

SH7216 グループ

R01AN0294JJ0110

Rev.1.10

USB ファンクションモジュール USB マルチファンクション動作例 2011.03.17

要旨

本アプリケーションノートは、SH7216 搭載の USB ファンクションモジュールの使用方法として、HID クラスとマスストレージクラスの両方に対応したサンプルプログラムの動作例について説明しています。

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 マルチファンクションの概要	5
4. 開発環境	20
5. サンプルプログラムの概要	24
6. 参考ドキュメント	32

1. はじめに

1.1 仕様

SH7216 の USB ファンクションモジュールを使用し、USB Human Interface Device (以下、HID) クラスによる PC 上のマウスポインタ操作と USB マスストレージクラスによるローカルディスクファイル操作の両方に対応した動作を行います。

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 エレクトロニクス製 High-performance Embedded Workshop Ver.4.07.00.007
C コンパイラ	ルネサス エレクトロニクス製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.03 Release 02
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 USB ファンクションモジュール USB マスストレージクラスアプリケーションノート (RJJ06B1071)
- SH7216 グループ USB ファンクションモジュール USB HID クラスアプリケーションノート (RJJ06B1072)

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、バルク転送、インタラプト転送、および USB マルチファンクション (HID クラスコマンド、マスストレージクラスコマンド) 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。

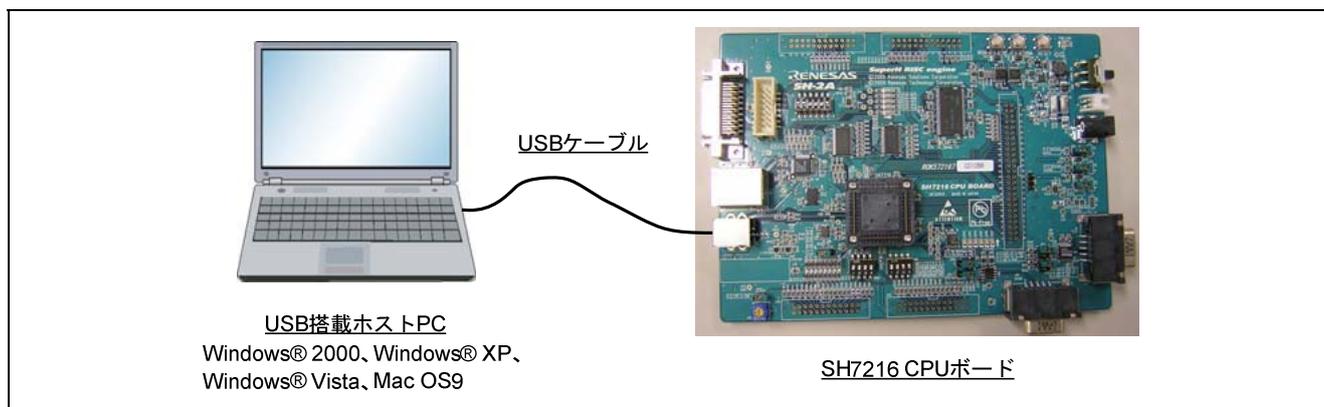


図 1 システム構成例

本システムは、SH7216 を搭載したルネサス エレクトロニクス製の 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、ホスト PC と SH7216 CPU ボードを USB で接続し、HID クラスによる擬似マウスデータ自動生成動作とマストレージクラスによる RAM Disk 動作との USB マルチファンクション機能動作を行うファームウェアです。

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、および USB マストレージクラス (Bulk-Only Transport) 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、インタラプト転送、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

【注】 SH7216 はアイソクロナス転送には対応していません。

3. USB マルチファンクションの概要

本プログラムの USB マルチファンクション処理は、USB マスストレージクラス (Bulk-Only Transport) と USB HID (Human Interface Devices) クラスを使用します。

3.1 USB HID クラスの概要

USB HID 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 USB 規格関連(4)と(5)をご参照ください。

3.1.1 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 を記述することが必要です。

3.1.2 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 を記述することが必要です。

3.1.3 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かを伝えるためには、Interface Descriptor の bInterfaceProtocol フィールドに値を記述することが必要です。

3.1.4 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。

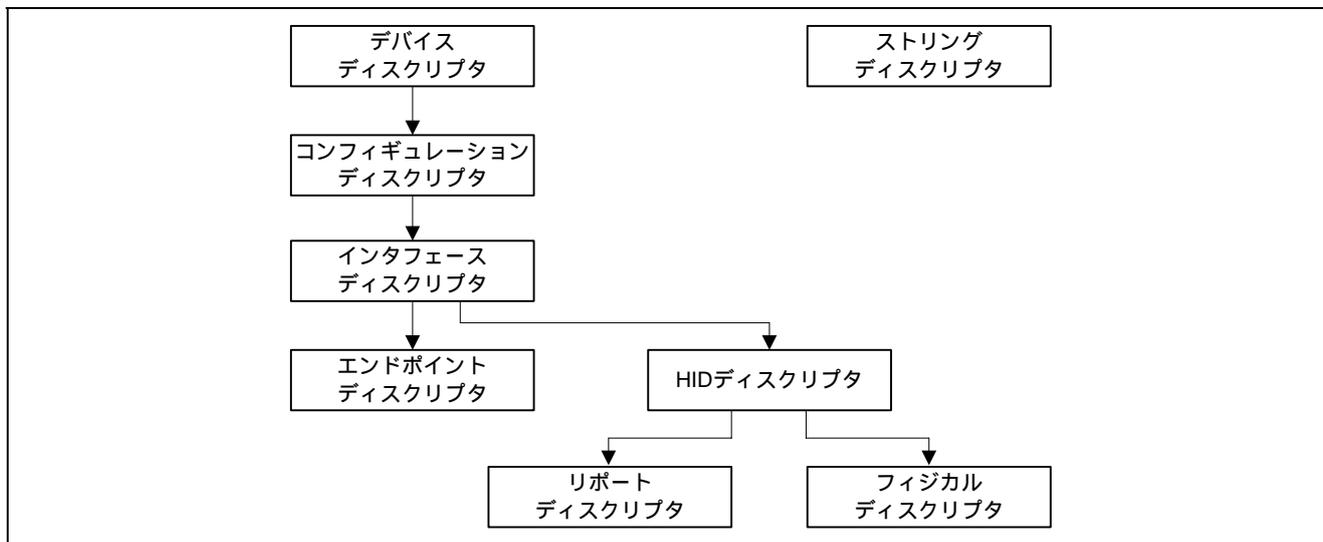


図 2 ディスクリプタ構成

3.1.5 HID ディスクリプタについて

HID ディスクリプタはレ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	レポートディスクリプタのサイズ

3.1.6 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。

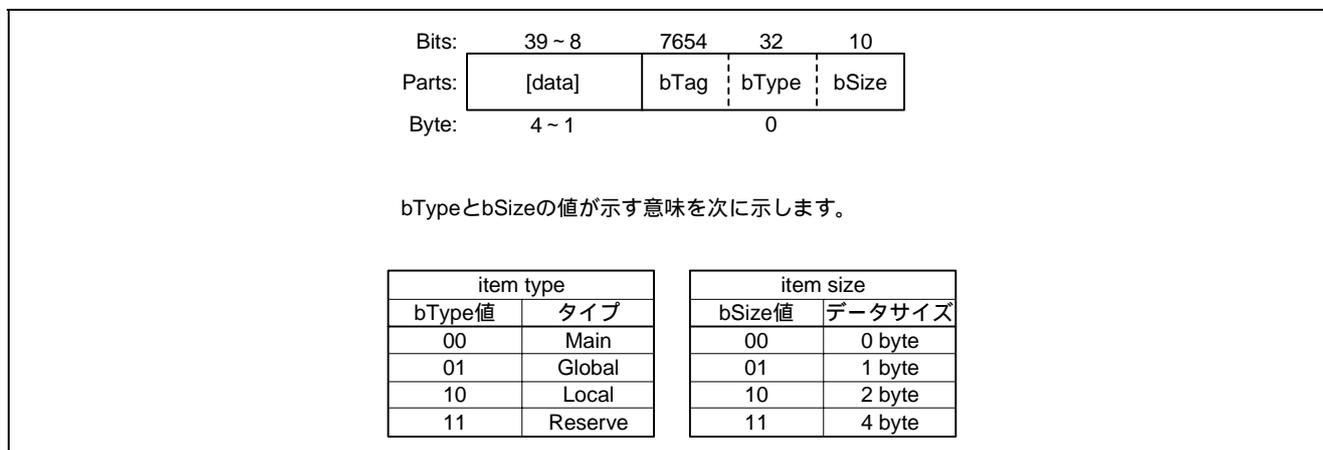


図 3 リポートディスクリプタアイテム

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が示します。

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。

(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。

(a) 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。(たとえば: 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

(b) 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

(c) 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、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とき、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、命名するとき使用します。
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

(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。たとえば、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。たとえば、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	レポート 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) 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が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 アイテムの始まりが終了を定義します。

(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

(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 個のリ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 ビットを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 個のリ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 ビットを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 ビットを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 個のリ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)

3.1.7 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。

3.1.8 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

3.1.9 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レポートプロトコル)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 はレ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 フィールドの下位 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レポートプロトコル)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 はレポートプロトコル)されます。なお、ファンクションはレポートプロトコルを初期値としています。

3.2 USB マスストレージクラス (Bulk-Only Transport) の概要

USB マスストレージクラスについて説明します。

USB のストレージ関連システムを開発する場合に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 USB 規格関連(2)と(3)をご参照ください。

3.2.1 USB マスストレージクラスについて

USB マスストレージクラスとは、大規模記憶装置をホスト PC に接続しデータの書き込み、読み出し等の動作を行う機器に適合するよう規格化されたクラスです。

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'08 を記述することが必要です。また、USB マスストレージクラスでは String Descriptor を用いて Serial Number をホストへ伝える必要があります。本サンプルプログラムではユニコードで 000000000001 を返信しています。

ホスト PC とファンクション間でデータ転送をする場合、USB に規定されている 4 つの転送方法 (コントロール転送、バルク転送、インタラプト転送、アイソクロナス転送) を用いてデータの転送を行います。どの転送方法をどのように使用するかは、プロトコルコードとして定められています。

USB マスストレージクラスではデータ転送プロトコルとして次の 2 種類があります。

- USB Mass Storage Class Bulk-Only Transport
- USB Mass Storage Class Control/Bulk/Interrupt (CBI) Transport

USB Mass Storage Class Bulk-Only Transport は名前の示すとおり、バルク転送のみ使用したデータ転送プロトコルです。

USB Mass Storage Class Control/Bulk/Interrupt (CBI) Transport は、コントロール転送、バルク転送、インタラプト転送を使用したデータ転送プロトコルです。CBI Transport は、さらにインタラプト転送を使用するデータ転送プロトコル、使用しないデータ転送プロトコルの 2 種類に分かれています。

本サンプルプログラムでは、USB Mass Storage Class Bulk-Only Transport をデータ転送プロトコルとして使用します。

ホスト PC がデータのロードやセーブをするために機器を使用する場合、ホスト PC からファンクションに対して命令 (コマンド) を与えます。ファンクションは送られたコマンドを実行することによりデータのロードやセーブが行えます。ホスト PC からファンクションに対して送られるコマンドはサブクラスコードとして定められています。

3.2.2 サブクラスコードについて

サブクラスコードとは、ホスト PC からコマンドトランスポートでファンクションに送られるコマンドフォーマットを表す値です。コマンドフォーマットの種類としては 7 種類あり、表 11 に示すサブクラスコードが定められています。

表 11 サブクラスコード

サブクラスコード	コマンドの規格
H'01	Reduced Block Commands (RBC), T10/1240-D
H'02	Attachment Packet Interface (ATAPI) for CD-ROMs. SFF-8020i, Multi-Media Command Set 2 (MMC-2)
H'03	Attachment Packet Interface (ATAPI) for Tape. QIC-157
H'04	USB Mass Storage Class UFI Command Specification
H'05	Attachment Packet Interface (ATAPI) for Floppies. SFF-8070i
H'06	SCSI Primary Commands -2 (SPC-2), Revision 3 or later

ホスト PC に、機器が対応しているコマンドフォーマットを伝えるためには、Interface Descriptor の `bInterfaceSubClass` フィールドにサブクラスコード値を記述する必要があります。

本サンプルプログラムでは、サブクラスコード値 H'06 の SCSI Primary Commands を使用します。

3.2.3 Bulk-Only Transport について

Bulk-Only Transport はバルク転送のみ使用し、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。

バルク転送は、データを送信する向きにより 2 つに分けることができます。ホストコントローラからファンクションにデータを送信する転送をバルクアウト転送。ホストコントローラにファンクションからデータを送信する転送をバルクイン転送と言います。

Bulk-Only Transport では、バルクアウト転送とバルクイン転送をあらかじめ定めた組み合わせにすることにより、ホスト-ファンクション間のデータ転送を行います。Bulk-Only Transport は必ず図 6 に示すバルク転送の組み合わせになります。それぞれのバルク転送には異なった意味がありステージ（トランスポート）として管理します。

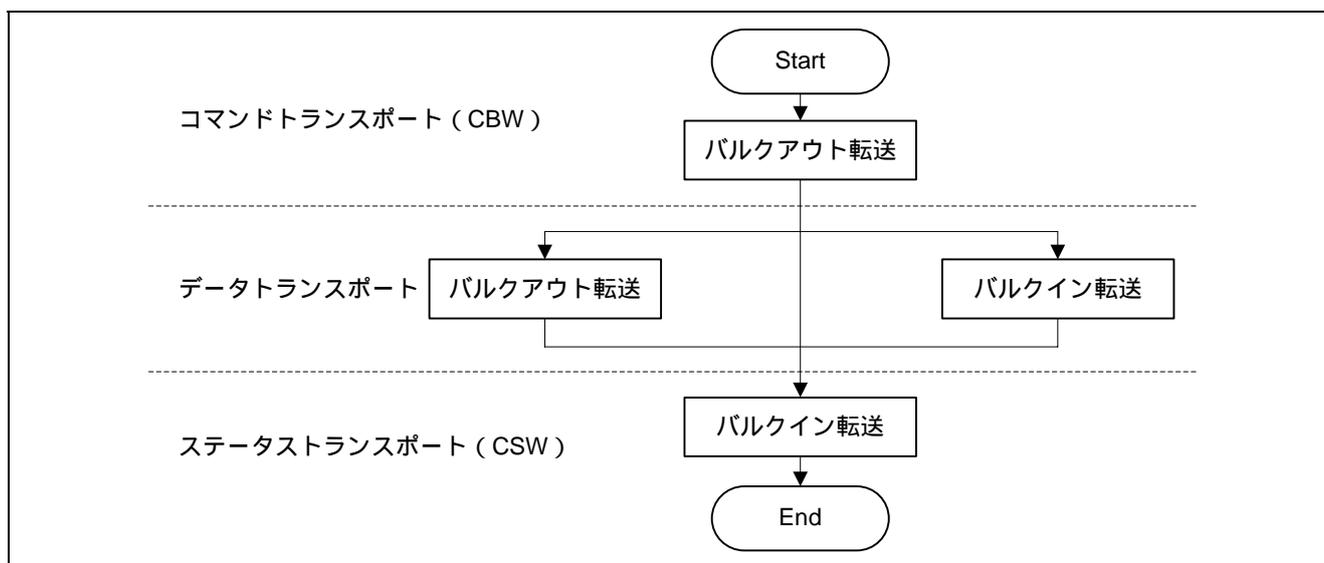


図 6 転送方法とトランスポートの関係

ホスト PC に、Bulk-Only Transport プロトコルの使用を伝えるためには、Interface Descriptor の `bInterfaceProtocol` フィールドに値 H'50 を記述する必要があります。

(1) コマンドトランスポートについて

コマンドトランスポートはホスト PC がファンクションにバルクアウト転送を用いてコマンドを送ります。このコマンドパッケージが Command Block Wrapper (CBW) として定義されており、Bulk-Only Transport は必ず CBW から始まります。

CBW は、ホスト PC からバルクアウト転送を使用して 31 バイト長のパッケージで送られて来ます。

内容は表 12 に示すフォーマットで送られます。

表 12 コマンドトランスポートフォーマット

	7	6	5	4	3	2	1	0
H'00-H'03	dCBWSignature							
H'04-H'07	dCBWTag							
H'08-H'0B	dCBWDataTransferLength							
H'0C	bmCBWFlags							
H'0D	リザーブ (0)				bCBWLUN			
H'0E	リザーブ (0)				bCBWCBLength			
H'0F-H'1E	CBWCB							

各フィールドの内容を記します。

- dCBWSignature:
データパッケージが CBW であると認知するためのフィールド。値は H'43425355 (リトルエンディアン) です。
- dCBWTag:
コマンドブロックタグ。CBW と対応する CSW を結びつけるために存在し、ホスト PC が指定します。
- dCBWDataTransferLength:
データトランスポートの予定データ長。ここが 0 の場合データトランスポートは存在しません。
- bmCBWFlags:
このフィールドのビットは、ビット 7 が 0 の場合、データトランスポートはバルクアウト転送で行われ、1 の場合、バルクイン転送で行われます。ビット 6~0 は 0 固定です。
- bCBWLUN:
コマンドブロックが送られている装置の論理ユニット番号 (Logical Unit Number)。
- bCBWCBLength:
次の CBWCB フィールドの有効バイト数を表します。
- CBWCB:
ファンクションによって実行されるコマンドブロックを格納するフィールド。このフィールドにホスト PC が実行したいコマンド (本サンプルプログラムでは SCSI コマンド) が入ります。

(2) ステータストランスポートについて

ステータストランスポートはファンクションがホスト PC にバルクイン転送を用いてコマンド実行結果を送ります。

このステータスパケットが Command Status Wrapper (CSW) として定義されており、Bulk-Only Transport は必ず CSW で終わります。

CSW は、ホスト PC へバルクイン転送を使用して 13 バイト長のパケットで送ります。

内容は表 13 に示すステータストランスポートフォーマットで送られます。

表 13 ステータストランスポートフォーマット

	7	6	5	4	3	2	1	0
H'0-H'3	dCSWSignature							
H'4-H'7	dCSWTag							
H'8-H'B	dCSWDataResidue							
H'C	bCSWStatus							

各フィールドの内容を記します。

- dCSWSignature:
データパケットが CSW であると認知するためのフィールド。値は H'53425355 (リトルエンディアン) です。
- dCSWTag:
コマンドブロックタグ。CBW に CSW を結びつけるために存在し、CBW の dCBWTag フィールドと同じ値が入ります。
- dCSWDataResidue:
CBW の dCBWDataTransferLength 値と実際にファンクションが処理したデータ量の相違を報告します。
- bCSWStatus:
コマンドの成功あるいは失敗を示します。コマンドが正常に完了した場合、ファンクションはこのフィールドを H'00 にセットします。ゼロ以外の値は次の通りとし、コマンド実行時の不具合を示します。コマンドフェイルは H'01、フェーズエラーは H'02。

(3) データトランスポートについて

データトランスポートは、ホスト PC とファンクション間のデータ転送を行うトランスポートです。たとえば、Read/Write コマンドでは、データトランスポートにてストレージ各セクタの実データを送信します。

データトランスポートは複数のパストランザクションで構成されます。

データトランスポートで行われるデータ転送はバルクアウト転送かバルクイン転送のどちらか一方です。どちらになるかは CBW データの bmCBWFlags フィールドで決定されます。

(a) データトランスポート (バルクアウト転送) について

データトランスポートがバルクアウト転送の場合について説明します。

この状態になるのは、CBW データの bmCBWFlags フィールドのビット 7 が 0 であり、CBW データの dCBWDataTransferLength フィールドが 0 ではない場合です。

ここでは CBW データの dCBWDataTransferLength フィールドで予定した長さのデータをファンクションが受信します。転送されるデータは、CBW データの CBWCB フィールドで指定された SCSI コマンドを実行する際に必要なデータです。

(b) データトランスポート (バルクイン転送) について

データトランスポートがバルクイン転送の場合について説明します。

この状態になるのは、CBW データの bmCBWFlags フィールドのビット 7 が 1 であり、CBW データの dCBWDataTransferLength フィールドが 0 ではない場合です。

ここでは CBW データの dCBWDataTransferLength フィールドで予定した長さのデータをファンクションがホスト PC に送信します。転送されるデータは、CBW データの CBWCB フィールドで指定された SCSI コマンドを実行した結果のデータです。

3.2.4 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 Mass Storage Class Bulk-Only Transport をデータ転送プロトコルとして使用する場合にサポートしなければならないコマンドは 2 種類あります。表 14 にクラスコマンドを示します。

表 14 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
255 (H'FF)	Bulk-Only Mass Storage Reset	インタフェースをリセットする
254 (H'FE)	Get Max LUN	サポートする LUN の数を調べる

Bulk-Only Mass Storage Reset コマンドを受信した場合、ファンクションは USB Mass Storage Class Bulk-Only Transport で使用するすべてのインタフェースをリセットします。

Get Max LUN コマンドを受信した場合、ファンクションは使用できる最大の論理ユニット番号を返答します。当サンプルシステムの場合、論理ユニットは 1 つなので返答値は 0 をホストに返答します。

3.2.5 サブクラスコード (SCSI transparent command set) について

ファンクションはホスト PC より送信される CBW 内サブクラスコマンドに対応し、各コマンドを処理する必要があります。

本サンプルプログラムでは、SCSI コマンドの中から表 15 に示す 11 個のコマンドをサポートしています。また、未サポートのコマンドについては、ホスト PC に対し CSW を使用し「コマンドフェイルである」と報告しています。

表 15 サポートコマンド一覧

Operation Code	コマンド名	コマンドの動作
H'00	TEST UNIT READY	メディアが使用可能か否かを調べる
H'03	REQUEST SENSE	前のコマンドでエラーが発生したときどのようなエラーが発生したかをホストに伝える
H'12	INQUIRY	ドライブに関する情報をホストに伝える
H'1A	MODE SENSE (6)	ドライブの状態をホストに伝える
H'1B	STOP/START UNIT	メディアの着脱を制御します
H'1E	PREVENT ALLOW MEDIUM REMOVAL	メディアの着脱を禁止 / 許可します
H'23	READ FORMAT CAPACITY	メディアのフォーマット情報をホストに伝える
H'25	READ CAPACITY	メディアのセクタに関する情報をホストに伝える
H'28	READ (10)	指定された読み出しセクタから、指定セクタ量のデータを読み出す
H'2A	WRITE (10)	指定された書き込みセクタから、指定セクタ量のデータを書き込む
H'2F	VERIFY (10)	メディア上のデータにアクセス可能を確かめる

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 エレクトロニクス製
- E10A-USB Emulator ルネサス エレクトロニクス製
- 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 エレクトロニクス製

4.1 ハードウェア環境

図 7 に各デバイスの接続形態を示します。

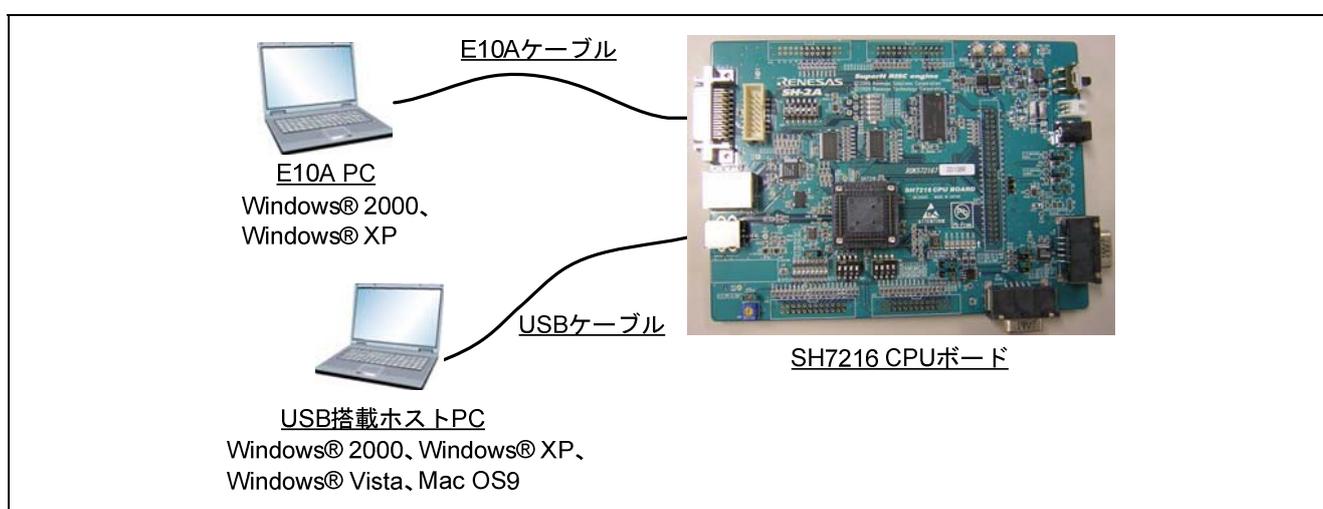


図 7 デバイスの接続形態

(1) SH7216 CPU ボード

本システムでは内蔵 ROM と SDRAM を使用するため、SH7216 CPU ボードを MCU 拡張モード（内蔵 ROM と SDRAM 有効）で動作させる必要があります。そこで、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 16 に示すように変更する必要があります。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 16 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 クラスと USB マスストレージクラス（Bulk-Only Transport）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 USB_MULTI.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 sh7216_usb_multi 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構成を図 8 に示します。

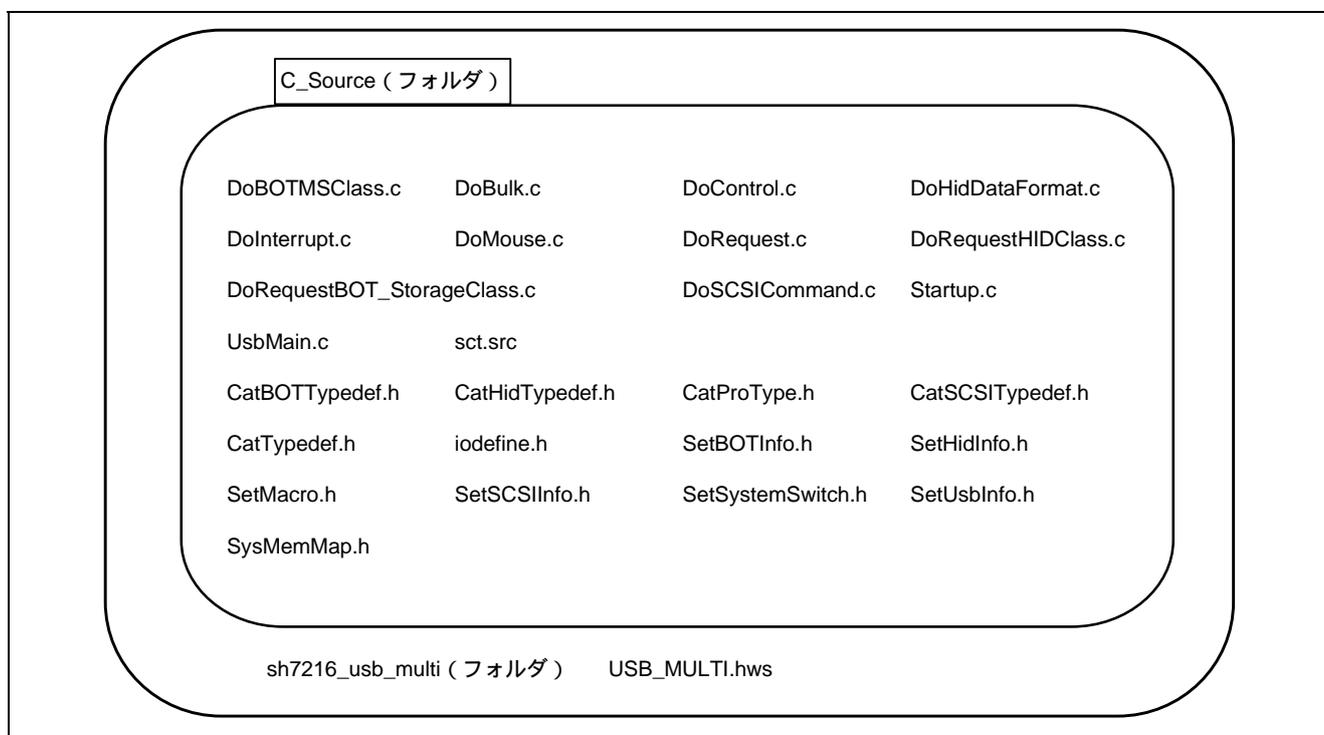


図 8 ファイル構成

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_multi フォルダ内の USB_MULTI.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 USB マルチファンクション機能 の実行方法

プログラムを実行した状態で、USB ケーブルのシリーズ B コネクタを SH7216 CPU ボードに挿入し、反対側のシリーズ A コネクタを USB ホスト PC に接続します。

本サンプルプログラムでは、USB マルチファンクション機能により USB HID クラスによるホスト PC のマウスポインタ移動デモンストレーションと USB マスストレージクラスによる USB 大容量記憶装置デバイス表示を実行します。

4.4.1 マウスポインタ移動デモンストレーション

ホスト PC は、接続後のコントロール転送により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、SH7216CPU ボードをマウスデバイスとして認識します。以降、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。

4.4.2 USB 大容量記憶装置デバイス表示

ホスト PC は、コントロール転送およびバルク転送を用いたエニュミレーション終了後、デバイスマネージャの USB コントローラの下に USB 大容量記憶装置デバイスが表示され、SH7216 CPU ボードを記憶デバイスとして認識します。ディスクドライブの下に RENESAS EX RAM Disk USB Device が表示され、マイコンピュータの中にローカルディスクがマウントされます。

次にローカルディスクをフォーマットします。

ローカルディスクを選択し、マウスの右ボタンをクリックし、フローティングメニュー内のフォーマットを選択します。ドライブのフォーマット選択ウインドが開かれるので、フォーマットの設定を行います。ファイルシステム選択項目が FAT であることを確認し、開始ボタンをクリックしてください。

フォーマットの実行確認画面が出力されるので、OK ボタンをクリックしてください。

フォーマットが完了するとフォーマット完了のメッセージウインドウが出力されるので、OK ボタンをクリックしてください。

ドライブのフォーマット選択ウインドに戻るので、閉じるボタンをクリックしてウインドを閉じてください。

以上で SH7216 CPU ボードを USB 接続の RAM Disk として使用できます。

5. サンプルプログラムの概要

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 ボード上で動作し、USB マスストレージクラスと HID クラスの両方を実行するファームウェアです。USB 転送は USB ファンクションモジュールからの割り込みによって開始します。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、USBRXIO、USBTXIO、USBRXII、USBTXII 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 9 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 9 のように 4 つの状態に遷移します。

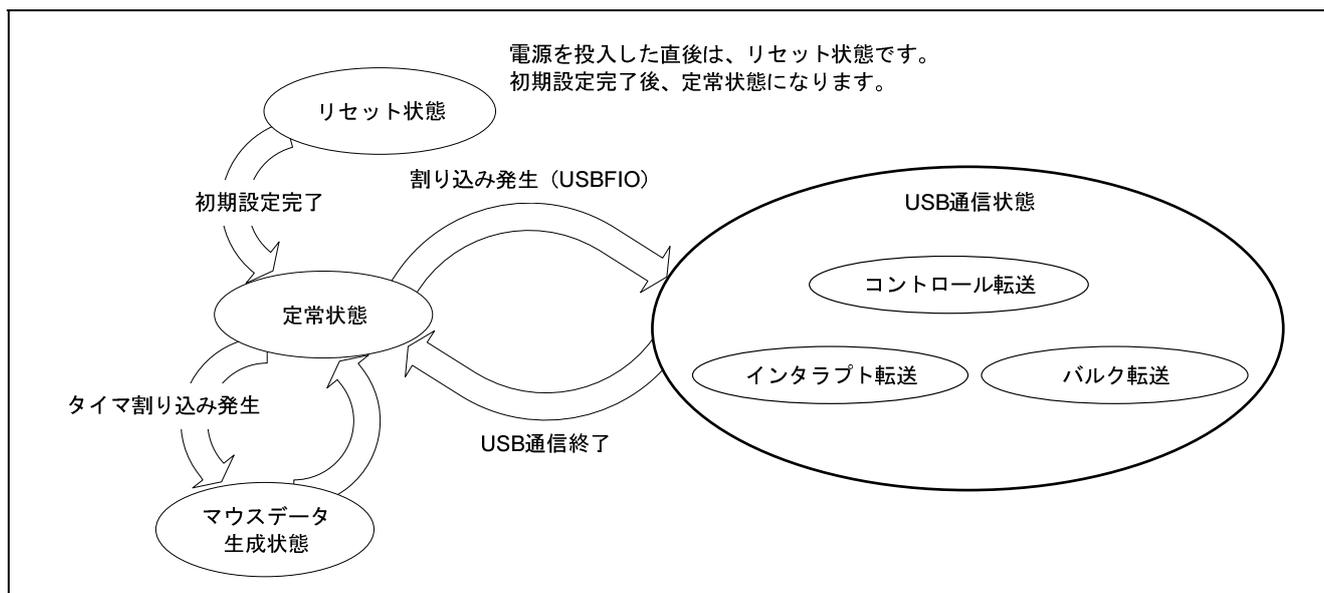


図 9 状態遷移図

- 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3つの状態に分類することができます。割込みが発生すると、まず USB 通信状態へと遷移しさらに割込みの種類に応じた各転送状態へ分岐します。

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 インタラプト転送について

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。

5.2.3 バルク転送について

バルク転送は時間的制約がない大量のデータを、エラーなく転送する場合に使用します。データの転送速度は保証されませんが、データの内容は保証されます。USB マスストレージクラス (Bulk-Only Transport) ではバルク転送を使用し、ホスト PC とファンクション間でストレージデータを転送します。

USB マスストレージクラス (Bulk-Only Transport) の一連の転送処理 (リードやライトなど) は、2 または 3 つのステージから構成されます。USB マスストレージクラス (Bulk-Only Transport) のステージは「コマンドトランスポート (CBW)」「データトランスポート」「ステータストランスポート (CSW)」に分類することができます。

5.3 ファイル構成

本サンプルプログラムのファイル構成を表 17 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 17 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
DoRequestBOT_StorageClass.c	USB マスストレージクラス (Bulk-Only Transport) クラスコマンドの処理
DoBulk.c	バルク転送を実行
DoBOTMSClass.c	USB マスストレージクラス (Bulk-Only Transport) のトランスポートを実行
DoSCSICommand.c	SCSI コマンドの解析および処理
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
CatBOTTypedef.h	Bulk-Only Transport 用構造体定義
CatProType.h	プロトタイプ宣言
CatSCSITypedef.h	SCSI 用構造体定義、FAT 情報構成のためのマクロ定義
CatTypedef.h	USB ファームウェアで使用する基本の構造体定義
SetBOTInfo.h	Bulk-Only Transport 対応に必要な変数の初期設定
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
SetSCSIInfo.h	SCSI 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 18～表 29 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、バルク転送に使用する RAM 領域のクリアを行います。

表 18 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
	InitSDRAM	SH7216 CPU ボードに搭載されている SDRAM の初期設定を行う
	_INIT_SCT	初期値がある変数を、RAM のワークエリアにコピーする
	InitMemory	バルク通信で使用する RAM 領域をクリアする
	InitSystem	USB パ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 19 UsbMain.c

格納ファイル	関数名	機能
UsbMain.c	BranchOfInt0	バスリセット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 マスストレージ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 マスストレージ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 (本サンプルアプリケーションでは使用しません)
	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 19 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 21 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 22 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 23 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 24 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 25 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。

- DoRequestBOT_StorageClass.c
USB マスストレージクラス(Bulk-Only Transport)コマンド(Bulk-Only Mass Storage Reset と Get Max LUN)に応じた処理を行います。
Bulk-Only Mass Storage Reset コマンドは Bulk-Only Transport で使用しているすべてのインタフェースをリセットします。
Get Max LUN コマンドは周辺装置が使用する最大の論理ユニット番号を返答します。当サンプルシステムの場合、論理ユニットは 1 つなので返答値は 0 をホストに返答します。

表 26 DoRequestBOT_StorageClass.c

格納ファイル	関数名	機能
DoRequestBOT_StorageClass.c	DecBOTClassCommands	USB マスストレージクラス (Bulk-Only Transport) コマンドの対応を行う

- DoBulk.c
バルク転送に関する処理を行います。USB マスストレージクラス (Bulk-Only Transport) では ActBulkInReady を使用しません。

表 27 DoBulk.c

格納ファイル	関数名	機能
DoBulk.c	ActBulkOut	バルクアウト転送を行う
	ActBulkIn	バルクイン転送を行う
	ActBulkInReady	バルクイン転送の準備を行う

- DoBOTMSClass.c
DoBOTMSClass.c では、USB マスストレージクラス (Bulk-Only Transport) の 2 ないし 3 つのステージ制御と仕様に従った動作を行います。

表 28 DoBOTMSClass.c

格納ファイル	関数名	機能
DoBOTMSClass.c	ActBulkOnly	Bulk-Only Transport のステージ別に振り分けを行う
	ActBulkOnlyCommand	Bulk-Only Transport の CBW の制御を行う
	ActBulkOnlyIn	(データステージがイン方向の転送) のデータトランスポートとステータストランスポートの制御を行う
	ActBulkOnlyOut	(データステージがアウト方向の転送) のデータトランスポートとステータストランスポートの制御を行う

- DoSCSICommand.c
ホスト PC から送られてきた SCSI コマンドを解析し、次のデータトランスポートまたはステータストランスポートの準備を行います。

表 29 DoSCSICommand.c

格納ファイル	関数名	機能
DoSCSICommand.c	DecBotCmd	ホストから Bulk-Only Transport で送られる SCSI コマンドの対応を行う
	SetBotCmdErr	SCSI コマンドのエラー時の処理を行う

5.4.1 セクション設定

表 30 に SH7216 用サンプルプログラムのセクション情報を示します。

表 30 SH7216 用サンプルプログラムのセクション情報

No.	アドレス	セクション名	内容
1	0x00000000	CStart	ベクタテーブル
2	0x00000400	P	プログラム領域
3		PURAM	CPG 設定プログラム領域
4		C	定数領域
5		D	初期化データ領域
6	0xFFFF80000	RPURAM	CPG 設定プログラム領域 (RAM 配置用)
7	0xFFFF80400	B	未初期化データ領域
8		R	初期化データ領域 (RAM 配置用)
9	0xFFFF8FC00	S	スタック領域

6. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A、SH2A-FPU ソフトウェアマニュアル (RJJ09B0086)
(最新版をルネサス エレクトロニクス 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
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 エレクトロニクス ホームページから入手してください。)
- SH7216CPU ボード ユーザーズマニュアル
SH7216CPU ボード R0K572167C001BR ユーザーズマニュアル Rev.0.03
- USB 規格関連
 - (1) Universal Serial Bus Specification, Revision 2.0
 - (2) Universal Serial Bus Mass Storage Class Specification Overview
 - (3) Universal Serial Bus Mass Storage Class (Bulk-Only Transport)
 - (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 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	2010.12.09	—	初版発行
1.10	2011.03.17	—	FRQCR 設定後のリードを追加

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認ください。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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て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。



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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■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：<http://japan.renesas.com/inquiry>