OMAP35x Graphics SDK Getting Started Guide

Literature Number: SPRUFZ8

April 28, 2009

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

EVALUATION BOARD/KIT IMPORTANT NOTICE

Texas Instruments (TI) provides the enclosed product(s) under the following conditions:

This evaluation board/kit is intended for use for **ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY** and is not considered by TI to be a finished end-product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards. This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and therefore may not meet the technical requirements of these directives or other related directives.

Should this evaluation board/kit not meet the specifications indicated in the User's Guide, the board/kit may be returned within 30 days from the date of delivery for a full refund. THE FOREGOING WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY SELLER TO BUYER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies TI from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

TI currently deals with a variety of customers for products, and therefore our arrangement with the user **is not exclusive**.

TI assumes **no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein**.

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages. For additional information on TI's environmental and/or safety programs, please contact the TI application engineer or visit www.ti.com/esh.

No license is granted under any patent right or other intellectual property right of TI covering or relating to any machine, process, or combination in which such TI products or services might be or are used.

Mailing Address: Texas Instruments Post Office Box 655303 Dallas, Texas 75265

Copyright © 2008, Texas Instruments Incorporated

FCC Warning

This evaluation board/kit is intended for use for **ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY** and is not considered by TI to be a finished end-product fit for general consumer use. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

Preface

About This Guide

The OMAP35x EVM is an evaluation platform that showcases the OMAP architecture and lets users evaluate the power and performance of OMAP as a flexible general purpose platform.

This guide gives you overview information about the board and the software provided with the board. It is intended to be used as an introductory document for the EVM. Other documents provide more in-depth information. See the EVM documentation section of the release notes for a complete list of documents that have been included with the product.

Notational Conventions

This document uses the following conventions:

Program listings, program examples, and interactive displays are shown in a `mono-spaced font`. Examples use **bold** for emphasis, and interactive displays use **`bold`** to distinguish commands that you enter from items that the system displays (such as prompts, command output, error messages, etc.).

Square brackets ([and]) identify an optional parameter. If you use an optional parameter, you specify the information within the brackets. Unless the square brackets are in a **bold** typeface, do not enter the brackets themselves.

Trademarks

The Texas Instruments logo and Texas Instruments are registered trademarks of Texas Instruments. Trademarks of Texas Instruments include: TI, OMAP, the OMAP logo, XDS, Code Composer, Code Composer Studio, Probe Point, Code Explorer, DSP/BIOS, RTDX, Online DSP Lab, OMAP, TMS320, TMS320C54x, TMS320C55x, TMS320C62x, TMS320C64x, TMS320C67x, TMS320C5000, and TMS320C6000.

MS-DOS, Windows, and Windows NT are trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

CodeSourcery and CodeSourcery's Sourcery G++ are trademarks or registered trademarks of CodeSourcery, Inc.

Solaris, SunOS, and Java are trademarks or registered trademarks of Sun Microsystems, Inc.

All other brand, product names, and service names are trademarks or registered trademarks of their respective companies or organizations.

Revision History:

Rev No.	Change	Date
0.1	Initial Draft	18 Nov 2008
0.2	Updated for review comments	24 Nov 2008
0.3	Modified the wiki link and more information provided for executing the Graphics driver tests	26 Nov 2008
1.0	Fixed the typo in section 3.4.1	28 Nov 2008
1.1	Source for graphics kernel modules included in the package Updated the document for building the kernel modules Updated the document for build the user developed demonstrations	18 Dec 2008
1.2	Updated the document to include details on patches added to Graphics SDK package	02 Feb 2009
1.3	Removed references to Graphics SDK version Number	11 Mar 2009
1.4	Updated the document for modifications in Beta v3.00.00.08	14 Apr 2009
1.5	Updated the document to include the following sections a. Building with TI LSP SDK v1.0.2 b. Selecting between FrontEGL and FlipEGL mode c. Testing support for ARGB8888 in framebuffer driver	28 April 2009

Table of contents

1 Overview

This chapter introduces the user to the OMAP35x Graphics SDK package along with the information on how to get further software updates on the OMAP35x Graphics SDK.

1.1 What's in the SDK Package?

The OMAP35x Graphics SDK package consists of the following folders

- GFX_Linux_SDK – This folder contains the OPENGL ES1.x, OPENGL ES2.0 and OPENVG demos
- GFX_Linux_KM – This folder contains the source for graphics kernel modules
- gfx_dbg – This folder contains the debug version of graphics drivers
- gfx_rel – This folder contains the release version of graphics drivers
- targetfs – This folder contains the pre-built targetfs
- patches – This folder contains the patches that are required for the graphics package to work with TI GIT based LSP release.
    - patches/psp_kernel_patches – This folder contains patches that needs to be applied on top of TI GIT based LSP v#.##.##.##. For more information on how to apply patches, refer to patch_readme.txt under the same folder.
    - patches/sgx_kernelmodule_patches - This folder contains patches that modify the OMAP35x Graphics SDK kernel modules to work with DSS2 framework used in TI GIT based LSP release v#.##.##.##. For more information on how to apply patches, refer to patch_readme.txt under the same folder.
- include – This folder contains the pvr2ds header files required for building QT for Linux
- utils (Contains unit test code to test ARGB8888)
- Getting Started Guide (This document)
- Rules.make (User modifications required to set the Graphics SDK install path)
- Makefile (Top level makefile used for building Graphics SDK demos and Installing in the NFS target file system)
- readme.pdf – Contains the release information

The drivers included in the package are:

- OpenGL ES 1.1
- OpenGL ES 2.0
- OpenVG
- EGL

1.2 What's next?

To get started on evaluating the OMAP35x Graphics capabilities and developing applications using the SGX accelerator for OMAP35x EVM, begin by using this Getting Started guide. It is suggested that the user follow the sequence described in the document. The sequence is outlined below.

- OMAP35x DVEVM Software Setup for Graphics SDK
- Executing the Graphics SDK Demonstration Software
- Building the NFS Target File System with Graphics SDK Demos
- Building the OMAP35x Graphics Demo Software
- Building the Graphics Kernel Modules
- Switching between FrontEGL and FlipEGL Modes
- Testing the support for ARGB8888 in Frame Buffer Driver

1.3 Command Prompts in this Guide

In this guide, commands are preceded by prompts that indicate the environment where the command is to be typed. For example:

- host $
  Indicates command to be typed into the shell window of the host Linux workstation.
- `OMAP3EVM #`
  Indicates commands to be typed into the U-Boot shell in a console window connected to the EVM board's serial port.
- `target $`
  Indicates commands to be typed into the Linux shell in the terminal window connected to the EVM board's serial port.

1.4 Updates & Further information

Please be sure to register your EVM at:

www.ti.com/omapregistration

Once registered you will be able to find software and documentation updates at:

www.ti.com/omapsoftwareupdates

A wiki based information site is available. User contributions are encouraged.

http://wiki.davincidsp.com/index.php?title=Main_Page

2 OMAP35x DVEVM Software Setup for Graphics SDK

2.1 Requirements

The following are the requirements for software setup:

- PC Linux Host
  - Root login or liberal sudo privileges
  - Most Linux Distributions should work with this Graphics SDK
  - This Graphics SDK has been tested on
    - Red Hat Enterprise Linux, Workstation V4, x86 32 bit
- Linux Support Package for OMAP35x EVM – This graphics SDK package has been built and tested with TI LSP GIT release version as specified in the readme.pdf file

2.2 Preparing to Install

Download the following installers to the temp folder /tmp from the OMAP3 software update site as mentioned in Section 1.4

- OMAP35x_SDK_setuplinux_#_##_##.bin
- OMAP35x_Graphics_SDK_setuplinux_#_##_##_##.bin

The # refers to the version numbers

The OMAP35x Graphics SDK contains the pre-built graphics drivers and kernel modules that are already built with the appropriate version of the OMAP35x_SDK_setuplinux_#_#_#.bin. Make sure that this dependency is maintained and the user uses the kernel image built from the correct version of the OMAP35x SDK as mentioned in Section 2.1

Ensure all the .bin files are set with executable permissions.

```
host $ chmod 777 *.bin
```

2.3 Installing the Software

Installing the software used by the OMAP35x EVM involves performing the following steps.

Section 2.3.1, *Installing the Target Linux Software*

Section 2.3.2, *Installing the Graphics SDK Software*

Section 2.3.3, *Installing the toolchain*

2.3.1 Installing the Target Linux Software

This section explains how to install Linux for use on the target board.

Note that separate versions of Linux are used by the target and your host Linux workstation. The following Linux host operating systems are supported for use with the OMAP35x DVEVM.

- Red Hat Enterprise Linux, Workstation V4, x86 32 bit

To install the Linux software, follow these steps:

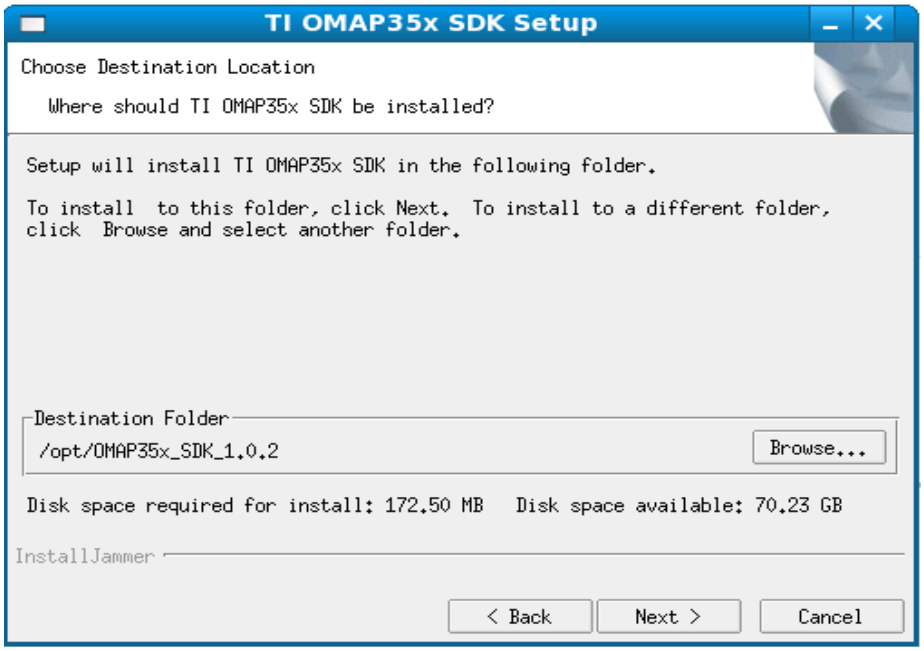
1. Execute **`OMAP35x_SDK_setuplinux_#_#_#.bin`** file (where **`#_#_#`** is the current version number) from the temporary location that they were copied to extract the installers for the Linux tools, Linux kernel, and the file system.

   ```
   host $ cd /tmp
   host $ ./OMAP35x_SDK_setuplinux_#_#_#.bin
   ```

2. The installer will start as a GUI application. Follow the instructions in the dialog boxes. You'll be asked to agree to the End User License Agreement.

3. You will be prompted to provide the installation directory. If installed from a normal user account the default install path will be "/home/<user_account>/OMAP35x_SDK_#.#.#". If installed with root permissions the default install path will be "/opt/OMAP35x_SDK_#.#.#". In future, all references in the document will assume that the user has installed the OMAP35x SDK using his user account and hence will refer to the OMAP35x SDK installation path as /home/<user_account>/OMAP35x_SDK_#.#.#

Note: The Linux Support Package (LSP) shipped with the OMAP35x DVEVM is a multi-platform LSP and is not configured for a particular platform. As shipped, this LSP cannot be used to build the demo or example applications. It must first be copied to a user area and configured/built for the OMAP35x EVM.



2.3.2 Installing the Graphics SDK Software

To install the Graphics SDK software using the Linux installer, follow these steps:

1. Log in using a **user account**. In the following steps, we refer to the home directory as "~".
2. Execute the OMAP35x Graphics SDK installer that you previously downloaded from the extranet location. For example:

   ```
   host $ ./OMAP35x_Graphics_SDK_setuplinux_#_##_##_##.bin
   ```

   The installer will start as a GUI application. Follow the instructions in the dialog boxes. You'll be asked to agree to the End User License Agreement.

   When you are prompted for an installation location, use the default installation location, that points to /home/<user_account>/OMAP35x_Graphics_SDK_#.##.##.##. This location will be used as the OMAP35x Graphics SDK installation folder through out this document.
3. You can now delete the .bin files that you loaded into the temporary directory.

Note: You can uninstall these components by using the uninstall file in the respective installation directories. For example:

```
host $ cd /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##
host $ ./uninstall
```

2.3.3 Installing the Toolchain

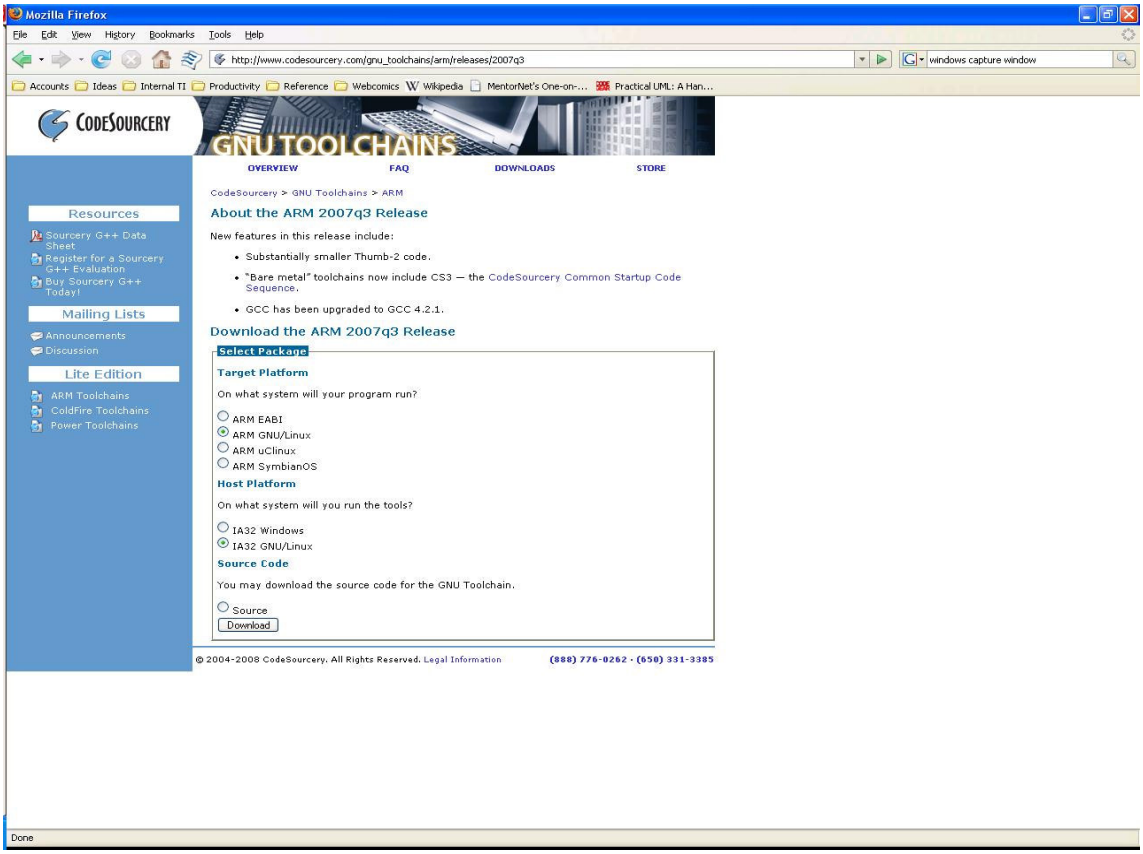
This guide assumes use of the LITE version of the CodeSourcery toolchain. See CodeSourcery documentation if you are using the evaluation version that comes with this EVM or for other cases.

The toolchain used is ARM GNU/Linux EABI 2007q3. It can be downloaded via the following URL:

http://www.codesourcery.com/gnu_toolchains/arm/releases/2007q3

Select the following options:

Target Platform:	ARM GNU/Linux
Host Platform:	IA32 GNU/Linux



To install the toolchain, follow the sequence below. These are to be executed on the Linux host platform.

```
$ mkdir -p /home/<user_account>/toolchain
$ cp arm-2007q3-51-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
/home/<user_account>/toolchain
$ cd /home/<user_account>/toolchain
$ tar -jxvf  arm-2007q3-51-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
$ rm arm-2007q3-51-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar
```

2.4 Setting up the Build/Development Environment

To set up the development and build environment, follow these steps:

1. Log in to your **user** account (and not as root) on the Linux host system, where the OMAP35x Graphics SDK is installed.

2. Add the `/host/<user_account>/toolchain/arm-2007q3/bin` directory to your path. This is typically done by adding an additional line to your shell resource file (~/.bashrc). For the path given above, the line to add to your .bashrc file is:

   PATH="/home/<user_account>/toolchain/arm-2007q3/bin:$PATH"

   This adds the CodeSourcery tools to your path and allows you to execute the tools using arm-none-linux-gnueabi-gcc (or other tools in the tool chain) from any directory.

3. Remember to use the following command after modifying your .bashrc file:
   `host $` **source .bashrc**

4. You can test that the toolchain is installed correctly by starting a new shell and using the following command:
   `host $` **arm-none-linux-gnueabi-gcc -v**

   When you execute this command, you will get an output like the one shown below.

```
Using built-in specs.
Target: arm-none-linux-gnueabi
Configured with: /scratch/paul/lite/linux/src/gcc-4.2/configure --build=i686-pc-linux-gnu --host=i686-pc-linux-gnu --target=arm-none-linux-gnueabi --enable-threads --disable-libmudflap --disable-libssp --disable-libgomp --disable-libstdcxx-pch --with-gnu-as --with-gnu-ld --enable-languages=c,c++ --enable-shared --enable-symvers=gnu --enable-__cxa_atexit --with-pkgversion=CodeSourcery Sourcery G++ Lite 2007q3-51 --with-bugurl=https://support.codesourcery.com/GNUToolchain/ --disable-nls --prefix=/opt/codesourcery --with-sysroot=/opt/codesourcery/arm-none-linux-gnueabi/libc --with-build-sysroot=/scratch/paul/lite/linux/install/arm-none-linux-gnueabi/libc --enable-poison-system-directories --with-build-time-tools=/scratch/paul/lite/linux/install/arm-none-linux-gnueabi/bin --with-build-time-tools=/scratch/paul/lite/linux/install/arm-none-linux-gnueabi/bin
Thread model: posix
gcc version 4.2.1 (CodeSourcery Sourcery G++ Lite 2007q3-51)
```

3 Executing the Graphics SDK Demonstration Software

This section explains how to run the graphics demos provided with the OMAP35x Graphics SDK.

3.1 Default Boot Parameter Settings

The boot parameter settings for OMAP35x EVM are similar to the settings as described in the OMAP35x EVM Getting Started Guide. Use the bootargs as mentioned below to execute the Graphics SDK demos. This is different from the bootargs used for running the OMAP3530 DVSDK demos and does not enable rotation and the memory available for linux operating system is 128MB or 256MB depending on the DDR2 part used in the processor board.

3.1.1 Boot parameter settings for NAND Flash Target File System

Kindly note that the bootargs are required to be set as below while running the OMAP35x Graphics SDK demos from NAND flash target file system

```
OMAP3EVM # setenv bootargs   mem=128M console=ttyS0,115200n8 noinitrd
root=/dev/mtdblock4 rw rootfstype=jffs2
```

NOTE:

Make sure that the size of the frame buffer is greater than 4MB. If not add omapfb.vram="4M" to bootargs. You could check this by executing the following command

```
target$ cat /sys/devices/platform/omapfb/framebuffers
```

3.1.2 Boot parameter settings for NFS

Kindly note that the bootargs are required to be set as below while running the OMAP35x Graphics SDK demos from NFS file system

```
OMAP3EVM # setenv bootargs console=ttyS0,115200n8 noinitrd rw ip=dhcp  root=/dev/nfs
nfsroot=<nfshost>:<rootpath>,nolock mem=128M
```

3.2 Pre-Built NFS Target File System

The pre-built NFS target file system for OMAP35x graphics SDK demonstration evaluation is available under **/home/<*user_account*>/OMAP35x_Graphics_SDK_#.##.##.##/targetfs** directory.

Alternatively, user could also build the target file system with the necessary binaries and executables for running Graphics SDK demos using the OMAP3503 SDK NFS target file system. Refer to Section 4.2 for more details.

Perform the following steps in order to use the pre-built target NFS.

1. Log in with a **user** account on the host Linux workstation. (In the following steps, we refer to the home user directory as "~".)

2. Perform the following commands to prepare a location for the OMAP35x EVM target file system.

   ```
   host $ cd ~
   host $ mkdir –p workdir/filesys
   host $ cd workdir/filesys
   ```

3. Switch user to "**root**" on the host Linux workstation.
   ```
   host $ su
   ```

password:

There will be a prompt for entering the password as shown above. Type the root password, for getting the root permissions

4. Copy the pre-built NFS target file system from /home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/targetfs/nfs.tar.gz to a new directory created in step 2. Perform the following commands.

   NOTE: Un-tar the file with root permissions. (Step 3)

   ```
   $ cp /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/targetfs/nfs.tar.gz .
   $ tar -xvzf nfs.tar.gz .
   ```

5. Perform the following command to make sure you can write into the target file system that you want to export as NFS, from your user account.

   host $ **chown -R *<user_account>* /home/<*user_account*>/workdir/filesys/.**
   host $ **chmod -R 755 /home/<*user_account*>/workdir/filesys/.**

6. Make sure the NFS server is configured and functioning properly. Add the following line to the /etc/exports file of the server. Ensure you have root permission before editing this file.

   ```
   /home/<user_account>/workdir/filesys  (rw,sync,no_subtree_check)
   ```

7. Then issue the following command to notify the NFS server about the new exported directory.
   host $ **/usr/sbin/exportfs -a**
   host $ /**sbin/service nfs restart**

8. Make sure you exit from having the root permissions after completing all the above steps
   host $ **exit**

3.3 Running the Standalone Demos

Out of the box, on EVM power on, the graphics SDK demos are played back if the graphics SDK demos are flashed in the NAND flash file system.

Refer to the OMAP35x EVM Getting Started Guide or OMAP3530 DVSDK Getting Started Guide for information on flash procedures and alternate boot methods.

By default, OpenGL ES1.x demos are executed and run once. After they are executed once, the demo stops. Press “Enter” in the console window on the PC to go to the command line and execute more demos from the command line as described in section 3.4

3.4 Running the Demos from Command Line

You can run the OMAP35x Graphics SDK demo applications from the Linux shell in a terminal window connected to the EVM board’s serial port.

Before running graphics SDK demo applications from the command line, the following commands need to be executed.

```
target $ cd /etc/init.d

target $ ./rc.pvr start

This will insert all the kernel modules and perform necessary initializations
```

3.4.1 Running Graphics SDK Sample Applications

These are standard Graphics SDK tests that are used to check the basic functionality of the graphics drivers. To run standard graphics SDK tests, perform the following steps:

```
target $ cd /usr/local/bin
target $ ./gles1test1 -t
target $ ./gles2test1 -t
target $ ./ovg_unit_test -t
```

To stop the above graphics SDK tests, press ctrl+c

Execute the ES1 texture stream test using the command given below. To stop gles1_texture_stream test, press 'q'

```
target $ ./gles1_texture_stream
```

Execute the following commands given below to test the SGX feature tests. These tests end on their own except for the sgx_render_flip_test. In this case, press ctrl+c to stop the test.

```
target $ ./ sgx_blit_test
target $ ./ sgx_flip_test
target $ ./ sgx_render_flip_test
target $ ./ sgx_render_test
```

3.4.2 Running Graphics SDK OpenGL ES1.x Demos

To run OpenGL ES1.x demos, perform the following steps:

```
target $ cd /opt/gfxsdkdemos/ogles
target $ ./OGLESChameleonMan
```

This will execute the OGLESChameleonMan demo. Press 'q' on the host machine console window (TeraTerm or HyperTerminal or Minicom) to stop the demo.

Similarly, the user could execute other OpenGL ES1.x demos.

For more information and command line options on the demos, refer to the OpenGL ES1.x SDK user guide available under OMAP35x_Graphics_SDK_#_##_##_##\GFX_Linux_SDK\OGLES\SDKPackage

3.4.3 Running Graphics SDK OpenGL ES2.0 Demos

To run OpenGLES2.0 demos, perform the following steps:

```
target $ cd /opt/gfxsdkdemos/ogles2
target $ ./OGLES2Shaders
```

This will execute the OGLES2Shaders demo. Press 'q' on the host machine console window (TeraTerm or HyperTerminal or Minicom) to stop the demo.

Similarly, the user could execute other OpenGL ES2.0 demos.

For more information and command line options on the demos, refer to the OpenGL ES2.x SDK user guide available under OMAP35x_Graphics_SDK_#_##_##_##\GFX_Linux_SDK\OGLES2\SDKPackage

4 Building the NFS Target File System with Graphics SDK Demos

The base NFS target file system can be picked up from the OMAP3503 SDK installation and the necessary executables and libraries required for running graphics SDK demos and development of applications using the graphics libraries can be built using the following steps.

4.1 Building the NFS target file system

Perform the following steps to build and export the NFS target file system.

1. Log in with a **user** account on the host Linux workstation. (In the following steps, we refer to the home user directory as "~".)

2. Perform the following commands to prepare a location for the OMAP35x EVM target file system.

   ```
   host $ cd ~
   host $ mkdir –p workdir/filesys
   host $ cd workdir/filesys
   ```

3. Switch user to "**root**" on the host Linux workstation.
   ```
   host $ su
   password:
   ```

   There will be a prompt for entering the password as shown above. Type the root password, for getting the root permissions

4. Copy the OMAP3503 SDK NFS target file system from /home/<*user_account*>/OMAP35x_SDK_#.#.#/bin/nfs.tar.gz to a new directory created in step 2. Perform the following commands.

   NOTE: Un-tar the file with root permissions. (Step 3)

   ```
   $ cp /home/<user_account>/OMAP35x_SDK_#.#.#/bin/nfs.tar.gz .
   $ tar –xvzf nfs.tar.gz .
   ```

5. Perform the following command to make sure you can write into the target file system that you want to export as NFS, from your user account.

   host $ **chown -R *<user_account>* /home/<*user_account*>/workdir/filesys/.**
   host $ **chmod -R 755 /home/<*user_account*>/workdir/filesys/.**

6. Make sure the NFS server is configured and functioning properly. Add the following line to the /etc/exports file of the server. Ensure you have root permission before editing this file.

   ```
   /home/<user_account>/workdir/filesys  (rw,sync,no_subtree_check)
   ```

7. Then issue the following command to notify the NFS server about the new exported directory.
   ```
   host $ /usr/sbin/exportfs –a
   host $ /sbin/service nfs restart
   ```

8. Make sure you exit from having the root permissions after completing all the above steps
   ```
   host $ exit
   ```

9. Perform steps 1 to 3 as mentioned in Section 5,

Building the OMAP35x Graphics Demo Software. Make sure to set the TARGETFS_INSTALL_DIR to point to the NFS target file system folder (/home/<user_account>/workdir/filesys) as created in this section, in the Rules.make file.

10. The 'make install' will install all the necessary libraries and executables required for running the Graphics SDK demos as well as application development using the graphics drivers.

5 Building the OMAP35x Graphics Demo Software

5.1 Building the OMAP35x Graphics SDK Demos

This section describes how to rebuild OMAP35x Graphics SDK demos. Ensure that the toolchain setup and install has been completed (see section 2.3.3).

To rebuild the OMAP35x Graphics SDK demos, perform the following steps.

1. Change the directory to ~/OMAP35x_Graphics_SDK_#_##_##_##.

2. Edit the ~/OMAP35x_Graphics_SDK_#_##_##_##/Rules.make file.

   - Set GRAPHICS_INSTALL_DIR to the top-level OMAP35x graphics installation directory as follows.

     HOME=/home/<*user_account*>
     GRAPHICS_INSTALL_DIR=${HOME}/OMAP35x_Graphics_SDK_#_##_##_##

   - Modify the TARGETFS_INSTALL_DIR to point to the correct location where the user's target file system resides

     TARGETFS_INSTALL_DIR=/home/<*user_account*>/workdir/filesys

   - Depending on the use of debug version or release version of the graphics driver libraries, set the GFX_LIB_SRCPATH accordingly.

     If debug version of the graphics driver library is desired, then
     GFX_LIB_SRCPATH = $(GFX_DBG_LIB_SRCPATH)

     If release version of the graphics driver library is desired, then
     GFX_LIB_SRCPATH = $(GFX_REL_LIB_SRCPATH)

   - Change the path of the CSTOOL_DIR to point to the location where you have installed the CodeSourcery tool-chain. (Refer Section 2.3.3)

     CSTOOL_DIR=/home/<user_account>/toolchain/arm-2007q3

3. While in the same directory that contains Rules.make, use the following commands to build the OMAP35x Graphics SDK demo applications and install the resulting binaries on the target file system specified by EXEC_DIR.

   ```
   host $ make all
   host $ make install
   ```

5.2 Building and Executing Your Own Demos

This section describes the important steps required to build user implemented demonstrations.

1. Place your demos in the same directory format as default graphics SDK demos
2. Copy the release or debug versions of the shared and static libraries from the appropriate gfx_rel or gfx_dbg folders to the OpenGL demo locations as shown below

   In case of OpenGL ES1.x demos, perform the following steps

   ```
   host $ mkdir -p /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
   GFX_Linux_SDK/OGLES/SDKPackage/Builds/OGLES/LinuxOMAP3/lib
   ```

   ```
   host $ cp /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/gfx_rel/*.so
   /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
   GFX_Linux_SDK/OGLES/SDKPackage/Builds/OGLES/LinuxOMAP3/lib
   ```

   ```
   host $ cp /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/gfx_rel/*.a
   /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
   GFX_Linux_SDK/OGLES/SDKPackage/Builds/OGLES/LinuxOMAP3/lib
   ```

   In case of OpenGL ES2.x demos, perform the following steps

   ```
   host $ mkdir -p /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
   GFX_Linux_SDK/OGLES2/SDKPackage/Builds/OGLES/LinuxOMAP3/lib
   ```

   ```
   host $ cp /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/gfx_rel/*.so
   /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
   GFX_Linux_SDK/OGLES2/SDKPackage/Builds/OGLES/LinuxOMAP3/lib
   ```

   ```
   host $ cp /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/gfx_rel/*.a
   /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
   GFX_Linux_SDK/OGLES2/SDKPackage/Builds/OGLES/LinuxOMAP3/lib
   ```

   Perform similar steps as above for OpenVG demos as well.

3. Set the following environment variables
   - Ensure that the ARM toolchain path is included in the PATH variable

     ```
     host $ export PATH=/home/<user_account>/toolchain/arm-2007q3/bin:$PATH
     ```

   - Set the PLATFORM environment variable to LinuxOMAP3

     ```
     host $ export PLATFORM = LinuxOMAP3
     ```

   - Set the LIBDIR environment variable to the path where the graphics driver libraries are copied.

     For OpenGL ES1.x demos, set the following

     ```
     host $ export LIBDIR = /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
     GFX_Linux_SDK/OGLES/SDKPackage/Builds/OGLES/LinuxOMAP3/lib
     ```

     For OpenGL ES2.x demos, set the following

     ```
     host $ export LIBDIR = /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/
     GFX_Linux_SDK/OGLES/SDKPackage/Builds/OGLES2/LinuxOMAP3/lib
     ```

   - Set the DISCIMAGE to your target file system install directory. This is required when the graphics and the dependent libraries are installed

```
host $ export DISCIMAGE = <your_target_file_system>
```

4. Install the graphics driver libraries and kernel modules by performing the following steps

```
host $ cd /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/gfx_rel

In case of debug library installation navigate to the gfx_dbg folder

host $ ./install.sh
```

5. Copy libstdc++.so, libstdc++.so.6 and libstdc++.so.6.0.9 from the toolchain path to your target file system's /usr/lib directory

```
host $ cp /home/<user_account>/toolchain/arm-2007q3/arm-none-linux-gnueabi/lib/libstdc++.so* /home/<user_account>/workdir/filesys/usr/lib
```

6. Perform the build of your demonstration

```
host $ cd /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/GFX_Linux_SDK/OGLES/SDKPackage/Demos/<your_OpenGL_ES1.x_demo>/OGLES/Build/LinuxGeneric

host $ make
```

7. Copy the built executables to the target file system and execute them on the target

```
host $ cp /home/<user_account>/OMAP35x_Graphics_SDK_#_##_##_##/GFX_Linux_SDK/OGLES/SDKPackage/Demos/<your_OpenGL_ES1.x_demo>/OGLES/Build/LinuxOMAP3/ReleaseRaw/<demo_executable> /home/<user_account>/workdir/filesys/opt/gfxdemos
```

8. On the target, perform the following steps for executing your demos

```
target $ cd /etc/init.d

target $ ./rc.pvr start

target $ cd /opt/gfxdemos

target $ ./<your_demo_exec>
```

6 Building the Graphics Kernel Modules

6.1 Building with TI OMAP35x LSP GIT

Perform the following steps to build the graphics kernel modules with the TI LSP GIT

1. Apply the patch under **/home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/patches/sgx_kernelmodule_patches** folder. Refer to patch_readme.txt under the same folder
2. Navigate to the kernel module build folder

   host $ **cd /home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/GFX_Linux_KM/eurasiacon/build/linux/omap3430_linux/kbuild**
3. Edit build_gfx_kernelmodules file
   - Set the kernel directory to point to the directory where the Linux kernel is extracted
   - Set the DISCIMAGE to point to your target file system
   - Make sure the PATH environment variable contains the tool chain path
   - For Debug Build, uncomment the line that contains 'make BUILD=debug' and comment the line that has 'make'
4. Perform the build by executing the following command

   host $ **./build_gfx_kernelmodules**
5. The kernel modules are built under **/home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/GFX_Linux_KM/eurasiacon/binary_omap3430_linux_release** or **/home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/GFX_Linux_KM/eurasiacon/binary_omap3430_linux_debug** depending upon the release or debug build performed
6. Copy the kernel modules to the <*your_graphics_install_dir*>/gfx_rel or <*your_graphics_install_dir*>/gfx_dbg based on the build performed.

   For example,

   host $ cp **/home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/GFX_Linux_KM/eurasiacon/binary_omap3430_linux_release/*.ko /home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/gfx_rel**

6.2 Building with TI OMAP35x LSP SDK v1.0.2

Perform the following steps to build the graphics kernel modules with the TI OMAP35x LSP SDK v1.0.2

1. Apply the patch under **/home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/patches/sgx_kernelmodule_patches** folder. Refer to patch_readme.txt under the same folder
2. Navigate to the kernel module build folder

   host $ **cd /home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/GFX_Linux_KM/eurasiacon/build/linux**

   Enable support for SUPPORT_TI_PM and SUPPORT_TI_DSS_FW in the makefile.shared_conf by assigning a value of 1 to the above mentioned variables.
3. Perform steps 3 to 6 mentioned in section 6.1

7 Switching between FrontEGL and FlipEGL Modes

Modify the powervr.ini file under the <your_target_file_system>/etc/init.d to select between the FrontEGL or FlipEGL mode.

Flip support requires increasing the virtual yres size in fb0 plane.

Use the following command to increase the virtual yres size in fb0 plane. The vyres size needs to be set to three times the height of the screen size for triple buffering

```
target $ fbset -vyres 1920
```

8 Testing the support for ARGB8888 in Frame Buffer Driver

The bootargs need to be changed as given below.

```
OMAP3EVM # setenv bootargs 'console=ttyS0,115200n8=noinitrd=rw ip=dhcp
root=/dev/nfs nfsroot=<your nfs_path>,nolock, mem=128M omapfb.vram="4M"
omapfb.video_mode="640x480MR-24@60" omap-dss.def_disp="dvi"'
```

Copy the setmode application available under **/home/<*user_account*>/OMAP35x_Graphics_SDK_#_##_##_##/utils** to your NFS area and execute the following command on the target

```
target $ setmode -d fb0 -x 640 -y 480 -v 640 -w 1440 -b 32 -i 1
```