FTL NOR RAM Flash Driver User Guide

Version 1.20

For use with FTL NOR RAM Flash Driver versions 2.05 and above

Date: 18-Aug-2017 11:46

Table of Contents

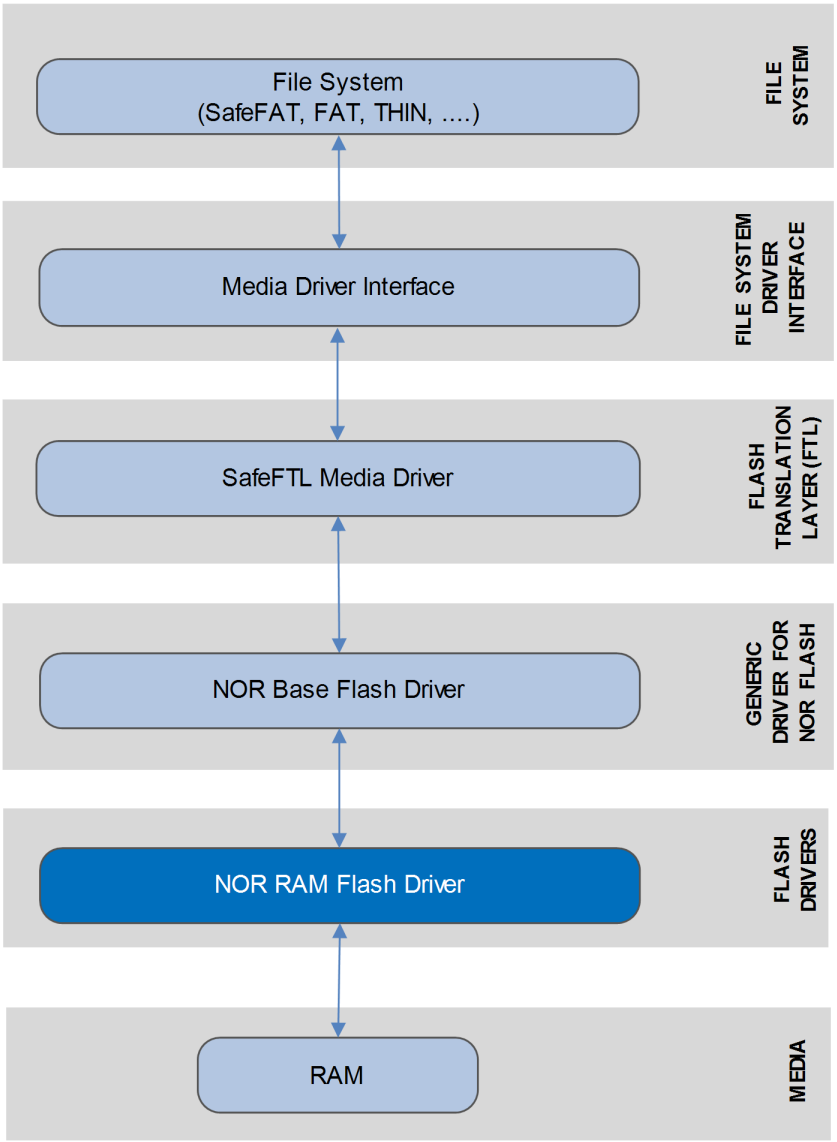
1 System Overview

1.1 Introduction

This guide is for those who wish to implement a NOR RAM flash driver. It covers all aspects of configuration and use. Read the guide thoroughly before implementing a driver.

An instance of the NOR RAM flash driver controls a drive in the RAM of the target system. The NOR RAM flash driver interfaces to the NOR base flash driver. It fully conforms to the HCC Media Driver Interface Specification.

This diagram shows how the NOR RAM flash driver fits into the system.



Note the following:

- The file system can be any HCC file system that addresses logical sector arrays (including SafeFAT, FAT, and THIN).
- The Flash Translation Layer (FTL), a system for attaching arrays of flash to a media driver, is the SafeFTL media driver. This has its own manual.
- The NOR base flash driver is a generic driver that handles the NOR flash drivers. Many different NOR flash drives can be attached simultaneously. The NOR base flash driver is designed for use with most standard types of NOR flash, in simple or complex configurations. The NOR base flash driver has its own manual.
- An instance of the NOR RAM flash driver controls a NOR flash drive connected to the system.
- For each NOR flash drive, an entry must be added to the SafeFTL drive list.

The NOR layer is assumed to support partial page writes.

1.2 Feature Check

For a full list of SafeFTL features, see the *HCC SafeFTL User Guide*.

The NOR RAM flash driver can handle a drive in virtually any area of RAM defined for it.

1.3 Packages and Documents

Packages

The table below lists the packages that you need in order to use this module:

Package	Description
hcc_base_doc	This contains the two guides that will help you get started.
media_drv_ftl_base	The base SafeFTL package.
media_drv_ftl_nor_base	The base flash driver used by all NOR flash devices. This enables the NOR flash to be managed by SafeFTL.
media_drv_ftl_nor_ram	The package described in this document.

Documents

For an overview of HCC file systems and flash management technologies, see Product Information on the main HCC website.

Readers should note the points in the HCC Documentation Guidelines on the HCC documentation website.

HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

HCC Media Driver Interface Guide

This document describes the specification for the upper layer interface that SafeFTL uses. This means that SafeFTL can be used as a set of drives by any file system using this Media Driver Interface standard.

HCC SafeFTL User Guide

The user's guide for HCC's Safe Flash Translation Layer, SafeFTL.

HCC FTL NOR Base Flash Driver User Guide

This describes the NOR base flash driver that handles the NOR flash driver.

HCC FTL NOR RAM Flash Driver User Guide

This is this document.

1.4 Change History

This section describes past changes to this manual.

- To download earlier manuals, see Archive: FTL NOR RAM Flash Driver User Guide.
- For the history of changes made to the package code itself, see History: media_drv_ftl_nor_ram.

The current version of this manual is 1.20. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.20	2017-08-18	2.05	Updated *Packages* list.
1.10	2017-06-27	2.05	New *Change History* format.
1.00	2016-10-03	2.05	First online version.

2 Source File List

This section describes all the FTL Media Driver NOR-RAM source code files. These files follow the HCC Embedded standard source tree system, described in the *HCC Source Tree Guide*. All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

> **Note:** Do not modify any files except the configuration file.

2.1 API Header File

The file **src/api/api_ftl_nor_ram.h** is the only file that should be included by an application using this module. For details of the API functions, see Application Programming Interface.

2.2 Configuration File

The file **src/config/config_ftl_nor_ram.h** contains all the configurable parameters. Configure these as required. For details of these options, see Configuration Options.

2.3 Source Code File

The file **src/media-drv/ftl/drivers/nor/ram/nor_ram.c** is the main source code file. **This file should only be modified by HCC**.

2.4 Version File

The file **src/version/ver_ftl_nor_ram.h** contains the version number of this module. This version number is checked by all modules that use this module to ensure system consistency over upgrades.

3 Configuration Options

Set the system configuration options in the file **src/config/config_ftl_nor_ram.h**. This section lists the available configuration options and their default values.

In general it is best to retain the default settings for most configurable parameters, but you may need to change some of them to meet your particular configuration needs.

3.1 Available Configuration Options

Emulated Physical Parameters

The first four options are emulated physical parameters of the SafeFTL NOR-RAM driver:

RAM_NOR_PAGE_SIZE

The RAM NOR page size. The default is 256.

RAM_NOR_BLOCK_SIZE

The RAM NOR block size. The default is 4096.

RAM_NOR_PAGE_PER_BLOCK

The number of pages per erasable block. The default is 16.

RAM_NOR_NUM_BLOCKS

The number of erasable blocks in the target flash array. The default is 1024.

Other Options

The remaining options are configuration settings which are passed to SafeFTL by FTL-NOR:

RAM_FTL_NOR_PAGE_DATA_SIZE

The data area available on one page. Set this to 512, 1024, or 2048 bytes, depending on the target flash device. The default is 512.

RAM_FTL_NOR_PAGE_TOTAL_SIZE

The total size of the page. The default is 528.

RAM_FTL_NOR_PAGE_PER_BLOCK

The number of pages in a block. The default is 7.

RAM_FTL_NOR_FREE_BLOCK_AVAILABLE

The number of free management blocks, including map, log and free blocks. These blocks absorb created bad blocks so increase their number in proportion to the size of the flash drive. The default is 24. The HCC reference drivers contain tested settings for this value.

RAM_FTL_NOR_LOG_BLOCK_AVAILABLE

The number of free log blocks available. The default is 47; do not set it below this. To ensure the efficiency of the system, increase this number proportionally to the size of the flash drive. The HCC reference drivers contain tested settings for this value.

RAM_FTL_NOR_NUM_OF_DIF_MAPBLOCK

The number of blocks used for mapping in the system. The default is 2. The HCC reference drivers contain tested settings for this value.

More map blocks will improve system performance. The maximum useful number of map blocks that can be set is given by:

```
((Number_blocks*8/z_pagedata)+1)
```

RAM_FTL_NOR_MAPBLOCK_SHADOW

The number of map shadow blocks. The default is 4. The system may be more efficient if more map shadow blocks are used, but each additional block reduces the number of free blocks in the system. The HCC reference drivers contain tested settings for this value.

RAM_FTL_NOR_RESERVED_BLOCKS

The number of reserved blocks, the blocks at the start of the flash area that driver should not use. The default is 0.

RAM_FTL_NOR_WEAR_STATIC_LIMIT

The maximum value that the difference between the maximum and minimum wear count can be. The default is 1024.

RAM_FTL_NOR_WEAR_STATIC_COUNT

The number of merge operations after which static wear checking must be run. The default is 128.

RAM_NOR_ERASE_CYCLES

Set this to a non-zero value to simulate ageing of blocks. The default is 1999.

3.2 Restrictions

The settings must satisfy the following expressions:

- RAM_FTL_NOR_FREE_BLOCK_AVAILABLE > MDRIVER_FTL_MAX_FREE_BLOCKS
- RAM_FTL_NOR_PAGE_PER_BLOCK > MDRIVER_FTL_MAX_PAGE_PER_BLOCK
- RAM_FTL_NOR_LOG_BLOCK_AVAILABLE > MDRIVER_FTL_MAX_LOG_BLOCK_AVAIL
- RAM_FTL_NOR_NUM_OF_DIF_MAPBLOCK < 1 || RAM_FTL_NOR_NUM_OF_DIF_MAPBLOCK > 16
- RAM_FTL_NOR_MAPBLOCK_SHADOW < 1
- RAM_FTL_NOR_FREE_BLOCK_AVAILABLE - (RAM_FTL_NOR_NUM_OF_DIF_MAPBLOCK * RAM_FTL_NOR_MAPBLOCK_SHADOW + 1) - RAM_FTL_NOR_LOG_BLOCK_AVAILABLE < 8
- RAM_FTL_NOR_FREE_BLOCK_AVAILABLE * 2 > RAM_FTL_NOR_PAGE_DATA_SIZE / 2

4 Application Programming Interface

This section documents the Application Programming Interface (API). It includes all the functions that are available to an application program.

4.1 nor_ram_init

This function initializes the driver for this device.

This is normally called automatically from SafeFTL, using its table of flash drives. Refer to the *HCC SafeFTL User Guide* for details.

Format

```
t_ftl_ret nor_ram_init (
    uint32_t               driver_param,
    t_ftl_nor_driver * *   pps_ftl_nor_driver )
```

Arguments

Argument	Description	Type
driver_param	The number of the drive to initialize.	uint32_t
pps_ftl_nor_driver	A pointer to a *t_ftl_nor_driver* structure.	t_ftl_nor_driver * *

Return values

Return value	Description
NOR_ST_OK	Successful execution.
NOR_ST_ERROR	Operation failed.

4.2 SafeFTL Flash Drive Structure Example

SafeFTL uses a flash drive structure containing all the available flash drives. Each available flash driver must have an entry in this table, specifying its initialization function and the parameter to be passed to it in that function. The flash drives are numbered from 0 to (MDRIVER_FTL_MAX_DRIVE-1). The index to this table is used to reference the flash drive.

This structure is held in the main SafeFTL package's **src/config/config_mdriver_ftl.c** file.

The following example shows how a NOR RAM drive appears in this structure. In this case the NOR RAM drive is preceded by a NAND RAM drive and followed by a NOR base driver.

```
t_ftldrive_init  as_ftldrive_init[MDRIVER_FTL_MAX_DRIVE] =
{
  { nand_ram_init, 0U }
  , { nor_ram_init, 0U }
  , { ftl_nor_init, 0U }
};
```

4.3 Error Codes

The possible return codes are shown in the table below:

Code	Value	Description
NOR_ST_OK	0	Successful execution.
NOR_ST_ERROR	2	Operation failed.

4.4 Types and Definitions

Structure t_ftl_nor_phy

This describes the real media under the FTL NOR RAM driver.

Element	Type	Description
n_pageperblock_nor	uint32_t	RAM NOR pages per block.
sz_page_nor	uint32_t	RAM NOR page size.

Structure t_ftl_nor_driver

The **nor_ram_init()** function returns a pointer to a *t_ftl_nor_driver* structure.

Element	Type	Description
user_data	uint32_t	User-defined data.
pf_nor_getphy	(* pf_nor_getphy)	Pointer to the **nor_getphy** function.
pf_nor_read_page	(* pf_nor_read_page)	Pointer to the **nor_read_page** function.
pf_nor_write_page	(* pf_nor_write_page)	Pointer to the **nor_write_page** function.
pf_nor_erase_block	(* pf_nor_erase_block)	Pointer to the **nor_erase_block** function.

Structure t_ftl_nor_init

The *t_ftl_nor_init* elements in *as_ftl_nor_init[]* enumerate the configured FTL-NOR drivers.

Element	Type	Description
pf_ll_nor_init	t_pf_ll_nor_init	The initfunc for the NOR driver if pf_ll_init was **ftl_nor_init()**.
ll_nor_param	uint32_t	The driver_param for the NOR driver.

5 Integration

This section describes all aspects of the module that require integration with your target project. This includes porting and configuration of external resources.

5.1 PSP Porting

The Platform Support Package (PSP) is designed to hold all platform-specific functionality, either because it relies on specific features of a target system, or because this provides the most efficient or flexible solution for the developer.

The module makes use of the following standard PSP functions:

Function	Package	Package	Description
psp_memcpy()	psp_base	psp_string	Copies a block of memory. The result is a binary copy of the data.
psp_memset()	psp_base	psp_string	Sets the specified area of memory to the defined value.