

RXファミリ

R01AN2027JJ0130

Rev.1.30

Mar 1, 2020

USB Host Communications Devices Class Driver (HCD)C) Firmware Integration Technology

要旨

本アプリケーションノートでは、Firmware Integration Technology (FIT) を使用した、USB Host コミュニケーションデバイスクラスドライバ (HCD)C) について説明します。本モジュールは USB Basic Host and Peripheral Driver (USB-BASIC-FW FIT モジュール) と組み合わせることで動作します。以降、本モジュールを USB HCD FIT モジュールと称します。

対象デバイス

RX65N/RX651 グループ
RX64M グループ
RX71M グループ
RX66T グループ
RX72T グループ
RX72M グループ
RX66N グループ
RX72N 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. RX64M 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. R01UH0377)
 5. RX71M 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. R01UH0493)
 6. RX65N/RX651 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. R01UH0590)
 7. RX65N/RX651-2M 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. R01UH0659)
 8. RX66T 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. R01UH0749)
 9. RX72T 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. R01UH0803)
 10. RX72M 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. R01UH0804)
 11. RX66N 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. R01UH0825)
 12. RX72N 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. R01UH0824)
 13. 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)
- ・ 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)	9
5. コミュニケーションデバイスクラス (CDC) ,PSTN and ACM.....	10
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 (HCD)	16
7. API.....	19
8. コンフィグレーション (r_usb_hcd_config.h)	20
9. アプリケーションの作成方法.....	21

1. 概要

USB HCDC FIT モジュールは、USB-BASIC-FW FIT モジュール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 モジュールに対し 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 Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)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

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。

1.3 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。

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 for USB-BASIC-FW
HCDC	: Host Communication Devices Class
HDCC	: Host Device Class Driver (Device Driver and USB Class Driver)
HUBCD	: Hub Class Driver
MGR	: Peripheral Device State Manager for HCD
Non-OS	: USB Driver for OS-less
RSK	: Renesas Starter Kits
RTOS	: USB Driver for the real-time OS
USB-BASIC-FW	: USB Basic Host and Peripheral Driver

1.5 USB HCD FIT モジュール

本モジュールは、`r_usb_basic` を使用したプロジェクトに組み込む必要があります。プロジェクトに組み込み後、API を使用することで USB の H/W 制御を行います。

2. ソフトウェア構成

Figure 2-1に HCD 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。

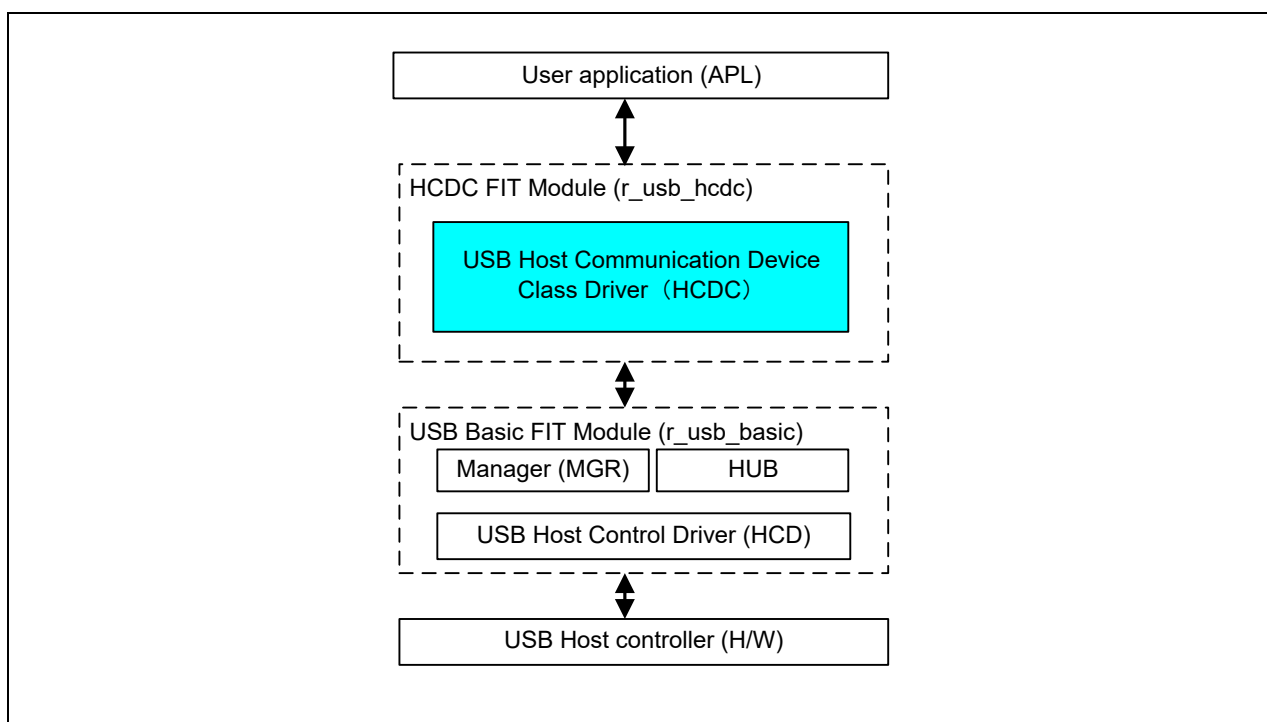


Figure 2-1 モジュール構成図

Table 2-1 モジュール説明

モジュール名	説明
HCD	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 HCD 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 ソフトウェアの要求

このドライバは以下のパッケージに依存しています。

- r_bsp
- r_usb_basic

3.3 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 (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 4.08.04.201902 (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 version 4.12.01
リアルタイム OS	FreeRTOS V.10.0.0 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
モジュールのバージョン	Rev.1.30
使用ボード	Renesas Starter Kits for RX64M Renesas Starter Kits for RX71M Renesas Starter Kits for RX65N, Renesas Starter Kits for RX65N-2MB Renesas Starter Kits for RX72T Renesas Starter Kits for RX72M Renesas Starter Kits for RX72N

3.4 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
RX64M RX71M	USBIO 割り込み(ベクタ番号: 189, 割り込み要因番号 : 62, 選択型割り込み B) USB D0FIFO0 割り込み(ベクタ番号: 34) / USB D1FIFO0 割り込み(ベクタ番号: 35) USBR0 割り込み(ベクタ番号:90)
	USBAR 割り込み(ベクタ番号: 94) USB D0FIFO2 割り込み(ベクタ番号: 32) / USB D1FIFO2 割り込み(ベクタ番号: 33)

RX65N RX651 RX72M RX72N RX66N	USBIO 割り込み(ベクタ番号: 185, 割り込み要因番号 : 62, 選択型割り込み B) USB D0FIFO0 割り込み(ベクタ番号: 34) / USB D1FIFO0 割り込み(ベクタ番号: 35) USBRO 割り込み(ベクタ番号:90)
RX66T RX72T	USBIO 割り込み(ベクタ番号: 174) / USBRO 割り込み(ベクタ番号: 90) USB D0FIFO0 割り込み(ベクタ番号: 34) / USB D1FIFO0 割り込み(ベクタ番号: 35)

3.5 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。

3.6 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。

3.7 コンパイル時の設定

コンパイル時の設定については、「8. コンフィグレーション (`r_usb_hcdc_config.h`)」章および 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)の「コンフィグレーション」章を参照してください。

3.8 ROM / RAM サイズ

本ドライバの ROM/RAM サイズを以下に示します。

1. CC-RX (最適化レベル: Default)

(1). Non-OS

	引数チェック実施時	引数チェック非実施時
ROM サイズ	37.5K バイト (Note 3)	37.0K バイト (Note 4)
RAM サイズ	9.8K バイト	9.8K バイト

(2). RTOS

a. FreeRTOS

	引数チェック実施時	引数チェック非実施時
ROM サイズ	48.3K バイト (Note 3)	47.8K バイト (Note 4)
RAM サイズ	35.6K バイト	35.6K バイト

b. RI600V4

	引数チェック実施時	引数チェック非実施時
ROM サイズ	51.4K バイト (Note 3)	50.9K バイト (Note 4)
RAM サイズ	17.2K バイト	17.2K バイト

2. GCC (最適化レベル: -O2)

	引数チェック実施時	引数チェック非実施時
ROM サイズ	42.0K バイト (Note 3)	41.5K バイト (Note 4)
RAM サイズ	8.4K バイト	8.4K バイト

3. IAR (最適化レベル: Medium)

	引数チェック実施時	引数チェック非実施時
ROM サイズ	37.9K バイト (Note 3)	37.3K バイト (Note 4)
RAM サイズ	11.8K バイト	11.8K バイト

[Note]

1. 上記のサイズには、BSP および USB Basic Driver の ROM/RAM サイズが含まれています。
2. 上記は V2 コアオプション指定時のサイズです。
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。
5. RTOS には、リアルタイム OS の ROM/RAM サイズが含まれています。

3.9 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 Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)内の「構造体」の章を参照してください。

3.10 FIT モジュールの追加方法

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、Smart Configurator を使用した(1)、(3)の追加方法を推奨しています。ただし、Smart Configurator は、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(2)、(4)の方法を使用してください。

- (1) e² studio 上で Smart Configurator を使用して FIT モジュールを追加する場合

e² studio の Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e² studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。

- (2) e² studio 上で FIT Configurator を使用して FIT モジュールを追加する場合

e² studio の FIT Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。

- (3) CS+上で Smart Configurator を使用して FIT モジュールを追加する場合

CS+上で、スタンドアロン版 Smart Configurator を使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「Renesas e² studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)」を参照してください。

- (4) CS+上で FIT モジュールを追加する場合

CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」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 Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)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 Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)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Table5-2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Table5-3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Table5-4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Table5-5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 <small>注</small>	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、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Table5-8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 <small>注</small>	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Table5-9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Table5-10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Table5-11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Table5-12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Table5-14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Table5-15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Table5-16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Table5-17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	ブレイク信号検出
D1	bTxCarrier	Data Set Ready：回線が接続されて通信可能
D0	bRxCarrier	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Table5-18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_AVAILABLE(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 (HCD)

6.1 基本機能

HCD 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。
HCD 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_hcd_if.h` ファイル内に定義されています。

6.2.1 HCD 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Table6-1に記します。

Table6-1 `usb_hcd_linecoding_t` 構造体

型	メンバ名	説明	備考
<code>uint32_t</code>	<code>dwdte_rate</code>	回線速度	単位 : bps
<code>uint8_t</code>	<code>bchar_format</code>	ストップビット設定	
<code>uint8_t</code>	<code>bparity_type</code>	パリティ設定	
<code>uint8_t</code>	<code>bdata_bits</code>	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Table6-2に示します。

Table6-2 `usb_hcd_controllinestate_t` Structure

型	メンバ名	説明	備考
<code>uint16_t</code> (D1)	<code>brts:1</code>	Carrier control for half duplex modems 0 - Deactivate carrier, 1 - Activate carrier	
<code>uint16_t</code> (D0)	<code>bdtr:1</code>	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Table6-3に示します。

Table6-3 `usb_hcd_encapsulated_t` Structure

型	メンバ名	説明	備考
<code>uint8_t</code>	<code>*p_data</code>	AT コマンドデータが格納されている領域	
<code>uint16_t</code>	<code>wlength</code>	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Table6-4に示します。

Table6-4 `usb_hcd_breakduration_t` Structure

型	メンバ名	説明	備考
<code>uint16_t</code>	<code>wtime_ms</code>	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 ノーティフィケーションフォーマット

Table6-8とTable6-9に CDC ノーティフィ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ティフィケーション“SerialState”の UART State Bitmap 構造体をTable6-10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 Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)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をお願いします。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. 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]

- a. 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。
- b. 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. アプリケーションの作成方法

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「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。

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	Apr 1, 2014	—	初版発行
1.10	Dec 26, 2014	—	1. 対象デバイスに RX71M を追加。 2. CDC デバイスの複数デバイス接続をサポート。 3. 複数デバイス接続を設定するためのマクロ定義を r_usb_hcdc_config.h 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	—	1. Technical Update(発行番号: TN-RX*-A172A/J)に対応しました。 2. 「API」の章を USB Basic Host and Peripheral Driver Firmware Integration Technology (ドキュメント番号:R01AN2025)のドキュメントへ移しました。
1.22	Sep 30, 2017	—	RX65N/RX651-2M をサポート
1.23	Mar 31, 2018	—	Smart Configurator に対応しました。
1.24	Dec 28, 2018	—	RTOS をサポート
1.25	Apr 16, 2019	—	対象デバイスに RX66T/RX72T を追加
1.26	May 31, 2019	—	1. GCC/IAR コンパイラをサポートしました。 2. 対象デバイスから RX63N を削除しました。
1.27	Jul 31, 2019	—	対象デバイスに RX72M を追加。
1.30	Mar 1, 2020	—	1. リアルタイム OS(uITRON:RI600V4)をサポートしました。 2. 対象デバイスに RX72N/RX66N 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うと、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まれます。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/>