OMAP35x Linux PSP

Datasheet

02.01.02.09

Published 16 June 2009

IMPORTANT NOTICE

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

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

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

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

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

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

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

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

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

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

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

Mailing Address:

Texas Instruments,
Post Office Box 655303,
Dallas, Texas 75265

Table of Contents

List of Figures

List of Tables

Read This First

About This Manual

This document provides an overview and performance data for each of the device drivers which are part of the OMAP35x Linux PSP package.

Notation of information elements

The document may contain these additional elements:

Warning

This is an example of warning message. It usually indicates a non-recoverable change, e.g. formatting a filesystem.

Caution

This is an example of caution message.

Important

This is an example of important message.

Note

This is an example of additional note. This usually indicates additional information in the current context.

Tip

This is an example of a useful tip.

If You Need Assistance

For any assistance, please send an mail to software support [mailto:softwaresupport@ti.com].

Trademarks

OMAP™ is a trademark of Texas Instruments Incorporated.

All other trademarks are the property of the respective owner.

OMAP35x Linux Drivers

Abstract

This chapter provides brief details on the device drivers supported in the Linux PSP release for OMAP35x platform.

1.1. Texas Instruments OMAP35x Linux Drivers

This chapter provides a brief description of the device drivers supported on OMAP35x along with any limitations.

Note

The constraints may vary across product releases. Please refer to the *Release Notes* accompanying the release for an updated list of constraints.

1.1.1. OMAP35x Linux PSP details

1. Supports OMAP35x EVM with ES2.1 and ES3.1 silicons.
2. Built with linux kernel version 2.6.29-rc3.
3. Compiled with code sourcery tool chain version arm-2008-q1.
4. Supports U-Boot version 2008.10.
5. Supports boot from Nand and MMC/SD.
6. Ships with sample root file system, ramdisk, pre-built u-boot, uImage binaries, docs and sample applications.

1.1.2. Device driver list

Peripheral	Description	Linux driver type
Display sub system (DSS)	Enables display on graphics pipeline using Fbdev and video pipeline using V4L2	Fbdev, V4L2
Video capture	Enables NTSC and PAL BT.656 capture	V4l2
Resizer	Enables upscaling to 4x and downscaling to 1/4x of input images	Character
Audio (McBSP)	Audio Record and Playback	ALSA SoC
Ethernet	Transmit/receive network data. Supports Auto negotiation with 10/100 Mbps link speed	Netdev
USB 2.0 MSC Host (MUSB, EHCI)	USB Mass Storage Class Host Driver	Block
USB 2.0 HID Host (MUSB, EHCI)	USB Human Interface Device Host Driver	Input driver
USB 2.0 ISO Host (MUSB, EHCI)	USB Isochronous class supporting audio and video	USB Host ISO
USB 2.0 MSC Slave	USB Mass Storage Class Slave Driver	USB Gadget
USB 2.0 CDC Slave	USB Communication Device Class	USB Gadget/Netdev
USB 2.0 RNDIS Slave	USB Remote Network Driver Interface Specification	USB Gadget/Netdev
OneNand, Micron Nand	Flash storage system	MTD Character and Block
MMC/SD	Interface to MultiMedia Secure Digital cards	Block
UART	Serial Communication Interface	Character
Touchscreen	Enables the LCD on OMAP35x EVM to be used as touch screen by using McSPI for communication	Input (event) driver
Power Management	Enables power management by supporting CPUIdle and dynamic tick.	CPUidle, dyntick
I2C	Inter-IC Communication	Character

Table 1.1. OMAP35x Peripheral Driver Support

1.1.3. Drivers not supported

1. SDIO -WLAN
2. RTC

Video Display Driver

Abstract

This chapter provides details on V4L2 display and Fbdev display drivers along with throughput and CPU load numbers.

2.1. Introduction

OMAP35x display hardware integrates one graphics pipeline, two video pipelines, and two overlay managers (one for digital and one for analog interface). Digital interface is used for LCD and DVI output and analog interface is used for TV out.

The primary functionality of the display driver is to provide interfaces to user level applications and management of OMAP35x display hardware. This includes, but is not limited to:

- GUI rendering through the graphics pipeline.
- Static image or video rendering through two video pipelines.
- Connecting each of three pipelines to either LCD or TV output so that the display layer is presented on the selected output path.
- Image processing (cropping, rotation, mirroring, color conversion, resizing, etc).

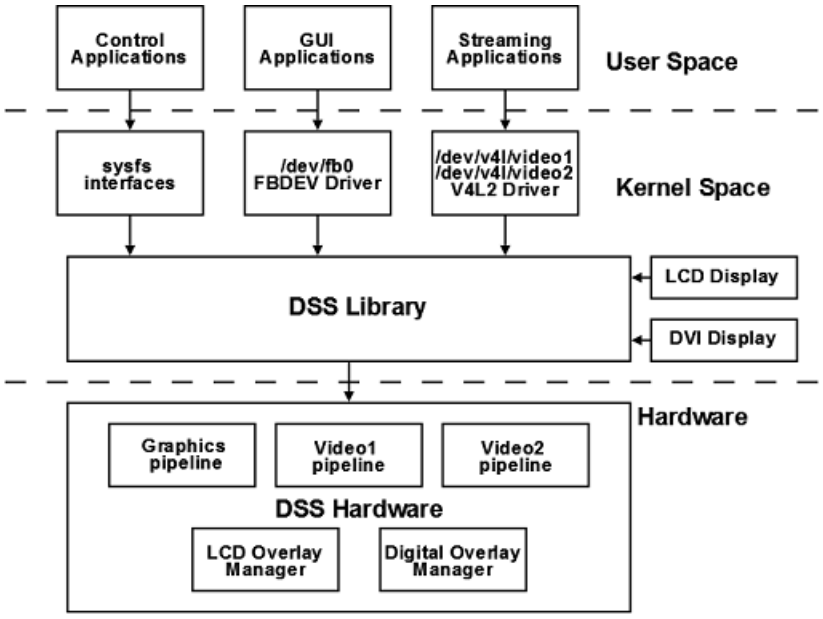

Figure 2.1. OMAP35x Display Subsystem Architecture

2.2. Features

Pixel formats supported on video plane are YUV, UYVU, RGB565, RGB24P and RGB24 unpacked on video1 and ARGB on video2 planes. RGB565, ARGB, RGBA, RGB24 packed and unpacked supported on graphics pipeline.

1. Video pipelines controlled through V4L2 user interface. Graphic pipeline controlled through FBDEV user interface.
2. Supports LCD display interface at VGA resolution (480*640)
3. Supports TV display interface at NTSC and PAL resolutions (only S-Video out is supported, composite out is not supported)
4. Configuration of parameters such as height and width of display screen, bits-per-pixel etc.
5. Supports setting up of OSD window destinations (TV or LCD) through sysfs on FBDEV interface while compile time option for V4L2 interface for video window.
6. Supports driver allocated (mmaped) and user, memory buffer in V4L2 and only driver allocated buffers in FBDEV.
7. Supports rotation - 0, 90, 180 and 270 degrees.
8. Supports DVI interface with 720P and 480P resolutions. Compile time selectable resolutions of DVI.
9. Supports scaling from 1/2x to 8x on video pipelines. Hardware supports scaling from 1/4x to 8x.
10. Supports Wait for Vsync and Panning feature under FBDEV.
11. Supports LCD Backlight control.
12. Supports Alpha blending, both global alpha and pixel alpha.
13. Supports Source and Destination color keying through V4L2 ioctls.

2.3. Features not supported

1. Mirroring not supported.
2. For RGB888 rotation is not supported because of hardware limitation.
3. Linking feature not supported on video pipelines.
4. Changing of modes in TV not supported.

2.4. Constraints

1. Sync Lost is observed when incorrect parameters are programmed on video and graphics pipeline.

2. Minimum 3 Buffers are required for streaming in V4L2.

3. PAL resolution can be set maximum to 720 x 574 instead of 720 x 576 because of hardware limitation. It is mentioned in hardware errata.

4. Upscaling and downscaling with images more that 720x574 resolution is not supported because of dss functional clock frequency limitation.

2.5. Supported system calls

`open(), close(), ioctl(), mmap(), munmap().`

Driver conforms to V4l2 framework. Please refer to this link for V4L2 ioctls- http://v4l2spec.bytesex.org/v4l2spec/v4l2.pdf

2.6. Supported Fbdev ioctls

IOCTL Command Name	Description
`FBIOGET_VSCREENINFO`	This I/O control is used to query the variable screen info. This allows an application to query the display mode, including the color depth, resolution, timing etc.
`FBIOPUT_VSCREENINFO`	This I/O control is used to set the variable screen info. This allows an application to change the display mode, including the color depth, resolution, timing etc.
`FBIOGET_FSCREENINFO`	This I/O control is used to get the fixed properties of the display, e.g. the start address of the framebuffer memory
`FBIOPUTCMAP`	This ioctl sets up the color map for frame buffer
`FBIOGETCMAP`	This ioctl gets the current color map for frame buffer
`FBIO_BLANK`	This ioctl is used to blank or unblank the frame buffer console
`FBIOPAN_DISPLAY`	This ioctl is used to pan the display buffer

Table 2.1. Supported fbdev ioctls

2.7. Supported V4L2 ioctls

IOCTL Command Name	Description
VIDIOC_ENUM_FMT	Enumerate supported formats by current decoder
VIDIOC_G_FMT	Gets the data format. Current settings for pixel format, width,height etc.
VIDIOC_S_FMT, VIDIOC_TRY_FMT	Sets the data format, try a format.
VIDIOC_G_CROP,VIDIOC_S_CROP	Queries or selects the current cropping rectangle.
VIDIOC_QUERYBUF	Queries the status of a buffer
VIDIOC_QBUF,VIDIOC_DQBUF	Exchanges a buffer with the driver.
VIDIOC_REQBUFS	Request buffers
VIDIOC_QUERYCAP	Queries the device capabilities
VIDIOC_STREAMON,VIDIOC_STREAMOFF	Starts or stops streaming I/O
VIDIOC_S_CTRL,VIDIOC_G_CTRL	These ioctls are used to set/get various V4L2 controls like rotation background color.
VIDIOC_QUERYCTRL	These ioctls are used to query various V4L2 controls like rotation and background color.
VIDIOC_CROPCAP	Get cropping parameters
VIDIOC_S_FBUF, VIDIOC_G_FBUF	Set/Get the overlay parameters like alpha blending status and color keying status.

Table 2.2. Supported V4L2 ioctls

2.8. DSS Performance and benchmarks

Setup details

Resource	Description
TV	Dell (LCD Monitor)
Ramdisk or NFS	NFS
CPU idle	Enabled

Table 2.3. DSS peformance setup details

Performance Numbers

Interface	Framework	Resolution	Frame rate (fps)	CPU Load (%)
LCD	Fbdev	VGA	60.26	0.24
LCD	V4L2	VGA	60.09	0.12
TV	V4L2	NTSC	30.00	0.15
TV	V4L2	PAL	25.03	0.17
TV	Fbdev	NTSC	30.06	0.18
TV	Fbdev	PAL	25.05	0.15
DVI	Fbdev	720P	60.31	0.36
DVI	Fbdev	480P	60.22	0.12
DVI	V4L2	480P	60.02	0.12
DVI	V4L2	720P	60.25	0.18

Table 2.4. DSS Performance

Video Capture Driver

Abstract

This chapter provides details on V4L2 Capture drivers along with throughput and CPU load numbers.

3.1. Introduction

The camera ISP is a key component for imaging and video applications such as video preview, video record, and still-image capture with or without digital zooming.

The camera ISP provides the system interface and the processing capability to connect RAW image-sensor modules and video decoders to the OMAP35x device.

The capture module consists of the following interfaces:

1. One S-video SD input in BT.656 format
2. One Composite SD input in BT.656 format

Both these video inputs are connected to one TVP5146 decoder and the application can select between these two inputs using standard V4L2 interface.Only one input can be captured or selected at any given time.

The following figure shows the basic block diagram of capture interface -

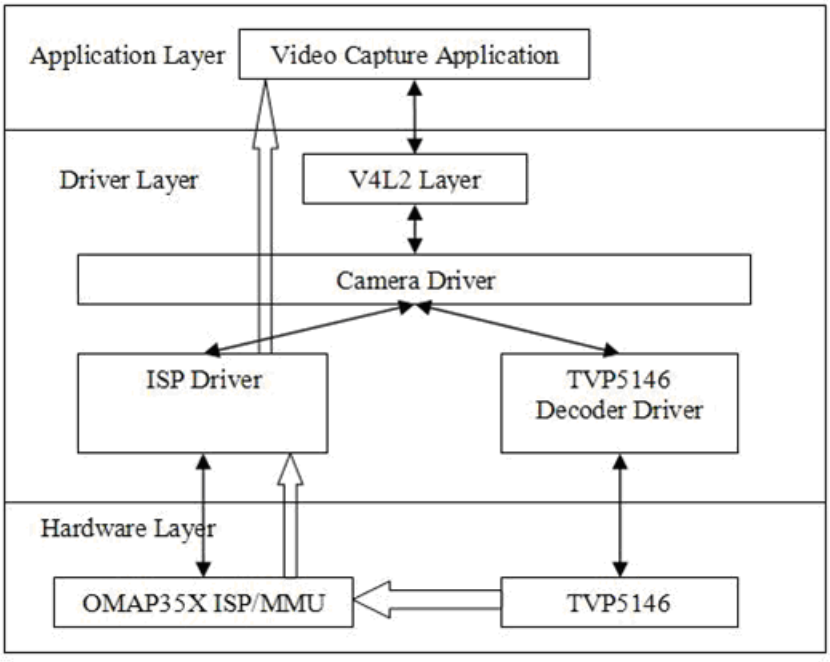

Figure 3.1. OMAP35x ISP-Capture Interface Block-Diagram

3.2. Features

1. Supports one software channel of capture and a corresponding device node (/dev/video0) is created.
2. Supports single I/O instance and multiple control instances.
3. Supports buffer access mechanism through memory mapping and user pointers
4. Supports dynamic switching among input interfaces with some necessary restrictions wherever applicable.
5. Supports NTSC and PAL standard on Composite and S-Video interfaces.
6. Supports 8-bit BT.656 capture in UYVY and YUYV interleaved formats.
7. Supports standard V4L2 IOCTLs to get/set various control parameters like brightness, contrast and saturation.
8. TVP5146 decoder driver module can be used statically or dynamically (insmod and rmmod supported).

3.3. Features not supported

1. Cropping and scaling operations and their corresponding V4L2 IOCTLs.
2. Raw capture (capture through sensor interface).

3.4. Constraints

1. Dynamic switching of resolution and dynamic switching of interfaces is not supported when streaming is on.
2. Driver buffer addresses and pitch must be aligned to 32 byte boundary.

3.5. Supported system calls

`open(), close(), ioctl(), mmap(), munmap().`

Driver conforms to V4l2 framework. Please refer to this link for V4L2 ioctls- http://v4l2spec.bytesex.org/v4l2spec/v4l2.pdf

3.6. Supported V4L2 ioctls

IOCTL Command Name	Description
`VIDIOC_ENUM_FMT`	Enumerate supported formats by current decoder
`VIDIOC_G_FMT`	Gets the data format. Current settings for pixel format, width,height etc.
`VIDIOC_S_FMT, VIDIOC_TRY_FMT`	Sets the data format, try a format.
`VIDIOC_QUERYBUF`	Queries the status of a buffer
`VIDIOC_QBUF,VIDIOC_DQBUF`	Exchanges a buffer with the driver.
`VIDIOC_REQBUFS`	Request buffers
`VIDIOC_QUERYCAP`	Queries the device capabilities
`VIDIOC_STREAMON,VIDIOC_STREAMOFF`	Starts or stops streaming I/O
`VIDIOC_S_CTRL,VIDIOC_G_CTRL`	These ioctls are used to set/get various V4L2 controls like saturation, brightness, hue etc.
`VIDIOC_QUERYCTRL`	These ioctls are used to query various V4L2 controls V4L2 controls like saturation, brightness, hue etc.

Table 3.1. Supported V4L2 ioctls

3.7. Capture Performance and benchmarks

Setup Details

Resource	Description
DVD player	Sony (DVP-NS51P)
TV	Dell (LCD)
Ramdisk or NFS	NFS
CPU idle	Enabled
Tool used (LPTB or open source)	LPTB

Table 3.2. Capture performance setup details

Performance Numbers

Resolution	Frame rate (fps)	CPU Load (%)
NTSC	30.03	3.47
PAL	26.02	3.59

Table 3.3. Capture Performance

Video Resizer Driver

Abstract

This chapter provides details on Resizer drivers along with throughput and CPU load numbers.

4.1. Introduction

OMAP35x Resizer module supports upscaling and downscaling. It resizes YUV422 image and stores output image in the RAM. The following figure shows the block diagram of resizer module

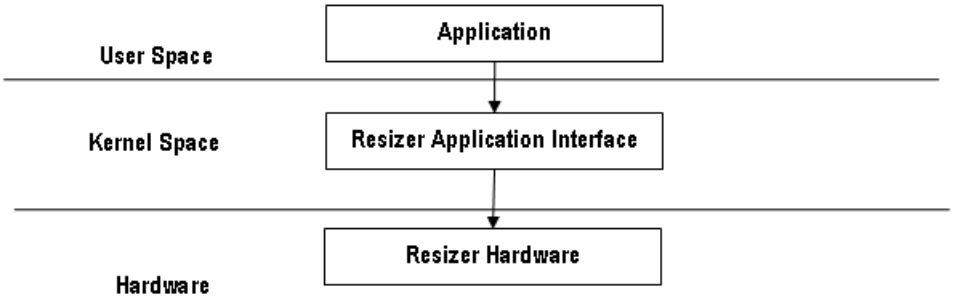

Figure 4.1. OMAP35x Resizer Interface Block-Diagram

4.2. Features

1. Resizes input frame stored in RAM and stores output frame in RAM.
2. Supports resizing from 1/4x to 4x.
3. Supports independent horizontal & vertical resizing.
4. Supports YUV422 packed and color separate data.
5. Supports driver allocated and user provided buffers.

4.3. Features not supported

1. On-the-fly mode of operation.

4.4. Constraints

1. All input,ouput addresses and pitch must be 32 bytes aligned.
2. Output image size cannot be more than 2047x2047.
3. Output width must be even.

4. Output width must be 16 byte aligned for vertical resizing.
5. The horizontal start pixel must be within range 0-15 for color interleaved and 0-31 for color separate data.

4.5. Supported system calls

`open(), close(), ioctl(), mmap(), munmap().`

4.6. Supported IOCTLS

Following IOCTL's are supported -

IOCTL Command Name	Description
RSZ_S_PARAM	Set the Resizer driver parameters
RSZ_G_PARAM	Get the Resizer driver parameters
RSZ_RESIZE	Starts the resizing process
RSZ_QUERYBUF	Request physical address of buffers allocated by RSZ_REQBUF
RSZ_REQBUF	Request to allocate buffers
RSZ_G_STATUS	Get the channel status for the particular current Resizer channel.
RSZ_S_EXP	Configure the Read cycle required for Resizer module.

Table 4.1. Supported Resizer Driver ioctls

4.7. Resizer Performance and benchmarks

Setup details

Resource	Description
Input image resolution(upscale)	176x144
Output image resolution(upscale)	704x576
Input image resolution(downscale)	704x576
Output image resolution(downscale)	176x144
CPU idle	Enabled

Table 4.2. Resizer performance setup details

Performance Numbers

Resize operation	Time taken (ms)	CPU Load (%)
Upscale to 4x	1.32	0.01
Downscale to 1/4x	3.12	0.01

Table 4.3. Resizer Performance

ALSA SoC Audio Driver

Abstract

This chapter provides details on ALSA SoC audio driver along with throughput and CPU load numbers.

5.1. Introduction

OMAP audio driver complies to the ALSA SoC framework. ASoC framework provides better audio support for embedded SoC procesors and portable audio codecs.

ASoC framework splits an embedded audio system into three components:

- Codec driver: The codec driver is generic and hardware independent code that configures the audio codec to provide audio capture and playback. It should contain no code that is specific to the target platform or machine.
- Platform driver: The platform driver can be divided into audio DMA and SoC Digital Audio Interface (DAI) configuration and control. The platform driver only targets the SoC CPU and must have no board specific code.
- Machine driver: The ASoC machine (or board) driver is the code that glues together the platform and codec drivers. It can contain codec and platform specific code. It registers the audio subsystem with the kernel as a platform device.

Following architecture diagram shows all the components and the interactions among them:

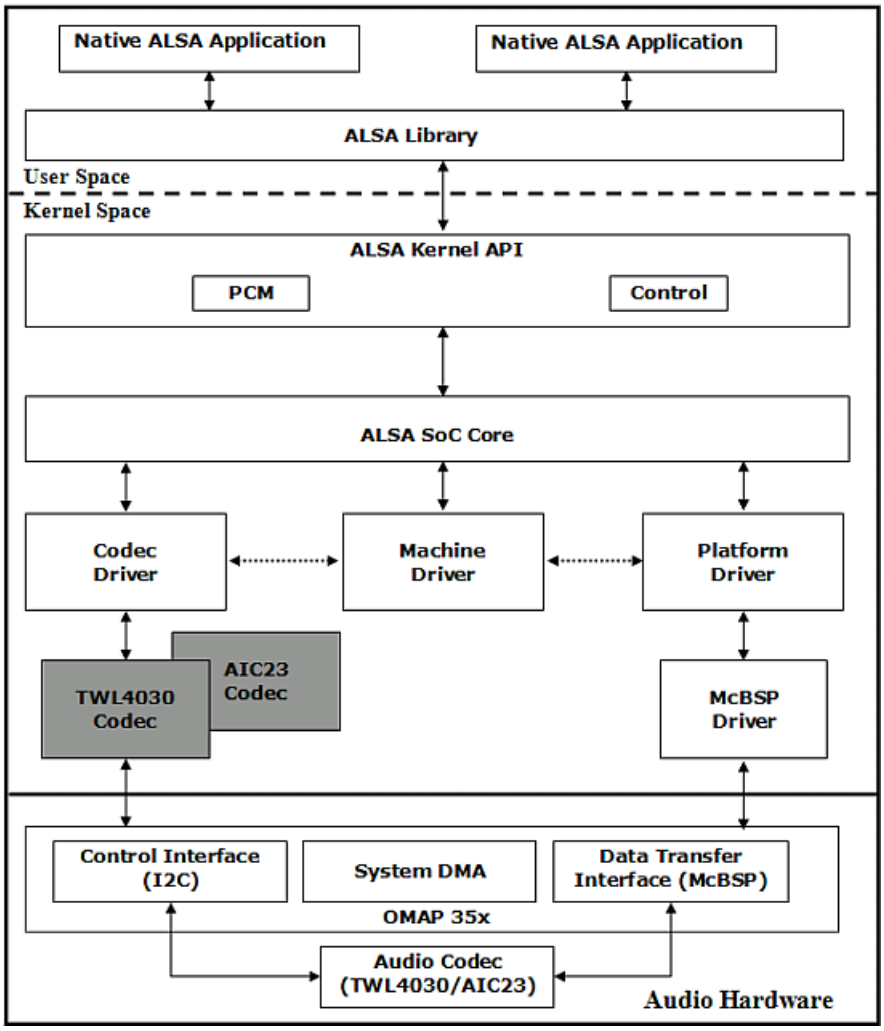

Figure 5.1. ALSA SoC Architecture

5.2. Driver Features

The driver supports the following features:

- Supports TWL4030 audio codec in ALSA SoC framework.
- Multiple sample rate support (8 KHz, 11.025 KHz, 12 KHz, 16 KHz, 22.05 KHz, 24 KHz, 32 KHz, 44.1 KHz and 48 KHz) for both capture and playback.
- Supports audio in both mono and stereo modes.
- Supports simultaneous playback and record (full-duplex mode).
- Start, stop, pause and resume feature.
- Supports mixer interface for TWL4030 audio codec.

5.3. Features Not Supported

None

5.4. Constraints

- Synthesizer and midi interfaces are not supported as many codecs do not support the same.
- Formats such as TDM, Left and Right Justified are currently not supported.
- The audio driver does not allow opening the same stream (playback/capture) multiple times.
- TWL4030 codec needs to be configured in master mode only, therefore McBSP can only be used as slave along with this codec.
- Configuration of playback and capture streams in different sampling rates is not supported.
- OSS emulation layer is not supported because of which OSS based applications (for e.g. madplay) may not work properly.

5.5. Supported System Calls

Refer ALSA project - the C library reference [http://www.alsa-project.org/alsa-doc/alsa-lib/] for API calls.

5.6. Supported IOCTLs

NA

5.7. Performance and Benchmarks

The performance numbers were captured using the following:

- Word length in bits = 16
- Number of channels per sample = 2
- CPU Idle: Disabled

Sampling Rate (in Hz)	**CPU Load (in %)**	**CPU Load (in %)**
	Normal Mode	**MMAP mode**
8000	0	0
32000	0	0
44100	1	1
48000	1	1

Table 5.1. Audio Write Performance

Sampling Rate (in Hz)	CPU Load (in %)	CPU Load (in %)
	Normal Mode	MMAP mode
8000	0	0
32000	0	0
44100	1	1
48000	1	1

Table 5.2. Audio Read Performance

Power Management

Abstract

This chapter describes the power management features.

6.1. Introduction

OMAP35x silicon provides a rich set of power management features. The features include Clock control at module level, multiple power and voltage domains etc.

This section provides provides an overview of power management features supported and steps to enable these features in the kernel configuration. It also provides the typical power consumption observed for different scenarios.

6.1.1. cpuidle

When idle loop is executed, kernel is not doing any useful 'work'. This is an opportunity to save power.

6.1.2. cpufreq

Only **basic** support for this driver is included in this release. It is not fully supported.

6.1.3. SmartReflex™

SmartReflex is a power-management technique for controlling the operating voltage of a device to reduce its active power consumption. It helps in achieving optimal performance/power trade-off for all devices across the technology process spectrum and across temperature variations.

6.2. Features

1. Supports Dynamic Tick framework.
2. Supports the *cpuidle* framework with MPU and Core transition to RETENTION and OFF states. The *menu* governor is supported.
3. Supports the suspend and resume feature defined in LDM.
4. Provides basic implementation for *cpufreq*.
5. Support SmartReflex with automatic (hardware-controlled) mode of operation.

6.3. Features Not Supported

1. Allow drivers and applications to limit the idle state that can be entered.
2. Support for SmartReflex with manual (software-controlled) mode of operation.

6.4. Constraints

1. Some of the drivers do not leverage the power-saving features supported by the silicon.

   They need to enable/ disable corresponding clocks via `clk_enable()` and `clock_disable()` only when the clocks are *really* needed.

2. After the system is suspended, the resume operation does not succeed from the keypad and touchscreen.

6.5. Power Measurements

The current and voltage were measured for VDD1 in these scenarios. All measurements were done using National Instruments DAQPAD 6015 using TI power consumption tool.

Measurements are done across jumpers J6 for a period of approx 1 minute. Summing average for this period is provided in the table below.

Scenario 1

- Build `uImage` with power management disabled and dynamic tick disabled.
- Boot the OMAP35x ES2.1 EVM with this `uImage` ramdisk.

Scenario 2

- Build `uImage` with power management disabled but dynamic tick enabled.
- Boot the OMAP35x ES2.1 EVM with this `uImage` ramdisk.

Scenario 3

- Build `uImage` with power management and dynamic tick enabled; but no cpuidle.
- Boot the OMAP35x ES2.1 EVM with this `uImage` ramdisk.
- After boot-up, execute these commands:

```
$ echo 1 > /sys/power/clocks_off_while_idle
$ echo 1 > /sys/power/sleep_while_idle
```

- Wait for 30 secs and start the measurement.

Scenario 4

- Build `uImage` with power management, dynamic tick and cpuidle enabled.
- Boot the OMAP35x ES2.1 EVM with this `uImage` ramdisk.
- After boot-up, execute these commands:

```
$ echo 1 > /sys/power/clocks_off_while_idle
$ echo 1 > /sys/power/sleep_while_idle
```

- Wait for 30 secs and start the measurement.

Scenario 5

- Build `uImage` with power management, dynamic tick and cpuidle enabled.
- Boot the OMAP35x ES2.1 EVM with this `uImage` ramdisk.
- After boot-up, execute these commands:

```
$ echo 1 > /sys/power/clocks_off_while_idle
$ echo 1 > /sys/power/sleep_while_idle
$ echo 1 > /sys/power/enable_off_mode
```

- Wait for 30 secs and start the measurement.

The measurements against these scenarios is recorded in the table below:

Scenario	Voltage (Volts)	Current (mA)
Scenario 1	1.19	7.50
Scenario 2	1.19	7.51
Scenario 3	1.20	0.11
Scenario 4	1.20	0.15
Scenario 5	1.20	0.12

Table 6.1. Power measurements

Ethernet Driver

Abstract

This chapter provides details on Ethernet driver along with throughput and CPU load numbers.

7.1. Introduction

The ethernet driver supports the linux netdev interface. The SMSC9115 module is connected to the OMAP chip over the GPMC interface.

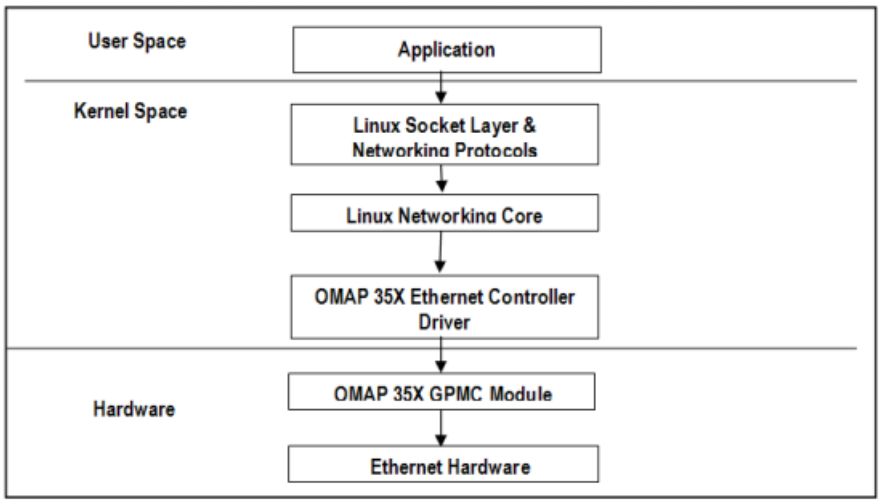

Figure 7.1. Linux Kernel Ethernet Driver

7.2. Driver Features

The driver supports the following features:

- 10/100 Mbps mode of operation.
- Auto negotiation.
- Support for multicast and broadcast frames.
- Promiscuous mode of operation.
- Full duplex and half duplex mode of operation.

7.3. Features Not Supported

- Linux NAPI support

7.4. Constraints

- Link might not be established if peer does not enable auto-negotiation.

7.5. Supported System Calls

None

7.6. Supported IOCTLs

None

7.7. Performance and Benchmarks

TCP Window Size (in KBytes)	Bandwidth(Tx +Rx) (in Mbits/sec)	Transfer size (in MBytes)	Interval (in Seconds)
16	46.22	331.20	60
32	46.30	331.3	60
64	46.73	333.97	60
128	46.85	335.4	60
208	46.45	332.66	60

Table 7.1. Ethernet 100Mbps Mode Performance

The performance numbers were captured using the iperf tool. Usage details are mentioned below:

- Server side command switch : "-s"
- Client side command : "-c <server ip> -w<window size> -d -t60"
- Iperf tool is run on the Win PC in server mode and on DUT in client mode. Version 1.7.0 is used on server side and 2.0.4 is used on client side.
- Data captured here is for "iperf" in client mode.
- Straight cable is used to measure performance.
- Speed is set to 100Mbps
- CPU idle : Enabled

Flash Driver

Abstract

This chapter describes the NAND flash driver architecture, driver features and performance numbers (throughput and CPU load) for NAND/OneNAND flash devices.

8.1. Introduction

The NAND flash driver is implemented as both - block driver and character driver, compliant with the standard MTD driver. The NAND flash driver creates the device nodes for user space access (`/dev/mtdblock0`, `/dev/mtdblock1`, `/dev/mtd0`, `/dev/mtd1` etc). Currently OneNAND and Micron NAND devices are supported by the driver.

This figure illustrates the stack diagram of flash driver in Linux.

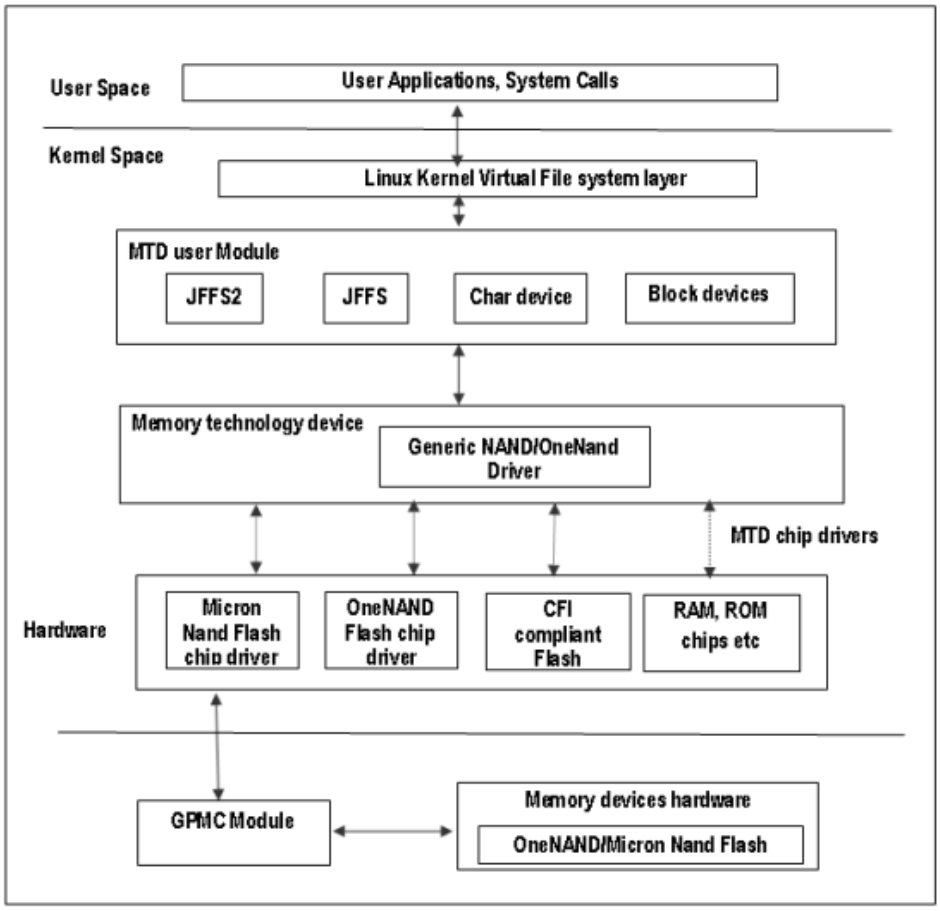

Figure 8.1. Linux Kernel NAND driver

8.2. Driver Features

The driver supports the following features:

1. JFFS2 file system
2. Supports Read/Write, Erase, Unlock Operations and BBM
3. OneNAND has been divided into 5 partitions:
    1. 512KB partition for X-loader
    2. 1920 KB (Read Only) partition for U-Boot
    3. 128 KB (Read Only) for environment variables
    4. 5 MB (Read/Write) partition for Linux
    5. Remainder for file system and others (Read/Write)

8.3. Features Not Supported

None

8.4. Constraints

None

8.5. Supported System Calls

`open(), close(), read(), write(), fread(), fwrite(), ioctl()`

8.6. Performance Benchmarks

8.6.1. OneNand write and read performance

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	3.94	100
256	100	6.99	100
512	100	6.97	100
1024	100	6.98	100
5120	100	6.95	100

Table 8.1. OneNand write performance values

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	11.94	100
256	100	12.99	100
512	100	12.99	100
1024	100	12.90	100
5120	100	14.04	100

Table 8.2. OneNand read performance values

The performance numbers are captured using the following:

1. OneNAND PART Number: Samsung 801 K5W1GACM-DL60
2. CPU idle : Enabled
3. File Format : JFFS2

8.6.2. Micron Nand write and read performance

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	4.09	100
256	100	4.09	100
512	100	5.36	100
1024	100	5.35	100
5120	100	5.30	100

Table 8.3. Micron Nand write performance values

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	6.44	100
256	100	6.42	100
512	100	6.42	100
1024	100	6.40	100
5120	100	6.41	100

Table 8.4. Micron Nand read performance values

The performance numbers are captured using the following:

1. Micron NAND PART Number: Micron 8AA18JY192

2. CPU idle : Enabled
3. File Format : JFFS2

MMC/SD Driver

Abstract

This chapter provides details on MMC/SD driver along with throughput and CPU load numbers.

9.1. Introduction

The MMC controller provides an interface to external MMC cards that follow the MMC specification v4.0. The MMC driver is implemented as a block driver. Block device nodes(such as `/dev/mmcblk0p1`) are created for user space access.

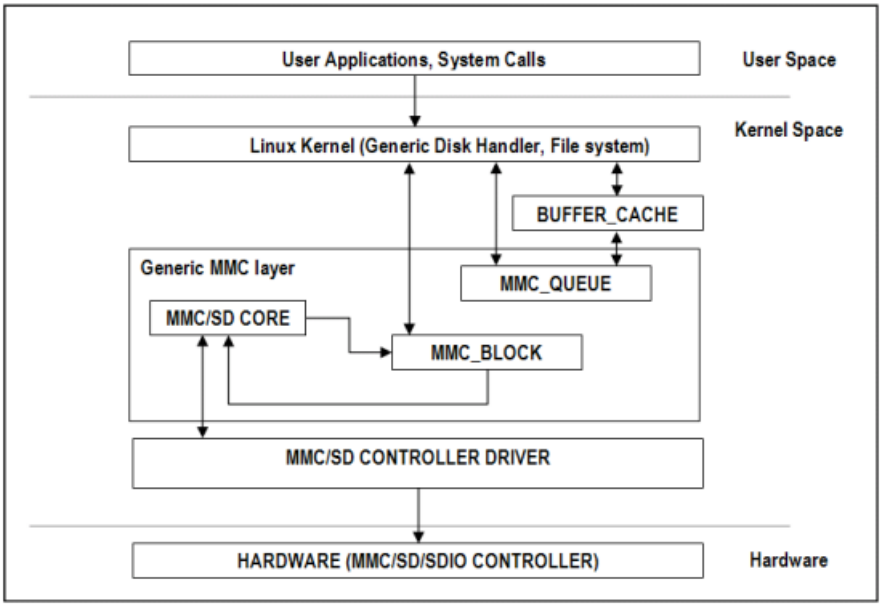

Figure 9.1. Linux Kernel MMC Driver

9.2. Driver Features

The driver supports the following features:

- MMC/SD native protocol command/response set
- Single/multiple block data transfers
- Linux file system and generic MMC layer abstract details of block devices (MMC)
- High-speed (SDv1.1) and High Capacity (SDv2.0) cards
- Support for MMCv4.x (MMC Plus)
- Support for 1/4 bit modes

- Auto-detection of card
- DMA for data transfer operations

9.3. Features Not Supported

Support for 8-bit mode of operation.

9.4. Constraints

MMC/SD cards should not be removed when the mount operation is in progress. If done so, data integrity cannot be guaranteed.

9.5. Supported System Calls

`open(),close(),read(),write()`

9.6. Supported IOCTLs

None

9.7. Performance and Benchmarks

Important

The performance numbers can be severely affected if the media is mounted in sync mode. Hot plug scripts in the filesystem mount removable media in sync mode to ensure data integrity. For performance sensitive applications, umount the auto-mounted filesystem and re-mount in async mode.

9.7.1. Performance using EXT2 file system

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	10.18	28.67
250	100	10.17	27.95
500	100	10.09	28.30
1024	100	10.16	28.20
5120	100	10.10	28.28

Table 9.1. Read performance values

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	6.67	24.59
250	100	6.57	26.46
500	100	6.59	26.13
1024	100	6.52	24.63
5120	100	6.57	25.72

Table 9.2. Write performance values

The performance numbers were captured using the following:

- SanDisk Extreme high speed Card
- CPU idle : Enabled
- File System: ext2

9.7.2. Performance using VFAT file system

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	10.11	29.54
250	100	10.12	28.67
500	100	10.12	28.38
1024	100	10.04	28.42
5120	100	10.11	28.21

Table 9.3. Read performance values

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	4.94	21.36
250	100	5.37	27.20
500	100	5.51	26.85
1024	100	5.41	25.85
5120	100	5.49	28.68

Table 9.4. Write performance values

The performance numbers were captured using the following:

- SanDisk Extreme high speed Card

- CPU idle : Enabled
- File System: VFAT

USB Driver

Abstract

This chapter provides details on EHCI and MUSB drivers along with throughput and CPU load numbers.

This chapter describes the USB (EHCI and MUSB) driver architecture, features supported/not supported, constraints and performance numbers.

10.1. EHCI/OHCI Controller

10.1.1. Description

USB EHCI is supported on Mistral daughter card over OMAP35x EVM. It is connected to HS USB port 2 of OMAP35x via SMSC83320 USB Phy.

10.1.2. Driver Features

The driver supports the following feature

1. Human Interface Class (HID) via a high speed hub.
2. Mass Storage Class (MSC)
3. USB isochronous devices (Audio and Video)
4. Hub Class

10.1.3. Features Not Supported

None

10.2. MUSB OTG controller

10.2.1. Description

The MUSB driver is implemented on top of Mentor OTG IP version 1.4 which supports all the speeds (High, Full and Low).MUSB uses inventra DMA for all the transfers.

10.2.2. Driver Features

The driver supports the following feature

Host Mode

1. Human Interface Class (HID)
2. Mass Storage Class (MSC)
3. USB isochronous devices (Audio and Video)

4. Hub Class

Slave Mode

1. Mass Storage Class (MSC)
2. Communication Device Class (CDC) Slave support
3. Remote Network Driver Interface Specification (RNDIS) Slave support

OTG Mode

1. Session Request Protocol (SRP)
2. Host Negotiation Protocol (HNP)

10.2.3. Features Not Supported

None

10.3. USB Mass Storage Class Host Driver

10.3.1. Description

This figure illustrates the stack diagram of the system with USB Mass Storage class.

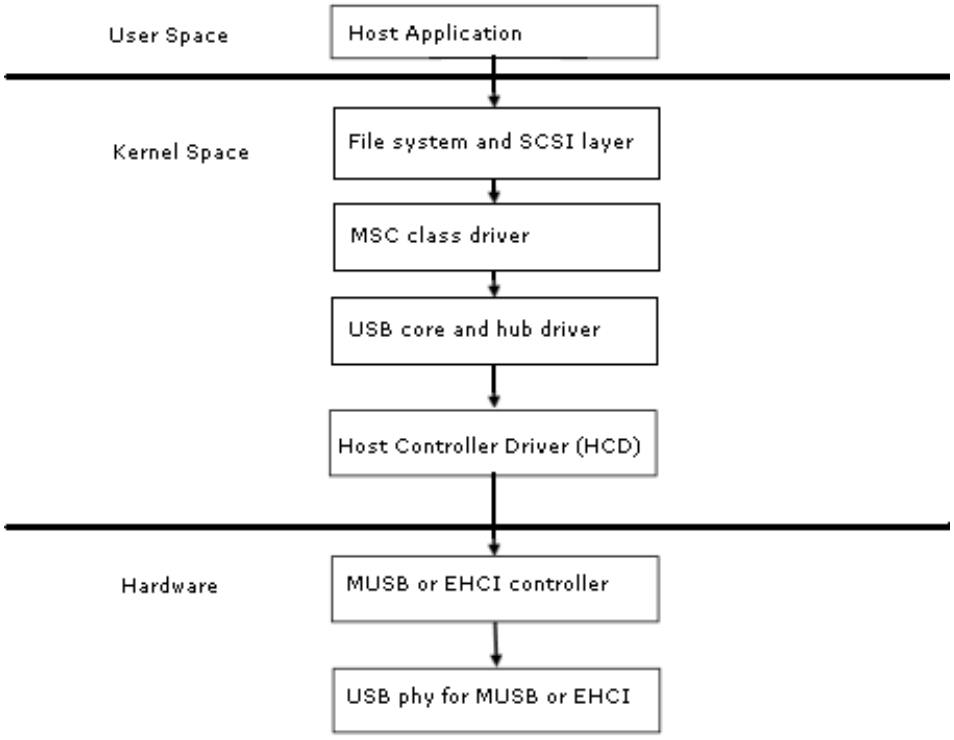

Figure 10.1. USB Driver: Illustration of Mass Storage Class

10.3.2. Driver Features

The driver supports the following feature

1. DMA mode
2. PIO mode

10.3.3. Features Not Supported

None

10.3.4. Constraint

1. There is a limitation in the power that is supplied by the charge pump. If you notice VBUSERR messages in the system console, then connect a self powered USB hub and then attach the device to the hub
2. VBUS must be re-enabled after disconnect and reconnect of the MSC device while the MUSB image is built in OTG mode

10.3.5. Supported System Calls

open(), close(), read(), write(), ioctl()

10.3.6. Supported IOCTLS

None

10.3.7. Performance Benchmarks

Important

For Mass-storage applications, the performance numbers can be severely affected if the media is mounted in sync mode. Hot plug scripts in the filesystem mount removable media in sync mode to ensure data integrity. For performance sensitive applications, umount the auto-mounted filesystem and re-mount in async mode.

10.3.7.1. USB-MSC MUSB Host-DMA-Ext2 Performance

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	8.18	17.56
256	100	9.29	20.90
512	100	8.49	19.04
1024	100	8.67	19.85
5120	100	8.66	19.93

Table 10.1. USB-MSC MUSB Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	22.14	32.21
256	100	22.25	33.40
512	100	21.25	36.71
1024	100	21.54	35.32
5120	100	21.61	36.04

Table 10.2. USB-MSC MUSB Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: Western Digital 5000KS, 465 GB
2. File format: ext2
3. CPU idle : Enabled

10.3.7.2. USB-MSC MUSB Host-DMA-VFAT Performance

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	8.81	21.80
256	100	9.04	22.50
512	100	9.07	21.89
1024	100	8.77	20.67
5120	100	9.17	20.93

Table 10.3. USB-MSC MUSB Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	12.55	29.17
256	100	14.38	33.01
512	100	15.80	38.91
1024	100	15.40	37.81
5120	100	15.87	39.70

Table 10.4. USB-MSC MUSB Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: Western Digital 5000KS, 465 GB
2. File format: vfat
3. CPU idle : Enabled

10.3.7.3. USB-MSC EHCI Host-DMA-Ext2 Performance

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	21.56	47.94
256	100	22.09	48.52
512	100	22.50	48.71
1024	100	22.36	47.97
5120	100	22.33	49.68

Table 10.5. USB-MSC EHCI Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	21.09	29.88
256	100	20.12	28.88
512	100	19.85	33.78
1024	100	20.62	34.19
5120	100	20.01	34.80

Table 10.6. USB-MSC EHCI Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: Western Digital 5000KS, 465 GB
2. File format: ext2
3. CPU idle : Enabled

10.3.7.4. USB-MSC EHCI Host-DMA-VFAT Performance

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	22.53	50.21
256	100	23.13	54.97
512	100	22.95	49.34
1024	100	22.96	50.88
5120	100	22.76	49.57

Table 10.7. USB-MSC EHCI Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	15.37	32.39
256	100	15.10	35.70
512	100	14.97	36.73
1024	100	14.85	39.51
5120	100	15.16	38.92

Table 10.8. USB-MSC EHCI Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: Western Digital 5000KS, 465 GB
2. File format: vfat
3. CPU idle : Enabled

10.4. USB Mass Storage Class Slave Driver

10.4.1. Description

This figure illustrates the stack diagram of the system with USB File Storage Gadget driver.

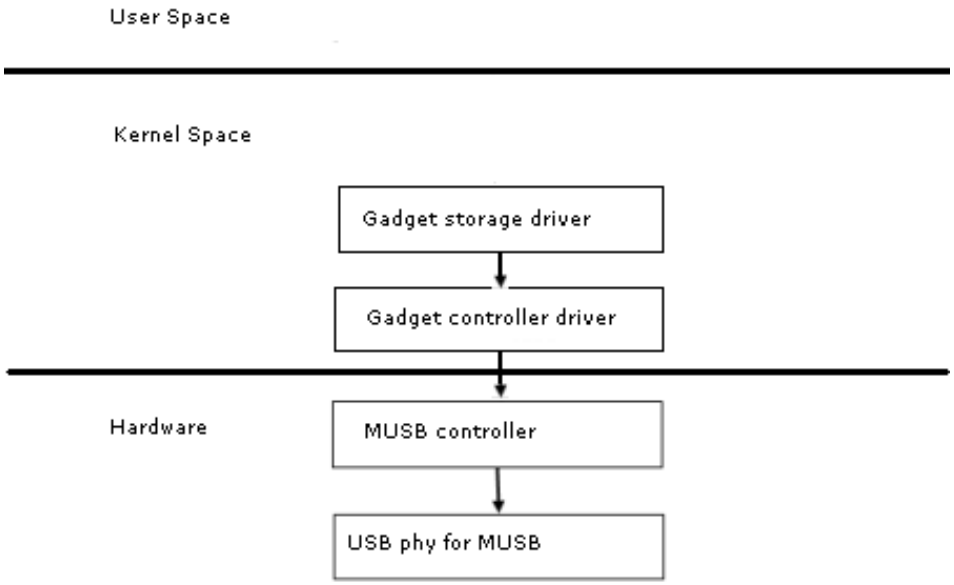

Figure 10.2. USB Driver: Illustration of File Storage Gadget Driver

10.4.2. Driver Features

The driver supports the following feature

1. DMA mode
2. PIO mode
3. File backed storage driver was tested with SD media as the storage medium

10.4.3. Features Not Supported

None

10.4.4. Constraint

None

10.4.5. Supported System Calls

NA

10.4.6. Supported IOCTLS

NA

10.4.7. Performance Benchmarks

10.4.7.1. USB Slave-DMA Performance

Bytes Transferred (MB)	Number of files transferred	Total Bytes transferred (MB)	Data Rate (MB/sec)
50	20	1000	7.95
10	100	1000	7.23
200	5	1000	6.28

Table 10.9. USB Slave-DMA-Read Performance values

Bytes Transferred (MB)	Number of files transferred	Total Bytes transferred (MB)	Data Rate (MB/sec)
50	20	1000	1.79
10	100	1000	2.52
200	5	1000	3.00

Table 10.10. USB Slave-DMA-Write Performance values

The performance numbers are captured using the following.

1. SanDisk SD card 2GB
2. File format: vfat on Windows XP
3. CPU idle : Disabled

10.5. USB CDC/RNDIS Slave Driver

10.5.1. Description

The CDC RNDIS gadget driver that is used to send standard Ethernet frames using USB. The driver will create an Ethernet device by the name usb0.

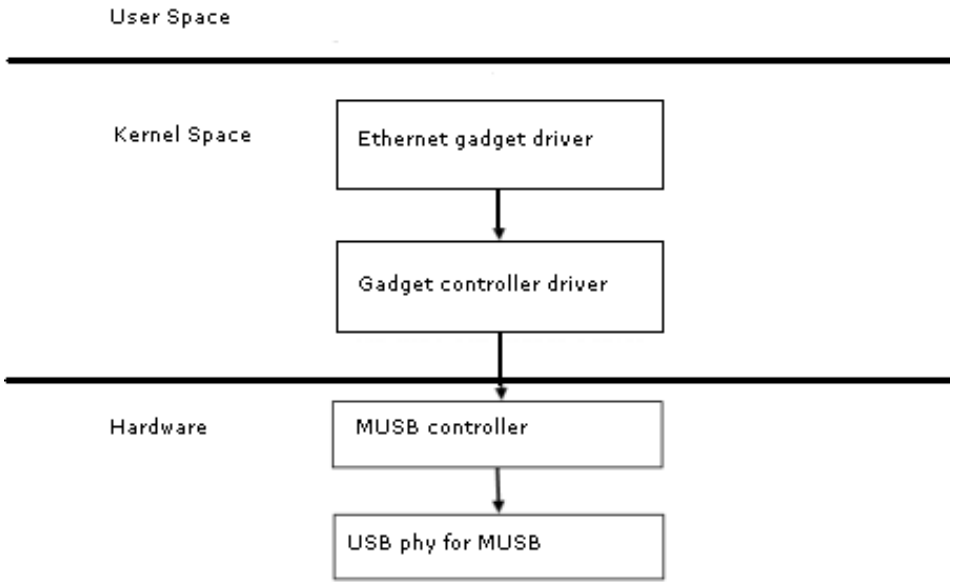

Figure 10.3. USB Driver: Illustration of CDC/RNDIS Gadget Driver

10.5.2. Driver Features

The driver supports the following feature

1. DMA mode
2. PIO mode
3. 10/100 Mbps speed.

10.5.3. Features Not Supported

None

10.5.4. Constraint

None

10.5.5. Supported System Calls

open(), close(), read(), write(), ioctl()

10.5.6. Supported IOCTLS

None

10.5.7. Performance Benchmarks

10.5.7.1. USB CDC-DMA Performance

TCP window size (in KBytes)	Bandwidth (Mbits/Sec)	Transfer size (in Mbytes)	Interval (in sec)
16	54.8	392	60
32	54.6	391	60
64	54.7	391	60
128	53.5	383	60
208	45.9	328	60

Table 10.11. USB CDC-DMA Performance values

10.5.7.2. USB RNDIS-DMA Performance

TCP window size (in KBytes)	Bandwidth (Mbits/Sec)	Transfer size (in Mbytes)	Interval (in sec)
16	60.5	433	60
32	55.0	393	60
64	55.3	396	60
128	46.8	336	60
208	46.7	234	60

Table 10.12. USB RNDIS-DMA Performance values

10.6. USB Human Interface Device (HID) Driver

10.6.1. Description

The event sub system creates /dev/input/event* devices with the help of mdev.

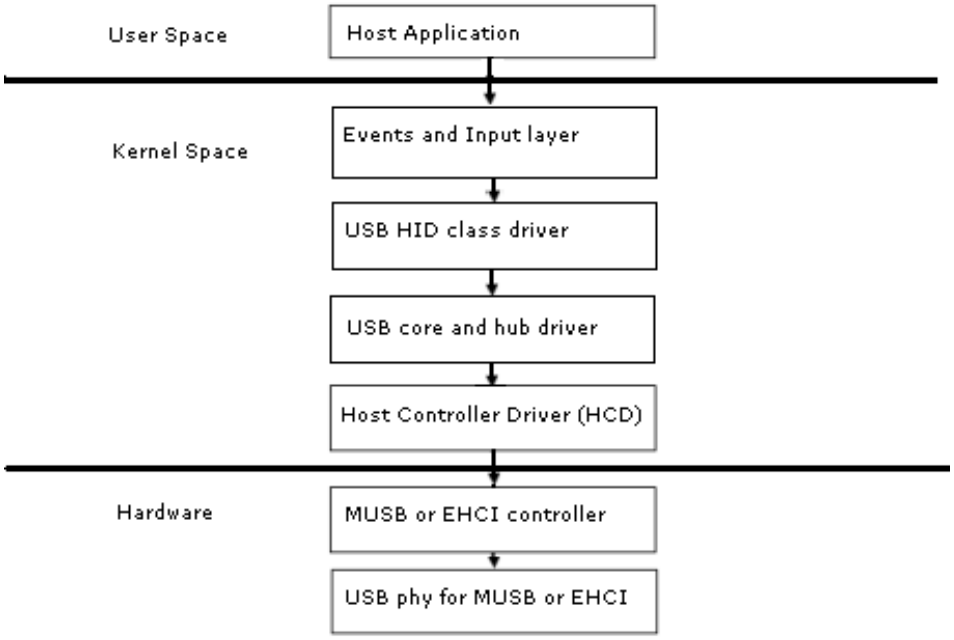

Figure 10.4. USB Driver: Illustration of HID Driver

10.6.2. Driver Features

The driver supports the following feature

1. DMA mode
2. PIO mode
3. USB Mouse and Keyboards that conform to the USB HID specifications

10.6.3. Features Not Supported

None

10.6.4. Constraint

None

10.6.5. Supported System Calls

NA

10.6.6. Supported IOCTLS

NA

10.6.7. Performance Benchmarks

NA

10.7. USB Isochronous Driver

10.7.1. Description

USB camera, speaker and mic uses isochronouse transfers.USB Video Class (UVC) is used by most of the USB cameras to capture image.

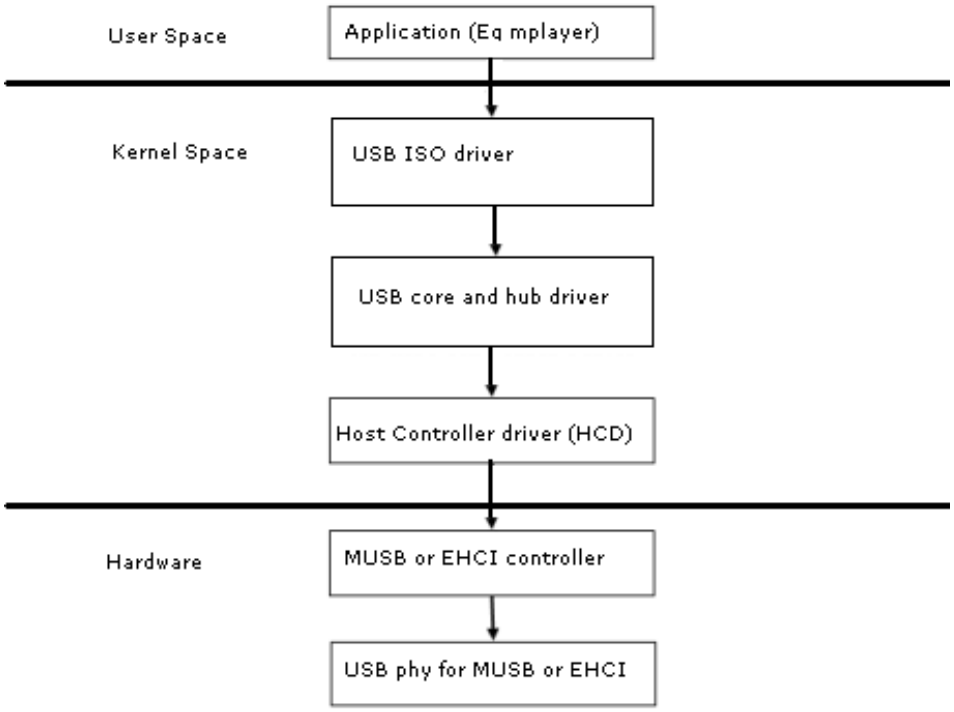

Figure 10.5. USB Driver: Illustration of ISO device Driver

10.7.2. Driver Features

The driver supports the following feature

1. DMA mode
2. PIO mode
3. Support for USB Audio and video class(UVC class)
4. Support for high bandwidth isochronous transfer

10.7.3. Features Not Supported

None

10.7.4. Constraint

None

10.7.5. Supported System Calls

NA

10.7.6. Supported IOCTLS

NA

10.7.7. Performance Benchmarks

NA

10.8. USB OTG Driver

10.8.1. Description

MUSB controller on OMAP supports USB On The Go (OTG).OTG protocol enables runtime role switch between USB host and device.This is achived using Session Request Protocol (SRP) and Host Negotiation Protocol (HNP).OTG driver is tested with OPT (OTG Protocol Tester) and details are available in Release Notes.

10.8.2. Driver Features

The driver supports the following feature

1. Both HNP and SRP are supported

10.8.3. Features Not Supported

None

10.8.4. Constraint

None

10.8.5. Supported System Calls

NA

10.8.6. Supported IOCTLS

NA

10.8.7. Performance Benchmarks

NA