

---

# RX600 シリーズ

R01AN0509JU0100

Rev.1.00

2011.09.27

---

## Printf および Scanf サポートの追加

---

### 要旨

本アプリケーションノートでは、HEW プロジェクトに printf および scanf サポートを追加する方法について説明しています。標準入力および出力がシミュレートされたコンソール、デバッグコンソールおよび標準シリアルポートにリダイレクトされる例を示しています。

### 動作確認デバイス

ルネサス RX600 シリーズ

### 目次

1. 概要 .....	2
2. シミュレータコンソールへの標準入出力のリダイレクト .....	3
3. HEW デバッグコンソールへの標準入出力のリダイレクト .....	5
4. プラットフォームシリアルポートへの標準入出力のリダイレクト .....	6
5. ユーザ独自のプロジェクトへのサポートの追加 .....	8

## 1. 概要

コンソールへの書き込みとコンソールからの読み出しを可能にすることは、プログラム作成および開発プロセスにおいて不可欠な要素です。ランタイム情報を表示するためには、`printf` が必要で、構成パラメータを収集するためには、`scanf` が必要です。さらに、ハードウェアアーキテクチャまたは開発段階によって、ターミナルエミュレーションを実行するためにシリアルポートに物理的にアクセスできない場合があるかもしれません。

本アプリケーションノートおよび付属の `RX_printf` プロジェクトの目的は、さまざまなコンソールに対する開発環境で `printf()` および `scanf()` 関数を取得する方法の例を示すことです。シミュレーション環境における HEW シミュレータコンソール、HEW デバッグコンソールまたはハードウェアデバッグ環境におけるハードウェアのシリアルポートにシリアル入出力をどのように転送できるかを示します。

対象コンソールに関係なく共通しているのは、低水準関数 `charput()` および `charget()` です。`RX_printf` プロジェクトは、これらの3つのシナリオを示すために個別のビルド構成を提供します。各構成から異なる低水準ソースファイルが作成され、適切なコンソールに入出力をリダイレクトするためにこれらの関数を提供します。

シリアル入出力リダイレクト	ソースコード	ビルド構成
HEW シミュレータコンソール	<code>lowlvl_Sim.src</code>	<code>Sim_Output</code>
HEW デバッグコンソール	<code>lowlvl_DebugConsole.src</code>	<code>DebugConsole_Output</code>
RSK+RX62N RS-232 ポート	<code>serial_printf.c</code>	<code>Serial_Output</code>

**【注】** このアプリケーションノートと関連するプロジェクト `RX_printf` は、RSK+RX62N 基準プラットフォーム上で実行されます。

## 2. シミュレータコンソールへの標準入出力のリダイレクト

最初に、HEW シミュレータコンソールへのシリアル入出力のリダイレクトについて説明します。第 1 章で説明したように、入出力ストリームリダイレクトは低水準ソフトウェアで制御されます。この例では、lowlvl\_sim.src を使用します。

### 2.1 lowlvl\_sim.src

このソースコード内の関数は、HEW において新規プロジェクトを作成するときに生成されます。プロジェクトディレクトリに lowlvl.src と表示されます。このデモンストレーションでは、使用目的が簡単に識別できるように名前が変更されています。

このコードを変更する必要はありません。charput() および charget() 関数は HEW と対話を行い、プログラムとワークステーション上で動作する HEW の間で文字を渡します。

### 2.2 サンプルプロジェクトのビルド

RX\_printf プロジェクトをワークスペースディレクトリにコピーします。HEW においてワークステーションを開き、「Build Configuration」プルダウンから "Sim\_Output" を選択し、「Debug Session」プルダウンから "SimSessionRX600" を選択します。

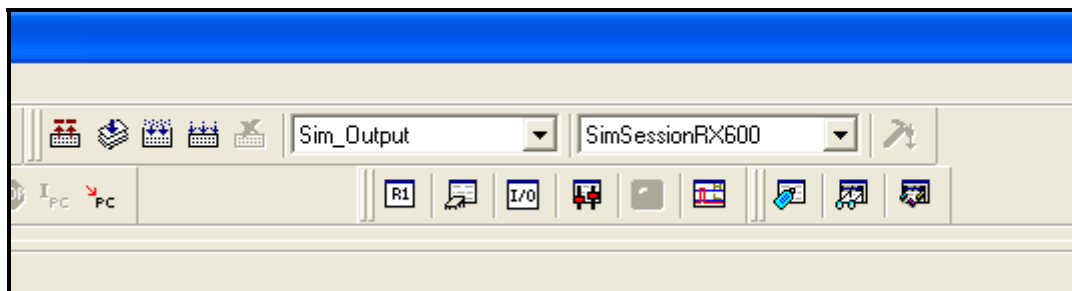


図 2.1 ビルドおよびデバッグ構成の選択

図 2.2 に示すように「Set Simulator」ポップアップでデフォルトを使用します。

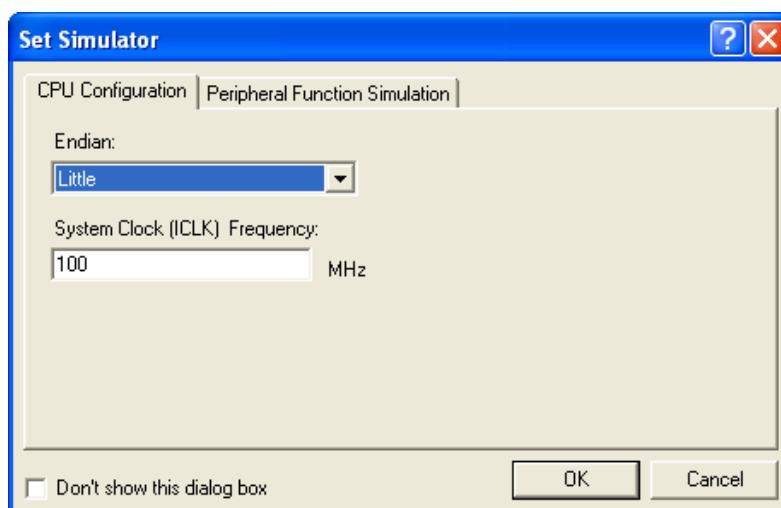


図 2.2 RX シミュレータの構成

図 2.3 に示す Project Navigation ペインでは lowlvl\_Sim.src がビルドに含まれ、その他の 2 つのファイルは除外されていることを確認してください。どの printf/scanf ファイルを含め、除外するかは、選択されたビルド構成によって制御されます。

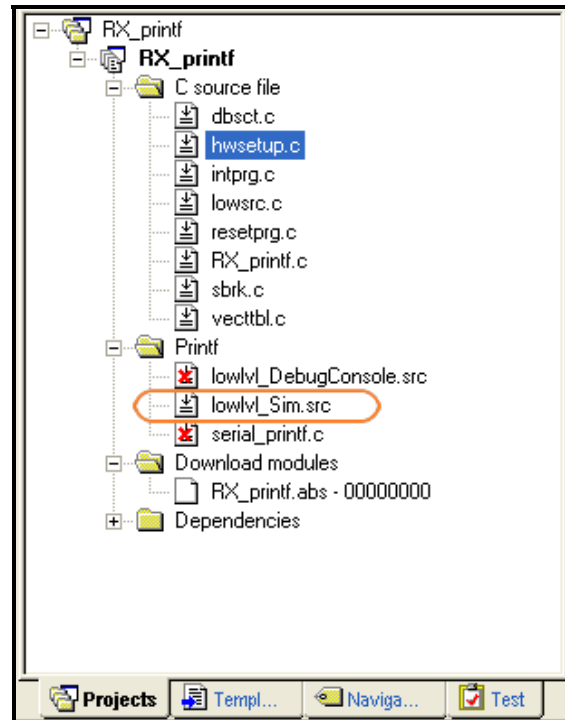



図 2.3 Project Navigation ペイン

「F7」を押すか、Build Build All の順に選択して、プロジェクトをビルドします。プロジェクトをビルドした後、'RX\_printf.abs' をダブルクリックして RX シミュレータにモジュールをダウンロードします。まだ開かれていない場合は、View CPU Simulated I/O の順に選択するか、HEW において Simulated I/O アイコン  をクリックすることにより Simulated I/O ウィンドウを開きます。

Debug Reset Go の順に選択してコードを実行します。main() の printf() および scanf() が図 2.4 に示すように HEW シミュレータコンソールに対してリダイレクトされることを確認してください。整数を入力するように求められた場合は、値を入力し、キーボードの「Enter」キーを押します。

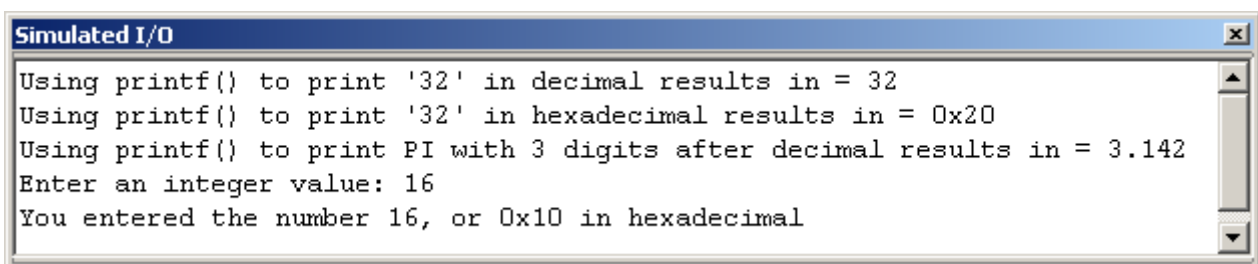


図 2.4 シミュレータコンソール出力

### 3. HEW デバッグコンソールへの標準入出力のリダイレクト

次の例は HEW デバッグコンソールを介した標準入出力の使い方を示します。この例の `charput()` および `charget()` 関数は `lowlvl_DebugConsole.src` から取り出されます。

#### 3.1 lowlvl\_DebugConsole.src

`lowlvl_DebugConsole.src` の低水準インタフェースルーチン `charput()` および `charget()` は、RX62N の「E1/E20 エミュレータ ユーザーズマニュアル別冊 (接続時の注意事項)」(R20UT0399JJ0300) に記載されたコード例から引用しています。使用している開発環境で動作するために変更する必要はありません。これらの関数がどのように動作するかの詳細については、上記のドキュメントの「デバッグコンソール機能」を参照してください。

#### 3.2 サンプルプロジェクトのビルド

「Build Configuration」プルダウンから "DebugConsole\_Output" を選択し、「Debug Session」プルダウンから "SessionRX600\_E1\_E20\_SYSTEM" を選択します。

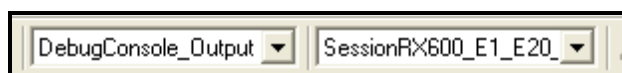


図 3.1 デバッグコンソールプロジェクト構成

`low_lvl_Debug_Console.src` がビルドに含まれていることを確認してください。

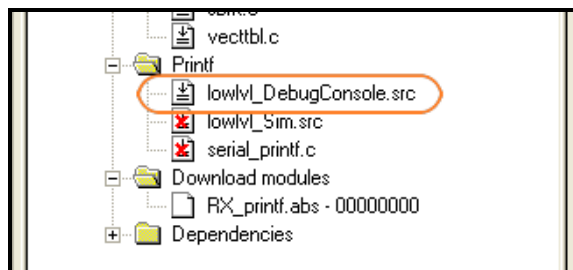


図 3.2 含まれるデバッグコンソールファイル

プロジェクトをビルドおよびロードし、前の例でシミュレータコンソールを開くために使用したボタンをクリックして、HEW デバッグコンソールを開きます。コードを実行し、`main()` の `printf()` および `scanf()` が HEW デバッグコンソールに対してリダイレクトされることを確認してください。デバッグコンソールは図 2.4 のようになります。

#### 4. プラットフォームシリアルポートへの標準入出力のリダイレクト

最後の例では、RS-232 シリアルポートを介した標準入出力の使い方を示します。このシナリオを実現するには、ハードウェアとカスタムソフトウェアに関する知識が必要です。RSK 基準ボード上の RX シリアル通信インタフェース (SCI) は、hwsetup.c の InitSCI() によって初期化されます。"get" および "put" 関数は serial\_printf.c によって提供されます。

##### 4.1 serial\_printf.c

このファイルのソースコードは RSK+RX62N ハードウェア用に特別に作成されています。単純なシリアル通信のために RX SCI 上でポーリングを行います。

セマフォ、割り込みハンドラまたは RTOS のような環境で通常使用されるその他のさまざまなメカニズムが関連する、より高度なソリューションが実装に必要になることがあります。さらに、これらの関数は、カスタムハードウェアとインタフェースを取るようカスタマイズする必要があります。

##### 4.2 サンプルプロジェクトのビルド

「Build Configuration」および「Debug Session」プルダウンから "Serial\_Output" および "SessionRX600\_E1\_E20\_SYSTEM" を選択します。

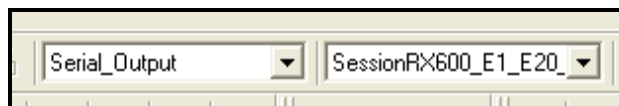


図 4.1 シリアル出力プロジェクト構成

ファイル serial\_printf.c はビルドに含まれます。このファイルと hwsetup.c の両方はプラットフォームに依存するファイルで、RSK+RX62N 基準プラットフォームで動作するように変更されています。

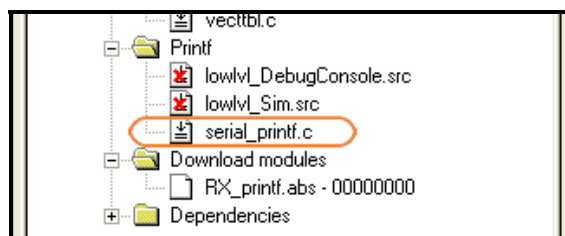


図 4.2 含まれる serial\_printf.c

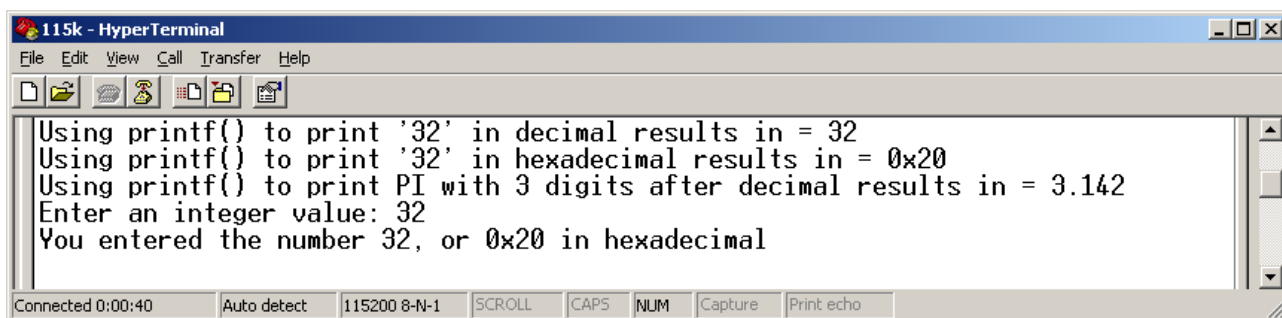
次に、RSK ボードとワークステーションの間にシリアルケーブルを接続し、ターミナルエミュレーションを開始します。以下のパラメータを使用してワークステーションを構成します。

- 115200 ビット/秒
- 8 ビットデータ
- パリティなし
- ストップビット 1
- フロー制御なし

【注】 RSK+RX62N 基準プラットフォームには、シリアルポートを利用するために設定するジャンパがあります。

- J6 と J15 は TxD2-A の位置に設定します。
- J5 と J16 は RxD2-A の位置に設定します。

プロジェクトをビルドしロードします。コードを実行して、main()の `charget()` および `charput()` がターミナルエミュレータに対してリダイレクトされることを確認してください。



The screenshot shows a HyperTerminal window titled "115k - HyperTerminal". The window contains the following text:

```
Using printf() to print '32' in decimal results in = 32
Using printf() to print '32' in hexadecimal results in = 0x20
Using printf() to print PI with 3 digits after decimal results in = 3.142
Enter an integer value: 32
You entered the number 32, or 0x20 in hexadecimal
```

The status bar at the bottom of the window shows: Connected 0:00:40, Auto detect, 115200 8-N-1, SCROLL, CAPS, NUM, Capture, Print echo.

## 5. ユーザ独自のプロジェクトへのサポートの追加

printf/scanf 関数をユーザ独自のプロジェクトに組み込む場合は、プロジェクト構成にいくつかの変更を含める必要があります。

### 5.1 新規プロジェクトの作成

新規プロジェクトを作成する場合は、セットアップウィンドウで設定するときに図 5.1 に示すように "Use I/O Library" オプションをチェックしてください。

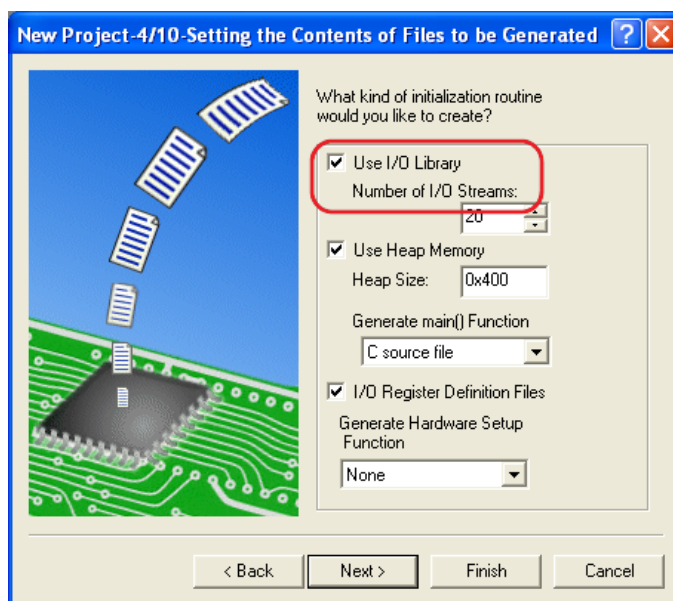


図 5.1 入出力ライブラリの選択

また、次の画面で stdio.h を選択します。

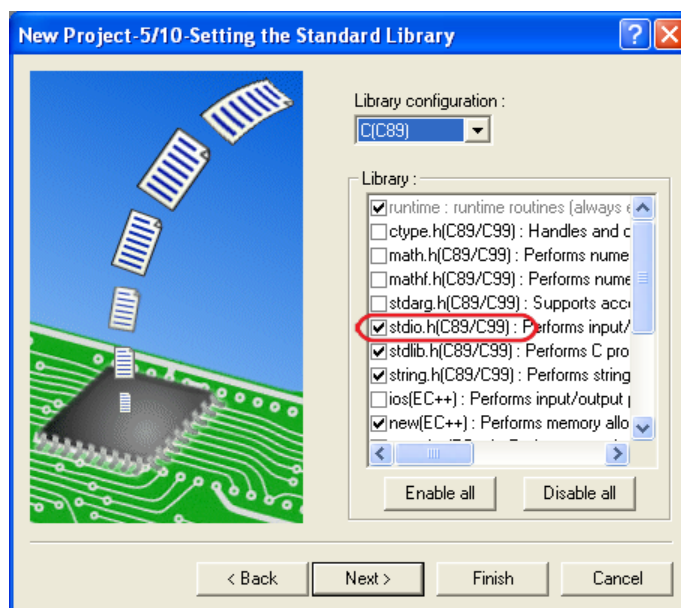


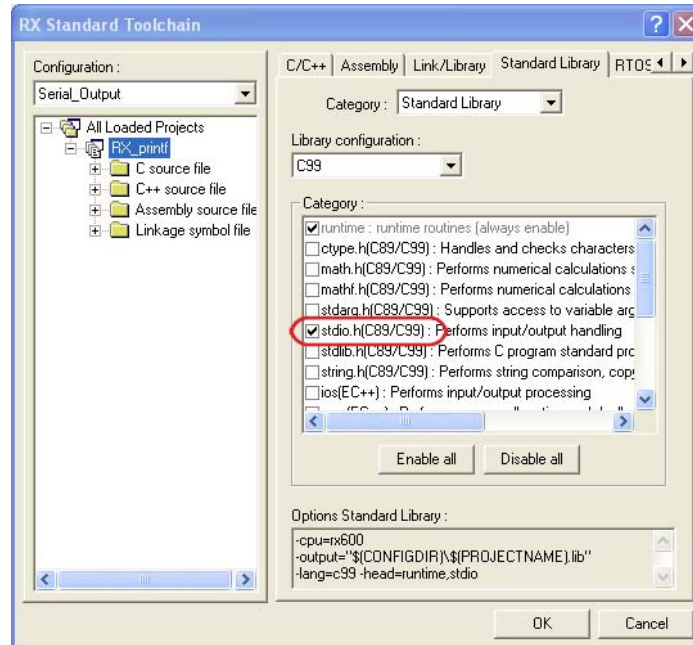
図 5.2 stdio.h の選択



## 5.2 既存のプロジェクトの更新

同様に、既存のプロジェクトを更新する場合は、stdio.h をビルドに含める必要があります。これは、HEW ツールバーから Build → RX Standard Toolchain の順に選択し、表示される構成ウィンドウの「Standard Library」タブをクリックして行います。

「Category」プルダウンの"Standard Library"を選択します。stdio.h がチェックされていることを確認します。



任意のシリアル入出力関数を実現するために適切な `charget()` および `charget()` 関数を組み込んでください。

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/>

お問い合わせ先

<http://japan.renesas.com/inquiry>

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
1.00	2011.09.27	—	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. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】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. リザーブアドレス（予約領域）のアクセス禁止

【注意】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、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）がありません。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。

### 4. クロックについて

【注意】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。

### 5. 製品間の相違について

【注意】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っていると、内部 ROM、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。型名が異なる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. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連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の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
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  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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おりますが、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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。お客様がかかる法令を遵守しないこと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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

- 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とその総株主の議決権の過半数を直接または間接に保有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。



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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■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：<http://japan.renesas.com/inquiry>