RL78ファミリ

オープンソースFATファイルシステム M3S-TFAT-Tinyへの SPIモード MMC/SDカード ドライバ・ソフトウェアの組み込み例

要旨

本アプリケーションノートでは、オープンソース FAT ファイルシステム M3S-TFAT-Tiny（以下、TFAT ライブラリと呼ぶ）と SPI モードマルチメディアカードドライバ・ソフトウェア（以下、MMC ドライバと呼ぶ）を組み合わせて使用する場合の組み込み方法について説明します。

対象デバイス

対象 MCU

「RL78 ファミリ　オープンソース FAT ファイルシステム M3S-TFAT-Tiny」と「RL78 ファミリ　SPI モードマルチメディアカードドライバ」が共に動作する MCU

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様

記憶メディアのファイル操作を可能にするための TFAT ライブラリに含まれるメモリドライバインタフェース関数のサンプルコードです。

以下に、機能概略を示します。

- MMC ドライバを使用する場合のメモリドライバインタフェース関数を提供し、ファイル操作を可能にします。
- TFAT ライブラリの R_tfat_f_sync()（ディスクキャッシュ制御）は、未サポートです。

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1　動作確認条件

項目	内容
使用マイコン	RL78/G23（プログラム ROM 128KB / RAM 16KB）
動作周波数	メイン・システム・クロック：32MHz 周辺ハードウェア・クロック：32MHz
動作電圧	3.3V
統合開発環境	ルネサス エレクトロニクス製 CS+ for CC V8.05.00
C コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.10.00
使用ボード	RL78G23-64p Fast Prototyping Board (RTK7RLG230CLG000BJ)
使用デバイス	SanDisk microSD 2GB

3. 関連アプリケーションノート／ユーザーズマニュアル

本アプリケーションノートに関連するアプリケーションノート／ユーザーズマニュアルを以下に示します。併せて参照してください。

- FAT ファイルシステム（M3S-TFAT-Tiny）（R20UW0078JJ）
- RL78 ファミリ　オープンソース FAT ファイルシステム M3S-TFAT-Tiny：導入ガイド（R20AN0159JJ）
- RL78 ファミリ　SPI モードマルチメディアカードドライバ： 導入ガイド（R02AN0158JJ）

4. ソフトウェア説明

4.1 動作概要

TFAT ライブラリに含まれるメモリドライバインタフェース関数を MMC ドライバ用に修正し、記憶メディアのファイル操作を可能にします。

図4.1にTFATライブラリとMMCドライバの構成を示します。

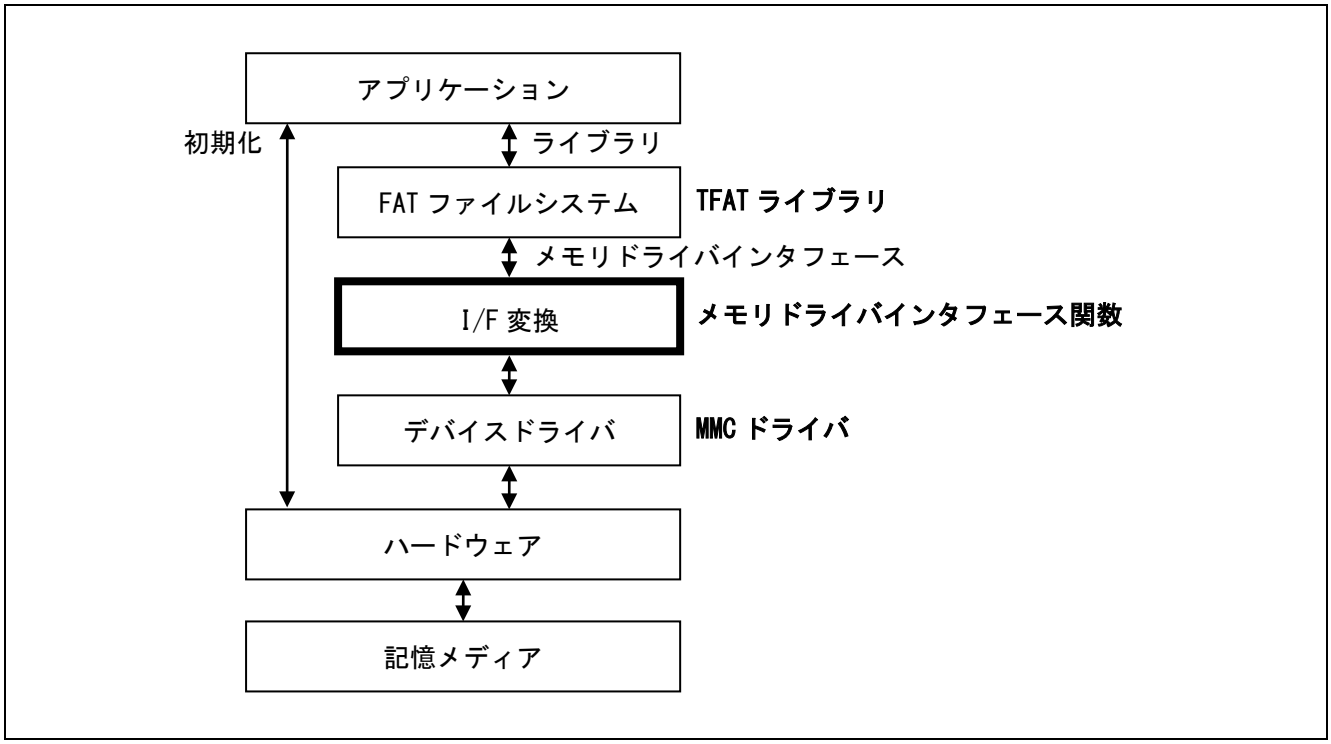

図4.1　TFAT ライブラリと MMC ドライバの構成

4.2　必要メモリサイズ

表4.1に必要メモリサイズを示します。

表4.1　必要メモリサイズ

使用メモリ	サイズ	備考
ROM	454 バイト	drv_if_sub.c r_tfat_drv_if.c r_target_io.c
RAM	20 バイト	drv_if_sub.c r_tfat_drv_if.c r_target_io.c
最大使用ユーザスタック	82 バイト	
最大使用割り込みスタック	-	割り込み未使用

【注１】必要メモリサイズは C コンパイラのバージョンやコンパイルオプションにより異なります。
【注２】スタックサイズは、MMC ドライバのスタック RAM サイズを含みます。

4.3　ファイル構成

表4.2にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表4.2　サンプルコードで使用するファイル

¥an_r01an1643jj0200_rl78_file_system		<DIR>	サンプルコードのフォルダ
	r01an1643jj0200_rl78.pdf		アプリケーションノート
	¥source	<DIR>	プログラム格納用フォルダ
		¥drv_if <DIR>	メモリドライバインタフェース用フォルダ
		drv_if_sub.c	メモリドライバインタフェース内部関数ファイル
		r_tfat_drv_if.c	メモリドライバインタフェース関数ファイル
		¥mcu_io <DIR>	ボードセットアップインタフェース用フォルダ
		r_target_io.c	ボードセットアップインタフェースソースファイル

4.4　関数一覧

4.4.1　メモリドライバインタフェース関数

表4.3にメモリドライバインタフェース関数を示します。

表4.3　メモリドライバインタフェース関数

関数名	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_outstream 関数（TFAT ライブラリのユーザーズマニュアルを参照）は、R_tfat_f_forward 関数を使用する場合に必要です。必要に応じて作成してください。

4.4.2　メモリドライバインタフェース内部関数

表4.4にメモリドライバインタフェース内部関数を示します。

表4.4　メモリドライバインタフェース内部関数

関数名	概要
R_card_insertion_chk()	カードの挿入チェック処理
R_init_card_detect_chat()	チャタリングカウンタ初期化

4.4.3　ボードセットアップインタフェース関数

表4.5にボードセットアップインタフェース関数を示します。

表4.5　ボードセットアップインタフェース関数

関数名	概要
storage_driver_init()	MMC ドライバの初期化

4.5 関数仕様

TFAT ライブラリのユーザーズマニュアルを参照してください。

4.5.1 メモリドライバインタフェース関数の修正

(1) インクルードファイルの修正

MMC ドライバのヘッダファイルと共通関数のヘッダファイルを設定してください。

■r_tfat_drv_if.c　36 行目付近

```
/*****************************************************************************
Includes <System Includes> , "Project Includes"
*****************************************************************************/
#include "r_cg_macrodriver.h"
#include "r_stdint.h"
#include "r_tfat_lib.h"
#include "r_mtl_com.h"
#include "r_mmc.h"
```

(2) R_tfat_disk_initialize()の修正

本関数は、MMC 検出処理（内部で R_mmc_Chk_Detect()関数をコール）とデバイス初期化設定（R_mmc_Init_Slot()関数をコール）を実行するものです。

したがって、TFAT ライブラリから MMC ドライバ API がコールされる前に、MMC ドライバの初期化処理(R_mmc_Init_Driver())を事前に行ってください。

MMC ドライバの初期化方法は、MMC ドライバのユーザーズマニュアルを参照してください。

■r_tfat_drv_if.c　77 行目付近

```
DSTATUS R_tfat_disk_initialize(void)
{
/* Please put the code for disk_initalize driver interface function over here */
/* Please refer the application note for details                                */
/* Please put the code for memory driver initialization, if any, over here    */
  int16_t ret_value;

  ret_value = R_card_insertion_chk(MMC_SLOT0);
  if (ret_value < MMC_OK)
  {
   R_init_card_detect_chat();
   return TFAT_STA_NODISK;
  }

  if (ret_value != MMC_TRUE)
  {
   return TFAT_STA_NODISK;
  }

  ret_value = R_mmc_Init_Slot(MMC_SLOT0);
  if (ret_value < MMC_OK)
  {
   R_init_card_detect_chat();
   return TFAT_STA_NOINIT;
  }

  ret_value = R_mmc_Get_MmcInfo(MMC_SLOT0, &card_info);
  if (ret_value < MMC_OK)
  {
   R_init_card_detect_chat();
   return TFAT_STA_NOINIT;
  }
  return  TFAT_RES_OK;
}
```

(3) R_tfat_disk_read()の修正

MMC ドライバの読み出し関数をコールします。

■r_tfat_drv_if.c　124 行目付近

```
DRESULT R_tfat_disk_read(
    uint8_t Drive,          /* Physical drive number          */
    uint8_t* Buffer,        /* Pointer to the read data buffer   */
    uint32_t SectorNumber, /* Start sector number           */
    uint8_t SectorCount     /* Number of sectors to read      */
)
{
  /*Please put the code for R_tfat_disk_read driver interface function over her *
/
  /*Please refer the application note for details                          */
  int16_t    Ret;

  Ret = R_mmc_Read_Data(0,SectorNumber,SectorCount,Buffer,MMC_MODE_NORMAL);
  if (Ret < MMC_OK)
  {
    return TFAT_RES_ERROR;
  }
  return TFAT_RES_OK;
}
```

(4) R_tfat_disk_write()の修正

MMC ドライバの書き込み関数をコールします。

■r_tfat_drv_if.c　154 行目付近

```
DRESULT R_tfat_disk_write(
    uint8_t Drive,          /* Physical drive number          */
    const uint8_t* Buffer,  /* Pointer to the write data       */
    uint32_t SectorNumber,  /* Sector number to write         */
    uint8_t SectorCount     /* Number of sectors to write      */
)
{
  /*Please put the code for R_tfat_disk_write driver interface function over her
e*/
  /*Please refer the application note for details                          */
  int16_t Ret;

  Ret = R_mmc_Write_Data(0,SectorNumber,SectorCount, (uint8_t *)Buffer,MMC_MODE
_NORMAL);
  if (Ret < MMC_OK)
  {
    return TFAT_RES_ERROR;
  }
  return TFAT_RES_OK;
}
```

(5) R_tfat_disk_ioctl()の修正

ディスクキャッシュ機能を持たせる場合に、適切な処理を記述してください。

詳細は、TFAT ライブラリのユーザーズマニュアルを参照してください。

以下は、何もせずに正常終了させる場合の処理例です。したがって、R_tfat_f_sync()を使用しないでください。

■r_tfat_drv_if.c　183 行目付近

```
DRESULT R_tfat_disk_ioctl(
    uint8_t Drive,          /* Drive number          */
    uint8_t Command,        /* Control command code    */
    void* Buffer           /* Data transfer buffer    */
)
{
  /*Please put the code for R_tfat_disk_ioctl driver interface function over her
e */
  /*Please refer the application note for details                          */
  return TFAT_RES_OK;
}
```

(6) R_tfat_disk_status ()の修正

ディスク・ドライブの状態を取得させる場合に、適切な処理を記述してください。

詳細は、TFAT ライブラリのユーザーズマニュアルを参照してください。

■r_tfat_drv_if.c　202 行目付近

```
DSTATUS R_tfat_disk_status(
    uint8_t Drive          /* Physical drive number   */
)
{
  /*Please put the code for R_tfat_disk_status driver interface function over he
re */
  /*Please refer the application note for details                          */
  return TFAT_RES_OK;
}
```

(7) R_tfat_get_fattime ()の修正

MMC ドライバが関連しない処理のため、説明を省略します。

4.5.2　メモリドライバインタフェース内部関数の修正

(1)　インクルードファイルの修正

MMC ドライバのヘッダファイルと共通関数のヘッダファイルを設定してください。

■drv_if_sub.c　1 行目付近

```
#include "r_cg_macrodriver.h"
#include <string.h>
#include "r_stdint.h"
#include "r_tfat_lib.h"
#include "r_data_file.h"
#include "r_board.h"
#include "r_mtl_com.h"
#include "r_mmc.h"
```

(2)　static 変数定義の修正

static 定義されている配列のインデックスの定義を修正してください。

■drv_if_sub.c　16 行目付近

```
static uint16_t          gDetChatCnt[MMC_SLOT_NUM];
static uint16_t          gDetSts_Old[MMC_SLOT_NUM];
```

(3) R_card_insertion_chk()の修正

MMC ドライバの挿入チェック関数をコールします。

■drv_if_sub.c　27 行目付近

```
int16_t R_card_insertion_chk(uint8_t slot_num)
{
  uint8_t          DetSts;              /* Detection status                    */
  int16_t          api_ret;

  /* Check MMC insertion. */
  api_ret = R_mmc_Chk_Detect(slot_num,&DetSts);
  if (api_ret < MMC_OK)
  {
    return api_ret;
  }
  if (DetSts != gDetSts_Old[slot_num])      /* Status Changed!                 */
  {
    if (DetSts == MMC_TRUE)                 /* Removal -> Insertion            */
    {
       DetChatCnt[slot_num]--;              /* Chattering counter decrement    */
       if (gDetChatCnt[slot_num] == 0)      /* counter =0                      */
       {
          gDetChatCnt[slot_num] = MMC_INS_CHAT;
          gDetSts_Old[slot_num] = DetSts;   /* Initialize MMC slot.            */
       }
    }
    else
    {
       /* Insertion -> Removal     */
       /* Do not care chattering. */
       gDetChatCnt[slot_num] = MMC_INS_CHAT;
       gDetSts_Old[slot_num] = DetSts;
    }
  }
  else
  {
    /* No change    */
    gDetChatCnt[slot_num] = MMC_INS_CHAT;
  }
  return gDetSts_Old[slot_num];
}
```

(4) R_init_card_detect_chat()の修正

対象の MMC のチャタリングカウンタの初期化を行います。

■drv_if_sub.c　73 行目付近

```
void R_init_card_detect_chat(void)   /* Initial process                          */
{
  uint8_t          slot_num;                /* Slot number                        */

  for (slot_num = MMC_SLOT0; slot_num < MMC_SLOT_NUM; slot_num++)
  {
    gDetChatCnt[slot_num] = MMC_INS_CHAT;   /* Reset the counter of chattering de
tection. */
    gDetSts_Old[slot_num] = MMC_FALSE;      /* Set the previous status of detecti
on to "removal".   */
  }
}
```

4.5.3 ボードセットアップインタフェース関数の修正

(1) インクルードファイルの修正

MMC ドライバのヘッダファイルと共通関数のヘッダファイルを設定してください。

■r_target_io.c　35 行目付近

```
#include "r_cg_macrodriver.h"

#include "r_mtl_com.h"
#include "r_mmc.h"
#include "r_board.h"
```

(2) storage_driver_init()の修正

MMC ドライバの初期化関数をコールします。

■r_target_io.c　54 行目付近

```
void storage_driver_init(void)
{
  R_mmc_Init_Driver();
```

5. 使用上の注意事項

5.1 ヘッダファイルのインクルード

アプリケーション・ソフトウェアに MMC ドライバのヘッダファイルと共通関数のヘッダファイルをインクルードしてください。

```
#include "r_mtl_com.h"
#include "r_mmc.h"
```

5.2 MMC ドライバを組み込み時に必要なファイル

MMC ドライバは、MMC を制御するためのソフトウェアです。

このソフトウェアは以下のアプリケーションノートに同梱されています。別途、ルネサスエレクトロニクスのホームページから入手してください。

・ RL78 ファミリ　SPI モードマルチメディアカードドライバ： 導入ガイド（R02AN0158JJ）

5.3 使用制限事項

ディスクキャッシュ機能を持たせる場合のメモリドライバインタフェース処理としての R_tfat_disk_ioctl()記述ではありません。本資料の記述内容の場合、TFAT ライブラリの使用上、R_tfat_f_sync()をコールしないでください。

5.4 MMC ドライバの初期化手順

以下の順で MMC ドライバの初期化とデバイスの初期化を実行してください。

①storage_driver_init()関数の実行
　MMC ドライバのドライバ初期化関数(R_mmc_Init_Driver())を実行します。
②R_tfat_disk_initialize()関数の実行
　MMC を検出し、デバイス初期化関数(R_mmc_Init_Slot())を実行します。

5.5 動作確認

下記アプリケーションノートのサンプルコードで動作確認済みです。

・ ドキュメントタイトル: 「RL78/G14 CPU ボードを用いた音声再生・録音デモ」(R20AN0194)

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/

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013.03.29	—	初版発行
2.00	2021.07.14	—	RL78/G23 に対応 メモリカードドライバを「SPI モード MMC/SD カードドライバ」から「SPI モード マルチメディアカードドライバ」へ変更

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. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   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、将来の拡張機能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。
8. 製品間の相違について
   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、フラッシュメモリ、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、ノイズ幅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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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も、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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。
13. 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。
14. 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.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。