**I SP20**

应用开发参考

文件标识：RK-SM-YF-366

发布版本：V1.1.1日期：2020-10-14

文件密级：□绝密 □秘密 □内部资料 ■公开

免责声明

本文档按“现状”提供，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。

© 2020

版权所有 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。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址： 福建省福州市铜盘路软件园A区18号网址： [w ww.rock-chips.com](http://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： [f ae@rock-chips.com](mailto:fae@rock-chips.com)

前言

概述

本文旨在介绍应用如何获取**Camera**数据流以及**RkAiq 3A Server**独立进程。

产品版本

|  |  |
| --- | --- |
| 芯片名称 | 内核版本 |
| RV1109/RV1126 | Linux-4.19 |

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 | Zack Zeng | 2020-06-10 | 初始版本 |
| V1.1.0 | CWW | 2020-10-02 | 修正文档路径 |
| V1.1.1 | Ruby Zhang | 2020-10-14 | 格式修正 |

目录

**ISP20**

应用开发参考

1. 概述
   1. 功能概述
   2. 数据流概述
2. 数据流获取
   1. 基于RKMEDIA获取数据流
      1. bypass节点数据流获取
      2. 三路scale down节点数据流获取
      3. FBC格式数据获取
   2. 基于v4l2-utils获取数据流
3. RkAiq 3A Server独立进程
   1. 如何确认RkAiq版本
      1. 如何确认RkAiq所匹配的ISP20驱动版本号
   2. 如何确认3A是否正常工作
      1. 没有看到ispserver进程
      2. 如何确定Sensor IQ配置文件(xml)文件名及路径
4. 缩略语

# 1 .

## 概述

### 1.1

#### 功能概述

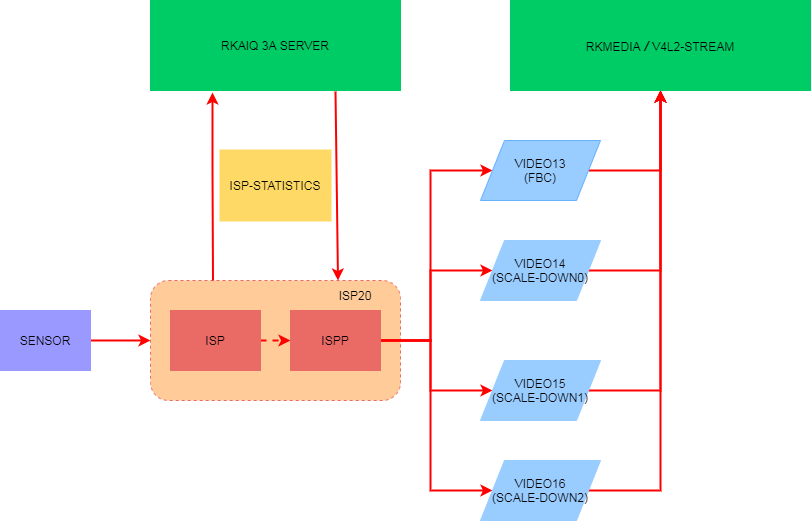


图1 数据流框图

Camera数据流通路如图1所示，Camera数据经过ISP20采集，ISP20再输出经过一系列图像处理算法后的数据，RkAiq不断从ISP20获取统计数据，并经过3A等算法生成新的参数反馈给ISP20，

RkAiq具体实现可以参考**docs/RV1126\_RV1109/Camera**目录中的文档

《Rockchip\_Development\_Guide\_ISP2x\_CN\_v1.2.0.pdf》。本文着重介绍应用层如何获取经过ISP20处理后的数据流。

**1.2** 数据流概述

|  |  |  |  |
| --- | --- | --- | --- |
| **entity name** | **video id** | **max width** | **support output fmt** |
| rkispp\_m\_bypass | /dev/video13 | 不支持设置分辨率，不支持缩放 | NV12/NV16/YUYV/**FBC0**/**FBC2**/ |
| rkispp\_scale0 | /dev/video14 | **max width: 3264,**  最大  **8**  支持 倍缩放 | NV12/NV16/YUYV |
| rkispp\_scale1 | /dev/video15 | **max width: 1280,**  最大  **8**  支持 倍缩放 | NV12/NV16/YUYV |
| rkispp\_scale2 | /dev/video16 | **max width: 1280,**  最大  **8**  支持 倍缩放 | NV12/NV16/YUYV |

[^注]：**video id** 节点不是固定的，可以使用media-ctl查看对应的节点。

表1 四路数据流

ISP20可以输出四路数据流，如表1所示，entity name及对应的设备节点id可以通过命令：media-ctl -p -d

/dev/media1（如有多个media设备，也尝试下/dev/media2）查看media设备的拓扑结构，如下截取部分输出：

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

# media-ctl -p -d /dev/media1

...

- entity 5: rkispp\_m\_bypass (1 pad, 1 link) //表示该entity是bypass type Node subtype V4L flags 0

device node name /dev/video13

pad0: Sink

//对应的设备节点id为/dev/video13

<- "rkispp-subdev":2 [ENABLED]

- entity 9: rkispp\_scale0 (1 pad, 1 link) //表示该entity是scale0

type Node subtype V4L flags 0

device node name /dev/video14 //对应的设备节点id为/dev/video14 pad0: Sink

<- "rkispp-subdev":2 [ENABLED]

- entity 13: rkispp\_scale1 (1 pad, 1 link) //表示该entity是scale1 type Node subtype V4L flags 0

device node name /dev/video15 //对应的设备节点id为/dev/video15

pad0: Sink

<- "rkispp-subdev":2 [ENABLED]

- entity 17: rkispp\_scale2 (1 pad, 1 link) //表示该entity是scale2 type Node subtype V4L flags 0

device node name /dev/video16 //对应的设备节点id为/dev/video16

pad0: Sink

<- "rkispp-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：

1. # grep '' /sys/class/video4linux/video\*/name
2. /sys/class/video4linux/video0/name:rkisp\_mainpath
3. /sys/class/video4linux/video1/name:rkisp\_selfpath
4. /sys/class/video4linux/video10/name:rkisp-input-params
5. /sys/class/video4linux/video11/name:rkisp-mipi-luma

6 /sys/class/video4linux/video12/name:rkispp\_input\_image 7

8

10

/sys/class/video4linux/video13/name:rkispp\_m\_bypass

应/dev/video13

/sys/class/video4linux/video14/name:rkispp\_scale0

应/dev/video14

/sys/class/video4linux/video15/name:rkispp\_scale1

应/dev/video15

/sys/class/video4linux/video16/name:rkispp\_scale2

应/dev/video16

//bypass节点对

//scale0节点对

9

//scale1节点对

//scale2节点对

11

12

13

14

15

16

17

18

19

20

/sys/class/video4linux/video17/name:rkispp\_input\_params

/sys/class/video4linux/video18/name:rkispp-stats

/sys/class/video4linux/video2/name:rkisp\_rawwr0

/sys/class/video4linux/video3/name:rkisp\_rawwr1

/sys/class/video4linux/video4/name:rkisp\_rawwr2

/sys/class/video4linux/video5/name:rkisp\_rawwr3

/sys/class/video4linux/video6/name:rkisp\_rawrd0\_m

/sys/class/video4linux/video7/name:rkisp\_rawrd1\_l

/sys/class/video4linux/video8/name:rkisp\_rawrd2\_s

/sys/class/video4linux/video9/name:rkisp-statistics

# 2 .

## 数据流获取

### RKMEDIA

#### 基于 获取数据流

RKMEDIA是RockChip Linux平台的多媒体库，详情请阅读**docs/RV1126\_RV1109/Multimedia**目录中的文档《Rockchip\_Instructions\_Linux\_Rkmedia\_CN.pdf》，本文着重介绍摄像头采集接口部分。

摄像头采集接口仅支持V4L2，源码参考

范例：

**external/rkmedia/examples/uintTest/stream/camera\_capture\_test.cc** （可能默认生成的固件里没有此可执行bin，需要到PC上生成的路径手动push到板端），使用命令查看使用方法：

1 # ./camera\_cap\_test -h

##### bypass

节点数据流获取

bypass这路数据流比较特殊，

其不支持设置分辨率

media-ctl查看拓扑结构确定ISP输入的分辨率。

，其输出分辨率由ISP输入的分辨率决定，可以通过

1

2

3

4

5

6

7

8

9

10

11

12

# media-ctl -p -d /dev/media1

...

-

entity

29:

rkispp-subdev (4

type V4L2 subdev device node name

pads, 7 links)

subtype Unknown flags 0

/dev/v4l-subdev0

pad0:

pad1:

Sink

[fmt:YUYV8\_2X8/2688x1520 field:none crop.bounds:(0,0)/2688x1520 crop:(0,0)/2688x1520]

<- "rkispp\_input\_image":0 [] Sink

<- "rkispp\_input\_params":0 [ENABLED]

13

14

15

16

17

18

19

20

21

pad2: Source

[fmt:YUYV8\_2X8/2688x1520 field:none]

->

->

->

->

"rkispp\_m\_bypass":0 [ENABLED]

"rkispp\_scale0":0 "rkispp\_scale1":0

"rkispp\_scale2":0

[ENABLED] [ENABLED]

[ENABLED]

pad3:

Source

-> "rkispp-stats":0 [ENABLED]

...

如上所示，bypass这路输出的分辨率为2688x1520。因此您可以运行如下命令来获取bypass节点的数据流：

1 camera\_cap\_test -i /dev/video13 -o output.yuv -w 2688 -h 1520 -f image:nv12

SDK video id

此外不同版本的 其对应的 设备 可能会不一样，但是

entity name

是唯一的

，因此也支持用

entity name代替video设备id获取数据流，命令如下：

1 camera\_cap\_test -i rkispp\_m\_bypass -o output.yuv -w 2688 -h 1520 -f image:nv12

##### scale down

###### 三路 节点数据流获取

三路scale down节点支持缩放，每一路支持的最大分辨率如 1.2 节 数据流概述中的表1所示，同时也支持

entity name和/dev/videoX来抓取数据流，以scale0为例：

1 camera\_cap\_test -i /dev/video14 -o output.yuv -w 2688 -h 1520 -f image:nv12 2

3 camera\_cap\_test -i rkispp\_scale0 -o output.yuv -w 2688 -h 1520 -f image:nv12

scale

建议三路 输出的分辨率相加后未超过主码流的分辨率

##### FBC

###### 格式数据获取

ISP20支持输出FBC格式数据， **rkispp\_m\_bypass**

仅 （

/dev/video13

）支持输出

**FBC**

格式数据

，FBC格式

数据有两种，FBC0与FBC2。其区别如下：以sensor os04a10为例：

1 # v4l2-ctl -d /dev/video13 --set-fmt-

video=width=2688,height=1520,pixelformat='FBC0' --verbose

2

3

4

5

6

7

8

9

10

11

Format

Video Capture Multiplanar: Width/Height

Pixel Format

Field

Number

Flags

of planes

Colorspace

Transfer Function

:

:

:

:

:

:

:

2688/1520

'FBC0' (Rockchip yuv420sp fbc encoder) None

1

YCbCr/HSV Encoding:

Quantization

:

Default Default Default

Full Range

12

13

14

Plane 0

:

Bytes per Line : 2688

Size Image

: 6386688

1

2

3

4

5

6

7

8

9

10

11

12

13

14

# v4l2-ctl -d /dev/video13 --set-fmt- video=width=2688,height=1520,pixelformat='FBC2' --verbose

Format Video Capture Multiplanar:

Width/Height Pixel Format Field

Number of planes Flags

Colorspace

: 2688/1520

: 'FBC2' (Rockchip yuv422sp fbc encoder)

: None

: 1

:

: Default

Transfer Function : Default YCbCr/HSV Encoding: Default Quantization

Plane 0

: Full Range

:

Bytes per Line : 2688

Size Image

: 8429568

其获取方法与其他格式相似，只是将格式改为FBC0/FBC2即可，如下所示：

1. camera\_cap\_test -i rkispp\_m\_bypass -o output.yuv -w 2688 -h 1520 -f image:fbc0
2. camera\_cap\_test -i rkispp\_m\_bypass -o output.yuv -w 2688 -h 1520 -f image:fbc2

或者

1. camera\_cap\_test -i /dev/video13 -o output.yuv -w 2688 -h 1520 -f image:fbc0
2. camera\_cap\_test -i /dev/video13 -o output.yuv -w 2688 -h 1520 -f image:fbc2

FBC

注意：此分辨率也不支持设置，建议主码流采用 格式数据（对带宽占用比较友好）。

### 2.2 v4l2-utils

#### 基于 获取数据流

ISP20驱动支持V4L2接口，因此获取数据流可以用v4l-utils包中的v4l2-ctl工具，

在调试过程中，建议先

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docs/Linux/Multimedia/camera/目录中的文档

《Rockchip\_Developer\_Guide\_Linux\_Camera\_CN.pdf》。v4l2-ctl也自带有详细的v4l2-ctl --help 文档。下面是一个简单的抓图命令：

1. v4l2-ctl -d /dev/video13 --set-ctrl="exposure=234,analogue\_gain=76" \
2. --set-selection=target=crop,top=0,left=0,width=2688,height=1520 --set-fmt- video=width=2688,height=1520,pixelformat=NV12 \
3. --stream-mmap=4 --stream-to=/tmp/output.nv12 --stream-count=1 --stream-poll

# . RkAiq 3A Server

## 独立进程

当Sensor输出RAW BAYER RGB格式时，如RGGB,BGGR,GBRG,GRBG等，需要ISP20提供一系列图像处理算法来优化图像效果，此时需要RkAiq模块介入。

SDK提供了一种3A独立进程的方式（ispserver）集成了RkAiq库librkaiq.so，旨在用第 2 章 数据流获取中的方法获取数据流时，能够得到具有ISP调试效果的图像。

~~Isps~~erver具体实现可以阅读**docs/RV1126\_RV1109/camera**目录中的文档

《Rockchip\_RV1109\_RV1126\_Developer\_Guide\_Linux\_Ispserver\_CN.pdf》，文档有具体介绍。

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\_rkaiq/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

### RkAiq

如何确认 版本

从源码中查看

1

2

# grep RK\_AIQ\_VERSION RkAiqVersion.h

#define RK\_AIQ\_VERSION "v0.1.6"

# 输出的v0.1.6是librkaiq.so的版本号

##### 3.1.1 RkAiq ISP20

如何确认 所匹配的 驱动版本号

从kernel源码中查看ISP以及ISPP驱动版本

1. # grep RKISP\_DRIVER\_VERSION drivers/media/platform/rockchip/isp/version.h
2. #define RKISP\_DRIVER\_VERSION KERNEL\_VERSION(0, 1, 0x5) # 输出的v0.1.5是rkisp驱动的版本号

3

1. # grep RKISPP\_DRIVER\_VERSION drivers/media/platform/rockchip/ispp/version.h
2. #define RKISPP\_DRIVER\_VERSION KERNEL\_VERSION(0, 1, 0x0) # 输出的v0.1.0是

rkispp驱动的版本号

从kernel log中查看ISP以及ISPP驱动版本

1

2

3

4

5

# dmesg | grep "rkisp driver version"

[

0.332831] rkisp ffb50000.rkisp: rkisp driver version: v00.01.05

# dmesg | grep "rkispp driver version"

[

0.340370] rkispp ffb60000.rkispp: rkispp driver version: v00.01.00

### 3A

如何确认 是否正常工作

如果产品带屏幕，可以直接预览，如果是IPC类产品，可以打开网页预览，针对没有屏幕也不是IPC类产品，可以通过第 2 章 数据流获取中的方法获取数据流，来确认AE, AWB等是否正常。

同时，通过查看后台是否有ispserver进程在执行，如下：

1

2

3

4

5

# ps -ef | grep ispserver

705 root

746 root

299m S

2408 S

ispserver

grep ispserver

# pidof ispserver

705

可以看到进程号705即是ispserver。

##### ispserver

###### 没有看到 进程

查看/var/log/syslog中是否有rkaiq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ispserver ，从另一个shell中抓图。获取ispserver对应的错误信息。

##### Sensor IQ (xml)

###### 如何确定 配置文件 文件名及路径

Sensor iq文件由三部分组成，

Sensor Type, 比如os04a10,imx347

Module Name, 在dts中定义，比如RV1126/RV1109 evb板上，该名称为"CMK-OT1607-FV1"

rockchip,camera-module-name = "CMK-OT1607-FV1";

rockchip,camera-module-

Module Lens Name, 在dts中定义，比如以下的"M12-4IR-4MP-F16":

lens-name = "M12-4IR-4MP-F16";

那么上例中的iq文件名为：os04a10\_CMK-OT1607-FV1\_M12-4IR-4MP-F16.xml, 如果有定义oem分区，则默认存放在/oem/etc/iqfiles/目录下，如果没有定义oem分区，则存放在/etc/iqfiles/,注意大小写有区分。

1. **.** 缩略语

|  |  |
| --- | --- |
| 缩写 | 全称 |
| 3A | AWB, AE, AF |
| AE | Auto Exposure |
| AF | Auto Focus |
| AWB | Auto White Balance |
| FBC | Frame Buffer Compressed |
| FBC0 | Rockchip yuv420sp fbc encoder |
| FBC2 | Rockchip yuv422sp fbc encoder |
| RkAiq | Rockchip Automatical Image Quality |
| IQ | Image Quality |
| ISP | Image Signal Process |
| ISPP | Image Signal Post Process |