

## USB Host Human Interface Device Class Driver (HHID) 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、Host 用ヒューマンインターフェースデバイス (HHID) クラスドライバについて説明します。本モジュールは USB Basic Firmware (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 HHID 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  
RX671 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 Human Interface Devices Version 1.1
3. HID Usage Tables Version 1.1  
【<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. RX671 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. R01UH0899)
14. 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.....	4
3. API情報 .....	5
4. ターゲットペリフェラルリスト (TPL) .....	9
5. ヒューマンインターフェースデバイスクラス (HID) .....	10
6. API.....	13
7. コンフィグレーション (r_usb_hhid_config.h) .....	18
8. コンフィグレーションファイル (RI600V4使用時のみ) .....	19
9. アプリケーションの作成方法 .....	20

## 1. 概要

USB HHID 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 ヒューマンインターフェースデバイスクラスドライバ（以降 HHID と記述）として動作します。

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。

- ・ 接続されたHIDデバイス（USBマウス、USBキーボード）とデータ通信が可能
- ・ 接続されたHIDデバイスに対し、HIDクラスリクエストを発行する
- ・ Interrupt OUT転送をサポート
- ・ 一つのUSBモジュールに対しUSB Hubを使って最大3つのHID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

本モジュールには以下の制限事項があります。

1. HHID はレポートディスクリプタ解析を行っておりません。デバイスから取得したインターフェースプロトコル（Keyboard/Mouse）からレポートフォーマットを決定しています。
2. 本ドライバは、一つの USB モジュールに対し USB Hub を使って最大 3 つの HID デバイスの接続が可能です。なお、Interrupt OUT 転送を使用する場合、一つの USB モジュールに対し二つ以上の HID デバイスを接続することはできません。
3. 本ドライバは DMA/DTC 転送をサポートしていません。

### 1.4 用語一覧

APL	:	Application program
HCD	:	Host Control Driver for USB-BASIC-FW
HDCD	:	Host Device Class Driver (Device Driver and USB Class driver)
HHID	:	USB Host Human Interface Device Class Driver
HID	:	Human Interface Device Class
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	:	Universal Serial Bus
USB-BASIC-FW	:	USB Basic Host and Peripheral Driver

### 1.5 USB HHID 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

USB HHID FIT モジュールは HID クラスドライバと、マウス、キーボードのデバイスドライバから構成されます。 接続された USB デバイスからデータを受け取ると、HCD を介して APL に通知します。又、APL から要求があった場合、HCD を介して USB デバイスに通知します。

Figure 2-1に、HHID 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。

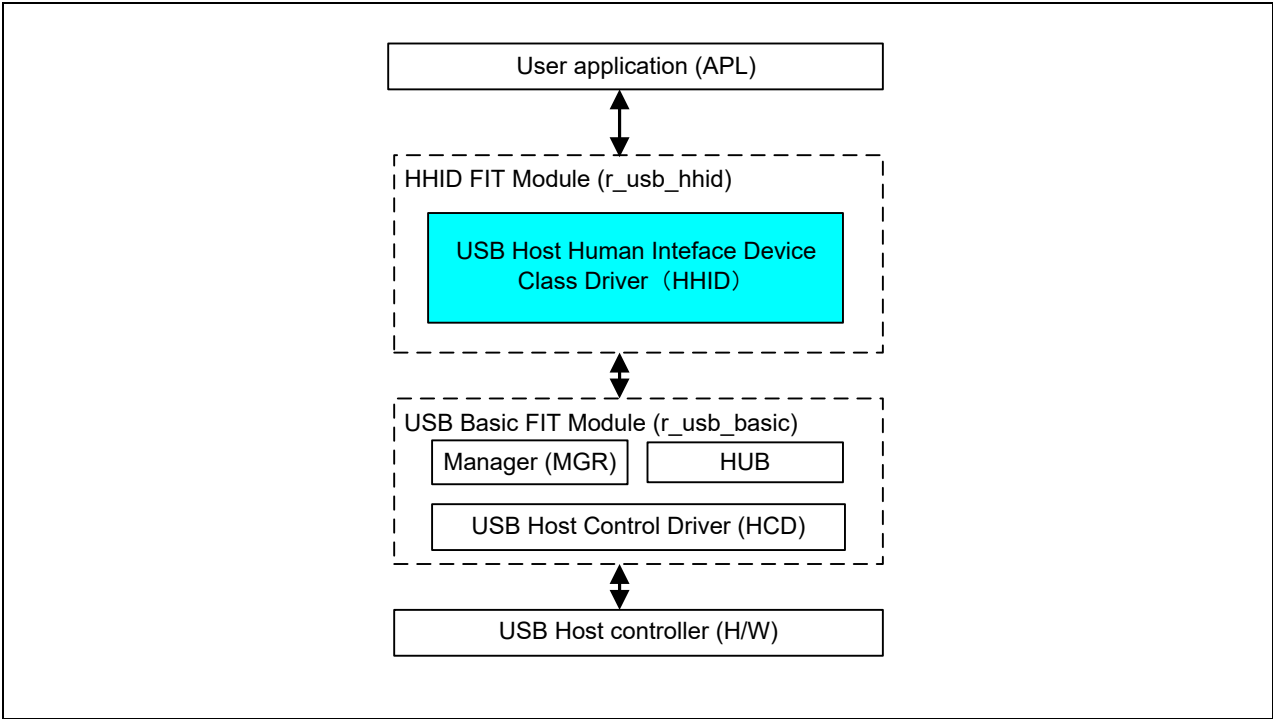


Figure 2-1 モジュール構成図

Table 2-1 モジュール機能概要

モジュール名	機能概要
HHID	USB デバイスからの要求を解析します。 HCD を介して、APL の SW 操作情報を USB デバイスに通知します。
HCD/MGR	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.3.03.00 (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.1.42
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 Renesas Starter Kits for RX671

#### 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)
	USBRO 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) USB R0 割り込み(ベクタ番号: 90)
RX66T RX72T	USBIO 割り込み(ベクタ番号: 174) / USB R0 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)
RX671	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) USB R0 割り込み(ベクタ番号: 90) ----- USB I1 割り込み(ベクタ番号: 182, 割り込み要因番号: 63, 選択型割り込み B) USB D0FIFO1 割り込み(ベクタ番号: 36) / USB D1FIFO1 割り込み(ベクタ番号: 37)

### 3.5 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_hhid_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、「7. コンフィグレーション (`r_usb_hhid_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	38.6K バイト (Note 3)	38.1K バイト (Note 4)
RAM サイズ	14.4K バイト	14.4K 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	49.3K バイト (Note 3)	48.8K バイト (Note 4)
RAM サイズ	35.9K バイト	35.9K 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	51.3K バイト (Note 3)	50.8K バイト (Note 4)
RAM サイズ	18.0K バイト	18.0K 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	44.8K バイト (Note 3)	44.2K バイト (Note 4)
RAM サイズ	14.2K バイト	14.2K 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	38.7K バイト (Note 3)	38.1K バイト (Note 4)
RAM サイズ	12.9K バイト	12.9K 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 for 文、while 文、do while 文について

FIT モジュールでは、レジスタの反映待ち処理等で for 文、while 文、do while 文（ループ処理）を使用しています。これらループ処理には、「WAIT\_LOOP」をキーワードとしたコメントを記述しています。そのため、ループ処理にユーザがフェイルセーフの処理を組み込む場合は、「WAIT\_LOOP」で該当の処理を検索できます。

### 3.11 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) e<sup>2</sup> studio 上で Smart Configurator を使用して FIT モジュールを追加する場合

e<sup>2</sup> 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 e<sup>2</sup> 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) e<sup>2</sup> studio 上で FIT Configurator を使用して FIT モジュールを追加する場合

e<sup>2</sup> 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 e<sup>2</sup> 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 e<sup>2</sup> 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. ヒューマンインターフェースデバイスクラス (HID)

### 5.1 基本機能

本ドライバは、ヒューマンインターフェースデバイスクラス仕様に準拠しています。  
本ドライバの主な機能を以下に示します。

1. HID デバイスの照合
2. HID デバイスへのクラスリクエスト通知
3. HID デバイスとのデータ通信

### 5.2 クラスリクエスト

本ドライバがサポートしているクラスリクエストをTable 5-1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。

Table 5-1 HID クラスリクエスト

対応 記号表	リクエスト	コード	説明
a	USB_GET_REPORT	0x01	USB デバイスにレポートを要求する
b	USB_SET_REPORT	0x09	USB デバイスにレポートを通知する
c	USB_GET_IDLE	0x02	USB デバイスに Duration 時間を要求する
d	USB_SET_IDLE	0x0A	USB デバイスに Duration 時間を通知する
e	USB_GET_PROTOCOL	0x03	USB デバイスにプロトコルを要求する
f	USB_SET_PROTOCOL	0x0B	USB デバイスにプロトコルを通知する
	USB_GET_REPORT_DESCRIPTOR	Standard	レポートディスクリプタを要求する
	USB_GET_HID_DESCRIPTOR	Standard	HID ディスクリプタを要求する

本ドライバ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。

#### a). GetReport リクエストフォーマット

Table 5-2に、GetReport リクエストのフォーマットを以下に示します。  
コントロール転送によりデバイスからレポートデータを受信します。

Table 5-2 GetReport フォーマット

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

#### b). SetReport リクエストフォーマット

Table 5-3に、SetReport リクエストのフォーマットを示します。  
コントロール転送によりレポートデータをデバイスに送信します。

Table 5-3 SetReport フォーマット

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

**c). GetIdle リクエストフォーマット**

Table 5-4に、GetIdle リクエストのフォーマットを示します。

レポート通知（インタラプト転送）の間隔時間を取得します。Idle rate は 4msec 単位です。

**Table 5-4 GetIdle フォーマット**

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

**d). SetIdle リクエストフォーマット**

Table 5-5に、SetIdle リクエストのフォーマットを示します。

レポート通知（インタラプト転送）の間隔時間を設定します。Duration は 4msec 単位です。

**Table 5-5 SetIdle フォーマット**

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

**e). GetProtocol リクエストフォーマット**

Table 5-6に、GetProtocol リクエストのフォーマットを示します。

現在設定されているプロトコル（ブートプロトコル又はレポートプロトコル）を取得します。

**Table 5-6 GetProtocol フォーマット**

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_PROTOCOL (0x03)	0	Interface	1	0(BootProtocol) / 1(ReportProtocol)

**f). SetProtocol リクエストフォーマット**

Table 5-7に、SetProtocol リクエストのフォーマットを示します。

プロトコル（ブートプロトコル又はレポートプロトコル）の設定を行います。

**Table 5-7 SetProtocol フォーマット**

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_PROTOCOL (0x0B)	0(BootProtocol) / 1(ReportProtocol)	Interface	0	Not applicable

## 5.3 レポートフォーマット

HID で扱うレポートフォーマットを以下に記します。

### 5.3.1 受信レポートフォーマット

Table 5-8に、HID デバイスから通知される受信レポートフォーマットを示します。  
インタラプト IN 転送及び、クラスリクエスト GetReport により受信します。

**Table 5-8 受信レポートフォーマット**

offset (データ長)	Keyboard モード (8 バイト)	Mouse モード (3 バイト)
0(Top Byte)	Modifier keys	b0 : Button 1 b1 : Button 2 b2-7 : Reserved
+1	Reserved	X displacement
+2	Keycode 1	Y displacement
+3	Keycode 2	—
+4	Keycode 3	—
+5	Keycode 4	—
+6	Keycode 5	—
+7	Keycode 6	—

### 5.3.2 送信レポートフォーマット

Table 5-9に、HID デバイスに通知する送信レポートフォーマットを示します。  
クラスリクエスト SetReport で送信を行います。

**Table 5-9 送信レポートフォーマット**

offset (データ長)	Keyboard モード (1 バイト)	Mouse モード (非サポート)
0(Top Byte)	b0 : LED 0(NumLock) b1 : LED 1(CapsLock) b2 : LED 2(ScrollLock) b3 : LED 3(Compose) b4 : LED 4(Kana)	—
+1~+16	—	—

### 5.3.3 注意事項

データ通信で用いるレポートフォーマットはレポートディスクリプタに従う必要があります。本ドライバではレポートディスクリプタの取得と解析は行わず、インターフェースプロトコルコードに従ってレポートフォーマットを決定しています。

## 6. API

Host Human Interface Device Class 固有の API を以下に示します。

API	説明
R_USB_HhidGetType()	接続された HID デバイスのタイプ情報取得
R_USB_HhidGetMxps()	接続された HID デバイスのマックスパケットサイズを取得

### [Note]

その他の 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。

6.1 R\_USB\_HhidGetType

接続された HID デバイスのタイプ情報を取得する

形式

usb\_err\_t R\_USB\_HhidGetType(usb\_ctrl\_t \*p\_ctrl, uint8\_t \*p\_type)

引数

p\_ctrl usb\_ctrl\_t 構造体領域へのポインタ  
p\_type タイプ情報を格納する領域へのポインタ

戻り値

USB\_SUCCESS 正常終了 (タイプ情報取得完了)  
USB\_ERR\_PARA パラメータエラー  
USB\_ERR\_NG その他のエラー

解説

usb\_ctrl\_t 構造体に指定された情報(メンバ module およびメンバ address)をもとに、接続された HID デバイスのタイプ情報(マウス、キーボード、その他)を取得します。タイプ情報は、第 2 引数(p\_type) が示す領域にセットされます。セットされるタイプ情報については、Table 6-1を参照してください。

Table 6-1 タイプ情報

タイプ情報	内容
USB_HID_KEYBOARD	キーボード
USB_HID_MOUSE	マウス
USB_HID_OTHER	キーボード、マウス以外の HID デバイス

補足

1. 本 API をコールする前に usb\_ctrl\_t 構造体のメンバ address およびメンバ module に対し、タイプ情報を取得したい HID デバイスのデバイスアドレスおよびその HID デバイスが接続された USB モジュール番号(USB\_IP0/USB\_IP1)を指定してください。なお、これらのメンバに対する指定に問題がある場合は、戻り値に USB\_ERR\_PARA が返されます。
2. ご使用の MCU が USB モジュールを 1 つしかサポートしていない場合、メンバ module に対し USB\_IP1 を指定しないでください。USB\_IP1 を指定した場合は、戻り値に USB\_ERR\_PARA が返されます。
3. 引数 p\_ctrl または引数 p\_type に対し USB\_NULL を指定した場合は、戻り値に USB\_ERR\_PARA が返されます。
4. USB デバイスが CONFIGURED 状態の場合に、本 API をコールすることができます。CONFIGURED 以外の状態で本 API をコールすると戻り値に USB\_ERR\_NG が返されます。

## 使用例

```
void usb_application( void )
{
    usb_ctrl_t ctrl;
    uint8_t type;

    :
    while (1)
    {
        switch (R_USB_GetEvent(&ctrl))
        {
            :
            case USB_STS_CONFIGURED:
                :
                ctrl.module = USB_IP0;
                ctrl.address = adr;
                R_USB_HhidGetType( &ctrl, &type );
                if( USB_HID_KEYBOARD == type )
                {
                    :
                }
                :
                break;
                :
            :
        }
    }
}
```

## 6.2 R\_USB\_HhidGetMxps

接続された HID デバイスのマックスパケットサイズを取得する

形式

usb\_err\_t R\_USB\_HhidGetMxps(usb\_ctrl\_t \*p\_ctrl, uint16\_t \*p\_mxps, uint8\_t dir)

引数

p_ctrl	usb_ctrl_t 構造体領域へのポインタ
p_mxps	MaxPacketSize を格納する領域へのポインタ
dir	転送方向 (USB_IN / USB_OUT)

戻り値

USB_SUCCESS	正常終了 (マックスパケットサイズ情報取得完了)
USB_ERR_PARA	パラメータエラー
USB_ERR_NG	その他のエラー

解説

usb\_ctrl\_t 構造体に指定された情報(メンバ module およびメンバ address)をもとに、接続された HID デバイスのマックスパケットサイズを取得します。マックスパケットサイズは、第 2 引数(p\_mxps)が示す領域にセットされます。

第 3 引数(dir)には、取得したい Max Pacet Size の転送方向(USB\_IN / USB\_OUT)を指定してください。

補足

1. 本 API をコールする前に usb\_ctrl\_t 構造体のメンバ address およびメンバ module に対し、タイプ情報を取得したい HID デバイスのデバイスアドレスおよびその HID デバイスが接続された USB モジュール番号(USB\_IP0/USB\_IP1)を指定してください。なお、これらのメンバに対する指定に問題がある場合は、戻り値に USB\_ERR\_PARA が返されます。
2. ご使用の MCU が USB モジュールを 1 つしかサポートしていない場合、メンバ module に対し USB\_IP1 を指定しないでください。USB\_IP1 を指定した場合は、戻り値に USB\_ERR\_PARA が返されます。
3. 引数 p\_mxps に対し USB\_NULL を指定した場合は、戻り値に USB\_ERR\_PARA が返されます。
4. 接続した HID デバイスが第 3 引数(dir)に指定した転送方向をサポートしていない場合、本 API は戻り値に USB\_ERR\_NG を返します。
5. USB デバイスが CONFIGURED 状態の場合に、本 API をコールすることができます。CONFIGURED 以外の状態で本 API をコールすると戻り値に USB\_ERR\_NG が返されます。



## 使用例

```
void usb_application( void )
{
    uint16_t  mxps;
    usb_ctrl_t ctrl;

    while (1)
    {
        switch (R_USB_GetEvent(&ctrl))
        {
            :
            case USB_STS_CONFIGURED:
                :
                ctrl.module = USB_IP0;
                ctrl.address = adr;
                R_USB_HhidGetMxps(&ctrl, &mxps, USB_IN);
                :
            break;
            :
        }
    }
}
```

## 7. コンフィグレーション (r\_usb\_hhid\_config.h)

お客様のシステムにあわせて以下の設定をお願いします。

### [Note]

必ず r\_usb\_basic\_config.h ファイルに対する設定もお願いします。r\_usb\_basic\_config.h については、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)内の「コンフィグレーション」の章を参照してください。

### 1. 使用パイプ設定

Interrupt IN や Interrupt OUT 転送で使用するパイプ番号(PIPE6 から PIPE9)を指定してください。なお、同じパイプ番号は指定しないでください。USB Hub を使用する場合は、下記の定義に対し USB\_PIPE9 を指定することはできません。

#define	USB_CFG_HHID_INT_IN	パイプ番号 (USB_PIPE6 から USB_PIPE9)
#define	USB_CFG_HHID_INT_IN2	パイプ番号 (USB_PIPE6 から USB_PIPE9)
#define	USB_CFG_HHID_INT_IN3	パイプ番号 (USB_PIPE6 から USB_PIPE9)
#define	USB_CFG_HHID_INT_OUT	パイプ番号 (USB_PIPE6 から USB_PIPE9)

### [Note]

USB\_CFG\_HHID\_INT\_IN2、USB\_CFG\_HHID\_INT\_IN3 および USB\_CFG\_HHID\_INT\_OUT 定義に対しパイプ番号を設定する必要が無い場合は、これらの定義に対し USB\_NULL を指定してください。

## 8. コンフィグレーションファイル (RI600V4 使用時のみ)

RI600V4 を使用する場合、HHID USB ドライバで使用する OS 資源を RI600V4 に登録する必要があります。以下の定義をコンフィグレーションファイルに追加してください。コンフィグレーションファイルの作成方法については、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「**RI600V4 (コンフィグレーションファイル作成)**」の章を参照してください。

### 8.1 メールボックス定義

name	:	ID_USB_RTOS_HHID_MBX
wait_queue	:	TA_FIFO
message_queue	:	TA_MFIFO

## 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/>
- お問い合わせ先  
<http://japan.renesas.com/inquiry>

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	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. HID デバイスの複数接続をサポート 3. 以下の API に対し引数 ipno を追加。 R_usb_hhid_GetReportLength, R_usb_hhid_get_hid_protocol
1.11	Sep 30, 2015	—	対象デバイスに RX63N と RX631 を追加。
1.20	Sep 30, 2016	—	1. 対象デバイスに RX65N/RX651 を追加 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
1.21	Mar 31, 2017	—	1. R_USB_GetEvent 関数の戻り値が USB_STS_READ_COMPLETE /USB_STS_WRITE_COMPLETE の時、usb_ctrl_t 構造体のメンバ type に対し、USB_HHID が設定されるように変更しました。 2. 「6. API」以外の API については USB Basic Host and Peripheral Driver Firmware Integration Technology (ドキュメント番号 :R01AN2025)へ移しました。
1.22	Sep 30, 2017	—	RX65N/RX651-2M をサポート
1.23	Mar 31, 2018	—	1. Smart Configurator に対応しました。 2. R_USB_HhidGetMxps 関数を追加しました。
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
1.31	Mar 1, 2021	—	対象デバイスに RX671 を追加
1.42	Sep 29, 2023	—	usb_hhid_task 関数の引数の型を rtos_task_arg_t へ変更。

## 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. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「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。

### 5. クロックについて

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。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}$  (Max.) から  $V_{IH}$  (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

### 8. 製品間の相違について

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、フラッシュメモリ、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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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も、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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しします。
13. 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。
14. 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.5.0-1 2020.10)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。