

## RX ファミリ

R20AN0157JJ0106

Rev.1.06

Mar.20.2025

## SHA 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 SHA-1/SHA-256/SHA-384 ライブラリ (以下 SHA FIT ライブラリ) を導入するための情報を記します。

SHA FIT ライブラリは SHA-1/SHA-256/SHA-384 のハッシュ演算処理を 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。SHA 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。SHA FIT 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 (R20UW0101JJ0200) 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 SHA FIT ライブラリとは	3
1.2 SHA 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 SHA FIT ライブラリの製品構成	3
1.5.1 アプリケーションノート構成	3
1.5.2 ファイル構成	4
2. API 情報	5
2.1 ハードウェアの要求	5
2.2 ソフトウェアの要求	5
2.3 制限事項	5
2.4 サポートされているツールチェーン	5
2.5 ヘッダファイル	5
2.6 整数型	5
2.7 ビルド時の注意事項	5
2.8 ライブラリ関数の使用方法	5
2.8.1 処理性能 vs. コードサイズ	5
2.9 SHA ライブラリ ROM / RAM / Stack Size / 処理サイクル数	7
2.9.1 ROM/RAM サイズ	7
2.9.2 スタックサイズ	7
2.9.3 処理サイクル数	7
2.10 FIT モジュールの追加方法	8
3. デモプロジェクト	9
3.1 sha_demo_65n_2m	9
3.2 ワークスペースにデモを追加する	9
4. 付録	10
4.1 動作確認環境	10
5. 参考ドキュメント	11
ホームページとサポート窓口	12
改訂記録	13

## 1. 概要

### 1.1 SHA 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 SHA 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 (R20UW0101JJ0200) を参照してください。

### 1.3 API 関数

SHA 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 (R20UW0101JJ0200) を参照してください。

表 1-1 SHA ライブラリの API 関数

API	Outline
R_Sha1_HashDigest	SHA-1 ハッシュ値の演算
R_Sha256_HashDigest	SHA-256 ハッシュ値の演算
R_Sha384_HashDigest	SHA-384 ハッシュ値の演算

### 1.4 バージョン情報

SHA-1/SHA-256/SHA-384 ライブラリでは、R\_sha\_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。

```
extern const char R_sha_version[];
```

### 1.5 SHA 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 1-2 アプリケーションノート構成となっています。

表 1-2 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
r20an0157jj0106-rx-sha.pdf	SHA FIT ライブラリ アプリケーションノート (日本語)
r20an0157ej0106-rx-sha.pdf	SHA FIT ライブラリ アプリケーションノート(英語)
r20uw0101jj0200-sha.pdf	SHA FIT ライブラリ ユーザーマニュアル (日本語)
r20uw0101ej0200-sha.pdf	SHA FIT ライブラリ ユーザーマニュアル (英語)
<b>FITDemos</b>	FIT モジュールデモプログラムフォルダ
sha_demo_65n_2m.zip	SHA FIT モジュール モプログラム
<b>FITModules</b>	FIT モジュールフォルダ
r_sha_rx_v1.06.zip	SHA FIT Module
r_sha_rx_v1.06.xml	SHA FIT Module XML ファイル
r_sha_rx_v1.06_extend.mdf	SHA FIT Module MDF ファイル

## 1.5.2 ファイル構成

r\_sha\_rx\_v1.06.zip を解凍したフォルダには、以下の表 1-3 ファイル構成のファイルが含まれます。

表 1-3 ファイル構成

ファイル / フォルダ(太字)名	内容
<b>r_sha_rx</b>	FIT Module フォルダ
<b>doc</b>	ドキュメントフォルダ
<b>en</b>	ドキュメントフォルダ(英語)
r20an0157ej0106-rx-sha.pdf	SHA FIT ライブラリ アプリケーションノート (英語)
r20uw0101ej0200-sha.pdf	SHA FIT ライブラリ ユーザーマニュアル (英語)
<b>ja</b>	ドキュメントフォルダ(日本語)
r20an0157jj0106-rx-sha.pdf	SHA FIT ライブラリ アプリケーションノート (日本
r20uw0101jj0200-sha.pdf	SHA FIT ライブラリ ユーザーマニュアル (日本語)
<b>ref</b>	参照フォルダ
r_sha_config_reference.h	コンフィグ参照ファイル
<b>src</b>	ソースコードフォルダ
sha1if.c	SHA-1 の API 関数の定義部
sha256if.c	SHA-256 の API 関数の定義部
sha384if.c	SHA-384 の API 関数の定義部(RL78 サポート外)
shaif.h	API 関数のコア部
sha1.c	SHA-1 計算のコア部
sha256.c	SHA-256 計算のコア部
sha512.c	SHA-384 / SHA-512 計算のコア部(RL78 サポート外)
r_sha_version.c	SHA ライブラリのバージョン情報定義
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_sha_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
<b>r_config</b>	コンフィグファイルフォルダ
r_sha_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_sha_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 モジュールと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	<code>r_jpegd_rx</code>
JPEG エンコーダモジュール	<code>r_jpege_rx</code>
組み込み用 TCP/IP M3S-T4-Tiny モジュール	<code>r_t4_rx</code>
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	<code>r_s2_rx</code>

## 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

ハッシュ関数の実装には、プログラムのコードサイズの最適化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_sha_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 <code>r_sha_config.h</code>	
定義	説明
<code>#define SHA_CFG_BUILD_OPTION</code> ※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	対象 API
__COMPILE_EMPHASIS_SPEED__	処理性能重視	R_Sha256_HashDigest / R_Sha384_HashDigest
__COMPILE_EMPHASIS_SIZE__	サイズ重視	R_Sha256_HashDigest / R_Sha384_HashDigest

## 2.9 SHA 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。

CCR<sub>X</sub> : 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

API	Little/Big Endian	処理性能重視 or サイズ重視	ROM size [byte]			RAM size [byte]		
			CCR <sub>X</sub>	GCC	IAR	CCR <sub>X</sub>	GCC	IAR
R_Sha1_HashDigest	Little/Big	-	421	576	483	0	0	0
R_Sha256_HashDigest	Little/Big	-	444	592	499	0	0	0
R_Sha384_HashDigest	Little/Big	-	397	552	469	0	0	0

【注】「-」は処理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/Big Endian	処理性能重視 or サイズ重視	Stack Size [byte]		
			CCR <sub>X</sub>	GCC	IAR
R_Sha1_HashDigest	Little/Big	-	32	36	28
R_Sha256_HashDigest	Little/Big	-	32	36	28
R_Sha384_HashDigest	Little/Big	-	32	36	28

【注】「-」は処理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

SHA 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。

入力メッセージ長 [byte]	Little/Big Endian	SHA-1 [cycles]	SHA-256 [cycles]	SHA-384 [cycles]
0	Little	4938	4628	16544
	Big	4954	4456	16512
64	Little	9358	8668	15824
	Big	9382	8316	15802
128	Little	13636	12556	30624
	Big	13666	12182	30576
192	Little	17768	16256	30456
	Big	17950	15902	30414
256	Little	21902	19970	44588
	Big	21934	19616	44526

【注】 メッセージは 1 回で入力し、内部のパディング処理を含む処理時間です。

## 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 sha\_demo\_65n\_2m

sha\_demo\_65n\_2m は、SHA 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、SHA-1、SHA-256、SHA-384 アルゴリズムを用いた、HASH 演算を行います。

#### 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.1.06)

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クが誤って破棄 (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
モジュールのバージョン	Rev.1.06
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
1.06	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
1.05	2022.10.31	-	動作確認環境更新
1.04	2022.07.16	-	FIT 化に伴い、タイトルと 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。 SHA-384 対応追加
1.00	2012.04.16	-	初版作成

## 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。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た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しします。
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](http://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/](http://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。

