

密级状态：绝密(     )     秘密(     )     内部资料(     )     公开( ☒ )

## RK3399 Linux 工程使用说明文档

文件状态： [   ] 草稿 [   ] 正式发布 [ <input checked="" type="checkbox"/> ] 正在修改	文件标识：	RK3399 Linux 工程使用说明文档
	当前版本：	1.0
	作     者：	hgc
	完成日期：	2017-01-09

文档修改记录:

日期	修订版本	修订内容	修改人	核定人
2017-01-09	V1.0	初始版本。	hgc	lby

## 一、 编译环境:

### 1、源码下载

repo init --repo-url ssh://git@10.10.10.29/tools/repo -u ssh://10.10.10.29:29418/linux/manifests -b master -m rk3399\_linux.xml

### 2、编译

#### 1) uboot 编译

make rk3399\_linux\_defconfig && make ARCHV=aarch64 -j12

#### 2) kernel 编译

make ARCH=arm64 rockchip\_linux\_defconfig && make ARCH=arm64 rk3399-sapphire-excavator-linux.img -j12

3) rootfs 系统及 app 编译: cd buildroot && make rockchip\_rk3399\_defconfig && cd .. && ./build\_all.sh && ./mkfirmware.sh

4) 最终固件在 rockimg 目录下

## 备注:

**1、交叉编译环境搭建:** 交叉编译工具位于 buildroot/output/host/usr/目录下, 需要将工具的 bin/目录和 arm-rockchip-linux-gnueabi/bin/目录设为环境变量。也可以在顶层目录直接"source envsetup.sh", 会自动配置环境变量, 但只对当前控制台有效。

**2、系统编译:** 执行"./build\_all.sh", 其会自动找到系统中的 rk\_make\_first.sh 和 rk\_make.sh, 然后执行编译命令。如果只需要编译单模块, 可以进入到模块目录下, 执行 rk\_make.sh 或 rk\_make\_first.sh 命令。

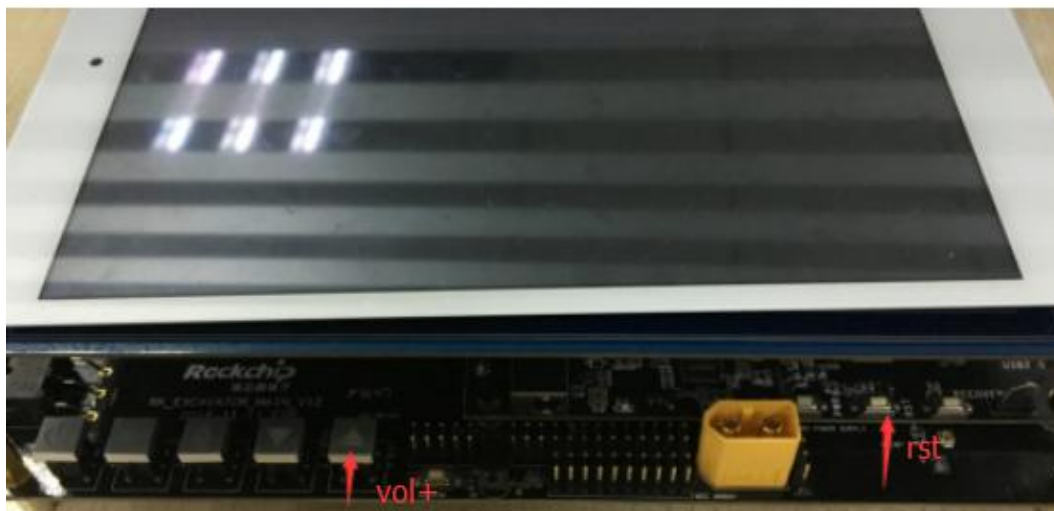
## 二、 烧写:

### 1、接上电源并接好 usb 下载接口



RK3399 挖掘机 样机

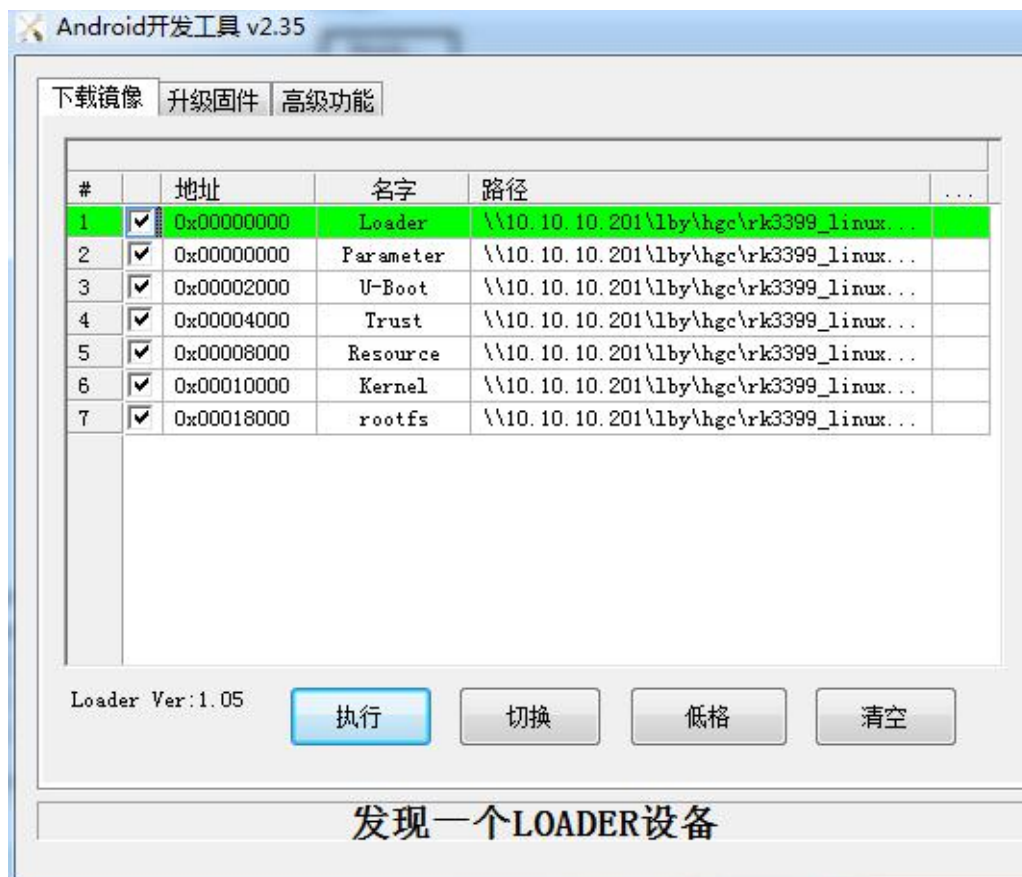
2、按住音量键 vol+不放并按下复位键 rst



RK3399 挖掘机 样机

2、在 PC 端运行烧写工具

2.1 Window 平台，运行 AndroidTool.exe 可执行程序，选上要烧写的固件单击“执行”按钮。

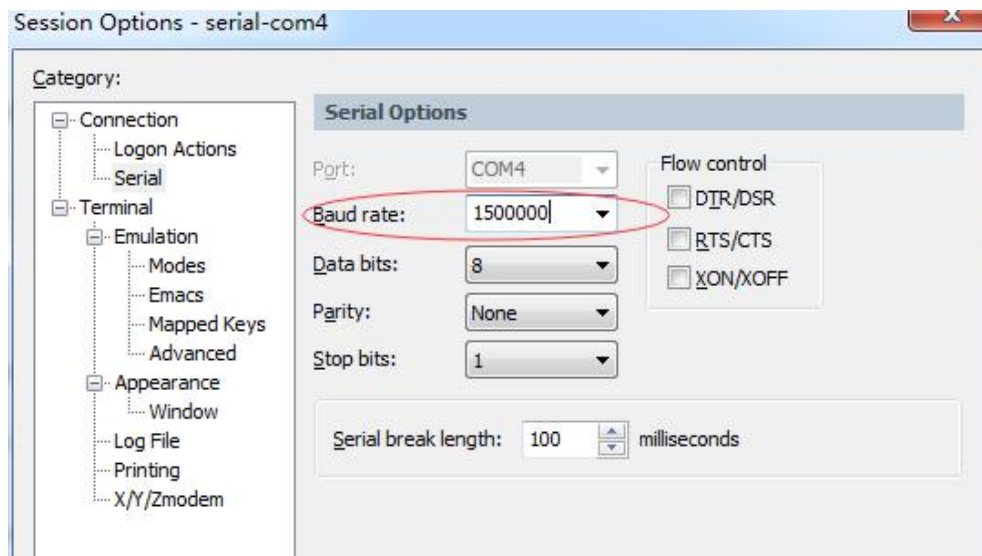


Windows 烧写工具操作界面

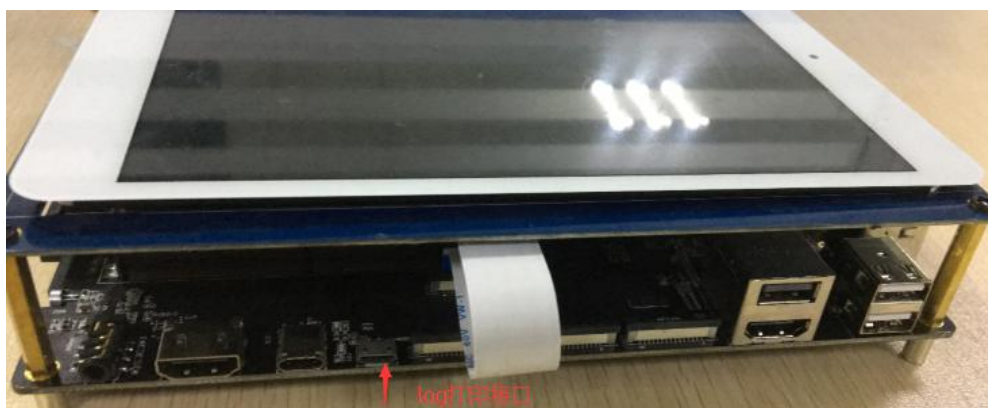
**附：**Windows 平台需要安装驱动，驱动目录在工具目录：USB 驱动  
 \DriverAssitant\_v4.4\DriverAssitant\_v4.4, 直接右键管理员身份执行  
 “DriverInstall.exe” 安装即可

### 三、SecureCRT

#### 1、波特率等参数配置



## 2、调试信息打印接口



RK3399 挖掘机 样机

## 四、工程目录介绍

进工程目录下有 buildroot、app、kernel、u-boot、device、common、config、docs、external、prebuilts、rockimg、rootfs、tools 等目录。每个目录或其子目录会对应一个 git 工程，提交需要在各自的目录下进行。

- 1) buildroot: 定制根文件系统
- 2) rootfs: 存放 debian 相关文件
- 3) app: 存放上层应用 app，主要是 Carmachin 和一些测试应用程序。
- 4) external: 相关库，包括音频、视频、网络等。
- 5) kernel: kernel 代码。
- 6) device/rockchip/rk3399: 存放开机初始化脚本，存放第三方库、bin、alsa/wifi 等配置文件；另还存放编译脚本，系统根目录的几个 sh 脚本都是在 repo sync 的时候，从这里拷贝出来的，所以在若要提交的修改的脚本，必须在 device/rockchip/rk3399 目录下进行。

build\_all.sh : 编译所有第三方库和应用的。

- mkfirmware.sh: 打包最终烧写的固件。
- envsetup.sh: 终端环境变量设置。
- 7) docs: 存放工程帮助文件。
- 8) prebuilts: 存放编译 kernel 需要的 gcc 和交叉编译工具 toolchain。
- 9) rocking: 存放编译输出固件。
- 10) tools: 存放烧写工具。