本次内容讲解：

1、mipi协议里面的csi-2层次结构定义介绍

2、rv1126之camera v4l2驱动框架介绍：

有sensor I2c 驱动源码

mipi实现驱动源码

isp实现驱动源码

一、csi-2层结构定义：

csi-2层次定义主要分为5部分：

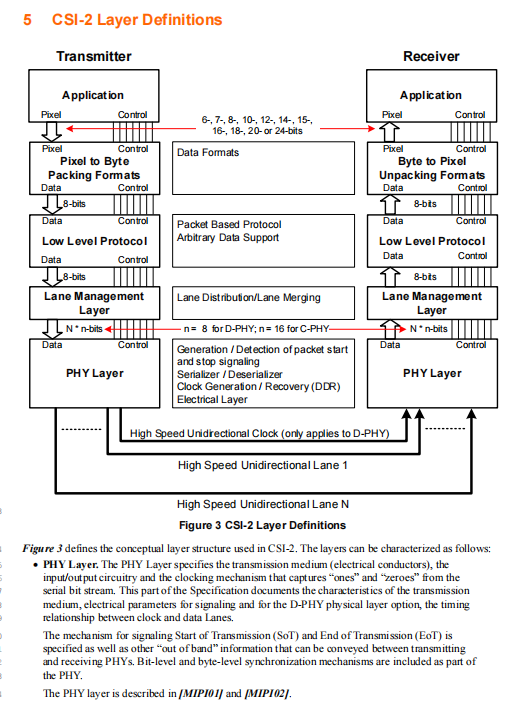
- 1、Application

- 2、Pixel to BytePacking Formats

- 3、Low Level Protocol

- 4、Lane Management Layer

- 5、PHY Layer



1. 物理层(PHY Layer):物理层指定传输介质(电导体)、输入/输出电路和从串行比特流中捕获“1”和“0”的时钟机制。

这一部分记录了传输介质的特性、信令和D-PHY物理层选项的电气参数、时钟和数据通道之间的时序关系，指定了传输开始(SoT)和传输结束(EoT)的信令机制，以及可以在发送和接收物理设备之间传递的其他“带外”信息。位级和字节级同步机制是PHY的一部分（主要就是记录传输媒介）

物理层在[MIPI01]和[MIPI02]中描述

协议层(protocol layer)：协议层由几个层组成，每个层都有不同的职责。CSI-2协议使用主机处理器上的单个接口支持多个数据流

协议层指定如何标记和交错多个数据流，以便每个数据流可以适当地重建。

2、通道管理(Lane Management):

CSI-2是lane可扩展的，以提高性能。数据通道的数量不受本规范的限制，可以根据应用程序的带宽需求来选择。(主要就是做数据的传输)

接口的发送端将(“分发者”功能)字节从传出数据流分发到一个或多个lane。

在接收端，接口从lane收集字节并合并(“合并”功能)它们一起合并成一个重新组合的数据流，恢复原始流序列。对于C-PHY物理层选项，

该层专门向数据通道分发或收集字节对(即16位)。在每车道的基础上加扰是一个可选的特性，在第9.15节中详细说明

协议层中的数据被组织为数据包。接口的发送端将报头和错误检查信息附加到要在低级协议层传输的数据上。

在接收端，报头在低级别协议层被剥离，并由接收端相应的逻辑进行解释。错误检查信息可用于测试传入数据的完整性

3、低级协议，低阶协定(Low Level Protocol.):

低电平协议(LLP)包括为在SoT(传输开始)和EoT(传输结束)事件之间传输的串行数据建立位级和字节级同步的方法。（主要就是完成从传输开始到结束的时候去做一些数据等的一个同步机制，如做字节大小端对齐之类的操作。）

并将数据传递到下一层。LLP的最小数据粒度为一个字节。LLP还包括字节内的位值解释赋值，即“Endian”赋值。

4、像素/字节打包/拆包层(Pixel/Byte Packing/Unpacking Layer)：

CSI-2规范支持具有不同像素格式的图像应用程序。在发送器中，该层在将数据发送到低级协议层之前，将来自应用层的像素打包成字节。

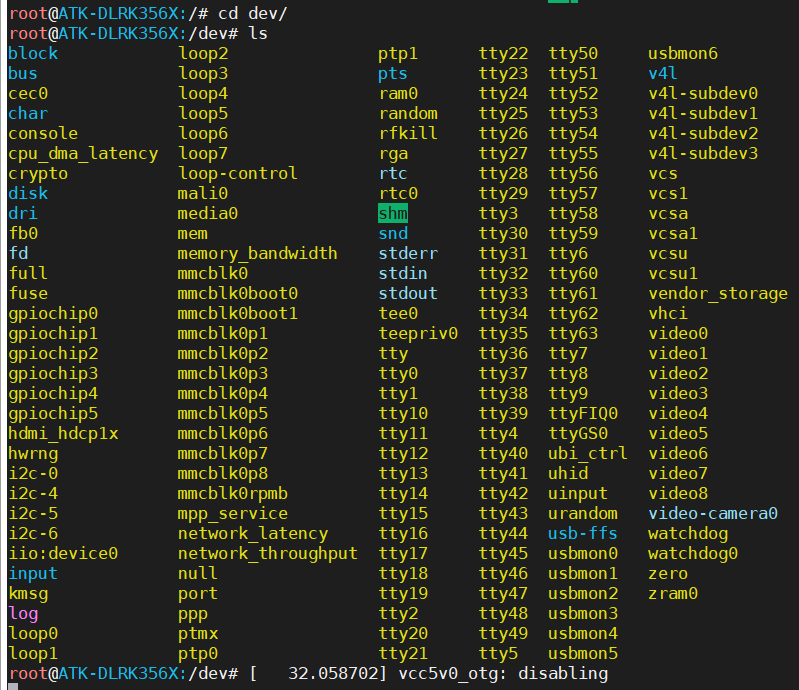
在接收层中，在将数据发送到应用层之前，该层将来自低级协议层的字节解包为像素。每像素8位的数据在这一层传输时保持不变。

5、应用层（Application）：

这一层描述了数据流中包含的数据的高级编码和解释，超出了本规范的范围。CSI-2规范描述了像素值到字节的映射

RK3568的设备：

/dev/ 目录下的设备节点：



媒体设备：

Media0

在RV1126上提供了一个media-ctl的一个拓扑图的介绍：

虽然这个和驱动没有关系，但是可以帮助我们做一些分析，例如下面介绍的：-p打印拓扑结构：

root@ATK-DLRK356X:/dev# media-ctl -h

media-ctl [options]

-d, --device dev Media device name (default: /dev/media0)

If <dev> starts with a digit, then /dev/media<dev> is used.

If <dev> doesn't exist, then find a media device that

reports a bus info string equal to <dev>.

-e, --entity name Print the device name associated with the given entity

-V, --set-v4l2 v4l2 Comma-separated list of formats to setup

--get-v4l2 pad Print the active format on a given pad

--get-dv pad Print detected and current DV timings on a given pad

--set-dv pad Configure DV timings on a given pad

-h, --help Show verbose help and exit

-i, --interactive Modify links interactively

-l, --links links Comma-separated list of link descriptors to setup

--known-mbus-fmts List known media bus formats and their numeric values

-p, --print-topology Print the device topology. If an entity

is specified through the -e option, print

information for that entity only.

--print-dot Print the device topology as a dot graph

-r, --reset Reset all links to inactive

-v, --verbose Be verbose

Links and formats are defined as

links = link { ',' link } ;

link = pad '->' pad '[' flags ']' ;

pad = entity ':' pad-number ;

entity = entity-number | ( '"' entity-name '"' ) ;

v4l2 = pad '[' v4l2-properties ']' ;

v4l2-properties = v4l2-property { ',' v4l2-property } ;

v4l2-property = v4l2-mbusfmt | v4l2-crop | v4l2-interval

| v4l2-compose | v4l2-interval ;

v4l2-mbusfmt = 'fmt:' fcc '/' size ; { 'field:' v4l2-field ; } { 'colorspace:' v4l2-colorspace ; }

{ 'xfer:' v4l2-xfer-func ; } { 'ycbcr-enc:' v4l2-ycbcr-enc-func ; }

{ 'quantization:' v4l2-quant ; }

v4l2-crop = 'crop:' rectangle ;

v4l2-compose = 'compose:' rectangle ;

v4l2-interval = '@' numerator '/' denominator ;

rectangle = '(' left ',' top, ')' '/' size ;

size = width 'x' height ;

where the fields are

entity-number Entity numeric identifier

entity-name Entity name (string)

pad-number Pad numeric identifier

flags Link flags (0: inactive, 1: active)

fcc Format FourCC

width Image width in pixels

height Image height in pixels

numerator Frame interval numerator

denominator Frame interval denominator

v4l2-field One of the following:

any

none

top

bottom

interlaced

seq-tb

seq-bt

alternate

interlaced-tb

interlaced-bt

v4l2-colorspace One of the following:

default

smpte170m

smpte240m

rec709

unknown

470m

470bg

jpeg

srgb

oprgb

bt2020

raw

dcip3

v4l2-xfer-func One of the following:

default

709

srgb

oprgb

smpte240m

none

dcip3

smpte2084

v4l2-quant One of the following:

default

full-range

lim-range

root@ATK-DLRK356X:/dev#

-p打印拓扑结构：

root@ATK-DLRK356X:/dev# media-ctl -p

Media controller API version 4.19.255

Media device information

------------------------

driver rkisp-vir0

model rkisp0

serial

bus info

hw revision 0x0

driver version 4.19.255

Device topology

- entity 1: rkisp-isp-subdev (4 pads, 7 links)

type V4L2 subdev subtype Unknown flags 0

device node name /dev/v4l-subdev0

pad0: Sink

[fmt:SGBRG10\_1X10/3864x2192 field:none

crop.bounds:(0,0)/3864x2192

crop:(12,16)/3840x2160]

<- "rkisp-csi-subdev":1 [ENABLED]

<- "rkisp\_rawrd0\_m":0 [ENABLED]

<- "rkisp\_rawrd2\_s":0 [ENABLED]

pad1: Sink

<- "rkisp-input-params":0 [ENABLED]

pad2: Source

[fmt:YUYV8\_2X8/3840x2160 field:none colorspace:smpte170m quantization:full-range

crop.bounds:(0,0)/3840x2160

crop:(0,0)/3840x2160]

-> "rkisp\_mainpath":0 [ENABLED]

-> "rkisp\_selfpath":0 [ENABLED]

pad3: Source

-> "rkisp-statistics":0 [ENABLED]

- entity 6: rkisp-csi-subdev (6 pads, 5 links)

type V4L2 subdev subtype Unknown flags 0

device node name /dev/v4l-subdev1

pad0: Sink

[fmt:SGBRG10\_1X10/3864x2192 field:none]

<- "rockchip-csi2-dphy0":1 [ENABLED]

pad1: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp-isp-subdev":0 [ENABLED]

pad2: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp\_rawwr0":0 [ENABLED]

pad3: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

pad4: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp\_rawwr2":0 [ENABLED]

pad5: Source

[fmt:SGBRG10\_1X10/3864x2192 field:none]

-> "rkisp\_rawwr3":0 [ENABLED]

- entity 13: rkisp\_mainpath (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video0

pad0: Sink

<- "rkisp-isp-subdev":2 [ENABLED]

- entity 19: rkisp\_selfpath (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video1

pad0: Sink

<- "rkisp-isp-subdev":2 [ENABLED]

- entity 25: rkisp\_rawwr0 (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video2

pad0: Sink

<- "rkisp-csi-subdev":2 [ENABLED]

- entity 31: rkisp\_rawwr2 (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video3

pad0: Sink

<- "rkisp-csi-subdev":4 [ENABLED]

- entity 37: rkisp\_rawwr3 (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video4

pad0: Sink

<- "rkisp-csi-subdev":5 [ENABLED]

- entity 43: rkisp\_rawrd0\_m (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video5

pad0: Source

-> "rkisp-isp-subdev":0 [ENABLED]

- entity 49: rkisp\_rawrd2\_s (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video6

pad0: Source

-> "rkisp-isp-subdev":0 [ENABLED]

- entity 55: rkisp-statistics (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video7

pad0: Sink

<- "rkisp-isp-subdev":3 [ENABLED]

- entity 61: rkisp-input-params (1 pad, 1 link)

type Node subtype V4L flags 0

device node name /dev/video8

pad0: Source

-> "rkisp-isp-subdev":1 [ENABLED]

- entity 67: rockchip-csi2-dphy0 (2 pads, 2 links)

type V4L2 subdev subtype Unknown flags 0

device node name /dev/v4l-subdev2

pad0: Sink

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

crop.bounds:(12,16)/3840x2160]

<- "m00\_b\_imx415 4-001a-1":0 [ENABLED]

pad1: Source

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

crop.bounds:(12,16)/3840x2160]

-> "rkisp-csi-subdev":0 [ENABLED]

- entity 70: m00\_b\_imx415 4-001a-1 (1 pad, 1 link)

type V4L2 subdev subtype Sensor flags 0

device node name /dev/v4l-subdev3

pad0: Source

[fmt:SGBRG10\_1X10/3864x2192@10000/300000 field:none

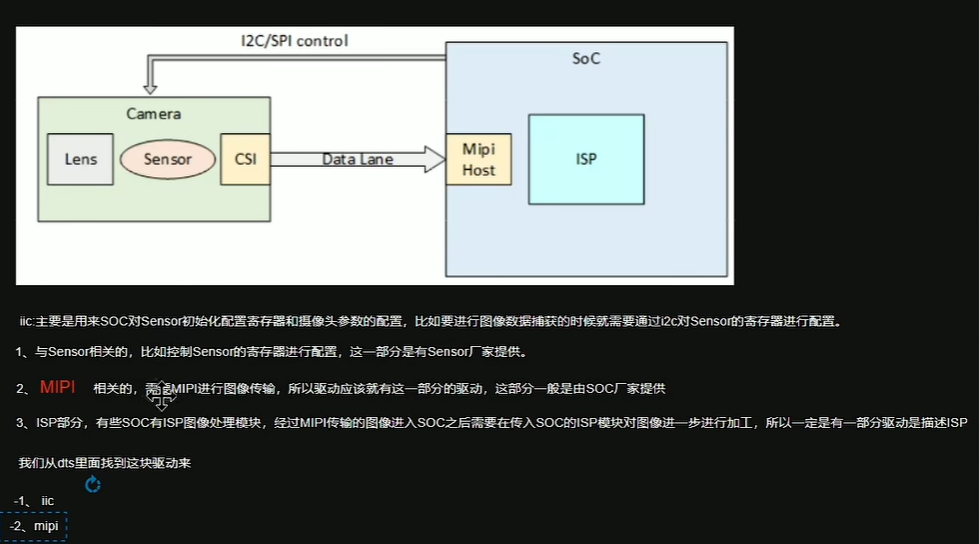
crop.bounds:(12,16)/3840x2160]

-> "rockchip-csi2-dphy0":0 [ENABLED]

root@ATK-DLRK356X:/dev#

1. v4l2驱动框架介绍：

注意这里并不是直接讲的v4l2的架构，而是基于Rv1126这一个Soc的v4l2架构，如果是单纯的v4l2架构还要复杂一些。



这就是硬件的架构：它通过如MIPI协议去做数据传输到Soc，而Soc可以通过I2c或Spi 对sensor做一些寄存器的配置以及摄像头参数的配置等，例如对图像数据捕获的时候就是要通过这个I2c对sensor的寄存器进行配置（刷新寄存器/读寄存器，此时就可以读取到这个I2c的设备）

通过该图我们可以看到，它是遵循MIPI协议的，它通过CSI-2数据通道发送到MIPI的一个主机。通过MIPI协议发送到主机的原始数据，还需要通过ISP的一个驱动进行处理，最终才会得到一个比较合理的像素图形。

再往上层，就可以通过H264编码器做算法的一个编码压缩。

我们主要需要掌握的文件：



注意：

Sensor驱动代码：

在本工程里面sensor模块分析源码是没有techpoint目录的，而是单纯的.c文件，例如gc2053.c文件 就是基于i2c sensor的 可参考源码。

Mipi物理层的源码位于：

/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/kernel/drivers/phy/rockchip/phy-rockchip-mipi-rx.c

Mipi解析模块位于：

/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/kernel/drivers/media/platform/rockchip/cif/mipi-csi2.c

ISP的驱动源码：

所以总共要学的驱动代码有：

1. sensor驱动代码 也就是 i2c代码
2. mipi物理层、解析层驱动代码
3. isp处理 驱动代码

我们通过设备树去找：

关于他的一些配置：

在该目录发现i2c0

/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/kernel/arch/arm64/boot/dts/rockchip/rk3568.dtsi

其头文件为：

#include <dt-bindings/clock/rk3568-cru.h>

#include <dt-bindings/interrupt-controller/arm-gic.h>

#include <dt-bindings/interrupt-controller/irq.h>

#include <dt-bindings/pinctrl/rockchip.h>

#include <dt-bindings/soc/rockchip,boot-mode.h>

#include <dt-bindings/phy/phy.h>

#include <dt-bindings/power/rk3568-power.h>

#include <dt-bindings/soc/rockchip-system-status.h>

#include <dt-bindings/suspend/rockchip-rk3568.h>

#include <dt-bindings/thermal/thermal.h>

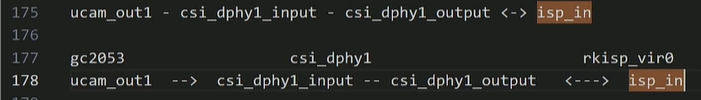
这几个文件位于kernel/include/dt-bindings目录下面。

而#include "rk3568-dram-default-timing.dtsi" 就位于

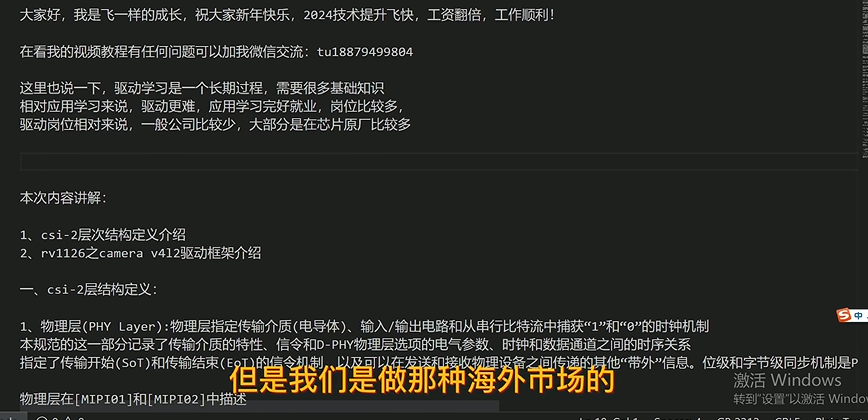
/home/alientek/RK3568/SDK/linux/rk3568\_linux\_sdk/kernel/arch/arm64/boot/dts/rockchip/ 目录。

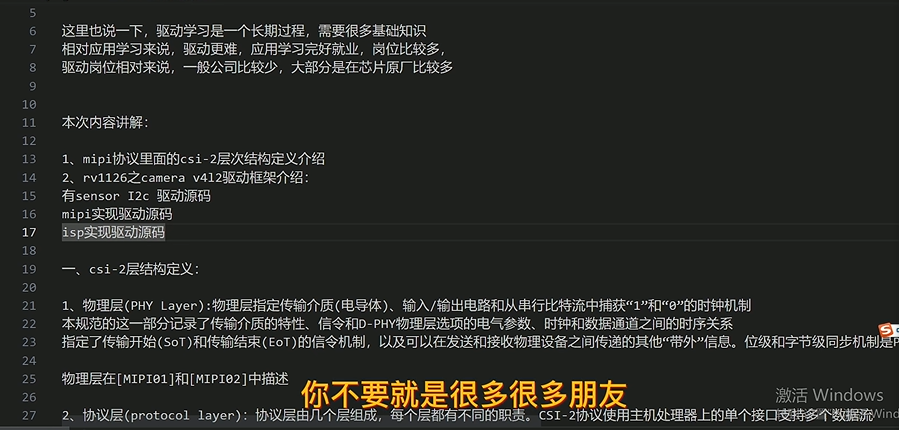
梳理：

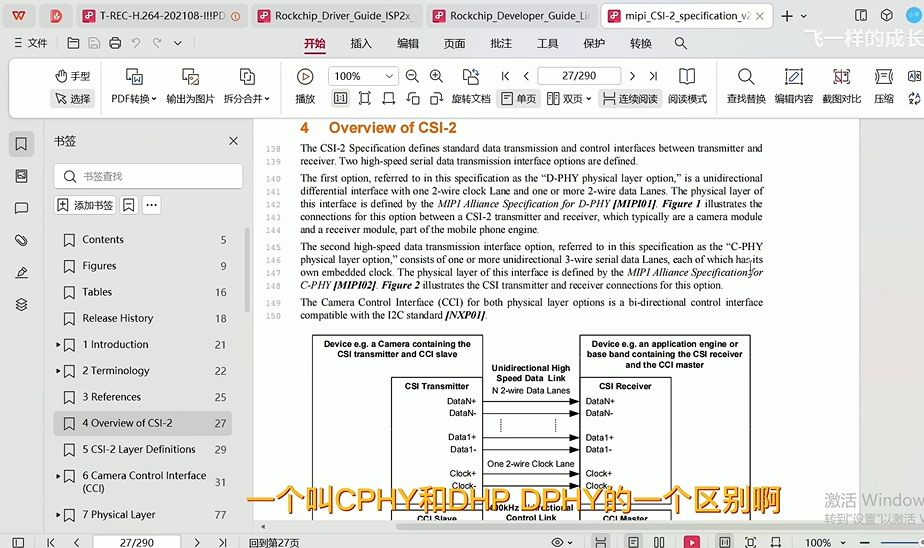
Camera到MIPI物理层的一个输入，然后再物理层输出后到达ISP的流程：



——————————————————————————————————







1、mipi协议里面的csi-2层次结构定义介绍

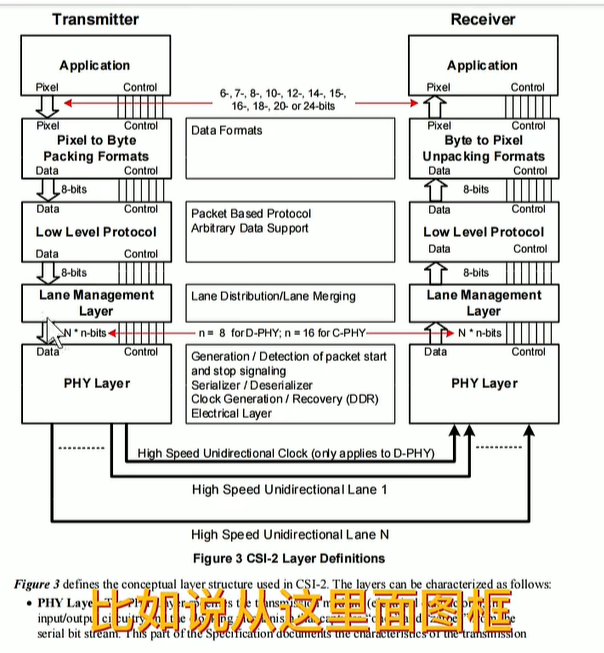
2、rv1126之camera v4l2驱动框架介绍：

有sensor I2c 驱动源码

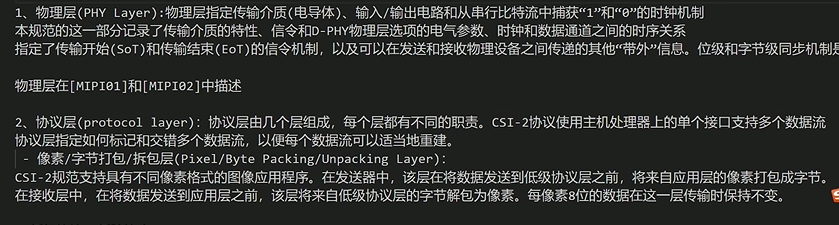
mipi实现驱动源码

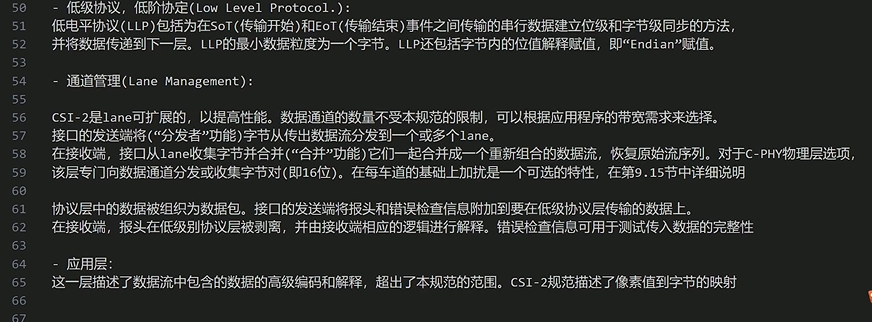
isp实现驱动源码

csi层次结构定义：

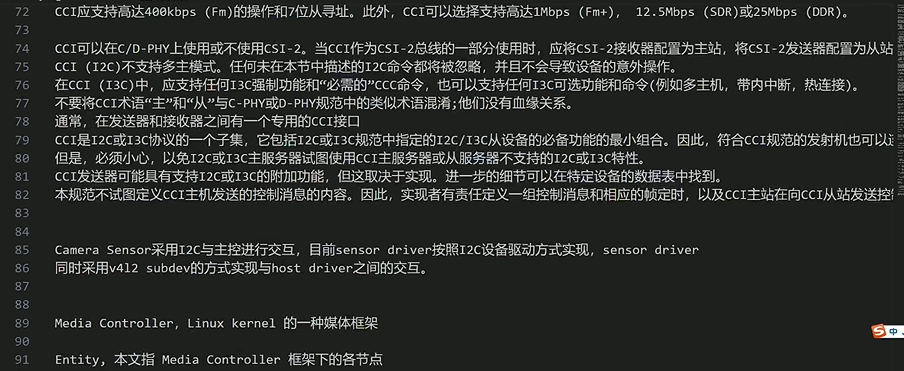


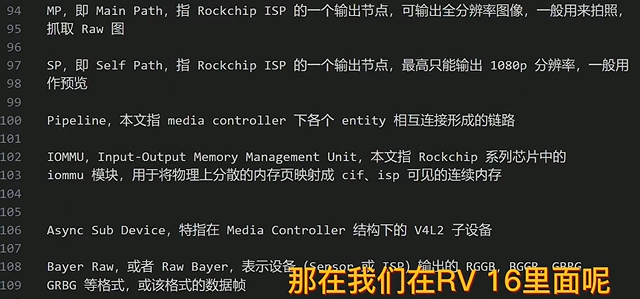


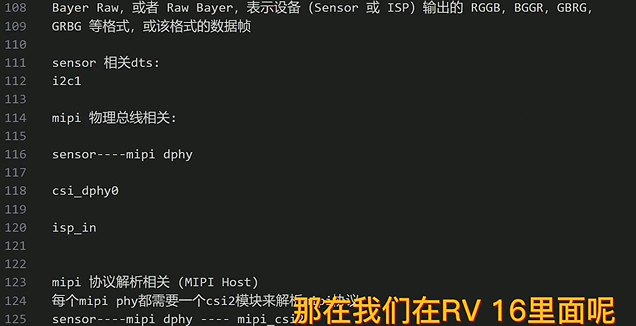




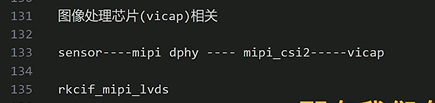






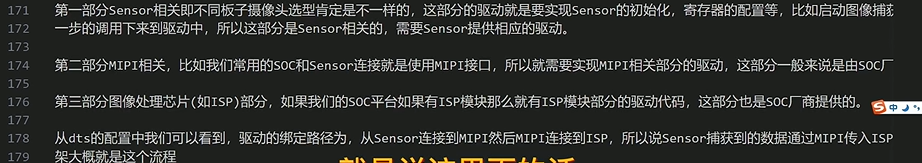


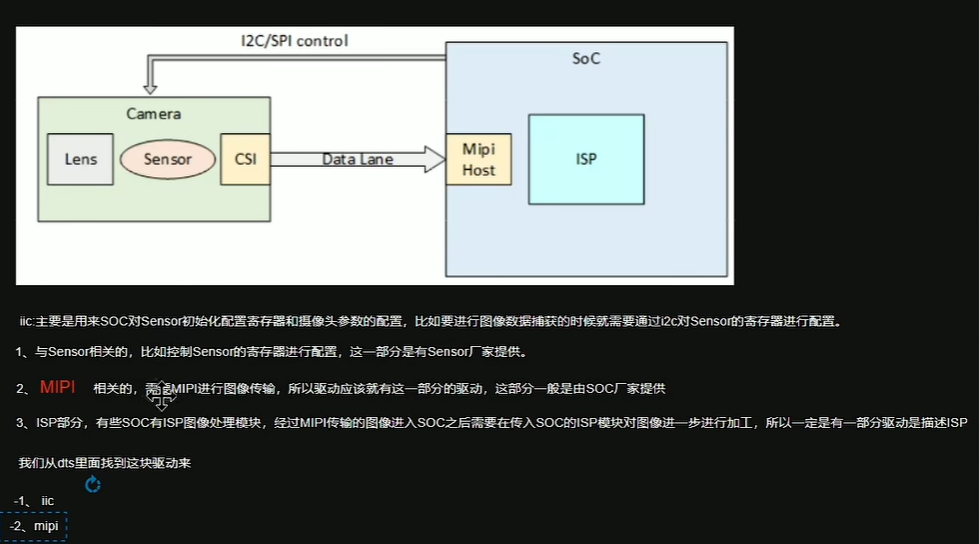












左边是一个sensor，比如：Camera。

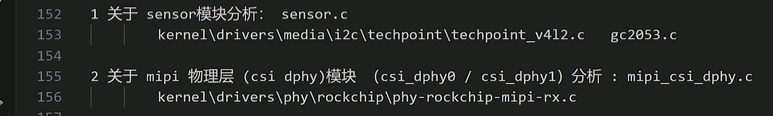
它通过硬件层次的如：MIPI协议的数据传输到SOC，soc再通过I2c或spi做控制。IIC用得更多并且主要是对sensor的寄存器进行配置和摄像头参数的一个配置。如：对图像数据捕获的时候就是要通过这个i2c的sensor计算器进行一个配置。或者刷/读寄存器。就可以读到i2c的设备的地址之类的。

从构造图知:camera是会遵循MIPI协议，通过CSI-2（图中CSI是老版本）的一个窗口（也就是camera series interface的一个窗口） 再过数据通道 发送到 MIPI的一个主机。

SOC端接收的数据是一个原始数据，还需要经过ISP的一个驱动处理。最终才会得到一个比较合理的像素/图像。在往上就是通过H264/H265的编码器算法的一个编码压缩。



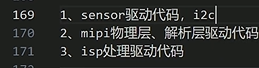
实际参考代码文件为：





结构总结：

重点分析这几个：



IMG_256

IMG_256