

Berxel 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	2021.07.28	Allen
V1.1		1 增加获取相机内参接口说明 2 增加设置图像镜像接口说明 2 增加获取点云数据的说明 3 增加深度帧转换深度值的说明	2021.08.02	Allen
V1.2		1 修改深度转点云接口	2021.08.27	Allen
V1.3		1 增加设置配准开关接口 2 增加设置流模式开关接口	2021.09.02	Allen
V1.4		1 增加设置系统时钟接口	2021.09.14	Allen
V2.0.16		1 增加 C#SDK 相关说明	2021.10.18	Allen
V2.0.18		1 修改 closeDevice 接口参数 2 修改 startUpgrade 接口参数	2021.10.19	Allen

目录

1. 概述.....	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 2 1 例子程序说明.....	7
2 2 2 例子程序 UI 显示说明.....	7
2.3. SDK 调用说明.....	7
2 3 1 Windows SDK API 调用步骤.....	7
2 3 2 Linux SDK API 调用步骤.....	7
3. SDK 接口说明.....	8
3.1. BernelHawkContext 模块说明.....	8
3.1.1. getBernelContext.....	8
3.1.2. destroyBernelContext.....	8
3.1.3. getDeviceList.....	9
3.1.4. openDevice.....	9
3.1.5. closeDevice.....	10
3.1.6. setDeviceStateCallback.....	10
3.2. BernelHawkDevice 模块说明.....	11
3.2.1. startStreams.....	11
3.2.2. stopStreams.....	12

3.2.3. getSupportFrameModes	13
3.2.4. setFrameMode	13
3.2.5. getCurrentFrameMode.....	14
3.2.6. readColorFrame	15
3.2.7. readDepthFrame.....	15
3.2.8. readIrrFrame.....	16
3.2.9. readLightIrrFrame	17
3.2.10. releaseFrame	17
3.2.11. startUpgrade.....	18
3.2.12. convertDepthToPointCloud.....	18
3.2.13. getVersion	19
3.2.14. getCurrentDeviceInfo.....	20
3.2.15. getCameraIntriscParams.....	20
3.2.16. getDeviceIntriscParams.....	21
3.2.17. setStreamMirror	21
3.2.18. setRegistrationEnable	22
3.2.19. setStreamFlagMode	22
3.2.20. setSystemClock	23
3.3. BerxelHawkFrame 模块说明	23
3.4. BerxelHawkDefine 模块说明	24
3.4.1. BerxelHawkPixelFormat 枚举说明.....	24
3.4.2. BerxelHawkStreamType 枚举说明.....	24
3.4.3. BerxelHawkStreamType 枚举说明.....	25
3.4.4. BerxelHawkDeviceInfo 结构体说明	25

3.4.5. BerxelHawkStreamFrameMode 结构体说明	25
3.4.6. BerxelHawkCameraIntrinsic 结构体说明	25
3.4.7. BerxelHawkDeviceIntrinsicParams 结构体说明	26
4. SDK 开发指引	27
4.1. SDK API 调用流程图	27
4.2. 获取彩色帧	28
4.3. 获取深度帧	28
4.4. 获取红外帧	28
4.5. 获取彩色+深度+红外混合帧.....	28
4.6. 获取版本号	28
4.7. 获取当前设备信息	28
4.8. 深度图数据转换为深度值.....	28
4.9. 深度图数据转换为点云数据.....	29
5. C#SDK 说明.....	30
5.1. C#SDK 整体说明	30
5.2. C#SDK 开发说明	30

1. 概述

Bixel SDK 是基于 Bixel 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。

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
Driver	设备 DFU 模式驱动，主要用于升级
Include	SDK 头文件
Lib	SDK 动态库，调用 SDK API 需包含 BixelHawk.lib 静态库
Samples	SDK 例子程序源码，使用 vs2010 编写
Thirdparty	SDK 例子程序显示所需的第三方库
Tools	Bixel SDK 演示工具

2.2. SDK Sample 说明

2.2.1. 例子程序说明

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
HawkMixDepthIr	演示获取 Mix 流的深度图+红外图流程（深度图分辨率为 400*640）
HawkMixHDColorDepth	演示获取 Mix 流的彩色图+深度图流程（深度图分辨率为 800*1280）

2.2.2. 例子程序 UI 显示说明

例子程序中使用的是 OpenGL 进行界面展示，在不同平台包含 GL 库及头文件需要注意，作如下说明：

 Windows：需要 freeglut 库，库文件与头文件位于 SDK 安装目录 Thirdparty 下面的 freeglut 文件里面

 Ubuntu：需在当前系统中手动安装 freeglut 库，执行如下指令：
`sudo apt-get install freeglut3-dev`

2.3. SDK 调用说明

2.3.1. Windows SDK API 调用步骤

-  包含 SDK 安装目录中 lib 文件夹里面的 BernelHawk.lib 静态库
-  包含 SDK 安装目录中 Include 文件夹下面的所有头文件
-  将 lib 库下面的所有文件拷贝到工程编译输出目录中

2.3.2. Linux SDK API 调用步骤

-  在 MakeFile 中链接 SDK 目录中 libs 里面的 BernelHawk.so
-  `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:XXX`，其中 xxx 表示 libs 的路径

3. SDK 接口说明

3.1. BernelHawkContext 模块说明

BernelHawkContext 模块主要维护设备列表，负责打开和关闭设备。参见 BernelHawkContext.h 头文件。

3.1.1. getBernelContext

[Description]

获取 BernelHawkContext 对象实例

[Function]

static BernelHawkContext* getBernelContext()

[Params]

无

[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(BernelHawkContext* &pBernelContext)

[Params]

pBernelContext [in]：BernelHawkContext 指针

[Return value]

无

3.1.3. getDeviceList

[Description]

获取设备列表

[Function]

```
int32_t getDeviceList(BerxelHawkDeviceInfo** pDeviceList, uint32_t* pDeviceCount)
```

[Params]

pDeviceList [out]: 设备列表

pDeviceCount [out]: 设备列表数量

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BerxelHawkDeviceInfo 为设备信息结构体，参见 BerxelHawkDefines.h 头文件

3.1.4. openDevice

[Description]

打开设备

[Function]

```
BerxelHawkDevice* openDevice(const BerxelHawkDeviceInfo deviceInfo)
```

[Params]

deviceInfo[in] : 需打开设备的设备信息结构体，由 getDeviceList 函数获取

[Return value]

BerxelHawkDevice 对象实例 ： 打开设备成功

NULL： 打开设备失败

[Remarks]

BerxelHawkDevice 为设备管理模块类， 参见 BerxelHawkDevice.h 头文件

BerxelHawkDeviceInfo 为设备信息结构体， 参见 BerxelHawkDefines.h 头文件

3.1.5. closeDevice

[Description]

关闭设备

[Function]

```
int32_t closeDevice((BerxelHawkDevice* hawkDevice)
```

[Params]

hawkDevice[in]： 当前已打开设备的设备句柄

[Return value]

0 ： 成功

非 0： 失败

3.1.6. setDeviceStateCallback

[Description]

设置设备状态回调， 以监听设备的断开与连接

[Function]

```
int32_t setDeviceStateCallback(BerxelHawkDeviceStatusChangeCallback callback, void* pUserData)
```

[Params]

Callback[in] : 设备状态回调函数

pUserData[in]: 用户数据, 在回调函数中作为入参传回

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BerxelHawkDeviceStatusChangeCallback 为设备状态回调函数, 参见

BerxelHawkDefines.h 头文件

3.2. BerxelHawkDevice 模块说明

BerxelHawkDevice 模块主要维护设备相关的操作, 包括开/关流、读取数据流、设备信息等, 参见 BerxelHawkDevice.h 头文件。

3.2.1. startStreams

[Description]

打开数据流

[Function]

int32_t startStreams(int streamFlags)

[Params]

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

[Return value]

0 : 成功

非 0：失败

[Remarks]

Bixel HawkStreamType 为流类型枚举，参见 BixelHawkDefines.h 头文件定义，共定义 4 种流类型：

BERXEL_HAWK_COLOR_STREAM :彩色流

BERXEL_HAWK_DEPTH_STREAM : 深度流

BERXEL_HAWK_IR_STREAM: 红外流

BERXEL_HAWK_LIGHT_IR_STREAM: 点红外流

3.2.2. stopStreams

[Description]

关闭数据流

[Function]

int32_t stopStreams(int streamFlags)

[Params]

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

[Return value]

0：成功

非 0：失败

[Remarks]

BixelHawkStreamType：参照 3.2.1 中 Remarks 关于 BixelHawkStreamTyp 说明

3.2.3. getSupportFrameModes

[Description]

获取数据流所支持的帧模式列表

[Function]

```
int32_t getSupportFrameModes(BixelHawkStreamType streamType,
                             const BixelHawkStreamFrameMode** pModes, uint32_t* pCount)
```

[Params]

streamType[in]: 流类型

pModes[out]: FrameMode 列表的指针

pCount [out]: FrameMode 的数量

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BixelHawkStreamType: 参照 3.2.1 中 Remarks 关于 BixelHawkStreamType说明

BixelHawkStreamFrameMode: 帧模式结构体, 参见 BixelHawkDefines.h 头文件

3.2.4. setFrameMode

[Description]

设置帧模式

[Function]

```
int32_t setFrameMode(BixelHawkStreamType streamType,
                     BixelHawkStreamFrameMode *mFrameMode)
```

[Params]

streamType[in]: 流类型

mFrameMode[in]: FrameMode指针

[Return value]

0 : 成功

非0: 失败

[Remarks]

Bixel HawkStreamType: 参照 3.2.1 中 Remarks 关于 Bixel HawkStreamType说明

Bixel HawkStreamFrameMode: 帧模式结构体, 参见 Bixel HawkDefines.h 头文件

3.2.5. getCurrentFrameMode

[Description]

获取当前的帧模式

[Function]

int32_t getCurrentFrameMode(Bixel HawkStreamType streamType,

Bixel HawkStreamFrameMode* frameMode)

[Params]

streamType[in]: 流类型

frameMode[out]: 当前 FrameMoe 的指针

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BerxelHawkStreamType: 参照 3.2.1 中 Remarks 关于 BerxelHawkStreamType说明

BerxelHawkStreamFrameMode: 帧模式结构体, 参见 BerxelHawkDefines.h 头文件

3.2.6. readColorFrame

[Description]

读取彩色数据帧

[Function]

int32_t readColorFrame(BerxelHawkFrame* &pFrame, int32_t timeout = 30)

[Params]

pFrame[out]: 数据帧的类指针

timeout[in]: 超时时间, 若超时, 则pFrame为NULL

[Return value]

0 : 成功

非0: 失败

[Remarks]

BerxelHawkFrame 为数据帧类, 参见BerxelHawkFrame.h头文件定义

3.2.7. readDepthFrame

[Description]

读取深度数据帧

[Function]

int32_t readDepthFrame(BerxelHawkFrame* &pFrame, int32_t timeout = 30)

[Params]

pFrame[out]: 数据帧的类指针

timeout[in]: 超时时间, 若超时, 则 pFrame 为 NULL

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BerxelHawkFrame 为数据帧类, 参见 BerxelHawkFrame.h 头文件定义

3.2.8. readIrFrame

[Description]

读取红外数据帧

[Function]

```
int32_t readIrFrame(BerxelHawkFrame* &pFrame, int32_t timeout = 30)
```

[Params]

pFrame[out]: 数据帧的类指针

timeout[in]: 超时时间, 若超时, 则 pFrame 为 NULL

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BerxelHawkFrame 为数据帧类, 参见 BerxelHawkFrame.h 头文件定义

3.2.9. readLightIrFrame

[Description]

获取点红外数据帧

[Function]

```
int32_t readLightIrFrame( BixelHawkFrame* &pSteamFrame, int32_t timeout = 30)
```

[Params]

pFrame[out]: 数据帧的类指针

timeout[in]: 超时时间, 若超时, 则 pFrame 为 NULL

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BixelHawkFrame 为数据帧类, 参见 BixelHawkFrame.h 头文件定义

3.2.10. releaseFrame

[Description]

释放读取的数据帧

[Function]

```
int32_t releaseFrame(BixelHawkFrame* &pHawFrame)
```

[Params]

pHawFrame[in]: 数据帧的指针

[Return value]

0 : 成功

非 0：失败

[Remarks]

Bixel HawkFrame 为数据帧类，参见 Bixel HawkFrame.h 头文件定义

3.2.11. startUpgrade

[Description]

升级固件，并设置升级状态回调

[Function]

```
int32_t startUpgrade(BixelHawkUpgradeProcessCallBack pCallbacks, void* pUserData,
const char* pFwFilePath)
```

[Params]

pCallbacks [in]: 升级状态的回调函数

pUserData[in]: 用户数据，在回调函数中作为入参传回

pFwFilePath[in]：设置固件路径

[Return value]

0：成功

非0：失败

[Remarks]

BixelHawkUpgradeProcessCallBack 为升级状态回调函数，参见BixelHawkDefines.h

头文件定义

3.2.12. convertDepthToPointCloud

[Description]

将深度数据转换为点数据

[Function]

```
int32_t convertDepthToPointCloud(BixelHawkFrame* pFrame, float factor, BixelHawkPoint3D*
pPointClouds)
```

[Params]

pFrame [in]: 数据帧

pFactor[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

pPointClouds[out]:输出点云数据

[Return value]

0 : 成功

非0: 失败

[Remarks]

Berxel HawkPoint3D 为点云数据结构体, 参见 Berxel HawkDefines.h 头文件定义

3.2.13. getVersion

[Description]

获取版本号

[Function]

int32_t getVersion(Berxel HawkVersions* versions)

[Params]

versions[out]: 版本号信息指针

[Return value]

0 : 成功

非 0: 失败

[Remarks]

Berxel HawkVersions 为版本信息结构体, 参见 Berxel HawkDefines.h 头文件定义

3.2.14. `getCurrentDeviceInfo`

[Description]

获取当前打开设备的设备信息

[Function]

```
int32_t getCurrentDeviceInfo(BixelHawkDeviceInfo* pDeviceInfo)
```

[Params]

pDeviceInfo [out]: 当前设备信息的指针

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BixelHawkDeviceInfo 为设备信息结构体，参见 BixelHawkDefines.h 头文件定义

3.2.15. `getCameraIntriscParams`

[Description]

获取相机内参，此相机内参可用于点云数据转换

[Function]

```
int32_t getCameraIntriscParams(BixelHawkCameraIntrinsic *pParams)
```

[Params]

pParams [out]: 相机内参的指针

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BixelHawkCameraIntrinsic 为相机内参结构体，参见 BixelHawkDefines.h 头文件定义

3.2.16. getDeviceIntrinsicParams

[Description]

获取当前设备内参，包括相机内参，平移矩阵，旋转矩阵等参数

[Function]

```
int32_t getDeviceIntrinsicParams(BixelHawkDeviceIntrinsicParams *pParams )
```

[Params]

pDeviceInfo[out]: 设备内参的指针

[Return value]

0 : 成功

非 0: 失败

[Remarks]

BixelHawkDeviceIntrinsicParams 为设备内参结构体，参见 BixelHawkDefines.h 头文件定义

3.2.17. setStreamMirror

[Description]

设置图像镜像，默认所有数据帧非镜像状态

[Function]

```
int32_t setStreamMirror(bool bNeedMirror)
```

[Params]

bNeedMirror[in]: true 表示为镜像, false 表示为非镜像。

[Return value]

0 : 成功

非 0: 失败

3.2.18. 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]

int32_t setRegistrationEnable(bool 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

3.2.19. setStreamFlagMode

[Description]

设置流模式, 分为单开模式和混合流模式, 默认为混合流模式。此接口需要在开流接口之前调用。

[Function]

int32_t setStreamFlagMode(BernelHawkStreamFlagMode flagMode)

[Params]

flagMode[in]: 流模式状态。

[Return value]

0 : 成功

非 0: 失败

3.2.20. 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]

int32_t setSystemClock()

[Params]

NULL

[Return value]

0 : 成功

非 0: 失败

3.3. BixelHawkFrame 模块说明

此模块主要是获取读取帧的详细信息，参见 BixelHawkFrame.h 头文件。具体函数说明如下

函数名称	函数说明
BixelHawkPixelFormat getPixelFormat()	获取帧图像的象数格式
BixelHawkStreamType getStreamType()	获取帧图像的流类型

uint32_t getFrameNum()	获取帧图像的帧号
uint64_t getTimeStamp()	获取帧图像时间戳
uint32_t getFPS()	获取帧图像的帧率
uint32_t getWidth()	获取帧图像的 X 轴分辨率
uint32_t getHeight()	获取帧图像的 Y 轴分辨率
void* getData()	获取帧图像数据
uint32_t getDataSize()	获取帧图像的 size 大小

3.4. BixelHawkDefine 模块说明

此模块主要是定义 SDK API 所需的相关枚举以及结构体信息，参 BixelHawkDefines.h 头文件。

3.4.1. BixelHawkPixelFormat 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_FORMAT_IMAGE_RGB24	彩色图像素格式，每个像素占用 3 字节
BERXEL_HAWK_PIXEL_FORMAT_DEP_16BIT	深度图像素格式，每个像素占用 2 字节
BERXEL_HAWK_PIXEL_FORMAT_IR_16BIT	红外图像素格式，每个像素占用 2 字节
BERXEL_HAWK_PIXEL_FORMAT_MIX_16BIT	Mix 流像素格式，彩色每个像素占 3 字节，深度/红外每个像素占用 2 个字
BERXEL_HAWK_PIXEL_INVALID_FORMAT	不支持像素格式

3.4.2. BixelHawkStreamType 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	获取彩色数据的流类型
BERXEL_HAWK_DEPTH_STREAM	获取深度数据的流类型
BERXEL_HAWK_IR_STREAM	获取泛红外数据的流类型
BERXEL_HAWK_LIGHT_IR_STREAM	获取点红外数据的流类型
BERXEL_HAWK_INVALID_STREAM	不支持的流类型

3.4.3. BixelHawkStreamType 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
BERXEL_HAWK_MIX_STREAM_FLAG_MODE	数据流为混合模式，深度图分辨率为 400*640
BERXEL_HAWK_MIX_HD_STREAM_FLAG_MODE	数据流为混合模式，深度图分辨率为 800*1280

3.4.4. BixelHawkDeviceInfo 结构体说明

此结构体表示设备信息，在开关设备函数中需要传入此结构体信息。具体说明如下：

描述	数据类型	说明
vendorId	uint16_t	供应商识别码
productId	uint16_t	产品识别码
deviceNum	uint32_t	设备号
serialNumber	char[]	设备序列号，在设备打开后才会有此信息
deviceAddress	char[]	设备地址，供设备打开时使用

3.4.5. BixelHawkStreamFrameMode 结构体说明

此结构体表示帧模式，在设置和获取 FrameMode 中需要使用。具体说明如下：

描述	数据类型	说明
pixelFormat	BixelHawkPixelFormat	表示帧像素格式
resolutionX	int16_t	表示帧数据横向分辨率
resolutionY	int16_t	表示帧数据纵向分辨率
framerate	int8_t	表示帧率

3.4.6. BixelHawkCameraIntrinsic 结构体说明

此结构体表示相机内参，相机内参在转换点云数据时候使用，具体说明如下：

描述	数据类型	说明
fxParam	float	X 方向焦距
fyParam	float	Y 方向焦距
cxParam	float	X 方向主光轴点位置
cyParam	float	Y 方向主光轴点位置
k1Param	float	径向畸变 K1
k2Param	float	径向畸变 K2

p1Param	float	径向畸变 K3
p2Param	float	切向畸变 p1
k3Param	float	切向畸变 p2

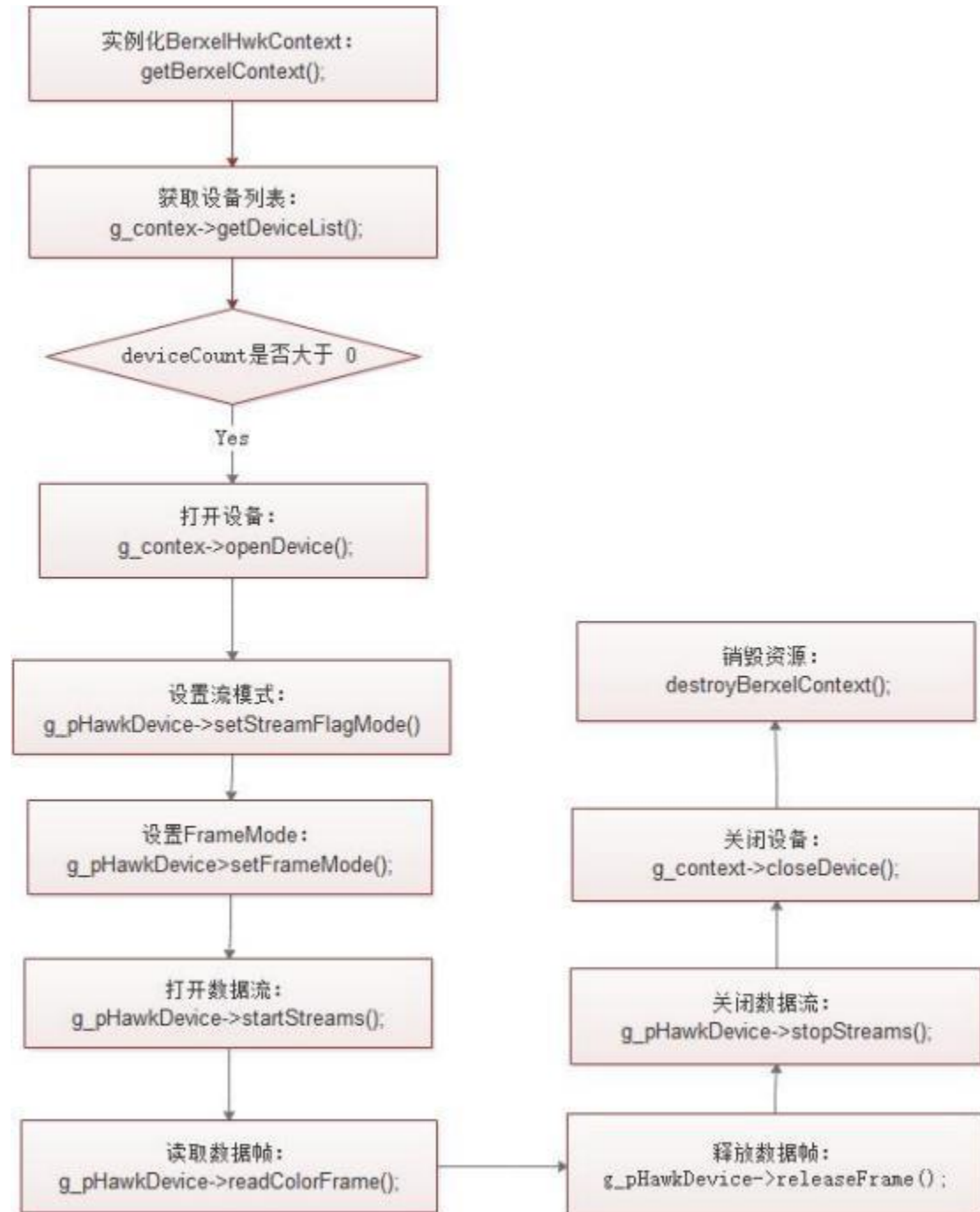
3.4.7. BixelHawkDeviceIntrinsicParams 结构体说明

此结构体表示设备内参，其中相机内参具体说明，参照 3.4.5 章节。

描述	数据类型	说明
colorIntrinsicParams	int8_t[]	彩色相机内参，9 位 float 大小
irIntrinsicParams	int8_t[]	泛红外相机内参，9 位 float 大小
liteIrIntrinsicParams	int8_t[]	点红外相机内参，9 位 float 大小
rotateIntrinsicParams	int8_t[]	旋转矩阵，9 位 float 大小
translationIntrinsicParams	int8_t[]	平移矩阵，3 位 float 大小

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+红外混合帧

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

4.6. 获取版本号

版本号结构体为 `BixelHawkVersions`，包括 SDK 版本号、固件版本号、硬件版本号。在 `BixelHawkDefines.h` 头文件中定义。示例代码如下：

```
berxel::BixelHawkVersions    tempVersions;  
g_pHawkDevice->getVersion(&tempVersions);
```

4.7. 获取当前设备信息

设备信息结构体为 `BixelHawkDeviceInfo`，包含 SN、vendorId、productId、deviceNum、serialNumber、deviceAddress 等信息，在 `BixelHawkDefines.h` 头文件中定义。示例代码如下：

```
berxel::BixelHawkDeviceInfo tempCurInfo;  
g_pHawkDevice->getCurrentDeviceInfo(&tempCurInfo);
```

4.8. 深度图数据转换为深度值

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 位为小数部分，单位是 0.0625mm。转换步骤如下：

- 👉 获取 16 位深度数据: `uint16_t depthOri = pDepth[i];` (其中 `pDepth` 代表深度数据指针)
- 👉 获取整数部分: `float depthFront = depthOri >> 4;`
- 👉 获取小数点部分: `float depthTail = (depthOri & 0x000f)/16;`
- 👉 完整深度值: `float depth = depthFront + depthTail;`

4.9. 深度图数据转换为点云数据

深度数据转换为点云数据，需要相机参数中 `focalXParam`，`focalYParam`，`opticXParam`，`opticYParam` 这 4 个参数，相机内参说明参照 3.4.5 章节，获取相机内参参照 3.2.15 章节关于 `getCameraIntriscParams` 函数的相关说明。具体步骤参照如下代码：

```
//Step1: 获取深度图
berxel::BixelHawkFrame *pHawkFrame = NULL;
g_pHawkDevice->readDepthFrame(pHawkFrame,30);

//Step2: 调用convertDepthToPointCloud 函数将深度图转换成点云图
static berxel::BixelHawkPoint3D m_3dPoints[400 *640];
g_pHawkDevice->convertDepthToPointCloud(pHawkFrame ,1000.0, m_3dPoints);

说明: convertDepthToPointCloud 函数相关说明请参照3.2.12章节
```

5. C#SDK 说明

5.1. C#SDK 整体说明

C#SDK 位于 Bernel\BernelSDK\C# 目录下，具体包括如下两部分

lib	封装的 C#SDK Driver
Samples	C#的例子演示程序，使用 VS2010 可直接编译

5.2. C#SDK 开发说明

- ① 首先需引用 Bernel\BernelSDK\C# 目录下面的 lib 文件夹里面的 BernelSdkDriver.dll。
- 2 因 BernelSdkDriver.dll 依赖 Bernel\BernelSDK\lib\ 文件夹里面的库文件，故需将此文件夹下面的所有 lib 库拷贝到工程所生成的 exe 路径里。