



Navigation

[Main Page](#)
[All pages](#)
[All categories](#)
[Popular pages](#)
[Popular authors](#)
[Popular categories](#)
[Category stats](#)
[Recent changes](#)
[Random page](#)
[Help](#)
[Google Search](#)

Print/export

[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Toolbox

[What links here](#)
[Related changes](#)
[Special pages](#)
[Permanent link](#)
[Browse properties](#)

Page [Discussion](#) [ce](#) [View history](#)

AM335x-PSP 04.06.00.08 Features and Performance Guide

AM335x-PSP

04.06.00.08

Features and Performance Guide

Translate this page to [cs](#) - [Česky](#)

Google™ Custom Search



AM335x-PSP 04.06.00.08 Features and Performance Guide

Linux PSP

Contents [\[hide\]](#)

- 1 Read This First
 - 1.1 About This Manual
 - 1.2 If You Need Assistance
- 2 Support Overview
 - 2.1 Boot Modes Supported
 - 2.2 U-Boot Support
- 3 Linux Kernel
 - 3.1 Kernel Virtual Memory Layout
 - 3.2 Interrupt Latency Measurement
 - 3.3 Boot-time Measurement
- 4 PSP Linux Drivers
 - 4.1 Device Driver List
 - 4.2 ALSA SoC Audio Driver
 - 4.2.1 Introduction
 - 4.2.2 Driver Features
 - 4.2.3 Features Not Supported
 - 4.2.4 Constraints
 - 4.2.5 Supported System Calls
 - 4.2.6 Performance and Benchmarks
 - 4.3 Ethernet Driver
 - 4.3.1 Introduction
 - 4.3.2 Driver Features
 - 4.3.3 Features Not Supported
 - 4.3.4 Supported System Calls
 - 4.3.5 Performance and Benchmarks
 - 4.3.5.1 TCP Performance
 - 4.3.5.2 UDP Performance
 - 4.4 Graphical LCD (GLCD) Driver
 - 4.4.1 Introduction
 - 4.4.2 Driver Features
 - 4.4.3 Features Not Supported
 - 4.4.4 Constraints
 - 4.4.5 Supported System Calls
 - 4.4.6 Performance Benchmarks
 - 4.5 NAND Driver
 - 4.5.1 Introduction
 - 4.5.2 Driver Features
 - 4.5.3 Features Not Supported
 - 4.5.4 Supported System Calls
 - 4.5.5 Performance Benchmarks
 - 4.6 SPI Flash Driver
 - 4.6.1 Introduction
 - 4.6.2 Driver Features
 - 4.6.3 Supported System Calls
 - 4.6.4 Performance Benchmarks
 - 4.7 MMC/SD Driver
 - 4.7.1 Introduction
 - 4.7.2 Driver Features
 - 4.7.3 Features Not Supported
 - 4.7.4 Constraints
 - 4.7.5 Supported System Calls

4.7.6 Performance and Benchmarks
4.7.6.1 EXT2 file system
4.7.6.2 VFAT file system
4.8 UART Driver
4.8.1 Introduction
4.8.2 Features Not Supported
4.8.3 Supported System Calls
4.8.4 Supported IOCTLs
4.8.5 Performance and Benchmarks
4.9 I2C Driver
4.9.1 Introduction
4.9.2 Driver Features
4.9.3 Features Not Supported
4.9.4 Supported System Calls
4.9.5 Supported IOCTLs
4.9.6 Performance and Benchmarks
4.10 EDMA Driver
4.10.1 Introduction
4.10.2 Driver Features
4.10.3 Features Not Supported
4.11 Touchscreen Driver
4.11.1 Introduction
4.11.2 Driver Features
4.11.3 Features Not Supported
4.12 Watchdog(WDT) Driver
4.12.1 Introduction
4.12.2 Driver Features
4.12.3 Supported System Calls
4.12.4 Supported IOCTLs
4.13 USB Driver
4.13.1 MUSB OTG controller
4.13.1.1 Description
4.13.1.2 Driver Features
4.13.1.3 Features Not Supported
4.13.2 USB Mass Storage Class Host Driver
4.13.2.1 Driver Features
4.13.2.2 Supported System Calls
4.13.2.3 Performance Benchmarks
4.13.2.3.1 USB MSC (MUSB) Host mode EXT2 File System Performance
4.13.2.3.2 USB MSC (MUSB) Host mode VFAT File System Performance
4.13.3 USB Mass Storage Class Slave Driver
4.13.3.1 Description
4.13.3.2 Driver Features
4.13.3.3 Performance Benchmarks
4.13.4 USB CDC/RNDIS Slave Driver
4.13.4.1 Description
4.13.4.2 Driver Features
4.13.4.3 Supported System Calls
4.13.4.4 Performance Benchmarks
4.13.4.4.1 USB CDC-DMA Performance
4.13.4.4.2 USB RNDIS-DMA Performance
4.13.5 USB Human Interface Device (HID) Driver
4.13.5.1 Description
4.13.5.2 Driver Features
4.14 DCAN Driver
4.14.1 Introduction
4.14.2 Driver Features
4.14.3 Features Not Supported
4.14.4 Supported System Calls
4.15 Power Management
4.15.1 Introduction
4.15.1.1 cpuidle
4.15.1.2 cpufreq
4.15.2 Power Measurements
4.15.2.1 At each OPP
4.15.2.2 When suspended

Read This First

All performance numbers provided in this document are gathered using AM335x Evaluation Module with ARM running at 720MHz and DDR2 configured at 266MHz unless otherwise specified.

About This Manual

This document provides a feature overview and performance data for each of the device drivers which are part of the Linux PSP package. This document should be used in conjunction with release notes provided with the PSP package for information on specific issues present

with drivers included in a particular release.

If You Need Assistance

For further information or to report any problems, contact <http://community.ti.com/> or <http://support.ti.com/>

Support Overview

Boot Modes Supported

Green colored box in the table below means that the particular boot mode is supported on the device in the release.

AM335x Supported Boot Modes			
Boot Mode	AM335x BeagleBone	AM335x Evaluation Module (EVM)	AM335x Evaluation Module-Starter Kit (EVM-SK)
SPI EEPROM	No	No	No
SPI Flash	No	Yes	No
NAND Flash	No	Yes	No
NOR Flash	No	No	No
I2C EEPROM	No	No	No
MMC/SD	Yes	Yes	Yes
EMAC Boot	No	Yes	No
UART Boot	Yes (UART0)	Yes (UART0)	Yes (UART0)

NOTE

These are supported boot modes in PSP software, the actual hardware may support many more boot modes than shown here. Please refer to hardware documentation for list of all supported boot modes.

U-Boot Support

U-Boot is the defacto bootloader for Linux kernel on ARM. The following features of U-Boot are supported in this release.

U-Boot supported feature table			
Feature	AM335x (BeagleBone)	AM335x (Evaluation Module)	AM335x Evaluation Module-Starter Kit (EVM-SK)
UART	Yes	Yes	Yes
Ethernet Download (TFTP)	Yes	Yes	Yes
USB DFU	No	No	No
MMC/SD	Yes	Yes	Yes
SPI Flash	No	Yes	No
NAND flash	No	Yes	No
USB Mass Storage	No	No	No

Linux Kernel

Kernel Virtual Memory Layout

The default AM335x kernel configuration, with "mem=256M" passed as boot argument, uses following Virtual Memory layout:

```
[ 0.000000] Memory: 256MB = 256MB total
[ 0.000000] Memory: 253436k/253436k available, 8708k reserved, 0K highmem
[ 0.000000] Virtual kernel memory layout:
[ 0.000000] vector : 0xffff0000 - 0xffff1000 ( 4 kB)
```

[0.000000]	fixmap	:	0xffff00000	-	0xfffe0000	(896	kB)
[0.000000]	vmalloc	:	0xd0000000	-	0xff000000	(744	MB)
[0.000000]	lowmem	:	0xc0000000	-	0xd0000000	(256	MB)
[0.000000]	modules	:	0xbf000000	-	0xc0000000	(16	MB)
[0.000000]	.text	:	0xc0000000	-	0xc057c000	(5584	kB)
[0.000000]	.init	:	0xc057c000	-	0xc05b9000	(244	kB)
[0.000000]	.data	:	0xc05ba000	-	0xc0617fc0	(376	kB)
[0.000000]	.bss	:	0xc0617fe4	-	0xc0644d64	(180	kB)

Interrupt Latency Measurement

The following table gives interrupt latency measurements taken across 1001 measurement samples.

Interrupt Latency			
Interrupt seconds)	Latency (in micro	Number	of
10		25	
11		265	
12		531	
13		167	
14		11	
15		2	

Boot-time Measurement

Boot-time measurement was done with kernel built using am335x_evm_defconfig. U-Boot environment variable bootargs is set to 'console=tty00,115200n8 quiet lpj=3590144 root=/dev/mmcblk0p2 rw rootfstype=ext3 rootwait ip=none'. U-Boot bootdelay variable is set to 0. Time from reset to "Login:" prompt was measured to be 16.063s.

Complete boot log with timestamps follows:

```
[Tue Jul 31 11:35:39.422 2012]
[Tue Jul 31 11:35:39.422 2012] U-Boot SPL 2011.09 (Jul 20 2012 - 14:58:05)
[Tue Jul 31 11:35:39.422 2012] Texas Instruments Revision detection unimplemented
[Tue Jul 31 11:35:39.907 2012] Found a daughter card connected
[Tue Jul 31 11:35:40.610 2012]
[Tue Jul 31 11:35:40.610 2012] U-Boot 2011.09 (Jul 20 2012 - 14:30:09)
[Tue Jul 31 11:35:40.610 2012]
[Tue Jul 31 11:35:40.610 2012] I2C: ready
[Tue Jul 31 11:35:40.657 2012] DRAM: 256 MiB
[Tue Jul 31 11:35:40.907 2012] WARNING: Caches not enabled
[Tue Jul 31 11:35:41.016 2012] Found a daughter card connected
[Tue Jul 31 11:35:41.094 2012] NAND: HW ECC Hamming Code selected
[Tue Jul 31 11:35:41.094 2012] 256 MiB
[Tue Jul 31 11:35:41.094 2012] MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1
[Tue Jul 31 11:35:41.579 2012] Net: cpsw
[Tue Jul 31 11:35:41.594 2012] Hit any key to stop autoboot: 0
[Tue Jul 31 11:35:41.735 2012] SD/MMC found on device 0
[Tue Jul 31 11:35:41.735 2012] reading uEnv.txt
[Tue Jul 31 11:35:41.751 2012]
[Tue Jul 31 11:35:41.751 2012] ** Unable to read "uEnv.txt" from mmc 0:1 **
[Tue Jul 31 11:35:41.751 2012] reading uImage
[Tue Jul 31 11:35:42.110 2012]
[Tue Jul 31 11:35:42.110 2012] 3163680 bytes read
[Tue Jul 31 11:35:42.126 2012] ## Booting kernel from Legacy Image at 80007fc0 ...
[Tue Jul 31 11:35:42.126 2012] Image Name: Arago/3.2.0-psp04.06.00.08.sdk/a
[Tue Jul 31 11:35:42.172 2012] Image Type: ARM Linux Kernel Image (uncompressed)
[Tue Jul 31 11:35:42.188 2012] Data Size: 3163616 Bytes = 3 MiB
[Tue Jul 31 11:35:42.188 2012] Load Address: 80008000
[Tue Jul 31 11:35:42.188 2012] Entry Point: 80008000
[Tue Jul 31 11:35:42.188 2012] Verifying Checksum ... OK
[Tue Jul 31 11:35:43.063 2012] XIP Kernel Image ... OK
[Tue Jul 31 11:35:43.079 2012] OK
[Tue Jul 31 11:35:43.079 2012]
[Tue Jul 31 11:35:43.079 2012] Starting kernel ...
[Tue Jul 31 11:35:43.094 2012]
[Tue Jul 31 11:35:43.094 2012] Uncompressing Linux... done, booting the kernel.
[Tue Jul 31 11:35:44.032 2012] [ 0.337524] mtdoops: mtd device (mtddev=name/number) must be supplied
[Tue Jul 31 11:35:44.235 2012] [ 0.541198] smartreflex smartreflex: am33xx_sr.probe: Zero NValue read from EFUSE
[Tue Jul 31 11:35:44.266 2012] [ 0.549591] sr_init: platform driver register failed
[Tue Jul 31 11:35:45.344 2012] INIT: version 2.86 booting
[Tue Jul 31 11:35:45.563 2012] Please wait: booting...
[Tue Jul 31 11:35:45.641 2012] Starting udev
[Tue Jul 31 11:35:46.891 2012] Remounting root file system...
[Tue Jul 31 11:35:47.079 2012] Caching udev devnodes
[Tue Jul 31 11:35:47.407 2012] ALSA: Restoring mixer settings...
[Tue Jul 31 11:35:47.501 2012] No state is present for card EVM
[Tue Jul 31 11:35:47.516 2012] Found hardware: "AM335X_EVM" "" "" "" ""
[Tue Jul 31 11:35:47.516 2012] Hardware is initialized using a generic method
[Tue Jul 31 11:35:47.516 2012] No state is present for card EVM
[Tue Jul 31 11:35:47.532 2012] Configuring network interfaces... udhcpc (v1.13.2) started
[Tue Jul 31 11:35:47.657 2012] Sending discover...
[Tue Jul 31 11:35:50.657 2012] Sending discover...
[Tue Jul 31 11:35:53.657 2012] Sending discover...
[Tue Jul 31 11:35:53.673 2012] Sending select for 158.218.103.110...
[Tue Jul 31 11:35:53.688 2012] Lease of 158.218.103.110 obtained, lease time 28800
[Tue Jul 31 11:35:53.794 2012] adding dns 158.218.108.21
[Tue Jul 31 11:35:53.907 2012] adding dns 157.170.32.67
[Tue Jul 31 11:35:53.907 2012] done.
[Tue Jul 31 11:35:53.907 2012] Setting up IP spoofing protection: rp_filter.
[Tue Jul 31 11:35:53.907 2012] Tue Jun 26 22:09:00 UTC 2012
[Tue Jul 31 11:35:53.907 2012] INIT: Entering runlevel: 5
[Tue Jul 31 11:35:53.923 2012] Starting system message bus: dbus.
[Tue Jul 31 11:35:53.985 2012] Starting telnet daemon.
[Tue Jul 31 11:35:54.016 2012] Starting syslogd/klogd: done
[Tue Jul 31 11:35:54.048 2012] Starting thttpd.
[Tue Jul 31 11:35:55.423 2012]
[Tue Jul 31 11:35:55.423 2012]
[Tue Jul 31 11:35:55.454 2012]
[Tue Jul 31 11:35:55.454 2012]
```

```
[Tue Jul 31 11:35:55.469 2012] |_| |_| |_| |_| |_| |_| |_| |_| |_|
[Tue Jul 31 11:35:55.469 2012] |_| |_| |_| |_| |_| |_| |_| |_| |_| |_|
[Tue Jul 31 11:35:55.469 2012] Arago Project http://arago-project.org am335x-evm tty00
[Tue Jul 31 11:35:55.469 2012]
[Tue Jul 31 11:35:55.469 2012] Arago 2011.09 am335x-evm tty00
[Tue Jul 31 11:35:55.469 2012]
[Tue Jul 31 11:35:55.485 2012] am335x-evm login:
```

PSP Linux Drivers




This section provides brief overview of the device drivers supported in the Linux PSP release.

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

Device Driver List

The following table list the various device drivers supported and the device they are supported on. On detailed information on specific features or limitations of a particular driver, refer to the chapter catering to that driver in this document.

Peripheral Driver Support

Peripheral 	Description 	Linux driver type 	DMA usage 
Audio (McASP)	Audio Record and Playback	ALSA SoC	EDMA
Ethernet	Ethernet Network driver	Netdev	Internal DMA
Controller Area Network (CAN)	Controller Area Network driver	SocketCAN	None
USB MSC Host	USB Mass Storage Class Host Driver	Block	USB DMA Internal
USB HID Host	USB Human Interface Device Host Driver	Input driver	USB DMA Internal
USB MUSB HCD	MUSB Host controller driver	USB HCD	USB DMA Internal
NAND Flash	Flash storage system	MTD Character and Block	Not Supported
GLCD	Graphical LCD driver	Frame Buffer	LCDC Internal DMA
SPI Flash	Flash storage system	MTD Character and Block	EDMA
MMC/SD	Interface to MultiMedia Secure Digital cards	Block	EDMA
UART	Serial Communication Interface	Character	Not Supported
I2C	Inter-IC Communication	Character	Not Supported
RTC	Real-time clock	Character	None
Watchdog	Watchdog Timer	Miscellaneous	None
SPI	Serial Peripheral Interface	Character	EDMA
Touchscreen	Touchscreen driver	Input driver	None
CPU Idle	Supports multiple CPU idle levels	NA	None
CPU Freq	Supports multiple SoC operating levels (OPPs)	NA	None
Suspend-to-RAM	Helps the device enter its lowest power mode not requiring a reboot.	NA	None

ALSA SoC Audio Driver

This section an overview of the ALSA SoC audio driver features along with the throughput and CPU load numbers.

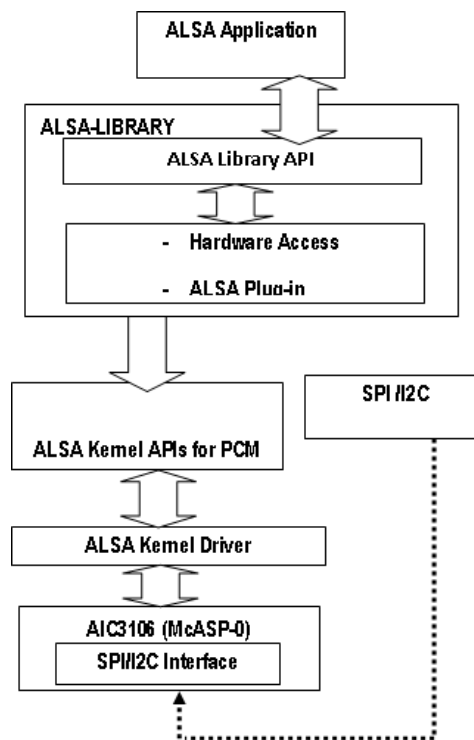
Introduction

AM335x Audio driver complies to the Advanced Linux Sound Architecture (ALSA) System on Chip (SoC) framework (ASoC).

The 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.



Driver Features

The driver supports the following features:

1. Supports AIC3106 audio codec in ALSA SoC framework.
2. Multiple sample rate support (8 KHz, 44.1 KHz and 48 KHz commonly used) for both capture and playback.
3. Supports audio in stereo mode
4. Supports simultaneous playback and record (full-duplex mode).
5. Supports mixer interface for the audio codec

Features Not Supported

1. OSS based applications, which use ALSA-OSS emulation layer, are not supported.
2. Formats such as TDM, Left and Right Justified are currently not supported.
3. Synthesizer and midi interfaces are not supported.

Constraints

1. By default, codec is configured in master mode and McASP is used as slave. Testing of the audio sub-system is done in this configuration only.
2. Sampling frequencies for playback and capture streams should be same.
3. The audio driver does not allow opening the same stream (playback/capture) multiple times.

Supported System Calls

Refer ALSA project - the C library reference [\[1\]](#) for API calls.

Performance and Benchmarks

1. Access type - RW_INTERLEAVED
2. Channels - 2
3. Format - S16_LE
4. Period size - 64

Audio playback

Sampling Rate (in Hz)	CPU Load (in %)
8000	0.06
11025	0.09
16000	0.20
22050	0.26
32000	0.86
44100	0.41
48000	0.63
88200	0.93
96000	0.00

Audio Capture

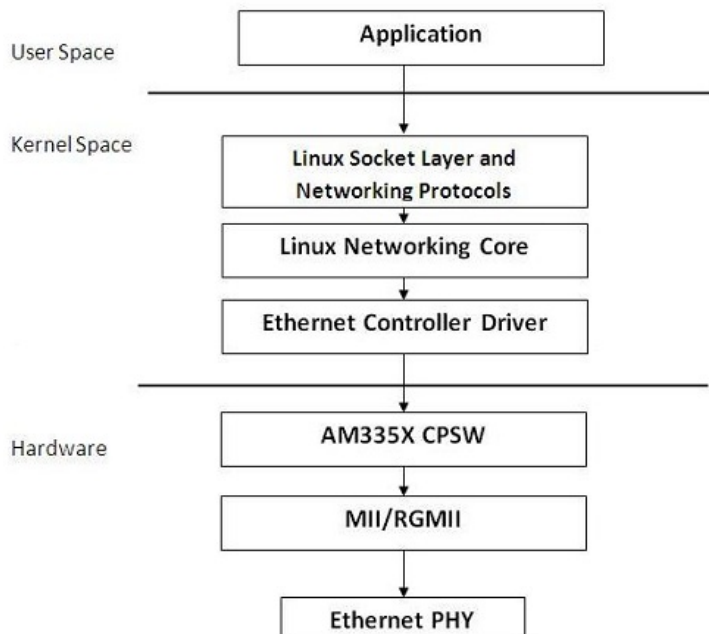
Sampling Rate (in Hz)	CPU Load (in %)
8000	0.21
11025	0.25
16000	0.30
22050	0.50
32000	0.00
44100	1.05
48000	0.86
88200	2.1
96000	3.66

Ethernet Driver

This section provides an overview of the Ethernet driver features along with throughput and CPU load numbers. Ethernet driver follows standard Linux network interface architecture.

Introduction

The Ethernet driver supports the Linux netdev interface.



Driver Features

The driver supports the following features:

1. 10/100/1000 Mbps mode of operation.
2. Auto negotiation.
3. Full duplex and half duplex mode of operation.
4. Linux NAPI support
5. Support for MII and RGMII interfaces to PHY
6. CPSW Interrupt Pacing. This driver uses Timer 5 & 6 for CPSW Interrupt Pacing. Re-using this timer for any other purpose will result in CPSW Interrupt Pacing not working correctly.
7. Operation of both external ports as independent network interfaces
8. VLAN and ALE configuration support

Features Not Supported

1. Promiscuous mode of operation.
2. IEEE 1588

Supported System Calls

Supports the socket() and related system calls in accordance with Linux architecture.

Performance and Benchmarks

TCP Performance

Ethernet Port0 TCP - 1000Mbps Mode Performance

TCP Window Size (in KBytes)	Bandwidth (without interrupt pacing, in Mbits/sec)	CPU Load (without interrupt pacing, in %)	Bandwidth (with interrupt pacing, in Mbits/sec)	CPU Load (with interrupt pacing, in %)
16	256.8	98.89	303.2	95.56
32	243.92	99.57	296	99.35
64	234.4	99.49	288	99.41
128	221.6	99.57	260	99.47
128	217.6	99.41	254.4	99.57

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

- iperf version 2.0.5
- On PC Host invoke iperf in the server mode.

```
iperf -s
```

- On the DUT iperf is invoked in client mode (bi-directional traffic for 60 seconds).

```
iperf -c <server ip> -w <window size> -m -f M -d -t 60
```

- Interrupt pacing feature enabled with pacing interval set to 500usecs.

```
ethtool -C eth0 rx-usecs 500
```

- DUT is connected to a gigabit network.

UDP Performance

Ethernet Port0 UDP - Transmit Performance (MTU Size packets)

Maximum Bandwidth limit (Mbits/sec)	Bandwidth observed (Mbits/sec)	Jitter (milliseconds)
500	280.8	0.04
750	280.8	0.04
1000	280.8	0.05

Ethernet Port0 UDP - Receive Performance (MTU Size packets)

Maximum Bandwidth limit (Mbits/sec)	Bandwidth observed (Mbits/sec)	Jitter (milliseconds)	Packet loss (percentage)
250	124.8	0.06	48

- iperf version 2.0.5
- For receive performance, on DUT, invoke iperf in server mode.

```
iperf -s -u
```

- For transmit performance, on DUT, invoke iperf in client mode.

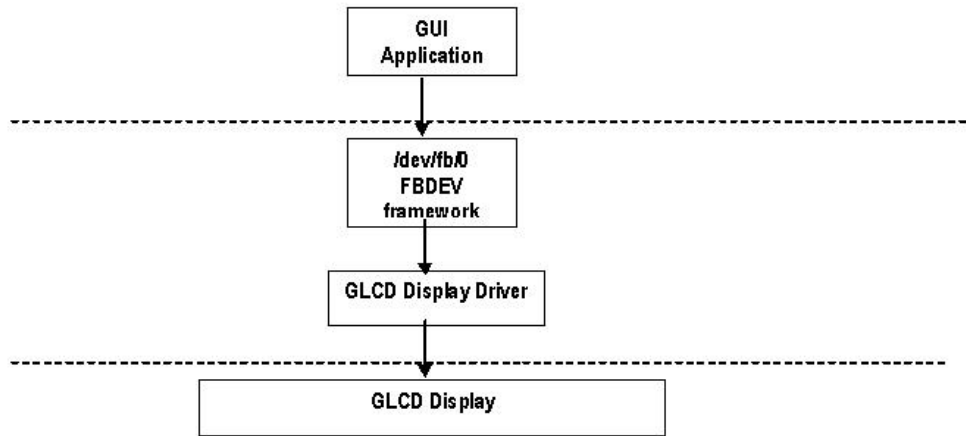
```
iperf -c <server ip> -b <bandwidth limit> -f M -t 60
```

Graphical LCD (GLCD) Driver

This chapter describes the GLCD driver architecture, driver features and performance numbers (throughput and CPU load).

Introduction

GLCD driver is based on Fbdev framework and it reuses da8xx-fb used on AM18x device. The LCDC IP on AM335x is an upgraded version of that found on AM18x SoC.



Driver Features

The driver supports the following features:

1. Supports WVGA display through Fbdev framework.
2. Supports display of ARGB:8888 images at max resolution 2048*2048, but pixel clock can be driven Max 150MHz.
3. Supports getting and setting the variable screen information.
4. Supports retrieving the fixed screen information.

Features Not Supported

1. Brightness and color control ioctls not supported.

Constraints

1. There is limit on Max pixel clock of 150MHz.

Supported System Calls

```
open(), close(), read(), mmap(), ioctl()
```

Performance Benchmarks

Performance and Benchmarks not available in this release.

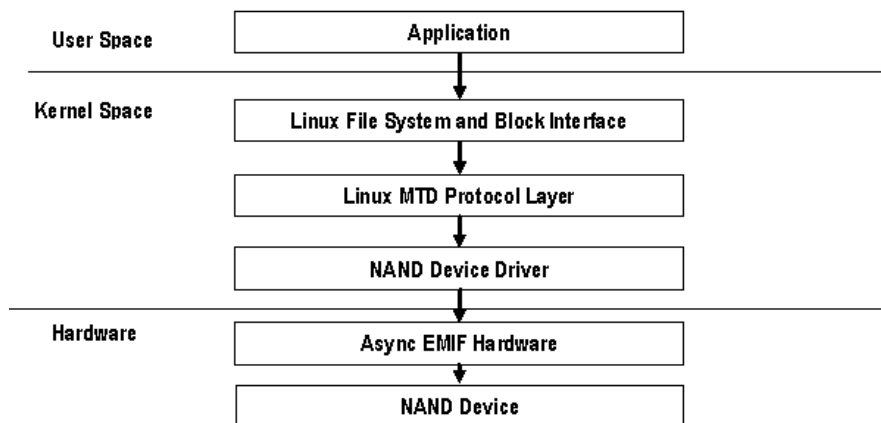
NAND Driver

This section provides an overview of the NAND flash driver features along with throughput and CPU load numbers.

Introduction

The NAND driver is implemented as a character and block driver, compliant with the Linux MTD subsystem interface. It supports various NAND Flash chips (see file `drivers/mtd/nand/nand_ids.c` in Linux kernel sources). The NAND driver creates the device nodes for user space access (`/dev/mtdblock0`, `/dev/mtdblock1`, `/dev/mtd0`, `/dev/mtd1` and so on.).

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



Driver Features

The driver supports the following features:

1. BCH8 with Hardware Error Correction is supported
2. Supports Read, Write and Erase
3. Bad Block Management

4. Polled Prefetch mode of transfer (enabled by default)
5. SLC NAND
6. Only 8-bit NAND is tested since EVM supports only a 8-bit NAND

Features Not Supported

1. DMA Mode of operation

Supported System Calls

Supports the system call support provided by Linux MTD interface viz. `open()`, `close()`, `read()`, `write()`, `ioctl()`

Performance Benchmarks

Please refer [MTD Test Setup](#) before proceeding.

NAND UBIFS Write (async mode mount)

Buffer size	Through put in Mbytes/sec	CPU Load (in %)
102400	2.48	100
262144	2.49	100
5242880	2.49	100

NAND UBIFS Read (async mode mount)

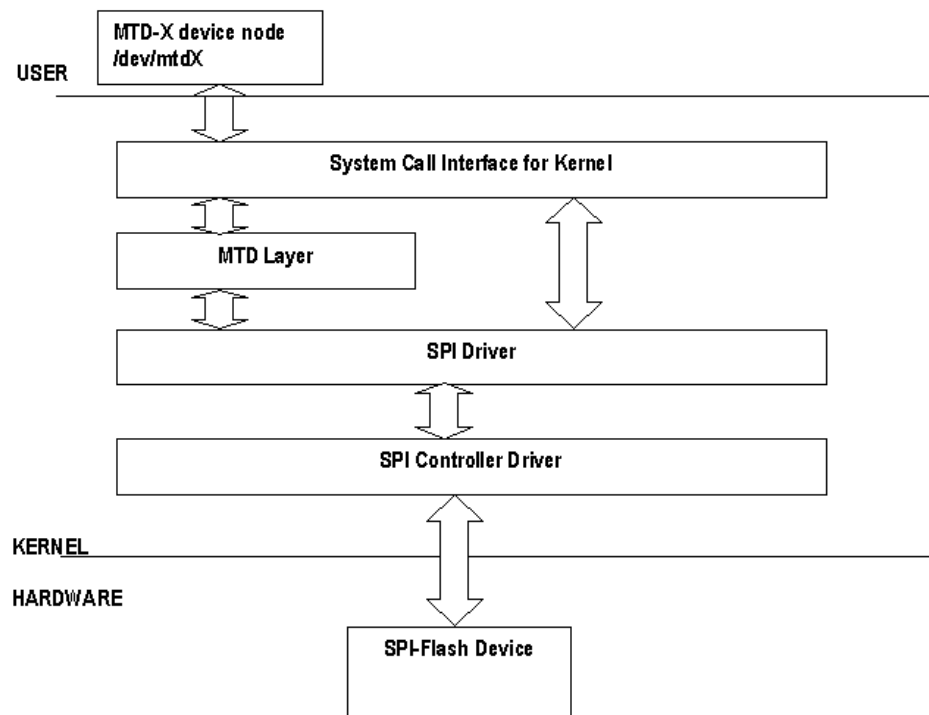
Buffer size	Through put in Mbytes/sec	CPU Load (in %)
102400	3.38	100
262144	3.41	100
5242880	3.41	100

SPI Flash Driver

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

Introduction

SPI Flash driver is implemented as block driver and compliant with standard MTD driver. It supports various flash devices. The SPI flash driver creates device node for user space access (example, `/dev/mtd1`).



Driver Features

The driver supports the following features:

1. PIO and DMA Mode of Operation are supported.

Supported System Calls

Supports the system call support provided by MTD interface viz. `open()`, `close()`, `read()`, `write()`, `ioctl()`

Performance Benchmarks

Test setup is available [here](#).

SPI Write

Buffer size	Through put in Mbytes/sec	CPU Load (in %)
102400	0.09	89.4
262144	0.09	89.41
524288	0.09	89.52
1048576	0.09	89.55

SPI Read

Buffer size	Through put in Mbytes/sec	CPU Load (in %)
102400	1.22	20.44
262144	1.22	16.76
524288	1.22	16.76
1048576	1.22	20

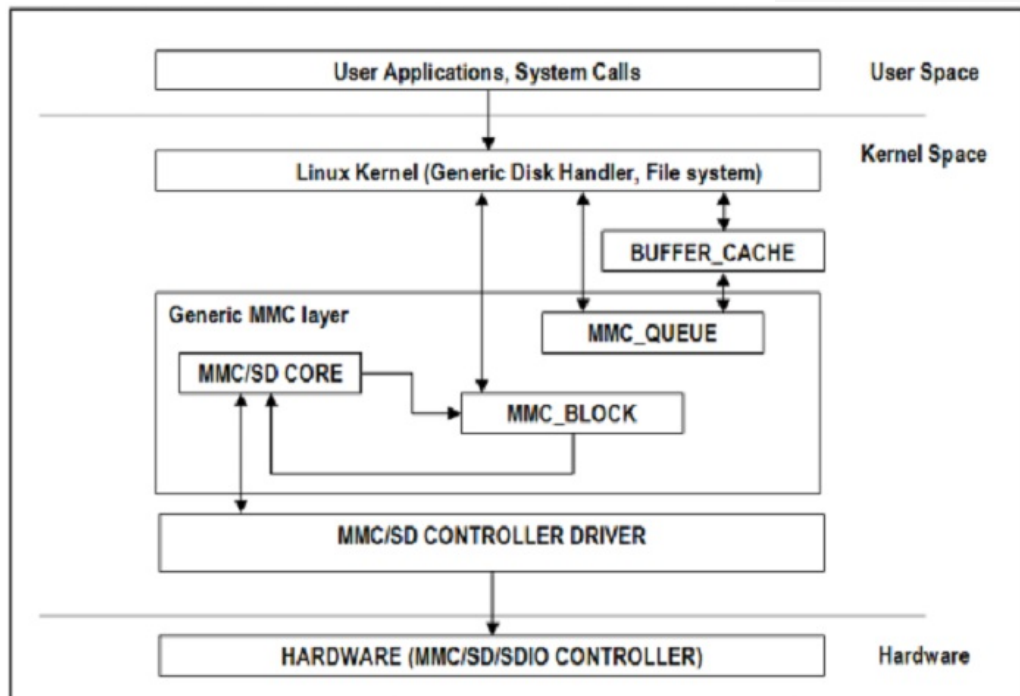
NOTE Over all SPI performance is low because we can only do a 24MHz clock on the board against the theoretical max of 48 MHz. Also high CPU load on write is expected because the SPI flash driver does a write every page size (256 bytes) resulting in high interrupt load.

MMC/SD Driver

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

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/mmcblkp1, /dev/mmcblkp2) are created for user space access.



Driver Features

The driver supports the following features:

1. MMC/SD native protocol command/response set
2. Single/multiple block data transfers
3. Linux file system and generic MMC layer abstract details of block devices (MMC)
4. High-speed (SDv1.1) and High Capacity (SDv2.0) cards
5. Support for 4 bit modes
6. Support for card detect and Write protect features
7. DMA and polled mode for data transfer operations

Features Not Supported

- 1. Support for 1-bit, 8-bit mode of operation.
- 2. SPI mode of operation
- 3. PIO mode of operation

Constraints

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

Supported System Calls

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

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.

Please refer [MMC/SD Test Setup](#) before proceeding.

EXT2 file system

SD - Write Performance values		
Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	7.58	9.38
1048576	8.68	9.05
5242880	8.04	8.29

SD - Read Performance values		
Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	19.37	20.38
1048576	19.39	24.97
5242880	19.37	25.53

The performance numbers were captured using the following:

- SD Card ([Sandisk Extreme 64G Class 10 SDXC card](#))
- File System: ext2
- Partition was mounted with async option

VFAT file system

SD - Write Performance values		
Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	9.31	18.60
1048576	10.23	15.70
5242880	9.96	17.23

SD - Read Performance values		
Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	18.16	27.43
1048576	18.15	26.14
5242880	18.11	27.08

The performance numbers were captured using the following:

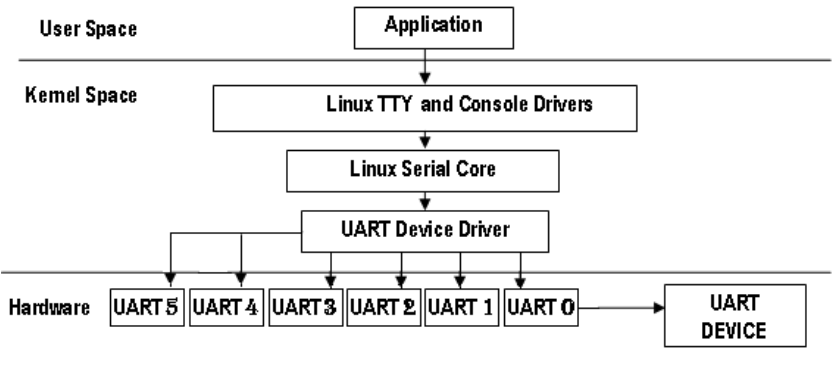
- SD Card ([Sandisk Extreme 64G Class 10 SDXC card](#))
- File System: vfat
- Partition was mounted with async option

UART Driver

This chapter provides details on UART driver.

Introduction

The UART driver is implemented as a serial driver, and can be accessed from user space as `/dev/ttyOX(X=0-5)`



Features Not Supported

- Hardware Flow Control

Supported System Calls

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

Supported IOCTLs

Constant	Description
TIOCGSERIAL	Gets device parameters from the UART (example, port type, port num, baud rate, base divisor, and so on.
TIOCSSERIAL	Sets UART device parameters (example, port type, port num, baud rate, base divisor, and so on)

Performance and Benchmarks

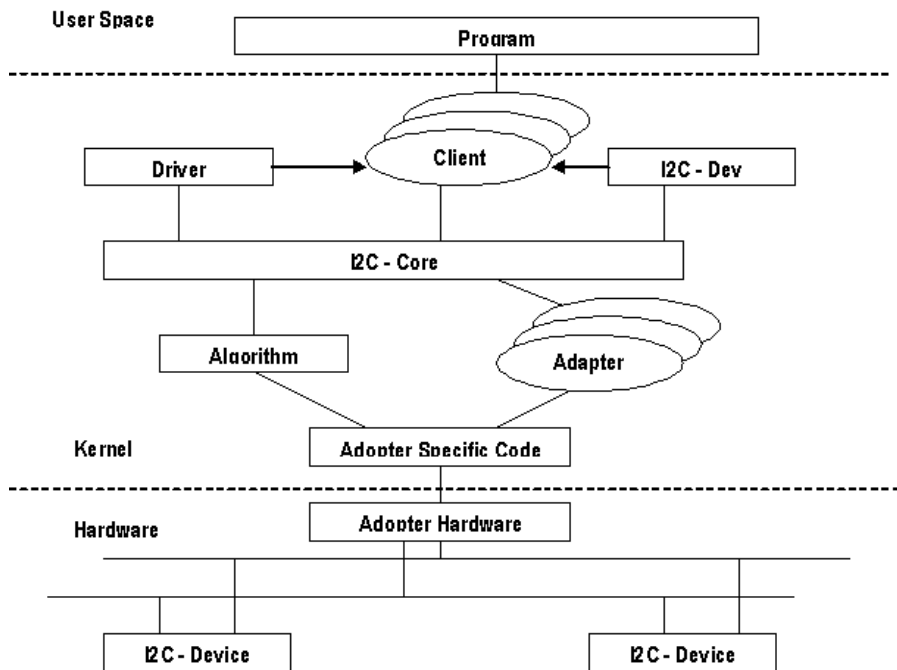
Performance and Benchmarks not available in this release.

I2C Driver

This chapter provides details on I2C driver.

Introduction

The I2C peripheral is compliant with the Philips Semiconductor I2C-bus specification version 2.1. The I2C driver is implemented as a serial driver. The I2C driver can be accessed from the user space as `/dev/i2c/0`.



Driver Features

The driver supports the following features:

1. 7-bit addressing mode
2. Fast mode
3. Interrupt mode

Features Not Supported

1. 7-bit and 10-bit addressing combined format is not supported
2. DMA mode is not supported

Supported System Calls

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

Supported IOCTLS

Constant	Description
I2C_SLAVE_FORCE	Changes slave address. Slave address is 7 or 10 bits. This changes the address, even if it is already considered.
I2C_FUNCS	Gets the adapter functionality
I2C_RDWR	Combined R/W transfer (one stop only)

Performance and Benchmarks

Performance and Benchmarks not available in this release.

EDMA Driver

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

Introduction

The EDMA controller handles all data transfers between the level-two (L2) cache/memory controller and the device peripherals. On AM335x EDMA has one instance of Channel controller. Each EDMA instance supports up to 32-dma channels and 8 QDMA channels. The EDMA consists of a scalable Parameter RAM (PaRAM) that supports flexible ping-pong, circular buffering, channel-chaining, auto-reloading, and memory protection. The EDMA allows movement of data to/from any addressable memory spaces, including internal memory (L2 SRAM), peripherals, and external memory.

The EDMA driver exposes only the kernel level API's. This driver is used as a utility by other drivers for data transfer.

Driver Features

The driver supports the following features:

1. Request and Free DMA channel
2. Programs DMA channel
3. Start and Synchronize with DMA transfers
4. Provides DMA transaction completion callback to applications
5. Multiple instances of EDMA driver on a single processor

Features Not Supported

1. QDMA is not supported.
2. Reservation of resources (channels and PaRAMs) for usage from other masters is not supported.

Touchscreen Driver

Introduction

Touchscreen controller on AM335x is an 8 channel general purpose ADC, with optional support for interleaving Touch Screen conversions for a 4-wire, 5-wire, or 8-wire resistive panel. A resistive touchscreen operates by applying a voltage across a resistive network and measuring the change in resistance at a given point on the matrix where the screen is touched by an input (stylus or finger). The change in the resistance ratio marks the location on the touchscreen.

Driver Features

The driver supports the following features:

- Resistive touchscreen
- Hardware synchronized one shot mode
- Programmable Open delay/Sampling delay before sampling each channel
- Programmable averaging of input samples - 16
- Support for servicing FIFOs via CPU.
- ADC sampled data is 12-bit wide.
- Pressure(z) measurements.

Features Not Supported

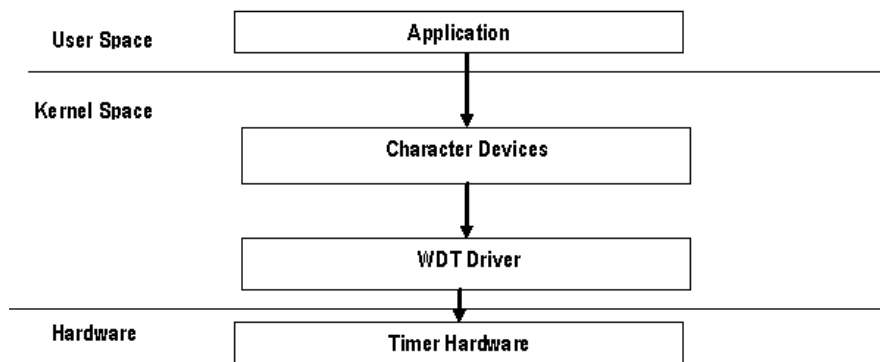
- DMA Request event.

Watchdog(WDT) Driver

This chapter provides details on Watchdog timer driver.

Introduction

AM335x has a 32-bit watchdog timer which can be used to reset the hardware in case of a software fault. Once the /dev/watchdog is opened, it will reboot the system unless a user space daemon resets the timer at regular intervals within a certain timeout period. If watchdog device node is closed before timeout also reboot won't happen. The WDT driver is registered as a misc device. Default timeout of this driver is 60 seconds.



Driver Features

The driver supports the following features:

1. Supports IOCTLs to set/get the timeout value, ping the watchdog & query the watchdog structure info.
2. Driver can be built as a loadable module and inserted dynamically.

Supported System Calls

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

Supported IOCTLs

Constant	Description
WDIOC_GETSUPPORT	This ioctl returns "struct watchdog_info", which tells what the device can do

WDIO_KEEPAIVE	This ioctl can be used to notify the watchdog timer that the user space application is alive
WDIO_SETTIMEOUT	Watchdog timeout or margin can be dynamically changed using this ioctl
WDIO_GETTIMEOUT	This ioctl returns the present watchdog timeout period in seconds

USB Driver

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

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

MUSB OTG controller

Description

The MUSB driver is implemented on top of Mentor OTG IP version 2.0 which supports all the speeds (High, Full and Low (host mode only)). On AM335x, MUSB uses CPPI 4.1 DMA for all the transfers.

Driver Features

The driver supports the following features:

Host Mode

1. Human Interface Class (HID)
2. Mass Storage Class (MSC)
3. Hub Class
4. USB Host Video Class (UVC)
5. USB Host Audio Class (UAC)
6. USB CDC HOST (USBNET)

Gadget mode

1. Mass Storage Class (MSC)
2. USB Networking - RNDIS/CDC

OTG mode

1. Host Negotiation Protocol (HNP)

Features Not Supported

USB Mass Storage Class Host Driver

Driver Features

The driver supports the following features:

1. DMA mode
2. PIO mode

Supported System Calls

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

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.

Setup : Western Digital 1TB (My Book Essential) USB hard disk is connected to usb0 port. File read/write performance data on usb0 port is captured. For detailed test setup refer to [USB MSC host test setup](#)

USB MSC (MUSB) Host mode EXT2 File System Performance

USB-MSC MUSB Host-DMA-Write Performance values

Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	21.94	21.19
1048576	22.12	19.13

5242880	22.42	17.58
---------	-------	-------

USB-MSC MUSB Host-DMA-Read Performance values

Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	16.89	28.83
1048576	17.14	29.44
5242880	18.11	30.63

USB MSC (MUSB) Host mode VFAT File System Performance

USB-MSC MUSB Host-DMA-Write Performance values

Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	18.3	30.7
262144	18.42	27.72

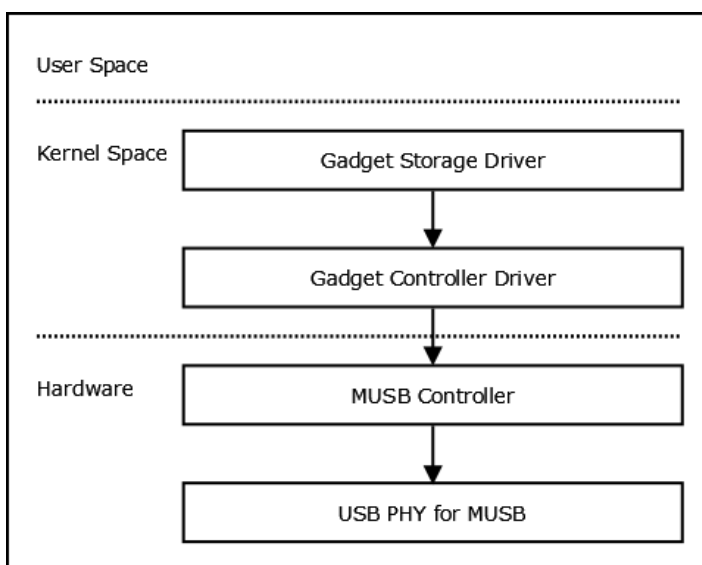
USB-MSC MUSB Host-DMA-Read Performance values

Buffer Size (in Bytes)	Transfer Rate (in MBytes/sec)	CPU Load (in %)
102400	16.01	25.9
262144	15.98	24.71

USB Mass Storage Class Slave Driver

Description

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



Driver Features

The driver supports the following features:

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

Performance Benchmarks

The performance numbers are captured using a file inside /dev/shm as storage gadget media exposed as removable media to an Ubuntu 10.04 host over USB. The storage media is formatted as vfat filesystem.

Storage gadget media file is created using the command:

```
# dd if=/dev/zero of=/dev/shm/50M bs=1M count=50"
```

Insert the USB mass storage gadget with /dev/shm/50M as the backing storage:

```
# insmod g_mass_storage.ko file=/dev/shm/50M
```

For test setup details refer to [USB MSC device setup](#)

USB Slave DMA Write Performance values

Bytes Transferred (MB)"/dev/shm/50M" file as storage device	Number of files transferred	Total Bytes transferred (MB)	Transfer Rate (MB/sec)
50	1	50	15.62

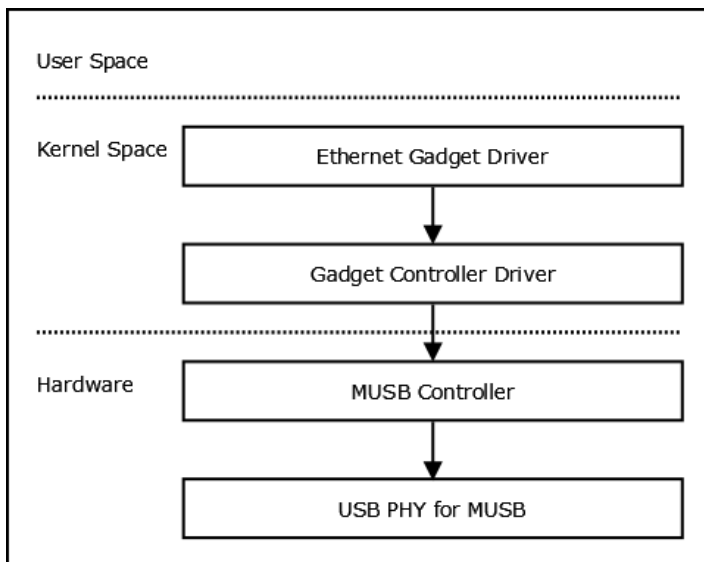
USB Slave DMA Read Performance values

Bytes Transferred (MB)"/dev/shm/50M" file as storage device	Number of files transferred	Total Bytes transferred (MB)	Transfer Rate (MB/sec) on Windows XP
50	1	50	19.23

USB CDC/RNDIS Slave Driver

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.



Driver Features

The driver supports the following features:

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

Supported System Calls

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

Performance Benchmarks

Performance benchmarks were collected using the Iperf tool and default options were used to collect the throughput numbers.

USB CDC-DMA Performance

USB CDC-DMA Performance values - Client

TCP Window Size(in KBytes)	Interval (in Seconds)	Bandwidth (Mbits/Sec)
16	60	105
32	60	104
64	60	105
128	60	104

USB RNDIS-DMA Performance

USB RNDIS-DMA Performance values - Client

TCP Window Size(in KBytes)	Interval (in Seconds)	Bandwidth (Mbps/Sec)
16	60	61.7
32	60	65.8
64	60	69.6
128	60	69.4

USB Human Interface Device (HID) Driver

Description

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

Driver Features

The driver supports the following features:

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

DCAN Driver

This section provides details on DCAN driver.

Introduction

The CAN (DCAN) driver supports the Linux netdev interface. CAN driver supports transmitting and receiving in-order packets, processing happens at various bit rates.

Driver Features

The driver supports the following features:

1. SocketCAN networking framework
2. Linux NAPI support for Rx operations
3. Local loopback of sent frames (IFF_ECHO)
4. Standard CAN bus bit-rates namely 10K, 20K, 50K, 100K, 125K, 250K, 500K & 1M

Features Not Supported

1. ID filtering / Acceptance masks - since Linux infrastructure does not support this the driver cannot provide the feature

Supported System Calls

Supports the `socket()` and related system calls in accordance with Linux architecture. Refer to `can.txt` under `Documentation/networking` folder

Power Management

Introduction

AM335x devices 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 an overview of power management features supported and steps to enable these features in the kernel configuration. It also provides the typical power consumption measurement observed for different scenarios.

cpuidle

When idle loop is executed, kernel is not doing any useful 'work'. This is an opportunity to save power. The `cpuidle` framework helps in saving power during the idle state.

cpufreq

CPU is not loaded evenly during execution. This provides an opportunity to save power by adjusting/scaling voltage and frequency based on the current cpu load.

Power Measurements

All the measurements below were done on the AM335x EVM. The definition of idle at Linux prompt is as per the tables shown below

Modules with active clocks

Name	Instances

ADC_TSC	NA
CPSW	NA
ELM	NA
EMIF	NA
EPWMSS	0
GPIO	0-3
GPMC	NA
LCD	NA
MCASP	1
MAILBOX	NA
OCMC	NA
RTC	NA
TIMER	1,2,5,6
TPCC	NA
TPTC	0-2
UART	0-5
USB	NA

Lock Frequency of various PLLs

PLL	Lock Frequency (MHz)
Core PLL	1000
PER PLL	960
MPU PLL	As per OPP (275/500/600/720)
DISP PLL	300
DDR PLL	266

At each OPP

This section indicates the power measured for all power rails at various OPPs (selected via `cpufreq`). The measurements were done with the `uImage` created using `am335x_evm_defconfig`. Measurements were taken while Linux was idle at command prompt. The measurements were done using EVM revision 1.2A.

Active Power Consumption at various OPPs (CPUFreq P states)

Power Rail	OPP 275 MHz (milliwatts)	OPP 500 MHz (milliwatts)	OPP 600 MHz (milliwatts)	OPP 720 MHz (milliwatts)
Total Power (Sum of all Rails)	468.91	513.76	559	607.23
VDD_CORE	185.79	186.42	186.89	187.44
VDD_MPU	55.46	98.5	141.86	187.81
VDDS	3.33	3.33	3.33	3.33
VDDS_RTC	0.80	0.80	0.80	0.80
VDDS_DDR	47.97	47.96	47.9	47.87
VDDS_SRAM_CORE_BG	2.57	2.61	2.68	2.73
VDDS_SRAM_MPU_BB	2.43	2.5	3.38	4.07
VDDS_PLL_DDR	1.88	1.88	1.88	1.88
VDDS_PLL_CORE_LCD	14.59	14.59	14.59	14.59
VDDS_PLL_MPU	1.9	2.95	3.52	4.23
VDDS_OSC	1.24	1.24	1.24	1.24

VDDA_1P8V_USB0_1	29.22	29.22	29.22	29.23
VDDS_A3P3V_USB0_1	11.48	11.52	11.47	11.47
VDDA_ADC	0.90	0.90	0.90	0.90
VDDSHV1	0.60	0.60	0.60	0.60
VDDSHV2	6.31	6.33	6.31	6.32
VDDSHV3	2.76	2.76	2.76	2.76
VDDSHV4	0.07	0.07	0.07	0.07
VDDSHV5	58	58.02	58.02	58.01
VDDSHV6	41.59	41.58	41.59	41.57

When suspended

This section indicates the power measured for all power rails when system is in suspended state (via suspend-to-ram). The measurements were done with the uImage created using am335x_evm_defconfig.

Suspend mode Power Consumption

Power Rail	Suspend Power (milliwatts)
Total Power (Sum of all Rails)	7.67
VDD_CORE	2.50
VDD_MPU	0.26
VDDS	0.38
VDDS_RTC	0.06
VDDS_DDR	0.02
VDDS_SRAM_CORE_BG	1.00
VDDS_SRAM_MPU_BB	0.01
VDDS_PLL_DDR	0.00
VDDS_PLL_CORE_LCD	0.00
VDDS_PLL_MPU	0.00
VDDS_OSC	0.00
VDDA_1P8V_USB0_1	0.00
VDDS_A3P3V_USB0_1	0.06
VDDA_ADC	0.81
VDDSHV1	0.31
VDDSHV2	0.07
VDDSHV3	0.10
VDDSHV4	0.06
VDDSHV5	0.20
VDDSHV6	1.79



Engage in the
TI E2E Community
Ask questions, share knowledge, explore ideas
and help solve problems with fellow engineers

For technical support please
post your questions at
<http://e2e.ti.com>. Please post
only comments about the article
AM335x-PSP 04.06.00.08
Features and Performance
Guide here.

Links



[ARM
Microcontroller
MCU](#)

[ARM
Processor](#)

[Digital
Media
Processor](#)

[Digital Signal
Processing](#)

[Microcontroller
MCU](#)

[Multi Core
Processor](#)

[Ultra Low Power
DSP](#)

[8 bit Microcontroller
MCU](#)

[16 bit Microcontroller
MCU](#)

[32 bit Microcontroller
MCU](#)

Categories: [AM335x](#) | [AM35x](#) | [AM37x](#) | [AM1x](#) | [AM18x](#) | [Linux](#) | [PSP](#)

[Leave a Comment](#)

This page was last modified on 1 August 2012, at 19:01.

This page has been accessed 165 times.

Content is available under [Creative Commons Attribution-Share Alike 3.0 license](#).

[Privacy policy](#) [About Texas Instruments Embedded Processors Wiki](#) [Disclaimers](#)

