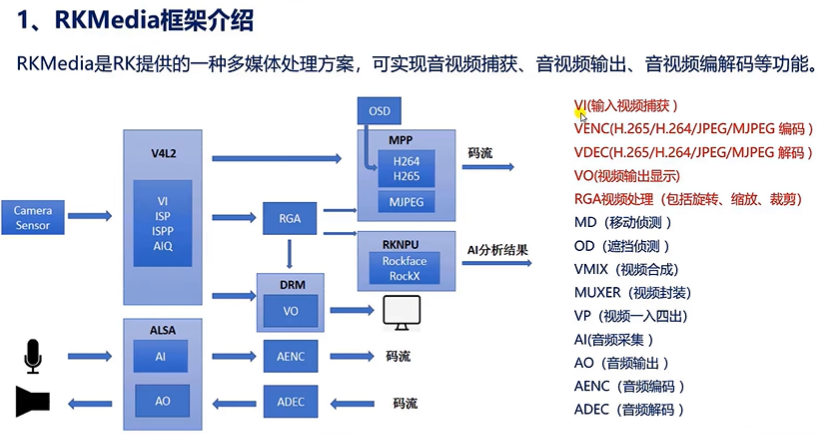


结构图：



上图中的RGA和MPP都是模块，不是组件，这些模块只是组件的一部分功能。这些模块都是对组件做了封装的。

MPP、rga的参考文档：



MPP组件可以用来视频编解码。（这个组件在

/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/external/mpp目录下存在）

其下面的base就是一些基础功能，

Codec:就是视频编码和视频解码相关的例程

Dec:视频解码

里面支持了H263、H264、H265、jpeg、mpg4、m2v、vp8、vp9等格式

Enc：视频编码，相关例程

支持h264、h265、jpeg、vp8编码格式

MPP参考A盘/08.RV1126参考资料/Linux/Multimedia/\_MPP\_CN.pdf

RGA组件

/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/external/linux-rga目录下存在）

是一个独立的2D硬件加速器，可以用来加速点和线的绘制，可以对图像进行拷贝、旋转、翻转、缩放、裁剪、合成、格式转换等等，一些常见的二维图像操作。

Samples:RGA组件相关的一些例程：

RgaRotation:图像旋转相关的例程

RgaCopy565ToRgba等：图像格式转化

RgaCopyScale：图像按照一定比例进行缩放的例程

RgaColorFill：对颜色进行填充的例程

RgaMirror:对图像进行水平翻转操作的例程。

RgaUserSpace:通过 “”的方式申请buffer内存的例程

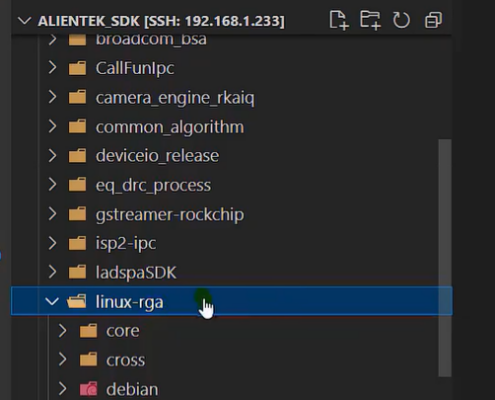
rgaClit:裁剪图像的例程

RGA参考A盘/08.RV1126参考资料/Linux/Multimedia/\_RGA\_.pdf

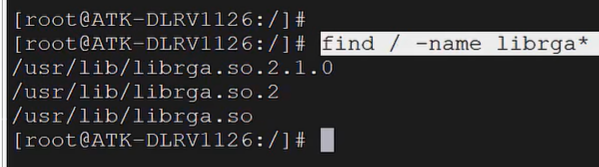
如果要单独调用RGA组件库实现叠加，可以在串口端输入命令搜索：

fine / -name librga\*

搜索出来的就是rga组件的库。



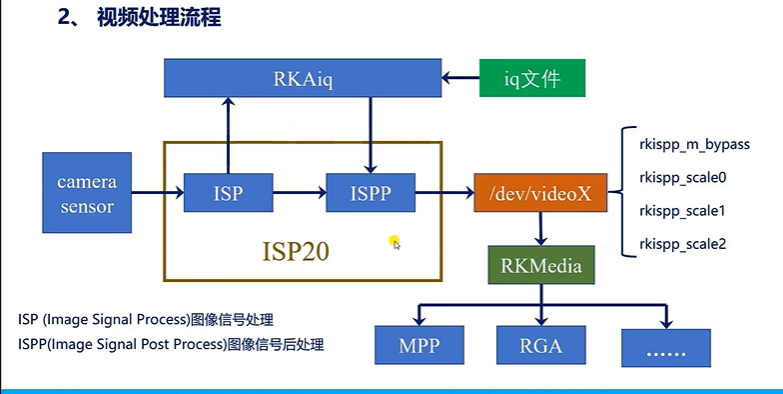
当模块不能实现功能，需要调用组件时：



使用该命令搜索，如上图，就是搜索出来的rga的库。

在结构图中的ISP和ISPP就是ISP20中的模块，ISP20主要就是对sensor采集到的raw图像进行优化。

VO模块就是用于显示图像的，VO模块其实有对DRM进行封装。DRM就是Linux下的主流的显示框架。



由于摄像头的传感器在拍摄彩色图像时用到了Bayer阵列，

故也称raw为Bayer raw。一般bayer阵列分为四种：

BGGR 、 RGGB 、GBRG 、GRBG

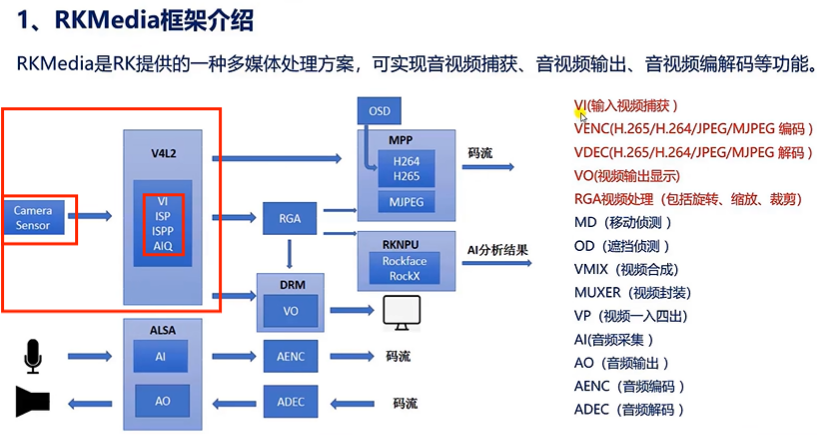
这四种阵列的差别就是像素排列的顺序是不一样的。

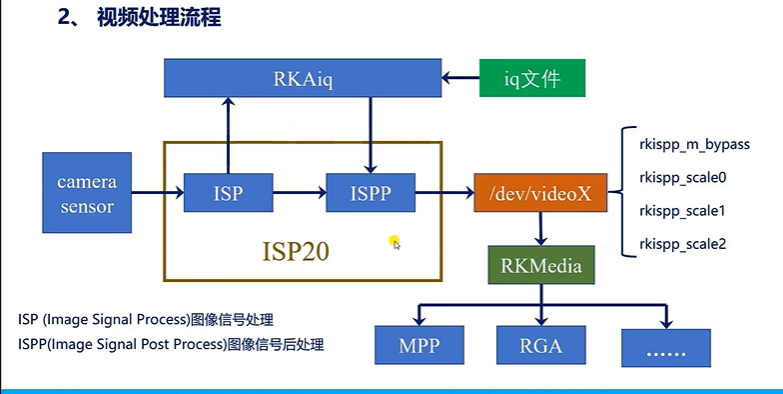
由于摄像头的sensor输出的1是bayer格式的原始图像，屏幕不能直接显示，所以需要先经过开发板内置的ISP进行处理，ISP和ISPP都是ISP20内部的模块，ISP20会对bayer格式的图像进行算法处理,经过算法处理之后，最后就输出了RV1126空间域 或 YUV空间域 或 其它色彩空间域 的图像。给后端的视频采集单元（如RKMedia的VI模块，来获取视频数据）

对于ISP20还有一点需要注意：RKAiq会将优化图像的一些算法参数传递给isp20，isp20的硬件就根据这个算法参数来优化图像质量，其中RKAiq会去获取iq文件。iq文件包含了优化图像要用到的相关算法参数。所以iq文件里的参数，如果不够好，可能会导致isp20优化得到的图像效果不佳。

总结：

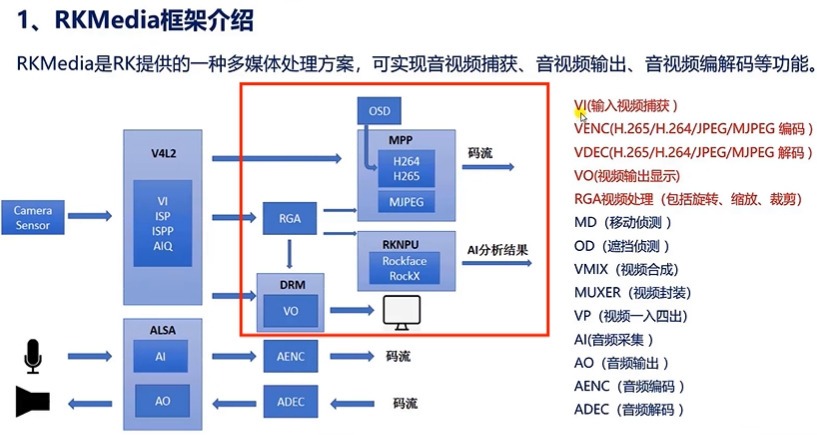
首先分析左上角的部分：

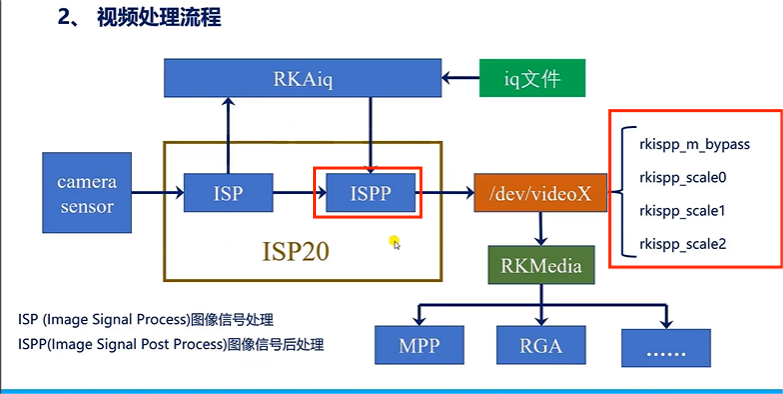




Isp20根据RKAiq传递的参数进行初步优化，初步优化好raw数据后，就可以从video节点输出出优化好的图像了（输出数据流/码流）。码流从videoX设备节点输出，videoX输出节点的路径总共有如图4路。那么RKMedia的VI模块就可以从这四路输出节点获取码流。然后RKMedia就可以对码流进行进一步的操作，与ISP20的初步优化不同，RKMedia进行的处理都是一些比较常见的处理。（例如：使用RKMedia的RGA模块对图像旋转、使用RGA模块对图像进行缩放，修改分辨率、图像格式转换：将图像的YUV格式转换为RGB格式。所以RKMedia是根据我们的需求对图像进行进一步的后续处理。而RKMedia的处理能力（使用模块）实质上就是依赖于MPP、RGA这些组件）。

经过上面的分析后，现在分析右上角部分：





1.上面提到：RKMedia模块，是从ISPP的四路输出节点获取数据，那么RKMedia从ISPP获取数据流之后，如果不进行处理就直接显示的话，RKMeida的VI就可以直接把数据流发给VO模块，来进行显示，那么屏幕就可以显示图像了。

2.RKMedia模块获取到数据流之后，将码流丢给RGA模块，经过RGA处理之后，可以进而传给RKNPU（专门跑AI算法的）进行AI模型的推理，在SDK中就有提供Rockface（需要瑞芯微授权之后，可以实现人脸识别功能）以及RockX（跑AI算法，正点原子发布的AI例程就是基于这个组件来做的）这两个组件

3.从VI输出的码流，经过RGA处理之后，从RGA出来的码流也可以直接给VENC模块进行压缩编码（H264/H265/ffmpeg格式的文件），编码的功能就是用到MPP这个组件相关的接口。

上面的OSD的作用就是在屏幕上显示一些文字/图形。例如录像上面显示的时间戳就可以通过OSD来实现。

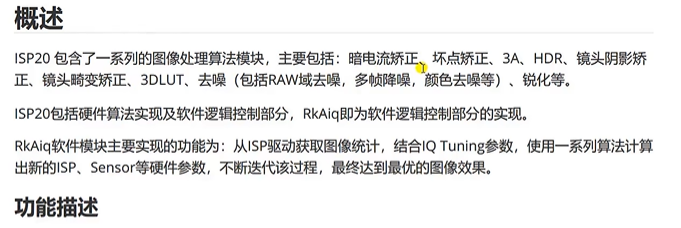
1. 从VI输出的码流也可以不经过RGA，而是直接丢给VENC模块来进行编码，编码之后就可以将码流输出。

要对视频进行处理，完全可以使用MPP、ISP、RGA等组件来完成。为什么正点原子的AI例程要用瑞芯微提供的RKMedia来做而不是直接调用组件？因为要完成音视频的采集和处理，就需要学MPP、ISP、RGA、DRM等等的组件使用方法。并且这些组件涉及的知识点不止于此，会涉及许多专业知识点和词汇。对于入门来说难度极大。对于初学者来说不知道什么时候才能完成视频的采集和显示。若以RK封装了这些组件来给我们使用。我们调用RKMedia相关的接口就可以实现视频的采集、处理显示等操作。从而简化开发难度。

建议后续研究FFmpeg和gstreamer。这两个就是开源的免费项目。ffmpeg是音视频处理的通用方案，具有广泛的用途。对于大多数的视频编解码，使用ffmpeg就可以了。

ISP的一些相关资料：

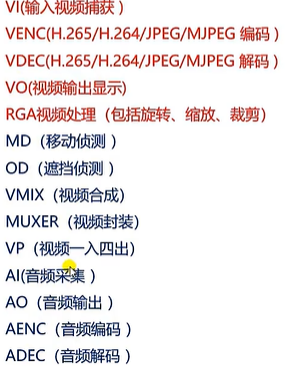
A盘/08、RV1126参考资料/RV1126\_RV1106/Camera/





RKMedia主要做一些常见的视频处理：经过ISP20后得到的图像需要旋转180度、图像的分辨率由1080p变成720p。也可以通过RKMedia的RGA模块对图像进行缩放操作、图像格式由YUV格式变成RGB格式。所以RKMedia的处理是根据我们的需求进行的一种后续的处理。

RKMedia的处理，依赖的就是这些组件：



OSD的作用就是在视频图像上面显示一些文字等。



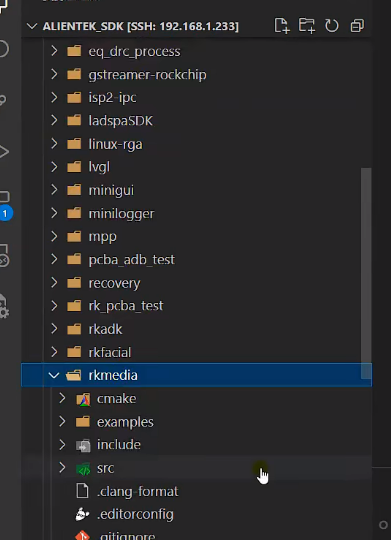
如果是深入研究音视频 特别是偏向视频处理 建议研究MPP（HiMPP/RKMPP）

如果是通用领域可以研究ffmpeg、gstreamer（免费的多媒体开源项目）。特别是ffmpeg，它是音视频处理的通用方案，具有广泛的用途，对于大多数的视频编解码使用ffmpeg就可以了。

学习RKMedia的参考资料：



学习参考文档的同时，可以参考代码例程：



MPP、rga的参考文档：



ISP的一些相关资料：



学习RKMedia的参考资料：

