



StarFive
赛昉科技

昉·惊鸿 7110 Camera开发和移植手册

昉·星光 2

版本：1.0

日期：2023/08/18

Doc ID：JH7110-DGCH-020

法律声明

阅读本文件前的重要法律告知。

版权注释

版权 ©上海赛昉科技有限公司，2023。版权所有。

本文档中的说明均基于“视为正确”提供，可能包含部分错误。内容可能因产品开发而定期更新或修订。上海赛昉科技有限公司（以下简称“赛昉科技”）保留对本协议中的任何内容进行更改的权利，恕不另行通知。

赛昉科技明确否认任何形式的担保、解释和条件，无论是明示的还是默示的，包括但不限于适销性、特定用途适用性和非侵权的担保或条件。

赛昉科技无需承担因应用或使用任何产品或电路而产生的任何责任，并明确表示无需承担任何及所有连带责任，包括但不限于间接、偶然、特殊、惩戒性或由此造成的损害。

本文件中的所有材料受版权保护，为赛昉科技所有。不得以任何方式修改、编辑或断章取义本文件中的说明，本文件或其任何部分仅限用于内部使用或教育培训。

联系我们：

地址：浦东新区盛夏路61弄张润大厦2号楼502，上海市，201203，中国

网站：<http://www.starfivetech.com>

邮箱：

- sales@starfivetech.com（销售）
- support@starfivetech.com（支持）

目录

表格清单.....	5
插图清单.....	6
法律声明.....	ii
前言.....	vii
1. 简介.....	8
2. Camera及驱动介绍.....	9
2.1. 驱动版本.....	9
2.2. 相关概念.....	9
3. v4l-utils工具及应用.....	10
3.1. 获取并编译v4l-utils.....	10
3.2. FourCC.....	10
3.3. mbus-fmt.....	10
3.4. Media-ctl及拓扑结构.....	11
3.4.1. Media-ctl: 拓扑结构.....	11
3.4.2. VIN拓扑结构.....	11
3.4.3. Media-ctl: entity、pad和link.....	12
3.4.4. Media-ctl: In-/Active Link.....	17
3.4.5. Media-ctl: 修改 fmt/size.....	17
3.5. v4l2-ctl设置controls及抓帧.....	17
3.5.1. 设置fmt并抓帧.....	17
3.5.2. 设置曝光、gain等control.....	18
4. VIN驱动及DTS配置.....	20
4.1. VIN驱动代码简介.....	20
4.2. VIN DTS配置.....	21
4.3. VIN驱动调试及常见问题.....	24
4.3.1. 判断VIN是否probe成功.....	24
4.3.2. 判断Sensor与VIN是否已经绑定.....	24
4.3.3. 打开debug开关.....	24
5. Sensor驱动开发移植.....	26
5.1. 上电时序.....	26
5.2. Sensor初始化寄存器列表.....	27
5.3. V4l2_subdev_ops回调函数.....	27
5.4. V4l2 controller.....	28

5.5. Probe函数及注册media entity、v4l2 subdev.....	28
5.6. DTS示例.....	29
5.7. Sensor调试及常见QA.....	30
5.7.1. Sensor是否注册成功.....	30
5.7.2. Sensor是否有数据输出.....	30
5.7.3. 检查control是否生效.....	30



表格清单

表 0-1 修订历史.....	vii
表 2-1 驱动版本.....	9



插图清单

图 3-1 VIN拓扑结构..... 12



前言

关于本指南和技术支持信息

关于本手册

本手册主要为SDK开发和移植提供赛昉科技新一代SoC平台——昉·惊鸿 7110 camera模块的编程基础和调试操作。

受众

本手册主要服务于与camera相关驱动程序的开发人员。如果您正在开发其他模块，请与您的销售或支持顾问联系，获取我们在昉·惊鸿 7110上的完整文档。






修订历史

表 0-1 修订历史

版本	发布说明	修订
1.0	2023/08/18	首次发布。

注释和注意事项

本指南中可能会出现以下注释和注意事项：

-  **提示：**
建议如何在某个主题或步骤中应用信息。
-  **注：**
解释某个特例或阐释一个重要的点。
-  **重要：**
指出与某个主题或步骤有关的重要信息。
-  **警告：**
表明某个操作或步骤可能会导致数据丢失、安全问题或性能问题。
-  **警告：**
表明某个操作或步骤可能导致物理伤害或硬件损坏。

1. 简介

本文档主要介绍昉·惊鸿 7110芯片的camera模块在昉·星光 2平台上的开发流程。



2. Camera及驱动介绍

本章主要包含以下两个方面：

- [驱动版本 \(第 9页\)](#)
- [相关概念 \(第 9页\)](#)

2.1. 驱动版本

Camera在本文泛指昉·惊鸿 7110芯片中的VIN及其连接的sensor，且本文描述的VIN驱动指的是基于Media Controller、Async sub device和 vb2、以符合V4L2框架标准为目标的新版本驱动。

目前昉·星光 2平台上camera硬件接口、linux内核版本以及camera驱动框架如下表所示。

表 2-1 驱动版本

平台	支持接口	是否具备ISP模块	linux 内核版本	camera驱动框架
昉·星光 2	MIPI CSI	有	5.15	V4L2框架

2.2. 相关概念

以下是本手册中涉及到的相关概念：

- **Async Sub Device**：特指在Media Controller结构下的V4L2子设备；
- **Bayer Raw (Raw Bayer)**：表示设备（Sensor或ISP）输出的RGGB、BGGR、GBRG、GRBG等格式，或该格式的数据帧；
- **Buildroot**：特指赛昉科技基于Buildroot发布的一系列Linux SDK；
- **Entity**：本文指Media Controller框架下的各节点；
- **VIN**：Video In，即视频输入，其中也包含了ISP模块；
- **V4L2**：即Video4Linux2，Linux kernel的视频处理模块；
- **Media Controller**：Linux kernel的一种媒体框架；
- **MIPI**：本文指 MIPI 协议；
- **MIPI-DPHY**：本文指MIPI-DPHY协议，或昉·惊鸿 7110芯片中符合MIPI-DPHY协议的控制

3. v4l-utils工具及应用

v4l-utils工具是由Linuxtv维护的一个V4L2开发套件，它提供一系列V4L2及media framework相关的工具，用来配置V4L2子设备的属性，测试V4L2设备，并提供如libv4l2.so开发库等等。

本章主要介绍v4l-utils中的两个命令行工具：`media-ctl`和`v4l2-ctl`。

- `media-ctl`: 用以查看、配置拓扑结构;
- `v4l2-ctl`: 用以配置v4l2 controls, 可抓帧, 设置vin、isp、sensor参数。



注:

不同版本的v4l-utils所对应的参数, format代码会有些不同, 特别是mbus-fmt部分。

3.1. 获取并编译v4l-utils

在发布的Linux SDK中, 默认已集成了v4l-utils包。用户可以通过buildroot的编译开关开启或关闭v4l-utils包。如SDK目录下, `conf/buildroot_initramfs_config` (配置成initramfs) 或`conf/buildroot_rootfs_config` (配置成rootfs) 文件中。

如下这一行代码可将v4l-utils编译选项打开:

```
BR2_PACKAGE_LIBV4L_UTILS=y
```

用户也可以在[linuxtv的官网](#)获取源码编译, 编译请参考 [wiki](#)。V4l-utils包在ubuntu系统下, 也可通过apt工具直接安装, 命令如下:

```
$ sudo apt-get install v4l-utils
```

3.2. FourCC

FourCC, 全称为Four Character Codes, 它用4个字符 (即32bit) 来命名图像格式。在Linux Kernel中, 它是一个宏, 定义如下:

```
#define v4l2_fourcc(a, b, c, d)\
((__u32)(a) | ((__u32)(b) << 8) | ((__u32)(c) << 16) | ((__u32)(d) << 24))
```



注:

FourCC所定义的格式, 是图像视频在内存中存储的格式。

3.3. mbus-fmt

Mbus-fmt，全称为Media Bus Pixel Codes，它描述的是用于在物理总线上传输的格式，比如sensor通过MIPI DPHY向ISP传输的图像格式，或者在ISP内部各子模块间传输的格式。特别需要将Mbus-fmt与上一小节的FourCC区分，后者是专指存储在Memory中的图像格式。

3.4. Media-ctl及拓扑结构

Media-ctl是v4l-utils包中的一个工具，主要用来配置media framework的各节点fmt、size和link。您可通过以下help命令来查询使用方法。

```
[root@px3se:/]# media-ctl --help
media-ctl [options]
-d, --device dev Media device name (default: /dev/media0)
-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
  --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
.....
```

接下来几个小节将介绍几种常用的例子。

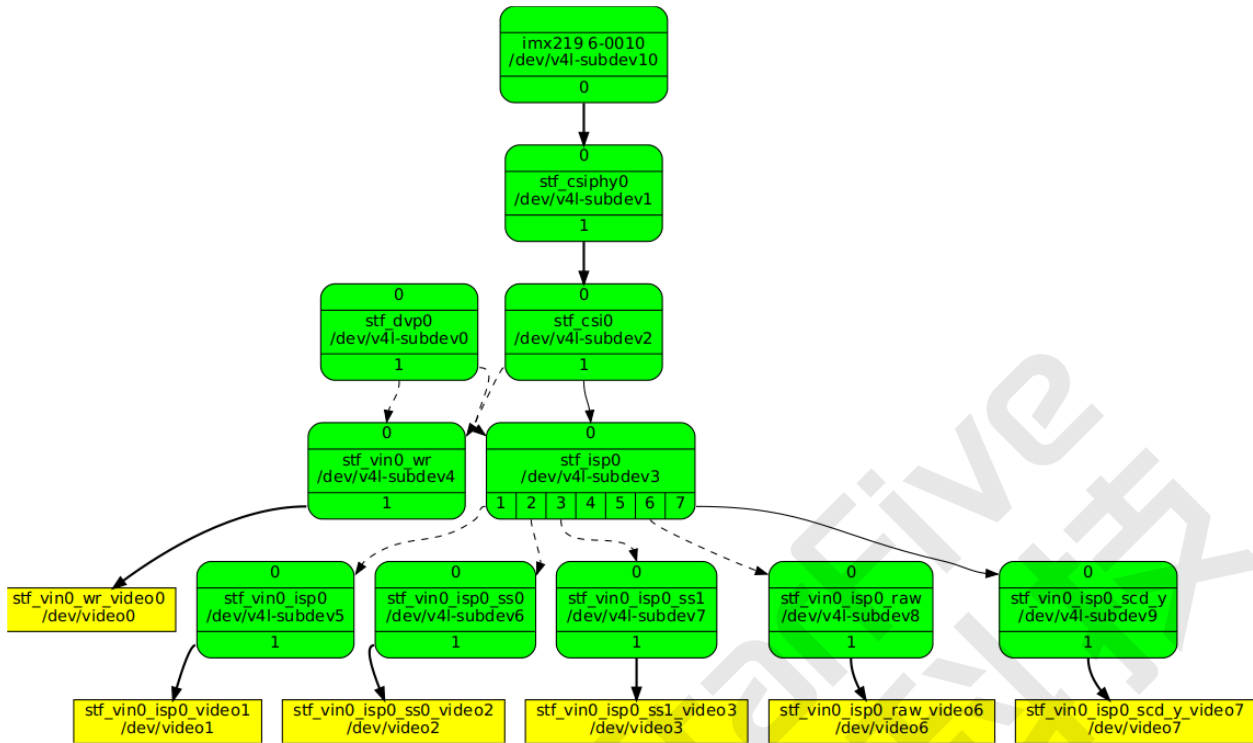
3.4.1. Media-ctl: 拓扑结构

驱动如果有支持Media Controller，在VIN加载成功后会创建media设备，如/dev/media0。利用media-ctl可以打印出当前的pipeline的连接情况。

3.4.2. VIN拓扑结构

下图是赛昉科技昉·星光 2上VIN的拓扑结构。IMX219作为sub-device Source端链接到VIN中csiiphy0 sink端。

图 3-1 VIN拓扑结构



3.4.3. Media-ctl: entity、pad和link

Entity在Media Controller中，表示一个节点。它包含一个或多个的输入输出pads。Link表示一条链接，它连接多个不同的pad。多个link组成了一条完整的pipeline。

Entity的名称可以从拓扑结构中查看，pad由数字表示，一个entity中可以包含多个pad，既可以有Source，也可以有Sink。Link连接两个“entity”：pad。以下代码块为昉·惊鸿 7110 Media device中entity、pad和link的详细信息。

```
# media-ctl -p -d /dev/media0
Media controller API version 5.15.0

Media device information
-----
driver          jh7110-vin
model           Starfive Camera Subsystem
serial          0123456789ABCDEF
bus info        platform:19800000.vin_sysctl
hw revision     0x1
driver version  5.15.0

Device topology
- entity 1: stf_dvp0 (2 pads, 2 links)
  type V4L2 subdev subtype Unknown flags 0
  device node name /dev/v4l-subdev0
  pad0: Sink
  [fmt:RGB565_2X8_LE/1920x1080 field:none colorspace:srgb]
```

```

pad1: Source
    [fmt:RGB565_2X8_LE/1920x1080 field:none colorspace:srgb]
    -> "stf_vin0_wr":0 []
    -> "stf_isp0":0 []

- entity 4: stf_csiphy0 (2 pads, 2 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev1
    pad0: Sink
        [fmt:SRGB10_1X10/1920x1080 field:none colorspace:srgb
xfer:srgb ycbcr:601 quantization:full-range]
        <- "imx219 6-0010":0 [ENABLED,IMMUTABLE]
    pad1: Source
        [fmt:SRGB10_1X10/1920x1080 field:none colorspace:srgb
xfer:srgb ycbcr:601 quantization:full-range]
        -> "stf_csi0":0 [ENABLED,IMMUTABLE]

- entity 7: stf_csi0 (2 pads, 3 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev2
    pad0: Sink
        [fmt:SRGB10_1X10/1920x1080 field:none colorspace:srgb
xfer:srgb ycbcr:601 quantization:full-range]
        <- "stf_csiphy0":1 [ENABLED,IMMUTABLE]
    pad1: Source
        [fmt:SRGB10_1X10/1920x1080 field:none colorspace:srgb
xfer:srgb ycbcr:601 quantization:full-range]
        -> "stf_vin0_wr":0 []
        -> "stf_isp0":0 [ENABLED]

- entity 10: stf_isp0 (8 pads, 7 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev3
    pad0: Sink
        [fmt:SRGB10_1X10/1920x1080 field:none colorspace:srgb
xfer:srgb ycbcr:601 quantization:full-range
crop.bounds:(0,0)/1920x1080
crop:(0,0)/1920x1080
compose.bounds:(0,0)/1920x1080
compose:(0,0)/1920x1080]
        <- "stf_dvp0":1 []
        <- "stf_csi0":1 [ENABLED]
    pad1: Source
        [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb xfer:srgb
ycbcr:601 quantization:full-range
crop.bounds:(0,0)/1920x1080
crop:(0,0)/1920x1080
compose.bounds:(0,0)/1920x1080
compose:(0,0)/1920x1080]
        -> "stf_vin0_isp0":0 []

```

```

    pad2: Source
        [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb xfer:srgb
ycbcr:601 quantization:full-range
        crop.bounds:(0,0)/1920x1080
        crop:(0,0)/1920x1080
        compose.bounds:(0,0)/1920x1080
        compose:(0,0)/1920x1080]
    -> "stf_vin0_isp0_ss0":0 []

    pad3: Source
        [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb xfer:srgb
ycbcr:601 quantization:full-range
        crop.bounds:(0,0)/1920x1080
        crop:(0,0)/1920x1080
        compose.bounds:(0,0)/1920x1080
        compose:(0,0)/1920x1080]
    -> "stf_vin0_isp0_ss1":0 []

    pad4: Source
        [fmt:SRGB10_1X10/1920x1080 field:none colorspace:srgb
xfer:srgb ybcr:601 quantization:full-range
        crop.bounds:(0,0)/1920x1080
        crop:(0,0)/1920x1080
        compose.bounds:(0,0)/1920x1080
        compose:(0,0)/1920x1080]

    pad5: Source
        [fmt:SRGB10_1X10/1920x1080 field:none colorspace:srgb
xfer:srgb ybcr:601 quantization:full-range
        crop.bounds:(0,0)/1920x1080
        crop:(0,0)/1920x1080
        compose.bounds:(0,0)/1920x1080
        compose:(0,0)/1920x1080]

    pad6: Source
        [fmt:SRGB12_1X12/1920x1080 field:none colorspace:srgb
xfer:srgb ybcr:601 quantization:full-range
        crop.bounds:(0,0)/1920x1080
        crop:(0,0)/1920x1080]
    -> "stf_vin0_isp0_raw":0 []

    pad7: Source
        [fmt:SRGB12_1X12/1920x1080 field:none colorspace:srgb
xfer:srgb ybcr:601 quantization:full-range
        crop.bounds:(0,0)/1920x1080
        crop:(0,0)/1920x1080]
    -> "stf_vin0_isp0_scd_y":0 [ENABLED]

- entity 19: stf_vin0_wr (2 pads, 3 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev4
    pad0: Sink
        [fmt:RGB565_2X8_LE/1920x1080 field:none colorspace:srgb]
    <- "stf_dvp0":1 []
    <- "stf_csi0":1 []

```

```

pad1: Source
    [fmt:RGB565_2X8_LE/1920x1080 field:none colorspace:srgb]
    -> "stf_vin0_wr_video0":0 [ENABLED,IMMUTABLE]

- entity 22: stf_vin0_wr_video0 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video0
pad0: Sink
    <- "stf_vin0_wr":1 [ENABLED,IMMUTABLE]

- entity 28: stf_vin0_isp0 (2 pads, 2 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev5
pad0: Sink
    [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb]
    <- "stf_isp0":1 []
pad1: Source
    [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb]
    -> "stf_vin0_isp0_video1":0 [ENABLED,IMMUTABLE]

- entity 31: stf_vin0_isp0_video1 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video1
pad0: Sink
    <- "stf_vin0_isp0":1 [ENABLED,IMMUTABLE]

- entity 37: stf_vin0_isp0_ss0 (2 pads, 2 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev6
pad0: Sink
    [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb]
    <- "stf_isp0":2 []
pad1: Source
    [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb]
    -> "stf_vin0_isp0_ss0_video2":0 [ENABLED,IMMUTABLE]

- entity 40: stf_vin0_isp0_ss0_video2 (1 pad, 1 link)
    type Node subtype V4L flags 0
    device node name /dev/video2
pad0: Sink
    <- "stf_vin0_isp0_ss0":1 [ENABLED,IMMUTABLE]

- entity 46: stf_vin0_isp0_ss1 (2 pads, 2 links)
    type V4L2 subdev subtype Unknown flags 0
    device node name /dev/v4l-subdev7
pad0: Sink
    [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb]
    <- "stf_isp0":3 []
pad1: Source
    [fmt:Y12_1X12/1920x1080 field:none colorspace:srgb]

```

```
    -> "stf_vin0_isp0_ssl_video3":0 [ENABLED,IMMUTABLE]

- entity 49: stf_vin0_isp0_ssl_video3 (1 pad, 1 link)
  type Node subtype V4L flags 0
  device node name /dev/video3
  pad0: Sink
    <- "stf_vin0_isp0_ssl":1 [ENABLED,IMMUTABLE]

- entity 55: stf_vin0_isp0_raw (2 pads, 2 links)
  type V4L2 subdev subtype Unknown flags 0
  device node name /dev/v4l-subdev8
  pad0: Sink
    [fmt:SBGGR12_1X12/1920x1080 field:none colorspace:srgb]
    <- "stf_isp0":6 []
  pad1: Source
    [fmt:SBGGR12_1X12/1920x1080 field:none colorspace:srgb]
    -> "stf_vin0_isp0_raw_video6":0 [ENABLED,IMMUTABLE]

- entity 58: stf_vin0_isp0_raw_video6 (1 pad, 1 link)
  type Node subtype V4L flags 0
  device node name /dev/video6
  pad0: Sink
    <- "stf_vin0_isp0_raw":1 [ENABLED,IMMUTABLE]

- entity 64: stf_vin0_isp0_scd_y (2 pads, 2 links)
  type V4L2 subdev subtype Unknown flags 0
  device node name /dev/v4l-subdev9
  pad0: Sink
    [fmt:SRGGB12_1X12/1920x1080 field:none colorspace:srgb
  xfer:srgb ycbcr:601 quantization:full-range]
    <- "stf_isp0":7 [ENABLED]
  pad1: Source
    [fmt:SRGGB12_1X12/1920x1080 field:none colorspace:srgb
  xfer:srgb ycbcr:601 quantization:full-range]
    -> "stf_vin0_isp0_scd_y_video7":0 [ENABLED,IMMUTABLE]

- entity 67: stf_vin0_isp0_scd_y_video7 (1 pad, 1 link)
  type Node subtype V4L flags 0
  device node name /dev/video7
  pad0: Sink
    <- "stf_vin0_isp0_scd_y":1 [ENABLED,IMMUTABLE]

- entity 93: imx219 6-0010 (1 pad, 1 link)
  type V4L2 subdev subtype Sensor flags 0
  device node name /dev/v4l-subdev10
  pad0: Source
    [fmt:SRGGB10_1X10/1920x1080@1/30 field:none colorspace:srgb
  xfer:srgb ycbcr:601 quantization:full-range
  crop.bounds:(8,8)/3280x2464
```



```
crop:(688,700)/1920x1080]
-> "stf_csiphy0":0 [ENABLED,IMMUTABLE]
```

3.4.4. Media-ctl: In-/Active Link

以下示例命令为将/dev/media0设置为Active和Inactive。

```
Active link:
media-ctl -d /dev/media0 -v1 "'stf_csi0':1 -> 'stf_isp0':0 [1]"
In-active link:
media-ctl -d /dev/media0 -v1 "'stf_csi0':1 -> 'stf_isp0':0 [0]"
```



注：

以上命令的格式为：`media-ctl -d "entity name": pad->"entity name": pad[Status]'`。

3.4.5. Media-ctl: 修改 fmt/size

以下命令为设置sensor输出1920x1080：

```
media-ctl -d /dev/media0 \
--set-v4l2 "'imx219 6-0010":0[fmt:SRGGB10_1X10/1920x1080]'
```

3.5. v4l2-ctl设置controls及抓帧

Media-ctl工具的操作是通过/dev/media0等media设备，它所管理是Media的拓扑结构中各个节点的format、大小、链接。V4l2-ctl工具则是针对/dev/video0、/dev/video1等video设备，它在video设备上进行set_fmt、reqbuf、qbuf、dobuf、stream_on、stream_off等一系列操作。

本文中主要用v4l2-ctl进行采集帧数据，设置曝光、gain等v4l2_control。

3.5.1. 设置fmt并抓帧

以下代码块为设置通过video1抓取nv12数据保存到test.yuv，分辨率为1920 × 1080。

```
media-ctl -d /dev/media0 -v1 "'stf_csi0':1 -> 'stf_isp0':0 [1]"
media-ctl -d /dev/media0 -v1 "'stf_isp0':1 -> 'stf_vin0_isp0':0 [1]"
media-ctl -d /dev/media0 \
--set-v4l2 "'imx219 6-0010":0[fmt:SRGGB10_1X10/1920x1080]'
```

```
v4l2-ctl -d /dev/video1
--set-fmt-video=width=1920,height=1080,pixelformat=NV12
--stream-mmap=4
```

```
--stream-to=test.yuv
--stream-skip=5
--stream-count=5
```



注：

以下为上述代码块的参数说明：

- **-d**：指定操作对象为/dev/media0设备。
- **--set-fmt-video**：指定了宽高及 pixelformat。
- **NV12**：即用FourCC表示的pixelformat。FourCC编码详见下文FourCC。
- **--stream-mmap**：指定buffer的类型为mmap，即由kernel分配的物理连续的或经过iommu映射的buffer。
- **--stream-to**：指定帧数据保存的文件路径。
- **--stream-skip**：指定丢弃（不保存到文件）前5帧。
- **--stream-count**：指定抓取的帧数，不包括--stream-skip丢弃的数量。

3.5.2. 设置曝光、gain等control

通过命令列出IMX219可控制的control（注意下方IMX219对应的节点是/dev/v4l-subdev10）。

```
# v4l2-ctl -d /dev/v4l-subdev10 -l

User Controls
                exposure 0x00980911 (int)      : min=4 max=1759
step=1 default=1600 value=1600
                horizontal_flip 0x00980914 (bool) : default=0 value=0
flags=modify-layout
                vertical_flip 0x00980915 (bool)  : default=0 value=0
flags=modify-layout

Camera Controls
                camera_orientation 0x009a0922 (menu) : min=0 max=2 default=1
value=1 (Back) flags=read-only
                camera_sensor_rotation 0x009a0923 (int) : min=0 max=0
step=1 default=0 value=0 flags=read-only

Image Source Controls
                vertical_blanking 0x009e0901 (int)  : min=4 max=64455
step=1 default=683 value=683
```

```

horizontal_blanking 0x009e0902 (int) : min=1528 max=1528
step=1 default=1528 value=1528 flags=read-only
analogue_gain 0x009e0903 (int) : min=0 max=232
step=1 default=208 value=208
red_pixel_value 0x009e0904 (int) : min=0 max=1023
step=1 default=1023 value=1023
green_red_pixel_value 0x009e0905 (int) : min=0 max=1023
step=1 default=1023 value=1023
blue_pixel_value 0x009e0906 (int) : min=0 max=1023
step=1 default=1023 value=1023
green_blue_pixel_value 0x009e0907 (int) : min=0 max=1023
step=1 default=1023 value=1023

```

Image Processing Controls

```

link_frequency 0x009f0901 (intmenu): min=0 max=0 default=0
value=0 (456000000 0x1b2e0200) flags=read-only
pixel_rate 0x009f0902 (int64) : min=182400000
max=182400000 step=1 default=182400000 value=182400000 flags=read-only
test_pattern 0x009f0903 (menu) : min=0 max=4 default=0
value=0 (Disabled)
digital_gain 0x009f0905 (int) : min=256 max=4095
step=1 default=256 value=256

```

用v4l2-ctl可以修改这些control，如修改exposure及analogue_gain，如下：

```
v4l2-ctl -d /dev/v4l-subdev10 --set-ctrl 'exposure=1216,analogue_gain=10'
```

4. VIN驱动及DTS配置

本章主要介绍VIN驱动及DTS的配置，并详细描述VIN驱动在各芯片上的测试验证情况。

4.1. VIN驱动代码简介

VIN内核驱动代码位于drivers/media/platform/starfive/v4l2_driver/目录，通过编译开关CONFIG_VIDEO_STF_VIN控制是否编译。

VIN驱动根据media controller、v4l2、vb2框架，完成硬件配置，帧中断处理，buffer轮转等功能。各文件的功能内容如下：

```
$ tree drivers/media/platform/starfive/v4l2_driver/
drivers/media/platform/starfive/v4l2_driver/
├── imx219_mipi.c
├── ov13850_mipi.c
├── ov4689_mipi.c
├── ov5640.c
├── Readme.txt
├── sc2235.c
├── stfcamss.c
├── stfcamss.h
├── stf_common.h
├── stf_csi.c
├── stf_csi.h
├── stf_csi_hw_ops.c
├── stf_csiphy.c
├── stf_csiphy.h
├── stf_csiphy_hw_ops.c
├── stf_dmabuf.c
├── stf_dmabuf.h
├── stf_dvp.c
├── stf_dvp.h
├── stf_dvp_hw_ops.c
├── stf_event.c
├── stf_isp.c
├── stf_isp.h
├── stf_isp_hw_ops.c
├── stf_isp_ioctl.h
├── stf_video.c
├── stf_video.h
├── stf_vin.c
├── stf_vin.h
└── stf_vin_hw_ops.c
```

**注:**

赛昉科技平台默认也会将sensor驱动移植到该目录，但是推荐的sensor存放目录为drivers/media/i2c。

4.2. VIN DTS配置

昉·惊鸿 7110 SoC平台的一般配置文件在以下路径:

```
linux/arch/riscv/boot/dts/starfive/jh7110.dtsi
```

以下代码块显示了VIN DTS配置文件的内容。

```
vin_sysctl: vin_sysctl@19800000 {
    compatible = "starfive,jh7110-vin";
    reg = <0x0 0x19800000 0x0 0x10000>,
        <0x0 0x19810000 0x0 0x10000>,
        <0x0 0x19820000 0x0 0x10000>,
        <0x0 0x19840000 0x0 0x10000>,
        <0x0 0x19870000 0x0 0x30000>,
        <0x0 0x11840000 0x0 0x10000>,
        <0x0 0x17030000 0x0 0x10000>,
        <0x0 0x13020000 0x0 0x10000>;
    reg-names = "csi2rx", "vclk", "vrst", "sctrl",
        "isp", "trst", "pmu", "syscrgr";
    clocks = <&clkisp JH7110_DOM4_APB_FUNC>,
        <&clkisp JH7110_U0_VIN_PCLK>,
        <&clkisp JH7110_U0_VIN_SYS_CLK>,
        <&clkisp
JH7110_U0_ISPV2_TOP_WRAPPER_CLK_C>,
        <&clkisp JH7110_DVP_INV>,
        <&clkisp JH7110_U0_VIN_CLK_P_AXIWR>,
        <&clkisp JH7110_MIPI_RX0_PXL>,
        <&clkisp JH7110_U0_VIN_PIXEL_CLK_IF0>,
        <&clkisp JH7110_U0_VIN_PIXEL_CLK_IF1>,
        <&clkisp JH7110_U0_VIN_PIXEL_CLK_IF2>,
        <&clkisp JH7110_U0_VIN_PIXEL_CLK_IF3>,
        <&clkisp JH7110_U0_M31DPHY_CFGCLK_IN>,
        <&clkisp JH7110_U0_M31DPHY_REFCLK_IN>,
        <&clkisp JH7110_U0_M31DPHY_TXCLKESC_LAN0>,
        <&clkgen JH7110_ISP_TOP_CLK_ISPCORE_2X>,
        <&clkgen JH7110_ISP_TOP_CLK_ISP_AXI>;
    clock-names
= "clk_apb_func", "clk_pclk", "clk_sys_clk",
    "clk_wrapper_clk_c", "clk_dvp_inv", "clk_axiwr",
    "clk_mipi_rx0_pxl", "clk_pixel_clk_if0",
```

```

        "clk_pixel_clk_if1", "clk_pixel_clk_if2",

"clk_pixel_clk_if3", "clk_m31dphy_cfgclk_in",

"clk_m31dphy_refclk_in", "clk_m31dphy_txclkesc_lan0",
        "clk_ispcore_2x", "clk_isp_axi";
    resets = <&rstgen RSTN_U0_ISPV2_TOP_WRAPPER_P>,
            <&rstgen RSTN_U0_ISPV2_TOP_WRAPPER_C>,
            <&rstgen RSTN_U0_VIN_N_PCLK>,
            <&rstgen RSTN_U0_VIN_N_SYS_CLK>,
            <&rstgen RSTN_U0_VIN_P_AXIRD>,
            <&rstgen RSTN_U0_VIN_P_AXIWR>,
            <&rstgen RSTN_U0_VIN_N_PIXEL_CLK_IF0>,
            <&rstgen RSTN_U0_VIN_N_PIXEL_CLK_IF1>,
            <&rstgen RSTN_U0_VIN_N_PIXEL_CLK_IF2>,
            <&rstgen RSTN_U0_VIN_N_PIXEL_CLK_IF3>,
            <&rstgen RSTN_U0_M31DPHY_HW>,
            <&rstgen RSTN_U0_M31DPHY_B09_ALWAYS_ON>,
            <&rstgen RSTN_U0_DOM_ISP_TOP_N>,
            <&rstgen RSTN_U0_DOM_ISP_TOP_AXI>;

    reset-names
= "rst_wrapper_p", "rst_wrapper_c", "rst_pclk",

"rst_sys_clk", "rst_axird", "rst_axiwr", "rst_pixel_clk_if0",

"rst_pixel_clk_if1", "rst_pixel_clk_if2", "rst_pixel_clk_if3",

"rst_m31dphy_hw", "rst_m31dphy_b09_always_on",
        "rst_isp_top_n", "rst_isp_top_axi";
    starfive,aon-syscon = <&aon_syscon 0x00>;
    power-domains = <&pwrc JH7110_PD_ISP>;
    /* irq nr: vin, isp, isp_csi, isp_scd, isp_csiline
*/
    interrupts = <92 87 88 89 90>;
    status = "disabled";
};

&vin_sysctl {
    status = "okay";

    ports {

        #address-cells = <1>;
        #size-cells = <0>;

        port@0 {
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;

            /* Parallel bus endpoint */

```

```

parallel_from_sc2235: endpoint@0 {
    reg = <0>;
    remote-endpoint = <&sc2235_to_parallel>;
    bus-type = <5>;          /* Parallel */
    bus-width = <8>;
    data-shift = <2>;       /* lines 9:2 are
used */

    hsync-active = <1>;
    vsync-active = <1>;
    pclk-sample = <1>;
    status = "okay";
};

};

port@1 {
    reg = <1>;
    #address-cells = <1>;
    #size-cells = <0>;

    /* CSI2 bus endpoint */
    csi2rx0_from_ov4689: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&ov4689_to_csi2rx0>;
        bus-type = <4>;      /* MIPI CSI-2 D-PHY */
        clock-lanes = <0>;
        data-lanes = <1 2 3 4>;
        status = "okay";
    };

    /* CSI2 bus endpoint */
    csi2rx0_from_imx219: endpoint@1 {
        reg = <1>;
        remote-endpoint = <&imx219_to_csi2rx0>;
        bus-type = <4>;      /* MIPI CSI-2 D-PHY */
        clock-lanes = <0>;
        data-lanes = <2 1>;
        lane-polarities = <1 1 1>;
        status = "okay";
    };

    csi2rx0_from_imx708: endpoint@2 {
        reg = <2>;
        remote-endpoint = <&imx708_to_csi2rx0>;
        bus-type = <4>;      /* MIPI CSI-2 D-PHY */
        clock-lanes = <0>;
        data-lanes = <2 1>;
        lane-polarities = <1 1 1>;
        status = "okay";
    };
};
};

```

```
};
};
```

以下为上述代码块的参数说明：

- **compatible**：兼容性信息，用于连接驱动程序和目标设备。
- **reg**：寄存器基本地址“0x19800000”和范围“0x10000”。
- **reg-names**：VIN模块使用到的寄存器名称。
- **clocks**：所需要的clocks。
- **clock-names**：上述时钟的名称，需要和驱动中定义的相同。
- **resets**：VIN模块使用到的复位信号。
- **reset-names**：上述复位信号的名称。
- **power-domains**：电源域。
- **interrupts**：硬件中断ID。
- **status**：VIN模块的工作状态。要启用模块，请将此位设置为“okay”；要禁用该模块，请将此位设置为“disabled”。

4.3. VIN驱动调试及常见问题

本小节介绍如何判断VIN设备的状态，如何打开debug开关，以及利用v4l2-ctl抓帧。

4.3.1. 判断VIN是否probe成功

输入以下命令，可根据查看/dev目录下是否有media设备，判断VIN是否probe成功：

```
# cat /sys/bus/media/devices/media0/model
Starfive Camera Subsystem
```



注：

media0不是固定的，0也可能是其他数字，如media1。

4.3.2. 判断Sensor与VIN是否已经绑定

输入以下命令可查看sensor和VIN是否连接：

```
media-ctl -p
```

4.3.3. 打开debug开关

输入以下命令可打开debug开关:

```
mount -t debugfs none /sys/kernel/debug/  
echo 255 > /sys/kernel/debug/stfcamss/dbg_mask  
echo 4 > /sys/kernel/debug/stfcamss/dbg_level
```

其中, `dbg_mask`的定义如下:

```
enum {  
    ST_DVP = 0x0001,  
    ST_CSIPHY = 0x0002,  
    ST_CSI = 0x0004,  
    ST_ISP = 0x0008,  
    ST_VIN = 0x0010,  
    ST_VIDEO = 0x0020,  
    ST_CAMSS = 0x0040,  
    ST_SENSOR = 0x0080,  
};
```

`dbg_level`的定义如下:

```
enum {  
    ST_NONE = 0x00,  
    ST_ERR = 0x01,  
    ST_WARN = 0x02,  
    ST_INFO = 0x03,  
    ST_DEBUG = 0x04,  
};
```

5. Sensor驱动开发移植

Sensor驱动与VIN驱动完全独立，二者异步注册，最后通过remote-endpoint建立连接。因此本章所描述的sensor驱动同时适用于VIN。

在Media Controller结构下，sensor一般作为sub-device并通过pad与VIN中的MIPI PHY链接在一起。本章主要介绍sensor驱动的代码，DTS配置及如何验证sensor驱动的正确性。

主要包括以下7个部分：

- [上电时序 \(第 26页\)](#)
- [Sensor初始化寄存器列表 \(第 27页\)](#)
- [V4l2_subdev_ops回调函数 \(第 27页\)](#)
- [V4l2 controller \(第 28页\)](#)
- [Probe函数及注册media entity、v4l2 subdev \(第 28页\)](#)
- [DTS示例 \(第 29页\)](#)
- [Sensor调试及常见QA \(第 30页\)](#)

5.1. 上电时序

不同sensor对上电时序要求不同，例如Sony Camera。在sensor厂家提供的数据手册中，一般会有上电时序图，只需要按顺序配置即可。以IMX219为例，其中imx219_power_on()即是用来给sensor上电的，如下。

```
static int imx219_power_on(struct device *dev)
{
    struct v4l2_subdev *sd = dev_get_drvdata(dev);
    struct imx219 *imx219 = to_imx219(sd);
    int ret;

    ret = regulator_bulk_enable(IMX219_NUM_SUPPLIES, imx219->supplies);
    if (ret) {
        dev_err(dev, "%s: failed to enable regulators\n",
                __func__);
        return ret;
    }

    ret = clk_prepare_enable(imx219->xclk);
    if (ret) {
        dev_err(dev, "%s: failed to enable clock\n", __func__);
        goto reg_off;
    }
}
```

```

    gpiod_set_value_cansleep(imx219->reset_gpio, 1);
    usleep_range(IMX219_XCLR_MIN_DELAY_US,
                IMX219_XCLR_MIN_DELAY_US +
                IMX219_XCLR_DELAY_RANGE_US);

    return 0;

reg_off:
    regulator_bulk_disable(IMX219_NUM_SUPPLIES, imx219->supplies);

    return ret;
}

```

IMX219的上电时序简要说明如下：

- 各路的vdd上电。
- 首先提供xclk。
- 紧接着reset pin使能。

5.2. Sensor初始化寄存器列表

在IMX219中，各定义了`struct imx219_mode supported_modes[]`用来表示sensor不同的初始化mode。Mode可以包括如分辨率、mbus code、寄存器初始化列表等。寄存器初始化列表，请按厂家提供的直接填入即可。

5.3. V4l2_subdev_ops回调函数

V4l2_subdev_ops回调函数是sensor驱动中逻辑控制的核心。回调函数包括非常多的功能，具体可以查看kernel代码`include/media/v4l2-subdev.h`。建议sensor驱动至少包括如下回调函数。以IMX219为例：

```

static const struct v4l2_subdev_core_ops imx219_core_ops = {
    .subscribe_event = v4l2_ctrl_subdev_subscribe_event,
    .unsubscribe_event = v4l2_event_subdev_unsubscribe,
};

static const struct v4l2_subdev_video_ops imx219_video_ops = {
    .s_stream = imx219_set_stream,
};

static const struct v4l2_subdev_pad_ops imx219_pad_ops = {
    .enum_mbus_code = imx219_enum_mbus_code,
    .get_fmt = imx219_get_pad_format,
    .set_fmt = imx219_set_pad_format,
    .get_selection = imx219_get_selection,
};

```

```

        .enum_frame_size = imx219_enum_frame_size,
    };

    static const struct v4l2_subdev_ops imx219_subdev_ops = {
        .core = &imx219_core_ops,
        .video = &imx219_video_ops,
        .pad = &imx219_pad_ops,
    };

```

以下为上述代码块的参数说明:

- **s_stream**: 即set stream, 包括stream on和stream off, 一般在这里配置寄存器, 使其输出图像。
- **enum_mbus_code**: 枚举驱动支持的mbus_code。
- **get_fmt**: 返回当前sensor选中的format/size。如果**get_fmt**缺失, media-ctl工具无法查看sensor entity当前配置的fmt。
- **set_fmt**: 设置sensor的format/size。
- **enum_frame_size**: 枚举驱动支持的分辨率。

5.4. V4l2 controller

对于需要配置fps、exposure、gain、blanking的场景, v4l2 controller部分是必要的。

5.5. Probe函数及注册media entity、v4l2 subdev

Probe函数中, 首先对DTS进行解析, 获取clk、regulator、gpio等信息用以对sensor上下电。其次注册v4l2 controller、media entity、v4l2 subdev。注意到v4l2 subdev的注册是异步。以下代码块为函数调用信息。

```

ret = imx219_init_controls(imx219);
    if (ret)
        goto error_power_off;

    /* Initialize subdev */
    imx219->sd.internal_ops = &imx219_internal_ops;
    imx219->sd.flags |= V4L2_SUBDEV_FL_HAS_DEVNODE |
V4L2_SUBDEV_FL_HAS_EVENTS;
    imx219->sd.entity.function = MEDIA_ENT_F_CAM_SENSOR;

    /* Initialize source pad */
    imx219->pad.flags = MEDIA_PAD_FL_SOURCE;

    /* Initialize default format */
    imx219_set_default_format(imx219);

```

```

ret = media_entity_pads_init(&imx219->sd.entity, 1, &imx219->pad);
if (ret) {
    dev_err(dev, "failed to init entity pads: %d\n", ret);
    goto error_handler_free;
}

ret = v4l2_async_register_subdev_sensor(&imx219->sd);
if (ret < 0) {
    dev_err(dev, "failed to register sensor sub-device: %d\n",
ret);
    goto error_media_entity;
}

```

5.6. DTS示例

Sensor的DTS配置大同小异，根据硬件的设计，主要是pinctl (iomux)、clk、gpio以及remote port。

以下示例是在昉·星光 2上的IMX219 DTS节点。

```

imx219: imx219@10 {
    compatible = "sony,imx219";
    reg = <0x10>;
    clocks = <&clk_ext_camera>;
    clock-names = "xclk";
    reset-gpio = <&gpio 18 0>;
    rotation = <0>;
    orientation = <1>; //CAMERA_ORIENTATION_BACK

    port {
        /* CSI2 bus endpoint */
        imx219_to_csi2rx0: endpoint {
            remote-endpoint = <&csi2rx0_from_imx219>;
            bus-type = <4>; /* MIPI CSI-2 D-PHY */
            clock-lanes = <0>;
            data-lanes = <2 1>;
            lane-polarities = <1 1 1>;
            link-frequencies = /bits/ 64 <456000000>;
        };
    };
};

```

以下为上述代码块的参数说明：

- **compatible**: 兼容性信息，用于连接驱动程序和目标设备。
- **reg**: 寄存器基本地址。

- **clocks**: 所需要的clocks。
- **clock-names**: 上述时钟的名称, 需要和驱动中定义的一样。
- **reset-gpio**: GPIO复位信号。
- **rotation**: Camera旋转。

5.7. Sensor调试及常见QA

5.7.1. Sensor是否注册成功

输入以下命令查看是否有sensor entity:

```
media-ctl -p -d /dev/media0
```

5.7.2. Sensor是否有数据输出

条件允许的情况下, 直接测量硬件信号是最准备的手段。是否有信号输出, 以及信号幅度是否正确等等。

软件上可以通过判断VIN是否有中断, 比如kernel log是否有VIN 报告错误。

5.7.3. 检查control是否生效

可以利用v4l2-ctl设置相关的参数, 如flip并生成图片, 查看sensor的controls是否有生效。例如改变flip, 图片亮度是否有翻转。