Rockchip RK628 For All Porting Guide

ID: RK-YH-YF-771

Release Version: V2.3

Release Date: 2024-01-03

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

DISCLAIMER

Trademark Statement

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

RK628 supports RGB/HDMI/BT1120 for input, and supports RGB / BT1120 / LVDS / GVI / DSI / CSI / HDMI and so on for output, please refer to datasheet for details. This document mainly describes the software configuration, debugging methods and common problem and resolution of rk628-for-all. It's expected that the logs of rk628-for-all are independent of hardware platforms and software versions, and the logs are still improving. Currently, the logs contain two parts, MISC (at drivers/misc/rk628/, supports RGB / BT1120 / HDMI input and RGB / BT1120 / LVDS / GVI / DSI / HDMI output) and MEDIA (at drivers/media/i2c/rk628/, supports HDMI input and CSI / BT1120 output), and are maintained separately. This document is explained in accordance with the different modules.

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/ LuoWei/RangLingpeng/GuoLinan	2021-08-28	Initial version
V1.1	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/ LuoWei/RangLingpeng/GuoLinan	2021-09-03	Add DTS DEMO
V1.2	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/ LuoWei/RangLingpeng/GuoLinan	2021-09-17	Complete FAQ
V1.3	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/ LuoWei/RangLingpeng/GuoLinan	2021-09-29	Complete FAQ of AUDIO
V1.4	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/ LuoWei/RangLingpeng/GuoLinan	2021-10-08	Add HDMIOUT description to MISC
V1.5	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/ LuoWei/RangLingpeng/GuoLinan	2021-10-27	Add DSI->CSI description and support for pinctrl
V1.6	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan	2021-11-02	Add basic hardware information
V1.7	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan	2021-11-26	Support figurating src_mode parameter according to source and generating edid data of HDMIRX automatically
V1.8	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan	2022-01-15	Add the description for part of FAQ and BT1120
V1.9	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan	2022-03-05	Complete the frequency point list supported by HDMIRX, add descriotion for multi-rk628 function to MISC
V2.0	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan	2022-04-10	Complete the solution for sound card login failure and HDMI-IN upper layer problems
V2.1	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan	2022-12-10	Add ideas and solutions for part of HDMI-IN and HDMI2GVI problems
V2.2	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan/ HuangZhibin/FanJianwei	2024-01-01	Add support for RK628F driver

Version	Author	Date	Change Description
V2.3	HuangGuochun/ChenShunqing/ LanShunhua/WenDingxian/LuoWei/ RangLingpeng/GuoLinan/HuangXiongshan/ HuangZhibin/FanJianwei	2024-01-03	Add support for RK628H driver

Contents

Rockchip RK628 For All Porting Guide

1. Preface
    - 1.1 RK628 block diagram
    - 1.2 RK628 typical design hardware diagram
    - 1.3 RK628 main specification
        - 1.3.1 RK628D main specification
        - 1.3.2 RK628F main specification
        - 1.3.3 RK628H main specification
    - 1.4 Improvement and optimization comparison of RK628F/H and RK628D
2. Communication of SOC and RK628 I2C
3. Misc
    - 3.1 MISC Driver Introduction
        - 3.1.1 Driver migration
        - 3.1.2 Driver contents structure
        - 3.1.3 Overview of DTS configuration
    - 3.2 Driver core configuration
        - 3.2.1 RK628 node configuration
        - 3.2.2 24MHz working clock configuration
        - 3.2.3 Input/output module selection
        - 3.2.4 Panel-end configuration
        - 3.2.5 Different display combination path applications
    - 3.3 Input Module Configuration
        - 3.3.1 RGB Input
            - 3.3.1.1 RGB input configuration
        - 3.3.2 BT1120 Input
            - 3.3.2.1 BT1120 input configuration
        - 3.3.3 HDMI Input
            - 3.3.3.1 HDMI input configuration
            - 3.3.3.2 Connected with Soc
            - 3.3.3.3 Connected by HDMI
            - 3.3.3.4 Audio configuration
    - 3.4 Output Module Configuration
        - 3.4.1 RGB output
            - 3.4.1.1 RGB output configuration
        - 3.4.2 BT1120 output
            - 3.4.2.1 BT1120 output configuration
        - 3.4.3 DSI Output
            - 3.4.3.1 DSI Output Configuration
        - 3.4.4 LVDS Output
            - 3.4.4.1 LVDS Output Configuration
        - 3.4.5 GVI Output
            - 3.4.5.1 GVI output configuration
            - 3.4.5.2 RK628F GVI test list
        - 3.4.6 HDMI Output
            - 3.4.6.1 HDMI output configuration
            - 3.4.6.2 Audio configuration
    - 3.5 Several Combinations Supported by DTS Currently
        - 3.5.1 RGB -> DSI Switch
        - 3.5.2 RGB -> LVDS Switch
        - 3.5.3 RGB -> GVI Switch
        - 3.5.4 RGB -> HDMI Switch
        - 3.5.5 HDMI -> DSI Switch
        - 3.5.6 HDMI -> LVDS Switch
        - 3.5.7 HDMI -> GVI Switch
    - 3.6 Basic Debug Command

3.6.1 Register debugging node
3.6.2 Self-test mode command
3.6.3 RGB IN debugging command (only applies for RK628F/H)
3.7 Common Display Questions and Resolution
3.7.1 Fail to generate the regmap node
3.7.2 I2C Communication Exception
3.7.3 DSI or GVI Display with blur
3.7.4 DSI can't display after rising the clock-frequency
3.7.5 Display offset
3.7.6 The operation of RK628 GPIO
4. Media
4.1 Driver Introduction
4.2 RK628F/H improvements compared with RK628D
4.3 Transplant Description
4.4 HDMI IN VIDEO Framework Description
4.4.1 HDMI IN APK Process
4.4.2 RK628 Driver Framework
4.5 dts Configuration Description
4.5.1 RK628 Node Configuration
4.5.2 Image Receiving Link Combination
4.5.3 HDMI2CSI Link
4.5.3.1 RK628D
4.5.3.2 RK628F/H
4.5.4 HDMI2DSI Switching
4.5.4.1 RK628D
4.5.4.2 RK628F/H
4.5.5 HDMI2BT1120 Switching
4.6 Enable HDCP Function
4.7 Enable scaler Function
4.8 csi supports 2 lanes
4.9 Continuous MIPI mode and non-continuous MIPI
4.10 Dual MIPI mode configuration (only RK628F/H supports)
4.10.1 Dual MIPI split mode explanation
4.10.2 Dual MIPI mode configuration
4.10.2.1 RK628F/H configuration
4.10.2.2 SOC driver code version requirements
4.11 Reception capability for different chip platforms
4.11.1 The method of isp overclock configuration
4.11.2 The method for configuring ISP by CMA memory
4.12 EDID Configuration Method
4.13 camera3_profiles.xml configuration
4.14 HDMI IN APK Adaptation Method
4.14.1 Android 9/10/11 Version
4.14.1.1 Get and Compile APK Source Codes
4.14.1.2 APK Source codes Adaptation
4.14.1.3 Preparation before APK debugging
4.14.2 Android 12+ version
4.14.2.1 APK source
4.14.2.2 APK preview explanation
4.14.2.3 The differences of TIF preview and camera preview
4.15 Driver Debugging Method
4.15.1 Get debugging tool
4.15.2 Examples for debugging command
4.16 Audio Module Introduction
4.16.1 HDMIRX
4.16.2 HDMITX
4.16.3 Common Audio Problems and Solutions
4.16.3.1 I2S without output

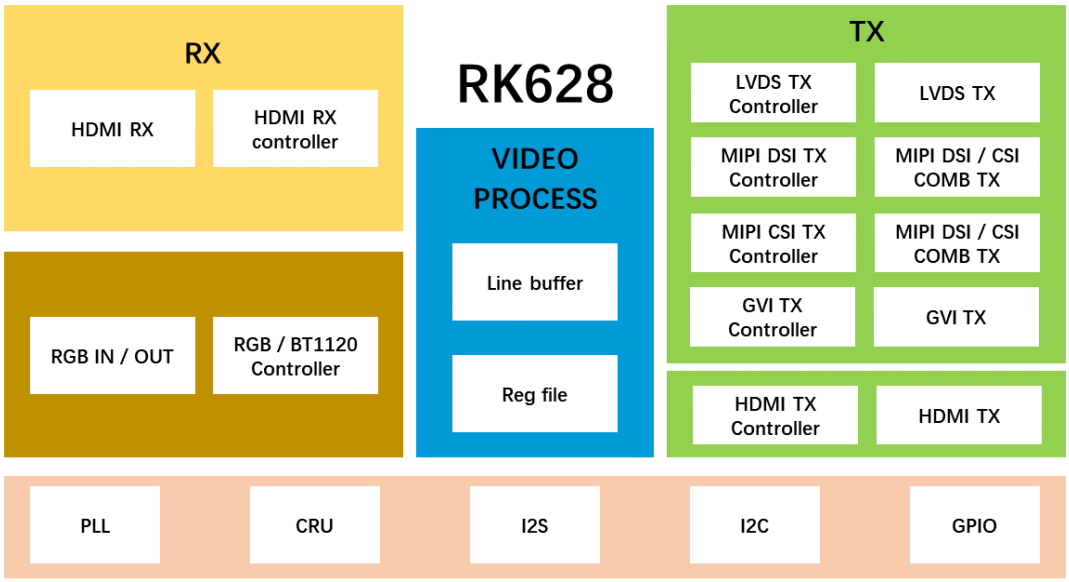
4.16.3.2 Enable to print v4l2_dbg
4.16.3.3 Application recording data noise problem
4.16.3.4 Set IOMUX directly
4.16.3.5 tmdsclk miscalculation
4.16.3.6 Set GPIO to output test clk
4.16.3.7 IOMUX Special Handing of RK356X
4.16.3.8 LRCK Special Handing of RK3399
4.16.3.9 HDMI-IN Sound card Select Error
4.16.4 Other Audio Documents Supplement
4.17 Common Problem Debugging
4.17.1 Open log switch
4.17.2 Register Write and Read
4.17.3 clk det abnormal problems
4.17.3.1 RK628D
4.17.4 Judging whether HDMI RX is normal
4.17.5 Open subdev Permission Exception
4.17.6 Can not recognize the signal
4.17.7 Display Anomaly
4.17.8 Only half of the screen is displayed
4.17.9 Capture Failure
4.17.10 APK Starting Failure
4.17.11 APK Preview Failure when DTS Connecting with rkcif
4.17.12 How to operate the GPIO of RK628
4.17.13 Sound Card Register Failure
5. Common requirements
5.1 The configuration of RK628 24M Crystal oscillator which comes from other SOC
5.1.1 Add 24M support to RK3399
5.1.2 Add 24M support to RK3288
5.1.3 Add 24M support to RK356X
5.2 Dual-RK628 Support
5.2.1 HDMI2CSI+HDMI2CSI support
5.2.1.1 Notes:
5.2.1.2 kernel dts configuration problems
5.2.1.3 android configuration questions
5.2.2 **HDMI2CSI+HDMI2DSI** support
5.2.2.1 kernel dts configuration
5.2.2.2 android configuration

1. Preface

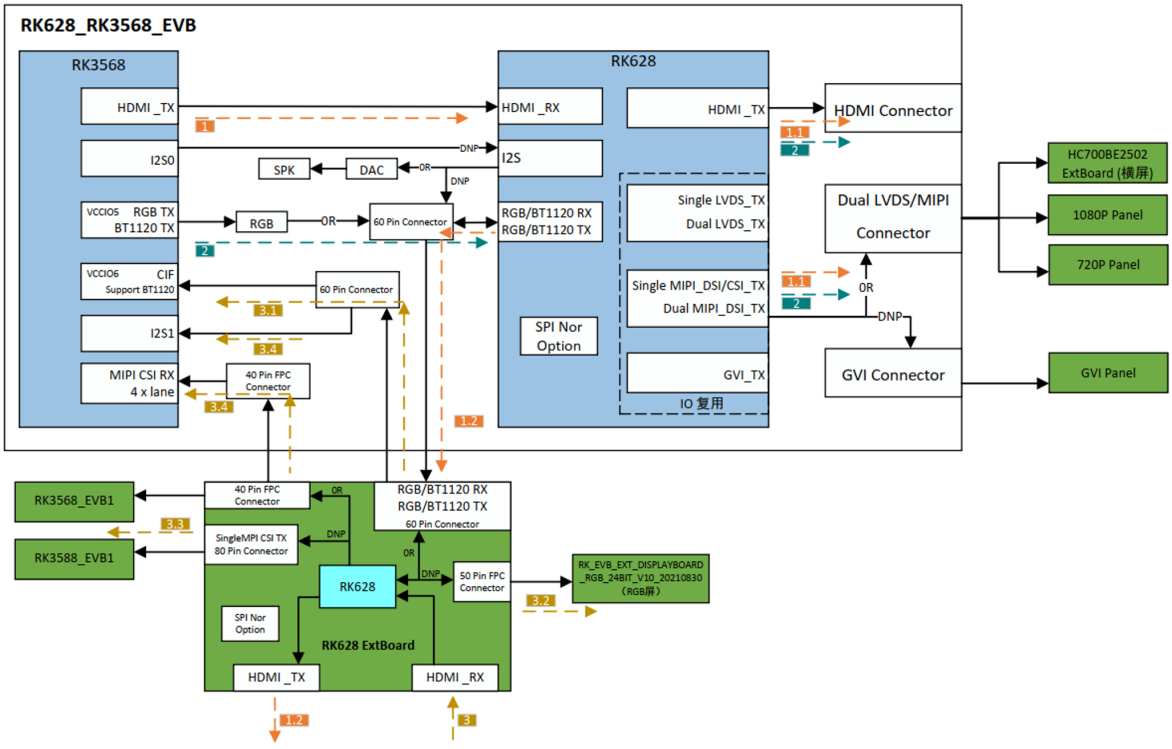
RK628 is divided into Display channel and HDMI IN channel. In SDK version, Display channel is based on DRM framework, while HDMI IN channel is based on V4L2 framework, different frameworks or different kernel versions need different drivers to adapt, and it's only suitable for RK platforms. For adapting every platform and making driver migration become more convenient, the For-All version is released. Current driver supports RK628D and RK628F/H, you can refer to 'Instruction for rk628 driver version' of sdk for patch version management of driver.

For-All version is also divided into Display channel and HDMI IN channel. The driver of Display channel is at drivers/misc/rk628/, and supports RGB / BT1120 / HDMI input and RGB / BT1120 / LVDS / GVI / DSI / HDMI output. The driver of HDMI IN channel is at drivers/media/i2c/rk628/. supports HDMI input and CSI / BT1120 output. Misc and Media followed represent this two drivers respectively, which are independent of each other and can be compiled and run separately.

1.1 RK628 block diagram



1.2 RK628 typical design hardware diagram



1.3 RK628 main specification

1.3.1 RK628D main specification

Video Iutput Interface	**HDMI/BT1120/RGB**	**Typical Format**
HDMI IN	4K@30	YUV420(4K@60) YUV422/YUV444/RGB888
BT1120 IN	1080P@60	YUV422 8bit
RGB IN	1080P@60	RGB888
VIDEO Output Interface	**HDMI/GVI/MIPI/LVDS/RGB/BT1120**	**Typical Format**
GVI OUT	4K@60	RGB888
MIPI CSI	4K@30	YUV422 8bit
MIPI DSI	1080P@60/2.5K@60	RGB888
LVDS OUT	720P@60/1080P@60	RGB888
BT1120 OUT	1080P@60	YUV422 8bit
RGB OUT	1080P@60	RGB888
HDMI OUT	1080P@60	RGB888/YUV444 8bit

1.3.2 RK628F main specification

Video Iutput Interface	**Typical Resolution**	**Typical Format**
HDMI IN	4K@60	RGB888 / YUV422 8bit / YUV422 10bit / YUV420 8bit / YUV420 10bit
BT1120 IN	1080P@60	YUV422 8bit
RGB IN	1080P@60	RGB888
VIDEO Output Interface	**Typical Resolution**	**Typical Format**
GVI OUT	4K@60	RGB888
MIPI CSI	Dual MIPI, 4K@60	YUV422 8bit
MIPI DSI	1080P@60 / 2.5K@60	RGB888
LVDS OUT	720P@60 / 1080P@60	RGB888
BT1120 OUT	1080P@60	YUV422 8bit
RGB OUT	1080P@60	RGB888
HDMI OUT	1080P@60	RGB888 / YUV444 8bit

1.3.3 RK628H main specification

Compare with RK628F, RK628H doesn't have GVI interface.

Video Iutput Interface	**Typical Resolution**	**Typical Format**
HDMI IN	4K@60	RGB888 / YUV422 8bit / YUV422 10bit / YUV420 8bit / YUV420 10bit
BT1120 IN	1080P@60	YUV422 8bit
RGB IN	1080P@60	RGB888
VIDEO Output Interface	**Typical Resolution**	**Typical Format**
MIPI CSI	Dual MIPI, 4K@60	YUV422 8bit
MIPI DSI	1080P@60 / 2.5K@60	RGB888
LVDS OUT	720P@60 / 1080P@60	RGB888
BT1120 OUT	1080P@60	YUV422 8bit
RGB OUT	1080P@60	RGB888
HDMI OUT	1080P@60	RGB888 / YUV444 8bit

1.4 Improvement and optimization comparison of RK628F/H and RK628D

Module	RK628D	RK628F/H
HDMIRX	Up to 4K@60 YUV420 TMDS CLK 297M	Up to 4K@60 RGB/YUV444 TMDS CLK 594M
	There are frequency restrictions, and specific resolution applications need to be evaluated, which may not meet the requirements	Support any frequency points from 27M to 594M to meet diverse resolution input
	Not support DVI MODE	Support DVI MODE
	Unable to detect and determine the input signal with yuv422/yuv420 video format	The controller can determine the video format of input signal
	When the amplitude of the clk lane of the input source is low, the RK628D rxphy cannot lock	Resolved
	There is error detection in HDMI RX	Resolved
MIPI CSI	Single channel 4lanes	Dual channel 8lanes
	Up to 4k@30 yuv422 8bit	Up to 4k@60 yuv422 8bit
GVI	Different types of screens have problems that the screen can't light up and other abnormalities	We have tested multiple models of screens available on the market and have not found any abnormalities.Please refer to the section of 'Output Module Configuration' - 'GVI Output' - 'RK628F GVI Test Model List' for guidance.
	Requires clock homology	No need for clock homology
audio	For the mclk without audio,if it connects to codec, then the host controller need provide mclk	Provide mclk
	I2S FIFO overflow and channel misswitching exist during HDMI plugging and unplugging	Solve the problem of inconsistent channel order when FIFO overflows
csc	For the different input and output interfaces, in some application scenarios, there may be slight deviations in image color when color gamut space conversion or upsampling/downsampling of image formats	Resolved
hdmi tx	Requires clock homology	Requires clock homology for rgb2hdmi No need for clock homology for hdmi2hdmi
multi-display	Support rgb/bt1120 and hdmi tx to output at the same time at the same resolution	Support rgb/bt1120, hdmi tx and gvi/dsi/csi/lvds to output at the same time at the same resolution
MCU	Without MCU inside	MCU inside

2. Communication of SOC and RK628 I2C

The SOC communicates with the RK628 by I2C. The typical 7bit I2C address of the RK628 is 0x50. When multiple RK628s are used on a same I2C bus, the I2C address can be changed through the GPIO of the RK628. Please refer to the following:

The i2c address consists of 7 bits, where the upper four bits are the identifier of the i2c device and the value is set to 4b'1010, the lower three bits are the device address. In order to meet the application of different scenarios, the I2C slave device address can be programmed through GOIO, the mapping of address to GPIO show as Table 5-1, the typical slave address is 7'b1010000.

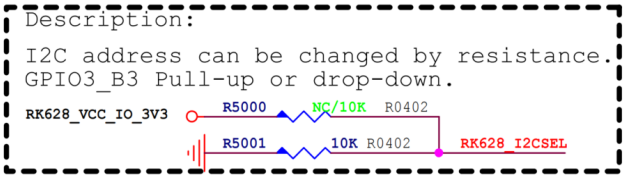
Table 5-2 Mapping of i2c slave address to GPIO

Addr bit	Pad Name	GPIO Setting
cfg_slvadr[2]	IO_GPIO0a1	GPIO0A_OE[1]=1'b1
cfg_slvadr[1]	IO_GPIO0a0	GPIO0A_OE[0]=1'b1
cfg_slvadr[0]	IO_GPIO3b3	GPIO3B_OE[2]=1'b1

Take 0x50 and 0x51 for example:

I2C address configuration:

I2C_SEL	I2C_ADDR
0	7'b1010000(0x50)
1	7'b1010001(0x51)



If GPIO0a0 and GPIO0a1 remain low and GPIO3b3 is pulled high, then the slave address is 0x51 (7'b1010001). If GPIO3b3 is pulled low, the slave address is 0x50 (7'b1010000).

3. Misc

3.1 MISC Driver Introduction

RK628 Misc driver is a driver code for RK628 Display channel, which implements display protocol conversion with RK628. It supports RGB, BT1120 or HDMI for input and LVDS, GVI, DSI, RGB, BT1120 or HDMI for output.

3.1.1 Driver migration

On RK platform, if you are adding the RK628 Misc driver for the first time or there is not RK628 driver under drivers/misc, you can directly copy the Misc-related RK628 driver to drivers/misc, modify driver/misc/Kconfig and drivers/misc/Makefile, and add the compilation of rk628:

```
diff --git a/drivers/misc/Kconfig b/drivers/misc/Kconfig
index 276c7c4fef15..8481b9613bc9 100644
--- a/drivers/misc/Kconfig
+++ b/drivers/misc/Kconfig
@@ -5,6 +5,8 @@

 menu "Misc devices"

+source "drivers/misc/rk628/Kconfig"
+
 config RK803
        tristate "RK803"
        default n
```

```
diff --git a/drivers/misc/Makefile b/drivers/misc/Makefile
index b296e760fd47..14642c54a2bc 100644
--- a/drivers/misc/Makefile
+++ b/drivers/misc/Makefile
@@ -3,6 +3,7 @@
 # Makefile for misc devices that really don't fit anywhere else.
 #

+obj-y                          += rk628/
 obj-$(CONFIG_RK803)            += rk803.o
 obj-y                          += rockchip/
 obj-$(CONFIG_LT7911D_FB_NOTIFIER) += lt7911d-fb-notifier.o
```

RK628 driver of Misc will be compiled by default, you just need select to run the corresponding driver by configuring dts.

3.1.2 Driver contents structure

```
drivers/misc/rk628/
├── Kconfig
├── Makefile
├── panel.c
├── panel.h
├── rk628.c
├── rk628_combrxphy.c
├── rk628_combrxphy.h
├── rk628_combtxphy.c
├── rk628_combtxphy.h
├── rk628_config.c
├── rk628_config.h
├── rk628_cru.c
├── rk628_cru.h
├── rk628_csi.c
├── rk628_csi.h
├── rk628_dsi.c
├── rk628_dsi.h
├── rk628_efuse.c
├── rk628_efuse.h
├── rk628_gpio.h
├── rk628_grf.h
```

```
├── rk628_gvi.c
├── rk628_gvi.h
├── rk628.h
├── rk628_hdmirx.c
├── rk628_hdmirx.h
├── rk628_hdmitx.c
├── rk628_hdmitx.h
├── rk628_lvds.c
├── rk628_lvds.h
├── rk628_pinctrl.c
├── rk628_pinctrl.h
├── rk628_post_process.c
├── rk628_post_process.h
├── rk628_rgb.c
├── rk628_rgb.h
```

3.1.3 Overview of DTS configuration

Take RGB -> DSI as an example:

```
&i2c2 {
    clock-frequency = <400000>;
    status = "okay";

    i2c2_rk628: rk628@50 {
        compatible = "rockchip,rk628";
        reg = <0x50>;
        interrupt-parent = <&gpio0>;
        interrupts = <20 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio0 RK_PC1 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio2 RK_PA2 GPIO_ACTIVE_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_reset>;
        status = "okay";
    };
};

&pinctrl {
    rk628 {
        rk628_reset: rk628-reset {
            rockchip,pins = <2 RK_PA2 RK_FUNC_GPIO &pcfg_pull_none>;
        };
    };
};

&i2c2_rk628 {
    /* soc_24M optional */
    pinctrl-names = "default";
    pinctrl-0 = <&rk628_reset &refclk_pins>;
    assigned-clocks = <&pmucru CLK_WIFI>;
    assigned-clock-rates = <24000000>;
    clocks = <&pmucru CLK_WIFI>;
    clock-names = "soc_24M";

    panel-backlight = <&backlight>;
```

```
    panel-power-supply = <&vcc3v3_lcd0_n>;
    panel-enable-gpios = <&gpio2 RK_PC6 GPIO_ACTIVE_HIGH>;

    panel-reset-delay-ms = <10>;
    panel-enable-delay-ms = <10>;
    panel-prepare-delay-ms = <60>;
    panel-unprepare-delay-ms = <10>;
    panel-disable-delay-ms = <60>;
    panel-init-delay-ms = <60>;

    rk628,rgb-in;
    rk628-dsi {
        dsi,eotp;
        dsi,video-mode;
        dsi,format = "rgb888";
        dsi,lanes  = <4>;
        status = "okay";

        rk628-panel {
            panel-init-sequence = [
                05 78 01 11
                05 78 01 29
            ];

            panel-exit-sequence = [
                05 00 01 28
                05 00 01 10
            ];
        };
    };

    display-timings {
        src-timing {
            clock-frequency = <148500000>;
            hactive = <1920>;
            vactive = <1080>;
            hfront-porch = <88>;
            hsync-len = <44>;
            hback-porch = <148>;
            vfront-porch = <4>;
            vsync-len = <5>;
            vback-porch = <36>;
            hsync-active = <1>;
            vsync-active = <1>;
            de-active = <0>;
            pixelclk-active = <0>;
        };

        dst-timing {
            clock-frequency = <148500000>;
            hactive = <1920>;
            vactive = <1080>;
            hfront-porch = <88>;
            hsync-len = <44>;
            hback-porch = <148>;
            vfront-porch = <4>;
            vsync-len = <5>;
            vback-porch = <36>;
```

```
            hsync-active = <1>;
            vsync-active = <1>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };
};
```

3.2 Driver core configuration

3.2.1 RK628 node configuration

The SOC uses I2C and RK628 to communicate, taking the RK3568 as an example, add the RK628 node (i2c2_rk628: rk628@50) under the I2C node (i2c2) of the DTS:

```
&i2c2 {
    clock-frequency = <400000>;
    status = "okay";

    i2c2_rk628: rk628@50 {
        compatible = "rockchip,rk628";
        reg = <0x50>;
        interrupt-parent = <&gpio0>;
        interrupts = <20 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio0 RK_PC1 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio2 RK_PA2 GPIO_ACTIVE_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_reset>;
        status = "okay";
    };
};

&pinctrl {
    rk628 {
        rk628_reset: rk628-reset {
            rockchip,pins = <2 RK_PA2 RK_FUNC_GPIO &pcfg_pull_none>;
        };
    };
};
```

RK628 node property

Property	Description	Option Value
compatible	compatible	The compatible of RK628 Misc driver needs to be set as "rockchip,rk628"
reg	RK628 I2C slave address	0x50/0x51, configure according to the hardware
interrupt-parent	RK628 interrupt reference	Configure according to the hardware
interrupts		
enable-gpios	RK628 enabling GPIO reference	Configure according to the hardware
reset-gpios	RK628 reset GPIO reference	Configure according to the hardware
pinctrl-names	set the multiplexing function of RK628 reset pin as GPIO	Configure according to the hardware
pinctrl-0		

3.2.2 24MHz working clock configuration

The 24MHz working clock of RK628 can be taken from an external 24M crystal oscillator on the hardware or from the 24MHz CLK pin of the SOC, and the following configuration can be ignored by using an external 24M crystal as the operating clock.

It should be noted that in the corresponding path scenarios of **RK628D HDMI / GVI OUT** and **RK628F/H RGB ->HDMI**, the 24MHz CLK pin of the SOC output is required as the working clock of the RK628 to achieve homology clock. Connecting an external 24M crystal oscillator as the working clock may cause the display to be abnormal in this scenario. (RK628F/H is optimized for homology and uses the TMDS CLK detected by the HDMI RX as the reference clock for the CRU PLL, so there is no need to configure the 24MHz CLK from the SOC as the reference clock in the HDMI IN path scenario)

Taking RK3568 as an example (for more RK platform configurations, please refer to the chapter "[RK628 24M crystal oscillator from other SOC configurations]"), RK3568 can output more pins of 24MHz, such as REF_CLKOUT (clk_wifi / gpio0_a0), CAM_CLKOUT1 (clk_cam1_out / gpio4_b0), ETH_REFCLK_25M_M0 (clk_mac1_out / gpio3_b0), let's take the first one as an example.

1. Reference the following configuration in the RK628 node of DTS

```
&i2c2_rk628 {
    /* soc_24M optional */
    pinctrl-names = "default";
    pinctrl-0 = <&rk628_reset &refclk_pins>;
    assigned-clocks = <&pmucru CLK_WIFI>;
    assigned-clock-rates = <24000000>;
    clocks = <&pmucru CLK_WIFI>;
    clock-names = "soc_24M";
    ......
};
```

2. The following configuration comes with rk3568-pinctrl.dtsi

```
&pinctrl {
    ......
    refclk {
        /omit-if-no-ref/
        refclk_pins: refclk-pins {
            rockchip,pins =
                /* refclk_ou */
                <0 RK_PA0 1 &pcfg_pull_none>;
        };
    };
    ......
};
```

24MHz CLK property

pinctrl-names	set the multiplexing function of refclk pin	Configure according to the hardware (it duplicates the RK628 node property above, so rk628_reset reference should be added)
pinctrl-0		
assigned-clocks	The clock reference assigned to the RK628	Configure according to the hardware
assigned-clock-rates	The clock rate allocated for the RK628 (Hz)	<24000000>
clocks	The handle list of reference clock	Configure according to the hardware
clock-names	set the name for this clock	"soc_24M"

3.2.3 Input/output module selection

In the RK628 node of DTS, configure the corresponding module properties (bool type) or nodes according to the requirements of the input and output modules, and configure the corresponding functional parameters of the module in the module node.

```
&i2c2_rk628 {
    ......
    rk628,rgb-in;
    rk628-dsi {
        ......
    };
    ......
};
```

Taking the RGB-> DSI path as an example, set the "rk628,rgb-in;" attribute to configure the RGB module as the input, and set the "rk628-dsi" node to configure the DSI module as the output. Currently, the Misc input module is RGB IN, BT1120 IN, or HDMI IN, and the output module is LVDS, GVI, DSI, or HDMI. For details of the configuration of each input and output module, please refer to the sections "[Input Module Configuration]and [Output Module Configuration].

3.2.4 Panel-end configuration

In the RK628 node of DTS, configure screen-related attributes according to the screen spec, including backlight, power supply, reset pin, on-screen power-on timing, display timing, etc.:

```
&i2c2_rk628 {
    ......
    panel-backlight = <&backlight>;
    panel-power-supply = <&vcc3v3_lcd0_n>;
    panel-enable-gpios = <&gpio2 RK_PC6 GPIO_ACTIVE_HIGH>;

    panel-reset-delay-ms = <10>;
    panel-enable-delay-ms = <10>;
    panel-prepare-delay-ms = <60>;
    panel-unprepare-delay-ms = <10>;
    panel-disable-delay-ms = <60>;
    panel-init-delay-ms = <60>;
    ......
    display-timings {
        src-timing {
            clock-frequency = <148500000>;
            hactive = <1920>;
            vactive = <1080>;
            hfront-porch = <88>;
            hsync-len = <44>;
            hback-porch = <148>;
            vfront-porch = <4>;
            vsync-len = <5>;
            vback-porch = <36>;
            hsync-active = <1>;
            vsync-active = <1>;
            de-active = <0>;
            pixelclk-active = <0>;
        };

        dst-timing {
            clock-frequency = <148500000>;
            hactive = <1920>;
            vactive = <1080>;
            hfront-porch = <88>;
            hsync-len = <44>;
            hback-porch = <148>;
            vfront-porch = <4>;
            vsync-len = <5>;
            vback-porch = <36>;
            hsync-active = <1>;
            vsync-active = <1>;
            de-active = <0>;
            pixelclk-active = <0>;
        };
    };
    ......
};
```

Screen-related property

Property	Description	Option Value
panel-enable-gpios	Screen enables GPIO reference[option]	Configure according to the hardware
panel-reset-gpios	Screen resets GPIO reference [option]	Configure according to the hardware
panel-backlight	Screen backlight[option]	Configure according to the hardware
panel-reset-delay-ms	On-screen power timing	Refer to the screen specification
panel-enable-delay-ms		
panel-prepare-delay-ms		
panel-unprepare-delay-ms		
panel-disable-delay-ms		
panel-init-delay-ms		
src-timing	The SOC is transferred to the RK628's display timing node	Scale separately by rows and columns with dst-timing
dst-timing	The timing sequence node is displayed on the screen	Refer to the screen specification
clock-frequency	Display timing sequence	Refer to the screen specification and actual scaler
hactive		
vactive		
hfront-porch		
hback-porch		
hsync-len		
vfront-porch		
vback-porch		
vsync-len		
hsync-active		
vsync-active		
de-active		
pixelclk-active		

If the resolution of the input and output of the RK628 is scaled, configure the timing of the input to src-timing and the timing of the output to dst-timing.
Before and after scaling, the timing of the display part needs to be consistent with the scaling ratio of the blanking part:

```
src: hactive / dst: hactive = src: hblanking / dst: hblanking
src: vactive / dst: vactive = src: vblanking / dst: vblanking
hblanking = hfront-porch + hback-porch + hsync-len
vblanking = vfront-porch + vback-porch + vsync-len
```

And the frame rates before and after scaling should be the same.

```
src: clock-frequency / (htotal * vtotal) = dst: clock-frequency / (htotal *
vtotal)
htotal = hactive + hfront-porch + hback-porch + hsync-len
vtotal = vactive + vfront-porch + vback-porch + vsync-len
```

If the input and output resolutions are the same, then src-timing and dst-timing are configured with the same target timing.

If you use two DSI single screens with the same resolution or two LVDS single screens with the same resolution, clock-frequency, hactive, hfront-porch, hback-porch, and hsync-len of the src-timing and dst-timing need to be multiplied by two in the original single-screen configuration.

3.2.5 Different display combination path applications

For more information, please refer to the section [Several combinations supported by DTS currently].

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2bt1120-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2dsi-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2dsi-dual-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2gvi-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2lvds-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2lvds-dual-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2gvi-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2hdmi-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2lvds-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2lvds-dual-ddr4-v10.dts
```

3.3 Input Module Configuration

3.3.1 RGB Input

3.3.1.1 RGB input configuration

1. Add the "rk628,rgb-in;" bool attribute to the RK628 node in DTS:

```
&i2c2_rk628 {
    ......
    rk628,rgb-in;
    ......
};
```

It should be noted that the RGB IN / RGB OUT / BT1120 IN / BT1120 OUT functions share the same pin, so these four functions cannot exist at the same time.

2. The SOC configures the RGB output, taking RK3568 as an example, the RGB output uses the DRM frame, and the RK628 is virtualized as a screen:

```
/ {
    panel@0 {
        compatible = "simple-panel";

        disp_timings3: display-timings {
            native-mode = <&rgb2dsi_timing>;
            rgb2dsi_timing: timing0 {
                clock-frequency = <148500000>;
                hactive = <1920>;
                vactive = <1080>;
                hfront-porch = <88>;
                hsync-len = <44>;
                hback-porch = <148>;
                vfront-porch = <4>;
                vsync-len = <5>;
                vback-porch = <36>;
                hsync-active = <1>;
                vsync-active = <1>;
                de-active = <0>;
                pixelclk-active = <0>;
            };
        };

        port {
            panel_in_rgb: endpoint {
                remote-endpoint = <&rgb_out_panel>;
            };
        };
    };
};

&route_rgb {
    status = "okay";
};

&rgb_in_vp2 {
    status = "okay";
};

&rgb {
    status = "okay";

    ports {
        port@1 {
            reg = <1>;
```

```
            rgb_out_panel: endpoint {
                remote-endpoint = <&panel_in_rgb>;
            };
        };
    };c
};
```

The timing of the "panel@0/display-timings/timing0" node must be the same as that of the "&i2c2_rk628/display-timings/src-timing" node.

3.3.2 BT1120 Input

3.3.2.1 BT1120 input configuration

Refer to "RGB Input Configuration", and change the "rk628,rgb-in;" bool attribute in the RK628 node of DTS to "rk628,bt1120-in;" to:

```
&i2c2_rk628 {
    ......
    rk628,bt1120-in;
    // bt1120-dual-edge;
    ......
};
```

If you want to configure the BT1120 as dual-edge, add the "bt1120-dual-edge;" bool attribute to the RK628 node. It should be noted that the RGB IN / RGB OUT / BT1120 IN / BT1120 OUT functions share the same pin, so these four functions cannot exist at the same time.

3.3.3 HDMI Input

3.3.3.1 HDMI input configuration

Configure in the RK628 node of DTS:

```
&i2c2_rk628 {
    ......
    rk628,hdmi-in;
    // plugin-det-gpios = <&gpio2 RK_PA4 GPIO_ACTIVE_LOW>;
    // hpd-output-inverted;
    // src-mode-4k-yuv420;
    ......
};
```

There are two types of HDMI input now, one is connected with Soc directly, the other one is connected by HDMI.

3.3.3.2 Connected with Soc

This module needs to output from source, so hot plug detection pin is no need. It doesn't support solution switching in this case, so the driver detects the HDMI signal by polling, and dts doesn't need to configure 'plugin-det-gpios' property.

SOC specifies an HDMI source with a specific resolution output:

SOC configures HDMI output, take RK3568 as an example, configure HDMI as the specified output, you can specify the timing and bus-format of the output source:

```
#include <dt-bindings/display/media-bus-format.h>

&hdmi {
    status = "okay";
    force-bus-format = <MEDIA_BUS_FMT_RGB888_1X24>;
    force-output;
    force_timing{
        clock-frequency = <594000000>;
        hactive = <3840>;
        vactive = <2160>;
        hback-porch = <296>;
        hfront-porch = <176>;
        vback-porch = <72>;
        vfront-porch = <8>;
        hsync-len = <88>;
        vsync-len = <10>;
        hsync-active = <1>;
        vsync-active = <1>;
        de-active = <0>;
        pixelclk-active = <0>;
    };
};

&hdmi_in_vp0 {
    status = "okay";
};

&hdmi_in_vp1 {
    status = "disabled";
};

&route_hdmi {
    status = "okay";
    force-bus-format = <MEDIA_BUS_FMT_RGB888_1X24>;
    force-output;
    force_timing{
        clock-frequency = <594000000>;
        hactive = <3840>;
        vactive = <2160>;
        hback-porch = <296>;
        hfront-porch = <176>;
        vback-porch = <72>;
        vfront-porch = <8>;
        hsync-len = <88>;
        vsync-len = <10>;
        hsync-active = <1>;
```

```
        vsync-active = <1>;
        de-active = <0>;
        pixelclk-active = <0>;
    };
};
```

Enabling '&route_hdmi' node is to show the kernel logo of boot for HDMIRX, the 'force-output', 'force-bus-format', 'force_timing' under "&route_hdmi' needs to be the same as 'hdmi' node.

In this mode, if the HDMI source outputs 4K YUV420 (i.e., the force-bus-format property is set to MEDIA_BUS_FMT_UYYVYY8_0_5X24 and the force_timing is set to 4K resolution), you need to configure the "src-mode-4k-yuv420" bool attribute under the RK628 node, because HDMIRX cannot determine whether the HDMI source is YUV420 output in advance.

When configuring the force-bus-format property, you need to add the header file <dt-bindings/display/media-bus-format.h> to the DTS, which contains various display data formats.

SOC is forced to output HDMI Patch

On the RK SOC platform, the specified resolution of HDMI forced output needs to be patched as follows (you can judge whether the patch has been added by comparing the patch content with the source code content):

patches list	commit message	patches explanation
0190-drm-bridge-dw-hdmi-qp-Support-hdmi-force-output-kernel-5-10.patch	drm/bridge: dw-hdmi-qp: Support hdmi force output	In the **kernel 5.10** of the RK platform, the HDMI TX driver does not have this patch, so it needs to be imported
0190-drm-bridge-synopsys-Support-hdmi-force-output-kernel-5-10.patch	drm/bridge: synopsys: Support hdmi force output	In the **kernel 5.10** of the RK platform, the HDMI TX driver does not have this patch, so it needs to be imported
0190-drm-bridge-synopsys-Support-hdmi-force-output-kernel-4-19.patch	drm/bridge: synopsys: Support hdmi force output	In the **kernel 4.19** of the RK platform, the HDMI TX driver does not have this patch, so it needs to be imported
0190-drm-bridge-synopsys-Support-hdmi-force-output.patch	drm/bridge: synopsys: Support hdmi force output	In the **kernel 4.4** of the RK platform, the HDMI TX driver does not have this patch, so it needs to be imported
0189-drm-bridge-synopsys-dw-hdmi-Support-force-logo-displ.patch	drm/bridge: synopsys: dw-hdmi: Support force logo display	In the **kernel 4.4** of the RK platform, the HDMI TX driver does not have this patch, so it needs to be imported

3.3.3.3 Connected by HDMI

HDMI connection needs hot plug detection and solution/color format switching, so you should configure 'plugin-det-gpios' property under RK628 node in this module by interrupt mode acquiescently.

'hpd-output-inverted' bool property: HPD inverting output configuration. If HPD output-level is inverted in the circuit, this configuration should be enabled:

If you need to modify the resolution of HDMIRX in HDMI cable connection mode, you only need to change the src-timing in the [Panel Configuration] section to the specified resolution.

3.3.3.4 Audio configuration

When used as HDMI input, RK628 outputs audio. I2S acts as the master to output audio data from I2S, and the dts sound card is configured with:

```
rk628_dc: rk628-dc {
    compatible = "rockchip,dummy-codec";
    #sound-dai-cells = <0>;
};

&i2s0 {
    rockchip,capture-only;
    status = "okay";
};

/ {
    hdmiin-sound {
        compatible = "simple-audio-card";
        simple-audio-card,format = "i2s";
        simple-audio-card,name = "rockchip,hdmiin";
        simple-audio-card,bitclock-master = <&dailink0_master>;
        simple-audio-card,frame-master = <&dailink0_master>;
        status = "okay";
        simple-audio-card,cpu {
            sound-dai = <&i2s0>;
        };
        dailink0_master: simple-audio-card,codec {
            sound-dai = <&rk628_dc>;
        };
    };
};
```

SOC I2S is used as a subordinate, when there is no clock signal, the driver will keep waiting, in the Android system, waiting too long is prone to crash, you can add the following patch to modify the timeout:

```
diff --git a/sound/soc/rockchip/rockchip_i2s.c
b/sound/soc/rockchip/rockchip_i2s.c
index 9bc29fdd13c5..230c350c6765
--- a/sound/soc/rockchip/rockchip_i2s.c
+++ b/sound/soc/rockchip/rockchip_i2s.c
@@ -468,6 +468,9 @@ static int rockchip_i2s_trigger(struct snd_pcm_substream
*substream,
                ret = -EINVAL;
                break;
        }
+       if(substream->stream == SNDRV_PCM_STREAM_CAPTURE) {
+               substream->wait_time = msecs_to_jiffies(100);
+       }

        return ret;
 }
diff --git a/sound/soc/rockchip/rockchip_i2s_tdm.c
b/sound/soc/rockchip/rockchip_i2s_tdm.c
index b0c4ce01e2be..0602df42c309
--- a/sound/soc/rockchip/rockchip_i2s_tdm.c
+++ b/sound/soc/rockchip/rockchip_i2s_tdm.c
```

```
@@ -1776,6 +1776,9 @@ static int rockchip_i2s_tdm_trigger(struct
snd_pcm_substream *substream,
                ret = -EINVAL;
                break;
        }
+       if(substream->stream == SNDRV_PCM_STREAM_CAPTURE) {
+               substream->wait_time = msecs_to_jiffies(100);
+       }

        return ret;
 }
```

3.4 Output Module Configuration

3.4.1 RGB output

3.4.1.1 RGB output configuration

Add the "rk628-rgb;" boot attribute to the RK628 node in DTS:

```
&i2c2_rk628 {
    ......
    rk628-rgb;
    ......
};
```

It should be noted that the RGB IN / RGB OUT / BT1120 IN / BT1120 OUT functions share the same pin, so these four functions cannot exist at the same time.

3.4.2 BT1120 output

3.4.2.1 BT1120 output configuration

Add the "rk628-bt1120;" boot attribute to the RK628 node in DTS:

```
&i2c2_rk628 {
    ......
    rk628-bt1120;
    // bt1120-dual-edge;
    ......
};
```

If you want to configure the BT1120 as a dual-edge, add the "bt1120-dual-edge;" bool attribute to the RK628 node.
It should be noted that the RGB IN / RGB OUT / BT1120 IN / BT1120 OUT functions share the same pin, so these four functions cannot exist at the same time.

3.4.3 DSI Output

3.4.3.1 DSI Output Configuration

Add 'rk628-dsi' node to the RK628 node of DTS:

```
&i2c2_rk628 {
    ......
    rk628-dsi {
        // rockchip,lane-mbps = <1100>;
        // rockchip,dual-channel;
        dsi,eotp;
        dsi,video-mode;
        // dsi,clk-non-continuous;
        dsi,format = "rgb888";
        dsi,lanes  = <4>;
        status = "okay";

        rk628-panel {
            panel-init-sequence = [
                ......
                05 78 01 11
                05 78 01 29
            ];

            panel-exit-sequence = [
                05 00 01 28
                05 00 01 10
            ];
        };
    };
    ......
};
```

Property description for 'rk628-dsi' node configuration

Property	Description	Option Value
rockchip,lane-mbps	Specify MIPI data bandwidth [option]	
rockchip,dual-channel	Single/Dual MIPI [option]	Single MIPI by default, add this proper to configure as dual MIPI
dsi,eotp	EOT PACKET [option]	
dsi,video-mode	Video/Command mode [option]	Command module by default, add this property to configure as Video mode
dsi,clk-non-continuous	Continuous/non-continuous clock [option]	Continuous clock by default, add this property to configure as non-continuous clock
dsi,format	DSI data format	"rgb888" / "rgb666" / "rgb666-packed" / "rgb565"
dsi,lanes	DSI lanes	Configure dual MIPI according to single MIPI lanes
panel-init-sequence	Screen init sequence	The first column represents data type, the second column represents mdelays, the third column represents payload_lenth of sending each command, the last columns represent payload of each command
panel-exit-sequence	Screen exit sequence	same as above

Common data types

data type	description	packet size
0x03	Generic Short WRITE, no parameters	short
0x13	Generic Short WRITE, 1 parameters	short
0x23	Generic Short WRITE, 2 parameters	short
0x29	Generic long WRITE,	long
0x05	DCS Short WRITE, no parameters	short
0x15	DCS Short WRITE, 1 parameters	short

3.4.4 LVDS Output

3.4.4.1 LVDS Output Configuration

Add 'rk628-lvds'node to RK628 node of DTS:

```
&i2c2_rk628 {
    ......
    rk628-lvds {
        bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;
        link-type = "single_link";
        status = "okay";
    };
    ......
};
```

Property description for 'rk628-lvds' node configuration

Property	Description	Option Value
bus-format	Bus format	<MEDIA_BUS_FMT_RGB666_1X7X3_SPWG> / <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG> / <MEDIA_BUS_FMT_RGB888_1X7X4_JEIDA> or "jeida_18" / "vesa_24" / "vesa_18"
link-type	Channel type	"single_link" (single channel) "dual_link_odd_even_pixels" (Dual channel, the left and right channels are odd and even channels) "dual_link_even_odd_pixels" (Dual channel, the left and right channels are even and odd channels) "dual_link_left_right_pixels" (Dual channel, the left and right channels are left and right screen) "dual_link_right_left_pixels" (Dual channel, the left and right channels are right and left screen)

3.4.5 GVI Output

3.4.5.1 GVI output configuration

Add the "rk628-gvi" node to the RK628 node in DTS:

```
&i2c2_rk628 {
    ......
    rk628-gvi {
        bus-format = "rgb888";
        gvi,lanes = <8>;
        // rockchip,division-mode;
        // rockchip, gvi-frm-rst;
        status = "okay";
    };
    ......
};
```

"rk628-gvi" node configuration property

Property	Description	Option Value
bus-format	bus farmat	"rgb666" / "rgb888" / "rgb101010" / "yuyv8" / "yuyv10"
gvi,lanes	GVI lanes	1, 2, 4, 8
rockchip,division-mode	one section / two section [option]	The default is one section mode, and the property is added to set to two section mode
rockchip, gvi-frm-rst	enable gvi rst when frame start [option]	The default is disabled, and the added property is set to be enable

3.4.5.2 RK628F GVI test list

TCON	Model	Screen manufacturer	resolution
IN8205A	M280DGJ-L30	Innolux	3840*2160
IN8205A	M315DJJ		
The early stage is IN8908A (Changed to IN8210 at the end of 2021)	S500DJ2-KS5		
The early stage is IN8908A (Changed to IN8210 at the end of 2021)	V500DJ2-KS5		
SW0894A	LC860EQY-FJA5	XUNSHI	
	LC550EGE	LGDisplay	
Hi3231V530	HV550QUB-N8D	BOE	
SW0894A	MZ860004-07	ACT	
	86S13DCX		
TLi2461MC	ACT-MZ750004-07		
LP71517			
SW08006A	848.2		
TLi2380EP	ist-ITV-016	FunTV	
SW08008			
10301	L22650USWL1	Lenove	
10302	HV860QUB-E1D		
CSQ12-B0S	SG8561D03-1		
CSTU01-A0H	H750D4-BA080AA		
CSTU01-A0W	SL(LX)-23		
KV7636-VPP	ST7461D02-6		
KV7626			
AUO-12417			
AUO-12415	ICB-VN65	HiteVision	
NT71120MFG-001			
EK76603E	V650DK-KS5		

iLITEK-2326			
IN8205A			
MST6M60FV	TN2A1FV4LCZZ		
	HS-86AW-L09PA		
	HV650QUB-F70		
	CEJZ650L07Q1	CSOT	
	MT5461D01-1	CSOT	
	BOEI650WQ1		
	UV650QUB-N90		
	LC650EQQ-SMA5		

3.4.6 HDMI Output

Because HDMI needs to support switching different resolutions, RK628 of Misc currently is implemented as the Bridge of DRM framework.

3.4.6.1 HDMI output configuration

Add the "rk628,hdmi-out" bool attribute to the RK628 node in DTS, and binding the "&rgb" node and the RK628 node by "remote-endpoint":

```
&i2c2_rk628 {
    pinctrl-names = "default";
    pinctrl-0 = <&rk628_reset &refclk_pins>;
    assigned-clocks = <&pmucru CLK_WIFI>;
    assigned-clock-rates = <24000000>;
    clock-names = "soc_24M";
    clocks = <&pmucru CLK_WIFI>;

    rk628,rgb-in;
    rk628,hdmi-out;
    status = "okay";

    port {
        rgb_in_hdmi: endpoint {
            remote-endpoint = <&rgb_out_hdmi>;
        };
    };
};

&route_rgb {
    status = "disabled";
};

&rgb_in_vp2 {
    status = "okay";
```

```
};

&rgb {
    status = "okay";

    ports {
        port@1 {
            reg = <1>;

            rgb_out_hdmi: endpoint {
                remote-endpoint = <&rgb_in_hdmi>;
            };
        };
    };
};
```

It should be noted that the HDMITX path needs to implement CLK homology, otherwise the path may be displayed abnormally, and please refer to the section "24MHz Working Clock Configuration" of the "Driver Core Configuration" chapter for the CLK homologous configuration.

3.4.6.2 Audio configuration

```
&i2s0 {
    rockchip,playback-only;
    status = "oaky";
};

/ {
    rk628_sound: rk628-sound {
        compatible = "simple-audio-card";
        simple-audio-card,format = "i2s";
        simple-audio-card,mclk-fs = <128>;
        simple-audio-card,name = "rockchip,hdmi-rk628";
        status = "oaky";
        simple-audio-card,cpu {
            sound-dai = <&i2s0>;
        };
        simple-audio-card,codec {
            sound-dai = <&i2c2_rk628>;
        };
    };
}
```

3.5 Several Combinations Supported by DTS Currently

3.5.1 RGB -> DSI Switch

Refer to DTS:

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-ddr4-v10.dtsi
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
```

The above DTS is based on the rk3568-evb.dtsi to add the function of RK628 to support RGB to MIPI DSI. For specific differences, please refer to the "Introduction to Input Modules" and "Introduction to Output Modules" chapters to compare the codes.

For the RGB of the specified timing of the SOC configuration output, refer to the [RGB Input] (#RGB Input) subsection of the [Input Module Configuration] chapter.

3.5.2 RGB -> LVDS Switch

Refer to DTS:

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-ddr4-v10.dtsi
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2lvds-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2lvds-dual-ddr4-v10.dts
```

The above DTS is based on the rk3568-evb.dtsi to add the function of RK628 to support RGB to MIPI DSI, in which rk3568-evb-rk628-rgb2lvds-ddr4-v10.dts outputs single-channel LVDS, and rk3568-evb-rk628-rgb2lvds-dual-ddr4-v10.dts outputs dual-channel LVDS, please refer to the "Introduction to the Input Module" and "Introduction to the Output Module"sections to compare the code by yourself.

For the RGB of outputting specified timing of the SOC configuration, please refer to the [RGB Input] subsection of the [Input Module Configuration] chapter.

3.5.3 RGB -> GVI Switch

Refer to DTS:

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-ddr4-v10.dtsi
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2gvi-ddr4-v10.dts
```

The above DTS is based on the rk3568-evb.dtsi to add the function of RK628 to support RGB to GVI, please refer to the "Introduction to the Input Module" and "Introduction to the Output Module"sections to compare the code by yourself.

For the RGB of outputting specified timing of the SOC configuration, please refer to the [RGB Input] subsection of the [Input Module Configuration] chapter.

3.5.4 RGB -> HDMI Switch

Refer to DTS:

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-ddr4-v10.dtsi
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2hdmi-ddr4-v10.dts
```

The above DTS is based on the rk3568-evb.dtsi to add the function of RK628 to support RGB to HDMI, please refer to the "Introduction to the Input Module" and "Introduction to the Output Module"sections to compare the code by yourself.

Because HDMITX needs to detect the resolution supported by the screen as the target resolution, there is no timing of RGB outputted by SOC configured in DTS.

3.5.5 HDMI -> DSI Switch

Refer to DTS:

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-ddr4-v10.dtsi
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2dsi-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2dsi-dual-ddr4-v10.dts
```

The above DTS is based on the rk3568-evb.dtsi to add the function of RK628 to support HDMI to DSI, connecting to SOC directly, in which rk3568-evb-rk628-hdmi2dsi-ddr4-v10.dts outputs single-channel LVDS, and rk3568-evb-rk628-rgb2lvds-dual-ddr4-v10.dts outputs dual-channel LVDS, please refer to the "Introduction to the Input Module" and "Introduction to the Output Module"sections to compare the code by yourself.

For the RGB of outputting specified timing of the SOC configuration, please refer to the [HDMI Input] subsection of the [Input Module Configuration] chapter.

3.5.6 HDMI -> LVDS Switch

Refer to DTS:

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-ddr4-v10.dtsi
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2lvds-ddr4-v10.dts
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2lvds-dual-ddr4-v10.dts
```

The above DTS is based on the rk3568-evb.dtsi to add the function of RK628 to support HDMI to LVDS, connecting to SOC directly, in which rk3568-evb-rk628-hdmi2lvds-ddr4-v10.dts outputs single-channel LVDS, and rk3568-evb-rk628-hdmi2lvds-dual-ddr4-v10.dts outputs dual-channel LVDS, please refer to the "Introduction to the Input Module" and "Introduction to the Output Module"sections to compare the code by yourself.

For the RGB of outputting specified timing of the SOC configuration, please refer to the [HDMI Input] subsection of the [Input Module Configuration] chapter.

3.5.7 HDMI -> GVI Switch

Refer to DTS:

```
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-ddr4-v10.dtsi
arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2gvi-ddr4-v10.dts
```

The above DTS is based on the rk3568-evb.dtsi to add the function of RK628 to support HDMI to GVI, connecting to SOC directly, please refer to the "Introduction to the Input Module" and "Introduction to the Output Module"sections to compare the code by yourself.

For the RGB of outputting specified timing of the SOC configuration, please refer to the [HDMI Input] subsection of the [Input Module Configuration] chapter.

3.6 Basic Debug Command

3.6.1 Register debugging node

under /sys/kernel/debug/regmap node

```
rk3568_t:/ # ls /sys/kernel/debug/regmap/
0-001c              2-0050-dsi1   2-0050-hdmirx     4-0050-gpio0
0-0020              2-0050-efuse  4-0050-adapter    4-0050-gpio1
0-0020-rk817-codec  2-0050-gpio0  4-0050-combrxphy  4-0050-gpio2
2-0050-adapter      2-0050-gpio1  4-0050-combtxphy  4-0050-gpio3
2-0050-combrxphy    2-0050-gpio2  4-0050-cru        4-0050-grf
2-0050-combtxphy    2-0050-gpio3  4-0050-csi        4-0050-gvi
2-0050-cru          2-0050-grf    4-0050-dsi0       4-0050-hdmi
2-0050-csi          2-0050-gvi    4-0050-dsi1       4-0050-hdmirx
2-0050-dsi0         2-0050-hdmi   4-0050-efuse      ......
```

or under /sys/kernel/debug/rk628/x-xxxx/registers node

```
rk3568_t:/ # ls /sys/kernel/debug/rk628/2-0050/registers/
adapter  combrxphy  combtxphy  cru  csi  dsi0  dsi1  efuse  gpio0  gpio1  gpio2
gpio3  grf  gvi  hdmi  hdmirx
```

Thereinto, 2 represents the I2C bus number and 0050 represents the RK628 SLAVE address.

1. Read registers (taking GRF registers as an example)

under /sys/kernel/debug/regmap node

```
rk3568_t:/ # cat /sys/kernel/debug/regmap/2-0050-grf/registers
000: 06000088
004: ffffffff
008: 00000000
00c: 00000000
010: 00000001
014: 00000000
018: 00050000
01c: 002c0898
020: 00c00840
......

rk3568_t:/ # cat /sys/kernel/debug/regmap/2-0050-grf/registers | grep 200:
200: 20230321
```

or under /sys/kernel/debug/rk628/x-xxxx/registers node

```
rk3568_t:/ # cat /sys/kernel/debug/rk628/2-0050/registers/grf
rk628_grf:

0x0000: 06000298 ffffffff 00000000 00000001
0x0010: 00000001 00000000 000404b6 002804ba
0x0020: 0046047e 000a079e 00140794 0046047e
0x0030: 00140794 00000004 00000000 00000000
0x0040: 04380004 80000000 00000000 00000000
```

```
0x0050: 80000000 00000000 10000000 00000000
0x0060: 000002ea 01f0ea00 00000000 01f00000
0x0070: 0000155c 00005602 00000000 00000000
0x0080: 00006000 00000000 00000000 00000000
......

rk3568_t:/ # cat /sys/kernel/debug/rk628/2-0050/registers/grf | grep 0200:
0x0200: 20230321 00000000 00000000 00000000
```

2. Write registers (taking GRF registers as an example)

under /sys/kernel/debug/regmap node

```
rk3568_t:/ # echo 0x010 0x02000200 > /sys/kernel/debug/regmap/2-0050-
grf/registers
```

or under /sys/kernel/debug/rk628/x-xxxx/registers node

```
rk3568_t:/ # echo 0x010 0x02000200 > /sys/kernel/debug/rk628/2-0050/registers/grf
```

3.6.2 Self-test mode command

When the output cannot be displayed properly, you can use the following command to determine whether the problem is with the RK628 input or RK628 output (only applies for RK628F /H):

```
# Enable horizontal color bar
rk3568_t:/ # echo 1 > /sys/kernel/debug/rk628/2-0050/scaler_color_bar

# Enable vertical color bar
rk3568_t:/ # echo 2 > /sys/kernel/debug/rk628/2-0050/scaler_color_bar

# Disable color bar
rk3568_t:/ # echo 0 > /sys/kernel/debug/rk628/2-0050/scaler_color_bar
```

If the scaler colorbar is displayed normally, check the configuration of the main control output, RK628 input, and RK628 Process; if it cannot be displayed normally, check the configuration of the RK628 output first, and determine whether the controller, corresponding phy, and screen end links of the output module are working properly through the colorbar of each of the following interfaces. After the following module colorbar works normally, the scaler color still cannot be displayed, continue to check the configuration of RK628 input.

The following command tests whether the controller, corresponding phy, and screen-end links of the output module are working properly:

1. HDMITX color bar

```
# Enable normal color bar
rk3568_t:/ # echo 1 > /sys/kernel/debug/rk628/2-0050/hdmitx_color_bar

# Enable special color bar
rk3568_t:/ # echo 2 > /sys/kernel/debug/rk628/2-0050/hdmitx_color_bar

# Enable black color bar
rk3568_t:/ # echo 3 > /sys/kernel/debug/rk628/2-0050/hdmitx_color_bar

# Disable color bar
rk3568_t:/ # echo 0 > /sys/kernel/debug/rk628/2-0050/hdmitx_color_bar
```

2. DSI color bar

```
# Enable dsi color bar
rk3568_t:/ # echo 1 > /sys/kernel/debug/rk628/2-0050/dsi_color_bar

# Disable dsi color bar
rk3568_t:/ # echo 0 > /sys/kernel/debug/rk628/2-0050/dsi_color_bar
```

3. GVI color bar

```
# Enable color bar
rk3568_t:/ # echo 1 > /sys/kernel/debug/rk628/2-0050/gvi_color_bar

# Disable color bar
rk3568_t:/ # echo 0 > /sys/kernel/debug/rk628/2-0050/gvi_color_bar
```

3.6.3 RGB IN debugging command (only applies for RK628F/H)

In the RK628F/H application, you can run the following command to check the actual resolution received by RGB IN and determine whether the RGB RX is normal.

```
rk3568_t:/ # cat /sys/kernel/debug/rk628/2-0050/rgb_resolution
1080x1920 pclk:131967000
```

3.7 Common Display Questions and Resolution

3.7.1 Fail to generate the regmap node

As the log followed, check whether there are any errors related to RK628 that make RK628 to fail to probe.

```
rk3568_t:/ # dmesg | grep rk628
[    0.294331] rk628 2-0050: the driver version is 0.1.0 of RK628-F/B/G
[    0.294426] rk628 2-0050: failed to request enable GPIO: -16
```

3.7.2 I2C Communication Exception

The log followed means that the I2C communication exception of RK628 leads to register failure of various modules, you need check the power sequence of RK628 and reference clock of 24MHz (refer to 'Common Display Questions and Resolution' section), and the iomux of related pin.

```
[    0.960609] rk628 1-0050: failed to access register: -6
```

3.7.3 DSI or GVI Display with blur

You can try to modify the positive and negative polarity of src-timing blanking in dts, and the method is valid for both DSI and GVI.

For example:

```
--- a/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
@@ -347,8 +347,8 @@ src-timing {
                                vfront-porch = <15>;
                                hsync-len = <2>;
                                vsync-len = <2>;
-                               hsync-active = <0>;
-                               vsync-active = <0>;
+                               hsync-active = <1>;
+                               vsync-active = <1>;
                                de-active = <0>;
                                pixelclk-active = <0>;
```

3.7.4 DSI can't display after rising the clock-frequency

In order to improve the refresh rate, it's common to rise the clock-frequency, for excample:

```
--- a/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
@@ -338,7 +338,7 @@ rk628-panel {

                display-timings {
                        src-timing {
-                               clock-frequency = <132000000>;
+                               clock-frequency = <148500000>;
                                hactive = <1080>;
                                vactive = <1920>;
                                hback-porch = <30>;
@@ -354,7 +354,7 @@ src-timing {
                        };

                        dst-timing {
-                               clock-frequency = <132000000>;
+                               clock-frequency = <148500000>;
                                hactive = <1080>;
                                vactive = <1920>;
```

```
                                hback-porch = <30>;
```

But rising CLK may cause flash or display failure, in this case, you can try to adjust the rate of LANE as followed to solve the problem.

```
--- a/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-rgb2dsi-ddr4-v10.dts
@@ -57,6 +57,7 @@ &i2c2_rk628 {

                rk628,rgb-in;
                rk628-dsi {
+                       rockchip,lane-mbps = <1000>;
                        //rockchip,dual-channel;
                        dsi,eotp;
                        dsi,video-mode;
```

3.7.5 Display offset

For example, HDMI is used for input, if the back-end diaplay offsets, according to the offset direction, you should adjust the timing of screen to blank the offset.(line blanking: hfront-porch/hback-porch/hsync-len and column blanking: vfront-porch/vback-porch/vsync-len in src-timing/dst-timing) At the same time, make sure that the clock-frequency of the back-end output and the HDMI input are the same, for example, they are both 148500000.

```
--- a/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2dsi-ddr4-v10.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3568-evb-rk628-hdmi2dsi-ddr4-v10.dts
@@ -314,7 +314,7 @@ src-timing {
                        hfront-porch = <60>;
                        vback-porch = <10>;
                        vfront-porch = <10>;
-                       hsync-len = <20>;
+                       hsync-len = <40>;
                        vsync-len = <10>;
                        hsync-active = <1>;
                        vsync-active = <1>;
```

3.7.6 The operation of RK628 GPIO

The GPIO interface can be directly called at the location where the GPIO needs to be controlled(the header files rk628_pinctrl.h and rk628_gpio.h must be included when calling), which includes IOMUX settings, GPIO direction settings, and level settings. For example, the following operation sets the I2S_D2_M0 interfaces as GPIO and output high level.

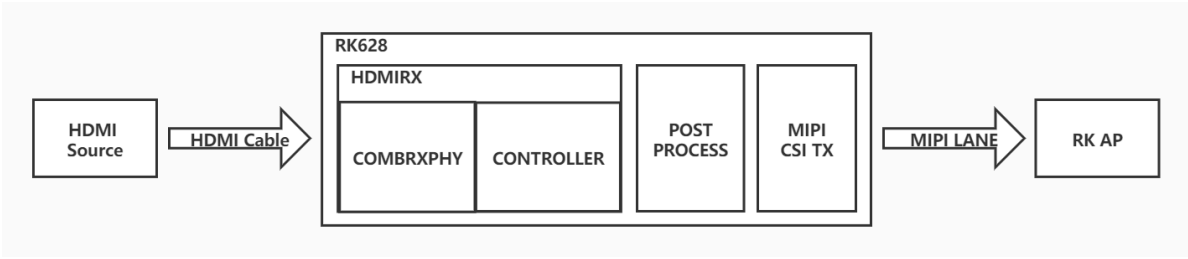
```
#include "rk628_pinctrl.h"
#include "rk628_gpio.h"

rk628_misc_gpio_direction_output(rk628, GPIO0_A6, 1);
```

If you want to recover, the following interfaces need to be used.

```
rk628_misc_pinctrl_set_mux(rk628, GPIO0_A6, I2SM0D2);
```

4. Media

4.1 Driver Introduction



Media is the driver code for RK628 HDMI IN channel, you can use the RK628 as a camera-like device to realize the function above.

The driver code of RK628 under the media framework needs to be differentiated according to the kernel version. Versions prior to kernel-5.10 use the rk628_for-all patch package, and kernel-5.10 versions use the kernel_5.10_rk628_media_patch patch package.

Driver directory structure of versions prior to kernel-5.10:

```
drivers/media/i2c/rk628
├── Makefile
├── rk628_bt1120_v4l2.c
├── rk628.c
├── rk628_combrxphy.c
├── rk628_combrxphy.h
├── rk628_combtxphy.c
├── rk628_combtxphy.h
├── rk628_cru.c
├── rk628_cru.h
├── rk628_csi.c
├── rk628_csi.h
├── rk628_csi_v4l2.c
├── rk628_dsi.c
├── rk628_dsi.h
├── rk628_gpio.h
├── rk628_grf.h
├── rk628.h
├── rk628_hdmirx.c
├── rk628_hdmirx.h
├── rk628_mipi_dphy.c
├── rk628_mipi_dphy.h
├── rk628_pinctrl.c
├── rk628_pinctrl.h
├── rk628_post_process.c
├── rk628_post_process.h
└── rk628_v4l2_controls.h
```

Driver directory structure of kernel-5.10:

```
drivers/media/i2c/rk628
├── Makefile
├── rk628_bt1120_v4l2.c
├── rk628.c
├── rk628_combrxphy.c
├── rk628_combrxphy.h
├── rk628_combtxphy.c
├── rk628_combtxphy.h
├── rk628_cru.c
├── rk628_cru.h
├── rk628_csi.c
├── rk628_csi.h
├── rk628_csi_v4l2.c
├── rk628_dsi.c
├── rk628_dsi.h
├── rk628.h
├── rk628_hdmirx.c
├── rk628_hdmirx.h
├── rk628_mipi_dphy.c
├── rk628_mipi_dphy.h
├── rk628_post_process.c
└── rk628_post_process.h
```

At present, the main driver files and their applicable scenarios are as follows:

rk628_csi.c: HDMI2CSI application scenario, for CameraHal1 and third-party SOCs;

rk628_csi_v4l2.c: HDMI2CSI / HDMI2DSI application scenario, for CameraHal3/TV input and other platforms based on v4l2 framework;

rk628_bt1120_v4l2.c: HDMI2BT1120 application scenario, for CameraHal3 and other platforms based on v4l2 framework.

4.2 RK628F/H improvements compared with RK628D

The media framework driver is mainly used in the scenario where HDMI2CSI, HDMI2DSI and HDMI2BT1120 are connected to the SOC master control platform.Compared with RK628D, the main improvements of RK628F/H are as followed:

Module	Main improvements
HDMIRX	Support HDMI2.0 Support RGB 4K60 input Add supports for more frequency points Support DVI mode Locking is faster and more stable
MIPI-CSI	It solves the color shift problem caused by formats conversion sampling of different image formats Add one MIPI-CSI and dual MIPI-CSI supporting 4K60 output
CSC	Matrix is added to make the image color gamut switch more flexibly

4.3 Transplant Description

Versions prior to kernel-5.10 use the rk628_for-all patch package, and kernel-5.10 versions use the kernel_5.10_rk628_media_patch patch package. RK628 drivers which are Media related are copied directly to drivers/media/i2c/rk628/, then modify drivers/Media/i2c/Makefile and add the compiling of rk628:

```
obj-y       += rk628/
```

All Media drivers are compiled by default, you just need to configure dts and run the corresponding driver.

4.4 HDMI IN VIDEO Framework Description

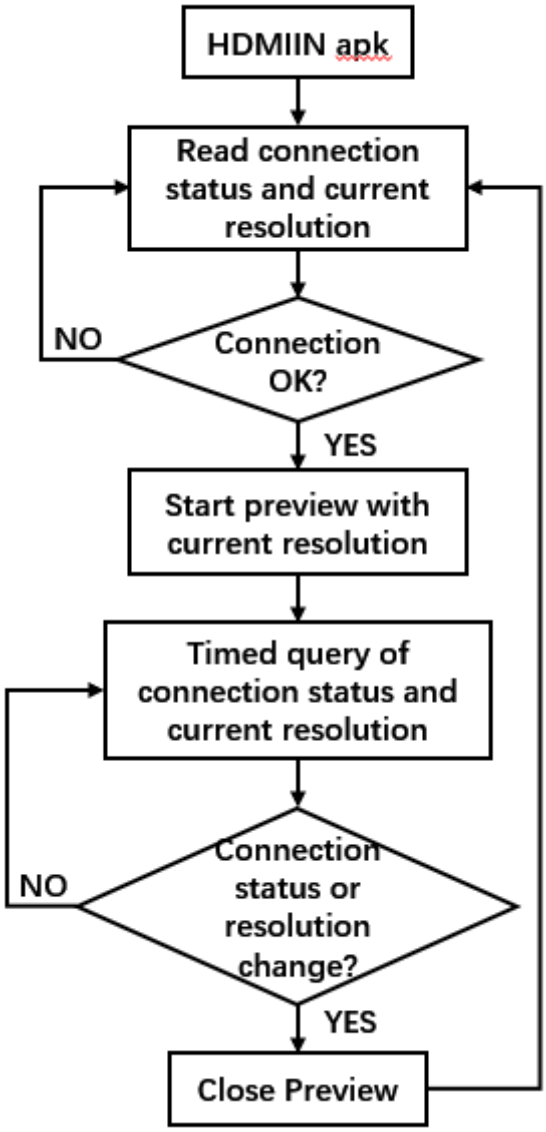
Parts of software implementation schemes of HDMI IN video are to simulate the RK628 into a MIPI SOC camera device, receive video datas by camera framework and display them in the APK. At the same time, add HDMI IN hot-plug and HDMI IN resolution adaptive support based on the application scenarios of HDMI IN.

Develop Android9/10/11 based on cameraHAL3 framework, please refer to the file "Rockchip_Developer_Guide_HDMI_IN_Based_On_CameraHal3_CN" under SDK directory RKDocs/common/camera/HAL3.

Develop Android12/13 based on cameraHAL3/TVinput framework, please refer to the file "Rockchip_Developer_Guide_Android12+_HDMI_IN_Bridge_CN" under SDK directory RKDocs/common/camera/HAL3.

4.4.1 HDMI IN APK Process

APK process:



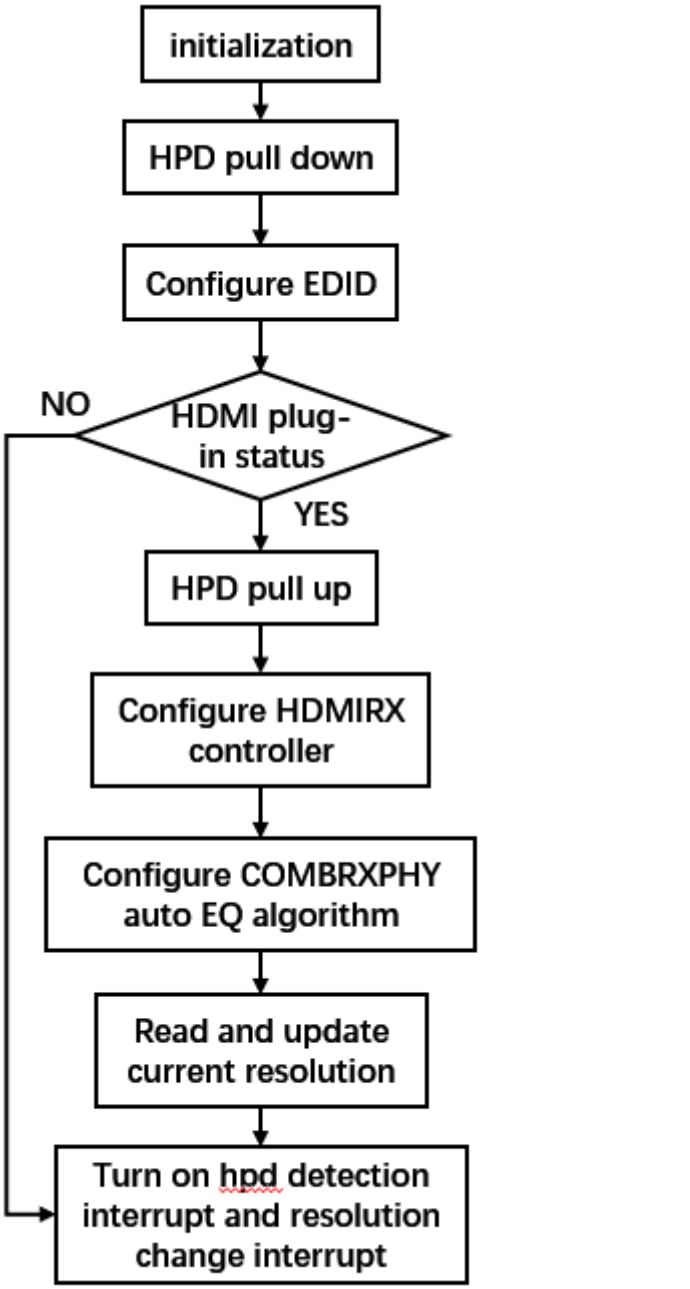
Based on kernel-4.19 or kernel-4.4 (Android 9/10/11), you can directly use rkCamera2 to open the preview.

Based on kernel-5.10 version (Android 12/13), you need to distinguish whether it is a camera framework or a TV framework, please refer to the documentation for details:

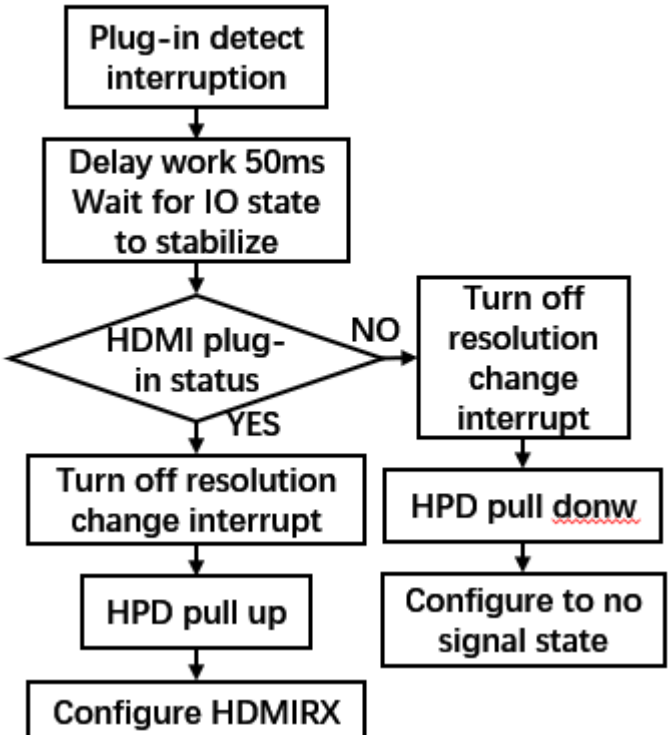
"Rockchip_Developer_Guide_Android12+_HDMI_IN_Bridge_CN"

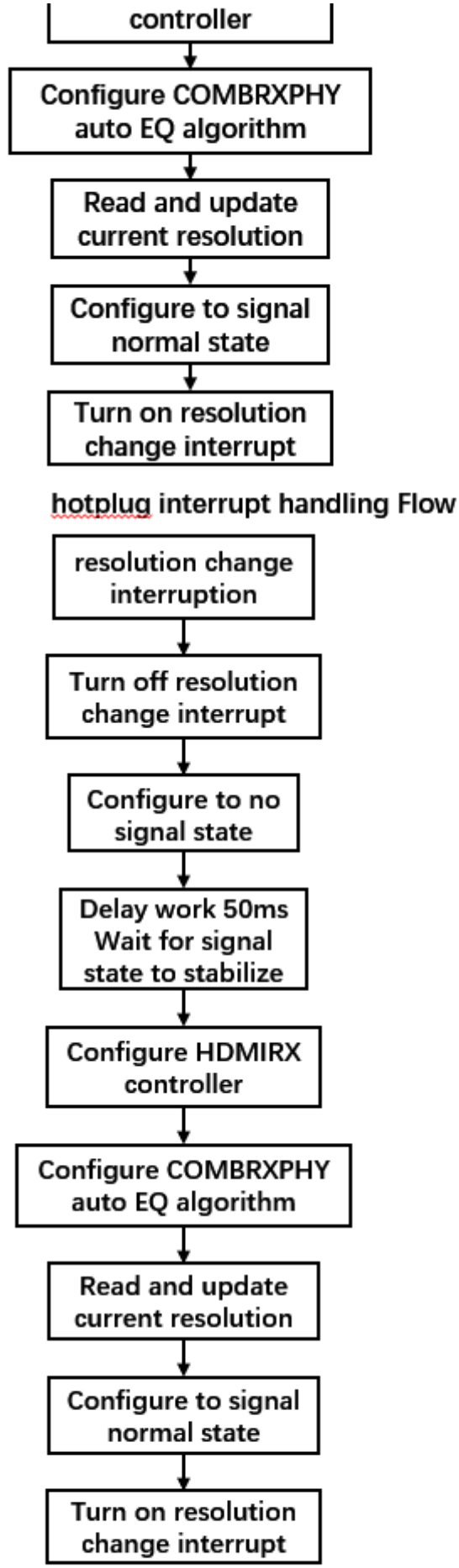
4.4.2 RK628 Driver Framework

There are 3 parts should be paid attention to for RK628 driver, initialization, hotplug interrupt handling, resolution switching interrupt handling. The flow chart is as followed:



Driver initialization Flow




controller
Configure COMBRXPHY auto EQ algorithm
Read and update current resolution
Configure to signal normal state
Turn on resolution change interrupt
hotplug interrupt handling Flow
resolution change interruption
Turn off resolution change interrupt
Configure to no signal state
Delay work 50ms Wait for signal state to stabilize
Configure HDMIRX controller
Configure COMBRXPHY auto EQ algorithm
Read and update current resolution
Configure to signal normal state
Turn on resolution change interrupt
resolution switching interrupt handling Flow

4.5 dts Configuration Description

4.5.1 RK628 Node Configuration

```
&i2c5 {
    status = "okay";

    rk628_csi: rk628_csi@50 {
        reg = <0x50>;
        compatible = "rockchip,rk628-csi-v4l2";
        status = "okay";
        power-domains = <&power RK3588_PD_VI>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_pin>;
        interrupt-parent = <&gpio2>;
        interrupts = <RK_PC4 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio1 RK_PA0 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio4 RK_PC6 GPIO_ACTIVE_HIGH>;
        plugin-det-gpios = <&gpio1 RK_PA1 GPIO_ACTIVE_LOW>;
        continues-clk = <1>;

        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "HDMI-MIPI";
        rockchip,camera-module-lens-name = "RK628-CSI";

        multi-dev-info {
            dev-idx-l = <0>;
            dev-idx-r = <1>;
            combine-idx = <0>;
            pixel-offset = <0>;
            dev-num = <2>;
        };

        port {
            hdmiin_out0: endpoint {
                remote-endpoint = <&hdmi_mipi0_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};
```

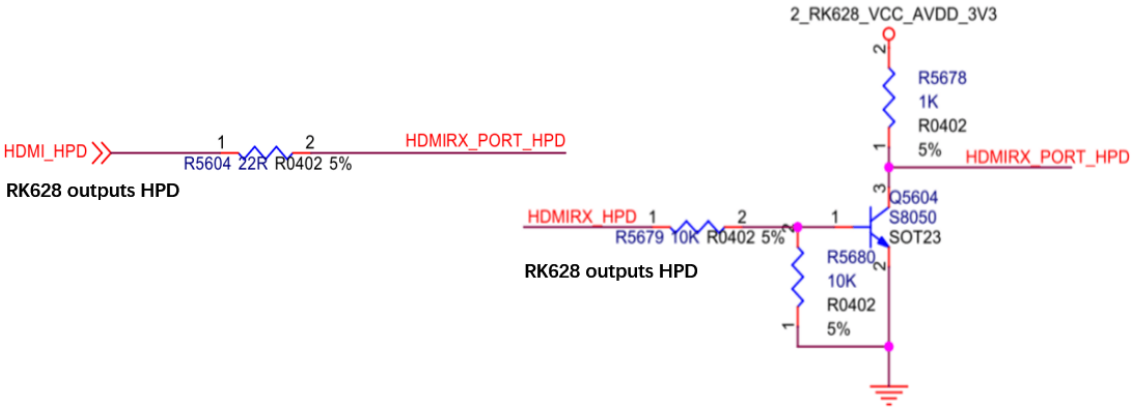
- **reg**: I2C Address: The typical 7bit I2C address of the RK628 is 0x50. When using multiple RK628s, the I2C address can be changed through the GPIO of the RK628, you can refer to 'Communication of SOC and RK628 I2C 'section.
- **compatible**:

  compatible = "rockchip,rk628-csi" corresponds rk628_csi.c;
  compatible = "rockchip,rk628-csi-v4l2" corresponds rk628_csi_v4l2.c;
  compatible = "rockchip,rk628-bt1120-v4l2" corresponds rk628_bt1120_v4l2.c;

  interrupt-parent/ interrupts: Connect the RK628 interrupt GPIO pin;

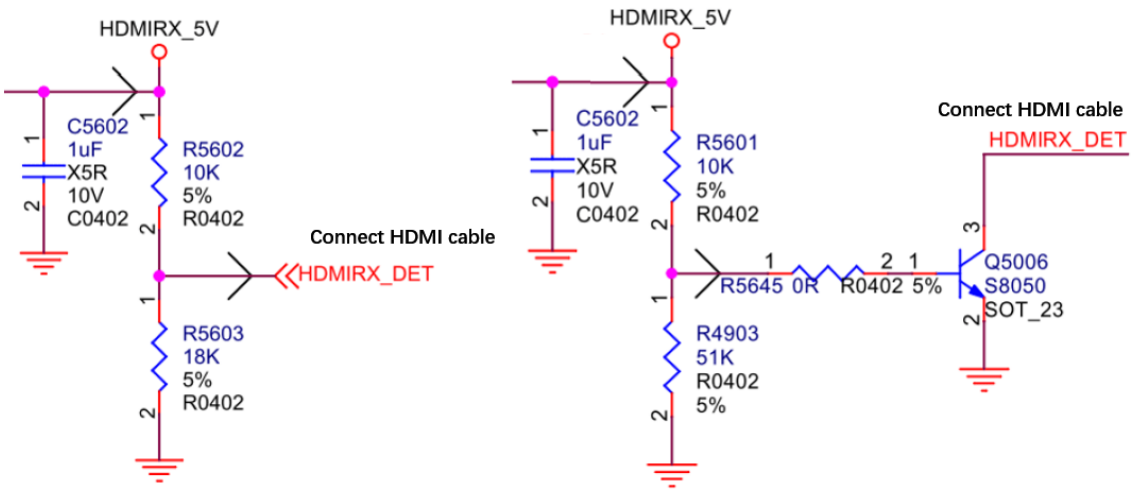
enable-gpios: RK628 power supply control GPIO pin (can be disabled for regular power supply);
reset-gpios: RK628 reset control GPIO pin;
hpd-output-inverted: HPD output negation configuration, if the HPD output level is negated on the circuit, you need to enable this configuration item;

- ◆ HPD not inverted design: ◆ HPD inverted design:



- **plugin-det-gpios**: HDMI inserted detection GPIO pin. Pay attention to whether there is a negative level on the circuit. The effective level needs to be configured correctly.

◆ HDMIRX_DET not inverted design: ◆ HDMIRX_DET inverted design:



power-gpios: MIPI RX power domain of RK master power control GPIO pin (can be disabled for regular power supply);

hdcp-enable: Enable HDCP function;

- **continues-clk**: Select MIPI CSI to output a CLK in continuous mode or non-continuous mode;
- **multi-dev-info**: This configuration is required in the dual-MIPI 4K60 scenario (only supported by RK628F/H), and currently only supported by RK3588/RK3562 SOC.

4.5.2 Image Receiving Link Combination

The RK628 conversion chip is developed as a camera-like device, which needs to be implemented a driver based on V4L2 framework like the camera sensor, and the method of data link configuration is consistent with the MIPI SOC Sensor. The followings are common link combinations.

- RK628 -> HDMI2CSI -> SOC
- RK628 -> HDMI2DSI -> SOC
- RK628 -> HDMI2BT1120 -> SOC

4.5.3 HDMI2CSI Link

4.5.3.1 RK628D

Take RK3288 + CSI_V4L2 for example, the link configuration of rk628 + isp1 is:

```
&rk628_csi {
    status = "okay";
    /*
     * If the hpd output level is inverted on the circuit,
     * the following configuration needs to be enabled.
     */
    /* hpd-output-inverted; */
    plugin-det-gpios = <&gpio0 13 GPIO_ACTIVE_HIGH>;
    power-gpios = <&gpio0 17 GPIO_ACTIVE_HIGH>;
    rockchip,camera-module-index = <0>;
    rockchip,camera-module-facing = "back";
    rockchip,camera-module-name = "RK628-CSI";
    rockchip,camera-module-lens-name = "NC";

    port {
        hdmiin_out0: endpoint {
            remote-endpoint = <&mipi_in>;
            data-lanes = <1 2 3 4>;
        };
    };
};

&mipi_phy_rx0 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi_in: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&hdmiin_out0>;
                data-lanes = <1 2 3 4>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            dphy_rx_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&isp_mipi_in>;
            };
```

```
        };
    };
};

&rkisp1 {
    status = "okay";
    port {
        #address-cells = <1>;
        #size-cells = <0>;

        isp_mipi_in: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&dphy_rx_out>;
        };
    };
};

&isp_mmu {
    status = "okay";
};
```

Take RK3568 + CSI_V4L2 for example, the link configuration of rk628 + isp2 is:

```
&i2c2 {
    status = "okay";
    pinctrl-names = "default";
    pinctrl-0 = <&i2c2m1_xfer>;

    rk628_csi: rk628_csi@32 {
        compatible = "rockchip,rk628-csi-v4l2";
        reg = <0x50>;
        interrupt-parent = <&gpio4>;
        interrupts = <16 IRQ_TYPE_LEVEL_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_irq>;
        reset-gpios = <&gpio4 RK_PD2 GPIO_ACTIVE_LOW>;
        plugin-det-gpios = <&gpio0 RK_PD6 GPIO_ACTIVE_LOW>;
        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "RK628-CSI";
        rockchip,camera-module-lens-name = "NC";
        port {
            rk628_out: endpoint {
                remote-endpoint = <&mipi_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&csi2_dphy_hw {
    status = "okay";
};

&csi2_dphy0 {
    status = "okay";
```

```
    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi_in: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&rk628_out>;
                data-lanes = <1 2 3 4>;
            };
        };
        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            csidphy0_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&isp0_in>;
            };
        };
    };
};

&rkisp {
    status = "okay";
};

&rkisp_mmu {
    status = "okay";
};

&rkisp_vir0 {
    status = "okay";

    port {
        #address-cells = <1>;
        #size-cells = <0>;

        isp0_in: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&csidphy0_out>;
        };
    };
};
```

RK3568, rk628 + vicap link configuration:

```
&i2c2 {
    status = "okay";
    pinctrl-names = "default";
    pinctrl-0 = <&i2c2m1_xfer>;

    rk628_csi: rk628_csi@32 {
```

```
        compatible = "rockchip,rk628-csi-v4l2";
        reg = <0x50>;
        //clocks = <&ext_cam_clk>;
        //clock-names = "xvclk";

        interrupt-parent = <&gpio4>;
        interrupts = <16 IRQ_TYPE_LEVEL_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_irq>;
        //power-gpios = <&gpio0 RK_PD5 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio4 RK_PD2 GPIO_ACTIVE_LOW>;
        plugin-det-gpios = <&gpio0 RK_PD6 GPIO_ACTIVE_LOW>;
        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "RK628-CSI";
        rockchip,camera-module-lens-name = "NC";
        port {
            rk628_out: endpoint {
                remote-endpoint = <&mipi_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&csi2_dphy_hw {
    status = "okay";
};

&csi2_dphy0 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi_in: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&rk628_out>;
                data-lanes = <1 2 3 4>;
            };
        };
        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            csidphy0_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&mipi_csi2_input>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
```

```
};

&mipi_csi2 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi_csi2_input: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&csidphy0_out>;
                data-lanes = <1 2 3 4>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi_csi2_output: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&cif_mipi_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&rkcif {
    status = "okay";
};

&rkcif_mipi_lvds {
    status = "okay";

    port {
        cif_mipi_in: endpoint {
            remote-endpoint = <&mipi_csi2_output>;
            data-lanes = <1 2 3 4>;
        };
    };
};

&rkcif_mmu {
    status = "okay";
};
```

4.5.3.2 RK628F/H

The configuration of the RK628F/H docking SOC DTS is basically the same as that of the RK628D, which can be referred to RK628D.

RK628F/H supports dual MIPI configuration, currently only supports RK3588/RK3562 main controller, the following takes RK3588 dual MIPI as an example, rk628F/H + vicap link configuration, RK628F/H CSI0/CSI1 is connected to RK3588 CSI0/CSI1 respectively.

```
&csi2_dphy0 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            hdmi_mipi2_in: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&hdmiin_out1>;
                data-lanes = <1 2 3 4>;
            };
        };
        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            csidphy0_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&mipi2_csi2_input>;
            };
        };
    };
};

&csi2_dphy0_hw {
    status = "okay";
};

&csi2_dphy1_hw {
    status = "okay";
};

&i2c3 {
    status = "okay";
    clock-frequency = <400000>;

    rk628_csi: rk628_csi@50 {
        reg = <0x50>;
        compatible = "rockchip,rk628-csi-v4l2";
        status = "okay";
        power-domains = <&power RK3588_PD_VI>;
```

```
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_pin>;
        interrupt-parent = <&gpio1>;
        interrupts = <RK_PB2 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio1 RK_PA7 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio1 RK_PB1 GPIO_ACTIVE_HIGH>;
        plugin-det-gpios = <&gpio2 RK_PB6 GPIO_ACTIVE_LOW>;
        continues-clk = <1>;

        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "HDMI-MIPI2";
        rockchip,camera-module-lens-name = "RK628-CSI";

        multi-dev-info {
            dev-idx-l = <2>;
            dev-idx-r = <4>;
            combine-idx = <2>;
            pixel-offset = <0>;
            dev-num = <2>;
        };

        port {
            hdmiin_out1: endpoint {
                remote-endpoint = <&hdmi_mipi2_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&mipi_dcphy0 {
    status = "okay";
};

&mipi_dcphy1 {
    status = "okay";
};

&mipi2_csi2 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi2_csi2_input: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&csidphy0_out>;
            };
        };

        port@1 {
```

```
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi2_csi2_output: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&cif_mipi_in2>;
            };
        };
    };
};

&rkcif {
    status = "okay";
};

&rkcif_mipi_lvds2 {
    status = "okay";

    port {
        cif_mipi_in2: endpoint {
            remote-endpoint = <&mipi2_csi2_output>;
        };
    };
};

&rkcif_mmu {
    status = "okay";
};
```

4.5.4 HDMI2DSI Switching

4.5.4.1 RK628D

For HDMI To MIPI CSI application scenarios, RK628D has color gamut space switching or rising and falling sampling of the image format, and a small amount of details in the image color may have a slight deviation. HDMI To MIPI DSI can replace HDMI To MIPI CSI scheme, the master receive RGB888 image directly, and the application layer uses the rgb2yuv algorithm library to convert its format, it is necessary to note that:

The maximum resolution supported is 1080P60;

The AP supporting to receive MIPI DSI currently: RK1109, RK1126, RK3566, RK3568, RK3588, RK3562;

The AP not supporting to receive MIPI DSI currently: RK3288, RK3326, RK3368, RK3399 and other older chips.

CSI is compatible with DSI codes, only dts compatible should be distinguished:

CSI is compatible = "rockchip,rk628-csi-v4l2";

DSI is compatible = "rockchip,rk628-dsi-v4l2";

Take RK3568 for example

```
&i2c2 {
    status = "okay";
```

```
    pinctrl-names = "default";
    pinctrl-0 = <&i2c2m1_xfer>;

    rk628_csi: rk628_csi@32 {
        compatible = "rockchip,rk628-dsi-v4l2";
        reg = <0x50>;

        interrupt-parent = <&gpio4>;
        interrupts = <16 IRQ_TYPE_LEVEL_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_irq>;
        //power-gpios = <&gpio0 RK_PD5 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio4 RK_PD2 GPIO_ACTIVE_LOW>;
        plugin-det-gpios = <&gpio0 RK_PD6 GPIO_ACTIVE_LOW>;
        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "RK628-CSI";
        rockchip,camera-module-lens-name = "NC";
        port {
            rk628_out: endpoint {
                remote-endpoint = <&hdmi2mipi_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&mipi_csi2 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi_csi2_input: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&csidphy_out>;
                data-lanes = <1 2 3 4>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            mipi_csi2_output: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&cif_mipi_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
```

```
};

&rkcif {
    status = "okay";
};

&rkcif_mipi_lvds {
    status = "okay";

    port {
        cif_mipi_in: endpoint {
            remote-endpoint = <&mipi_csi2_output>;
            data-lanes = <1 2 3 4>;
        };
    };
};

&rkcif_mmu {
    status = "okay";
};

&csi2_dphy_hw {
    status = "okay";
};

&csi2_dphy0 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            hdmi2mipi_in: endpoint@0 {
                reg = <1>;
                remote-endpoint = <&rk628_out>;
                data-lanes = <1 2 3 4>;
            };
        };
        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            csidphy_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&mipi_csi2_input>;
            };
        };
    };
};
```

If the master is RK3568 and the code version is old, the CIF needs to include the following commits to support RGB888 reception:

The RK3588 kernel-5.10 driver code supports RGB888 by default.

```
From e925e385a41a857e8e50020b5df9a2d41b166d83 Mon Sep 17 00:00:00 2001
From: Zefa Chen <zefa.chen@rock-chips.com>
Date: Thu, 02 Sep 2021 15:07:13 +0800
Subject: [PATCH] media: rockchip: cif fix errors in rgb24 data format

Signed-off-by: Zefa Chen <zefa.chen@rock-chips.com>
Change-Id: I0766997860f06dc25e604c2ea8425049a987fdc5
---

diff --git a/drivers/media/platform/rockchip/cif/capture.c
b/drivers/media/platform/rockchip/cif/capture.c
index 6c0aa7d..36bd2c9 100644
--- a/drivers/media/platform/rockchip/cif/capture.c
+++ b/drivers/media/platform/rockchip/cif/capture.c
@@ -1785,13 +1785,11 @@
      * needs aligned with :ALIGN(bits_per_pixel * width * 2, 8), to optimize
reading and
      * writing of ddr, aliged with 256
      */
-    if (fmt->fmt_type == CIF_FMT_TYPE_RAW && stream->is_compact) {
+    if (fmt->fmt_type == CIF_FMT_TYPE_RAW && stream->is_compact &&
+        fmt->csi_fmt_val != CSI_WRDDR_TYPE_RGB888) {
         channel->virtual_width = ALIGN(channel->width * fmt->raw_bpp / 8, 256);
     } else {
-        if (fmt->fmt_type == CIF_FMT_TYPE_RAW && fmt->csi_fmt_val !=
CSI_WRDDR_TYPE_RAW8)
-            channel->virtual_width = ALIGN(channel->width * 2, 8);
-        else
-            channel->virtual_width = ALIGN(channel->width * fmt->bpp[0] / 8, 8);
+        channel->virtual_width = ALIGN(channel->width * fmt->bpp[0] / 8, 8);
     }

     if (channel->fmt_val == CSI_WRDDR_TYPE_RGB888)
@@ -3240,7 +3238,8 @@

         if (fmt->fmt_type == CIF_FMT_TYPE_RAW && stream->is_compact &&
             (dev->active_sensor->mbus.type == V4L2_MBUS_CSI2 ||
-             dev->active_sensor->mbus.type == V4L2_MBUS_CCP2)) {
+             dev->active_sensor->mbus.type == V4L2_MBUS_CCP2) &&
+             fmt->csi_fmt_val != CSI_WRDDR_TYPE_RGB888) {
             bpl = ALIGN(width * fmt->raw_bpp / 8, 256);
         } else {
             bpp = rkcif_align_bits_per_pixel(stream, fmt, i);
@@ -4659,13 +4658,11 @@
      * needs aligned with :ALIGN(bits_per_pixel * width * 2, 8), to optimize
reading and
      * writing of ddr, aliged with 256
      */
-    if (fmt->fmt_type == CIF_FMT_TYPE_RAW && stream->is_compact) {
+    if (fmt->fmt_type == CIF_FMT_TYPE_RAW && stream->is_compact &&
+        fmt->csi_fmt_val != CSI_WRDDR_TYPE_RGB888) {
         *crop_vwidth = ALIGN(raw_width * fmt->raw_bpp / 8, 256);
     } else {
-        if (fmt->fmt_type == CIF_FMT_TYPE_RAW)
-            *crop_vwidth = ALIGN(raw_width * 2, 8);
```

```
-        else
-            *crop_vwidth = ALIGN(raw_width * fmt->bpp[0] / 8, 8);
+        *crop_vwidth = ALIGN(raw_width * fmt->bpp[0] / 8, 8);
     }

     if (channel->fmt_val == CSI_WRDDR_TYPE_RGB888)
```

rk628-dsi outputs command mode by default, cif receiving also need use command mode, the modification is as follows:

```
diff --git a/drivers/media/platform/rockchip/cif/capture.c
b/drivers/media/platform/rockchip/cif/capture.c
index 36bd2c99c707..fc4181daed2c 100644
--- a/drivers/media/platform/rockchip/cif/capture.c
+++ b/drivers/media/platform/rockchip/cif/capture.c
@@ -1753,7 +1753,7 @@ static int rkcif_csi_channel_init(struct rkcif_stream
*stream,

        channel->fmt_val = stream->cif_fmt_out->csi_fmt_val;

-       channel->cmd_mode_en = 0; /* default use DSI Video Mode */
+       channel->cmd_mode_en = 1; /* default use DSI Video Mode */

        if (stream->crop_enable) {
                channel->crop_en = 1;
```

Connect to camera framework

If the main controller is RK356X series, it can only be connected to the camera framework, but the camera framework does not support RGB to display directly, so you need to add the rgb2yuv algorithm library.

CameraHal needs to add rgb2yuv, if the code version of SDK is Android11 R9, then you can add the patches followed directly. Refer to rk356x_HAL3_support_rgb2yuv_patch.rar for the details.

If the code version is R10 or R11, add the modification followed in hardware/rockchip/camera, and then add the patches above.

```
diff --git a/psl/rkisp2/RKISP2GraphConfig.cpp b/psl/rkisp2/RKISP2GraphConfig.cpp
index 2a5fa5a..3d2409c 100755
--- a/psl/rkisp2/RKISP2GraphConfig.cpp
+++ b/psl/rkisp2/RKISP2GraphConfig.cpp
@@ -76,6 +76,7 @@ const string MEDIACTL_POSTVIEWNAME = "postview";

 const string MEDIACTL_STATNAME = "rkisp1-statistics";
 const string MEDIACTL_VIDEONAME_CIF = "stream_cif_dvp_id0";
+const string MEDIACTL_VIDEONAME_CIF_MIPI_ID0 = "stream_cif_mipi_id0";

 RKISP2GraphConfig::RKISP2GraphConfig() :
        mManager(nullptr),
@@ -2620,6 +2621,13 @@ status_t RKISP2GraphConfig::getImguMediaCtlConfig(int32_t
cameraId,
                addLinkParams("rkisp-isp-subdev", 2, "rkisp_mainpath", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
                addLinkParams("rkisp-isp-subdev", 2, "rkisp_selfpath", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
            }
+       } else if(mipName2.find("mipi") != std::string::npos) {
```

```
+            addLinkParams(mipName, mipSrcPad, mipName2, csiSinkPad, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            addLinkParams(mipName2, 1, "stream_cif_mipi_id0", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            addLinkParams(mipName2, 2, "stream_cif_mipi_id1", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            addLinkParams(mipName2, 3, "stream_cif_mipi_id2", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            addLinkParams(mipName2, 4, "stream_cif_mipi_id3", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            mSensorLinkedToCIF = true;
         } else {
             addLinkParams(mipName, mipSrcPad, csiName, csiSinkPad, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
             addLinkParams(csiName, csiSrcPad, IspName, ispSinkPad, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
@@ -2628,6 +2636,12 @@ status_t RKISP2GraphConfig::getImguMediaCtlConfig(int32_t
cameraId,
             addLinkParams(csiName, 5, "rkisp_rawwr3", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
         }
     }
+    if(mSensorLinkedToCIF){
+        addImguVideoNode(IMGU_NODE_VIDEO, MEDIACTL_VIDEONAME_CIF_MIPI_ID0,
mediaCtlConfig);
+        addFormatParams(MEDIACTL_VIDEONAME_CIF_MIPI_ID0, mCurSensorFormat.width,
mCurSensorFormat.height,
+                0, V4L2_PIX_FMT_NV12, 0, 0, mediaCtlConfig);
+        return OK;
+    }
     // isp input pad format and selection config
     addFormatParams(IspName, ispInWidth, ispInHeight, ispSinkPad, ispInFormat,
0, 0, mediaCtlConfig);
     addSelectionParams(IspName, ispInWidth, ispInHeight, 0, 0,
V4L2_SEL_TGT_CROP, ispSinkPad, mediaCtlConfig);
```

NOTE: these patches can only be used for RK3568+ANDROID11 platform.

Docking TV Framework

If the HDMI to DSI channel is used with RK3588, the application framework can use the TV framework, and it can support RGB888 format to send the display directly, and there is no need to add the above rgb2yuv algorithm library, which is described in the APK adaptation method section.

4.5.4.2 RK628F/H

RK628F/H solves the color shift problem caused by the sampling of format conversion when RK628D is input with different image formats.

However, in some scenarios with high requirements for image quality, RK628F/H can output RGB888 image format through DSI interface. After RK3588 and other main controllers receive RGB888 format images, send them directly to display through the TV frame without color conversion, ensuring that the image quality is not lost. The RK628F/H has two MIPI-DSIs, which also support dual MIPI mode to connect with the RK3588 master.

Take RK3588 + RK628F/H dual MIPI as an example

All you need to do is to change compatible to DSI.

```
......
&i2c3 {
    status = "okay";
    clock-frequency = <400000>;

    rk628_csi: rk628_csi@50 {
        reg = <0x50>;
        compatible = "rockchip,rk628-dsi-v4l2";
        status = "okay";
        power-domains = <&power RK3588_PD_VI>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_pin>;
        interrupt-parent = <&gpio1>;
        interrupts = <RK_PB2 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio1 RK_PA7 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio1 RK_PB1 GPIO_ACTIVE_HIGH>;
        plugin-det-gpios = <&gpio2 RK_PB6 GPIO_ACTIVE_LOW>;

        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "HDMI-MIPI2";
        rockchip,camera-module-lens-name = "RK628-CSI";

        multi-dev-info {
            dev-idx-l = <2>;
            dev-idx-r = <4>;
            combine-idx = <2>;
            pixel-offset = <0>;
            dev-num = <2>;
        };

        port {
            hdmiin_out1: endpoint {
                remote-endpoint = <&hdmi_mipi2_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};
......
```

4.5.5 HDMI2BT1120 Switching

For HDMI2BT1120 link configuration, the RK628D is basically the same as the RK628F/H, and will not be repeated here.

Take RK3568 + BT1120_V4L2 for example

```
&i2c4 {
    pinctrl-names = "default";
    pinctrl-0 = <&i2c4m1_xfer>;
    clock-frequency = <400000>;
    status = "okay";
```

```
    rk628_bt1120: rk628_bt1120@50 {
        compatible = "rockchip,rk628-bt1120-v4l2";
        reg = <0x50>;
        status = "disabled";
        pinctrl-names = "default";
        pinctrl-0 = <&cif_dvp_clk &cif_dvp_bus16 &cif_dvp_bus8>;
        interrupt-parent = <&gpio2>;
        interrupts = <15 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio2 RK_PC0 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio2 RK_PB0 GPIO_ACTIVE_LOW>;
        plugin-det-gpios = <&gpio1 RK_PA2 GPIO_ACTIVE_LOW>;
        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "RK628-BT1120";
        rockchip,camera-module-lens-name = "NC";
        dual-edge = <1>;

        port {
            lt8619c_out: endpoint {
                remote-endpoint = <&cif_para_in>;
                bus-width = <16>;
                pclk-sample = <1>;
            };
        };
    };
};

&rkcif_dvp {
    status = "okay";

    port {
        /* Parallel bus endpoint */
        cif_para_in: endpoint {
            remote-endpoint = <&lt8619c_out>;
        };
    };
};

&rkcif {
    status = "okay";
};

&rkcif_mmu {
    status = "okay";
};
```

4.6 Enable HDCP Function

RK628 HDMIRX supports HDCP1.4, HDCP function is disabled by default, you should enable it if you need:

1. Enable HDCP in dtsRK628 nodes

```
                reset-gpios = <&gpio7 RK_PB4 GPIO_ACTIVE_LOW>;
                plugin-det-gpios = <&gpio0 13 GPIO_ACTIVE_HIGH>;
                power-gpios = <&gpio0 17 GPIO_ACTIVE_HIGH>;
+               hdcp-enable = <1>;
```

2. Flash HDCP RX Key

   HDCP Key needs to be purchased from HDCP association, We don't provide the Key. Please note that HDCP distinguishes TX Key and RX Key, the one we need is RX key;

   After getting the Key source file, use KeyConverter to split the Key and convert it into a flashing file with skf suffix . Please refer to the instructions of the tool for details.

   After converting to a.skf file, use RKDevInfoWriteTool for flashing, the tool will encrypt the Key, please refer to the tool instructions for details.

4.7 Enable scaler Function

Some platforms do not support MIPI CSI 4K reception, but RK628 HDMI IN can support 4K input, so no matter what the resolution input, CSI output is 1080P, dts only needs to add the following configuration:

```
                reset-gpios = <&gpio7 RK_PB4 GPIO_ACTIVE_LOW>;
                plugin-det-gpios = <&gpio0 13 GPIO_ACTIVE_HIGH>;
                power-gpios = <&gpio0 17 GPIO_ACTIVE_HIGH>;
+               scaler-en = <1>;
```

4.8 csi supports 2 lanes

csi configuration is 4 lanes by default:

```
csi->csi_lanes_in_use = USE_4_LANES;
```

If it needs to be configurated into 2 lanes, then configurate csi->csi_lanes_in_use into 2;

4.9 Continuous MIPI mode and non-continuous MIPI

Set continuous MIPI mode, which can be configured in DTS:

```
continues-clk = <1>;
```

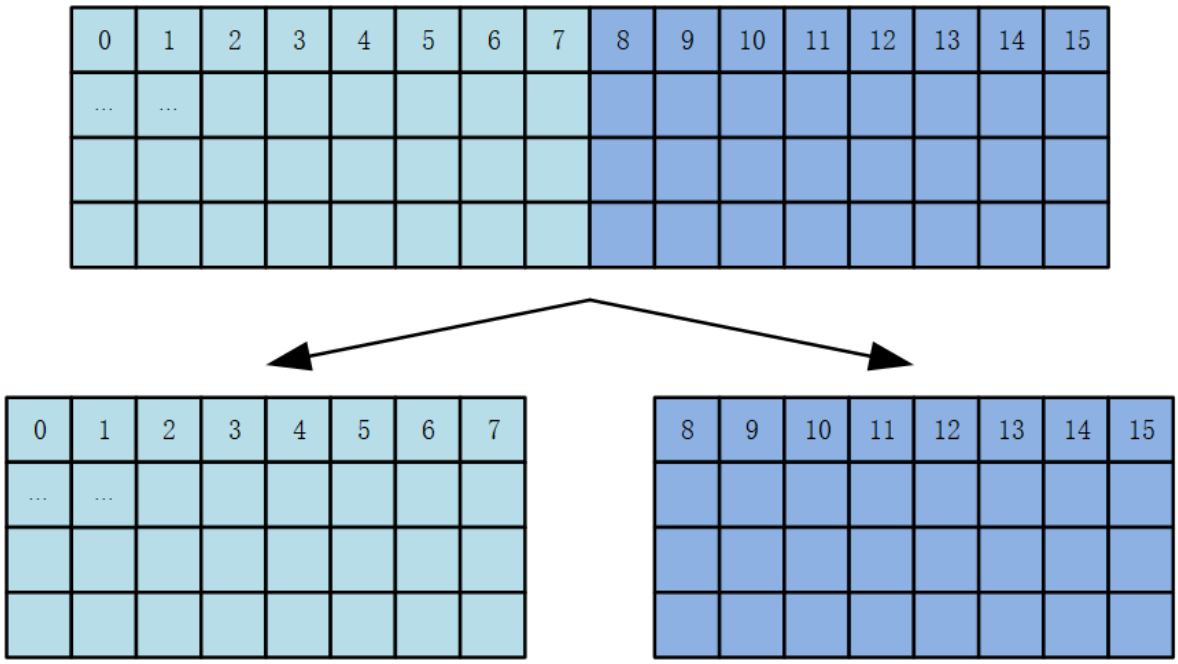
4.10 Dual MIPI mode configuration (only RK628F/H supports)

RK628F/H is limited by the MIPI DPHY rate, one MIPI can only support 4K30 at most, RK628F/H has two MIPI-CSIs, when docking with RK3588/RK3562 and other multi-channel MIPI-CSI master control platforms, in the 4K60 scenario, the 4K60 video data can be split into left and right halves, and the images can be transmitted through two 4lane MIPI-CSI respectively, and the main control platform will synthesize the images into 4K60.

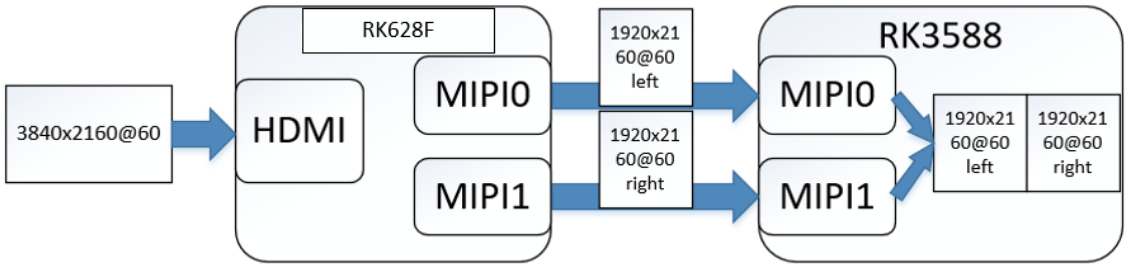
Note: Docking with SOC platforms such as RK3588, image stitching and synthesis can be completed directly in the VI driver, without additional delay and load.

4.10.1 Dual MIPI split mode explanation

RK628F/H internal can support to split the image received by HDMIRX into left and right halves, and send them to two CSI channels for transmission, as shown in the following figure:



Based on the above, when docking with a SOC platform with multiple MIPI interfaces, such as RK3588, you can use dual MIPI mode to receive image data at 4K60 resolution, as shown in the following figure.



4.10.2 Dual MIPI mode configuration

The following describes the configuration of the RK628F/H dual MIPI mode to connect with the RK3588 master controller to implement the 4K60 HDMI-IN function.

For details about the debugging of dual MIPI mode docking RK SOC, please refer to the document (SDK directory RKDocs/common/camera/HAL3)

"Rockchip_RK3588_HDMI_To_CSI_Dual_Mipi_Developer_Guide"

4.10.2.1 RK628F/H configuration

The RK628F/H driver version already supports dual MIPI mode, and by default, when the resolution of HDMI-IN is 4K60, two MIPI-CSI are used for transmission, and the DTS configuration multi-dev-info is parsed as follows:

> dev-idx-l:The MIPI interface number corresponding to the image on the left
>
> dev-idx-r: The MIPI interface number corresponding to the image on the right
>
> combine-idx: Merge the left and right images into one of the IDXs, and the links of the DTS are based on this IDX
>
> pixel-offset: Pixel offset, default setting 0
>
> dev-num: The number of MIPI devices that are spliced

For example, RK628F/H CSI0/1 is connected to RK3588 CSI0/1 RX separately:

```
&i2c3 {
    status = "okay";
    clock-frequency = <400000>;

    rk628_csi: rk628_csi@50 {
        reg = <0x50>;
        compatible = "rockchip,rk628-csi-v4l2";
        status = "okay";
        power-domains = <&power RK3588_PD_VI>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_pin>;
        interrupt-parent = <&gpio1>;
        interrupts = <RK_PB2 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio1 RK_PA7 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio1 RK_PB1 GPIO_ACTIVE_HIGH>;
        plugin-det-gpios = <&gpio2 RK_PB6 GPIO_ACTIVE_LOW>;
        continues-clk = <1>;

        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "HDMI-MIPI2";
        rockchip,camera-module-lens-name = "RK628-CSI";

        multi-dev-info {
            dev-idx-l = <2>;
            dev-idx-r = <4>;
            combine-idx = <2>;
            pixel-offset = <0>;
            dev-num = <2>;
        };

        port {
            hdmiin_out1: endpoint {
                remote-endpoint = <&hdmi_mipi2_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};
```

4.10.2.2 SOC driver code version requirements

At present, only RK3588/RK3562 supports dual MIPI mode, and only kernel-5.10 and later versions can support. The code version of the SOC master is required, so it is recommended to update the SDK (Android 12/13) code to the latest. If you only update the kernel-5.10 RK628 driver code, it may cause the compilation to fail, that is, the SDK code version is too old and does not support dual MIPI, please update the SDK code version or contact the RK engineers in redmine to solve.

4.11 Reception capability for different chip platforms

Because the isp/vicap performances from various chip platforms are different, whose maximum reception capability for image are different also. Please refer to :

chip platform	reception controller	maximum resolution supported
RK3288/RK3326/RK3368	isp	1920x1080P60
RK3399	isp	3840x2160P30 (NOTE: isp needs overclock)
RK3566/RK3568	vicap/isp	3840x2160P30
RK3588/RK3562	vicap	3840x2160P60 (dual MIPI)

4.11.1 The method of isp overclock configuration

- RK3399 configures PLL_NPLL to 650M:

```
--- a/arch/arm64/boot/dts/rockchip/rk3399-vop-clk-set.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399-vop-clk-set.dtsi
@@ -148,7 +148,7 @@
                         <50000000>, <100000000>,
                         <75000000>, <75000000>,
                         <816000000>, <816000000>,
-                        <600000000>, <200000000>,
+                        <650000000>, <200000000>,
                         <800000000>, <150000000>,
                         <75000000>, <37500000>,
                         <300000000>, <100000000>,
```

- Change the maximum frequency supported of rk3399 isp to 650M:

```
--- a/drivers/media/platform/rockchip/isp1/dev.c
+++ b/drivers/media/platform/rockchip/isp1/dev.c
@@ -757,7 +757,7 @@ static const unsigned int rk3368_isp_clk_rate[] = {

 /* isp clock adjustment table (MHz) */
 static const unsigned int rk3399_isp_clk_rate[] = {
-       300, 400, 600
+       300, 400, 650
 };

 static struct isp_irqs_data rk1808_isp_irqs[] = {
```

- If the rk3288 occurs isp error probabilistically , the isp can also try to be raised the frequency, generally only the highest frequency needs to be saved, and the modification is only used to locate the problem.

```
--- a/drivers/media/platform/rockchip/isp1/dev.c
+++ b/drivers/media/platform/rockchip/isp1/dev.c
@@ -849,7 +849,7 @@ static const unsigned int rk1808_isp_clk_rate[] = {

 /* isp clock adjustment table (MHz) */
static const unsigned int rk3288_isp_clk_rate[] = {
-       150, 384, 500, 594
+       594
 };
```

- Configurate the isp frequency in the driver of the conversion chip

```
#define RK628_CSI_PIXEL_RATE_HIGH       600000000
...

static int rk628_csi_set_fmt(struct v4l2_subdev *sd,
                struct v4l2_subdev_pad_config *cfg,
                struct v4l2_subdev_format *format)
{
...
        if ((mode->width == 3840) && (mode->height == 2160)) {
                v4l2_dbg(1, debug, sd,
                        "%s res wxh:%dx%d, link freq:%llu, pixrate:%u\n",
                        __func__, mode->width, mode->height,
                        link_freq_menu_items[1], RK628_CSI_PIXEL_RATE_HIGH);
                __v4l2_ctrl_s_ctrl(csi->link_freq, 1);
                __v4l2_ctrl_s_ctrl_int64(csi->pixel_rate,
                        RK628_CSI_PIXEL_RATE_HIGH);
        }
...
}
```

In isp driver, a margin of 25% can be added to the frequency configured, so configure the appropriate frequency RK628_CSI_PIXEL_RATE_HIGH in the driver.

```
drivers/media/platform/rockchip/isp1/dev.c

static int __isp_pipeline_s_isp_clk(struct rkisp1_pipeline *p)
{
...
```

```
        ctrl = v4l2_ctrl_find(sd->ctrl_handler, V4L2_CID_PIXEL_RATE);
        if (!ctrl) {
                v4l2_warn(sd, "No pixel rate control in subdev\n");
                return -EPIPE;
        }

        /* calculate data rate */
        data_rate = v4l2_ctrl_g_ctrl_int64(ctrl) *
                    dev->isp_sdev.in_fmt.bus_width;
        data_rate >>= 3;
        do_div(data_rate, 1000 * 1000);

        /* increase 25% margin */
        data_rate += data_rate >> 2;
...
}
```

4.11.2 The method for configuring ISP by CMA memory

When HDMI IN receives image data on some platforms, according to the actual system load, there may be problems such as frame loss or MIPI receiving anomaly due to insufficient bandwidth. The anomaly log is as follows:

```
[ 228.999567] rkisp1: MIPI mis error: 0x00800000
[ 228.999925] rkisp1: CIF_ISP_PIC_SIZE_ERROR (0x00000001)
[ 228.999976] rkisp1: CIF_ISP_PIC_SIZE_ERROR (0x00000001)rkisp1:
CIF_ISP_PIC_SIZE_ERROR (0x00000001)
[ 229.000081] rkisp1: CIF_ISP_PIC_SIZE_ERROR (0x00000001)rkisp1:
CIF_ISP_PIC_SIZE_ERROR (0x00000001)
[ 229.000187] rkisp1: CIF_ISP_PIC_SIZE_ERROR (0x00000001)rkisp1:
CIF_ISP_PIC_SIZE_ERROR (0x00000001)
[ 229.000294] rkisp1: CIF_ISP_PIC_SIZE_ERROR (0x00000001)rkisp1:
CIF_ISP_PIC_SIZE_ERROR (0x00000001)
```

You should rise the DDR frequency at this time, if there is no use, use CMA memory for ISP to resolve this problem:

- Configutate CMA memory 128MB reserved in rockchip_defconfig

```
CONFIG_CMA=y
CONFIG_CMA_SIZE_MBYTES=128
```

- Configurate ISP in dts to close the IOMMU, and use CMA memory

```
&isp_mmu {
        status = "disabled";
};
```

4.12 EDID Configuration Method

RK628 supports EDID configuration, the resolutions supported by EDID in the driver codes currently are:

3840x2160P60, 3840x2160P30, 1920x1080P60, 1920x1080P30, 1280x720P60, 720x576P50, 720x480P60 and so on.

If you need to modify the resolution supported, you can modify the EDID in the driver codes directly:

```
static u8 rk628f_edid_init_data[] = {
    0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
    0x24, 0xD0, 0x8F, 0x62, 0x01, 0x00, 0x00, 0x00,
    0x2D, 0x21, 0x01, 0x03, 0x80, 0x78, 0x44, 0x78,
    0x0A, 0xCF, 0x74, 0xA3, 0x57, 0x4C, 0xB0, 0x23,
    0x09, 0x48, 0x4C, 0x21, 0x08, 0x00, 0x61, 0x40,
    0x01, 0x01, 0x81, 0x00, 0x95, 0x00, 0xA9, 0xC0,
    0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x08, 0xE8,
    0x00, 0x30, 0xF2, 0x70, 0x5A, 0x80, 0xB0, 0x58,
    0x8A, 0x00, 0xC4, 0x8E, 0x21, 0x00, 0x00, 0x1E,
    0x02, 0x3A, 0x80, 0x18, 0x71, 0x38, 0x2D, 0x40,
    0x58, 0x2C, 0x45, 0x00, 0xB9, 0xA8, 0x42, 0x00,
    0x00, 0x1E, 0x00, 0x00, 0x00, 0xFC, 0x00, 0x49,
    0x46, 0x50, 0x20, 0x44, 0x69, 0x73, 0x70, 0x6C,
    0x61, 0x79, 0x0A, 0x20, 0x00, 0x00, 0x00, 0xFD,
    0x00, 0x3B, 0x46, 0x1F, 0x8C, 0x3C, 0x00, 0x0A,
    0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x01, 0xA8,

    0x02, 0x03, 0x39, 0xF2, 0x4D, 0x01, 0x03, 0x12,
    0x13, 0x84, 0x22, 0x1F, 0x90, 0x5D, 0x5E, 0x5F,
    0x60, 0x61, 0x23, 0x09, 0x07, 0x07, 0x83, 0x01,
    0x00, 0x00, 0x6D, 0x03, 0x0C, 0x00, 0x10, 0x00,
    0x00, 0x44, 0x20, 0x00, 0x60, 0x03, 0x02, 0x01,
    0x67, 0xD8, 0x5D, 0xC4, 0x01, 0x78, 0xC0, 0x00,
    0xE3, 0x05, 0x03, 0x01, 0xE4, 0x0F, 0x00, 0x18,
    0x00, 0x02, 0x3A, 0x80, 0x18, 0x71, 0x38, 0x2D,
    0x40, 0x58, 0x2C, 0x45, 0x00, 0xB9, 0xA8, 0x42,
    0x00, 0x00, 0x1E, 0x08, 0xE8, 0x00, 0x30, 0xF2,
    0x70, 0x5A, 0x80, 0xB0, 0x58, 0x8A, 0x00, 0xC4,
    0x8E, 0x21, 0x00, 0x00, 0x1E, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x93,
};
```

An EDID editing tool recommended: http://www.quantumdata.com/support/downloads/980/release_5_05/R_980mgr_5.05_Win32.msi

4.13 camera3_profiles.xml configuration

Use camera frame to preview, you need to adapt the camera3_profiles.xml and register the camera device.

The camera3_profiles.xml file corresponds to the file of the specific chip platform in the SDK directory:

```
hardware/rockchip/camera/etc/camera/camera3_profiles_rk3xxx.xml
```

The main configuration precautions are as follows, please refer to the configuration method of SOC Sensor for details:

- **name**: It must be the same as the driver name, and there is case distinction.
- **moduleId**: must be the same as the index configured in the driver dts;

```
    </Profiles>
    <Profiles cameraId="0" name="rk628-csi" moduleId="m00">
        <Supported_hardware>
            <hwType value="SUPPORTED_HW_RKISP1"/>
```

```
&rk628_csi {
        status = "okay";
        /*
         * If the hpd output level is inverted on the circuit,
         * the following configuration needs to be enabled.
         */
        /* hpd-output-inverted; */
        plugin-det-gpios = <&gpio0 13 GPIO_ACTIVE_HIGH>;
        power-gpios = <&gpio0 17 GPIO_ACTIVE_HIGH>;
        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "RK628-CSI";
        rockchip,camera-module-lens-name = "NC";
```

- scaler.availableStreamConfigurations/scaler.availableMinFrameDurations/ scaler.availableStallDurations: The resolution supported by the driver and the minimum frame interval time need to be correctly configured, and if new resolution support needs to be added to the driver, the configuration should be added accordingly.

```
<scaler.availableStreamConfigurations value="BLOB,3840x2160,OUTPUT,
    BLOB,1920x1080,OUTPUT,
    BLOB,1280x720,OUTPUT,
    BLOB,720x576,OUTPUT,
    BLOB,720x480,OUTPUT,
    YCbCr_420_888,3840x2160,OUTPUT,
    YCbCr_420_888,1920x1080,OUTPUT,
    YCbCr_420_888,1280x720,OUTPUT,
    YCbCr_420_888,720x576,OUTPUT,
    YCbCr_420_888,720x480,OUTPUT,
    IMPLEMENTATION_DEFINED,3840x2160,OUTPUT,
    IMPLEMENTATION_DEFINED,1920x1080,OUTPUT,
    IMPLEMENTATION_DEFINED,1280x720,OUTPUT,
    IMPLEMENTATION_DEFINED,720x576,OUTPUT,
    IMPLEMENTATION_DEFINED,720x480,OUTPUT" />
<scaler.availableMinFrameDurations value="BLOB,3840x2160,33333333,
    BLOB,1920x1080,16666667,
    BLOB,1280x720,16666667,
    BLOB,720x576,20000000,
    BLOB,720x480,16666667,
    YCbCr_420_888,3840x2160,33333333,
    YCbCr_420_888,1920x1080,16666667,
    YCbCr_420_888,1280x720,16666667,
    YCbCr_420_888,720x576,20000000,
    YCbCr_420_888,720x480,16666667,
    IMPLEMENTATION_DEFINED,3840x2160,33333333,
    IMPLEMENTATION_DEFINED,1920x1080,16666667,
    IMPLEMENTATION_DEFINED,1280x720,16666667,
    IMPLEMENTATION_DEFINED,720x576,20000000,
    IMPLEMENTATION_DEFINED,720x480,16666667"/>
<scaler.availableStallDurations value="BLOB,3840x2160,33333333,
    BLOB,1920x1080,16666667,
    BLOB,1280x720,16666667,
    BLOB,720x576,20000000,
    BLOB,720x480,16666667"/>
```

- **sensor.orientation**: Image rotation angle, supports 0, 90, 180, 270.

```
<sensor.maxAnalogSensitivity value="2400"/> <!-- HAL
<sensor.orientation value="0"/>
<sensor.profileHueSatMapDimensions value="0,0,0"/>
```

4.14 HDMI IN APK Adaptation Method

4.14.1 Android 9/10/11 Version

For Android 9/10/11 version, the corresponding kernel version is kernel-4.4/kernel-4.19, please use the for-all code patch package, and only the cameraHAL3 framework can be used.

4.14.1.1 Get and Compile APK Source Codes

APK source codes are provided in the SDK directory:

```
RKDocs/common/hdmi-in/apk/rkCamera2_based_on_CameraHal3_V1.3.tar.gz
```

Copy and unzip the source codes to the directory:

```
packages/apps/
```

Take RK3288 for example, you can modify the codes in device/rockchip/rk3288/ directory according to the following to add rkCamera2 APK compiling:

```
diff --git a/device.mk b/device.mk
index 6667b5c..96f08f1 100644
--- a/device.mk
+++ b/device.mk
@@ -17,7 +17,8 @@
 PRODUCT_PACKAGES += \
     memtrack.$(TARGET_BOARD_PLATFORM) \
     WallpaperPicker \
-    Launcher3
+    Launcher3 \
+    rkCamera2

 #$_rbox_$_modify_$_zhengyang: add displayd
 PRODUCT_PACKAGES += \
```

4.14.1.2 APK Source codes Adaptation

The APK accesses the RK628 device node by ioctl to obtain the current connection status and resolution. RK628 device nodes may be different on isp1/isp2/vicap links. You need to modify the APK source code for your needs. You can refer to Examples for debugging command section for the device node.

```
rkCamera2/jni/native.cpp
```

```
static void openDevice(JNIEnv *env, jobject thiz)
{
    (void)*env;
    (void)thiz;

    char video_name[64];
    memset(video_name, 0, sizeof(video_name));
    strcat(video_name, "/dev/v4l-subdev2");

    camFd = open(video_name, O_RDWR);
    if (camFd < 0) {
        LOGE("open %s failed,erro=%s",video_name,strerror(errno));
    } else {
        LOGD("open %s success,fd=%d",video_name,camFd);
    }
}
```

The codes to get the current connection status and resolution are as follows:

```
static void getDeviceFormat(int *format)
{
    struct v4l2_control control;
    memset(&control, 0, sizeof(struct v4l2_control));
    control.id = V4L2_CID_DV_RX_POWER_PRESENT;
    int err = ioctl(camFd, VIDIOC_G_CTRL, &control);
    if (err < 0) {
        LOGV("Set POWER_PRESENT failed ,%d(%s)", errno, strerror(errno));
    }

    unsigned int noSignalAndSync = 0;
    ioctl(camFd, VIDIOC_G_INPUT, &noSignalAndSync);
    LOGV("noSignalAndSync ? %s",noSignalAndSync?"YES":"NO");

    struct v4l2_dv_timings dv_timings;
    memset(&dv_timings, 0 ,sizeof(struct v4l2_dv_timings));
    err = ioctl(camFd, VIDIOC_SUBDEV_QUERY_DV_TIMINGS, &dv_timings);
    if (err < 0) {
        LOGV("Set VIDIOC_SUBDEV_QUERY_DV_TIMINGS failed ,%d(%s)", errno, strerror(errno));
    }

    format[0] = dv_timings.bt.width;
    format[1] = dv_timings.bt.height;
    format[2] = control.value && !noSignalAndSync;
}
```

Since the device node is being accessed on the APK, we need to make sure selinux is turned off, which we can check with getenforce:

```
rk3288:/ # getenforce
Enforcing
rk3288:/ # setenforce 0
rk3288:/ #
rk3288:/ # getenforce
Permissive
```

4.14.1.3 Preparation before APK debugging

Drivers need to be debugged first before APK debugging. Refer to the Driver Debugging Method section. The second step is to make sure that the camera device is correctly registered with CameraHal, which you can check with the following commands. If it is not, you need to check the camera3_profiles.xml configuration. Refer to the camera3_profiles.xml configuration file description section.

```
rk3288:/ #
rk3288:/ # dumpsys media.camera

== Service global info: ==

Number of camera devices: 1
Number of normal camera devices: 1
    Device 0 maps to "0"
Active Camera Clients:
[]
Allowed user IDs: 0

== Camera service events log (most recent at top): ==
  04-08 11:08:31 : USER_SWITCH previous allowed user IDs: <None>, current allowed user IDs: 0
  01-18 08:50:15 : ADD device 0, reason: (Device status changed from 0 to 1)
  01-18 08:50:15 : ADD device 0, reason: (Device added)
```

4.14.2 Android 12+ version

Android 12/13 version, the corresponding kernel version is kernel-5.10, and the driver patch uses kernel-5.10 patch pack. The application framework is divided into the camera framework and the TV framework (the same as the TIF followed). For details, please refer to the document "Rockchip_Android12+_HDMI_TO_MIPI_Developer_Guide" for debugging.

4.14.2.1 APK source

- packages/apps/TV/partner_support/samples: Provides the TV source data service through the framework and HAL
  layer, previewing the APK to interact. Because it is a hidden service running when boot, the APK is hidden icons on the desktop.
- hardware/rockchip/tv_input: TVHAL layer code, switching currents, hot swaps, resolution switching and other events
  to interact with the driver with commands.
- hardware/rockchip/camera: the cameraHAL layer code, which implements functions such as camera frame stream fetching and taking photos.
- vendor/rockchip/hardware/interfaces/hdmi: uses camera framework, is responsible for monitoring resolution changes and hot-swap events, and interacting with drivers and APKs.
- packages/apps/rkCamera2 : Preview apk, this app has 2 interfaces. The default MainActivity interface uses the TIF scheme to preview and interact with the above TV source data service through the framework layer, while the RockchipCamera2 interface uses the Camera scheme to preview and use the standard Camera API to open the cameraid of the MIPI node.
  The APK icon name on the desktop is HdmiIn, and customers usually redevelop and replace it with their own preview APK.
- In the default code of the SDK, HDMI IN function is disabled, and to enable the HDMI IN function, the following properties need to be configured. When enabled, the related modules including the above APKs will be compiled:

```
vim device/rockchip/rk3588/BoardConfig.mk
BOARD_HDMI_IN_SUPPORT := true
```

- Enable when using camera framework:

```
vim device/rockchip/rk3588/BoardConfig.mk
CAMERA_SUPPORT_HDMI := true
```

Only CAMERA_SUPPORT_HDMI is configured, not BOARD_HDMI_IN_SUPPORT. In this case, if you want to use rkCamera2 for camera preview, you need to add rkCamera2 to the compilation and configure the property persist.sys.hdmiinmode to be 2. Please refer to the document for detail:

"Rockchip_Developer_Guide_HDMI_RX_CN"

```
PRODUCT_PACKAGES += \
    rkCamera2
```

4.14.2.2 APK preview explanation

The APK supports RK3588 HDMI RX channel data preview and HDMI to MIPI-CSI path data preview, it should be switched when using.

For TIF preview mode, you need to set the MIPI-CSI2 channel:

```
setprop vendor.tvinput.hdmiin.type 1
```

For camera preview mode, you need to open RockchipCamera2 interface, pay attention to enabling CAMERA_SUPPORT_HDMI.

```
setprop persist.sys.hdmiinmode 2
```

4.14.2.3 The differences of TIF preview and camera preview

	TIF	**Camera**
advantage	Low latency	The app can get the preview data for post-processing
disadvantage	Screen rotation, split-screen, and picture-in-picture functions are not supported; The app can not get the preview buffer data; Screenshot commands in screencap mode are not supported	The latency is higher than TIF

4.15 Driver Debugging Method

Driver debugging method is the same as SOC Sensor debugging, and you can refer to the redmine for details:

https://redmine.rock-chips.com/documents/53

4.15.1 Get debugging tool

You need to use media-ctl and v4l2-ctl tools. At present, the SDK will copy and integrate automatically when compiling the firmware, which is placed in the SDK directory:

```
hardware/rockchip/camera/etc/tools/
```

If the SDK version is old, you can get it from redmine:

https://redmine.rock-chips.com/documents/104

Push media-ctl and v4l2-ctl to /vender/bin/ directory of the device with adb.

4.15.2 Examples for debugging command

For example, when RK3288 + RK628 receive 1920x1080P resolution, the specific debugging needs to be modified according to the actual situation. Note that the following debug commands need to be typed from top to bottom, or they may not work due to a lack of configuration.

- Check link and topological structure:

Carry out the commands to check the topological structure of media node and connect according to specific links on different chips, it may be /dev/media0 or /dev/media1.

```
media-ctl -d /dev/media0 -p
```

Take parts of RK628 codes for example, RK628 device is /dev/v4l-subdev2, and HDMI IN resolution recognized is 1920x1080.

```
...
- entity 8: m00_b_rk628-csi rk628-csi (1 pad, 1 link)
            type V4L2 subdev subtype Sensor
            device node name /dev/v4l-subdev2
        pad0: Source
                [fmt:UYVY2X8/1920x1080]
                -> "rockchip-mipi-dphy-rx":0 [ENABLED]
...
```

- Configure the link connection:

Once the device is reset and started, the link is connected by default. When you open and then exit by HDMI IN APK, the link will be disconnected. You should check the topological structure, and reconnect if there is no [ENABLED].

```
media-ctl -d /dev/media0 -l \
'"m00_b_rk628-csi rk628-csi":0->"rockchip-mipi-dphy-rx":0 [1]'
media-ctl -d /dev/media0 -l \
'"rockchip-mipi-dphy-rx":1->"rkisp1-isp-subdev":0 [1]'
media-ctl -d /dev/media0 -l '"rkisp1-input-params":0->"rkisp1-isp-subdev":1 [1]'
media-ctl -d /dev/media0 -l '"rkisp1-isp-subdev":2->"rkisp1_mainpath":0 [1]'
media-ctl -d /dev/media0 -l '"rkisp1-isp-subdev":2->"rkisp1_selfpath":0 [1]'
media-ctl -d /dev/media0 -l '"rkisp1-isp-subdev":3->"rkisp1-statistics":0 [1]'
```

- Configurate the resolution:

```
media-ctl -d /dev/media0 \
--set-v4l2 '"rkisp1-isp-subdev":0[fmt:UYVY2X8/1920x1080]'
media-ctl -d /dev/media0 \
--set-v4l2 '"rkisp1-isp-subdev":0[crop:(0,0)/1920x1080]'
media-ctl -d /dev/media0 \
--set-v4l2 '"rkisp1-isp-subdev":2[fmt:UYVY2X8/1920x1080]'
media-ctl -d /dev/media0 \
--set-v4l2 '"rkisp1-isp-subdev":2[crop:(0,0)/1920x1080]'
```

- Get the image data flow:

Check configuration result:

```
media-ctl -d /dev/media0 -p
```

Get the image data flow:

```
v4l2-ctl --verbose -d /dev/video0 --set-fmt-
video=width=1920,height=1080,pixelformat='NV12' --stream-mmap=4
```

Grab the image yuv file, and adb pull and check with 7yuv or other tools:

```
v4l2-ctl -d /dev/video0 --set-fmt-video=width=1920,height=1080,pixelformat='NV12'
--stream-mmap=3 --stream-skip=4 --stream-to=/data/1920x1080p60_nv12.yuv --stream-
count=5 --stream-poll
```

If everything is OK with receiving the image data normally, there will be frame rates.

Refer to the log followed:

```
VIDIOC_QUERYCAP: ok
VIDIOC_G_FMT: ok
VIDIOC_S_FMT: ok
Format Video Capture Multiplanar:
        Width/Height      : 1920/1080
        Pixel Format      : 'NV12'
        Field             : None
        Number of planes  : 1
        Flags             :
        Colorspace        : Default
        Transfer Function : Default
        YCbCr Encoding    : Default
        Quantization      : Full Range
```

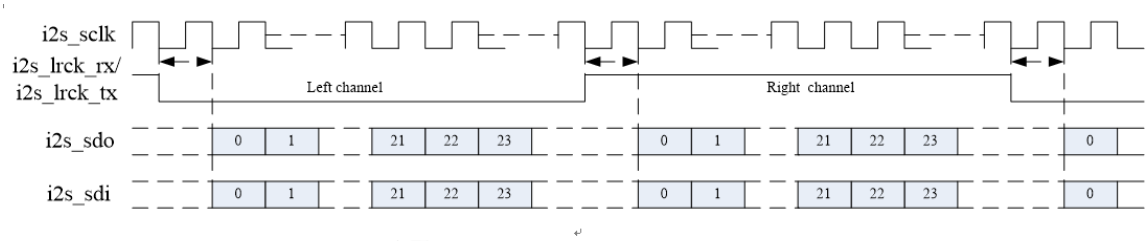
```
        Plane 0           :
           Bytes per Line : 1920
           Size Image     : 3110400
VIDIOC_G_SELECTION: ok
VIDIOC_S_SELECTION: ok
VIDIOC_REQBUFS: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_QUERYBUF: ok
VIDIOC_QBUF: ok
VIDIOC_STREAMON: ok
idx: 0 seq:      0 bytesused: 3110400 ts: 131.560377
idx: 1 seq:      1 bytesused: 3110400 ts: 131.577023 delta: 16.646 ms
idx: 2 seq:      2 bytesused: 3110400 ts: 131.593697 delta: 16.674 ms
idx: 3 seq:      3 bytesused: 3110400 ts: 131.610363 delta: 16.666 ms
idx: 0 seq:      4 bytesused: 3110400 ts: 131.627033 delta: 16.670 ms fps: 60.01
idx: 1 seq:      5 bytesused: 3110400 ts: 131.643721 delta: 16.688 ms fps: 59.99
idx: 2 seq:      6 bytesused: 3110400 ts: 131.660390 delta: 16.669 ms fps: 59.99
idx: 3 seq:      7 bytesused: 3110400 ts: 131.677058 delta: 16.668 ms fps: 59.99
```

4.16 Audio Module Introduction

The audio component can be set as output or input, depending on the scenarios of HDMIRX or HDMITX. In the scenario of HDMIRX, the audio data received by RK628 HDMI is unpackaged and output through the I2S interface. But in the scenario of HDMITX, RK628D I2S is used as a data input to receive audio data, and then output through HDMITX after packaging. The configuration methods of these two scenarios:

4.16.1 HDMIRX

HDMIRX usually initializes the audio module when the driver is loaded, and no additional configuration is needed. When HDMITX plays audio data, the audio signal is output through the RK628 I2S (RK628 must be the master). The RK628 I2S can be directly connected to DAC and SOC that support the I2S interface. The I2S format is as follows:



I2S normal mode timing format

When connecting to the SOC, the RK628 I2S does not need additional configuration; a sound card device can be created for the system by dummy_codec:

```
rk628_dc: rk628-dc {
        compatible = "rockchip,dummy-codec";
        #sound-dai-cells = <0>;
```

```
};

hdmiin-sound {
        compatible = "simple-audio-card";
        simple-audio-card,format = "i2s";
        simple-audio-card,name = "rockchip,hdmiin";
        simple-audio-card,bitclock-master = <&dailink0_master>;
        simple-audio-card,frame-master = <&dailink0_master>;
        status = "okay";
        simple-audio-card,cpu {
                        sound-dai = <&i2s0>;
        };
        dailink0_master: simple-audio-card,codec {
                        sound-dai = <&rk628_dc>;
        };
};
```

When connecting with the DAC, the software does not need the above configuration, the DAC can directly output analog signals. But currentlly, most of the DAC is required to provide a MCLK. When RK628 is designing, there is no reserved MCLK, so now, we can use RK628 GPIO1_A0 (H7) as TEST_CLKO to output 128FS clock. This method can make it compatible with most of the current DAC, and we have verified: CS4344, ES7144. You should enable GPIO1_A0 outputing MCLK during initialization, and call the following interfaces. The modification is as follows (please pay attention to the driver used in your project) :

```
--- a/drivers/media/i2c/rk628/rk628_hdmirx.h
+++ b/drivers/media/i2c/rk628/rk628_hdmirx.h
@@ -392,5 +392,5 @@ bool rk628_audio_ctsnints_enabled(HAUDINFO info);
 void rk628_csi_isr_ctsn(HAUDINFO info, u32 pdec_ints);
 void rk628_csi_isr_fifoints(HAUDINFO info, u32 fifo_ints);
 int rk628_is_avi_ready(struct rk628 *rk628, bool avi_rcv_rdy);
+void rk628_hdmirx_audio_set_mclk_output(HAUDINFO info);

--- a/drivers/media/i2c/rk628/rk628_csi.c
+++ b/drivers/media/i2c/rk628/rk628_csi.c
@@ -957,6 +957,9 @@ static void rk628_csi_initial_setup(struct rk628_csi *csi)
 {
        struct v4l2_subdev_edid def_edid;

+       //enable rk628 mclk
+       rk628_hdmirx_audio_set_mclk_output(csi->audio_info);

--- a/drivers/media/i2c/rk628/rk628_csi_v4l2.c
+++ b/drivers/media/i2c/rk628/rk628_csi_v4l2.c
@@ -1096,6 +1096,9 @@ static void rk628_csi_initial_setup(struct v4l2_subdev *sd)
        struct rk628_csi *csi = to_csi(sd);
        struct v4l2_subdev_edid def_edid;

+       //enable rk628 mclk
+       rk628_hdmirx_audio_set_mclk_output(csi->audio_info);
```

For the RK3288 EVB, I2S2 of RK3288 I2S and RT5651 are used. and RK628 I2S is connected with I2S2 of RT5651. The function of switching different channels to record and play can be realized by switching the internal route of RT5651 during using. The corresponding dts configuration is as follows:

```
hdmiin-sound {
        compatible = "rockchip,rockchip-rt5651-rk628-sound";
        rockchip,cpu = <&i2s>;
        rockchip,codec = <&rt5651>;
        status = "okay";
};
```

By default, i2s can output only when it is previewed. If it needs to output all the time, DTS needs to add the i2s-enable-default configuration.

```
&i2c4 {
        clock-frequency = <400000>;
        status = "okay";
        rk628_csi_v4l2: rk628_csi_v4l2@50 {
                reg = <0x50>;
                compatible = "rockchip,rk628-csi-v4l2";
                ....
                i2s-enable-default;
                ....
        };
};
```

4.16.2 HDMITX

HDMITX configuration is relatively simple, you just need to configure the HDMI sound card. The RK628 I2S is connected to the I2S0 of the SOC as follows:

```
hdmi_sound: hdmi-sound {
        compatible = "simple-audio-card";
        simple-audio-card,format = "i2s";
        simple-audio-card,name = "hdmi-sound";
        status = "okay";
        simple-audio-card,cpu {
                sound-dai = <&i2s0>;
        };
        simple-audio-card,codec {
                sound-dai = <&rk628_hdmi>;
        };
};
```

4.16.3 Common Audio Problems and Solutions

4.16.3.1 I2S without output

```
echo 0x000 0x6000220 > /sys/kernel/debug/regmap/0-0050-grf/registers // IOMUX of
i2s
echo 0x70 0xfffff55c > /sys/kernel/debug/regmap/0-0050-grf/registers
echo 0x70 0x155c155c > /sys/kernel/debug/regmap/0-0050-grf/registers // set
output
```

4.16.3.2 Enable to print v4l2_dbg

```
diff --git a/drivers/media/i2c/rk628_csi.c b/drivers/media/i2c/rk628_csi.c
index 5e8e3710a82f..638ac2acc472 100644
--- a/drivers/media/i2c/rk628_csi.c
+++ b/drivers/media/i2c/rk628_csi.c
@@ -34,7 +34,7 @@
 #include <video/videomode.h>
 #include "rk628_csi.h"

-static int debug;
+static int debug = 3;
 module_param(debug, int, 0644);
 MODULE_PARM_DESC(debug, "debug level (0-3)");

diff --git a/include/media/v4l2-common.h b/include/media/v4l2-common.h
index cdc87ec61e54..f159118d4a6b 100644
--- a/include/media/v4l2-common.h
+++ b/include/media/v4l2-common.h
@@ -82,7 +82,7 @@
 #define v4l2_dbg(level, debug, dev, fmt, arg...)                        \
        do {                                                             \
                if (debug >= (level))                                    \
-                       v4l2_printk(KERN_DEBUG, dev, fmt , ## arg);      \
+                       v4l2_printk(KERN_INFO, dev, fmt , ## arg);       \
        } while (0)

 /**
```

4.16.3.3 Application recording data noise problem

This problem may cause by the difference of HDMITX sampling rate and HAL recording sampling rate .

1. Check the sampling rate of HDMITX

Open the kernel log:`echo 3 > /sys/module/rk628_csi/parameters/debug`

2. After openning, check the sampling rate of TX from following logs:

```
m00_b_rk628-csi rk628-csi: rk628_hdmirx_audio_fs: clkrate:1500 tmdsclk:74250000,
n_decoded:6144, cts_decoded:74250, fs_audio:48000
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_FILLSTS1:0xfe, single offset:0, total offset:-2
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_ISTS:0x4
m00_b_rk628-csi rk628-csi: rk628_hdmirx_audio_fs: clkrate:1500 tmdsclk:74250000,
n_decoded:6144, cts_decoded:74250, fs_audio:48000
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_FILLSTS1:0xfc, single offset:-2, total offset:-4
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_ISTS:0x4
m00_b_rk628-csi rk628-csi: rk628_hdmirx_audio_fs: clkrate:1500 tmdsclk:74250000,
n_decoded:6144, cts_decoded:74250, fs_audio:48000
```

```
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_FILLSTS1:0xfc, single offset:0, total offset:-4
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_ISTS:0x4
m00_b_rk628-csi rk628-csi: rk628_hdmirx_audio_fs: clkrate:1500 tmdsclk:74250000,
n_decoded:6144, cts_decoded:74250, fs_audio:48000
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_FILLSTS1:0xfc, single offset:0, total offset:-4
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_ISTS:0x4
m00_b_rk628-csi rk628-csi: rk628_hdmirx_audio_fs: clkrate:1500 tmdsclk:74250000,
n_decoded:6144, cts_decoded:74250, fs_audio:48000
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_FILLSTS1:0xfa, single offset:-2, total offset:-6
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_ISTS:0x4
m00_b_rk628-csi rk628-csi: rk628_hdmirx_audio_fs: clkrate:1500 tmdsclk:74250000,
n_decoded:6144, cts_decoded:74250, fs_audio:48000
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_FILLSTS1:0xfc, single offset:2, total offset:-4
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_ISTS:0x4
m00_b_rk628-csi rk628-csi: rk628_hdmirx_audio_fs: clkrate:1500 tmdsclk:74250000,
n_decoded:6144, cts_decoded:74250, fs_audio:48000
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_FILLSTS1:0xfa, single offset:-2, total offset:-6
m00_b_rk628-csi rk628-csi: rk628_csi_delayed_work_audio:
HDMI_RX_AUD_FIFO_ISTS:0x4
```

3. HAL recording sampling rate:

```
getprop vendor.hdmiin.audiorate
```

4.16.3.4 Set IOMUX directly

```
 static void rk628_hdmirx_initial_setup(struct rk628_hdmirx *hdmirx)
 {
        struct v4l2_subdev_edid def_edid;
@@ -783,6 +937,8 @@ static void rk628_hdmirx_initial_setup(struct rk628_hdmirx
*hdmirx)

        // ddc and hpd pinctrl
        regmap_write(hdmirx->grf, GRF_GPIO1AB_SEL_CON, 0x07000700);
+       //i2s pinctrl
+       regmap_write(hdmirx->grf, GRF_GPIO0AB_SEL_CON, 0x155c155c);

        rk628_hdmirx_controller_reset(hdmirx);
```

4.16.3.5 tmdsclk miscalculation

1 bitmask is the same to the clock

```
git sholsh@rk-intel-1:~/rk-sdk/android11-rk3399/kernel$ git show
007131063748ea69facbf4bbe7aaee71c34fd921
commit 007131063748ea69facbf4bbe7aaee71c34fd921
Author: Shunhua Lan <lsh@rock-chips.com>
Date:   Fri Apr 23 18:43:59 2021 +0800

    media: i2c: rk628csi: fix mask for clkrate and fs audio align to 100

    Signed-off-by: Shunhua Lan <lsh@rock-chips.com>
    Change-Id: I15b290319463f1b41e6908e54caa99ef9c6db4f4

diff --git a/drivers/media/i2c/rk628_csi.c b/drivers/media/i2c/rk628_csi.c
index 89c4a926a985..a970bc621e0e 100644
--- a/drivers/media/i2c/rk628_csi.c
+++ b/drivers/media/i2c/rk628_csi.c
@@ -1064,7 +1064,7 @@ static void rk628_csi_delayed_work_audio(struct work_struct
*work)

        /* fout=128*fs=ftmds*N/CTS */
        regmap_read(csi->hdmirx_regmap, HDMI_RX_HDMI_CKM_RESULT, &clkrate);
-       clkrate = clkrate & 0xfff;
+       clkrate = clkrate & 0xffff;
        /* tmdsclk = (clkrate/1000) * 49500000 */
        tmdsclk = clkrate * (49500000 / 1000);
        regmap_read(csi->hdmirx_regmap, HDMI_RX_PDEC_ACR_CTS, &cts_decoded);
@@ -1073,6 +1073,8 @@ static void rk628_csi_delayed_work_audio(struct work_struct
*work)
        if (cts_decoded != 0) {
                fs_audio = div_u64((tmdsclk * n_decoded), cts_decoded);
                fs_audio = div_u64(fs_audio, 128);
+               fs_audio = div_u64(fs_audio + 50, 100);
+               fs_audio *= 100;
        }
        v4l2_dbg(2, debug, sd,
                "%s: clkrate:%d tmdsclk:%llu, n_decoded:%d, cts_decoded:%d,
fs_audio:%llu\n",
```

Logs that show errors are as follows:

```
[67.440094] m00_b_rk628-csi rk628-csi:rk628_csi_delayed_work_audio:clkrate:1904
tmdsclk:94248000,n_decoded:6144,cts_decoded:297000,fs_audio:15232
```

For 4K, tmdsclk frequency calculated should be 297M:

tmdsclk = clkrate * (49500000 / 1000);

1904 ---> 0x770

6000 ---> 0x1770

tmdsclk 6*49500000 = 297000000

4.16.3.6 Set GPIO to output test clk

Quick Setting

```
grf: gpio0 iomux switching related, configurated as an IO port:
echo 0x70 0x0fff0000 > registers

gpio0 configurated as an output port, 1-0051-rk628-pinctrl:
echo 0xd0008 0x00EC00EC > registers

grf, choose the level to pull the signal:
HDMI_RX:
echo 0x300 0x11 > registers
ASYNC_IN:
echo 0x300 0x14 > registers
ASYNC_OUT:
echo 0x300 0x15 > registers
SCALER:
echo 0x300 0x16 > registers
```

4.16.3.7 IOMUX Special Handing of RK356X

For RK356X, because each I2S can reuse multiple sets of pins, we need to configure the GRF register GRF_IOFUNC_SEL4 additionally:

Bit	Attr	Reset Value	Description
14	RW	0x0	i2s3_iomux_sel I2S3 IO mux selection 1'b0:M0 mux solution 1'b1:M1 mux solution
12	RW	0x0	i2s2_iomux_sel I2S2 IO mux selection 1'b0:M0 mux solution 1'b1:M1 mux solution
11:10	RW	0x0	i2s1_iomux_sel I2S1 IO mux selection 2'b00:M0 mux solution 2'b01:M1 mux solution 2'b10:M2 mux solution 2'b11: Reserved

The pinctrl driver will configure the bit corresponded with GRF according to the mclk selected by the user, as follows:

```
static struct rockchip_mux_route_data rk3568_mux_route_data[] = {
    ....
    RK_MUXROUTE_GRF(1, RK_PA2, 1, 0x0310, WRITE_MASK_VAL(11, 10, 0)), /* I2S1 IO
mux M0 */
    RK_MUXROUTE_GRF(3, RK_PC6, 4, 0x0310, WRITE_MASK_VAL(11, 10, 1)), /* I2S1 IO
mux M1 */
    RK_MUXROUTE_GRF(2, RK_PD0, 5, 0x0310, WRITE_MASK_VAL(11, 10, 2)), /* I2S1 IO
mux M2 */
    RK_MUXROUTE_GRF(2, RK_PC1, 1, 0x0310, WRITE_MASK_VAL(12, 12, 0)), /* I2S2 IO
mux M0 */
    RK_MUXROUTE_GRF(4, RK_PB6, 5, 0x0310, WRITE_MASK_VAL(12, 12, 1)), /* I2S2 IO
mux M1 */
    RK_MUXROUTE_GRF(3, RK_PA4, 4, 0x0310, WRITE_MASK_VAL(14, 14, 0)), /* I2S3 IO
mux M0 */
    RK_MUXROUTE_GRF(4, RK_PC4, 5, 0x0310, WRITE_MASK_VAL(14, 14, 1)), /* I2S3 IO
mux M1 */
    ....
}
```

For the soc similar to rk628, which doesn't need mclk application, the corresponding bit will not be set. So you need to add manually, generally i2s will use sclk, here we add sclk (the newer RK356X SDK will add this patch) :

```
diff --git a/drivers/pinctrl/pinctrl-rockchip.c b/drivers/pinctrl/pinctrl-
rockchip.c
index 871c49ef0c2a..3d10e6c2ed27 100644
--- a/drivers/pinctrl/pinctrl-rockchip.c
+++ b/drivers/pinctrl/pinctrl-rockchip.c
@@ -1046,13 +1046,26 @@ static struct rockchip_mux_route_data
rk3568_mux_route_data[] = {
        RK_MUXROUTE_GRF(2, RK_PB0, 3, 0x0310, WRITE_MASK_VAL(9, 8, 0)), /* UART9
IO mux M0 */
        RK_MUXROUTE_GRF(4, RK_PC5, 4, 0x0310, WRITE_MASK_VAL(9, 8, 1)), /* UART9
IO mux M1 */
        RK_MUXROUTE_GRF(4, RK_PA4, 4, 0x0310, WRITE_MASK_VAL(9, 8, 2)), /* UART9
IO mux M2 */
+
        RK_MUXROUTE_GRF(1, RK_PA2, 1, 0x0310, WRITE_MASK_VAL(11, 10, 0)), /* I2S1
IO mux M0 */
        RK_MUXROUTE_GRF(3, RK_PC6, 4, 0x0310, WRITE_MASK_VAL(11, 10, 1)), /* I2S1
IO mux M1 */
        RK_MUXROUTE_GRF(2, RK_PD0, 5, 0x0310, WRITE_MASK_VAL(11, 10, 2)), /* I2S1
IO mux M2 */
+       RK_MUXROUTE_GRF(1, RK_PA3, 1, 0x0310, WRITE_MASK_VAL(11, 10, 0)), /* I2S1
IO mux sclk tx M0 */
+       RK_MUXROUTE_GRF(1, RK_PA4, 1, 0x0310, WRITE_MASK_VAL(11, 10, 0)), /* I2S1
IO mux sclk rx M0 */
+       RK_MUXROUTE_GRF(3, RK_PC7, 4, 0x0310, WRITE_MASK_VAL(11, 10, 1)), /* I2S1
IO mux sclk tx M1 */
+       RK_MUXROUTE_GRF(4, RK_PA6, 5, 0x0310, WRITE_MASK_VAL(11, 10, 1)), /* I2S1
IO mux sclk rx M1 */
+       RK_MUXROUTE_GRF(2, RK_PD1, 5, 0x0310, WRITE_MASK_VAL(11, 10, 2)), /* I2S1
IO mux sclk tx M2 */
+       RK_MUXROUTE_GRF(3, RK_PC3, 5, 0x0310, WRITE_MASK_VAL(11, 10, 2)), /* I2S1
IO mux sclk rx M2 */
```

```
        RK_MUXROUTE_GRF(2, RK_PC1, 1, 0x0310, WRITE_MASK_VAL(12, 12, 0)), /* I2S2
IO mux M0 */
        RK_MUXROUTE_GRF(4, RK_PB6, 5, 0x0310, WRITE_MASK_VAL(12, 12, 1)), /* I2S2
IO mux M1 */
+       RK_MUXROUTE_GRF(2, RK_PC2, 1, 0x0310, WRITE_MASK_VAL(12, 12, 0)), /* I2S2
IO mux sclk tx M0 */
+       RK_MUXROUTE_GRF(2, RK_PB7, 1, 0x0310, WRITE_MASK_VAL(12, 12, 0)), /* I2S2
IO mux sclk rx M0 */
+       RK_MUXROUTE_GRF(4, RK_PB7, 4, 0x0310, WRITE_MASK_VAL(12, 12, 1)), /* I2S1
IO mux sclk tx M1 */
+       RK_MUXROUTE_GRF(4, RK_PC1, 5, 0x0310, WRITE_MASK_VAL(12, 12, 1)), /* I2S1
IO mux sclk rx M1 */
        RK_MUXROUTE_GRF(3, RK_PA2, 4, 0x0310, WRITE_MASK_VAL(14, 14, 0)), /* I2S3
IO mux M0 */
        RK_MUXROUTE_GRF(4, RK_PC2, 5, 0x0310, WRITE_MASK_VAL(14, 14, 1)), /* I2S3
IO mux M1 */
+       RK_MUXROUTE_GRF(3, RK_PA3, 4, 0x0310, WRITE_MASK_VAL(14, 14, 0)), /* I2S3
IO mux sclk M0 */
+       RK_MUXROUTE_GRF(4, RK_PC3, 5, 0x0310, WRITE_MASK_VAL(14, 14, 1)), /* I2S3
IO mux sclk M1 */
        RK_MUXROUTE_GRF(1, RK_PA4, 3, 0x0314, WRITE_MASK_VAL(1, 0, 0)), /* PDM IO
mux M0 */
        RK_MUXROUTE_GRF(1, RK_PA6, 3, 0x0314, WRITE_MASK_VAL(1, 0, 0)), /* PDM IO
mux M0 */
        RK_MUXROUTE_GRF(3, RK_PD6, 5, 0x0314, WRITE_MASK_VAL(1, 0, 1)), /* PDM IO
mux M1 */
```

4.16.3.8 LRCK Special Handing of RK3399

If the direction is misconfigured, it may cause the sound card is opened without receiving any data, the LOG shows followed (the node path needs to be modified according to the actual situation) :

```
console:/ # cat /proc/asound/cards
 0 [rockchiphdmi   ]: rockchip_hdmi - rockchip,hdmi
                      rockchip,hdmi
 1 [rockchiphdmiin ]: rockchip_hdmiin - rockchip,hdmiin
                      rockchip,hdmiin

console:/ # cat /proc/asound/card1/pcm0c/sub0/status
state: RUNNING
owner_pid   : 1415
trigger_time: 1632716135.151992694
tstamp      : 0.000000000
delay       : 0
avail       : 0
avail_max   : 0
-----
hw_ptr      : 0
appl_ptr    : 0
```

The log when configuration is correct is similar to following:

```
console:/ # cat /proc/asound/card1/pcm0c/sub0/status
state: RUNNING
owner_pid   : 1421
trigger_time: 1632892868.412331044
tstamp      : 1632892881.181113577
delay       : 232
avail       : 232
avail_max   : 264
-----
hw_ptr      : 543736
appl_ptr    : 543504
```

This is because RK628 is the master and RK3399 is the slave device, which needs to be configured according to the circuit connection mode. If LRCK_TX is connected to RK628, the "rockchip,clk-trcm" configuration is as follows:

```
+&i2s0 {
+       rockchip,clk-trcm = <1>;
+       status = "okay";
+};

console:/ # io -4 0xff880008
ff880008:  18071f1f
```

If LRCK_RX is connected to RK628, "rockchip,clk-trcm" configuration is as follows:

```
+&i2s0 {
+       rockchip,clk-trcm = <2>;
+       status = "okay";
+};
Normal register is as followed:
console:/ # io -4 0xff880008
ff880008:  28071f1f
```

The explanations of I2S_CKR register and bit29:28 are as follows:

```
Tx and Rx Common Use
2'b00/2'b11:tx_lrck/rx_lrck are used as synchronous signal for TX
/RX respectively.
2'b01:only tx_lrck is used as synchronous signal for TX and RX.
2'b10:only rx_lrck is used as synchronous signal for TX and RX.
```

The code of RK628 can be updated to v15-210926 or newer versions, and refer to the audio configuration of this DTS: arch/arm64/boot/dts/rockchip/rk3399-evb-ind-lpddr4-rk628-hdmi2csi-v4l2-avb.dts

4.16.3.9 HDMI-IN Sound card Select Error

When the APK is opened, it will match the name of the sound card. In the early codes, there may be a name matching error, which leads to the problem that HDMI-IN sound card cannot be opened. The error message is as follows:

```
09-28 07:03:13.341   261  1386 D AudioHardwareTiny: card0 id:rockchiphdmi
```

```
09-28 07:03:13.341   261  1386 D AudioHardwareTiny: SPEAKER card, got
card=0,device=0
09-28 07:03:13.341   261  1386 D AudioHardwareTiny: HDMI card, got
card=0,device=0
09-28 07:03:13.341   261  1386 D AudioHardwareTiny: SPDIF card, got
card=0,device=0
09-28 07:03:13.341   261  1386 D AudioHardwareTiny: BT card, got card=0,device=0
09-28 07:03:13.342   261  1386 D AudioHardwareTiny: card1 id:rockchiphdmiin
09-28 07:03:13.342   261  1386 D AudioHardwareTiny: SPEAKER card, got
card=1,device=0
09-28 07:03:13.342   261  1386 D AudioHardwareTiny: HDMI card, got
card=1,device=0
09-28 07:03:13.342   261  1386 D AudioHardwareTiny: SPDIF card, got
card=1,device=0
09-28 07:03:13.342   261  1386 D AudioHardwareTiny: BT card, got card=1,device=0
09-28 07:03:13.342   261  1386 D AudioHardwareTiny: No exist
proc/asound/card2/id, break and finish parsing
09-28 07:03:13.343   261  1386 D AudioHardwareTiny: dump out device info
09-28 07:03:13.343   261  1386 D AudioHardwareTiny: dev_info SPEAKER  card=1,
device:0
09-28 07:03:13.343   261  1386 D AudioHardwareTiny: dev_info HDMI  card=1,
device:0
09-28 07:03:13.343   261  1386 D AudioHardwareTiny: dev_info SPDIF  card=1,
device:0
09-28 07:03:13.343   261  1386 D AudioHardwareTiny: dev_info BT  card=1, device:0
```

HDMI card, got card=1,device=0. the message is different from the true, so the sound card fails to be opened.

You need to update following patches to resolve, or update RK628 codes to V15-210926 or later versions.

```
hardware/rockchip/audio
commit 1b51a62fc62e9d63850e4e5a39f1e3cff0aa9b88 (HEAD)
Author: Shunhua Lan <lsh@rock-chips.com>
Date:   Tue Sep 28 17:25:44 2021 +0800

    [audio hal] use levenshtein distance for sound card matching

    Signed-off-by: Shunhua Lan <lsh@rock-chips.com>
    Change-Id: I16f1692715d710a0693ae74875b0272669a04ba2
```

4.16.4 Other Audio Documents Supplement

If RK628 is connected to SOC directly, please refer to 'RK628-DIRECT-TO-SOC.pdf'.

If RK628 is connected to SOC by AUDIO CODEC, please refer to 'RK628-RT5640-AUDIO-CONFIG.pdf'.

For the audio problems related to HDMI-IN APK, please refer to 'RK628-HDMIIN-APP-AUDIO.pdf'.

If the contents of other documents conflict with this document, please refer to this document.

4.17 Common Problem Debugging

4.17.1 Open log switch

You can open the log switch of RK628 by following commands, and then get kernel log by demsg commands:

```
echo 1 > /sys/module/rk628_csi/parameters/debug
```

If you want to grab the log of the power-on process, it's suggested to modify the code and to recompile and reflash kernel parts:

```
diff --git a/drivers/media/i2c/rk628_csi.c b/drivers/media/i2c/rk628_csi.c
index c763a9558169..bd7f3effb45a 100644
--- a/drivers/media/i2c/rk628_csi_v4l2.c
+++ b/drivers/media/i2c/rk628_csi_v4l2.c
@@ -34,7 +34,7 @@
 #include <video/videomode.h>
 #include "rk628_csi.h"

-static int debug;
+static int debug = 1;
 module_param(debug, int, 0644);
 MODULE_PARM_DESC(debug, "debug level (0-3)");

diff --git a/include/media/v4l2-common.h b/include/media/v4l2-common.h
index 1cc0c5ba16b3..e74f3a85f0b8 100644
--- a/include/media/v4l2-common.h
+++ b/include/media/v4l2-common.h
@@ -75,7 +75,7 @@
 #define v4l2_dbg(level, debug, dev, fmt, arg...)                        \
        do {                                                             \
                if (debug >= (level))                                    \
-                       v4l2_printk(KERN_DEBUG, dev, fmt , ## arg);      \
+                       v4l2_printk(KERN_INFO, dev, fmt , ## arg);       \
        } while (0)
```

4.17.2 Register Write and Read

RK628 register debugging node is as follows. And in this example, RK628 is connect to I2C1, and the address is 0x51:

```
rk3288:/ # ls -l /sys/kernel/debug/regmap/
total 0
drwxr-xr-x 2 root root 0 1970-01-01 00:00 0-001b
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-adapter
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-combrxphy
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-combtxphy
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-cru
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-csi
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-grf
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-hdmirx
drwxr-xr-x 2 root root 0 1970-01-01 00:00 2-001a
drwxr-xr-x 2 root root 0 1970-01-01 00:00 ff890000.i2s
drwxr-xr-x 2 root root 0 1970-01-01 00:00 ff96c000.video-phy
```

Register nodes are read-only by default, but if you want the register to be writable, you need to add the following modification:

```
diff --git a/drivers/base/regmap/regmap-debugfs.c b/drivers/base/regmap/regmap-
debugfs.c
index 3f0a7e262d69..b819645edd84 100644
--- a/drivers/base/regmap/regmap-debugfs.c
+++ b/drivers/base/regmap/regmap-debugfs.c
@@ -259,7 +259,7 @@ static ssize_t regmap_map_read_file(struct file *file, char
__user *user_buf,
                                    count, ppos);
 }

-#undef REGMAP_ALLOW_WRITE_DEBUGFS
+#define  REGMAP_ALLOW_WRITE_DEBUGFS
 #ifdef REGMAP_ALLOW_WRITE_DEBUGFS
 /*
  * This can be dangerous especially when we have clients such as
```

Read the register:

```
rk3288:/ # cat /sys/kernel/debug/regmap/1-0051-cru/registers
c0000: 00001063
c0004: 00001442
c0008: 00000000
c000c: 00000007
c0010: 00007f00
c0020: 00006010
c0024: 00000581
c0028: 00ef348b
c002c: 00000007
c0030: 00007f00
...
```

Write the register:

```
rk3288:/ # echo 0x000 0xffffffff > /sys/kernel/debug/regmap/1-0051-grf/registers
```

4.17.3 clk det abnormal problems

4.17.3.1 RK628D

If the signal is not detected by COMBRXPHY, it is possible that the HDMI is inserted into the valid level or the HPD valid level is configured incorrectly, resulting in HDMI signal not being input properly. It is necessary to check the plugin-det-gpios and hpd-outport-inverted configuration of the rk628_csi node, and at the same time, the HPD level status can be tested with the multimeter.

Note: It will retry when abnormally and the retry times are up to 2. The signal is normal if retry successfully.

```
rk628-csi-v4l2 1-0051: clk det over cnt:10, reg_0x6654:0x403f0000
rk628-csi-v4l2 1-0051: |d2_p| level detection anomaly
rk628-csi-v4l2 1-0051: clock detected failed, cfg resistance manual!
rk628-csi-v4l2 1-0051: err, clk not stable, reg_0x6630:0x87000d,
reg_0x6608:0x110100
m00_b_rk628-csi 1-0051: hdmi rxphy power on failed
```

According to the above logs, we can see that data2_p voltage detecting is abnormal. It's because of no signal or low level amplitude.

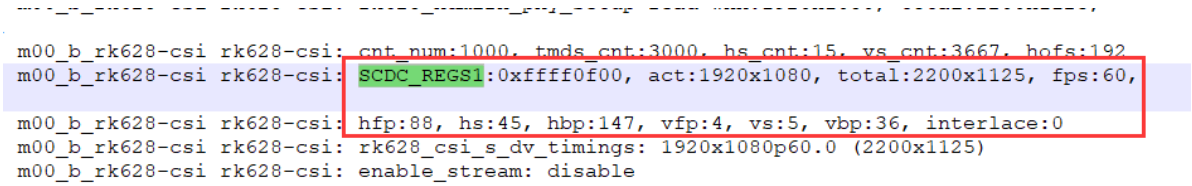
```
rk628-csi-v4l2 1-0051: clk det over cnt:10, reg_0x6654:0x403f0000
rk628-csi-v4l2 1-0051: Clock detection anomaly
rk628-csi-v4l2 1-0051: clock detected failed, cfg resistance manual!
rk628-csi-v4l2 1-0051: err, clk not stable, reg_0x6630:0x87000d,
reg_0x6608:0x110100
m00_b_rk628-csi 1-0051: hdmi rxphy power on failed
```

The above logs mean the frequency point detecting is abnormal, maybe it's because the frequency point is not within the support range of RK628.

```
25175, 27000, 33750, 40000, 59400, 65000, 68250, 74250,
75000, 83500, 85500, 88750, 928125, 101000, 102250, 108000,
118800, 119000, 135000, 148500, 150000, 162000, 165000, 297000
```

4.17.4 Judging whether HDMI RX is normal

After COMBRXPHY is locked normally, HDMI RX controller can parse to Timing normally. Timing is created by calculating, there may be some small errors, generally may be 1. For detailed Timing, please refer to the CEA standard documentation.



4.17.5 Open subdev Permission Exception

```
D JNI     : JNI CAMERA CALL init
I HdmiInput-navtive: JNI OnLoad
I HdmiInput-navtive: Apk Version: V1.2
E HdmiInput-navtive: openDevice(91): open /dev/v4l-subdev2 failed,erro=Permission denied
D RockchipCamera2: remove take pic button
D RockchipCamera2: recreatTextureview
I RockchipCamera2: textureView remove
D RockchipCamera2: onResume
```

You should comfirm whether the 666 permission is provided to /dev/v4l-subdev*.

Please check in device command line.

```
rk3288:/ # cat /vendor/ueventd.rc | grep subdev
/dev/v4l-subdev*          0666   media      camera
```

Currently SDK codes are including this submit, if there is not, you can modify in device/rockchip/common/ according to the codes followed.

```
wendingxian@ubuntu:~/rk3288_9.0_int/device/rockchip/common$ git show
commit 17112e1430b0f10a88b57c73ad19203e58f0eeff (HEAD -> mid_9.0, rk/rk/rk33/mid/9.0/develop)
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Tue Apr 27 21:26:48 2021 +0800

    ueventd.rockchip.rc: modify the /dev/v4l-subdev* permission to 666

    Signed-off-by: Dingxian Wen <shawn.wen@rock-chips.com>
    Change-Id: Id4848209fd983f7e525761f19c7f0420b9fee747

diff --git a/ueventd.rockchip.rc b/ueventd.rockchip.rc
index 1914490b..52935d49 100755
--- a/ueventd.rockchip.rc
+++ b/ueventd.rockchip.rc
@@ -178,7 +178,7 @@
 /dev/ovr0               0664   system         system

 #for rk_isp1
-/dev/v4l-subdev*         0660   media      camera
+/dev/v4l-subdev*         0666   media      camera

 /dev/video0          0660   media      camera
 /dev/video1          0660   media      camera
```

4.17.6 Can not recognize the signal

Step1. When HDMI is unplugged, there is no RK628 related logs printed on the serial port, which indicates that the gpio configuration of the det pin is wrong.

Step2. There is RK628 log printing when pulling out HDMI, however, there is no RK628 log printing when inserting. It indicates that the polarity of plugin-det-gpios pin is set oppositely. If the transistors inversion is added, then you should set transistors. To judge the plugging status, you can modify static int debug; to static int debug =3; in rk628_csi_v4l2.c , and it's correct if printing tx_5v_power_present: 1 when inserting, or printing tx_5v_power_present: 0 when pulling out.

Step3. There is no normal signal detected all the time, it may be the situations followed.

Case1. hpd pin inversion problem, only when RK628 provides the high level to hdmiin connector, hdmi can output. If the inverter is added, RK628 pin needs low output, which becomes high level in connector after inverting. Then hpd-output-inverted; should be added to the dts.

Case2. phy can not be locked, please refer to clk det abnormal problems;

Case3. resolution can not be locked, maybe HDMIRX detects that the bit rate is out of the specified range.

Case4. resolution can be lock, avi_rcv_rdy will set to 1 after signal locks, but if avi signal is not ready, it is possible that the interrupt pin is not configured. For example, the log followed is printing avi_rcv_rdy:0 all the time, it means it can recognize the normal resolutions. But avi_rcv_rdy is 0, so the signal is not locked, you should check whether the configuration of interrupt pin is correct first.

```
rk628d 1-0050: rk628_is_avi_ready PDEC_AVI_PB:0x10000840, avi_rcv_rdy:0
rk628d 1-0050: SCDC_REGS1:0x80000f00, act:1920x1080, total:2200x1125, fps:60,
pixclk:148500000
```

In this case, there will be probabilistic image segmentation interface when rk3288 grabs the picture.

4.17.7 Display Anomaly

1. Confirm whether the input source is DVI Mode, RK628D doesn't support DVI currently, you can check the log:

   ```
   rk628-csi-v4l2 1-0051: DVI mode detected
   ```

2. Confirm whether there is HDCP encryption in the input source or not, if there is, you need enable HDCP function of RK628 HDMIRX.

4.17.8 Only half of the screen is displayed

The biggest possible reason of this problem is that the source output of HDMI in the test scene is 4K60, but the docking master is an old platform, such as RK3399, RK3568, RK3288, etc., which can only support a single MIPI SOC, so RK628F/H will split the 4K60 into two halves of the left and right images, so the main controller receives only half of the images. In this scenario, it is recommended to directly modify the edid to support 4K30, that is, directly use the edid of RK628D:

```
diff --git a/drivers/media/i2c/rk628/rk628_csi_v4l2.c
b/drivers/media/i2c/rk628/rk628_csi_v4l2.c
index 32182d4433605..36568f5a081e0 100644
--- a/drivers/media/i2c/rk628/rk628_csi_v4l2.c
+++ b/drivers/media/i2c/rk628/rk628_csi_v4l2.c
@@ -1347,7 +1347,7 @@ static void rk628_csi_initial_setup(struct v4l2_subdev *sd)
        def_edid.start_block = 0;
        def_edid.blocks = 2;
        if (csi->rk628->version >= RK628F_VERSION)
-               def_edid.edid = rk628f_edid_init_data;
+               def_edid.edid = edid_init_data;
        else
                def_edid.edid = edid_init_data;
        rk628_csi_s_edid(sd, &def_edid);
```

4.17.9 Capture Failure

1. For the platforms after Android9.0 and linux

   You can capture by v4l2-ctl for the platforms after Android9.0 or linux.

   Check `media-ctl -d /dev/mediaX -p` first, the numbers of media nodes are different in different platforms. If you are not sure, in general, it will be media0 when turning off the cif controller; it may be media1 when rk3399 is connected to phy1. Please refer to the camera configurations.

   There are rk628 nodes under media-ctl topology, if not, then dts configurations are faulty. Please check by yourselves. In addition, all resolutions of the topology do not match the resolutions you input, so they must be set to the same resolution by media-ctl.

   And then execute:

   ```
   v4l2-ctl -d /dev/video0 --set-fmt-
   video=width=1920,height=1080,pixelformat=NV12 --stream-mmap=3
   ```

   If it prints <<<<<<<<<< all the time, capture succeeds, and the driver is normal already.

   If it keeps reporting errors, or reporting errors probabilistically:

   ```
   rkisp1: MIPI mis error:
   ```

   Generally it's related to the hardware, so please check whether the hdmi has furcation or the impedance matching is the same. Otherwise, most of customers prefer to connect the resistor to hdmi (it doesn't connect to hdmi in the schematic diagram by default), in this case, we set the resistance value to 0 if it's not necessary, because there may be errors probabilistically when the value is not 0. You should analyze detailedly, and check the signal from rk628 to the master.

   Or

   ```
   rkisp-vir0: MIPI error: overflow: 0x00000001
   ```

   This log is the problem which occurs easily when isp of rk356x rv11xx inputs by 4K, and it's suggested to use rk628+vicap, please refer to the vicap configuration mentioned in this document.

   In addition, if the SIZE_ERROR is printed as follows, maybe because the input is 720p,but the collection is 1080p. In this situation that resolution does not match, you should check all resolutions of topology, as well as whether v4l2-ctl collection resolution is correct. Of course, it is recommended to use 1080p to test at the beginning.

   ```
   rkisp1 ff910000.rkisp1: CIF_ISP_PIC_SIZE_ERROR
   ```

   By default, /dev/v4l-subdev2 is used to determine the resolution, but if the topology is /dev/v4l-subdev3 by `media-ctl -p`:

   ```
    - entity 70: m00_b_rk628-csi rk628-csi (1 pad, 1 link)
              type V4L2 subdev subtype Sensor
              device node name /dev/v4l-subdev3
   pad0: Source
       [fmt:UYVY2X8/1280x720]
       -> "rockchip-csi2-dphy0":0 []
   ```

In this case, the resolution cannot be recognized, neither can apk, and CIF_ISP_PIC_SIZE_ERROR may be printed. To resolve this problem, you need modify the jni/native.cpp of android apk:

```
strcat(video_name, "/dev/v4l-subdev2");
```

To:

```
strcat(video_name, "/dev/v4l-subdev3");
```

2. Can't get the image at 4K resolution

This is because 4K is the higher frequency, but the hardware signal quality is not good. Otherwise, rk3399 isp needs overlock, please refer to The method of isp overclock configuration

4.17.10 APK Starting Failure

The released patch will come with the rkCamera2 source code, please compile it on your system (can be directly put in the root directory, then mmm apk directory).
Apk crashing may be caused by:

1. Dependent Libraries Problem

   If you only install the apk to the system, then it will report:

   ```
   AndroidRuntime: java.lang.UnsatisfiedLinkError: dlopen failed: library
   "/system/lib64/libhdmiinput_jni.so" needed or dlopened by
   "/apex/com.android.runtime/lib64/libnativeloader.so" is not accessible for
   the namespace "classloader-namespace"
   ```

   It's suggested to compile and pack in the SDK, and the dependent libraries will pack it to img.

2. File Permission Problem

   APK accesses the RK628D device node by ioctl, for example:

   ```
   HdmiInput-navtive: openDevice(113): open /dev/v4l-subdev2
   failed,erro=Permission denied
   ```

   Confirm the file permission first. The patches is set to 666, since it accesses the device node in APK, you need to confirm whether you turn off selinux with getenforce.

3. Superstratum Configuration Problem

   If the version is after Android 9.0, configurate camera3_profiles.xml, if it is before Android 9.0, configure cam_board.xml. You can check by dumpsys media.camera, if not, it's probably not configured.

   ```
   $ dumpsys media.camera

   == Service global info: ==

   Number of camera devices: 1
   Number of normal camera devices: 1
   ```

4. Camera Library Crash

For example, if there are following errors, it may caused by camera name errors, because the name getDataFromXmlFile gets is different from the name driver gets, leading to CRASH.

```
09-29 04:43:30.673   405   405 W ServiceManagement: Waited one second for
android.hardware.camera.provider@2.4::ICameraProvider/legacy/0. Waiting
another...
09-29 04:43:31.673   405   405 W ServiceManagement: Waited one second for
android.hardware.camera.provider@2.4::ICameraProvider/legacy/0. Waiting
another...
...
09-29 04:43:31.904  1404  1404 F DEBUG   : backtrace:
09-29 04:43:31.904  1404  1404 F DEBUG   :     #00 pc 0007214e
/vendor/lib/hw/camera.rk30board.so
(android::camera2::ChromeCameraProfiles::handleAndroidStaticMetadata(char const*,
char const**)+546)
09-29 04:43:31.904  1404  1404 F DEBUG   :     #01 pc 00007895  /system/lib/vndk-
28/libexpat.so (doContent+432)
09-29 04:43:31.905  1404  1404 F DEBUG   :     #02 pc 0000637b  /system/lib/vndk-
28/libexpat.so (contentProcessor+40)
09-29 04:43:31.905  1404  1404 F DEBUG   :     #03 pc 00003825  /system/lib/vndk-
28/libexpat.so (XML_ParseBuffer+84)
09-29 04:43:31.905  1404  1404 F DEBUG   :     #04 pc 000711ad
/vendor/lib/hw/camera.rk30board.so
(android::camera2::CameraProfiles::getDataFromXmlFile()+148)
09-29 04:43:31.905  1404  1404 F DEBUG   :     #05 pc 00071e97
/vendor/lib/hw/camera.rk30board.so
(android::camera2::ChromeCameraProfiles::init()+86)
09-29 04:43:31.905  1404  1404 F DEBUG   :     #06 pc 000734c7
/vendor/lib/hw/camera.rk30board.so (android::camera2::PlatformData::init()+198)
09-29 04:43:31.905  1404  1404 F DEBUG   :     #07 pc 000cecb3
/vendor/lib/hw/camera.rk30board.so (initCameraHAL()+46)
...
```

For some platforms needed v4l2 (android9.0 and later), you need change name="rk628-csi" of camera3_profiles_rk3399.xml to name="rk628-csi-v4l2", please refer to the modification followed:

```
hardware/rockchip/camera
--- a/etc/camera/camera3_profiles_rk3399.xml
+++ b/etc/camera/camera3_profiles_rk3399.xml
@@ -769,7 +769,7 @@

 <!-- *****************PSL specific section end
************************************************************-->
     </Profiles>
-    <Profiles cameraId="0" name="rk628-csi" moduleId="m00">
+    <Profiles cameraId="0" name="rk628-csi-v4l2" moduleId="m00">
```

5. Crash caused by APK Permission

   The log related to this problem is like:

   For some platforms needed v4l2 (android9.0 and later), you need change the name of "rk628-csi" to "rk628-csi-v4l2", the modification is followed:

```
09-24 10:21:20.154  1552  1570 E AndroidRuntime: FATAL EXCEPTION: Thread-2
09-24 10:21:20.154  1552  1570 E AndroidRuntime: Process:
com.android.rockchip.camera2, PID: 1552
09-24 10:21:20.154  1552  1570 E AndroidRuntime: java.lang.IllegalStateException:
startRecording() called on an uninitialized AudioRecord.
09-24 10:21:20.154  1552  1570 E AndroidRuntime:        at
android.media.AudioRecord.startRecording(AudioRecord.java:983)
09-24 10:21:20.154  1552  1570 E AndroidRuntime:        at
com.android.rockchip.camera2.AudioStream$recordSound.run(AudioStream.java:199)
09-24 10:21:20.154  1552  1570 E AndroidRuntime:        at
java.lang.Thread.run(Thread.java:764)
09-24 10:21:20.164   464   820 W ActivityManager:   Force finishing activity
com.android.rockchip.camera2/.RockchipCamera2
09-24 10:21:20.183  1552  1552 D RockchipCamera2: onPause
```

You can use the following patches to resolve the problem.

```
packages/apps/rkCamera2
--- a/AndroidManifest.xml
+++ b/AndroidManifest.xml
@@ -4,7 +4,8 @@
     package="com.android.rockchip.camera2">

         <uses-permission android:name="android.permission.CAMERA" />
-        <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
+       <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
+       <uses-permission android:name="android.permission.RECORD_AUDIO" />
     <application
         android:allowBackup="true"
         android:icon="@mipmap/ic_launcher"
diff --git a/src/com/android/rockchip/camera2/RockchipCamera2.java
b/src/com/android/rockchip/camera2/RockchipCamera2.java
index 9dc29b8..02ee104 100755
--- a/src/com/android/rockchip/camera2/RockchipCamera2.java
+++ b/src/com/android/rockchip/camera2/RockchipCamera2.java
@@ -100,9 +100,11 @@ public class RockchipCamera2 extends Activity {
        if (ActivityCompat.checkSelfPermission(this,
            Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED
            && ActivityCompat.checkSelfPermission(this,
-                   Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
+                   Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED
+           && ActivityCompat.checkSelfPermission(this,
+                   Manifest.permission.RECORD_AUDIO) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(RockchipCamera2.this,
-               new String[] { Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE },
+               new String[] { Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.RECORD_AUDIO},
                    REQUEST_CAMERA_PERMISSION);
            return;
        }
```

4.17.11 APK Preview Failure when DTS Connecting with rkcif

RK356X android11 code version is R10/R11, dts configurates RK628 to connect to the rkcif, if capture image is normal, but apk preview fails to open, maybe because parts of the cameraHAL of R10 and R11 codes do not support the pipeline link, and R9 codes do not have this problem. Troubleshooting and solving methods are as follows:

1. Turn on cameraHAL switch:

```
setprop persist.vendor.camera.hal.debug 5
```

2. Grab the logcat for the error when openning the apk, and the key log information is as follows:

```
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:rockchip-csi2-dphy0, srcPad:1, sinkName:none, sinkPad:0, enable:1, flags:1
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:none, srcPad:1, sinkName:none, sinkPad:0, enable:1, flags:1
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:none, srcPad:2, sinkName:rkisp_rawwr0, sinkPad:0, enable:1, flags:1
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:none, srcPad:4, sinkName:rkisp_rawwr2, sinkPad:0, enable:1, flags:1
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:none, srcPad:5, sinkName:rkisp_rawwr3, sinkPad:0, enable:1, flags:1
I RkCamera: <HAL> RKISP2GraphConfig: @addFormatParams, entityName:none, width:1920, height:1080, pad:0, format:0x2006:V4L2_MBUS_FMT_UYVY8_2X8, field:0
I RkCamera: <HAL> RKISP2GraphConfig: @addSelectionParams, width:1920, height:1080, left:0, top:0, target:0, pad:0, entityName:none
I RkCamera: <HAL> RKISP2GraphConfig: @addSelectionParams, width:1920, height:1080, left:0, top:0, target:0, pad:2, entityName:none
I RkCamera: <HAL> RKISP2GraphConfig: @addFormatParams, entityName:none, width:1920, height:1080, pad:2, format:0x2008:V4L2_MBUS_FMT_YUYV8_2X8, field:0
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:none, srcPad:3, sinkName:none, sinkPad:0, enable:1, flags:1
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:none, srcPad:0, sinkName:none, sinkPad:1, enable:1, flags:1
D RkCamera: <HAL> RKISP2GraphConfig: @ isNeedPathCrop : stream ratios: 1.777778
D RkCamera: <HAL> RKISP2GraphConfig: @ isNeedPathCrop : mp_need_crop 1, sp_need_crop 0, sp_enabled 0
D RkCamera: <HAL> RKISP2GraphConfig: @cal_crop : src_ratio:1.777778, dst_ratio:1.777778, src(1920x1080), dst(1920x1080)
I RkCamera: <HAL> RKISP2GraphConfig: @addSelectionVideoParams, width:1920, height:1080, left:0, top:0, target:0, type:1, flags:0 entityName:none
I RkCamera: <HAL> RKISP2GraphConfig: @addFormatParams, entityName:none, width:1920, height:1080, pad:0, format:0x3231564e:V4L2_PIX_FMT_NV12, field:0
I RkCamera: <HAL> RKISP2GraphConfig: @addLinkParams, srcName:none, srcPad:2, sinkName:none, sinkPad:0, enable:1, flags:1
I RkCamera: <HAL> RKISP2GraphConfig: @getImguMediaCtlConfig : No need for selfPath
```

If above log appears, camereaHAL configurates the pipeline to the rkisp by default, and the configuration of dts is corresponding to rkcif.

3. Solution

   Under hardware/rockchip/camera directory, add the following modification:

```
diff --git a/psl/rkisp2/RKISP2GraphConfig.cpp b/psl/rkisp2/RKISP2GraphConfig.cpp
index 2a5fa5a..3d2409c 100755
--- a/psl/rkisp2/RKISP2GraphConfig.cpp
+++ b/psl/rkisp2/RKISP2GraphConfig.cpp
@@ -76,6 +76,7 @@ const string MEDIACTL_POSTVIEWNAME = "postview";

 const string MEDIACTL_STATNAME = "rkisp1-statistics";
 const string MEDIACTL_VIDEONAME_CIF = "stream_cif_dvp_id0";
+const string MEDIACTL_VIDEONAME_CIF_MIPI_ID0 = "stream_cif_mipi_id0";

 RKISP2GraphConfig::RKISP2GraphConfig() :
         mManager(nullptr),
@@ -2620,6 +2621,13 @@ status_t RKISP2GraphConfig::getImguMediaCtlConfig(int32_t
cameraId,
                 addLinkParams("rkisp-isp-subdev", 2, "rkisp_mainpath", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
                 addLinkParams("rkisp-isp-subdev", 2, "rkisp_selfpath", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
             }
+       } else if(mipName2.find("mipi") != std::string::npos) {
+            addLinkParams(mipName, mipSrcPad, mipName2, csiSinkPad, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            addLinkParams(mipName2, 1, "stream_cif_mipi_id0", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            addLinkParams(mipName2, 2, "stream_cif_mipi_id1", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            addLinkParams(mipName2, 3, "stream_cif_mipi_id2", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
```

```
+            addLinkParams(mipName2, 4, "stream_cif_mipi_id3", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
+            mSensorLinkedToCIF = true;
         } else {
             addLinkParams(mipName, mipSrcPad, csiName, csiSinkPad, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
             addLinkParams(csiName, csiSrcPad, IspName, ispSinkPad, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
@@ -2628,6 +2636,12 @@ status_t RKISP2GraphConfig::getImguMediaCtlConfig(int32_t
cameraId,
             addLinkParams(csiName, 5, "rkisp_rawwr3", 0, 1,
MEDIA_LNK_FL_ENABLED, mediaCtlConfig);
         }
     }
+    if(mSensorLinkedToCIF){
+       addImguVideoNode(IMGU_NODE_VIDEO, MEDIACTL_VIDEONAME_CIF_MIPI_ID0,
mediaCtlConfig);
+       addFormatParams(MEDIACTL_VIDEONAME_CIF_MIPI_ID0, mCurSensorFormat.width,
mCurSensorFormat.height,
+               0, V4L2_PIX_FMT_NV12, 0, 0, mediaCtlConfig);
+        return OK;
+    }
     // isp input pad format and selection config
     addFormatParams(IspName, ispInWidth, ispInHeight, ispSinkPad, ispInFormat,
0, 0, mediaCtlConfig);
     addSelectionParams(IspName, ispInWidth, ispInHeight, 0, 0,
V4L2_SEL_TGT_CROP, ispSinkPad, mediaCtlConfig);
```

4.17.12 How to operate the GPIO of RK628

The GPIO interface can be directly called at the location where the GPIO needs to be controlled, which includes IOMUX settings, GPIO direction settings, and level settings. For example, the following operation sets the I2C_SDA_HDMI and I2C_SCL_HDMI interfaces to GPIO.

```
rk628_gpio_direction_output(rk628, GPIO1_B1, GPIO_REG_HIGH);
rk628_gpio_direction_output(rk628, GPIO1_B2, GPIO_REG_HIGH);
```

To recover, the following interfaces need to be called:

```
rk628_pinctrl_set_mux(rk628, GPIO1_B1, DDCM0SDARX);
rk628_pinctrl_set_mux(rk628, GPIO1_B2, DDCM0SCLRX);
```

4.17.13 Sound Card Register Failure

As follows, can't find the hdmiin sound card after system turns on.

```
# cat /proc/asound/cards
 1 [rockchiphdmiin ]: rockchip_hdmiin - rockchip,hdmiin
                      rockchip,hdmiin
```

Then you should firstly check whether soc dai, codec dai devices which the sound card depends on load successfully :

```
rk3399:/ # cat /sys/kernel/debug/asoc/components
...
dummy-codec
ff8a0000.i2s
ff8a0000.i2s
ff880000.i2s
ff880000.i2s
...
rk3399:/ # cat /sys/kernel/debug/asoc/dais
...
ff8a0000.i2s
ff880000.i2s
dummy-codec
....
```

If the nodes above fail to load, you can check defconfig, dts.

In the case of soc dai, codec dai are registered successfully, the sound card cannot be registered, which may be due to insufficient dma resources, especially in the case of rk3399. In kernel 4.19 and above, the following attributes can be added to the i2s node to solve the problem:

```
--- a/arch/arm64/boot/dts/rockchip/rk3399.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399.dtsi
@@ -1788,6 +1788,7 @@
                clocks = <&cru SCLK_I2S2_8CH>, <&cru HCLK_I2S2_8CH>;
                resets = <&cru SRST_I2S2_8CH>, <&cru SRST_H_I2S2_8CH>;
                reset-names = "reset-m", "reset-h";
+               rockchip,capture-only;
                power-domains = <&power RK3399_PD_SDIOAUDIO>;
                status = "disabled";
```

5. Common requirements

5.1 The configuration of RK628 24M Crystal oscillator which comes from other SOC

In the following, take several ROCKCHIP master controllers for example to describe the configuration methods. Among them, the CLK configuration of DTS is enabled by default in the code. If it is not enabled, 24M CLK will be turned off after startup, resulting in I2C communication exception.

5.1.1 Add 24M support to RK3399

1. Cite the configuration followed in rk628 dts

```
pinctrl-names = "default";
pinctrl-0 = <&rk628_rst>, <&clk_testout2>;
assigned-clocks = <&cru SCLK_TESTCLKOUT2>;
assigned-clock-rates = <24000000>;
clocks = <&cru SCLK_TESTCLKOUT2>;
clock-names = "soc_24M";
```

2. Add PINCTRL of IO pin according to actual hardware connection

```
&pinctrl {
        test {
                clk_testout2: clk_testout2 {
                rockchip,pins = <0 8 RK_FUNC_3 &pcfg_pull_none>;
        };
};
```

5.1.2 Add 24M support to RK3288

1. Cite the configuration followed in rk628 dts

```
pinctrl-names = "default";
pinctrl-0 = <&test_clkout>;
assigned-clocks = <&cru SCLK_TESTOUT_SRC>;
assigned-clock-parents = <&xin24m>;
clocks = <&cru SCLK_TESTOUT>;
clock-names = "soc_24M";
```

2. Add PINCTRL of IO pin according to actual hardware connection

```
&pinctrl {
        test {
                test_clkout: test-clkout {
                        rockchip,pins = <0 17 RK_FUNC_1 &pcfg_pull_none>;
                };
        };
};
```

5.1.3 Add 24M support to RK356X

RK356X has many pins which can output 24MHZ , such as REF_CLKOUT (clk_wifi/gpio0_a0), CAM_CLKOUT1 (clk_cam1_out/gpio4_b0) and ETH_REFCLK_25M_M0 (clk_mac1_out/gpio3_b0), now we take clk_wifi for example.

1. Cite the configuration followed in rk628 dts

```
&i2c2_rk628 {
    pinctrl-names = "default";
    pinctrl-0 = <&rk628_reset &refclk_pins>;
    assigned-clocks = <&pmucru CLK_WIFI>;
    assigned-clock-rates = <24000000>;
    clocks = <&pmucru CLK_WIFI>;
```

```
    clock-names = "soc_24M";
};

&pinctrl {
    rk628 {
        rk628_reset: rk628-reset {
            rockchip,pins = <2 RK_PA2 RK_FUNC_GPIO &pcfg_pull_none>;
        };
    };
};
```

2. The configuration followed is came with in the rk3568-pinctrl.dtsi

```
&pinctrl {
    ......
    refclk {
        /omit-if-no-ref/
        refclk_pins: refclk-pins {
            rockchip,pins =
                /* refclk_ou */
                <0 RK_PA0 1 &pcfg_pull_none>;
        };
    };
    ......
};
```

Other chips not mentioned, can be configurated according to "Rockchip_Develop_Guide_Gpio_Output_Clocks_CN.pdf".

5.2 Dual-RK628 Support

5.2.1 HDMI2CSI+HDMI2CSI support

5.2.1.1 Notes:

1. Dual-RK628 is similar to dual camera. The dual RK628 can work at the same time, and the master controller needs to support two isp controllers and to deal with two input datas at the same time, such as RK3399.
2. In the hardware design, two RK628s should be under different I2Cs or one I2C designed into different I2C addresses.
3. Note that the dts configuration should be consistent with the hardware design. rk628 hardware is connected to mipi_phy_rx0 or mipi_phy_tx1rx1, and the corresponding configuration is also required in the dts.
4. Realize HDMI2CSI by dual-rk628, it's suggested RESET and INT control io of dual rk628 cannot be the same, otherwise it will be abnormal.

5.2.1.2 kernel dts configuration problems

Now take RK3399 configurating dual RK628 for example.

1. rk628->mipi_dphy_rx0->rkisp1_0, the first channel index is 0, facing is configurated as back

```
&i2c1 {
    status = "okay";

    rk628_csi_v4l2: rk628_csi_v4l2@50 {
        compatible = "rockchip,rk628-csi-v4l2";
        reg = <0x50>;
        //clocks = <&ext_cam_clk>;
        //clock-names = "xvclk";

        interrupt-parent = <&gpio4>;
        interrupts = <16 IRQ_TYPE_LEVEL_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_irq>;
        //power-gpios = <&gpio0 RK_PD5 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio4 RK_PD2 GPIO_ACTIVE_LOW>;
        plugin-det-gpios = <&gpio0 RK_PD6 GPIO_ACTIVE_LOW>;
        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "RK628-CSI";
        rockchip,camera-module-lens-name = "NC";
        port {
            rk628_out: endpoint {
                remote-endpoint = <&hdmi2mipi_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&mipi_dphy_rx0 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            hdmi2mipi_in: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&rk628_out>;
                data-lanes = <1 2 3 4>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            dphy_rx0_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&isp0_mipi_in>;
            };
        };
```

```
    };
};

&rkisp1_0 {
    status = "okay";

    port {
        #address-cells = <1>;
        #size-cells = <0>;

        isp0_mipi_in: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&dphy_rx0_out>;
        };
    };
};
```

2. rk628->mipi_dphy_tx1rx1->rkisp1_1, the second channel index is 1. facing is configurated as front

```
&i2c4 {
    status = "okay";

    rk628_csi_v4l2_1: rk628_csi_v4l2_1@50 {
        compatible = "rockchip,rk628-csi-v4l2";
        reg = <0x50>;
        //clocks = <&ext_cam_clk>;
        //clock-names = "xvclk";

        interrupt-parent = <&gpio3>;
        interrupts = <12 IRQ_TYPE_LEVEL_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_irq1>;
        //power-gpios = <&gpio0 RK_PD5 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio4 RK_PD3 GPIO_ACTIVE_LOW>;
        plugin-det-gpios = <&gpio0 RK_PD7 GPIO_ACTIVE_LOW>;
        rockchip,camera-module-index = <1>;
        rockchip,camera-module-facing = "front";
        rockchip,camera-module-name = "RK628-CSI";
        rockchip,camera-module-lens-name = "NC";
        port {
            rk628_out1: endpoint {
                remote-endpoint = <&hdmi2mipi_in1>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&mipi_dphy_tx1rx1 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
```

```
            #address-cells = <1>;
            #size-cells = <0>;

            hdmi2mipi_in1: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&rk628_out1>;
                data-lanes = <1 2 3 4>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            dphy_tx1rx1_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&isp1_mipi_in>;
            };
        };
    };
};

&rkisp1_1 {
    status = "okay";

    port {
        #address-cells = <1>;
        #size-cells = <0>;

        isp1_mipi_in: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&dphy_tx1rx1_out>;
        };
    };
};
```

5.2.1.3 android configuration questions

1. camera3_profiles.xml configuration notes:

moduleId: need be the same to the index in the dts

first channel of rk628:

```
</Profiles>
<Profiles cameraId="0" name="rk628-csi" moduleId="m00">
    <Supported_hardware>
        <hwType value="SUPPORTED_HW_RKISP1"/>
    </Supported_hardware>
```

second channel of rk628:

```
</Profiles>
<Profiles cameraId="1" name="rk628-csi" moduleId="m01">
    <Supported_hardware>
        <hwType value="SUPPORTED_HW_RKISP1"/>
    </Supported_hardware>
```

1. If you need to use two channels of rk628 in the same time, add the support to cameraHAL. You can add the modification in the directory of SDK/hardware/rockchip/camera$.

```
diff --git a/common/platformdata/PlatformData.cpp
b/common/platformdata/PlatformData.cpp
index 36d2ac9..d3fcb4b 100755
--- a/common/platformdata/PlatformData.cpp
+++ b/common/platformdata/PlatformData.cpp
@@ -1047,7 +1047,7 @@ CameraHWInfo::CameraHWInfo() :
     mProductName = "<not_set>";
     mManufacturerName = "<not set>";
     mCameraDeviceAPIVersion = CAMERA_DEVICE_API_VERSION_3_3;
-    mSupportDualVideo = false;
+    mSupportDualVideo = true;
     mSupportExtendedMakernote = false;
     mSupportFullColorRange = true;
     mSupportIPUAcceleration = false;
```

5.2.2 HDMI2CSI+HDMI2DSI support

5.2.2.1 kernel dts configuration

HDMI2CSI+HDMI2DSI: one of the RK628 channels is used by camera frame, which is connected to the master controller, and the other one is connected to the screen. Take RK3399 for example:

1. HDMI2CSI configuration

rk628->mipi_phy_rx0->rkisp1_0 or rk628->mipi_dphy_tx1rx1->rkisp1_1, take rkisp1_0 for example:

```
&i2c1 {
    status = "okay";

    rk628_csi_v4l2: rk628_csi_v4l2@50 {
        compatible = "rockchip,rk628-csi-v4l2";
        reg = <0x50>;
        //clocks = <&ext_cam_clk>;
        //clock-names = "xvclk";

        interrupt-parent = <&gpio4>;
        interrupts = <16 IRQ_TYPE_LEVEL_LOW>;
        pinctrl-names = "default";
        pinctrl-0 = <&rk628_irq>;
        //power-gpios = <&gpio0 RK_PD5 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio4 RK_PD2 GPIO_ACTIVE_LOW>;
        plugin-det-gpios = <&gpio0 RK_PD6 GPIO_ACTIVE_LOW>;
        rockchip,camera-module-index = <0>;
        rockchip,camera-module-facing = "back";
        rockchip,camera-module-name = "RK628-CSI";
        rockchip,camera-module-lens-name = "NC";
```

```
        port {
            rk628_out: endpoint {
                remote-endpoint = <&hdmi2mipi_in>;
                data-lanes = <1 2 3 4>;
            };
        };
    };
};

&mipi_dphy_rx0 {
    status = "okay";

    ports {
        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            hdmi2mipi_in: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&rk628_out>;
                data-lanes = <1 2 3 4>;
            };
        };

        port@1 {
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;

            dphy_rx0_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&isp0_mipi_in>;
            };
        };
    };
};

&rkisp1_0 {
    status = "okay";

    port {
        #address-cells = <1>;
        #size-cells = <0>;

        isp0_mipi_in: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&dphy_rx0_out>;
        };
    };
};
```

2. HDMI2DSI configuration

Please refer to HDMI2DSI Switching in the section 2.4.5.

5.2.2.2 android configuration

1. HDMI2CSI: please refer to one of rk628 HDMI2CSI channel, and configure camera3_profiles.xml correctly.
2. HDMI2DSI: the part of android needn't be configured.