

RX ファミリ

MPC モジュール

Firmware Integration Technology

R01AN1724JJ0231
Rev. 2.31
2017.10.31

要旨

本ドキュメントは、Firmware Integration Technology (FIT)を使用した MPC モジュールについて説明します。本モジュールを使って、マルチファンクションピンコントローラ (MPC) をシステムに組み込んでご使用いただけます。

以降、本モジュールを MPC FIT モジュールと称します。

対象デバイス

本モジュールは以下のデバイスで使用できます。

- RX110 グループ、RX111 グループ、RX113 グループ
- RX130 グループ
- RX210 グループ
- RX230 グループ
- RX231 グループ
- RX23T グループ
- RX24T グループ
- RX24U グループ
- RX63N グループ
- RX64M グループ
- RX651、RX65N グループ
- RX71M グループ

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。

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)
- e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)
- CS+に組み込む方法 Firmware Integration Technology (R01AN1826)
- GPIO モジュール Firmware Integration Technology (R01AN1721)

目次

1.	概要	3
2.	API 情報.....	4
2.1	ハードウェアの要求	4
2.2	ハードウェアリソースの要求	4
2.2.1	MPC レジスタ	4
2.3	ソフトウェアの要求	4
2.4	制限事項	4
2.5	対応ツールチェーン	4
2.6	ヘッダファイル	4
2.7	整数型	4
2.8	コンパイル時の設定	5
2.9	コードサイズ	5
2.10	API データ型	6
2.10.1	MPC の端子設定	6
2.11	戻り値	7
2.12	FIT モジュールをプロジェクトに追加する方法	7
3.	API 関数.....	8
3.1	概要	8
3.2	R_MPC_Write	9
3.3	R_MPC_Read ().....	11
3.4	R_MPC_GetVersion()	12
4.	デモプロジェクト.....	13
4.1	mpc_demo_rskrx113、mpc_demo_rskrx64m、mpc_demo_rskrx71m、 mpc_demo_rskrx65n、 mpc_demo_rskrx65n_2m	13
4.2	mpc_demo_rskrx231	13
4.3	ワークスペースへのデモ追加	13
4.4	デモのダウンロード方法.....	13
5.	付録	14
5.1	動作確認環境	14
5.2	トラブルシューティング	15

1. 概要

最近の MCU は小ピンパッケージを維持する一方で、装備される周辺機能は増加し続けています。小ピンパッケージで多数の周辺機能を備えるには、各端子に複数の機能を割り当てることになります。RX MCU では、端子への機能の割り当ては、MPC によって制御されます。本モジュールでは、GPIO FIT モジュールと連動して MPC 機能が使用できます。

2. API 情報

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。

2.1 ハードウェアの要求

本モジュールを使用するには、ご使用の MCU が以下の機能をサポートしていることが要求されます。

- マルチファンクションピンコントローラ (MPC) : MPC によって端子に周辺機能を割り当てます。

2.2 ハードウェアリソースの要求

ここでは、本モジュールが要求するハードウェアの周辺機能について説明します。特に記載がない場合、ここで説明するリソースは本モジュールが使用できるように、ユーザのプログラムでは使用しないでください。

2.2.1 MPC レジスタ

本モジュールの API 関数は、すべての MPC レジスタに書き込みが可能です。また、ユーザのプログラムでも、これらのレジスタを変更できます。

2.3 ソフトウェアの要求

本モジュールは以下のソフトウェアに依存します。

- ルネサスボードサポートパッケージ (r_bsp)
- GPIO モジュール (r_gpio_rx)

2.4 制限事項

特になし

2.5 対応ツールチェーン

本モジュールは下記ツールチェーンで動作確認を行っています。

Renesas RX Toolchain v.2.02.00 (RX110、RX111、RX113、RX210、RX231、RX63N、RX64M、RX71M)
Renesas RX Toolchain v.2.03.00 (RX130、RX230、RX23T、RX24T)
Renesas RX Toolchain v.2.05.00 (RX24U、RX65N)
Renesas RX Toolchain v.2.06.00 (RX24U)
Renesas RX Toolchain v.2.07.00 (RX65N-2MB、RX130-512KB)

2.6 ヘッドファイル

すべての API 呼び出しと対応するインタフェース定義は、r_mpc_rx_if.h ファイルに記述されています。

r_mpc_rx_config.h ファイルで、ビルド時に設定可能なコンフィギュレーションオプションを選択あるいは定義できます。

上記 2 ファイルは、ユーザアプリケーションによってインクルードする必要があります。

2.7 整数型

コードをわかりやすく、また移植が容易に行えるように、本プロジェクトでは ANSI C99 (Exact width integer types (固定幅の整数型)) 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。

2.8 コンパイル時の設定

ビルド時に設定可能なコンフィギュレーションオプションは `r_mpc_rx_config.h` ファイルに含まれます。下表に各設定の概要を示します。

コンフィギュレーションオプション (r_mpc_rx_config.h)	
定義	説明
MPC_CFG_PARAM_CHECKING_ENABLE	<ul style="list-style-type: none"> 1: ビルド時にパラメータチェック処理をコードに含めます。 0: ビルド時にパラメータチェック処理をコードから省略します。 BSP_CFG_PARAM_CHECKING_ENABLE (デフォルト): システムのデフォルト設定を使用します。 注: ビルド時にパラメータチェックのコードを省略することで、コードサイズを小さくすることができます。

表 1: コンフィギュレーション情報

2.9 コードサイズ

本モジュールのコードサイズを下表に示します。

ROM (コードおよび定数) と RAM (グローバルデータ) のサイズは、ビルド時の「2.8 コンパイル時の設定」のコンフィギュレーションオプションによって決まります。掲載した値は、「2.5 サポートされているツールチェーン」の C コンパイラでコンパイルオプションがデフォルト時の参考値です。コンパイルオプションのデフォルトは最適化レベル: 2、最適化のタイプ: サイズ優先、データ・エンディアン: リトルエンディアンです。コードサイズは 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。

ROM、RAM およびスタックのコードサイズ			
項目	分類	使用メモリ	備考
パラメータ チェック 処理あり	ROM	125 バイト	
	RAM	0 バイト	
	最大使用スタックサイズ	24 バイト	
パラメータ チェック 処理なし	ROM	107 バイト	
	RAM	0 バイト	
	最大使用スタックサイズ	24 バイト	

表 2: ROM、RAM およびスタックのコードサイズ

2.10 API データ型

2.10.1 MPC の端子設定

本データ構造体は、端子の機能設定に使用します。"pin_function"に有効な設定は、お使いの MCU のハードウェアマニュアルで「マルチファンクションピンコントローラ (MPC)」の章をご覧ください。対象の端子のポートに対応した端子機能制御レジスタを選択します。マニュアルでは、選択されたポートの各端子に割り当て可能な機能の一覧が示されています。以下に例を示します。

```
/* 端子の MPC レジスタ設定オプション */
```

```
typedef struct
```

```
{
```

```
    uint8_t pin_function;    //端子に割り当てる周辺機能
```

```
    bool    irq_enable;      //端子を IRQ 端子として使用
```

```
    bool    analog_enable;   //端子を ADC 入力端子、または DAC 出力端子として、  
                                //または LVD (CMPA2) に使用
```

```
} mpc_config_t;
```

2.11 戻り値

以下に R_MPC_Write()関数で利用できる戻り値を示します。

/* 関数の戻り値 */

```
typedef enum
```

```
{
```

```
    MPC_SUCCESS = 0,
```

```
    MPC_ERR_INVALID_CFG, // 指定された設定はこの端子には適用できません。
```

```
} mpc_err_t;
```

2.12 FIT モジュールをプロジェクトに追加する方法

FIT モジュールは、本モジュールを使用するプロジェクトごとに追加する必要があります。ルネサスは、(1)または(3)で説明された「スマート・コンフィグレータ」の使用を推奨します。ただし、「スマート・コンフィグレータ」はすべての RX デバイスをサポートしているわけではありません。サポートしていない RX デバイスについては、(2)または(4)の方法で対応してください。

- (1) e² studio の「スマート・コンフィグレータ」を使用して FIT モジュールをプロジェクトに追加する方法
e² studio の「スマート・コンフィグレータ」を使うと、FIT モジュールは自動的にプロジェクトに追加されます。詳しくは、「Renesas e² 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) e² studio の「FIT コンフィグレータ」を使用して FIT モジュールをプロジェクトに追加する方法
e² studio の FIT コンフィグレータを使うと、FIT モジュールは自動的にプロジェクトに追加されます。詳しくは、「FIT モジュールをプロジェクトに追加する方法(R01AN1723)」を参照してください。
- (3) CS+の「スマート・コンフィグレータ」を使用して FIT モジュールをプロジェクトに追加する方法
CS+の「スタンドアロン版スマート・コンフィグレータ」を使うと、FIT モジュールは自動的にプロジェクトに追加されます。詳しくは、「Renesas e² 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) FIT モジュールを CS+のプロジェクトに追加する方法
CS+では、FIT モジュールを手動でプロジェクトに追加してください。詳しくは、「FIT モジュールを CS+のプロジェクトに追加する方法(R01AN1826)」を参照してください。

3. API 関数

3.1 概要

本モジュールには以下の関数が含まれます。

関数	説明
R_MPC_Write()	端子の機能を設定します。
R_MPC_Read()	端子の機能設定を読み出します。
R_MPC_GetVersion()	本モジュールのバージョン番号を返します。

3.2 R_MPC_Write

この関数は端子の機能を設定します。

Format

```
mpc_err_t R_MPC_Write(gpio_port_pin_t pin, mpc_config_t * pconfig);
```

Parameters

pin

設定する端子

pconfig

端子の設定情報を含む構造体へのポインタ。セクション2.10.1参照。

Return Values

MP_C_SUCCESS /*成功; 端子が設定されました。*/

MP_C_ERR_INVALID_CFG /*エラー; 無効な設定が入力されました。*/

Properties

ファイル *r_mpc_rx_if.h* にプロトタイプ宣言されています。

Description

本関数は"mpc_config_t"構造体の情報を元に端子を設定します。すべての端子が同じ機能をサポートしているわけではありません。例えば、どの端子でも ADC や DAC のアナログ端子として設定できるわけではありません。また、機能の組み合わせもどれでもいいというわけではなく、アナログ端子と周辺機能の端子としての機能は同時に設定できません。

端子に対して割り当て可能な機能については、お使いの MCU のハードウェアマニュアルで「マルチファンクションピンコントローラ (MPC)」の章をご覧ください。対象の端子のポートに対応した端子機能制御レジスタを選択します。マニュアルでは、選択されたポートの各端子に割り当て可能な機能表をご覧ください。

本関数によって設定する端子は、*r_gpio_rx* モジュールの"gpio_port_pin_t"型を使って定義されます。

Reentrant

本関数は、設定するポートが異なる場合、再入可能 (リエントラント) です。

Example

```
mpc_config_t      config;
gpio_port_pin_t   pin;

/* PE0 を ADC のアナログ端子として使用するよう設定 */
pin = GPIO_PORT_E_PIN_0;

config.analog_enable = true;
config.irq_enable = false;
config.pin_function = 0;

if (MPC_SUCCESS != R_MPC_Write(pin, &config))
{
    /* エラー、端子はこの設定に対応していません。ハンドルエラー */
    ...
}

/* P27 を IRQ 端子、および SCI の処理で使用するよう設定 */
pin = GPIO_PORT_2_PIN_7;

config.analog_enable = false;
config.irq_enable = true;
config.pin_function = 0xA;

if (MPC_SUCCESS != R_MPC_Write(pin, &config))
{
    /* エラー、端子はこの設定に対応していません。ハンドルエラー */
    ...
}
```

3.3 R_MPC_Read ()

この関数は端子の機能設定を読み出します。

Format

```
void R_MPC_Read(gpio_port_pin_t pin, mpc_config_t * pconfig);
```

Parameters

pin

設定情報を読み出す端子

pconfig

端子の設定情報を含む構造体へのポインタ。セクション2.10.1参照。

Return Values

なし

Properties

ファイル `r_mpc_rx_if.h` にプロトタイプ宣言されています。

Description

本関数は端子の設定情報を読み出し、ユーザによって提供される構造体に格納します。

Reentrant

本関数は再入可能（リエントラント）です。

Example

```
mpc_config_t config;

/* P03 は DAC のアナログ端子として既に設定されているかどうかを確認します。*/
R_MPC_Read(GPIO_PORT_0_PIN_3, &config);

if (config.analog_enable == true)
{
    /* P03 は既にアナログ端子として設定されています。*/
    ...
}
else
{
    /* P03 はまだ設定されていません。ここで設定してください。*/
    ...
}
```

3.4 R_MPC_GetVersion()

この関数は実行時に本モジュールのバージョンを返します。

Format

```
uint32_t R_MPC_GetVersion(void);
```

Parameters

なし

Return Values

本モジュールのバージョン

Properties

ファイル `r_mpc_rx_if.h` にプロトタイプ宣言されています。

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。例えば、ver. 4.25 の場合、“0x00040019” が返されます。

Reentrant

この関数は再入可能（リエントラント）です。

Example

```
uint32_t cur_version;

/* 実行中の r_mpc_rx API のバージョンを取得 */
cur_version = R_MPC_GetVersion();

/* 本アプリケーションを使用可能なバージョンであることを確認 */
if (MIN_VERSION > cur_version)
{
    /* お使いの r_mpc_rx バージョンは、本アプリケーションに必要な XXX 機能を備えていない旧バージョン
    です。警告 */
    ....
}
```

Special Notes:

本関数は `r_mpc_rx.c` のインライン関数となります。

4. デモプロジェクト

デモプロジェクトはスタンドアロンプログラムです。デモプロジェクトには、FIT モジュールとそのモジュールが依存するモジュール（例：r_bsp）を使用する main()関数が含まれます。本 FIT モジュールには以下のデモプロジェクトが含まれます。

4.1 mpc_demo_rskrx113、mpc_demo_rskrx64m、mpc_demo_rskrx71m、mpc_demo_rskrx65n、mpc_demo_rskrx65n_2m

mpc_demo_rskrx113、mpc_demo_rskrx64m、mpc_demo_rskrx71m、mpc_demo_rskrx65n および mpc_demo_rskrx65n_2m は RX113、RX64M、RX71M、RX65N および RX65N-2MB における MPC FIT モジュールのデモプログラムです。これらのプログラムで、MPC を使って、割り込みの入力としてポートビットを設定する方法をデモします。デモでは割り込みとして IRQ2 が選択され、SW2 の押下の検出に使用されます。コードがコンパイルされ、対象のボードにダウンロードして実行されると、SW2 を押して、IRQ2（RX65N では IRQ9、RX65N-2MB では IRQ13）割り込みを発生させることができます。次に IRQ2（RX65N では IRQ9、RX65N-2MB では IRQ13）割り込み処理で、LED3 の状態がトグルされます。

4.2 mpc_demo_rskrx231

mpc_demo_rskrx231 プロジェクトは、RX231 MPC FIT モジュールのデモプログラムです。プログラムの動作は mpc_demo_rskrx113 と同様ですが、SW2 の押下の検出に IRQ4 が使用されます。

4.3 ワークスペースへのデモ追加

デモプロジェクトは、本アプリケーションノートで提供されるファイルの FITDemos サブディレクトリにあります。ワークスペースにデモプロジェクトを追加するには、「ファイル」>>「インポート」を選択し、「インポート」ダイアログから「一般」の「既存プロジェクトをワークスペースへ」を選択して「次へ」ボタンをクリックします。「インポート」ダイアログで「アーカイブ・ファイルの選択」ラジオボタンを選択し、「参照」ボタンをクリックして FITDemos サブディレクトリを開き、使用するデモの zip ファイルを選択して「終了」をクリックします。

4.4 デモのダウンロード方法

デモプロジェクトは、RX Driver Package には同梱されていません。デモプロジェクトを使用する場合は、個別に各 FIT モジュールをダウンロードする必要があります。「スマートブラウザ」の「アプリケーションノート」タブから、本アプリケーションノートを右クリックして「サンプル・コード（ダウンロード）」を選択することにより、ダウンロードできます。

5. 付録

5.1 動作確認環境

このセクションでは、GPIO FIT モジュールの動作確認用の環境について説明します。

表 5.1 動作確認環境 (Rev.2.31)

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 V6.0.0
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 V2.07.00 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
モジュールのリビジョン	Rev2.31
使用ボード	Renesas Starter Kit+ for RX 65N-2MB (型名：RTK50565N2CxxxxxBR) Renesas Starter Kit+ for RX130-512KB (型名：RTK5051308CxxxxxBR)

表 5.2 動作確認環境 (Rev.2.30)

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 V6.0.0
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 V2.07.00 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
モジュールのリビジョン	Rev2.30
使用ボード	Renesas Starter Kit+ for RX 65N-2MB (型名：RTK50565N2CxxxxxBR) Renesas Starter Kit+ for RX130-512KB (型名：RTK5051308CxxxxxBR)

5.2 トラブルシューティング

- (1) Q : プロジェクトに FIT モジュールを追加し、プロジェクトをビルドしました。次のエラーが発生しました。「ソースファイル「platform.h」を開くことができません。」

A : FIT モジュールがプロジェクトに対して正しく追加されていない可能性があります。次の文書で、FIT モジュールを追加した方法が正しいかどうか確認してください。

- 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)』
- e² 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 e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)』

FIT モジュールを追加する場合、ボードサポートパッケージの 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。『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。

- (2) Q : プロジェクトに FIT モジュールを追加し、プロジェクトをビルドしました。次のエラーが発生しました。「現在の r_mpc_rx モジュールはこの MCU をサポートしていません。」

A : 追加した FIT モジュールは、現在のプロジェクトで選択されている対象デバイスをサポートしていない可能性があります。追加した FIT モジュールがサポートしているデバイスを確認してください。

- (3) Q : プロジェクトに FIT モジュールを追加し、プロジェクトをビルドしました。構成の設定が誤っている場合に対応するエラーが発生しました。

A : 「r_mpc_rx_config.h」ファイル内の設定が誤っている可能性があります。
「r_mpc_rx_config.h」ファイルを確認してください。誤った設定が存在している場合、その設定にとって正しい値を設定してください。詳細については、「2.8 コンパイル時の設定」を参照してください。

テクニカルアップデートの対応について

本モジュールは以下のテクニカルアップデートの内容を反映しています。
対応しているテクニカルアップデートはありません。

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
1.70	2015.09.30	—	初版発行
1.80	2015.10.01	—	FIT モジュールの RX130 グループ対応
1.90	2015.12.01	— 1, 6 12	FIT モジュールの RX24T グループ対応 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 Firmware Integration Technology」のドキュメント番号変更 「4. デモプロジェクト」追加
2.00	2016.02.01	— 13	FIT モジュールの RX230 グループ対応 「テクニカルアップデートの対応について」追加
2.01	2016.07.29	12	「4. デモプロジェクト」に RSKRX64M を追加 「4.1 mpc_demo_rskrx113、mpc_demo_rskrx64m、mpc_demo_rskrx71m」の誤記修正 ・ ” cmt_demo_rskrx113” -> “ mpc_demo_rskrx113” ・ ” cmt_demo_rskrx71M” -> “ mpc_demo_rskrx71M”
2.10	2016.10.01	— 5	FIT モジュールの RX65N グループ対応 2.9「コードサイズ」の表フォーマットを変更 2.9「コードサイズ」に RX65N グループのコードサイズを追加
2.20	2017.02.28	— 4	FIT モジュールの RX24U グループ対応 「2.5 対応ツールチェーン」に RXC v2.06.00 を追加
2.30	2017.07.21	- 4 7	FIT モジュールの RX130 グループ（ROM 512KB 版）と RX65N グループ（ROM 2MB 版）対応 「2.5 対応ツールチェーン」に RXC v2.07.00 を追加 「2.12 FIT モジュールをプロジェクトに追加する方法」を更新
2.31	2017.10.31	13 14	RSKRX65N と RSKRX65N-2MB を、「4. デモプロジェクト」に追加 「4.4 デモのダウンロード方法」を追加 「5. 付録」を追加

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、その他の不適切に使用しないでください。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、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、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。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
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. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 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. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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。

(Rev.3.0-1 2016.11)



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/>