



DaVinci dm355 Digital Video Software Development Kit (DVSDK)

DVSDK 2.00.00.22

Release Notes Tue May 26 15:05:49 PDT 2009

Document Revision: i30

Introduction

This DVSDK Software release coupled with the Demonstration Version of Montavista Linux Pro v5.0 for DaVinci gives developers the ability to evaluate the hardware and software capabilities of the DaVinci platform. In conjunction with the demonstration version of Montavista Linux, developers will be able to experience the ARM side Linux programming environment and easily utilize pre-built codec executables for performance capability evaluation of the 64x+ DSP. Developers will be able to evaluate the ARM Linux programming environment and easily utilize the powerful hardware support of the dm355 SoC for JPEG and MPEG4 codecs.

Notice

This release of the DVSDK does not install the DM6446 and DM6467 Codec Servers when the DVSDK is installed. This is a change from previous releases and allows the DVSDK to be distributed anonymously. The CODEC Servers may be downloaded from the same site as the corresponding DVSDK.

Before attempting to build this DVSDK, please follow the additional steps listed here for installing this DVSDK.

1. Install DVSDK as described in the Getting Started Guide. The CODECS Servers for your platform must then be installed in the same directory as the DVSDK (Usually \$HOME/2_00_00_22)
2. Download and install the CODEC Servers for your platform (dm6446_codecs_setuplinux_2_00_00_22.bin or dm6467_codecs_setuplinux_2_00_00_22.bin). Remember to install the Servers in the same directory as you installed the DVSDK. If you decide to install them elsewhere, please remember to update the location in Rules.make (CODEC_INSTALL_DIR=).
3. Follow the rest of the instructions in the GSG

Example command line setup would be:

```
<Download Location>/dvsdk_setuplinux_2_00_00_22.bin --mode console --prefix $HOME
```

Then either:

```
<Download Location>/dm6446_codecs_setuplinux_2_00_00_22.bin --mode console --prefix $HOME OR  
<Download Location>/dm6467_codecs_setuplinux_2_00_00_22.bin --mode console --prefix $HOME
```

Please note, since DM355 does not use CODEC Servers, no changes are required for this platform.

This document is divided into the following sections:

- [Documentation](#)
- [What's New](#)
- [Installation and Usage](#)
- [Upgrade and Compatibility Information](#)
- [Host Support](#)
- [Dependencies](#)
- [Device Support](#)
- [Validation Information](#)
- [Upgrade Information](#)
- [Known Issues](#)
- [Versioning](#)
- [Technical Support and Product Updates](#)

Documentation

- Latest up to the minute DaVinci information and updates may be found on [DaVinci Technology Developers Wiki](#), [TI eXpressDSP Software Wiki](#) or [DVSDK 2.0 Errata](#).
- [DVEVM Getting Started Guide](#) - Hardware and software overview, including how to run demos, install software, and build the demos.
- [Bios Utilities](#).
- [Codec Engine](#).
- [Contiguous Memory Allocator](#).
- [DMAI API](#) Davinci Multimedia Application Interface API.
- [Decode Demo Information](#)
- [Encode Demo Information](#)
- [Encodedecode Information](#)
- [Framework Components](#)
- [XDAIS documentation](#)
- [Linux Kernel documentation](#)
- [RTSC Codec and Server Package Wizards](#)
- [Quilt Installation for MVL 5.0 Demo Version](#)

Adding Quilt To Montavista Linux 5.0 Demonstration Version

The Quilt software package is used to manage changes to the Linux kernel. Future changes to the kernel will be distributed as "patches" which can be added or removed to the base kernel using Quilt. Please see the instructions on Quilt installation [here](#).

Adding GDB to Montavista Linux 5.0 Demonstration Version

The GDB Debugger is not provided with this release. Customers wishing to obtain a GDB Debugger have several options:

- Purchase the DVSBP product from TI. This provides access to a fully supported, pre-built GDB (and other tools) for the OMAP-L137 device from Monta Vista.
- Download the appropriate GDB and related component sources from www.gnu.org at no charge and build them.
- Download the G++ tools for ARM GNU/LINUX from www.codesourcery.com. Both no charge evaluation versions and fully supported versions are available. Please note that developers should only use the GDB from these tools and NOT attempt to replace the gcc compiler version in the DVSDK as this may result in build issues.

Instructions for building GDB server may be found at http://wiki.davincidsp.com/index.php?title=Debugging_Linux_Application_on_OMAP-L_137.

What's New

Following have been updated since the last release

- LSP 2.00 (2.6.18 kernel)
- Separation of Servers and DVSDK
- Unified DVTB
- Demo Updates
- MFP Components v2.23.01
- Demo MontaVista Tools v5.0

The DVSDK contains the following components.

biosutils_1_01_00	Bios Utilities.
cg_xml_2_12_00	Perl scripts used to process the XML files that come from the TI code generation tools
linuxutils_2_23_01	Contiguous memory allocator for Linux
clips	Demo A/V clips. Please note, A/V clips are not populated until after performing the steps under "Installing the A/V Demo Files" of the GSG
ceutils_1_06	CE utils is a collection of utilities that aid CodecEngine usability.
codec_engine_2_23_01	The Codec Engine provides a framework for creating and interacting with multimedia codecs
dm355_codecs_1_13_000	Codecs for both encoding and decoding H.264 and decoding MPEG2.
dvsdk_demos_2_00_00_07/dm355	

"DaVinci DM355 DVSDK 2.00 Release Notes"

	Demo applications that illustrate usage of Linux drivers and codecs
dmai_1_20_00_06	Davinci Multimedia Application Interface
dsplink_1_61_03	Foundation software for the inter-processor communication across the GPP-DSP boundary.
dvtb_4_00_08	Digital Video Test Bench (DVTB) an interactive application for evaluating codec performance
edma3_ild_1_05_00	Drivers for submitting and synchronizing with EDMA3 based DMA transfers.
framework_components_2_23_01	Framework Components, a collection of framework-independent utility libraries which other software frameworks can build upon.
kernel_binaries	Platform/LSP specific prebuilt .ko files
xdais_6_23	xDAIS product contains the DSP Algorithm Interface Standard specification and related documentation and examples.
PSP_02_00_00_140	Linux Kernel Files Device and Board boot support: a low-level initial bootloader (UBL) and u-boot 1.2.0, with support for booting from NAND and Ethernet.

In addition, a complete Linux development and runtime environment is provided. Each of the components listed below is packaged in a separate installer, described in the Getting Started Guide (GSG).

- Montavista Linux Tools: cross-compiler tools for ARM
- Montavista Linux Target File system: a complete bootable file system with extensive Linux tools and runtimes.
- Linux 2.6.18 kernel or Linux Support Package (LSP) for dm355: complete ARM Linux kernel with device drivers for peripherals on the dm355 SoC device and the dm355 EVM board.

Installation and Usage

You should follow the [DVEVM Getting Started Guide \(GSG\) spruf73c.pdf](#) for installation and usage instructions.

U-Boot and Linux Kernel

Please note that the kernel associated with this release is not pre-built for a particular platform. Please refer to the [kernel documentation](#) for information. You must insure that you are running the versions of U-Boot and Linux kernel supplied with this release in the `mv1_5_0_demo_lsp_setuplinux_xx_xx_xx_xxx.bin` installer. Consult the Release Notes under `dvsdk_2_00_00_22/PSP_02_00_00_140` for directions on working with U-Boot and the Linux kernel, including how to flash the board.

You should see the following version string or later for U-Boot when the board is first turned on:

- U-Boot 1.2.0 (Mar 13 2009 - 12:24:35) U-Boot 1.2.0 (May 20 2009 - 01:05:32)

This is a combined LSP so the uImage must be configured and built for each platform. An exact LSP string is not possible, but running `"uname -srvm"` should produce a date equal to or greater than 3 December, 2008:

"DaVinci DM355 DVSDK 2.00 Release Notes"

- Linux 192.168.1.92 2.6.18_pro500-davinci_evm-arm_v5t_le #1 PREEMPT Wed Dec 3 09:10:43 PST 2008 armv5tejl GNU/Linux

After re-compiling U-Boot or the Linux kernel, the version string will remain the same but the date/time string will change to when the kernel was recompiled.

Host Support

This release supports installation and development on Linux Redhat 4 and 5 workstations.

Dependencies

The only dependency known at this time is for Code Composer Studio (CCS) 3.3. CCS 3.3 and a compatible emulator is needed for initial (or recovery) flashing of the initial bootloader (UBL) and U-Boot. As long as these components remain functional, CCS 3.3 will not be needed.

Device Support

This release supports the Texas Instruments dm355 SoC as well as the Spectrum Digital dm355 Evaluation Module (EVM). For Spectrum Digital dm355 Evaluation Module drivers and firmware, please visit the Spectrum Digital dm355 site at <http://c6000.spectrumdigital.com/evmdm355/revd/>.

Validation Information

This release has been through a complete test cycle.

Upgrade and Compatibility Information

Since this is the initial release of the DM355 DVSDK, there is no need for upgrade or compatibility information. However, some users may be migrating or porting from the DM644x DVSDK. In general, the Linux support for DM355 is the same as for DM644x and most applications should run unchanged (or at most require a re-compile). The biggest difference between DM355 and DM644x DVSDK support is in the area of codecs.

- DM355 only supports JPEG and MPEG4 encoding and decoding.
- There are no audio codecs included with this DVSDK.
- There is an example G.711 speech codec. This is supplied in source form and is not considered production quality. There is no documentation for this example.

Demos and Examples

1. Installation of the software is described in Chapter 4 of the GSG.
2. Currently, the only supported target file system is a NFS-based file system. Directions for configuring a NFS server with the supplied target file system as well as directions for u-boot configuration are in Sections A5.3 and A5.4 of the GSG. After completion of these steps, you should be able to boot the

"DaVinci DM355 DVSDK 2.00 Release Notes"

EVM and get a Linux console on the serial port.

3. When building the demos, issue a "make" before a "make install"
4. Build the demos as described in Section 4.6 and then install them on (copy to) the target file system. See the encode.txt, decode.txt and encodedecode.txt files in the respective directories for usage directions on how to invoke the demo applications via the command line. A sample invocation is as follows:

- ◆ EVM# cd /opt/dv sdk
- ◆ EVM# ./loadmodules.sh
- ◆ EVM# ./encode -v /tmp/video.mpeg4 -t 15
- ◆ EVM# ./decode -v /tmp/video.mpeg4

This sample will record 15 seconds (-t 15) of MPEG4 encoded video to the file /tmp/video.mpeg4 using whatever source is connected to the composite video input port on the EVM. Then, the decode application will decode and display the captured video from /tmp/video.mpeg4 to the device connected to the composite video output port.

Examples

There are four examples in the examples sub-directory that illustrate basic usage of the MPEG4 and JPEG codecs. These examples simply read and write files. The basic usage model is to build the example, copy to the target file system along with an input file (test vector), and run on the target. See the comments in the source code for more details on usage.

MPEG4 Video Encode

- host \$ cd [path-to-dv sdk-install]/examples/venc/[path-to-dv sdk-install]
- host \$ make clean
- host \$ make
- host \$ cp venc foreman_vga_422i.yuv [path-to-nfs-target-fs]/opt/dv sdk/[path-to-nfs-target-fs]
- EVM # cd /opt/dv sdk
- EVM # ./venc foreman_vga_422i.yuv foreman_vga.m4v

On your development host, you can decode or view the resulting MPEG4 elementary stream (foreman_vga.m4v above) using Linux tools such as VLC.

MPEG4 Video Decode

- host \$ cd [path-to-dv sdk-install]/examples/vdec/[path-to-dv sdk-install]
- host \$ make clean
- host \$ make
- host \$ cp vdec akiyo_160x128.bits [path-to-nfs-target-fs]/opt/dv sdk/[path-to-nfs-target-fs]
- EVM # cd /opt/dv sdk
- EVM # ./vdec akiyo_160x128.bits akiyo_160x128_422ILE.yuv

On your development host, you can view the resulting YUV422 interleaved frames using Linux frame-based video or image tools.

JPEG Image Encode

- host \$ cd [path-to-dv sdk-install]/examples/jpegenc/[path-to-dv sdk-install]

"DaVinci DM355 DVSDK 2.00 Release Notes"

- host \$ make clean
- host \$ make
- host \$ cp jpegenc shrek_720x480_422I.yuv [path-to-nfs-target-fs]/opt/dvSDK/[path-to-nfs-target-fs]
- EVM # cd /opt/dvSDK
- EVM # ./jpegenc shrek_720x480_422I.yuv shrek.jpg

On your development host, you can view the resulting JPEG image using Linux image tools.

JPEG Image Decode

- host \$ cd [path-to-dvSDK-install]/examples/jpegdec/[path-to-dvSDK-install]
- host \$ make clean
- host \$ make
- host \$ cp jpegdec shrek_720x480.jpg [path-to-nfs-target-fs]/opt/dvSDK/[path-to-nfs-target-fs]
- EVM # cd /opt/dvSDK
- EVM # ./jpegdec shrek_720x480.jpg shrek_720x480_422I.yuv

On your development host, you can view the resulting YUV422 interleaved image using Linux image tools. You can also copy the YUV422 file directly to the display framebuffer:

- EVM # cat /dev/zero > /dev/fb/2
- EVM # ./jpegdec shrek_720x480.jpg /dev/fb/3

The first command sets the OSD blending attributes so that the video plane is visible. The second command decodes the JPEG file and writes it directly to the framebuffer for display.

Known Issues

- Deinterlacing support: Since MPEG4 SP does not include support for handling interlaced content, the encoded quality of interlaced video will be degraded. A limited "smoothing" function is available in this build. This is not a complete de-interlacer, but it does improve visual quality on high-motion video.
- The MPEG4 decoder has limited compatibility, so it cannot decode all MPEG4 bitstreams. For this release, the decoder will only decode bitstreams encoded by the DM355 MPEG4 encoder. Consult the MPEG4 decoder documentation for a detailed description of the decoder's limitations.
- The JPEG decoder also has limitations on what JPEG streams can be decoded. Consult the JPEG decoder documentation for a detailed description of the decoder's limitations.
- The encode demo application has a maximum recorded file size limit of 2GB.
- When running the encodedecode demo from the command line, better quality video will result by passing in -d (SDSCM00024417).
- The NAND flash file system uses YAFFS2 without checkpointing, consequently its boot-up time is proportional to the number files it contains. The DM355 DVEVM has a minimal file system on the flash to ensure fast boot time. Note that large file sizes may impact boot-up time as well, and the root cause for this issue is still under investigation. Hence, you may encounter this issue if you save a large video file using the encode demo.
- the links to the GSGs in the LSP release notes will not work.
- In some scenarios with VBR the FPS can drop to 23 frames per second

"DaVinci DM355 DVSDK 2.00 Release Notes"

- Under codec engine examples, video_copy/dualcpu_separateconfig_dll/enduser_app and video_copy/dualcpu_separateconfig/enduser_app will fail to build.
- To restore UBL and U-Boot binaries on your EVM, please follow the "LSP 2.00 DaVinci Linux EVM Installation User's Guide" at PSP_##_##_##_###/docs. The instructions for upgrading U-Boot from U-Boot (both NAND and NOR boot modes) do not work. So you can restore U-Boot binaries only using CCS.

Outstanding Defects

SDOCM00052209	DM355-dvtb. Board during Video+Speech Decode operations
SDOCM00052462	DM6467 Demo. Ctrl+C during demo operations causes the board to crash
SDOCM00053390	DM6446, dvtb demo decode, the video test files play back with poor quality (jumpy, jerkey)
SDOCM00053399	DM355 DVTB Goes into infinite loop during jpeg encode operation
SDOCM00053504	DM6467, dvtb, aac encode ext params, I have some test showing cmem error I am using default loadmodules.sh
SDOCM00053526	DM6467, dvtb, aac-ext, encode most test create files that play back too slow or too fast see test example
SDOCM00054294	Using CTRL+C on DM6446 decode demo causes kernel oops
SDOCM00054322	Codec Engine example applications fail to build
SDOCM00054325	DVTB Dm6446 System crashes if an audio decode request is made before a previous request has finished
SDOCM00054393	dsplinkk buildmodules script not in unix format
SDOCM00054406	Demos do not display block diagram
SDOCM00054460	DM6446: Updated u-boot from u-boot in NAND instructions fail
SDOCM00054508	

"DaVinci DM355 DVSDK 2.00 Release Notes"

	u-boot build has interworking compiler warnings
SDOCM00054554	Section on building drivers as modules is incorrect is LSP User's Guide
SDOCM00054556	DM6467, DVTB-aac-extparms setp aacheenc1 crcFlag 0 causes "Audio Encode Control failed (-1)"
SDOCM00054559	PSP examples for DM6446 do not compile without warnings
SDOCM00054560	PSP Example apps need usage documentation
SDOCM00054617	DVTB DM355. Area decode does not work for Image decode operations using JPEG extended parameters
SDOCM00054638	DM6467: stop button does not return to demo setup screen
SDOCM00054662	DM6467: codec engine example applications fail to build
SDOCM00054734	DM6467: instructions for upgrading u-boot from u-boot do not work without errors
SDOCM00054774	DM6467: DMAI fails to rebuild
SDOCM00054907	DMAI Video Decode io2 does not not flush the decoded video frames completely
SDOCM00055344	Dm355, DM6446, DM6467 demo encode speech with mic creates a file with very low levels.
SDOCM00057384	AAC files result in noisy output
SDOCM00057491	ARM CPU utilization values might not be accurate - large values above 200% are seen in few cases
SDOCM00057986	DM6467, AACEXT encode failed with error in shell
SDOCM00058026	DVSDK_200 LSP 200 when using boot from nand and nfs FS boot up hangs can not get IP address

Versioning

This is build 2.00.00.22 of DVSDK 2.0.

Technical Support and Product Updates

General Support

- For questions and support on the 2.00 DVSDK, please visit support.ti.com.
- Please be sure to read the DVSDK Read Me 1st guide, printed documentation and [Getting Started Guide](#) for general DVSDK information.

Custom Support

Third-Party Developer Support for DaVinci (c) technology: Valued members of the TI DSP Developer Network provide integral components and tools that complement DaVinci technology. Third-party developers offer various levels of video system integration, optimization and system expertise on products based on DaVinci Technology worldwide. Visit www.ti.com/davinci3p for a list of third-party developers who support DaVinci technology and information on their application.

Additional Resources

For more information, visit www.ti.com.