RXファミリ

R01AN2297JJ0132
Rev.1.32
May 15, 2026

USB ペリフェラル ヒューマンインタフェースデバイスクラスドライバ(PHID) for USB Mini Firmware による USB ホストとの USB 通信を行うサンプルプログラム Firmware Integration Technology

要旨

本資料は、USB Peripheral Human Interface Devices Class Driver for USB Mini Firmware Firmware Integration Technology を使用したサンプルファームウェアの説明資料です。以降、本サンプルファームウェアを PHID と記述します。

実際のソフトウェア開発時には、必ず" USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) Firmware Integration Technology アプリケーションノート"(Document No:R01AN2166)および各マイコンのユーザーズマニュアル(ハードウェア編)と併用してご利用ください。また、必要に応じて USB Peripheral Human Interface Device Class Driver for USB Mini Firmware Firmware Integration Technology アプリケーションノート"(Document No:R01AN2171)も参照してください。なお、USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) Firmware Integration Technology アプリケーションノート"(Document No:R01AN2166)は、パッケージ内の"reference_documents"フォルダにあります。

対象デバイス

RX111 グループ
RX113 グループ
RX231 グループ
RX23W グループ
RX261 グループ

本プログラムは Renesas Starter Kits (RSK)、Renesas Solution Starter Kit(RSSK)または EK を使って動作確認を行っています。

目次

1. はじめに 2

2. ソフトウェア構成 4

3. セットアップ 5

4. サンプルアプリケーション 9

5. ディスクリプタ 22

6. クラスリクエスト 23

7. RI600V4プロジェクトをCS+で使用する場合 24

8. e² studio用プロジェクトをCS+で使用する場合 29

1. はじめに

1.1 機能概要

PHID は、USB ヒューマンインターフェースデバイスクラス仕様（以降 HID と記述）に準拠し、USB Host と通信を行うことが可能です。

PHID の機能を以下に示します。

1. Full-Speed (フル・スピード：12Mbps)デバイスとして動作します。
2. RSK / RSSK/EK を USB Host に接続すると、RSK / RSSK/EK が HID デバイスとして認識され、USB Host とのデータ転送を行います。

1.2 FIT モジュール構成

PHID は以下の FIT モジュールとサンプルアプリケーションで構成されています。

Table 1-1 FIT モジュール構成

FIT モジュール名	フォルダ名
RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology	r_bsp
RX ファミリ USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) Firmware Integration Technology	r_usb_basic_mini
RX ファミリ USB Peripheral Human Interface Devices Class Driver for USB Mini Firmware Firmware Integration Technology	r_usb_phid_mini
RX Family LPC (Low Power Consumption) Module Firmware Integration Technology	r_lpc_rx
RX ファミリ IRQ モジュール Firmware Integration Technology	r_irq_rx

各 FIT モジュールの詳細は、関連ドキュメントを参照してください。また、本サンプルファームウェアで使用している FIT モジュールの最新バージョンは下記のホームページよりダウンロードが可能です。

ルネサスエレクトロニクスホームページ http://japan.renesas.com/

1.3 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。

1.4 動作確認済環境

PHID を動作させるために必要な環境を以下に示します。

Table 1-2　動作確認環境

項目	内容
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 (統合開発環境のデフォルト設定に"-lang = c99"オプションを追加)
	GCC for Renesas RX (統合開発環境のデフォルト設定に"-std = gnu99"オプションを追加)
	IAR C/C++ Compiler for Renesas
リアルタイム OS	FreeRTOS RI600V4
エンディアン	リトルエンディアン / ビッグエンディアン
モジュールのリビジョン	Rev.1.32
使用ボード	Renesas Starter Kit for RX111 Renesas Starter Kit for RX113 Renesas Starter Kit for RX231 Renesas Solution Starter Kit for RX23W EK-RX261
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® 11

2. ソフトウェア構成

2.1 モジュール構成

PHID は、HID クラスドライバとデバイスドライバから構成されます。APL からデータ転送要求により場合、PCD を介して USB Host とのデータ転送を行います。

Figure 2-1に PHID 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。

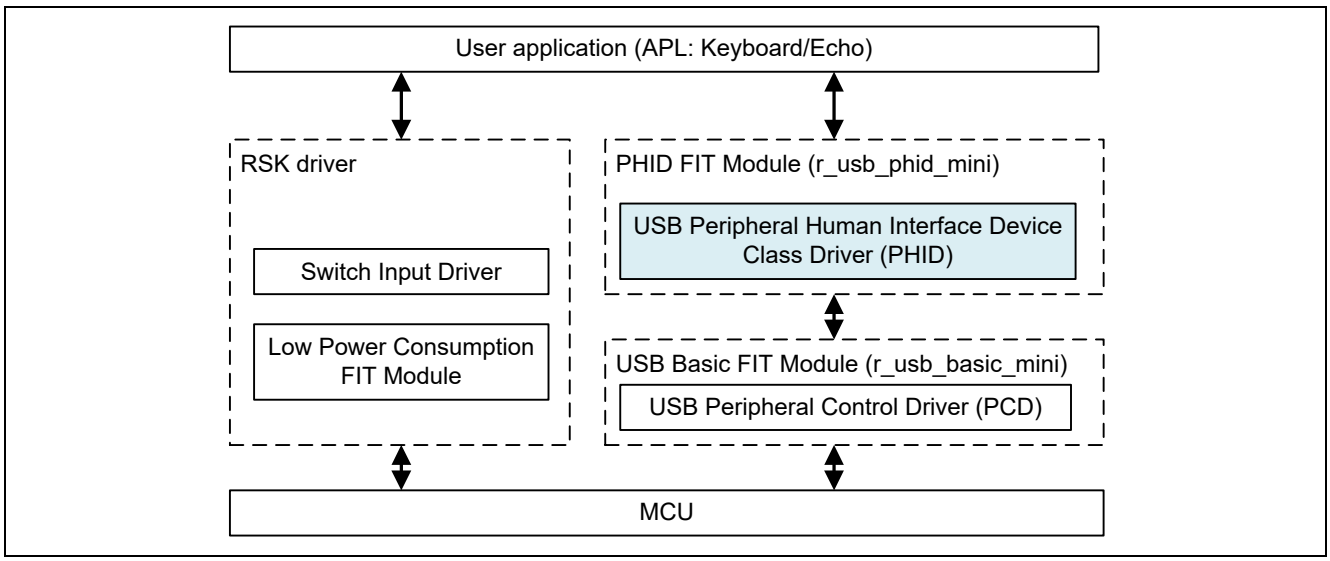


Figure 2-1　モジュール構成図

Table 2-1　モジュール機能概要

モジュール名	機能概要
APL	デモサンプルアプリケーションプログラム
PHID (r_usb_phid_mini)	USB Host からの要求を解析します。 APL からの要求を PCD 経由で USB Host に通知します。
PCD (r_usb_basic_mini)	USB Peripheral H/W 制御ドライバ

3. セットアップ

3.1 ハードウェア

3.1.1 動作環境例

PHID の動作環境例をFigure 3-1に示します。評価ボードのセットアップ、エミュレータなどの使用方法については各取扱説明書を参照してください。

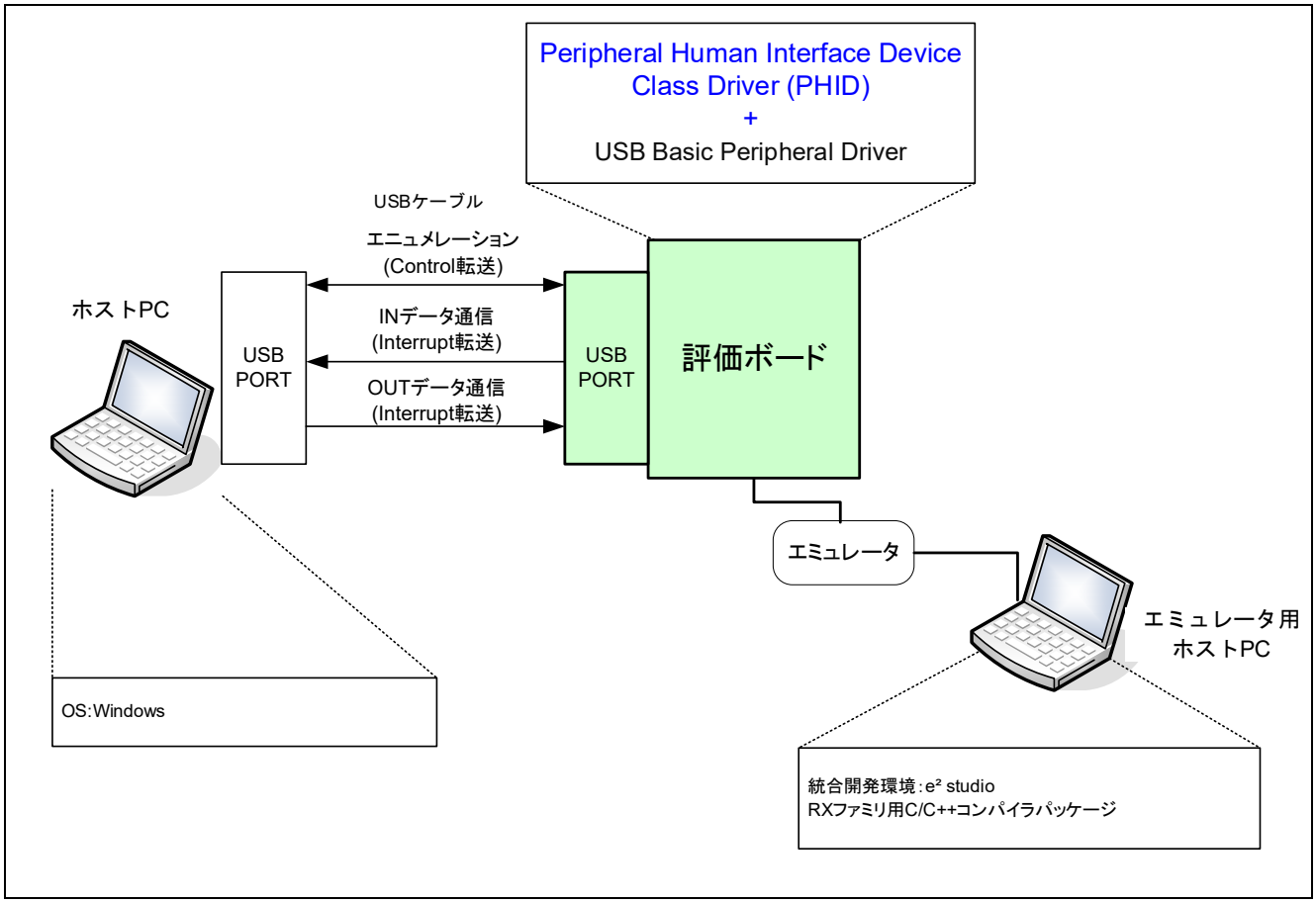


Figure 3-1　動作環境例

動作確認済みの評価ボードをTable 3-1に示します。

Table 3-1　PHID 動作確認済みの評価ボード

マイコン	評価ボード
RX111	RSKRX111
RX113	RSKRX113
RX231	RSKRX231
RX23W	RSSKRX23W
RX261	EK-RX261

3.1.2 RSK / RSSK / EK 設定

RSK / RSSK / EK を USB Peripheral モードに設定する必要があります。設定内容は以下を参照してください。

Table 3-2 **ジャンパ設定**

RSK / RSSK	ジャンパ設定
RSKRX111	J12: Shorted Pin2-3
RSKRX113	J12: Shorted Pin2-3
RSKRX231	J15: Shorted Pin2-3
RSSKRX23W	J5: Shorted Pin1-2
EK-RX261	J18: Shorted Pin2-3, J19: Shorted Pin2-3

[Note]

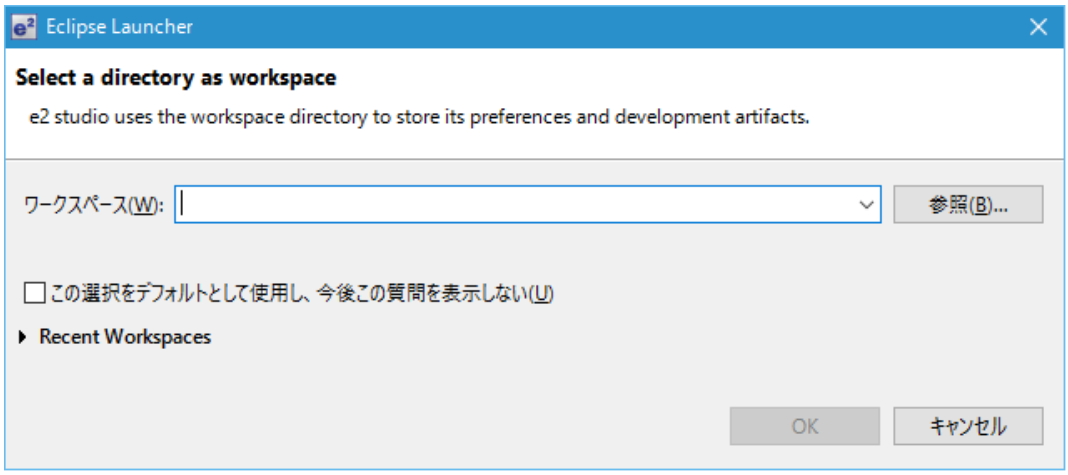
RSK / RSSK /EK ジャンパ設定の詳細については、RSK / RSSK /EK のユーザーズマニュアルを参照してください。

3.2 ソフトウェア

(1). e² studio を起動

a) e² studio を起動してください。

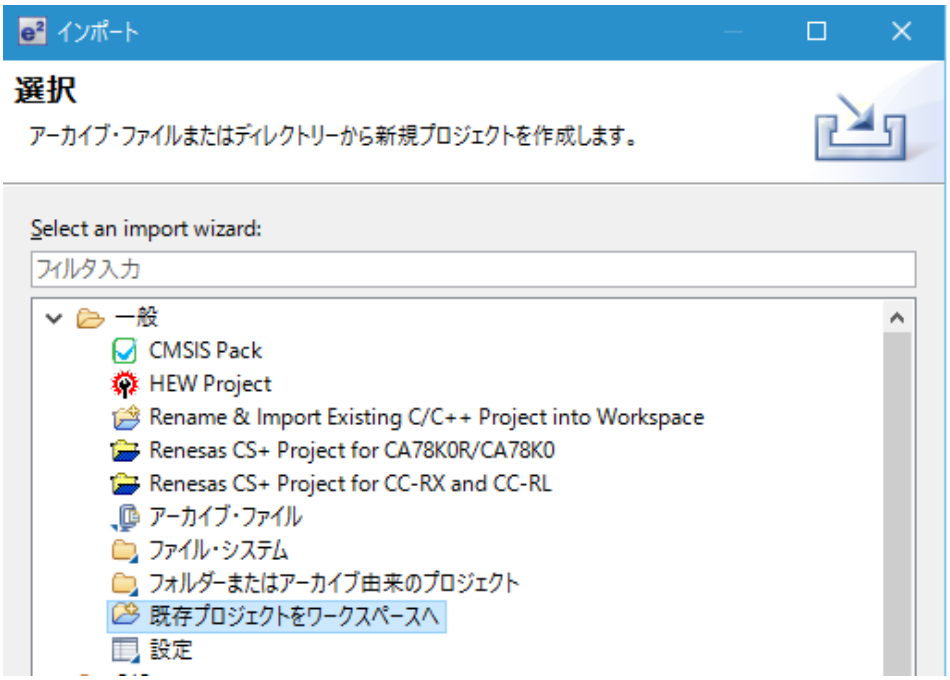
b) はじめて e² studio を起動する場合、Eclipse Launcher ダイアログが表示されますので、プロジェクトを格納するためのフォルダを指定してください。



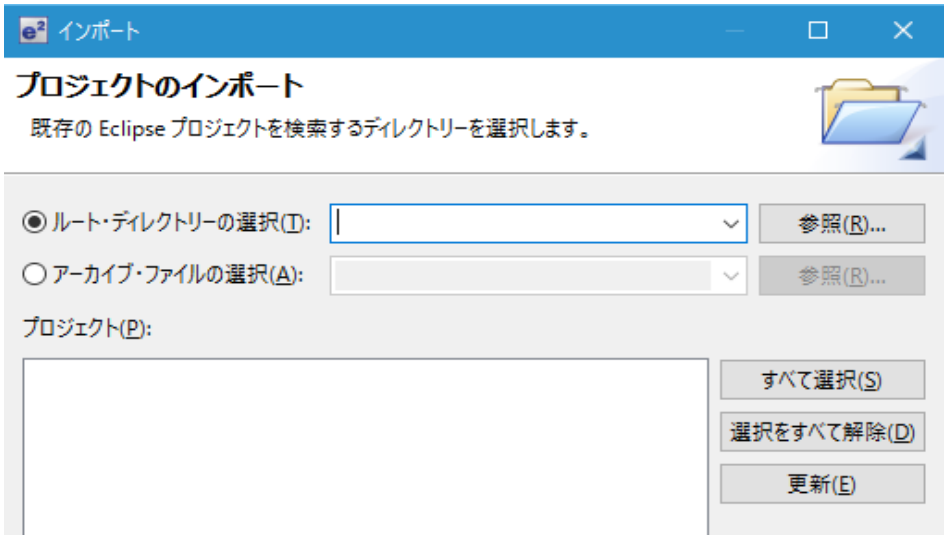
(2). プロジェクトをワークスペースへインポート

a) [ファイル] --> [インポート]を選択してください。

b) [一般] => [既存プロジェクトをワークスペースへ]を選択してください。



c) プロジェクトファイル".cproject"が格納されたフォルダを”Select root directory”に入力してください。



d) “Finish”をクリック

プロジェクトのワークスペースへのインポートが完了しました。同様の方法で他のプロジェクトを同一のワークスペースへインポートすることができます。

(3). “Build”ボタンをクリックし、実行プログラムを生成してください。

(4). デバッガへの接続を行い、実行プログラムをダウンロードしてください。“Run”ボタンをクリックすると、プログラムが実行されます。

4. サンプルアプリケーション

4.1 アプリケーション仕様

PHID のサンプルアプリケーション(以降、APL)の主な機能を以下に示します。

1. Keyboard mode: キーボード機能

   RSK/RSSK/EK を USB Host に接続すると、USB Host は RSK/RSSK/EK をキーボードとして認識します。 RSK/RSSK/EK はキーボードとして動作し、Interrupt IN 転送によりキーボードデータを USB Host に送信します。

2. Echo mode: USB ループバック機能(Interrupt IN/OUT データ転送)

   RSK/RSSK/EK を USB Host に接続し、USB Host との Interrupt IN/OUT のデータ転送を行います。この機能では、USB Host から受信したデータをそのまま USB Host へ送信する処理を行います。

3. 消費電力低減機能

   USB の状態に応じて MCU を消費電力低減モードに遷移させる機能です。

   (1). USB サスペンド状態時に MCU をスリープモードに遷移させます。

   (2). USB デタッチ（切断）状態時に、MCU をソフトウエアスタンバイモードに遷移させます。

[Note]

1. Keyboard mode/Echo mode の選択は、r_usb_phid_apl_config.h 内で行えます。
2. 消費電力低減機能の有効/無効設定は、r_usb_phid_apl_config.h 内で行えます。
3. Echo mode の場合は、USB ループバック機能をサポートしている USB Host と通信を行います。Keyboard mode の場合は、Windows 10 の OS をサポートしている PC(USB Host)との USB 通信が可能です。

4.2 アプリケーション処理概要 (Non-OS)

APL は初期化処理、メインループの 2 つの部分から構成されます。以下にそれぞれの処理概要を示します。

4.2.1 初期設定

初期設定では、MCU の端子設定、USB ドライバの設定、USB コントローラの初期設定を行います。

4.2.2 メインループ (Keyboard mode)

Keyboard mode では、RSK/RSSK/EK 上のスイッチ情報を USB Host に送信する処理が行われます。RSK/RSSK/EK (HID デバイス)を USB Host(PC)に接続すると、RSK/RSSK/EK がキーボードとして認識され、メインループはスイッチ情報をキー入力データとして USB Host に送信する処理をメインに行います。スイッチ情報(キー入力データ)については、「4.6.1 スイッチ仕様」を参照してください。

1. USB Host との Enumeration が完了すると USB Host は、HID クラスリクエストを HID デバイスに送信します。HID デバイスが HID クラスリクエストを受信した後、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_REQUST がセットされます。APL では、USB_STS_REQUEST を確認すると、受信したクラスリクエストを解析し、そのクラスリクエストに対応する処理を行います。
2. 上記1でのクラスリクエスト処理完了後、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_REQUST_COMPLETE がセットされます。APL では、リクエスト情報の設定処理等を行っています。
3. APL は、RSK/RSSK 上のスイッチ押下を確認します。スイッチが押下されていれば、状態管理変数を参照しデータ送信中かどうかを確認します。送信中でなければ、R_USB_Write 関数をコールし、押下されたスイッチをキー情報として送信します。
4. 上記3での HID データ送信完了後、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_WRITE_COMPLETE がセットされます。APL では、USB_STS_WRITE_COMPLETE を確認すると R_USB_Write 関数をコールし、ゼロデータ(8 バイト)の送信要求を行います。(キーボードの場合、キー入力がリリースされたことを USB Host に通知するためゼロデータの送信が必要です。)
5. 上記4でのゼロデータ送信完了後、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_WRITE_COMPLETE がセットされます。APL では、USB_STS_WRITE_COMPLETE を確認するとゼロデータ送信が完了したかどうかを確認し、ゼロデータであれば、状態管理変数に対する設定(NO_WRITING)を行います。
6. 上記1から5までの処理を繰り返している間に、USB Host からのサスペンド信号の受信や DETACH を確認すると、APL は HID デバイス(RSK/RSSK)を低消費電力モードに移行するための処理を行います。消費電力低減モードについては、「4.4 MCU消費電力低減処理」を参照してください。なお、サスペンド信号受信や DETACH 確認は R_USB_GetEvent 関数の戻り値(USB_STS_SUSPEND / USB_STS_DETACH)により行います。

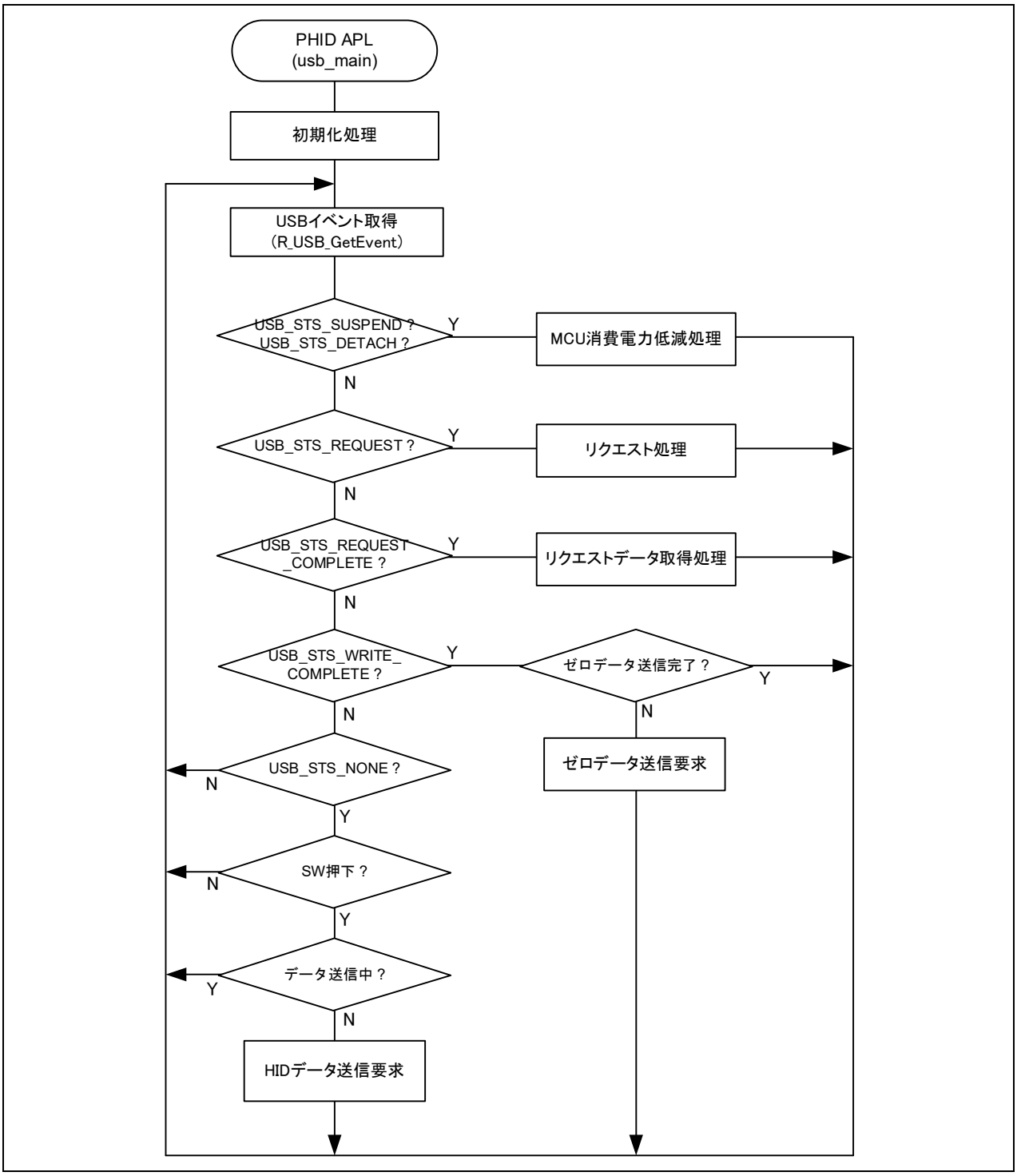


Figure 4-1　メインループ処理 (Keyboard mode)

4.2.3 メインループ (Echo mode)

Echo mode のメインループでは、USB Host から送信されるデータを受信し、そのまま USB Host へ送信するループバック処理をメインに行います。以下にメインループの処理概要を示します。

1. USB Host との Enumeration 完了後に R_USB_GetEvent 関数をコールすると戻り値に USB_STS_CONFIGURED がセットされます。APL では、USB_STS_CONFIGURED を確認すると R_USB_Read 関数をコールし、USB Host から送信されるデータのデータ受信要求を行います。
2. USB Host からのデータ受信が完了し、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_READ_COMPLETE がセットされます。APL では、USB_STS_READ_COMPLETE を確認すると R_USB_Write 関数をコールし、受信データを USB Host に送信するため R_USB_Write 関数をコールします。
3. USB Host へのデータ送信が完了し、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_WRITE_COMPLETE がセットされます。APL では、USB_STS_WRITE_COMPLETE を確認すると R_USB_Read 関数をコールし、USB Host から送信されるデータのデータ受信要求を行います。
4. 上記2と3の処理が繰り返し行われます。
5. USB Host からのサスペンド信号の受信や DETACH を確認すると、APL は CDC デバイス(RSK/RSSK)を低消費電力モードに移行するための処理を行います。消費電力低減モードについては、「4.4 MCU 消費電力低減処理」を参照してください。なお、サスペンド信号の受信や DETACH の確認は、R_USB_GetEvent 関数の戻り値(USB_STS_SUSPEND/USB_STS_DETACH)により行います。

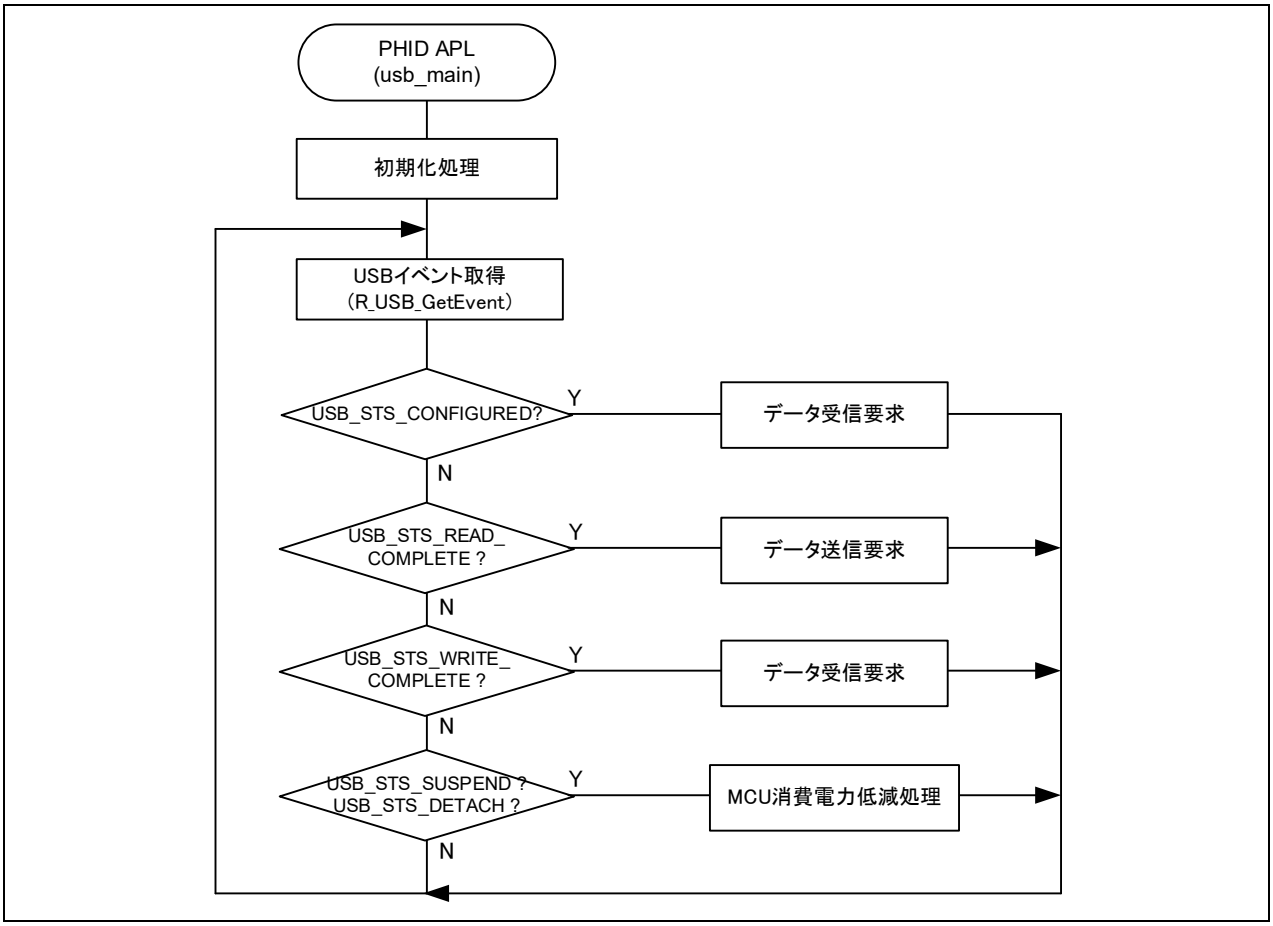


Figure 4-2 メインループ処理 (Echo mode)

4.3 アプリケーション処理概要 (RTOS)

APL は初期化処理、メインループの 2 つの部分から構成されます。以下にそれぞれの処理概要を示します。

4.3.1 初期設定

初期設定では、MCU の端子設定、USB ドライバの設定、USB コントローラの初期設定を行います。

4.3.2 メインループ (Keyboard mode)

Keyboard mode では、ボード上のスイッチ情報を USB Host に送信する処理が行われます。ボード(HID デバイス)を USB Host(PC)に接続すると、RSK がキーボードとして認識され、メインループはスイッチ情報をキー入力データとして USB Host に送信する処理をメインに行います。スイッチ情報(キー入力データ)については、「4.6.1 スイッチ仕様」を参照してください。

1. USB 関連のイベントが完了すると USB ドライバはコールバック関数(usb_apl_callback)をコールします。コールバック関数(usb_apl_callback)では、リアルタイム OS の機能を使って USB 完了イベントを APL(アプリケーションタスク)に通知します。
2. APL ではコールバック関数から通知された USB 完了イベント等の情報を FreeRTOS の機能を使って取得します。
3. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_REQUEST の場合、APL はこの USB 完了イベントをもとに、受信したリクエストに対応する処理を行います。
4. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_REQUEST_COMPLETE の場合、APL は、リクエスト情報の設定処理等を行います。
5. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_WRITE_COMPLETE の場合、APL では、送信完了したデータがゼロデータかどうかを確認します。なお、送信処理は、下記(1)で行われます。
   (1). 送信完了したデータがゼロデータではない場合、APL は R_USB_Write 関数をコールし、ゼロデータ(8 バイト)の送信要求を行います。(キーボードの場合、キー入力がリリースされたことを USB Host に通知するためゼロデータの送信が必要です。)
   (2). 送信完了したデータがゼロデータの場合、状態管理変数に対する設定(NO_WRITING)を行います。
6. 上記で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_SUSPEND または、USB_STS_DETACH の場合、APL は HID デバイス(RSK)を低消費電力モードに移行するための処理を行います。消費電力低減モードについては、「4.4 MCU消費電力低減処理」を参照してください。
7. 上記の処理後、APL は RSK 上のスイッチ押下を確認します。押下されたスイッチが SW2/SW3 の場合、状態管理変数を参照し、データ送信中かどうかを確認します。送信中でなければ、R_USB_Write 関数をコールし、押下されたスイッチをキー情報として送信します

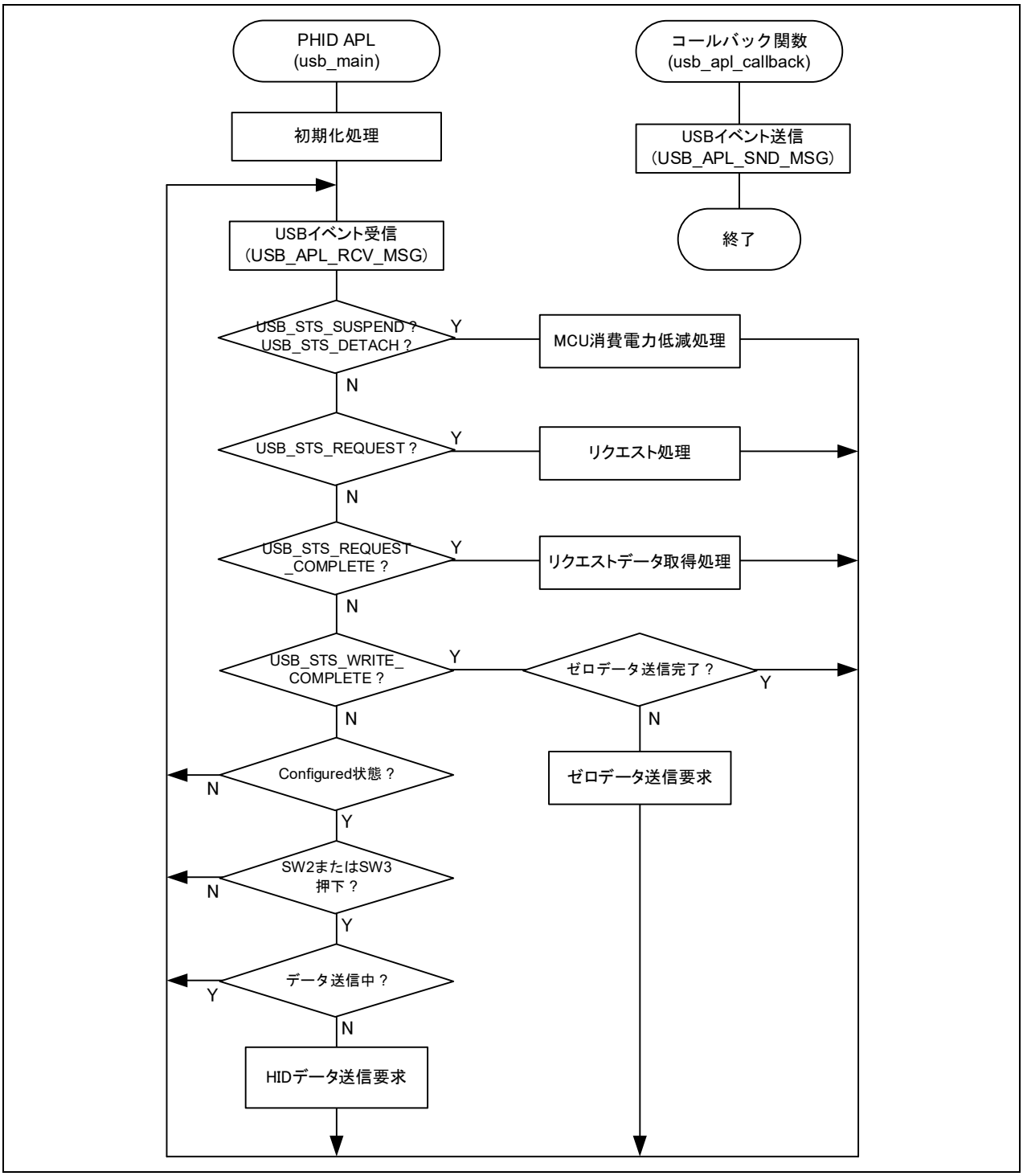


Figure 4-3　メインループ処理 (Keyboard mode)

4.3.3 メインループ (Echo mode)

Echo mode のメインループでは、USB Host から送信されるデータを受信し、そのまま USB Host へ送信するループバック処理をメインに行います。以下にメインループの処理概要を示します。

1. USB 関連のイベントが完了すると USB ドライバはコールバック関数(usb_apl_callback)をコールします。コールバック関数(usb_apl_callback)では、リアルタイム OS の機能を使って USB 完了イベントを APL(アプリケーションタスク)に通知します。
2. APL ではコールバック関数から通知された USB 完了イベント等の情報をリアルタイム OS の機能を使って取得します。
3. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_CONFIGURED の場合、APL はこの USB 完了イベントをもとに、R_USB_Read 関数をコールし、USB Host からの送信されるデータのデータ受信要求を行います。
4. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_READ_COMPLETE の場合、APL は R_USB_Write 関数をコールし、受信したデータを USB Host へ送信するためのデータ送信要求を行います。
5. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_REQUEST の場合、APL はこの USB 完了イベントをもとに、受信したリクエストに対応する処理を行います。
6. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_SUSPEND または、USB_STS_DETACH の場合、APL は CDC デバイス(RSK)を低消費電力モードに移行するための処理を行います。消費電力低減モードについては、「4.4 MCU消費電力低減処理」を参照してください。
7. 上記の処理が繰り返し行われます。

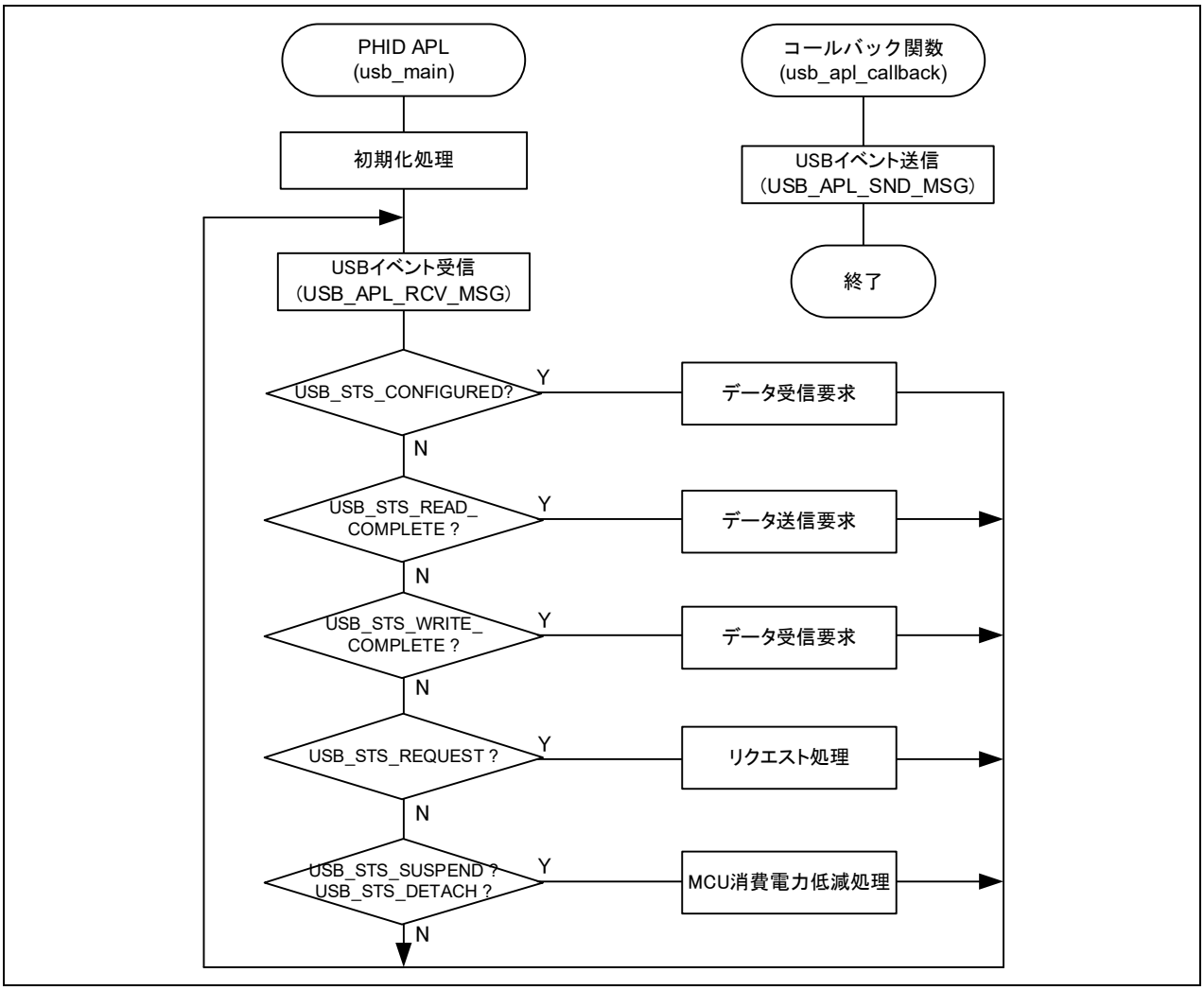


Figure 4-4　メインループ処理 (Echo mode)

4.4 MCU 消費電力低減処理

MCU 消費電力低減処理は、Table 4-1/Table 4-2の条件が成立すると消費電力低減モードに移行する処理を行います。なお、この処理を有効にするには、“r_usb_phid_apl_config.h”ファイル内の“USB_SUPPORT_LPW”定義に対し"USB_APL_ENABLE"を指定してください。

1. Non-OS

Table 4-1 消費電力低減機能状態遷移条件

遷移条件		遷移状態
VBUS	USB ステート	
OFF	ー	ソフトウエアスタンバイモード
ON	Suspend Configured	スリープモード
ON	Suspend Configured 以外	通常モード（プログラム実行状態）

(1). HID デバイス(RSK/RSSK)が USB Host からデタッチ（VBUS OFF）されると、APL は MCU をソフトウエアスタンバイモードに遷移するための処理を行います。ソフトウエアスタンバイモードからの復帰は、HID デバイス(RSK/RSSK/EK)を USB Host にアタッチすることにより行われます。

(2). HID デバイス(RSK/RSSK/EK)を USB Host に接続した状態で、USB Host から送信されるサスペンド信号を受信すると APL は、MCU をスリープモードに遷移するための処理を行います。なお、スリープモードからの復帰は、USB Host から送信されるレジューム信号の受信により行われます。

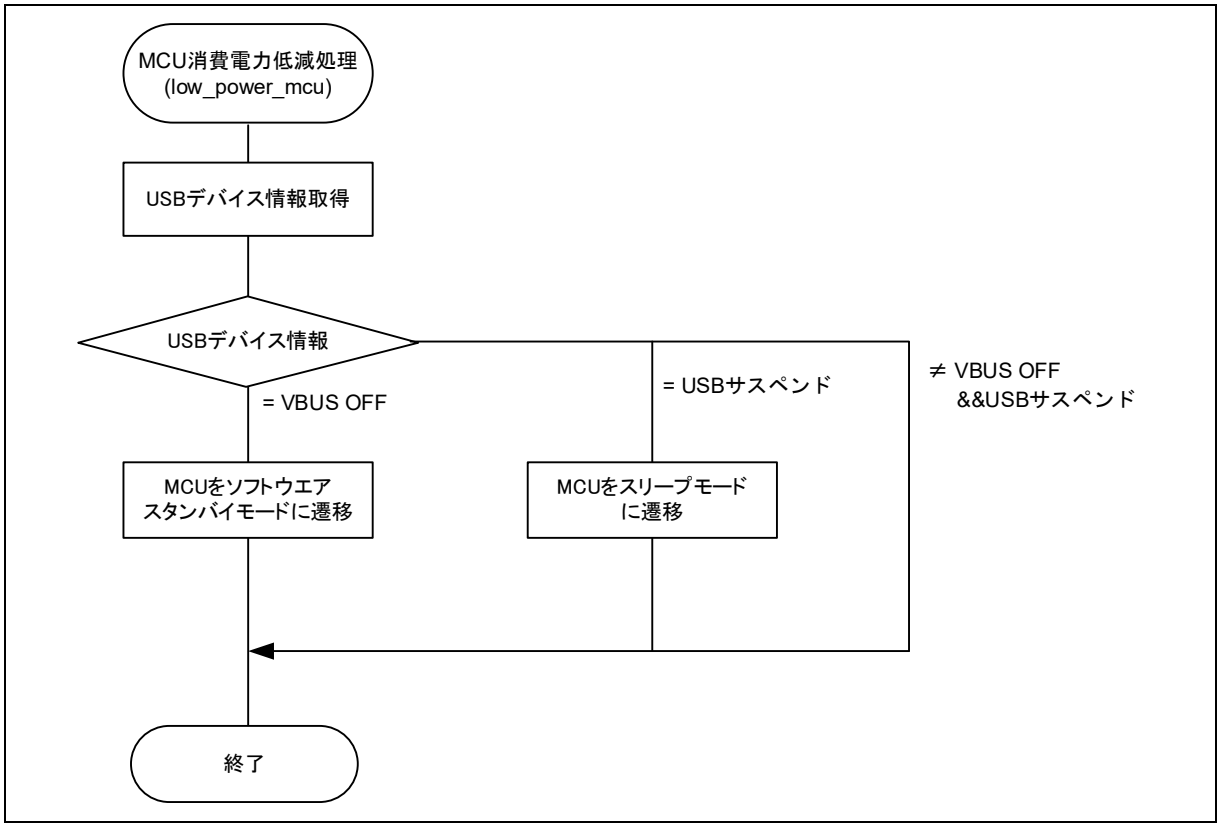


Figure 4-5 消費電力低減処理フロー

2. RTOS (FreeRTOS のみ)

Table 4-2 消費電力低減機能状態遷移条件

遷移条件		遷移状態
VBUS	USB ステート	
OFF	ー	ソフトウエアスタンバイモード
ON	Suspend Configured	ソフトウエアスタンバイモード
ON	Suspend Configured 以外	通常モード（プログラム実行状態）

(1). HID デバイス(RSK)が USB Host からデタッチ（VBUS OFF）されると、APL は MCU をソフトウエアスタンバイモードに遷移するための処理を行います。ソフトウエアスタンバイモードからの復帰は、HID デバイス(RSK)を USB Host にアタッチすることにより行われます。

(2). HID デバイス(RSK)を USB Host に接続した状態で、USB Host から送信されるサスペンド信号を受信すると APL は、MCU をソフトウエアスタンバイモードに遷移するための処理を行います。なお、ソフトウエアスタンバイモードからの復帰は、USB Host から送信されるレジューム信号の受信により行われます。

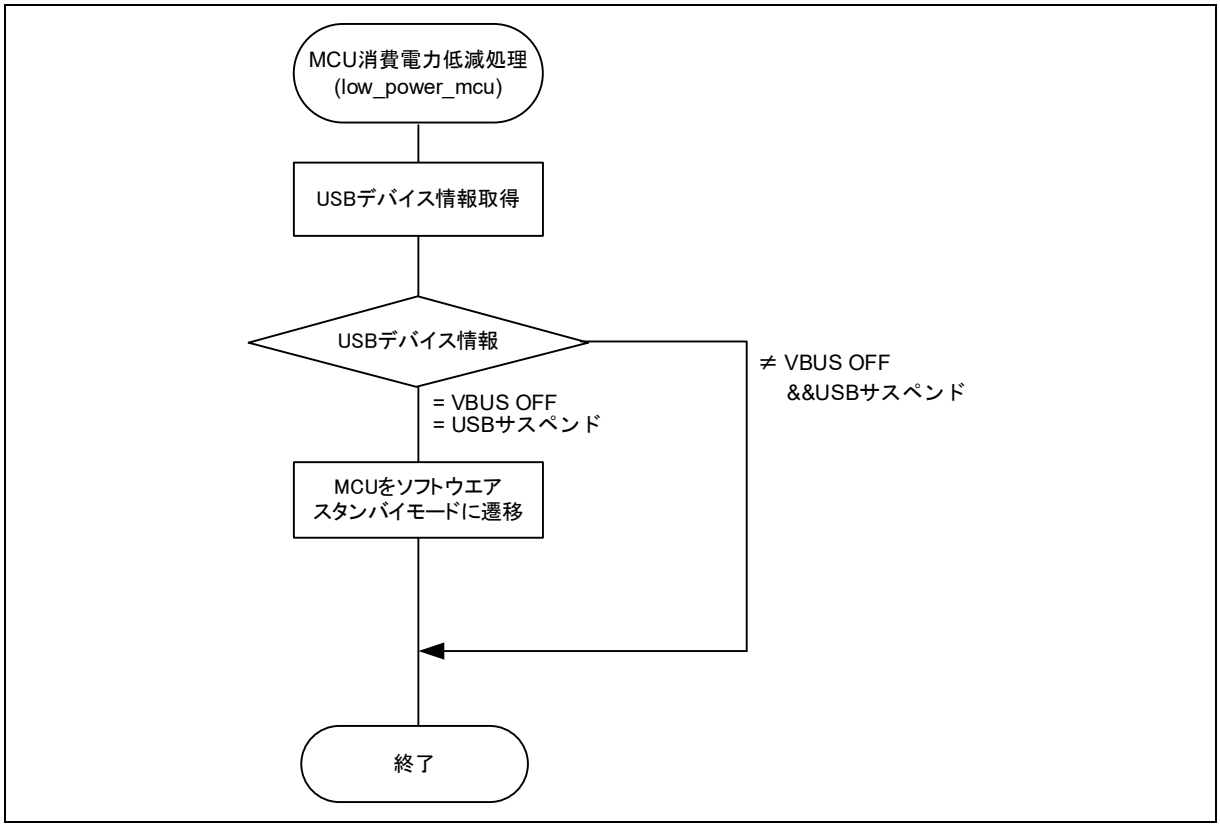


Figure 4-6 消費電力低減処理フロー

4.5 アプリケーションプログラム用コンフィグレーションファイル (r_usb_phid_apl_config.h)

1. OPERATION_MODE 定義

OPERATION_MODE 定義に対し、以下のいずれかを指定してください。

```
#define    OPERATION_MODE    HID_KEYBOARD    //Keyboard モード
#define    OPERATION_MODE    HID_ECHO        //Echo モード
```

2. 消費電力低減機能定義

消費電力低減機能の使用/非使用を指定してください。消費電力低減機能を使用する場合は、USB_SUPPORT_LPW 定義に対し USB_APL_ENABLE を指定し、消費電力低減機能を使用しない場合は、USB_SUPPORT_LPW 定義に対し USB_APL_DISABLE を指定してください。

```
#define    USB_SUPPORT_LPW    USB_APL_DISABLE    //消費電力低減機能を非使用
#define    USB_SUPPORT_LPW    USB_APL_ENABLE     //消費電力低減機能を使用
```

3. USB_SUPPORT_RTOS 定義

リアルタイム OS を使用するかどうかを指定します。リアルタイム OS を使用する場合は、USB_SUPPORT_RTOS 定義に対し USB_APL_ENABLE を指定してください。

```
#define    USB_SUPPORT_RTOS    USB_APL_DISABLE    // RTOS 非使用
#define    USB_SUPPORT_RTOS    USB_APL_ENABLE     // RTOS 使用
```

4. 注意事項

上記はアプリケーションプログラム用のコンフィグレーション設定です。上記の設定の他に USB ドライバのコンフィグレーション設定が必要です。USB ドライバのコンフィグレーション設定については、「USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) Firmware Integration Technology アプリーションノート」(Document No. R01AN2166)を参照してください。

4.6 キーボード動作

Keyboard モードでは、RSK/RSSK/EK を HID デバイスとして動作させるため RSK/RSSK/EK 上のスイッチを使用します。スイッチ入力情報がキーボードのキーデータとして使用されます。

4.6.1 スイッチ仕様

キーボードで使用するスイッチ仕様を以下に示します。なお、このスイッチ仕様は、スイッチが押下されただけではそのスイッチの押下は認識されず、スイッチが押下→リリースされることにより、スイッチの押下が認識されます。

1. RSK の場合

スイッチ番号	動作説明
スイッチ 1(SW1)	USB Host に対し Remote Wakeup 信号を送信します。
スイッチ 2(SW2)	スイッチを押すたびに“a”–“z”、“Enter”のキーコードを 1 つ通知します。
スイッチ 3(SW3)	スイッチを押すたびに“1”–“9”、"0"、“Enter”のキーコードを 1 つ通知します。

2. RSSK の場合

スイッチ番号	動作説明
スイッチ 1(SW1)	スイッチを押すたびに“a”–“z”、“Enter”のキーコードを 1 つ通知します。
スイッチ 2(SW2)	USB Host に対し Remote Wakeup 信号を送信します。

3. EK の場合

スイッチ番号	動作説明
スイッチ 1(SW1)	USB Host に対し Remote Wakeup 信号を送信します。
スイッチ 2(SW2)	スイッチを押すたびに“a”–“z”、“Enter”のキーコードを 1 つ通知します。

[Note]

1. SW(RSK/EK:SW1, RSSK:SW2)に割り当てられているポートを以下に示します。

	ポート	備考
RX111	P30	
RX113	PA4	消費電力低減機能使用
	PJ0	消費電力低減機能非使用
RX231	P31	
RX23W	P30	
RX261	PD0	

2. SW(RSK/EK:SW1, RSSK:SW2)を押下すると低消費電力状態が解除されます。本アプリケーションプログラムは、低消費電力状態を解除するために以下の IRQ 割り込みを使用しています。

	ポート	IRQ
RX111	P30	IRQ0
RX113	PA4	IRQ5
RX231	P31	IRQ1
RX23W	P30	IRQ0
RX261	PD0	IRQ0

3. RSKRX113 出荷時、SW1 は PJ0 に割り当てられていますので、RSK に対し以下の設定が必要です。
   (1). R112 に実装されているオプションリンク抵抗(0Ω)を外し、R111 へ実装してください。
   (2). R57 に実装されているオプションリンク抵抗(0Ω)を外し、R56 へ実装してください。

4.6.2 データフォーマット

USB ホストに転送するデータのフォーマットと USB から通知されるデータのフォーマットを下表に示します。これらのデータフォーマットは USB ホストに転送している HID のレポートディスクリプタの内容と併せて設定しています。

Table 4-3 ホストに通知するデータフォーマット

offset	キーボードモード (8Bytes)
0	Modifier keys
1	Reserved
2	Keycode 1
3	Keycode 2
4	Keycode 3
5	Keycode 4
6	Keycode 5
7	Keycode 6

Table 4-4　キーボード OUTPUT レポートフォーマット

offset (byte)	値
0	b0：LED 0 (NumLock) b1：LED 1 (CapsLock) b2：LED 2 (ScrollLock) b3：LED 3 (Compose) 0: 消灯、1: 点灯

5. ディスクリプタ

PHID のディスクリプタ情報は r_usb_phid_descriptor.c に記述しています。なお、Vendor ID は、必ずお客様用の Vendor ID をご使用いただきますようお願いします。

6. クラスリクエスト

PHID がサポートしているクラスリクエストをTable 6-1に示します。

Table 6-1　対応する基本リクエスト及び HID クラスリクエスト

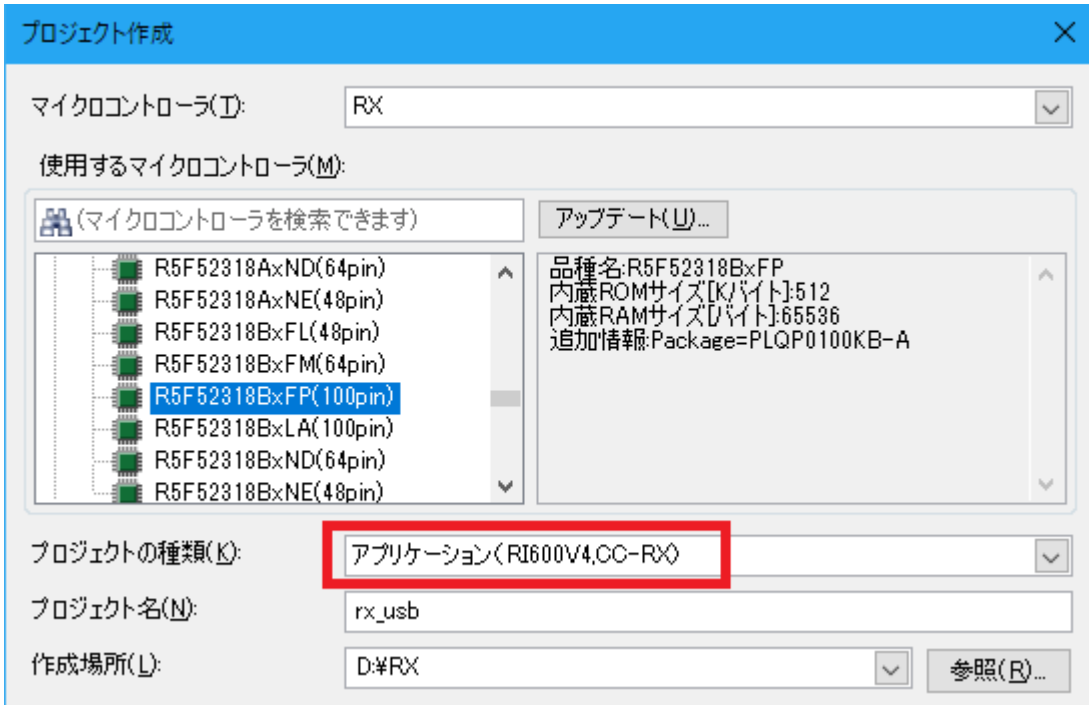
リクエスト	コード	説明	対応(※)
Get Report	0x01	USB Host にレポートを送信する	×
Set Report	0x09	USB Host からのレポートを受信する	×
Get Idle	0x02	USB Host に Duration 時間を送信する	×
Set Idle	0x0A	USB Host からの Duration 時間が受信する	×
Get Protocol	0x03	USB Host にプロトコルを送信する	×
Set Protocol	0x0B	USB Host からのプロトコルを受信する	×
Get Report Descriptor	Standard	USB Host にレポートディスクリプタを送信する	○
Get HID Descriptor	Standard	USB Host に HID ディスクリプタを送信する	○

7. RI600V4 プロジェクトを CS+で使用する場合

パッケージ内の RI600V4 用プロジェクトは CS+をサポートしていません。RI600V4 用プロジェクトを CS+で使用する場合、以下の手順に従って CS+用のプロジェクトを作成する必要があります。

7.1 CS+上で新規プロジェクトを作成

プロジェクトの種類には、「アプリケーション(RI600V4, CC-RX)」を選択してください。



7.2 スマートコンフィグレータを起動

1. クロック設定 (「クロック」タブを選択)

(1). USB クロック(UCLK)に 48MHz が設定されるよう関連クロックを設定してください。

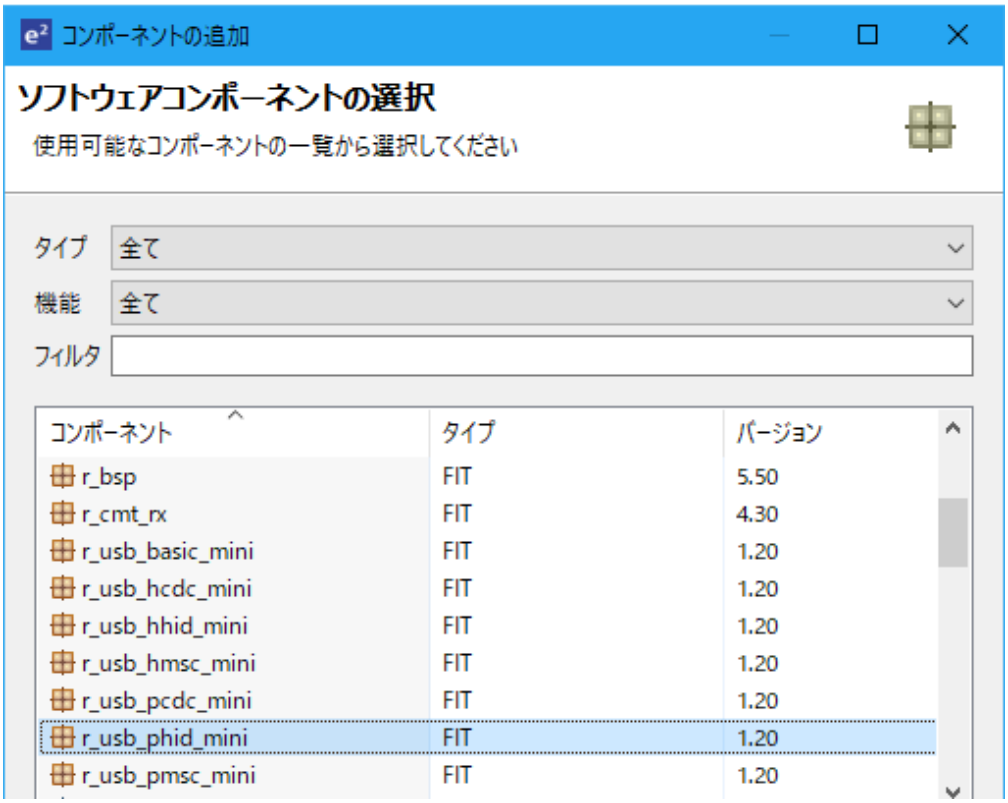
以下は、発振子(8MHz)を使用した場合の設定例です。

(2). メインクロックの発振安定時間(赤枠)を"61"に変更してください。

2. コンポーネント設定 (「コンポーネント」タブを選択)

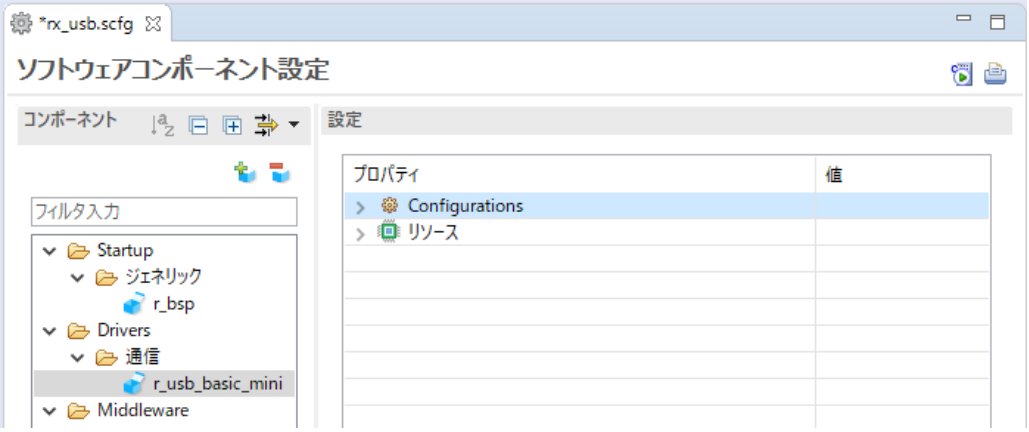
(1). USB FIT モジュールをインポート

r_usb_phid_mini モジュールを選択し、「終了」ボタンを押してください。r_usb_basic_mini モジュールも同時に組み込まれます。



(2). コンフィグレーション

a. r_usb_basic_mini

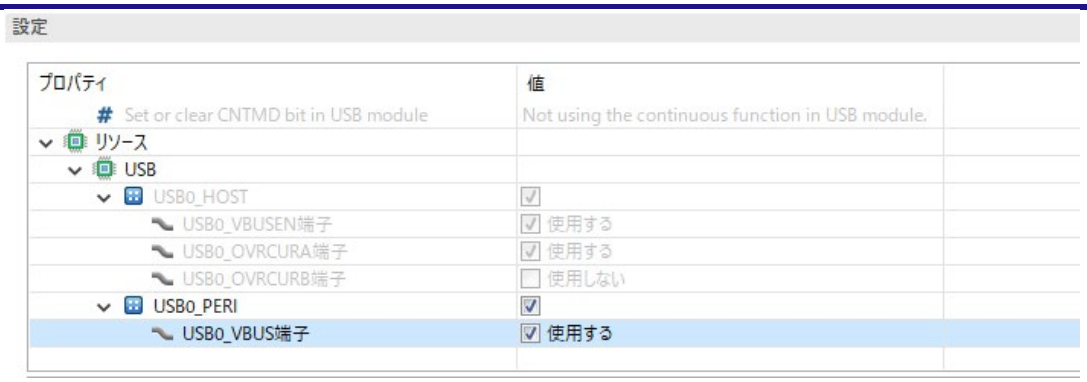


(a). Configurations

USB Basic Mini 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.R01AN2166)の「コンフィグレーション」章を参照いただきますようお願いします。

(b). リソース

USBx_VBUS 端子をチェックしてください。

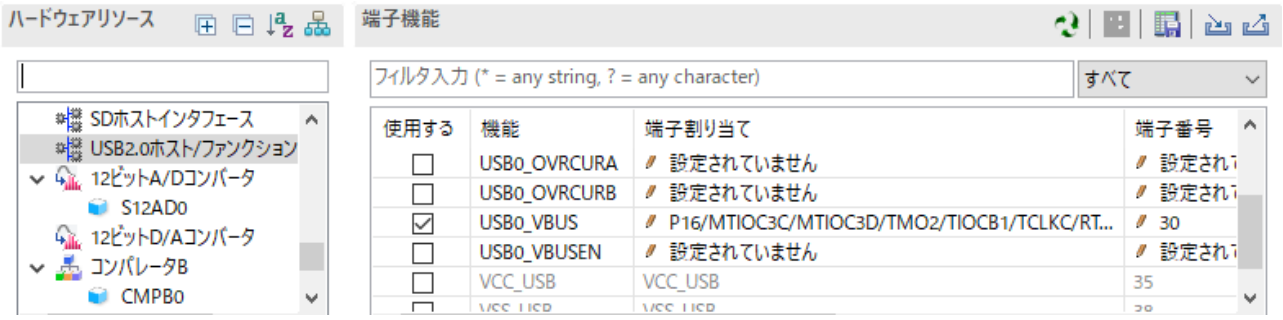


b.　r_usb_phid_mini

USB Peripheral Human Interface Devices Class Driver for USB Mini Firmware Firmware Integration Technologyアプリケーションノート(ドキュメントNo. R01AN2171)の「コンフィグレーション」章を参照いただきますようお願いします。

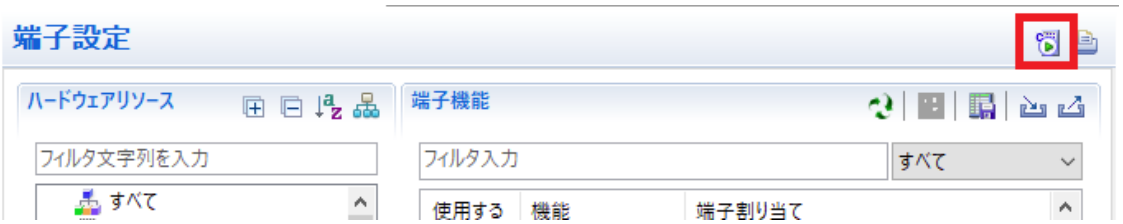
3.　端子設定 (「端子」タブを選択)

お客様のシステムに合った USB 端子のポート選択を行ってください。



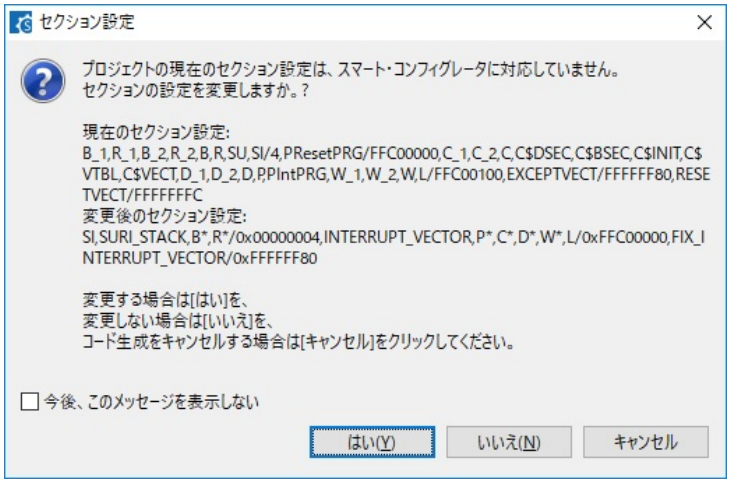
4.　コード生成

「コードの生成」ボタンをクリックすると、スマートコンフィグレータは<ProjectDir>¥src¥smc_gen フォルダに USB FIT モジュールのソースコードおよび端子設定のコードを生成します。



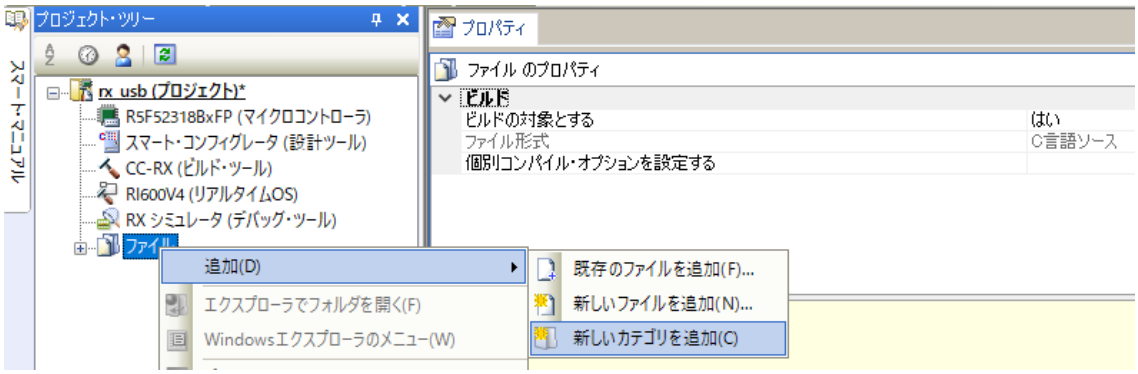
Note:

以下のダイアログが出力されますので、「はい(Y)」を選択してください。



7.3　アプリケーションプログラムおよびコンフィグレーションファイルの追加

1. 本パッケージ内の demo_src フォルダを<ProjectDir>¥src フォルダにコピーしてください。
2. 本パッケージ内の RI600V4 用コンフィグレーションファイル(.cfg ファイル)を<ProjectDir>フォルダにコピーしてください。
3. プロジェクトツリー内の「ファイル」を選択、右クリック。「追加」→「新しいカテゴリを追加」を選択し、アプリケーションプログラムを格納するカテゴリを作成してください。次に「既存のファイルを追加」を選択し、上記2でコピーしたアプリケーションプログラムおよびコンフィグレーションファイルを登録してください。



Note:

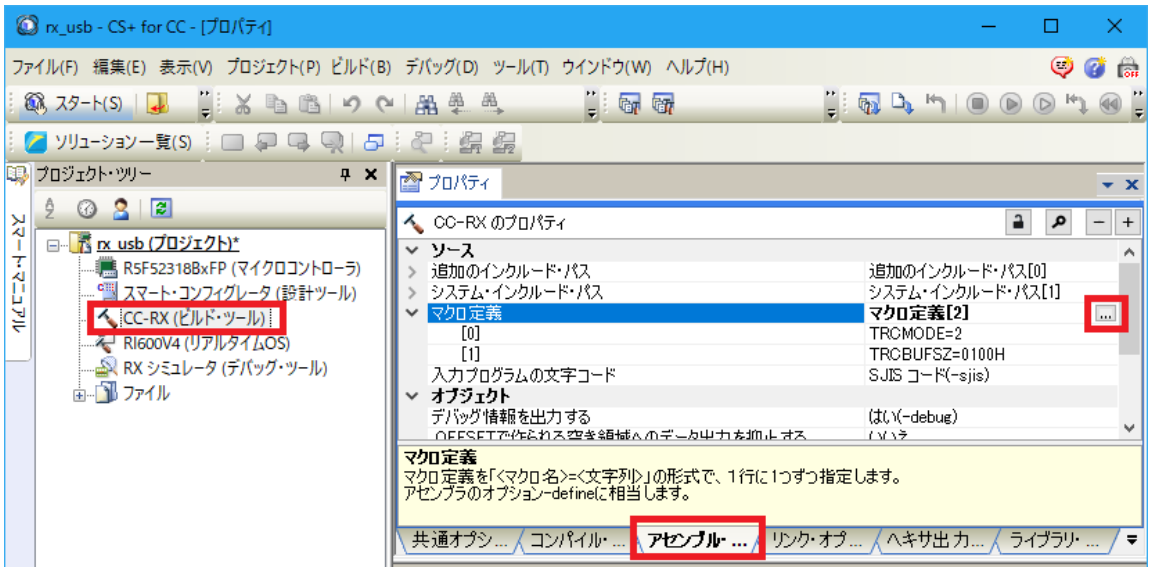
新規に生成された<ProjectDir>フォルダ内の task.c ファイルおよび sample.cfg ファイルを削除してください。

7.4　マクロ定義削除

新規生成したプロジェクトには以下のマクロが定義されていますので、これらのマクロ定義を削除してください。

CC-RX(ビルド・ツール) -> 「アセンブルオプション」タブを選択し、以下のマクロ定義を削除。

1. TRCMODE = 2
2. TRCBUFSZ = 0100H



7.5　ビルド実行

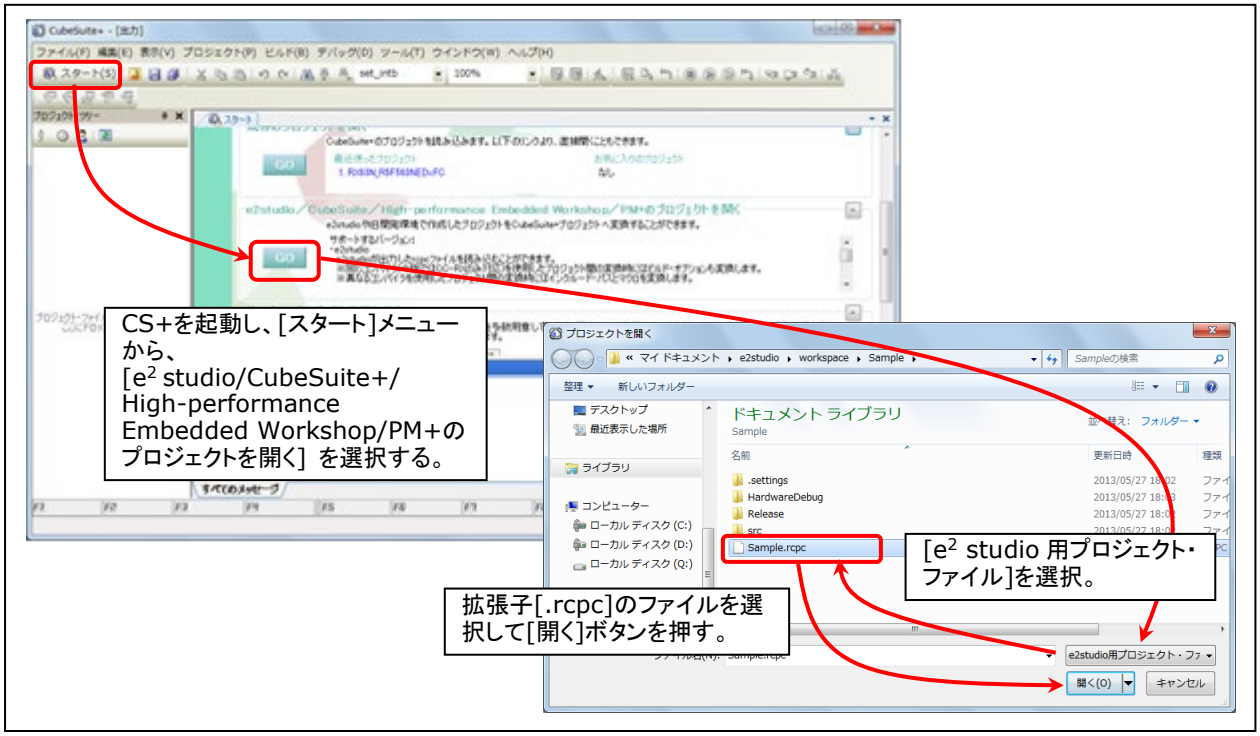
ビルドを実行し、実行プログラムを生成してください。

8. e² studio 用プロジェクトを CS+で使用する場合

PHID のプロジェクトは、統合開発環境 e² studio で作成されています。PHID を CS+で動作させる場合は、下記の手順にて読み込んでください。

[Note]

1. 「プロジェクト変換設定」ウィンドウ内の「変換直前のプロジェクト構成ファイルをまとめてバックアップする」のチェックを外してください。
2. RI600V4 をご使用の場合、以下の方法をサポートしていません。「**7. RI600V4プロジェクトをCS+で使用する場合**」を参照してください。



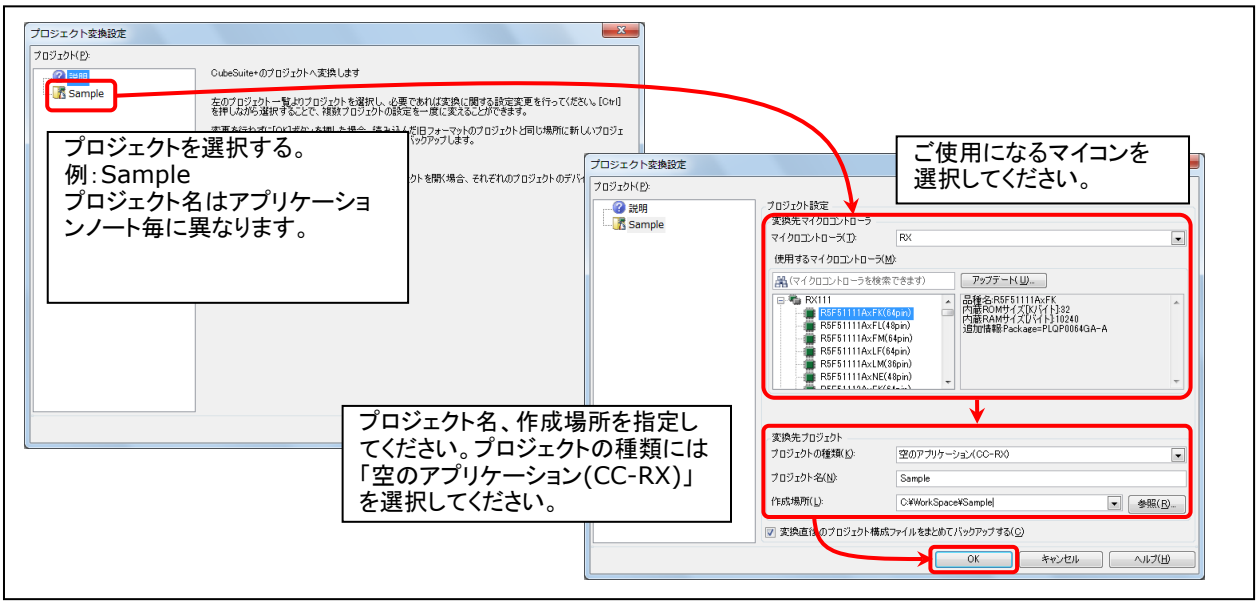


Figure 8-1　e² studio 用プロジェクトの CS+読み込み方法

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/

お問合せ先
http://japan.renesas.com/contact/

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	Dec 1, 2014	—	初版発行
1.01	Jun 1, 2015	—	RX231 を対象デバイスに追加
1.02	Dec 28, 2015	—	1. OUT 転送用 API を使った Echo モード用のサンプルアプリケーションプログラムを追加 2. Windows® 10 との動作確認を実施。
1.10	Nov 30, 2018	—	1. 以下の章を追加しました。 (1). 3.1.2 RSK/RSSK 設定 2. 以下の章を変更しました。 (1). 4. サンプルアプリケーション
1.12	Jun 30, 2019	—	RX23W を対象デバイスに追加
1.20	Jun 1, 2020	—	リアルタイム OS をサポートしました。
1.30	Jul 31, 2024	—	RX261 を対象デバイスに追加
1.32	May 15, 2026	—	spdx ファイルと json ファイルをパッケージに同梱

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。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

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。

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/

© 2026 Renesas Electronics Corporation. All rights reserved.