

SafeFLASH NAND Driver for Micron MT29F2G16 User Guide

Version 1.20

For use with SafeFLASH NAND Driver for Micron
MT29F2G16 versions 1.05 and above

Date: 31-Aug-2017 15:54

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

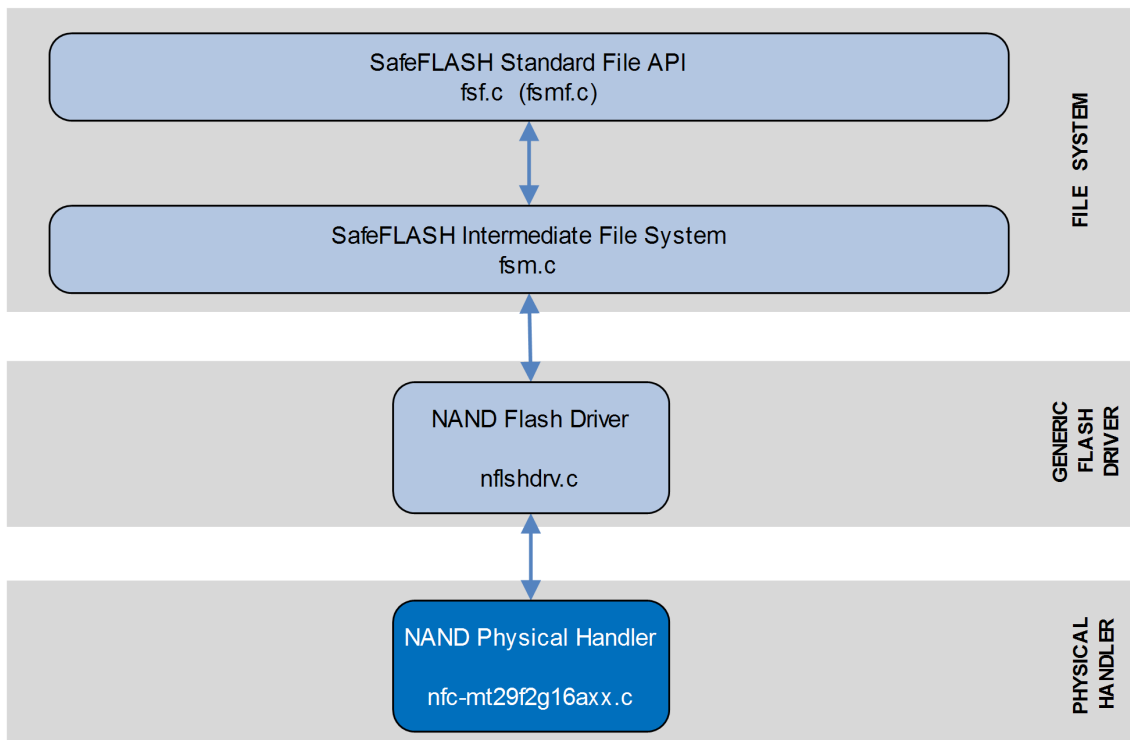
System Overview	3
Introduction	3
Feature Check	4
Fail-safety	5
Device Description	5
Packages and Documents	6
Packages	6
Documents	6
Change History	7
Source File List	8
System Files	8
Version File	8
Platform Support Package (PSP) Files	8
PSP Porting	10
psp_nfc_init	10

1 System Overview

1.1 Introduction

This guide is for those who want to implement an HCC Embedded SafeFLASH NAND driver for MT29F2G16 and similar flash devices from Micron[®] Technology Inc. This is for use with HCC's SafeFLASH file system.

The following diagram illustrates the structure of the file system software:



The main SafeFLASH package provides the file API and intermediate file system. This is described in the [HCC SafeFLASH File System User Guide](#). This design is highly portable and gives excellent performance. The basic device architecture includes a high level driver for each general media type that shares some common properties. This driver handles issues of FAT maintenance, wear leveling, and so on.

Also shown in this diagram are the following:

- The NAND Flash driver – the device driver for the NAND flash, provided by the base NAND package. This is described in the [HCC SafeFLASH File System NAND Drive User Guide](#).
- The NAND physical handler – provided by this module, this performs the translation between the driver and the physical flash hardware.

Note: HCC Embedded offers hardware and firmware development consultancy to assist developers with the implementation of flash file systems.

1.2 Feature Check

The main features of the module are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Supports Micron[®] MT29F2G16 NAND flash and is easily configurable for similar NAND flash devices.
- Supports static and dynamic wear leveling.
- Provides bad block management.

1.3 Fail-safety

This driver for Micron MT29F2G16 NAND flash is designed as part of HCC's SafeFLASH file system. SafeFLASH guarantees a defined level of fail-safety (see the [SafeFLASH File System User Guide](#)). For the system to be able to guarantee fail-safety, each component must provide a defined quality of service.

For this driver the following must be guaranteed to ensure the system is fail-safe:

- All write operations must be committed to flash in the sequence in which they are provided to the driver.
- Any write operation that fails must return an error.
- Any erase operation that fails must return an error.
- The system must ensure that there is at most one partially complete write or erase operation. At this point the file system should be restarted so that it can be recovered.

To achieve this in practice, the target hardware should ensure that in the event of a falling voltage the system resets or signals when the level approaches the specified programming level of the flash chip and inhibits further flash access.

There are other ways to manage this, for instance by adding a capacitance to ensure power is still available to complete an operation after a hardware error or reset condition is detected.

By using these techniques, the system can guarantee correct operation even after an unexpected system reset.

1.4 Device Description

For Micron devices, the digit after the "F" gives the size in Gb, that is 2 Gb.

The address lines are multiplexed with the data input/output signals on a multiplexed x16 input/output bus.

1.5 Packages and Documents

Packages

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

Package	Description
<code>hcc_base_doc</code>	This contains the two guides that will help you get started.
<code>fs_safe_nand</code>	The SafeFLASH NAND flash driver.
<code>fs_safe_nand_drv_nfc</code>	The SafeFLASH NAND driver for Micron MT29F2G16 package described in this document.

Documents

For an overview of HCC file systems and guidance on choosing a file system, 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 SafeFLASH File System User Guide

This document describes the base SafeFLASH System.

HCC SafeFLASH File System NAND Drive User Guide

This document describes the SafeFLASH NAND driver that is used with the Micron driver.

HCC SafeFLASH NAND Driver for Micron MT29F2G16 User Guide

This is this document.

1.6 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see [Archive: SafeFLASH NAND Driver for MT29F2G16 User Guide](#).
- For the history of changes made to the package code itself, see [History: fs_safe_nand_drv_nfc](#).

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	2017-08-31	1.05	Corrected <i>Packages</i> list.
1.10	2017-06-26	1.05	New <i>Change History</i> format.
1.00	2017-04-24	1.05	First online version.

2 Source File List

The following sections describe 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 PSP files.

2.1 System Files

These files are in the directory `src/safe-flash/nand/phy/micron`. **These files should only be modified by HCC.**

File	Description
<code>nfc-mt29f2g16axx.c</code>	Driver source code.
<code>nfc-mt29f2g16axx.h</code>	Driver header file.

2.2 Version File

The file `src/version/ver_safe_nand_mt29f2g16_nfc.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.

2.3 Platform Support Package (PSP) Files

There are sets of files in `psp_k70f` and `psp_mcf5441x`. These provide the `psp_nfc_init()` function that configures the hardware. Modify these files as required for your hardware.

Note: These are PSP implementations for the specific microcontroller and board; you may need to modify these to work with a different microcontroller and/or development board.

The following files are in the directory `src/psp_k70f/target`:

File	Description
<code>include/hcc_mk70f_reg.h</code>	Register definitions.
<code>nand-nfc/psp_nand-nfc.c</code>	Low level initialization function.
<code>nand-nfc/psp_nand-nfc.h</code>	Header file.

The following files are in the directory **src/psp_mcf5441x/target**:

File	Description
include/hcc_mcf5441x_reg.h	Register definitions.
nand-nfc/psp_nand-nfc.c	Low level initialization function.
nand-nfc/psp_nand-nfc.h	Header file.

3 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 files `psp_nand-nfc.c` and `psp_nand-nfc.h` define the `psp_nfc_init()` function that configures the hardware. Modify these files as required for your hardware.

3.1 psp_nfc_init

Use this function to initialize the hardware. This configures pins and clocks.

Format

```
void psp_nfc_init ( void )
```

Arguments

None.

Return Values

None.