V4L2 Status Update

Hans Verkuil
Cisco Systems Norway

Colorspaces

Colorspaces Updates

- Deprecate V4L2_YCBCR_ENC_SYCC: really just the same as V4L2_YCBCR_ENC_601 after careful study of the standards.
- Add support to colorspace conversion hardware. Needs support to tell a capture device what colorspace the application expects.
- New v4l2_pix_format(_mplane) flag for that: V4L2_PIX_FMT_FLAG_HAS_CSC. If set, then the driver will use the S_FMT colorspace, ycbcr_enc and quantization fields to try and convert what it receives to what the application expects.
- Needs core support to set up any CSC matrices/vectors.
- Wrote code to do the calculations to go from any format to any format as long as both use the same V4L2_COLORSPACE value.

Colorspaces Updates

- Complex to decide what to send out on an HDMI transmitter, and what is received on an HDMI receiver.
- Creating helper functions that can make that decision. Too complex to expect drivers to figure that out themselves.
- Still work in progress, I need a fully functioning HDMI-to-HDMI loop before I can clean everything up and prepare patches.
- Initial implementation will be for adv7604 and adv7511 (V4L2 driver) with adv7842 to follow soon afterwards.
- CSC calculations should probably become part of lib since this can be shared with DRM.

Duplicate subdev ops

Duplicate subdev ops

- Several v4l2_subdev ops are duplicated as both video and pad ops. Bad, bad idea. Bridge drivers won't know which one to use, so this prevent reusability.
- Problem: pad ops could not be used in bridge drivers since some expect a struct v4l2_fh, which is not available in a bridge driver. The v4l2_fh struct is used to store the 'TRY' configuration.
- Solved by replacing v4l2_fh by struct v4l2_subdev_pad_config which contains the TRY configuration.
- Clarify that setting the ACTIVE configuration does not use struct v4l2_subdev_pad_config, so drivers can leave that to NULL.
- The enum_framesizes/intervals only supported the TRY configuration, extend that by also supporting the ACTIVE configuration. This is what bridge drivers need and want.
- The duplicate enum_framesizes/intervals video ops have been removed.

Duplicate subdev ops

- Still to do: remove duplicate video cropping ops.
- Patch is available, but I have problems getting my renesas soc-camera based board to work in order to test.
- Remove duplicate video enum/g/try/s_mbus_fmt ops. Initial conversion done, but needs cleanup and double checking. Missing here is support to get TRY format since g_mbus_fmt only supports getting the ACTIVE format. I might address that in a separate patch.
- This work will also go part of the way to make soc-camera more generic. The goal it to let soc-camera work with any sensor instead of requiring explicit soc-camera sensor support.
- Lesson learned: never, ever allow for duplicate ops again.

Compliance Testing & vivid Improvements

v4l2-compliance Improvements

- New man page!
- The driver state is restored after doing the tests.
- Can test streaming for all formats/sizes/intervals and the main crop/compose combinations.
- Can do automated tests to check for correct color format implementation using red, green and blue inputs and streaming with every supported format.
- To do: support this for output as well by sending red, green or blue output for every supported non-compressed format.
- To do: if the V4L2_PIX_FMT_FLAG_HAS_CSC flag is implemented, then test this for various colorspace combinations as well.
- To do: test correct implementation of the selection flags.
- To do (?): add tests for the only two untested ioctls: VIDIOC_S_FBUF and VIDIOC_OVERLAY.

vivid Improvements

- Added YUV 4:2:0 format support.
- Many new formats are now supported by the test pattern generator.
- Upcoming: support for CVT/GTF formats.
- To do: further improve colorspace handling.
- To do: double check 4:2:0 subsampling used by the TPG.

Request API

Request API

- Prepare for Android CameraHAL v3.
- Per-buffer configuration.
- Also combining multiple buffers + their configuration in one request.
- Internally implemented as an extension of the control framework.

Request API: Public API Changes

- struct v4l2_buffer gets a new `__u32 request` field.
- struct v4l2_ext_controls and struct v4l2_query_ext_ctrl both get a new `__u32 request` field.
- This associates buffers with their configuration.
- The request IDs can be any number > 0, but drivers will limit the total number of requests to prevent insanity. A `__u32 max_reqs` field is added to struct v4l2_query_ext_ctrl to let the user know what the maximum is for that control. It is 0 if this control does not support requests.
- Internally request controls are stored in a hash table of `VIDEO_MAX_FRAME` buckets, each bucket a linked list of control values. The hash function is just `request % VIDEO_MAX_FRAME`.

Request API: Public API Changes

- Add a new ioctl: `VIDIOC_REQUEST_CMD`.
-
```
struct v4l2_request_cmd {
        __u32 cmd;
        __u16 request;
        __u16 flags;
        union {
                struct {
                        __u32 data[8];
                } raw;
        };
};
```

- Request commands: V4L2_REQ_CMD_BEGIN/END/DELETE/APPLY.
- Flag for CMD_BEGIN: V4L2_REQ_CMD_BEGIN_FL_KEEP.
- BEGIN stores the request in struct v4l2_fh, END removes it again.

Request API: Public API Changes

- Missing: queuing all requests (buffers + configuration).
- Buffers for a request can be prequeued by calling VIDIOC_PREPARE_BUF.
- But queuing them all for a request requires top-level synchronization.
- Idea: add V4L2_REQ_CMD_QUEUE_ALL. Will call a top-level struct v4l2_device queue_all() callback and after that it is up to the driver to do the work.
- Postpone this until a driver really needs this.

Request API: To Do

- Need agreement on the API.
- Need a bit more thought regarding the KEEP flag.
- Needs more testing.
- Add documentation.