

Windows 开发指南

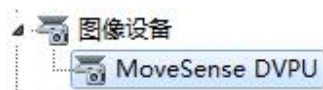
1、简介

MoveSense DVPU 双目深度相机支持 Windows 环境开发，在 Windows 下 MoveSense DVPU 作为 UVC 标准设备免去驱动安装的步骤，自动识别为视频设备。人加目前为 MoveSense DVPU 提供了 C++版本的 SDK，使得用户能在 Windows 平台下开发基于双目深度相机的应用。

2、快速使用说明

在 Windows 平台下提供了快速测试的软件程序，将设备插入 Windows 系统的主机，打开用于测试的 Windows 可执行程序即可对设备进行快速测试，具体步骤如下：

Step 1.连接 USB2.0 线缆或 USB3.0 线缆至测试主机，电脑端会自动加载驱动程序，设备管理器中会多出一个视频设备，如下图所示。



Step 2.运行 Windows 版的 SDK 测试程序便可查看图像。

3、工作模式说明

MoveSense DVPU 支持三种工作模式，三种工作模式参数设置如下，可根据需要自行选择：

D_VPU_320X240_LR_30FPS	显示左右图
D_VPU_320X240_LD_30FPS	显示左图和视差图
D_VPU_320X240_LRD_30FPS	显示左右图和视差图

```
int sel = D_VPU_320X240_LD_30FPS;  
movesense::MoveSenseCamera c(sel);
```

4、可设置接口说明

- (1) void OpenCamera():打开相机。
 - (2) void CloseCamera():关闭相机。
 - (3) void EnableCam(bool val):可在相机打开的过程中使能和停止相机流，
一般使用不到（true-使能，false-禁止）。
 - (4) void SetUndistort(bool val): 打开/关闭矫正开关（true-打开；false--关闭）。
 - (5) int GetCameraID():获取相机的 ID 号，默认会打印 ID 号（4 个字节），
并有一个整形的返回值。
 - (6) void GetCamParas(MsgPkg& p):获取相机参数 t_P2(在示例程序中有示例
代码),关于相机参数 t_P2 的使用请看“相机参数/相机参数说明.pdf”
 - (7) int GetImageData(unsigned char * &data,int &len):获取相机的图像数据。
- (注：函数的声明在 MoveSenseCamera.h 中)

*****如需自己开发 SDK 要看以下部分*****

5、Set/Get 接口的具体实现

具体命令的实现过程是：采用 VideoProcAmp_Saturation 通道去实现 I2C 寄存器地址的传输；

```
void SendI2CAddr(int addr){
    m_pVideoProcAmp->Set (VideoProcAmp_Saturation, addr, 2);
    //printf("SendI2CAddr--is--%d--\n", addr);
}
```

此外用 VideoProcAmp_WhiteBalance 通道去更新该寄存器地址对应的数值（其中 VIDIOC_S_CTRL 用来实现寄存器的写，VIDIOC_G_CTRL 用来实现寄存器的读）。

```
void SetI2CValue(int val){
    m_pVideoProcAmp->Set (VideoProcAmp_WhiteBalance, val, 2);
    //printf("SetI2CValue--is--%d--\n", val);
}

unsigned char GetI2CValue() {
    long val[1];
    long cnt[1];
    m_pVideoProcAmp->Get (VideoProcAmp_WhiteBalance, val, cnt);
    //printf("GetI2CValue--is--%d--%d--\n", (unsigned char)val[0], cnt[0]);
    return (unsigned char)val[0];
}
```

Example:

以 void SetUndistort(bool val)为例，这个命令是校正开关的设置，其对应的 I2C 寄存器是 0xB4。

①假如我们想关闭校正开关，即：SetUndistort(false)，该函数的具体实现过程是：

```
int addr = 0xB4;
Bool val = false;
Set(VideoProcAmp_Saturation, addr, 2);
Sleep(1); //等待 1ms
Set(VideoProcAmp_WhiteBalance, val, 2);
```

②假如我们想获取校正开关对应寄存器的数值，具体做法是

```
int addr = 0xB4;
Long val[1],cnt[1];
Set(VideoProcAmp_Saturation, addr, 2);
```

Sleep(1); //等待 1ms

Get(VideoProcAmp_WhiteBalance, val, cnt);//返回值在 val[0]中

printf("the register value is %d \n",val[0]);