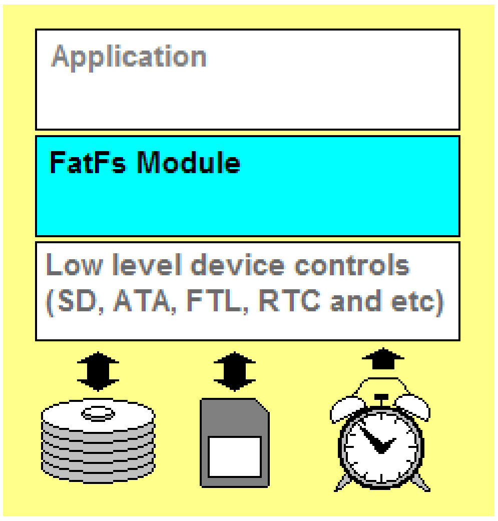
FatFs - Generic FAT File System Module

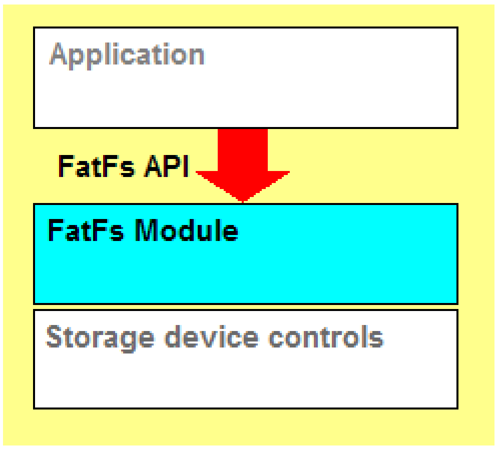


FatFs is a generic FAT file system module for small embedded systems. The FatFs module is written in compliance with ANSI C (C89) and completely separated from the disk I/O layer. Therefore it is independent of the platform. It can be incorporated into small microcontrollers with limited resource, such as 8051, PIC, AVR, ARM, Z80, 78K and etc. Also Petit FatFs module for tiny microcontrollers is available here.

Features

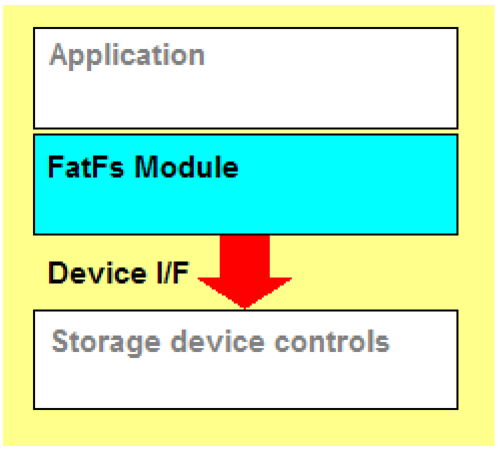
- Windows compatible FAT file system.
- Platform independent. Easy to port.
- Very small footprint for code and work area.
- Various configuration options:
  - Multiple volumes (physical drives and partitions).
  - Multiple ANSI/OEM code pages including DBCS.
  - Long file name support in ANSI/OEM or Unicode.
  - RTOS support for multi-task operation.
  - Multiple sector size support upto 4KB.
  - Read-only, minimized API, I/O buffer and etc...

Application Interface



- File Access
  - f_open - Open/Create a file
  - f_close - Close an open file
  - f_read - Read data
  - f_write - Write data
  - f_lseek - Move read/write pointer, Expand size
  - f_truncate - Truncate size
  - f_sync - Flush cached data
  - f_forward - Forward data to the stream
  - f_gets - Read a string
  - f_putc - Write a character
  - f_puts - Write a string
  - f_printf - Write a formatted string
  - f_tell - Get current read/write pointer
  - f_eof - Test for end-of-file
  - f_size - Get size
  - f_error - Test for an error
- Directory Access
  - f_opendir - Open a directory
  - f_closedir - Close an open directory
  - f_readdir - Read an item
  - f_findfirst - Open a directory and read first item found
  - f_findnext - Read a next item found
- File/Directory Management
  - f_stat - Check existance of a file or sub-directory
  - f_unlink - Remove a file or sub-directory
  - f_rename - Rename or move a file or sub-directory
  - f_chmod - Change attribute of a file or sub-directory
  - f_utime - Change timestamp of a file or sub-directory
  - f_mkdir - Create a sub-directory
  - f_chdir - Change current directory
  - f_chdrive - Change current drive
  - f_getcwd - Retrieve the current directory and drive
- Volume Management
  - f_mount - Register/Unregister a work area of a volume
  - f_mkfs - Create an FAT volume on the logical drive
  - f_fdisk - Create logical drives on the physical drive
  - f_getfree - Get total size and free size on the volume
  - f_getlabel - Get volume label
  - f_setlabel - Set volume label

Device Control Interface



Since the FatFs module is a file system layer, it is completely separated from the physical devices, such as memory card, harddisk and any type of storage devices. FatFs accesses the storage devices via a simple interface shown below. The low level device control module is not a part of FatFs module. It is provided by implementer. Also sample implementations for some platforms are available in the downloads.

- disk_status - Get device status
- disk_initialize - Initialize device
- disk_read - Read sector(s)
- disk_write - Write sector(s)
- disk_ioctl - Control device dependent features
- get_fattime - Get current time

Resources

The FatFs module is a free software opened for education, research and development. You can use, modify and/or redistribute it for personal projects or commercial products without any restriction under your responsibility. For further information, refer to the application note.

- Read first: FatFs module application note March 18, 2015
- Download: FatFs R0.11 | Updates | Patches March 9, 2015
- Download: FatFs sample projects for various platforms February 9, 2015
- Download: Old Releases
- Community: FatFs User Forum
- Read SD card with FatFs on STM32F4xx devices by Tilen Majerle↗ (Quick and easy implementation for STM32F4-Discovery)
- Nemuisan's Blog↗ (Well written implementations for STM32F/SDIO and LPC2300/MCI)
- ARM-Projects by Martin THOMAS↗ (Examples for LPC2000, AT91SAM and STM32)
- FAT32 Specification by Microsoft↗ (The authorized document on FAT file system)
- The basics of FAT file system [ja]
- How to Use MMC/SDC
- Benchmark 1 (ATmega64/9.2MHz with MMC via SPI, HDD/CFC via GPIO)
- Benchmark 2 (LPC2368/72MHz with MMC via MCI)
- Demo movie of an application (this project is in ffsample.zip/lpc23xx)

Return