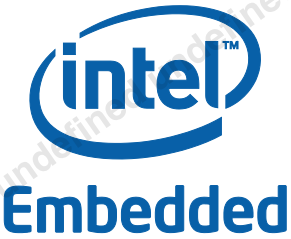


White Paper
Raynald Lim
Software Engineer
Intel Corporation
Andrey Larionov
Software Engineer
Intel Corporation
Alan Previn Teres Alexis
Software Architect
Intel Corporation
P. Kumaran Kalaiyappan
Software Engineer
Intel Corporation

Video Encode Acceleration via Intel® Media SDK Framework

for Intel® Atom™ Processor E6xx Series

May, 2011

325559-001

Executive Summary

Intel® EMGD for Intel® Atom™ Processor E6xx Series System-on-Chip supports video encode acceleration via the Intel® Media Software Development Kit (Intel® Media SDK) framework on Microsoft Windows* 7 and Microsoft Windows* Embedded Standard 7 operating systems. This application note details the encode software stack (including the Windows Display Driver Model (WDDM) driver and middleware), capability, tool set, and environment required to make full use of the platform GPU accelerated encode capability.

Intel® EMGD uses the Intel Media SDK framework to provide a standardized encode interface and an entry-point to unlock the full encode capability supported by the driver and hardware. This ensures that applications written for the Intel Media SDK framework will work across multiple platforms with little-to-no-enabling effort on the part of the user.

Intel® EMGD uses the Intel® Media SDK framework to provide a standardized encode interface and an entry point to unlock the full encode capability supported by the driver and hardware.

Aside from standardizing the encode acceleration interface, offloading encoding workload to the GPU includes these benefits:

- Reduction in CPU utilization.
- Speed. Fixed function encode engine is optimized for speed.

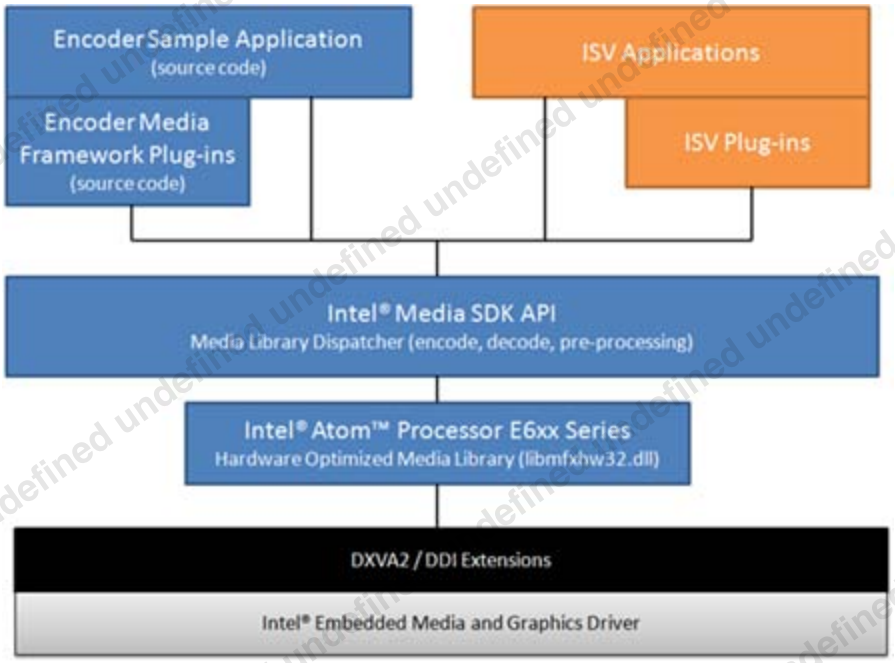
Contents

Background

Intel Media SDK framework was originally developed for Intel® HD Graphics media block to enable Intel® Quick Sync hardware accelerated transcoding capability. The framework is adopted for Intel® EMGD to provide a cohesive experience and a standardized interface that exposes video encode acceleration.

An overview of the encode stack is illustrated in Figure 1.

Figure 1. Intel® EMGD Encode Stack



Many Intel Media SDK resources, including the Media SDK version 2.0 package and future packages, can be downloaded from:

http://software.intel.com/en-us/articles/media/

Components in blue in Figure 1 are part of Intel Media SDK package. Intel® Atom™ Processor E6xx Series hardware optimized media library is installed as part of Intel® EMGD setup process. DirectX Video Acceleration 2.0 (DXVA2) and Device Driver Interface (DDI) extensions are part of the Microsoft Windows 7 and Windows Embedded Standard 7 software stack.

Note: Intel® EMGD currently supports encode functionality via Intel Media SDK framework. Future work will include decode and transcode support via Intel Media SDK framework. Hardware accelerated decode is supported by Intel® EMGD via direct calls to DXVA2 Application Programming Interface (API).

Operating System Support

Hardware accelerated video encode via Intel Media SDK framework is supported only on Microsoft Windows 7 and Windows Embedded Standard 7.

CODEC Support

Intel® Atom™ Processor E6xx Series encode engine support a variety of formats (see White Paper: *Video Encoding Accelerator Solution for Intel® Atom™ Processor E6xx Series*). The Intel® EMGD Windows 7 and Windows Embedded Standard 7 encode stack, however, support a limited subset of these format as illustrated in Table 1.

Table 1. Intel® Atom™ E6xx Supported Encoder Format, Profiles and Levels

Codec	Profile	Level	Max Bit Rate (bps)	Typical Picture and Frame Rate
H.264	BP	1b	128K	QCIF@15fps
H.264	BP	L1.1	192K	QCIF@30fps
H.264	BP	L1.2	384K	CIF@15fps or QVGA@20fps
H.264	BP	L2.0	2M	CIF@30fps or QVGA@30fps
H.264	BP	L3.0	10M	720*480@30fps or 720*576@25fps or VGA@30fps
H.264	MP(*1)	L3.1	14M	1280*720@30fps

*1. The stream will use only tools common to the Baseline Profile that are also available on Main Profile.

Intel® Atom™ Processor E6xx Series real-time encode engine outputs an H.264 elementary stream that could be muxed into a variety of container formats made possible by the Intel Media SDK Plug-ins. As of version 2.0, Intel Media SDK supports multiplexing to two container formats, MPEG2TS and MPEG4 (MP4).

Graphic Driver Supported

The full encode engine capability is unlocked using Intel® EMGD. Intel® EMGD for Windows 7 adopts the Windows Display Driver Model (WDDM) architecture. The driver's encode software stack has both user mode and kernel mode components; the latter interfaces with the hardware encode engine during the encoding process while the former implements the Direct 3D 9 (D3D9) Device Driver Interfaces (DDI).

There are currently no standardized interfaces defined for hardware accelerated video encode. Intel® EMGD utilizes a proprietary front-end design that allows DXVA2-like encode processes to communicate with the underlying driver using existing D3D9 DDI.

The core encode engine allows output of an H.264 coded elementary stream using the Baseline Profile (BP) tool set, which implies no B-frame coding. High Definition (HD) 1280x720 resolution coding is supported at 30 fps. The encoder expects a raw 4:2:0 planar YUV (FourCC: NV12) frame input. All input of other formats needs to be converted (real-time or otherwise) to the supported NV12 format.

Note: Intel Media SDK 3.0 DxShow H.264 Encoder plug-in supports native FourCC YUY2 input and automatically converts to the supported pixel format (NV12) of Intel® EMGD encode engine. The pixel conversion process uses fast Intel® Integrated Performance Primitives (IPP) libraries or if supported by the driver, the DXVA2 VideoProcessBlt interface.

Media Framework and Middleware

The application stack consists of the Intel Media SDK framework as well as supporting OS stack libraries including DXVA2 and D3D DDI.

Intel® Media SDK Framework

The Intel® Media Software Development Kit (Intel® Media SDK) is a collection of software libraries that expose the media acceleration interfaces to application programs. The API supports a wide range of encoding, decoding, and pixel preprocessing capabilities for a wide range of Intel platforms. The Intel Media SDK also provides a rich set of sample applications that media developers can utilize to begin using the API right away.

Note: Intel® EMGD currently supports encode functionality through the Intel Media SDK framework. Future work will include decode, transcode, and video pre-processing support via Intel Media SDK framework.

The Intel Media SDK programming interfaces are exposed to applications via the Media Dispatching layer. The Dispatcher is a static library that is responsible for exposing the entry points for the encoding, decoding, and video preprocessing routines. The Media Dispatching layer is also responsible for detecting and loading the appropriate implementation library for the client machine. By default, dispatcher tries to load and use the hardware-optimized media library. If it is not possible, the dispatcher loads the software-optimized media library.

Intel® Atom™ Processor E6xx Series hardware-optimized media library is installed as part of the Intel® EMGD setup process. The hardware-optimized media library has a fallback option that utilizes the software-optimized media library if conditions for hardware accelerated encoding are not met, usually related to conflicting encode parameters or an unsupported platform.

There is a limited set of variables that the Intel® EMGD encode stack exposes through the Media SDK library to the end user. These include:

- Profiles and levels of H.264
- Bit rate (see Table 1 for supported bit rate)
- Rate control mode (CBR or VBR)
- Resolution (width and height in pixel)
- Frame rate (intended frame rate in fps)

The Intel Media SDK sample application source and binary reside in the src and bin subdirectories respectively within the root Intel Media SDK directory. The `sample_encode.exe` sample application supported by Intel® EMGD requires raw YUV input of NV12 or YV12 format. YV12 format is automatically converted to NV12 using fast IPP libraries. The conversion process does have an impact on performance.

Note: Rate control mode support is not exposed via sample_encode.exe, however, it uses the default CBR mode. Rate control mode can be tweaked otherwise using Media Framework plug-ins. Target usage option (-u) is not supported on the Intel® Atom™ Processor E6xx Series.

Media Framework plug-ins are available as part of the Intel Media SDK package. Both Media Foundation (MF) and DirectShow (DxShow) framework Plug-ins are optional installs. Plug-ins can be invoked and used within a media application by calling an existing MF or DxShow framework API, or by loading a filter graph using tools such as GraphEdit or TopoEdit for DxShow and MF framework respectively.

The list of Intel Media SDK media framework plug-ins include splitters, transform filters (encoder/decoder), and muxers. Source code is also provided within the samples plug-ins subdirectory. A sample use case is provided in the following section.

Note: Intel recommends using the DxShow framework plug-in with Intel® EMGD. Interoperability with MF framework plug-ins with Intel® EMGD has not been tested.

DXVA2 and D3D DDI

DXVA2 and D3D9 DDI are the underlying infrastructures that Intel Media SDK interacts with. These are Microsoft provided OS stacks that come pre-installed with every Windows 7 image.

DXVA2 is an API with corresponding DDI to support hardware accelerated decode and video post processing. Intel® EMGD uses a proprietary front-end design that allows DXVA2-like encode processes to communicate with the underlying driver using existing D3D9 DDI.

For an in-depth explanation of DXVA2, visit the MSDN site at http://msdn.microsoft.com/en-us/library/cc307941(VS.85).aspx

For an in-depth explanation of D3D9 DDI, visit the MSDN site at http://msdn.microsoft.com/en-us/library/ff552927(v=VS.85).aspx

Exercising Hardware Video Encode Accelerator

The examples below require Intel® EMGD to be installed on the Intel® Atom™ Processor E6xx Series platform on Windows 7 or WES 7 operating systems.

Using the Sample Encoder Application

To use the Intel Media SDK sample encode application:

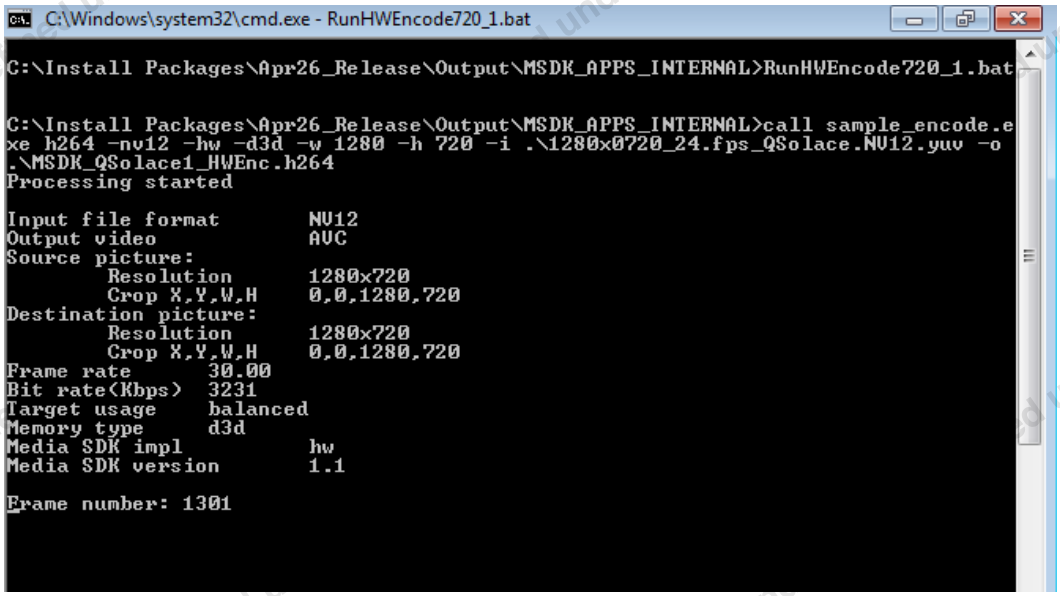
1. Download and install Intel Media SDK 2.0 (or higher) from http://software.intel.com/en-us/articles/media/
2. In the Intel Media SDK root directory (default is `C:\Program Files\Intel\Media SDK\2.0.12.XXXXX\`), go to the `..\bin\win32` subdirectory and search for `sample_encode.exe`.
3. Open a console command prompt window in administrator mode and go to the subdirectory mentioned in step 2.
4. Invoke `sample_encode.exe` using the example below:

   sample_encode.exe h264 –nv12 –d3d –hw –i InputNV12File.yuv –o outputH264ElementaryStream.h264 –w 1280 –h 720

Note: If no bit rate is chosen a default bit rate is calculated based on the selected resolution (for example, for 1280x720 resolution default bit rate is 14 Mb).

After the encode process begins the Frame number field increments every 100 frames as depicted in Figure 2.

Figure 2. Intel® EMGD Hardware Accelerated Video Encoding in Progress



5. Ensure that the input parameters displayed on screen are correct and that the `Media SDK impl` field reads `hw`, indicating hardware implementation instead of `sw`, indicating software.
6. After the video encoding process is completed, the resulting coded H.264 elementary stream is placed within the path defined by the –o option.

Note: There are a limited number of players that support H.264 elementary stream playback. To view the encoded stream, Intel recommends downloading and using VLC Media Player from http://www.videolan.org/. Alternatively, mux the elementary stream to any widely supported container format using a third-party tool.

Using Media Framework Plug-ins

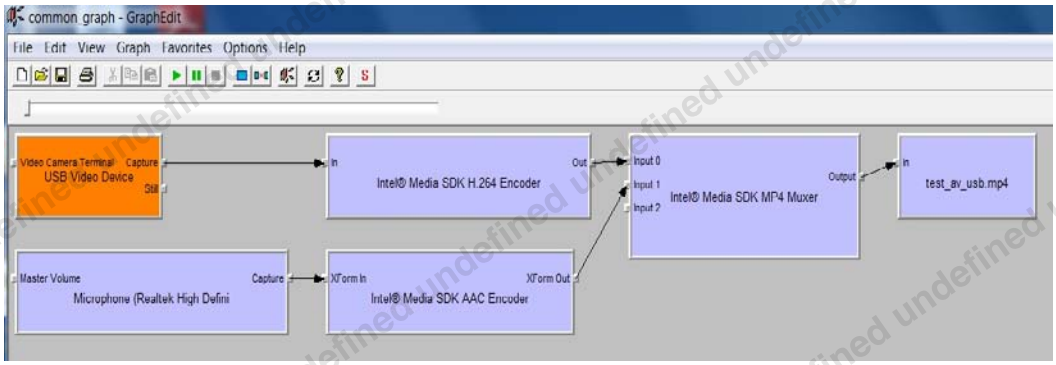
The Intel Media SDK package includes MF and DxShow plug-ins. This example shows how to build a DxShow video encoding filter graph that takes a USB camera output as source input, encodes both the video and audio components of the input, and muxes the resulting bitstream to an MPEG (MP4) container. Note that audio encoding is implemented in software.

1. Download and install Intel Media SDK 2.0 (or higher). Ensure that the optional DxShow plug-in components are selected for installation.
   http://software.intel.com/en-us/articles/media/
2. Install GraphEdit tool available as part of Microsoft Windows SDK.

3. Select Graph, Insert Filter and insert each filter component shown in Figure 3.

Note: USB Video Device and Microphone can be found under Video Capture Sources and Audio Capture Sources respectively. Intel Media SDK filters can be found under DirectShow Filters. The resulting output sink filter is also found under DirectShow Filters as `File Writer`.

Figure 3. Filter Graph of Hardware Accelerated Video Encoding Using USB Camera as Source



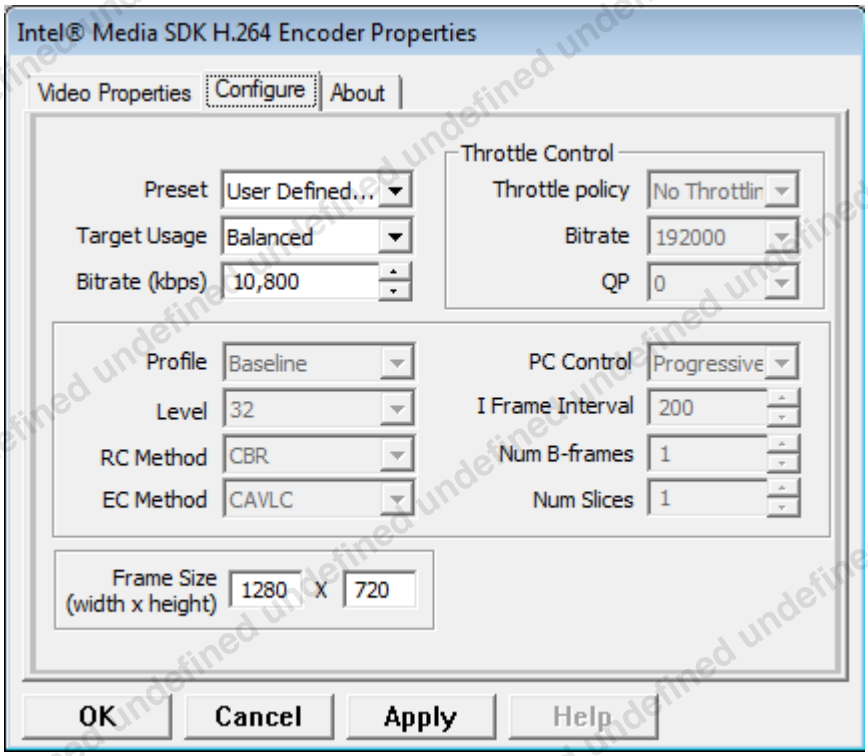
Note: Most USB cameras support output of pixel format YUY2. The Intel Media SDK H.264 Encoder transform plug-in supports only NV12 pixel format. While connecting these two filters, the negotiation process returns an error because the two filters could not agree on a common pixel format. To resolve this:

- If you are using Intel Media SDK 2.0, add supporting code to modify the Intel Media SDK H.264 Encoder transform filter to support YUY2 input and the conversion of YUY2 pixel format to NV12.
- Download and install the Intel Media SDK 3.0 package.

4. Configure Intel Media SDK H.264 Encoder accordingly with these limitations of the Intel® Atom™ Processor E6xx Series:
   a. Profile: Only Baseline is supported
   b. Level: Refer to Table 1 for limitation
   c. RC Method: CBR or VBR
   d. EC Method: Only CAVLC
   e. Num B-frames: Must be 1 to indicate no B-frames.

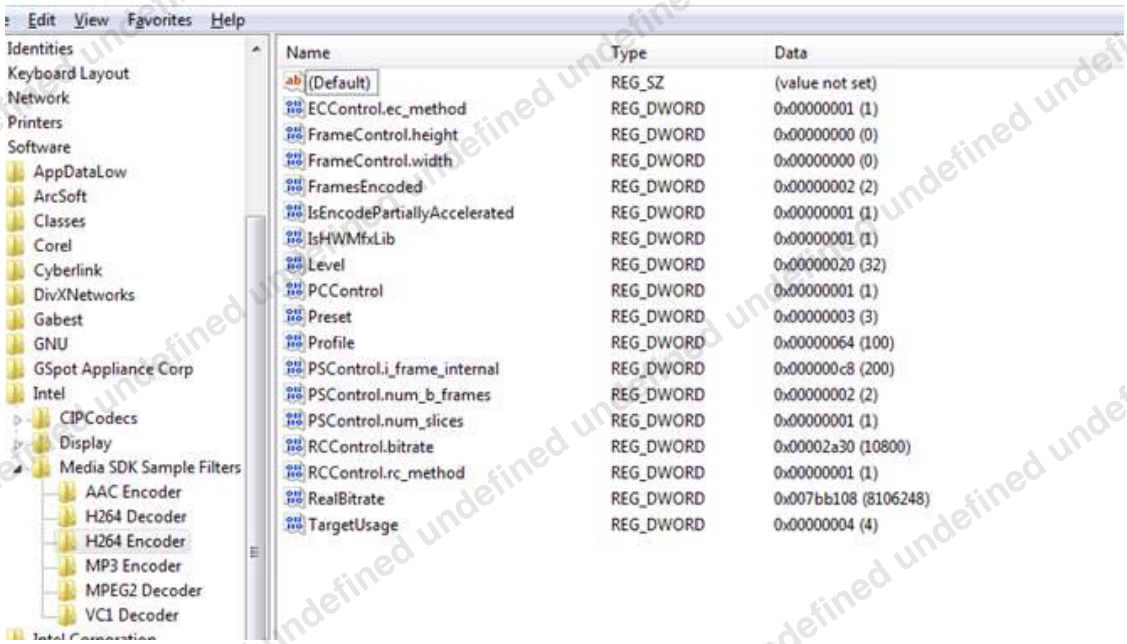
   Figure 4 shows a typical configuration supported on the Intel® Atom™ Processor E6xx Series platform.

Figure 4. Example Encoder Configuration Supported on Intel® Atom™ Processor E6xx Series



5. The Intel Media SDK H.264 Encoder configuration can be changed using the registry editor. All key entries are located at HKEY_CURRENT_USER ->Software ->Intel->Media SDK Sample Filters as depicted in Figure 5.

Figure 5. Registry Entries for Intel® Media SDK H.264 Encoder



6. Click **Play** to initiate the encoding process. The resulting muxed stream is stored according to the path defined by the File Writer filter.

Note: To verify that hardware accelerated video encode is used instead of software accelerated video encode, check that `IsHWMfxLib` is set and `IsEncodePartiallyAccelerated` is cleared.

Note: The encode engine has a different firmware for each rate control (RC) mode. Because the RC mode can be determined only at the onset of encoding the first frame, there would be considerable latency with having to load the RC mode firmware. This latency causes video and audio de-synchronization at early stages of encoding the first few frames that would eventually work itself out. This is a known limitation and a workaround will be added for future releases.

Summary

Hardware accelerated video encoding can achieve 2x improvement in performance over software video encode on the Intel® Atom™ Processor E6xx Series platform. Using a standardized Intel Media SDK interface ensures uniform experience and cross platform conformity for all media applications written for the framework. Intel® EMGD unlocks the full potential of the Intel® Atom™ Processor E6xx Series, providing a seamless experience and performance. Future work will integrate and expose many more Intel Media SDK features.

References

1. Media Developer Guide v1.0
   http://software.intel.com/file/33987
2. Intel® Media SDK Package
   http://software.intel.com/en-us/articles/media/
3. Video Encoding Accelerator Solution for Intel® Atom™ Processor E6xx Series
   http://download.intel.com/design/intarch/PAPERS/324328.pdf

Authors

Raynald Lim is a Graphic Software Engineer with Intel Embedded and Communication Group (ECG).

Andrey Larionov is a Graphic Software Engineer with Intel Embedded and Communication Group (ECG).

Alan Previn Teres Alexis is a Graphic Software Architect with Intel Embedded and Communication Group (ECG).

Periyakaruppan Kumaran Kalaiyappan is a Graphic Software Engineer with Intel Embedded and Communication Group (ECG).

Acronyms

API	Application Programming Interface
CBR	Constant Bit-Rate
CPU	Central Processing Unit
D3D	Direct 3D
DDI	Device Driver Interface
DXVA	DirectX Video Acceleration
EMGD	Embedded Media and Graphic Driver
GPU	Graphics Processing Unit
SDK	Software Development Kit
VBR	Variable Bit-Rate
WDDM	Windows Display Driver Model

§