Micriμm

Empowering Embedded Systems

μC/FS

V4.04.02

Release Notes

www.Micrium.com

Revision History

Version	**Date**	**Description**
V4.04.02	2012 January	New features, changes & corrections.
V4.04.01	2011 December	Improvements, changes & corrections.
V4.04	2010 November	New features, improvements, changes & corrections.
V4.03	2010 May	New features, improvements, changes & corrections.
V4.02	2009 Jun	New features, improvements, changes & corrections.
V4.01	2009 Apr	New features, improvements, changes & corrections.
V4.00	2009 Mar	Initial release.

Required Modules

Version 4.04.02

µC/Clk	Version 1.09.01
µC/CPU	Version 1.29.00
µC/CRC	Version 1.08.02
µC/LIB	Version 1.36.01

Version 4.04.01

µC/Clk	Version 1.09.01
µC/CPU	Version 1.29.00
µC/CRC	Version 1.08.02
µC/LIB	Version 1.36.01

Version 4.04

µC/Clk	Version 1.09
µC/CPU	Version 1.27
µC/CRC	Version 1.07
µC/LIB	Version 1.33

Version 4.03

µC/CPU	Version 1.26
µC/CRC	Version 1.05
µC/LIB	Version 1.32

Version 4.02

µC/CPU	Version 1.22
µC/CRC	Version 1.04
µC/LIB	Version 1.30

Version 4.01

µC/CPU	Version 1.22
µC/CRC	Version 1.03
µC/LIB	Version 1.29

Version 4.00

µC/CPU	Version 1.22
µC/CRC	Version 1.02
µC/LIB	Version 1.28

New Features

Version 4.04.02

V4.04.02-001

Added optional support for 64-bits LBA. Configurable through define FS_CFG_64_BITS_LBA_EN in fs_cfg.h.

Version 4.04.01

None.

Version 4.04

None.

Version 4.03

V4.03-001

Added functions to get and set volume label:

`FSVol_LabelGet()`	Get volume label
`FSVol_LabelSet()`	Set volume label

V4.03-002

Added functions to get open device, directory, file and volume counts, and maximum possible device, directory, file and volume counts:

`FSDev_GetDevCnt()`	Get number of open devices.
`FSDev_GetDevCntMax()`	Get maximum possible number of open devices.
`FSDir_GetDirCnt()`	Get number of open directories.
`FSDir_GetDirCntMax()`	Get maximum possible number of open dirs.
`FSFile_GetFileCnt()`	Get number of open files.
`FSFile_GetFileCntMax()`	Get maximum possible number of open files.
`FSVol_GetVolCnt()`	Get number of open volumes.
`FSVol_GetVolCntMax()`	Get maximum possible number of open vols.

See also ‘Changes V4.03-001’.

V4.03-003

Added volume cache functions:

`FSVol_CacheAssign()`	Assign cache to a volume.
`FSVol_CacheClean()`	Clean cache on a volume.

`FSVol_CacheFlush()`	Flush cache on a volume.

These functions are enabled with the configuration:

`FS_CFG_CACHE_EN`	Enable/disable volume cache.

The file *fs_cache.c* must be included in your build if volume cache is enabled.

V4.03-004
Added caching of file entry on a file-by-file basis by OR'ing the flag `FS_FILE_ACCESS_MODE_CACHED` into the mode argument of `FSFile_Open()`. If this is used on a file opened in write mode, the directory entry update will be delayed until the file is closed.

V4.03-005
Added working directory support. Functions can be used to get/set the working directory:

`FS_WorkingDirSet()`	Set working directory.
`FS_WorkingDirGet()`	Get working directory.
`fs_chdir()`	Set working directory.
`fs_getcwd()`	Get working directory.

If working directory support is enabled, functions that accept path names (e.g., `fs_fopen()`, `FSFile_Open()`, `opendir()`, `FSDir_Open()`, `fs_rename()`, `FSEntry_Rename()`, etc.) will accept relative path names. This functionality is enabled with the configuration:

`FS_CFG_WORKING_DIR_EN`	Enable/disable working directories.

V4.03-006
Added FAT12 support.

Version 4.02

V4.02-001

Added driver for parallel and serial NOR flash devices:

`FSDev_NOR`	NOR driver
`FSDev_NOR_STM25`	Physical-layer driver for ST M25 serial NOR
`FSDev_NOR_SST25`	Physical-layer driver for SST SST25 serial NOR
`FSDev_NOR_STM39`	Physical-layer driver for SST SST39 parallel NOR
`FSDev_NOR_AMD_1X08`	Physical-layer driver for CFI-compatible NOR using AMD instruction set on 8-bit bus.
`FSDev_NOR_AMD_1X16`	Physical-layer driver for CFI-compatible NOR using AMD instruction set on 16-bit bus.
`FSDev_NOR_Intel_1X08`	Physical-layer driver for CFI-compatible NOR using Intel instruction set on 8-bit bus.
`FSDev_NOR_Intel_1X16`	Physical-layer driver for CFI-compatible NOR using Intel instruction set on 16-bit bus.

V4.02-006

Added application time get/set functions:

`FSTime_GetTime()`	Get current date/time
`FSTime_SetTime()`	Set current date/time

See also 'Changes V4.02-006'.

V4.02-007

Added new µC/Shell commands:

`fs_date`	Output the date & time, or set the system date & time.
`fs_df`	Report disk free space. (Replaces `fs_vol`.)
`fs_umount`	Mount a volume.
`fs_touch`	Change file access and modification times.
`fs_umount`	Unmount a volume.
`fs_wc`	Determine the number of newlines, words and bytes in file.

Version 4.01

V4.01-001

Added time and timestamp conversion functions:

`FSTime_Time_to_Str()`	Convert time to string.
`FSTime_Time_to_TS()`	Convert time to timestamp.
`FSTime_TS_to_Str()`	Convert timestamp to string.
`FSTime_TS_to_Time()`	Convert timestamp to time.
`FSTime_TimeCorrect()`	Make time valid.
`fs_asctime_r()`	Convert time to string.
`fs_ctime_r()`	Convert timestamp to string.
`fs_localtime_r()`	Convert time to timestamp.
`fs_mktime()`	Convert timestamp to time.

Added members to `FS_DATE_TIME` for day of week and day of year (`DayOfWeek` and `DayOfYear`, respectively). See also 'Changes V4.01-001':

V4.01-002

Added functions for entry statistics:

`FSEntry_Query()`	Get information about a file or directory.
`fs_fstat()`	Get information about a file.
`fs_stat()`	Get information about a file or directory.

See also 'Changes V4.01-002'.

V4.01-004

Added journaling for FAT system driver. The interface functions have been added:

`FS_FAT_JournalOpen()`	Open journal.
`FS_FAT_JournalClose()`	Close journal.
`FS_FAT_JournalStart()`	Start journaling.
`FS_FAT_JournalStop()`	Stop journaling.

The configuration has been added:

`FS_FAT_CFG_JOURNAL_EN`	Enable/disable journaling.

Version 4.00

Initial release.

Improvements

Version 4.04.02

None.

Version 4.04.01

V4.04.01-001

MISRA-2004 compliance enhanced (see FS MISRA-C 2004 Compliance Matrix.xls).

V4.04.01-002

Improved the performance of the filesystem layer by optimizing and/or avoiding data copy/clear operations.

V4.04.01-003

Improved performance by enabling cache in the Clk module.

V4.04.01-004

Diminished footprints (used ROM) by merging the similar code from FAT12, FAT16 and FAT32.

V4.04.01-005

Improved compliance to the FAT spec (and compliance check) when formatting a volume.

Version 4.04

V4.04-001

`#define FS_CFG_CONCURRENT_ENTRIES_ACCESS_EN` is added in file `fs_cfg.h` to enable/disable concurrent access to files. When `DEF_DISABLED`, opening the same file more than one time with write permissions is not allowed. This mode is of course safer. When `DEF_ENABLED`, user must manage file access security in the application but the same file can be open multiple times with write permissions.

Version 4.03

V4.03-001

`FSEntry_Copy()` can now copy a file from one volume to another.

V4.03-002

`FSEntry_Rename()` can now rename a file from one volume to another. It CANNOT rename a directory from one volume to another.

V4.03-003

NOR driver now provides BSP function to wait for device ready signal or poll for device readiness:

`FSDev_NOR_BSP_WaitWhileBusy()`	Wait while NOR is busy.

Version 4.02

V4.02-002

IDE/CF driver now configures bus interface for mode-specific timing parameters:

`FSDev_IDE_BSP_GetModesSupported()`	Get supported transfer modes.
`FSDev_IDE_BSP_SetMode()`	Set transfer mode.

Configured timing is based on device capabilities. In addition, DMA read & write is supported now:

`FSDev_IDE_BSP_DMA_Start()`	Setup DMA for command.
`FSDev_IDE_BSP_DMA_End()`	End DMA transfer.

Version 4.01

V4.01-003

Improved directory read functions to place directory information into structure passed by user rather than member of `FS_DIR`:

```
void  FSDir_Rd    (FS_DIR             *p_dir,
                   FS_DIR_ENTRY       *p_dir_entry,
                   FS_ERR             *p_err);

int   fs_readdir_r(FS_DIR             *dirp,
                   struct fs_dirent   *entry,
                   struct fs_dirent  **result);
```

V4.01-006

Added IDE/CF driver port function to obtain the unit's drive number, thereby allowing two devices (one master, one slave) to share a bus:

```
CPU_INT08U  FSDev_IDE_BSP_GetDrvNbr (FS_QTY  unit_nbr);
```

V4.01-008

File access now follows POSIX standard, with and without buffer assigned, for files opened in read/write (update) mode. In general, a read cannot be followed by a write without an intervening call to a file positioning function (`fs_fseek()`, `fs_fsetpos()`, `fs_rewind()`, `FSFile_PosSet()`), unless the previous read encountered the end-of-file; and a write cannot be followed by a read without an intervening call to a file positioning function (`fs_fseek()`, `fs_fsetpos()`, `fs_rewind()`, `FSFile_PosSet()`) or the buffer flush function (`fs_flush()`, `FSFile_BufFlush()`).

Version 4.00

Initial release.

Changes

Version 4.04.02

V4.04.02-001

SD card BSP functions now take an `FS_DEV_SD_CARD_ERR` error pointer instead of a `FS_ERR` pointer.

Version 4.04.01

V4.04.01-001

The master include files were replaced by a more standard include strategy.

V4.04.01-002

Header files are now prevented from multiple inclusion by include guards.

Version 4.04

V4.04-001

Time management previously defined in the `fs_time` module has been replaced by **µC/Clk**. The same functionality is maintained, with flexibility added and easier integration with other Micriµm modules. For example, if you buy **µC/SNTP**, you could easily use the time got from NTP servers to update the access time of the files.

Here is a list of equivalence from `fs_time` to **µC/Clk**.

`FS_DATE_TIME`	replaced by	`CLK_DATE_TIME`
`FSTime_TimeGet()`	replaced by	`Clk_GetDateTime()`
`FSTime_TimeSet()`	replaced by	`Clk_SetDateTime()`
`FSTime_Time_to_Str()`	replaced by	`Clk_DateTimeToStr()`
`FSTime_Time_to_TS()`	replaced by	`Clk_DateTimeToTS_Unix()`
`FSTime_TS_to_Time()`	replaced by	`Clk_TS_UnixToDateTime()`

```
Day of month  is now [1, 31 ] instead of [0, 30 ].
Month of year is now [1, 12 ] instead of [0, 11 ].
Day of week   is now [1, 7  ] instead of [0, 6  ].
Day of year   is now [1, 366] instead of [0, 365].
```

`#define  FS_CFG_GET_TS_FROM_OS` has been removed from file `fs_cfg.h`. Everything regarding time is now managed in `clk_cfg.h`. If you want the time to be managed by **µC/OS-II** or **µC/OS-III**, you must `#define CLK_CFG_EXT_EN` to `DEF_DISABLED` in file `clk_cfg.h` and include file `clk_os.c` in your project.

For more information, see **µC/Clk** user manual.

V4.04-002
Function `FSFile_Rd()` does not return error `FS_ERR_EOF` anymore. Instead, the flag `FlagEOF` of struct type `fs_file` is set to `DEF_YES` to indicate EOF. This way, EOF can be managed differently than an error easier.
V4.04-003
Parameter `dir` of function `FSEntry_Create()` has been replaced. This is the new declaration :

```
void  FSEntry_Create(CPU_CHAR        *name_full,
                     FS_FLAGS         entry_type,
                     CPU_BOOLEAN      excl,
                     FS_ERR          *p_err);
```

Parameter `entry_type` can take the following values:

```
-  FS_ENTRY_TYPE_FILE
-  FS_ENTRY_TYPE_DIR
```

V4.04-004
Parameter `file` of function `FSEntry_Del()` has been replaced. This is the new declaration :

```
void  FSEntry_Del(CPU_CHAR        *name_full,
                  FS_FLAGS         entry_type,
                  FS_ERR          *p_err);
```

Parameter `entry_type` can take the following values:

```
-  FS_ENTRY_TYPE_FILE
-  FS_ENTRY_TYPE_DIR
-  FS_ENTRY_TYPE_ANY
```

V4.04-005
File `fs_bsp.c` and `fs_bsp.h` have been removed from µC/FS.

Version 4.03
V4.03-001
Renamed functions to get open device and volume counts:

`FSDev_GetDevCnt()` Get number of open devices
(renamed from `FSDev_GetNbrDevs()`)

`FSVol_GetVolCnt()`	Get number of open volumes

(renamed from `FSVol_GetNbrVols()`)

See also ‘New Features V4.03-002’.

V4.03-002
Removed `WorkingDirCnt` member of `FS_CFG` structure.

V4.03-003
BSP functions renamed:

`FS_BSP_GetTime()`	Get current date/time.

(renamed from `FS_GetDateTime()`)

`FS_BSP_SetTime()`	Set current date/time.

(renamed from `FS_SetDateTime()`)

V4.03-004
Function `FS_FAT_Chk()` renamed to `FS_FAT_VolChk()`.

V4.03-005
The first partition is now identified with 0 instead of 1. `FSDev_PartitionAdd()` returns FS_INVALID_PARTITION_NBR, if an error occurs, instead of 0 in the previous version.

See also ‘Corrections V4.03-002’.

V4.03-006
Added `flag` argument to `FSEntry_TimeSet()` (renamed from `FSEntry_DateTimeSet()`). This argument indicates which Date/Time should be set.

V4.03-007
Certain bsp functions that are OS dependent were renamed and moved to the µC/FS OS layer (fs_os.c). These functions are listed below,

`FS_BSP_SemCreate()`	renamed to	`FS_OS_SemCreate()`
`FS_BSP_SemDel()`	renamed to	`FS_OS_SemDel()`
`FS_BSP_SemPend()`	renamed to	`FS_OS_SemPend()`
`FS_BSP_SemPost()`	renamed to	`FS_OS_SemPost()`

V4.03-008
Added configuration constant:

```
FS_CFG_GET_TS_FROM_OS
```

This constant allows the application to select if the timestamps are calculated in µC/FS OS layer (fs_os.c) or provided by the application.

Version 4.02

V4.02-002

IDE driver BSP changed (see also 'Improvements V4.02-002').

V4.02-003

SPI BSP functions (SD/MMC SPI, serial NOR flash) must be placed in an appropriately-named `FS_DEV_SPI_API` structure. Under certain conditions, drivers can share SPI APIs (see user manual).

V4.02-004

Removed `FSDev_Fmt()`. If a device must be formatted, a volume should be opened on partition 0 and `FSVol_Fmt()` executed.

Removed `FSDev_FmtLow()`. If a device must be low-level formatted, the driver low-level format function should be used.

V4.02-006

Time set function can be optionally implemented in BSP:

`FS_SetDateTime()`	Set current date/time

This function is called by the API function `FSTime_SetTime()`. If unimplemented, it still must be defined as an empty function. See also 'New Features V4.02-006'.

Version 4.01

V4.01-001

Revised definition of `FS_DATE_TIME`:

`Day`	Day of the month [1..31].
`Year`	Years since 1900.

Renamed `FS_FILE_INFO` to `FS_ENTRY_INFO` (see `FSEntry_Query()` and `FSFile_Query()`); changed its `DateTimeCreate` and `DateTimeWr` members to type `FS_TS`. New function `FSTime_TS_to_Time()` should be used to convert a `FS_TS` to a `FS_DATE_TIME`. See also 'New Features V4.01-001'.

V4.01-002

Removed date/time and attribute get functions:

`FSEntry_AttribGet()`	Get a file or directory's attributes.
`FSEntry_DateTimeGet()`	Get a file or directory's date/time.

`FSEntry_Query()` should be used instead. See also 'New Features V4.01-002'.

V4.01-005

Renamed volume check enable/disable configuration defines:

`FS_FAT_CFG_VOL_CHK_EN` (renamed from `FS_CFG_VOL_CHK_EN`)	Configure volume check support.
`FS_FAT_CFG_VOL_CHK_MAX_LEVELS` (renamed from `FS_CFG_VOL_CHK_MAX_LEVELS`)	Configure max levels checked.

Renamed volume check function:

`FS_FAT_VolChk()` (renamed from `FSVol_Chk()`)	Check file system integrity.

Version 4.00

Initial release.

Corrections

Version 4.04.02

V4.04.02-001
In function `fs_stat`, the assignment to the fields `st_ctime` and `st_mtime` of the structure `p_info` was interchanged.

V4.04.02-002
`FS_VERSION` #define in fs.h now has the correct value (was 40500 in V4.04.01).

V4.04.02-003
`FS_CFG_BUILD` default value in fs_cfg.h template corrected to `FS_BUILD_FULL` instead of `FS_BUILD_DEV_ONLY`.

Version 4.04.01

V4.04.01-001
Corrected the mechanism to verify if a file is already open.

V4.04.01-002
Revised cluster allocation strategy to allocate a full chain.

V4.04.01-003
Added contextual reverse chain deletion to improve robustness

V4.04.01-004
Corrected the journaling module to avoid lost cluster chains.

V4.04.01-005
Corrected a bug in the cache module that could cause a recursion leading to data corruption when in write-back mode.

Version 4.04

V4.04-001
Fixed incorrect behavior when deleting/renaming an open file.

V4.04-002
Fixed incorrect determination of FAT type when formatting.

V4.04-003
Fixed incorrect determination of FAT type when mounting existing FAT filesystem.

V4.04-004
Fixed `FS_FAT_VolChk()` when `FS_CFG_UTF8_EN` is enabled.

Version 4.03

V4.03-001

Fixed erroneous calculation of position in a file opened for read or write when either `fs_fseek()` or `FSFile_PosSet()` is used.

V4.03-002

Creating several partitions on a drive was not handled correctly, this issue was corrected. In this release, up to four partitions can be created.

See also ‘Changes V4.03-005’.

V4.03-003

The Number of Hidden Sectors in partition (offset 0x1C of partition BPB) is now set correctly to the partition start sector.

V4.03-004

In certain functions in the previous version, the pointer to the error code `p_err` returned by a call to a particular function was overwritten by a call to another function causing an unexpected behavior of µC/FS. This issue is now fixed.

V4.03-005

Corrected erroneous device name usage. Now, an attempt to open a device twice returns the error `FS_ERR_DEV_ALREADY_OPEN`.

V4.03-006

In the previous version, an attempt to open a file in append mode was erroneously not allowed. The issue is now corrected.

V4.03-007

Corrected erroneous return value FS_ERR_EOF early during buffered read.

V4.03-008

Corrected erroneous file position calculation when truncating a new created file to a given size.

V4.03-009

Corrected erroneous return value from the call to `fs_readdir_r()`. Now, the function returns 0 if no error occurs, otherwise it returns 1.

V4.03-010

Corrected erroneous return value from the call to `fs_fflush()`. Now, the function returns 0 if no error occurs, otherwise it returns `FS_ERR_EOF`.

V4.03-011

Corrected erroneous devices count setting during the initialization of µC/FS modules.

V4.03-012
Previously, in certain conditions an attempt to open a volume fails. This issue was corrected by clearing a temporary buffer used to hold the volume name in `FS_PathParse()` function.

V4.03-013
Corrected erroneous return value from certain functions in the template file fs_app.c

V4.03-014
Previously the return value of `FSVol_Lock()` function was not correctly handled when this function is called in other internal functions. This issue is now corrected.

V4.03-015
Previously for certain SD card models, the initialization failed. This issue was corrected by changing the number of retries before reporting a timeout error. The whole amount of time doesn't exceed the specified initialization delay.

V4.03-016
Corrected the consistency in the returned error codes when a drive (e.g. CF) is removed during certain µC/FS operations (e.g. open, read and write to a file).

V4.03-017
Previously, in certain conditions formatting a volume returns the following error,
`FS_ERR_PARTITION_INVALID_SIZE`
This issue is now corrected.

V4.03-018
The time functions could return impossible dates. The issue is now resolved.

Version 4.02
V4.02-005
Directory creation previously failed if user-specified name ended with path separator character (`'\'`). Directory names are now allowed to end with path separator character.

Version 4.01
V4.01-007
Corrected erroneous generation of tail for LFNs that are valid SFNs.

V4.01-008
File EOF indicator NOT set upon read when previous access read until (but not past) the EOF. See also 'Improvements V4.01-008'.

Version 4.00

Initial release.

Known Problems

Version 4.04.02

V4.03-001 (Unresolved)

V4.00-002 (Unresolved)

Version 4.04.01

V4.03-001 (Unresolved)

V4.00-002 (Unresolved)

Version 4.04

V4.03-001 (Unresolved)

V4.00-002 (Unresolved)

Version 4.03

V4.03-001
Access date is not updated when a file is accessed in read mode.

V4.03-002
Attempting to close a volume or a device when it still referenced is not correctly handled. An error should be returned.

V4.00-001 (Unresolved)
V4.00-002 (Unresolved)

Version 4.02

V4.00-001 (Unresolved)
V4.00-002 (Unresolved)

Version 4.01

V4.00-001 (Unresolved)
V4.00-002 (Unresolved)

Version 4.00

V4.00-001
Files and directories not protected from deletion/rename/etc. while open.

V4.00-002

Rename does NOT check for condition "`p_name_full_new` contains a path prefix that names `p_name_full_old`".

Limitations

001

Following features NOT supported:

(a) Logical device driver
(b) Extended partitions
(c) NAND device driver

Contacts

Micriμm
949 Crestview Circle
Weston, FL 33327
USA
+1 954 217 2036
+1 954 217 2037 (FAX)
e-mail: Licensing@Micrium.com
WEB: www.Micrium.com