

1. 注意事项
   1. 输入ISP的图像，宽度要满足16对齐，高度要满足8对齐。
   2. 使用IMX415模组时，ISPP最大只能输出3840x2160分辨率大小的图像
   3. 使用IMX335模组时，ISPP最大只能输出2592x1944 分辨率大小的图像
   4. 如果要修改链路的分辨率，在整个ISP链路中，后级的链路不能比前级链路的分辨率还要大，分辨率应该要满足

Sensor==MIPI\_DPHY>=isp input>=isp output，默认是满足这个条件的.不建议去修改，因为容易出问题。

* 1. 当ISP以bypass模式获取Sensor RAW图(如RGGB,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

(见RK手册的： Raw数据存储格式)

* 1. ISP只有2路输出，分别是MP和SP，MP对应的video设备可以出RAW图，SP是不能支持RAW图输出的。
  2. 如果Sensor输出RAW格式的图像，在v4l2命令抓图时，要想得到比较好的图像，需要开启ispserver。（也就是说如果摄像头模组输出的raw格式的图像那么后期如果我们要在ispp节点，通过v4l2命令抓取图片时，我们需要开启ispserver,开启后才能抓取到比较好的效果图，ispserver就是3A（自动对焦，自动曝光，自动白平衡）的守护进程。）
  3. 抓图的时候，注意rkispp\_m\_bypass、rkispp\_scale0、rkispp\_scale1和rkispp\_scale2的缩放倍数和最大宽度范围。
  4. 抓图保存的文件格式，没有什么要求，你写为cif.raw、bypass.nvl2、bypass.out、bypass.bin等等都是可以的
  5. 抓取rkispp\_m\_bypass、rkispp\_scale0、rkispp\_scale1和rkispp\_scale2的图片时，要开启ispserver，抓取MP和SP的图片时，不用开启ispserver。
  6. 当采集ISPP对应的节点图像时（rkispp\_m\_bypass、rkispp\_scale0、rkispp\_scalel和rkispp\_scale2），确保链路下这些节点的状态是这样子的：rkisp-bridge-ispp必须是ENABLED的（1表示[ENABLED]，0表示disable[])。开发板下默认rkisp-bridge-ispp是ENABLED的！



如果不是开启的状态，可以执行如上红色的命令来开启。上面的命令表示让rkisp-isp-subdev 的端口2 到 rkisp-bridge-ispp的端口0的链路开启。

1. v4l2-ctl抓图命令：



前面使用的命令就是这两个工具下的命令，也就是说v4l-utils工具包含media-ctl和 v4l2-ctl工具。

media-ctl命令主要就是对媒体设备进行管理。可以用来查看媒体设备的拓扑结构。可以查看媒体设备的拓扑结构中各个节点的格式，还有图像的大小。以及链路情况。

V4l2-ctl用于管理video设备，可以用来设置video设备的图像像素格式，还可以申请buffer和取消buffer。可以打开或关闭video节点的数据流。以及抓图等。（v4l2-ctl --list-formats-ext --device /dev/videoX 命令就是用来查看某一个video节点支持的像素格式）

在Ubuntu下安装v4l-utils工具的命令就是：sudo apt-get install v4l-utils

在开发板中已经有这个工具，这个是通过配置buildroot文件系统来配置这个工具的。

比如：

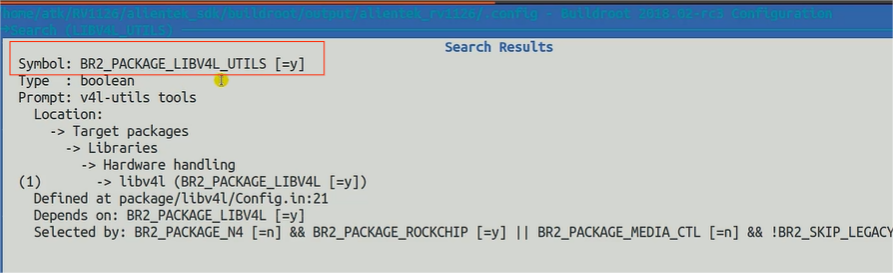
现在来查看buildroot的配置：

在SDK目录下输入：./envsetup.sh

选择 1

执行make menuconfig

然后可以搜索：LIBV4L\_UTILS:



他就是在buildroot下配置了这个工具，“ =y”就表示使能了这个工具。所以开发板就有这个工具了。

抓图命令的使用：

