R20AN0044JJ0108
Rev.1.08
Mar.20.2025

RXファミリ
AESライブラリ 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 RX ファミリ AES ライブラリ (以下 AES FIT ライブラリ) を導入するための情報を記します。AES FIT ライブラリは AES 暗号処理を RX マイコンで実現するためのソフトウェアライブラリです。AES FIT ライブラリは RX マイコンを用いて効率よく処理が出来るように設計されています。

またこの AES FIT ライブラリは Galois/Counter Mode(GCM)に対応した GCM ドライバの機能も含めます。

AES ライブラリの使用方法については、パッケージに格納されているユーザーズマニュアル (R20UW0068JJ0200) を参照してください。

動作確認デバイス

RX ファミリ

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。

ターゲットコンパイラ

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

GCC for Renesas RX

IAR Embedded Workbench for Renesas RX

コンパイラの動作確認環境に関する詳細な内容は、セクション「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)

e2studio に組み込む方法 Firmware Integration Technology (R01AN1723)

CS+に組み込む方法 Firmware Integration Technology (R01AN1826)

Renesas e2studio スマート・コンフィグレータ ユーザガイド (R20AN0451)

目次

1. 概要 3
1.1 AES FIT ライブラリとは 3
1.2 AES FIT ライブラリの概要 3
1.3 API関数 3
1.4 バージョン情報 3
1.5 AES FITライブラリの製品構成 4
1.5.1 アプリケーションノート構成 4
1.5.2 ファイル構成 4

2. API情報 6
2.1 ハードウェアの要求 6
2.2 ソフトウェアの要求 6
2.3 制限事項 6
2.4 サポートされているツールチェーン 6
2.5 ヘッダファイル 6
2.6 整数型 6
2.7 ビルド時の注意事項 6
2.8 ライブラリ関数の使用方法 6
2.8.1 処理性能 vs. コードサイズ 6
2.9 AESライブラリROM / RAM / Stack Size / 処理サイクル数 8
2.9.1 ROM/RAMサイズ 8
2.9.2 スタックサイズ 9
2.9.3 処理サイクル数 10
2.10 FITモジュールの追加方法 11

3. デモプロジェクト 12
3.1 aes_demo_65n_2m 12
3.2 ワークスペースにデモを追加する 12

4. 付録 13
4.1 動作確認環境 13

5. 参考ドキュメント 14

ホームページとサポート窓口 15

改訂記録 16

1. 概要

1.1 AES FIT ライブラリとは

本ライブラリは、API として、プロジェクトに組み込んで使用します。本ライブラリの組み込み方については、「2.9 FITモジュールの追加方法」を参照してください。

1.2 AES FIT ライブラリの概要

パッケージに格納されているユーザーズマニュアル (R20UW0068JJ0200) を参照してください。

1.3 API 関数

RX 用 AES ライブラリは以下の関数をサポートしています。

各 API 関数の詳細については、ユーザマニュアル (R20UW0068JJ0200) を参照してください。

表 1-1 AES ライブラリの API 関数

API	Outline
R_Aes_128_Keysch	AES 128-bit 拡大鍵生成関数
R_Aes_128_Ecbenc	AES 128-bit 暗号化関数 (ECB モード)
R_Aes_128_Ecbdec	AES 128-bit 復号関数 (ECB モード)
R_Aes_128_Cbcenc	AES 128-bit 暗号関数 (CBC モード)
R_Aes_128_Cbcdec	AES 128-bit 復号関数 (CBC モード)
R_Aes_256_Keysch	AES 256-bit 拡大鍵生成関数
R_Aes_256_Ecbenc	AES 256-bit 暗号化関数 (ECB モード)
R_Aes_256_Ecbdec	AES 256-bit 復号関数 (ECB モード)
R_Aes_256_Cbcenc	AES 256-bit 暗号化関数 (CBC モード)
R_Aes_256_Cbcdec	AES 256-bit 復号関数 (CBC モード)

RX 用 GCM ライブラリは以下の関数をサポートしています。

表 1-2 GCM ライブラリの API 関数

API	Outline
R_gcm_enc	GCM 暗号化関数
R_gcm_dec	GCM 復号関数
R_gcm_enc_start	GCM 暗号化開始関数
R_gcm_dec_start	GCM 復号開始関数
R_gcm_repeat	GCM 処理継続関数

1.4 バージョン情報

AES では、R_aes_version 変数に文字列でバージョン情報を格納しています。以下の extern 宣言により、この変数にアクセスすることが出来ます。また、本製品のライブラリに格納されているデータは以下の通りです。

```
extern const mw_version_t R_aes_version;
```

また、GCM では、R_gcm_version 変数に文字列でバージョン情報を格納しています。以下の extern 宣言により、この変数にアクセスすることが出来ます。また、本製品のライブラリに格納されているデータは以下の通りです。

```
extern const mw_version_t R_gcm_version;
```

1.5　AES FIT ライブラリの製品構成

1.5.1　アプリケーションノート構成

本アプリケーションノートは、以下の表 1-3 アプリケーションノート構成となっています。

表 1-3 アプリケーションノート構成

ファイル / フォルダ(**太字**)名	内容
r20an0044jj0108-rx-aes.pdf	AES FIT ライブラリ アプリケーションノート (日本語)
r20an0044ej0108-rx-aes.pdf	AES FIT ライブラリ　アプリケーションノート(英語)
r20uw0068jj0200-aes.pdf	AES FIT ライブラリ　ユーザーマニュアル (日本語)
r20uw0068ej0200-aes.pdf	AES FIT ライブラリ　ユーザーマニュアル (英語)
FITDemos	FIT モジュールデモプログラムフォルダ
aes_demo_65n_2m	AES FIT モジュール モプログラム
FITModules	FIT モジュールフォルダ
r_aes_rx_v1.08.zip	AES FIT Module
r_aes_rx_v1.08.xml	AES FIT Module XML ファイル
r_aes_rx_v1.08_extend.mdf	AES FIT Module MDF ファイル

1.5.2　ファイル構成

r_aes_rx_v1.08.zip を解凍したフォルダには、以下の表 1-4 ファイル構成のファイルが含まれます。

表 1-4 ファイル構成

ファイル / フォルダ(**太字**)名	内容
r_aes_rx	FIT Module フォルダ
doc	ドキュメントフォルダ
en	ドキュメントフォルダ(英語)
r20an0044ej0108-rx-aes.pdf	AES FIT ライブラリ アプリケーションノート (英語)
r20uw0068ej0200-aes.pdf	AES FIT ライブラリ　ユーザーマニュアル (英語)
ja	ドキュメントフォルダ(日本語)
r20an0044jj0108-rx-aes.pdf	AES FIT ライブラリ アプリケーションノート (日本語)
r20uw0068jj0200-aes.pdf	AES FIT ライブラリ　ユーザーマニュアル (日本語)
ref	参照フォルダ
r_aes_config_reference.h	コンフィグ参照ファイル
src	ソースコードフォルダ
aes	AES ソースコードフォルダ
aes128Ecb_small.c	128-bit ECB モード AES の API 関数の定義部
aes128Cbc_small.c	128-bit CBC モード AES の API 関数の定義部
aes256Ecb_small.c	256-bit ECB モード AES の API 関数の定義部
aes256Cbc_small.c	256-bit CBC モード AES の API 関数の定義部
aes128Ecb_big.c	128-bit ECB モード AES の API 関数の定義部
aes128Cbc_big.c	128-bit CBC モード AES の API 関数の定義部
aes256Ecb_big.c	256-bit ECB モード AES の API 関数の定義部
aes256Cbc_big.c	256-bit CBC モード AES の API 関数の定義部
aes128.h	128-bit CBC モード AES のコア部
aes256.h	256-bit CBC モード AES のコア部

r_aesEcb.h	ECB モード AES のコア部
r_aes_version.c	AES ライブラリのバージョン情報定義
r_aesSbox.h	AES 用 SBOX テーブルの定義
r_aes_development.h	AES ライブラリ関数名定義マクロヘッダファイル
r_aes.h	AES ライブラリ関数名定義ヘッダファイル
gcm	GCM ソースコードフォルダ
r_gcm.c	GCM ライブラリ本体
r_gcm_version.c	GCM ライブラリのバージョン情報定義
r_gcm.h	GCM ライブラリヘッダファイル
r_gcm_driver.c	GCM ドライバ本体
r_mw_version.h	バージョン情報ヘッダファイル
r_stdint.h	型定義ヘッダファイル
r_aes_rx_if.h	API インタフェース定義ヘッダファイル
readme.txt	readme ファイル
r_config	コンフィグファイルフォルダ
r_aes_config.h	コンフィグファイル(デフォルト設定)

2. API 情報

2.1 ハードウェアの要求

ハードウェアの要求はありません。

2.2 ソフトウェアの要求

ソフトウェアの要求はありません。

2.3 制限事項

ソフトウェアに関する制限事項はありません。

2.4 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。

2.5 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_aes_rx_if.h に記載してあります。

2.6 整数型

このプロジェクトは、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 ビルド時の注意事項

本 FIT モジュールを以下の FIT モジュールのいずれかと一緒に利用する場合、プロジェクトの設定によってビルドエラーが発生する場合があります。

これらの FIT モジュールと本 FIT モジュールを一緒に利用する際は、本 FIT モジュールが先にビルドされるようにビルド順を調整してください。

FIT モジュール	省略名
JPEG デコーダモジュール	r_jpegd_rx
JPEG エンコーダモジュール	r_jpege_rx
組み込み用 TCP/IP M3S-T4-Tiny モジュール	r_t4_rx
音声録音・再生システム(独自 ADPCM コーデック) M3S-S2-Tiny モジュール	r_s2_rx

2.8 ライブラリ関数の使用方法

2.8.1 処理性能 vs. コードサイズ

RX 用 AES ライブラリは、プログラムのコードサイズの最適化により処理性能を重視する（速くする）場合（以下、処理性能重視）と、処理性能の向上よりプログラムのコードサイズを重視する（小さくする）場合（以下、サイズ重視）の 2 種類のアプローチがあります。

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_aes_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 option in r_aes_config.h	
定義	説明
#define AES_CFG_BUILD_OPTION ※デフォルト値は「”0” ：処理性能重視」が設定されます。	処理性能重視とサイズ重視のいずれかを選択できます。 0： SPEED (処理性能重視) 1： SIZE (サイズ重視)

上記、オプション設定により以下のマクロ定義が有効になります。

マクロ名	選択される実装
__COMPILE_EMPHASIS_SPEED__	処理性能重視
__COMPILE_EMPHASIS_SIZE__	サイズ重視

2.9　AES ライブラリ ROM / RAM / Stack Size / 処理サイクル数

以下の最適化オプションを指定してビルドした際の各種サイズや処理サイクルを参考として記します。

CCRX：最適化レベル 2

GCC　：-O2

IAR　：レベル高 (サイズ)

2.9.1　ROM/RAM サイズ

関数名	Little or Big Endian	実装	ROM size [byte]			RAM size [byte]		
			CCRX	GCC	IAR	CCRX	GCC	IAR
R_Aes_128_Keysch	Little	処理性能重視	221	248	238	0	0	0
		サイズ重視	170	256	174			
R_Aes_128_Ecbenc	Little	処理性能重視	2389	2848	2420	0	0	0
		サイズ重視	331	1160	578			
R_Aes_128_Ecbdec	Little	処理性能重視	2535	3184	2535	0	0	0
		サイズ重視	481	1808	1280			
R_Aes_128_Cbcenc	Little	処理性能重視	454	544	525	0	0	0
		サイズ重視	240	552	238	0	0	0
R_Aes_128_Cbcdec	Little	処理性能重視	558	688	614	0	0	0
		サイズ重視	308	688	298			
R_Aes_256_Keysch	Little	処理性能重視	469	560	499	0	0	0
		サイズ重視	405	576	411			
R_Aes_256_Ecbenc	Little	処理性能重視	3201	3832	3234	0	0	0
		サイズ重視	411	1368	648			
R_Aes_256_Ecbdec	Little	処理性能重視	3347	4232	3347	0	0	0
		サイズ重視	562	2096	1350			
R_Aes_256_Cbcenc	Little	処理性能重視	454	544	525	0	0	0
		サイズ重視	240	552	238	0	0	0
R_Aes_256_Cbcdec	Little	処理性能重視	558	688	614	0	0	0
		サイズ重視	308	688	298			
R_gcm_enc	Little	-	487	552	443	0	0	0
R_gcm_dec	Little	-	512	560	465	0	0	0
R_gcm_enc_start	Little	-	103	112	100	0	0	0
R_gcm_dec_start	Little	-	103	112	100	0	0	0
R_gcm_repeat	Little	-	989	1048	851	0	0	0

【注1】サンプルプログラムを実行した場合の値です。ユーザがユーザ定義関数の実装を変更した場合は、スタックサイズが変化します。

【注2】それぞれの CBC モードの enc/dec 関数は ECB モードの enc/dec 関数を呼び出すため、2 関数の ROM サイズの合計が必要（例えば、R_Aes_128_Cbcenc 関数は R_Aes_128_Ecbenc 関数を呼び出すため、R_Aes_128_Cbcenc 関数を使用する際は、両者の関数の合計の ROM サイズが必要）。

【注3】「-」は性能重視とサイズ重視でコードに変更はありません。

2.9.2　スタックサイズ

API	Little or Big Endian	処理性能重視 or サイズ重視	Stack Size [byte]		
			CCRX	GCC	IAR
R_Aes_128_Keysch	Little	処理性能重視	28	32	24
		サイズ重視	12	32	12
R_Aes_128_Ecbenc	Little	処理性能重視	84	108	72
		サイズ重視	104	108	56
R_Aes_128_Ecbdec	Little	処理性能重視	276	464	244
		サイズ重視	276	308	228
R_Aes_128_Cbcenc	Little	処理性能重視	68	112	60
		サイズ重視	80	116	60
R_Aes_128_Cbcdec	Little	処理性能重視	96	136	76
		サイズ重視	108	136	76
R_Aes_256_Keysch	Little	処理性能重視	32	32	24
		サイズ重視	16	36	16
R_Aes_256_Ecbenc	Little	処理性能重視	84	120	72
		サイズ重視	120	104	56
R_Aes_256_Ecbdec	Little	処理性能重視	340	592	308
		サイズ重視	356	376	292
R_Aes_256_Cbcenc	Little	処理性能重視	68	112	60
		サイズ重視	80	116	60
R_Aes_256_Cbcdec	Little	処理性能重視	96	136	76
		サイズ重視	108	136	76
R_gcm_enc	Little	-	260	252	232
R_gcm_dec	Little	-	232	256	232
R_gcm_enc_start	Little	-	24	60	28
R_gcm_dec_start	Little	-	24	60	28
R_gcm_repeat	Little	-	176	192	64

【注 1】「-」は性能重視とサイズ重視でコードに変更はありません。

2.9.3　処理サイクル数

AES ライブラリの処理サイクル数

測定条件は、CC-RX 最適化レベル 2 と処理性能重視です。

API	Little/Big Endian	性能（単位：サイクル）		
		1 block	2 block	3 block
R_Aes_128_Keysch	Little	752		
	Big	768		
R_Aes_256_Keysch	Little	1060		
	Big	908		
R_Aes_128_Ecbenc	Little	1692	3274	4858
	Big	1704	3292	4882
R_Aes_128_Ecbdec	Little	3116	4564	6012
	Big	3224	4670	6116
R_Aes_128_Cbcenc	Little	1920	3656	5548
	Big	1930	3666	5404
R_Aes_128_Cbcdec	Little	3448	6684	10068
	Big	3554	7044	10388
R_Aes_256_Ecbenc	Little	2244	4386	6528
	Big	2248	4390	6530
R_Aes_256_Ecbdec	Little	4294	6412	8222
	Big	4296	6260	8224
R_Aes_256_Cbcenc	Little	2476	4768	7060
	Big	2478	4760	7042
R_Aes_256_Cbcdec	Little	4622	9036	13602
	Big	4626	9040	13602

GCM ライブラリの処理サイクル数

測定条件は、CC-RX 最適化レベル 2 と処理性能重視です。

API	Little/Big Endian	鍵	性能（単位：サイクル）
R_gcm_enc	Little	128 bit	51288
		256 bit	197262
	Big	128 bit	51826
		256 bit	198228
R_gcm_dec	Little	128 bit	52040
		256 bit	197892
	Big	128 bit	52460
		256 bit	197738

【注】 1. atag = 16 byte、 ivec = 12 byte、 aad = 1 ブロック で計測しています。
2. 入力する平文/暗号文の値によって速度は変動します。

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) e² studio 上で Smart Configurator を使用して FIT モジュールを追加する場合
e² 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 e² studio スマート・コンフィグレータ ユーザガイド (R20AN0451)」を参照してください。

(2) e² studio 上で FIT Configurator を使用して FIT モジュールを追加する場合
e² 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 e² 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 e2 studio スマート・コンフィグレータ ユーザガイド (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. 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、本 FIT モジュールとそのモジュールに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あります。

3.1 aes_demo_65n_2m

aes_demo_65n_2m は、AES ライブラリの API の使い方を示しています。このデモプロジェクトでは、AES128-CBC、AES128-ECB、AES256-CBC、AES256-ECB、AES128-GCM、AES256-GCM アルゴリズムを用いた、暗復号処理を行います。

3.2 ワークスペースにデモを追加する

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圧縮ファイルの中の 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」を選択します。
- 「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ます。
- 「参照」ボタンをクリックして、FITDemos サブディレクトリを開きます。
- 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。

上記処理で、ワークスペースにデモプロジェクトの追加ができます。

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.1.08)

項目	内容
総合開発環境	ルネサスエレクトロニクス製 e2 studio 2025-01 IAR Embedded Workbench for Renesas RX 5.10.1
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 V3.07.00 コンパイルオプション:総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 8.3.0.202411 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 リンクオプション : 「Optimize size (サイズ最適化) (-Os)」を使用する場合、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 -Wl,--no-gc-sections これは、FIT 周辺機器モジュール内で宣言されている割り込み関数をリンカが誤って破棄（discard）することを回避（work around）するための対策です。
	IAR C/C++ Compiler for Renesas RX version 5.10.1 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.1.08
使用ボード	Target Board for RX65N Target Board for RX130 Renesas Envision Kit for RX72N

5. 参考ドキュメント

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。

なし

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
1.08	2025.3.20	-	動作確認環境更新
		-	プログラムの免責事項を変更しました
		-	2.7 ビルド時の注意事項を追加しました
1.07	2022.10.31	-	動作確認環境更新
1.06	2022.08.10	-	FIT 化に伴い、タイトル修正と FIT 関連情報を追加しました。 ライブラリの提供形態を Lib.形式から C ソースに変更しました。
1.04	2013.05.30	-	バージョン表記を V.1.04 に修正しました。 ユーザーズマニュアルを rev 1.05 から rev 1.08 に更新しました。 Introduction に GCM の説明を追加しました
1.03	2012.12.27	-	GCM サンプルコードを修正しました。 NIST テストベクタ CAVS 10.1 に対応しました。 標準型を使用するように変更しました。 ライブラリバージョン情報の持ち方を修正しました。 ユーザーズマニュアルを rev 1.03 から rev 1.05 に更新しました。
1.02	2012.04.16	-	RX200 シリーズをサポートマイコンに追加しました。 GCM のソースコードをサンプルに収録しました。 処理サイクル数を追加しました。
1.01	2011.05.06	-	GCM サポートを追加しました。
1.00	2011.02.18	-	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. 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。

© 2025 Renesas Electronics Corporation. All rights reserved.