

Renesas USB MCU

R01AN2238JJ0121

Rev 1.21

Mar 31, 2017

USB Peripheral Communications Devices Class Driver (PCDC) Using Firmware Integration Technology Modules

要旨

本資料は、USB Peripheral Communications Devices Class Driver 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 PCDC と記述します。

実際のソフトウェア開発時には、必ず”USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート”(Document No:R01AN2025JJ)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 Communication Device Class Driver (PCDC) Firmware Integration Technology アプリケーションノート”(Document No:R01AN2030JJ)も参照してください。なお、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート”(Document No:R01AN2025JJ)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

RX63N/RX631 Group

RX65N/RX651 Group

RX64M Group

RX71M Group

本プログラムは Renesas Starter Kit (RSK) 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.....	7
3. サンプルアプリケーション	8
4. クラスドライバ概要	15
5. e ² studio 用プロジェクトを CS+で使用する場合	16

1. はじめに

1.1 機能概要

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 PC と通信を行うことが可能です。

PCDC の機能を以下に示します。

- ・ USB-シリアル変換機能及び、USB ループバック通信機能（エコーモード）を実装しています。
- ・ USB ホストと接続時、コミュニケーションクラス（仮想 COM）として認識される。
- ・ ターミナルソフトで仮想 COM ポートを指定することで、通信を行うことが可能。

1.2 FIT モジュール構成

PCDC 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	バージョン
RX Family Board Support Package Module Using Firmware Integration Technology	r_bsp	3.40
Renesas USB MCU USB Basic Host and Peripheral firmware Firmware Integration Technology	r_usb_basic	1.21
Renesas USB MCU USB Peripheral Communications Devices Class Driver(CDC) Firmware Integration Technology	r_usb_pcdc	1.21
RX Family DTC Module Firmware Integration Technology (RX64M and RX71M)	r_dtc_rx	2.05
RX Family DMA Controller DMACA Control Module Firmware Integration Technology (RX64M, RX71M and RX65N)	r_dmaca_rx	1.04
RX Family BYTEQ Module Using Firmware Integration Technology	r_byteq	1.50
RX Family SCI Multi-Mode Module Using Firmware Integration Technology	r_sci_rx	1.80

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 動作確認環境

PCDC の動作確認環境を以下に示します。

<評価ボード>

Renesas Starter Kit+ for RX63N (RSK+RX63N)

RX63N グループ ルネサスマイコン開発スタータキット ルネサスエレクトロニクス製

Renesas Starter Kit+ for RX65N (RSK+RX65N)

RX65N グループ ルネサスマイコン開発スタータキット ルネサスエレクトロニクス製

Renesas Starter Kit+ for RX64M (RSK+RX64M)

RX64M グループ ルネサスマイコン開発スタータキット ルネサスエレクトロニクス製

Renesas Starter Kit+ for RX71M (RSK+RX71M)

RX71M グループ ルネサスマイコン開発スタータキット ルネサスエレクトロニクス製

<開発環境>

統合環境 e² studio ルネサスエレクトロニクス製

RX ファミリー用 C/C++コンパイラパッケージ CC-RX V.2.05 ルネサスエレクトロニクス製

E1 エミュレータまたは E20 エミュレータ ルネサスエレクトロニクス製

<その他>

CDC ホスト(PC: Windows® 7、Windows® 8.1、Windows® 10)

USB ケーブル

ユーザケーブル (E1 エミュレータまたは E20 エミュレータに同梱)

エミュレータ用ケーブル (E1 エミュレータまたは E20 エミュレータに同梱)

1.5 セットアップ

1.5.1 ハードウェア

PCDC の動作環境例をFigure 1-1 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。

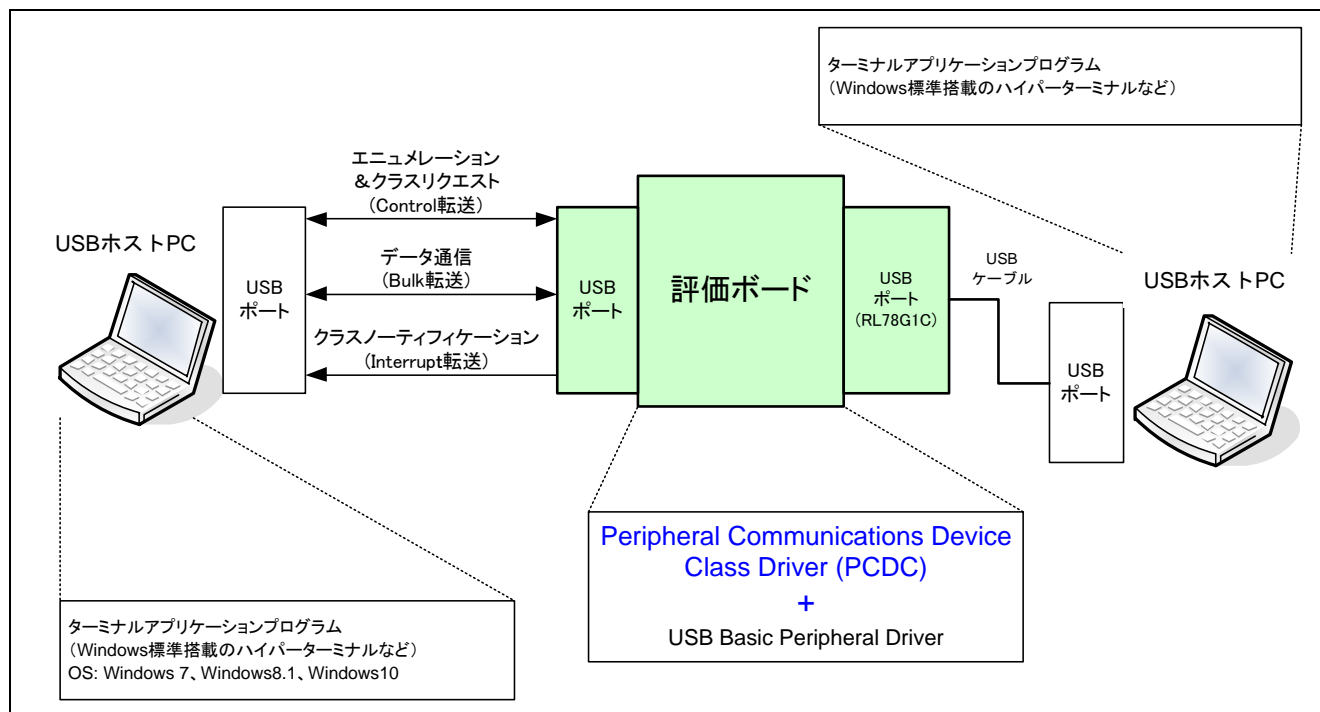


Figure 1-1 Example Operating Environment

動作確認済みの評価ボードをTable 1-2に示します。

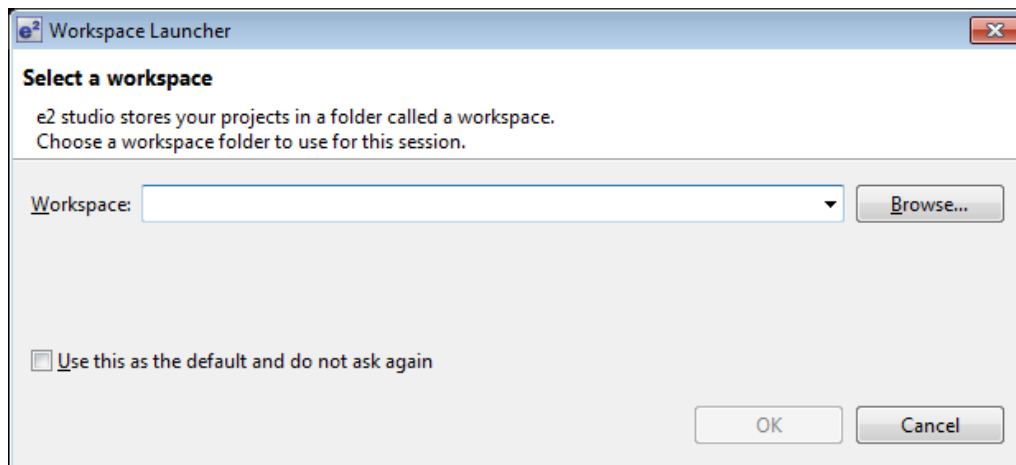
Table 1-2 PCDC 動作確認済みの評価ボード

MCU	評価ボード
RX63N	RSK+RX63N
RX65N	RSK+RX65N
RX64M	RSK+RX64M
RX71M	RSK+RX71M

1.5.2 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、Workspace 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。



- c) Workbench アイコンをクリックしてください。

Welcome to e²studio



Overview

Get an overview of the features



Renesas Tutorials

Go through Renesas Tutorials



Renesas Samples

Try out the Renesas Samples



What's New

Find out what is new



First Steps

Take your first steps

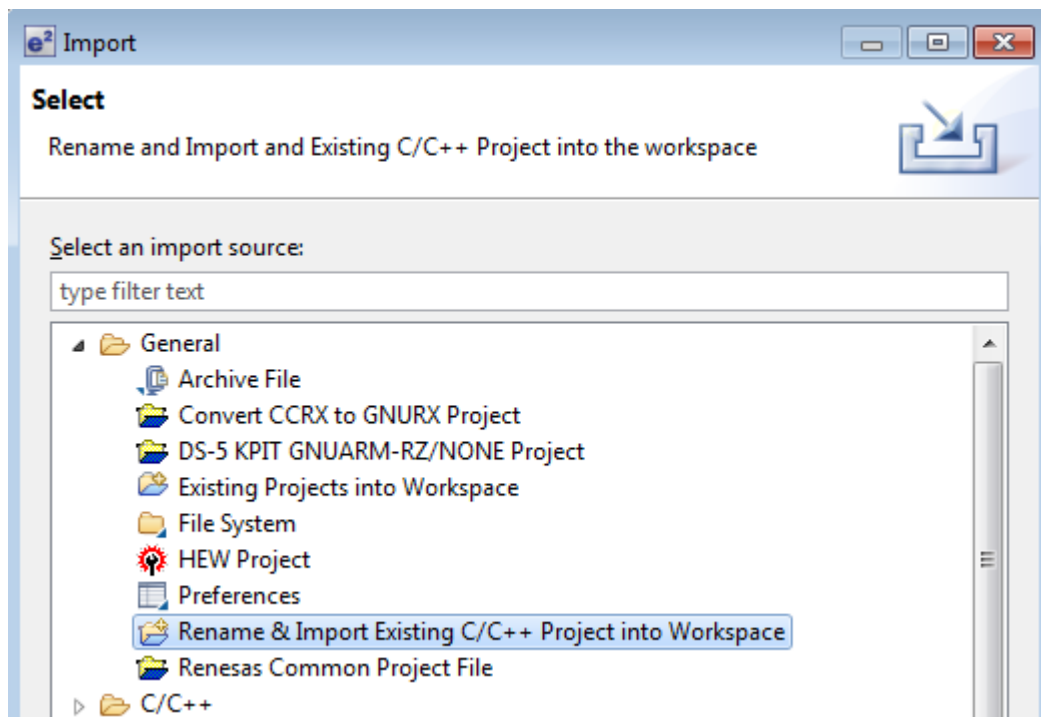


Workbench

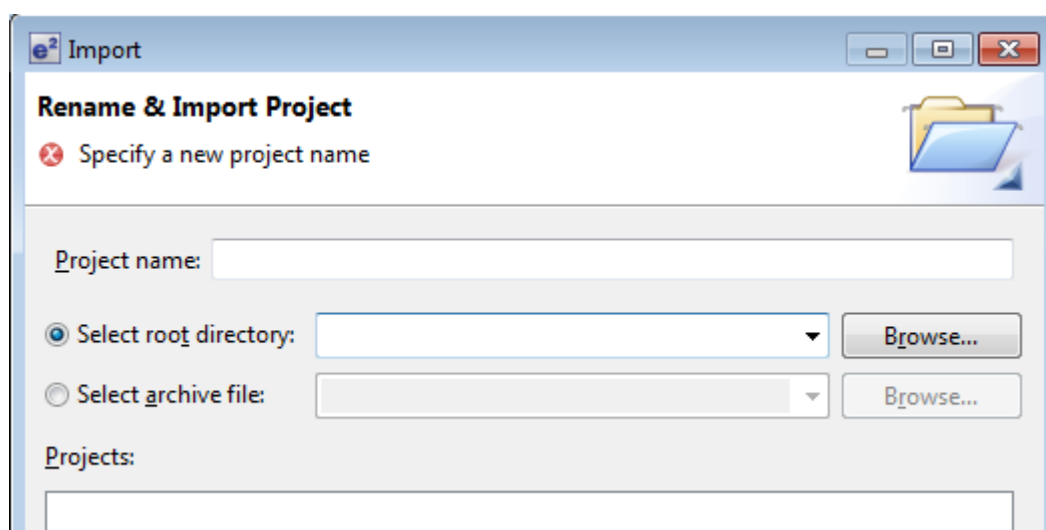
Go to the e2 studio workbench

(2). ワークスペースへのプロジェクトの登録

- a) [File] --> [Import]を選択してください。
- b) General => Rename & Import Existing C/C++ Project into Workspace を選択してください。



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。



- c) "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。

2. ソフトウェア構成

2.1 モジュール構成

PCDC は USB-シリアル変換機モード及び、USB ループバックモード（エコーモード）があり、USB-シリアル変換モードではシリアル・コミュニケーション・インタフェース（SCI）を使用します。LCD 表示や低消費電力制御処理はサンプルアプリケーションとして実装しています。

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。

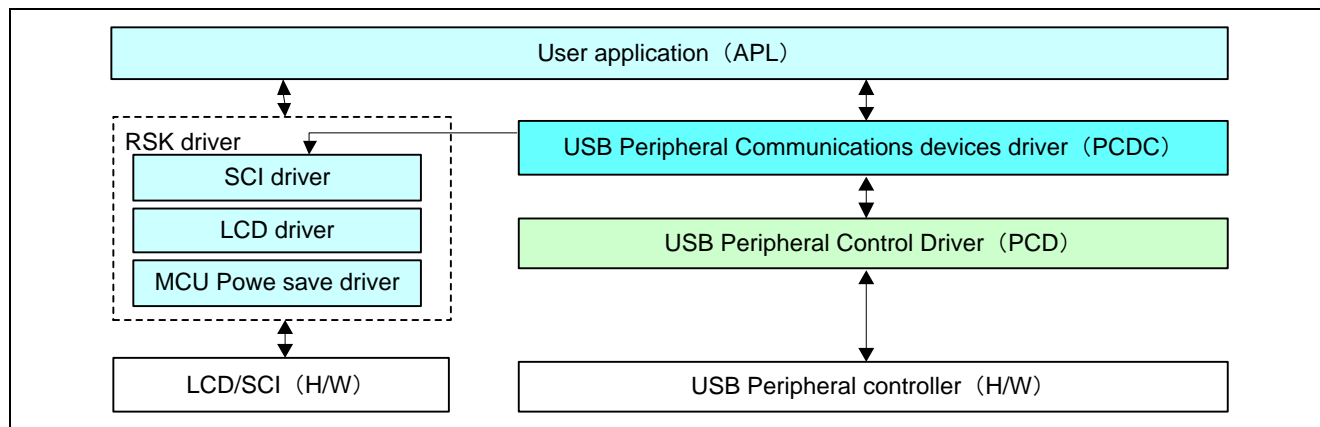


Figure 2-1 モジュール構成図

Table 2-1 モジュール機能概要

モジュール名	機能概要
APL	サンプルアプリケーションプログラム
RSK driver	RSK 上の各周辺機能を使用するためのサンプルアプリケーション
PCDC (r_usb_pcdc)	CDC クラスドライバ <ul style="list-style-type: none"> ・ USB ホストからの要求を解析する。 ・ PCD を介して APL と USB ホスト間のデータ転送サービスを提供する。
PCD (r_usb_basic)	USB Peripheral H/W 制御ドライバ

3. サンプルアプリケーション

3.1 アプリケーション仕様

PCDC のサンプルアプリケーション(以降、APL)は、RSK 上に実装された液晶表示器(LCD)を使用します。APL の主な機能を以下に示します。

1) Echo(ループバック)モード (注 1)

USB ホストから受信したデータを、USB ホストへ送信します。

2) USB-シリアル変換モード (注 1) (注 2) (注 3) (注 4)

USB ホストから受信したデータを COM ポート(SCI7)に送信し、COM ポート(SCI7)から受信したデータを USB ホストに送信します。なお、COM ポートのエラー (パリティエラー、フレーミングエラー、オーバーランエラー)が発生した場合、USB ホストにクラスノーティフィケーション“SerialState”を通知します。

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。

(注1) Echo モード/USB-シリアル変換モードの選択は、“r_usb_pcdc_apl_config.h”ファイル内で行ってください。

(注2) COM ポート(SCI7)は RSK+RX71M/RSK+RX64M/RSK+RX65N 上で RL78G1C と接続しており、G1CUSB0 を使用した USB-USB 通信となります。

(注3) USB-シリアル変換モードの場合、RSK に対し以下のジャンパ設定が必要です。

Table 3-1 USB-シリアル変換モードジャンパ設定

RX64M / RX71M		
ポート	信号	ジャンパ設定
P90	A-TXD7	J16.2-3
P92	A-RXD7	J19.2-3
RX63N		
ポート	信号	ジャンパ設定
P81	TXD0	J12.1-2
P75	RXD0	J13.1-2
RX65N		
ポート	信号	ジャンパ設定
P50	TXD2	--
P52	RXD2	--

(注4) USB-シリアル変換モードの場合、以下の FIT モジュールが必要です。

a) RX Family Serial Communication Interface Firmware Integration Technology

b) RX Family BYTEQ Module Firmware Integration Technology

3.2 アプリケーション処理概要

APL は、初期設定、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。

3.2.1 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。

3.2.2 メインループ (Echo モード)

Echo モードでは、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 に送信するためのデータ送信要求を行います。
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、「3.3 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。

以下に、APL の処理概要を示します。

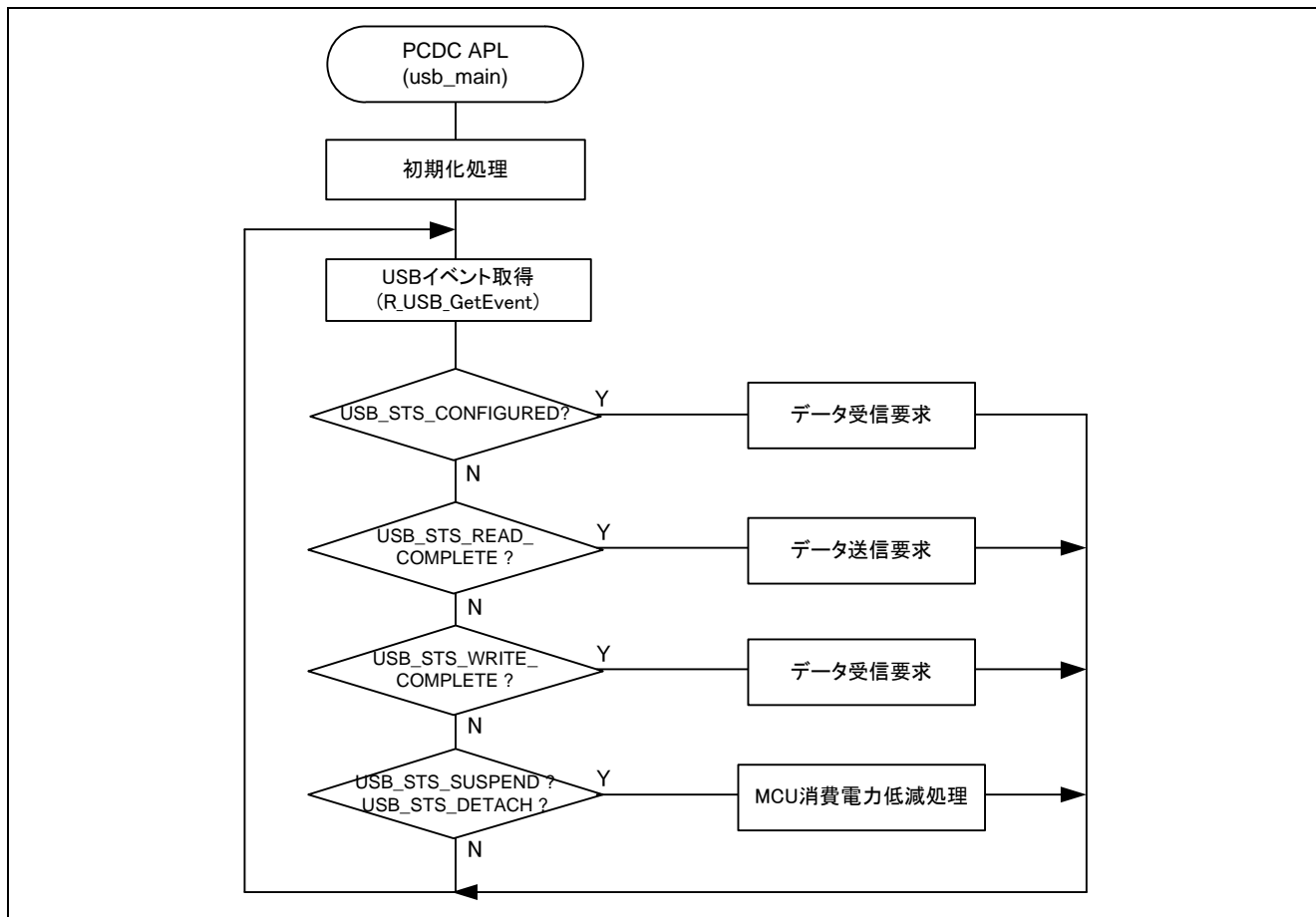


Figure 3-1 メインループ処理 (ループバックモード)

3.2.3 メインループ (USB-シリアル変換モード)

USB-シリアル変換バックモードは、以下の処理を行います。

- a. USB ホストからデータを受信し、その受信したデータの COM ポートへの送信。
- b. COM ポートから受信したデータの USB ホストへの送信。

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 Bulk データのデータ受信要求を行います。
2. USB Host との Enumeration が完了すると USB Host は、CDC クラスリクエストを CDC デバイスに送信します。CDC クラスリクエスト受信後、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 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。
3. 上記2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。
4. USB Host からの Bulk 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 を確認すると受信した Bulk データを COM ポートへ送信するため SCI 送信要求フラグをセットします。このフラグは、下記6の処理で参照されます。
5. USB Host への USB データ送信(下記6参照)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 usb_ctrl_t 構造体のメンバ type を参照し、データ送信が完了したデバイスクラス種別を確認します。送信が完了したデバイスクラス種別に応じ、該当の完了フラグをセットします。このフラグは、下記6の処理で参照されます。
6. USB イベントが発生していない状態で、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_NONE がセットされます。USB_STS_NONE の処理では、以下の送信処理を行います。なお、以下の送信処理を行う前に上記4と5でセットされたフラグを参照し、送信可能かどうかを判断しています。
 - (1). USB Host から受信した Bulk データを COM ポートへ送信する SCI 送信処理。
 - (2). SCI エラー(Parity error/Framing error/Overrun error など)を検出した場合、USB Host へ通知するための Class Notification (Serial State)送信要求処理
 - (3). COM ポートから受信したデータを USB Host へ送信するためデータ送信要求処理。
7. 上記4から6の処理が繰り返し行われている間に、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 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、「3.3 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。

以下に、APL の処理概要を示します。

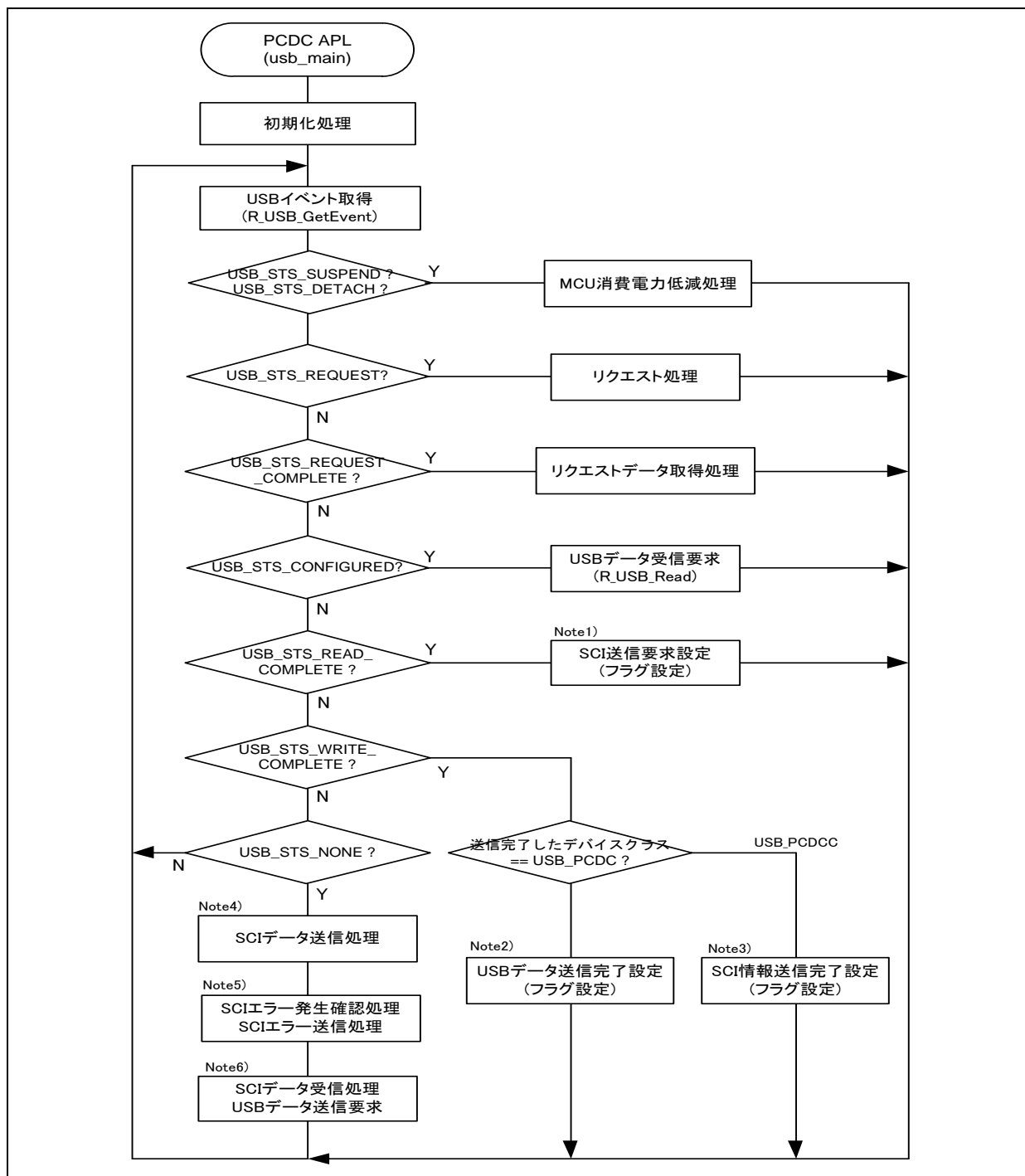


Figure 3-2 メインループ処理 (USB-シリアル変換モード)

- (Note1) SCI 送信要求フラグ設定を行います。このフラグは (Note4) の処理で参照されます。
- (Note2) USB データ送信完了のフラグ設定を行います。このフラグは (Note6) の処理で参照されます。
- (Note3) SCI 情報(SerialState)送信完了のフラグ設定を行います。このフラグは (Note5) の処理で参照されます。
- (Note4) (Note1) のフラグ状態を参照し、SCI データ送信処理を行います。SCI データ送信が正常終了した場合、SCI 送信要求フラグに対するクリア設定を行います。SCI データ送信に失敗した場合は、再度 SCI データ送信処理を行うため SCI 送信要求フラグのクリア設定を行いません。

- (Note5) SCI 状態にエラーが発生したかどうかの確認処理を行います。エラーが発生した場合、R_USB_Write 関数を使用し、USB Host に対しそのエラー情報を送信します。
- (Note6) (Note2) のフラグ状態を参照し、USB 送信が可能かどうかを確認します。送信可能であれば、USB_Write 関数を使用し、受信した SCI データを USB Host に送信します。なお、USB Host へのデータ送信が完了するまでは、COM ポートからの受信確認は行いません。

3.3 MCU 消費電力低減処理

MCU 消費電力低減処理は、Table 3-2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_pcdc_apl_config.h”ファイル内に“USE_LPW”定義を定義してください。

Table 3-2 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	—	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）

- a) CDC 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、CDC 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。
- b) CDC 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。

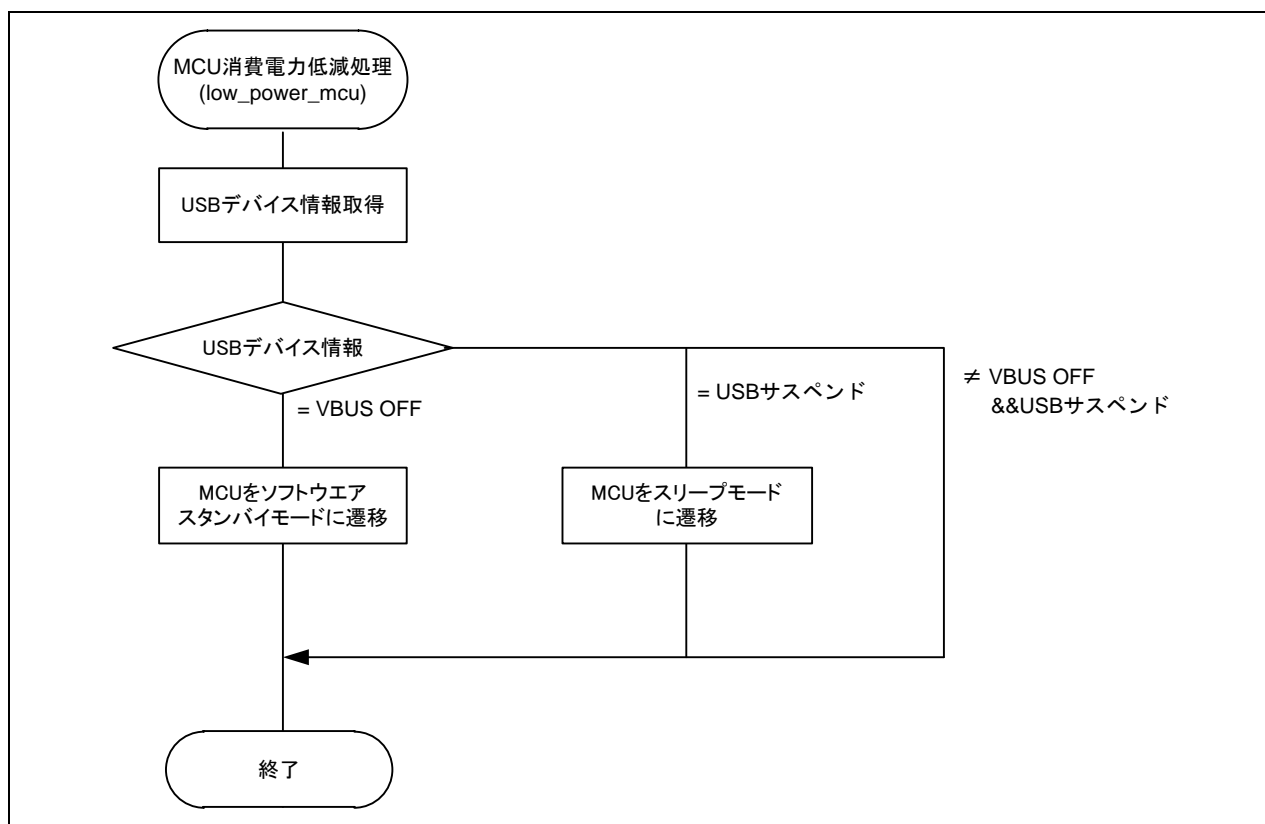


Figure 3-3 MCU 消費電力低減処理概略フロー

3.4 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_pcdc_apl_config.h)

以下の各定義に対する設定を行ってください。

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          USE_USBIP0    // USB0 モジュールを使用する場合
#define USE_USBIP          USE_USBIP1    // USB1 モジュールを使用する場合
```

[Note]

RX64M または RX71M をご使用の場合に、USE_USBIP1 を指定することができます。RX64M および RX71M 以外の MCU をご使用の場合は、USE_USBIP0 を指定してください。

2. USB_SUPPORT_SPEED 定義

CDC デバイス(RSK)がサポートする USB 動作スピード(Hi-speed/Full-speed)を指定してください。

```
#define USB_SUPPORT_SPEED  HI_SPEED      // Hi-Speed 設定
#define USB_SUPPORT_SPEED  FULL_SPEED    // Full-Speed 設定
```

[Note]

RX71M をご使用の場合に、HI_SPEED を指定することができます。RX71M 以外の MCU をご使用の場合は、FULL_SPEED を指定してください。

3. 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     USB_ECHO      // Echo モード
#define OPERATION_MODE     USB_UART      // USB-シリアル変換モード
```

4. 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、USE_LPW 定義を定義し、消費電力低減機能を使用しない場合は、USE_LPW 定義を定義しないでください。

```
#define USE_LPW            //消費電力低減機能を使用
```

5. 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 Host and Peripheral Interface Driver"アプリケーションノート」(Document No. R01AN3293JJ)を参照してください。

3.5 ディスクリプタ

PCDC のディスクリプタ情報は r_usb_pcdc_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。

4. クラスドライバ概要

4.1 クラスリクエスト（ホストからデバイスへの要求）

PCDC がサポートしているクラスリクエストをTable 4-1に示します。

Table 4-1 対応する CDC クラスリクエスト

リクエスト	コード	説明
SetLineCoding	0x21	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	0x22	通信回線設定を通知する。
SetControlLineState	0x23	通信回線制御信号 RTS、DTR の設定を行う。

4.2 データフォーマット

CDC データクラスでは、データフォーマットはありません。任意のデータ転送が可能です。

5. e² studio 用プロジェクトを CS+で使用する場合

PCDC 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。PCDC 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。

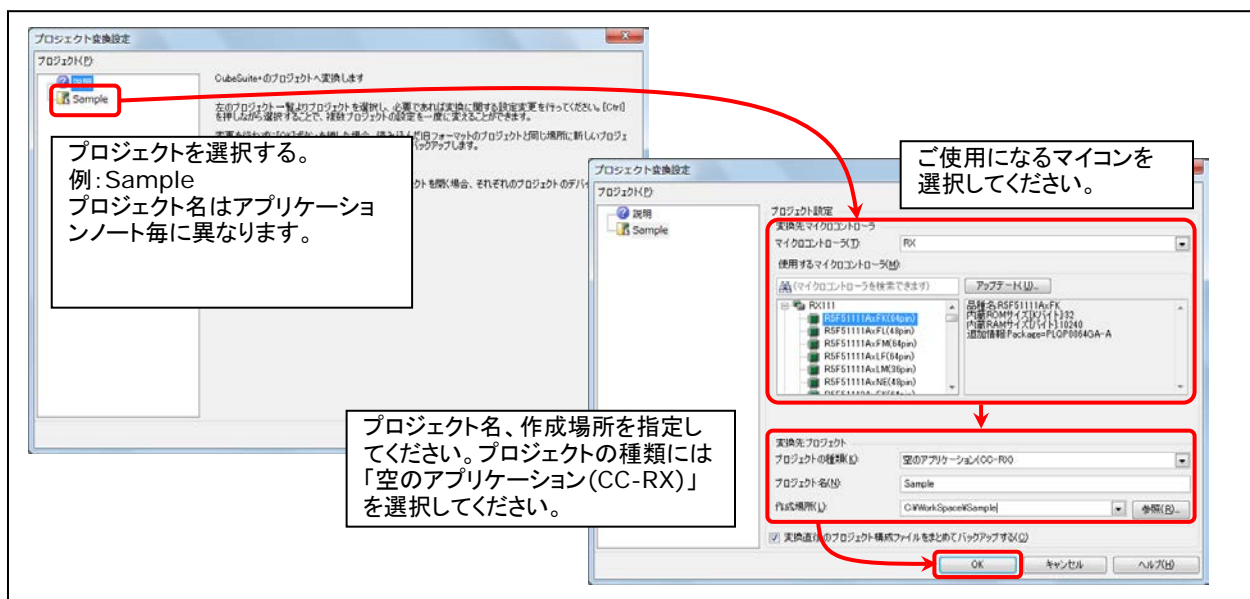
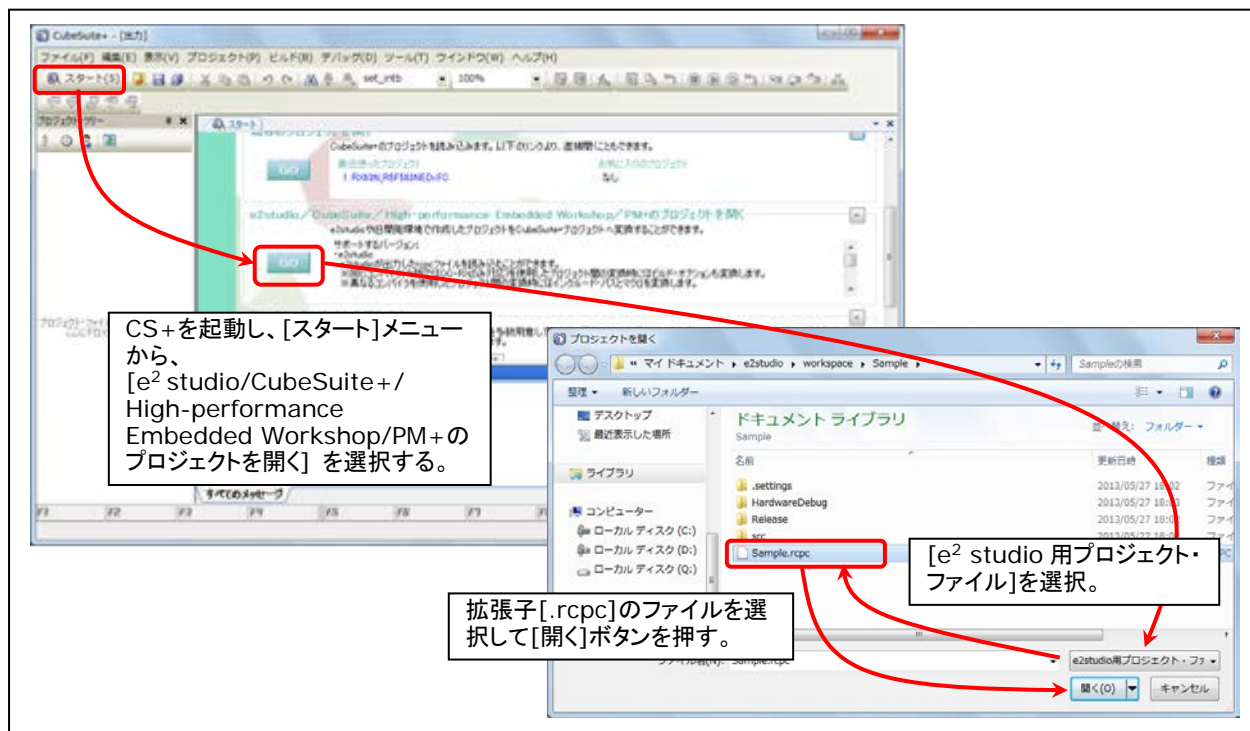


Figure 5-1 e² studio 用プロジェクトの CS+読み込み方法

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/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	Oct 16, 2014	—	初版発行
1.10	Dec 26, 2014	—	RX71M を対象デバイスに追加。
1.11	Sep 30, 2015	—	RX63N/RX631 を対象デバイスに追加。
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	—	USB Basic driver のリビジョンを Up しました。

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、電氣的特性の範囲で、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、その他の不適切に使用しないでください。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、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生じさせるおそれのある機器・システム（宇宙、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損害が生じても、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。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
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. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
 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. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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。

(Rev.3.0-1 2016.11)



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/>