

Release Notes for Microchip Memory Disk Drive File System

Version 1.01
February 4, 2008

1. Description

This library is intended to provide an interface to file systems compatible with ISO/IEC specification 9293 (commonly referred to as FAT12 and FAT16). This library includes four different physical interface files: one for SecureDigital card interface using the SPI module, one for CompactFlash card interface using manual bit toggling, one for CompactFlash card interface using the Parallel Master Port module included on several 16-bit PIC microcontrollers, and one template interface file that can be modified by the user to create a custom interface layer to an unsupported device..

2. Changes In This Release

- a. FindFirst and FindNext will now return the create time/data in the timestamp field of a SearchRec object when they return values for a directory.
- b. Corrects a bug in the FindEmptyCluster function when searching for files beyond the end of a storage device.
- c. Automatically aligns buffers for 16-bit architectures.
- d. For the SPI interface, prescaler divides will now be determined dynamically based on the system clock speed defined in FSconfig.h.
- e. The DiskMount, LoadMBR, LoadBootSector, and FSFormat functions, as well as the gDiskData, gFATBuffer, and gDataBuffer structures are now located in FSIO.c instead of in the interface files.
- f. The SectorRead function will now do a dummy read of the sector and discard the data if it is called with NULL as the data pointer.
- g. Replaced the device initialization code in the FSFormat function with calls to InitIO and MediaInitialize.
- h. The MediaDetect function is not de-bounced. In order to determine that a device is available, you must call MediaDetect, wait for an appropriate amount of time, and then call it again.
- i. The sample linker script in the MDD File System-PIC18-CF-DynMem-UserDefClock project has been modified. Previously, several databanks were merged together; this caused an issue accessing variables that spanned multiple data banks. C18 only allows users to access variables like these using pointers.
- j. Added a new user function. The FSrename function will allow the user to rename files and directories. A version that accepts a ROM filename is available for PIC18 (FSrenamepgm). The API is:

```
Function:      int FSrename (const char *fileName, FSFILE * fo)
PreCondition: None
Input:        fileName    - The new name of the file
```

```

Output:      fo          - The file to rename
Side Effects: int          - Returns 0 if success, -1 otherwise
Overview:   None
Note:       Change the name of a file or directory
            This function will change the name of the current
            working directory if 'fo' equals NULL.

```

3. Known Issues

a. This implementation does not support long file names. When using the FSremove or FSremovepgm functions on a file with long file names, the file's FAT entries and short name directory entry will be deleted successfully, but any long file name entries will not be removed.

4. Compiler Version Used

This library was compiled using MPLAB C18 v.3.10 and MPLAB C30 v.3.00 C compilers.

5. Memory Size

Unoptimized memory usage for the file interface library using the SD-SPI physical layer is given in Table 1. 512 bytes of data memory are used for the data buffer, and an additional 512 are used for the file allocation table buffer. Additional data memory will be needed based on the number of files opened by the user at once. The default data memory values provided include space for three files opened in static allocation mode. The C18 data memory value includes a 512 byte stack. The first row of the table indicates the smallest amount of memory that the library will use (for read-only mode), and each subsequent row indicates the increase in memory caused by enabling other functionality. Optimized and unoptimized totals for program and data memory with all functions enabled are listed after the table.

Table 1: Memory Usage (Unoptimized)

Functions Included	Program Memory (C30)	Data Memory (C30)	Program Memory (C18)	Data Memory (C18)
All extra functions disabled (read only mode)	11313 bytes	1224 bytes	19337 bytes	1775 bytes
File Search enabled	+1824 bytes	+0 bytes	+4040 bytes	+0 bytes
Write enabled	+6513 bytes	+0 bytes	+12262 bytes	+0 bytes
Format enabled (Write must be enabled)	+2448 bytes	+0 bytes	+4732 bytes	+0 bytes
Directories enabled (Write must be enabled)	+7905 bytes	+70 bytes	+15006 bytes	+79 bytes

Functions Included	Program Memory (C30)	Data Memory (C30)	Program Memory (C18)	Data Memory (C18)
FSprintf enabled	+4749 bytes	+0 bytes	+5419 bytes	+0 bytes
File Search and Directories enabled	+42 bytes	+0 bytes	+232 bytes	+0 bytes
Pgm functions enabled	N/A	N/A	+1704 bytes	+0 bytes

Total memory usage

C18:

Unoptimized Program memory- 62732 bytes

Unoptimized Data memory- 1854 bytes

Optimized Program memory- 35490 bytes

Optimized Data Memory- 1854 bytes

C30:

Unoptimized Program memory- 34791 bytes

Unoptimized Data memory- 1294 bytes

Optimized Program memory- 21180 bytes

Optimized Data memory- 1294 bytes

6. More Information

More detailed information about the operation of this library is available in Application Note 1045, available from www.microchip.com.