EFFS-THIN

EFFS-THIN – Overview

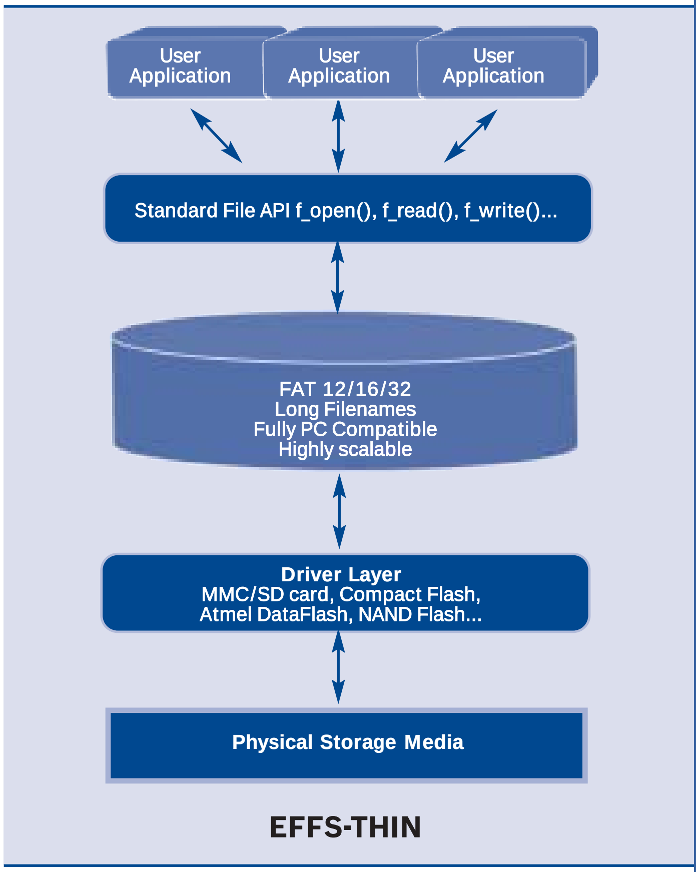
HCC-Embedded are specialist providers of embedded storage solutions.

EFFS-THIN is a full-featured file system targeted at embedded device developers – with limited resources available to them – who want to connect FAT12/16/32 compatible media to their product. The code has been highly optimized for both speed and footprint (within the constraint of having limited resource available) to allow the developer to extract the most out of their system for the minimum of effort. Additionally many build options have been included to allow the developer to easily make trade-offs between the system requirements and the available ROM/RAM and performance.

EFFS-THIN can be used on microcontrollers with <1Kbytes of available RAM!

Key Features

- FAT12/16/32
- Long FileNames
- Standard API (fopen(), fread() etc.)
- Highly Scalable
- Minimal Footprint
- Multiple Simultaneous Files Open
- Handles Media Errors
- RTOS Independent
- Royalty Free Source
- Comprehensive Programmer's Guide



RAM Footprint

RAM
<700 bytes in super-thin build for a read/write file system – including all data and stack.
For full featured build with multiple simultaneous open files <1.5K plus 0.5K for each file open.

ROM Footprint

ROM
For a read-only file system this maybe <4K and for a read/write system <9K-15K depending on options enabled.
Nb. All footprint values are typical values and depend upon the target type and compiler etc.

Drivers

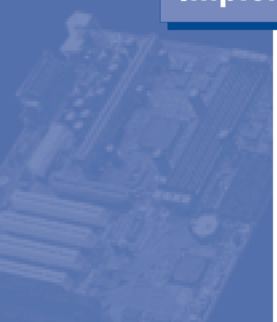
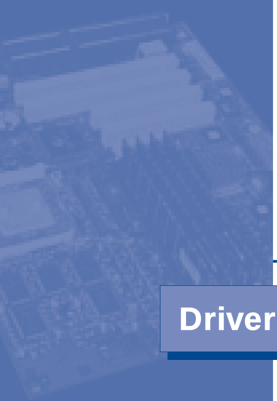
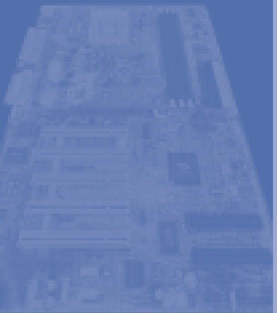
EFFS-THIN is supplied with sample drivers for:

- Compact Flash Cards – standard sample drivers supplied for all modes
- MMC/SD cards – standard sample drivers supplied for SPI mode

Also available are assembler optimized drivers for certain targets including 8051 to get the best possible performance.
Also available as additional options are drivers for:

- Atmel DataFlash® – the DFML gives a reliable interface to these devices.
- NAND Flash – either the EFFS-FTL or EFFS-SFTL may be used.

EFFS-THIN

Implementation

EFFS-THIN has been designed and crafted to run on Microcontrollers with limited resources. It has been tested and optimized on many microcontrollers including 8051, MSP430, XC16x, ARM7 and is suitable for use on most 8, 16 and 32bit microcontrollers. Also available are hardware reference designs for many of these targets – already proven with our sample drivers.

Included are many build options to enable the developer to build exactly the system they need and no more. Also there are options to trade resource for performance. For example, an option is included to accelerate free cluster searches - these can be very time consuming on large storage devices.

The detailed programmer's guide explains all the available options and their affect on the system. Typically customers have a file system up and running within hours of receiving the system!

Application Programmer's Interface

A standard file API is provided with standard call parameters. Function names have been changed to avoid library conflicts i.e. fopen() has become f_open(). These routines allow easy initialization and management of files, directories with complete wild card support. Applications written for other platforms can work without modification with EFFS-THIN. Also provided with the system is a comprehensive test suit for exercising the system to verify the correctness of the implementation.
The file API functions provided are:

```
f_getversion       f_rmdir       f_rename
f_initvolume       f_open        f_delete
f_mountdrive       f_close       f_filelength
f_format           f_write       f_findfirst
f_hardformat       f_read        f_findnext
f_getfreespace     f_seek        f_settimedate
f_setlabel         f_tell        f_gettimedate
f_getlabel         f_eof         f_setattr
f_getcwd           f_rewind      f_getattr
f_mkdir            f_putc
f_chdir            f_getc
```

Driver Interface

The interface between the file system and the physical devices is through the driver interface. This is straightforward to implement – it is complemented by a set of sample drivers and detailed documentation. The functions that must be implemented in the driver are:

`GetPhy()`	get information about device attached (e.g. number of sectors, FAT type etc.)
`GetStatus()`	get status of the device (e.g. card removed, card changed, write protect)
`ReadSector()`	read the specified sector of the attached device
`WriteSector()`	write data to the specified sector

HCC-Embedded Storage Solutions

HCC are embedded storage specialists – we care about every byte of ROM and RAM; and every CPU cycle – our systems are designed to give the developer reliable solutions with the minimum of effort. HCC has a range of products designed specifically to efficiently match different embedded storage requirements with different targets – consult our web-site for details of all our solutions.

Our pre- and post- sales technical support is second to none. They give fast and technical responses to your development issues – 95% of support enquiries get a comprehensive technical response within 24 hours.

All HCC's Embedded Products are shipped royalty free, with full 'C' source code. 6 months technical support and 1 year's free upgrades. HCC-Embedded offers a 30-day money back guarantee on all its products.

HCC also provide many other products and solutions for embedded storage – for the latest information please see our web site or contact **sales@hcc-embedded.com**.