

RX ファミリ

FTP/Web サーバ用ファイルドライバモジュール

Firmware Integration Technology

R20AN0333JJ0102

Rev.1.02

2016.10.01

要旨

本アプリケーションノートは、組み込み用 TCP/IP M3S-T4-Tiny を用いた Web サーバ(以降、Web サーバ)や組み込み用 TCP/IP M3S-T4-Tiny を用いた FTP サーバ(以降、FTP サーバ)から呼び出されるファイルシステムインタフェースについての説明を行います。

FIT 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 FIT モジュールを使用するには、下記 FIT モジュールが必要です。

表 1 FIT モジュール一覧

機能	ミドルウェア製品	ウェブページ
Board Support Package	ボードサポートパッケージ(BSP) (R01AN1685)	https://www.renesas.com/ja-jp/software/D3018116.html
TCP/IP (※1)	M3S-T4-Tiny(以降、T4) (R20AN0051)	http://www.renesas.com/mw/t4
FTP サーバ (※2)	組み込み用 TCP/IP M3S-T4-Tiny を用いた FTP サーバ (R20AN0078)	http://www.renesas.com/mw/t4
Web サーバ (※2)	組み込み用 TCP/IP M3S-T4-Tiny を用いた Web サーバ (R20AN0075)	http://www.renesas.com/mw/t4
Ethernet インタ フェース (※1)	Ethernet ドライバと組み込み用 TCP/IP M3S-T4-Tiny のインタフェース変換モ ジュール(以降、T4 インタフェース) (R20AN0311)	http://www.renesas.com/mw/t4
Ethernet ドライバ (※1)	イーサネットモジュール (R01AN2009)	http://www.renesas.com/mw/t4
タイマ (※1)	コンペアマッチタイマ(CMT)モジュール (R01AN1856)	https://www.renesas.com/ja-jp/software/D6000037.html
ファイルシステム (※3)	M3S-TFAT-Tiny (以降、TFAT) TFAT(R20AN0038)	http://www.renesas.com/mw/tfat
メモリドライバ インタフェース(※4)	TFAT メモリドライバインタフェースモ ジュール(R20AN0335)	http://www.renesas.com/mw/tfat
USB ドライバ (※4)	USB Basic Host and Peripheral Driver USB Host Mass Storage Class Driver	http://www.renesas.com/driver/usb

※1 必須です。

※2 どちらか一方は必要です。

※3 外部メモリを使う場合は必須です。

※4 評価される環境に応じてお使いください。

動作確認デバイス

RX ファミリ

目次

1. 概要.....	3
1.1 本アプリケーションノートについて.....	3
1.2 ファイル/フォルダ構成	3
1.3 ソフトウェア構成	4
2. API 情報	5
3. API 関数	8
3.1 データ構造体	9
3.2 change_dir.....	9
3.3 file_close	10
3.4 file_delete	10
3.5 file_open	11
3.6 file_read.....	11
3.7 file_rename.....	12
3.8 file_exist	12
3.9 file_write	13
3.10 get_file_info	13
3.11 get_file_list_info	14
3.12 get_file_size	15
3.13 make_dir.....	15
3.14 remove_dir	16
3.15 R_server_file_drv_drive_regist	16

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は、Web サーバや FTP サーバから呼び出されるファイルシステムインタフェースについて説明します。

ファイルシステムインタフェースは、使用するファイルシステムなどに応じてカスタム対応するユーザ定義関数として、サンプルプログラムが提供されてきました。本アプリケーションノートは、それらを集約して FIT 対応したものです。

本アプリケーションノートで提供されるインタフェースモジュールは、外部メモリを使用するパターンと内部メモリを使用するパターンの 2 つが用意されており、コンフィグレーション設定で切り替え可能です。

- 外部メモリを使用するパターン : ファイルシステムとして TFAT を使用。
- 内部メモリを使用するパターン : ファイルシステムは使用せず。カレントディレクトリ上でのファイル制御のみ対応。
内蔵 RAM などでの使用を想定しての実装。
ファイルリストは起動時に初期化。

本モジュールで提供する API は、Web サーバか FTP サーバから呼び出されますので、ユーザ側が呼び出す必要はありません。

外部メモリを使用する場合に、本モジュールから TFAT の各 API を呼び出しますが、初期設定関数は呼び出していません。使用するメモリドライバ仕様に依拠してユーザ側で対応が必要です。

1.2 ファイル/フォルダ構成

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 2 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
r20an0333jj0102-rx-t4.pdf		アプリケーションノート
reference_documents		
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_t4_file_driver_v1.02.xml		FIT プラグイン XML
r_t4_file_driver_v1.02.zip		FIT プラグイン ZIP
コンフィグレーション (r_config)		
r_t4_file_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_t4_file_driver_rx)		
ドキュメント(doc)		
en		
r20an0333ej0102-rx-t4.pdf		アプリケーションノート(英語)
ja		
r20an0333jj0102-rx-t4.pdf		アプリケーションノート(日本語)
コンフィグレーションリファレンス(ref)		
r_t4_file_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

1.3 ソフトウェア構成

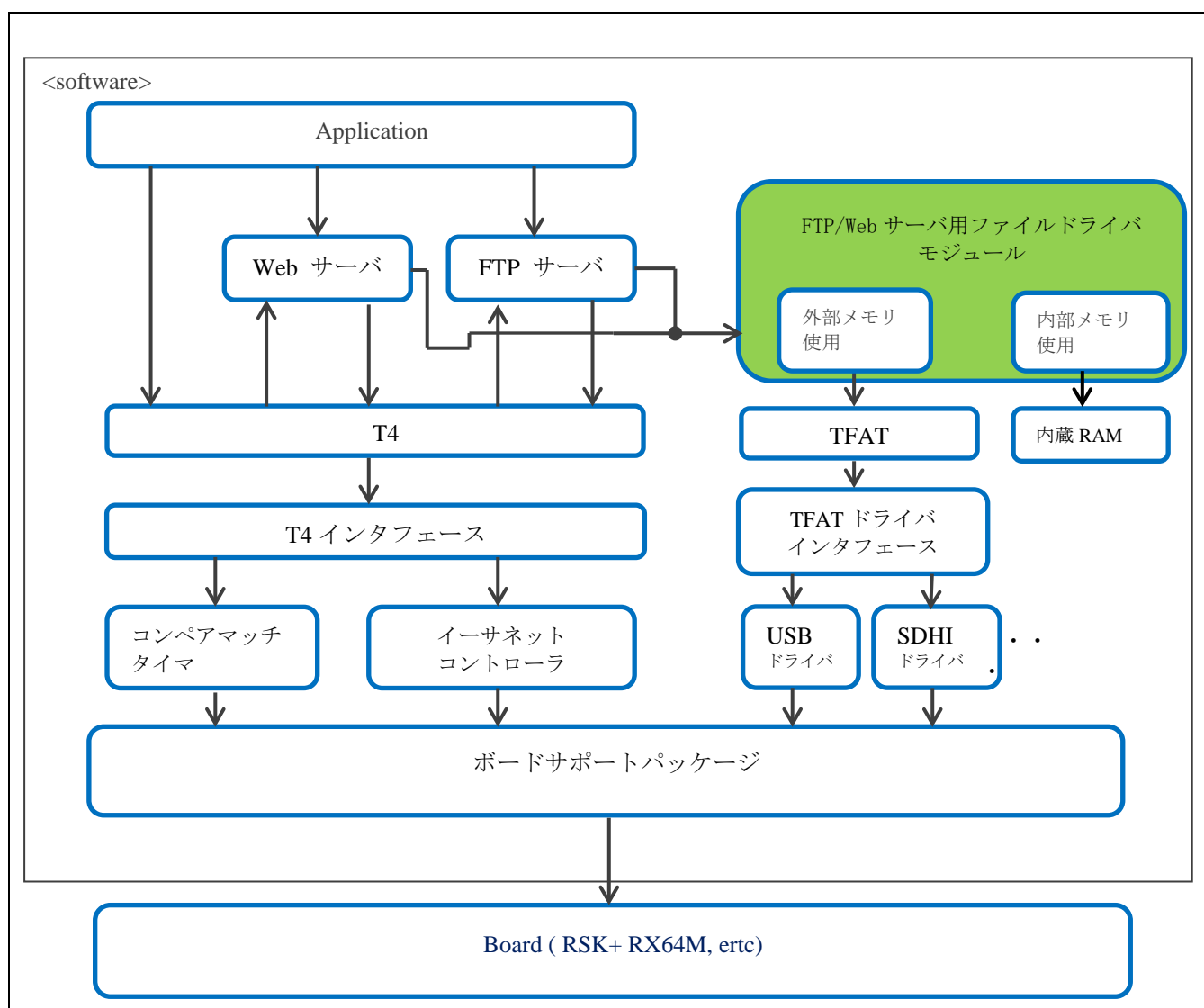


図 1 ソフトウェア構成

表 3 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)	3.40
M3S-T4-Tiny(T4)	2.05
組み込み用 TCP/IP M3S-T4-Tiny を用いた FTP サーバ	1.04
組み込み用 TCP/IP M3S-T4-Tiny を用いた Web サーバ	1.05
Ethernet ドライバと組み込み用 TCP/IP M3S-T4-Tiny のインタフェース変換モジュール(T4 インタフェース)	1.05
イーサネットコントローラ(ETHERC)	1.11
コンペアマッチタイマ(CMT)	3.00
M3S-TFAT-Tiny(TFAT)	3.03
TFAT ドライバインタフェースモジュール	1.03
USB ドライバ	
USB Basic Host and Peripheral Driver	1.11
USB Host Mass Storage Class Driver	1.11

2. API 情報

2.1 ハードウェア要件

なし

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_t4_rx
- r_tfat_rx
- r_t4_http_server_rx
- r_t4_ftp_server_rx

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_t4_file_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_t4_file_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_t4_file_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。

表 4 コンフィグレーション

Configuration options in <code>r_t4_file_driver_rx_config.h</code>	
#define SERVER_FILE_DRV_USE_EXTERNAL - Default value = (0)	データの保存に外部メモリを使用する場合に 1 を設定します。
#define SERVER_FILE_DRV_USE_INTERNAL - Default value = (0)	データの格納に内部メモリを使用する場合に 1 を設定します。 SERVER_FILE_DRV_USE_EXTERNAL も設定されている場合は、そちらが優先されます。
#define SERVER_MAX_OPEN_FILE - Default value = (10)	ファイルオープンの状態を監視する数を設定します。 Web ブラウザなどで一度に開けるファイルの上限になります。
#define SERVER_DEF_CONTENTS - Default value = (100)	内部メモリ使用時のみ有効です。 初期設定時のファイルリストに入るファイル数を設定します。
#define SERVER_CONTENTS_LIST ¥ -Default value (none)	内部メモリ使用時のみ有効です。 初期設定時のファイルリストを記載します。
#define SERVER_MAX_CONTENTS_LIST - Default value = (0)	内部メモリ使用時のみ有効です。 ファイルリストの上限を設定します。
#define SERVER_CONTENTS_STORE_HEAD_ADR - Default value = (NULL)	内部メモリ使用時のみ有効です。 書き込み要求がきたデータを格納する領域の先頭アドレスを設定します。領域の確保はユーザ側で行ってください。
#define SERVER_CONTENTS_STORE_MEM_SIZE - Default value = (0)	内部メモリ使用時のみ有効です。 書き込み要求がきたデータデータを格納する領域サイズを設定します。 本設定と SERVER_MAX_CONTENTS_LIST の組み合わせで書き込み可能な上限が決まります。

2.7 引数

```
/* DATE info.*/
typedef struct date_info_
{
    uint16_t year;                // 2011, 2012, ...
    uint8_t month[4];             // Jan, Feb, Mar, ...
    uint8_t day;                  // 1-31
    uint8_t day_of_the_week[4];   // Sun, Mon, Tus, ...
    uint16_t hour;                // 0-23
    uint16_t min;                 // 0-59
    uint16_t sec;                 // 0-59
}DATE_INFO;

/* file list info.*/
typedef struct file_list_
{
    uint8_t file_name[13];        // file name
    uint32_t file_size;           // file size
    uint32_t file_attr;           // file attribute
    DATE_INFO date_info;          // date info.
}FILE_LIST;

/* contents info. */
typedef struct _contents
{
    uint8_t file_name[256];       // file name
    uint8_t *file_address;        // file data pointer
    int32_t file_size;            // file size
    uint32_t attr;                // date info
}CONTENTS;
```

2.8 戻り値

なし

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

FTP サーバや Web サーバから本関数群が呼び出されます。ここに記載された関数は、コンフィグレーション設定に応じて、下位層の関数を呼び分けます。主となる処理は内部関数側になります。

表 5 関数一覧

関数名	機能概要
change_dir()	作業ディレクトリの変更
file_close()	ファイルのクローズ
file_delete()	ファイルの削除
file_open()	ファイルのオープン
file_read()	ファイルの読み込み
file_rename()	ファイル名の変更
file_exist()	ファイルの有無を確認
file_write()	ファイルの書き込み
get_file_info()	ファイル情報の取得
get_file_list_info()	ファイルリストの取得
get_file_size()	ファイルサイズの取得
make_dir()	ディレクトリの作成
remove_dir()	ディレクトリの削除
R_server_file_drv_drive_regist	ドライブ番号の切り替え

3.1 データ構造体

【マクロ定義】

```
#define FILE_WRITE (0x10)
#define FILE_READ (0x01)

/* File attribute bits for FILE_LIST->file_attr */
#define FILE_ATTR_RDO 0x01 /* Read only */
#define FILE_ATTR_HID 0x02 /* Hidden */
#define FILE_ATTR_SYS 0x04 /* System */
#define FILE_ATTR_VOL 0x08 /* Volume label */
#define FILE_ATTR_DIR 0x10 /* Directory */
#define FILE_ATTR_ARC 0x20 /* Archive */
```

3.2 change_dir

Description

本関数は引数で指定されたディレクトリパスを作業ディレクトリに設定します。ディレクトリパスはフルパスで指定します。作業ディレクトリの情報は、通信端点毎に管理されます。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t change_dir(uint8_t *dir_path);
```

Parameters

dir_path	入力	指定されたディレクトリパスの格納先
----------	----	-------------------

Return Value

-1	ディレクトリが存在しない
0	正常終了

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、change_dir_sub()となります。

3.3 file_close

Description

本関数は引数で指定された ID 値に対応するファイルをクローズし、管理情報を破棄します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_close(int32_t file_id);
```

Parameters

file_id	入力	クローズするファイルの ID 値
---------	----	------------------

Return Value

-1	エラー
0	正常終了

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、file_close_sub()となります。

3.4 file_delete

Description

本関数は引数で指定されたファイルを削除します。ファイルの指定はルートディレクトリからのフルパスで指定します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_delete(uint8_t *file_path);
```

Parameters

file_path	入力	ファイルのフルパスの格納先
-----------	----	---------------

Return Value

-1	エラー
0	正常終了

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、file_delete_sub()となります。

3.5 file_open

Description

本関数は第 1 引数で指定されたファイルを第 2 引数で指定されたモードでオープンし、管理情報を独自で保存します。また、保存した管理情報を FTP サーバが ID 参照できるように、戻り値として管理情報の ID 値を指定します。保存した管理情報はファイルのクローズ関数で ID 値が指定されるまで保持しなければなりません。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_open(uint8_t *file_path, uint8_t mode_flag);
```

Parameters

file_path	入力	ファイルのフルパスの格納先
mode_flag	入力	ファイルオープンモード (FILE_WRITE または FILE_READ)

Return Value

-1	エラー
0 以上	オープンしたファイルの ID 値

Remark

ファイルオープン状態はファイルのクローズ関数で対応する ID 値が指定されるまで保持しなければなりません。

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、file_open_sub()となります。

3.6 file_read

Description

本関数は第 1 引数で指定された ID 値に対応するファイルデータを、第 2 引数が示すアドレスに、最大第 3 引数の値で示すサイズ分読み込まれます。第 1 引数の ID 値に対応する管理情報内のファイルポインタは読み込んだ分だけ更新され、ファイルのクローズ関数が呼び出されるまで保持します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_read(int32_t file_id, uint8_t *buf, int32_t read_size);
```

Parameters

file_id	入力	読み込むファイルの ID 値
buf	出力	読み込んだファイルデータの格納先
read_size	入力	読み込むファイルサイズ

Return Value

-1	エラー
0 以上	読み込んだデータサイズ

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、file_read_sub()となります。

3.7 file_rename

Description

本関数は第 1 引数で指定されたファイルまたはディレクトリを第 2 引数で指定された名前に変更します。第 1 引数、第 2 引数ともにルートディレクトリからのフルパスで指定します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_rename(uint8_t *old_name, uint8_t *new_name);
```

Parameters

old_name	入力	変更対象のファイルまたはディレクトリ
new_name	入力	変更後の名前

Return Value

-1	エラー
0	正常終了

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、file_rename_sub()となります。

3.8 file_exist

Description

本関数は引数で指定されたファイルまたはディレクトリの有無を確認します。引数はルートディレクトリからのフルパスで指定します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_exist(uint8_t *file_path);
```

Parameters

file_path	入力	有無を確認するファイルまたはディレクトリ
-----------	----	----------------------

Return Value

-1	存在しない
0	存在する

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、file_exist_sub()となります。

3.9 file_write

Description

本関数は第 1 引数で指定された ID 値に対応するファイルに対し、第 2 引数で指定されたアドレスから第 3 引数で指定されたサイズ分のデータを書き込みます。第 1 引数の ID 値に対応する管理情報内のファイルポインタは書き込んだ分だけ更新され、ファイルのクローズ関数が呼び出されるまで保持します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t file_write(int32_t file_id, uint8_t *buf, int32_t write_size);
```

Parameters

file_id	入力	書き込むファイルの ID 値
buf	入力	書き込むデータの先頭アドレス
write_size	入力	書き込むサイズ

Return Value

-1	エラー
0	正常終了

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、file_write_sub()となります。

3.10 get_file_info

Description

本関数は第 1 引数で指定された ID 値に対応するファイルの管理情報を読み込み、ファイルの日付情報を第 2 引数で示す日付情報構造体へ書き出します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t get_file_info(int32_t file_id, DATE_INFO *date_info);
```

Parameters

file_id	入力	読み込むファイルの ID 値
date_info	出力	日付情報の格納先

Return Value

-1	エラー
0	正常終了

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、get_file_info_sub()となります。

3.11 get_file_list_info

Description

本関数は第 1 引数で指定されたディレクトリパスに格納されているファイルまたはディレクトリの情報を第 2 引数で指定されたファイルリスト構造体へ書き出します。一度に書き出す最大情報個数は第 3 引数で指定し、第 4 引数でファイルリストの読み出し開始位置を指定します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t get_file_list_info(uint8_t *dir_path, FILE_LIST *file_list, uint32_t num_file_list, int32_t read_index);
```

Parameters

dir_path	入力	読み出すディレクトリパスの格納先
file_list	出力	読み出したファイルリストの格納先 リストの最後にはファイル名格納領域の先頭に'¥0'を格納します。
num_file_list	入力	一度に読み出すファイルリスト情報の最大個数
read_index	入力	ファイルリストの読み出し開始位置

Return Value

-1	エラー
0 以上	読み出したファイルの個数

Remark

戻り値が num_file_list より小さい値を返した場合は、ファイルリスト情報の読み出しが終了したことを示し、num_file_list と同じ値を返した場合はファイルリスト情報に続きがあることを示します。本関数はファイルリストの続きを読み出す際に、read_index にファイルリストの読み出し開始位置を指定して呼び出します。

dir_path の終端には'/'が付く場合と付かない場合があります。使用するファイルシステムにあわせて'/'の有無を調整してください。

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、get_file_info_sub()となります。

3.12 get_file_size

Description

本関数は引数で指定された ID 値に対応するファイルの管理情報を読み込み、ファイルサイズを返します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t get_file_size(int32_t file_id);
```

Parameters

file_id	入力	読み込むファイルの ID 値
---------	----	----------------

Return Value

-1	エラー
0 以上	ファイルサイズ

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、get_file_size_sub()となります。

3.13 make_dir

Description

本関数は引数で指定されたディレクトリを作成します。ディレクトリパスはフルパスで指定します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t make_dir(uint8_t *dir_path);
```

Parameters

dir_path	入力	作成するディレクトリ名
----------	----	-------------

Return Value

-1	エラー
0	正常終了

Remark

dir_path の終端には'/'が付く場合と付かない場合があります。使用するファイルシステムにあわせて'/'の有無を調整してください。

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、make_dir_sub()となります。

3.14 remove_dir

Description

本関数は引数で指定されたディレクトリを削除します。ディレクトリパスはフルパスで指定します。

Usage

```
#include <stdint.h>
#include "r_t4_file_driver_rx_if.h"
int32_t remove_dir(uint8_t *dir_path);
```

Parameters

dir_path	入力	削除するディレクトリ名
----------	----	-------------

Return Value

-1	エラー
0	正常終了

Remark

dir_path の終端には'/'が付く場合と付かない場合があります。使用するファイルシステムにあわせて'/'の有無を調整してください。

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、remove_dir_sub()となります。

3.15 R_server_file_drv_drive_regist

Description

本関数は引数で指定されたドライブ番号でファイルシステムの制御を行います。

Usage

```
#include <stdint.h>
int32_t R_server_file_drv_drive_regist(uint8_t drive);
```

Parameters

drive	入力	ドライブ番号
-------	----	--------

Return Value

-1	エラー
0	正常終了

Remark

ドライブの初期値は「0」です。変更が必要な場合は、ユーザ側で呼び出してください。

変更ドライブに対する TFAT ライブラリ初期設定はユーザ側で行ってください。

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。

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_server_file_drv_drive_regist_sub ()となります。

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://www.renesas.com/>

お問合せ先

<http://www.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
1.02	2016.10.01	-	FIT 用 xml ファイルを更新しました。
1.01	2015.01.05	-	サポートマイコンを追加しました。
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くと、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、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情報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/>