ルネサスマイクロコンピュータ
SHA ハッシュ関数ライブラリ: ユーザーズマニュアル

要旨

本ライブラリは、ルネサスマイクロコンピュータ用の SHA-1/SHA-256/SHA-384 のハッシュ値を演算するソフトウェアライブラリです (以下 SHA ハッシュ関数ライブラリとします) 。本資料は SHA ハッシュ関数ライブラリの関数リファレンスを示します。マイコンに依存した情報については対応マイコン毎に用意している「導入ガイド」を合わせて参照してください。

■参考文献

以下の文書は、SHA ハッシュ関数ライブラリに関連した標準規格文書です。

- FIPS PUB 180-4, SECURE HASH STANDARD

SHA-1, SHA-256, SHA-384 は、米国のアメリカ合州国連邦情報処理標準として国立標準技術研究所によって公表された暗号ハッシュ関数です。 任意のデータ長 (< 2^64 bits) のデジタルデータを入力値(＝メッセージ) とし、160bits (SHA-1), 256bits (SHA-256) , 384bits(SHA-384)のハッシュ値 (メッセージダイジェスト) を出力します。

SHA-1, SHA-256, SHA-384 アルゴリズムは、以下の計算が不可能なため安全であると言われています。

- 出力値から入力値を推測すること
- 同じ出力値を持つ異なる入力値を見つけること

ただし、昨今のコンピュータ技術の進歩により、これらが安全ではない可能性も高まり、より強固なハッシュアルゴリズムも考案されています。

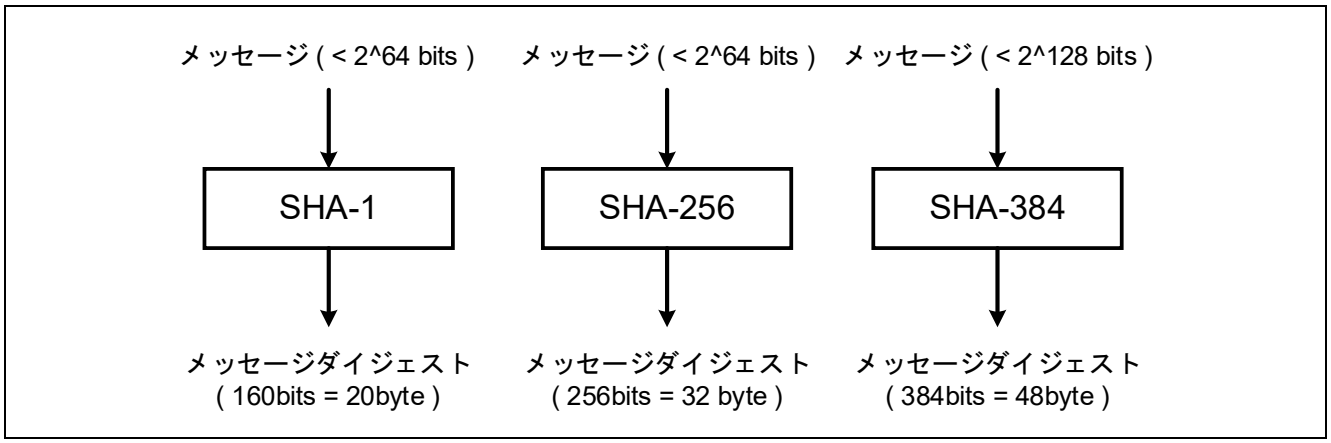

図 1-1 イメージ

SHA ハッシュ関数ライブラリでは入力可能なメッセージ長の上限が標準規格と異なります。

動作確認デバイス

ルネサスマイクロコンピュータ

目次

1. ライブラリ仕様

SHA ハッシュ関数ライブラリの仕様について説明します。

表 1-1　仕様一覧

項目	内容
入力メッセージ	byte 単位。合計で 2^29 byte ((2^32 bits) / 8) 未満であること。1 回または任意の回数に分けて入力することが可能。(3 章制御フラグを参照)
出力メッセージダイジェスト	160 bits (SHA-1), 256 bits (SHA-256), 384 bits(SHA-384)
ブロックサイズ	64 byte (SHA-1, SHA-256), 128 byte (SHA-384)
パディング処理	ライブラリ内部で処理します。(処理を抑制することも可能)

2. データ型定義

データタイプライブラリが使用するデータ型です。

表 2-1　データ型

Datatype	Typedef
unsigned char	uint8_t
unsigned short	uint16_t
unsigned long	uint32_t
signed char	int8_t
signed short	int16_t
signed long	int32_t

・RL78 の場合
struct{
uint32_t work[51 + 24];
}
 	
・RL78 以外の場合
struct{
uint32_t work[51 + 2 * 88];
}
	R_sha_handle

3.　制御フラグ

メッセージは、制御フラグにより 1 回または任意の回数に分けて入力することができます。

1 回でメッセージを入力しメッセージダイジェストを得る場合は、"R_SHA_INIT"と"R_SHA_FINISH"の両方を指定します。

任意の回数に分けてメッセージを入力する場合は、初回は"R_SHA_INIT"、最後は"R_SHA_FINISH"を指定します。メッセージの中間部分を入力するときは"R_SHA_ADD"を指定します。

表 3-1　制御フラグ

シンボル	値	説明
R_SHA_ADD	0	追加（途中の入力）
R_SHA_INIT	1	初期化（最初の入力）
R_SHA_FINISH	2	完了（最後の入力）
R_SHA_NOPADDING	4	関数内部でパディング処理を行わない

"R_SHA_NOPADDING"はライブラリ内部のパディング処理を抑制するフラグで、"R_SHA_FINISH"と同時に指定する必要があります。この場合、上位アプリケーションでメッセージのパディング処理を行い、合計 64×n byte のデータとして入力する必要があります。入力メッセージをビット単位で行いたい場合に使用します。

3.1　組み合わせ

制御フラグは以下の組み合わせで使用することが可能です。その他の組み合わせや未使用部分 のビットをオンにした場合の動作は保証しません。

表 3-2　組み合わせ一覧

組み合わせ	値(2 進数)
R_SHA_ADD	0 (000)
R_SHA_INIT	1 (001)
R_SHA_FINISH	2 (010)
R_SHA_INIT | R_SHA_FINISH	3 (011)
R_SHA_FINISH | R_SHA_NOPADDING	6 (110)
R_SHA_INIT | R_SHA_FINISH | R_SHA_NOPADDING	7 (111)

3.2　使用例

3.2.1　1 回で入力

メッセージを 1 回で入力する場合は、以下のような関数呼び出しを行います。

```
uint8_t mdat[] = "abcdef";
uint8_t hdat[20];
R_sha_handle work;

R_Sha1_HashDigest(mdat, hdat, 6,
                  (R_SHA_INIT | R_SHA_FINISH), (void *)&work);
```

3.2.2　2 回以上に分割して入力

メッセージを 2 回に分けて入力する場合は、以下のような関数呼び出しを行います。

```
uint8_t mdat1[] = "abc";
uint8_t mdat2[] = "def";
uint8_t hdat[20];
R_sha_handle work;

R_Sha1_HashDigest(mdat1, hdat, 3, R_SHA_INIT, (void *)&work);
R_Sha1_HashDigest(mdat2, hdat, 3, R_SHA_FINISH, (void *)&work);
```

メッセージを 3 回に分けて入力する場合は、以下のような関数呼び出しを行います。

```
uint8_t mdat1[] = "ab";
uint8_t mdat2[] = "cd";
uint8_t mdat3[] = "ef";
uint8_t hdat[20];
R_sha_handle work;

R_Sha1_HashDigest(mdat1, hdat, 2, R_SHA_INIT, (void *)&work);
R_Sha1_HashDigest(mdat2, hdat, 2, R_SHA_ADD, (void *)&work);
R_Sha1_HashDigest(mdat3, hdat, 2, R_SHA_FINISH, (void *)&work);
```

3.2.3　パディング処理を行わない

パディング済みのメッセージを入力する場合は、以下のような関数呼び出しを行います。

```
uint8_t mdat[] = {          /* 1 zero bits */
    0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01
};
uint8_t hdat[20];
R_sha_handle work;

R_Sha1_HashDigest(mdat, hdat, 64,
                  (R_SHA_INIT | R_SHA_FINISH | R_SHA_NOPADDING),
                  (void *)&work);
```

パディング済みのメッセージを分割して入力することも可能です。パディング済みのメッセージを 2 回に分けて入力する場合は、以下のような関数呼び出しを行います。

```
uint8_t mdat1[] = {         /* 1 zero bits */
    0x40
};
uint8_t mdat2[] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01
};
uint8_t hdat[20];
R_sha_handle work;

R_Sha1_HashDigest(mdat1, hdat, 1, R_SHA_INIT , (void *)&work);
R_Sha1_HashDigest(mdat2, hdat, 63,
                  (R_SHA_FINISH | R_SHA_NOPADDING), (void *)&work);
```

4. エラーコード

関数は以下のエラーコードを返します。

表 4-1　エラーコード

シンボル	値	説明
R_PROCESS_COMPLETE	0	正常終了
R_SHA_ERROR_POINTER	-1	ポインタ hadt または work が NULL です
R_SHA_ERROR_FLAG	-2	flag の指定が間違っています
R_SHA_ERROR_LENGTH	-3	ポインタ mdat が NULL 且つ len>0 を指定しています

5. 関数仕様

ライブラリの関数仕様について説明します。

ライブラリ関数詳細の見方は以下の通りです。

書式

関数の呼び出し形式を示します。#include "ヘッダファイル"で示すヘッダファイルは、この関数の実行に必要なヘッダファイルです。必ずインクルードしてください。

引数

関数の引数を示します。"I/O"には引数がそれぞれ入力値、出力値であることを示します。"説明"には"引数名"についての説明を示します。

戻り値

関数の戻り値を示します。"説明"には戻り値の値についての説明を示します。

説明

関数の仕様を示します。

注意事項

関数を使用する際の注意事項を示します。

5.1　R_Sha1_HashDigest

SHA-1 ハッシュ値演算

書式

```
#include "r_sha.h"

int8_t R_Sha1_HashDigest (
                        const uint8_t *mdat ,
                        uint8_t *hdat ,
                        uint16_t len ,
                        uint8_t flag ,
                        void *work    );
```

引数

引数名	I/O	説明
mdat	I	メッセージの格納アドレス
hdat	O	メッセージダイジェストの格納アドレス
len	I	メッセージのサイズ　(単位：byte)
flag	I	制御フラグ
work	I/O	ワークエリアの指定

戻り値

4 章エラーコードを参照してください。

説明

この関数は、SHA-1 アルゴリズムで定められたメッセージダイジェストを演算します。

アプリケーションは、本関数の第 1 引数「mdat」にメッセージのアドレス、第 2 引数「hash」にメッセージダイジェストのアドレスを指定します。第 3 引数「len」に「mdat」データ長を指定します。

第 4 引数「flag」に制御フラグを指定します。制御フラグの内容は 3 章制御フラグを参照してください。

第 5 引数「work」にユーザが確保したワークエリアを指定します。メッセージダイジェストの演算が終わるまで内容を保持する必要があります。

メッセージダイジェストは、flag に"R_SHA_FINISH"を指定したとき格納します。

注意事項

無し。

5.2　R_Sha256_HashDigest

SHA-256 ハッシュ値演算

書式

```
#include "r_sha.h"

int8_t R_Sha256_HashDigest (
                        const uint8_t *mdat ,
                        uint8_t *hdat ,
                        uint16_t len ,
                        uint8_t flag ,
                        void *work    );
```

引数

引数名	I/O	説明
mdat	I	メッセージの格納アドレス
hdat	O	メッセージダイジェストの格納アドレス
len	I	メッセージのサイズ　(単位：byte)
flag	I	制御フラグ
work	I/O	ワークエリアの指定

戻り値

4 章エラーコードを参照してください。

説明

この関数は、SHA-256 アルゴリズムで定められたメッセージダイジェストを演算します。

アプリケーションは、本関数の第 1 引数「mdat」にメッセージのアドレス、第 2 引数「hash」にメッセージダイジェストのアドレスを指定します。第 3 引数「len」に「mdat」データ長を指定します。

第 4 引数「flag」に制御フラグを指定します。制御フラグの内容は 3 章制御フラグを参照してください。

第 5 引数「work」にユーザが確保したワークエリアを指定します。メッセージダイジェストの演算が終わるまで内容を保持する必要があります。

メッセージダイジェストは、flag に"R_SHA_FINISH"を指定したとき格納します。

注意事項

無し。

5.3　R_Sha384_HashDigest

SHA-384 ハッシュ値演算

書式

```
#include "r_sha.h"

int8_t R_Sha384_HashDigest (
                        const uint8_t *mdat ,
                        uint8_t *hdat ,
                        uint16_t len ,
                        uint8_t flag ,
                        void *work    );
```

引数

引数名	I/O	説明
mdat	I	メッセージの格納アドレス
hdat	O	メッセージダイジェストの格納アドレス
len	I	メッセージのサイズ　(単位：byte)
flag	I	制御フラグ
work	I/O	ワークエリアの指定

戻り値

4 章エラーコードを参照してください。

説明

この関数は、SHA-384 アルゴリズムで定められたメッセージダイジェストを演算します。

アプリケーションは、本関数の第 1 引数「mdat」にメッセージのアドレス、第 2 引数「hash」にメッセージダイジェストのアドレスを指定します。第 3 引数「len」に「mdat」データ長を指定します。

第 4 引数「flag」に制御フラグを指定します。制御フラグの内容は 3 章制御フラグを参照してください。

第 5 引数「work」にユーザが確保したワークエリアを指定します。メッセージダイジェストの演算が終わるまで内容を保持する必要があります。

メッセージダイジェストは、flag に"R_SHA_FINISH"を指定したとき格納します。

注意事項

無し。

6. サンプルプログラム

SHA ハッシュ関数ライブラリのサンプルプログラムについて説明します。

サンプルプログラムは、SHA-1 ハッシュ値を演算し期待値と一致するか確認します。

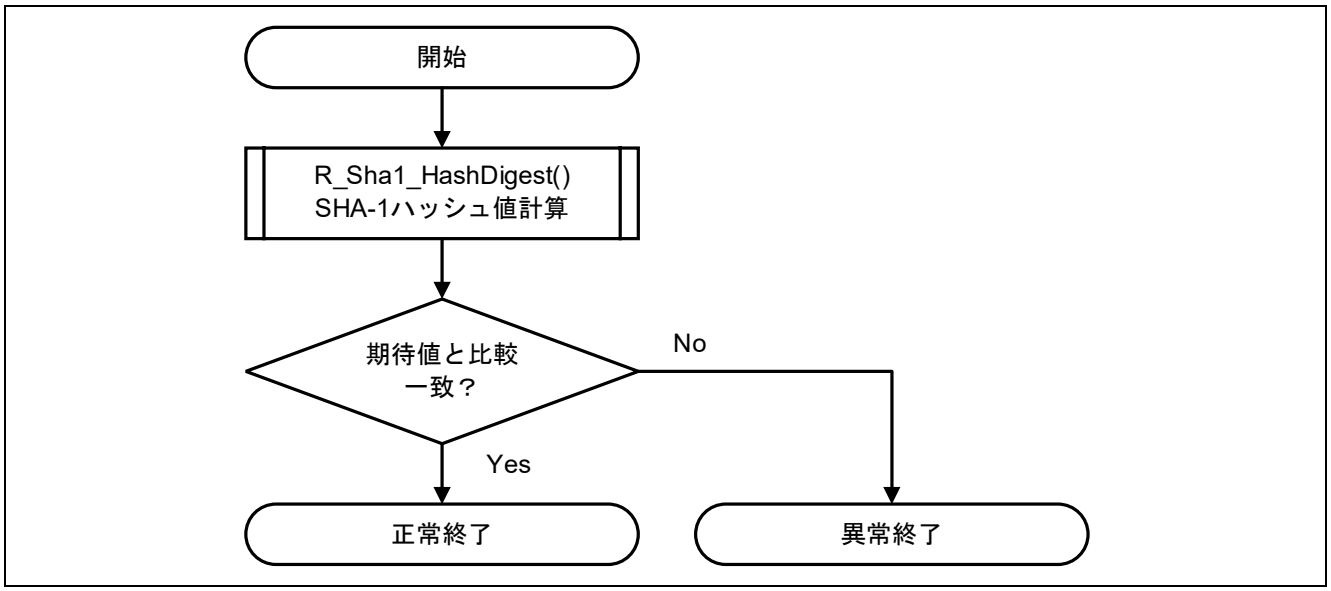

図 6-1 サンプルプログラムのフロー

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	Aug.1.2014	−	初版発行
2.00	Apr.23.2021	−	SHA-384 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. 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。