

# **Berxel Android SDK Document**

## 目录

1.	Abstract.....	5
1.1.	SDK Introduction .....	5
1.2.	SDK Compatibility.....	5
2.	Description of Development kit.....	5
2.1.	SDK Development Kit Module .....	5
2.2.	SDK Sample Program .....	5
2.3.	Call SDK API Steps.....	6
2.3.1.	Config AndroidManifest.xml .....	6
3.	SDK Interface Description.....	6
3.1.	BeroxelHawkContext Module Description.....	6
3.1.1.	getBeroxelContext.....	7
3.1.2.	destroyBeroxelContext.....	7
3.1.3.	CreateDevice .....	7
3.1.4.	addDeviceStatusCallBack .....	8
3.1.5.	removeDeviceStatusCallBack .....	8
3.2.	BeroxelHawkDevice Module Description.....	8
3.2.1.	openDevice.....	9
3.2.2.	closeDevice.....	9
3.2.3.	startStreams .....	9
3.2.4.	stopStreams.....	10
3.2.5.	getSupportFrameModes .....	11
3.2.6.	setFrameMode .....	11

3.2.7.	getCurrentFrameMode .....	11
3.2.8.	readColorFrame .....	12
3.2.9.	readDepthFrame .....	12
3.2.10.	readIrrFrame .....	13
3.2.11.	readLightIrrFrame.....	13
3.2.12.	startUpgrade .....	13
3.2.13.	convertDepthToPointCloud.....	14
3.2.14.	getVersions.....	14
3.2.15.	getCurrentDeviceInfo.....	15
3.2.16.	getCameraIntriscParams .....	15
3.2.17.	getDeviceIntriscParams.....	15
3.2.18.	setStreamMirror .....	16
3.2.19.	setRegistrationEnable .....	16
3.2.20.	setStreamFlagMode.....	16
3.2.21.	setSystemClock .....	17
3.2.22.	setDenoiseStatus .....	17
3.2.23.	setFrameSync.....	17
3.2.24.	setTemperaTureCompensationStatus .....	18
3.2.25.	enableColorAutoExposure .....	18
3.2.26.	setColorExposureGain.....	18
3.2.27.	setDepthAStatus .....	19
3.2.28.	setDepthGain .....	19

3.2.29.	setDepthExposure.....	19
3.2.30.	setDepthElectricCurrent .....	20
3.2.31.	enableDeviceSlaveMode.....	20
3.2.32.	getDeviceMasterSlaveMode .....	20
3.2.33.	setEdgeOptimizationStatus.....	21
3.2.34.	enableHightFpsMode.....	21
3.2.35.	setDeviceTransferMode .....	21
3.3.	BixelHawkFrame Module Description .....	22
3.4.	admitenum 与 admitmode Module Description .....	22
3.4.1.	BixelHawkPixelFormatEnum Module Description .....	22
3.4.2.	BixelHawkStreamTypeEnum Enumeration Description .....	23
3.4.3.	BixelHawkStreamFlagEnum Enumeration Description .....	23
3.4.4.	BixelHawkDeviceInfo Class Description .....	24
3.4.5.	BixelHawkStreamFrameMode Class Description .....	25
3.4.6.	BixelHawkCameraIntrinsic 类说明 .....	25
4.	SDK Guidelines .....	26
4.1.	Call SDK API Flowchart.....	26
4.2.	Get Color Frame .....	26
4.3.	Get Depth Frame.....	27
4.4.	Get Color and Depth Mix Frame .....	27
4.5.	Get Version.....	27
4.6.	Get Current Device Info .....	27
4.7.	Depth Raw Data Converted To The Distance Value .....	27
4.7.1.	BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D.....	27
4.7.2.	BERXEL_HAWK_PIXEL_TYPE_DEP_16BIT_12I_4D.....	28
4.8.	Set Device Status Monitoring .....	28

# 1. Abstract

## 1.1. SDK Introduction

The Bernel SDK is a software development kit based on the Bernel 3D camera. The product can be widely applied to the application scenarios requiring 3D images in industrial control, consumer electronics and other fields. The SDK supports the Android/Windows/Linux/ROS platform.

## 1.2. SDK Compatibility

- 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. Description of Development kit

## 2.1. SDK Development Kit Module

Module Name	Description
Document	Bernel SDK Development Documentation
Lib	Bernel Android SDK
Samples	Sample program source code

## 2.2. SDK Sample Program

Sample name	Description
-------------	-------------

HawkColor	Demonstrate the process of obtaining color frame
HawkDepth	Demonstrate the process of obtaining depth frame
HawkIr	Demonstrate the process of obtaining ir frame
HawkColorDepth	Demonstrate the process of obtaining color and depth frame (resolution is 640 * 400)
HawkHDCoDepth	Demonstrate the process of obtaining color and depth frame(resolution is 1280 * 800)
HawkDepthMattingColor	Demonstrate registration function

## 2.3. Call SDK API Steps

- Copy the so library in armeabi/armeabi-v7/arm64-v8a in the lib folder of the SDK to the app\libs project directory
- Copy BernelSDK.jar from the lib folder in the SDK package to the app\libs project directory
- Modify the build.gradle configuration in the development project app folder and configure the project reference so library and Jar package

### 2.3.1. Config 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	Allow APP networking permissions
android.permission.WRITE_EXTERNAL_STORAGE	Allows APP write function
android.permission.READ_EXTERNAL_STORAGE	Allows APP read function

## 3. SDK Interface Description

### 3.1. BernelHawkContext Module Description

The “BernelHawkContext” module is mainly used to maintain the device list and open or close device. Please refer to the file “BernelContext.java” in the package BernelSDK.jar.

### 3.1.1. getBernelContext

**[Description]**

Gets the instance of the BernelHawkContext object.

**[Function]**

```
BernelHawkContext getBernelContext(Context context)
```

**[Params]**

context[IN]: App Context (cannot be null)

**[Return value]**

The instance of the BernelHawkContext object: Successfully created

null: Failed to create

### 3.1.2. destroyBernelContext

**[Description]**

Destory the instance of the BernelHawkContext object.

**[Function]**

```
static void destroyBernelContext()
```

**[Params]**

NULL

**[Return value]**

None

### 3.1.3. CreateDevice

**[Description]**

Create a device object instance

**[Function]**

```
BernelHawkDevice CreateDevice()
```

**[Params]**

NULL

**[Return value]**

The instance of the BernelHawkDevice object: Successfully created

null: Failed to create

**[Remarks]**

The BernelHawkDevice class is referred to in the BernelHawkDevice.java file in BernelSDK.jar

### 3.1.4. addDeviceStatusCallBack

**[Description]**

Added the device status monitoring callback to monitor the connection and disconnection status of the device

**[Function]**

boolean addDeviceStatusCallBack(BernelHawkContext.DeviceStatusChangedCallBack statusCallBack)

**[Params]**

statusCallBack[IN]: Device state callback interface

**[Return value]**

true: Success  
false: Failed

**[Remarks]**

The DeviceStatusChangedCallBack interface is referred to in the BernelHawkContext.java file in BernelSDK.jar

### 3.1.5. removeDeviceStatusCallBack

**[Description]**

Remove the device status monitoring callback

**[Function]**

boolean removeDeviceStatusCallBack(BernelHawkContext.DeviceStatusChangedCallBack statusCallBack)

**[Params]**

statusCallBack[IN]: Device state callback interface

**[Return value]**

true: Success  
false: Failed

**[Remarks]**

The DeviceStatusChangedCallBack interface is referred to in the BernelHawkContext.java file in BernelSDK.jar.

## 3.2. BernelHawkDevice Module Description

The “BernelHawkDevice” module is mainly used to maintain the related operations of the device, including open/close the stream, read the data from the device and get the device information, Please refer to the file



“BernelHawkDevice.java” in the package BernelSDK.jar.

### 3.2.1. openDevice

**[Description]**

Open device

**[Function]**

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

**[Params]**

callBack[IN] : Device success or failure callback. The value cannot be null

**[Return value]**

None

**[Remarks]**

The OpenDeviceStatusCallBack interface is referred to in the BernelHawkDevice.java file in BernelSDK.jar

### 3.2.2. closeDevice

**[Description]**

Close device

**[Function]**

```
int closeDevice()
```

**[Params]**

NULL

**[Return value]**

0 : Success

Other: Failed

### 3.2.3. startStreams

**[Description]**

Open stream

**[Function]**

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

**[Params]**

streamFlags[IN] : The value for the desired open stream type, using the stream type defined by BernelHawkStreamTypeEnum

**[Return value]**

0 : Success

Other: Failed

#### [Remarks]

BixelHawkStreamTypeEnum : Stream type, see section 3.4.2

Streamflags:

streamFlags 参数	说明
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue()	Open rgb stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	Open depth stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_IR_STREAM</i> .getValue()	Open ir stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_LIGHT_IR_STREAM</i> .getValue()	Open light ir stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue()   BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	Open both rgb and depth streams

### 3.2.4. stopStreams

#### [Description]

Stop stream

#### [Function]

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

#### [Params]

streamFlags[IN] : The value for the desired close stream type, using the stream type defined by BixelHawkStreamTypeEnum

#### [Return value]

0 : Success

Other: Failed

#### [Remarks]

BixelHawkStreamTypeEnum : Stream type, see section 3.4.2

streamflags 参数说明:

streamFlags	说明
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue()	Close rgb stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	Close depth stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_IR_STREAM</i> .getValue()	Close ir stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_LIGHT_IR_STREAM</i> .getValue()	Close light ir stream
BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_COLOR_STREAM</i> .getValue()   BixelHawkStreamTypeEnum. <i>BERXEL_HAWK_DEPTH_STREAM</i> .getValue()	Close both rgb and depth streams

### 3.2.5. getSupportFrameModes

**[Description]**

Gets a list of frame modes supported by the data stream

**[Function]**

```
ArrayList<BernelHawkStreamFrameMode>  
getSupportFrameModes (BernelHawkStreamTypeEnum streamType)
```

**[Params]**

streamType[IN]: Stream type

**[Return value]**

ArrayList<BernelHawkStreamFrameMode> : List of frame modes supported by the current data stream  
null : Failed

**[Remarks]**

BernelHawkStreamTypeEnum : Stream type, see section 3.4.2  
BernelHawkStreamFrameMode: Frame mode, see section 3.4.5

### 3.2.6. setFrameMode

**[Description]**

Set frame mode

**[Function]**

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

**[Params]**

streamType[IN]: Stream type  
mode[in]: Frame mode

**[Return value]**

0 : Success  
Other: Failed

**[Remarks]**

BernelHawkStreamTypeEnum : Stream type, see section 3.4.2  
BernelHawkStreamFrameMode: Frame mode, see section 3.4.5

### 3.2.7. getCurrentFrameMode

**[Description]**

Gets the current frame mode

**[Function]**

```
BernelHawkStreamFrameMode getCurrentFrameMode (BernelHawkStreamTypeEnum
```

streamType)

**[Params]**

streamType[IN]: Stream type

**[Return value]**

BernelHawkStreamFrameMode : Current frame mode

null: Failed

**[Remarks]**

BernelHawkStreamTypeEnum : Stream type, see section 3.4.2

BernelHawkStreamFrameMode: Frame mode, see section 3.4.5

### 3.2.8. readColorFrame

**[Description]**

Read the rgb frame

**[Function]**

BernelHawkFrame readColorFrame(int timeout)

**[Params]**

timeout[IN]: Timeout (ms). If no data frame is obtained during timeout, null is returned

**[Return value]**

BernelHawkFrame : Data frame

null: Failed

**[Remarks]**

BernelHawkFrame: Data frame class, see section 3.3.

### 3.2.9. readDepthFrame

**[Description]**

Read the depth frame

**[Function]**

BernelHawkFrame readDepthFrame(int timeout)

**[Params]**

timeout[IN]: Timeout (ms). If no data frame is obtained during timeout, null is returned.

**[Return value]**

BernelHawkFrame: Data frame

null: Failed

**[Remarks]**

BernelHawkFrame: Data frame class, see section 3.3.

### 3.2.10. readIrFrame

**[Description]**

Read the ir frame.

**[Function]**

BernelHawkFrame readIrFrame(int timeout)

**[Params]**

timeout[IN]: Timeout (ms). If no data frame is obtained during timeout, null is returned.

**[Return value]**

BernelHawkFrame: Data frame

null: Faile

**[Remarks]**

BernelHawkFrame: Data frame class, see section 3.3.

### 3.2.11. readLightIrFrame

**[Description]**

Read the light ir frame

**[Function]**

BernelHawkFrame readLightIrFrame(int timeout)

**[Params]**

timeout[IN]: Timeout (ms). If no data frame is obtained during timeout, null is returned.

**[Return value]**

BernelHawkFrame: Data frame

null: Failed

**[Remarks]**

BernelHawkFrame: Data frame class, see section 3.3.

### 3.2.12. startUpgrade

**[Description]**

This interface is used to upgrade the device.

**[Function]**

int startUpgrade(String fwFilePath)

**[Params]**

fwFilePath[IN] : Upgrade file path

**[Return value]**

0 : Success  
Other: Failed

**[Remarks]**

This interface is used with the setUpgradeStatusCallBack function in the BernelHawkUpgrade module. setUpgradeStatusCallBack is used to listen for the status callback of a device upgrade.

### 3.2.13. convertDepthToPointCloud

**[Description]**

Convert depth data to point cloud data

**[Function]**

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

**[Params]**

frame [IN]: data frame  
factor[IN]: If the output point cloud coordinates are in m, enter 1000.0, and enter 1.0 in mm  
pointClouds[OUT]: Output point cloud data

**[Return value]**

0 : Success  
Other: Failed

### 3.2.14. getVersions

**[Description]**

Get device version.

**[Function]**

```
BernelHawkVersion getVersions()
```

**[Params]**

NULL

**[Return value]**

BernelHawkVersion: Device Version  
null: Failed.

**[Remarks]**

The BernelHawkVersions class is referred to in the BernelHawkDevice.java file in BernelSDK.jar

### 3.2.15. `getCurrentDeviceInfo`

**[Description]**

Gets information about the currently open device

**[Function]**

`BernelHawkDeviceInfo getCurrentDeviceInfo()`

**[Params]**

NULL

**[Return value]**

`BernelHawkDeviceInfo`: Device info

null: Failed

**[Remarks]**

`BernelHawkDeviceInfo`: Device info, see section 3.4.4

### 3.2.16. `getCameraIntriscParams`

**[Description]**

Get camera intrisc params.

**[Function]**

`BernelHawkCameraIntrinsic getCameraIntriscParams()`

**[Params]**

NULL

**[Return value]**

`BernelHawkCameraIntrinsic` : Camera intrisc params(640 \* 400)

null: Failed.

**[Remarks]**

`BernelHawkCameraIntrinsic`: Camera intrisc params, see section 3.4.6.

Calibration based on 640 \* 400 resolution.

### 3.2.17. `getDeviceIntriscParams`

**[Description]**

Obtain the internal and external parameters of the current device, including camera color camera internal parameters, infrared camera internal parameters, translation matrix, rotation matrix, etc

**[Function]**

`int getDeviceIntriscParams(FloatBuffer deviceParams)`

**[Params]**

`deviceParams[out]`: device params

**[Return value]**

0 : Success  
Other: Failed

**[Remarks]**

Calibration based on 640 \* 400 resolution. See section 3.4.7.

### 3.2.18. setStreamMirror

**[Description]**

Set image mirror.

**[Function]**

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

**[Params]**

bMirror[IN]: true : Mirrored, false: Not mirrored.

**[Return value]**

0 : Success  
Other: Failed

### 3.2.19. setRegistrationEnable

**[Description]**

Set the status of the registration function for depth and color maps. The registration function is off by default.

**[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 : Success  
Other: Failed

### 3.2.20. setStreamFlagMode

**[Description]**

Set the stream mode, which can be single-open mode, Mix VGA mode, Mix



HD mode. The default is Mix VGA mode. This interface needs to be called before the open stream interface.

**[Function]**

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

**[Params]**

flagMode[in]: Stream flag.

**[Return value]**

0 : Success

Other: Failed

**[Remarks]**

BernelHawkStreamFlagEnum : Stream flag, see section 3.4.3.

### 3.2.21. setSystemClock

**[Description]**

Synchronize the current host system clock to the camera.

**[Function]**

```
int setSystemClock()
```

**[Params]**

NULL

**[Return value]**

0 : Success

Other: Failed

### 3.2.22. setDenoiseStatus

**[Description]**

Set the depth image noise reduction function status.

**[Function]**

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

**[Params]**

bEnable[in]: true: Enable noise reduction, false: disable noise reduction.

**[Return value]**

0 : Success

Other: Failed

### 3.2.23. setFrameSync

**[Description]**

Set the frame synchronization status.

**[Function]**

```
int setFrameSync(boolean bEnable)
```

**[Params]**

bEnable[in]: true: Open the frame synchronization status, false : Close the frame synchronization status.

**[Return value]**

0 : Success  
Other: Failed

### 3.2.24. setTemperaTureCompensationStatus

**[Description]**

Set the temperature compensation function. The temperature compensation function is disabled by default.

**[Function]**

```
int setTemperaTureCompensationStatus(boolean bEnable)
```

**[Params]**

bEnable[in]: true: Open the temperature compensation, false: Close the temperature compensatuin.

**[Return value]**

0 : Success  
Other: Failed

### 3.2.25. enableColorAutoExposure

**[Description]**

Turn on the Color AE function

**[Function]**

```
int enableColorAutoExposure()
```

**[Params]**

NULL

**[Return value]**

0 : Success  
Other: Failed

### 3.2.26. setColorExposureGain

**[Description]**

Set exposure time and gain for color images.

**[Function]**

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

**[Params]**

exposureTime[in]: Color image exposure time. The range[1000-20000]

gain[in]: Color image gain. The range[100-300]

**[Return value]**

0 : Success

Other: Failed

### 3.2.27. setDepthAESTatus

**[Description]**

Set the depth image Auto Expoure function.

**[Function]**

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

**[Params]**

bEnable[in]: true : Open AE status, false: close AE status.

**[Return value]**

0 : Success

Other: Failed

### 3.2.28. setDepthGain

**[Description]**

Set the gain of the depth image.

**[Function]**

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

**[Params]**

value[in]: Depth gain. The range [1-4]

**[Return value]**

0 : Success

Other: Failed

### 3.2.29. setDepthExposure

**[Description]**

Set the exposure time for the depth image.

**[Function]**

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

**[Params]**

value[in]: Exposure time. The range [1-43]

**[Return value]**

0 : Success

Other: Failed

### 3.2.30. setDepthElectricCurrent

**[Description]**

Set the laser working current.

**[Function]**

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

**[Params]**

value[in]: The laser working current (mA), range [800-2000] .

**[Return value]**

0 : Success

Other: Failed

### 3.2.31. enableDeviceSlaveMode

**[Description]**

Set the device master/slave mode.

**[Function]**

```
int enableDeviceSlaveMode (boolean bEnable)
```

**[Params]**

bEnable[in]: true:Set slave mode false:Set master mode

**[Return value]**

0 : Success

Other: Failed

### 3.2.32. getDeviceMasterSlaveMode

**[Description]**

Gets the device's master/slave mode status.

**[Function]**

```
boolean getDeviceMasterSlaveMode ()
```

**[Params]**

NULL

**[Return value]**

true : Slave mode

false: Master mode

### 3.2.33. setEdgeOptimizationStatus

**[Description]**

Set edge optimization status.

**[Function]**`int setEdgeOptimizationStatus(boolean bEnable)`**[Params]**

bEnable [in]: true : Enable edge optimization false : Disable

**[Return value]**

0 : Success

Other: Failed

### 3.2.34. enableHightFpsMode

**[Description]**

Set high frame rate mode, which is disabled by default

**[Function]**`int enableHightFpsMode(boolean bEnable)`**[Params]**

bEnable [in]: true : Enable high frame mode false : Disable

**[Return value]**

0 : Success

Other: Failed

### 3.2.35. setDeviceTransferMode

**[Description]**

Set the USB device transfer mode

**[Function]**`int setDeviceTransferMode(BernelHawkUVCModeEnum mode)`**[Params]**

mode[in]: Device transfer mode, ISOC: BERNEL\_HAWK\_DEVICE\_ISOC\_MODE

BULK: BERNEL\_HAWK\_DEVICE\_BULK\_MODE

**[Return value]**

0 : Success  
Other: Failed

### 3.3. BeroxelHawkFrame Module Description

This module is mainly used to get the details of the frame, refer to the file "BeroxelHawkFrame.java" .

Function	Description
int getPixelType()	Get the pixformat of the image frame.
int getStreamType()	Get the stream type of the image frame.
int getFrameIndex()	Get the frame index of the image frame.
long getTimeStamp()	Get the timestamp of the image frame.
int getFPS()	Get the frame rate of the image frame.
int getWidth()	Get the width of the image frame.
int getHeight()	Get the height of the image frame.
ByteBuffer getData()	Get the frame data.
int getDataSize()	Get the size of the image frame.

### 3.4. admitenum 与 admitmode Module Description

#### 3.4.1. BeroxelHawkPixelFormatEnum Module Description

This enumeration represents the pixel format and is described as follows:

Definition	Description
BERXEL_HAWK_PIXEL_TYPE_IMAGE_RGB24	The pixel format of the color frame. 3 bytes per pixel. (RGB888)
BERXEL_HAWK_PIXEL_TYPE_DEPTH_16BIT_12I_4D	The pixel format of the depth frame. 2 bytes per pixel. The first 12 bits represent integer bits. The last 4 digits represent decimal places.
BERXEL_HAWK_PIXEL_TYPE_DEPTH_16BIT_13I_3D	The pixel format of the depth frame. 2 bytes per pixel. The first 13 bits represent integer bits. The last 3 digits represent decimal

	places.
BERXEL_HAWK_PIXEL_TYPE_IR_16BIT	The pixel format of the infrared frame. 2 bytes per pixel
BERXEL_HAWK_PIXEL_INVALID_TYPE	Unsupported pixel format

### 3.4.2. BixelHawkStreamTypeEnum Enumeration Description

This enumeration represents the stream type, as follows:

Definition	Description
BERXEL_HAWK_COLOR_STREAM	The stream type of the color frame.
BERXEL_HAWK_DEPTH_STREAM	The stream type of the depth frame.
BERXEL_HAWK_IR_STREAM	The stream type of the ir frame.
BERXEL_HAWK_LIGHT_IR_STREAM	The stream type of the light ir frame.
BERXEL_HAWK_INVALID_STREAM	Unsupported stream types.

### 3.4.3. BixelHawkStreamFlagEnum Enumeration Description

This enumeration represents a stream mode type, in SINGULAR mode, only switches for a single stream are supported. In MIX mode, a hybrid switch for multiple data streams is supported, as follows:

Definition	Description
BERXEL_HAWK_SINGULAR_STREAM_FLAG_MODE	The data stream is in single mode, and only one type of data stream can be opened.
BERXEL_HAWK_MIX_STREAM_FLAG_MODE	The data stream is a mixed-stream VGA mode, which supports opening multiple data streams at the same time
BERXEL_HAWK_MIX_HD_STREAM_FLAG_MODE	The data stream is in mixed streaming HD mode, which supports opening multiple data streams at the same time
BERXEL_HAWK_MIX_QVGA_STREAM_FLAG_MODE	The data stream is a mixed-stream QVGA mode, which supports opening multiple data streams at the same time

The horizontal device supports the following resolutions:

Stream Mode	Color Resolution	Depth Resolution	IR Resolution
BERXEL_HAWK_SINGULAR_STREAM_FLAG_MODE	1920*1080@30fps 640*400@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	Not supported
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	Not supported

### 3.4.4. BixelHawkDeviceInfo Class Description

This structure represents the device information, as follows:

Definition	Description
int getVendorId()	Vendor id
int getProductId()	Product id
String getSerialNumber()	Serial Number



### 3.4.5. BeroxelHawkStreamFrameMode Class Description

This struct represents the frame mode and is required in setting and get the FrameMode, as follows:

描述	说明
BeroxelHawkPixelFormatEnum getPixelFormat()	Pixel format, refer to the section 3.4.1
int getResolutionX()	Width
int getResolutionY()	Height
int getmFps()	Fps
void setPixelFormat(BeroxelHawkPixelFormatEnum pixelType)	Set pixel format, refer to the section 3.4.1
void setResolutionX(int resolutionX)	Set Width.
void setResolutionY(int resolutionY)	Set Height
void setFps(int fps)	Set Fps

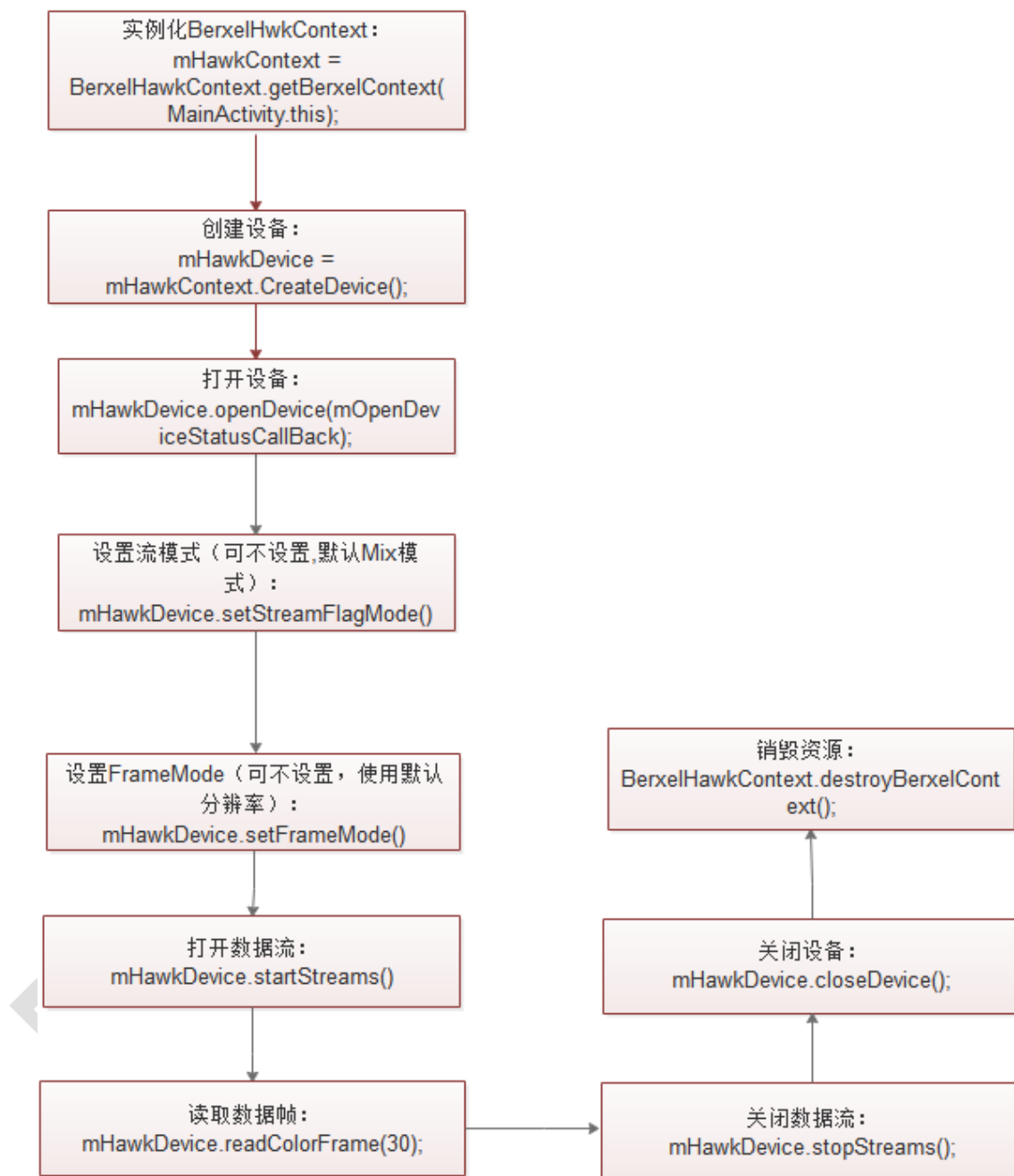
### 3.4.6. BeroxelHawkCameraIntrinsic 类说明

This structure represents the camera parameters, which are used when converting point cloud data, as follows:

描述	说明
float getFxParam()	X-direction focal length
float getFyParam()	Y-direction focal length
float getCxParam()	X-direction primary optical pivot point position
float getCyParam()	Y-direction primary optical pivot point position
float getK1Param()	Radial distortion K1
float getK2Param()	Radial distortion K2
float getP1Param()	Tangential distortion p1
float getP2Param()	Tangential distortion p2
float getK3Param()	Radial distortion K3

## 4. SDK Guidelines

### 4.1. Call SDK API Flowchart



### 4.2. Get Color Frame

Refer to HawkColor example program in SDK.

### 4.3. Get Depth Frame

Refer to HawkDepth example program in SDK.

### 4.4. Get Color and Depth Mix Frame

Refer to HawkMixColorDepth example program in SDK.

### 4.5. Get Version

The version number structure is defined in the BernelHawkVersion.java, including SDK version number, firmware version number, and hardware version number. The sample code is as follows:

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

### 4.6. Get Current Device Info

The device info structure is defined in the BernelHawkDeviceInfo.java, including SN, vendorId, productId, deviceNum, serialNumber, and deviceAddress. The sample code is as follows:

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

### 4.7. Depth Raw Data Converted To The Distance Value

The depth raw data has two data formats, BERNEL\_HAWK\_PIXEL\_TYPE\_DEP\_16BIT\_12I\_4D and BERNEL\_HAWK\_PIXEL\_TYPE\_DEP\_16BIT\_13I\_3D, The two formats are distinguished according to the getPixelFormat interface in the BernelHawkFrame.

#### 4.7.1. BERNEL\_HAWK\_PIXEL\_TYPE\_DEP\_16BIT\_12I\_4D

The depth raw data has 16 bits. The upper 12 bits are integer parts and the lower 4 bits are decimal parts, The conversion steps are as follows:

- ① Get raw data: uint16\_t depthOri = pDepth[i]; (pDepth is depth data pointer)

- ② Get integer part: float depthFront = depthOri >> 4;
- ③ Get decimal part: float depthTail = (depthOri & 0x000f)/16;
- ④ Get depth value: float depth = depthFront + depthTail;

#### 4.7.2. BERNEL\_HAWK\_PIXEL\_TYPE\_DEP\_16BIT\_12I\_4D

The depth raw data has 16 bits. The upper 13 bits are integer parts and the lower 3 bits are decimal parts, The conversion steps are as follows:

- ① Get raw data: uint16\_t depthOri = pDepth[i]; (pDepth is depth data pointer)
  - ② Get integer part: float depthFront = depthOri >> 3;
  - ③ Get decimal part: float depthTail = (depthOri & 0x0007)/8;
- Get depth value: float depth = depthFront + depthTail;

## 4.8. Set Device Status Monitoring

The onDeviceStatusChange function can be used to monitor the disconnection and connection status of the device, The sample code is as follows:

```
private class MainDeviceStatusChangedCallBack implements BernelHawkContext.DeviceStatusChangedCallBack {
    @Override
    public void onDeviceStatusChanged(int vid, int pid, BernelHawkDeviceStatusEnum status) {
        if(status == BernelHawkDeviceStatusEnum.BERNEL_HAWK_DEVICE_STATUS_DISCONNECT) {
        }
        else {
        }
    }
}

mDeviceStatusChangeCallBack = new MainDeviceStatusChangedCallBack();
mHawkContext.addDeviceStatusCallBack(mDeviceStatusChangeCallBack);
```