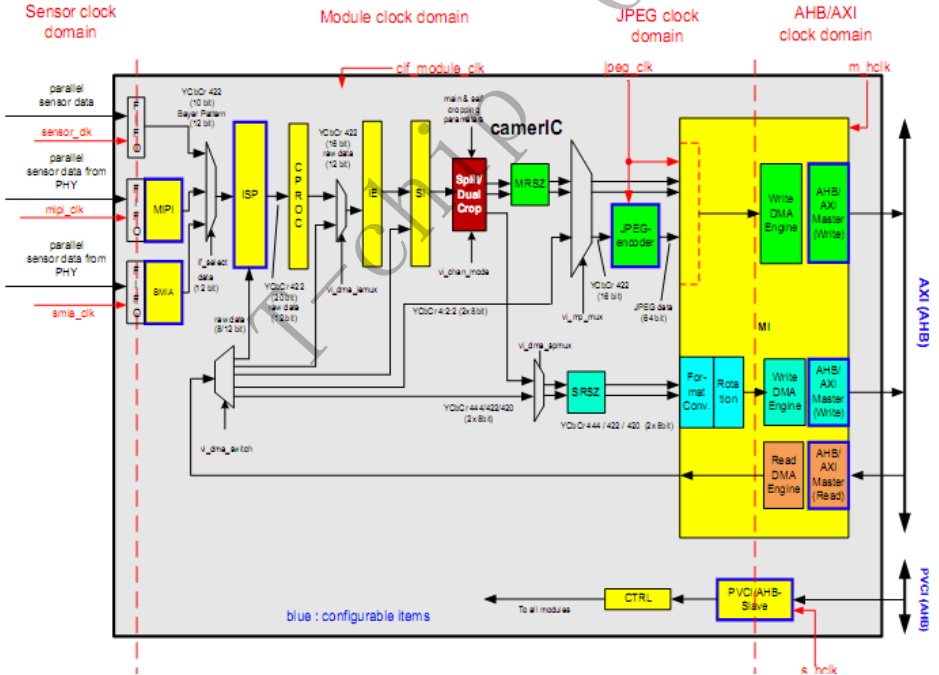
# rk3288 isp driver分析

1. **isp hardware unit**

下图摘自rk3288的trm文档，是一个isp的整体框图，如下所示：



ISP有以下的子单元：

⚫MIPI serial camera interface

⚫Image Signal Processing (ISP)

⚫Color Processing (CPROC)

⚫Image Effects (IE)

⚫Superimpose (SI)

⚫Luminance /Chrominance Splitter (Y/C Split)

⚫Main and Self Crop (Dual Crop)

⚫Main Resize (MRSZ)

⚫JPEG Encoder

⚫MPMUX for selection of main path data flow

⚫SPMUX for selection of self-scaler input data

⚫Self Resize (SRSZ)

⚫Memory Interface (MI) including YCbCr to RGB conversion for self-picture and image  
rotation

⚫Control Unit

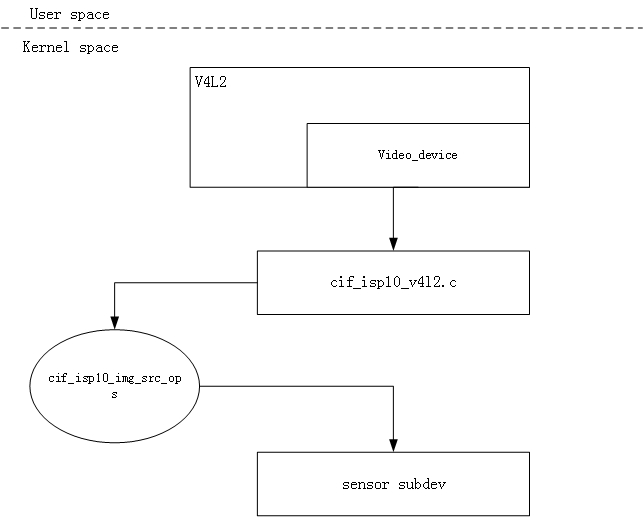
Rk3288有两个图像输入接口，一个为MIPI，另一个为CIF，通过一个MUX单元控制INPUT。图像进来后，经过各个单元的处理，最后有两种路径输出图像到内存，分别为MP和SP路径。

MP路径：图像经过MRSZ后，经过MI传输到内存buf，在这个路径，MI没有进行颜色空间转换，最后输出到内存的为YUV格式；

SP路径：图像经过SRSZ，经过MI单元里面的颜色空间转换单元，可以设置输出到内存中像素的格式，从而可以直接输出到framebuffer，直接显示，这一路，又叫，overlay显示。

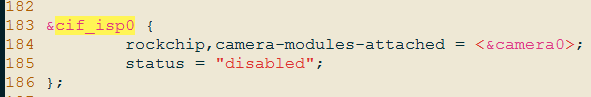
1. **ISP驱动框架**

Rk3288的isp驱动是基于intel mobile修改的，大部分的地方一样，采用标准的v4l2框架，各大块之间的关系如下图所示：

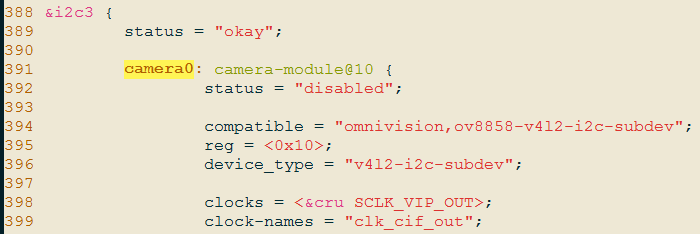


isp驱动采用了subdev，但没有采用subdev框架，取而代之的是，通过设备树的方法。cif\_isp10\_v4l2.c找到对应client的subdev，并保存一个指向subdev的指针，这样，cif\_isp10\_v4l2.c通过该subdev指针，直接调用到sensor subdev的底层函数。

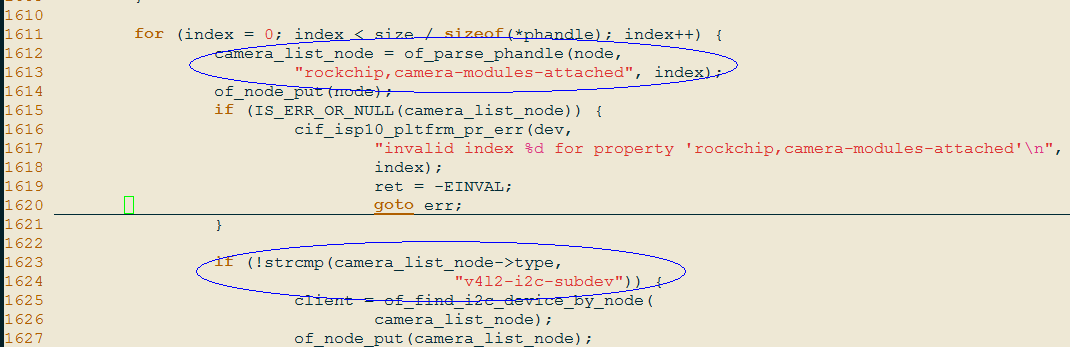
cif\_isp10\_v4l2.c对应的设备树节点有如下一个属性，如下图所示：



sensor i2c client的设备树节点如下所示：

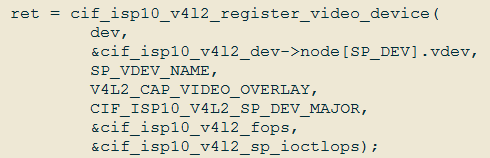


cif\_isp10\_v4l2.c在驱动中，找到sensor的方法如下图所示：



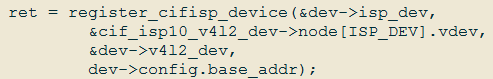
1. rockchip,camera-modules-attached位cif-isp0设备树节点的一个属性，该属性引用了一个sensor i2c client。驱动通过of提供的接口，获取到i2c client；
2. 对应于sensor i2c client的驱动，在probe函数已经初始化好subdev结构体变量，并把subdev保存到client的drvdata。cif\_isp10\_v4l2.c在序号1中获取到client，然后获取client的drvdata，即可得到subdev。
3. **注册video device**

cif\_isp10\_v4l2.c注册了4个video device，每个video device有不同的用途，各个video device的注册方式如下所示：

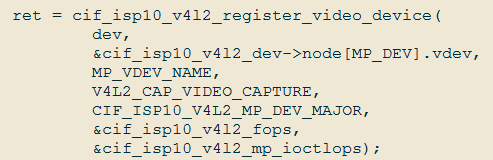


SP\_DEV video device

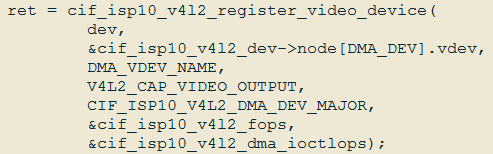
SP\_DEV video device是一个处理SP路径的video device，通过该video device，可以输出图像到显示屏。其中，cif\_isp10\_v4l2\_fops为video device的fops，cif\_isp10\_v4l2\_sp\_ioctlops为video device的ioctl\_ops。



第二个video device为一个ISP\_DEV，这个video device主要是用于控制ISP（是第一节图中的ISP单元）里面的单元。通过video device的iotcl函数，控制各个单元的开与关，需要打开哪个、或者需要关闭哪个单元，都可以通过ioctl接口关闭对应的单元。



上图注册的是一个MP\_DEV video device。该video device用于处理MP路径的图像，与SP\_DEV差不多，但处理过程不一样，最后也是输出到内存BUF。



该video device用于处理从内存获取图像，然后放到isp处理，最后经过MP或者SP路径输出，简而言之，ISP单元不仅可以处理来自内存中的图像，而且还可以处理来自采集接口的图像。

1. **request buffer说明**

SP和MP有continus和sg两种方式申请内存buf。申请内存buf是在vb2\_queue的mem\_ops完成，mem\_ops的初始化是根据DMA\_CONTINUS和DMA\_SG来决定的，如果两个宏同时打开，那么采用的是DMA\_SG的mem\_ops。

采用sg dma需要IOMMU的支持，IOMMU主要用来把分散的CPU物理地址映射到连续的CPU外设isp的外设连续地址，采用sg dma效率高，而且减少系统保留大量连续内存的压力。

1. **in the end**

了解了ISP硬件框架和驱动框架，对图像的数据传输流程和驱动的调用流程都有了一个大致的了解，那么剩下的就是具体接口的实现方式。

open->

req\_buffer->

mmap->

set\_fmt->

qbuf->

stream\_on->

dqbuf->

一系列接口的实现。