AM3517 Linux PSP

Datasheet

03.00.00.02

Publication date 30 September 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 AM3517 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.

AM3517 Linux Drivers

Abstract

This chapter provides brief description on the device drivers supported in the Linux PSP release for AM3517 platform.

1.1. Texas Instruments AM3517 Linux Drivers

This chapter provides a brief description of the device drivers supported on AM3517 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.

Note

This is a preliminary draft of the Datasheet that describes the features supported by the individual drivers. Actual performance numbers will be updated after the completion of system test cycle.

1.1.1. AM3517 Linux PSP details

1. Built with linux kernel version 2.6.31-rc7
2. Compiled with code sourcery tool chain version arm-2009-q1.
3. Based on U-Boot version 2009.03.
4. Supports boot from Nand and MMC/SD.
5. 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

Peripheral	Description	Linux driver type
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)	USB Mass Storage Class Host Driver	Block
USB 2.0 MSC Slave	USB Mass Storage Class Slave Driver	USB Gadget
USB EHCI Host Controller	USB EHCI Host Controller	USB Host Controller Driver
Micron Nand	Flash storage system	MTD Character and Block
MMC/SD	Interface to MultiMedia Secure Digital cards	Block
UART	Serial Communication Interface	Character
Power Management	CPUIdle and dynamic tick.	CPUidle, dyntick
I2C	Inter-IC Communication	Character

Table 1.1. AM3517 Peripheral Driver Support

1.1.3. Drivers not supported

1. Keypad driver
2. Touchscreen driver

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).

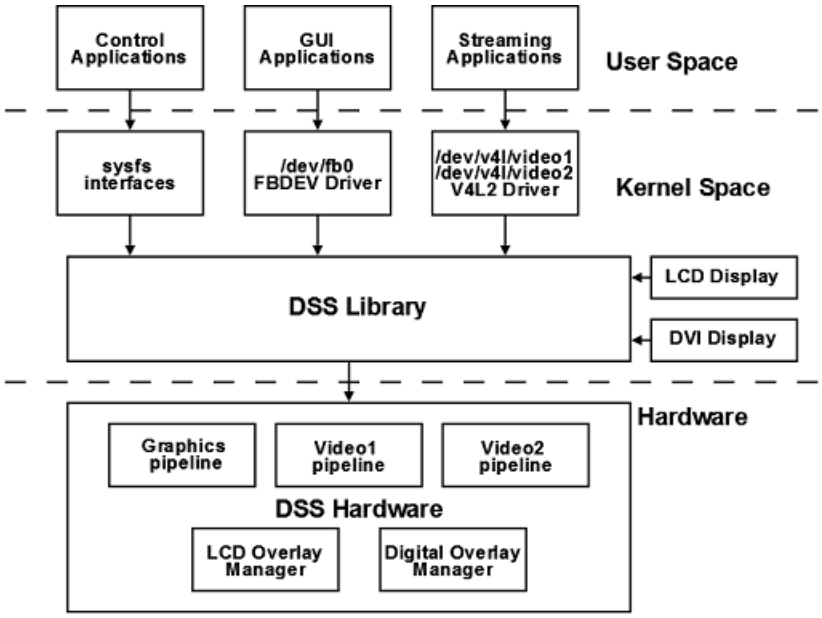

Figure 2.1. OMAP35x Display Subsystem Architecture

2.2. Features

Pixel formats supported on video plane are YUV, UYVY, 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 (Both S-Video out and composite out is 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, configurable resolution through SYSFS or bootime.
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 Alpha blending, both global alpha and pixel alpha.
12. Supports Source and Destination color keying through V4L2 ioctls.
13. Mirroring not supported.

2.3. Features not supported

1. For RGB888 rotation is not supported because of hardware limitation.
2. Linking feature not supported on video pipelines.
3. 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.

IOCTL Command Name	Description
`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), 4.3" WQVGA LCD Panel (LQ043T1DG01)
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	WQVGA	62	0
LCD	V4L2	WQVGA	62	0
TV	V4L2	NTSC	NA	NA
TV	V4L2	PAL	25	0
TV	Fbdev	NTSC	NA	NA
TV	Fbdev	PAL	25	1
DVI	Fbdev	VGA	NA	NA
DVI	V4L2	VGA	NA	NA

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 module is a key component for still-image capture.

It provides the system interface and the processing capability to connect RAW image-sensor modules and video decoders to the AM3517 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 -

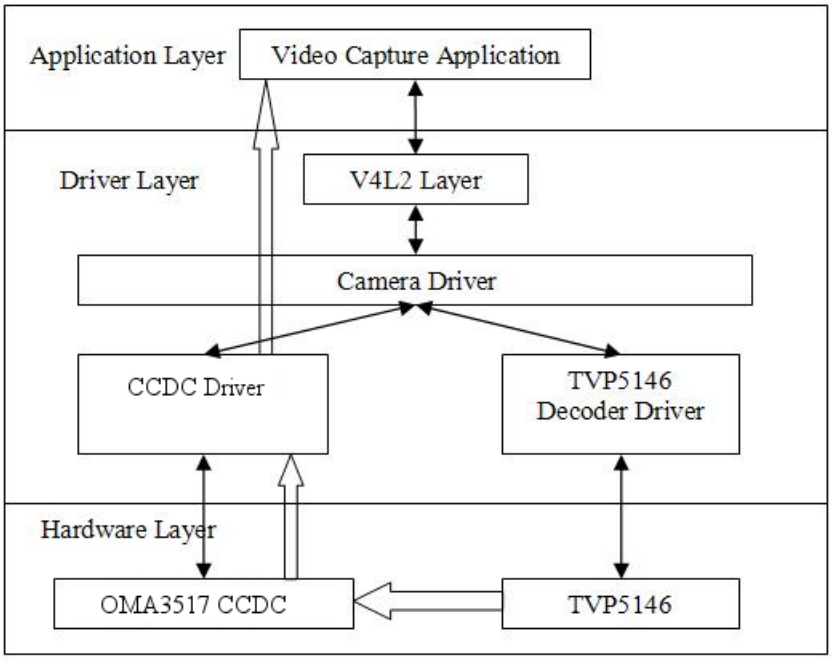

Figure 3.1. AM3517 Capture Interface Block-Diagram

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 10-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. Video quality issues observed, greenish image is observed.
4. Field id is not coming properly, this results in flickering of image.

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)

Resource	Description
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	8-25	1
PAL	8-25	1

Table 3.3. Capture Performance

ALSA SoC Audio Driver

Abstract

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

4.1. Introduction

AM3517 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:

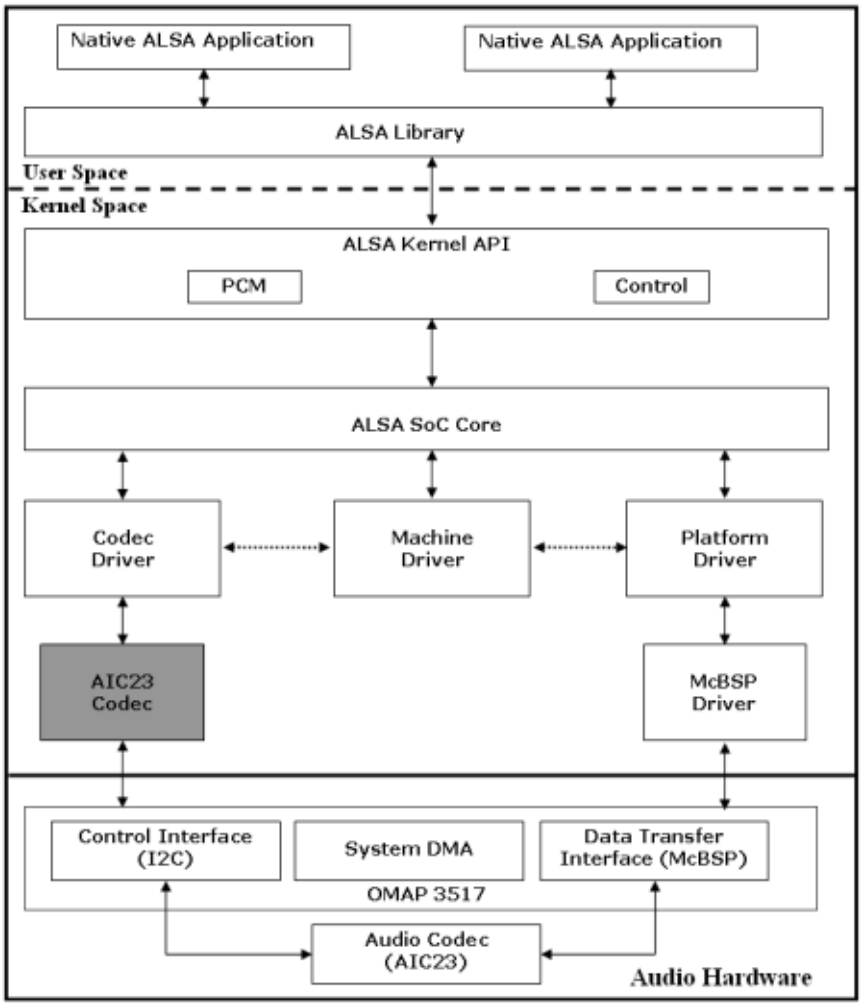

Figure 4.1. ALSA SoC Architecture

4.2. Driver Features

The driver supports the following features:

- Supports AIC23 audio codec in ALSA SoC framework.
- Supports multiple sample rates (8KHz, 16KHz, 22.05KHz, 32KHz, 44.1KHz, 48KHz, 64KHz, 88.2KHz and 96KHz) for both playback and capture.
- 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 AIC23 audio codec.

4.3. Features Not Supported

None

4.4. 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.
- By default, AIC23 codec is configured in master mode and McBSP is used as slave. Testing of the audio sub-system is done in this configuration only.
- OSS emulation layer is not supported because of which OSS based applications (for e.g. madplay) may not work properly.
- Configuration of playback and capture streams in different sampling rates is not supported.

4.5. Supported System Calls

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

4.6. Supported IOCTLs

NA

4.7. 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
44100	1	1
48000	1	1

Table 4.1. Audio Write Performance

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

Table 4.2. Audio Read Performance

Power Management

Abstract

AM3517 and AM3505 do not support any of the power management features traditionally supported by the OMAP35X family of devices.

Ethernet Driver

Abstract

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

6.1. Introduction

The ethernet driver supports the linux netdev interface. The MAC module interfaces to the PHY on the board using RMII interface. The driver uses the MDIO serial interface to program/communicate with the PHY

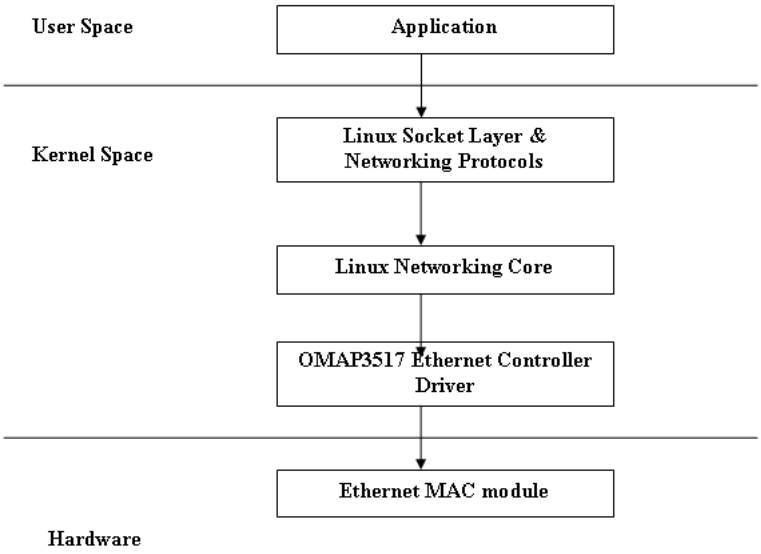

Figure 6.1. Linux Kernel Ethernet Driver

6.2. 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.
- Linux NAPI support for interrupt load mitigation.

6.3. Features Not Supported

- None

6.4. Constraints

- None

6.5. Supported System Calls

Supports socket/read/write family of system calls that can be exercised on any network device

6.6. Supported IOCTLs

NA

6.7. Performance and Benchmarks

TCP Window Size (in KBytes)	Bandwidth(Tx +Rx) (in Mbits/sec)	Transfer size (in MBytes)	Interval (in Seconds)
16	10.24	73.2	60
32	18.43	131.7	60
64	37.40	267.0	60
128	49.70	356.0	60

Table 6.1. 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 DUT1 in server mode and on DUT2 in client mode. Version 1.7.0 is used on both sides.
- Data captured here is for "iperf" in client mode.
- Cross cable is used to measure performance.
- Speed is set to 100Mbps

Flash Driver

Abstract

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

7.1. 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 Micron NAND devices is supported by the driver.

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

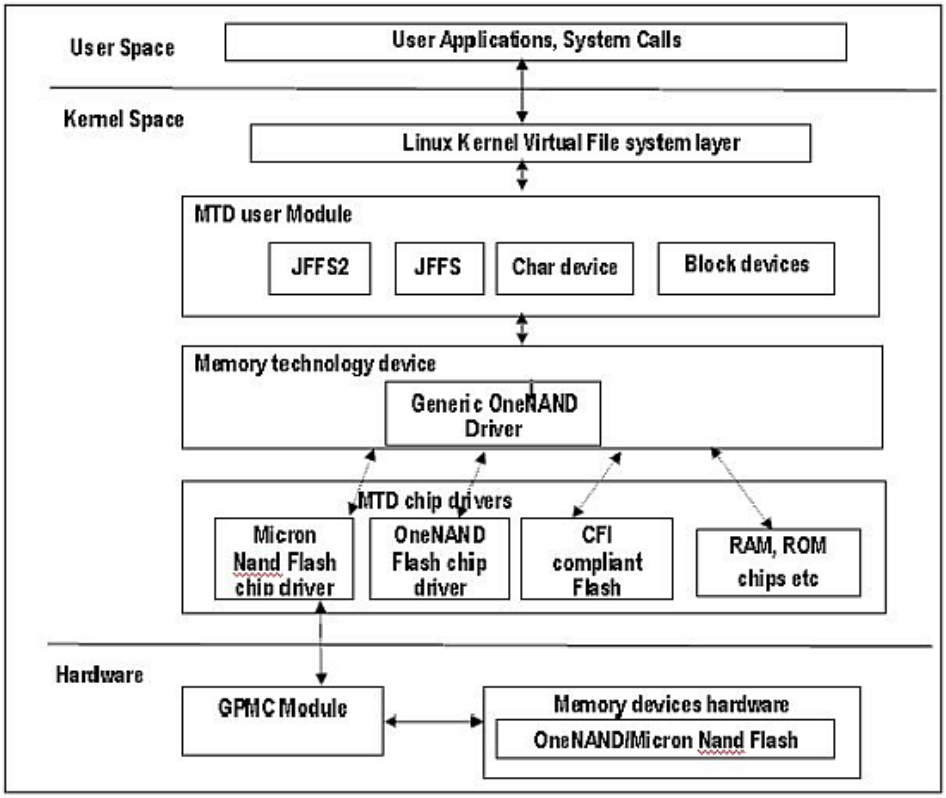

Figure 7.1. Linux Kernel NAND driver

7.2. Driver Features

The driver supports the following features:

1. JFFS2 file system
2. Supports Read/Write, Erase, Unlock Operations and BBM
3. Nand Flash 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)

7.3. Features Not Supported

None

7.4. Constraints

None

7.5. Supported System Calls

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

7.6. Performance Benchmarks

7.6.1. 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	4.81	99.95
256	100	3.27	100.00
512	100	2.82	100.00
1024	100	2.47	100.00
5120	100	2.20	100.00

Table 7.1. Micron Nand write performance values

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	6.55	100.00
256	100	5.90	100.00
512	100	5.91	100.00
1024	100	5.91	100.00
5120	100	5.90	100.00

Table 7.2. Micron Nand read performance values

The performance numbers are captured using the following:

1. Micron NAND PART Number: Micron MT29F4G16ABCHC
2. File Format = JFFS2

MMC/SD Driver

Abstract

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

8.1. 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/mmcblockp1, /dev/mmcblockp2`) are created for user space access.

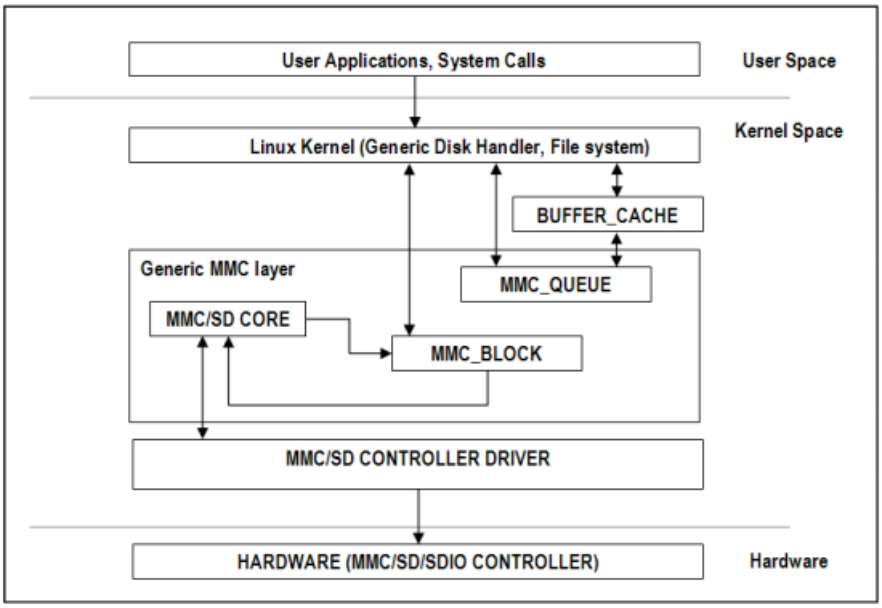

Figure 8.1. Linux Kernel MMC Driver

8.2. 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

8.3. Features Not Supported

Support for 8-bit mode of operation.

8.4. Constraints

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

8.5. Supported System Calls

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

8.6. Supported IOCTLs

None

8.7. Performance and Benchmarks

Important

The performance numbers can be severely affected if the media is mounted in sync mode. For performance sensitive applications, umount the auto-mounted filesystem and re-mount in async mode.

8.7.1. Performance using EXT2 file system

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	12.25	14.62
250	100	12.15	15.08
500	100	12.42	15.62
1024	100	12.24	15.30
5120	100	12.37	14.30

Table 8.1. Read performance values

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	17.66	19.36
250	100	17.78	22.07
500	100	17.50	20.87
1024	100	18.27	21.99
5120	100	17.17	21.44

Table 8.2. Write performance values

The performance numbers were captured using the following:

- SD Card - Patriot 8GB Class 6 card
- File System: ext2 mounted in async mode

8.7.2. Performance using VFAT file system

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	11.54	10.82
250	100	11.49	11.10
500	100	11.49	10.32
1024	100	11.50	11.21
5120	100	11.48	11.18

Table 8.3. Read performance values

Buffer Size (in KBytes)	Total Bytes Transferred (in MBytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
100	100	9.08	17.95
250	100	8.88	18.05
500	100	9.15	18.50
1024	100	9.20	18.72
5120	100	8.73	17.65

Table 8.4. Write performance values

The performance numbers were captured using the following:

- SD Card : Patriot 8GB Class 6
- File System: VFAT mounted in async mode

USB Driver

Abstract

This chapter provides details on MUSB driver 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.

9.1. EHCI/OHCI Controller

9.1.1. Description

There are two EHCI ports on AM3517, one on baseboard which is connected to HSUSB port-1 and another on UI card which is connected to HSUSB port-2.EHCI on UI card (port-2) will be functional only if LCD is not enabled. This limitation is due to shared pins.

9.1.2. 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

9.1.3. Features Not Supported

None

9.2. MUSB OTG controller

9.2.1. Description

The MUSB driver is implemented on top of Mentor OTG IP version 1.9 which supports all the speeds (High, Full and Low).MUSB uses interrupt mode (non-DMA) or CPPI4.1 DMA for all the transfers.

9.2.2. 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)

9.2.3. Features Not Supported

None

9.3. USB Mass Storage Class Host Driver

9.3.1. Description

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

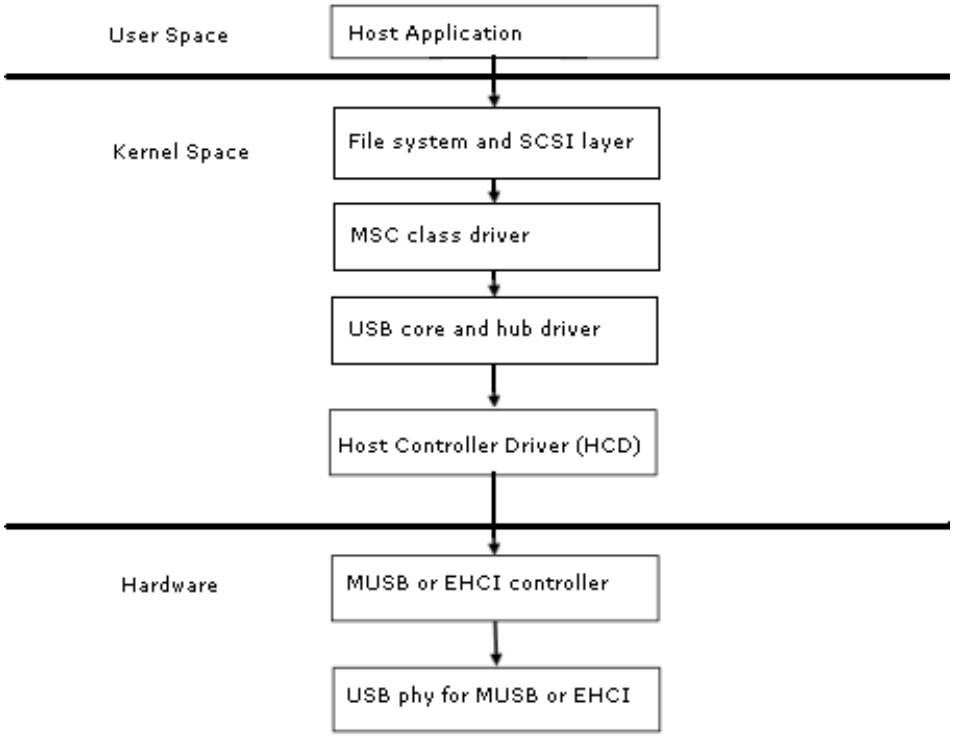

Figure 9.1. USB Driver: Illustration of Mass Storage Class

9.3.2. Driver Features

The driver supports the following feature

1. DMA mode
2. PIO mode

9.3.3. Features Not Supported

None

9.3.4. Constraint

1. VBUS must be re-enabled after disconnect and reconnect of the MSC device while the MUSB image is built in OTG mode

9.3.5. Supported System Calls

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

9.3.6. Supported IOCTLS

None

9.3.7. 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.

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

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	21.54	51.38
256	100	20.82	56.47
512	100	20.70	54.65
1024	100	21.11	54.12
5120	100	20.05	57.79

Table 9.1. USB-MSC MUSB Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	24.62	33.49
256	100	24.66	35.45
512	100	24.66	31.13
1024	100	24.63	35.53
5120	100	24.68	37.18

Table 9.2. USB-MSC MUSB Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: IOMEGA, 120GB
2. File format: EXT2
3. Mount mode: Async
4. CPU IDLE state: Disabled

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

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	21.99	35.29
256	100	21.72	33.47
512	100	21.99	33.89
1024	100	21.92	33.54
5120	100	21.89	32.08

Table 9.3. USB-MSC MUSB Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	21.59	43.60
256	100	21.86	48.23
512	100	22.29	48.30
1024	100	22.57	50.11
5120	100	21.23	43.41

Table 9.4. USB-MSC MUSB Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: IOMEGA, 120GB
2. File format: VFAT
3. Mount mode: Async
4. CPU IDLE state: Disabled

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

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	12.06	18.19
256	100	12.06	18.14
512	100	12.06	18.37

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
1024	100	12.06	19.10
5120	100	12.08	19.16

Table 9.5. USB-MSC EHCI Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	10.87	12.33
256	100	10.85	12.97
512	100	10.73	12.70
1024	100	10.68	13.10
5120	100	10.53	12.95

Table 9.6. USB-MSC EHCI Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: IOMEGA, 120GB
2. File format: EXT2
3. Mount mode: Async
4. CPU IDLE state: Disabled

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

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	11.66	22.04
256	100	11.66	21.14
512	100	11.66	20.92
1024	100	11.67	19.82
5120	100	11.66	20.74

Table 9.7. USB-MSC EHCI Host-DMA-Read Performance values

Buffer Size (KByte)	Total Bytes Transferred (MB)	Transfer Rate (MB/s)	CPU Load (in %)
100	100	11.52	15.87
256	100	11.23	15.10
512	100	12.33	15.22
1024	100	11.61	16.17
5120	100	11.19	15.08

Table 9.8. USB-MSC EHCI Host-DMA-Write Performance values

The performance numbers are captured using the following.

1. Hard disk: IOMEGA, 120GB
2. File format: EXT2
3. Mount mode: Async
4. CPU IDLE state: Disabled

9.4. USB Mass Storage Class Slave Driver

9.4.1. Description

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

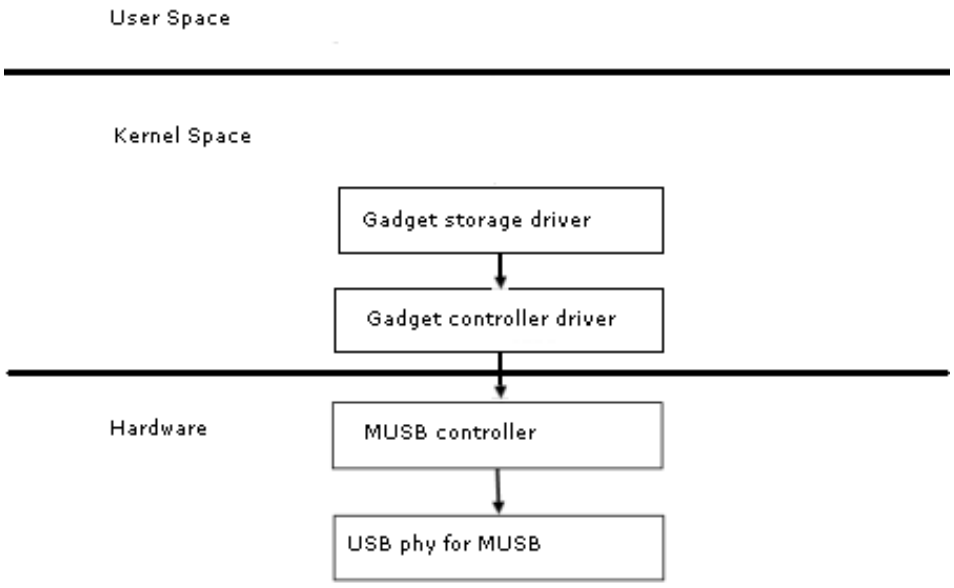

Figure 9.2. USB Driver: Illustration of File Storage Gadget Driver

9.4.2. 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

9.4.3. Features Not Supported

None

9.4.4. Constraint

None

9.4.5. Supported System Calls

NA

9.4.6. Supported IOCTLS

NA

9.4.7. Performance Benchmarks

9.4.7.1. USB Slave-DMA Performance

Bytes Transferred (MB)	Number of files transferred	Total Bytes transferred (MB)	Data Rate (MB/sec)
50	20	1000	14.62
10	100	1000	13.65
200	5	1000	14.62

Table 9.9. USB Slave-DMA-Read Performance values

Bytes Transferred (MB)	Number of files transferred	Total Bytes transferred (MB)	Data Rate (MB/sec)
50	20	1000	NA
10	1	10	0.153
200	5	1000	NA

Table 9.10. USB Slave-DMA-Write Performance values

The performance numbers are very low as there are some print messages coming continuously.This issue will be fixed in next release

The performance numbers are captured using the following.

1. USB HDD (Toshiba, 80GB) connected to EHCI port is used as storage media
2. File format: VFAT on Windows XP
3. CPU idle : Disabled

9.5. USB CDC/RNDIS Slave Driver

9.5.1. 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.

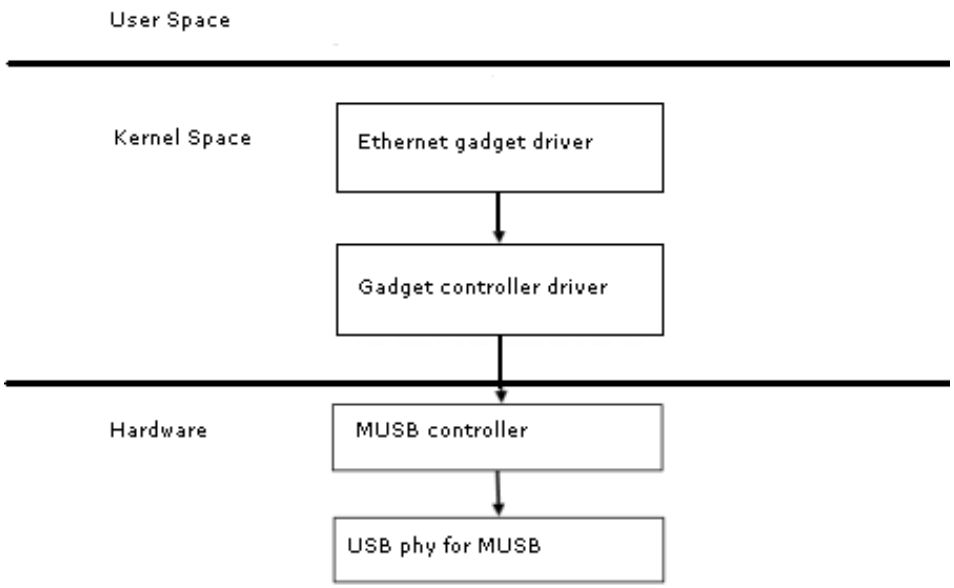

Figure 9.3. USB Driver: Illustration of CDC/RNDIS Gadget Driver

9.5.2. Driver Features

The driver supports the following feature

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

9.5.3. Features Not Supported

None

9.5.4. Constraint

None

9.5.5. Supported System Calls

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

9.5.6. Supported IOCTLS

None

9.5.7. Performance Benchmarks

9.5.7.1. USB CDC-DMA Performance

TCP window size (in KBytes)	Bandwidth (Mbits/Sec)	Transfer size (in Mbytes)	Interval (in sec)
16	NA	NA	60
32	NA	NA	60
64	NA	NA	60
128	NA	NA	60
208	NA	NA	60

Table 9.11. USB CDC-DMA Performance values

9.5.7.2. USB RNDIS-DMA Performance

TCP window size (in KBytes)	Bandwidth (Mbits/Sec)	Transfer size (in Mbytes)	Interval (in sec)
16	NA	NA	60
32	NA	NA	60
64	NA	NA	60
128	NA	NA	60
208	NA	NA	60

Table 9.12. USB RNDIS-DMA Performance values

9.6. USB Human Interface Device (HID) Driver

9.6.1. Description

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

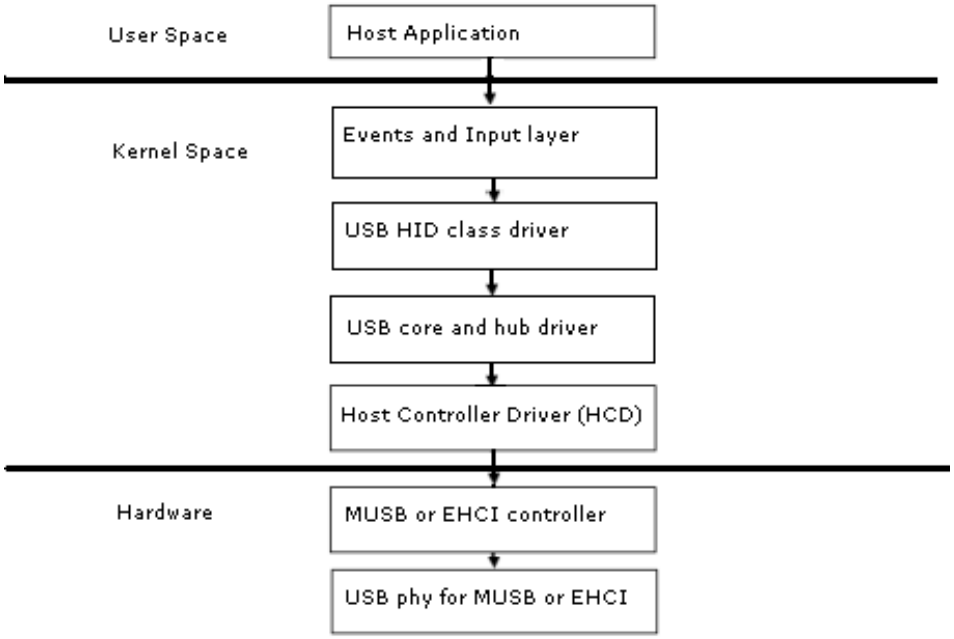

Figure 9.4. USB Driver: Illustration of HID Driver

9.6.2. 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

9.6.3. Features Not Supported

None

9.6.4. Constraint

None

9.6.5. Supported System Calls

NA

9.6.6. Supported IOCTLS

NA

9.6.7. Performance Benchmarks

NA

9.7. USB Isochronous Driver

9.7.1. Description

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

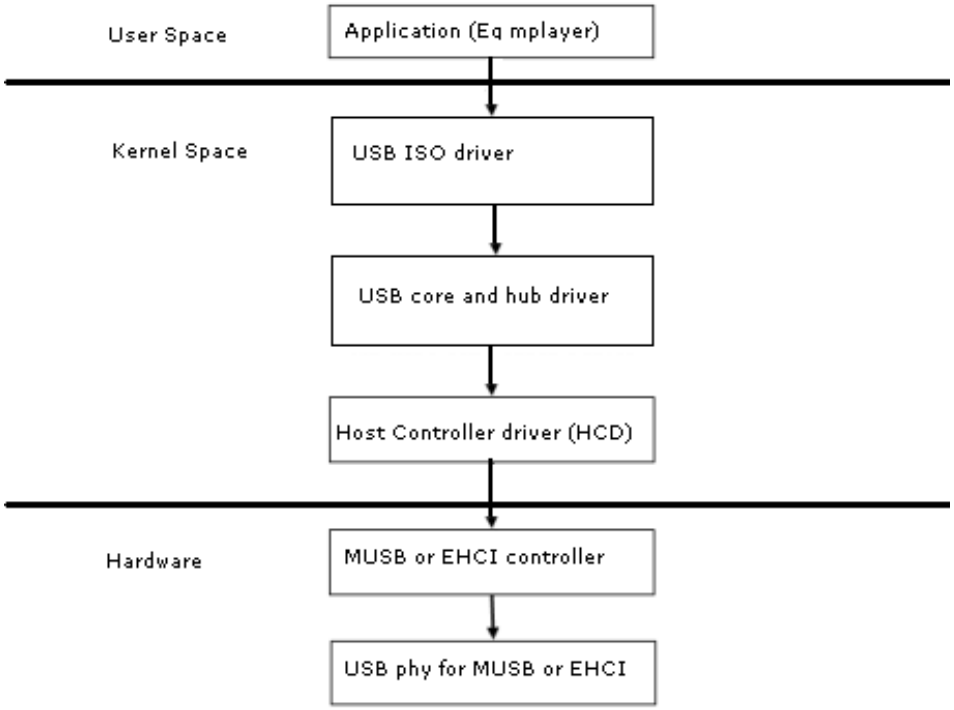

Figure 9.5. USB Driver: Illustration of ISO device Driver

9.7.2. 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

9.7.3. Features Not Supported

None

9.7.4. Constraint

None

9.7.5. Supported System Calls

NA

9.7.6. Supported IOCTLS

NA

9.7.7. Performance Benchmarks

NA

9.8. USB OTG Driver

9.8.1. Description

MUSB controller on AM3517EVM 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.

9.8.2. Driver Features

The driver supports the following feature

1. Both HNP and SRP are supported

9.8.3. Features Not Supported

None

9.8.4. Constraint

None

9.8.5. Supported System Calls

NA

9.8.6. Supported IOCTLS

NA

9.8.7. Performance Benchmarks

NA