FatFs Module Source Files R0.10b (C)ChaN, 2014

FILES

ffconf.h Configuration file for FatFs module.

ff.h Common include file for FatFs and application module.

ff.c FatFs module.

diskio.h Common include file for FatFs and disk I/O module.

diskio.c An example of glue function to attach existing disk I/O module to FatFs.

integer.h Integer type definitions for FatFs.

option Optional external functions.

Low level disk I/O module is not included in this archive because the FatFs

module is only a generic file system layer and not depend on any specific

storage device. You have to provide a low level disk I/O module that written

to control your storage device.

AGREEMENTS

FatFs module is an open source software to implement FAT file system to

small embedded systems. This is a free software and is opened for education,

research and commercial developments under license policy of following trems.

Copyright (C) 2014, ChaN, all right reserved.

\* The FatFs module is a free software and there is NO WARRANTY.

\* No restriction on use. You can use, modify and redistribute it for

personal, non-profit or commercial product UNDER YOUR RESPONSIBILITY.

\* Redistributions of source code must retain the above copyright notice.

REVISION HISTORY

Feb 26, 2006 R0.00 Prototype

Apr 29, 2006 R0.01 First release.

Jun 01, 2006 R0.02 Added FAT12.

Removed unbuffered mode.

Fixed a problem on small (<32M) patition.

Jun 10, 2006 R0.02a Added a configuration option \_FS\_MINIMUM.

Sep 22, 2006 R0.03 Added f\_rename.

Changed option \_FS\_MINIMUM to \_FS\_MINIMIZE.

Dec 11, 2006 R0.03a Improved cluster scan algolithm to write files fast.

Fixed f\_mkdir creates incorrect directory on FAT32.

Feb 04, 2007 R0.04 Supported multiple drive system. (FatFs)

Changed some APIs for multiple drive system.

Added f\_mkfs. (FatFs)

Added \_USE\_FAT32 option. (Tiny-FatFs)

Apr 01, 2007 R0.04a Supported multiple partitions on a plysical drive. (FatFs)

Fixed an endian sensitive code in f\_mkfs. (FatFs)

Added a capability of extending the file size to f\_lseek.

Added minimization level 3.

Fixed a problem that can collapse a sector when recreate an

existing file in any sub-directory at non FAT32 cfg. (Tiny-FatFs)

May 05, 2007 R0.04b Added \_USE\_NTFLAG option.

Added FSInfo support.

Fixed some problems corresponds to FAT32. (Tiny-FatFs)

Fixed DBCS name can result FR\_INVALID\_NAME.

Fixed short seek (0 < ofs <= csize) collapses the file object.

Aug 25, 2007 R0.05 Changed arguments of f\_read, f\_write.

Changed arguments of f\_mkfs. (FatFs)

Fixed f\_mkfs on FAT32 creates incorrect FSInfo. (FatFs)

Fixed f\_mkdir on FAT32 creates incorrect directory. (FatFs)

Feb 03, 2008 R0.05a Added f\_truncate().

Added f\_utime().

Fixed off by one error at FAT sub-type determination.

Fixed btr in f\_read() can be mistruncated.

Fixed cached sector is not flushed when create and close without write.

Apr 01, 2008 R0.06 Added f\_forward(). (Tiny-FatFs)

Added string functions: fputc(), fputs(), fprintf() and fgets().

Improved performance of f\_lseek() on move to the same or following cluster.

Apr 01, 2009, R0.07 Merged Tiny-FatFs as a buffer configuration option.

Added long file name support.

Added multiple code page support.

Added re-entrancy for multitask operation.

Added auto cluster size selection to f\_mkfs().

Added rewind option to f\_readdir().

Changed result code of critical errors.

Renamed string functions to avoid name collision.

Apr 14, 2009, R0.07a Separated out OS dependent code on reentrant cfg.

Added multiple sector size support.

Jun 21, 2009, R0.07c Fixed f\_unlink() may return FR\_OK on error.

Fixed wrong cache control in f\_lseek().

Added relative path feature.

Added f\_chdir().

Added f\_chdrive().

Added proper case conversion for extended characters.

Nov 03, 2009 R0.07e Separated out configuration options from ff.h to ffconf.h.

Added a configuration option, \_LFN\_UNICODE.

Fixed f\_unlink() fails to remove a sub-dir on \_FS\_RPATH.

Fixed name matching error on the 13 char boundary.

Changed f\_readdir() to return the SFN with always upper case on non-LFN cfg.

May 15, 2010, R0.08 Added a memory configuration option. (\_USE\_LFN)

Added file lock feature. (\_FS\_SHARE)

Added fast seek feature. (\_USE\_FASTSEEK)

Changed some types on the API, XCHAR->TCHAR.

Changed fname member in the FILINFO structure on Unicode cfg.

String functions support UTF-8 encoding files on Unicode cfg.

Aug 16,'10 R0.08a Added f\_getcwd(). (\_FS\_RPATH = 2)

Added sector erase feature. (\_USE\_ERASE)

Moved file lock semaphore table from fs object to the bss.

Fixed a wrong directory entry is created on non-LFN cfg when the given name contains ';'.

Fixed f\_mkfs() creates wrong FAT32 volume.

Jan 15,'11 R0.08b Fast seek feature is also applied to f\_read() and f\_write().

f\_lseek() reports required table size on creating CLMP.

Extended format syntax of f\_printf function.

Ignores duplicated directory separators in given path names.

Sep 06,'11 R0.09 f\_mkfs() supports multiple partition to finish the multiple partition feature.

Added f\_fdisk(). (\_MULTI\_PARTITION = 2)

Aug 27,'12 R0.09a Fixed assertion failure due to OS/2 EA on FAT12/16.

Changed f\_open() and f\_opendir() reject null object pointer to avoid crash.

Changed option name \_FS\_SHARE to \_FS\_LOCK.

Jan 23,'13 R0.09b Added f\_getlabel() and f\_setlabel(). (\_USE\_LABEL)

Oct 02,'13 R0.10 Added selection of character encoding on the file. (\_STRF\_ENCODE)

Added f\_closedir().

Added forced full FAT scan for f\_getfree(). (\_FS\_NOFSINFO)

Added forced mount feature with changes of f\_mount().

Improved behavior of volume auto detection.

Improved write throughput of f\_puts() and f\_printf().

Changed argument of f\_chdrive(), f\_mkfs(), disk\_read() and disk\_write().

Fixed f\_write() can be truncated when the file size is close to 4GB.

Fixed f\_open(), f\_mkdir() and f\_setlabel() can return incorrect error code.

Jan 15,'14 R0.10a Added arbitrary strings as drive number in the path name. (\_STR\_VOLUME\_ID)

Added a configuration option of minimum sector size. (\_MIN\_SS)

2nd argument of f\_rename() can have a drive number and it will be ignored.

Fixed f\_mount() with forced mount fails when drive number is >= 1.

Fixed f\_close() invalidates the file object without volume lock.

Fixed f\_closedir() returns but the volume lock is left acquired.

Fixed creation of an entry with LFN fails on too many SFN collisions.

Mar 19,'14 R0.10b Fixed a hard error in the disk I/O layer can collapse the directory entry.

Fixed LFN entry is not deleted on delete/rename an object with lossy converted SFN.