

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事業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。

H8S/2218 USB ファンクションモジュール Human Interface Devices (HID) Class

アプリケーションノート

ルネサス16ビットシングルチップマイクロコンピュータ
H8S ファミリ/H8S/2200 シリーズ

ご注意

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ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ルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾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の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 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全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりますとは、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<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。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位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。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

はじめに

本アプリケーションノートは、H8S/2218 内蔵の USB ファンクションモジュール HID クラスについて説明したものであり、お客様が USB ファンクションモジュールファームウェア作成の際に、御参考として役立てて頂ける様にまとめました。本アプリケーションノートでは HID クラスの通信を例に、H8S/2218 内蔵 USB ファンクションモジュールの構成を説明します。本アプリケーションノートの内容およびソフトウェアは、USB ファンクションモジュールの使用例として説明しているものであり、その内容を保証するものではありません。

また、開発に際しましては、本書のほか以下に関連マニュアルもあわせて御覧ください。

【関連マニュアル】

- Universal Serial Bus Specification Revision 1.1
- Universal Serial Bus Device Class Definition for Human Interface Devices(HID)
- H8S/2218グループ、H8S/2212グループ ハードウェアマニュアル
- H8S/2218 Solution Engine CPUボード (MS2218CP01) 取扱説明書

【注】 本資料に記載してあるサンプルプログラムでは、USB の転送タイプのうち「バルク」に関するファームウェアは準備しておりません。「バルク」(H8S/2218 グループ、H8S/2212 グループ ハードウェアマニュアル 14.5.6～7 参照)の転送タイプを御使用になる場合は、別途お客様でプログラムを作成していただく必要があります。

また、本アプリケーションノートには、上記システムの開発時に必要と思われる H8S/2218、H8S/2218 Solution Engine のハードウェア仕様を記載してありますが、詳細は H8S/2218 グループ、H8S/2212 グループ ハードウェアマニュアル、ならびに H8S/2218 Solution Engine の取扱説明書を御覧ください。

【商標】 Microsoft Windows® 95、Microsoft Windows® 98、Microsoft Windows® Me、Microsoft Windows® 2000、Microsoft Windows® XP は、米国 Microsoft Corp.の米国およびその他の国における登録商標です。

目次

1. 概要	1-1
2. Human Interface Devices (HID) Class の概要	2-1
2.1 HID Classについて	2-1
2.2 サブクラスコードについて	2-1
2.3 プロトコルコードについて	2-1
2.4 HID Classのディスクリプタについて	2-2
2.5 HIDディスクリプタについて	2-2
2.6 リポートディスクリプタについて	2-3
2.6.1 Main items	2-4
2.6.2 Global items	2-6
2.6.3 Local items	2-7
2.6.4 リポートディスクリプタの例	2-8
2.6.5 リポートディスクリプタの説明	2-9
2.7 フィジカルディスクリプタについて	2-10
2.8 HIDデータの転送フォーマットについて	2-10
2.9 クラスコマンドについて	2-11
3. 開発環境	3-1
3.1 ハードウェア環境	3-2
3.2 ソフトウェア環境	3-3
3.2.1 サンプルプログラム	3-3
3.2.2 コンパイルおよびリンク	3-4
3.3 プログラムのロードと実行方法	3-5
3.3.1 プログラムのロード	3-6
3.3.2 プログラムの実行	3-6
3.4 擬似マウス操作（カーソル移動）の実行方法	3-6
4. サンプルプログラム概要	4-1
4.1 状態遷移	4-1
4.2 USB通信状態	4-2
4.3 ファイル構成	4-3
4.4 関数の機能	4-4

5.	サンプルプログラムの動作.....	5-1
5.1	メインループ.....	5-1
5.2	割り込みの種類.....	5-2
5.2.1	各転送への分岐方法.....	5-4
5.3	USB動作クロック安定割り込み.....	5-5
5.4	ケーブル接続時（VBUS）割り込み.....	5-6
5.5	バスリセット時（BRST）割り込み.....	5-7
5.6	コントロール転送.....	5-8
5.6.1	セットアップステージ.....	5-9
5.6.2	データステージ.....	5-11
5.6.3	ステータスステージ.....	5-13
5.7	インタラプト転送.....	5-15
5.7.1	インタラプトイン転送.....	5-15
5.8	マウスデータの生成.....	5-17
6.	アナライザのデータ.....	6-1
6.1	デバイス接続時のコントロール転送.....	6-1
6.2	HIDデータのインタラプトイン転送.....	6-6

1. 概要

本アプリケーションノートは、H8S/2218 の USB ファンクションモジュールの使用法、およびファームウェアの作成例について説明したものです。

H8S/2218 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。

- USB1.1に準拠したUDC (USB Device Controller) を内蔵
- 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。)
- フルスピード (12Mbps) 転送に対応
- USB送受信に必要な各種割り込み信号を生成
- バストランシーバを内蔵

エンドポイントの構成

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 Byte	8 Byte	
	EP0i	コントロールイン	64 Byte	64 Byte	
	EP0o	コントロールアウト	64 Byte	64 Byte	
エンドポイント 1	EP1	バルクイン	64 Byte	64 × 2 (128 Byte)	可能
エンドポイント 2	EP2	バルクアウト	64 Byte	64 × 2 (128 Byte)	可能
エンドポイント 3	EP3	インタラプト (イン)	64 Byte	64 Byte	

1. 概要

システム構成例を図1.1に示します。

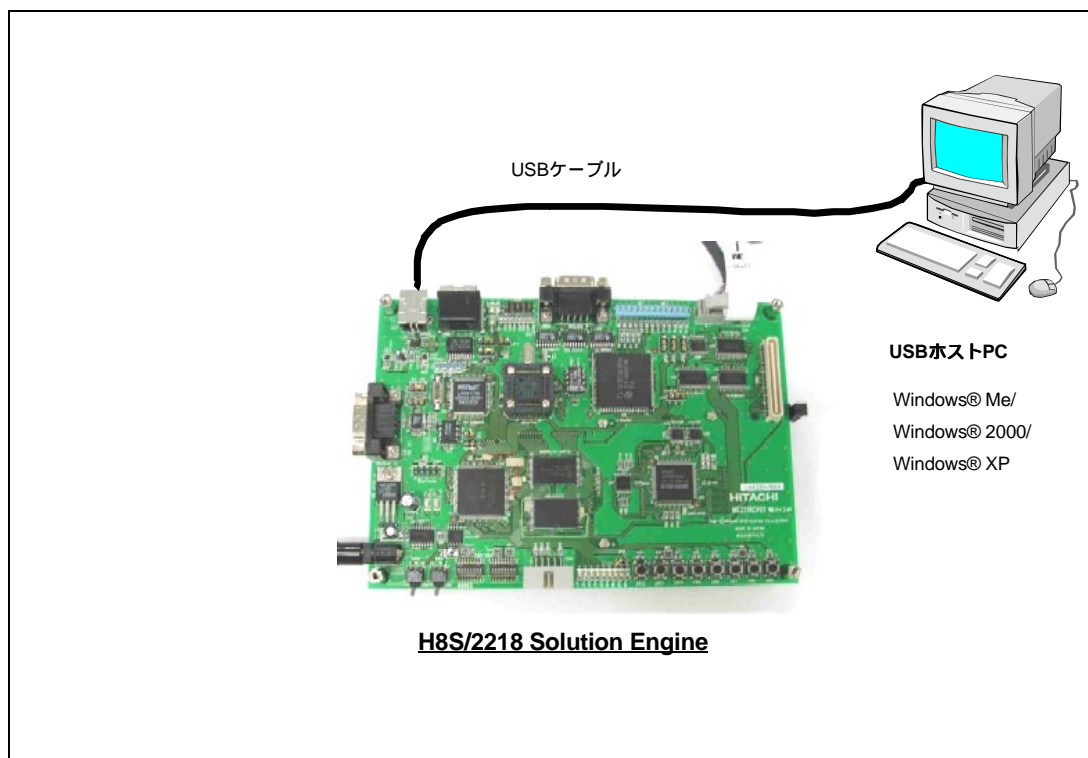


図 1.1 システム構成

本システムは、H8S/2218を搭載した日立超 LSI システムズ社製の H8S/2218 Solution Engine (以下 MS2218CP)、Windows® Me/ Windows® 2000/ Windows® XP を OS とした PC によって構成されています。

本システムは、MS2218CP 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、Windows® Me/ Windows® 2000/ Windows® XP 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。

- サンプルプログラムにより、H8S/2218のUSBモジュールを短期間で評価可能
- サンプルプログラムは、USBのコントロール転送、インタラプト転送をサポート
- E10Aを使用することができ、効率的なデバッグが可能

2. Human Interface Devices (HID) Class の概要

この章では、Human Interface Devices (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際に、ご参考としてお使いください。なお、規格の詳細につきましては、「Device Class Definition for Human Interface Devices (HID) Version 1.11」「HID Usage Tables Version 1.11」をご覧ください。

2.1 HID Class について

USB HID Class 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 0x03 を記述する必要があります。

2.2 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 0x01 を記述する必要があります。

2.3 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 : 0x01) と "マウス" (値 : 0x02) 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必要があります。

2.4 HID Class 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」と「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 2.1 に示します。

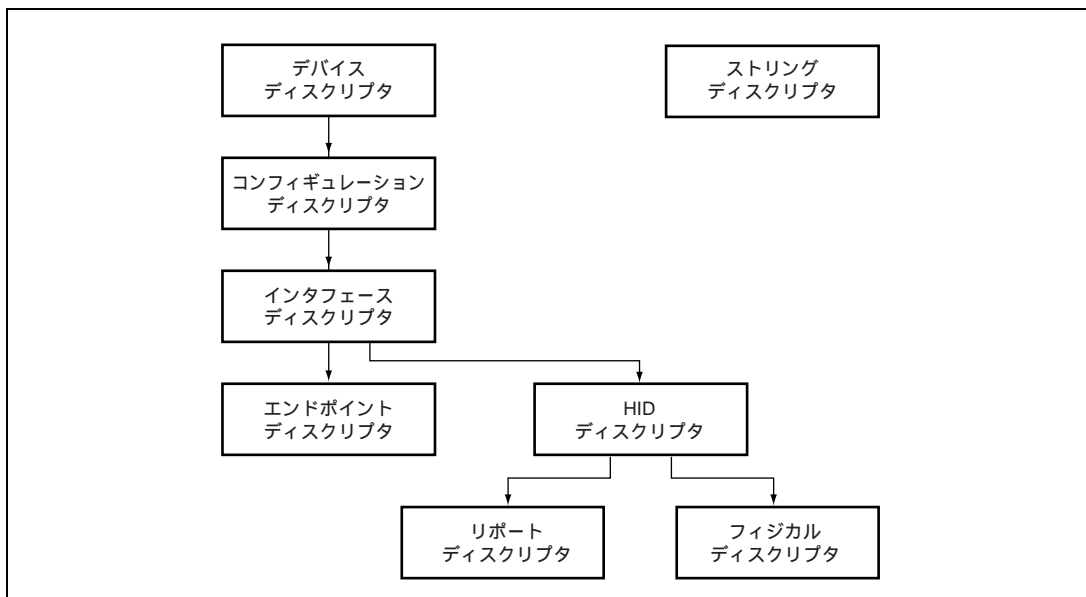


図 2.1 ディスクリプタ構成

2.5 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は表 2.1 に示すとおりです。

表 2.1 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(0x09で固定)
bDescriptorType	1	ディスクリプタのタイプ(0x21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0x22)
wDescriptorLength	2	レポートディスクリプタのサイズ

2.6 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。

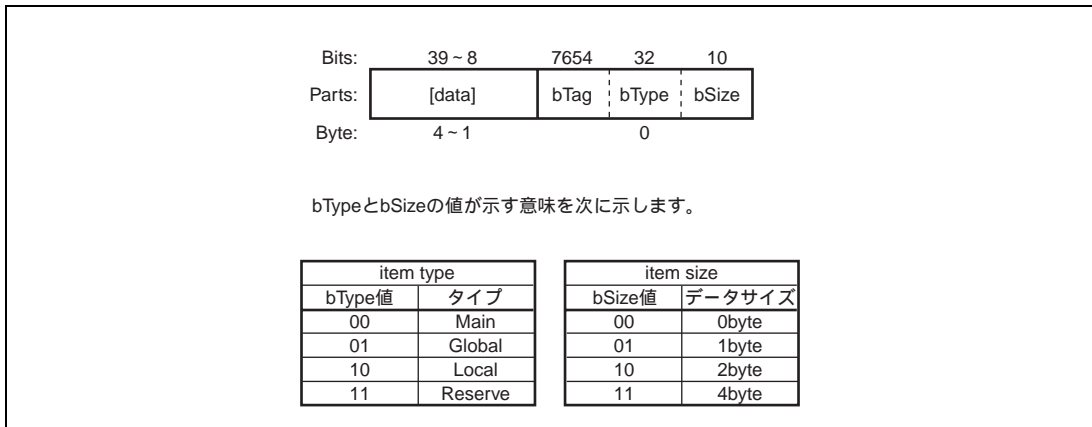


図 2.2 リポートディスクリプタアイテム

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。

2. Human Interface Devices (HID) Class の概要

2.6.1 Main items

Main item type の item tag 5 種類を表 2.2 に示します。

表 2.2 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 種類あり各 1Bit ごとに指定します。内容を表 2.3 に示します。

表 2.3 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。
	1	Constant	アイテムは定数を報告。
1	0	Array	アイテムはアレイデータフィールドを報告。
	1	Variable	アイテムは変数を報告。
2	0	Absolute	アイテムは絶対値を報告。
	1	Relative	アイテムは最後の報告からの偏差を報告。
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します。
	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ません。
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 及び 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 2.4 に Output 及び Feature item tag のパラメータを示します。

表 2.4 Output 及び 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 1Byte の値で指定します。内容を表 2.5 に示します。

表 2.5 Collection item tag のパラメータ

値	内容	意 味
0x00	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等で、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。
0x01	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。
0x02	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。
0x03	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。
0x04	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。
0x05	Usage Switch	これは、これが含まれている 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。
0x06	Usage Modifier	これは、包含するコレクションに付けられた usag の意味を修正します。usag 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。
0x07-7F	Reserved	リザーブ。
0x80-FF	Vendor-defined.	ベンダ定義。

2. Human Interface Devices (HID) Class の概要

2.6.2 Global items

Global item type の item tag 12 種類を表 2.6 に示します。

表 2.6 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	0101	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。

2.6.3 Local items

Local item type の item tag10 種類を表 2.7 に示します。

表 2.7 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 アイテムの始まりか終了を定義します。

2. Human Interface Devices (HID) Class の概要

2.6.4 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 2.3 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 2.3 リポートディスクリプタ

2.6.5 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 2.8 に示します。

表 2.8 リポートディスクリプタ

Item	Value (Hex)	Item 区分	意 味
Usage Page (Generic Desktop Control)	0x05 01	Global	用法ページ (Usage Page) を示す値。0x01 は"Generic Desktop Control"を表します。
Usage (Mouse)	0x09 02	Local	アイテム使用法のインデックス。0x02 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)	0xA1 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)	0x09 01	Local	アイテム使用法のインデックス。0x01 は"Pointer"を表します。"Pointer"の使用法タイプは"Collection Physical"です。
Collection (Physical)	0xA1 00	Main	ポインターとして多数のセンサーのポジション (ボタン、X 軸、Y 軸、ロータリー・コントロール) を 1 つに集めます。
Usage Page (Button)	0x05 09	Global	用法ページ (Usage Page) を示す値。0x09 は"Button"を表します。
Usage Minimum (1)	0x19 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)	0x29 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)	0x15 00	Global	アイテムが報告する最小値は 0 です。
Logical Maximum (1)	0x25 01	Global	アイテムが報告する最大値は 1 です。
Report Count (3)	0x95 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)	0x75 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)	0x81 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)	0x95 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)	0x75 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)	0x81 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)	0x05 01	Global	用法ページ (Usage Page) を示す値。0x01 は"Generic Desktop Control"を表します。
Usage (X)	0x09 30	Local	アイテム使用法のインデックス。0x30 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)	0x09 31	Local	アイテム使用法のインデックス。0x31 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。

2. Human Interface Devices (HID) Class の概要

Item	Value (Hex)	Item 区分	意味
Usage (Wheel)	0x09 38	Local	アイテム使用法のインデックス。0x38 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ー・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)	0x15 81	Global	アイテムが報告する最小値は-127 です。
Logical Maximum(127)	0x25 7F	Global	アイテムが報告する最大値は 127 です。
Report Size (8)	0x75 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)	0x95 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)	0x81 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	0xC0	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	0xC0	Main	データセットとして 1 つに纏める終点を表します。(Application)

2.7 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。

2.8 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出来ます。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 2.4 に示します。



図 2.4 レポートプロトコルフォーマット

2.9 クラスコマンドについて

クラスコマンドとは、USB の各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 Class のクラスコマンドは 6 種類あります。表 2.9 にクラスコマンドを示します。

表 2.9 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
0x01	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
0x02	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
0x03	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リポートプロトコル)を報告する
0x09	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
0x0A	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
0x0B	SET_PROTOCOL	アクティブなプロトコル(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 バイトでリポートタイプを指定し、wValue フィールドの下位 1 バイトでリポート ID を指定します。リポート ID を使用しない場合は値 0 が指定されません。

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。尚、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

2. Human Interface Devices (HID) Class の概要

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。尚、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。

3. 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。本システムの開発には、以下のデバイス（ツール）を使用しました。

- H8S/2218 Solution Engine（以下MS2218CP 型名MS2218CP01）日立超LSIシステムズ社製
- E10Aエミュレータ ルネサステクノロジ製
- PCMCIA（またはPCI/ISA/USB）スロット搭載のPC（Windows® 95/Windows® 98/Windows® Me/Windows® 2000/Windows® XP）
- USBホスト用PC（Windows® Me/Windows® 2000/Windows® XP）
- USBケーブル
- Debugging Interface（以下HDI）ルネサステクノロジ製
- High-performanceEmbedded Workshop（以下HEW）ルネサステクノロジ製

3. 開発環境

3.1 ハードウェア環境

図 3.1 に各デバイスの接続形態を示します。

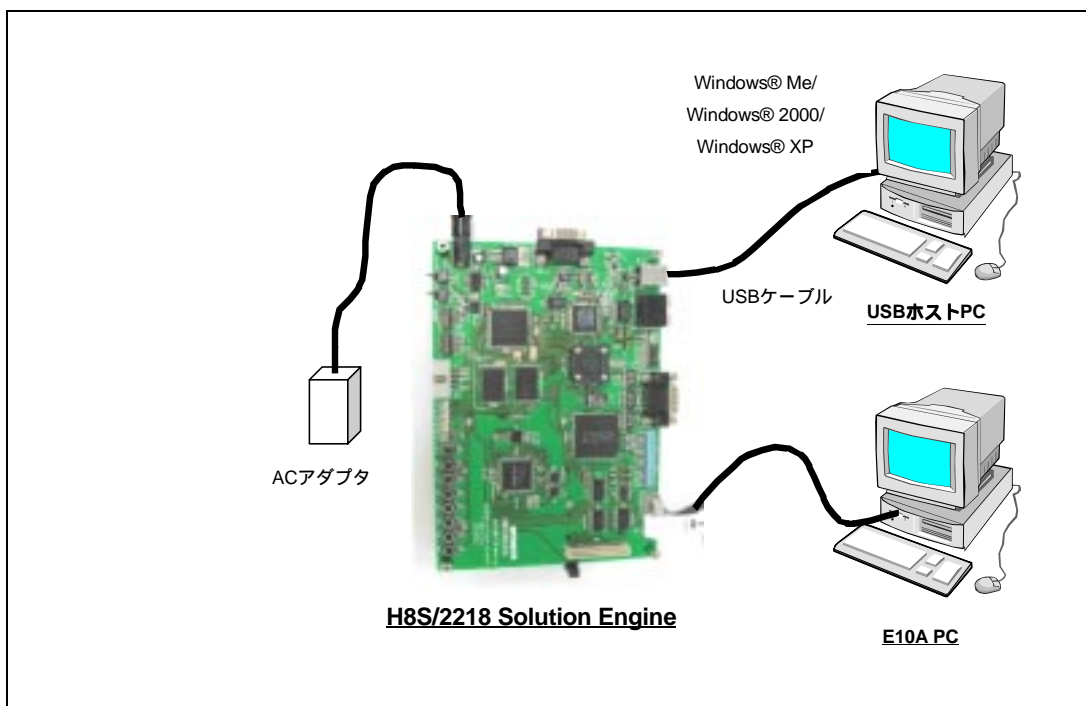


図 3.1 デバイスの接続形態

(1) MS2218CP

MS2218CP ボードのディップスイッチとジャンパピンのいくつかを出荷時の設定から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とジャンパピンを変更する必要はありません。

表 3.1 ディップスイッチとジャンパピンの設定

スイッチ	出荷時	変更後	ディップスイッチとジャンパピンの機能
SW1-1	OFF	ON	選択モード 6 を選択
SW1-2	OFF	OFF	
SW1-3	OFF	OFF	
SW1-5	OFF	ON	E10A エミュレータモードを選択
J-3	CLOSE	OPEN	USB セルフパワーモードを選択
J-9	CLOSE	OPEN	ビッグエンディアンモードを選択

(2) USB ホスト PC

USB ポート搭載の、Windows® Me/Windows® 2000/Windows® XP をインストールしたパソコンをホスト PC として使用します。本システムでは、Windows® Me/Windows® 2000/Windows® XP に標準で搭載されている 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

E10A 用 PC と E10A エミュレータの接続インターフェースは PCMCIA を使用しました。

PC カードスロットに E10A カードを挿入し、接続用のケーブルを介して E10A と MS2218CP を接続してください。接続後、HDI を起動してエミュレーションを行います。

3.2 ソフトウェア環境

サンプルプログラムと、今回使用したコンパイラおよびリンカについて説明します。

3.2.1 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 H8S2218 フォルダ内に収められています。HEW、HDI 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いただくと、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 3.2 に示します。

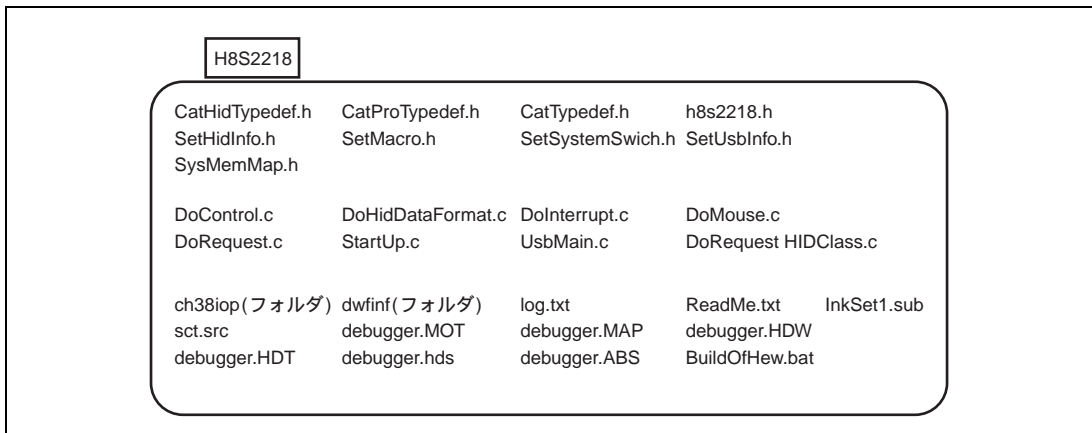


図 3.2 フォルダに含まれるファイル

3. 開発環境

3.2.2 コンパイルおよびリンク

サンプルプログラムのコンパイルおよびリンクは、以下のソフトウェアにより行いました。

Hitachi Embedded Workshop Version1.0 (release9) (以下 HEW)

HEW を C:\Hew にインストールした場合、コンパイルおよびリンクの手順は以下のようになります。

まず、コンパイル時に作業用として Tmp という名前のフォルダを C:\Hew のフォルダ内に作成してください(図 3.3)。

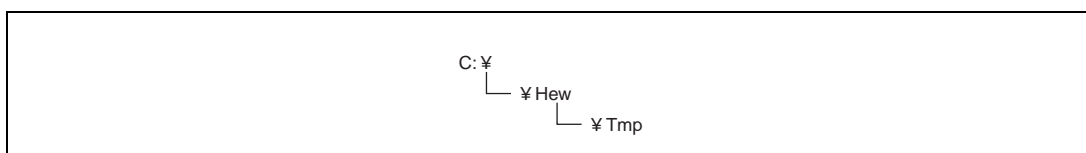


図 3.3 作業フォルダの作成

次に、サンプルプログラムが格納されているフォルダ (H8S2218) を、C:\Usr にコピーしてください(もしくは、任意の場所にコピーし、フォルダ内に含まれている debugger.hds ファイルに記述されている C:\Usr\H8S2218 をコピーしたパスに変更してください)。この中には、サンプルプログラムと共に BuildOfHew.bat というバッチファイルが含まれています。このバッチファイルでは、パスの設定、コンパイルオプションの指定、コンパイルおよびリンク結果を示すログファイルの指定等を行っています。BuildOfHew.bat を実行すると、コンパイルおよびリンクが行われます。その結果、フォルダ内にはファイル名 debugger.MOT のモトローラ S タイプフォーマットファイルが作成されます。これが実行ファイルとなります。このとき同時にマップファイル debugger.MAP とログファイル log.txt が作成されます。マップファイルにはプログラムのサイズ、および変数のアドレスが示されています。コンパイルの結果(エラーの有無等)はログファイルに記録されます。

【注】 * HEW を C:\HEW 以外にインストールした場合、BuildOfHew.bat 内の「コンパイラパスの設定」と「コンパイラが使用する環境変数の設定」、lnkSet1.sub 内の「ライブラリーの指定」を変更する必要があります。この場合、コンパイラパスの設定は ch38.exe のパス、コンパイラが使用する環境変数 ch38 の設定は machine.h のフォルダ、ch38tmp の設定はコンパイル作業フォルダをそれぞれ指定してください。また、ライブラリーの指定は c8s26a.lib のパスを指定してください。

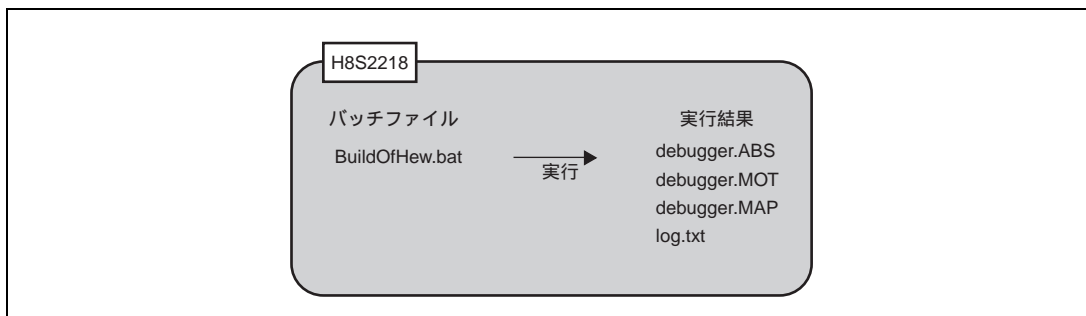


図 3.4 コンパイル結果

3.3 プログラムのロードと実行方法

図 3.5 にサンプルプログラムのメモリマップを示します。

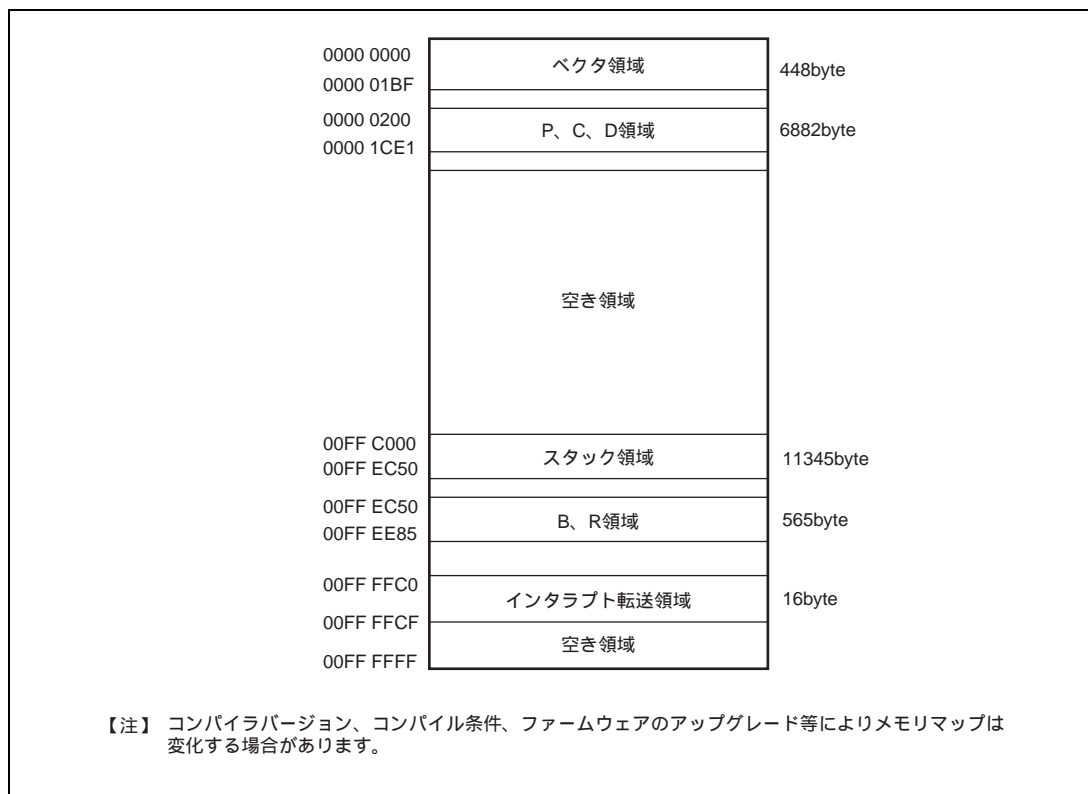


図 3.5 メモリマップ

図 3.5 のように、本サンプルプログラムはベクタ、P、C、D の各領域をエリア 0 内蔵フラッシュメモリ領域上に配置、スタック、B、R を内蔵 RAM 上に配置しています。

これらのメモリへの割り付けは、H8S2218 フォルダ内に含まれる lnkSet1.sub で指定します。プログラム配置を変更する場合は、このファイルを変更してください。

3. 開発環境

3.3.1 プログラムのロード

MS2218CP 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。

1. HDIをインストールしたE10A用PCにE10Aを接続してください。
2. E10AケーブルでE10AとMS2218CPを接続してください。
3. MS2218CPの電源を投入し、起動してください。
4. H8S2218フォルダ内のdebugger.hdsを実行してください。
5. 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（16 or 24MHz）の周波数を入力してください。
6. レジストリの入力を求められるので、0を入力してください。

以上の操作で、E10A が起動します。

3.3.2 プログラムの実行

「3.3.1 プログラムのロード」でロードしたプログラムを実行するためには、プログラムカウンタ（PC）を設定する必要があります。

メニューバーの View Register Window を選択し、Registers ウィンドを開きます。ウィンド内の該当レジスタ(PC)の数値エリアをダブルクリックすると、ダイアログボックスが開きレジスタの値を変更することができます。このダイアログボックスで PC を H'0000 0200 に設定してください。

以上の設定後、メニューバーの Run Go を選択するとプログラムが内蔵フラッシュメモリー上に書き込まれた後、実行されます。

3.4 擬似マウス操作（カーソル移動）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擬似マウス操作(カーソル移動)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 コネクタを MS2218CP に、シリーズ A コネクタを USB ホスト PC に接続します。コントロール転送終了後、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 MS2218CP をマウスデバイスとして認識します。

ホスト PC に接続後、MS2218CP の SW5 ~ 12 を押す事により、マウスの移動データが生成されます。

SW5 を押すとマウスの左ボタンが押されたデータを生成、SW6 を押すとマウスの右ボタンが押されたデータを生成、SW8 を押すとポインタが移動するデモンストレーションを行い、SW9 を押すとポインタが Y 方向下方に移動するデータを生成、SW10 を押すとポインタが Y 方向上方に移動するデータを生成、SW11 を押すとポインタが X 方向右側に移動するデータを生成、SW12 を押すとポインタが X 方向左側に移動するデータを生成し、ホスト PC からのインタラプトイン転送に応じて MS2218CP がマウス擬似データを送出します。その結果、USB ホスト PC 上のカーソルが自動的に移動を開始します。

4. 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 MS2218CP 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データを生成することにより、マウスの動作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。H8S/2218 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、EXIRQ0、EXIRQ1、IRQ6 の 3 種類ですが、本サンプルでは EXIRQ0 のみ使用しています。

本サンプルプログラムの特長を以下に示します。

- コントロール転送を行うことができます。
- インタラプトイン転送でホスト PC にマウス擬似データを送信することができます。

4.1 状態遷移

図 4.1 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 4.1 のように 4 つの状態に遷移します。

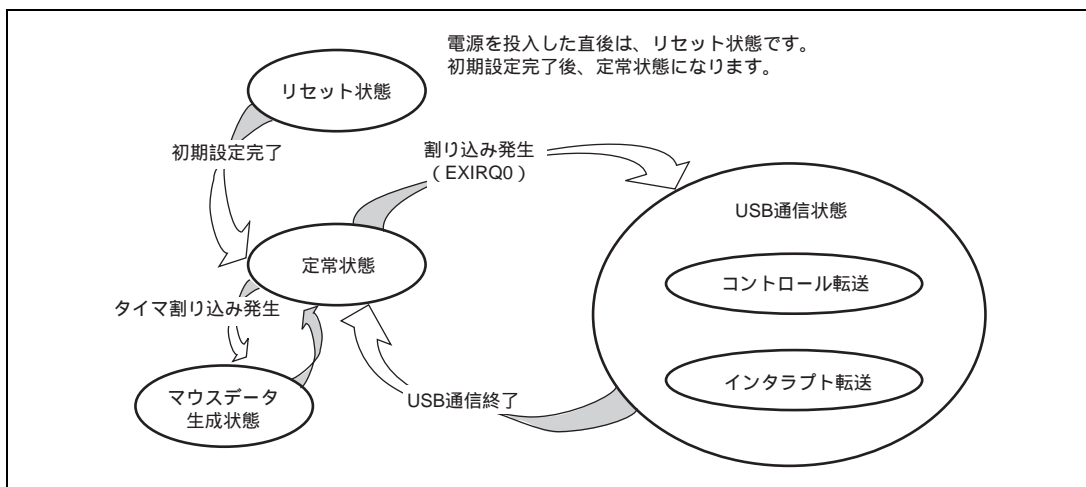


図 4.1 状態遷移図

4. サンプルプログラム概要

- 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H8S/2218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、3 (UIFR0、1、3) によって示され、計9種類です。割り込み要因が発生すると、UIFR0,1,3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TGRA_2のコンペアマッチ割り込みが発生すると、この状態になります。マウスデータ生成状態ではボード上のボタンを使用し、マウスの代わりにマウスデータを生成もしくは、移動データを自動生成します。コンペアマッチ割り込みは16または10ms間隔で発生します。

4.2 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 4.2 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。分岐の方法については「第5章 サンプルプログラムの動作」で説明します。

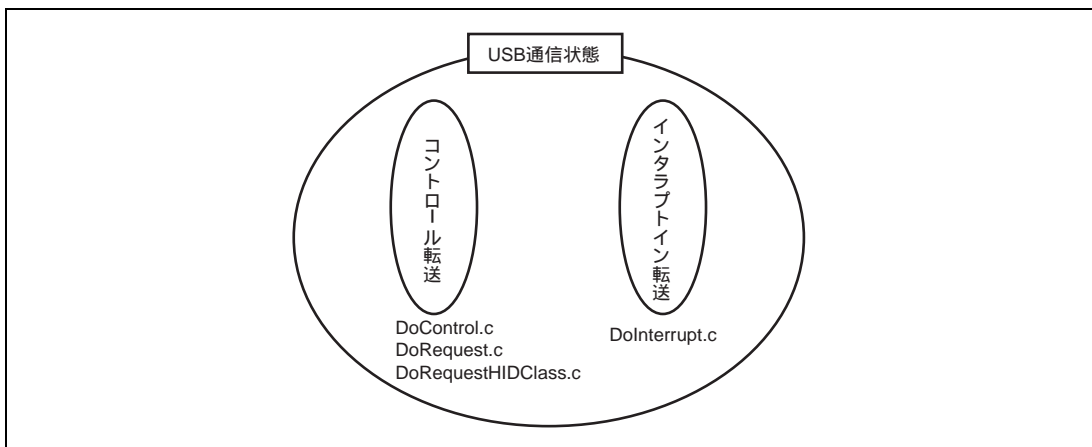


図 4.2 USB 通信状態

4.3 ファイル構成

本サンプルプログラムは、8個のソースファイルと9個のヘッダーファイルで構成されています。全構成ファイルを表4.1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 4.1 ファイル構成

ファイル名	主な役割
StartUp.c	マイコンの初期設定
UsbMain.c	割り込み要因の判定 パケットの送受信
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
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	HID クラスコマンドの処理
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
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
h8s2218.h	H8S/2218 レジスタ定義
SetHidInfo.h	HID クラス対応に必要な変数の初期設定
SetMacro.h	マクロ定義
SetSystemSwitch.h	システムの動作設定
SetUsbInfo.h	USB ファームウェアに必要な変数の初期設定
SysMemMap.h	MS2218CP のアドレス定義

4. サンプルプログラム概要

4.4 関数の機能

表 4.2～表 4.9 に、各ファイルに含まれる関数とその機能を示します。

表 4.2 UsbMain.c

格納ファイル	関数名	機能
UsbMain.c	BranchOfInt	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。(本サンプルプログラムでは使用しません)
	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版高速版。
	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(本サンプルプログラムでは使用しません)
	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版高速版 (本サンプルプログラムでは使用しません)
	SetControlOutContents	ホスト PC から送られたデータを書き換える
	SetUsbModule	USB モジュールの初期設定
	ActBusReset	バスリセット受信時に FIFO のクリアを行う
	ActBusVcc	USB ケーブル接続、切断時に D+プルアップと USB モジュールの制御を行う。
	ConvRealIn	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。	

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 4.3 StartUp.c

格納ファイル	関数名	機 能
StartUp.c	SetPowerOnSection	BSC、端子、割り込み割り込みコントローラの設定、各初期化ルーチン呼び出しを行い、メインループへ移行
	_INITISCT	初期値を持つ変数を、RAMのワークエリアにコピー
	InitMemory	メモリ領域の設定
	InitSystem	USBクロックの設定、システム割り込みマスクの設定、タイマの設定

StartUp.cでは、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H8S/2218の初期設定や、USBクロック等の初期設定を行います。

表 4.4 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

DoRequest.cでは、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045B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 4.5 DoRequestHIDClass.c

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

DoRequestHIDClass.cでは、HID Class コマンド (GET_REPORT, GET_IDLE, GET_PROTOCOL, SET_REPORT, SET_IDLE, SET_PROTOCOL) に応じた処理を行います。

- 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レポートプロトコル)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レポートプロトコル)を設定します。

4. サンプルプログラム概要

表 4.6 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

DoControl.c では、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 4.7 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。

DoInterrupt.c では、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 4.8 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。

DoHidDataFormat.c では、ホスト PC に送信する HID データの送信準備を行う。

表 4.9 DoMouse.c

格納ファイル	関数名	機 能
DoMouse.c	MousePushedDataInput	タイマ割込で起動し、キースキャンを行うか自動でマウス移動データを作成するかを決める。
	MousePushedDataInput1	キースキャンを行い、マウスのデータを作成する
	MousePushedDataInput2	タイマでの時間経過に応じてマウス移動データを生成する。

DoMouse.c では、タイマ割込みを使用してマウスのデータを生成します。

図 4.3 に、表 4.2～4.9 で説明した関数の相関関係を示します。上位側の関数が、下位側の関数を呼び出すことができます。また、複数の関数が同一の関数を呼び出すこともあります。定常状態では、SetPowerOnSection が他の関数を呼び出し、割り込みの発生によって遷移する USB 通信状態では、BranchOfInt が他の関数を呼び出します。図 4.3 は、関数の上下関係を示しているもので、関数が呼び出される順序は示していません。関数がどのような順序で呼び出されるかについては、「第 5 章 サンプルプログラムの動作」のフローチャートをご覧ください。

4. サンプルプログラム概要

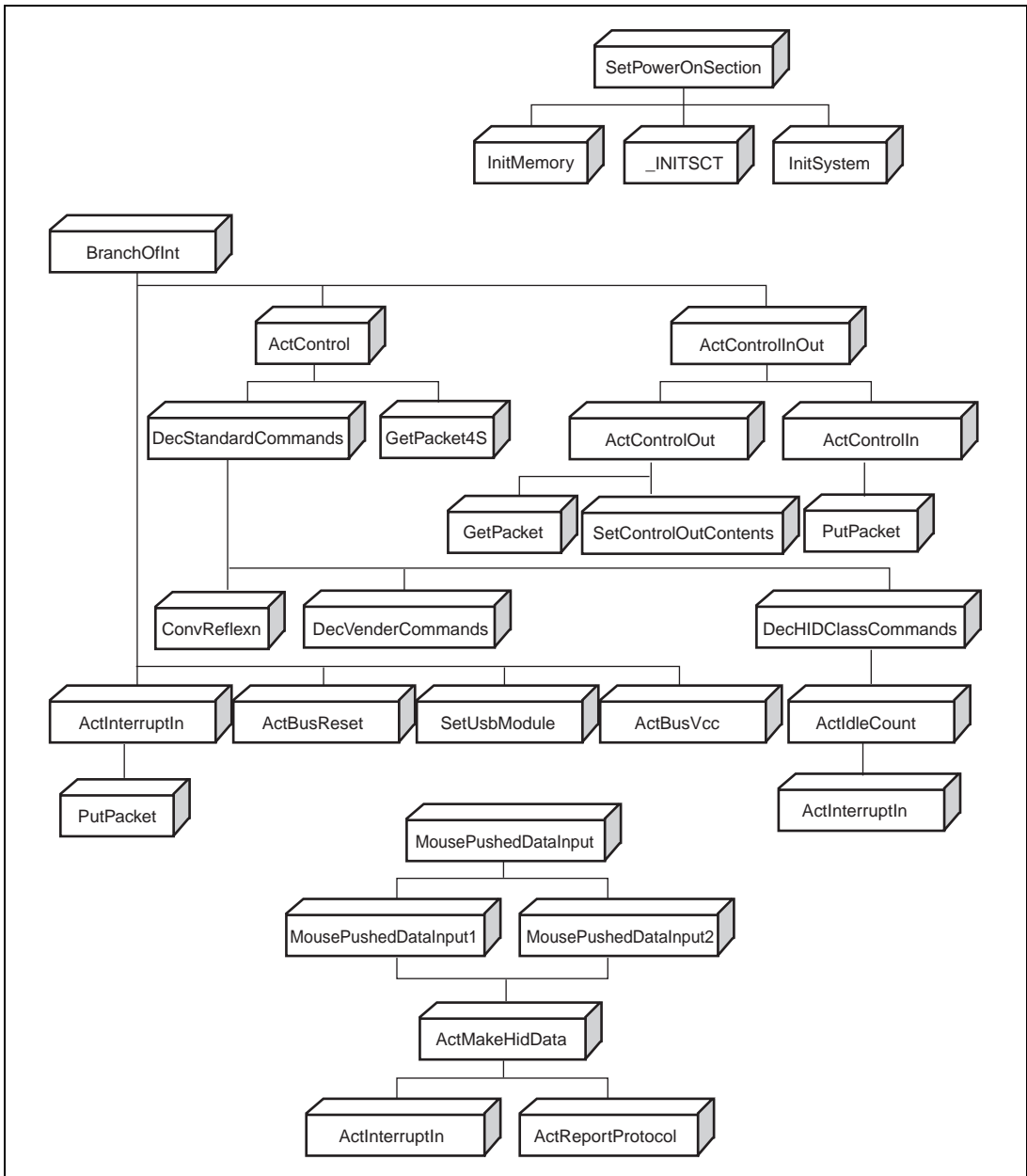


図 4.3 関数の相関関係

5. サンプルプログラムの動作

この章ではサンプルプログラムの動作を、USB ファンクションモジュールの動作と関連付けて説明します。

5.1 メインループ

マイコンがリセット状態になると、CPU の内部状態と内蔵周辺モジュールのレジスタが初期化されます。次に StartUp.c の関数 SetPowerOnSection が呼び出され、CPU の初期化をします。図 5.1 に SetPowerOnSection のフローチャートを示します。

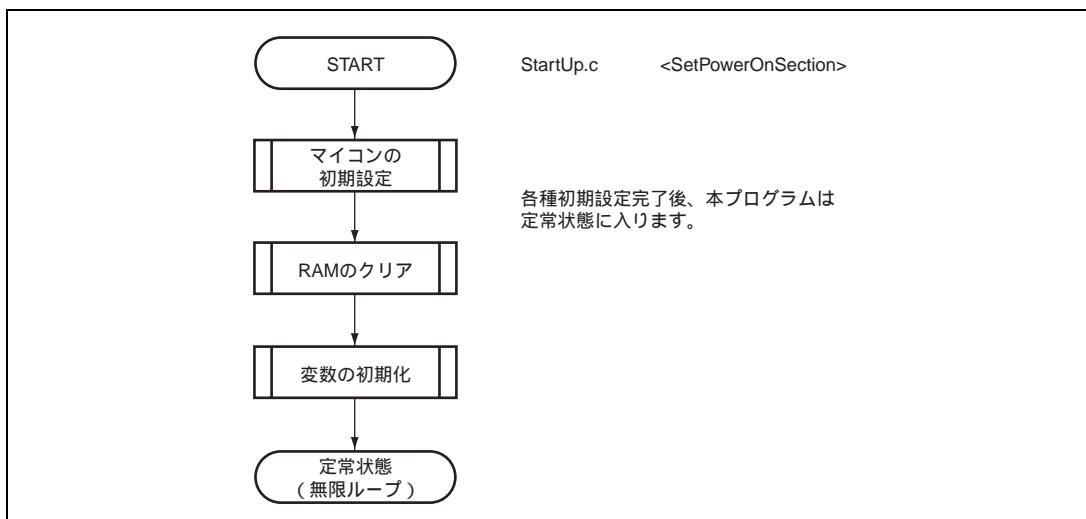


図 5.1 メインループ

5.2 割り込みの種類

4章で説明したように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、3 (UIFR0、1、3) によって示される計9種類です。割り込み要因が発生すると、割り込みフラグレジスタの対応するビットに1がセットされ、CPU に対して EXIRQ0 割り込みを要求します。サンプルプログラムでは、この割り込み要求によって割り込みフラグレジスタをリードし、それに対応する USB 通信を行います。図 5.2 に割り込みフラグレジスタと、USB 通信の関係を示します。

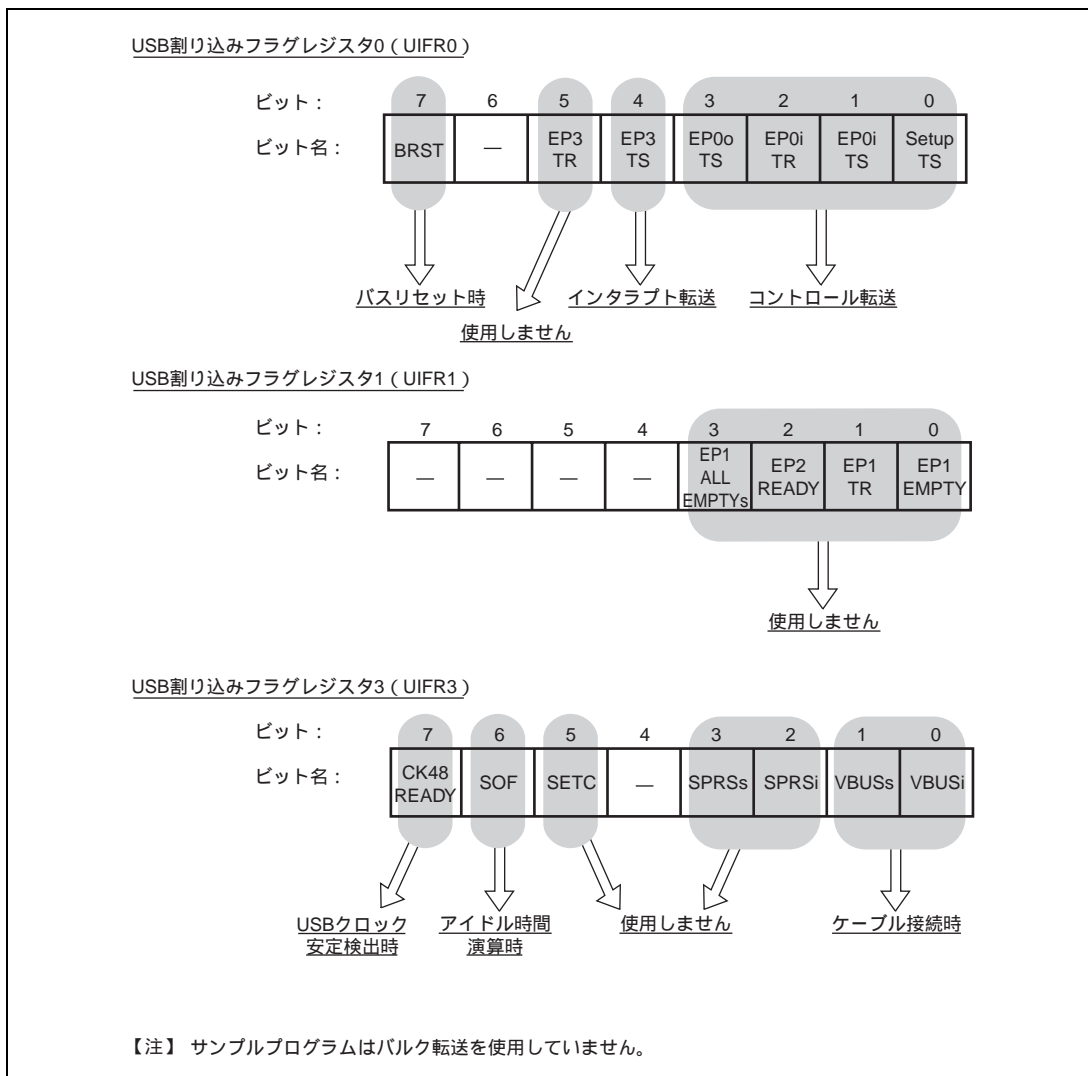


図 5.2 割り込みフラグの種類

5.2.1 各転送への分岐方法

サンプルプログラムでは、USB モジュールからの割り込みの種類によって転送方式を決定しています。各転送方式への分岐は、UsbMain.c の BranchOfInt が実行します。表 5.1 に割り込みの種類と、BranchOfInt が呼び出す関数の関係を示します。

表 5.1 割り込みの種類と分岐先関数

レジスタ名	ビット	ビット名	呼び出す関数名
UIFR0	7	BRST	ActBusReset
	6	-	-
	5	EP3 TR	-
	4	EP3 TS	ActInterruptIn
	3	EP0o TS	ActControlInOut
	2	EP0i TR	ActControlInOut
	1	EP0i TS	ActControlInOut
	0	SETUP TS	ActControl
UIFR3	7	CK48 READY	SetUSBModule
	6	SOF	ActIdleCount
	5	SETC	-
	4	-	-
	3	SPRSs	-
	2	SPRSi	-
	1	VBUSs	-
	0	VBUSi	ActBusVcc

EP0i TS と EP0o TS 割り込みは、コントロールイン、アウト転送の両方で使用します。従って、コントロール転送の方向とステージを管理するために、サンプルプログラムは TRANS_IN、TRANS_OUT、WAIT の 3 つのステートを持っています。詳細は、「5.6 コントロール転送」をご覧ください。

H8S/2218 グループ、H8S/2212 グループハードウェアマニュアルには、割り込み発生時の USB ファンクションモジュールの動作と、アプリケーション側の動作概略が示してあります。次節からは、アプリケーション側ファームウェアの詳細を USB の転送方式ごとに説明します。

5.3 USB 動作クロック安定割り込み

このモジュールは USB モジュールストップ解除後、48MHz の USB 動作クロック安定時間を自動的にカウントした後、発生します。割り込み受信後、USB ケーブル接続待ち状態へ移行します。

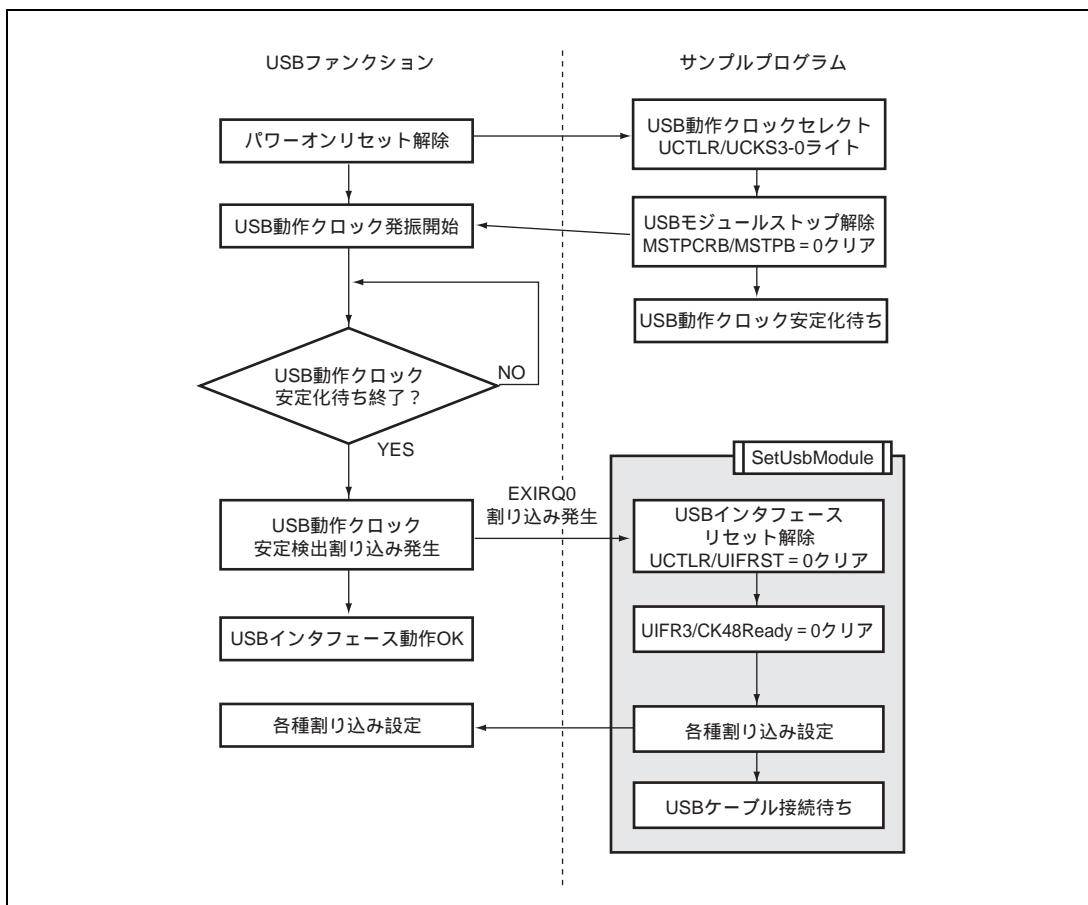


図 5.3 USB 動作クロック安定検出割り込み

5.4 ケーブル接続時 (VBUS) 割り込み

USB ファンクションモジュールのケーブルを、ホストコントローラに接続した際に発生します。アプリケーション側はマイコンの初期設定完了後、汎用出力ポートを使用して USB データバスの D+ をプルアップします。このプルアップによって、ホストコントローラはデバイスが接続されたことを認識します。(図 5.4 参照)

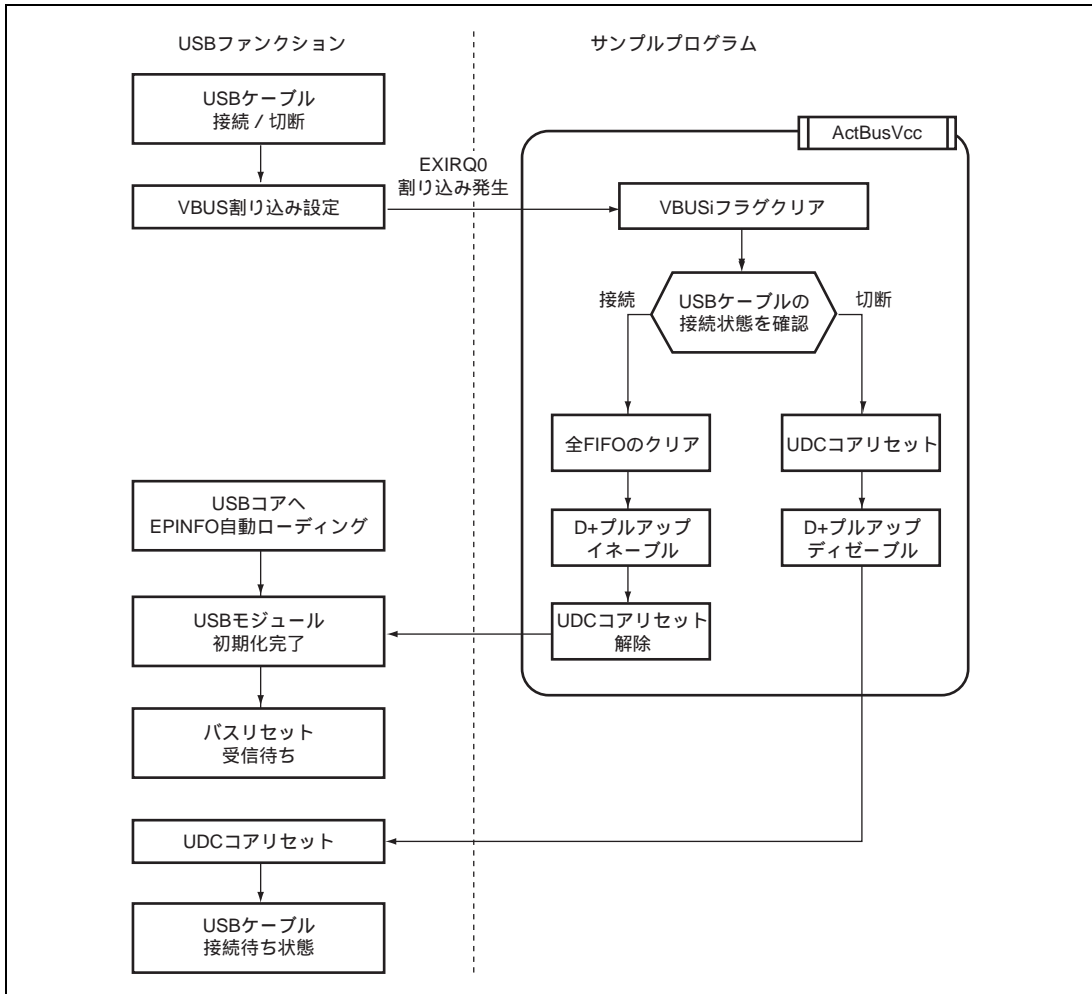


図 5.4 ケーブル接続時割り込み

5.5 バスリセット時 (BRST) 割り込み

ホストコントローラが USB データバスにデバイスが接続されたことを認識すると、バスリセット信号を出力します。ホストからのバスリセット信号を受信するとバスリセット割り込みが発生します。

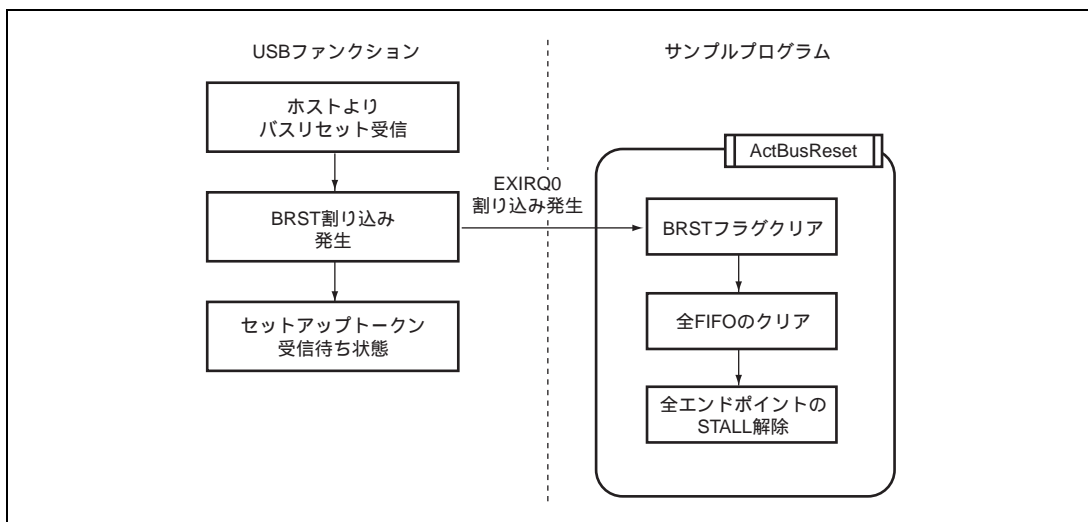


図 5.5 バスリセット割り込み

5.6 コントロール転送

コントロール転送には、割り込みフラグレジスタのビット 0、1、3 を使用します。コントロール転送は、データステージにおけるデータの向きによって、2 つに分けることができます。（図 5.6 参照）

データステージにおいて、ホストコントローラから USB ファンクションへデータ転送する場合はコントロールアウト転送、反対の場合がコントロールイン転送です。

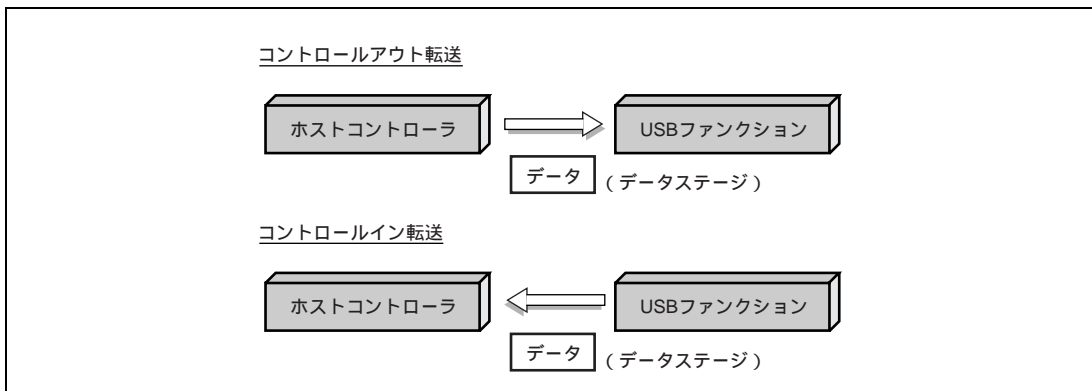


図 5.6 コントロール転送

コントロール転送は、セットアップ、データ(ない場合もある)、ステータスの 3 つのステージで構成されます(図 5.7)。また、データステージは、複数のパストランザクションで構成されます。

コントロール転送では、データの向きが反転することによってステージが切り替わったことを認識します。したがって同じ割り込みフラグを使用して、コントロールイン転送または、コントロールアウト転送を行う関数を呼び出します(表 5.1 参照)。このため、現在イン、アウトどちらのコントロール転送が行われているかをファームウェアがステートによって管理し(図 5.7 参照)、適切な関数を呼び出す必要があります。データステージにおけるステート (TRANS_IN、TRANS_OUT) は、セットアップステージで受信するコマンドによって決定します。

5. サンプルプログラム概要

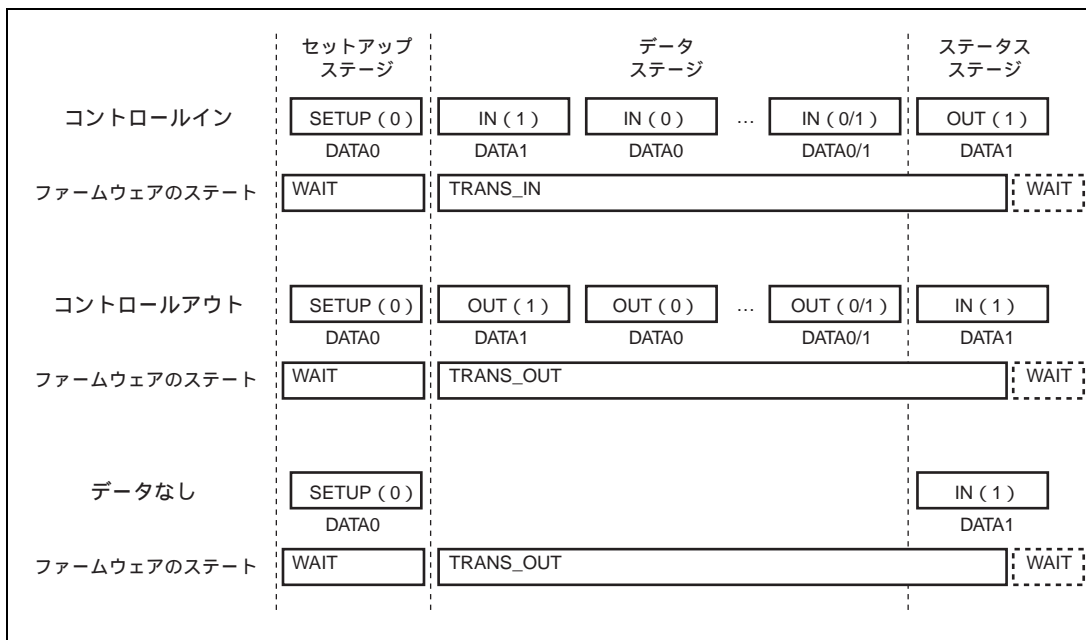


図 5.7 コントロール転送における各ステージ

5.6.1 セットアップステージ

セットアップステージでは、ホストとファンクションがコマンドの送受信を行います。コントロールイン転送、コントロールアウト転送共に、ファームウェアのステートは WAIT になります。また発行されるコマンドの種類によって、コントロールイン転送またはアウト転送の区別を行い、データステージにおけるファームウェアのステート (TRANS_IN、TRANS_OUT) を決定します。

- TRANS_INとなるコマンド・・・GetDescriptor (標準コマンド)

図 5.8 にセットアップステージにおけるサンプルプログラムの動作を示します。図の左側は、USB ファンクションモジュールの動作を示しています。

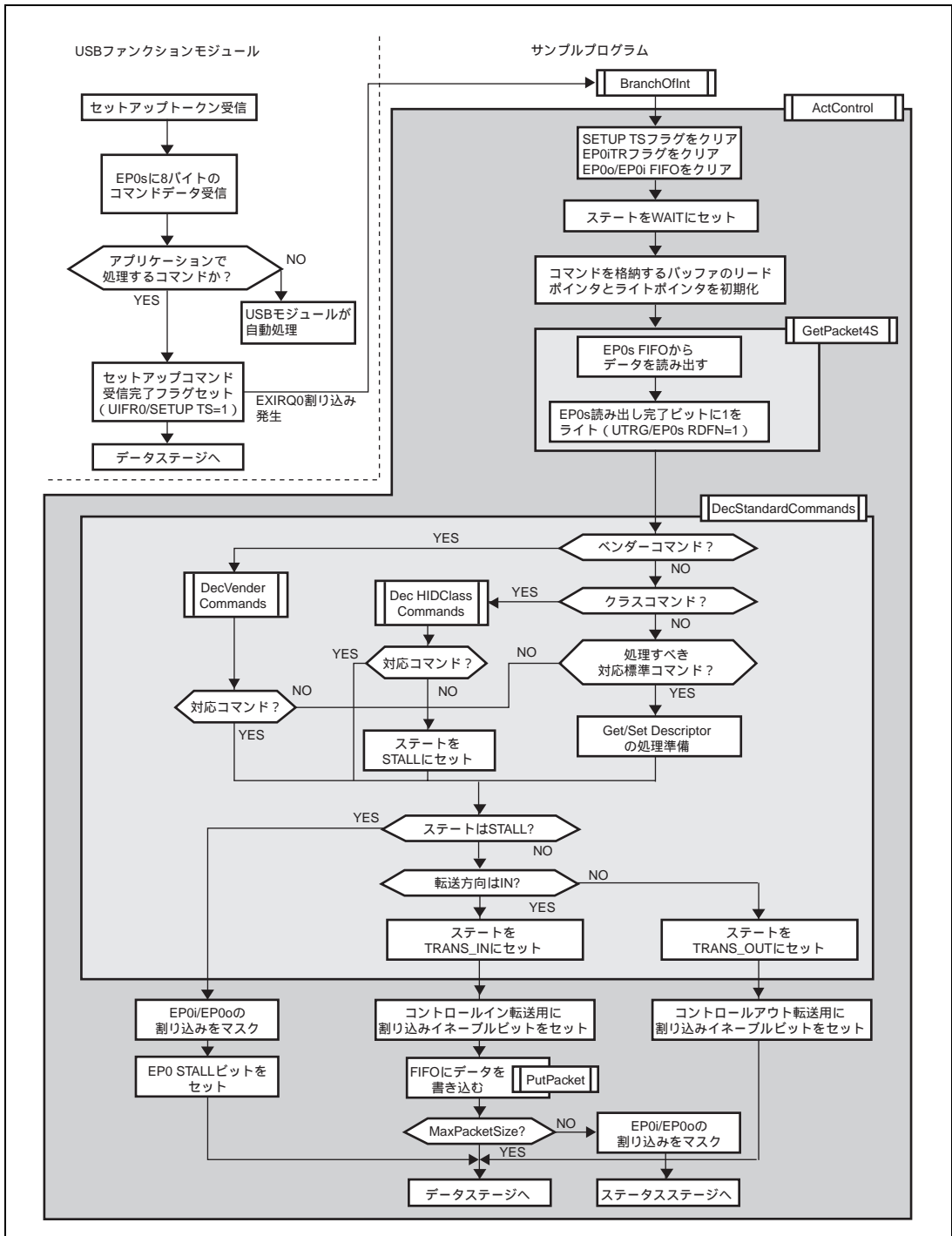


図 5.8 セットアップステージ

5. サンプルプログラム概要

5.6.2 データステージ

データステージでは、ホストとファンクションがデータの送受信を行います。ファームウェアのステートは、セットアップステージで行ったコマンドのデコード結果によって、コントロールイン転送の場合は TRANS_IN に、コントロールアウト転送の場合は TRANS_OUT になります。図 5.9、図 5.10 にコントロール転送のデータステージにおけるサンプルプログラムの動作を示します。

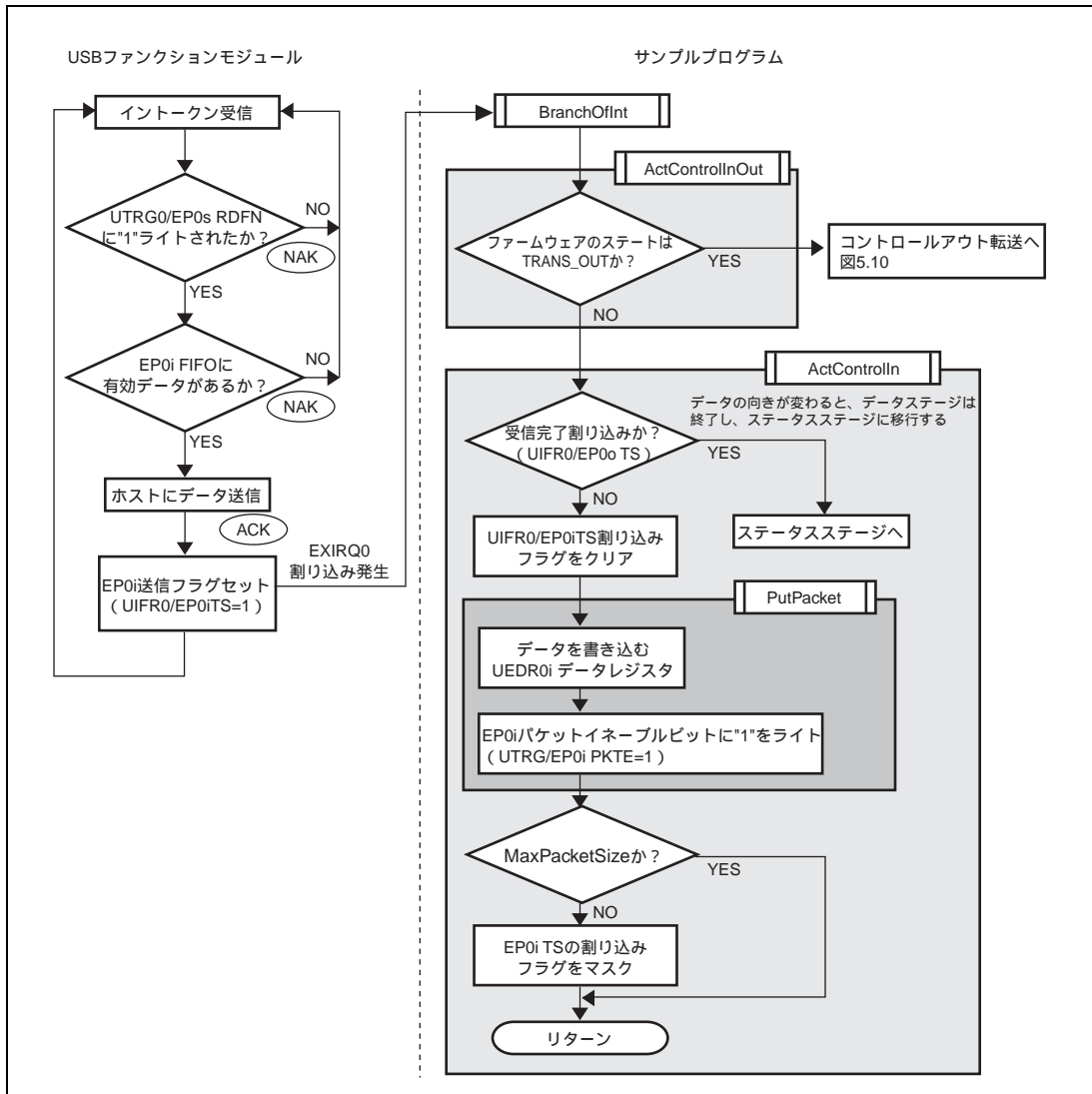


図 5.9 データステージ (コントロールイン転送)

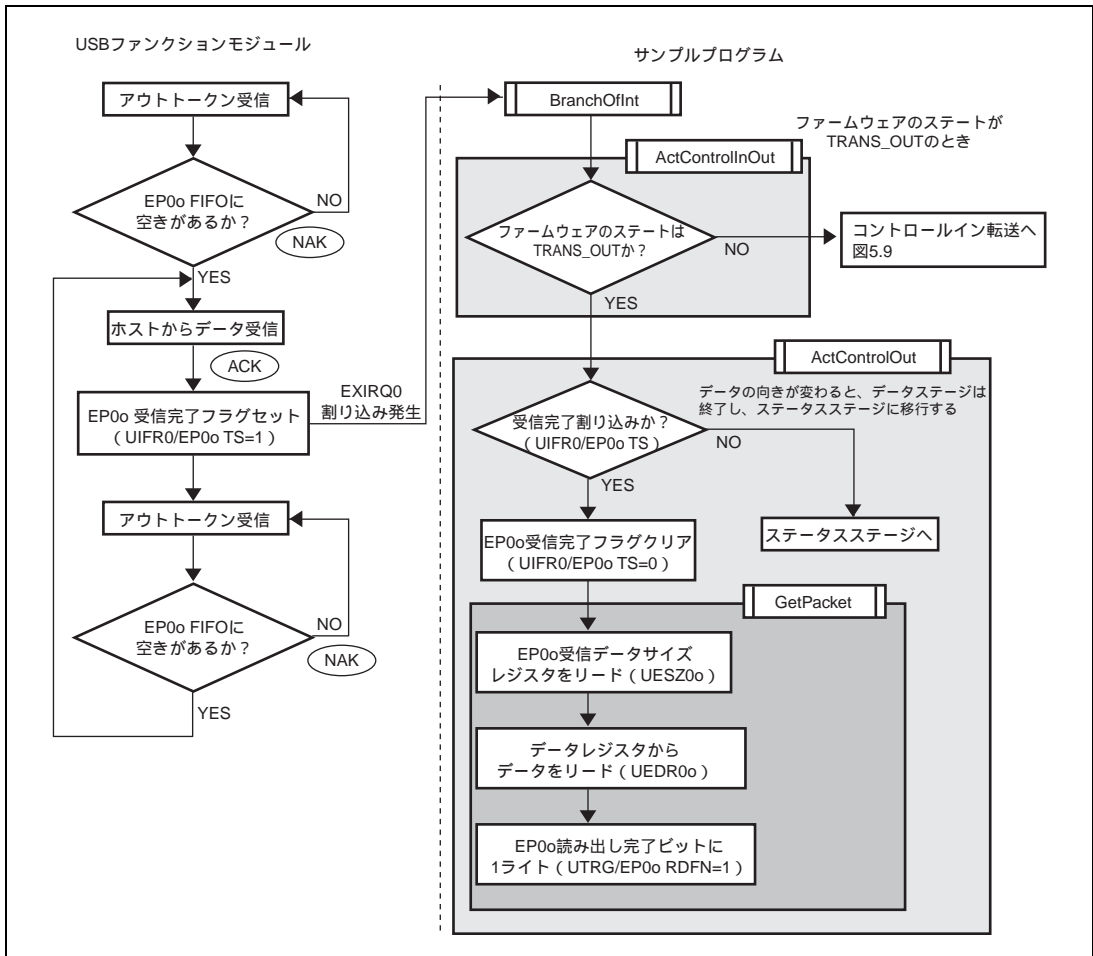


図 5.10 データステージ (コントロールアウト転送)

5. サンプルプログラム概要

5.6.3 ステータスステージ

ステータスステージは、データステージと反対方向のトークンによって開始されます。つまり、コントロールイン転送では、ホストコントローラからのアウトトークンによってステータスステージが開始され、コントロールアウト転送では、ホストコントローラからのイントトークンによってステータスステージが開始されます。図 5.11、図 5.12 にコントロール転送のステータスステージにおけるサンプルプログラムの動作を示します。

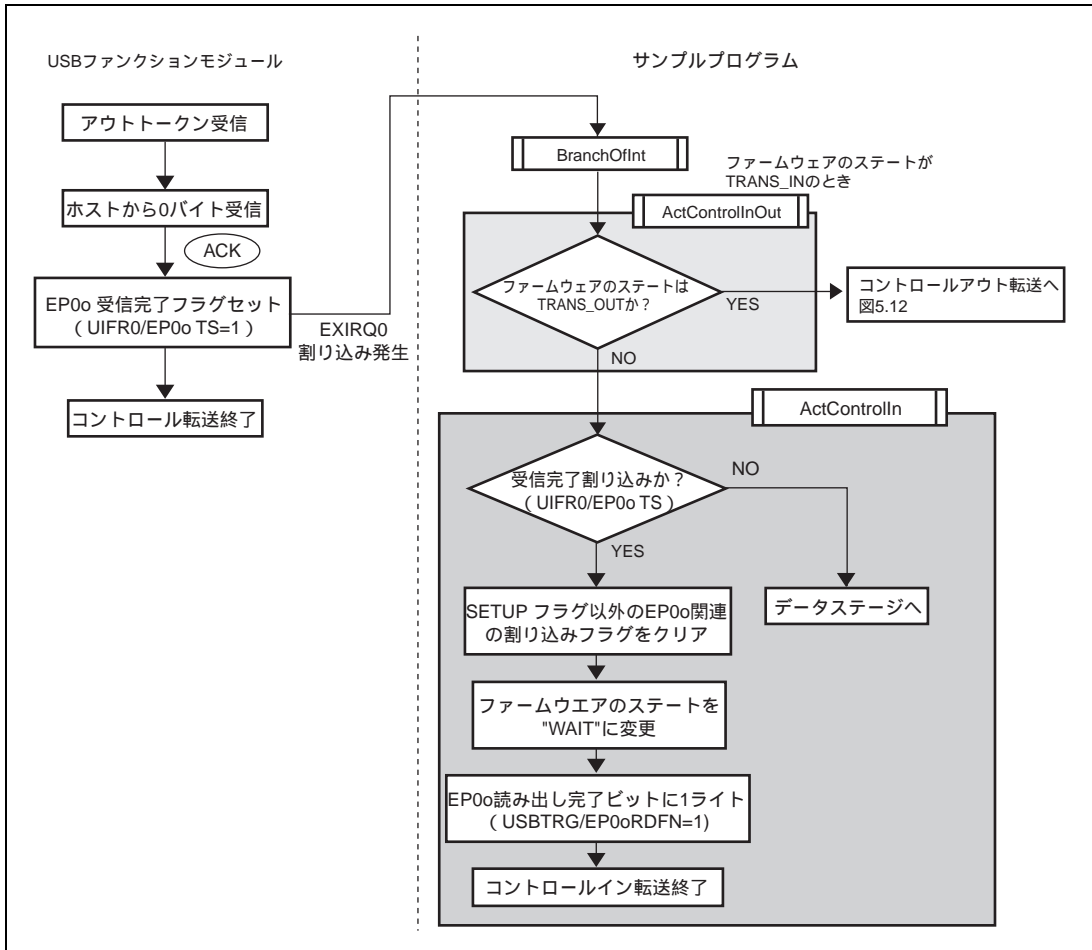


図 5.11 ステータスステージ (コントロールイン転送)

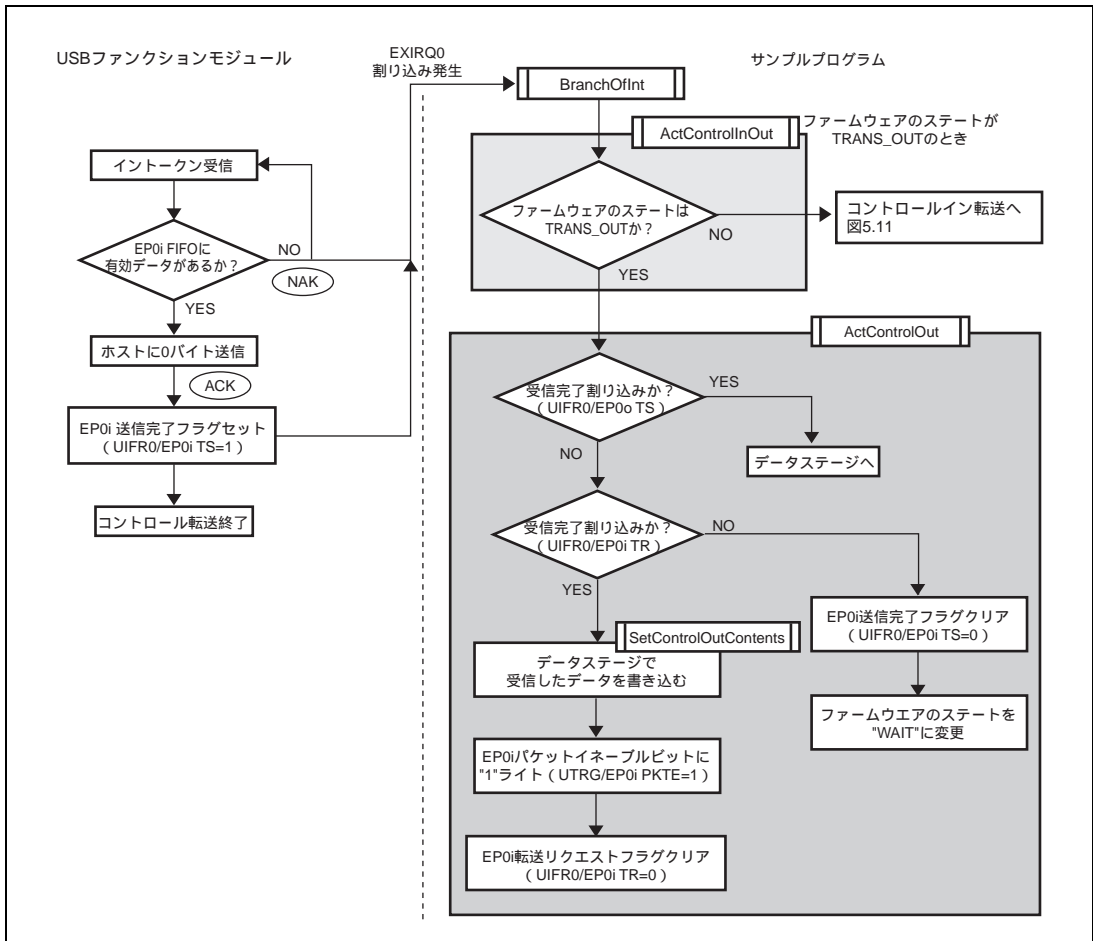


図 5.12 ステータスステージ (コントロールアウト転送)

5.7 インタラプト転送

インタラプト転送はデータを送信する向きによって、2つに分けることができます。USB ファンクションからホストコントローラへデータ転送する場合をインタラプトイン転送、反対の場合をインタラプトアウト転送と呼びます。H8S/2218 はインタラプトイン転送のみサポートしています。（図 5.13 参照）

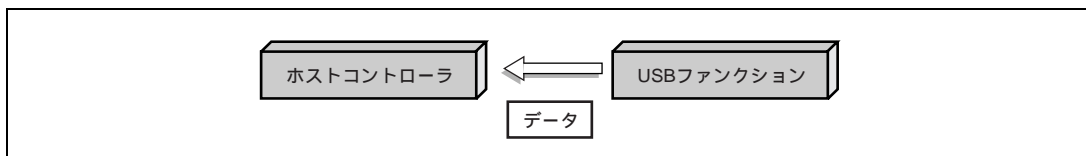


図 5.13 インタラプト転送

5.7.1 インタラプトイン転送

インタラプトイン転送には、割り込みフラグレジスタ 0 のビット 4 (EP3TS) を使用します。USB ホストからのイントークンに対して、USB ファンクションは、EP3 FIFO に有効データがなかった場合、NAK ハンドシェイクを送信し EP3TR フラグがセットされます。また、EP3 FIFO に有効データがあった場合、USB ホストにデータを送信し、USB ホストからの ACK ハンドシェイクを受信すると EP3TS フラグがセットされます。

USB ファンクションは EP3TS のフラグがセットされると ActInterruptIn 関数を実行します。そして、ActInterruptIn 関数では送信する HID データがある場合、USB エンドポイントデータレジスタ 3 (UEDR3) にデータをライトし、USB ホストからのイントークンに備えます。

この時ファームウェアのステートは “WAIT” 又は “TRANS_IN” どちらかの状態です。サンプルプログラムのインタラプトイン転送における動作を図 5.14 に示します。図の左側は、USB ファンクションモジュールの動作を示しています。

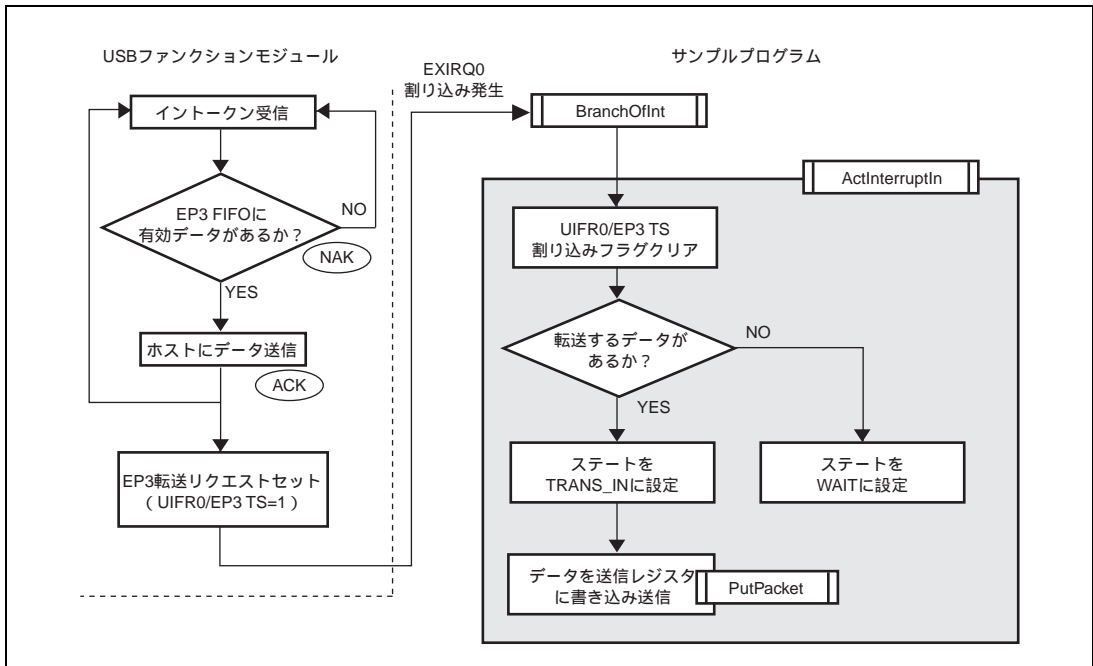


図 5.14 インタラプトイン転送

5.8 マウスデータの生成

MS2218CP ボードにはマウスが接続できないため、本サンプルプログラムでは MS2218CP ボード上の SW5 ~ 12 を使用して USB マウスの疑似データ (HID データ) を生成し USB マウスとして動作させています。

H8S/2218 の 16 ビットタイマ割り込みを使い MS2218CP ボード上 SW5 ~ 12 のキースキャンを行うことにより、HID データを生成します。

- SW5を押すことでマウスの左ボタンが押されたデータを生成。
- SW6を押すことでマウスの右ボタンが押されたデータを生成。
- SW8を押すことでポインタの移動データをデータテーブルから読み出しポインタが移動するデモンストレーションを行います。
- (デモモード)
- SW9を押すことでポインタがY方向下側に移動するデータを生成。
- SW10を押すことでポインタがY方向上側に移動するデータを生成。
- SW11を押すことでポインタがX方向右側に移動するデータを生成。
- SW12を押すことでポインタがX方向左側に移動するデータを生成。

これらの SW を使用し生成したデータを、ActMakeHidData 関数に引き渡すことにより、インタラプト転送を使用してホスト PC に HID データを送信します。サンプルプログラムの HID データ生成動作を図 5.15 に示します。

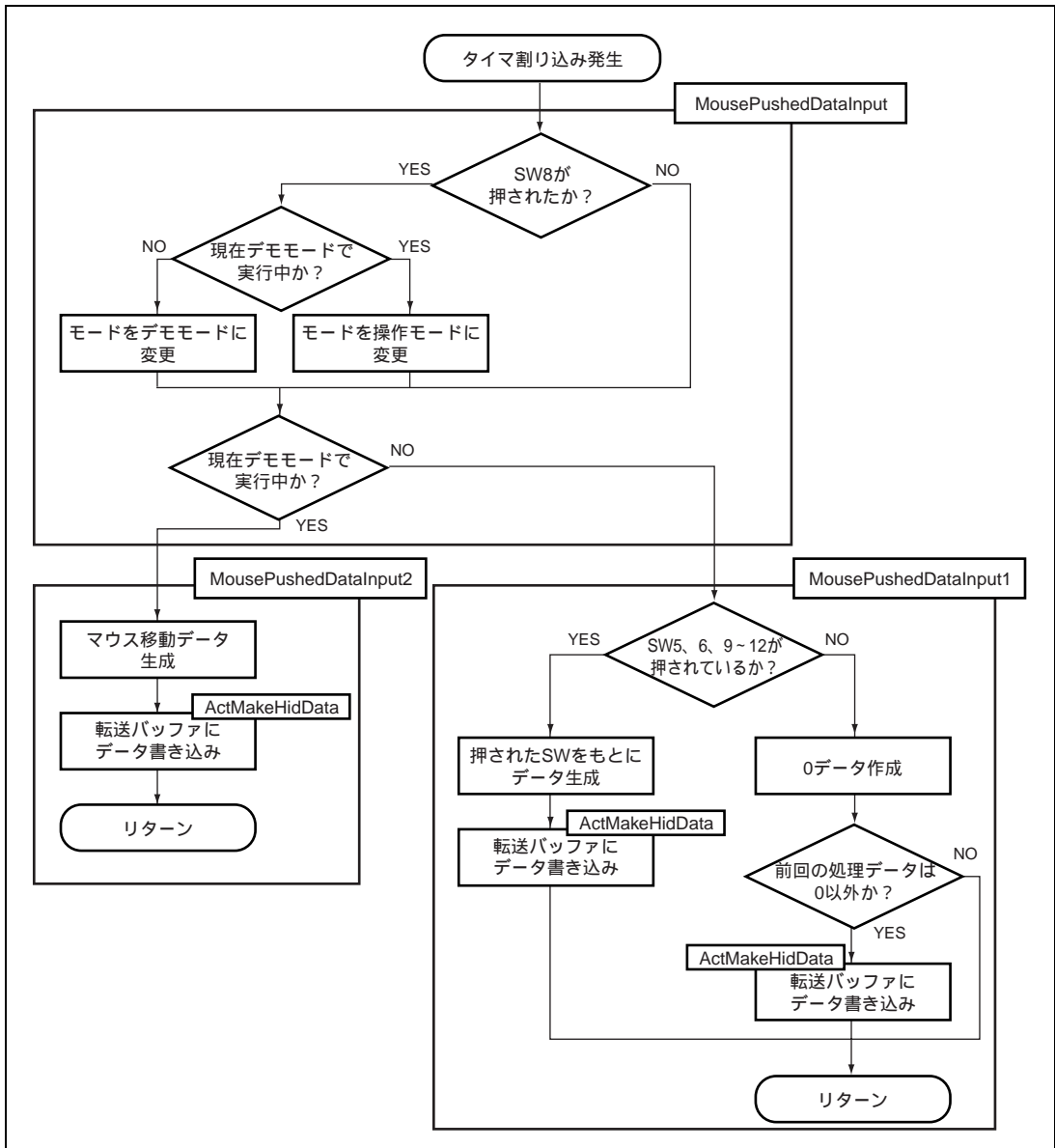


図 5.15 マウスデータ生成

6. アナライザのデータ

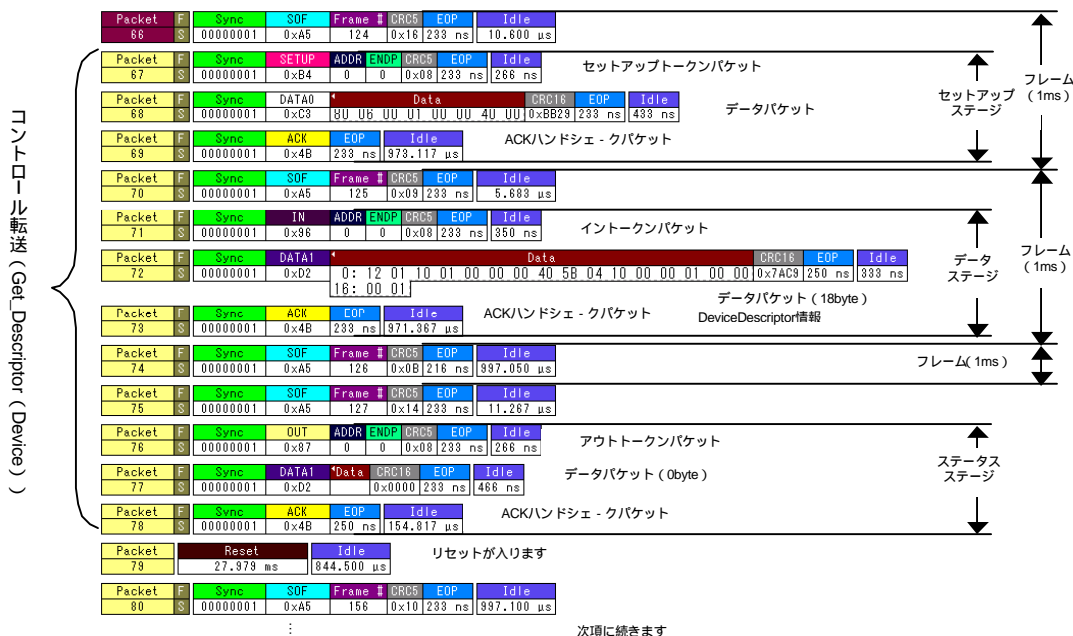
この章では、H8S/2218 内蔵 USB ファンクションモジュールを使用して、CATC 社製 USB プロトコルアナライザ「USB Advisor」（国内：（株）東陽テクニカ（<http://www.toyo.co.jp/>））を用いた測定を行い、実際にバスを流れているデータについて「デバイス接続時のコントロール転送」および「HID データのインタラプトイン転送」を例に説明します。

【注】 各パケットの前面にある「Packet」は測定時のパケット通し番号です。
後部にある「Idle」はパケット間のアイドルとなります。

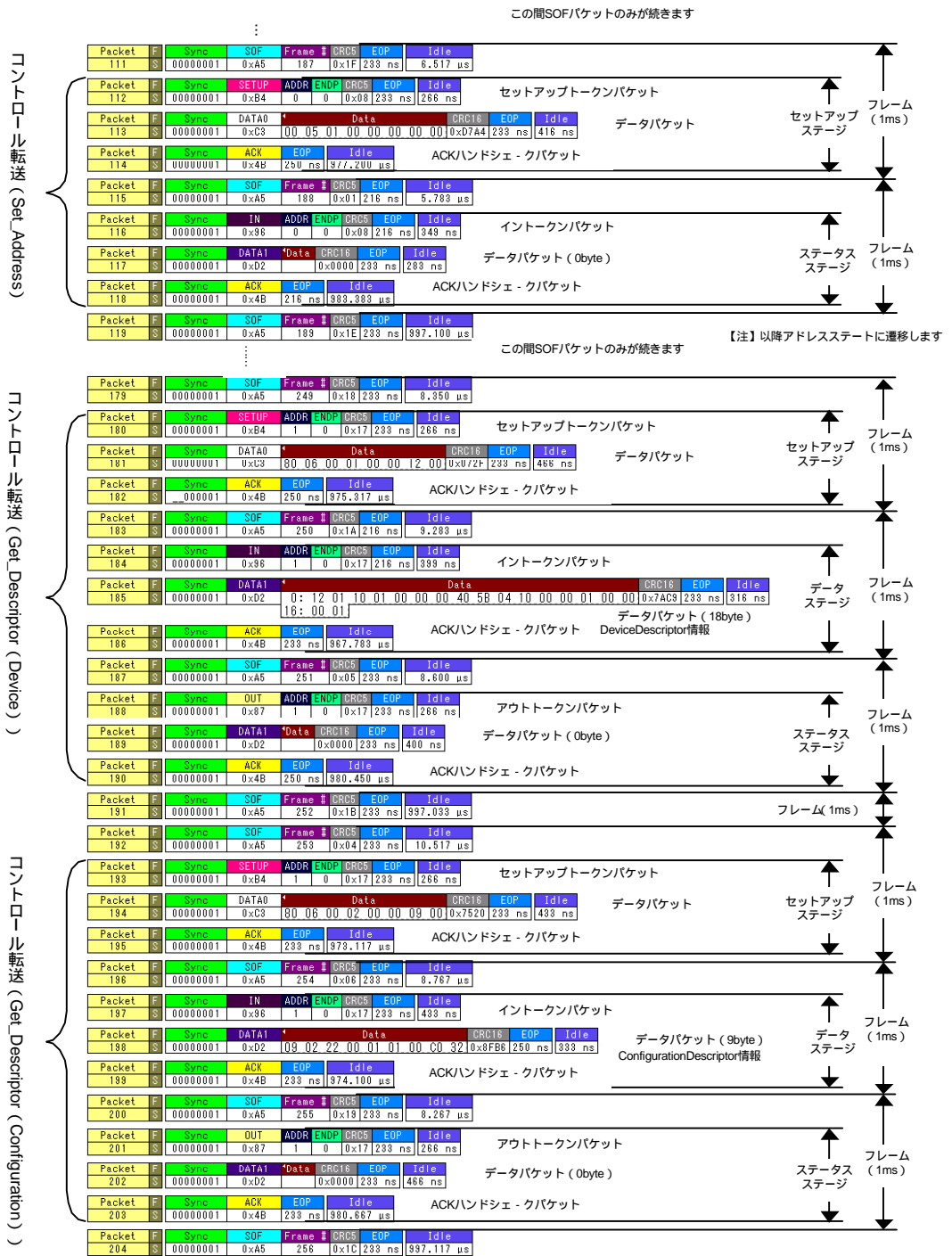
6.1 デバイス接続時のコントロール転送

本デバイスをホストコントローラに接続し、Vbus に電源が供給されている（電源投入ステート）状態から、デバイスが使用可能になる状態（構成ステート）に至るまでを測定したものです。

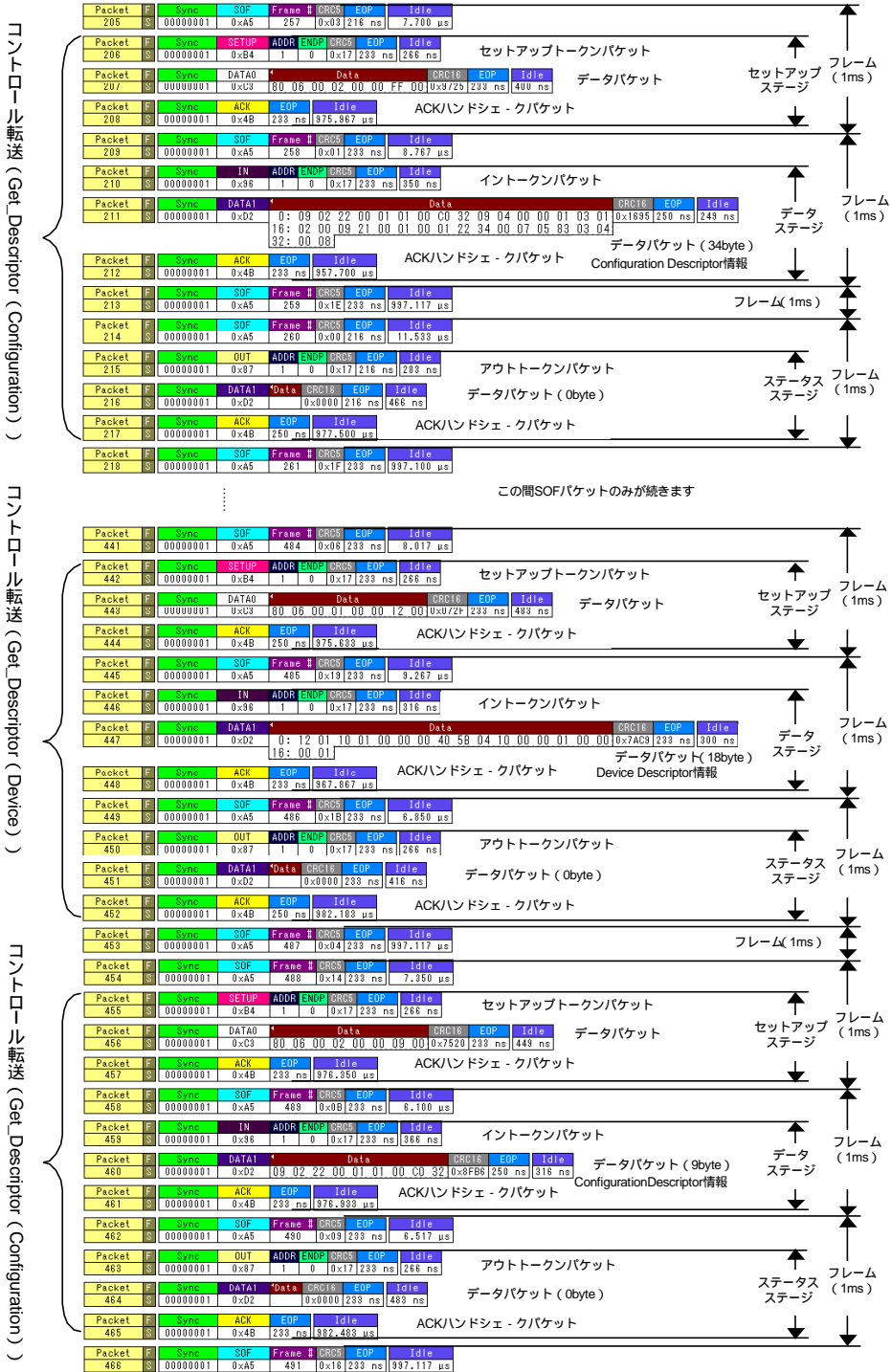
なお、ホストコントローラによりパケットのスケジューリングが本図と異なる場合がありますが、構成ステートに至るまでのコマンドの流れは同一です。



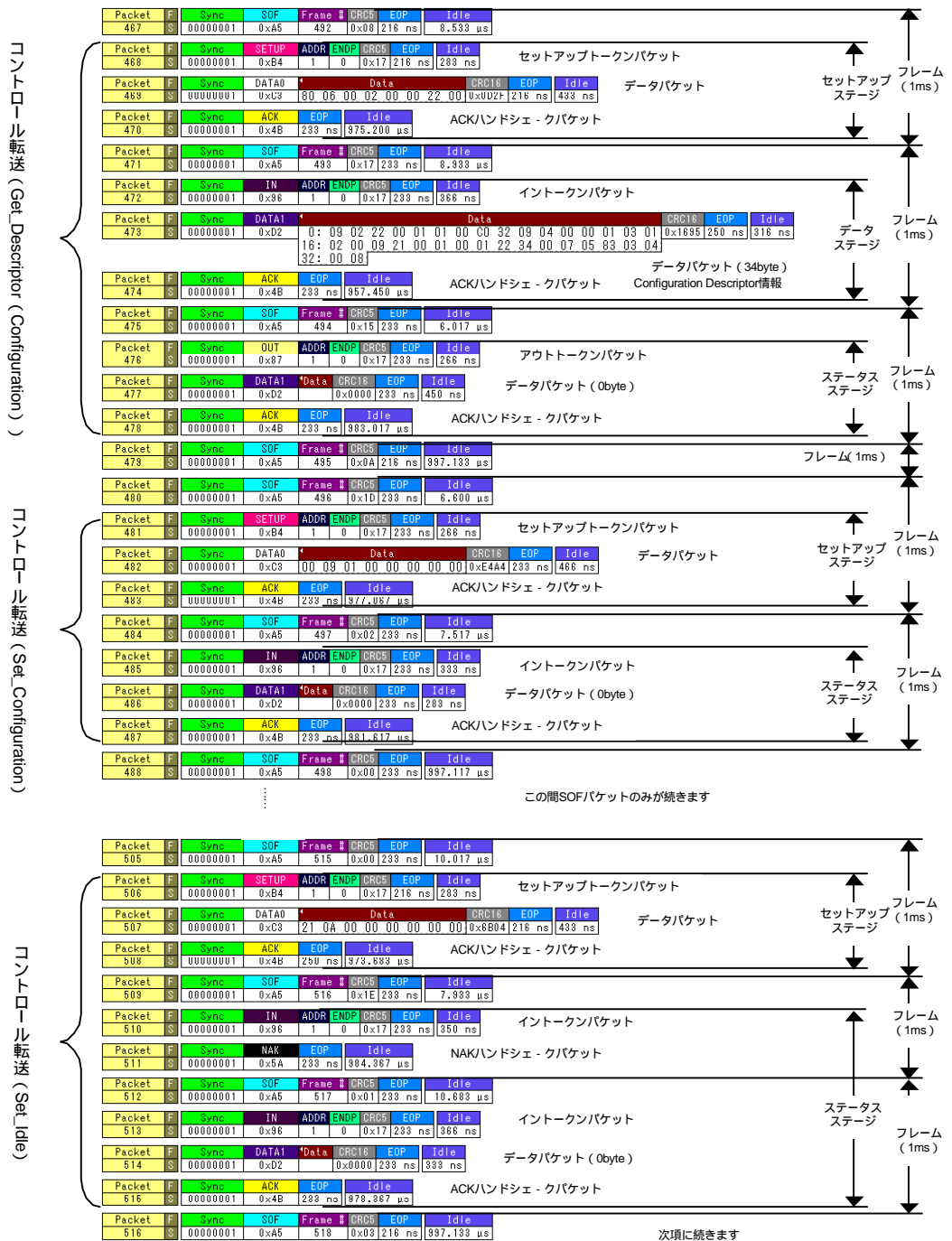
6. アナライザのデータ



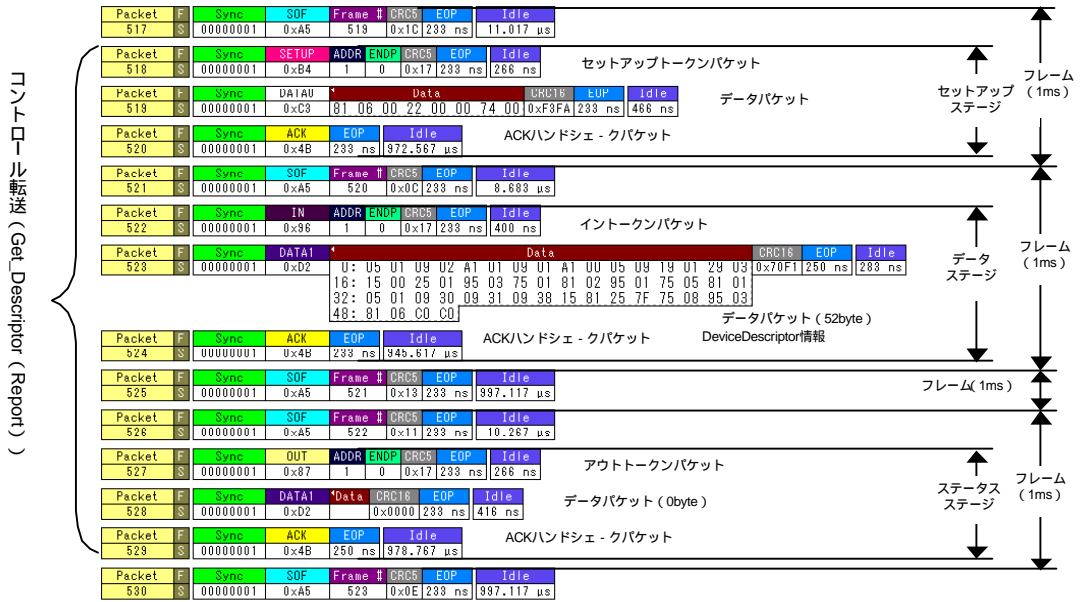
6. アナライザのデータ



6. アナライザのデータ

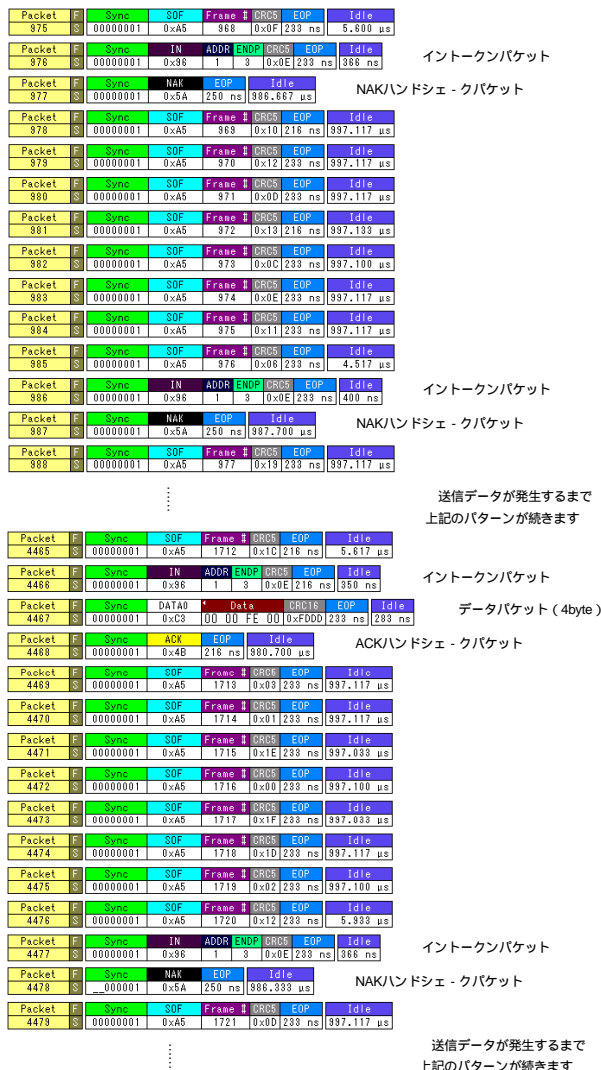


6. アナライザのデータ



6.2 HID データのインタラプトイン転送

は、本デバイスから USB ホストに向けてインタラプトイン転送で HID データの送信を行っている際の測定結果です。USB ホストが発行するインタラプトイン転送に対し、送信可能なデータがない場合は NAK を送信します。送信可能なデータがある場合は 4 バイトの HID データを送信します。USB ホストは HID データを受信すると ACK を発行します。



H8S/2218 USBファンクションモジュール
Human Interface Devices (HID) Class アプリケーションノート

発行年月 2003年10月20日 Rev.1.00
発行 株式会社ルネサス テクノロジ 営業企画統括部
〒100-0004 東京都千代田区大手町 2-6-2
編集 株式会社ルネサス小平セミコン 技術ドキュメント部



営業お問合せ窓口
株式会社ルネサス販売

<http://www.renesas.com>

本			社	〒100-0004	千代田区大手町2-6-2 (日本ビル)	(03) 5201-5350
京	浜	支	社	〒212-0058	川崎市幸区鹿島田890-12 (新川崎三井ビル)	(044) 549-1662
西	東	支	社	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル2F)	(042) 524-8701
札	幌	支	店	〒060-0002	札幌市中央区北二条西4-1 (札幌三井ビル5F)	(011) 210-8717
東	北	支	社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア13F)	(022) 221-1351
い	わ	支	店	〒970-8026	いわき市平小太郎町4-9 (損保ジャパンいわき第二ビル3F)	(0246) 22-3222
茨	城	支	社	〒312-0034	ひたちなか市堀口832-2 (日立システムプラザ勝田1F)	(029) 271-9411
新	潟	支	店	〒950-0087	新潟市東大通1-4-2 (新潟三井物産ビル3F)	(025) 241-4361
松	本	支	社	〒390-0815	松本市深志1-2-11 (昭和ビル7F)	(0263) 33-6622
中	部	営	本	〒460-0008	名古屋市中区栄3-13-20 (栄センタービル4F)	(052) 261-3000
浜	松	支	店	〒430-7710	浜松市板屋町111-2 (浜松アクタワー10F)	(053) 451-2131
西	部	営	本	〒541-0044	大阪市中央区伏見町4-1-1 (大阪明治生命館ランドアクスタワー10F)	(06) 6233-9500
北	陸	支	社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル8F)	(076) 233-5980
中	国	支	社	〒730-0036	広島市中区袋町5-25 (広島袋町ビルディング8F)	(082) 244-2570
松	山	支	店	〒790-0003	松山市三番町4-4-6 (GEエジソンビル松山2号館3F)	(089) 933-9595
鳥	取	支	店	〒680-0822	鳥取市今町2-251 (日本生命鳥取駅前ビル)	(0857) 21-1915
九	州	支	社	〒812-0011	福岡市博多区博多駅前2-17-1 (ヒロカネビル本館5F)	(092) 481-7695
鹿	児	支	店	〒890-0053	鹿児島市中央町12-2 (明治生命西鹿児島ビル2F)	(099) 284-1748

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：カスタマサポートセンタ E-Mail: csc@renesas.com

H8S/2218 USB ファンクションモジュール Human Interface Devices (HID) Class
アプリケーションノート



ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668

RJJ06B0214-0100Z