Sub-device configuration models

Sakari Ailus <sakari.ailus@linux.intel.com>

2025-05-13

Why Sub-device Configuration Models?

- V4L2 API is standardised on IOCTL level but there is variance across supported hardware
  - Variance in how V4L2 API is used to control hardware features
  - Partly also historical reasons
- The objective is to provide **unified API semantics and behaviour**

Common Raw Sensor Model

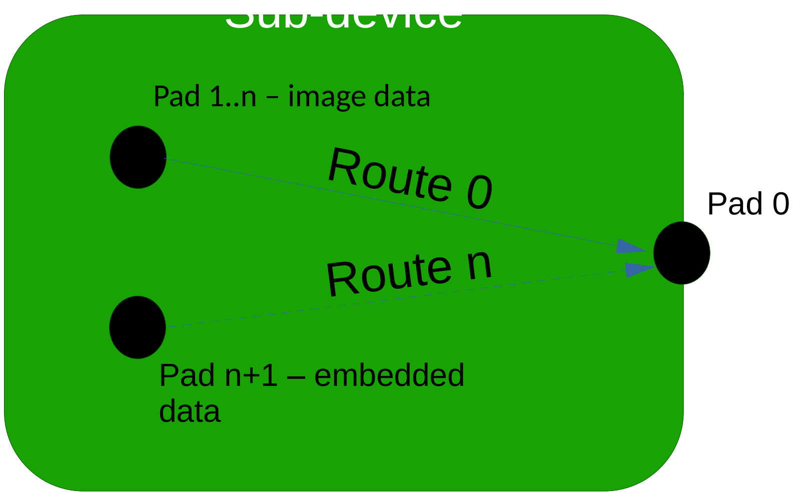
- Intended to be fit for at least 95 % of raw sensors
- One or more image data streams
  - HDR
- One embedded data stream
- Further streams are possible, with device specific documentation
  - Enables implementing device specific functions while remaining compliant with the model

Sensor internal pixel pipeline

- Pixel array
- Analogue crop
- Binning and sub-sampling
- Digital crop
- Scaling
- Digital crop

Common Raw Sensor Model Pads

- 0: source pad
- 1..n: image data internal pad
- n+1: embedded data internal pad (optional)



Sensor Image Data Path in Common Raw Sensor Model (*)

Pad/ Stream	Format/Selection Target	Synopsis
1/0	Format	Native image data format
1/0	Crop	Analogue crop
1/0	Compose	Binning and sub-sampling
0/0	Crop	Digital crop
0/0	Compose	Scaling
0/0	Format	Source format

*: Assuming n == 1

Sensor Embedded Data in Common Raw Sensor Model

Pad/ Stream	Format/Selection Target	Synopsis
2/0	Format	Native embedded data format
0/1	Format	Generic metadata format

Discussion

- Frame time configuration
  - Common Raw Sensor model offers reliable access to analogue crop rectangle
  - Previously this was spotty (e.g. CCS)
- Mode based sensor drivers
  - While Common Raw Sensor model exposes more information to the user space, it as such does not enable mode selection on mode-list based drivers, which have a hard-coded list of pre-set ”mode” configurations

Discussion

- Mbus formats
  - Greyscale formats vs. explicitly generic format?
  - Should we use generic formats also on internal pads, or controls to express pixel order?