

# The Virtual Media Controller Driver (vimc)

The vimc driver emulates complex video hardware using the V4L2 API and the Media API. It has a capture device and three subdevices: sensor, debayer and scaler.

## Topology

The topology is hardcoded, although you could modify it in vimc-core and recompile the driver to achieve your own topology. This is the default topology:

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\media\linux-master) (Documentation) (admin-guide) (media) vimc.rst, line 17)
```

Unknown directive type "kernel-figure".

```
.. kernel-figure:: vimc.dot
   :alt: Diagram of the default media pipeline topology
   :align: center

Media pipeline graph on vimc
```

## Configuring the topology

Each subdevice will come with its default configuration (pixelformat, height, width, ...). One needs to configure the topology in order to match the configuration on each linked subdevice to stream frames through the pipeline. If the configuration doesn't match, the stream will fail. The v4l-utils package is a bundle of user-space applications, that comes with media-ctl and v4l2-ctl that can be used to configure the vimc configuration. This sequence of commands fits for the default topology:

```
media-ctl -d platform:vimc -V '"Sensor A":0[fmt:SBGGR8_1X8/640x480] '
media-ctl -d platform:vimc -V '"Debayer A":0[fmt:SBGGR8_1X8/640x480] '
media-ctl -d platform:vimc -V '"Sensor B":0[fmt:SBGGR8_1X8/640x480] '
media-ctl -d platform:vimc -V '"Debayer B":0[fmt:SBGGR8_1X8/640x480] '
v4l2-ctl -z platform:vimc -d "RGB/YUV Capture" -v width=1920,height=1440
v4l2-ctl -z platform:vimc -d "Raw Capture 0" -v pixelformat=BA81
v4l2-ctl -z platform:vimc -d "Raw Capture 1" -v pixelformat=BA81
```

## Subdevices

Subdevices define the behavior of an entity in the topology. Depending on the subdevice, the entity can have multiple pads of type source or sink.

vimc-sensor:

Generates images in several formats using video test pattern generator. Exposes:

- 1 Pad source

vimc-debayer:

Transforms images in bayer format into a non-bayer format. Exposes:

- 1 Pad sink
- 1 Pad source

vimc-scaler:

Re-size the image to meet the source pad resolution. E.g.: if the sink pad is configured to 360x480 and the source to 1280x720, the image will be stretched to fit the sink resolution. Works for any resolution within the vimc limitations (even shrinking the image if necessary). Exposes:

- 1 Pad sink
- 1 Pad source

vimc-capture:

Exposes node /dev/videoX to allow userspace to capture the stream. Exposes:

- 1 Pad sink
- 1 Pad source

## Module options

Vimc has a module parameter to configure the driver.

- allocator=<unsigned int>

memory allocator selection, default is 0. It specifies the way buffers will be allocated.

- 0: vmaalloc
- 1: dma-contig