

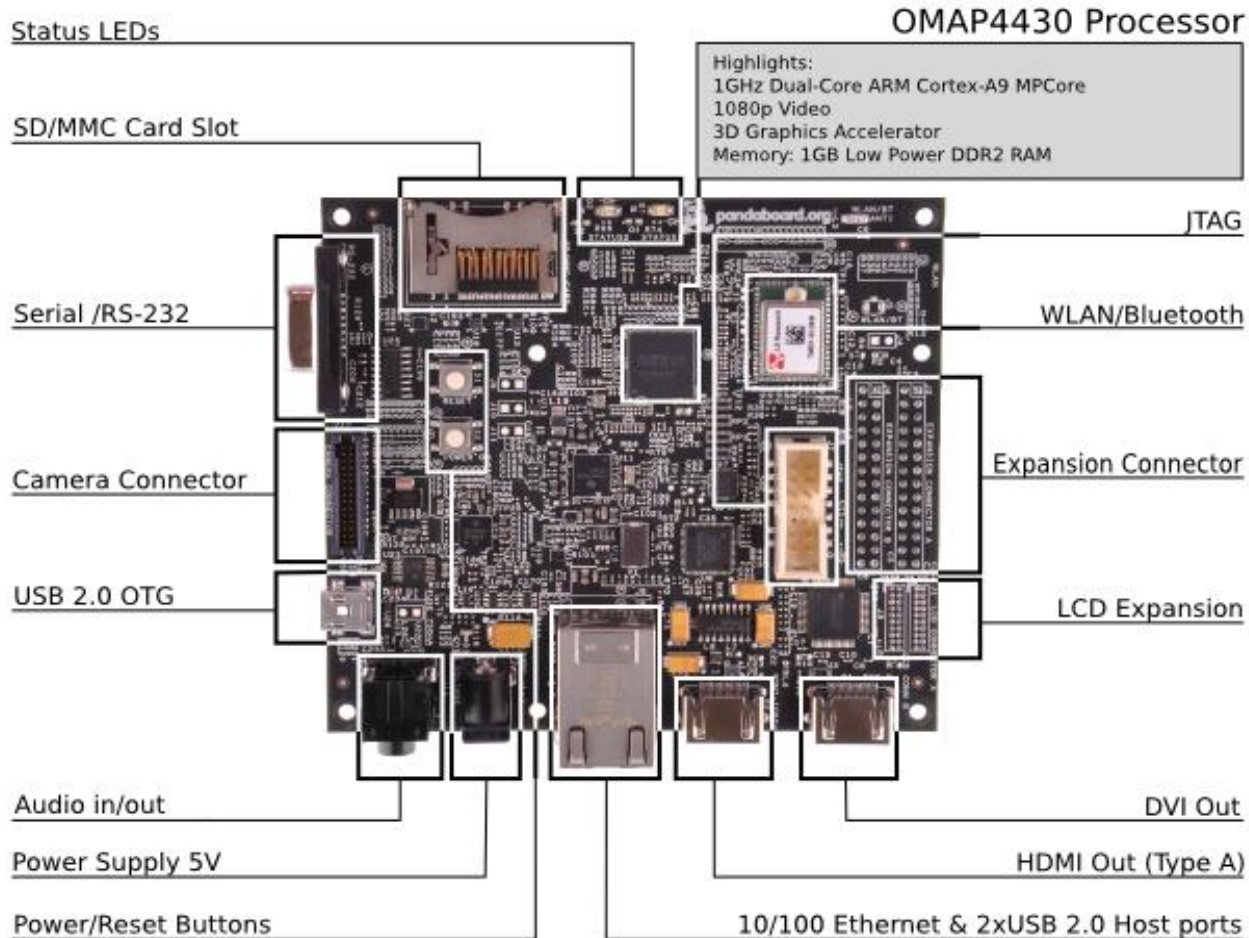
# Разработка видеоплеера на Pandaboard с использованием GStreamer

**Дмитрий Комягин, Иван Кутень**  
Инновационная компания Promwad

## Содержание

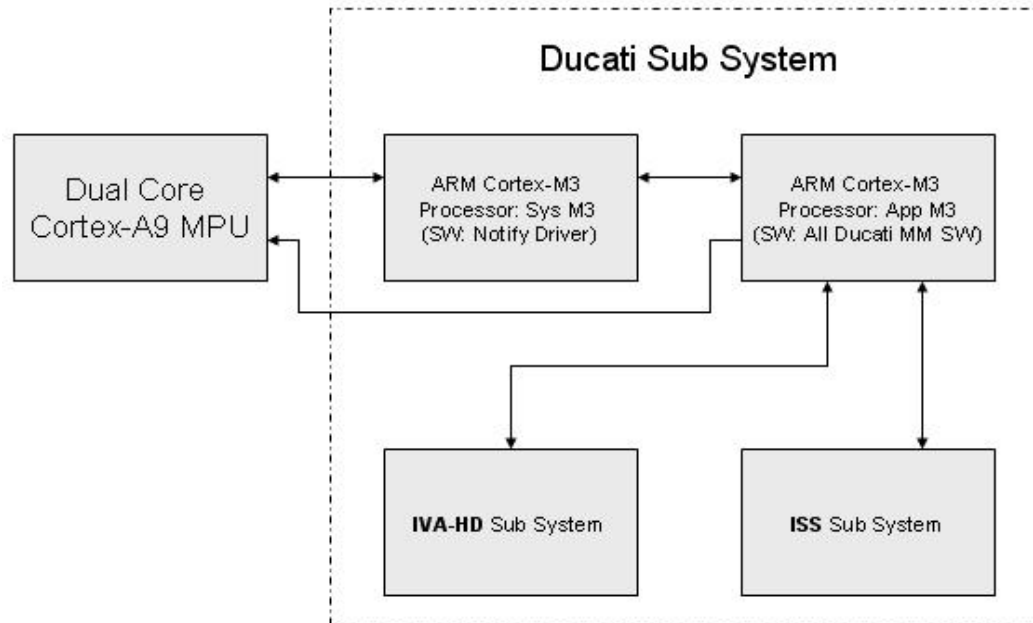
- Pandaboard (OMAP4430)
- Ducati
- IVA-HD
- Distributed OpenMAX
- Distributed Codec Engine
- Библиотеки для работы с видео
- GStreamer
- GStreamer видеоплеер

# Pandaboard (OMAP4430)



Board Dimensions: W:4.0" (101.6 mm) X H: 4.5" (114.3 mm)

# Ducati



- Dual Core Cortex-A9 MPU является хост-процессором OMAP4, на котором запускается ОС Linux (Android)
- На одном из 2 процессоров Cortex-M3 (Sys M3) запускается драйвер сообщений, который принимает команды от запущенного на Linux ПО и передает их для выполнения на другой процессор Cortex-M3 (App M3)

## IVA-HD

- 1080p30 / 1080i60
- Поддерживается аппаратное ускорение следующих кодеков:
  - H.264 BP/MP/HP encode/decode
  - MPEG-4: SP/ASP encode/decode
  - DivX 5.x and higher encode/decode
  - H.263 Profile 0/3 decode, profile 0 encode
  - MPEG-2 SP/MP encode/decode
  - MPEG-1 encode/decode
  - VC-1/WMV9 encode/decode
  - On2® VP6/VP7 decode
  - RealVideo® 8/9/10 Decode
  - JPEG/MJPEG baseline encode/decode
  - H.264 Annex H MVC (stereo) up to 720p30

## Distributed OpenMAX

- OpenMax — Proxy
- DOMX AppM3:
  - DOMX
  - Исходный код закрыт  
(предоставляются только бинарные файлы)

## Distributed Codec Engine (DCE)

- libdce — библиотека предоставляет удаленный доступ к аппаратному ускорению декодирования видео на IVA-HD через слой syslink/rcm
- DCE AppM3:
  - TI Codec Engine (CE)
  - Исходный код открыт (под лицензией TI)

## Библиотеки для работы с видео

Варианты использования hw-декодирования видео:

- Использовать напрямую libDCE
- GStreamer (плагин gst-ducati )
- OpenMAX/DOMX/Gstreamer (gst-openmax)
- QtMobility (для Qt4-приложений)





### gstreamer tools

gst-inspect  
gst-launch  
gst-editor

media player

VoIP & video conferencing

streaming server

video editor

(...)

### multimedia applications

**gstreamer core framework**

media agnostic  
base classes  
message bus  
media type negotiation  
plugin system  
data transport  
synchronization

**pipeline architecture**

```

    graph LR
      subgraph pipeline
        direction LR
        S[File source] --> D[mpeg-demuxer]
        D --> AD[audio-decoder]
        D --> VD[video-decoder]
        AD --> AS[audio-sink]
        VD --> VS[video-sink]
      end
  
```

**protocols**

- file:
- http:
- rtsp:
- ...

**sources**

- alsa
- v4l2
- tcp/udp
- ...

**formats**

- avi
- mp4
- ogg
- ...

**codecs**

- mp3
- mpeg4
- vorbis
- ...

**filters**

- converters
- mixers
- effects
- ...

**sinks**

- alsa
- xvideo
- tcp/udp
- ...

3rd party plugins

**gstreamer plugins**  
gstreamer includes over 250 plugins

**3rd party plugins**

## GStreamer видеоплеер (Pandaboard)

Необходимые компоненты видеоплеера с использованием аппаратного декодирования видео и вывода на экран:

- GStreamer/gst-plugins
- Libdce/tisyslink/timemmgr
- Декодер: gst-ducati
- Вывод: pvrvideosink (EGL)
- <http://gitorious.org/gstreamer-omap>
- <http://github.com/robclark>

**Проверка работы аппаратного декодирования видео:**

```
>gst-launch playbin2 uri="file:///videofile.mkv" video-sink="pvrvideosink"
```

# GStreamer видеоплеер (Pandaboard)

## Создание pipeline:

```
// Создаем Gstreamer конвейер
GstElement *pipeline = gst_pipeline_new("xvoverlay");

// Создаем элемент playbin2
GstElement *src = gst_element_factory_make("playbin2", "player");

// Создаем видео-вывод
GstElement *bin = gst_bin_new("video_sink_bin");
GstElement *videoSink = gst_element_factory_make("pvrvideosink", NULL);
GstStateChangeReturn sret = gst_element_set_state(videoSink, GST_STATE_READY);
if (sret != GST_STATE_CHANGE_SUCCESS)
{
    gst_element_set_state(videoSink, GST_STATE_NULL);
    gst_object_unref(videoSink);
    return;
}
gst_bin_add(GST_BIN(bin), videoSink);
GstPad *pad = gst_element_get_static_pad(videoSink, "sink");
GstPad *ghostpad = gst_ghost_pad_new("sink", pad);
gst_element_add_pad(bin, ghostpad);
gst_element_set_state(bin, GST_STATE_READY);

// Добавляем элемент playbin2 в конвейер
gst_bin_add_many(GST_BIN(pipeline), src, NULL);

// Указываем использовать в качестве видео-вывода pvrvideosink
g_object_set(src, "video-sink", bin, NULL);
```

# GStreamer видеоплеер (Pandaboard)

## Запуск воспроизведения:

```
// Указываем имя видео файла
g_object_set(G_OBJECT(src), "uri", g_filename_to_uri("/videofile.avi", NULL, NULL), NULL);

// Можно указать в каком окне выводить видео
gst_x_overlay_set_window_handle(GST_X_OVERLAY(videoSink), winId());

// Устанавливаем конвейер в состояние проигрывания
gst_element_set_state(pipeline, GST_STATE_PLAYING);
```

## Пауза:

```
gst_element_set_state(pipeline, GST_STATE_PAUSED);
```

## Остановка воспроизведения:

```
gst_element_set_state(pipeline, GST_STATE_NULL);
```

# GStreamer видео-плеер (Pandaboard)

## Текущий прогресс:

```
gint64 position()
{
    gint64 time = -1;
    GstFormat fmt = GST_FORMAT_TIME;
    gst_element_query_position(pipeline, &fmt, &time);
    return time / GST_MSECOND;
}
```

## Перемотка:

```
void setPosition(gint64 value)
{
    if (!gst_element_seek(pipeline, 1.0, GST_FORMAT_TIME, GST_SEEK_FLAG_FLUSH,
        GST_SEEK_TYPE_SET, value * GST_MSECOND,
        GST_SEEK_TYPE_NONE, GST_CLOCK_TIME_NONE))
    {
        g_error("Couldn't seek.");
    }
}
```

## Спасибо за внимание

**Инновационная  
компания  
Promwad**

**Центр разработок в Москве:**  
4-й Лесной переулок, 4  
Тел: +7(495) 642-82-43  
Эл. адрес: [promwad@promwad.ru](mailto:promwad@promwad.ru)  
Сайт: [www.promwad.ru](http://www.promwad.ru)

**Центр разработок в Минске:**  
ул. Ольшевского, 22, 8-й этаж  
Тел. +375 (17) 312-12-46  
Эл. адрес: [info@promwad.com](mailto:info@promwad.com)  
Сайт: [www.promwad.com](http://www.promwad.com)

