

RV1126/RV1109 Linux SDK 快速入门

文档标识: RK-JC-YF-360

发布版本: V1.4.0

日期: 2020-07-16

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

版权所有 © 2020 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

本文主要描述了RV1126/RV1109 Linux SDK的基本使用方法，旨在帮助开发者快速了解并使用RV1126/RV1109 SDK开发包。SDK下载后，可以查看docs/RV1126_RV1109/RV1126_RV1109_Release_Note.txt，确认当前SDK版本。

产品版本

芯片名称	内核版本
RV1126/RV1109	Linux 4.19

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V0.0.1	CWW	2020-04-28	初始版本
V0.0.2	CWW	2020-05-09	更新5.1.2节RK IPCamera Tool界面
V0.0.3	CWW	2020-05-20	编译环境添加libssl-dev和expect
V1.0.0	CWW	2020-05-25	1. 更新第3节以及第4.4和4.5节 2. 增加快速开机版本编译 3. 增加5.4节
V1.1.0	CWW	2020-06-08	1. 更新公司名称 2. 更新文档排版 3. 更新第2节
V1.2.0	HJC	2020-06-22	增加智能USB Camera产品章节
V1.2.1	CWW	2020-06-29	1. 更新4.4章节 2. 增加编译环境安装fakeroot工具
V1.3.0	CWW	2020-07-09	1. 增加模块目录以及文档说明 2. 增加编译不同板级配置
V1.3.1	CWW	2020-07-15	1. 修正eMMC拼写
V1.4.0	CWW	2020-07-16	1. 增加ISP工具RKISP2.x_Tuner说明 2. 增加开发工具说明 3. 增加板级配置选择说明

目录

RV1126/RV1109 Linux SDK 快速入门

1. 开发环境搭建
2. SDK 配置框架说明
 - 2.1 SDK 目录说明
 - 2.2 RV1109/RV1126 模块代码目录说明
 - 2.3 RV1109/RV1126 开发相关文档
 - 2.4 RV1109/RV1126 开发相关工具
 - 2.4.1 Windows工具
 - 2.4.2 Linux工具
 - 2.5 SDK 配置框架图
3. SDK编译说明
 - 3.1 选择不同板级配置
 - 3.2 查看编译命令
 - 3.3 U-Boot编译
 - 3.4 Kernel编译
 - 3.5 Recovery编译
 - 3.6 Rootfs编译
 - 3.7 固件打包
 - 3.8 全自动编译
4. 刷机说明
 - 4.1 EVB板正面示意图
 - 4.2 EVB板背面示意图
 - 4.3 硬件接口功能表
 - 4.4 Windows 刷机说明
 - 4.5 Linux 刷机说明
5. EVB板功能说明
 - 5.1 如何访问3路RTSP和1路RTMP网络码流
 - 5.1.1 使用串口或ADB连上EVB板子获取设备IP地址
 - 5.1.2 使用RK IPCamera Tool获取设备IP地址
 - 5.1.3 访问网络码流
 - 5.2 如何通过网页访问设备信息
 - 5.3 如何测试人脸识别功能
 - 5.4 如何通过网络调试EVB板
 - 5.4.1 通过SSH登陆EVB板调试
 - 5.4.2 通过SCP调试
6. 智能USB Camera产品配置
 - 6.1 产品编译说明
 - 6.1.1 选择对应板级配置
 - 6.1.2 编译命令
 - 6.2 产品软件框架
 - 6.2.1 uvc_app
 - 6.2.2 mediaserver
 - 6.2.3 其它
 - 6.3 功能说明
 - 6.3.1 如何显示USB Camera预览
 - 6.3.2 如何测试AI模型后处理
 - 6.3.3 如何测试EPTZ功能

1. 开发环境搭建

Ubuntu 16.04系统： 编译环境搭建所依赖的软件包以及安装命令如下：

```
sudo apt-get install repo git-core gitk git-gui gcc-arm-linux-gnueabi 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 dlatex graphviz  
python-matplotlib libc6:i386 libssl-dev expect fakeroot
```

Ubuntu 17.04系统： 除了上述软件包外还需如下依赖包：

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

2. SDK 配置框架说明

2.1 SDK 目录说明

进入工程目录下有buildroot、app、kernel、u-boot、device、docs、external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。

- buildroot: 定制根文件系统。
- app: 存放上层应用程序。
- external: 相关库，包括音频、视频等。
- kernel: kernel代码。
- device/rockchip: 存放每个平台的一些编译和打包固件的脚本和预备文件。
- docs: 存放开发指导文件、平台支持列表、工具使用文档、Linux 开发指南等。
- prebuilts: 存放交叉编译工具链。
- rkbin: 存放固件和工具。
- rockdev: 存放编译输出固件。
- tools: 存放一些常用工具。
- u-boot: U-Boot代码。

2.2 RV1109/RV1126 模块代码目录说明

部分模块代码目录路径	模块功能描述
external/recovery	recovery
external/rkwifibt	Wi-Fi和BT
external/libdrm	DRM接口
external/rk_pcba_test	PCBA测试代码
external/isp2-ipc	图像信号处理服务端
external/mpp	编解码代码
external/rkmedia	Rockchip 多媒体封装接口
external/rkupdate	Rockchip升级代码
external/camera_engine_rkaiq	图像处理算法模块
external/rknpu	NPU驱动
external/rockface	人脸识别代码
external/CallFunIpc	应用进程间通信代码
external/common_algorithm	音视频通用算法库
external/rknn-toolkit	模型转换、推理和性能评估的开发套件
app/libIPCProtocol	基于dbus，提供进程间通信的函数接口
app/mediaserver	提供多媒体服务的主应用
app/ipc-daemon	系统守护服务
app/dbserver	数据库服务
app/netserver	网络服务
app/storage_manager	存储管理服务
app/ipcweb-backend	web后端
app/librddb	数据库接口
app/ipcweb-ng	web前端，采用Angular 8框架

2.3 RV1109/RV1126 开发相关文档

```

├─ docs
│   └─ Linux
│       └─ ApplicationNote (Rockchip应用开发框架介绍、网页端开发指南)
│           └─ Rockchip_Developer_Guide_Linux_Application_Framework_CN.pdf
│           └─ Rockchip_Instructions_Linux_MediaServer_CN.pdf
│           └─ Rockchip_Instructions_Linux_Web_Configuration_CN.pdf
│       └─ Multimedia (ISP开发指南、编解码以及接口封装开发指南)
│           └─ camera

```

```

| | | | └─ Rockchip_Developer_Guide_ISP20_RkAiq_CN.pdf
| | | | └─ Rockchip_Instruction_Linux_Application_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 (快速开发指南、硬件开发指南、发布说明、编解码说明)
|   └─ 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 (模型转换、推理和性能评估的开发套件文档)
        └─ 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 开发文档)
                └─ 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 调试工具)
            └─ RKISP2.x_Tuner_User_Manual_v1.0.pdf

```

2.4 RV1109/RV1126 开发相关工具

2.4.1 Windows工具

工具说明文档: `tools/windows/ToolsRelease.txt`

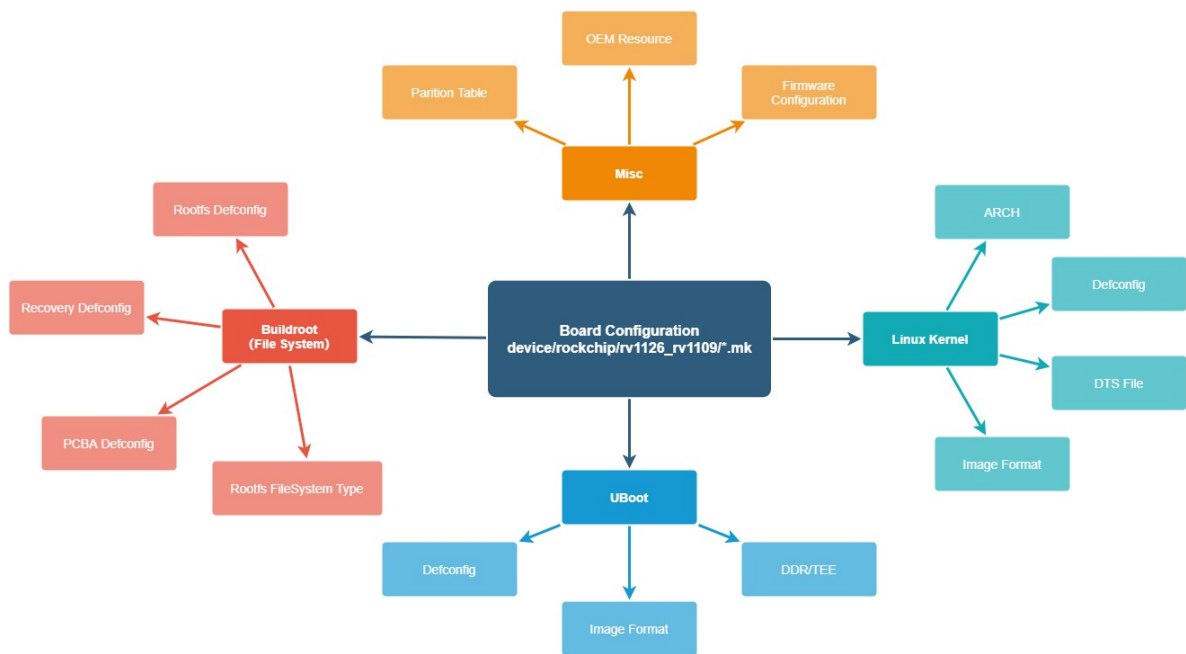
工具名称	工具用途
RKDevTool	分立升级固件及整个update升级固件工具
FactoryTool	量产升级工具
SecureBootTool	固件签名工具
efuseTool	efuse烧写工具
RKDevInfoWriteTool	写号工具
SDDiskTool	SD卡镜像制作
SpiImageTools	烧录器升级工具
DriverAssitant	驱动安装工具
RKImageMaker	打包工具(打包成update.img)
SpeakerPCBATool	音箱PCBA测试工具
RKDevTool_Release	固件烧录工具
ParameterTool	分区表修改工具
RKISP2.x_Tuner	ISP工具
RK_IPCamera_Tool	IPC设备搜索工具

2.4.2 Linux工具

工具说明文档：tools/linux/ToolsRelease.txt

工具名称	工具用途
Linux_Pack_Firmware	固件打包工具(打包成update.img)
Linux_Upgrade_Tool	烧录固件工具
Linux_SecureBoot	固件签名工具
Firmware_Merger	SPI NOR固件打包工具(生成的固件可以用于烧录器)

2.5 SDK 配置框架图



3. SDK编译说明

3.1 选择不同板级配置

SDK下载地址：

```
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
```

芯片	板级配置 (目录 device/rockchip/rv1126_rv1109)	存储介 质	EVB板	支持 快 速 开 机
RV1109	BoardConfig-38x38-spi-nand-rv1109.mk	SPI NAND	RV1126_RV1109_38X38_SPI_DDR3P216DD6_V10_20200511LXF	NO
RV1109	BoardConfig-rv1109.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	NO
RV1109	BoardConfig-tb-rv1109.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	YES
RV1126	BoardConfig-spi-nand.mk	SPI NAND	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	NO
RV1126	BoardConfig.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	NO
RV1126	BoardConfig-tb.mk	eMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY	YES

切换板级配置命令：

```
### 方法1
### ./build.sh 后面加上板级配置文件，例如：
### 选择通用版本的板级配置
./build.sh device/rockchip/rv1126_rv1109/BoardConfig.mk
### 选择快速开机的板级配置
./build.sh device/rockchip/rv1126_rv1109/BoardConfig-tb.mk
```



```

### 方法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-rv1109.mk
2. BoardConfig-rv1109-uvcc.mk
3. BoardConfig-rv1109.mk
4. BoardConfig-spi-nand.mk
5. BoardConfig-tb-rv1109.mk
6. BoardConfig-tb.mk
7. BoardConfig-uvcc.mk
8. BoardConfig.mk
Which would you like? [0]: 8
switching to board:
/home/user/rv1109/device/rockchip/rv1126_rv1109/BoardConfig.mk

```

3.2 查看编译命令

在根目录执行命令: `./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'.

```

查看部分模块详细编译命令，例如：./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编译

```
### U-Boot编译命令
./build.sh uboot

### 查看U-Boot详细编译命令
./build.sh -h uboot
```

3.4 Kernel编译

```
### Kernel编译命令
./build.sh kernel

### 查看Kernel详细编译命令
./build.sh -h kernel
```

3.5 Recovery编译

```
### Recovery编译命令
./build.sh recovery

### 查看Recovery详细编译命令
./build.sh -h recovery
```

3.6 Rootfs编译

```
### Rootfs编译命令
./build.sh rootfs

### 查看Rootfs详细编译命令
./build.sh -h rootfs
```

3.7 固件打包

固件打包命令：./mkfirmware.sh

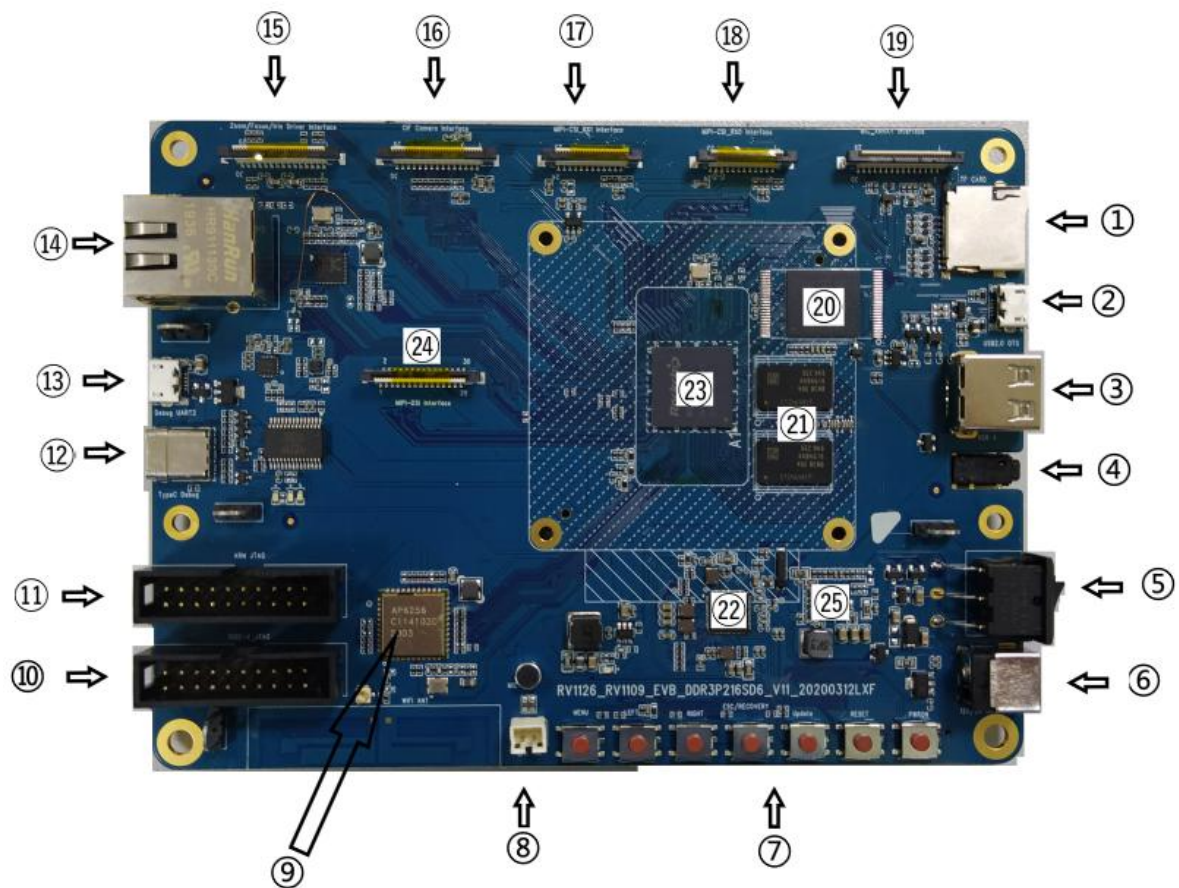
3.8 全自动编译

进入工程根目录执行以下命令自动完成所有的编译:

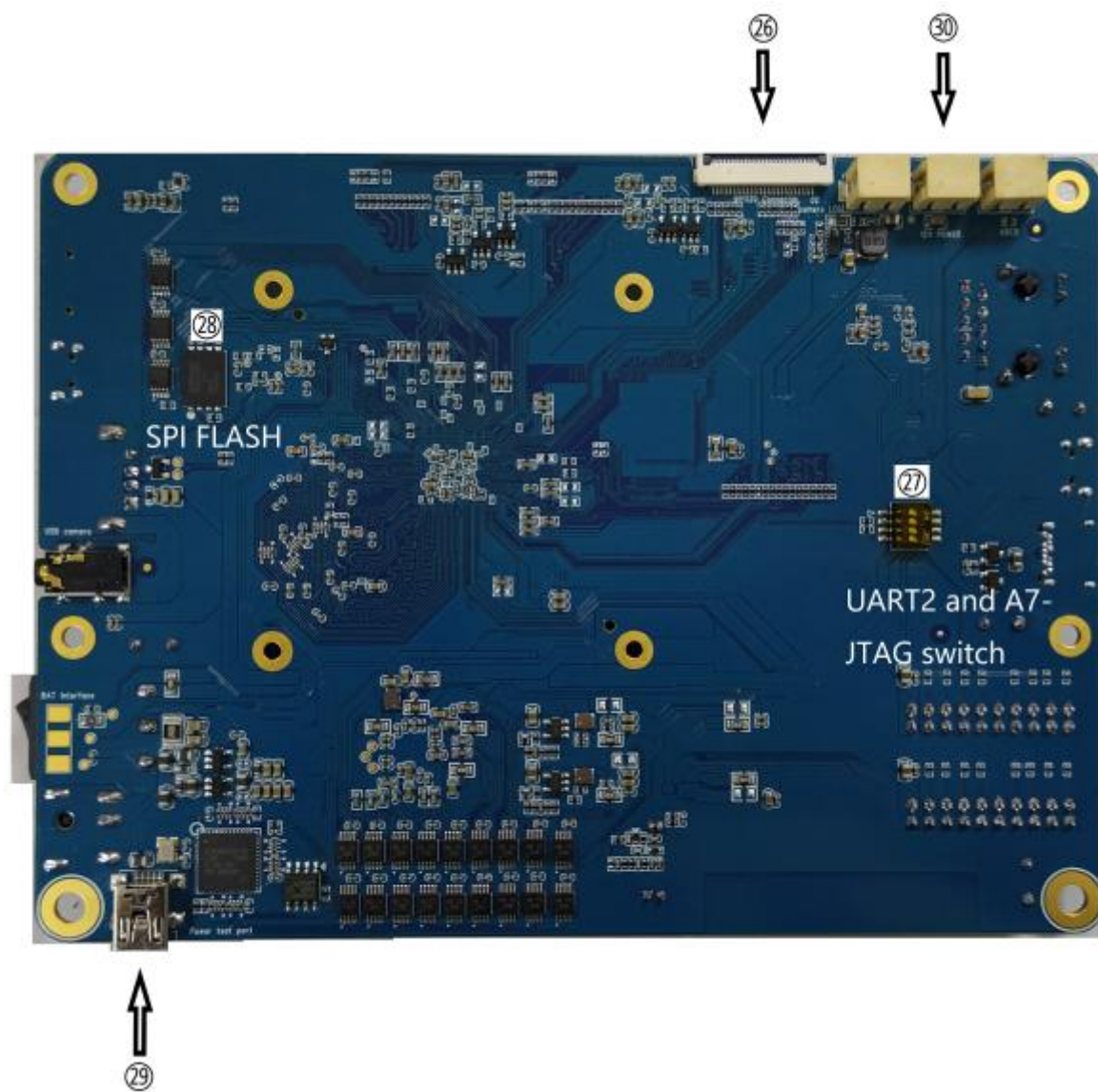
```
./build.sh all
```

4. 刷机说明

4.1 EVB板正面示意图



4.2 EVB板背面示意图



4.3 硬件接口功能表

序号 Item	功能部分 Function Part	要求 Requirement
1	TF Card	正常识别TF Card
2	USB Micro-B Port	可以认到ADB设备，可以下载固件
3	USB Type-A Port	可以识别device设备，且功能正常
4	USB camera input	正常识别USB camera
5	the boat switch	直流适配器输入的12v电源，可以通过船型开关来控制打开或关闭
6	12V power supply input	
7	KEY BAORD	所有按键功能正常
8	CLASS D output	喇叭功能正常
9	WI-FI/BT	AP6256模组功能正常
10	RISC-V JTAG	芯片验证调试
11	V7-JTAG	
12	TPYEC	
13	USB Micro-B Port	串口可以正常输入和输出
14	以太网Ethernet	网络连接正常
15	Zoom/ Iris Driver Interface	开发板预留Zoom/Focus/Iris连接座，方便客户进行CAMERA设备调试开发。
16	CIF camera	摄像头功能正常，CIF摄像头输入
17	MIPI Camera 1	摄像头功能正常，默认MIPI摄像头输入
18	MIPI Camera 2	摄像头功能正常，默认MIPI摄像头输入
19	MIC-ARRAY	MIC阵列输入
20	eMMC Flash	可以正常识别容量16GByte
21	DDR DDR3	可识别到总容量8Gbit
22	PMIC RK809-2	各路电源正常输出，电池电量检测准确
23	CPU	RV1126_RV1109
24	MIPI屏 MIPI panel	屏幕图像显示正常
25	BQ24171	双节电池充放电正常
Bottom Layer		
26	BT1120 Camera	摄像头功能正常，BT1120 摄像头输入
27	功能切换SWITCH	UART2和A7-JTAG功能切换
28	SPI flash	验证SPI flash功能
29	USB Micro-B Port	用于功耗测试
30	Camera_LED 驱动输出	Warm up lamp drive

4.4 Windows 刷机说明

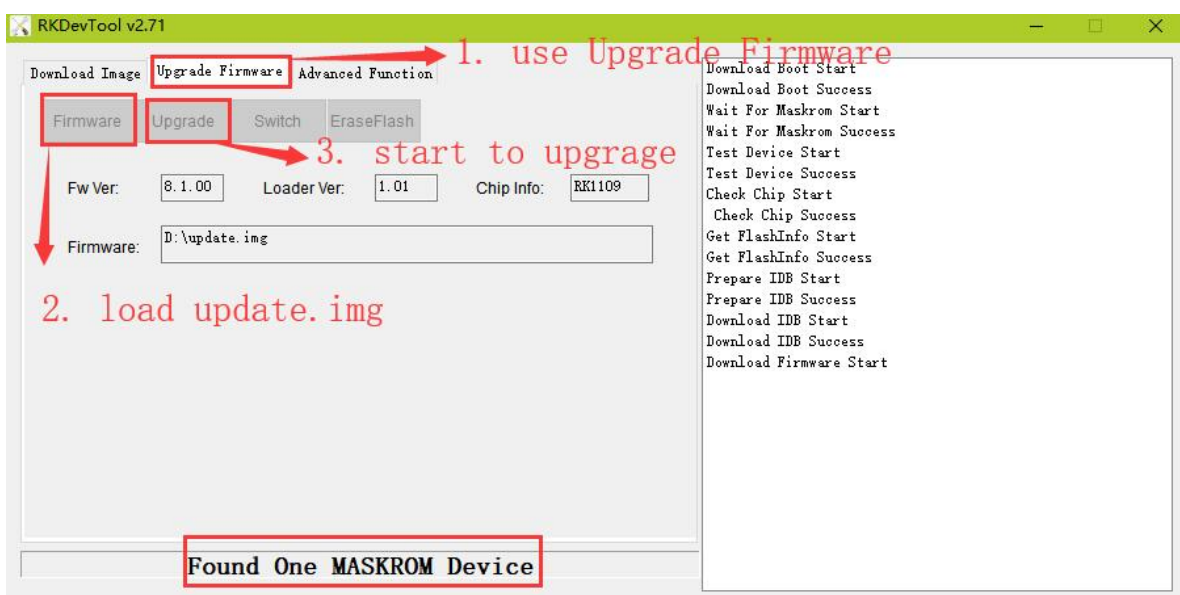
SDK 提供 Windows 烧写工具(工具版本需要 V2.71 或以上)，工具位于工程根目录：

```
tools/
└─ windows/RKDevTool
```

如下图，编译生成相应的固件后，设备烧写需要进入 MASKROM 或 BootROM 烧写模式，连接好 USB 下载线后，按住按键“Update”不放并按下复位键“RESET”后松手，就能进入 MASKROM 模式，加载编译生成固件的相应路径后，点击“执行”进行烧写，也可以按“recovery”按键不放并按下复位键“RESET”后松手进入 loader 模式进行烧写，下面是 MASKROM 模式的分区偏移及烧写文件。(注意：Windows PC 需要在管理员权限运行工具才可执行)



烧录update.img方法:



注:

1. 除了MiniLoaderAll.bin和parameter.txt，实际需要烧录的分区根据rockdev/parameter.txt配置为准。
2. 烧写前，需安装最新 USB 驱动，驱动详见：

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

4.5 Linux 刷机说明

Linux 下的烧写工具位于 tools/linux 目录下(Linux_Upgrade_Tool 工具版本需要 V1.49 或以上)，请确认你的板子连接到 MASKROM/loader rockusb。比如编译生成的固件在 rockdev 目录下，升级命令如下：

除了MiniLoaderAll.bin和parameter.txt，实际需要烧录的分区根据rockdev/parameter.txt配置为准。

```
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 rockdev/rootfs.img
sudo ./upgrade_tool di -userdata rockdev/userdata.img
sudo ./upgrade_tool rd
```

或升级整个 firmware 的 update.img 固件：

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

或在根目录，机器在 MASKROM 状态运行如下升级：

```
./rkflash.sh
```

5. EVB板功能说明

EVB板支持如下功能：

- 支持3路RTSP和1路RTMP网络码流
- 支持本地屏幕1280x720显示
- 支持保存主码流到设备
- 支持网页端访问设备
- 支持人脸识别

5.1 如何访问3路RTSP和1路RTMP网络码流

使用网线接到EVB板的网口，上电开机。默认会自动获取IP地址。

5.1.1 使用串口或ADB连上EVB板子获取设备IP地址

```
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
```

使用串口连接EVB板子的PC端配置如下：

波特率：1500000
数据位：8
停止位：1
奇偶校验：none
流控：none

5.1.2 使用RK IPCamera Tool获取设备IP地址

安装SDK目录tools/windows/RK_IPCamera_Tool-V1.1.zip工具。打开工具，通过EVB板网口连接到电脑所在局域网，查看RK IPCamera Tool工具设备总数列表获取设备IP地址。



说明：

1. 点击“开启搜索”，进行设备搜索
2. 选择一个设备
3. 取消自动获取IP，改为静态IP
4. 设置静态IP
5. 设置IP
6. 打开预览

5.1.3 访问网络码流

使用支持RTSP或RTMP的播放器访问，例如（VLC播放器）。

RTSP访问地址：

- rtsp://设备IP地址/live/mainstream
- rtsp://设备IP地址/live/substream
- rtsp://设备IP地址/live/thirdstream

RTMP访问地址:

- rtmp://设备IP地址:1935/live/substream

5.2 如何通过网页访问设备信息

打开Web浏览器（推荐Chrome浏览器）访问地址:

```
http://设备IP地址
```

网页端详细的操作说明请参考SDK目录docs下的文档
Rockchip_Instructions_Linux_Web_Configuration_CN.pdf。

5.3 如何测试人脸识别功能

使用播放器访问RTSP主码流: rtsp://设备IP地址/live/mainstream

SDK的人脸识别功能默认授权的测试时间是30~60分钟，授权失效后主码流预览会有“人脸算法软件未授权”提示，需要重启才能再测试。

5.4 如何通过网络调试EVB板

5.4.1 通过SSH登陆EVB板调试

接上以太网，通过第5.1.2节 [使用RK IPCamera Tool获取设备IP地址](#) 获取EVB板IP地址。保证PC电脑可以ping通EVB板。

```
### 清除上次登陆信息（EVB板的IP地址192.168.1.159）
ssh-keygen -f "$HOME/.ssh/known_hosts" -R 192.168.1.159
### 使用SSH命令登陆
ssh root@192.168.1.159
### 输入默认密码: rockchip
```

5.4.2 通过SCP调试

```
### 从PC端上传文件test-file到EVB板的目录/userdata
scp test-file root@192.168.1.159:/userdata/
root@192.168.1.159's password:
### 输入默认密码: rockchip

### 下载EVB板上的文件/userdata/test-file下载到PC端
scp root@192.168.1.159:/userdata/test-file test-file
root@192.168.1.159's password:
### 输入默认密码: rockchip
```

6. 智能USB Camera产品配置

智能USB Camera产品支持如下功能：

- 支持标准UVC Camera功能，最高支持4k预览（RV1126）
- 支持多种NN算法，包括人脸检测，人体姿态或骨骼检测，人脸关键点检测跟踪等，支持第三方算法扩展
- 支持USB复合设备稳定传输（RNDIS/UAC/ADB等）
- 支持NN前处理和数据后处理通路
- 支持智能电视或PC等多种终端设备预览
- 支持EPTZ功能

6.1 产品编译说明

智能USB Camera产品编译配置基于公版SDK，采用单独的rv1126_rv1109_linux_ai_camera_release.xml代码清单管理更新。

6.1.1 选择对应板级配置

SDK下载地址：

```
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
```

芯片	板级配置 (目录 device/rockchip/rv1126_rv1109)	存储介 质	EVB板
RV1109	BoardConfig-rv1109-uvcc.mk	EMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY
RV1126	BoardConfig-uvcc.mk	EMMC	RV1126_RV1109_EVB_DDR3P216SD6_V12_20200515KYY

切换板级配置命令：

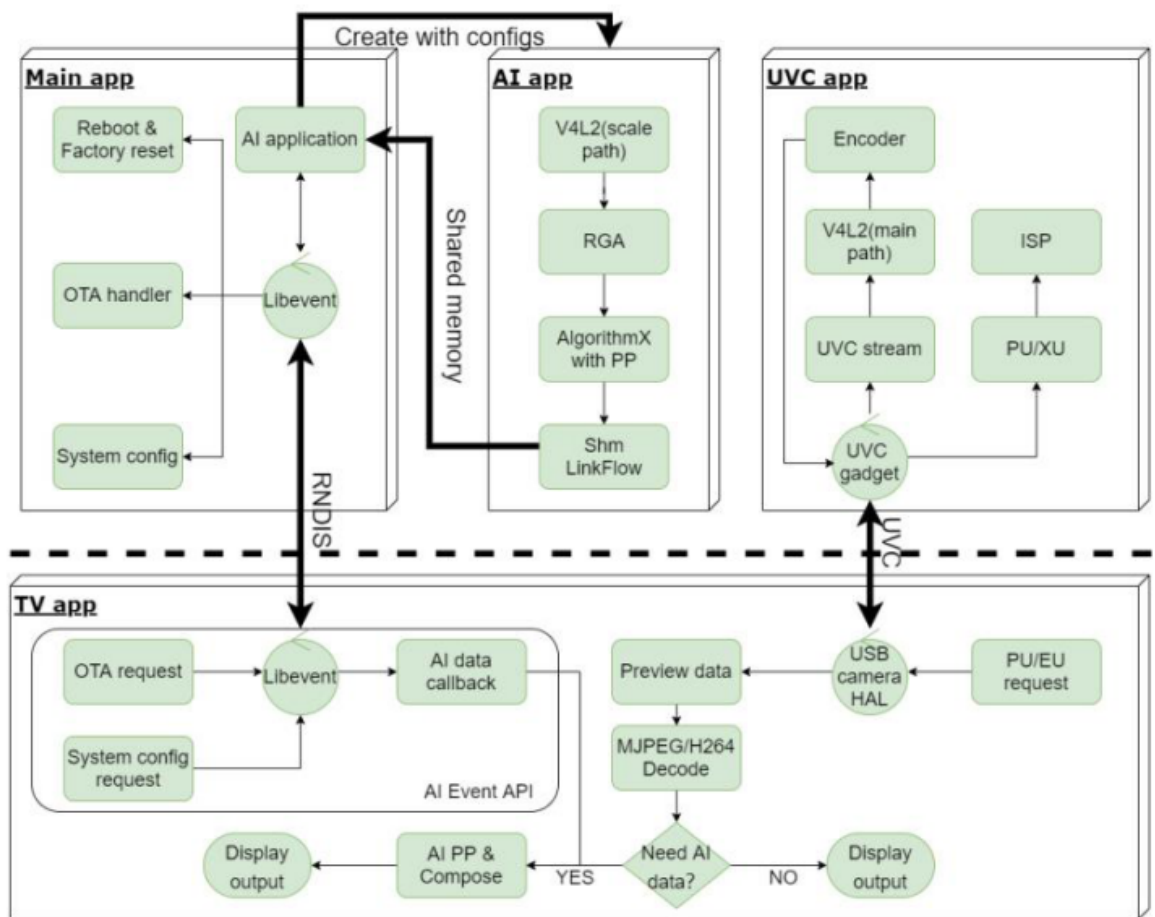
```
### 选择智能USB Camera版本的板级配置
./build.sh device/rockchip/rv1126_rv1109/BoardConfig-uvcc.mk
```

6.1.2 编译命令

智能USB Camera产品的编译命令同SDK，参考第三节SDK编译说明即可。

6.2 产品软件框架

总体结构如下：



其中,RV1109/RV1126端应用与源码程序对应关系如下:

- 1.main app 对应/app/smart_display_service: 负责RNDIS 服务端功能实现, 命令处理, NN数据转发等操作;
- 2.AI app 对应/app/mediaserver: 负责将一路camera数据送到NPU做对应NN算法处理, 通过共享内存机制传递给main app;
- 3.uvc app 对应/external/uvc_app: 负责UVC camera完整功能的实现和控制。

6.2.1 uvc_app

请参考:

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

6.2.2 mediaserver

请参考:

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

6.2.3 其它

其它linux应用框架或模块资料, 请参考下列目录对应文档:

6.3 功能说明

6.3.1 如何显示USB Camera预览

使用USB线连接EVB的USB OTG口与上位机，如TV端或PC端USB host 口，上电开机。默认会自动启动UVC camera应用及RNDIS服务。使用串口连上EVB板子运行ifconfig usb0可获取预配置的RNDIS 虚拟网口IP地址。

```
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)
```

使用串口连接EVB板子的PC端配置如下：

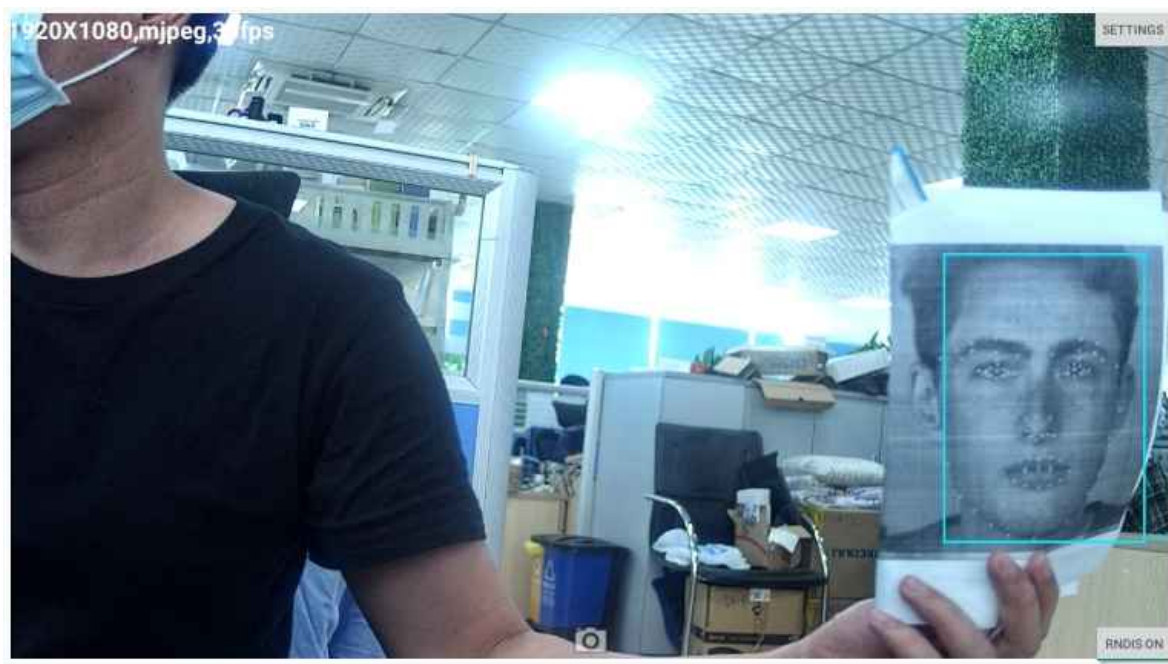
```
波特率：1500000
数据位：8
停止位：1
奇偶校验：none
流控：none
```

Android智能电视使用RKAICameraTest应用或其他标准camera应用，PC端推荐使用如Amcap或Potplayer等第三方UVC camera应用，打开即可看到预览，切换格式或分辨率参考上位机上camera应用的设置菜单中功能切换即可。



6.3.2 如何测试AI模型后处理

在电视端打开RKAICameraTest应用，看到预览后点击RNDIS按钮连接RNDIS，成功后点击SETTINGS按钮选择“模型算法切换”选项，选择要使用的模型算法，默认为人脸检测算法，然后点击“AI后处理开关”，当人脸在镜头前出现即可看到AI处理效果：



6.3.3 如何测试EPTZ功能

在电视端打开RKAICameraTest应用，看到预览后点击RNDIS按钮连接RNDIS，成功后点击SETTINGS按钮选择“EPTZ模式切换”选项，在倒计时完成后，再打开应用即可，此时在界面左上角会显示是EPTZ模型还是普通智能预览模式：

