手册：C:\Program Files (x86)\MVS\Development\Documentations

1.2 相机参数类型查询

对于相机的每一个参数，在 MVS 客户端中都可以找到相应的类型、节点名称、取值范围、步进等信息，从图 1 可以看出，相机高度节点名称为”Height”，节点类型”Integer”，取值范围”32-1080”，步进值为 4。通过节点类型我们可以看出，应该使用 MV_CC_GetIntValueEx()获取参数值，使用 MV_CC_SetIntValueEx()设置参数值。其他参数用法类似。

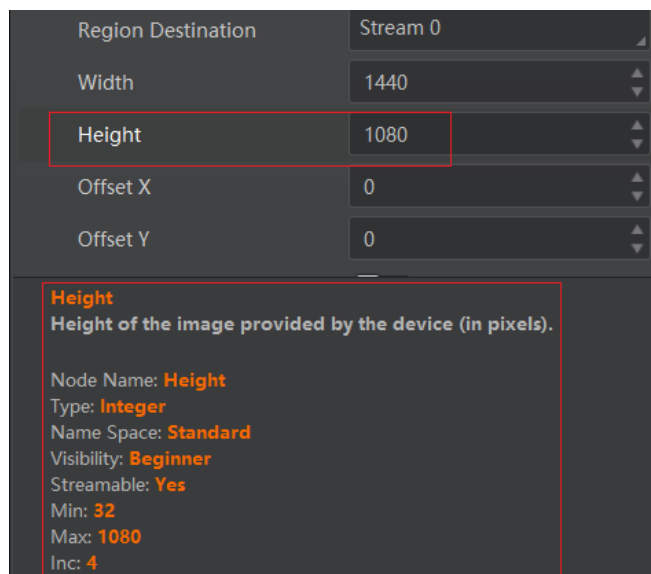
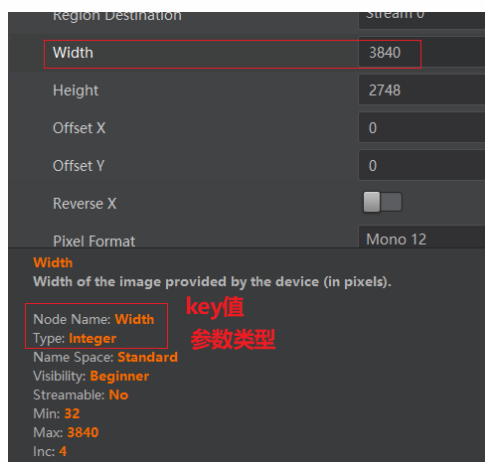


图 1 MVS 查看相机参数信息

1.3 常用参数设置获取示例

1) Integer 型参数设置与获取—以图像宽度获取设置为例

- MVS 查看参数类型及节点名称



- 调用对应函数接口

```
//获取int型参数
MVCC_INTVALUE_EX struIntValue = { 0 };
nRet = MV_CC_GetIntValueEx(handle, "Width", &struIntValue);
if (MV_OK != nRet)
{
    printf("Error: GetIntValue fail [%x]\n", nRet);
}
//打印当前宽度的最大值、最小值、步进
printf("Width:%I64d,Width_Max:%I64d,Width_min:%I64d,Width_Inc:%I64d\n", struIntValue.nCurValue,
struIntValue.nMax, struIntValue.nMin, struIntValue.nInc);

//设置int型参数
/*


- value值必须是步进值(Inc)的整数倍，否则会失败
- 宽度、高度等参数设置时，只有在MV_CC_OpenDevice之后，MV_CC_Startgrab接口调用前才能设置，取流过程中，不能修改宽高
- 宽度、高度等参数设置时，若有Offset X、Y偏移，应当先调用相关接口，将偏移量置0


int64_t nValue = 1000;
nRet = MV_CC_SetIntValueEx(handle, "Width", nValue);
if (MV_OK != nRet)
{
    printf("Error: SetIntValue fail [%x]\n", nRet);
}
*/
```

2) Command 型参数——以软触发设置为例

```
//设置Command型节点-发送软触发命令
//需要先打开【触发模式】，触发源选择【软触发】后才可以设置软触发命令
nRet = MV_CC_SetCommandValue(handle, "TriggerSoftware");
if (MV_OK != nRet)
{
    printf("Error: SetCommandValue fail [%x]\n", nRet);
}
}
```

3) Float 型参数设置与获取——以曝光获取、设置为例

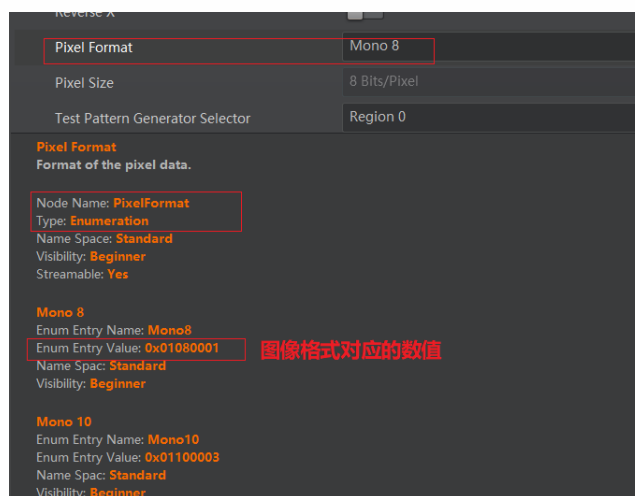
```
//曝光参数获取
MVCC_FLOATVALUE struFloatValue = { 0 };
nRet = MV_CC_GetFloatValue(handle, "ExposureTime", &struFloatValue);
if (MV_OK != nRet)
{
    printf("Error: GetFloatValue fail [%x]\n", nRet);
}
}
```

```

//设置float型参数-曝光值
//设置曝光时，需要注意是否已经开启自动曝光、曝光模式，它们都会影响曝光参数的设置范围及是否可以设置，可在MVS中进行观察
float fValue = 1000;
nRet = MV_CC_SetFloatValue(handle, "ExposureTime", fValue);
if (MV_OK != nRet)
{
    printf("Error: SetFloatValue fail [%x]\n", nRet);
}
    
```

4) Enumeration 型参数设置与获取—以相机图像格式获取、设置为例

- MVS 查看参数类型及节点名称



- 调用对应函数接口

```

//获取Enum型参数—相机图像格式
MVCC_ENUMVALUE struEnumValue = { 0 };
nRet = MV_CC_GetEnumValue(handle, "PixelFormat", &struEnumValue);
if (MV_OK != nRet)
{
    printf("Error: GetEnumValue fail [%x]\n", nRet);
}
printf("PixelFormat:[0x%x]\n\n", struEnumValue.nCurValue); //eg: PixelFormat:[0x01080001]
    
```

```

//设置Enum型参数-相机图像格式
//只有在MV_CC_OpenDevice之后， MV_CC_Startgrab接口调用前才能设置图像格式
nRet = MV_CC_SetEnumValue(handle, "PixelFormat", PixelType_Gvsp_Mono8);
//可以直接写PixelFormat_Gvsp_Mono8或对应数值0x01080001
if (MV_OK != nRet)
{
    printf("Error: SetEnumValue fail [%x]\n", nRet);
}
    
```

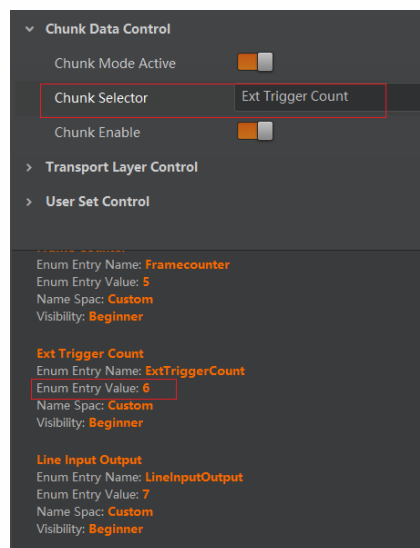
5) String 参数获取与设置—用户名称

```
//获取string型参数
MVCC_STRINGVALUE struStringValue = { 0 };
nRet = MV_CC_GetStringValue(handle, "DeviceUserID", &struStringValue);
if (MV_OK != nRet)
{
    printf("Error: GetStringValue fail [%x]\n", nRet);
}
printf("DeviceUserID:[%s]\n", struStringValue.chCurValue);
```

```
//设置string型参数-自定义用户名称
//只有在MV_CC_OpenDevice之后，MV_CC_Startgrab接口调用前才能设置用户ID
nRet = MV_CC_SetStringValue(handle, "DeviceUserID", "HikrobotCamera");
if (MV_OK != nRet)
{
    printf("Error: SetStringValue fail [%x]\n", nRet);
}
```

6) Boolean 参数设置—触发计数水印信息获取

- MVS 查看参数类型及节点名称



- 调用对应函数接口

```
//开启水印信息需要按照步骤，才能一步步进行操作
//只有在MV_CC_OpenDevice之后，MV_CC_Startgrab接口调用前才能设置水印信息
nRet = MV_CC_SetBoolValue(handle, "ChunkModeActive", 1); //开启水印模式
if (MV_OK != nRet)
{
    printf("Error: ChunkModeActive fail! nRet [0x%x]\n", nRet);
}
nRet = MV_CC_SetEnumValue(handle, "ChunkSelector", 6); //选择水印信息：触发计数
if (MV_OK != nRet)
{
    printf("Error: ChunkSelector fail! nRet [0x%x]\n", nRet);
}
nRet = MV_CC_SetBoolValue(handle, "ChunkEnable", 1); //使能水印
if (MV_OK != nRet)
{
    printf("Error: ChunkEnable fail! nRet [0x%x]\n", nRet);
}
```