

# Development of hardware-based elements for GStreamer v1.0

Víctor Manuel Jáquez Leal

2012-08-27 Mon



Once upon a time...



Once upon a time...



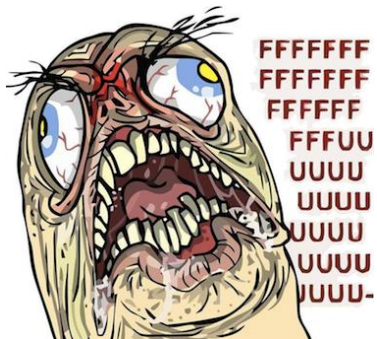
 *gstreamer*

 OpenMAX™

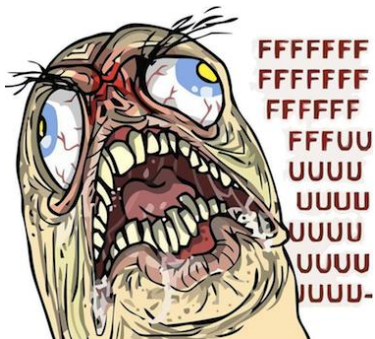
 OMAP™

TEXAS INSTRUMENTS TECHNOLOGY

Yeah...



Yeah...



OpenMAX™

# What I learned

*Everything should be made as simple as possible,  
but not simpler.*

*– Einstein / Zukofsky / Sessions / Ockham / ...*

# Meanwhile in GUADEC 2012

*Drop abstraction, make our layers thin  
Don't stack things anymore  
– Lennart Poettering*

# Then I moved to Spain





The love is still in the air...



**AMIP**  
(Asymmetric Multiprocessing)

# GStreamer v0.10

- ▶ Workarounds
- ▶ Reinventing the wheel

# What's AMP?

Asymmetric MultiProcessing

# What's IPC?

Inter-Processors Communication

What do I want?



# Wat!?

- ▶ A simple,
- ▶ bare bones,
- ▶ home brew

## video player

# What do I need?

- ▶ A simple video sink
- ▶ A video decoder which uses IVAHD

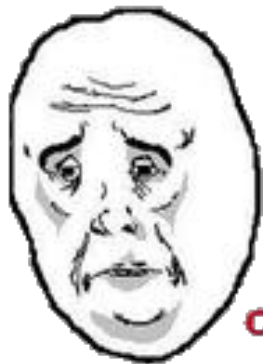
Let's go





# Video Sink

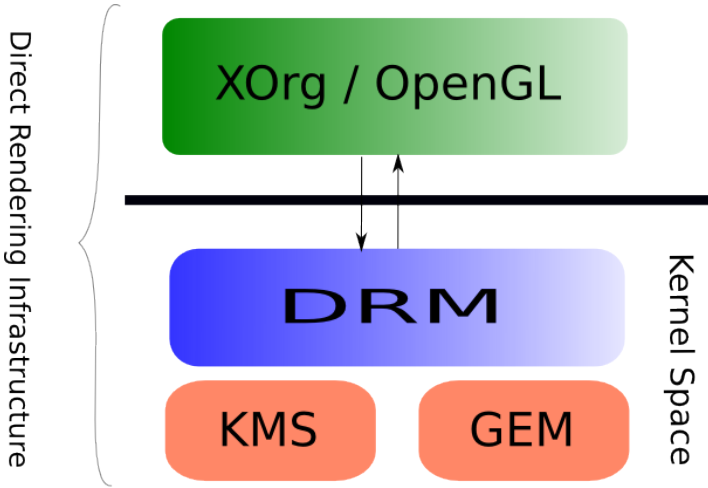
No frame buffer



But there is DRM/KMS



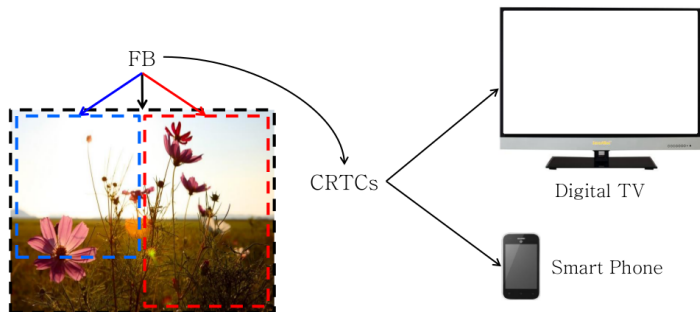
# Direct Rendering Management



# Kernel Mode Settings

- ▶ framebuffer
- ▶ CRTC
- ▶ Overlay
- ▶ Encoder
- ▶ Connector

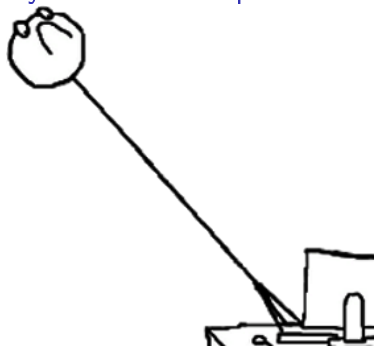
# Kernel Mode Settings



© 2011 SAMSUNG Electronics Co.

# Video Decoders

SysLink 2 is deprecated



omapdce

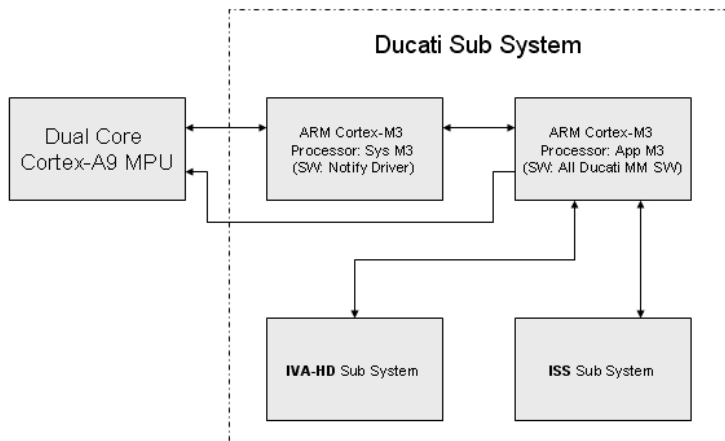
**CHALLENGE ACCEPTED**

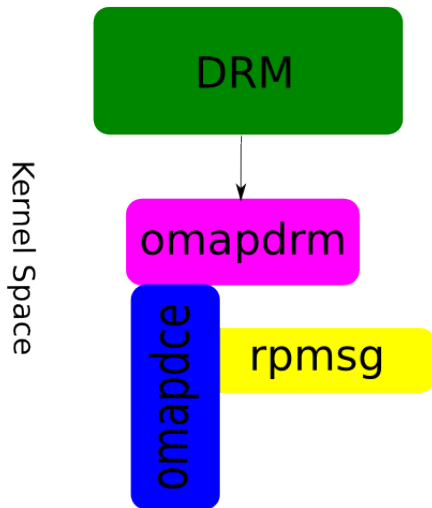


# Distributed Codec Engine (DCE)

- ▶ It provides access to the Codec Engine interface on the co-processor (ducati/M3) from the host.
- ▶ It invokes the hardware accelerated codecs on IVA-HD via `rpmsg/remoteproc`

# Ducati







# History of the TI's IPCs?

- ▶ **dsp-gateway** - OMAP1/OMAP2
- ▶ **DSPLink** - OMAP2/OMAP3/DaVinci
- ▶ **DSP/Bridge** - OMAP3
- ▶ **SysLink2** - OMAP4
- ▶ **RPMsg** - OMAP4/OMAP5

# RPMsg

- ▶ remoteproc
- ▶ rpmsg

# remoteproc

- ▶ Loading & Bootup
- ▶ Power Management
- ▶ Exception Management
- ▶ Error Recovery

- ▶ virtio-based messaging bus that allows kernel drivers to communicate with remote processors

HIC SVNT DRACONES



# GstBuffer

- ▶ `gst_buffer_map()`
- ▶ `gst_buffer_unmap()`
- ▶ Buffers can contain a list of `GstMemory` objects

# GstMemory

- ▶ It is a lightweight & refcounted object.
- ▶ It wraps a region of memory.
- ▶ Memory is usually created by allocators.

# GstKMSMemory

```
struct _GstKMSMemory {  
    GstMemory mem;  
    uint32_t fb_id;  
    uint32_t handle[4];  
    void *bo[4];  
};
```



# GstAllocator

- ▶ Allocate memory blocks
- ▶ Virtual methods
  - ▶ alloc
  - ▶ free
- ▶ Callbacks
  - ▶ map
  - ▶ unmap
  - ▶ copy
  - ▶ share
- ▶ Parameters
  - ▶ align
  - ▶ prefix
  - ▶ padding

## GstKMSAllocator (alloc)

```
static GstMemory *
gst_kms_allocator_alloc (GstAllocator * allocator,
                        gsize size,
                        GstAllocationParams * params)
{
    GstKMSAllocator *self = GST_KMS_ALLOCATOR (allocator);
    GstKMSMemory *buf = g_slice_alloc (sizeof (GstKMSMemory));
    buf->fb_id = -1;
    buf->bo[0] = omap_bo_new (self->dev, size,
                            OMAP_BO_SCANOUT | OMAP_BO_WC);
    buf->handle[0] = omap_bo_handle (buf->bo[0]);
    gst_memory_init (GST_MEMORY_CAST (buf),
                    GST_MEMORY_FLAG_NO_SHARE, allocator, NULL, size,
                    0, 0, size);
    return GST_MEMORY_CAST (buf);
}
```

## GstKMSAllocator (free)

```
static void
gst_kms_allocator_free (GstAllocator * self,
                        GstMemory * mem)
{
    GstKMSMemory *buf = GST_MEMORY_CAST_KMS (mem);
    omap_bo_del (buf->bo[0]);
    g_slice_free1 (sizeof (GstKMSMemory), mem);
}
```

## GstKMSAllocator (map / unmap)

```
static void *
gst_kms_allocator_map (GstMemory * mem,
                      gsize maxsize,
                      GstMapFlags flags)
{
    GstKMSMemory *buf = GST_MEMORY_CAST_KMS (mem);
    omap_bo_cpu_prep (buf->bo[0], OMAP_GEM_WRITE);
    return omap_bo_map (buf->bo[0]);
}
```

```
static void
gst_kms_allocator_unmap (GstMemory * mem)
{
    GstKMSMemory *buf = GST_MEMORY_CAST_KMS (mem);
    omap_bo_cpu_fini (buf->bo[0], OMAP_GEM_WRITE);
}
```

# GstMeta

- ▶ Describes the low-level properties of the buffer content.
- ▶ It contains variables that can change between each buffer
  - ▶ timestamp, duration, offset, interlacing
  - ▶ video alignment, cropping, panning information
  - ▶ extra container information
  - ▶ global buffer properties

# GstBufferPool

- ▶ Pre-allocate and recycle buffers with the same properties.
- ▶ After the buffer pool is created, it needs to be configured.
  
- ▶ Buffer pools increase performance
  - ▶ reduces allocation overhead
  - ▶ add more possibilities to implement zero-copy memory

# GstVideoBufferPool

- ▶ GstBufferPool + Video Meta + Video Alignment

# GstVideoContext

- ▶ It shares video context between neighbour elements and the application.
- ▶ `gst-plugins-bad / GST_USE_UNSTABLE_API`
- ▶ Sharing the DRM file descriptor between the decoder and the sink.



```
gst_kms_sink_query (GstBaseSink * bsink,  
GstQuery * query)
```

```
{  
    GstKMSSink *sink = GST_KMS_SINK (bsink);  
    const char **types =  
        gst_video_context_query_get_supported_types (query);  
    if (!types)  
        return GST_BASE_SINK_CLASS (parent_class)->query (bsink);  
  
    for (i = 0; types[i]; i++) {  
        if (!strcmp (types[i], "drm-fd")) {  
            GstStructure *structure =  
                gst_query_writable_structure (query);  
            gst_structure_set (structure, "video-context-type",  
                G_TYPE_STRING, "drm-fd", "video-context",  
                G_TYPE_INT, sink->fd, NULL);  
        }  
    }  
}
```

# GstVideoDecoder

- ▶ Base class for converters of encoded data into raw video frames.
- ▶ Configuration.
  - ▶ `_start ()`
  - ▶ `_set_format ()`
- ▶ Data processing.
  - ▶ `get` or `_parse ()` frames
  - ▶ `_handle_frame ()`
  - ▶ call `gst_video_decoder_finish_frame ()` or `gst_video_decoder_drop_frame ()`
- ▶ Shutdown phase.
  - ▶ `_stop ()`

# @TODO

- ▶ kmssink should support more DRM backends
- ▶ dcevddec should support more decoders
- ▶ Implement a DRM buffer metadata
- ▶ Push kmssink and dcevddec (??) to bugzilla
- ▶ Watch a movie with my video player

# Lessons I learned

- ▶ I suck at communication
  - ▶ Communicate your development openly
- ▶ GStreamer v1.0 looks great!
- ▶ omapdce/rpmsg + remoteproc too!

# Thanks to

- ▶ Rob Clark
- ▶ Ohad Ben-Cohen
- ▶ Nicolas Dechesne
- ▶ Alessandro Decina
- ▶ Sebastian Dröge
- ▶ GStreamer community

# Credits

Clear Inner Vision - CC BY-NC-ND 2.0

The 1265 Psalter world map

Rage Comics

Questions?

