

Pixel data transmitter and receiver drivers

V4L2 supports various devices that transmit and receive pixel data. Examples of these devices include a camera sensor, a TV tuner and a parallel or a CSI-2 receiver in an SoC.

Bus types

The following busses are the most common. This section discusses these two only.

MIPI CSI-2

CSI-2 is a data bus intended for transferring images from cameras to the host SoC. It is defined by the [MIPI alliance](#).

Parallel

[BT.601](#) and [BT.656](#) are the most common parallel busses.

Transmitter drivers

Transmitter drivers generally need to provide the receiver drivers with the configuration of the transmitter. What is required depends on the type of the bus. These are common for both busses.

Media bus pixel code

See [ref`v4l2-mbus-pixelcode`](#).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\media\linux-master) (Documentation) (driver-api) (media) tx-rx.rst, line 43); [backlink](#)

Unknown interpreted text role "ref".

Link frequency

The [ref`V4L2_CID_LINK_FREQ`](#) control is used to tell the receiver the frequency of the bus (i.e. it is not the same as the symbol rate).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\media\linux-master) (Documentation) (driver-api) (media) tx-rx.rst, line 48); [backlink](#)

Unknown interpreted text role "ref".

`.s_stream()` callback

The struct `struct v4l2_subdev_video_ops->s_stream()` callback is used by the receiver driver to control the transmitter driver's streaming state.

CSI-2 transmitter drivers

Pixel rate

The pixel rate on the bus is calculated as follows:

$$\text{pixel_rate} = \text{link_freq} * 2 * \text{nr_of_lanes} * 16 / k / \text{bits_per_sample}$$

where

variables in pixel rate calculation

variable or constant	description
<code>link_freq</code>	The value of the <code>V4L2_CID_LINK_FREQ</code> integer64 menu item.
<code>nr_of_lanes</code>	Number of data lanes used on the CSI-2 link. This can be obtained from the OF endpoint configuration.
<code>2</code>	Data is transferred on both rising and falling edge of the signal.
<code>bits_per_sample</code>	Number of bits per sample.
<code>k</code>	16 for D-PHY and 7 for C-PHY

Note

The pixel rate calculated this way is **not** the same thing as the pixel rate on the camera sensor's pixel array which is indicated by the `ref: V4L2_CID_PIXEL_RATE <v4l2-cid-pixel-rate>` control.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\media\linux-master) (Documentation) (driver-api) (media) tx-rx.rst, line 89); [backlink](#)

Unknown interpreted text role "ref".

LP-11 and LP-111 modes

As part of transitioning to high speed mode, a CSI-2 transmitter typically briefly sets the bus to LP-11 or LP-111 state, depending on the PHY. This period may be as short as 100 μ s, during which the receiver observes this state and proceeds its own part of high speed mode transition.

Most receivers are capable of autonomously handling this once the software has configured them to do so, but there are receivers which require software involvement in observing LP-11 or LP-111 state. 100 μ s is a brief period to hit in software, especially when there is no interrupt telling something is happening.

One way to address this is to configure the transmitter side explicitly to LP-11 or LP-111 mode, which requires support from the transmitter hardware. This is not universally available. Many devices return to this state once streaming is stopped while the state after power-on is LP-00 or LP-000.

The `.pre_streamon()` callback may be used to prepare a transmitter for transitioning to streaming state, but not yet start streaming. Similarly, the `.post_streamoff()` callback is used to undo what was done by the `.pre_streamon()` callback. The caller of `.pre_streamon()` is thus required to call `.post_streamoff()` for each successful call of `.pre_streamon()`.

In the context of CSI-2, the `.pre_streamon()` callback is used to transition the transmitter to the LP-11 or LP-111 mode. This also requires powering on the device, so this should be only done when it is needed.

Receiver drivers that do not need explicit LP-11 or LP-111 mode setup are waived from calling the two callbacks.

Stopping the transmitter

A transmitter stops sending the stream of images as a result of calling the `.s_stream()` callback. Some transmitters may stop the stream at a frame boundary whereas others stop immediately, effectively leaving the current frame unfinished. The receiver driver should not make assumptions either way, but function properly in both cases.