SafeFLASH File System RAM Drive User's Guide

Version 1.10

For use with SafeFLASH File System RAM Drive Versions 1.01 and above

Date: 20-Aug-2014 16:11

Table of Contents

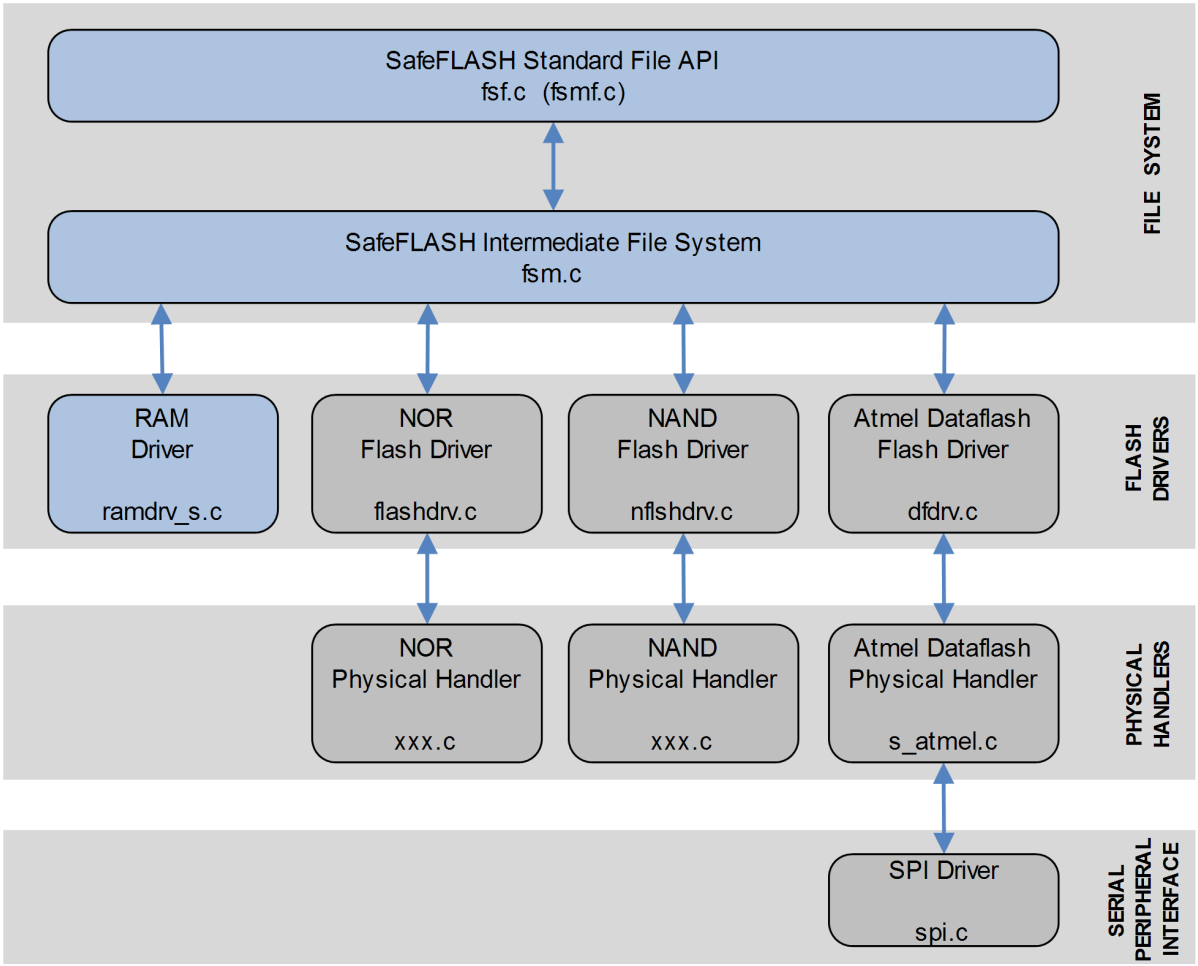
1 System Overview

1.1 Introduction

This guide is for those who wish to implement a RAM drive for HCC's SafeFLASH file system.

The SafeFLASH file system driver design is highly portable while still maintaining 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. Implementing a RAM driver for the file system is simple as there is no physical driver associated with the RAM driver.

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



In this diagram:

- The main SafeFLASH package provides the file API and intermediate file system. This is described in the *HCC SafeFLASH File System User's Guide*.
- The RAM driver is the device driver. This guide shows how to add this to the build. Using the available sample drivers as a model, you can create a driver that meets your specific needs.

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

1.2 Packages and Documents

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.
fs_safe	The SafeFLASH base package.
fs_safe_ram	The SafeFLASH RAM package described in this document.

Documents

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's Guide

This document describes the base SafeFLASH System.

HCC SafeFLASH File System RAM Drive User's Guide

This is this document.

Other HCC SafeFLASH User Guides

These describe other SafeFLASH components:

- *HCC SafeFLASH System NOR Drive User's Guide* – documents the SafeFLASH system for NOR flash.
- *HCC SafeFLASH System NAND Drive User's Guide* – documents the SafeFLASH system for NAND flash.
- *HCC SafeFLASH System Atmel® DataFlash Drive User's Guide* – documents the SafeFLASH system for Atmel® DataFlash.

2 Source File List

This section lists and describes all the source code files included in the system. These files follow HCC Embedded's 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 Configuration File

The file **src/config/config_safe_ram.h** contains the configurable parameters of the system. Configure these as required. For detailed explanation of these options, see Configuration Options.

2.2 System Files

These files should only be modified by HCC.

File	Description
src/safe-flash/ram/ramdrv_s.c	RAM driver source code.
src/safe-flash/ram/ramdrv_s.h	RAM driver header file.

2.3 Version File

The file **src/version/ver_safe_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 configuration options in the file **src/config/config_safe_ram.h**. This section lists the available configuration options and their default values.

RAM_SECSIZE

The sector size. The default is 4096.

MEMCPY_LONG

Set this value to 1 (the default) if the system supports 32 bit memory access. This accelerates memory access time and results in better performance.

4 Implementing a RAM Driver

Implementing a RAM driver for the file system is simple. There is no physical driver associated with the RAM driver.

1. Include the **ramdrv_s.c** and **ramdrv_s.h** files in your file system build. This ensures that it can be mounted.
2. Call **f_init()** as shown in the example below.
3. Call **f_mountdrive()** with a pointer to the memory area and the size of the area to be used for the driver.

```
#define RAM_DRIVE_SIZE 0x1000000
void main(void)
{
    f_init();                   /* initialize the file system */
    f_enterFS();                /* get task access to the file system */
    /* mount first drive – A */
    f_mountdrive(
        0,                      /* specifies drive 'A' */
        malloc (RAM_DRIVE_SIZE), /* get required buffer pointer */
        RAM_DRIVE_SIZE,         /* size of RAM drive to be used */
        fs_mount_ramdrive,      /* ramdrive mount function (in ramdrv_s.c) */
        0                       /* no physical */
    )
}
```

The RAM drive may now be used as a standard drive.

5 File System Test

The test suite is provided for exercising the file system and ensuring that it is working correctly. Most of the operational features of the file system are exercised by this program, including file read/write/append/seek/file content, directory and file manipulation functions.

The test program requires the functions defined and implemented (as samples) in the file **testport_ram_s.c**. This is part of the **fs_safe** base package. The full path of this file is **src/safe-flash/test/testport_ram_s.c**.

Port the functions to your system. Refer to the comments and simple code for reference.

To use the test program, call the following to execute the test code:

```
void f_dotest( void )
```