

基于CameraHal3的HDMI IN开发指南

前言

本文档是基于 RK3288/RK3399/RK3566/RK3568 ANDROID 9/10/11等平台开发 HDMI IN功能的帮助文档。

概述

本文档主要以RK628D转换芯片基于Android 9.0为例，介绍如何使用RK AP与RK628D芯片组合实现HDMI IN功能，支持HDMI IN热拔插，支持HDMI IN输入自适应分辨率：4KP30、1080P60、720P60、576P50、480P60等。

产品版本

芯片名称	Kernel版本	Android版本
RK3288/RK3326/RK3368/RK3399/RK3566/RK3568	Linux 4.4/ Linux 4.19	Android 9/10/11

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	温定贤	2021-05-08	初始版本发布
V1.1.0	温定贤	2021-06-02	1、增加RK628寄存器读写方法说明； 2、增加配置ISP使用CMA内存的方法说明；

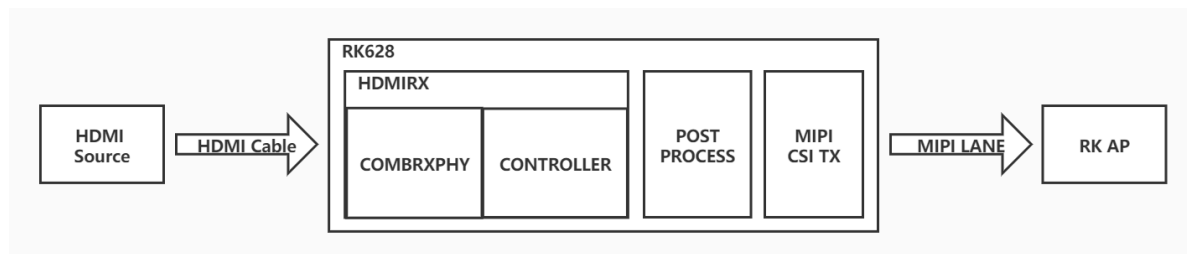
目录

基于CameraHal3的HDMI IN开发指南

1. HDMI IN相关驱动代码说明
 - 1.1 基于RK628D实现HDMI IN功能
 - 1.2 基于其他转换芯片实现HDMI IN功能
2. HDMI IN VIDEO框架说明
 - 2.1 HDMI IN APK工作流程
 - 2.2 RK628D驱动架构
3. 配置方法说明
 - 3.1 SDK代码版本要求
 - 3.2 板级配置说明
 - 3.2.1 功能模块配置
 - 3.2.2 硬件连接配置
 - 3.2.3 图像接收链路dts配置
 - 3.3 camera3_profiles.xml配置文件说明
 - 3.4 不同芯片平台的接收能力
 - 3.4.1 RK3399配置isp超频的方法
 - 3.4.2 配置ISP使用CMA内存的方法
 - 3.5 EDID的配置方法
4. HDMI IN APK适配方法
 - 4.1 获取和编译APK源码
 - 4.2 APK源码的适配
 - 4.3 APK调试前的准备
5. 驱动调试方法
 - 5.1 调试工具获取
 - 5.2 调试命令举例
6. 常见问题排查方法
 - 6.1 打开log开关
 - 6.2 寄存器读写
 - 6.3 MFD设备报错排除
 - 6.4 Clk det 异常问题
 - 6.5 HDMI RX正常的判断方法
 - 6.6 Open subdev 权限异常

1. HDMI IN相关驱动代码说明

1.1 基于RK628D实现HDMI IN功能



RK628D作为MFD设备，包含MFD设备驱动和各个接口功能模块驱动。RK628D HDMI RX To MIPI CSI TX用于HDMI IN应用场景，将RK628D作为类camera设备使用，基于V4L2框架实现相关驱动。由于与显示DRM框架不同，除COMBRXPHY、COMBTXPHY驱动复用外，HDMI RX Controller、Post Process和MIPI CSI TX等功能模块均在V4L2驱动框架中重新实现。

相关驱动代码：

```
drivers/mfd/rk628.c

drivers/gpu/drm/rockchip/rk628/rk628_combrxphy.c

drivers/gpu/drm/rockchip/rk628/rk628_combtxphy.c

drivers/media/i2c/rk628_csi.c
```

Kernel Config配置：

```
CONFIG_MFD_RK628=y

CONFIG_VIDEO_RK628_CSI=y

CONFIG_DRM_ROCKCHIP_RK628=y
```

1.2 基于其他转换芯片实现HDMI IN功能

其他转换芯片则未使用MFD设备驱动，仅在V4L2框架基础上实现I2C设备驱动。例如：TC358743/TC358749/LT6911UXC等：

相关驱动代码：

```
drivers/media/i2c/tc35874x.c

drivers/media/i2c/lt6911uxc.c
```

Kernel Config配置：

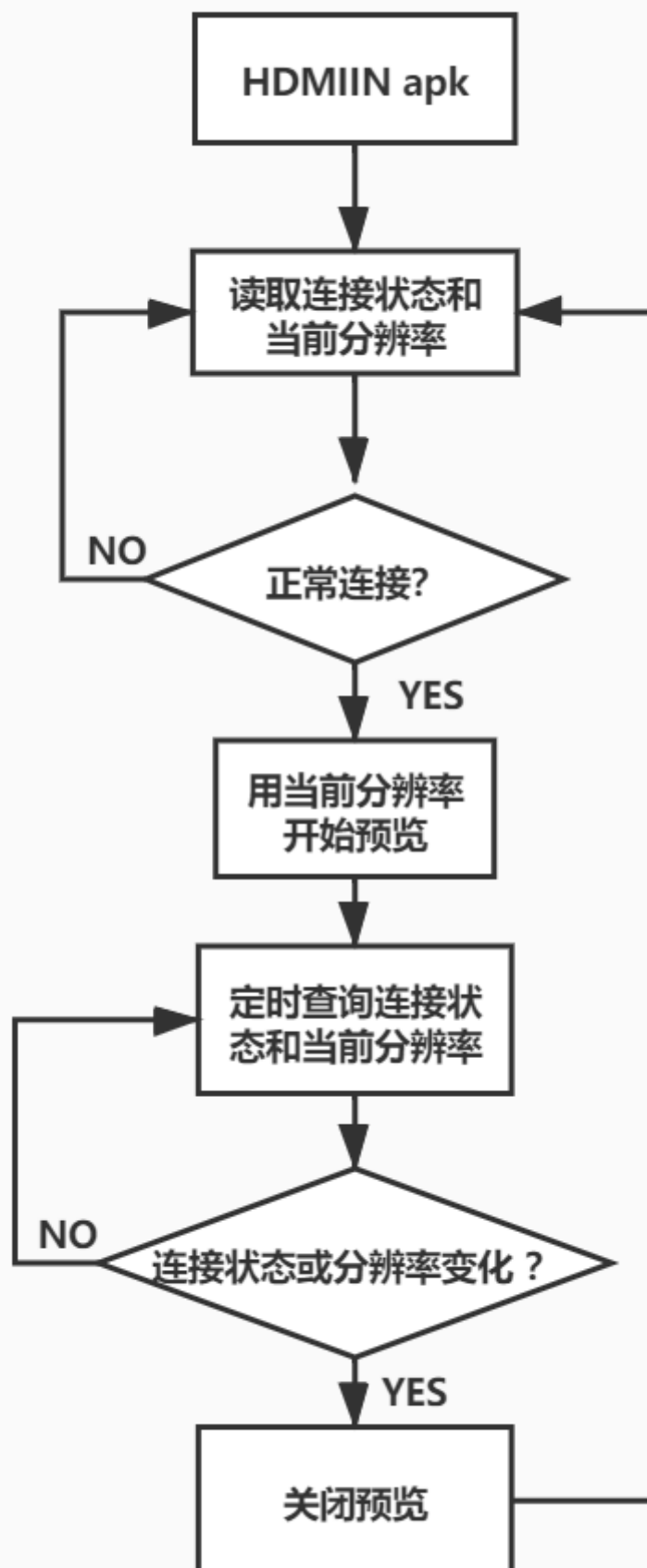
```
CONFIG_VIDEO_TC35874X=y
```

```
CONFIG_VIDEO_LT6911UXC=y
```

2. HDMI IN VIDEO框架说明

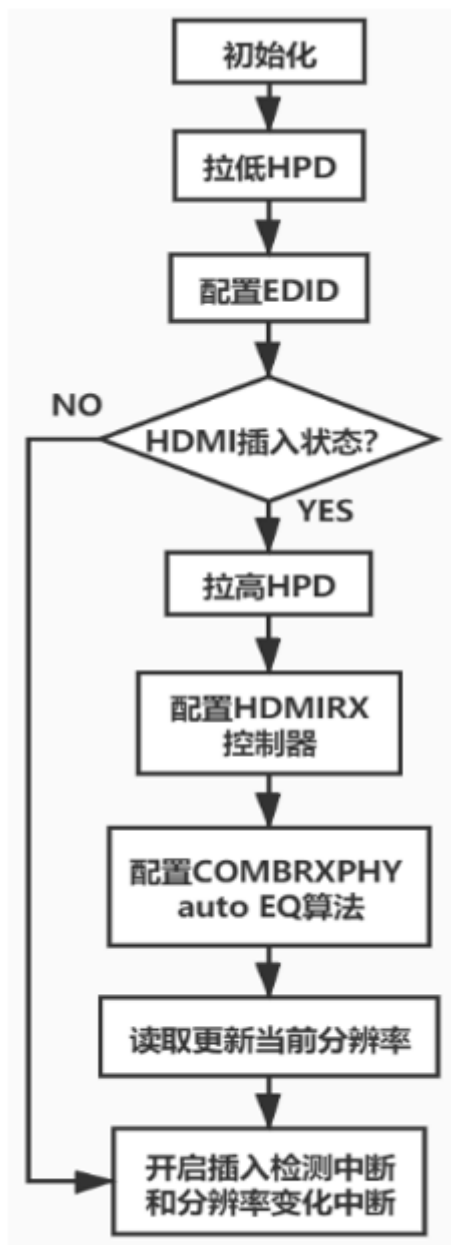
HDMI IN video部分的软件实现方案是将RK628D模拟成一个MIPI SOC camera设备，通过camera框架接收video数据并在APK进行显示，同时基于HDMI IN的应用场景需要，增加HDMI IN热拔插和HDMI IN分辨率自适应支持。

2.1 HDMI IN APK工作流程

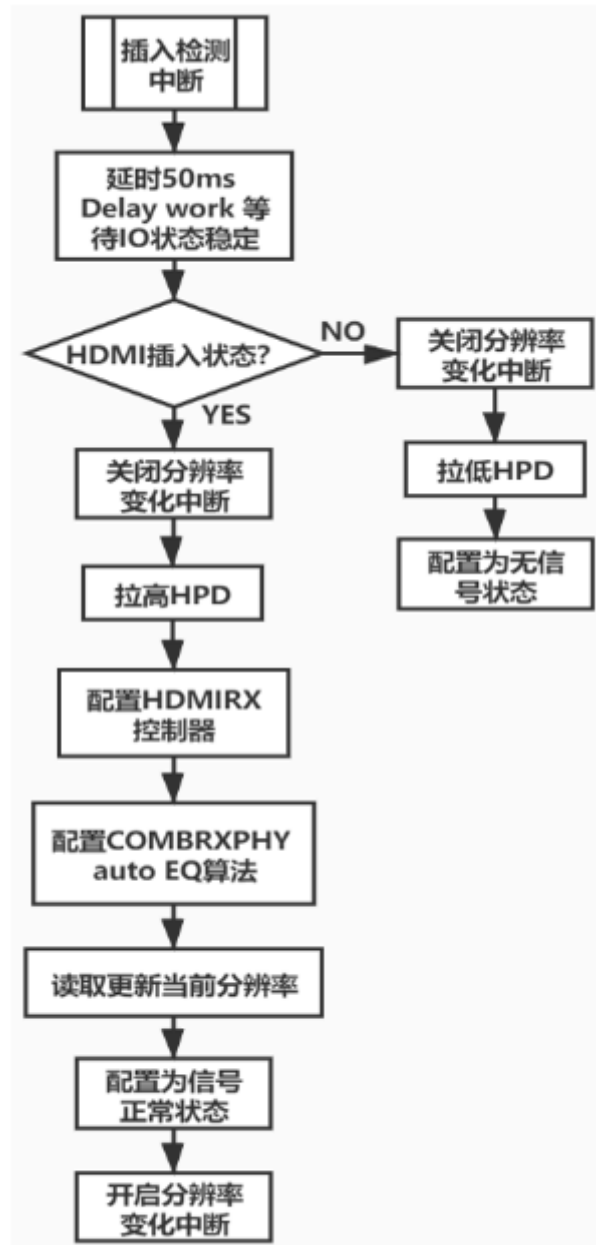


2.2 RK628D驱动架构

RK628D驱动需要重点关注的主要有三个部分：初始化、热拔插中断处理、分辨率切换中断处理。流程图参考如下：



驱动初始配置流程



热拔插中断处理程序





分辨率变化中断处理程序

3. 配置方法说明

支持HDMI IN功能，需要将SDK升级到指定的版本以上，同时板级配置需要根据实际硬件电路连接进行修改。

3.1 SDK代码版本要求

（注意：commit id可能会有差异）

- kernel/ 代码需要包含以下提交：

```
git log drivers/gpu/drm/rockchip/rk628/
```

```
wendingxian@ubuntu:~/rk3288_9.0_int/kernel$ git log drivers/gpu/drm/rockchip/rk628/
commit 2420e34afd4dbd0045e13222165b159bb62df151
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Wed Apr 21 10:19:19 2021 +0800

    drm: rockchip: rk628: the combrxphy cable mode wait for clk to stabilize

    Signed-off-by: Dingxian Wen <shawn.wen@rock-chips.com>
    Change-Id: I7502b8f71f4fcc376a925045adbe6148ebdc2d3f

commit 403cf034772cd268eb6554f577d98462c007a90f
Author: Shunqing Chen <csq@rock-chips.com>
Date:   Wed Apr 7 09:10:10 2021 +0800

    drm: rockchip: rk628: post_process: recalculate dst clock

    Signed-off-by: Shunqing Chen <csq@rock-chips.com>
    Change-Id: I93388ba499f0d74c5f5c549decc83f3225ae1b82

commit 71d16a0f377366d398af6b4f2ea7f53b4fbaf91b
Author: Guochun Huang <hero.huang@rock-chips.com>
Date:   Thu Jan 7 16:15:30 2021 +0800

    drm/rockchip/rk628: fix no connector found when find_connector_by_bridge

    Change-Id: Id530565a1165957ffc1cb4604fe9a14fe7537ed5
    Signed-off-by: Guochun Huang <hero.huang@rock-chips.com>

commit 744285a44b5bdf864cad77dd18450c8ec0a1129e
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Mon Dec 14 15:25:19 2020 +0800

    drm: rockchip: rk628: add rk628 combrx-phy support for HDMIRX cable mode

    Signed-off-by: Dingxian Wen <shawn.wen@rock-chips.com>
    Change-Id: I4c02fb6add7cce5fef8a52853c35f113b53040ff
```

```
git log drivers/media/i2c/rk628_csi.c
```

```
wendingxian@ubuntu:~/rk3288_9.0_int/kernel$ git log drivers/media/i2c/rk628_csi.c
commit 57edf24f52778f6eafad004bc70bd1f71525a28c (HEAD -> develop-4.4)
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Wed Apr 28 16:25:13 2021 +0800

    media: i2c: rk628csi: upgrade the driver version to v0.0.3

    Modify RK628_CSI_PIXEL_RATE_LOW to 400M.

    Signed-off-by: Dingxian Wen <shawn.wen@rock-chips.com>
    Change-Id: I024a0c24f7e61e9314e9e6f24c78f24ec403e4e4

commit ed3d2ca11c2e0635b423d24977cdab6894e90400
Author: Shunhua Lan <lsh@rock-chips.com>
Date:   Sun Apr 25 18:48:28 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

commit 2ac8d31dbbcfa7cd0b43d20834cd31341c6da7be
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Wed Apr 21 15:14:20 2021 +0800

    media: i2c: rk628csi: upgrade the driver version to v0.0.2

    1.The hpd effective level can be configured to be inverted, when the hpd
    output circuit is reversed.
```

- hardware/rockchip/camera/ 代码需要包含以下提交:

```
commit 4bd9632ed59261e8fc43057b7545cedcde2d927a
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Tue Apr 27 21:16:59 2021 +0800

    Add rk628 configuration for each platform

    include:
    camera3_profiles_rk3288.xml
    camera3_profiles_rk3326.xml
    camera3_profiles_rk3368.xml
    camera3_profiles_rk3399.xml
    camera3_profiles_rk3399pro.xml
    camera3_profiles_rk356x.xml

    Signed-off-by: Dingxian Wen <shawn.wen@rock-chips.com>
    Change-Id: I8d13feca46739df37f7f01e3a02014106ee8e48c
```

- device/rockchip/common/ 代码需要包含以下提交:

```
commit 17112e1430b0f10a88b57c73ad19203e58f0eeff (HEAD -> mid_9.0, rk/rk)
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
```

- RKDocs/common/ 代码需要包含以下提交：

```
commit b1a130c7759b7f8588931acc82d5d9b88b38af45 (HEAD -> rk3
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Fri May 7 17:27:12 2021 +0800

    add hdmi-in apk based on camerahal3

Signed-off-by: Dingxian Wen <shawn.wen@rock-chips.com>
Change-Id: I0dc677d4e0f4bb39fc693bdea73a2eb8a4087935
```

3.2 板级配置说明

SDK代码中基于RK3288 RK628D EVB提交了一版dts，可参考：

```
arch/arm/boot/dts/rk3288-evb-rk628-hdmi2csi-avb.dts
```

3.2.1 功能模块配置

RK628D实现HDMI IN功能需要使用COMBRXPHY、COMBTXPHY、CSI功能模块，因此需要在dts中分别使能。注意其他未提及的模块，参照rk628.dtsi的默认配置即可，不需要使能，特别是：

rk628_post_process、rk628_hdmi、rk628_hdmirx，是用于RK628D点屏等显示通路时使用，在HDMI IN场景中不需要使能。

```
&rk628_combrxphy {
    status = "okay";
};

&rk628_combtxphy {
    status = "okay";
};

&rk628_csi {
    status = "okay";
    ...
}
```

3.2.2 硬件连接配置

RK628D是I2C设备，需要配置在对应的I2C总线下，主要硬件连接配置说明如下：

```

&i2c1 {
    clock-frequency = <400000>;
    status = "okay";

    rk628: rk628@50 {
        reg = <0x50>;
        interrupt-parent = <&gpio7>;
        interrupts = <15 IRQ_TYPE_LEVEL_HIGH>;
        enable-gpios = <&gpio5 RK_PC2 GPIO_ACTIVE_HIGH>;
        reset-gpios = <&gpio7 RK_PB6 GPIO_ACTIVE_LOW>;
        status = "okay";
    };
};

```

- reg: I2C地址；RK628D典型的7bit I2C地址为0x50，使用多片RK628D时，可通过RK628D的GPIO改变I2C地址，参考如下：

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

- interrupt-parent/ interrupts: 连接RK628D中断的GPIO引脚；
- enable-gpios: RK628D供电控制GPIO引脚（若为常供电可不配置）；
- reset-gpios: RK628D复位控制GPIO引脚；

RK628_CSI功能模块也涉及一些硬件连接配置，参考如下：

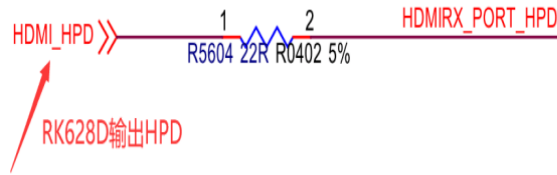
```

&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>;
    ...
}

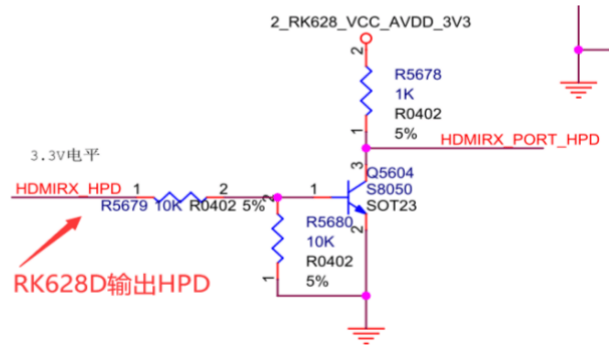
```

- hpd-output-inverted: HPD输出取反配置，若HPD输出电平在电路做了取反，则需要使能此配置项；

◆ HPD未取反设计：

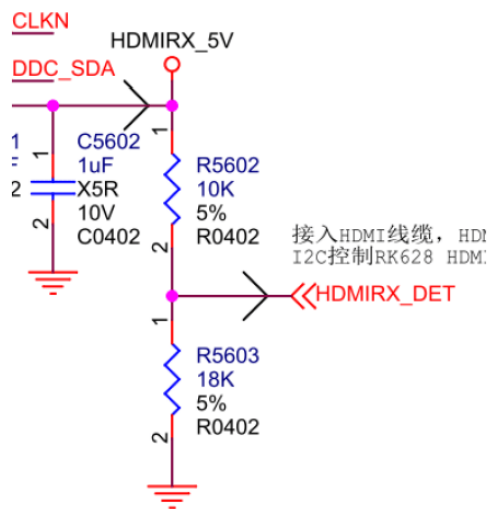


◆ HPD取反设计：

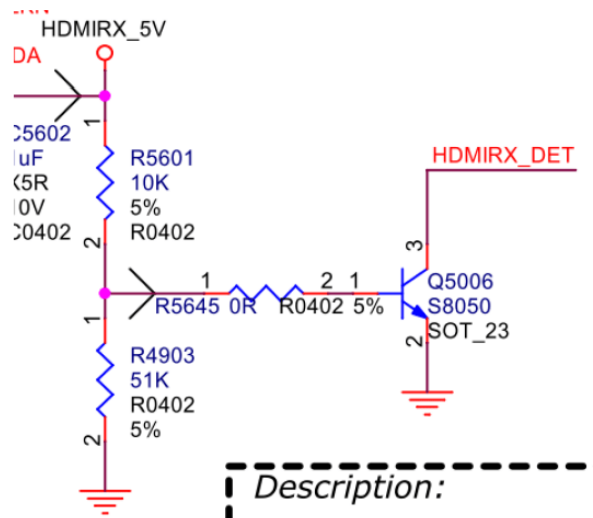


- plugin-det-gpios: HDMI插入检测GPIO引脚，注意电路是否有对电平取反，有效电平需要配置正确；

◆ HDMIRX_DET未取反设计：



◆ HDMIRX_DET取反设计：



- power-gpios: RK主控端MIPI RX电源域供电控制GPIO引脚（若为常供电可不配置）；

3.2.3 图像接收链路dts配置

将RK628D/TC35874x/LT6911等转换芯片作为类camera设备进行开发，与camera sensor一样需要实现基于V4L2框架的驱动，数据链路配置的方法与MIPI SOC Sensor一致。

- 以RK3288为例，rk628 + isp1链路配置：

```
&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";
};
```

- 以RK356x为例，rk628 + isp2链路配置：

```
&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>;
        };
    };
};

&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 = <&hdmiin_out0>;
                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>;
        };
    };
};
};

```

- 以RK356x为例，rk628 + vicap链路配置：

```

&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>;
        };
    };
};

&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 = <&hdmi_in_out0>;
                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";
};

```

3.3 camera3_profiles.xml配置文件说明

camera3_profiles.xml文件对应SDK目录下具体芯片平台的文件:

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

主要配置注意事项如下，详情可参考SOC Sensor的配置方法:

- name: 需要与驱动名称一致，有大小写区别;
- moduleId: 需要与驱动dts中配置的index一致;

```

</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:
需要正确配置驱动支持的分辨率和最小的帧间隔时间，若驱动中需要增加新的分辨率支持，在这里也要相应地增加配置；

```

<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：图像旋转角度，支持0、90、180、270；

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

3.4 不同芯片平台的接收能力

由于各个芯片平台isp/vicap的性能不同，对图像的最大接收能力也不同。可参考下表：

芯片平台	接收控制器	支持最大分辨率
RK3288/RK3326/RK3368	isp	1920x1080P60
RK3399	isp	3840x2160P30（注：isp需要超频）
RK3566/RK3568	vicap/isp	3840x2160P30

3.4.1 RK3399配置isp超频的方法

- 配置PLL_NPLL为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 @@
                                <500000000>, <1000000000>,
                                <750000000>, <750000000>,
                                <816000000>, <816000000>,
-                               <600000000>, <200000000>,
+                               <650000000>, <200000000>,
                                <800000000>, <150000000>,
                                <750000000>, <375000000>,
                                <300000000>, <100000000>,
```

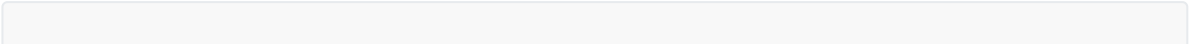
- 修改rk3399 isp最大支持频率为650M：

```
--- 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[] = {
```

- 转换芯片驱动中配置isp频率



```

#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);
    }

    ...
}

```

isp驱动中会对配置的频率再加25%的余量，所以在驱动中配置适当的频率RK628_CSI_PIXEL_RATE_HIGH即可。

```

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

static int __isp_pipeline_s_isc_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;

    ...
}

```

3.4.2 配置ISP使用CMA内存的方法

部分平台HDMI IN接收图像数据时，根据实际系统负载，可能会存在带宽不足导致丢帧或MIPI接收异常等问题，参考异常log如下：

```
[ 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)
```

此时需要提高DDR频率，若仍无改善，可给ISP预留使用CMA内存，以改善解决此问题：

- 在rockchip_defconfig配置预留CMA内存128MB

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

- 在dts配置ISP关闭IOMMU，使用CMA内存

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

3.5 EDID的配置方法

RK628D支持EDID配置，目前驱动代码中EDID支持的分辨率为：

3840x2160P30、1920x1080P60、1920x1080P30、1280x720P60、720x576P50、720x480P60

若需要修改分辨率支持，可直接在驱动代码中修改EDID：

```
drivers/media/i2c/rk628_csi.c

static u8 edid_init_data[] = {
    0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00,
    0x52, 0x62, 0x01, 0x88, 0x00, 0x88, 0x88, 0x88,
    0x1C, 0x15, 0x01, 0x03, 0x80, 0x00, 0x00, 0x78,
    0x0A, 0x0D, 0xC9, 0xA0, 0x57, 0x47, 0x98, 0x27,
    0x12, 0x48, 0x4C, 0x00, 0x00, 0x00, 0x01, 0x01,
    0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
    0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x02, 0x3A,
    0x80, 0x18, 0x71, 0x38, 0x2D, 0x40, 0x58, 0x2C,
    0x45, 0x00, 0xC4, 0x8E, 0x21, 0x00, 0x00, 0x1E,
    0x01, 0x1D, 0x00, 0x72, 0x51, 0xD0, 0x1E, 0x20,
    0x6E, 0x28, 0x55, 0x00, 0xC4, 0x8E, 0x21, 0x00,
    0x00, 0x1E, 0x00, 0x00, 0x00, 0xFC, 0x00, 0x54,
    0x37, 0x34, 0x39, 0x2D, 0x66, 0x48, 0x44, 0x37,
    0x32, 0x30, 0x0A, 0x20, 0x00, 0x00, 0x00, 0xFD,
    0x00, 0x14, 0x78, 0x01, 0xFF, 0x1D, 0x00, 0x0A,
    0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x01, 0x7B,
```

```
0x02, 0x03, 0x1A, 0x71, 0x47, 0x5F, 0x90, 0x22,  
0x04, 0x11, 0x02, 0x01, 0x23, 0x09, 0x07, 0x01,  
...
```

4. HDMI IN APK适配方法

4.1 获取和编译APK源码

APK源码提供在SDK目录：

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

需要将源码拷贝并解压到目录：

```
packages/apps/
```

以RK3288平台为例，在device/rockchip/rk3288/目录参照如下修改，增加rkCamera2 APK编译：

```
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 += \  

```

4.2 APK源码的适配

APK通过ioctl的方式访问RK628D设备节点，获取当前的连接状态和分辨率。RK628D设备节点在isp1/isp2/vicap链路上可能会差异。需要根据实际情况修改APK源码，获取设备节点的方法可参考[调试命令举例](#)章节。

```
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,errno=%s",video_name,strerror(errno));
    } else {
        LOGD("open %s success,fd=%d",video_name,camFd);
    }
}
```

获取当前的连接状态和分辨率代码如下：

```
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;
}
```

由于在APK访问了设备节点，所以需要确认是否关闭了selinux，可通过getenforce命令查看：

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

4.3 APK调试前的准备

APK调试前首先需要先将驱动调试完成，参考[驱动调试方法](#)章节。第二步需要确认camera设备是否正确注册到CameraHal，可通过以下命令查看，若未正确注册，则需要检查camera3_profiles.xml配置，参考[camera3_profiles.xml配置文件说明](#)章节。

```
rk3288:/ #
rk3288:/ # dumsys 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)
```

5. 驱动调试方法

驱动调试方法与SOC Sensor的调试方法一致，更多信息请参考redmine文档：

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

5.1 调试工具获取

需要使用media-ctl和v4l2-ctl工具，目前SDK编译固件时会自动拷贝集成，具体是放置在SDK目录：

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

若SDK版本较老，可通过redmine获取：

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

将media-ctl和v4l2-ctl用adb push 到设备的 /vendor/bin/ 目录。

5.2 调试命令举例

以RK3288 + RK628D 接收1920x1080P分辨率为例，具体调试时需要根据实际情况修改。

- 查看链路拓扑结构：

执行命令查看media节点的拓扑结构，根据不同芯片平台具体的链路连接，有可能是 /dev/media0 或 /dev/media1。

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

截取RK628D部分为例，可知RK628D设备为：/dev/v4l-subdev2，识别到HDMI IN分辨率为：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]
...
```

- 配置链路连接：

设备复位启动后，链路默认是连接上的。用HDMI IN APK打开再退出时，链路会被断开，查看拓扑结构，没有 [ENABLED] 时才需要重新链接。

```
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]'
```

- 配置分辨率：

```
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]'
```

- 获取图像数据流：

查看配置结果：

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

获取图像数据流：

```
v4l2-ctl --verbose -d /dev/video0 \
--set-fmt-video=width=1920,height=1080,pixelformat='NV12' \
--stream-mmap=4 \
--set-selection=target=crop,flags=0,top=0,left=0,width=1920,height=1080
```

抓取图像yuv文件，可adb pull上来用7yuv等工具查看：

```
v4l2-ctl --verbose -d /dev/video0 \  
--set-fmt-video=width=1920,height=1080,pixelformat='NV12' \  
--stream-mmap=4 --stream-skip=5 --stream-count=10 \  
--stream-to=/data/rk628_1920x1080.yuv --stream-poll
```

若一切正常，能接收到图像数据，会打出帧率，参考log如下：

```
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
```

6. 常见问题排查方法

6.1 打开log开关

可通过如下命令打开RK628D的log开关，然后通过dmesg命令获取kernel log：

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

若要抓取上电开机过程的log，建议直接修改代码并重新编译烧写kernel相关部分：

```
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.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 = 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)
```

6.2 寄存器读写

RK628寄存器调试节点如下，当前示例RK628接在I2C1，地址为0x51：

```
rk3288:/ # ls -l /d/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-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 1-0051-key_map
drwxr-xr-x 2 root root 0 1970-01-01 00:00 1-0051-rk628-pinctrl
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
```

寄存器节点默认只读，如果需要寄存器可写，需要添加如下修改：

```
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
```

读寄存器：

```
rk3288:/ # cat /d/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
...
```

写寄存器：

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

6.3 MFD设备报错排除

RK628D是MFD设备，在刚上电时，由于父设备还没有probe完成，子设备如COMBRXPHY等probe有可能会报错，返回-517，会去重新probe，所以刚上电时的报错没有影响。

```

0.950153] rk3x-i2c ff650000.i2c: Initialized RK3xxx I2C bus at e15c6000
0.951507] rk628 1-0051: GPIO lookup for consumer enable
0.951550] rk628 1-0051: using device tree for GPIO lookup
0.951677] rk628 1-0051: GPIO lookup for consumer reset
0.951696] rk628 1-0051: using device tree for GPIO lookup
0.998162] rk628-pinctrl rk628-pinctrl: bank->clk get error -517
0.998958] rk628-combrxphy rk628-combrxphy: failed to get pclk: -517
1.007020] rk628-hdmi rk628-hdmi: could not find pctldev for node /i2c@ff

1.008377] rk3x-i2c ff140000.i2c: Initialized RK3xxx I2C bus at e15c8000
1.009677] rk3x-i2c ff660000.i2c: Initialized RK3xxx I2C bus at e15ca000
1.010890] rk628-csi rk628-csi: could not find pctldev for node /i2c@ff14

```

6.4 Clk det 异常问题

COMBRXPY没有检测到信号，有可能是HDMI插入有效电平或是HPD有效电平配置出错，导致输入源HDMI信号没有正常输入。需要检查rk628_csi节点下plugin-det-gpios和hpd-output-inverted配置项，同时可以用万用表测试HPD电平状态。

注意异常时会进行retry，最多到retry 2。若retry后能够正常，则可认为信号正常。

```

m0_b_rk628-csi rk628-csi: tx_5v_power_present hdp det gpio val:0x1!
:628-combrxphy rk628-combrxphy: rk628-combrxphy power_on
:628-combrxphy rk628-combrxphy: clk det over cnt:10, reg_0x6654:0x501f0000
:628-combrxphy rk628-combrxphy: clock detected failed, cfg resistance manual!
:628-combrxphy rk628-combrxphy: error,clock error!
y phy-rk628-combrxphy.4: phy poweron failed --> -22
m0_b_rk628-csi rk628-csi: hdmi rxphy power on failed
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:1
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:2
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:3
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:4
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:5
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:6
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:7
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:8
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:9
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:10
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:11
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:12
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:13
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:14
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup read wxh:0x0, total:0x0, SCDC_REGS1:0x0, cnt:15
m0_b_rk628-csi rk628-csi: rk628_hdmirx_phy_setup hdmi rxphy lock failed, retry:0
:628-combrxphy rk628-combrxphy: rk628-combrxphy power_on
:628-combrxohv rk628-combrxohv: clock detected!

```

6.5 HDMI RX正常的判断方法

COMBRXPY正常锁定后，HDMI RX控制器能正常解析到Timing，Timing是计算出来，可能会有一些小误差，一般可能会差1。详细的Timing可参考CEA标准文档。

```

m00_b_rk628-csi rk628-csi: cnt_num:1000, tmds_cnt:3000, hs_cnt:15, vs_cnt:3667, hofs:192
m00_b_rk628-csi rk628-csi: SCDC_REGS1:0xffff0f00, act:1920x1080, total:2200x1125, fps:60,
m00_b_rk628-csi rk628-csi: hfp:88, hs:45, hbp:147, vfp:4, vs:5, vbp:36, interlace:0
m00_b_rk628-csi rk628-csi: rk628_csi_s_dv_timings: 1920x1080p60.0 (2200x1125)
m00_b_rk628-csi rk628-csi: enable_stream: disable

```

6.6 Open subdev 权限异常

```
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
```

需要确认是否已经给/dev/v4l-subdev*提供666的权限，在设备命令行查询：

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

目前SDK代码已经包含了本提交，若未包含，可在device/rockchip/common/参考如下修改：

```
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
```