FaceRecognition
Android SDK 开发者指南

型号： DCAM305

版本： V1.0.8

关于本指南

本指南文档主要介绍如何使用 Vzense DCAM305 及 Vzense Android SDK 进行开发。

文档结构

章节	标题	内容
1	概述	介绍 Vzense 产品及 Android SDK 的概况
2	安装	介绍 Vzense 产品及 Android SDK 的安装
3	SDK 使用说明	介绍如何使用 Vzense Android SDK
4	SDK 接口介绍	介绍 Vzense Android SDK 的接口

版本发布记录

日期	版本	发布说明
2019/07/24	V1.0.3	Release 正式版本
2019/10/11	V1.0.4	新增升级以及重启功能
2019/10/12	V1.0.5	变更升级接口，更改 SDK 导入方式
2019/11/05	V1.0.6	添加 3.3.5；添加新 API
2019/11/15	V1.0.7	更新升级相关内容
2020/03/20	V1.0.8	添加 SDK Sample 升级使用说明

目录

1. 概述

Vzense TOF RGBD Camera（型号：DCAM305）是青岛维感科技公司采用飞行时间（Time of Flight，TOF）技术研发的一款 3D 成像模组，具有精度高、环境适应性强、尺寸小等优点。其输出的深度信息可适用于下一代基于手势识别的人机交互、TV 游戏体感交互、人脸识别与活体检测、机器人避障、先进汽车视觉系统、工业控制等前沿创新技术领域。

图 1 Vzense TOF RGBD Camera: DCAM305

Vzense Android SDK 是基于 Vzense DCAM305 TOF RGBD Camera 开发的软件开发工具包，该开发包目前适用于 Android 系统的平板或者智能手机，为应用开发者提供了一系列友好的 API 和简单的应用示例程序。

用户基于该开发包，可获取高精度的深度数据信息、灰度图像信息和彩色图像信息，方便用户开发人脸识别与活体检测、手势识别、投影触控、疲劳检测、三维重建、导航避障等应用。

2. 安装

2.1. 推荐系统配置

配置项	推荐配置
开发环境	Android API 21 及以上 JDK1.7.0_01 及以上
运行环境	Android 5.0 及以上 ARMv7a/ARMv8a @ 1.4GHz+ 512M RAM USB 2.0(支持 Host 功能)

2.2. 安装说明

USB 连接线一端连接模组，另一端连接平板或者智能手机的 USB 接口，如图 2.1 所示。

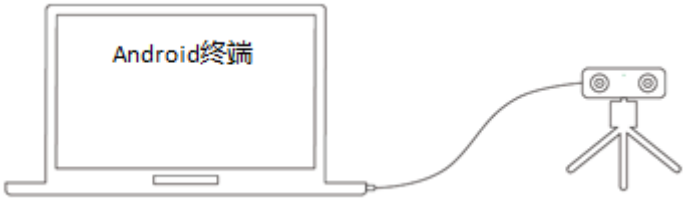

图 2 硬件模组安装示意图

3. SDK 使用说明

3.1. SDK 目录结构

Vzense Android SDK 包含 SDK，Sample，安装包，说明文档等。目录结构如下图所示：

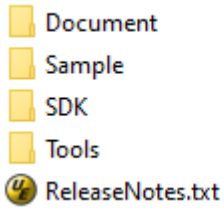

图 3 Android SDK 目录结构

- Document：文档详细描述了 SDK 的开发使用说明。
- Sample：包含使用 Vzense Android SDK 开发的例程工程。
- SDK：包含 Vzense Android SDK 的 jar 文件以及 so 库。
- Tools：包含 Sample APK，压力测试工具，图像质量检测工具。
- ReleaseNotes.txt：介绍历次更新的主要内容。

3.2. 应用程序安装以及运行效果

将 Vzense 深度摄像头连接到 Android 设备的 USB 接口，把 apk 文件拷贝到设备上，双击安装，安装完成打开程序，会启动一个包括图像预览以及菜单按钮的界面，如下图所示。Sample 及 APK 默认进入扫脸模式并显示 RGB 图像，可以点击界面上的 设置菜单 来 设置各种模式。

图 4 VzenseSdkSample.apk 运行效果

3.3. 开发流程

3.3.1. 导入 jar 文件

新建一个 AndroidStudio 工程，拷贝 VzenseCamera.jar 文件到 app/libs 目录，点击 File，在下拉里选择 Project Structure，弹出工程组件界面，如下图所示，选择 Modules 下面的工程，点击右边的 Dependencies 菜单，然后点击右上角的“+”,在弹出的菜单中选择 Jar Dependency，选择上面导入的 VzenseCamera.jar，点击确定。

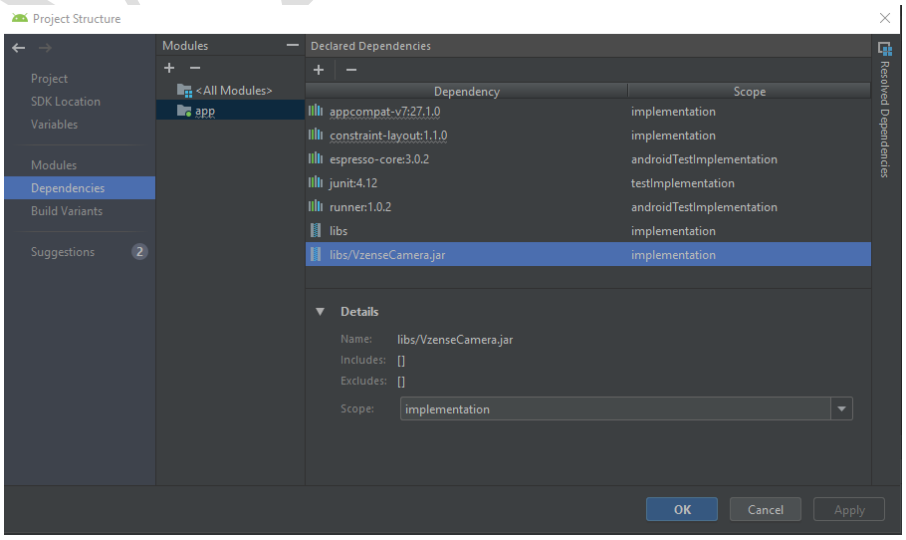

图5 导入jar文件

3.3.2. 导入 so 库文件

把 so 库拷贝到 app/libs 目录下，打开工程的 build.gradle 文件，添加以下代码，导入 so。导入库完成之后就可以在工程中调用接口进行开发，如下图 demo 示例：

```
android {
    sourceSets {
        main {
            jniLibs.srcDirs = ['libs']
        }
    }
}
```

图6 导入so文件

3.3.3. 接口调用流程

1．导入接口类

```
import com.vzense.sdk.PsCamera;
```

2．创建 PsCamera 类，进行初始化,需要传入当前设备连接状态的回调

```
mVzenseCamera  = new PsCamera();
if (mVzenseCamera  != null) {
    mVzenseCamera .init(this，mOnVzenseCameraConnectLister);
}
```

3．创建图像数据回调

```
mFrameCallback = new FrameCallback();
```

4．打开 camera，设置工作模式，以及图像回调

```
mVzenseCamera .setFrameCallback(mFrameCallback);
mVzenseCamera .setWorkMode(mWorkMode);// mWorkMode 默认为 1，人脸模式
mVzenseCamera .start(this);
```

5．在重载的回调函数里取出数据

```
public class FrameCallback implements IFrameCallback {
    @Override
    public void onFrame(PsFrame DepthFrame,PsFrame IrFrame,PsFrame RgbFrame)
{
            //DataProcess
             }
}
```

6．待设备启动连接正常后，在设备状态的回调中进行读取 Sn，fwVersion 等操作

```
@Override
public void onConnect() {
if (DEBUG) Log.i(TAG, "onConnect");
if(mVzenseCamera   != null) {
      String sn = mVzenseCamera .getSn();
      String fwVer = mVzenseCamera .getFWVerion();
      String hwVer = mVzenseCamera .getHWVerion();
      String sdkVersion = mVzenseCamera .getSDKVerion();
      String deviceName = mVzenseCamera .getDeviceName();
}
```

3.3.4. 工作模式切换流程描述

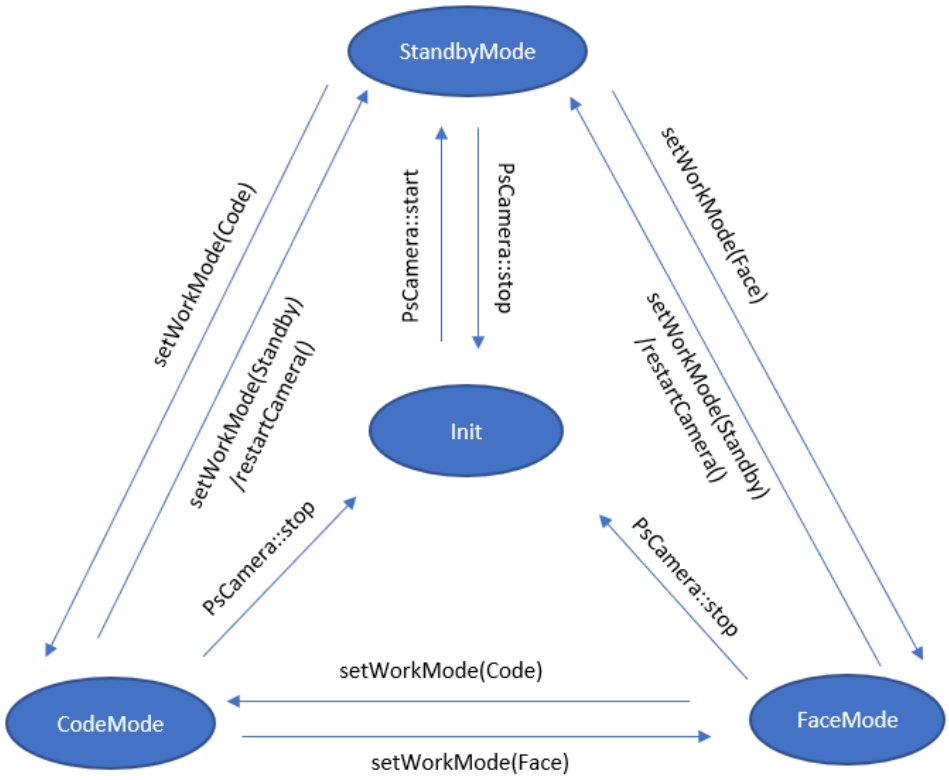

图 7 工作模式切换流程图

- 启动应用，通过 setWorkMode(Face)设置默认进入 FaceMode，此模式下默认以 Depth/IR/RGB 和单 RGB 交替输出图像，Depth，IR，RGB 图像的分辨率都为 480x640，Depth 和 IR 的帧率为 15Hz，RGB 图像为 30hz，应用从 SDK 获取的 Depth 和 IR 是与 RGB 对齐之后的图像。如果图像异常可以调用 restartCamera()重置相机，重置后工作模式切换为 Standby 模式。
- 通过 setWorkMode(Code)切换工作模式进入 CodeMode，此模式下 TOF 默认关闭，只有 RGB 图像，RGB 图像的分辨率为 1080x1920，可以通过 setRgbResolution 接口切换到其他分辨率，也可以通过 setTofFrameEnabled 开启或者关闭 TOF 图像数据。如果图像异常可以调用 restartCamera()重置相机，重置后工作模式切换为 Standby 模式。

- 调用 setWorkMode(Standby)切换进入到待机模式，此模式下 TOF 和 RGB 均默认关闭，只能通过 setWorkMode 切换回人脸或扫码模式来获取图像。
- 以上三种工作模式可以通过 setWorkMode 来实时切换，非待机模式下，RGB 图像分辨率以及 TOF 图像数据的开关都可以通过接口来实时切换。
- 在人脸模式和扫码模式下，可以打开或者关闭图像镜像和 TOF/RGB 自动曝光功能。图像镜像是默认关闭的，可以通过 setImageMirror 接口来切换镜像功能的开和关。TOF 和 RGB 的自动曝光是默认打开的，可以通过 setRgbAecEnabled 和 setTOFAecEnabled 来打开或者关闭自动曝光。在关闭自动曝光的前提下，可以通过 setRgbExposureTimeAndGain 来手动设定 RGB 的曝光时长和 gain 值。

3.3.5. restartCamera 接口使用流程

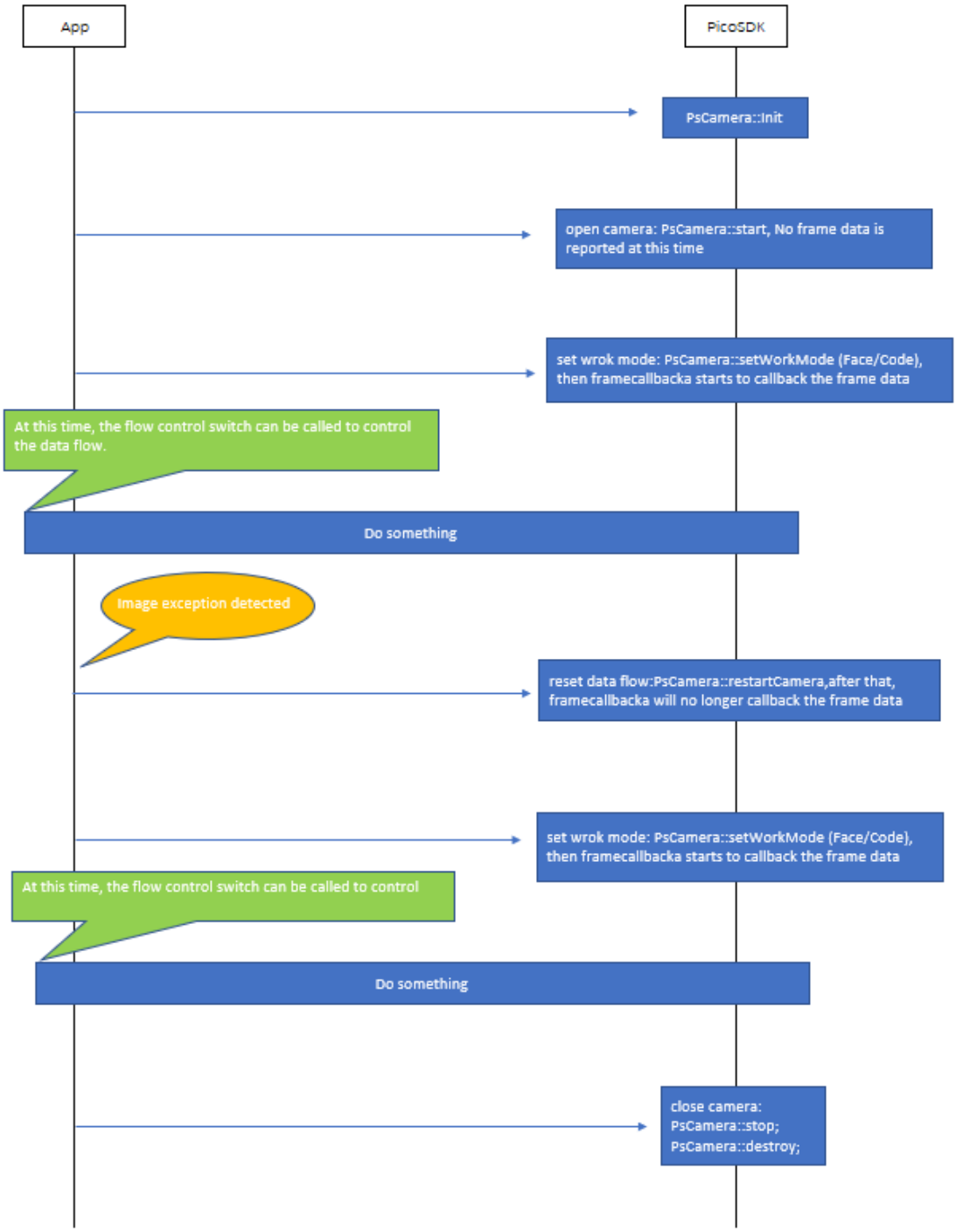

图 8 restartCamera 接口调用流程

3.4. 使用 SDK Sample 升级相机固件说明

把相机固件复制到 Android 设备的 sdcard 根目录，并重命名固件文件为 Firmware.img。运行 SDKSample，待正常显示图像后，点击【Upgrade】按钮，之后静待升级完成。升级过程中相机会多次重启，请不要手动拔插相机。升级过程中，升级进度会在页面上显示。

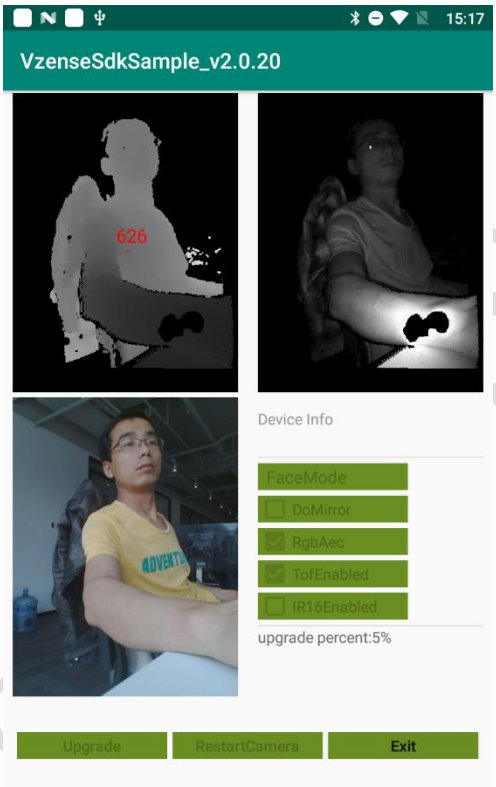

图 9 相机固件升级

4. SDK 接口介绍

4.1. Enum 数据类型

4.1.1. FrameType

功能:

图像数据流类型

枚举值:

- **DepthFrame** 表示 16 位深度图像流
- **IRFrame** 表示 16 位 IR 灰度图像流
- **RGBFrame** 表示 24 位 3 通道 RGB 图像流

4.1.2. PixelFormat

功能:

图像的像素类型

枚举值:

- **PixelFormatDepthMM16** 表示每像素数据为 16 位的深度值，以毫米为单位
- **PixelFormatGray16** 表示每像素数据为 16 位的灰度值
- **PixelFormatGray8** 表示每像素数据为 8 位的灰度值
- **PixelFormatRGB888** 表示每像素数据为 24 位的 RGB 值
- **PixelFormatBGR888** 表示每像素数据为 24 位的 BGR 值
- **PixelFormatRGBA8888** 表示每像素数据为 32 位的 RGBA 值

4.2. Class

4.2.1. CameraParameter

功能:

相机内参和畸变系数，说明见下面表格

成员：

参数	说明
fx，fy，cx，cy	相机的内参
k1，k2，k3，p1，p2	相机的畸变参数

4.2.2. CameraExtrinsicParameter

功能：

相机的外参

成员：

参数	说明
rotation[1-9]	TOF 相机到 RGB 相机的旋转矩阵
translation[1-3]	TOF 相机到 RGB 相机的平移矩阵
e[1-9]	输出本征矩阵
f[1-9]	输出基础矩阵

4.2.3. PsFrame

功能：

图像信息，说明见下面表格

成员：

参数	说明
frameIndex	帧号
frameType	图像数据类型

pixelFormat	像素类型
frameData	图像数据
dataLength	数据长度，以字节为单位
timeStamp	时间戳，单位为 ms
fps	帧率
width	图像的宽度
height	图像的长度
bytePerPixel	每个像素点的字节数

4.2.4. IFrameCallback

功能：

图像数据的回调接口

说明：

上层应用需要创建一个接口对象，通过 setFrameCallback 接口设置到 native 层，native 层会通过接口类的 OnFrame 方法把数据 callback 到应用层。

4.2.5. IUpgradeStatusCallback

功能：

升级状态的回调接口

说明：

上层应用需要创建一个接口对象，通过 setUpgradeStatusCallback 接口设置到 native 层，native 层会通过接口类的 onUpgradeStatus 方法把状态 callback 到应用层。

API	void onUpgradeStatus(int stage, int params)
说明	stage：升级状态。

	params：升级状态的结果。为-1 时，代表失败。 状态说明： ➢ DEVICE_PRE_UPGRADE_IMG_COPY ：复制 Firmware.img 固件到 Camera。 ➢ DEVICE_UPGRADE_IMG_CHECK_DOING ：校验 Firmware.img。 ➢ DEVICE_UPGRADE_IMG_CHECK_DONE ：校验 Firmware.img 完毕。 ➢ DEVICE_UPGRADE_UPGRAD_DOING ：升级固件中。params 此时代表进度百分比（0 ~ 100）。 ➢ DEVICE_UPGRADE_RECHECK_DOING ：升级后再校验。 ➢ DEVICE_UPGRADE_RECHECK_DONE ：升级后再校验完毕。 ➢ DEVICE_UPGRADE_UPGRAD_DONE ：升级完成。

4.2.6. PsCamera

功能：

接口类，用户可以通过此类来进行 camera 打开，关闭，设置参数，获取数据等操作，详细接口信息见下表

说明：

API	void init(Context context, final OnVzenseCameraConnectListener

	listener)
说明	SDK 的初始化，在启动的时候必须最先调用此接口，listen 为设备状态的回调。 如想获取设备连接状态，需要传入此回调，如果不需要获取则置为 null

API	void destroy()
说明	SDK 的资源释放，在 stop 之后调用

API	void start(Context context)
说明	开始图像采集，在 init 之后调用

API	void stop()
说明	停止图像采集

API	void setFrameCallback(final IFrameCallback callback)
说明	设置图像数据的回调函数，在 start 之前调用

API	int setGmmGain(int gmmGain)
说明	设置 IR 图像的 gmmgain 值，用于调整 IR 图像的亮度 **参数:** gmmGain: IR 图像的 gmmGain 值，取值范围为 0-4095 **返回值:**

	➢ 0: 成功 ➢ 其他: 失败

API	int getGmmGain()
说明	获取当前 IR 图像的 gmmgain 值 **返回值：**当前 IR 图像的 Gmmgain 值

API	int setRgbResolution(int resolutionIndex)
说明	设置 RGB 图像分辨率，参数取值范围为 0-3 **参数:** resolutionIndex: ➢ 0 : 1080x1920 ➢ 1 : 720x1280 ➢ 2 : 480x640 ➢ 3 : 360x640 **返回值:** ➢ 0: 成功 ➢ 其他: 失败

API	void getDepthCameraParameter(CameraParameter mDepthParameter)
说明	获取 TOF 相机内参，详细说明参见 4.2.1

API	void getRgbCameraParameter(CameraParameter mRgbParameter)
说明	获取 RGB 相机内参，详细说明参见 4.2.1

API	void getCameraExtrinsicParameter(CameraExtrinsicParameter mExtrinsicParameter)
说明	获取相机外参，详细说明参见 4.2.2

API	String getSn()
说明	获取设备的序列号，例如 PD3051AGD5130013M

API	String getFWVersion()
说明	获取设备的固件版本号，例如 2019.0518.02

API	String getHWVersion()
说明	获取设备的硬件版本号，例如 R2

API	String getSDKVersion()
说明	获取 SDK 软件版本号，例如 2.0.20

API	String getDeviceName()
说明	获取设备的名称，例如 Vzense RGBD DCAM305

API	int setWorkMode(int workMode)
说明	设置工作模式，可以设置的工作模式有三种,取值分别为 1,2,3 **参数:** workMode: ➢ 1: 人脸模式，此模式下默认为 TOF 数据 15hz，RGB 数据 30hz，TOF 和 RGB 分辨率都为 640x480，可以通过 setTofFrameEnabled 来打开或者关闭 TOF 数据，TOF 图像与 RGB 图像是同步以及对齐的 ➢ 2: 扫码模式，此模式下默认只有 RGB 数据，RGB 图像的分辨率为 640x480，用户可以通过 setTofFrameEnabled 来打开或者关闭 TOF 数据，如果打开了 TOF，TOF 图像与 RGB 图像是同步以及对齐的 ➢ 3: 待机模式，此模式下 TOF 和 RGB 均默认关闭，只能通过 setWorkMode 切换回人脸或扫码模式来获取图像 **返回值:** ➢ 0: 成功 ➢ 其他: 失败

API	int setRgbAecEnabled(boolean bEnabled)
说明	设置是否打开 RGB 相机的自动曝光,默认为 true **参数:** bEnabled: ➢ True: 打开 RGB 相机的自动曝光 ➢ False: 关闭 RGB 相机的自动曝光

	返回值: ➢ 0: 成功 ➢ 其他: 失败

API	int setTofFrameEnabled(boolean bEnabled)
说明	设置是否获取 TOF 图像数据，扫码模式下默认关闭 TOF，如果想获取 TOF 数据，需要调用此接口设置 TOF 状态为 true **参数:** bEnabled: ➢ True: 打开 TOF 数据 ➢ False: 关闭 TOF 数据 **返回值:** ➢ 0: 成功 ➢ 其他: 失败

API	int setImageMirror(int mirrorValue)
说明	设置图像的镜像处理 **参数:** mirrorValue: ➢ 0：不做镜像处理 ➢ 1：左右镜像 ➢ 2：上下镜像

	➢ 3：上下及左右镜像(旋转 180 度) **返回值:** ➢ 0: 成功 ➢ 其他: 失败

API	int setRgbExposureTimeAndGain(float exposureTime,float gain)
说明	设置 RGB 图像的曝光时长和 Gain 值，在此接口的调用之前，需要先调用 setRgbAecEnabled 接口先关闭自动曝光 **参数:** ➢ exposureTime : RGB 图像曝光时长，取值范围为[0.0015-0.03]，单位秒 ➢ gain：RGB 图像的 gain 值，取值范围为[1.0-15.5] **返回值:** ➢ 0 ： 成功 ➢ 其他 ： 失败

API	int restartCamera()
说明	重置相机，重置后工作模式变为Standby模式。 **参数:** 无 **返回值:** ➢ 0: 成功 ➢ 其他: 失败

API	int StartUpgradeFirmWare(String imagePath)
说明	启动 Camera 升级 **参数:** imagePath: 要刷机的镜像文件的路径。 **返回值:** ➢ 0：成功 ➢ 1：提示 build version 相同，可以升级。 ➢ -1：升级过程中，不要重复调用。 ➢ -2：固件检查失败 ➢ -3：版本太低，不支持升级

API	int setRgbFrameEnabled(boolean bEnabled)
说明	设置是否获取 RGB 图像数据。Standby 模式下设置无效。 **参数:** bEnabled: ➢ True: 打开 RGB 数据 ➢ False: 关闭 RGB 数据 **返回值:** ➢ 0: 成功 ➢ 其他: 失败

API	int setFramePixelFormat(PsFrame.FrameType type, PsFrame.PixelFormat format)
说明	设置图像的像素格式。 **参数:** type：图像类型。当前仅支持：IRFrame。 format：像素格式。 当前仅支持：PixelFormatGray16、PixelFormatGray8。 **返回值:** ➢ 0: 成功 ➢ 其他: 失败