

X30 Android 平台

用户手册



深圳市九鼎创展科技有限公司
www.9tripod.com

版权声明

本手册版权归属深圳市九鼎创展科技有限公司所有，并保留一切权力。非经九鼎创展同意(书面形式)，任何单位及个人不得擅自摘录本手册部分或全部，违者我们将追究其法律责任。

敬告：

在售开发板的手册会经常更新，请在 <http://www.9tripod.com> 网站下载最新手册，不再另行通知。

版本说明

版本号	日期	作者	描述
Rev.01	2019-2-21	owen	原始版本

技术支持

如果您对文档有所疑问，您可以在办公时间（星期一至星期五上午 9:00~12:00；下午 1:30~6:00）通过拨打技术支持电话、E-mail、留言到 BBS 论坛（<http://bbs.9tripod.com>）。

网 址：www.9tripod.com

联系电话： 18565758976

E - mail：supports@9tripod.com

销售与服务网络

公司：深圳市九鼎创展科技有限公司

地址：深圳市宝安区留仙二路中粮商务公园 2 栋 1703A

电话：0755-33121205/18038105325

网址：<http://www.9tripod.com>

论坛：<http://bbs.9tripod.com>，<http://x.9tripod.com>

淘宝：<http://armeasy.taobao.com>

阿里：<http://armeasy.1688.com>

速卖通：www.aliexpress.com/store/2340163

技术交流 QQ 群	QQ 群号
x210/i210 一群	23831259
x210/i210 二群	211127570
x4412/ibox4412 论坛	16073601
x4412/x4418 论坛	211128231
X4418/ibox4418 论坛	199358213
x6818/ibox6818 论坛	189920370
九鼎 RK 平台 1 群	159144256
九鼎 RK 平台 2 群	573696929



热烈欢迎广大同仁扫描右侧九鼎创展官方公众微信号，关注有礼，您将优先得知九鼎创展最新动态！



目录

版权声明.....	II
版本说明.....	III
目录.....	5
第 1 章 android 开发平台的搭建	8
1.1 使用 U 盘安装 ubuntu.....	8
1.2 设置 XP 为开机默认启动.....	9
1.3 ubuntu 下安装源码比较工具.....	10
1.4 ubuntu 下安装串口终端 minicom.....	10
1.5 ubuntu 下安装串口终端 picocom.....	11
第 2 章 Android 开发工具	12
2.1 SDK.....	12
2.2 adb 工具.....	12
2.2.1 安装 adb 工具.....	12
2.2.2 查看设备的连接状态.....	12
2.2.3 进入 adb shell.....	13
2.3 串口工具 secureCRT.....	14
第 3 章 安装 android 源码包	16
3.1 安装 android 源码依赖包.....	16
3.2 安装交叉编译工具链.....	16
3.3 安装 64 位系统必要的一些补丁包.....	16
3.4 指定 GCC 交叉编译器.....	16
3.5 安装 android 源码包.....	17
第 4 章 编译 android 源码包	18
4.1 编译 uboot.....	18
4.2 编译 android 内核.....	18
4.3 编译 android 文件系统.....	18
4.4 查看编译帮助.....	18
第 5 章 烧写 android 映像文件	19
5.1 Windows 系统映像烧写.....	19
5.1.1 RKTool 驱动安装.....	19
5.1.2 烧录方法一:单个升级固件 update-android.img.....	19
5.1.3 烧录方法二:多设备升级固件 update.img.....	21
5.2 Linux 系统烧写映像.....	22
5.2.1 生成固件 update.img.....	22
5.2.2 烧录固件 update-android.img.....	22
5.2.3 使用 upgrade_tool 指令烧写映像.....	24
5.2.4 使用 Rkflashkit 烧写映像.....	24
5.3 使用 TF 卡升级固件.....	25
5.3.1 功能说明.....	25
5.3.2 软件说明.....	26
5.3.3 制作前软件配置.....	26



5.3.4	制作.....	27
第 6 章	android 开发指南	28
6.1	命令终端.....	28
6.2	播放 mp3.....	28
6.3	播放视频.....	29
6.4	使用 WIFI 上网	31
6.5	使用蓝牙传输数据.....	33
6.6	使用蓝牙播放音乐.....	36
6.7	使用 USB 鼠标键盘.....	37
6.8	挂载 TF 卡	37
6.9	挂载 U 盘.....	37
6.10	屏幕旋转.....	37
6.11	拍照照相.....	38
6.12	使用有线以太网上网.....	38
6.13	使用遥控器操作开发板.....	39
6.14	开关机.....	39
6.15	休眠唤醒.....	39
第 7 章	Android 测试程序	40
7.1	液晶屏测试.....	40
7.2	触摸屏测试.....	40
7.3	发光二极管测试.....	41
7.4	蜂鸣器测试.....	41
7.5	背光测试.....	42
7.6	按键测试.....	42
7.7	电池测试.....	43
7.8	数模转换测试.....	43
7.9	重力传感器测试.....	44
7.10	音频测试.....	44
7.11	摄像头测试.....	45
7.12	无线网络测试.....	45
7.13	网络连接测试.....	46
7.14	串口测试.....	46
7.15	外部存储器测试.....	47
7.16	优盘测试.....	47
第 8 章	android 内核驱动	48
8.1	G-sensor 驱动	48
8.2	电容触摸屏驱动.....	48
8.3	液晶屏驱动.....	48
8.4	按键驱动.....	48
8.5	WIFI/BT 模块驱动.....	48
8.6	摄像头驱动.....	48
8.7	proc 文件系统.....	48
8.7.1	启动环境变量查询.....	48



8.7.2	CPU 信息查询.....	48
8.7.3	内存信息查询.....	49
8.7.4	磁盘分区信息查询.....	50
8.7.5	内核版本查询.....	51
8.7.6	网络设备查询.....	52
8.7.7	查看内核启动信息.....	52
第 9 章	android 开发项目实战.....	53
9.1	实战 1: 如何修改 init.rc 文件.....	53
9.1.1	永久修改.....	53
9.1.2	临时修改.....	53
9.2	实战 2: 如何修改开机动画.....	55
9.3	实战 3: 如何修改 uboot 中的开机 LOGO.....	55
9.4	实战 4: 如何修改内核中的 LOGO.....	55
9.5	配置使用 MIPI 屏.....	55
第 10 章	其他产品介绍.....	56
10.1	核心板系列.....	56
10.2	开发板系列.....	56
10.3	卡片电脑系列.....	56



第1章 android 开发平台的搭建

Android 系统编译整套源码包对 PC 机硬件要求很高，请直接安装 Linux 操作系统，充分发挥 PC 机的性能，不要企图通过虚拟机顺利开发。我们这里以 ubuntu14.04 64 位系统机器为例讲解，如果您是新手，建议与我们版本保持一致。

1.1 使用 U 盘安装 ubuntu

使用 U 盘安装 ubuntu 系统简单快捷，强烈推荐使用此方法安装。

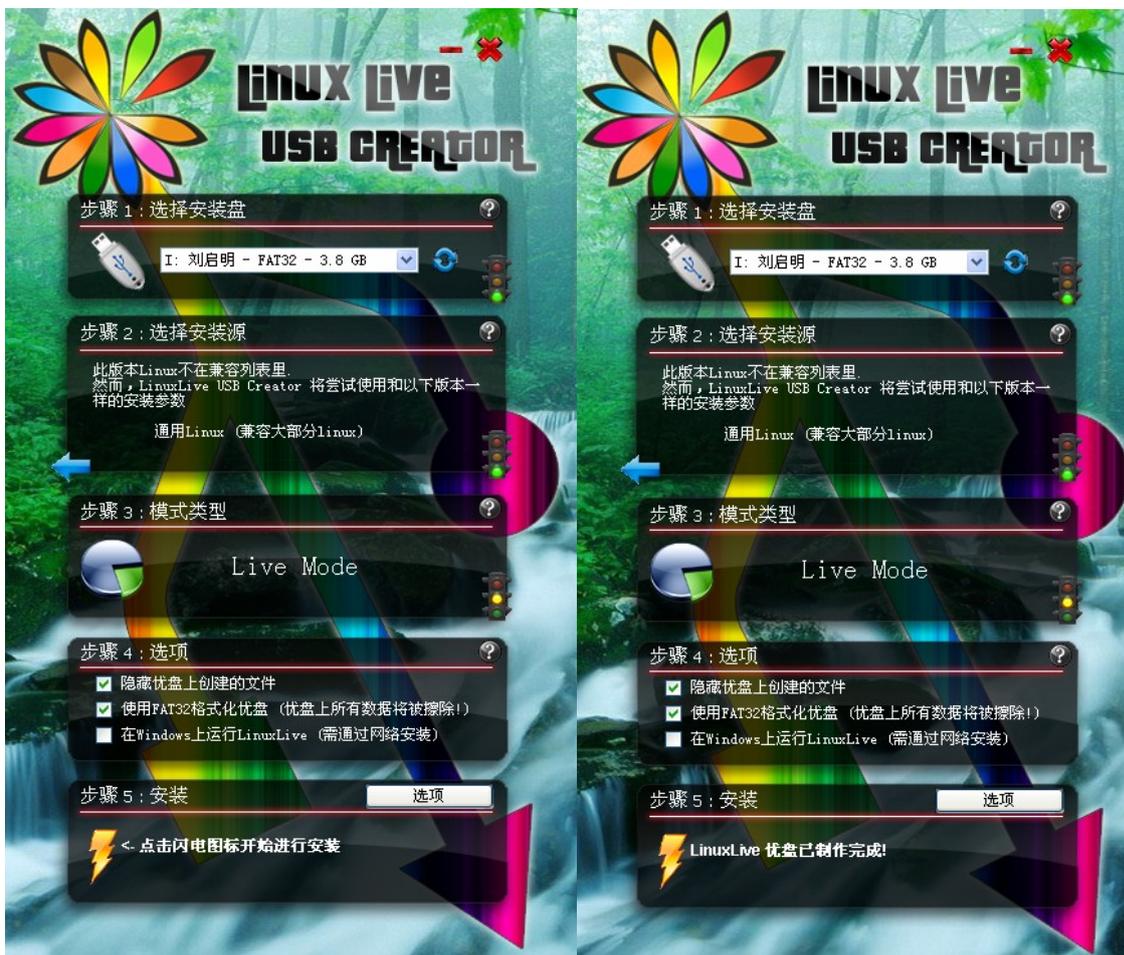
安装工具：

- 2G 以上 U 盘一个
- lili usb creator 软件，下载地址：<http://www.linuxliveusb.com/>
- ubuntu 最新系统，下载地址：<http://www.ubuntu.com/download/>
- PC 机一台

安装方法：

第一步：下载好 ubuntu 的 ISO 文件，和 lili usb creator 这个软件并安装。

第二步：插入 usb，并打开 usb creator 这个软件，根据软件提示设置，在步骤 1 中选择安装盘，找到识别出的 U 盘；在步骤 2 中找到下载的 ubuntu 映像文件；步骤 3 默认，步骤 4 选中隐藏优盘上创建的文件，使用 FAT32 格式化 U 盘；最后在步骤 5 中点击闪电图标开始安装，直到提示优盘已安装完成为止。





第三步：重启电脑，开机时，看清屏幕下方的提示，进入 BIOS 设置菜单，选择 U 盘启动。一般台式机是按 DEL 键，笔记本有些是按 F2，有些按 F10 进入。设置完成后保存退出。

第四步：再次重启系统，这时已经可以看到 ubuntu 的安装界面了，选择中文，继续；

第五步：选择 install（您也可以选择 live mode 可以体验下系统），继续；

第六步：也选择中文，点击继续：再继续；

第七步：配置网络，可以安装时升级，也可以不升级，等安装完系统后再手动升级；

第八步：第一个选项是把以前的系统升级到 ubuntu14.04，如果你只要单系统的可以选择第一项，第二项是升级到 ubuntu14.04 并把其他的资料删除掉，第三项是我们用的最多的，比较灵活，选 something else，继续，在这里我分出了两个区给 ubuntu，一个 / 和一个 /home，分区可以新建，可以对它格式化，具体根据需要选择；

第九步：这是设置区域，选择上海就行了；

第十步：选择键盘布局。选中国；

第十一步：选择用户名和密码，到此配置完毕，点击继续直接安装，喝一会儿咖啡，待安装完毕，重启之后，就可以看到美丽的 ubuntu 世界了。

1.2 设置 XP 为开机默认启动

ubuntu 安装后每次开机都是默认进入 ubuntu 系统的，对于以 windows 为主的朋友，每次开机都要守在画面切换到 XP 启动，可见十分麻烦，通过下面，你可以设置让你的 XP 系统为第一启动。

修改/boot/grub/grub.cfg 文件，可以看到最末尾有如下语句：

```
menuentry "Microsoft Windows XP Professional (on /dev/sda1)" {
    insmod part_msdos
    insmod ntfs
    set root='(hd0,msdos1)'
    search --no-floppy --fs-uuid --set 0E48A65048A6367D
    drivemap -s (hd0) ${root}
    chainloader +1
}
### END /etc/grub.d/30_os-prober ###
```

仔细阅读该文件，发现有不少 menuentry，这些正是对应了开机启动时的选择项，将上面语句放在第一个 menuentry 的前面，这样 Windows XP 就为默认的启动系统了。修改后的部分代码如下：

```
### BEGIN /etc/grub.d/05_debian_theme ###
set menu_color_normal=white/black
set menu_color_highlight=black/light-gray
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/30_os-prober ###
menuentry "Microsoft Windows XP Professional (on /dev/sda1)" {
    insmod part_msdos
    insmod ntfs
    set root='(hd0,msdos1)'
    search --no-floppy --fs-uuid --set 0E48A65048A6367D
```



```
drivemap -s (hd0) ${root}
chainloader +1
}
### END /etc/grub.d/30_os-prober ###

### BEGIN /etc/grub.d/10_linux ###
menuentry 'Ubuntu, with Linux 2.6.35-30-generic' --class ubuntu --class gnu-linux --class gnu
--class os {
    recordfail
    insmod part_msdos
    insmod ext2
    set root='(hd0,msdos9)'
    search --no-floppy --fs-uuid --set 4833f619-6388-4dd3-acd7-6fa3eacb9a15
    linux    /boot/vmlinuz-2.6.35-30-generic
    root=UUID=4833f619-6388-4dd3-acd7-6fa3eacb9a15 ro    quiet splash
    initrd  /boot/initrd.img-2.6.35-30-generic
}
```

1.3 ubuntu 下安装源码比较工具

ubuntu 下源码比较工具很多, 比较常见的是 `meld` 工具, 使用如下指令安装:

```
sudo apt-get install meld
```

安装完成后, 可以在应用程序->编程中打开。

1.4 ubuntu 下安装串口终端 minicom

使用如下指令安装:

```
sudo apt-get install minicom
```

安装完成后, 需要设置 `minicom`。如果直接使用串口, 通常设置为 `ttyS0`, 如果使用 USB 转串口, 通常设置为 `ttyUSB0`。输入如下指令:

```
sudo minicom -s
```

选择 Serial port setup, 选择 A, 输入正确的串口终端, 选择 E, 输入 115200 8N1, 选择 F 和 G, 都设置为 No, 不使用流控, 再回车, 选择 Save setup as dfl。注意, 只有 root 用户才有权限保存参数。笔记本用户通常使用的 USB 转串口延长线, 目前市面上大多都是 `pl2303` 方案, 插上 USB 转串口延长线后, 输入如下命令查询驱动是否正常加载:

```
lsmod |grep pl2303
```

正常加载时会提示如下信息:

```
lqm@lqm:~$ lsmod |grep pl2303
pl2303                11756  1
usbserial             33100  3 pl2303
```

再使用如下命令查询系统的一些信息:

```
dmesg | tail -f
```

正常情况下会出现如下提示:

```
lqm@lqm:~$ dmesg |tail -f
[ 383.093851] ERROR! H2M_MAILBOX still hold by MCU. command fail
```



```
[ 383.148849] ---> RTMPFreeTxRxRingMemory
[ 383.148903] <--- RTMPFreeTxRxRingMemory
[ 383.180580] RTUSB disconnect successfully
[ 387.762330] usb 2-4: USB disconnect, address 3
[ 387.762566] pl2303 ttyUSB0: pl2303 converter now disconnected from ttyUSB0
[ 387.762601] pl2303 2-4:1.0: device disconnected
[ 392.164589] usb 2-4: new full speed USB device using ohci_hcd and address 5
[ 392.379898] pl2303 2-4:1.0: pl2303 converter detected
[ 392.412998] usb 2-4: pl2303 converter now attached to ttyUSB0
```

表示串口设备名称为 ttyUSB0。有时会提示如下错误：

```
lqm@lqm:~$ dmesg | tail -f
[ 408.910351] 0x1300 = 00073200
[ 413.945752] ==>rt_ioctl_giwscan. 8(8) BSS returned, data->length = 1177
[ 419.047006] ==>rt_ioctl_giwscan. 7(7) BSS returned, data->length = 1067
[ 419.047302] ==>rt_ioctl_siwfreq::SIOCSIWFREQ[cmd=0x8b04] (Channel=1)
[ 419.392535] wlan0: no IPv6 routers present
[ 433.902136] ==>rt_ioctl_giwscan. 6(6) BSS returned, data->length = 960
[ 473.902907] ==>rt_ioctl_giwscan. 8(8) BSS returned, data->length = 1271
[ 533.900777] ==>rt_ioctl_giwscan. 8(8) BSS returned, data->length = 1200
[ 613.904091] ==>rt_ioctl_giwscan. 9(9) BSS returned, data->length = 1435
[ 713.904199] ==>rt_ioctl_giwscan. 8(8) BSS returned, data->length = 1263
```

一般情况下重插拔一次 USB 转串口线即可。

1.5 ubuntu 下安装串口终端 picocom

picocom 是一款 linux 下的串口调试工具，它比 minicom 精巧的多，在使用 minicom 调试时，有时我们会手动清除之前的调试信息，这时 minicom 再次接收串口调试数据时，不会再从顶端显示打印信息，而是从最末一行，程序员看起来会很别扭。而 picocom 并不会出现这种 BUG。使用如下指令安装 picocom：

```
sudo apt-get install picocom
```

安装完成后，执行如下指令打开 picocom：

```
sudo picocom -b 115200 /dev/ttyUSB0
```

这里 115200 表示波特率，/dev/ttyUSB0 表示 PC 机上的调试串口的设备节点，用户根据实际情况进行调整。



第2章 Android 开发工具

2.1 SDK

谷歌公司开发的 Android Studio 提供用于为各类 Android 设备开发应用的最快速的工具。利用世界一流的代码编辑、调试、性能工具、一套灵活的构建系统以及一套即时构建/部署系统，您可以专注于构建独特的高品质应用。

网上下载最新的 SDK，下载地址为：<http://developer.android.com/sdk/index.html>。用户可根据自身的 PC 机系统选择对应的安装包，如下图所示：

平台	Android Studio 软件包	大小	SHA-1 校验和
Windows (64 位)	android-studio-bundle-162.3871768-windows.exe 包含 Android SDK (推荐)	1,876 MB (1,968,176,480 bytes)	8cfa10645b7fe1a89d4c454533763bfa34be830f4c4a5adc42afa363e0492150
	android-studio-ide-162.3871768-windows.exe 无 Android SDK	412 MB (433,012,472 bytes)	95ca44467d399e609e86bf874eba00f8f2e6e371ae294b7f1e88cfc8689e14dd
	android-studio-ide-162.3871768-windows.zip 无 Android SDK, 无安装程序	429 MB (450,490,546 bytes)	96d4cec9d7b97a451af0250de4eaad29031fc62e97c4368b370e0736e82e274d
Windows (32 位)	android-studio-ide-162.3871768-windows32.zip 无 Android SDK, 无安装程序	429 MB (449,931,461 bytes)	ad0cd9630b148e3848d4381d2b8898f87148ae0574e561a8a5559acb0cbc3c63
Mac	android-studio-ide-162.3871768-mac.dmg	425 MB (445,810,938 bytes)	f8a414f7f4111a9aba059c7b85a3f0aba6abc950552a270042daa488922db377
Linux	android-studio-ide-162.3871768-linux.zip	429 MB (450,391,500 bytes)	36520f21678f80298b5df5fe5956db17a5984576f895fdcaa36ab0dbfb408433

2.2 adb 工具

2.2.1 安装 adb 工具

在光盘中找到 fastboot_adb.rar 文件，解压出来，将 fastboot 文件夹下面的 adb.exe、AdbWinApi.dll、AdbWinUsbApi.dll 以及 fastboot.exe 四个文件拷贝到 C:\Windows\System32 目录，如果是 64 位系统，需要同时拷贝到 C:\Windows\SysWOW64 目录，adb 和 fastboot 即安装成功。

2.2.2 查看设备的连接状态

开发板上电，进入 setting，选择“developer options”，勾选“USB debugging”进入 WINDOWS 下的 CMD 命令行，输入如下命令验证开发板是否连接：

```
adb devices
```

显示下面内容表示成功连接：



```
ca. 管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>d:

D:\>cd fastboot

D:\fastboot>adb devices
List of devices attached
0123456789abcdef      device

D:\fastboot>_
```

2.2.3 进入 adb shell

使用如下命令进入开发板终端：

```
adb shell
```

如下图所示：

```
ca. 管理员: C:\Windows\system32\cmd.exe - adb shell

C:\Users\Administrator>d:

D:\>cd fastboot

D:\fastboot>adb devices
List of devices attached
0123456789abcdef      device

D:\fastboot>adb shell
shell@drone2:/ $ ls
ls
acct
cache
config
d
data
default.prop
dev
etc
file_contexts
fstab.drone2
init
init.drone2.rc
```

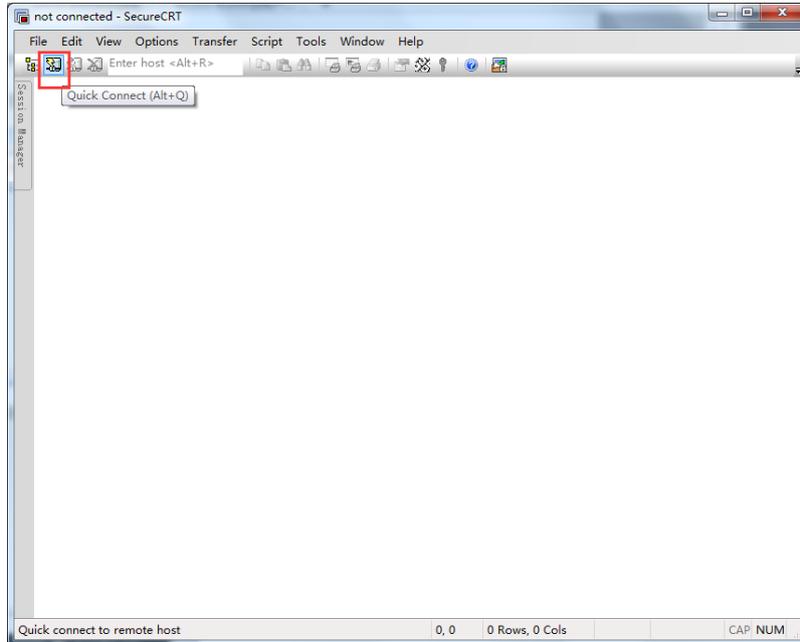
输入 exit 退回 DOS 操作界面。注意，有时候，执行 adb devices 命令时，会提示 error: more than one device and emulator，很有可能是播放了 USB 设备造成的。这时已经无法再通过 adb



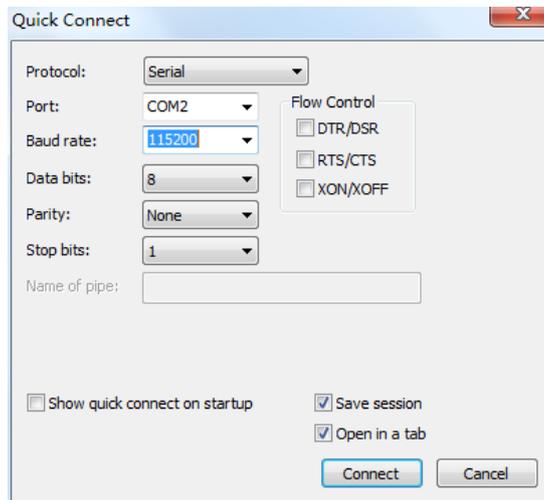
传输数据，解决的办法很简单，如果是使用 windows，直接在进程中干掉 adb.exe，再启动 adb 即可。

2.3 串口工具 secureCRT

安装 secureCRT 工具，在计算机设备管理器中找到 COM 口号，点击“快速连接”，如下图：



协议选择 serial（我的设备是 COM2），其他选项按照下图所示选择，**右侧三个流控制选项不能勾选！**



接下来用 USB 转串口线接到开发板的调试串口与 PC 即可看到调试串口打印的 log 信息。



```
Serial-COM4 - SecureCRT
File Edit View Options Transfer Script Tools Window Help
Enter host <Alt+R>
Serial-COM4 x
Channel a: DDR3 200MHz
Bus width=32 Col=10 Bank=8 Row=15 CS=1 Die Bus-width=16 Size=1024MB
Channel b: DDR3 200MHz
Bus width=32 Col=10 Bank=8 Row=15 CS=1 Die Bus-width=16 Size=1024MB
Memory OK
Memory OK
OUT

U-Boot 2014.10-RK3288-10-g45ba757 (Feb 19 2016 - 17:53:13)

CPU: rk3288
CPU's clock information:
  arm p11 = 600000000HZ
  periph p11 = 297000000HZ
  ddr p11 = 200000000HZ
  codec p11 = 384000000HZ
Board: Rockchip platform Board
DRAM:  Found dram banks:1
Adding bank:0000000000000000(0000000080000000)
128 MiB
storage init OK!
Using default environment

GetParam
check parameter success
failed to prepare fdt from boot!
no power key node
```



第3章 安装 android 源码包

3.1 安装 android 源码依赖包

说明：本文档所有开发全部基于 ubuntu14.04 64 位系统，后续不再声明。

使用如下命令安装所需的软件包（安装前使用 `sudo apt-get update` 获得一下最近的软件包的列表）：

```
sudo apt-get install git-core gnupg flex bison gperf libssl-dev libwxgtk2.8-dev build-essential zip curl libncurses5-dev zlib1g-dev genromfs u-boot-tools libxml2-utils texinfo mercurial subversion whois
```

px30 android8 编译需要 jdk8，软件包可以从 sun 官网下载，也可以使用网盘提供的压缩包（网盘资料 DVD_X30/tools/JDK/）`java-8-openjdk-amd64.tar.bz2` 文件，将此文件下载后复制到 PC Ubuntu 系统下，解压到 `/usr/lib/jvm/` 目录下：

```
sudo tar xjf java-8-openjdk-amd64.tar.bz2 -C /usr/lib/jvm
```

注意：这样安装了 jdk8 后不会覆盖 PC Ubuntu 系统原来使用的 jdk，当 px30 android8 编译需要使用时键入命令：

```
export PATH=/usr/lib/jvm/java-8-openjdk-amd64/bin:$PATH
```

另，Android8 编译时为了操作方便，统一使用编译脚本 `mk.sh`（编译脚本一般放在整套 SDK 源码的根目录，命名为 `mk.sh` 或 `mk`）来编译，以上命令可以在编译脚本看到。

3.2 安装交叉编译工具链

我们已经将交叉编译工具链集成到源码包中，无需再手动安装。交叉编译工具链路径：

```
prebuilts/gcc/linux-x86/aarch64/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu/  
prebuilts/gcc/linux-x86/aarch64/aarch64-linux-android-4.9/
```

注意：如果需要详细获取具体使用哪种交叉编译器，可以进入文件 `kernel/Makefile` 文件，搜索关键词 `CROSS_COMPILE`，在 `arm64` 的那项可发现。

3.3 安装 64 位系统必要的一些补丁包

```
apt-get install lsb-core libc6-dev-i386 g++-multilib lib32z1-dev lib32ncurses5-dev
```

3.4 指定 GCC 交叉编译器

注：本节是其他代码编译需要准备的环境，不适用 PX30 Android8 环境，仅供参考。

在 ubuntu 系统上安装最新的 GCC 交叉编译器时，版本已经超过 4.4 了，使用如下指令可查询 GCC 的版本：

```
gcc -version
```

可能出现的界面如下：

```
terry@ubuntu-server:~$ gcc --version  
gcc (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.3  
Copyright © 2011 Free Software Foundation, Inc.  
terry@ubuntu-server:~$
```

上面显示 4.6.3 版本，默认我们给出的包，在 4.6.3 上编译会提示一些错误，都是新的 GCC 引出的错误，网上都有解决办法，如果不想修改这些错误，可将 GCC 版本降至 4.4 即可。解决办法：



```
sudo apt-get install gcc-4.4 g++-4.4 g++-4.4-multilib
```

装完后，开始降级 gcc ,这不会影响系统，仅仅是改个链接而已，不喜欢的话改回来即可。

```
cd /usr/bin
sudo mv gcc gcc.bk
sudo ln -s gcc-4.4 gcc
sudo mv g++ g++.bk
sudo ln -s g++-4.4 g++
```

3.5 安装 android 源码包

PX30 Android8 源码包获取方式有：从网盘获取，从 gitlab 获取。

方法 1：从网盘获取

从网盘 DVD_X30/source/目录下获取 Android8.1 的源码包 px30_oreo.tar.bz2，将其复制到本地 PC Ubuntu 系统用户目录下，解压：

```
tar xjf px30_oreo.tar.bz2
```

压缩包解压完成后在 px30_oreo 目录下只有一个目录.git，需要下面的操作才可以获取源码：

```
cd px30_oreo
git checkout .
```

稍等片刻待此指令执行完成后即可获取源码。

如果需要更新源码，首先确定是否包含 http://gitlab.com/9tripod/px30_oreo.git 这一项：

```
$ git config --list
```

```
remote.origin.url=git@192.168.0.199:9tripod/rockchip/px30_oreo.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
```

如果不存在，请执行以下指令添加：

```
git remote add gitlab http://gitlab.com/9tripod/px30_oreo.git
```

添加后再次执行 git config --list 即可看到以上内容，然后更新：

```
git pull gitlab
```

方法 2：从 gitlab 获取代码

进入普通用户某目录下，键入指令：

```
git clone http://gitlab.com/9tripod/px30_oreo.git
```

等待下载完成后，可在当前目录下获取 px30_oreo，进入目录即可获取源码。

如果因为网络问题总是中断，可从网盘下载初始包（见上面方法 1），同样获取到源码，然后更新（注意更新源操作同方法 1，添加同样使用 git remote add）：

```
git pull gitlab
```

说明：源码包名称可能会因发布日期等有所不同，具体以网盘中实际名称为准。



第4章 编译 android 源码包

说明：编译映像时一定要使用普通权限编译。编译生成的镜像统一放在 out/release 目录下，包括 MiniLoaderAll.bin、uboot.img、trust.img、resource.img、kernel.img、boot.img、misc.img、recovery.img、system.img、vendor.img、oem.img、update-android.img

4.1 编译 uboot

在 android 源码目录下执行如下命令编译 uboot，编译完成后映像文件会释放到 out/release 目录：MiniLoaderAll.bin、uboot.img、trust.img

```
./mk -u
```

4.2 编译 android 内核

在 android 源码目录下执行如下命令编译 android 内核，编译完成后映像文件 kernel.img、resource.img 会释放到 out/release 目录。

```
./mk -k
```

4.3 编译 android 文件系统

在 android 源码目录下执行如下命令编译 android 映像文件，编译完成后映像文件会释放到 out/release 目录。

```
./mk -s
```

编译完文件系统，在 out/release 目录下会生成如下文件：

- MiniLoaderAll.bin、uboot.img、trust.img： uboot 映像文件
- Kernel.img： 内核映像
- Resource.img： 资源映像，内含开机图片和内核的设备树信息。
- Boot.img： Android 的初始文件映像，负责初始化并加载 system 分区。
- System.img： android 的 system 分区映像，ext4 文件格式系统。
- Recovery.img： 急救模式映像。
- Misc.img： 分区映像，负责启动模式切换和急救模式的参数传递。
- oem.img： 用于给厂家使用，存放厂家的 APP 或数据。只读
- update-android.img： 整个升级文件的单一映像，包括了 uboot，内核，文件系统等。

4.4 查看编译帮助

执行如下指令可查询 mk 脚本使用方法：

```
./mk -h
```

“-u”表示编译 uboot

“-k”表示编译内核

“-s”表示编译 Android 文件系统

“-U”表示所有的镜像编译完成后执行打包程序（生成 update-android.img 镜像）

“-j=”表示多线程编译（比如-j=20）

“-a”表示所有的都执行。注意，“-a”等同于“./mk.sh -u -k -s -U”



第5章 烧写 android 映像文件

5.1 Windows 系统映像烧写

5.1.1 RKTool 驱动安装

解压 RKTools.rar，工具路径：SDK\RKTools\windows\Release_DriverAssitant, 打开“DriverInstall.exe”，点击“驱动安装”，提示安装驱动成功即可。



注意事项：

- 1.目前支持的操作系统包括：XP,Win7_32,Win7_64,Win8_32,Win8_64。
- 2.XP 系统在驱动安装完后，若还提示“发现新设备”，安装驱动时选择“自动安装”。
- 3.若之前已经安装过老版本驱动，请先点击“驱动卸载”后再进行“驱动安装”。

5.1.2 烧录方法一:单个升级固件 update-android.img

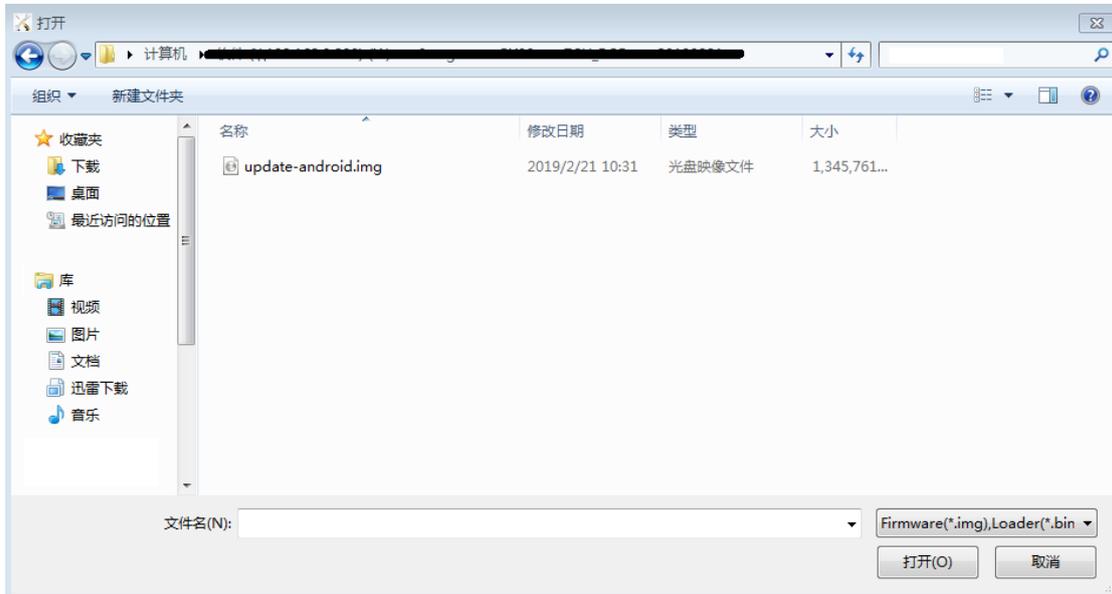
5.1.2.1 生成统一固件 update-android.img

我们已经在 mk 脚本中集成了生成统一固件的方法，把所有的镜像编译完成后执行-U 即可打包，释放到 out/release 目录。

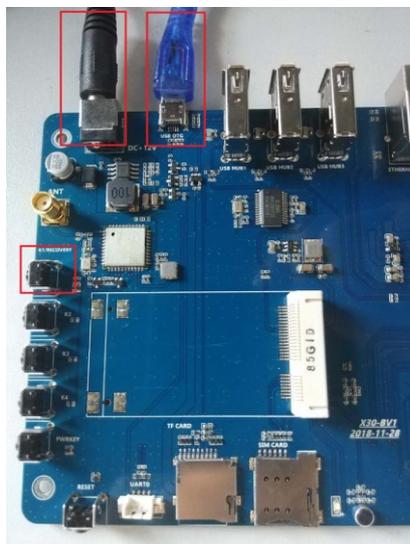
5.1.2.2 烧录固件 update-android.img

打开源码包目录下的

RKTools\windows\AndroidTool\AndroidTool_Release_v2.58\AndroidTool.exe,选择“升级固件”选项卡，点击“固件”，在弹出窗口中选择已经生成的 update.img 文件，如下图所示。

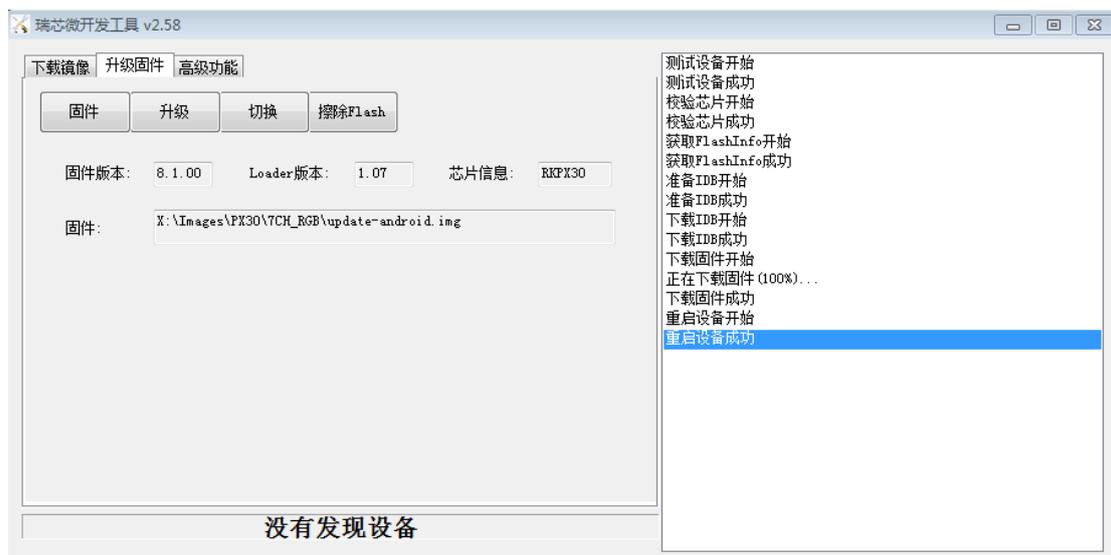
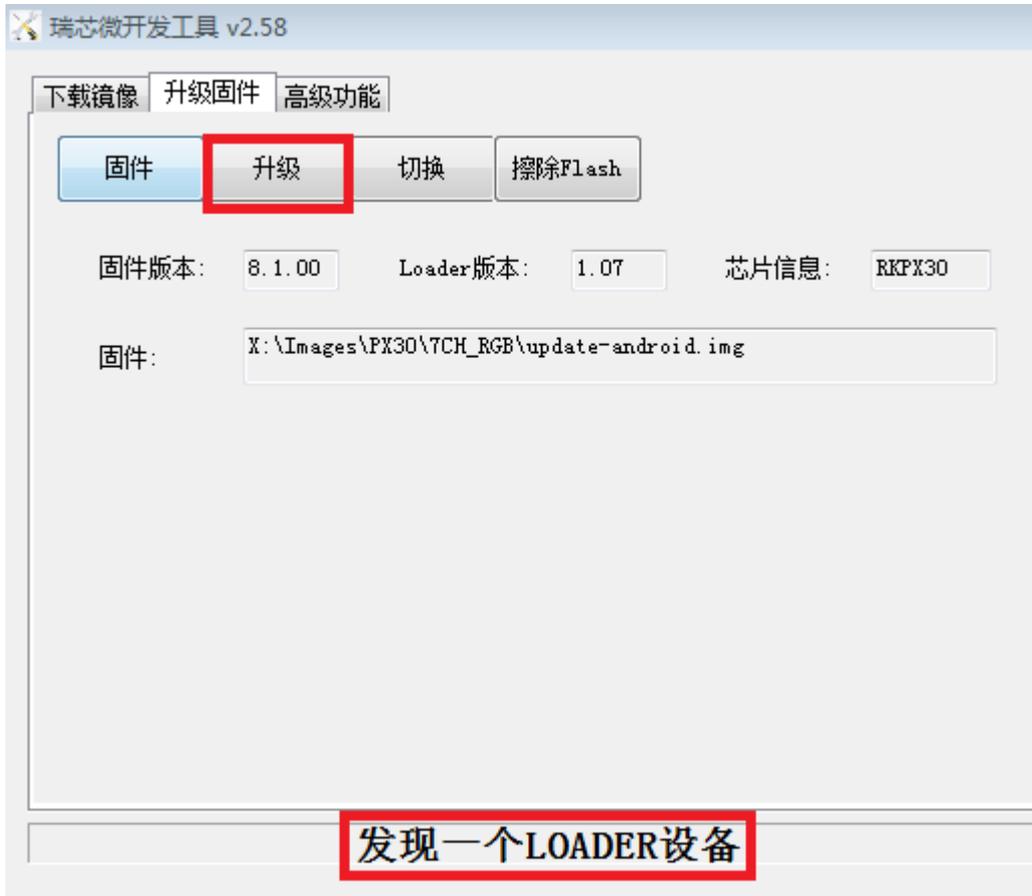


工具配置好后，连接开发板（如下图）：





按下 RECOVERY 键，然后插上 microUSB 线与 12V DC 电源线，烧录工具界面会提示发现一个 LOADER 设备，然后点击升级，即可开始升级过程（注：如果提示发现一个 ADB 设备，点击切换按钮切换成 LOADER 设备即可）。



上图为升级完成截图。

5.1.3 烧录方法二:多设备升级固件 update.img

该工具适合用户批量刷机，可以同时给多台开发板刷机。

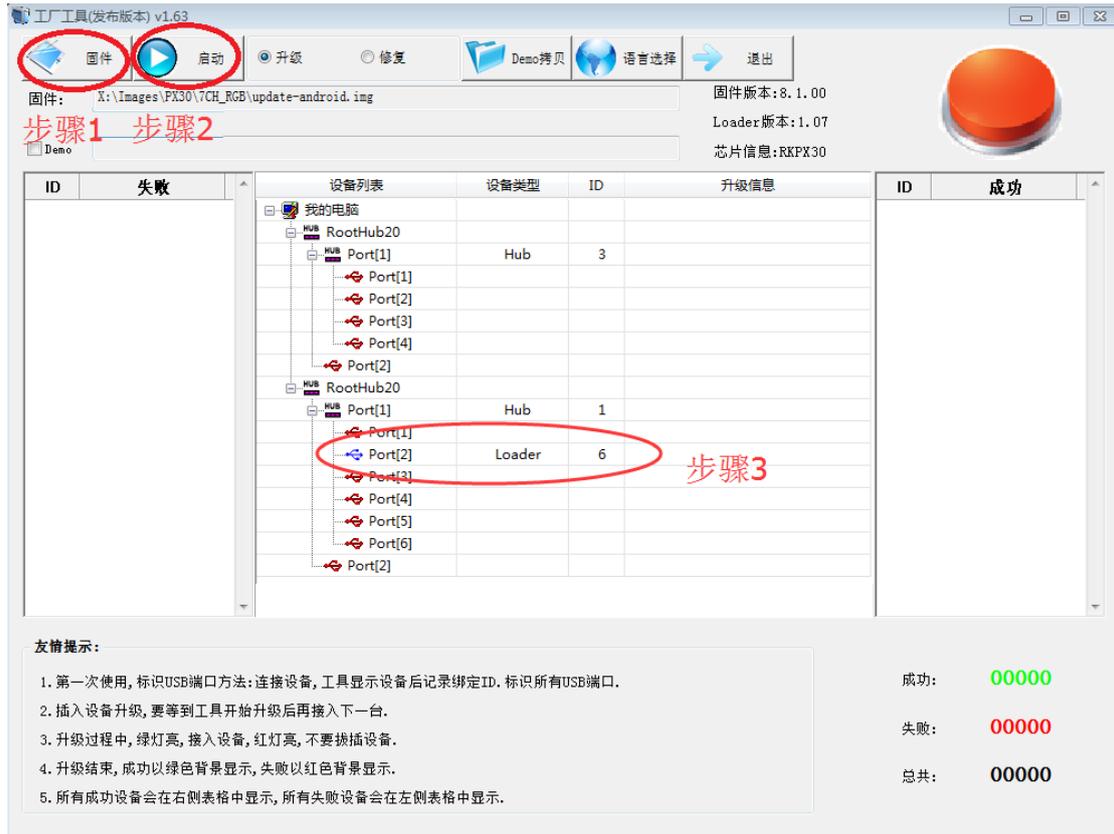


从代码目录下获取文件 RKTools\windows\FactoryTool_v1.63.zip，将其解压，打开“FactoryTool.exe”，点击“固件”选择 update-android.img，勾选“升级”，点击“启动”，如下图所示：

步骤 1：点击固件，选择 update-android.img;

步骤 2：点击启动(选择升级按钮);

步骤 3：连接开发板 USB、DC 电源，按下 recovery 键，对应 USB 口发现设备，并实现自动升级；然后重复步骤 3 即可同时升级第二台、第三台设备，升级成功或者失败的设备会在两边的列表中列出，移除成功或者失败的设备后可以继续连接需要升级的设备。



5.2 Linux 系统烧写映像

5.2.1 生成固件 update.img

我们已经在 mk 脚本中集成了生成统一固件的方法，在编译 android 文件系统时(./mk -s)，会自动生成 update.img，并释放到 out/release 目录。

5.2.2 烧录固件 update-android.img

工具路径：RKTools\linux\Linux_Upgrade_Tool_v1.2

在升级之前将 update.img 拷贝到 upgrade_tool 相同目录下，运行 upgrade_tool(需要 sudo)

```
work@ubuntu:~/3288/Linux_Upgrade_Tool_v1.2/cp rockdev/update.img .
```

```
work@ubuntu:~/3288/Linux_Upgrade_Tool_v1.2$ sudo ./upgrade_tool
```

执行结果如下图，发现设备列表，输入要升级的 DevNo（设备号）选择设备



```
ubuntu: ~/3288/Linux_Upgrade_Tool_v1.2
Download IDB Start
^Cwork@ubuntu:~/3288/Linux_Upgrade_Tool_v1.2$ sudo ./upgrade_tool
List of rockusb connected
DevNo=1 Vid=0x2207,Pid=0x320a,LocationID=128 Loader
Found 1 rockusb,Select input DevNo,Rescan press <R>,Quit press <Q>:1
```

选择设备后弹出工具使用菜单如下图，左侧是功能描述，右侧是命令语法，升级相关操作都在 upgrade command 列表下，忘记命令语法可以输入 H 进行查看，清屏输入 CS，退出按 Q。

```
ubuntu: ~/3288/Linux_Upgrade_Tool_v1.2
work@ubuntu:~/3288/Linux_Upgrade_Tool_v1.2$ sudo ./upgrade_tool
List of rockusb connected
DevNo=1 Vid=0x2207,Pid=0x320a,LocationID=105 Loader
Found 1 rockusb,Select input DevNo,Rescan press <R>,Quit press <Q>:1
-----Tool Usage -----
Help:          H
Quit:          Q
Version:       V
Clear Screen:  CS
-----Upgrade Command -----
ChooseDevice:  CD
SwitchDevice:  SD
UpgradeFirmware:  UF <Firmware>
UpgradeLoader:  UL <Loader>
DownloadImage:  DI <-p|-b|-k|-s|-r|-m image> [parameter file]
DownloadBoot:  DB <Loader>
EraseFlash:    EF <Loader|firmware>
LowerFormat:   LF
-----Professional Command -----
TestDevice:    TD
ResetDevice:   RD [subcode]
ResetPipe:     RP [pipe]
ReadFlashID:   RID
ReadFlashInfo: RFI
ReadChipInfo:  RCI
ReadSector:   RS <BeginSec> <SectorLen> [-decode] [File]
WriteSector:   WS <BeginSec> <File>
ReadLBA:       RL <BeginSec> <SectorLen> [File]
WriteLBA:      WL <BeginSec> <File>
EraseBlock:    EB <CS> <BeginBlock> <BlockLen> [--Force]
```

- CD 命令：选择设备，当执行的命令有包含设备重启操作时，需重新选择设备，当改变操作设备时需重新选择
- SD 命令：msc 切换到 rockusb 升级模式。当切换执行成功后，需要重新选择设备
- UF 命令：升级完整 update.img 固件，当执行成功后需要重新选择设备
- UL 命令：升级 loader 功能，当执行成功后需要重新选择设备
- DI 命令：下载单独 image 镜像到指定扇区，例如升级 kernel.img 或者 system.img 都可以直接使用此功能。例如下载 kernel.img：DI -k kernel.img parameter //如果之前通过 DI 下载过 parameter，则再下载 kernel.img 时就可以不用指定最后的 parameter 参数
- DB 命令：下载 boot，在 maskrom 状态下，可以通过此功能，让 maskrom 设备进行 Rockusb 协议通讯
- EF 命令：擦除整个 nandflash
- LF 命令：低格保留块后面区域，只有在 loader 模式下使用

执行 uf update.img 开始更新固件，下图为更新完成截图。

```
Rockusb>uf update.img
```



```
Rockusb>uf update.img
Loading firmware...
Support Type:RK32      FW Ver:4.4.02   FW Time:2016-02-20 15:28:14
Loader ver:2.19 Loader Time:2016-02-19 18:04:38
Upgrade firmware ok.
Rockusb>
```

备注：也可通过配置 config.ini 文件配置升级映像文件，只需输入 UF 即可升级，请用户自行尝试。

5.2.3 使用 upgrade_tool 指令烧写映像

上一节我们介绍了通过 upgrade_tool 烧写统一固件 update.img 的方法，熟悉三星平台的开发者会发现，这种方法并不是很高效，真正操作起来，它远没有 fastboot 工具来的迅速。其实，upgrade_tool 工具同样支持类似于 fastboot 的烧写方式。

为了烧写方便，在 mk 脚本中，默认在编译系统时，已经将烧写工具 upgrade_tool 拷贝到 out/release 目录了。

第一步：打开串口终端，并打开 minicom，用于适时监控串口调试信息；

第二步：按住 RECOVERY 键，连接 USB OTG 线和电源线，这时 uboot 打印信息将会提示已经进入 USB 下载模式。如果接通电源后没来得及按住 RECOVERY 键，在按住 RECOVERY 键的同时，再按下复位键即可。注意，使用

第二步：打开第二个串口终端，进入 out/release 目录；

第三步：在 out/release 目录下敲击如下指令，烧写相应的映像。

```
sudo upgrade_tool di -k kernel.img(烧写内核)
sudo upgrade_tool di -s system.img(烧写文件系统)
sudo upgrade_tool di resource resource.img(烧写资源文件)
sudo upgrade_tool di -r recovery.img(烧写急救文件)
sudo upgrade_tool ul RK3288UbootLoader_V2.30.10.bin(烧写 bootloader)
sudo upgrade_tool uf update.img(烧写统一固件)
```

5.2.4 使用 Rkflashkit 烧写映像

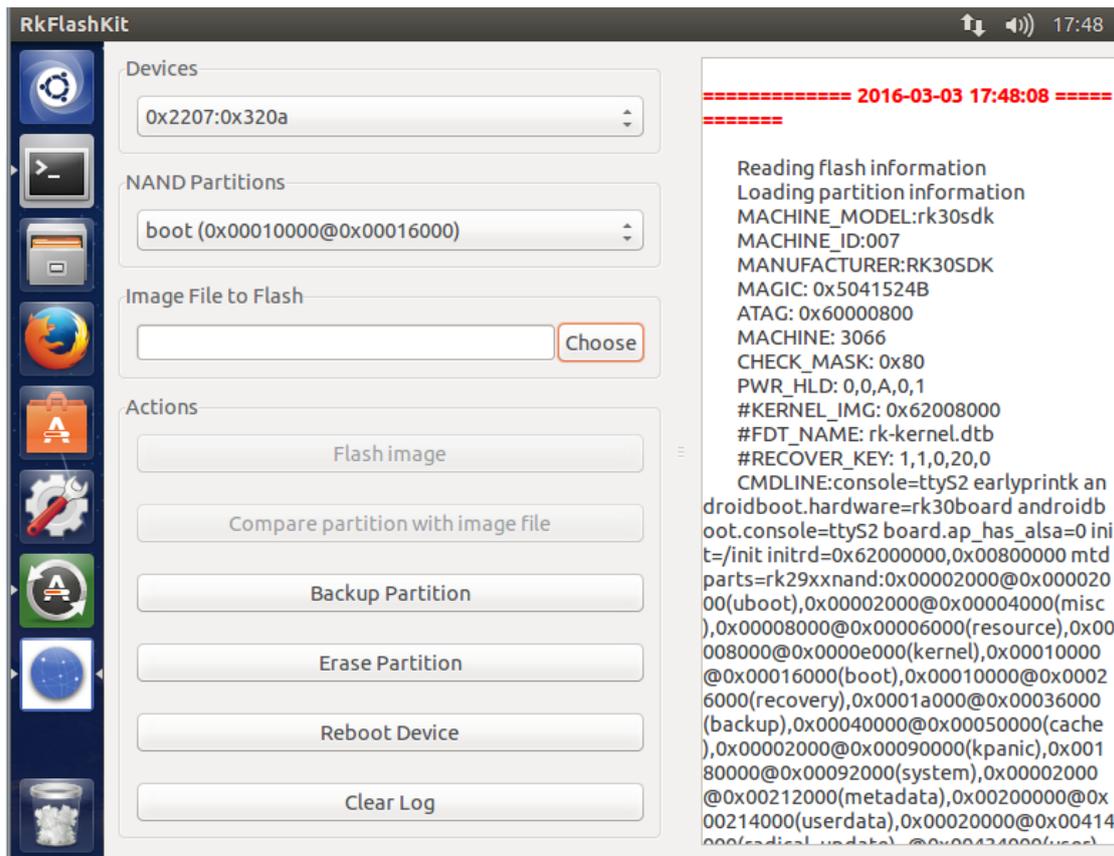
rkflashkit 有图形界面，后加了命令行支持，更是好用。

```
work@ubuntu:~/rktool$ sudo apt-get install build-essential fakeroot
work@ubuntu:~/rktool$ git clone https://github.com/linuxerwang/rkflashkit
work@ubuntu:~/rktool$ cd rkflashkit
work@ubuntu:~/rktool$ ./waf debian
work@ubuntu:~/rktool$ sudo apt-get install python-gtk2
work@ubuntu:~/rktool$ sudo dpkg -i rkflashkit_0.1.4_all.deb
```

注意：rkflashkit_0.1.4_all.deb 会因版本更新，版本数字可能会有所变化，如果执行失败，执行 ls 命令查看下即可。

```
work@ubuntu:~/rktool/$ sudo rkflashkit
```

如下是图形界面，在 Devices 下选择设备，选择要烧写的分区和对应的映像文件，点击 Flash image 即可。



该工具也支持命令行，使用 help 命令查看使用方法

```
work@ubuntu:~/rktool/rkflashkit$ rkflashkit --help
```

```
Usage: <cmd> [args] [<cmd> [args]...]
```

```
part                               List partition
flash @<PARTITION> <IMAGE FILE>   Flash partition with image file
cmp @<PARTITION> <IMAGE FILE>     Compare partition with image file
backup @<PARTITION> <IMAGE FILE>  Backup partition to image file
erase @<PARTITION>                 Erase partition
reboot                              Reboot device
```

For example, flash device with boot.img and kernel.img, then reboot:

```
sudo rkflashkit flash @boot boot.img @kernel.img kernel.img reboot
work@ubuntu:~/rktool/rkflashkit$
```

5.3 使用 TF 卡升级固件

5.3.1 功能说明

注：此功能等待后续验证，暂未实现，以下是其他平台的参考。

SD Card Boot 功能是主控在上电时，优先从 SD 卡上查找启动代码，然后加载引导，从而实现特定功能。目前 SD Card Boot 已经实现两个功能：SD 卡升级和 SD 卡运行。

SD 卡升级功能，将 SD 卡启动代码写到 SD 卡的保留区，然后将固件拷贝到 SD 卡可见

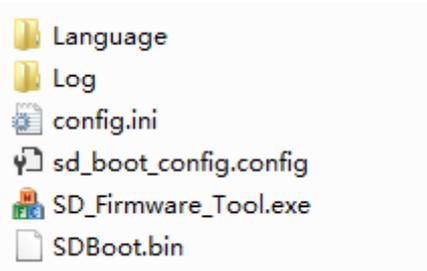


分区上，主控从 SD 卡启动时，SD 卡启动代码和升级代码将固件升级到本地主存储中，并支持 PCBA 测试和 Demo 文件的拷贝。可以完全做到脱离 PC 机进行固件升级，提高生产效率。

SD 卡运行功能，将固件升级到 SD 卡保留区中，把 SD 卡当作主存储使用。主控从 SD 卡启动时，固件已经存放在 SD 卡上，有没有本地主存储都可以正常工作。目前主要应用是板卡厂做 PCBA 测试，而不会破坏 flash 数据。

5.3.2 软件说明

工具目录有如下文件：



SD_Firmware_Tool.exe: 制卡工具

Config.ini: 配置文件

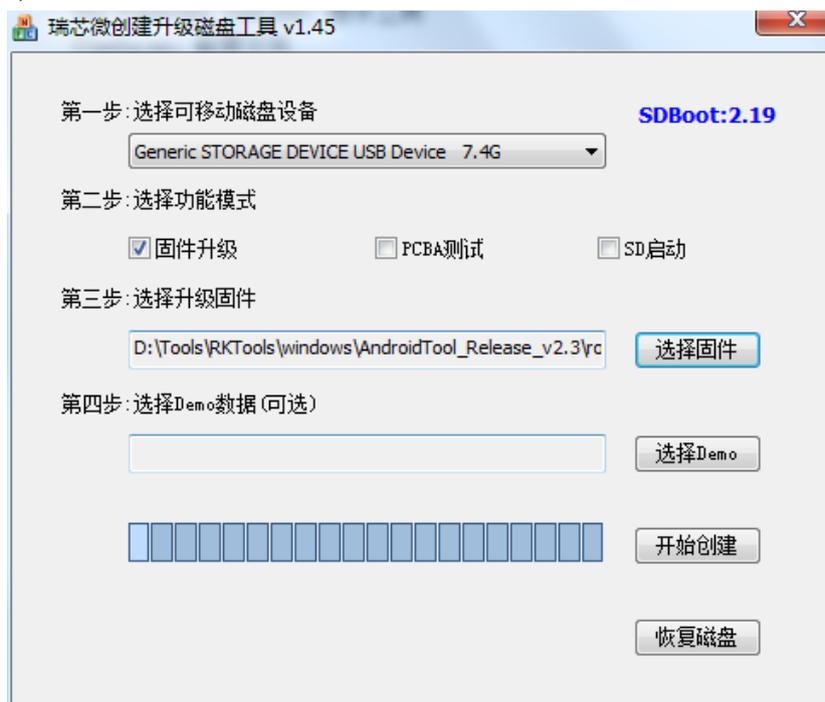
SDBoot.bin: SDRK2926、RK2928、RK3166 和 RK3188 使用 SDBoot.bin 支持 SD 卡升级和启动功能；**RK3288 使用 RK3288Loader_uboot_V2.16.bin 及以后版本。**

5.3.3 制作前软件配置

编辑 config.ini 配置文件以下项目设置为 TRUE

- #当值为 TRUE 时，新卡格式适用 3288 项目
- USE_FW_LOADER=TRUE

工具界面如下：





5.3.4 制作

注意：制作启动卡会格式化 SD 卡，用户需要注意备份重要数据，防止误删。

第一步，选择对应的可移动磁盘设备；

第二步，选择功能模式：固件升级、PCBA 测试、SD 启动；

- 默认会勾选“固件升级”，即只升级固件。
- 如果在升级固件前需要做“PCBA 测试”，则同时勾选“固件升级”、“PCBA 测试”。
- 如果不升级固件，只需要测试 PCBA，则单选“PCBA 测试”，不选择“固件升级”。
- “SD 启动”会把固件烧写到 SD 中，所有分区都分配在 SD 卡上，运行时不会修改内部存储的数据（相当于 SD 卡做主存储设备）。

注意：SD 卡运行模式，对 kernel 的编译有要求，需要把 kernel 的 SD 卡驱动配置去掉，不参与编译。

```
Now, there are two SDMMC controllers set
[ ] old driver (DEPRECATED)
[*] RK29 SDMMC0 controller support(sdmmc)
[ ] write-protect for SDMMC0
[ ] use the gpio-interrupt to detect card
[*] RK29 SDMMC1 controller support(sdio)
[ ] write-protect for SDMMC1
[ ] sdio-irq from gpio
[ ] Secure Digital Host Controller Interface support
```

修改为：

```
Now, there are two SDMMC controllers set
[ ] old driver (DEPRECATED)
[ ] RK29 SDMMC0 controller support(sdmmc)
[*] RK29 SDMMC1 controller support(sdio)
[ ] write-protect for SDMMC1
[ ] sdio-irq from gpio
[ ] Secure Digital Host Controller Interface support
```

第三步，选择对应平台的 update.img 固件；

第四步（可选），如果需要拷贝 demo 文件到用户盘根目录，点击选择 Demo 按钮，选择需要拷贝的文件目录。选择的目录下所有文件会拷贝到 SD 根目录下的 Demo 目录中，在 SD 引导固件升级后，Demo 目录下的文件会拷贝到样机用户盘的根目录下。

第五步，点击开始创建即可。



第6章 android 开发指南

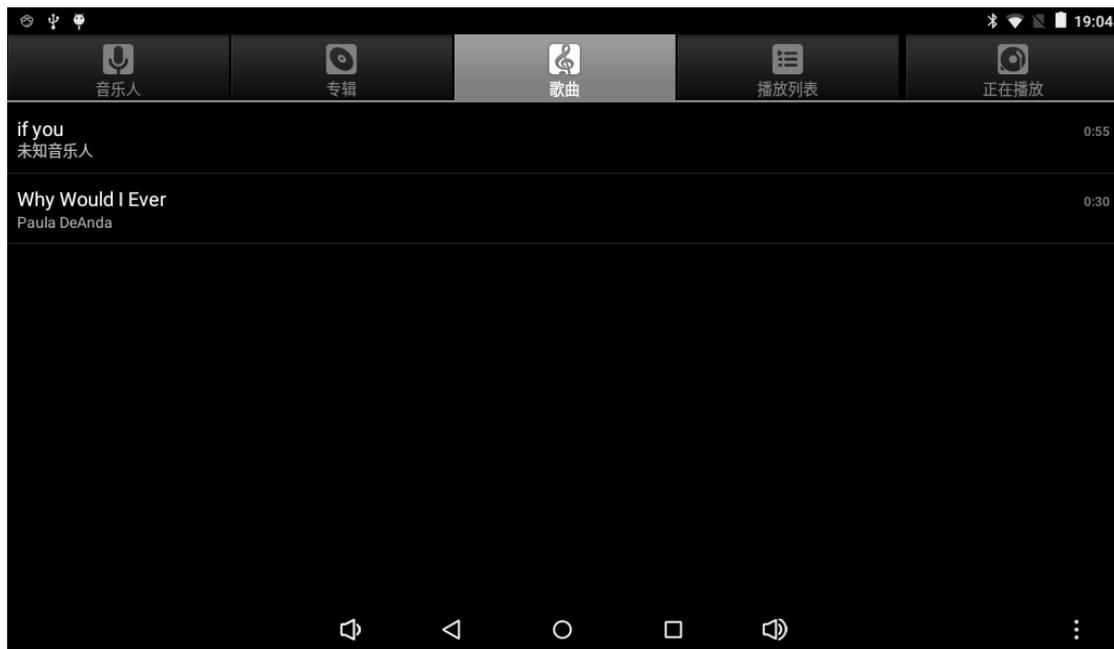
6.1 命令终端

将串口连接开发板上调试串口，进入 android 系统后，会自动进入 android 终端，如下图所示：

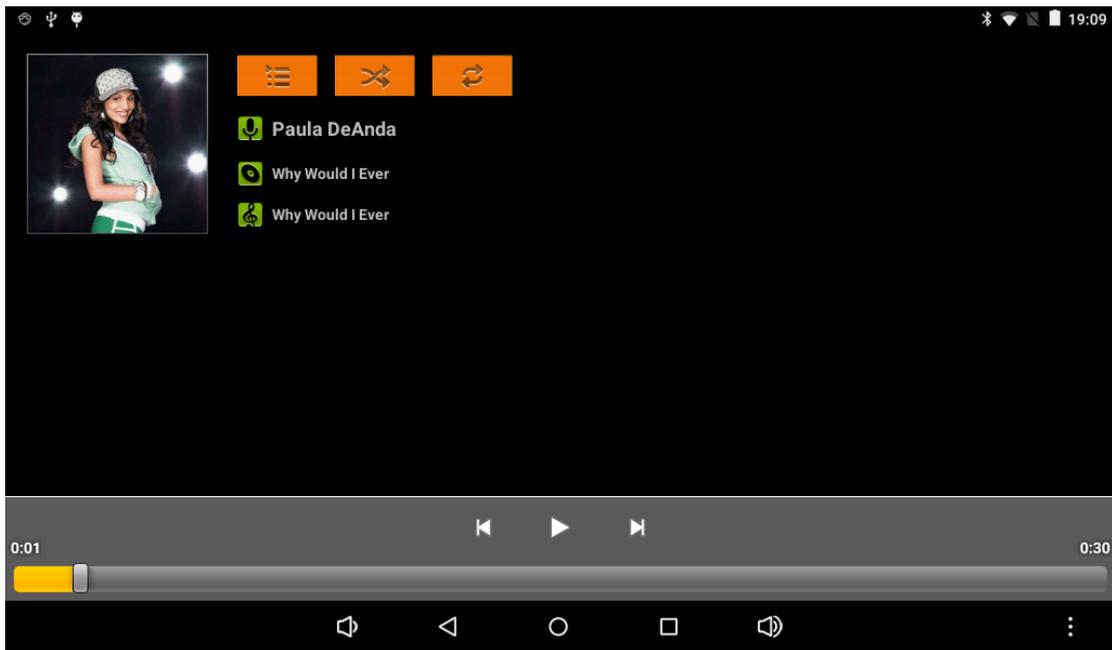
```
px30_evb: / $
```

6.2 播放 mp3

确保外置的 SD 卡中存在 mp3 文件，点击音乐，播放器会自动识别音频文件，如下图：

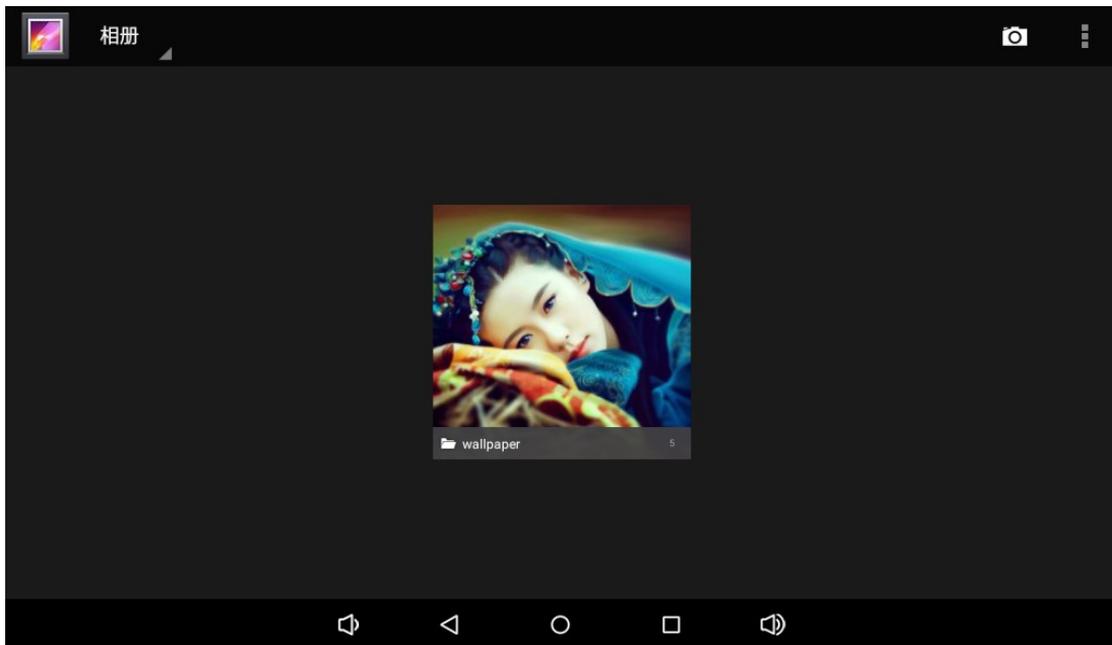


点击相应的音频文件即可播放。播放时界面如下：

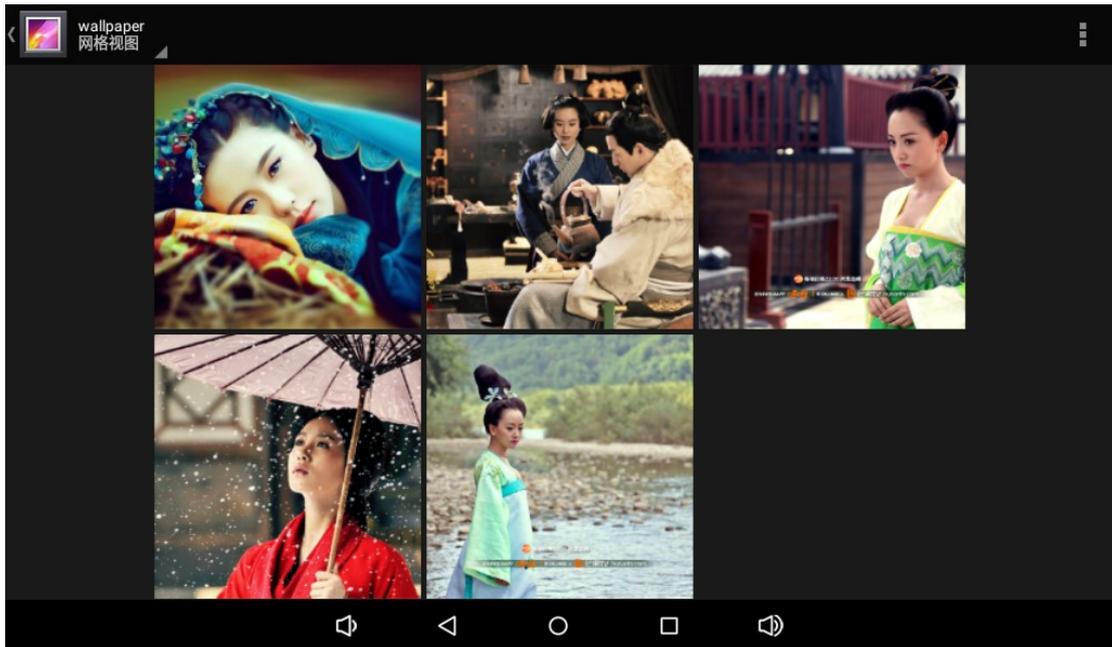


6.3 播放视频

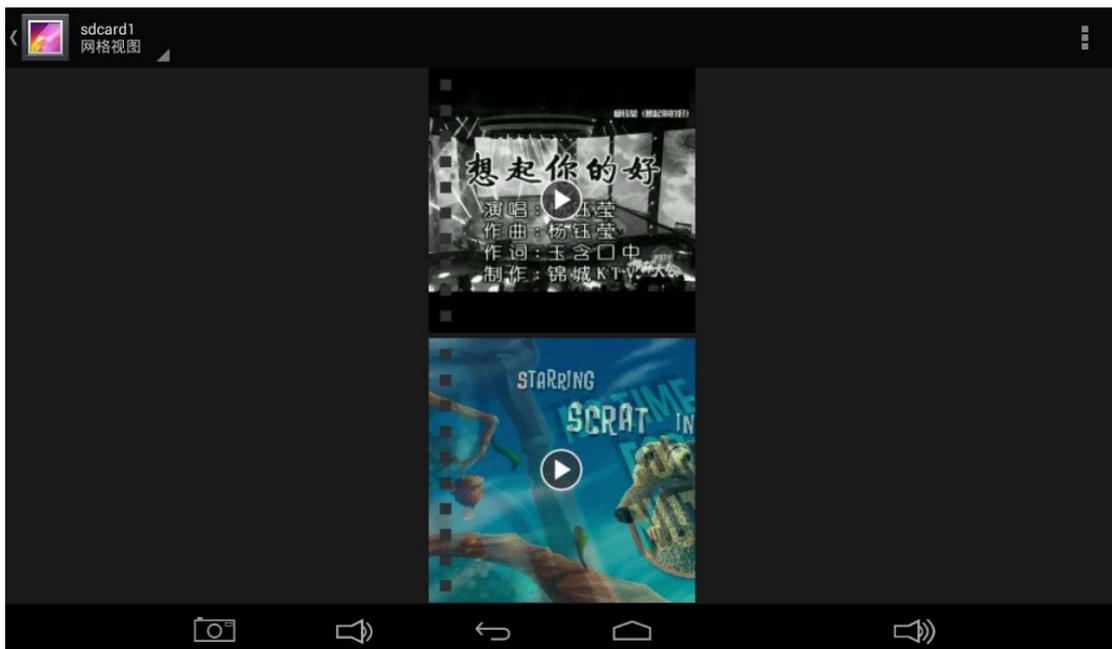
android 自带视频处理功能，在 android 应用界面显示为图库。点击图库按钮，会在外置 SD 卡中自动寻找能够识别的视频和图片文件，如下图：



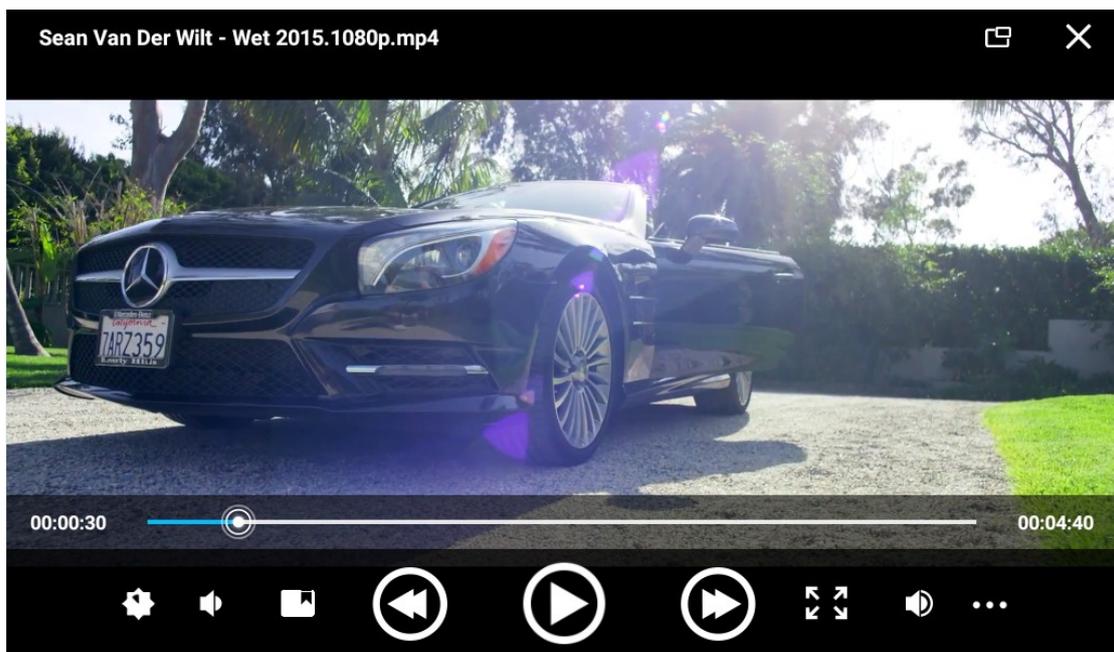
点击中间的按钮：



上图中，带有播放符号的即为视频文件，不带的为图片文件。点击带播放符号的文件：

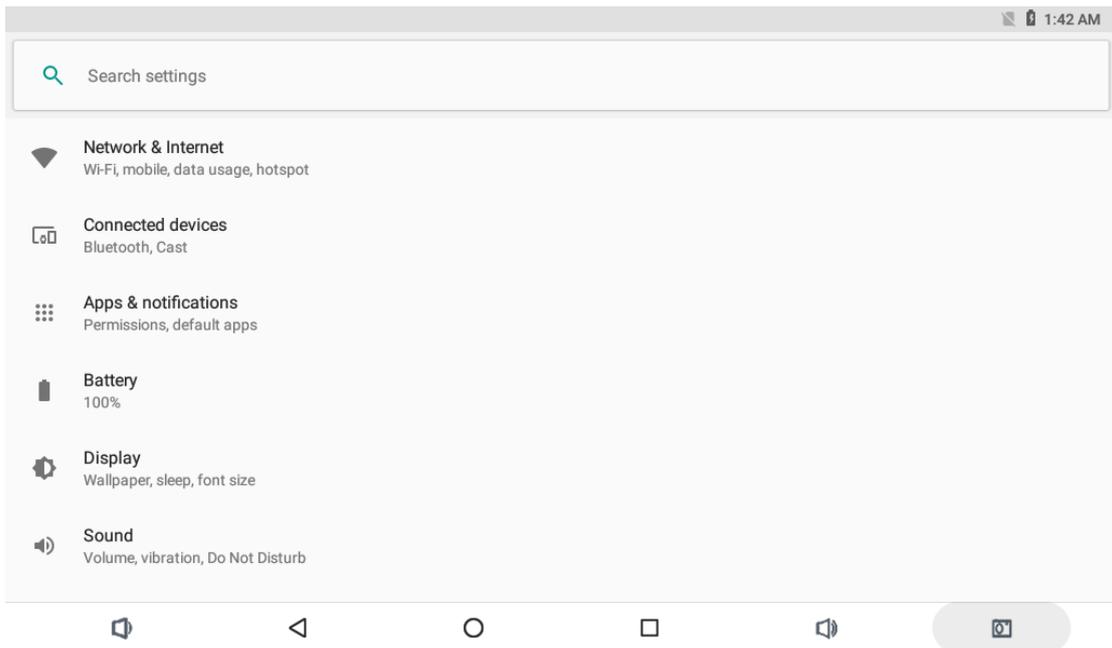


再点击播放按钮：

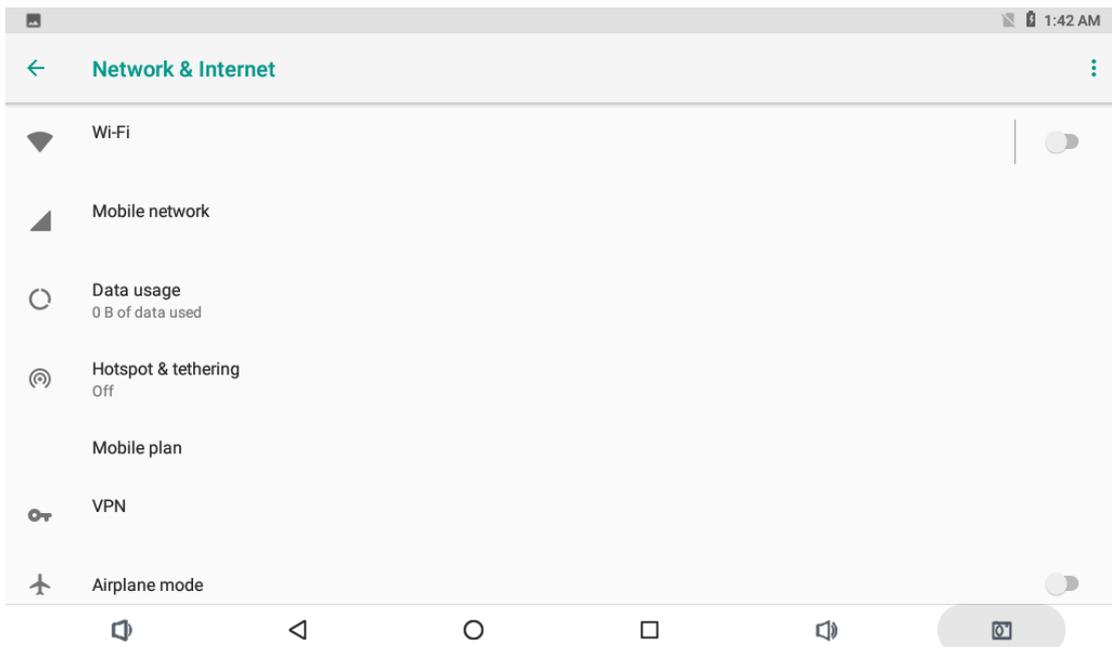


6.4 使用 WIFI 上网

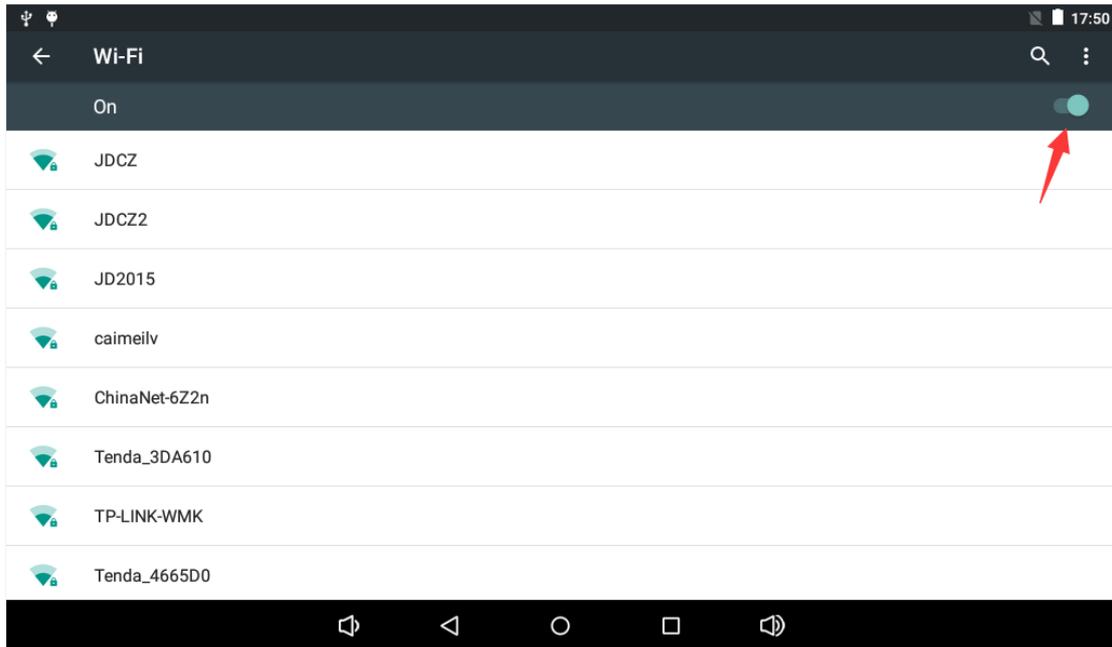
X30 开发板自带 wifi/BT 二合一模组，无需额外 USBwifi 即可无线上网。启动开发板，点击设置，点击 Network & Internet:



在 Wi-Fi 一栏的方框中有个关闭按钮，将他拨到右边，即打开状态，如下图：

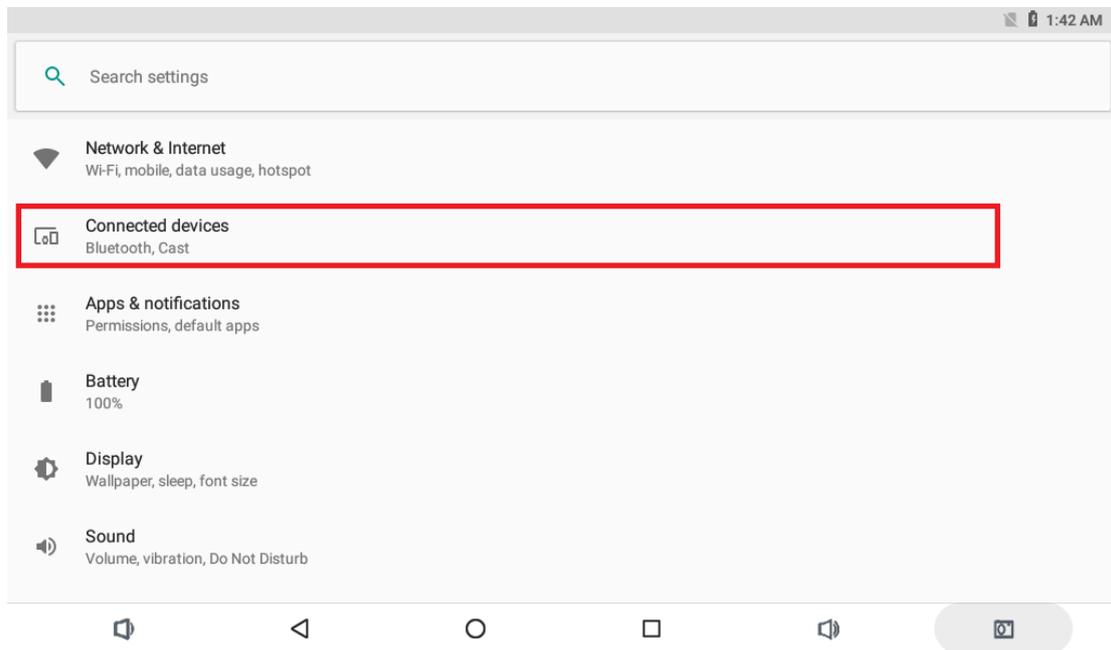


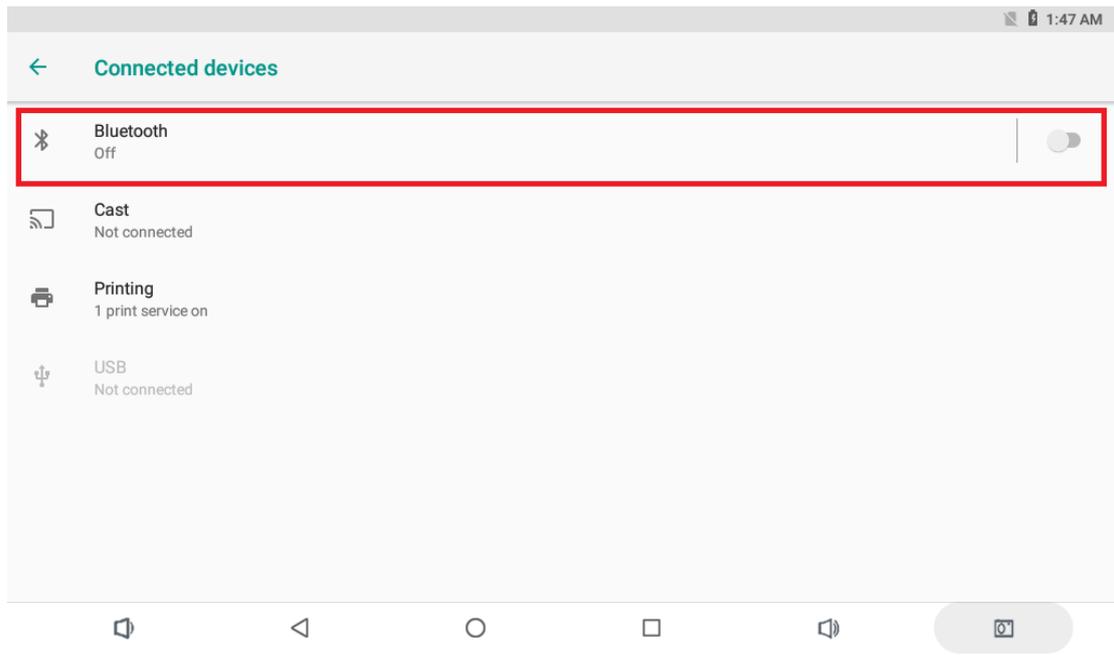
再点击 Wi-Fi 一栏，进入 wifi 界面，找到属于自己的无线网络信号并点击，输入密码后，点击连接即可上网。



6.5 使用蓝牙传输数据

x30 开发板支持 AP6212 芯片的 WIFI/BT 二合一模块。点击设置->连接设备（Connected devices）->Bluetooth，如下图所示：





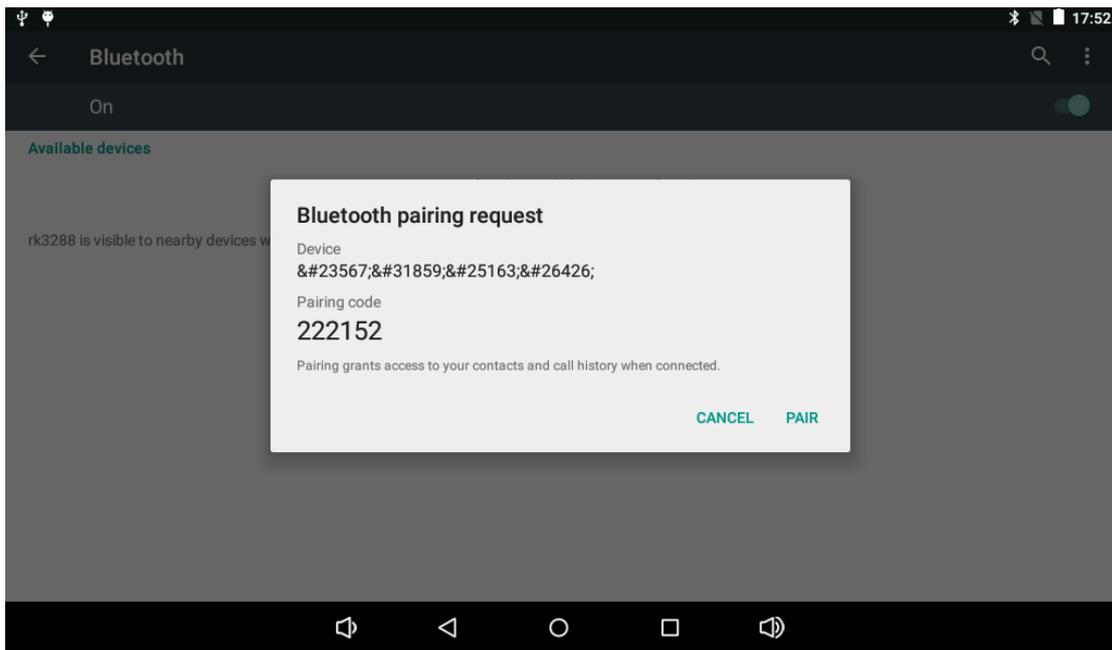
Available devices

小米手机

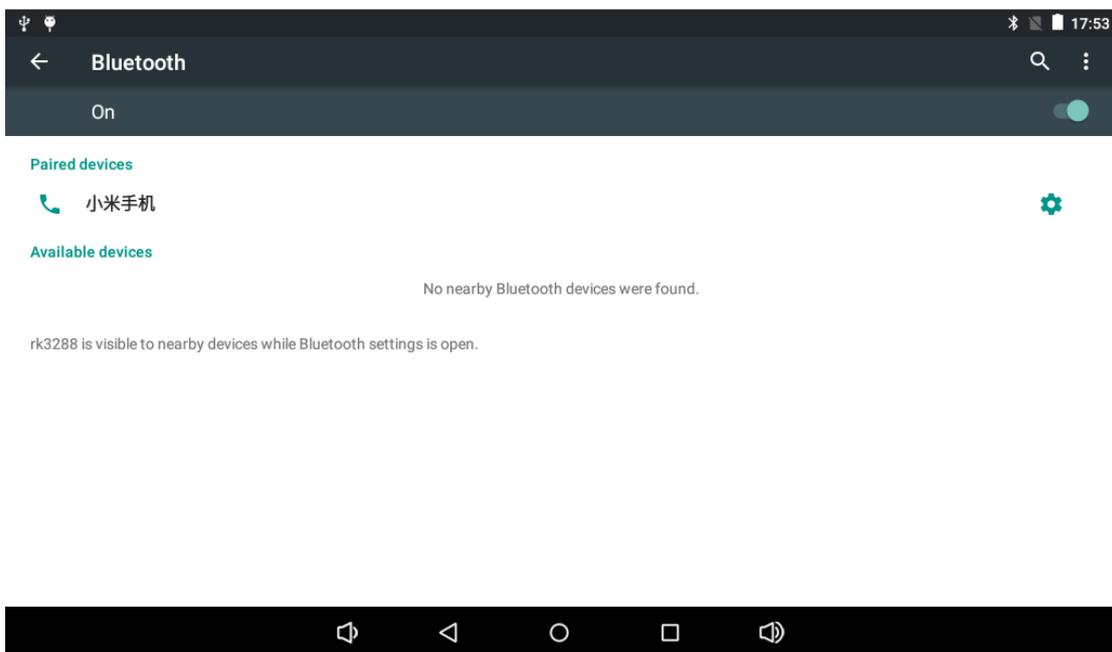
rk3288 is visible to nearby devices while Bluetooth settings is open.



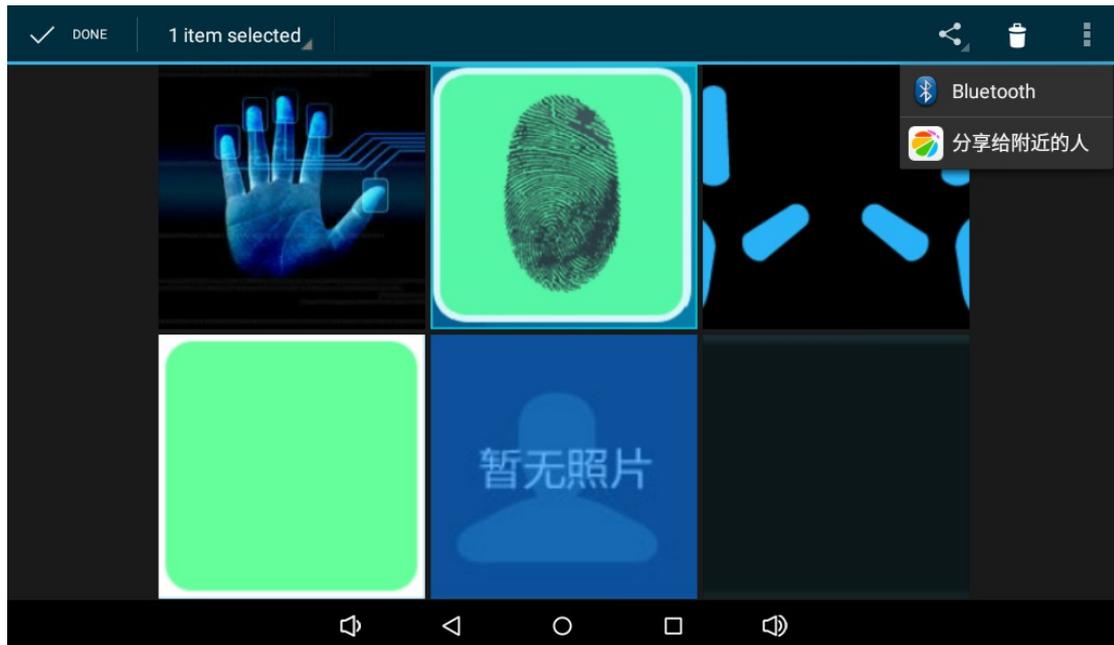
点击蓝牙，进入设置界面，找到一个支持蓝牙设备的安卓手机，并打开蓝牙，且可被发现。在 x30 开发板的蓝牙设置界面的右上角点击搜索设备，在可用设备一栏中将会显示出所有找到的蓝牙设备。找到需要匹配的蓝牙设备并点击，在开发板上弹出配对请求：



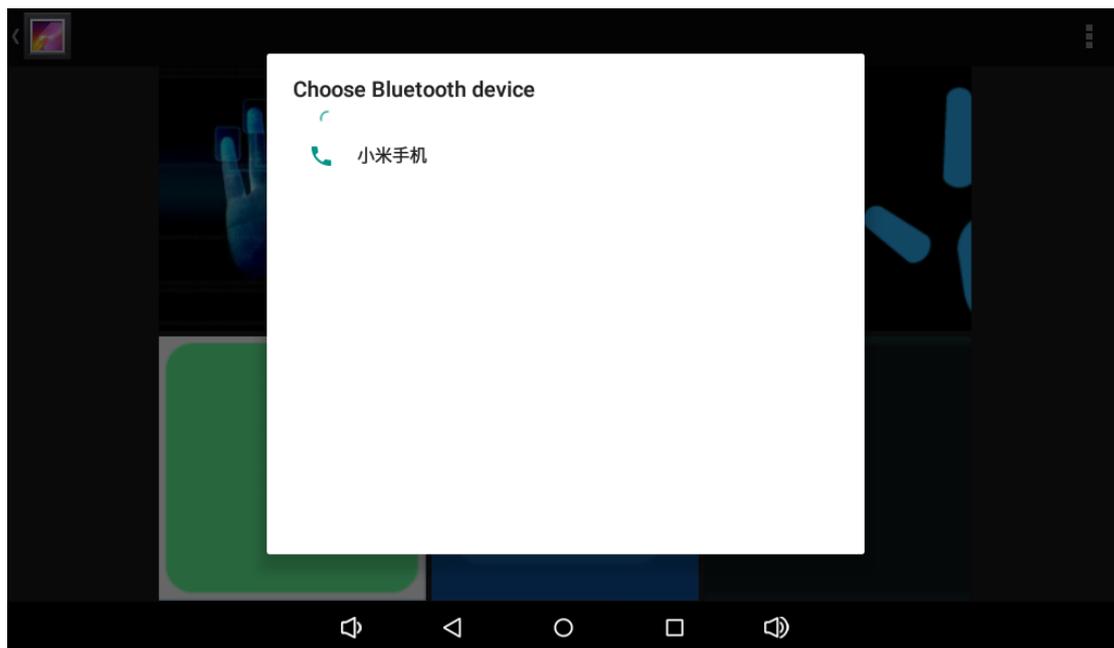
点击配对，同时，在手机端也会弹出类似的画面，也点击配对即可。成功配对后的界面如下：



这时，配对的设备间就可以通过蓝牙共享文件了。点击开发板的图库，找到一张图片并选中：



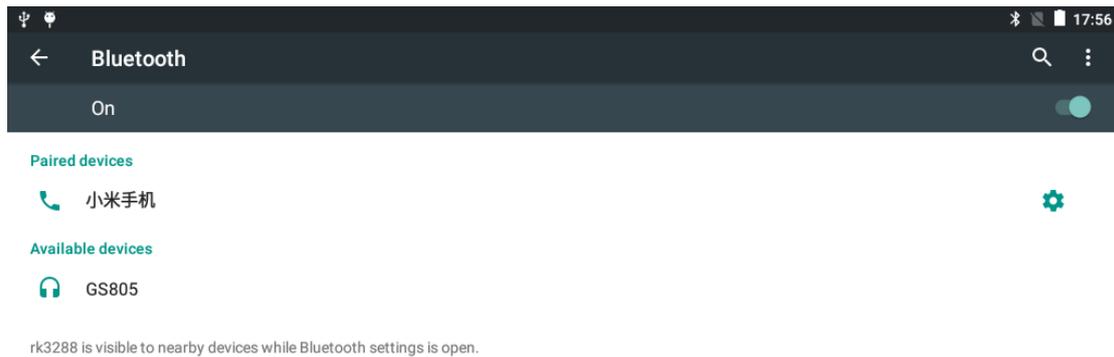
点击右上角的分享按钮，选择蓝牙，弹出如下界面：



选择配对成功的蓝牙设备，这时在配对设备上将会弹出一个接收文件的对话框，点击接收即可。

6.6 使用蓝牙播放音乐

x30 开发板可支持通过蓝牙连接蓝牙音箱，并播放音乐。准备一个蓝牙音箱，并切换到蓝牙模式，进入开发板的蓝牙界面并打开，点击右上角的搜索设备，将会找到蓝牙音箱，下图中的 GS805 即是蓝牙音箱：



点击 GS805，稍等几秒，将会配对成功，无需任何确认动作。这时，随意在开发板上播放音视频，音乐将会通过蓝牙音箱播放出来。

6.7 使用 USB 鼠标键盘

启动开发板，将 USB 鼠标或者 USB 无线鼠标键盘接到 USB HOST 接口，即可使用鼠标键盘操作 android 界面了。

6.8 挂载 TF 卡

系统启动后，会自动挂载卡槽中的 TF 卡到/storage/目录，并根据 UUID 识别并挂载，如下图所示：

```
px30_evb:/ #
px30_evb:/ # cd /storage/
px30_evb:/storage # ls
B4BA-F8E1 emulated self
px30_evb:/storage # cd B4BA-F8E1/
px30_evb:/storage/B4BA-F8E1 # ls
StabilityTest.apk x4418-android4-2PCS-2GB x6818-qt
ddr-test x4418-qt
px30_evb:/storage/B4BA-F8E1 #
px30_evb:/storage/B4BA-F8E1 #
```

6.9 挂载 U 盘

启动开发板，并进入 android 系统。插入 U 盘后，系统会将 U 盘自动挂载到/storage 目录。

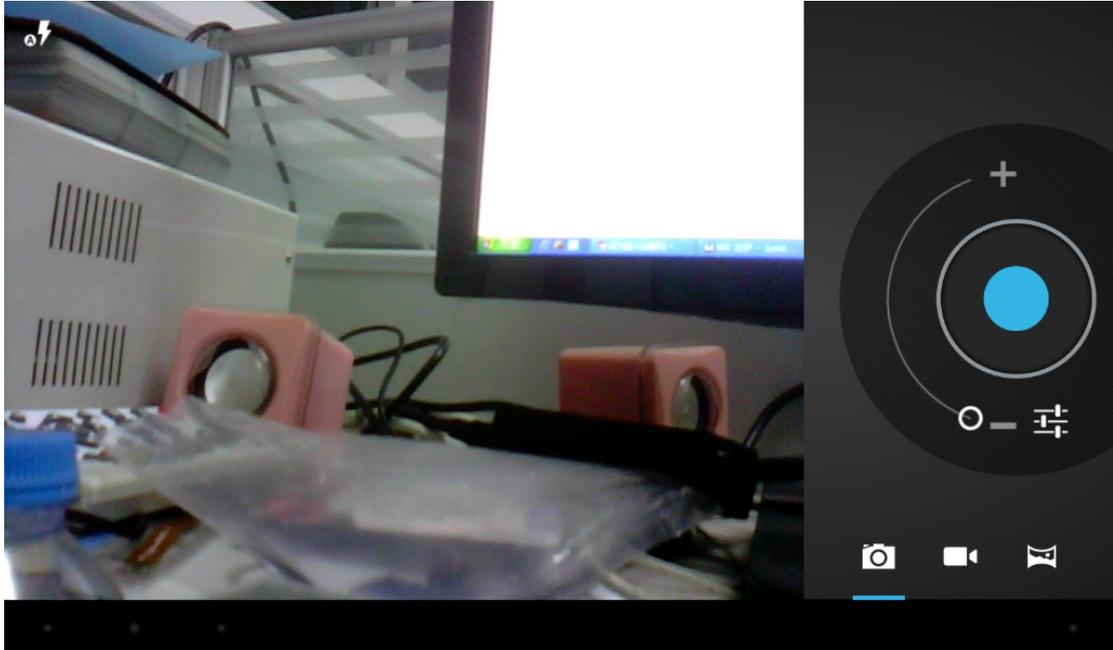
6.10 屏幕旋转

重力传感器已经集成到开发板上，将开发板移动到四周任一方向，界面会随之改变。当然并不是所有应用程序都会随之改变，有部分应用程序不支持屏幕旋转。



6.11 拍照摄相

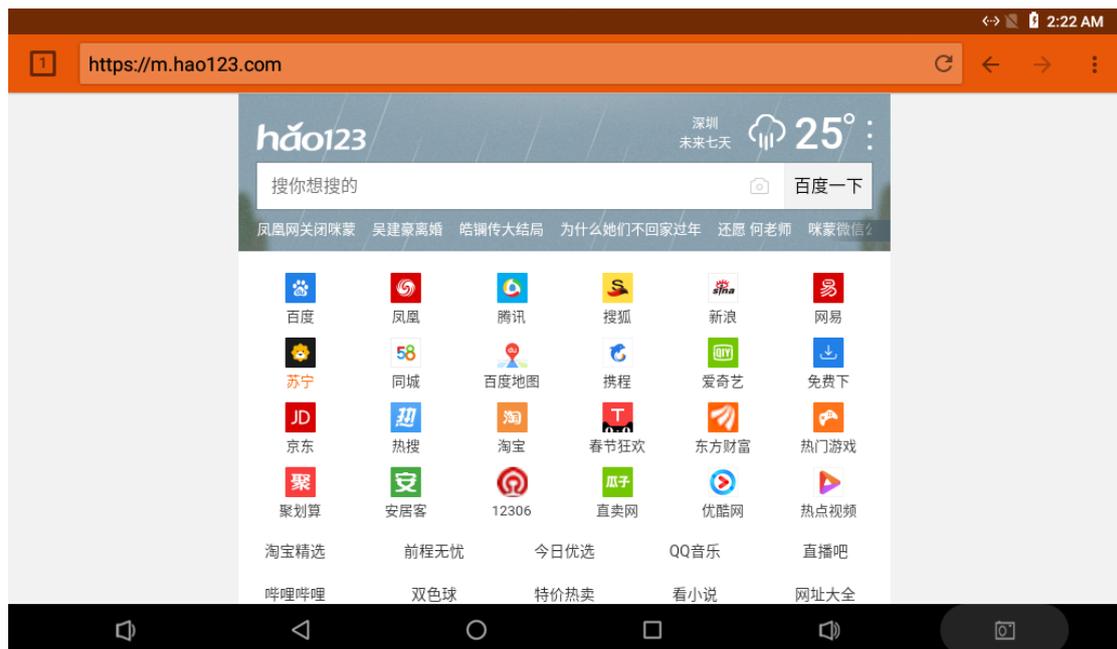
点击 android 应用中的相机，会进入图像预览模式。点击右下角的拍照按钮即可拍照，如下图所示：



右下角可以切换拍照和录像功能。

6.12 使用有线以太网上网

将可以正常上网的网线连接到开发板的网口座上，网口座的指示灯会正常闪烁，正常上网界面如下：



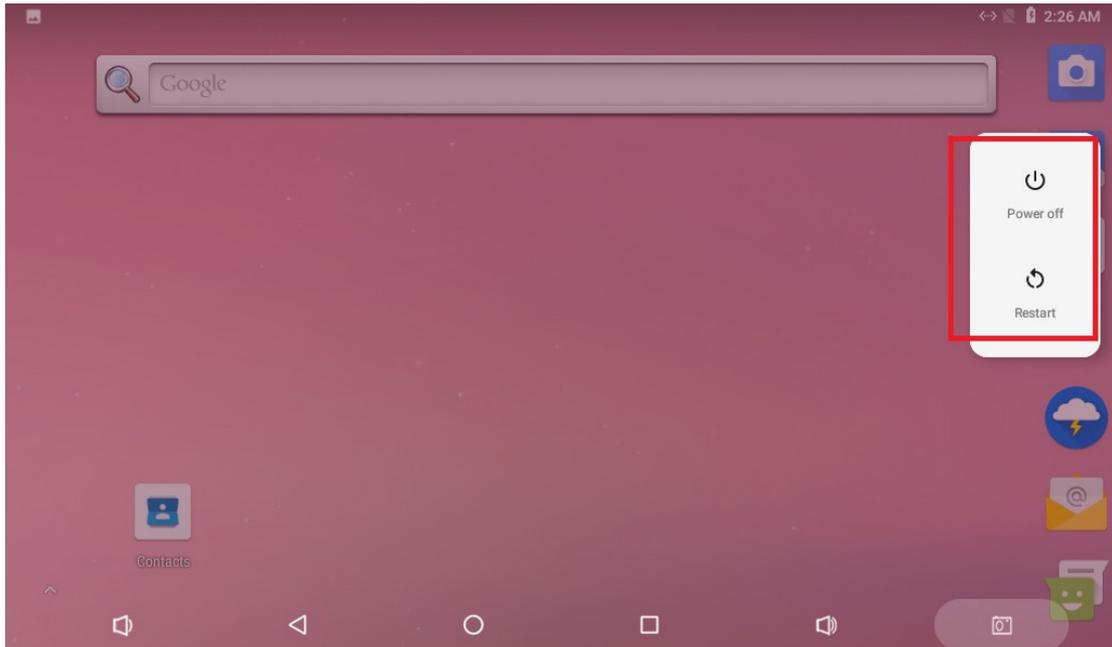


6.13 使用遥控器操作开发板

默认 x30 开发板硬件支持红外遥控器，我们可以通过遥控器操作 android 界面，在泰捷视频等电视界面通过遥控器操作，开发板就完全变成了一个机顶盒。

6.14 开关机

x30 开发板使用了 PMU 进行电源管理。当外接上 12V 的电源适配器后，x30 开发板将会被自动点亮，进入系统后，长按开机键，会弹出一个选择对话框，点击关机按钮将会弹出确认关机的对话框，点击确认即可关机。



6.15 休眠唤醒

进入 android 文件系统后，轻按开关机键，屏幕将会熄灭，然后进入深度休眠状态。可以通过串口查看进入休眠的打印信息。再次轻按开关机键，开发板将会唤醒。



第7章 Android 测试程序

我们在 x3288, x3399, x30 开发板上开发了强大的安卓测试软件，基本上可以测试开发板的所有硬件功能，它在产品量产，程序开发上有很大的参考价值。在 APP 界面点击安卓测试，即可进入测试界面，使用触摸屏左右滑动，或者用鼠标滑动可切换测试的硬件。

7.1 液晶屏测试



在液晶屏测试界面，点击中间的任意纯色位置，会有不同的颜色变化，我们可以观察 LCD 是否有丢色，坏点等。

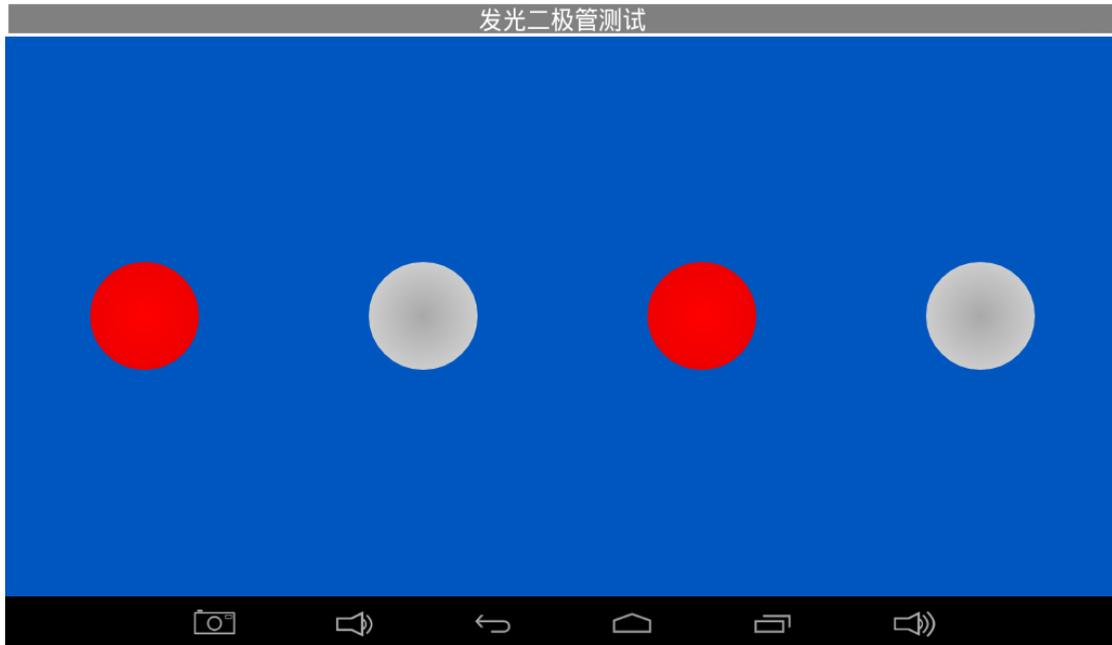
7.2 触摸屏测试





在触摸屏测试界面，点击开始测试，即可在屏幕上任意手写，在批量生产时，我们通常通过画对角线测试触摸电路是否正常。

7.3 发光二极管测试



点击图片上任意的灯，为红色时，对应开发板上的 LED 灯亮，为灰色时，对应 LED 灯灭。

7.4 蜂鸣器测试



按住开始测试键，蜂鸣器会鸣叫，松开开始测试键，蜂鸣器停止鸣叫。

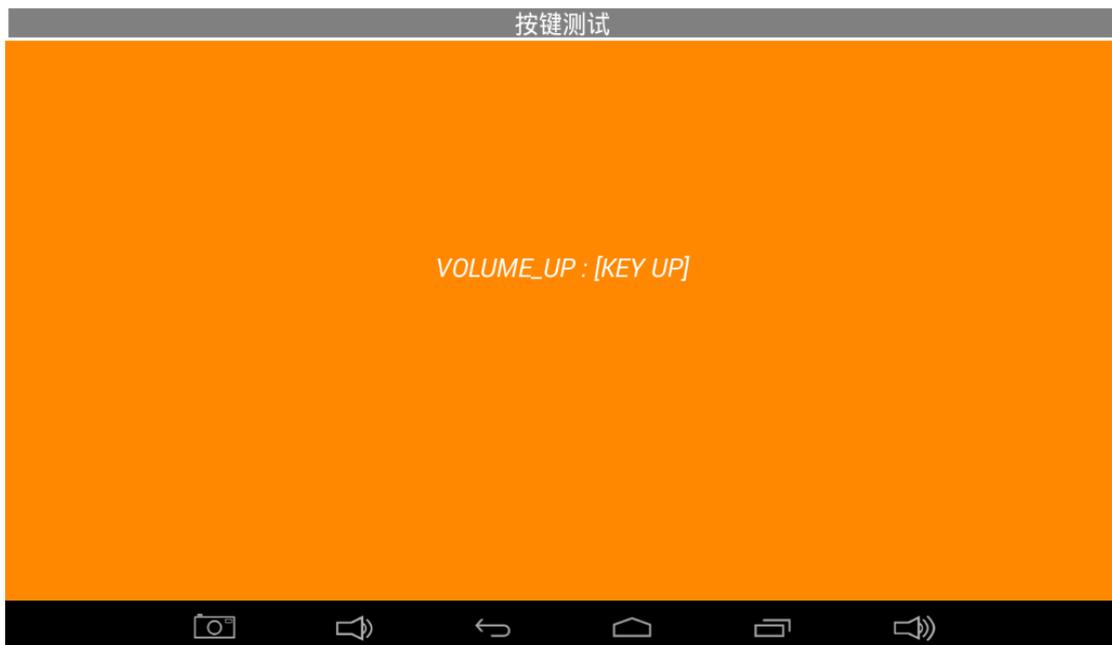


7.5 背光测试



划动中间的圆圈，背光亮度会随之发生变化。

7.6 按键测试



按下或抬起开发板上四个独立按键的任意一个，界面上将会提示对应的按键的按下和抬起动作。



7.7 电池测试



该界面反应了接在开发板上的电池电量信息。

7.8 数模转换测试



监测四路 ADC 的电压。我们可以通过调节开发板上的精密可调电位器，观察上面对应通道的电压是否变化。



7.9 重力传感器测试



旋转开发板时，上面的 X，Y，Z 轴的值会随之发生变化。

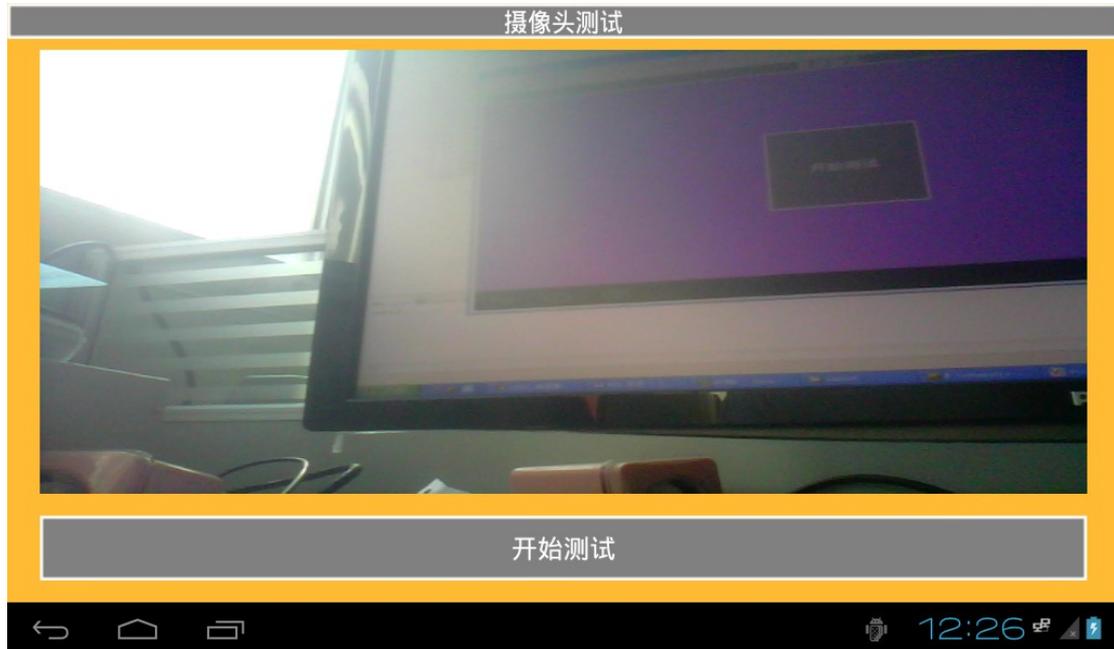
7.10 音频测试



点击开始测试，会听到有清脆的声音出来。



7.11 摄像头测试



装上摄像头，点击开始测试，会看到摄像头抓到的预览界面显示出来。

7.12 无线网络测试



使用 WIFI 连上无线网络后，无线网络测试界面会搜索到附近的网络并列出来。



7.13 网络连接测试



当有线或无线网络正常连接时，网络连接测试界面可以浏览网页。

7.14 串口测试



将需要测试的串口的 TXD 和 RXD 短路，再点击开始测试，界面上会提示对应串口会自发自收 0123456789。如果不能自发自收，说明对应的串口硬件有问题。



7.15 外部存储器测试



将 TF 卡插到开发板上，点击开始测试，界面上会显示 TF 卡的相关信息。

7.16 优盘测试



将 U 盘插到开发板上，点击开始测试，界面上会显示 U 盘的相关信息。



第8章 android 内核驱动

8.1 G-sensor 驱动

路径: kernel/drivers/input/sensors/accel/lis3dh.c
hardware/rockchip/sensor/st

文件: lis3dh.c 及 st 目录

8.2 电容触摸屏驱动

路径: kernel/drivers/input/touchscreen/

源码: gsIX680.c

8.3 液晶屏驱动

路径: kernel/drivers/gpu/drm/panel/

源码: panel-simple.c

注: 显示驱动基于 drm 框架, 不止这些文件, 基本位于 kernel/drivers/gpu/drm 目录下

8.4 按键驱动

路径: kernel/drivers/input/keyboard

源码: rk_keys.c adc-keys.c

8.5 WIFI/BT 模块驱动

路径: kernel/drivers/net/wireless/rockchip_wlan

源码: 该目录下所有源码文件

8.6 摄像头驱动

路径: hardware/rockchip/camera/SiliconImage/isi/drv

源码: 该目录下

8.7 proc 文件系统

8.7.1 启动环境变量查询

使用如下指令查询启动环境变量配置:

```
cat /proc/cmdline
```

会有如下类似打印信息:

```
storagemedia=emmc          androidboot.mode=emmc          androidboot.oem_unlocked=0
earlyprintk=uart8250,mmio32,0xff160000          swiotlb=1          console=ttyFIQ0
androidboot.baseband=N/A  androidboot.veritymode=enforcing  androidboot.hardware=rk30board
androidboot.console=ttyFIQ0  init=/init  kpti=0
```

8.7.2 CPU 信息查询

使用如下指令查询 CPU 信息:

```
cat /proc/cpuinfo
```

会有如下类似打印信息:

```
px30:/ # cat /proc/cpuinfo
```



```
processor: 0
BogoMIPS   : 48.00
Features   : fp asimd evtstrm aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant   : 0x0
CPU part      : 0xd04
CPU revision  : 2
```

```
processor: 1
BogoMIPS   : 48.00
Features   : fp asimd evtstrm aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant   : 0x0
CPU part      : 0xd04
CPU revision  : 2
```

```
processor: 2
BogoMIPS   : 48.00
Features   : fp asimd evtstrm aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant   : 0x0
CPU part      : 0xd04
CPU revision  : 2
```

```
processor: 3
BogoMIPS   : 48.00
Features   : fp asimd evtstrm aes pmull sha1 sha2 crc32
CPU implementer : 0x41
CPU architecture: 8
CPU variant   : 0x0
CPU part      : 0xd04
CPU revision  : 2
```

```
Serial      : edb5c6db7629d0f1
```

8.7.3 内存信息查询

使用如下指令查询内存使用信息:

```
cat /proc/meminfo
```

会有如下类似打印信息:

```
px30:/ # cat /proc/meminfo
```



```
MemTotal:      977612 kB
MemFree:       37360 kB
MemAvailable:  568044 kB
Buffers:       3152 kB
Cached:        566344 kB
SwapCached:    0 kB
Active:        285836 kB
Inactive:      498400 kB
Active(anon):  107816 kB
Inactive(anon): 135952 kB
Active(file):  178020 kB
Inactive(file): 362448 kB
Unevictable:   256 kB
Mlocked:       256 kB
SwapTotal:     488268 kB
SwapFree:      488256 kB
Dirty:         432 kB
Writeback:     0 kB
AnonPages:     215068 kB
Mapped:        405048 kB
Shmem:         29056 kB
Slab:          64960 kB
SReclaimable: 26980 kB
SUnreclaim:   37980 kB
KernelStack:  12512 kB
PageTables:    20084 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:  977072 kB
Committed_AS: 53788924 kB
VmallocTotal: 258867136 kB
VmallocUsed:   0 kB
VmallocChunk:  0 kB
CmaTotal:      16384 kB
CmaFree:       15060 kB
```

8.7.4 磁盘分区信息查询

使用如下命令查询磁盘分区信息:

```
cat /proc/partitions
```

会有如下类似打印信息:

```
px30:/ # cat /proc/partitions
major minor #blocks name
```



```
1      0      8192 ram0
1      1      8192 ram1
1      2      8192 ram2
1      3      8192 ram3
1      4      8192 ram4
1      5      8192 ram5
1      6      8192 ram6
1      7      8192 ram7
1      8      8192 ram8
1      9      8192 ram9
1     10      8192 ram10
1     11      8192 ram11
1     12      8192 ram12
1     13      8192 ram13
1     14      8192 ram14
1     15      8192 ram15
254    0     488272 zram0
179    0     7634944 mmcblk2
179    1         4096 mmcblk2p1
179    2         4096 mmcblk2p2
179    3         4096 mmcblk2p3
179    4        16384 mmcblk2p4
179    5        32768 mmcblk2p5
179    6        32768 mmcblk2p6
179    7        65536 mmcblk2p7
179    8       114688 mmcblk2p8
179    9         4096 mmcblk2p9
179   10       393216 mmcblk2p10
179   11      1572864 mmcblk2p11
179   12        16384 mmcblk2p12
179   13       393216 mmcblk2p13
179   14       131072 mmcblk2p14
179   15          512 mmcblk2p15
179   16      4840943 mmcblk2p16
179   96          512 mmcblk2rpmb
179   64         4096 mmcblk2boot1
179   32         4096 mmcblk2boot0
```

8.7.5 内核版本查询

使用如下命令查询内核版本：

```
cat /proc/version
```

会有如下类似打印信息：



```
px30:/ # cat /proc/version
```

```
Linux version 4.4.138 (lxb@9tripod-server) (gcc version 6.3.1 20170404 (Linaro GCC 6.3-2017.05)) #9 SMP PREEMPT Wed Feb 20 20:21:04 CST 2019
```

8.7.6 网络设备查询

使用如下命令查询网络设备信息：

```
cat /proc/net/dev
```

会有如下类似打印信息：

```
px30:/ # cat /proc/net/dev
```

Inter-	Receive										Transmit			
face	bytes	packets	errs	drop	fifo	frame	compressed	multicast	bytes	packets	errs	drop		
fifo	colls	carrier	compressed											
sit0:		0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	0	0		0							
lo:		0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	0	0		0							
wlan0:		0	0	0	0	0	0	0	0	0	168			
2	0	0	0	0	0		0							
eth0:	68045	1003	0	0	0	0	0	0	0	0	29109			
272	0	0	0	0	0		0							

8.7.7 查看内核启动信息

使用如下命令查询内核启动信息：

```
cat /proc/kmsg
```



第9章 android 开发项目实战

9.1 实战 1：如何修改 init.rc 文件

9.1.1 永久修改

在 system/core/rootdir 目录下找到 init.rc 文件，修改后执行 ./mk -s，将会生成新的 system.img 文件，重新烧写 system.img 即可更新 init.rc。注意，在编译时，首先需要先将 out 目录下的 init.rc 删除后再编译，否则可能不会生效。

9.1.2 临时修改

默认 init.rc 在 ramdisk.img 中，ramdisk.img 又被打包到 boot.img 中，因此，如果我们需要临时修改 boot.img，首先需要解压 boot.img，再手动修改我们需要修改的内容，再打包即可。为此，我们特别编写了解压和压缩的脚本文件。

在 android 根目录下，新建脚本文件 ramdisk.sh，编辑内容如下：

```
#!/bin/bash
# create: liuqiming
# date: 2016-11-22
# mail: liuqiming@9tripod.com

echo "Modify boot.img"
echo "1.unzip boot.img"
echo "2.Create boot.img"
echo "3.delete cache files"
echo "4.exit"

#准备工作，环境变量的声明
SOURCE_DIR=$(cd `dirname $0` ; pwd)
TOOLS_DIR=${SOURCE_DIR}/tools
TARGET_DIR=${SOURCE_DIR}/out/release
OUT=${SOURCE_DIR}/out/target/product/rk3288

PATH=${SOURCE_DIR}/out/host/linux-x86/bin:$PATH;

#读取要执行的操作索引
read -p "Choose:" CHOOSE

#解压 boot.img
if [ "1" = ${CHOOSE} ];then
    echo "unzip boot.img"
    cd $TARGET_DIR
#第一步：新建 ramdisk 目录
    [ -e "ramdisk" ] || { echo "mkdir ramdisk"; mkdir ramdisk;}
#第二步：判断是否存在 boot.img 文件
```



```
[ -f "$TARGET_DIR/boot.img" ] || { echo "error! can't find boot.img!"; rm -rf ramdisk;exit; }
cd ramdisk
#第三步: 拷贝 boot.img
cp ../boot.img .
#第四步: 解压 boot.img 文件, 得到 boot.img-ramdisk.gz 文件
[ -f "/bin/split_bootimg.pl" ] || { cp $TOOLS_DIR/split_bootimg.pl /bin; }
split_bootimg.pl boot.img
rm -f boot.img
#第五步: 解压 boot.img-ramdisk.gz 文件
gzip -dc boot.img-ramdisk.gz | cpio -i
echo "^_^ unzip boot.img finished!"
rm -f boot.img-ramdisk.gz
rm -f boot.img-second.gz
rm -f boot.img-kernel
exit

#打包 boot.img
elif [ "2" = ${CHOOSE} ];then
    echo "create boot.img with kernel..."
#第一步: 判断是否存在 ramdisk 目录
[ -e "$TARGET_DIR/ramdisk" ] || { echo "can't find [ramdisk],please unzip boot.img first!"; exit; }
#第二步: 去除默认的 root.img 文件
rm -f $TARGET_DIR/boot.img
cd $TARGET_DIR
#第三步: 将当前目录所有文件打包到 boot.img
[ -d $TARGET_DIR/ramdisk ] && \
mkbootfs $TARGET_DIR/ramdisk | minigzip > $TARGET_DIR/ramdisk.img && \
truncate -s "%4" $OUT/ramdisk.img && \
mkbootimg --kernel ${OUT}/kernel --ramdisk ${OUT}/ramdisk.img --second $TARGET_DIR/resource.img
--output $TARGET_DIR/boot.img
rm -rf ramdisk
exit

#清除残余文件
elif [ "3" = ${CHOOSE} ];then
    cd ${TARGET_DIR}
    rm -rf ramdisk
    echo "^_^ delete cache files ok!"

#退出脚本
elif [ "4" = ${CHOOSE} ];then
    exit
fi
```



执行 ./ramdisk.sh, 输入 1, 回车, 将会自动解压 out/release 目录下的 boot.img, 并存放到 out/release/ramdisk 目录。

执行 ./ramdisk.sh, 输入 2, 回车, 将会自动将 out/release/ramdisk 目录下的文件打包成 boot.img。

9.2 实战 2: 如何修改开机动画

默认 android 启动时, 会有 android 的英文字母不断高亮滚动显示, 构成了一幅美丽的开机动画。其实这个开机动画是可以定制的, 有兴趣的读者可以自行尝试。

9.3 实战 3: 如何修改 uboot 中的开机 LOGO

瑞芯微官方提供的 uboot 会读取 kernel 根目录下的 logo.bmp 和 logo_kernel.bmp 两个文件, 分别用于做 uboot 和内核的开机 LOGO, uboot 和内核的 LOGO 显示全部是由 uboot 完成的。其中 logo.bmp 用作 uboot 的开机 LOGO, logo_kernel.bmp 用作内核的开机 LOGO。值得注意的是, 如果我们要将 uboot 和内核显示的 LOGO 设置得相同, 并不能简单的复制, 因为他们的图片格式是有区别的。logo.bmp 支持 8b 的 bmp 格式图片, 而 logo_kernel.bmp 只支持压缩后的 8b 的 bmp 格式图片, 因此 logo_kernel.bmp 实际上要比 logo.bmp 的尺寸要小得多。而且在实际显示时, 很明显的看到显示内核 LOGO 时, 如果颜色比较丰富, 就会失真。为了只显示一幅图片, 我们直接将 kernel 目录下的 logo_kernel.bmp 删除即可。

如果我们需要换 uboot 的开机 LOGO, 找一张分辨率不是太大的图片, 转换成 BMP 格式, 然后使用 WINDOWS 的图库编辑功能, 转换成 8b 模式, 保存并替换 kernel 目录下的 logo.bmp 即可。注意, 如果 LOGO 的分辨率设置得过大, 也可能导致无法显示。如果需要全屏开机 LOGO, 默认的 uboot 代码就不适用了, 需要重新移植。

9.4 实战 4: 如何修改内核中的 LOGO

同上, 找一张分辨率不是太大的图片, 转换成 BMP 格式, 在 WINDOWS 的图片编辑软件下转换成 8b 模式后, 还需要借助于 linux 下强大的 gimp 工具, 将图片压缩, 内核 LOGO 才能显示出来。默认瑞芯微提供的这套方案, 内核 LOGO 分辨率不够大, 而且会明显失真, 无法显示真彩图片, 如果开发的工程师一定要显示完美的内核 LOGO, 建议使用 linux 下标准的开机 LOGO 方案。

9.5 配置使用 MIPI 屏

在 kernel/arch/arm64/boot/dts/rockchip 目录下存放了多个已经调试好的液晶屏的配置文件, 7 寸 RGB 屏的配置文件为 lcd_1024x600_rgb.dtsi, 7 寸 MIPI 屏的配置文件为 lcd_1024x600_mipi.dtsi。在 kernel/arch/arm64/boot/dts/rockchip/x30_evb.dts 中选择配置使用的液晶模组即可, 如需要使用 7 寸 MIPI 屏, 示例代码如下:

```
#include "lcd_1024x600_mipi.dtsi" //7ch mipi panel
##include "lcd_1024x600_rgb.dtsi" // 7ch rgb panel
```



第10章 其他产品介绍

10.1 核心板系列

处理器型号	核心板型号	备注
S3C6410	X6410CV1	200PIN 插针接口
S5PV210	X210CV3	180PIN 邮票孔接口
	X210CV4	144PIN 邮票孔接口
	G210CV1	200PIN 金手指接口
	I210CV2	200PIN 插针接口
Exynos4412	X4412CV3	180PIN 邮票孔接口
S5P4418	X4418CV3.3	180PIN 邮票孔接口
	I4418CV2	200PIN 板对板连接器
S5P6818	X6818CV3.3	180PIN 邮票孔接口
	I6818CV2	200PIN 板对板连接器
RK3128	X3128CV2	144PIN 邮票孔接口
RK3288	X3288CV3	180PIN 邮票孔接口
	I3288CV1	220PIN 邮票孔接口
RK3399	X3399CV3	200PIN 邮票孔接口
RK3399	X3399CV4	200PIN 邮票孔接口
PX30	X30CV1	144PIN 邮票孔接口

10.2 开发板系列

处理器型号	开发板型号	备注
S3C6410	x6410 开发板	x6410cv1 评估板
S5PV210	x210 开发板	x210cv3 评估板
	g210 开发板	g210cv1 评估板
	i210 开发板	i210cv2 评估板
Exynos4412	x4412 开发板	x4412cv3 评估板
S5P4418	x4418 开发板	x4418cv3 评估板
S5P6818	x6818 开发板	x6818cv3 评估板
	i6818 开发板	i6818cv2 评估板
RK3128	x3128 开发板	x3128cv2 评估板
RK3288	x3288 开发板	x3288cv3 评估板
	i3288 开发板	i3288cv1 评估板
RK3399	x3399 开发板	x3399cv3/x3399cv4 评估板
PX30	x30 开发板	x30cv1 评估板

10.3 卡片电脑系列

处理器型号	卡片电脑型号	备注
Exynos4412	ibox4412 卡片电脑	
S5P4418	ibox4418 卡片电脑	



S5P6818	ibox6818 卡片电脑	
RK3399	ibox3399 卡片电脑	

说明：产品详细规格，以及更多其他产品请关注九鼎创展官方网站和论坛。