RK356X Linux USB Camera SDK Quick Start

ID: RK-JC-YF-541

Release Version: V1.1.0

Release Date: 2021-11-08

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]mailto:fae@rock-chips.com)

Preface

Overview

This article describes the basic usage of the RK356X Linux USB Camera SDK and aims to help developers quickly understand and use the RK356X Linux USB Camera SDK development kit.
This development kit is applicable to, but not limited to, USB camera products, and provides a flexible data path combination interface to meet customers' customization needs.

Product Version

Chip Name	Kernel Version
RK356X	Linux 4.19

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Revision Date	Version	Author	Revision Description
2021-08-17	V1.0.0	WT	Initial version
2021-11-08	V1.1.0	WT	Add USB 3.0, HDMI Add Hibernation wake up patch instructions

Content

1. Development Environment Setup

1.1 Command prompt conventions

The following are the conventions for the execution environment when this article deals with command input.
Execute on Linux server

```
Server $
```

Execute on the device control terminal

```
RK $
```

1.2 Linux server configuration

Ubuntu 16.04 system:
The packages that the compiler environment is built on and the installation commands are as follows.

```
Server $ 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
```

Ubuntu 17.04 system:
In addition to the above packages, the following dependency packages are required.

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

2. SDK Directories Introduction

The SDK directory contains buildroot, app, kernel, u-boot, device, docs, external and other directories. Each directory or its subdirectories corresponds to a git project, and submissions need to be made in their respective directories.

- buildroot: Customize the root file system.
- app: Save the upper-level application.
- external: Related libraries, including multimedia related, UVC, etc.
- kernel: kernel code
- device/rockchip: Save the scripts and prep files for compiling and packaging firmware on each platform.

- docs: Save development guides, platform support lists, tool usage documents, Linux development guides, etc.
- prebuilts: Save the cross-compilation toolchain.
- rkbin: Save firmware and tools.
- rockdev: Save compiled firmware.
- tools: Save some common tools.
- u-boot: U-Boot code

3. SDK Compilation Instruction

3.1 Switching CAMERA product configurations

Execute the command in the root directory:

- Select Product

```
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

===========================================
...
```

- Select board configuration

```
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 Check compile command

Execute the command in the root directory: ./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'.
```

Check detailed build commands for some of the modules, e.g.: ./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 compiling

U-Boot compile command: `./build.sh uboot`

```
### Check U-Boot compile command
Server $ ./build.sh -h uboot
###Current SDK Default [ uboot ] Build Command###
Server $ cd u-boot
Server $ ./make.sh rk3568
```

3.4 Kernel compiling

Kernel compile command: `./build.sh kernel`

```
### Check Kernel compile command
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 compiling

Recovery compile command: `./build.sh recovery`

```
### Check Recovery compile command
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 compiling

Rootfs compile command: `./build.sh rootfs`

```
### Check Roofs compile command
Server $ ./build.sh -h rootfs
###Current SDK Default [ rootfs ] Build Command###
Server $ source envsetup.sh rockchip_rk3568_uvc
Server $ make
```

3.7 Firmware packaging

Firmware packaging command: `./mkfirmware.sh`

Firmware directory: rockdev

3.8 Fully automatic compiling

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

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

4. Flashing Instruction

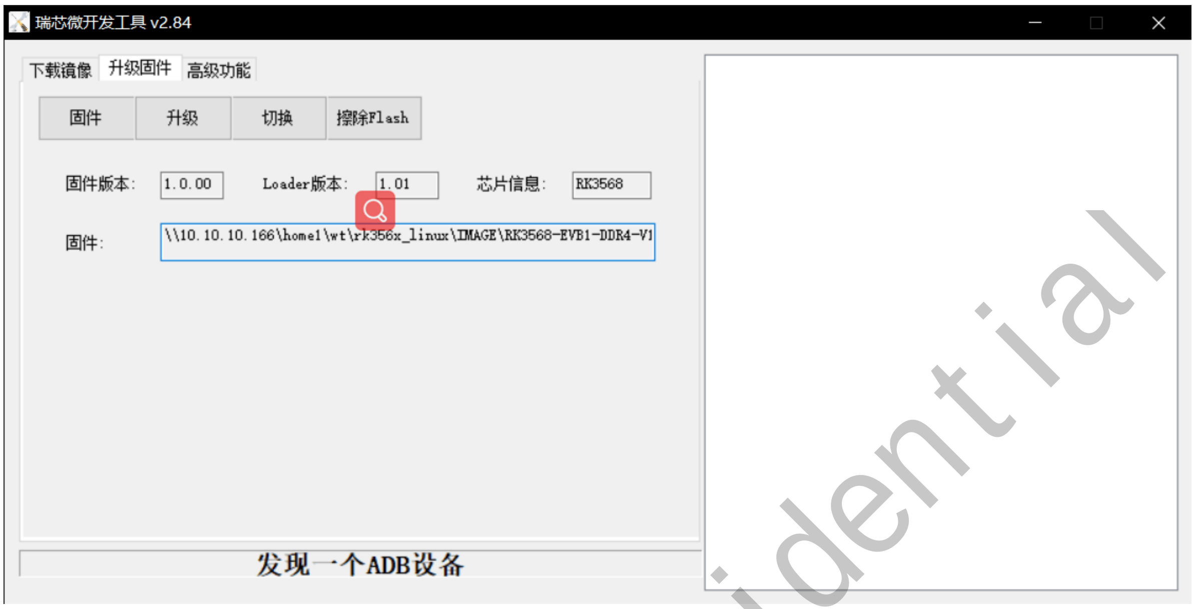
4.1 Windows flashing instructions

The SDK provides Windows burn tool (tool version V2.84 or above) , the tool is located in the project root directory:

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

After compiling and generating the firmware, the device needs to be into MASKROM or BootRom programming mode. After connecting the USB download cable, press and hold the "Maskrom" button and press the "RESET" button, then release to enter the MASKROM mode. After loading the path of firmware, click "Execute" to program, or press and hold "V+" button and press the "RESET" button, then release to enter the loader mode to program.

Here is how to burn Update.img (Note: Windows PC needs to run the tool with administrator privileges)



固件版本: 1.0.00 Loader版本: 1.01 芯片信息: RK3568

固件: \\10.10.10.166\home1\wt\rk356x_linux\IMAGE\RK3568-EVB1-DDR4-V1

发现一个ADB设备

Note: Before flashing, need to install newest USB driver, see details:

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

4.2 Linux flashing instructions

The Linux program tool is located in tools/linux (Linux_Upgrade_Tool version needs to be V1.49 or above), make sure your board is connected to MASKROM/loader rockusb. For example, the compiled firmware is located in rockdev directory, the upgrade command is as follows:

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

Or upgrade the entire update.img firmware:

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

Or, in the root directory, run the following upgrade with the machine in MASKROM state:

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

5. EVB Board Function Instruction

The EVB board USB Camera firmware supports the following features:

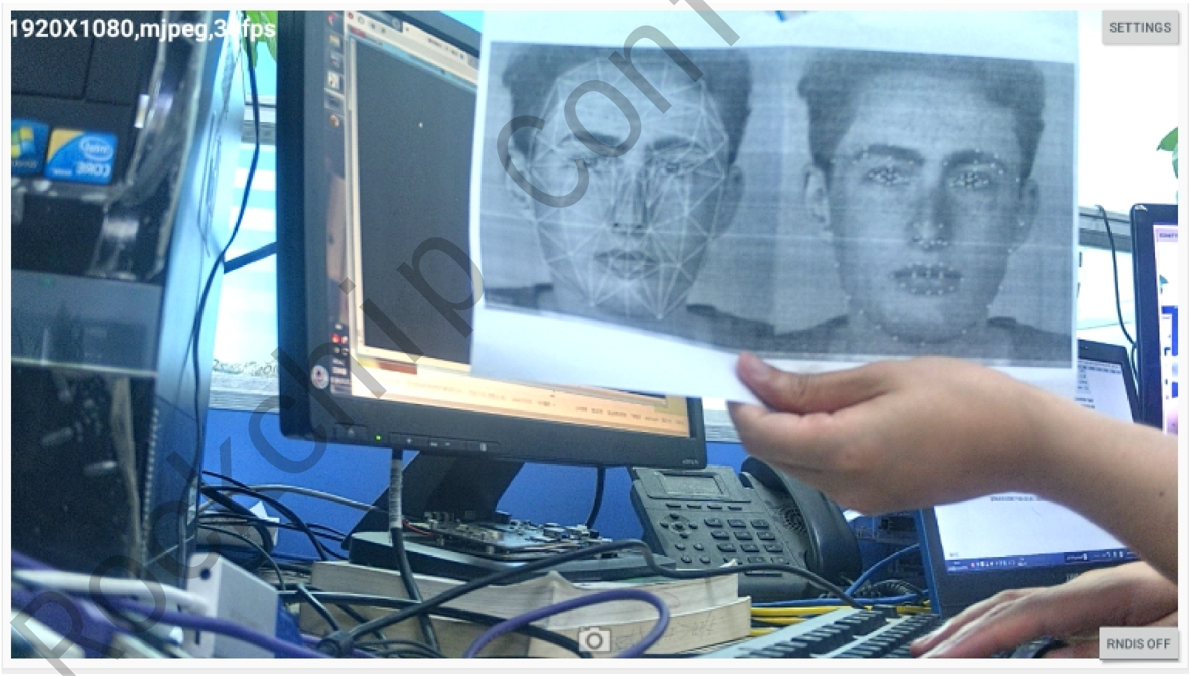
- Support standard UVC Camera function with up to 4k preview (RK356X)
- Support USB composite device stable transmission
- Support multiple terminal devices such as smart TV or PC to preview

5.1 How to display the main camera preview

The configuration of the PC which connected with the EVB board by the serial port is as follows:

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

PC run the amcap/PolPlayer or other USB camera application, Android Host run RKAICameraTes or other standard camera application, you can preview when open it. Please refer to the host application for switching format or resolution.



5.2 UVC project

RK356X supports UVC project, the following patch needs to be applied to the kernel first:

```
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-u1u2-quirk;
        extcon = <&usb2phy0>;
        status = "okay";
 };
```

5.3 HDMI display

RK356X can support external HDMI display, if necessary, you need to open the vop module in the kernel.

```
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 Hibernation wake-up

RK356X can support IR hibernation wake-up, if needed, you need to configure suspend in kernel dts.

```
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. Application Software Framework

The RK356X application corresponds to the source code program as follows:

> **1.aiserver corresponds to /app/aiserver: Send one camera data to uvc/gpu/vo to realize usb camera/ distortion correction /HDMI display and so on;**
>
> **2.uvc_app corresponds to /external/uvc_app: Implement and control the full function of UVC camera.**

6.1 uvc_app

lease refer to the following documents:

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

6.2 aiserver

lease refer to the following documents

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

6.3 Other

For other linux application frameworks or modules, please refer to the following documents:

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

7. FAQ

7.1 Reprogram a module

Taking the mpp module as an example, you can run the following command to reprogram:

```
make mpp-rebuild
```

7.2 GDB debugging

Open buildroot's gdb configuration, then recompile the packaged filesystem.

```
# buildroot directory
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-uvc/"
```

Note: When running gdb, you need to configure the signals manually after gdb starts.

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