

(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、GPT HAL モジュール(GPT HAL Module)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。

汎用 PWM タイマ (GPT) HAL モジュール(General PWM Timer (GPT) HAL module)は、タイマアプリケーション(timer applications)向けのハイレベル HAL (high-level API) で、r_gpt 内に実装されています。GPT HAL モジュールは、Synergy MCU 上にある GPT 周辺回路(peripheral)を使用します。タイマイベント(timer event)に応答(respond)するために、ユーザ定義のコールバックを作成できるようになります。

目次

1. GPT HAL モジュールの機能(GPT HAL Module Features)	3
2. GPT HAL モジュールの API の概要(GPT HAL Module APIs Overview)	3
3. GPT HAL モジュールの動作の概要(GPT HAL Module Operational Overview)	3
4. アプリケーションでの GPT HAL モジュールの使用(Including the GPT HAL Module in an Application)	3
5. GPT HAL モジュールの設定(Configuring the GPT HAL Module)	3
6. アプリケーションでの GPT HAL モジュールの使用(Using the GPT Module in an Application)	3
7. GPT HAL モジュールのアプリケーションプロジェクト(The GPT HAL Module Application Project)	3
8. ターゲットアプリケーションに対応する GPT HAL モジュールのカスタマイズ(Customizing the GPT HAL Module for a Target Application)	6
9. GPT HAL 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 GPT HAL Module Application Project)	7
10. GPT HAL モジュールのまとめ(GPT HAL Module Conclusion)	8
11. GPT HAL モジュールの次の手順(GPT HAL Module Next Steps)	8

12. GPT HAL モジュールの参考情報 (GPT HAL Module Reference Information)8

1. GPT HAL モジュールの機能 (GPT HAL Module Features)
2. GPT HAL モジュールの API の概要 (GPT HAL Module APIs Overview)
3. GPT HAL モジュールの動作の概要 (GPT HAL Module Operational Overview)
4. アプリケーションでの GPT HAL モジュールの使用 (Including the GPT HAL Module in an Application)
5. GPT HAL モジュールの設定 (Configuring the GPT HAL Module)
6. アプリケーションでの GPT HAL モジュールの使用 (Using the GPT Module in an Application)
7. GPT HAL モジュールのアプリケーションプロジェクト (The GPT HAL Module Application Project)

このモジュールガイドに関連するアプリケーションプロジェクトは、設計全体の手順を示します。ISDE でアプリケーションプロジェクトをインポート (import) して開き、GPT HAL 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 (complete design) における GPT HAL モジュール API を理解するために、<GPT_HAL_MG.c> 内のコードを確認することもできます。

このアプリケーションプロジェクトは、3 つの GPT モジュールを使用する GPT HAL API の一般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メインスレッドのエントリは、タイマの初期化と起動、および期間パラメータとデューティサイクル (duty cycle) パラメータの再設定を行います。GPT 0 と GPT 2 が期限切れ (expire) になった時点で、個別のコールバック呼び出しを通じて、LED をトグル (点灯と消灯を切り替え) します。GPT 1 は、PWM タイマとして設定済みです。タイマに関する情報は、共通のセミホスト機能を使用して、デバッグコンソールに表示されます。以下の表は、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バージョンを示します。

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)
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
IAR EW for Synergy	7.71.2 またはそれ以降	IAR Embedded Workbench® for Renesas Synergy™
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

以下の図は、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。

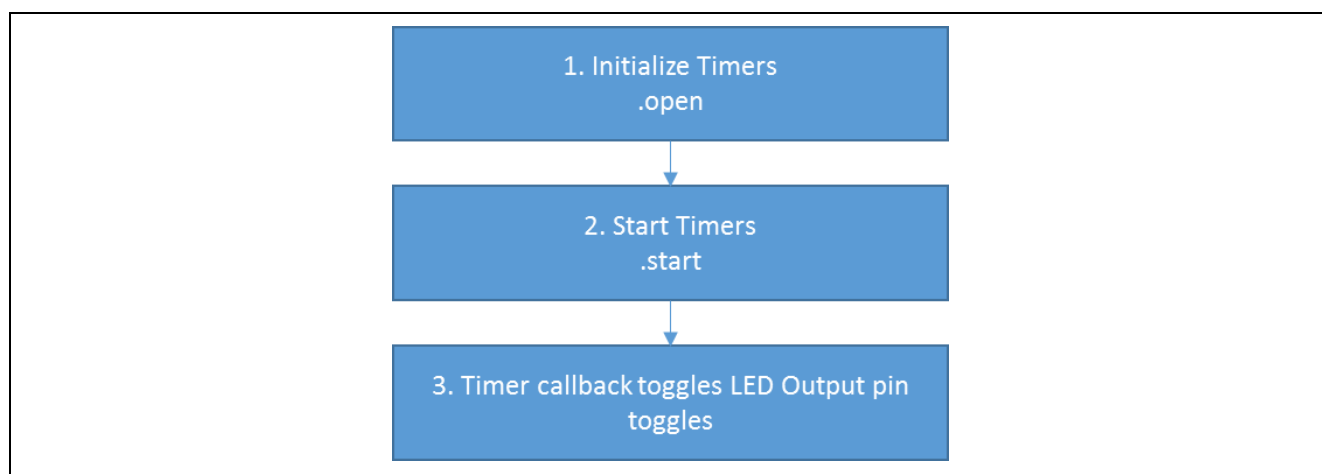


図 1 GPT HAL モジュールのアプリケーションプロジェクトのフロー

このプロジェクトでは、以下の 3 個のタイマを使用します。

1. g_timer_gpt_0 は、周期的 GPT としてセットアップします。
2. g_timer_gpt_1 は、PWM GPT としてセットアップします。
3. g_timer_gpt_2 は、ワンショット(one-shot) GPT としてセットアップします。

以下の表に、各タイマが使用する設定プロパティを示します。他の設定項目はデフォルト値のままにします。

表 2 g_timer_gpt_0 に対応する設定プロパティ

ISDE のプロパティ	設定済みの値
Name (名前)	g_timer_gpt_0
Channel (チャンネル)	0
Mode (モード)	Periodic
Period Value (期間の値)	1
Period Unit (期間の単位)	Seconds
Auto Start (オートスタート)	False
Callback (コールバック)	timer0_callback
Interrupt Priority (割り込みの優先順位)	Priority 2

表 3 g_timer_gpt_1 に対応する設定プロパティ

ISDE のプロパティ	設定済みの値
Name (名前)	g_timer_gpt_1
Channel (チャンネル)	1
Mode (モード)	PWM
Period Value (期間の値)	1
Period Unit (期間の単位)	Seconds
Auto Start (オートスタート)	False
GTIOCB Output Enabled (GTIOCB 出力の有効化)	True
Interrupt Priority (割り込みの優先順位)	Priority 5

表 4 g_timer_gpt_2 に対応する設定プロパティ

ISDE のプロパティ	設定済みの値
Name (名前)	g_timer_gpt_2
Channel (チャンネル)	2
Mode (モード)	One Shot
Period Value (期間の値)	10
Period Unit (期間の単位)	Seconds
Auto Start (オートスタート)	False
Callback (コールバック)	timer_2_callback
Interrupt Priority (割り込みの優先順位)	Priority 5

[Pin] (端子) タブウィンドウで、以下の端子を設定します。これらの端子をオシロスコープに接続し、タイマ出力を表示することもできます g_timer_gpt_1:

Peripherals > Timer:GPT > GPT1.

• Pin Group Selection > Mixed

• Operation Mode: GTIOCA もしくは GTIOCB

• GTIOCB > P406 次に P406 を展開し、P406 の設定ページに移動します。GPT1_GTIOCB を Chip input/output 設定する必要があります。

GPT モジュールの使用において、ピングループ A の P101 は GPT の出力の代わりに使用できます。GTIOCA も GTIOCB の代わりに選択することができます。ユーザは、出力用に使いたいピンに従って、ピングループとオペレーションモードのオプションから選択できます。

GPT_HAL_MG.c ファイルは、このプロジェクトを ISDE にインポートすると、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 の主な使い方を確認するための説明を参照できます。

GPT_HAL_MG.c の最初のセクションは、タイマの設定に使用する複数の関数のプロトタイプを記述しています。Hal_entry.c は init_timers() 関数を呼び出し、この関数はタイマを開き、起動して設定します。open_timers() 関数を使用してこれらのタイマを開きます。この関数は、各タイマに対して open API を呼び出します。次に、start_timers()関数は、各タイマに対して start API を呼び出します。

コード内の次のセクションは、期間(period)と期間のデューティサイクル(duty cycle)を設定します。実行時(at run time)にこれらのパラメータを変更する必要がない場合、このセクションは不要です。期間を変更するには、set_timer_period 関数を使用します。この関数は、periodSet API を呼び出します。デューティサイクルを変更するには、set_timer_pwm を使用します。この関数は、dutyCycleSet API を呼び出します。コードスニペット(code snippet)は、以下の作業を実行します。

1. g_timer_gpt_0 のタイマ期間を 2 秒に変更します。オンボード LED1 は 2 秒ごとにトグル (点灯と消灯を切り替え) します。
2. g_timer_gpt_1 のデューティサイクルを 30% に変更します。P406 にプローブを接続して、PWM 出力をオシロスコープで表示することができます。
3. すべてのタイマのカウンタ値を取得し、デバッグコンソールに出力します。get_timer_value 関数は、counterGet API を呼び出す方法で、この操作を実行します。
4. すべてのタイマの情報を取得し、デバッグコンソールに出力します。get_timer_info 関数は、infoGet API を呼び出す方法で、この操作を実行します。

最後のセクションは、ユーザコールバック関数(user callback function)です。timer0_callback は、g_timer_gpt_0 期限切れになった場合のために定義されているコールバック関数です。g_timer_gpt_0 が期限切れになったときに必ずこのコールバックが呼び出され、LED0 のトグルを実行します。timer2_callback は、g_timer_gpt_2 期限切れになった場合のために定義されているコールバック関数です。g_timer_gpt_2 が期限切れになったときにこのコールバックが呼び出され、LED1 のトグルを実行します。

アプリケーションプロジェクトが想定する出力

1. LED0 は 2 秒ごとにトグル (点灯と消灯を切り替え) します。この動作は、g_timer_gpt_0 の期間に基づいています。
2. LED1 は 1 回トグルします。この LED を設定しているのは、ワンショットタイマである g_timer_gpt_2 に対応するコールバックだからです。g_timer_gpt_2 をもう一度開始して、LED1 を複数回トグルさせることもできます。
3. P406 の 30% PWM 出力。

注記: この説明は、Synergy ソフトウェアパッケージ内のデバッグコンソールで printf() を使用方法をユーザが理解していることを想定しています。このような経験がない場合は、下記 WEB サイトの FAQ 2000008 「Synergy ソフトウェアパッケージのデバッグコンソールで Printf_使用方法」という記事を参照してください。デバッグモードで変数ウォッチ機能を使用して結果を表示することもできます。

<https://ja-support.renesas.com/knowledgeBase/17792531>

8. ターゲットアプリケーションに対応する GPT HAL モジュールのカスタマイズ (Customizing the GPT HAL Module for a Target Application)

いくつかの設定項目は通常、アプリケーションプロジェクトで示している値に対し、ユーザが変更を加えます。たとえば、ユーザは [Clock] (クロック) タブで PCLKD を更新する方法により、GPT クロックに関する設定項目を簡単に変更することができます。また、ユーザは GPT の出力ポート端子を変更することもできます。この変更を行うには、コンフィギュレータの [Pins] (端子) タブを使用します。

このアプリケーションプロジェクトは、タイマをリセットし、停止して閉じる関数も公開しています。ユーザのアプリケーションに従って、これらの関数を使用することができます。

9. GPT HAL 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 GPT HAL Module Application Project)

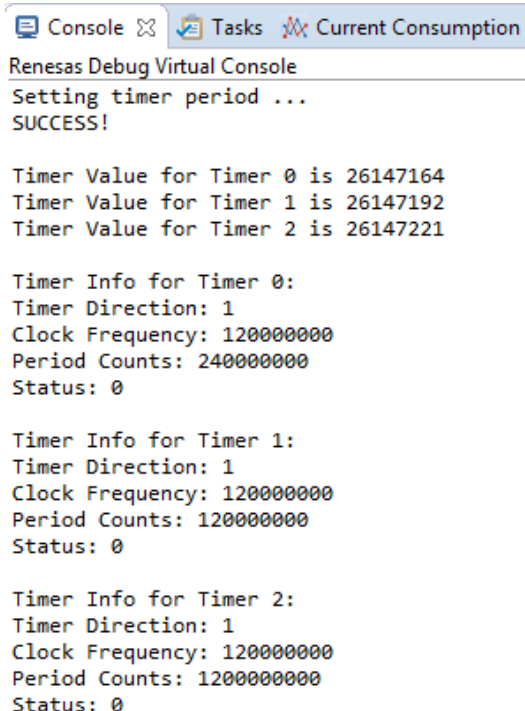
GPT HAL モジュールのアプリケーションプロジェクトの動作を確認するために、ターゲットキットで ISDE にこのプロジェクトをインポートし、コンパイルしてデバッグを実行することができます。

新しいプロジェクト内で GPT HAL モジュールアプリケーションを実装するには、ターゲットキットで定義、設定、ファイルの自動生成、コードの追加、コンパイル、デバッグを行う以下の手順に従います。このガイドに示す手順に従うことで SSP での開発プロセスをより実践的に習得するのに役立ちます。

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』の最初にあるいくつかの章を参照してください。

GPT HAL モジュールのアプリケーションプロジェクトを作成し、実行するために、以下の手順に従ってください。

1. <GPT_HAL_MG_AP> という名称で、<SK-S7G2> に対応する新しい Renesas Synergy プロジェクトを作成します。
2. タイマドライバを <HAL/Common> に追加します。
3. [Generate Project Content] (プロジェクトコンテンツの生成) ボタンをクリックします。
4. 付属のプロジェクトファイル <GPT_HAL_MG>.c、<GPT_HAL_MG>.h、<hal_entry>.c からコードを追加するか、生成された同名のファイルに上書きする形でこれらの付属ファイルをコピーします。
5. micro USB ケーブルで SK-S7G2 の J19 コネクタとホスト PC を接続します。
6. アプリケーションのデバッグを開始します。
7. 出力は、Renesas Debug Virtual Console (Renesas デバッグ仮想コンソール) に表示されます。



```
Console Tasks Current Consumption
Renesas Debug Virtual Console
Setting timer period ...
SUCCESS!

Timer Value for Timer 0 is 26147164
Timer Value for Timer 1 is 26147192
Timer Value for Timer 2 is 26147221

Timer Info for Timer 0:
Timer Direction: 1
Clock Frequency: 120000000
Period Counts: 240000000
Status: 0

Timer Info for Timer 1:
Timer Direction: 1
Clock Frequency: 120000000
Period Counts: 120000000
Status: 0

Timer Info for Timer 2:
Timer Direction: 1
Clock Frequency: 120000000
Period Counts: 120000000
Status: 0
```

図 2 タイマドライバアプリケーションプロジェクトのサンプル出力

また、PWM GPT 出力をオシロスコープで表示することができます。PWM GPT は、P406 に出力されます。

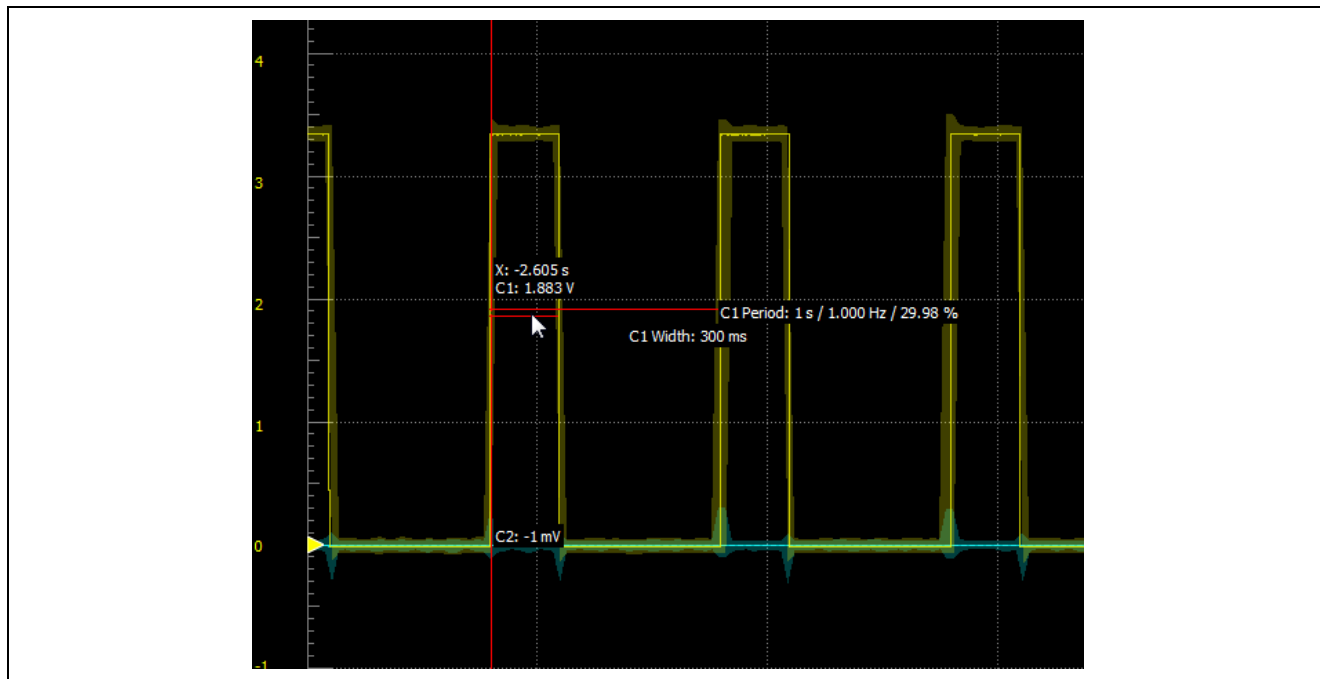


図3 g_timer_gpt_1 を、期間 1 秒、デューティサイクル 30% の PWM モードに設定した場合
オシロスコープ波形

10. GPT HAL モジュールのまとめ (GPT HAL Module Conclusion)

このモジュールガイドは、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、また間違いが起こ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で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. GPT HAL モジュールの次の手順 (GPT HAL Module Next Steps)

シンプルな GPT HAL モジュールのプロジェクトをマスターした後、より複雑なサンプルを確認することができます。GPT HAL モジュールを、ルネサスが提供する他のモジュールと結合することも検討できます。

ユーザのタイミングに関する要求によっては、クロックソース PCLKD の周波数を変更する必要があります。

12. GPT HAL モジュールの参考情報 (GPT HAL 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、Renesas Synergy™ WEBサイトのSSPページ

<https://www.renesas.com/jp/ja/products/synergy/software/ssp.html>から pdf を入手することもできます。

最新版の r_gpt モジュールの参考資料やリソースへのリンクは、以下の Synergy WEBサイトから入手できます。

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

ホームページとサポート窓口

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>

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
1.01	2019.06.03		・初版 ・英文版(R11AN0091EU0101、Rev.1.01、2017.Aug.23)の 巻頭と第 7 章以降を翻訳

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、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、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いたします。
 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)



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/>