RV1126/RV1109 Linux SDK Quick Start

ID: RK-JC-YF-360

Release Version: V1.1.0

Release Date: 2020-06-08

Security Level: □Top-Secret □Secret □Internal ■Public

DISCLAIMER

THIS DOCUMENT IS PROVIDED “AS IS”. ROCKCHIP ELECTRONICS CO., LTD.(“ROCKCHIP”)DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS,MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip’s registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

The document presents the basic usage of Rockchip RV1126/RV1109 Linux SDK, aiming to help engineers get started with RV1126/RV1109 Linux SDK faster. After the SDK is downloaded, you can check docs/RV1126_RV1109/RV1126_RV1109_Release_Note.txt to confirm the current SDK version.

Product Version

Chipset	Kernel Version
RV1126/RV1109	Linux 4.19

Intended Audience

This document (this guide) is mainly intended for:

- Technical support engineers
- Software development engineers

Revision History

Version	Author	Date	Revision History
V0.0.1	CWW	2020-04-28	Initial version
V0.0.2	CWW	2020-05-09	Update the interface of RK IPCamera Tool
V0.0.3	CWW	2020-05-20	Add libssl-dev and expect for building environment
V1.0.0	CWW	2020-05-25	1. update chapter 3 & 4.4 & 4.5 2. add fast boot compile guide 3. add chapter 5.4
V1.1.0	CWW	2020-06-08	1. update company name 2. update document style 3. update chapter 2
V1.1.1	CWW	2020-06-16	fix company name

Contents

RV1126/RV1109 Linux SDK Quick Start

- 1 Set up an Development Environment
- 2 SDK Configuration Framework Introduction
  - 2.1 SDK Project Directory Introduction
  - 2.2 SDK Configuration Framework
- 3 SDK Building Introduction
  - 3.1 To Select Board Configure
  - 3.2 To View Building Commands
  - 3.3 U-Boot Building
  - 3.4 Kernel Building
  - 3.5 Recovery Building
  - 3.6 Rootfs Building
  - 3.7 Firmware Package
  - 3.8 Full Automatic Building
- 4 Upgrade Introduction
  - 4.1 TOP Surface of the EVB
  - 4.2 Bottom Surface of the EVB
  - 4.3 EVB Function Table
  - 4.4 Windows Upgrade Introduction
  - 4.5 Linux Upgrade Introduction
- 5 EVB Function Introduction
  - 5.1 How to Access 3 RTSP and 1 RTMP Network Stream
    - 5.1.1 Get Device IP Address by Serial Port or ADB of the EVB
    - 5.1.2 Get Device IP Address by RK IPCamera Tool
    - 5.1.3 Access Network Stream
  - 5.2 How to Access Device Information via Web
  - 5.3 How to Test Face Recognition Function
  - 5.4 How to Debug With EVB via Network
    - 5.4.1 Debug With SSH
    - 5.4.2 Debug With SCP

1 Set up an Development Environment

Ubuntu 16.04 system: Please install software packages with below commands to set up a building environment:

```
sudo apt-get install repo 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 python-linaro-image-tools linaro-image-tools autoconf
autotools-dev libsigsegv2 m4 intltool libdrm-dev curl sed make binutils build-
essential gcc g++ bash patch gzip gawk bzip2 perl tar cpio python unzip rsync
file bc wget libncurses5 libqt4-dev libglib2.0-dev libgtk2.0-dev libglade2-dev
cvs git mercurial rsync openssh-client subversion asciidoc w3m dblatex graphviz
python-matplotlib libc6:i386 libssl-dev expect
```

Ubuntu 17.04 or later version system:

In addition to the above software packages, the following dependencies is needed:

```
sudo apt-get install lib32gcc-7-dev  g++-7  libstdc++-7-dev
```

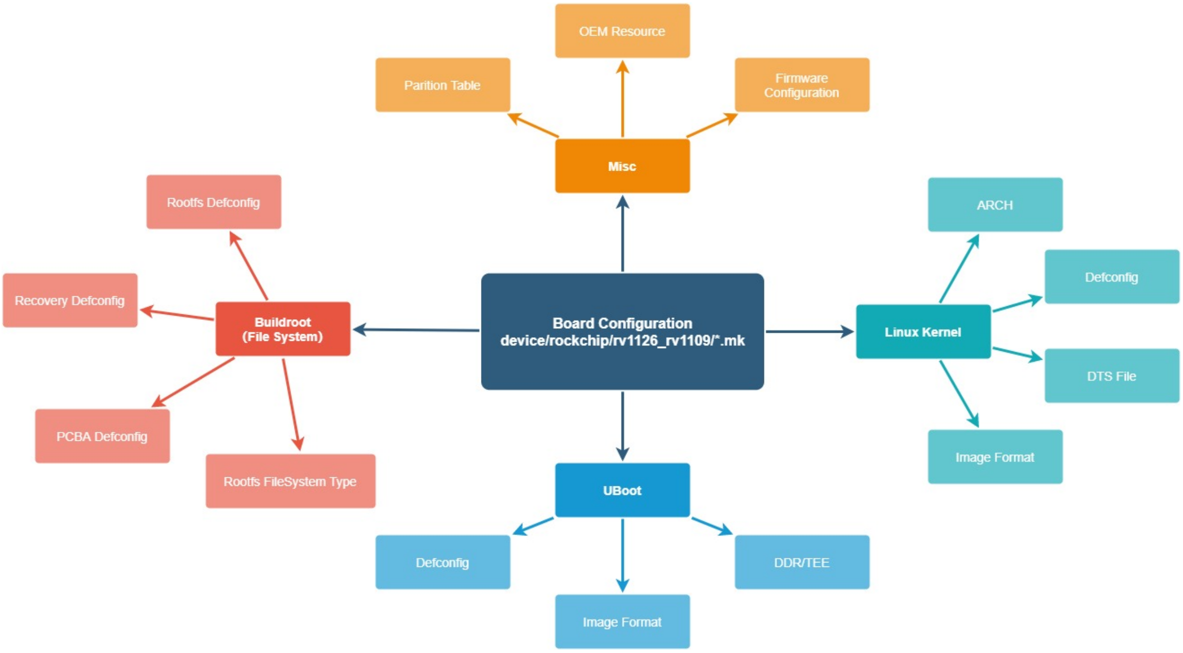
2 SDK Configuration Framework Introduction

2.1 SDK Project Directory Introduction

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

- buildroot: customized root file system.
- app: store applications.
- external: related libraries, including audio and video.
- kernel: kernel code.
- device/rockchip: stores some scripts and prepared files for building and packaging firmware of each chip.
- docs: stores development guides, platform support lists, tool usage, Linux development guides, and so on.
- prebuilts: stores cross-compilation toolchain.
- rkbin: stores firmware and tools.
- rockdev: stores building output firmware.
- tools: stores some commonly used tools.
- u-boot: U-Boot code.

2.2 SDK Configuration Framework



3 SDK Building Introduction

3.1 To Select Board Configure

SDK Download Address:

```
repo init --repo-url ssh://git@www.rockchip.com.cn/repo/rk/tools/repo -u
ssh://git@www.rockchip.com.cn/linux/rk/platform/manifests -b linux -m
rv1126_rv1109_linux_release.xml
```

Board Configuration	Comment
device/rockchip/rv1126_rv1109/BoardConfig.mk	General Version
device/rockchip/rv1126_rv1109/BoardConfig-tb.mk	Support Fast Boot

Command of select board configure:

```
### select general version board configuration
./build.sh device/rockchip/rv1126_rv1109/BoardConfig.mk

### select fast boot board configuration
./build.sh device/rockchip/rv1126_rv1109/BoardConfig-tb.mk
```

3.2 To View Building Commands

Execute the following command in the root directory: `./build.sh -h|help`

```
./build.sh help
Usage: build.sh [OPTIONS]
Available options:
BoardConfig*.mk    -switch to specified board config
uboot              -build uboot
spl                -build spl
kernel             -build kernel
modules            -build kernel modules
```

```
toolchain          -build toolchain
rootfs             -build default rootfs, currently build buildroot as default
buildroot          -build buildroot rootfs
ramboot            -build ramboot image
multi-npu_boot     -build boot image for multi-npu board
yocto              -build yocto rootfs
debian             -build debian9 stretch rootfs
distro             -build debian10 buster rootfs
pcba               -build pcba
recovery           -build recovery
all                -build uboot, kernel, rootfs, recovery image
cleanall           -clean uboot, kernel, rootfs, recovery
firmware           -pack all the image we need to boot up system
updateimg          -pack update image
otapackage         -pack ab update otapackage image
save               -save images, patches, commands used to debug
allsave            -build all & firmware & updateimg & save

Default option is 'allsave'.
```

To view detailed building commands for some modules, for example: `./build.sh -h kernel`

```
./build.sh -h kernel
###Current SDK Default [ kernel ] Build Command###
cd kernel
make ARCH=arm rv1126_defconfig
make ARCH=arm rv1126-evb-ddr3-v10.img -j12
```

3.3 U-Boot Building

```
### U-Boot building command
./build.sh uboot

### to view detailed U-Boot build command
./build.sh -h uboot
```

3.4 Kernel Building

```
### Kernel building command
./build.sh kernel

### to view detailed Kernel build command
./build.sh -h kernel
```

3.5 Recovery Building

```
### Recovery building command
./build.sh recovery

### to view detailed Recovery build command
./build.sh -h recovery
```

3.6 Rootfs Building

```
### Rootfs building command
./build.sh rootfs

###  to view detailed Rootfs build command
./build.sh -h rootfs
```

3.7 Firmware Package

Firmware packaging command: `./mkfirmware.sh`

Firmware directory: rockdev

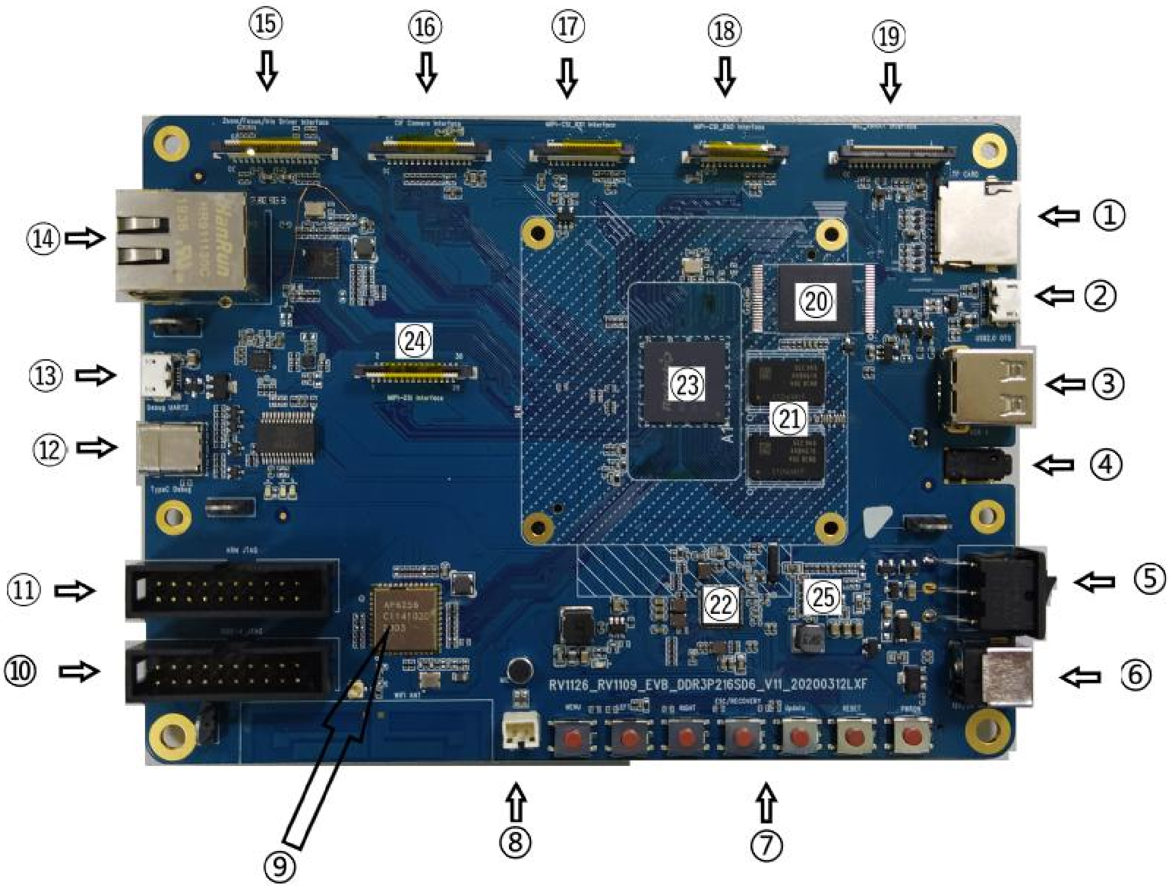
3.8 Full Automatic Building

Enter the project root directory and execute the following command to automatically complete all buildings:

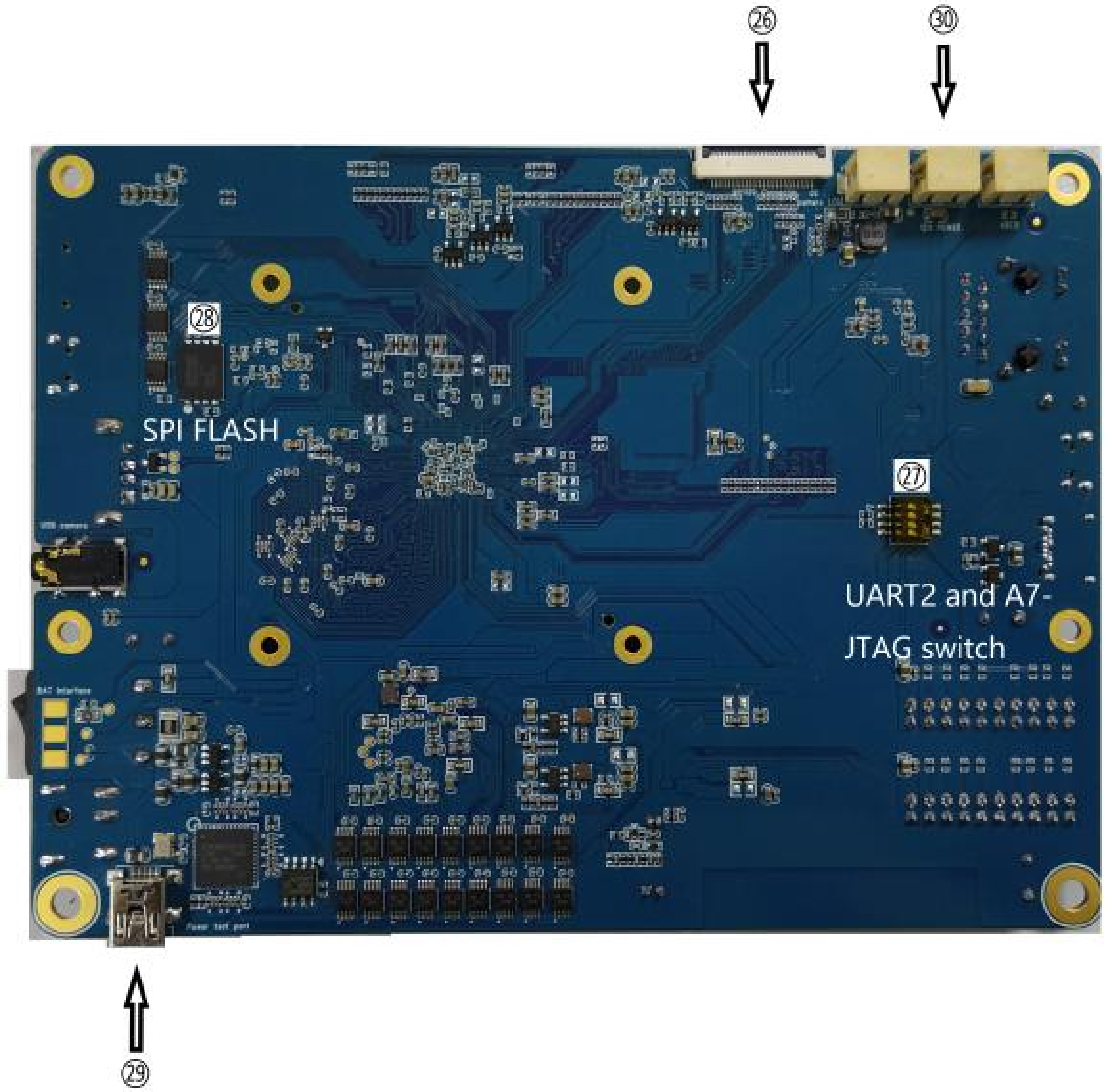
```
./build.sh all
```

4 Upgrade Introduction

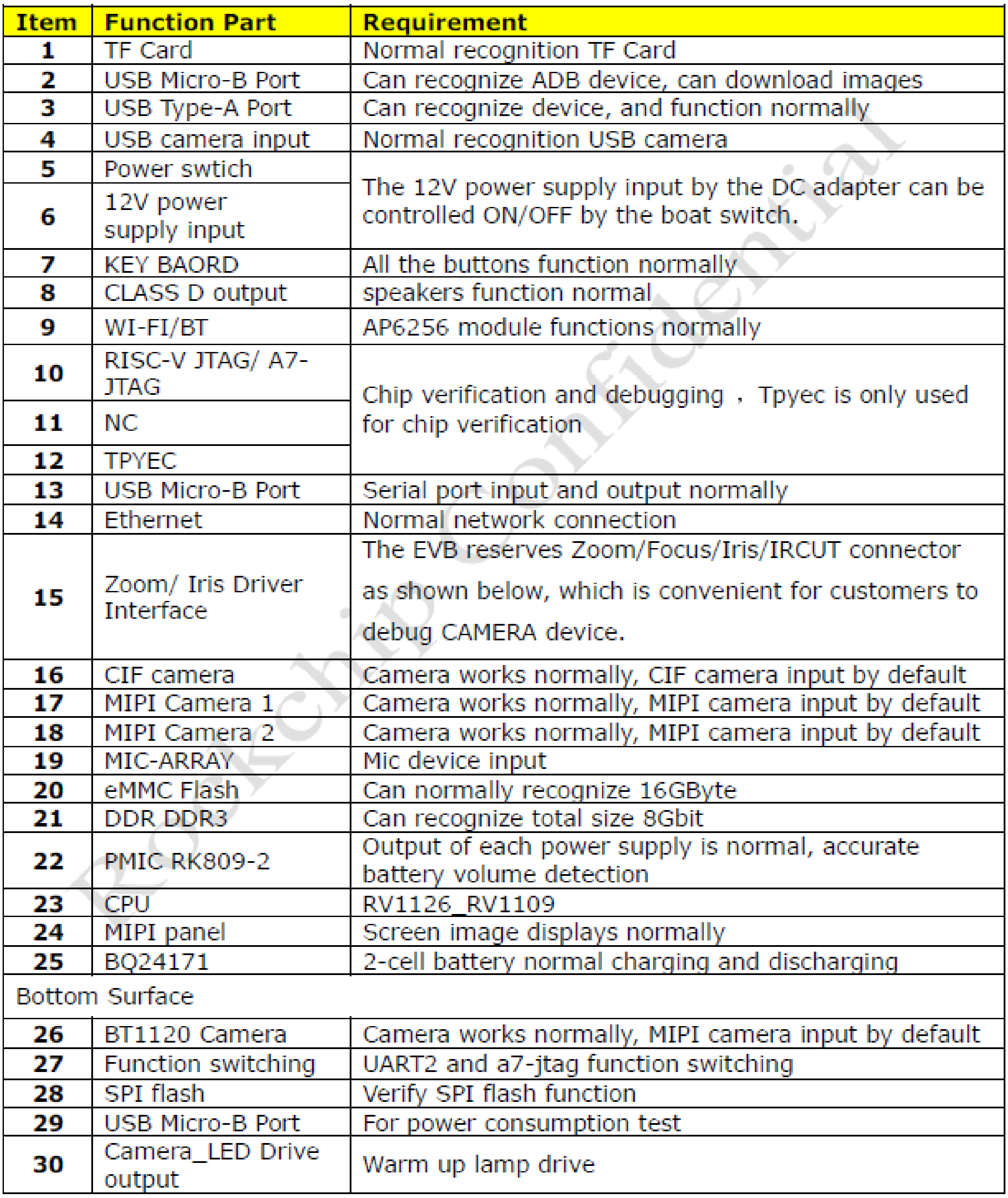
4.1 TOP Surface of the EVB



4.2 Bottom Surface of the EVB



4.3 EVB Function Table



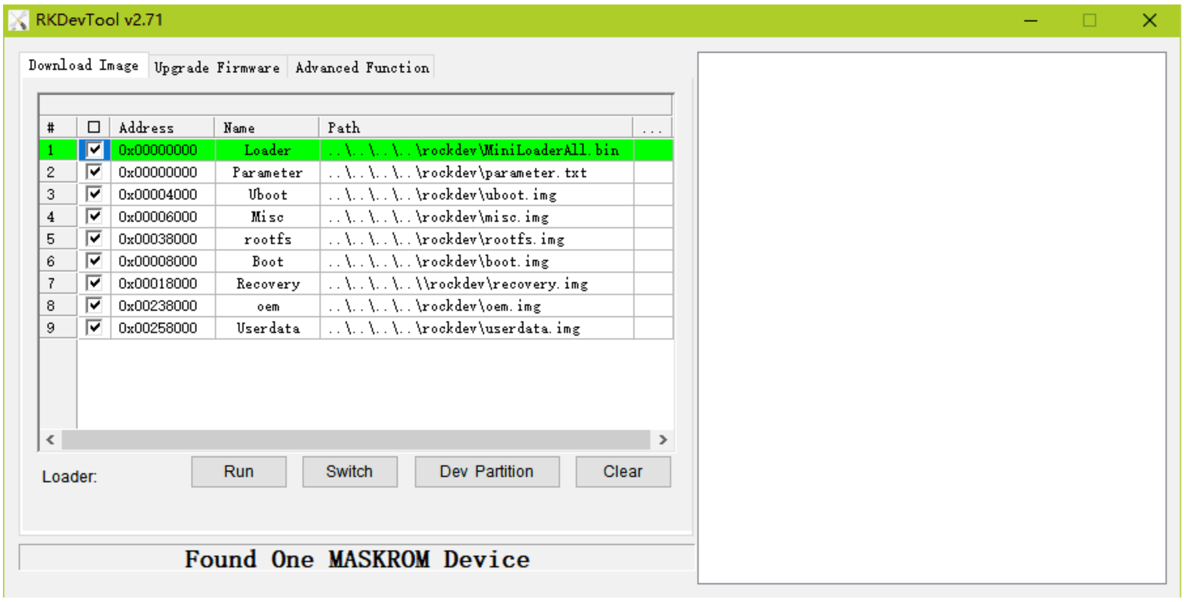
Item	Function Part	Requirement
1	TF Card	Normal recognition TF Card
2	USB Micro-B Port	Can recognize ADB device, can download images
3	USB Type-A Port	Can recognize device, and function normally
4	USB camera input	Normal recognition USB camera
5	Power swtich	The 12V power supply input by the DC adapter can be controlled ON/OFF by the boat switch.
6	12V power supply input	
7	KEY BAORD	All the buttons function normally
8	CLASS D output	speakers function normal
9	WI-FI/BT	AP6256 module functions normally
10	RISC-V JTAG/ A7-JTAG	Chip verification and debugging ， Tpyec is only used for chip verification
11	NC	
12	TPYEC	
13	USB Micro-B Port	Serial port input and output normally
14	Ethernet	Normal network connection
15	Zoom/ Iris Driver Interface	The EVB reserves Zoom/Focus/Iris/IRCUT connector as shown below, which is convenient for customers to debug CAMERA device.
16	CIF camera	Camera works normally, CIF camera input by default
17	MIPI Camera 1	Camera works normally, MIPI camera input by default
18	MIPI Camera 2	Camera works normally, MIPI camera input by default
19	MIC-ARRAY	Mic device input
20	eMMC Flash	Can normally recognize 16GByte
21	DDR DDR3	Can recognize total size 8Gbit
22	PMIC RK809-2	Output of each power supply is normal, accurate battery volume detection
23	CPU	RV1126_RV1109
24	MIPI panel	Screen image displays normally
25	BQ24171	2-cell battery normal charging and discharging
Bottom Surface		
26	BT1120 Camera	Camera works normally, MIPI camera input by default
27	Function switching	UART2 and a7-jtag function switching
28	SPI flash	Verify SPI flash function
29	USB Micro-B Port	For power consumption test
30	Camera_LED Drive output	Warm up lamp drive

4.4 Windows Upgrade Introduction

The SDK provides a windows flash tool (this tool should be V2.71 or later version) which is located in project root directory:

```
tools/
├── windows/AndroidTool
```

As shown below, after building and generating the firmware, device needs to enter MASKROM or BootROM mode for flashing. After connecting USB cable, long press the "Update" button and press "RESET" button at the same time and then release, device will enter MASKROM mode. Then you should load the paths of the corresponding images and click "Run" to start update. You can also press the "recovery" button and press “RESET" button "RESET" then release to enter loader mode to update. Partition offset and update files of MASKROM Mode are shown as follows (Note: you have to run the tool as an administrator in Windows PC):



#	□	Address	Name	Path	...
1	☑	0x00000000	Loader	..\..\..\..\rockdev\MiniLoaderAll.bin	
2	☑	0x00000000	Parameter	..\..\..\..\rockdev\parameter.txt	
3	☑	0x00004000	Uboot	..\..\..\..\rockdev\uboot.img	
4	☑	0x00006000	Misc	..\..\..\..\rockdev\misc.img	
5	☑	0x00038000	rootfs	..\..\..\..\rockdev\rootfs.img	
6	☑	0x00008000	Boot	..\..\..\..\rockdev\boot.img	
7	☑	0x00018000	Recovery	..\..\..\..\\rockdev\recovery.img	
8	☑	0x00238000	oem	..\..\..\..\rockdev\oem.img	
9	☑	0x00258000	Userdata	..\..\..\..\rockdev\userdata.img	

Found One MASKROM Device

Note:

1. In addition to MiniLoader All.bin and parameter.txt, the actual partition to be burned is based on rockdev / parameter.txt configuration.
2. before upgrade, please install the latest USB driver, which is in the below directory:

```
<SDK>/tools/windows/DriverAssitant_v4.91.zip
```

4.5 Linux Upgrade Introduction

The Linux upgrade tool (Linux_Upgrade_Tool should be v1.49 or later versions) is located in "tools/linux" directory. Please make sure your board is connected to MASKROM/loader rockusb, if the generated firmware is in rockdev directory, upgrade commands are as below:

```
### In addition to MiniLoader All.bin and parameter.txt, the actual partition to
be burned is based on rockdev / parameter.txt configuration.
sudo ./upgrade_tool ul rockdev/MiniLoaderAll.bin
sudo ./upgrade_tool di -p rockdev/parameter.txt
sudo ./upgrade_tool di -u rockdev/uboot.img
sudo ./upgrade_tool di -misc rockdev/misc.img
sudo ./upgrade_tool di -b rockdev/boot.img
sudo ./upgrade_tool di -recovery rockdev/recovery.img
sudo ./upgrade_tool di -oem rockdev/oem.img
sudo ./upgrade_tool di -rootfs rocdev/rootfs.img
sudo ./upgrade_tool di -userdata rockdev/userdata.img
sudo ./upgrade_tool rd
```

Or upgrade the whole update.img firmware after packaging:

```
sudo ./upgrade_tool uf rockdev/update.img
```

Or in root directory, run the following command on your device to upgrade in MASKROM state:

```
./rkflash.sh
```

5 EVB Function Introduction

The EVB supports the following functions:

- Support 3 RTSP and 1 RTMP network stream
- Support 1280x720 local screen display
- Support to save the main stream to the device
- Support access device from web
- Support face recognition

5.1 How to Access 3 RTSP and 1 RTMP Network Stream

Connect a network cable to the network port of the EVB, power on and start. It will obtain the IP address automatically by default.

5.1.1 Get Device IP Address by Serial Port or ADB of the EVB

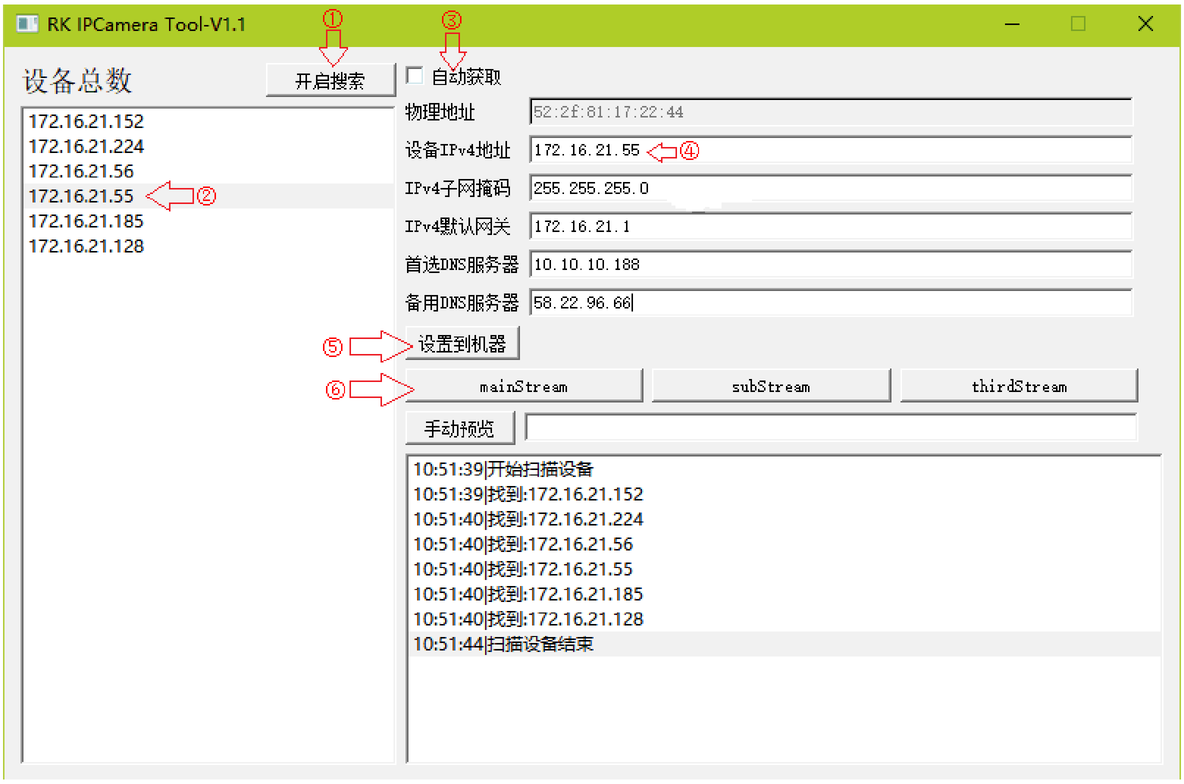
```
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:E0:F9:16:7E:E9
          inet addr:172.16.21.218  Bcast:172.16.21.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:199225 errors:0 dropped:2231 overruns:0 frame:0
          TX packets:372371 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20874811 (19.9 MiB)  TX bytes:522220899 (498.0 MiB)
          Interrupt:56
```

Connect to the EVB through the serial port, you have to configure as follows:

```
Baud rate: 1500000
Data bits: 8
Stop bit: 1
Parity: none
Flow control: none
```

5.1.2 Get Device IP Address by RK IPCamera Tool

Install the tool in the SDK directory `tools/windows/RK_IPCamera_Tool-V1.1.zip`. Open the tool and connect the EVB board to the computer through the network port. In the local area network, check the RK IPCamera Tool device list to obtain the device IP address.



172.16.21.152
172.16.21.224
172.16.21.56
172.16.21.55
172.16.21.185
172.16.21.128

□ 自动获取

物理地址 52:2f:81:17:22:44

设备IPv4地址 172.16.21.55

IPv4子网掩码 255.255.255.0

IPv4默认网关 172.16.21.1

首选DNS服务器 10.10.10.188

备用DNS服务器 58.22.96.66

```
10:51:39|开始扫描设备
10:51:39|找到:172.16.21.152
10:51:40|找到:172.16.21.224
10:51:40|找到:172.16.21.56
10:51:40|找到:172.16.21.55
10:51:40|找到:172.16.21.185
10:51:40|找到:172.16.21.128
10:51:44|扫描设备结束
```

Note:

- Step 1: click "开启搜索" to search devices
- Step 2: select a device
- Step 3: cancel "自动获取" and change to static IP
- Step 4: set a static IP
- Step 5: set the IP to device
- Step 6: open to preview

5.1.3 Access Network Stream

Use a player that supports RTSP or RTMP to access, for example (VLC player).

RTSP access address:

- rtsp://**IP address of the device**/live/mainstream
- rtsp://**IP address of the device**/live/substream
- rtsp://**IP address of the device**/live/thirdstream

RTMP access address:

- rtmp://**IP address of the device**:1935/live/substream

5.2 How to Access Device Information via Web

Open a web browser (Chrome browser is recommended) to access the address:

```
http://IP address of the device
```

For detailed operation instructions on the web, please refer to the documents under the SDK docs directory.

5.3 How to Test Face Recognition Function

Use a player to access RTSP main stream: rtsp://**IP address of the device**/live/mainstream.

The default authorization test time of the SDK's face recognition function is 30 ~ 60 minutes. When the authorization is invalid, the main stream preview will prompt "gace algorithm software is not authorized", and you have to restart to test again.

5.4 How to Debug With EVB via Network

5.4.1 Debug With SSH

Connect EVB with network, get EVB board's IP address with the Chapter 5.1.2 Get Device IP Address by RK IPCamera Tool. Ensure that the PC can ping the EVB board.

```
### Clean last login message (EVB IP address: 192.168.1.159)
ssh-keygen -f "$HOME/.ssh/known_hosts" -R 192.168.1.159
### Command of SSH
ssh root@192.168.1.159
### input the default passwd: rockchip
```

5.4.2 Debug With SCP

```
### Upload the test-file from PC to EVB board dirctory /userdata
scp test-file root@192.168.1.159:/userdata/
root@192.168.1.159's password:
### input the default passwd: rockchip

### Download the EVB file (/userdata/test-file) to PC
scp root@192.168.1.159:/userdata/test-file test-file
root@192.168.1.159's password:
### input the default passwd: rockchip
```