

White Paper

Lim Siew Hoon

Graphics Software
Engineer

Intel Corporation

Kumaran Kalaiyappan

Graphics Software
Engineer

Intel Corporation

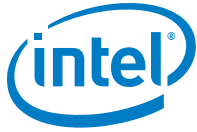
Tay Boon Wooi

Graphics Software
Engineer

Intel Corporation

Video Encoding on Intel[®] Atom[™] Processor E38XX Series using Intel[®] EMGD and GStreamer

April, 2014

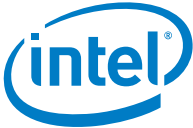


Executive Summary

Video encoding is a feature to compress raw data input into one of the compressed format for storage, transmission and transcoding purposes.

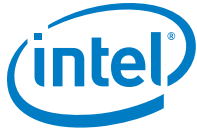
This paper describes the support for hardware accelerated video encoding on Intel® Embedded Media Graphics Driver (EMGD) utilizing GStreamer framework on Intel® Atom™ Processor E38XX Series on Linux distributions.

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. http://www.intel.com/p/en_US/embedded.



Contents

- Introduction.....4
- Hardware Accelerated Encode Support.....5
- Software Stack.....6
- Sample Commands.....8
- Conclusion.....10
- References.....10



Introduction

Hardware accelerated video encoding is a significant feature for embedded market segments. This feature allows compression of raw video data from cameras and network to be compressed for storage and transmission. In addition, this feature together with the hardware accelerated video decode feature provides hardware accelerated transcoding capabilities. With hardware acceleration, it is possible to achieve full definition video compression capability with much less CPU usage.

Intel® Atom™ Processor E38XX Series supports hardware accelerated video encoding based on AVC (H.264) and MPEG2. This feature is supported by Intel® EMGD on Linux utilizing the open source libva-intel-driver and gstreamer-vaapi software components.

From an application point of view, the gstreamer-vaapi software components are provided as GStreamer plugins that plug into the GStreamer 1.0 framework. The usage of GStreamer 1.0 provides the advantage to use the various available open source GStreamer plugins.

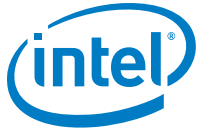


Hardware Accelerated Encode Support

Intel® Atom™ Processor E38XX Series supports hardware accelerated video encoding based on AVC (H.264) and MPEG2.

For H.264, the profiles supported are Constrained Baseline, Main and High profiles, 8 bit only up to Level 4.1 up to 40 Mbps bitstream. There is no support for Baseline, Extended, and High-10 Profiles. B frames are supported.

For MPEG2, Main and Simple profiles are supported.



Software Stack

The software stack at the application level is based on GStreamer 1.0. The usage of GStreamer provides the advantage and flexibility to use the dictionary of readily available GStreamer plugins. The plugins are provided by gstreamer-vaapi.

This component provides encode, decode, display, and post processing plugins. These are open source components which the end user can modify according to their needs.

The installation instructions and versions of the software components to use can be obtained from the user guide, *Linux_UsersGuide.pdf* located in:

https://linuxlink.timesys.com/login/?timesys_redirect=/intel/linux/baytrail

The gstreamer-vaapi plugins are available from:

<http://www.freedesktop.org/software/vaapi/releases/gstreamer-vaapi/>

The gstreamer-vaapi utilizes the VA-APIs (Video Acceleration API) from:

<http://www.freedesktop.org/wiki/Software/vaapi/>

The user space video driver used is libva-intel-driver from:

<http://cgit.freedesktop.org/vaapi/intel-driver>

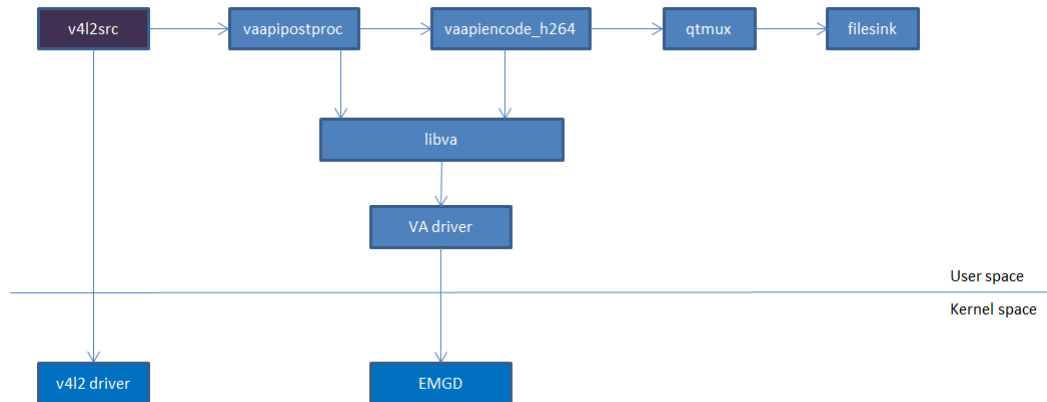
The kernel driver (drm) used is Intel® EMGD from:

https://linuxlink.timesys.com/login/?timesys_redirect=/intel/linux/baytrail

The software stack for encoding from camera is as shown in figure 1 below.

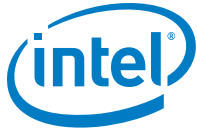


Figure 1. Software stack for hardware accelerated video encode from camera



The description of the key plug-ins used is as below:

- `v4l2src` is a GStreamer plugin to read data from the connected USB cameras. For files, this can be replaced by `filesrc`.
- `vaapipostproc` is a post processing plugin provided by `gststreamer-vaapi` that is used to convert the incoming raw data to NV12 tiled format (as the encode engine accepts only NV12 tiled formats) and load the data into a VA Surface.
- `vaapiencode_h264` is the H.264 encode plugin provided by `gststreamer-vaapi` that is used to hardware accelerate the H.264 encode. For MPEG2, the plugin used is `vaapiencode_mpeg2`.



Sample Commands

1. Encode video from camera.

MPEG2 encode:

```
gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=1800 !  
video/x-raw,format=I420,width=640,height=480 !  
vaapipostproc ! queue ! vaapiencode_mpeg2 ! mpegtsmux !  
filesink location=test2.mpg
```

H264 encode:

```
gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=1800 !  
video/x-raw,format=I420,width=640,height=480 !  
vaapipostproc ! queue ! vaapiencode_h264 ! qtmux ! filesink  
location=test2.mp4
```

2. Encode video from camera and display the incoming video.

MPEG2 encode:

```
gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=1800 !  
video/x-raw,format=I420,width=640,height=480,  
framerate=30/1 ! tee name=enc ! queue ! vaapipostproc ! queue !  
vaapiencode_mpeg2 ! mpegtsmux ! filesink location=test2.mpg  
enc. ! queue ! vaapisink
```

H264 encode:

```
gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=1800 !  
video/x-raw,format=I420,width=640,height=480,  
framerate=30/1 ! tee name=enc ! queue ! vaapipostproc ! queue !  
vaapiencode_h264 ! qtmux ! filesink location=test2.mp4 enc. !  
queue ! vaapisink
```

3. Encode video from raw file.

MPEG2 encode:

```
gst-launch-1.0 filesrc blocksize=10000000 location=infile.yuv !  
videoparse format=i420 height=1080 width=1920  
framerate=30/1 ! vaapipostproc ! queue ! vaapiencode_mpeg2 !  
mpegtsmux ! filesink location=test2.mpg
```

H264 encode:

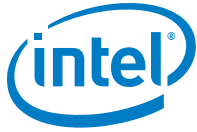
```
gst-launch-1.0 filesrc blocksize=10000000 location=infile.yuv !  
videoparse format=i420 height=1080 width=1920  
framerate=30/1 ! vaapipostproc ! queue ! vaapiencode_h264 !  
qtmux ! filesink location=test2.mp4
```




4. Encode video from raw file and display the video.

H264 encode:

```
gst-launch-1.0 filesrc block size=1000000 location=infile.yuv !  
videoparse format=i420 height=1080 width=1920  
framerate=30/1 ! tee name=enc ! vaapipostproc ! queue !  
vaapiencode_h264 ! qtmux ! filesink location=test2.mp4 enc. !  
queue ! vaapisink
```



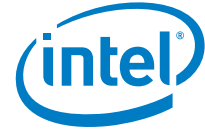
Conclusion

This paper described how the hardware accelerated video encoding feature in the Intel® Atom™ Processor E38XX Series can be exercised on Linux utilizing Intel® EMGD, the open source libva-intel-driver, and gstreamer-vaapi software components.

Additional features will be implemented in the future including the buffer sharing implementation between the camera and graphic component to encode without buffer copy.

References

1. GStreamer: Open source multimedia framework,
<<http://gstreamer.freedesktop.org/>>
2. Video Acceleration API,
<<http://freedesktop.org/wiki/Software/vaapi>>
3. Intel® Embedded Media and Graphics Driver
<https://linuxlink.timesys.com/login/?timesys_redirect=/intel/linux/baytrail>
4. Intel® OpenSource HD Graphics Programmer's Reference Manual (PRM)
Volume 2 Part 3: Multi-Format Transcoder – MFX (Ivy Bridge)
<https://01.org/linuxgraphics/sites/default/files/documentation/ivb_ihd_os_vol2_part3.pdf>



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. http://www.intel.com/p/en_US/embedded.

Authors

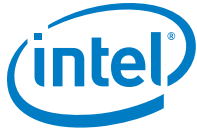
Lim Siew Hoon is a Graphics Software Engineer at Intel Corporation.

Kumaran Kalaiyappan is a Graphics Software Engineer at Intel Corporation.

Tay Boon Wooi is a Graphics Software Engineer at Intel Corporation.

Acronyms

EMGD Intel® Embedded Media Graphics Driver



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice.

This paper is for informational purposes only. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2014 Intel Corporation. All rights reserved.