Technical Note

e·MMC™ Linux Enablement

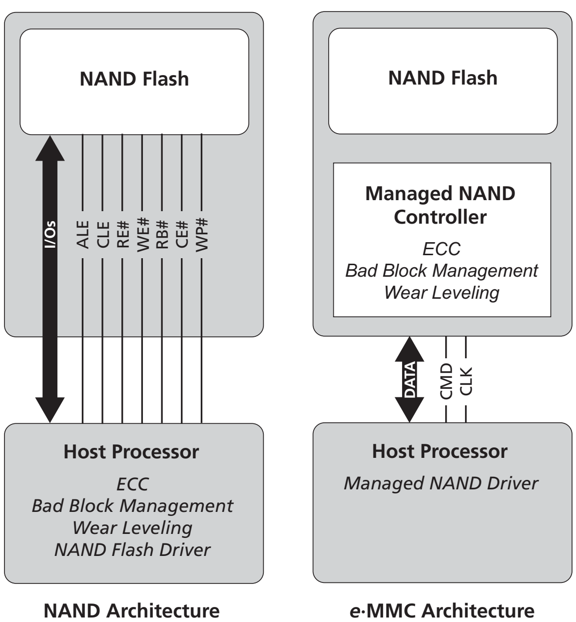
Introduction

Linux supports *e*·MMC™ devices and allows their integration within its subsystems. However, Linux does not support all *e*·MMC features introduced by the JEDEC standard in releases 4.1 through 4.41. This technical note describes supported and unsupported *e*·MMC features and how to enable them. It includes a discussion of the *e*·MMC standard (4.41) introduced in March 2010. This release includes two features designed specifically for the requirements of wireless market: Background operation and high priority interrupt (HPI).

An *e*·MMC device offers a low-cost, high-performance data storage solution. *e*·MMC is designed for use in a wide range of wireless and embedded applications. It can be used to store data and code. In addition, boot operation features enable the loading of boot and pre-boot code from an *e*·MMC device.

e·MMC architecture combines a NAND Flash memory device with a high-speed *e*·MMC controller in a single package. Its managed interface runs data management firmware modules (such as FTL, WL. and BBM) with error correction code (ECC), wear leveling, and bad block management. This architecture simplifies hardware and software integration by providing a standard interface specification that minimizes the need for host software to accommodate process node migrations and vendor-specific NAND Flash characteristics.

Figure 1: NAND vs. e·MMC Architecture



Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.

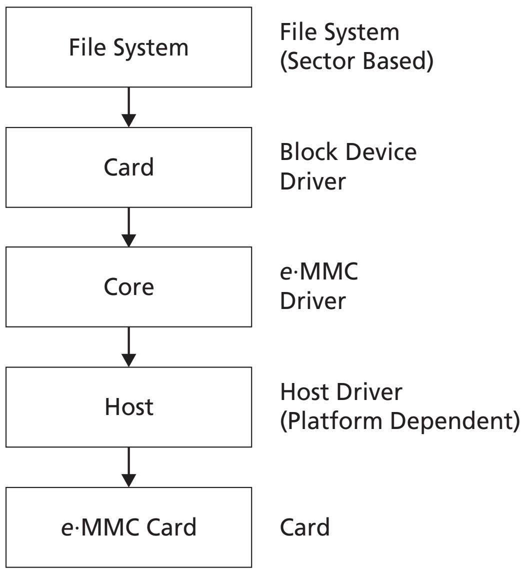
Products and specifications discussed herein are for evaluation and reference purposes only and are subject to change by Micron without notice. Products are only warranted by Micron to meet Micron's production data sheet specifications. All information discussed herein is provided on an "as is" basis, without warranties of any kind.

Linux Support for e·MMC

As described in the previous section, *e*·MMC devices have a built-in controller that runs flash translation layer (FTL) firmware. All FTL devices have interfaces providing block I/O access, and Linux provides an abstraction for block devices. This architecture allows several general purpose file systems to interface with it, such as FAT, EXTx, and so forth.

Figure 2 illustrates the Linux architecture for the *e*·MMC subsystem:

Figure 2: Linux Architecture for the e·MMC Subsystem



The *e*·MMC subsystem (./drivers/mmc) is organized into three layers:

- The *card layer* is the external interface. It is seen as a block device driver, depending on the enablement of block layer support for the kernel and interfaces within the file system.
- The *core layer* implements all *e*·MMC-dependent functionality, such as main operations, command management, host setting, management functions, and high-level functions.
- The *host layer* implements the driver for most known platform controllers and is platform dependent.

Linux enables *e*·MMC device integration within its subsystems. However, it does not support all *e*·MMC features introduced by the JEDEC standard. The next sections describe the supported and unsupported features and how to enable them.

Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.

Support History for *e*·MMC Features in the Linux Kernel

Support for some *e*·MMC features was introduced and/or improved in the following Linux kernel releases:

Table 1: Linux Kernel *e*·MMC Feature Support

Kernel Version	Feature Support Added
2.6.20	• Support for MMC 4.0 high-speed and wide-bus modes • Support for high speed (50Hz clock speed) SD cards • SDHCI high-speed support
2.6.22	• Support for MMC 4.2 sector-based cards
2.6.23	• Bounce requests for simple hosts • ENE controller ID for SDHCI
2.6.24	• SPI/SDIO support in the MMC layer
2.6.26	• OMAP: – Back cover switch support – New multislot structure – Changed driver to use the new multislot structure • MMC host test driver
2.6.27	• S3C24XX MMC/SD driver • Host driver for Ricoh® Bay1 controllers • Support for card-detection polling • Removed multiwrite capability • SDHCI: – Handle hot-remove – Support JMicron secondary interface – Scatter-gather (ADMA) support • au1xmmc: – Suspend/resume implementation – SDIO IRQ support • Bounce buffer highmem support for mmc_block • mmc_spi: Support for card-detection polling • at91_mci: Support for block size not modulo 4 • atmel-mci: Driver for Atmel on-chip MMC controllers
2.6.28	• sdio: High-speed support • s3cmci: cpufreq support • atmel-mci: – Experimental DMA support (commit) – Platform code for supporting multiple MMC slots • Support multiple MMC slots
2.6.29	• Support for 8-bit bus width • sdricoh_cs: Support for bay controller devices • mmc_spi: Support for OpenFirmware bindings • ricoh_mmc: Handle newer models of Ricoh controllers
2.6.30	• SDIO driver for Marvell SoCs • OpenFirmware bindings for SDHCI driver

Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.

Table 1: Linux Kernel *e*·MMC Feature Support (Continued)

Kernel Version	Feature Support Added
2.6.31	• Platform driver for secure digital host controller interface • Support for hosts that are only capable of 1-bit transfers • pxamci: Regulator support • New via-sdmmc host controller driver • Driver for CB710/720 memory card reader (MMC part) • SDHCI: – Platform driver for SDHCI – Support for hosts that are only capable of 1-bit transfers – sdhci-s3c: Samsung S3C-based SDHCI controller glue
2.6.32	• Core SDIO "real" suspend/resume support • MMC sleep and wake support (JEDEC 4.3 spec). • msm_sdccc: Driver for the HTC® Dream™ • Added MMC card sleep and awake support • MMC/SD Support for dm365 EVM
2.6.33	• Blackfin SD host controller driver • sdhci-of: Support for the Nintendo® Wii™ SDHCI controller • Reorganized driver to support additional hardware
2.6.36	• ERASE, SECURE ERASE, TRIM, and SECURE TRIM operations (JEDEC 4.4) • mmc_block: Discard and secure discard support • SD-combo (IO+mem) support • Performance tests
2.6.37	• New sdhci-pxa driver for Marvell SoCs • MMC 4.4 DDR support • sdhci-pltfm: Platform driver for imx35/51 • USB SD host controller (USHC) driver
2.6.39	• mxs-mmc: MMC host driver for i.MX23/28
3.0	• MMC CMD+ACMD passthrough IOCTL reliable write support • MMC boot partition support • New VUB300 USB-to-SD/SDIO/MMC driver • SD: Support for signal voltage switch procedure
3.2	• Enabled HPI for MMC cards that support this feature • Cache control for *e*·MMC 4.5 devices • *e*·MMC hardware reset support • Random fault injection • General-purpose MMC partition support (JEDEC 4.4) • SDHCI: *e*·MMC hardware reset support • sdhci-pci: Runtime PM support • mmc-test: *e*·MMC hardware reset test

Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.

e·MMC Enablement

Additional *e*·MMC features described in JEDEC standard 4.3 to 4.41 can be enabled within the Linux kernel. These features, which require different implementation efforts, are described in the following sections:

- Access protection
- Replay protected memory block (RPMB) area partition
- Reliable write
- Background operation and HPI

Access Protection (WRITE PROTECT and LOCK/UNLOCK) (JEDEC 4.3)

To enable the host to protect data against unwanted erases or writes, *e*·MMC supports two levels of WRITE PROTECT commands:

- The entire card may be write protected by setting the permanent or temporary write protect bits in the CSD.
- Specific segments of the cards may be write protected.

The CMD28, CMD29, and CMD 30 commands respectively protect, clear, and retrieve information about a write protect group against unwanted erases or writes. The properties of write protection are coded in the card-specific data, which also dictates the type of write protection.

The card LOCK/UNLOCK command (CMD42) enables the host to lock the card by providing a password. The card LOCK/UNLOCK command can only be performed when the card operates in single data rate mode. The password protection feature can be disabled permanently by setting the permanent password disable bit in the extended CSD.

To enable the write protect and lock/unlock features:

1. Enable EXT_CSD access (*e*·MMC driver).
2. Add a new API that enables CMD28, CMD29, and CMD30 commands.
3. Export it through the IOCTL operation.

Replay Protected Memory Block (RPMB) Area Partition (JEDEC 4.4)

This function provides a means for the system to store data to a specific memory area in an authenticated and replay-protected manner. This is provided by first programming authentication key information to the *e*·MMC memory (shared secret).

As the system cannot yet be authenticated in this phase, programming the authentication key must take place in a secure environment, such as in an OEM production facility. Later, the authentication key is used to sign the READ and WRITE accesses made to the replay-protected memory area with a message authentication code.

Usage of random number generation and count registers provide additional protection against the replaying of messages where messages could be recorded and played back later by an attacker.

To enable RPMB:

1. Add support to READ/WRITE the RPMB partition access.
2. Extend the EXT_CSD decoding to catch and change the RPMB partition setting.
3. Add support to set the access to the RPMB partition (similar to GPAP partitions).
4. Add the new API in the core layer.
5. Export it through an IOCTL operation.

Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.

Reliable Write (JEDEC 4.3)

In a RELIABLE WRITE operation, the old data pointed to by a logical address must remain unchanged until the new data written to the same logical address has been successfully programmed. This is to ensure that the target address updated by the reliable write transaction never contains undefined data. Data must remain valid, even if a sudden power loss occurs during programming.

To enable reliable write capability:

1. Enable EXT_CSD access (*e*·MMC driver).
2. Add support for reliable writes (define a flag to address a reliable flag request).
3. Make the file system aware of this feature.

File systems can use reliable writes to make power-loss safe WRITE operations (for example: to write metadata sectors).

Background Operation and HPI (JEDEC 4.41)

The JEDEC 4.41 standard introduced two features:

- Background operation
- High priority interrupt (HPI)

The background operation feature improves the use of idle time, which is the period of time when the device is not explicitly used. These time slots can be used to perform maintenance operations and increase system performance. Maintenance operations include:

- **Garbage collection**: Used to clean the memory of obsolete data and free blocks for further use. It includes algorithms to reduce the effects of read disturbances.
- **Wear leveling**: Ensures that all blocks are used evenly to avoid the premature wearing out of any blocks.

Operations are then separated into two types:

- **Foreground operations**: Operations that the host needs serviced, such as READ and WRITE commands.
- **Background operations**: Operations that the device executes while not servicing the host.

The background operation feature is still under discussion within the Linux-MMC e-mail list. However, there is a patch (2.6.35 kernel) that makes this feature available, which is available for download: https://lkml.org/lkml/2010/12/3/103.

HPI enables concurrently accessing different tasks. Different levels of priority are assigned to the tasks. HPI suspends any ongoing operation to execute a high priority memory operation, then resumes the suspended operation. The entire operation is performed by the operating system's kernel. The HPI operation was made available in kernel version 3.2.

Note: Patches to enable HPI in kernel 2.6.35 are still available: https://lkml.org/lkml/2010/12/3/105 and https://lkml.org/lkml/2010/12/3/107.

For more detailed information about these features, see "Taking Advantage of the New Features in e·MMC™ 4.41 for Mobile Applications" and "Software Considerations for e·MMC™ Devices Conforming to the 4.3 and 4.4 Specifications" on our web site.

Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.

Conclusion

Linux supports *e*·MMC devices and allows their integration within its subsystems. While Linux does not support all *e*·MMC features introduced by the JEDEC standard, patches are available to enable these features, such as background operation. For more details, contact your Micron representative or visit micron.com.

8000 S. Federal Way, P.O. Box 6, Boise, ID 83707-0006, Tel: 208-368-3900
www.micron.com/productsupport Customer Comment Line: 800-932-4992
Micron and the Micron logo are trademarks of Micron Technology, Inc.
e·MMC is a trademark of the JEDEC Solid State Technology Association.
All other trademarks are the property of their respective owners.

Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.

Revision History

Rev. B 03/12

- Updated hypertext links in document
- Updated the following sections
  - Linux Kernel e·MMC Feature Support table
  - *e*·MMC Enablement
  - Background Operation and HPI (JEDEC 4.41)
- Removed the following sections:
  - Boot Partition Management (JDEC 4.3)
  - Partition Management (JDEC 4.4)
  - Secure Erase and Secure Trim Erase (JDEC 4.4)
  - Patches for Linux Enablement

Rev. A 06/11

- Initial release

Micron Technology, Inc., reserves the right to change products or specifications without notice.
©2012 Micron Technology, Inc. All rights reserved.