

RXファミリ

R01AN0275JJ0233

Rev.2.33

Jul 31, 2019

USB Host Communications Devices Class Driver (HCDC)

要旨

本アプリケーションノートは、USB Host コミュニケーションデバイスクラスドライバ (HCDC) について説明します。本ドライバは USB Basic Host Drive(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 HCDC と称します。

対象デバイス

RX62N/RX621 グループ

RX63N/RX631 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
【<http://www.usb.org/developers/docs/>】
 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.R01UH0033)
 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.R01UH0041)
 6. 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)
 7. 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.....	5
3. API情報	6
4. ターゲットペリフェラルリスト (TPL)	8
5. コミュニケーションデバイスクラス (CDC) ,PSTN and ACM	9
6. USBホストコミュニケーションデバイスクラスドライバ (HCDC)	14
7. API.....	17
8. コンフィグレーション (r_usb_hcdc_config.h).....	18
9. サンプルアプリケーション	19
10. セットアップ	23
11. アプリケーションの作成方法	26
12. e ² studio用プロジェクトをCS+で使用する場合	27

1. 概要

HCDC は、と組み合わせることで、USB Host コミュニケーションデバイスクラスドライバ（以降 HCDC と記述）として動作します。HCDC 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 PSTN デバイス・サブクラス Abstract Control Model に準拠し、CDC ペリフェラル装置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。

- ・ 接続デバイスの照合
- ・ 通信回線設定の実施
- ・ 通信回線の状態取得
- ・ CDC ペリフェラルデバイス機器とのデータ通信
- ・ 一つの USB チャンネルに対し USB Hub を使って最大 2 つの CDC デバイスの接続が可能。

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 をご使用の場合、必ず"reference_documents"フォルダ下のドキュメント (r01an0275jj0232_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. USB Hub は 1 段のみ使用することができます。
2. USB Hub および USB Hub ダウンポートに接続した CDC デバイスに対するサスペンド/レジュームには対応しておりません。
3. データ転送中のサスペンドはサポートしていません。データ転送が完了したことを確認の上、サスペンドを実行して下さい。
4. CDC クラスをサポートしている複合 USB デバイスはサポートしていません。
5. DMA/DTC をサポートしていません。

1.4 用語一覧

APL	: Application program
CDC	: Communications Devices Class
CDCC	: Communications Devices Class – Communications Class Interface
CDCD	: Communications Devices Class – Data Class Interface
HCD	: Host control driver of
HCDC	: Host用 Communication Devices Class
HDCD	: Host device class driver (device driver and USB class driver)
HUBCD	: Hub class sample driver
MGR	: Peripheral device state manager of HCD
non-OS	: USB basic firmware for OS less system
RSK	: Renesas Starter Kit

USB : Universal Serial Bus
: USB Basic Host Driver for (non-OS/ RTOS)

2. ソフトウェア構成

Figure 2-1に HCDC 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。

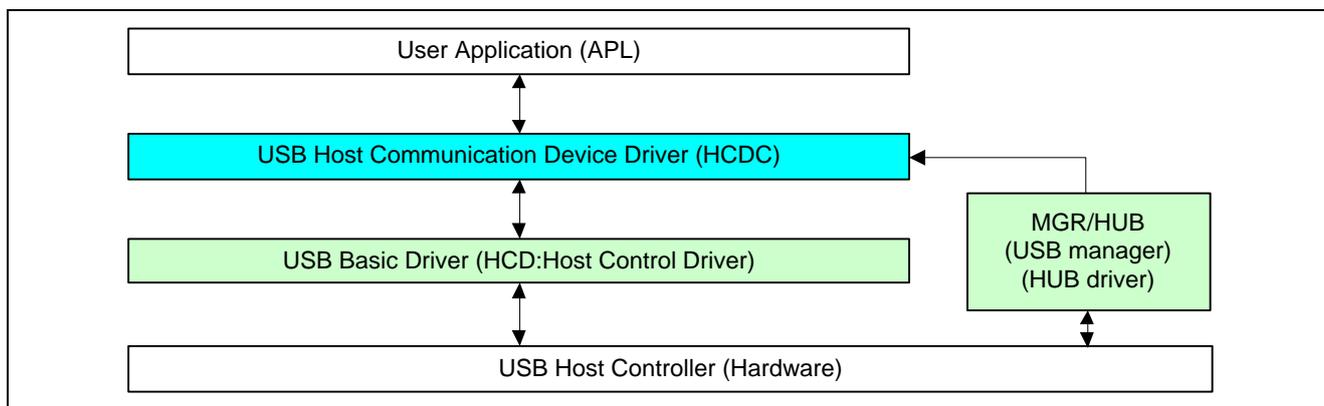


Figure 2-1 モジュール構成図

Table 2-1 モジュール説明

モジュール名	説明
HCDC	APL からの CDC に関するリクエストおよびデータ通信を HCD へ要求します。
MGR / HUB	接続されたデバイスとエnumレーションをして HCDC を起動します。またデバイスの状態管理も行います。
HCD	USB Host H/W 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 V3.01.00 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

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/RX631	USBIO 割り込み(ベクタ番号: 35) / USBR0 割り込み(ベクタ番号: 90)

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_hcdc_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「8. コンフィグレーション (`r_usb_hcdc_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	33.3K バイト (Note 3)	32.7K バイト (Note 4)
RAM サイズ	16.6K バイト	16.6K 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. ターゲットペリフェラルリスト (TPL)

TPL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「ターゲットペリフェラルリスト(TPL)の設定方法」の章を参照してください。

5. コミュニケーションデバイスクラス (CDC) ,PSTN and ACM

本 S/W は、コミュニケーションデバイスクラス仕様 Abstract Control Model (ACM) サブクラスに準拠しています。なお、Abstract Control Model 仕様は、「関連ドキュメント」に記載されている PSTN に仕様が定められています。

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。

5.1 基本機能

HCDC の主な機能を以下に示します。

1. 接続デバイスの照合
2. 通信回線設定の実施
3. 通信回線の状態取得
4. CDC ペリフェラルデバイス機器とのデータ通信

5.2 クラスリクエスト (ホスト→デバイスへの要求)

本ドライバがサポートしているクラスリクエストをTable5-1に示します。

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。

Table5-1 CDC クラスリクエスト

リクエスト	コード	説明
SendEncapsulatedCommand	0x00	プロトコルで定義された AT コマンド等を送信する。
GetEncapsulatedResponse	0x01	SendEncapsulatedCommand で送信したコマンドに対するレスポンスを要求する。
SetCommFeature	0x02	機器固有の 2 バイトコードや、カントリー設定の禁止/許可を設定する。
GetCommFeature	0x03	機器固有の 2 バイトコードや、カントリー設定の禁止/許可状態を取得する。
ClearCommFeature	0x04	機器固有の 2 バイトコードや、カントリー設定の禁止/許可設定をデフォルト状態に戻す。
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。
SendBreak	0x23	ブレイク信号の送信を行う。

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。

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記します。

5.2.1 SendEncapsulatedCommand

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

Table5-2 SendEncapsulatedCommand データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SEND_ENCAPSULATED_COMMAND (0x00)	0x0000	0x0000	データレングス	制御プロトコルコマンド

【注】 Data にはモデム制御の為に AT コマンド等を設定し、wLength にはその長さを設定します。

5.2.2 GetEncapsulatedResponse

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

Table5-3 GetEncapsulatedResponse データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	GET_ENCAPSULATED_RESPONSE (0x01)	0x0000	0x0000	データレングス	プロトコルに依存したデータ

【注】 Data には SendEncapsulatedCommand に対する応答データが渡され、wLength にはその長さが設定されます。

5.2.3 SetCommFeature

SetCommFeature データフォーマットを以下に示します。

Table5-4 SetCommFeature データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_COMM_FEATURE (0x02)	Feature Selector 注	0x0000	データレングス	State Feature Selector によりカントリーコード又は Abstract Control Model アイドル設定、多重化設定の何れかになります。

【注】 Table5-6 Feature Selector 設定一覧参照

5.2.4 GetCommFeature

GetCommFeature データフォーマットを以下に示します。

Table5-5 GetCommFeature データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	GET_COMM_FEATURE (0x03)	Feature Selector 注	0x0000	データレングス	Status Feature Selector によりカントリーコード又は Abstract Control Model アイドル設定、多重化設定の何れかになります。

【注】 Table5-6 Feature Selector 設定一覧参照

Feature selector 設定をTable5-6に、ABSTRACT_STATE 時の Status フォーマットをTable5-7に示します。

Table5-6 Feature Selector 設定一覧

Feature Selector	Code	Targets	Length of Data	Description
RESERVED	00h	None	None	リザーブ
ABSTRACT_STATE	01h	Interface	2	Abstract Control Model のアイドルステート、多重化信号についての設定を選択します。
COUNTRY_SETTING	02h	Interface	2	ISO3166 で定義される 16 進形式のカントリーコードを選択します。

Table5-7 ABSTRACT_STATE 選択時、Status フォーマット

ビット position	Description
D15~D2	リザーブ
D1	データ多重化設定 1 : Data クラスでコールマネジメントコマンドの多重化を許可 0 : 多重化禁止
D0	アイドル設定 1 : 対象インタフェースの全てのエンドポイントにおいてホストからのデータを受け入れず、ホストに対してデータを提供しません。 0 : エンドポイントにおいてデータの受け入れ、提供をし続けます。

5.2.5 ClearCommFeature

ClearCommFeature データフォーマットを以下に示します。

Table5-8 ClearCommFeature データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	CLEAR_COMM_FEATURE (0x04)	Feature Selector 注	0x0000	0x0000	None

【注】 Table5-6 Feature Selector 設定一覧参照

5.2.6 SetLineCoding

SetLineCoding データフォーマットを以下に示します。

Table5-9 SetLineCoding データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_LINE_CODING(0x20)	0x0000	0x0000	0x0007	Line Coding Structure Table5-10 Line Coding Structure フォーマット参照

Line Coding Structure フォーマットを以下に示します。

Table5-10 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 ビット 1 - 1.5 Stop ビット

2-2 Stop ビット					
5	bParityType	1	Number	パリティ	0 - None 1 - Odd 2 - Even 3 - Mask 4 - Space
6	bDataBits	1	Number	データビット	(5、6、7、8)

5.2.7 GetLineCoding

GetLineCoding データフォーマットを以下に示します。

Table5-11 GetLineCoding データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_LINE_CODING (0x21)	0x0000	0x0000	0x0007	Line Coding Structure Table5-10 Line Coding Structure フォーマット 参照

5.2.8 SetControlLineState

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

Table5-12 SetControlLineState データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_CONTROL_LINE_ STATE (0x22)	Control Signal ビット map Table5-13 Control Signal ビット map フォーマット参照	0x0000	0x0000	None

Table5-13 Control Signal ビット map フォーマット

ビット Position	Description
D15...D2	予約 (0 をセット)
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

5.2.9 SendBreak

SendBreak データフォーマットを以下に示します。

Table5-14 SendBreak データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SEND_BREAK (0x23)	ブレーク 信号出力 時間	0x0000	0x0000	None

5.3 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）

HCDC のクラスノーティフィケーション対応/非対応を以下に示します。

Table5-15 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	○

5.3.1 SerialState

SerialState データフォーマットを以下に示します。

Table5-16 SerialState フォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	SERIAL_STATE (0x20)	0x0000	0x0000	0x0002	UART State ビット map Table5-17 UART State ビットマップフォーマット 参照

UART State ビットマップフォーマットを以下に示します。

Table5-17 UART State ビットマップフォーマット

ビット	Field	Description
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

5.3.2 ResponseAvailable

ResponseAvailable データフォーマットを以下に示します。

Table5-18 ResponseAvailable データフォーマット

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	RESPONSE_AVAI LABLE(0x01)	0x0000	0x0000	0x0000	None

6. USB ホストコミュニケーションデバイスクラスドライバ (HCDC)

6.1 基本機能

HCDC 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。HCDC の主な機能を以下に示します。

- (1). CDC ペリフェラルデバイスに対してクラスリクエスト要求
- (2). CDC ペリフェラルデバイスとのデータ通信
- (3). CDC ペリフェラルデバイスからのシリアル通信エラー情報受信

6.2 構造体 / 共用体

以下の構造体/共用体は、r_usb_hcdc_if.h ファイル内に定義されています。

6.2.1 HCDC クラスリクエスト構造体

CDC のクラスリクエスト SetLineCoding 及び、GetLineCoding で使用する UART 設定パラメータ用の構造体を以下に示します。

Table6-1 usb_hcdc_linecoding_t 構造体

型	メンバ名	説明	備考
uint32_t	dwdte_rate	回線速度	単位 : bps
uint8_t	bchar_format	ストップビット設定	
uint8_t	bparity_type	パリティ設定	
uint8_t	bdata_bits	データビット長	

CDC リクエスト SetControlLineState で使用する UART 設定パラメータ用の構造体を以下に示します。

Table6-2 usb_hcdc_controllinestate_t Structure

型	メンバ名	説明	備考
uint16_t (D1)	brts:1	Carrier control for half duplex modems 0 - Deactivate carrier, 1 - Activate carrier	
uint16_t (D0)	bdtr:1	Indicates to DCE if DTE is present or not 0 - Not Present, 1 - Present	

CDC リクエスト SendEncapsulatedCommand および GetEncapsulatedResponse で使用する AT コマンドパラメータ用の構造体を以下に示します。

Table6-3 usb_hcdc_encapsulated_t Structure

型	メンバ名	説明	備考
uint8_t	*p_data	AT コマンドデータが格納されている領域	
uint16_t	wlength	AT コマンドデータのサイズ	単位 : byte

CDC リクエスト SendBreak で使用する Break 信号送出パラメータ用の構造体を以下に示します。

Table6-4 usb_hcdc_breakduration_t Structure

型	メンバ名	説明	備考
uint16_t	wtime_ms	Duration of Break	単位 : ms

6.2.2 CommFeature 機能選択共用体

CDC リクエスト *SetCommFeature* および *GetCommFeature* で使用する“Feature Selector”パラメータ用の構造体をTable6-5とTable6-6に、共用体をTable6-7に示します。

Table6-5 usb_hcdc_abstractstate_t Structure

型	メンバ名	説明	備考
uint16_t	bdms:1	Data Multiplexed State	
uint16_t	bis:1	Idle Setting	

Table6-6 usb_hcdc_countrysetting_t Structure

型	メンバ名	説明	備考
uint16_t	country_code	Country code in hexadecimal format as defined in [ISO3166],	

Table6-7 usb_hcdc_commfeature_t 共用体

型	メンバ名	説明	備考
usb_hcdc_abstractstate_t	abstract_state	Abstract Control Model 選択時パラメータ	
usb_hcdc_countrysetting_t	country_setting	Country Setting 選択時パラメータ	

6.2.3 CDC ノートیفケーションフォーマット

Table6-8とTable6-9に CDC ノートیفケーションのデータフォーマットを示します。

Table6-8 Response_Available notification format

Type	Member	Description	Remarks
uint8_t	bmRequestType	0xA1	
uint8_t	bRequest	RESPONSE_AVAILABLE(0x01)	
uint16_t	wValue	0x0000	
uint16_t	wIndex	Interface	
uint16_t	wLength	0x0000	
uint8_t	Data	none	

Table6-9 Serial_State notification format

Type	Member	Description	Remarks
uint8_t	bmRequestType	0xA1	
uint8_t	bRequest	SERIAL_STATE(0x20)	
uint16_t	wValue	0x0000	
uint16_t	wIndex	Interface	
uint16_t	wLength	0x0002	
uint16_t	Data	UART State bitmap	Table6-10参照

UART ポート状態変化の検出によりホストに通知されるクラスノートیفケーション“SerialState”の UART State Bitmap 構造体を以下に示します。

Table6-10 usb_hcdc_serialstate_t 構造体

型	メンバ名	説明	備考
uint16_t (D6)	bover_run:1	オーバーランエラー検出	
uint16_t (D5)	bparity:1	パリティエラー検出	
uint16_t (D4)	bframing:1	フレミングエラー検出	
uint16_t (D3)	bring_signal:1	着信 (Ring signal) を感知	
uint16_t (D2)	bbreak:1	ブレイク信号検出	
uint16_t (D1)	btx_carrier:1	回線が接続されて通信可能	Data Set Ready
uint16_t (D0)	brx_carrier:1	回線にキャリア検出	Data Carrier Detect

7. 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。

8. コンフィグレーション (r_usb_hcdc_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. CDC デバイス複数接続設定

同時に複数の CDC デバイスを接続し、USB 通信を行う場合は以下の定義に対し、USB_CFG_ENABLE を指定してください。同時に複数の CDC デバイスを接続することが無い場合は、USB_CFG_DISABLE を指定してください。

```
#define USB_CFG_HCDC_MULTI USB_CFG_ENABLE // 複数接続対応
#define USB_CFG_HCDC_MULTI USB_CFG_DISABLE // 複数接続非対応
```

2. CDC クラス設定

接続する CDC デバイスのデバイスクラス ID を指定してください。

```
#define USB_CFG_HCDC_IFCLS USB_CFG_CDC // CDC クラス対応デバイス
#define USB_CFG_HCDC_IFCLS USB_CFG_VEN // Vendor クラスデバイス
```

[Note]

市販の USB-シリアル変換デバイスの場合、デバイスクラス ID が Vendor クラスになっている場合があります。ご使用前に CDC デバイスの仕様をご確認ください。Vendor クラスの場合は、USB_CFG_VEN を指定してください。

3. 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、同じパイプ番号は指定しないでください。

```
#define USB_CFG_HCDC_BULK_IN     パイプ番号 (USB_PIPE1 から USB_PIPE5)
#define USB_CFG_HCDC_BULK_OUT    パイプ番号 (USB_PIPE1 から USB_PIPE5)
#define USB_CFG_HCDC_BULK_IN2    パイプ番号 (USB_PIPE1 から USB_PIPE5)
#define USB_CFG_HCDC_BULK_OUT2   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。同じパイプ番号は指定しないでください。なお、USB Hub を使用する場合は、下記の定義に対し PIPE9 を指定することはできません。

```
#define USB_CFG_HCDC_INT_IN     パイプ番号 (USB_PIPE6 から USB_PIPE9)
#define USB_CFG_HCDC_INT_IN2    パイプ番号 (USB_PIPE6 から USB_PIPE9)
```

[Note]

- 上記1の USB_CFG_HCDC_MULTI 定義に対し USB_CFG_ENABLE を指定した場合にのみ、USB_CFG_HCDC_BULK_IN2, USB_CFG_HCDC_BULK_OUT2 および USB_CFG_HCDC_INT_IN2 定義に対するパイプ番号を指定してください。
- USB_CFG_HCDC_MULTI 定義に対し USB_CFG_DISABLE を指定した場合には、USB_CFG_HCDC_BULK_IN2, USB_CFG_HCDC_BULK_OUT2 および USB_CFG_HCDC_INT_IN2 定義に対しては、ともに USB_NULL を指定してください。

9. サンプルアプリケーション

9.1 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. CDC デバイスに対し、受信要求 (Bulk In 転送) を行い、受信データを取得する。
2. Bulk Out 転送により受信データを CDC デバイスへ送信する (ループバック)。
3. 通信速度等の設定は、クラスリクエスト SET_LINE_CODING を CDC デバイスに送信することにより行います。このクラスリクエストにより通信速度、データビット数、ストップビット長、パリティビットの設定を行えます。

9.1.1 データ転送イメージ

データ転送イメージを以下に示します。

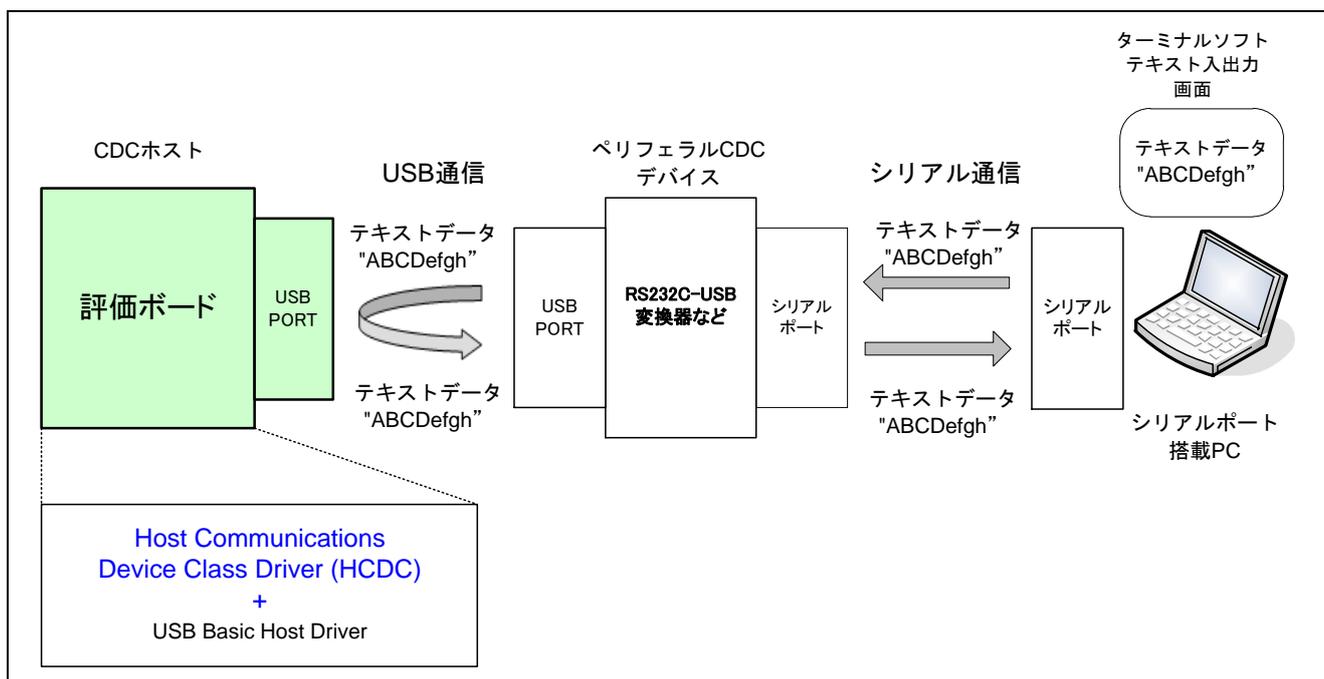


Figure 9-1 データ転送 (ループバック通信) イメージ

9.2 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。

9.2.1 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。

9.2.2 メインループ

このメインループでは、CDC デバイスから受信したデータをそのまま CDC デバイス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. RSK に CDC デバイスが ATTACH され、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 を確認するとクラスリクエスト SET_LINECODING を CDC デバイスに送信します。
2. クラスリクエスト処理を完了を確認すると R_USB_Read 関数をコールし、CDC デバイスから送信されるデータのデータ受信要求を行います。なお、データ受信要求のほか、CDC デバイスからの Class Notification 受信要求も行います。
3. CDC デバイス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。受信したデータは外部変数 g_data に格納されています。受信データサイズは、usb_ctrl_t 構造体のメンバ size により確認できます。APL では、メンバ size が 0(ゼロ)の場合、Null パケット受信と判断し、CDC デバイスに対し、再度データ受信要求を行います。メンバ size が 0(ゼロ)以外の場合、CDC デバイスからの送信データを受信したと判断します。受信したデータは CDC デバイスに対し、データ送信要求が行われます。
4. CDC デバイス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、CDC デバイス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。

以下に、APL の処理概要を示します。

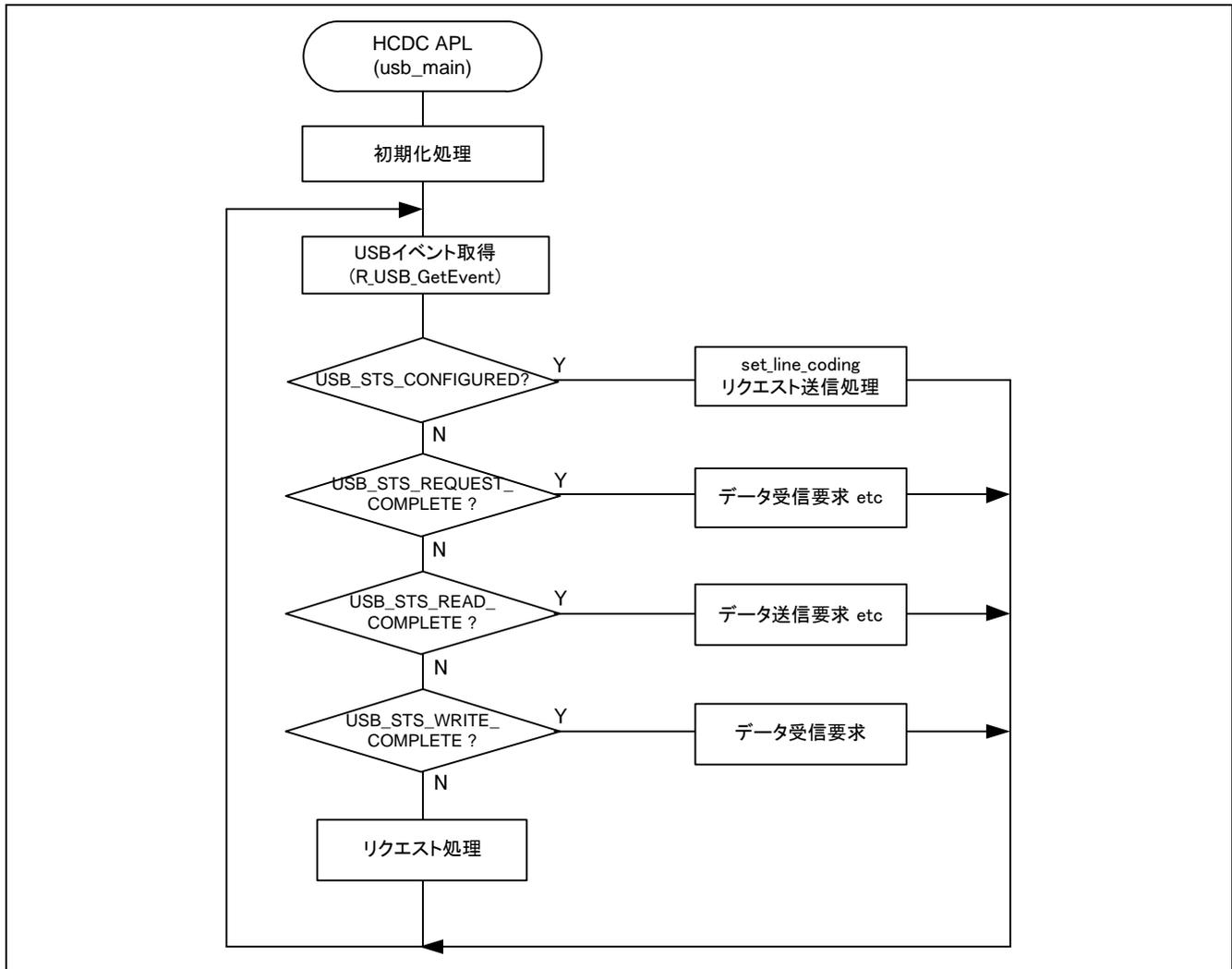


Figure 9-2 メインループ処理

9.3 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_hcdc_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
#define USE_USBIP (USE_USBIP0|USE_USBIP1) // USB0 と USB1 モジュールを使用する場合
  
```

2. COM_SPEED 定義

ボーレート値を指定してください。このボーレート値はクラスリクエスト SET_LINE_CODING によって CDC デバイスに設定されます。COM_SPEED 定義に対して、BPS_9600 / BPS_14400 / BPS_19200 / BPS_38400 / BPS_57600 / BPS_115200 のいずれかを指定してください。

```

#define COM_SPEED BPS_57600 // ボーレート値
  
```

3. COM_PARITY_BIT 定義

パリティビットを指定してください。このパリティビットはクラスリクエスト SET_LINE_CODING によって CDC デバイスに設定されます。COM_PARITY_BIT 定義に対して、PARITY_EVEN / PARITY_ODD / PARITY_NONE のいずれかを指定してください。

```
#define COM_PARITY_BIT PARITY_NONE // パリティビット
```

4. COM_STOP_BIT 定義

ストップビットを指定してください。このストップビットはクラスリクエスト SET_LINE_CODING によって CDC デバイスに設定されます。COM_STOP_BIT 定義に対して、STOP_BIT1(1 ビット) / STOP_BIT2 (2 ビット)のいずれかを指定してください。

```
#define COM_STOP_BIT STOP_BIT1 // ストップビット
```

5. COM_DATA_BIT 定義

データビットを指定してください。このデータビットはクラスリクエスト SET_LINE_CODING によって CDC デバイスに設定されます。COM_DATA_BIT 定義に対して、DATA_BIT7 (7 ビット) / DATA_BIT8 (8 ビット)のいずれかを指定してください。

```
#define COM_DATA_BIT DATA_BIT8 // データビット
```

6. 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。

9.4 複数の CDC デバイスを接続する場合

USB Hub 等を使用し、複数の CDC デバイスを接続するアプリケーションプログラムを開発する場合は、以下の参考プログラムを参照してください。

```
r_usb_hcdc_apl_multi.c
```

10. セットアップ

10.1 ハードウェア

10.1.1 動作環境例

HCDC の動作環境例を以下に示します。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ください。

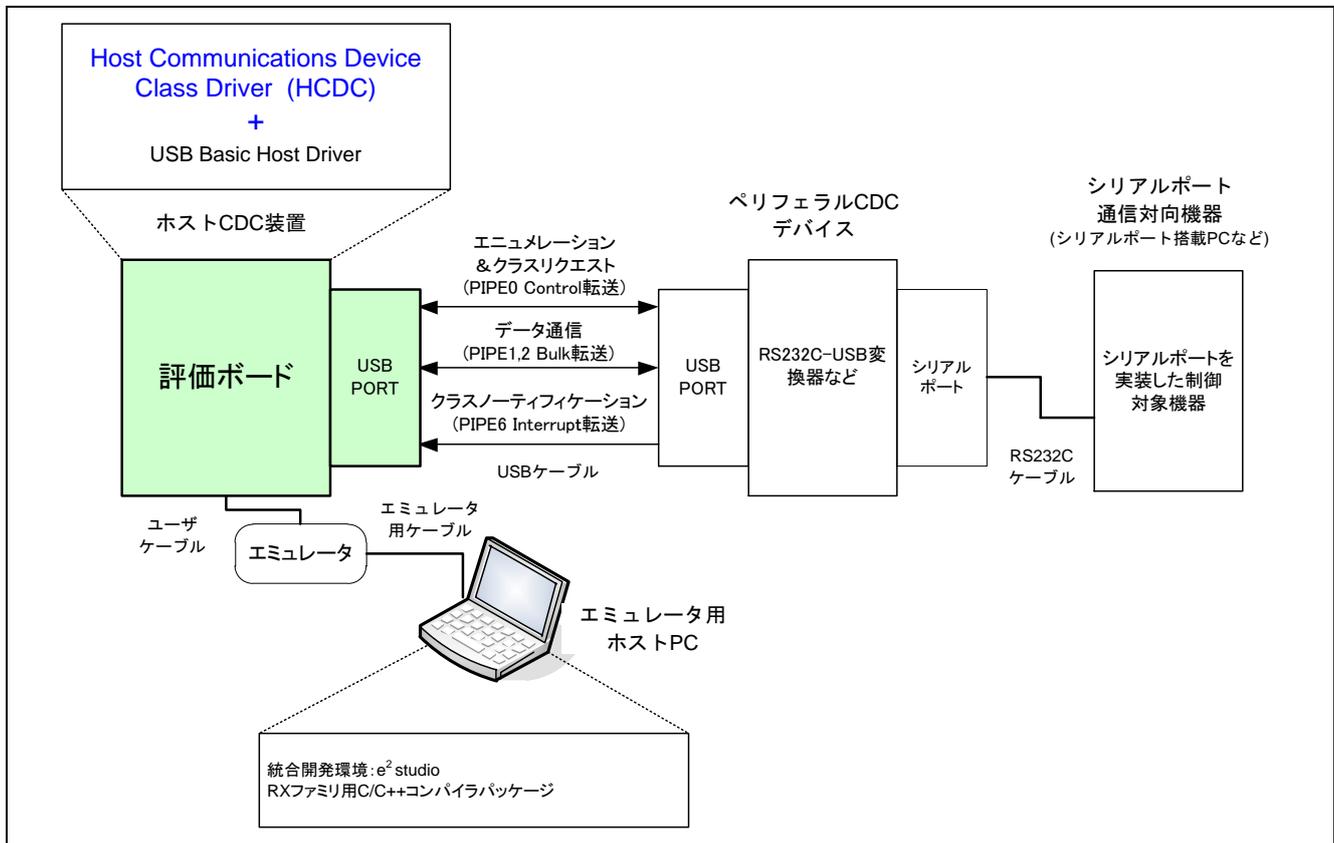


Figure 10-1 動作環境例

10.1.2 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。

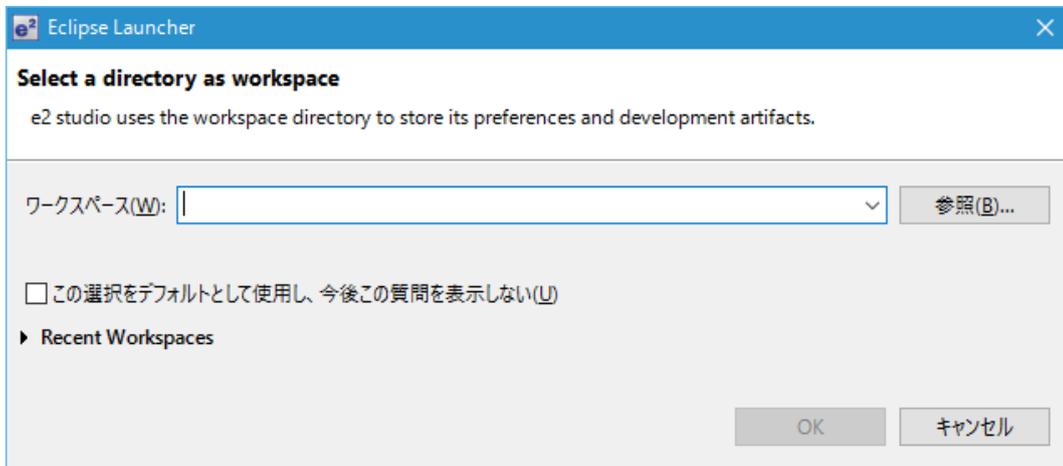
Table 10-1 RSK 設定

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

10.2 ソフトウェア

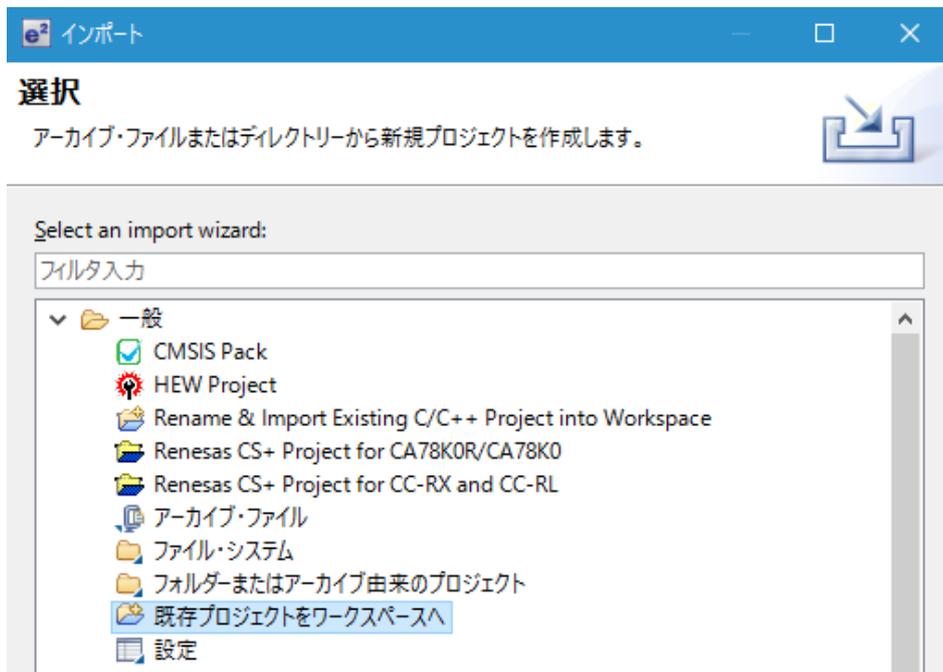
(1). e² studio を起動

- 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。

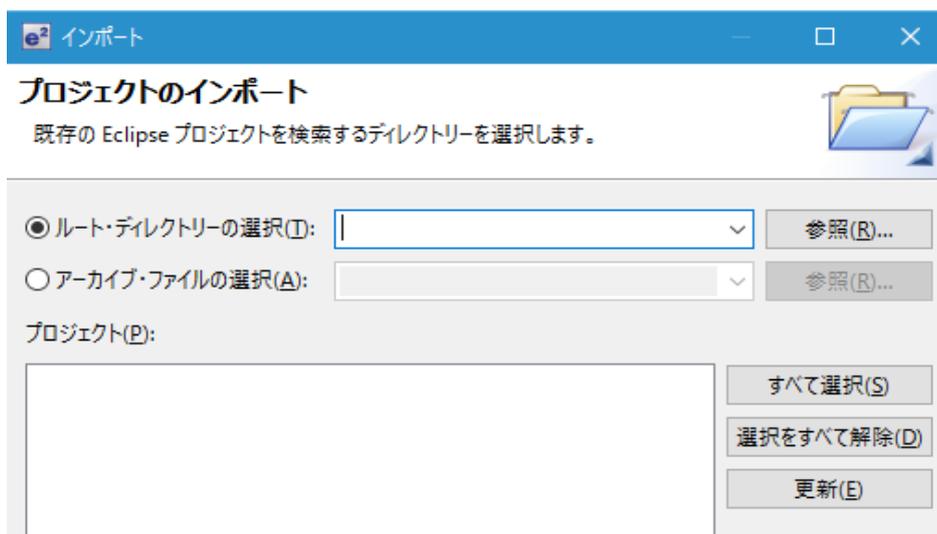


(2). ワークスペースへのプロジェクトの登録

- 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。

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.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。

12. e² studio 用プロジェクトを CS+で使用する場合

HCDC のプロジェクトは、統合環境 e² studio で作成されています。HCDC 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。

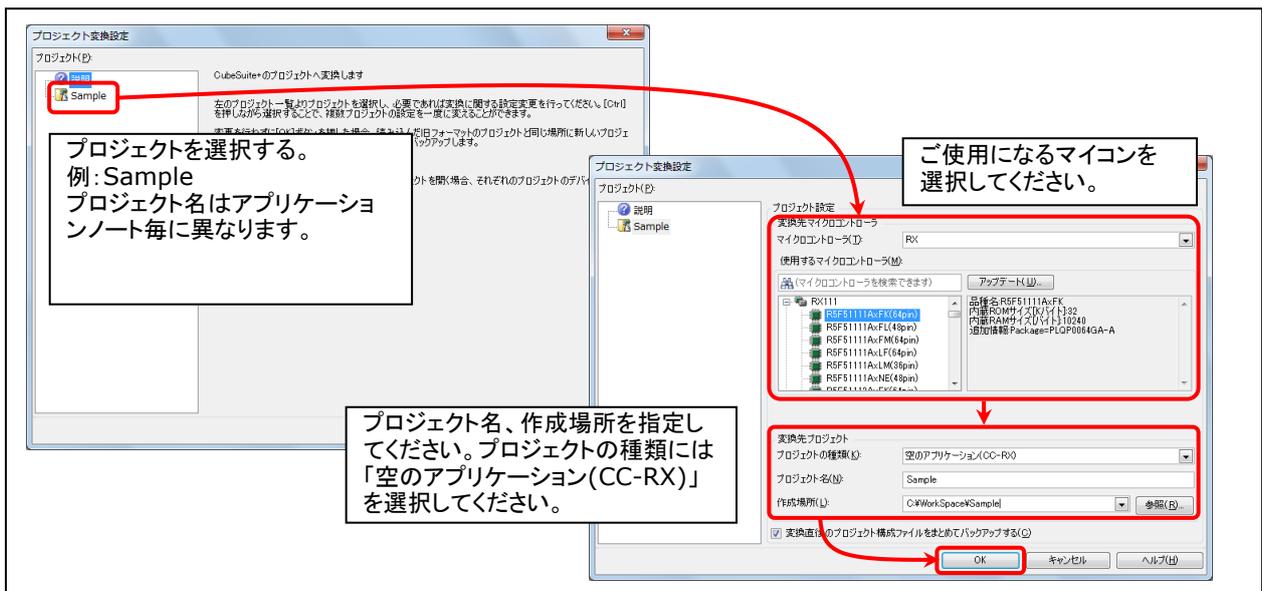
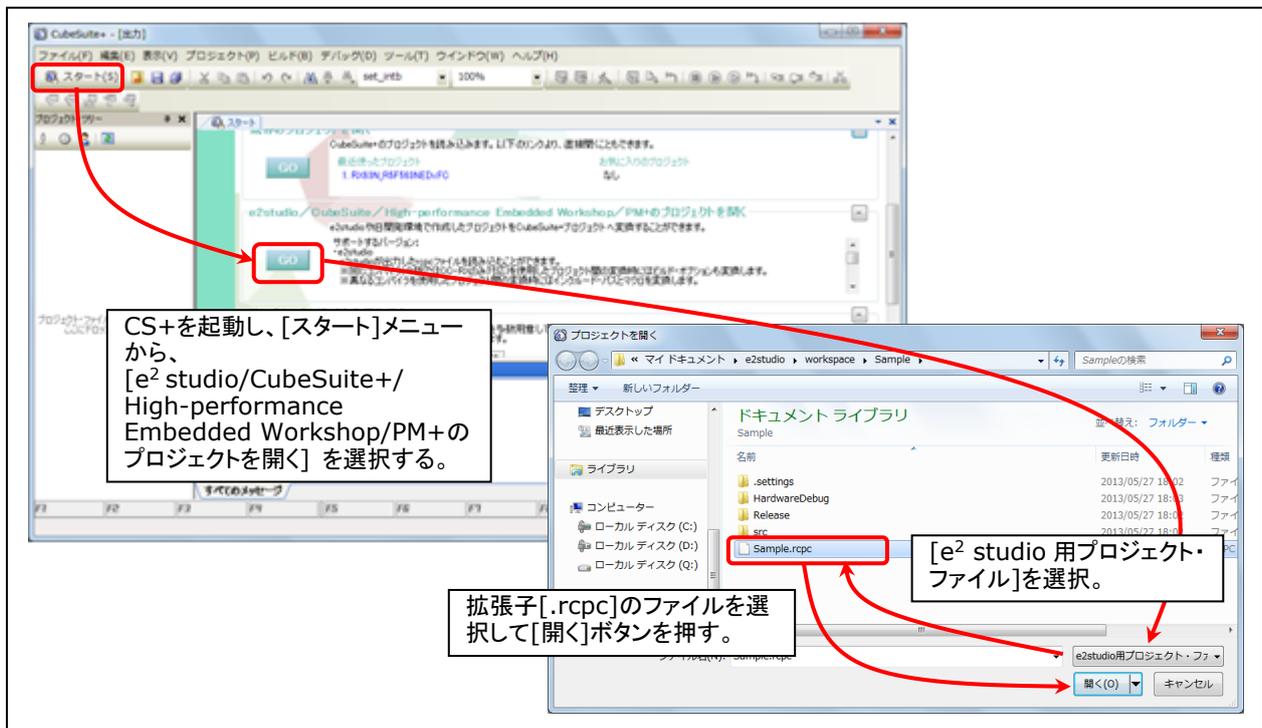


Figure 12-1 e² studio 用プロジェクトの CS+読み込み方法

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	2011.09.01	—	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 USB)に関する内容を追加
		33	5 ホスト用CDCサンプルアプリケーションプログラム (APL) のアプリケーション仕様変更 ・ SW2押下時通信速度設定変更 ・ SW3押下時通信速度設定
2.00	2012.3.28	—	V.2.00 用 First Release
2.01	2012.2.1	—	“1.4 本書の読み方”追加、ケアレスミスを修正
2.10	2013.4.1	—	V.2.10 用 First Release 動作確認デバイスに RX63T を追加。この追加に伴い RX63T 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 CDC デバイスの複数接続をサポート
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 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/>