

# 一般論文

## 組込みファイルシステム USFilesPlus exFAT の開発

Development of embedded file system USFilesPlus exFAT

谷 端 学\*  
M. Tanihata

### 概 要

“USFilesPlus exFAT”とは、当社取扱製品である組込みファイルシステム“USFilesPlus”に、(Microsoft社が開発した)フラッシュドライブ向けファイルシステム“exFAT”の対応を追加した製品である。本稿では、FATとexFATの仕様について記載する。そして、最後にUSFilesPlus exFATの特徴、構成について記述する。

### Synopsis

USFilesPlus exFAT is a product that added exFAT filesystem for the flash drive developed by Microsoft Corporation to USFilesPlus, our embedded filesystem. This paper lists the specification of FAT and exFAT and finally describes the feature and composition of USFilesPlus exFAT.

### 1. はじめに

近年の技術の進歩により、ハードディスクやSDメモリカードをはじめとする記憶媒体の大容量化が進んでいる。それに伴い、デジタルムービーカメラなどの動画データや長期間収集するログデータなどの大容量のデータを扱うことが必須となり、家電製品、デジタル機器、産業機器などでの組込みファイルシステムとしての対応が急務となってきた。

記憶媒体のメディアとしてSDメモリカードを例にあげると、SDカード、SDHCカード、SDXCカードと容量により名称が異なった媒体が市場に出回っている。これらは記憶方式の仕組みに違いがある。

この仕組みとして、フロッピーディスクやハードディスクの中に記憶されるデータの管理を行うためのファイルシステムFile Allocation Tables (以下、FATと記述する)が使用されている。このFATから、記憶媒体の大容量化に伴い、FATからFile Allocation Table 32 (以下、FAT32と記述する)、Extended File Allocation Table (以下、exFATと記述する)と拡張されてきた。ただし、この拡張過程において、対応できていないOSなどもある。(表1 SDカ

ードの種類、表2 ファイルシステムの対応形式 参照)

表1 SDカードの種類

SDメモリカード	形式	容量
SDカード	FAT	4GB未満
SDHCカード	FAT32	4GB以上、32GB以下
SDXCカード	exFAT	32GB超

表2 ファイルシステムの対応形式

OS	標準対応		
	FAT	FAT32	exFAT
Windows 9x	○	○	×
Windows Me	○	○	×
Windows NT	○	×	×
Windows 2000	○	○	×
Windows XP	○	○	△*
Windows Server 2003	○	○	×
Windows Vista (SP1)	○	○	○
Windows 7	○	○	○

\*exFAT対応の更新プログラムを使用した場合

\* (株)日新システムズ

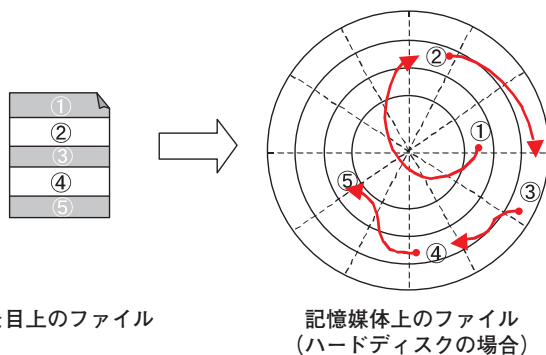
これらは、使用するPCや組込み機器などのファイルシステムが、記憶方式の仕組みであるFAT、FAT32、exFATに対応しているかどうかの影響している。

次章では、FATとexFATとの差異及び、機能制限等について説明し、当社日新システムズの製品であるUSFilesPlus exFATのファイルシステムについて説明する。対象デバイスは、これまでの例で記述したSDメモリーカードとして記載する。

また、FATに関しては、FAT12、FAT16、FAT32とあるが、現在あまり使用されていないFAT12の説明については省略し、FAT16をFATとして記載する。

## 2. ファイルシステムとは

たとえば、ファイルを作成した場合、記憶媒体上にはそのままの形で登録されるわけではなく、セクタと呼ばれる固定長の記憶領域に分割されて登録されている。(図1ファイルのイメージ参照)



見た目上のファイル

記憶媒体上のファイル  
(ハードディスクの場合)

図1 ファイルのイメージ

基本的には、記憶媒体上のファイルはランダムに記憶されるのではなく、記憶媒体の形式に沿った形で登録される。この形式がFATなどである。そして、その形式に合わせてバラバラに登録されたファイルを連続した1つのファイルとして見せ、記憶媒体にファイルやフォルダ(ディレクトリ)を作成、移動、削除、ファイル名の変更などにより、データを管理する仕組みがファイルシステムである。

また、これらの形式には、FAT、exFATなどがある。尚、次章以降で詳細は説明するが、FATのみ対応したファイルシステムがexFATの記憶媒体(SDXCカードなど)を認識することはできない。また、その逆も然りである。

これは、記憶媒体の情報から、FATやexFATを識別し、対応する形式を判断した後、それぞれの形式に合わせた処理を行っているからである。また、FATとexFATに関しては、この情報を格納する構造自体が異なる。

たとえば、exFATでフォーマットされたSDXCカードをWindowsXPのPCでアクセスした場合、ファイルを開くことができない。(図2未対応形式の表示参照)

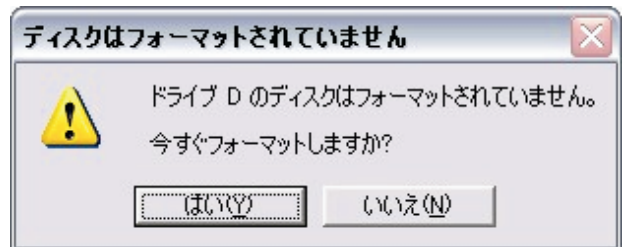


図2 未対応形式の表示

これは、WindowsXPのファイルシステムがこの判別を行う情報を確認した際に、対応していない形式のSDカードを認識したために表示されている。そのため、ファイルシステム自体は対応した形式の記憶媒体に対してのみ有効であると言える。尚、USFilesPlus exFATでは、exFATだけではなく、FATにも対応しているため、exFATかFATかの判別を行った後、それぞれの対応形式に合わせたファイルシステムで処理を行う。

## 3. FATとその問題について

FATに関して説明する前に、基本となる「セクタ(sector)」、「クラスタ(cluster)」について説明する。

セクタとは、データを読み書きする単位である。対して、クラスタとは、ディスク装置の記憶領域の単位である。これらの関係はセクタの集合体がクラスタということである。(図3セクタとクラスタの関係参照)

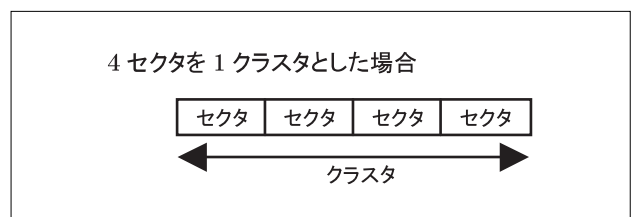


図3 クラスタとセクタの関係

次に、FATにはFAT12、FAT16、FAT32などの種類がある。これらは、FATが管理するクラスタの単位が12ビット、16ビット、32ビットと異なる。FATのエントリは、メモリ上のクラスタと1対1で対応し、該当するクラスタの使用状況を表す。そのため、FATのテーブルにはクラスタが鎖の様に繋がっている。(図4クラスタチェーン参照)

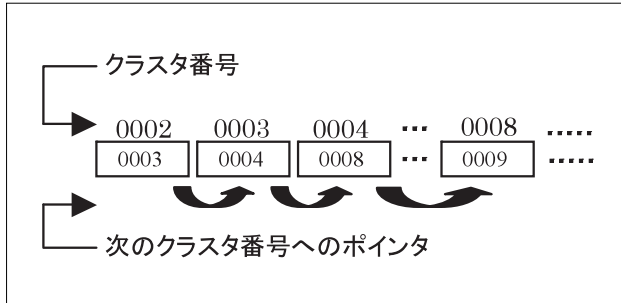


図4 クラスタチェーン

これらは使用する記憶媒体の容量にあわせて拡張するために、FAT12、FAT16、FAT32と大きくなっていったものである。たとえば、現在では、FAT12はフロッピーディスクなどで使用され、FAT16は4GB未満のSDメモリーカード、FAT32は、4GB以上32GB以下のSDメモリーカードなどで使用されている。構成自体は基本的にFAT12とFAT16はほぼ同じである。

しかしながら、FAT32に関しては、FAT12/16とは若干異なる点がある。

たとえば、SDメモリーカードのバイナリデータをダンプして確認すると、Master Boot Record (以下、MBRとする)、BIOS Parameter Block (以下、BPBと記述する)、FATなどの領域が確認できる。(図5 FAT12/16とFAT32の比較、表3 FAT構成 参照)

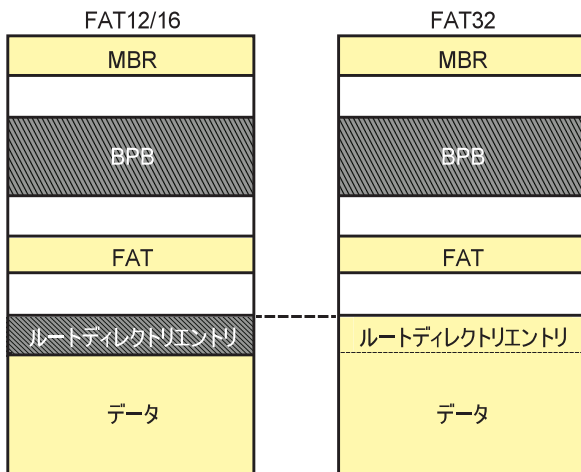


図5 FAT12/16とFAT32の比較

表3 FAT構成

MBR	パーティションの関係位置、サイズ等を記録
BPB	パーティション情報を記録
ルートディレクトリエントリ	ルートディレクトリのファイルエントリを格納

MBRは、共通の構造であり、FATは先述の管理クラスタの単位に違いはあるが、基本的には同じ構造である。しかしながら、ルートディレクトリエントリに関しては異なる。

FAT12/16では、ルートディレクトリエントリ情報の数が512個(固定値)としてBPBの領域で定義され、そして、FATの次のセクタからルートディレクトリエントリの領域として記録される。また、ルートディレクトリ以外のファイルやディレクトリに関してはデータ領域に記載される。一方、FAT32の場合は、ルートディレクトリエントリとしての領域自体は存在せず、ファイルやディレクトリ同様にデータ領域に記載される。そのため、ディレクトリのエントリ自体は、クラスタチェーンで構成されることになる。そのため、ファイル数の制限に違いが生じる。(表4 FATの制限 参照)

表4 FATの制限

内容	FAT12	FAT16	FAT32
1ファイルのサイズ制限	32MB	2GB	4GB-1Byte
ディレクトリ当りのファイル数	512まで		65535まで
ディスクサイズ	32MB	2GB	2TB

※ファイル名が8.3形式の場合に限る

それは、1ファイルのサイズの上限が4GBということである。(正確には、4GBから1Byte引いた値)これはディレクトリエントリのファイルサイズ(バイト単位)情報を格納している領域のサイズが32ビットであるため、4GB(2の32乗)の値になるからである。

そのため、以下の問題があるFATに替わる形式として、exFATが開発された。

- ◆ 対応可能な記憶媒体のサイズに制限がある
- ◆ 1ファイル当たりのサイズに制限がある

最後に、FAT16での各情報により、記憶媒体内のデータに行き着くまでのシーケンスについて説明する。

たとえば、SDカードに「TEST.TXT」のデータファイルが存在しているとする。

まず、FATの開始セクタを求める。なぜなら、FATは1つのファイルが複数クラスタにまたがっている場合、続きがどのクラスタにあるか、という情報を保持しているからである。(図6 FAT領域 参照)



図6 FAT領域

尚、FATの開始セクタは、BPBのセクタの次のセクタに存在するため、BPBのセクタに+1を行った値がFATのセクタになる。

次に、ファイル名やデータの先頭位置(開始クラスタ)などの情報が格納されているルートディレクトリエントリの開始セクタを求める。ルートディレクトリエントリの開始セクタは、FATの次のセクタに位置するため、BPBの「FAT1つあたりのセクタ数」と「FATの数」から算出する事ができる。(図7 BPB領域 参照)

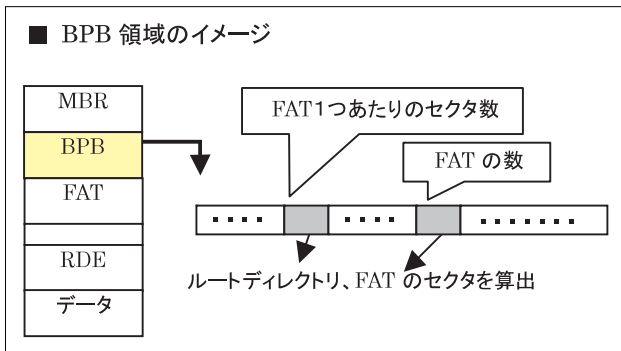


図7 BPB領域

このルートディレクトリエントリ情報の「開始クラスタ番号」から、データが格納されているセクタが確認できる。(図8 データまでのシーケンス 参照)

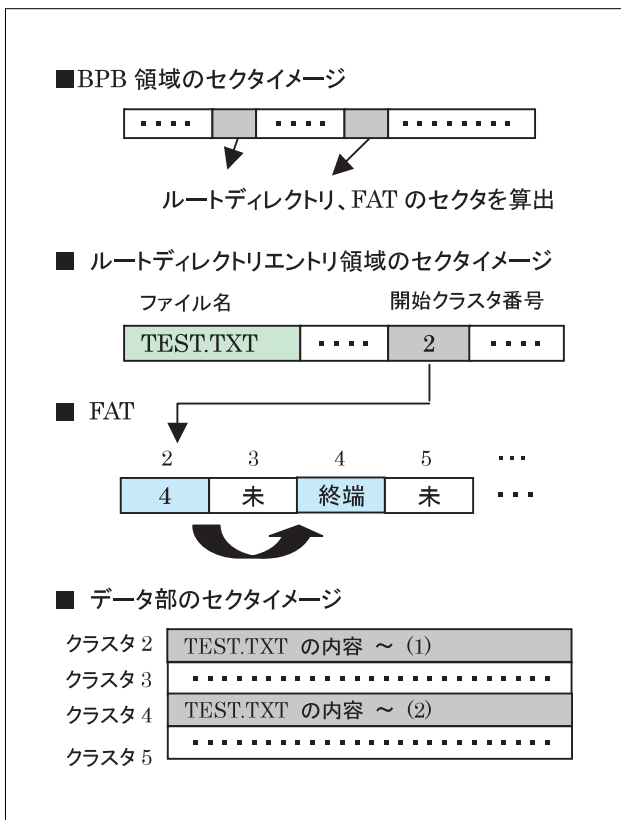


図8 データまでのシーケンス

#### 4. exFATについて

exFATのデータ構成は、FATと比べると、BPB、FATの領域に格納している情報の構成が異なる。また、ディレクトリエントリに関してはFATとは全く構成が異なる。加えて、新たな情報の領域も確認することができる。(図9 SDXCカードのダンプデータ 参照)



図9 SDXCカードのダンプデータ

exFATから、大容量メディアへの対応及び、それに伴う機能が実装されている。

まず、大容量の記憶媒体への対応に関しては、FATのディレクトリエントリの構造を全面的に改定することで、大容量のデータの扱いが可能となっている。(表5 exFATの制限 参照)

表5 exFATの制限

内容	制限
1ファイルのサイズ制限	16EBまで
ディレクトリ当たりのファイル数	2796202まで
ディスク最大サイズ	16EBまで

また、大容量の記憶媒体に対応することで、読み書きの高速化及び、信頼性が必要になる。そのため、以下の情報を持つ仕組みが新たに追加されている。

- ① アロケーションビットマップ
- ② 大文字変換テーブル情報

①は、クラスタの使用状況を示したビットマップである。このアロケーションビットマップの格納されているアドレスとサイズは、ディレクトリエントリの情報に格納されている。尚、内容としては、そのビットマップの1ビットが1クラスタに対応しており、ビットが立ってい

れば、そのクラスタは使用中であることを意味する。

また、FAT 上での使用状況とクラスタビットマップ上での使用状況が一致していない場合は、アロケーションビットマップの方を優先する仕様となっている。

次に、②の大文字変換テーブル情報に関しては、ファイル名のハッシュ値を求めるために、ユニコードのファイル名を大文字に変換する際に使用する。このテーブルが格納されているアドレス及び、サイズについてもディレクトリエントリの情報に格納されている。

exFATでは、ファイル名のハッシュ値を用いることで、大量のファイルの検索における効率化が可能となった。なぜなら、これまでのFATでは、指定されたファイル名と、登録されているファイル名の文字列同士の比較を行っていた。しかし、exFATでは、ハッシュ値を用いることでLONG型の数値（ハッシュ値）で比較を行うようになったからである。ただし、ハッシュ値自体が重複してしまう可能性があるため、ハッシュ値が同じであるファイルが検出された場合は、ファイル名での比較も必要である。

最後に、exFATでの各情報により、記憶媒体内のデータに行き着くまでのシーケンスについて説明する。

まず、FAT同様に、BPBの情報から解析を行う。FATの開始セクタは、BPBの情報にて格納されている。

次に、ルートディレクトリエントリの開始セクタに関しては、BPBの「ルートディレクトリエントリのクラスタ数」及び、「クラスタサイズシフト数」などの値から算出する。(図9 exFATのBPB 参照)

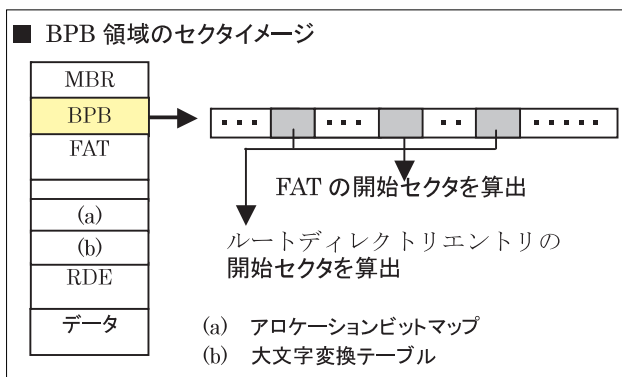


図9 exFATのBPB

算出方法に関しては、以下の方法にて行う。

**指定クラスタからセクタ番号の算出方法**

指定クラスタのNのセクタ番号＝  

$$\text{隠しセクタ数} + \text{データ開始セクタ番号} + (\text{ルートディレクトリエントリの開始セクタ} - 2) \ll \text{クラスタサイズシフト数}$$

上記の算出結果から、ディレクトリエントリの内容を確認する。exFATのディレクトリエントリは、FATと構成が異なり、各エントリの先頭にタイプを現す設定値がある。(表6 タイプ一覧 参照)

表6 タイプ一覧

値	説明
0x00	ディレクトリエントリの終了
0x81	アロケーションビットマップの情報
0x82	大文字変換テーブル情報
0x83	ボリュームラベル
0x85	ファイル情報
0xC0	拡張情報
0xC1	名前

尚、0x81、0x82、0x83については、ルートディレクトリで使用される。また、0x83は、ディスクのボリュームラベルの情報になる。たとえば、SDXCカードの場合、フォーマットを行う際に設定することができる。もし、設定していない場合は0x83のエントリの記載はない。0x81、0x82に関しては、アロケーションビットマップ及び、大文字変換テーブルの情報に格納されている。(図10 exFATのディレクトリエントリ (1) 参照)

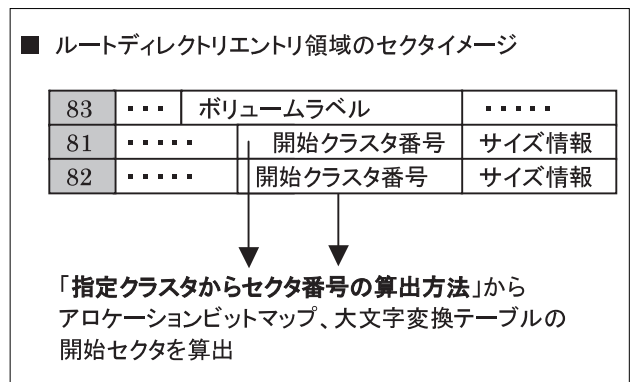


図10 exFATのディレクトリエントリ (1)

次にファイルデータに関しては、0x85、0xC0、0xC1を1組(1ファイル、1ディレクトリ)として登録される。

その後、ファイルのデータは、0xC0内に設定されている「ファイルデータが格納されているクラスタ番号」から、セクタ番号を算出することでデータ領域の確認ができる。

また、ディレクトリの中にファイルが存在する場合は、ファイル情報の0x85内のファイル属性の情報にて「ディレクトリ」と設定され、尚且つ、0xC0の「ファイルデータが格納されているクラスタ番号」に、サブディレク

トリのエントリが記載されているクラスタ番号登録されている。尚、これらのクラスタ番号から開始セクタ番号を算出するには、前述の「指定クラスタからセクタ番号の算出方法」を用いて算出する。(図11 exFATのディレクトリエントリ (2) 参照)

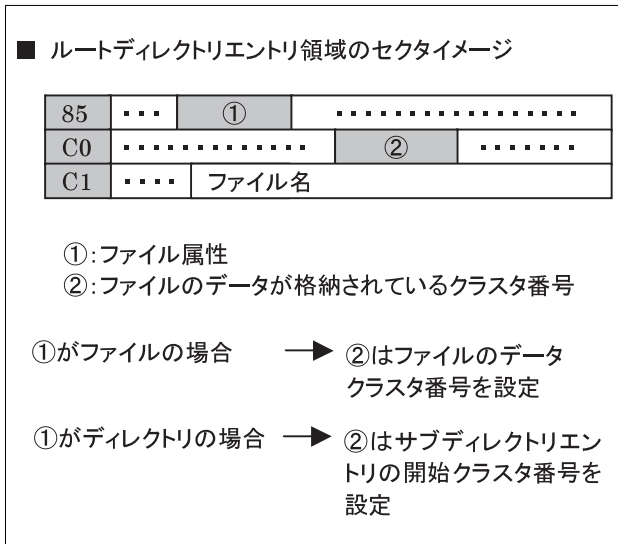


図11 exFATのディレクトリエントリ (2)

## 5. USFilesPlus exFAT について

当社では、FAT及び、exFATに対応した組込みファイルシステムUSFilesPlusを提供している。

このUSFilesPlus は、豊富な実績を持つ米国 LANTRONIX 社 (旧 US Software 社) のMS-DOS、Windows互換FATファイルシステム USFiles を日新システムズがリニューアルさせた製品である。特徴としては、以下のものがあげられる。

- FAT12/FAT16/FAT32/VFAT/exFATに対応 (※)
- シンプルな構成
- 組込み向けのコンパクトなコードサイズ
- ファイルシステムユーティリティを標準添付  
※パッケージにより、扱う事ができるFAT形式が異なります。

USFilesPlusu には以下のパッケージがある。(図12 パッケージ構成 参照)

- FAT12/16 パッケージ  
2GB未満の記憶媒体にしか対応できないが、その分、最小限のリソースでファイルシステムを使用することができる。
- FAT12/16/VFAT パッケージ  
2GB未満の記憶媒体にしか対応できないが、ロングファイルネームを扱うことができる。

- FAT32 パッケージ  
2GB以上の記憶媒体に対応し、尚且つ、ロングファイルネームを扱うことができる。  
尚、本パッケージにより、FAT12/16も扱うことが可能なフルパッケージになる。
- USFilesPlus exFAT (オプション製品)  
上記のパッケージに、本オプション製品を使用することで、FAT及び、exFATに対応することができる。

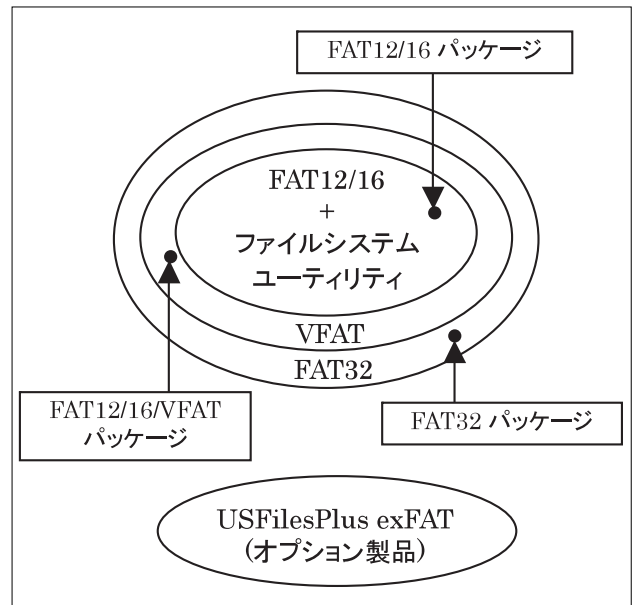


図12 パッケージ構成

構成に関しては、「Stream I/O」、「Manager」、「Driver/Device」の大きく3つの層の構成である。(図13 USFilesPlus構成図 参照)

まず、第1層は、ファイルの操作及び管理を行うためのAPIが位置している。これらのAPIはANSI C のファイルI/Oインターフェイスを意識して作成しているため、引数等の指定方法は同様である。違いとして、ANSI C のファイルI/OインターフェイスのAPI名に「mt\_」が付加されている点が上げられる。(例. fopen () ⇒ mt\_fopen ())

次に、第2層は、ファイルシステムの核となる部分であり、FAT12/16/FAT32/VFAT/exFAT等の解析や処理はここで行われる。先述のFATやexFATの解析や管理はこの部分にて実装されている。

最後に、第3層は、使用する環境に依存する部分になり、プラットフォーム毎に作成する必要がある。一般的に、この部分はサポート対象外に位置付けれるが、USFilesPlus では、サンプルとして幾つかのドライバを提供しているため、これらを参考に作成を行うことで、比較的容易にドライバを作成できる。

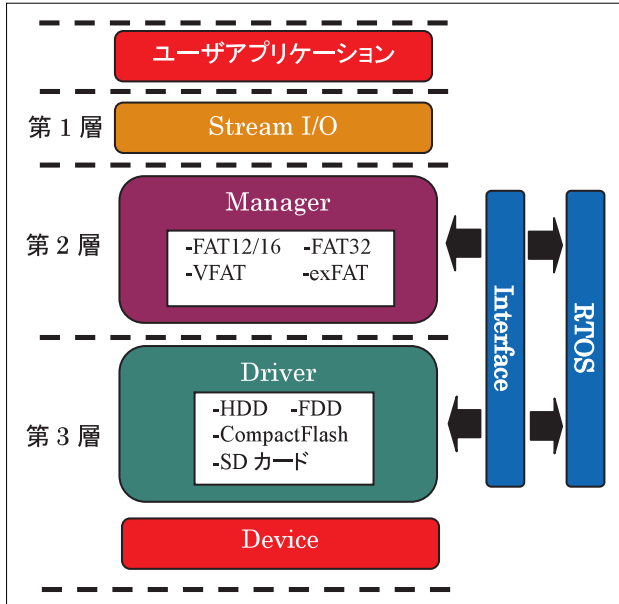


図13 USFilesPlus構成図

以上のことから、USFilesPlus exFAT は、exFATのみの対応だけではなく、構成上、パッケージ内容にもよるが、基本的にはFAT12/16または、FAT32に対応している。

最後に、以下の環境でのパフォーマンス及び、コードサイズを記載する。

本パフォーマンス測定方法は、1MBのファイルをSDカードに書き込みを行った際の計測データである。また、その環境のコードサイズを計測した結果も記載する。

パーソナルメディア株式会社製 RTEV850E/MA3

CPU	: V850E/MA3 (CPU 80MHz, BUS 40MHz)
コンパイラ	: GNU
OS	: $\mu$ ITRON4.0対応RTOS
デバイス	: SDカード SPIモード

## パフォーマンス

	exFAT	FAT32
書き込み	262.563 (262.563)	270.612 (264.804)
	270.184 (269.189)	369.141 (369.141)
読み込み	285.713 (285.953)	274.309 (278.866)
	285.633 (285.474)	368.743 (369.541)

(単位: KB/Sec)

上段: ルートディレクトリ (ロングネーム)

下段: サブディレクトリ (ロングネーム)

## コードサイズ

パッケージ	ROM	RAM
FAT12/16	23	7
FAT32	29	7
FAT32+exFAT	39	7

(単位: KB)

ユーティリティ及び、デバイスドライバのコードサイズは含んでいません。

## 6. あとがき

本稿では、exFAT ファイルシステムの仕様および組込みファイルシステム USFilesPlus exFATについて記述した。exFAT に対するニーズは、組込み機器が搭載するフラッシュドライブの大型化に伴い、今後も更に高まっていくものと考えられる。小型の組込みファイルシステムである USFilesPlus exFAT の採用により、お客様のプロジェクト開発に貢献できるよう、今後ともソフトウェアの開発・支援に積極的に取り組んでいく所存である。

(注) Windows、MS-DOSは、米国Microsoft Corporationの米国およびその他の国における商標または登録商標です。

## 執筆者紹介



谷端 学 Manabu Tanihata

(株)日新システムズ

エンベデッド・ソリューション・

ビジネスユニット プロダクト技術部