

Rockchip RK356X Linux USB Camera SDK

快速入门

文档标识: RK-JC-YF-541

发布版本: V1.1.0

日期: 2021-11-08

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 © 2021 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RK356X Linux USB Camera SDK的基本使用方法，旨在帮助开发者快速了解并使用RK356X Linux USB Camera SDK开发包。

本开发包适用但不限于USB camera产品，提供灵活的数据通路组合接口，满足客户自由组合的客制化需求。

产品版本

芯片名称	内核版本
RK356X	Linux 4.19

读者对象

本文档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

日期	版本	作者	修改说明
2021/08/17	1.0.0	WT	初始版本
2021/11/08	1.1.0	WT	添加USB 3.0、HDMI、休眠唤醒补丁说明

目录

Rockchip RK356X Linux USB Camera SDK 快速入门

1. 开发环境搭建
 - 1.1 命令提示约定
 - 1.2 Linux服务器配置
2. SDK目录结构说明
3. SDK编译说明
 - 3.1 切换CAMERA产品配置
 - 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 Windows 刷机说明
 - 4.2 Linux 刷机说明
5. EVB板功能说明
 - 5.1 如何显示主camera预览
 - 5.2 UVC项目
 - 5.3 HDMI显示
 - 5.4 休眠唤醒
6. 应用软件框架
 - 6.1 uvc_app
 - 6.2 aiserver
 - 6.3 其它
7. FAQ
 - 7.1 如何重编一个模块
 - 7.2 使用gdb调试

Rockchip

Confidential

1. 开发环境搭建

1.1 命令提示约定

以下是本文涉及命令输入时的执行环境的约定：

执行在 Linux 服务器

```
Server $
```

执行在device端控制台

```
RK $
```

1.2 Linux服务器配置

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：

```
Server $ 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 dblatex graphviz python-  
matplotlib libc6:i386
```

Ubuntu 17.04系统：

除了上述软件包外还需如下依赖包：

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

2. SDK目录结构说明

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，包括多媒体相关，uvc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。

3. SDK编译说明

3.1 切换CAMERA产品配置

在根目录执行命令:

- 选择产品

```
Server $ source envsetup.sh rockchip_rk3568_uvc
Top of tree: /home1/wt/rk356x_linux
=====

#TARGET_BOARD=rk3568
#OUTPUT_DIR=output/rockchip_rk3568_uvc
#CONFIG=rockchip_rk3568_uvc_defconfig

=====
...
```

- 选择板级配置

```
Server $ ./build.sh BoardConfig-rk3568-uvc-evb1-ddr4-v10.mk
processing option: BoardConfig-rk3568-uvc-evb1-ddr4-v10.mk
switching to board:
/home1/wt/rk356x_linux/device/rockchip/rk356x/BoardConfig-rk3568-uvc-evb1-
ddr4-v10.mk
```

3.2 查看编译命令

在根目录执行命令: `./build.sh -h|help`

```
Server $ ./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'.
```

查看部分模块详细编译命令，例如：`./build.sh -h kernel`

```
Server $ ./build.sh -h kernel
###Current SDK Default [ kernel ] Build Command###
Server $ cd kernel
Server $ make ARCH=arm64 rockchip_linux_defconfig
Server $ make ARCH=arm64 rk3568-evb1-ddr4-v10-linux.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
Server $ ./build.sh -h uboot
###Current SDK Default [ uboot ] Build Command###
Server $ cd u-boot
Server $ ./make.sh rk3568
```

3.4 Kernel编译

Kernel编译命令：`./build.sh kernel`

```
### 查看Kernel详细编译命令
Server $ ./build.sh -h kernel
###Current SDK Default [ kernel ] Build Command###
Server $ cd kernel
Server $ make ARCH=arm64 rockchip_linux_defconfig
Server $ make ARCH=arm64 rk3568-evb1-ddr4-v10-linux.img -j12
```

3.5 Recovery编译

Recovery编译命令: `./build.sh recovery`

```
### 查看Recovery详细编译命令
Server $ ./build.sh -h recovery
###Current SDK Default [ recovery ] Build Command###
Server $ source envsetup.sh rockchip_rk356x_recovery
Server $ /home/user/sdk/device/rockchip/common/mk-ramdisk.sh recovery.img
rockchip_rk356x_recovery
```

3.6 Rootfs编译

Rootfs编译命令: `./build.sh rootfs`

```
### 查看Rootfs详细编译命令
Server $ ./build.sh -h rootfs
###Current SDK Default [ rootfs ] Build Command###
Server $ source envsetup.sh rockchip_rk3568_uvc
Server $ make
```

3.7 固件打包

固件打包命令: `./mkfirmware.sh`

固件目录: rockdev

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
```

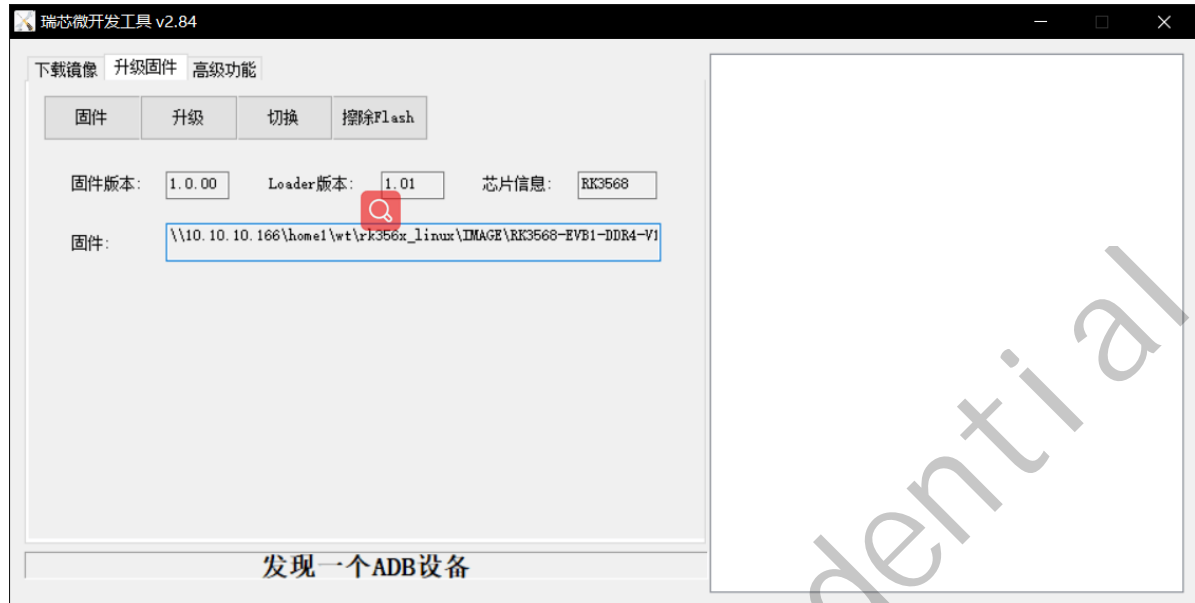
4. 刷机说明

4.1 Windows 刷机说明

SDK 提供 Windows 烧写工具(工具版本需要 V2.84 或以上), 工具位于工程根目录:

```
tools/
├─ windows/AndroidTool
```

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“Maskrom”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“V+”按键不放并按下复位键“RESET”后松手进入 loader 模式进行烧写，下面是烧写 Update.img 的方式(注意：Windows PC 需要在管理员权限运行工具才可执行)



注：烧写前，需安装最新 USB 驱动，驱动详见：

```
<SDK>/tools/windows/DriverAssitant_v5.11.zip
```

4.2 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：

```
Server $ sudo ./upgrade_tool ul rockdev/MiniLoaderAll.bin
Server $ sudo ./upgrade_tool di -p rockdev/parameter.txt
Server $ sudo ./upgrade_tool di -u rockdev/uboot.img
Server $ sudo ./upgrade_tool di -misc rockdev/misc.img
Server $ sudo ./upgrade_tool di -b rockdev/boot.img
Server $ sudo ./upgrade_tool di -recovery rockdev/recovery.img
Server $ sudo ./upgrade_tool di -oem rockdev/oem.img
Server $ sudo ./upgrade_tool di -rootfs rockdev/rootfs.img
Server $ sudo ./upgrade_tool di -userdata rockdev/userdata.img
Server $ sudo ./upgrade_tool rd
```

或升级整个 firmware 的 update.img 固件：

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

或在根目录，机器在 MASKROM 状态运行如下升级：

```
Server $ ./rkflash.sh
```


5. EVB板功能说明

EVB板USB Camera固件支持如下功能：

- 支持标准UVC Camera功能，最高支持4k预览（RK356X）
- 支持USB复合设备稳定传输
- 支持智能电视或PC等多种终端设备预览

5.1 如何显示主camera预览

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
```

PC端使用amcap或PotPlayer等USB camera应用，Android Host使用RKAICameraTest应用或其他标准camera应用，打开即可看到预览，切换格式或分辨率参考上位机所用应用切换即可。



5.2 UVC项目

RK356X支持UVC工程，需先在kernel打上以下补丁。

```

diff --git a/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi
b/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi
index 18d6115341cd..268f5d2b1c05 100644
--- a/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi
@@ -1778,6 +1778,8 @@

&usbdrd_dwc3 {
    dr_mode = "otg";
+   snps,tx-fifo-resize;
+   snps,dis-ulu2-quirk;
    extcon = <&usb2phy0>;
    status = "okay";
};

```

5.3 HDMI显示

RK356X可支持外接HDMI显示，如有需要，需在kernel打开vop模块。

```

diff --git a/arch/arm64/boot/dts/rockchip/rk3568-linux.dtsi
b/arch/arm64/boot/dts/rockchip/rk3568-linux.dtsi
index 5623aa8dd15c..e1b283d240ad 100644
--- a/arch/arm64/boot/dts/rockchip/rk3568-linux.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3568-linux.dtsi
@@ -97,7 +97,3 @@

&rockchip_suspend {
    status = "okay";
};
-
-&vop {
-   disable-win-move;
-};

```

5.4 休眠唤醒

RK356X可支持红外休眠唤醒，如有需要，需在kernel dts中配置suspend。

```

diff --git a/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10-linux.dts
b/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10-linux.dts
index b6b618bb561a..30b531dac360 100644
--- a/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10-linux.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3568-evb1-ddr4-v10-linux.dts
@@ -15,3 +15,25 @@

&vp1 {
    cursor-win-id = <ROCKCHIP_VOP2_CLUSTER1>;
};
+
+&rockchip_suspend {
+   status = "okay";
+   rockchip,sleep-mode-config = <
+       (0
+       | RKPM_SLP_ARMOFF

```

```
+         | RKPM_SLP_CENTER_OFF
+         | RKPM_SLP_HW_PLLS_OFF
+         | RKPM_SLP_PMUALIVE_32K
+         | RKPM_SLP_PMIC_LP
+         | RKPM_SLP_32K_PVTM
+     )
+ >;
+     rockchip,wakeup-config = <
+         (0
+         | RKPM_GPIO_WKUP_EN
+         | RKPM_PWM0_WKUP_EN
+         | RKPM_CPU0_WKUP_EN
+         )
+     >;
+};
```

6. 应用软件框架

其中,RK356X端应用与源码程序对应关系如下:

- 1.aiserver 对应/app/aiserver: 负责将一路camera数据送到uvc/gpu/vo, 实现usb camera/畸变矫正/HDMI显示器显示等;
- 2.uvc_app 对应/external/uvc_app: 负责uvc camera完整功能的实现和控制。

6.1 uvc_app

请参考:

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

6.2 aiserver

请参考:

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

6.3 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:

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

7. FAQ

7.1 如何重编一个模块

以mpp模块为例，重编可以使用：

```
make mpp-rebuild
```

7.2 使用gdb调试

打开buildroot的gdb配置，然后重新编译打包文件系统

```
# buildroot目录下
diff --git a/configs/rockchip_rk3568_uvc_defconfig
b/configs/rockchip_rk3568_uvc_defconfig
index 237a380ccf..cd219c8db9 100644
--- a/configs/rockchip_rk3568_uvc_defconfig
+++ b/configs/rockchip_rk3568_uvc_defconfig
@@ -14,6 +14,7 @@
 #include "rk356x_arm64.config"
 #include "test.config"
 #include "wifi.config"
+#include "gdb.config"
 BR2_PACKAGE_RKWIFIBT_AP6398S=y
 BR2_PACKAGE_RKWIFIBT_BTUART="ttyS8"
 BR2_ROOTFS_OVERLAY="board/rockchip/rk356x/fs-overlay-uvic/"
```

注：运行gdb时，需要在gdb启动后，手动配置信号

```
RK $ handle SIGILL pass nostop noprint
```