freescale™
semiconductor

Controlled version is located in the project repository. Printed versions are uncontrolled unless stamped “Controlled Copy” in RED.

NVMe User Manual for Freescale iSSD

Product Requirement ID	N/A
Version	A0-00
Date	1/11/2017
iCAP Classification	Freescale Internal

REVIEW / OFFICIAL SIGN-OFF

Role	Name	Date

Table of Contents

1 Introduction

This User Manual describes the software setup of NVMe storage including:

- Instructions to configure Nvme storage card to work with NVMe application.
- Admin Commands supported by the NVMe application.
- IO Commands supported by the NVMe application.

1.1 Pre-requisites

Following are required to work with NVMe software setup:

- NVMe storage card
- x86 linux system with PCIe slot

NOTE: x86 linux system with3.13 or 3.19 kernel release are recommended (Ubuntu distribution is used for testing).

2 NVMe Software setup

2.1 Configuring x86 system for Storage Card

The x86 linux system going to use with NVMe storage card should be configured. Following are the instructions to configure ubuntu (distribution) based linux system for the first time only.

- Make as a root user, by issuing the following command from the terminal

```
$sudo su
```

- As a root user append the following line in the */etc/default/grub* file.

```
GRUB_CMDLINE_LINUX_DEFAULT=”mem=8G modprobe.blacklist=nvme”
```

For example,

The /etc/default/grub file has a line with Macro *GRUB_CMDLINE_LINUX_DEFAULT* like below:

```
GRUB_CMDLINE_LINUX_DEFAULT=”quiet splash”
```

Then append the word “mem=8G modprobe.blacklist=nvme” to that macro like below:

```
GRUB_CMDLINE_LINUX_DEFAULT=”quiet splash mem=8G modprobe.blacklist=nvme”
```

- Save the file and exit.
- As a root user update the grub by issuing the following command from the terminal

```
#update-grub
```

- **Restart the x86 system to reflect the changes done**.

NOTE: The above commands works with ubuntu (distribution) based linux (x86 system) system.

2.1.1 NVMe Host driver

The host driver (nvme driver) has to be compiled manually to work with NVMe application. Host drivers differs for different linux kernel versions. The recommended kernel verisons are 3.13 and 3.19. Host driver for both versions are shared as a tar file.

Take the host driver tar file according to the linux kernel version, extract the tar file, compile the driver files and load the host driver. Following are the instructions to do it.

- To check the linux kernel release verison, issue following command from the terminal

```
$uname –r
```

With the help of kernel release information, choose the host driver tar starts with same kernel version from the shared drivers (as a tar).

- Extract the tar by issuing the below command from the terminal

For kernel release 3.13,

```
$tar –xvzf 3.13_host_nvme_driver.tar.gz
```

For kernel release 3.19,

```
$tar –xvzf 3.19_host_nvme_driver.tar.gz
```

When extract is completed, a directory appears with same name of tar file but without .tar.gz extension, that directory contains driver files.

For example,

3.19_host_nvme_driver

NOTE: Taking 3.19 version as example to explain the remaining steps. Below steps are also applicable to 3.13 version only difference is version number (3.19 should be replaced with 3.13).

- Go to the directory by issuing the following command

```
$cd 3.19_host_nvme_driver
```

- To see the list of files present in that directory, issue the below command, and files will be listed are: nvme-core.c and nvme-scsi.c

```
$ls
```

- Should compile the driver as a root user by issuing the below command

```
$sudo make && echo SUCCESS
```

- This combination of commands just displays SUCCESS message on terminal after a successful make. Or else SUCCESS is not printed. In case of errors, it is advised to verify the kernel verisons. After the successful compilation, that directory contains additional files. These files are generated as a result of compilation.
- Use the **ls** command to list those files. Specifically, **nvme.ko** file is generated (if it is not then driver has not been compiled properly).
- Now the host nvme driver module can be loaded using generated nvme.ko file by issuing the following command.

```
$sudo insmod nvme.ko
```

2.2 Creating filesystem for Storage card

Partitions can be created in the storage card using **fdisk** command with option “n” .

The device node will be under */dev*. To check that, use the following command and hit <TAB>

```
$ls /dev/nvme
```

The device node should be detetced as ***/dev/nvme0n1.***

This device node **CANNOT BE USED DIRECTLY** because it does not contains any filesystem the first time. To make it usable, create filesystem by issuing the following command

```
$sudo mkfs.ext2 /dev/nvme0n1
```

NOTE: Filesystem can be anything like ext2,ext4,vfat,etc...

Filesystem should be created for partitions too.

For example,

```
$sudo mkfs.ext2 /dev/nvme0n1p1
```

2.3 Admin commands

The Admin commands supported by the NVMe application are listed below:

- Delete I/O Submission Queue
- Create I/O Submission Queue
- Get Log Page
- Delete I/O Completion Queue
- Create I/O Completion Queue
- Identify
- Abort
- Set Features
- Get Features
- Asynchronous Event Request

2.4 IO commands

The IO commands supported by the NVMe application are listed below:

- Read command
- Write command
- Flush command

NOTE: Host driver will issue Read/Write commands accordingly based on files copying from the storage card to x86 or files copying from x86 to storage card in GUI.

2.5 Safe Removal of NVMe Partition

Once all the operations have been performed and the user needs to remove the memory device, it can be done like normal memory drive removal from GUI (Right click on device and select unmount).

For removing the nvme driver:

```
$sudo rmmod nvme
```

3 Additional Info

1.The Host outbound Address differ for both LS2088 and LS2085.According to your setup,change the

HOST_OUTBOUND_ADDR macro in common.h

For LS2085: #define HOST_OUTBOUND_ADDR 1400000000

For LS2088: #define HOST_OUTBOUND_ADDR 3000000000

2.Insert the module fsl-dpaa2-qdma and qdma_client in LS2 before starting issd_nvme application.

Example:

- modprobe fsl-dpaa2-qdma
- modprobe qdma_client

3. Run the flash_mgnt_app in background before starting issd_nvme application.The NAND page mapping table is compressed and store in SD card(while rmmod of nvme driver in host).

4. Some NAND DIMM will have EEPROM and some DIMM doesnt have it. To skip the i2c detection for DIMM without EEPROM,the runtime argument is used with issd_nvme application.

Example:

./issd_nvme 0 ===> For NAND DIMM with EEPROM

./issd_nvme 1 ===> For NAND DIMM without EEPROM