

RXファミリ

R01AN0273JJ0233

Rev.2.33

Jul 31, 2019

USB Peripheral Communications Device Class Driver (PCDC)

要旨

本アプリケーションノートは、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 PCDC 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. 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. 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.R01UH0033JJ)
5. 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.R01UH0033JJ)
6. 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.R01UH0040JJ)
7. 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.R01UH0238JJ)
8. USB Basic Host and Peripheral Driver アプリケーションノート(ドキュメント No. R01AN0512JJ)

— 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	7
5. API.....	11
6. コンフィグレーション (r_usb_pcdc_config.h).....	12
7. サンプルアプリケーション	13
8. CDCドライバのインストール.....	20
9. セットアップ.....	22
10. CDCドライバのインストール.....	25
11. アプリケーションの作成方法	27
12. e ² studio用プロジェクトをCS+で使用する場合	28

1. 概要

PCDC は、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 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。

1.2 注意事項

1. 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。
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(r01an0273jj0232_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。

1.3 制限事項

本ドライバには以下の制限があります。

1. DMA/DTC をサポートしていません。

1.4 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
H/W	:	Renesas USB device
PCD	:	Peripheral control driver of
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
SCI	:	Serial Communication Interface
USB	:	Universal Serial Bus
	:	USB Basic Peripheral Driver for (non-OS)

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。

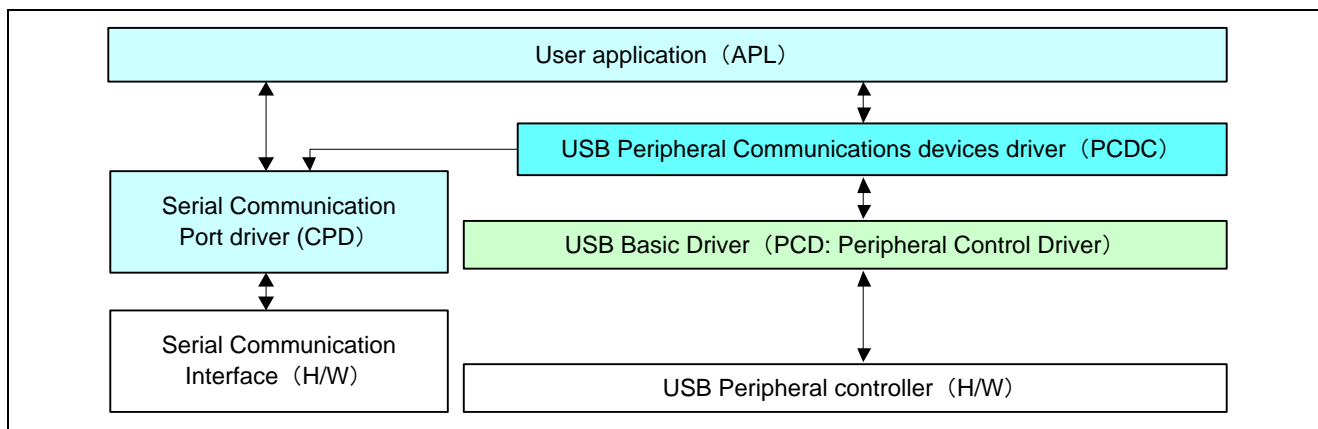


Figure 2-1 モジュール構成図

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 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
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	USBIO 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 ヘッダファイル

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。

3.5 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_pcdc_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	17.6K バイト (Note 3)	17.1K 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. 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 SW 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)

PCDC で対応しているクラスリクエストをTable 4-1に示します。

Table 4-1 CDC クラスリクエスト

リクエスト	コード	説明	対応
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 t	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	bOverRun	オーバーランエラー検出	○
D5	bParity	パリティエラー検出	○
D4	bFraming	フレーミングエラー検出	○
D3	bRingSignal	着信 (Ring signal) を感知した	×
D2	bBreak	ブレーク信号検出	×
D1	bTxCarrier	Data Set Ready : 回線が接続されて通信可能	×
D0	bRxCarrrier	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（Full-Speed:64Byte）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** 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_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 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。

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. サンプルアプリケーション

7.1 アプリケーション仕様

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)

USB ホストから受信したデータを COM ポートに送信し、COM ポート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) 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 7-1 USB-シリアル変換モードジャンパ設定

RSK+RX63N		
ポート	信号	ジャンパ設定
P20	TXD0	J12.1-2
P21	RXD0	J13.1-2

7.2 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。

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 (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 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. 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。
4. 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。
5. 上記3と4の処理が繰り返し行われます。
6. 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、「7.4 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。

以下に、APL の処理概要を示します。

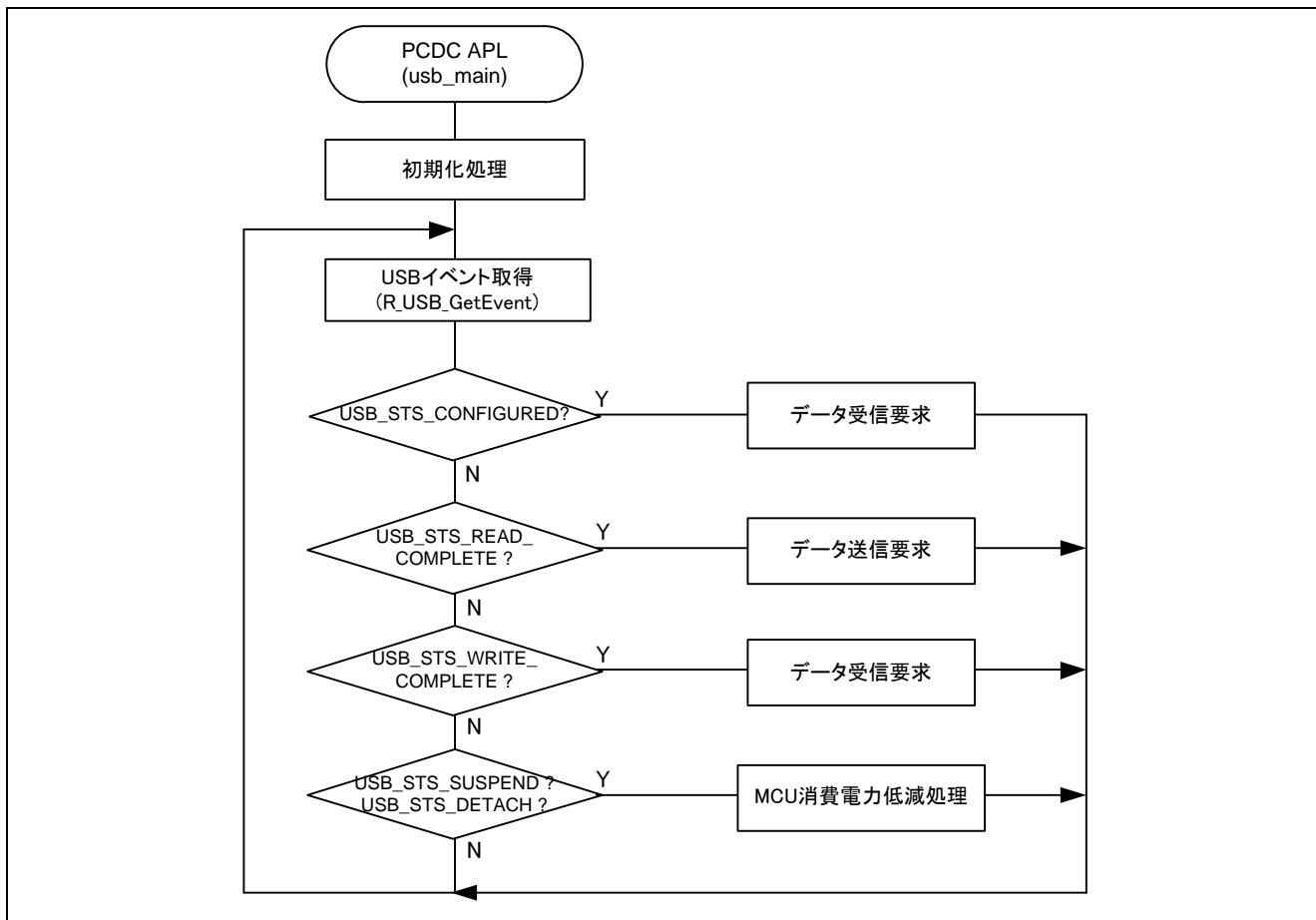


Figure 7-1 メインループ処理 (Echo モード)

7.2.3 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. 上記の処理が完了した後、以下の送信処理が行われます。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、「7.4 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。

以下に、APL の処理概要を示します。

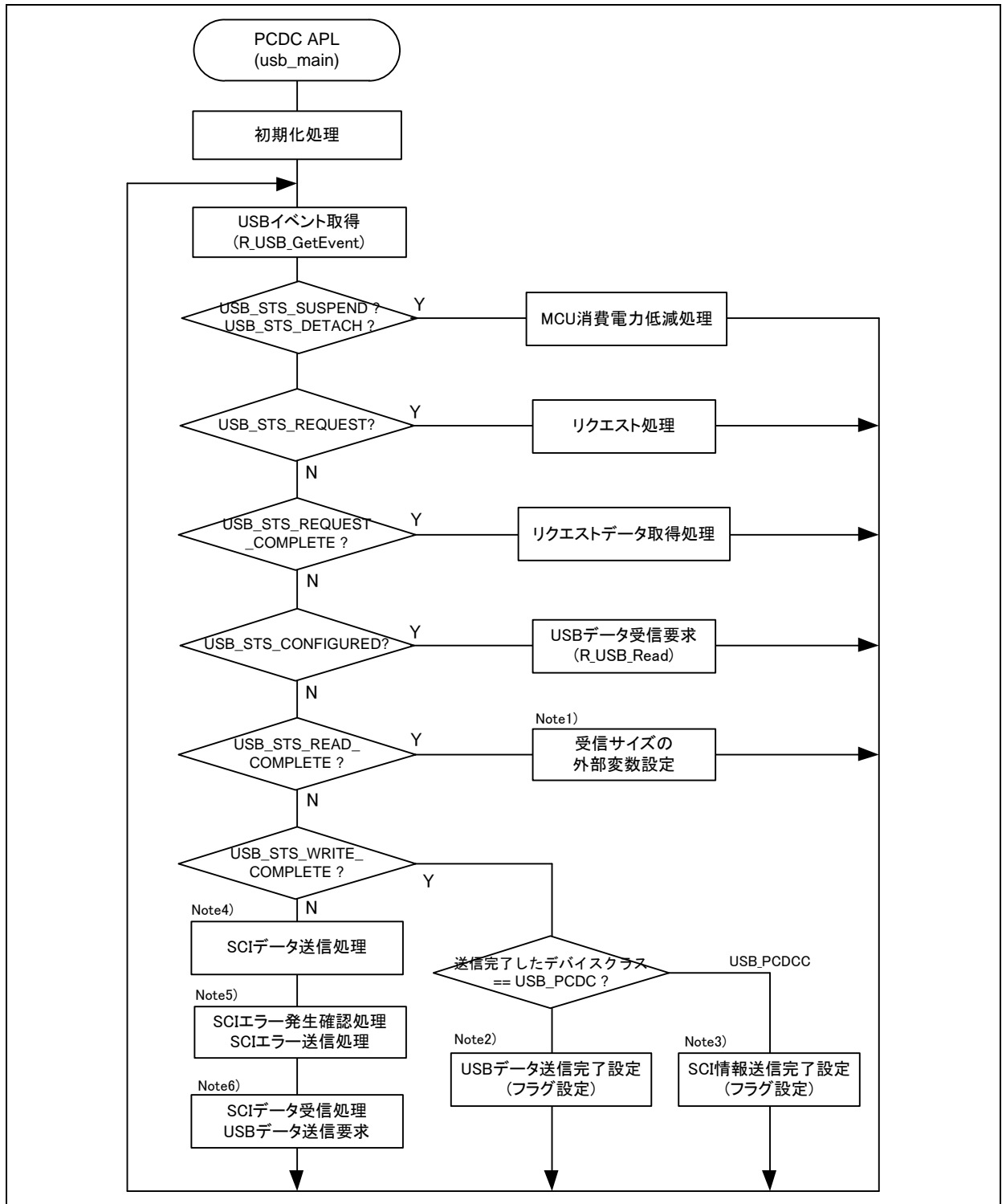


Figure 7-2 メインループ処理 (USB-シリアル変換モード)

(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。

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_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。USE_USBIP0/USE_USBIP1 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]

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

2. USB_SUPPORT_SPEED 定義

CDC デバイス(RSK)を接続する USB Host の USB 動作スピード(Full-speed)を指定してください。

```
#define USB_SUPPORT_SPEED USB_FS // Full-Speed 設定
```

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
```

[Note]

必ず r_usb_basic_config.h ファイル内の USB_CFG_CLASS_REQUEST 定義に対し USB_CFG_ENABLE を指定してください。

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、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
```

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 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 MCU 消費電力低減処理

MCU 消費電力低減処理は、以下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。

Table 7-2 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)

- 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、MSC 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。
- 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。

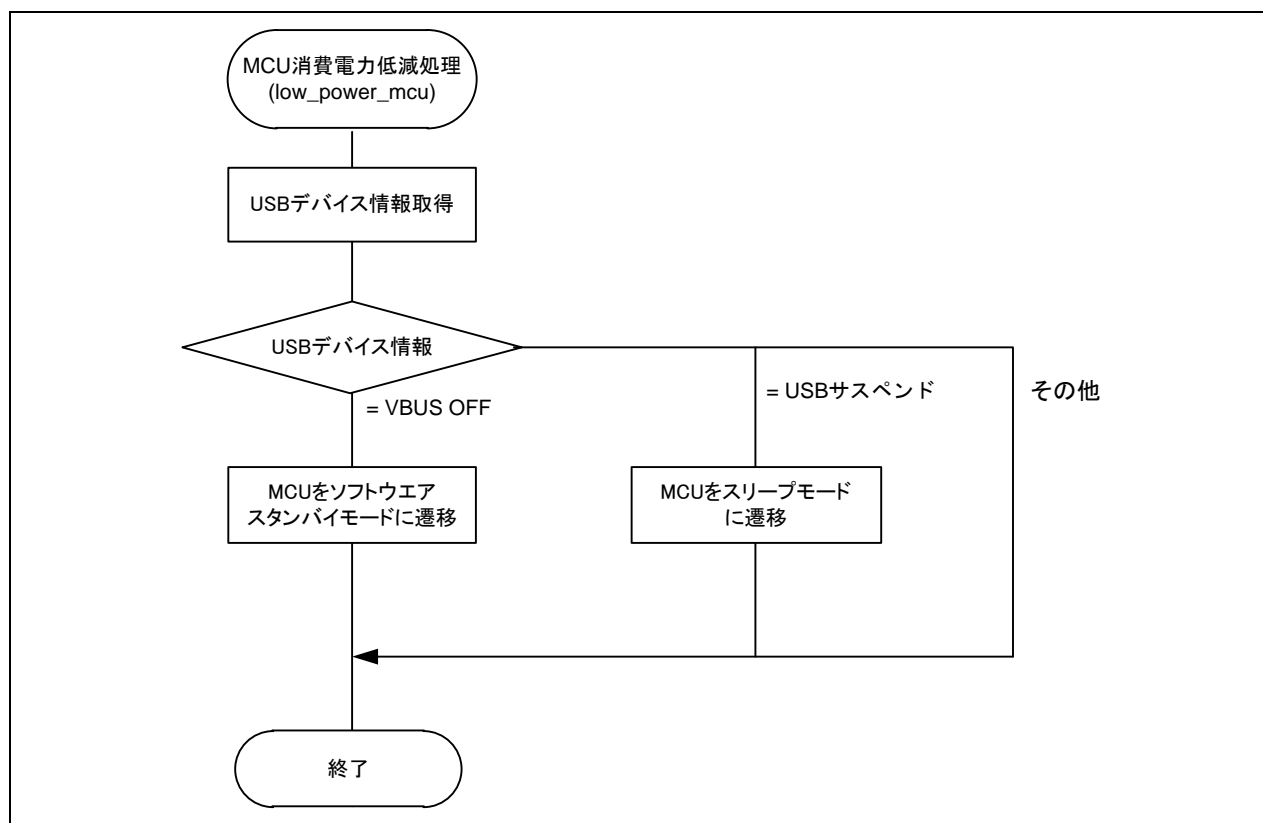


Figure 7-3 MCU 消費電力低減処理概略フロー

7.5 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。

8. CDC ドライバのインストール

USB HostがPCの場合、そのPCに対しCDCドライバをインストールする必要があります。本サンプルプログラムの書き込みを行ったRSKをPCに接続すると、Figure 10-1に示すウィザードが表示され、CDCドライバのインストールが行われます。

- (1). デバイス・マネージャより、ドライバーソフトウェアの更新を選択します。
- (2). 《コンピューターを参照してドライバーソフトウェアを検索します (R) 》を選択します。

Note:

- (1). PCのOSがWindows 10の場合、CDCドライバのインストール作業は不要です。
- (2). PCのOSがWindows® 8.1の場合、デジタル署名済のカタログファイルが必要になります。デジタル署名済のカタログファイルはお客様により作成いただく必要があります。

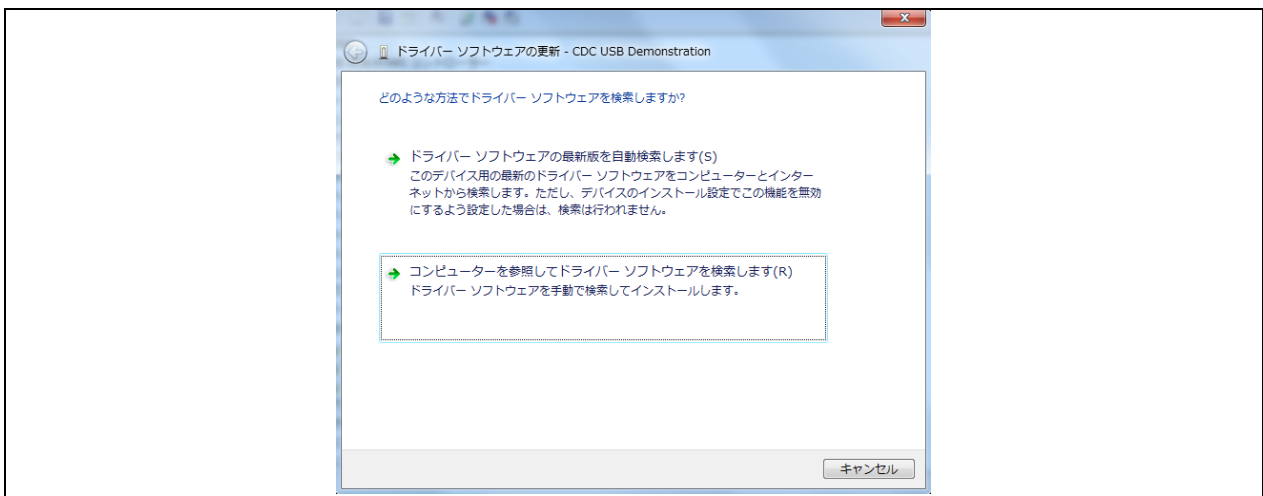


Figure 8-1 新しいハードウェアの検索ウィザード

- (3). 《次の場所で最適なドライバーソフトウェアを検索します》を選択します。

“参照 (R)” をクリックして“CDC_Demo.inf”の存在するフォルダを指定し、“次へ (N)” をクリックしてください。

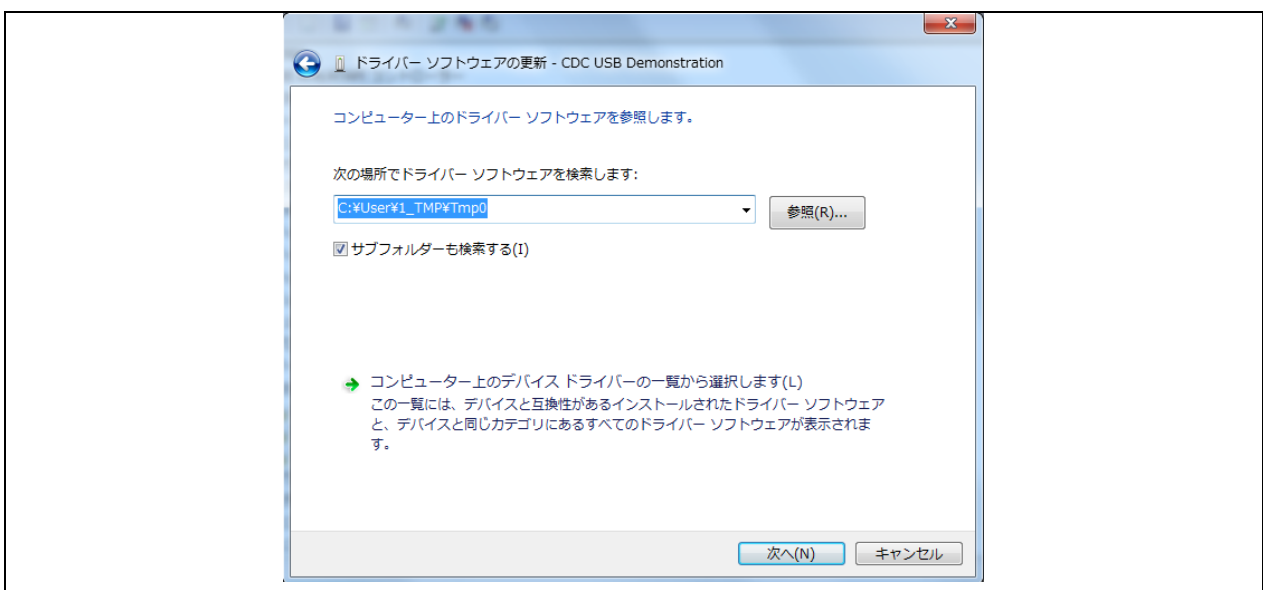


Figure 8-2 ドライバの場所の選択

Note:

CDC_Demo.inf ファイルは、パッケージ内の "r_usb_pcdc¥utilities" フォルダに格納されています。

- (4). 次のインストール確認画面が表示される場合は、“このドライバーソフトウェアをインストールします (I)” をクリックしてください。

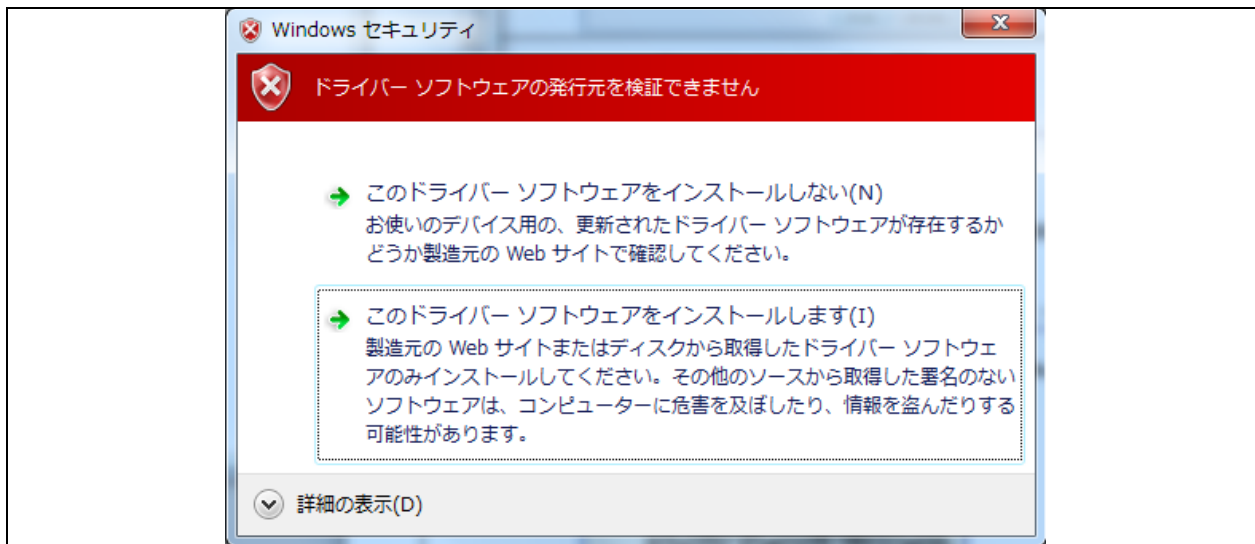


Figure 8-3 インストール確認

- (5). 次のウィンドウが表示されたら、CDC ドライバのインストールは完了です。“閉じる” をクリックしてください。

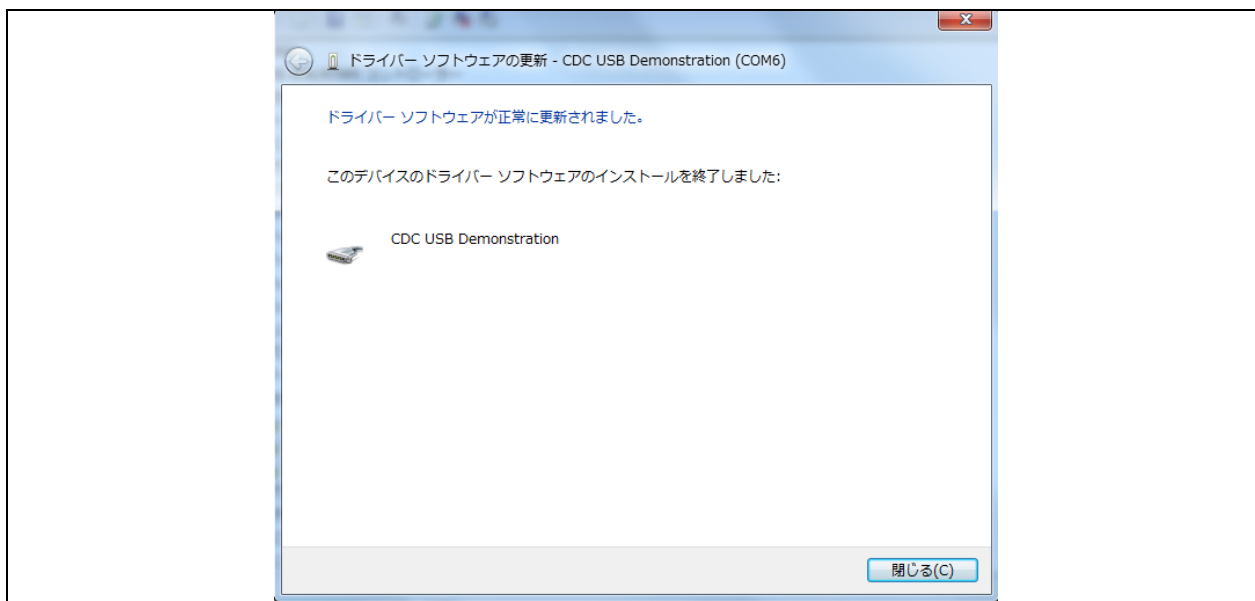


Figure 8-4 インストール完了

9. セットアップ

9.1 ハードウェア

9.1.1 動作環境例

PCDC の動作環境例をFigure 9-1 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。

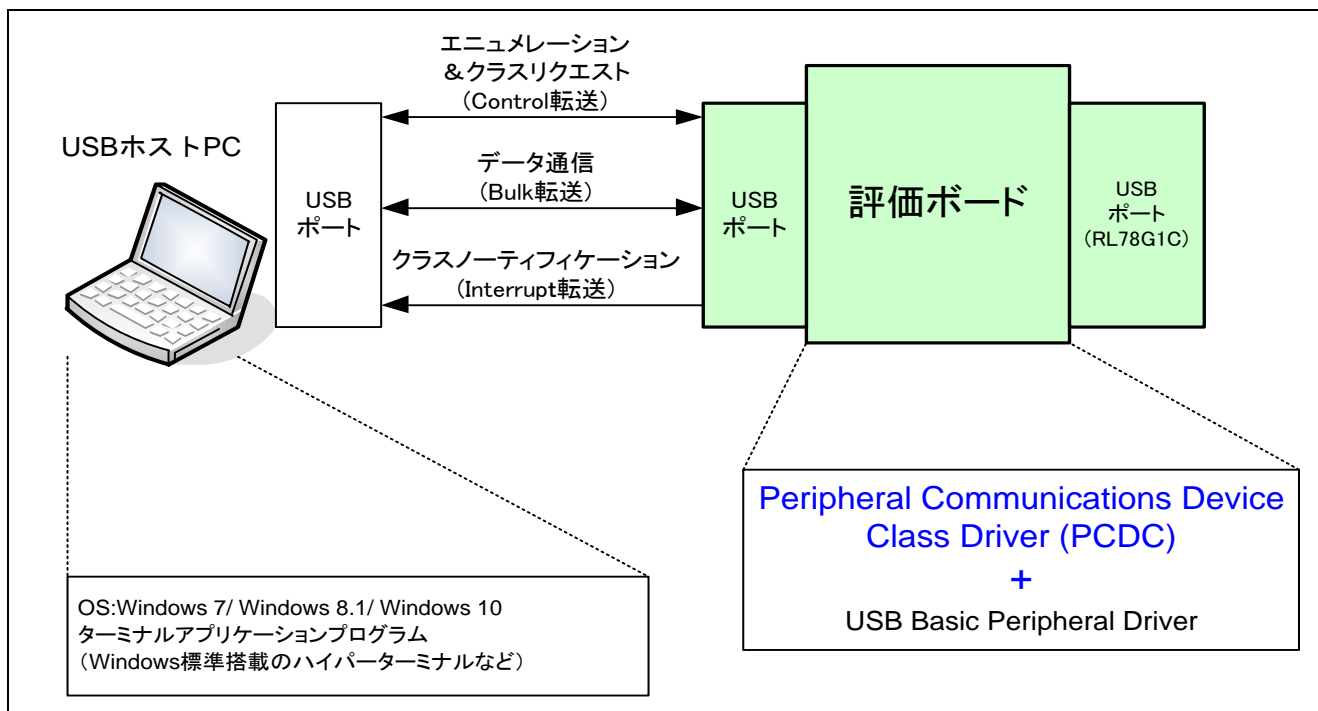


Figure 9-1 動作環境例

9.1.2 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 9-1 RSK 設定

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

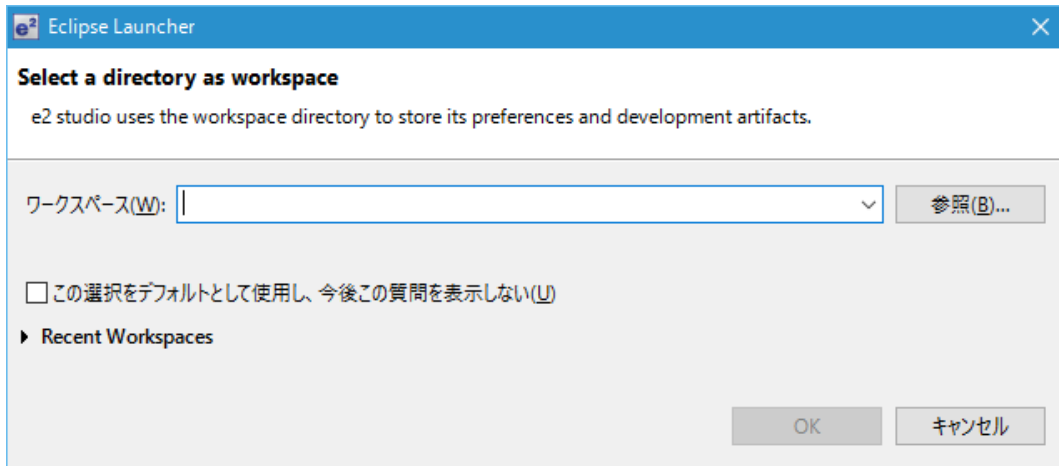
[Note]

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。

9.2 ソフトウェア

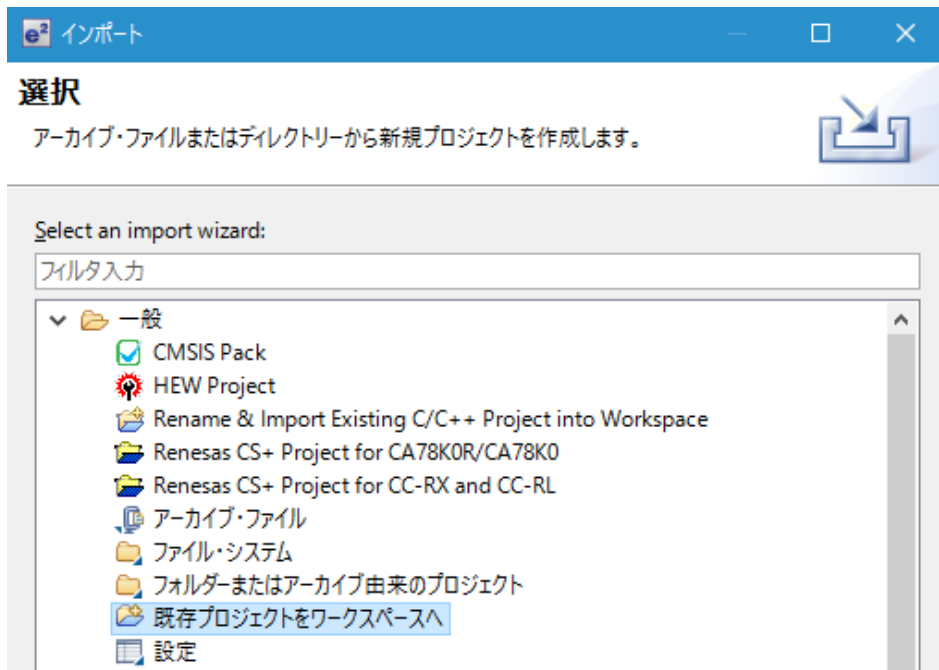
(1). e² studio を起動

- 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。

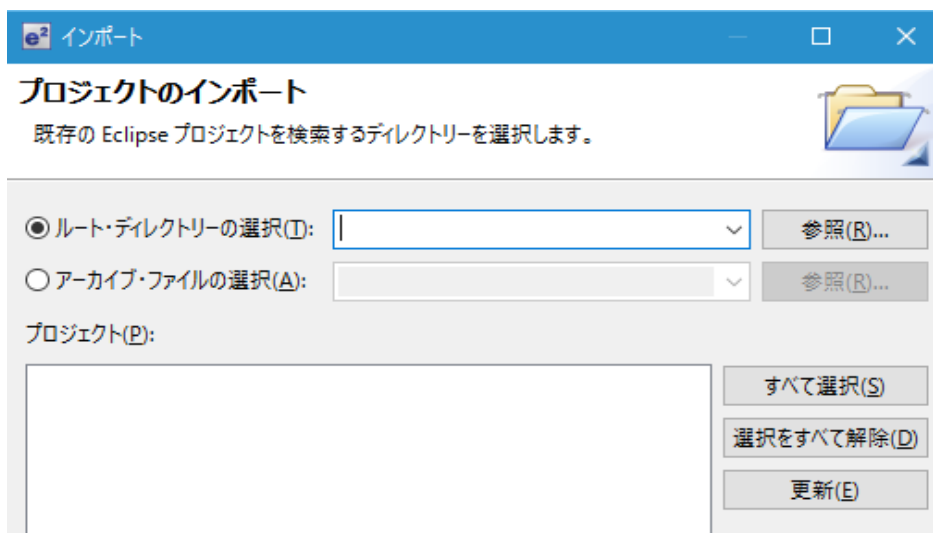


(2). ワークスペースへのプロジェクトの登録

- 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。



- c) 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。



- d) “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。

- (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。

10. CDC ドライバのインストール

USB HostがPCの場合、そのPCに対しCDCドライバをインストールする必要があります。本サンプルプログラムの書き込みを行ったRSKをPCに接続すると、Figure 10-1に示すウィザードが表示され、CDCドライバのインストールが行われます。

- (6). デバイス・マネージャより、ドライバーソフトウェアの更新を選択します。
- (7). 《コンピューターを参照してドライバーソフトウェアを検索します (R) 》を選択します。

Note:

PCのOSがWindows 10の場合、CDCドライバのインストール作業は不要です。

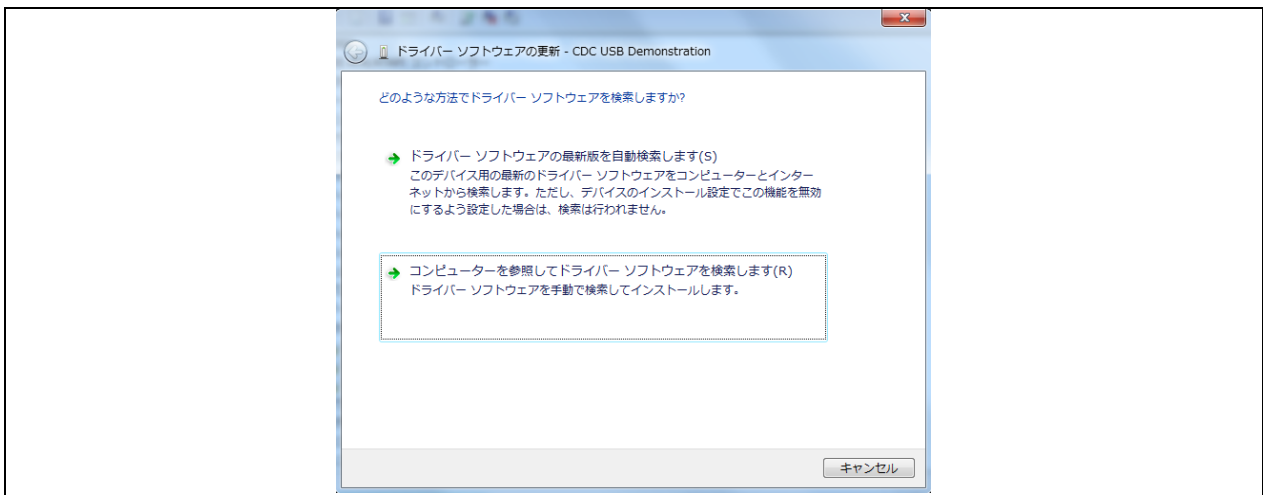


Figure 10-1 新しいハードウェアの検索ウィザード

- (8). 《次の場所で最適のドライバーソフトウェアを検索します》を選択します。

“参照 (R) ” をクリックして“CDC_Demo.inf”の存在するフォルダを指定し，“次へ (N) ” をクリックしてください。

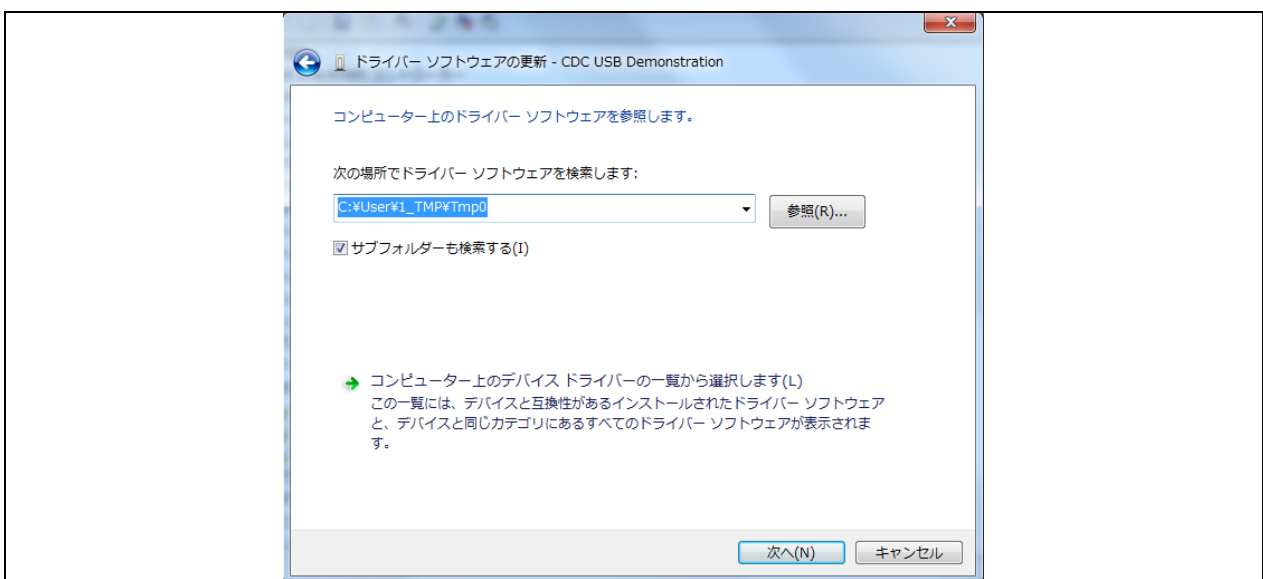


Figure 10-2 ドライバの場所の選択

Note:

CDC_Demo.inf ファイルは、パッケージ内の "r_usb_pcdc¥utilities" フォルダに格納されています。

- (9). 次のインストール確認画面が表示される場合は、“このドライバーソフトウェアをインストールします (I)” をクリックしてください。

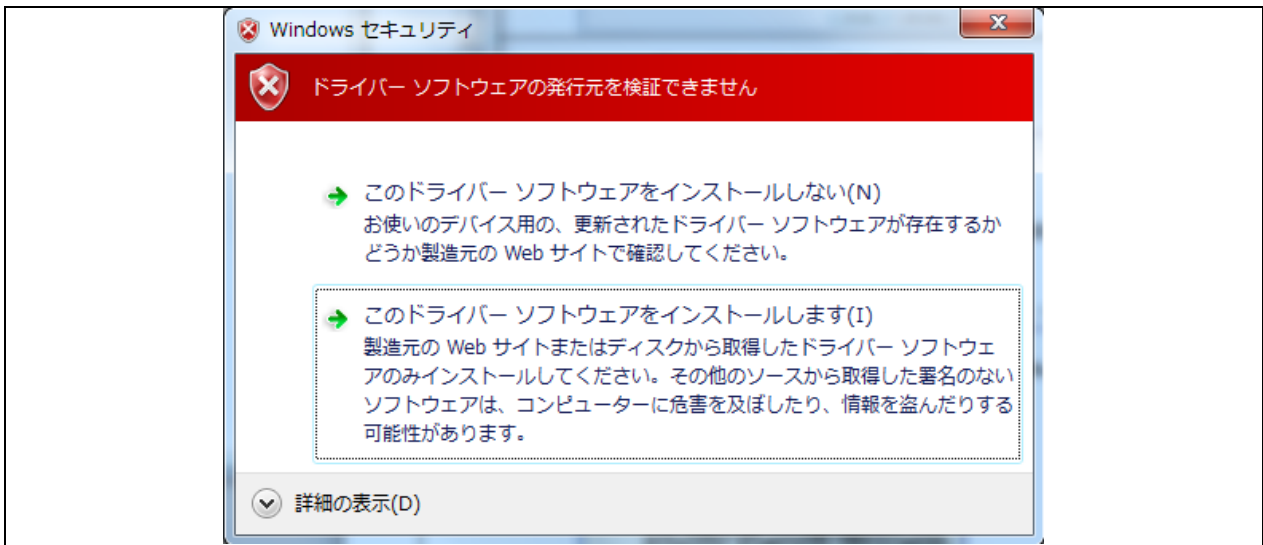


Figure 10-3 インストール確認

- (10). 次のウィンドウが表示されたら、CDC ドライバのインストールは完了です。“閉じる” をクリックしてください。

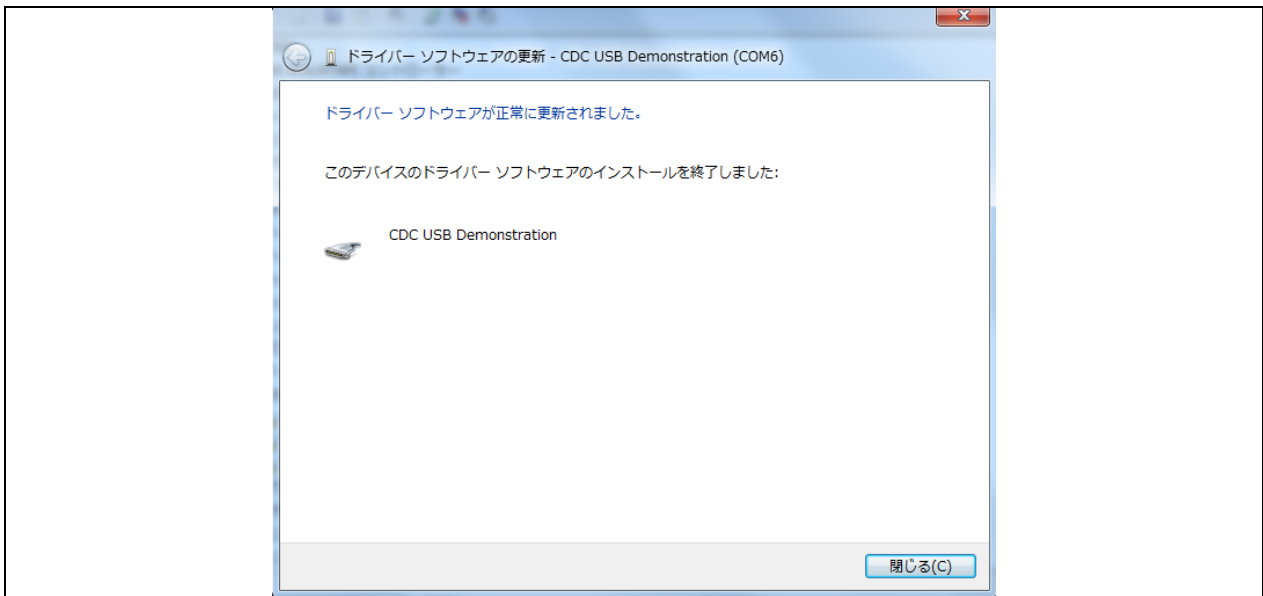


Figure 10-4 インストール完了

※ Windows 8.1 環境でドライバをインストールする場合、インストール確認が出ずにエラーになる場合があります。

11. 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.R01AN0512JJ)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。

12. 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。

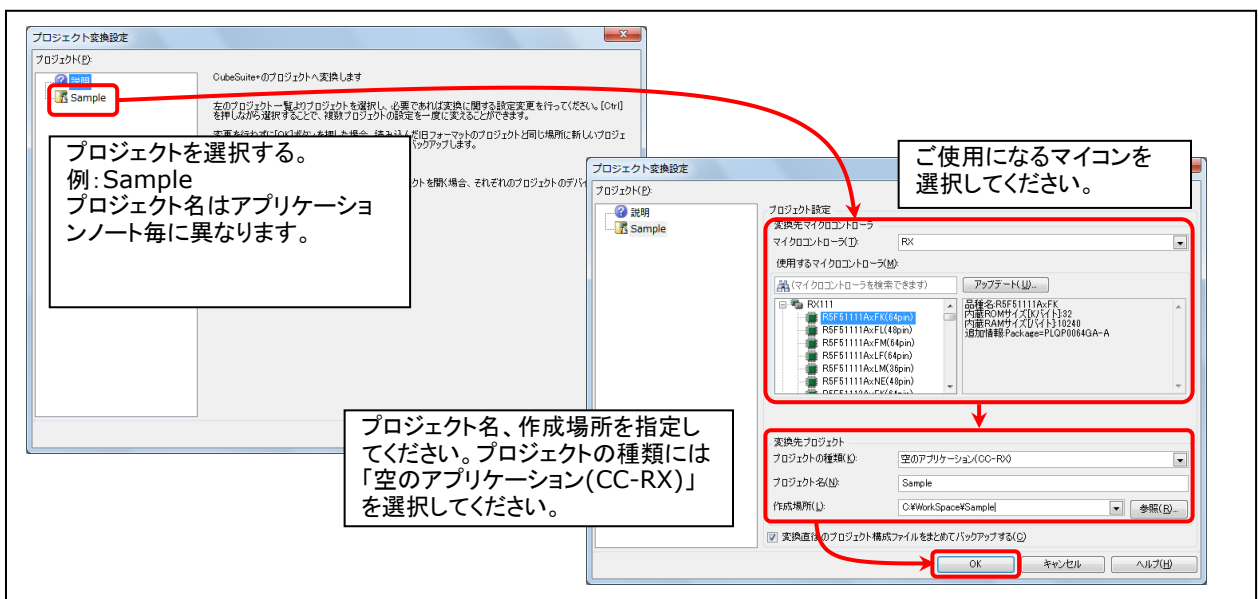
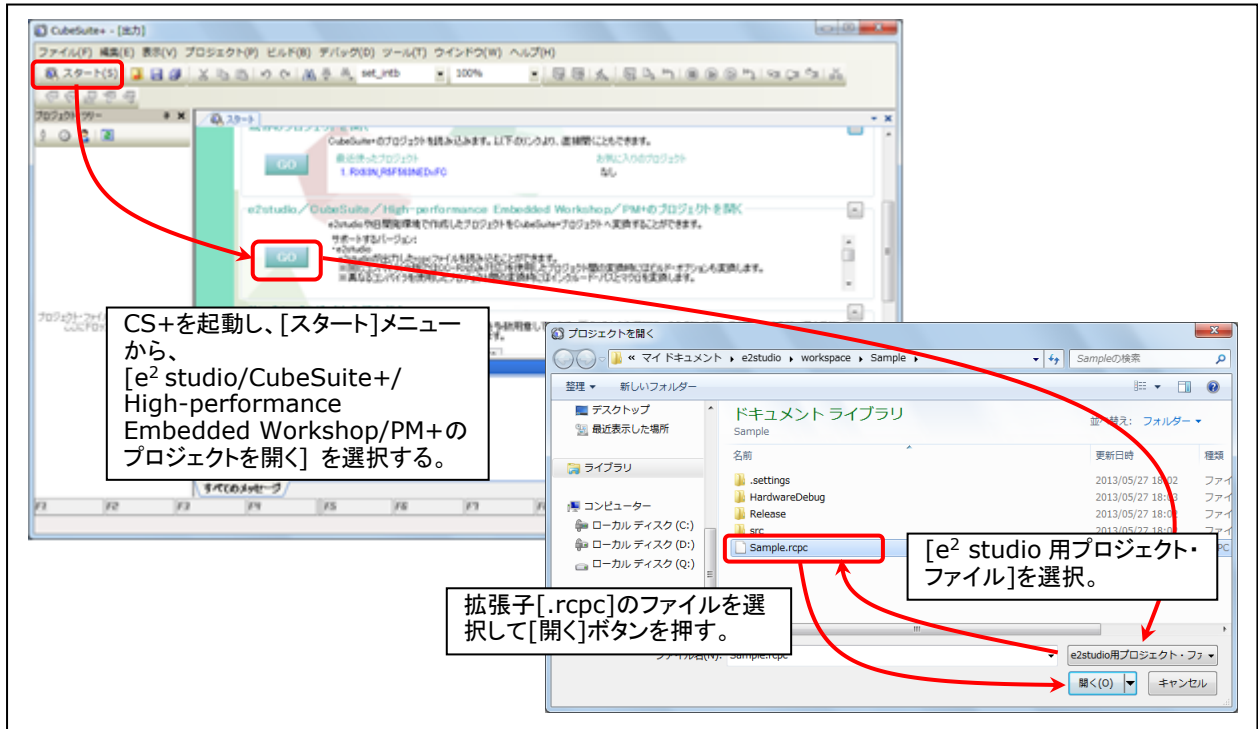


Figure 12-1 e² studio 用プロジェクトの CS+読み込み方法

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/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	2011.03.22	—	初版発行
1.10	2011.08.10	表紙,他	動作確認デバイスに RX630、R8A66597 の追加 上記追加に伴い、RX630 に関する内容、R8A66597(Hi-Speed USB)に関する内容を追加
		4	2.2 ファイル一覧のシリアルポートドライバのファイル削除
		29	5.ペリフェラル用 CDC サンプルアプリケーション(APL)のアプリケーション仕様変更 ・ SW1 使用禁止 ・ SW 押下表記を機能スイッチ表記に変更
		50	6.コミュニケーションポートドライバ(CPD)の関数名、定義値の変更 ・ "R_usb_ComPort"、"usb_ComPort"を"USB_cpu_Sci"に変更 ・ "USB_PCDC_"を"USB_SCI"に変更
2.00	2012.7.1	—	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	2013.2.1	—	"1.4 本書の読み方"を追加、その他ケアレスミスの修正
2.10	2013.4.1	—	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	Sep 30, 2016	—	1. DMA 転送をサポート 2. 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	Sep 30, 2017	—	1. DMA/DTC 転送を非サポートに変更 2. 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	Mar 31, 2018	—	USB Basic Driver のリビジョンをアップしました。
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/>