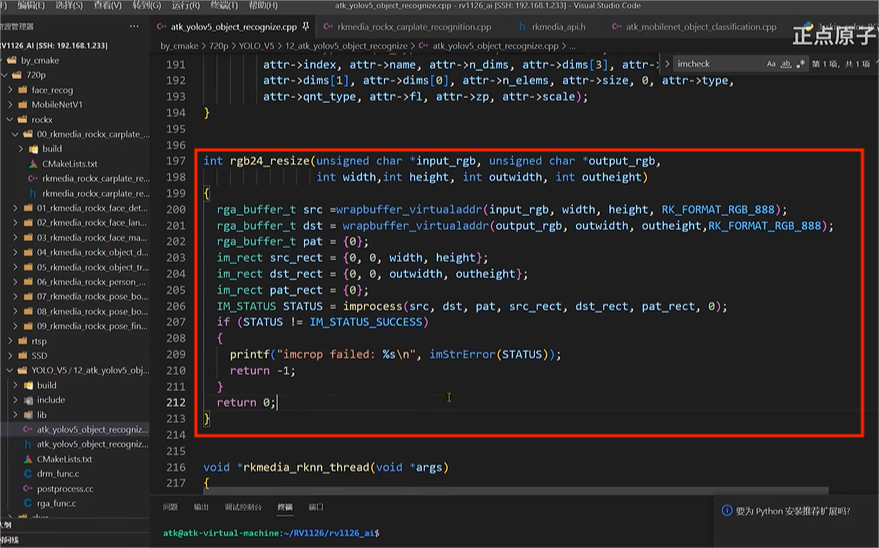
前面已将RGA组件的功能介绍了一边。可以知道，其内部实际上就是调用API来完成对图像的处理的。

/\*

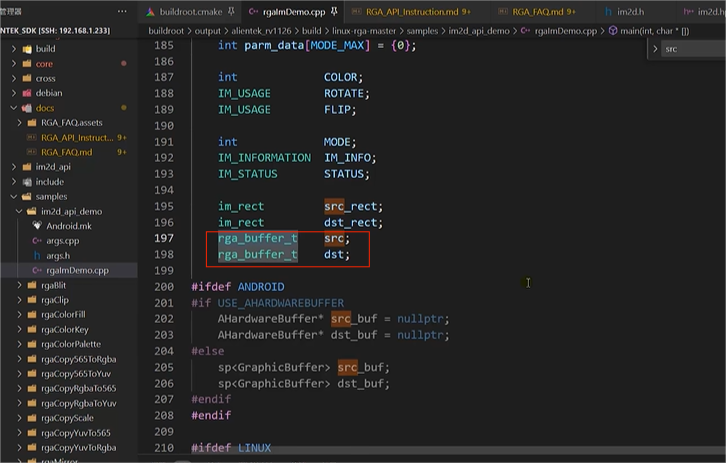
如下：rgb24\_resize函数就是用来修改rgb888图像的大小的。



\*/

/\*

如下结构体，就是用于定义rga的buffer类型的：



\*/

/\*

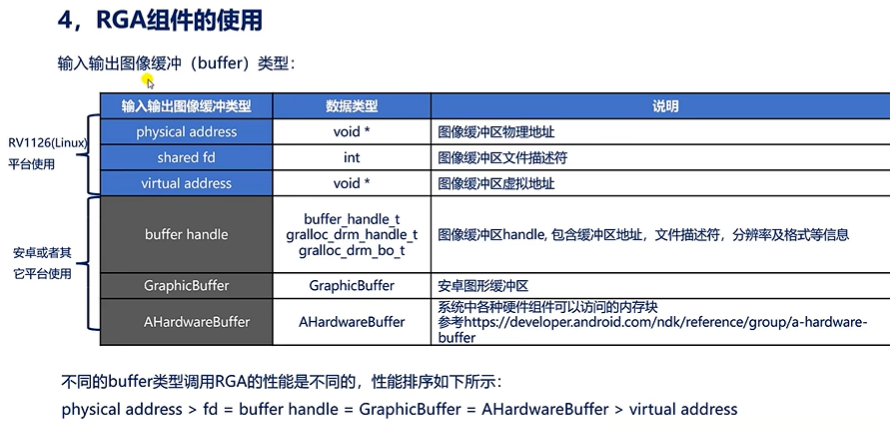
下面就来看一下rga支持的输入输出图像缓冲类型：也就是说在rga中无论进行什么操作，都应该先将输入输出图像的缓冲类型转化为统一的rga\_buffer\_t类型。转化为统一的这种结构体。

转化为统一的图像缓冲类型一般就是在相同的缓冲类型中进行操作的。

这里就列出了rga中可能会用到的图像缓冲类型：

前三个是Linux下会用到的。Rv1126跑的就是Linux操作系统。所以使用的主要就是前三种buffer类型。后面的三种一般是Android系统会用到的。

前三种buffer名称不同，作用也就不同。调用不同的buffer类型，调用rga的性能是不一样的。性能最好的就是图像缓冲区物理地址。其次是图像缓冲区文件描述符。最后才是图像缓冲区虚拟地址：



\*/

/\*

如何声明buffer类型?

如下列出了3种buffer的声明方式：

图像缓冲区虚拟地址buffer:



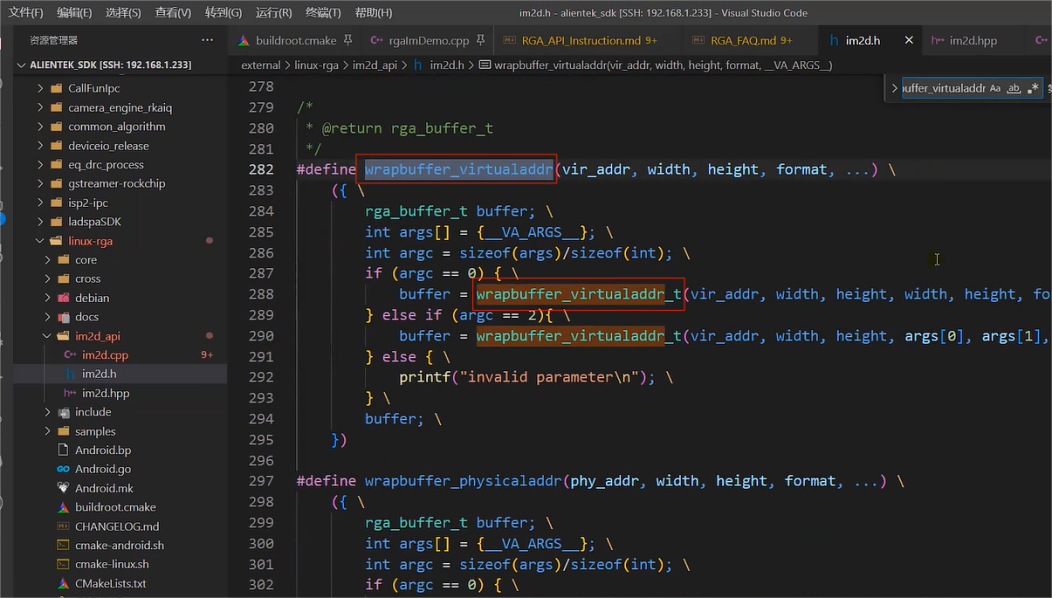
图像缓冲区物理地址buffer：wrapbuffer\_physicaladdr

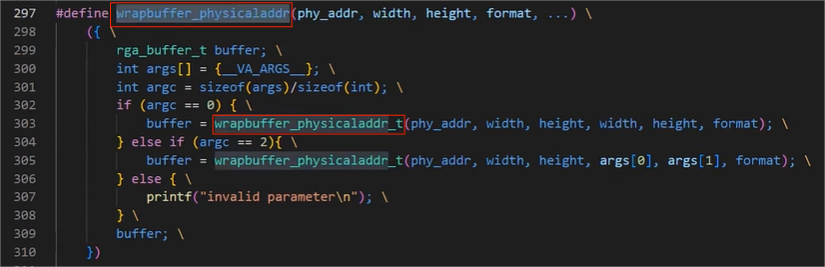


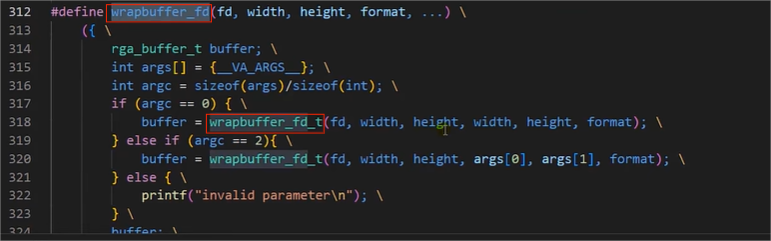
图像缓冲区文件描述符buffer：



这三种buffer声明于：im2d.h文件中：







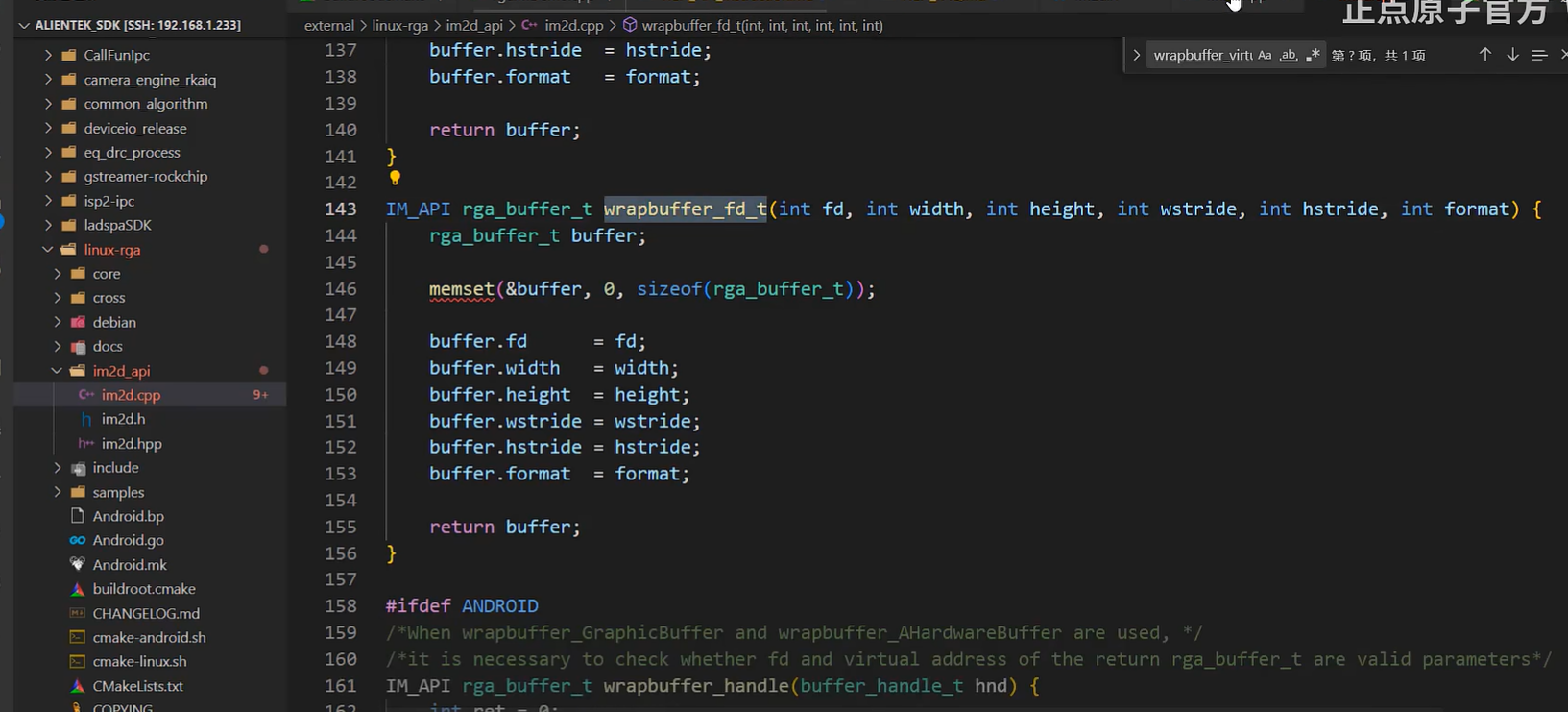
可以看到其内部又调用了对应的api函数：

虚拟：wrapbuffer\_virtualaddr -> wrapbuffer\_virtualaddr\_t

物理：wrapbuffer\_physicaladdr -> wrapbuffer\_physicaladdr\_t

文件：wrapbuffer\_fd -> wrapbuffer\_fd\_t

这几个api又定义于：im2d.cpp文件中：于PPT参数一致。



\*/