

# **Enabling Hardware Accelerated Playback for Intel® Atom™ Processor N2000/D2000 Series**

Case Study Using MPlayer on Ubuntu 12

July 2012

---

## Document Revision History

<b>Revision</b>	<b>Date</b>	<b>Description</b>
0.01	6/13/12	Initial Version
0.02	7/13/202	Updated based on reviewer comments.

Intel and Intel Atom are trademarks of Intel Corporation in the U.S. and/or other countries.

\* Other names and brands may be claimed as the property of others.

Copyright © 2012, Intel Corporation. All rights reserved.

---

# Table of Contents

1	Executive Summary .....	4
2	Overview of Hardware Accelerated Decode .....	5
3	Build and Installation Overview .....	7
4	Step-by-Step Instructions.....	8
4.1	Cedar Trail Platform Preparation.....	8
4.2	Ubuntu 12 Installation on Cedar Trail Target Platform .....	8
4.3	Build Kernel on Target and Install.....	9
4.4	Install Packages on Target .....	12
4.5	MPlayer compilation on Target .....	12
5	Playback Video and Check CPU Utilization .....	13
6	Run Glxgears and Check CPU Utilization.....	13
7	References .....	14

---

# 1 Executive Summary

---

The Cedar Trail platform is based on the Intel® Atom™ Processor N2000 and D2000 Series processor (formerly called Cedarview) and the Intel® NM10 Express Chipset (formerly called Tiger Point) . This processor series is based on a 32 nm process and features new levels of performance-per-watt opening the door to always-on, always-connected embedded devices.

The N2000/D2000 series processors include an integrated graphics controller with advanced media handling capabilities such as smooth full HD (up to 1080p) video playback along with support for wide range of outputs such as VGA, LVDS, HDMI\*, DP\* and eDP<sup>1</sup>. Media playback is optimum when the video decoding is handled by the video engine in the integrated graphics controller. Using the video engine can significantly reduce the CPU workload and also help improves the quality of playback.

Intel has released Graphics/Media drivers (referred to as PVR-CDV drivers in this document) for the MeeGo open-source distribution for the Cedar Trail platform. The PVR-CDV drivers exploit the video and graphics acceleration capabilities of the media engine. This paper is a case study on integrating these drivers with the Ubuntu distribution and then demonstrating the video acceleration capability of the system with the MPlayer video player. The steps to demonstrate Graphics acceleration using Glxgears demo utility are also provided.

The case study was done on a system with Intel Atom N2800 processor and Intel NM10 Express Chipset. However, the findings are applicable in general to any Cedar Trail system.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by-step guidance, application reference solutions, training, Intel's tool loaner program and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today.

[www.intel.com/embedded/edc](http://www.intel.com/embedded/edc)

---

<sup>1</sup> Version 1.0of PVR-CDV drivers do not support DP and eDP ports; added in v.1.0.1

---

## 2 Overview of Hardware Accelerated Decode

---

To meet the demands of low power and high performance, the Cedarview processor provides dedicated graphics and video decode acceleration hardware for delivering fast video and graphics rendering. Hardware acceleration frees up most of the CPU bandwidth for other time critical tasks. In addition to lowering CPU utilization, hardware acceleration also helps improve the quality of media play back by reducing or eliminating frame drop and reducing Audio/Video synchronization issues. The diagram below provides a simplified overview of the Graphics & Video acceleration stack.

VA-API is a standard API that exposes offloading of video decoding to acceleration hardware. “libVA” is an open source library implementation of the VA-API specification. This library provides access to the hardware used for acceleration of video processing. It enables hardware accelerated video decode at various entry-points (VLD, IDCT, Motion Compensation, deblocking) for the prevailing coding standards today (MPEG-2, MPEG-4 ASP/H.263, MPEG-4 AVC/H.264, and VC-1/WMV3). The VA hardware driver is the Cedarview hardware specific video decode driver.

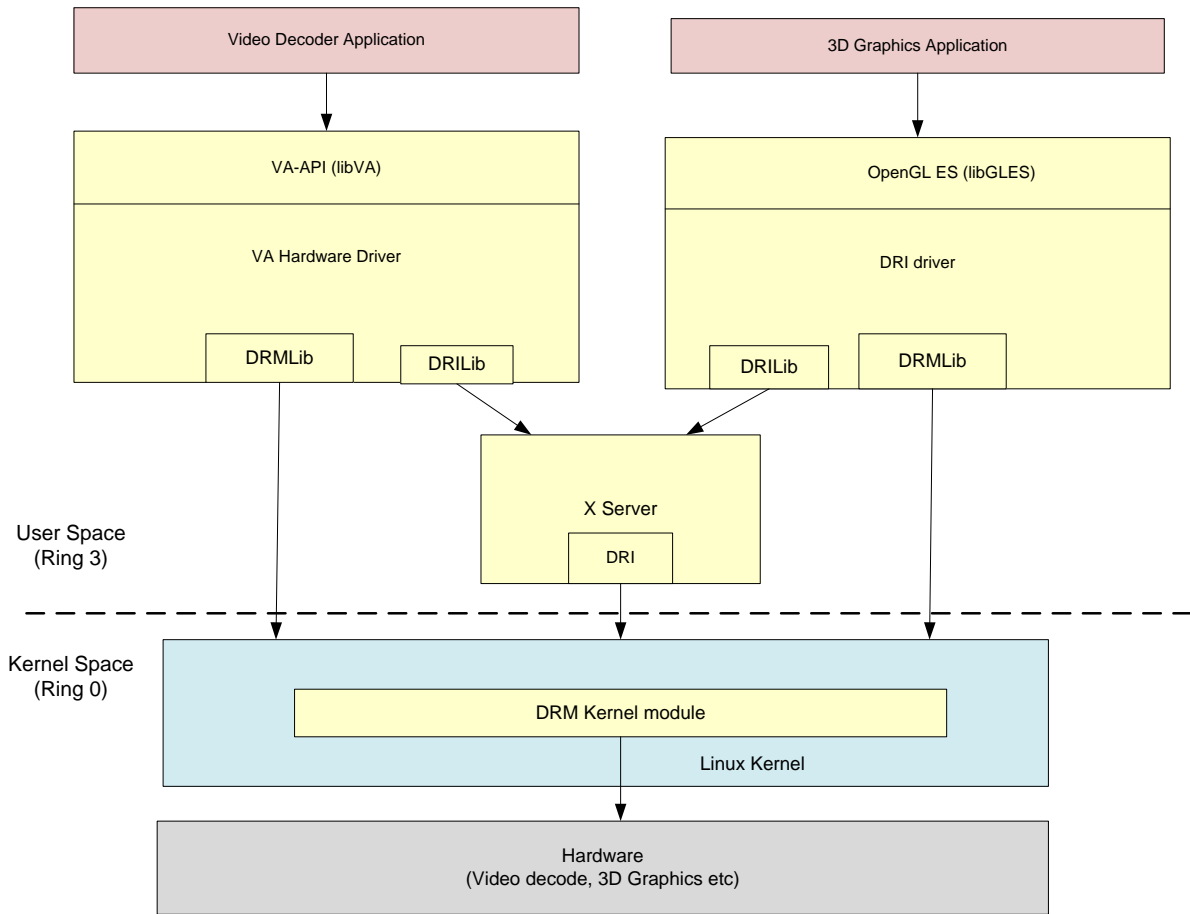
OpenGL ES API is used to expose 3D Graphics acceleration to the Graphics application<sup>2</sup>. libGLES is the open source library implementation of the OpenGL ES API. The DRI driver converts libGLES commands to the graphics hardware accelerator specific commands.

The X server provides the basic services for managing windows displays and input devices. It provides the basic framework for building GUIs.

DRM (Direct Rendering Manager) kernel module is the Kernel space portion of the "Direct Rendering Infrastructure" (DRI). The DRM kernel module provides synchronized access to the graphics controller from the Video and Graphics Drivers and the X server.

---

<sup>2</sup> OpenGL API is partially supported by PVR-CDV drivers. However, OpenGL ES is the recommended API for 3D Graphics applications.



**Figure 1. Overview of Hardware Accelerated Decode**

### 3 Build and Installation Overview

---

The build and installation of the Cedar Trail software stack is done as follows:

- First, build and install the Cedar Trail target kernel with required patches. The target kernel is built by patching the 3.1.0 kernel release from kernel.org with PVR-CDV kernel patches. Note that the target kernel can be natively built either on the target itself or cross-compiled on a host machine. This document provides instructions for both cases.
- Installation of drivers, libraries, and other packages needed for hardware accelerated playback is done next. The graphics driver used in this case study is from the MeeGo v1.2 release for Cedar Trail platform. The driver has the following dependencies.
  - Mesa GL 7.11
  - X Server 1.11
  - Kernel 3.1.0
  - Patched libwsbm 1.1.0+ (provided in release)
  - llibc >= 2.11.90
  - libva >= 1.0.14
  - pixman >= 0.22.0
  - gcc >= 4.5.1
  - libdrm >= 2.4.25
- Build and install MPlayer on Target machine. The MPlayer is patched to use VA-API based hardware acceleration for Video decoding instead of using CPU for video decode.

---

## 4 Step-by-Step Instructions

---

### 4.1 Cedar Trail Platform Preparation

1. Ensure that the board has DDR memory installed (Minimum of 1GB recommended)
2. Connect a VGA or HDMI monitor to the board.  

This study was done with a VGA and HDMI monitor, however other displays supported by the Cedar Trail platform and driver can also be used such as: LVDS, DP or eDP.
3. Connect a USB keyboard and a mouse to the board.
4. Connect a hard disk to the board
5. Connect a DVD player to the board using a SATA cable
6. Connect the power supply to the board.

### 4.2 Ubuntu 12 Installation on Cedar Trail Target Platform

Follow the following steps to install the Linux Ubuntu12, 32 bit (i686) OS on the Cedar Trail target board.

1. Disconnect the LAN cable during the installation because Ubuntu12 will try to go online and update the packages causing the wrong kernel version to be installed on the system.
2. Turn ON the power supply connected to the target board
3. Power-On the board. (On most systems this would typically require momentarily pushing the Power-On button)
4. Insert Ubuntu 12 DVD in the DVD player.
5. Press enter and it starts installing the Ubuntu OS in the hard disk.
6. Select “Install new system” or “upgrade” from the menu. Select “test the Media” before installation
7. Click “next” when the Ubuntu 12 Menu comes
8. Select “English” as a installation language
9. Select “U.S.English” for keyboard
10. Select “Basic Storage Devices” and click “next”
11. Select “Fresh Installation” to start new installation (the old data will be lost)
12. Use the default (localhost.localdomain) host name and click “next”
13. Select your time zone and click “next”
14. Type the root password, store it in a safe place and click “next”
15. Select “Replace existing Linux System” and click “next”
16. Select “Write Changes to Disk”
17. Once Installation is complete, take the DVD from the player and select “reboot”



18. Reconnect the LAN cable.
19. After reboot, you will see the welcome menu. click “Forward”
20. Click forward in the screen “License Information”
21. Create user name and password and click “Forward”
22. Set appropriate date and time in the calendar, click “Forward”
23. Click “Finish”
24. Login and select Default on desktop option  
NOTE: This will not take effect until the system is rebooted.
25. Enable “eth0” to activate the internet connection using Network Manager. It should be a button on the top right of screen.

### 4.3 Build Kernel on Target and Install

Follow these steps only if you intend to natively build the Kernel to be installed on the Cedar Trail target board itself. The build time is approximately 3 hours. Now we have a hard disk with Ubuntu 12 installed for the target based on the steps described in section 4.2.

1. Go to the website:

<http://www.kernel.org/pub/linux/kernel/v3.x/>

2. Look for the following file and download it

linux-3.1.tar.gz

3. Get the patch tarball from your FAE (cdv-gfx-drivers-1.0.1\_bee.tar.bz2)

4. Extract the contents of the patch in home directory

```
tar xjvf cdv-gfx-drivers-1.0.1_bee.tar.bz2
```

5. Untar the kernel in home directory

```
tar xzf linux-3.1.tar.gz
```

It creates a directory named “ linux-3.1” and untars all the files into this directory.

6. Apply the kernel patches by using the patch utility

```
cd linux-3.1
```

```
patch -p1 < ../cdv-gfx-drivers-1.0.1_bee/src/kernel-ttm-clear-high.patch
```

```
patch -p1 < ../cdv-gfx-drivers-1.0.1_bee/cedarview-kernel-v1.0.1_bee.patch
```

7. Edit drivers/staging/Kconfig and add the following line:

```
source “drivers/staging/cdv/Kconfig”
```

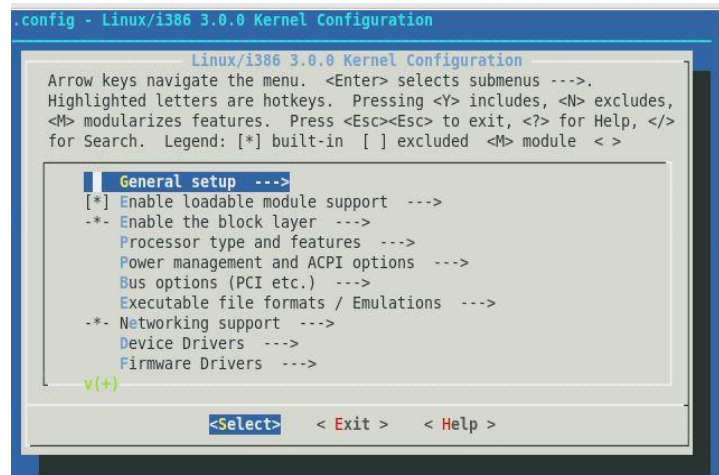
8. Edit drivers/staging/Makefile and add the following line:

```
obj-$(CONFIG_DRM_INTEL_CDV) += cdv/
```

9. We are now ready to build the kernel. Note that the next step will use the config file from the current installations /boot directory which is a PAE image. We will want to make sure that we reset that option with menuconfig.
10. Configure and Compile the kernel

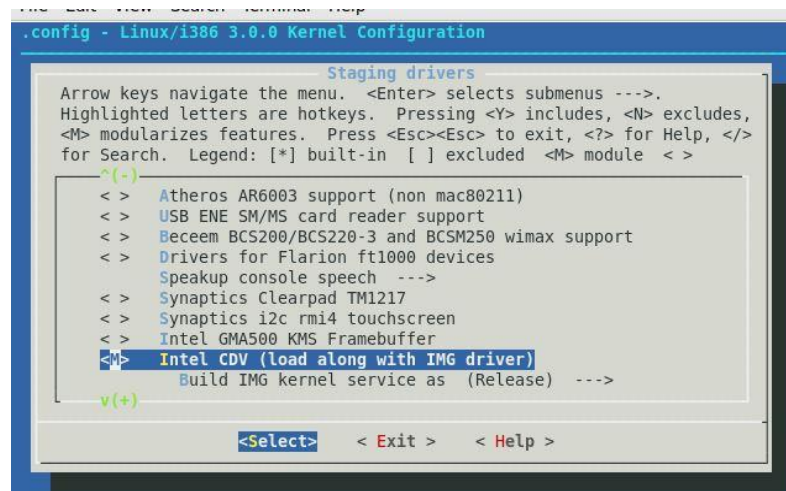
### make menuconfig

The kernel configuration window pops-up and looks like the following:



**Figure 2: Menu Config Utility Screen**

- a) Select “Processor type and features --->” by using the arrow key and enter
- b) Select “High Memory Support (64GB) --->” by using the arrow key and enter
- c) Move the cursor to 4 GB and hit spacebar to select
- d) Exit one level
- e) Select “Device driver --->” by using arrow key and enter
- f) Select “staging drivers --->” by using arrow key and enter
- g) Select “Intel CDV (load along with IMG driver)” and hit the space bar. This will load the release driver.



**Figure 3: Menu config Utility - Intel CDV Driver selection**

23. Exit and save

## 32. Make and Install the Kernel

```
make
```

```
sudo su
```

```
make modules_install
```

```
make install
```

If the make is successful, then the kernel should be there. Check that ramdisk was made in the /boot dir.

## 33. Unpack and install cedarview-libwsbm binary tarball relative to your root filesystem. Assume patch tarball was extracted in /home/gnair:

```
cd /
```

```
tar xvf /home/gnair/cdv-gfx-drivers-1.0.1_bee/bin/cedarview-libwsbm-1.1.0.tar
```

## 34. Install libva and dependencies[get these packages from the web or get the files from FAE]

```
dpkg -i i965-va-driver_1.0.14-1.i386.deb
```

```
dpkg -i libva1_1.0.14-1.i386.deb
```

```
dpkg -i libva-glx1_1.0.14-1.i386.deb
```

```
dpkg -i libva-tpi1_1.0.14-1.i386.deb
```

```
dpkg -i vainfo_1.0.14-1.i386.deb
```

```
dpkg -i libva-dev_1.0.14-1.i386.deb
```

## 35. Unpack and install the appropriate userspace (Xorg, 2D, 3D) Cedarview drivers relative to your root filesystem. While the PowerVR driver is a Mesa replacement for EGL and GL ES, mesa-libGL of the stated version is required for OpenGL operation. Note that development headers are installed as well.

```
tar xvf /home/gnair/cdv-gfx-drivers-1.0.1_bee/bin/cedarview-userspace-v1.0.1_bee.tar
```

## 36. Unpack the PVR VA-API driver for Cedartrail-accelerated H.264, MPEG-2, and VC1 streams:

```
tar xvf /home/gnair/cdv-gfx-drivers-1.0.1_bee/bin/cedarview-vaapi-v1.0.1_bee.tar
```

## 37. Edit /etc/default/grub

- a. Change **GRUB\_DEFAULT**=<new menu number>

Eg: 2 > 0 [ 2 is for previous menu and 0 is for 3.1 kernel]

- b. Type

```
update-grub
```

## 38. Edit /boot/grub/grub.cfg as follows:

1. Add “video=LVDS-1:d” in the kernel line if your display is not LVDS. This is a work-around to an issue that causes driver to believe that LVDS display is connected.
2. Add “vmalloc=256MB” in the kernel line.

## 4.4 Install Packages on Target

There are several dependencies for the Graphics driver and MPlayer. The steps below show how to install the required packages on the target system.

1. Install Xorg Packages as root

```
sudo apt-get install libXi-dev libXfont-dev libXext-dev
sudo apt-get install libdrm-dev
sudo apt-get install libpciaccess-dev pixman-dev
sudo apt-get install expat-dev
sudo apt-get install libva-dev freeglut-dev
sudo apt-get install libXfixes-dev libXv-dev libXpm-dev
sudo apt-get install libX11-dev libXres-dev libXrender-dev libdmx-dev
sudo apt-get install libxkbfile-dev libXinerama-dev libXdamage-dev
sudo apt-get install libXxf86vm-dev mesa-utils
sudo apt-get install libtool autoconf
```

## 4.5 MPlayer Compilation on Target

We are going to build MPlayer with hardware accelerated video decoding in the target system. Before proceeding, ensure that the system has an internet connection.

1. Install yasm and git

```
sudo apt-get install yasm git
```
2. Install some alsa dependencies

```
sudo apt-get install libasound2
```
3. Clone MPlayer git repository

```
git clone git://gitorious.org/vaapi/mplayer.git
```

If the command executes successfully, then MPlayer tree is downloaded in target under “mplayer” directory
4. Configure MPlayer

```
cd mplayer
git checkout -t origin/hwaccel-vaapi
./configure
```

Hit “enter” at “No FFmpeg” checkout prompt

```
cd ffmpeg
git checkout -b ffmpeg-0.6.3 [you can use the latest version tag]
```
5. Get out of ffmpeg directory

```
cd ..
```

39. Now we are in Mplayer directory. Configure MPlayer with options as follows
- ```
./configure --prefix=/usr/local --enable-xv --enable-gl --enable-vaapi --disable-vidpau --disable-mencoder --disable-faad --enable-runtime-cpudetection
```
6. Compile Mplayer
- ```
make  
make install
```

---

## 5 Playback Video and Check CPU Utilization

---

1. Ensure that Mplayer is built with vaapi enabled  
**mplayer -vo help**  
“vaapi” should be among the list of video output drivers listed by the above command.
2. Play the video with Mplayer using hardware acceleration.  
**mplayer -vo vaapi -va vaapi -ao alsa <example\_video\_clip>.mp4**
3. Check that CPU utilization using “top” command  
Open another terminal and type  
**top**  
  
Max CPU utilization for Mplayer application should be about 10% (Wwith 1080p videos)

---

## 6 Run Glxgears and Check CPU Utilization

---

### **glxinfo|grep direct**

The string “direct rendering: Yes” should be output, indicating that Graphics hardware acceleration is enabled

4. Launch the glxgears demo  
**glxgears -fullscreen**  
A full screen animation of three rotating gears would start to play  
Hit “Esc” key to stop glxgears
5. Launch the glxgears demo (non-Full screen)  
**glxgears**  
  
The glxgears animation will start in a new window.
6. Check that CPU utilization using “top” command  
Open another terminal and type  
**top**  
Max CPU utilization for glxgears application should be less than 2 %.

## 7 References

---

- 1) VA-API,  
<http://www.freedesktop.org/wiki/Software/vaapi>
- 2) MeeGo. The MeeGo project site hosts the Cedar Trail Graphics Drivers and Kernel patches (Released as part of the Cedar Trail Netbook platform). Refer to this site to check for any updates.  
<<https://meego.com/>>

<b>Acronyms</b>	
API	Application Programming Interface
AVC	Audio Video Coding
DP	DisplayPort
eDP	Embedded Display Port
HDMI	High Definition Multimedia Interface
IDCT	Inverse Discrete Cosine Transform
LVDS	Low Voltage Differential Signaling
HD	High Definition
VA-API	Video Acceleration API
VGA	Video Graphics Array
VLD	Variable Length Decoding
WMV3	FourCC notation for Windows Media Video 9

**Table 1: Acronym Table**