

Android 版 SDK 常见问题解答

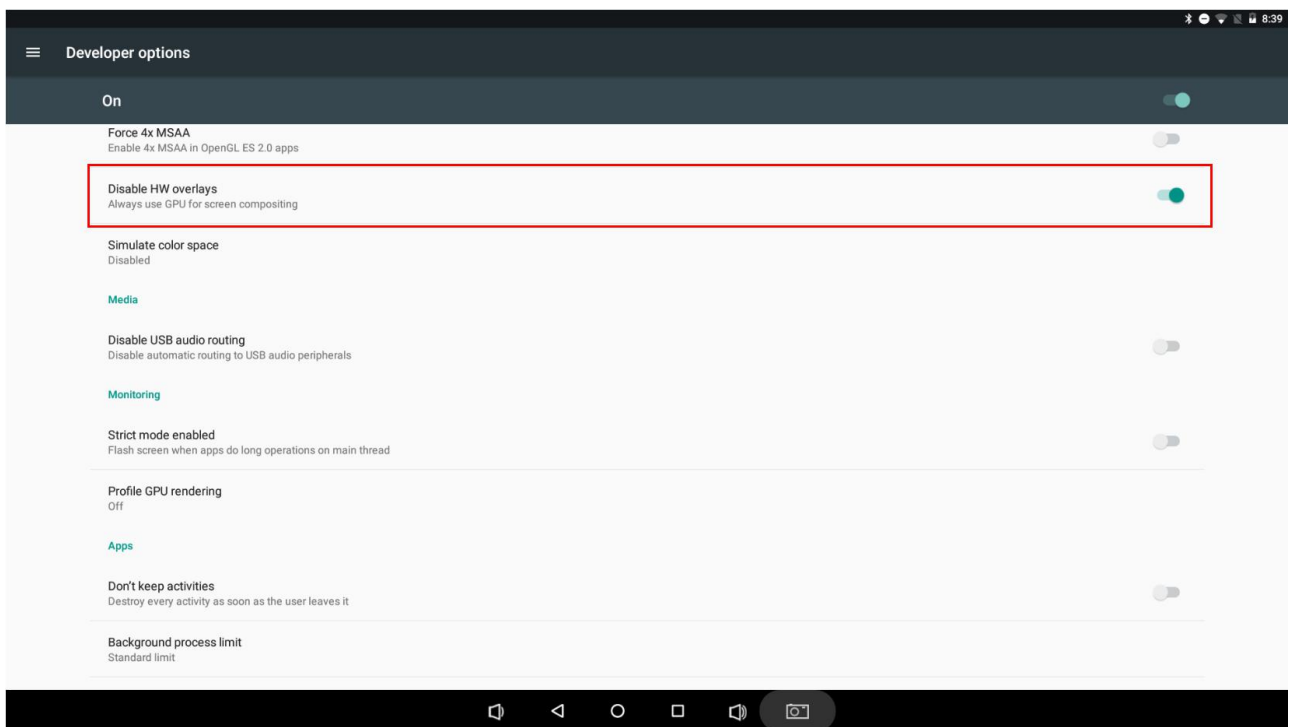
问题 1：使用 getImageBySurface 接口采图时系统重启

可能的原因：

开启了 HW 叠加层设置。

解决方法：

进入 Android 的设置选项，进入开发人员选项，找到停用 HW 叠加层，将其打开。如下图所示。



问题 2：Apk 程序安装失败

可能的原因：

Android Apk 安装和卸载有两种方式：

1) 将 Apk 文件存放在 PC 上指定目录里 ,通过执行 adb shell 命令“adb install *.apk”和“adb uninstall com.image.gxviewer”的方式安装和卸载；

2) 将 Apk 文件传输到 Android 系统的文件系统指定目录里，在 Android 系统下选择文件安装，卸载也是在 Android 系统下在设置中卸载；

这两种安装卸载方式需要配套使用，混合使用可能出现安装失败的问题。

解决方法：

使用 adb 命令安装的 Apk 程序，要使用 adb 命令卸载。

问题 3：在小内存系统中程序闪退

可能的原因：

SDK 库在开采时会申请一定数量的图像采集缓存区（默认为 5 个），用来对已采集到的但未被上层程序及时从库中取出的图像进行缓存。当使用较大分辨率的相机，申请的缓存区总量过大时，使程序占用的总内存超过系统配置的最大值时，Android 系统将会强行使程序退出。

解决方法：

1) 修改程序，减小 URB 数量和增加单个 URB 的大小。修改方法如下所示。

```
// 打开设备
device =
DeviceManager.getInstance().openDeviceByIndex(deviceInfo.getIndex());
// 检查设备是否打开成功
if(device == null) {
    return null;
}
// 设置 URB 数量，默认值为 64
device.getDataStream(0).StreamTransferNumberUrb.set(1);
// 设置单个 URB 的大小，默认值为 64 * 1024
device.getDataStream(0).StreamTransferSize.set(4 * 1024 * 1024);
```

2) 修改程序，减少申请的图像采集缓存区的个数。修改方法如下所示。

```
// 打开设备
device =
DeviceManager.getInstance().openDeviceByIndex(deviceInfo.getIndex());
// 检查设备是否打开成功
if(device == null) {
    return null;
}
// 设置缓存区数量，默认值为 5
device.getDataStream(0).setAcquisitionBufferNumber(2);
```

问题 4：相机连接 Android 设备后，相机指示灯不亮或为红灯

可能的原因：

- 1) Android 设备的 USB 接口供电能力不足，导致无法驱动相机；
- 2) Android 系统限制了 USB 接口访问，例如部分手机在关闭 OTG 功能时，无法识别相机。

解决方法：

- 3) 添加带供电功能的集线器 (Hub) 来连接 Android 设备和相机；
- 4) 检查 Android 设备是否关闭了自身的 OTG 功能，或是否对 USB 接口的访问进行了限制。

问题 5：安卓工程中引入 U3 库报错，提示无法找到库文件

错误现象：

按照 Android 开发指南添加 U3 的库文件和编写程序后，调试是提示找不到库文件，如下所示：



问题原因：

Android Studio 高版本的 Gradle 下，编译出来的应用程序无法找到库的问题，这个是 Gradle 版本升级导致的问题。Android Studio 工程配置的有个属性：android:extractNativeLibs，这个属性决定软件包安装程序是否将原生库从 APK 提取到文件系统。

在 AGP >= 3.6.0 且 minSdkVersion >= 23 这个值得默认值就变成了 false，之前的版本默认值是 true。

因此只需要将工程 app/src/main/AndroidManifest.xml 里添加这个属性值，并将其设置为 true 即可。

测试 Gradle 版本和插件版本对应关系如下：

Gradle 插件版本	Gradle 版本	测试结果
4.0.1	6.5	OK
4.2.1	6.8	OK
7.0.1	7.0.2	OK

解决方法：

创建的 Android Studio 工程，找到 app/src/main/AndroidManifest.xml，添加属性值：android:extractNativeLibs="true"，如下所示：

这个属性添加位置如下所示：

