Embedded Linux Conference Europe

OpenIoTSummit Europe

Port your camera in 2018

Jacopo Mondi
ELC-E 2018
Edinburgh, 10/2018

jacopo@jmondi.org

THE LINUX FOUNDATION

Hello!

- My name's Jacopo
  - jacopo@jmondi.org
  - irc: jmondi @ freenode
- Embedded Linux and free software developer
- Working as consultant
- Renesas Electronics upstream kernel team

Hello!

- Thanks Renesas for supporting me in this activity!

Talk outline

- *soc_camera:* what's happening?

- What has changed since the soc_camera days:
  - Board files vs firmware support
  - Device discovery and linking
  - Power Management

- Practical example: ov772x.c
- Q/A

The soc_camera framework

- *soc_camera* was GREAT

- Widely adopted
  - Mainline drivers
  - In BSPs and downstream kernel

What's happening to soc_camera

In mainline kernels:

- *soc_camera* to be 'soon' deprecated (with 'soon' to be defined)
- Last *soc_camera* dependent bridge driver has been re-implemented last year
- *soc_camera* sensor drivers are not officially supported anymore

What's happening to soc_camera

File organization:

- `drivers/media/platform/`
  `drivers/media/platform/soc_camera/`

- `drivers/media/i2c/`
  `drivers/media/i2c/soc_camera/`

- `drivers/media/i2c/`**`mt9t112.c`**
  `drivers/media/i2c/soc_camera/`**`mt9t112.c`**

soc_camera: the good

- Provides and abstraction to the crude V4L2 APIs
- Ease linking bridge drivers to sensor drivers
- Abstract driver operations to a structure of function pointers

soc_camera: the bad

- Designed for board-files:
  - OF and ACPI support is limited (hold on...)
- Use on a set of deprecated operations and frameworks
- Media-controller and subdev APIs are game changers

  The single device node abstraction is showing its age...

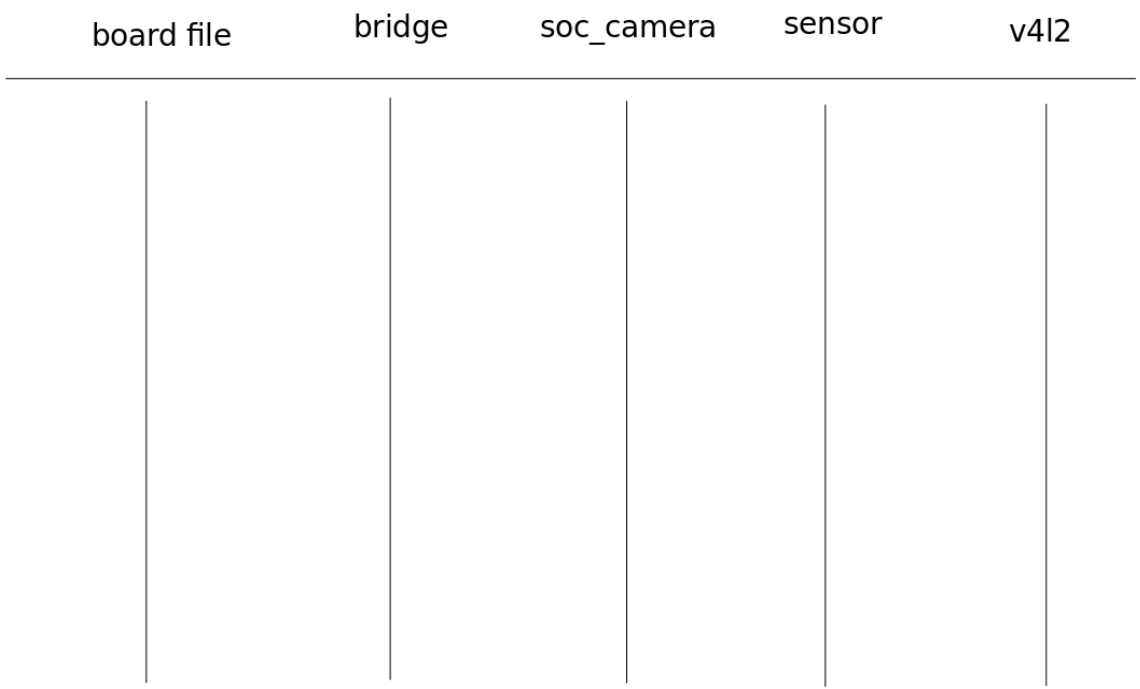
What has changed?

- device discovery and linking:
  - notifier(s) and async matching
- power management
- clocks and regulators handling

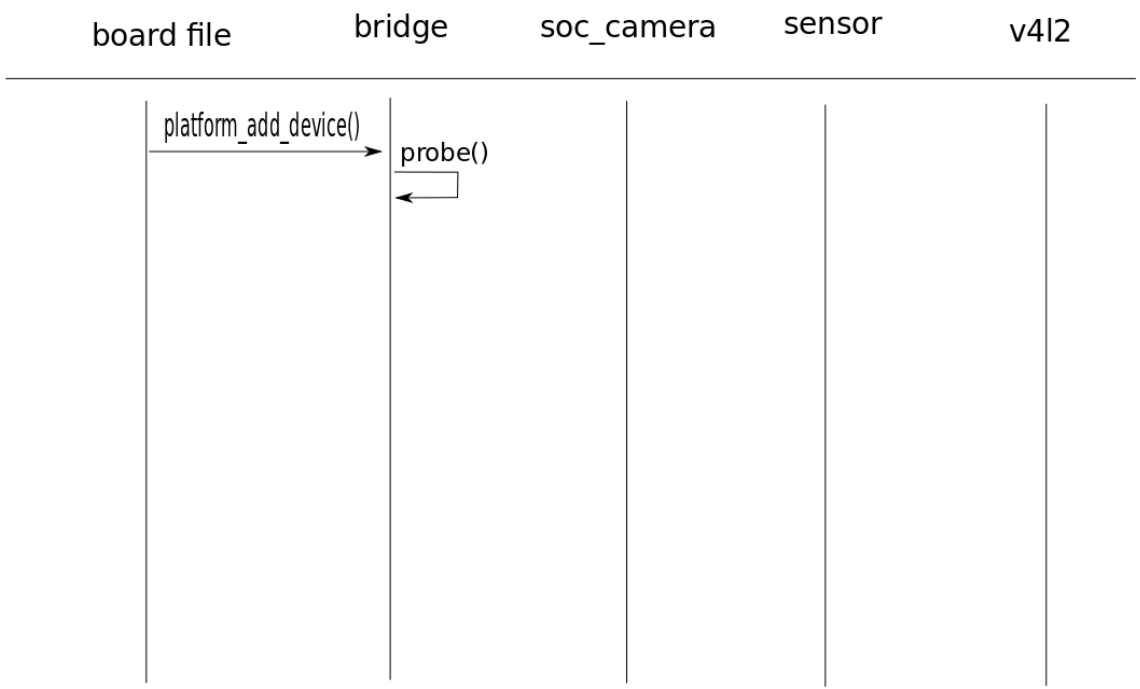
video4linux2: device probing

- Device matching and probing:
  - The old world: i2c addresses based
  - The new world: firmware based

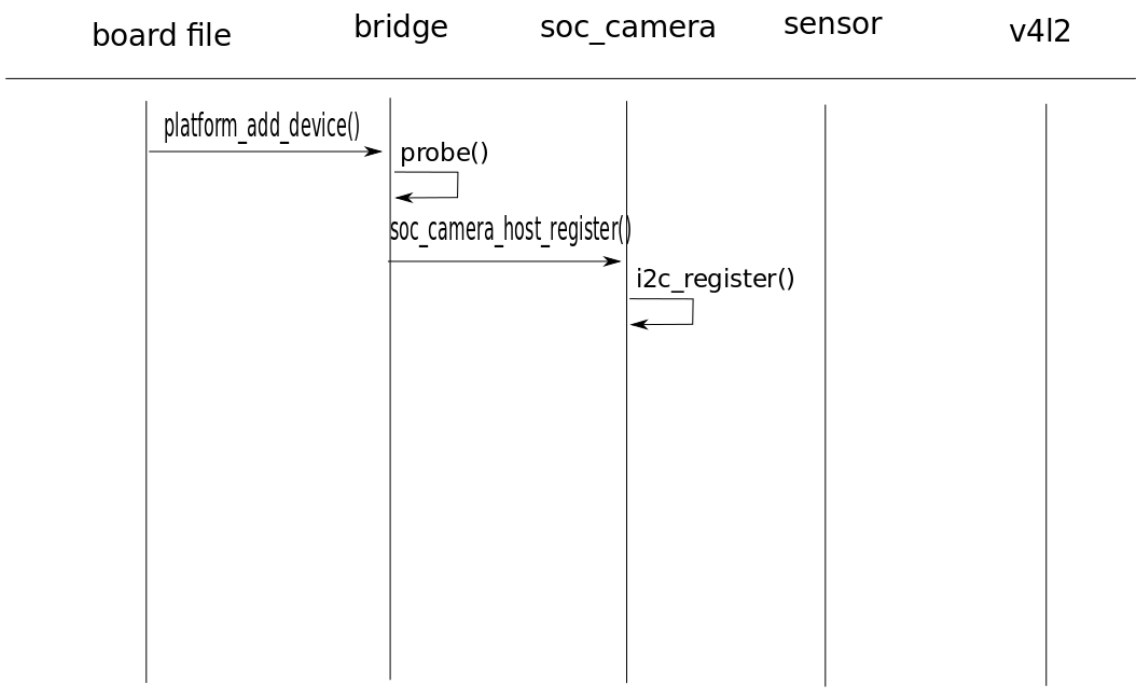
video4linux2: device probing



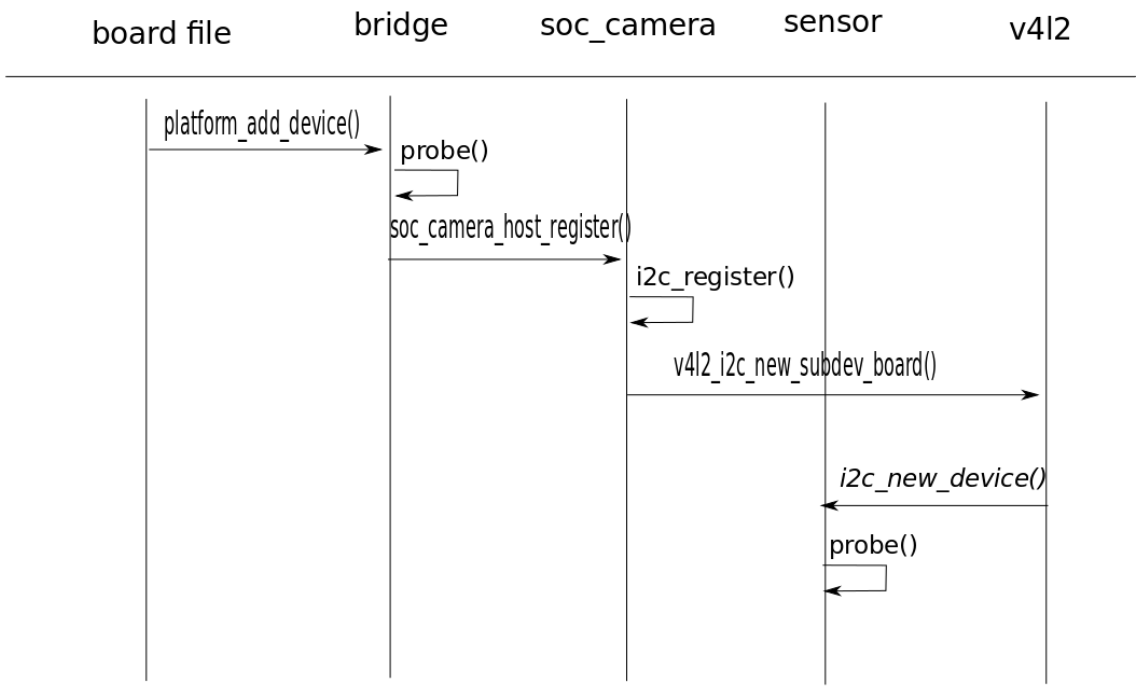
video4linux2: device probing



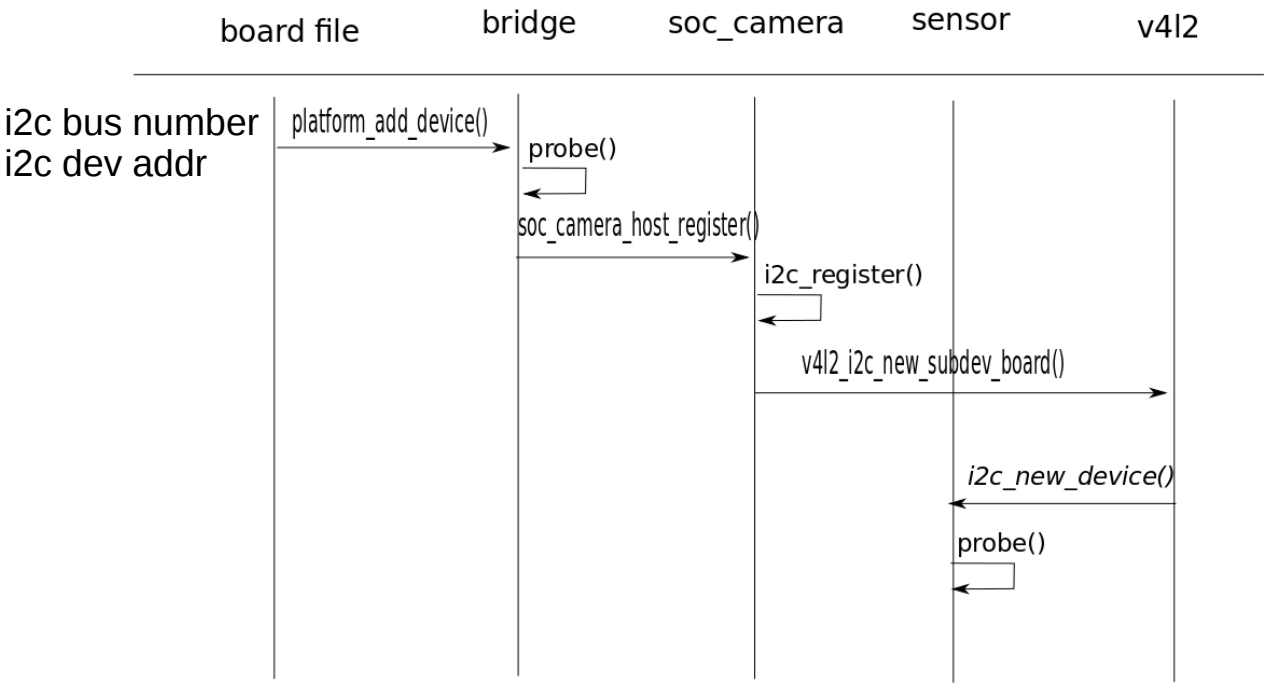
video4linux2: device probing



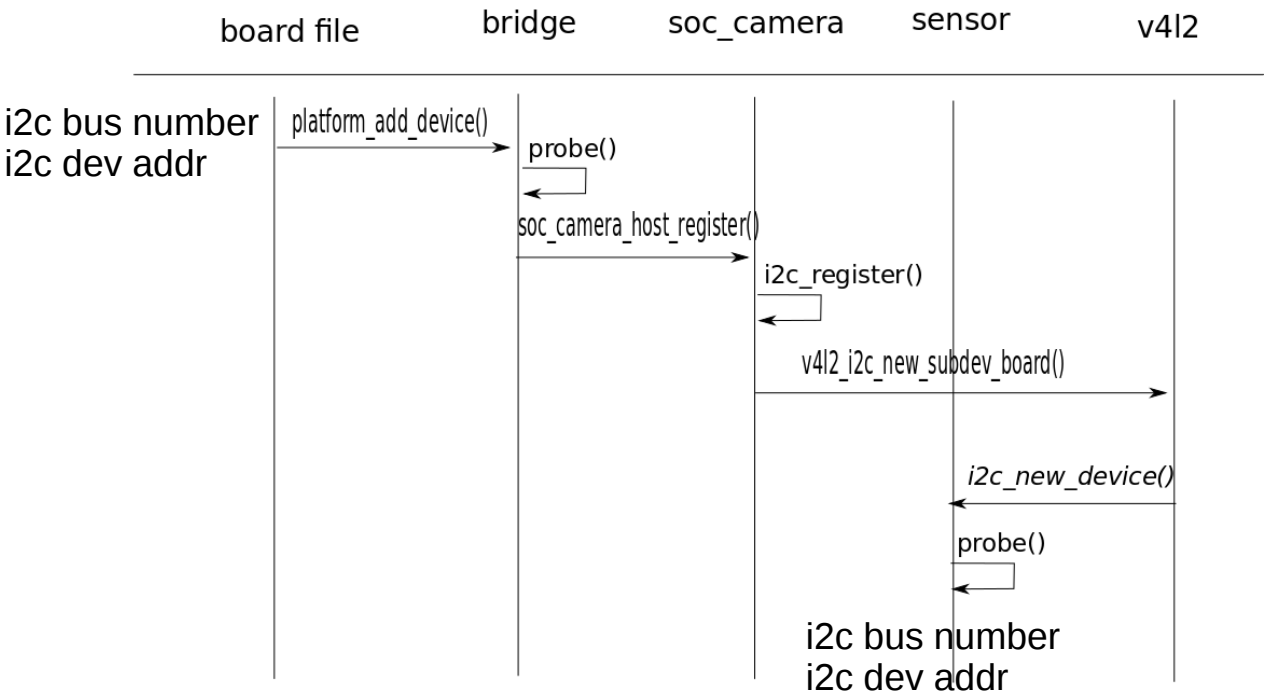
video4linux2: device probing



video4linux2: device probing



video4linux2: device probing



video4linux2: device probing

- Old world
  - Device identified by i2c bus number and i2c address
  - Device probing is sequential
  - Guarantees a bridge driver probes before the sensor driver it will connect to

video4linux2: device probing

- New world
  - Devices are created parsing a firmware description of the system (OF or ACPI)
  - Device are identified by they ‘firmware node’ references
  - No guarantees on the probing order

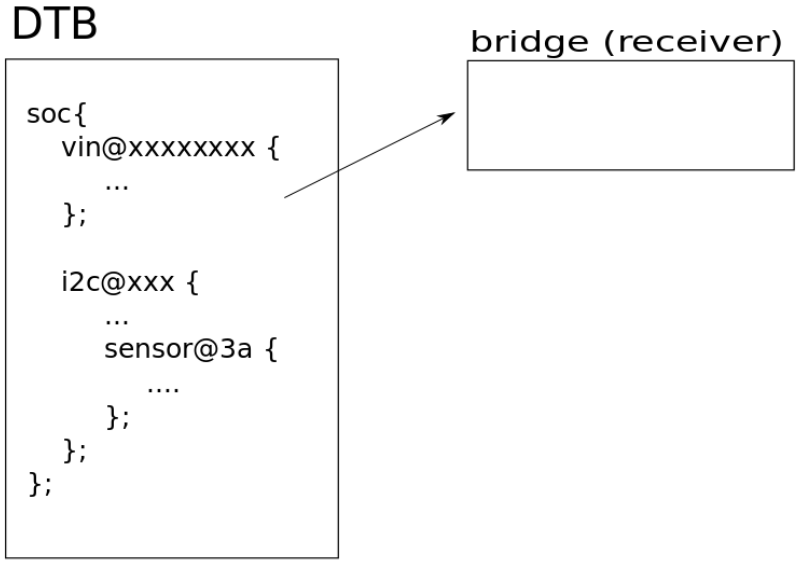
video4linux2: device probing

DTB

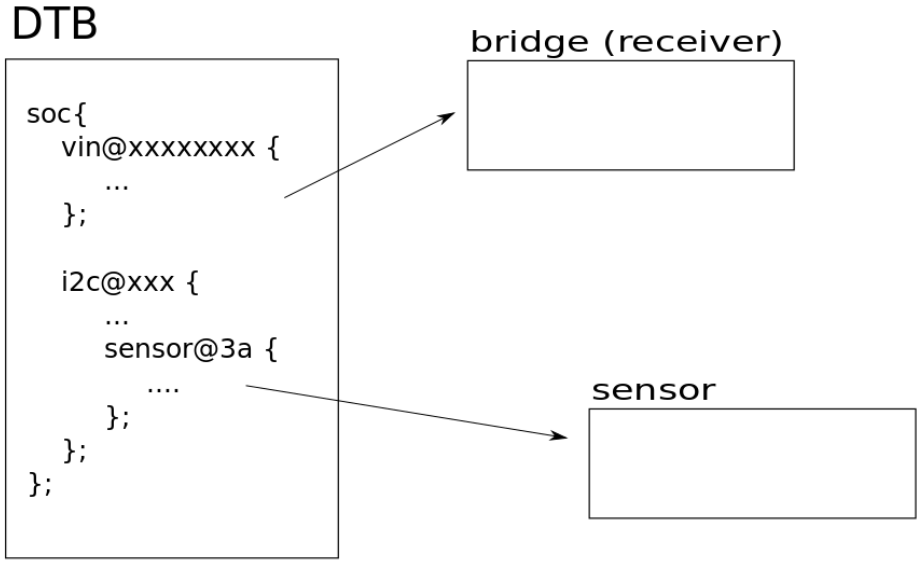
```
soc{
   vin@xxxxxxxx {
      ...
   };

   i2c@xxx {
      ...
      sensor@3a {
         ....
      };
   };
};
```

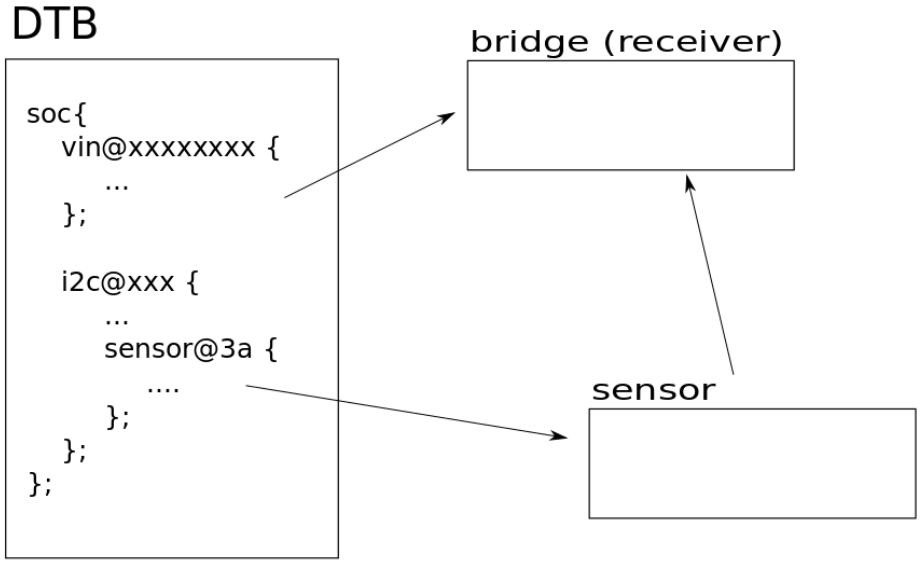
video4linux2: device probing



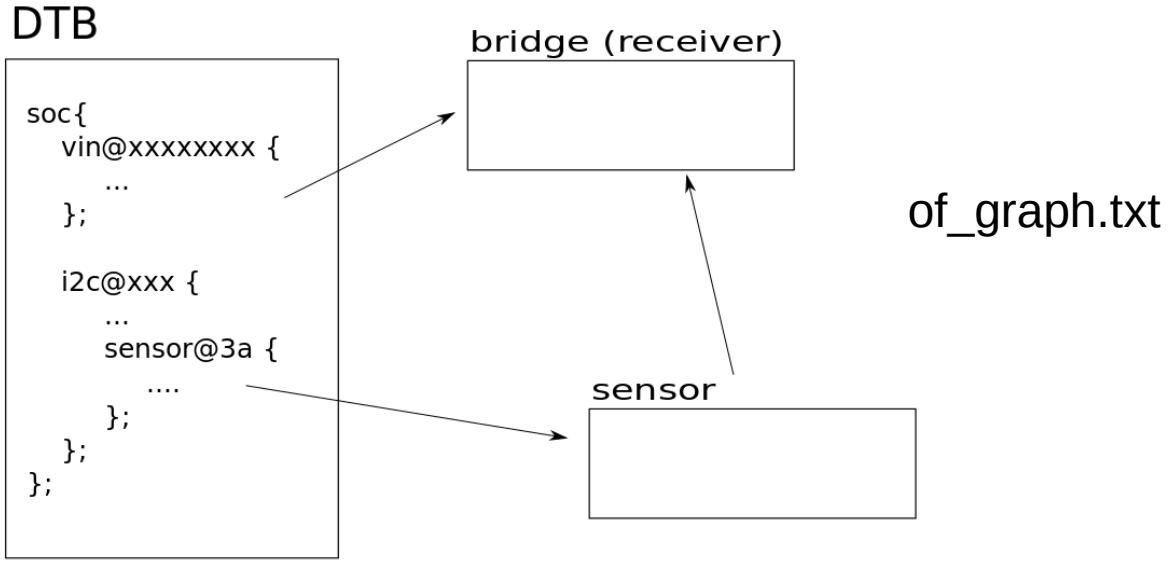
video4linux2: device probing



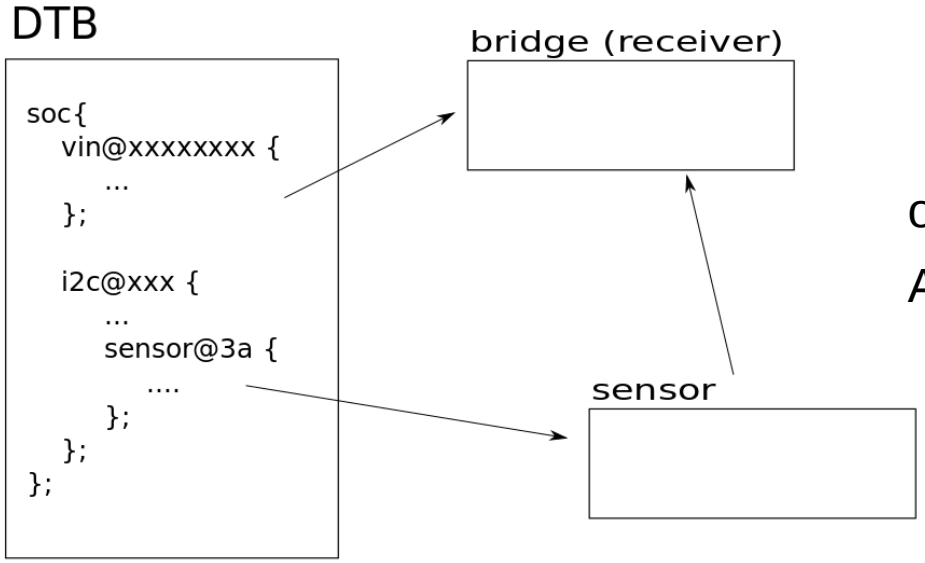
video4linux2: device probing



video4linux2: device probing



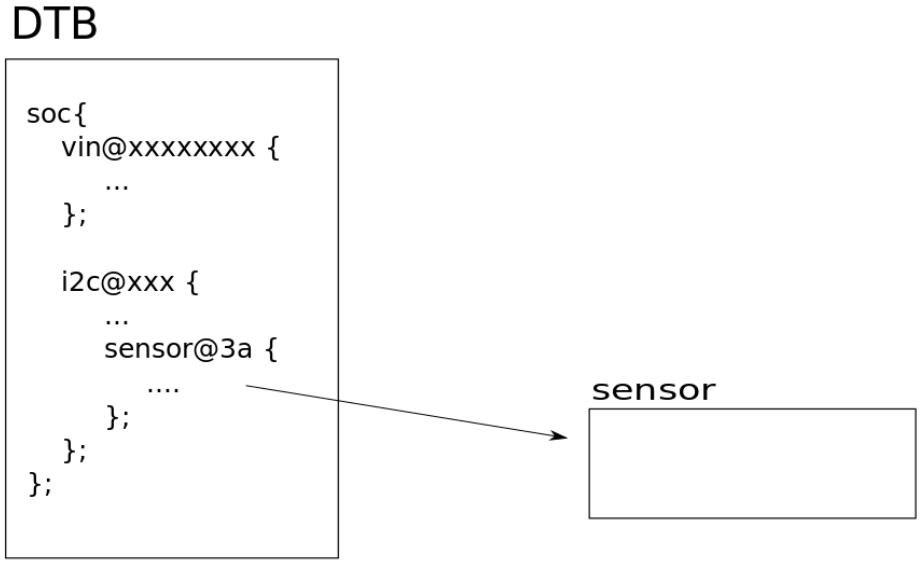
video4linux2: device probing



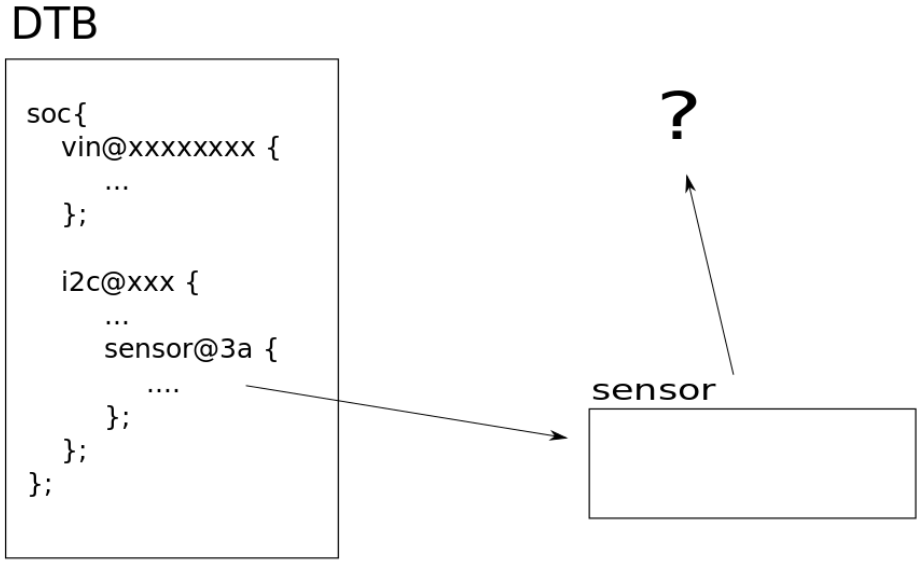
of_graph.txt

ACPI_DSD extension

video4linux2: device probing



video4linux2: device probing



video4linux2: device probing

- Device probing is now *asynchronous*
- Devices are now identified by their firmware node references

video4linux2: device probing

- Device probing is now *asynchronous*
- Devices are now identified by their firmware node references

- V4L2 framework support
  - *v4l2-async*
  - *v4l2-fwnode*

video4linux2: device probing

Bridge (receiver)

v4l2-fwnode

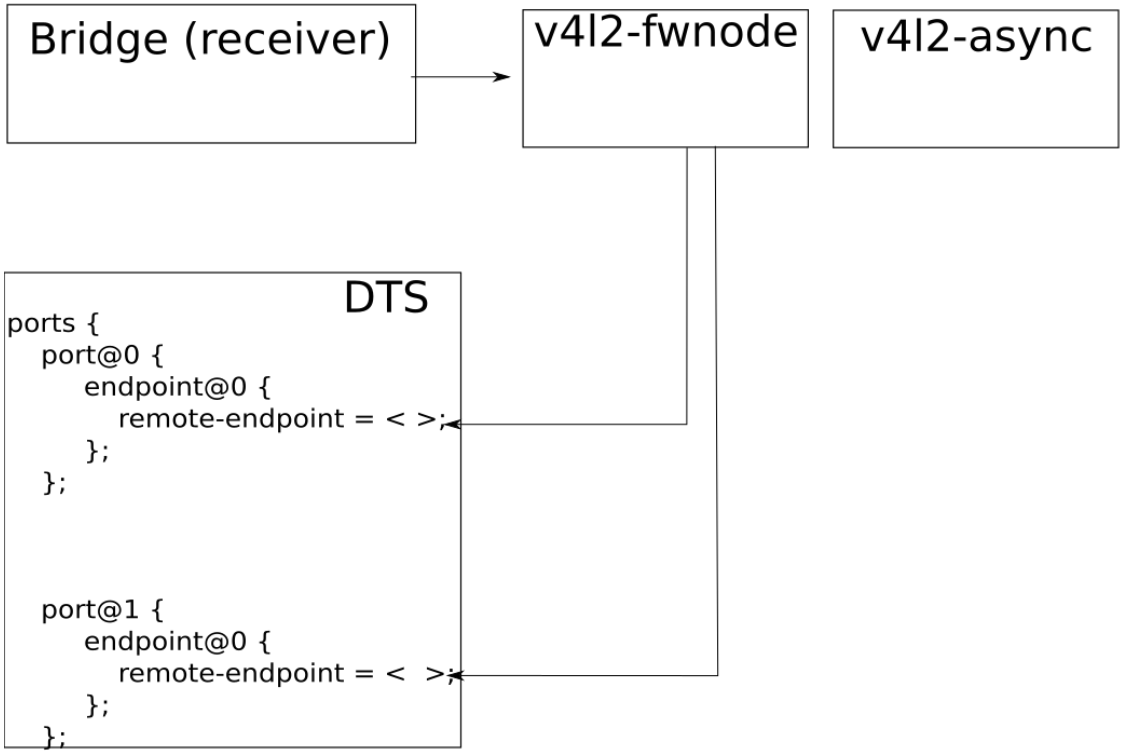
v4l2-async

DTS

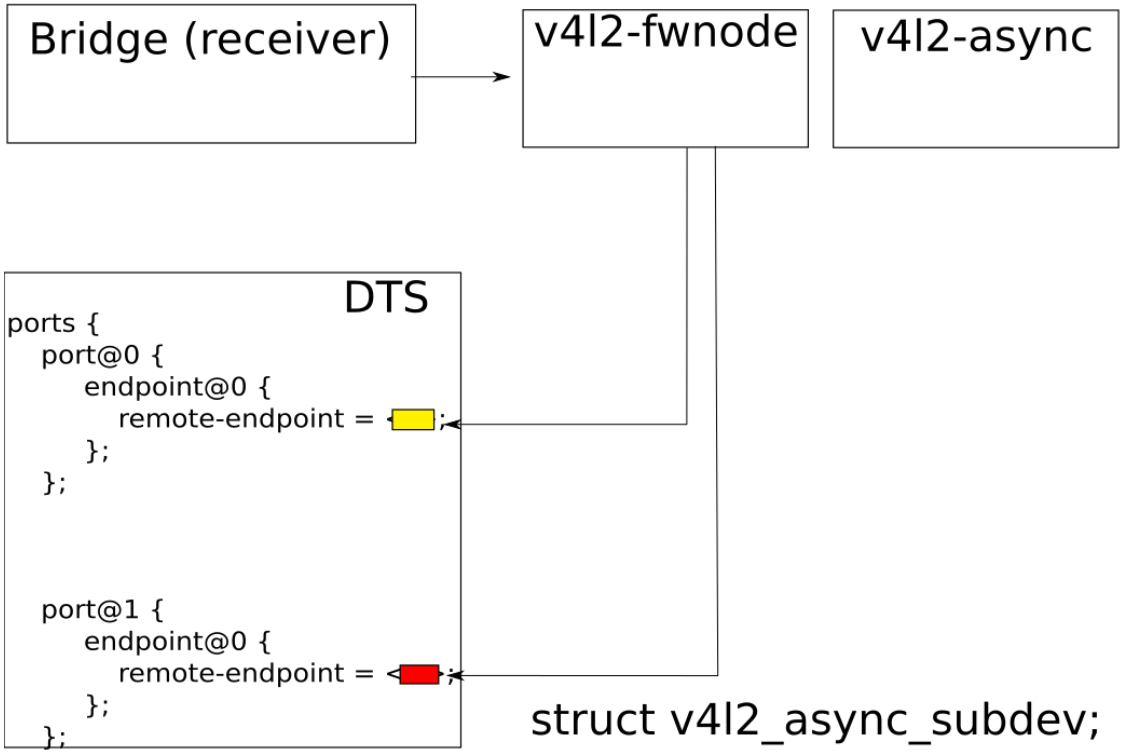
```
ports {
    port@0 {
        endpoint@0 {
            remote-endpoint = < >;
        };
    };


    port@1 {
        endpoint@0 {
            remote-endpoint = <  >;
        };
    };
```

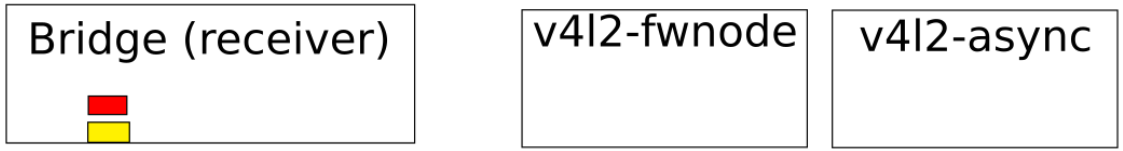
video4linux2: device probing



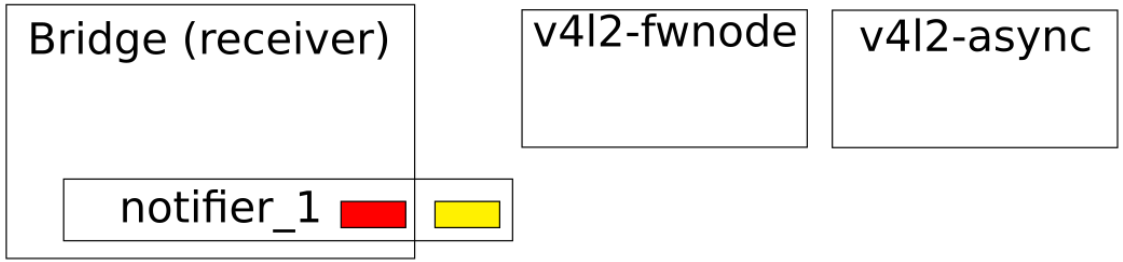
video4linux2: device probing



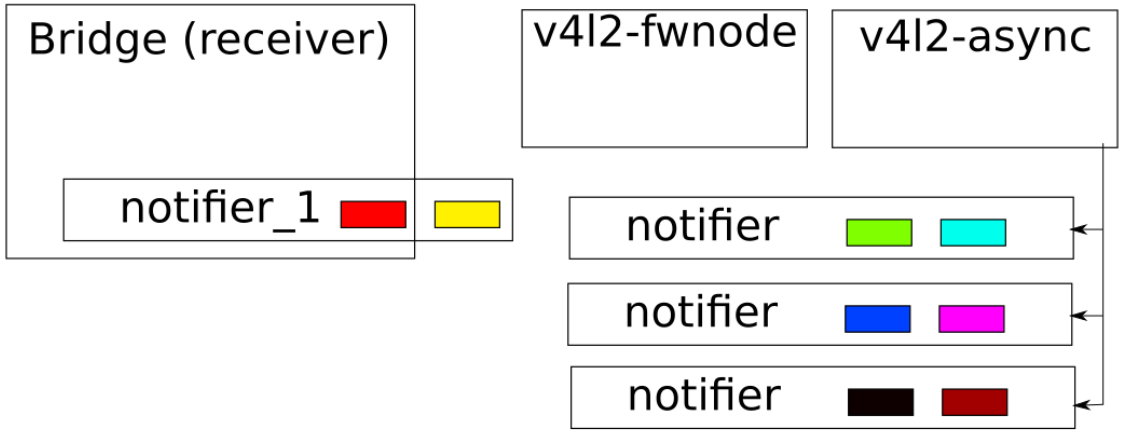
video4linux2: device probing



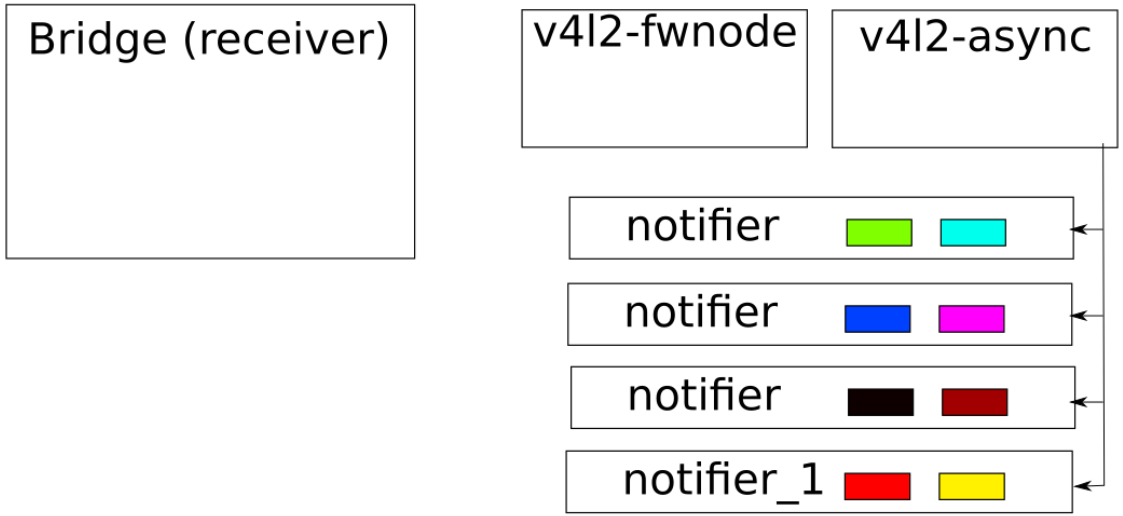
video4linux2: device probing



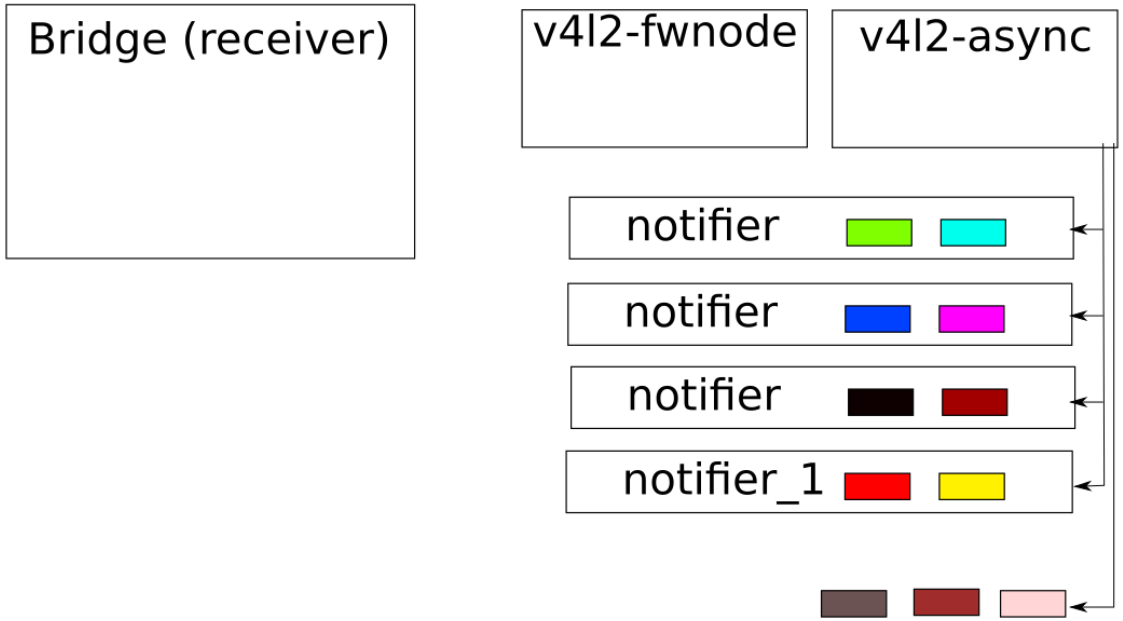
video4linux2: device probing



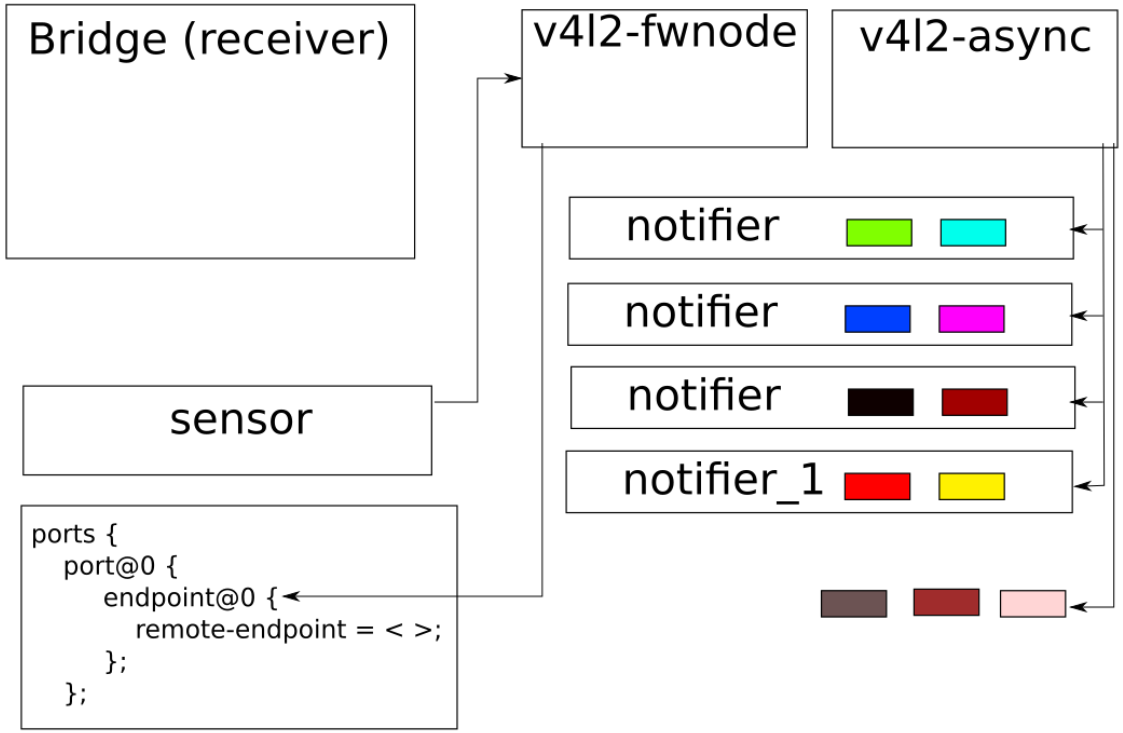
video4linux2: device probing



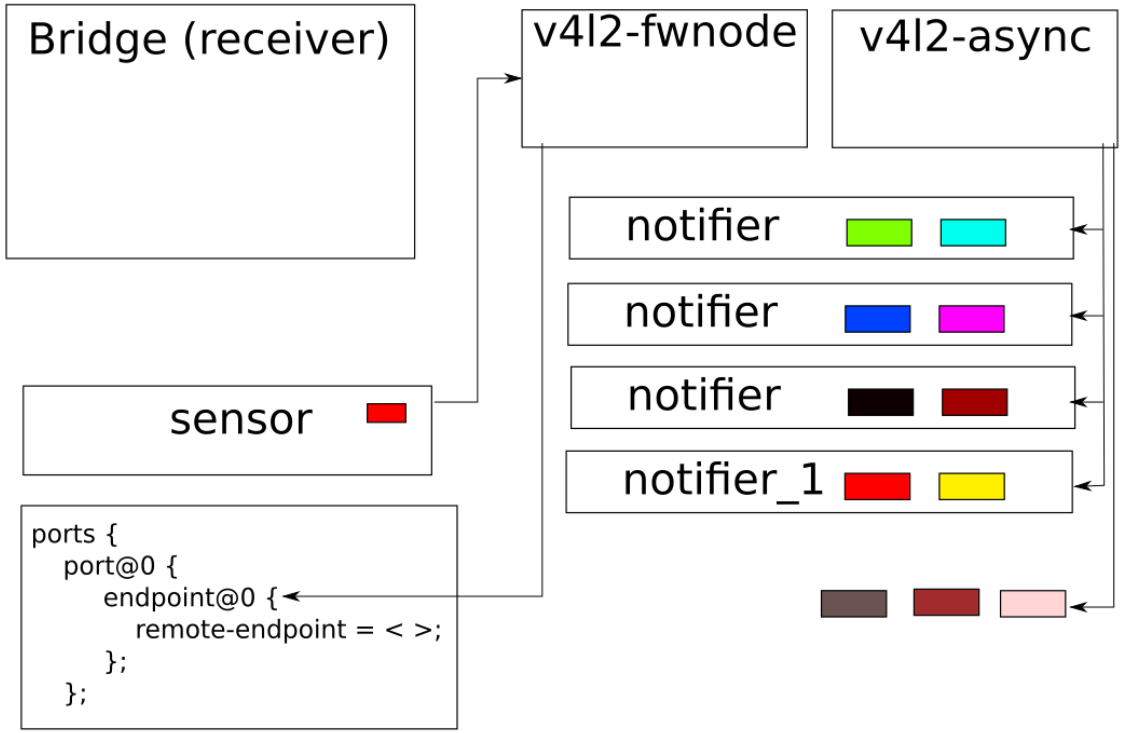
video4linux2: device probing



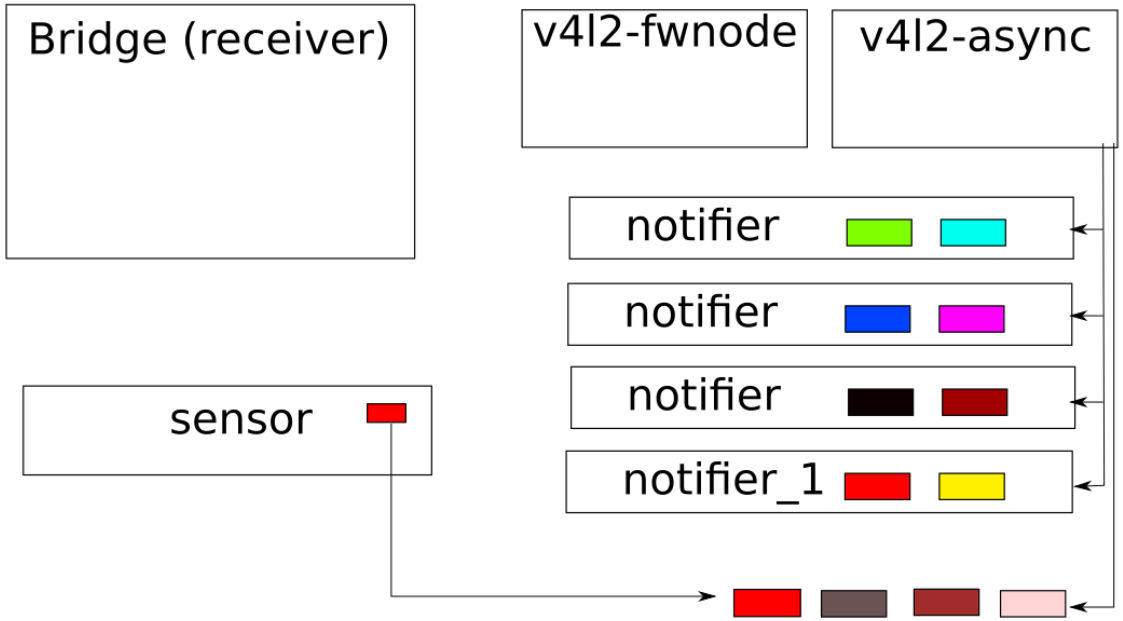
video4linux2: device probing



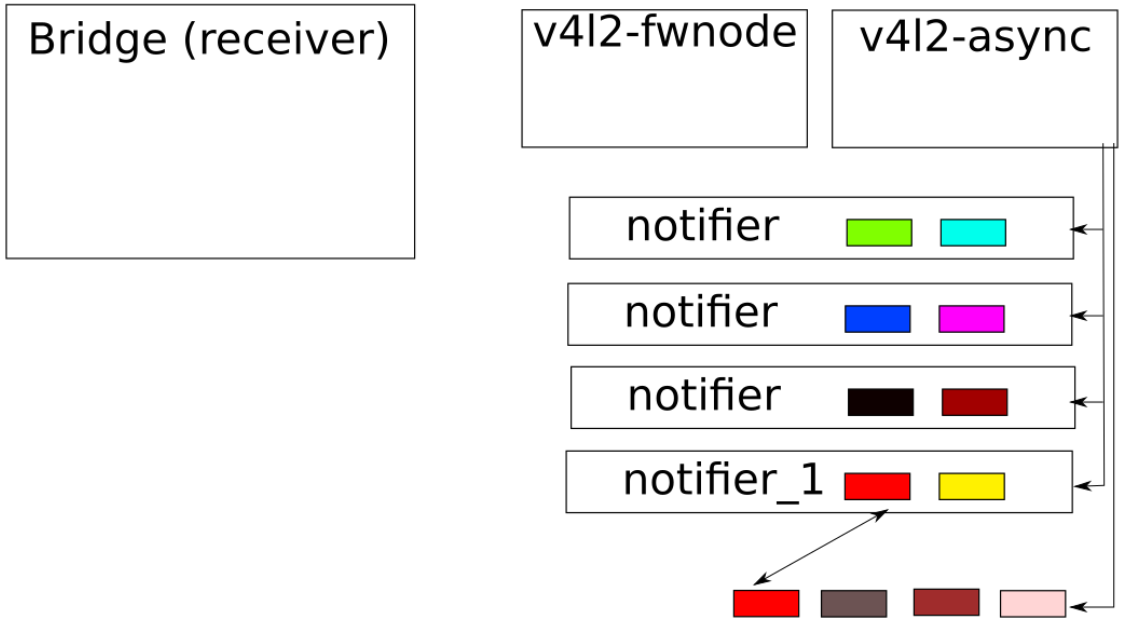
video4linux2: device probing



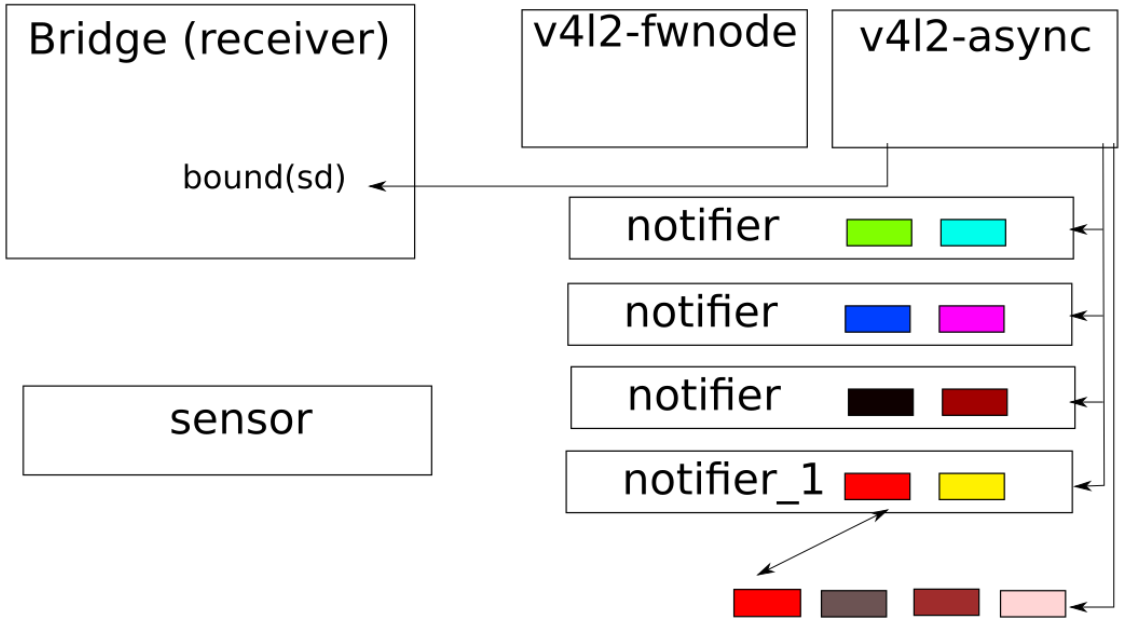
video4linux2: device probing



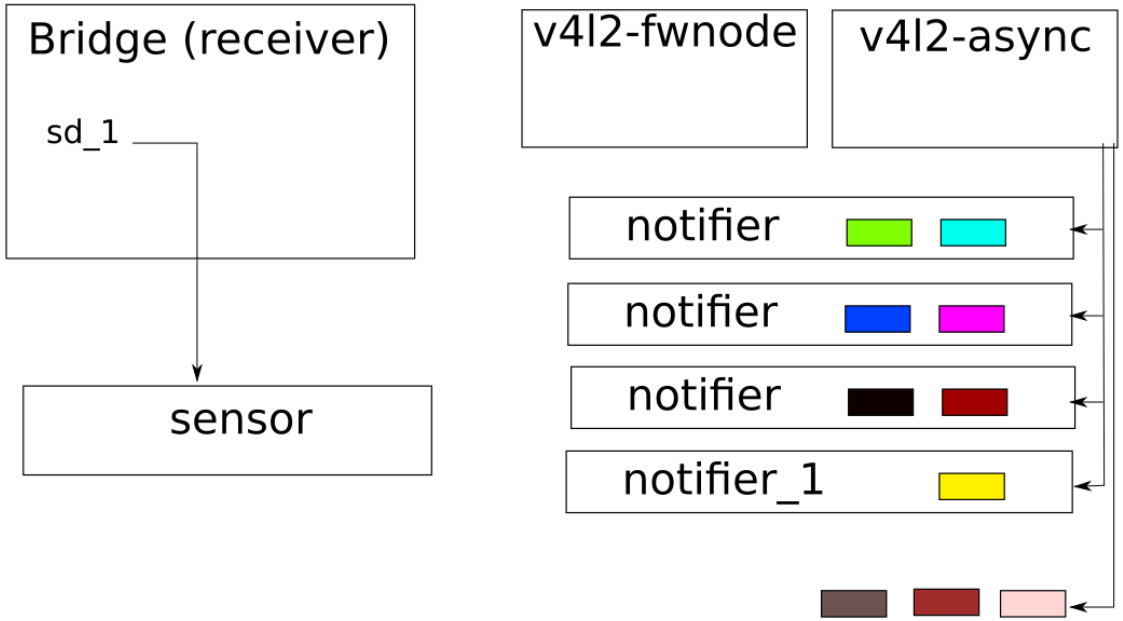
video4linux2: device probing



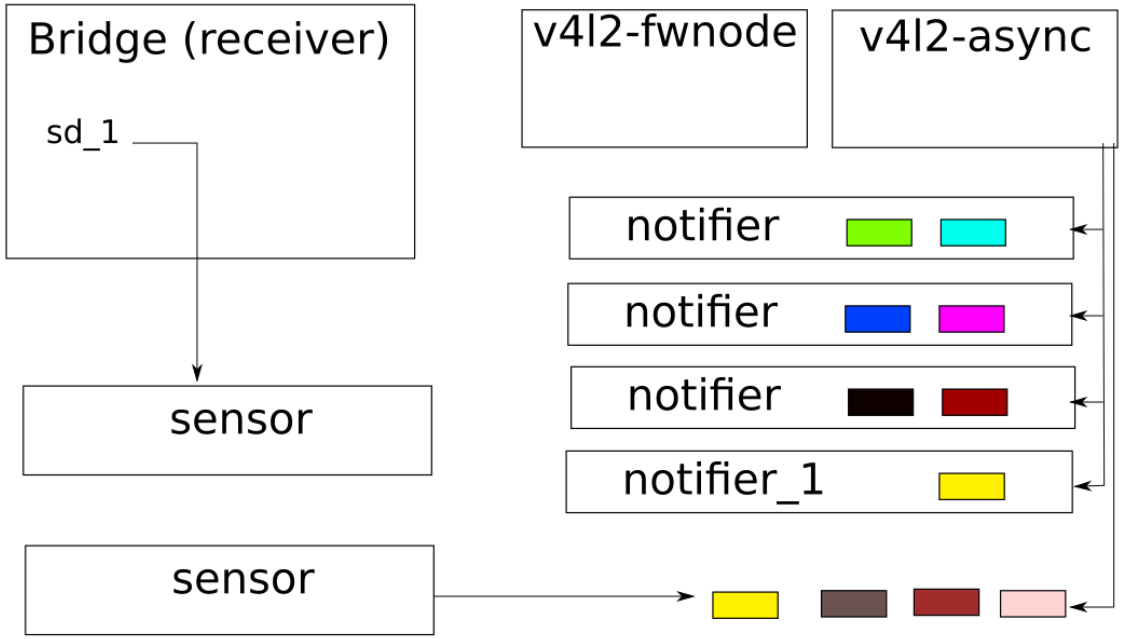
video4linux2: device probing



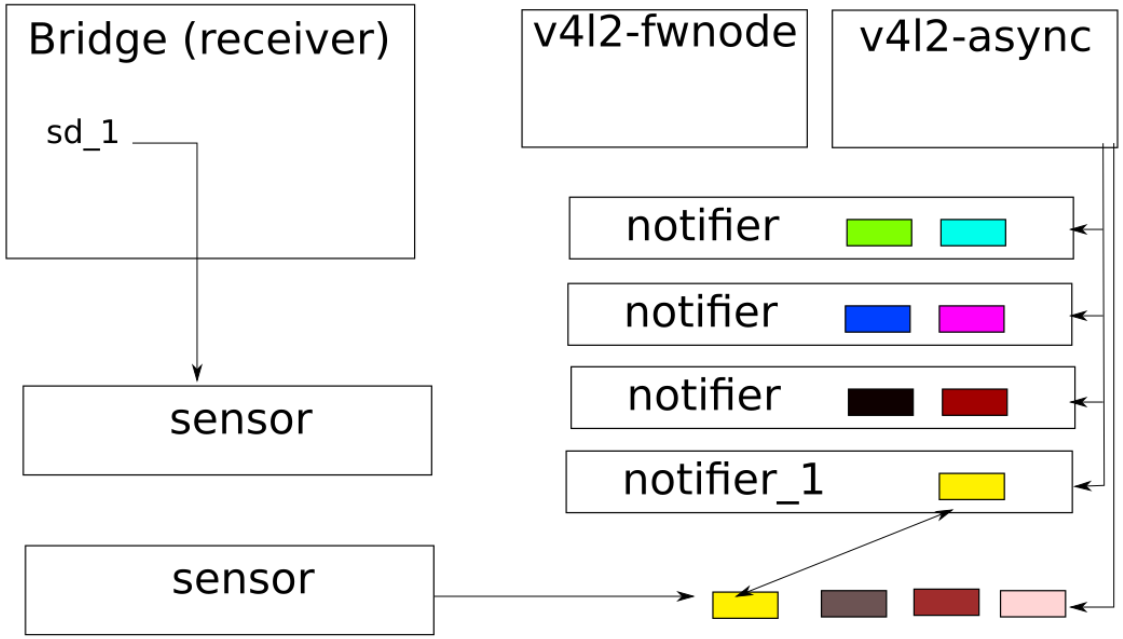
video4linux2: device probing



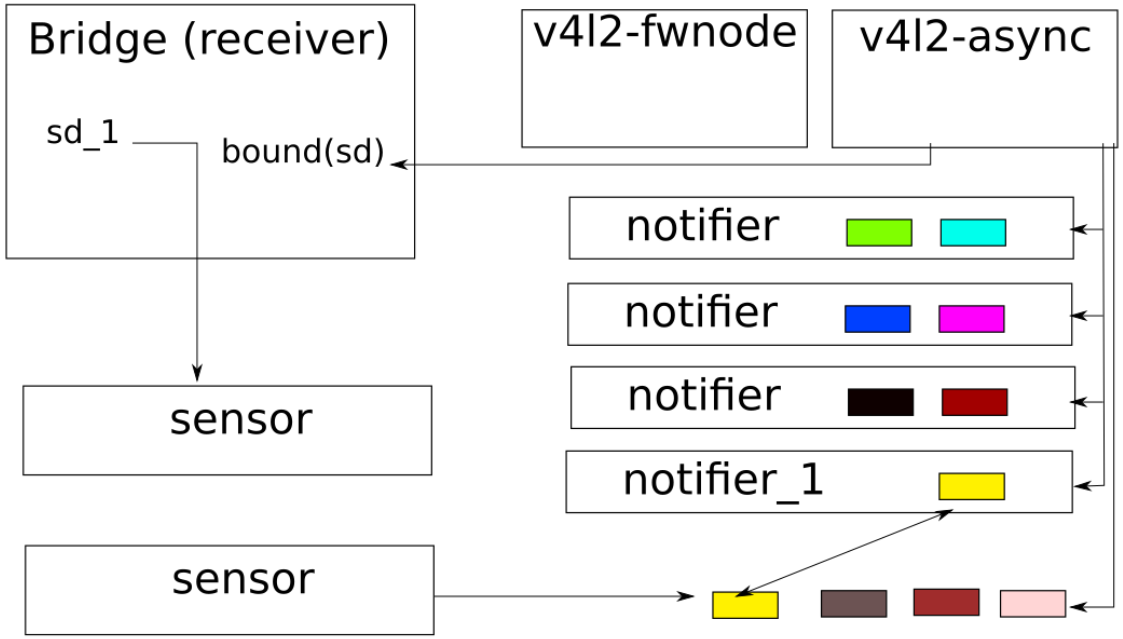
video4linux2: device probing



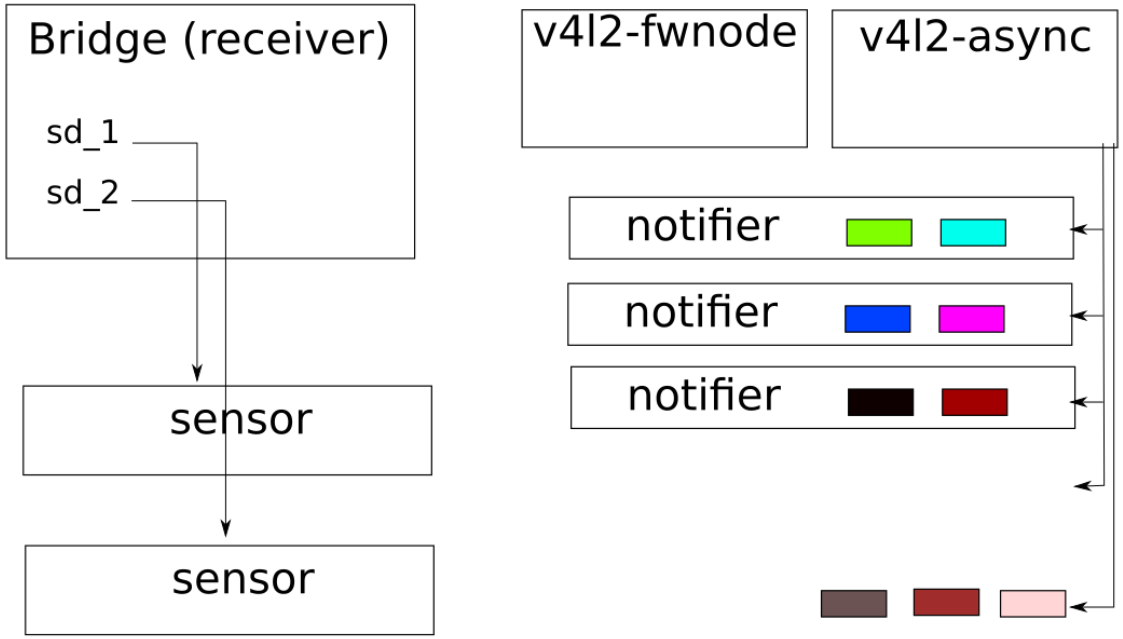
video4linux2: device probing



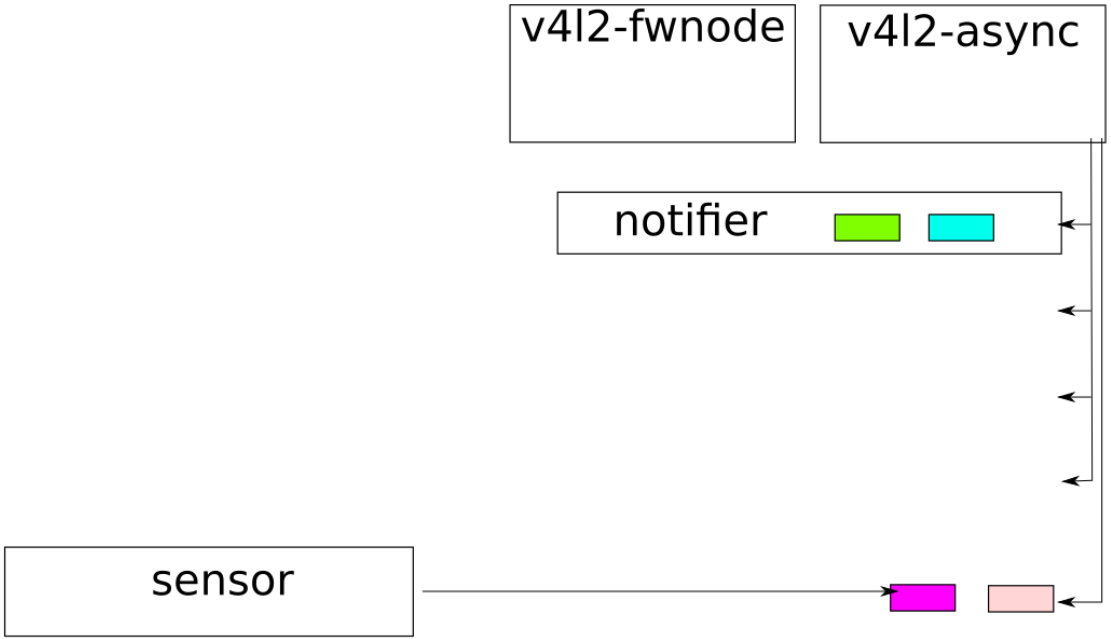
video4linux2: device probing



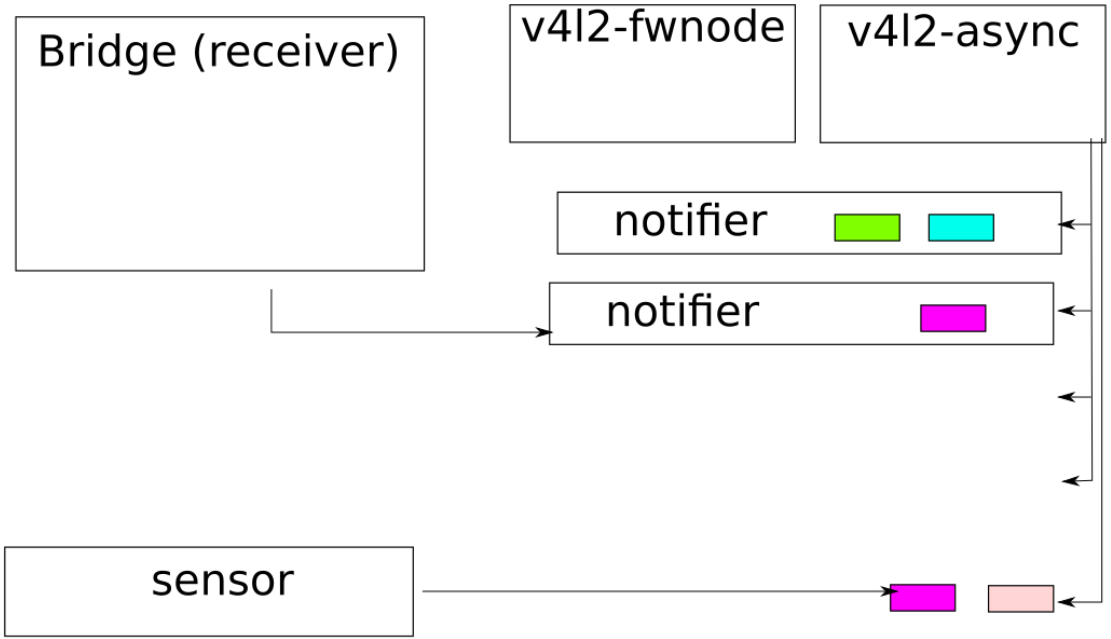
video4linux2: device probing



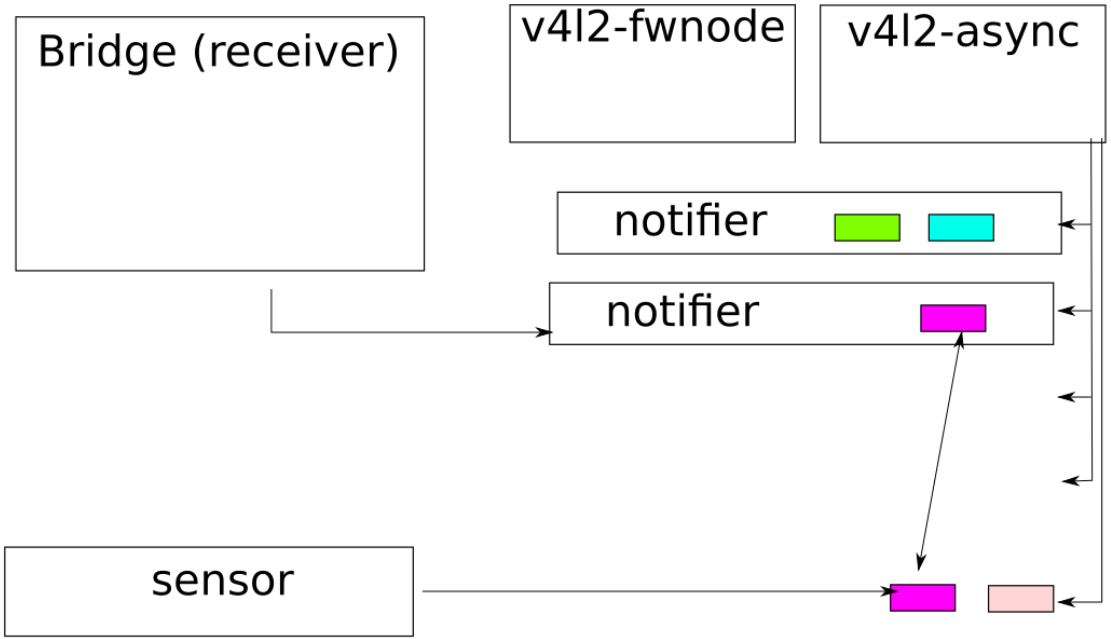
video4linux2: device probing



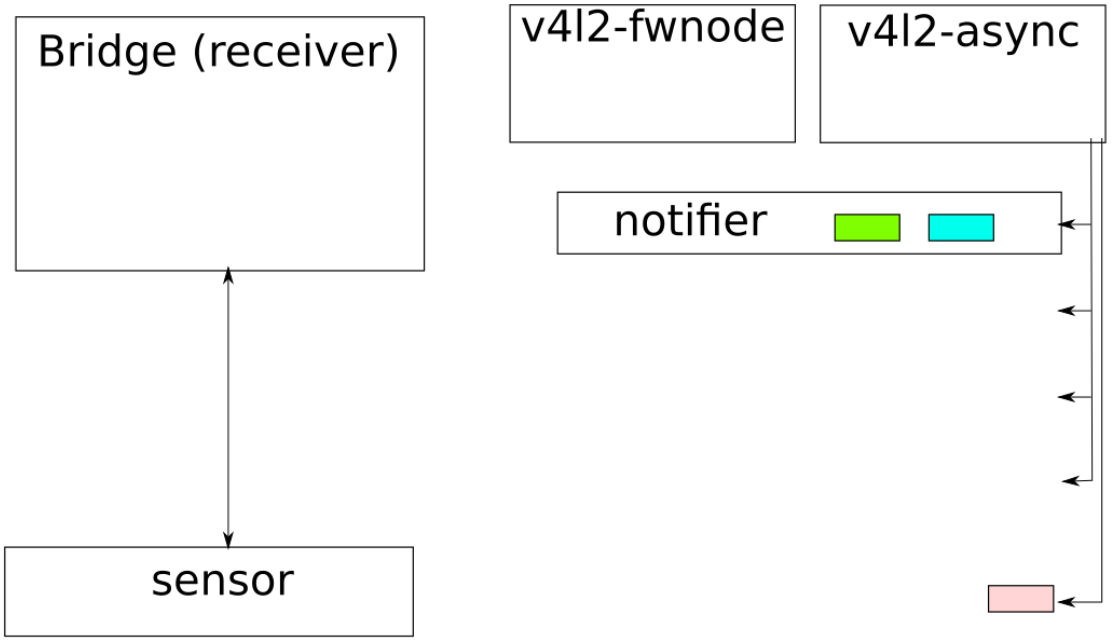
video4linux2: device probing



video4linux2: device probing



video4linux2: device probing



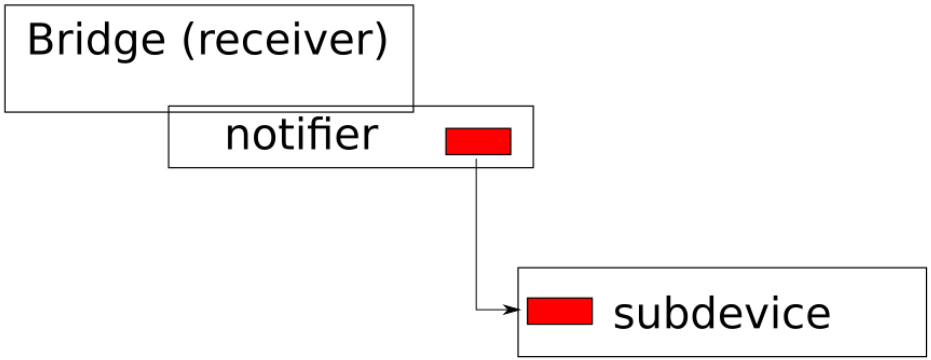
video4linux2: device probing

- soc_camera can do that
  - Uses v4l2-async
  - Uses v4l2-fwnode
- What has changed then?

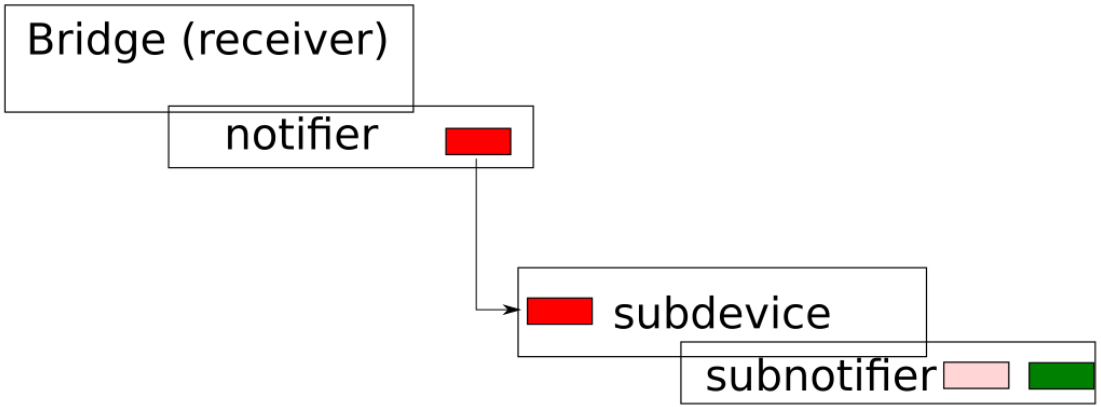
video4linux2: device probing

- soc_camera can do that
  - Uses v4l2-async
  - Uses v4l2-fwnode
- What has changed then?
  - That now subdevs can have notifiers too!
  - Introducing subnotifiers (late 2017)
    Niklas, Sakari Laurent – rcar-csi.c

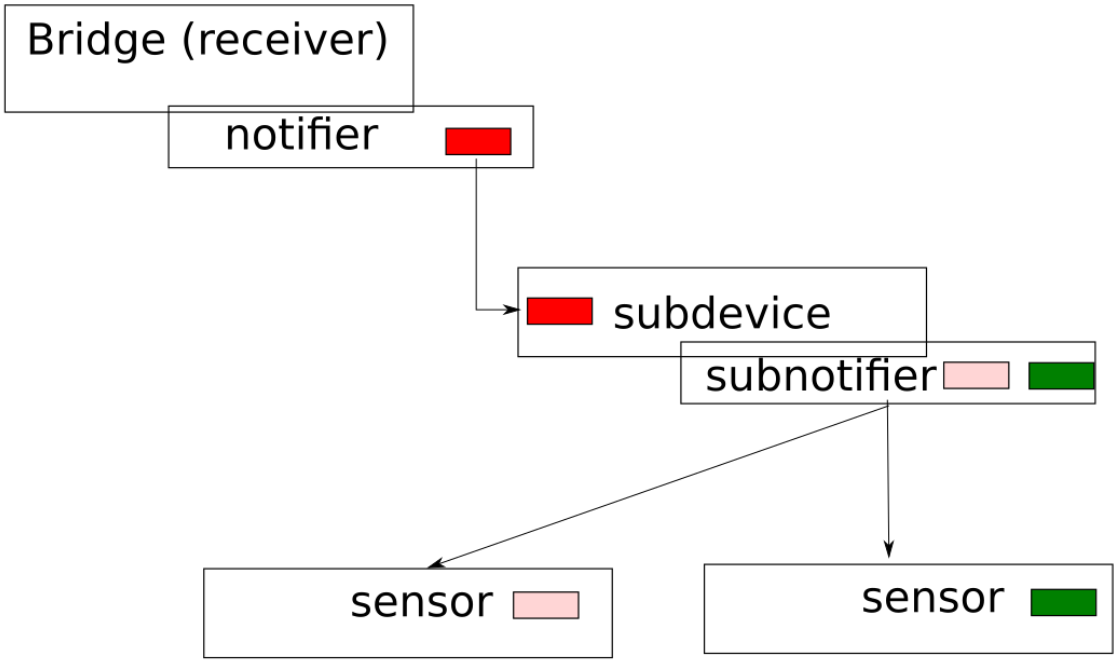
video4linux2: device probing



video4linux2: device probing



video4linux2: device probing



video4linux2: power management

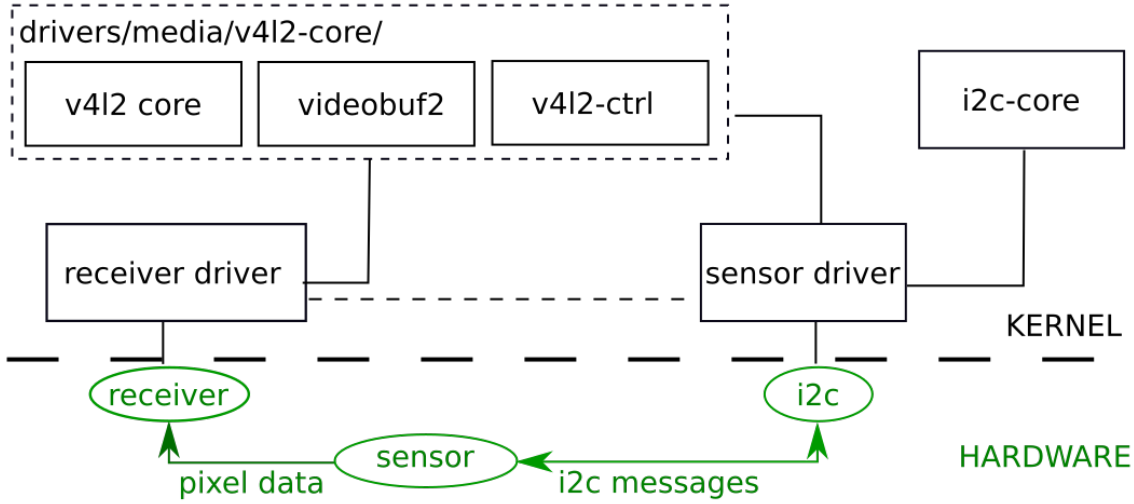
- Non media controller equipped device
  - Single device node abstraction
  - Sequential operations flow

video4linux2: power management

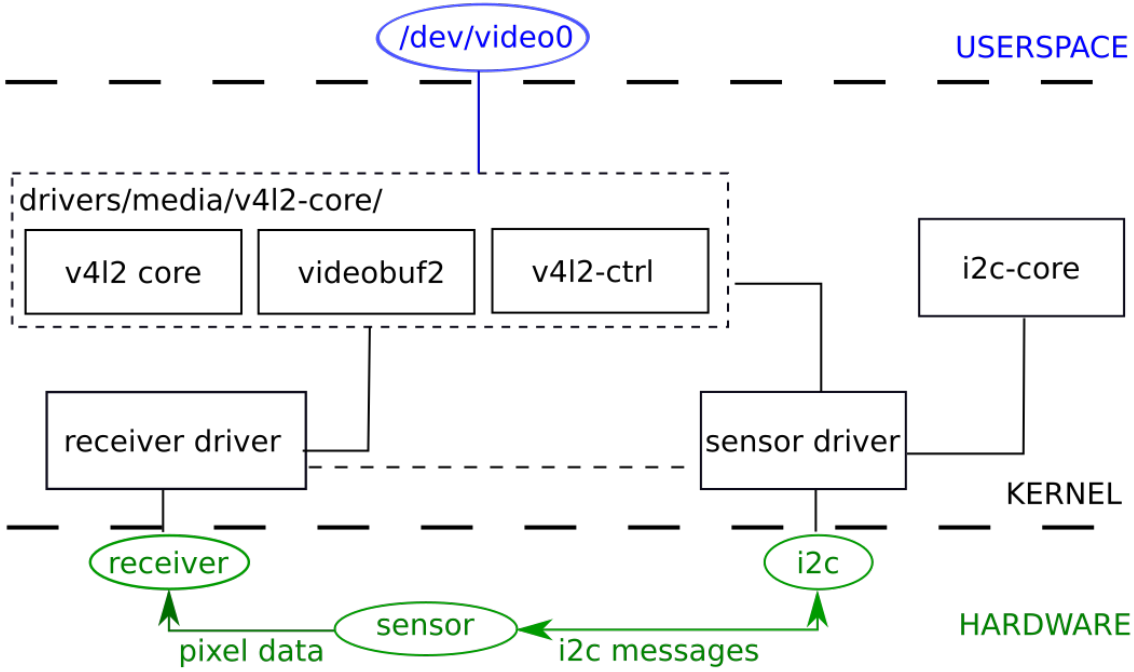
- Non media controller equipped device
  - Single device node abstraction
  - Sequential operations flow
- Media controller equipped device
  - Video device node and subdevice nodes
  - Asynchronous sequence of operations

video4linux2: power management

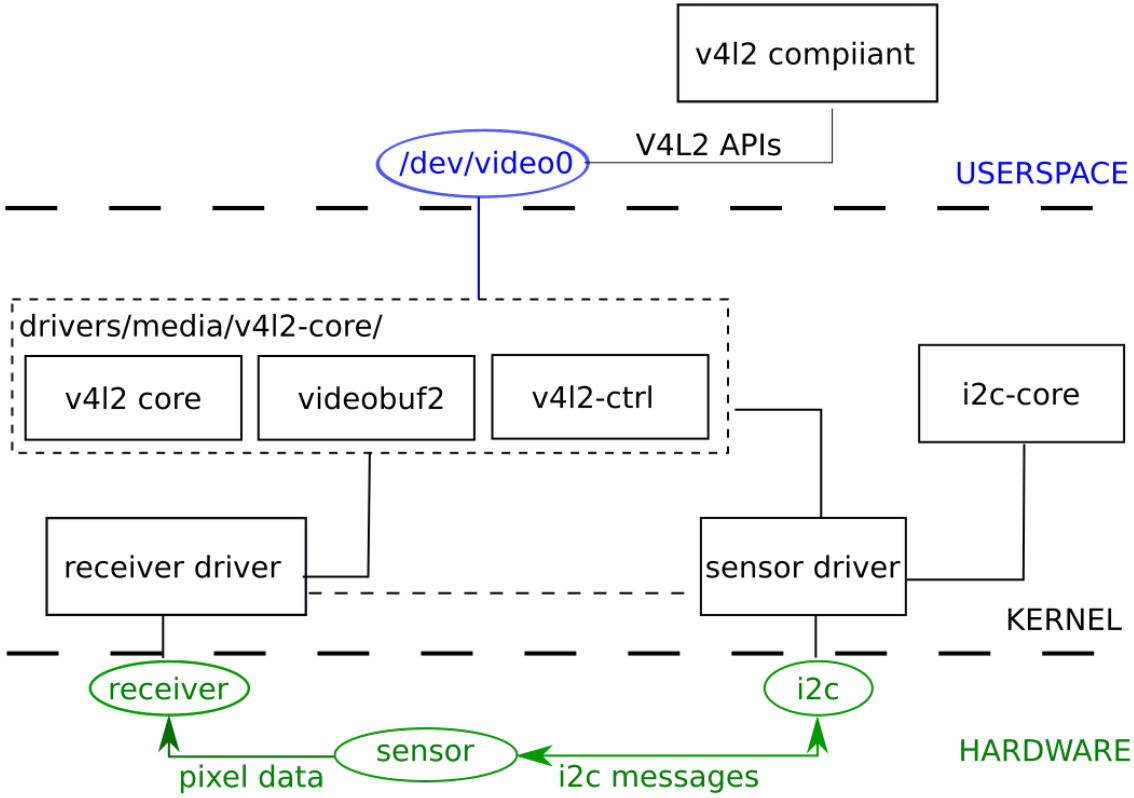
video4linux2: power management



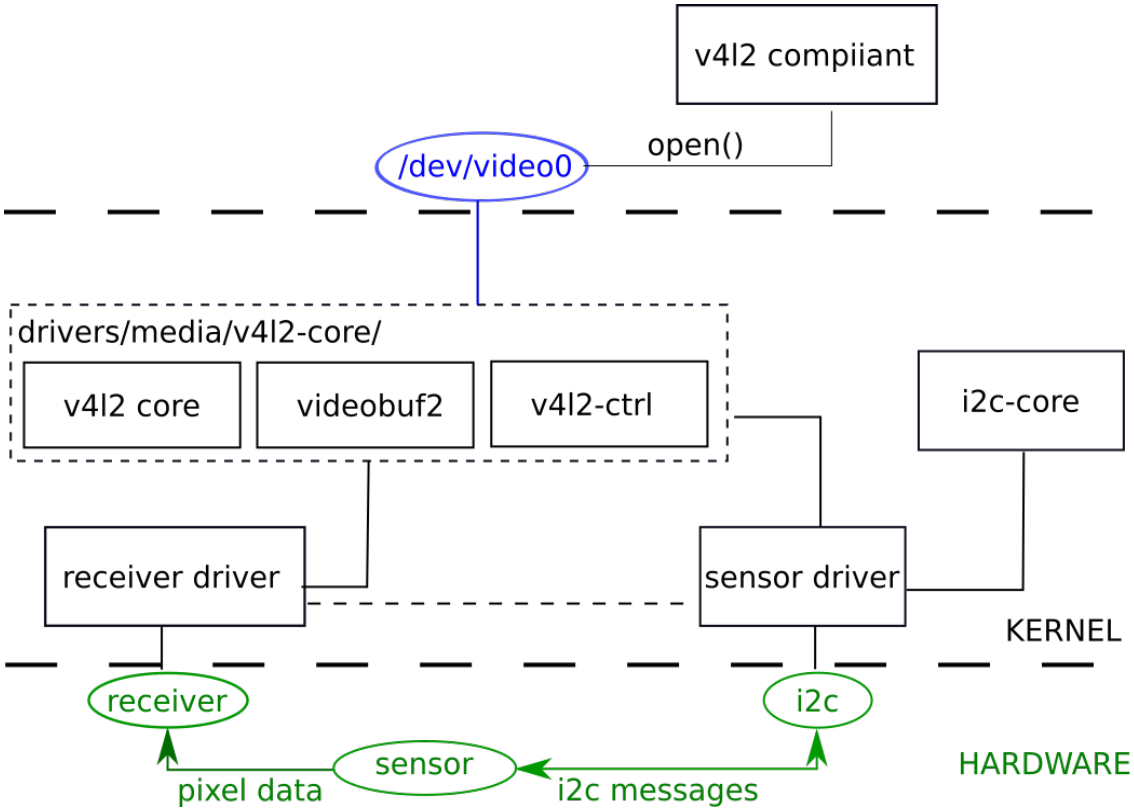
video4linux2: power management



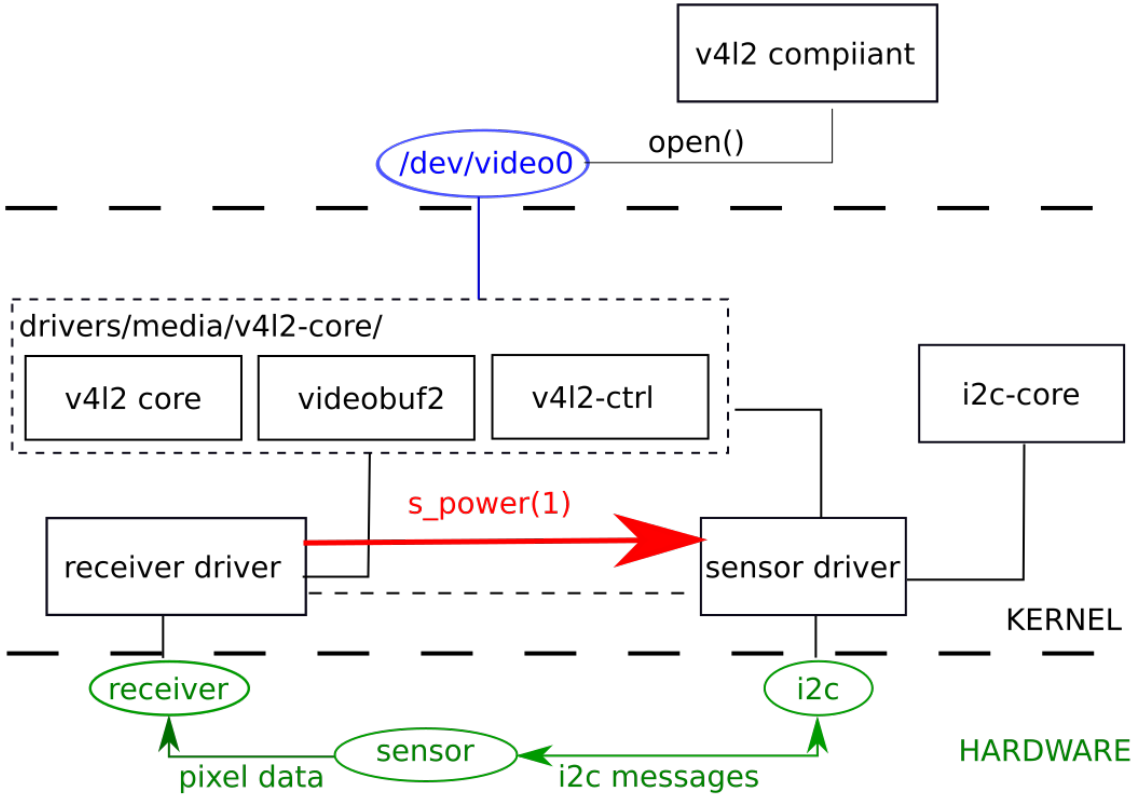
video4linux2: power management



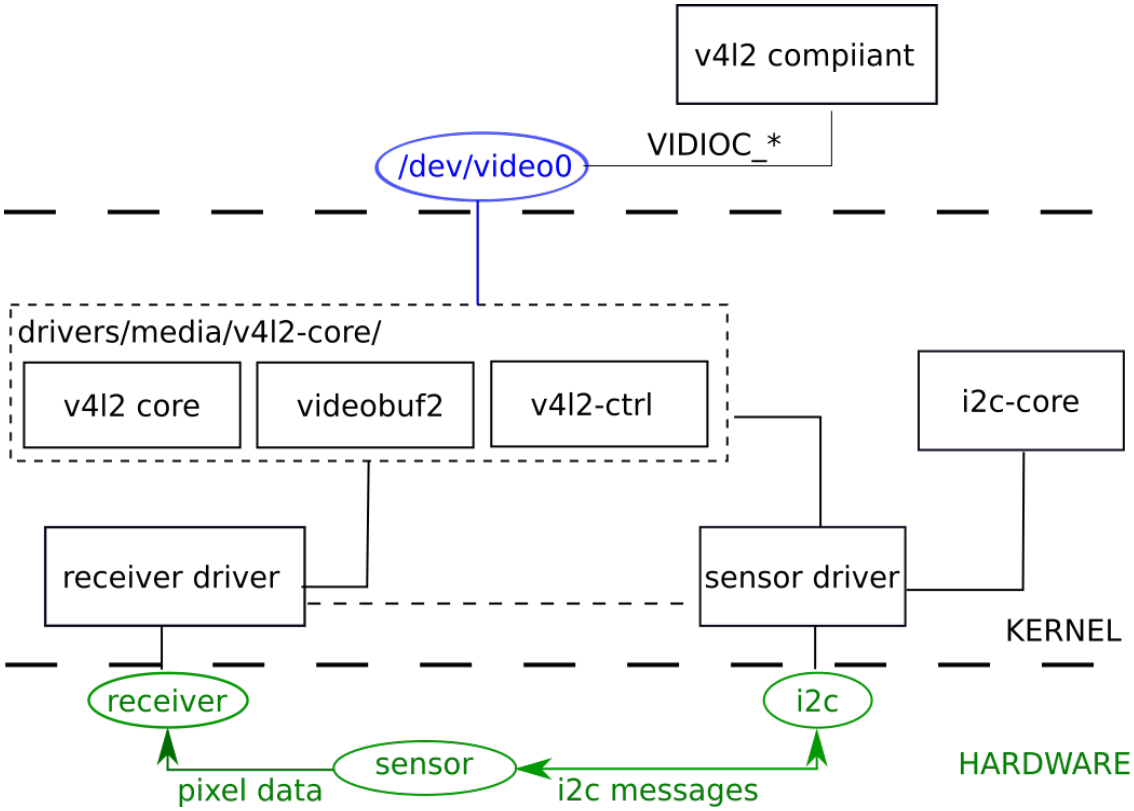
video4linux2: power management



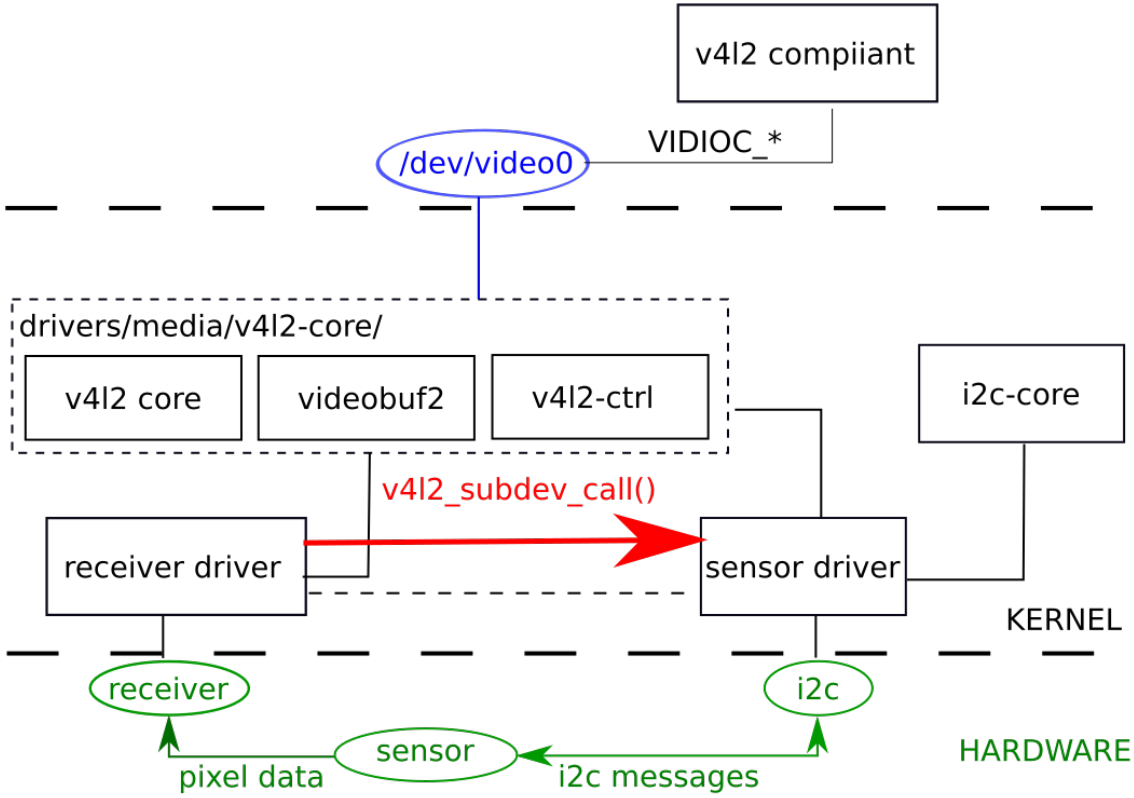
video4linux2: power management



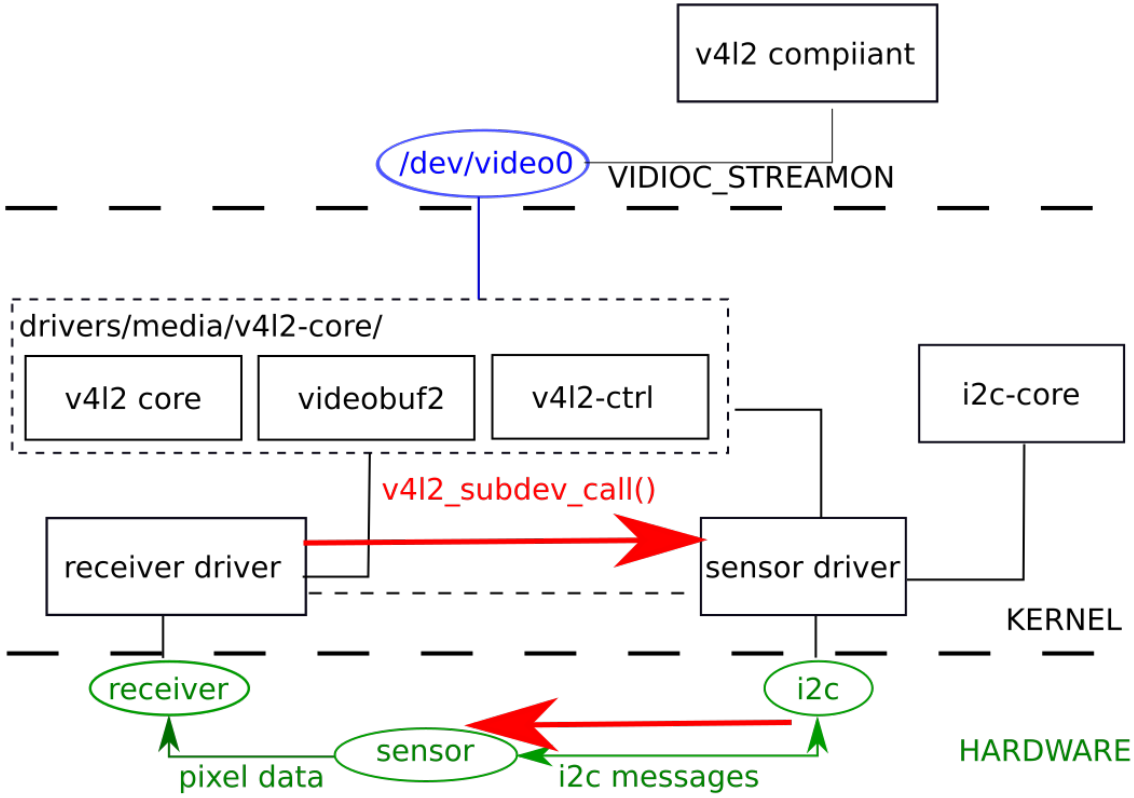
video4linux2: power management



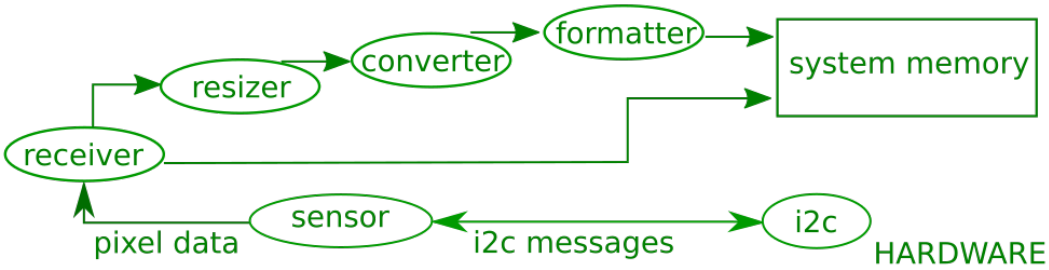
video4linux2: power management



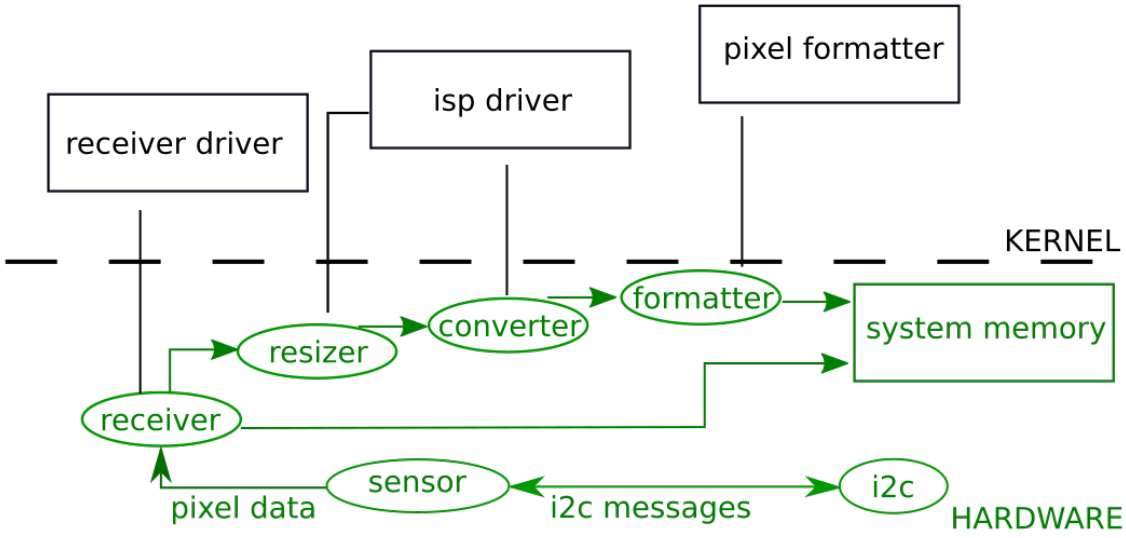
video4linux2: power management



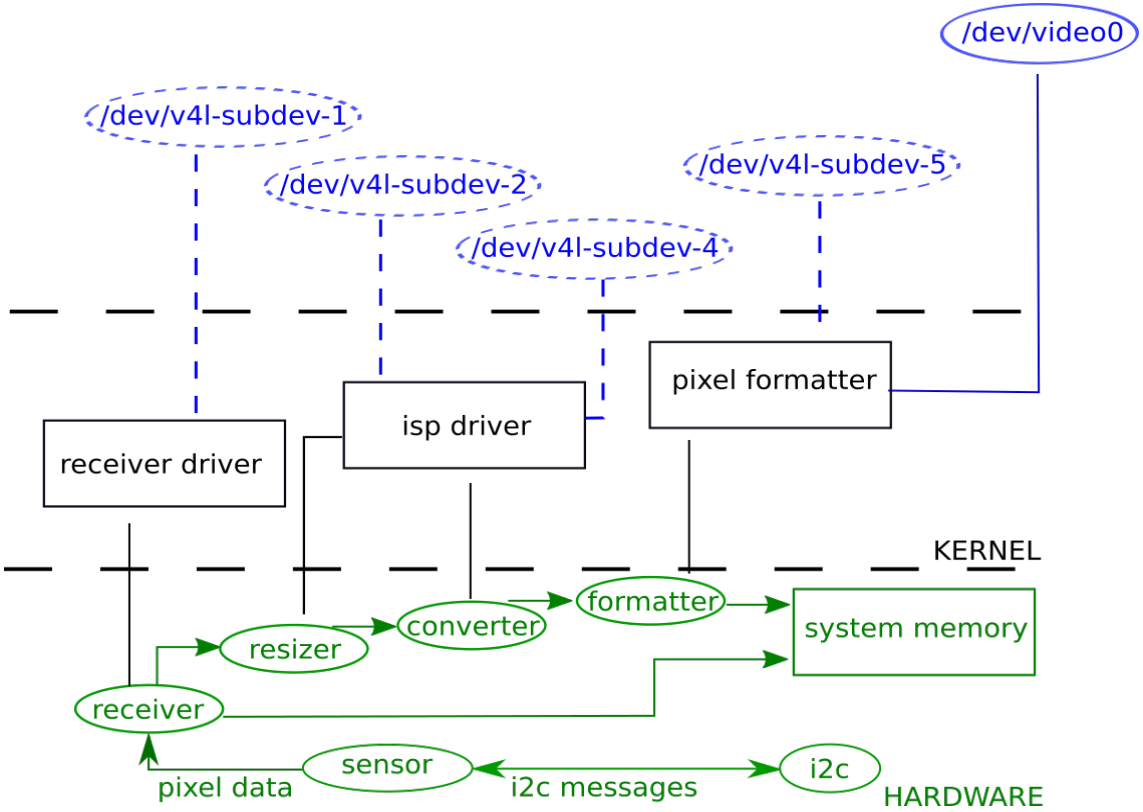
video4linux2: power management



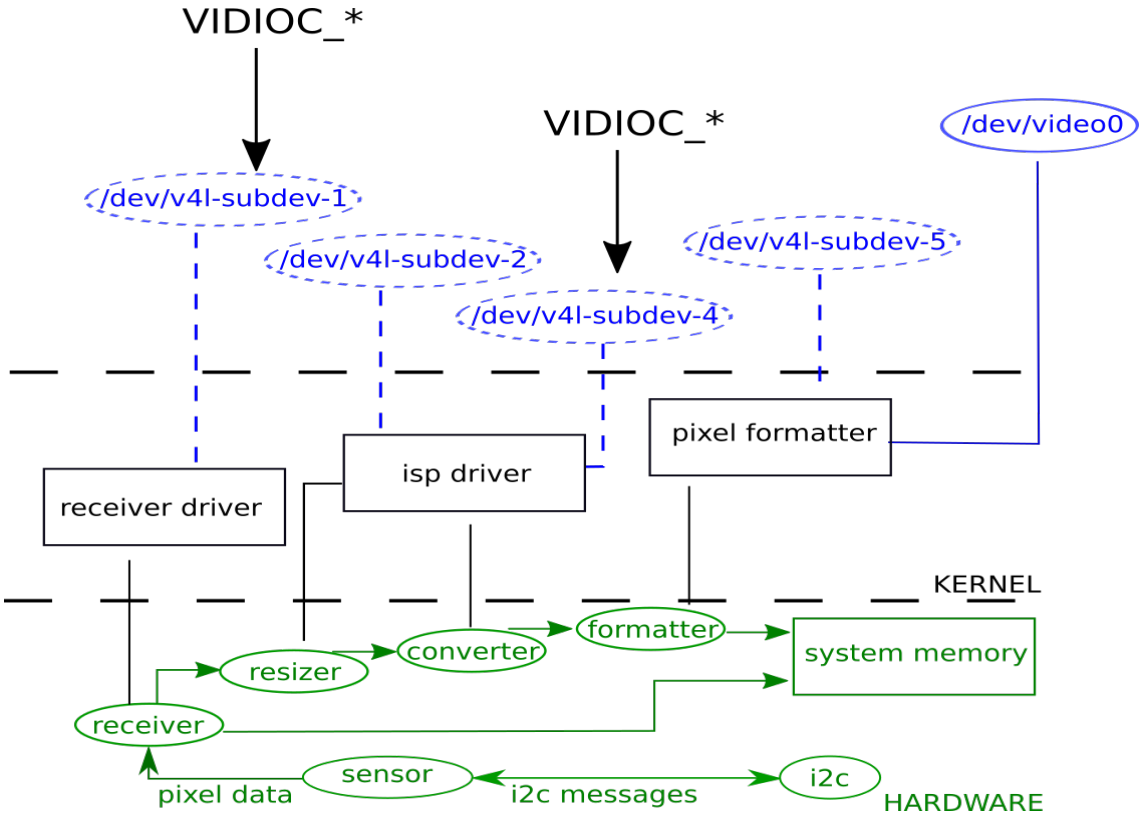
video4linux2: power management



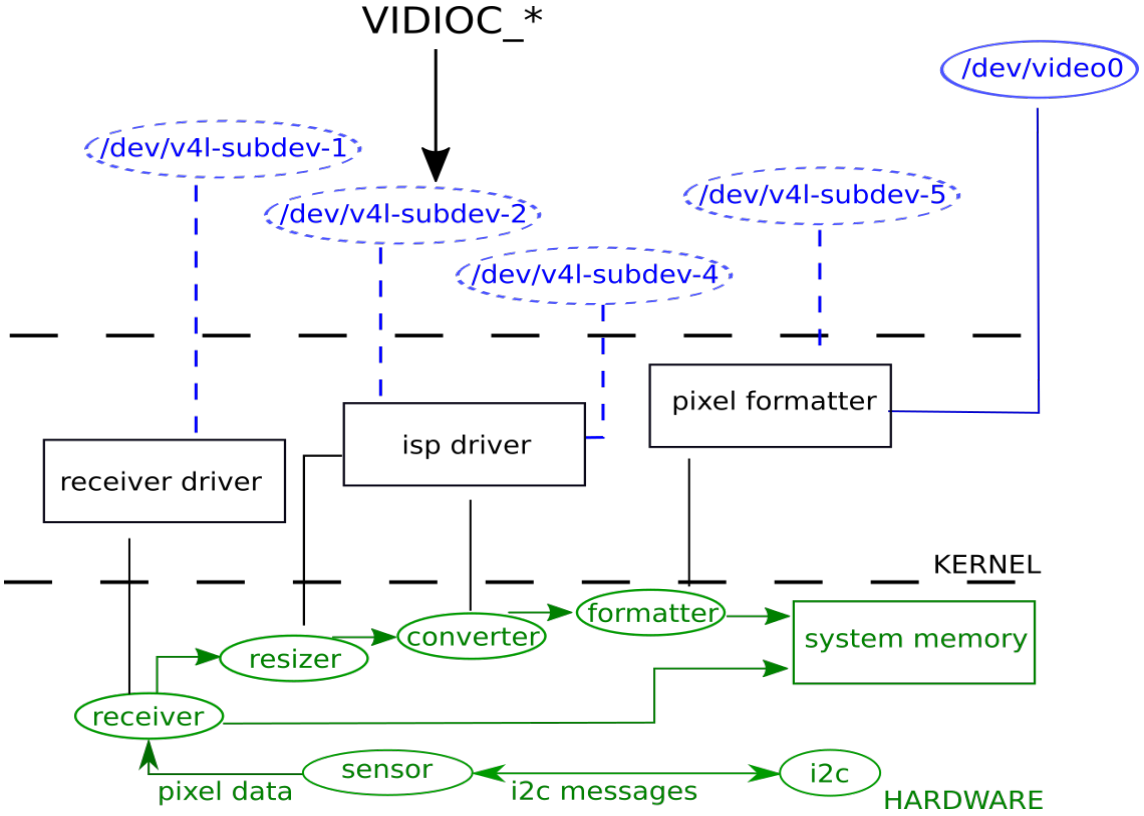
video4linux2: power management



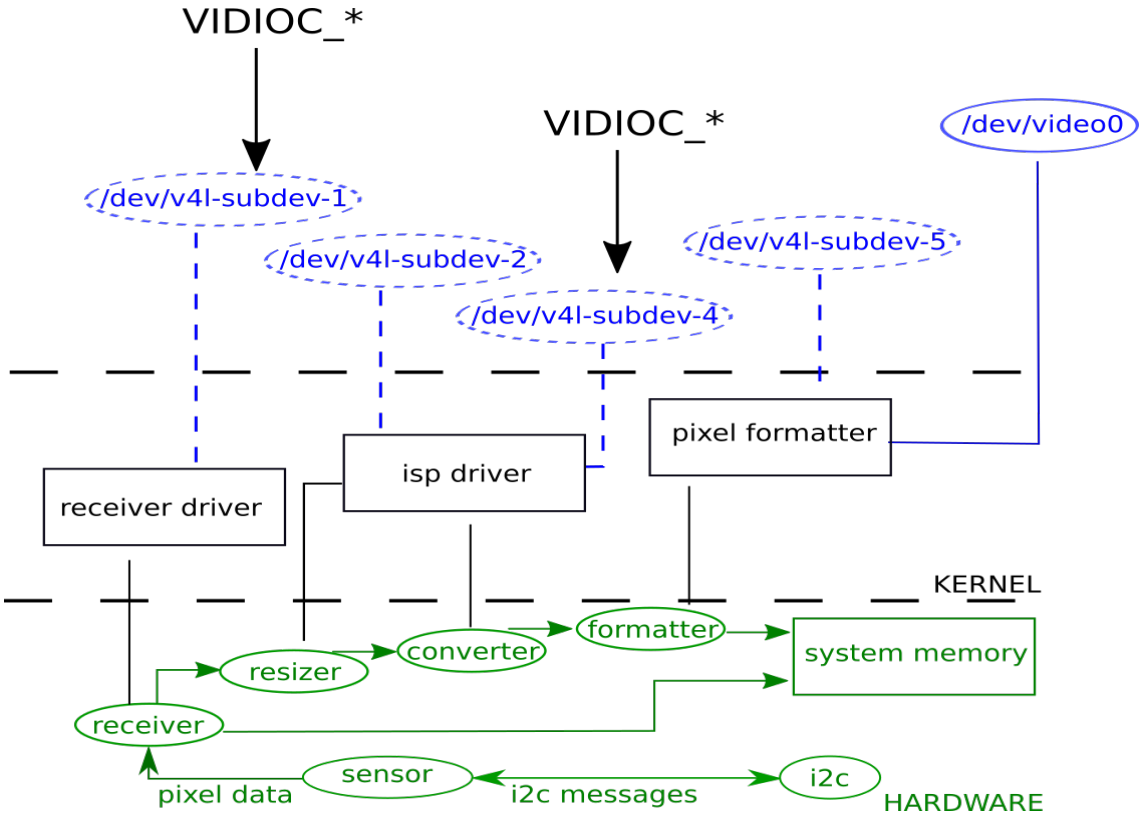
video4linux2: power management



video4linux2: power management



video4linux2: power management



video4linux2: power management

- CACHE

video4linux2: power management

- CACHE YOUR

video4linux2: power management

- CACHE YOUR SETTINGS

video4linux2: power management

- CACHE YOUR SETTINGS
- Maintain a driver-wise ‘power state’ notion

video4linux2: power management

- CACHE YOUR SETTINGS
- Maintain a *ref-counted* driver-wise ‘power state’ notion
- Apply the cached settings at stream-on time

video4linux2: power management

- CACHE YOUR SETTINGS
- Maintain a *ref-counted* driver-wise ‘power state’ notion
- Apply the cached settings at stream-on time
- Use RUNTIME_PM whenever possible

video4linux2: clks, gpios, regulators

- Legacy board files provided to drivers power management routines

```
static struct soc_camera_link ov7725_link = {
          .power           = ov7725_power,
          .board_info      = &migor_i2c_camera[0],
          .i2c_adapter_id = 0,
          .priv            = &ov7725_info,
   };
```

video4linux2: clks, gpios, regulators

- Use gpio, clock, and regulator frameworks
- Collect references from firmaware

```
*_get(dev, “name”);
```

- Have the driver enable/disable them at s_power time

Practical: ov772x.c

- Copy the driver code from *soc_camera*

```
media: i2c: Copy ov772x soc_camera sensor driver

Copy the soc_camera based driver in v4l2 sensor driver directory.
This commit just copies the original file without modifying it.

No modification to KConfig and Makefile as soc_camera framework
dependencies need to be removed first in next commit.

drivers/media/i2c/ov772x.c | 1124 ++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++
1 file changed, 1124 insertions(+)
```

Practical: ov772x.c

- Remove soc_camera dependencies

```
media: i2c: ov772x: Remove soc_camera dependencies

 Remove soc_camera framework dependencies from ov772x sensor
 driver.
 - Handle clock and gpios
 - Register async subdevice
 - Remove soc_camera specific g/s_mbus_config operations
 - Change image format colorspace from JPEG to SRGB as the two use
   the same colorspace information but JPEG makes assumptions on
   color components quantization that do not apply to the sensor
 - Remove sizes crop from get_selection as driver can't scale
 - Add kernel doc to driver interface header file
 - Adjust build system
```

Practical: ov772x.c

- Here is where the fun begins:

```
$ git log –-oneline drivers/media/i2c/ov772x.c

3508949 media: i2c: ov772x: Support frame interval handling
4b610d6 media: ov772x: add media controller support
c2cae89 media: ov772x: support device tree probing
34af7d9 media: ov772x: handle nested s_power() calls
95f5a45 media: ov772x: avoid accessing registers under power saving mode
7b9998c media: ov772x: make set_fmt() and s_frame_interval() return -EBUSY while streaming
795bce4 media: ov772x: create subdevice device node
```

Thank you for the attention

- Questions?

This presentation was made with inkscape and LibreOffice: a big thanks to the developers of these two softwares!

THE LINUX FOUNDATION

Embedded Linux Conference Europe

OpenIoTSummit Europe

THE LINUX FOUNDATION