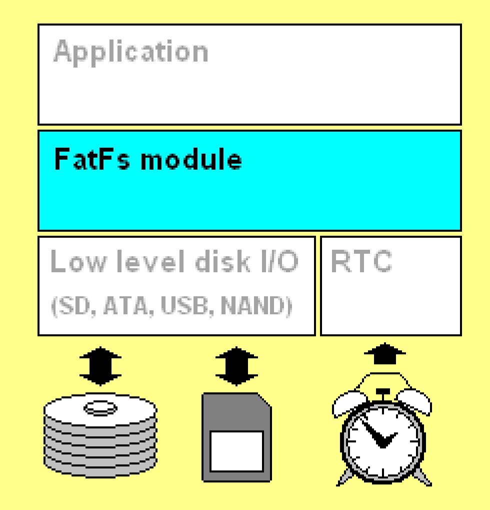
FatFs Generic FAT File System Module



FatFs is a generic FAT file system module for small embedded systems. The FatFs is written in compliance with ANSI C and completely separated from the disk I/O layer. Therefore it is independent of hardware architecture. It can be incorporated into low cost microcontrollers, such as AVR, 8051, PIC, ARM, Z80, 68k and etc..., without any change. Petit FatFs module is also 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.
  - Multiple sector size support.
  - Read-only, minimized API, I/O buffer and etc...

Application Interface

FatFs module provides following functions to the applications. In other words, this list describes what FatFs can do to access the FAT volumes.

- f_mount - Register/Unregister a work area
- f_open - Open/Create a file
- f_close - Close a file
- f_read - Read file
- f_write - Write file
- f_lseek - Move read/write pointer, Expand file size
- f_truncate - Truncate file size
- f_sync - Flush cached data
- f_opendir - Open a directory
- f_readdir - Read a directory item
- f_getfree - Get free clusters
- f_stat - Get file status

- f_mkdir - Create a directory
- f_unlink - Remove a file or directory
- f_chmod - Change attribute
- f_utime - Change timestamp
- f_rename - Rename/Move a file or directory
- f_chdir - Change current directory
- f_chdrive - Change current drive
- f_getcwd - Retrieve the current directory
- f_forward - Forward file data to the stream directly
- f_mkfs - Create a file system on the drive
- f_fdisk - Divide a physical drive
- f_gets - Read a string
- f_putc - Write a character
- f_puts - Write a string
- f_printf - Write a formatted string
- f_tell - Get the current read/write pointer
- f_eof - Test for end-of-file on a file
- f_size - Get size of a file
- f_error - Test for an error on a file

Disk I/O Interface

Since the FatFs module is completely separated from disk I/O layer, it requires following functions at least to access the physical media. When O/S related feature is enabled, it will require process/memory functions in addition. However the low level disk I/O module is not a part of FatFs module so that it must be provided by user. The sample drivers are also available in the resources.

- disk_initialize - Initialize disk drive
- disk_status - Get disk status
- 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.

- *FatFs User Forum*
- Read first: FatFs module application note September 6, 2011
- Download: FatFs R0.09 | Updates | Patches November 14, 2011
- Download: FatFs sample projects (AVR, PIC24, LPC2300, LPC1700, V850ES, H8/300H, SH-2A, RX62N, Win32 and Generic uC) January 24, 2012
- Download: Previous versions FatFs R0.08b | R0.08a | R0.08 | R0.07e | R0.07c | R0.07a | R0.07 | R0.06 | < R0.06
- 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 reference document on FAT file system)
- 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