

## Orbbec-OpenNI2扩展API使用说明

ORBEC Confidential

更改履历					
日期	修改状态	页码	更改说明	修改人	批准人
2017-6-30	V0.1	7	首次发行		

## 目录

1. 目的 .....	4
2. 适用范围 .....	4
3. 简介 .....	4
4. 获取设备序列号 .....	4
5. 获取设备类型 .....	4
6. 获取相机保存标定参数 .....	4
7. 保存相机标定参数到设备 .....	5
8. 获取 IR 增益 .....	6
9. 设置 IR 增益 .....	6
10. 获取 IR 曝光 .....	7
11. 设置 IR 曝光 .....	7
12. 设置 LDP 开关，需要重新插拔 USB 设备生效 .....	7
13. 设置 LDM 激光开关 .....	7

# 1. 目的

Orbbec 在 Release Orbbec 设备相关 SDK 时，对 OpenNI2 的功能进行了扩展，使 OpenNI2 接口可以实现读取序列号，获取设备类型，使用 Flash 保存，读取标定参数，IR 参数调节，LDP 控制，LDM 模块控制等功能。本文针对这些新增功能进行描述，并示范基本操作流程。

# 2. 适用范围

本文档适用于 Astra 系列摄像头，包括但不限于 Astra，Astra Pro，Astra Mini 全系列。适用平台包括 windows 7 及以上，Ubuntu 14.04 及以上。

# 3. 简介

OpenNI2 扩展 API 依赖最新发布的 Orbbec-OpenNI2-2.3.1 版 SDK，包括 Windows X86/X64, Linux X86/X64, Arm32, Arm64 和 Android 版本，用户如果需要使用这些扩展功能，在调用相关扩展 API 之前，需要先确保 OpenNI2 Initialize 成功，设备 open 成功，下面详细介绍所有扩展 API 的使用方法。

# 4. 获取设备序列号

```
char serNumber[12];
int dataSize = sizeof(serNumber);
memset(serNumber, 0, dataSize);
g_Device.getProperty(openni::OBEXTENSION_ID_SERIALNUMBER, (uint8_t *)&serNumber, &dataSize);
```

# 5. 获取设备类型

```
char devType[32];
int dataSize = sizeof(devType);
memset(devType, 0, dataSize);
g_Device.getProperty(openni::OBEXTENSION_ID_DEVICETYPE, (uint8_t *)&devType, &dataSize);
```

# 6. 获取相机保存标定参数

目前 Astra 系列只有 Astra S 在出厂时有保存标定参数。其他型号可以手动标定并按第 7 节做法保存到设备。

```
typedef struct OBCameraParams
{
    float l_intr_p[4];    //[fx,fy,cx,cy]
    float r_intr_p[4];    //[fx,fy,cx,cy]
    float r2l_r[9];      //[r00,r01,r02;r10,r11,r12;r20,r21,r22]
    float r2l_t[3];      //[t1,t2,t3]
    float k[5];          //[k1,k2,p1,p2,k3]
    int is_mirror;
}OBCameraParams;

OBCameraParams m_CamParams;
int dataSize = sizeof(OBCameraParams);
g_Device.getProperty(openni::OBEXTENSION_ID_CAM_PARAMS, (uint8_t *)&m_CamParams, &dataSize);
```

## 7. 保存相机标定参数到设备

```
typedef struct OBCameraParams
{
    float l_intr_p[4];    //[fx,fy,cx,cy]
    float r_intr_p[4];    //[fx,fy,cx,cy]
    float r2l_r[9];      //[r00,r01,r02;r10,r11,r12;r20,r21,r22]
    float r2l_t[3];      //[t1,t2,t3]
    float k[5];          //[k1,k2,p1,p2,k3]
    int is_mirror;
} OBCameraParams;

OBCameraParams m_CamParams = { 0 };
int dataSize = sizeof(OBCameraParams);

m_CamParams.l_intr_p[0] = 577.318970;
m_CamParams.l_intr_p[1] = 577.318970;
m_CamParams.l_intr_p[2] = 308.729004;
m_CamParams.l_intr_p[3] = 269.143005;

m_CamParams.r_intr_p[0] = 517.447998;
m_CamParams.r_intr_p[1] = 517.447998;
m_CamParams.r_intr_p[2] = 305.432007;
m_CamParams.r_intr_p[3] = 250.410995;

m_CamParams.r2l_r[0] = 0.999972;
m_CamParams.r2l_r[1] = -0.005735;
```

```
m_CamParams.r2l_r[2] = 0.004735;
m_CamParams.r2l_r[3] = 0.005736;
m_CamParams.r2l_r[4] = 0.999983;
m_CamParams.r2l_r[5] = -0.000298;
m_CamParams.r2l_r[6] = -0.004733;
m_CamParams.r2l_r[7] = 0.000325;
m_CamParams.r2l_r[8] = 0.999989;

m_CamParams.r2l_t[0] = -25.147900;
m_CamParams.r2l_t[1] = 0.015202;
m_CamParams.r2l_t[2] = -0.648167;

m_CamParams.k[0] = -0.077348;
m_CamParams.k[1] = 0.208761;
m_CamParams.k[2] = -0.196780;
m_CamParams.k[3] = 0.000617;
m_CamParams.k[4] = 0.001059;

m_CamParams.is_mirror = 0;
g_Device.setProperty(openni::OBEXTENSION_ID_CAM_PARAMS, (uint8_t *)&m_CamParams, dataSize);
```

## 8. 获取 IR 增益

```
int gain = 0;
int dataSize = 4;
g_Device.getProperty(openni::OBEXTENSION_ID_IR_GAIN, (uint8_t *)&gain, &dataSize);
printf("ir gain value : 0x%x\n", gain);
```

## 9. 设置 IR 增益

```
int gain = 0;
int dataSize = 4;
g_device.getProperty(openni::OBEXTENSION_ID_IR_GAIN, (uint8_t *)&gain, &dataSize);
printf("ir gain value : 0x%x\n", gain);
gain++;
g_device.setProperty(openni::OBEXTENSION_ID_IR_GAIN, (uint8_t *)&gain, dataSize);
```

## 10. 获取 IR 曝光

```
int exposure = 0;
int dataSize = 4;
g_device.getProperty(openni::OBEXTENSION_ID_IR_EXP, (uint8_t*)&exposure, &dataSize);
printf("ir exposure value : 0x%x\n", exposure);
```

## 11. 设置 IR 曝光

```
int exposure = 0;
int dataSize = 4;
g_device.getProperty(openni::OBEXTENSION_ID_IR_EXP, (uint8_t*)&exposure, &dataSize);
printf("ir exposure value : 0x%x\n", exposure);
exposure += 256;
g_device.setProperty(openni::OBEXTENSION_ID_IR_EXP, (uint8_t*)&exposure, dataSize);
```

## 12. 设置 LDP 开关，需要重新插拔 USB 设备生效

```
int dataSize = 4;
int ldp_en = enable;
g_Device.setProperty(openni::OBEXTENSION_ID_LDP_EN, (uint8_t*)&ldp_en, dataSize);
```

## 13. 设置 LDM 激光开关

```
int dataSize = 4;
int laser_en = enable;
g_Device.setProperty(openni::OBEXTENSION_ID_LASER_EN, (uint8_t*)&laser_en, dataSize);
```