DM3730/AM3703 SOM-LV, DM3730/AM3703 Torpedo SOM, and DM3730/AM3703 Torpedo + Wireless SOM

Software Test Plan

Linux Base BSP 2.x

Revision 3.4

Published: July 27, 2021

Revision History

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
1.0	Dawn Edlund	Initial Revision		**Feb 2, 2011**
2.0	DH, DW, NK DH DH DH DH DH DH DH DH DH DH DH DH DH	Added additional tests for DM37x Changes made per: DM37SQA-37 Changes made per:DM37SQA-40 (1-9) Changes made per: DM37SQA-40 (10 - 12) Changes made per: DM37SQA-46 (1 – 16) Changes made per: DM37SQA-46 (17 – 35) Changes made per: DM37SQA-50 (1 – 13) and DM37SQA-51 Changes made per DM37SQA-57 (1 - 5) and DM37SQA-56 (1 – 12) Changes made per DM37SQA-60 (1 – 17) Changes made per DM37SQA-60 (18 – 22) Changes made per DM37SQA-62 (1 - 36) Changes made per DM37SQA-62 (37 – 46) Added section 19 per DM37SQA-62 # 47		**Aug 6, 2012**
2.1	LN	Corrected section number 19.02 per note in DM37SQA-62		**Sep 5, 2012**
2.2	LN	Changes made per DM37SQA-70 (1 – 15)		**Sep 19, 2012**
2.3	LN	Changes made per DM37SQA-71 (1 – 2)		**Sep 27, 2012**
2.4	DH, LN	Changes made per DM37SQA-73, DM37SQA-78		**Jan 24, 2013**
2.5	DH	Changes made per DM37SQA-78 (32 – 39) and DM37SQA-79		**Apr 18, 2013**
2.6	DH	Changes made per DM37SQA-86 (1 - 15)		**May 15, 2013**
2.7	LN	Changes made per DM37SQA-89 (1-7)		**Jul 11, 2013**
2.8	LN	Changes made per DM37SQA-91, DM37SQA-95		**Aug 8, 2013**
2.9	LN	Changes made per DM37SQA-100, DM37SQA-97, DM37SQA-93		**Oct 10, 2013**
3.0	BSB, DH	Changes made per additions added to LTIB 2.4-3 (DM37LINUX-1014) + formatting adjustments		**Jul 14, 2014**
3.1	DH	Changes made per DM37LINUX-DM37LINUX-1090 (0-40) + formatting adjustments		**Aug 24, 2014**
3.2	DH	Changes made per DM37LINUX-1118 (after 2.4-4 testing)		**Sep 28, 2015**
3.3	BSB	Added erase to 10.01.01 NOR JFFS2 (SOMLV Only) for new NOR flash		**Jan 1, 2018**
3.4	BSB	Replaced dashes with hyphens for all commands so to allow copy/paste of the commands. Changes made related to FCC/IC certification following CONTINUATION-251 Epson, TG2016SMN 26.0000M-ECGNNM, Oscillator TCXO and CONTINUATION-258 Abracon, AMPMAFB-26.0000T, Oscillator. This effort is tracked in [DM37LINUX-1130] Incorporate Updated .ini & .bts files into the wireless driver for -32 model. Bring back the following tests: 16.09.02 .ini file – top half of NVS file 16.09.03 .ini file - lower half Updated all WiFi tests to include loading the module and setting the country to US. Changed from Logic PD to Beacon Embedded template. Updated Bluetooth tests	**RF**	**July 27, 2021**

Table of Contents

1. Introduction

Purpose, Revision Control, Scope, Terminology and Acronyms, and References

1.1 Purpose

This document describes the software tests and software verification procedures for the DM37x Linux Base BSP.

A primary benefit to the DM37x Linux Base BSP platform is a flexible software architecture. While this approach greatly enhances manufacturability and flexibility, it is critical that each software component (and each model supported by the software) be tested fully before the software is released into the field. This document is meant to provide a structured approach to this testing.

Regression testing of this type is statistically significant in a single-unit sample size. These tests are intended to discover repeatable flaws in the software and validate the operation of every available software feature, every communication interface, and the ability of the software to correctly interpret external stimuli.

Besides this introduction, this document is divided into two main sections: functional tests and test support appendices. The appendices are merely present for the reference of the test operator.

1.2 Revision Control

This document was prepared by and is under revision control at Beacon EmbeddedWorks. Changes to this document are to be approved by the software team leader. Should an error be discovered within this document, this document may be updated while test execution is in progress. Should this document change while test execution is in progress, some test cases may be invalidated and repeated at the discretion of the project manager.

1.3 Scope

Testing efforts described here are limited to product tests of software only. It is understood that regression testing does not exhaustively test control algorithms and the durability of hardware components. This document is intended to augment a host of other tests that validate the items outside the scope of this document.

This document is meant to be a guideline for conducting a comprehensive functional test. While it is not practical to test every software scenario, the software test outlined here will yield an adequate investigation into the integrity of software that is to be released.

Software under test (SUT) includes two kernel binary images to be used throughout testing. See version-specific release notes for further information regarding specific binary images. For purposes of test, “standard image” is defined as the designated linux RAMDisk elf image.

1.4 Terminology and Acronyms

Key terms and acronyms used in this document are listed in Table 1: Terminology.

Term	Definition
/RESET	An input pin to the microcontroller that may be used to demand a software and hardware reset.
ACK/NACK	Command response: Acknowledge (ACK) / Negative Acknowledge (NACK).
DUT	Device Under Test
Errata Log	A detailed description of all “non-pass” test items, as recorded by the test operator.
JIRA	An issue tracking system used by Beacon EmbeddedWorks. (http://jira.beaconembedded.com/secure/BrowseProject.jspa?id=10120)
N/A	Not Applicable
Packet	A group of data sent in sequence from one node to another in a relatively short time.
PC	Personal Computer
Test Progress Log	Also “Test Log”. A detailed description of test progress and results as recorded by the test operator.

Table 1: Terminology

1.5 Conventions

This document follows certain typographic conventions:

Convention	Description
Bold	Used in test procedures for commands, programs, and options. All terms shown in bold are typed literally.
Italic	Used in test procedures to show arguments and variables that should be replaced with user-supplied values. Italic is also used to introduce new terms, indicate filenames and directories, and to highlight comments in examples.
Bold Italic	Used in test procedures to indicate a particular key on the keyboard to be pressed.
`Constant Width`	Used in test procedures to show the contents of files or the output from commands.
`Constant Width Bold`	Used in test procedures to show commands or other text that should be typed literally by the user.
`Constant Width Italic`	Used in test procedures to show text that should be replaced with user-supplied values.
`losh>`	Used in test procedures to show the LogicLoader shell prompt (>).
[ ]	Square brackets orange in color with Rx-xxx enclosed are used in test procedures to indicate applicable SRS requirement tested in that particular test procedure.
//	When used following a command, text following // are comments.
< >	Used to indicate information to be replaced with user-supplied values. The <> should never be entered.
{ }	When used with text light gray in color, indicates Clear Text. Clear Text is used in test procedures to provide addition information and/or clarification.

01.01.01 - Table 2: Conventions

1.6 References

- DM37X_Linux_Base_BSP_SRS
    - http://viewvc.beaconembedded.com/viewvc.cgi/software/Linux/DM37x/admin/Requirements/DM37X_Linux_Base_BSP_SRS.pdf?root=eps_svn&view=log
- Required PC Software
- Linux OS (for this test plan, Beacon EmbeddedWorks used Debian)
- TFTP server
- Minicom or some other serial terminal emulator program (for Linux)
- Telnet installed
- Wireless Ethernet card (if required)
- HP USB Formatting Tool, v2.0.6

2. Test Setup

Environment and Test Setup Diagram

Environment

All software tests should be conducted at room temperature and ambient pressure.

Test Setup

For all tests, the Debug Serial UART should be connected to a PC where you can interact with the DUT. Various tests will have you connecting other peripherals as required.

3. Required Equipment

General Supplies and Measurement Instruments

General Supplies

- DM3730 Development Kit, including
  - Baseboard
  - One of the following System on Modules (SOMs):
    - DM3730/AM3703 SOM-LV
    - DM3730/AM3703 Torpedo SOM
    - DM3730/AM3703 Torpedo + Wireless SOM
  - Beacon EmbeddedWorks LCD-4.3-WQVGA-20R Display (28)
  - Null-modem serial cable
  - Ethernet cable
  - USB Mini-B to Standard-A cable
- PC configured with Linux
- Headphones or speakers with 1/8th inch stereo jack
- Leopard Imaging LI-5M04 camera
- Beacon EmbeddedWorks LCD-4.3-WQVGA-10R Display (15)
- Logictech C210 USB Webcam
- USB Devices
  - USB keyboard

- USB mouse
- USB mass storage devices (flash/thumb drives):
  - High capacity thumb drive
    - SanDisk 16 GB Cruzer USB 2.0 Flash Drive SDCZ36-016G-A11
  - Non-high capacity thumb drive
    - SanDisk 256 MB Cruzer Mini USB 2.0 Flash Drive SDCZ2-256-A10
  - Bus-powered USB mass storage device
    - Seagate Expansion 500 GB USB 2.0 Portable External Hard Drive ST905004EXA101-RK
  - Non-bus powered USB mass storage device
    - Western Digital WD Elements 1.5 TB USB 2.0 Desktop External Hard Drive
- USB audio device (MP3 player, iPod, etc)
- USB webcam
- 1.1 and 2.0 devices
- Low, full, and high speed devices
- Powered and unpowered hubs
- USB Mini-B to Standard-A (male) cable (supplied in development kits)
- USB OTG Mini-A to Standard-A (female) adapter (*Mini-A ID Pin is grounded)*

+ SD cards
+ Two MMC/SD storage cards: one <=2 GB, and one >=4 GB.
+ SDIO wireless card
+ SDIO serial card
+ SD card reader/writer
+ 1 GB switch (CISCO SLM2008)
+ Aardvark I2C/SPI activity board (with connectors)
+ Crossover Ethernet cable

- Wireless router (a/b/g/n capable)
    - D-Link DGL-4500
- Expansion board

Measurement Equipment

Stopwatch or timer

4. Testing Procedure Requirements and Guidelines

Acceptable Test Results, Test Progress Reporting, Failure Reporting, and Issue Tracking

In order to provide constructive feedback to the software development team, this software test plan must be conducted in a manner that yields usable results. All test results must be recorded in written and/or electronic form within the Test Progress Log and all failures must be documented and entered in the issue tracking system (JIRA). Failures may also be accompanied by a verbal description at the discretion of the test engineer.

This test plan assumes that the test engineer will use his or her intuition and creativity to conduct the test plan in a way that is effective and efficient. The test engineer has the freedom to conduct the test plan in their own style, provided that the instructions, requirements, and guidelines are followed as described in this document.

In some cases, undesired or unexplained operation may be detected while no specific testing is in progress. It is expected that the test engineer will record any undesired or unexplained operation in the Test Progress Log as if it were a failure that was detected during a specific test.

Acceptable Test Results

All test results must be recorded within the Test Progress Log. Reference to an external document (e.g., written documentation generated by the test engineer, email, and photos) is permitted, but discouraged.

Each test case has its own Pass/Fail conditions. It is possible for the DUT to produce the desired outcome but fail to meet the required response time. In this event, the condition tested should be listed as a failure. Distinctions made between failures due to timing and failures due to unexpected operation should be documented in the Test Progress Log. For each test case, indicate the test result with exactly one of the following quantifiers:

Pass - Test passes without any indication of failure.

Fail - Test fails and may represent a significant issue.

N/A – Test is not applicable. See comments for reason the test is N/A.

Every test result must be recorded in the Test Progress Log. For each test case that does not earn a Pass rating, an additional entry indicating the JIRA issue identifier is required. The JIRA issue shall describe the undesired operation. All non-passing test results must be retested or dismissed by the program/project manager before the software revision under test may be released.

Test Progress Reporting

The primary means for the test engineer to explain test progress, sequencing, and timing is the Test Progress Log. The Test Progress Log is to be appended after each individual test execution. Transferring data from a laboratory notebook into the Test Progress Log in larger sections is permissible but is not recommended.

The Test Progress Log may be completed in written or electronic form. All fields in the Test Progress Log must be completed for each test case that is evaluated, regardless of any information that may seem redundant.

Tests may be repeated at any time. Test repetitions and results should be recorded in the Test Progress Log as if they had not been conducted previously.

The Test Progress Log will track testing progress, test sequencing, and test results. See the Test Progress Log Procedure section for details and examples of appropriate entries.

Failure Reporting

The primary means for the test engineer to explain failures, undesired operation, and unexplained operation to the development team is via JIRA. All non-passing test results must be marked with a failure indicator in the Test Progress Log and be documented in JIRA. The test engineer is also responsible for including all unexplained operation in the issue, even if the unexplained operation was not witnessed while a specific test was in progress.

The Test Progress Log may be completed in written or electronic form. All fields in the Test Progress Log must be completed for each entry, regardless of any information that may seem redundant.

A test that successfully meets all required conditions should not be recorded in the issue tracking system.

Issue Tracking

The software team leader is responsible for updating the open items in JIRA. The Test Progress Log from this test plan will be the primary source of issue reporting to the development team and should be completed at the completion of test execution activities.

5. Functional Tests

Specific Test Cases, Organized by Feature

Unless specified otherwise, no test case is dependent on another test case. The tests need not be conducted in any particular order.

Each test case must be recorded in the Test Progress Log as it is conducted. See the Test Progress Log Procedure for usage requirements and guidelines.

5.1 Test 01 Display Driver Functional Tests

Purpose

This test verifies that the functional requirements of the Display. The DM3730 products support two Beacon EmbeddedWorks displays: LCD-4.3-WQVGA-20R (display 28), LCD-4.3-WQVGA-10R (display 15).

Setup

Load any standard image

Test Steps

Perform the following subtests and record results in the action log for each display as follows:

All tests in Section 1 are to be run with display 28 (4.3") connected unless otherwise directed.

*NOTE: Section 01.01, 01.02, and 01.03 are to be done using Beacon EmbeddedWorks LCD touchscreens. When using LCD screens, jumper JP2 on the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 Torpedo + Wireless SOM and jumper JP5 on the DM3730/AM3703 SOM-LV must be in the **lcd 1-2 pin position**.*

01.01 Linux Frame Buffer Support

01.00.00 Default to 4.3" display

[DISP-01-004]

Boot and break into U-Boot.

Reset the board to the default state:

```
OMAP Logic # env default -f; saveenv
OMAP Logic # printenv display
```

Verify that the display environmental variable is set to 28

Boot the SOM:

```
OMAP Logic # boot
```

Verify that the board boots and you can login

Verify that the display variable is correctly fed to the kernel by reading the == Kernel bootargs == section of the output. There is a delay in while loading right after the boot command and then the == Kernel bootargs == section is right after. The "display" variable should be on the first line of the section.

01.01.01 Console Frame Buffer

[DISP-01-001] [DISP-01-006]

Load image and break into U-Boot.

Set the console in the kernel parameter and boot:

```
OMAP Logic # setenv display 28
OMAP Logic # setenv consoledevice tty0
OMAP Logic # boot
```

Verify that the boot text is displayed on the device display.

01.01.02 Graphic Performance

[DISP-01-001] [DISP-01-002] [PERF-11-018]

Load any standard image and login:

```
DM-37X# df_dok
```

Verify the image is displayed without errors.

Hit ctrl-c to stop the df_xxx

Enter the results of the graphics performance testing on the performance sheet of the TPL

Load any standard image and login:

```
DM-37X# df_knuckles
```

Verify the image is displayed without errors.

Hit ctrl-c to stop the df_xxx

Load any standard image and login:

```
DM-37X# df_fire
```

Verify the image is displayed without errors.

Hit ctrl-c to stop the df_xxx

01.02 Beacon EmbeddedWorks Display Kit Support

01.02.01 LCD-4.3-WQVGA-20R (28)

[DISP-01-003] [DISP-01-004] [DISP-08-001] [DISP-01-007] [DISP-01-009]

Connect the LCD-4.3-WQVGA-20R (28) display.

Enter Session

In U-Boot, set the 4.3” display and boot:

```
OMAP Logic # setenv display 28
OMAP Logic # boot
```

On the device, login and execute draw-test in the background:

```
DM-37x# draw-test &
```

Verify draw-test executed successfully, that is, the bars appear clean without misalignment or other artifacts.

End Session

Enter Session

In U-Boot, set the 4.3” display and boot:

```
OMAP Logic # setenv display 480:271:2:3:4:2:42:11:9000:0x100000:16
OMAP Logic # boot
```

On the device, login and execute draw-test in the background: `DM-37x# draw-test &`

Verify draw-test executed successfully, that is, the bars appear clean without misalignment or other artifacts.

End Session

For each from x = {0,1,2,3}

Enter Session

In U-Boot, set the 4.3” display and rotation parameters (each one), and boot:

```
OMAP Logic # setenv rotation x {where x=0 for normal, 1 for 90, 2 for 180, 3 for 270 degrees}
OMAP Logic # setenv otherbootargs ${otherbootargs} consoleblank=0
OMAP Logic # boot
```

On the device, login and execute draw-test in the background: `DM-37x# draw-test &`

Verify draw-test executed successfully, that is, the bars appear clean without misalignment or other artifacts.

Verify draw-test executed successfully, and is rotated 0, 90, 180 and 270 degrees. (rotation = 0,1,2,3)

End Session

End For each from x = {0,1,2,3}

01.02.02 LCD-6.4-VGA-10R (5)

<Test Removed.>

01.02.03 LCD-3.6-QVGA-10R (3)

<Test Removed.>

01.02.04 Invalid Display Value

<Test Removed>

For each from [LCD-4.3-WQVGA-20R (28), LCD-4.3-WQVGA-10R (15)]:

01.03 Backlight On / Off / Variable Tests

01.03.00 df_andi

[DISP-01-002]

Load any standard image and login.

Run the df_andi application:

```
DM-37X# df_andi &
```

Verify the image is displayed without errors.

Keep the image running through all of the backlight testing tests.

01.01.02 -

01.03.01 Turn backlight off

[DISP-01-014]

Turn off the backlight:

```
DM-37x# echo 0 > /sys/class/backlight/omap3logic/brightness
```

Verify the backlight turns off.

01.03.02 Turn backlight on

[DISP-01-014]

Turn the backlight on full:

```
DM-37x# echo 255 > /sys/class/backlight/omap3logic/brightness
```

Verify the backlight turns on.

01.03.03 Variable Brightness

[DISP-08-017]

Adjust the backlight to partial brightness:

```
DM-37x# echo xxx > /sys/class/backlight/omap3logic/brightness
```

{where xxx is a number between zero and 255 to change brightness levels}

Verify variable brightness is functional by verifying boundaries and mid range values.

End For each from [LCD-4.3-WQVGA-20R (28), LCD-4.3-WQVGA-10R (15)]

01.03.04 Backlight off due to Inactivity

<Test combined with 01.07.02 (11/5/2012)>

01.04 HDMI/DVI Linux Frame Buffer Support

NOTE: Sections 01.04 and 01.05 are to be run using external monitors in DVI (16-bit) or HDMI (24-bit) mode, not Beacon EmbeddedWorks LCD touchscreens.
When using external monitors, jumper JP2 on the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 Torpedo + Wireless SOM and jumper JP5 on the DM3730/AM3703 SOM-LV must be in the LCD 1-2 pin position when using the LCD to DVI attachment and testing in DVI.
When using external monitors, jumper JP2 on the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 Torpedo + Wireless SOM and jumper JP5 on the DM3730/AM3703 SOM-LV must be in the HDMI 2-3 pin position when using HDMI port and testing in HDMI.

FOR TESTS 01.04.01 AND 01.04.02:
Do both tests on all board configurations in VGA mode (640x480)
On each of the "KIT" boards, repeat the tests with the remaining resolutions:
SVGA (800x600), XGA (1024x768), 720p (1280x720)
Ensure that HDMI and DVI are run on both the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 SOM-LV: One DM3730/AM3703 SOM-LV runs HDMI, one runs DVI, any others can run either. The setup is the same with the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 Torpedo + Wireless SOM. If you only have one of either board, you must run both HDMI and DVI on that board.

FOR DVI testing: Use the 24 bit interface provided by the HDMI interface. Do NOT use the LCD to DVI adapter.

01.04.01 Console Frame Buffer

[DISP-09-010] [DISP-09-011]

Load image and break into U-Boot.

Set the console in the kernel parameter and boot:

```
OMAP Logic # setenv display xxxx  {vga, svga, xga, 720p }
```

i.e. setenv display vga-dvi or setenv display vga-hdmi

setenv display vga-16 or setenv display vga-24

```
OMAP Logic # boot
```

Verify that the SOM booted the Linux image correctly.

01.04.02 DirectFB

[DISP-01-002] [DISP-09-010] [DISP-09-011]

Load any standard image and login:

```
DM-37X# df_andi
```

Verify the image is displayed without errors.

Hit ctrl-c to stop the df_xxx

01.05 Set display kit at kernel build time

01.05.01 Verify that the display kit can be set at kernel boot time.

[DISP-01-008]

<Verify via code inspection>

01.06 Video driver functionality after resuming from suspend

01.06.01 Verify display of image after resuming from suspend.

[DISP-01-013] [DISP-01-016]

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Display image:

```
DM-37X# df_knuckles &
```

Verify the image is displayed without errors.

01.07 Display controller

01.07.01 Turning the display controller on and off at runtime.

[DISP-01-011]

Put an image on the screen.

```
DM-37x# draw-test &
```

Turn off the display controller.

```
DM-37x# echo 0 > /sys/devices/omapdss/display0/enabled
```

Verify that the screen went black.

Verify that when you shine a flashlight into the screen, you cannot see the lines of the draw-test.

Turn on the display controller.

```
DM-37x# echo 1 > /sys/devices/omapdss/display0/enabled
```

Verify that the draw test is displayed properly.

01.07.02 Display controller off due to inactivity

[DISP-01-012] [DISP-01-015]

POR

Wait until the screen goes dark due to inactivity (This can take up to 10 min).

Verify that the screen is black. Shine a flashlight into the screen and verify that tux image is not there.

5.2 Test 02 Audio Output Tests

Purpose

This test verifies that the functional requirements of audio output.

Setup

As preparation for this test, execute the following in sequence:

- Load any standard image
- Copy the following audio files to a desktop PC and an SD card.
  - LPCM-test-8bps.wav
  - LPCM-test-16bps.wav
  - LPCM-test-mono.wav
  - LPCM-test-stereo.wav
  - LPCM-test-{8,11,16,22,44,48}.KHz.wav
  - Any MP3 file

Test Steps

Perform the following subtests and record results in the action log:

02.01 Sounds

02.01.01 Playback

[AUDIO-01-001] [AUDIO-01-003] [AUDIO-01-004] [AUDIO-01-005] [AUDIO-01-006]

Copy the audio files for test to a SD card formatted properly.

Boot the device.

Connect headphones or speakers to the device.

Insert the SD card with the audio files.

```
DM-37x# mount
```

Verify the SD card is mounted (/dev/mmcblk0p1 on /mnt/mmcblk0p1 type vfat... is displayed)

For .wav files:

```
DM-37x# aplay /mnt/mmcblk0p1/<audio filename>
```

Verify that the audio on the device to the audio on the computer and verify they sound the same.

End For .wav files

For .MP3 files:

```
DM-37x# mp3play /mnt/mmcblk0p1/<audio filename>
```

Verify that the audio on the device to the audio on the computer and verify they sound the same.

End For .MP3 files

02.01.02 Volume

[AUDIO-01-006] [AUDIO-01-008]

Play an audio file in the background:

```
DM-37x# aplay /mnt/mmcblk0p1/<filename> &
```

Adjust the various 'amixer' controls:

```
DM-37x# amixer set "DAC1 Digital Fine" 15
```

Verify that the sound in the headphones responds appropriately and is lower in volume

```
DM-37x# amixer set "DAC1 Digital Fine" 60
```

Verify that the sound in the headphones responds appropriately and is higher in volume

```
DM-37x# amixer set "DAC1 Analog" 4
```

Verify that the sound in the headphones responds appropriately and is lower in volume

```
DM-37x# amixer set "DAC1 Analog" 18
```


Verify that the sound in the headphones responds appropriately and is higher in volume

```
DM-37x# amixer set "DAC1 Digital Coarse" 0
```

Verify that the sound in the headphones responds appropriately and is lower in volume

```
DM-37x# amixer set "DAC1 Digital Coarse" 2
```

Verify that the sound in the headphones responds appropriately and is higher in volume

02.01.03 Mute

[AUDIO-01-006] [AUDIO-01-007]

Play an audio file in the background:

```
DM-37x# aplay /mnt/mmcblk0p1/<filename> &
```

Mute the audio:

```
DM-37x# amixer set Master mute
```

Verify mute is set (sound may be very faint)

Unmute the audio:

```
DM-37x# amixer set Master unmute
```

Verify mute is not set and sound returns at the prior level

02.02 Audio driver functionality after resuming from suspend

02.02.01 Verify audio files play properly after resuming from suspend.

[AUDIO-01-011]

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Play an audio file:

```
DM-37x# aplay /mnt/mmcblk0p1/<audio filename>
```

Verify the audio file plays properly.

POR

02.03 Audio Input

02.03.01 Record and playback

[AUDIO-16-015]

Plug a microphone or headphones into the Audio in port of the DUT.

If you are using a microphone rather than a Line-in device:

Set the bias for mic-in:

```
DM-37x# i2cset -f -y 1 0x49 0x04 0x04
```

Record some sound:

```
DM-37x#      arecord -f cd /tmp/test.wav
```

Hit CTRL-c to stop recording.

Plug headphones or speakers into the Audio out port of the DUT

Play the file that was recorded

```
DM-37x# aplay /tmp/test.wav
```

Verify that the audio plays and sound like what you recorded.

5.3 Test 03 Wired Ethernet Tests

Purpose

This test verifies that the functional requirements for wired Ethernet are met.

Setup

As preparation for this test, execute the following in sequence:

- Connect the board to an Ethernet network that has a DHCP server running.
- Determine a static IP address that can be used for these tests.
- Load any standard image

Test Steps

Perform the following subtests and record results in the action log:

03.01 Configuring an IP

03.01.01 Verify boot with DHCP is successful.

<Test combined with 03.01.03 (11/5/2012)>

03.01.02 Verify boot with Static IP address is successful.

[ETHER-01-004] [ETHER-01-005] [ETHER-01-008] [ETHER-01-009] [ETHER-01-010] [ETHER-01-013]

Enter Session

In U-Boot, set kernel parameters:

```
OMAP Logic # setenv otherbootargs ${otherbootargs}
ip=<ipaddr>:::<gatewayip>:<netmask>::
```

```
OMAP Logic # boot
```

Login when boot is complete.

```
DM-37x# ifconfig
```

Verify the static ipaddress was obtained

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was successful.

POR

03.01.03 Verify DHCP ifup/ifdown successfully initiates and terminates the network connection.

[ETHER-01-001] [ETHER-01-002] [ETHER-01-003] [ETHER-01-004] [ETHER-01-006] [ETHER-01-008] [ETHER-01-009] [ETHER-01-010] [ETHER-01-013]

Enter Session

In U-Boot, set kernel parameters:

```
OMAP Logic # setenv otherbootargs ${otherbootargs} ip=::::::dhcp (note:  there are 6
colons)
```

```
OMAP Logic # boot
```

Login when boot is complete.

```
DM-37x# ifconfig
```

Verify an ip address was obtained

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was successful

Use vi to Edit /etc/network/interfaces to configure eth0 for a DHCP IP:

```
DM-37x# vi /etc/network/interfaces
```

Make the ‘iface eth0’ line look as follows if it does not already:

```
iface eth0 inet dhcp
```

Save the changes and quit vi by hitting the ”**escape**” key and typing “**:x<enter>**”

Bring down and then back up the eth0 interface:

```
DM-37x# ifdown eth0
```

```
DM-37x# ifup eth0
```



```
DM-37x# ifconfig eth0
```

Verify the network is operational.

Ping a known host:

```
DM-37x# ping <known host>
```

Verify the network is operational.

Bring Ethernet down:

```
DM-37x# ifdown eth0
DM-37x# ifconfig eth0
```

Verify the network is not operational.

Ping a known host:

```
DM-37x# ping <known host>
```

Verify the network is not operational.

03.01.04 Verify Static ifup/ifdown successfully initiates and terminates the network connection.

[ETHER-01-001] [ETHER-01-002] [ETHER-01-003] [ETHER-01-005] [ETHER-01-008] [ETHER-01-009] [ETHER-01-010] [ETHER-01-013]

Ensure test 03.01.03 has been performed before this test

Use vi to Edit /etc/network/interfaces on the device, to configure eth0 for a static IP:

```
DM-37x# vi /etc/network/interfaces
```

Make the 'iface eth0' line look as follows if it does not already:

```
iface eth0 inet static
     address <ipaddr>
     netmask <netmask>
     gateway <gatewayip>
```

Where "<ipaddr>" is the static IP address (check with IT to make sure you don't pick another machine's IP address causing network conflict), "<netmask> is the appropriate netmask (255.255.255.0 probably), and "<gatewayip>" is the IP address of the gateway for the network the board is connected to:

Save the changes and quit vi by hitting the "**escape**" key and typing "**:x<enter>**"

Bring down and then back up the eth0 interface:

```
DM-37X# ifdown eth0
DM-37X# ifup eth0
DM-37X# ifconfig eth0
```

Verify the network is operational.

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was successful.

Bring down Ethernet:

```
DM-37x# ifdown eth0
DM-37X# ifconfig eth0
```

Verify the network is not operational.

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was not successful.

POR

03.01.05 Verify ifconfig successfully initiates and terminates the network connection.

[ETHER-01-001] [[ETHER-01-003] [ETHER-01-005] [ETHER-01-008] [ETHER-01-009] [ETHER-01-010] [ETHER-01-013]

Boot and login

Using the static ip address entered above, bring up the network connection

```
DM-37X# ifconfig eth0 <ipaddr> netmask <netmask>
DM-37X# ifconfig eth0
```

Verify the network is operational. (UP BROADCAST RUNNING MULTICAST)

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was successful.

Bring Ethernet down:

```
DM-37x# ifconfig eth0 down
DM-37X# ifconfig eth0
```

Verify the network is not operational (BROADCAST MULTICAST)

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was not successful.

03.01.06 Verify boot does not hang when using a Static IP address.

[ETHER-01-005] [ETHER-01-019]

**NOTE: You can log Teraterm and turn on timestamps to get accurate boot times*

On the Build machine, configure ltib to start networking:

```
logic@logic-<ltib folder location>/./ltib -c
```

Target System configuration

Options →

[*] start networking (Select)

Network setup → (Hit Enter)

[ ] get network parameters using dhcp (unselect)

(192.168.0.254) IP address (put in the IP address you want)

(255.255.255.0) netmask

(192.168.0.255) broadcast address

(192.168.0.1) gateway address

(192.168.0.1) nameserver IP address

Fill in the IP address and other information needed for a static IP address connection

Keep exiting until asked to save and select YES

When build is complete, make boot cards:

```
logic@logic<ltib folder location>/ ./bin/mkLogicFATcard.sh -c
```

Insert the boot card and boot to the Linux prompt using the newly build Static IP address image with the Ethernet cable inserted and with the Ethernet cable NOT inserted.

Write the boot times on the PERF page of the TPL.

Verify that the login prompt is displayed on the terminal interface within 1 minute if booting from NAND and within 1 minute 15 seconds if booting via SD Boot card for both the cable in and cable out boots.

Login and do an ifconfig command:

```
DM-37x# ifconfig
```

Verify that the ifconfig command is successful shows the static IP address you put into the build.

03.01.07 Verify boot does not hang when using DHCP IP address.

[ETHER-01-019]

**NOTE: You can log Teraterm and turn on timestamps to get accurate boot times*

On the Build machine, configure ltib to start networking:

```
logic@logic-<ltib folder location>/./ltib -c
```


Target System configuration

Options → (Hit Enter)

[*] start networking (Select)

Network setup → (Hit Enter)

[*] get network parameters using dhcp (select)

Keep exiting until asked to save and select YES

When build is complete, make boot cards:

```
logic@logic<ltib folder location>/ ./bin/mkLogicFATcard.sh -c
```

Insert the boot card and boot to the Linux prompt using the newly build DHCP image with the Ethernet cable inserted and with the Ethernet cable NOT inserted.

Write the boot times on the PERF page of the TPL.

Verify that the login prompt is displayed on the terminal interface within 1 minute if booting from NAND and within 1 minute 15 seconds if booting via SD Boot card for both the cable in and cable out boots.

Login and do an ifconfig command:

```
DM-37x# ifconfig
```

Verify that the ifconfig command is successful shows an IP address when the Ethernet cable is plugged in and does not show an IP address when the Ethernet cable is not plugged in.

03.02 inetd

03.02.01 Verify telnet to the device daytime port is successful.

[ETHER-01-007] [ETHER-01-009] [ETHER-01-010]

Enter Session

In U-Boot, set kernel parameters:

```
OMAP Logic # setenv otherbootargs ${otherbootargs} ip=::::::dhcp (note:  there are 6 colons)
```

```
OMAP Logic # boot
```

Login when boot is complete

Use vi to edit /etc/inetd.conf

```
DM-37x# vi /etc/inetd.conf
```

Uncomment the daytime line in /etc/inetd.conf (just the tcp one).

Save the changes and quit vi by hitting the "**escape**" key and typing "**:x<enter>**"

Lookup pid of inetd:

```
DM-37x# ps agux
```

Send a HUP (hangup) signal to inetd:

```
DM-37x# kill -HUP <pid of inetd>
```


From desktop telnet to the DM37x board at the daytime port(13):

```
DM-37x# telnet <ipaddr> 13
```

Verify that a date and time stamp is returned (it likely won't be correct) and that the telnet session closes.

03.02.02 Verify telnet to the device using telnetd-tcp is successful.

[ETHER-01-007] [ETHER-01-009] [ETHER-01-010]

Ensure that Test 03.02.01 has been executed previous to this test.

From desktop telnet to the DM37X board:

```
DM-37x# telnet <ipaddr>
```

Verify successful login and exit AS uid: user, pw: user

03.03 IEEE802.3 Feature Tests

01.01.03 -

03.03.01 Change MAC

[ETHER-01-009] [ETHER-01-010] [ETHER-01-011]

Boot the device and login

Note the mac address of the device:

```
DM-37x# ifconfig
```

If eth0 is up and running, enter:

```
DM-37x# ifconfig eth0 down
```

Change the MAC address. Choose a MAC address that is not in use (use the MAC for the 802.11 interface):

```
DM-37x# ifconfig eth0 hw ether 00:08:EE:xx:xx:xx (where xx.xx.xx is chosen address)
```

Start eth0:

```
DM-37x# ifconfig eth0 up
DM-37x# ifconfig eth0
```

Verify hardware address is correct:

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping is successful.

Restore the original MAC address noted above.

```
DM-37x# ifconfig eth0 hw ether 00:08:EE:xx:xx:xx (where xx.xx.xx is original address)
```

Start eth0:

```
DM-37x# ifconfig eth0 up
DM-37x# ifconfig eth0
```

Verify hardware address is correct (original address):

POR

03.03.02 Change/force Speed / Duplex

[ETHER-01-012] [ETHER-01-014] [ETHER-01-015]

Boot the device and login

```
DM-37x# ifconfig
```

Ensure Ethernet is running

```
DM-37x# ifconfig eth0 up
```

```
DM-37x# ethtool -s eth0 speed 100 duplex full autoneg on
DM-37x# ethtool eth0
```

Verify Speed is 100Mb/s and Duplex is Full.

```
DM-37x# ethtool -s eth0 speed 10 duplex full autoneg off
DM-37x# ethtool eth0
```

Verify Speed is 10Mb/s and Duplex is Full.

```
DM-37x# ethtool -s eth0 speed 10 duplex half autoneg off
DM-37x# ethtool eth0
```

Verify Speed is 10Mb/s and Duplex is Half.

```
DM-37x# ethtool -s eth0 speed 100 duplex half autoneg on
DM-37x# ethtool eth0
```

Verify Speed is 100Mb/s and Duplex is Half.

```
DM-37x# ethtool -s eth0 speed 100 duplex full autoneg on
```

POR

03.03.04 Cross-over Private network

[ETHER-01-009] [ETHER-01-010] [ETHER-01-016]

Boot the device and login

Connect PC and DUT with a cross-over cable.

Set IP of workstation:

```
bash$ sudo ifconfig eth0 172.20.1.1 netmask 255.0.0.0 up
```

Set IP of DM37X:

```
DM-37x# ifconfig eth0 172.20.1.2 netmask 255.0.0.0 up
```

Ping the workstation:

```
DM-37x# ping 172.20.1.1
```


Verify ping is successful.

Ping the device:

```
bash$ ping 172.20.1.2
```

Verify ping is successful.

Disconnect the crossover cable, and reconnect the regular network Ethernet cable.

POR

03.03.05 Cable events

[ETHER-01-017] [ETHER-01-018]

Make sure /etc/network/interfaces has a valid configuration for eth0 (such as dhcp)

Boot the device and login

```
DM-37x# ifconfig
```

Ensure Ethernet is running

```
DM-37x# ifup eth0
```

Unplug the network cable.

Verify "`eth0: link down`" message is displayed.

Plug the cable back in.

Verify "`eth0: link up, <speed>, <duplex>`" message is displayed.

POR

03.03.06 Auto-negotiate to fastest compatible speed

[ETHER-01-009] [ETHER-01-010] [ETHER-11-020]

Set up a 1 GB switch. **//Hook into the network, doesn't matter what the connection speed is**

Boot the device and login

Ensure eth0 is running

```
DM-37x# ifconfig
DM-37x# ifup eth0
DM-37x# ethtool eth0
```

Verify Speed is 100Mb/s

Ping the SUT from the workstation

```
DM-37x# ping <Workstation IP address>
```

Verify successful ping

POR

03.04 Wired Ethernet driver functionality after resuming from suspend

03.04.01 Verify valid IP address is obtained from DHCP after resuming from suspend.

[ETHER-01-009] [ETHER-01-010] [ETHER-01-021]

Enter Session

In U-Boot, set kernel parameters:

```
OMAP Logic #setenv otherbootargs ${otherbootargs} ip=::::::dhcp (note:  there are 6 colons)
OMAP Logic # boot
```

Login when boot is complete

```
DM-37x# ifconfig
```

Verify an ip address was obtained

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was successful

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Get the ip address of the SOM:

```
DM-37x# ifconfig
```

Verify an ip address was obtained

Ping a known host:

```
DM-37x# ping <known host>
```

Verify ping was successful

POR

5.4 Test 04 USB Host

Purpose

This test verifies that the functional requirements for USB host are met.

Setup

As preparation for this test, the following information is critical:

- Load any standard image.
- On the baseboard, ensure J42 does not have a jumper.

Test Steps

Perform the following subtests and record results in the action log:

04.01 USB Sub-System

04.01.01 USBFS

[USBH-01-001]

Do a mount command:

```
DM-37x# mount | grep usbfs
```

Verify that usbfs is mounted on /proc/bus/usb.

04.01.02 HID – Keyboard / Mouse

[USBH-01-002] [USBH-01-006] [USBH-01-007] [USBH-01-008] [USBH-01-017]

Plug a USB keyboard into the USB port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify by checking for an 'I:' line with `Driver=usbhid.`

Verify that the matching 'E:' line has the interrupt attribute. (`Atr=03(Int.)`*)*

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Hexdump eventX to see input from USB device:

```
DM-37x# hexdump /dev/eventX (or hexdump /dev/input/eventX where X is the device found for the new device.  X is usually 2.)
```

Verify that pressing keys on the keyboard results in data on the device

Press ctrl-c

Ensure the hexdump command closes properly

Repeat the previous 4 steps three times.

Unplug the device.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify the device has been removed.

Read out dmesg:

```
DM-37x# dmesg
```

Verify the device has been removed.

Plug a USB Mouse into the USB port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify by checking for an 'I:' line with `Driver=usbhid`.

Verify that the matching 'E:' line has the interrupt attribute. (`Atr=03(Int.)`*)*

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Hexdump eventX to see input from USB device:

```
DM-37x# hexdump /dev/eventX (or hexdump /dev/input/eventX where X is the device found for the new device.  X is usually 2.)
```

Verify that moving the mouse results in data on the device

Press ctrl-c

Ensure the hexdump command closes properly

Repeat the previous 4 steps three times.

Unplug the device.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify the device has been removed.

Read out dmesg:

```
DM-37x# dmesg
```

Verify the device has been removed.

04.01.03 USB Mass Storage

[USBH-01-003] [USBH-01-006] [USBH-01-007] [USBH-01-008] [USBH-01-017] [USBH-14-24] [USBH-14-25] [USBH-14-26] [USBH-14-27]

Make sure the SD card has a text file called README.TXT

**NOTE*: You may use all four devices in a hub at once to save time, but if it fails you must do them individually."*

For each USB device in (SanDisk 256MB Cruzer USB stick, SanDisk 16 GB Cruzer USB stick, 500GB Seagate external hard drive, Western Digital WD Elements 1.5 TB External Hard Drive):

Plug the USB device into the USB Host port on the baseboard.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the matching 'E:' line has the interrupt attribute. (`Atr=02(Bulk)`)

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Do a mount command:

```
DM-37x# mount | grep sd
```

Note the name of the USB device. It should be sdx1 where x is a, b, etc.

Copy the README file to the USB device:

```
DM-37x# cat /mnt/mmcblk0p1/README.TXT
DM-37x# cp /mnt/mmcblk0p1/README.TXT /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is listed

Read the contents of the copied file:

```
DM-37x# cat /mnt/sda1/README-2.TXT
```

Verify the content of README-2.TXT is the same as README.TXT

Remove the copied file:

```
DM-37x# rm /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is NOT listed

Unplug the device.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify the device has been removed.

Read out dmesg:

```
DM-37x# dmesg
```

Verify the device has been removed

End for each USB device

04.01.04 USB On Boot

[USBH-01-006] [USBH-01-007] [USBH-01-008] [USBH-01-009] [USBH-01-017] [USBH-14-24] [USBH-14-25] [USBH-14-26] [USBH-14-27]

Make sure the SD card has a text file called README.TXT

**NOTE*: You may use all four devices in a hub at once to save time, but if it fails you must do them individually."*

For each USB device in (SanDisk 256MB Cruzer USB stick, SanDisk 16 GB Cruzer USB stick, 500GB Seagate external hard drive, Western Digital WD Elements 1.5 TB External Hard Drive)

Plug the USB device into the USB Host port on the baseboard.

Power on device

View USB devices:

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the matching 'E:' line has the interrupt attribute. (`Atr=02(Bulk)`)

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Do a mount command:

```
DM-37x# mount | grep sd
```

Note the name of the USB device. It should be sdx1 where x is a, b, etc.

Copy the README file to the USB device:

```
DM-37x# cat /mnt/mmcblk0p1/README.TXT
DM-37x# cp /mnt/mmcblk0p1/README.TXT /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is listed

Read the contents of the copied file

```
DM-37x# cat /mnt/sda1/README-2.TXT
```

Verify the content of README-2.TXT is the same as README.TXT

Remove the copied file from the USB device:

```
DM-37x# rm /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is NOT listed

Unplug the device.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify the device has been removed.

Read out dmesg:

```
DM-37x# dmesg
```

Verify the device has been removed

End for each USB device

04.01.05 USB Audio (tested with Logitech C210 USB webcam)

[USBH-01-004]

**Note: the kernel does not have a USB Audio Class – LPD does not support. Note however, that USB Audio behaves the same as USB Camera and therefore, it depends on the device.*

Plug a USB audio device into the USB Host port on the baseboard.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the matching 'E:' line has the isochronous attribute set. (`Atr=(Isoc)`)

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines say that a new audio device has been registered. As R4-01-004 is "should", it's not a requirement for release. Use an audio/linux driver that works together to test.

04.01.06 USB Web Camera (tested with Logitech C210 USB webcam)

[USBH-01-005]

Note: Use the DSP + Camera build from test 18.03.01 for this test.
Note: Use a UVC Web camera for this test. The USB Webcam may indicate audio functionality if functional.

Plug a USB web camera device into the USB Host port on the baseboard.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify by checking for an 'I:' line with `Driver=uvcvideo`

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about an UVCVideo device.

04.02 USB Spec

04.02.01 USB 1.1

[USBH-01-010]

Plug a USB 1.1 device into the device Host port.


```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device is listed and that the 'D:' line for the device shows Ver= 1.10.

04.02.02 USB 2.0

[USBH-01-011]

Plug a USB 2.0 device into the device Host port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device is listed and that the 'D:' line for the device shows Ver= 2.00.

04.02.03 Low Speed

[USBH-01-012]

Plug a low-speed USB device into the device Host port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device's 'T:' line shows Spd=1.5.

04.02.04 Full Speed

[USBH-01-013]

Plug a full-speed USB device into the device Host port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device's 'T:' line shows Spd=12.

04.02.05 High Speed

[USBH-01-014]

Plug a high-speed USB device into the device Host port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device's 'T:' line shows Spd=480.

04.02.06 Hubs

[USBH-01-015] [USBH-01-016]

NOTE: For USB OTG Host tests, connections will depend upon device availability.

Plug an unpowered hub into the device Host port.

Plug a USB Storage device into the hub.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that both the hub and the storage device are listed.

Unplug the hub from the device Host port.

Plug a powered hub into the device Host port.

Plug a USB Storage device into the hub.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that both the hub and the storage device are listed.

Unplug the hub from the device Host port.

04.02.07 Over-Current Conditions (Applicable to the DM3730/AM3703 SOM-LV only – N/A for the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 Torpedo + Wireless SOM)

[USBH-01-018]

USB Host – Attach a SDK2 Header board

Boot the device.

Load the USB host standard kernel image and login.

Plug any USB storage device into the USB port.

Find a file on the USB storage device to view

```
DM-37x#      ls /mnt/sda1
```

View a file on the USB storage device.

```
DM-37x#      cat /mnt/sda1/<filename>
```

For USB2: Use a jumper to short pin 79 and 73 (USB2_nOC ====> row J15).

For USB4: Use a jumper to short pin 79 and 61 (USB4_nOC ====> row J15).

For USB5: Use a jumper to short pin 79 and 59 (USB5_nOC ====> row J15).

Verify "over-current change on port x" interrupt message is received on the console.

Try to view a different file while the over-current condition is in effect.

```
DM-37x# cat/mnt/sda1/<different filename>
```

Verify that the file does not display

Do a dmesg command:

```
DM-37x# dmesg
```

*Verify that you see a message similar to the following indicating that the USB device was disconnected by the Over Current condition: (**usb 1-2.1: USB disconnect, device number 3)***

Repeat the test for all 3 USB host ports on the SomLV

04.03 USB Host driver functionality after resuming from suspend

04.03.01 Verify USB mass storage device works after resuming from suspend.

[USBH-01-020]

Plug a USB mass storage device into the board

Boot the board and login.

Find a file on the USB storage device to view

```
DM-37x#      ls /mnt/sda1
```

View a file on the USB storage device.

Read out a file on the USB device:

```
DM-37x#      cat /mnt/sda1/<filename>
```

Put the device into suspend mode.

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Try to view a different file.

```
DM-37x# cat/mnt/sda1/<different filename>
```

Verify that the file is viewable.

04.04 USB Host driver functionality after resuming from suspend

04.04.01 Verify lsusb works

[POWER-12-006]

Boot the board and login.

Run lsusb command

```
DM-37x#      lsusb -v
```

View the output. This should report over 250 lines of information.

```
Minimum devices seen:
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```


04.05 Confirm separate MAC addresses assigned to LAN Ethernet and USB Ethernet Dongle

< Test Removed 09/28/2015 >

04.06 USB Host auto suspend / resume

04.06.01 Verify that the USB Host port is automatically 'suspended' when a USB device is removed and 'resumed' when a USB device is plugged it.

[USBH-08-021] [USBH-08-022]

POR and login.

Using a Standard A to Mini B USB cable (or adapter), connect the baseboard to your computer that has Beacon EmbeddedWorks's Wattson program on it.

On Wattson, wait for a bit for the levels to steady, then take note of the Kit: Main Battery value. This must be done with no USB Host port devices plugged in.

Read out the stats of the USB Host ports:

```
DM-37x# cat /sys/bus/usb/devices/usb1/power/runtime_status
```

Verify that the runtime_status is "suspended"

Plug in a USB device, preferably one that would not draw extra power (no lights, laser on mouse, etc…. For testing the Transcend USB to MMC card Reader was used).

Read out the stats of the USB Host ports:

```
DM-37x# cat /sys/bus/usb/devices/usb1/power/runtime_status
```

Verify that the runtime_status is "active"

On Wattson, take note of the Kit: Main Battery value.

Verify that the Kit: Main Battery value has increased now that there is a USB device plugged into a USB Host slot.

Unplug the USB device.

Read out the stats of the USB Host ports:

```
DM-37x# cat /sys/bus/usb/devices/usb1/power/runtime_status
```

Verify that the runtime_status is "suspended"

On Wattson, take note of the Kit: Main Battery value.

Verify that the Kit: Main Battery value has decreased back to the previous "suspended" value (roughly). It may take a few seconds to get there.

5.5 Test 05 Serial Tests

Purpose

This test verifies that the functional requirements for serial UART support.

Setup

- Load any standard image.
- Connect DB-9 adapter to J25 for UARTB access (J45 for SOM-LV)
- Connect DB-9 adapter to J27 for UARTC access (J46 for SOM-LV)

Test Steps

Perform the following subtests and record results in the action log:

05.01 Linux Serial Port

05.01.01 Linux Serial Port Support

[UART-01-002] [UART-01-003]

Boot the device with any standard image.

Verify that you can login.

05.01.02 Changing baud at boot

[UART-01-002] [UART-01-003] [UART-01-004] [SW-08-002]

Boot the device with any standard image and break into U-Boot.

In U-Boot, set the baudrate kernel parameter and boot:

```
OMAP Logic # setenv baudrate 38400
```

Change the baud rate for the workstation terminal program to 38400

```
OMAP Logic # boot
```

Verify login was successful.

05.01.03 ttyO[0-2]

[UART-01-001]

Return the TTY information for various ports:

DM-37x# **stty -F /dev/ttyO0**, *verify that it returns a non-error.*

DM-37x# **stty -F /dev/ttyO1**, *verify that it returns a non-error.*

DM-37x# **stty -F /dev/ttyO2**, *verify that it returns a non-error.*

05.02 RS-232 Support Tests

05.02.01 Baud rates

[UART-01-005] [UART-01-006]

Change the baud rate for the workstation terminal program to 2400 (also change on Teraterm):

DM-37x# **stty 2400**

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the baud rate for the workstation terminal program to 9600 (also change on Teraterm):

DM-37x# **stty 9600**

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the baud rate for the workstation terminal program to 19200 (also change on Teraterm):

DM-37x# **stty 19200**

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the baud rate for the workstation terminal program to 38400 (also change on Teraterm):

DM-37x# **stty 38400**

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the baud rate for the workstation terminal program to 57600 (also change on Teraterm):

DM-37x# **stty 57600**

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the baud rate for the workstation terminal program to 115200 (also change on Teraterm):

DM-37x# **stty 115200**

Verify terminal is working correctly. (Type some commands, output should be readable.)

05.02.02 Data character size 7,8 bits

[UART-01-007]

Change the data for the workstation terminal program to 7 bits:


```
DM-37x# stty cs7
```

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the data for the workstation terminal program to 8 bits:

```
DM-37x# stty cs8
```

Verify terminal is working correctly. (Type some commands, output should be readable.)

05.02.03 Parity

[UART-01-008]

Change the data for the workstation terminal program to odd parity:

```
DM-37x# stty parenb                //enable parity
DM-37x# stty parodd
```

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the data for the workstation terminal program to even parity:

```
DM-37x# stty -parodd
```

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the data for the workstation terminal program to no parity.

```
DM-37x# stty -parenb        //no parity
```

Verify terminal is working correctly. (Type some commands, output should be readable.)

05.02.04 Stop bits

[UART-01-009]

Change the data for the workstation terminal program to two stop bits.

```
DM-37x# stty cstopb                //two stop bits
```

Verify terminal is working correctly. (Type some commands, output should be readable.)

Change the data for the workstation terminal program to one stop bit.

```
DM-37x# stty -cstopb               //one stop bit
```

Verify terminal is working correctly. (Type some commands, output should be readable.)

05.02.05 RTS/CTS Handshaking

<Test removed. Redundant with test 05.03.02.>

05.03 DM3730/AM3703 Torpedo SOM / DM3730/AM3703 Torpedo + Wireless SOM – UART Support

05.03.01 DM3730/AM3703 Torpedo SOM and DM3730/AM3703 Torpedo + Wireless SOM - UART Support

[UART-01-001]

Need a db9 female to UART male connector

For DM3730/AM3703 Torpedo SOM and DM3730/AM3703 Torpedo + Wireless SOM Only:

Connect the UART male to a standard serial female to female connector.

Connect the db9 to J25==UARTB, J27==UARTC ensuring the red line of the ribbon cable is on the Pin1/2 side of the Torpedo pins.

Open a terminal session on the PC to the appropriate serial port using

115200, 8-bit data, no parity, 1 stop bit.

Connect the DB9 connector to J25==UARTB

Boot image and login

```
DM-37x# stty 115200 < /dev/ttyO2
DM-37x# echo "Testing /dev/ttyO2" > /dev/ttyO2
```

Verify "Testing /dev/ttyO2" is output to the terminal console

Connect the DB9 connector to J27==UARTC

NOTE: UARTC (J27) is available only if pin 28 and pin 30 of J31 are not jumper together. UARTC is NOT available when these 2 pins are jumped for the Wireless feature in the Torpedo + Wireless SOM.

```
DM-37x# stty 115200 < /dev/ttyO1
DM-37x# echo "Testing /dev/ttyO1" > /dev/ttyO1
```

Verify "Testing /dev/ttyO1" is output to the terminal console

POR

05.03.02 Hardware flow control

[UART-01-001] [UART-01-010]

**NOTE: This test works well with plugging two SOMs to each other but can also be run attached to a PC.*

On both SOMs:

Turn on hardware flow control:

```
DM-37x# stty -F /dev/ttyO2 crtscts -cooked -echo 115200
```

Plug in a null modem cable (serial) between the dev kits on the standard serial ports (DB-9)

On the receive board:

On the receive board, create a file to compare to:

```
DM-37x# seq 20000 > /dev/shm/hfc_file2
```

Start the data transfer with automatic hardware control start and stops:

```
DM-37x# cat /dev/ttyO2 > /dev/shm/hfc_file &
while true;do killall -CONT cat;sleep 1;killall -STOP cat; ls -la /dev/shm; sleep 5;done
```

On the transmit board:

```
DM-37x# seq 20000 > /dev/ttyO2
```

When the transfer is complete:

Hit CTRL-C to stop the ls -la command output

Do a cmp (file compare) on the 2 files:

```
DM-37x# cmp /dev/shm/hfc_file /dev/shm/hfc_file2
```

Verify that the files are the same (cmp returns nothing)

05.04 Serial driver functionality after resuming from suspend

05.04.01 Verify change of baud rate after resuming from suspend.

[UART-01-013]

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

```
DM-37x# stty 2400
```

Change the baud rate for the workstation terminal program to 2400.

Verify terminal is working correctly. (Type some commands, output should be readable.)

POR

Change the baud rate for the workstation terminal program back to 115200.

05.05 Debug Serial output through UARTB/UARTC.

**NOTE:*

Create two bootable SD Card using custom boot image for UARTB and UARTC:

To choose the console, run "./ltib -c" on the build machine, then scroll down to "Options" after "Target System Configuration", hit enter, then you can select the console prefix - note that OMAP linux-2.6.32 uses "ttyS" as the prefix and linux-3.0 (and newer) uses "ttyO" as the prefix. You can also choose between UARTC and UARTB (ttyO1 and ttyO2)

05.05.01 Verify debug serial output from UARTB / UARTC.

[UART-16-017]

Connect UARTB (or C) using a serial adapter (setup for section 5 above)

Boot system with custom boot image built for serial debug to UARTB / UARTC.

Verify that the output for U-boot is flowing through the UARTB / UARTC port, not the standard UARTA (serial connection). This should be the same output you would usually see when booting normally.

In U-boot, set up the preboot variable:

```
DM-37x# env default -f; setenv preboot; saveenv; boot
```

Let the device finish booting and login.

Verify that the remainder of the boot output also flows through the UARTB / UARTC port and that you were able to login.

Repeat this test with both the UARTB and UARTC builds.

Make sure to go back to the standard build and U-boot environment after this test (env default -f in U-boot)

05.06 UART Loopback Mode

05.06.01 Loopback mode test

[UART-01-011]

Make sure that the program “loopback_test” (from testing files folder in SVN) is on the SD car and copy it to /dev/shm:

```
DM-37x# cp /mnt/mmcblk0p1/loopback_test /dev/shm/
```

Give permission to execute the test:

```
DM-37x# chmod 777 /dev/shm/loopback_test
```

Move to /dev/shm:

```
DM-37x# cd /dev/shm
```

Set the device to loopback to /dev/tty02:

```
DM-37x# stty -echo -cooked -F /dev/ttyO2
DM-37x # cat /dev/ttyO2 &
```

Run the Loopback test on /dev/tty02:

```
DM-37x # ./loopback_test 1 /dev/ttyO2
DM-37x # echo “Does this echo” > /dev/ttyO2
```

Verify that “Does this echo” is “looped back” (argument 1 is enable)

Run the Loopback test on /dev/tty02:

```
DM-37x # ./loopback_test 0 /dev/ttyO2
DM-37x # echo "Does this echo" > /dev/ttyO2
```

Verify that nothing is "looped back" (argument of 0 is disable)

5.6 Test 06 SSMTP (email)

Purpose

This test verifies that the SOM is able to send emails

Setup

As preparation for this test, execute the following:

- Load any standard image.

Test Steps

Perform the following subtests and record results in the action log:

06.01 Sending an email

06.01.01 Send email

<Test Removed>

5.7 Test 07 MMC/SD Tests

Purpose

This test verifies that the functional requirements for MMC/SD cards are met.

Setup

As preparation for this test, execute the following:

- Create a FAT file system on the MMC/SD cards if there is not one already present.
- Load any standard image.

Test Steps

Perform the following subtests and record results in the action log:

07.01 Linux System Support

07.01.01 Detect Insert

[SDMMC-01-008]

Boot the device.

```
DM-37x# mount | grep /dev/mmcblk0p1
```

Ensure /mnt/mmcblk0p1 is not mounted (umount if necessary)

```
DM-37x# ls /dev
```

Verify that device dev/mmcblk0 and /dev/mmcblk0p1 do not exist

Insert a SD card

Verify the card was detected (a message similar to below is displayed following insertion)
mmc0: new SD card at address 80ca

mmcblk0: mmc0:80ca SD256 241 MiB

mmcblk0: p1

View the /dev folder:

```
DM-37x# ls /dev
```

Verify that device dev/mmcblk0 and /dev/mmcblk0p1 (if card has a primary partition) exist

Read out dmesg:

```
DM-37x# dmesg
```

Review the output for a displayed message similar to the detection message following insertion.

Verify the message is displayed.

POR

07.01.03 SD card as filesystem – before boot

[SDMMC-01-007] [SDMMC-01-009]

Power off device.

Insert an MC/SD card.

Boot the device.

```
DM-37x# dmesg
```

**NOTE: If the message about the SD card mounting is not at the bottom of the dmesg list, try using: dmesg | grep SD*

Verify the SD card was detected (message is displayed)

View the /dev folder:

```
DM-37x# ls /dev
```

Verify that the devices /dev/mmcblk0 and /dev/mmcblk0p1 exists.

View mounted devices:

```
DM-37x# mount | grep /dev/mmcblk0p1
```

Verify the SD card is mounted (/dev/mmcblk0p1 on /mnt/mmcblk0p1 type vfat... is displayed)

View mounted devices:

```
DM-37x# df
```

Verify the SD card is mounted (/dev/mmcblk0p1 on /mnt/mmcblk0p1 is displayed)

Unmount the SD card

```
DM-37x# umount /mnt/mmcblk0p1
```

*NOTE: the umount command does *not* contain an *n*...*

View mounted devices:

```
DM-37x# mount | grep /dev/mmcblk0p1
```

Verify the SD card is not mounted (/dev/mmcblk0p1 on /mnt/mmcblk0p1 is not displayed). The mount | grep command should return nothing.

Remove the MMC/SD card.

```
DM-37x# dmesg
```

Verify the removal was detected. Also verify that the device /dev/mmcblk0 and /dev/mmcblk0p1 was removed

POR

07.01.04 Hardware write protect

[SDMMC-01-002] [SDMMC-01-003] [SDMMC-01-008] [SDMMC-01-013]

**NOTE: The DM3730/AM3703 Torpedo SOM and DM3730/AM3703 Torpedo + Wireless SOM do not support hardware write protection. This test is N/A for the DM3730/AM3703 Torpedo and DM3730/AM3703 Torpedo + Wireless SOMs.*

Flip write protect switch on MMC/SD card.

Boot the device.

Insert the card

```
DM-37x# mount | grep /dev/mmcblk0p1
```

*Verify the SD card is mounted read-only (/dev/mmcblk0p1 on /mnt/mmcblk0p1 type vfat (**ro**,…))*

POR

07.01.05 SD card as filesystem – after boot

[SDMMC-01-002] [SDMMC-01-003] [SDMMC-01-004] [SDMMC-01-005] [SDMMC-01-006] [SDMMC-01-008] [SDMMC-01-011] [SDMMC-01-016]

Boot the device.

Insert an SD card (use a selection of SD cards that includes at least one card > 4GB and 1 card < 2 GB)

Verify the SD card was detected (message is displayed)

View mounted devices:

```
DM-37x# mount | grep /dev/mmcblk0p1
```

Verify the SD card is mounted (/dev/mmcblk0p1 on /mnt/mmcblk0p1 type vfat... is displayed)

Create some random test data on the device:

```
DM-37x# dd if=/dev/urandom of=/testdata bs=1K count=16K
```

Copy the /testdata file to the SD card:

```
DM-37x# cp /testdata /mnt/mmcblk0p1/testdata
```

Ensure the file copied to the SD card is the same that was created (no message returned following command):

```
DM-37x# cmp /testdata /mnt/mmcblk0p1/testdata
DM-37x#  umount /mnt/mmcblk0p1
```

**NOTE: the umount command does *not* contain an *n*...*

View mounted devices:

```
DM-37x#  mount | grep /dev/mmcblk0p1
```

Verify the SD card is not mounted (/dev/mmcblk0p1 on /mnt/mmcblk0p1 type vfat... is not displayed). The mount | grep command should return nothing.

POR

07.01.06 SDIO Wireless

<Test removed. (11/5/2012)>

07.01.07 Ready times

[SDMMC-01-017]

Boot the device and login

While watching a timer or stopwatch, quickly insert a MMC/SD card and then press **Enter** to run the following command.

```
DM-37x# ls /mnt/mmcblk0p1
```

Verify the command returns without error at 2 seconds of inserting the MMC/SD card.

POR

07.01.08 Reinsert time

[SDMMC-01-018]

Boot the device and login

Insert a MMC/SD card.

Eject the card and reinsert it immediately.

Verify the card is recognized as inserted within 6 seconds of ejecting the card. Look for a message similar to '`mmc0: new SD card at address 0001`' printed on the console.

POR

07.01.09 Uboot.bin location on SD cards

<Test Removed (11/05/2012)>

07.02 MMC/SD driver functionality after resuming from suspend

07.02.01 Detect insertion of card after resuming from suspend.

[SDMMC-01-020]

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

View mounted devices:

```
DM-37x# mount | grep /dev/mmcblk0p1
```

Verify /mnt/mmcblk0p1 is not mounted (umount if necessary)

View the /dev folder:

```
DM-37x# ls /dev
```

Verify that device dev/mmcblk0 and /dev/mmcblk0p1 do not exist Insert an SD card.

Verify the card was detected (a message similar to below is displayed following insertion)

```
mmc0: new SD card at address 80ca
mmcblk0: mmc0:80ca SD256 241 MiB
mmcblk0: p1
```

View the /dev folder:

```
DM-37x# ls /dev
```

Verify that device dev/mmcblk0 and /dev/mmcblk0p1 (if card has a primary partition) exist

Read out dmesg:

```
DM-37x# dmesg
```


Review the output for a displayed message similar to the detection message following insertion.

Verify the message is displayed.

Create a file on the SD card using the 'echo' command:

```
DM-37x# echo "Echo test file" > /mnt/mmcblk0p1/echo_test_file.txt
```

Read the contents of echo_test_file.txt:

```
DM-37x# cat /mnt/mmcblk0p1/echo_test_file.txt
```

Verify that the contents are correct

Remove the created file:

```
DM-37x# rm /mnt/mmcblk0p1/echo_test_file.txt
```

POR

07.03 The MMC/SD device driver uses DMA for data transfers

07.03.01 The MMC/SD device driver uses DMA for data transfers

[SDMMC-01-015]

Move to the SD card:

```
DM-37x# cd /mnt/mmcblk0p1
```

Cat the interrupts looking for DMA transfers:

```
DM-37x# cat /proc/interrupts | grep DMA
  12:  197  INTC  DMA  (note the number of DMA interrupts, in this case 197)
```

Dump the contents of a large file out through /dev/null

```
DM-37x# dd if=<largest file on card> of=/dev/null
```

Cat the interrupts looking for DMA transfers again:

```
DM-37x# cat /proc/interrupts | grep DMA
  12:  409   INTC  DMA (note the number of DMA interrupts)
```

Verify that dumping the data from the card to /dev/null made the number of DMA interrupts increase

5.8 Test 08 Touch Screen Tests

Purpose

This test verifies that the functional requirements for Touch Screen for supported Beacon EmbeddedWorks displays are met.

Setup

As preparation for this test, perform the following:

- Load any standard image
- After booting and logging in, execute the following commands:
  - export TSLIB_TSDEVICE=/dev/input/eventX (X is the device found for the new device. X is usually 0.)
  - export TSLIB_CALIBFILE=/etc/pointercal
  - export TSLIB_CONSOLEDEVICE=none
  - export TSLIB_FBDEVICE=/dev/fb0

Test Steps

Perform the following subtests and record results in the action log:

Linux Input System Interface

08.01.01 Basic Test

[TOUCH-01-001] [TOUCH-01-002] [TOUCH-01-007]

For each from [LCD-4.3-WQVGA-20R (28), LCD-4.3-WQVGA-10R (15)] except as noted:

Run the ts_test application:

```
DM-37x# ts_test
```

Verify that touches respond by the crosshair moving to where you touched.

Hit Ctl-C to exit and return to prompt

End For each from [LCD-4.3-WQVGA-20R (28), LCD-4.3-WQVGA-10R (15)] except as noted:

08.01.02 Calibrate at run time

[TOUCH-01-002] [TOUCH-01-006]

For each from [LCD-4.3-WQVGA-20R (28), LCD-4.3-WQVGA-10R (15)] except as noted:

```
DM-37x# ts_calibrate
```

Follow the instructions on the LCD to calibrate.

Verify touchscreen still works by navigating around a bit, and that the calibration is correct.

End For each from [LCD-4.3-WQVGA-20R (28), LCD-4.3-WQVGA-10R (15)] except as noted:

08.01.03 Touch with mouse or keyboard.

[TOUCH-01-001] [TOUCH-01-003]

Run the ts_test application:

```
DM-37x# ts_test
```

Verify that touches respond by the crosshair moving to where you touched.

Plug in a keyboard and mouse.

Verify touchscreen works by navigating around a bit.

Unplug mouse, but leave keyboard plugged in.

Verify touchscreen works by navigating around a bit.

Unplug keyboard, plug in mouse.

Verify touchscreen works by navigating around a bit.

08.01.04 Raw data test

<Test removed. (11/05/2012)>

08.01.05 Display configuration data

[TOUCH-01-004]

View the pointercals:

```
DM-37x# ls /etc/pointercal*
```

Verify that pointercal-28 and pointercal-15 are listed.

08.01.06 Touch Screen interrupt

[TOUCH-01-008]

View interrupts:

```
DM-37x# cat /proc/interrupts
```

Verify interrupt #313: tsc2004 is listed

08.01.07 Sample rate

[TOUCH-01-009]

Run the ts_print application:

```
DM-37x# ts_print
```

Touch and drag a finger across the screen to view sample intervals and timestamps.

Verify samples are returned at a level of at least 80 samples/second.

Example of output and determination of samples/second:

946728473.751495: 274 177 234
946728473.760284: 275 182 234


```
946728473.769043:  275  187  234
946728473.777771:  276  190  234
946728473.786468:  275  191  234
946728473.795257:  273  190  234
946728473.803955:  266  188  234
946728473.812835:  257  184  234
946728473.821472:  246  176  234
946728473.830261:  235  168  234
946728473.847656:  223  152  234
```

Each line is a touch sample, and the number on the left is time, so in this case, 11 samples showed up in 946728473.847656 - 946728473.751495 seconds, or 11 samples in 0.096161 seconds, or 0.008741 seconds per sample, or 114.39 samples per second.

08.02 Touch screen driver functionality after resuming from suspend

08.02.01 Detect touch screen position after resuming from suspend.

[TOUCH-01-011]

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Run the ts_test application:

```
DM-37x# ts_test
```

Verify that touches respond by the crosshair moving to where you touched.

POR

08.02.02 Suspend after write to cards and exercise touch screen

<Test Removed 11/05/2012.>

5.9 Test 09 SPI Tests

Purpose

This test verifies that the functional requirements for the SPI interface are met.

Setup

As preparation for this test, perform the following:

- This test is applicable to the DM3730/AM3703 SOM-LV only.
- Aardvark I2C/SPI Activity board setup
- Expansion board setup

- Connect the Aardvark board to the expansion board
- Connect the Expansion board to the DUT

Test Steps

Perform the following subtests and record results in the action log:

09.01 SPI Interface

01.01.04 -

09.01.01 SPI Test

** NOTE: This test is valid on SomLV devices only.*

[SPI-01-001] [SPI-01-002] [SPI-01-003] [SPI-01-004] [SPI-01-005] [SPI-01-006] [SPI-01-007] [SPI-01-008]

Connect the Aardvark I2C/SPI Activity board to the expansion board as follows:
Activity board <--> Expansion board
+5V (J5.) <--> VIO_1V8 (J18.68)
GND (J5.) <--> GND (J21.1-6)
SCK (J5) <--> SPI_SCLK (J4.1)
MOSI (J5) <--> SPI TX (J4.3)
MISO (J5) <--> SPI RX (J4.5)

For chip select lines, choose:
SS (J5) <-> uP_SPI_CS0 (J4.7)
or
SS (J5) <-> uP_SPI_CS1 (J18.3)

Connect the expansion board with Aardvark to the baseboard with SOMLV.

Boot the image and login

For "/dev/spidev3.0", or use uP_SPI_CS0:

Ensure the Aardvark to Expansion board is connected as follows:
SS (J5) <-> uP_SPI_CS0 (J4.7)

Run the spi-test:

```
DM-37X# spi-test
```

Verify message is displayed "SPI EEPROM Read/Write test successful!" is displayed

Run the spi-test:

```
DM-37X# spi-test -1
```

Verify message is displayed "SPI EEPROM Read/Write failed." is displayed
End For "/dev/spidev3.0", or use uP_SPI_CS0:

If spidev 3.0 setup passes, then the test is complete. If spidev3.0 setup fails, try spidev 3.1 test (make sure to change the expansion board connection for the Spidev 3.1 connection.

For "/dev/spidev3.1", or use uP_SPI_CS1:

Ensure the Aardvark to Expansion board is connected as follows:
SS (J5) <-> uP_SPI_CS1 (J18.3)

Run the spi-test:

```
DM-37X# spi-test -1
```

Verify message is displayed "SPI EEPROM Read/Write test successful!" is displayed

Run the spi-test:

```
DM-37X# spi-test
```

Verify message is displayed "SPI EEPROM Read/Write failed." is displayed

End For "/dev/spidev3.1", or use uP_SPI_CS1:
POR

09.02 SPI driver functionality after resuming from suspend

09.02.01 Verify SPI test functionality after resuming from suspend.

** NOTE: This test is valid on SomLV devices only.*

[SPI-01-001] [SPI-01-002] [SPI-01-003] [SPI-01-004] [SPI-01-005] [SPI-01-006] [SPI-01-007] [SPI-01-008] [SPI-01-010]

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Perform test 09.01.01 after the suspend / resume.

09.03 SPI Interface to on board SPI Flash - AT25160BN (U33)

09.03.01 SPI Test

** NOTE: This test is valid on Torpedo and Torpedo+Wireless devices only.*

[SPI-01-001] [SPI-01-002] [SPI-01-003] [SPI-01-004] [SPI-01-005] [SPI-01-006] [SPI-01-007] [SPI-01-008]

Boot the image and login

For "/dev/spidev1.0":

DM-37X# spi-test /dev/spidev1.0

Verify message is displayed "SPI EEPROM Read/Write test successful!" is displayed

DM-37X# **spi-test /dev/spidev1.1**

Verify message is displayed "SPI EEPROM Read/Write failed." is displayed

End For "/dev/spidev1.0":

09.04 SPI driver functionality after resuming from suspend

09.04.01 SPI Interface to on board SPI Flash - AT25160BN (U33)04.01 Verify SPI test functionality after resuming from suspend.

** NOTE: This test is valid on Torpedo and Torpedo+Wireless devices only.*

[SPI-01-001] [SPI-01-002] [SPI-01-003] [SPI-01-004] [SPI-01-005] [SPI-01-006] [SPI-01-007] [SPI-01-008] [SPI-01-010]

Put device into suspend mode:

DM-37x# **echo mem > /sys/power/state**

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Perform test 09.03.01 after the suspend / resume.

5.10 Test 10 MTD File System Tests

Purpose

This test verifies that the functional requirements for the MTD interface are met.

Setup

Load standard image

Test Steps

Perform the following subtests and record results in the action log:

10.00 – NOR flash MTD support

10.00.00 Flash combinations

[FLASH-01-010]

Verify that all available NOR-flash / Nand-flash combinations are being tested (i.e. 8Mb flash with 256Mb Nand etc…)

10.00.01 Load a standard image.

[FLASH-01-010]

List partitions:

```
DM-37x# cat /proc/mtd
```

For SOMs with NOR-flash:

Verify the NOR flash is supported (mtd6: xxxxxxxx xxxxxxxx "nor-flash")

For SOMs without NOR-flash:

*Verify the NOR flash is NOT supported (mtd6: 00800000 00020000 "nor-flash" is *NOT* displayed)*

End Session

10.01 Linux File System Interface

10.01.01 NOR JFFS2 (boards with NOR flash only)

[FLASH-01-001] [FLASH-01-006] [FLASH-01-007] [FLASH-01-008] [FLASH-01-009]

Load a standard image.

```
DM-37x# flash_eraseall /dev/mtd6
DM-37x# mkdir -p /mnt/mtd-nor
DM-37x# cat /proc/mtd
     //List partitions
```

DM-37x# **mount -t jffs2 /dev/mtdblock6 /mnt/mtd-nor** {If errors occur, restart the device and, using LogicLoader, erase flash i.e., erase /dev/flash0 B0 B64.}

```
DM-37x# mount | grep jffs2
```

Verify the jffs2 partition was mounted.

Create a file with touch command and view it:

```
DM-37x# touch /mnt/mtd-nor/newfile.txt
DM-37x# ls /mnt/mtd-nor/newfile.txt
```

Verify the commands executed properly.

Remove the created file and view:

```
DM-37x# rm /mnt/mtd-nor/newfile.txt
DM-37x# ls /mnt/mtd-nor/newfile.txt
```

Verify the commands executed properly.

Umount and view:

```
DM-37x# umount /mnt/mtd-nor
DM-37x# mount | grep /mnt/mtd-nor
```

Verify the partition was unmounted (/mnt/mtd-nor is not displayed). mount | grep command should return nothing.

10.01.02 NAND JFFS2

[FLASH-01-001] [FLASH-01-003] [FLASH-01-006] [FLASH-01-007] [FLASH-01-008] [FLASH-01-009]

Load a standard kernel image and login

Mount and view a Nand JFFS2 partition:

```
DM-37x# mkdir -p /mnt/mtd-nand
DM-37x# cat /proc/mtd
```

DM-37x# **mount -t jffs2 /dev/mtdblock5 /mnt/mtd-nand** {If errors occur, restart the device and, using LogicLoader, erase NAND i.e., erase /dev/nand0 BX BY where X/Y are replaced by the blocks that span the particular partition, or in u-boot use the 'nand erase.chip' command .}.

```
DM-37x# mount | grep jffs2
```

Verify the jffs2 partition was mounted.

Create a file and view it:

```
DM-37x# touch /mnt/mtd-nand/newfile.txt
DM-37x# ls /mnt/mtd-nand/newfile.txt
```

Verify the commands executed properly.

Remove the created file and view:

```
DM-37x# rm /mnt/mtd-nand/newfile.txt
DM-37x# ls /mnt/mtd-nand/newfile.txt
```

Verify the commands executed properly.

Unmount the partition:

```
DM-37x# umount /mnt/mtd-nand
DM-37x# mount | grep jffs2
```

Verify the partition was unmounted (/mnt/mtd-nand is not displayed). mount | grep command should return nothing.

10.01.03 NAND YAFFS2

[FLASH-01-001] [FLASH-01-005] [FLASH-01-006] [FLASH-01-007] [FLASH-01-008] [FLASH-01-009]

Load the standard image

Make a yaffs2 partition and mount it:

```
DM-37x# mkdir -p /mnt/mtd-nand
DM-37x# cat /proc/mtd
     //List partitions
```

DM-37x# **mount -t yaffs2 /dev/mtdblock5 /mnt/mtd-nand** {If errors occur, restart the device and, using LogicLoader, erase NAND i.e., erase /dev/nand0 BX BY where the blocks mentioned (X for start, Y for size) span the particular NAND partition, or in u-boot use the 'nand erase.chip' command.}

```
DM-37x# mount | grep yaffs2
```

Verify the yaffs2 partition was mounted.

Create and file and view it:

```
DM-37x# touch /mnt/mtd-nand/newfile.txt
DM-37x# ls /mnt/mtd-nand/newfile.txt
```

Verify the commands executed properly.

Remove the created file and view:

```
DM-37x# rm /mnt/mtd-nand/newfile.txt
DM-37x# ls /mnt/mtd-nand/newfile.txt
```

Verify the commands executed properly.

Unmount the partition:

```
DM-37x# umount /mnt/mtd-nand
DM-37x# mount | grep yaffs2
```

Verify the partition was unmounted (/mnt/mtd-nand is not displayed). mount | grep command should return nothing.

10.01.04 Root on NAND Flash JFFS2

<Test Removed>

10.01.05 Root on NAND Flash YAFFS2

[FLASH-01-001] [FLASH-01-002] [FLASH-01-005] [FLASH-01-006] [FLASH-01-007] [FLASH-01-008] [FLASH-01-009]

In Uboot erase NAND:

```
OMAP Logic # nand erase.chip
```

Reboot

Load a standard image.

Copy the root file system to NAND flash {this assumes that mtd partition 3 is on NAND and is big enough for the root file system (>18M). Adjust the partition number as needed.}

```
DM-37x# cat /proc/mtd
     //List partitions
DM-37x# mount -t yaffs /dev/mtdblock5 /mnt/src
DM-37x# cp -a /bin /dev /etc /home /lib /linuxrc /opt /root /sbin /tmp /usr /var /mnt/src
```

{NOTE: expect a delay while directories are copied}

```
DM-37x# mkdir /mnt/src/{mnt,proc,sys}
DM-37x# ls /mnt/src
```

Verify that the directories were created or copied.

```
DM-37x# umount /mnt/src
```

Reboot.

Load a standard kernel image.

Break into U-Boot.

Set the kernel parameters and boot:

```
OMAP Logic # setenv error
OMAP Logic # mmc init
```

mmc1 is available

```
OMAP Logic # run _burncommon
```

reading mlo

```
OMAP Logic # setenv rootfs_device /dev/mtdblock5
OMAP Logic # setenv rootfs_type yaffs
OMAP Logic # setenv rootfs_location /dev
OMAP Logic # setenv kernel_location nand
OMAP Logic # saveenv
```

Remove the SD card before rebooting.

POR

Verify that the device comes up and login is successful.

View mounted devices:

```
DM-37x# mount
```

Verify the rootfs is of type yaffs (/dev/root on / type yaffs (rw))

Reboot and break into U-Boot

Erase the contents of nand:

```
OMAP Logic # nand erase.chip
```

Reset and save the default environmental variables:.

```
OMAP Logic # env default -f; saveenv
```

Reboot before running any other tests.

10.01.14 Burn a rootfs into NAND via U-Boot for YAFFS

[FLASH-01-006] [FLASH-01-014] [SW-08-002][SW-08-003]

NAND BOOT

POR

Boot the board and break into U-Boot

Reset the default environment

```
OMAP Logic # env default -f
```

Erase the nand on the SOM

```
OMAP Logic # nand erase.chip
```

Setup the board to boot from nand (non yaffs)

Makenandboot requires the same files as the SD boot card

```
OMAP Logic # run makenandboot
```

When the makenandboot script is finished it will say “Done”

Remove the SD boot card

POR

Verify that the board boots correctly with NAND boot and that you can login.

POR

YAFFS BOOT

Boot the board and break into U-Boot

Reset the default environment

```
OMAP Logic # env default -f
```

Erase the nand on the SOM

```
OMAP Logic # nand erase.chip
```

Setup the board to boot from nand yaffs

makeyaffsboot requires mlo, u-boot.bin.ift, uImage and rootfs.yaffs2 files

```
OMAP Logic # run makeyaffsboot
```

When the makenandboot script is finished it will say “Done”

Remove the SD boot card

POR

Verify that the board boots correctly with NAND YAFFS boot and that you can login.

POR

RESET BOARD

Boot the board and break into U-Boot

Reset the default environment

```
OMAP Logic # env default -f
```

Erase the nand on the SOM

```
OMAP Logic # nand erase.chip
```

The board is now back to normal, requiring a SD boot card to run.

**Note If the "makenandboot" and "makeyaffsboot" scripts are not available in the U-Boot environment, use the directions below. You will have to adjust the directions to fit the current way of setting up U-Boot.*

Boot the board and break into U-Boot

View the mounted partitions

```
OMAP Logic # mtdpart
```

device nand0 <omap2-nand.0>, # parts = 5

#: name	size	offset	mask_flags
0: x-loader	0x00080000	0x00000000	0
1: u-boot	0x001a0000	0x00080000	0
2: u-boot-env	0x00060000	0x00220000	0
3: kernel	0x00400000	0x00280000	0
4: fs	0x1f980000	0x00680000	0

Load/burn x-loader into the first four blocks of NAND (i.e. four copies of x-loader into the x-loader partition):

```
OMAP Logic # mmc init
OMAP Logic # fatload mmc 0 $loadaddr MLO
OMAP Logic # nand erase.part x-loader
OMAP Logic # nandecc hw
OMAP Logic # nand write $loadaddr 0x0 0x10000
OMAP Logic # nand write $loadaddr 0x20000 0x10000
OMAP Logic # nand write $loadaddr 0x40000 0x10000
OMAP Logic # nand write $loadaddr 0x60000 0x10000
```

There should be an output similar to below for each of the 4 writes:

NAND write: device 0 offset 0x60000, size 0x10000

65536 bytes written: OK

Load/burn u-boot.bin

```
OMAP Logic # fatload mmc 0 $loadaddr u-boot.bin
OMAP Logic # nand erase.part u-boot
OMAP Logic # nandecc $defaultecc
OMAP Logic # nand write $loadaddr u-boot
```

There should be an output similar to below:

NAND write: device 0 offset 0x80000, size 0x1a0000

1703936 bytes written: OK

Remove the MMC card and reboot. The board should boot into U-Boot

Verify that the boards have booted into U-Boot.

Re-insert the SD card.

Load/burn kernel from SD card into NAND (kernel partition

```
OMAP Logic # mmc init
OMAP Logic # fatload mmc 0 $loadaddr uImage
OMAP Logic # setenv nandkernelsize $filesize
OMAP Logic # nand erase.part kernel
OMAP Logic # nand write.i $loadaddr kernel $filesize
```

There should be an output similar to below:

NAND write: device 0 offset 0x280000, size 0x3c490c

3950860 bytes written: OK

Load/burn YAFFS rootfs from SD card into NAND (fs partition):

```
OMAP Logic # fatload mmc 0 $loadaddr rootfs.yaffs2
OMAP Logic # nand erase.part fs
OMAP Logic # nandecc $defaultecc
OMAP Logic # nand write.yaffs $loadaddr fs $filesize
```

There should be an output similar to below:

NAND write: device 0 offset 0x680000, size 0x281db40

442064704 bytes written: OK

Set a few environmental variables.

```
OMAP Logic # setenv bootcmd run defaultboot
OMAP Logic # setenv kernel_location nand
OMAP Logic # setenv rootfs_location /dev
OMAP Logic # setenv rootfs_type yaffs
OMAP Logic # setenv rootfs_device /dev/mtdblock5
OMAP Logic # saveenv
```

Remove the SD card.

Reboot the boards. They should do a full boot from nand and end up at the Linux login prompt.

Verify that the boards have successfully booted from nand.

10.02 MTD file system driver functionality after resuming from suspend

10.02.01 Mount a YAFFS file system in NAND after resuming from suspend.

[POWER-12-006]

Load a standard kernel image and login.

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Make and mount a yaffs partition:

```
DM-37x# mkdir -p /mnt/mtd-nand
DM-37x# cat /proc/mtd
     //List partitions
```

DM-37x# **mount -t yaffs2 /dev/mtdblock5 /mnt/mtd-nand** {If errors occur, restart the device and, using LogicLoader, erase NAND i.e., erase /dev/nand0 BX BY where X/Y are replaced by the blocks that span the particular partition.}

```
DM-37x# mount | grep yaffs2
```

Verify the yaffs partition was mounted.

Create and view a file:

```
DM-37x# touch /mnt/mtd-nand/newfile.txt
DM-37x# ls /mnt/mtd-nand/newfile.txt
```

Verify the commands executed properly.

Remove the created file and view:

```
DM-37x# rm /mnt/mtd-nand/newfile.txt
DM-37x# ls /mnt/mtd-nand/newfile.txt
```

Verify the commands executed properly.

Unmount the partition:

```
DM-37x# umount /mnt/mtd-nand
DM-37x# mount | grep yaffs2
```

Verify the partition was unmounted (/mnt/mtd-nand is not displayed). mount | grep command should return nothing.

POR

10.03 Nand Stress testing

10.03.01 Nand Stress Test – 12 hour

[FLASH-01-001]

Per the directions in: Appendix B: NAND STRESS Testing, perform a 12 hour nand stress test

Verify that the testing passed.

10.04 Reflash NAND from within Linux testing flashtool command

01.01.05 -

10.04.01 Reflash a YAFFS file system in NAND after resuming from within Linux testing flashtool command suspend.

[FLASH-01-005]

Load a standard u-boot image and erase all of NAND memory

```
OMAP Logic # nand erase.chip
OMAP Logic # boot
```

Continue to boot to there kernel

Login into at the kernel console

```
Program MLO
DM-37x# flashtool -w -e --all-blocks --hamming /dev/mtd0 /mnt/mmcblk0p1/MLO
```

**NOTE: if you get messages about bad blocks, you will need to clear them. To clear the bad blocks, reboot and then in u-boot enter the command "nand scrub.part x-loader". Entering the command "nand bad" will list all of the bad blocks.*

Program u-boot:

```
DM-37x# flash_eraseall /dev/mtd1
DM-37x# nandwrite -p /dev/mtd1 /mnt/mmcblk0p1/u-boot.bin.ift
```

Program kernel:

```
DM-37x flash_eraseall /dev/mtd3
DM-37x nandwrite --markbad --pad /dev/mtd3 /mnt/mmcblk0p1/uImage
```

Program rootfs:

```
DM-37x flash_eraseall /dev/mtd4
DM-37x nandwrite --markbad --pad /dev/mtd4 /mnt/mmcblk0p1/rootfs.ext2.gz.uboot
```

Remove the SD-Card

Reboot:

```
DM-37x# reboot
```

Load the u-boot image from NAND and configure u-boot environment variables to boot from NAND:

```
OMAP Logic # setenv kernel_location nand-part
OMAP Logic # setenv rootfs_location nand-part
```



```
Boot from NAND:
   OMAP Logic # boot
```

Verify the commands executed properly and the system boots from NAND to the Linux kernel console.

10.05 Running Linux BSP with SD Card as root file system

10.05.01 Running Linux BSP with SD Card as root file system

[SDMMC-01-010]

A script is provided to simplify the steps required to create an SD card with root filesystem.

Download the create_sdcard script from here:
http://support.beaconembedded.com/Portals/0/Users/049/05/305/create_sdcard.zip

Place the create_sdcard script into the LTIB root directory in the build machine (virtual machine)

Give yourself permission to run the script.

```
logic@logic-Virtualbox:<location of LTIB folder> chmod 777 create_sdcard.sh
```

Insert an SD card into your Linux host PC.

Run the script. :

```
logic@logic-Virtualbox:<location of LTIB folder> ./create_sdcard.sh
  Devices available:
  sdb is 1.8GB - Card Reader
  Enter device: sdb
  Setting up sdb
  Do you wish to continue? (y/N) y
  Partitioning sdb.
  [sudo] password for logic:
  mke2fs 1.42 (29-Nov-2011)
  Mounting bootloader partition
  Mounting root partition
  Flushing data to SD card
  Unmounting bootloader partition
  Unmounting root partition
```

After the script has completed, remove the SD card and insert it into your DM3730/AM3703 SOM system.

Boot the system and press any key to pause at the U-Boot prompt

Update the following U-Boot variables (break into U-Boot).

```
OMAP Logic # nand erase.chip
OMAP Logic # env default -f
OMAP Logic # setenv rootfs_location /dev
OMAP Logic # setenv rootfs_type ext3
```



```
OMAP Logic # setenv rootfs_device /dev/mmcblk0p2
OMAP Logic # setenv kernel_location mmc
OMAP Logic # saveenv
OMAP Logic # reset
```

Verify your system root filesystem is running from /dev/mmcblk0p2.

Check the disk filesystem:

```
DM-37x# df
Filesystem    1K-blocks      Used           Available              Use%           Mounted
on
/dev/root            1631756                203676 1345188                13%
      /
tmpfs                        120400          52                    120348          0%
                     /dev
/dev/mmcblk0p2       1631756        203676  1345188        13%     /mnt/mmcblk0p2
/dev/mmcblk0p1       307016  4384                  302632  1%              /mnt/mmcblk0p1
shm                                                120400  0
      120400  0%             /dev/shm
rwfs                                               512                   0
                     512                   0%             /mnt/rwfs
```

Verify that the card is mounted on both /mnt/mmcblk0p1 and /mnt/mmcblk0p2

```
DM-37x# cat /proc/cmdline
 nand-ecc=chip console=ttyO0,115200n8 display=28 ignore_loglevel early_printk
no_console_suspend mtdparts=omap2-nand.0:512k(x-loader),1664k(u-boot),384k(u-boot-
env),5m(kernel),20m(ramdisk),-(fs) root=/dev/mmcblk0p2 rw rootfstype=ext3 rootwait DM-
37x#
```

Verify that the /proc/cmdline shows that root is on /dev/mmcblk0p2

5.11 Test 11 System-Wide Power Management

Purpose

This test verifies that the functional requirements for wireless power management are met.

Setup

As preparation for this test, perform the following:

- Load any standard image

Test Steps

Perform the following subtests and record results in the action log:

11.01 Suspend Resume

11.01.01 Suspend/Resume combinations

[TOUCH-01-011] [POWER-12-006]

Suspend resume via every possible combination available. (i.e. suspend via S2, wakeup S2)

NOTE: Do not use S2 to resume unless you use S2 to suspend

Suspends

- echo mem > /sys/power/state
- S2 button

Resumes

- Touch keyboard
- Touch display (lcd)
- S2 button

Verify that all combinations of ways to suspend and resume are able to be performed without error.

11.01.02 Resume via Touch

<Combined into 11.01.01 (01/01/13)>

11.01.03 Enter Suspend via S2 button

<Combined into 11.01.01 (01/01/13)>

11.01.04 Resume via Console

<Combined into 11.01.01 (01/01/13)>

11.01.05 Suspend/Resume Stress Test

[POWER-12-006]

Run the steps for Suspend / Resume stress testing:

```
DM-37x# echo 3 > /sys/kernel/debug/pm_debug/wakeup_timer_seconds
DM-37x# mkdir /tmp
DM-37x# cp /mnt/mmcblk0p1/suspend.sh /tmp/.
DM-37x# cd /tmp
DM-37x# chmod 777 suspend.sh
DM-37x# ./suspend.sh
```

Verify that the script runs without errors.

11.03 CPU Frequency

01.01.06 -

11.03.01 CPU Frequency

[POWER-12-013] [POWER-12-014]

NOTE: Hardware limits for DM3730 are 300 MHz - 1000 MHz and must be entered as Hz
cpufreq governors: conservative, ondemand, userspace, powersave, performance

Boot the device and login

Find out the max cpufreq programmed in Product ID chip

```
DM-37x# cat /sys/class/product_id/speed_mhz
```

View cpu frequency info:

```
DM-37x# cpufreq-info
```

<AFS> is the list of “available frequency steps”.

<Governor> is the current governor.

Note this value. It will be used in later step.

Verify *the maximum CPU frequency is the same as the one from the step above.*

Verify current CPU frequency is from the list of available frequencies.

This is the <default frequency>.

For each from Available Frequency Steps <AFS>:

```
DM-37x# cpufreq-set -f <AFS>
DM-37x# cpufreq-info
```

Verify CPU frequency is new AFS

Read out the time in state for each frequency twice:

```
DM-37x# cat /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state
DM-37x# cat /sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state
```

Verify transition time in state has been updated.

End For each from Available Frequency Steps <AFS>:

Reset the frequency to its default state.

```
DM-37x# cpufreq-set -f <default frequency>
DM-37x# cpufreq-info
```

Verify CPU frequency is set to the default frequency.

<IFS>: Each <AFS> + 1000 (i.e.300000 + 1000)

For each from Invalid Frequency Steps <IFS>:

```
DM-37x# cpufreq-set -f <IFS>
DM-37x# cpufreq-info
```


Verify that each frequency is the value of the <IFS> rounded up to the nearest <AFS> unless it is higher than the top <AFS> in which case it will round down.

End For each from Invalid Frequency Steps <IFS>:

Reset the frequency to its default state.

```
DM-37x# cpufreq-set -f <default frequency>
DM-37x# cpufreq-info
```

Verify CPU frequency is set to the default frequency.

<BFS>: Each <AFS> - 1000 (i.e.600000 – 1000)

For each from Border Frequency Steps <BFS>:

```
DM-37x# cpufreq-set -f <BFS>
DM-37x# cpufreq-info
```

Verify that each frequency is the value of the <BFS> rounded up to the nearest <AFS>.

End For each from Border Frequency Steps <BFS>:

Reset the frequency to its default state.

```
DM-37x# cpufreq-set -f <default frequency>
DM-37x# cpufreq-info
```

Verify CPU frequency is set to the default frequency.

<NULL>: blank, 0

For each from Invalid Frequency <NULL>:

```
DM-37x# cpufreq-set -f <NULL>
```

Verify the cpufreq help options is displayed.

End For each from Invalid Frequency <NULL>:

For each from <governor> = {performance, userspace, ondemand, conservative}:

Set the governor:

```
DM-37x# cpufreq-set -g <governor>
DM-37x# cpufreq-info
```

Verify governor is set to the selected governor.

End For each from <governor> = {performance, userspace, ondemand, conservative}:

Reset the frequency to its default state.

```
DM-37x# cpufreq-set -f <default frequency>
DM-37x# cpufreq-info
```

Verify CPU frequency is set to the default frequency.

Reset the governor to its default state.


```
DM-37x# cpufreq-set -g <governor>
DM-37x# cpufreq-info
```

Verify governor is set to the default governor.

POR

5.12 Test 12 Wireless Ethernet Tests

Purpose

To test the WL1283 wireless.

Setup

The following figure shows the recommended test setup.

Test Steps

Perform the following subtests and record results in the action log:

Testing Tools

The following tools are required for testing:

- Iperf: A traffic generator. The Iperf tool generates data at a given rate. It runs as an embedded component on the host PC. Iperf supports transport Layer 4 TCP and UDP protocols.
- An access point (AP i.e. a router).
- One workstation (STA).
- A DUT to test wireless on
- A Wifi analyzer application / program to find channels with minimal traffic

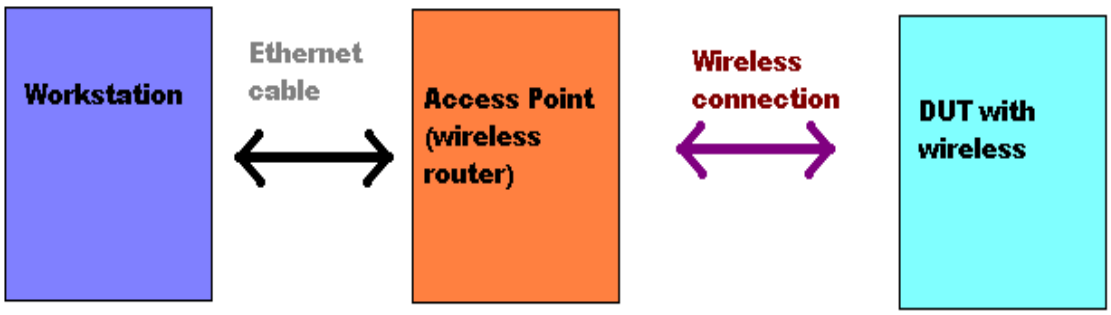
IMPORTANT!!! REMOVE THE ETHERNET CABLE TO ENSURE THAT ALL CONNECTIONS ARE MADE WITH WIRELESS!!!

**NOTE: You may need to turn off your workstations firewall to successfully ping it from the DUT.*

To reconfigure Wi-Fi in either AP or Station, mode execute either of the following commands and answer the questions:

```
modprobe wlcore_sdio
                # preload drivers
iw reg set US
                # set CRDA registry to US (need for 5 GHz mode)
/etc/rc.d/init.d/network-wifi-ap init          # start wireless in ap mode
or
/etc/rc.d/init.d/network-wifi-station init     # start wireless in station mode
```

**Note*: You can start/stop Wi-Fi in either mode by executing the previous script with "stop" or "start" (instead of "init") to stop/start the Wi-Fi and associated services.*



* SQA uses a wireless router that is approximately 1 – 3 feet from the DUTs, and may or may not have a single cubicle wall in between the router and the DUT.

12.01 Wireless connection

12.01.01 WPA Wireless connection – station mode

[WIFI-06-005] [WIFI-06-009] [WIFI-06-010] [WIFI-06-017]

POR

Setup the wireless station:

```
DM37x#      modprobe wlcore_sdio
DM37x#      iw reg set US
```



```
DM37x#      /etc/rc.d/init.d/network-wifi-station init
Set the SSID:
  wpa_supplicant: Enter WiFi SSID to connect to: <AP SSID>
Set the Encryption mode:
  wpa_supplicant: Enter Encryption mode (NONE, WEP40, WEP128, WPA, WPA2): WPA
Enter the WPA passkey
  wpa_supplicant: Enter WiFi WPA hex key: <WPA Passkey>
```

NOTE: setting cipher mode to AES and TKIP is recommended

Get an IP address from the system:

```
DM37x#      udhcpc -i wlan0
DM37x#      ifconfig
```

Verify that an IP address was given (wlan0)

Ping the workstation:

```
DM37x#      ping <Workstation IP Address>
```

On the workstation, ping the device:

```
c:\ ping <DUT IP Address>
```

Verify that the ping was successful on both the workstation and the DUT.

Bring the wireless connection down:

```
DM37x#      ifdown wlan0
```

Ping the workstation:

```
DM37x#      ping <Workstation IP Address>
```

Verify that the ping was unsuccessful.

Bring the wireless connection back up:

```
DM37x#      ifup wlan0
```

Ping the workstation:

```
DM37x#      ping <Workstation IP Address>
```

Verify that the ping was successful.

12.01.02 WPA2 Wireless connection – AP mode

[WIFI-06-009] [WIFI-06-010] [WIFI-06-011] [WIFI-06-017]

NOTE: This test must be done one board at a time. They will all have the same IP address.

POR

Setup the wireless access point (AP):

```
DM37x#      modprobe wlcore_sdio
DM37x#      iw reg set US
DM37x#      /etc/rc.d/init.d/network-wifi-ap init
```



```
iptables:NAT/masquerading necessary? Enter y/n: y
hostapd: Enter WiFi SSID to advertise: <wifi SSID>
hostapd: Enter mode (a = 5GHz, g = 2.4 GHz): g
hostapd: Enter WiFi channel to advertise on: 3
hostapd: Enter Encription mode (NONE, WEP40, WEP128, WPA, WPA2): WPA2
hostapd: Enter WiFi WPA2 hex key: <WPA2 Passkey>
```

**NOTE (use wifi analyzer to find channel that has minimal traffic [try 157 for 5GHz])*

Get the IP address of the DUT

```
DM37x#      ifconfig wlan0
```

Connect the workstation (or and other SOM)to the DUT using a the wireless connection with the SSID that you entered above

Ping the workstation (or other SOM):

```
DM37x#      ping <Workstation/other SOM IP Address>
```

On the workstation, ping the device:

```
c:\ ping <DUT IP Address>
```

Verify that the ping was successful on both the workstation / other SOM and the DUT.

Repeat the test at both 2.4 and 5 GHz (SomLV is 2.4 GHz only)

12.01.03 Static IP Address

[WIFI-06-002] [WIFI-06-003] [WIFI-06-004] [WIFI-06-007]

POR

Setup the wireless station:

```
DM37x#      modprobe wlcore_sdio
DM37x#      iw reg set US
DM37x#      /etc/rc.d/init.d/network-wifi-station init
Set the SSID:
  wpa_supplicant: Enter WiFi SSID to connect to: <AP SSID>
Set the Encription mode:
  wpa_supplicant: Enter Encription mode (NONE, WEP40, WEP128, WPA, WPA2): NONE
```

Give wlan0 a static IP address:

```
DM37x#  ifconfig wlan0 <static IP address> netmask <router netmask>
```

Check that the static ip address holds:

```
DM37x#  ifconfig wlan0
```

Verify that the IP address is the static IP address that you assigned above.

Ping the workstation:

```
DM37x#      ping <Workstation IP Address>
```

On the workstation, ping the device:

```
c:\ ping <DUT static IP Address>
```


Verify that the ping was successful on both the workstation and the DUT.

12.01.04 Changing wireless networks and starting and stopping

[WIFI-06-001] [WIFI-06-008]

POR

**NOTE*: Make sure that you have two SSIDs available to connect to.*

Connect to the first wireless SSID:

```
DM37x#      modprobe wlcore_sdio
DM37x#      iw reg set US
DM37x#      /etc/rc.d/init.d/network-wifi-station init
Set the SSID:
  wpa_supplicant: Enter WiFi SSID to connect to: <First SSID>
Set the Encription mode:
  wpa_supplicant: Enter Encription mode (NONE, WEP40, WEP128, WPA, WPA2): WPA
```

Get an IP address from the system:

```
DM37x#      udhcpc -i wlan0
DM37x#      ifconfig
```

Verify that you an IP address was given (wlan0):

Ping the workstation :

```
DM37x#      ping <Workstation IP Address>
```

On the workstation, ping the device:

```
c:\ ping <DUT IP Address>
```

Verify that the ping was successful on both the workstation and the DUT.

Shutdown the wireless connection:

```
DM37x#      /etc/rc.d/init.d/network-wifi-station stop
```

Ping the workstation:

```
DM37x#      ping <Workstation IP Address>
```

Verify that the ping was unsuccessful.

Restart the wireless connection:

```
DM37x#      /etc/rc.d/init.d/network-wifi-station start
```

Get an IP address from the system:

```
DM37x#      udhcpc -i wlan0
```

Ping the workstation:

```
DM37x#      ping <Workstation IP Address>
```

Verify that the ping was successful.

Connect to the second wireless SSID:


```
DM37x#      /etc/rc.d/init.d/network-wifi-station init
Set the SSID:
  wpa_supplicant: Enter WiFi SSID to connect to: <Second SSID>
Set the Encription mode:
  wpa_supplicant: Enter Encription mode (NONE, WEP40, WEP128, WPA, WPA2): WPA2
```

Get an IP address from the system:

```
DM37x#      udhcpc -i wlan0
DM37x# ifconfig wlan0
```

Verify that you an IP address was given (wlan0):

Ping the workstation :

```
DM37x#      ping <Workstation IP Address>
```

On the workstation, ping the device:

```
c:\ ping <DUT IP Address>
```

Verify that the ping was successful on both the workstation and the DUT.

12.02 Wireless throughput

12.02.01 Wireless Throughput and Power– OPEN mode , station mode

[PERF-11-010]

POR

For each from: Wireless {A,B,G,N}:

Setup the wireless station:

```
DM37x#      modprobe wlcore_sdio
DM37x#      iw reg set US
DM37x#      /etc/rc.d/init.d/network-wifi-station init
```

Set the SSID:

```
  wpa_supplicant: Enter WiFi SSID to connect to: <AP SSID>
Set the Encription mode:
  wpa_supplicant: Enter Encription mode (NONE, WEP40, WEP128, WPA, WPA2): NONE
```

Get an IP address from the system:

```
DM37x#      udhcpc -i wlan0
DM37x#      ifconfig
```

Verify that you an IP address was given (wlan0):

VERIFY THAT THE MAC ADDRESS OF WLAN0 ON ALL TORPEDO+WIRELESS IS A BEACON EMBEDDED MAC ADDRESS (I.E. STARTS WITH 00:08:EE) THIS IS THE RESULTS FOR TEST 16.09.02, AND SHOULD HAVE NO EFFECT ON WHETHER THIS TEST (12.02.01) PASSES OR NOT.

Connect the DUT to Wattson using a USB Standard-A to USB Mini-a cable connected to the DUT's USB OTG port.

Set the workstation in server mode in Iperf:

```
c:\ iperf -s -i 2 -p 50000
```

Set the DUT in client mode in iperf and run traffic for 20 seconds:

```
DM37x# iperf -c <workstation IP address> -i 2 -p 50000 -t 20
```

Enter the throughput (transmission rate) and Power statistics on the performance sheet of the TPL

Set the DUT in server mode in Iperf:

```
DM37x# iperf -s -i 2 -p 50000
```

Set the workstation in client mode in iperf and run traffic for 20 seconds:

```
C:\     iperf -c <DUT IP address> -i 2 -p 50000 -t 20
```

Enter the throughput (transmission rate) and Power statistics on the performance sheet of the TPL

End For each from: Wireless {A,B,G,N}

12.02.02 Wireless Throughput and Power– OPEN mode , AP mode

[PERF-11-010]

**NOTE*: This test must be done 1 board at a time. They will all have the same ip address.*

POR

Setup the wireless station:

```
DM37x#      modprobe wlcore_sdio
DM37x#      iw reg set US
DM37x#      /etc/rc.d/init.d/network-wifi-ap init
iptables:NAT/masquerading necessary? Enter y/n: y
hostapd: Enter WiFi SSID to advertise: beacon_testing
hostapd: Enter mode (a = 5GHz, g = 2.4 GHz): g
hostapd: Enter WiFi channel to advertise on: 3
hostapd: Enter Encription mode (NONE, WEP40, WEP128, WPA, WPA2): NONE
```

Get an IP address from the system:

```
DM37x#      ifconfig
```

Verify that you an IP address was given (wlan0):

Connect the DUT to Wattson using a USB Standard-A to USB Mini-a cable connected to the DUT's USB OTG port.

Set the workstation in server mode in Iperf:

```
c:\ iperf -s -I 2 -p 50000
```

Set the DUT in client mode in iperf and run traffic for 20 seconds:

```
DM37x#      iperf -c <workstation IP address> -i 2 -p 50000 -t 20
```

Enter the throughput (transmission rate) and Power statistics on the performance sheet of the TPL

Set the DUT in server mode in Iperf:

```
DM37x# iperf -s -I 2 -p 50000
```

Set the workstation in client mode in iperf and run traffic for 20 seconds:

```
C:\  iperf -c <DUT IP address> -i 2 -p 50000 -t 20
```

Set the workstation in client mode in iperf (-R to reverse transfer direction) and run traffic for 20 seconds:

```
C:\  iperf -c <DUT IP address> -i 2 -p 50000 -t 20 -R
```

Enter the throughput (transmission rate) and Power statistics on the performance sheet of the TPL

Repeat this test in 2.4 and 5 GHz modes

12.03 Wireless connection after suspend / resume

12.03.01 Wireless connection after suspend / resume

[WIFI-06-013]

POR

Following the directions in the previous tests, setup a wireless connection in either station mode or AP mode, with either open or WEP security.

Start pinging the workstation (keep it running by starting the ping in the background):

```
DM-37x#      ping <Workstation IP Address> &
```

Verify that the ping is running.

Put the board into a suspend stated:

```
DM-37x# echo mem > /sys/power/state
```

Verify that the board has dropped into suspend.

Wake up the board by typing a character into Teraterm.

Verify that the ping resumes after the board wakes up.

12.04 Ad-Hoc Networking

12.04.01 Create an ad-hoc network on the SOM.

<Test Removed>

12.04.02 Connect to the ad-hoc network with the Workstation (Windows 7)

<Test Removed>

12.04.03 Connect to the ad-hoc network with another SOM.

<Test Removed>

12.05 CRDA

12.05.01 Using updated CRDA

< Test Removed 09/28/2015 >

12.06 Wireless Ethernet uses open source driver

12.06.01 Wireless Ethernet uses open source driver

[WIFI-13-017]

On the Linux build machine read out the wl12xx.h file:

```
logic@logic-Virtualbox:  clear; cat
~/logic/Logic_BSPs/Linux_3.0/1027480_LogicPD_Linux_BSP_2.4-
4/rpm/BUILD/linux/drivers/net/wireless/wl12xx/wl12xx.h | head -30
```

Verify that you see an output similar to below and that it specifically mentions the GNU General Public License version 2.

```
/*
* This file is part of wl1271
*
* Copyright (C) 1998-2009 Texas Instruments. All rights reserved.
* Copyright (C) 2008-2009 Nokia Corporation
*
* Contact: Luciano Coelho <luciano.coelho@nokia.com>
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License
* version 2 as published by the Free Software Foundation.
*
* This program is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
* 02110-1301 USA
*
*/
```

5.13 Test 13 Bluetooth Tests

Purpose

This test verifies that the functional requirements for Bluetooth interface are met.

Setup

As preparation for this test, execute the following in sequence:

- Load image build from /config/platform/omap_logic/defconfig.new-bluez. This will include Bluez4.101 stack or later.
- Flash image into NAND using 'run makeyaffsboot'
- Attach wireless antenna to socket J7 on SOM.
- Prepare a Bluetooth-capable device (Laptop, Bluetooth adapter, etc) with known Bluetooth address.
- Antenna required (Bluetooth may function without; however, with antenna is the intended use case).
- TI SensorTagV2 (CC2650STK) required for BLE testing
- After booting and logging in, execute the following commands:

  (The first 2 commands are not required for Bluez v2.0)

  - DM-37# /usr/bin/dbus-daemon --config-file=/etc/dbus-1/system.conf --fork
  - DM-37# bluetoothd
  - DM-37# modprobe st_drv
  - DM-37# /home/root/wl12xx/uim&
  - DM-37# modprobe btwilink

Test Steps

Perform the following subtests and record results in the action log:

13.01 Start/Stop Bluetooth interface

13.01.01 Verify the Bluetooth interface can be successfully started

[BT-01-001]

Bring up Bluetooth interface:

```
DM-37x# /usr/bin/dbus-daemon --config-file=/etc/dbus-1/system.conf --fork
DM-37x# bluetoothd
DM-37x# modprobe st_drv
DM-37x# /home/root/wl12xx/uim&
DM-37x# modprobe btwilink
DM-37x# hciconfig hci0 up
DM-37x# hcitool scan
```

Verify the wireless network can be seen.

13.01.02 Verify the Bluetooth interface can be successfully terminated

[BT-01-001]

Continue from previous test.

Bring down Bluetooth interface (must be up):

```
DM-37x# hciconfig hci0 down
DM-37x# hcitool scan
```

Verify the wireless network cannot be seen (error message "Device is not available: No such device")

13.02 Configure Bluetooth interface

13.02.01 Modify Bluetooth configuration. Verify a name is successfully assigned to the device.

[BT-01-002]

Continue from previous test.

Give the device a Bluetooth name:

```
DM-37x# hciconfig hci0 up
DM-37x# hciconfig hci0 name <name> {where <name> is at least 4 characters}
DM-37x# hciconfig -a hci0
```

Verify the Name was successfully assigned to the device.

13.02.02 Modify Bluetooth configuration. Verify a device class is successfully assigned to the device.

[BT-01-002]

Continue from previous test.

Designate the Bluetooth device as "Computer, Laptop":

```
DM-37x# hciconfig hci0 class 0x0a010c {new Device class is "Computer, Laptop"}
DM-37x# hciconfig -a hci0
```

Verify the Bluetooth configuration is displayed.

Verify the Device Class: Computer, Laptop was successfully assigned to the device.

13.02.03 View the hcitool info of the SOM module.

[BT-01-003]

Continue from previous test.

Make the Bluetooth device discoverable:

```
DM-37x# hciconfig hci0 piscan
```

On a computer or another DUT scan to see if the DUT is listed in the scan (is discoverable):

```
bash#  hcitool scan
bash#  hcitool info <<SOMBluetooth_address>
Example output:
Requesting information ...
        BD Address:  00:18:30:4E:B7:E9
        OUI Company: Texas Instruments (00-18-30)
        Device Name: Beacon
        LMP Version: 4.0 (0x6) LMP Subversion: 0x2b5d
        Manufacturer: Texas Instruments Inc. (13)
        Features page 0: 0xff 0xfe 0x2d 0xfe 0xdb 0xff 0x7b 0x87
                <3-slot packets> <5-slot packets> <encryption> <slot offset>
                <timing accuracy> <role switch> <hold mode> <sniff mode>
                <RSSI> <channel quality> <SCO link> <HV2 packets>
                <HV3 packets> <u-law log> <A-law log> <CVSD> <power control>
                <transparent SCO> <EDR ACL 2 Mbps> <EDR ACL 3 Mbps>
                <enhanced iscan> <interlaced iscan> <interlaced pscan>
                <inquiry with RSSI> <extended SCO> <EV4 packets> <EV5 packets>
                <AFH cap. slave> <AFH class. slave> <LE support>
                <3-slot EDR ACL> <5-slot EDR ACL> <sniff subrating>
                <pause encryption> <AFH cap. master> <AFH class. master>
                <EDR eSCO 2 Mbps> <EDR eSCO 3 Mbps> <3-slot EDR eSCO>
                <extended inquiry> <LE and BR/EDR> <simple pairing>
                <encapsulated PDU> <err. data report> <non-flush flag> <LSTO>
                <inquiry TX power> <EPC> <extended features>
```

Verify hcitool info can be retrieved from the module.

13.03 Scan and inquiry remote Bluetooth devices

13.03.01 Verify remote Bluetooth devices name and addresses are displayed when scanned.

[BT-01-003]

Continue from previous test.

On a computer or another DUT set it as discoverable:

```
bash# hciconfig hci0 piscan
```

Perform a Bluetooth scan:

```
DM-37x# hcitool scan
```

Verify the name and Bluetooth address for each detected device is displayed.

13.03.02 Verify a particular remote Bluetooth device may be queried.

[BT-01-003]

Continue from previous test.

View the info of another Bluetooth device (from scan):

```
DM-37x# hcitool info <remoteBluetooth_address>
Example Output:
Requesting information ...
        BD Address:  00:25:CA:14:A6:DE
        Device Name: beacon-imx8mm-kit
        LMP Version:  (0x7) LMP Subversion: 0x6109
        Manufacturer: not assigned (305)
        Features page 0: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x8f
                <3-slot packets> <5-slot packets> <encryption> <slot offset>
                <timing accuracy> <role switch> <sniff mode> <RSSI>
                <channel quality> <SCO link> <HV2 packets> <HV3 packets>
                <u-law log> <A-law log> <CVSD> <paging scheme> <power control>
                <transparent SCO> <broadcast encrypt> <EDR ACL 2 Mbps>
                <EDR ACL 3 Mbps> <enhanced iscan> <interlaced iscan>
                <interlaced pscan> <inquiry with RSSI> <extended SCO>
                <EV4 packets> <EV5 packets> <AFH cap. slave>
                <AFH class. slave> <LE support> <3-slot EDR ACL>
                <5-slot EDR ACL> <sniff subrating> <pause encryption>
                <AFH cap. master> <AFH class. master> <EDR eSCO 2 Mbps>
                <EDR eSCO 3 Mbps> <3-slot EDR eSCO> <extended inquiry>
                <LE and BR/EDR> <simple pairing> <encapsulated PDU>
                <err. data report> <non-flush flag> <LSTO> <inquiry TX power>
                <EPC> <no. 59> <extended features>
        Features page 1: 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        Features page 2: 0x30 0x08 0x00 0x00 0x00 0x00 0x00 0x00
```

Verify information particular to that device is displayed.

**NOTE: You might need to try a couple of different devices if the first one does not return any values.*

13.04 Data transfer with a remote Bluetooth device

13.04.01 Successfully ping a remote Bluetooth device.

[BT-01-004]

Continue from previous test.

Perform an l2ping (usually between SOMs):

```
DM-37x# l2ping -s 200 -c 5 -t 10  <remoteBluetooth_address>
```

Verify a response is received from the remote Bluetooth device.

**NOTE: You might need to try to ping a couple of different devices or a different SOM if the first one does not return any values. Try to ping "DUT A" from and "DUT B" from DUT A at the same time (when the first one starts, start the next one.*

Enter the AVERAGE BT packet transmit time (BT only) on the performance sheet of the TPL

13.04.02 Bluetooth l2ping + Wireless iperf simultaneously

[BT-01-004]

**NOTE: For T+W, run this test with the new wifi module (wlcore_sdio) and Bluez bluetooth module.*
**NOTE: Not applicable on SomLVs*

POR

Bring up Bluetooth interface:

```
DM-37x# /usr/bin/dbus-daemon --config-file=/etc/dbus-1/system.conf --fork
DM-37x# bluetoothd
DM-37x# modprobe st_drv
DM-37x# /home/root/wl12xx/uim&
DM-37x# modprobe btwilink
```

Setup the wireless connection on the DUT:

```
DM-37x# modprobe wlcore_sdio
DM-37x# iw reg set US
DM-37x# /etc/rc.d/init.d/network-wifi-station init
```

Set the SSID, Encryption and Passkeys as necessary for the connection.

On the workstation, setup iperf in server mode

```
C:/  iperf -s -i 2 -p 50000
```

Perform an l2ping (usually between SOMs):

```
DM-37x# l2ping -s 200 -c 7 -t 10 <remoteBluetooth_address> &
```

Verify that both the iperf command and l2ping were successful.

Run iperf for 30 seconds to get numbers of iperf alone.

Enter the AVERAGE iperf transmission rate (iperf only) on the performance sheet of the TPL

```
DM-37x# iperf -c <workstation_ip_address> -i 2 -p 50000 -t 7
```

**NOTE*: BEWARE OF ISSUE WITH COPIED DASHES(-) AS LONG DASHES*

Enter the AVERAGE iperf and BT packet transmission rates (BT + wireless simultaneously) on the performance sheet of the TPL.

13.05 Support customers to program and build Bluetooth applications.

13.05.01 Verify the Bluez library and utilities built.

[BT-01-005]

<This is tested by 13.04.01. (If l2ping is successful, then the Bluez library and utilities built.)>

13.05.02 Verify the Bluetooth demo application performs successfully

[BT-01-005]

<This is tested by 13.01.01. (If hcitool scan is successful, then the demo application performs successfully.) >

13.05.03 Verify the Bluetooth demo application performs successfully

[BT-01-005] [BT-01-009]

Continue from previous test.

View the version of bluez in the SOM:

```
DM-37x# cat /usr/lib/pkgconfig/bluez.pc
```

Verify that the version is 4.101.

13.06 Verify the version Linux BSP can connect to a BLE device

13.06.01 Connect to a BLE device

[BT-01-009] [BT-01-010]

Continue from previous test.

Capture the SensorTagV2 MAC address using 'hcitool lescan'. You must use CNTL-C to escape the continuous scan.

```
DM-37x# hcitool lescan
```

The example output below has a MAC address of B0:B4:48:B8:BE:00. Your SensorTagV2 will have a different MAC. Capture your address to be used in the steps below.

```
DM-37x# hcitool lescan
LE Scan ...
B0:B4:48:B8:BE:00 (unknown)
B0:B4:48:B8:BE:00 CC2650 SensorTag
```


Verify that the output shows the SensorTagV2 and provides its MAC address.

Start GATTTool in interactive mode:

```
DM-37x# gatttool -i hci0 -b <MAC ADDRESS> -I
```

Here is an example showing the expected output. You must enter 'connect' once you receive the initial prompt following the 'GATTTOOL' command.

```
DM-37x# gatttool -i hci0 -b B0:B4:48:B8:BE:00 -I
[   ][B0:B4:48:B8:BE:00][LE]> connect
[CON][B0:B4:48:B8:BE:00][LE]>
```

Verify that the connection was successfully created (The [CON] seen in the third line above represents a connection to the SensorTag..)

Read the ScanTagV2 Device Name:

Read out the Device Name of the SensorTagV2

```
[CON][B0:B4:48:B8:BE:00][LE]> char-read-hnd 0x02
Characteristic value/descriptor: 02 03 00 00 2a
```

Verify that the SensorTagV2 EVM has a value of '02 03 00 00 2a' for its device name.

13.06.02 Connection information holds through reboot / power cycle

< Test Removed >

5.14 Test 14 DSP Tests

Purpose

This section tests the functionality of the DVDSK (DSP) build.

Setup

The tests in this section of the STP should be run using the build created in test 18.03.01 from section 18 of this STP.

**NOTE: These tests requires SOMs with DSP, therefore they are not applicable to AM3703 SOMs.*

Test Steps

Perform the following subtests and record results in the action log:

14.01 DSPLINK

14.01.01 DSPLink Tests

[DSP-11-001]

Change the DSPLink directory:

```
DM-37x#              cd /usr/share/ti/ti-dsplink-examples
```

Run the DSPLink test:

```
DM-37x#              ./loopgpp ./loop.out 1000 5000 0
```

Verify that the test completed running without erroring out.

14.02 C6Accel

14.02.01 C6Accel Tests

<Test removed since C6Accel is no longer supported by TI (11/12/2012)>

14.03 C6Run

14.03.01 C6Run Tests

<Test removed since C6Run is no longer supported by TI (11/12/2012)>

14.04 DMAI

14.04.01 DMAI Tests

[DSP-11-001]

**NOTE: This test is NOT applicable to AM3703 SOMs.*

Change to the DMAI directory:

```
DM-37x               cd /usr/share/ti/ti-dmai-apps/
```

Run the DMAI tests:

```
DM-37x               ./image_encode_io1_dm3730.x470MV -c jpegenc -i /dev/zero -o
jpeg_test_encoded.jpeg -r 720x576 --iColorSpace 3 --benchmark
```

Verify that the command completed without errors.

```
DM-37x               ./image_encode_io1_dm3730.x470MV -c jpegenc -i /dev/urandom -o
jpeg_test_encoded.jpeg -r 720x576 --iColorSpace 3 --benchmark
```

Verify that the command completed without errors.

DM-37x **./image_decode_io1_dm3730.x470MV -c jpegdec -i jpeg_test_encoded.jpeg -o jpeg_test_decoded.yuv --oColorSpace 3 --benchmark**

Verify that the command completed without errors.

DM-37x **./video_encode_io1_dm3730.x470MV -c h264enc -i /dev/zero -n 100 -o h264_test_encoded.h264 -r 720x576 --benchmark**

Verify that the command completed without errors.

DM-37x **./video_encode_io1_dm3730.x470MV -c h264enc -i /dev/urandom -n 10 -o h264_test_encoded.h264 -r 720x576 --benchmark**

Verify that the command completed without errors.

DM-37x **./video_decode_io2_dm3730.x470MV -c h264dec -i h264_test_encoded.h264 -n 10 -o h264_test_decoded.yuv --benchmark**

Verify that the command completed without errors.

14.05 GStreamer

14.05.01 GStreamer Tests

<Removed. This test is similar to Test 20 - Camera. (11/12/2012)>

14.06 DSP testing for 2011 toolchain

14.06.01 DSP testing for 2011 toolchain

[DSP-11-001]

Perform the tests 14.01.01 and 14.04.01 the 2011 toolchain build from 18.03.02.

Verify that the testing passes correctly.

5.15 Test 15 USB OTG Tests

Purpose

This test verifies that the functional requirements for USB OTG host and USB OTG device are met.

Setup

As preparation for this test, the following information is critical:

- Any standard image may be loaded.
- On the baseboard, ensure a jumper is not placed at J42.
- For all OTG tests, always insert and remove the USB Device AND the adapter from the USB OTG port.

For USB OTG Host tests:

- A USB OTG Mini-A to Mini-B cable plugged into the device at boot. The Mini-A end should be plugged into the device. An "A" is embossed on the Mini-A connector end of the cable. The Mini-B connector end of the cable may, or may not, be connected to a device.
- A USB OTG Mini-A to Standard-A (female) adapter may be used, and must also be plugged into the device at boot. The Mini-A end should be plugged into the device. The Standard-A connector end of the cable may, or may not, be connected to a device.
- Beacon EmbeddedWorks uses an USB OTG Mini-A to Standard-A (female) adapter. The adapter was obtained from Target and was included in the 6-in-1 USB Cable Kit.

For USB OTG device tests:

- Beacon EmbeddedWorks supplies a USB Mini-B to Standard-A (male) cable as part of the development kit. This cable should be used when testing USB OTG device mode. It may or may not be plugged in at boot.

Test Steps

Perform the following subtests and record results in the action log:

15.01 USB OTG Host-Mode Tests – USB Sub-System

15.01.01 USBFS

[USBOTG-01-001]

```
DM-37x# mount | grep usbfs
```

Verify that usbfs is mounted on /proc/bus/usb.

15.01.02 HID – Keyboard / Mouse

[USBOTG-01-001]

Plug a USB keyboard into the USB OTG Mini-A to Standard-A (female) adapter

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify by checking for an 'I:' line with Driver=usbhid.

Verify that the matching 'E:' line has the interrupt attribute. (Atr=03(Int.))

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Use hexdump to see input from USB device:

DM-37x# **hexdump /dev/eventX** (or **hexdump /dev/input/eventX**, where eventX is the device found for the new device. X is usually 2.)

Verify that pressing keys on the keyboard results in data on the device

Press ctrl-c

Ensure the hexdump command closes properly

Repeat the previous 4 steps three times.

Remove the USB OTG Mini-A to Standard-A (female) adapter from the USB OTG port.

DM-37x# **cat /proc/bus/usb/devices**

Verify the device has been removed.

Read out dmesg:

DM-37x# **dmesg**

Verify the device has been removed.

Plug a USB Mouse into the USB OTG Mini-A to Standard-A (female) adapter

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port.

DM-37x# **cat /proc/bus/usb/devices**

Verify by checking for an 'I:' line with Driver=usbhid.

Verify that the matching 'E:' line has the interrupt attribute. (Atr=03(Int.))

Read out dmesg:

DM-37x# **dmesg**

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Use hexdump to see input from USB device:

DM-37x# **hexdump /dev/eventX** (or **hexdump /dev/input/eventX**, where eventX is the device found for the new device. X is usually 2.)

Verify that moving the mouse results in data on the device

Press ctrl-c

Ensure the hexdump command closes properly

Repeat the previous 4 steps three times.

Remove the USB OTG Mini-A to Standard-A (female) adapter from the USB OTG port.

DM-37x# **cat /proc/bus/usb/devices**

Verify the device has been removed.

Read out dmesg:

DM-37x# **dmesg**

Verify the device has been removed.

15.01.03 USB Mass Storage

[USBOTG-01-001] [USBOTG-14-018] [USBOTG-14-020] [USBOTG-14-021] [USBOTG-14-022] [USBOTG-14-023]

Make sure the SD card has a text file called README.TXT

For each USB device in (SanDisk 256MB Cruzer USB stick, SanDisk 16 GB Cruzer USB stick, 500GB Seagate external hard drive, Western Digital WD Elements 1.5 TB External Hard Drive):

Plug all 4 of the USB devices into a powered USB hub.

Plug the USB devices into a USB OTG Mini-A to Standard-A (female) adapter.

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port on the baseboard.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the matching 'E:' line has the interrupt attribute. (Atr=02(Bulk))

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Note the name of the USB device. It should be sdx1 where x is a, b, etc.

```
DM-37x# mount | grep sd
```

Copy the README.TXT file onto the USB device:

```
DM-37x# cat /mnt/mmcblk0p1/README.TXT
DM-37x# cp /mnt/mmcblk0p1/README.TXT /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is listed

Read the copied file:

```
DM-37x# cat /mnt/sda1/README-2.TXT
```

Verify the content of README-2.TXT is the same as README.TXT

Remove the copied file:

```
DM-37x# rm /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is NOT listed

Unplug the device.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify the device has been removed.

Read out dmesg:

```
DM-37x# dmesg
```

Verify the device has been removed

End For each USB device in (SanDisk 256MB Cruzer USB stick, SanDisk 16 GB Cruzer USB stick, 500GB Seagate external hard drive, Western Digital WD Elements 1.5 TB External Hard Drive):

15.01.04 USB On Boot

[USBOTG-01-001] [USBOTG-14-018] [USBOTG-14-020] [USBOTG-14-021] [USBOTG-14-022] [USBOTG-14-023]

Make sure the SD card has a text file called README.TXT

01.01.07 -

For each USB device in (SanDisk 256MB Cruzer USB stick, SanDisk 16 GB Cruzer USB stick, 500GB Seagate external hard drive, Western Digital WD Elements 1.5 TB External Hard Drive):

Plug all 4 of the USB devices into a powered USB hub.

Plug the USB hub into a USB OTG Mini-A to Standard-A (female) adapter.

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port on the baseboard.

Power on device

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the matching 'E:' line has the interrupt attribute. (Atr=02(Bulk))

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about setting up the device just plugged in.

Note the name of the USB device. It should be sdx1 where x is a, b, etc.

```
DM-37x# mount | grep sd
```

Copy the README.TXT file onto the USB device:

```
DM-37x# cat /mnt/mmcblk0p1/README.TXT
DM-37x# cp /mnt/mmcblk0p1/README.TXT /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is listed

Read the copied file:

```
DM-37x# cat /mnt/sda1/README-2.TXT
```

Verify the content of README-2.TXT is the same as README.TXT

Remove the copied file:

```
DM-37x# rm /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```


Verify file README-2.TXT is NOT listed

Unplug the device.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify the device has been removed.

Read out dmesg:

```
DM-37x# dmesg
```

Verify the device has been removed

End For each USB device in (SanDisk 256MB Cruzer USB stick, SanDisk 16 GB Cruzer USB stick, 500GB Seagate external hard drive, Western Digital WD Elements 1.5 TB External Hard Drive):

15.01.05 USB Audio

[USBOTG-01-001]

**Note: The kernel does not have a USB Audio Class. Note however, that USB Audio behaves the same as USB Camera and therefore, it depends on the device.*

Plug a USB Audio device into the USB OTG Mini-A to Standard-A (female) adapter

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the matching 'E:' line has the isochronous attribute set. (Atr=(Isoc))

Read out dmesg:

```
DM-37x# dmesg
```

Verify the most recent lines say that a new audio device has been registered. As R4-01-004 is "should", it's not a requirement for release. Use an audio/linux driver that works together to test.

15.01.06 USB Web Camera

[USBOTG-01-001]

**Note: Use the DSP + Camera build from test 18.03.01 for this test.*

**Note: Use a UVC Web camera for this test. The USB Webcam may indicate audio functionality if functional.*

Plug a USB web camera device into a powered hub. Plug the hub into the USB OTG Mini-A to Standard-A (female) adapter.

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify by checking for an 'I:' line with Driver=uvcvideo

Read out dmesg:


```
DM-37x# dmesg
```

Verify the most recent lines starting with USB talk about an UVCVideo device.

15.02 USB OTG Host-Mode Tests – USB Spec

15.02.01 USB 1.1

[USBOTG-01-001] [USBOTG-14-011]

Plug a USB 1.1 device into the device using a USB OTG Mini-A to Mini-B cable or adapter.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device is listed and that the 'D:' line for the device shows Ver= 1.10.

15.02.02 USB 2.0

[USBOTG-01-001] [USBOTG-14-012]

Plug a USB 2.0 device into the device using a USB OTG Mini-A to Mini-B cable or adapter.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device is listed and that the 'D:' line for the device shows Ver= 2.00.

15.02.03 Low Speed

[USBOTG-01-001] [USBOTG-14-013]

Plug a low-speed USB device into the device using a USB OTG Mini-A to Mini-B cable or adapter.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device's 'T:' line shows Spd=1.5.

15.02.04 Full Speed

[USBOTG-01-001] [USBOTG-14-014]

Plug a full-speed USB device into the device using a USB OTG Mini-A to Mini-B cable or adapter.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device's 'T:' line shows Spd=12.

15.02.05 High Speed

[USBOTG-01-001] [USBOTG-14-015]

Plug a high-speed USB device into the device using a USB OTG Mini-A to Mini-B cable or adapter.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the device's 'T:' line shows Spd=480.

15.02.06 Hubs

[USBOTG-01-001] [USBOTG-14-016] [USBOTG-14-017]

Plug an unpowered hub into a the USB OTG Mini-A to Standard-A adapter.

Plug a USB Storage device into the hub.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that both the hub and the storage device are listed.

Remove the USB OTG Mini-A to Mini-B adapter and hub.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that removal was detected.

Reboot

Plug a powered hub into a USB OTG Mini-A to Mini-B cable. Connect the Mini-A connector on the cable in to the device.

Plug a USB Storage device into the hub.

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that both the hub and the storage device are listed.

Remove the USB OTG Mini-A to Mini-B cable

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that removal was detected (USB disconnect message).

15.03 USB OTG Device-Mode Tests

15.03.01 Ethernet over USB (usbnet) - USBOTG mass storage device – Ethernet over USB (USBOTG)

[USBOTG-01-003] [USBOTG-01-005]

Boot the device using a standard image.

Connect the device under test to a Linux workstation using a USB mini-B to Standard-A cable (one comes in the kit).

Verify message "g_ether gadget: …" is displayed.

Configure the Linux workstation to use the new network connection:

Enable the connection

```
bash$ sudo ifconfig usb0 172.20.1.1 netmask 255.0.0.0 up
```

Verify the configuration.

```
bash$ ifconfig usb0
```

Configure the device under test to use the network connection:

Enable the connection

```
DM-37x# ifconfig usb0 172.20.1.2 netmask 255.0.0.0 up
```

Verify the configuration

```
DM-37x# ifconfig usb0
```

Test the connection by pinging from workstation to device and device to workstation:

```
bash$ ping 172.20.1.2
DM-37x# ping 172.20.1.1
```

Unplug the USB OTG cable.

*Verify that the device can switch from *device* mode to *host* mode.*

Plug a USB mass storage device into a USB OTG Mini-A to Standard-A (female) adapter.

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port on the baseboard.

Try to read the README.TXT file from the mass storage device (or other file):

```
DM-37x# cat /mnt/sda1/README.TXT
```

Verify that the output from the cat command is as expected.

Remove the mass storage device and adapter.

*Verify that the device can switch from *host* mode to *device* mode.*

Connect the device under test to a Linux workstation using a USB mini-B to Standard-A cable (one comes in the kit).

Verify message "g_ether gadget: …" is displayed

Configure the Linux workstation to use the new network connection:

Enable the connection

```
bash$ sudo ifconfig usb0 172.20.1.1 netmask 255.0.0.0 up
```

Verify the configuration

```
bash$ ifconfig usb0
```

Configure the device under test to use the network connection:

Enable the connection

```
DM-37x# ifconfig usb0 172.20.1.2 netmask 255.0.0.0 up
DM-37x# ifconfig usb0
```

Verify the configuration

Cleanup:

Remove the connection from the workstation

```
bash$ sudo ifconfig usb0 down
```

Unplug the USB OTG cable

Halt the device under test

Reboot the device

15.03.02 File Backed Storage (FSG) gadget support

[USBOTG-01-004]

Boot the device using the standard image.

Remove the Ethernet kernel module:

```
DM-37x#  rmmod g_ether
```

Ensure a message is displayed indicating the module was removed.

Load the file backed storage kernel module:

To expose the file backed storage, we need to load the appropriate kernel module; passing it as a parameter the name of the file or device we want it to use. Assume you have an ext2fs or FAT file system on an SD/MMC card living at /dev/mmcblk0p1 on /mnt/mmcblk0p1.

Unmount the SD card before proceeding:

```
DM-37x# umount /mnt/mmcblk0p1
```

Load the kernel module:

```
DM-37x# modprobe g_file_storage file=/dev/mmcblk0p1
```

Connect the device under test to a Linux workstation using a USB mini-B to Standard-A cable (one comes in the kit). At some point, Ubuntu or other modern distributions should automatically detect the device and mount it as normal removable storage. Note: this may take some time to finish (1-2 minutes). It is complete when the Linux File Browser window pops up.

On the Linux desktop, create a 16MB file of random data in the root directory:

```
bash$ dd if=/dev/urandom of=testdata bs=1K count=16K
bash$ md5sum testdata
```

Copy the file to the device under test:

**NOTE: You /may/ need to be root to do this, depending on how your laptop is set up. For example, on a Ubuntu system, the device is likely to be mounted on /media/disk-1. Copying the file to the device then requires:*

```
bash$ sudo cp testdata /media/disk-1
```

On the laptop, unmount the device under test in the bash window or File Browser application.

```
bash$ sudo umount /media/disk-1
```

Unplug the device's USB OTG cable from the Linux workstation.

On the device, mount the file backed storage and verify that the file copied exists and was copied correctly


```
DM-37x# mount /dev/mmcblk0p1 /mnt/mmcblk0p1
DM-37x# ls -l /mnt/mmcblk0p1
DM-37x# md5sum /mnt/mmcblk0p1/testdata
```

Verify the md5sum from the Linux workstation matches that of the device.

15.04 USB OTG driver functionality after resuming from suspend

15.04.01 Verify USB OTG device works after resuming from suspend

[USBOTG-01-007]

Make sure the SD card has a text file called README.TXT

Plug the USB hub into a USB OTG Mini-A to Standard-A (female) adapter.

Plug the USB OTG Mini-A to Standard-A (female) adapter into the USB OTG port on the baseboard.

Power on device

```
DM-37x# cat /proc/bus/usb/devices
```

Verify that the matching 'E:' line has the interrupt attribute. (Atr=02(Bulk))

Note the name of the USB device. It should be sdx1 where x is a, b, etc.

```
DM-37x# mount | grep sd
```

Copy the README file to the USB device:

```
DM-37x# cat /mnt/mmcblk0p1/README.TXT
DM-37x# cp /mnt/mmcblk0p1/README.TXT /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is listed

Read the copied file:

```
DM-37x# cat /mnt/sda1/README-2.TXT
```

Verify the content of README-2.TXT is the same as README.TXT

Remove the copied file:

```
DM-37x# rm /mnt/sda1/README-2.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-2.TXT is NOT listed

Put device into suspend mode:

```
DM-37x# echo mem > /sys/power/state
```

Verify the device is in suspend mode.

Hit a key on the console to wake the device up.

Verify that the device is awake.

Copy the README file to the USB device:

```
DM-37x# cp /mnt/mmcblk0p1/README.TXT /mnt/sda1/README-3.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-3.TXT is listed

Read the copied file:

```
DM-37x# cat /mnt/sda1/README-3.TXT
```

Verify the content of README-3.TXT is the same as README.TXT

Remove the copied file:

```
DM-37x# rm /mnt/sda1/README-3.TXT
DM-37x# ls /mnt/sda1
```

Verify file README-3.TXT is NOT listed

5.16 Test 16 General Tests

Purpose

This test verifies that the functional requirements for general features such as Real Time Clock (RTC), product ID and GPIO are met.

Setup

As preparation for this test, the following information is critical:

- Any standard image may be loaded

Test Steps

Perform the following subtests and record results in the action log:

16.01 Real Time Clock (RTC)

01.01.08 -

16.01.01 Time transitions from am to pm, Day, Month & Year remain the same

[RTC-08-001] [RTC-08-002] [RTC-08-003]

**NOTE: In this set of tests you will be setting the date/time and watching as the date/time/ transitions between states (i.e. am to pm, year to year, etc...) and the process from setting the time to setting up the while loop to observe the transitions needs to be done relatively quickly, as the transition will happen in 20 seconds.*

Set the date to just before noon on 10/02/2010 :

```
DM-37x# date
DM-37x# date 2010.10.02-11:59:40
```


```
DM-37x# date
```

Verify the system date

Write the system time into the RTC and observe the transitions:

```
DM-37x# hwclock -w
DM-37x# while true; do hwclock -r; sleep 1; done
```

Verify the time transitions from am to pm

Verify the month, day and year remains 10/02/2010

Ctl-Z to return to the DM-37x# prompt

Note time for comparison purposes:

DM-37x# **date**

Reset the board (press the Reset/S1 button)

Verify the date is the same as the previous date (prior to RESET)

Verify the time is proportionate to the previous time (prior to RESET)

End Session (POR)

Enter Session

Login at the login prompt DM-37x# prompt

```
DM-37x# date
```

For the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 Torpedo + Wireless SOM, verify that the date is the same as the previous date (prior to RESET)

For the DM3730/AM3703 Torpedo SOM and the DM3730/AM3703 Torpedo + Wireless SOM, verify the time is proportionate to the previous time (prior to RESET)

16.01.02 Time transitions from pm to am, Day increments, Month & Year remains the same

[RTC-08-001] [RTC-08-002]

Set the date to just before Midnight 10/02/2010:

```
DM-37x# date
DM-37x# date 2010.10.02-23:59:40
DM-37x# date
```

Verify the system date

Write the system time into the RTC and observe the transitions:

```
DM-37x# hwclock -w //write the system time into the RTC
DM-37x# while true; do hwclock -r; sleep 1; done
```

Verify the time transitions from pm to am

Verify the day increments to 03

Verify the month remains 10

Verify the year remains 2010

Ctl-Z to return to the DM-37x# prompt

16.01.03 Time transitions from pm to am, Month, Day and Year increment

[RTC-08-001] [RTC-08-002]

Set the date to just before Midnight 12/31/2010:

```
DM-37x# date
DM-37x# date 2010.12.31-23:59:40
DM-37x# date
```

Verify the system date

Write the system time into the RTC and observe the transitions:

```
DM-37x# hwclock -w
DM-37x# while true; do hwclock -r; sleep 1; done
```

Verify the time transitions from pm to am

Verify the day increments to 01

Verify the month rolls over to 01

Verify the year increments to 2011

Ctl-Z to return to the DM-37x# prompt

16.01.04 Transition Feb 28 --> Feb 29, LEAP YEAR (2012) //Time transitions from pm to am, Day increments, Month & Year remains the same

[RTC-08-001] [RTC-08-002]

Set date to just before Midnight 2/28/2012:

```
DM-37x# date
DM-37x# date 2012.02.28-23:59:40
DM-37x# date
```

Verify the system date

Write the system time into the RTC and observe the transitions:

```
DM-37x# hwclock -w
DM-37x# while true; do hwclock -r; sleep 1; done
```

Verify the time transitions from pm to am

Verify the day increments to 29

Verify the month remains 02

Verify the year remains 2012

Ctl-Z to return to the DM-37x# prompt

16.01.05 Transition Feb 28 --> Mar 1, NON LEAP YEAR (2011) //Time transitions from pm to am, Day and Month increment, Year remains the same

[RTC-08-001] [RTC-08-002]

Set date to just before Midnight 2/28/2011:

```
DM-37x# date
DM-37x# date 2011.02.28-23:59:40
DM-37x# date
```

Verify the system date

Write the system time into the RTC and observe the transitions:

```
DM-37x# hwclock -w
DM-37x# while true; do hwclock -r; sleep 1; done
```

Verify the time transitions from pm to am

Verify the day increments to 01

Verify the month is March 03

Verify the year remains 2011

Ctl-Z to return to the DM-37x# prompt

16.02 Product ID Information

01.01.09 -

16.02.01 Physical verification of product ID info

[RTC-08-004]

Break into U-Boot and examine the product ID information output in the first few lines.

Verify DRAM matches the hardware configuration.

Verify NAND matches the hardware configuration.

Verify Board matches the hardware configuration.

Verify NOR matches the hardware configuration. (may say "already initialized in sync mode")

Verify Part Number matches the hardware configuration.

Verify Model Name matches the hardware configuration.

Verify Serial Number matches the hardware configuration.

16.02.02 Verification of software access to product ID info

[RTC-08-004]

POR

```
DM-37x# cd /sys/class/product_id/
```



```
DM-37x# ls
```

Verify "model_name", "part_number", serial_number" and "zone2_data" files are displayed.

NOTE: Torpedo+Wireless does not have zone2_data, (skip this for T+W).

View the model name:

```
DM-37x# cat model_name
```

Verify the model name applicable to the DUT is displayed and is correct.

View the part number:

```
DM-37x# cat part_number
```

Verify the part number applicable to the DUT is displayed and is correct.

View the serial number:

```
DM-37x# cat serial_number
```

Verify the serial number applicable to the DUT is displayed and is correct.

NOTE: Torpedo+Wireless does not have zone2_data, (skip this for T+W).

View the zone2 data:

```
DM-37x# cat zone2_data
```

Verify the NOR entry applicable to the DUT is displayed and correct.

Verify the NAND entry applicable to the DUT is displayed and correct.

Verify the SDRAM entry applicable to the DUT is displayed and correct.

Guide to interpreting zone2_data:

nor: XXYY nand: ZZAA sdram: BBCC

processor: 0000 platform_bits: 00000001 hardware_revision 00000001

where:

XX is the size of NOR0 flash (power of two in hex)

YY is the size of NOR1 flash (power of two in hex)

ZZ is the size of NAND0 flash (power of two in hex)

AA is the size of NAND1 flash (power of two in hex)

BB is the size of DRAM0 (power of two in hex)

CC is the size of DRAM1 (power of two in hex)

If a field is 0x00 or 0xff, then none populated.

EXAMPLE:

for the 1015061 zone2_data looks like:

nor: ffff nand: 1dff sdram: 1cff

processor: 0000 platform_bits: 00000001 hardware_revision 00000001

In this case, no NOR, NAND0=536870912, no NAND1, DRAM0=268435456, no DRAM1.

16.03 GPIO Access

16.03.01 GPIO as input

[RTC-08-006]

Boot image and login

For DM3730/AM3703 Torpedo SOM and DM3730/AM3703 Torpedo + Wireless SOM: (use S4 button)

```
DM-37x# mkdir /debug
DM-37x# mount -t debugfs debug /debug
DM-37x# echo 0x11c > /debug/omap_mux/sys_boot5
DM-37x# echo 7 > /sys/class/gpio/export
DM-37x# echo in > /sys/class/gpio/gpio7/direction
DM-37x# cat /sys/class/gpio/gpio7/value
```

For SOMLV: (use S4 button)

```
DM-37x# mkdir /debug
DM-37x# mount -t debugfs debug /debug
DM-37x# echo 0x11c > /debug/omap_mux/cam_xclkb
DM-37x# echo 111 > /sys/class/gpio/export
DM-37x# echo in > /sys/class/gpio/gpio111/direction
DM-37x# cat /sys/class/gpio/gpio111/value
```

Push the button and while holding it down repeat the "cat /sys/class/gpio/gpio7/value" command.

Verify the value changes from 1 to 0.

POR

16.03.02 GPIO as output

[RTC-08-006]

Connect the SDK breakout board (n/a on Torpedo & Torpedo+Wireless)

Boot image and login

```
DM-37x# mkdir /debug
DM-37x# mount -t debugfs debug /debug
DM-37x# echo 0x11c > /debug/omap_mux/cam_wen
DM-37x# echo 167 > /sys/class/gpio/export
```

Verify the following is displayed: direction edge power subsystem uevent value.

```
DM-37x# ls /sys/class/gpio/gpio167
DM-37x# echo out > /sys/class/gpio/gpio167/direction
DM-37x# echo 0 > /sys/class/gpio/gpio167/value
```

Verify J18 pin 64 to GND on the expansion board is low (0.0)


```
DM-37x# echo 1 > /sys/class/gpio/gpio167/value
```

Verify J18 pin 64 to GND on the expansion board is high (~1.8).

POR

16.05 Reboot Linux using 'reboot' command

16.05.01 Reboot

<Test Removed>

16.06 Supported Hardware

16.06.01 The BSP shall support all DM3730/AM3703 hardware modules as specified in the PRS as of date of release.

[HW-08-001]

Verify that the model numbers listed in the "Model Number Information" section of the DM3730/AM3703 Torpedo + Wireless SOM PRS are ALL currently being tested as a part of this test run.

16.06.02 Booting between Linux and Android [NONE]

<Test Removed>

16.06.03 Upgrading to Linux BSP Under Test (SUT)

<Test Removed>

16.07 LTIB builds (using Virtual Machine)

<Builds are now covered in section 18>

16.08 Temperature Sensor

16.08.01 Temperature sensor

[TEMPSENS-16-001]

POR

Boot the SOM and login to Linux.

Wait at least one full minute for the temperature to stabilize after booting.

Get the value of the temperature sensor as a Celsius value:

```
DM-37x# cat /sys/devices/platform/temp_sensor/celsius
```

Get the value of the temperature sensor as a raw value:

```
DM-37x# cat /sys/devices/platform/temp_sensor/raw
```

Write down the values in the temperature sensor for comparison.

Run the df_dok and df_andi applications at the same time to stress the system and raise the temperature:

```
DM-37x# df_andi &
DM-37x# df_dok
```

Hit ctrl-c to exit the df_dok application when it is finished.

Get the value of the temperature sensor as a Celsius value:

```
DM-37x# cat /sys/devices/platform/temp_sensor/celsius
```

Get the value of the temperature sensor as a raw value:

```
DM-37x# cat /sys/devices/platform/temp_sensor/raw
```

Verify that the temperature values in terms of Celsius and raw data have increased.

Using freeze spray to cool the SOM. Be careful not to cause condensation on the SOM. (if freeze spray is not available, you can blow on the SOM to try to cool it down).

Get the value of the temperature sensor as a Celsius value:

```
DM-37x# cat /sys/devices/platform/temp_sensor/celsius
```

Get the value of the temperature sensor as a raw value:

```
DM-37x# cat /sys/devices/platform/temp_sensor/raw
```

Verify that the temperature values in terms of Celsius and raw data have decreased.

16.09 Verify FCC versions of .ini and .bts are included in latest Linux releases

16.09.01 .bts file

[DM37LINUX-1130]

POR

Boot the SOM and login to Linux.

Move to the root directory:

```
DM-37x# cd /
```

Find the name and location of the Bluetooth .bts file:

```
DM-37x# find -name *.bts
```


Get the md5 sum of the Bluetooth .bts file (Logic_TIInit_xx.xx.xx.bts file):

```
DM-37x# md5sum <location/name of .bts file>
```

Compare the md5sum of the .bts file in the SOM to the .bts file in the SVN location provided above (BT folder):

Verify that the md5sums match.

For convenience, the md5sum of the .bts file on the SOM should be:

SOM	MD5SUM
SOMDM3730-11-xxxxxxxx-x	30b1d15c3c09d4cfb9b017a0b681c0dc
SOMDM3730-31-xxxxxxxx-x	8d84eefba73cf513428fdfc3aef6d901
SOMDM3730-32-xxxxxxxx-x	fef86b7088f2167538c70c85c21f38bf

16.09.02 .ini file – top half of NVS file

Verify that the top half of the NVS file is correct by verifying that the MAC address is a MAC address. This should be done in test 12.02.01.

16.09.03 .ini file - lower half

[DM37LINUX-1130]

NOTE: This test is to be performed on Torpedo+Wireless SOMs only.

Verify that the md5sum of the NVS file is correct. This verifies that the lower half of the .ini file is correct.

1. POR
2. Copy target NVS files to /tmp directory:

   DM-37x# cp /lib/firmware/ti-connectivity/wl128x-nvs*.bin /tmp/.

3. Cut off the first 468 bytes (that is overlaid with the product ID data) of the new wl128x-nvs.bin file:

   DM-37x# dd if=/tmp/<target NVS file>.bin of=/tmp/<target NVS file>-ini.bin bs=1 skip=468

For SOMDM3730-30-xxxxxxxx-x:

```
DM-37x# md5sum /tmp/wl128x-nvs-tw30-ini.bin
```

d1e95be932dd721349472e7d83c74c36

For SOMDM3730-31-xxxxxxxx-x:

```
DM-37x# md5sum /tmp/wl128x-nvs-tw31-ini.bin
```

a2f37b855cc9743421061cf496564995

For SOMDM3730-32-xxxxxxxx-x:

```
DM-37x# md5sum /tmp/wl128x-nvs-tw32-ini.bin
```


5ef185f3cbe0599d06c9e3077fd082fa

These md5sums were gotten by doing an md5sum of the wl128x-nvs.bin file after performing the directions in Appendix D: Verification of NVS file to determine accuracy of lower half of NVS file (lower half of NVS = .ini file). It is the same on all boards.

If the base file in SVN changes in any way the md5sums will no longer match and you will need to follow the steps in Appendix D: Verification of NVS file

16.10 Verify that the bash "Shellshock" security bug is fixed

16.10.01 Verify that the bash "Shellshock" security bug is fixed

< Test Removed 09/28/2015 >

16.11 Battery testing

**NOTE: This battery section (16.11) is not applicable on SomLVs. It is only applicable on Torpedo and Torpedo+Wireless baseboards with the battery socket (J20).*

16.11.01 Linux Power Supply class

[BAT-13-001]

Boot the DM37x.

Do and 'ls' of the /sys/class/power folder:

```
DM-37x# ls /sys/class/power_supply
```

Verify that there is a folder named bq27000-battery

16.11.02 AC power charging support

[BAT-13-002] [BAT-13-005]

Power on the SOM and break into U-boot.

Make sure that disable charging is set to no, save the environment and boot:

```
OMAP Logic # disablecharging no; saveenv; boot
```

Log in at the prompt.

With a Battery plugged in and AC power CONNECTED (the flat black type in socket J20) read out the voltage of the battery and charging system:

```
DM-37x# cat /sys/class/power_supply/bq27000-battery/voltage_now
```

Verify, on the Torpedo baseboard, that the charging LED is lit (D7 on the baseboard, near the heartbeat LED and Power On LED).

Unplug the AC power to the board and run df_dok to put a drain on the battery

```
DM-37x# df_dok
```

Hit CTRL-C when df_dok finishes

With a Battery plugged in and AC power DISCONNECTED (the flat black type in socket J20) read out the voltage of the battery and charging system:

```
DM-37x# cat /sys/class/power_supply/bq27000-battery/voltage_now
```

Verify, on the Torpedo baseboard, that the charging LED is NOT lit (D7 on the baseboard, near the heartbeat LED and Power On LED).

Verify that the value returned from the cat command of voltage_now is LOWER than before.

Plug the AC power to the board.

Wait for about 10 seconds

With a Battery plugged in and AC power CONNECTED (the flat black type in socket J20) read out the voltage of the battery and charging system:

```
DM-37x# cat /sys/class/power_supply/bq27000-battery/voltage_now
```

Verify, on the Torpedo baseboard, that the charging LED is NOT lit (D7 on the baseboard, near the heartbeat LED and Power On LED).

Verify that the value returned from the cat command of voltage_now is HIGHER than before.

16.11.03 USB power charging support

[BAT-13-003] [BAT-13-005]

**NOTE: Since USB charging is more of a trickle charge, it can be hard to see changes in battery level when the battery is close to fully charged. It is recommended that you get the charge to < 3950000 for the discharge state for this test.*

Power on the SOM and break into U-boot.

Make sure that disablecharging is set to no, save the environment and boot:

```
OMAP Logic # disablecharging no; saveenv; boot
```

Log in at the prompt.

Enter the following commands to enable USB charging:

```
DM-37x# echo 4 > /debug/omap_mux/mcbsp1_fsx
DM-37x# echo 4 > /debug/omap_mux/mcbsp1_clkr
DM-37x# echo 156 > /sys/class/gpio/export
DM-37x# echo 161 > /sys/class/gpio/export
```



```
DM-37x# echo out > /sys/class/gpio/gpio156/direction
DM-37x# echo out > /sys/class/gpio/gpio161/direction
DM-37x# echo 1 > /sys/class/gpio/gpio156/value
DM-37x# echo 1 > /sys/class/gpio/gpio161/value
```

With a Battery plugged in and USB cable CONNECTED from your computer to the OTG port on the baseboard (the flat black type in socket J20) read out the voltage of the battery and charging system:

```
DM-37x# cat /sys/class/power_supply/bq27000-battery/voltage_now
```

Verify, on the Torpedo baseboard, that the charging LED is lit (D7 on the baseboard, near the heartbeat LED and Power On LED).

Unplug the USB power to the board and run df_dok to put a drain on the battery

```
DM-37x# df_dok
```

Hit CTRL-C when df_dok finishes

With a Battery plugged in and USB cable DISCONNECTED from your computer to the OTG port on the baseboard (the flat black type in socket J20) read out the voltage of the battery and charging system:

```
DM-37x# cat /sys/class/power_supply/bq27000-battery/voltage_now
```

Verify, on the Torpedo baseboard, that the charging LED is NOT lit (D7 on the baseboard, near the heartbeat LED and Power On LED).

Verify that the value returned from the cat command of voltage_now is LOWER than before.

Plug the USB power to the board.

Wait for about 10 seconds

With a Battery plugged in and USB cable CONNECTED from your computer to the OTG port on the baseboard (the flat black type in socket J20) read out the voltage of the battery and charging system:

```
DM-37x# cat /sys/class/power_supply/bq27000-battery/voltage_now
```

Verify, on the Torpedo baseboard, that the charging LED is NOT lit (D7 on the baseboard, near the heartbeat LED and Power On LED).

Verify that the value returned from the cat command of voltage_now is HIGHER than before.

5.17 Test 17 GPS

Purpose

This test verifies that GPS is functioning properly

Setup

Test Steps

Perform the following subtests and record results in the action log:

17.01 GPS

**Note: This test is applicable to the Torpedo + Wireless SOMs only.*

17.01.01 GPS

[GPS-13-001] [GPS-13-002]

Load / boot the GPS unit

```
DM37x# /etc/rc.d/init.d/wl1283-gps start
```

Or

(all three lines)

```
DM-37x# modprobe gps_drv
DM-37x# /home/root/wl12xx/uim&
DM-37x# navd --android_log NAVD  -p3 -nav\"-
c/system/etc/gps/config/pathconfigfile.txt\" &
```

After the setup finishes, run the gps using the following:

```
DM-37x# testgps
```

Verify that the gps unit is working. There should be flowing output something like below:

*$GPGLL,,,,,,V,N*64*

*$GPRMC,,V,,,,,,,,,N*53*

*$GPGGA,,,,,,0,,,,,,,,*66*

*$GPVTG,,T,,M,,N,,K,N*2C*

*$GPGSA,M,1,,,,,,,,,,,,,,*12*

*$GPGSV,1,1,00*79*

*Verify that the $GPGLL sentence has the coordinates of the MPLS office which is around 44ºN and 93ºW. Example: $GPGLL,4459.147440,N,09316.463388,W,161402.00,A,A*7A*

NOTE: It might take 10 minutes or more (especially the first time) to have the coordinates available.

17.01.02 GPS after suspend / resume

[GPS-13-004]

**Note: This test is applicable to the Torpedo + Wireless SOMs only.*

This test continues from test 17.01.01.

(Run test 17.01.01 if not already so)

Press S2 to suspend.

Press keyboard to resume.

Verify that the gps unit is still running. There should be flowing output something like below:

*$GPGLL,,,,,,V,N*64*

*$GPRMC,,V,,,,,,,,,,N*53*

*$GPGGA,,,,,,0,,,,,,,,*66*

*$GPVTG,,T,,M,,N,,K,N*2C*

*$GPGSA,M,1,,,,,,,,,,,,,,,*12*

*$GPGSV,1,1,00*79*

*Verify that the $GPGLL sentence has the coordinates of the MPLS office which is around 44ºN and 93ºW. Example: $GPGLL,**4459.147440,N,09316.463388,W**,161402.00,A,A*7A*

17.02 GPS Stress Test

17.02.01 GPS 48 Hour Test

<Test Removed>

5.18 Test 18 LTIB Builds

Purpose

This test verifies the various types of builds that are made from LTIB.

Setup

Test Steps

Perform the following subtests and record results in the action log:

**NOTE: Builds should be done on a machine that has already had at least one successful build completed on it (except for virtual box builds). If your machine has not, go to the wiki and look at the "DM37linux" page at the section 5 (How to Setup LTIB to Build on your Linux Machine) and section 6 (First build of LTIB on your Linux Machine) or section 7 (First build of LTIB for linux 3.0 kernel).*

18.01 Virtual Box Build (regular build)

18.01.01 LTIB Build on Virtual Box

<Test Removed>

18.02 GTK+ Build

18.02.01 LTIB Build on Virtual Box (GTK+ version)

<Test Removed>

18.03 DSP Build

18.03.01 DVDSK (DSP) Build

[DSP-11-001]

**NOTE*: If you ran any other build previous to this, you will want to unselect the options that you selected for that build when you select your options for this build.*

Go to the Beacon EmbeddedWorks wiki page "DM37linux". Look at section 16 (Enable the DSP). If you have not done a build already, sections 5 (how to setup a build) & 7 First build of LTIB for linux 3.0).

Using one of the links provided, download the installer for Code Generation Tools v6.1.14. You'll have to register with TI to get it.

Change the permissions of the downloaded code:

```
logic@logic-desktop:~$ chmod a+x ~/Downloads/ti_cgt_c6000_6.1.14_setup_linux_x86.bin
```

Run the downloaded code:

```
logic@logic-desktop:~$ sudo ~/Downloads/ti_cgt_c6000_6.1.14_setup_linux_x86.bin
```

**Note: Do the typical install setup. DO NOT set the environmental variables that it suggests at the end of the install / setup process.*

It is suggested that you make the screen full size as trying to change the size after the GUI for LTIB starts may cause the GUI to look funky and unreadable.

Run LTIB

```
logic@logic-desktop/<location of ltib folder>:~$ ./ltib -c
```

Before setting up for DSDVK, we need to setup to make the build a yaffs based build. The DVDSK footprint is far too large to run from RAM, so we must boot and run from NAND, the SD card or NFS. We will run from NAND using a YAFFS based file system.

Arrow down to "Options --->" on the line under "Target Image Generation"

On the second line "Target image: (ext2.gz ramdisk) --->" hit enter and in the next menu, select [ ]yaffs2.

Arrow right and exit the image selection screen. You should now be on the original selection screen that you got after running./ltib -c

Scroll down to "Package list --->" and hit enter

Scroll down to "TI DVSDK Packages --->" and hit enter

Select all available DVDSK packages

Press the right arrow to highlight "< Exit >" and hit enter.

Scroll down and enable the "gstreamer-plugins-good", "gstreamer-plugins-bad", "gstreamer-plugins-ugly", and "gstreamer-plugins-ffmpeg" packages.

Scroll down and enable the "mediactl", and "yavta" packages. (NOTE: These packages are needed for Camera use, and not required for DSP build.)

Once your selections are complete, keep exiting until you're asked "Do you want to save your new configuration", then highlight "< Yes >" and hit enter.

After the build finishes (successfully) create the SD boot card with the new image by formatting a SD card with the HP formatting tool, and copying the following files onto the card (MLO must be first):

MLO (x-load.bin.ift) in <ltib folder>/rootfs/boot

u-boot.bin in <ltib folder>/rootfs/boot

u-boot.bin.ift in <ltib folder>/rootfs/boot

uImage in <ltib folder>/rootfs/boot

rootfs.yaffs2 in <ltib folder>

Put the SD card into the DUT and power on.

Break into U-Boot and run the yaffs build setup:

```
OMAP Logic # run makeyaffsboot
```

makeyaffsboot will take a couple of minutes setting your SUT up for running the yaffs based file system. When it is complete, power cycle the board (POR).

Break into U-Boot and set the otherbootargs environmental variable:

```
OMAP Logic # setenv otherbootargs 'ignore_loglevel early_printk
no_console_suspend mem=55M@0x80000000 mem=128M@0x88000000'
OMAP Logic # saveenv
```

POR

Boot and login

Verify that the board booted and you were able to login

Change directories to a DSP specific directory

```
DM-37x# cd /usr/share/ti/ti-dsplink-examples
```

Verify that the directory exists (you were able to move to it).

**NOTE: USE THIS IMAGE TO PERFORM THE TESTS IN SECTION 14 – DSP TESTS, AND SECTION 20 – CAMERA TESTS*

18.03.02 DVDSK (DSP) Build (2001 toolchain)

[DSP-11-001]

On the build machine, in the LTIB folder, remove rootfs, rpm/BUILD and rpm/RPMS:

```
logic@logic-Virtualbox:<LTIB folder> rm -rf rpm/BUILD
logic@logic-Virtualbox:<LTIB folder> rm -rf rpm/RPMS
logic@logic-Virtualbox:<LTIB folder> sudo rm -rf rootfs*
```

Follow the directions in 18.03.01 except that in ./ltib -c scroll down to Toolchain selection and select the CodeSourcery-2011-.09.70 toolchain.

Verify that the build is successful and passes based on the steps in 18.03.01 DVDSK (DSP) Build

18.04 Qtopia Build

18.04.01 Qtopia Build

[DISP-06-007] [UI-13-001]

**NOTE*: If you ran any other build previous to this, you will want to unselect the options that you selected for that build when you select your options for this build.*

It is suggested that you make the screen full size as trying to change the size after the GUI for LTIB starts will cause the GUI to look funky and unreadable.

Setup the build computer to do a Qtopia build:

```
logic@logic-Virtualbox:<LTIB folder> ./ltib -p qtopia-4.spec -m prep
```

Run LTIB

```
logic@logic-Virtualbox:<LTIB folder> ./ltib -c
```

Before setting up for Qtopia, we need to setup to make the build a yaffs based build. The Qtopia footprint is far too large to run from RAM, so we must boot and run from NAND, the SD card or NFS. We will run from NAND using a YAFFS based file system.

Arrow down to “Options --->” on the line under “Target Image Generation”

On the second line “Target image: (ext2.gz ramdisk) --->” hit enter and in the next menu, select [ ]yaffs2.

Arrow right and exit the image selection screen. You should now be on the original selection screen that you got after running ./ltib -c

Scroll down to "Package list --->" and hit enter

Scroll down to "Qtopia 4.7.4--->" and hit space

Highlight "< Exit >" and hit enter. Keep "exiting" until you're asked "Do you want to save your new configuration", then highlight "< Yes >" and hit enter.

At this point the build should kick off. It could take up to an hour or more for it to complete depending on the speed of your machine.

When the build completes, create an SD boot card by using a freshly formatted card and putting onto it the following files.

Alternatively, you can use the script mkLogicFATcard.sh to create the SD boot card:

```
logic@logic-Virtualbox:<LTIB folder> ./bin/mkLogicFATcard.sh -cy
```

.

MLO (x-load.bin.ift) in <ltib folder>/rootfs/boot

u-boot.bin
in <ltib folder>/rootfs/boot

u-boot.bin.ift
in <ltib folder>/rootfs/boot

uImage
in <ltib folder>/rootfs/boot

rootfs.yaffs2
in <ltib folder>

Put the SD card into the DUT and power on.

Break into U-Boot and run the yaffs build setup:

```
OMAP Logic #  run makeyaffsboot
```

makeyaffsboot will take a couple of minutes setting your SUT up for running the yaffs based file system. When it is complete, power cycle the board (POR).

Boot and login

Verify that the board booted and you were able to login

Kickoff the Qtopia 4.74 desktop demo

```
DM-37x# cd /usr/local/qt
DM-37x# export QTDIR=$PWD
DM-37x# export LD_LIBRARY_PATH=$QTDIR/lib:$LD_LIBRARY_PATH
DM-37x# demos/mainwindow/mainwindow -qws
```

Verify that Qtopia runs and you get to the Qtopia desktop demo.

18.05 Benchmark Build

18.05.01 BenchMark Build

<Follow the directions in Appendix E.>

18.06 Generic Build Tests

18.06.01 LTIB build shared libraries correctly

< *Test Removed 09/28/2015* >

18.06.02 LTIB config issues

< *Test Removed 09/28/2015* >

18.07 Backport tests

18.07.01 Backports

< Test Removed 09/28/2015 >

5.19 Test 19 Beacon EmbeddedWorks U-Boot Commands

Purpose
This test verifies that Beacon EmbeddedWorks's custom U-Boot commands are functioning properly

Setup
Test Steps
Perform the following subtests and record results in the action log:

<Test Section 19 removed (19.01.01 – 19.01.12 and 19.02.01 – 19.02.10)>

5.20 Test 20 Camera Tests

Purpose

This test verifies the functionality of previewing on LCD, capturing and displaying a JPEG image, capturing and playing an H.264 or MJPEG video for the following cameras:

Setup

Testing the Logitech C210 USB webcam requires the DSP build from section 18.03. When running this build, remember to adjust the memory location appropriately in U-Boot:

```
DM37# setenv otherbootargs 'ignore_loglevel early_printk no_console_suspend
mem=55M@0x80000000 mem=128M@0x88000000'

DM37x# saveenv
```

Camera's tested:

- Leopard Imaging LI-5M04
- Logitech C210 USB webcam

Testing the Leopard Imaging LI-5M04 camara requires a specific build, hereafter known as "Camera build." Building a camera build is the same as building the DSP build in section 18.03, except that the following 2 packages must also be included: yavta package and media-ctl package. It's also necessary to adjust the memory location as described in the above paragraph. This Camera build can be used to test the Logitech camera as well.

When testing the Leopard LI-5M04 camera, a jumper must be on pins 1 and 2 of JP5 to enable 8-bit video data bus. This camera requires cpufreq of 800 MHz or faster. Therefore, cpufreq must be adjustable to 800 MHz when running with 'ondemand' governor, or set to at least 800 MHz when running with 'userspace' governor.

**NOTE: IN ALL VIDEO PLAYBACK SITUATIONS, THE PLAYBACK SHOULD BE PERFORMED 5 TIMES TO CATCH INTERMITTENT ISSUES.*

Test Steps

Perform the following subtests and record results in the action log:

20.01 Leopard Imaging LI-5M04 Tests

20.01.01 Preview, Capture and Play Image and Video

[CAM-13-001] [CAM-13-004] [CAM-13-006]

**NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.*

Connect the camera to J6. (This connector is available on Torpedo Baseboard 1021669. It's NOT available on Torpedo Baseboard 1017984, and NOT on SOM-LV Baseboard.)

Load the Camera image.

POR (Must power on with the camera plugged in to J6.)

(Note: make sure the cpufreq is set to or can be adjusted to 800 MHz, and a jumper is on pin 1 and 2 of JP5 .)

Prepare the system for using the LI-5M04 camera. These commands disable the virtual console screen timeout, set the exposure and gain on the camera, and setup the video input hardware pipeline:

Make sure tux is displayed on the screen. If not, cycle power the DUT.

```
DM-37x# echo -ne "\033[9;0]\033[?25l" > /dev/tty0
DM-37x# yavta -w '0x00980911 720' /dev/v4l-subdev8
DM-37x# yavta -w '0x00980913 16' /dev/v4l-subdev8
DM-37x# yavta -w '0x0098090e 125' /dev/v4l-subdev8
DM-37x# yavta -w '0x0098090f 175' /dev/v4l-subdev8
DM-37x# media-ctl -v -r -l '"mt9p031":0->"OMAP3 ISP CCDC":0[1], "OMAP3 ISP CCDC":2->"OMAP3 ISP preview":0[1], "OMAP3 ISP preview":1->"OMAP3 ISP resizer":0[1], "OMAP3 ISP resizer":1->"OMAP3 ISP resizer output":0[1]'
```

Preview the LI-5M04 camera on the LCD:

```
DM-37x# media-ctl -v -f '"mt9p031":0 [SGRBG8 1298x970 (664,541)/1298x970], "OMAP3 ISP CCDC":2 [SGRBG10 1298x970], "OMAP3 ISP preview":1 [UYVY 1298x970], "OMAP3 ISP resizer":1 [UYVY 640x480]'

DM-37x# gst-launch v4l2src device=/dev/video6 num-buffers=150 always-copy=false queue-size=4 ! 'video/x-raw-yuv,format=(fourcc)UYVY,width=640,height=480,framerate=30/1' ! queue ! tidisplaysink2 mmap-buffer=true
```

Move the camera around and view the image on LCD.

Verify that the image displayed on the LCD is what the camera is pointed at.

Capture a JPEG image from the LI-5M04 camera and display it:

```
DM-37x# media-ctl -v -f '"mt9p031":0 [SGRBG8 2610x1954 (7,49)/2610x1954], "OMAP3 ISP CCDC":2 [SGRBG10 2610x1954], "OMAP3 ISP preview":1 [UYVY 2610x1954], "OMAP3 ISP resizer":1 [UYVY 2592x1944]'

DM-37x# gst-launch v4l2src device=/dev/video6 num-buffers=1 ! 'video/x-raw-yuv,format=(fourcc)UYVY,width=2592,height=1944' ! TIImgenc1 engineName=codecServer codecName=jpegenc resolution=2592x1944 iColorSpace=UYVY oColorSpace=YUV420P qValue=97 ! filesink location=still.jpg

DM-37x# gst-launch filesrc location=still.jpg ! jpegdec ! ffmpegcolorspace ! videoscale ! 'video/x-raw-rgb,width=320,height=240' ! fbdevsink device=/dev/fb0
```

Verify that the image displayed on the LCD is the image taken by the camera.

Capture an H.264 video from the LI-5M04 camera and play it back:

```
DM-37x# media-ctl -v -f '"mt9p031":0 [SGRBG8 1298x970 (664,541)/1298x970], "OMAP3 ISP CCDC":2 [SGRBG10 1298x970], "OMAP3 ISP preview":1 [UYVY 1298x970], "OMAP3 ISP resizer":1 [UYVY 640x480]'
```



```
DM-37x# gst-launch v4l2src device=/dev/video6 num-buffers=300 always-copy=false queue-
size=4 ! 'video/x-raw-yuv,format=(fourcc)UYVY,width=640,height=480,framerate=24/1' !
TIPrepEncBuf numOutputBufs=4 contiguousInputFrame=false ! tee name=tee tee. ! queue !
tidisplaysink2 mmap-buffer=true tee. ! queue ! TIVidenc1 engineName=codecServer
codecName=h264enc contiguousInputFrame=true ! queue ! avimux ! filesink
blocksize=65536 location=video640x480.avi
```

```
DM-37x# gst-launch filesrc location=video640x480.avi ! avidemux ! TIViddec2
engineName=codecServer codecName=h264dec ! queue ! tidisplaysink2
```

Move the camera around as it is recording.

Verify that the video displayed on the LCD is the video recorded by the camera.

20.01.02 LI-5M04 Camera Performance after Suspend/Resume

[CAM-13-001][CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

Connect the camera to J6. (This connector is available on Torpedo Baseboard 1021669. It's NOT available on Torpedo Baseboard 1017984, and NOT on SOM-LV Baseboard.)

The following section in { } can be skipped if this test is performed immediately after test 20.01.01:

{

Load the Camera image.

POR (Must power on with the camera plugged in to J6)

```
DM-37x# echo -ne "\033[9;0]\033[?25l" > /dev/tty0
DM-37x# yavta -w '0x00980911 720' /dev/v4l-subdev8
DM-37x# yavta -w '0x00980913 16' /dev/v4l-subdev8
DM-37x# yavta -w '0x0098090e 125' /dev/v4l-subdev8
DM-37x# yavta -w '0x0098090f 175' /dev/v4l-subdev8
```

```
DM-37x# media-ctl -v -r -l '"mt9p031":0->"OMAP3 ISP CCDC":0[1], "OMAP3 ISP CCDC":2-
>"OMAP3 ISP preview":0[1], "OMAP3 ISP preview":1->"OMAP3 ISP resizer":0[1], "OMAP3 ISP
resizer":1->"OMAP3 ISP resizer output":0[1]'
```

```
DM-37x# media-ctl -v -f '"mt9p031":0 [SGRBG8 1298x970 (664,541)/1298x970], "OMAP3 ISP
CCDC":2 [SGRBG10 1298x970], "OMAP3 ISP preview":1 [UYVY 1298x970], "OMAP3 ISP
resizer":1 [UYVY 640x480]'
```

```
DM-37x# gst-launch v4l2src device=/dev/video6 num-buffers=300 always-copy=false queue-
size=4 ! 'video/x-raw-yuv,format=(fourcc)UYVY,width=640,height=480,framerate=24/1' !
TIPrepEncBuf numOutputBufs=4 contiguousInputFrame=false ! tee name=tee tee. ! queue !
tidisplaysink2 mmap-buffer=true tee. ! queue ! TIVidenc1 engineName=codecServer
codecName=h264enc contiguousInputFrame=true ! queue ! avimux ! filesink
blocksize=65536 location=video640x480.avi
```



```
DM-37x# gst-launch filesrc location=video640x480.avi ! avidemux ! TIViddec2
engineName=codecServer codecName=h264dec ! queue ! tidisplaysink2
```

Move the camera around as it's recording.

Verify that the video displayed on the LCD is the video recorded by the camera.

}

Press S2 button or type "echo mem > /sys/power/state" to suspend.

Verify the DUT enters Suspend mode.

Touch the LCD, press a key on the keyboard, or press S2 to resume.

Verify the DUT resumes from Suspend mode.

Capture an H.264 video from the LI-5M04 camera and play it back:

```
DM-37x# media-ctl -v -f '"mt9p031":0 [SGRBG8 1298x970 (664,541)/1298x970], "OMAP3 ISP
CCDC":2 [SGRBG10 1298x970], "OMAP3 ISP preview":1 [UYVY 1298x970], "OMAP3 ISP
resizer":1 [UYVY 640x480]'
```

```
DM-37x# gst-launch v4l2src device=/dev/video6 num-buffers=300 always-copy=false queue-
size=4 ! 'video/x-raw-yuv,format=(fourcc)UYVY,width=640,height=480,framerate=24/1' !
TIPrepEncBuf numOutputBufs=4 contiguousInputFrame=false ! tee name=tee tee. ! queue !
tidisplaysink2 mmap-buffer=true tee. ! queue ! TIVidenc1 engineName=codecServer
codecName=h264enc contiguousInputFrame=true ! queue ! avimux ! filesink
blocksize=65536 location=video640x480.avi
```

```
DM-37x# gst-launch filesrc location=video640x480.avi ! avidemux ! TIViddec2
engineName=codecServer codecName=h264dec ! queue ! tidisplaysink2
```

Move the camera around as it is recording.

Verify that the video displayed on the LCD is the video recorded by the camera.

20.02 Logitech C210 USB Webcam Tests

20.02.01 Preview, Capture and Play Image and Video

[CAM-13-001] [CAM-13-004] [CAM-13-006]

** NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.*

Load the DSP or Camera image.

POR (Must plug in the camera after power on.)

On the Torpedo baseboard, connect the Logitech C210 USB webcam to a powered hub, and connect the powered hub to the USB OTG port.

On the SOM-LV baseboard, connect the Logitech C210 USB webcam to the USB port.

(Note: make sure the cpufreq is set to or can be adjusted to 800 MHz)

Prepare the system for using the C210 USB webcam. These commands disable the virtual console screen timeout.

```
DM-37x# echo -ne "\033[9;0]\033[?25l" > /dev/tty0
```

Preview the C210 USB webcam on the LCD.

```
DM-37x# gst-launch v4l2src num-buffers=90 device=/dev/video9 !
'image/jpeg,width=320,height=240,framerate=30/1' ! jpegdec ! ffmpegcolorspace !
tidisplaysink2
```

Move the camera around and view the image on LCD.

Verify that the image displayed on the LCD is what the camera is pointed at.

Capture a JPEG image from the C210 USB webcam and display it.

```
DM-37x# gst-launch v4l2src num-buffers=1 device=/dev/video9 !
'image/jpeg,width=640,height=480,framerate=15/1' ! jpegdec ! jpegenc ! filesink
location=usbstill.jpg
```

```
DM-37x# gst-launch filesrc location=usbstill.jpg ! jpegdec ! ffmpegcolorspace !
videoscale ! 'video/x-raw-rgb,width=320,height=240' ! fbdevsink device=/dev/fb0
```

Verify that the image displayed on the LCD is the image taken by the camera.

Capture an MJPEG video from the C210 USB webcam and play it back.

```
DM-37x# gst-launch v4l2src num-buffers=90 device=/dev/video9 !
'image/jpeg,width=640,height=480,framerate=15/1' ! queue ! avimux ! filesink
blocksize=65536 location=usbvideo.avi
```

```
DM-37x# gst-launch filesrc location=usbvideo.avi ! avidemux ! jpegdec !
ffmpegcolorspace ! tidisplaysink2
```

Move the camera around as it is recording.

Verify that the video displayed on the LCD is the video recorded by the camera.

20.02.02 Logitech C210 USB Webcam Performance after Suspend/Resume

[CAM-13-001] [CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

The following section in { } can be skipped if this test is performed immediately after test 20.02.01:

{

POR (Must plug in the camera after power on.)

On the Torpedo baseboard, connect the Logitech C210 USB webcam to a powered hub, and connect the powered hub to the USB OTG port.

On the SOM-LV baseboard, connect the Logitech C210 USB webcam to the USB port.


```
DM-37x# echo -ne "\033[9;0]\033[?25l" > /dev/tty0
```

```
DM-37x# gst-launch v4l2src num-buffers=90 device=/dev/video9 !
'image/jpeg,width=640,height=480,framerate=15/1' ! queue ! avimux ! filesink
blocksize=65536 location=usbvideo.avi
```

```
DM-37x# gst-launch filesrc location=usbvideo.avi ! avidemux ! jpegdec !
ffmpegcolorspace ! tidisplaysink2
```

Move the camera around as it's recording.

Verify that the video displayed on the LCD is the video recorded by the camera.

}

Press S2 button or type "echo mem > /sys/power/state" to suspend.

Verify the DUT enters Suspend mode.

Touch the LCD, press a key on the keyboard, or press S2 to resume.

Verify the DUT resumes from Suspend mode.

Capture an MJPEG video from the C210 USB webcam and play it back.

```
DM-37x# gst-launch v4l2src num-buffers=90 device=/dev/video9 !
'image/jpeg,width=640,height=480,framerate=15/1' ! queue ! avimux ! filesink
blocksize=65536 location=usbvideo.avi
```

```
DM-37x# gst-launch filesrc location=usbvideo.avi ! avidemux ! jpegdec !
ffmpegcolorspace ! tidisplaysink2
```

Move the camera around as it is recording.

Verify that the video displayed on the LCD is the video recorded by the camera.

20.03 Leopard Imaging LI-5M04 Tests (2011 toolchain testing)

20.03.01 Preview, Capture and Play Image and Video

[CAM-13-001] [CAM-13-004] [CAM-13-006]

** NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.*

Perform test 20.01.01 but with the 2011 toolchain (18.03.02 build).

Verify that the testing was successful.

20.03.02 Preview, Capture and Play Image and Video after Suspend/Resume (2011 toolchain testing)

[CAM-13-001] [CAM-13-004] [CAM-13-006]

** NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.*

Perform test 20.01.02 but with the 2011 toolchain (18.03.02 build).

Verify that the testing was successful.

01.01.10 -

20.04 Logitech C210 USB Webcam Tests (toolchain 2011 testing)

20.04.01 Preview, Capture and Play Image and Video (2011 toolchain testing)

[CAM-13-001] [CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

Perform test 20.02.0 but with the 2011 toolchain (18.03.02 build).

Verify that the testing was successful.

20.04.02 Preview, Capture and Play Image and Video after Suspend/Resume (2011 toolchain testing)

[CAM-13-001] [CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

Perform test 20.02.02 but with the 2011 toolchain (18.03.02 build).

Verify that the testing was successful.

01.01.11 -

20.05 Leopard Imaging LI-5M04 Tests (s - video testing)

20.05.01 Preview, Capture and Play Image and Video

[CAM-13-001] [CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

Perform test 20.01.01 but with the following additions:

In uboot:

```
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapdss.def_disp=tv
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapfb.mode=tv:ntsc
```


After booting:

```
DM-37x# peekpoke -l -w 0x48050cc4 0xd
DM-37x# peekpoke -l -w 0x48050040 0x78
```

Verify that the testing was successful.

20.05.02 Preview, Capture and Play Image and Video after Suspend/Resume (s - video testing)

[CAM-13-001] [CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

Perform test 20.01.02 but with the following additions:

In uboot:

```
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapdss.def_disp=tv
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapfb.mode=tv:ntsc
```

After booting:

```
DM-37x# peekpoke -l -w 0x48050cc4 0xd
DM-37x# peekpoke -l -w 0x48050040 0x78
```

Verify that the testing was successful.

20.06 Logitech C210 USB Webcam Tests – S Video

20.06.01 Preview, Capture and Play Image and Video (s - video testing)

[CAM-13-001] [CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

Perform test 20.02.01 but with the following additions:

In uboot:

```
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapdss.def_disp=tv
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapfb.mode=tv:ntsc
```

After booting:

```
DM-37x# peekpoke -l -w 0x48050cc4 0xd
DM-37x# peekpoke -l -w 0x48050040 0x78
```

Verify that the testing was successful.

20.06.02 Preview, Capture and Play Image and Video after Suspend/Resume (S Video testing)

[CAM-13-001] [CAM-13-004] [CAM-13-006]

* NOTE: This test requires SOMs with DSP, therefore is not applicable to AM3703 SOMs.

Perform test 20.02.02 but with the following additions:

In uboot:

```
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapdss.def_disp=tv
OMAP_LOGIC # setenv otherbootargs ${otherbootargs} omapfb.mode=tv:ntsc
```

After booting:

```
DM-37x# peekpoke -l -w 0x48050cc4 0xd
DM-37x# peekpoke -l -w 0x48050040 0x78
```

Verify that the testing was successful.

5.21 Test 21 Performance

Purpose

This test verifies that the non-functional requirements for performance are met. Included are performance, power and documentation requirements.

Setup

Test Steps

Perform the following subtests and record results in the action log:

21.01 FPU

21.01.01 NEON + FPU testing

[PERF-08-001] [PERF-11-017] [PERF-16-018]

Load the Linux image under test.

```
DM-37x# test-neon
```

Verify that the Linux BSP supports the NEON coprocessor via the following output:

Simple NEON test program to add an integer(5) to a 16 value vector

original vector = 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16

data (+5) = 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21

```
DM-37x# speedtest-neon
```

Verify that the Neon FPU performance shall be at least 35mflops for 100M addition and 100M multiplications and at least 9 mflops for 1000M divisions.

```
DM-37x# speedtest-vfpv3
```

Verify that the Neon FPU performance shall be at least 12 mflops for 100M additions and 100M multiplications and at least 9mflops for 100M divisions on an 800MHz processor.

Enter the output of the tests on the PERF tab of the TPL

21.02 Boot Time

21.02.01 Boot time

[PERF-13-010] [PERF-13-011] [PERF-13-012] [DISP-12-012]

This test is to track the boot times of the DM37x Linux 2.x. You will need a stopwatch (or some other way to track time) for this test.

Load the Linux image and start the stopwatch (linux must be tested with a SD boot card).

Verify that the splash screen is present on the LCD screen within 3 seconds of boot.

Verify that the Linux Penguin (Tux) is present on the LCD screen within 18 seconds of boot

Verify that the login prompt is displayed on the terminal interface within 31 seconds of boot

Enter the load time on the performance sheet of the TPL

21.03 Power Usage per State

21.03.01 Power Usage, Run / Idle State, Standard demo image

[PERF-08-004] [PERF-08-008] [POWER-12-001]

**WARNING!!!! THE JUMPER JP6 ON THE SDK2 BASEBOARD MUST BE IN THE 2-3 DGND POSITION AND NOT THE 1-2 5V POSITION WHILE USING WATTSON OR THE READINGS WILL BE INCORRECT"*

Load the Linux image under test.

Login

Using a USB A to USB mini-b cable, connect the SUT (at the PWR MEAS port) to the workstation.

Start the Wattson Power Measurement Tool on the workstation

Using Wattson on the workstation, determine the power consumption of the SUT while running the standard demo image. Do this by logging the power consumption for 10 seconds and taking the average.

Verify that the power consumption (average) of the SUT running the standard demo image is less than 3W.

Enter the average power consumption on the performance sheet of the TPL

21.03.02 Power Usage, Suspend state via "echo mem"

[PERF-08-005] [PERF-08-008] [PERF-13-017] [POWER-12-001] [POWER-12-005]

**WARNING!!!! THE JUMPER JP6 ON THE SDK2 BASEBOARD MUST BE IN THE 2-3 DGND POSITION AND NOT THE 1-2 5V POSITION WHILE USING WATTSON OR THE READINGS WILL BE INCORRECT"*

Load the Linux image under test.

Login

Put the SUT into a suspend state


```
DM3730Logic# echo mem > /sys/power/state
```

With the device in suspend mode, take power measurements of the SOM and the KIT with the Wattson Power Measurement Tool.

Verify that the power consumption of the SOM ONLY in suspend mode is less than 10 mW.

Verify that the power consumption of the KIT in suspend mode is less than 30 mW.

Enter the average power consumption on the performance sheet of the TPL

21.03.03 Power Usage, Suspend state via S2 button

[PERF-08-005] [PERF-08-008] [PERF-13-017] [POWER-12-001] [POWER-12-005]

**WARNING!!!! THE JUMPER JP6 ON THE SDK2 BASEBOARD MUST BE IN THE 2-3 DGND POSITION AND NOT THE 1-2 5V POSITION WHILE USING WATTSON OR THE READINGS WILL BE INCORRECT"*

Load the Linux image under test.

Login

Put the SUT into a suspend state by pressing the S2 button

With the device in suspend mode, take power measurements of the SOM and the KIT with the Wattson Power Measurement Tool.

Verify that the power consumption of the SOM ONLY in suspend mode is less than 10 mW.

Verify that the power consumption of the KIT in suspend mode is less than 30 mW.

Enter the average power consumption on the performance sheet of the TPL

21.03.04 Power Usage, idle state

<Test Removed>

21.04 Read/Write Performance

21.04.01 SD Read/Write performance

[PERF-11-011]

For each from {class 2 SD card, class 6 SD card}

Load the Linux image under test.

Insert a SD card with at least 10MB of free space.

Navigate to /home directory.

```
DM-37x# cd /home
```

Create file of random data (10MB)

```
DM-37x# dd if=/dev/urandom of=/home/testdata bs=1K count=10K
```

Make sure that the file testdata was created

```
DM-37x# ls /home -l
```

Calculate the time to copy the file from RAM to SD card add time to copy and time sync and enter in the Performance sheet of TPL

```
DM-37x# time cp /home/testdata /mnt/mmcblk0p1/testdata; time sync
```

Make sure the file was written to the SD card

```
DM-37x# ls -l /mnt/mmcblk0p1
```

Enter the SD write performance on the performance sheet of the TPL

Erase the testdata file from RAM

```
DM-37x# rm /home/testdata
```

Calculate the time to copy the file from SD card to RAM add time to copy and time sync and enter in the Performance sheet of TPL

```
DM-37x# time cp /mnt/mmcblk0p1/testdata /home/testdata; time sync
```

Enter the SD read performance on the performance sheet of the TPL

Erase the testdata file from SD card and RAM:

```
DM-37x# rm /mnt/mmcblk0p1/testdata
```

```
DM-37x# rm /home/testdata
```

End for each from {class 2 SD card, class 6 SD card}

21.04.03 USB Read/Write performance

[PERF-11-013]

Load the Linux image under test.

Insert a 256MB Mini Cruzer USB stick with at least 10 megs of free space.

Navigate to /home directory.

```
DM-37x#      cd /home
```

Create file of random data (10 megs)

```
DM-37x# dd if=/dev/urandom of=/home/testdata bs=1K count=10K
```

Make sure that the file testdata was created

```
DM-37x# ls /home -l
```

Calculate the time to copy the file from RAM to USB MSD card add time to copy and time sync and enter in the Performance sheet of TPL

```
DM-37x# time cp /home/testdata /mnt/sda1/testdata; time sync
```

Make sure the file was written to the USB MSD card

```
DM-37x# ls -l /mnt/sda1
```

Enter the USB MSD write performance on the performance sheet of the TPL

Erase the testdata file from RAM

```
DM-37x# rm /home/testdata
```

Calculate the time to copy the file from USB MSD card to RAM add time to copy and time sync and enter in the Performance sheet of TPL

```
DM-37x# time cp /mnt/sda1/testdata /home/testdata; time sync
```

Enter the USB MSD read performance on the performance sheet of the TPL

Erase the testdata file from USB MSD and RAM

```
DM-37x# rm /mnt/sda1/testdata
```

```
DM-37x# rm /home/testdata
```

21.04.04 NAND Read/Write performance

[PERF-11-016]

Load the Linux image under test.

Create the NAND directory.

```
DM-37x# mkdir -p /mnt/mtd-nand
```

View partitions

```
DM-37x# cat /proc/mtd
```

Note: Be sure the following command uses a NAND partition. Use a mtdblock other than 4 if necessary

Mount the NAND partition.

```
DM-37x# mount -t yaffs2 /dev/mtdblock5 /mnt/mtd-nand
```

Navigate to /home directory.

```
DM-37x#  cd /home
```

Create file of random data (10 megs)

```
DM-37x# dd if=/dev/urandom of=/home/testdata bs=1K count=10K
```

Make sure that the file testdata was created

```
DM-37x# ls /home -l
```

Calculate the time to copy the file from RAM to the NAND partition add time to copy and time sync and enter in the Performance sheet of TPL

```
DM-37x# time cp /home/testdata /mnt/mtd-nand/testdata; time sync
```

Make sure the file was written to the NAND partition

```
DM-37x# ls -l /mnt/mtd-nand
```

Erase the testdata file from RAM

```
DM-37x# rm /home/testdata
```

Calculate the time to copy the file from NAND partition to RAM add time to copy and time sync and enter in the Performance sheet of TPL

```
DM-37x# umount /mnt/mtd-nand
DM-37x# mount -o sync -t yaffs2 /dev/mtdblock5 /mnt/mtd-nand
DM-37x# time cp /mnt/mtd-nand/testdata /home/testdata; time sync
```

Erase the testdata file from the NAND partition and from RAM

```
DM-37x# rm /mnt/mtd-nand/testdata
DM-37x# rm /home/testdata
```

21.04.05 RAM Read/Write performance

[PERF-11-015]

Load the Linux image under test.

Navigate to /dev/shm directory.

```
DM-37x# cd /dev/shm
```

Check available memory (/dev/shm). Adjust the count of the dd command below if there is not enough free memory to accommodate 2 copies of the random data file (2 x 10 megs).

```
DM-37x# df -h
```

Create file of random data (10 megs)

```
DM-37x# dd if=/dev/urandom of=/dev/shm/testdata bs=1K count=10K
```

Make sure that the file testdata was created

```
DM-37x# ls /dev/shm -l
```

Calculate the time to copy the file from RAM to RAM card add time to copy and time sync and enter in the Performance sheet of TPL

```
DM-37x# time cp /dev/shm/testdata /dev/shm/testdata2; time sync
```


Make sure the file was written to RAM

```
DM-37x# ls -l /dev/shm
```

Erase the testdata and testdata2 files from RAM

```
DM-37x# rm /dev/shm/testdata
DM-37x# rm /dev/shm/testdata2
```

21.04.06 SD read performance in Uboot

<Test Removed>

21.05 Bandwidths

21.05.02 Bandwidth and system power (Ethernet)

[PERF-11-010]

Load the linux image under test

Plug USB standard A to USB Mini B cord into Workstation and PWR MEAS port

Start Wattson on the Workstation

Using Wattson, measure the average power consumption while running iperf by hitting the “Reset Min/Max/Avg” button on Wattson as you start iperf (see below)

Connect the Workstation and the SUT to the router via Ethernet cables

Put the SUT into 10BASE-T mode:

```
DM-37x# ifconfig eth0 up
DM-37x# ethtool -s eth0 speed 10 duplex full autoneg on
```

Make sure that the DUT is at: speed = 10Mb/s

```
DM-37x# ethtool eth0
```

Get an IP Address for eth0

```
DM-37x# udhcpc -i eth0
```

Run iperf with the SUT in client mode for 30 seconds

(See Appendix C for details on running iperf)

Enter the performance on the performance sheet of the TPL (outgoing)

Put the SUT into 100BASE-T mode:

```
DM-37x# ethtool -s eth0 speed 100 duplex full autoneg on
```

Make sure that the DUT is at: speed = 100Mb/s

```
DM-37x# ethtool eth0
```

Run iperf with the SUT in client mode for 30 seconds

(See Appendix C for details on running iperf)

Enter the performance on the performance sheet of the TPL (outgoing)

21.06 Card Recognition Times

21.06.01 Insertion and removal times

[PERF-11-014]

Load the linux image under test

Using a stopwatch, measure the amount of time the system takes to recognize insertion and removal of USB mass storage, SD and Ethernet.

For each from (SD, USB mass storage, Ethernet):

*NOTE: *For the system to see insertion/removal of Ethernet, you must bring it up first with:*

```
DM-37x# ifconfig eth0 up
```

Plug in device (or unplug if already plugged in)

Enter the performance on the performance sheet of the TPL (outgoing)

Unplug in device (or plug in if already unplugged)

Enter the performance on the performance sheet of the TPL (outgoing)

End For each from (SD, USB mass storage, Ethernet):

21.07 Bandwidth Tracking

21.07.01 Ethernet bandwidth tracking

[PERF-11-010] [PERF-16-022]

Open the TPL from the previous round of Linux testing.

Using the values on the PERF tab of the TPL, compare the bandwidth of the Ethernet transfers in test 21.05.02 from the previous TPL to the results achieved in Test 21.05.02 of this current round of testing

Verify that the results have not gotten slower (current bandwidth >= previous bandwidth)

Enter the performance on the performance sheet of the TPL

21.08 Transition Times

21.08.01 Run / Idle to Suspend transition time

[PERF-11-009] [PERF-13-013] [PERF-13-014]

Load the Linux image and log in.

Load the Wattson Power Measurement Program and connect the DUT via a USBOTG cable.

Put the board into Suspend state

```
DM-37x# echo mem > /sys/power/state
```

Using Wattson, determine the time taken to transition from the run / idle state to the suspend state by measuring how long it takes the system to transition from the “run / idle” power level to the “suspend” power level.

Verify that the user interface is suspended within 1 second. (screen goes black)

Verify that the transition time from the “run / idle” state to the “suspend” state is less than 1.5 seconds. (power bottoms out)

The time of transition from Run / Idle to Suspend should be determined using the SOM power line, not the KIT power line.

Enter the run / idle to suspend transition time on the performance sheet of the TPL

21.08.02 Suspend to Run / Idle transition time

[PERF-11-009] [PERF-13-015] [PERF-13-016]

Load the Linux image and log in.

Load the Wattson Power Measurement Program and connect the DUT via a USBOTG cable.

Put the board into Suspend state

```
DM-37x# echo mem > /sys/power/state
```

Before bringing board back from suspend, read verification steps below:

Verify that the transition time from the "suspend" state to the "run / idle" state is less than 1 sec.

Verify that the Linux prompt returns from typing on keyboard or touching LCD within 3.5 sec.

Take the board out of suspend state by typing on the keyboard or touching the LCD screen.

Using Wattson, determine the time taken to transition from the Suspend state to the Run / Idle state by measuring how long it takes the system to transition from the "suspend" power level to the "run / idle" power level.

The time of transition from Suspend to Run / Idle should be determined using the SOM power line, not the KIT power line.

Enter the suspend to run / idle transition time on the performance sheet of the TPL

21.08.03 Run to Idle / Idle to Run transition time

<Test Removed>

21.09 Benchmarking tests

**Note: If a benchmark build is not available, follow the instructions in Appendix E to create one. Use that build to test the following.*

21.09.01 Whetstones

[PERF-16-019]

POR

cd to /usr/bin

```
DM-37x# cd /usr/bin
```

Run the Whetstones benchmarking program

```
DM-37x# ./whetstones 60000
```

Verify that the Whetstones gets a score of 140 MIPS or higher.

Enter the whetstones benchmark output into the PERF tab of the TPL

21.09.02 Dhrystones

[PERF-16-020]

POR

cd to /usr/bin

```
DM-37x# cd /usr/bin
```

Run the Dhrystones benchmarking program

```
DM-37x# ./dhrystones
```

When prompted for how many runs, enter “10000000”

Verify that the Dhrystones gets a score of 1,000,000 or higher.

Enter the dhrystones benchmark output into the PERF tab of the TPL

21.09.03 Linpack

[PERF-16-021]

POR

cd to /usr/bin

```
DM-37x# cd /usr/bin
```

Run the Linpack benchmarking program

```
DM-37x# ./linpack
```

Verify that the Linpack gets a score of 18,000 KFLOPS or higher.

Enter the linpack benchmark output into the PERF tab of the TPL

21.09.04 DSP frequency benchmark

< Test Removed 09/28/2015 >

21.09.05 DSP frequency voltage levels

< Test Removed 09/28/2015 >

Appendix A: Test Progress Log (TPL)

The Test Progress Log (TPL) must be appended at the completion of each individual test case. Only five test result qualifiers are permitted. See the section on test requirements and guidelines for additional information. For ease of reference, a list of the acceptable test results has been duplicated here.

The TPL is an excel spreadsheet maintained in the version control system.

Pass - Test passes without any indication of failure.

Fail - Test fails and may represent a significant issue. A failure will be entered into the issue tracking system.

N/A – Test is not applicable. See comments for reason the test is N/A.

Example Entries

The following are examples of proper Test Progress Log entries. Note that all of the tests that are not recorded as PASS in the Progress Log would also be documented with greater detail in the issue tracking system.

Section	Test #	Test Results	RC Tested	Failing DUTs	Failure Description	Issue #	Notes
Display Driver Tests	01.00.00	pass	RC1				
	01.01.01	fail	RC1	SomLV	Display colors wrong	DM37LINUX-xxxx	
	01.02.01	na					Not tested in this release

Appendix B: NAND STRESS Testing

Purpose

This test stresses the NAND filesystem and verifies the system is in tact following load testing.

Setup

Test Steps

Perform the following subtests and record results in the action log:

1) Start a new log on Tera Term to track the results of this test. When the test is finished, put the logs and an overall results file (that you will create) into a folder called
Nand _Stress_test_results in the results folder for the SUT.

2) Create a Linux DM3730/AM3703 Torpedo + Wireless SD boot card with the SUT if you do not already have one.

3) In the folder that you got this STP from is another folder called “Linux_nand_stress_test_files”. View the “linux_nand_readme.txt” file and make the necessary changes to run the NAND stress test.

4) Copy the contents of the Linux_nand_stress_test_files from the folder that you got this STP from.

5) Create a LogicLoader SD Boot card with the current version of LogicLoader if you do not already have one.

6) In Uboot, erase Nand:

```
OMAP Logic # nand erase.chip
```

7) Using the Linux DM3730/AM3703 Torpedo + Wireless SD boot card you created, boot the DUT and log in as root.

8) Verify the SD card is mounted (look in /mnt/mmcblk0p1 for your files).

```
DM-37x# ls /mnt/mmcblk0p1
```

9) Execute the stress test.

```
DM-37x# sh /mnt/mmcblk0p1/nand-stress-test.sh
```

NOTE: If you use Windows to copy the file, it uses CRLF as line endings as opposed to LF
(at linux expects). To kill the line endings, do:

```
DM-37x# tr -d '\r' < /mnt/mmcblk0p1/nand-stress-test.sh > /tmp/nand-stress-test.sh
DM-37x# sh /tmp/nand-stress-test.sh
```

Create a text file of the results of the NAND Stress testing that lists all of the boards tested and how far they got in testing. Put this file in the Nand _Stress_test_results folder in the results folder for the SUT.

Appendix C: Using Iperf

Iperf is a freeware traffic generation tool. It is a client/server application that should run on both peers, and generates UDP and TCP traffic from the client to the server, according to the configuration described in this appendix.

You should run the server before the client.

C.1 TCP Iperf Command

TCP Client Command: This command should be run on the peer that generates and transmits the data. ./Iperf -c 'Destination IP' -I 'Interval' -p 'TCP Port' -t 'Time in Seconds' -r

For Example: To transfer data in both directions (first upload then download) to/from the server for 90 seconds, and display the data in intervals of 2 seconds:

```
# iperf -c 192.168.1.10 -i 2 -p 6000 -t 90 -r
```

Client connecting to 192.168.1.10, TCP port 6000 TCP window size: 16.0 kByte (default)

TCP Server Command: This command should be run on the peer that receives the data. ./Iperf -s -I 'Interval' -p 'TCP Port' For Example: Running the TCP server to the SUT.

```
# iperf -s -I 2 -p 6000
```

Server listening on TCP port 6000 TCP window size: 85.3 kByte (default)

C.2 UDP Iperf Command

UDP Client Command: This command should be run on the peer the generates and transmits the data. ./Iperf -c 'Destination IP' -u -b 'bandwidth' -i 'Interval' -p 'UDP Port' -t 'Time in Seconds' For Example: Running UDP 1Mbps traffic from the SUT.

```
# iperf -c 192.168.1.10 -u -b 1M -i 2 -p 6000 -t 90 -
r
```

Client connecting to 192.168.1.10, UDP port 6000 Sending 1470 byte datagrams UDP buffer size: 99.0 kByte (default)

UDP Server Command: This command should be run on the peer that receives the data. ./Iperf -s -I -u 'Interval' -p 'UDP Port' For Example: Running the UDP server on the SUT.

```
# iperf -s -i 2 -p 6000 -u
```

Server listening on UDP port 6000 Receiving 1470 byte datagrams UDP buffer size: 99.0 kByte (default)

Appendix D: Verification of NVS file

If the md5sum of the NVS file on the SOM does not match the one provided in test 16.09.03, you will need to run the steps below to test whether or not the NVS file is based off of one of the proper files in SVN.

To compare the lower half of the NVS file (this is the .ini file) perform the following steps:

POR

Pull the lower half of the NVS file out into a separate file. This is the .ini file.

```
DM-37x#  dd if=/lib/firmware/ti-connectivity/wl128x-nvsxxx.bin of=/tmp/test-ini.bin bs=1 skip=468
```

Where:

xxx = blank *for SOMDM3730-30-xxxxxxxx-x*

xxx = *-tw31* for SOMDM3730-31-xxxxxxxx-x

xxx = *-tw32* for SOMDM3730-32-xxxxxxxx-x

Do an md5sum of the newly created .ini file:

```
DM-37x#  md5sum /tmp/test-ini.bin
```

Save the md5 number for later comparison.

Copy the FCC .ini (see table below) file from Omnify onto an SD card, renaming it “golden-FCC.ini”. The Omnify Part numbers for the FCC .ini file can be found in the test section

“16.09 Verify FCC versions of .ini and .bts are included in latest Linux releases”

Wi-fi .ini files for md5sum comparison:

1024269_RevA.ini	SOMDM3730 -31
1031274_RevA.ini	SOMDM3730 -32

Move to the wl12xx folder

```
DM-37x# cd /home/root/wl12xx
```

Run the calibrator on the kit using the FCC .ini file and generate a new wl128x-nvs.bin:

```
DM-37x# ./calibrator set ref_nvs /mnt/mmcblk0p1/ golden-FCC.ini /tmp/omnify_nvs.bin
```

Cut off the first 468 bytes (that is overlaid with the product ID data) of the new wl128x-nvs.bin file.

```
DM-37x# dd if=/tmp/omnify_nvs.bin of=/tmp/omnify-ini.bin bs=1 skip=468
```

Calculate its md5sum:

```
DM-37x# md5sum /tmp/omnify-ini.bin
```

Verify that the md5sum of the newly created ini file (omnify-ini.bin) matches the md5sum (test-ini.bin) found above.

Appendix E: Benchmark Build

Below are the instructions on how to build an image to run benchmark tests in section 21.09.

**NOTE*: If you ran any other build previous to this, you will want to unselect the options that you selected for that build when you select your options for this build.*

It is suggested that you make the screen full size as trying to change the size after the GUI for LTIB starts will cause the GUI to look funky and unreadable.

Run LTIB

```
logic@logic-desktop:~$ ./ltib -c
```

Scroll down to "Package list --->" and hit enter

Scroll down to "[] miscellaneous benchmarks " and hit select (it is in the “b” section, not “m” section)

Highlight "< Exit >" and hit enter. Keep "exiting" until you're asked "Do you want to save your new configuration", then highlight "< Yes >" and hit enter.

At this point the build should kick off. It could take up to an hour or more for it to complete depending on the speed of your machine.

When the build completes, insert the SD card into your workstation and make it a bootable SD card.

```
logic@logic-desktop:~$      cd /home/logic/Logic_BSPs/Linux_3.0/xxx where xxx is the
location of LTIB
logic@logic-desktop:~$      ./bin/mkLogicFATcar.sh -c
```

Alternatively, you can create an SD boot card by using a freshly formatted card and putting onto it the following files:

MLO (x-load.bin.ift) in <ltib folder>/rootfs/boot

u-boot.bin in <ltib folder>/rootfs/boot

u-boot.bin.ift in <ltib folder>/rootfs/boot

uImage in <ltib folder>/rootfs/boot

rootfs.ext2.gz.uboot in <ltib folder>

Put the SD card into the DUT and power on.

Boot and login

Verify that the board booted and you were able to login

USE THIS IMAGE TO PERFORM THE TESTS IN SECTION 21.09 – Benchmarking tests (Whetstone, Dhrystone, Linpack)

Appendix F: STP Addendum (documentation, Location in the DM37 Linux User's Guide)

Display

DISP-01-017:	The method of including and excluding the frame buffer device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces
DISP-01-018:	The method of enabling and disabling the frame buffer device driver shall be documented in the BSP's User Manual.	3.2.2.1 Configure Display
DISP-01-019:	The methods of selecting the proper Beacon EmbeddedWorks display kit shall be documented in the BSP's User Manual.	Section 3.2.2.1 Configure Display
DISP-01-020:	The screen sizes and resolutions supported by the frame buffer device driver should be documented in the BSP's User Manual.	DM3730 Technical Reference Manual (SPRUGN4R)

Audio

AUDIO-01-012:	The method of including and excluding the audio device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces

Wired Ethernet

ETHER-01-022:	The method of including and excluding the Ethernet device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces

ETHER-01-023: The method of enabling and disabling the Ethernet device driver shall be documented in the BSP's User Manual.	4.5.1.4 Using DHCP 4.5.1.5 Using Static IP Bringing down Ethernet Not Done
ETHER-01-024: All supported kernel boot parameters affecting the Ethernet device driver's performance shall be documented in the BSP's User Manual.	4.5.1 Assign Development Kit IP Address
ETHER-01-025: The mechanisms to force the Ethernet hardware into specific speed and/or duplex modes shall be documented in the BSP's User Manual.	4.5.2 Set Speed, Duplex, and Auto-Negotiate

USB Host

USBH-01-021: The method of including and excluding the USB device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces

UART

UART-01-014: The method of including and excluding the serial device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces
UART-01-016: All supported kernel boot parameters affecting the serial device driver's performance shall be documented in the BSP's User Manual.	3.2.2 Configure Boot

MMC/SD

SDMMC-01-021:The method of including and excluding the MMC/SD device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces

SDMMC-01-023: All supported kernel boot parameters affecting the MMC/SD device driver's performance shall be documented in the BSP's User Manual.	3.2.10 Boot with X-Loader, U-Boot, Kernel, and Root Filesystem on SD Card

Touch Screen

TOUCH-01-012: The method of including and excluding the touch screen device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces

SPI

SPI-01-011: The method of including and excluding the SPI device driver from the Linux kernel shall be well documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces
SPI-01-014: The data sizes supported by the SPI device driver shall be well documented in the BSP's User Manual.	SPI Documentation can be found in the Linux distribution: 1027480_LogicPD_Linux_BSP_2.4-4/rpm/BUILD/linux/Documentation/spi/ spi-summary
SPI-01-015: The clock frequencies supported by the SPI device driver shall be well documented in the BSP's User Manual.	DM3730 Technical Reference Manual section 20.4.2.1 Clocking
SPI-01-016: The clock polarities and phases supported by the SPI device driver shall be well documented in the DM3730 Technical Reference Manual provided by TI.	DM3730 Technical Reference Manual section 20.4.2.1 Clocking

MTD and File Systems

FLASH-01-011: The method of including and excluding the flash file system	10.3.2 Remove Specific Interfaces

support from the Linux kernel shall be documented in the BSP's User Manual.	
FLASH-01-012: The types and combinations of flash file systems supported by the Linux BSP shall be documented in the BSP's User Manual.	8.1 Prepare Build Tools (DVSDK) (YAFFS)
FLASH-01-013: All supported kernel boot parameters affecting flash file system configuration and performance shall be documented in the BSP's User Manual.	3.2.9.5 YAFFS Root Filesystem Boot Example

Power Management

POWER-12-009:The method of entering each system power state shall be documented in the BSP's User Manual.	4.24 Run/Idle/Suspend
POWER-12-010:The method of exiting each system power state shall be documented in the BSP's User Manual.	4.24 Run/Idle/Suspend
POWER-12-011:The state of the SOM's hardware while in each power state shall be documented in the BSP's User Manual in language consistent with the hardware data sheets.	4.24 Run/Idle/Suspend – Not quite accurate

Wireless Ethernet

WIFI-06-014: The method of including and excluding the wireless Ethernet device driver from the Linux	10.3.2 Remove Specific Interfaces

	kernel shall be documented in the BSP's User Manual.	
WIFI-06-015:	The method of enabling and disabling the wireless Ethernet device driver shall be documented in the BSP's User Manual.	4.13.1 Start Wireless Interface in Station Mode 4.13.2 Start Wireless Interface in AP Mode

Bluetooth

BT-01-006:	The method of including and excluding the Bluetooth device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces
BT-01-007:	The method of enabling and disabling the Bluetooth device driver shall be documented in the BSP's User Manual.	4.14 Bluetooth Networking

DSP

DSP-07-014:	The Linux User Guide shall describe how to configure the Linux BSP to include DSPLink	8 Cameras and DSP (DVSDK)
DSP-07-015:	Documentation available to customers shall describe how to exercise DSP examples.	8 Cameras and DSP (DVSDK)

USB OTG

USBOTG-01-008:	The method of including and excluding the USB OTG device driver from the Linux kernel shall be documented in the BSP's User Manual.	10.3.2 Remove Specific Interfaces

GPS

GPS-13-005:	The method of including and excluding the GPS device driver from the Linux kernel shall be	10.3.2 Remove Specific Interfaces

	documented in the BSP's User Manual.	
GPS-13-006:	The method of enabling and disabling the GPS device driver shall be documented in the BSP's User Manual.	6.2.3 Run GPS Demo Application