女娲星-XB40/XB40L/X100 (Nuwa-XB40/XB40L/X100) Windows SDK 开发规范

修订历史

版本号	时间	修定记录	修订部门
V1.0.0	2022/04/22		软件部

目录

图表

1.简介

1.1 产品概述

安思疆 Linux-SDK API 为第三方应用集成摄像头模组功能提供了一站式服务。基于 SDK 可快速为客户提供二次开发服务,防止在使用 Angstrong-Linux SDK API 的过程中由于不规范的调用而引起其他问题。

SDK 用于驱动安思疆科技女娲星系列模组，提供实时获取深度图像、点云图等功能

图 1 女娲星产品实物图

1.2 环境要求

1.2.1 运行环境

Windows 平台

1.2.2 系统要求

Windows 平台

1.3 SDK 功能介绍

1.3.1 获取深度

支持驱动安思疆女娲星系列模组，实时获取深度图；
深度数据的空间格式如下

表 1 深度图格式

位宽度	16
单位	mm
类型	Short
深度范围	27cm~~4m

1.3.2 获取点云

提供接口对深度图进行转化点云图；也可注册回调，直接得到点云图数据

2.SDK 集成指南

2.1 包结构说明

SDK

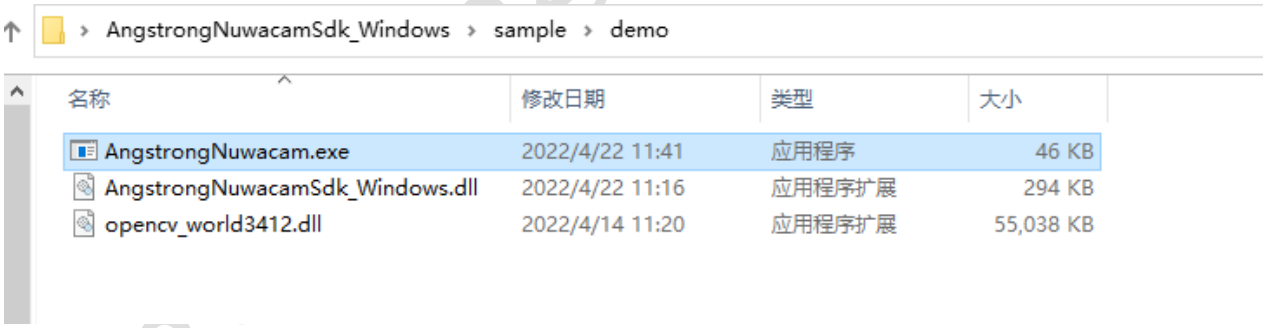
```
├── libs
|   ├── include                    SDK 头文件
|   └── lib                        SDK 库文件

├── sample
|   ├── demo                       windows 系统下的运行 demo 及其运行依赖库
|   ├── src                        windows 系统下的参考例子
```

2.2 示例程序运行说明

2.2.1 运行 window 系统示例程序

进入 SDK 包的 sample 文件夹，然后再进入 demo 文件夹，可以看到示例的 exe，和其运行需要的的 dll 动态库。如下图所示



2.3 调用流程图

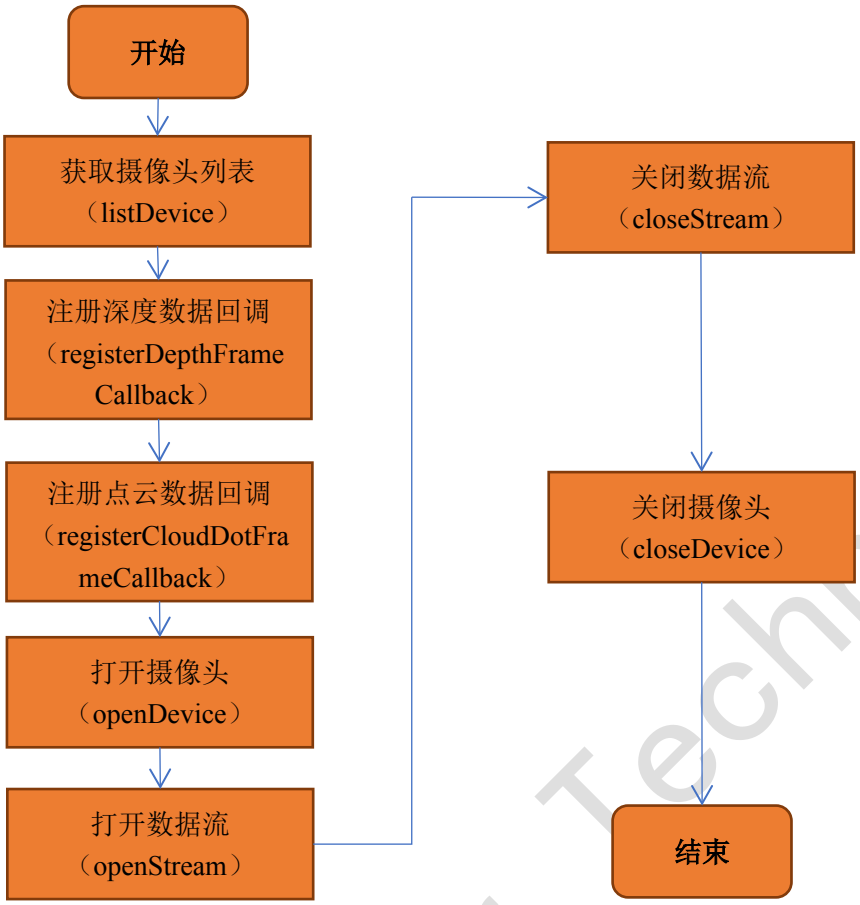


图 2 调用流程图

3 ANG SDK 核心 API 说明

3.1 获取深度图像接口

3.1.1 获取摄像头列表

```
1.   /**
2.     * 获取摄像头列表
3.     * 参数: [out] deviceIDs 设备号的集合，里面设备号唯一，但不一定连续
4.     * 返回值 ： 连接的设备个数
5.     */
6.    int listDevice(std::vector<int>& deviceIDs);
```


3.1.2 释放资源

```
/**
    * 释放资源
    * 参数：无
    * 返回值 ： 0 success ； != 0 fail
    */
  int deinitCapture();
```

3.1.3 打开摄像头

```
/**
    * 打开摄像头
    * 参数:[in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
    * 返回值 ： 0 success ； != 0 fail
    */
  int openDevice(int deviceID);
```

3.1.4 关闭摄像头

```
/**
    * 关闭摄像头
    * 参数:[in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
    * 返回值 ： 0 success ； != 0 fail
    */
  int closeDevice(int deviceID);
```

3.1.5 打开深度数据流

```
 /**
    * 打开深度数据流
    * 参数:[in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
    * 返回值 ： 0 success ； != 0 fail
    */
  int openStream(int deviceID);
```

3.1.6 关闭深度数据流

```
/**
   * 关闭深度数据流
   * 参数: [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   * 返回值 : 0 success ; != 0 fail
   */
int closeStream(int deviceID);
```

3.1.7 读取深度数据

```
/**
   * 读取深度数据（用 3.1.8 注册深度数据回调函数代替）
   * 参数: [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *       [out] angFrame 数据帧
   * 返回值 : 0 success ; != 0 fail
   */
int readFrame(int deviceID,AngFrame *angFrame);
```

3.1.8 注册深度数据回调函数

```
/**
   * 注册深度数据回调函数
   * 参数 [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   * 参数 : [in] pstCallback 深度数据回调函数
   * 返回值 : 0 success ; != 0 fail
   */
int registerDepthImageCallback(int deviceID, const
AS_DEPTHIMAGE_Callback *pstCallback);
```

3.1.9 注册点云数据回调函数

```
/**
   * 注册点云数据回调函数
   * 参数 [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   * 参数 : [in] pstCallback 点云数据回调函数
   * 返回值 : 0 success ; != 0 fail
   */
int registerPointCloudCallback(int deviceID, const
AS_POINTCLOUD_Callback *pstCallback);
```

3.1.10 设置分辨率

```
/**
   * 设置分辨率
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *       [in] zoom 缩小 DEFAULT(640*400) X2(320*200)
   * 返回值 : 0 success ; != 0 fail
   */
int setDownSample(int deviceID, Zoom zoom);
```

3.1.11 深度转点云

```
/**
   * 深度转点云
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *       [in] angFrame 深度数据帧
   *       [out] cloudDot 点云数据
   * 返回值 : cloudDot 点云数据大小
   */
int convertDepthToCloudDot(int deviceID, AngFrame angFrame, float*
cloudDot);
```

3.1.12 旋转数据流

```
/**
   * 旋转数据流
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *       [in] rotate 旋转角度 ANGLE_0(0) CLOCKWISE_90(-90)
   *CLOCKWISE_180(-180) CLOCKWISE_270(-270) ANTICLOCKWISE_90(90)
   *ANTICLOCKWISE_180(180) ANTICLOCKWISE_270(270)
   * 返回值 : 0 success ; != 0 fail
   */
int setRotateAngle(int deviceID,eAngle rotate);
```

3.1.13 配置相机参数

```
/**
   * 配置相机参数
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *        [in] pstCamCfg 参数配置,
   *        isAtcOn: 是否开启温度补偿，温漂功能需固件支持，如固件不支持，则配置 false
   *        isDenoisesOn: 是否开启降噪功能;
   *        isFillHolesOn: 是否开启深度图补洞功能;
   *        isAntiDistortion: 是否开启深度图抗畸变功能;
   * 返回值 : 0 success ; != 0 fail
   */
int setAsCamCfg(int deviceID, const AS_CAM_CFG_S *pstCamCfg);
```

3.1.14 设置帧率

```
/**
   * 设置深度图帧率，此功能需固件支持切换帧率
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *        [in] enFps 帧率大小,
   * 返回值 : 0 success ; != 0 fail
   */
int setFrameRate(int deviceID, AS_FPS_E enFps);
```

3.1.15 设置投射器

```
 /**
   * 设置摄像头图像投射器开关
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *        [in] enable true 打开  false 关闭
   * 返回值 : 0 success ; != 0 fail
   */
int setVCSEL(int deviceID, bool enable);;
```

3.2 扩展 API 接口

3.2.1 获取 SDK 版本号

```
/**
   * 获取 SDK 版本号
   * 参数: [out] version 版本号
   * 返回值 : 0 success ;  != 0 fail
   */
   int getSDKSoftwareVersion(std::string &version);
```

3.2.4 获取摄像头 SN 号

```
/**
   * 获取 SN 号
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *       [out] SN  sn 号
   * 返回值 :  0 success ;  != 0 fail
   */
int getSN(int deviceID, std::string &SN);
```

3.2.4 获取内参

```
/**
   * 获取内参
   * 参数 : [in] deviceID 设备号 取值范围:(listDevice 中容器 deviceIDs 的设备号)
   *       [out] irRgbparameter 内参结构体
   * 返回值 :  0 success ;  != 0 fail
   */
int getIrRgbParameter(int deviceID, IrRgbParameter *irRgbParameter);
```

4.获取深度视频流和点云代码实现

```
/*深度数据回调函数*/
static void AS_SDK_DepthImageCallback(int deviceId, const
       AS_DEPTHIMAGE_Data     *pstFrame, void *privateData)
{

}

```

```
/*点云数据回调函数*/
static void AS_SDK_PointCloudCallback(int deviceId, const
        AS_POINTCLOUD_Data    *pstPointCloud, void *privateData)
{

}
std::vector<int> deviceIDs;
/* Main program for angstrong nuwa camera sdk demo */
int main(int argc, char *argv[])
{
    int ret = 0;
    Capture capture;
     std::string sdkVersion;

     LOG(INFO) << "Hello, angstrong NUWA camera" << std::endl;
     capture.getSDKSoftwareVersion(sdkVersion);      /*获取摄像头 sdk*/
     int deviceNum = capture.listDevice(deviceIDs);   /*获取连接的摄像头列表
*/
     if (deviceNum <= 0) {
         LOG(INFO) << "no device!\n";
         return 0;
     }
     LOG(INFO) << "Found " << deviceNum << " nuwa camera" << std::endl;

     for (int idx = 0; idx < deviceNum; idx++) {
         AS_CAM_CFG_S stCamCfg;
         /*register callback function*/
         AS_DEPTHIMAGE_Callback stDepthImageCallback = {
             .callback = AS_SDK_DepthImageCallback,
             .privateData = (void *)0x1234,
         };
         AS_POINTCLOUD_Callback stPointCloudCallback = {
             .callback = AS_SDK_PointCloudCallback,
             .privateData = (void *)0x1234,
         };

         /* 注册深度数据回调函数 */
         capture.registerDepthImageCallback(deviceIDs.at(idx),
                                    &stDepthImageCallback);
         /* 注册点云数据回调函数 */
         capture.registerPointCloudCallback(deviceIDs.at(idx),
                                    &stPointCloudCallback);
         /* 打开相机 */
         if (capture.openDevice(deviceIDs.at(idx))!= 0) {
             LOG(ERROR) << "device id " << deviceIDs.at(idx) << " open
device failed!\n";
```



```
            continue;
        }
        LOG(INFO) << "Open camera " << deviceIDs.at(idx) << " success" << std::endl;

        /* 获取默认参数 */
        capture.getAsCamCfg(deviceIDs.at(idx), &stCamCfg);
        LOG(INFO) << " default camera cfg: Atc[" << stCamCfg.isAtcOn << "]"\
                  << " Denoises[" << stCamCfg.isDenoisesOn << "]" \
                  << " FillHoles[" << stCamCfg.isFillHolesOn << "]" \
                  << " AntiDistortion[" << stCamCfg.isAntiDistortion << "]"\
                  << std::endl;
        // stCamCfg.isAtcOn = false;/*温飘功能需模组固件支持*/
        // stCamCfg.isDenoisesOn = true;/*降噪功能*/
        // stCamCfg.isFillHolesOn = false;/*补洞功能*/
        // stCamCfg.isAntiDistortion = false;/*抗畸变功能*/
        /* 配置相机参数 */
        // capture.setAsCamCfg(deviceIDs.at(idx), &stCamCfg);

        /* 设置帧率，此功能需模组固件支持 */
        // capture.setFrameRate(deviceIDs.at(idx), AS_FPS_15);
       /* 打开深度数据流 */
        if (capture.openStream(deviceIDs.at(idx)) != 0) {
            capture.closeDevice(deviceIDs.at(idx));
            LOG(ERROR) << "device id " << deviceIDs.at(idx) << " open stream failed!\n";
            continue;
        }

        /* 获取相机序列号 */
        std::string SN;
        capture.getSN(deviceIDs.at(idx), SN);
        LOG(INFO) << "SN: " << SN << std::endl;

        /* 获取相机内参 */
        IrRgbParameter irRgbParameter;
        capture.getIrRgbParameter(deviceIDs.at(idx), &irRgbParameter);
        LOG(INFO) << "fx: " << irRgbParameter.irFx << std::endl;
        LOG(INFO) << "fy: " << irRgbParameter.irFy << std::endl;
        LOG(INFO) << "cx: " << irRgbParameter.irCx << std::endl;
        LOG(INFO) << "cy: " << irRgbParameter.irCy << std::endl;
    }

    system("pause");
```



```
93.
94.     return 0;
95. }
```