

RXファミリ

R01AN0401JJ0233

Rev.2.33

Jul 31, 2019

USB Peripheral Human Interface Device Class Driver

要旨

本アプリケーションノートは、USB Peripheral ヒューマンインタフェースクラスドライバ (PHID) について説明します。本ドライバは USB Basic Peripheral Driver(USB-BASIC-FW)と組み合わせることで動作します。以降、本ドライバを PHID と称します。

対象デバイス

RX62N/RX621 グループ

RX63N/RX631 グループ

RX630 グループ

RX63T 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. RX62N/RX621 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.R01UH0033)
3. 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.R01UH0041J)
4. RX630 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.R01UH0040)
5. RX63T 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.R01UH0238)
6. USB Basic Host and Peripheral Driver アプリケーションノート(ドキュメント No. R01AN0512)

— 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. ヒューマンインタフェースデバイスクラス (HID)	7
5. API.....	9
6. コンフィグレーション (r_usb_phid_config.h)	10
7. サンプルアプリケーション	11
8. セットアップ	18
9. アプリケーションの作成方法	21
10. e ² studio用プロジェクトをCS+で使用する場合	22

1. 概要

PHID は、と組み合わせることで、USB Peripheral ヒューマンインタフェースデバイスクラスドライバ（以降 PHID と記述）として動作します。PHID は、USB ヒューマンインタ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 Host とのデータ送受信
- ・ HID クラスリクエストに応答
- ・ USB Host からの機能照会に対する応答
- ・ Interrupt OUT 転送

1.1 必ずお読みください

1. 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** アプリケーションノート(ドキュメント No.R01AN0512)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。
2. RX62N/RX621/RX63T/RX630 をご使用の場合、必ず"reference_documents"フォルダ下のドキュメント(r01an0401jj0232_usb.pdf, r01an0512jj0232_usb.pdf)をご使用いただきますようお願いいたします。

1.2 注意事項

このドライバは、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。

API	: Application programing Interface
APL	: Application program
HID	: Human Interface Device class
PCD	: Peripheral control driver of
PDCD	: Peripheral device class driver (device driver and USB class driver)
PHID	: Peripheral Human Interface Devices
RSK	: Renesas Starter Kits
USB	: Universal Serial Bus
	: USB Basic Peripheral Driver

2. ソフトウェア構成

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。

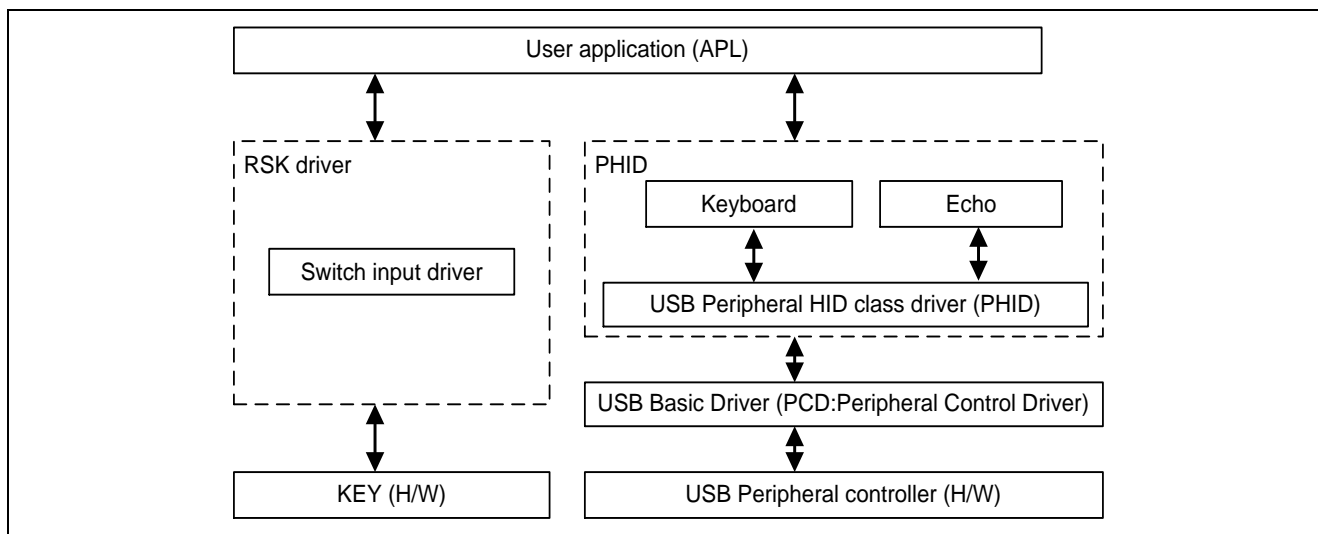


Figure 2-1 モジュール構成図

Table 2.1 各モジュール機能概要

モジュール名	機能概要
PHID	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	USB ペリフェラルコントロールドライバです。 (ハードウェア制御とデバイスステート管理を行います。)

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 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
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 V.3.01.00 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
モジュールのリビジョン	Rev.2.33
使用ボード	Renesas Starter Kits for RX63N
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

3.3 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	USBI0 割り込み(ベクタ番号: 35) / USBR0 割り込み(ベクタ番号: 90)
RX631	USBI1 割り込み(ベクタ番号: 38) / USBR1 割り込み(ベクタ番号: 91)

3.4 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_phid_if.h` に記載されています。

3.5 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。

3.6 コンパイル時の設定

コンパイル時の設定については、本書の「6. コンフィグレーション (`r_usb_phid_config.h`)」と USB Basic Host and Peripheral Driver アプリケーションノート(ドキュメント No.R01AN0512)内の「コンフィグレーション」の章を参照してください。

3.7 ROM/RAM サイズ

本ドライバの ROM/RAM サイズを以下に示します。

	引数チェック実施時	引数チェック非実施時
ROM サイズ	17.5K バイト (Note 3)	17.0K バイト (Note 4)
RAM サイズ	9.0K バイト	9.0K バイト

[Note]

1. 上記のサイズには、USB Basic Driver のコードサイズが含まれています。
2. 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。
3. 「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。
4. 「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。

3.8 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 アプリケーションノート(ドキュメント No.R01AN0512)内の「構造体」の章を参照してください。

4. ヒューマンインタフェースデバイスクラス (HID)

4.1 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 アプリケーションノート(ドキュメント No.R01AN0512)内の「クラスリクエスト」の章を参照してください。

Table 4.1 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 へ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 へ Duration 時間を送信する。	×
Set_Idle	0x0A	USB Host からの Duration 時間設定を受信する。	×
Get_Protocol	0x03	USB Host へ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_Descriptor Descriptor Type : Class Class Descriptor Type : Report	0x06 (Standard)	レポートディスクリプタを送信する。	○
Get_Descriptor Descriptor Type : Class Class Descriptor Type : HID	0x06 (Standard)	HID ディスクリプタを送信する。	○

※詳細は“USB Device Class Definitions for Human Interface Device, Revision1.1”の7章を参照ください。

4.2 クラスリクエストのデータフォーマット

本ドライバがサポートしているクラスリクエストのデータフォーマットを以下に示します。

1. GetReport リクエストフォーマット

Table 4-1 GetReport Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_REPORT (0x01)	ReportType & ReportID	Interface	ReportLength	Report

2. SetReport リクエストフォーマット

Table 4-2 SetReport Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_REPORT (0x09)	ReportType & ReportID	Interface	ReportLength	Report

3. GetIdle リクエストフォーマット

Table 4-3 GetIdle Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_IDLE (0x02)	0(Zero) & ReportID	Interface	1(one)	Idle rate

4. SetIdle リクエストフォーマット

Table 4-4 SetIdle Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_IDLE (0x0A)	Duration & ReportID	Interface	0(zero)	Not applicable

5. GetProtocol リクエストフォーマット

Table 4-5 GetProtocol Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_PROTOCOL (0x03)	0 (zero)	Interface	1(one)	0 (Boot Protocol) / 1 (Report Protocol)

6. SetProtocol リクエストフォーマット

Table 4-6 SetProtocol Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_PROTOCOL (0x0B)	0 (Boot Protocol) / 1 (Report Protocol)	Interface	0(zero)	Not applicable

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 アプリケーションノート(ドキュメント No.R01AN0512)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_phid_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

Interrupt IN, Interrupt OUT 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。なお、USB_CFG_PHID_INT_IN と USB_CFG_PHID_INT_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_PHID_INT_IN	パイプ番号 (USB_PIPE6 から USB_PIPE9)
#define	USB_CFG_PHID_INT_OUT	パイプ番号 (USB_PIPE6 から USB_PIPE9)

[Note]

OUT 転送をサポートしないシステムの場合は、USB_CFG_PHID_INT_OUT 定義に対し USB_NULL を設定してください。

7. サンプルアプリケーション

7.1 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 を USB Host に接続すると、USB Host は RSK をキーボードとして認識します。RSK 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 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。
 - a) 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。
 - b) USB デタッチ (切断) 状態時に、MCU をソフトウェ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 および Mouse mode の場合は、Windows 7/Windows 8.1/Windows 10 等の OS をサポートしている PC(USB Host)との USB 通信が可能です。

7.2 アプリケーション処理概要

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。

7.2.1 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。

7.2.2 メインループ (Keyboard mode)

Keyboard mode では、RSK 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(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、「7.4.1 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、HID クラスリクエストを受信後、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_REQUEST がセットされます。APL では、USB_STS_REQUEST_COMPLETE 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_REQUEST_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 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(HID_NONE)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)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、「7.2.4 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。

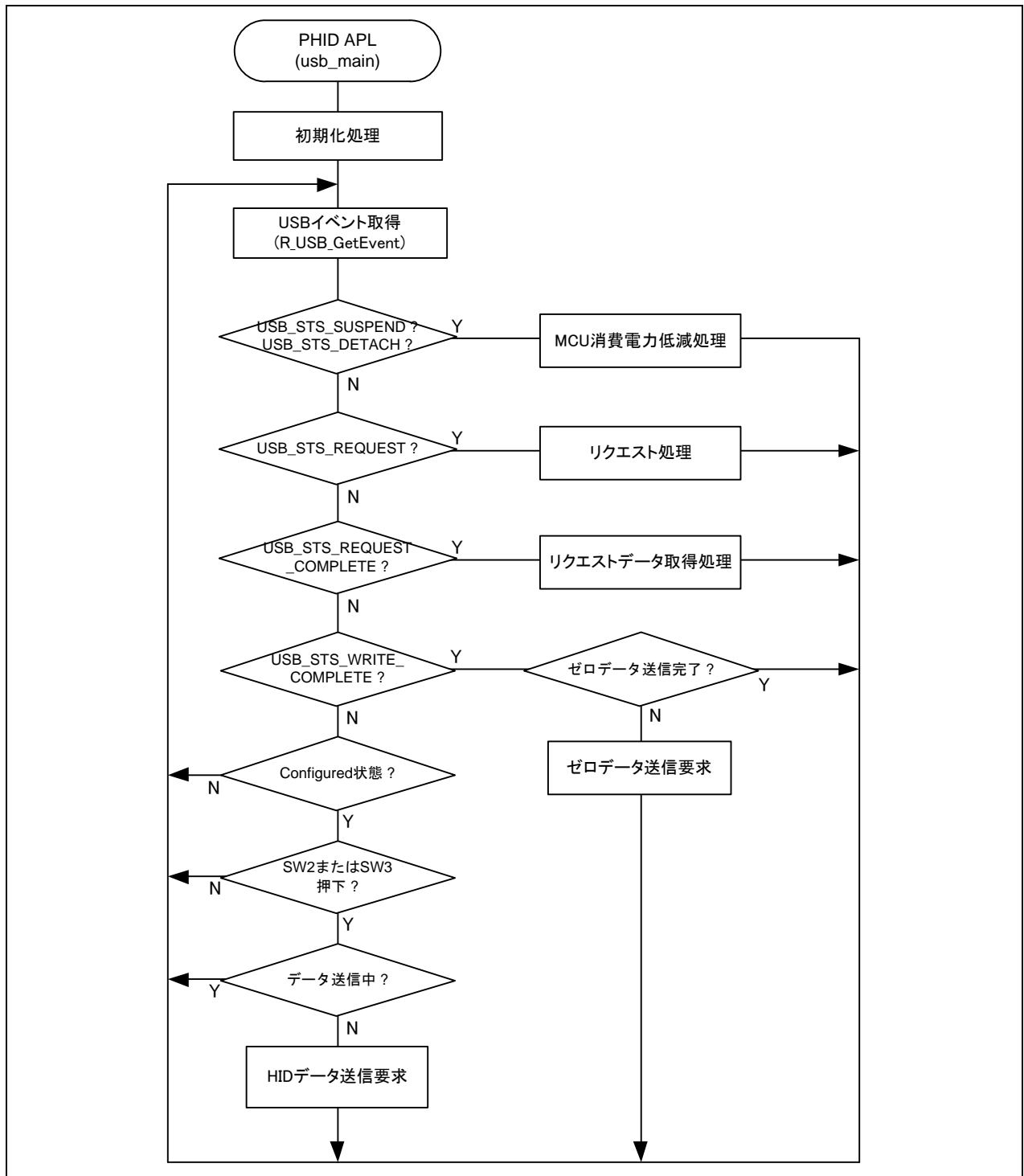


Figure 7-1 メインループ処理 (Keyboard mode)

7.2.3 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)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、「7.2.4 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。

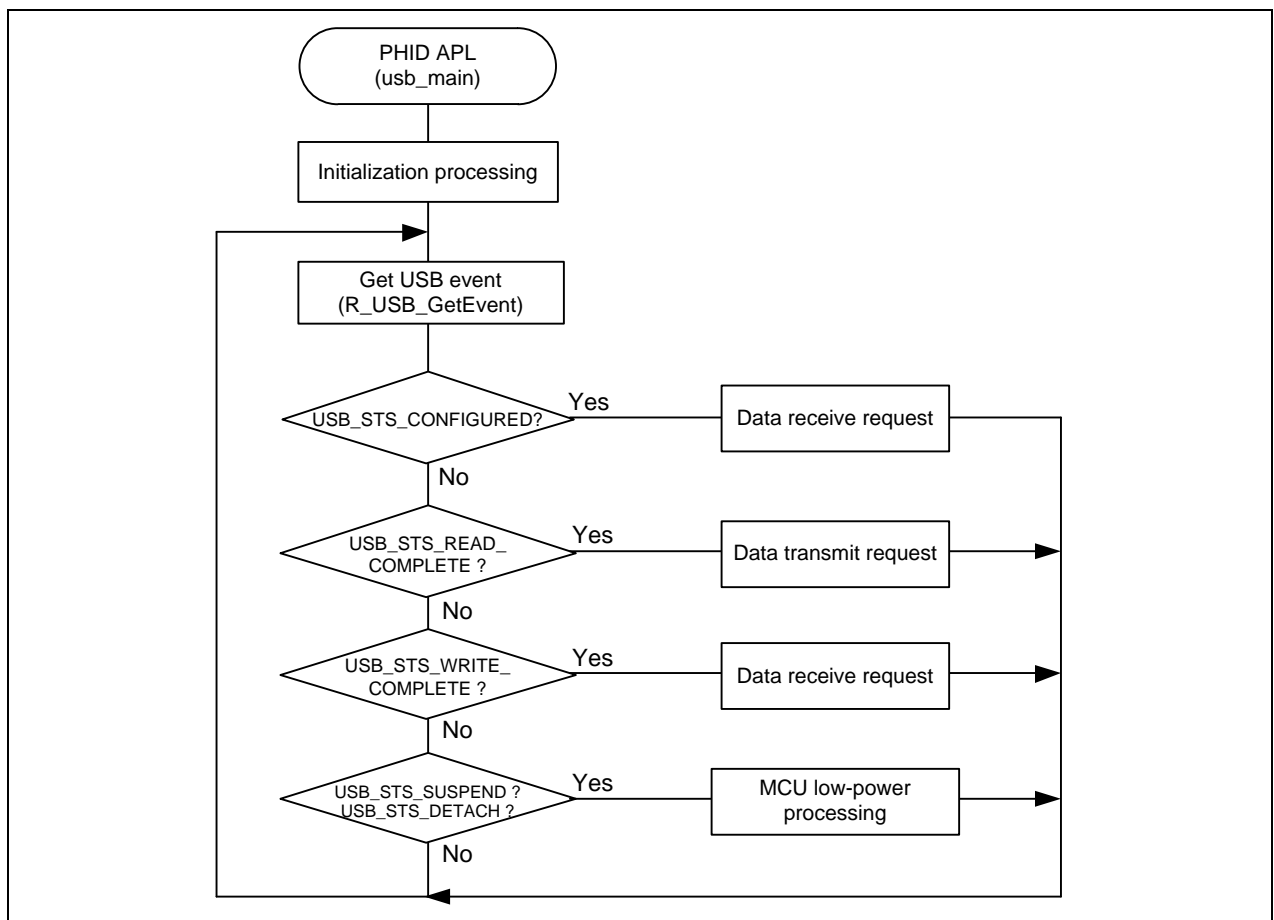


Figure 7-2 メインループ処理 (Echo mode)

7.2.4 MCU 消費電力低減処理

MCU 消費電力低減処理は、Table 7-1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”ファイル内に“USE_LPW”定義を定義してください。

Table 7-1 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	—	ソフトウェ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）

- HID デバイス(RSK)が USB Host からデタッチ（VBUS OFF）されると、APL は MCU をソフトウェアスタンバイモードに遷移するための処理を行います。ソフトウェ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。
- 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。

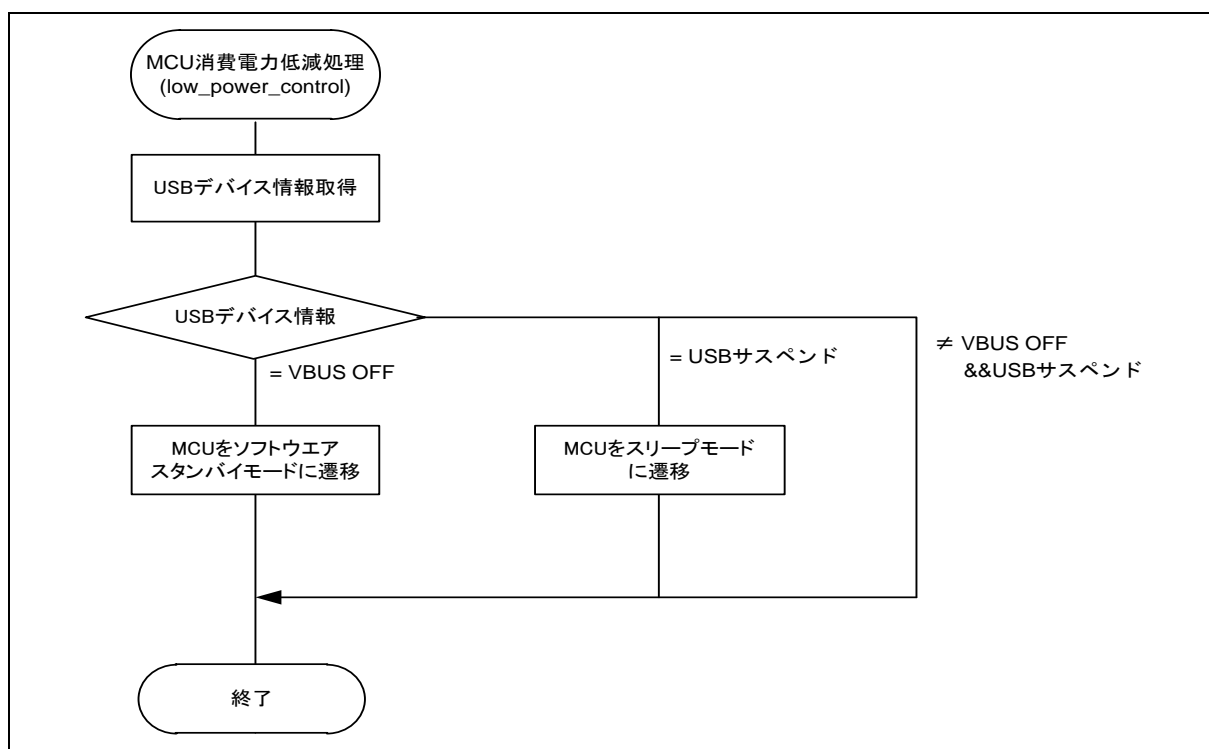


Figure 7-3 消費電力低減処理フロー

7.3 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. USE_USBIP 定義

使用する USB モジュールのモジュール番号を指定してください。USB_IP0/USB_IP1 のいずれかを指定してください。

```
#define USE_USBIP USB_USBIP0 // USB0 モジュールを使用する場合
#define USE_USBIP USB_USBIP1 // USB1 モジュールを使用する場合
```

[Note]

RX63T または RX630 をご使用の場合、USE_USBIP0 を指定してください。

2. 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
```

3. 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
```

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 Host and Peripheral Driver アプリケーションノート」(Document No. R01AN0512JJ)を参照してください。

7.4 キーボード動作

Keyboard モードでは、RSK を HID デバイスとして動作させるため RSK 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。

7.4.1 スイッチ仕様

マウスおよび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。

スイッチ番号	動作説明
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。

7.4.2 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 7-2 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

7.5 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いたします。

8. セットアップ

8.1 ハードウェア

8.1.1 動作環境例

PHID の動作環境例をFigure 8-1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。

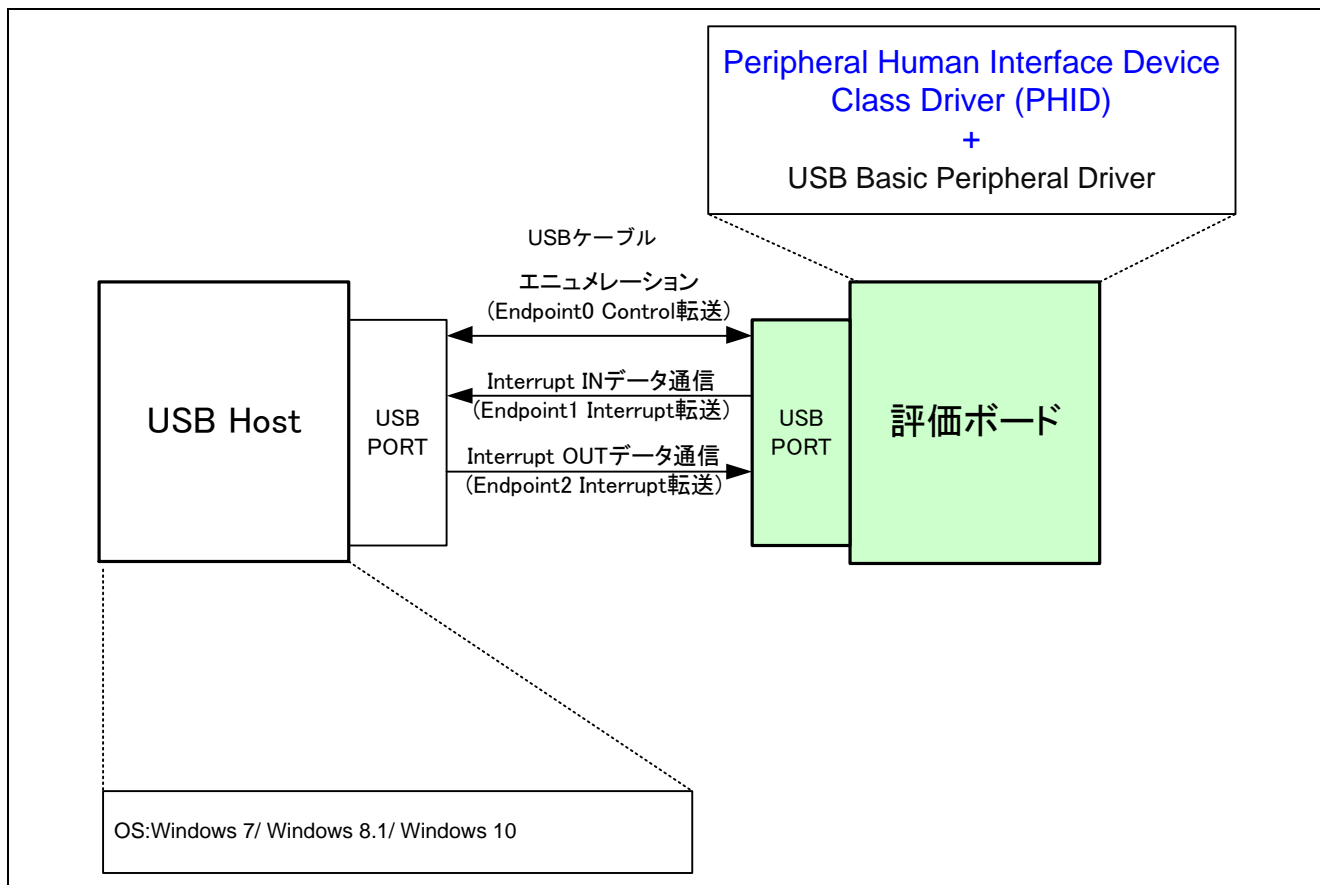


Figure 8-1 動作環境例

8.1.2 RSK 設定

RSK を USB Host 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 8-1 RSK 設定

RSK	ジャンパ設定
RSK+RX63N	J3: Shorted Pin1-2 J4: Shorted Pin1-2 J18: Shorted Pin2-3

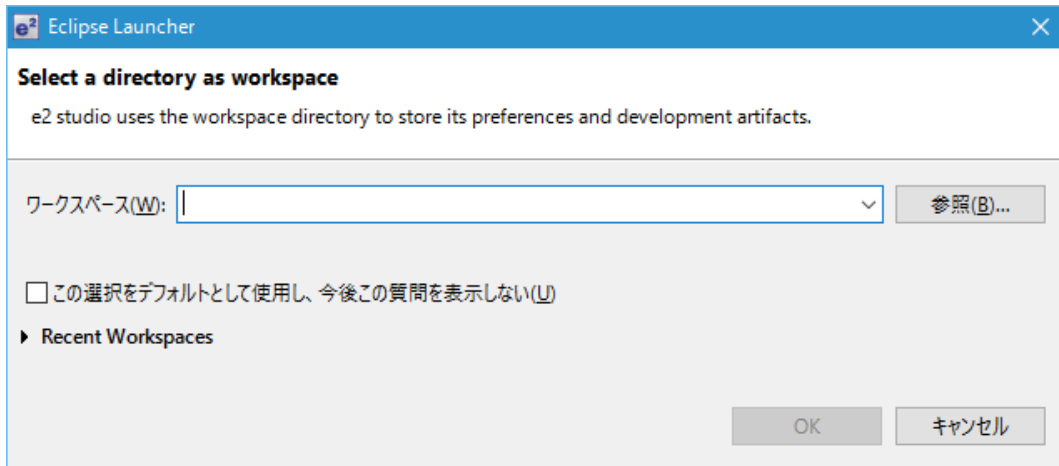
[Note]

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

8.2 ソフトウェア

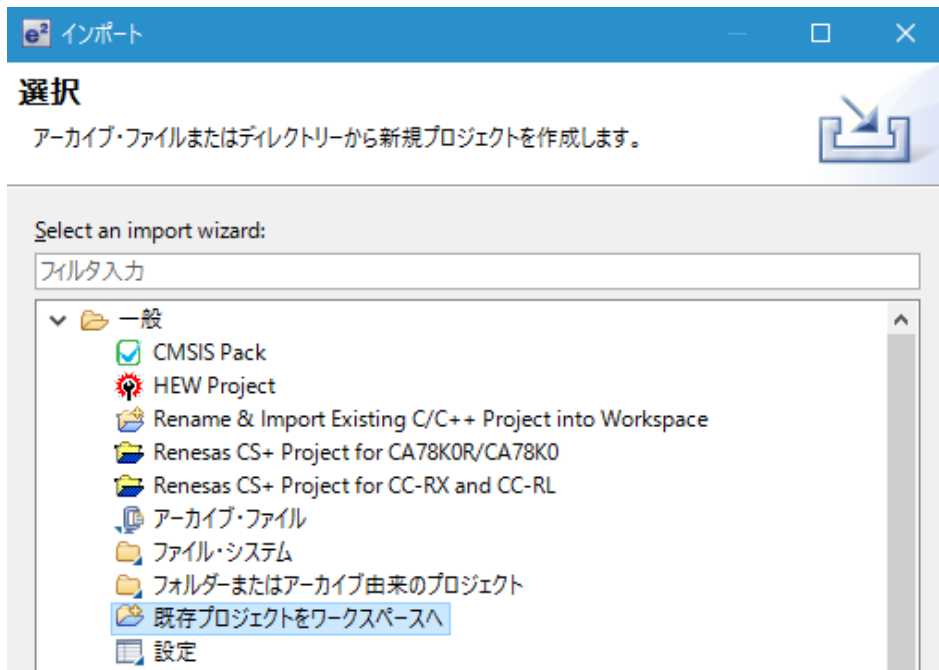
(1). e² studio を起動

- 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。

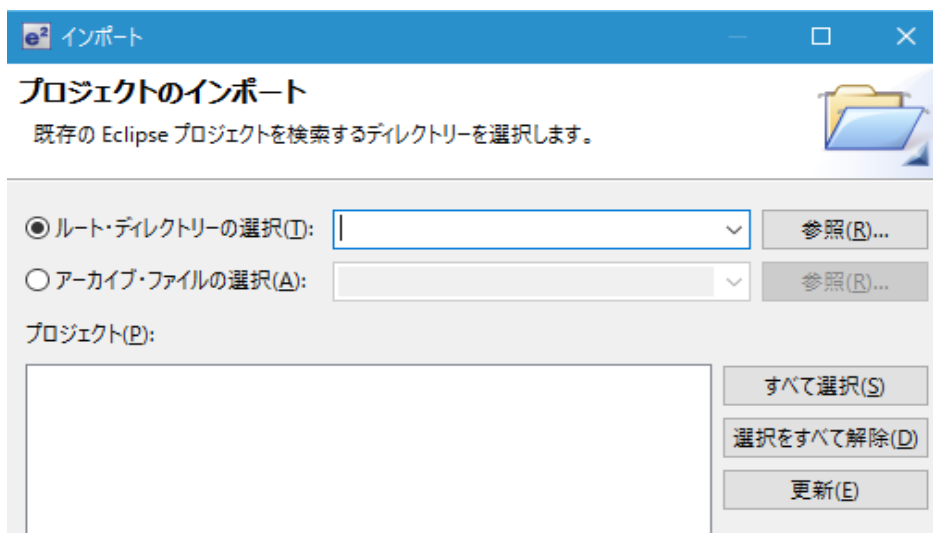


(2). ワークスペースへのプロジェクトの登録

- 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。



- (3). プロジェクトファイル".cproject"が格納されたフォルダを"ルート・ディレクトリの選択"に入力してください。



- c) “終了”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。

- (4). “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。
- (5). 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。

9. アプリケーションの作成方法

USB Basic Host and Peripheral Driver アプリケーションノート(ドキュメント No.R01AN0512)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。

10. 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]

「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。

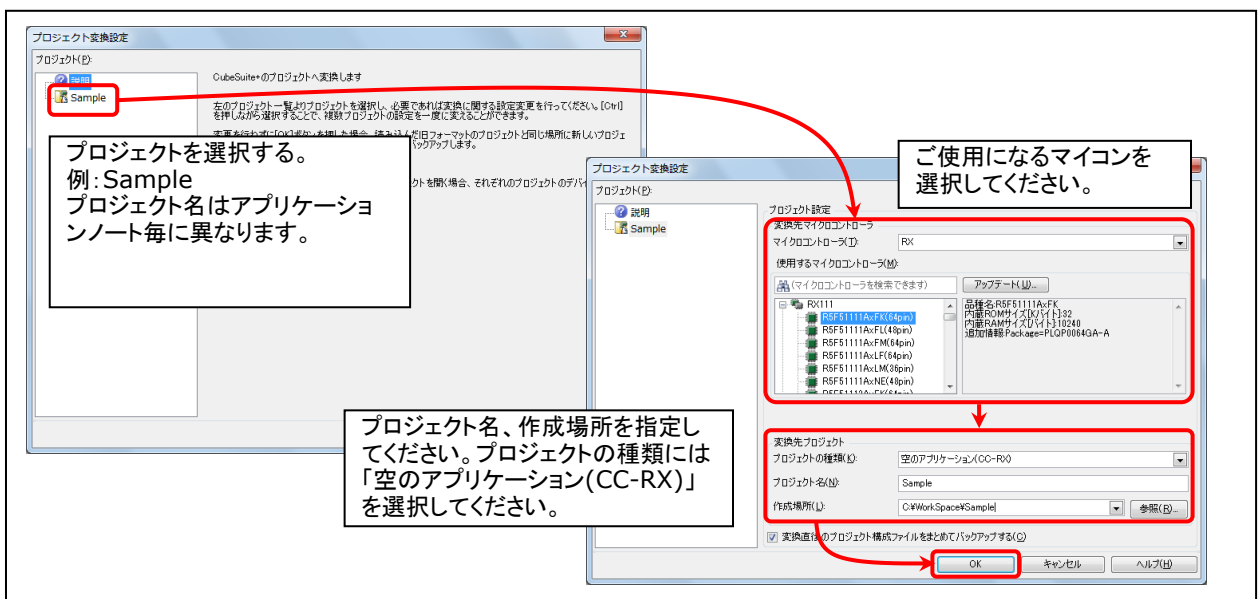
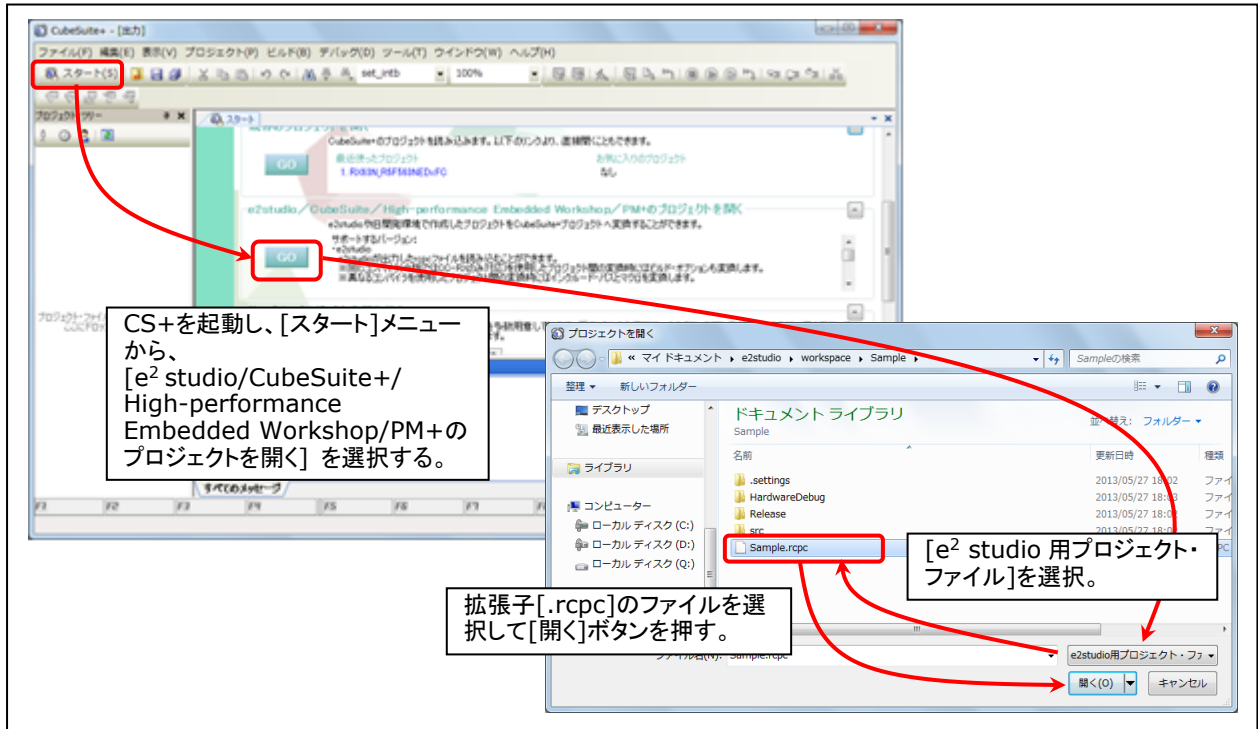


Figure 10-1 e² studio 用プロジェクトの CS+読み込み方法

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	2011.04.11	—	初版発行
1.10	2011.08.10	—	動作確認デバイス RX630、R8A66597 の追加 上記追加に伴い、RX630 に関する内容、R8A66597(Hi-Speed) に関する内容を追加
		21	5.ペリフェラル用 HID サンプルアプリケーション(APL)のアプリケーション使用変更 ・ SW1 使用禁止 ・ SW 押下表記を機能スイッチ表記に変更
2.00	2012.09.30	—	ファームウェアアップデートによるドキュメントの改訂
2.01	2012.2.1	—	"1.5 本書の読み方"追加、"ファイル一覧"の変更
2.10	2012.4.1	—	V.2.10 用 First Release 動作確認デバイスに RX63T, R8A66593 を追加。この追加に伴いこれらのデバイスに関する内容を追加
2.20	2015.9.30	—	アプリケーションプログラムを変更 フォルダ構成を変更 対象デバイスから RX63N, RX631, R8A66597 および R8A66593 を削除
2.30	2016.9.30	—	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
2.31	2017.9.30	—	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)の記載内容を本ドキュメントに移行し、USB Host and Peripheral Interface Driver アプリケーションノートを削除した。
2.32	2018.5.31	—	USB Basic Driver のリビジョンを Up しました。
2.33	Jul 31, 2019	—	対象デバイスに RX63N/RX631 を追加

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っていると、内部ROM、レイアウトパターンなどの相違などにより、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。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（豊洲フォレストシア）

■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/>