

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ものですが、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、パソコン機器、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が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。

お客様各位

---

## 資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

---

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

ルネサステクノロジ ホームページ (<http://www.renesas.com>)

2003年4月1日

株式会社ルネサス テクノロジ  
カスタマサポート部

## ご注意

### 安全設計に関するお願い

- 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留意ください。

### 本資料ご利用に際しての留意事項

- 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。
- 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。
- 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しては、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。
- 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が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。
- 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は負いません。
- 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。
- 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。
- 本資料に関し詳細についてのお問い合わせ、その他お気付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。

# SH7727 USB ファンクションモジュール Mass Storage Class (Bulk-Only Transport)

アプリケーションノート

ルネサスSuperH™ RISC engine

SH7727

HD6417727

## ご注意

- 1 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合、または国外に持ち出す場合は日本国政府の許可が必要です。
- 2 本書に記載された情報の使用に際して、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に対する保証または実施権の許諾を行うものではありません。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
- 3 製品及び製品仕様は予告無く変更する場合がありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書をお求めになりご確認ください。
- 4 弊社は品質・信頼性の向上に努めておりますが、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は、事前に弊社営業担当迄ご相談をお願い致します。
- 5 設計に際しては、特に最大定格、動作電源電圧範囲、放熱特性、実装条件及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いただきますようお願い致します。  
保証値を越えてご使用された場合の故障及び事故につきましては、弊社はその責を負いません。また保証値内のご使用であっても半導体製品について通常予測される故障発生率、故障モードをご考慮の上、弊社製品の動作が原因でご使用機器が人身事故、火災事故、その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
- 6 本製品は耐放射線設計をしておりません。
- 7 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
- 8 本書をはじめ弊社半導体についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

---

# はじめに

---

本アプリケーションノートは、SH7727 内蔵の USB ファンクションモジュールを用いた Mass Storage Class (Bulk-Only Transport) ファームウエア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 ファームウエア作成の際に、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、本書のほかに以下の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 Mass Storage Class Specification Overview Revision 1.1
- Universal Serial Bus Mass Storage Class(Bulk-Only Transport) Revision 1.0
- SH7727 ハードウエアマニュアル
- SH7727 Solution Engine ( MS7727SE01 ) 取扱説明書
- SH7727 E10A エミュレータユーザーズマニュアル

**【ご注意】** 本アプリケーションノート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「インタラプト」に関するファームウエアは準備しておりません。「インタラプト」( SH7727 ハードウエアマニュアル 23-1 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 SH7727、SH7727 Solution Engine のハードウェア仕様を記載しておりますが、詳細は SH7727 のハードウエアマニュアル、ならびに SH7727 Solution Engine の取扱説明書を御覧ください。



---

# 目次

---

1.	概要 .....	1-1
2.	USB Mass Storage Class ( Bulk-Only Transport ) の概要 .....	2-1
2.1	USB Mass Storage Classについて .....	2-1
2.2	サブクラスコードについて .....	2-2
2.3	Bulk-Only Transportについて .....	2-2
2.3.1	コマンドトランSPORTについて .....	2-3
2.3.2	ステータストラNSPORTについて .....	2-4
2.3.3	データトランSPORTについて .....	2-5
2.3.4	クラスコマンドについて .....	2-6
2.4	サブクラスコードSCSI transparent command setについて .....	2-6
3.	開発環境 .....	3-1
3.1	ハードウエア環境 .....	3-1
3.2	ソフトウエア環境 .....	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 .....	3-3
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 .....	3-7
3.4	RAM Diskの使用方法 .....	3-7
4.	サンプルプログラム概要 .....	4-1
4.1	状態遷移図 .....	4-1
4.2	USB通信状態 .....	4-2
4.2.1	コントロール転送について .....	4-3
4.2.2	バルク転送について .....	4-3
4.3	ファイル構成 .....	4-4
4.4	関数の機能 .....	4-6
4.5	RAM-Diskについて .....	4-10
4.6	サポートするSCSIコマンドの動作について .....	4-11
4.7	エラー時の処理について .....	4-12

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 .....	5-1
5.2.1	各転送への分岐方法.....	5-3
5.3	ケーブル接続時 ( BRST ) 割り込み.....	5-4
5.4	コントロール転送 .....	5-5
5.4.1	セットアップステージ .....	5-6
5.4.2	データステージ .....	5-8
5.4.3	ステータスステージ .....	5-10
5.5	パルク転送 .....	5-12
5.5.1	コマンドトランスポート .....	5-13
5.5.2	データトランスポート .....	5-15
5.5.3	ステータストラنسポート .....	5-19
6.	アナライザのデータ .....	6-1

---

# 1. 概要

---

本アプリケーションノートは、SH7727 の USB ファンクションモジュールの使用方法、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。

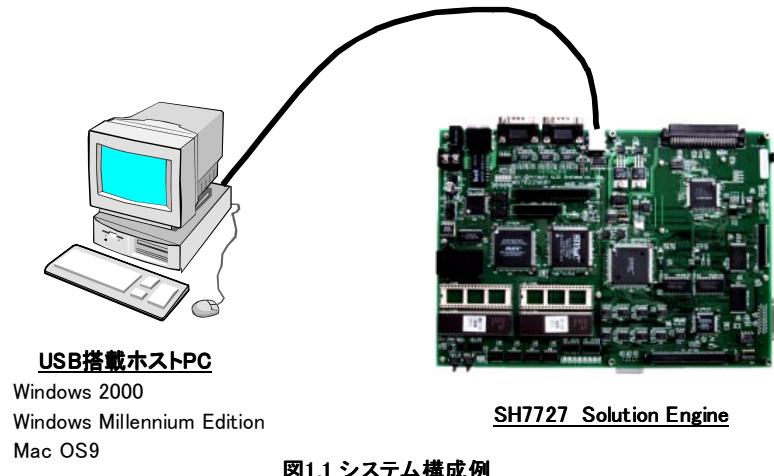
SH7727 内蔵 USB ファンクションモジュールの特長を以下に示します。

- 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
- EXCPGによる内部システムクロック / 外部入力 ( 48MHz ) を選択可能
- 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	8 Byte	8 Byte	
	EP0o	コントロールアウト	8 Byte	8 Byte	
エンドポイント 1	EP1	バルクアウト	64 Byte	64 × 2 ( 128 Byte )	可能
エンドポイント 2	EP2	バルクイン	64 Byte	64 × 2 ( 128 Byte )	可能
エンドポイント 3	EP3	インターラプト	8 Byte	8 Byte	

システム構成例を図 1.1 に示します。

**図1.1 システム構成例**

本システムは、SH7727を搭載した日立超LSIシステムズ社製のSH7727 Solution Engine( 以下 SH7727SE )「Windows 2000/Windows Millennium Edition」または「Mac OS9」OSを搭載したPCによって構成されています。

本システムは、ホストPCとSH7727SEをUSBで接続し、SH7727SE上のSD-RAMをRAM Diskとして動作させることにより、ホストPCからSH7727SEのSD-RAM上へデータの保存、読み出しができます。

また、上記OSに標準で付属しているUSB Mass Storage Class ( Bulk-Only Transport ) のデバイスドライバを使用することができます。

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、SH7727のUSBモジュールを短期間で評価可能
2. サンプルプログラムは、USBのコントロール転送、バルク転送をサポート
3. E10A ( PCカード型エミュレータ ) を使用することができ、効率的なデバッグが可能
4. プログラムを追加作成することにより、インタラプト転送についても対応可能\*

【注】\* インタラプト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。

---

## 2. USB Mass Storage Class ( Bulk-Only Transport ) の概要

---

この章では、USB Mass Storage Class ( Bulk-Only Transport ) 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際に、ご参考として御使用ください。なお、規格の詳細については、

「Universal Serial Bus Mass Storage Class Specification Overview Revision 1.1」

「Universal Serial Bus Mass Storage Class ( Bulk-Only Transport ) Revision 1.0」

をご覧ください。

### 2.1 USB Mass Storage Class について

USB Mass Storage Class とは、大規模記憶装置（ストレージ）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 0x08 を記述することが必要です。

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、インターラプト転送、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。

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、インター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、更にインター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の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。

## 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SPORT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 2.1 に示すサブクラスコードが定められています。

表 2.1

サブクラスコード	コマンドの規格
0x01	Reduced Block Commands ( RBC ) , T10/1240-D
0x02	Attachment Packet Interface ( ATAPI ) for CD-ROMs. SFF-8020i, Multi-Media Command Set 2 ( MMC-2 )
0x03	Attachment Packet Interface ( ATAPI ) for Tape. QIC-157
0x04	USB Mass Storage Class UFI Command Specification
0x05	Attachment Packet Interface ( ATAPI ) for Floppies. SFF-8070i
0x06	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ことが必要です。

本サンプルプログラムでは、サブクラスコード値 0x06 の SCSI Primary Commands を使用します。

## 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 2.1 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SPORT ) として管理します。

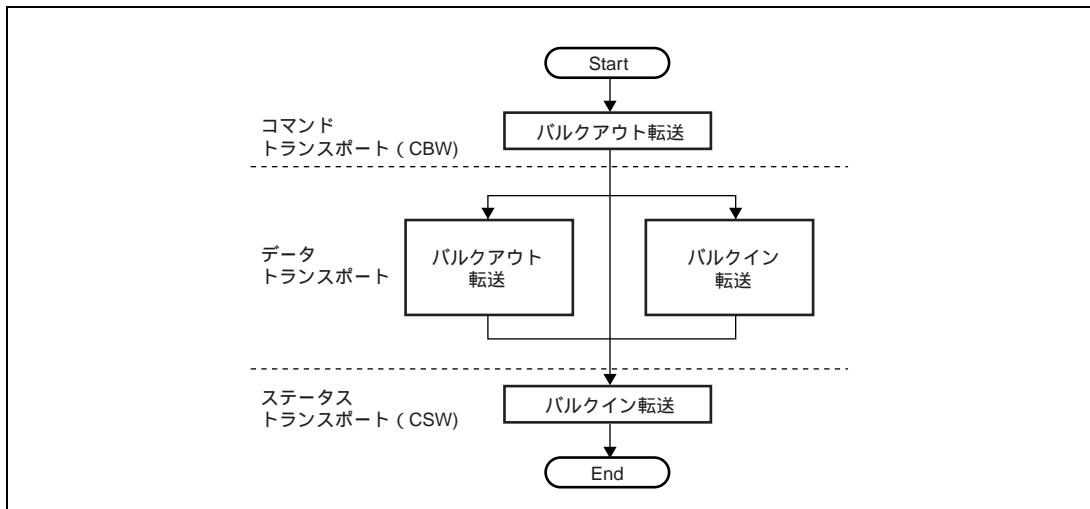


図 2.1 転送方法とトランSPORTの関係

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 0x50 を記述することが必要です。

### 2.3.1 コマンドトランSPORTについて

コマンドトランSPORTはホスト PC がファンクションにパルクアウト転送を用いてコマンドを送ります。このコマンドパケットが 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 からパルクアウト転送を使用して 31 バイト長のパケットで送られて来ます。

内容は表 2.2 に示すフォーマットで送られます。

表 2.2

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

各フィールドを下記に説明いたします。

- dCBWSignature : データパケットが CBW であると認知するためのフィールド。  
値は 43425355h ( 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  
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。ここにホ  
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。

### 2.3.2 ステータストラנסポートについて

ステータストラヌー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 2.3 に示すフォーマットで送られます。

表 2.3

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

各フィールドを下記に説明いたします。

- 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 53425355h ( 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 0x00 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 0x01、フェーズエラーは 0x02。

### 2.3.3 データトランSPORTについて

データトランSPORT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SPORTです。例えば、

Read/Write コマンド ( 4.6 章参照 ) では、データトランSPORT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SPORTは複数のバストランザクションで構成されます。

データトランSPORT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。

#### ( 1 ) データトランSPORT ( バルクアウト転送 ) について

データトランSPORT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。ここで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。

#### ( 2 ) データトランSPORT ( バルクイン転送 ) について

データトランSPORT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。ここで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。

### 2.3.4 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 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 2.4 にクラスコマンドを示します。

表 2.4 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 ( 0xFF )	Bulk-Only Mass Storage Reset	インターフェースをリセットする
254 ( 0xFE )	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ー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。

## 2.4 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 2.6 に示す 9 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 2.6 サポートコマンド一覧

Operation Code	コマンド名	コマンドの動作
12	INQUIRY	ドライブに関する情報をホストに伝える
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
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
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
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
1A	MODE SENSE ( 10 )	ドライブの状態をホストに伝える
1E	PREVENT ALLOW MEDIUM REMOVAL	メディアの着脱を禁止 / 許可します
00	TEST UNIT READY	メディアが使用可能か否かを調べる
2F	VERIFY ( 10 )	メディア上のデータにアクセス可能を確かめる

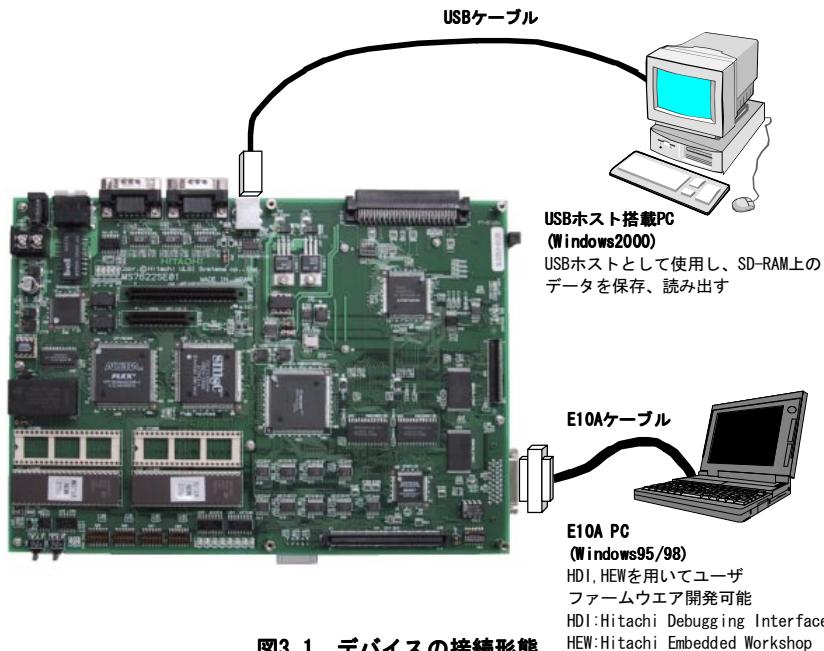
## 3. 開発環境

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。本システムの開発は、以下のデバイス(ツール)を使用しました。

- SH7727 Solution Engine ( 以下SH7727SE 型名MS7727SE01 ) 日立超LSIシステムズ社製
- SH7727 E10A Emulator 日立製作所製
- PCMCIAスロット搭載のPC ( Windows95/98 )
- USBホスト用PC ( Windows 2000/Windows Millennium EditionまたはMac OS9 )
- USBケーブル
- Hitachi Debugging Interface ( 以下HDI ) 日立製作所製
- Hitachi Embedded Workshop ( 以下HEW ) 日立製作所製

### 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。



## (1) SH7727SE

SH7727SE ボードのディップスイッチ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を変更する必要はありません。

表 3.1 ディップスイッチの設定

出荷時	変更後	ディップスイッチの機能
SW1-6 OFF	SW1-6 ON	エンディアンの選択
SW1-8 OFF	SW1-8 ON	E10A エミュレータの選択
SW4-1 OFF	SW4-1 ON	SCIF2 ポーレートの設定
SW4-2 OFF	SW4-2 ON	SCIF2 ポーレートの設定

## (2) USB ホスト PC

USB ポート搭載の、Windows 2000/Windows Millennium Edition または Mac OS9 をインストールしたパソコンを USB ホストとして使用します。本システムでは、上記 OS に標準で搭載されている USB Mass Storage Class ( 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

PC カードスロットに E10A を挿入し、接続用のケーブルを介して SH7727SE と接続してください。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ク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 SH7727 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以下図 3.2 に示します。

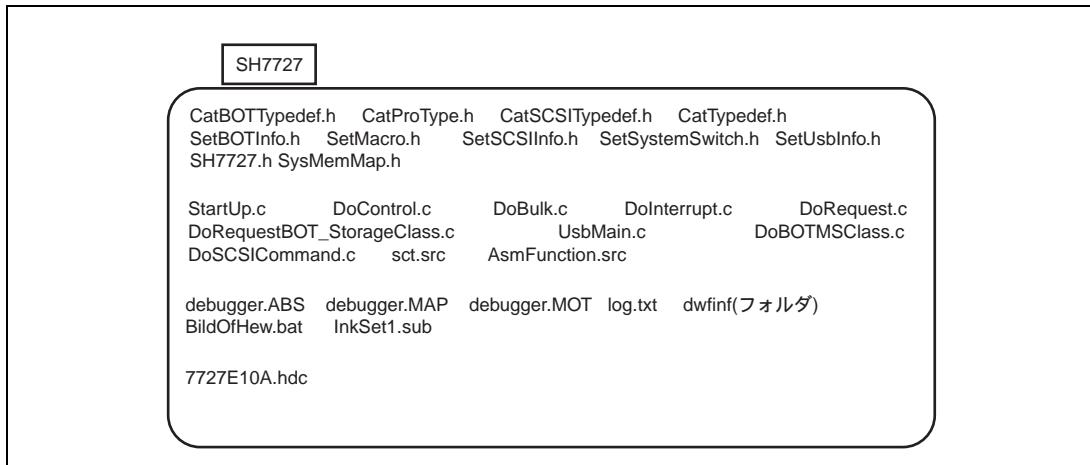


図 3.2 フォルダに含まれるファイル

### 3.2.2 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<sup>\*</sup>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)。

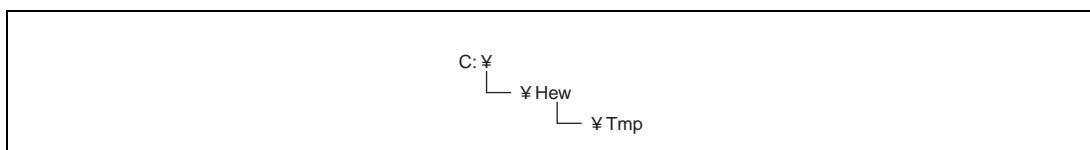


図 3.3 作業フォルダの作成

次に、サンプルプログラムが格納されているフォルダ（SH7727）を、任意のドライブにコピーしてください。この中には、サンプルプログラムと共に BildOfHew.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。BildOfHew.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.ABS の実行ファイルが作成されます。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（図 3.4）。

【注】\* HEW を C:\HEW 以外にインストールした場合、BildOfHew.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 shc.exe のパス、コンパイラが使用する環境変数 shc\_lib の設定は shc.exe のフォルダ、shc\_inc の設定は machine.h のフォルダ、shc\_tmp 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 shcpic.lib のパスを指定してください。

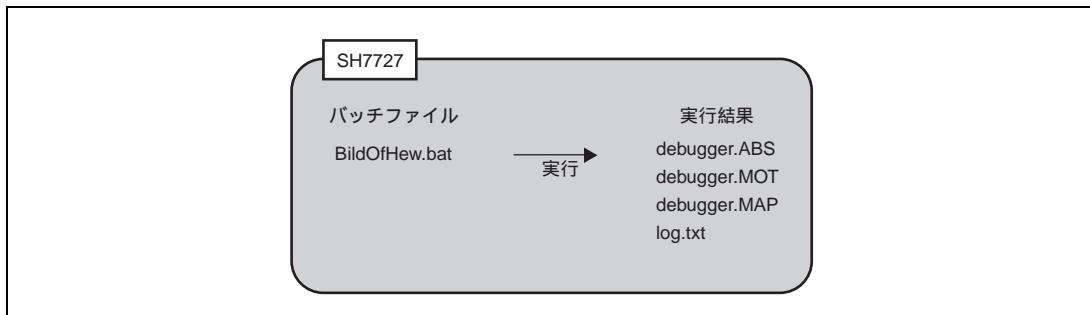


図 3.4 コンパイル結果

### 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。

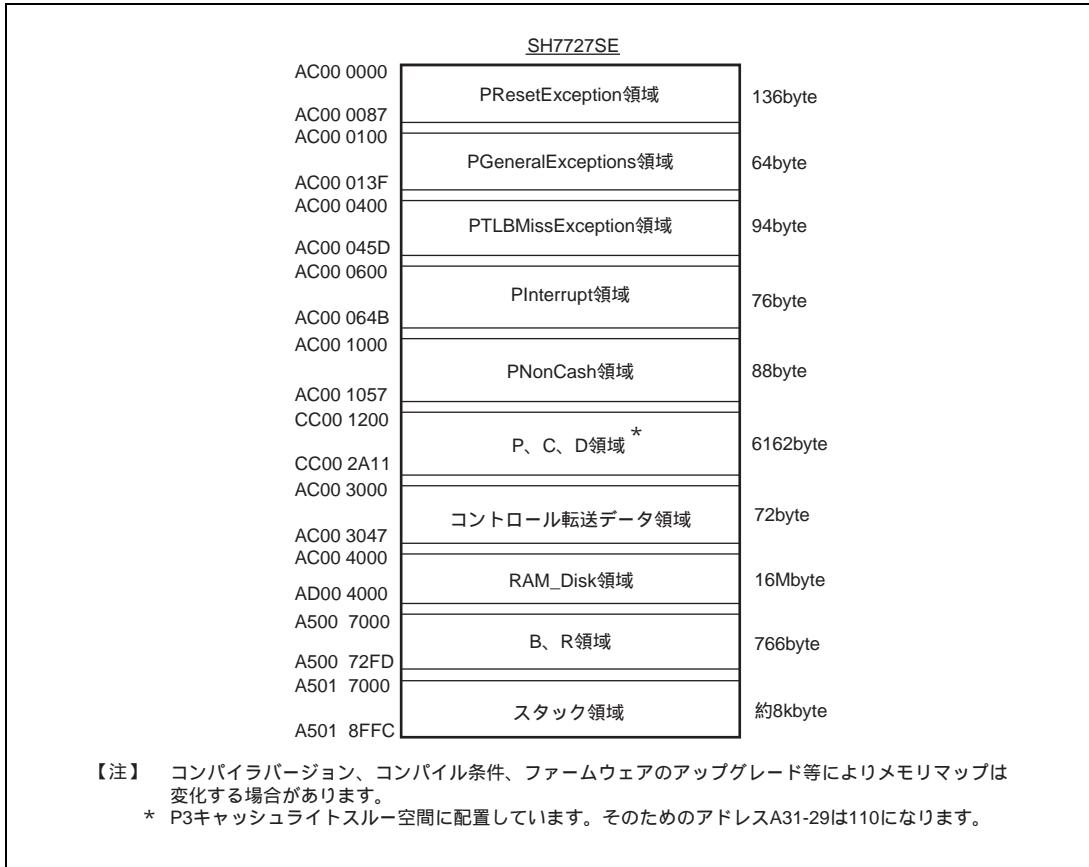


図 3.5 メモリマップ

図 3.5 のように、本サンプルプログラムは PResetException、 PGeneralExceptions、 PTLBMissException、 PInterrupt、 PNonCash、 P、 C、 D の領域を SDRAM 上に、 R、 B 領域を内蔵メモリ上に配置しています。E10A でブレイク等の機能を使用するためには、この様にプログラムを RAM に配置する必要があります。これらのメモリへの割り付けは、 SH7727 フォルダ内に含まれる lnkSet1.sub で指定します。フラッシュ等にプログラムを書き込み ROM 化する場合は、このファイルを変更してください。

### 3.3.1 プログラムのロード

SH7727SE の SDRAM へ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。

- HDIをインストールしたE10A用PCにE10Aを挿入し、ユーザケーブルでE10AとSH7727SEを接続してください。
- E10A用PCの電源を投入し、起動してください。
- HDIを起動してください。
- SH7727SEの電源を投入してください。
- PCの画面にダイアログ（図3.6参照）が表示されるので、SH7727SEのリセットスイッチ（SW1）をONにし、CPUをリセット後、「OK」ボタンをクリックまたは、<Enter>キーを押してください。
- メニューバーのView CommandLineを選択し、ウインドを開き（図3.7参照）、左上のBatchFileボタンをクリックし、SH7727フォルダ内の7727E10A.hdcを指定してください。この操作によりBSCが設定され、SDRAMへのアクセスが可能となります。
- ファイルメニューからLoadProgram...を選択し、Load Programダイアログボックスで、SH7727フォルダ内のdebugger.ABSを指定してください。

以上の操作で、サンプルプログラムを SH7727SE の SDRAM 上にロードすることができます。

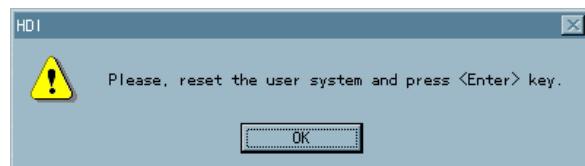


図 3.6 リセット要求ダイアログ

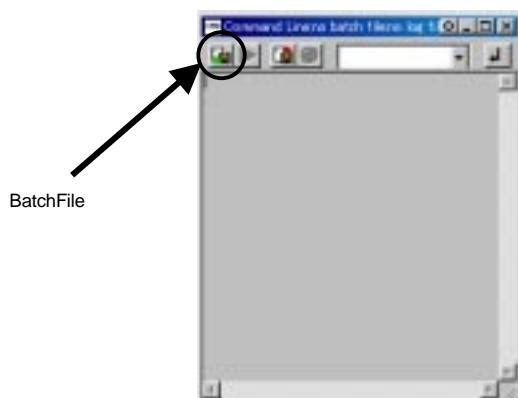


図 3.7 コマンドライン入力

### 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 ウィンドウを開きます。ウィンドウ内の該当レジスタ（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'AC00 0000 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実行されます。

## 3.4 RAM Disk の使用方法

Windows 2000 を用いた場合を例に以下に説明します。

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 SH7727SE 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。

コントロール転送およびパ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ーの 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、ディスクドライブの下に HITACHI EX RAM Disk USB Device が表示されます。その結果、ホスト PC は SH7727SE を記憶デバイスとして認識し、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ウが開かれるので、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。

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 SH7727SE を USB 接続の RAM-Disk として使用できます。



## 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 SH7727SE 上で動作し SH7727SE が RAM-Disk として動作します。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。SH7727 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、USBFI0、USBFI1 の 2 種類ですが、本サンプルプログラムでは USBFI0 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。
- パルクアウト転送でホストコントローラからデータを受信することができます。
- パルケイン転送でホストコントローラにデータを送信することができます。
- SCSIコマンドに対応するRAM-Diskとして動作します。

### 4.1 状態遷移図

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 3 つの状態に遷移します。

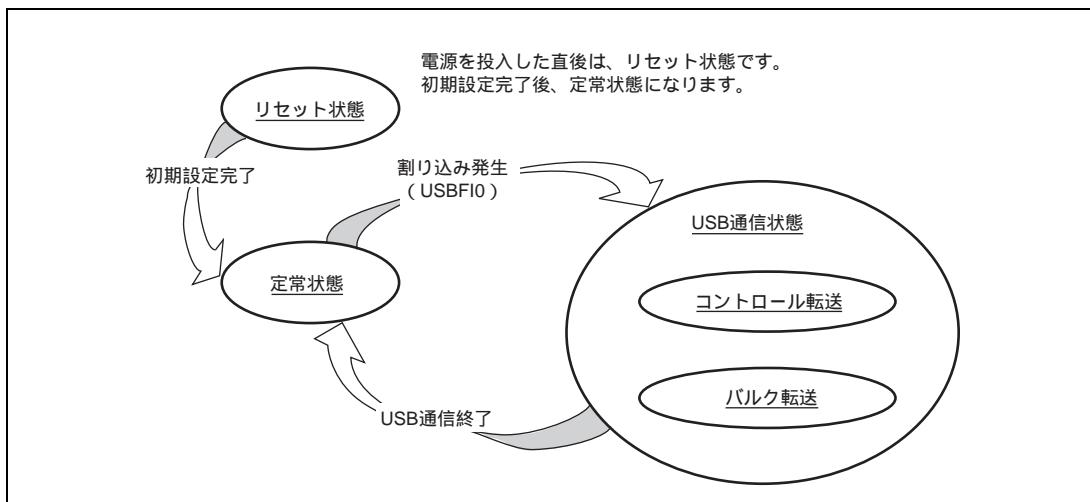


図 4.1 状態遷移図

## 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SH7727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なります。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(USBIFR0)によって示される計8種類です。割り込み要因が発生すると、USBIFR0の対応するビットに1がセットされます。

### 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。

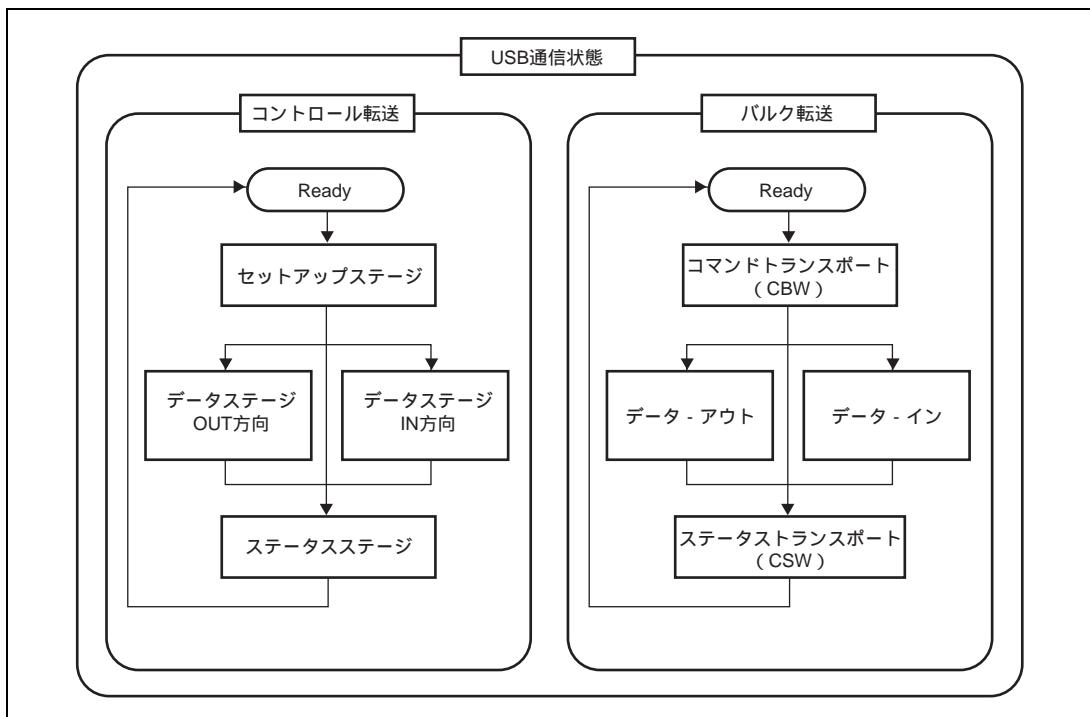


図 4.2 USB 通信状態

#### 4.2.1 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。

#### 4.2.2 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 Mass Storage Class ( 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 Mass Storage Class ( 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 Mass Storage Class( Bulk-Only Transport )のステージは「コマンドトランSPORT(CBW)」「データトランSPORT」「ステータストランSPORT ( CSW )」に分類することができます。

### 4.3 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。図4.3に各ファイルの関係を階層構造で示します。

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
DoRequest.c	ホストが発行するセットアップコマンドの処理
DoRequestBOT_StorageClass.c	Mass Storage Class ( Bulk-Only Transport ) クラスコマンドの処理
DoControl.c	コントロール転送を実行
DoBulk.c	バルク転送を実行
DoBOTMSClass.c	Mass Storage Class ( Bulk-Only Transport ) を実行
DoSCSICommand.c	SCSI コマンドの解析および処理
AsmFunction.src	スタッカの設定
SH7727.h	SH7727 レジスタ定義
SysMemMap.h	SH7727SE のメモリマップのアドレス定義
CatProType.h	プロトタイプ宣言
CatTypedef.h	USB ファームウエアで使用する基本の構造体定義
CatBOTTypedef.h	Bulk-Only Transport 用構造体定義
CatSCSITypedef.h	SCSI 用構造体定義
SetUsbInfo.h	USB 対応に必要な変数の初期設定
SetBOTInfo.h	Bulk-Only Transport 対応に必要な変数の初期設定
SetSCSIIInfo.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
SetMacro.h	マクロ定義

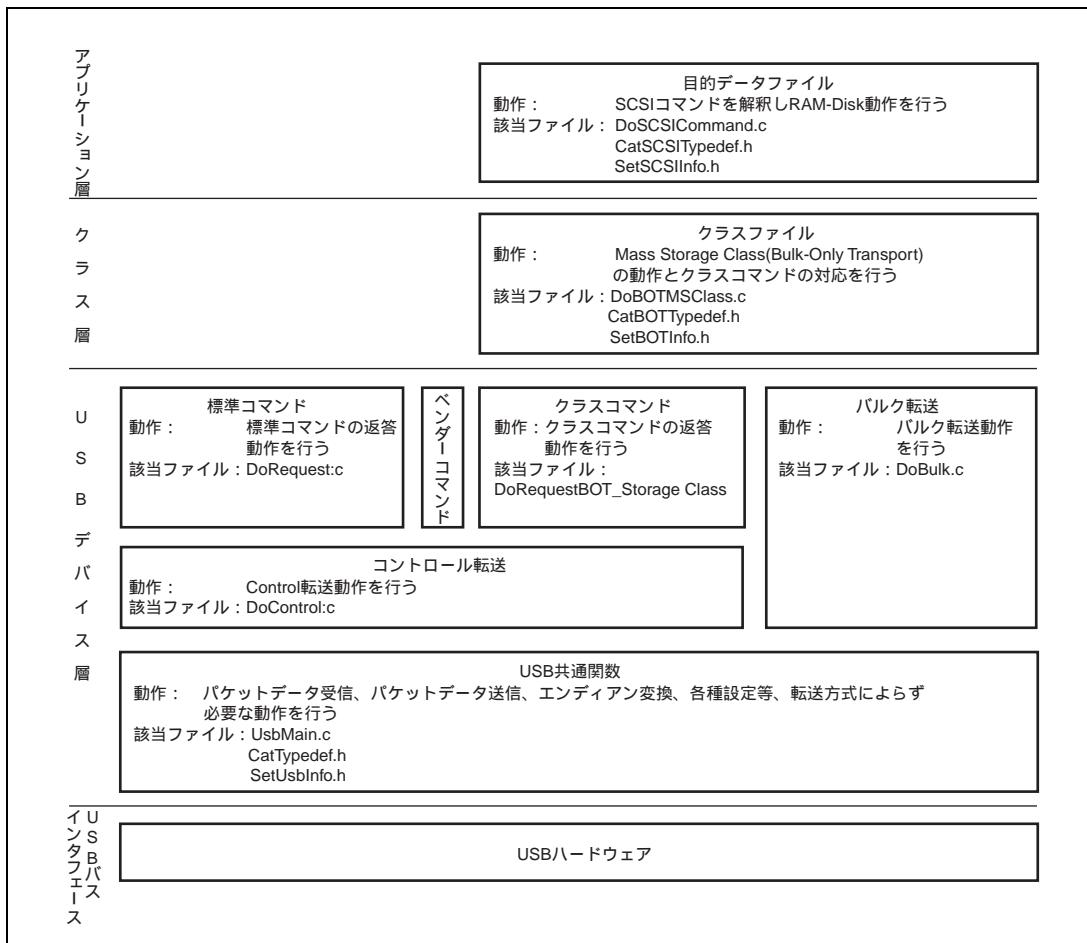


図 4.3 ストレージクラス (BOT) ファームウェアの階層構造

## 4.4 関数の機能

表 4.2 に各ファイルに含まれる関数とその機能を示します。

表 4.2-1 StartUp.c

格納ファイル	関数名	機能
StartUp.c	CallResException	リセット例外に対応する動作をし、引き続き実行する関数を呼び出す
	CallGeneralException	TLB ミス発生以外の一般例外に対応する関数を呼び出す
	CallTLBMissException	TLB ミス発生に対応する関数を呼び出す
	CallInterrupt	割り込み要求に対応する関数を呼び出す
	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
	_INITSCT	初期値がある変数を、RAM のワークエリアにコピー
	InitMemory	バルク通信で使用する RAM 領域をクリア
	InitSystem	USB バスのプルアップ制御

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 SH7727 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 4.2-2 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
	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
	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 のクリアを行う
	ConvRealign	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 4.2-3 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
	DecVendorCommands	ベンダコマンドの対応を行う

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 ( ベンダ:日立 ) 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、DecVendorCommands 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 4.2-4 DoRequestBOT\_StorageClass.c

格納ファイル	関数名	機能
DoRequestBOT_StorageClass.c	DecBOTClass Commands	USB Mass Storage Class ( Bulk-Only Transport ) コマンドの対応を行う

Mass Storage Class ( 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ー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 4.2-5 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

コントロール転送の割り込み ( EP0oTS ) が入ると、ActControl がコマンドを取得し、DecStandardCommands でデコードを行います。その後コマンドの種類に応じて、ActControlIn または ActControlOut によってデータステージと、ステータスステージを行います。

## 4. サンプルプログラム概要

表 4.2-6 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

バルク転送に関する処理を行います。Mass Storage Class ( Bulk-Only Transport ) では ActBulkInReady を使用します。

表 4.2-7 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SPORTとステータストランSPORT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SPORTとステータストランSPORTの制御を行う

DoBOTMSClass.c では、Mass Storage Class ( 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 4.2-8 DoSCSICommand.c

格納ファイル	関数名	機能
DoSCSI Command.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

DoSCSICommand.c では、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SPORTまたはステータストランSPORTの準備を行います。

図 4.4 に、表 4.2 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、CallResetException が他の関数を呼び出します。割り込みの発生によって遷移する USB 通信状態では、割り込み関数である CallInterrupt が BranchOfInt を呼び出し、BranchOfInt が他の関数を呼び出します。図 4.4 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。

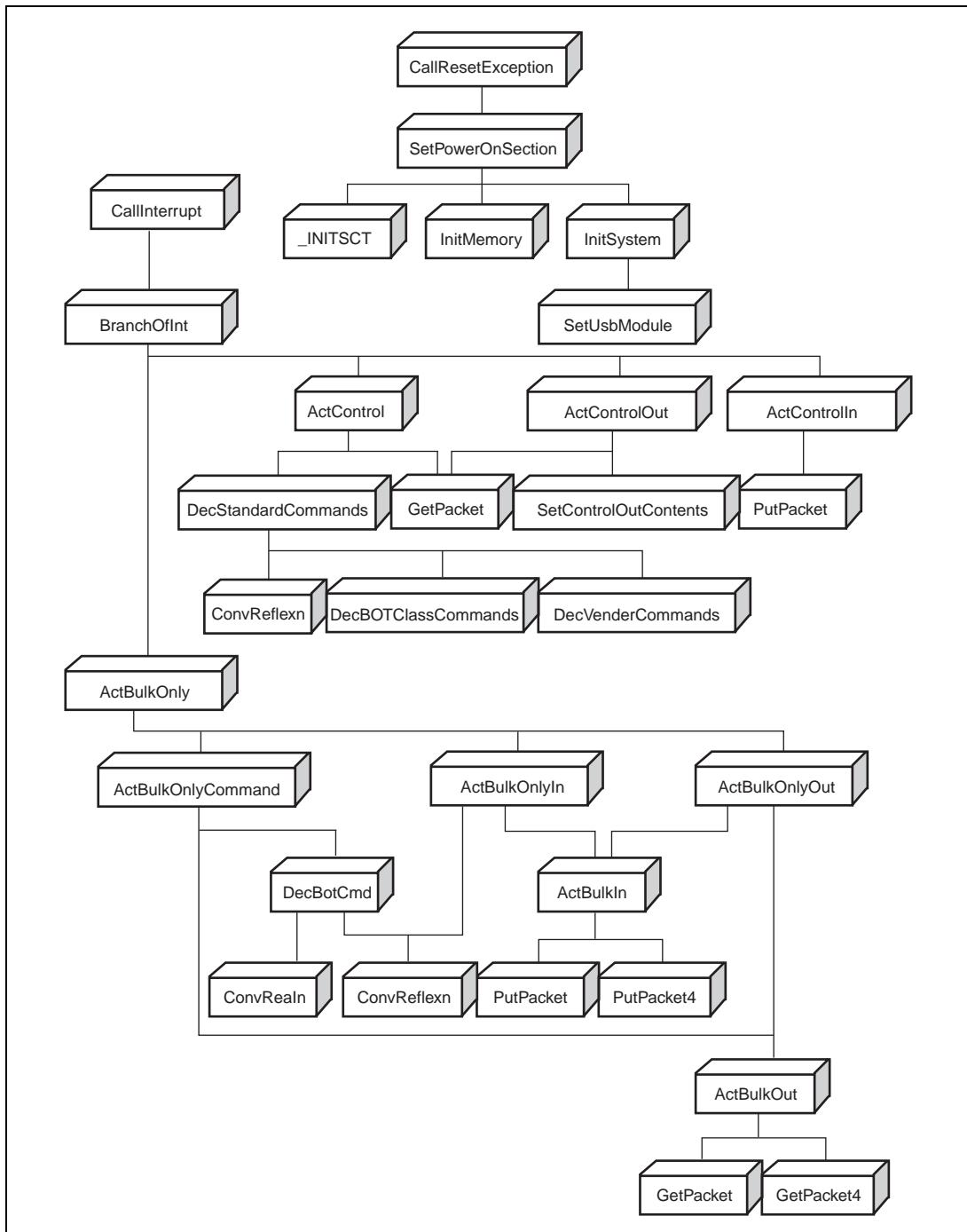


図 4.4 関数の相関関係

## 4.5 RAM-Diskについて

本サンプルプログラムではSH7727SE上のSD-RAMをDisk装置に見立て、ホストPCに対しSH7727SE(ファンクション)はDiskであると報告しています。

ファンクションのDisk装置には図4.5に示すようにマスターブートブロックと、パーティションブートブロックが存在しています。システム立ち上げ時に初期化ルーチンを用いてSD-RAM上のRAM-Disk領域にマスターブートブロックと、パーティションブートブロックを書き込みます。

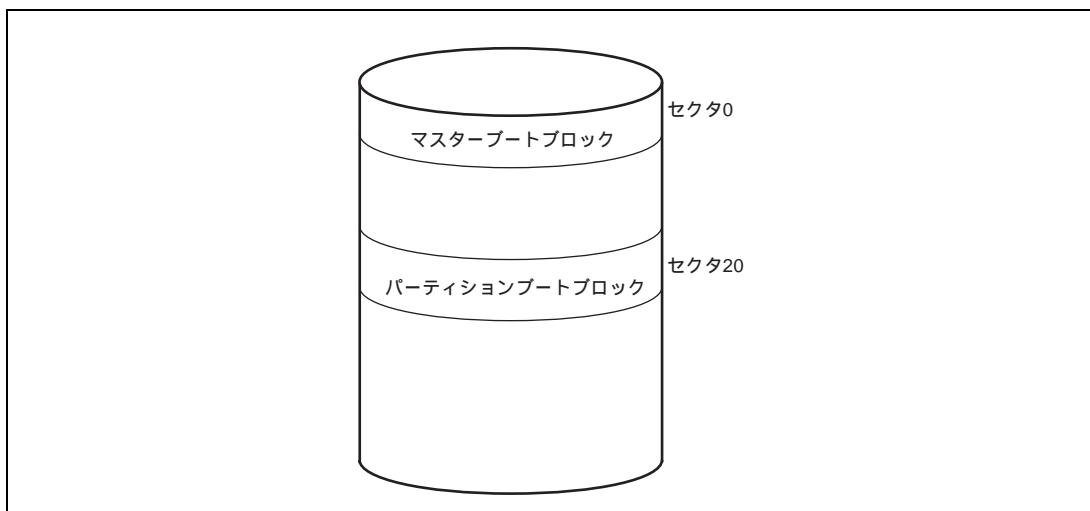


図4.5 Diskの構造

ホストPCからファンクションに対するアクセス(データの保存、読み出し)はSCSIコマンドを使用します。SCSIコマンドの動作を行う場合、図4.5の構造を理解し動作を書く必要があります。

## 4.6 サポートする SCSI コマンドの動作について

本サンプルプログラムがサポートする SCSI コマンドの動作を表 4.3 に示します。

表 4.3 SCSI コマンド動作表

コマンド名	トランスポート名	動作内容
INQUIRY	CBW	コマンドをデコードし、INQUIRY コマンドであることを認識後、ROM に格納してある INQUIRY 情報 (96 バイト) の送信準備を行います。
	データ	ホスト PC に対し INQUIRY 情報をパルクイン転送にて送信します。
	CSW	ホスト PC に対しコマンド実行結果を送信します。送信データが 96 バイト以下であれば正常終了を送信します。
READ CAPACITY	CBW	コマンドをデコードし、READ CAPACITY コマンドであることを認識後、SD-RAM 上に展開してある Disk 装置内にあるパーティションブートブロック内の 1 セクタ当たりのバイト数と、ディスクの総セクタ数に格納されている値を読み出し READ CAPACITY 情報 (8 バイト) の送信準備を行います。
	データ	ホスト PC に対し READ CAPACITY 情報をパルクイン転送にて送信します。
	CSW	ホスト PC に対しコマンド実行結果を送信します。送信データが 8 バイト以下であれば正常終了を送信します。
READ (10)	CBW	コマンドをデコードし、READ (10) コマンドであることを認識後、SD-RAM 上に展開してある Disk 装置内の指定された読み出しセクタから、指定セクタ量のデータ送信準備を行います。
	データ	ホスト PC に対し読み出しセクタのデータをパルクイン転送にて送信します。
	CSW	ホスト PC に対し READ (10) コマンド実行結果を送信します。送信データが読み出しバイト数以下であれば正常終了を送信します。
WRITE (10)	CBW	コマンドをデコードし、WRITE (10) コマンドであることを認識後、SD-RAM 上に展開してある Disk 装置内の指定された書き込みセクタから、指定セクタ量のデータ受信準備を行います。
	データ	ホスト PC から書き込みセクタのデータをパルクアウト転送にて受信します。
	CSW	ホスト PC に対し正常終了を送信します。
REQUEST SENSE	CBW	コマンドをデコードし、REQUEST SENSE コマンドであることを認識後、返答値 (直前の SCSI コマンドを実行した結果) の送信準備を行います。
	データ	ホスト PC に対し返答値をパルクイン転送にて送信します。
	CSW	ホスト PC に対し本コマンド実行結果を送信します。送信データが 18 バイト数以下であれば正常終了を送信します。
PREVENT ALLOW MEDIUM REMOVAL	CBW	コマンドをデコードし、PREVENT ALLOW MEDIUM REMOVAL コマンドであることを認識後、ホスト PC に対し正常終了の送信準備を行います。本サンプルソフトの記録メディアは SD-RAM であり常に着脱不能です。このため本コマンドの返答は常に正常終了となります。
	データ	本コマンドにデータトランスポートは存在しません。
	CSW	ホスト PC に対し正常終了を送信します。

## 4. サンプルプログラム概要

コマンド名	トランSPORT名	動作内容
TEST UNIT READY	CBW	コマンドをデコードし、TEST UNIT READY コマンドであることを認識後、ホスト PC に対し正常終了の送信準備を行います。本サンプルソフトの記録メディアは SD-RAM でありプログラム実行中は常にアクセス可能となっています。このため本コマンドの返答は常に正常終了となります。
	データ	本コマンドにデータトランSPORTは存在しません。
	CSW	ホスト PC に対し正常終了を送信します。
VERIFY ( 10 )	CBW	コマンドをデコードし、VERIFY ( 10 ) コマンドであることを認識後、ホスト PC に対し正常終了の送信準備を行います。本サンプルソフトの記録メディアは SD-RAM でありプログラム実行中は常にアクセス可能となっています。このため本コマンドの返答は常に正常終了となります。
	データ	本コマンドにデータトランSPORTは存在しません。
	CSW	ホスト PC に対し正常終了を送信します。
MODE SENSE ( 10 )	CBW	コマンドをデコードし、MODE SENSE ( 10 ) コマンドであることを認識後、ホスト PC に対しモードパラメータの送信準備を行います。 本サンプルソフトではモードパラメータヘッダのみ値を用意しています。
	データ	ホスト PC に対しモードパラメータをパルクイン転送にて送信します。
	CSW	ホスト PC に対しコマンド実行結果を送信します。送信データが 8 バイト以下であれば正常終了を送信します。
未サポートコマンド	CBW	コマンドをデコードし、未サポートコマンドであれば、REQUEST SENSE の返答値に INVALID FIELD IN CDB を設定後、データトランSPORTの準備を行います。
	データ	ホスト PC がパルクイン転送にてデータを要求した場合、ホストが要求した量と同量のデータ ( 0x00 ) を送信します。
		ホスト PC がパルクアウト転送にてデータを送信した場合、受信バイト数のカウントを行います。
		データトランSPORTが無い場合、何も動作は行いません。
	CSW	ホスト PC に対しコマンドフェイルを送信します。

## 4.7 エラー時の処理について

Mass Storage Class ( Bulk-Only Transport ) の転送を行う際、ホスト PC とファンクション間で発生するエラーとエラー時のファンクション側の対応動作を示します。

Bulk-Only Transport の規格では次にあげるエラーケースが規定されています。

- CBWが有効でない場合。
- ホストの期待とファンクションが意図する動作 ( SCSIコマンドで指定された動作 ) の相違 ( 10ケース )。

以上の 2 種類があります。これ以外の状態については規格書には定められていません。

ホスト-ファンクション間のデータ転送については表 4.4 と表 4.5 に示す 13 種類の状態が存在します。このうち CASE ( 1 ) ( 6 ) ( 12 ) は正常な転送状態です。

表 4.4 ホスト-ファンクション間のデータ転送状態

		ホストは		
		データ転送なしを期待	ファンクションからのデータ受信を期待	ファンクションへのデータ送信を期待
ファンクションは	データ転送なしを意図	(1) Hn = Dn	(4) Hi > Dn	(9) Ho > Dn
	ホストへのデータ送信を意図	(2) Hn < Di	(5) Hi > Di	(10) Ho < > Di
			(6) Hi = Di	
			(7) Hi < Di	
	ホストからのデータ受信を意図	(3) Hn < Do	(8) Hi < > Do	(11) Ho > Do
				(12) Ho = Do
				(13) Ho < Do

表 4.5 ホスト-ファンクション間データ転送状態解説

CASE	ホスト-ファンクション間での関係
1	ホストはデータ転送なしを期待し、ファンクションもデータ転送なしを意図する場合
2	ホストはデータ転送なしを期待し、ファンクションはホストへのデータ送信を意図する場合
3	ホストはデータ転送なしを期待し、ファンクションはホストからのデータ受信を意図する場合
4	ホストはファンクションからのデータ受信を期待し、ファンクションはホストへのデータ転送なしを意図する場合
5	ホストが期待したファンクションからのデータ受信数より、ファンクションがホストへ送信するデータ数が少ない場合
6	ホストが期待したファンクションからのデータ受信数と、ファンクションがホストへ送信するデータ数が同じ場合
7	ホストが期待したファンクションからのデータ受信数より、ファンクションがホストへ送信するデータ数が多い場合
8	ホストはファンクションからのデータ受信を期待し、ファンクションはホストからのデータ受信を意図する場合
9	ホストはファンクションへのデータ送信を期待し、ファンクションはデータ転送なしを意図する場合
10	ホストはファンクションへのデータ送信を期待し、ファンクションはホストへのデータ送信を意図する場合
11	ホストが期待したファンクションへのデータ送信数より、ファンクションがホストから受信するデータ数が少ない場合
12	ホストが期待したファンクションへのデータ送信数と、ファンクションがホストから受信するデータ数が同じ場合
13	ホストが期待したファンクションへのデータ数より、ファンクションがホストから受信するデータ数が多い場合

## 4. サンプルプログラム概要

---

表 4.6 に発生する可能性のあるエラー状況例を示します。

表 4.6 エラー状況例

CASE	エラー状況
2	ホストから READ コマンドが発行される際、USB のデータトランSPORTで転送するデータ数が 0 で、SCSI コマンドで指定されたデータ数が 0 以外の場合
3	ホストから WRITE コマンドが発行される際、USB のデータトランSPORTで転送するデータ数が 0 で、SCSI コマンドで指定されたデータ数が 0 以外の場合
4	ホストから READ コマンドが発行される際、USB のデータトランSPORTで転送するデータ数が 0 で、SCSI コマンドで指定されたデータ数が 0 の場合
5	ホストから READ コマンドが発行される際、USB のデータトランSPORTで転送するデータ数より、SCSI コマンドで指定されたデータ数が少ない場合
7	ホストから READ コマンドが発行される際、USB のデータトランSPORTで転送するデータ数より、SCSI コマンドで指定されたデータ数が多い場合
8	ホストから WRITE コマンドが発行されたのに、ホストが USB のデータトランSPORTでデータを要求する場合
9	ホストから WRITE コマンドが発行される際、USB のデータトランSPORTで転送するデータ数が 0 以外で、SCSI コマンドで指定されたデータ数が 0 の場合
10	ホストから READ コマンドが発行されたのに、ホストが USB のデータトランSPORTでデータを送ってくる場合
11	ホストから WRITE コマンドが発行される際、USB のデータトランSPORTで転送するデータ数より、SCSI コマンドで指定されたデータ数が少ない場合
13	ホストから WRITE コマンドが発行される際、USB のデータトランSPORTで転送するデータ数より、SCSI コマンドで指定されたデータ数が多い場合

エラー状況に対するファンクションの対応動作は表 4.7 のようになります。

表 4.7 エラー対応動作表

CASE	エラー時におけるデータトランSPORTでのファンクション対応動作
2、3	<ul style="list-style-type: none"> <li>CSW のステータスに 0x02 を設定する。</li> </ul>
4、5	<ul style="list-style-type: none"> <li>ファンクションは dCBWDataTransferLength で示されたデータ長になるようにデータを付加し、ホストにデータを送信する。</li> <li>CSW の dCBWDataResidue にデータトランSPORTで付加したデータ数を設定する。</li> <li>CSW のステータスに 0x01 を設定する。</li> </ul>
7、8	<ul style="list-style-type: none"> <li>ファンクションは dCBWDataTransferLength で示されたデータ長まで、ホストにデータを送信する。</li> <li>CSW のステータスに 0x02 を設定する。</li> </ul>
9、11	<ul style="list-style-type: none"> <li>ファンクションは dCBWDataTransferLength で示されたデータ長分、データを受信する。</li> <li>データトランSPORTで受信したデータ数とファンクションで処理したデータ数の差を CSW の dCBWDataResidue に設定する。</li> <li>CSW のステータスに 0x01 を設定する。</li> </ul>
10、13	<ul style="list-style-type: none"> <li>ファンクションは dCBWDataTransferLength で示されたデータ長分、データを受信する。</li> <li>CSW のステータスに 0x02 を設定する。</li> </ul>

データ転送時のエラー処理フローは、図 4.6、4.7、4.8 のようになります。

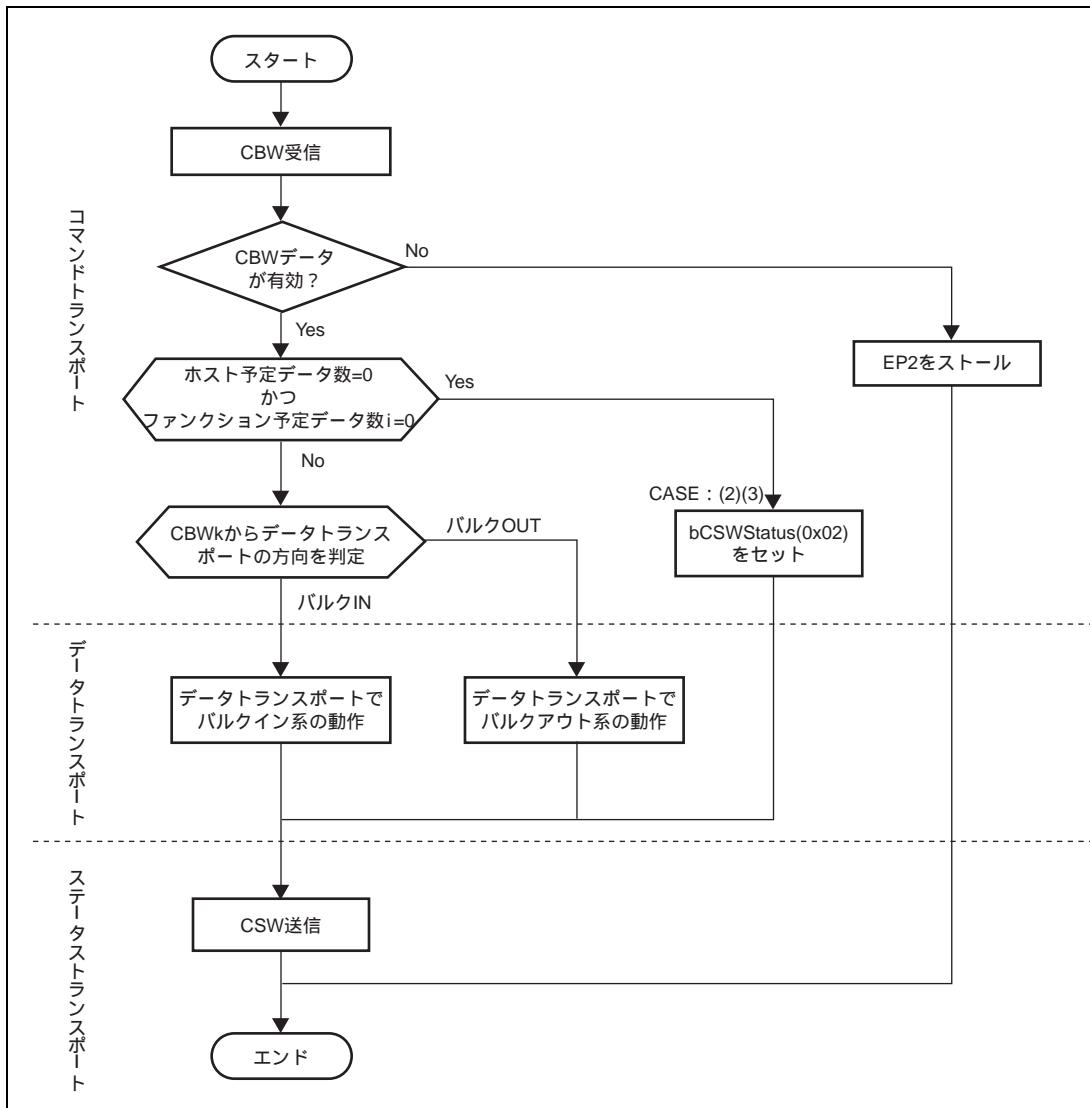


図 4.6 データ転送時のエラー処理フロー（1）

#### 4. サンプルプログラム概要

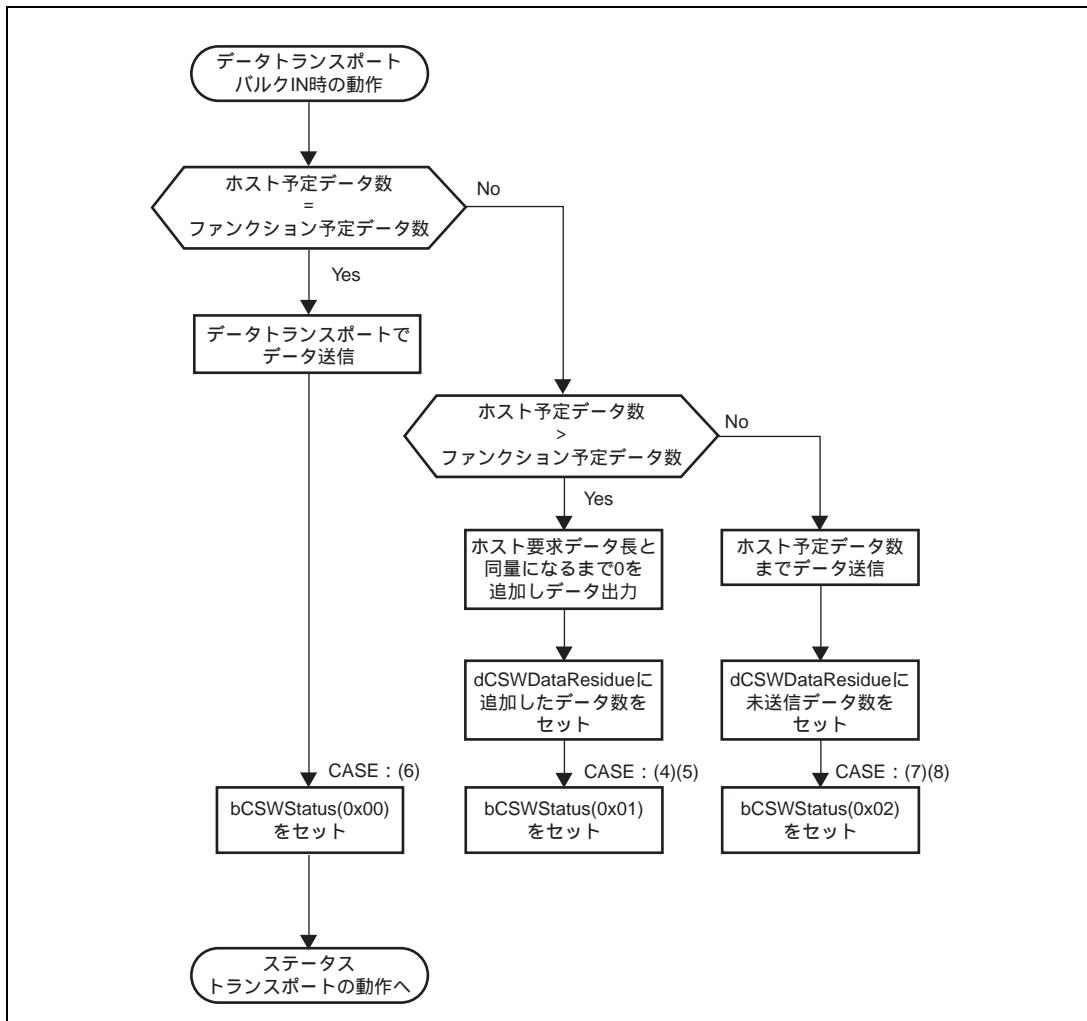


図 4.7 データ転送エラー発生時の処理フロー（2）

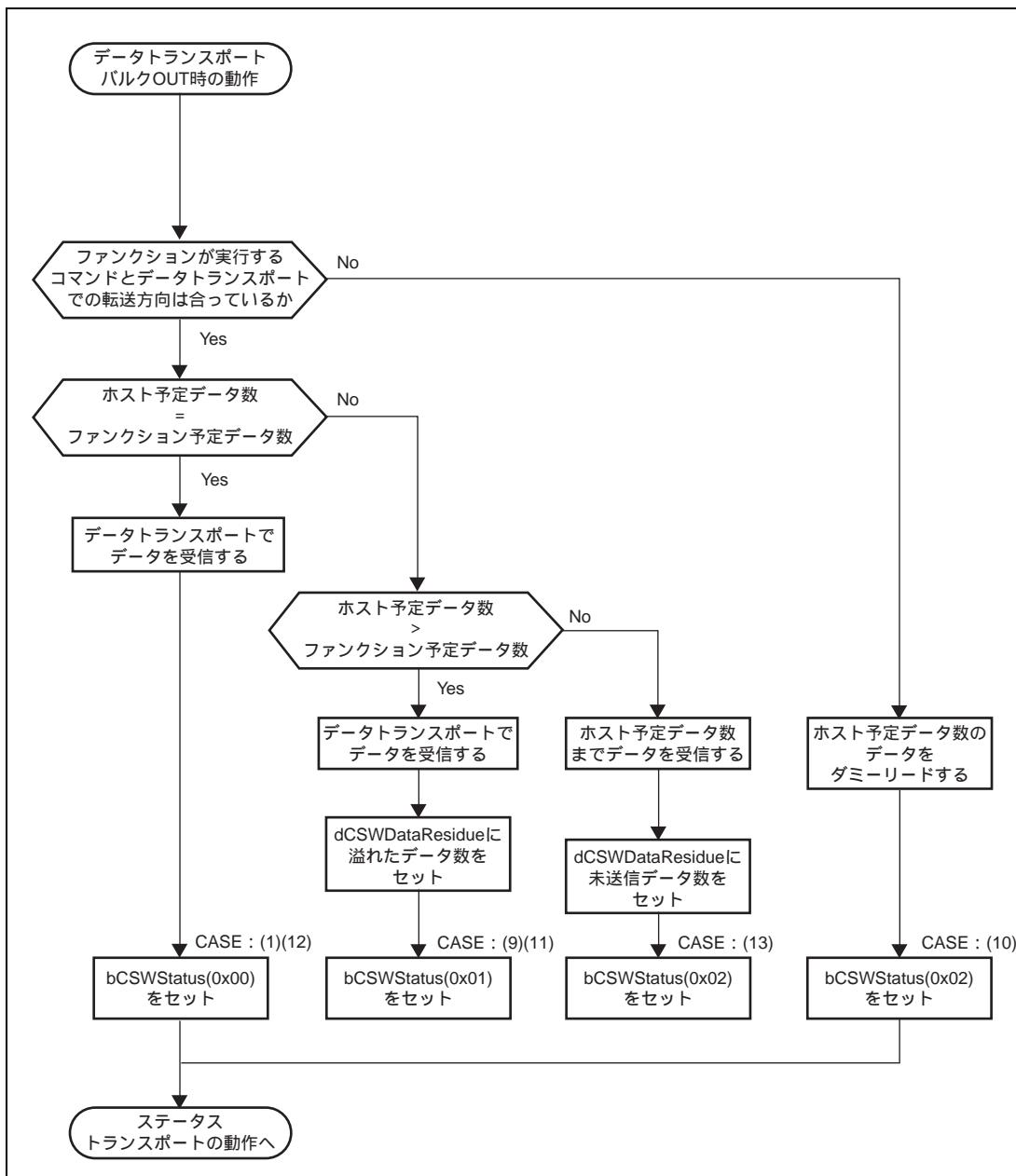


図 4.8 データ転送エラー発生時の処理フロー（3）

## 4. サンプルプログラム概要

Mass Storage Class ( Bulk-Only Transport ) の転送を行う際、CBW トランSPORTで一連のデータ転送が始まり、ホスト PC に CSW トランSPORTで一連の転送結果 (ステータス) を返します。このためデータ転送処理を行う際に、CSW トランSPORTで返答する内容も作成します。返答内容としては 2 項目あり、転送処理の結果を表す dCSWStatus と、データ転送エラーバイト数を表す dCSWDataResidue があります。

本サンプルプログラムでは、この 2 項目を作成するために、

- CBW パケットの dCBWDataTransferLength フィールド
- CSW パケットの dCSWDataTransferResidue フィールド

を使用します。

CBW パケットの dCBWDataTransferLength フィールドはホスト PC が指定するデータトランSPORTで扱うデータバイト数を入れる変数として使用します。

CSW パケットの dCSWDataTransferResidue フィールドはファンクションがデータトランSPORTで扱うデータバイト数を入れる変数として使用します。

CBW トランSPORTが終了すると、dCBWDataTransferLength フィールドと dCSWDataTransferResidue フィールドにはデータトランSPORTで扱う予定データバイト数がそれぞれ格納されます。

データトランSPORTでデータ転送する際にはフロー図で示した流れで動作を行います。

ホスト-ファンクション間でエラーなく処理が行われるときは、データトランSPORTでデータ転送する度に dCBWDataTransferLength フィールドと dCSWDataTransferResidue フィールドの値を転送バイト数分減算します。それ以外の場合は、PC が要求するデータトランSPORTで扱うデータバイト数とファンクションがデータトランSPORTで扱ったデータバイト数の「差」を、CSW パケットの dCSWDataTransferResidue フィールドに設定し、ステータストランSPORTに移行します。

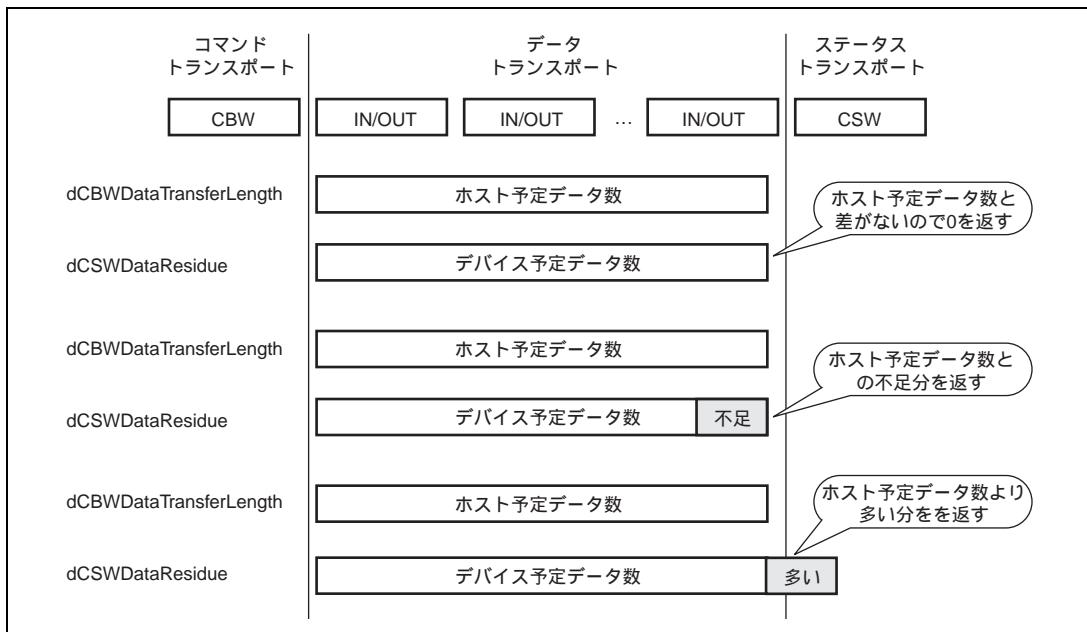


図 4.9 Bulk-Only Transport における各ステージ

## 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。次にリセット割り込み関数 CallResetException が呼び出されリセット例外処理を行い、関数 SetPowerOnSection を呼び出します。図 5.1 にリセット割り込み発生から定常状態までのフローチャートを示します。

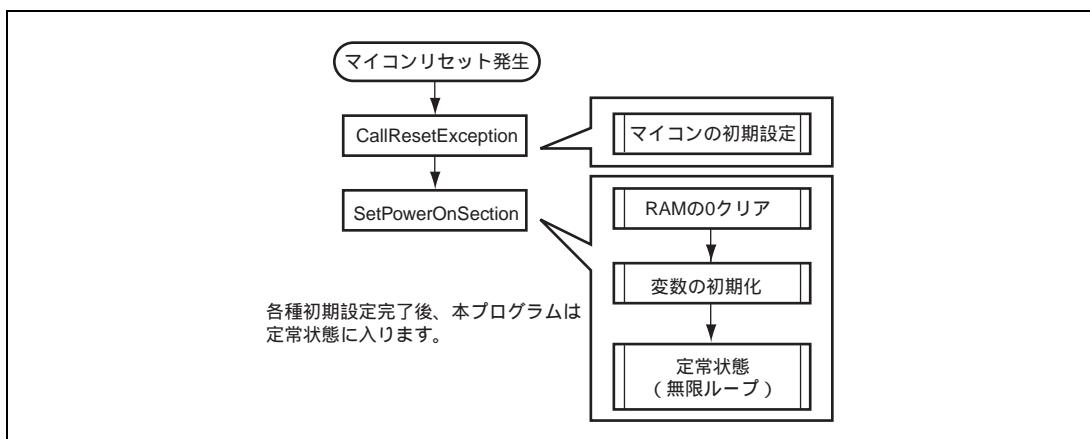


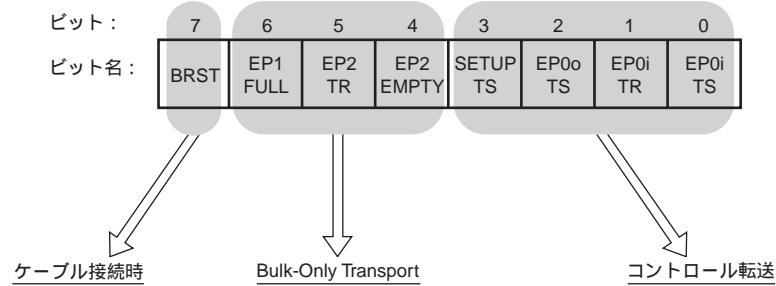
図 5.1 メインループ

### 5.2 割り込みの種類

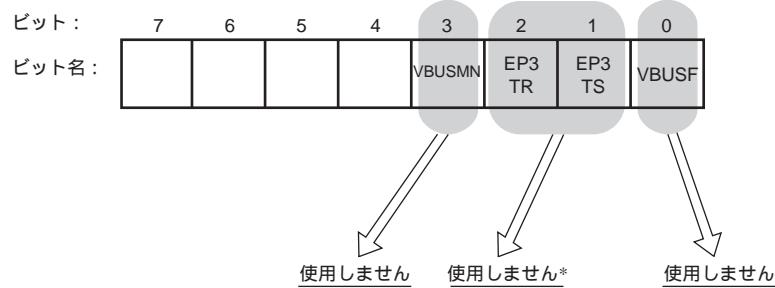
4 章で説明したように本サンプルプログラムで使用する割り込みは割り込みフラグレジスタ 0 ( USBIFR0 ) によって示される計 8 種類です。割り込み要因が発生すると、割り込みフラグレジスタの対応するビットに 1 がセットされ、CPU に対して USBFI0 割り込みを要求します。サンプルプログラムでは、この割り込み要求によって割り込みフラグレジスタをリードし、それに対応する USB 通信を行います。図 5.2 に割り込みフラグレジスタと、USB 通信の関係を示します。

## 5. サンプルプログラムの動作

USB割り込みフラグレジスタ0 ( USBIFR0 )



USB割り込みフラグレジスタ1 ( USBIFR1 )



【注】\* サンプルプログラムはインターラプト転送をサポートしていないため、EP3に関する割り込みは使用しません。

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
USBIFR0	0	EPOi TS	ActControlIn、ActControlOut
	1	EPOi TR	ActControlOut
	2	EPOo TS	ActControlIn、ActControlOut
	3	SETUP TS	ActControl
	4	EP2 EMPTY	ActBulkOnly
	5	EP2 TR	ActBulkOnly
	6	EP1 FULL	ActBulkOnly
	7	BRST	ActBusReset

EPOi TS と EPOo 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、「5.4 コントロール転送」をご覧ください。

SH7727 の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 ケーブル接続時 (BRST) 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、汎用専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事を認識します。（図 5.3 参照）

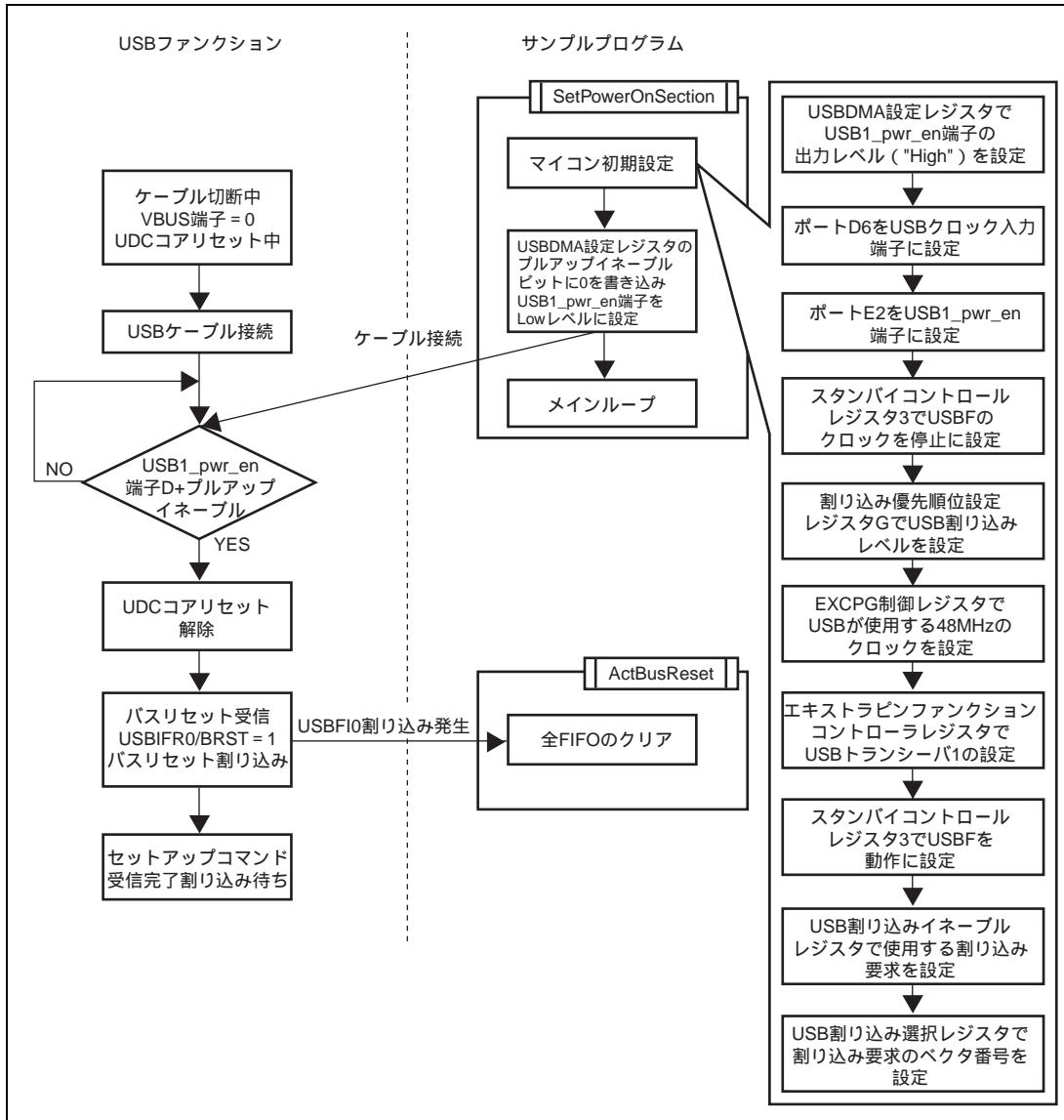


図 5.3 ケーブル接続時割り込み

## 5.4 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ピット 0~3 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 つに分ける事ができます。（図 5.4 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が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。

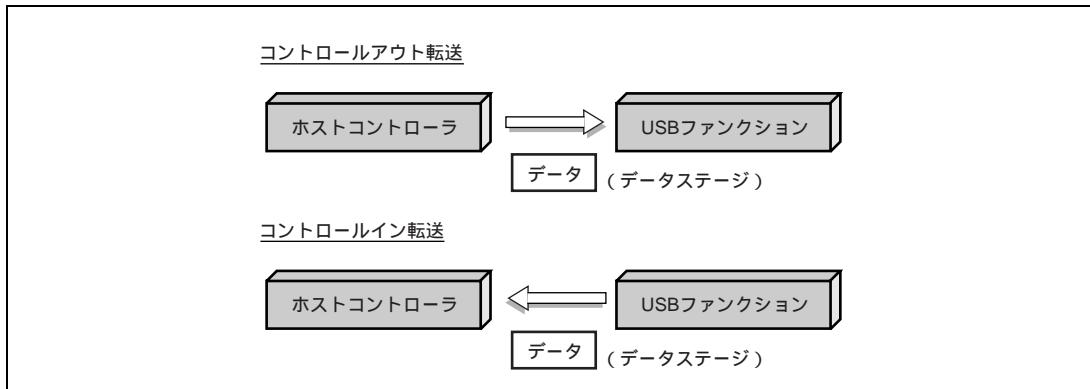


図 5.4 コントロール転送

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 5.5)。また、データステージは、複数のバ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事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 5.5 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タート(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の動作

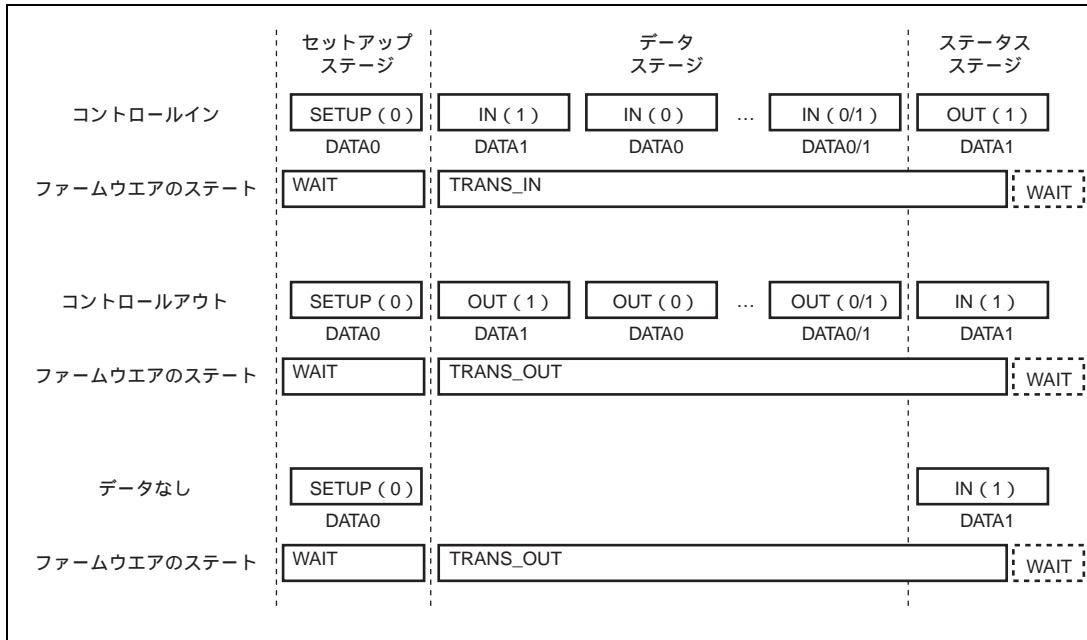


図 5.5 コントロール転送における各ステージ

### 5.4.1 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)  
Get Max LUN (クラスコマンド)
- TRANS\_OUTとなるコマンド . . . . . Bulk-Only Mass Storage Reset (クラスコマンド)

図 5.6 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。

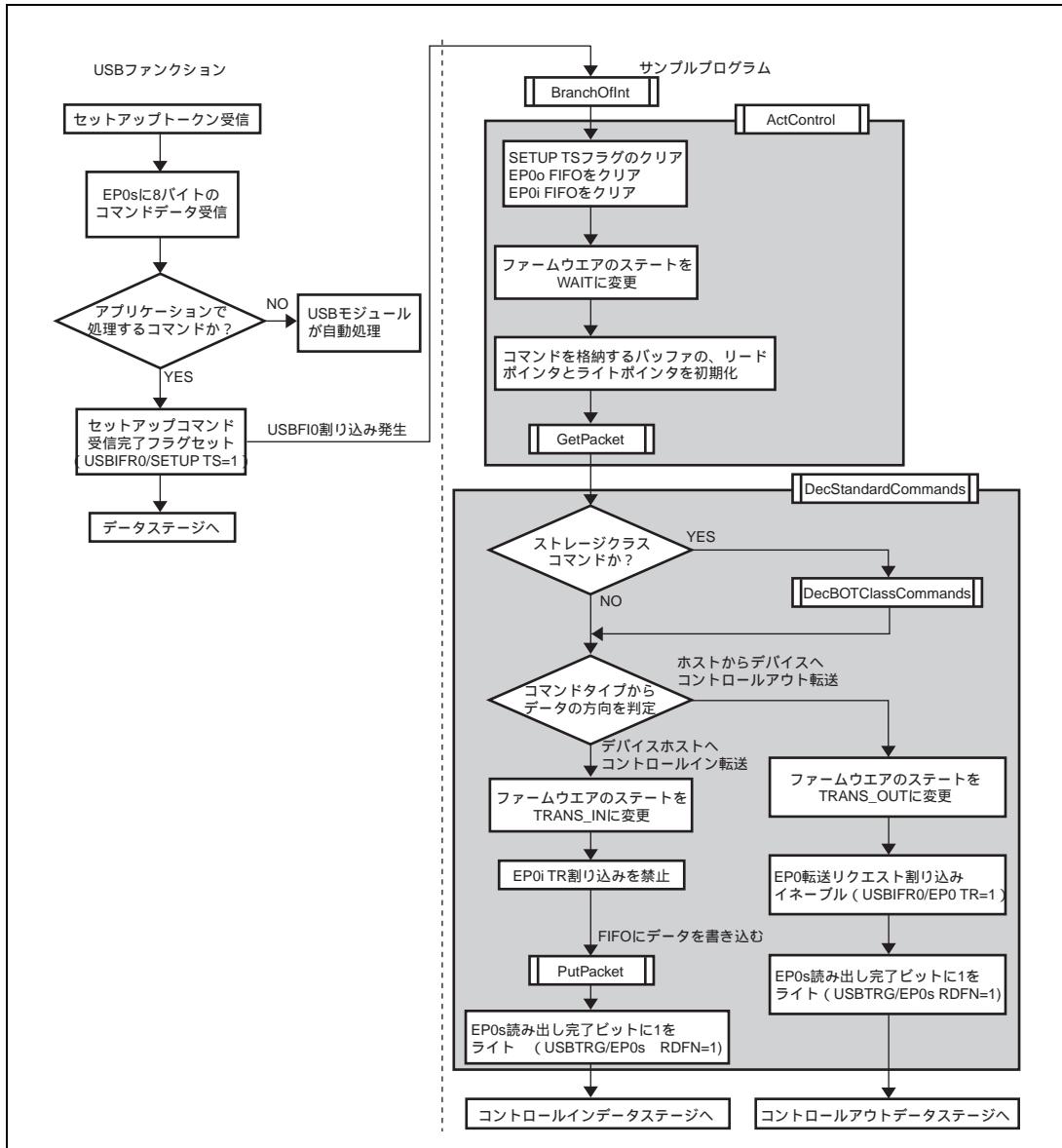


図 5.6 セットアップステージ

## 5. サンプルプログラムの動作

### 5.4.2 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。ファームウエアのステートは、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 5.7、図 5.8 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。

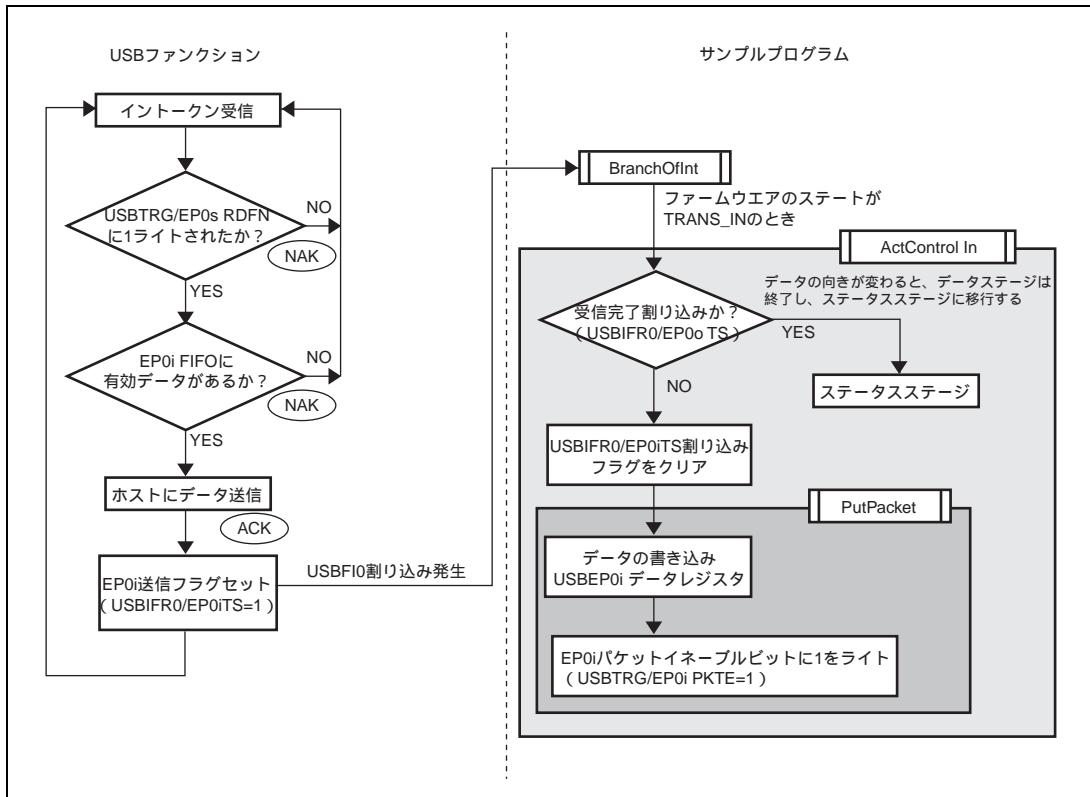


図 5.7 データステージ (コントロールイン転送)

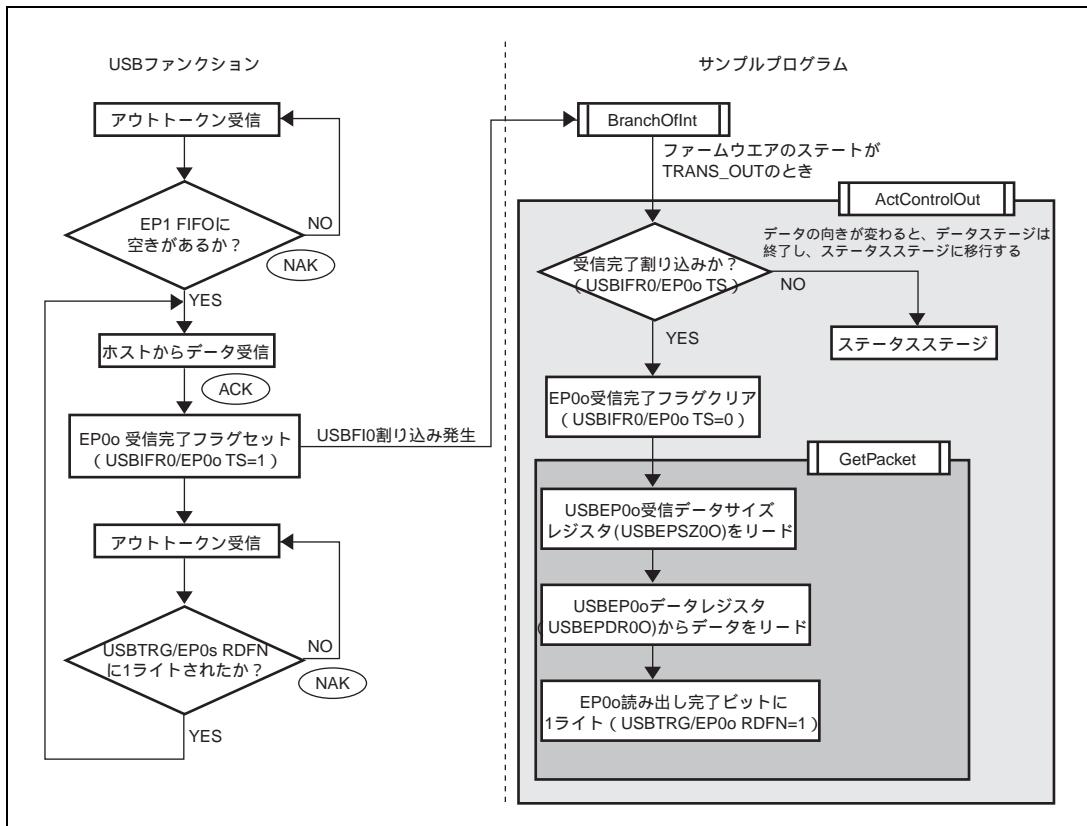


図 5.8 データステージ(コントロールアウト転送)

## 5. サンプルプログラムの動作

### 5.4.3 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。

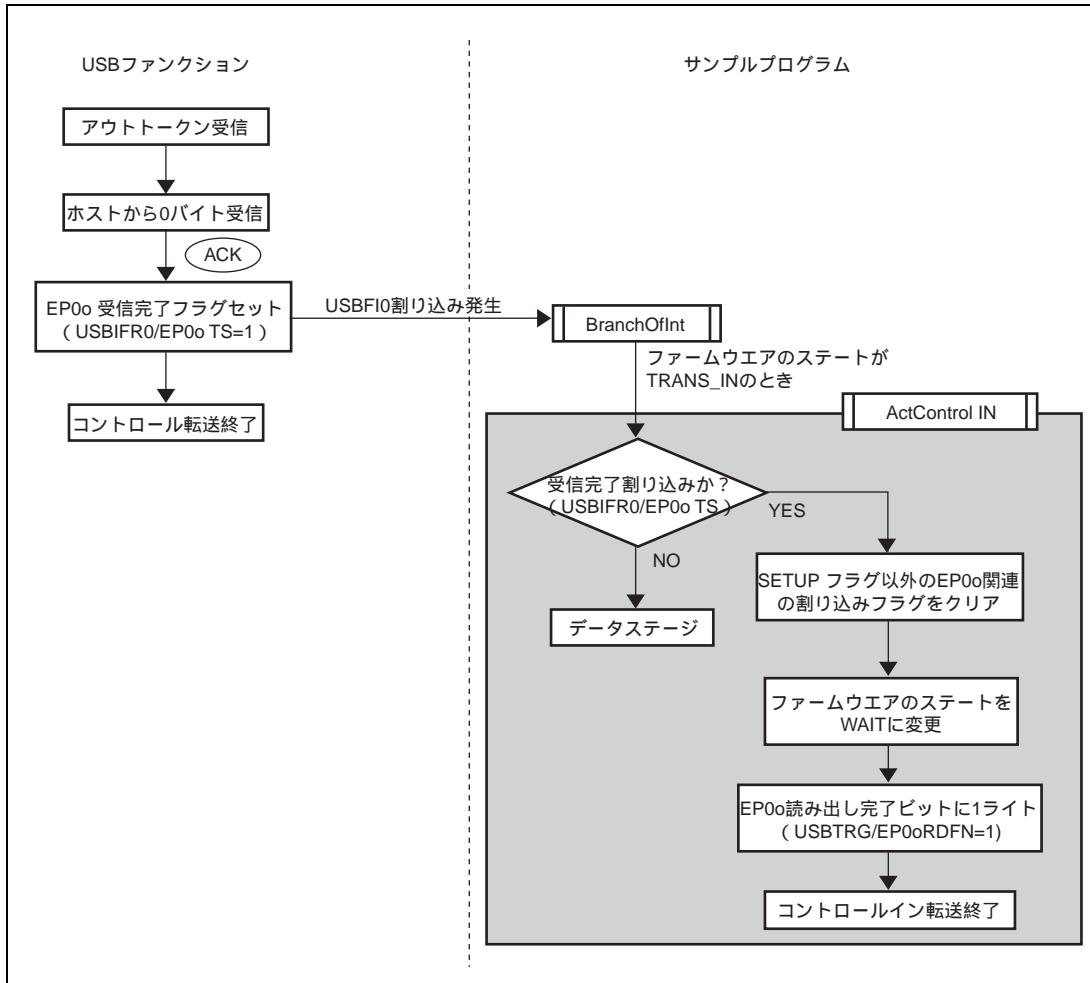


図 5.9 ステータスステージ（コントロールイン転送）

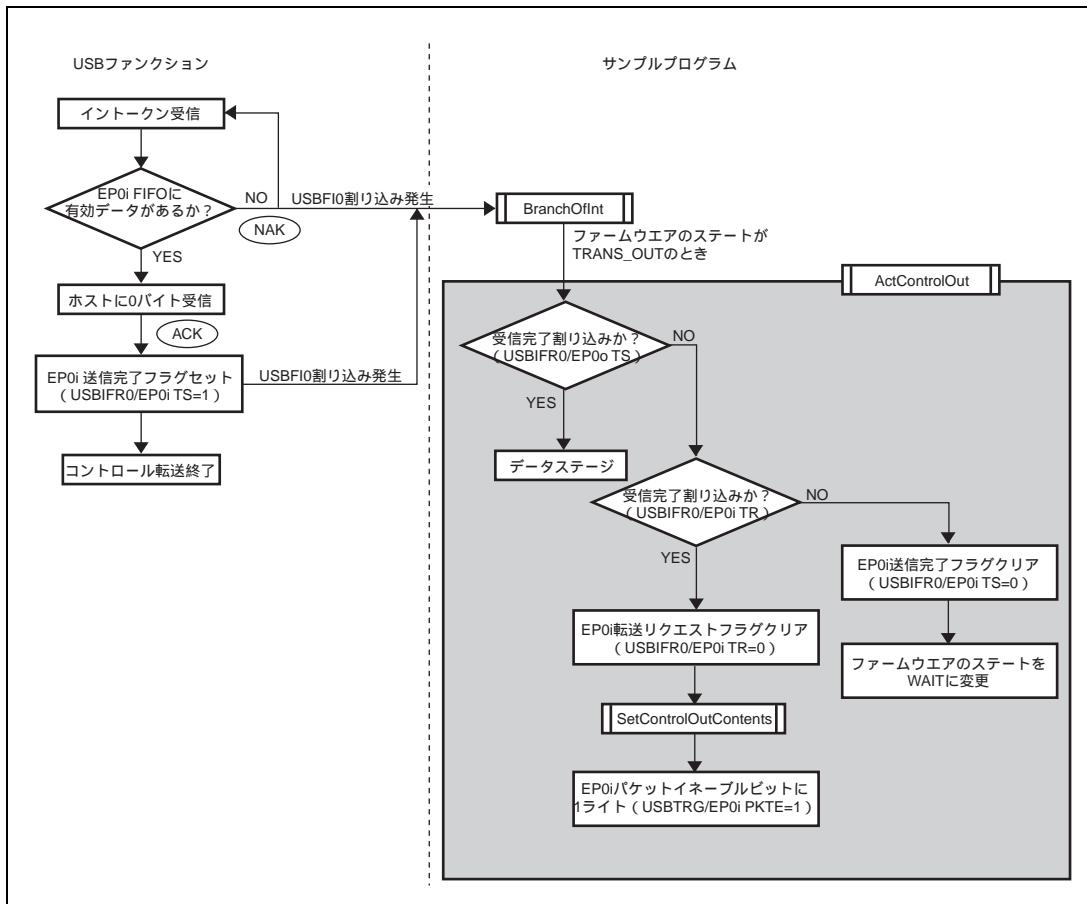


図 5.10 ステータスステージ(コントロールアウト転送)

### 5.5 バルク転送

バルク転送には、割り込みフラグレジスタのビット 4~6 を使用します。バルク転送もデータを送信する向きによって、2 つに分ることができます。（図 5.11 参照）

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をバルクアウト転送、反対の場合をバルクイン転送と呼びます。

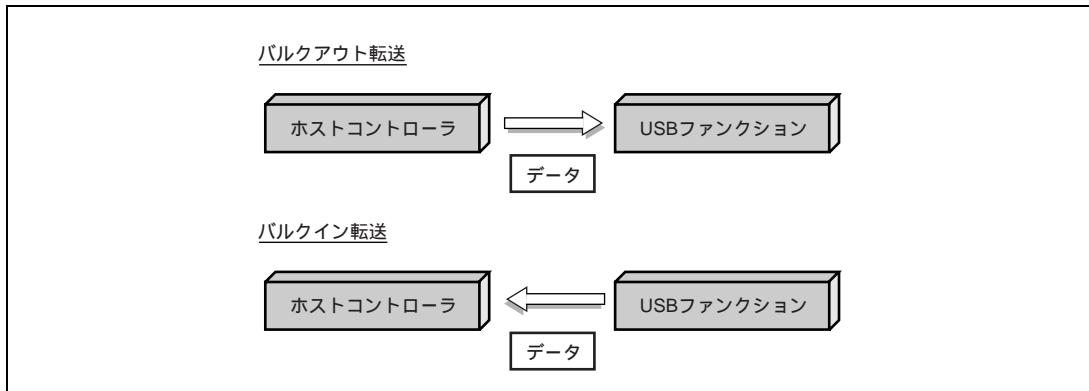


図 5.11 バルク転送

USB Mass Storage Class の Bulk-Only Transport は、バルクイン転送とバルクアウト転送で構成されています。

Bulk-Only Transport は、「コマンドトランSPORT (CBW)」、「データトランSPORT」（ない場合もある）「ステータストラNSPORT (CSW)」の 2 ないし 3 つのステージで構成されます（図 5.12）。また、データトランSPORT は、複数のバストラNSPORT で構成されます。

Bulk-Only Transport では、コマンドトランSPORT (CBW) はバルクアウト転送。ステータストラNSPORT (CSW) はバルクイン転送。データトランSPORT はデータを送信する向きによってバルクイン転送・バルクアウト転送どちらかの転送が行われます。

データトランSPORT でバルクイン・バルクアウトどちらの転送が行われるかは、コマンドトランSPORT で受信する CBW データにより決定します。ファームウェアはデータトランSPORT がバルクイン転送・バルクアウト転送どちらの転送が行われるかをステート (TRANS\_IN、TRANS\_OUT) で管理し（図 5.12 参照）、適切な関数を呼び出す必要があります。

また、データトランSPORT からステータストラNSPORT へのステージの遷移は、ホスト PC がリクエストしたデータトランSPORT での予定データ長のデータを送信または受信する事で、ステータストラNSPORT へステージが遷移します。したがって、ファームウェアがデータトランSPORT で送信または受信したデータ長を管理し、ステージの遷移後ステータストラNSPORT でデータをホスト PC に送信する必要があります。

もし、コマンドトランSPORT で受信する CBW データが有効と認められない場合、エンドポイントをストールし一切のバルク転送を行いません。

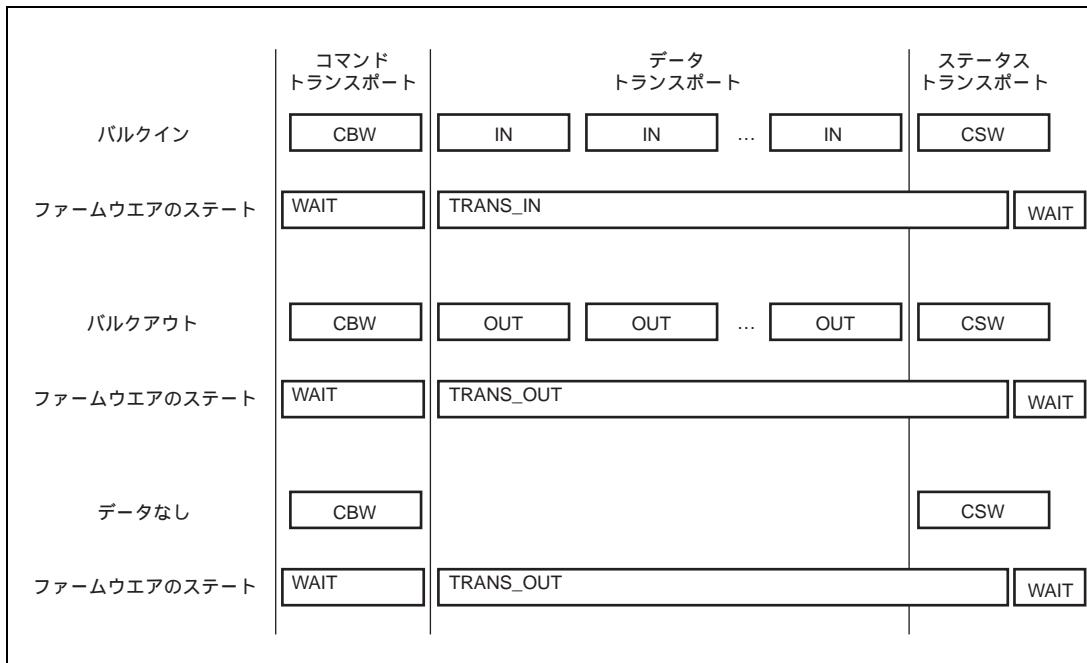


図 5.12 Bulk-Only Transport における各ステージ

### 5.5.1 コマンドトランSPORT

コマンドトランSPORTでは、ホストからファンクションへの CBW データを受信します。

このときファームウェアのステートは WAIT 状態で実行されます。CBW データ受信後このステージでは、次に示す 5 点の処理を行います。

1. CBWデータをEPIデータレジスタからワーク領域に格納。
2. CBWデータの有効判定。
3. CSWデータの準備。
4. CBWデータの内容をデコードし、データトランSPORTで転送するデータがある場合は、データの準備を行う。  
(関数DecBotCmd内にて処理)
5. データトランSPORTがバルクインまたはバルクアウト転送どちらかの区別を行い、ファームウェアのステート (TRANS\_IN、TRANS\_OUT) を決定。

図 5.13 にサンプルプログラムのコマンドトランSPORTにおける動作を示します。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。

## 5. サンプルプログラムの動作

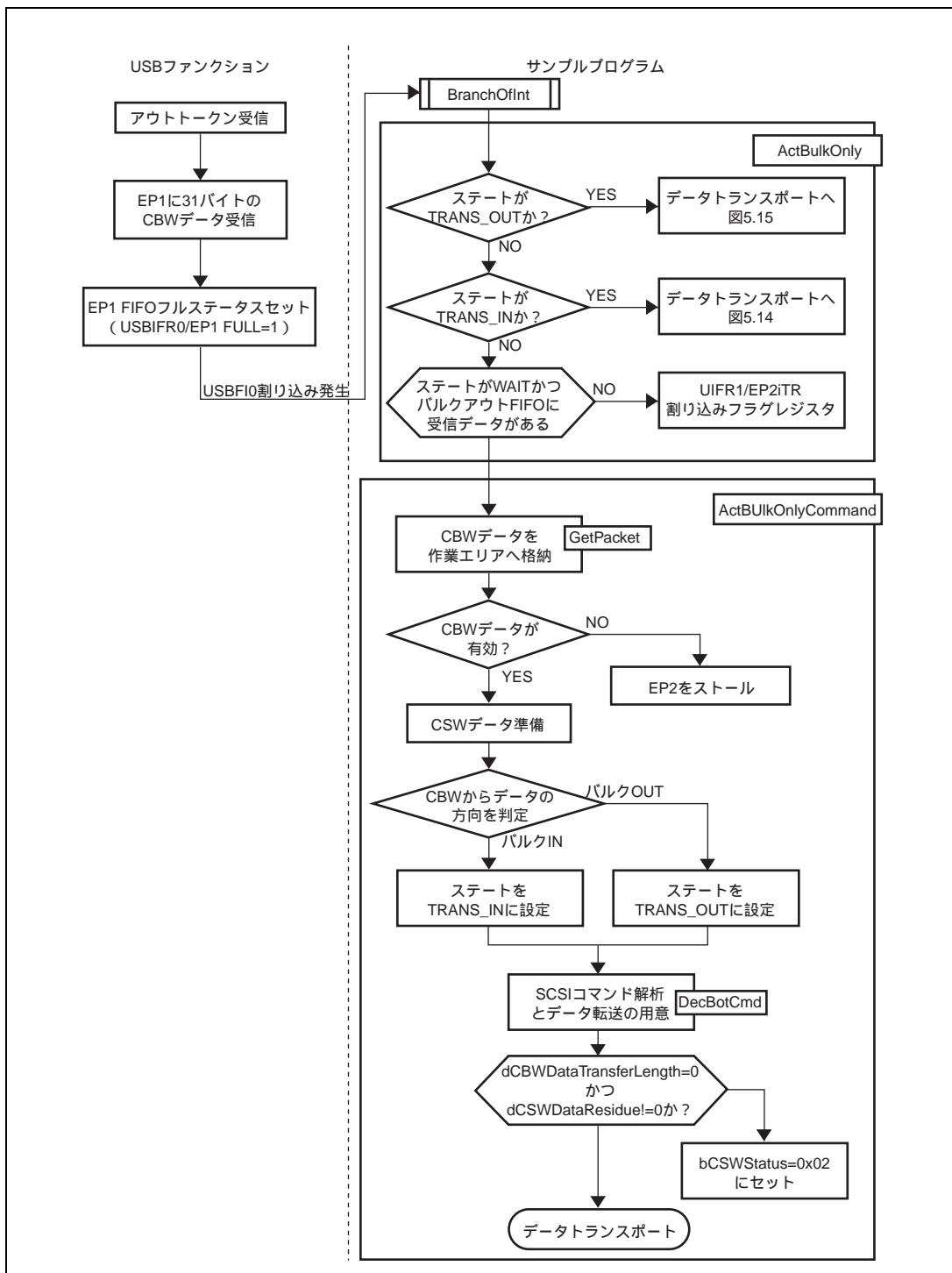


図 5.13 コマンドトランSPORT

### 5.5.2 データトランSPORT

データトランSPORT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。

このときファームウエアのステートは TRANS\_IN または TRANS\_OUT どちらかの状態で実行されます。

ファームウエアのステートが TRANS\_IN 状態の場合（バルクイン転送）、次に示す 3 点の処理を行います。

1. ファンクションからホストへ向けデータ送信処理。
2. ホストが予定したデータ長に対しファンクションが送信するデータ長が短い場合の 0 付加処理。
3. CSW で送信する情報の作成。

サンプルプログラムのデータトランSPORT（バルクイン転送）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。

このサンプルソフトでは、ホストが要求するデータ長に対しファンクションが送信するデータ長が短い場合、USB Mass Storage Class の Bulk-Only Transport に記載されている通り、ファンクションが送信するデータの後に 0 を付加しホストが要求する長さのデータを送信後、ステータストランSPORTにおいて 0 を何バイト付加したかを報告しています。

この動作を行うために、CBW データの dCBWDataTransferLength、CSW データの dCSWDataResidue、CSW データの bCSWStatus をグローバル変数として使用しています。

## 5. サンプルプログラムの動作

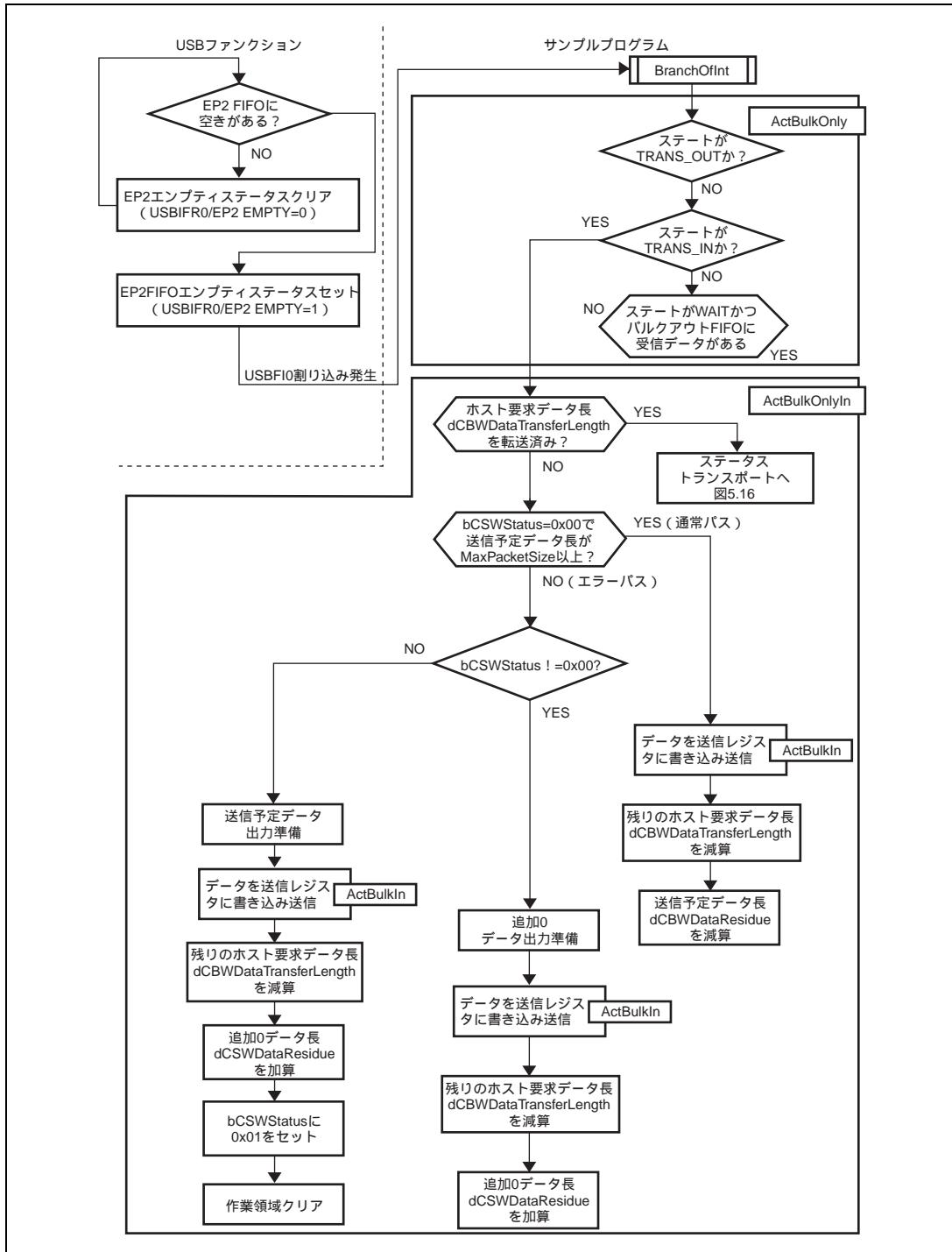


図 5.14 データトランSPORT (バルクイン転送)

サンプルプログラムのデータトランSPORT（バルクアウト転送）における動作を図 5.15 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。

ファームウエアのステートが TRANS\_OUT 状態の場合（バルクアウト転送）、次に示す 3 点の処理を行います。

1. ホストからファンクションに対するデータ受信処理。
2. データ長演算処理
3. CSWで送信する情報の作成。

このサンプルソフトでは、ホストが予定したデータ長に対しファンクションが受信したデータ長が短い場合、USB Mass Storage Class の Bulk-Only Transport に記載されている通り、データトランSPORTにおいてファンクションが受信する不足データ長をステータストランSPORTにおいて報告しています。

この動作を行うために、CBW データの dCBWDataTransferLength と、CSW データの dCSWDataResidue をグローバル変数として使用しています。

## 5. サンプルプログラムの動作

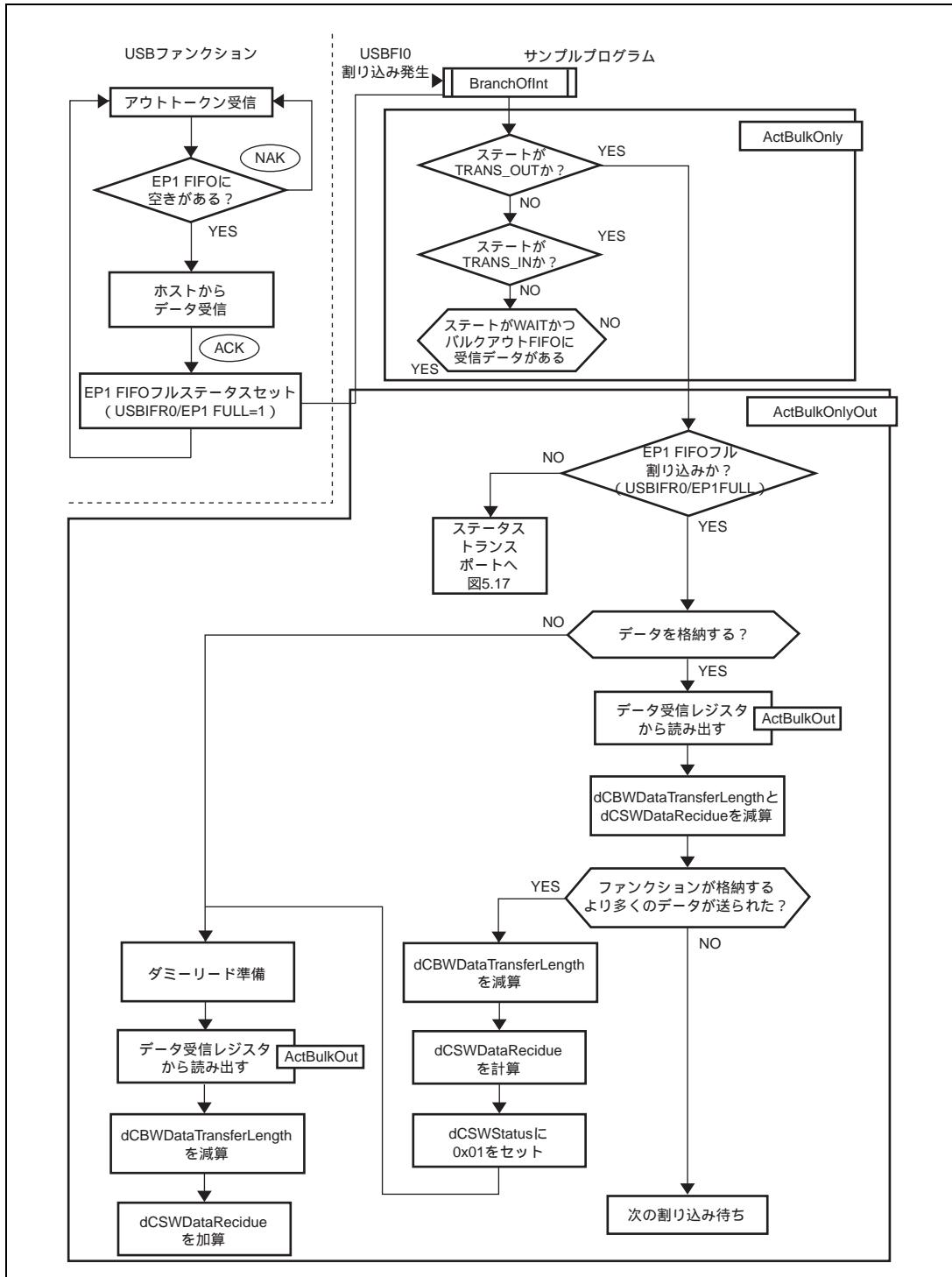


図 5.15 データトランsport (バルクアウト転送)

### 5.5.3 ステータストラנסポート

ステータストラنسポートでは、ファンクションからホストに対しデータの送信を行います。

このときファームウエアのステートは TRANS\_IN または TRANS\_OUT どちらかの状態で実行されます。ファームウエアのステートが TRANS\_IN 状態の場合（バルケイン転送）、次に示す 4 点の処理を行います。

1. EP2エンブティステータス割り込みの禁止。
2. CSWデータの転送準備。
3. CSWデータ発行。
4. ファームウエアのステートをWAITに設定。

図 5.16 にサンプルプログラムのステータストラنسポート（データトランスポートバルケイン転送）における動作を示します。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。

## 5. サンプルプログラムの動作

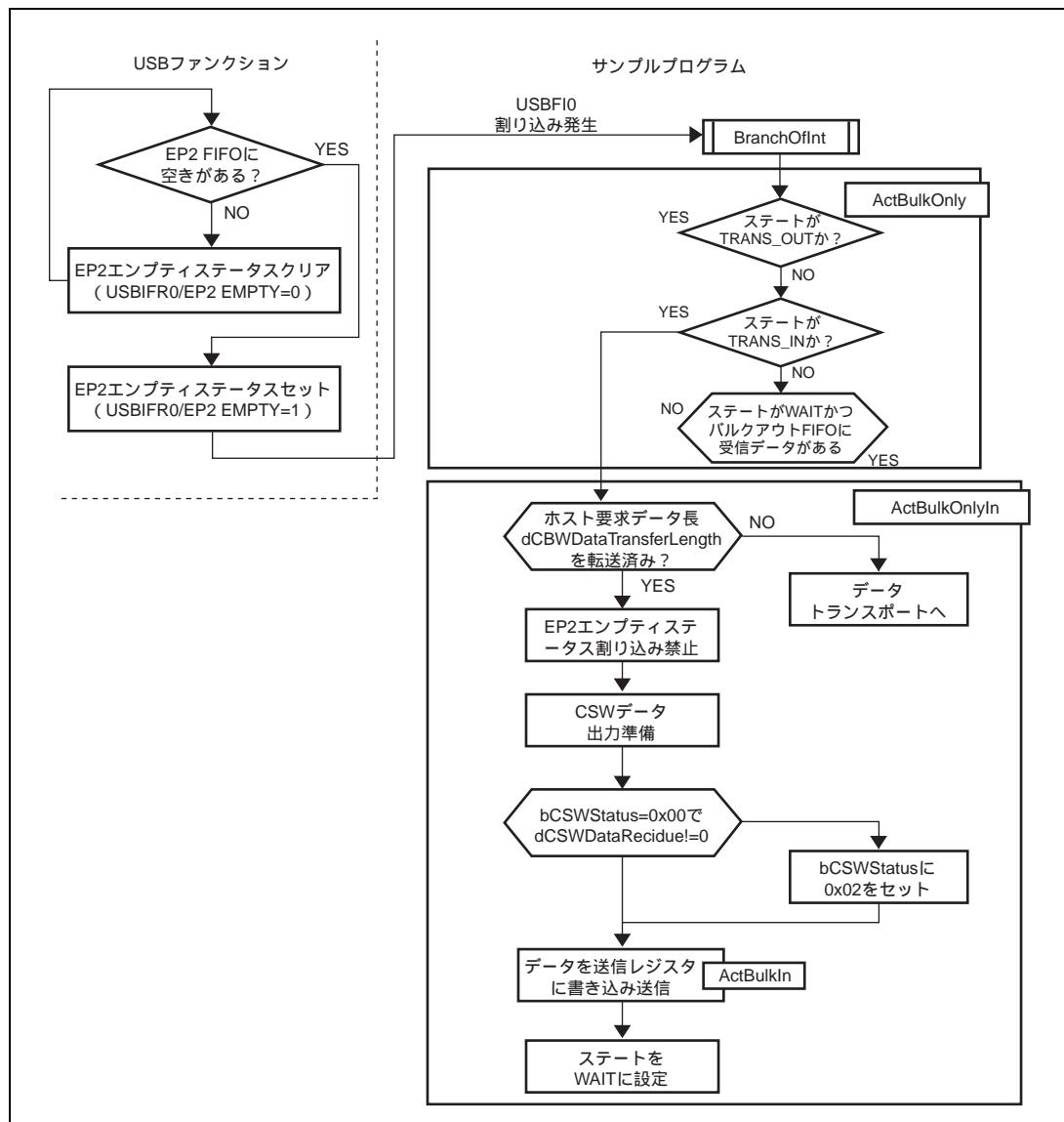


図 5.16 ステータストランsport (データトランsportバルクイン転送)

サンプルプログラムのステータストラ nsポート（データトランSPORTバルクアウト転送）における動作を図 5.17 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。

ファームウエアのステートが TRANS\_OUT 状態の場合（バルクアウト転送）、次に示す 4 点の処理を行います。

1. CSWデータの転送準備。
2. 受信不足データチェック。
3. CSWデータ発行。
4. ファームウエアのステートをWAITに設定。

このサンプルソフトでは、ホストが予定したデータ長に対しファンクションが受信したデータ長が短い場合、USB Mass Storage Class の Bulk-Only Transport に記載されている通り、データトランSPORTにおいてファンクションが受信する不足データ長をステータストラ nsポートにおいて報告する動作を行うために、ファンクションが受信する不足データ長をチェックし不足発生時は CSW データの bCSWStatus の値を 0x02（フェーズエラー）に設定します。

## 5. サンプルプログラムの動作

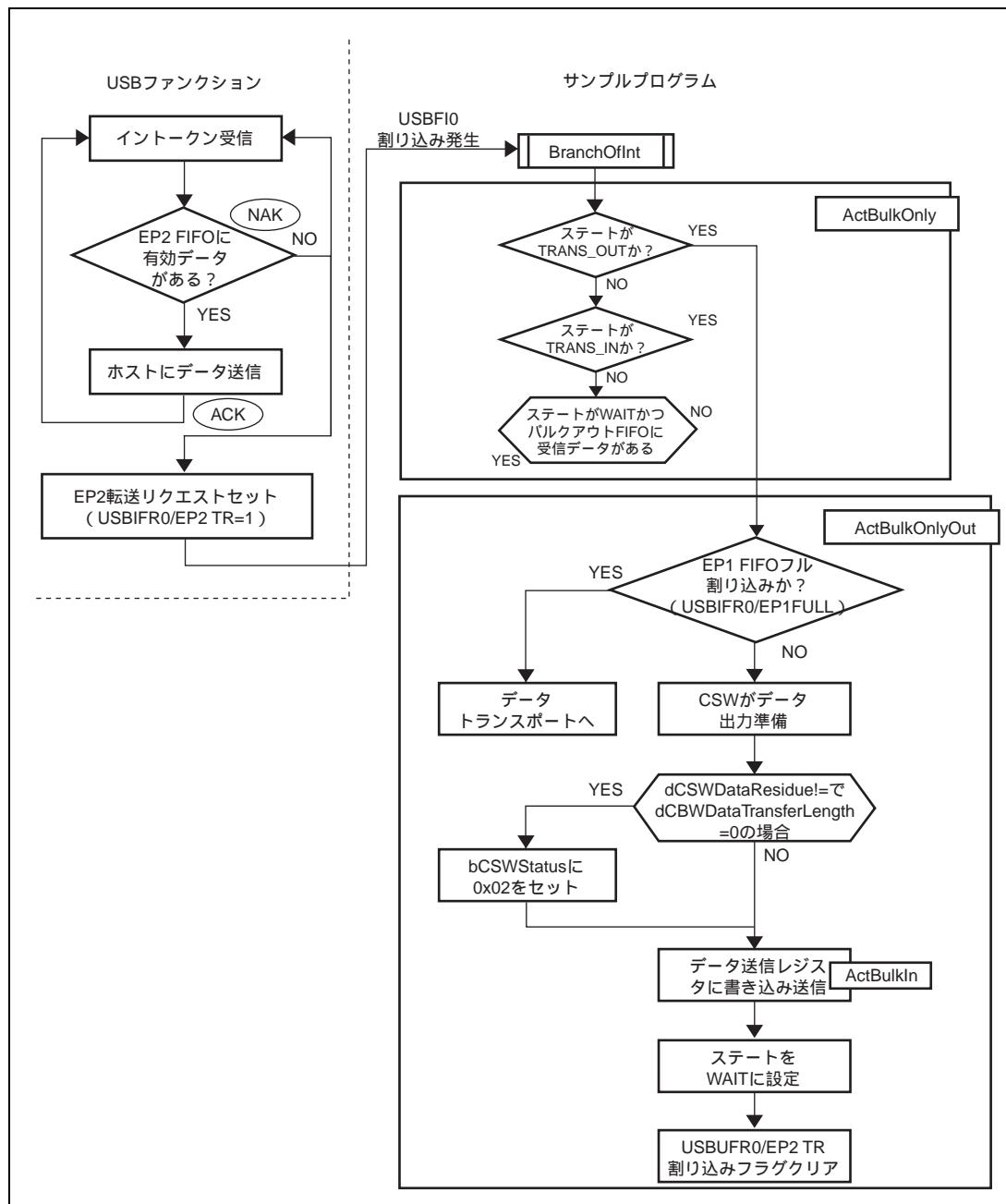


図 5.17 ステータストラנסポート（データトランスポートバルクアウト転送）

## 6. アナライザのデータ

この章では、SH7727 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 Inspector」（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説明します。なお、パケットの詳細につきましては 2.3 章をご参照ください。

【注】各パケットの前部にある「Packet#」は測定時のパケット通し番号です。

- INQUIRYコマンド

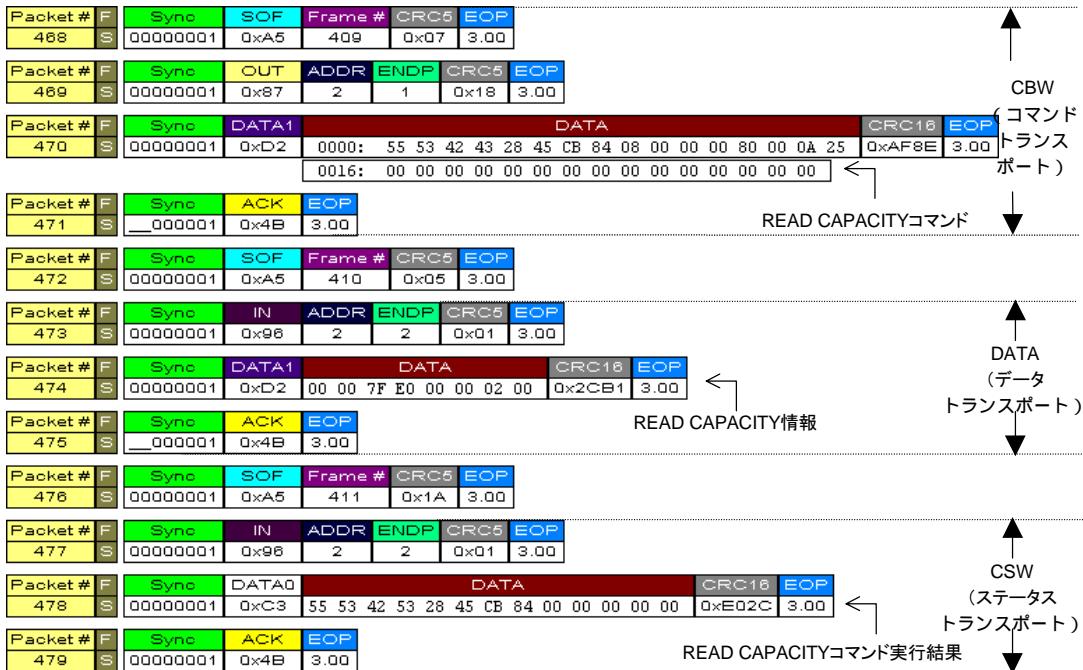
Packet #	F	Sync	SOF	Frame #	CRC5	EOP
349	S	00000001	0xA5	380	0x02	3.00
Packet #	F	Sync	OUT	ADDR	ENDP	CRC5 EOP
350	S	00000001	0x87	2	1	0x18 3.00
Packet #	F	Sync	DATA0	DATA		CRC16 EOP
351	S	00000001	0xC3	0000: 55 53 42 43 28 45 CB 84 24 00 00 00 80 00 06 12	0x9B16 3.00	CBW
				0016: 00 00 00 24 00 00 00 00 00 00 00 00 00 00 00 00		← (コマンドトランスポート)
Packet #	F	Sync	ACK	EOP		
352	S	00000001	0x4B	3.00		
Packet #	F	Sync	SOF	Frame #	CRC5	EOP
353	S	00000001	0xA5	381	0x1D	3.00
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP
354	S	00000001	0x96	2	2	0x01 3.00
Packet #	F	Sync	DATA0	DATA		CRC16 EOP
355	S	00000001	0xC3	0000: 00 80 02 02 5B 00 00 00 00 48 49 54 41 43 48 49 20	0x803C 3.00	DATA
				0016: 45 58 20 52 41 4D 20 44 69 73 6B 20 20 20 20 20		← (データトランスポート)
				0032: 31 2E 31 31		
Packet #	F	Sync	ACK	EOP		
356	S	00000001	0x4B	3.00		
Packet #	F	Sync	SOF	Frame #	CRC5	EOP
357	S	00000001	0xA5	382	0x1F	3.00
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP
358	S	00000001	0x96	2	2	0x01 3.00
Packet #	F	Sync	DATA1	DATA		CRC16 EOP
359	S	00000001	0xD2	55 53 42 53 28 45 CB 84 00 00 00 00 00	0xE02C 3.00	CSW (ステータス トランスポート)
Packet #	F	Sync	ACK	EOP		
360	S	00000001	0x4B	3.00		INQUIRYコマンド実行結果

## 6. アナライザのデータ

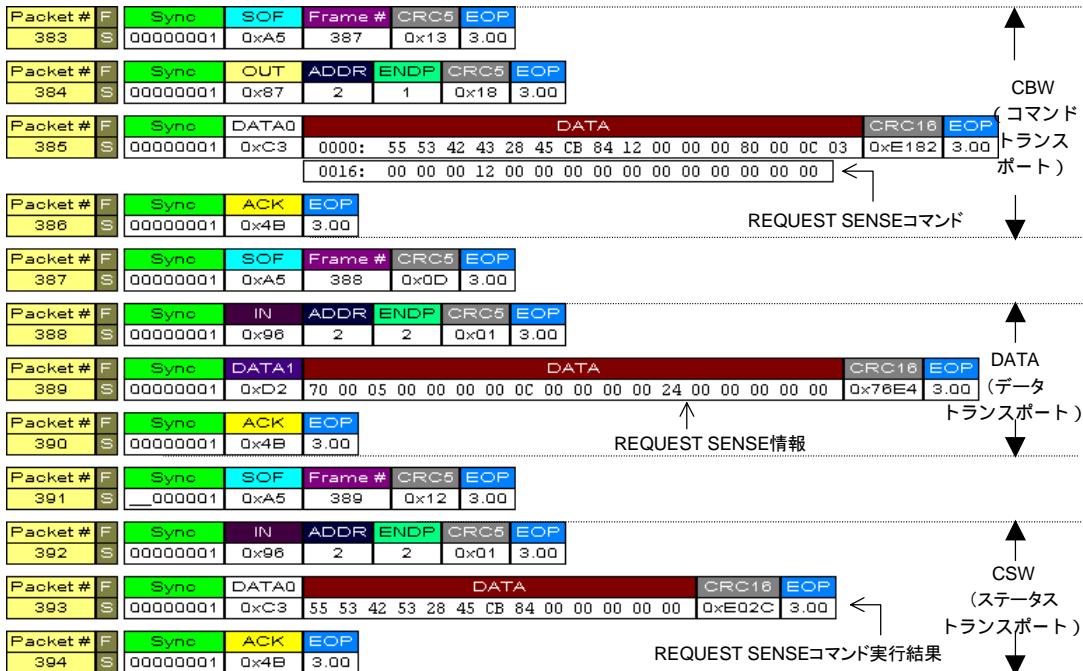
- READ FORMAT CAPACITIESコマンド

Packet #	F	Sync	SOF	Frame #	CRC5	EOP		
362	S	00000001	0xA5	384	0x11	3.00		
Packet #	F	Sync	OUT	ADDR	ENDP	CRC5	EOP	
363	S	00000001	0x87	2	1	0x18	3.00	
Packet #	F	Sync	DATA1		DATA		CRC16	EOP
364	S	00000001	0xD2	0000:	55 53 42 43 28 45 CB 84 FC 00 00 00 80 00 0A 23		0x028E	3.00 (コマンド)
				0016:	00 00 00 00 00 00 00 FC 00 00 00 00 00 00 00 00 00			CBW トランスポート
Packet #	F	Sync	ACK	EOP				
365	S	00000001	0x4B	3.00				
Packet #	F	Sync	SOF	Frame #	CRC5	EOP		
366	S	00000001	0xA5	385	0x0E	3.00		
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
367	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA0		DATA		CRC16	EOP
368	S	00000001	0xC3	0000:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xFD0B	3.00
				0016:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0032:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0048:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
Packet #	F	Sync	ACK	EOP				
369	S	00000001	0x4B	3.00				
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
370	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA1		DATA		CRC16	EOP
371	S	00000001	0xD2	0000:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xFD0B	3.00
				0016:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0032:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0048:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
Packet #	F	Sync	ACK	EOP				
372	S	00000001	0x4B	3.00				
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
373	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA0		DATA		CRC16	EOP
374	S	00000001	0xC3	0000:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xFD0B	3.00
				0016:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0032:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0048:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
Packet #	F	Sync	ACK	EOP				
375	S	00000001	0x4B	3.00				
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
376	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA1		DATA		CRC16	EOP
377	S	00000001	0xD2	0000:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0x0024	3.00
				0016:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0032:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0048:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
Packet #	F	Sync	ACK	EOP				
378	S	00000001	0x4B	3.00				
Packet #	F	Sync	SOF	Frame #	CRC5	EOP		
379	S	00000001	0xA5	386	0x0C	3.00		
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
380	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA0		DATA		CRC16	EOP
381	S	00000001	0xC3	55 53 42 53 28 45 CB 84 FB 00 00 00 01			0xC404	3.00
Packet #	F	Sync	ACK	EOP				
382	S	00000001	0x4B	3.00				

- READ CAPACITYコマンド



- EQUEST SENSEコマンド



## 6. アナライザのデータ

- READ (10) コマンド

Packet #	F	Sync	SOF	Frame #	CRC5	EOP
478	S	00000001	0xA5	412	0x04	3.00
479	S	00000001	0xB7	2	1	0x18 3.00
480	S	00000001	0xC3	DATA00 0000: 55 53 42 43 68 (S 0D) A1 00 02 00 00 80 00 0A 28 0016: 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00	CRC16 0x123F	EOP 3.00 CBW (コマンド トランスポート)
481	S	00000001	0x4B	3.00		READ(10)コマンド
482	S	00000001	0xA5	413	0x1B	3.00
483	S	00000001	0x96	IN ADDR ENDP CRC5 EOP 2 2 0x01 3.00		
484	S	00000001	0xD2	DATA 0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	CRC16 0xFDOB	EOP 3.00
485	S	00000001	0x4B	3.00		READ(10)情報
486	S	00000001	0x96	IN ADDR ENDP CRC5 EOP 2 2 0x01 3.00		
487	S	00000001	0xC3	DATA 0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	CRC16 0xFDOB	EOP 3.00
488	S	00000001	0x4B	3.00		DATA (データ トランスポート)

Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP
489	S	00000001	0x96	2	2	0x01	3.00
490	S	00000001	0xD2	DATA		CRC16	EOP
				0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xFD0B	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
491	S	00000001	0x4B	3.00	Sync	ACK	EOP
492	S	00000001	0x96	2	2	0x01	3.00
493	S	00000001	0xC3	DATA		CRC16	EOP
				0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xFD0B	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
494	S	00000001	0x4B	3.00	Sync	ACK	EOP
495	S	00000001	0x96	2	2	0x01	3.00
496	S	00000001	0xD2	DATA		CRC16	EOP
				0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xFD0B	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
				0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
497	S	00000001	0x4B	3.00	Sync	ACK	EOP

DATA  
(データ  
トランスポート)

## 6. アナライザのデータ

Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
498	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA0		DATA		CRC16 EOP	
499	S	00000001	0xC3		0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xFD0B	3.00
Packet #	F	Sync	ACK	EOP				
500	S	00000001	0x4B	3.00				
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
501	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA1		DATA		CRC16 EOP	
502	S	00000001	0xD2		0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0x7B08	3.00
Packet #	F	Sync	ACK	EOP				
503	S	00000001	0x4B	3.00				
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
504	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA0		DATA		CRC16 EOP	
505	S	00000001	0xC3		0000: 00 00 04 00 00 20 00 00 E0 7F 00 00 00 00 0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		0xED33	3.00
Packet #	F	Sync	ACK	EOP				
506	S	00000001	0x4B	3.00				
Packet #	F	Sync	SOF	Frame #	CRC5	EOP		
507	S	00000001	0xA5	414	0x19	3.00		
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	
508	S	00000001	0x96	2	2	0x01	3.00	
Packet #	F	Sync	DATA1		DATA		CRC16 EOP	
509	S	00000001	0xD2	55 53 42 53 68 C5 D1 81 00 00 00 00 00 00 00		0x1973	3.00	
Packet #	F	Sync	ACK	EOP				
510	S	00000001	0x4B	3.00				

READ(10)コマンド実行結果

CSW  
(ステータス  
トランスポート)

- TEST UNIT READYコマンド

Packet #	F	Sync	SOF	Zoom In	#	CRC5	EOP
7021	S	00000001	0xA5	953	0x1C	3.00	
Packet #	F	Sync	OUT	ADDR	ENDP	CRC5	EOP
7022	S	00000001	0x87	2	1	0x18	3.00
Packet #	F	Sync	DATA0	DATA			CRC16 EOP
7023	S	00000001	0xC3	0000:	55 53 42 43 08 E0 0D 81 00 00 00 00 00 00 00 06 00	0xF7AE	3.00
				0016:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
Packet #	F	Sync	ACK	EOP			
7024	S	00000001	0x4B	3.00			
Packet #	F	Sync	SOF	Frame #	CRC5	EOP	
7025	S	00000001	0xA5	954	0x1E	3.00	
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP
7026	S	00000001	0x98	2	2	0x01	3.00
Packet #	F	Sync	NAK	EOP			
7027	S	00000001	0x5A	3.00			
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP
7028	S	00000001	0x98	2	2	0x01	3.00
Packet #	F	Sync	NAK	EOP			
7029	S	00000001	0x5A	3.00			
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP
7030	S	00000001	0x98	2	2	0x01	3.00
Packet #	F	Sync	DATA1	DATA			CRC16 EOP
7031	S	00000001	0xD2	55 53 42 53 08 E0 0D 81 00 00 00 00 00 00 00 00 00	0x4FD7	3.00	
Packet #	F	Sync	ACK	EOP			
7032	S	00000001	0x4B	3.00			

- VERIFY (10) コマンド

Packet #	F	Sync	SOF	Frame #	CRC5	EOP		
9487	S	00000001	0xA5	1913	0x0E	3.00		
Packet #	F	Sync	OUT	ADDR	ENDP	CRC5 EOP		
9488	S	00000001	0x87	2	1	0x18 3.00		
Packet #	F	Sync	DATA0	DATA			CRC16	EOP
9489	S	00000001	0xC3	0000: 55 53 42 43 68 19 22 80 00 00 00 00 00 00 0A 2F	0x92D1	3.00	CBW (コマンド)	
				0016: 00 00 00 06 20 00 02 00 00 00 00 00 00 00 00 00 00 00			← トランスポート	
Packet #	F	Sync	ACK	EOP			VERIFY(10)コマンド	
9490	S	00000001	0x4B	3.00			↓	
Packet #	F	Sync	SOF	Frame #	CRC5	EOP		
9491	S	00000001	0xA5	1914	0x0C	3.00		
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP		
9492	S	00000001	0x96	2	2	0x01 3.00		
Packet #	F	Sync	NAK	EOP				
9493	S	00000001	0x5A	3.00				
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP		
9494	S	00000001	0x96	2	2	0x01 3.00		
Packet #	F	Sync	NAK	EOP				
9495	S	00000001	0x5A	3.00				
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP		
9496	S	00000001	0x96	2	2	0x01 3.00		
Packet #	F	Sync	DATA1	DATA			CRC16	EOP
9497	S	00000001	0xD2	55 53 42 53 68 19 22 80 00 00 00 00 00 00 00 00	0x522B	3.00	← トランスポート	
Packet #	F	Sync	ACK	EOP			VERIFY(10)実行結果	
9498	S	00000001	0x4B	3.00			↓	

## 6. アナライザのデータ

- WRITE (10) コマンド

Packet #	F	Sync	SOF	Frame #	CRC5	EOP
26886	S	00000001	0xA5	417	0x1F	3.00
26887	S	00000001	0x87	2	1	0x18 3.00
26888	S	00000001	0xC3	DATA	CRC16	EOP
				0000: 55 53 42 43 C8 CC OB 81 00 02 00 00 00 00 0A 2A	0xF42C	3.00 (コマンド)
				0016: 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00		← CBW
26889	S	00000001	0x4B	3.00		
26890	S	00000001	0xA5	418	0x1D	3.00
26891	S	00000001	0x87	2	1	0x18 3.00
26892	S	00000001	0xD2	DATA	CRC16	EOP
				0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0xFD0B	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		← WRITE(10)コマンド
				0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
				0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		↓ トランスポート
26893	S	00000001	0x4B	3.00		
26894	S	00000001	0x87	2	1	0x18 3.00
26895	S	00000001	0xC3	DATA	CRC16	EOP
				0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0xFD0B	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		← WRITE(10)情報
				0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
				0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
26896	S	00000001	0x4B	3.00		
26897	S	00000001	0x87	2	1	0x18 3.00
26898	S	00000001	0xA5	419	0x1D	3.00
26899	S	00000001	0x87	2	1	0x18 3.00
26900	S	00000001	0xC3	DATA	CRC16	EOP
				0000: 00 00 04 00 00 00 20 00 00 00 E0 7F 00 00 00 00	0xED33	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		↑ DATA (データ)
				0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
				0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		↑ トランスポート
26901	S	00000001	0x87	2	1	0x18 3.00
26902	S	00000001	0xD2	DATA	CRC16	EOP
				0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0x7B08	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		↑ DATA (データ)
				0032: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
				0048: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
26903	S	00000001	0x4B	3.00		
26904	S	00000001	0xA5	420	0x1D	3.00
26905	S	00000001	0x87	2	1	0x18 3.00
26906	S	00000001	0xC3	DATA	CRC16	EOP
				0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0x55AA	3.00
				0016: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		↓ トランスポート
26907	S	00000001	0x4B	3.00		

Packet #	F	Sync	SOF	Frame #	CRC5	EOP
26915	S	00000001	0xA5	419	0x02	3.00
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP
26916	S	00000001	0x96	2	2	0x01 3.00
Packet #	F	Sync	NAK	EOP		
26917	S	00000001	0x5A			3.00
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP
26918	S	00000001	0x96	2	2	0x01 3.00
Packet #	F	Sync	NAK	EOP		
26919	S	00000001	0x5A			3.00
Packet #	F	Sync	IN	ADDR	ENDP	CRC5 EOP
26920	S	00000001	0x96	2	2	0x01 3.00
Packet #	F	Sync	DATA0	DATA	CRC16	EOP
26921	S	00000001	0xC3	55 53 42 53 C8 CC 0B 81 00 00 00 00 00 00 0x63C5	3.00	
Packet #	F	Sync	ACK	EOP		
26922	S	00000001	0x4B			3.00

↑ CSW  
(ステータス)  
トランスポート

↓ WRITE(10)コマンド実行結果

## 6. アナライザのデータ

---

SH7727 USBファンクションモジュール  
Mass Storage Class ( Bulk-Only Transport )  
アプリケーションノート

発行年月 2002年 3月 第1版  
発 行 株式会社 日立製作所  
半導体グループビジネス企画本部  
編 集 株式会社 日立小平セミコン  
技術ドキュメントグループ

©株式会社 日立製作所 2002

# SH7727 USB ファンクションモジュール Mass Storage Class (Bulk-Only Transport) 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

ADJ-502-088