# CameraManager库使用说明

# 1.代码仓库

路径：[git@gitlab.slcisv.com:chuwangcai/CameraManager.git](mailto:git@gitlab.slcisv.com:chuwangcai/CameraManager.git)

分支：AllWinner-T3 （用于T3平台）

分支：Freescale-i.MX6 （用于飞思卡尔平台）

# 2.仓库功能

为方便平台移植和管理和Camera相关的外部视频输入显示，封装库整合管理代码，具体内容如下：

1.封装Camera调用相关的接口（打开摄像头，关闭摄像头，打开预览,视频信号判断等）

2.封装Camera分屏显示功能（支持后屏显示，pip视频显示）

3.封装多应用使用Camera的互斥交互逻辑(如aux,dvr,avm,dvd)

# 代码设计框架

## 1.模块介绍

软件有3部分组成，

1. CameraManagerLib.jar 动态jar包，提供接口，用于功能实现

2. CameraManagerSdk.apk 服务apk，用于和jar包沟通，管理多应用apk使用系统Camera的互斥交互逻辑

3. CameraManagerLib.jar.xml 用于指定动态jar包的位置

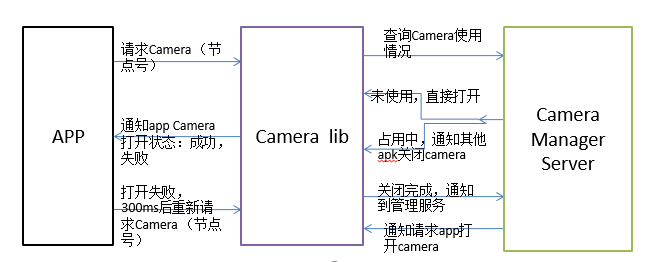
## 2.软件设计逻辑图：

Camera互斥逻辑

1：每个app在初始化CameraManager类的时候都会传递一个context，请确保此context持有生命周期为APP整个生命周期（服务或者appcontext)



1. app请求打开摄像头时传入appid(确保一个项目中每个app传递的id不同）



## 3.代码介绍

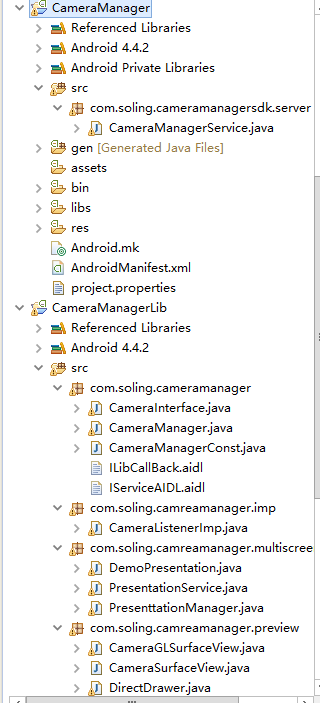
软件有3部分组成，

1. CameraManagerLib.jar 动态jar包，提供接口，用于功能实现

2. CameraManagerSdk.apk 服务apk，用于和jar包沟通，管理多应用apk使用系统Camera的互斥交互逻辑

3. CameraManagerLib.jar.xml 用于指定动态jar包的位置

具体代码：



**CameraManagerLib工程（jar）**

**com.soling.cameramanager**

封装Camera调用相关的接口及管理相关代码

1. 接口类CameraInterface.java （调用系统camera函数）
2. 管理类CameraManager.java(库的核心代码，提供接口给app调用,沟通管理服务，双屏逻辑等)
3. IServiceAIDL.aidl 用于和CameraMangerSever沟通，jar调用server接口传递状态
4. ILibCallBack.aidl 用于Sever控制jar执行命令动作
5. CameraListenerImp.java 重载CameraOpenStateListener，方便回调

**Com.soling.camreamanager.multiscreen**

封装双屏异显相关代码

1. 代理类DemoPresentation.java （后屏代理类，提供后屏显示UI）
2. 服务类PresentationService.java(后屏控制服务，拥有控制后屏代理类显示隐藏)
3. 管理类PresenttationManager.java (后屏管理控制入口，分析是否支持后屏，提供接口给cameraManger调用)

**com.soling.camreamanager.preview**

封装GLSurfaceView, SurfaceTexture 绘图方法

1. 代理类CameraGLSurfaceView.java （扩展glsufaceview类，加入渲染绘图功能）
2. DirectDrawer.java(GL绘图工具）
3. TextureUtil.java(SurfaceTexture代理类，用户获取摄像头视频数据）

**com.soling.camreamanager.util**

辅助工具类包

# 4. 2套做法介绍

1.退出关闭摄像头做法（不支持后枕屏）

特点：1.使用SurfaceView预览图像

1. 每次进入界面打开摄像头，开启预览，退出 界面关 闭摄像头，结束预览

2.退出app不关闭摄像头（支持后枕屏）

特点：1.使用GLSurfaceview ,Render渲染绘图预 览图像

2.只有其他app申请时候才重新打开关闭摄像 头

2种做法的对比：

1. 使用SurfaceView 相对CPU使用率较低，但不支持分屏显示，
2. 使用GLsurfaceView绘图可以使用SurfaceTexture,可以支持分屏显示，但是CPU会相对高。

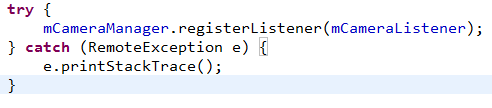
建议：1.不需要使用分屏功能的APP建议使用SurfaceView ,使用分屏功能的请使用GLsurfaceView，同一个项目可以使用不同的2种做法（兼容的），但同一个app只能使用一种方法。

# 5. APP应用使用CameraManger库的步骤及注意事项

1.初始化CameraManger类



2.注册监听



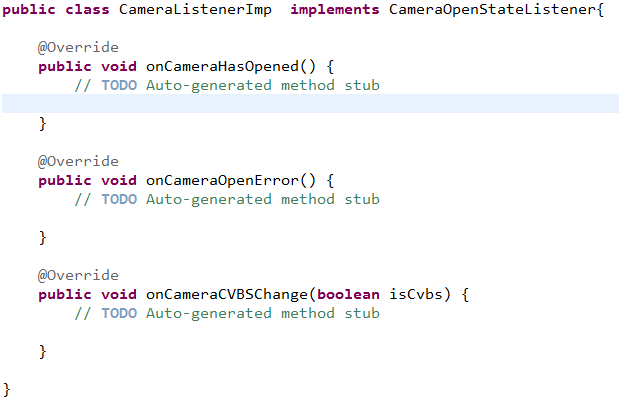
1. app请求打开摄像头时传入appid(确保一个项目中每个app传递的id不同）
2. 根据监听状态返回处理Ui刷新

onCameraHasOpened 摄像头请求后的打开成功状态，打开成功后在开启预览

onCameraOpenError 摄像头打开失败状态（打开失败后在重新请求下，正常 不会打开失败，除非有其他apk在占用）

onCameraCVBSChanger 摄像头信号状态回调，isCvbs不是说是否有信号

也可以通过getCvbsState()获取信号状态



# 6.视频信号判断

1.信号判断原理 ：打开预览后根据预览回调数据分析

注册buf监听



buf数据回调分析

# 

2.判断方式，根据分析data 数据，如果无视频信号输入时data的所有数据位均相同，有信号则不相同

3. T3平台和飞思卡尔平台的值不同，t3无视频信号输入时都是16，飞思卡尔无视频信号输入时均是52

# 7.后枕屏显示的功能实现介绍

1.驱动支持hmdi屏幕

2.app获取 屏幕的display对象，用于控制屏幕显示

3.利用后屏的display对象创建屏幕显示Presentation

4.Presentation 中可以加载需要显示的布局文件

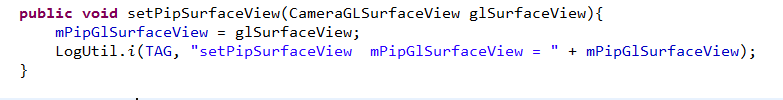
5.创建显示后屏的服务，获取Presentation 布局中的GLsurfaceView对象，并和视频的Textsurface连接起来。

核心代码见com.soling.camreamanager.multiscreen

# 8.pip功能实现

1.app创建pip window,加载GLsurfaceView布局

2.传递glsurfaceview对象给到cameramanger



1. 将视频图像渲染到glsurfaceview上



# 注意事项：

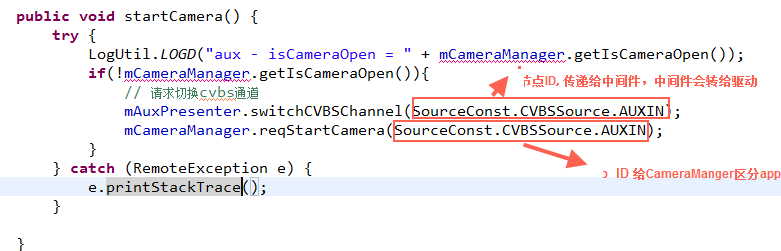
1. app ID和视频节点ID的区别

AppID : app请求打开摄像头时传入appid(确保一个项目中每个app传递的id不同）,给到CameraManger区分不同的app

视频节点ID: 给中间件转给驱动切换视频节点

对app来看2者ID相同，对中间件，会根据不同客户类型切换tvd节点和i2c节点（tvd 节点只有0-3 4路

如：IX25项目 倒车，tv,aux tvd节点均为3，是通过i2c节点来区分视频



1. 摄像头亮度调节功能由驱动实现功能，只支持tvd通道的亮度调节，不是tvd通道的不支持调节

如AVN项目 ，dvd视频通道不是走tvd视频不能调节（具体定义需要和驱动沟通确认）

1. tvd视频亮度 饱和度 对比度设置如果是走标准流程不通过i2c切换，每个视频都有单独的tvdid则记忆功能由中间件记忆，如果有i2c切换，公用的有app自行区分，即中间件只提供tvd单独设置功能记忆，不提供共用tvdid是视频亮度设置记忆

