

オープンソースFATファイルシステム M3S-TFAT-Tiny

ユーザーズマニュアル
ルネサスマイクロコンピュータ ミドルウェア

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、
 家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
 防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

はじめに

本書は、ルネサスマイクロコンピュータ用のオープンソースFATファイルシステム M3S-TFAT-Tiny（以下TFATライブラリとします）について説明します。

TFATライブラリはFatFsをベースに作られました。

各ソースバージョンの関係を以下に示します。

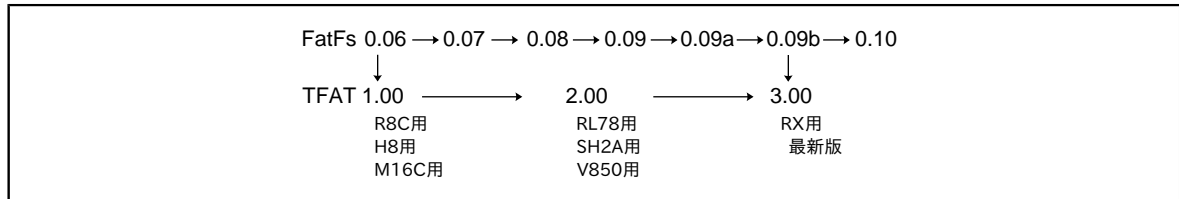


図1.TFATライブラリとFatFsのバージョンの関係

FatFsとは？

FatFsは小規模な組み込みシステム向けの汎用FATファイルシステム・モジュールです。

FatFsはChaN氏によって開発されたソフトウェアです。

ChaN氏はFatFsを含め、インターネット上で無償で組み込み向けソフトウェアを提供されています。

ChaN氏及び、FatFsの詳細に関しては以下URLを参照してください。

http://elm-chan.org/fsw/ff/00index_j.html

ロングファイル名(以下LFN)に関する注意事項：

マイクロソフト社はFATファイル・システムについていくつかの特許を保有しています。

いずれもLFN機能の実装に関するもので、その利用に対してライセンス料を要求しています。

TFATライブラリ及びFatFsではLFN機能を構成オプションで任意にON/OFFできるため、OFFでご使用ください。

すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

はじめに	1
1. ライブラリ仕様	1-1
1.1. TFATライブラリの仕様	1-1
1.2. 構成	1-1
1.3. TFATライブラリの機能	1-2
1.4. TFATライブラリのconfig設定	1-2
1.5. TFATとFatFsのAPI対応表	1-3
2. ライブラリの型の定義	2-1
3. ライブラリ構造体	3-1
3.1. FATFS - ファイルシステムオブジェクト構造体	3-1
3.2. DIR - ディレクトリオブジェクト構造体	3-1
3.3. FIL - ファイルオブジェクト構造体	3-2
3.4. FILINFO - ファイルステータス構造体	3-2
4. ライブラリ定数	4-1
4.1. FRESULT - ライブラリ関数戻り値	4-1
4.2. ファイル属性情報	4-1
4.3. ディスクステータスのためのマクロ	4-2
4.4. Enum - DRESULT	4-2
5. ライブラリ関数	5-1
R_tfat_f_mount	5-2
R_tfat_f_open	5-3
R_tfat_f_close	5-5
R_tfat_f_read	5-6
R_tfat_f_write	5-7
R_tfat_f_lseek	5-8
R_tfat_f_truncate	5-9
R_tfat_f_sync	5-10
R_tfat_f_opendir	5-11
R_tfat_f_readdir	5-12
R_tfat_f_getfree	5-13
R_tfat_f_stat	5-14
R_tfat_f_mkdir	5-15
R_tfat_f_unlink	5-16
R_tfat_f_chmod	5-17
R_tfat_f_utime	5-18
R_tfat_f_rename	5-19
R_tfat_f_forward	5-20
6. メモリドライバインタフェース関数	6-1
R_tfat_disk_initialize	6-2
R_tfat_disk_read	6-3
R_tfat_disk_write	6-4
R_tfat_disk_ioctl	6-5
R_tfat_disk_status	6-6
R_tfat_get_fattime	6-7
R_tfat_outstream	6-8

図目次

1. TFATライブラリとFatFsのバージョンの関係	1
1.1. TFATライブラリの構成	1-2
5.1. ライブラリ関数詳細の見方	5-1

表目次

1.1. TFATライブラリの仕様	1-1
1.2. 機能一覧	1-2
1.3. config設定一覧	1-3
1.4. TFATとFatFsのAPI対応表	1-3
2.1. TFATライブラリの型	2-1
3.1. FATFSのメンバ	3-1
3.2. DIRのメンバ	3-1
3.3. FILのメンバ	3-2
3.4. FILINFOのメンバ	3-2
4.1. FRESULTの値	4-1
4.2. ファイル属性情報のマクロ	4-2
4.3. ディスクステータスのためのマクロ	4-2
4.4. DRESULTの値	4-2

1. ライブラリ仕様

1.1. TFATライブラリの仕様

TFATライブラリの主な仕様を以下に示します。

表1.1.TFATライブラリの仕様

項目	仕様
オープンソースベース	FatFs R0.09b
FATタイプ	FAT16、FAT32
ファイル名	8.3形式（8文字のファイル名および3文字の拡張子）
ファイルシステムのフォーマット機能	なし
ドライブ数	最大 10
セクタサイズ	512byte
ROMサイズ(※1)	約10Kbyte
ドライブのマウントに必要なワーク領域(※1)	約500byte
1ファイルのアクセスに必要なワーク領域(※1)	約50byte
必要スタックサイズ(※1)	約300byte

(※1)詳細な数値はFITモジュールアプリケーションノートをご参照ください。

1.2. 構成

TFATライブラリの構成を以下に示します。

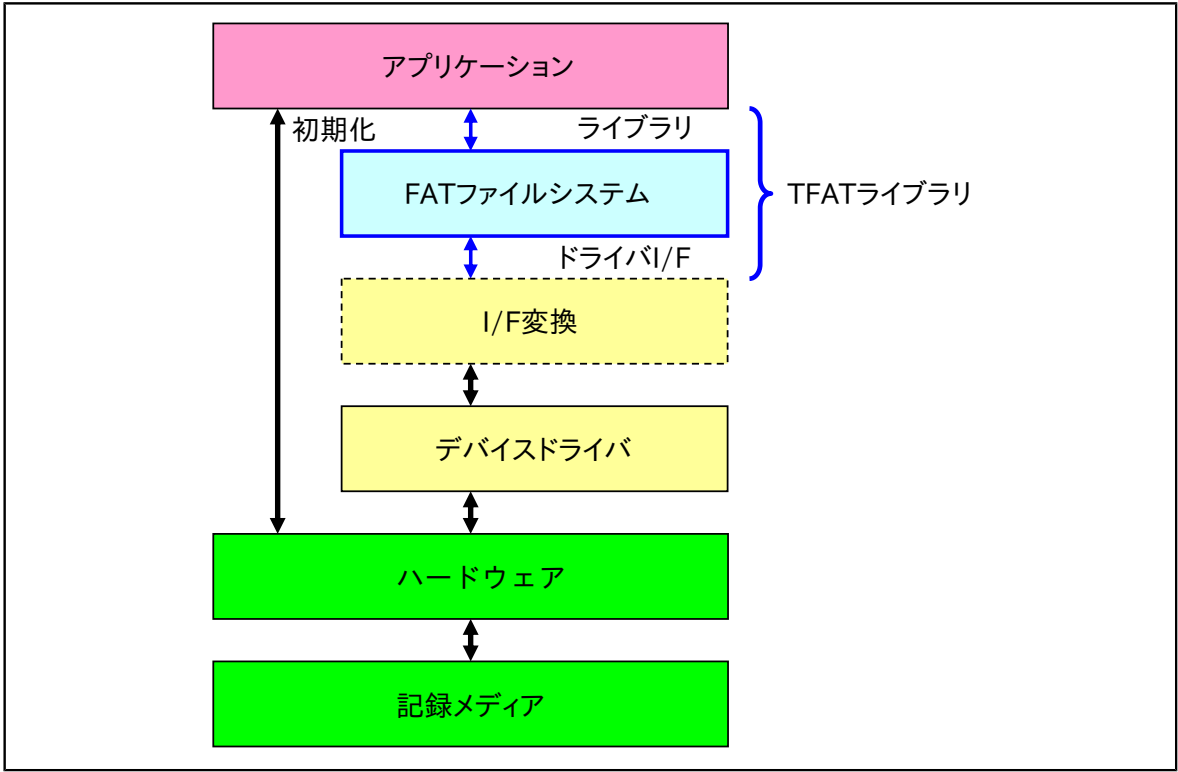


図1.1.TFATライブラリの構成

サンプルプログラムは各種Renesas Starter Kitで動作します。

各種Renesas Starter Kitにメモリカードスロット等の拡張基板が必要な場合があります。詳細はFITモジュールアプリケーションノートをご参照ください。

1.3. TFATライブラリの機能

TFATライブラリの機能一覧を以下に示します。

表1.2.機能一覧

機能	TFAT	FatFs	Windows
対応 ファイルシステム	FAT16 FAT32	FAT16 FAT32	FAT12 FAT16 FAT32 NTFS
対応 2バイトコード	日本語（S-JIS）	日本語（S-JIS） 簡体字中国語（GB18030） 韓国語（EUC） 繁体字中国語（Big5） 等	日本語（S-JIS） 等

1.4. TFATライブラリのconfig設定

TFATライブラリのconfig設定一覧を以下に示します。現バージョンでは_USE_LFN以外は設定変更できません。

表1.3.config設定一覧

オプションフラグ	説明	変更可否	規定値
_FS_TINY	標準構成と最小構成の選択	×	1
_FS_READONLY	読み込み専用のライブラリとして構築する	×	0
_FS_MINIMIZE	機能制限を行う	×	0
_USE_STRFUNC	ファイルに対し文字列を入出力する関数（fputs, fprintf等）の有効/無効の切り替え	×	0
_USE_MKFS	MKFSを使用する	×	0
_USE_FASTSEEK	高速seekを使用する	×	0
_USE_LABEL	ドライブにボリュームラベルを使用する	×	0
_USE_FORWARD	f_forwardを有効にする	×	1
_CODE_PAGE	2バイト文字の文字コードを設定する	×	932
_USE_LFN	LFNを有効にする	○	0（注1）
_MAX_LFN	LFNで利用できる文字数を規定する	×	255
_LFN_UNICODE	LFNでUNICODEを使用する	×	0
_FS_RPATH	パス制御を有効にする	×	0
_VOLUMES	ドライブ数を設定する	×	10
_MAX_SS	セクターサイズを設定する	×	512
_MULTI_PARTITION	複数パーティションを有効にする	×	0
_USE_ERASE	セクタ消去を有効にする	×	0
_WORD_ACCESS	2バイトアクセスを有効にする	×	0
_FS_REENTRANT	リエントラントを有効にする	×	0
_FS_TIMEOUT	タイムアウト値を設定する	×	1000
_SYNC_t	オブジェクトの同期の種別を設定する	×	HANDLE
_FS_LOCK	ロック機能を有効にする	×	0

（注1）0または1のみ使用可能です。

1.5. TFATとFatFsのAPI対応表

表1.4.TFATとFatFsのAPI対応表

TFAT	FatFs
R_tfat_f_mount	f_mount
R_tfat_f_open	f_open
R_tfat_f_read	f_read
R_tfat_f_write	f_write
R_tfat_f_lseek	f_lseek
R_tfat_f_close	f_close
R_tfat_f_opendir	f_opendir
R_tfat_f_readdir	f_readdir
R_tfat_f_stat	f_stat

TFAT	FatFs
R_tfat_f_getfree	f_getfree
R_tfat_f_truncate	f_truncate
R_tfat_f_sync	f_sync
R_tfat_f_unlink	f_unlink
R_tfat_f_mkdir	f_mkdir
R_tfat_f_chmod	f_chmod
R_tfat_f_utime	f_utime
R_tfat_f_rename	f_rename
R_tfat_f_forward	f_forward

2. ライブラリの型の定義

このセクションでは、TFATライブラリで使用する型の定義について詳細に説明します。

表2.1.TFATライブラリの型

Datatype	Typedef
signed char	int8_t
unsigned char	uint8_t
signed short	int16_t
unsigned short	uint16_t
signed long	int32_t
unsigned long	uint32_t
unsigned char	DSTATUS

3. ライブラリ構造体

このセクションでは、ライブラリで使用する構造体について詳細に説明します。

3.1. FATFS - ファイルシステムオブジェクト構造体

FATFS構造体は、論理ドライブに必要な作業領域を保持します。これは、アプリケーションプログラムによって割り当てられ、R_tfat_f_mount関数を使用して登録または登録解除されます。下表にFATFS構造体のメンバの詳細を示します。この構造体のメンバはアプリケーションプログラムから変更することはできません。

表3.1.FATFSのメンバ

データ型	メンバ名	説明
uint8_t	fs_type	FATサブタイプ
uint8_t	drv	物理ドライブ番号
uint8_t	csize	クラスタあたりのセクタ数
uint8_t	n_fats	FATコピー数
uint8_t	wflag	win[]ダーティフラグ（1:書き戻す必要がある）
uint8_t	fsi_flag	fsinfoダーティフラグ（1:書き戻す必要がある）
uint16_t	id	ファイルシステムマウントID
uint16_t	n_rootdir	ルートディレクトリエントリ数
uint16_t	ssize	セクタあたりのバイト数
uint32_t	last_clust	最後に割り当てられたクラスタ
uint32_t	free_clust	空きクラスタ数
uint32_t	fsi_sector	fsinfoセクタ
uint32_t	cdir	カレントディレクトリ開始クラスタ
uint32_t	n_fatent	FATエントリ数（クラスタ数+2）
uint32_t	fsize	FATあたりのセクタ
uint32_t	volbase	ドライブ開始セクタ
uint32_t	fatbase	FAT開始セクタ
uint32_t	dirbase	ルートディレクトリ開始セクタ
uint32_t	database	データ開始セクタ
uint32_t	winsect	win[]にある現在のセクタ
uint8_t	win[512]	ディレクトリ/FAT/ファイルのディスクアクセスウィンドウ

3.2. DIR - ディレクトリオブジェクト構造体

DIR構造体（ディレクトリオブジェクト）は、ディレクトリに関連するデータを保持します。ディレクトリに関連するデータは、R_tfat_f_opendirおよびR_tfat_f_readdir関数によってDIR構造体に格納されます。

表3.2.DIRのメンバ

データ型	メンバ名	説明
FATFS*	fs	ファイルシステムオブジェクトを指すポインタ
uint16_t	id	オーナーファイルシステムマウントID

データ型	メンバ名	説明
uint16_t	index	現在のインデックス
uint32_t	sclust	開始クラスタ
uint32_t	clust	現在のクラスタ
uint32_t	sect	現在のセクタ
uint8_t*	dir	win[]にある現在のSFN数のポインタ
uint8_t*	fn	現在のセクタ
uint16_t*	lfn	LFN作業用バッファのポインタ
uint16_t	lfn_idx	最後に合致したLFNインデックス値

3.3. FIL - ファイルオブジェクト構造体

FIL構造体（ファイルオブジェクト）はファイルの状態を保持します。R_tfat_f_open関数によって作成され、R_tfat_f_close関数によって破棄されます。この構造体のメンバはアプリケーションプログラムで変更することはできません。

表3.3.FILのメンバ

データ型	メンバ名	説明
FATFS*	fs	オーナファイルシステムを指すポインタ
uint16_t	id	ファイルシステムマウントID
uint8_t	flag	ファイルステータスフラグ
uint8_t	pad1	パディング
uint32_t	fptr	ファイルR/Wポインタ
uint32_t	fsize	ファイルサイズ
uint32_t	sclust	ファイル開始クラスタ
uint32_t	clust	現在のクラスタ
uint32_t	dsect	現在のセクタ
uint32_t	dir_sect	ディレクトリエントリを含むセクタ
uint8_t*	dir_ptr	ウィンドウ内のディレクトリエントリを指すポインタ

3.4. FILINFO - ファイルステータス構造体

FILINFO構造体は、R_tfat_f_stat()およびR_tfat_f_readdir()関数によって返されるファイル情報を保持します。

表3.4.FILINFOのメンバ

データ型	メンバ名	説明
uint32_t	fsize	ファイルのサイズをバイト単位で格納します。これは、ディレクトリの場合、常に0です。
uint16_t	fdate	ファイルが修正された日付またはディレクトリが作成された日付を格納します。 bit15:9 - 1980年からの年（0～127の範囲の値） bit8:5 - 月（1～12の範囲の値） bit4:0 - 日（1～31の範囲の値）

データ型	メンバ名	説明
uint16_t	ftime	ファイルが修正された時刻またはディレクトリが作成された時刻を格納します。 bit15:11 - 時間 (0～23の範囲の値) bit10:5 - 分 (0～59の範囲の値) bit4:0 - 秒/2 (0～29の範囲の値)
uint8_t	fattrib	ファイル／ディレクトリ属性を格納します。
uint8_t	fname[8+1+3+1]	ファイル／ディレクトリ名を8.3フォーマットのNULLで終了する文字列で格納します。
uint8_t*	lfname	LFNバッファのポインタ
uint16_t	lfsize	LFNバッファのサイズ

4. ライブラリ定数

このセクションでは、ライブラリで使用する定数について詳細に説明します。以下の定数は `r_tfat_lib.h` に定義されています。

4.1. FRESULT - ライブラリ関数戻り値

ライブラリ関数の戻り値は、enum型で定義されます。

表4.1.FRESULTの値

名称	値	意味
TFAT_FR_OK	0	関数は正常に実行されました。
TFAT_FR_DISK_ERR	1	エラーは、LOWレベル・ディスクI/Oレイヤーで発生しました。
TFAT_FR_INT_ERR	2	前提条件が満たされていません。
TFAT_FR_NOT_READY	3	物理ドライブは動作できません。
TFAT_FR_NO_FILE	4	関数はファイル（またはディレクトリ）を見つけることができませんでした。
TFAT_FR_NO_PATH	5	関数がパスを見つけることができませんでした。
TFAT_FR_INVALID_NAME	6	パス名が無効です。
TFAT_FR_DENIED	7	固有のエラーです。
TFAT_FR_EXIST	8	同じ名前のファイル（またはディレクトリ）がすでに存在します。
TFAT_FR_INVALID_OBJECT	9	ファイル（またはディレクトリ）オブジェクトが無効です。
TFAT_FR_WRITE_PROTECTED	10	記録メディアが書き込み保護されているので、書き込みモードで開いたり、作成したりすることはできません。
TFAT_FR_INVALID_DRIVE	11	論理ドライブ番号は無効です。
TFAT_FR_NOT_ENABLED	12	論理ドライブに作業領域がありません。
TFAT_FR_NO_FILESYSTEM	13	ディスク上に有効なFATパーティションがありません。
TFAT_FR_MKFS_ABORTED	14	<code>f_mkfs()</code> はパラメータ異常の為に中止されました。
TFAT_FR_TIMEOUT	15	タイムアウトしました。
TFAT_FR_LOCKED	16	ロックされています。
TFAT_FR_NOT_ENOUGH_CORE	17	LFN 作業領域は割り当てることが出来ません。
TFAT_FR_TOO_MANY_OPEN_FILES	18	オープンできるファイル数を超えています。
TFAT_FR_INVALID_PARAMETER	19	パラメータ異常です。

4.2. ファイル属性情報

以下のマクロはFILINFO構造体の `fattrib` メンバに設定される各ビットの内容を示します。

表4.2.ファイル属性情報のマクロ

マクロ名	値	意味
TFAT_AM_RDO	0x01	このフラグがセットされている場合、該当するファイルまたはディレクトリは読み取り専用であることを示します。
TFAT_AM_HID	0x02	このフラグがセットされている場合、該当するファイル(またはディレクトリ)は隠しファイル(またはディレクトリ)であることを示します。
TFAT_AM_SYS	0x04	このフラグがセットされている場合、該当するファイル(またはディレクトリ)はシステムファイル(またはディレクトリ)であることを示します。
TFAT_AM_DIR	0x10	このフラグがセットされている場合、ディレクトリであることを示します。
TFAT_AM_ARC	0x20	このフラグがセットされている場合、該当するファイル(またはディレクトリ)はアーカイブであることを示します。

4.3. ディスクステータスのためのマクロ

これらのマクロはDSTATUS型に設定されるディスクのステータスを示します。ユーザはメモリドライバインタフェース関数で該当するマクロを設定し、結果をライブラリに渡します。

表4.3.ディスクステータスのためのマクロ

マクロ	値	説明
TFAT_STA_NOINIT	0x01	このフラグは、ディスクドライブが初期化されていないことを示します。このフラグは、システムリセット時、ディスク取り出し時、R_tfat_disk_initialize関数の失敗時にセットされ、R_tfat_disk_initialize関数の成功時にクリアされます。
TFAT_STA_NODISK	0x02	このフラグがセットされている場合、ドライブに記録メディアが挿入されていないことを示します。ドライブに記録メディアが挿入されている場合、このフラグはクリアされます。
TFAT_STA_PROTECT	0x04	このフラグは、記録メディアが書き込み保護されていることを示します。これは、書き込み保護ノッチをサポートしていないドライブでは常にクリアされます。このフラグは、TFAT_STA_NODISKがセットされていない場合は有効ではありません。

4.4. Enum - DRESULT

ユーザがメモリドライバインタフェース関数によって実行されたディスク動作の結果を示すためにこのenumを使用します。

表4.4.DRESULTの値

名称	enum 値	意味
TFAT_RES_OK	0	関数の実行が成功しました。
TFAT_RES_ERROR	1	関数実行時にエラーが発生しました。

名称	enum 値	意味
TFAT_RES_WRPRT	2	ディスクは書き込み保護されています。
TFAT_RES_NOTRDY	3	ディスクドライブが初期化されていません。
TFAT_RES_PARERR	4	無効な引数が関数に渡されました。

5. ライブラリ関数

このセクションでは、ライブラリのAPIについて詳細に説明します。

関数名		分類
機能概要		
書式	関数の呼び出し形式を示します。#include "ヘッダファイル"で示すヘッダファイルは、この関数の実行に必要なヘッダファイルです。必ずインクルードしてください。	
引数	関数の引数を示します。"I/O"には引数がそれぞれ入力値、出力値であることを示します。"説明"には"引数名"についての説明を示します。	
戻り値	関数の戻り値を示します。"説明"には戻り値の値についての説明を示します。	
説明	関数の仕様を示します。	
注意事項	関数を使用する際の注意事項を示します。	
使用例	関数の使用例を示します。	
作成例	関数の作成例を示します。	

図5.1.ライブラリ関数詳細の見方

R_tfat_f_mount

— ワークエリアの登録・削除

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_mount (
    uint8_t Drive ,
    FATFS *FileSystemObject );
```

引数

引数名	I/O	説明
Drive	I	作業領域を登録／登録解除するための論理ドライブ番号。0以上で_VOLUMESの設定値未満
FileSystemObject	I	登録する作業領域（ファイルシステムオブジェクト構造体）を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

この関数は、記録メディアの作業領域（ファイルシステムオブジェクト）を登録／登録解除します。この作業領域は、その他のファイル操作関数を使用する前に登録しなければなりません。登録済み作業領域を破棄するには、FileSystemObjectにNULLを指定します。

この関数は、指定された作業領域を初期化して、そのアドレスを内部テーブルに登録するだけで、ディスクI/Oレイヤへのアクセスは行いません。ボリュームマウントプロセスは、R_tfat_f_mountまたは記録メディア変更後の最初のファイルアクセス時に実行されます。

ファイル関数を使用する前に、R_tfat_f_mount関数を使用して作業領域を論理ドライブに指定しなければなりません。すべてのライブラリ関数は、この手順実行後にのみ機能します。

使用例

```
FATFS fatfs;
FRESULT res;
res = R_tfat_f_mount(0, &fatfs);
```

R_tfat_f_open

— ファイルのオープン・作成

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_open (
    FIL *FileObject ,
    const uint8_t *FileName ,
    uint8_t ModeFlags );
```

引数

引数名	I/O	説明
FileObject	I/O	作成するファイルオブジェクトを指すポインタ。R_tfat_f_open関数が成功した後、ファイルを閉じるまで、このファイルオブジェクトを使用してファイルにアクセスすることができます。
FileName	I	作成するか、開くファイルの名前を指定するNULLで終了する文字列を指すポインタ。LFNが有効の場合は、ドライブ名、パス名を含め255文字以下で指定。記述方法は以下のとおりです。使用例は後述します。 "＜ドライブ名＞:＜ファイル名＞"
ModeFlags	I	ファイルのアクセスおよび開くメソッドの型を指定します。詳細については、以下の表を参照してください。

ModeFlagsは以下のマクロの組み合わせとして指定します。

値	説明
TFAT_FA_READ	オブジェクトへの読み取りアクセスを指定します。ファイルからデータを読み取ることができます。読み取り書き込みアクセスのためにTFAT_FA_WRITEと組み合わせます。
TFAT_FA_WRITE	オブジェクトへの書き込みアクセスを指定します。ファイルにデータを書き込むことができます。読み取り書き込みアクセスのためにTFAT_FA_READと組み合わせます。
TFAT_FA_OPEN_EXISTING	ファイルを開きます。ファイルが存在しない場合、関数は失敗します。（デフォルト）
TFAT_FA_OPEN_ALWAYS	存在する場合、ファイルを開きます。存在しない場合は、関数は新しいファイルを作成します。
TFAT_FA_CREATE_NEW	新しいファイルを作成します。ファイルがすでに存在している場合、関数は失敗します。
TFAT_FA_CREATE_ALWAYS	新しいファイルを作成します。ファイルが存在する場合は、切り捨てられ、上書きされます。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

FRESULTがTFAT_FR_DENIEDの場合、以下の理由で必要なアクセスが拒否されました。

- ・ 読み取り専用ファイルを書き込みモードで開きました。
- ・ 同じ名前の読み取り専用ファイル（またはディレクトリ）がすでに存在しているため、ファイル（またはディレクトリ）を作成することはできませんでした。
- ・ ディレクトリテーブルまたはディスクがいっぱいなので、ファイルを作成することはできません。

説明

この関数は、ファイルにアクセスするために使用するファイルオブジェクトを作成します。

このファイルオブジェクトは、ライブラリ関数を使用して以降のファイル操作のために使用されます。

使用例

```
FIL file; // Open a new file file.txt in the write mode
FRESULT res;
res = R_tfat_f_open(&file, "0:file.txt", TFAT_FA_CREATE_ALWAYS | TFAT_FA_WRITE);
```

R_tfat_f_close

— ファイルのクローズ

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_close (
    FIL *FileObject );
```

引数

引数名	I/O	説明
FileObject	I/O	閉じるファイルオブジェクトを指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

この関数は、開いているファイルオブジェクトを閉じます。データがファイルに書き込まれている場合、ファイルのキャッシュされた情報がディスクに書き戻されます。関数が成功すると、ファイルオブジェクトは有効でなくなり、破棄することができます。ファイルオブジェクトを読み取り専用モードで開いている場合は、この関数を呼び出さずに破棄することができます。

使用例

```
FIL file;                                // File structure variable
FRESULT res;                             // Result
res = R_tfat_f_open(&file, "0:file.txt", TFAT_FA_WRITE); // Open file
// File operations
res = R_tfat_f_close(&file);              // Close file
```

R_tfat_f_read

— ファイルの読み出し

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_read (
    FIL *FileObject ,
    void *Buffer ,
    uint16_t BytesToRead ,
    uint16_t *BytesRead );
```

引数

引数名	I/O	説明
FileObject	I/O	開いているファイルオブジェクトを指すポインタ。
Buffer	O	読み出したデータを格納するバッファを指すポインタ。
BytesToRead	I	uint16_tの範囲で読み出すバイト数。
BytesRead	O	読み出したバイト数を返すuint16_t変数を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

FRESULTがTFAT_FR_DENIEDの場合、以下の理由で必要なアクセスが拒否されました。

- ・ ファイルが非読み出しモードで開いています。

説明

R_tfat_f_read関数は、現在のファイルポインタからデータを読み出します。

ファイルオブジェクトのファイルポインタは、読み出したバイト数だけインクリメントされます。関数が成功すると、*BytesReadをチェックしてファイルの終わりを確認する必要があります。*BytesReadがBytesToRead未満の場合は、R/Wポインタが読み取り動作時にファイルの終端に達したことを意味します。

使用例

```
FIL file;                                // File structure
uint8_t Buff[128];                       // Data Buffer
FRESULT res;                             // Result
uint16_t file_read_cnt;                  // Data read count
R_tfat_f_open(&file, "0:file.txt", TFAT_FA_READ); // Open file in read mode
res = R_tfat_f_read(&file, Buff, 128, &file_read_cnt); // Read 128 bytes
R_tfat_f_close(&file);                   // Close file
```

R_tfat_f_write

— ファイルの書き込み

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_write (
    FIL *FileObject ,
    void *Buffer ,
    uint16_t BytesToWrite ,
    uint16_t *BytesWritten );
```

引数

引数名	I/O	説明
FileObject	I/O	開いているファイルオブジェクトを指すポインタ。
Buffer	I	書き込むバッファを指すポインタ。
BytesToWrite	I	uint16_tの範囲で書き込むバイト数を指定します。
BytesWritten	O	書き込まれたバイト数を返すuint16_t変数を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

FRESULTがTFAT_FR_DENIEDの場合、以下の理由で必要なアクセスが拒否されました。

- ・ ファイルが非書き込みモードで開いています。

説明

R_tfat_f_write関数は、ファイルにデータを書き込みます。

ファイルオブジェクト内の読み取り／書き込みポインタは、書き込まれたバイト数だけインクリメントされます。関数が成功した後、*ByteWrittenをチェックして、ディスクがいっぱいであるかどうかを確認してください。*BytesWrittenがBytesToWrite未満である場合は、書き込み動作時にディスクがいっぱいになったことを意味します。

使用例

```
FIL file; // File structure
uint8_t Buff[] = "Hello"; // Data Buffer
FRESULT res; // Result
uint16_t file_write_cnt; // Data write count
R_tfat_f_open(&file, "0:file.txt", TFAT_FA_WRITE); // Open file in write mode
res = R_tfat_f_write(&file,Buff,(sizeof(Buff)-1),&file_write_cnt); //Write data
R_tfat_f_close(&file); // Close file
```


R_tfat_f_lseek

— リード/ライト・ポインタの移動, ファイルの拡張

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_lseek (
    FIL *FileObject ,
    uint32_t Offset );
```

引数

引数名	I/O	説明
FileObject	I/O	開いているファイルオブジェクトを指すポインタ。
Offset	I	ファイルの先頭からのバイト数のオフセット。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

この関数は、開いているファイルオブジェクトのファイル読み取り／書き込みポインタを移動します。この関数を使用してファイルサイズを拡張することもできます。

R_tfat_f_lseek関数は、開いているファイルのファイルR/Wポインタを移動します。オフセットはファイルの先頭からのみ指定することができます。ファイルサイズを超えるオフセットを書き込みモードで指定すると、ファイルサイズはオフセットに拡張され、拡張された領域のデータは未定義になります。これは、迅速な書き込み動作のために大きなファイルを迅速に作成するのに適しています。R_tfat_f_lseek関数が成功したら、R/Wポインタが正しく移動されたことを確認するためにファイルオブジェクトのメンバfptrをチェックしてください。fptrが期待値未満である場合は、以下のいずれかが生じた可能性があります。

- ・ 読み取り専用モードで、オフセットはファイルサイズに切り捨てられます。
- ・ ファイル拡張プロセス中にドライブがいっぱいになりました。
- ・ FAT構造体にエラーがあります。

使用例

```
FRESULT res;                // Result
// Move to offset of 5000 from top of the file.
res = R_tfat_f_lseek(&file, 5000);

// Forward 3000 bytes
res = R_tfat_f_lseek(&file, file.fptr + 3000);
```

R_tfat_f_truncate

— ファイル・サイズの切り詰め

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_truncate (
    FIL *FileObject );
```

引数

引数名	I/O	説明
FileObject	I/O	切り捨てる、開いているファイルオブジェクトを指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

FRESULTがTFAT_FR_DENIEDの場合、以下の理由で必要なアクセスが拒否されました。

- ・ ファイルが読み取り専用モードで開いています。

説明

R_tfat_f_truncate関数は、ファイルサイズを切り捨てます。

R_tfat_f_truncate関数は、現在のファイルR/Wポインタまでファイルサイズを切り捨てます。ファイルR/Wポインタがすでにファイルの終わりにある場合は、効果はありません。現在のR/Wポインタは、R_tfat_f_lseek関数を使用して希望する切り捨てポイントにすることができます。

使用例

```
FRESULT res;                                // Result
// To truncate a file at 5000 bytes from the start.
R_tfat_f_lseek(&file, 5000);                 // Offset of 5000 from top of the file.
res = R_tfat_f_truncate(&file);              // Truncate the file
```

R_tfat_f_sync

— キャッシュされたデータのフラッシュ

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_sync (
    FIL *FileObject );
```

引数

引数名	I/O	説明
FileObject	I/O	フラッシュする、開いているファイルオブジェクトを指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

この関数は、R_tfat_f_close関数と同じプロセスを実行しますが、ファイルは開いたままで、ファイルに対する読み取り／書き込み／シークを続けることができます。これは、データロガーなどの、ファイルを書き込みモードで長時間開いておく用途に適しています。R_tfat_f_writeの後に定期的にまたは直ちにR_tfat_f_syncを実行すると、突然の電源切断または偶発的なディスク取り出しによるデータ損失のリスクを最小限に抑えることができます。

使用例

```
FRESULT res;                // Result
// To synchronize the file at any point.
res = R_tfat_f_sync(&file);
```

R_tfat_f_opendir

— ディレクトリのオープン

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_opendir (
    DIR *DirObject ,
    const uint8_t *DirName );
```

引数

引数名	I/O	説明
DirObject	I/O	作成する空のディレクトリオブジェクトを指すポインタ。
DirName	I	開くディレクトリ名を指定するNULLで終了する文字列を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

R_tfat_f_opendir関数は、ディレクトリを開きます。

この関数は、既存のディレクトリを開き、ディレクトリオブジェクトを作成します。このディレクトリオブジェクトはR_tfat_f_readdirとともに使用することができます。ディレクトリオブジェクト構造体は、手順を使用せずにいつでも破棄することができます。

使用例

```
DIR dir;
FRESULT res;                                // Result
res = R_tfat_f_opendir(&dir, "0:abc");      // Open a directory named "abc"
```

R_tfat_f_readdir

— ディレクトリの読み出し

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_readdir (
    DIR *DirObject ,
    FILINFO *FileInfo );
```

引数

引数名	I/O	説明
DirObject	I/O	開いているディレクトリ構造体を指すポインタ。
FileInfo	O	読み取られたアイテムを格納するファイル情報構造体を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

R_tfat_f_readdir関数は、ディレクトリエントリを読み取ります。

この関数は、ディレクトリ内のエントリを読み取ります。R_tfat_f_readdir関数を繰り返し呼び出すことによりディレクトリ内のすべてのアイテムを読み取ることができます。すべてのディレクトリアイテムが読み取られ、さらに読み取るアイテムがない場合、関数はエラーなしにNULL文字列をfname[]メンバに返します。ファイル情報の詳細については、FILINFO構造体を参照してください。

この関数は、R_tfat_f_opendir関数を使用して有効なディレクトリをDIR*ポインタ変数に割り当てずに使用することはできません。

使用例

```
DIR dir;
FILINFO finfo;
FRESULT res;                                // Result
R_tfat_f_opendir(&dir, "0:abc");             // Open a directory named "abc"
res = R_tfat_f_readdir(&dir, &finfo);        // Read directory "abc" into finfo
```

R_tfat_f_getfree

— ボリューム空き領域の取得

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_getfree (
    const uint8_t *Path ,
    uint32_t *Clusters ,
    FATFS **FileSystemObject );
```

引数

引数名	I/O	説明
Path	I	調べる対象の論理ドライブを示すパス名を示すNULLで終了する文字列のポインタを指定します。ヌル文字列はカレント・ドライブを意味します。
Clusters	O	空きクラスタ数を格納する変数を指すポインタ。
FileSystemObject	O	対応するファイルシステムオブジェクトを格納するポインタを指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

R_tfat_f_getfree関数は、空きクラスタ数を取得します。 ファイルシステムオブジェクトのメンバcsizeは、クラスタあたりのセクタ数を指定し、セクタ単位の空きスペースを計算することができます。 FAT32でフォーマットされたドライブでは、FSINFOセクタの状態によっては不正確な値を返したり、処理に時間がかかったりする可能性があります。

使用例

```
FATFS *fs;
uint32_t clust, tot_disk_space, free_disk_space;
FRESULT res;                                // Result
res = R_tfat_f_getfree("0:", &clust, &fs);    // Get free clusters
// Total disk space available
tot_disk_space = (fs->max_clust - 2) * fs->csiz / 2;
// Free disk space available
free_disk_space = clust * fs->csiz / 2;
```

R_tfat_f_stat

— ファイル・ステータスの取得

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_stat (
    const uint8_t *FileName ,
    FILINFO *FileInfo );
```

引数

引数名	I/O	説明
FileName	I	情報を取得するためのファイルまたはディレクトリを指定するNULLで終了する文字列を指すポインタ。
FileInfo	O	情報を格納する空のFILINFO構造体を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

関数R_tfat_f_statは、ファイルまたはディレクトリの情報を取得します。
情報は、構造体ポインタによって指される構造体に格納されます。

使用例

```
FILINFO filinfo;
FRESULT res;                                // Result
res = R_tfat_f_stat ("0:file.txt", &filinfo); // Retrieve file information
```

R_tfat_f_mkdir

— ディレクトリの作成

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_mkdir (
    const uint8_t *DirName );
```

引数

引数名	I/O	説明
DirName	I	作成するディレクトリの名前を指定するNULLで終了する文字列を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

FRESULTがTFAT_FR_DENIEDの場合、ディレクトリテーブルまたはディスクが一杯なので、ディレクトリを作成することができませんでした。

説明

関数R_tfat_f_mkdirは、新しいディレクトリを作成します。

使用例

```
FRESULT res;           // Result
res = R_tfat_f_mkdir("0:abc"); // Make a directory "abc"
```


R_tfat_f_unlink

— ファイル/ディレクトリの削除

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_unlink (
    const uint8_t *FileName );
```

引数

引数名	I/O	説明
FileName	I	削除するファイル（またはディレクトリ）を指定する NULLで終了する文字列を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数 実行の結果

FRESULTがTFAT_FR_DENIEDの場合、以下の理由で関数が拒否されました。

- ・ ファイル（またはディレクトリ）は読み取り専用属性を持っています。
- ・ ディレクトリが空ではありません。

説明

関数R_tfat_f_unlinkは、ファイル（またはディレクトリ）を削除します。

使用例

```
FRESULT res;                                // Result
res = R_tfat_f_unlink("0:file.txt");         // Remove file "file.txt"
```

R_tfat_f_chmod

— ファイル/ディレクトリの属性の変更

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_chmod (
    const uint8_t *FileName ,
    uint8_t Attribute ,
    uint8_t AttributeMask );
```

引数

引数名	I/O	説明
FileName	I	ファイル（またはディレクトリ）を指定するNULLで終了する文字列を指すポインタ。
Attribute	I	以下の表の1つまたは複数のフラグを組み合わせで設定する属性フラグ指定されたフラグが設定され、他のフラグはクリアされます。
AttributeMask	I	どの属性が変更されたかを指定する属性マスク指定された属性が設定またはクリアされます。

Attribute, AttributeMaskは以下のマクロの組み合わせとして指定します。

値	属性
TFAT_AM_RDO	読み取り専用
TFAT_AM_HID	隠しファイル
TFAT_AM_SYS	システム
TFAT_AM_ARC	アーカイブ

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

R_tfat_f_chmod関数は、ファイル（またはディレクトリ）の属性を変更します。

使用例

```
FRESULT res;                                // Result
// Set read-only flag, clear archive flag and retain others
res = R_tfat_f_chmod("0:file.txt", TFAT_AM_RDO, TFAT_AM_RDO | TFAT_AM_ARC);
```

R_tfat_f_utime

— ファイル/ディレクトリのタイムスタンプの変更

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_utime (
    const uint8_t *FileName ,
    const FILINFO *TimeDate );
```

引数

引数名	I/O	説明
FileName	I	ファイル（またはディレクトリ）を指定するNULLで終了する文字列を指すポインタ。
TimeDate	I	メンバfdateおよびftimeに格納されているタイムスタンプを持つファイル情報構造体を指すポインタ。その他の構造体メンバ値は関連がありません。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

R_tfat_f_utime関数は、ファイル（またはディレクトリ）のタイムスタンプを変更します。日付および時刻値は、引数として渡すFILINFO構造体の対応するメンバに設定する必要があります。

使用例

```
FILINFO filinfo;
FRESULT res;                                // Result
// please refer sec 3.4 FILINFO structure explanation
filinfo.fdate = 0x3A4D;                     // Date set to 13th Feb 2009
filinfo.ftime = 0x7322;                     // Time set to 14:25:04
res = R_tfat_f_utime("0:file.txt", &filinfo); // Update the time
```

R_tfat_f_rename

— ファイル/ディレクトリの名前変更・移動

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_rename (
    const uint8_t *OldName ,
    const uint8_t *NewName );
```

引数

引数名	I/O	説明
OldName	I	名前を変更するファイル（またはディレクトリ）を指定するNULLで終了する文字列を指すポインタ。
NewName	I	NULLで終了する文字列を指すポインタは、ドライブ番号のない新しいファイル／ディレクトリ名を指定します。既存のファイル名を指定することはできません。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

FRESULTがTFAT_FR_DENIEDの場合、何らかの理由でファイルの新しい名前を作成することはできませんでした。

説明

R_tfat_f_rename関数は、ファイル（またはディレクトリ）の名前を変更します。また、この関数はファイルを別のディレクトリに移動することができます。

使用例

```
FRESULT res;                                // Result
// Rename file or directory
res = R_tfat_f_rename("0:oldname.txt", "newname.txt");

// Rename and move file to other directory simultaneously
res = R_tfat_f_rename("0:oldname.txt", "dir1/newname.txt");
```

R_tfat_f_forward

— ファイル・データをストリーム関数に転送

書式

```
#include "r_tfat_lib.h"

FRESULT R_tfat_f_forward (
    FIL *FileObject ,
    uint16_t (*Func)(const uint8_t*,uint16_t) ,
    uint16_t ByteToFwd ,
    uint16_t *ByteFwd );
```

引数

引数名	I/O	説明
FileObject	I/O	開いているファイルオブジェクトを指すポインタ。
Func	I	データを渡すユーザ定義関数へのポインタ。
ByteToFwd	I	uint16_tの範囲で転送するバイト数を指定します。
ByteFwd	O	実際に転送されたバイト数を返すuint16_t変数を指すポインタ。

戻り値

戻り値	説明
FRESULT	「FRESULT - ライブラリ関数戻り値」に記載した関数実行の結果

説明

R_tfat_f_forward関数は、ファイルのデータをバッファに読み出さずにユーザ定義関数へ直接渡します。

第2引数に与えるユーザ定義関数の詳細は、「R_tfat_outstream」を参照してください。

使用例

```
FRESULT res;                // Result
FIL file;                   // File structure
uint16_t file_forwarded_cnt; // Data forwarded count

R_tfat_f_open(&file, "0:file.txt", TFAT_FA_READ); // Open file in read mode

// File output stream
res = R_tfat_f_forward(&file,R_tfat_outstream,10,&file_forwarded_cnt);
```

6. メモリドライバインタフェース関数

このセクションでは、メモリドライバインタフェース関数について詳細に説明します。これらの関数のプロトタイプと各関数の実装で必要な処理について説明します。これらの関数の実装は、ユーザに使用可能なメモリドライバとともに使用できるように作成する必要があります。これらはユーザ定義関数です。

R_tfat_disk_initialize

— ディスク・ドライブの初期化

書式

```
#include "r_tfat_lib.h"

DSTATUS R_tfat_disk_initialize (
    uint8_t pdrv);
```

引数

引数名	I/O	説明
Drive	I	初期化するドライブ番号を指定します。

戻り値

戻り値	説明
DSTATUS	関数実行後のディスクのステータス。

説明

この関数は、ディスクドライブを初期化するコードで構成する必要があります。この関数は以下を実行します。

- ・ メモリカードへの電源を有効にします。
- ・ カードが使用可能な電圧に対応しているかどうかをチェックします。
- ・ カードタイプを識別します。

R_tfat_disk_read

— ディスクからの読み込み

書式

```
#include "r_tfat_lib.h"

DRESULT R_tfat_disk_read (
    uint8_t Drive ,
    uint8_t *Buffer ,
    uint32_t SectorNumber ,
    uint8_t SectorCount );
```

引数

引数名	I/O	説明
Drive	I	物理的なドライブ番号を指定します。値は常に0になります。
Buffer	O	読み取りデータを格納するバッファを指すポインタ。
SectorNumber	I	開始セクタ番号を論理ブロックアドレス（LBA）で指定します。
SectorCount	I	読み取るセクタ数を指定します。値は1～255の範囲です。

戻り値

戻り値	説明
DRESULT	「Enum - DRESULT」に記載した関数実行の結果

説明

この関数は、ディスクドライブから読み取るコードで構成する必要があります。読み取るデータ位置に関する詳細は引数で指定します。

R_tfat_disk_write

— ディスクへの書き込み

書式

```
#include "r_tfat_lib.h"

DRESULT R_tfat_disk_write (
    uint8_t Drive ,
    uint8_t *Buffer ,
    uint32_t SectorNumber ,
    uint8_t SectorCount );
```

引数

引数名	I/O	説明
Drive	I	物理的なドライブ番号を指定します。値は常に0になります。
Buffer	I	書き込むデータを指すポインタ。
SectorNumber	I	開始セクタ番号を論理ブロックアドレス(LBA)で指定します。
SectorCount	I	読み取るセクタ数を指定します。値は1～255の範囲です。

戻り値

戻り値	説明
DRESULT	「Enum - DRESULT」に記載した関数実行の結果

説明

この関数は、ディスクドライブにデータを書き込むためのコードで構成する必要があります。書き込むデータに関する詳細は引数で指定します。

R_tfat_disk_ioctl

— その他のドライブ制御

書式

```
#include "r_tfat_lib.h"

DRESULT R_tfat_disk_ioctl (
    uint8_t Drive ,
    uint8_t Command ,
    void *Buffer );
```

引数

引数名	I/O	説明
Drive	I	物理的なドライブ番号を指定します。値は常に0になります。
Command	I	コマンド値を指定します。コマンド値は常に0になります。
Buffer	I	ポインタは常にNULLポインタになります。

戻り値

戻り値	説明
DRESULT	「Enum - DRESULT」に記載した関数実行の結果

説明

R_tfat_disk_ioctl関数は、すべてのTFATライブラリ関数の中でR_tfat_f_sync関数によってのみ使用されます。R_tfat_f_sync関数をアプリケーションで使用しないユーザは、この特定のドライバインタフェース関数の実装をスキップすることができます。

R_tfat_f_sync関数をアプリケーションで使用するユーザは、この特定のドライバインタフェース関数を実装しなければなりません。このドライバ関数は、保留中の書き込みプロセスを終了するためのコードから構成する必要があります。ディスクI/Oモジュールが書き戻しキャッシュを持つ場合、ダーティセクタは直ちにフラッシュされます。R_tfat_f_sync関数は、引数として渡すファイルオブジェクトと関連する保存されていないデータを保存します。

R_tfat_disk_status

— ディスク・ドライブの状態取得

書式

```
#include "r_tfat_lib.h"

DSTATUS R_tfat_disk_status (
    uint8_t Drive );
```

引数

引数名	I/O	説明
Drive	I	物理的なドライブ番号を指定します。値は常に0になります。

戻り値

戻り値	説明
DSTATUS	関数実行後のディスクのステータス。

説明

この関数は、ディスクをチェックするコードから構成し、現在のディスクステータスを返します。ディスクステータスは、「ディスクステータスのためのマクロ」に記載するように3つの値のいずれかになります。ディスクステータスは、ディスクステータスと関連するマクロを使用して戻り値を更新することにより、返すことができます。

R_tfat_get_fattime

— 日付・時刻の取得

書式

```
#include "r_tfat_lib.h"

uint32_t R_tfat_get_fattime (
    void );
```

引数

なし

戻り値

uint32_t - 戻り値の説明については、以下の表を参照してください。

ビット範囲	値範囲	意味
31～25	0～127	1980からの年
24～21	1～12	月
20～16	1～31	日
15～11	0～23	時間
10～5	0～59	分
4～0	0～29	秒/2の値(2秒単位)

説明

この関数は、現在の日付と時刻を返します。

ファイル操作時に日付を取得するためのライブラリ関数によって使用されます。

R_tfat_outstream

— ストリームへの転送

書式

```
uint16_t R_tfat_outstream (
    uint8_t *ptr ,
    uint16_t cnt );
```

引数

引数名	I/O	説明
ptr	I	(引数cntが0の場合)無効 (引数cntが1の場合)転送するデータの先頭アドレス
cnt	I	(0の場合)ストリームの状態確認 (0以外の場合)転送バイト数

戻り値

戻り値	説明
uint16_t	(引数cntが0の場合)0: ビジー, 1: レディ (引数cntが1の場合)転送したバイト数

説明

この関数の実装は、R_tfat_f_forward関数を実行する場合に必要です。関数名はR_tfat_outstreamである必要はありません。

オープンソースFATファイルシステム M3S-TFAT-Tiny
ユーザーズマニュアル

発行年月日	2014年 4月 1日	Rev.3.01
	2014年 4月 1日	Rev.3.01
発行	ルネサス エレクトロニクス株式会社	
	〒211-8668 神奈川県川崎市中原区下沼部 1753	



ルネサス エレクトロニクス株式会社

営業お問合せ窓口

<http://www.renesas.com>

営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町 2-6-2 (日本ビル)

技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>

オープンソースFATファイルシステム M3S-TFAT-Tiny
ユーザーズマニュアル