

RX ファミリ

R20AN0335JJ0103

Rev.1.03

2016.04.01

M3S-TFAT-Tiny メモリドライバインタフェースモジュール Firmware Integration Technology

要旨

本アプリケーションノートでは、Firmware Integration Technology (以降、FIT)を使用した RX ファミリオープンソース FAT ファイルシステム M3S-TFAT-Tiny V.3.03 Release 00 (以降、TFAT ライブラリと呼ぶ)と各メモリドライバとをつなぐメモリドライバインタフェースについて説明します。

FIT モジュールについては、

<http://japan.renesas.com/products/mpumcu/rx/child/fit.jsp>

をご覧ください。

本アプリケーションノートで提供されるドライバインタフェースモジュールで対応しているメモリドライバは、USB と SDHI,USB Mini の 3 つです。使用するには、下記 FIT モジュールが必要です。

機能	ミドルウェア製品	ウェブページ
ファイルシステム (※1)	M3S-TFAT-Tiny (R20AN0038)	http://japan.renesas.com/mw/tfat
USB ドライバ (※2)	USB Basic Host and Peripheral Driver USB Host Mass Storage Class Driver	http://japan.renesas.com/driver/usb
SDHI ドライバ(※2)	SDHI ドライバ	公開しておりません。 お問い合わせください。
USB Mini ドライバ (※2)	USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) USB Host Mass Storage Class Driver for USB Mini Firmware	http://japan.renesas.com/driver/usb

※1 必須です。

※2 どちらか一方の入手で構いません。お使いの評価環境に応じて入手してください。

動作確認デバイス

RX ファミリ

目次

1. 概要.....	4
1.1 本アプリケーションノートについて.....	4
1.2 アプリケーションノート構成	4
1.3 ソフトウェア構成	5
2. API 情報	6
2.1 ハードウェア要件	6
2.2 ソフトウェア要件	6
2.3 サポートされているツールチェーン.....	6
2.4 ヘッドファイル	6
2.5 整数型.....	7
2.6 コンフィグレーションの概要	7
2.7 引数	8
2.8 戻り値.....	8
2.9 プロジェクトへのモジュールの追加.....	8
3. API 関数	9
3.1 R_tfat_disk_initialize	9
3.2 R_tfat_disk_read	10
3.3 R_tfat_disk_write.....	10
3.4 R_tfat_disk_ioctl.....	11
3.5 R_tfat_disk_status.....	11
3.6 R_tfat_get_fattime	12
3.7 R_tfat_drv_change_alloc	13
4. 内部関数.....	14
4.1 USB 用	14
4.1.1 R_tfat_usb_disk_initialize	14
4.1.2 R_tfat_usb_disk_read	15
4.1.3 R_tfat_usb_disk_write.....	15
4.1.4 R_tfat_usb_disk_ioctl.....	16
4.1.5 R_tfat_usb_disk_status.....	16
4.1.6 R_usb_hmsc_WaitLoop	17
4.2 SDHI 用	17
4.2.1 R_tfat_sdhi_disk_initialize.....	18
4.2.2 R_tfat_sdhi_disk_read	18
4.2.3 R_tfat_sdhi_disk_write.....	19
4.2.4 R_tfat_sdhi_disk_ioctl	19
4.2.5 R_tfat_sdhi_disk_status.....	20
4.3 USB Mini 用	20
4.3.1 R_tfat_usb_mini_disk_initialize.....	21
4.3.2 R_tfat_usb_mini_disk_read	21
4.3.3 R_tfat_usb_mini_disk_write	22
4.3.4 R_tfat_usb_mini_disk_ioctl	23
4.3.5 R_tfat_usb_mini_disk_status	23

4.3.6 R_usb_mini_hmsc_WaitLoop	24
--------------------------------------	----

1. 概要

1.1 本アプリケーションノートについて

本アプリケーションノートでは、FIT を使用した TFAT ライブラリと各メモリドライバとをつなぐメモリドライバインタフェースについて説明します。

メモリドライバインタフェースは、コンフィグファイルの変更によって制御対象を切り替えることができます。

本モジュールで提供する API は、TFAT ライブラリから呼び出されます。ユーザ側で新たに呼び出す必要はありません。

TFAT ライブラリで管理するドライブ番号と USB、SDHI ドライバで管理するドライブ番号は等しくありません。その為、本モジュール内で変換テーブルを持っています。初期値は、コンフィグレーション設定で決まります。動的に変更したい場合は、3.7 章を参照してください。

1.2 アプリケーションノート構成

本アプリケーションノートは、以下のものから構成されています。

表 1.2.1 アプリケーションノート構成

ファイル/ディレクトリ名		内容
R20AN0335JJ0103_rx_tfat.pdf		アプリケーションノート
reference_document		
r01an1723ju0111_rx.pdf		e ² studio に組み込む方法
r01an1826jj0102_rx.pdf		CS+に組み込む方法
FITModules		
r_tfat_driver_rx_v1.03.xml		FIT プラグイン XML
r_tfat_driver_rx_v1.03.zip		FIT プラグイン ZIP
コンフィグレーション (r_config)		
r_tfat_driver_rx_config.h		コンフィグレーションファイル(デフォルト設定)
FIT Module 本体 (r_tfat_driver_rx)		
ドキュメント(doc)		
英語版(en)		
r20an0335ej0103_rx_tfat.pdf		アプリケーションノート(英語版)
日本語版(ja)		
e20an0335jj0103_rx_tfat.pdf		アプリケーションノート(日本語版)
コンフィグレーションリファレンス(ref)		
r_tfat_driver_rx_config_reference.h		コンフィグレーションファイル(テンプレート)
ソースコード(src)		
readme (readme.txt)		readme
r_tfat_driver_rx_if.h		ヘッダファイル

1.3 ソフトウェア構成

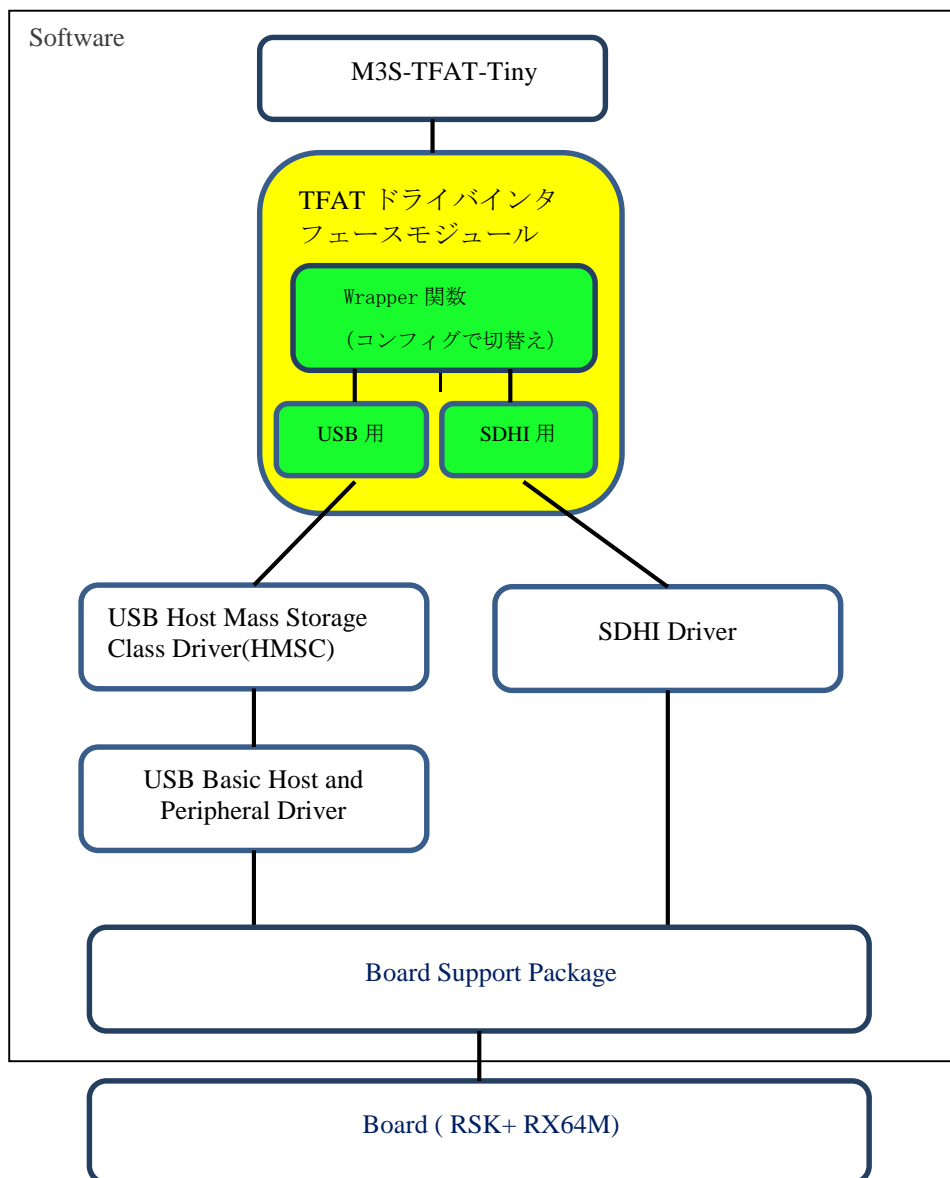


表 1.3.1 使用した FIT モジュールバージョン

ミドルウェア製品	version
ボードサポートパッケージ(BSP)	3.30
M3S-TFAT-Tiny	3.03
USB ドライバ	
USB Basic Host and Peripheral Driver	1.11
USB Host Mass Storage Class Driver	1.11
SDHI ドライバ	1.00
USB Mini ドライバ	
USB Basic Host and Peripheral Driver (USB Mini Firmware)	1.02
USB Host Mass Storage Class Driver for USB Mini Firmware	1.02

2. API 情報

2.1 ハードウェア要件

ご使用になる MCU が以下の機能をサポートしている必要があります。

- USB
- SDHI

2.2 ソフトウェア要件

本 FIT モジュールは、以下のパッケージに依存しています。

- r_bsp
- r_tfat_rx
- r_usb_basic
- r_usb_hmsc
- r_sdhi_rx
- r_usb_basic_mini
- r_usb_hmsc_mini

2.3 サポートされているツールチェーン

本 FIT モジュールは、以下のツールチェーンで動作を確認しています。

Renesas RxC Toolchain v.2.04.01

2.4 ヘッダファイル

すべての API 呼び出しはこのソフトウェアのプロジェクトコードとして提供されている 1 個のファイル「r_tfat_driver_rx_if.h」をインクルードすることによって行われます。

ビルドタイムのコンフィグレーションオプションは、ファイル「r_tfat_driver_rx_config.h」で選択または定義されます。

2.5 整数型

このプロジェクトでは、コードをわかりやすく、移植性をより大きくするために、ANSI C99 の固定長整数型（exact width integer type）を使用しています。これらの型は `stdint.h` で定義されています。

2.6 コンフィグレーションの概要

本モジュールのコンフィギュレーションオプションの設定は、`r_tfat_driver_rx_config.h` で行います。
オプション名および設定値に関する説明を下表に示します。

Configuration options in <code>r_tfat_driver_rx_config.h</code>	
<code>#define</code> <code>TFAT_USB_DRIVE_NUM</code> - Default value = (0)	USB として割り当てるドライブ数 未使用時は(0)としてください。
<code>#define</code> <code>TFAT_SDHI_DRIVE_NUM</code> - Default value = (0)	SDHI として割り当てるドライブ数 未使用時は(0)としてください。
<code>#define</code> <code>TFAT_USB_MINI_DRIVE_NUM</code> - Default value = (0)	USB Mini として割り当てるドライブ数 未使用時は(0)としてください。
<code>#define</code> <code>TFAT_DRIVE_ALLOC_NUM_i</code> <code>i = 0~9</code> - Default value = (NULL)	各ドライブ番号に対して、使用するデバイスを割り当てます。 USB で使用するドライブ = (TFAT_CTRL_USB) SDHI で使用するドライブ=(TFAT_CTRL_SDHI) USB Mini で使用するドライブ=(TFAT_CTRL_USB_MINI) としてください。 使用しないドライブ=(NULL) としてください。 このデータを元に、ドライブ番号(TFAT ライブラリ)とメモリドライバのドライブ番号との関連付けを行います。 メモリドライバのドライブ番号は、昇順に割り当てられます。動的に変更したい場合は、3.7 章を参照してください。

2.7 引数

TFAT ライブラリの API を呼び出す際のドライブ番号の定義としてご使用ください。

```
typedef enum
{
    TFAT_DRIVE_NUM_0 = 0x00,
    TFAT_DRIVE_NUM_1,
    TFAT_DRIVE_NUM_2,
    TFAT_DRIVE_NUM_3,
    TFAT_DRIVE_NUM_4,
    TFAT_DRIVE_NUM_5,
    TFAT_DRIVE_NUM_6,
    TFAT_DRIVE_NUM_7,
    TFAT_DRIVE_NUM_8,
    TFAT_DRIVE_NUM_9,
}TFAT_DRV_NUM;
```

2.8 戻り値

それぞれ TFAT モジュールの"r_tfat_lib.h"で定義されています。

```
/* Disk Status Bits (DSTATUS) */
typedef uint8_t DSTATUS;

- #define TFAT_STA_NOINIT      0x01  /* Drive not initialized */
- #define TFAT_STA_NODISK     0x02  /* No medium in the drive */
- #define TFAT_STA_PROTECT    0x04  /* Write protected */

/* Results of Disk Functions */
typedef enum
{
    TFAT_RES_OK = 0,          /* 0: Successful */
    TFAT_RES_ERROR,          /* 1: R/W Error */
    TFAT_RES_WRPRT,          /* 2: Write Protected */
    TFAT_RES_NOTRDY,         /* 3: Not Ready */
    TFAT_RES_PARERR          /* 4: Invalid Parameter */
} DRESULT;
```

2.9 プロジェクトへのモジュールの追加

e2 studio/CS+に組み込む方法は、” r01an1723ju0111_rx.pdf”(e2 studio)か”r01an1826jj0102_rx.pdf” (CS+)をご参照ください。

3. API 関数

TFAT モジュールから本関数群は呼び出されます。ここに記載された関数は、一部例外(※1)を除いてコンフィグレーション設定に応じて、メモリ種類ごとに用意された下位層の関数を呼び分けます。下位層の関数は、「4.内部関数」に記載しています。

表 3.1 関数一覧

関数名	機能概要
R_tfat_disk_initialize	ディスク・ドライブの初期化
R_tfat_disk_read	ディスクからの読み込み
R_tfat_disk_write	ディスクへの書き込み
R_tfat_disk_ioctl	その他のドライブ制御
R_tfat_disk_status	ディスク・ドライブの状態取得
R_tfat_get_fattime (※1)	日付・時刻の取得
R_tfat_drv_change_alloc (※1)	TFAT モジュールのドライブ番号とメモリドライバのドライブ番号(もしくはチャンネル番号)との関連付けの変更

※1 内部関数へ展開されない関数

3.1 R_tfat_disk_initialize

Description

本関数は、ディスク・ドライブの初期化を行います。

Usage

```
#include "r_tfat_lib.h"
DSTATUS R_tfat_disk_initialize (uint8_t drive);
```

Parameters

drive 入力 初期化するドライブ番号を指定します。

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

この関数は、コンフィグレーション設定に応じて内部関数を呼び出します。内部関数は、R_tfat_(memory name)_disk_initialize となります。

3.2 R_tfat_disk_read

Description

本関数は、ディスクからの読み込みを行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_disk_read (    uint8_t  drive ,
                              uint8_t  *buffer ,
                              uint32_t  sector_number ,
                              uint8_t  sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	出力	読み取りデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	読み取るセクタ数を指定します。値は 1～255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

この関数は、コンフィグレーション設定に応じて内部関数を呼び出します。内部関数は、R_tfat_(memory name)_disk_read となります。

3.3 R_tfat_disk_write

Description

本関数は、ディスクへの書き込みを行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_disk_write (    uint8_t  drive ,
                              uint8_t  *buffer ,
                              uint32_t  sector_number ,
                              uint8_t  sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	入力	読み取りデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	読み取るセクタ数を指定します。値は 1～255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

この関数は、コンフィグレーション設定に応じて内部関数を呼び出します。内部関数は、R_tfat_(memory name)_disk_write となります。

3.4 R_tfat_disk_ioctl

Description

本関数は、その他のドライブ制御を行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_disk_ioctl (    uint8_t  drive ,
                               uint8_t  command ,
                               void      *buffer );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
command	入力	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	入力	読み取りデータを格納するバッファを指すポインタ。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

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。

この関数は、コンフィグレーション設定に応じて内部関数を呼び出します。内部関数は、R_tfat_(memory name)_disk_ioctl となります。

3.5 R_tfat_disk_status

Description

本関数は、ディスク・ドライブの状態取得を行います。

Usage

```
#include "r_tfat_lib.h"
DSTATUS R_tfat_disk_status (uint8_t  drive );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
-------	----	-------------------

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

この関数は、コンフィグレーション設定に応じて内部関数を呼び出します。内部関数は、R_tfat_(memory name)_disk_status となります。

3.6 R_tfat_get_fattime

Description

本関数は、日付・時刻の取得を行います。

Usage

```
#include "r_tfat_lib.h"
uint32_t R_tfat_get_fattime (void );
```

Parameters

なし

Return Value

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)

Remark

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。

3.7 R_tfat_drv_change_alloc

Description

本関数は、TFAT モジュールのドライブ番号とメモリドライバのドライブ番号(もしくはチャンネル番号)との関連付けの変更を行います。

Usage

```
#include "r_tfat_lib.h"
#include "r_tfat_driver_rx_if.h"
DRESULT R_tfat_drv_change_alloc (TFAT_DRV_NUM tfat_drv,
                                  uint8_t dev_type,
                                  uint8_t dev_drv_num );
```

Parameters

tfat_drv	入力	TFAT ライブラリで管理するドライブ番号
dev_type	入力	デバイスタイプ(TFAT_CTRL_USB/TFAT_CTRL_SDHI)
dev_drv_num	入力	メモリドライバで管理するドライブ番号(チャンネル番号)

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

この関数は、ドライブ番号の関連付けテーブルを更新します。

関連付けテーブルの初期設定は、コンフィグレーション設定で行います。 変更したい場合は、ユーザ側で本関数を呼び出してください。

本モジュールの API を呼び出している最中に、本関数を呼び出さないでください。

4. 内部関数

USB 向けと SDHI 向けと USB Mini 向けの関数が用意されています。それぞれの関数内でメモリドライバを呼び出します。

4.1 USB 用

「2.6 コンフィグレーションの概要」の TFAT_USB_DRIVE_NUM と TFAT_DRIVE_ALLOC_NUM_i(i=0-9) で TFAT_CTRL_USB が設定されている場合に表 4.1.1 の関数が呼び出されます。

表 4.1.1 関数一覧

関数名	機能概要
R_tfat_usb_disk_initialize	ディスク・ドライブの初期化
R_tfat_usb_disk_read	ディスクからの読み込み
R_tfat_usb_disk_write	ディスクへの書き込み
R_tfat_usb_disk_ioctl	その他のドライブ制御
R_tfat_usb_disk_status	ディスク・ドライブの状態取得

表 4.1.2 その他の関数一覧

関数名	機能概要
R_usb_hmsc_WaitLoop	データのリード/ライド完了待ち

4.1.1 R_tfat_usb_disk_initialize

Description

本関数は、ディスク・ドライブの初期化を行います。

Usage

```
#include "r_tfat_lib.h"
DSTATUS R_tfat_usb_disk_initialize (uint8_t drive);
```

Parameters

drive 入力 初期化するドライブ番号を指定します。

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

USB ドライバの呼び出し制限(起動後、1 度のみ)により、本 API では USB ドライバの初期設定は行っていません。ユーザ側での対応が必要です。

4.1.2 R_tfat_usb_disk_read

Description

本関数は、ディスクからの読み込みを行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_usb_disk_read (    uint8_t    drive ,
                                   uint8_t    *buffer ,
                                   uint32_t    sector_number ,
                                   uint8_t    sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	出力	読み取りデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	読み取るセクタ数を指定します。値は 1～255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

この関数は、ディスク・ドライブからデータを読み取ります。読み取るデータ位置に関する詳細は引数で指定します。

4.1.3 R_tfat_usb_disk_write

Description

本関数は、ディスクへの書き込みを行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_usb_disk_write (    uint8_t    drive ,
                                   uint8_t    *buffer ,
                                   uint32_t    sector_number ,
                                   uint8_t    sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	入力	読み取りデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	読み取るセクタ数を指定します。値は 1～255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

この関数は、ディスク・ドライブにデータを書き込みます。書き込むデータに関する詳細は引数で指定します。

4.1.4 R_tfat_usb_disk_ioctl

Description

本関数は、その他のドライブ制御を行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_usb_disk_ioctl(    uint8_t  drive ,
                                  uint8_t  command ,
                                  void      *buffer );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
command	入力	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	入力	読み取りデータを格納するバッファを指すポインタ。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

R_tfat_usb_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。

4.1.5 R_tfat_usb_disk_status

Description

本関数は、ディスク・ドライブの状態取得を行います。

Usage

```
#include "r_tfat_lib.h"
DSTATUS R_tfat_usb_disk_status(uint8_t  drive );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
-------	----	-------------------

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

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、「2.8 戻り値」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。

4.1.6 R_usb_hmsc_WaitLoop

Description

本関数は、データリード/ライドの完了待ちを行います。

Usage

```
void R_usb_hmsc_WaitLoop (void );
```

Parameters

なし

Return Value

なし

Remark

処理内容の詳細は、USB ドライバ側のドキュメントをご参照ください。

4.2 SDHI 用

「2.6 コンフィグレーションの概要」の TFAT_SDHI_DRIVE_NUM と TFAT_DRIVE_ALLOC_NUM_i(i=0-9) で TFAT_CTRL_SDHI が設定されている場合に表 4.2.1 の関数が呼び出されます。

表 4.2.1 関数一覧

関数名	機能概要
R_tfat_sdhi_disk_initialize	ディスク・ドライブの初期化
R_tfat_sdhi_disk_read	ディスクからの読み込み
R_tfat_sdhi_disk_write	ディスクへの書き込み
R_tfat_sdhi_disk_ioctl	その他のドライブ制御
R_tfat_sdhi_disk_status	ディスク・ドライブの状態取得

[SDHI 使用時の注意]

初期設定、マウント処理、VDD 電源電圧供給処理は、本モジュールでは行いません。SDHI モジュールのドキュメントを参考にユーザ側で対応してください。それらの設定を行わないと本モジュールは正常動作しません。

4.2.1 R_tfata_sdhi_disk_initialize

Description

本関数は、ディスク・ドライブの初期化を行います。

Usage

```
#include "r_tfata_lib.h"  
DSTATUS R_tfata_sdhi_disk_initialize (uint8_t drive);
```

Parameters

drive	入力	初期化するドライブ番号を指定します。
-------	----	--------------------

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

本関数では、SDHI ドライバの初期設定は行っていません。ユーザ側での対応が必要です。

4.2.2 R_tfata_sdhi_disk_read

Description

本関数は、ディスクからの読み込みを行います。

Usage

```
#include "r_tfata_lib.h"  
DRESULT R_tfata_sdhi_disk_read (    uint8_t    drive ,  
                                   uint8_t    *buffer ,  
                                   uint32_t    sector_number ,  
                                   uint8_t    sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	出力	読み取りデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	読み取るセクタ数を指定します。値は 1～255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

SD メモリからデータを読み出します。ブロック毎に実施します。

4.2.3 R_tfat_sdhi_disk_write

Description

本関数は、ディスクへの書き込みを行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_sdhi_disk_write (    uint8_t  drive ,
                                     uint8_t  *buffer ,
                                     uint32_t  sector_number ,
                                     uint8_t  sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	入力	書き込みデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	書き込むセクタ数を指定します。値は 1～255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

SD メモリにデータを書き込みます。ブロック毎に実施します。

4.2.4 R_tfat_sdhi_disk_ioctl

Description

本関数は、その他のドライブ制御を行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_sdhi_disk_ioctl (    uint8_t  drive ,
                                     uint8_t  command ,
                                     void      *buffer );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
command	入力	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	入力	読み取りデータを格納するバッファを指すポインタ。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

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。

本モジュールでは実装されていません。

4.2.5 R_tfat_sdhi_disk_status

Description

本関数は、ディスク・ドライブの状態取得を行います。

Usage

```
#include "r_tfat_lib.h"
DSTATUS R_tfat_sdhi_disk_status (uint8_t drive);
```

Parameters

drive 入力 物理的なドライブ番号を指定します。

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

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、「2.8 戻り値」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。

本モジュールでは実装されていません。

4.3 USB Mini 用

「2.6 コンフィグレーションの概要」のTFAT_USB_MINI_DRIVE_NUMとTFAT_DRIVE_ALLOC_NUM_i(i=0-9)でTFAT_CTRL_USB_MINIが設定されている場合に表4.3.1の関数が呼び出されます。

表 4.3.1 関数一覧

関数名	機能概要
R_tfat_usb_mini_disk_initialize	ディスク・ドライブの初期化
R_tfat_usb_mini_disk_read	ディスクからの読み込み
R_tfat_usb_mini_disk_write	ディスクへの書き込み
R_tfat_usb_mini_disk_ioctl	その他のドライブ制御
R_tfat_usb_mini_disk_status	ディスク・ドライブの状態取得

表 4.3.2 その他の関数一覧

関数名	機能概要
R_usb_mini_hmsc_WaitLoop	データのリード/ライド完了待ち

4.3.1 R_tfata_usb_mini_disk_initialize

Description

本関数は、ディスク・ドライブの初期化を行います。

Usage

```
#include "r_tfata_lib.h"
DSTATUS R_tfata_usb_mini_disk_initialize (uint8_t drive);
```

Parameters

drive	入力	初期化するドライブ番号を指定します。
-------	----	--------------------

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

USB ドライバの呼び出し制限(起動後、1 度のみ)により、本 API では USB ドライバの初期設定は行っていません。ユーザ側での対応が必要です。

4.3.2 R_tfata_usb_mini_disk_read

Description

本関数は、ディスクからの読み込みを行います。

Usage

```
#include "r_tfata_lib.h"
DRESULT R_tfata_usb_disk_read (    uint8_t    drive ,
                                   uint8_t    *buffer ,
                                   uint32_t    sector_number ,
                                   uint8_t    sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	出力	読み取りデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	読み取るセクタ数を指定します。値は 1~255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

この関数は、ディスク・ドライブからデータを読み取ります。読み取るデータ位置に関する詳細は引数で指定します。

4.3.3 R_tfat_usb_mini_disk_write

Description

本関数は、ディスクへの書き込みを行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_usb_mini_disk_write (    uint8_t    drive ,
                                         uint8_t    *buffer ,
                                         uint32_t    sector_number ,
                                         uint8_t    sector_count );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
buffer	入力	読み取りデータを格納するバッファを指すポインタ。
sector_number	入力	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。
sector_count	入力	読み取るセクタ数を指定します。値は 1～255 の範囲です。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

この関数は、ディスク・ドライブにデータを書き込みます。書き込むデータに関する詳細は引数で指定します。

4.3.4 R_tfat_usb_mini_disk_ioctl

Description

本関数は、その他のドライブ制御を行います。

Usage

```
#include "r_tfat_lib.h"
DRESULT R_tfat_usb_mini_disk_ioctl ( uint8_t  drive ,
                                     uint8_t  command ,
                                     void      *buffer );
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
command	入力	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	入力	読み取りデータを格納するバッファを指すポインタ。

Return Value

DRESULT 「2.8 戻り値」に記載した関数実行の結果

Remark

R_tfat_usb_mini_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。

4.3.5 R_tfat_usb_mini_disk_status

Description

本関数は、ディスク・ドライブの状態取得を行います。

Usage

```
#include "r_tfat_lib.h"
DSTATUS R_tfat_usb_mini_disk_status (uint8_t drive);
```

Parameters

drive	入力	物理的なドライブ番号を指定します。
-------	----	-------------------

Return Value

DSTATUS 「2.8 戻り値」に記載した関数実行後のディスクステータス

Remark

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、「2.8 戻り値」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。

4.3.6 R_usb_mini_hmsc_WaitLoop

Description

本関数は、データリード/ライドの完了待ちを行います。

Usage

```
void R_usb_mini_hmsc_WaitLoop(void);
```

Parameters

なし

Return Value

なし

Remark

処理内容の詳細は、USB ドライバ側のドキュメントをご参照ください。

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/contact/>

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.03	2016.04.01	-	サポート(RX ファミリ)追加
1.02	2015.06.30	-	サポートマイコン(RX231)追加
1.01	2015.01.05	-	サポートマイコン追加
1.00	2014.12.01	-	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子

（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

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にご使用ください。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、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。



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 エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■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/>