RX ファミリ

システムタイマモジュール Firmware Integration Technology

要旨

本アプリケーションノートは、Firmware Integration Technology (FIT)を使用したシステムタイマモジュールについて説明します。本モジュールは CMT を使用して、年月日時間分秒のカウント、簡易的なスケジューラ動作を行います。以降、本モジュールをシステムタイマ FIT モジュールと称します。

動作確認デバイス

RX ファミリ

対象コンパイラ

・Renesas Electronics C/C++ Compiler Package for RX Family

・GCC for Renesas RX

・IAR C/C++ Compiler for Renesas RX

各コンパイラの動作確認内容については 4.1 動作確認環境を参照してください。

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^2 studio に組み込む方法 Firmware Integration Technology (R01AN1723)
- CS+に組み込む方法 Firmware Integration Technology (R01AN1826)
- RX スマート・コンフィグレータ ユーザーガイド: e^2 studio 編(R20AN0451)
- RX スマート・コンフィグレータ ユーザーガイド: CS+編(R20AN0470)
- RX スマート・コンフィグレータ ユーザーガイド: IAREW 編(R20AN0535)
- CMT モジュール Firmware Integration Technology (R01AN1856)

目次

1. 概要

1.1 システムタイマ FIT モジュールとは

本モジュールは API として、プロジェクトに組み込んで使用します。本モジュールの組み込み方については、「2.12FIT モジュールの追加方法」を参照してください。

1.2 システムタイマ FIT モジュールの概要

システムタイマ FIT モジュールは CMT を使用して、年月日時間分秒をカウントします。また、簡易的なスケジューラ機能も持ちます。ユーザはシステムタイマに呼び出し間隔を指定して関数ポインタを登録することで、定常的に行う必要のある処理をシステムタイマに実行させることができます。

1.3 API の概要

表 1.1 に本モジュールに含まれる API 関数を示します。

表 1.1　API 関数一覧

関数	関数説明
R_SYS_TIME_Open()	システムタイマモジュールを開始します
R_SYS_TIME_GetCurrentTime()	システムタイマからシステム時間を取得します
R_SYS_TIME_SetCurrentTime()	システムタイマにシステム時間を設定します
R_SYS_TIME_ConvertUnixTimeToSystemTime()	Unix 時間をシステムタイマの書式に変換します
R_SYS_TIME_RegisterPeriodicCallback()	周期起動関数を登録します。（最大 30 個）
R_SYS_TIME_UnregisterPeriodicCallback()	周期起動関数の登録を解除します
R_SYS_TIME_IsPeriodicCallbackRegistered()	周期起動関数の登録の有無を確認します
R_SYS_TIME_Close()	システムタイマモジュールを終了します
R_SYS_TIME_GetVersion()	システムタイマモジュールのバージョンを取得します。

1.4　ファイル構成

本アプリケーションノートは、以下の表 1-2 のファイルが含まれます。

表 1-2　ファイル構成 1

ファイル/ディレクトリ**(太字)**名	内容
r20an0431jj0101-rx-middle.pdf	アプリケーションノート(日本語、本書)
r20an0431ej0101-rx-middle.pdf	アプリケーションノート(英語)
FITModules	FIT モジュールフォルダ
r_sys_time_rx_v1.01.zip	システムタイマモジュール
r_sys_time_rx_v1.01.xml	システムタイマモジュール e2 studio FIT コンフィグレータ用 XML ファイル

r_sys_time_rx_v1.01.zip を解凍したフォルダには、以下の表 1-3 のファイルが含まれます。

表 1-3　ファイル構成 2

ファイル/ディレクトリ**(太字)**名	内容
r_config	システムタイマコンフィグファイルフォルダ
r_sys_time_rx_config.h	システムタイマコンフィグファイル(デフォルト設定)
r_sys_time_rx	システムタイマ FIT Module フォルダ
src	システムタイマソースコードフォルダ
r_sys_time_rx.c	システムタイマソースコード
r_sys_time_rx_private.h	システムタイマ内部ヘッダファイル
doc	システムタイマ ドキュメントフォルダ
ja	システムタイマ ドキュメントフォルダ(日本語)
r20an0431jj0101-rx-middle.pdf	システムタイマ アプリケーションノート(日本語)
en	システムタイマドキュメントフォルダ(英語)
r20an0431ej0101-rx-middle.pdf	システムタイマアプリケーションノート(英語)
ref	システムタイマコンフィグファイル(テンプレート)フォルダ
r_sys_time_rx_config_reference.h	システムタイマ コンフィグファイル(テンプレート)
r_sys_time_rx_if.h	システムタイマヘッダファイル
readme.txt	readme

2. API 情報

本 FIT モジュールは、下記の条件で動作を確認しています。

2.1 ハードウェアの要求

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。

- CMT

2.2 ソフトウェアの要求

このドライバは以下の FIT モジュールに依存しています。

- r_bsp
- r_cmt_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「4.1 動作確認環境」に示すツールチェーンで動作確認を行っています。

2.4 使用する割り込みベクタ

なし

2.5 ヘッダファイル

すべての API 呼び出しとそれをサポートするインタフェース定義は r_sys_time_rx_if.h に記載しています。

2.6 整数型

このドライバは ANSI C99 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。

2.7 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_sys_time_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 r_sys_time_rx_config.h	
オプションなし	—

2.8　コードサイズ

本モジュールの ROM サイズ、RAM サイズ、最大使用スタックサイズを下表に示します。

下表の値は下記条件で確認しています。

モジュールリビジョン: r_sys_time_rx Rev.1.01

コンパイラバージョン: Renesas Electronics C/C++ Compiler Package for RX Family V3.01.00
(統合開発環境のデフォルト設定に”-lang = c99”オプションを追加)
GCC for Renesas RX 4.8.4.201801
(統合開発環境のデフォルト設定に”-std=gnu99”オプションを追加)
IAR C/C++ Compiler for Renesas RX version 4.11.1
(統合開発環境のデフォルト設定)

コンフィグレーションオプション: デフォルト設定

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		
		Renesas Compiler	GCC	IAR Compiler
RX64M	ROM	**2.0k バイト**	**3.2k バイト**	**2.5k バイト**
	RAM	**0.5k バイト**	**0.5k バイト**	**0.5k バイト**
	スタック (注 1)	**88 バイト**	**-**	**80 バイト**

(注 1) R_SYS_TIME_SetCurrentTime()実行時

2.9　引数

API 関数の引数である構造体を示します。この構造体は、API 関数のプロトタイプ宣言とともに r_sys_time_rx_if.h に記載されています。

2.10　戻り値

API 関数の戻り値を示します。この列挙型は、API 関数のプロトタイプ宣言とともに r_sys_time_rx_if.h で記載されています。

```
typedef enum e_sys_time_err
{
     SYS_TIME_SUCCESS=0,                 /* Normally terminated. */
     SYS_TIME_ERR_BAD_CHANNEL,           /* Non-existent channel number. */
     SYS_TIME_ERR_BAD_INTERVAL,          /* Bad interval parameter is
                                            specified. */
     SYS_TIME_ERR_BAD_TIME_OFFSET,       /* Bad time offset is set. */
     SYS_TIME_ERR_BAD_FUNCTION_POINTER,  /* Bad function pointer is set. */
     SYS_TIME_ERR_BAD_SYS_TIME,          /* Bad system timer value is input */
     SYS_TIME_ERR_ALREADY_STARTED,       /* System timer is already started. */
     SYS_TIME_ERR_NOT_STARTED,           /* System timer is not started. */
     SYS_TIME_ERR_FULL_REGISTERED,       /* All register table is used. */
     SYS_TIME_ERR_ALREADY_REGISTERED,    /* Specified function pointer has been
                                            already registered. */
}
sys_time_err_t;
```

2.11　コールバック関数

本モジュールでは、CMT 割り込みが発生したタイミングで。ユーザが設定したコールバック関数を呼び出します。

コールバック関数は、「R_SYS_TIME_ConvertUnixTimeToSystemTime()」に記載された引数 "function_pointer" に、ユーザの関数のアドレスを格納することで設定されます。

```
コールバック関数例：
void my_sys_time_callback(void)
{
  ...
}
```

2.12　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^2 studio 上で Smart Configurator を使用して FIT モジュールを追加する場合
e^2 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^2 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^2 studio 上で FIT Configurator を使用して FIT モジュールを追加する場合
e^2 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^2 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^2 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。

2.13　for 文、while 文、do while 文について

本モジュールでは、レジスタの反映待ち処理等で for 文、while 文、do while 文（ループ処理）を使用しています。これらループ処理には、「WAIT_LOOP」をキーワードとしたコメントを記述しています。そのため、ループ処理にユーザがフェイルセーフの処理を組み込む場合は、「WAIT_LOOP」で該当の処理を検索できます。

以下に記述例を示します。

```
while 文の例：
/* WAIT_LOOP */
while(0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized. */
}

for 文の例：
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}

do while 文の例：
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /* WAIT_LOOP */
```

2.14　セクション情報

システムタイマモジュールはデフォルトセクションを使用します。

3. API 関数

R_SYS_TIME_Open ()

Format

#include “r_sys_time_rx_if.h”

sys_time_err_t R_SYS_TIME_Open(void);

Parameters

なし

Return Values

SYS_TIME_SUCCESS	正常終了
SYS_TIME_ERR_BAD_CHANNEL	CMT のチャネルの空きが無い
SYS_TIME_ERR_ALREADY_STARTED	既に開始済み

Description

システムタイマモジュールを起動します。

R_SYS_TIME_GetCurrentTime ()

Format

#include “r_sys_time_rx_if.h”

sys_time_err_t R_SYS_TIME_GetCurrentTime(SYS_TIME *sys_time);

Parameters

sys_time	入力/出力	システムタイマからシステム時間を取得する領域

Return Values

SYS_TIME_SUCCESS	正常終了

Description

システムタイマからシステム時間を取得します。

R_SYS_TIME_SetCurrentTime ()

Format

#include “r_sys_time_rx_if.h”

sys_time_err_t R_SYS_TIME_SetCurrentTime(SYS_TIME *sys_time);

Parameters

sys_time	入力/出力	システムタイマにシステム時間を設定する領域

Return Values

SYS_TIME_SUCCESS	正常終了
SYS_TIME_ERR_BAD_SYS_TIME	異常なシステム時間が設定されました。

Description

システムタイマにシステム時間を設定します。システム時間に内蔵される Unix 時間も更新します。

R_SYS_TIME_ConvertUnixTimeToSystemTime ()

Format

#include “r_sys_time_rx_if.h”

```
sys_time_err_t R_SYS_TIME_ConvertUnixTimeToSystemTime(
        uint32_t unix_time, SYS_TIME *sys_time, uint8_t *time_offset);
```

Parameters

unix_time	入力	Unix 時間
sys_time	入力/出力	システムタイマにシステム時間を設定する領域
time_offset	入力	タイムゾーンを指定する文字列

Return Values

SYS_TIME_SUCCESS	正常終了
SYS_TIME_ERR_BAD_TIME_OFFSET	異常なタイムゾーンが設定されました。

Description

Unix 時間を用いてシステムタイマにシステム時間を設定します。time_offset にタイムゾーンを表す文字列を指定することで、システム時間をタイムゾーンに補正できます。タイムゾーンを表す文字列は、r_sys_time_rx_if.h に定義されています。

R_SYS_TIME_RegisterPeriodicCallback ()

Format

#include “r_sys_time_rx_if.h”

sys_time_err_t R_SYS_TIME_RegisterPeriodicCallback(

callback_from_sys_time_t function_pointer, uint32_t interval)

Parameters

function_pointer	入力	関数ポインタ
interval	入力	起動周期 (単位=10ms)

Return Values

SYS_TIME_SUCCESS	正常終了
SYS_TIME_ERR_BAD_FUNCTION_POINTER,	異常な関数ポインタが指定された
SYS_TIME_ERR_BAD_INTERVAL	異常な起動周期が指定された
SYS_TIME_ERR_FULL_REGISTERED	関数ポインタ登録数の上限に達した
SYS_TIME_ERR_ALREADY_REGISTERED	既に登録済みの関数ポインタが指定された

Description

ユーザは本関数を使用し起動周期を指定して関数ポインタを登録することで、定常的に行う必要のある処理をシステムタイマに実行させることができます。最大 30 個の関数ポインタを登録することができます。関数ポインタは CMT 割り込み内で起動されます。この CMT 割り込みでは多重割り込みを許可しています。リアルタイム性の必要な処理は、CMT 割り込みの優先度(r_cmt_rx_config.h で設定)より高い優先度の割り込みで処理してください。

R_SYS_TIME_UnregisterPeriodicCallback ()

Format

#include “r_sys_time_rx_if.h”

sys_time_err_t R_SYS_TIME_UnregisterPeriodicCallback(callback_from_sys_time_t function_pointer);

Parameters

function_pointer	入力	関数ポインタ

Return Values

SYS_TIME_SUCCESS	正常終了
SYS_TIME_ERR_BAD_FUNCTION_POINTER	異常な関数ポインタが指定された

Description

周期起動関数の登録を解除できます。

R_SYS_TIME_IsPeriodicCallbackRegistered ()

Format

#include “r_sys_time_rx_if.h”

bool R_SYS_TIME_IsPeriodicCallbackRegistered(callback_from_sys_time_t function_pointer);

Parameters

function_pointer入力	関数ポインタ

Return Values

true	登録済み
false	登録無し

Description

周期起動関数の登録を確認できます。

R_SYS_TIME_Close ()

Format

#include “r_sys_time_rx_if.h”

sys_time_err_t R_SYS_TIME_Close(void);

Parameters

なし

Return Values

SYS_TIME_SUCCESS	正常終了
SYS_TIME_ERR_BAD_CHANNEL	CMT チャネルのクローズに失敗
SYS_TIME_ERR_NOT_STARTED	システムタイマが起動していない

Description

システムタイマを停止します。

R_SYS_TIME_GetVersion ()

Format

#include “r_sys_time_rx_if.h”

uint32_t R_SYS_TIME_GetVersion(void);

Parameters

なし

Return Values

本モジュールのバージョン

Description

この関数は本モジュールのバージョンを返します。バージョン番号は符号化され、最上位の 2 バイトがメジャーバージョン番号を、最下位の 2 バイトがマイナーバージョン番号を示しています。

4. 付録

4.1 動作確認環境

本 FIT モジュールの動作確認環境を以下に示します。

表 4.1 動作確認環境 (Rev.1.01)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e^2 studio V7.3.0 IAR Embedded Workbench for Renesas RX 4.11.1
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 V3.01.00 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
	GCC for Renesas RX 4.8.4.201801 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 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.11.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev.1.01
使用ボード	Renesas Starter Kit+ for RX65N-2MB (RTK50565Nxxxxxx) Renesas Starter Kit+ for RX64M (R0K50564Mxxxxxx)

4.2 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^2 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^2 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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	2019/06/28	-	対象コンパイラを追加しました: GCC for Renesas RX, IAR C/C++ Compiler for Renesas RX 2.10 戻り値の定義名を変更しました。 以下、不具合修正 R_SYS_TIME_GetCurrentTime()実行中に割り込みが発生した時、時間情報が不正になる問題を修正。
1.00	2016/11/30	-	新規作成

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。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も、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)

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。