/\*

本节介绍：

画出数据链路的拓扑图。

也就是，在执行media-ctl -p -d /dev/media0查看节点拓扑关系时：这些打印的信息怎么分析，以及画出链路拓扑图，使用拓扑图来分析节点之间的拓扑关系：

root@ATK-DLRK356X:/lib/modules/4.19.232# media-ctl -p -d /dev/media0

Media controller API version 4.19.255

Media device information

------------------------

driver rkisp-vir0

model rkisp0

serial

bus info

hw revision 0x0

driver version 4.19.255

Device topology

- entity 1: rkisp-isp-subdev (4 pads, 7 links)

type V4L2 subdev subtype Unknown flags 0

device node name /dev/v4l-subdev0

pad0: Sink

[fmt:SGBRG10\_1X10/3864x2192 field:none

crop.bounds:(0,0)/3864x2192

crop:(12,16)/3840x2160]

<- "rkisp-csi-subdev":1 [ENABLED]

<- "rkisp\_rawrd0\_m":0 [ENABLED]

<- "rkisp\_rawrd2\_s":0 [ENABLED]

pad1: Sink

<- "rkisp-input-params":0 [ENABLED]

pad2: Source

[fmt:YUYV8\_2X8/3840x2160 field:none colorspace:smpte170m quantizatiol-range

crop.bounds:(0,0)/3840x2160

crop:(0,0)/3840x2160]

-> "rkisp\_mainpath":0 [ENABLED]

-> "rkisp\_selfpath":0 [ENABLED]

pad3: Source

-> "rkisp-statistics":0 [ENABLED]

- entity 6: rkisp-csi-subdev (6 pads, 5 links)

type V4L2 subdev subtype Unknown flags 0

device node name /dev/v4l-subdev1

pad0: Sink

[fmt:SGBRG10\_1X10/3864x2192 field:none]

<- "rockchip-csi2-dphy0":1 [ENABLED]

pad1: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp-isp-subdev":0 [ENABLED]

pad2: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp\_rawwr0":0 [ENABLED]

pad3: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

pad4: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp\_rawwr2":0 [ENABLED]

pad5: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp\_rawwr3":0 [ENABLED]

- entity 13: rkisp\_mainpath (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video0

pad0: Sink

<- "rkisp-isp-subdev":2 [ENABLED]

- entity 19: rkisp\_selfpath (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video1

pad0: Sink

<- "rkisp-isp-subdev":2 [ENABLED]

- entity 25: rkisp\_rawwr0 (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video2

pad0: Sink

<- "rkisp-csi-subdev":2 [ENABLED]

- entity 31: rkisp\_rawwr2 (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video3

pad0: Sink

<- "rkisp-csi-subdev":4 [ENABLED]

- entity 37: rkisp\_rawwr3 (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video4

pad0: Sink

<- "rkisp-csi-subdev":5 [ENABLED]

- entity 43: rkisp\_rawrd0\_m (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video5

pad0: Source

-> "rkisp-isp-subdev":0 [ENABLED]

- entity 49: rkisp\_rawrd2\_s (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video6

pad0: Source

-> "rkisp-isp-subdev":0 [ENABLED]

- entity 55: rkisp-statistics (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video7

pad0: Sink

<- "rkisp-isp-subdev":3 [ENABLED]

- entity 61: rkisp-input-params (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video8

pad0: Source

-> "rkisp-isp-subdev":1 [ENABLED]

- entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)

type V4L2 subdev subtype Unknown flags 0

device node name /dev/v4l-subdev2

pad0: Sink

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

crop.bounds:(12,16)/3840x2160]

<- "m00\_b\_imx415 4-001a-1":0 [ENABLED]

pad1: Source

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

crop.bounds:(12,16)/3840x2160]

-> "rkisp-csi-subdev":0 [ENABLED]

- entity 70: m00\_b\_imx415 4-001a-1 (1 pad, 1 link)

type V4L2 subdev subtype Sensor flags 0

device node name /dev/v4l-subdev3

pad0: Source

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

crop.bounds:(12,16)/3840x2160]

-> "rockchip-csi2-dphy0":0 [ENABLED]

root@ATK-DLRK356X:/lib/modules/4.19.232#

\*/

/\*



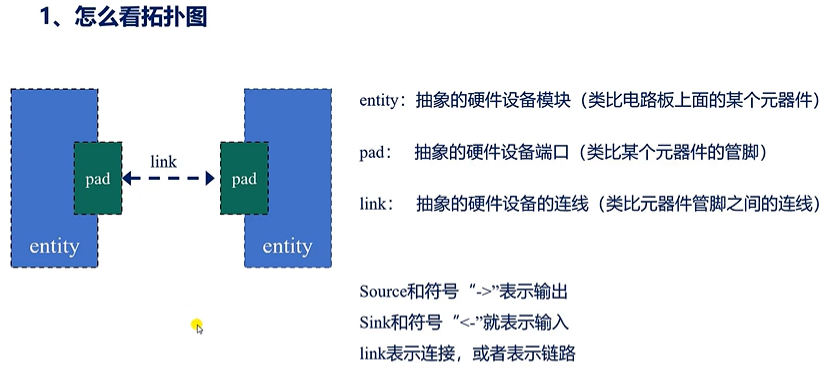
1. 怎么看拓扑图：

怎么看拓扑图？也就是怎么看这些节点弹出的信息？

执行media-ctl -p -d /dev/media0 命令之后弹出的信息怎么看呢？

首先介绍节点之间的拓扑关系是怎么联系起来的：

在弹出的信息里面：entity、pad、Sink、Source等是什么关系？



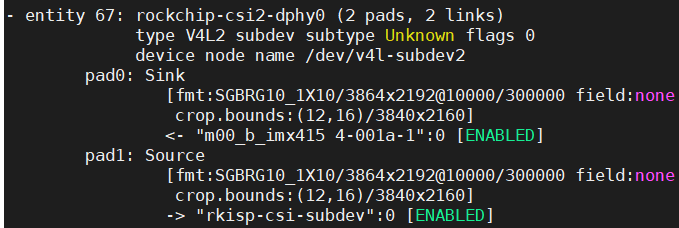
由图：

entity：是抽象的硬件设备模块（类比电路板某个元器件/芯片）

例如：entity 67: rockchip-csi2-dphy0 67是索引号

pad ：是抽象的硬件设备端口（类比于某个元器件的管脚/芯片某个管脚）

例如 entity 67就有很多的pad，也就是说他有很多的端口：



link ：抽象的硬件设备的连线。（类比于元器件管教之间的连线），用于连接元器件的，例如上面的两个元器件通过连线把两个管脚连接起来。link就是连线。 例如 entity就有 2个pad 2个link。

Source 和 -> 符号：表示输出的含义。表示从端口输出数据到某一个端口。例如上面的 entity：rockchip-csi2-dphy0 的端口1（pad1）就输出到了rkisp-csi-subdev 这个entity的端口0（rkisp-csi-subdev:0）了。

Sink 和 <- 符号就表示输入：就表示从某一个entity端口输入到这个entity的端口，例如：entity m00\_b\_imx415 4-001a-1 的端口0 输出到了 entity rockchip-csi2-dphy0的端口0。

1. 怎么画数据链路的拓扑图：

/\*

设备树分析的拓扑结构：

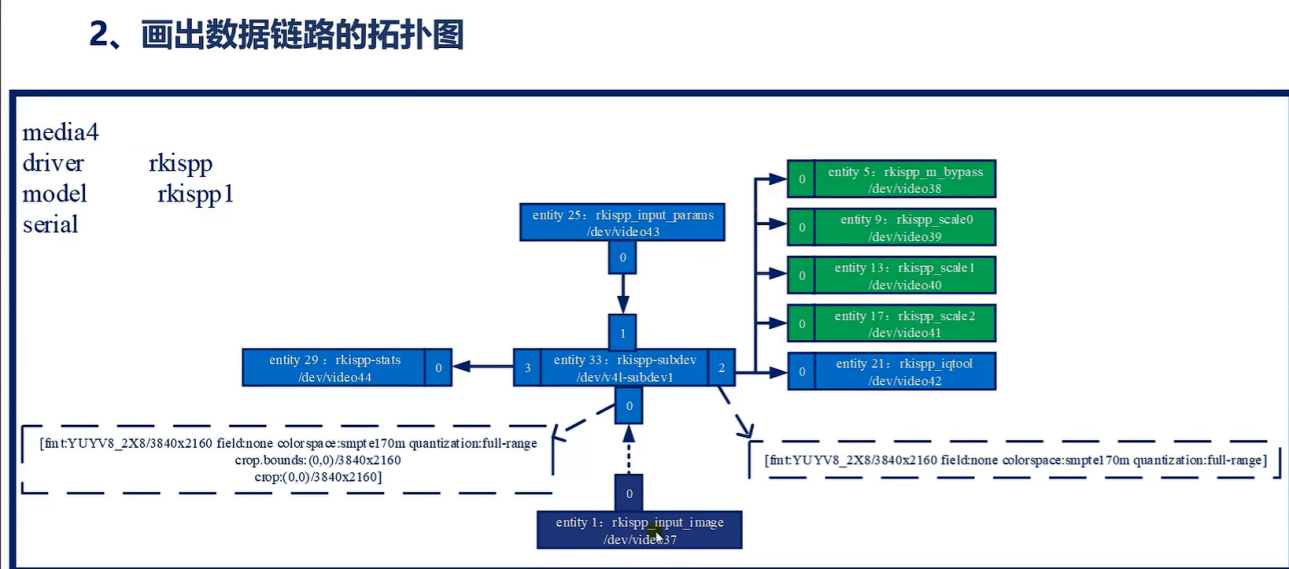
Imx415/Imx335(I2C4) -- CSI2 dphy0 -- rkisp\_vir0

\*/

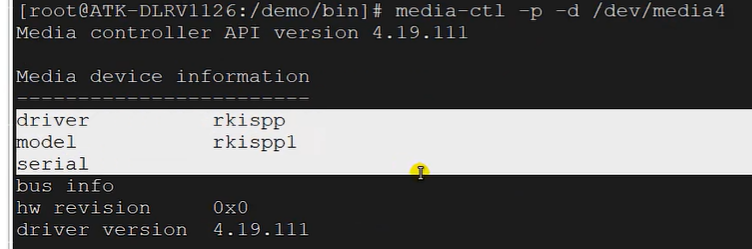
知道了上面的基础知识，我们就来分析我们的开发板的拓扑结构：（已知我们使用的是I2C4 的MIPI CSI接口）

/\*

他meida4的拓扑图：



左上角的就是这些信息：driver 就是rkispp model 就是 rkispp1



下面来讲解这个图是怎么画出来的:

\*/

———————————————————————————————————————

首先我们先来找到 entity 1: 也就是rkisp-isp-subdev。

Entity 1：rkisp-isp-subdev：

这个entity的索引是1，名称是rkisp-isp-subdev。并且这个entity是一个v4l2子设备，这个entity 有4个端口，有7个连线。

这个端口是0、1、2、3。

然后它的video节点 是/dev/v4l-subdev0。

再来看端口的连线：

Pad0:

从 rkisp-csi-subdev的端口1、

rkisp-rawrd0\_m 的端口0、

Rkisp-rawrd2\_s 的端口0接收

Pad1:

从 rkisp-input-params 的端口0接收

Pad2:

输出到：

rkisp\_mainpath 的端口0 和

rkisp\_selfpath 的端口0

Pad3:

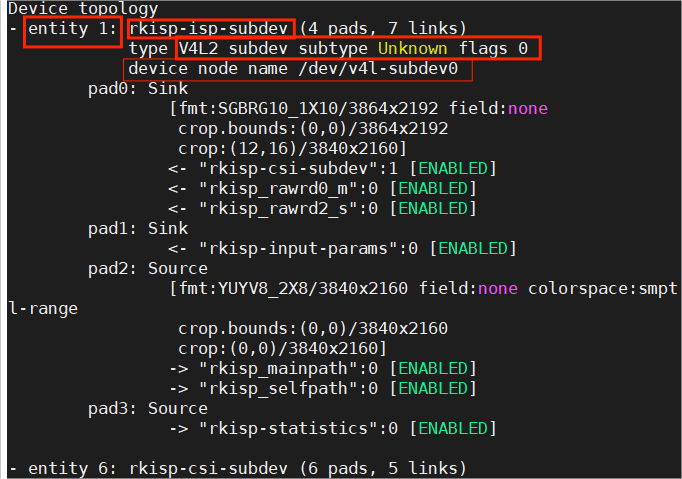
输出到：

rkisp-statistics的端口0

注意：

如果方括号内是空的，表示当前链路没有被占用（使用）。也就是说这个链路没有使能。（所以他下面的线是虚线）

如果使能了的话就是ENABLED，就表示这个链路被占用了，被使用了，所以画的是实线。



———————————————————————————————————————

然后：

entity 6: rkisp-csi-subdev：

它也是一个v4l2子设备，有6个pad和5个link

它的video节点名称为：/dev/v4l-subdev1

Pad0:

从rockchip-csi2-dphy0 的端口1 输入

Pad1:

输出到rkisp-isp-subdev 的端口0

Pad2:

输出到 rkisp\_rawwr0的端口0

Pad3

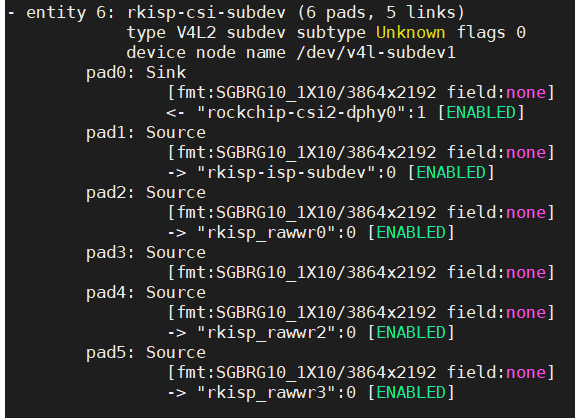
无输出

Pad4:

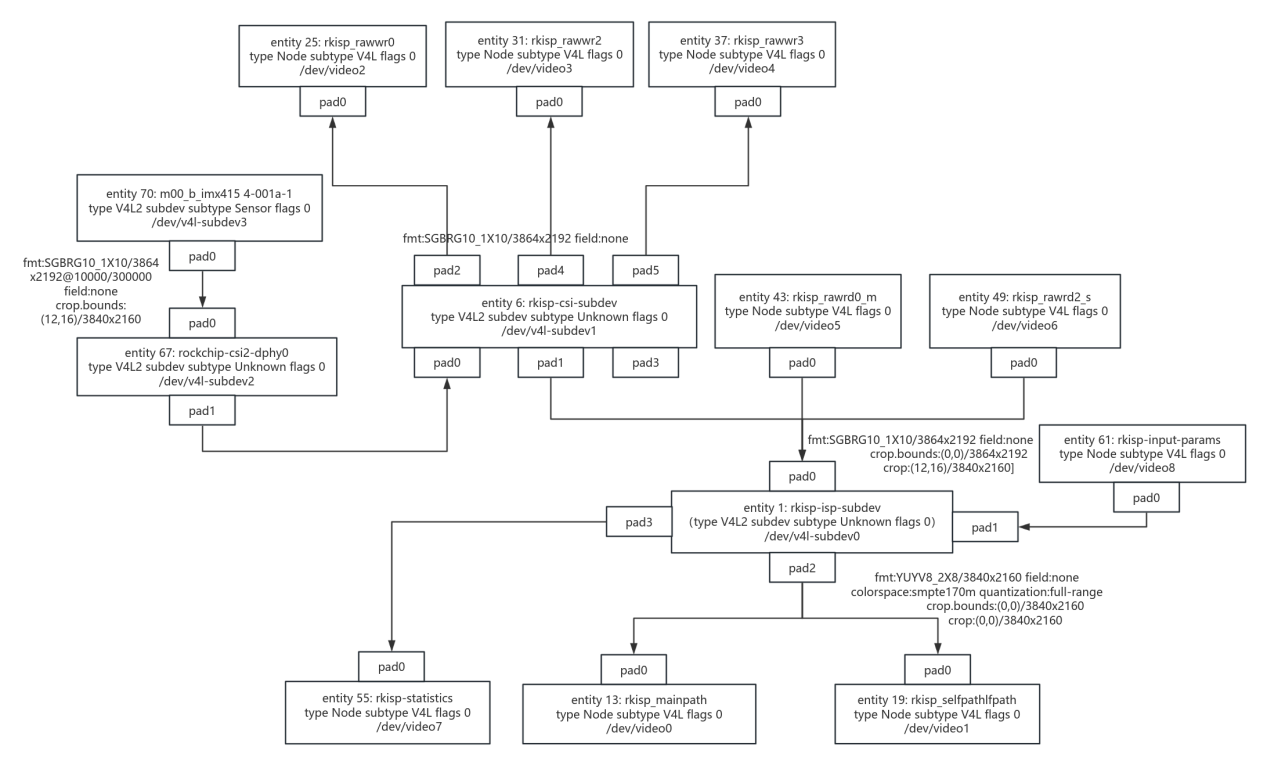
输出到 rkisp\_rawwr2 的端口0

Pad5:

输出到 rkisp\_rawwr3 的端口0



同理，绘制后，最终得：



entity 70: m00\_b\_imx415 4-001a-1 (1 pad, 1 link)

type V4L2 subdev subtype Sensor flags 0

device node name /dev/v4l-subdev3

pad0: Source

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

crop.bounds:(12,16)/3840x2160]

-> "rockchip-csi2-dphy0":0 [ENABLED]

我们由这一个entity以及上图得：entity 70就是IMX415 Sensor的entity。如果开发板上插了摄像头，但没有看到这个Sensor的entity，说明Sensor的注册有问题 或者 摄像头没插好等。

我们重点来分析这个Sensor的entity：

这个entity的索引是70，名称是m00\_b\_imx415 4-001a-1。

其中m00表示索引值，m表示模块model，00就是它的索引值。这个值就表示这个Sensor插在开发板的哪个MIPI CSI接口上。

（如果插在了MIPI CSI0上就显示m00，如果插在MIPI CSI1上，就显示m01。）

中间的b就表示位置：

如果是b，就表示back，也就是后面的意思，就表示在其它entity的后面，

也就表示位于rockchip-csi2-dphy0的后面

如果是f，就表示front 也就是前面的意思，就表示在其它entity的前面。

Imx415就是模组Sensor的名称。

4-001a:4 表示当前的Sensor占用了哪一个I2C，这里显示4，就表示当前的Sensor占用了I2C4，后面的001a，表示Sensor的从机地址。在前面讲过IMX415/IMX335的从机地址就是16进制的0x1a。

1 pad 1 link 表示一个端口，和一个连线。它的端口就是端口0，它的连线就是:

-> "rockchip-csi2-dphy0":0 [ENABLED]

也就表示从entity的端口0输出到rockchip-csi2-dphy0的端口0。

type V4L2 subdev subtype Sensor flags 0：表示它是v4l2子设备，而且是一个Sensor。

它的设备名称是： /dev/v4l-subdev3

最后就是这一段信息:

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

crop.bounds:(12,16)/3840x2160]

这段信息他就表示图像的像素格式。像素格式为：GBRG，这个是一种媒体总线像素格式的缩写。表示在媒体总线上传输图像的格式，也就是像素的格式。媒体总线上的像素格式，和内存中的像素格式是不一样的，在内存中像素的格式我们经常用4个字符来表示（例如：NV12、YUYV，像这种四个字符的像素格式，一般就是内存中的像素格式），内存中的像素格式和我们这里媒体总线上的像素格式是不一样的。

GBRG表示的就是Byer格式，也就是raw数据。后面的10就表示这个像素分量的位数是10位。比如RGB888就表示红绿蓝色分量各有8位。

1x10:后面的10就表示每个样本有多少位。也就表示，媒体总线的宽度是10位。前面的1就表示样本的个数。也可以理解为媒体总线的个数是1。一般，比总线宽度 还大的 像素 必须在多个样本中传输，也就是说：比总线宽度 宽 的像素，必须在多个总线中传输。常见的总线样本数有：1，2，3。这里总线样本数就是1。

SGBRG10\_1X10这个媒体总线上传输的这个像素格式，我们也称为Media Bus Pixel Codes(Mbus-code)。我们可以在内核中找到定义： kernel/include/uapi/linux/media-bus-format.h

这个文件就定义了媒体总线的像素格式。

我们也可以在开发板上执行对应的命令查看当前开发板上媒体总线的像素格式有哪些：

media-ctl --known-mbus-fmts ：知道 媒体总线 格式。

这些内容就是在media-bus-format.h这个文件中定义的。（不用纠结，只要了解GBRG表示传输的是Byer格式的数据即可）

/\*

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

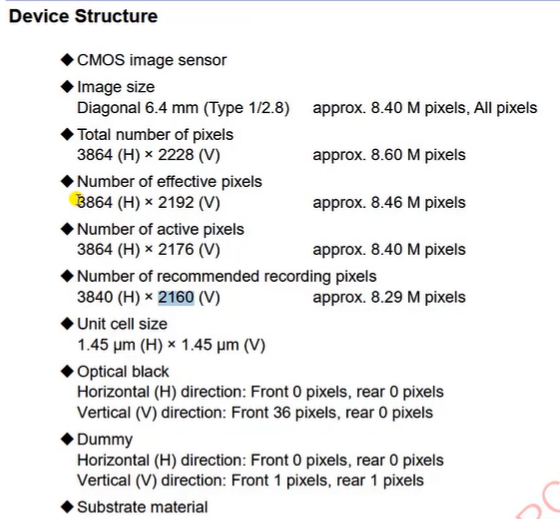
S5C\_UYVY\_JPEG\_1X8 0x5001

AHSV8888\_1X32 0x6001

root@ATK-DLRK356X:/#

\*/

后面的3864x2192：就表示像素的分辨率，在datesheet里面也可以查到。2864x2192就是有效像素值：

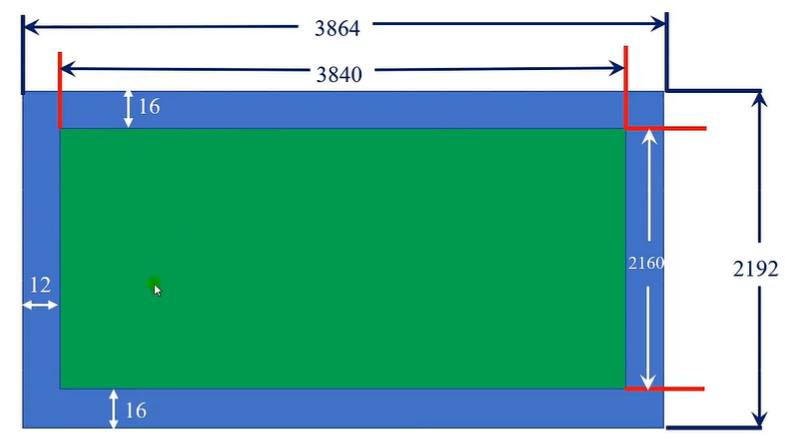


后面@10000/300000表示帧率。这个帧率是30帧每秒。也就是300000是分子，10000是分母。所以是30帧每秒。IMX415器件的帧率最大可以达到90帧每秒，但这里显示30帧是因为RV1126的isp以及硬件视频编解码模块性能支持的就是30帧每秒，所以在驱动程序上就设置成了30帧

（也就是说：视频编解码性能只支持30帧，所以需要30帧，故驱动设置成了最高30帧，但摄像头硬件最高支持90帧。由于驱动设置成了最大是30帧，所以在应用层不论怎么设置帧率，最大仅只支持30帧。）

crop.bounds:(12,16)：crop表示裁剪，bounds表示边界的意思。所以crop.bounds:(12,16)表示可以裁剪掉的 边界的 一些像素。

我们上面分析知：从Sensor出来后，总线上传输的像素就是3864x2192，然后就被裁剪了：

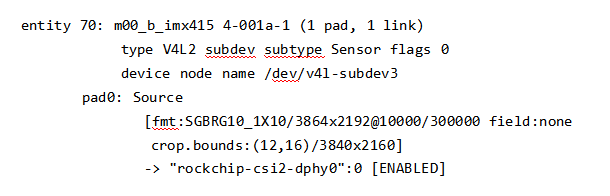


也就是说:从原来的3864长度左右各裁剪了12个像素。

从原来的2192宽度上下各裁剪了16个像素。

经过裁剪之后，最终得到的图像的分辨率就是3840x1260：

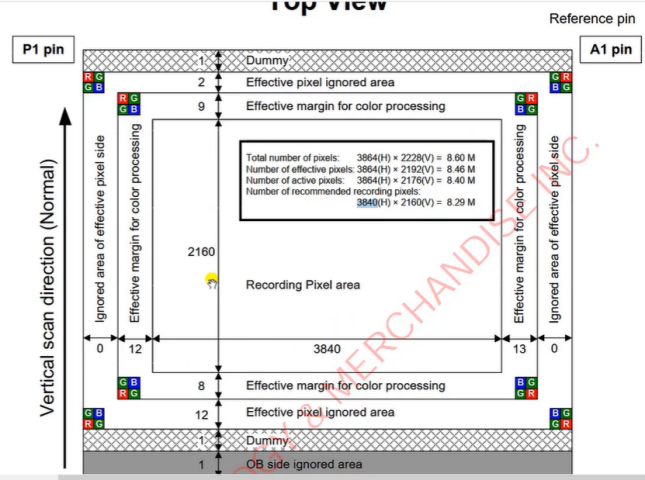
并且上面的手册里，推荐的有效像素也是3840x1260。



也就是说它发送到 rickchip-csi2-dphy0的时候像素是3840x1260的。

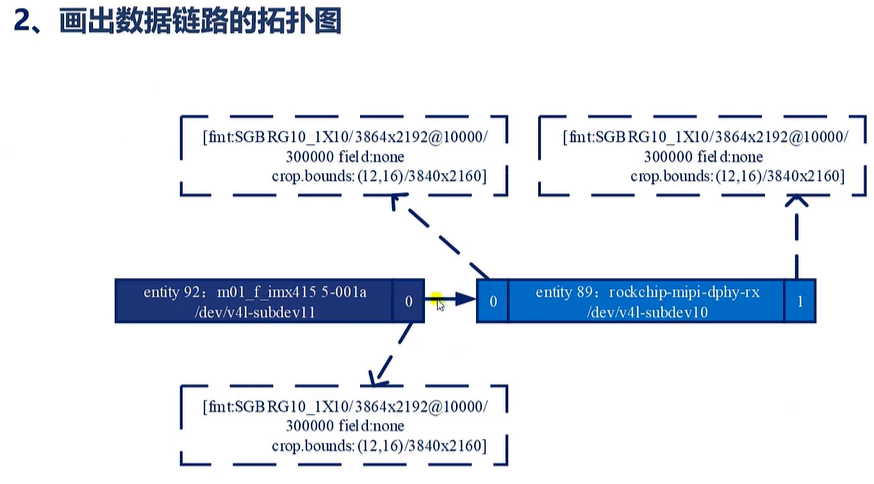
/\*

这个裁剪与摄像头硬件设计有关，了解即可：

\*/

/\*

在他的图中就是：



\*/

\*/