

TW-AC3399-Android 使用手册

修订历史

版本	日期	修改人	原因
V1.00	2021/6/16		创建文档

目 录

1 编译环境搭建.....	1
1.1 硬件要求.....	1
1.2 软件配置.....	1
1.3 源码获取.....	1
1.4 编译.....	1
2 镜像升级.....	3
2.1 OTG 烧写.....	3
2.1.1 升级流程分析.....	3
2.1.2 升级操作步骤.....	4
2.1.3 切换到升级模式.....	4
2.1.4 擦除 FLASH 操作.....	5
2.1.5 整包固件升级.....	6
2.2 TF 卡烧写.....	6
2.2.1 TF 卡制作与烧写.....	6
3 Android 功能使用及测试.....	8
3.1 主界面展示.....	8
3.2 应用程序.....	8
3.3 设置语言.....	9
3.4 查看图片和视频.....	10
3.5 播放音乐.....	12
3.6 录音（支持 Mic 输入）.....	15
3.7 调节音量.....	17
3.8 背光控制.....	18
3.9 设置时间(RTC).....	19
3.10 以太网测试.....	21
3.11 WiFi 测试.....	24
3.12 蓝牙测试.....	26
3.13 TF 卡及 USB 存储测试.....	28
3.14 USB 鼠标测试.....	29
3.15 串口测试.....	29
3.16 4G 拨号上网测试.....	31
3.17 USB 摄像头测试.....	33
4 系统定制.....	36
4.1 界面去掉顶部 Google 搜索栏.....	36
4.2 系统预装 APP 的目录.....	37
4.3 系统永不休眠设置.....	39
4.4 开机 logo 动画修改.....	41
4.4.1 第一阶段.....	41
4.4.2 第二阶段.....	41
4.5 默认壁纸修改.....	41
4.6 隐藏屏蔽通知栏、状态栏.....	41

4.7 修改导航栏.....	42
4.8 默认语言的修改.....	43
4.9 默认时区修改.....	43
4.10 打开或关闭触摸声音.....	44
4.11 去除 USB 权限弹窗.....	44
4.11.1 指定包名.....	44
4.11.2 允许所有 APP 无弹窗.....	45
5 ADB 的使用.....	46
5.1 USB 的方式.....	46
5.2 网络的方式.....	46
5.3 常用命令.....	46
6 注意事项与建议.....	47
6 免责声明.....	48

1 编译环境搭建

注意: 如无特别说明所有操作均在普通用户, 及用户所有目录下执行, 请勿使用 `sudo` 或者 `root`

1.1 硬件要求

编译环境有硬件要求, 如下

64bit x86 cpu

物理内存+ 交换内存大于等于 8GB

大于等于 200GB 的硬件空间, android 8.1 编译完成后需要占 143G 左右空间

1.2 软件配置

本文档默认使用 64bit ubuntu 14.04, 其他 linux 版本请自行查阅相关资料安装配置。

安装 openJDK1.8

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install openjdk-8-jdk
```

安装编译 android 其他依赖软件包

```
sudo apt-get install uuid-dev liblzo2-dev realpath git-core gnupg flex
bison gperf libssl1.2-dev libesd0-dev libwxgtk2.8-dev squashfs-tools
build-essential zip curl libncurses5-dev zlib1g-dev pngcrush schedtool
libxml2 libxml2-utils xsltproc lzop libc6-dev schedtool g++-multilib
lib32z1-dev lib32ncurses5-dev lib32readline-gplv2-dev gcc-multilib
libswitch-perl libssl1.0.0 libssl-dev sharutils gawk netpbm
```

1.3 源码获取

源码发布在网盘, 百度网盘发布的包包括源码、文档、镜像、工具等, 网盘上的源码包是基础包, 后续将以 `git` 补丁形式持续更新。网盘上的源码是整个 android 项目的 `.git tar` 包, 保存在网盘根目录下的 `source` 目录中:

```
TW-AC3399-Android-SDK.tar
```

下载完成后, 拷贝 `king_rp_rk3399_DDR3_8.1_SDK.tar` 到 linux 用户主目录下
`cd ~ #` 进入用户主目录

```
tar xvf TW-AC3399-Android-SDK.tar# 解压完成, 生成相应目录
```

1.4 编译

源码根目录下提供编译脚本 `make.sh`, 在源码根目录下执行即可。

```
./make.sh clean u-boot # 清除 u-boot 编译文件
./make.sh clean kernel # 清除 kernel 编译文件
./make.sh clean Android # 清除 Android 编译文件
./make.sh clean # 清除 u-boot kernel Android 编译文件
```

```
./make.sh u-boot # 编译 u-boot
./make.sh kernel # 编译 kernel
./make.sh Android # 编译 Android
./make.sh # 编译 u-boot kernel Android 并打包量产固件，量产固
件命名如 update_20180110_1828.img (20180110_1828 为年月日_时分)
./make.sh ota # 打包固件为 OTA 格式固件
```

单独编译 Android 上层模块需要完整编译一次之后先执行 `source build/envsetup.sh && lunch rk3399_mid-userdebug` 再单独编译上层模块

`./make.sh` 编译完成后生成文件在：Android 顶层目录/rockdev/Image-rk3399_mid/

如果只编译 u-boot 和 kernel，生成的固件直接对应应在 u-boot 和 kernel 下面

备注 1：打包固件为 OTA 格式后可用于将固件放在 SD 卡上面重启升级

备注 2：OTA 格式，普通格式为两种不同固件格式，(友情提示：运行过程中，普通固件在 kernel 阶段使用 kernel.img OTA 格式在 kernel 阶段使用 boot.img 内的 kernel，即打包过程拷贝的 zImage)

2 镜像升级

2.1 OTG 烧写

升级工具：AndroidTool_Release_v2.55 或 2.61

升级工具路径：cdrom_rk3399\cdrom-rk3399-Linux\Tools-usb-drivers\

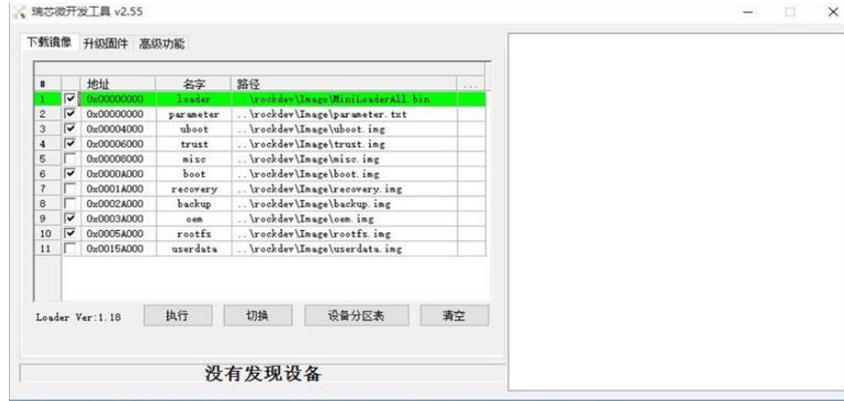


图 2.1 工具界面

2.1.1 升级流程分析

升级流程分析，说明图如下所示

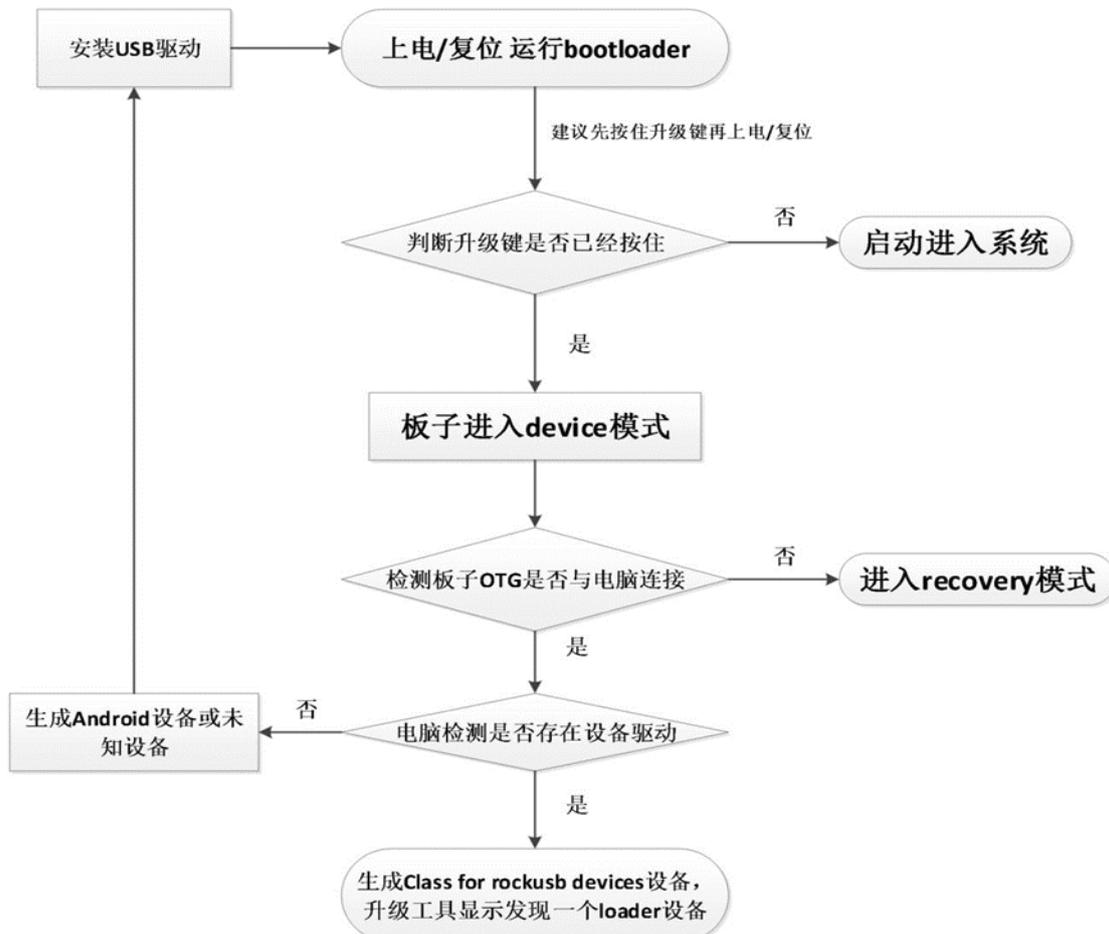


图 2.2 升级流程

当板子从 bootloader 开始启动；
程序会检测升级按键是否给按住；
如果给按住了，会进入 device 模式，并与电脑进行通信；
如果跟电脑没有连接 OTG 线，进入 recovery 模式；
如果跟电脑有连接 OTG 线，进入 device 模式；
当进入 device 模式后；
如果没有安装驱动，电脑设备管理器生成一个设备；
如果已经安装驱动，电脑设备管理器生成 RK 专有的设备驱动名称（Class for rockusb devices），升级工具显示为一个 loader 模式。

2.1.2 升级操作步骤

- 1、板子断开电源
- 2、OTG 口连接电脑
- 3、按住升级按键不松开
- 4、插上电源
- 5、等待 3 秒让板子升级模式
- 6、松开升级按键
- 7、升级工具里点击执行进行升级

2.1.3 切换到升级模式

切换模式前需要保证通过 usb 数据线使 rk3399 的 otg 口与电脑保持连接。这是电脑识别到 rk3399 的前提。使用提供的 usb 数据线，数据线另一个口连接电脑，具体接口位置查看下图标注。

rk3399 切换升级模式有两种方式：

1. 长按升级键不放，插上 12V 电源
2. 如果已经上电，长按升级键不放，按一下复位键使用 rk3399 重启；

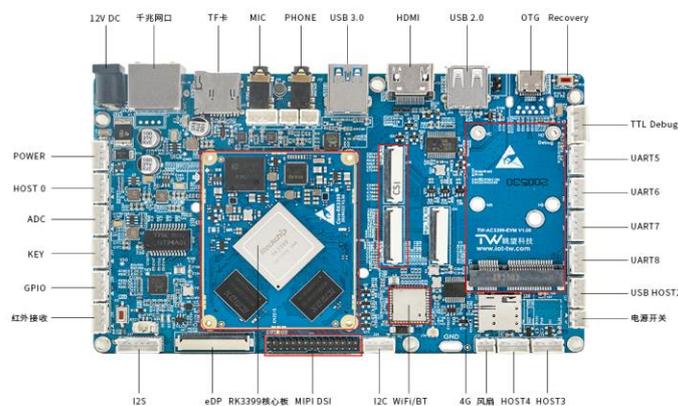


图 2.3 TW-AC3399

只要确定 rk3399 换到升级模式，就可以松开升级按键了。驱动安装成功后，只要 rk3399 切换到升级模式，可以从设备管理器看到 Class for rockusb devices。

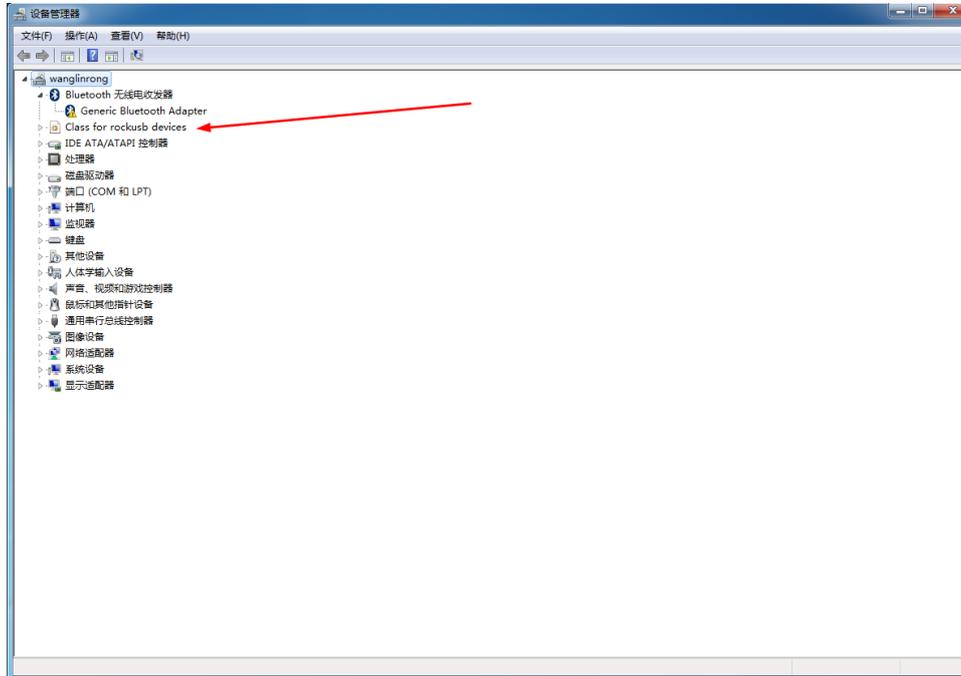


图 2.4 设备管理器

从 AndroidTool 工具也可以看到 LOADER 设备

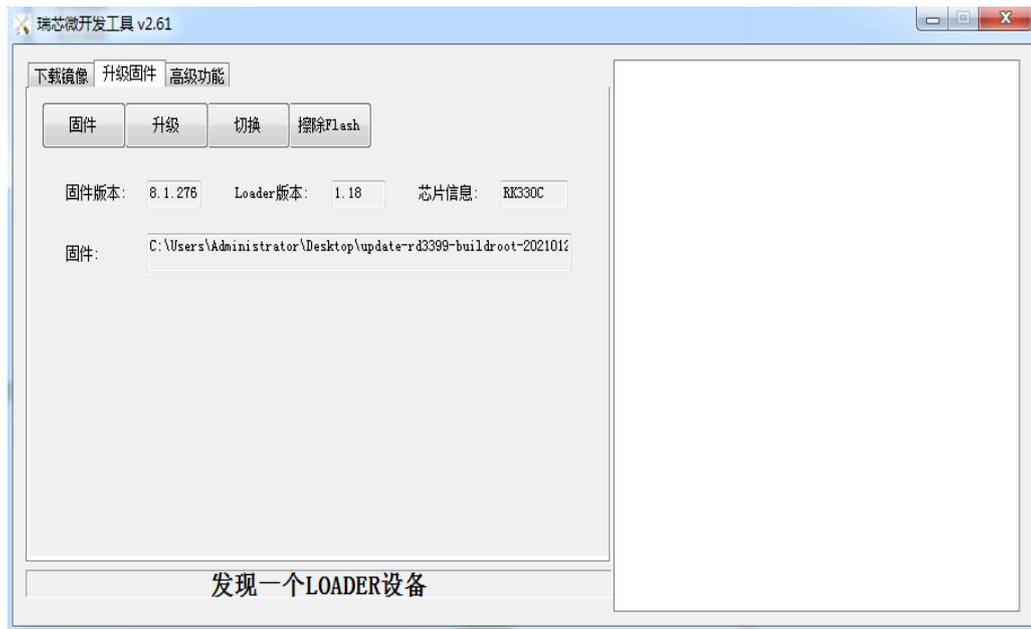


图 2.5 烧写工具 AndroidTool

2.1.4 擦除 FLASH 操作

板子切换到 loader 模式，选择对应的升级固件，进入升级工具进入擦除 Flash

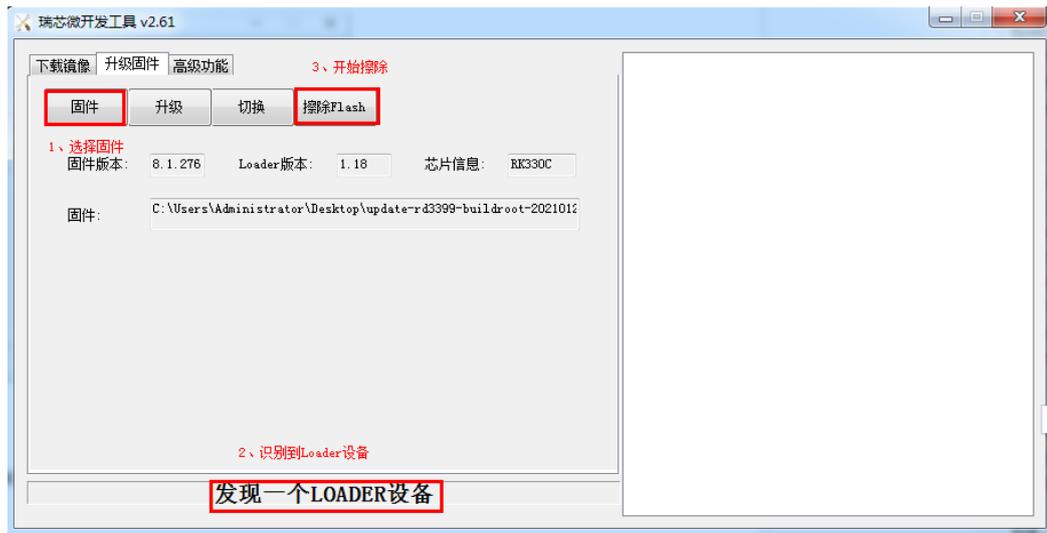


图 2.6 AndroidTool 擦除 Flash

2.1.5 整包固件升级

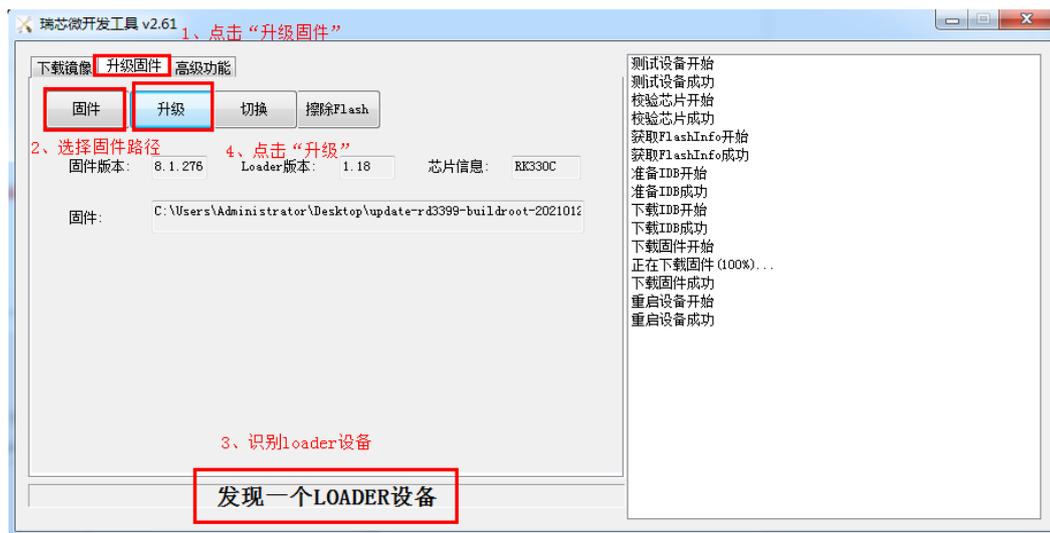


图 2.7 AndroidTool 升级固件

2.2 TF 卡烧写

2.2.1 TF 卡制作与烧写

首先格式化 TF 卡，将固件拷贝到 TF 卡中，命名为 `sdupdate.img`，同时创建一个新文件，文件名为 `sd_boot_config.config`，并添加一下内容

```
loader_update = 0
display_led = 1
display_lcd = 1
pcba_test = 0
fw_update = 1
```

demo_copy = 0

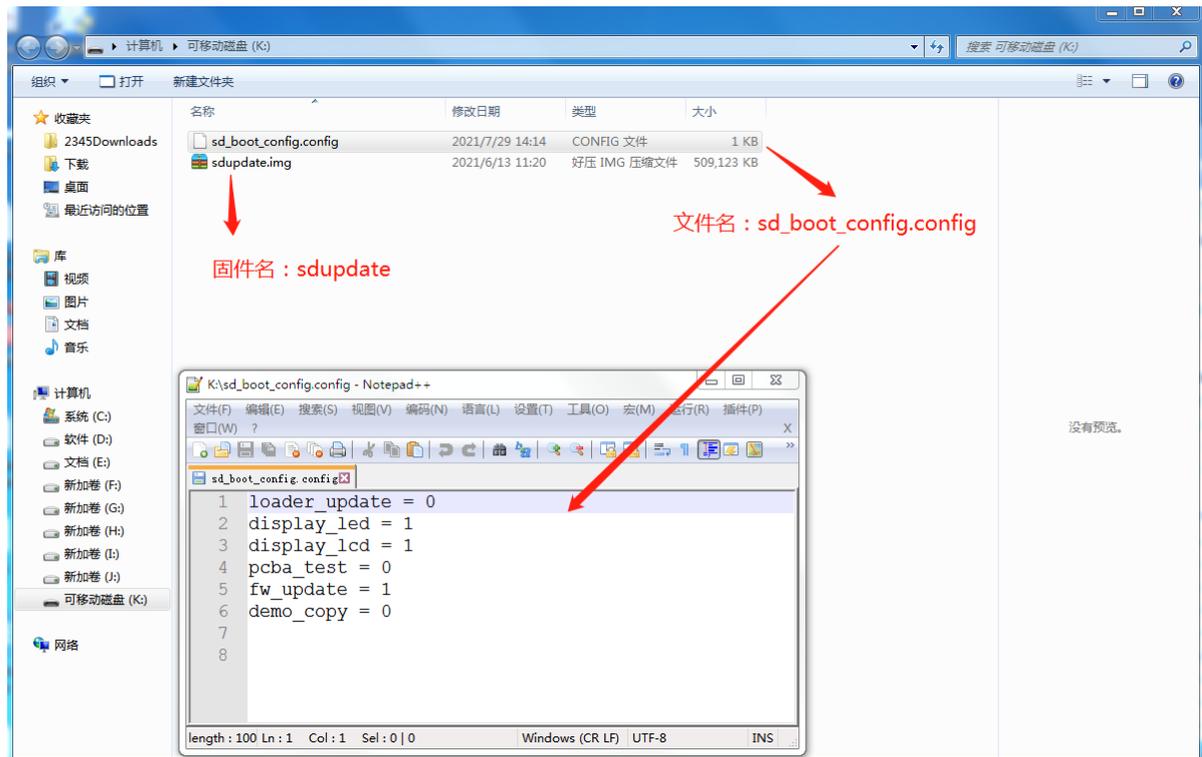


图 2.8 TF 卡升级

将 TF 卡插入开发板并启动，系统将自动进入烧写流程。烧写完成后屏幕和串口都将提示：

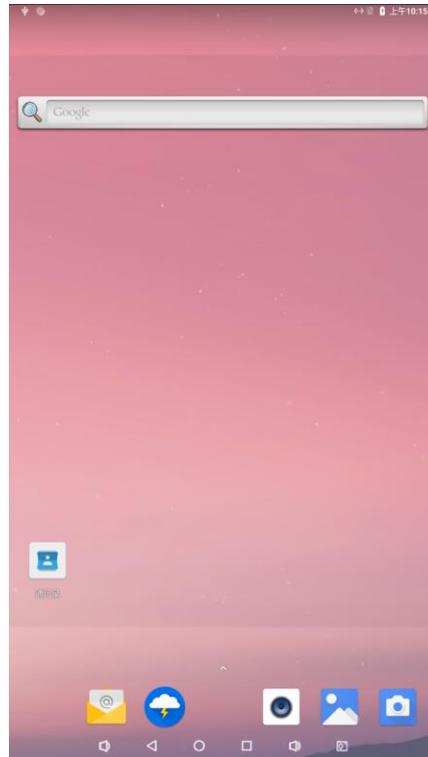
SD upgrade ok.

Doing Actions succeeded.please remove the sdcard.....

此时，拔出 TF 卡，系统自动重新启动。

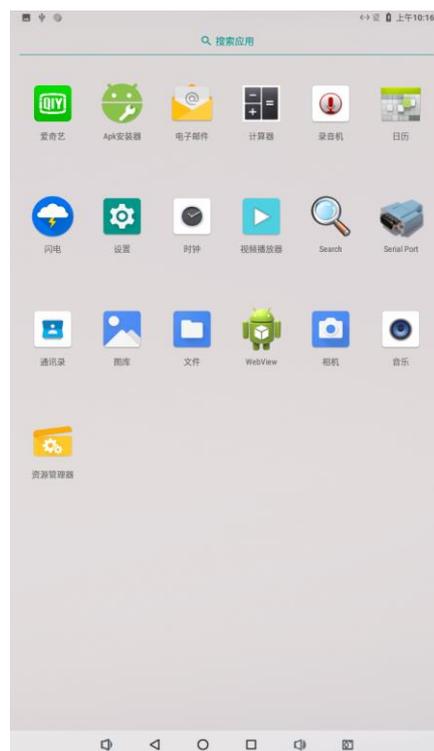
3 Android 功能使用及测试

3.1 主界面展示



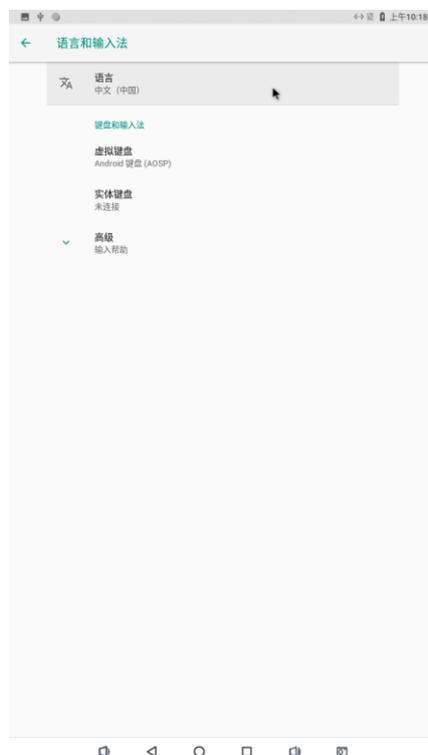
3.2 应用程序

在主界面向上滑动，即可出现下面的界面。

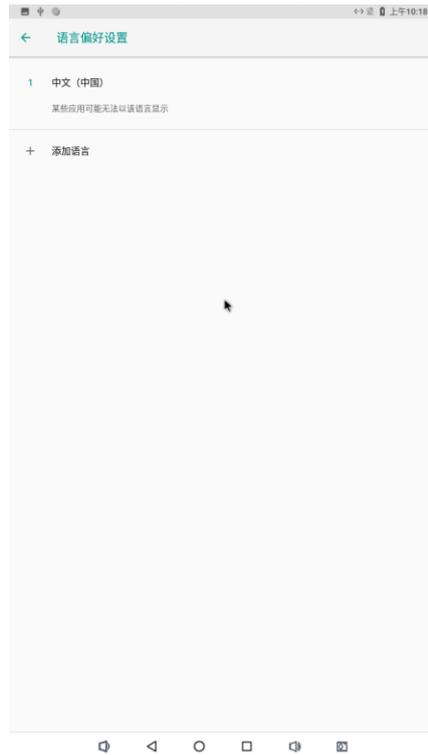


3.3 设置语言

在应用程序界面点击“”，进入设置界面，点击系统
点击“语言和输入法”，进入语言设置界面：



点击“语言”，进入语言选择界面：



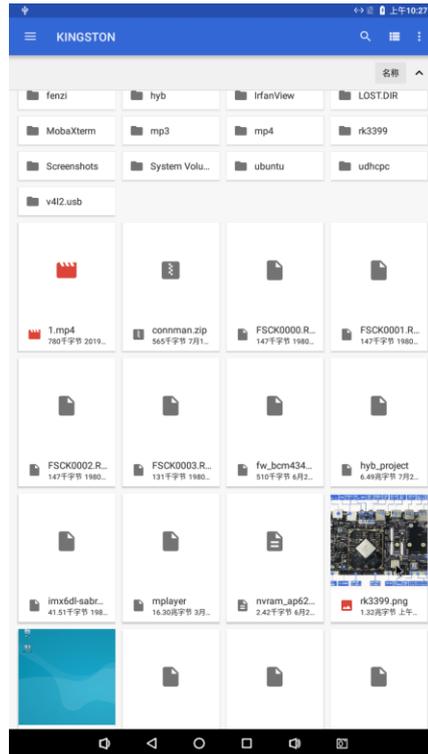
点击“添加语言”进行新的语言添加以及设置默认语言。

3.4 查看图片和视频

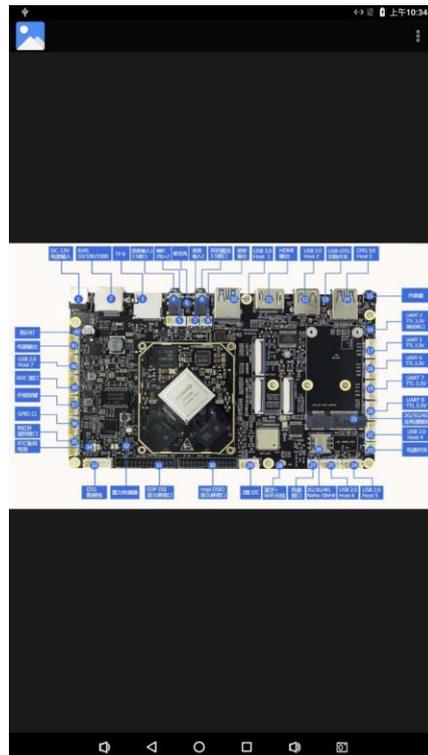
将存有图片和视频的 U 盘插入到 TW-AC3399 开发板中。



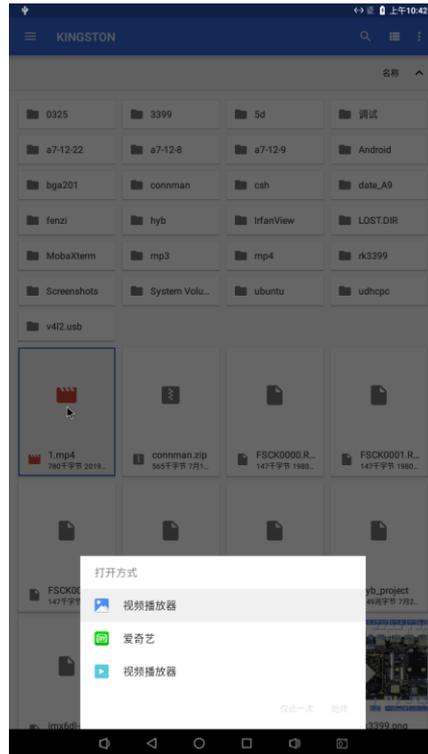
系统识别到 U 盘插入后将在菜单盘显示 U 盘菜单，点击浏览查看 U 盘文件，即可以看到 U 盘中的所有文件。



点击图片文件



触摸屏支持多点触摸，因此可以使用手指滑动进行放大缩小操作。
返回到 U 盘界面，点击视频文件。



首次播放，需要选择播放器，选择始终或仅此一次后，开始播放视频。

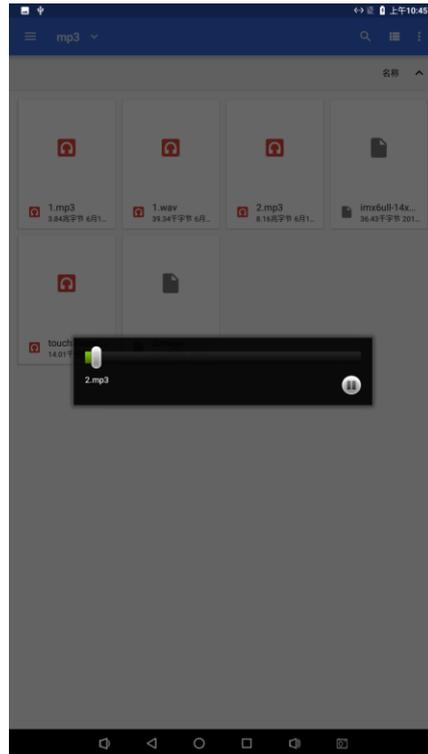


点击“允许”，批准播放器访问设备多媒体文件。

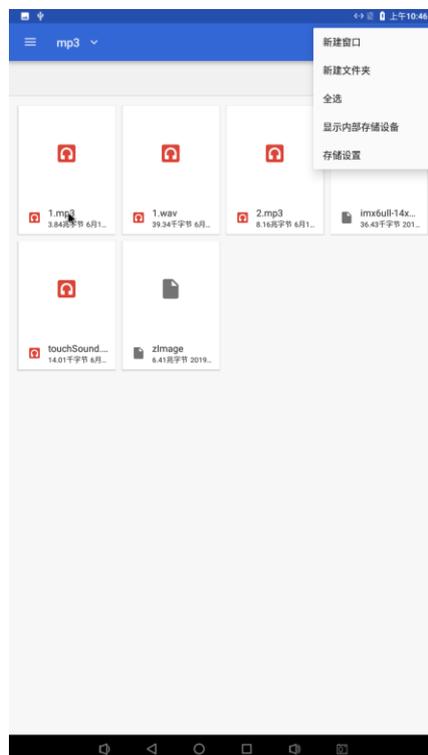
3.5 播放音乐

将存有音频文件的 U 盘插入到 TW-AC3399 开发板中。

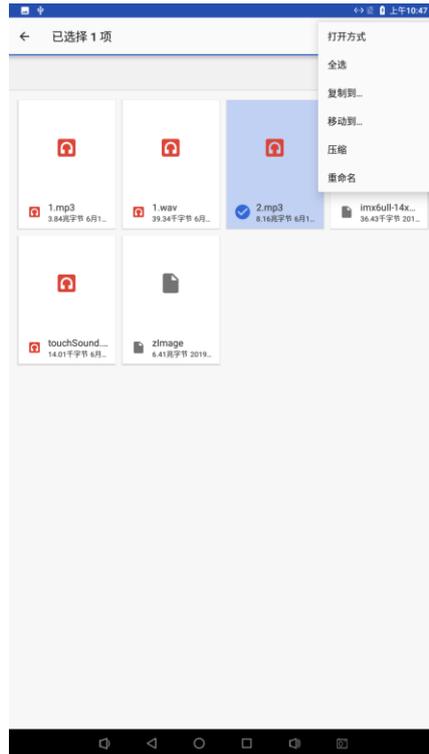
在此处播放音频无法后台，可以点击右上角的设置按钮显示内部存储。



此种播放模式下无法后台，可以将音频文件复制到内部存储中，打开音频应用进行播放。首先点击右上角的设置按钮选择“显示内部存储设备”。



长按需要播放的音频文件，点击右上角的设置按钮，选择“复制到...”



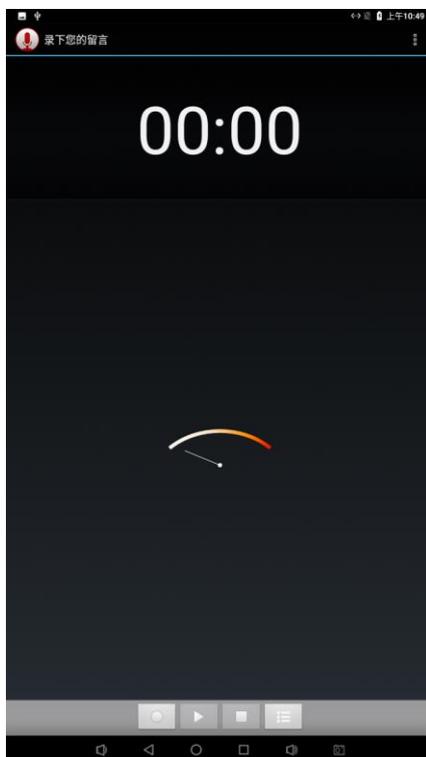
在内部存储中找到 Music 文件夹，点击右下角的“复制”按钮。



打开桌面上音乐应用，点击音乐名称，自动进行播放。

3.6 录音（支持 Mic 输入）

选择桌面 “” 录音机应用进入录音：

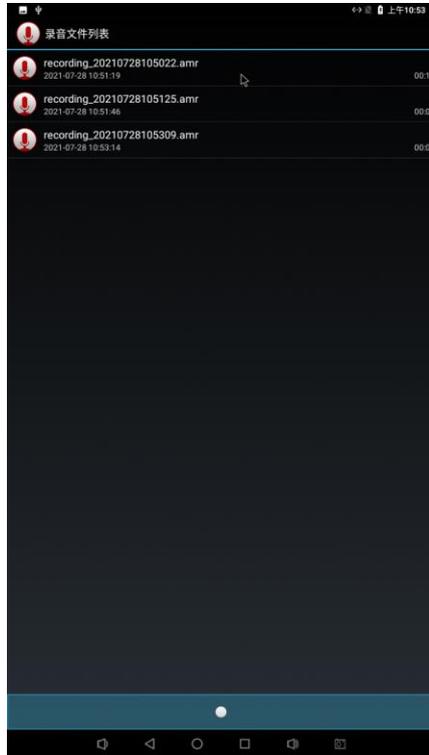


点击圆形按钮开始进行录制

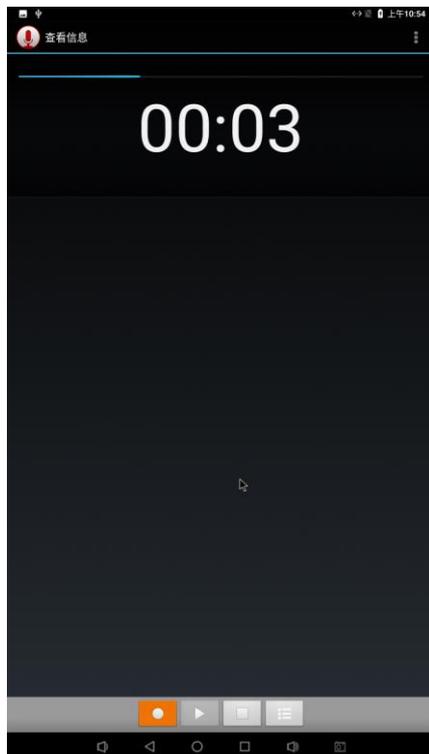


点击正方形按钮停止录制，点击三角形按钮进行播放，完成后，可以点击存入 sd 卡或者存入本地。

点击最右边的“☰”，打开录制的音频文件列表



点击录制的音频文件可以播放试听。



3.7 调节音量

在应用程序界面点击“”，进入设置界面：



在设置界面点击“声音”，进入音量设置界面



在此界面可以调节各个部分的音量。

3.8 背光控制

在应用程序界面点击“”，进入设置界面：



在设置界面点击“显示”，进入显示设置界面选择“亮度”进行背光设置，出现亮度调节滑块，调节亮度。TW-AC3399 出厂时默认设置为永不休眠，如果您需要休眠唤醒，请点击休眠选项，选择休眠时间。



拖动滑动条可以进行背光设置，点击休眠按钮可以选择屏幕休眠时间。



3.9 设置时间(RTC)

在应用程序界面点击“”，进入设置界面：



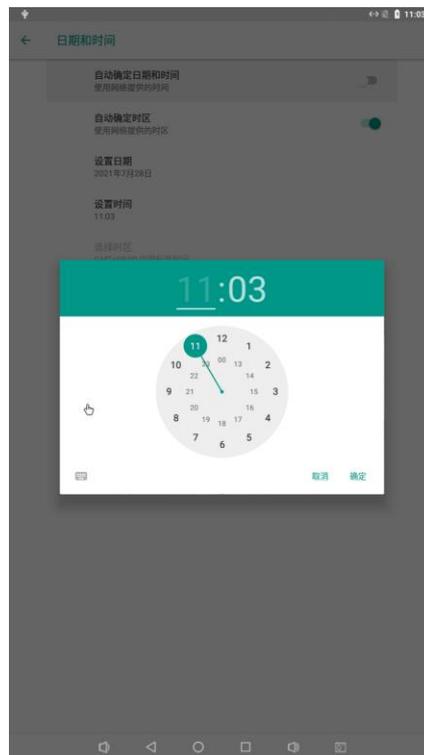
选择“系统”，在这里可以更改日期和时间，并且在您断电之后时间仍可同步更新（确保板上已经安装了纽扣电池）。



关闭“自动确定日期和时间”，分别设置日期和时间。



点击“设置日期”



点击“设置日期”和“设置时间”设置好之后，就可以给板子断电再上电，再次进入时间设置界面，就会看到时间已经同步更新了。

3.10 以太网测试

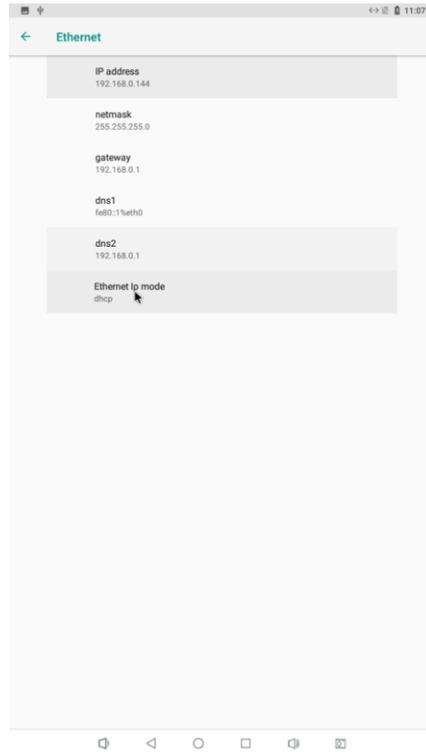
插入网线后，打开设置,如下图：



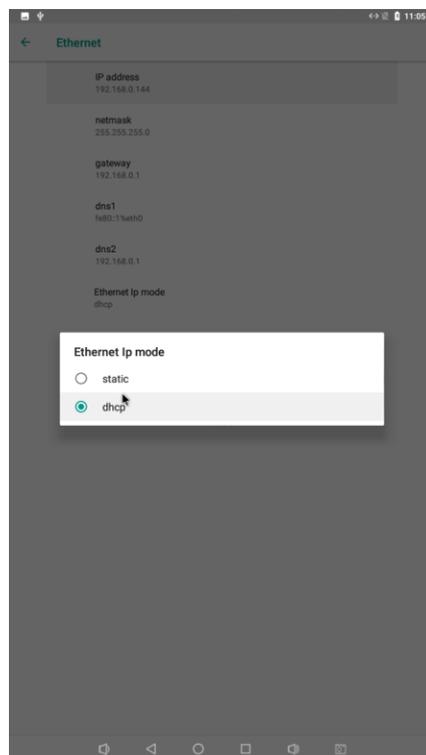
点击网络和互联网

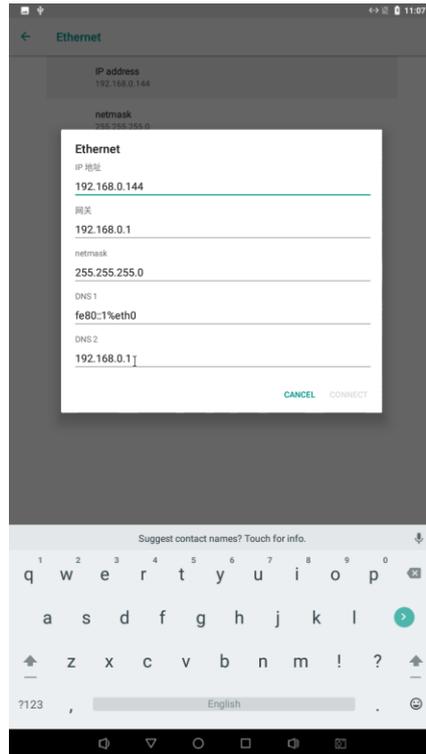


点击“Ethernet”可以选择自动获取 IP DHCP 或者静态 IP。推荐使用 DHCP，如果设置静态 IP，请确保您的网络参数可用。



默认获取 IP 方式为“dhcp”，如果要设置静态 IP，点击“Ethernet mode”，进行模式选择。





配置完成后点击 connect，保存设置。

注意：在 4G、wifi、以太网同时存在的情况下，默认优先使用以太网。

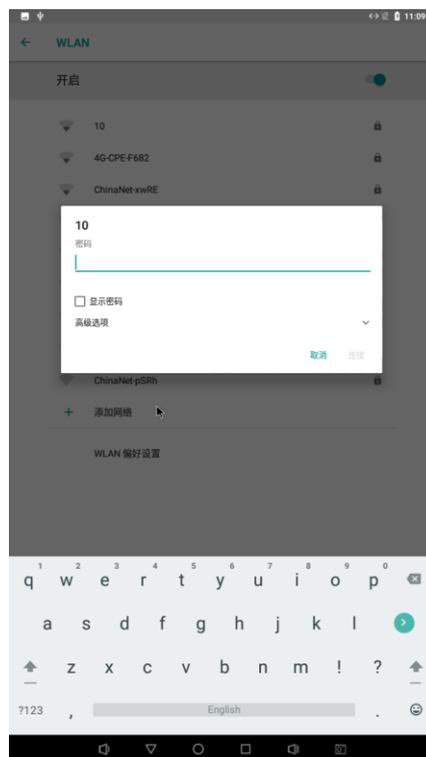
3.11 WiFi 测试

WiFi 测试使用 wifi&Bluetooth 一体模块，选择设置，界面如下：



注意：测试 WiFi 时，将有线网络拔掉。

点击“网络和互联网”，打开 WIFI 开关。



点击需要连接的热点并输入密码。连接成功后可以打开桌面的浏览器进行网上测试。



3.12 蓝牙测试

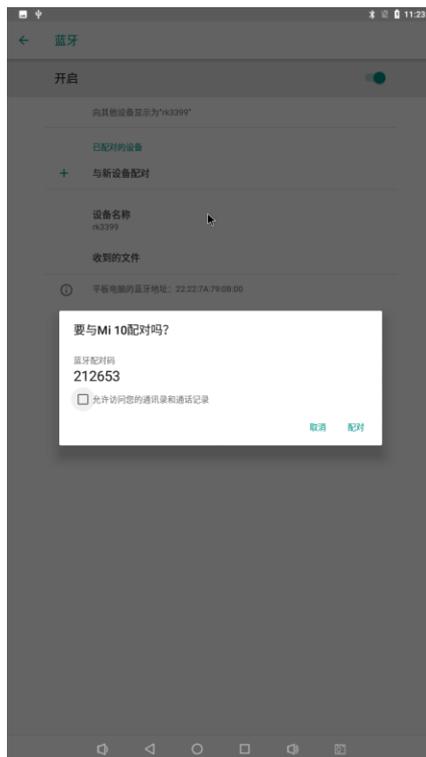
蓝牙功能测试使用 WiFi&Bluetooth 一体模块，测试方法如下：
选择设置：



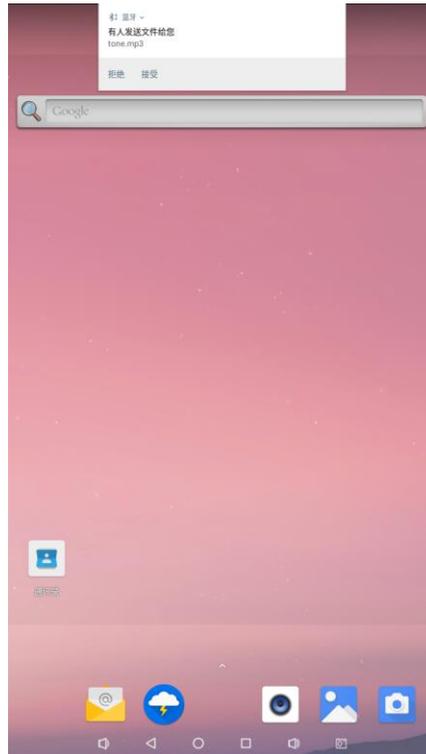
点击“蓝牙”进入蓝牙设置界面。



点击右上角图标打开蓝牙开关，同时打开手机蓝牙进行扫描，点击“rk3399”进行配对。



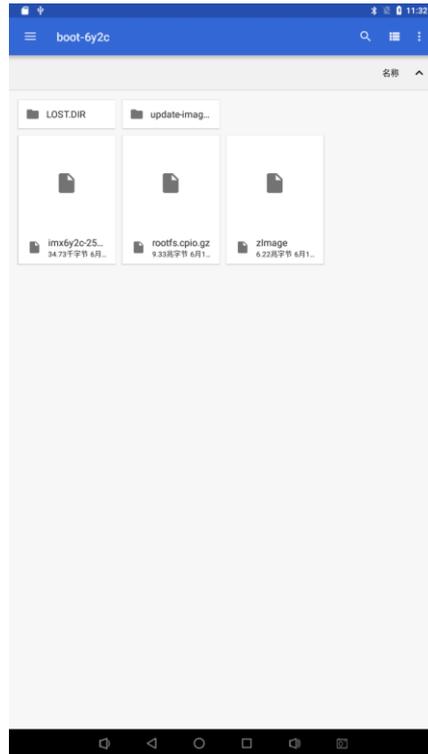
点击可用设备中的手机设备（以实际为准），开始配对。配对成功后可以在手机上通过蓝牙发送文件给开发板。



3.13 TF 卡及 USB 存储测试

本测试为 TF 卡及 usb 存储设备的测试。将 TF 卡设备插入到开发板中。系统将自动检测到 TF 卡的插入。例如这里点击“SD 卡”，进行文件浏览。





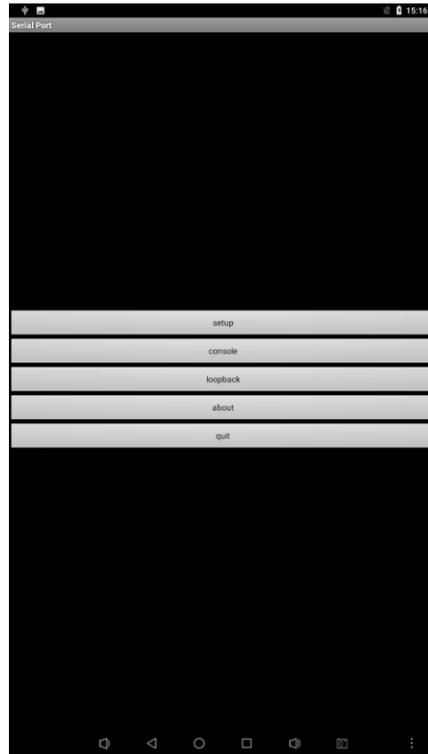
3.14 USB 鼠标测试

系统运行之后,在 USB host 上插入 USB 鼠标,您就会在界面内看到鼠标光标“”,您可以通过鼠标操作 Android 系统。

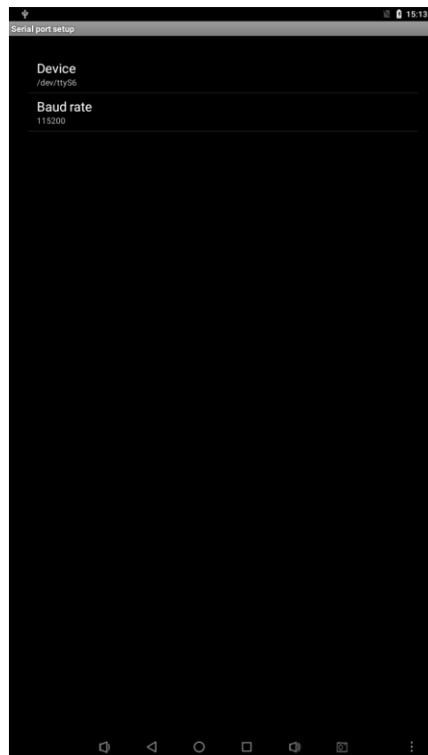
3.15 串口测试



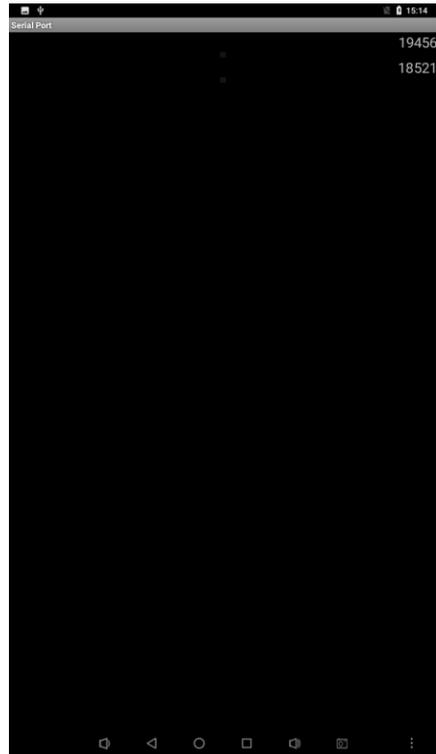
打开桌面上的串口测试程序



测试串口 `uart6` 可以采用回环测试的方法，将 `uart6` 的收发引脚进行短接，点击 `setup`，Device 选择 `/dev/ttyS6`，Baudrate 选择 `115200`



然后点击上一级菜单中的“Loopback”选项进行回环测试。



3.16 4G 拨号上网测试

TW-AC3399 支持 EC20 CEHC 4G 模块上网，支持中国移动与中国联通 4G、3G、2G。断电，将 EC20 模块安装到底板的 MINI-PCIE 接口后，上电测试。



打开 4G 开关，点击桌面浏览器应用进行 4G 上网测试。



关于首选网络类型：

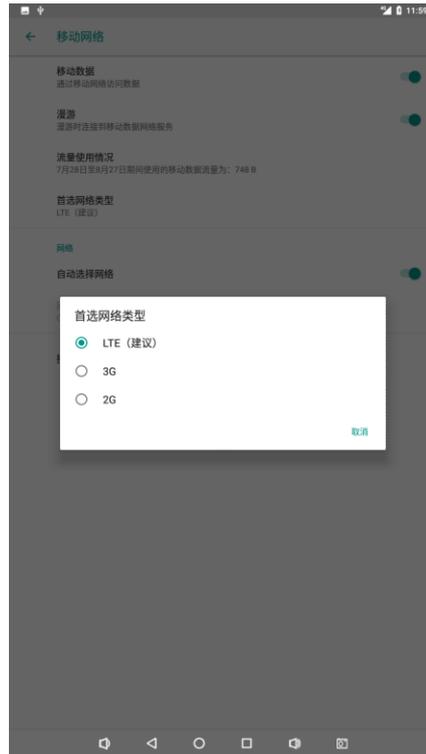
默认镜像启动时默认的首选网络类型可能不符合实际客户需求，例如插 4G 电信卡时，镜像默认的首选网络类型为 3G，当然您可以在设置中将其修改为 4G 网络。

如果您想设置镜像默认的首选网络类型可以进行如下操作：

1、判断您想要的网络类型 id

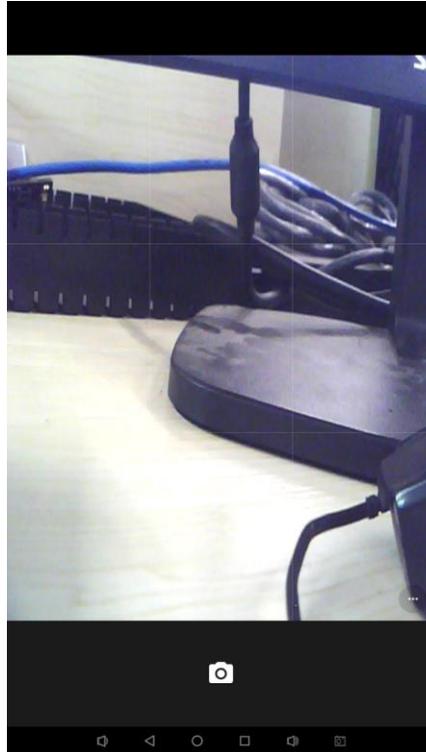
```
logcat -s NetworkSettings | grep settingsNetworkMode
```

然后手动修改首选网络类型，选择您想要的网络类型

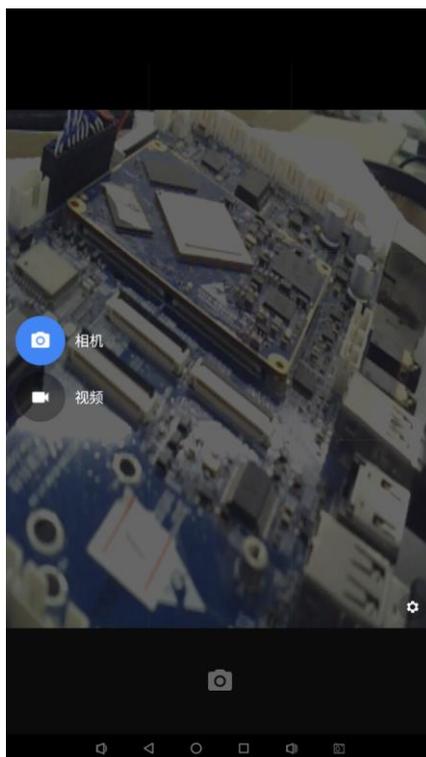


3.17 USB 摄像头测试

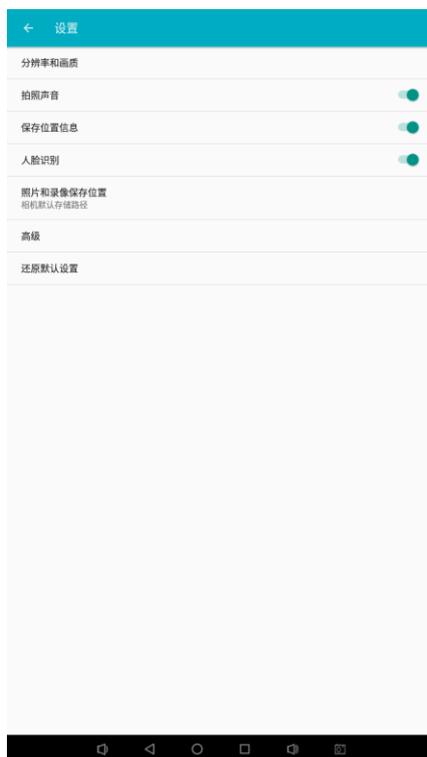
将 USB 摄像头插到 TW-AC3399 开发板，打开相机，并点击拍照按钮进行拍照测试



在屏幕左侧向右滑动，将出现设置页面。在此界面可以进行拍照模式和录像模式的切换，同时也可以点击右上角的设置按钮进行参数设置。



点击右上角的设置按钮，进入参数设置界面。



点击下面视频按钮开始录像。



4 系统定制

4.1 界面去掉顶部 Google 搜索栏

在/build/target/produced/core.mk

删掉 QuickSearchBox

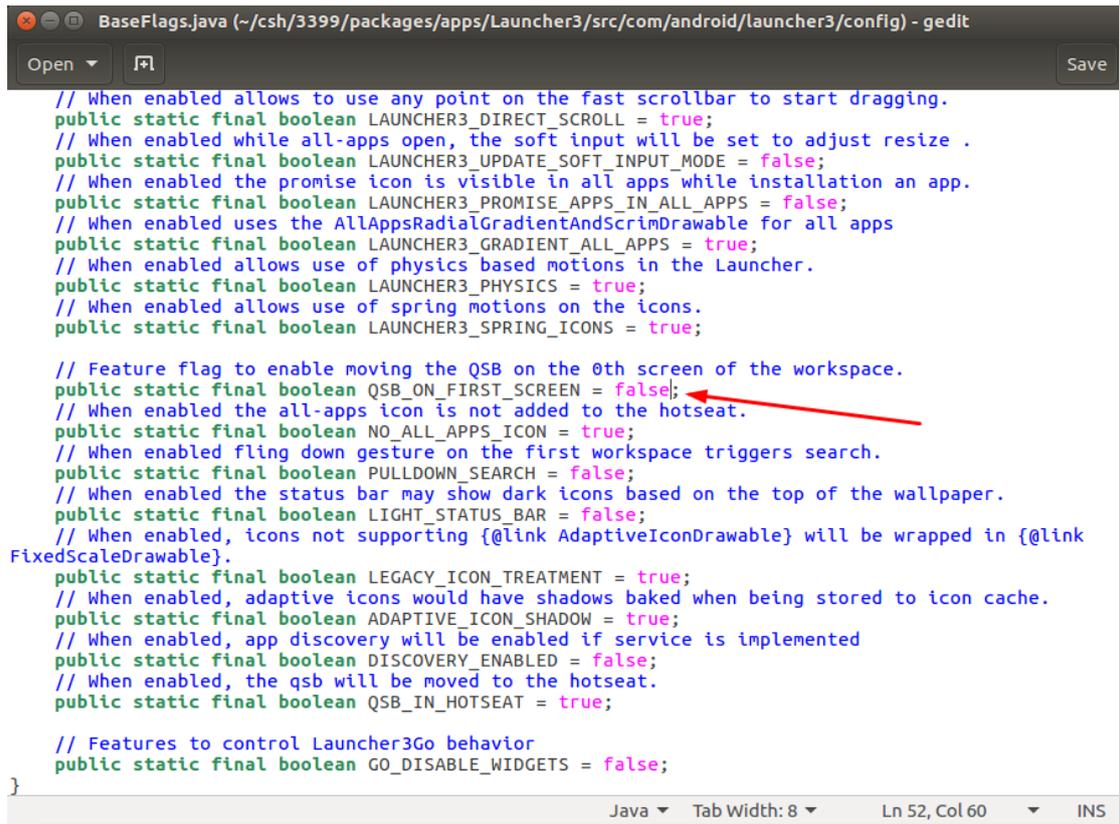
```
InputDevices \
KeyChain \
Keyguard \
LatinIME \
Launcher2 \
ManagedProvisioning \
MtpDocumentsProvider \
PicoTts \
PacProcessor \
libpac \
PrintSpooler \
PrintRecommendationService \
ProxyHandler \
QuickSearchBox
Settings \
SharedStorageBackup \
StorageManager \
Telecom \
TeleService \
VpnDialogs \
vr \
MmsService

# The set of packages whose code can be loaded by the system server.
PRODUCT_SYSTEM_SERVER_APPS += \
    FusedLocation \
    InputDevices \
    KeyChain \
    Telecom \

# The set of packages we want to force 'speed' compilation on.
PRODUCT_DEXPREOPT_SPEED_APPS += \
```

图 4.1 core.mk

将 packages/apps/Launcher3/src/com/android/launcher3/config/BaseFlags.java 中的 public static final boolean QSB_ON_FIRST_SCREEN = true; 修改为 false



```
BaseFlags.java (~/.csh/3399/packages/apps/Launcher3/src/com/android/launcher3/config) - gedit
Open Save
// When enabled allows to use any point on the fast scrollbar to start dragging.
public static final boolean LAUNCHER3_DIRECT_SCROLL = true;
// When enabled while all-apps open, the soft input will be set to adjust resize .
public static final boolean LAUNCHER3_UPDATE_SOFT_INPUT_MODE = false;
// When enabled the promise icon is visible in all apps while installation an app.
public static final boolean LAUNCHER3_PROMISE_APPS_IN_ALL_APPS = false;
// When enabled uses the AllAppsRadialGradientAndScrimDrawable for all apps
public static final boolean LAUNCHER3_GRADIENT_ALL_APPS = true;
// When enabled allows use of physics based motions in the Launcher.
public static final boolean LAUNCHER3_PHYSICS = true;
// When enabled allows use of spring motions on the icons.
public static final boolean LAUNCHER3_SPRING_ICONS = true;

// Feature flag to enable moving the QSB on the 0th screen of the workspace.
public static final boolean QSB_ON_FIRST_SCREEN = false;
// When enabled the all-apps icon is not added to the hotseat.
public static final boolean NO_ALL_APPS_ICON = true;
// When enabled fling down gesture on the first workspace triggers search.
public static final boolean PULLDOWN_SEARCH = false;
// When enabled the status bar may show dark icons based on the top of the wallpaper.
public static final boolean LIGHT_STATUS_BAR = false;
// When enabled, icons not supporting {@link AdaptiveIconDrawable} will be wrapped in {@link
FixedScaleDrawable}.
public static final boolean LEGACY_ICON_TREATMENT = true;
// When enabled, adaptive icons would have shadows baked when being stored to icon cache.
public static final boolean ADAPTIVE_ICON_SHADOW = true;
// When enabled, app discovery will be enabled if service is implemented
public static final boolean DISCOVERY_ENABLED = false;
// When enabled, the qsb will be moved to the hotseat.
public static final boolean QSB_IN_HOTSEAT = true;

// Features to control Launcher3Go behavior
public static final boolean GO_DISABLE_WIDGETS = false;
}
```

图 4.2 BaseFlags.java

4.2 系统预装 APP 的目录

在 device/rockchip/rk3399 目录下，存在 preinstall、preinstall_del、preinstall_del_forever，把需要安装的软件放在对应目录即可。

Preinstall: 预置不可卸载 apk 将 apk 文件放置该目录即可 apk 最好不带中文。

preinstall_del: 预置可卸载 apk (恢复出厂设置可恢复) 将 apk 文件放置该目录即可 apk 最好不带中文。

preinstall_del_forever: 预置可卸载 apk (恢复出厂设置不可恢复) 将 apk 文件放置该目录即可 apk 最好不带中文。

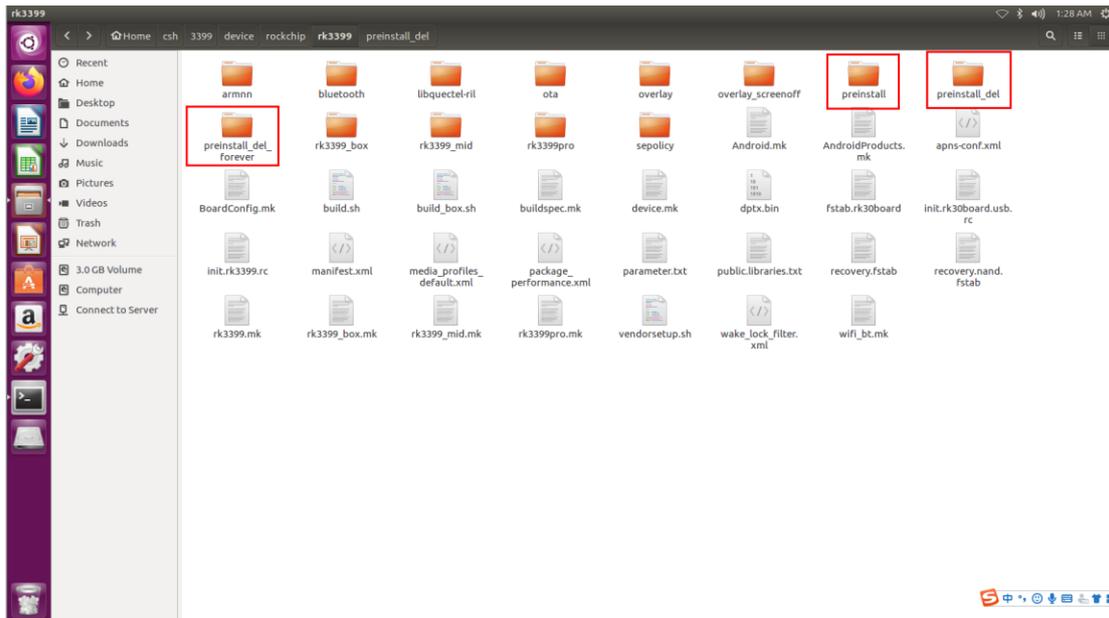


图 4.3 apk 安装路径

删除系统自带的 app

```
/vendor/rockchip/rpdzkj/apps/preinstall
/out/target/product/rk3399/vendor/usr/preinstall
```

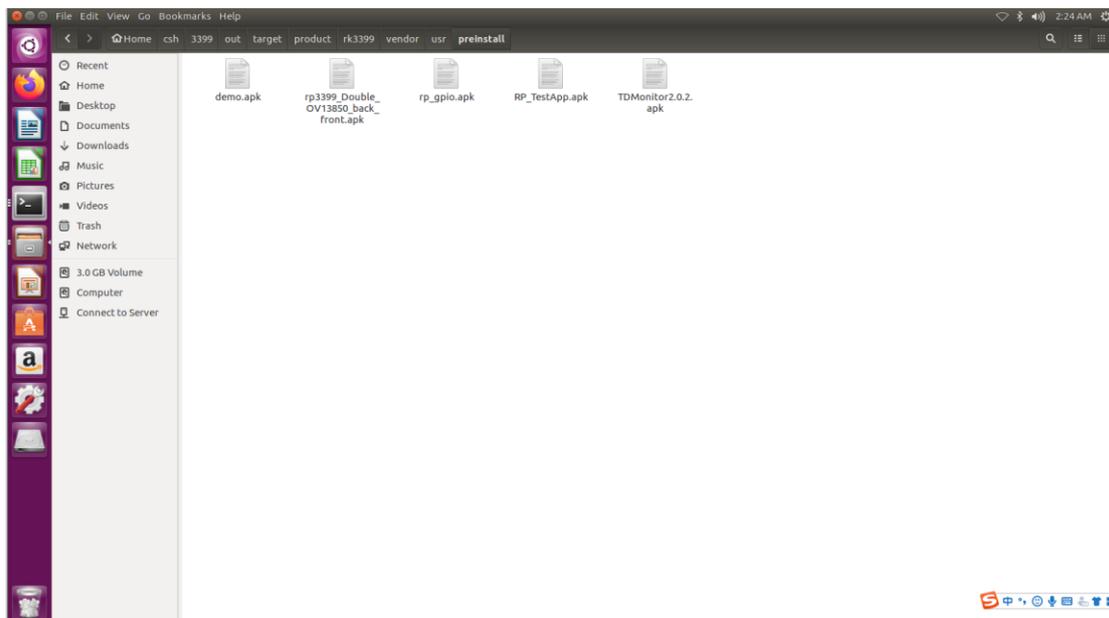
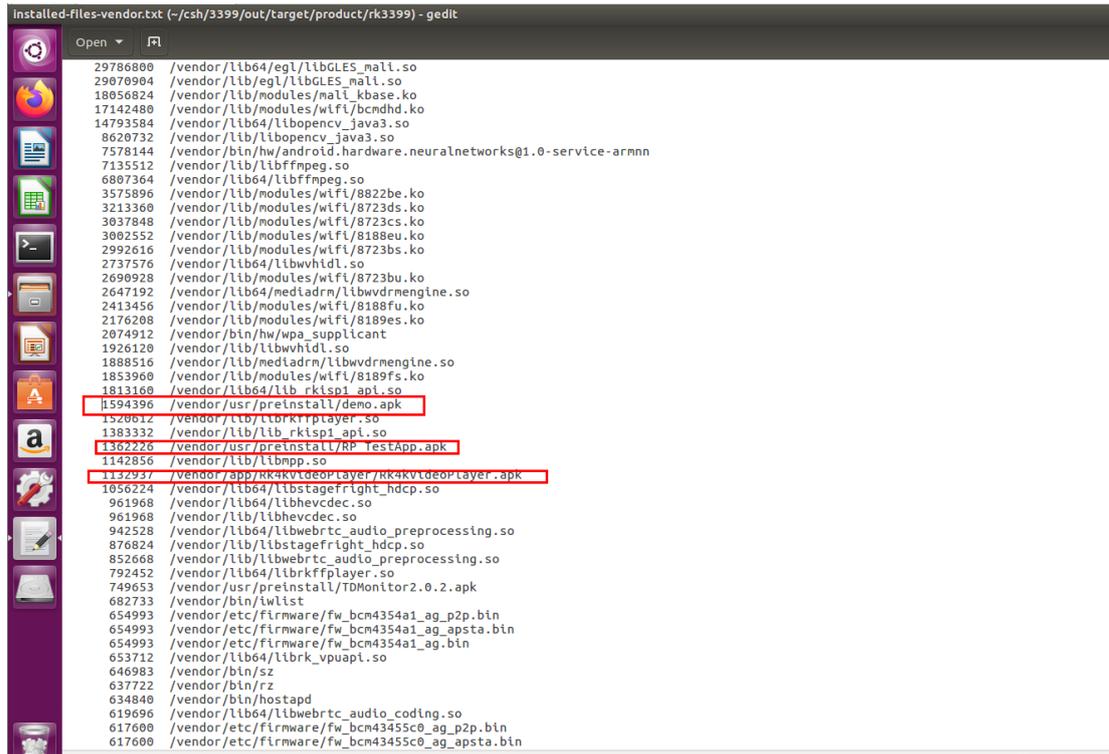


图 4.4 preinstall

还可以将 `out/target/product/rk3399/installed-files-vendor.txt` 文件中找到需要删除的 apk 配置，将其删除。



```
installed-files-vendor.txt (-/csh/3399/out/target/product/rk3399) - gedit
29786800 /vendor/lib64/egl/libGLES_mali.so
29878984 /vendor/lib/egl/libGLES_mali.so
18056824 /vendor/lib/modules/mali_kbase.ko
17142480 /vendor/lib/modules/wifi/bcmdhd.ko
14793584 /vendor/lib64/libopencv_java3.so
8620732 /vendor/lib/libopencv_java3.so
7578144 /vendor/bin/hw/android.hardware.neuralnetworks@1.0-service-armnn
7135512 /vendor/lib/libffmpeg.so
6807364 /vendor/lib64/libffmpeg.so
3575896 /vendor/lib/modules/wifi/8822be.ko
3213360 /vendor/lib/modules/wifi/8723ds.ko
3037848 /vendor/lib/modules/wifi/8723cs.ko
3002552 /vendor/lib/modules/wifi/8188eu.ko
2992616 /vendor/lib/modules/wifi/8723bs.ko
2737576 /vendor/lib64/libwhidl.so
2690928 /vendor/lib/modules/wifi/8723bu.ko
2647192 /vendor/lib64/mediadrn/libwvdrngine.so
2413456 /vendor/lib/modules/wifi/8188fu.ko
2176208 /vendor/lib/modules/wifi/8189es.ko
2074912 /vendor/bin/hw/wpa_supplicant
1926120 /vendor/lib/libwhidl.so
1888516 /vendor/lib/mediadrn/libwvdrngine.so
1853960 /vendor/lib/modules/wifi/8189fs.ko
1813160 /vendor/lib64/lib_rkisp1_apl.so
1594396 /vendor/usr/preinstall/demo.apk
1520612 /vendor/lib/lib_rkisp1_apl.so
1383332 /vendor/lib/lib_rkisp1_apl.so
1362226 /vendor/usr/preinstall/RP_TestApp.apk
1142856 /vendor/lib/libnpp.so
1132937 /vendor/app/RK4KV1deorlayer/RK4KV1deorlayer.apk
1056224 /vendor/lib64/libstagefright_hdcp.so
961968 /vendor/lib64/libhevcdcc.so
961968 /vendor/lib/libhevcdcc.so
942528 /vendor/lib64/libwebrtc_audio_preprocessing.so
876824 /vendor/lib/libstagefright_hdcp.so
852668 /vendor/lib/libwebrtc_audio_preprocessing.so
792452 /vendor/lib64/librkffplayer.so
749653 /vendor/usr/preinstall/TDMonitor2.0.2.apk
682733 /vendor/bin/iwlist
654993 /vendor/etc/firmware/fw_bcm4354a1_ag_p2p.bin
654993 /vendor/etc/firmware/fw_bcm4354a1_ag_apsta.bin
654993 /vendor/etc/firmware/fw_bcm4354a1_ag.bin
653712 /vendor/lib64/librk_vpupl.so
646983 /vendor/bin/sz
637722 /vendor/bin/rz
634840 /vendor/bin/hostapd
619696 /vendor/lib64/libwebrtc_audio_coding.so
617600 /vendor/etc/firmware/fw_bcm43455c0_ag_p2p.bin
617600 /vendor/etc/firmware/fw_bcm43455c0_ag_apsta.bin
```

图 4.5/installed-files-vendor.txt

4.3 系统永不休眠设置

修改 frameworks/base/packages/SettingsProvider/res/values/defaults.xml 中的

def_screen_off_timeout 值，改为 2147483647，系统支持最大不休眠时间为 2147483647，大概 200 小时。

```

device.mk (~/.csh/3399/device/rockchip/common) - gedit
$ cat ../../../../hardware/tegrta/rk31/rk31.mk)

ifeq ($(strip $(TARGET_BOARD_PLATFORM_PRODUCT)), box)
    include device/rockchip/COMMON/samba/rk31_samba.mk
    PRODUCT_COPY_FILES += \
        $(LOCAL_PATH)/init.box.samba.rc:root/init.box.samba.rc

    PRODUCT_PROPERTY_OVERRIDES += \
        ro.rk.screenoff_time=2147483647
else
    PRODUCT_PROPERTY_OVERRIDES += \
        ro.rk.screenoff_time=2147483647 ←
endif

# Flash Lock Status reporting,
# GTS: com.google.android.gts.persistentdata.
# PersistentDataHostTest#testTestGetFlashLockState
PRODUCT_DEFAULT_PROPERTY_OVERRIDES += \
    ro.oem_unlock_supported=1

# setup dm-verity configs.
# uncomment the two lines if use verity
PRODUCT_SUPPORTS_BOOT_SIGNER := false
PRODUCT_SYSTEM_VERITY_PARTITION := /dev/block/by-name/system
PRODUCT_VENDOR_VERITY_PARTITION := /dev/block/by-name/vendor
$(call inherit-product, build/target/product/verity.mk)

# Add for function frp
ifeq ($(strip $(BUILD_WITH_GOOGLE_MARKET)), true)
ifeq ($(strip $(BUILD_WITH_GOOGLE_FRP)), true)
    PRODUCT_PROPERTY_OVERRIDES += \
        ro.frp.pst=/dev/block/by-name/frp
endif
endif

Makefile Tab Width: 8 Ln 661, Col 38 INS
    
```

图 4-6 defaults.xml

在 device/rockchip/common/devices.mk 中，将 else 下面的时间改为 2147483647。

```

defaults.xml (~/.csh/3399/frameworks/base/packages/SettingsProvider/res/values) - gedit
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
-->
<resources>
    <bool name="def_dim_screen">true</bool>
    <integer name="def_screen_off_timeout">2147483647</integer> ←
    <integer name="def_sleep_timeout">-1</integer>
    <bool name="def_airplane_mode_on">false</bool>
    <bool name="def_theater_mode_on">false</bool>
    <!-- Comma-separated list of bluetooth, wifi, and cell. -->
    <string name="def_airplane_mode_radios" translatable="false">cell,bluetooth,wifi,nfc,wimax</
string>
    <string name="airplane_mode_toggleable_radios" translatable="false">bluetooth,wifi,nfc</string>
    <string name="def_bluetooth_disabled_profiles" translatable="false">0</string>
    <bool name="def_auto_time">true</bool>
    <bool name="def_auto_time_zone">true</bool>
    <bool name="def_accelerometer_rotation">true</bool>
    <!-- Default screen brightness, from 0 to 255. 102 is 40%. -->
    <integer name="def_screen_brightness">255</integer>
    <bool name="def_screen_brightness_automatic_mode">false</bool>
    <fraction name="def_window_animation_scale">100%</fraction>
    <fraction name="def_window_transition_scale">100%</fraction>
    <bool name="def_haptic_feedback">true</bool>

    <bool name="def_bluetooth_on">true</bool>
    <bool name="def_wifi_display_on">false</bool>
    <bool name="def_install_non_market_apps">false</bool>
    <bool name="def_package_verifier_enable">true</bool>
    <!-- Comma-separated list of location providers.
    Network location is off by default because it requires
    user opt-in via Setup Wizard or Settings.
    
```

图 3-7 devices.mk

4.4 开机 logo 动画修改

4.4.1 第一阶段

准备 logo

制作新的 logo.bmp

图片属性和默认的 logo.bmp 一致，否则会出现颠倒异常
width, height 都为偶数

替换 logo

替换 kernel/logo.bmp 和 kernel/logo_kernel.bmp

4.4.2 第二阶段

蒙版图片动画是通过两张图片叠加移动实现动画效果，所以只需修改这 2 张图片
镂空蒙版图片:(要求背景为黑色，字体中间镂空)

```
frameworks/base/core/res/assets/images/android-logo-mask.png
```

镂空蒙版后面的闪光图片:

```
frameworks/base/core/res/assets/images/android-logo-shine.png
```

更改图片后，执行一下命令，需更新图片时间戳，防止源码不编译图片

```
touch frameworks/base/core/res/assets/images/android-logo-mask.png
```

4.5 默认壁纸修改

将一下目录的图片替换即可

720dp 路径: frameworks/base/core/res/res/drawable-sw720dp-nodpi/default_wallpaper.png

600dp 路径: frameworks/base/core/res/res/drawable-sw600dp-nodpi/default_wallpaper.png

默认: frameworks/base/core/res/res/drawable-nodpi

4.6 隐藏屏蔽通知栏、状态栏

修改 frameworks\base\core\res\res\values\dimens.xml 中的 status_bar_height 为:

```
<dimen name="status_bar_height">0dp</dimen>
```

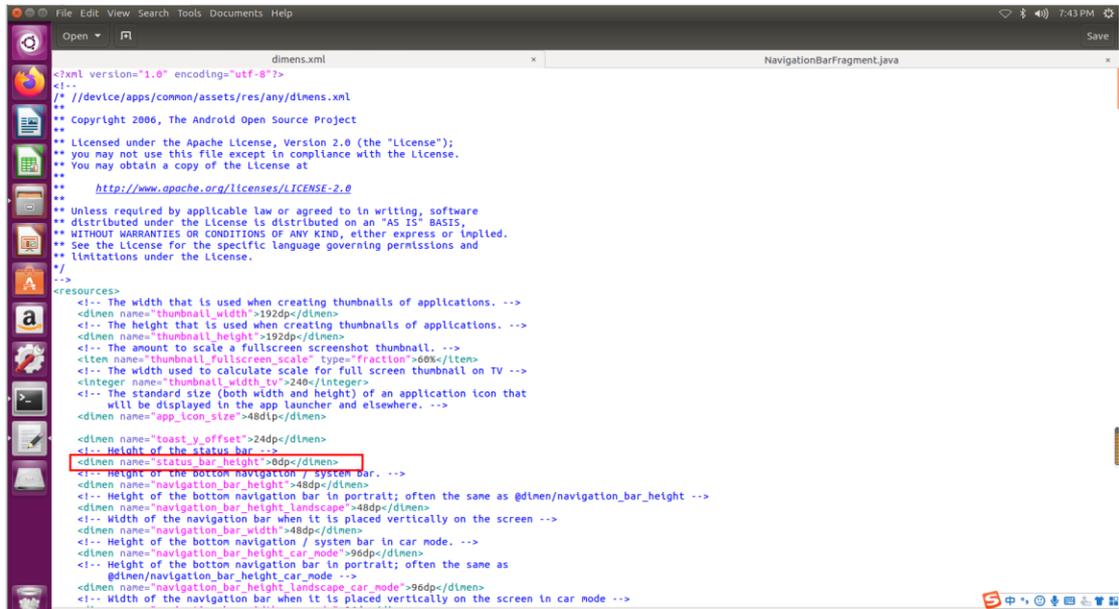


图 4.8 dimens.xml

其中配置项 status_bar_height 的值设为 0dp 就会隐藏 android 系统的状态栏

4.7 修改导航栏

修改导航栏高度，配置文件地址

/frameworks/base/core/res/res/values/dimens.xml

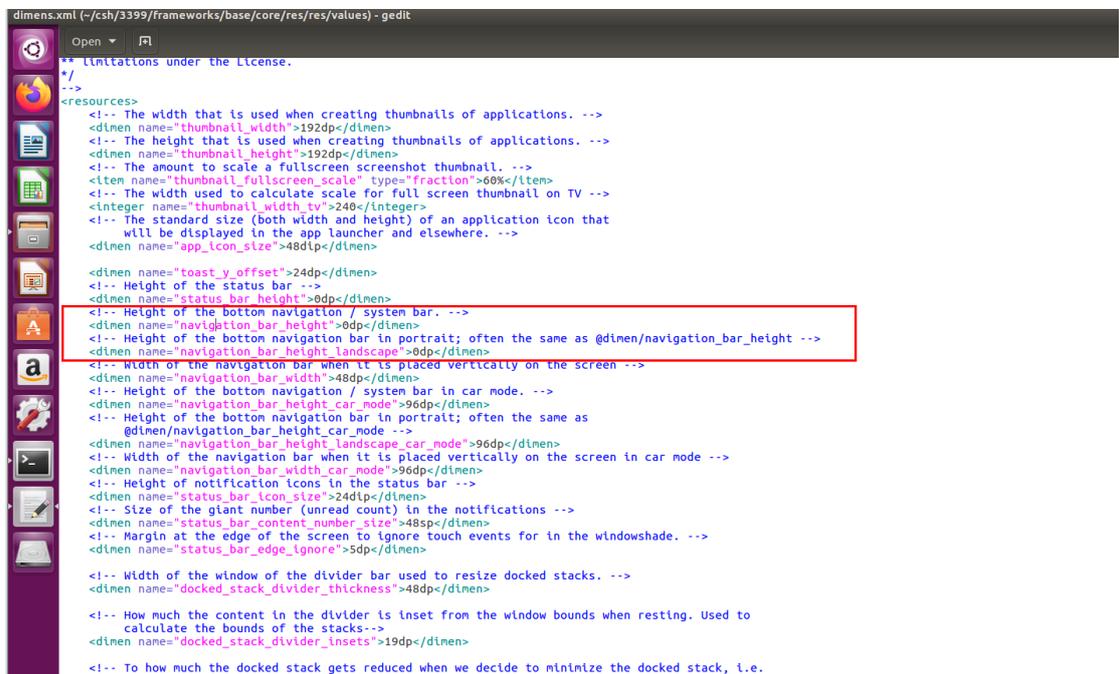


图 4.9 dimens.xml

高度为 0 即为隐藏。

4.8 默认语言的修改

语言的支持列表可以查看

```
build/make/target/product/locales_full.mk
```

代码修改

在 build/make/target/product/full_base.mk 里修改 PRODUCT_LOCALES 的值

```
PRODUCT_LOCALES := zh_CN
```

en-US: 英文

zh-CN: 中文

4.9 默认时区修改

在 build/tools/buildinfo.sh 中，添加

```
echo "persist.sys.timezone=Asia/Shanghai"
```

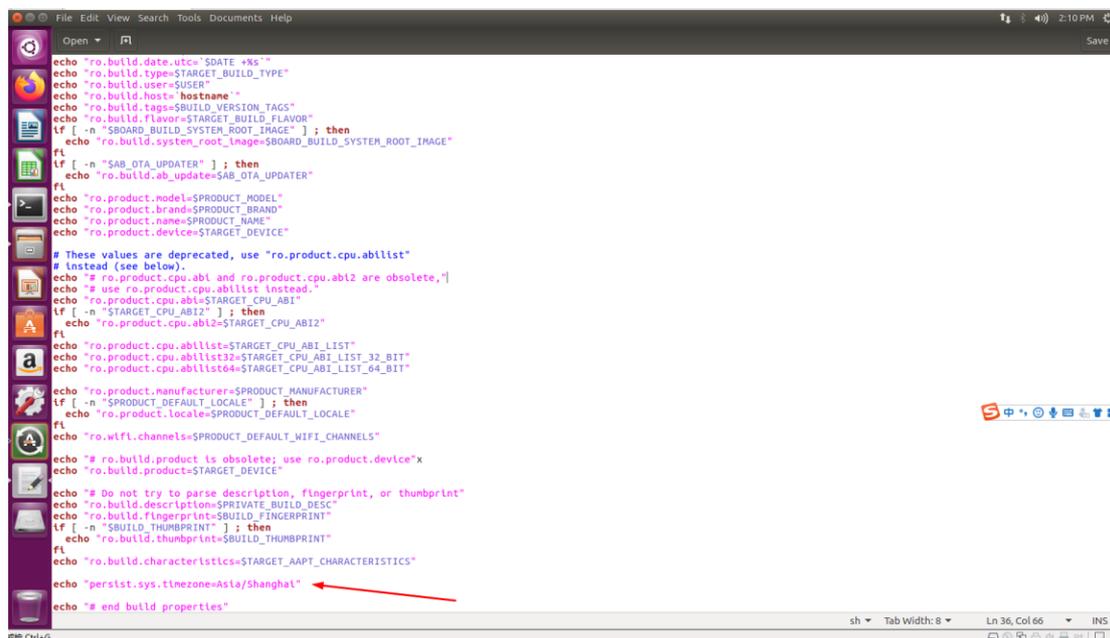


图 4.10 buildinfo.sh

在 device/rockchip/common/system.prop 中，修改

```
persist.sys.timezone = Asia/Shanghai
```

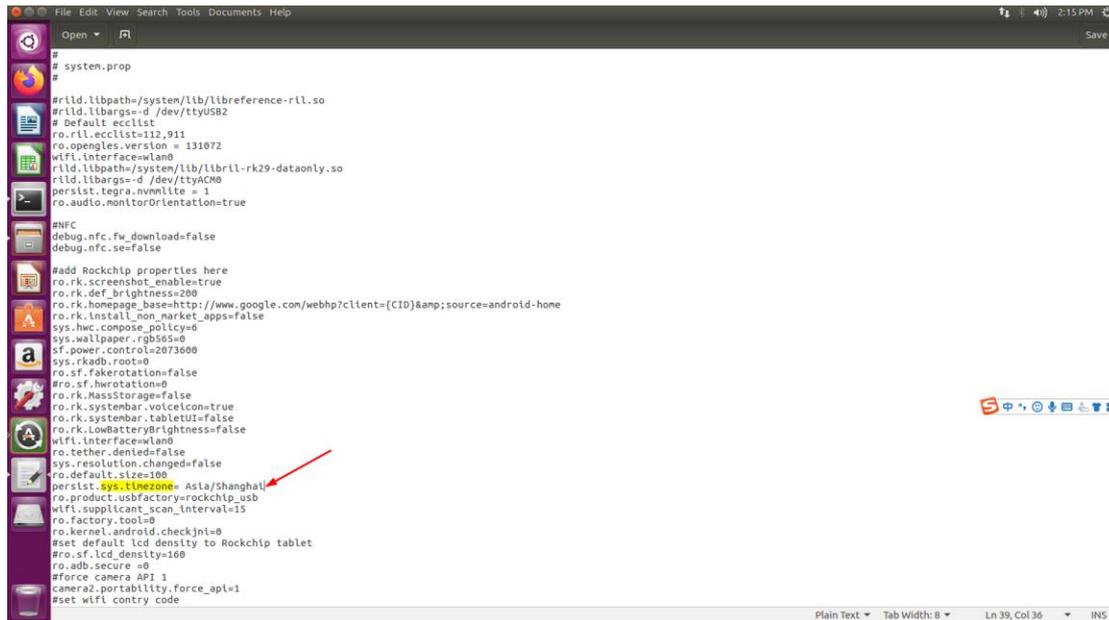


图 4.11 system.prop

时区整理

Asia/Shanghai	北京/中国
Asia/Hong_Kong	香港/中国
Asia/Taipei	台北时间 (台北)

4.10 打开或关闭触摸声音

在 frameworks/base/packages/SettingsProvider/res/values/defaults.xml 文件里，修改 def_sound_effects_enabled 的值。

false：关闭

true：打开

4.11 去除 USB 权限弹窗

在

frameworks/base/packages/SystemUI/src/com/android/systemui/usb/UsbPermissionActivity.java 中的 onCreate () 方法中修改：setupAlert ()

4.11.1 指定包名

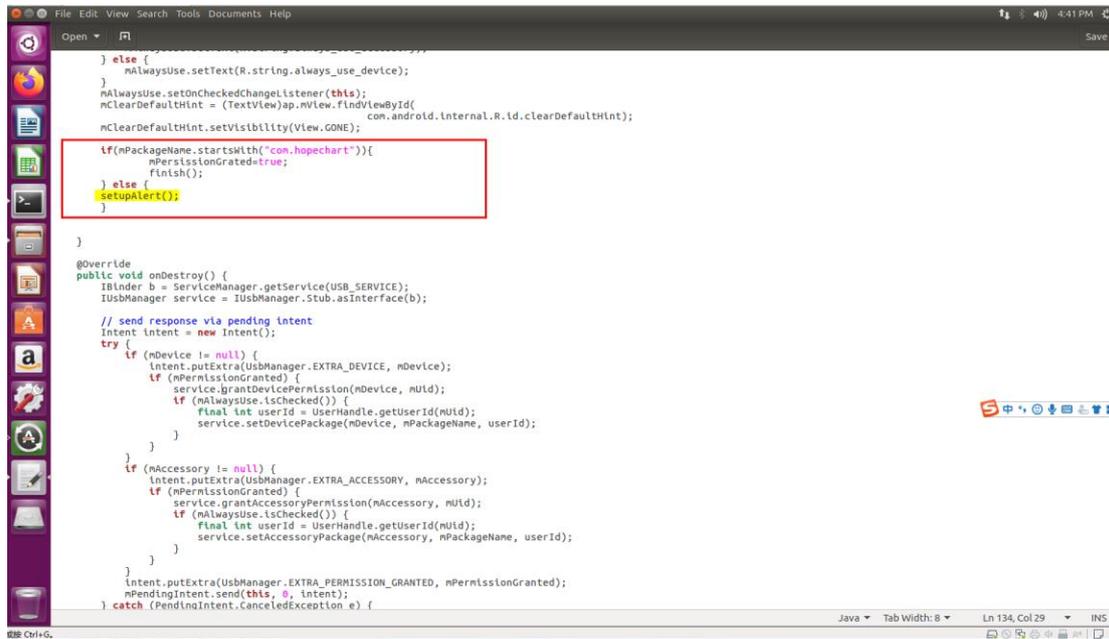


图 4.12 指定包名去掉弹窗

4.11.2 允许所有 APP 无弹窗

setupAlert 修改为:

mPermissionGranted = true;

finish();

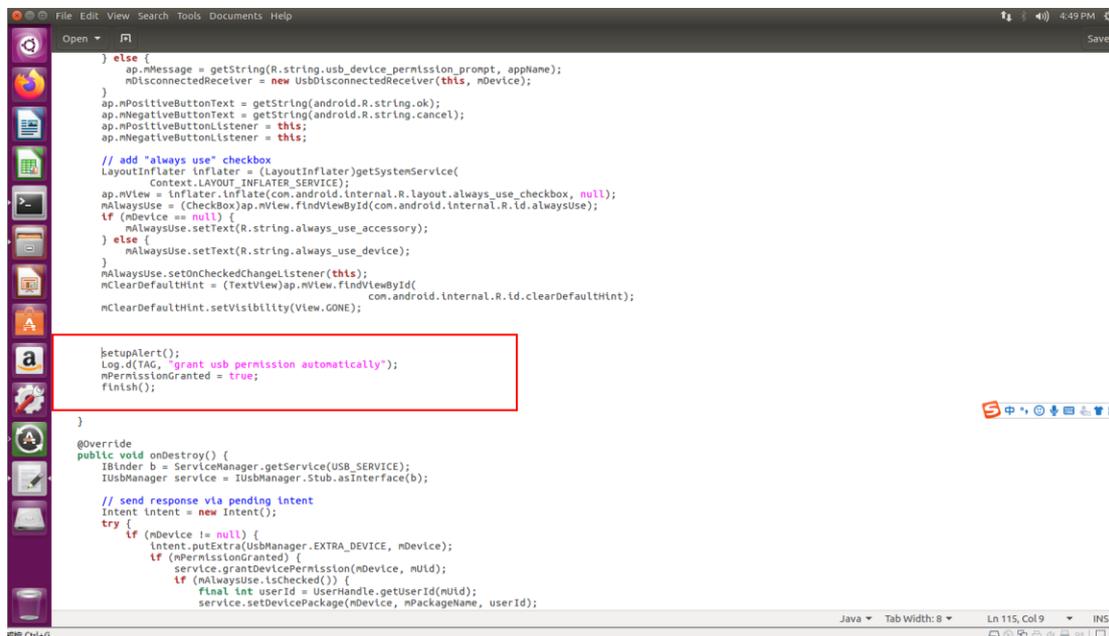


图 4.13 所有 APP 去掉弹窗

5 ADB 的使用

5.1 USB 的方式

主机通过 USB 连接设备

USB Type C 的接口

设置->系统->开发者选项->打开 USB 调试

5.2 网络的方式

主机通过网络连接设备（主机和设备在同一个局域网内）

在 windows cmd 命令窗口，输入 adb shell 进入 Android 的 shell 模式

```
adb shell
```

设置 adb 的 tcp 端口为 5555，重启 Android 设备上的 adbd 进程，让设置生效

```
setprop service.adb.tcp.port 5555
```

```
su root
```

```
stop adbd && start adbd
```

查看设备的 IP 地址

设置->网络和互联网->Ethernet->IP address

通过 IP 地址连接到设备

```
adb connect device_ip_address
```

5.3 常用命令

查询设备

```
adb devices
```

停止 adb 服务器

```
adb kill-server
```

安装应用

```
adb install path_to_apk
```

将文件复制到设备

```
adb push local remote
```

从设备复制文件

```
adb pull remote local
```

发出 shell 命令

```
adb [-d | -e | -s serial_number] shell shell_command
```



6 注意事项与建议

6 免责声明

本文档提供有关广州眺望电子科技有限公司产品的信息。本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。

除眺望电子在其产品的销售条款和条件中声明的责任之外，眺望电子不承担任何其它责任。并且，眺望电子对产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。

眺望电子产品并非设计用于医疗、救生或维生等用途。眺望电子可能随时对产品规格及产品描述做出修改，恕不另行通知。

在订购产品之前，请您与当地的广州眺望电子科技有限公司销售处或分销商联系，以获取最新的规格说明。广州眺望电子科技有限公司保留所有权利。