Security Classification:

Most Confidential ( )   Confidential ( )   Internal ( )   Public (√)

RK3399_Linux_Buildroot_V1.2

Development Guide

（Technical Department）

Mark：	Version：	V1.2
[ ] Editing [√] Issued	Author：	Huang Guochun
	Completed Date：	2017-06-08

福州瑞芯微电子股份有限公司

Fuzhou   Rockchips   Semiconductor   Co . , Ltd

Revision History

Date	Version No.	Revision Note	修改人	核定人
2017-01-16	V1.0	Initial version	HGC	LBY
2017-02-27	V1.1	Add Linux PC download tool	HGC	LBY
2017-06-08	V1.2	U-boot release branch	HGC	LBY

Content

1 Overview

This SDK is based on linux 64 bit system with kernel 4.40. It is applicable to the development of RK3399 excavator V12 and all other linux products based on it.

This SDK supports VPU hardware decoding, GPU 3D, Wayland display, QT and other functions. For specific function debugging and interface instructions, please refer to related documents under the project's docs/ directory.

2 Main Functions

Function	Module name
data communication	Wi-Fi、ethernet card、usb、sdcard
application	audio, video playback

3 SDK Acquisition

The SDK is released via Github. The setup instruction of compilation and development environment, as well as the development guide are published through the Rockchip opensource website.

http://opensource.rock-chips.com/wiki_Main_Page

3.1 Github Address

All source code of SDK will be published and updated here:

https://github.com/rockchip-linux

3.2 Opensource Address

Rockchip publishes reference documentations via the open source wiki website, including TRM, Datasheet, Schematic & Layout Guide, Development environment setup,

Compilation configuration and Command instructions.

http://opensource.rock-chips.com/

3.3 Repo Installation

sudo apt-get install repo

3.4 SDK Download and Sync

Use step 3.3 to obtain repo for initialization. The download address of RK3399_Linux
is as follows:

repo init --repo-url=https://github.com/rockchip-linux/repo -u https://github.com/rockchip-linux/manifests -b buildroot

And then execute the command below to download the entire project code in the
current directory :

repo sync

4 SDK Compilation Instruction

Below are install commands of software package needed for compiling environment
setup:

sudo apt-get install git-core gitk git-gui gcc-arm-linux-gnueabihf u-boot-tools

device-tree-compiler gcc-aarch64-linux-gnu mtools parted libudev-dev libusb-1.0-0-dev

4.1 U-boot Compilation

Execute mk-uboot.sh in the project root directory to get idbloader.img、trust.img、
uboot.img、rk3399_loader_v1.08.106.bin:

rk3399-sapphire-excavator development board：

./build/mk-uboot.sh rk3399-excavator

rk3399 Firfly development board：

./build/mk-uboot.sh firefly-rk3399

Compiled files will be copied to out / u-boot which is under the project root directory：

u-boot/

├── idbloader.img

```
├── rk3399_loader_v1.08.106.bin
├── trust.img
└── uboot.img
```

4.2 Kernel Compilation

Execute the following command to automatically compile and package Kernel in the project root directory:

rk3399-sapphire-excavator development board：

```
./build/mk-kernel.sh rk3399-excavator
```

rk3399 Firfly development board：

```
./build/mk-kernel.sh firefly-rk3399
```

The generated boot.img after compilation will be copied to the out/ in the project root directory.

```
out/
├── boot.img
├── kernel
│       ├── Image
│       └── rk3399-sapphire-excavator-linux.dtb
```

4.3 Rootfs System and App Compilation

```
cd buildroot &&  make rockchip_rk3399_defconfig && cd .. &&   ./build_all.sh
&& ./mkfirmware.sh
```

After executing the command ./mkfirmware.sh, it will pack rootfs.img to the out/ directory.

Note：

If want to compile a single module or a third-party application, you need to configure the cross-compilation environment.

The cross-compilation tool is located in the buildroot/output/host/usr directory, you need to set the bin/ and aarch64-rockchip-linux-gnueabihf/bin/ directories as the environmental variables, and then execute the script which will automatically configure the environmental variables in the top-level directory.(only valid for the current console):

source envsetup.sh

Enter the command to view：

aarch64-**linux-gcc --version**

The following log will be printed for the successful configuration:

aarch64-linux-gcc.br_real (Buildroot 2016.08.1-00150-gc031b95) 5.4.0

System compilation

Execute "./build_all.sh", which will automatically find "rk_make_first.sh" and "rk_make.sh" in the system, and then execute the compilation command. If only need to compile a single module, you can execute the command of "rk_make.sh" or "rk_make_first.sh" in the specific module directory. If need to add an application module, you can compile your own application by adding "rk_make.sh" or "rk_make_first.sh".

5 Download Instruction
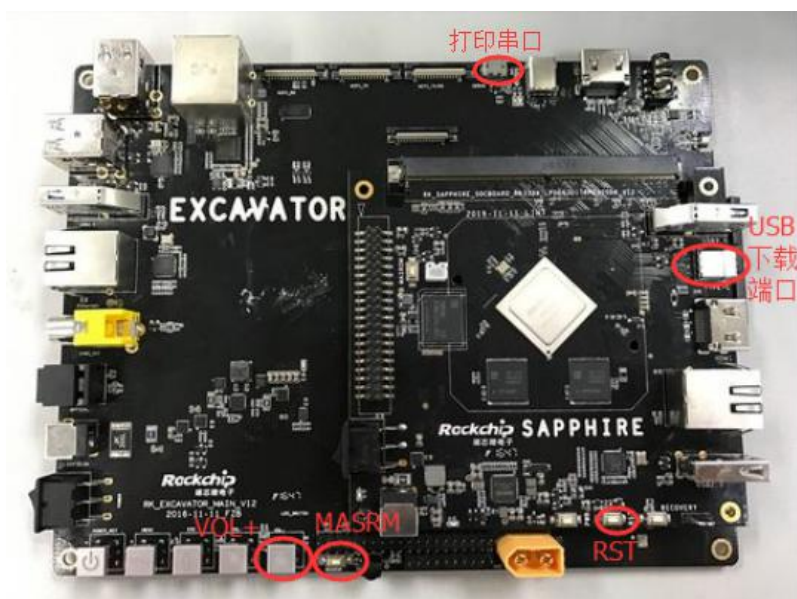
Pic1 RK3399 Excavator Board

5.1 Windows Download Instruction

SDK provides windows flashing tools which are located in the project root directory:

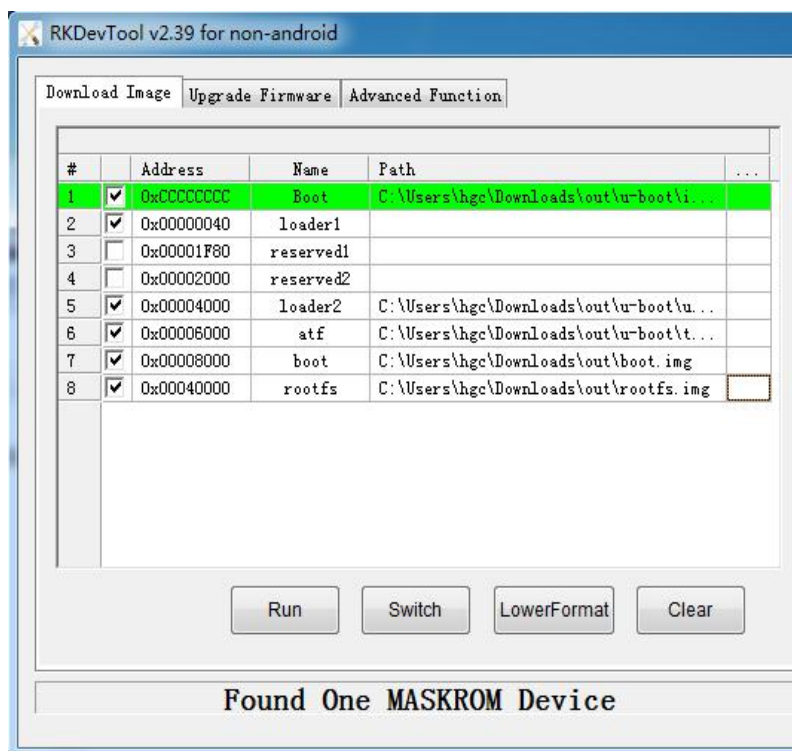rkbin/tools/

├── AndroidTool_Release_v2.39

As shown below, the device needs to enter the MASKROM mode for the first time flashing. After connect the usb cable, long press the button "MSROM", and press the reset button "RST" and then release, the device will enter the MASKROM Mode. Then you should load the paths of the corresponding images and click the "Run" for flashing.

Partition offset and downloading files

Address	Name		Path
	boot	--	out/u-boot/rk3399_loader_v1.08.106.bin
0x40	loader1	--	out/u-boot/idbloader.img
0x4000	loader2	--	out/u-boot/uboot.img
0x6000	atf	--	out/u-boot/trust.img
0x8000	boot	--	out/boot.img
0x40000	rootfs	--	out/rootfs.img

Please refer to the following URL for more details about the partition information:

http://opensource.rock-chips.com/wiki_Partitions

Pic2 Flashing Tool AndroidTool.exe

Note：Before flashing, need to install the latest USB driver, which is in：

rkbin/tool/

DriverAssitant_v4.5

5.2 Linux Download Instruction

The flashing tool rkdeveloptool is located in rkbin/tools/ directory. Please make sure the board is connected to maskrom rockusb. The flashing commands are as below:

sudo rkdeveloptool db rk3399_loader_v1.08.106.bin

sudo rkdeveloptool wl 0x40 idbloader.img

sudo rkdeveloptool wl 0x4000 uboot.img

sudo rkdeveloptool wl 0x6000 trust.img

sudo rkdeveloptool wl 0x8000 boot.img

sudo rkdeveloptool wl 0x40000 rootfs.img

sudo rkdeveloptool rd

After flashing successfully with two ways above, press the "Enter" key using the serial port to enter the command line configuration mode after reboot. Use the following command to write the gpt partition table into the decive, and then the system will reboot and load the rootfs.

gpt write mmc 0 $partitions

boot

6 Secure CRT Parameters Setting

If want to use the secure CRT software to print debugging information log, need to set the serial port parameters detailed as below:
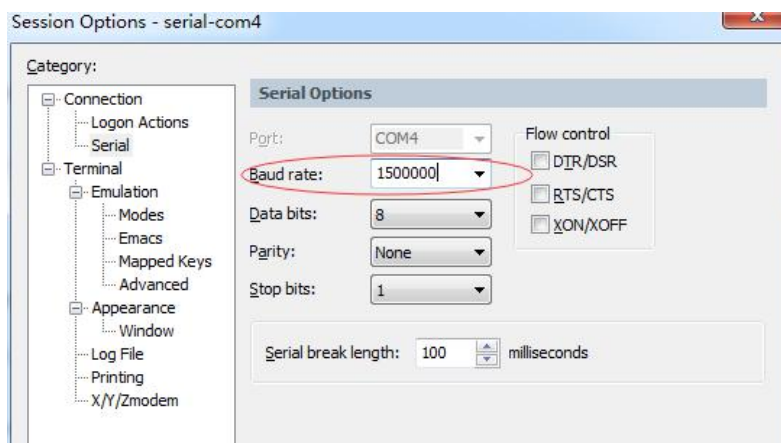
Pic3 Secure CRT Parameters Setting

7 RK3399_Linux Project Directory Introduction

There are buildroot, app, kernel, uboot, device, docs, external and other directories in the project. Each directory or its sub-directories will correspond to a git project, and the submission should be done in the respective directory.

1）buildroot：customize the root file system;

2）app：deposit the upper application app, mainly Carmachin and some test applications;

3）external：related libraries, including audio, video, network and so on;

4）kernel：kernel source code;

5）device/rockchip/rk3399：deposit the boot-up initialization script，third-party libraries, bin, alsa/wifi and other configuration files, as well as compiling scripts. Several scripts of the system root directory are all copied from here while repo sync. So if want to submit the modified script, it must be done in the device/rockchip/rk3399 directory.

build_all.sh: compile all third party libraries and applications;

mkfirmware.sh: pack the final firmware;

envsetup.sh: terminal environment variable setup;

6）docs: deposit the project help file;

7）out: deposit the output firmware;

8）rkbin: deposit firmware and tools;

9）build: deposit the compilation script;