Android 11.0.1\_mid接口

使用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 文档编号 | API-V1.0.211211 | 保密等级 | I级 |
| 修订人 | 梁斌斌 | 最后修改日期 | 2021.12.11 |
| 审核人 |  | 最后审批日期 |  |
| 批准人 |  | 最后批准日期 |  |

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订说明 | 修订人 |
| 2021.12.11 | V1.0.211211 | 初始版本 | 梁斌斌 |

目录

[第一章 Android Studio中的接口使用步骤 1](#_Toc18219)

[第二章 接口使用示例及参数详解 1](#_Toc5735)

[2.1 系统导航栏控制 1](#_Toc6236)

[（1） 导航栏隐藏 1](#_Toc11443)

[（2） 导航栏显示 1](#_Toc9594)

[（3） 获取导航栏状态 1](#_Toc19567)

[2.2 屏幕方向控制 1](#_Toc19365)

[（1） 设置屏幕方向 1](#_Toc14836)

[（2） 获取屏幕方向参数 1](#_Toc4767)

[2.3 屏幕背光控制 1](#_Toc10123)

[（1） 设置屏幕亮度 1](#_Toc12017)

[2.4 定时开关机接口 1](#_Toc7531)

[（1） 重启系统 1](#_Toc32363)

[（2） 关机 1](#_Toc26325)

[（3） 定时开关机 1](#_Toc6469)

[2.5 系统截屏接口 1](#_Toc21783)

[（1） 系统截屏 1](#_Toc9648)

[2.6 恢复出厂设置 1](#_Toc28521)

[（1） 恢复出厂设置 1](#_Toc18592)

[2.7 用户GPIO接口 1](#_Toc7947)

[（1） 设置GPIO的状态 1](#_Toc9732)

[（2） 获取GPIO口的状态 1](#_Toc8470)

[（3） 对应GPIO引脚编号 1](#_Toc13277)

[2.8 以太网参数设置 1](#_Toc5061)

[（1） 获取以太网IP地址 1](#_Toc32451)

[（2） 获取以太网mac地址 1](#_Toc25433)

[（4） 获取以太网网关 1](#_Toc13274)

[（5） 获取以太网子网掩码 1](#_Toc15036)

[（6） 获取以太网dns1 1](#_Toc9043)

[（7） 获取以太网dns2 1](#_Toc32395)

[（8） 设置以太网为静态模式 1](#_Toc18394)

[（9） 设置以太网为动态模式 1](#_Toc19768)

[2.9 WIFI参数设置 1](#_Toc684)

[（1） 获取wifi的IP地址 1](#_Toc5465)

[（2） 获取wifi的MAC地址 1](#_Toc31016)

[（3） 获取wifi的子网掩码 1](#_Toc7014)

[（4） 获取wifi 的网关 1](#_Toc10263)

[（5） 获取wifi的dns1 1](#_Toc21109)

[（6） 获取wifi的dns2 1](#_Toc8702)

[（7） 设置wifi 为静态模式 1](#_Toc18348)

[（8） 设置wifi 为动态模式 1](#_Toc22837)

[3.0 无线ADB控制 1](#_Toc7819)

[（1） 打开无线ADB调试 1](#_Toc7076)

[（2） 关闭无线ADB调试 1](#_Toc19514)

[3.1 获取设备内核信息 1](#_Toc21828)

[（1） 获取设备内核信息 1](#_Toc29303)

[第三章 常见问题汇总 1](#_Toc6108)

# 第一章 Android Studio中的接口使用步骤

1、本文使用的集成开发环境为Android Studio 3.4.1，如下图1-1所示。



图1-1 Android Studio 3.4.1

2、在项目中导入ZY-Interface.jar文件,将ZY-Interface.jar文件拷贝到项目中的libs目录下，如下图1-2所示。

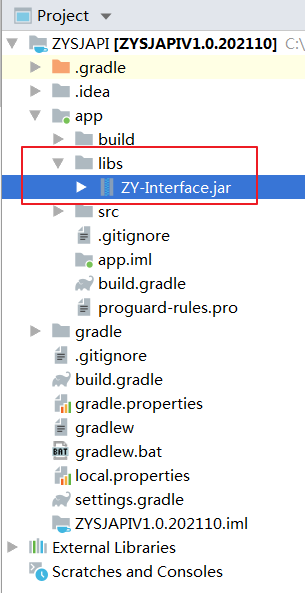


图1-2 拷贝zY-Interface.jar到libs目录

1. 右键点击libs文件夹中的ZY-Interface.jar文件，然后选择“Add As Librayr...”，到此为止已成功导入ZY-Interface.jar文件，效果如下图1-3所示。

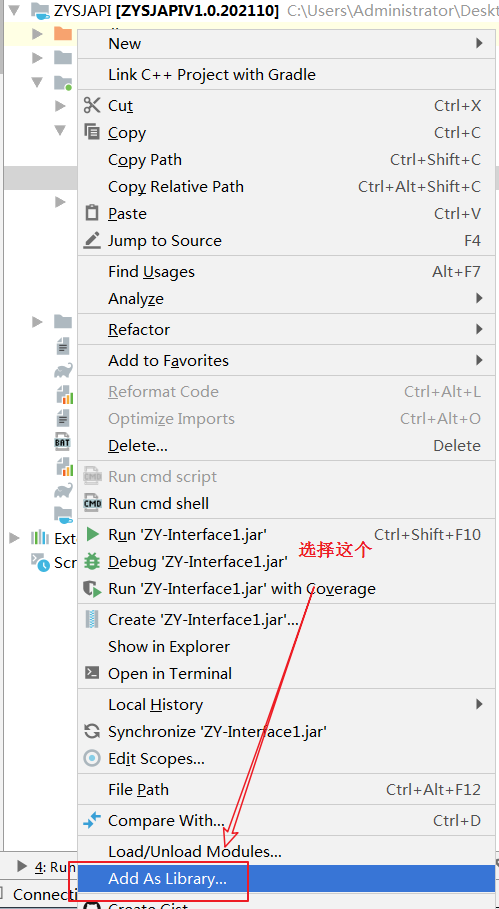


图1-3 导入zY-Interface.jar文件

注意：

此API调用前都需要初始化：

ZysjSystemManager manager=(ZysjSystemManager)getSystemService("zysj");

开始使用API

首先要声明对象，然后就可以开始使用API，如下所示。

//声明manager对象

private ZysjSystemManager manager;

manager = (ZysjSystemManager)getSystemService(“zysj”);

//使用API

manager.zYSystemBar(0);

# 第二章 接口使用示例及参数详解

## 2.1 系统导航栏控制

导航栏：即系统主界面下方的7个软件按钮和最顶端的显示时间和wifi相关图的显示栏； 通过点击界面下方的向下软件图标，可以实现些导航栏的隐藏，使用鼠标或触摸屏延最下方往上滑动可以实现导航栏的显示；其次，通过系统设置-显示-永久隐藏状态栏可以实现导航栏的隐藏并且使用滑动的方向 无法显示出来。

接口作用：可以实现导航栏的显示和隐藏，当使用此接口隐藏导航栏时，无法通过滑动的形式显示，只能通过接口或系统设置才能显示出来。

应用场景：主要使用在不允许终端客户操作系统或无法退出当前发布系统界面的项目；

1. 导航栏隐藏

表2-1-1 导航栏隐藏

|  |  |  |
| --- | --- | --- |
| int zYSystemBar(int cmd) | 参数（cmd） | 返回值 |
| 0：隐藏导航栏 | 0：设置成功  -1：设置失败 |
| 示例 | int result = manager.zYSystemBar(0); | |

1. 导航栏显示

表2-1-2 导航栏显示

|  |  |  |
| --- | --- | --- |
| int zYSystemBar(int cmd) | 参数（cmd） | 返回值 |
| 1：显示导航栏 | 0：设置成功  -1：设置失败 |
| 示例 | int result = manager.zYSystemBar(1); | |

1. 获取导航栏状态

表2-1-3 获取导航栏状态

|  |  |  |
| --- | --- | --- |
| int zYSystemBar(int cmd) | 参数（cmd） | 返回值 |
| 2：获取导航栏状态 | 0：处于隐藏状态  1：处于显示状态  -1：获取失败 |
| 示例 | int result = manager.zYSystemBar(2); | |

## 2.2 屏幕方向控制

屏幕显示方向：即系统内容显示的方向 ；此方向可以通过三种方式进行修改：1、通过系统设置-显示-屏幕方向进行修改；2、通过重力感应功能自动旋转；3、通过我们提供的接口进行修改；

接口作用：接口可以实现对屏幕的4个方向(0,90,180,270)进行修改；且可以通过此接口获取到系统当前的屏幕方向；

应用场景：主要使用在非0度显示的项目，可以屏幕方向进行修改；

1. 设置屏幕方向

表2-2-1 设置屏幕方向

|  |  |  |
| --- | --- | --- |
| int zYsetScreenDirection(int rotate); | 参数(rotate) | 返回值 |
| 0：旋转0度 | 0：设置成功  -1：设置失败 |
| 1：旋转90度 |
| 2：旋转180度 |
| 3：旋转270度 |
| 示例 | int result=manager.zYsetScreenDirection(1); | |

1. 获取屏幕方向参数

表2-2-2 获取屏幕方向

|  |  |
| --- | --- |
| int zYgetScreenDirection()； | 返回值 |
| 0 : 0度  1 : 90度  2 : 180度  3 : 270度  -1: 获取失败 |
| 示例 | int rotate = manager.zYgetScreenDirection(); |

## 2.3 屏幕背光控制

屏幕显示背光控制：即显示屏的亮度 ；此方向可以通过两种方式进行修改：1、通过系统设置-显示-亮度； 2、通过我们提供的接口进行修改；

接口作用：接口可以实现对屏幕亮度进行0~245个等级进行控制调节；

应用场景：在没人的时候可以适当将屏幕亮度降低，从而达到延长屏幕寿命的效果；

1. 设置屏幕亮度

表2-3-1 设置屏幕亮度

|  |  |  |
| --- | --- | --- |
| int zYsetBackLight(int backageLight); | 参数（cmd） | 返回值 |
| 需要设置的背光值  （0~245） | 0：设置成功  -1：设置失败 |
| 示例 | int result = manager. zYsetBackLight (144); | |

（2）获取屏幕亮度

表2-3-2 获取屏幕亮度

|  |  |
| --- | --- |
| int zYgetBackLight(); | 返回值 |
| 0~245：屏幕背光  -1：获取失败 |
| 示例 | int result = manager. zYgetBackLight (); |

## 2.4 定时开关机接口

定时开关机功能：即实际系统的定时开机和定时关机功能；通过系统设置-无障碍-定时开关机可以实现；或通过本文件介绍的接口开发第三方APK实现；

接口作用：可以设置系统定时开机或定时关机；

应用场景：主要应用在无人看守高挂广告机项目中比较多；

1. 重启系统

表2-4-1 重启系统

|  |  |  |
| --- | --- | --- |
| int zYRebootSys(); | 参数（cmd） | 返回值 |
| 无 | 0: 成功 |
| 示例 | int result = manager.zYRebootSys(); | |

1. 关机

表2-4-2 关机

|  |  |  |
| --- | --- | --- |
| int zYShutdownSys(); | 参数（cmd） | 返回值 |
| 无 | 0: 成功 |
| 示例 | int result = manager. zYShutdownSys (); | |

1. 定时开关机

表2-4-3 定时开关机

|  |  |  |  |
| --- | --- | --- | --- |
| 函数 | int setTimeToRtc(int cmd, String date, String time); | | |
| 参数(cmd) | 参数(date) | 参数(time) | 返回值 |
| 1 : 设置关机  2 : 设置开机  3 : 清除开关机  4：清除关机  5：清除开机 | 开关机时间 | 开关机日期 | 0: 设置成功  -1:参数不在1~5的范围内  -2：设置的日期为空或时间为空  -3：设置的时间小于系统当前时间  -4：关机时间与开机时间间隔小于3分钟 |
| 示例 | int result=  manager.setTimeToRtc(1,"2021-6-13","16:00"); | | |

## 2.5 系统截屏接口

截屏功能：即截取屏幕的画面；在系统设置-截屏设置 可以将截屏按钮显示出来，进行点击截屏；或使用本文讲述的接口进行截屏；

接口作用：可以使第三方APK实现截屏功能；

应用场景：主要应用在通过截屏方式了解远程设置的屏幕显示内容；

1. 系统截屏

表2-5-1 系统截屏

|  |  |  |
| --- | --- | --- |
| String zYShotScreen(String picPath, String fileName, int screenId); | | |
| 参数（cmd） | | 返回值 |
| 1、picPath：存放路径  2、fileName：截图名称  3、screenId：主副屏ID（0为主屏ID，1为副屏ID） | | “0”：截屏成功  “-1”：picPath为空  “-2”：fileName为空  “-3”：screenId小于0  “-4”：文件不存在 |
| 示例 | String result = manager.zYShotScreen("/mnt/sdcard/Screenshot/","text.png",1); | |

## 2.6 恢复出厂设置

恢复出厂设置功能：即系统的所有设置都恢复到初始状态；在系统设置-备份和重置 可以进行恢复出厂设置操作；或通过本文讲述的接口进行操作；

接口作用：可以使系统恢复到出厂设置；

应用场景：主要应用在常期使用的机器出问题的时候，可以使用此接口进行远程恢复出厂设置；

1. 恢复出厂设置

表2-6-1 恢复出厂设置

|  |  |  |
| --- | --- | --- |
| int zYresetSys(); | 参数（cmd） | 返回值 |
| 无 | 0：成功 |
| 示例 | int result = manager. zYresetSys (); | |

## 2.7 用户GPIO接口

接口作用：控制用户GPIO口；

应用场景：主要用于控制或检测外部设备的状态；

1. 设置GPIO的状态

表2-7-1 设置GPIO的状态

|  |  |  |  |
| --- | --- | --- | --- |
| int set\_zYsj\_gpio\_value(int which, int value); | 参数（which） | 参数  （value） | 返回值 |
| 表示需要控制的IO口 | 0：表示低电平  1：表示高电平 | 0：设置成功。  -1：设置失败 |
| 示例 | int result=manager.set\_zysj\_gpio\_value (0，1);//将IO口 0设为高电平 | | |

1. 获取GPIO口的状态

表2-7-2 获取GPIO口的状态

|  |  |  |
| --- | --- | --- |
| int get\_zYsj\_gpio\_value (int which); | 参数（which） | 返回值 |
| 表示需要获取状态的IO口 | 0：获取到低电平。  1：获取到高电平  -1：获取失败 |
| 示例 | int result=manager.get\_zysj\_gpio\_value (0);//获取IO 口0的状态 | |

1. 对应GPIO引脚编号

GPIO01 ~ GPIO06 对应的编号为1~6；

其中补光灯的编号为：24；

调用详解：

设置GPIO的值：

Power.set\_zysj\_gpio\_value(x,y); (其中：x为GPIO号,对应板卡上的编号,取值范围为：1，2，3，4，5，6；y为设置GPIO的值，取值为：0为底电平，1为高电平)

获取GPIO的值：

Power.get\_zysj\_gpio\_value(x); (其中：x为GPIO号,对应板卡上的编号,取值范围为：1，2，3，4，5，6；返回值：0表示底电平，1表示高电平)

注意：

由于各款板卡目前还没有统一GPIO编号，如果遇到没有标准编号的板卡，都按从GND开始为：GND,04,03,02,01,VCC 这样的顺序与接口对应，且仅小板提供6个GPIO。

## 2.8 以太网参数设置

设置以太网参数接口：即可以通过此接口可以设置机器的以态网静态IP，网关等参数 ；以态网相关参数进行修改有两种方式：1、通过系统设置-更多-以太网 以太网模式-静态中进行设置； 2、通过我们提供的接口进行修改；

接口作用：接口可以实现通过第三方APK设置以太网的静态IP,网关，DNS等参数；

应用场景：远程修改以态网相关参数；

1. 获取以太网IP地址

表2-8-1 获取以太网IP地址

|  |  |  |
| --- | --- | --- |
| String zYgetEthIp(); | 参数（cmd） | 返回值 |
| 无 | 以太网的 ip地址 |
| 示例 | String result = manager.zYgetEthIp(); | |

1. 获取以太网mac地址

表2-8-2 获取以太网mac地址

|  |  |  |
| --- | --- | --- |
| String zYgetEthMacAddress(); | 参数（cmd） | 返回值 |
| 无 | 以太网的 MAC地址 |
| 示例 | String result =  manager.zYgetEthMacAddress(); | |

1. 获取以太网网关

表2-8-3 获取以太网网关

|  |  |  |
| --- | --- | --- |
| String zYgetEthGatWay(); | 参数（cmd） | 返回值 |
| 无 | 以太网的网关 |
| 示例 | String result = manager.zYgetEthGatWay(); | |

1. 获取以太网子网掩码

表2-8-4 获取以太网子网掩码

|  |  |  |
| --- | --- | --- |
| String zYgetEthNetMask(); | 参数（cmd） | 返回值 |
| 无 | 以太网的子网掩码 |
| 示例 | String result = manager.zYgetEthNetMask(); | |

1. 获取以太网dns1

表2-8-5 获取以太网dns1

|  |  |  |
| --- | --- | --- |
| String zYgetEthDns1(); | 参数（cmd） | 返回值 |
| 无 | 以太网的 dns1 |
| 示例 | String result = manager.zYgetEthDns1(); | |

1. 获取以太网dns2

表2-8-6 获取以太网dns2

|  |  |  |
| --- | --- | --- |
| String zYgetEthDns2(); | 参数（cmd） | 返回值 |
| 无 | 以太网的 dns2 |
| 示例 | String result = manager.zYgetEthDns2(); | |

1. 设置以太网为静态模式

表2-8-7 设置以太网为静态模式

|  |  |  |  |
| --- | --- | --- | --- |
| void zYsetEthernetParams  (int mode,Map params); | 参数（mode） | 参数（params） | 返回值 |
| 以太网模式  0：动态模式  1：静态模式 | 由ip、gateway、netmask、dns1、dns2组成的集合  ip: ip地址  gatway：网关  netmask：子网掩码dns1：dns1  dns2：dns2  注：集合中的参数不设置时可以传null | 无 |
| 示例 | Map<String,String> params = new HashMap<>(); params.put(zYsjSystemManager.STATIC\_IP,"192.168.0.158"); params.put(zYsjSystemManager.STATIC\_NETMASK,"192.168.0.1"); params.put(zYsjSystemManager.STATIC\_GATEWAY,"255.255.255.0"); params.put(zYsjSystemManager.STATIC\_DNS1,null); params.put(zYsjSystemManager.STATIC\_DNS2,null);  manager.zYsetEthStaticMode (1, params ); | |  |

1. 设置以太网为动态模式

表2-8-8 设置以太网为动态模式

|  |  |  |  |
| --- | --- | --- | --- |
| void zYsetEthernetParams  (int mode,Map params); | 参数(mode) | 参数(params) | 返回值 |
| 以太网模式  0：动态模式  1：静态模式 | 由ip、gateway、netmask、dns1、dns2组成的集合  ip: ip地址  gatway：网关  netmask：子网掩码dns1：dns1  dns2：dns2  注：集合中的参数不设置时可以传null | 无 |
| 示例 | Manager.zYsetEthernetParams(0,null); | |  |

## 2.9 WIFI参数设置

WIFI参数接口：即可以通过此接口可以设置机器WIFI的IP，网关等参数 ；WIFI相关参数进行修改有两种方式：1、通过系统设置-WLAN，连接时下面出现的高级选项中进行调； 2、通过我们提供的接口进行修改；

接口作用：接口可以实现通过第三方APK设置wifi的IP,网关，DNS等参数；

应用场景：远程修改wifi相关参数；

1. 获取wifi的IP地址

表2-9-1 获取wifi的IP地址

|  |  |  |
| --- | --- | --- |
| String zYgetWifiIp(); | 参数 | 返回值 |
| 无 | Wifi的IP地址 |
| 示例 | String result = manager.zYgetWifiIp (); | |

1. 获取wifi的MAC地址

表2-9-2 获取wifi的MAC地址

|  |  |  |
| --- | --- | --- |
| String zYgetWifiMacAddress(); | 参数 | 返回值 |
| 无 | Wifi的MAC地址 |
| 示例 | String result = manager.zYgetWifiMacAddress(); | |

1. 获取wifi的子网掩码

表2-9-3 获取wifi的子网掩码

|  |  |  |
| --- | --- | --- |
| String zYgetWifiNetMask(); | 参数（cmd） | 返回值 |
| 无 | Wifi的 子网掩码 |
| 示例 | String result = manager.zYgetWifiNetMask(); | |

1. 获取wifi 的网关

表2-9-4 获取wifi的网关

|  |  |  |
| --- | --- | --- |
| String zYgetWifiGatWay(); | 参数（cmd） | 返回值 |
| 无 | Wifi的 网关 |
| 示例 | String result = manager.zYgetWifiGatWay(); | |

1. 获取wifi的dns1

表2-9-5 获取wifi的dns1

|  |  |  |
| --- | --- | --- |
| String zYgetWifiDns1(); | 参数（cmd） | 返回值 |
| 无 | Wifi的 dns1 |
| 示例 | String result = manager.zYgetWifiDns1(); | |

1. 获取wifi的dns2

表2-9-6 获取wifi的dns2

|  |  |  |
| --- | --- | --- |
| String zYgetWifiDns2(); | 参数（cmd） | 返回值 |
| 无 | Wifi的 dns2 |
| 示例 | String result = manager.zYgetWifiDns2(); | |

1. 设置wifi 为静态模式

表2-9-7 设置wifi 为静态模式

|  |  |  |  |
| --- | --- | --- | --- |
| void zYsetWifiParams(int mode,Map params) | 参数（mode） | 参数（params） | 返回值 |
| WIFI模式  0：动态模式  1：静态模式 | 由ip、gateway、netmask、dns1、dns2组成的集合  ip: ip地址  gatway：网关  netmask：子网掩码  dns1：dns1  dns2：dns2  注：集合中的参数不设置时可以传null | 无 |
| 示例 | Map<String,String> params = new HashMap<>(); params.put(zYsjSystemManager.STATIC\_IP,"192.168.0.158"); params.put(zYsjSystemManager.STATIC\_NETMASK,"192.168.0.1"); params.put(zYsjSystemManager.STATIC\_GATEWAY,"255.255.255.0"); params.put(zYsjSystemManager.STATIC\_DNS1,"8.8.8.8"); params.put(zYsjSystemManager.STATIC\_DNS2,"8.8.4.4");  manager.zYsetWifiParams(1,params); | | |

1. 设置wifi 为动态模式

表2-9-8 设置wifi 为动态模式

|  |  |  |  |
| --- | --- | --- | --- |
| void zYsetWifiParams(int mode,Map params) | 参数（mode） | 参数（params） | 返回值 |
| WIFI模式  0：动态模式  1：静态模式 | 由ip、gateway、netmask、dns1、dns2组成的集合  ip: ip地址  gatway：网关  netmask：子网掩码  dns1：dns1  dns2：dns2  注：集合中的参数不设置时可以传null | 无 |
| 示例 | manager.zYsetWifiParams(0,null); | | |

## 3.0 无线ADB控制

无线ADB调试：即设备与电脑不须要通过数据线连接进行软件调试；打开和关闭的方法有两种：1、通过系统设置-无障碍-无线ADB调试 选项进行打开或关闭；2、通过本文描述的无线ADB调试接口；

接口作用：通过此接口可以实现无线ADB功能的打开或关闭。

应用场景：主要应用在设备与电脑使用数据线连接不方便的情况下使用；

1. 打开无线ADB调试

表3-0-1 打开无线ADB调试

|  |  |  |
| --- | --- | --- |
| void zYsetAdbWiress(boolean enable) | 参数（cmd） | 返回值 |
| false：关闭  true：打开 | 无 |
| 示例 | mzYsjSystemManager.zYsetAdbWiress (true); | |

1. 关闭无线ADB调试

表3-0-2 关闭无线ADB调试

|  |  |  |
| --- | --- | --- |
| void zYsetAdbWiress(boolean enable) | 参数（enable） | 返回值 |
| false：关闭  true：打开 | 无 |
| 示例 | manager.zYsetAdbWiress (false); | |

## 3.1 获取设备内核信息

设备内核信息参数接口：即可以通过此接口获取到设备内核的相关信息 ；接口作用：接口可以实现通过第三方APK获取设备内核的相关信息；

应用场景：获取设备内核的相关信息；

1. 获取设备内核信息

表3-1-1 获取设备内核信息

|  |  |  |
| --- | --- | --- |
| int zYgetDeviceKernelInfo（） | 参数（cmd） | 返回值 |
| 无 | 设备内核相关信息 |
| 示例 | String DeviceKernelInfo = manager.zYgetDeviceKernelInfo(); | |

# 第三章 常见问题汇总

FAQ1: 暂时没发现问题点。