

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサス テクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサス エレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサス エレクトロニクス株式会社

【発行】ルネサス エレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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おりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。お客様がかかる法令を遵守しないこと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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

SH7216 グループ

USB ファンクションモジュール USB HID クラス

要旨

本アプリケーションノートは、SH7216 の USB ファンクションモジュールの使用方法、および USB HID (Human Interface Devices) クラスに対応したファームウェアの作成例について掲載しています。

本アプリケーションノートの内容とソフトウェアは USB ファンクションモジュールの応用例を説明するもので、その内容を保障するものではありません。

動作確認デバイス

SH7216

目次

1. はじめに	2
2. 概要	3
3. USB HID クラスの概要	5
4. 開発環境	14
5. サンプルプログラム概要	17
6. 参考ドキュメント	23

1. はじめに

1.1 仕様

本アプリケーションノートは、SH7216 の USB ファンクションモジュールの使用方法、および USB Human Interface Devices (以下、HID) クラスに対応したファームウェアの作成例について掲載しています。

1.2 使用機能

- 割り込みコントローラ (INTC)
- マルチファンクションタイマパルスユニット 2 (MTU2)
- ピンファンクションコントローラ (PFC)
- USB ファンクションモジュール (USB)

1.3 適用条件

- マイコン: SH7216
- 動作周波数: 内部クロック 200 MHz
バスクロック 50 MHz
周辺クロック 50 MHz
- 統合開発環境: ルネサス テクノロジ製
High-performance Embedded Workshop Ver.4.07.00
- C コンパイラ: ルネサス テクノロジ製
SuperH RISC engine ファミリー C/C++ コンパイラパッケージ Ver.9.02 Release 00
- コンパイルオプション: High-performance Embedded Workshop でのデフォルト設定
(-cpu=sh2afpu -pic=1 -object="\$(CONFIGDIR)¥\$(FILELEAF).obj"
-debug -gbr=auto -chgincpath -errorpath -global_volatile=0
-opt_range=all -infinite_loop=0 -del_vacant_loop=0
-struct_alloc=1 -nologo)

1.4 関連アプリケーションノート

- SH7216 グループ アプリケーションノート USB ファンクションモジュール USB マスストレージクラス (RJJ06B1071)

2. 概要

本プログラムでは USB ファンクションモジュール (USB) を使用したコントロール転送, インタラプト転送, および HID クラスコマンド対応処理を行います。

SH7216 内蔵 USB ファンクションモジュールの特長を以下に示します。

- USB プロトコルを自動処理
- エンドポイント 0 に対する USB 標準コマンドを自動処理 (一部コマンドはファームウェアで処理する必要があります。)
- 転送スピード: フルスピード
- 割り込み要求: USB 送受信に必要な各種割り込み信号を生成
- クロック: USB 発振器 (48MHz) による外部入力
- 低消費電力モードを搭載
- バストランシーバを内蔵
- エンドポイント構成: 表 1 に示す構成

表 1 エンドポイント構成

エンドポイント名	名称	転送タイプ	最大パケットサイズ	FIFO バッファ容量	DMA 転送
エンドポイント 0	EP0s	セットアップ	8 バイト	8 バイト	—
	EP0i	コントロールイン	16 バイト	16 バイト	—
	EP0o	コントロールアウト	16 バイト	16 バイト	—
エンドポイント 1	EP1	バルクイン	64 バイト	64×2 (128) バイト	可能
エンドポイント 2	EP2	バルクアウト	64 バイト	64×2 (128) バイト	可能
エンドポイント 3	EP3	インタラプトイン	16 バイト	16 バイト	—
エンドポイント 4	EP4	バルクイン	64 バイト	64×2 (128) バイト	可能
エンドポイント 5	EP5	バルクアウト	64 バイト	64×2 (128) バイト	可能
エンドポイント 6	EP6	インタラプトイン	16 バイト	16 バイト	—
エンドポイント 7	EP7	バルクイン	64 バイト	64 バイト	—
エンドポイント 8	EP8	バルクアウト	64 バイト	64 バイト	—
エンドポイント 9	EP9	インタラプトイン	16 バイト	16 バイト	—

システム構成例を図 1 に示します。

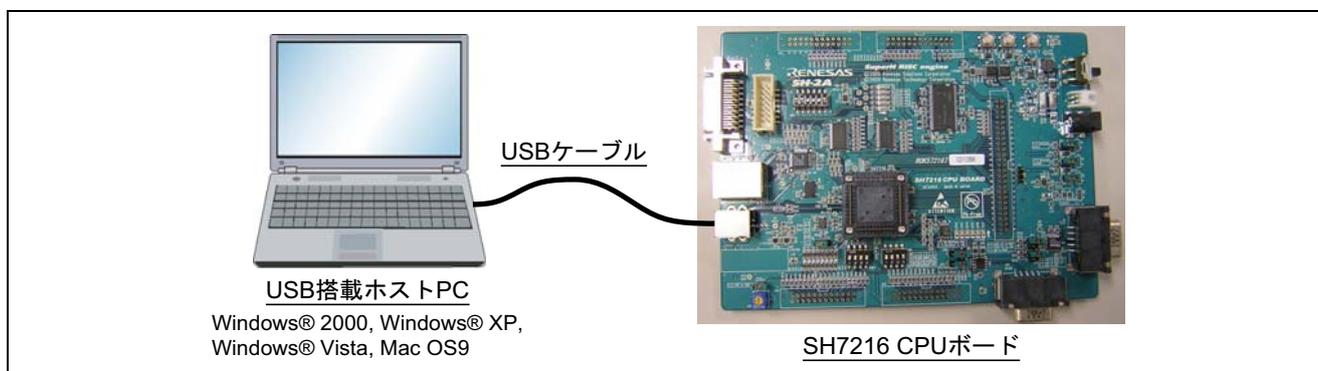


図 1 システム構成例

本システムは、SH7216 を搭載したルネサス テクノロジ製の SH7216 CPU ボード、Windows® 2000、Windows® XP、Windows® Vista、または Mac OS9 を OS として搭載する PC によって構成されています。

本システムは、SH7216 CPU ボード上で擬似マウスデータを自動生成し、USB を通じてホスト PC へマウスデータ (以下、HID データ) を出力する HID クラスファームウェアです。

また、上記 OS に標準で付属している USB HID クラスのデバイスドライバを使用することが可能です。

本システムの特長を以下に示します。

1. サンプルプログラムにより、SH7216 の USB モジュールを評価可能
2. サンプルプログラムは USB のコントロール転送とバルク転送をサポート
3. E10A (USB 接続型エミュレータ) によるデバッグが可能
4. プログラムを追加作成することでバルク転送*も対応可能

【注】 * バルク転送のプログラムは、お客様で作成していただく必要があります。
なお、SH7216 はアイソクロナス転送には対応しておりません。

3. USB HID クラスの概要

この章では、USB HID (Human Interface Devices) クラスについて説明します。

USB HID クラスのデバイスを開発する場合にご参考としてください。なお、規格の詳細は、「6. 参考ドキュメント」の(3)と(4)をご参照ください。

3.1 USB HID クラスについて

USB HID クラスとは、人が PC の操作をする機器に適合するよう規格化されたクラスです。代表的なものとしてはマウス、キーボード、ジョイスティックなどがあります。

ホスト PC に、このクラスのファンクションであることを伝えるためには、Interface Descriptor の bInterfaceClass フィールドに値 H'03 を記述する必要があります。

3.2 サブクラスコードについて

サブクラスでは HID クラスのデバイスが使用する特定のプロトコルを識別するために考えられていましたが、人が使用するデバイスは種類が多くサブクラスのプロトコル定義が非現実的であるため、HID クラスは、ほとんどのプロトコルを定義するためにはサブクラスを使用しません。その代わりに HID クラスデバイスでは、レポートディスクリプタでプロトコルを判別します。

しかし、BIOS がサポートするデバイス (ブートデバイス) は、使用するプロトコルの判別に単純な方法が必要でした。そこで、HID クラスデバイスがマウスあるいはキーボード (すなわち、デバイスがブートデバイスとして用いられることができる) の場合、前もって定められたプロトコル (ブートプロトコル) をサポートするデバイスを示すためにサブクラスを使用します。

ホスト PC に、デバイスがブートプロトコルに対応していることを伝えるためには、Interface Descriptor の bInterfaceSubClass フィールドに H'01 を記述する必要があります。

3.3 プロトコルコードについて

デバイスがブートプロトコルに対応している (サブクラスコードが 0 以外) 場合、対象デバイスを示すために使用します。対象としては「キーボード」(値: H'01) と「マウス」(値: H'02) です。ここで対象デバイスを指定することによりそれぞれのデバイスに適合したプロトコルが使用可能であることを示します。

ホスト PC に、デバイスが何であることを伝えるためには、Interface Descriptor の bInterfaceProtocol フィールドに値を記述する必要があります。

3.4 HID クラスのディスクリプタについて

HID クラスのファンクションデバイスには、他の USB ファンクションデバイスが持っているディスクリプタ情報に加え「HID ディスクリプタ」、「レポートディスクリプタ」、および「フィジカルディスクリプタ」(オプション) が必要になります。HID デバイスのディスクリプタ構成を図 2 に示します。

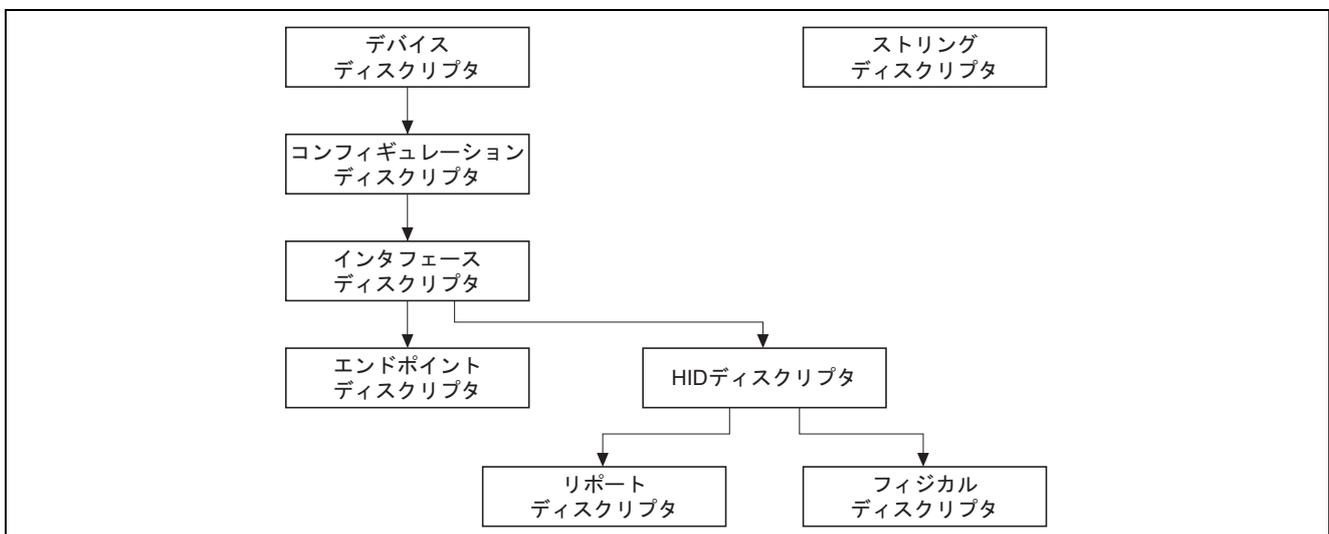


図 2 ディスクリプタ構成

3.5 HID ディスクリプタについて

HID ディスクリプタはリポートディスクリプタ、フィジカルディスクリプタ (オプション) を纏めるために存在します。HID ディスクリプタのフォーマットを表 2 に示します。

表 2 HID ディスクリプタ

フィールド	サイズ (バイト)	内容
bLength	1	ディスクリプタのサイズ (H'09 で固定)
bDescriptorType	1	ディスクリプタのタイプ (H'21 で固定)
bcdHID	2	BCD 表現の HID バージョン
bCountryCode	1	地域固有デバイスのための国識別番号 (必要がなければ 0)
bNumDescriptors	1	クラスディスクリプタの数
bDescriptorType	1	クラスディスクリプタの型 (HIDREPORT の場合 H'22)
wDescriptorLength	2	リポートディスクリプタのサイズ

3.6 リポートディスクリプタについて

リポートディスクリプタは、ホスト PC とデバイス間で転送するデータのフォーマットを決めるために存在します。リポートディスクリプタには他のディスクリプタのように規定されたフォーマットはなく、デバイスの報告あるいは報告のために必要とされるデータフィールドの数に依存して、リポートディスクリプタの長さや内容は変化します。

リポートディスクリプタはデバイスに関する情報を提供するまとまり (アイテム) から構成されます。アイテムには、短いアイテムと長いアイテムの 2 種類があります。ここでは短いアイテムを用いて説明します。

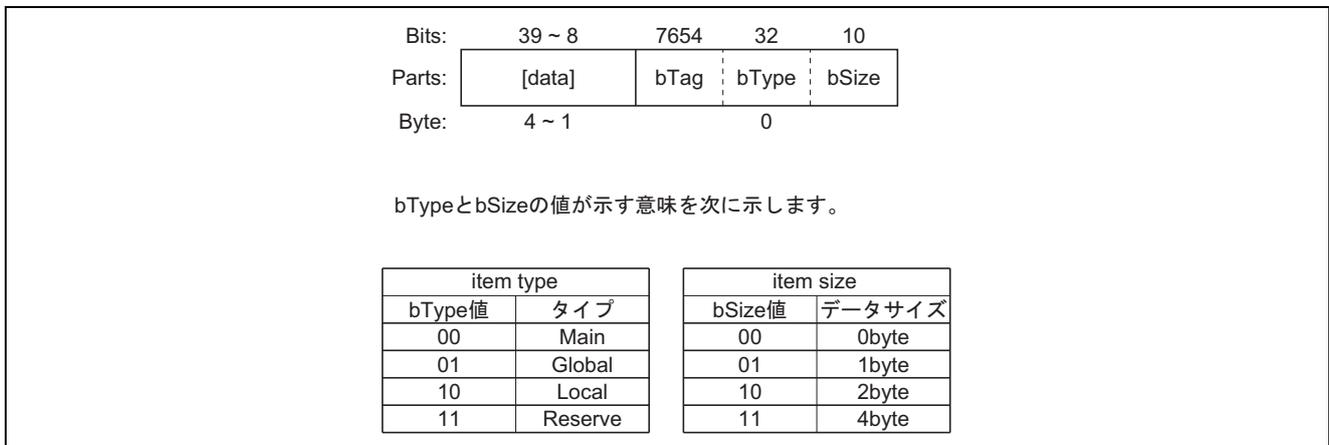


図 3 リポートディスクリプタアイテム

アイテムは「data」、「item tag」、「item type」と「itemSize」の 4 種類から構成されます。アイテムはフィールドを使いどのような種類の情報かを示します。

item type には「Main」「Global」「Local」の 3 種類があり、Main item type (リポートディスクリプタ内のデータフィールドを定義もしくはグループ化に使用) は 5 種類、Global item type (データを記述に使用) は 12 種類、Local item type (特性を定義に使用) は 10 種類の item tag があります。

これらの item tag を組み合わせることにより、ホスト PC とデバイス間で転送するデータのフォーマットを記述します。

3.6.1 Main items

Main item type の item tag 5 種類を表 3 に示します。

表 3 Main item type の item tag

item tag	bTag	bType	bSize	意味
Input	1000	00	nn	1つ以上の物理的なコントロールが提供するデータに関する情報について記述します。
Output	1001	00	nn	出力データフィールドを定義するために使用します。
Feature	1011	00	nn	デバイスに送ることができるデバイスコンフィギュレーション情報について記述します。
Collection	1010	00	nn	2つ以上のデータ (Input, Output あるいは Feature) の関係を纏め始めます。
End Collection	1100	00	nn	Collection に対応し, 2つ以上のデータ (Input, Output あるいは Feature) の関係を終了します。

(1) Input item tag

Input item tag のパラメータ (データフィールド) は 8 種類あり, 1 ビットごとに指定します。内容を表 4 に示します。

表 4 Input item tag のパラメータ

Bit	値	内容	意味
0	0	Data	アイテムはデータを報告する
	1	Constant	アイテムは定数を報告する
1	0	Array	アイテムはアレイデータフィールドを報告する
	1	Variable	アイテムは変数を報告する
2	0	Absolute	アイテムは絶対値を報告する
	1	Relative	アイテムは最後の報告からの偏差を報告する
3	0	No Wrap	アイテムが報告する値はロールオーバーしない
	1	Wrap	アイテムが報告する値はロールオーバーする。(例えば: 0~10 の値を出力するダイヤルで, ダイヤルを回し続けると値 10 の次は 0 を出力する)
4	0	Linear	アイテムはコントロールする物の状態をリニアに報告する
	1	Non Linear	アイテムは生データを処理し, コントロールする物の状態をリニアに報告しない
5	0	Preferred State	アイテムはユーザがコントロールしていない時, 戻る状態を持っている
	1	No Preferred	アイテムはユーザがコントロールしていない時, 戻る状態を持っていない
6	0	No Null position	アイテムは無意味なデータを送る状態を持っていない
	1	Null state	アイテムは無意味なデータを送る状態を持っている
7	0	Reserved	リザーブ
8	0	Bit Field	アイテムはビットフィールドを発する
	1	Buffered Bytes	アイテムは 1 バイト固定サイズのストリームを発する
9-31	0	Reserved	リザーブ

(2) Output item tag & Feature item tag

Output および Feature item tag のパラメータ (データフィールド) は 9 種類あり、内容は Bit7 をのぞき Input item tag と同じです。表 5 に Output および Feature item tag のパラメータを示します。

表 5 Output および Feature item tag のパラメータ

Bit	値	内容	意味
1-6	—	—	Input item tag と同様
7	0	Non Volatile	アイテムの値はホスト対話の有無にかかわらず変化することができない
	1	Volatile	アイテムの値はホスト対話の有無にかかわらず変化することができる
8-31	—	—	Input item tag と同様

(3) Collection item tag

Collection item tag のパラメータ (データフィールド) は 8 種類あり、1 バイトの値で指定します。内容を表 6 に示します。

表 6 Collection item tag のパラメータ

値	内容	意味
H'00	Physical	これは、1 つに集められたデータアイテムに使用します。これは、単一のポイントに正確なデータあるいは感知したデータを関連させる必要がある装置に使用します。 これはデータがキーボードのような、1 つのデバイスから来ることを示さず、多数のセンサーポジションを報告するデバイスなどで、いずれのデータもそれぞれの別のセンサーから来ることを示すために使用します。
H'01	Application	これは、アプリケーションレベルでだけ使われる Usage を識別します。このコレクションが HID デバイスあるいは複雑なデバイスの機能的な下位グループであることを示します。オペレーティング・システムはデバイスをコントロールするアプリケーションあるいはドライバにリンクするために、このコレクションと結び付けられた Usage を使用します。
H'02	Logical	これは、データアイテムが複合したデータ構造を構成する時、使用します。
H'03	Report	これは、フィールドをすべて包む、論理的な収集を定義します。レポート ID はこの収集に含まれます。アプリケーションが容易にデバイスのある機能をサポートするかどうか決定することができます。
H'04	Named Array	これは、データアイテムが複合したデータ構造を構成し、命名する時使用します。
H'05	Usage Switch	これは、これが含んでいる Usage の意味を変更する、論理的な収集です。その収集中の Usage の目的を修正する、論理的な収集に適用された Usage を識別します。
H'06	Usage Modifier	これは、包含するコレクションに付けられた Usage の意味を修正します。Usage は、典型的にはコントロールのために単一操作のモードを定義します。これは、コントロールの操作の方法が拡張されることを可能にします。
H'07-H'7F	Reserved	リザーブ
H'80-H'FF	Vendor-defined.	ベンダ定義

3.6.2 Global items

Global item type の item tag12 種類を表 7 に示します。

表 7 Global item type の item tag

item tag	bTag	bType	bSize	意味
Usage Page	0000	01	nn	現在の Usage Page を指定している値。アイテム使用法のインデックスを定義します。
Logical Minimum	0001	01	nn	変数または配列のアイテムが報告する最小値です。例えば、0~128 まで X 位置値を報告するマウスは、0 の論理的な最小値を持つでしょう。
Logical Maximum	0010	01	nn	変数または配列のアイテムが報告する最大値です。例えば、0~128 まで X 位置値を報告するマウスは、128 の論理的な最大値を持つでしょう。
Physical Minimum	0011	01	nn	可変アイテムの物理的な最小範囲値。
Physical Maximum	0100	01	nn	可変アイテムの物理的な最大範囲値。
Unit Exponent	0101	01	nn	基礎 10 のユニット指数値。
Unit	0110	01	nn	ユニット値。
Report Size	0111	01	nn	ビットでレポートフィールドの大きさを指定している符号なしの値。
Report ID	1000	01	nn	リポート ID を指定する符号なしの値。
Report Count	1001	01	nn	アイテムのためにデータフィールドの数を指定する。符号なしの整数が何個のフィールドがこの特定のアイテムのためにレポートに含まれるか決定する (したがってビットが何個であるかがレポートに付け加えられる)。
Push	1010	01	nn	スタックに Global アイテムステートテーブルのコピーを置きます。
Pop	1011	01	nn	アイテムステートテーブルをスタックからのトップに取り替えます。

3.6.3 Local items

Local item type の item tag10 種類を表 8 に示します。

表 8 Local item type の item tag

item tag	bTag	bType	bSize	意味
Usage	0000	10	nn	現在の Usage を指定している値。アイテム使用法のインデックスを定義します。
Usage Minimum	0001	10	nn	アレイあるいはビットマップと関係づけた Usage のスタートを定義します。
Usage Maximum	0010	10	nn	アレイあるいはビットマップと関係づけた用法 (Usage) のエンドを定義します。
Designator Index	0011	10	nn	コントロールのために使用された身体部分を決定します。
Designator Minimum	0100	10	nn	アレイあるいはビットマップと関係づけた Designator のスタートインデックスを定義します。
Designator Maximum	0101	10	nn	アレイあるいはビットマップと関係づけた Designator のエンドインデックスを定義します。
String Index	0111	10	nn	ストリングのディスクリプタのインデックス。ストリングが特別のアイテムかコントロールに関係することを可能にします。
String Minimum	1000	10	nn	配列かビットマップ中のコントロールに連続するストリングのグループに帰属する場合、第 1 のストリング・インデックスを指定します。
String Maximum	1001	10	nn	配列かビットマップ中のコントロールに連続するストリングのグループに帰属する場合、最後のストリング・インデックスを指定します。
Delimiter	1010	10	nn	1 セットの Local アイテムの始まりか終了を定義します。

3.6.4 リポートディスクリプタの例

本サンプルのリポートディスクリプタを図 4 に示します。

Usage Page (Generic Desktop),	: 05 01
Usage (Mouse),	: 09 02
Collection (Application),	: A1 01
Usage (Pointer),	: 09 01
Collection (Physical),	: A1 00
Usage Page (Buttons),	: 05 09
Usage Minimum (01),	: 19 01
Usage Maximum (03),	: 29 03
Logical Minimum (0),	: 15 00
Logical Maximum (1),	: 25 01
Report Count (3),	: 95 03
Report Size (1),	: 75 01
Input (Data, Variable, Absolute), ; 3 button bits	: 81 02
Report Count (1),	: 95 01
Report Size (5),	: 75 05
Input (Constant), ; 5 bit padding	: 81 01
Usage Page (Generic Desktop),	: 05 01
Usage (X),	: 09 30
Usage (Y),	: 09 31
Usage (Wheel),	: 09 38
Logical Minimum (-127),	: 15 81
Logical Maximum (127),	: 25 7F
Report Size (8),	: 75 08
Report Count (3),	: 95 03
Input (Data, Variable, Relative), ; 2 position bytes (X & Y)	: 81 06
End Collection,	: C0
End Collection	: C0

図 4 リポートディスクリプタ

3.6.5 リポートディスクリプタの説明

本サンプルで使用するリポートディスクリプタの説明を表9に示します。

表9 リポートディスクリプタ

Item	Value (Hex)	Item 区分	意味
Usage Page (Generic Desktop Control)	H'05 01	Global	用法ページ (Usage Page) を示す値。H'01 は"Generic Desktop Control"を表します。
Usage (Mouse)	H'09 02	Local	アイテム使用法のインデックス。H'02 は"Mouse"を表します。オペレーティング・システムがデバイスをアクティブなアプリケーションあるいはドライバにマウスとしてリンクします。"Mouse"の使用法タイプは"Collection Application"です。
Collection (Application)	H'A1 01	Main	アプリケーションにマウスとして"Pointer"を伝えます。
Usage (Pointer)	H'09 01	Local	アイテム使用法のインデックス。H'01 は"Pointer"を表します。"Pointer"の使用法タイプは"Collection Physical"です。
Collection (Physical)	H'A1 00	Main	ポインタとして多数のセンサのポジション (ボタン, X 軸, Y 軸, ロータリ・コントロール) を1つに集めます。
Usage Page (Button)	H'05 09	Global	用法ページ (Usage Page) を示す値。H'09 は"Button"を表します。
Usage Minimum (1)	H'19 01	Local	アレイあるいはビットマップと関係づけた用法 (Usage) は1からスタートすることを定義します。
Usage Maximum (3)	H'29 03	Local	アレイあるいはビットマップと関係づけた用法 (Usage) は3で終わることを定義します。
Logical Minimum (0)	H'15 00	Global	アイテムが報告する最小値は0です。
Logical Maximum (1)	H'25 01	Global	アイテムが報告する最大値は1です。
Report Count (3)	H'95 03	Global	アイテムにデータフィールドをいくつ使うかを示す。ここでは3個のレポートフィールドを使うことを表します。
Report Size (1)	H'75 01	Global	レポートフィールドの大きさを示す。ここでは1ビット使うことを表します。
Input (Data, Variable, Absolute)	H'81 02	Main	入力するアイテムはどのようなものかを示します。入力は変化するデータで、絶対値を報告します。
Report Count (1)	H'95 01	Global	アイテムにデータフィールドをいくつ使うかを示す。ここでは1個のレポートフィールドを使うことを表します。
Report Size (5)	H'75 05	Global	レポートフィールドの大きさを示す。ここでは5ビット使うことを表します。
Input (Constant)	H'81 01	Main	入力するアイテムはどのようなものかを示します。入力は定数を報告します。
Usage Page (Generic Desktop Control)	H'05 01	Global	用法ページ (Usage Page) を示す値。H'01 は"Generic Desktop Control"を表します。
Usage (X)	H'09 30	Local	アイテム使用法のインデックス。H'30 は"X"を表します。コントローラが報告する値はX方向の値で、ユーザから見て「左」から「右」にコントローラを動かした時、リニアに値が増加します。
Usage (Y)	H'09 31	Local	アイテム使用法のインデックス。H'31 は"Y"を表します。コントローラが報告する値はY方向の値で、ユーザから見て「離れている」から「近い」にコントローラを動かした時、リニアに値が増加します。
Usage (Wheel)	H'09 38	Local	アイテム使用法のインデックス。H'38 は"Wheel"を表します。ダイヤルとは異なり、回転され可変値を生成するロータリ・コントロールです。コントローラが前方に、(ユーザから遠ざかるように) 回転すると、値は増加します。
Logical Minimum (-127)	H'15 81	Global	アイテムが報告する最小値は-127です。
Logical Maximum (127)	H'25 7F	Global	アイテムが報告する最大値は127です。
Report Size (8)	H'75 08	Global	レポートフィールドの大きさを示す。ここでは8ビット使うことを表します。
Report Count (3)	H'95 03	Global	アイテムにデータフィールドをいくつ使うかを示す。ここでは3個のレポートフィールドを使うことを表します。
Input (Data, Variable, Relative)	H'81 06	Main	入力するアイテムはどのようなものかを示します。入力は変化するデータで、前回の入力からの変化分を報告します。
End Collection	H'C0	Main	データセットとして1つに纏める終点を表します。(Physical)
End Collection	H'C0	Main	データセットとして1つに纏める終点を表します。(Application)

3.7 フィジカルディスクリプタについて

フィジカルディスクリプタはデバイスをコントロールしている人の体（あるいは体の特定の部分）に関する情報を提供するために存在します。このディスクリプタはオプションであり、省略することが可能です。本サンプルプログラムでは省略しています。

3.8 HID データの転送フォーマットについて

ホスト PC とファンクション間で HID データを転送する場合、主にインタラプト転送を使用し（コントロール転送も使用可）、ホスト PC とファンクション間でデータの転送が行われます。

ブートデバイスが使用可能なプロトコルは、レポートプロトコルとブートプロトコルの 2 種類。非ブートデバイスが使用可能なプロトコルは、レポートプロトコルの 1 種類です。

レポートプロトコルがデータ転送に使用するフォーマットはレポートディスクリプタで記述します。

ブートプロトコルがデータ転送に使用するフォーマットは規格書にデータフォーマットが記されています。

ブートデバイスのデフォルトプロトコルはレポートプロトコルですが、ブートプロトコルまたはレポートプロトコルどちらを使用するかはクラスコマンドで指定することができます。本サンプルプログラムのレポートプロトコルフォーマットを図 5 に示します。



図 5 レポートプロトコルフォーマット

3.9 クラスコマンドについて

クラスコマンドとは、USB のクラス定義ごとに定められているコマンドです。クラスコマンドはコントロール転送を使用します。

USB HID クラスのクラスコマンドは 6 種類あります。表 10 にクラスコマンドを示します。

表 10 クラスコマンド一覧

bRequest フィールド値	コマンド	コマンドの意味
H'01	GET_REPORT	コントロール転送を使いデバイスからホスト PC に HID データを転送する
H'02	GET_IDLE	インタラプト転送を止めている時間レートの現在値を返す
H'03	GET_PROTOCOL	現在アクティブなプロトコル（ブートプロトコルまたはレポートプロトコル）を報告する
H'09	SET_REPORT	コントロール転送を使いホスト PC からデバイスに HID データを転送する
H'0A	SET_IDLE	インタラプト転送を止めている時間レートを設定する
H'0B	SET_PROTOCOL	アクティブなプロトコル（ブートプロトコルまたはレポートプロトコル）を設定する

- 【注】 1. すべてのデバイスが GET_REPORT をサポートする必要があります。
2. ブートデバイスは GET_PROTOCOL と SET_PROTOCOL をサポートする必要があります。

GET_REPORT コマンドを受信した場合、ファンクションはコントロール転送のデータステージを使用して HID データをホストに送信します。セットアップデータ内 wValue フィールドの上位 1 バイトでレポートタイプを指定し、wValue フィールドの下位 1 バイトでレポート ID を指定します。レポート ID を使用しない場合は値 0 が指定されます。

GET_IDLE コマンドを受信した場合、ファンクションはインタラプト転送を止めている時間を返答します。返答する時間は、4ms を 1 単位とするタイムレートで返答します。ホストはセットアップデータ内 wValue フィールドの下位 1 バイトで返答するレポート ID を指定します。この値が 0 の場合、該当デバイスの全インタラプト転送のタイムレートを返答します。

GET_PROTOCOL コマンドを受信した場合、ファンクションはコントロール転送のデータステージを使用して現在選択されているプロトコル（ブートプロトコルまたはリポートプロトコル）をホストに返答します。返答値 0 はブートプロトコル、返答値 1 はリポートプロトコルを表します。

SET_REPORT コマンドを受信した場合、ファンクションはコントロール転送のデータステージを使用した HID データを受信します。しかし、ファンクションはホストからの指示を無視するかもしれません。

SET_IDLE コマンドを受信した場合、ファンクションは指定された時間インタラプト転送を止めます。指定はセットアップデータ内 wValue フィールドの上位 1 バイトで指定されます。なお、指定される時間は、タイムレートで指定され、1 単位は 4ms を表します。wValue フィールドの下部 1 バイトはレポート ID が指定されます。この値が 0 以外の場合、指定されたレポート ID の転送を停止します。この値が 0 の場合、該当デバイスの全インタラプト転送を停止します。

SET_PROTOCOL コマンドを受信した場合、ファンクションはそれ以降使用するプロトコル（ブートプロトコルまたはリポートプロトコル）を設定します。指定はセットアップデータ内 wValue フィールドで指定（値 0 はブートプロトコル、値 1 はリポートプロトコル）されます。なお、ファンクションはリポートプロトコルを初期値としています。

4. 開発環境

この章では、本システムの開発に使用した開発環境について説明します。本システムの開発は、以下のデバイス (ツール) を使用します。

- SH7216 CPU ボード (型名 R0K572167) ルネサス テクノロジ製
- E10A-USB Emulator ルネサス テクノロジ製
- E10A PC (Windows® 2000, Windows® XP)
- USB ホスト PC (Windows® 2000, Windows® XP, Windows® Vista)
- USB ケーブル
- High-performance Embedded Workshop 4 (以下 HEW4) ルネサス テクノロジ製

4.1 ハードウェア環境

図 6 に各デバイスの接続形態を示します。

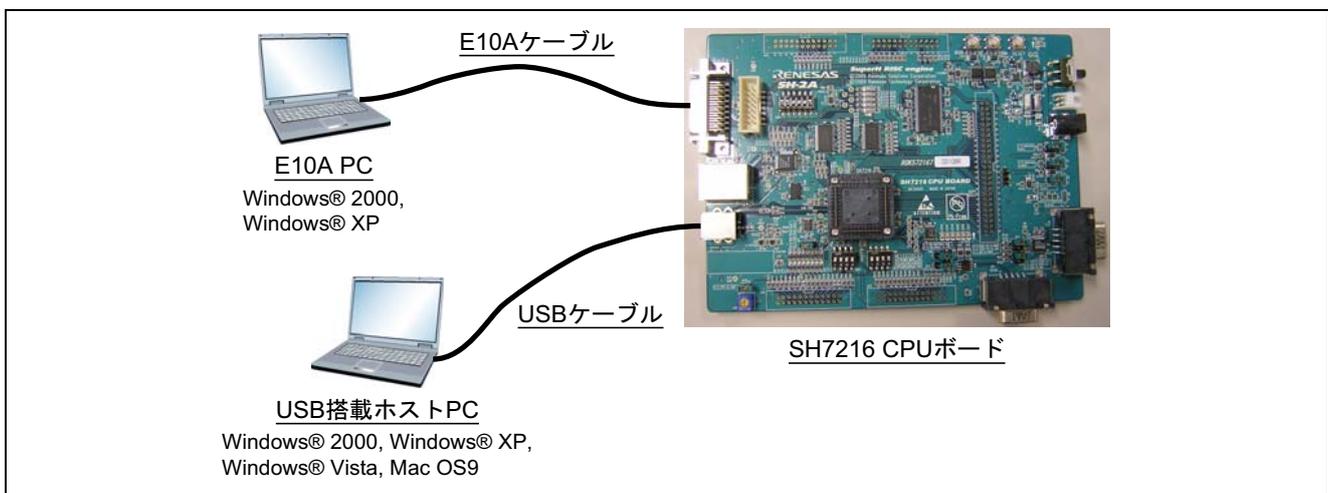


図 6 デバイスの接続形態

(1) SH7216 CPU ボード

本システムでは内蔵 ROM を使用するため、SH7216 CPU ボードを内蔵 ROM が有効なモードで動作させる必要があります。例えば、SH7216 CPU ボードのディップスイッチの SW1 を出荷時の設定から表 11 に示すように変更することで、内蔵 ROM が有効な MCU 拡張モード 2 (内蔵 ROM と SDRAM 有効) で動作させることができます。電源を投入する前に、これらの設定をよくご確認ください。その他のディップスイッチを変更する必要はありません。

表 11 ディップスイッチの設定

出荷時 (モード 6)	変更後 (モード 2)	ディップスイッチの機能
SW1-1 (FWE) OFF	SW1-1 (FWE) ON	内蔵フラッシュメモリの書き込み/消去プロテクト
SW1-2 (MD1) OFF	SW1-2 (MD1) OFF	MD1 端子状態
SW1-3 (MD0) ON	SW1-3 (MD0) ON	MD0 端子状態

(2) USB ホスト PC

USB ポート搭載の Windows® 2000, Windows® XP, Windows® Vista または Mac OS9 をインストールしたパソコンを USB ホスト PC として使用します。本システムでは、上記 OS に標準で搭載されている USB HID クラスのデバイスドライバを使用しますので、新たにドライバをインストールする必要はありません。

(3) E10A PC

USB ポート搭載の Windows® 2000, Windows® XP をインストールしたパソコンを E10A PC として使用します。E10A-USB 用 PC の USB コネクタに E10A-USB エミュレータを接続し、接続用のケーブルを介して E10A-USB と CPU ボードを接続してください。接続後、HEW4 を起動してエミュレーションを行います。

4.2 ソフトウェア環境

ソースコードのコンパイル、リンク、およびデバッグは HEW4 で行ってください。HEW4 は本フォルダ内 HID.hws をダブルクリックすることで起動します。

4.2.1 サンプルプログラム

サンプルプログラムとして必要なファイルは、すべて HID フォルダ内に収められています。HEW4 がインストールされたパソコンに、このフォルダごと移動して頂くと、すぐにサンプルプログラムを使用することができます。

フォルダに含まれるファイルを図 7 に示します。

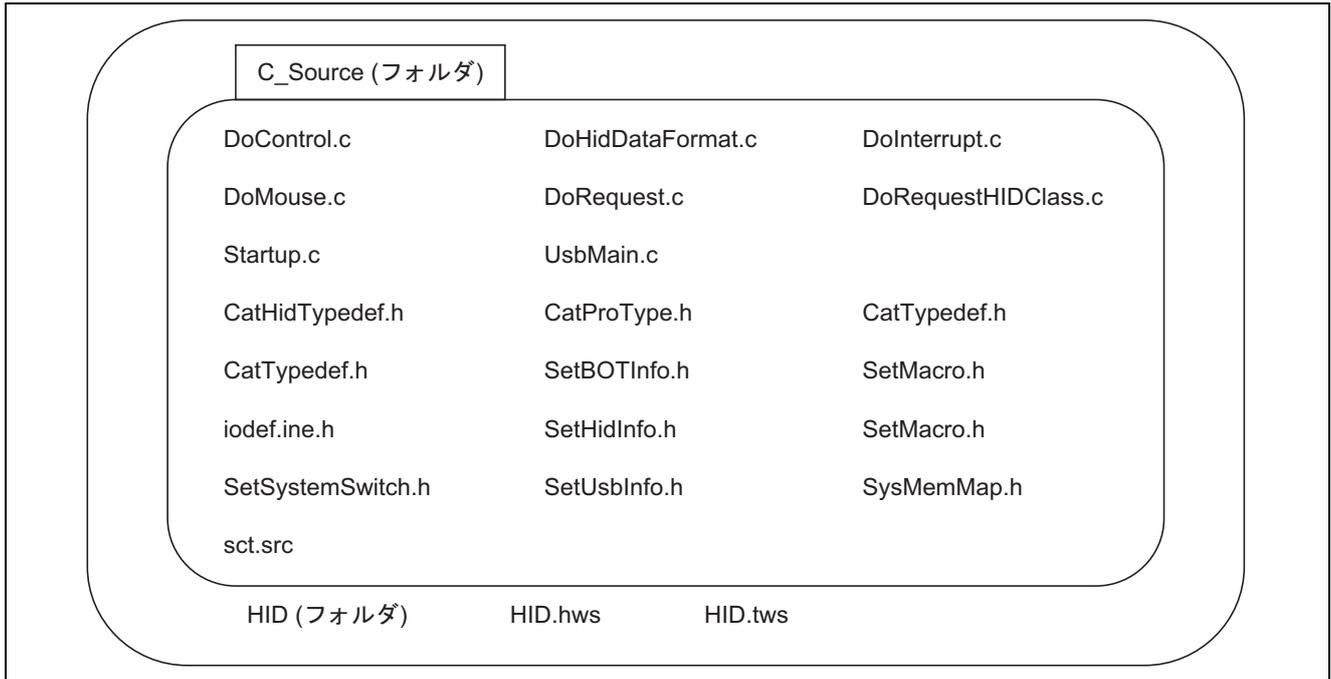


図 7 フォルダに含まれるファイル

4.2.2 コンパイルおよびリンク

ソースコードのコンパイルは HEW4 で行ってください。

4.3 プログラムのロードと実行方法

プログラムのロードと実行方法について説明します。

4.3.1 プログラムのロード

SH7216 CPU ボードにサンプルプログラムをロードするには、以下のような手順で行います。

- HEW4 をインストールした E10A-USB 用 PC に E10A-USB を接続してください。
- ユーザケーブルで E10A-USB と SH7216 CPU ボードを接続してください。
- SH7216 CPU ボードの電源を投入してください。
- sh7216_usb_hid フォルダ内の HID.hws を実行してください。
- 「デバッグ > 接続」を選択してください。
- エミュレータのモードの選択を要求されるので、「SH7216 (R5F72167A)」, 「E10A-USB Emulator」を選択してください。
- SH7216 CPU ボードのリセットスイッチを押し、「OK」ボタンを押してください。
- 動作周波数の入力を求められるので、実装している水晶発振子 (12.50MHz) の周波数を入力してください。
- ID Code の入力を求められるので、「E10A」を入力してください。
- 「デバッグ > ダウンロード > All Download Modules」を選択することでプログラムのダウンロードが行われます。

4.3.2 プログラムの実行

「デバッグ > リセット後実行」を選択することでプログラムが実行されます。

4.4 デモンストレーション (マウスポインタ移動) の実行方法

本サンプルプログラムではマウスを接続せずにホスト PC のマウスポインタが移動するデモンストレーションを実行します。

プログラムを実行した状態で、USB ケーブルのシリーズ B コネクタ SH7216CPU ボードに、シリーズ A コネクタを USB ホスト PC に接続します。コントロール転送後、デバイスマネージャにヒューマンインタフェースデバイス/USB ヒューマンインタフェースデバイスが表示され、ホスト PC は SH7216CPU ボードをマウスデバイスとして認識します。

ホスト PC に接続後、マウスポインタが移動するデモンストレーションを開始します。ホスト PC からのインタラプトイン転送に応じて SH7216CPU ボードがマウスポインタ移動データを送出します。その結果、USB ホスト PC 上のマウスポインタが自動的に移動を開始します。

5. サンプルプログラム概要

この章ではサンプルプログラムの特長やその構成について説明します。本サンプルプログラムは SH7216 CPU ボード上で動作する HID クラスファームウェアです。本ボード上でマウスポインタの移動データを生成することにより、ホスト PC 上でマウスポインタの移動をエミュレートします。本サンプルプログラムはホスト PC からのトークンによって USB 転送を開始します。SH7216 内蔵モジュールの割り込みのうち、USB ファンクションモジュールに関連する割り込みは、USI0、USI1、USBRXI0、USBTXI0、USBRXI1、USBTXI1 の 6 種類ですが、本サンプルでは USI0、USI1 のみ使用しています。

5.1 状態遷移図

図 8 に、本サンプルプログラムの状態遷移図を示します。本サンプルプログラムは、図 5 のように 4 つの状態に遷移します。

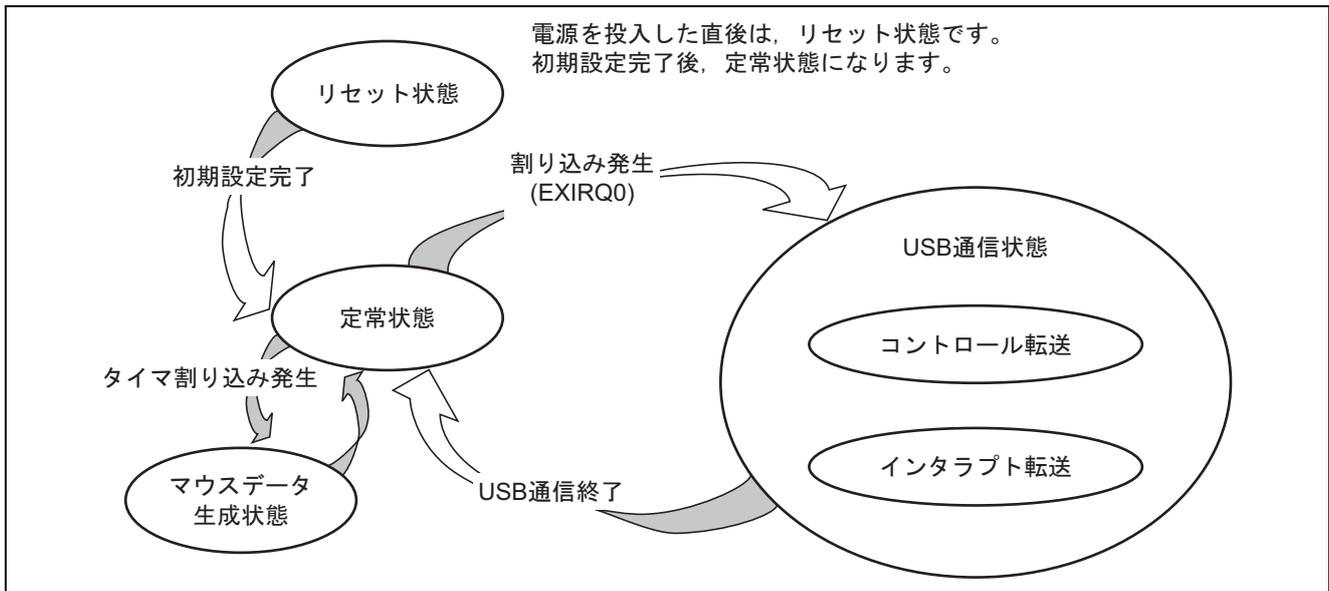


図 8 状態遷移図

- リセット状態
パワーオンリセット・マニュアルリセットの際には、この状態になります。リセット状態では、主に SH7216 の初期設定を行います。
- 定常状態
初期設定が完了すると、メインループで定常状態となります。
- USB 通信状態
定常状態において、USB モジュールから割り込みが発生するとこの状態に遷移します。USB 通信状態では、割り込みの種類に応じた転送方式によるデータ転送を行います。本サンプルプログラムで使用する割り込みは割り込みフラグレジスタ 0, 1, 2, 3, 4 (USBIFR0, 1, 2, 3, 4) によって示されます。割り込み要因が発生すると、USBIFR0, 1, 2, 3, 4 の対応するビットに 1 がセットされます。
- マウスデータ生成状態
定常状態において、16 ビットタイマである MTU2 のオーバーフロー割り込みが発生するとこの状態に遷移します。マウスデータ生成状態ではマウスポインタの移動データを自動生成します。オーバーフロー割り込みは 10ms 間隔で発生します。

5.2 USB 通信状態

USB 通信状態は、転送方式ごとに2つの状態に分類することができます (図9 参照)。割り込みが発生すると、まず USB 通信状態へと遷移し、さらに割り込みの種類に応じて各転送状態へ分岐します。

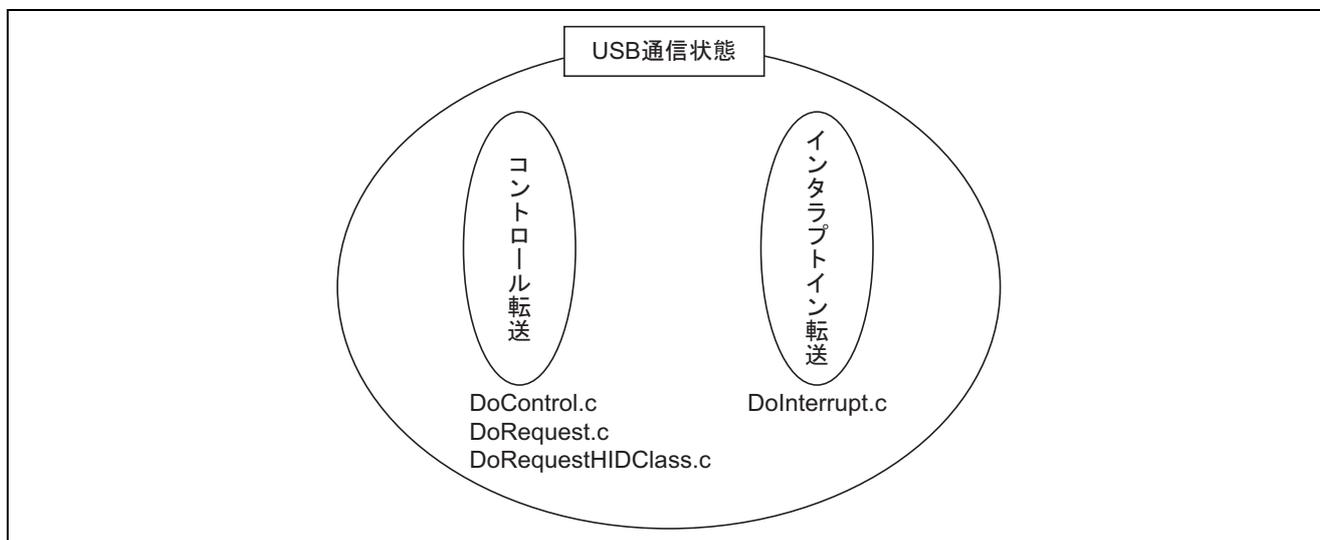


図9 USB 通信状態

5.2.1 コントロール転送について

コントロール転送は主に、デバイス情報の取得、デバイスの動作状態を設定する際などに使用されます。そのため、ホスト PC にファンクションを接続した際、最初に行われる転送でもあります。

コントロール転送の一連の転送処理は、2 または 3 つのステージから構成されます。コントロール転送のステージは、「セットアップステージ」「データステージ」「ステータスステージ」に分類することができます。

5.2.2 インタラプティン転送について

インタラプト転送は、一定の周期でデータを転送する方式で、データの内容が保証されます。USB HID クラスではインタラプト転送を使用し、ホスト PC とファンクション間でマウスやキーボード等のデータを転送します。

5.3 ファイル構成

本サンプルプログラムは、9個のソースファイルと11個のヘッダファイルで構成されています。全構成ファイルを表12に示します。各関数は、転送方式または機能ごとに1つのファイルにまとめてあります。

表 12 ファイル構成

ファイル名	主な機能
Startup.c	USB ファンクションの初期設定
UsbMain.c	割り込み要因の判定 パケットの送受信
DoRequest.c	ホスト PC が発行するセットアップコマンドの処理
DoRequestHIDClass.c	USB HID クラスコマンドの処理
DoControl.c	コントロール転送を実行
DoInterrupt.c	インタラプトイン転送を実行
DoHidDataFormat.c	転送する HID データのフォーマット処理
DoMouse.c	マウスデータ生成処理
sct.src	変数の初期値等の ROM から RAM への転送
CatHidTypedef.h	HID クラス固有の型、構造体定義
CatProType.h	プロトタイプ宣言
CatTypedef.h	USB ファームウェアで使用する基本の構造体定義
SetMacro.h	マクロ定義
SetHidInfo.h	HID クラスコマンド対応に必要な変数の初期設定
SetSystemSwitch.h	システムの動作設定
SetUsbInfo.h	USB ファームウェアで使用する変数の初期設定
SysMemMap.h	メモリマップのアドレス定義
iodef.h	SH7216 のレジスタ定義

5.4 関数の機能

各ファイルに含まれる関数とその機能を表13～表20に示します。

- Startup.c
 パワーオンリセット、またはマニュアルリセットの際には、Startup.c の SetPowerOnSection が呼び出されます。ここでは SH7216 の初期設定や USB クロック等の初期設定を行います。

表 13 Startup.c

格納ファイル	関数名	機能
Startup.c	SetPowerOnSection	モジュールおよびメモリの初期化を行い、メインループへ移行
	_INITISCT	初期値がある変数を、RAM のワークエリアにコピーする
	InitSystem	USB バスのプルアップ制御
	Set_EPInfoR	エンドポイント情報の書き込みを行う

- UsbMain.c

UsbMain.c では、主に USB 割り込みフラグレジスタによって割り込み要因を判定し、割り込みの種類に応じた関数の呼び出しを行います。また、ホストコントローラとファンクションモジュール間におけるパケットの送受信を行います。

表 14 UsbMain.c

格納ファイル	関数名	機能
UsbMain.c	BranchOfInt0	バスリセット、USB ケーブル接続、およびエンドポイント 0 の割り込み要因の判定と、割り込みに応じた関数を呼び出す
	BranchOfInt1	エンドポイント 1 から 9 までの割り込み要因の判定と、割り込みに応じた関数を呼び出す
	GetPacket	ホストコントローラから転送されたデータを、RAM に書き込む
	GetPacket4	ホストコントローラから転送されたデータを、ロングワードサイズで RAM に書き込む (リングバッファ対応版) (USB HID クラスでは使用しません)
	GetPacket4S	ホストコントローラから転送されたデータを、ロングワードサイズで RAM に書き込む (リングバッファ非対応、高速版)
	PutPacket	ホストコントローラに転送するデータを USB モジュールに書き込む
	PutPacket4	ホストコントローラに転送するデータを、ロングワードサイズで USB モジュールに書き込む (リングバッファ対応版) (USB HID クラスでは使用しません)
	PutPacket4S	ホストコントローラに転送するデータを、ロングワードサイズで USB モジュールに書き込む (リングバッファ非対応、高速版)
	SetControlOutContents	ホストから送られたデータに書き換える
	SetUsbModule	USB モジュールの初期設定
	ActBusReset	バスリセット受信時に FIFO のクリア等を行う
	ActBusVcc	USB ケーブル接続割り込み動作を行う
	ConvRealn	指定した番地から指定バイト長のデータを読み出す
	ConvReflexn	指定した番地から指定バイト長のデータを逆順に読み出す

- DoRequest.c

コントロール転送時に、ホストコントローラから送られてくるコマンドをデコードし、コマンドに応じた処理を行います。本サンプルプログラムでは、ベンダ ID の値に H'045B (ベンダ: ルネサス) を使用しています。お客様にて製品を開発される場合は「USB Implementers Forum」にてお客様のベンダ ID を取得願います。また、ベンダコマンドは使用していないため、DecVenderCommands では何も行っていません。ベンダコマンドを使用する際には、お客様でプログラムを作成願います。

表 15 DoRequest.c

格納ファイル	関数名	機能
DoRequest.c	DecStandardCommands	ホストコントローラが発行したコマンドをデコードし、そのうち標準コマンドの対応を行う
	DecVenderCommands	ベンダコマンドの対応を行う

- DoRequestHIDClass.c
 HID クラスコマンド (GET_REPORT, GET_IDLE, GET_PROTOCOL, SET_REPORT, SET_IDLE, SET_PROTOCOL) に応じた下記の処理を行います。
 - GET_REPORT コマンドはコントロール転送を使いデバイスからホスト PC に HID データを転送します。
 - GET_IDLE コマンドはインタラプト転送を止めている時間のレート値を返します。
 - GET_PROTOCOL コマンドは現在選択されているプロトコル (ブートプロトコルまたはリポートプロトコル) を返します。
 - SET_REPORT コマンドはコントロール転送を使いホスト PC からデバイスに HID データを転送するコマンドですが、本サンプルでは HID データのアウト方向の通信をサポートしないので、データの受信のみ行います。
 - SET_IDLE コマンドはインタラプト転送を止めている時間のレートを設定します。
 - SET_PROTOCOL コマンドは使用するプロトコル (ブートプロトコルまたはリポートプロトコル) を設定します。

表 16 DoRequestHIDClass.c

格納ファイル	関数名	機能
DoRequestHIDClass.c	DecHIDClassCommands	HID クラスコマンドの対応を行う
	ActIdleCount	SOF 割り込みで呼び出され、インタラプト転送を止めている時間を計算します。

- DoControl.c
 コントロール転送の割り込み (SETUP TS) が入ると、ActControl がコマンドを取得し、DecStandardCommands でデコードを行いコマンドの転送方向を判別します。その後、コントロール転送の割り込み (EP0o TS, EP0i TR, EP0i TS) が発生すると ActControlInOut がコマンドの転送方向により、ActControlIn または ActControlOut を呼び出しデータステージとステータスステージを行います。

表 17 DoControl.c

格納ファイル	関数名	機能
DoControl.c	ActControl	コントロール転送のセットアップステージの制御を行う
	ActControlIn	コントロールイン転送 (データステージがイン方向の転送) のデータステージとステータスステージの制御を行う
	ActControlOut	コントロールアウト転送 (データステージがアウト方向の転送) のデータステージとステータスステージの制御を行う
	ActControlInOut	コントロール転送のデータステージとステータスステージを ActControlIn と ActControlOut に振り分ける

- DoInterrupt.c
 ホスト PC からのインタラプト転送イントークンに対応し、インタラプト転送バッファが空き次第、次に転送するデータの準備を行う。

表 18 DoInterrupt.c

格納ファイル	関数名	機能
DoInterrupt.c	ActInterruptIn	インタラプト転送のイントークンに対応し FIFO が空き次第データ転送バッファからデータを取り出しインタラプト転送の準備を行う。

- DoHidDataFormat.c
 ホスト PC に送信する HID データの送信準備を行います。

表 19 DoHidDataFormat.c

格納ファイル	関数名	機能
DoHidDataFormat.c	ActMakeHidData	この関数は HID データ伝送のプログラム・インタフェースです。 ActReportProtocol 関数を呼び出した後、インタラプト転送が停止しているなら、ActInterruptIn 関数を呼び出す。
	ActReportProtocol	転送するデータの並びをリポートディスクリプタで決められたフォーマットに整え、送信バッファにデータを書き込む。

- DoMouse.c
 タイマ割り込みを使用してマウスポインタの移動データを生成します。

表 20 DoMouse.c

格納ファイル	関数名	機能
DoMouse.c	MousePushedData Input2	タイマ割り込みで起動し、時間経過に応じてマウスポインタ移動データを生成する。

6. 参考ドキュメント

- ソフトウェアマニュアル
 - (1) SH-2A, SH2A-FPU ソフトウェアマニュアル (RJJ09B0086)
(最新版をルネサス テクノロジホームページから入手してください)
- ハードウェアマニュアル
 - (2) SH7216 グループ ハードウェアマニュアル (RJJ09B0575)
(最新版をルネサス テクノロジホームページから入手してください)
- USB 規格関連
 - (3) Universal Serial Bus Specification
 - (4) Device Class Definition for Human Interface Devices (HID)
 - (5) HID Usage Tables
 - USB 開発者向けホームページ
<http://www.usb.org/developers>

ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.07.15	—	初版発行
2.00	2010.02.10	2	動作周波数を修正 統合開発環境のバージョンを修正
		15	図 7 を修正
		16	4.3 の説明を修正

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。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事事務の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認いただきますとともに、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したのですが、万一本資料の記述の誤りに起因する損害がお客様に生じた場合においても、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて製品をご使用された場合の故障および事故につきましては、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊社製品の故障または誤動作が生じた場合も人身事故、火災事故、社会的損害など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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
 - 1 1. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
 - 1 2. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
 - 1 3. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会ください。

D039444