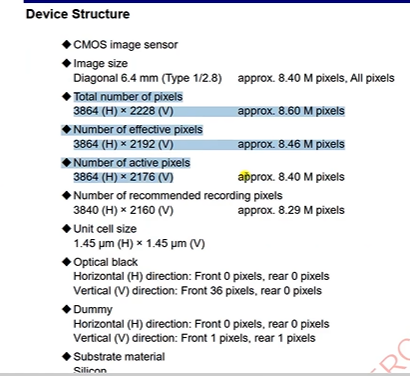


这四路节点支持全分辨率输出，也就是说sensor输出多大，这几个节点就输出多大的数据。

IMX415硬件的Data序



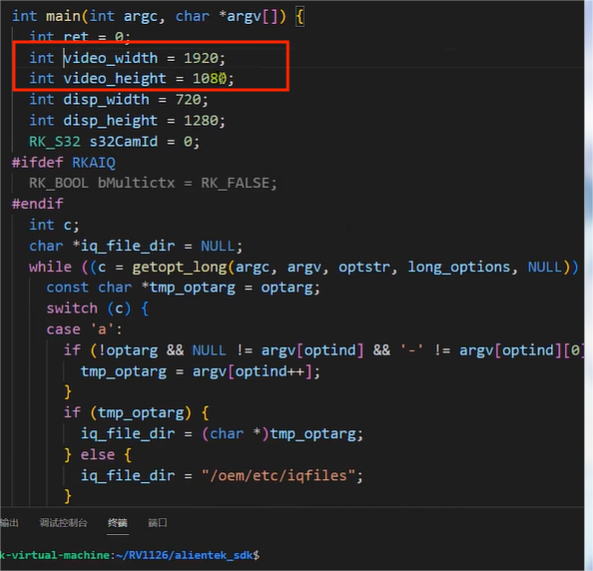
全分辨率：3864\*2228

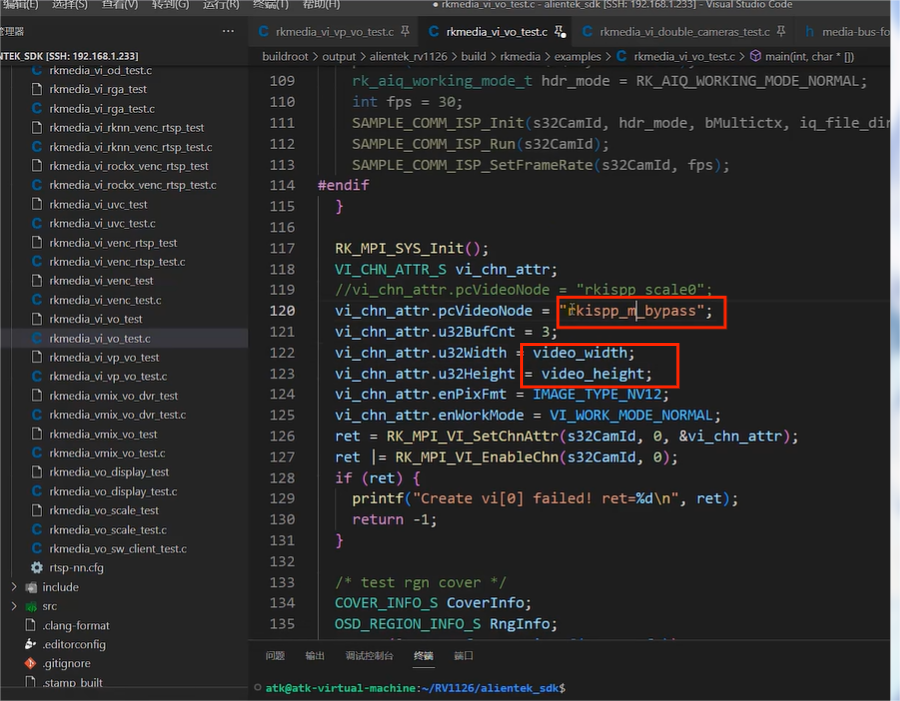
有效分辨率：3864\*2176

推荐使用的有效分辨率是：3840\*2160

在SDK的驱动下，配置的sensor出来的分辨率就是3840\*2160

所以这四个节点对于IMX415来说能支持的全分辨率的图像 大小就是3840\*2160个像素。



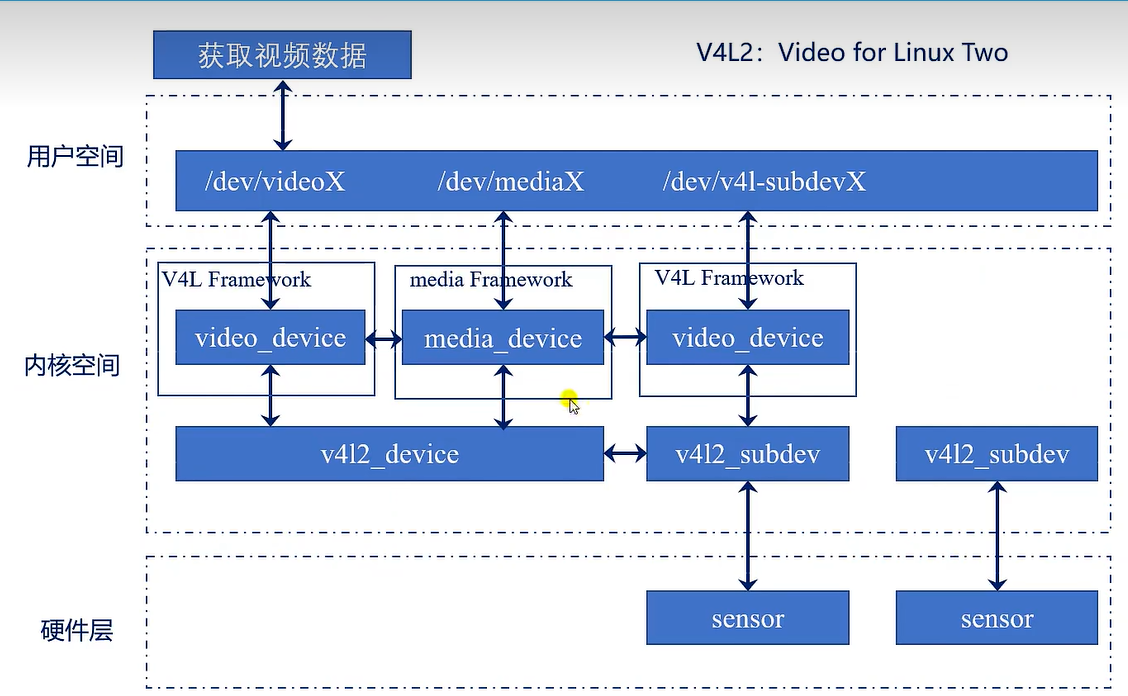


也就是说：当pcvideoNode被修改成rkispp\_m\_bypass输出时，是不支持修改分辨率的，也就是默认使用3840\*2160，但后面修改了u32width为video\_width(1920)和u32Height为video\_height(1080),也就是说修改成了1920\*1080。

也就是说vi模块这里对图像进行缩放操作了，对图像设置了分辨率。

（rkispp\_m\_bypass节点不支持缩放，但使用了vi模组进行了缩放。那么会报错）

对应支持的输出格式（NV12/NV16/YUYV等）如上面首图。

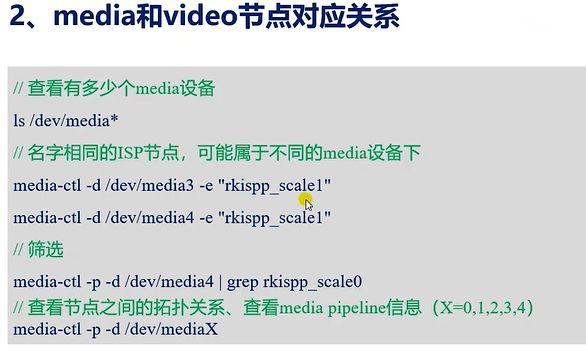


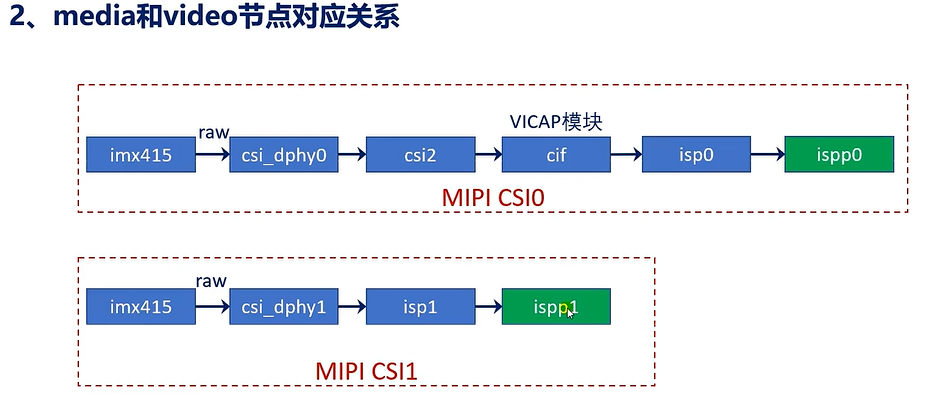
通过Video字符设备控制接口 就可以获取视频数据。

通过媒体设备mediaX可以获取媒体设备的拓扑结构，可以控制子设备间的数据流向。查看拓扑结构的命令就是：



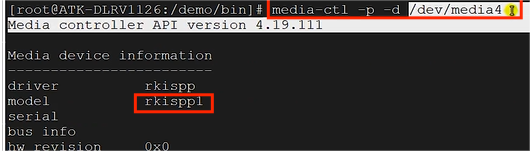
在应用层我们操作对应的video节点，来获取视频流。ISPP节点可以成为媒体节点。



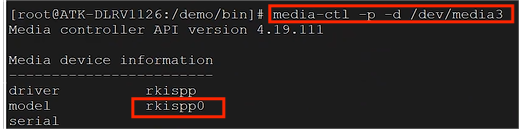


MIPI CSI0使用上面的链路，MIPI CSI1使用下面的链路。

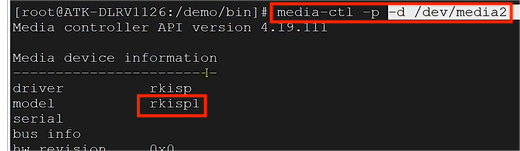
media-ctl -p -d /dev/media4 ispp1 ——CSI1



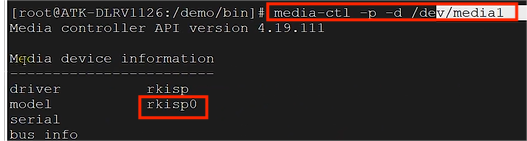
media-ctl -p -d /dev/media3 ispp0 ——CSI0



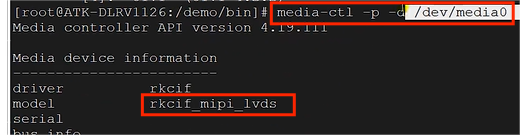
media-ctl -p -d /dev/media2 isp1 —— CSI1



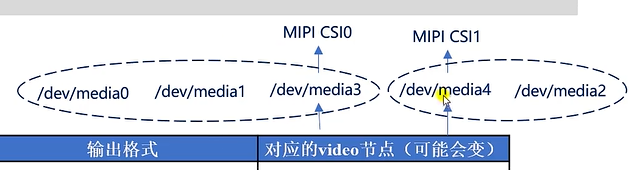
media-ctl -p -d /dev/media1 isp0 —— CSI0



media-ctl -p -d /dev/media0 cif —— CSI0



所以才有：



（就是通过上面的信息和设备树的信息来看的：

IMG_256）