

RX ファミリ

R01AN2030JJ0122

Rev.1.22

Sep 30, 2017

USB Peripheral Communications Device Class Driver (PCDC) Firmware Integration Technology

要旨

本アプリケーションノートでは、Firmware Integration Technology (FIT) を使用した、USB Peripheral コミュニケーションデバイスクラスドライバ (PCDC) について説明します。本モジュールは USB Basic Host and Peripheral Driver(USB-BASIC-FW FIT モジュール)と組み合わせることで動作します。以降、本モジュールを USB PCDC FIT モジュールと称します。

対象デバイス

RX63N/RX631 グループ
RX65N/RX651 グループ
RX64M グループ
RX71M グループ

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。

関連ドキュメント

1. Universal Serial Bus Revision 2.0 specification
2. USB Class Definitions for Communications Devices Revision 1.2
3. USB Communications Class Subclass Specification for PSTN Devices Revision 1.2
【<http://www.usb.org/developers/docs/>】
4. RX63N/RX631 グループユーザーズマニュアル ハードウェア編 (ドキュメント No.R01UH0041JJ)
5. RX64M グループユーザーズマニュアル ハードウェア編 (ドキュメント No.R01UH0377JJ)
6. RX71M グループユーザーズマニュアル ハードウェア編 (ドキュメント No.R01UH0493JJ)
7. RX65N/RX651 グループユーザーズマニュアル ハードウェア編 (ドキュメント No. R01UH0590JJ)
8. RX65N/RX65-2M グループユーザーズマニュアル ハードウェア編 (ドキュメント No. R01UH0659JJ)
9. USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート
(ドキュメント No. R01AN2025JJ)

— ルネサス エレクトロニクスホームページ

【<http://japan.renesas.com/>】

— USB デバイスページ

【<http://japan.renesas.com/prod/usb/>】

目次

1. 概要	3
2. ソフトウェア構成.....	4
3. API 情報.....	5
4. コミュニケーションデバイスクラス (CDC) 、PSTN and ACM	8
5. API.....	12
6. コンフィグレーション (r_usb_pcdc_config.h).....	13
7. アプリケーションの作成方法	14

1. 概要

USB PCDC FIT モジュールは、USB-BASIC-FW FIT モジュールと組み合わせることで、USB Peripheral コミュニケーションデバイスクラスドライバ（以降 PCDC と記述）として動作します。PCDC は、USB コミュニケーションデバイスクラス仕様（以降 CDC と記述）の Abstract Control Model に準拠し、USB ホストとの通信を行うことができます。

以下に、本モジュールがサポートしている機能を示します。

- ・ USB ホストとのデータ転送
- ・ CDC クラスリクエストに応答
- ・ コミュニケーションデバイスクラスノーティフィケーション送信サービスの提供

1.1 必ずお読みください

このドライバを使ってアプリケーションプログラムを作成する場合は、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)を参照いただきますようお願いいたします。このアプリケーションノートは、パッケージ内の"reference_documents"フォルダにあります。

1.2 注意事項

本ドライバは、USB 通信動作を保証するものではありません。システムに適用される場合は、お客様における動作検証はもとより、多種多様なデバイスに対する接続確認を実施してください。

1.3 用語一覧

本資料で使用される用語と略語は以下のとおりです。

ACM	: Abstract Control Model.
APL	: Application program
CDC	: Communications Devices Class
CDCC	: Communications Devices Class Communications Interface Class
CDCD	: Communications Devices Class Data Class Interface
CPD	: Serial Communication Port Driver
H/W	: Renesas USB device
PCD	: Peripheral control driver of USB-BASIC-FW
PCDC	: Peripheral用 Communications Devices Class
PCDCD	: Peripheral Communications Devices Class Driver
PDCD	: Peripheral device class driver (device driver and USB class driver)
PSTN	: Public Switched Telephone Network, contains the ACM (above) standard.
RSK	: Renesas Starter Kits
USB	: Universal Serial Bus
USB-BASIC-FW	: USB Basic Host and Peripheral Driver for RXファミリ

1.4 USB PCDC FIT モジュール

本モジュールは、r_usb_basic を使用したプロジェクトに組み込む必要があります。プロジェクトに組み込み後、API を使用することで USB の H/W 制御を行います。

2. ソフトウェア構成

Figure 2-1に PCDC のモジュール構成、Table 2-1にモジュール機能概要を示します。

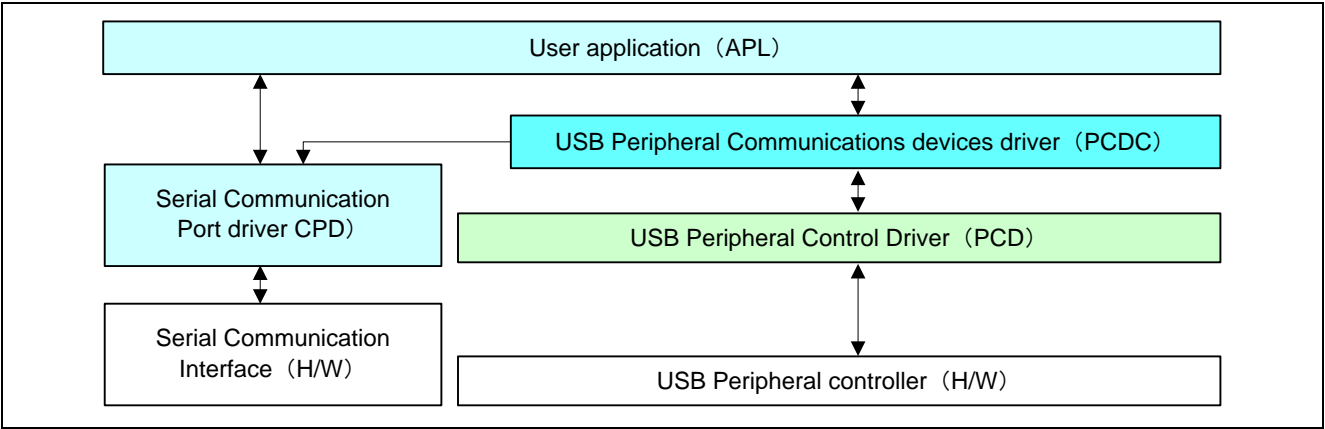


Figure 2-1 モジュール構成図

Table 2-1 各モジュール機能概要

モジュール名	機能概要
PCDC	APL からの CDC に関するリクエストおよび、データ通信を PCD へ要求します。
PCD	USB Peripheral H/W 制御ドライバです。
CPD	シリアルポートの制御ドライバです。

3. API 情報

本ドライバの API はルネサスの API の命名基準に従っています。

3.1 ハードウェアの要求

ご使用になる MCU が以下の機能をサポートしている必要があります。

- USB

3.2 ソフトウェアの要求

このドライバは以下のパッケージに依存しています。

- r_bsp
- r_usb_basic

3.3 動作確認環境

このドライバの動作確認環境を以下に示します。

Table 3-1 動作確認環境

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e ² studio V.6.0.0
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 V2.07.00 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 -lang = c99
エンディアン	リトルエンディアン / ビッグエンディアン
モジュールのバージョン	Rev.1.22
使用ボード	Renesas Starter Kit for RX63N Renesas Starter Kit for RX64M Renesas Starter Kit for RX71M Renesas Starter Kit for RX65N, Renesas Starter Kit for RX65N-2MB
ホスト環境	下記の OS に接続し動作確認を行っています。 1. Windows® 7 2. Windows® 8.1 3. Windows® 10

3.4 使用する割り込みベクタ

このドライバが使用する割り込みベクタを以下に示します。

Table 3-2 使用する割り込みベクター一覧

デバイス	割り込みベクタ
RX63N RX631	USBI0 割り込み(ベクタ番号: 35) / USBR0 割り込み(ベクタ番号: 90)
	USB D0FIFO0 割り込み(ベクタ番号: 33) / USB D1FIFO0 割り込み(ベクタ番号: 34)
	USBI1 割り込み(ベクタ番号: 38) / USBR1 割り込み(ベクタ番号: 91)
	USB D0FIFO1 割り込み(ベクタ番号: 36) / USB D1FIFO1 割り込み(ベクタ番号: 37)

RX64M RX71M	USBIO(GROUPB)割り込み(ベクタ番号: 189, グループ割り込み要因番号: 62) USB D0FIFO0 割り込み(ベクタ番号: 34) / USB D1FIFO0 割り込み(ベクタ番号: 35) USBRO 割り込み(ベクタ番号:90)
	USBAR 割り込み(ベクタ番号: 94) USB D0FIFO2 割り込み(ベクタ番号: 32) / USB D1FIFO2 割り込み(ベクタ番号: 33)
RX65N RX651	USBIO(GROUPB)割り込み(ベクタ番号: 185, グループ割り込み要因番号: 62) USB D0FIFO0 割り込み(ベクタ番号: 34) / USB D1FIFO0 割り込み(ベクタ番号: 35) USBRO 割り込み(ベクタ番号:90)

3.5 ヘッドファイル

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_usb_basic_if.h` と `r_usb_pcdc_if.h` に記載されています。

3.6 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。

3.7 コンパイル時の設定

コンパイル時の設定については、「6. コンフィグレーション (`r_usb_pcdc_config.h`)」章および USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No. R01AN2025)の「コンフィグレーション」章を参照してください。

3.8 ROM / RAM サイズ

本ドライバの ROM/RAM サイズを以下に示します。

1. RX64M, RX71M, RX65N/RX651

	引数チェック実施時	引数チェック非実施時
ROM サイズ	19.7K バイト (Note 3)	19.3K バイト (Note 4)
RAM サイズ	9.4K バイト	9.4K バイト

2. RX63N/RX631

	引数チェック実施時	引数チェック非実施時
ROM サイズ	16.1K バイト (Note 3)	15.6K バイト (Note 4)
RAM サイズ	9K バイト	9K バイト

[Note]

- 上記のサイズには、BSP および USB Basic Driver の ROM/RAM サイズが含まれています。
- コンパイラの最適化オプションには、Default オプションが指定されています。
- 「引数チェック実施時」の ROM サイズは、`r_usb_basic_config.h` ファイル内の `USB_CFG_PARAM_CHECKING` 定義に対し `USB_CFG_ENABLE` を指定した時の値です。
- 「引数チェック非実施時」の ROM サイズは、`r_usb_basic_config.h` ファイル内の `USB_CFG_PARAM_CHECKING` 定義に対し `USB_CFG_DISABLE` を指定した時の値です。

3.9 引数

API 関数の引数に使用される構造体については、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)内の「構造体」の章を参照してください。

3.10 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)の追加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 ユーザーガイド (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 ユーザーガイド (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。

4. コミュニケーションデバイスクラス（CDC）、PSTN and ACM

4.1 基本機能

CDC は、コミュニケーションデバイスクラス仕様 Abstract Control Model サブクラス（4.2章参照）に準拠しています。

4.2 Abstract Control Model 概要

Abstract Control Model サブクラスは、USB 機器と従来のモデム（RS-232C 接続）との間を埋める技術で、従来のモデムを使用するアプリケーションプログラムが使用可能です。

以下に本 S/W でサポートするクラスリクエスト・クラスノーティフィケーションを記します。

4.2.1 クラスリクエスト（ホスト→デバイスへの通知）

本ドライバは以下のクラスリクエストを受信するとアプリケーションプログラムに通知します。

アプリケーションプログラムでのクラスリクエスト処理については、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)内の「クラスリクエスト」の章を参照してください。

Table 4-1 CDC クラスリクエスト

リクエスト	コード	説明
SetLineCoding	0x20	通信回線設定を行う。（通信速度,データ長,パリティビット,ストップビット長）
GetLineCoding	0x21	通信回線設定状態を取得する。
SetControlLineState	0x22	通信回線制御信号 RTS、DTR の設定を行う。

Abstract Control Model リクエストについては、USB Communications Class Subclass Specification for PSTN Devices Revision 1.2 の Table11 : Requests-Abstract Control Model を参照して下さい。

4.2.2 クラスリクエストのデータフォーマット

CDC が対応するクラスリクエストのデータフォーマットを以下に記します。

1. SetLineCoding

UART 回線設定を行う為にホストがデバイスに対して送信するクラスリクエストです。

SetLineCoding データフォーマットを以下に示します。

Table 4-2 SetLineCoding フォーマット

bmRequestType t	bReques	wValue	wIndex	wLength	Data
0x21	SET_LINE_CODING (0x20)	0x00	0x00	0x07	Line Coding Structure Table 4-3 Line Coding Structure フォーマット参照

Table 4-3 Line Coding Structure フォーマット

Offset	Field	Size	Value	Description
0	DwDTERate	4	Number	データ端末の速度 (bps)
4	BcharFormat	1	Number	ストップビット 0 - 1 Stop bit 1 - 1.5 Stop bit 2 - 2 Stop bit
5	BparityType	1	Number	パリティ 0 - None 1 - Odd 2 - Even
6	BdataBits	1	Number	データビット (5、6、7、8)

2. GetLineCoding

UART 回線設定状態を要求する為にホストがデバイスに対して送信するクラスリクエストです。

GetLineCoding データフォーマットを以下に示します。

Table 4-4 GetLineCoding フォーマット

bmRequestType t	bReques	wValue	wIndex	wLength	Data
0xA1	GET_LINE_CODING (0x21)	0x00	0x00	0x07	Line Coding Structure Table 4-3 Line Coding Structure フォーマット参照

3. SetControlLineState

UART のフロー制御用信号を設定する為にホストがデバイスに対して送信するクラスリクエストです。

本 S/W では RTS/DTR の制御をサポートしていません。

SET_CONTROL_LINE_STATE データフォーマットを以下に示します。

Table 4-5 SET_CONTROL_LINE_STATE フォーマット

bmRequestType t	bReques	wValue	wIndex	wLength	Data
0x21	SET_CONTROL_ LINE_STATE (0x22)	Control Signal Bitmap Table 4-6 Control Signal Bitmap フォーマット参照	0x00	0x00	None

Table 4-6 Control Signal Bitmap フォーマット

Bit Position	Description
D15~D2	予約 (0 にリセット)
D1	DCE の送信機能を制御 0 - RTS OFF 1 - RTS ON
D0	DTE がレディ状態かの通知 0 - DTR OFF 1 - DTR ON

4.2.3 クラスノーティフィケーション（デバイス→ホストへの通知）

本 S/W のクラスノーティフィケーション対応/非対応をTable 4-7下表に示します。

Table 4-7 CDC クラスノーティフィケーション

ノーティフィケーション	コード	説明	対応
NETWORK_CONNECTION	0x00	ネットワーク接続状況を通知する	×
RESPONSE_AVAILABLE	0x01	GET_ENCAPSLATED_RESPONSE への応答	×
SERIAL_STATE	0x20	シリアル回線状態を通知する	○

1. SerialState

UART ポートに状態変化を検出した場合、ホストへ状態通知を行います。

本 S/W ではオーバーランエラー、パリティエラー、フレーミングエラー検出をサポートしています。状態通知は正常状態からエラー検出した場合に行います。エラーを連続検出しても状態通知を連続送信しません。

SerialState データフォーマットを以下に示します。

Table 4-8 SerialState フォーマット

bmRequestType	bReques	wValue	wIndex	wLength	Data
0xA1	SERIAL_STATE (0x20)	0x00	0x00	0x02	UART State bitmap Table 4-9 UART State bitmap フォーマット参照

Table 4-9 UART State bitmap フォーマット

Bits	Field	Description	対応
D15～D7		予約	—
D6	b_over_run	オーバーランエラー検出	○
D5	b_parity	パリティエラー検出	○
D4	b_framing	フレーミングエラー検出	○
D3	b_ring_signal	着信（Ring signal）を感知した	×
D2	b_break	ブレイク信号検出	×
D1	btx_carrier	Data Set Ready：回線が接続されて通信可能	×
D0	brx_carrier	Data Carrier Detect：回線にキャリア検出	×

4.3 PC の仮想 COM ポートについて（参考）

Windows OS 搭載 PC は CDC デバイスを仮想 COM ポートとして利用することが可能です。

Windows OS 搭載 PC に本 S/W を実装した RSK ボードを接続すると、エニユメレーション設定に続き、CDC クラスリクエストの **GetLineCoding** 及び **SetControlLineState** を行った後、仮想 COM デバイスとしてデバイスマネージャに登録されます。Windows デバイスマネージャに仮想 COM ポートとして登録された後は、WindowsOS 標準搭載のハイパーターミナル等のターミナルアプリで CDC デバイスとデータ通信が可能です。

ターミナルアプリのシリアルポート設定を行うことで、クラスリクエスト **SetLineCoding** による UART 設定が可能です。ターミナルアプリのウインドウから入力したデータ（又はファイル送信）は EP2 を使用して RSK ボードへ転送され、RSK ボード側から PC へのデータ転送は EP1 を使用して行われます。

ターミナルアプリによっては最後に受信したデータが MAX パケットサイズの場合、継続するデータがあると判断して受信データをターミナルに表示しないことがあります。この場合、MAX パケットサイズ未満のデータを受信することで、それまでに受信したデータがターミナルに表示されます。

5. API

アプリケーションプログラム内で使用する API については、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)内の「API」の章を参照してください。

6. コンフィグレーション (r_usb_pcdc_config.h)

お客様のシステムにあわせて以下の設定をお願いします。

[Note]

必ず r_usb_basic_config.h ファイルに対する設定もお願いします。r_usb_basic_config.h については、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)内の「コンフィグレーション」の章を参照してください。

1. 使用パイプ設定

データ転送で使用するパイプ番号を設定してください。

(1). Bulk IN, Bulk OUT 転送

Bulk IN, Bulk OUT 転送で使用するパイプ番号(PIPE1 から PIPE5)を指定してください。なお、USB_CFG_PCDC_BULK_IN と USB_CFG_PCDC_BULK_OUT に対し、同じパイプ番号は指定しないでください。

#define	USB_CFG_PCDC_BULK_IN	パイプ番号 (USB_PIPE1 から USB_PIPE5)
#define	USB_CFG_PCDC_BULK_OUT	パイプ番号 (USB_PIPE1 から USB_PIPE5)

(2). Interrupt IN 転送

Interrupt IN 転送で使用するパイプ番号(PIPE6 から PIPE9)を指定してください。

#define	USB_CFG_PCDC_INT_IN	パイプ番号 (USB_PIPE6 から USB_PIPE9)
---------	---------------------	--------------------------------

7. アプリケーションの作成方法

USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)内の「アプリケーションプログラムの作成方法」の章を参照してください。

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	Aug 1, 2014	—	初版発行
1.10	Sep 30, 2015	—	対象デバイスに RX71M を追加。
1.20	Sep 30, 2016	—	1. 対象デバイスに RX65N/RX651 を追加 2. DMA 転送をサポート 3. USB Host and Peripheral Interface Driver アプリケーションノート (ドキュメント No.R01AN3293JJ)に対応
1.21	Mar 31, 2017	—	1. Technical Update(発行番号: TN-RX*-A172A/J)に対応しました。 2. 「API」の章を USB Basic Host and Peripheral Driver Firmware Integration Technology (ドキュメント番号:R01AN2025)のドキュメントへ移しました。
1.22	Sep 30, 2017	—	RX65N/RX651-2M をサポート

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。

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うと、内部ROM、レイアウトパターンの相違などにより、電气的特性の範囲で、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。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、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。
2. 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。
3. 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に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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）、交通用信号機器、防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
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。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。
9. 本資料に記載されている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。また、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に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
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。

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/>