

FTL NOR Media Driver for Microchip SST Serial Flash User Guide

Version 1.40

For use with FTL NOR Driver for Microchip SST Serial
Flash versions 1.03 and above

Date: 18-Aug-2017 12:06

All rights reserved. This document and the associated software are the sole property of HCC Embedded. Reproduction or duplication by any means of any portion of this document without the prior written consent of HCC Embedded is expressly forbidden.

HCC Embedded reserves the right to make changes to this document and to the related software at any time and without notice. The information in this document has been carefully checked for its accuracy; however, HCC Embedded makes no warranty relating to the correctness of this document.

Table of Contents

System Overview	3
Introduction	3
Feature Check	5
Device Compatibility	6
Packages and Documents	7
Packages	7
Documents	7
Change History	8
Source File List	9
API Header File	9
Configuration File	9
Source File	9
Version File	9
Configuration Options	10
Device ID	10
Real Physical Parameters	10
Other Options	11
Restrictions	12
Application Programming Interface	13
sst25vf032_spi_init	13
SafeFTL Flash Drive Structure Example	14
Error Codes	15
Types and Definitions	16
Structure t_ftl_nor_phy	16
Structure t_ftl_nor_driver	16
Structure t_ftl_nor_init	16
Integration	17
PSP Porting	17

1 System Overview

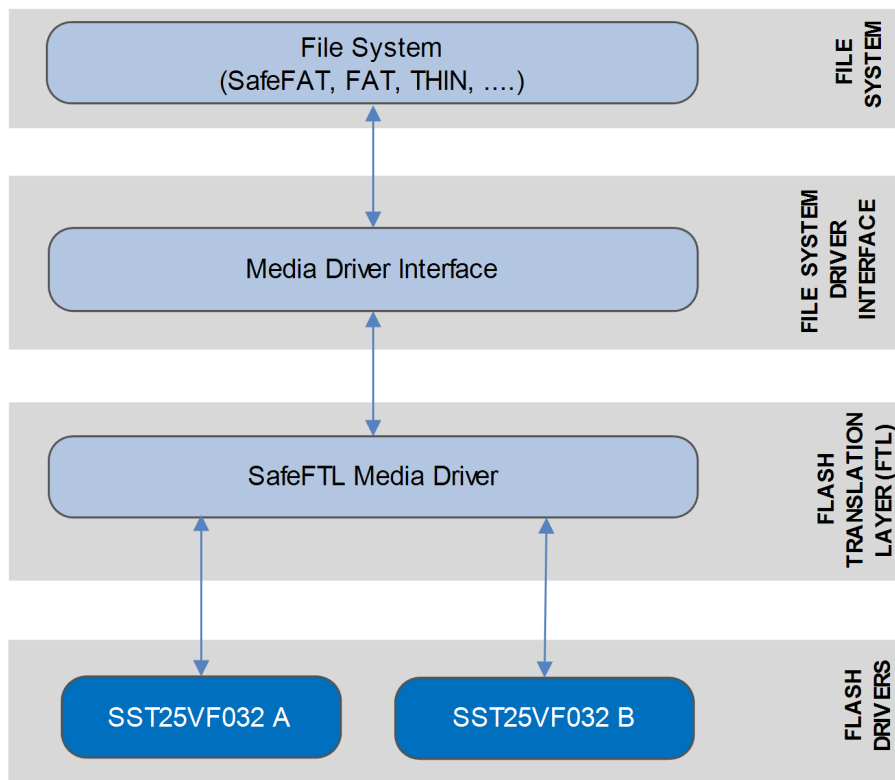
1.1 Introduction

This guide is for those who wish to use HCC's FTL NOR Media Driver for Microchip SST Serial Flash with the HCC FTL. Once deployed, the FTL provides a deterministic interface for the NOR flash and manages all aspects of the flash, including wear leveling. The SST NOR flash devices have a Serial Physical Interface (SPI) and are built by Microchip Technology Inc.

This guide covers all aspects of configuration and use. Read it thoroughly before implementing a driver.

Each instance of the flash driver controls one NOR flash drive connected to the system. The flash driver interfaces to the NOR base flash driver. It fully conforms to the [HCC Media Driver Interface Specification](#).

This diagram shows how the flash driver fits into the system. This example shows two SST25VF032 devices attached:



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. See the [HCC SafeFTL User Guide](#) for details.

- 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 Microchip SST Serial 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.

1.2 Feature Check

For a full list of SafeFTL features, see the [HCC SafeFTL User Guide](#).

The features of this media driver are as follows:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Conforms to the low level NOR flash interface defined by HCC's SafeFTL.
- Uses HCC's abstraction of Serial Physical Interface (SPI), *psp_spi*, so can be used with any SPI driver.
- Supports multiple Microchip SST serial Flash devices.
- With minor modification of the configuration parameters, it can work with any Microchip SST serial Flash device.

1.3 Device Compatibility

This driver is specifically tested and validated with the SST25VF032 serial flash device. In addition, note that:

- With minor modification, the driver should work with all SST25xx032 devices, including the SST25LF032, SST25VF032, SST25PF032, SST25WF032 and SST25VF064C.
- With minor modification, the driver should work with all SST26xyyy devices, including the SST26VF016, SST26VF016B, SST26VF032, SST26VF064 and SST26VF064B.
- With minor modification of the configuration parameters, the driver should also work with all other sizes of SST25xyyy and SST26xyyy series serial flash.

Note: HCC Embedded provides a porting and testing service if a proven driver is required in your configuration.

This table summarizes the relevant properties of the Microchip® SST25xyyy and SST26xyyy device types:

Name	Description
SST	Microchip prefix
25/26	Product family 25 = SPI flash memory. These have a clock frequency of 80MHz. Product family 26 = latest family of SPI flash memory. These have a clock frequency of 104MHz and support Serial Flash Discoverable Parameters (SFDP).
xx	LF Supply voltage = 3.0V to 3.6V PF Supply voltage = 2.3V to 3.6V VF Supply voltage = 2.7 - 3.6V WF Supply voltage = 1.65V to 1.95V
yyy	016 = Density 16Mb. 032 = Density 32Mb. 064 = Density 64Mb.

1.4 Packages and Documents

Packages

All HCC software is delivered as a set of modular 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_sst25vf032_spi	The package described in this document.
psp_template_spi	The SPI Platform Support Package (PSP).

Documents

For an overview of HCC file systems and data storage, 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 Media Driver for Microchip SST Serial Flash User Guide

This is this document.

1.5 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see [Archive: FTL NOR Media Driver for Microchip SST User Guide](#).
- For the history of changes made to the package code itself, see [History: media_drv_ftl_nor_sst25vf032_spi](#).

The current version of this manual is 1.40. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.40	2017-08-18	1.03	Updated <i>Packages</i> list.
1.30	2017-06-21	1.03	New <i>Change History</i> format.
1.20	2017-04-28	1.03	New configuration options. Renamed manual.
1.10	2015-10-29	1.03	Added <i>Device Compatibility</i> section.
1.00	2015-10-26	1.03	First online version, named: <i>HCC FTL NOR Media Driver for SST25VF032 SPI User Guide</i>

2 Source File List

This section describes all the source code files included in the system. 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_sst25vf032_spi.h` is the only file that should be included by an application using this module. For details of the single API function, see [Application Programming Interface](#).

2.2 Configuration File

The file `src/config/config_ftl_nor_sst25vf032_spi.h` contains all the configurable parameters. Configure these as required. For details of these options, see [Configuration Options](#).

2.3 Source File

The file `src/media-driv/ftl/drivers/nor/sst/nor_sst25vf032_spi.c` is the source code file. **This file should only be modified by HCC.**

2.4 Version File

The file `src/version/ver_ftl_nor_sst25vf032_spi.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_sst25vf032_spi.h`. This section lists the available configuration options and their default values.

3.1 Device ID

SST25VF032_NOR_ID_MFG

The manufacturer ID. The default is 0xBF.

SST25VF032_NOR_ID_TYPE

The device type. The default is 0x25.

SST25VF032_NOR_ID_CAP

The capacity. The default is 0x4A.

3.2 Real Physical Parameters

These options are emulated physical parameters of the driver:

SST25VF032_NOR_PAGE_SIZE

The NOR page size. The default is 256.

SST25VF032_NOR_BLOCK_SIZE

The NOR block size. The default is 4096.

SST25VF032_NOR_PAGE_PER_BLOCK

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

SST25VF032_NOR_NUM_BLOCKS

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

3.3 Other Options

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

SST25VF032_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.

SST25VF032_FTL_NOR_PAGE_TOTAL_SIZE

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

SST25VF032_FTL_NOR_PAGE_PER_BLOCK

The number of pages in a block. This must not be greater than MDRIVER_FTL_MAX_PAGE_PER_BLOCK. The default is 7.

SST25VF032_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 18.

This must not be greater than MDRIVER_FTL_MAX_FREE_BLOCKS. The HCC reference drivers contain tested settings for this value.

SST25VF032_FTL_NOR_LOG_BLOCK_AVAILABLE

The number of free log blocks available. The default is 5; do not set it below this. This must not be greater than MDRIVER_FTL_MAX_LOG_BLOCK_AVAIL.

To ensure the efficiency of the system, increase this number in proportion to the size of the flash drive. The HCC reference drivers contain tested settings for this value.

SST25VF032_FTL_NOR_NUM_OF_DIF_MAPBLOCK

The number of blocks used for mapping in the system. The default is 2. The valid range is 1 to 16.

Using more map blocks improves system performance. The maximum useful number of map blocks that can be set is given by $((\text{Number_blocks} * 8 / z_pagedata) + 1)$. The HCC reference drivers contain tested settings for this value.

SST25VF032_FTL_NOR_MAPBLOCK_SHADOW

The number of map shadow blocks. The default is 2. This value cannot be 0.

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.

SST25VF032_FTL_NOR_RESERVED_BLOCKS

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

SST25VF032_FTL_NOR_WEAR_STATIC_LIMIT

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

SST25VF032_FTL_NOR_WEAR_STATIC_COUNT

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

3.4 Restrictions

The following value must be 8 or greater:

$$\begin{aligned} & \text{SST25VF032_FTL_NOR_FREE_BLOCK_AVAILABLE} - \\ & (\text{SST25VF032_FTL_NOR_NUM_OF_DIF_MAPBLOCK} * \text{SST25VF032_FTL_NOR_MAPBLOCK_SHADOW} + \\ & 1) - \\ & \text{SST25VF032_FTL_NOR_LOG_BLOCK_AVAILABLE} \end{aligned}$$
$$\begin{aligned} & \text{SST25VF032_FTL_NOR_FREE_BLOCK_AVAILABLE} * 2 \text{ must not be greater than} \\ & \text{SST25VF032_FTL_NOR_PAGE_DATA_SIZE} / 2 \end{aligned}$$

4 Application Programming Interface

This section documents the Application Programming Interface (API). It describes the single function and the structures used.

4.1 sst25vf032_spi_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 sst25vf032_spi_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 <i>t_ftl_nor_driver</i> structure.	<i>t_ftl_nor_driver</i> * *

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_mdriber_ftl.c** file.

The following example shows how an SST25VF032 drive would appear in this structure. In this case it is the first NOR drive, preceded by a NAND RAM drive.

```
t_ftldrive_init as_ftldrive_init[MDRIVER_FTL_MAX_DRIVE] =
{
  { nand_ram_init, 0U }
  , { sst25vf032_spi_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 SST25VF032 SPI driver.

Element	Type	Description
<code>n_pageperblock_nor</code>	<code>uint32_t</code>	NOR pages per block.
<code>sz_page_nor</code>	<code>uint32_t</code>	NOR page size.

Structure `t_ftl_nor_driver`

The `nor_sst25vf032_spi_init()` function returns a pointer to a `t_ftl_nor_driver` structure.

Element	Type	Description
<code>user_data</code>	<code>uint32_t</code>	User-defined data.
<code>pf_nor_getphy</code>	<code>(* pf_nor_getphy)</code>	Pointer to the <code>nor_getphy()</code> function.
<code>pf_nor_read_page</code>	<code>(* pf_nor_read_page)</code>	Pointer to the <code>nor_read_page()</code> function.
<code>pf_nor_write_page</code>	<code>(* pf_nor_write_page)</code>	Pointer to the <code>nor_write_page()</code> function.
<code>pf_nor_erase_block</code>	<code>(* pf_nor_erase_block)</code>	Pointer to the <code>nor_erase_block()</code> 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
<code>pf_ll_nor_init</code>	<code>t_pf_ll_nor_init</code>	The <code>initfunc</code> for the NOR driver if <code>pf_ll_init</code> was <code>ftl_nor_init()</code> .
<code>ll_nor_param</code>	<code>uint32_t</code>	The <code>driver_param</code> 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	Element	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.

The module makes use of the following standard PSP SPI functions. For details of these, see the [HCC SPI Driver PSP User Guide](#).

Function	Package	Element	Description
psp_spi_init()	psp_base	psp_spi	Initializes the SPI port.
psp_spi_start()	psp_base	psp_spi	Starts the SPI port.
psp_spi_cs_hi()	psp_base	psp_spi	Sets chip select high.
psp_spi_cs_lo()	psp_base	psp_spi	Sets chip select low.
psp_spi_set_baudrate()	psp_base	psp_spi	Sets the baudrate.
psp_spi_rx()	psp_base	psp_spi	Receives a number of bytes.
psp_spi_rx1()	psp_base	psp_spi	Receives a single byte.
psp_spi_tx1()	psp_base	psp_spi	Transmits one byte.