

# RX ファミリ

R20AN0335JJ0104

Rev.1.04

## M3S-TFAT-Tiny メモリドライバインタフェースモジュール

2018.06.29

### 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、

<https://www.renesas.com/ja-jp/solutions/rx-applications/fit.html>

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 SD メモリカード、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)	<a href="http://www.renesas.com/mw/tfat">http://www.renesas.com/mw/tfat</a>
USB ドライバ (※2)	USB Basic Host and Peripheral Driver USB Host Mass Storage Class Driver	<a href="http://www.renesas.com/driver/usb">http://www.renesas.com/driver/usb</a>
SD メモリカードドライバ(※2)	SD メモリカードドライバ	<a href="https://www.renesas.com/driver/rtm0rx0000dsdd">https://www.renesas.com/driver/rtm0rx0000dsdd</a>
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	<a href="http://www.renesas.com/driver/usb">http://www.renesas.com/driver/usb</a>

※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

## 関連ドキュメント

- Firmware Integration Technology ユーザーズマニュアル(R01AN1833)
- ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)
- e<sup>2</sup> studio に組み込む方法 Firmware Integration Technology (R01AN1723)
- CS+に組み込む方法 Firmware Integration Technology (R01AN1826)
- Renesas e<sup>2</sup> studio スマート・コンフィグレータ ユーザーガイド(R20AN0451)
- RX ファミリ オープンソース FAT ファイルシステム M3S-TFAT-Tiny モジュール  
Firmware Integration Technology (R20AN0038)
- RX ファミリ SD モード SD メモリカードドライバ Firmware Integration Technology(R01AN4233)
- Renesas USB MCU USB Host Mass Storage Class Driver (HMSC) Firmware Integration Technology(R01AN2029)
- Renesas USB MCU USB Host Mass Storage Class Driver for USB mini  
Firmware Integration Technology(R01AN2169)
- RX ファミリ システムタイマモジュール Firmware Integration Technology(R20AN0431)

## 目次

1. 概要.....	5
1.1 本アプリケーションノートについて.....	5
1.2 アプリケーションの概要.....	5
1.2.1 アプリケーション構成.....	5
1.2.2 ソフトウェア構成.....	6
1.3 API の概要.....	8
2. API 情報.....	9
2.1 ハードウェア要件.....	9
2.2 ソフトウェア要件.....	9
2.3 サポートされているツールチェーン.....	9
2.4 ヘッダファイル.....	9
2.5 整数型.....	10
2.6 コンパイル時の設定.....	10
2.7 コードサイズ.....	11
2.8 引数.....	12
2.9 戻り値.....	12
2.10 FIT モジュールの追加方法.....	13
3. API 関数.....	14
3.1 R_tfat_disk_initialize.....	14
3.2 R_tfat_disk_read.....	15
3.3 R_tfat_disk_write.....	15
3.4 R_tfat_disk_ioctl.....	16
3.5 R_tfat_disk_status.....	16
3.6 R_tfat_get_fatime.....	17
3.7 R_tfat_drv_change_alloc.....	18
4. 内部関数.....	19
4.1 USB 用.....	19
4.1.1 R_tfat_usb_disk_initialize.....	20
4.1.2 R_tfat_usb_disk_read.....	20
4.1.3 R_tfat_usb_disk_write.....	21
4.1.4 R_tfat_usb_disk_ioctl.....	22
4.1.5 R_tfat_usb_disk_status.....	22
4.1.6 R_usb_hmsc_WaitLoop.....	23
4.2 SD メモリカード用.....	24
4.2.1 R_tfat_sdmem_disk_initialize.....	24
4.2.2 R_tfat_sdmem_disk_read.....	25
4.2.3 R_tfat_sdmem_disk_write.....	25
4.2.4 R_tfat_sdmem_disk_ioctl.....	26
4.2.5 R_tfat_sdmem_disk_status.....	26
4.3 USB Mini 用.....	27
4.3.1 R_tfat_usb_mini_disk_initialize.....	27

---

4.3.2	R_tfat_usb_mini_disk_read .....	28
4.3.3	R_tfat_usb_mini_disk_write .....	28
4.3.4	R_tfat_usb_mini_disk_ioctl .....	29
4.3.5	R_tfat_usb_mini_disk_status .....	29
4.3.6	R_usb_mini_hmsc_WaitLoop .....	30
5.	付録 .....	31
5.1	動作確認環境 .....	31
5.2	トラブルシューティング .....	32
6.	参考ドキュメント .....	33

## 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、SD メモリカード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、2.6 章を参照してください。

### 1.2 アプリケーションの概要

#### 1.2.1 アプリケーション構成

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 1.1 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
r20an0335jj0104-rx-tfat.pdf		アプリケーションノート
reference_document		
r01an1723ju0121_rx.pdf		e <sup>2</sup> studio に組み込む方法
r01an1826jj0110_rx.pdf		CS+に組み込む方法
FITModules		
r_tfat_driver_rx_v1.04.xml		FIT プラグイン XML
r_tfat_driver_rx_v1.04_extend.mdf		スマート・コンフィグレータ 設定ファイル
r_tfat_driver_rx_v1.04.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)		
r20an0335ej0104-rx-tfat.pdf		アプリケーションノート(英語版)
日本語版(ja)		
r20an0335jj0104-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

## 1.2.2 ソフトウェア構成

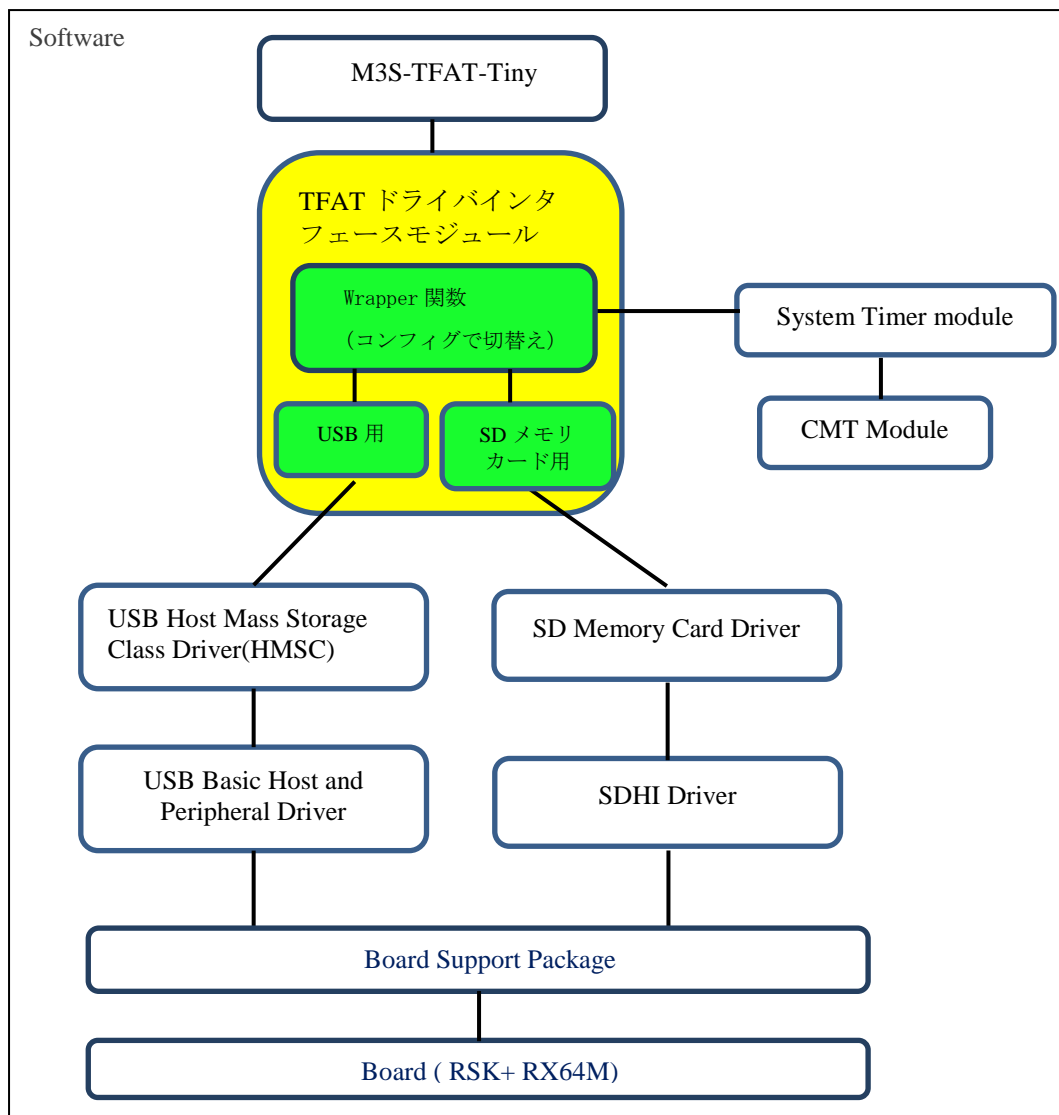


図 1-1 アプリケーション構成図 (RSK RX64M 接続時)

表 1.2 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)	3.60
M3S-TFAT-Tiny	3.03
USB ドライバ	
USB Basic Host and Peripheral Driver	1.23
USB Host Mass Storage Class Driver	1.23
SD メモリカードドライバ	
SD Memory Card Driver	2.02
SDHI	2.01
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
システムタイマモジュール	
System timer module	1.00
CMT module	3.21

### 1.3 API の概要

表 1.3 に本モジュールに含まれる API 関数を示します。

表 1.3 API 関数一覧

関数	関数説明
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出し
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・ドライブ状態取得
R_tfat_get_fatime	日付・時刻の取得
R_tfat_drv_change_alloc	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との関連付け変更



## 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
- CMT

### 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\_usb\_hmssc
- r\_usb\_hmssc\_mini
- r\_sdmemory\_rx
- r\_sys\_time\_rx

使用するメモリドライバの種類は r\_tfat\_driver\_rx\_config.h で設定することができます。

表 1.3.2 使用可能なメモリドライバ

	RX 110	RX 111	RX 113	RX 210	RX 231	RX 23T	RX 63N	RX 64M	RX 71M	RX 65N
r_usb_hmssc	—	—	—	—	—	—	✓	✓	✓	✓
r_usb_hmssc_mini	—	✓	✓	—	✓	—	—	—	—	—
r_sdmemory_rx	—	—	—	—	✓	—	—	✓	✓	✓

SD メモリカードと USB は同時に使用することができます。

### 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.2.08.00

### 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 コンパイル時の設定

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_SDMEM_DRIVE_NUM</code>  - Default value = (0)	SD メモリカード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> <i>i</i> = 0～9  - 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) SD メモリカードで使用するドライブ=(TFAT_CTRL_SDMEM) 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 コードサイズ

本ドライバのコードサイズを下表に示します。

本ドライバのコードサイズを下表に示します。

表 2-1 コードサイズ

ROM、RAM およびスタックのコードサイズ				
デバイス	分類		使用メモリ	備考
RX64M	ROM	SD メモリカード使用時	711 バイト	
		USB 使用時	741 バイト	
		SD メモリカードと USB 使用時	968 バイト	
	RAM	SD メモリカード使用時	26 バイト	
		USB 使用時	26 バイト	
		SD メモリカードと USB 使用時	30 バイト	
	最大使用スタック		360 バイト	
RX231	ROM	SD メモリカード使用時	422 バイト	
		USB mini 使用時	684 バイト	
		SD メモリカードと USB mini 使用時	941 バイト	
	RAM	SD メモリカード使用時	24 バイト	
		USB mini 使用時	26 バイト	
		SD メモリカードと USB mini 使用時	30 バイト	
	最大使用スタック		360 バイト	

---

## 2.8 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;
```

---

## 2.9 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;
```

---

## 2.10 FIT モジュールの追加方法

---

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、Smart Configurator を使用した(1)、(3)の追加方法を推奨しています。ただし、Smart Configurator は、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(2)、(4)の方法を使用してください。

- (1) e<sup>2</sup> studio 上で Smart Configurator を使用して FIT モジュールを追加する場合  
e<sup>2</sup> studio の Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e<sup>2</sup> studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。
- (2) e<sup>2</sup> studio 上で FIT Configurator を使用して FIT モジュールを追加する場合  
e<sup>2</sup> studio の FIT Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e<sup>2</sup> studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。
- (3) CS+上で Smart Configurator を使用して FIT モジュールを追加する場合  
CS+上で、スタンドアロン版 Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e<sup>2</sup> studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。
- (4) CS+上で FIT モジュールを追加する場合  
CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」を参照してください。

### 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	日付・時刻の取得
R_tfat_drv_change_alloc	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

#### 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

TFAT_RES_OK	正常終了
TFAT_RES_OK 以外	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

TFAT_RES_OK	正常終了
TFAT_RES_OK 以外	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。

System timer モジュール R\_SYS\_TIME\_GetCurrentTime()を利用して日付情報を取得します。

本 API では System timer モジュールならびに CMT モジュールの初期設定を行いません。

ユーザ側で対応が必要になります。

---

### 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_SDMEM/ TFAT_CTRL_USB_MINI)
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 SD メモリカード向けと 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<sub>i</sub>(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**

TFAT_RES_OK	正常終了
TFAT_RES_OK 以外	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\_tf\_f\_sync 関数によってのみ使用されます。 R\_tf\_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\_tf\_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\_tf\_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**

TFAT_RES_OK	正常終了
TFAT_RES_OK 以外	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 SD メモリカード用

「2.6 コンフィグレーションの概要」の TFAT\_SDMEM\_DRIVE\_NUM と TFAT\_DRIVE\_ALLOC\_NUM\_i(i=0-9) で TFAT\_CTRL\_SDMEM 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_sdmem_disk_initialize	ディスク・ドライブの初期化
R_tfat_sdmem_disk_read	ディスクからの読み込み
R_tfat_sdmem_disk_write	ディスクへの書き込み
R_tfat_sdmem_disk_ioctl	その他のドライブ制御
R_tfat_sdmem_disk_status	ディスク・ドライブの状態取得

### [SD メモリカード使用時の注意]

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。SD メモリカード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\_tfat\_sdmem\_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_sdmem_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**

TFAT_RES_OK	正常終了
TFAT_RES_OK 以外	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、SD メモリカードドライバの初期設定は行っていないです。ユーザ側での対応が必要です。



## 4.2.2 R\_tfat\_sdmem\_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_sdmem_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 メモリカード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\_sdmem\_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_sdmem_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 メモリカード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\_sdmem\_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_sdmem_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\_sdmem\_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_sdmem_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**

TFAT\_RES\_OK 正常終了

TFAT\_RES\_OK 以外 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/ライト完了待ち

#### 4.3.1 R\_tfat\_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_tfat_lib.h"
DSTATUS R_tfat_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**

TFAT\_RES\_OK                      正常終了  
TFAT\_RES\_OK 以外              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\_tfat\_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_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。

#### 4.3.3 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\_tfata\_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_tfata_lib.h"
DRESULT R_tfata_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\_tfata\_usb\_mini\_disk\_ioctl 関数は、すべての TFAT ライブラリ関数の中で R\_tfata\_f\_sync 関数によってのみ使用されます。 R\_tfata\_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\_tfata\_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\_tfata\_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\_tfata\_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_tfata_lib.h"
DSTATUS R_tfata_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**

TFAT\_RES\_OK 正常終了

TFAT\_RES\_OK 以外 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。

5. 付録

5.1 動作確認環境

本ドライバの動作確認環境を以下に示します。

表 5-1 動作確認環境

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio V6.3.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V2.08.00
	コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのバージョン	Ver.1.04
使用ボード	Renesas Starter Kit for RX64M (型名：R0K50564MSxxxBE)
	Renesas Starter Kit for RX231 (型名：R0K505231SxxxBE)

## 5.2 トラブルシューティング

- (1) Q : 本 FIT モジュールをプロジェクトに追加しましたが、ビルド実行すると「Could not open source file "platform.h"」エラーが発生します。

A : FIT モジュールがプロジェクトに正しく追加されていない可能性があります。プロジェクトへの追加方法をご確認ください。

- CS+を使用している場合

アプリケーションノート RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」

- e<sup>2</sup> studio を使用している場合

アプリケーションノート RX ファミリ e<sup>2</sup> studio に組み込む方法 Firmware Integration Technology (R01AN1723)」

また、本 FIT モジュールを使用する場合、ボードサポートパッケージ FIT モジュール(BSP モジュール)もプロジェクトに追加する必要があります。BSP モジュールの追加方法は、アプリケーションノート「ボードサポートパッケージモジュール(R01AN1685)」を参照してください。



## 6. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：開発環境

RX ファミリ CC-RX コンパイラ ユーザーズマニュアル (R20UT3248)

(最新版をルネサス エレクトロニクスホームページから入手してください。)

## 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>

お問合せ先

<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
1.04	2018.06.29	-	1.2.2 図 1-1 呼び出しモジュールに System timer、CMT 追加 1.3 API 概要 追加 2.6 SD メモリカード定義名変更 2.7 コードサイズ 追加 3.6 R_fat_get_fattime() 説明修正 4.2.1-4.2.5 API 名変更 5 付録 追加 6 参考ドキュメント 追加
1.03	2016.10.01	-	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に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えば、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。型名が異なる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生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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）、交通制御（信号）、大規模通信機器、  
金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。

6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。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. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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直接的、間接的に支配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。

(Rev.4.0-1 2017.11)



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：<https://www.renesas.com/contact/>