Technology Consulting Company
Research, Development &
Global Standard

Optimizing Gstreamer Video Plugins
- A Case Study with Renesas SoC Platform

Embedded Linux Conference 2013

Katsuya MATSUBARA
IGEL Co., Ltd

GStreamer Overview

igel

A Multimedia framework designed to be cross-platform

- The de-facto standard media framework especially for Linux systems
  - Tizen
  - GStreamer for Android
- **Various types of media processing can be realized by describing data flows, called ‘pipelines’, with components, called ‘plugins’.**
- Over 200 plugins exist
  - classified as base, good, bad, ugly
  - 3rd party plugins such as gst-openmax and gst-ffmpeg

Plugins（Elements）

igel

Source element : Generates data

- **filesrc** : Reads a file
- **videotestsrc** : Creates a test video stream
- **v4l2src** : Reads frames from a V4L2 device

Filter or filter-like element : Receives and provides data

- **ffmpegcolorspace** : Converts video from one colorspace to another
- **videocrop** : Crops video image into a sub-region
- **qtdemux** : Demultiplexes mp4, mov, 3gp container files into audio and video streams
- **ffdec_h264** : Decodes H.264 video streams using ffmpeg

Sink element : Accepts data

- **fbdevsink, xvimagesink, dfbvideosink** : Video rendering
- **filesink** : Writes stream into a file

GStreamer Overview (contd.)

igel

- gst-launch : a test tool to easily describe media processing pipelines
- Two series of GStreamer releases exist
  0.10 - Popular and widely used, The latest is 0.10.36.
  1.0 (previously 0.11) – The current stable version since Sep., 2012. ABI/API are not compatible with 0.10. Some of plugins are not migrated yet. The latest is 1.0.5.

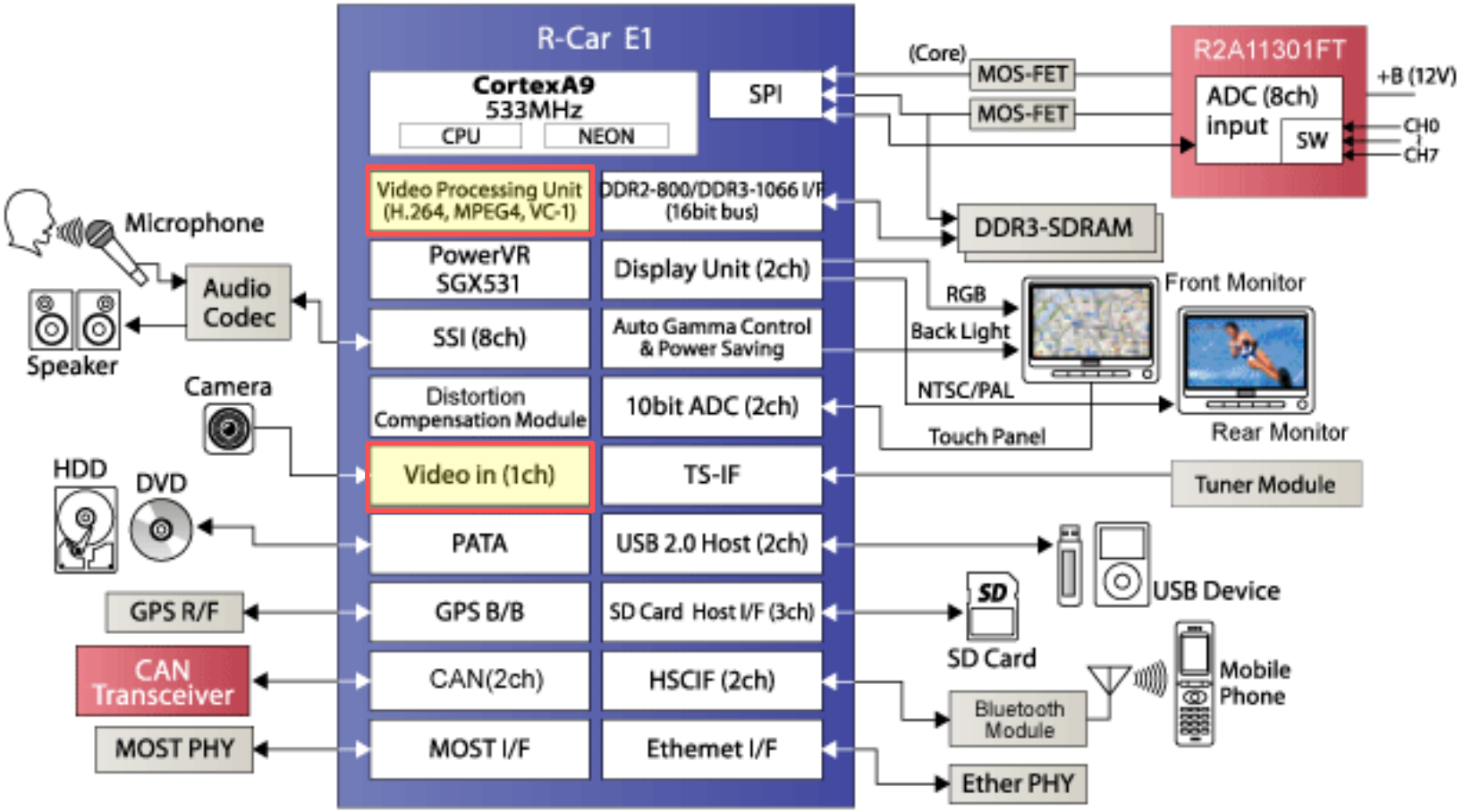
My work is based on GStreamer 0.10.36.

Goals

igel

- Construct typical video applications on the target SoC platform with existing GStreamer plugins
  1. Video (camara) monitoring
  2. Video playback
- Optimize performance where possible
  - Enable full utilization of hardware accelerators in SoC.
  - Get rid of CPU intensive overhead such as memcpy().

Renesas R-Car E1 SoC

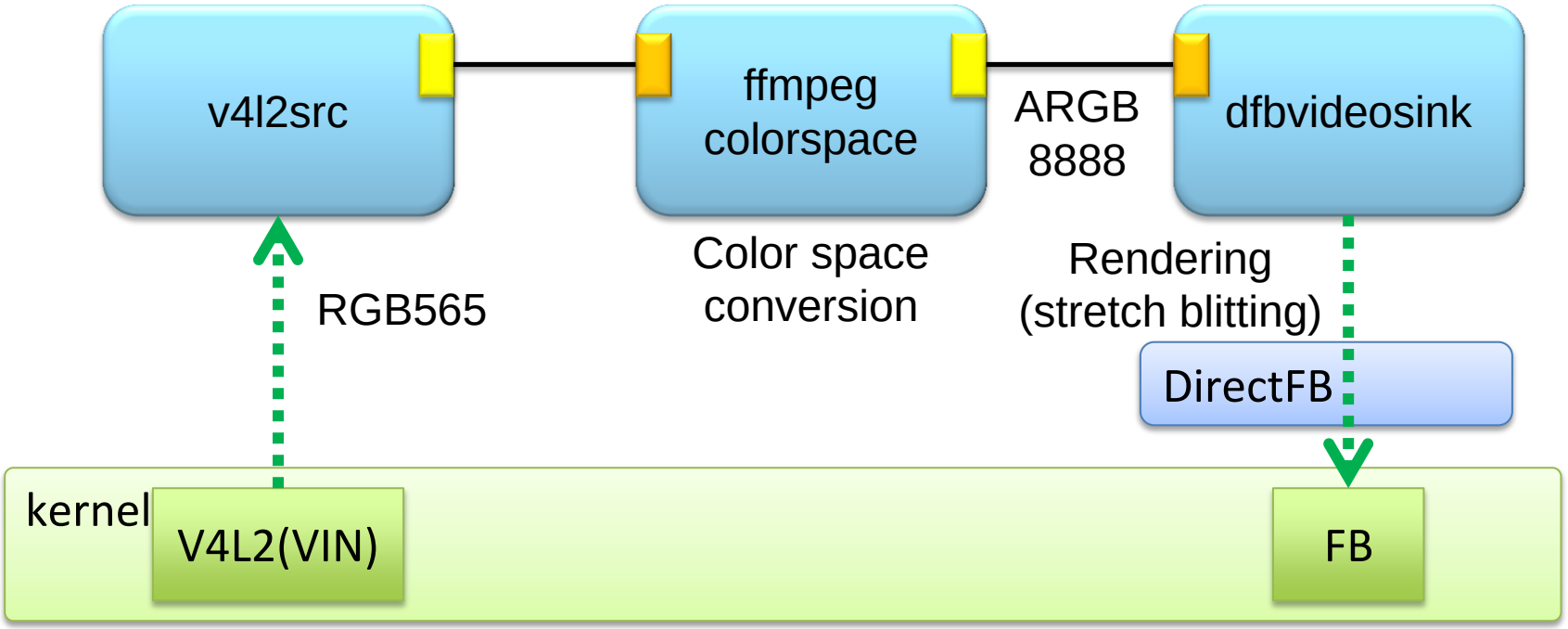


Quoted from http://www.renesas.com/applications/automotive/cis/cis_highend/rcar_e1/index.jsp

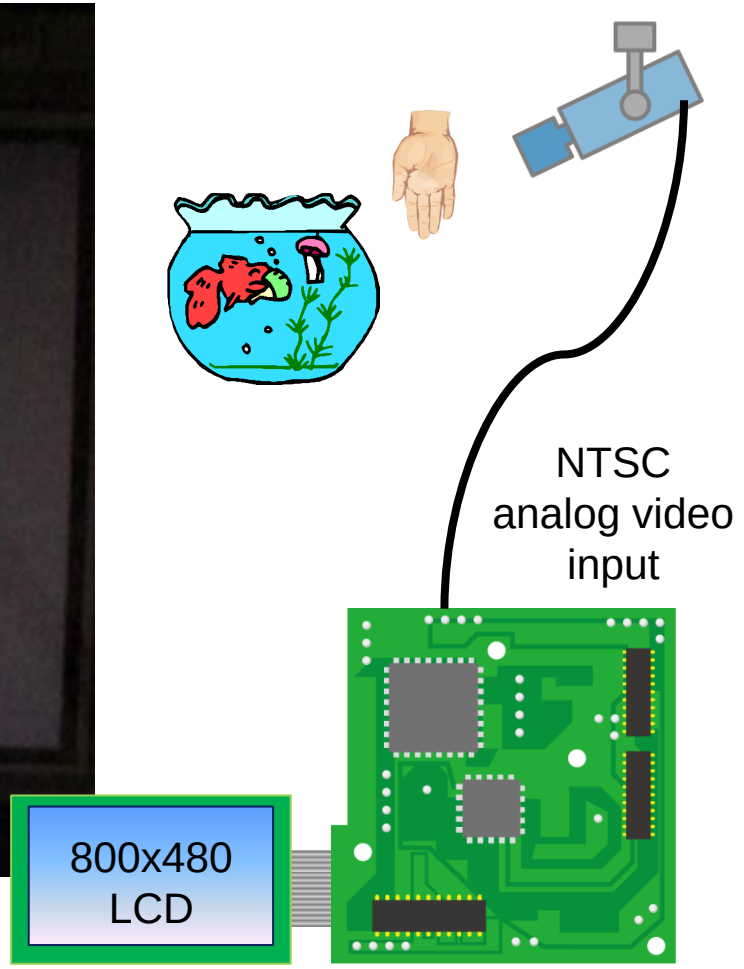
1. VIDEO (CAMERA) MONITORING

Pipeline for Video Monitoring

v4l2src ! ffmpegcolorspace ! \
video/x-raw-rgb, bpp=32, depth=32, ... ! dfbvideosink



Video Monitoring on the Renesas R-CarE1 Board



Why such poor performance?

igel

v4l2src

- CPU memcpy() has been executed on each video frame to copy data from V4L2 buffer to GST buffer.
- VIN operates in a lower speed operation mode.
  - VIN has two mode, the single capturing mode (-15fps) and the continuous capturing mode (-30fps).
  - The driver decides which mode can be adopted **according to the number of buffers prepared**.

ffmpegcolorspace

- Color space conversion is implemented in S/W.

dfbvideosink

- Stretch blitting (scaling and rendering) operations have been realized in S/W.

Optimizing Video Monitoring

igel

v4l2src

1. Suppress the CPU memcpy() operation
2. Activate the continuous capturing mode of VIN by supplying sufficient number of buffers

dfbvideosink

3. Use acceleration hardware in SoC for color space conversion and stretch blitting
   - The ffmpegcolorspace plugin is no longer necessary.

Tuning the v4l2src Plugin

igel

Element properties : Show configuration or configure the element

Quoted from result of running "`gst-inspect v4l2src`"

```
Element Properties:
...
queue-size          : Number of buffers to be enqueud in
                      the driver in streaming mode
                      flags: readable, writable
                      Unsigned Integer. Range: 1 - 16
                      Default: 2 Current: 2
always-copy         : If the buffer will or not be used
                      directly from mmap
                      flags: readable, writable
                      Boolean. Default: true Current: true
```

Set 'queue-size=4 or more' to activate the continuous capturing mode, and set 'always-copy=false' to suppress memcpy()s.

Utilize Hardware Acceleration in the dfbvideosink Plugin

igel

Requirements

- Hardware access from user-space
  - read/write registers of hardware
  - handle interrupts signaled by hardware
- DMA (physical contiguous) memory allocation in user-space
  - hardware requires memory which can be accessed by DMA.
- Physical address of buffer memory held in user-space
  - hardware can only handle physical addresses.
  - user-space driver directly manages hardware transactions.

Buffer Features Used in the Current Pipeline

igel

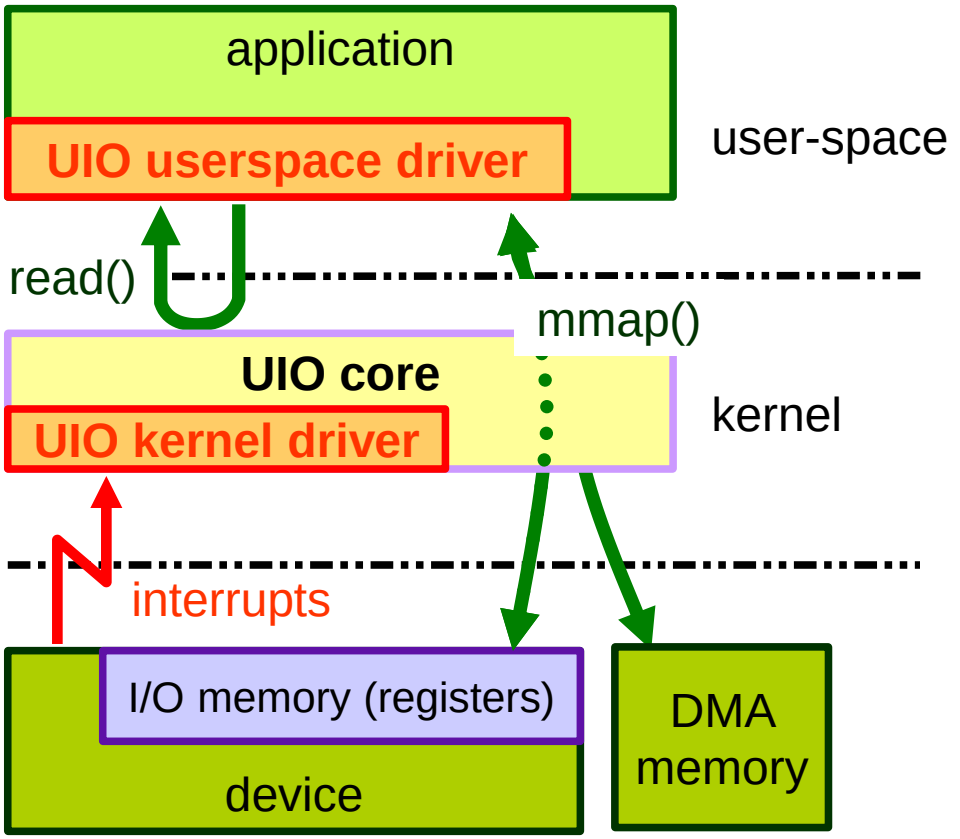
	Hardware can access it through DMA?	Physical address of buffer can be seen from user-space?
V4L2 buffers (allocated by V4L2 driver)	○	×
GST buffers (typically allocated by malloc())	×	×
Frame buffer (allocated by FBDEV driver)	○	○ through ioctl(FBIOGET_FSCREENINFO)

Hardware Access & DMA Memory Allocation in User-space

igel

UIO (User-space I/O)

- Allows access to I/O memory (registers) through mmap().
- Allocates DMA (physical contiguous) memory, exports physical address of the allocated region, and fulfills access through mmap().
- Handles interrupts through read().



Physical Address of Buffer Memory in User-space

libuiomux: a library to manage UIO resources

- Resource management for DMA memory
- Virtual-physical address transition for I/O memory and DMA memory region exported through UIO.
- Exclusive access control for UIO devices.

Eliminate more memcpy()s

igel

V4L2_MEMORY_USERPTR

- To use buffers prepared in user-space instead of ones allocated by the kernel V4L2 driver.

In v4l2src,
DMA memory region allocated through UIO can be assigned to V4L2 buffers.

In dfbvideosink,
the buffers can be read directly by hardware using corresponding physical address.

Buffer Features Used in the Optimized Pipeline

igel

	Hardware can access through DMA?	Physical address of buffer can be seen from user-space?
V4L2 buffers (allocated by UIO kernel driver)	○ given from user-space with V4L2_MEMORY_USERPTR	○ managed by libuiomux
GST buffers (typically allocated by malloc())	(never used by eliminating memcpy()s)	
Frame buffer (allocated by FBDEV driver)	○	○ through ioctl(FBIOGET_FSCREENINFO)

Stretch Blitting and Color Space Conversion by Hardware

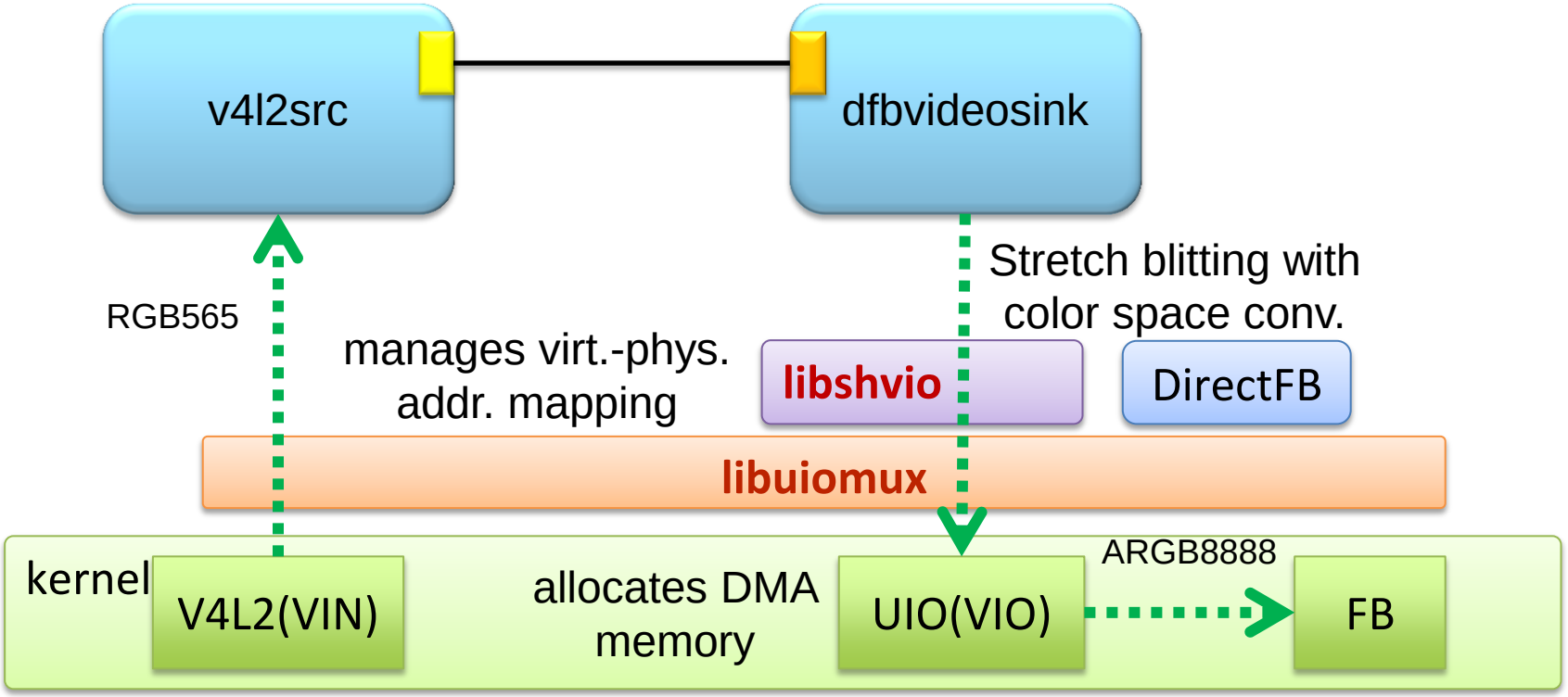
- libshvio: a library that works as UIO user-space driver

Main on-chip peripheral functions	■ Video input interface × 1 channels ■ VPU5HD2 (H.264/AVC, MPEG-4, VC-1) ■ Video image processing (color conversion, image expansion, reduction, filter processing) ■ Distortion compensation module (image renderer) ■ SD card host interfaces × 3 channels ■ Multimedia card interface ■ Serial audio/sound interfaces × 8 channels ■ Media local bus (MLB) interface ×1 (MediaLB Ver2.0, 512 fs(max) support) ■ USB 2.0 HS × 2 channels ■ GPS baseband processing module ■ TS interface

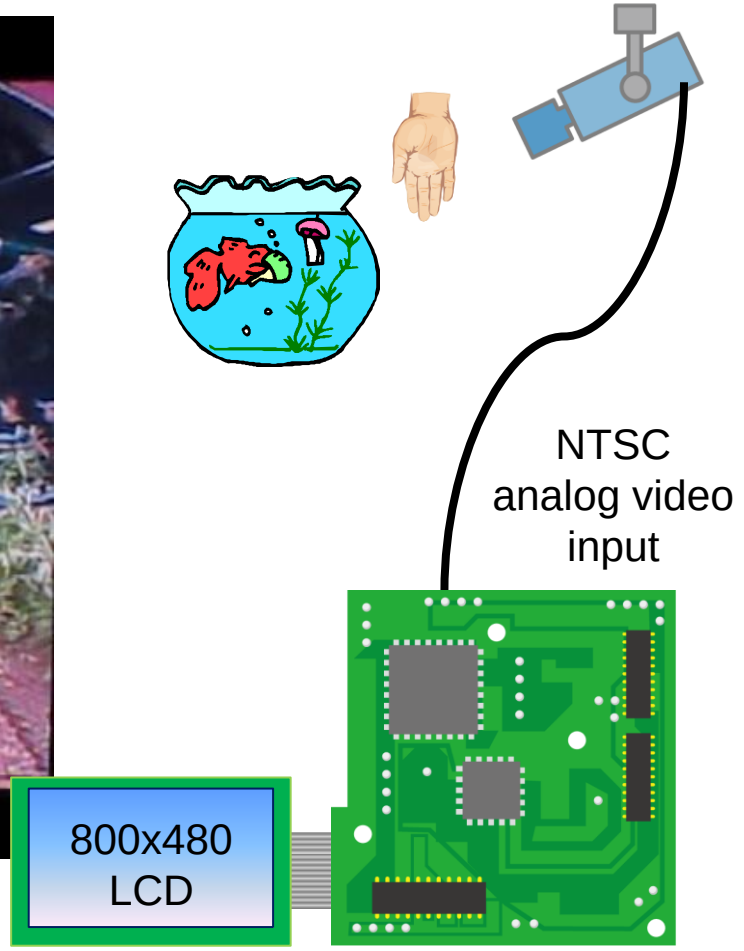
Quoted from http://www.renesas.com/applications/automotive/cis/cis_highend/rcar_e1/index.jsp

Optimized Pipeline for Video Monitoring

v4l2src queue-size=5 always-copy=false ! dfbvideosink



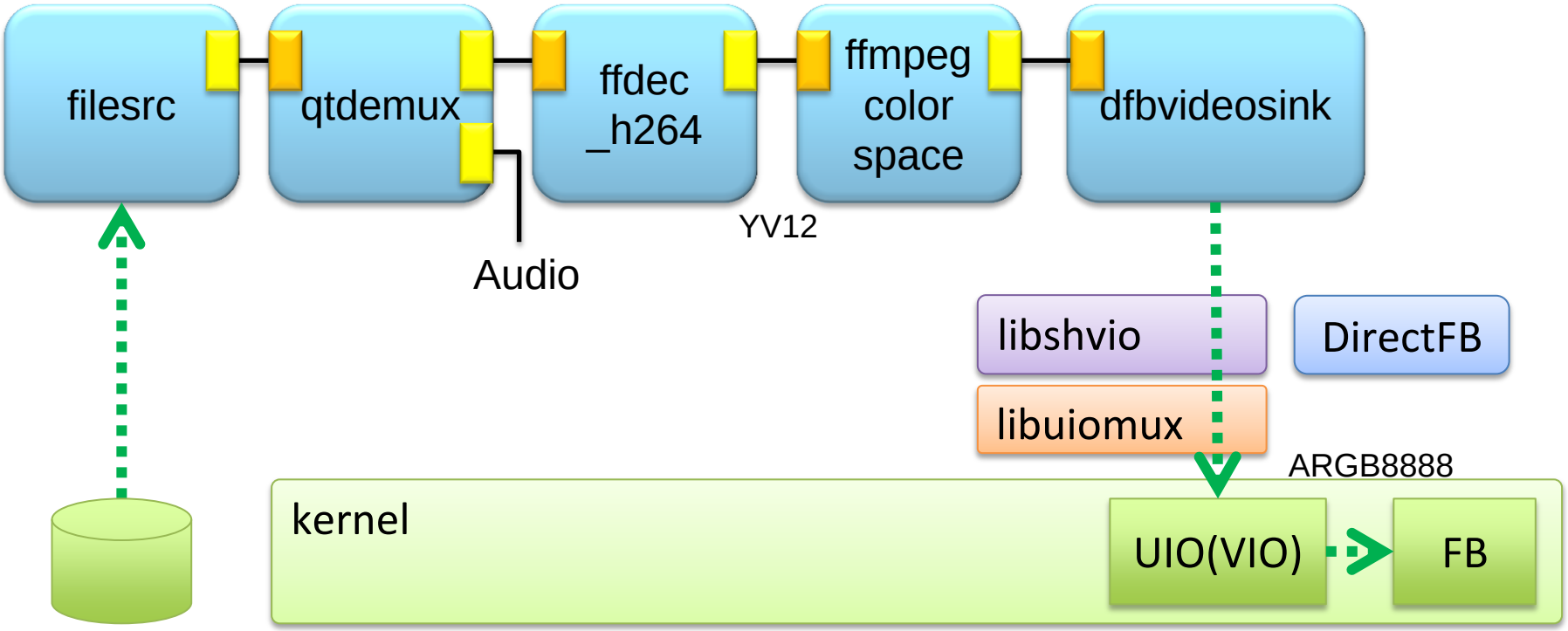
Optimized Video Monitoring on the Renesas R-CarE1 Board



2. VIDEO PLAYING (DECODING)

A Pipeline for H.264 Video Playback

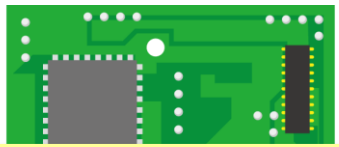
filesrc location=video.mp4 ! qtdemux name=dmx \
dmx.video_00 ! ffdec_h264 ! ffmpegcolorspace ! \
video/x-raw-rgb, bpp=32, .depth=32, .. ! dfbvideosink



Video Playback on the Renesas R-CarE1Board

igel

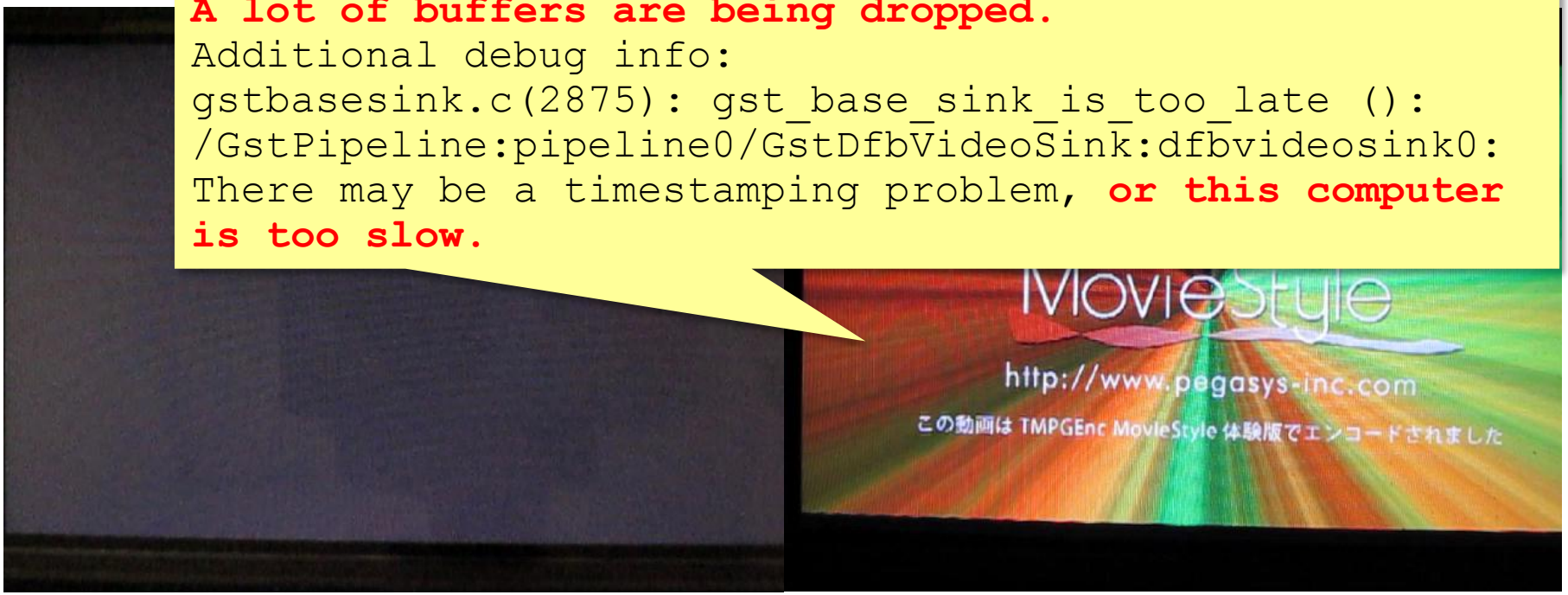
176x144
H.264 video



720p
H.264 video

After a while, the following logs appear in console, ...

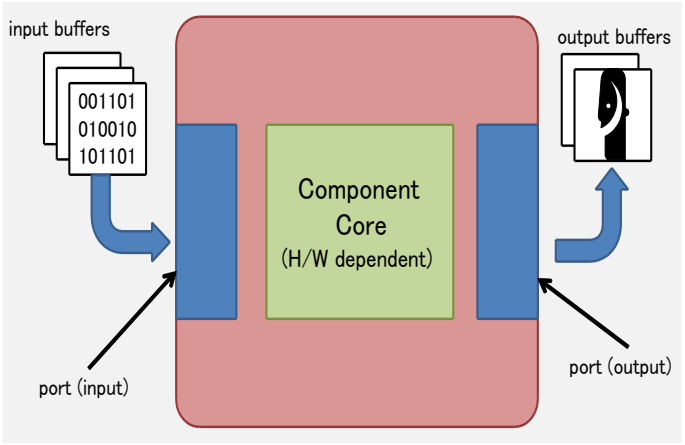
```
WARNING: from element
/GstPipeline:pipeline0/GstDfbVideoSink:dfbvideosink0:
A lot of buffers are being dropped.
Additional debug info:
gstbasesink.c(2875): gst_base_sink_is_too_late ():
/GstPipeline:pipeline0/GstDfbVideoSink:dfbvideosink0:
There may be a timestamping problem, or this computer
is too slow.
```



Use Hardware Decoder: OpenMAX IL (OMXIL)

igel

- Standard interface for media components
  http://www.khronos.org/openmax/
- Often used as the standard API of codec (decoder and encoder) engine
- Adopted as the codec interface by Android
- Simple and flexible specification
  - GetParameter(), SetParameter()
  - FillThisBuffer(), EmptyThisBuffer()
  - FillBufferDone(), EmptyBufferDone()
- Component (binaries) often distributed by chip vendors or board suppliers.



gst-openmax

igel

A GStreamer plugin to control OMXIL components

- Filter Elements (decoder, encoder)
  - H.263, H.264, MPEG4, WMV(VC-1)
  - AAC, ADPCM, AMRNB, AMRWB, Vorbis, MP2, MP3, G711, G729
  - JPEG, Volume
- Source Element, Sink Element
- OMXIL 1.1.1 client implementation

Unfortunately most OMXIL components often require ‘calibration’ against OMXIL client because spec. is not strictly defined. cf. quirks in Android Stagefright

Integrating a Vender's OMXIL Component

igel

- Granularity of data input
  Example : H.264 decoder
  - qtdemux's output: frame per buffer, SPS and PPS units put into 'codec_data' in caps rather than in buffer
  - ffdec_h264's input: any segment acceptable, SPS/PPS units taken from the caps (meta-data attached to input/output)
  - omx_h264dec + REL OMXIL's input: one NAL unit per buffer, SPS/PPS units also input through buffers
- Who allocates buffers
  - UseBuffer() : Buffers allocated by anyone else can be used.
  - AllocateBuffer() : Buffers must be allocated by OMXIL component itself.

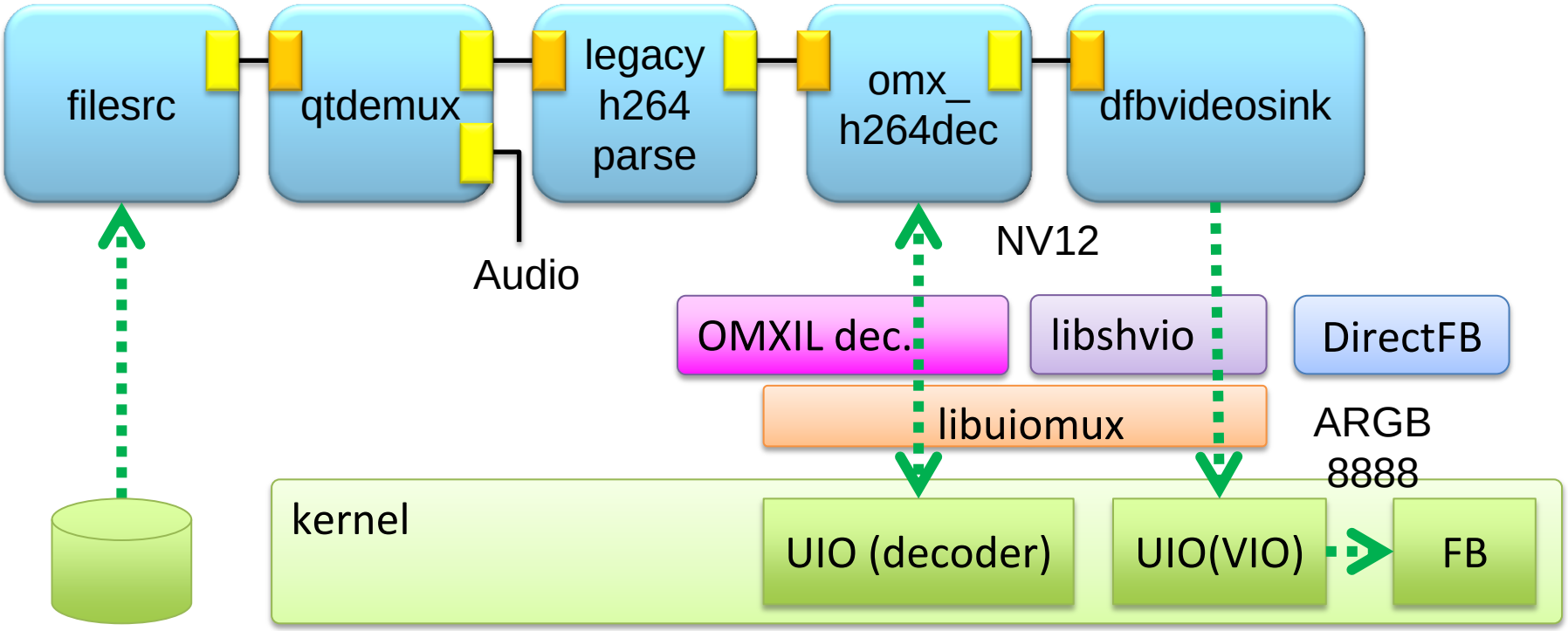
Integrating a Vender's OMXIL Component (contd.)

igel

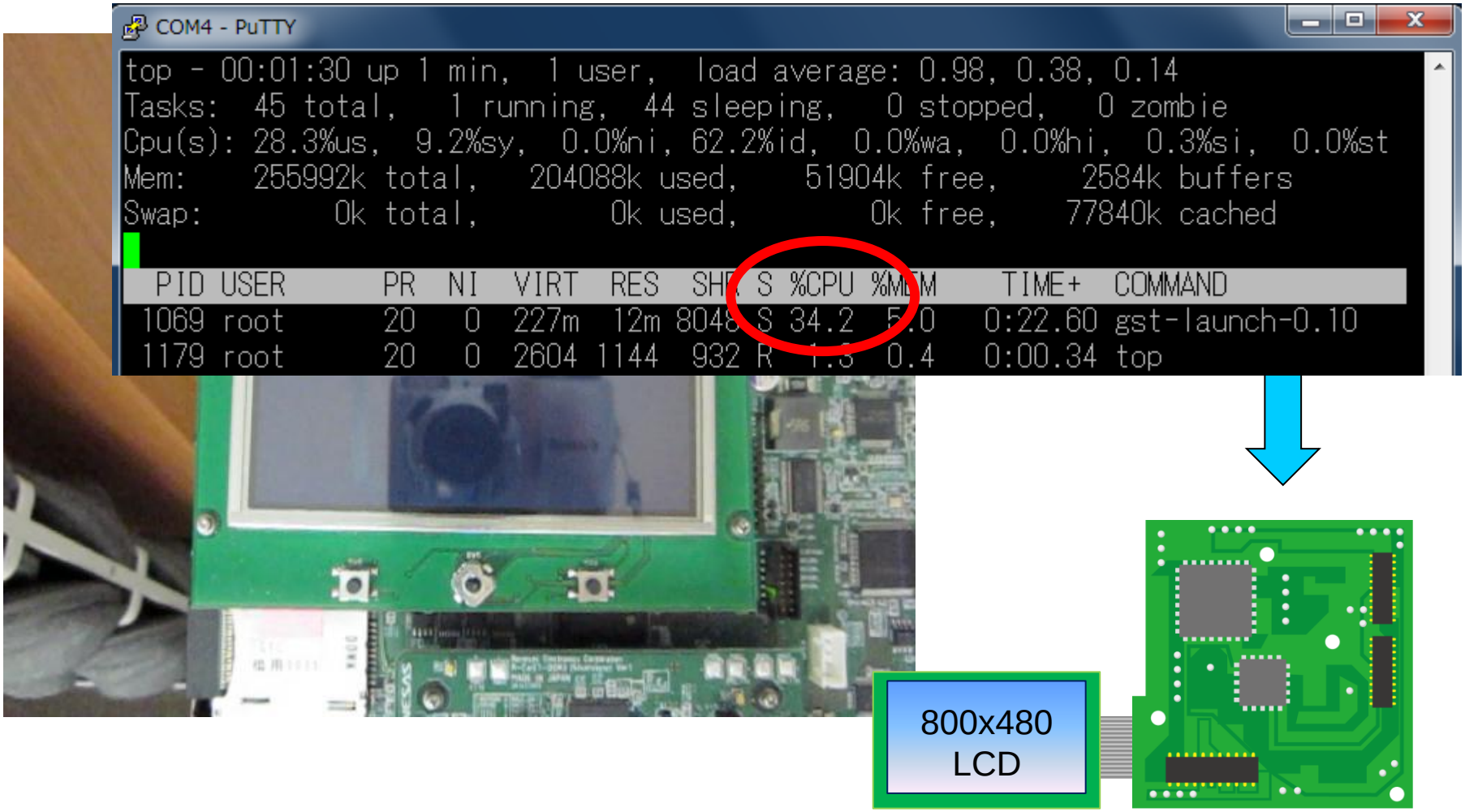
- Set up vendor-specific parameters
  - May need to configure the internal setting of OMXIL component
- Deal with vendor-specific behavior
  - Example : May require an explicit buffer flush whenever the SEEK command is issued.
- Inform additional output data attributes through caps
  - Row stride of decoded image
    - alignment restriction may be induced by hardware
  - Tiled-linear(T/L) addressing
    - Tiled video frame may be output for optimal performance.
  - Composition of interlaced image : de-interlaced, Top-Bottom, or Top-Bottom sequential ordered

Optimized Pipeline for H.264 Video Playback

filesrc location=video.mp4 ! qtdemux name=dmx \
dmx.video_00 ! legacyh264parse output-format=1 \
split-packetized=true ! omx_h264dec ! dfbvideosink



Optimized Video Playback on the Renesas R-CarE1 Board



Conclusion

igel

- Utilized hardware accelerators via user-space plugins.
  - UIO + image processing H/W
    - Implemented a user-space device driver, libshvio
  - gst-openmax + vendor's OMXIL decoder component
    - Adjusted for vendor-specific requirements
- Organized buffer memory management for hardware usage.
  - Assigned appropriate (physical contiguous) memory to buffers
  - Realized virtual-physical address transition for buffers in user-space
- Eliminated CPU memcpy()s in plugins and pipelines.

Conclusion (contd.)

igel

Future Work

- Migrate to GStreamer 1.0
  - GstBuffer in 1.0 offers more flexibility when handling special memory, especially for hardware optimization.
  - Necessary to submit my work to the community.

Links

- Patches for GStreamer plugins, libraries and kernel https://github.com/matsu/
- Install guide for Renesas R-CarE1 Silverstone https://github.com/matsu/gst-openmax/wiki/Quick-Install-Guide-for-Renesas-R-CarE1-Silverstone