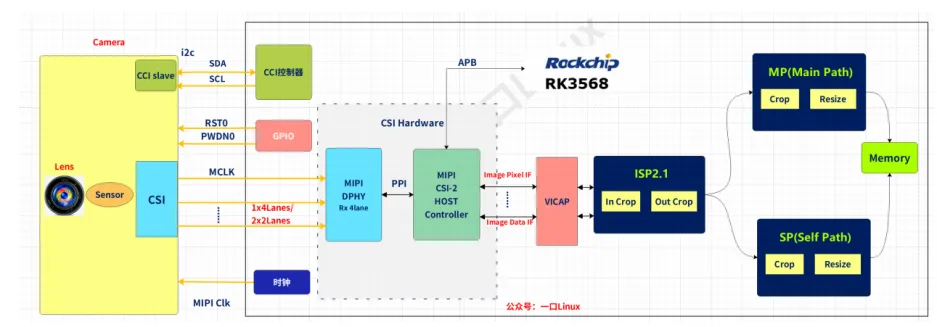
## rk3568之IMX415摄像头驱动和移植实战开发第二期：

1、介绍一下cmos摄像头整体的数据流向处理框架：

* usb
* dvp
* mipi

2、写imx415驱动代码框架

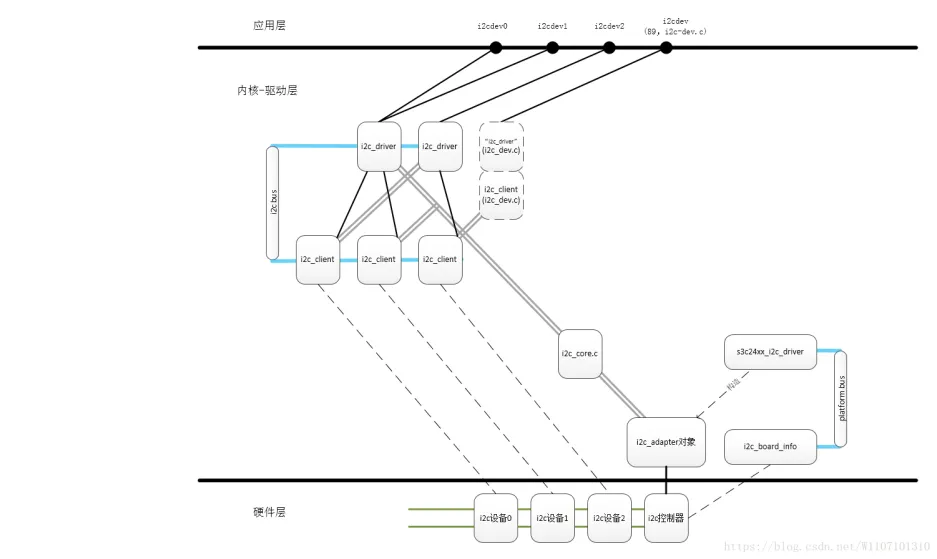
为什么要去写这个sensor驱动源码，那是因为，有时候sdk里面没有对应的sensor驱动或者sensor原厂没有给对应的驱动源码



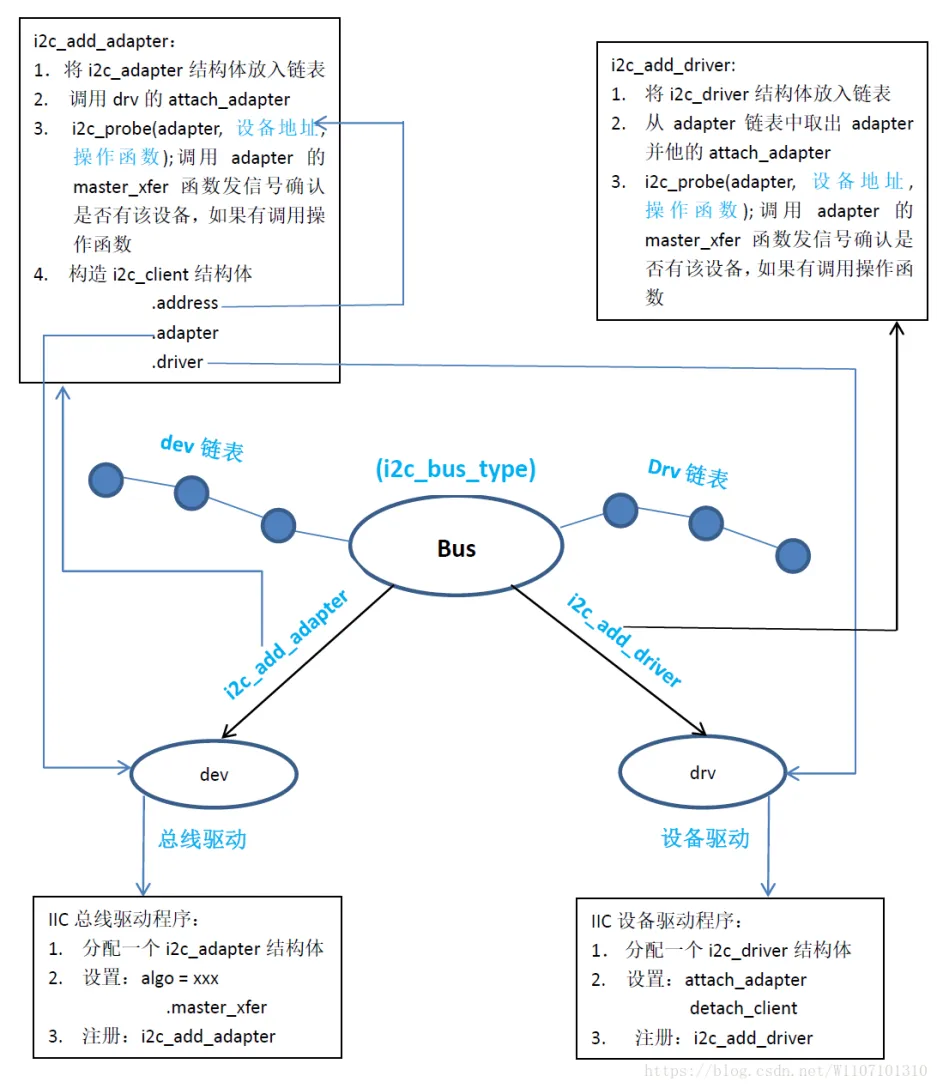
soc通过I2C与摄像头通信，通过CCI控制器控制Sensor做各种处理。

CSI主要起数据传输的作用。传输的时候会涉及到MIPI协议。

SOC这里数据接收是通过DPHY（而不是CPHY），然后再通过PPI（物理层的一个协议）再传输到CSI,经过处理再传到瑞芯微的VICAP（视频输入，原称为CIF），VICAP得到的数据再经过ISP处理，然后输出到Main层面，然后用MIPI屏幕显示。



参考：[https://blog.csdn.net/W1107101310/article/details/79871029](https://blog.csdn.net/W1107101310/article/details/79871029" \t "_blank)



怎么去写这个imx415 sensor驱动，

在rk的芯片平台上，sdk里面其实已经给了很多sensor驱动代码框架，

那么我们去写新款的sensor驱动源码或者如果有这个sensor源码，如果有驱动有问题的话,

你也可以按照这个方式去快速掌握sensor驱动代码流程

这里面会涉及到v412，这个我们去详细讲解，讲解以官方wiki为主。

#include <linux/clk.h>

#include <linux/device.h>

#include <linux/delay.h>

#include <linux/gpio/consumer.h>

#include <linux/i2c.h>

#include <linux/module.h>

#include <linux/pm\_runtime.h>

#include <linux/of.h>

#include <linux/of\_graph.h>

#include <linux/regulator/consumer.h>

#include <linux/sysfs.h>

#include <linux/slab.h>

#include <linux/pinctrl/consumer.h>

#include <linux/version.h>

#include <linux/rk-camera-module.h>

//下面的都是关于media（v4l2）的一些头文件，红色警告不用管

#include <media/v4l2-async.h>

#include <media/media-entity.h>

#include <media/v4l2-common.h>

#include <media/v4l2-ctrls.h>

#include <media/v4l2-device.h>

#include <media/v4l2-event.h>

#include <media/v4l2-fwnode.h>

#include <media/v4l2-image-sizes.h>

#include <media/v4l2-mediabus.h>

#include <media/v4l2-subdev.h>

//这些都是系统自带的头文件。

#define IMX415\_NAME "imx415"

//1.定义和设置i2c\_drv取得结构体

static struct i2c\_driver imx415\_i2c\_driver = {

.driver = {

.name = IMX415\_NAME,

.pm = &imx415\_pm\_ops,

.of\_match\_table = of\_match\_ptr(imx415\_of\_match),

},

.probe = &imx415\_probe,

.remove = &imx415\_remove,

.id\_table = imx415\_match\_id,

};

//2.注册imx415\_i2c\_driver结构体

static int \_\_init sensor\_mod\_init(void)

{

return i2c\_add\_driver(&imx415\_i2c\_driver);

}

static void \_\_exit sensor\_mod\_exit(void)

{

i2c\_del\_driver(&imx415\_i2c\_driver);

}

device\_initcall\_sync(sensor\_mod\_init);

module\_exit(sensor\_mod\_exit);

MODULE\_DESCRIPTION("IMX415 CMOS Image Sensor driver");

MODULE\_LICENSE("GPL v2");

//这就是一个sensor的驱动框架。