RV1126/RV1109 Linux SDK Quick Start

ID: RK-JC-YF-360

Release Version: V1.5.0

Release Date: 2020-08-07

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

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
V1.2.0	HJC	2020-06-22	Add smart usb camera product section
V1.2.1	CWW	2020-06-29	1. update chapter 4.4 2. Add install fakeroot for building environment
V1.3.0	CWW	2020-07-09	1. Add module path and document introduction 2. Add Build different Boards
V1.3.1	CWW	2020-07-15	1. Remove unused board config introduction 2. fix same content3. fix eMMC typo
V1.4.0	CWW	2020-07-16	1. Add ISP Tool RKISP2.x_Tuner introduction 2. Add Develop Tool introduction 3. Add board config introduction
V1.4.1	CWW	2020-07-17	1. Update chapter of SDK Building Introduction 2. Update Tool's Description
V1.5.0	CWW	2020-08-07	1. Update SDK board configure and compile instruction 2. Add install cmake for development environment

Contents

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 fakeroot cmake
```

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 RV1109/RV1126 Modules Dirctory Introduction

some of modules directory path	modules introduction
external/recovery	recovery
external/rkwifibt	Wi-Fi and BT
external/libdrm	DRM interface
external/rk_pcba_test	PCBA test code
external/isp2-ipc	Image signal processing server
external/mpp	encoding and decoding
external/rkmedia	Rockchip Multimedia interface
external/rkupdate	Rockchip upgrade code
external/camera_engine_rkaiq	Image processing algorithm module
external/rknpu	NPU driver
external/rockface	Face recognition code
external/CallFunIpc	Application interprocess communication code
external/common_algorithm	Audio and video general algorithm library
external/rknn-toolkit	Development kit for model transformation, reasoning and performance evaluation
app/libIPCProtocol	Based on dbus, provides a functional interface for inter-process communication
app/mediaserver	Main application providing multimedia services
app/ipc-daemon	System guard service
app/dbserver	Database service
app/netserver	Network services
app/storage_manager	Storage management service
app/ipcweb-backend	web backend
app/librkdb	Database interface
app/ipcweb-ng	Web front end, using Angular 8 framework

2.3 RV1109/RV1126 Develop Document

```
├── docs
│   ├── Linux
│   │   ├── ApplicationNote (Rockchip Application Framework Develop Introduction
and Web Develop Guide)
│   │   │   ├── Rockchip_Developer_Guide_Linux_Application_Framework_CN.pdf
```

```
│   │   │   ├── Rockchip_Instructions_Linux_MediaServer_CN.pdf
│   │   │   └── Rockchip_Instructions_Linux_Web_Configuration_CN.pdf
│   │   └── Multimedia (1. ISP Develop Guide 2. Multimedia Encoding and Decoding
Interface Develop Guide)
│   │       ├── camera
│   │       │   ├── Rockchip_Developer_Guide_ISP20_RkAiq_CN.pdf
│   │       │   ├── Rockchip_Instruction_Linux_Appliction_ISP20_CN.pdf
│   │       │   ├──
Rockchip_RV1109_RV1126_Developer_Guide_Linux_Ispserver_CN.pdf
│   │       │   └── Rockchip_User_Manual_Linux_ISP2_CN.pdf
│   │       ├── Rockchip_Developer_Guide_MPP_CN.pdf
│   │       ├── Rockchip_Developer_Guide_MPP_EN.pdf
│   │       └── Rockchip_Instructions_Linux_Rkmedia_CN.pdf
│   └── RV1126_RV1109 (1. Quick Start Guide 2. Hardware Develop Guide 3. SDK
Release Note 4. Encoding&Decoding Introduction)
│       ├── Rockchip_RV1126_RV1109_EVB_User_Guide_V1.0_CN.pdf
│       ├── Rockchip_RV1126_RV1109_EVB_User_Guide_V1.0_EN.pdf
│       ├── Rockchip_RV1126_RV1109_Linux_SDK_V1.0.0_20200616_CN.pdf
│       ├── Rockchip_RV1126_RV1109_Linux_SDK_V1.0.0_20200616_EN.pdf
│       ├── Rockchip_RV1126_RV1109_Quick_Start_Linux_CN.pdf
│       ├── Rockchip_RV1126_RV1109_Quick_Start_Linux_EN.pdf
│       ├── RV1109 Multimedia Codec Benchmark v1.2.pdf
│       └── RV1126 Multimedia Codec Benchmark v1.1.pdf
├── external
│   ├── rknn-toolkit (Development kit for model transformation,
│   │   │                 reasoning and performance evaluation)
│   │   └── doc
│   │       ├── Rockchip_Developer_Guide_RKNN_Toolkit_Custom_OP_V1.3.2_CN.pdf
│   │       ├── Rockchip_Developer_Guide_RKNN_Toolkit_Custom_OP_V1.3.2_EN.pdf
│   │       ├── Rockchip_Quick_Start_RKNN_Toolkit_V1.3.2_CN.pdf
│   │       ├── Rockchip_Quick_Start_RKNN_Toolkit_V1.3.2_EN.pdf
│   │       ├── Rockchip_Trouble_Shooting_RKNN_Toolkit_V1.3.2_CN.pdf
│   │       ├── Rockchip_Trouble_Shooting_RKNN_Toolkit_V1.3.2_EN.pdf
│   │       ├── Rockchip_User_Guide_RKNN_Toolkit_V1.3.2_CN.pdf
│   │       ├── Rockchip_User_Guide_RKNN_Toolkit_V1.3.2_EN.pdf
│   │       ├── Rockchip_User_Guide_RKNN_Toolkit_Visualization_V1.3.2_CN.pdf
│   │       └── Rockchip_User_Guide_RKNN_Toolkit_Visualization_V1.3.2_EN.pdf
│   └── rknpu
│       └── rknn (Rockchip NPU Develop Guide)
│           └── doc
│               ├── Rockchip_User_Guide_RKNN_API_V1.3.3_CN.pdf
│               └── Rockchip_User_Guide_RKNN_API_V1.3.3_EN.pdf
└── tools
    └── windows
        └── RKISP2.x_Tuner (ISP Tool)
            └── RKISP2.x_Tuner_User_Manual_v1.0.pdf
```

2.4 RV1109/RV1126 Develop Tools

2.4.1 Windows Tools

Release Note: tools/windows/ToolsRelease.txt

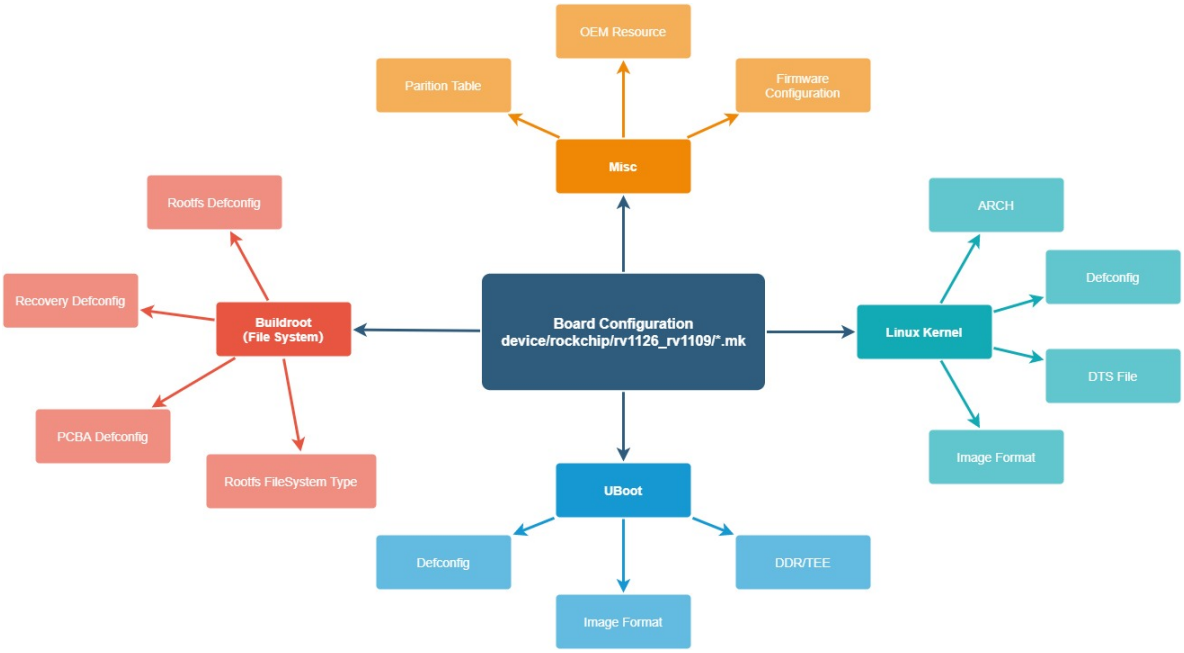
Tool's Name	Tool's Description
RKDevTool	discrete firmware upgrade and the entire update.img firmware upgrade tool
FactoryTool	factory production upgrade tool
SecureBootTool	firmware signing tool
efuseTool	efuse flash tool
RKDevInfoWriteTool	serial number burning tool
SDDiskTool	sd card firmware create tool
SpiImageTools	burner upgrade tool
DriverAssitant	rockchip pc driver for upgrade tool
RKImageMaker	firmware package tool (generate update.img)
SpeakerPCBATool	soundbox PCBA test tool
RKDevTool_Release	rockchip firmware flash tool
ParameterTool	rockchip partition modify tool
RKISP2.x_Tuner	ISP tool
RK_IPCamera_Tool	rockchip ipc camera search tool

2.4.2 Linux Tools

Release Note: tools/linux/ToolsRelease.txt

Tool's Name	Tool's Description
Linux_Pack_Firmware	firmware package tool (generate update.img)
Linux_Upgrade_Tool	rockchip firmware flash tool
Linux_SecureBoot	firmware signing tool
Firmware_Merger	SPI NOR firmware package tool (generate firmware.img for burner)

2.5 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
```

Chip Name	Board Configuration (path: device/rockchip/rv1126_rv1109)	Storage Medium	EVB Board name	Support Thunder Boot
RV1126/RV1109	BoardConfig-38x38-spi-nand.mk	SPI NAND	RV1126_RV1109_38X38_SPI_DDR3P216DD6_V10_20200511LXF	NO
RV1126/RV1109	BoardConfig-robot.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V13_20200630LXF	NO
RV1126/RV1109	BoardConfig-tb-v12.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	YES
RV1126/RV1109	BoardConfig-tb-v13.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V13_20200630LXF	YES
RV1126/RV1109	BoardConfig-spi-nand.mk	SPI NAND	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	NO
RV1126/RV1109	BoardConfig.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V13_20200630LXF	NO
RV1126/RV1109	BoardConfig-v12.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	NO
RV1126/RV1109	BoardConfig-v10-v11.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V11_20200312LXF	NO
RV1126/RV1109	BoardConfig-facial_gate.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	NO

Command of select board configure:

Method 1 ./build.sh "path to board config file", for example:

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

### select fast boot board configuration, apply to EVB Board
RV1126_RV1109_EVB_DDR3P216SD6_V13_20200630LXF
./build.sh device/rockchip/rv1126_rv1109/BoardConfig-tb-v13.mk
```

Method 2

```
./build.sh lunch
processing board option: lunch
processing option: lunch

You're building on Linux
Lunch menu...pick a combo:

0. default BoardConfig.mk
1. BoardConfig-38x38-spi-nand.mk
2. BoardConfig-facial_gate.mk
3. BoardConfig-ramboot-uvc.mk
4. BoardConfig-robot.mk
5. BoardConfig-sl.mk
6. BoardConfig-spi-nand.mk
7. BoardConfig-tb-v12.mk
8. BoardConfig-tb-v13.mk
9. BoardConfig-uvcc.mk
10. BoardConfig-v10-v11.mk
11. BoardConfig-v12.mk
12. BoardConfig.mk
Which would you like? [0]:
switching to board:
/home/cww/rv1109/device/rockchip/rv1126_rv1109/BoardConfig.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
lunch              -list current SDK boards and switch to specified board config
uboot              -build uboot
spl                -build spl
loader             -build loader
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

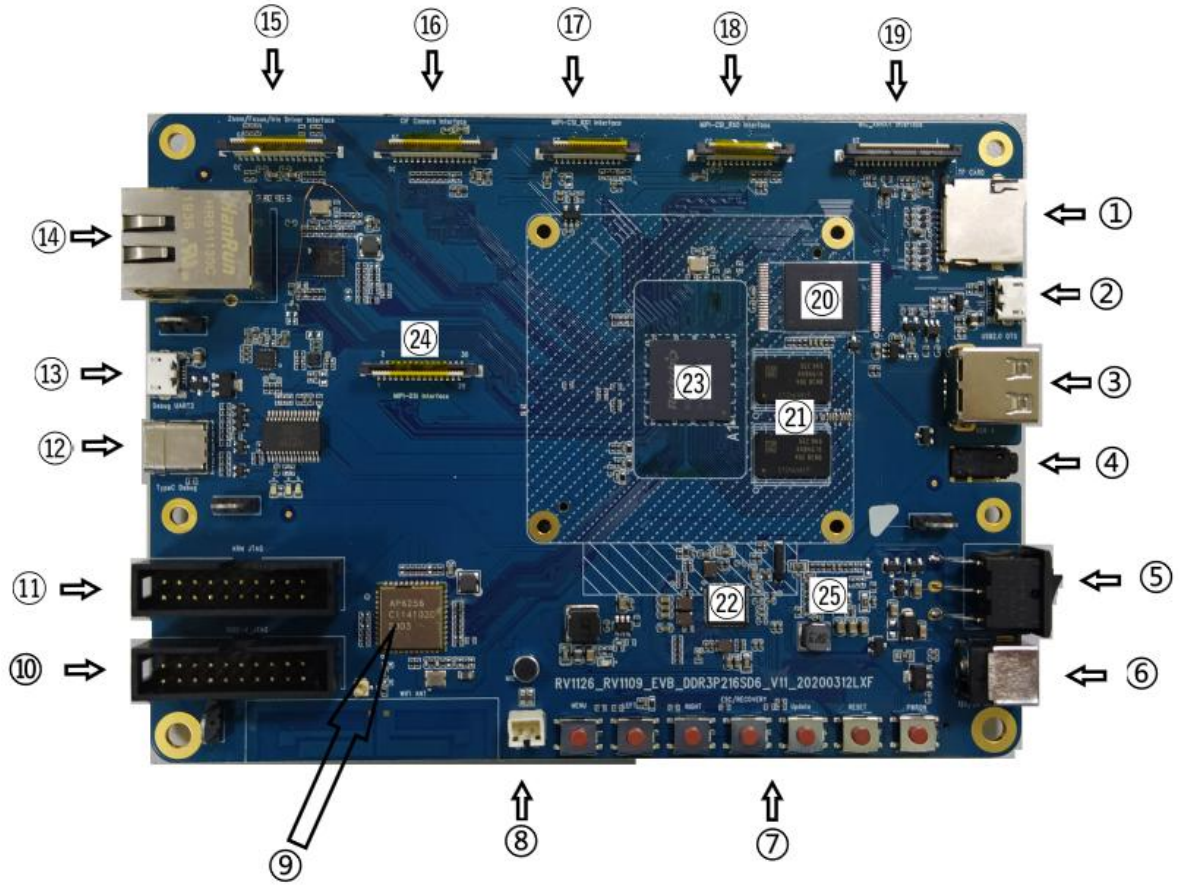
3.8 Full Automatic Building

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

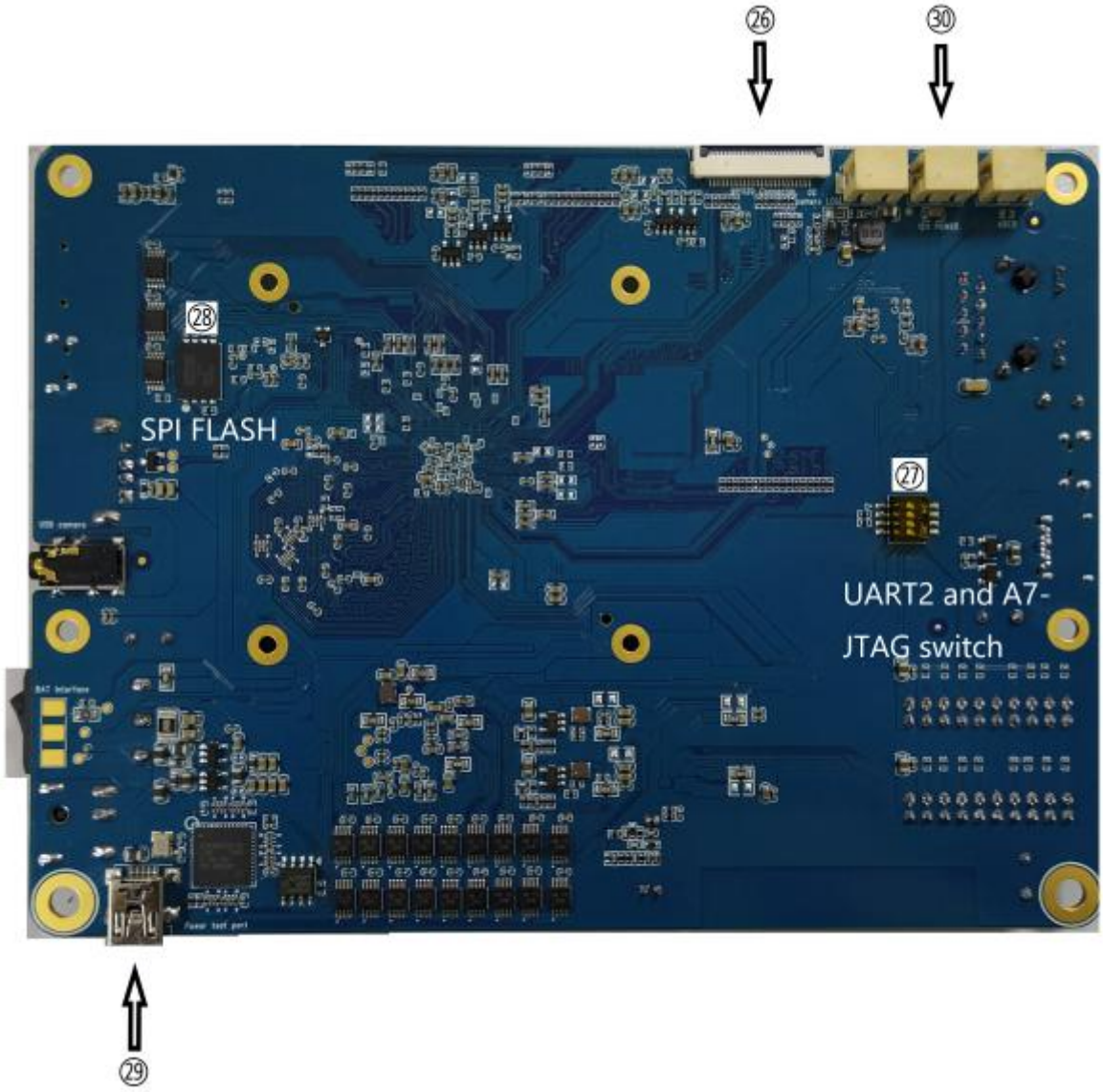
```
./build.sh all
```

4. Upgrade Introduction

4.1 TOP Surface of the EVB



4.2 Bottom Surface of the EVB



4.3 EVB Function Table

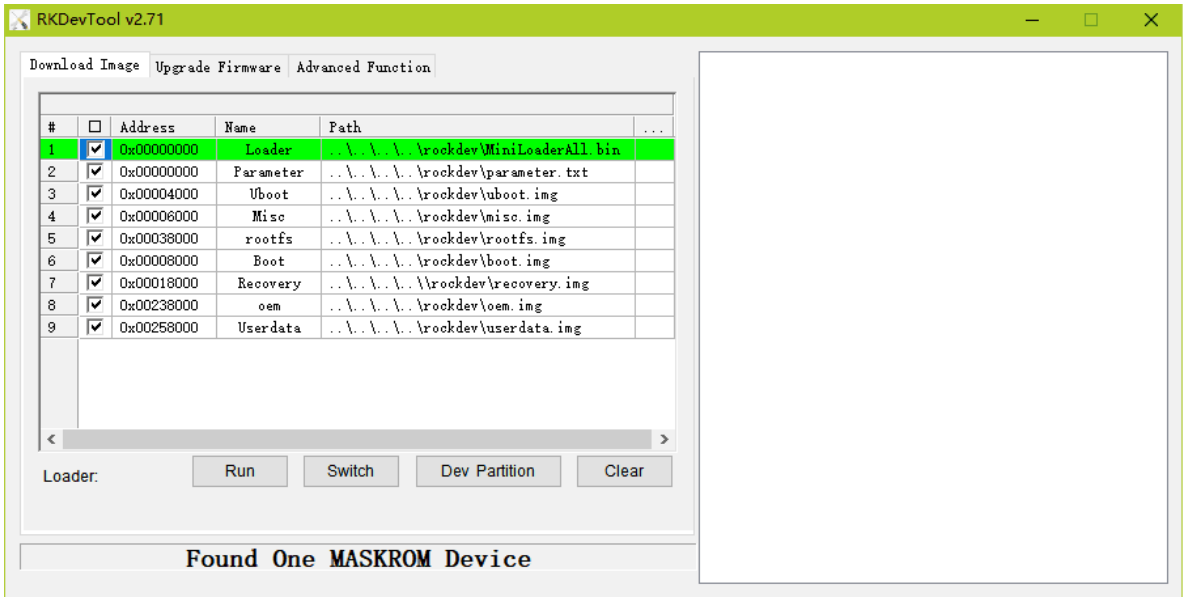
Item	Function Part	Requirement
1	TF Card	Normal recognition TF Card
2	USB Micro-B Port	Can recognize ADB device, can download images
3	USB Type-A Port	Can recognize device, and function normally
4	USB camera input	Normal recognition USB camera
5	Power swtich	The 12V power supply input by the DC adapter 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

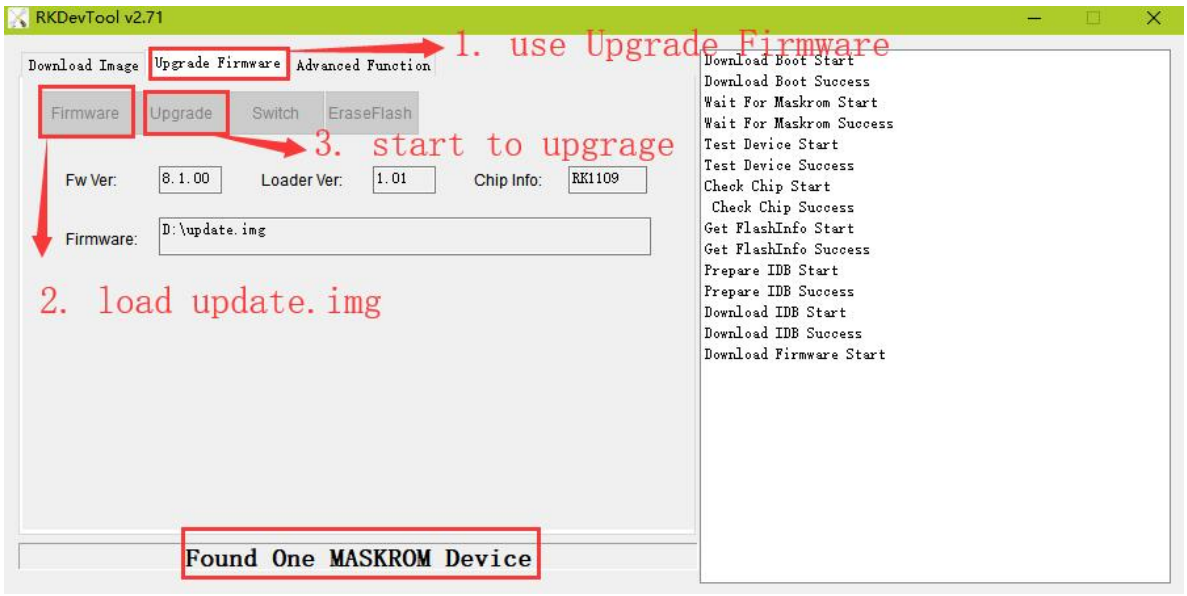
The SDK provides a windows flash tool (this tool should be V2.71 or later version) which is located in project root directory:

```
tools/
├── windows/RKDevTool
```

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):



The method to flash update.img:



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

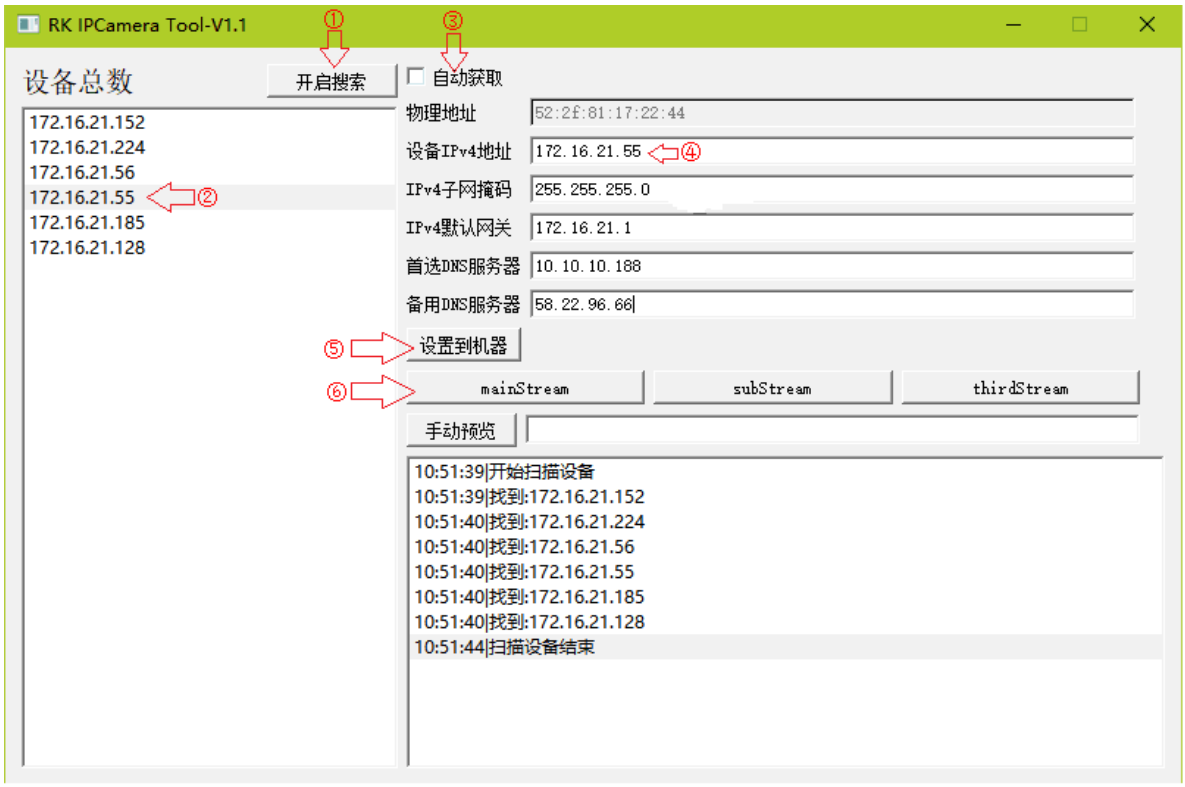
```
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:E0:F9:16:7E:E9
          inet addr:172.16.21.218  Bcast:172.16.21.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:199225 errors:0 dropped:2231 overruns:0 frame:0
          TX packets:372371 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.



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 which named Rockchip_Instructions_Linux_Web_Configuration_CN.pdf.

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
```

6. Smart USB Camera Product

The smart USB camera product supports the following functions:

- Support standard UVC camera function, up to 4K preview (RV1126)
- Support a variety of NN algorithms, including face detection, human posture or bone detection, face key point detection and tracking, and support third-party algorithm expansion
- Support USB composite device stable transmission (RNDIS/ UAC / ADB, etc.)
- Support NN preprocessing and data post-processing path
- Support preview of multiple terminal devices such as smart TV or PC
- Support for eptz function

6.1 Product Building Introduction

The compilation and configuration of the smart USB camera product is based on the public SDK and adopts the separate rv1126_rv1109_linux_Ai_camera_release.xml Code list management update.

6.1.1 Select Board Configuration for USB Camera Product

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_ai_camera_release.xml
```

Board Configuration	Comment
device/rockchip/rv1126_rv1109/BoardConfig-uvcc.mk	Smart USB Camera board config

Chip Name	Board Configuration (path: device/rockchip/rv1126_rv1109)	Storage Medium	EVB Board name
RV1126/RV1109	BoardConfig-uvcc.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V13_20200630LXF

Command of selecting board configure :

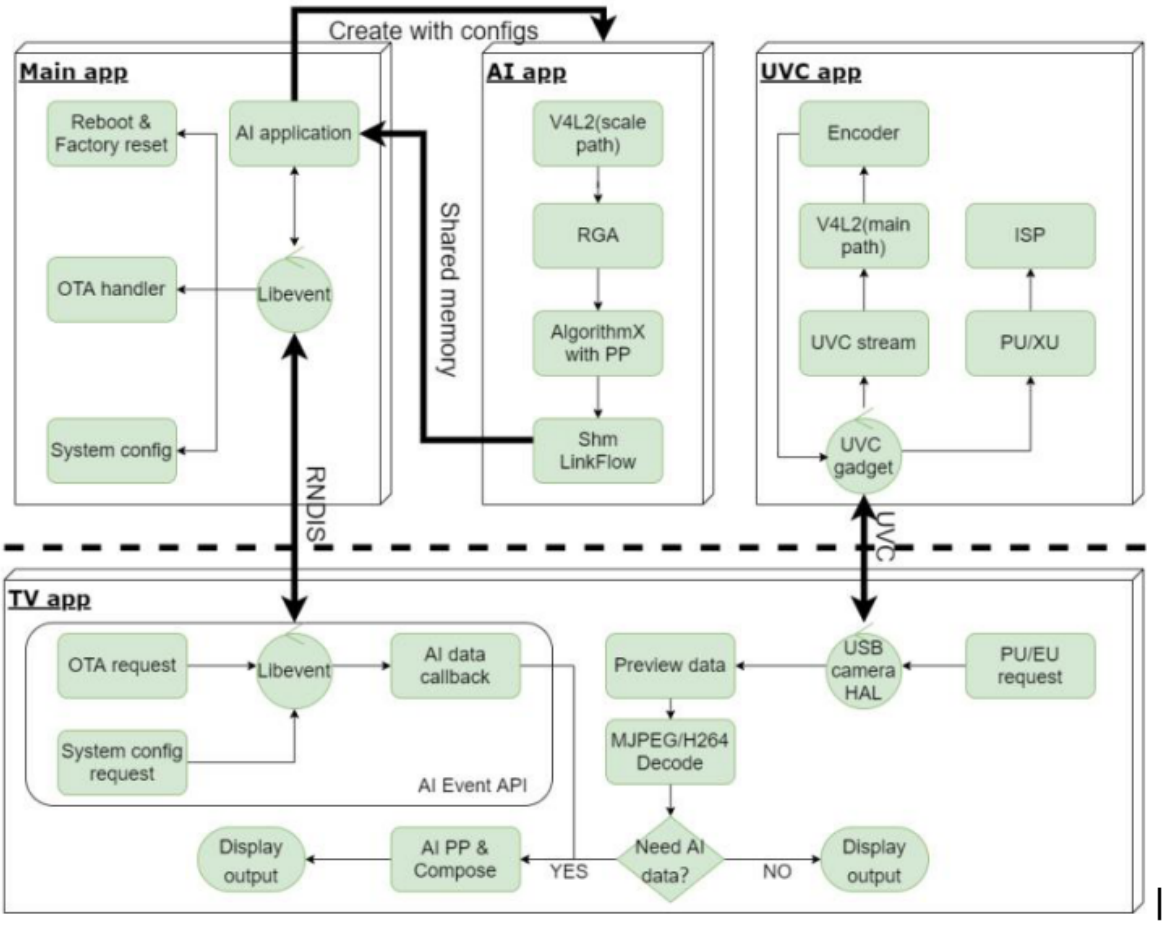
```
### To select Smart USB Camera board config
./build.sh device/rockchip/rv1126_rv1109/BoardConfig-uvcc.mk
```

6.1.2 Building

The building command of the intelligent USB camera product is the same as that of the SDK. Please refer to the SDK Building Introduction in Section 3.

6.2 Product Software Framework

The overall structure is as follows：



The corresponding relationship between rv1109 / rv1126 end application and source code program is as follows:

> **1.main app : source code patch:/app/smart_display_service： Responsible for RNDIS server function implementation, command processing, NN data forwarding and other operations；**
>
> **2.AI app : source code patch: /app/mediaserver： Responsible for sending all camera data to NPU for corresponding NN algorithm processing, and passing it to main app through shared memory mechanism ;**
>
> **3.uvc app:source code patch: /external/uvc_app: Responsible for the implementation and control of the complete functions of UVC camera.**

6.2.1 uvc_app

Please refer to：

```
<SDK>/external/uvc_app/doc/zh-cn/uvc_app.md
```

6.2.2 mediaserver

Please refer to：

```
<SDK>/docs/Linux/AppcationNote/Rockchip_Instructions_Linux_MediaServer_CN.pdf
```

6.2.3 Other

For other Linux application framework or module materials, please refer to the corresponding documents in the following directory：

```
<SDK>/docs/Linux/
```

6.3 Function Introduction

6.3.1 How To Preview For USB Camera

Use USB cable to connect the USB OTG port of EVB with the host computer, such as the USB host port of TV or PC, and power on. By default, the UVC camera application and rndis service will be started automatically. Use serial port to connect EVB board and run ifconfig usb0 to obtain the pre configured IP address of rndis virtual network port.

```
RK $ ifconfig usb0
usb0      Link encap:Ethernet  HWaddr 8E:F3:7D:36:13:34
          inet addr:172.16.110.6  Bcast:172.16.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4884 errors:0 dropped:16 overruns:0 frame:0
          TX packets:4843 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:257305 (251.2 KiB)  TX bytes:787936 (769.4 KiB)
```

Use serial port to connect the PC end of EVB board as follows：

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

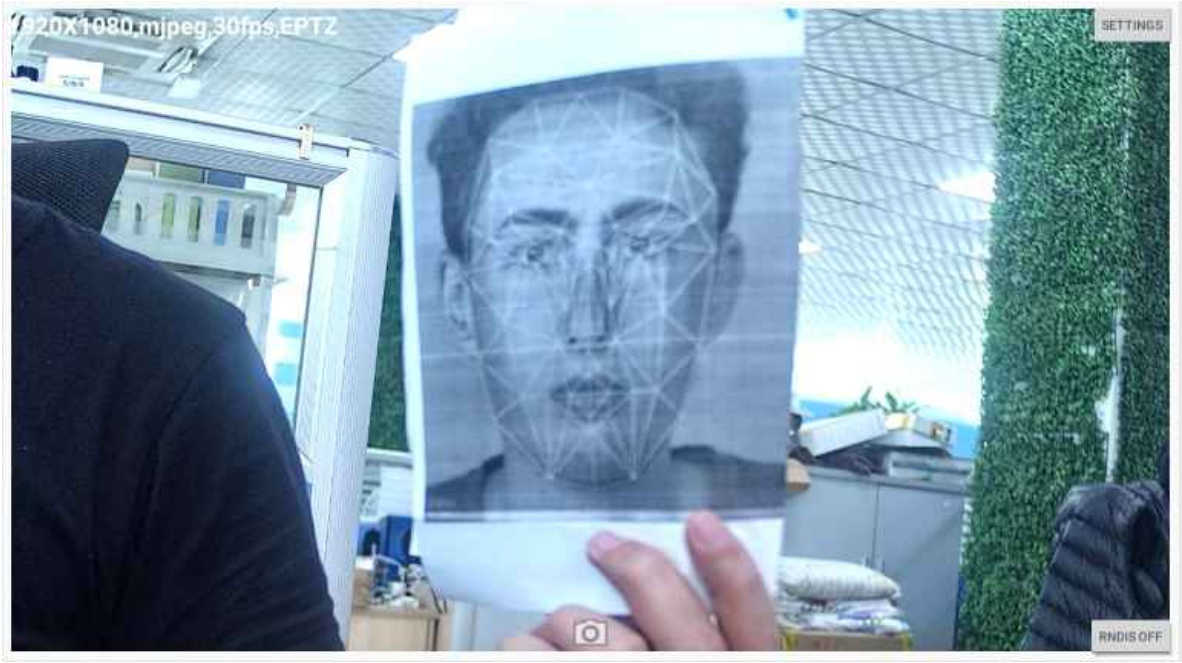
Android smart TV uses RKAICameraTest application or other standard camera applications. PC side recommends the use of a third-party UVC camera application such as Amcap or Potplayer. When it is opened, you can see the preview, and the switching format or resolution can refer to the application switching of upper computer.

6.3.2 How To Test AI Model Post-processing

Open the RKAICameraTest application on the TV terminal, click the rndis button to connect with rndis after seeing the preview, click the settings button to select the "model algorithm switching" option after success, select the model algorithm to be used, which is the face detection algorithm by default, and then click the "Ai post-processing switch", when the face appears in front of the camera, you can see the AI processing effect:

6.3.3 How To Test EPTZ

Open the RKAICameraTest application at the TV end, click the rndis button to connect to rndis after seeing the preview, click the settings button to select the "eptz mode switch" option after the countdown is completed, and then open the application, at this time, the top left corner of the interface will display whether it is the eptz model or the general intelligent preview mode:


920X1080,mjpeg,30fps,EPTZ
SETTINGS
RNDIS OFF