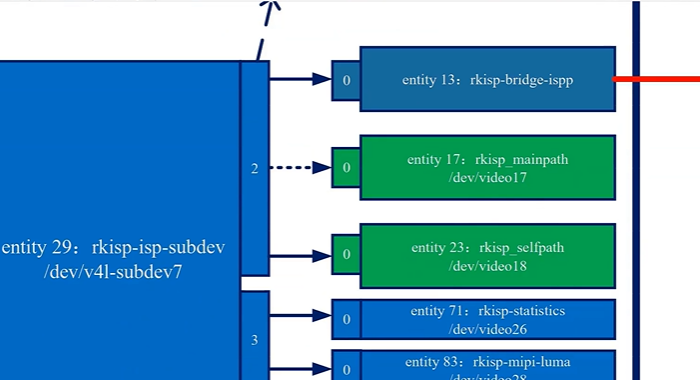
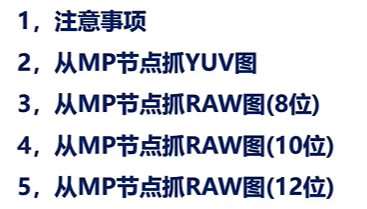
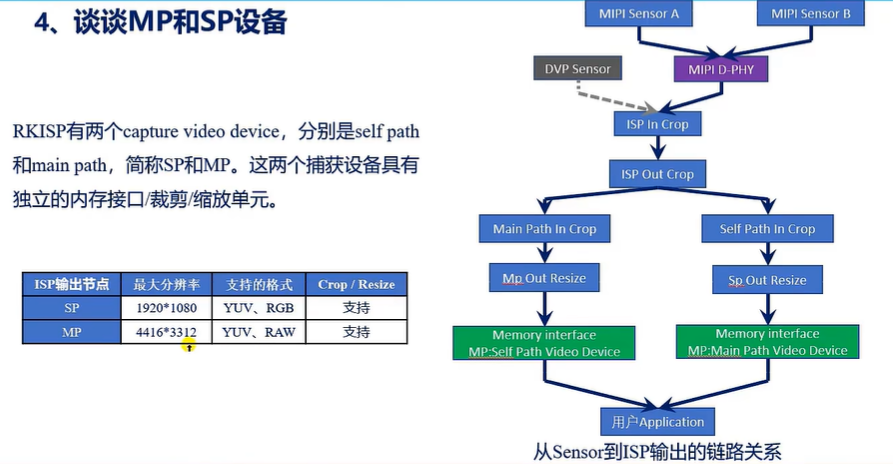
从ISP节点抓取YUV/RAW图。

这里提到的ISP节点就是前面提到的MP（mainpath）和SP节点（selfpath）。我们说的ISP节点就是指这两个节点：

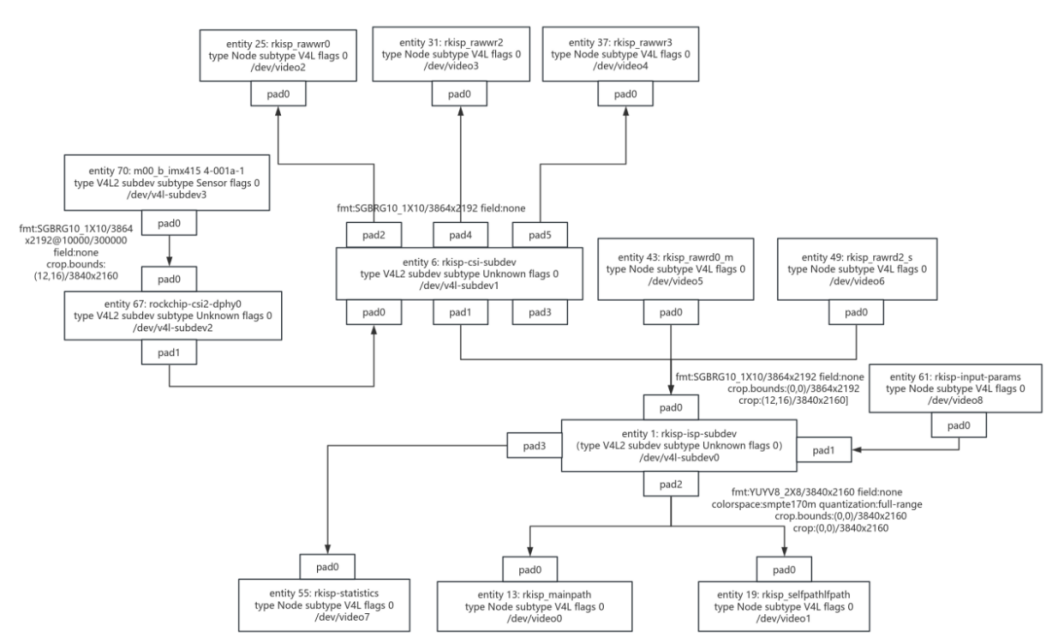


本节课程内容：





/\*



\*/

1. 注意事项：

1、IMX415或者IMX335摄像头要插在MIPI CSI1上

2、不能开启ispserver

3、只能从MP节点抓图

4、MP支持YUV、RGB格式的图像

5、 MP最大分辨率能支持4416\*3312，但是IMX415出图最大是3840\*2160，IMX335最大出图是2592\*1944

6、MP支持对图像进行裁剪缩放操作，所以，也可以从MP节点捕捉其它分辨率的图像。

7、所设置的分辨率，宽度尽量满足64对齐，高度尽量设置为偶数

8、抓取RAW图时，需要设置媒体总线上的像素格式。

9、注意紧凑型和非紧凑型数据存储格式

/\*

在开发板执行命令：

media-ctl --known-mbus-fmts 查看当前系统中支持哪些媒体总线像素格式:

root@ATK-DLRK356X:/# media-ctl --known-mbus-fmts

RGB444\_1X12 0x1016

RGB444\_2X8\_PADHI\_BE 0x1001

RGB444\_2X8\_PADHI\_LE 0x1002

RGB555\_2X8\_PADHI\_BE 0x1003

RGB555\_2X8\_PADHI\_LE 0x1004

RGB565\_1X16 0x1017

BGR565\_2X8\_BE 0x1005

BGR565\_2X8\_LE 0x1006

RGB565\_2X8\_BE 0x1007

RGB565\_2X8\_LE 0x1008

RGB666\_1X18 0x1009

RBG888\_1X24 0x100e

RGB666\_1X24\_CPADHI 0x1015

RGB666\_1X7X3\_SPWG 0x1010

BGR888\_1X24 0x1013

BGR888\_3X8 0x101b

GBR888\_1X24 0x1014

RGB888\_1X24 0x100a

RGB888\_2X12\_BE 0x100b

RGB888\_2X12\_LE 0x100c

RGB888\_3X8 0x101c

RGB888\_1X7X4\_SPWG 0x1011

RGB888\_1X7X4\_JEIDA 0x1012

ARGB8888\_1X32 0x100d

RGB888\_1X32\_PADHI 0x100f

RGB101010\_1X30 0x1018

RGB121212\_1X36 0x1019

RGB161616\_1X48 0x101a

Y8\_1X8 0x2001

UV8\_1X8 0x2015

UYVY8\_1\_5X8 0x2002

VYUY8\_1\_5X8 0x2003

YUYV8\_1\_5X8 0x2004

YVYU8\_1\_5X8 0x2005

UYVY8\_2X8 0x2006

VYUY8\_2X8 0x2007

YUYV8\_2X8 0x2008

YVYU8\_2X8 0x2009

Y10\_1X10 0x200a

Y10\_2X8\_PADHI\_LE 0x202c

UYVY10\_2X10 0x2018

VYUY10\_2X10 0x2019

YUYV10\_2X10 0x200b

YVYU10\_2X10 0x200c

Y12\_1X12 0x2013

UYVY12\_2X12 0x201c

VYUY12\_2X12 0x201d

YUYV12\_2X12 0x201e

YVYU12\_2X12 0x201f

Y14\_1X14 0x202d

UYVY8\_1X16 0x200f

VYUY8\_1X16 0x2010

YUYV8\_1X16 0x2011

YVYU8\_1X16 0x2012

YDYUYDYV8\_1X16 0x2014

UYVY10\_1X20 0x201a

VYUY10\_1X20 0x201b

YUYV10\_1X20 0x200d

YVYU10\_1X20 0x200e

VUY8\_1X24 0x2024

YUV8\_1X24 0x2025

UYYVYY8\_0\_5X24 0x2026

UYVY12\_1X24 0x2020

VYUY12\_1X24 0x2021

YUYV12\_1X24 0x2022

YVYU12\_1X24 0x2023

YUV10\_1X30 0x2016

UYYVYY10\_0\_5X30 0x2027

AYUV8\_1X32 0x2017

UYYVYY12\_0\_5X36 0x2028

YUV12\_1X36 0x2029

YUV16\_1X48 0x202a

UYYVYY16\_0\_5X48 0x202b

SBGGR8\_1X8 0x3001

SGBRG8\_1X8 0x3013

SGRBG8\_1X8 0x3002

SRGGB8\_1X8 0x3014

SBGGR10\_ALAW8\_1X8 0x3015

SGBRG10\_ALAW8\_1X8 0x3016

SGRBG10\_ALAW8\_1X8 0x3017

SRGGB10\_ALAW8\_1X8 0x3018

SBGGR10\_DPCM8\_1X8 0x300b

SGBRG10\_DPCM8\_1X8 0x300c

SGRBG10\_DPCM8\_1X8 0x3009

SRGGB10\_DPCM8\_1X8 0x300d

SBGGR10\_2X8\_PADHI\_BE 0x3003

SBGGR10\_2X8\_PADHI\_LE 0x3004

SBGGR10\_2X8\_PADLO\_BE 0x3005

SBGGR10\_2X8\_PADLO\_LE 0x3006

SBGGR10\_1X10 0x3007

SGBRG10\_1X10 0x300e

SGRBG10\_1X10 0x300a

SRGGB10\_1X10 0x300f

SBGGR12\_1X12 0x3008

SGBRG12\_1X12 0x3010

SGRBG12\_1X12 0x3011

SRGGB12\_1X12 0x3012

SBGGR14\_1X14 0x3019

SGBRG14\_1X14 0x301a

SGRBG14\_1X14 0x301b

SRGGB14\_1X14 0x301c

SBGGR16\_1X16 0x301d

SGBRG16\_1X16 0x301e

SGRBG16\_1X16 0x301f

SRGGB16\_1X16 0x3020

JPEG\_1X8 0x4001

S5C\_UYVY\_JPEG\_1X8 0x5001

AHSV8888\_1X32 0x6001

root@ATK-DLRK356X:/#

v4l2-ctl --list-formats-ext --device /dev/video0 查看MP支持的像素格式：

root@ATK-DLRK356X:/# v4l2-ctl --list-formats-ext --device /dev/video0

ioctl: VIDIOC\_ENUM\_FMT

Type: Video Capture Multiplanar

[0]: 'UYVY' (UYVY 4:2:2)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[1]: '422P' (Planar YUV 4:2:2)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[2]: 'NV16' (Y/CbCr 4:2:2)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[3]: 'NV61' (Y/CrCb 4:2:2)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[4]: 'YM16' (Planar YUV 4:2:2 (N-C))

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[5]: 'NV21' (Y/CrCb 4:2:0)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[6]: 'NV12' (Y/CbCr 4:2:0)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[7]: 'NM21' (Y/CrCb 4:2:0 (N-C))

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[8]: 'NM12' (Y/CbCr 4:2:0 (N-C))

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[9]: 'YU12' (Planar YUV 4:2:0)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[10]: 'YM24' (Planar YUV 4:4:4 (N-C))

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[11]: 'RGGB' (8-bit Bayer RGRG/GBGB)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[12]: 'GRBG' (8-bit Bayer GRGR/BGBG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[13]: 'GBRG' (8-bit Bayer GBGB/RGRG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[14]: 'BA81' (8-bit Bayer BGBG/GRGR)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[15]: 'RG10' (10-bit Bayer RGRG/GBGB)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[16]: 'BA10' (10-bit Bayer GRGR/BGBG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[17]: 'GB10' (10-bit Bayer GBGB/RGRG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[18]: 'BG10' (10-bit Bayer BGBG/GRGR)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[19]: 'RG12' (12-bit Bayer RGRG/GBGB)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[20]: 'BA12' (12-bit Bayer GRGR/BGBG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[21]: 'GB12' (12-bit Bayer GBGB/RGRG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[22]: 'BG12' (12-bit Bayer BGBG/GRGR)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

root@ATK-DLRK356X:/#

在Ubuntu系统下执行 ffmpeg -pix\_fmts 命令 显示ffmpeg支持的显示格式：



alientek@alientek-virtual-machine:~$ ffmpeg -pix\_fmts

ffmpeg version 6.1 Copyright (c) 2000-2023 the FFmpeg developers

built with gcc 9 (Ubuntu 9.4.0-1ubuntu1~20.04.2)

configuration: --prefix=/usr/local/ffmpeg --enable-shared

libavutil 58. 29.100 / 58. 29.100

libavcodec 60. 31.102 / 60. 31.102

libavformat 60. 16.100 / 60. 16.100

libavdevice 60. 3.100 / 60. 3.100

libavfilter 9. 12.100 / 9. 12.100

libswscale 7. 5.100 / 7. 5.100

libswresample 4. 12.100 / 4. 12.100

Pixel formats:

I.... = Supported Input format for conversion

.O... = Supported Output format for conversion

..H.. = Hardware accelerated format

...P. = Paletted format

....B = Bitstream format

FLAGS NAME NB\_COMPONENTS BITS\_PER\_PIXEL BIT\_DEPTHS

-----

IO... yuv420p 3 12 8-8-8

IO... yuyv422 3 16 8-8-8

IO... rgb24 3 24 8-8-8

IO... bgr24 3 24 8-8-8

IO... yuv422p 3 16 8-8-8

IO... yuv444p 3 24 8-8-8

IO... yuv410p 3 9 8-8-8

IO... yuv411p 3 12 8-8-8

IO... gray 1 8 8

IO..B monow 1 1 1

IO..B monob 1 1 1

I..P. pal8 1 8 8

IO... yuvj420p 3 12 8-8-8

IO... yuvj422p 3 16 8-8-8

IO... yuvj444p 3 24 8-8-8

IO... uyvy422 3 16 8-8-8

..... uyyvyy411 3 12 8-8-8

IO... bgr8 3 8 3-3-2

.O..B bgr4 3 4 1-2-1

IO... bgr4\_byte 3 4 1-2-1

IO... rgb8 3 8 2-3-3

.O..B rgb4 3 4 1-2-1

IO... rgb4\_byte 3 4 1-2-1

IO... nv12 3 12 8-8-8

IO... nv21 3 12 8-8-8

IO... argb 4 32 8-8-8-8

IO... rgba 4 32 8-8-8-8

IO... abgr 4 32 8-8-8-8

IO... bgra 4 32 8-8-8-8

IO... gray16be 1 16 16

IO... gray16le 1 16 16

IO... yuv440p 3 16 8-8-8

IO... yuvj440p 3 16 8-8-8

IO... yuva420p 4 20 8-8-8-8

IO... rgb48be 3 48 16-16-16

IO... rgb48le 3 48 16-16-16

IO... rgb565be 3 16 5-6-5

IO... rgb565le 3 16 5-6-5

IO... rgb555be 3 15 5-5-5

IO... rgb555le 3 15 5-5-5

IO... bgr565be 3 16 5-6-5

IO... bgr565le 3 16 5-6-5

IO... bgr555be 3 15 5-5-5

IO... bgr555le 3 15 5-5-5

..H.. vaapi 0 0 0

IO... yuv420p16le 3 24 16-16-16

IO... yuv420p16be 3 24 16-16-16

IO... yuv422p16le 3 32 16-16-16

IO... yuv422p16be 3 32 16-16-16

IO... yuv444p16le 3 48 16-16-16

IO... yuv444p16be 3 48 16-16-16

..H.. dxva2\_vld 0 0 0

IO... rgb444le 3 12 4-4-4

IO... rgb444be 3 12 4-4-4

IO... bgr444le 3 12 4-4-4

IO... bgr444be 3 12 4-4-4

IO... ya8 2 16 8-8

IO... bgr48be 3 48 16-16-16

IO... bgr48le 3 48 16-16-16

IO... yuv420p9be 3 13 9-9-9

IO... yuv420p9le 3 13 9-9-9

IO... yuv420p10be 3 15 10-10-10

IO... yuv420p10le 3 15 10-10-10

IO... yuv422p10be 3 20 10-10-10

IO... yuv422p10le 3 20 10-10-10

IO... yuv444p9be 3 27 9-9-9

IO... yuv444p9le 3 27 9-9-9

IO... yuv444p10be 3 30 10-10-10

IO... yuv444p10le 3 30 10-10-10

IO... yuv422p9be 3 18 9-9-9

IO... yuv422p9le 3 18 9-9-9

IO... gbrp 3 24 8-8-8

IO... gbrp9be 3 27 9-9-9

IO... gbrp9le 3 27 9-9-9

IO... gbrp10be 3 30 10-10-10

IO... gbrp10le 3 30 10-10-10

IO... gbrp16be 3 48 16-16-16

IO... gbrp16le 3 48 16-16-16

IO... yuva422p 4 24 8-8-8-8

IO... yuva444p 4 32 8-8-8-8

IO... yuva420p9be 4 22 9-9-9-9

IO... yuva420p9le 4 22 9-9-9-9

IO... yuva422p9be 4 27 9-9-9-9

IO... yuva422p9le 4 27 9-9-9-9

IO... yuva444p9be 4 36 9-9-9-9

IO... yuva444p9le 4 36 9-9-9-9

IO... yuva420p10be 4 25 10-10-10-10

IO... yuva420p10le 4 25 10-10-10-10

IO... yuva422p10be 4 30 10-10-10-10

IO... yuva422p10le 4 30 10-10-10-10

IO... yuva444p10be 4 40 10-10-10-10

IO... yuva444p10le 4 40 10-10-10-10

IO... yuva420p16be 4 40 16-16-16-16

IO... yuva420p16le 4 40 16-16-16-16

IO... yuva422p16be 4 48 16-16-16-16

IO... yuva422p16le 4 48 16-16-16-16

IO... yuva444p16be 4 64 16-16-16-16

IO... yuva444p16le 4 64 16-16-16-16

..H.. vdpau 0 0 0

IO... xyz12le 3 36 12-12-12

IO... xyz12be 3 36 12-12-12

IO... nv16 3 16 8-8-8

..... nv20le 3 20 10-10-10

..... nv20be 3 20 10-10-10

IO... rgba64be 4 64 16-16-16-16

IO... rgba64le 4 64 16-16-16-16

IO... bgra64be 4 64 16-16-16-16

IO... bgra64le 4 64 16-16-16-16

IO... yvyu422 3 16 8-8-8

IO... ya16be 2 32 16-16

IO... ya16le 2 32 16-16

IO... gbrap 4 32 8-8-8-8

IO... gbrap16be 4 64 16-16-16-16

IO... gbrap16le 4 64 16-16-16-16

..H.. qsv 0 0 0

..H.. mmal 0 0 0

..H.. d3d11va\_vld 0 0 0

..H.. cuda 0 0 0

IO... 0rgb 3 24 8-8-8

IO... rgb0 3 24 8-8-8

IO... 0bgr 3 24 8-8-8

IO... bgr0 3 24 8-8-8

IO... yuv420p12be 3 18 12-12-12

IO... yuv420p12le 3 18 12-12-12

IO... yuv420p14be 3 21 14-14-14

IO... yuv420p14le 3 21 14-14-14

IO... yuv422p12be 3 24 12-12-12

IO... yuv422p12le 3 24 12-12-12

IO... yuv422p14be 3 28 14-14-14

IO... yuv422p14le 3 28 14-14-14

IO... yuv444p12be 3 36 12-12-12

IO... yuv444p12le 3 36 12-12-12

IO... yuv444p14be 3 42 14-14-14

IO... yuv444p14le 3 42 14-14-14

IO... gbrp12be 3 36 12-12-12

IO... gbrp12le 3 36 12-12-12

IO... gbrp14be 3 42 14-14-14

IO... gbrp14le 3 42 14-14-14

IO... yuvj411p 3 12 8-8-8

I.... bayer\_bggr8 3 8 2-4-2

I.... bayer\_rggb8 3 8 2-4-2

I.... bayer\_gbrg8 3 8 2-4-2

I.... bayer\_grbg8 3 8 2-4-2

I.... bayer\_bggr16le 3 16 4-8-4

I.... bayer\_bggr16be 3 16 4-8-4

I.... bayer\_rggb16le 3 16 4-8-4

I.... bayer\_rggb16be 3 16 4-8-4

I.... bayer\_gbrg16le 3 16 4-8-4

I.... bayer\_gbrg16be 3 16 4-8-4

I.... bayer\_grbg16le 3 16 4-8-4

I.... bayer\_grbg16be 3 16 4-8-4

..H.. xvmc 0 0 0

IO... yuv440p10le 3 20 10-10-10

IO... yuv440p10be 3 20 10-10-10

IO... yuv440p12le 3 24 12-12-12

IO... yuv440p12be 3 24 12-12-12

IO... ayuv64le 4 64 16-16-16-16

..... ayuv64be 4 64 16-16-16-16

..H.. videotoolbox\_vld 0 0 0

IO... p010le 3 15 10-10-10

IO... p010be 3 15 10-10-10

IO... gbrap12be 4 48 12-12-12-12

IO... gbrap12le 4 48 12-12-12-12

IO... gbrap10be 4 40 10-10-10-10

IO... gbrap10le 4 40 10-10-10-10

..H.. mediacodec 0 0 0

IO... gray12be 1 12 12

IO... gray12le 1 12 12

IO... gray10be 1 10 10

IO... gray10le 1 10 10

IO... p016le 3 24 16-16-16

IO... p016be 3 24 16-16-16

..H.. d3d11 0 0 0

IO... gray9be 1 9 9

IO... gray9le 1 9 9

IO... gbrpf32be 3 96 32-32-32

IO... gbrpf32le 3 96 32-32-32

IO... gbrapf32be 4 128 32-32-32-32

IO... gbrapf32le 4 128 32-32-32-32

..H.. drm\_prime 0 0 0

..H.. opencl 0 0 0

IO... gray14be 1 14 14

IO... gray14le 1 14 14

IO... grayf32be 1 32 32

IO... grayf32le 1 32 32

IO... yuva422p12be 4 36 12-12-12-12

IO... yuva422p12le 4 36 12-12-12-12

IO... yuva444p12be 4 48 12-12-12-12

IO... yuva444p12le 4 48 12-12-12-12

IO... nv24 3 24 8-8-8

IO... nv42 3 24 8-8-8

..H.. vulkan 0 0 0

..... y210be 3 20 10-10-10

IO... y210le 3 20 10-10-10

IO... x2rgb10le 3 30 10-10-10

..... x2rgb10be 3 30 10-10-10

IO... x2bgr10le 3 30 10-10-10

..... x2bgr10be 3 30 10-10-10

IO... p210be 3 20 10-10-10

IO... p210le 3 20 10-10-10

IO... p410be 3 30 10-10-10

IO... p410le 3 30 10-10-10

IO... p216be 3 32 16-16-16

IO... p216le 3 32 16-16-16

IO... p416be 3 48 16-16-16

IO... p416le 3 48 16-16-16

IO... vuya 4 32 8-8-8-8

I.... rgbaf16be 4 64 16-16-16-16

I.... rgbaf16le 4 64 16-16-16-16

IO... vuyx 3 24 8-8-8

IO... p012le 3 18 12-12-12

IO... p012be 3 18 12-12-12

..... y212be 3 24 12-12-12

IO... y212le 3 24 12-12-12

....B xv30be 3 30 10-10-10

IO... xv30le 3 30 10-10-10

..... xv36be 3 36 12-12-12

IO... xv36le 3 36 12-12-12

..... rgbf32be 3 96 32-32-32

..... rgbf32le 3 96 32-32-32

..... rgbaf32be 4 128 32-32-32-32

..... rgbaf32le 4 128 32-32-32-32

IO... p212be 3 24 12-12-12

IO... p212le 3 24 12-12-12

IO... p412be 3 36 12-12-12

IO... p412le 3 36 12-12-12

IO... gbrap14be 4 56 14-14-14-14

IO... gbrap14le 4 56 14-14-14-14

alientek@alientek-virtual-machine:~$

总结得：

（media-ctl --known-mbus-fmts )当前系统中支持的byer媒体总线格式：

SBGGR8\_1X8 0x3001

SGBRG8\_1X8 0x3013

SGRBG8\_1X8 0x3002

SRGGB8\_1X8 0x3014

SBGGR10\_ALAW8\_1X8 0x3015

SGBRG10\_ALAW8\_1X8 0x3016

SGRBG10\_ALAW8\_1X8 0x3017

SRGGB10\_ALAW8\_1X8 0x3018

SBGGR10\_DPCM8\_1X8 0x300b

SGBRG10\_DPCM8\_1X8 0x300c

SGRBG10\_DPCM8\_1X8 0x3009

SRGGB10\_DPCM8\_1X8 0x300d

SBGGR10\_2X8\_PADHI\_BE 0x3003

SBGGR10\_2X8\_PADHI\_LE 0x3004

SBGGR10\_2X8\_PADLO\_BE 0x3005

SBGGR10\_2X8\_PADLO\_LE 0x3006

SBGGR10\_1X10 0x3007

SGBRG10\_1X10 0x300e

SGRBG10\_1X10 0x300a

SRGGB10\_1X10 0x300f

SBGGR12\_1X12 0x3008

SGBRG12\_1X12 0x3010

SGRBG12\_1X12 0x3011

SRGGB12\_1X12 0x3012

SBGGR14\_1X14 0x3019

SGBRG14\_1X14 0x301a

SGRBG14\_1X14 0x301b

SRGGB14\_1X14 0x301c

SBGGR16\_1X16 0x301d

SGBRG16\_1X16 0x301e

SGRBG16\_1X16 0x301f

SRGGB16\_1X16 0x3020

MP支持的byer像素格式：

（v4l2-ctl --list-formats-ext --device /dev/video0）

[11]: 'RGGB' (8-bit Bayer RGRG/GBGB)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[12]: 'GRBG' (8-bit Bayer GRGR/BGBG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[13]: 'GBRG' (8-bit Bayer GBGB/RGRG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

Ubuntu系统下ffmpeg支持的显示byer格式：

I.... bayer\_bggr8 3 8 2-4-2

I.... bayer\_rggb8 3 8 2-4-2

I.... bayer\_gbrg8 3 8 2-4-2

I.... bayer\_grbg8 3 8 2-4-2

因此，为了能够直接在ffplay上显示byer格式的图像，我们可以

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

对于一帧 3840x2160 的 NV12 图像，其大小计算过程如下：

计算总像素数：

图像的分辨率为 3840x2160，即表示该图像横向具有 3840 个像素，纵向具有 2160 个像素，那么总像素数为：

3840×2160=8294400

（像素）

计算占用字节数：

在 NV12 格式中，每个像素的亮度分量 Y 占用 1 个字节，色度分量 U 和 V 是按照 2×2 的块进行采样，每 4 个 Y 分量共用一组 U、V 分量，U 和 V 分量总共占用半个字节。所以每个像素在 NV12 格式下占用的存储空间为

1+0.5=1.5字节。

则该图像占用的总字节数为：

8294400×1.5=12441600（字节）

单位换算为 M

因为

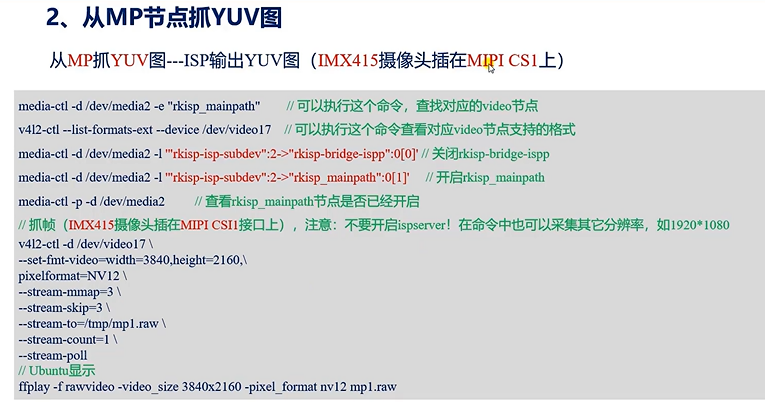
1M=1024×1024=1048576

字节，所以将总字节数换算为 M 可得：

12441600÷1048576≈11.86M≈12M

\*/

1. 从MP节点抓取YUV图：



抓取命令：

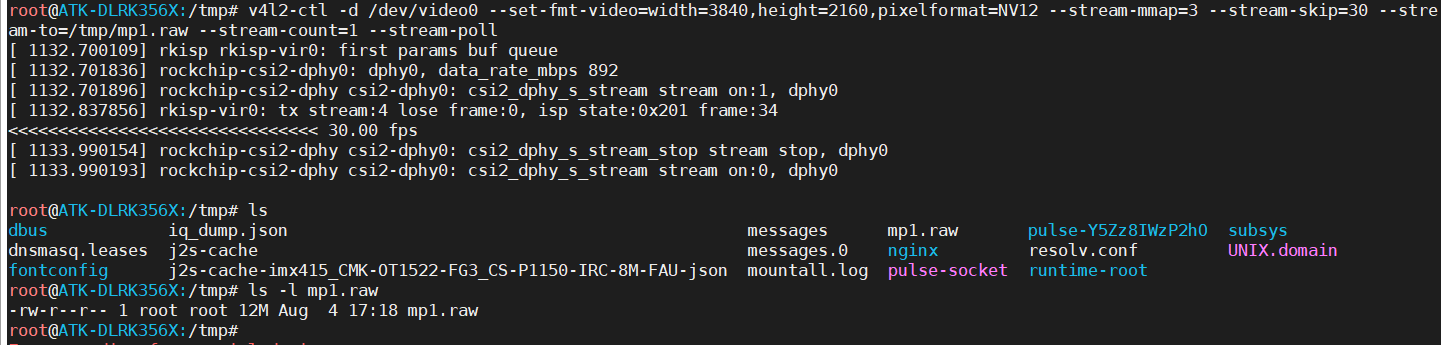
v4l2-ctl -d /dev/video0 --set-fmt-video=width=3840,height=2160,pixelformat=NV12 --stream-mmap=3 --stream-skip=30 --stream-to=/tmp/mp1.raw --stream-count=1 --stream-poll

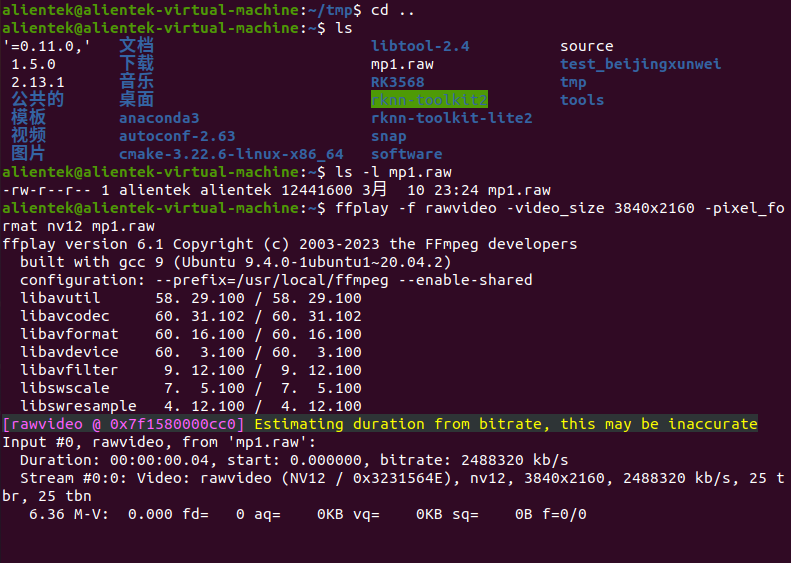
由于MP输出图像可缩放裁剪，所以也可以输出： 1920x1080的图像

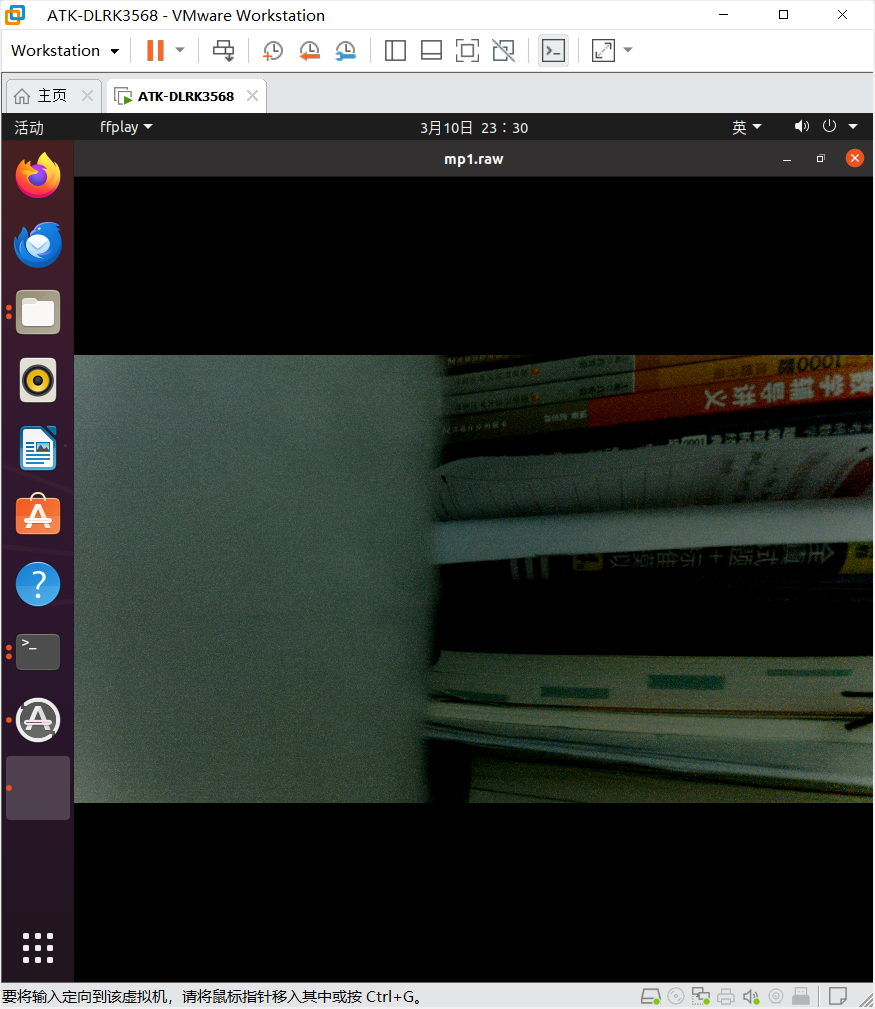
显示命令：

ffplay -f rawvideo -video\_size 3840x2160 -pixel\_format nv12 mp1.raw

/\*

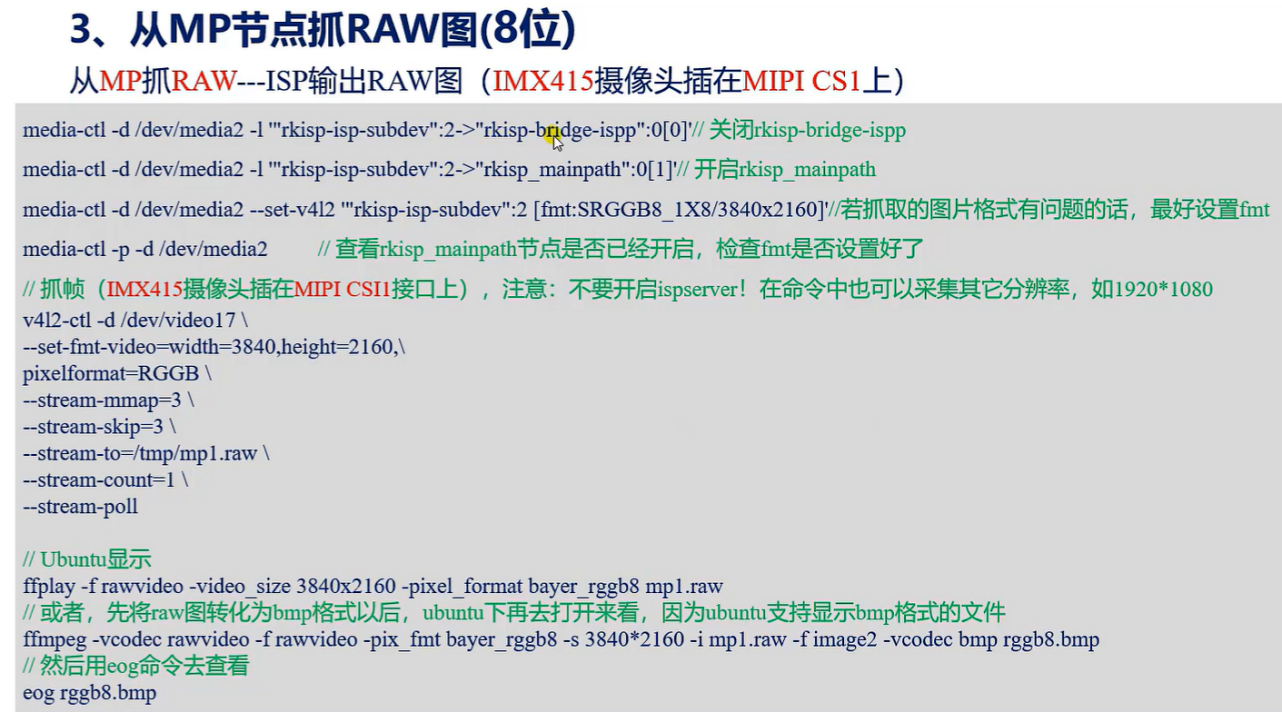






\*/

1. 从MP节点抓取RAW图（8位）



前面说了，这条媒体总线默认的像素格式是YUV格式的，我们要抓取raw图就必须要设置媒体总线的像素格式以后，再去抓，才能确保我们抓取的raw图的格式是正确的。

我们上面查看了开发板下的媒体总线支持的像素格式：

SBGGR8\_1X8 0x3001

SGBRG8\_1X8 0x3013

SGRBG8\_1X8 0x3002

SRGGB8\_1X8 0x3014

以及 MP支持的像素格式：

1. : 'RGGB' (8-bit Bayer RGRG/GBGB)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

1. : 'GRBG' (8-bit Bayer GRGR/BGBG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

[13]: 'GBRG' (8-bit Bayer GBGB/RGRG)

Size: Stepwise 32x16 - 3840x2160 with step 8/8

那么如果我们想要RGGB、GRBG、GBRG格式的图像，就需要依次去设置 媒体总线的像素格式：

RGGB —— SRGGB8\_1X8

GRBG —— SGRBG8\_1X8

GBRG —— SGBRG8\_1X8

media-ctl -p -d /dev/media0

执行命令：设置媒体总线的像素格式。

media-ctl -d /dev/media0 --set-v4l2 '"rkisp-isp-subdev":2[fmt:SRGGB8\_1X8/3840x2160]'

抓取命令：

v4l2-ctl -d /dev/video0 --set-fmt-video=width=3840,height=2160,pixelformat=RGGB --stream-mmap=3 --stream-skip=30 --stream-to=/tmp/mp1.raw --stream-count=1 --stream-poll

显示命令：

ffplay -f rawvideo -video\_size 3840x2160 -pixel\_format bayer\_rggb8 mp1.raw

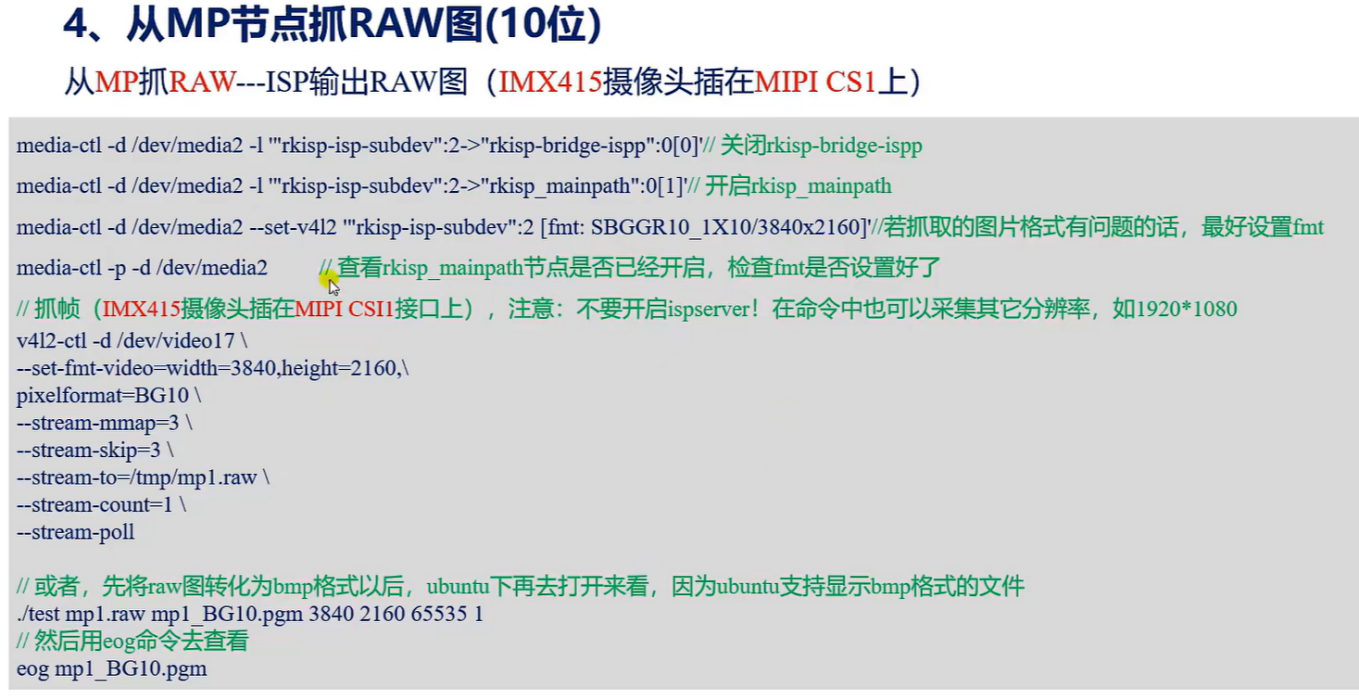
或者先将raw图转化为bmp格式之后，再在Ubuntu打开：

ffmpeg -vcodec rawvideo -f rawvideo -pix\_fmt bayer\_rggb8 -s 3840\*2160 -i mp1.raw -f image2 -vcodec bmp rggb8.bmp

然后再用eog命令查看：

eog rggb8.bmp

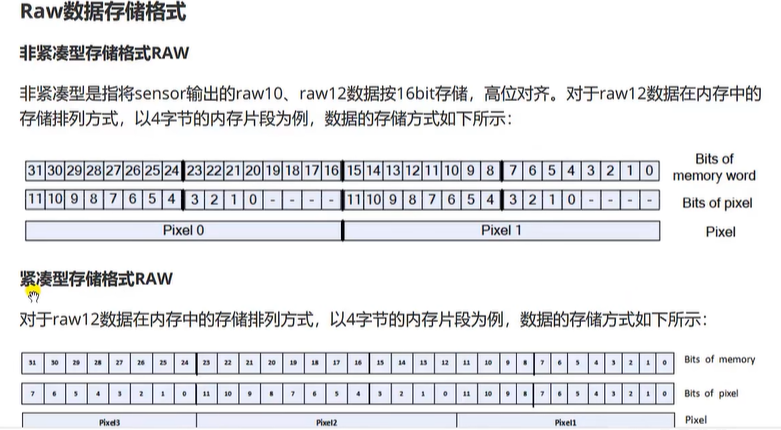
1. 从MP节点抓取RAW图（10位）



/\*

紧凑型和非紧凑型：见参考文档Rockchip\_Driver\_Guide\_VI\_CN：

"C:\Users\zhongqing\Desktop\笔记\08、RV1126参考资料\RV1126\_RV1109\Camera\Rockchip\_Driver\_Guide\_VI\_CN\_v1.0.8.pdf"





所以采集后一帧数据的文件大小计算方式为：

总像素：3840x2160

采用16位存储，所以共这么多位：3840x2160x16

转化为字节：3840x2160x16/8

转化为M字节：3840x2160x16/8/1024/1024 = 15.820... = 16M

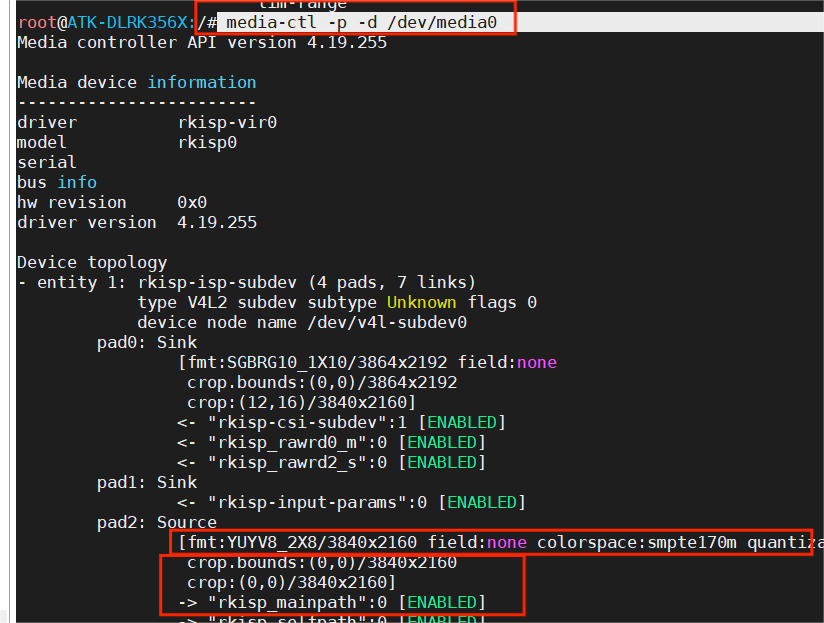
\*/

1. 从MP节点抓取RAW图（12位）



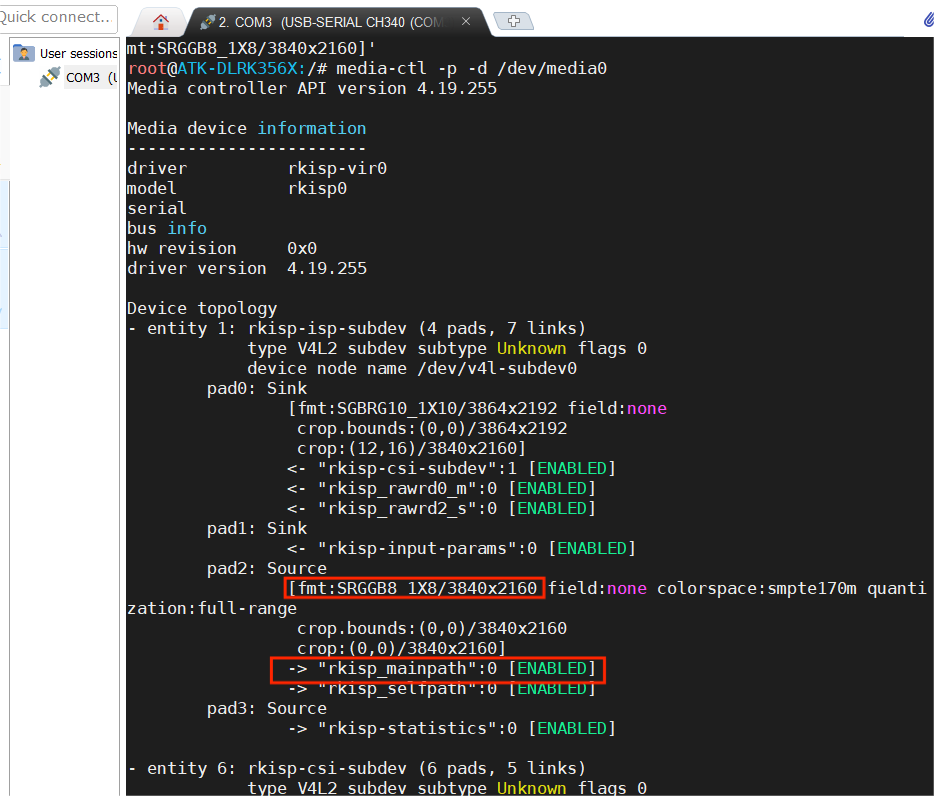
/\*

media-ctl -p -d /dev/media0



media-ctl -d /dev/media0 --set-v4l2 '"rkisp-isp-subdev":2[fmt:SRGGB8\_1X8/3840x2160]'

media-ctl -p -d /dev/media0



\*/