

(注1)本資料は英語版を翻訳した参考資料です。内容に相違がある場合には英語版を優先します。資料によっては英語版のバージョンが更新され、内容が変わっている場合があります。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

(注2)本資料の第6章まで(要旨除く)の日本語訳は、「[「Synergy™ Software Package \(SSP\) v1.5.0 ユーザーズマニュアル モジュール概要編\(参考資料\)」](#)」の第4章「モジュールの概要」に掲載されていますのでそちらを参照ください。

要旨(Introduction)

本モジュールガイドは、USBX™ 通信フレームワークモジュールを効果的に使用してシステムが開発できるようになることを目的としています。このモジュールガイドを習得することで、開発システムへのモジュールの追加とターゲットアプリケーション向けの正確な設定(configuration)ができ、さらに付属のアプリケーションプロジェクトコードを参照して、効率的なコード記述が行えるようになります。

より詳細なAPIや、より高度なモジュール使用法を記述した他のアプリケーションプロジェクト例もルネサスWEBサイト(本書末尾の「参考文献」の項を参照)から入手でき、より複雑な設計に役立ちます。

USBX™ 通信フレームワークは、通信フレームワークアプリケーション向けのハイレベルAPI(high-level API)で、USBX デバイスクラスCDC-ACM(Communications Device Class-Abstract Control Model)を使用して実装されています。この通信フレームワークは、Synergy MCU デバイス上にあるUSB周辺回路(peripheral)を使用します。

目次

1. Communications Framework on USBX Module Features	3
2. Communications Framework on USBX Module APIs Overview	3
3. Communications Framework on USBX Module Operational Overview	3
4. Including the Communications Framework on USBX Module in an Application	3
5. Configuring the Communications Framework on USBX Module	3
5.1 Configuration Settings for the Communications Framework on USBX Lower-Level Drivers.....	3
6. Using the Communications Framework on USBX Module in an Application	3
7. USBX 通信フレームワークモジュールのアプリケーションプロジェクト(The Communications Framework on USBX Module Application Project)	3
8. ターゲットアプリケーションに対応するUSBX 通信フレームワークモジュールのカスタマイズ(Customizing the Communications Framework on USBX Module for a Target Application)	7
9. USBX 通信フレームワークモジュールのアプリケーションプロジェクトの実行(Running the Communications Framework on USBX Module Application Project)	8

10. USBX 通信フレームワークモジュールのまとめ (Communications Framework on USBX Module Conclusion)	9
11. USBX 通信フレームワークモジュールの次の手順 (Communications Framework on USBX Module Next Steps)	9
12. USBX 通信フレームワークモジュールの参考情報 (Communications Framework on USBX Module Reference Information)	9

1. **Communications Framework on USBX Module Features**
2. **Communications Framework on USBX Module APIs Overview**
3. **Communications Framework on USBX Module Operational Overview**
4. **Including the Communications Framework on USBX Module in an Application**
5. **Configuring the Communications Framework on USBX Module**
 - 5.1 **Configuration Settings for the Communications Framework on USBX Lower-Level Drivers**
6. **Using the Communications Framework on USBX Module in an Application**
7. **USBX 通信フレームワークモジュールのアプリケーションプロジェクト
(The Communications Framework on USBX Module Application Project)**

このモジュールガイドで説明するアプリケーションプロジェクトを実際に使うことで、設計全体の手順を体験することができます。このプロジェクトは、このドキュメントの末尾にある「参考情報」章に掲載されているリンクにあります。ISDE でアプリケーションプロジェクトをインポートして開き、通信フレームワークモジュールに対応する設定項目を表示することができます。また、完成した設計で、通信フレームワーク API を示すために使用している `usb_thread_entry.c` 内のコードを確認することもできます。

本アプリケーションプロジェクトは、通信フレームワーク API の標準的な使用方法を示します。このアプリケーションプロジェクトの USBX スレッドのエントリ (thread entry) は、USBX 通信フレームワークを初期化 (initialize) し、フレームワークを通じて開始メッセージ (welcome message) を送信した後、ユーザ入力を待ち受け (listen) します。ユーザが適切なキー (1、2、3 のいずれか) を押すたびに、対応する LED がトグル (toggle、オンとオフの切り替え) し、すべての LED のステータスがフレームワーク経由のメッセージで送信されます。ボードとの通信を取り扱ってメッセージを表示するには、Tera Term のようなターミナルアプリケーションが必要です。

表 1 このアプリケーションプロジェクトが使用するソフトウェアとハードウェアのリソース

リソース	リビジョン	説明
e ² studio	5.3.1 またはそれ以降	統合ソリューション開発環境 (ISDE)
IAR EW for Synergy	7.71.2 またはそれ以降	IAR Workbench IDE for Synergy プラットフォーム
SSP	1.2.0 またはそれ以降	Synergy ソフトウェアプラットフォーム
SSC	5.3.1 またはそれ以降	Synergy Standalone Configurator
SK-S7G2	v3.0 と v3.1	スタータキット

次の図に、このアプリケーションプロジェクトの簡単なフローを示します。

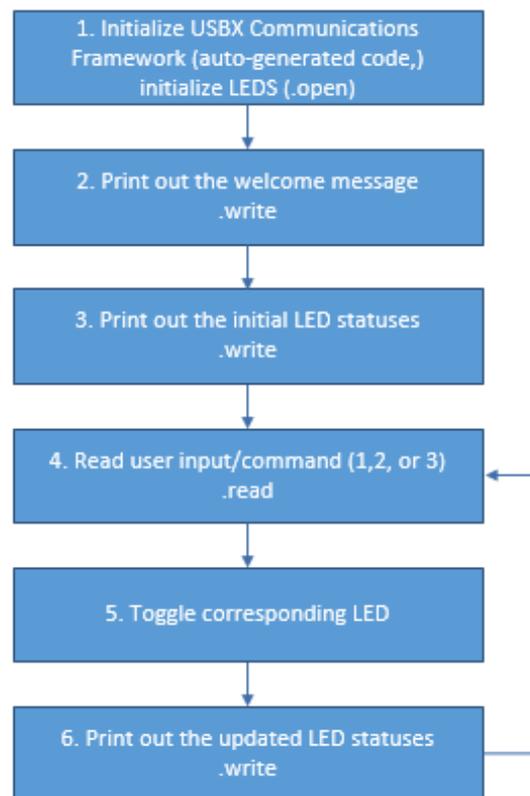


図 1 通信フレームワークのアプリケーションプロジェクトのフロー

usb_thread_entry.c ファイルは、このプロジェクトを ISDE にインストールすると、プロジェクト内に配置されます。ISDE でこのファイルを開き、API の使い方のガイドを受けることができます。

usb_thread_entry.c の最初のセクションには、①通信フレームワークインスタンス構造体 (Communication Framework instance structure) を参照する複数のヘッダファイル (header files) と、②LED 端子のレベルをトグル (点灯と消灯を切り替え) し、フレームワーク経由で送信するメッセージを生成するための関数プロトタイプ (function prototype) を記述するコードセクション (code section)、③LED 端子のレベルを表すグローバル変数 (global variables) があります。次のセクションは、メインプログラム制御セクション (main program-control section) に対応するエントリー関数 (entry function) です。最初に、LED 情報構造体 (LED information structure) を初期化し、すべての LED を消灯します。次に、ティック (tick) 1000 回分の時間にわたってスレッドをスリープ状態に維持し、ボードが USB デバイスとして確実にセットアップされるようにします。スリープが終了した後、このアプリケーションは write API を使用し、通信フレームワークを通じて開始メッセージ (welcome message) と LED ステータスメッセージを送信します。その後、このプログラムは無限ループに入り、そのループ内で通信フレームワークの read API を使用してキー (key) を読み取ります。次に入力を解析 (parse) し、受信したキーが 1、2、3 のいずれかである場合、それぞれ LED 1、2、3 をトグルするコマンドに変換します。LED のトグルを実行した後、このアプリケーションは write API を使用して LED ステータスメッセージを送信します。

続くセクションに記述されている各種関数は、LED レベルのトグル、LED 端子レベルの更新、LED ステータスメッセージの生成と送信を行います。

ターゲットボードと MCU に必須な操作と物理プロパティ (physical properties) をサポートするために、このアプリケーションプロジェクトではいくつかの重要なプロパティを設定しています。以下の表に、それらのプロパティと、このプロジェクトで設定した値を示します。実際に、このアプリケーションプロジェクトを開き、[Properties] (プロパティ) ウィンドウでこれらの設定を表示することができます。

表 2 アプリケーションプロジェクトに対応する通信フレームワークの設定項目

ISDE のプロパティ	設定値
Parameter Checking (パラメータチェック)	Default (BSP) (デフォルト (BSP))
Read Input Buffer Size (Bytes) (入力バッファの読み取りサイズ (バイト))	128
Timeout in ticks (ティック単位のタイムアウト)	1000
Name (名前)	g_sf_comms
Name of the sf comms initialization function (sf comms 初期化関数の名前)	sf_comms_init
Auto sf comms Initialization (sf comms の自動初期化)	Enable (有効)

表 3 アプリケーションプロジェクトに対応する USBX デバイスクラス CDC-ACM の設定項目

ISDE のプロパティ	設定値
Name (名前)	g_ux_device_class_cdc_acm0
USBX CDC-ACM instance_activate Function Callback (USBX CDC-ACM インスタンスアクティブ化関数のコールバック)	ux_cdc_device0_instance_activate
USBX CDC-ACM instance_deactivate Function Callback (USBX CDC-ACM インスタンス非アクティブ化関数のコールバック)	ux_cdc_device0_instance_deactivate

表 4 アプリケーションプロジェクトに対応する USBX デバイスの設定項目

ISDE のプロパティ	設定値
Vendor ID (ベンダ ID)	0x045B
Product ID (製品 ID)	0x5300 この表の下にある注記も参照してください。
Device Release Number (デバイスのリリース番号)	0x0000
Index of Manufacturing String Descriptor (製造に関する文字列記述子のインデックス)	0x00
Index of Product String Descriptor (製品に関する文字列記述子のインデックス)	0x00
Index of Serial Number String Descriptor (シリアル番号に関する文字列記述子のインデックス)	0x00
Class Code (クラスコード)	Miscellaneous (その他)。 この表の下にある注記も参照してください。
Index of String Descriptor describing this configuration (この構成を表す文字列記述子のインデックス)	0x00
Size of USB Descriptor in bytes for this configuration (Modify this value only for Vendor-specific Class, otherwise set zero) (この構成を表す USB 記述子のバイト単位のサイズ (ベンダ固有クラスを使用する場合のみこの値を変更し、それ以外の場合は 0 に設定してください))	0x00
Number of Interfaces (Modify this value only for Vendor-specific Class, otherwise set zero) (インタフェースの数 (ベンダ固有クラスを使用する場合のみこの値を変更し、それ以外の場合は 0 に設定してください))	0x00
Self-Powered (USB セルフパワー)	Enable (有効)
Remote Wakeup (リモートウェイクアップ)	Disable (無効)
Maximum Power Consumption (in 2 mA units) (最大消費電力 (2 mA を 1 単位として取り扱い))	50
Supported Language Code (サポートされている言語コード)	0x0409

ISDE のプロパティ	設定値
Name of USBX String Framework (USBX 文字列フレームワークの名前)	NULL
Total index number of USB String Descriptors in USB String Framework (USB 文字列フレームワーク内にある USB 文字列記述子の合計インデックス番号)	0
Name of USBX Language Framework (USBX 言語フレームワークの名前)	NULL
Number of Languages to support (US English is applied if zero is set) (サポートされている言語の数 (0 に設定した場合は、米国の英語を適用))	0

注記: Microsoft® Windows® 7 と Microsoft® Windows® 8 のコンピュータの場合、署名済みドライバ (signed driver) が必須です。このため、Installing Synergy Signed USB CDC Driver (R11AN0303EU0100) 内の USB ドライバを使用します。このドライバは、Renesas の Web サイトからダウンロードできます。
class code (クラスコード) を「miscellaneous」(その他) に設定してください。
なお、Microsoft® Windows® 10 のコンピュータの場合、署名済みドライバは必須ではありません。Windows は class (クラス) と sub-class (サブクラス) を使用して、どのドライバを使用するかを決定するからです。Installing Synergy Signed USB CDC Driver (R11AN0303EU0100) 内の USB ドライバを使用する場合、Product ID (製品 ID) を 0x5300 に設定してください。

表 5 アプリケーションプロジェクトに対応する USBX インタフェースの設定項目

ISDE のプロパティ	設定値
Name (名前)	g_usb_interface_desc_cdcacm_0
Interface Number of Communications Class interface (通信クラスインタフェースのインタフェース番号)	0x00
Interrupt Transfer endpoint to use for Communications Class (通信クラスで使用する割り込み転送エンドポイント)	Endpoint 3 (エンドポイント 3)
Polling period for Interrupt Endpoint (in ms/125 µs units for FS/HS) (割り込みエンドポイントのポーリング期間 (フルスピード/ハイスピードの場合、それぞれ ms/125 µs を 1 単位として取り扱い))	0x0F
Interface Number of Data Class interface (データクラスインタフェースのインタフェース番号)	0x01
Bulk In Transfer endpoint to use for Data Class (データクラスで使用するバルクイン転送エンドポイント)	Endpoint 1 (エンドポイント 1)
Bulk Out Transfer endpoint to use for Data Class (データクラスで使用するバルクアウト転送エンドポイント)	Endpoint 2 (エンドポイント 2)
Index of String Descriptor Describing Communications Class interface (通信クラスインタフェースについて記述する文字列記述子のインデックス) (Interface Descriptor: Interface) (インタフェース記述子: インタフェース)	0x00
Index of String Descriptor Describing Data Class interface (データクラスインタフェースについて記述する文字列記述子のインデックス) (Interface Descriptor: Interface) (インタフェース記述子: インタフェース)	0x00

表 6 アプリケーションプロジェクトに対応して、USBFS で使用する sf_el_ux 上の USBX ポート DCD の設定項目

ISDE のプロパティ	設定値
Full Speed Interrupt Priority (フルスピードの割り込みの優先順位)	Priority 2 (優先順位 2)
Name (名前)	g_sf_el_ux_dcd_fs_0
USB Controller Selection (USB コントローラの選択)	USBFS

表 7 アプリケーションプロジェクトに対応する ux 上の USBX の設定項目

ISDE のプロパティ	設定値
USBX Pool Memory Name (USBX プールメモリの名前)	g_ux_pool_memory
USBX Pool Memory Size (USBX プールメモリのサイズ)	18,432
User Callback for Host Event Notification (Only valid for USB Host) (ホストイベント通知用のユーザコールバック (USB ホストの場合のみ有効))	NULL

8. ターゲットアプリケーションに対応する USBX 通信フレームワークモジュールのカスタマイズ (Customizing the Communications Framework on USBX Module for a Target Application)

USBHS (ハイスピード) も使用することができます。また、開発するプロジェクトのニーズに応じて、Class Code (クラスコード)、Vendor ID (ベンダ ID)、Product ID (製品 ID) を変更することも可能です。ユーザが複数のスレッドを使用して read API と write API を呼び出す必要がある場合、ロックメカニズム (locking mechanism) を使用することが可能です。

9. USBX 通信フレームワークモジュールのアプリケーションプロジェクトの実行(Running the Communications Framework on USBX Module Application Project)

USBX 通信フレームワークのアプリケーションプロジェクトを実行し、ターゲットキットでその動作を確認するために、本プロジェクトの ISDE へのインポート、コンパイル、およびデバッグを容易に実行することができます。

注記: Synergy 開発プロセスの基本的な流れを経験したことのあるユーザーにとって、以下の手順は十分詳細なものです。これらの手順をまだ理解していない場合、このドキュメントの末尾にある「参考情報」の章に掲載されている『SSP ユーザーズマニュアル』の最初の数章を参照してください。

1. e² studio または IAR EW for Synergy™ にプロジェクトをインポートし、アプリケーションをビルドして実行する手順については、『Renesas Synergy プロジェクトインポートガイド』(下記WEB)を参照してください。

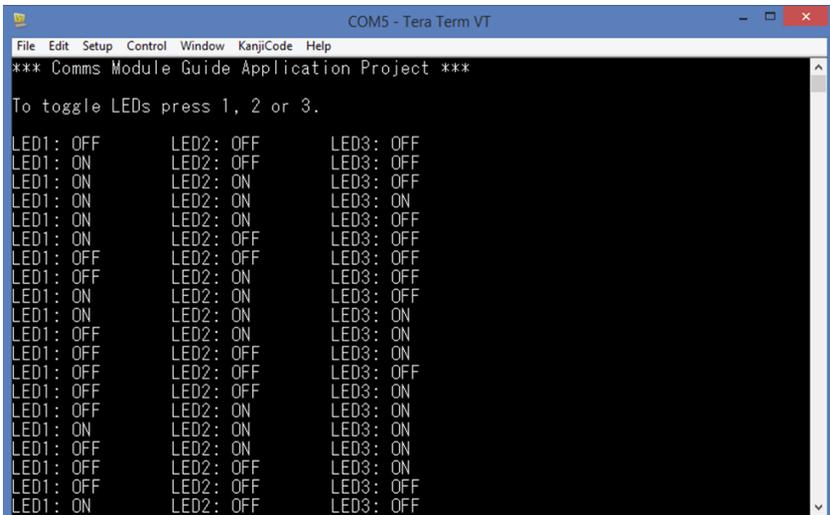
英語版:

<https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0023eu0121-synergy-ssp-import-guide.pdf>

日本語版(参考資料):

<https://www.renesas.com/jp/ja/doc/products/renesas-synergy/apn/r11an0023ju0121-synergy-ssp-import-guide.pdf>

2. micro USB ケーブルを SK-S7G2 ボードの J19 につなぎ、ホスト PC に接続します。
3. micro USB ケーブルを SK-S7G2 ボードの J5 につなぎ、ホスト PC に接続します。
4. アプリケーションのデバッグを開始します。
5. Tera Term アプリケーションを起動し、適切なシリアルポート(たとえば、COM5)で接続を確立します。さらにデバッグセッションを実施するために、Tera Term で接続状態を維持します。最初のデバッグセッションを実行する際は、開始メッセージが表示されない可能性もあります。
6. PC のキーボードで、1、2、3 のいずれかを押します。次の図に示すように、Tera Term アプリケーションで出力を確認することができます。



```
COM5 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
*** Comms Module Guide Application Project ***
To toggle LEDs press 1, 2 or 3.
LED1: OFF      LED2: OFF      LED3: OFF
LED1: ON       LED2: OFF      LED3: OFF
LED1: ON       LED2: ON       LED3: OFF
LED1: ON       LED2: ON       LED3: ON
LED1: ON       LED2: ON       LED3: OFF
LED1: ON       LED2: OFF      LED3: OFF
LED1: OFF      LED2: OFF      LED3: OFF
LED1: OFF      LED2: ON       LED3: OFF
LED1: ON       LED2: ON       LED3: OFF
LED1: ON       LED2: ON       LED3: ON
LED1: OFF      LED2: ON       LED3: ON
LED1: OFF      LED2: OFF      LED3: ON
LED1: OFF      LED2: OFF      LED3: OFF
LED1: OFF      LED2: ON       LED3: ON
LED1: OFF      LED2: ON       LED3: ON
LED1: ON       LED2: ON       LED3: ON
LED1: OFF      LED2: ON       LED3: ON
LED1: OFF      LED2: OFF      LED3: ON
LED1: OFF      LED2: OFF      LED3: OFF
LED1: ON       LED2: OFF      LED3: ON
```

図 2 USBX 通信フレームワークのアプリケーションプロジェクトのサンプル出力

10. USBX 通信フレームワークモジュールのまとめ (Communications Framework on USB Module Conclusion)

このモジュールガイドでは、サンプルプロジェクトでモジュールの選択、追加、設定、使用を行うために必要な背景となる情報全般を説明しました。従来の組み込みシステムでは、これらの手順を理解することに多くに時間を必要とし、また間違いが起りやすい操作でした。Renesas Synergy プラットフォームにより、これら手順の所要時間が短くなり、設定項目の競合や、ローレベルドライバの誤った選択など、誤りが防止できるようになりました。アプリケーションプロジェクトで示したように、ハイレベル API を使用することで高いレベルの開発からスタートし、ローレベルドライバを作成するような従来の開発環境が必要とされる時間が不要になり、開発時間を短縮できます。

11. USBX 通信フレームワークモジュールの次の手順 (Communications Framework on USB Module Next Steps)

シンプルな USBX 通信フレームワークのプロジェクトをマスターした後、NX または UART を使用する他の通信フレームワーク実装を確認することができます。

12. USBX 通信フレームワークモジュールの参考情報 (Communications Framework on USB Module Reference Information)

『SSP ユーザーズマニュアル』: SSP ディストリビューションパッケージの一部として html 形式が入手できるほか、Synergy WEB SSP サイトから pdf を入手できます。

<https://www.renesas.com/jp/ja/products/synergy/software/ssp.html>

sf_el_ux_comms モジュールの参考資料やリソースに関する最新版のリンクは、以下の Synergy ナレッジベース WEB ページで利用できます。

<https://www.renesas.com/jp/ja/products/synergy.html>

Web サイトおよびサポート

サポート: <https://synergygallery.renesas.com/support>

テクニカルサポート:

- アメリカ: https://renesas.zendesk.com/anonymous_requests/new
- ヨーロッパ: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.05.21		<ul style="list-style-type: none">・初版・英語版(R11AN0136EU0100, Rev.1.00, 2017.Aug.31)の巻頭と第 7 章以降を翻訳・RFP のドライバの記述を削除し、USB ドライバの記述に入れ替え

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まれます。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したします。
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（豊洲フォレストシア）

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。