OMAP35x Linux Performance Test Bench

User Guide

Literature Number: SPRUXXX

JULY 2008

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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
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
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Read This First

About This Manual

This document describes how to install and work with Performance Test Bench for OMAP35x platform for Linux 2.6.22. The pspTest Package serves to provide application software for performance measurement of Linux Device Drivers. This abstracts the functionality provided by the OMAP35X LINUX PSP Package. The product forms the basis for measurement of performance on this platform

This release is based on OMAP35XX PSP REL 0.9.8. The Package has been compiled with Code sourcery based on respective EVM OMAP35X LINUX PSP release notes.

Intended Audience

This document is intended for the users of OMAP35X LINUX PSP package who might want to measure/verify the performance of various device drivers using the Linux Performance Test Bench. This would facilitate the user to identify / rectify performance bottlenecks in their respective systems design.

This document assumes that the user has hands on experience with the Linux platform and some knowledge regarding the OMAP35X LINUX PSP device drivers for which performance and CPU load parameters are being measured.

How to Use This Manual

This document includes the following chapters:

- **Chapter 1 – Introduction**, gives the brief introduction about the pspTest tool.
- **Chapter 2 - Installation**, describes the installation procedure for OMAP35x Linux PSP package and pspTest tool.
- **Chapter 3 - Build**, describes the build procedure for U-Boot, Linux kernel, and pspTest tool.
- **Chapter 4 - Using pspTest Application**, describes the test setup details, procedure to be followed for running the scripts that are provided as part of pspTest tool, and executing the pspTest through command line.
- **Appendix A - General Setup Details**, provides information about the EVM setting details and output/input console setting details.

- **Appendix B - Adding New Test Case**, provides details on how a test case can be added to pspTest tool.

Before you proceed with the installation, see performance_test_bench_releasenotes.pdf file available in the release package.

Terms and Abbreviations

The following terms and abbreviations are used in this document.

Term/Abbreviation	Description
DUT	Device Under Test
API	Application Programming Interface
IO	Input/Output
IOCTL	Input Output ConTroL
MMCSD	MultiMedia Card / San Disk
USB	Universal Synchronous Bus
Fps	Frames Per Second
NTSC	National Television System Committee
PAL	Phase Alternating Line
TV	TeleVision
DSS	Display Sub System

If You Need Assistance

For any assistance, send a mail to dsppsp_val@list.ti.com.

Text Conventions

The following conventions are used in this document:

- Text inside back-quotes (‘‘) represents pseudo-code.
- Program source code, function and macro names, parameters, and command line commands are shown in a `mono-spaced` font.

Trademarks

Code Composer Studio, DSP/BIOS, eXpressDSP, TMS320, TMS320C64x, TMS320C6000 and TMS320C64x+ are trademarks of Texas Instruments.

Contents

Figures

Tables

Revision History

Date	Author	Comments	Version
June 28, 2008	Somasekar	Initial Draft	0.1.0

Chapter 1

Introduction

This chapter describes the services, features, limitations, and requirements of the Linux Performance Test Bench.

1.1 Overview

Linux Performance Test Bench supports benchmarking of various Linux device drivers supplied as part of the Linux Support Packages (OMAP35X LINUX PSP) for TI platforms. The current package support throughput and CPU load measurements for the device driver IO operations. This product can be scaled up to add support for new drivers, new platforms and additional performance parameters.

1.1.1 Supported Services

Linux Performance Test Bench provides the code to get performance and CPU load parameters for the following device drivers:

- q DSS
- q Video Capture
- q Audio
- q oneNAND
- q Micron NAND
- q MMC/SD
- q USB MSC Host

1.1.2 Supported Features

- q Linux Performance Test Bench supports throughput measurement for both User Level and Kernel Level device drivers.
- q Linux Performance Test Bench supports CPU load measurement for User Level device drivers.
- q Using the scripts available in the package for all the device drivers, throughput can be measured with minimal manual effort.
- q Using the command line, user can get throughput of all the user level device drivers for various input parameters like different buffer sizes, sampling rates etc.
- q Common methods are used for buffer allocation, time measurement for performance calculations.

1.2 Limitations

- q Boundary checking for Input parameters given through command line is not taken care. So user should give the input parameters accordingly.

- CPU load measurement for kernel level modules and memory requirements while measuring throughput will be implemented in later phase.
- Directory structure is prone to changes when adding support for new platforms.

1.3 Basic Hardware and Software Requirements

1.3.1 Hardware Requirements

The Hardware required for using the Linux Performance Test Bench is

- OMAP35x EVM with 5V, 3A Power Supply
- OMAP3EVM-OMAP3-ES2.0-LinuxSoftware-1.0 utilities for flashing x-loader and u-boot
- UART and Ethernet Cables
- OMAP35x Capture Card

The specific hardware requirements for individual module throughput measurement have been mentioned in Chapter 4.

1.3.2 Software Requirements

Linux Support Package for TI Platforms which includes

- Device drivers required for OMAP35x
- Source for U-Boot

Chapter 2

Installation

This chapter describes the installation procedure for OMAP35x Linux PSP package and pspTest Tool.

Topic	Page

2.1 Release Access

See performance_test_bench_releasenotes.pdf file available in the release package, for release access details.

2.2 System Requirements

See ReleaseNotes_0_9_8.pdf available in OMAP35x Linux PSP Package, for system requirements.

2.3 Installation

The following points provide information on installation:

- UserGuide_0_9_8.pdf available in OMAP35x Linux PSP Package, for installation of OMAP35x EVM and Flashing of U-Boot.
- Install linux_performance_test_bench_2.1.0.tar.gz by unzipping the package using the command tar –xvzf linux_performance_test_bench_2.1.0.tar.gz in Linux Operating System.

Chapter 3

Build

This chapter describes the build procedure for U-Boot, Linux kernel, and pspTest Tool.

Topic **Page**

3.1 Compiling U-Boot

See ReleaseNotes_0_9_8.pdf available in OMAP35X LINUX PSP Package, for Compiling U-Boot.

3.2 Compiling Linux Kernel

See ReleaseNotes_0_9_8.pdf available in OMAP35X LINUX PSP Package, for Compiling Linux Kernel.

3.3 Compiling pspTest

1. Export the PSP_TEST_HOME to psp_test_bench directory of the performance test bench installation
2. Refer to the PSP_TEST_HOME/README.txt and PSP_TEST_HOME/HowtoConfigure.txt for configuration and build details
3. Following variables in GENDEFS file needs to be updated:
   a. TOOL_CHAIN- Defines the installation directory of MontaVista Toolchain. By default, this tool chain is assumed to be under /opt. If this installation is in a different location, then this variable needs to be changed.
   b. INSTALL_DIR - Defines the directory where the target binaries and utilities need to be copied. By default, this will refer to TOOL_CHAIN/target/pspTestTarget. Modify this variable to install the pspTest target binaries and utilities at different location.
   c. RELEASE - Defines the OMAP35X LINUX PSP installation directory. Modify this variable to point to the location of OMAP35X LINUX PSP, which has been configured and built for OMAP35x.
   d. KERNEL_DIR – Defines the Kernel directory path of the OMAP35X LINUX PSP release. Modify this variable to point to the location of OMAP35X LINUX PSP, which has been configured and built for OMAP35x.
   e. CC – GCC compiler name.Modify this variable according to your tool chain.
4. Using the `make` command at //PSP_TEST_HOME/make/target/. This will create the binary executable in //PSP_TEST_HOME/bin directory. The `make` supports the following additional features:
   a. `make clean` - This deletes the pspTest executable and all other object files. It also deletes the INSTALL_DIR/pspTestTarget folder.
   b. `make install` - Copies the pspTest target binaries and utilities to INSTALL_DIR/pspTestTarget.

Chapter 4

Using pspTest Application

This chapter describes the test setup details, procedure for running the scripts that are provided as part of pspTest tool, and executing the pspTest through the command line.

Topic **Page**

4.1 DSS

This section provides the steps to execute the DSS performance tests using the scripts or command line utility. It also provides the performance parameters, test setup information, test environment, and command line arguments. The section has been further subdivided into V4L2 (Video for Linux 2) and FBDEV (Frame buffer devices) frameworks.

4.1.1 V4L2 Framework

4.1.1.1 Performance Parameters

Following DSS performance parameters will be obtained using pspTest tool with the V4L2 framework:

a. Display frame rate for the output stream in fps (frames per sec)

b. Percentage of CPU load

4.1.1.2 Test Setup

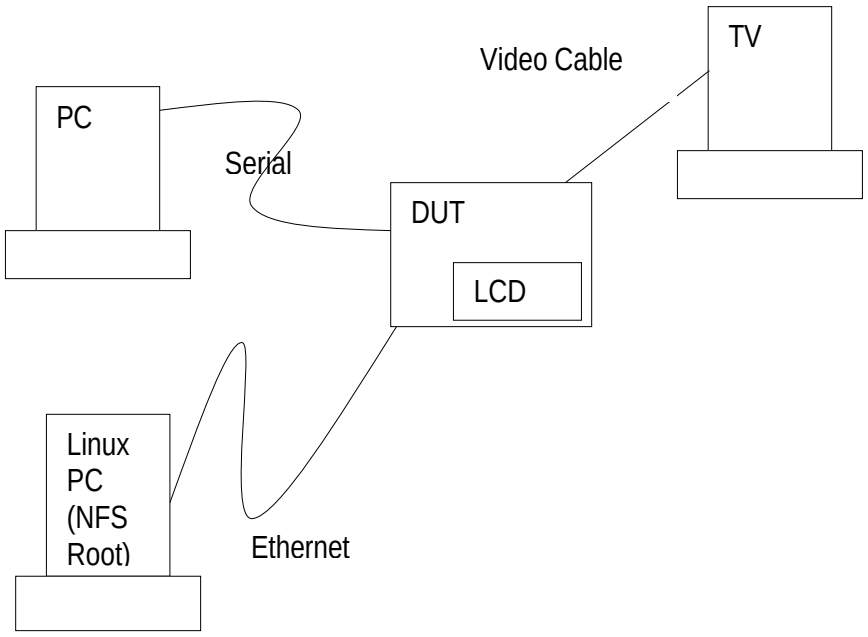

Figure 4-1 DSS Test Setup

4.1.1.3 Test Environment

- DUT (Device Under Test), serial console
- TV to display the output

4.1.1.4 Using the V4L2 Display Application

The V4L2 display performance measurement application can be run either through the command line or by running the script.

4.1.1.4.1 Using the Command line for V4L2 Display

This section describes how to run V4L2 display performance measurement application through the command line.

Go to //./PSP_TEST_HOME/bin and run the executable pspTest with the following input parameters:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String FRv4l2display for display.

iii. Display device (for OMAP35x, can be /dev/video1)

iv. Number of buffers enqueued (any number from three to five)

v. Number of frames to be displayed (can be anything less than 10000)

vi. Display interface can be LCD or TV (S-VIDEO)

Example:

```
./pspTest ThruPut FRv4l2display /dev/video1 3 500 lcd
```

Note:

? To run for more than 10000 frames, change `MAXLOOPCOUNT` in v4l2display_omap35x.c file located in //./PSP_TEST_HOME/performanceTest/throughput/userlevel/video/v4l2/src directory to the required value.

4.1.1.4.2 Using Script for V4L2 Display

To use V4L2 display performance measurement application by running the script:

1. See to OMAP35x Linux PSP Release User Guides and flash the EVM.
2. In Power switch off mode, connect the TV to the DUT using any one of the interfaces (S-video).
3. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
4. See OMAP35x Linux PSP Release User Guides for enabling DSS, compiling and running Linux Kernel.

5. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the command `printenv` to see the environment variables and update them, if necessary.

6. After bootup, modify sysfs as follows:

   For display time-out setting enter `echo 0 > /sys/power/fb_timeout_value and press` **Enter**

   For lcd display enter echo lcd > `/sys/class/display_control/omap_display/video1` and press **Enter**

   For S-Video display, enter echo tv > `/sys/class/display_control/omap_display/video1` and press **Enter**

   For PAL, enter `echo PAL > /sys/class/display_control/omap_display/tv_standard`, and press **Enter**.

   For NTSC, enter `echo NTSC > /sys/class/display_control/omap_display/tv_standard`, and press **Enter**.

7. Setup is now ready to run the display performance test.

8. In the console, run the DSS – V4L2 display throughput script, run_omap35x_v4l2display_tests.sh, available at //./pspTestTarget/scripts/throughput using the command ./run_omap35x_v4l2display_tests.sh and press **Enter**.

V4L2 Display Script Details

The run_omap35x_v4l2display_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on the target:

1. Displays a scrolling color bars pattern for 500 frames on /dev/video1 on the LCD by enqueuing 3 buffers.

2. Running the script provides the frame rate measurements for display of the particular mode.

4.1.1.4.3 Sample Logs for V4L2 Display

Following are the logs for NTSC:

- v4l2display_omap35x.c:v4l2display_perf:385:Running Display:
- v4l2display_omap35x.c:v4l2display_perf:648:Display frame rate: 30.124572

Following are the log for PAL:

- v4l2display_omap35x.c:v4l2display_perf:385:Running Display:

- v4l2display_omap35x.c:v4l2display_perf:648:Display frame rate: 25.091287

4.1.2 FBDEV Framework

4.1.2.1 Performance Parameters

Following DSS performance parameters will be obtained using pspTest tool with the FBDEV framework:

a. Display frame rate for the output stream in fps (frames per sec)

b. Percentage of CPU load

4.1.2.2 Test Setup

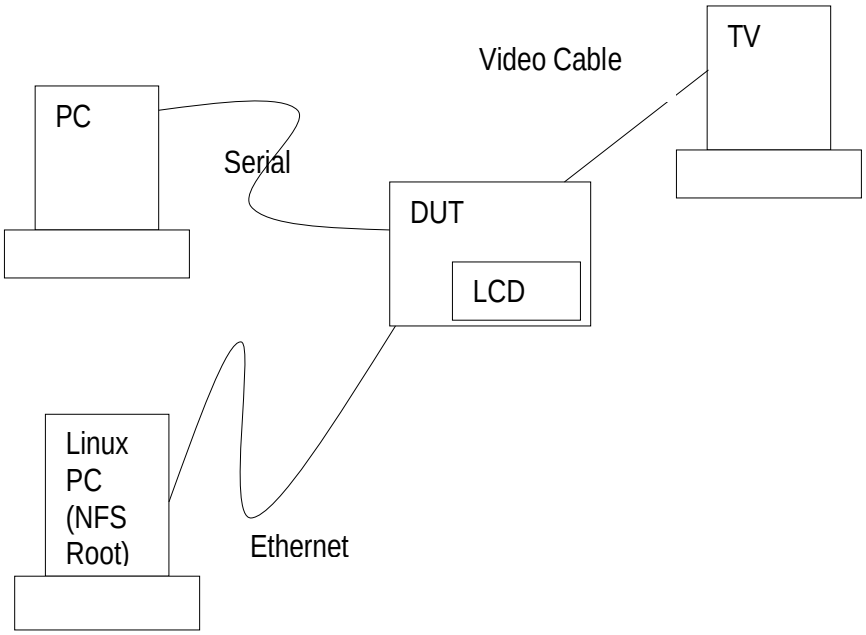

Figure 4-2 DSS Test Setup

4.1.2.3 Test Environment

- DUT (Device Under Test), serial console
- TV to display the output

4.1.2.4 Using the FBDEV Display Application

The FBDEV display performance measurement application can be run either through the command line or by running the script.

4.1.2.4.1 Using the Command line for FBDEV Display

This section describes how to run FBDEV display performance measurement application through the command line.

Go to //./PSP_TEST_HOME/bin and run the executable pspTest with the following input parameters:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String FRfbdevdisplay for display.

iii. Display device (for OMAP35x, can be /dev/fb0)

iv. Number of buffers enqueued (any number from three to eight)

v. Number of frames to be displayed (can be anything less than 10000 and anything more than 500)

vi. Display interface is LCD or TV (SVIDEO)

Example:

```
/ pspTest ThruPut FRfbdevdisplay /dev/fb0 3 500 lcd
```

Note:

? To run for more than 10000 frames, change `MAXLOOPCOUNT` in fbdevdisplay_omap35x.c file located in //./PSP_TEST_HOME/performanceTest/throughput/userlevel/video/fbdev/src directory to the required value.

4.1.2.4.2 Using Script for FBDEV Display

To use FBDEV display performance measurement application by running the script:

1. See to OMAP35x Linux PSP Release User Guides and flash the EVM.
2. In Power switch off mode, connect the TV to the DUT using any one of the interfaces (S-video).
3. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
4. See OMAP35x Linux PSP Release User Guides for enabling DSS, compiling and running Linux Kernel.
5. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the command `printenv` to see the environment variables and update them, if necessary.
6. After bootup, modify sysfs as follows:

   For display time-out setting enter `echo 0 > /sys/power/fb_timeout_value and press` **Enter**

   For lcd display enter echo lcd > `/sys/class/display_control/omap_display/video1` and press **Enter**

For S-Video display, enter echo tv > `/sys/class/display_control/omap_display/video1` and press **Enter**

For PAL, enter `echo PAL > /sys/class/display_control/omap_display/tv_standard`, and press **Enter**.

For NTSC, enter `echo NTSC > /sys/class/display_control/omap_display/tv_standard`, and press **Enter**.

7. Setup is now ready to run the display performance test.
8. In the console, run the DSS – FBDEV display throughput script, run_omap35x_fbdevdisplay_tests.sh, available at //./pspTestTarget/scripts/throughput using the command ./run_omap35x_fbdevdisplay_tests.sh and press **Enter**.

FBDEV Display Script Details

The run_omap35x_fbdevdisplay_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on the target:

1. Displays a scrolling color bars pattern for 500 frames on /dev/fb0 on the LCD by enqueuing 3 buffers.
2. Running the script provides the frame rate measurements for display of the particular mode.

4.1.2.4.3 Sample Logs for FBDEV Display

Following are the logs for NTSC:

- fbdevdisplay_omap35x.c:fbdevdisplay_perf:385:Running Display:
- fbdevdisplay_omap35x.c:fbdevdisplay_perf:648:Display frame rate: 30.346510
- fbdev: display: percentage cpu load: 1.3%

Following are the log for PAL:

- fbdevdisplay_omap35x.c:fbdevdisplay_perf:385:Running Display:
- fbdevdisplay_omap35x.c:fbdevdisplay_perf:648:Display frame rate: 25.293847
- fbdev: display: percentage cpu load: 1.3%

4.2 Video-Capture

This section provides the steps to execute the Video capture performance tests using the scripts or command line utility. It also provides the

performance parameters, test setup information, test environment, and command line arguments.

4.2.1 V4L2 Framework

4.2.1.1 Performance Parameters

Following Video capture performance parameters will be obtained using pspTest tool with the V4L2 framework:

a. Capture frame rate for the input stream in fps (frames per sec)

b. Percentage of CPU load

4.2.1.2 Test Setup

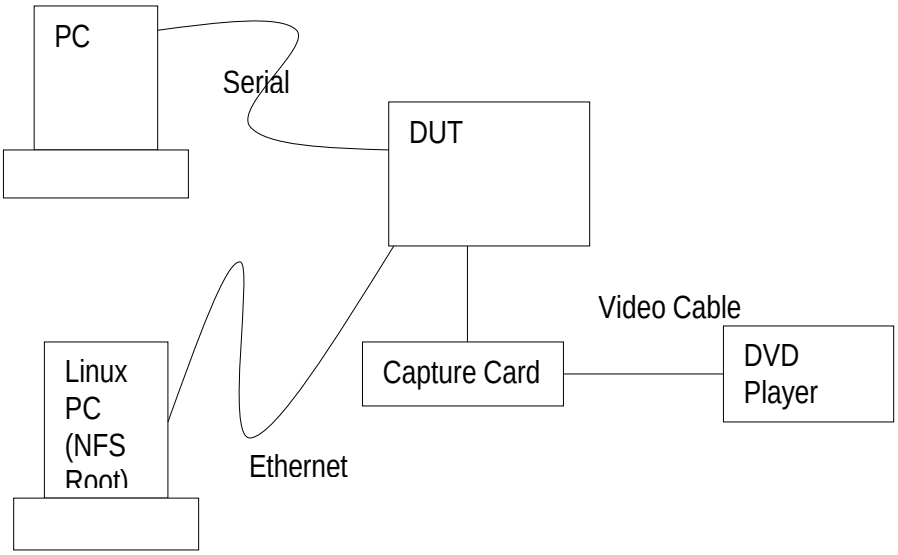

Figure 4-3 Video Capture Test Setup

4.2.1.3 Test Environment

- DUT (Device Under Test), serial console
- OMAP35x Capture card
- DVD Player to give the input stream

4.2.1.4 Using the V4L2 Capture Application

The V4L2 capture performance measurement application can be run either through the command line or by running the script.

4.2.1.4.1 Using the Command line for V4L2 Capture

This section describes how to run V4L2 capture performance measurement application through the command line.

Go to //./PSP_TEST_HOME/bin and run the executable pspTest with the following input parameters:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String FRv4l2capture for capture.

iii. Capture device (for OMAP35x, can be /dev/video0)

iv. Number of buffers enqueued (any number from three to five)

v. Number of frames to be captured (can be anything less than 10000)

Example:

```
/pspTest ThruPut FRv4l2capture /dev/video0 4 500
```

Note:

? To run for more than 10000 frames, change `MAXLOOPCOUNT` in v4l2capture_omap35x.c file located in //./PSP_TEST_HOME/performanceTest/throughput/userlevel/video/v4l2/src directory to the required value.

4.2.1.4.2 Using Script for V4L2 Capture

To use V4L2 capture performance measurement application by running the script:

1. See to OMAP35x Linux PSP Release User Guides and flash the EVM.
2. In Power switch off mode, connect the DVD Player to the DUT using any one of the interfaces (S-Video).
3. The input stream can be of the following resolutions/interfaces:
   a. NTSC on S-Video interface. Expected fps : 30
   b. PAL on S-Video interface. Expected fps : 25
4. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
5. See OMAP35x Linux PSP Release User Guides for enabling DSS, compiling and running Linux Kernel.
6. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the command `printenv` to see the environment variables and update them, if necessary.
7. Setup is now ready to run the capture performance test.
8. In the console, run the V4L2 capture throughput script, run_omap35x_v4l2capture_tests.sh, available at

//./pspTestTarget/scripts/throughput using the command ./run_omap35x_v4l2capture_tests.sh and press **Enter**.

V4L2 Capture Script Details

The run_omap35x_v4l2capture_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on the target:

1. Performs a capture of 500 frames on /dev/video0 using the S-Video interface by enqueuing 4 buffers.
2. Running the script provides the frame rate measurements for capture of the input stream.

4.2.1.4.3 Sample Logs for V4L2 Capture

Following are the log for NTSC:

- v4l2capture_omap35x.c:v4l2capture_perf:385:Running Capture:
- v4l2capture_omap35x.c:v4l2capture_perf:648:Capture frame rate: 30.000051

Following are the log for PAL:

- v4l2capture_omap35x.c:v4l2capture_perf:385:Running Capture:
- v4l2capture_omap35x.c:v4l2capture_perf:648:Capture frame rate: 25.051748

4.3 Audio

This section provides the steps to execute the Audio performance tests using the scripts or command line utility. It also provides the performance parameters, test setup information, test environment, and command line arguments.

4.3.1 Performance Parameters

Following Audio performance parameters will be obtained using pspTest tool:

a. Time taken in seconds for read/write of given data size

b. Data rate for read/write in bytes/sec

c. Percentage of CPU load

4.3.2 Test Setup

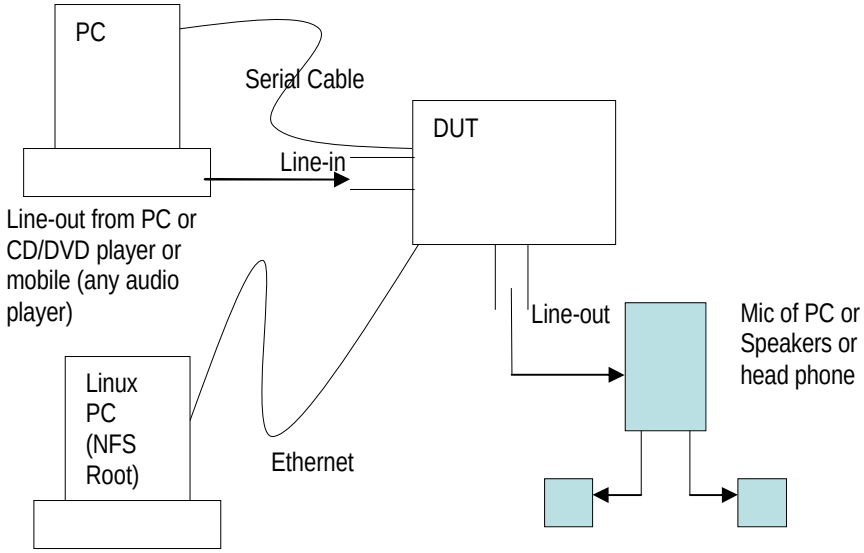

Figure 4-4 Audio Test Setup

4.3.3 Test Environment

A. DUT (Device Under Test), serial console

B. Speaker or head phone or microphone port of PC

C. Line-out of PC or CD/DVD player or any audio player with coaxial out cable

4.3.4 Using The Application

The performance measurement application can be run either through the command line or by running the script.

4.3.4.1 Using the Command Line

pspTest utility supports the following two throughputs for measuring performance of audio through command line:

- Audio Throughput and Percentage of CPU load
- Audio-File Throughput and Percentage of CPU load

Audio Throughput and Percentage of CPU load

Go to //./pspTestTarget/bin and run the executable pspTest with the following configurable parameters as arguments:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String `FRaudioalsawrite` for write or `FRaudioalsaread` for read

iii. Device node for read or write

iv. Sampling rate (Hz) for which performance is carried out

v. Application buffer size (bytes)

vi. Data size (bytes)

Example:

```
./ pspTest ThruPut FRaudioalsaread plughw:0,0 8000 4096
5242880
```

Audio-File Throughput and Percentage of CPU load

Go to //./pspTestTarget/bin and run the executable pspTest with the following configurable parameters as arguments:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String `FRaudioalsawritefromfile` for write or `FRaudioalsareadtofile` for read

iii. Device node for read or write

iv. Absolute path of the file used for writing recorded audio data and reading audio data for playback

v. Sampling rate (Hz) for which performance is carried out

vi. Application buffer size (bytes)

vii. Data size (bytes)

Example:

```
./ pspTest ThruPut FRaudioalsareadtofile "plughw:0,0"
$MOUNT_DIR/perf1.txt 8000 4096 5242880
```

4.3.4.2 Using Script

To use performance measurement application by running the script:

1. See the OMAP35x Linux PSP Release User Guides and flash the EVM.
2. In power switch off mode, connect audio input/line-in and output/line-out devices to the EVM. For further details on connecting hardware, see EVM User Guide.
3. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
4. See the OMAP35x Linux PSP Release User Guides for enabling Audio, compiling, and running Linux Kernel.
5. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the

command `printenv` to see the environment variables and update them, if necessary.

6. Setup is now ready to run the performance test.

7. pspTest utility provides individual scripts for two throughputs for measuring performance and Percentage of CPU load of audio. Steps for running the scripts are:

    a. On the Output/Input console, run the audio throughput script, run_audio_alsa_tests.sh, available at //./pspTestTarget/scripts/throughput using the command ./run_audio_alsa_tests.sh and press **Enter**.

    b. On the Output/Input console, run the audio-file throughput script, run_audio_alsa_filethroughput_tests.sh, available at //./pspTestTarget/scripts/throughput using the command ./run_audio_alsa_filethroughput_tests.sh and press **Enter**.

Script Details

pspTest utility provides individual scripts for two throughputs for measuring performance and Percentage of CPU load of audio. Following are the script details for measuring performance of audio with direct throughput and audio to file throughput.

Audio Throughput Script

The run_audio_alsa_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on the target:

1. Performs read/record of data size 5242880 bytes and application buffer of size 4096 bytes with sampling rates of various values like 8000, 32000, 44100, and 48000 Hz.

2. Performs write/playback of data size 5242880 bytes and application buffer of size 4096 bytes with sampling rates of various values like 8000, 32000, 44100, and 48000 Hz.

Audio-File Throughput Script

The run_audio_alsa_filethroughput_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on the target:

1. Creates a directory /mnt/audio_perfTest on root file system, which will be used for writing the recorded audio data to a file and reading the file for audio playback.

2. Performs read/record of data size 5242880 bytes and application buffer of size 4096 bytes with sampling rates of various values like 8000, 32000, 44100, and 48000 Hz. The recorded data will be written to a file in /mnt/audio_pspTest directory.

3. Performs write/playback of data size 5242880 bytes and application buffer of size 4096 bytes with sampling rates of various values like 8000, 32000, 44100, and 48000 Hz. Audio data used for playback will be read from a file in /mnt/audio_pspTest directory.

4. Deletes the .txt files created while performing read/write.
5. Deletes the directory /mnt/audio_perfTest.

4.3.5 Sample Logs

Following are the logs for read and write:

- audio: read: Word Length in bits: 16
- audio: read: No. of channels per sample: 2
- audio: read: Sampling Rate in Hz: 8000
- audio: read: Duration in Sec: 163.888481
- audio: read: No. of bits/Sec: 255924
- audio: read: percentage cpu load: 1.3%
- audio: write: Word Length in bits: 16
- audio: write: No. of channels per sample: 2
- audio: write: Sampling Rate in Hz: 8000
- audio: write: Duration in Sec: 163.504268
- audio: write: No. of bits/Sec: 256526
- audio: write: percentage cpu load: 1.3%

4.4 OneNAND

This section provides the steps to execute the OneNAND performance tests using the scripts or command line utility. It also provides the performance parameters, test setup information, test environment, and command line arguments.

4.4.1 Performance Parameters

Following OneNAND performance parameters will be obtained using pspTest tool:

a. Time taken in micro seconds for read/write of given data size
b. Data rate for read/write in MBytes/sec
c. Percentage of CPU load

4.4.2 Test Setup

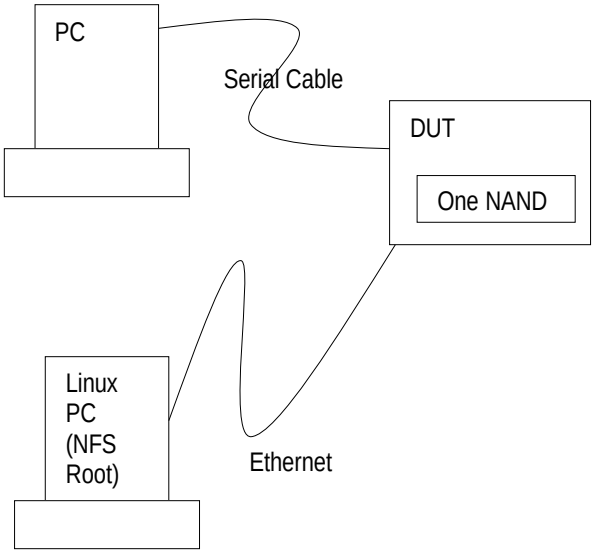

Figure 4-5 OneNAND Test Setup

4.4.3 Test Environment

A. DUT (Device Under Test), serial console

4.4.4 Using the Application

The performance measurement application can be run either through the command line or by running the script.

4.4.4.1 UsingCommand Line

Select proper switch settings (see **Error! Reference source not found.**). Go to //./pspTestTarget/bin and run the executable pspTest with the following configurable parameters as arguments:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String MTDBlkWrite for write or MTDBlkRead for read

iii. Absolute path of the file used for I/O

iv. Application buffer size (bytes)

v. Data size (bytes)

Example:

```
./pspTest ThruPut MTDBlkWrite /dev/mtd4 102400 52428800
```

Note:

? To obtain performance values without cache influence during read operation power cycle the EVM after every write and then read the data written.

? Data size should be less than 123MB (for /dev/mtd4 NAND partition) as NAND available on the OMAP35x EVM is 128MB.

4.4.4.2 Execution by Script

To use the performance measurement application by running the script:

1. See the OMAP35x Linux PSP Release User Guides and flash the EVM.
2. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
3. See the OMAP35x Linux PSP Release User Guides for enabling One NAND device driver, compiling, and running Linux Kernel.
4. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the command `printenv` to see the environment variables and update them, if necessary.
5. Setup is now ready to run the performance test.
6. On the Output/Input console, run the NAND throughput script, run_omap35x_nand_tests.sh, available at //./pspTestTarget/scripts/throughput using the command ./run_omap35x_nand_tests.sh and press **Enter**.

Script Details

The run_omap35x_nand_tests.sh script available at //./pspTestTarget/scripts/throughput performs the following operations on target:

1. Performs read and write of data size 52428800 bytes with an application buffer of various sizes like 102400, 262144, 524288, 1048576, and 5242880 bytes on /dev/mtd4 partition.
2. Prints the performance measurement for each read write operation in MB/s.

4.4.5 Sample Logs

Following are the logs for read and write:

q filewrite: Buffer Size in bytes: 102400

q filewrite: FileSize in bytes: 52428800

q filewrite: Durartion in usecs: 18171310

- filewrite: Mega Bytes/Sec: 2.751788
- filewrite: percentage cpu load: 100.00%

- fileread: Buffer Size in bytes: 102400
- fileread: FileSize in bytes: 52428800
- fileread: Durartion in usecs: 60653471
- fileread: Mega Bytes/Sec: 0.824402
- fileread: percentage cpu load: 100.00%

4.5 Micron NAND

This section provides the steps to execute the Micron NAND performance tests using the scripts or command line utility. It also provides the performance parameters, test setup information, test environment, and command line arguments.

4.5.1 Performance Parameters

Following Micron NAND performance parameters will be obtained using pspTest tool:

a. Time taken in micro seconds for read/write of given data size

b. Data rate for read/write in MBytes/Sec

c. Percentage of CPU load

4.5.2 Test Setup

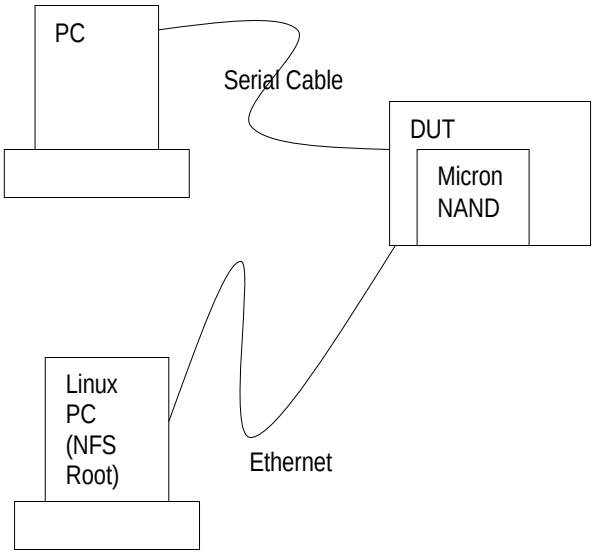

Figure 4-6 Micron NAND Test Setup

4.5.3 Test Environment

- DUT (Device Under Test), serial console

4.5.4 Using the Application

The performance measurement application can be run either through the command line or by running the script.

4.5.4.1 Using the Command Line

Go to //./pspTestTarget/bin and run the executable pspTest with the following configurable parameters as arguments:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String MTDBlkWrite for write or MTDBlkRead for read

iii. Absolute path of the file used for I/O

iv. Application buffer size (bytes)

v. Data size (bytes)

Example:

```
./pspTest ThruPut MTDBlkWrite /dev/mtd4 102400 10485760
```

Note:

- To obtain performance values without cache influence during read operation, power cycle the EVM after every write and then read the data written.
- Data size should be less than 250MB (for /dev/mtd4 Micron NAND partition) as Micron NAND available on the OMAP35x EVM is 256 MB.

4.5.4.2 Execution by Script

To use the performance measurement application by running the script:

1. See the OMAP35x Linux PSP Release User Guides and flash the EVM.
2. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
3. See the OMAP35x Linux PSP Release User Guides for enabling Micron NAND device driver,
4. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the command `printenv` to see the environment variables and update them, if necessary.
5. Setup is now ready to run the performance test.

6. In the Output/Input console, run the Micron NAND throughput script, run_omap35x_nand_tests.sh, available at //./pspTestTarget/scripts/throughput using the command ./run_omap35x_micron_nand_tests.sh and press **Enter**.

Script Details

The run_omap35x_nand_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on target:

1. Performs read and write of data size 10485760 bytes with an application buffer of various sizes like 102400, 262144, 524288, 1048576, and 5242880 bytes on /dev/mtd4 partition.
2. Prints the performance measurement for each read write operation in MB/s.

4.5.5 Sample Logs

Following are the logs for read and write:

- filewrite: Buffer Size in bytes: 102400
- filewrite: FileSize in bytes: 52428800
- filewrite: Durartion in usecs: 18171310
- filewrite: Mega Bytes/Sec: 2.751788
- filewrite: percentage cpu load: 100.00%

- fileread: Buffer Size in bytes: 102400
- fileread: FileSize in bytes: 52428800
- fileread: Durartion in usecs: 60653471
- fileread: Mega Bytes/Sec: 0.824402
- fileread: percentage cpu load: 100.00%

4.6 MMC/SD

This section provides the steps to execute the MMC/SD performance tests using scripts and command line utility. It also provides the performance parameters, test setup information, test environment, and command line arguments.

4.6.1 Performance Parameters

Following MMC/SD performance parameters will be obtained using pspTest tool:

a. Time taken in micro seconds for read/write of given data size

b. Data rate for read/write in MBytes/sec

c. Percentage of CPU load

4.6.2 Test Setup

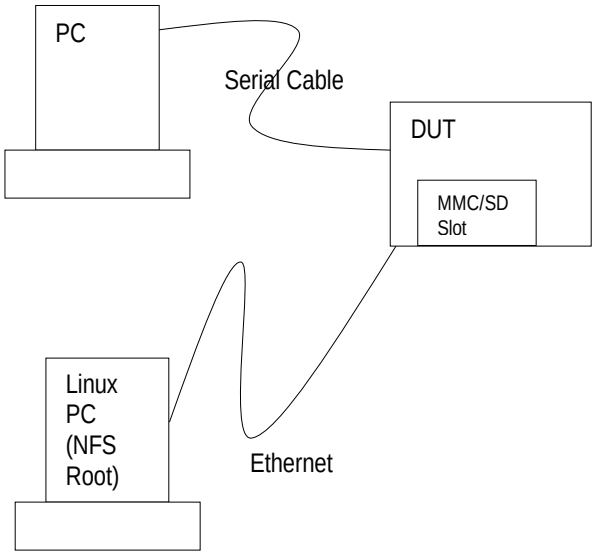

Figure 4-7 MMC/SD Test Setup

4.6.3 Test Environment

- DUT (Device Under Test), serial console
- MMC/SD cards of different sizes

4.6.4 Using the Application

The performance measurement application can be run either through the command line or by running the script.

4.6.4.1 Using Command Line

Go to //./pspTestTarget/bin and run the executable pspTest with the following configurable parameters as arguments:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String TPfswrite for write or TPfsread for read

iii. Absolute path of the file used for I/O

iv. Application buffer size (bytes)

v. Data size (bytes)

Example:

```
./pspTest ThruPut TPfswrite  /mnt/mmcsd/writetest1
102400 104857600
```

Note:

Mount point (/mnt/mmcsd) needs to be created before running the application in command line.

4.6.4.2 Using Script

To use performance measurement application by running the script:

1. See the OMAP35x Linux PSP Release User Guides and flash the EVM.
2. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
3. See OMAP35x Linux PSP Release User Guides for enabling MMC/SD device driver, compiling, and running Linux Kernel.
4. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the command printenv to see the environment variables and update them, if necessary.
5. Setup is now ready to run the performance test.
6. Insert the MMC/SD card of minimum 512MB size in the slot provided on the EVM and check the card gets detected and an entry is created in the /dev directory.
7. Delete any existing partitions if any and create only one partition using fdisk command.
8. In the Output/Input console, run `mkfs -t ext3 /dev/mmcblk0p1` command (this will format the MMC/SD card) and press **Enter**.
9. In the Output/Input console, run the MMC/SD throughput script, run_ mmcsd_tests.sh, available at //./pspTestTarget/scripts/throughput using the command `./run_ mmcsd_tests.sh` and press **Enter**.

Script Details

The run_mmcsd_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on target:

1. Creates a directory /mnt/mmcsd/ on root file system, which will be used for mounting MMC/SD card.
2. Unmounts the directory to ensure it is not mounted already.
3. Mounts /dev/mmcblk0p1 device to the location /mnt/mmcsd/.

4. Performs read and write of data size 104857600 bytes with an application buffer of various sizes like 102400, 262144, 524288, 1048576, and 5242880 bytes.
5. Unmounts using umount /mnt/mmcsd/.

4.6.5 Sample Logs

Following are the logs for read and write:

- filewrite: Buffer Size in bytes: 102400
- filewrite: FileSize in bytes: 104857600
- filewrite: Durartion in usecs: 331939932
- filewrite: Mega Bytes/Sec: 0.301259
- filewrite: percentage cpu load: 100.00%

- fileread: Buffer Size in bytes: 102400
- fileread: FileSize in bytes: 104857600
- fileread: Durartion in usecs: 54305014
- fileread: Mega Bytes/Sec: 1.841281
- fileread: percentage cpu load: 100.00%

4.7 USB MSC Host

This section provides the steps to execute the USB MSC host performance tests using scripts and command line utility. It also provides the performance parameters, test setup information, test environment, and command line arguments.

4.7.1 Performance Parameters

Following USB MSC Host performance parameters will be obtained using this tool:

a. Time taken in micro seconds for read/write of given data size
b. Data rate for read/write in MBytes/sec
c. Percentage of CPU load

4.7.2 Test Setup

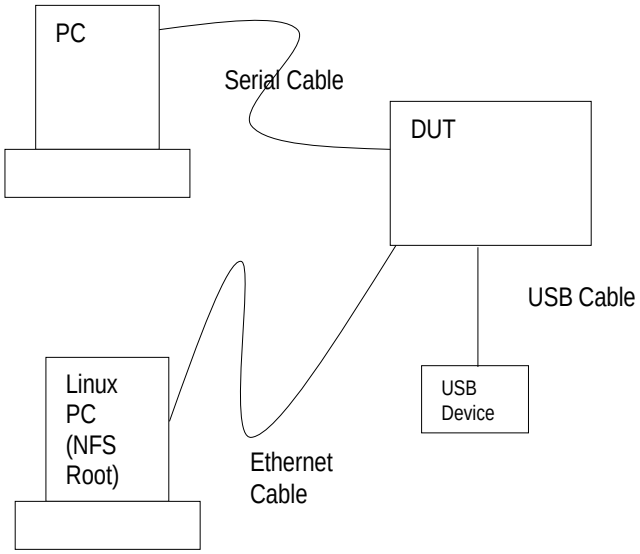

Figure 4-8 USB MSC Host Test Setup

4.7.3 Test Environment

- q DUT (Device Under Test), serial console
- q USB Device (eg: Pendrive)
- q USB Cable

4.7.4 Using The Application

The performance measurement application can be run either through the command line or by running the script.

4.7.4.1 Using Command Line

Go to //./pspTestTarget/bin and run the executable pspTest with the following configurable parameters as arguments:

i. String ThruPut for Throughput performance and Percentage of CPU load

ii. String TPfswrite for write or TPfsread for read

iii. Absolute path of the file used for I/O

iv. Application buffer size (bytes)

v. Data size (bytes)

Example:

```
./pspTest ThruPut TPfswrite  /mnt/usbmsc/perf.txt
102400 104857600
```

Note:

Mount point (/mnt/usbmsc) needs to be created before running the application in command line.

4.7.4.2 Using Script

To run performance measurement application by running the script:

1. See the OMAP35x Linux PSP Release User Guides and flash the EVM.
2. In Power switch off mode, connect the USB Device to the EVM via USB Cable. For further details on connecting hardware, see EVM user guide.
3. Open HyperTerminal/Teraterm (Output/Input console). Set the required settings for HyperTerminal/TeraTerm (see section A.1). Switch on the power for EVM to boot.
4. See the OMAP35x Linux PSP Release User Guides for enabling USB-MSC Host device driver, compiling and running Linux Kernel.
5. Environment variables needs to be updated as specified in OMAP35x Linux PSP Release User Guides. Press **CTRL+C** to stop booting immediately after bootup prints starts. Use the command printenv to see the environment variables and update them, if necessary.
6. Setup is now ready to run the performance test.
7. Verify the presence of /dev/sda1 device entry.
8. In the Output/Input console, run the USB MSC Host throughput script, run_usb_msc_host_tests.sh, available at //./pspTestTarget/scripts/throughput using the command `./run_usb_msc_host_tests.sh` and press **Enter**.(this will format the USB device).

Script Details

The run_usb_msc_host_tests.sh script available at //./pspTestTarget/scripts/throughput will perform the following operations on target:

1. Formats the sda1 partition with ext2 filesystem.
2. Creates a directory as defined by MOUNT_POINT in the script for mounting.
3. Unmounts the mount directory to ensure it is not mounted already.
4. Mounts /dev/sda1 partition to the location defined by MOUNT_POINT in the script.
5. Performs read and write of data size 104857600 bytes with an application buffer of various sizes like 102400, 262144, 524288, 1048576, and 5242880 bytes.

6. Removes the .txt files created while doing read/write.

4.7.5 Sample Logs

Following are the logs for read and write:

- filewrite: Buffer Size in bytes: 102400
- filewrite: FileSize in bytes: 104857600
- filewrite: Durartion in usecs: 6797321
- filewrite: Mega Bytes/Sec: 15.426312
- filewrite: percentage cpu load: 1050401

- fileread: Buffer Size in bytes: 102400
- fileread: FileSize in bytes: 104857600
- fileread: Durartion in usecs: 8347341
- fileread: Mega Bytes/Sec: 12.561796
- filewrite: percentage cpu load: 1050401

Appendix A

General Setup Details

This appendix provides Output/Input console setting details.

A.1 HyperTerminal/TeraTerm Settings

Table A-1. HyperTerminal/TeraTerm Settings

Serial Port	Go to setup > Serial Port and select the following: ? Port: COM1 ? Baud rate: 115200 ? Data: 8 bit ? Parity: none ? Stop: 1 bit ? Flow Control: none
General	Go to setup > General and select the following: ? Default port: COM1 ? Language: English

Appendix B

Adding New Test Case

This appendix provides details on how a test case can be added to PspTest Tool.

B.1 Adding a New pspTest Module

PspTest is designed to be extensible in terms of test methodologies. Under tests directory, each module can be added. Each module should cover a particular driver or a specific multi-driver scenario. To add a new module:

1. Create the new test module for user-level module under //./psp_test_bench/performanceTest/throughput/userlevel/tests/ and for kernel level module under //./psp_test_bench/performanceTest/throughput/kernellevel/tests/. Choose a name that reflects the scope of the test module.
2. Update tests/DIRS with the name of the test module so that make is aware of the new test module.
3. Create two files - Makefile and SOURCES in the test module directory. Any existing test modules can be used as an example. The only file that will need to change is the SOURCES file. Changes to SOURCES are described in the section B.2.

B.2 Adding a New pspTest Command

To add a new pspTest command:

1. Write the tests under the targeted test module. The entry point to a test should be a function that has the following signature:

   ```
   int test_name(int, const char **);
   ```

2. Update SOURCES in the test module to include the new file(s).

> **Note:**
>
> Only .c sources needs to be added here and not headers.

3. Update throughputEngine.c at //./psp_test_bench/main to call the test function.
4. Write the script under //./pspTestTarget/scripts/throughput/ to execute the tests.

B.2.1 Updating throughputEngine.c File

The throughputEngine.c requires the following changes:

1. Declare your test entry function at the top of the file by extending it into throughputEngine.c

2. Get the hash value of the command using the pspTest hash <commandString> command. Define the command in throughputEngine.c file. Add the command to throughputTestArray array with the function pointer pointing to the test entry function. You can use the tests already defined as a sample.