

V4L2 Codec Interface

V4L2 Brainstorm Meeting, Warsaw

S/W Platform Team

2011.03.07

Contents

1

Contents

2

Codec devices

3

Codecs in V4L2 and the Codec Class

4

Requirements

5

Use of fourccs

6

Controls for decoding/encoding

7

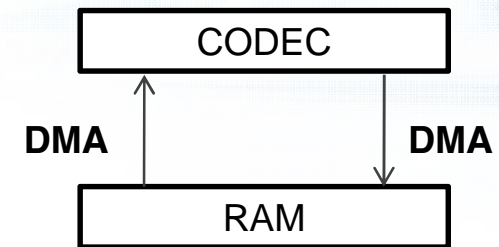
Capabilities

8

Summary

Codec devices

- **The name „codec” is a portmanteau of two words – coder and decoder.**
- **Function: changing the format of the video (audio) stream.**
 - Mostly between decompressed and compressed
- **Memory-to-memory device**
- **Many parameters to tweak in case of encoding**
- **Such hardware is available not only in embedded SoCs, but also GPUs are capable of coding/decoding video**
 - PC applications could also benefit from this interface.



Codecs in V4L2 and the Codec Class

- **Codec class stub is present in the V4L2**
- **For some time the codec related development in V4L2 has been suspended because of lack of experience with such devices**

- **Current definition is similar to M2M devices**

„A V4L2 codec can compress, decompress, transform, or otherwise convert video data from one format into another format, in memory. Applications send data to be converted to the driver through a write() call, and receive the converted data through a read() call. For efficiency a driver may also support streaming I/O.”

- **The above definition is very broad and applies to M2M devices more than a codec device. A codec is particular case of the M2M device.**

- **Proposed Solution**

- M2M class:
 - Transformation, resizing, conversion etc.
- Codec class:
 - Video/Audio compression/decompression

Requirements

- **M2M device**
- **Buffers can be dequeued in arbitrary order**
 - Some frames have to be kept as reference frames for others
 - Hardware may need a pool of buffers and return them in an order other than the order the buffers were queued
- **One buffer on the OUTPUT is not equivalent to one buffer on the CAPTURE queue**
 - Slice interface on the OUTPUT (many → one)
 - Packed PB frames on the OUTPUT (one → many)
- **Processing a single queue is necessary**
 - CAPTURE stream parameters cannot be determined before parsing the header.
- **Encoder and decoder separation**
 - Two video nodes – one for decoding and one for encoding

Use of fourccs

- **How to determine what kind of data is expected on the CAPTURE and OUTPUT side of the codec?**
 - For example if a codec was a „normal” M2M device with RGB32 on one side and YUV420 on the other then using FOURCC would be an obvious way to go.
 - What to do in case the data is compressed?
- **Defintion according to the <http://fourcc.org>**
 - „FOURCC is short for "four character code" - an identifier for a video codec, compression format, color or pixel format used in media files.”
- **Video compression formats mentioned on the <http://fourcc.org>**
 - H263, H264, DIV3, DIV4, DIVX, DX50, WVC1, XVID
- **But not all (other used for example by MFC)**
 - MPEG forucc is very broad MPEG 1, 2 and 4
 - How to specify? Should it be specified by fourcc or in another way?
 - VC1 and VC1 RCV differentiation (RCV is a container) (?)
 - DivX 5.02 and 5.03 – more than DX50 (defined at fourcc.org) ; DX52, DX53 or a dedicated control?

Controls for decoding/encoding

- **Goal – isolate the controls that are generic for decoding/encoding devices**
 - Problem: Lack of documentation for other SoC codecs
- **Proposed solution – inspire the control set on ffmpeg/x264**
 - If an option is available in software decoders/encoders then it is highly possible that it will be common in hardware codecs
- **V4L2_CTRL_CLASS_CODEC**
 - Decoder and encoder specific control separation – adding an offset (this way adding decoder/encoder specific controls in the future should be easy)

Controls for decoding/encoding - continued

- **Example controls for decoding**

- V4L2_CID_CODEC_LOOP_FILTER_MPEG4_ENABLE
- V4L2_CID_CODEC_H264_DISPLAY_DELAY
- V4L2_CID_CODEC_REQ_NUM_BUFS
- V4L2_CID_CODEC_SLICE_INTERFACE
- V4L2_CID_CODEC_PACKED_PB
- V4L2_CID_CODEC_FRAME_TAG

- **Example controls for encoding**

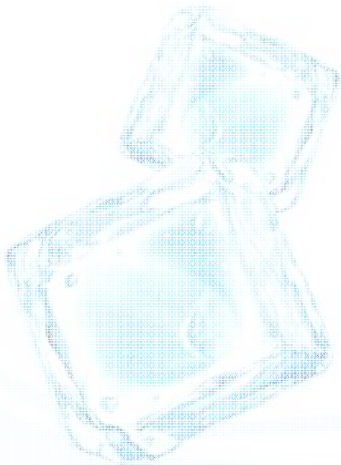
- It would be best to have the controls as general as possible
- V4L2_CID_CODEC_I_PERIOD
- V4L2_CID_CODEC_MIN_QP
- V4L2_CID_CODEC_MAX_QP
- For example the QP for H264 is in the rang of 0..51, and for MPEG4/H263 is 1..31 (how to handle this?) The meaning is the same, so it makes sense to have a common name.
- V4L2_CID_CODEC_H264_LOOP_FILTER_ALPHA (H264 specific)

- **Idea – distinguish between decoder and encoder:**

- Add `_DEC_` and `_ENC_` to the names of controls that are specific to decoding/encoding
- V4L2_CID_DECODER_* and V4L2_CID_ENCODER_*

Capabilities

- **New capabilities V4L2_CAP_VIDEO_DECODER and V4L2_CAP_VIDEO_ENCODER**
 - This way those devices would not show in application expecting a camera (for example Camorama) (addressing Hans de Goede doubts from Helsinki summit)
 - A codec device is an particular case of a M2M device



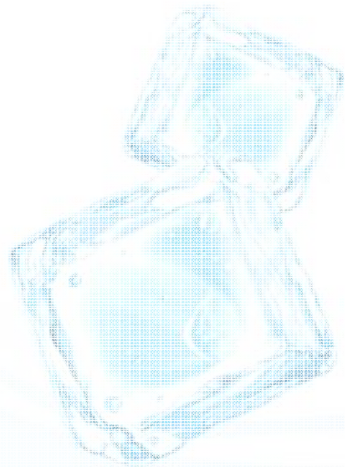
Summary

Main issues to discuss on the brainstorm meeting:

- **Use of FOURCCs for compressed formats**
 - Many formats are already well defined
 - Trouble with more specific separation (like DivX 5.02, DivX 5.03)
- **V4L2 Controls for codecs**
 - Separate control class
- **New capability for codecs**
 - Differentiate codec devices from Output, Capture and M2M devices

Question, comments

Questions.



Thank you.

