

Rockchip Linux4.4 Camera Trouble Shooting

ID: RK-PC-YF-331

Release Version: V1.0.1

Release Date: 2020-11-13

Security Level: ☐Top-Secret ☐Secret ☐Internal ☒Public

DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". ROCKCHIP ELECTRONICS CO., LTD.("ROCKCHIP")DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

All rights reserved. ©2020. Rockchip Electronics Co., Ltd.

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

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

Overview

This document will introduce some frequently asked problems and troubleshooting ideas in the debugging process of RKISP and camera.

Product Version

Chipset	Kernel Version
RK3xxx	Linux 4.4 or latest version

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Date	Author	Version	Change Description
2020-02-03	ZhengSQ	V1.0.0	Initial version
2020-11-13	Ruby Zhang	V1.0.1	Update the document format

Contents

Rockchip Linux4.4 Camera Trouble Shooting

1. Whether the Sensor is Lighted up
 - 1.1 Sensor ID Cannot be Recognized, I2C Communication Fails
 - 1.1.1 What is 7-bits Address
 - 1.1.2 The 24M mclk and vdd Power Supply Cannot be Detected After Power on
 - 1.1.3 The 24M mclk Still Cannot be Detected
 - 1.1.4 There is a mclk, But the Voltage Amplitude is Wrong and Different from Actual Power Domain
 - 1.1.5 Check Whether the Power-on Sequence of Sensor Meets the Requirements
 - 1.2 What are the Default Values of `exp_def`, `hts_def`, and `vts_def` in Sensor Driver
 - 1.3 What Should be `link_freq` and `pixel_rate` Values
 - 1.4 How to Make Sure a Sensor is Lighted Up
 - 1.5 Sensor AVL List
2. MIPI/ISP Abnormal
 - 2.1 What Parameters Need to be Set in MIPI
 - 2.2 No Frame Data is Received, and No Error is Reported from ISP/MIPI
 - 2.3 MIPI Error
 - 2.3.1 The Detailed MIPI Error Message Table
 - 2.3.2 How to Deal with SOT/SOT_SYNC Errors
 - 2.3.3 How to Deal with CRC/Checksum(CS), ECC/ECC1/ECC2 Errors
 - 2.3.4 How to Deal with ERR_PROTOCOL/ERR_F_BNDRY Errors
 - 2.3.5 Receive Frames Normally, but MIPI Errors Appear Occasionally
 - 2.3.6 Report Many MIPI Errors or Even Crashes
 - 2.3.7 How to Deal with ISP PIC_SIZE_ERROR
3. About Capturing Images
 - 3.1 Ways to Capture Pictures
 - 3.2 The Color of the Captured Image is Wrong, and the Brightness is Obviously Darker or Lighter
 - 3.3 What is ISP Topology and How to Use media-ctl Command
 - 3.3.1 How does One ISP Connect to Multiple Sensors
 - 3.4 Whether Captured RAW Images are Exactly the Same as the Original Images
 - 3.5 How does ISP Output two Channels (MP, SP) Simultaneously
 - 3.6 Does ISP Have Amplify Function
 - 3.7 Does ISP Have Rotation Function
 - 3.8 How to Capture a Gray Image
 - 3.9 Which Formats are Supported by RGB Images
 - 3.10 How to Quickly Preview a Board Without Screen
 - 3.11 How to Distinguish SP and MP
4. About 3A
 - 4.1 How to Confirm the Version of camera_engine_rkisp
 - 4.1.1 How to Confirm the Version Number of rkisp Kernel Driver Required by camera_engine_rkisp
 - 4.2 How to Upgrade camera_engine_rkisp
 - 4.3 How to Confirm Whether 3A is Working Properly
 - 4.3.1 Did not Found the rkisp_3A_server Process
 - 4.3.2 How to Start rkisp_3A_server
 - 4.3.3 How to Make Sure the Name and Path of Sensor IQ Configuration File (xml)
 - 4.4 How to Manually Expose
 - 4.5 How to Open librkisp Log
5. Application Development
 - 5.1 C Language Reference Demo
 - 5.2 What is DMA Buffer and What are the Benefits

1. Whether the Sensor is Lighted up

1.1 Sensor ID Cannot be Recognized, I2C Communication Fails

If sensor ID is not recognized, it has nothing to do with RKISP or RKCIF, but the sensor power-on sequence does not meet the requirements.

Please check the following items in orders:

1. Is the **7-bits** i2c slave id of the sensor correct? Is it written as 8-bits by mistake?
2. Whether the 24M mclk has output and whether the voltage amplitude is correct?
3. Whether the power-on sequence of the sensor meets the requirements, including avdd, dovdd, dvdd, power down, reset, etc.

1.1.1 What is 7-bits Address

The least significant bit (LSB) of the 8bits represents R/W, and the higher 7bits is i2c slave id we needed.

1.1.2 The 24M mclk and vdd Power Supply Cannot be Detected After Power on

In the implementation of sensor driver, mclk and power are generally turned on only when needed, so mclk and power are turned off by default after booting.

When debugging, you can comment out the implementation of the `power_off()` function in driver, so that it will not be powered off and is convenient for measurement.

1.1.3 The 24M mclk Still Cannot be Detected

When using an oscilloscope, check whether the bandwidth of the oscilloscope is sufficient. A bandwidth of at least 48M is recommended.

1. When sensor does not open mclk correctly, please refer to the operation of mclk in `drivers/media/i2c/ov5695.c`.
2. When the gpio is occupied by other modules, there will be some tips in kernel log. You can also use io command to check whether the pin-ctrl register setting is correct.

1.1.4 There is a mclk, But the Voltage Amplitude is Wrong and Different from Actual Power Domain

This is due to a misconfiguration of io-domains. Generally, the voltage of io-domains is 1.8v or 3.3v, depending on your schematic design. Check the schematic diagram and modify io-domains accordingly, such as:

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

    vccio1-supply = <&vcc1v8_soc>;
    vccio2-supply = <&vccio_sd>;
    vccio3-supply = <&vcc1v8_dvp>;
    vccio4-supply = <&vcc_3v0>;
    vccio5-supply = <&vcc_3v0>;
};
```

In the above example, the io port of vccio3 is powered by vcc1v8_dvp, and its power domain is 1.8v, so the detected mclk should also be 1.8v.

During the boot process, io_domains may be initialized later than sensor. When sensor id reading fails, try to retry several times, as follows:

```
diff --git a/drivers/media/i2c/ov7251.c b/drivers/media/i2c/ov7251.c
index e0c608f..7842812 100644
--- a/drivers/media/i2c/ov7251.c
+++ b/drivers/media/i2c/ov7251.c
@@ -961,7 +961,8 @@ static int ov7251_check_sensor_id(struct ov7251 *ov7251,
                                OV7251_REG_VALUE_16BIT, &id);
    if (id != CHIP_ID) {
        dev_err(dev, "Unexpected sensor id(0x%x), ret(%d)\n", id, ret);
-       return -ENODEV;
+       return -EPROBE_DEFER;
    }
}
```

If io_domains has not been initialized when sensor is initialized, then io_domains will use the default value. If the default value is different from the actual hardware power domain, the voltage of mclk will not meet expectations. Returning `-EPROBE_DEFER` at this time will let sensor try probe again in the later boot process.

1.1.5 Check Whether the Power-on Sequence of Sensor Meets the Requirements

The datasheet of a sensor generally describes the power-on sequence and interval requirements of each power supply in details. Please check if it meets the requirements by an oscilloscope.

There is no time sequence requirement for power supply vdd of some sensors, such as ov5695, which may use `regulator_bulk` to manage the power supply in its driver; but some sensors **have sequential requirements**, such as ov2685.c, which uses multiple regulators to control separately in driver. such as `avdd_regulator`, `dovdd_regulator`. Please choose accordingly.

1.2 What are the Default Values of `exp_def`, `hts_def`, and `vts_def` in Sensor Driver

If you have the contact information of the sensor vendor, please contact them to get. Otherwise, please find the corresponding register from the datasheet and find the value configured during initialization from the register list. Take `ov2685.c` as an example:

```
#define OV2685_REG_VTS                0x380e

...

    {0x380e, 0x05},
    {0x380f, 0x0e},

...

    .vts_def = 0x050e,
```

The 0x380e and 0x380f are vts corresponding registers. The value configured during initialization is 0x050e, then `vts_def` is 0x050e. The `exp` and `hts` use default values and can be found directly from the datasheet.

If you don't like application to adjust exposure and frame rate, `exp`, `hts`, `vts` are not needed. Generally, a sensor in RAW format requires these three parameters.

1.3 What Should be `link_freq` and `pixel_rate` Values

The `link_freq` refers to the actual frequency of MIPI clk. **Note that it is not 24M mclk, but MIPI dn/dp clk.**

Check from the vendor first, or find whether the datasheet has relevant parameters.

Generally, the actual value of `link_freq` will not be less than the calculation result of the following formula with unit is (Hz)

```
width * height * fps * bits_per_pixel / lanes / 2
```

If you really don't know the actual value of `link_freq`, you can measure by an oscilloscope.

The `pixel_rate` refers to the number of pixels transmitted per second. After `link_freq` is determined, the following formula can be used to calculate:

```
link_freq * 2 * lanes / bits_per_pixel
```

1.4 How to Make Sure a Sensor is Lighted Up

Firstly, the sensor id can be recognized, that is, there should be no abnormalities in the reading and writing of i2c. At this time, the detailed information of the sensor such as the name and resolution should be able to checked by `media-ctl -p -d /dev/media0`.

Secondly, when upper layer captures pictures, MIPI must be able to output data without reporting MIPI/ISP errors, and application layer can receive frames.

1.5 Sensor AVL List

RGB sensor AVL is located at https://redmine.rockchip.com.cn/projects/rockchip_camera_module/support_list/camera. The detailed information of sensor modules are shown in the support list.

If it is other non-RGB sensor, such as YUV sensor, you can directly find the drivers/media/i2c/ directory of the kernel source code. The drivers have all been tuned by Rockchip.

2. MIPI/ISP Abnormal

In the early stage of sensor tuning, the following problems maybe frequently encountered:

1. No frame data is received, and no error is reported by ISP/MIPI
2. Keep printing MIPI errors log
3. ISP reports PIC_SIZE_ERROR
4. MIPI errors occasionally
5. MIPI keeps reporting errors until it crashes

2.1 What Parameters Need to be Set in MIPI

There are 4 parameters need to be set in MIPI communication between sensor and ISP, please **confirm** the correctness of the 4 MIPI parameters.

1. Resolution of sensor output
2. The image format output from sensor is YUV or RGB RAW, 8-bits, 10-bits, or 12-bits
3. The `link_freq` output from sensor's MIPI
4. If there are several MIPI lanes in sensor, please make sure they are configured correctly in `dtb`

2.2 No Frame Data is Received, and No Error is Reported from ISP/MIPI

1. Confirm whether there are any errors about MIPI in kernel log. For example, check if there are any errors by `dmesg | grep MIPI`.
2. Confirm whether there is a sensor's i2c read and write failure in kernel log. If sensor fails to configure registers, the sensor may not be initialized correctly and enable output.
3. Measure whether there is information output from clk and data lines of MIPI. If not, it is recommended to analyze sensor initialization and hardware.

4. **There is MIPI signal** output in measure, but no error is reported and no data can be received:
- Please check according to the Section 2.1 [What Parameters Need to be Set in MIPI](#) again
 - Please confirm that there is no error in I2C communication, and the sensor register initialization list have all been written to the sensor
 - In sensor driver, the last to enable MIPI output is `s_stream()` . Please make sure that before this function, especially `s_power()` , there is no MIPI signal output. The reason is MIPI controller is not actually ready to receive data before `s_stream()` . If data is output before `s_stream()` , it may cause the loss of MIPI protocol header SOT signal.
 - Switch the clock lane on camera sensor side from continue mode to no continues.

2.3 MIPI Error

2.3.1 The Detailed MIPI Error Message Table

For RK3288/RK3399/RK3368, the error message table is as follows:

Error bit (Bit)	Abbreviation	Description
25	ADD_DATA_OVFLW	additional data fifo overflow occurred
24	FRAME_END	A frame is received normally, not an error
23	ERR_CS	checksum error
22	ERR_ECC1	1-bit ecc error
21	ERR_ECC2	2-bit ecc error
20	ERR_PROTOCOL	packet start detected within current packet
19:16	ERR_CONTROL	PPI interface control error occured, one bit per lane
15:12	ERR_EOT_SYNC	MIPI EOT(End Of Transmission) sync, one bit per lane
11:8	ERR_SOT_SYNC	MIPI SOT(Start Of Transmission) sync, one bit per lane
7:4	ERR_SOT	MIPI SOT(Start Of Transmission), one bit per lane
3:0	SYNC_FIFO_OVFLW	synchronization fifo overflow occurred, one bit per lane

For RK3326/PX30/RK1808, the three error message tables are as follows:

ERR1 Error Bit (Bit)	Abbreviation	Description
28	ERR_ECC	ECC ERROR
27:24	ERR_CRC	CRC ERROR
23:20	ERR_FRAME_DATA	Frame transmission is complete, but at least one CRC error is included
19:16	ERR_F_SEQ	Frame Number is not continuous will not meet expectations
15:12	ERR_F_BNDRY	Frame start and Frame end do not match
11:8	ERR_SOT_SYNC	MIPI PHY SOT(Start Of Transmission) sync error
7:4	ERR_EOT_SYNC	MIPI PHY EOT(End Of Transmission) sync error

ERR2 Error Bit (Bit)	Abbreviation	Description
19:16	ERR_CONTROL	
15:12	ERR_ID	
11:8	ERR_ECC_CORRECTED	
7:4	ERR_SOTHS	PHY SOTHS error
3:0	ERR_ESC	PHY ESC error

Please read the following chapters for analysis of frequently encountered problems.

2.3.2 How to Deal with SOT/SOT_SYNC Errors

The SOT signal needs to comply with **MIPI_D-PHY Specification**. If you need in-depth analysis, please search pdf documents directly from Internet, and it is suggested to focus on the following items:

1. High-Speed Data Transmission
2. Start-of-Transmission Sequence
3. HS Data Transmission Burst
4. High-Speed Clock Transmission
5. Global Operation Timing Parameters

But generally speaking, if the sensor has tuned on other platforms, there will be little possibility of not complying with MIPI protocol. It is recommended to customers that:

1. Please first confirm to sensor vendor whether the sensor has actually successfully transmit data on MIPI interface
2. **Reconfirm whether link_freq is correct.** Because the `Ths-settle` in SOT sequence needs to be configured correctly at MIPI receiving side, `link_freq` is so important.

3. If multiple lanes are used, see if the sensor vendor can modify it to 1 lane for transmission.

2.3.3 How to Deal with CRC/Checksum(CS), ECC/ECC1/ECC2 Errors

When ECC error and CS check error occurred, it means that data was incomplete during transmission. It is suggested that:

1. **Check hardware signals firstly**
2. If multiple lanes are used, check if the sensor vendor can modify it to 1 lane for transmission. Because multiple lanes are not synchronized well, ECC errors may also occur.

2.3.4 How to Deal with ERR_PROTOCOL/ERR_F_BNDRY Errors

This error indicates that the expected EOT/SOT was not received. And SOT and EOT should appear in pairs. It is recommended to check the waveform.

2.3.5 Receive Frames Normally, but MIPI Errors Appear Occasionally

If it is a MIPI error, please refer to the above error description. It is recommended to analyze the signal related errors from hardware.

In particular, if MIPI error occurs only at the beginning of capturing images, it is possible that there is MIPI signal output during sensor power-on but it does not conform to the protocol, so the error is reported.

In this case, you can try to modify as follows:

1. Put the initialization of the complete sensor register into `s_power()`. Because MIPI receiver has not yet started receiving data at this time, all data will be ignored.
2. At the end of the `s_power()` function, turning off the output of sensor, in other words, to call `stop_stream()`
3. In `start_stream()` and `stop_stream()`, output of MIPI is turned on or off only.

2.3.6 Report Many MIPI Errors or Even Crashes

It may be a worse situation mentioned in section [2.3.5 Receive Frames Normally, but MIPI Errors Appear Occasionally](#).

The reason for this is that MIPI signal does not meet the requirement, and some errors at MIPI receiving side are voltage interruptions, which leads to an interruption storm and eventually crashes.

You can try to check whether it will take effect according to section [2.3.5 Receive Frames Normally, but MIPI Errors Appear Occasionally](#).

2.3.7 How to Deal with ISP PIC_SIZE_ERROR

Picture size error is an ISP level error, which indicates that the expected number of rows and columns have not been received. So check resolution size from all levels.

If the previous level (that is MIPI) reported an error, the MIPI error should be resolved first.

Please check the following items:

1. Is DDR frequency too small? This error can also occur when DDR frequency is too low and the response speed is insufficient. Try to fix DDR frequency to the highest frequency to see if it will go wrong:

```
echo performance > /sys/class/devfreq/dmc/governor
```

2. In the whole ISP link, is there a case that the resolution of the latter stage is greater than that of the previous stage. You can use `media-ctl -p -d /dev/media0` to check the topology. The resolution should satisfy `Sensor == MIPI_DPHY >= isp_sd input >= isp_sd output`. If you haven't modified it manually, it should meet this condition by default.
3. Is the output resolution of sensor correct? Try to force the resolution to be smaller in the driver code. For example, the default resolution in `ov7251.c` is 640x480:

```
static const struct ov7251_mode supported_modes[] = {
    {
        .width = 640,
        .height = 480,
```

Change the width and height to be smaller, such as 320x240, the register configuration does not need to be changed, in order to confirm whether the configuration size of the sensor will exceed the actual output size.

```
static const struct ov7251_mode supported_modes[] = {
    {
        .width = 320,
        .height = 240,
```

3. About Capturing Images

This chapter will going to introduce common problems related to capturing images.

3.1 Ways to Capture Pictures

RKISP and RKCIF drivers support v4l2 interface, you can use the following ways to capture images:

1. Use the `v4l2-ctl` tool in the `v4l-utils` package to capture images. **In the tuning process, it is firstly recommended to use this tool to check whether pictures can successfully output.** The `v4l2-ctl` capture pictures and saved them as a file, but it cannot parse images and display them. If you need parse, `mplayer` can be used in Ubuntu/Debian, and `7yuv`, etc. can

be used in Windows. For detailed instructions of v4l2-ctl and mplayer tools, please refer to "Rockchip_Developer_Guide_Linux_Camera_CN.pdf". The v4l2-ctl also comes with detailed `v4l2-ctl --help` document.

2. The Linux SDK provided by RK contains qcamera application app. Open the camera app on the desktop directly, select the video device and preview. **The app is a demo based on Qt, for reference only. In customer projects, it is recommended to develop apps by yourselves.**
3. Use v4l2src plugin of gstreamer to get images from /dev/video devices and **display them on the screen**. The Linux SDK provided by RK will contain some scripts under the directory /rockchip_test/camera/, please refer to it first.

Note: RK has provided multiple versions of gstreamer isp plugin, such as rkisp, rkiv4l2src, which are no longer supported. Please use the v4l2src plugin that comes with gstreamer directly. There are two main reasons:

- 3A no longer needs to be adjusted in rkisp or rkiv4l2src. Please refer to Chapter 4 [About 3A](#) for details
 - v4l2 interface driven by rkisp is more standardized
4. Open source tools such as vlc are used in Debian system

After installing vlc through apt, you can use the following command to display the camera image on the screen:

```
vlc v4l2:///dev/video1:width=640:height=480
```

Note that the permission of the video user group or root super user permission is required.

3.2 The Color of the Captured Image is Wrong, and the Brightness is Obviously Darker or Lighter

According to types of sensors:

1. If sensor is output of RAW RGB, such as RGGB, BGGR, etc. It needs 3A to run normally. Please refer to Chapter 4 [About 3A](#). When 3A is running normally, please check again whether the format used when parsing/displaying images is correct and whether the uv component is reversed.
2. If sensor is yuv output, or RGB such as RGB565, RGB888, at this time ISP is in bypass state,
 - If the color is wrong, please confirm whether the output format of sensor is misconfigured and whether the uv component is reversed. When you confirmed that they are correct, it is recommended to contact the sensor vendor
 - If the brightness is obviously wrong, please contact the sensor vendor

3.3 What is ISP Topology and How to Use media-ctl Command

RKISP or RKCIF can be connected to multiple sensors, time-sharing multiplexing; at the same time, RKISP also has a multi-level cutting function. Therefore, each node is connected by topology, and the parameters can be configured separately through media-ctl. About the use of media-ctl, there is a more complete description in the "Rockchip_Developer_Guide_Linux_Camera_CN.pdf" document.

3.3.1 How does One ISP Connect to Multiple Sensors

Multiple sensors can be connected, but only time-sharing multiplexing. By configuring dts, after linking multiple sensors to MIPI DPHY, the sensors can be switched through media-ctl.

3.4 Whether Captured RAW Images are Exactly the Same as the Original Images

When ISP capture sensor RAW images (such as RGGB, BGGR) in the bypass mode, 8bit alignment is required, and if it is less than 8bit, the low bit will be filled with 0, that is

- If they are 8bit, 16bit original images, images captured by the application are the original, and not filled
- If they are 10bit, 12bit original images, the low bit of each pixel will be filled with 0 to 16bit

Only the video device corresponding to MP can output RAW images, SP cannot support RAW image output.

3.5 How does ISP Output two Channels (MP, SP) Simultaneously

RKISP has two outputs: SP and MP, that is, an image from sensor, SP and MP can cut and change format of the image separately, and output at the same time.

For the detailed video processing capabilities of SP and MP, please refer to "Rockchip_Developer_Guide_Linux_Camera_CN.pdf".

Only when SP and MP both output RGB or YUV, they can output at the same time. If MP outputs RAW images, then there will be no output from SP

3.6 Does ISP Have Amplify Function

This function is supported on the hardware, but it is not recommended to use it. And it is also turned off by default in the driver.

3.7 Does ISP Have Rotation Function

No. If you need the rotation function, it is recommended to:

- If it is flip, mirror, first check if the sensor has this function, if so, use it directly. This is the most efficient
- If there is no sensor flip, mirror, RGA module is consideration, its code and demo are located in the external/linux-rga/ directory, and related documents are located in the docs/ directory

3.8 How to Capture a Gray Image

As long as ISP can output YUV or sensor output is Y8 gray image, application can always use V4L2_PIX_FMT_GREY (FourCC for GREY) format to capture images directly.

3.9 Which Formats are Supported by RGB Images

First of all, only SP can support RGB output, the format are: V4L2_PIX_FMT_XBGR32, V4L2_PIX_FMT_RGB565. The XBGR32 (corresponding FourCC is XR24) contains four components of R, G, B, and X, and X component is always 0.

Does not support RGB888, that is, 24bit format output.

3.10 How to Quickly Preview a Board Without Screen

The external/uvic_app/ directory in the SDK provides the function of simulating the board into an uvc camera. Please refer to the introduction file and code in the directory. After the board is connected to a PC, an usb camera can be recognized and images can be previewed.

3.11 How to Distinguish SP and MP

The `media-ctl -p -d /dev/media0` (if there are multiple media devices, /dev/media1, /dev/media2 can be tried, too) can be used to see topology, and list part of the output as follows:

```
# media-ctl -p -d /dev/media0
...
- entity 2: rkisp1_mainpath (1 pad, 1 link)           //Means the entity is
MP(MainPath)
    type Node subtype V4L flags 0
    device node name /dev/video1                     //The corresponding
device node is /dev/video1
    pad0: Sink
        <- "rkisp1-isp-subdev":2 [ENABLED]

- entity 3: rkisp1_selfpath (1 pad, 1 link)           //Means the entity is
SP(SelfPath)
    type Node subtype V4L flags 0
```

```

        device node name /dev/video2                                //The corresponding
device node is/dev/video2
        pad0: Sink
            <- "rkisp1-isp-subdev":2 [ENABLED]
...

```

In a few cases, if there is no media-ctl command, you can search through /sys/ node, such as:

```

# grep '' /sys/class/video4linux/video*/name
/sys/class/video4linux/video0/name:stream_cif
/sys/class/video4linux/video1/name:rkisp1_mainpath           # Corresponding MP
node/dev/video1
/sys/class/video4linux/video2/name:rkisp1_selfpath          # Corresponding SP
Node/dev/video2
/sys/class/video4linux/video3/name:rkisp1_rawpath
/sys/class/video4linux/video4/name:rkisp1_dmapath
/sys/class/video4linux/video5/name:rkisp1-statistics
/sys/class/video4linux/video6/name:rkisp1-input-params

```

4. About 3A

If a sensor needs 3A tuning, like RAW BAYER RGB format of RGGB, BGGR ,etc. format output from sensor, then RKISP is required to provide image processing.

Depending on the version of camera_engine_rkisp, 3A processing methods are different. It is recommended to upgrade camera_engine_rkisp to the latest version as much as possible.

Please first confirm whether the module is in the support list

- For those already in the support list, there will be a corresponding xml file in the external/camera_engine_rkisp/iqfiles/ directory
- Otherwise, please apply a module tuning application by business

4.1 How to Confirm the Version of camera_engine_rkisp

1. Check from the source code

```

# grep CONFIG_CAM_ENGINE_LIB_VERSION interface/rkisp_dev_manager.h
define CONFIG_CAM_ENGINE_LIB_VERSION "v2.2.0"           # The output v2.2.0 is
the version number of librkisp.so

```

2. Check from running log

```

# persist_camera_engine_log=0x4000 rkisp_3A_server --mmedia=/dev/media1 | grep
"CAM ENGINE LIB VERSION"
CAM ENGINE LIB VERSION IS v2.2.0                         # The output v2.2.0 is
the version number of librkisp.so

```


If the version number is lower than v2.2.0, please consider upgrading to v2.2.0 or a newer version

4.1.1 How to Confirm the Version Number of rkisp Kernel Driver Required by camera_engine_rkisp

The camera_engine_rkisp has requirements for kernel driver version, so you need to ensure that the rkisp driver is latest enough.

1. Check ISP driver version from the kernel source code

```
# grep RKISP1_DRIVER_VERSION drivers/media/platform/rockchip/isp1/version.h
define RKISP1_DRIVER_VERSION KERNEL_VERSION(0, 1, 0x5) # The output v0.1.5 is
the version number of the rkisp driver
```

2. Check ISP driver version from the kernel log

```
# dmesg | grep "rkisp1 driver version"
[ 0.867864] rkisp1 ff4a0000.rkisp1: rkisp1 driver version: v00.01.05
```

4.2 How to Upgrade camera_engine_rkisp

There are three parts:

1. The body of camera_engine_rkisp

It is located in the external/camera_engine_rkisp directory of the SDK, it can be updated directly through git or repo tools. You can only update the directory without affecting the directories in other SDKs.

2. Kernel is upgraded according to needs of camera_engine_rkisp

By checking `git log` in the external/camera_engine_rkisp directory, you can find the version number of the kernel rkisp driver. For example:

```
# git log
commit e456a50a5524792d64dac384604d4136a697deac
Author: ZhongYichong <zyc@rock-chips.com>
Date: Mon Jul 1 11:26:32 2019 +0800

    librkisp: v2.2.0

    (BY ZSQ: UPDATE v2.2.0 iq version: from v1.4.0 to v1.5.0)

3A lib version:
    af: v0.2.17
    awb: v0.0.e
    aec: v0.0.e
iq version: v1.5.0
iq magic version code: 706729
```

```
matched rkisp1 driver version:
v0.1.5                # The needed kernel driver version is v0.1.5

Change-Id: I3d2adb949dadec259b9ba587a3e3b2770a1c155d
Signed-off-by: ZhongYichong <zyc@rock-chips.com>
Signed-off-by: Shunqian Zheng <zhengsq@rock-chips.com>
```

3. The compilation script of camera_engine_rkisp in buildroot

It is located in the directory of buildroot/package/rockchip/camera_engine_rkisp. If it is not convenient to update the whole buildroot, you can update this directory alone.

4.3 How to Confirm Whether 3A is Working Properly

After confirming that camera_engine_rkisp is already v2.2.0 or later version. By capturing an image, check whether the color and exposure of the image are normal.

At the same time, by checking whether there is rkisp_3A_server process in the background, as follows:

```
# ps -ef | grep rkisp_3A_server
706 root    9176 S    /usr/bin/rkisp_3A_server --mmedia=/dev/media1
746 root    2408 S    grep rkisp_3A_server
# pidof rkisp_3A_server
706
```

You can see that the process number 706 is rkisp_3A_server.

4.3.1 Did not Found the rkisp_3A_server Process

1. First confirm whether the /usr/bin/rkisp_3A_server executable file exists, if not, please check the camera_engine_rkisp version and build it.
2. Check whether there are rkisp_3A related errors in /var/log/syslog. If so, see what the detailed error is and whether the xml corresponding to the sensor module is not found or does not match.
3. Execute `rkisp_3A_server --mmedia=/dev/media0` in a shell (if there are multiple /dev/media devices, select the one corresponding to /dev/video), and capture pictures from another shell, To get the error message corresponding to rkisp_3A_server

4.3.2 How to Start rkisp_3A_server

In the Linux SDK, rkisp_3A_server is started by the script /etc/init.d/S40rkisp_3A and executed in the background.

If the /etc/init.d/S40rkisp_3A file is not found, check the version of camera_engine_rkisp and the buildroot package compilation script.

4.3.3 How to Make Sure the Name and Path of Sensor IQ Configuration File (xml)

The sensor iq file consists of three parts,

1. Sensor Type, such as ov5695, imx327
2. Module Name, is defined in dts, for example, rk3326/px30 rk evb board, the name is "TongJu":

```
rockchip,camera-module-name = "TongJu";
```

3. Module Lens Name, defined in dts, such as the following "CHT842-MD":

```
rockchip,camera-module-lens-name = "CHT842-MD";
```

The iq file name in the above example is: ov5695_TongJu_CHT842-MD.xml, which is stored in the /etc/iqfiles/ directory. Note the case-sensitive.

4.4 How to Manually Expose

In the case of manual exposure, the rkisp_3A_server process must exit first. Then you can refer to the source code of rkisp_demo.cpp program or librkisp_api.so.

4.5 How to Open librkisp Log

By setting the environment variable persist_camera_engine_log, the corresponding bits are list as follows:

```
bits:    23-20    19-16 15-12  11-8  7-4    3-0
module: [xcore]  [ISP]  [AF]   [AWB] [AEC] [NO]

0: error
1: warning
2: info
3: verbose
4: debug
```

For example, open the debug log of ISP and AWB:

```
# /etc/init.d/S40rkisp_3A stop
# export persist_camera_engine_log=0x040400
# /usr/bin/rkisp_3A_server --mmedia=/dev/media0
```

5. Application Development

5.1 C Language Reference Demo

1. The Linux SDKs provided by RK include rkisp_demo tool and source code

rkisp_demo is a simple tool that can be used to capture images. Like v4l2-ctl tool, rkisp_demo can't display images either. It mainly provides source code for reference.

The source code is located in the external/camera_engine_rkisp/apps directory. If your code is out of date, please refer to the source code is located in the external/camera_engine_rkisp/tests directory.

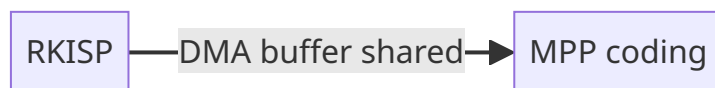
2. The Linux SDKs provided by RK include rkisp_api.so dynamic link library and source code, you can modify base on it or directly use C language to develop programs

The source code is located in the external/camera_engine_rkisp/apps directory. If your code is out of date and cannot find the directory, please update.

5.2 What is DMA Buffer and What are the Benefits

DMA buffer is a part of memory allocated by driver. This buffer can be shared among multiple kernel modules to reduce memory copy. Especially for image processing, it can optimize performance and reduce DDR load.

For example, the camera image needs to be encoded by mpp:



RKISP can receive DMA buffer from other modules (such as MPP, RGA), and can also allocate memory and export DMA buffer to other modules.

For details, please refer to the source code of libriskisp_api.so library.