ISO 9001 certified

embeddedsoftware
solutions

emFile®

ISO/ANSI C source code

Proprietary file system available

No royalties

File system
for embedded applications

+ + + 8 / 16 / 32 - bit + + +

Headquarter:
Email: info@segger.com
http://www.segger.com

U.S. office:
Email: info@segger-us.com
http://www.segger-us.com

embedded**software**
# solutions

SEGGER

emFile is a file system for embedded applications which can be used on any type of storage device. emFile is a high performance library that has been optimized for minimum memory consumption in RAM and ROM, high speed and versatility. It is written in ANSI-"C "and can be used on any CPU.

Basic Software Structure

emFile is organized in different layers:

▲ API Layer

The API Layer is the interface between emFile and the user application. It is subdivided in two parts, Storage API and File System API. The File System API contains file functions in ANSI C stdio style, such as FS_FOpen(), FS_FWrite() etc. The API Layer transfers these calls to the File System Layer.

▲ File System Layer

The File System Layer translates file operations to logical block (sector) operations. After such a translation, the File System calls the Storage Layer and specifies the corresponding device driver for a device.

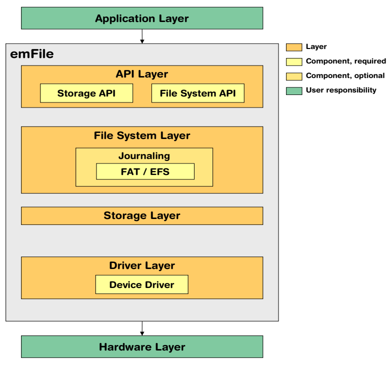
▲ Storage Layer

The main purpose of the Storage Layer is to synchronize accesses to a device driver. It furthermore provides a simple interface for the File System API. The Storage Layer calls a device driver to perform a block operation. It also contains the cache mechanism.

Start project

▲ Device Driver Layer

Device drivers are low-level routines that are used to access sectors of the device and to check status. It is hardware independent but depends on the storage medium.



Layer structure

▲ Hardware Layer

These are the low level routines to access your hardware. These routines simply read and store fixed lenght sectors. The structure of the device driver is simple in order to allow easy integration of your own hardware.

Format/Check Utilities

emFile contains utilities to verify the integrity of the file system's data structures (checkdisk functionality). The format utility allows formatting of supported medias. The formatted medias are compatible with MS DOS/Windows and can be used with PC-card readers.

Device Drivers

emFile has been designed to cooperate with any kind of embedded system and storage device. To use a specific medium with emFile, a so called device driver for that medium is required. The device driver consists of basic I/O functions for accessing the hardware and a global table, which holds pointers to these functions. If you are going to use a proprietary storage device, you can write your own device driver. Currently the following device drivers are available: Multimedia (MMC) / Secure digital (SD), RAM disk, Compact Flash / IDE, Smart Media and NOR/NAND flash.

Main Features

- MS-DOS/MS-Windows compatible FAT12, FAT16 and FAT32 support.
- Multiple device driver support. Different device drivers, which allow access to different types of hardware with the file system at the same time.
- Multiple media support. A device driver allows access to different medias at the same time.
- OS support. emFile can easily be integrated into any OS. Therefore file operations are possible in a multithreaded environment.
- ANSI C stdio.h like API for user applications. An application using standard "C"I/O library can easily be modified to use emFile.
- Very simple device driver structure. emFile device drivers need only very basic functions for reading and writing blocks. Therefore it is very simple to support any custom hardware.
- Proprietary File System available.

Memory requirements*

The memory requirements depend on the used CPU, compiler, memory model, as well as on various other factors such as configuration switches and selected drivers.

- ROM app.14-40 kb
- RAM app. 2 kb

*Precise values depend on the functionality used. Values are measured on a specific target system and will be different on other systems.