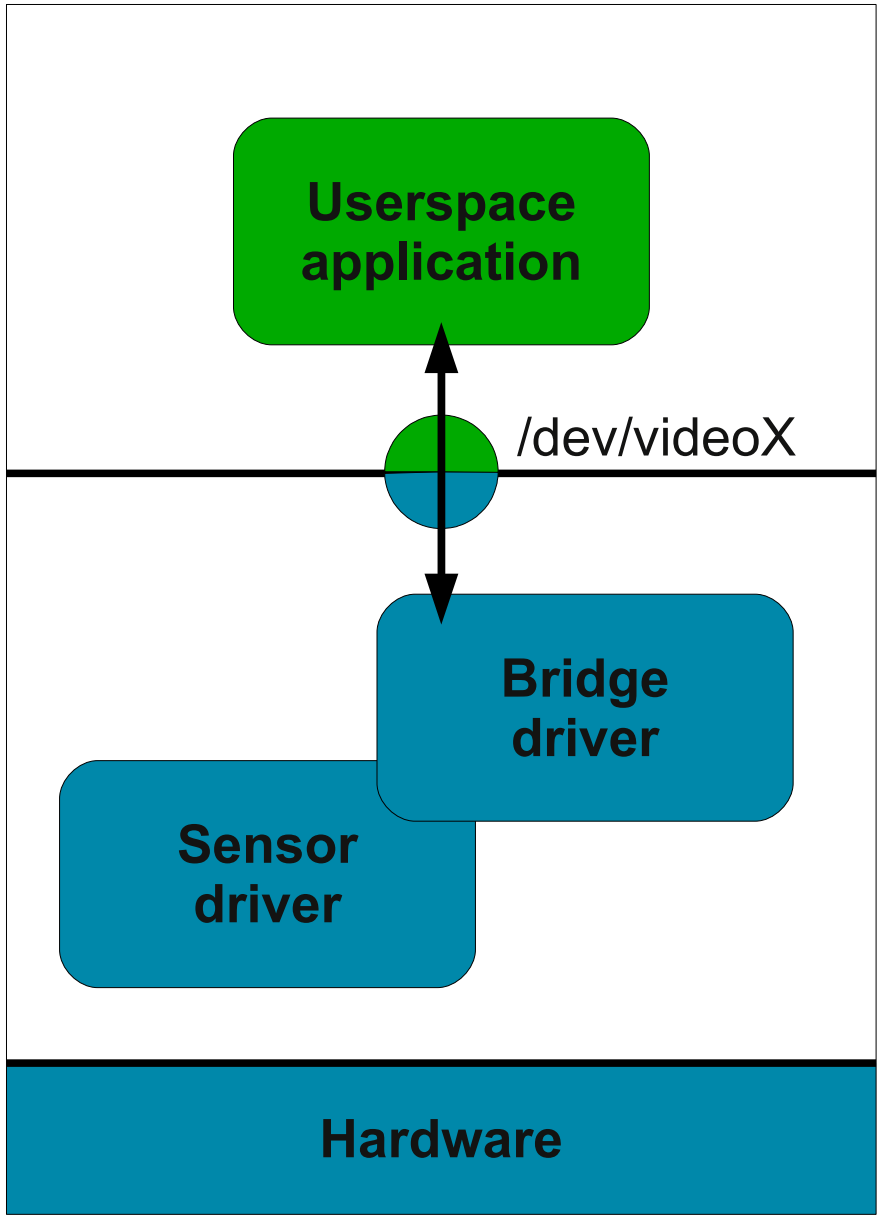
Media controller

Finally getting audio and video to play together

Linux Plumbers Conference 2010

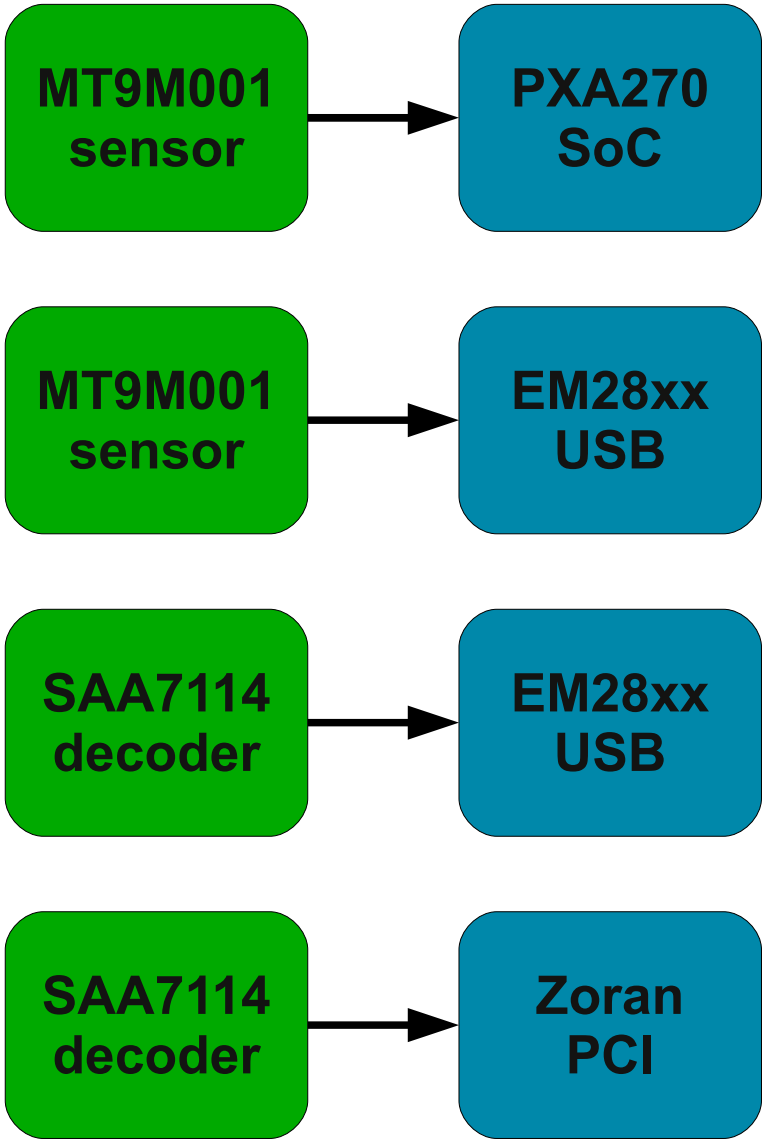
Laurent Pinchart
laurent.pinchart@ideasonboard.com



Embedded SoC camera

- soc_camera
- v4l2_device
- v4l2_subdev

Embedded camera

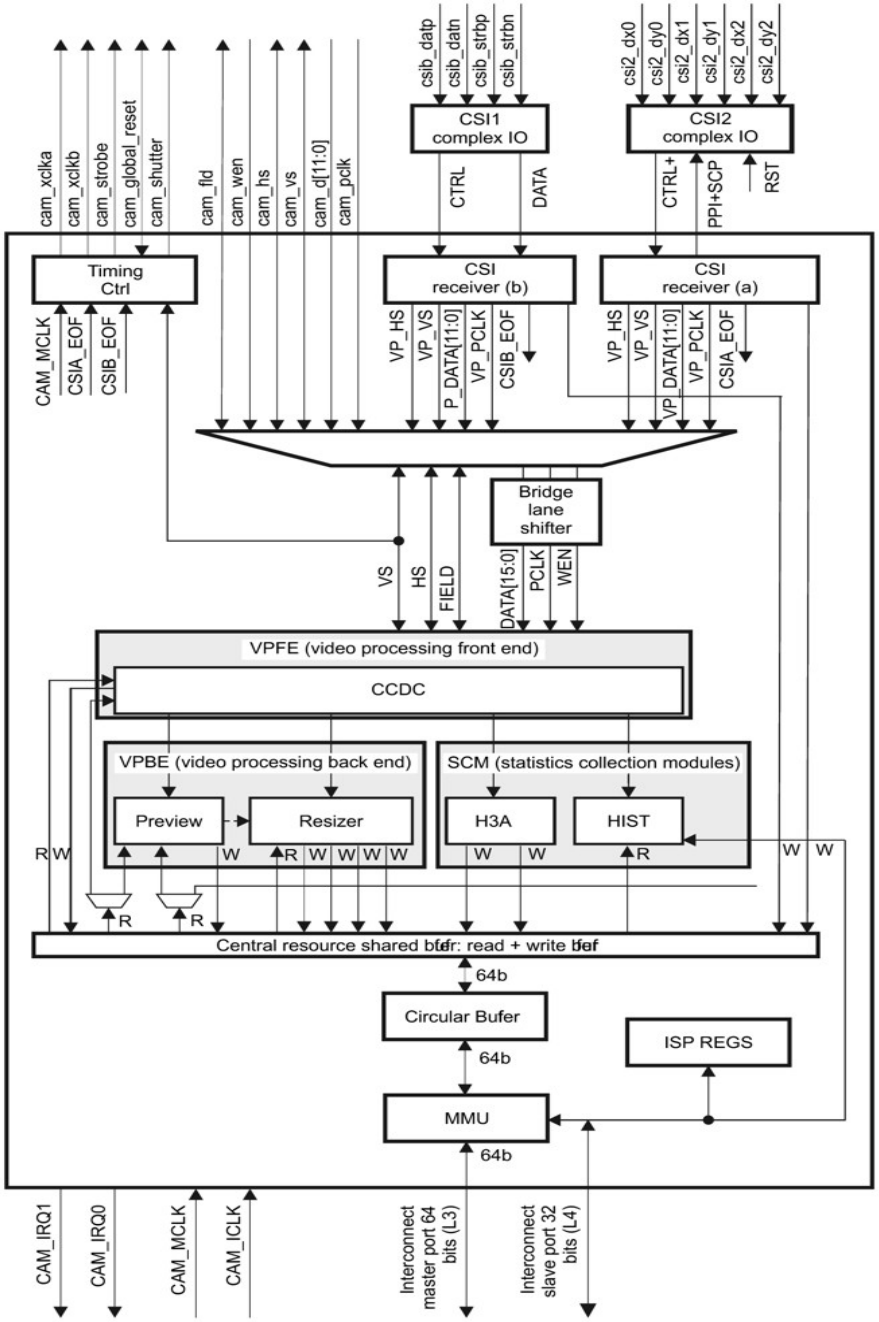


- In-kernel functional abstraction layer developed by Hans Verkuil
- Designed for on-board external devices (sensors, tuners, audio codecs, …)
- Reusability, Reusability, Reusability

V4L2 subdevice

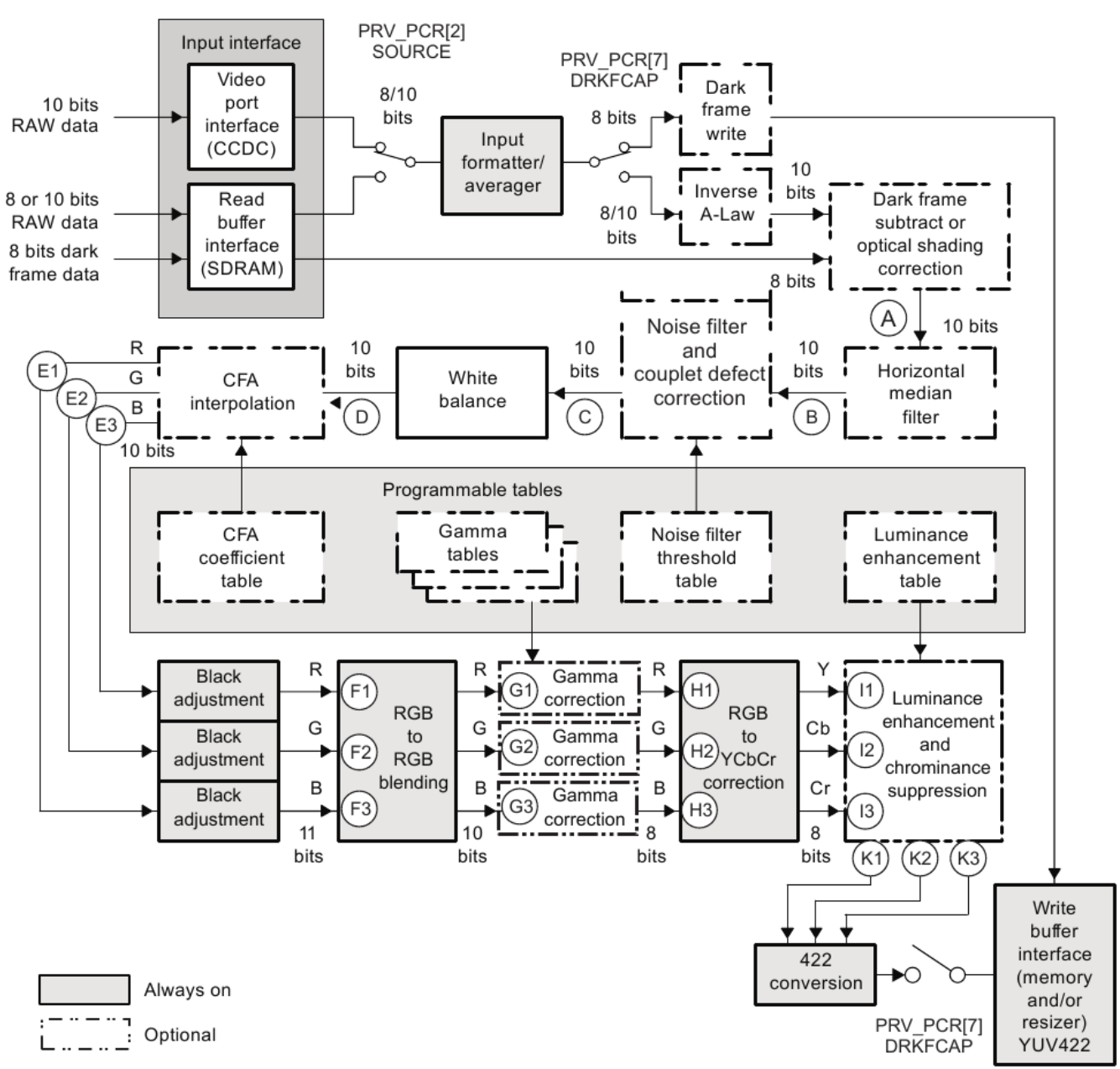
OMAP3430 ISP

- Reconfigurable pipeline
- Parallel processing
- Memory-to-memory paths
- Fine-grain parameters

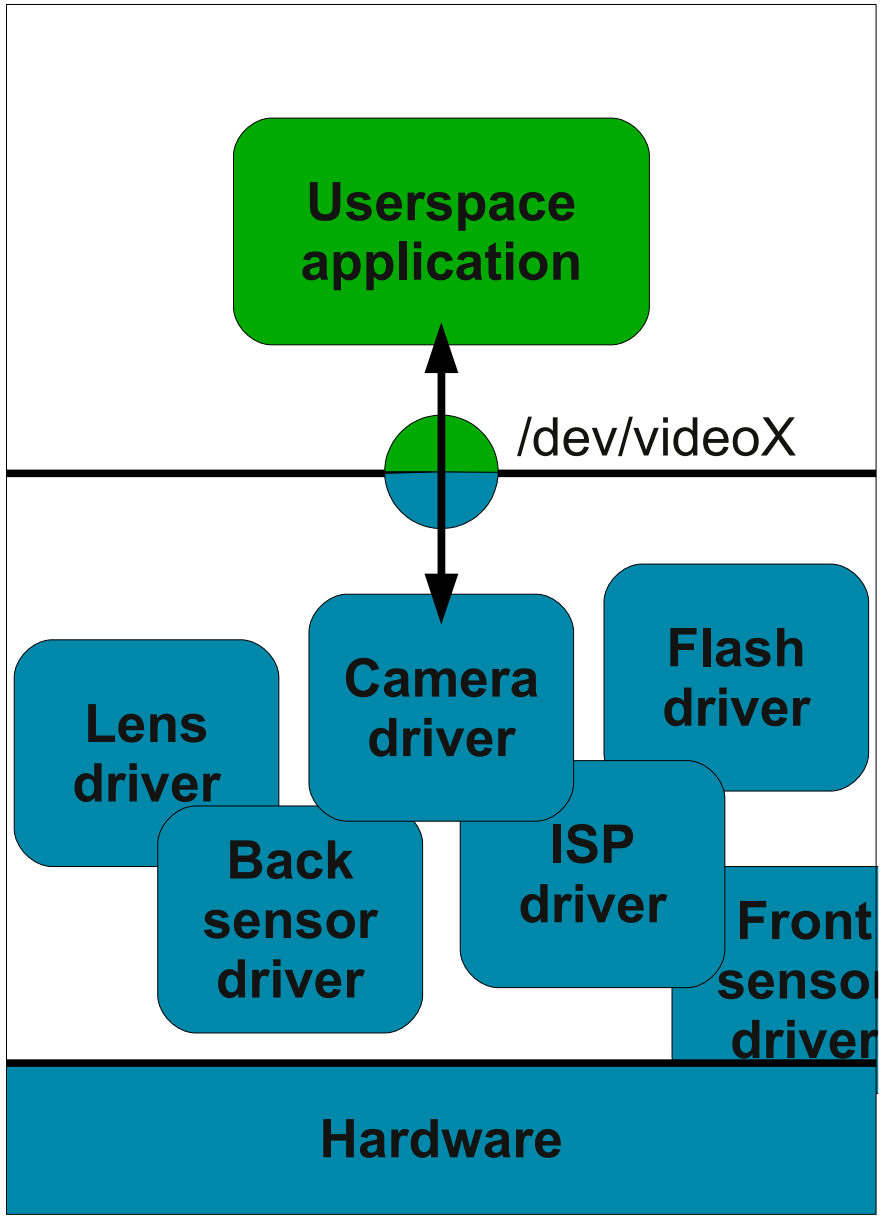


Drawing is © Texas Instrument

OMAP3430 ISP



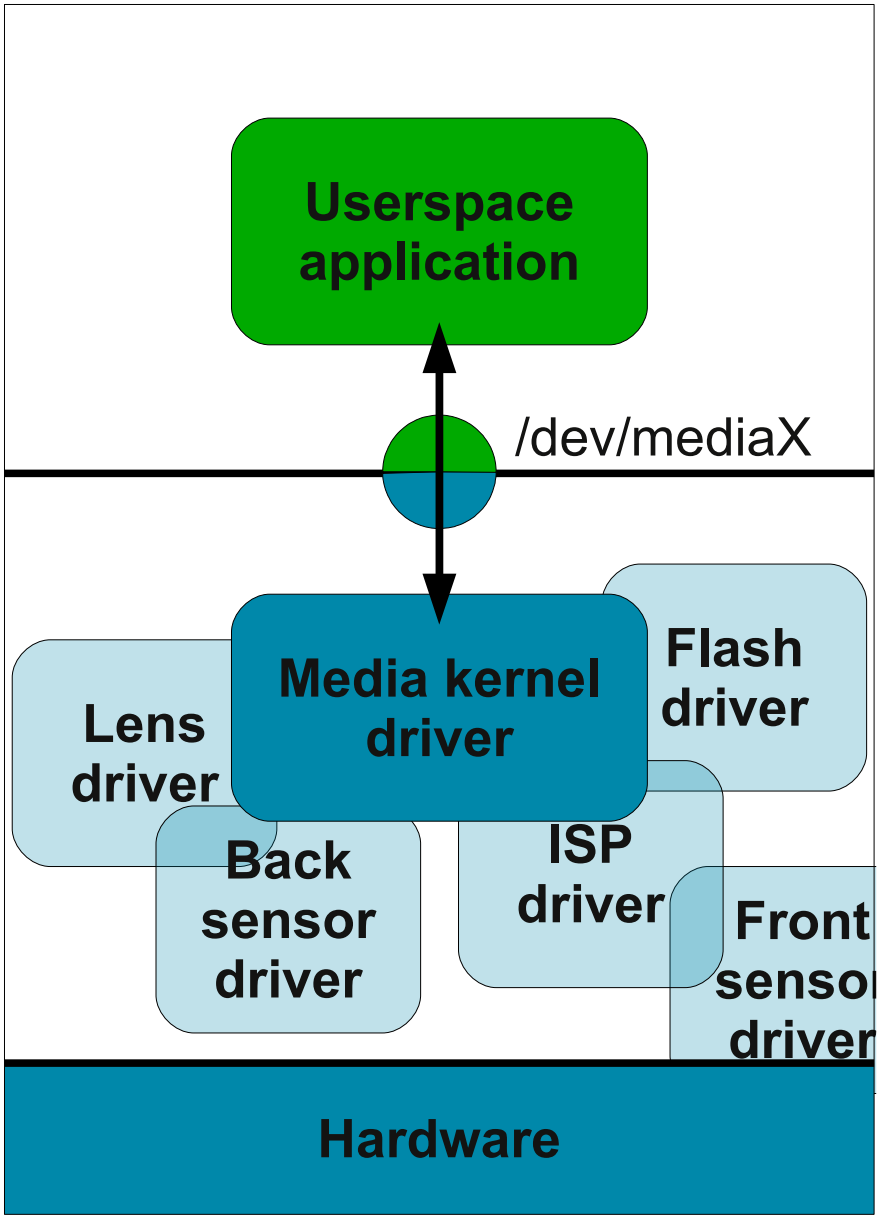
OMAP3430 ISP Preview engine



Highly complex devices

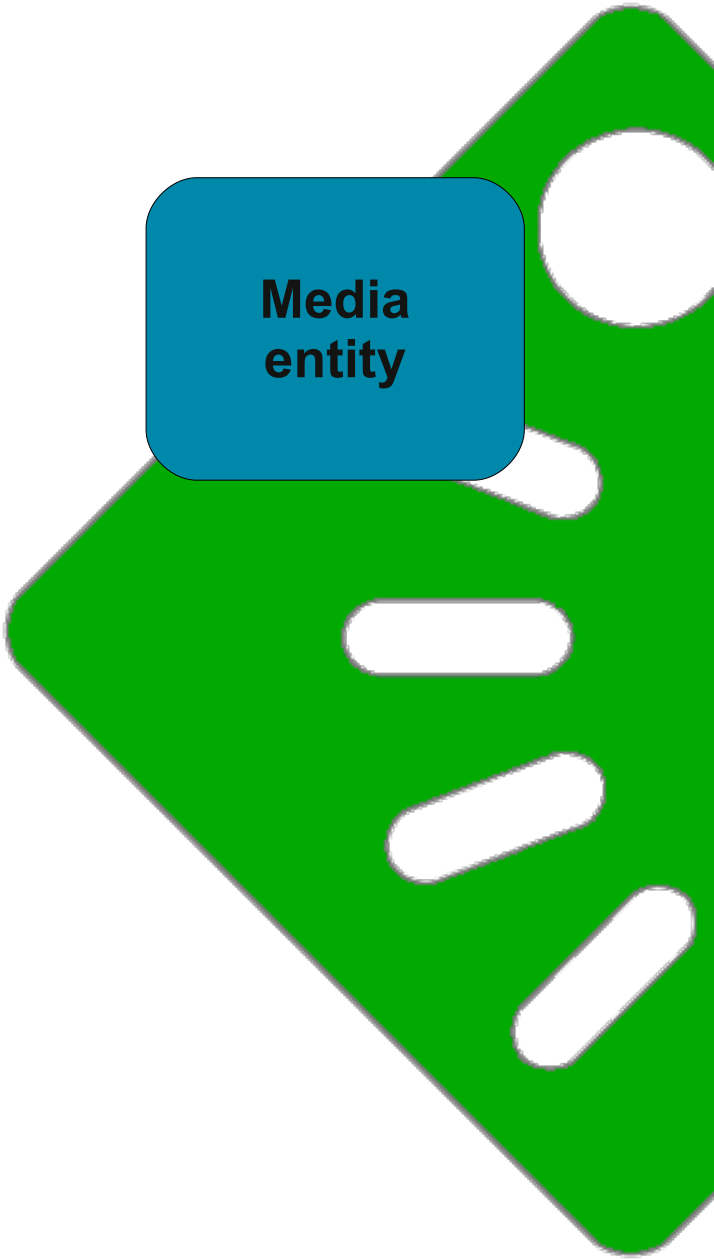
- Multiple inputs
- Multiple streams
- Configurable pipeline

Embedded mess



Media controller

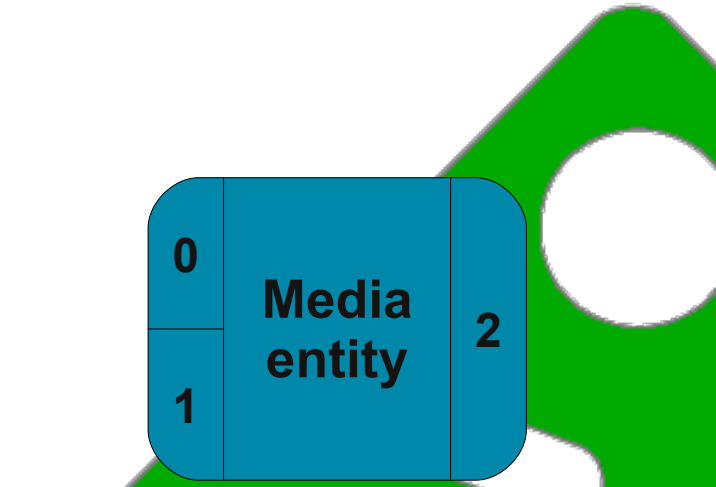
```
struct media_entity
{
    u32 id;
    const char *name;
    u32 type;
    u32 revision;
    unsigned long flags;
    u32 group_id;
    ...
};
```



- media_entity::type
  - MEDIA_ENTITY_TYPE_NODE
  - MEDIA_ENTITY_TYPE_SUBDEV
- media_entity::flags
  - MEDIA_ENTITY_FLAG_DEFAULT

Media entity

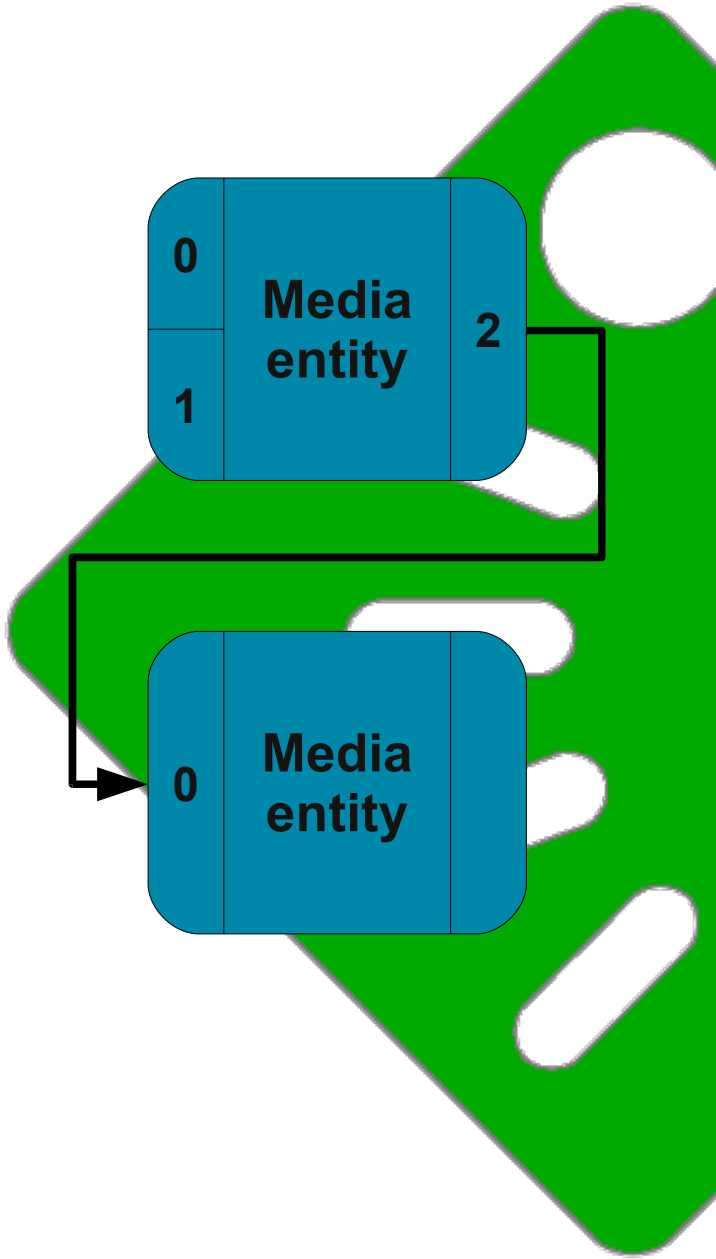
```
struct media_entity
{
    ...
    u16 num_pads;
    struct media_pad *pads;
    ...
};

struct media_pad
{
    u16 index;
    unsigned long flags;
};
```



- media_entity_pad::flags
  - MEDIA_PAD_FLAG_INPUT
  - MEDIA_PAD_FLAG_OUTPUT

Media entity - Pads

```
struct media_entity
{
    ...
    u16 num_links;
    struct media_link *links;
    ...
};

struct media_entity_link
{
    struct media_pad *source;
    struct media_pad *sink;
    unsigned long flags;
};
```



- media_entity_link::flags
  - MEDIA_LINK_FLAG_ACTIVE
  - MEDIA_LINK_FLAG_IMMUTABLE

Media entity - Links

Media entity

Initialize entity

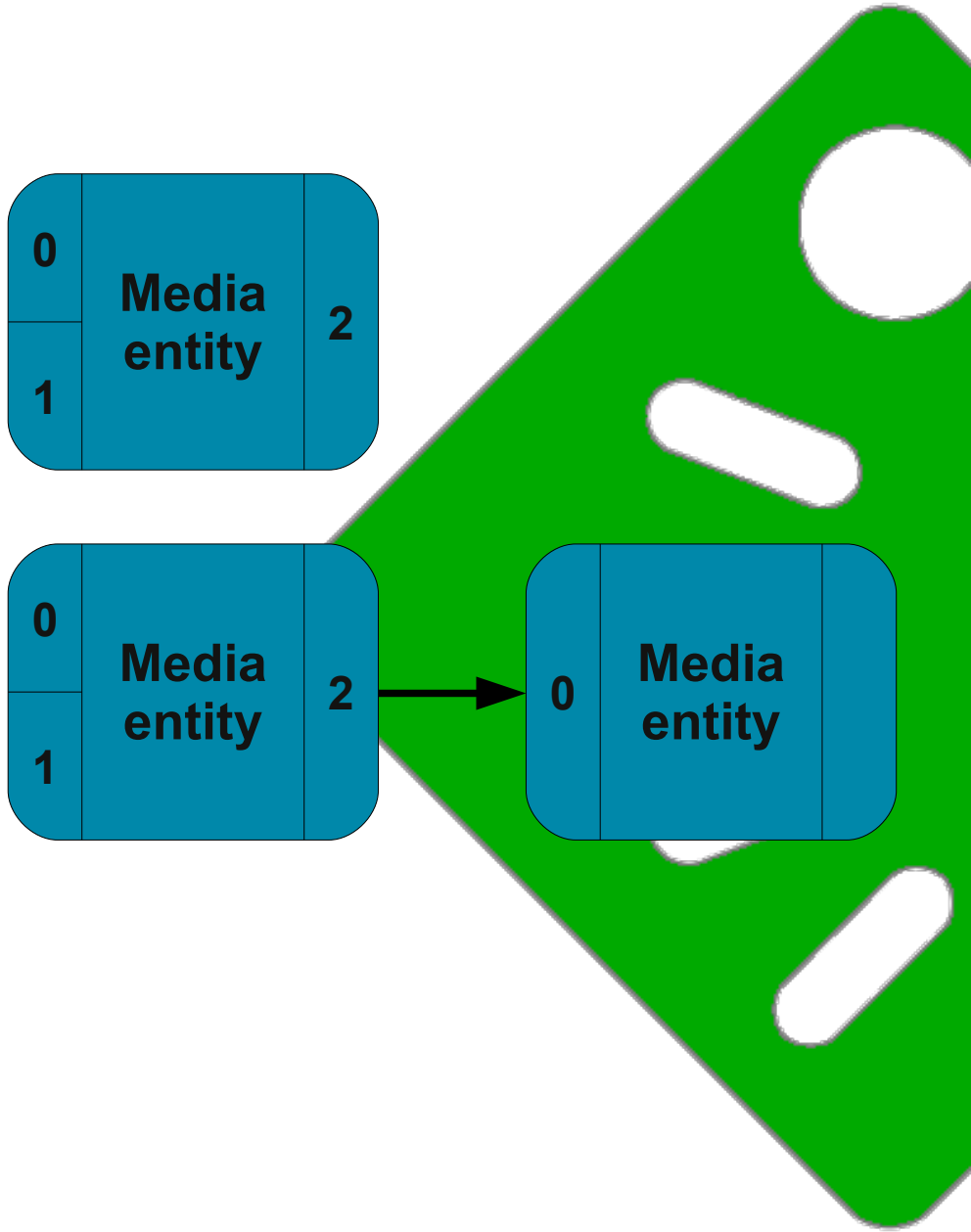
media_entity_init

Create links

media_entity_create_link

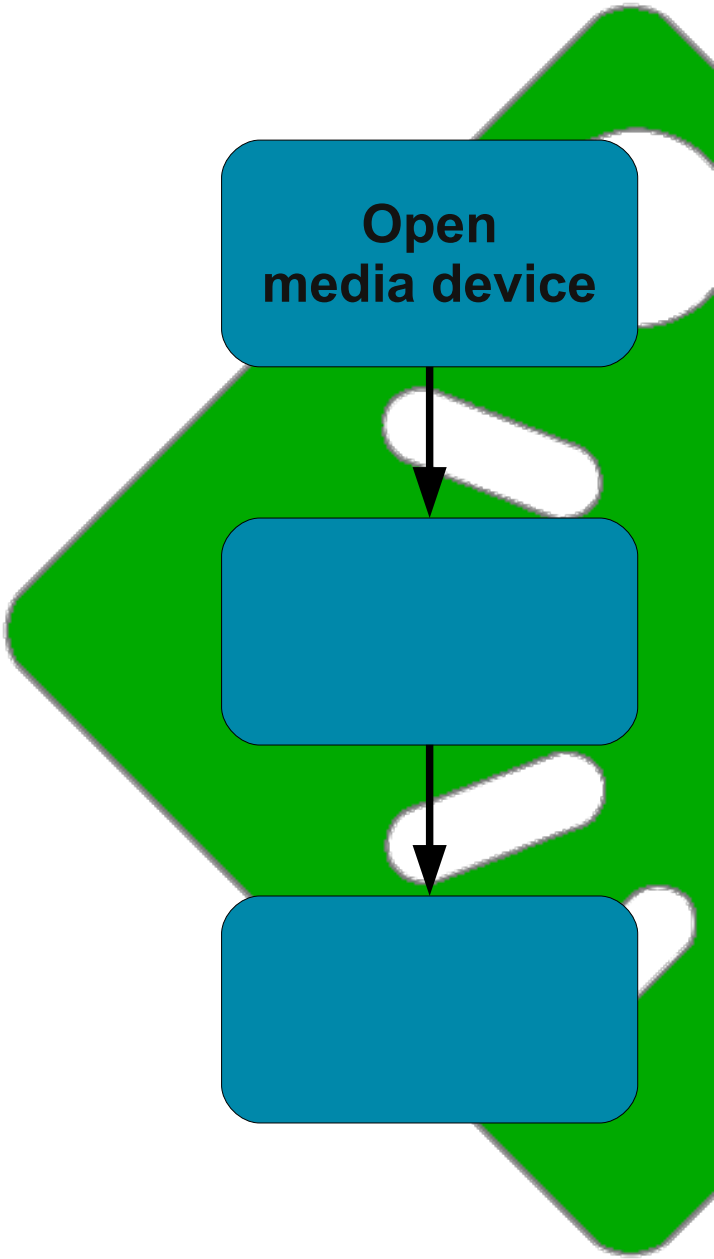
Register entity

media_device_register_entity



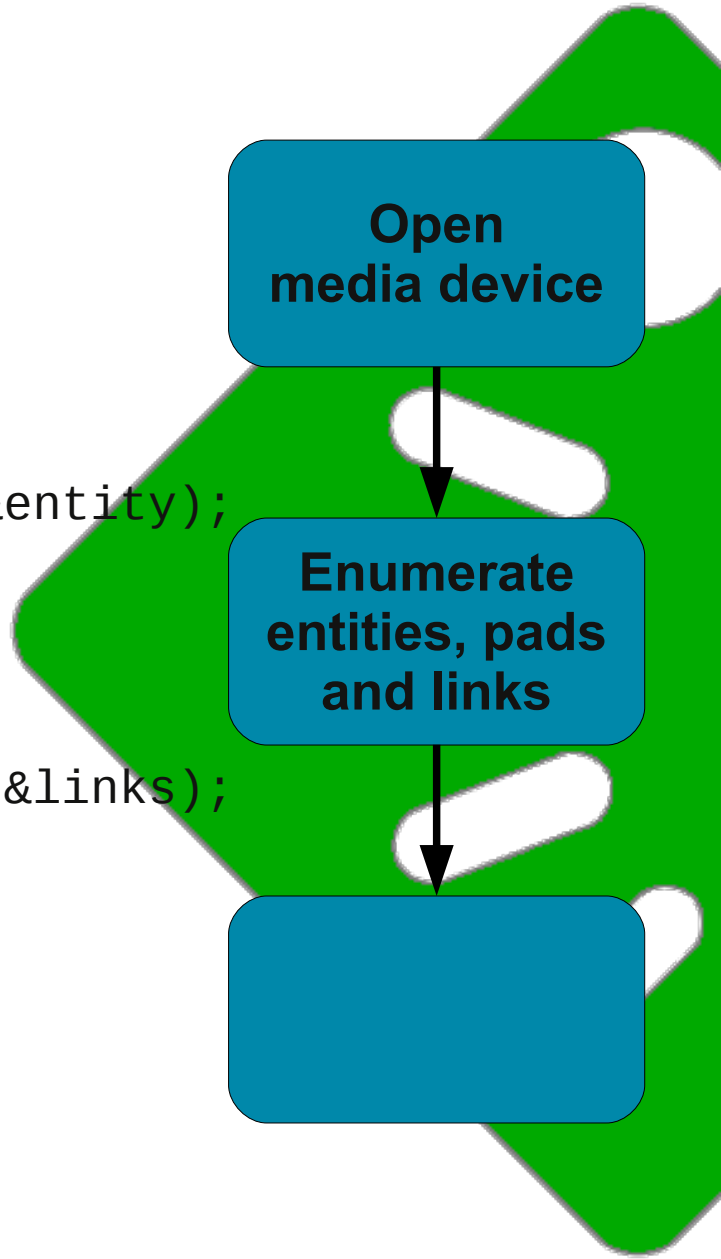
Media entity – Kernel API

```
int fd;

fd = open(“/dev/media0”, O_RDWR);
```

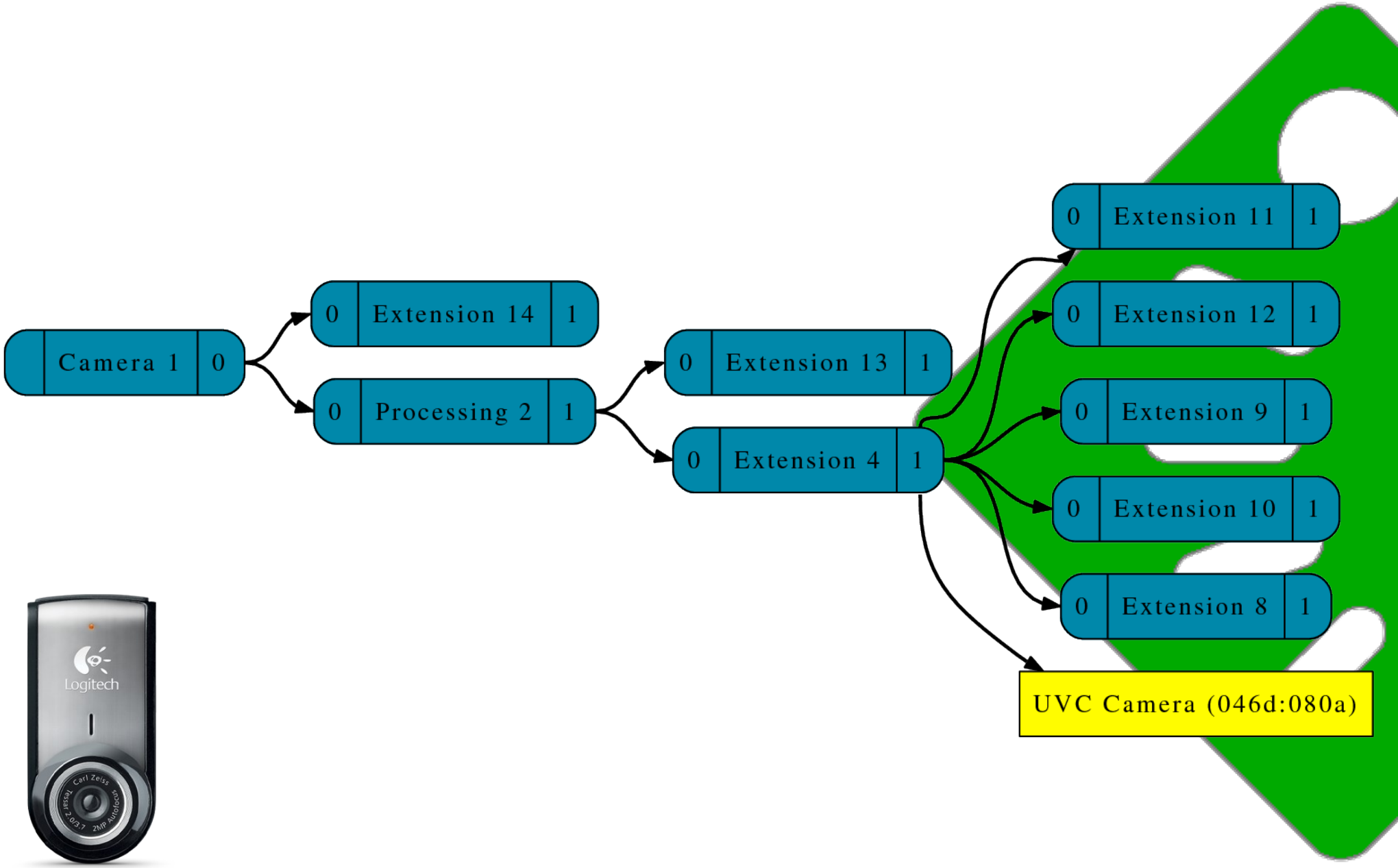


Media controller – Userspace API

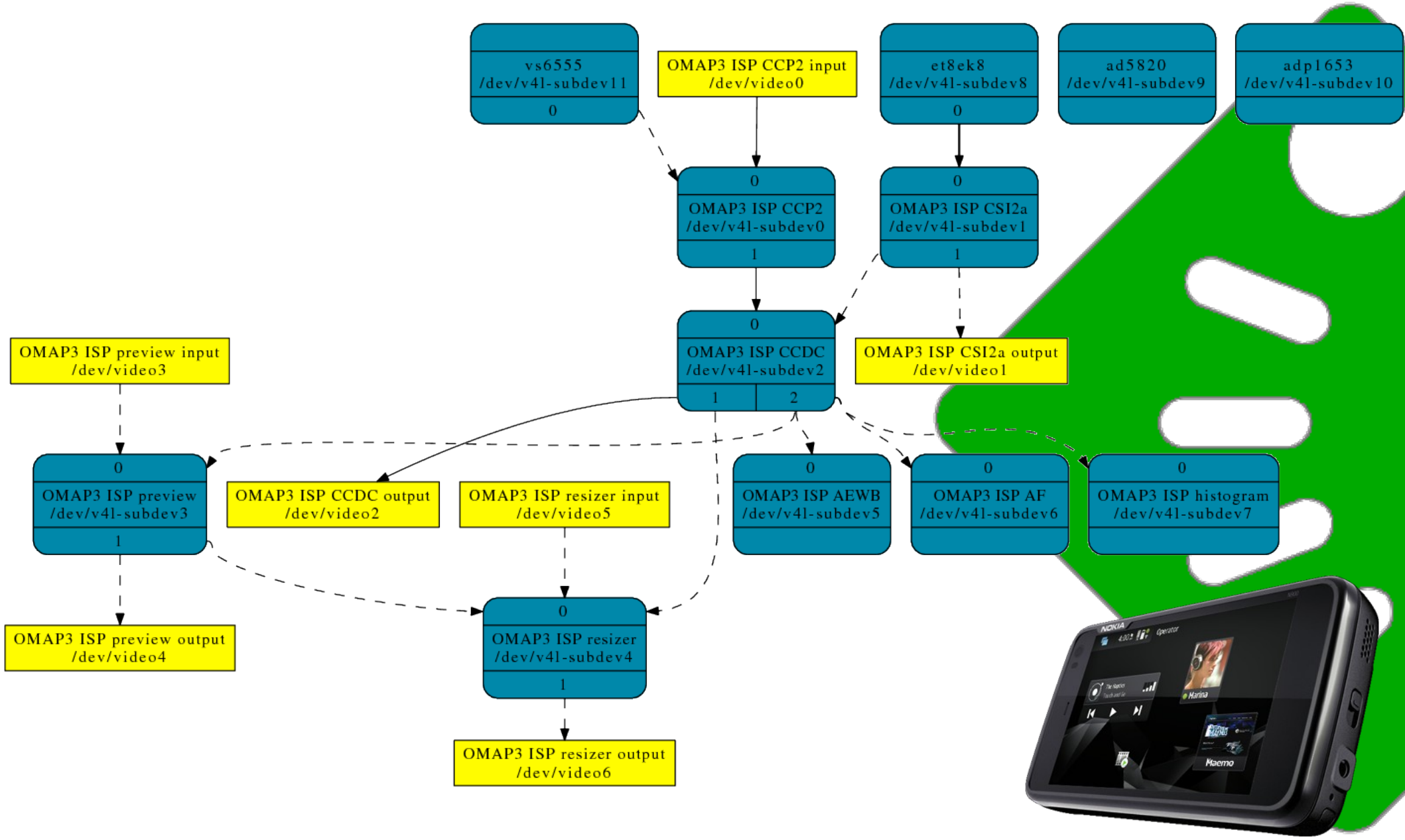
```
int fd;

fd = open(“/dev/media0”, O_RDWR);

while (1) {
    struct media_entity entity;
    struct media_links links;

    ret = ioctl(fd, MEDIA_IOC_ENUM_ENTITIES, &entity);
    if (ret < 0)
        break;

    while (1) {
        ret = ioctl(fd, MEDIA_IOC_ENUM_LINKS, &links);
        if (ret < 0)
            break;
    }
}
```



Media controller – Userspace API

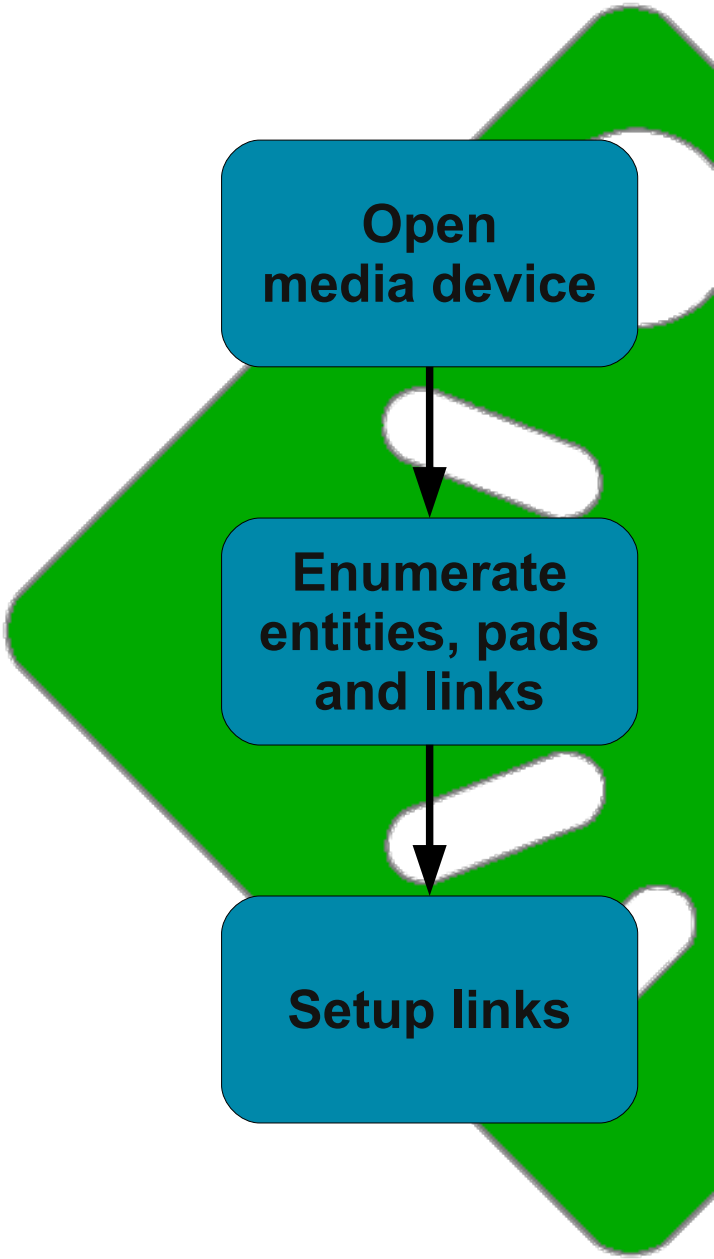


Logitech Portable Webcam C905


vs6555
/dev/v4l-subdev11
0
OMAP3 ISP CCP2 input
/dev/video0
et8ek8
/dev/v4l-subdev8
0
ad5820
/dev/v4l-subdev9
adp1653
/dev/v4l-subdev10
0
OMAP3 ISP CCP2
/dev/v4l-subdev0
1
0
OMAP3 ISP CSI2a
/dev/v4l-subdev1
1
0
OMAP3 ISP CCDC
/dev/v4l-subdev2
1
2
OMAP3 ISP preview input
/dev/video3
OMAP3 ISP CSI2a output
/dev/video1
0
OMAP3 ISP preview
/dev/v4l-subdev3
1
OMAP3 ISP CCDC output
/dev/video2
OMAP3 ISP resizer input
/dev/video5
0
OMAP3 ISP AEWB
/dev/v4l-subdev5
0
OMAP3 ISP AF
/dev/v4l-subdev6
0
OMAP3 ISP histogram
/dev/v4l-subdev7
OMAP3 ISP preview output
/dev/video4
0
OMAP3 ISP resizer
/dev/v4l-subdev4
1
OMAP3 ISP resizer output
/dev/video6

Nokia N900

```
struct media_link link;

link.source.entity = OMAP3_ISP_ENTITY_CCDC;
link.source.index = 2;
link.sink.entity = OMAP3_ISP_ENTITY_PREVIEW;
link.sink.index = 0;
link.flags = 0;

ioctl(fd, MEDIA_IOC_SETUP_LINK, &link);

link.source.entity = OMAP3_ISP_ENTITY_CCDC;
link.source.index = 1;
link.sink.entity = OMAP3_ISP_ENTITY_CCDC_OUT;
link.sink.index = 0;
link.flags = MEDIA_LINK_FLAG_ACTIVE;


ioctl(fd, MEDIA_IOC_SETUP_LINK, &link);
```



Media controller – Userspace API

- http://git.linuxtv.org/pinchartl/media.git
  - Documentation/DocBook/v4l
- http://git.ideasonboard.org/?p=media-ctl.git
- http://www.ideasonboard.org/media/

Source code & Documentation