

Berxel Android SDK 开发文档

修订历史

版本号	修订章节	修订记录	修订日期	作者
V1.0		初稿	2021.11.28	Allen

目录

1. 概述.....	6
1.1. SDK 介绍.....	6
1.2. SDK 兼容性.....	6
2. 开发工具包说明.....	6
2.1. SDK 开发工具包模块说明.....	6
2.2. SDK Sample 说明.....	7
2.3. SDK 调用说明.....	7
2.3.1. Android SDK API 调用步骤.....	7
2.3.2. 配置 AndroidManifest.xml.....	7
3. SDK 接口说明.....	7
3.1. BerxelHawkContext 模块说明.....	7
3.1.1. getBerxelContext.....	8
3.1.2. destroyBerxelContext.....	8
3.1.3. CreateDevice.....	8
3.1.4. addDeviceStatusCallBack.....	9
3.1.5. removeDeviceStatusCallBack.....	9
3.2. BerxelHawkDevice 模块说明.....	10
3.2.1. openDevice.....	10
3.2.2. closeDevice.....	10
3.2.3. startStreams.....	11
3.2.4. stopStreams.....	11
3.2.5. getSupportFrameModes.....	12

3.2.6. setFrameMode.....	12
3.2.7. getCurrentFrameMode.....	13
3.2.8. readColorFrame.....	13
3.2.9. readDepthFrame.....	14
3.2.10. readIrFrame.....	14
3.2.11. readLightIrFrame.....	15
3.2.12. startUpgrade.....	15
3.2.13. convertDepthToPointCloud.....	16
3.2.14. getVersions.....	16
3.2.15. getCurrentDeviceInfo.....	17
3.2.16. getCameraIntriscParams.....	17
3.2.17. getDeviceIntriscParams.....	18
3.2.18. setStreamMirror.....	18
3.2.19. setRegistrationEnable.....	18
3.2.20. setStreamFlagMode.....	19
3.2.21. setSystemClock.....	19
3.2.22. setDenoiseStatus.....	20
3.2.23. setTemperaTureCompensationStatus.....	20
3.2.24. enableColorAutoExposure.....	20
3.2.25. setColorExposureGain.....	21
3.2.26. setDepthAEStatus.....	21
3.2.27. setDepthGain.....	22

3.2.28. setDepthExposure.....	22
3.2.29. setDepthElectricCurrent.....	22
3.2.30. setDepthCloseRangeDefaultGainAndExposure.....	23
3.3. BerxelHawkFrame 模块说明.....	23
3.4. admitenum 与 admitmode 模块说明.....	24
3.4.1. BerxelHawkPixelFormatEnum 枚举说明.....	24
3.4.2. BerxelHawkStreamTypeEnum 枚举说明.....	24
3.4.3. BerxelHawkStreamFlagEnum 枚举说明.....	24
3.4.4. BerxelHawkDeviceInfo 类说明.....	27
3.4.5. BerxelHawkStreamFrameMode 类说明.....	27
3.4.6. BerxelHawkCameraIntrinsic 类说明.....	27
4. SDK 开发指引.....	28
4.1. SDK API 调用流程图.....	28
4.2. 获取彩色帧.....	28
4.3. 获取深度帧.....	29
4.4. 获取红外帧.....	29
4.5. 获取彩色+深度混合帧.....	29
4.6. 获取版本号.....	29
4.7. 获取当前设备信息.....	29
4.8. 深度图数据转换为深度值.....	29
4.8.1. BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D.....	30
4.8.2. BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D.....	30
4.9. 设置设备状态监听.....	30

1. 概述

1.1.SDK 介绍

Beroxel SDK 是基于 Beroxel 3D 摄像头的软件开发工具包。产品可广泛适用于工业控制、消费类电子等领域中对三维图像有要求的应用场景,支持 Android/Windows/Linux/ROS 等平台。

1.2.SDK 兼容性

- Android OS 5.1 及以上操作系统
- USB2.0 , 电流 2A, 支持 HOST
- RAM 2G 或以上

2. 开发工具包说明

2.1.SDK 开发工具包模块说明

模块名称	模块说明
Document	SDK 开发文档说明书
Lib	Beroxel Android SDK, 包含动态库与 Jar 包
Samples	SDK 例子程序源码

2.2.SDK Sample 说明

例子程序名称	例子说明
HawkColor	演示获取彩色图流程
HawkDepth	演示获取深度图流程
HawkIr	演示获取红外图流程
HawkColorDepth	演示获取 Mix 流的深度图+红外图流程（分辨率为 400*640）
HawkHDCoDepth	演示获取 Mix 流的彩色图+深度图流程（分辨率为 800*1280）
HawkDepthMattingColor	演示配准功能

2.3.SDK 调用说明

2.3.1.Android SDK API 调用步骤

- 将 SDK 开发包中 lib 文件夹中 armeabi/armeabi-v7/arm64-v8a 里的 so 库拷贝到 app\libs 工程目录下
- 将 SDK 开发包中 lib 文件夹中的 BeroxelSDK.jar 拷贝到 app\libs 工程目录下
- 修改开发工程 app 文件夹下的 build.gradle 配置，配置工程引用 so 库和 Jar 包

2.3.2.配置 AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

android.permission.INTERNET	允许 APP 联网功能
android.permission.WRITE_EXTERNAL_STORAGE	允许 APP 写功能
android.permission.READ_EXTERNAL_STORAGE	允许 APP 读功能

3. SDK 接口说明

3.1.BeroxelHawkContext 模块说明

BeroxelHawkContext 模块主要用于 SDK 的初始化和销毁，以及创建设备，具体参见 BeroxelSDK.jar 中 BeroxelContext.java 源文件。

3.1.1. getBernelContext

[Description]

获取 BernelHawkContext 对象实例

[Function]

BernelHawkContext getBernelContext(Context context)

[Params]

context[IN]: 应用上下文 (不能为 null)

[Return value]

BernelHawkContext 对象实例 : 创建成功

null: 创建失败

3.1.2. destroyBernelContext

[Description]

销毁 BernelHawkContext 对象实例

[Function]

static void destroyBernelContext()

[Params]

NULL

[Return value]

无

3.1.3. CreateDevice

[Description]

创建设备对象实例

[Function]

BernelHawkDevice CreateDevice()

[Params]

NULL

[Return value]

BernelHawkDevice 对象实例：创建成功
null：失败

[Remarks]

BernelHawkDevice 类参见 BernelSDK.jar 中 BernelHawkDevice.java 文件

3.1.4. addDeviceStatusCallBack

[Description]

增加设备状态监听回调，监听设备的连接和断开状态

[Function]

boolean addDeviceStatusCallBack(BernelHawkContext.DeviceStatusChangedCallBack
statusCallBack)

[Params]

statusCallBack[IN]: 设备状态函数接口

[Return value]

true: 成功
false: 失败

[Remarks]

DeviceStatusChangedCallBack 接口参见 BernelSDK.jar 中 BernelHawkContext.java 文件

3.1.5. removeDeviceStatusCallBack

[Description]

移除设备状态监听回调

[Function]

boolean removeDeviceStatusCallBack(BernelHawkContext.DeviceStatusChangedCallBack
statusCallBack)

[Params]

statusCallBack[IN]: 设备状态函数接口

[Return value]

true: 成功

false: 失败

[Remarks]

DeviceStatusChangedCallBack 接口参见 BernelSDK.jar 中 BernelHawkContext.java 文件。此函数和 addDeviceStatusCallBack 配对使用

3.2. BernelHawkDevice 模块说明

BernelHawkDevice 模块主要维护设备相关的操作，包括开/关设备、开/关流、读取数据流、读取设备信息等，参见 BernelSDK.jar 中 BernelHawkDevice.java 头文件。

3.2.1. openDevice

[Description]

打开设备

[Function]

```
public void openDevice(OpenDeviceStatusCallBack callBack)
```

[Params]

callBack[IN] : 设备打开成功或者失败回调，不能为 null

[Return value]

无

[Remarks]

OpenDeviceStatusCallBack 参见 BernelSDK.jar 中 BernelHawkDevice.java 文件

3.2.2. closeDevice

[Description]

关闭设备

[Function]

```
int closeDevice()
```

[Params]

NULL

[Return value]

0 : 成功
非 0: 失败

3.2.3. startStreams

[Description]

打开数据流

[Function]

```
int startStreams(int streamFlags)
```

[Params]

streamFlags[IN] : 所需打开流类型的值, 使用 BixelHawkStreamTypeEnum 定义的流类型

[Return value]

0 : 成功
非 0: 失败

[Remarks]

BixelHawkStreamTypeEnum : 流类型, 参照 3.4.2 章节

streamflags 参数说明:

streamFlags 参数	说明
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue()	打开彩色数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	打开深度数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_IR_STREAM</i> .getValue()	打开泛光源红外数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_LIGHT_IR_STREAM</i> .getValue()	打开点红外数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue() BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	同时打开彩色与深度数据流

3.2.4. stopStreams

[Description]

关闭数据流

[Function]

```
int stopStreams(int streamFlags)
```

[Params]

streamFlags[IN] : 所需关闭流类型的值, 使用 BixelHawkStreamTypeEnum 定义的流

类型

[Return value]

0 : 成功
非 0: 失败

[Remarks]

BixelHawkStreamTypeEnum : 流类型, 参照 3.4.2 章节

streamflags 参数说明:

streamFlags	说明
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue()	关闭彩色数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	关闭深度数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_IR_STREAM</i> .getValue()	关闭泛光源红外数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_LIGHT_IR_STREAM</i> .getValue()	关闭点红外数据流
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue() BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	同时关闭彩色与深度数据流

3.2.5. getSupportFrameModes

[Description]

获取数据流所支持的帧模式列表

[Function]

ArrayList<BixelHawkStreamFrameMode>
getSupportFrameModes(BixelHawkStreamTypeEnum streamType)

[Params]

streamType[IN]: 流类型

[Return value]

ArrayList<BixelHawkStreamFrameMode> : 当前数据流支持的帧模式列表
null : 获取失败

[Remarks]

BixelHawkStreamTypeEnum : 流类型, 参照 3.4.2 章节
BixelHawkStreamFrameMode: 帧模式类定义, 参照 3.4.5 章节

3.2.6. setFrameMode

[Description]

设置帧模式

[Function]

```
int setFrameMode(BernelHawkStreamTypeEnum streamType,  
                BernelHawkStreamFrameMode mode)
```

[Params]

streamType[IN]: 流类型

mode[in]: 帧模式类定义

[Return value]

0 : 成功

非0: 失败

[Remarks]

BernelHawkStreamTypeEnum : 流类型, 参照 3.4.2 章节

BernelHawkStreamFrameMode: 帧模式类定义, 参照 3.4.5 章节

3.2.7. getCurrentFrameMode

[Description]

获取当前的帧模式

[Function]

```
BernelHawkStreamFrameMode getCurrentFrameMode(BernelHawkStreamTypeEnum  
streamType)
```

[Params]

streamType[IN]: 流类型

[Return value]

BernelHawkStreamFrameMode : 当前数据流帧模式类

null: 获取失败

[Remarks]

BernelHawkStreamTypeEnum : 流类型, 参照 3.4.2 章节

BernelHawkStreamFrameMode: 帧模式类定义, 参照 3.4.5 章节

3.2.8. readColorFrame

[Description]

读取彩色数据帧

[Function]

```
BernelHawkFrame readColorFrame(int timeout)
```

[Params]

timeout[IN]: 超时时间(ms), 若超时未获取到数据帧, 则返回null

[Return value]

BernelHawkFrame : 获取到的数据帧

null: 失败

[Remarks]

BernelHawkFrame 为数据帧类, 参照3.3章节

3.2.9. readDepthFrame

[Description]

读取深度数据帧

[Function]

```
BernelHawkFrame readDepthFrame(int timeout)
```

[Params]

timeout[IN]: 超时时间(ms), 若超时未获取到数据帧, 则返回null

[Return value]

BernelHawkFrame: 获取到的数据帧

null: 失败

[Remarks]

BernelHawkFrame为数据帧类, 参照3.3章节

3.2.10. readIrFrame

[Description]

读取泛光源红外数据帧

[Function]

```
BernelHawkFrame readIrFrame(int timeout)
```

[Params]

timeout[IN]: 超时时间(ms), 若超时未获取到数据帧, 则返回null

[Return value]

BernelHawkFrame: 获取到的数据帧

null: 失败

[Remarks]

BernelHawkFrame为数据帧类，参照3.3章节

3.2.11. readLightIrFrame

[Description]

读取点光源红外数据帧

[Function]

BernelHawkFrame readLightIrFrame(int timeout)

[Params]

timeout[IN]: 超时时间(ms)，若超时未获取到数据帧，则返回null

[Return value]

BernelHawkFrame: 获取到的数据帧

null: 失败

[Remarks]

BernelHawkFrame为数据帧类，参照3.3章节

3.2.12. startUpgrade

[Description]

此接口用于设备进行升级

[Function]

int startUpgrade(String fwFilePath)

[Params]

fwFilePath[IN] : 需升级的固件路径

[Return value]

0 : 成功

非0: 失败

[Remarks]

此接口需配合 BernelHawkUpgrade 模块中 setUpgradeStatusCallBack 函数使用，setUpgradeStatusCallBack 用于监听设备升级的状态回调。

3.2.13. convertDepthToPointCloud

[Description]

将深度数据转换为点云数据

[Function]

```
int convertDepthToPointCloud(BernelHawkFrame frame, float factor,
FloatBuffer pointClouds)
```

[Params]

frame [IN]: 数据帧

factor[IN]: 若输出点云坐标以m为单位，传入1000.0，以mm为单位则输入1.0

pointClouds[OUT]:输出的点云数据

[Return value]

0 : 成功

非0: 失败

3.2.14. getVersions

[Description]

获取版本号

[Function]

```
BernelHawkVersion getVersions()
```

[Params]

NULL

[Return value]

BernelHawkVersion: 成功获取版本信息

null: 获取版本信息失败

[Remarks]

BernelHawkVersions 为版本信息类，参见 BernelSDK.jar 中 BernelHawkVersion.java 文件

3.2.15. `getCurrentDeviceInfo`

[Description]

获取当前打开设备的设备信息

[Function]

`BernelHawkDeviceInfo getCurrentDeviceInfo()`

[Params]

NULL

[Return value]

`BernelHawkDeviceInfo`: 成功获取设备信息

null: 获取设备信息失败

[Remarks]

`BernelHawkDeviceInfo`: 设备信息类, 参照 3.4.4 章节。

此函数需在打开设备成功以后调用。

3.2.16. `getCameraIntriscParams`

[Description]

获取红外相机内参, 此相机内参可用于深度数据转换为点云数据

[Function]

`BernelHawkCameraIntrinsic getCameraIntriscParams()`

[Params]

NULL

[Return value]

`BernelHawkCameraIntrinsic` : 成功获取相机内参

null: 获取相机内参失败

[Remarks]

`BernelHawkCameraIntrinsic`: 相机内参类, 参照 3.4.6 章节。

参数是以 400*640 (或 640*400) 分辨率标定的。

3.2.17. `getDeviceIntriscParams`

[Description]

获取当前设备内参、外参，包括相机彩色相机内参，红外相机内参，平移矩阵，旋转矩阵等参数

[Function]

```
int getDeviceIntriscParams(FloatBuffer deviceParams)
```

[Params]

deviceParams[out]: 设备相机内外参。

[Return value]

0 : 成功
非 0: 失败

[Remarks]

相机内参是以 800*1280（或 1280*800）分辨率标注的。参照 3.4.7 章节

3.2.18. `setStreamMirror`

[Description]

设置图像镜像，默认所有数据帧非镜像状态

[Function]

```
int setStreamMirror(boolean bMirror)
```

[Params]

bMirror[IN]: true 表示为镜像，false 表示为非镜像。

[Return value]

0 : 成功
非 0: 失败

3.2.19. `setRegistrationEnable`

[Description]

设置深度图和彩色图的配准功能是否打开，配准功能默认为关闭状态。

[Function]

```
int setRegistrationEnable(boolean bEnable)
```

[Params]

bEnable[in]: true 表示为打开配准, false 表示为关闭配准。

[Return value]

0 : 成功
非 0: 失败

3.2.20. setStreamFlagMode

[Description]

设置流模式, 分为单开模式, Mix VGA 流模式, Mix HD 流模式, 默认为 Mix VGA 流模式。此接口需要在开流接口之前调用。

[Function]

```
int setStreamFlagMode(BernelHawkStreamFlagEnum flagMode)
```

[Params]

flagMode[in]: 流模式状态。

[Return value]

0 : 成功
非 0: 失败

[Remarks]

BernelHawkStreamFlagEnum : 流模式, 参照 3.4.3 章节。

3.2.21. setSystemClock

[Description]

同步当前上位机系统时钟到相机中。此函数在开流之前调用, 可以使获取的数据帧的时间戳和上位机的系统时钟同步。

[Function]

```
int setSystemClock()
```

[Params]

NULL

[Return value]

0 : 成功

非 0：失败

3.2.22. setDenoiseStatus

[Description]

设置是否打开当前设备的降噪功能，此功能默认是打开状态。

[Function]

```
int32_t setDenoiseStatus(bool bEnable)
```

[Params]

bEnable[in]: true 表示打开降噪, false 表示关闭降噪, 默认是打开状态。

[Return value]

0 : 成功
非 0: 失败

3.2.23. setTemperatureCompensationStatus

[Description]

设置温度补偿功能是否打开，温度补偿功能默认为关闭状态。

[Function]

```
int setTemperatureCompensationStatus(boolean bEnable)
```

[Params]

bEnable[in]: true 表示为打开温补功能, false 表示为关闭温补功能。

[Return value]

0 : 成功
非 0: 失败

3.2.24. enableColorAutoExposure

[Description]

打开彩色 AE 功能

[Function]

```
int enableColorAutoExposure()
```

[Params]

NULL

[Return value]

0 : 成功
非 0: 失败

3.2.25. setColorExposureGain

[Description]

设置彩色图像的曝光时间与增益。

[Function]

```
int setColorExposureGain(int exposureTime, int gain)
```

[Params]

exposureTime[in]: 彩色图像曝光时间,可设置参数为10000, 20000, 30000这3个参数。
gain[in]: 彩色图像增益, 可设置参数范围在[100-300]之间。

[Return value]

0 : 成功
非 0: 失败

3.2.26. setDepthAESTatus

[Description]

设置深度图像 Auto Expoure 功能。

[Function]

```
int setDepthAESTatus(boolean bEnable)
```

[Params]

bEnable[in]: true 打开 AE 功能, false 关闭 AE 功能。

[Return value]

0 : 成功
非 0: 失败

3.2.27. setDepthGain

[Description]

设置深度图像的增益。

[Function]

```
int setDepthGain(int value)
```

[Params]

value[in]: 深度图像增益值, 可设置参数范围在[1-4]之间。

[Return value]

0 : 成功
非 0: 失败

3.2.28. setDepthExposure

[Description]

设置深度图像的曝光时间。

[Function]

```
int setDepthExposure(int value)
```

[Params]

value[in]: 深度图像曝光时间, 可设置参数范围在[1-43]之间。

[Return value]

0 : 成功
非 0: 失败

3.2.29. setDepthElectricCurrent

[Description]

设置深度 sensor 的电流。

[Function]

```
int setDepthElectricCurrent(int value)
```

[Params]

value[in]: 深度 sensor 的电流(单位 mA), 可设置范围在[200-2000]之间。

[Return value]

0 : 成功
非 0: 失败

3.2.30. setDepthCloseRangeDefaultGainAndExposure

[Description]

在近距离使用场景时，设置深度图的默认值 gain 值以及曝光时间，只支持 100H 设备。

[Function]

```
int setDepthCloseRangeDefaultGainAndExposure()
```

[Params]

NULL

[Return value]

0 : 成功
非 0: 失败

3.3. BeroxelHawkFrame 模块说明

此模块主要是获取读取帧的详细信息, 参见 BeroxelSDK.jar 中 BeroxelHawkFrame.java

文件。具体函数说明如下

函数名称	函数说明
int getPixelType()	获取帧图像的象数格式
int getStreamType()	获取帧图像的流类型
int getFrameIndex()	获取帧图像的帧号
long getTimeStamp()	获取帧图像时间戳
int getFPS()	获取帧图像的帧率
int getWidth()	获取帧图像的 X 轴分辨率
int getHeight()	获取帧图像的 Y 轴分辨率
ByteBuffer getData()	获取帧图像数据
int getDataSize()	获取帧图像的 size 大小

3.4.admitenum 与 admitmode 模块说明

3.4.1. BernelHawkPixelFormatEnum 枚举说明

此枚举表示像素格式，具体说明如下：

描述	说明
BERXEL_HAWK_PIXEL_TYPE_IMAGE_RGB24	彩色图像素格式，每个像素占用 3 字节
BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D	深度图像素格式，每个像素占用 2 字节，共 16 位，前 12 位表示整数部分，后 4 位表示小数部分
BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_13I_3D	深度图像素格式，每个像素占用 2 字节，共 16 位，前 13 位表示整数部分，后 3 位表示小数部分
BERXEL_HAWK_PIXEL_TYPE_IR_16BIT	红外图像素格式，每个像素占用 2 字节
BERXEL_HAWK_PIXEL_INVALID_TYPE	不支持像素格式

3.4.2. BernelHawkStreamTypeEnum 枚举说明

此枚举表示流类型，在调用开/关流函数中需要传入此参数，具体说明如下：

描述	说明
BERXEL_HAWK_COLOR_STREAM	彩色流类型
BERXEL_HAWK_DEPTH_STREAM	深度流类型
BERXEL_HAWK_IR_STREAM	泛红流类型
BERXEL_HAWK_LIGHT_IR_STREAM	点红外流类型

3.4.3. BernelHawkStreamFlagEnum 枚举说明

此枚举表示流模式类型，在 SINGULAR 模式下，只支持单一数据流的开关。在 MIX 模式下，支持多种数据流的混合开关。具体说明如下：

描述	说明
BERXEL_HAWK_SINGULAR_STREAM_FLAG_MODE	数据流为单开模式，同时只支持打开一种数据流
BERXEL_HAWK_MIX_STREAM_FLAG_MODE	数据流为混合流 VGA 模式，支持同时打开多种数据流
BERXEL_HAWK_MIX_HD_STREAM_FLAG_MODE	数据流为混合流 HD 模式，支持同时打开多种数据流

BERXEL_HAWK_MIX_QVGA_STREAM_FLAG_MODE	数据流为混合流 QVGA 模式，支持同时打开多种数据流
---------------------------------------	-----------------------------

iHawk100H（横版设备）各模式下支持分辨率如下：

流模式	彩色分辨率	深度分辨率	红外分辨率
BERXEL_HAWK_SINGULAR_STREAM_FLAG_MODE	1920*1080@30fps 1280*800@30fps 640*400@30fps 320*200@30fps	320*200@5fps 320*200@10fps 320*200@15fps 320*200@20fps 320*200@25fps 320*200@30fps 640*400@5fps 640*400@10fps 640*400@15fps 640*400@20fps 640*400@25fps 640*400@30fps 1280*800@10fps	640*400@30fps
BERXEL_HAWK_MIX_STREAM_FLAG_MODE	640*400@30fps	640*400@5fps 640*400@10fps 640*400@15fps 640*400@20fps 640*400@25fps 640*400@30fps	640*400@30fps
BERXEL_HAWK_MIX_HD_STREAM_FLAG_MODE	1280*800@10fps	1280*800@10fps	不支持
BERXEL_HAWK_MIX_QVGA_STREAM_FLAG_MODE	320*240@30fps	320*200@5fps 320*200@10fps 320*200@15fps 320*200@20fps 320*200@25fps 320*200@30fps	不支持

iHawk100V（竖版设备）各模式下支持分辨率如下：

流模式	彩色分辨率	深度分辨率	红外分辨率
BERXEL_HAWK_SINGULAR_STREAM_FLAG_MODE	1080*1920@30fps 800*1280@30fps 400*640@30fps	200*320@5fps 200*320@10fps 200*320@15fps	400*640@30fps

	200*320@30fps	200*320@20fps 200*320@25fps 200*320@30fps 400*640@5fps 400*640@10fps 400*640@15fps 400*640@20fps 400*640@25fps 400*640@30fps 800*1280@10fps	
BERXEL_HAWK_MIX_STREAM_FLAG_MODE	400*640@30fps	400*640@5fps 400*640@10fps 400*640@15fps 400*640@20fps 400*640@25fps 400*640@30fps	400*640@30fps
BERXEL_HAWK_MIX_HD_STREAM_FLAG_MODE	800*1280@10fps	800*1280@10fps	不支持
BERXEL_HAWK_MIX_QVGA_STREAM_FLAG_MODE	200*320@30fps	200*320@5fps 200*320@10fps 200*320@15fps 200*320@20fps 200*320@25fps 200*320@30fps	不支持

iHawk100E(网络设备)各模式下支持分辨率如下:

流模式	彩色分辨率	深度分辨率	红外分辨率
BERXEL_HAWK_SINGULAR_STREAM_FLAG_MODE	640*400@30fps 1280*800@30fps 1920*1080@30fps	640*400@30fps 1280*800@10fps	640*400@30fps
BERXEL_HAWK_MIX_STREAM_FLAG_MODE	640*400@30fps	640*400@30fps	640*400@30fps
BERXEL_HAWK_MIX_HD_STREAM_FLAG_MODE	1280*800@10fps	1280*800@10fps	不支持
BERXEL_HAWK_MIX_QVGA_STREAM_FLAG_MODE	不支持	不支持	不支持

3.4.4. BeroxelHawkDeviceInfo 类说明

此类表示设备信息。具体说明如下：

描述	说明
<code>int getVendorId()</code>	获取供应商识别码
<code>int getProductId()</code>	获取产品识别码
<code>String getSerialNumber()</code>	获取设备序列号

3.4.5. BeroxelHawkStreamFrameMode 类说明

此类表示帧模式，在设置和获取 `FrameMode` 中需要使用。具体说明如下：

描述	说明
<code>BeroxelHawkPixelFormatEnum getPixelFormat()</code>	获取帧像素格式，参照 3.4.1 章节
<code>int getResolutionX()</code>	获取帧模式横向分辨率
<code>int getResolutionY()</code>	获取帧模式纵向分辨率
<code>int getmFps()</code>	获取帧率
<code>void setPixelFormat(BeroxelHawkPixelFormatEnum pixelType)</code>	设置帧像素格式，参照 3.4.1 章节
<code>void setResolutionX(int resolutionX)</code>	设置帧模式横向分辨率
<code>void setResolutionY(int resolutionY)</code>	设置帧模式纵向分辨率
<code>void setFps(int fps)</code>	设置帧率

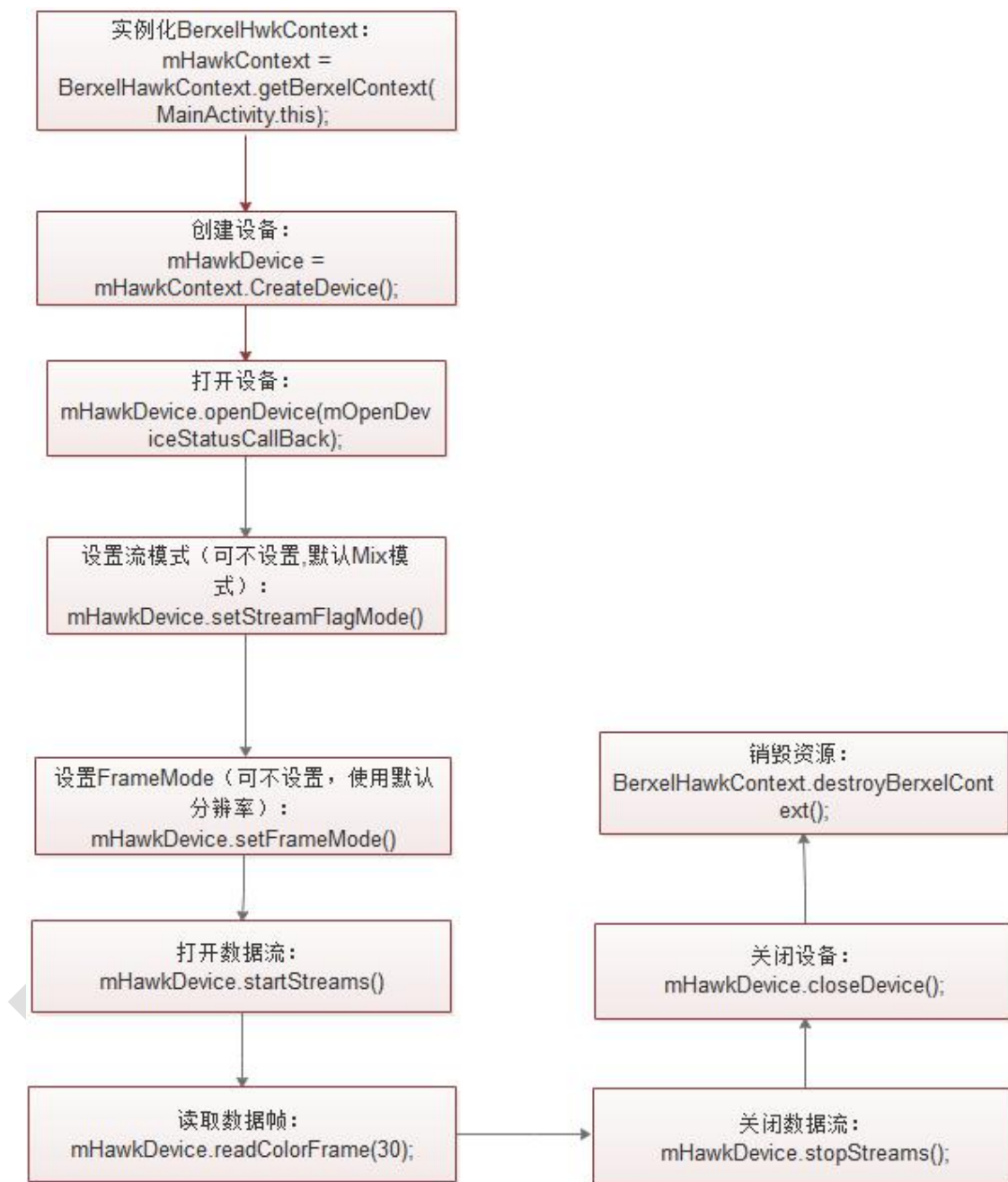
3.4.6. BeroxelHawkCameraIntrinsic 类说明

此结类表示相机内参，相机内参在转换点云数据时候使用，具体说明如下：

描述	说明
<code>float getFxParam()</code>	X 方向焦距
<code>float getFyParam()</code>	Y 方向焦距
<code>float getCxParam()</code>	X 方向主光轴点位置
<code>float getCyParam()</code>	Y 方向主光轴点位置
<code>float getK1Param()</code>	径向畸变 K1
<code>float getK2Param()</code>	径向畸变 K2
<code>float getP1Param()</code>	切向畸变 p1
<code>float getP2Param()</code>	切向畸变 p2
<code>float getK3Param()</code>	径向畸变 K3

4. SDK 开发指引

4.1. SDK API 调用流程图



4.2. 获取彩色帧

参照 SDK 开发工具包 Samples 模块中 HawkColor 例子程序。

4.3. 获取深度帧

参照 SDK 开发工具包 Samples 模块中 HawkDepth 例子程序。

4.4. 获取红外帧

参照 SDK 开发工具包 Samples 模块中 HawkIr 例子程序。

4.5. 获取彩色+深度混合帧

参照 SDK 开发工具包中 Samples 中 HawkColorDepth 例子程序。

4.6. 获取版本号

版本号类为在 `BeroxelHawkVersion.java` 中，包括 SDK 版本号、固件版本号、硬件版本号。示例代码如下：

```
BeroxelHawkVersion version = null;
version = mHawkDevice.getVersions();
```

4.7. 获取当前设备信息

设备信类构在 `BeroxelHawkDeviceInfo.java` 文件中，包括 SN、vendorId、productId 信息，此接口在设备打开成功以后调用，示例代码如下：

```
BeroxelHawkDeviceInfo deviceInfo = null;
deviceInfo = mHawkDevice.getCurrentDeviceInfo();
```

4.8. 深度图数据转换为深度值

深度图有两种格式，根据 `BeroxelHawkFrame` 中 `getPixelType` 接口获取的 `PixelType` 来区分，分别是 `BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D` 和 `BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_13I_3D`，两者的精度和有效距离不一致。

4.8.1. BEROXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D

深度图每帧 Raw 数据共 16 位,前面 12 位为整数部分,单位是 1mm,后面 4 位位小数部分,单位是 0.0625mm。转换步骤如下:

$\text{float depth} = \text{oriDepth} / 16$ (oriDepth 表示 16 位深度图);

4.8.2. BEROXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D

深度图每帧 Raw 数据共 16 位,前面 13 位为整数部分,单位是 1mm,后面 3 位位小数部分,单位是 0.125mm。转换步骤如下:

$\text{float depth} = \text{oriDepth} / 8$ (oriDepth 表示 16 位深度图);

4.9. 设置设备状态监听

此接口是为监听设备的断开和连接状态,具体示例代码如下:

```
private class MainDeviceStatusChangedCallBack implements BeroxelHawkContext.DeviceStatusChangedCallBack {  
    @Override  
    public void onDeviceStatusChanged(int vid, int pid, BeroxelHawkDeviceStatusEnum status) {  
        if(status == BeroxelHawkDeviceStatusEnum.BEROXEL_HAWK_DEVICE_STATUS_DISCONNECT) {  
        }  
        else {  
        }  
    }  
}  
  
mDeviceStatusChangeCallBack = new MainDeviceStatusChangedCallBack();  
mHawkContext.addDeviceStatusCallBack(mDeviceStatusChangeCallBack);
```

Berxel Confidential