

密级状态：绝密() 秘密() 内部() 公开(√)

RK1108_CVR_V0.2_20161030 发布说明

(技术部，第三系统产品部)

文件状态： [] 正在修改 [√] 正式发布	当前版本：	V0.2
	作 者：	廖华平
	完成日期：	2016-10-30
	审 核：	洪锦坤 余智超
	完成日期：	2016-10-30

福州瑞芯微电子股份有限公司

Fuzhou Rockchips Semiconductor Co., Ltd

(版本所有,翻版必究)

版 本 历 史

版本号	作者	修改日期	修改说明	备注
V0.1	廖华平	2016.09.04	初始发布 Beta 版本	
V0.2	廖华平	2016.10.30	发布 V0.2 版本	

目 录

1	概述	4
2	主要支持功能	4
3	SDK 获取说明	4
3.1	REPO 下载.....	4
3.2	SDK 下载和同步	4
4	软件开发指南	5
5	SDK 编译说明	5
5.1	交叉编译环境配置.....	5
5.2	kernel 编译步骤	6
5.3	应用层编译步骤.....	6
5.4	固件打包步骤.....	6
6	固件烧写	6
6.1	Windows 平台.....	6
6.2	Linux 平台.....	7
7	按键布局说明	8
附录 A	编译开发环境搭建	9
附录 B	SSH 公钥操作说明	11

1 概述

本 SDK 是基于 Linux 32bit 系统，内核基于 Linux-kernel 3.10。适用于 RK1108_CVR 开发板等产品的开发

本次发布为 bate v0.2 版本，该版本为测试版，仅用于开发使用。该版本支持 1080P 视频录制和回放，支持 OV2710、OV4689、IMX323 摄像头，支持碰撞检测、ADAS、移动侦测等功能。具体的 Camera 调试说明和其他功能接口说明，请阅读工程目录 docs/目录下文档。若有遇到 Bug 反馈到我司 Redmine 系统上。

2 主要支持功能

功能	模块名
数据通信	Wi-Fi、以太网卡、USB、SDCARD
应用程序	视频录制、视频播放

3 SDK 获取说明

SDK 通过我司代码服务器对外发布。搭建编译开发环境参考[附录 A 编译开发环境搭建](#)。

客户向我司技术窗口申请 SDK，需同步提供 SSH 公钥进行服务器认证授权，获得授权后即可同步代码。关于我司代码服务器 SSH 公钥授权，请参考[附录 BSSH 公钥操作说明](#)。

3.1 REPO 下载

repo 是用来管理调用 git 的一个脚本，主要用来下载管理软件仓库，务必使用我司提供的 repo 进行初始化操作。repo 工程下载地址如下：

```
git clone ssh://git@git.rockchip.com.cn:2222/argus-release/argus/repo.git
```

下载后执行即可获取到 repo：

```
cd repo
```

```
tar xvf repo.tar.gz
```

拷贝解压出来的 repo 目录到任意位置，以供下一步使用。建议放到 (/home/user/repo)

3.2 SDK 下载和同步

使用步骤 3.1 获取的 repo 工程进行初始化。假如 repo 工程目录为/home/user/repo，那

么 RK1108_CVR_SDK 下载地址如下：

```
/home/uers/repo/repo init -u ssh://git@git.rockchip.com.cn:2222/argus-release/manifest.git  
-b master -m rk1108_cvr_release.xml
```

然后在当前目录下执行即可下载整个工程代码：

```
.repo/repo/repo sync
```

4 软件开发指南

RK1108_CVR 内核版本：Linux 3.10，上层应用使用我司基于 Linux 自制的一套编译开发系统。为帮助开发工程师更快上手熟悉 SDK 的开发调试工作，随 SDK 发布《RK1108_CVR 软件开发指南》、《各版本 SDK 编译使用指南》。

上述文档可在工程目录下的 docs/目录中获取，并会随我司代码服务器不断更新。

5 SDK 编译说明

5.1 交叉编译环境配置

5.1.1 环境检测

应用层的编译依赖于交叉编译环境。请将系统中之前配置的交叉编译工具移除，避免现混乱。查看之前是否有配置交叉编译工具，请输入命令：

```
arm-linux-gcc --version
```

若此命令有输出，则说明之前有配置过其他的交叉编译环境，此时则需要将配置的环境选项移除。

5.1.2 配置交叉编译环境

交叉编译工具位于 prebuilts/toolschain/usr 目录下，需要将工具的 bin/目录和 arm-rkcvr-linux-uclibcgnueabi/hf/bin/目录设为环境变量。使用以下命令编辑：

```
sudo vim /etc/environment
```

会看到：

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin"
```

将上述两个目录的绝对路径，设置到行末 “” 的前面，注意目录间要用 “:” 隔开。

再次输入命令查看：

```
arm-linux-gcc --version
```

此时会打印出以下 log 即标志为配置成功：

```
arm-linux-gcc.br_real (Buildroot 2016.05-rc2) 4.8.5
```

5.2 kernel 编译步骤

进入工程目录下的 kernel/ 目录，然后执行下列命令，生成 .config：

```
make rk1108_defconfig
```

执行下列命令，编译 kernel：

```
make rk1108-evb-v10.img -j4
```

编译完成后，在 kernel 根目录，会生成 kernel.img，resource.img 两个镜像文件。

5.3 应用层编译步骤

若未按照步骤 5.1 搭建交叉编译环境，或者在服务器开发无法搭建环境，那么可执行命令生成临时的编译环境，这个环境仅限于当前终端使用，掉电丢失：

```
source config/envsetup.sh
```

执行下列命令，即会编译应用所有应用和库：

```
./build_all.sh
```

此命令支持 -jxx、-clean 和 -distclean 参数

5.4 固件打包步骤

执行以下命令，即会打包应用层、根文件系统、kernel.img 和 dtb，最终在 \rockimg\Image-cvr 目录生成 Firmware.img，用于烧写。执行下列命令：

```
./mkfirmware.sh rk1108-evb-v10
```

请注意：不同版本的 SDK 编译和打包方式有些差异，详情请参考《各版本 SDK 开发差异说明》。

6 固件烧写

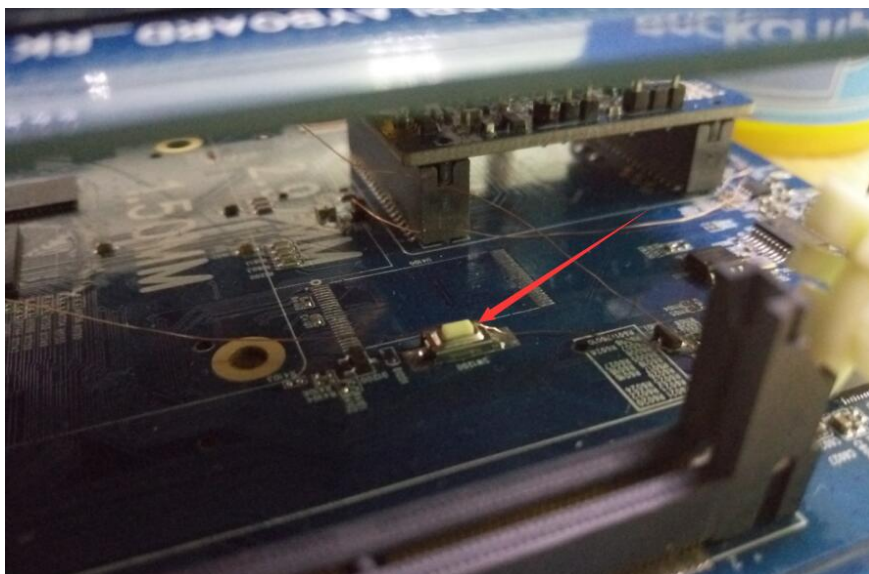
6.1 Windows 平台

此工具可在 winxp/win7/win8/win10 下使用。

初次使用开发工具，需要安装 Rockchip MASKROM 设备专属驱动，驱动存于“tools\CVRTools_Release_v1.00\DriverAssitant_v4.5”目录，运行 DriverInstall.exe 即会自动安装。

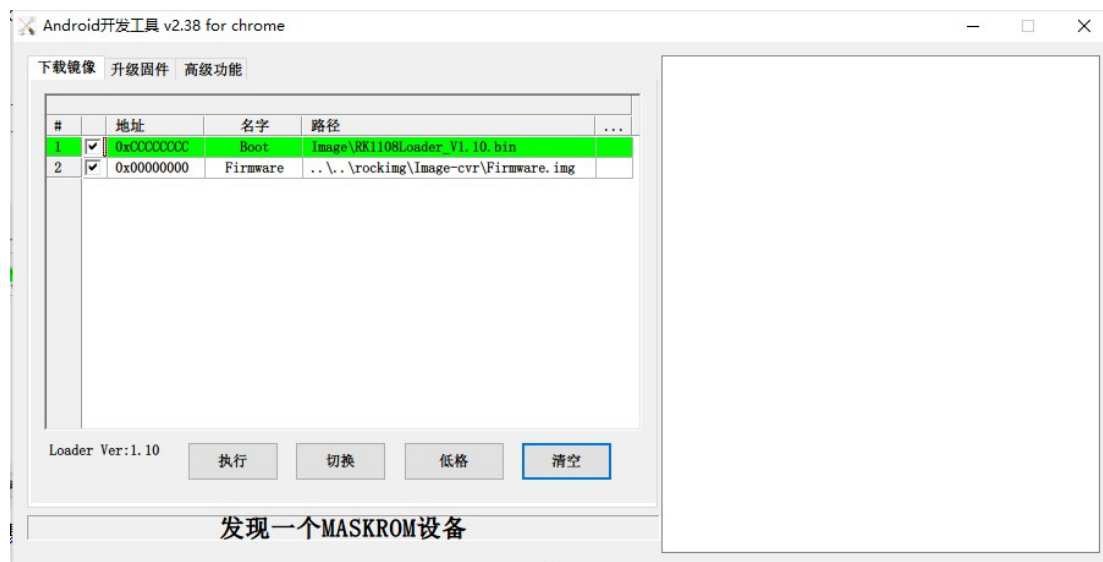
烧写需要进入 MASKROM 并烧写所有固件。在掉电的情况下，按住下图中的白色小按钮，

并上电，就能进入 MASKROM 模式。



MASKROM 上电示意图

如下图，工具下方会提示是否有发现设备，当前设备有进入哪个模式。勾选需要烧写的固件并点击执行，那么固件就会烧写到板子上，等待烧写完毕后，需要重新上电，系统方可运行。



烧写工具介绍

6.2 Linux 平台

Linux 平台下，设备进入 MASKROM 的方式和前面介绍的方法一样。设备进入 MASKROM 模式后，进入工程目录下 tools/Linux_Upgrade_Tool_v1.24/，执行下列命令等待烧写完毕：

```
./cvr_upgrade.sh
```

当出现如下图所示的打印信息表明烧写成功：

```
Download boot ok.  
Write LBA from file (100%)  
Reset Device OK.
```

7 按键布局说明

按键功能说明如下图所示：

复位：重新上电

模式切换：切换拍照、录像、播放模式

设置切换：进入设置窗口，并且切换各个设置选项

录像：在预览模式下，按下“UP”开始录制，再按停止。



按键布局示意图

附录 A 编译开发环境搭建

1. 初始化开发环境

本部分内容包括如何搭建用于 RK1108_CVR 开发的本地环境。您需要在 Linux 或者 Mac OS 环境下搭建，建议使用 Ubuntu14.04 64bit 开发，与我司的开发环境统一，避免出现环境问题。

注意：开发 RK1108_CVR 需要空间约为 35G。如果您只想要搭建一个简单的开发环境，那么您至少需要 45GB 的空间。对于搭建一个完整的环境则将会需要 100GB 以上的空间。

注意：务必要使用 64bit 的操作系统，否则交叉编译工具无法运行。

2. 配置一个 Linux 开发环境

本创建步骤是基于最新的 Ubuntu LTS(14.04)版本，但是大部分发行版本必须保证所需的工具可以运行。

注意：您也可以在虚拟机中搭建环境。如果您在虚拟机中运行 Linux，您需要至少 2GB 的 RAM/swap，或者 30GB 以上的磁盘空间来创建编译环境。

在 Ubuntu 或者 MacOS 下，通常您需要安装如下工具：

- A. Python 2.6 -- 2.7, 您可以从 python.org 下载.
- B. GNU Make 3.81 -- 3.82, 您可以从 gnu.org 下载.
- C. Git 1.7 or newer. 您可以从 git-scm.com 下载.

3. 安装所需的安装包（基于 Ubuntu 14.04）

您需要一个 64 位版本的 Ubuntu，推荐使用 Ubuntu14.04。注意：使用老版本 Ubuntu 可能会有兼容性问题。用下面命令来安装 Ubuntu 所需的包：

```
$ sudo apt-get install git gnupg flex bison gperf build-essential \
zip tar curl libc6-dev libncurses5-dev:i386 x11proto-core-dev \
libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
libgl1-mesa-dev g++-multilib mingw32 cmake tofrodos \
python-markdown libxml2-utils xsltproc zlib1g-dev:i386 lzop
```

```
$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gn
u/libGL.so
```

4. 配置 USB 接入

在 GUN/Linux 系统中（特别是在 Ubuntu 系统中），在默认配置下用户不能直接接入 USB 设备。系统需要配置为可允许接入。推荐方法：创建一个/etc/udev/rules.d/51-android.rules (使用 root 用户)，然后复制下列内容到文件中。 <username> 必须是经授权的用 USB 连接设备的用户的实际用户名。

```
# adb protocol on passion (Rockchip products)SUBSYSTEM=="usb",
ATTR{idVendor}=="2207",ATTR{idProduct}=="0010",MODE="0600",
OWNER="<username>"
```

新的配置将在下一次设备接入时生效，因此必须重新拔插设备。本方法支持 Ubuntu Hardy Heron (8.04.x LTS) 以及 Lucid Lynx(10.04.x LTS)。其他版本的 Ubuntu 或者其他类型的 GNU/Linux 可能需要不同的配置。

请参考：<http://source.android.com/source/initializing.html>

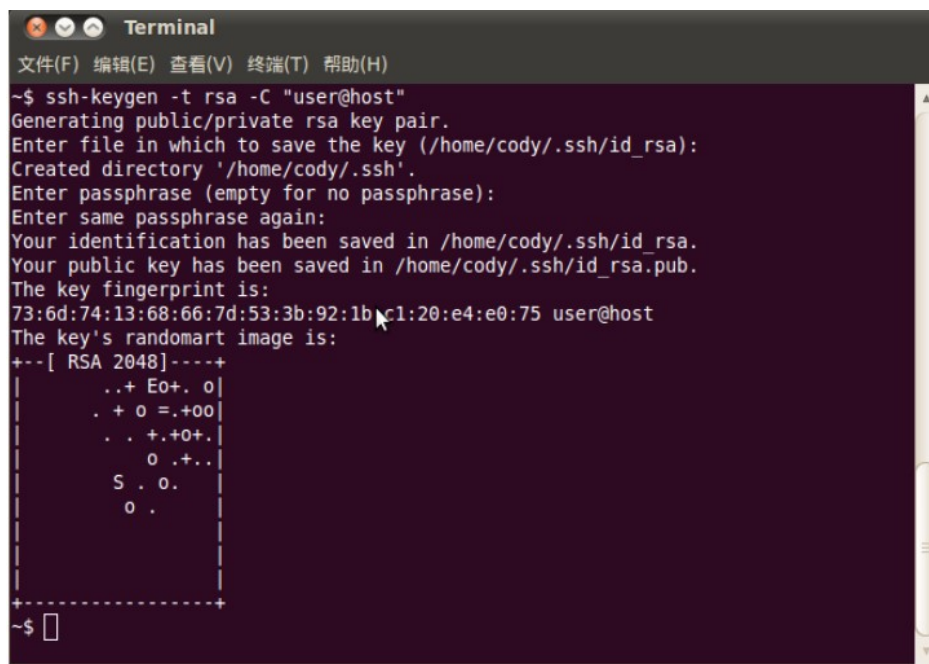
附录 B SSH 公钥操作说明

附录 B-1 SSH 公钥生成

使用如下命令生成：

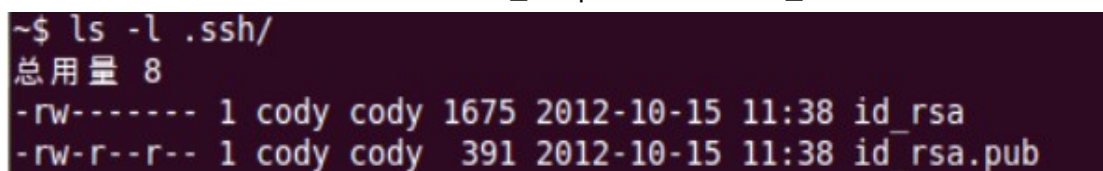
```
ssh-keygen -t rsa -C "user@host"
```

请将 **user@host** 替换成您的邮箱地址。



```
Terminal
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
~$ ssh-keygen -t rsa -C "user@host"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cody/.ssh/id_rsa):
Created directory '/home/cody/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cody/.ssh/id_rsa.
Your public key has been saved in /home/cody/.ssh/id_rsa.pub.
The key fingerprint is:
73:6d:74:13:68:66:7d:53:3b:92:1b:c1:20:e4:e0:75 user@host
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .+ Eo+. o |
|    . + 0 =. +00 |
|   . . +. +0+. |
|      o .+. |
|     S . 0. |
|      o . |
+-----+
~$
```

命令运行完成会在您的目录下生成公钥（id_rsa.pub）和私钥（id_rsa）。



```
~$ ls -l .ssh/
总用量 8
-rw----- 1 cody cody 1675 2012-10-15 11:38 id_rsa
-rw-r--r-- 1 cody cody 391 2012-10-15 11:38 id_rsa.pub
```

请妥善保管生成的私钥文件 **id_rsa** 和密码，并将公钥 **id_rsa.pub** 发邮件给 SDK 发布服务器的管理员。

附录 B-2 Git 权限申请说明

参考上述章节，生成公钥文件，发邮件至 fae@rock-chips.com，申请开通 SDK 代码下载权限。