

Rockchip Linux4.4 Camera Trouble Shooting CN

文件标识：RK-PC-YF-331

发布版本：V1.0.0

日期：2020-02-04

文件密级：公开资料

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有© 2020福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

本文记录RKISP 及 Camera 在调试过程中常见的一些问题与排查思路。

产品版本

芯片名称	内核版本
RK3xxx	Linux 4.4及以上

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

日期	作者	版本	主要内容
2020.02.03	ZhengSQ	V1.0.0	初始版本

目录

Rockchip Linux4.4 Camera Trouble Shooting CN

前言

目录

1 Sensor点亮相关

1.1 Sensor ID识别不到, I2C通讯失败

1.1.1 什么是7-bits地址

1.1.2 开机后, 测量不到24M mclk 和 vdd电源

1.1.3 仍然测量不到24M mclk

1.1.4 有mclk, 但电压幅度不对, 与实际的电源域不同

1.1.5 检查Sensor的上电时序是否满足要求

1.2 Sensor驱动中的 exp_def、hts_def、vts_def 默认值是多少

1.3 link_freq 与 pixel_rate 值应该是多少

1.4 怎么才算点亮Sensor

1.5 Sensor AVL列表

2 MIPI/ISP异常相关

2.1 MIPI需要设置哪些参数

2.2 没有收到帧数据, 也没有看到ISP/MIPI有报错

2.3 MIPI报错

2.3.1 MIPI错误信息详细表

2.3.2 如何处理SOT/SOT_SYNC错误

2.3.3 如何处理CRC/Checksum(CS)、ECC/ECC1/ECC2错误

2.3.4 如何处理ERR_PROTOCOL/ERR_F_BNDRY错误

2.3.5 能正常收帧, 但偶现MIPI错误

2.3.6 报很多MIPI错误甚至死机

2.4 如何处理ISP PIC_SIZE_ERROR

3 获取图像相关

3.1 有哪些方式可以抓图

3.2 抓到的图颜色不对, 亮度也明显偏暗或偏亮

3.3 什么是ISP的拓扑结构(topology, 链路结构), 如何使用media-ctl命令

3.3.1 一个ISP怎样接多个Sensor

3.4 抓取RAW图是否与原图完全一致

3.5 ISP怎样双路(MP, SP)同时输出

3.6 ISP是否具有放大功能

3.7 ISP是否具有旋转功能

3.8 怎样抓灰度(GREY)图

3.9 RGB图支持哪些格式

3.10 无屏板卡如何快速预览

3.11 如何区分SP与MP

4 3A相关

4.1 如何确认camera_engine_rkisp的版本

4.1.1 如何确认camera_engine_rkisp所需要的rkisp kernel驱动的版本号

4.2 如何升级camera_engine_rkisp

4.3 如何确认3A是否正在工作

4.3.1 没有看到rkisp_3A_server进程

4.3.2 rkisp_3A_server是如何启动的

4.3.3 如何确定Sensor iq配置文件(xml)文件名及路径

4.4 怎样手动曝光

4.5 如何打开librkisp的log

5 应用开发相关

5.1 C语言参考demo

5.2 什么是DMA buffer, 有什么好处

1 Sensor点亮相关

1.1 Sensor ID识别不到, I2C通讯失败

Sensor ID如果未识别到, 这与RKISP或RKCIF没有任何关系, 仅仅是Sensor上电时序未满足要求。

请按以下顺序排查:

1. Sensor 的7-bits i2c slave id 是否正确, 是否误写成8-bits
2. 24M mclk是否有输出, 电压幅度是否正确。
3. Sensor的上电时序是否满足要求, 主要包括avdd, dovdd, dvdd, power down, reset等

1.1.1 什么是7-bits地址

8bits中的最低位(LSB)表示R/W, 高7bits即是我们需要的i2c slave id。

1.1.2 开机后, 测量不到24M mclk 和 vdd电源

在Sensor 驱动的实现中, 一般是只有在需要时才开启mclk及电源, 因此开机后mclk及电源默认是关闭的。

调试时, 可以将驱动中的 `power_off()` 函数的实现注释掉, 这样不会下电, 方便测量。

1.1.3 仍然测量不到24M mclk

使用示波器时, 检查下示波器的带宽是否足够, 建议至少48M以上的带宽。

1. Sensor没有正确打开mclk, 请参考如 `drivers/media/i2c/ov5695.c` 中对mclk的操作。
2. 该gpio被其它模块占用了, 这种情况时, 一般kernel log会有相应的提示。还可以通过io命令去查看pin-ctrl寄存器设置是否正确。

1.1.4 有mclk, 但电压幅度不对, 与实际的电源域不同

这是由于io-domains配置错误, 一般io-domains的电压是1.8v或3.3v, 根据您的原理图的设计来决定。通过查看原理图, 并根据实际情况修改io-domains, 比如:

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

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

上例中, 比如vccio3这组io口是由vcc1v8_dvp供电, 他的电源域是1.8v, 因此测量到的mclk也应该是1.8v。

在开机过程中, io_domains有可能比Sensor更晚初始化, 当Sensor id读取失败时, 尝试重试几次, 如下:

```
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(%06x), ret(%d)\n", id, ret);
-       return -ENODEV;
+       return -EPROBE_DEFER;
    }
}
```

如果Sensor初始化时，io_domains还未初始化，那么io_domains会使用默认值，如果默认值与实际的硬件电源域不同时，mclk的电压也会不符合预期。此时返回 `-EPROBE_DEFER` 会让Sensor在稍后的启动过程中再尝试probe。

1.1.5 检查Sensor的上电时序是否满足要求

Sensor 的Datasheet中一般会详细描述每路电源的上电顺序及间隔要求，请通过示波器检查是否满足。有一些Sensor的电源vdd在上电时是**没有时间先后要求**的，如 `ov5695`，它的驱动中可能是用 `regulator_bulk` 来管理电源；但有一些是**有先后要求**的，如 `ov2685.c`，它在驱动中是用多个 `regulator`去分别控制，具体如 `avdd_regulator`，`dovdd_regulator`。请根据实际情况选择。

1.2 Sensor驱动中的 `exp_def`、`hts_def`、`vts_def` 默认值是多少

如果有Sensor原厂的联系方式，请联系原厂获取。否则，请从datasheet中查找到对应的寄存器，并从寄存器列表中找到初始化时配置的值即可。以 `ov2685.c` 为例：

```
#define OV2685_REG_VTS                                0x380e

...

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

...

    .vts_def = 0x050e,
```

0x380e与0x380f是vts对应的寄存器，在初始化时配置的值是0x050e，那么vts_def就是0x050e。exp与hts采用默认值，可直接从datasheet中查找。

如果不期望应用程序去调节曝光、帧率时，可以不必要用到exp, hts, vts。一般RAW格式的Sensor需要这三个参数。

1.3 `link_freq`与`pixel_rate`值应该是多少

`link_freq` 指的是MIPI clk的实际频率。**注意不是24M的mclk，而是MIPI dn/dp clk。**优先通过原厂窗口查问，或查找datasheet是否有相关的参数。

一般情况下，`link_freq` 实际值**不会小于**如下公式的计算结果，单位是(Hz)

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

如果实在不知道link_freq的实际值，可以用示波器测量。

`pixel_rate` 指的是每秒传输的像素个数，在 `link_freq` 确定下之后，可用以下公式计算：

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

1.4 怎么才算点亮Sensor

首先需要能认到Sensor id，即i2c的读写不能有异常。这时用 `media-ctl -p -d /dev/media0` 应该能够看到Sensor的具体信息如名称、分辨率等。

其次，上层抓图时，MIPI要能输出数据，且不报MIPI/ISP相关错误，应用层能接收到帧。

1.5 Sensor AVL列表

RGB Sensor AVL 位于

https://redmine.rockchip.com.cn/projects/rockchip_camera_module_support_list/camera。支持列表中显示了sensor模组的详细信息。

如果是其它非RGB Sensor，如YUV Sensor，可以直接查看kernel源码的drivers/media/i2c/目录，其中驱动作者是Rockchip的，驱动都是有调试过的。

2 MIPI/ISP异常相关

Sensor调试初期，比较经常碰到的几类问题是：

1. 没有收到帧数据，也没看到ISP/MIPI有报错
2. 看到log不停打印MIPI错误
3. ISP报PIC_SIZE_ERROR
4. 偶现MIPI错误
5. MIPI不停报错，直至死机

2.1 MIPI需要设置哪些参数

在Sensor与ISP之间MIPI通讯需要设置4个参数，请**务必**确认4个MIPI参数的正确性。

- Sensor输出的分辨率大小
- Sensor输出的图像格式，是YUV或RGB RAW，8-bits、10-bits、或12-bits
- Sensor的MIPI实际输出 link freq
- Sensor使用了几个MIPI lane，这需要在dts中2个位置都配置正确

2.2 没有收到帧数据，也没有看到ISP/MIPI有报错

1. 确认kernel log中有没有关于MIPI的报错，比如用 `dmesg | grep MIPI` 看看有没有出错信息。
 2. 确认kernel log中有没有出现Sensor 的i2c读写失败，如果Sensor 在配置寄存器时失败了，Sensor也可能没有正确初始化并使能输出。
 3. 实际量测下MIPI的clk及data线上有没有信息输出。如果没有，建议从Sensor初始化及硬件方面分析。
 4. 实际量测**有MIPI信号**输出，但没报错也收不到数据
- 请再次检查[2.1 MIPI需要设置哪些参数](#)，
 - 请确认I2C通讯没有错，Sensor的寄存器初始化列表有全部写到Sensor中，
 - 在Sensor驱动中，最后使能MIPI输出的是 `s_stream()`，请确认在这个函数前，特别是 `s_power()`，不要让MIPI信号输出。这是因为在 `s_stream()` 前，MIPI控制器还未实际准备好接收数据，如果在 `s_stream()` 前输出数据，可能导致MIPI协议头SOT信号丢失，
 - 也可以将 Camera Sensor 端 clock lane 由 continue 模式切换到 no continues。

2.3 MIPI报错

2.3.1 MIPI错误信息详细表

针对RK3288/RK3399/RK3368，错误信息表如下：

错误位 (Bit)	简称	描述
25	ADD_DATA_OVFLW	additional data fifo overflow occurred
24	FRAME_END	正常收到一帧，不是错误
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 occurred, 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

针对RK3326/PX30/RK1808，3个错误信息表如下：

ERR1 错误位 (Bit)	简称	描述
28	ERR_ECC	ECC ERROR
27:24	ERR_CRC	CRC ERROR
23:20	ERR_FRAME_DATA	Frame 传输完毕，但至少包含一个CRC错误
19:16	ERR_F_SEQ	Frame Number 不连续不符合预期
15:12	ERR_F_BNDRY	Frame start与Frame end没有匹配
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 错误位 (Bit)	简称	描述
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

常见的错误分析如下小章节。

2.3.2 如何处理SOT/SOT_SYNC错误

SOT信号需要符合 **MIPI_D-PHY_Specification**。如果需要深入分析，请直接从网上搜索该pdf文档，并建议重要参考：

- High-Speed Data Transmission
- Start-of-Transmission Sequence
- HS Data Transmission Burst
- High-Speed Clock Transmission
- Global Operation Timing Parameters

但一般来讲，Sensor如果有在其它平台调通过，那么不符合MIPI 协议的可能性比较小，建议客户：

- 首先向Sensor厂家确认该Sensor是否有实际成功使用过MIPI接口传输数据，
- **再次确认link_freq是否正确**。因为SOT时序中的Ths-settle需要在MIPI接收端配置正确，所以link_freq很关键，
- 如果使用了多lane，看Sensor原厂有没有办法修改成1 lane传输。

2.3.3 如何处理CRC/Checksum(CS)、ECC/ECC1/ECC2错误

出现了ECC错误，CS检验错误，说明数据在传输时不完整。建议：

- **优先排查硬件信号**，
- 如果使用了多lane，看Sensor原厂有没有办法修改成1 lane传输。因为多lane之间没有同步好，也有可能出现ECC错误。

2.3.4 如何处理ERR_PROTOCOL/ERR_F_BNDRY错误

该错误说明没有收到预期的EOT/SOT。SOT,EOT应该成对匹配出现。建议实测波形检查。

2.3.5 能正常收帧，但偶现MIPI错误

如果是MIPI错误，参考前面的错误描述。与信号相关建议从硬件信号上分析。

特别地，如果MIPI错误只在刚开始抓图时有，有可能是Sensor在上电的过程中MIPI信号有输出但并不符合协议，从而报错。

这种情况下，可以尝试按如下流程修改：

- 将完整的Sensor寄存器的初始化放到 `s_power()` 中。
因为此时MIPI接收端尚未开始接收数据，会忽略所有数据。
- 在 `s_power()` 函数的最后，关闭sensor的输出，即相当于调用了 `stop_stream()`
- 在 `start_stream()` 与 `stop_stream()` 中，仅打开或关闭MIPI的输出。

2.3.6 报很多MIPI错误甚至死机

这可能是[2.3.5 能正常收帧，但偶现MIPI错误](#)的更坏的情况。

碰到过这样的现象，其原因是MIPI信号不符合要求，而且MIPI接收端某些错误是电平中断，导致中断风暴并最终死机。

可以尝试按[2.3.5 能正常收帧，但偶现MIPI错误](#)的方法看是否有效。

2.4 如何处理ISP PIC_SIZE_ERROR

Picture size error是ISP级的错误，它提示未接收到预期的行数，列数。因此从各级的分辨率大小检查。

如果前级（即MIPI）有报错，应该先解决MIPI错误。

请从如下几点检查：

- DDR频率是否太小。当DDR频率太低时，响应速度不够时，也会出现该错误。尝试将DDR定频到最高频率看还会不会出错：

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

- 整个ISP链路中，有没有出现后级比前级的分辨率还大的情况。可以用 `media-ctl -p -d /dev/media0` 去查看拓扑结构。
分辨率应该要满足 `Sensor == MIPI_DPHY >= isp_sd input >= isp_sd output`。如果您没有手动修改过，默认应该是满足这个条件的。
- Sensor的输出分辨率大小是否正确。尝试在驱动代码中将分辨率强制改小。比如ov7251.c中默认分辨率是640x480，

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

将width, height都改小些，比如320x240，寄存器的配置不用改。这是为了确认Sensor的配置大小不会超过实际输出的大小。

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

3 获取图像相关

这部分主要涉及与抓图相关的常见问题。

3.1 有哪些方式可以抓图

RKISP及RKCIF驱动支持v4l2接口，获取图像可以使用：

- v4l-utils包中的v4l2-ctl工具获取图像。在调试过程中，建议首先使用该工具检验能否成功出图。v4l2-ctl抓图保存成文件，它不能解析图像并显示出来。如需要解析，Ubuntu/Debian环境下可以使用mplayer，Windows下可以使用如7yuv等。
对v4l2-ctl, mplayer工具的详细说明，请参考《Rockchip_Developer_Guide_Linux_Camera_CN.pdf》。v4l2-ctl也自带详细的 `v4l2-ctl --help` 文档。
- RK提供的Linux SDK中包含qcamera应用app
直接打开桌面上的camera app，选择video设备并预览。
该app是一个基于qt的demo，仅用于参考。客户项目中，建议自行开发app。
- 使用gststreamer的v4l2src plugin可以从/dev/video设备中获取图像并显示在屏幕上
RK提供的Linux SDK会在目录/rockchip_test/camera/下包含一些脚本，请先参考。
特别需要注意： RK先后提供过多个版本的gststreamer isp plugin，如rkisp, rkiv4l2src，都已经 **不再继续支持**。请直接使用gststreamer自带的**v4l2src** plugin。主要有两点原因：
 - 3A不再需要在rkisp或rkiv4l2src中调整。3A部分请直接参考[4.3A相关](#)
 - rkisp驱动的v4l2接口更加标准化
- Debian系统中使用如vlc等开源工具
通过apt 安装vlc后，可以使用如下命令显示Camera图像于屏幕上：

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

注意，需要video用户组的权限，或者root超级用户权限。

3.2 抓到的图颜色不对，亮度也明显偏暗或偏亮

需要根据Sensor分情况：

- Sensor是RAW RGB的输出，如RGGB、BGGR等，需要3A正常跑起来。可以参考[4.3A相关](#)
3A 确认正常在跑时，请再次检查解析/显示图像时使用的格式是否正确，uv分量有没有弄反。
- Sensor是yuv输出，或RGB如RGB565、RGB888，此时ISP处于bypass状态，
 - 如果颜色不对，请确认sensor的输出格式有没有配置错误，uv分量有没有弄反。确认无误时，建议联系Sensor原厂
 - 如果亮度明显不对，请联系Sensor原厂

3.3 什么是ISP的拓扑结构(topology, 链路结构)，如何使用media-ctl命令

RKISP或RKCIF可以接多个的Sensor，分时复用；同时RKISP还有多级的裁剪功能。因此用链接的方式将各个节点连接，并可通过media-ctl分别配置参数。关于media-ctl的使用，在《Rockchip_Developer_Guide_Linux_Camera_CN.pdf》文档中有较完整的描述。

3.3.1 一个ISP怎样接多个Sensor

可以接多个Sensor，但只能分时复用。通过配置dts，将多个Sensor链接到MIPI DPHY后，可通过media-ctl切换Sensor。

3.4 抓取RAW图是否与原图完全一致

当ISP以bypass模式获取Sensor RAW图（如RGB, BGGR）时，需要8bit对齐，不足8bit会低位填充0，即

- 如果是8bit, 16bit的原图，应用获取到的是原图，没有填充
- 如果是10bit, 12bit的原图，会每个像素低位补0到16bit

只有MP对应的video设备可以出RAW图，SP是不能支持RAW图输出的。

3.5 ISP怎样双路(MP, SP)同时输出

RKISP有SP, MP两路输出，即Sensor出来一张图像，SP, MP可以分别对该图像做裁剪、格式转换，并可同时输出。

SP, MP具体不同的视频处理能力，详细请参考《Rockchip_Developer_Guide_Linux_Camera_CN.pdf》。

只有当SP, MP都输出RGB或YUV时才可以同时输出。如果MP输出RAW图，那么SP不可以出图。

3.6 ISP是否具有放大功能

硬件上有该功能，但不建议使用，驱动中也是默认关闭该功能。

3.7 ISP是否具有旋转功能

没有。如果需要使用旋转功能，建议：

- 如果是flip, mirror，首先查看Sensor是不是有该功能，如果有，直接使用。这样效率最高
- 如果无法使用Sensor flip, mirror，考虑使用RGA模块，它的代码及demo位于external/linux-rga/目录，且有相关文档位于docs/目录下

3.8 怎样抓灰度(GREY)图

只要ISP可以输出YUV、或者Sensor输出是Y8灰度图时，应用程序总是可以使用V4L2_PIX_FMT_GREY(FourCC为GREY)格式直接获取图像。

3.9 RGB图支持哪些格式

首先，只有SP这一路可以支持RGB输出，格式为：V4L2_PIX_FMT_XBGR32, V4L2_PIX_FMT_RGB565。其中XBGR32（对应的FourCC为XR24）是包含R、G、B、X四个分量，其中X分量总是为0。

不支持RGB888，即24bit的格式输出。

3.10 无屏板卡如何快速预览

SDK中external/uvic_app/目录提供了将板卡模拟成uvic camera的功能，请参考该目录中的说明文件及代码，将板卡连接到PC机后可识别出usb camera，并可预览图像。

3.11 如何区分SP与MP

可通过 media-ctl -p -d /dev/media0 (如有多个media设备，也尝试下/dev/media1, /dev/media2) 去查看拓扑结构，如下截取部分输出：

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

```

- entity 2: rkisp1_mainpath (1 pad, 1 link)           //表示该entity是
MP(MainPath)
    type Node subtype V4L flags 0
    device node name /dev/video1                     //对应的设备节点
是/dev/video1
    pad0: Sink
        <- "rkisp1-isp-subdev":2 [ENABLED]

- entity 3: rkisp1_selfpath (1 pad, 1 link)           //表示该entity是
SP(SelfPath)
    type Node subtype V4L flags 0
    device node name /dev/video2                     //对应的设备节点
是/dev/video2
    pad0: Sink
        <- "rkisp1-isp-subdev":2 [ENABLED]

...

```

少数情况下如果没有media-ctl命令，可以通过/sys/节点查找，如：

```

# grep '' /sys/class/video4linux/video*/name
/sys/class/video4linux/video0/name:stream_cif
/sys/class/video4linux/video1/name:rkisp1_mainpath    # MP节点对应/dev/video1
/sys/class/video4linux/video2/name:rkisp1_selfpath    # SP节点对应/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 3A相关

如果Sensor需要3A tuning，如Sensor输出格式RGGB, BGGR等这样的RAW BAYER RGB格式，那么需要RKISP提供图像处理。

根据camera_engine_rkisp版本的不同，3A处理方式有差别。建议尽量将camera_engine_rkisp升级到最新的版本。

请首先确认该模组是否在支持列表中，

- 已经在支持列表中的，在external/camera_engine_rkisp/iqfiles/目录下会有一份对应的xml文件
- 否则请向业务窗口发起模组调试申请

4.1 如何确认camera_engine_rkisp的版本

- 从源码中查看

```
# grep CONFIG_CAM_ENGINE_LIB_VERSION interface/rkisp_dev_manager.h
define CONFIG_CAM_ENGINE_LIB_VERSION "v2.2.0" # 输出的v2.2.0是
librkisp.so的版本号
```

- 从运行时log看

```
# persist_camera_engine_log=0x4000 rkisp_3A_server --mmedia=/dev/media1 |
grep "CAM ENGINE LIB VERSION"
CAM ENGINE LIB VERSION IS v2.2.0 # 输出的v2.2.0是
librkisp.so的版本号
```

如果版本号低于v2.2.0，请考虑升级到v2.2.0甚至更新的版本

4.1.1 如何确认camera_engine_rkisp所需要的rkisp kernel驱动的版本号

camera_engine_rkisp对kernel驱动版本有要求，需要保证rkisp驱动足够新。

- 从kernel源码中查看ISP驱动版本

```
# grep RKISP1_DRIVER_VERSION drivers/media/platform/rockchip/isp1/version.h
define RKISP1_DRIVER_VERSION KERNEL_VERSION(0, 1, 0x5) # 输出的v0.1.5是rkisp驱
动的版本号
```

- 从kernel log中查看ISP驱动版本

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

4.2 如何升级camera_engine_rkisp

包含有三部分

1. camera_engine_rkisp本身

位于SDK的external/camera_engine_rkisp目录，直接通过git或repo工具可以更新。可以仅更新该目录而不影响其它SDK中的目录。

2. kernel根据camera_engine_rkisp的需要相应升级

在external/camera_engine_rkisp目录下通过查看 `git log`，可以找到它所需要的kernel rkisp驱动的版本号。例如：

```
# 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                # 所需要的kernel驱动版本为v0.1.5

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

3. buildroot中camera_engine_rkisp的编译脚本

位于buildroot/package/rockchip/camera_engine_rkisp目录下，如果不方便更新整个buildroot，可以只单独更新这个目录。

4.3 如何确认3A是否正在工作

在确认camera_engine_rkisp已经是v2.2.0版本或以上之后。通过抓取图像，查看图像的色彩及曝光是否正常。

同时，通过查看后台是否有rkisp_3A_server进程在执行，如下：

```
# 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
```

可以看到进程号706即是rkisp_3A_server。

4.3.1 没有看到rkisp_3A_server进程

- 首先确认/usr/bin/rkisp_3A_server可执行文件是否存在，如不存在，请检查camera_engine_rkisp版本及编译。
- 查看/var/log/syslog中是否有rkisp_3A相关的错误，如有看具体错误是什么，是否Sensor模组对应的xml没有找到，或不匹配。
- 在shell中执行 `rkisp_3A_server --mmedia=/dev/media0`（如有多个/dev/media设备，选择/dev/video对应的那一个），从另一个shell中抓图。获取rkisp_3A_server对应的错误信息

4.3.2 rkisp_3A_server是如何启动的

Linux SDK中，rkisp_3A_server由脚本/etc/init.d/S40rkisp_3A 启动并在后台执行。
如果/etc/init.d/S40rkisp_3A文件未找到，检查camera_engine_rkisp的版本及buildroot package编译脚本。

4.3.3 如何确定Sensor iq配置文件(xml)文件名及路径

Sensor iq文件由三部分组成，

- Sensor Type, 比如ov5695, imx327
- Module Name, 在dts中定义，比如rk3326/px30 rk evb板上，该名称为"TongJu"
`rockchip,camera-module-name = "TongJu";`
- Module Lens Name, 在dts中定义，比如以下的"CHT842-MD":
`rockchip,camera-module-lens-name = "CHT842-MD";`

那么上例中的iq文件名为：ov5695_TongJu_CHT842-MD.xml, 存放在/etc/iqfiles/目录下。注意大小写有区分。

4.4 怎样手动曝光

需要手动曝光的情况下，rkisp_3A_server进程必须先退出。然后可参考rkisp_demo.cpp程序或librkisp_api.so的源码。

4.5 如何打开librkisp的log

通过设置环境变量persist_camera_engine_log，其对应的位表示如下：

```
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
```

例如，打开ISP及AWB的debug log：

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

5 应用开发相关

5.1 C语言参考demo

- RK提供的Linux SDK中包含rkisp_demo工具及源码
rkisp_demo是一个简单的工具，可以用于获取图像。类似于v4l2-ctl工具，rkisp_demo也不能显示图像，它主要是提供源码供参考。
源码位于external/camera_engine_rkisp/apps目录下，如果您的代码较旧，源码位于external/camera_engine_rkisp/tests目录下。
- RK提供的Linux SDK中包含rkisp_api.so动态链接库及源码，可以基于此做修改或直接使用C语言开发程序
源码位于external/camera_engine_rkisp/apps目录下。如果您的代码较旧找不到该目录，请更新。

5.2 什么是DMA buffer，有什么好处

DMA buffer 是一片由驱动分配的内存，该buffer 可以在多个内核模块之间共享，从而减少内存拷贝。特别是对图像处理时能优化性能，减小DDR负载。

例如，camera图像需要由mpp编码：



RKISP可以接受来自其它模块（如MPP, RGA）的DMA buffer，也可以分配内存，并导出DMA buffer给其它模块使用。

细节可以参考librkisp_api.so库的源码。