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い申し上げます。

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、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、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。

SH7285/SH7286　USB ファンクションモジュール

USB シリアル変換アプリケーションノート

要旨

本資料は、SH7285/SH7286 の USB ファンクションモジュールの応用例として、USB シリアル変換システムの実現方法を掲載しています。本資料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応用例を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。

動作確認デバイス

SH7285，SH7286

目次

1.　はじめに

1.1　仕様

本資料は、SH7285/SH7286 の USB ファンクションモジュールの使用方法、および USB ファンクションモジュールの応用例として、USB シリアル変換システムの実現方法を掲載しています。

1.2　使用機能

- USB ファンクションモジュール
- FIFO 付シリアルコミュニケーションインタフェース

1.3　適用条件

- マイコン　：SH7285 / SH7286
- 動作周波数　：内部クロック　100MHz
  バスクロック　50MHz
  周辺クロック　50MHz
- C コンパイラ　：ルネサス テクノロジ製
  SuperH RISC engine ファミリ C/C++ コンパイラパッケージ Ver.9.01 Release01
- コンパイルオプション　：-cpu=sh2a -include="$(WORKSPDIR)¥inc"
  -object="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -gbr=auto
  -chgincpath-errorpath -global_volatile=0 -opt_range=all
  -infinite_loop=0-del_vacant_loop=0 -struct_alloc=1 -nologo

1.4　関連アプリケーションノート

本資料の参考プログラムは、SH7280初期設定アプリケーションノートの設定条件にて動作を確認しています。そちらも合わせてご参照ください。

2.　応用例の説明

本プログラム例では USB ファンクションモジュール (USB) を使用したコントロールイン転送、コントロールアウト転送、バルクイン転送およびバルクアウト転送を行います。また、USB 通信とシリアル通信の変換を行います。

2.1　使用機能の動作概要

USB ファンクションモジュールは、USB1.1 に準拠した UDC（USB Device Controller）を内蔵しており、USB プロトコルを自動処理します。また、各転送タイプに対応したエンドポイントを 4 つ実装しており、USB ホストとのコントロール転送、バルクアウト転送、バルクイン転送およびインタラプト転送を容易に実現することが可能です。

SH7285/SH7286 の 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。）
- 転送スピード：フルスピード
- 割り込み要求：USB 送受信に必要な各種割り込み信号を生成
- クロック：　外部入力（48MHz）
  内部入力（EXTAL 12MHz 選択時のみ）
- 低消費電力モード
  USB ケーブル切断時、UDC 内部クロック停止による低消費電力化が可能
- エンドポイント構成

表 2.1　エンドポイント構成

エンドポイント名	名称	転送タイプ	最大パケットサイズ	FIFO バッファ容量	DMA 転送
エンドポイント 0	EP0s	セットアップ	8 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	インタラプト	8 Byte	8 Byte	—

2.2　USB ファンクションモジュールを使用した USB 通信

USBファンクションモジュールを使用したUSB通信の例として、参考プログラムは表 2.2に示すUSB通信機能を実現しています。

表 2.2　USB 通信機能

USB 通信機能	説明
USB ホストへの接続検出処理	ポートによる D+端子プルアップ制御により接続検出を行います。
コントロール転送処理	コントロール転送で USB ホストから送信される USB コマンドに対し、コマンドデコード、データステージおよびステータスステージ処理を実行します。
バルクイン／アウト転送処理	バルクイン／アウト転送処理を実行します。

2.3　USB ホストへの接続検出処理

USB ホストへの接続検出処理は、USB デバイスのケーブルを USB ホストに接続した際に発生するケーブル接続割り込み（USBIFR0/BRST）を使用して行われます。

参考プログラムは、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、USBホストはUSBデバイスが接続されたことを認識します。図 2.1にUSBホストへの接続検出時の参考プログラムの動作フローを、図 2.2にUSB周辺ブロック回路例を示します。

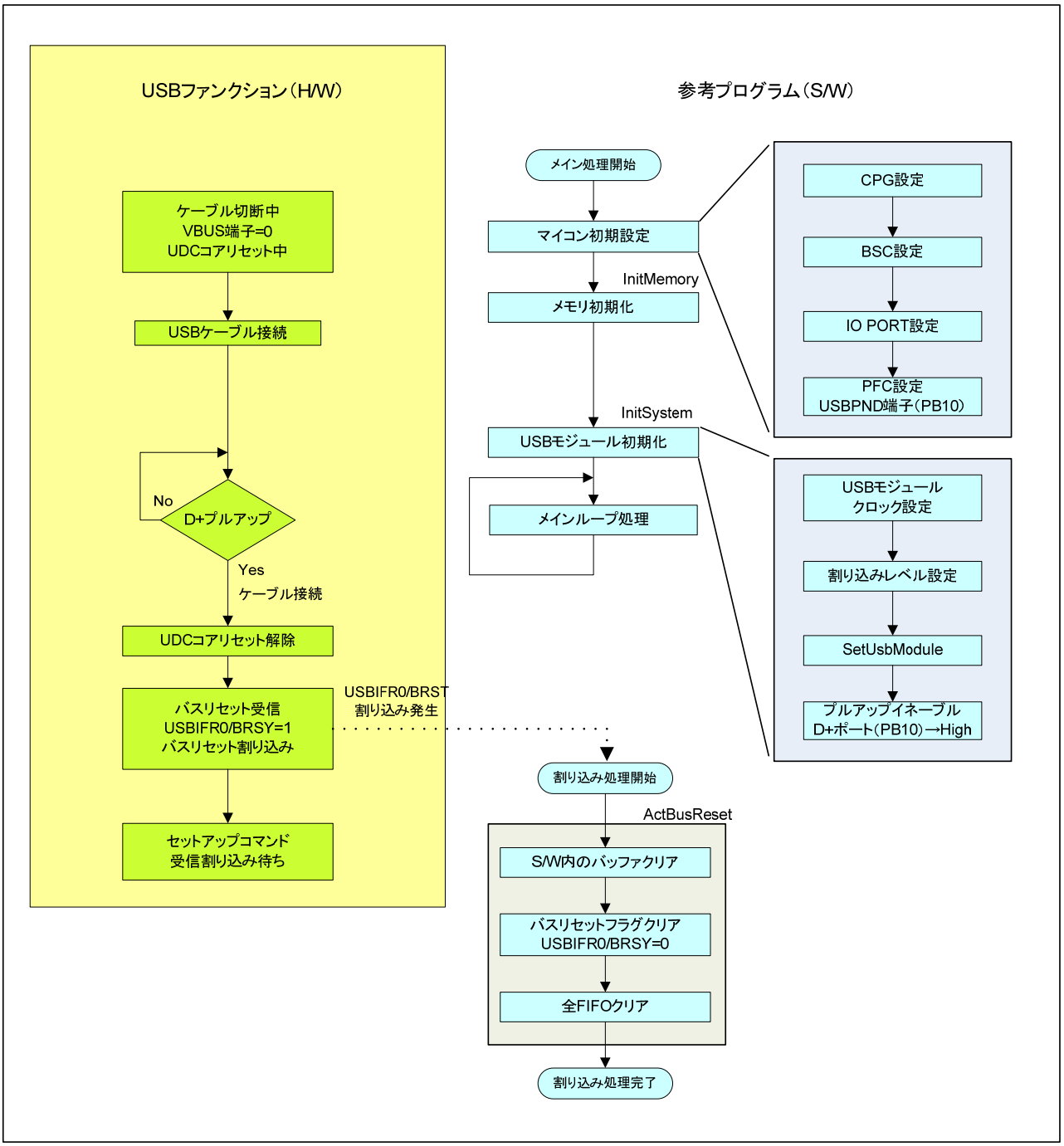

図 2.1　USB ホストへの接続検出フロー

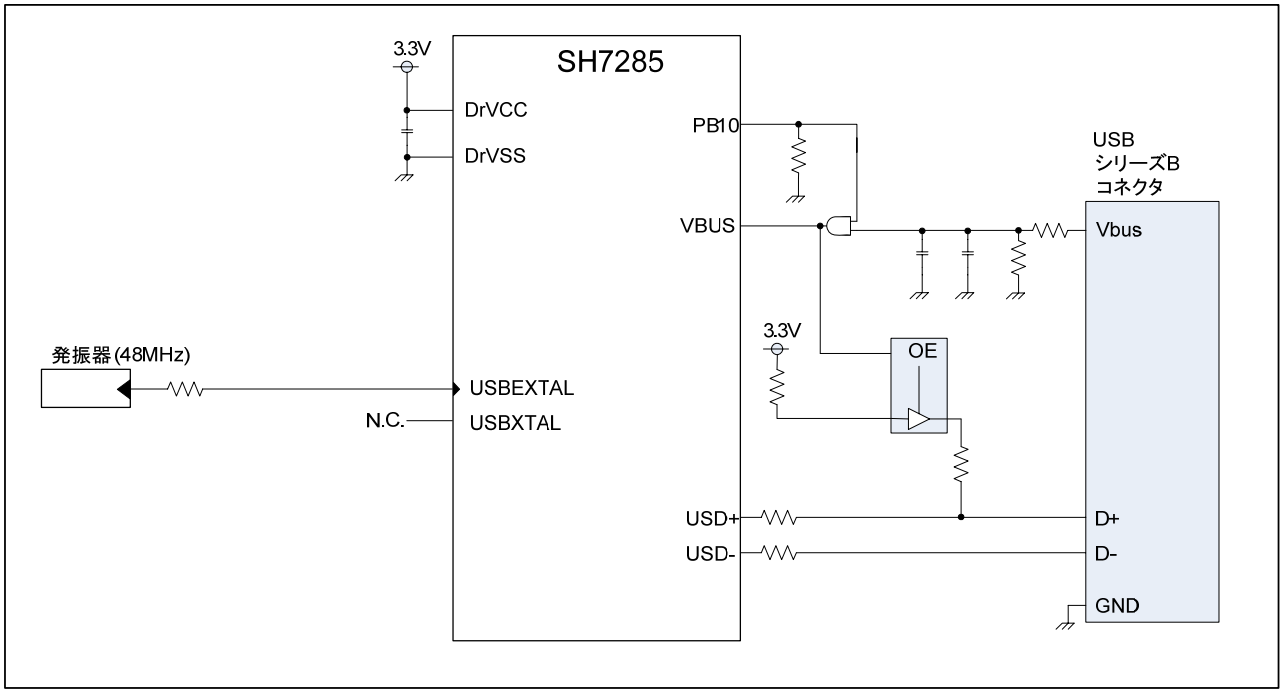

図 2.2　USB モジュール周辺ブロック回路例

2.4　コントロール転送処理

コントロール転送は、エンドポイント 0 のデフォルトパイプを使用した USB 転送処理であり、すべての USB デバイスでサポートする必要があります。USB ホストはコントロール転送により、USB 標準コマンドを USB デバイスに発行し、USB デバイスをコンフィギュレーションします。また、コントロール転送は、クラスまたはベンダ固有のコマンドを発行する場合にも使用されます。

コントロール転送は、セットアップ、データ（ない場合もあります。）、ステータスの 3 つのステージで構成され、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。また、コントロール転送は、データステージにおけるデータの方向によって、2 つに分ける事ができます。データステージにおいて、USB ホストから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”となります。データステージは、USBホストがデータの方向を反転したトークンを送信することにより完了し、この完了を示すトークンを送信するステージがステータスステージになります。図 2.3にデータステージにおけるデータの方向を、図 2.4にコントロール転送の各ステージの構成を示します。

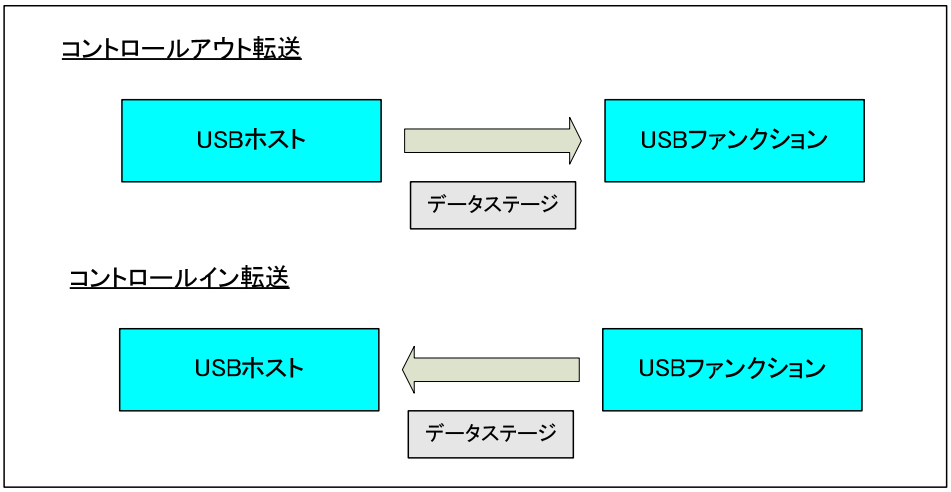

図 2.3　データステージにおけるデータの方向

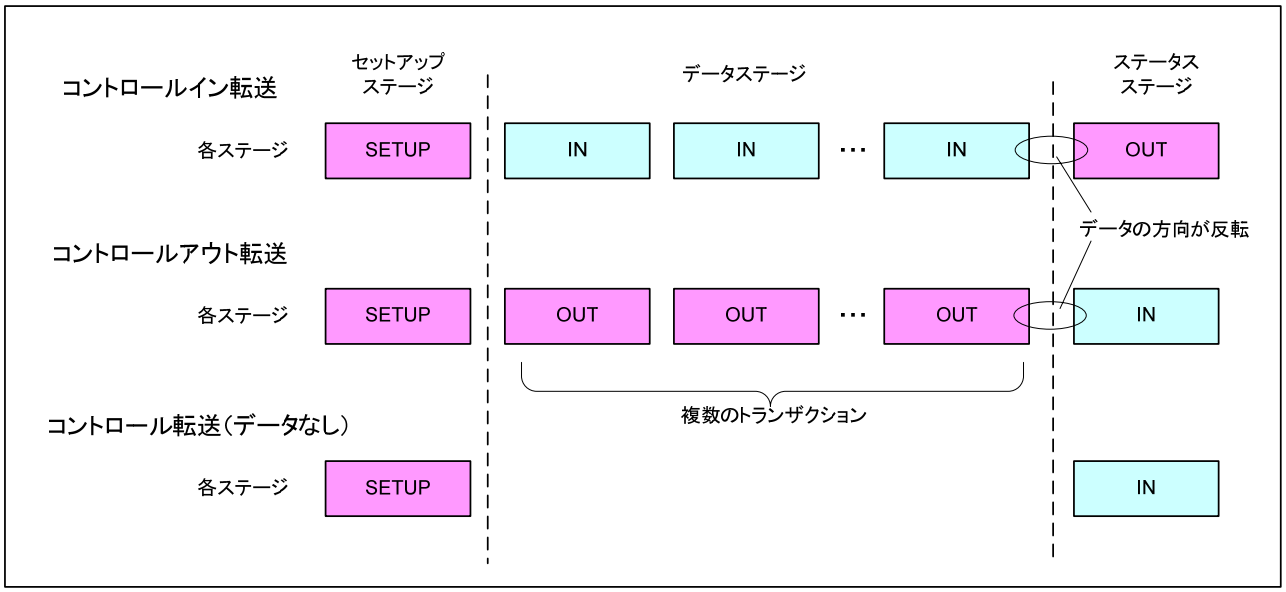

図 2.4　コントロール転送における各ステージの構成

USB ファンクションモジュールは、USB 標準コマンドに対して、ハードウェアで自動的にコマンドデコード、データステージおよびステータスステージ処理を実行しますが、一部の USB 標準コマンドおよびクラスコマンド、ベンダーコマンドは、ソフトウェアにてコマンドデコード、データステージおよびステータスステージ処理を実行する必要があります。

参考プログラムは、ソフトウェア処理が必要なUSB標準コマンドであるGet Descriptorコマンドに対し処理を実行します。また、応用例であるUSB通信とシリアル通信の変換を実現するために、USB COMクラスのコマンドに対し、処理を実行します。表 2.3にUSBコマンドと参考プログラム処理内容を示します。

対応していない USB コマンドを受信した場合、参考プログラムは、ソフトウェアにて STALL 応答処理を実行します。

表 2.3　USB コマンドと参考プログラム処理

USB コマンド	分類	参考プログラム処理
Clear Feature Get Configuration Get Interface Get Status Set Address Set Configuration Set Feature Set Interface	USB 標準 コマンド	ハードウェアにて自動的にコマンドデコード、データステージおよびステータスステージ処理が実行されます。 ソフトウェアは何もしません。
Get Descriptor		ソフトウェアにてコマンドデコード、データステージおよびステータスステージ処理を実行します。
Set Line Coding Set Control Line State Send Break Get Line Coding	USB COM クラス コマンド	
上記以外の USB コマンド	−	ソフトウェアにて STALL 応答処理を実行します。

2.4.1　セットアップステージ処理

セットアップステージは、1つのセットアップトランザクションで構成されます。USBホストがセットアップトークンおよびデータ（USBコマンド）を送信し、USBデバイスが受信したデータ（USBコマンド）に対して、ハンドシェイク応答します。図 2.5にセットアップトランザクションのフローを示します。

USBファンクションモジュールは、USB標準コマンド（一部を除く）に対し、ハードウェアが自動的にセットアップステージ、データステージおよびステータスステージを実行しますが、USB標準コマンド以外のUSBコマンドの場合、受信したUSBコマンドをEP0Sのデータレジスタ（USBEPDR0S）内に保持し、セットアップコマンド受信完了割り込み（USBIFR0/SETUPTS）を発生します。

参考プログラムは、割り込み処理内でデータレジスタ（USBEPDR0S）内に保持されたUSBコマンドを読み出し、USBコマンドをデコードし、以降のステージ処理を決定します。また、USBコマンドをデコードした結果、コントロールイン転送を実行するUSBコマンドの場合は、データステージにて、USBホストに転送する最初のデータをEP0iFIFO内に書き込んだ後に処理を完了します。図 2.6にセットアップステージにおける参考プログラムの動作を示します。また、参考プログラムはUSB COMクラスコマンド処理をDecComCommand関数にて処理します。

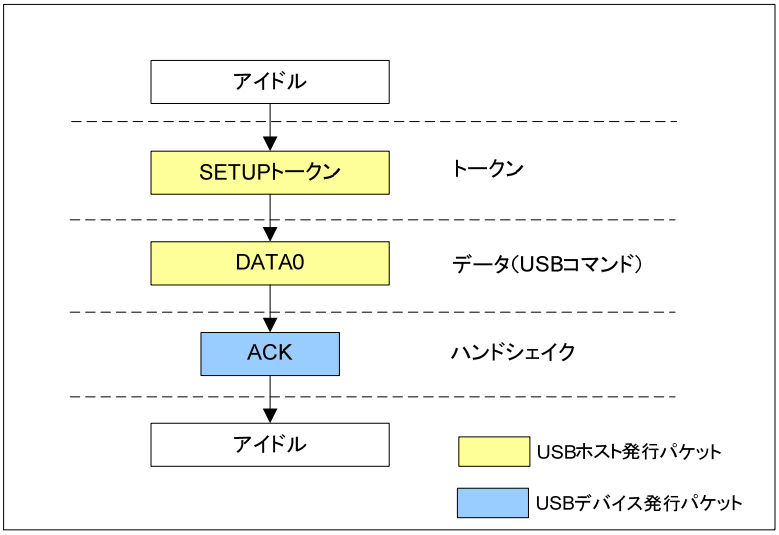

図 2.5　セットアップトランザクションのフロー

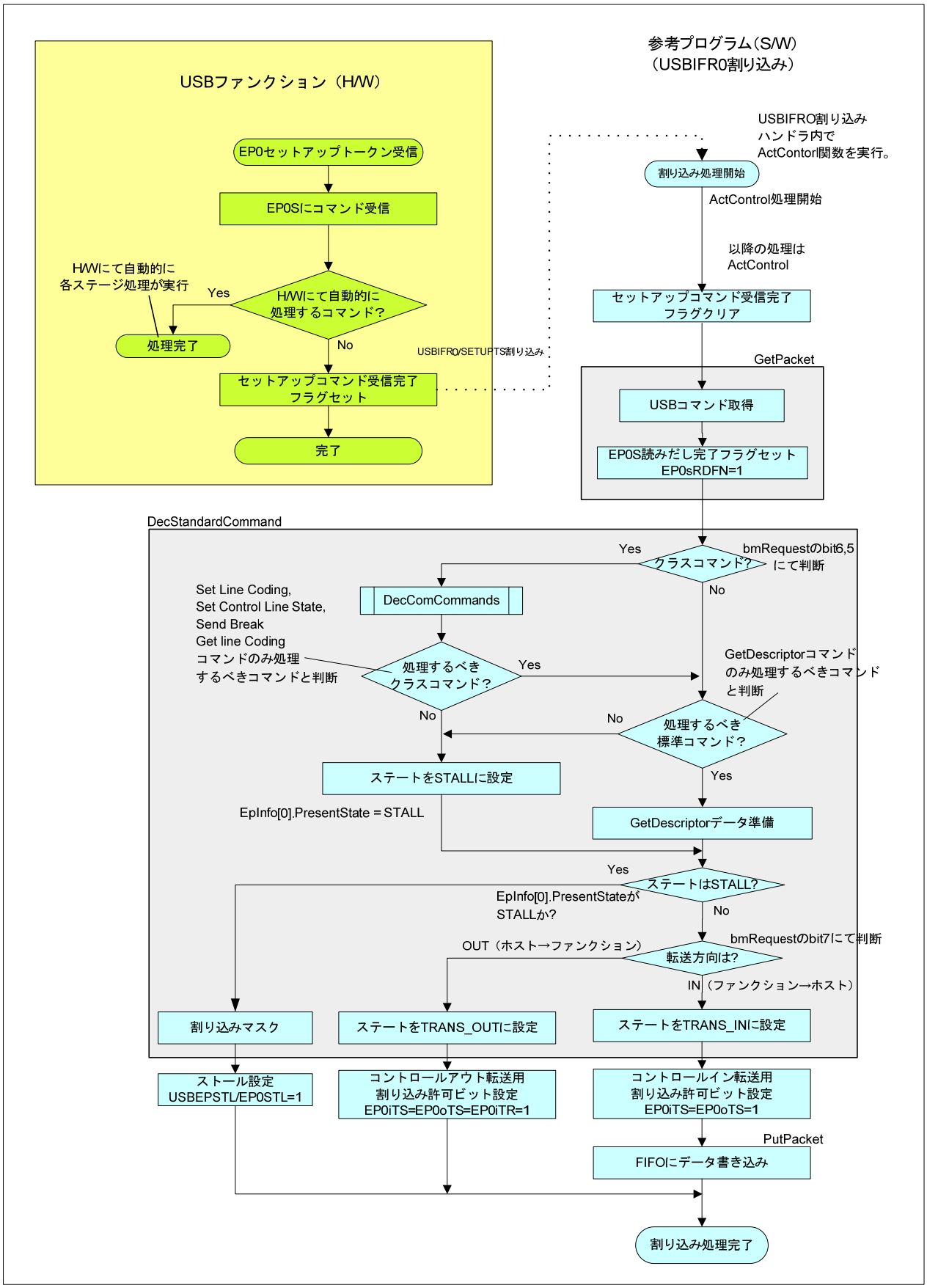

図 2.6　セットアップステージ

2.4.2　　データステージ処理

データステージは、1つまたは複数のデータトランザクションで構成されます。コントロールイン転送のデータステージはデータイントランザクションで構成され、この処理をデータインステージ処理とします。コントロールアウト転送のデータステージはデータアウトトランザクションで構成され、この処理をデータアウトステージ処理とします。

(1)データインステージ処理

USBホストがイントークンを送信します。USBデバイスはイントークンを受信するとUSBホストにデータを送信し、USBホストからのACK応答を待ちます。USBデバイスはデータを送信できない状態で、イントークンを受信した場合、USBホストにNAKを送信します（NAK応答）。図 2.7にデータイントランザクションのフローを示します。

USBファンクションモジュールは、EP0iFIFO内に有効なデータがない状態でイントークンを受信した場合、ハードウェアが自動的にUSBホストにNAKを送信します。EP0iFIFO内に有効なデータがある状態でイントークンを受信した場合は、EP0iFIFO内のデータをUSBホストに送信し、USBホストからのACK応答を待ちます。ACK応答を受信すると、データ送信完了割り込み（USBIFR0/EP0iTS）を発生します。また、イントークンではなく、データインステージ完了を示すアウトトークンを受信するとデータ受信完了割り込み（USBIFR0/EP0oTS）を発生します。

参考プログラムは、割り込み処理内で割り込みの種類を確認します。データ受信完了割り込み（USBIFR0/EP0oTS）の場合は、ステータスステージ処理に移行します。データ送信完了割り込み（USBIFR0/EP0iTS）の場合は、次にUSBホストに送信するべきデータをEP0iFIFOに書き込み、次の割り込み発生を待ちます。図 2.8に参考プログラムにおけるデータインステージ処理の動作フローを示します。

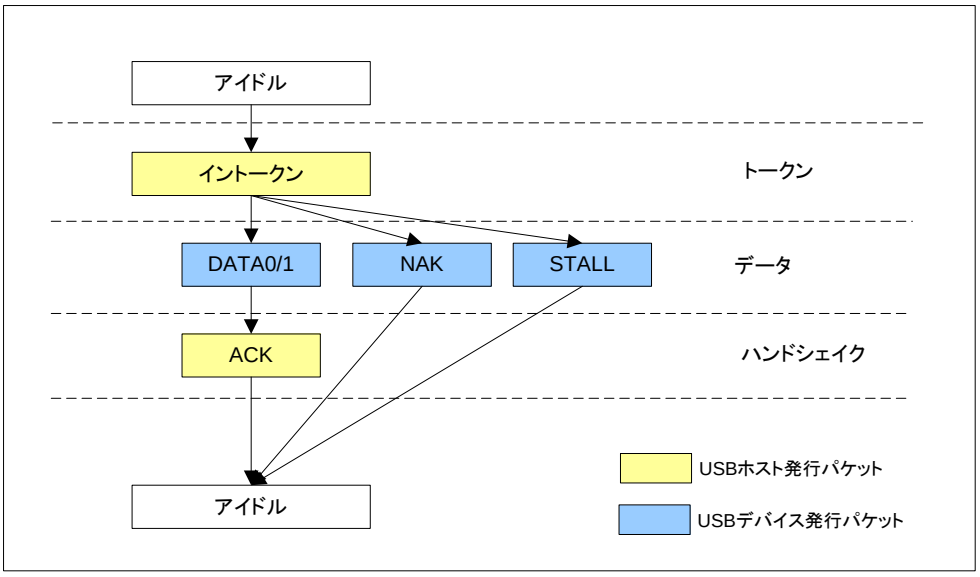

図 2.7　データイントランザクション

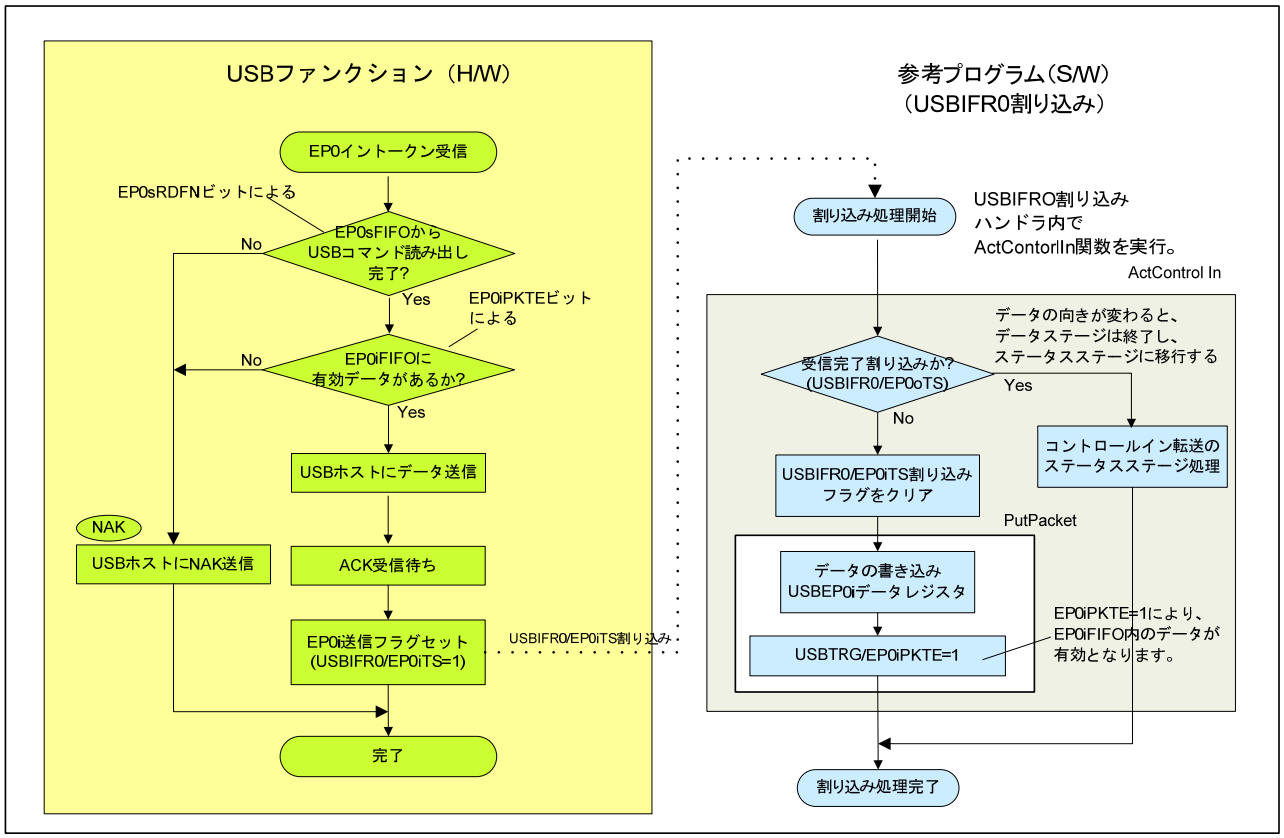

図 2.8　データインステージ（コントロールイン転送）

(2)データアウトステージ処理

USBホストがアウトトークンおよびデータを送信します。USBデバイスはアウトトークンを受信後、データを受信し、ACKを送信します（ACK応答）。USBデバイスはデータを受信できない状態で、アウトトークンを受信した場合、続けて送信されるデータを無視し、ハンドシェイクにてNAKを送信します。USBホストはハンドシェイクにてNAKを受信した場合は、再度アウトトークンおよびデータを送信します。図 2.9にデータアウトトランザクションのフローを示します。

USBファンクションモジュールは、データを受信できない状態で、アウトトークンを受信した場合、USBホストから続けて送信されるデータをハードウェアが自動的に破棄し、ハンドシェイクにてUSBホストにNAKを送信します。データを受信できる状態で、アウトトークンを受信した場合は、USBホストから送信されるデータをEP0oFIFO内に保持し、ハンドシェイクにてUSBホストにACKを送信します。USBファンクションモジュールはACK送信を完了すると、データ受信完了割り込み（USBIFR0/EP0oTS）を発生します。また、データアウトステージ完了を示すイントークンを受信するとイントークン受信割り込み（USBIFR0/EP0iTR）を発生します。

参考プログラムは、割り込み処理内で割り込みの種類を確認します。データ受信完了割り込みではない場合、ステータスステージ処理に移行します。データ受信完了割り込み（USBIFR0/EP0oTS）の場合は、EP0oFIFO内のデータを読み出し、EP0oFIFO読み出し完了ビットを設定し、次の割り込み発生を待ちます。図 2.10に参考プログラムのデータアウトステージ処理の動作フローを示します。

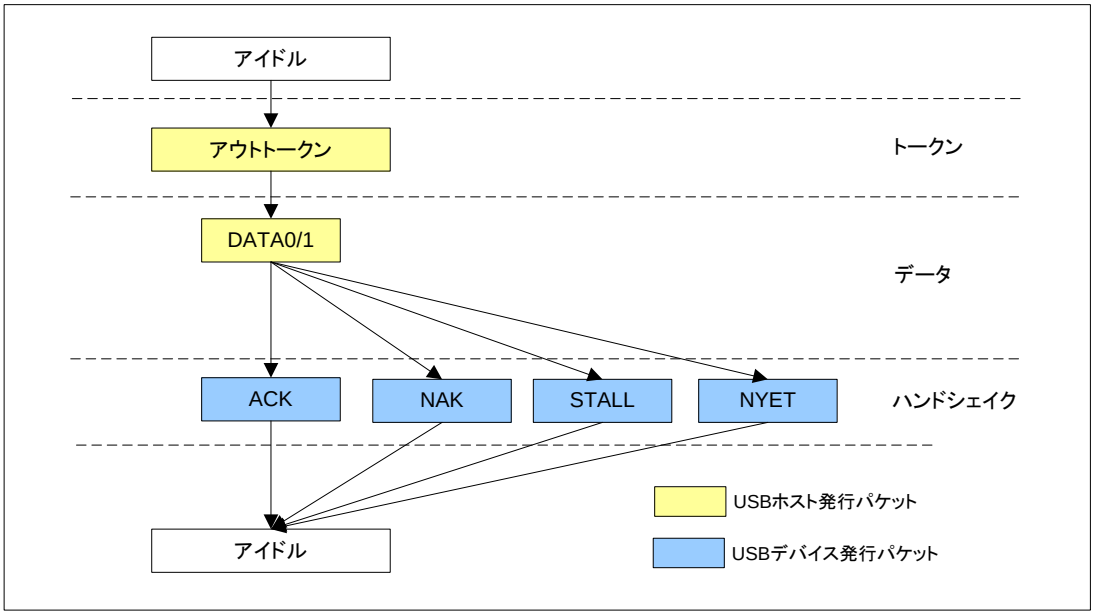

図 2.9　データアウトトランザクション

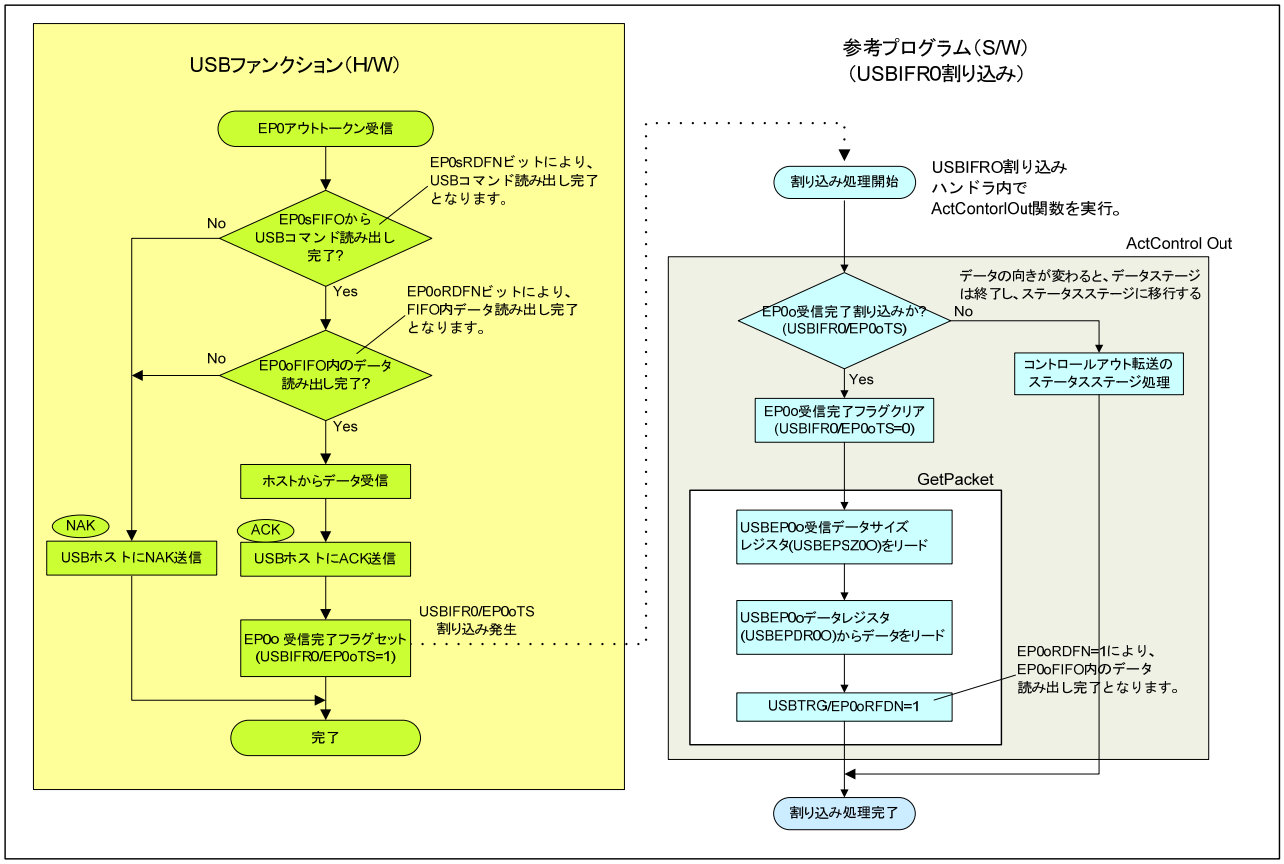

図 2.10 データアウトステージ（コントロールアウト転送）

2.4.3　ステータスステージ処理

ステータスステージでは、データステージと反対方向のデータトランザクションが実行されます。コントロールイン転送のステータスステージでは、データアウトトランザクションが実行され、コントロールアウト転送のステータスステージでは、データアウトトランザクションが実行されます。

(1)コントロールイン転送のステータスステージ処理

USBホストがアウトトークンおよび0バイトのデータを送信します。USBデバイスはアウトトークンを受信後、0バイトのデータを受信し、ACKを送信します（ACK応答）。

USBファンクションモジュールは、アウトトークンを受信した後、0バイトのデータを受信し、ハードウェアにて自動的にUSBホストにACKを送信します。USBファンクションモジュールはACK送信を完了すると、データ受信完了割り込み（USBIFR0/EP0oTS）を発生します。

参考プログラムは、割り込み処理内でEP0oFIFO読み出し完了ビット（USBTRG/EP0oRFDN）を設定し、次の割り込み発生を待ちます。図 2.11に参考プログラムのコントロールイン転送のステータスステージ処理の動作フローを示します。

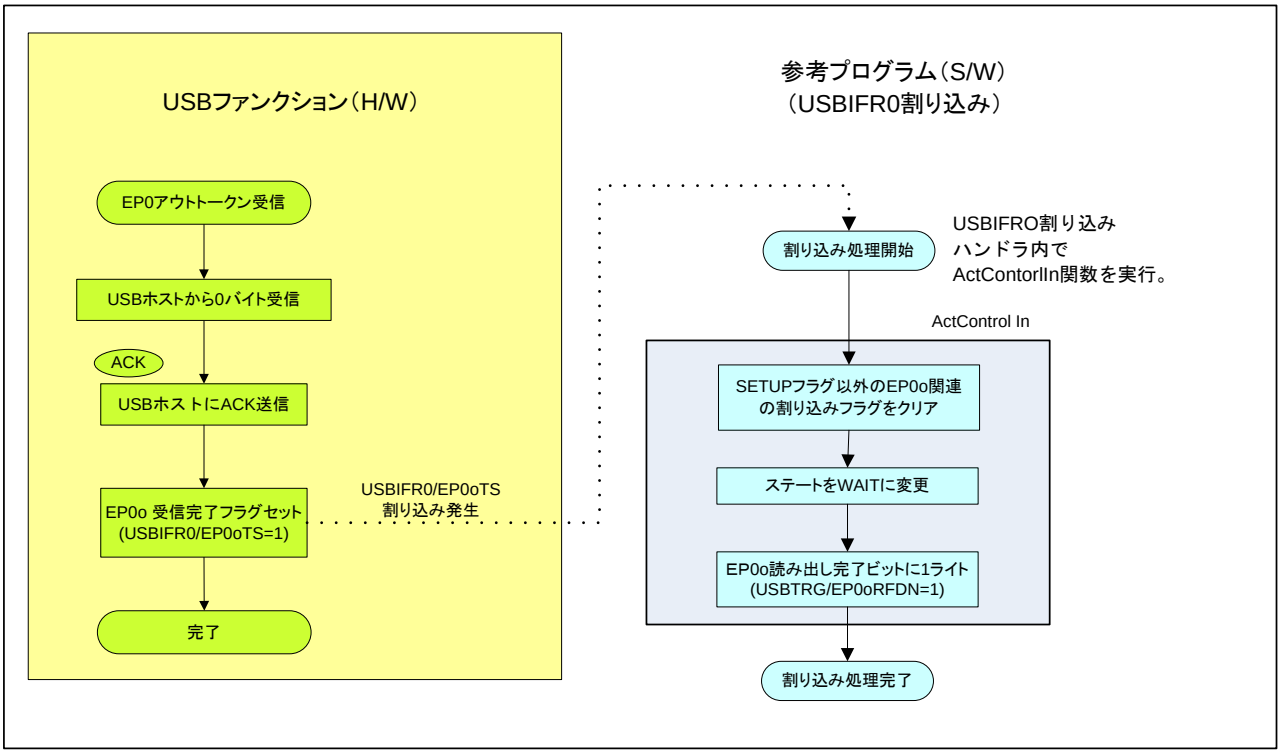

図 2.11 ステータスステージ（コントロールイン転送）

(2)コントロールアウト転送のステータスステージ処理

USBホストがイントークンを送信します。USBデバイスはイントークンを受信後、0バイトのデータをUSBホストに送信し、USBホストからのACK応答を待ちます。

USBファンクションモジュールは、イントークンを受信するとイントークン受信割り込み（USBIFR0/EP0iTR）を発生します。USBファンクションモジュールはEP0iFIFO内に0バイトの有効なデータがない状態で、イントークンを受信した場合、ハードウェアにて自動的にUSBホストにNAKを送信します（NAK応答）。EP0iFIFO内に0バイトの有効なデータがある状態で、イントークンを受信した場合は、USBホストに0バイトのデータを送信し、USBホストからのACK応答を待ちます。USBホストからACKを受信するとデータ送信完了割り込み（USBIFR0/EP0iTS）を発生します。

参考プログラムは、割り込み処理内で割り込みの種類を確認します。データ送信完了割り込み（USBIFR0/EP0iTS）の場合は、EP0iFIFO送信完了（USBIFR0/EP0iTS）をクリアし、コントロール転送を完了します。イントークン受信割り込み（USBIFR0/EP0iTR）の場合は、EP0iFIFO内 0 バイトのデータを有効（USBTRG/EP0iPKTE=1）にし、次の割り込み発生を待ちます。図 2.12に参考プログラムにおけるコントロールアウト転送のステータスステージ処理の動作フローを示します。また、参考プログラムはUSB COMクラスコマンド（Set Line Coding）コマンドである場合、USB COMクラスコマンド処理をSciInit関数にて処理します。

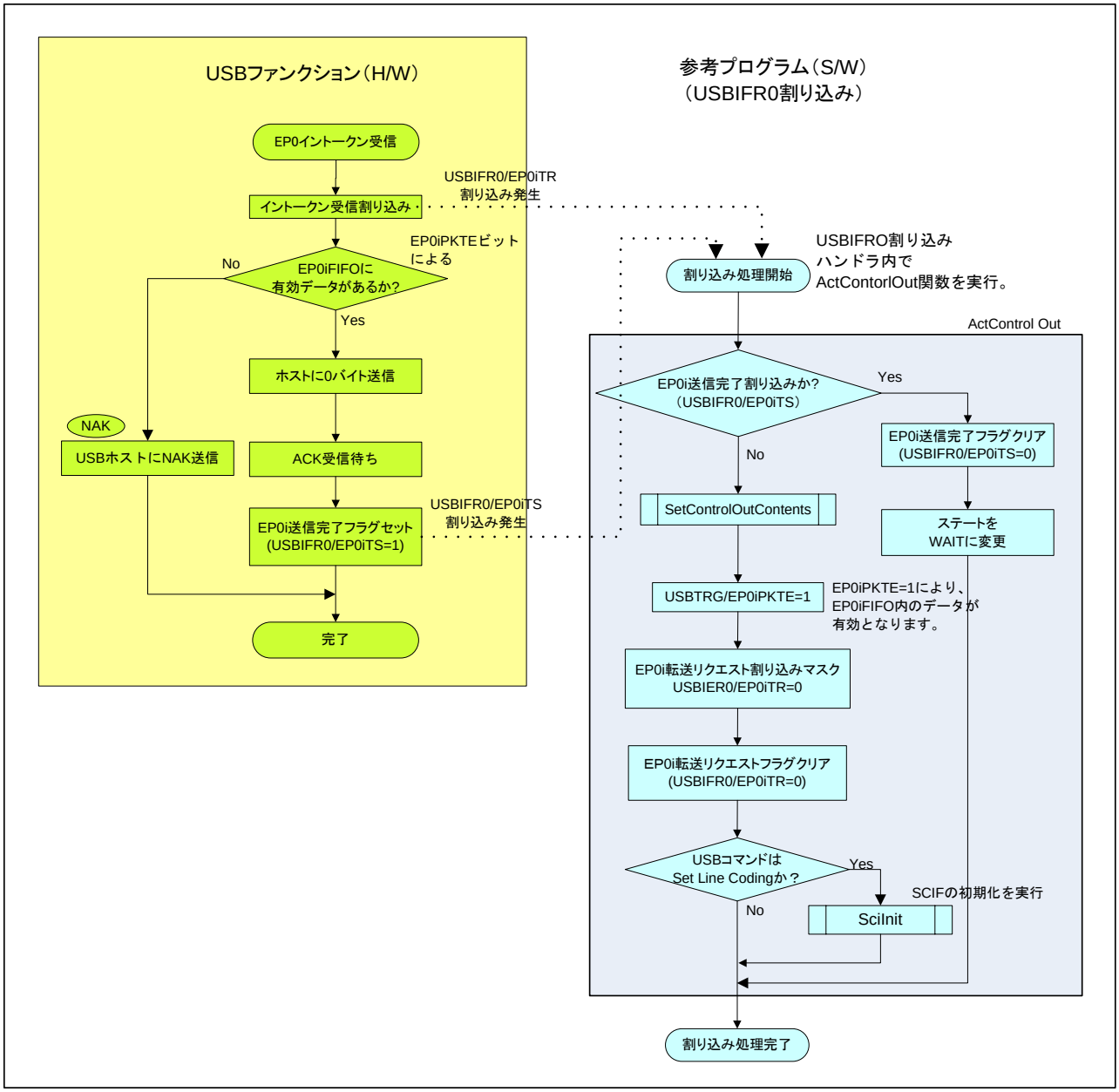

図 2.12 ステータスステージ（コントロールアウト転送）

2.5　バルク転送処理

バルク転送は、USBホストとUSBデバイスの間で大量のデータを送受信する際に使用されるUSB転送処理です。また、バルク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ホストからUSBデバイス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 2.13にバルクイン転送とバルクアウト転送のデータの方向を示します。

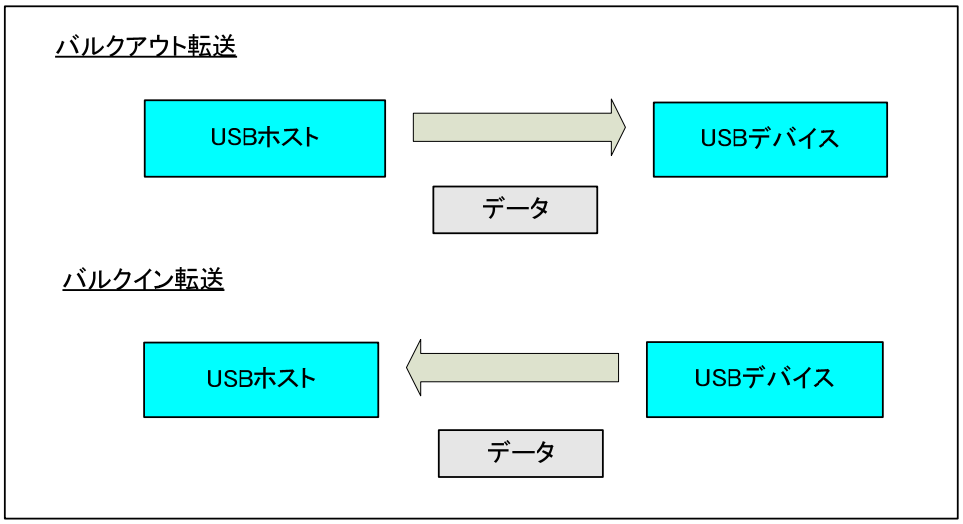

図 2.13 バルク転送

2.5.1　バルクアウト転送

バルクアウト転送は、1 つまたは複数のアウトトランザクションにより構成されます。アウトトランザクションでは、USBホストがアウトトークンおよびデータを送信します。USBデバイスはアウトトークンを受信後、データを受信し、ACKを送信します（ACK応答）。USBデバイスはデータを受信できない状態で、アウトトークンを受信した場合、続けて送信されるデータを無視し、ハンドシェイクにてNAKを送信します。USBホストはハンドシェイクにてNAKを受信した場合は、再度アウトトークンおよびデータを送信します。図 2.14にアウトトランザクションのフローを示します。

USB ファンクションモジュールは、データを受信できない状態で、アウトトークンを受信した場合、USB ホストから続けて送信されるデータをハードウェアが自動的に破棄し、ハンドシェイクにて USB ホストに NAK を送信します。データを受信できる状態で、アウトトークンを受信した場合は、USB ホストから送信されるデータを EP1FIFO 内に保持し、ハンドシェイクにて USB ホストに ACK を送信します。USB ファンクションモジュールは ACK 送信を完了すると、データ受信完了割り込み（USBIFR0/EP1FULL）を発生します。

参考プログラムは、割り込み処理内で EP1FIFO 内のデータをバルクアウト転送用 RAM に格納し、EP1 読み出し完了フラグ（USBTRG/EP1RDFN=1）を設定し、次の割り込み発生を待ちます。割り込み発生時に EP1FIFO 内のデータを格納する領域がない場合は、EP1FULL 割り込みを禁止し、発生した割り込みを保留した状態で割り込み処理を終了します。他処理（シリアル通信処理）により、バルクアウト転送用 RAM に空きが発生し、EP1FULL 割り込みを許可すると、保留されていた EP1FULL 割り込みが発生し、割り込み処理内で EP1FIFO 内のデータをバルクアウト転送用 RAM に格納し、EP1 読み出し完了フラグを設定し、次の割り込み発生を待ちます。

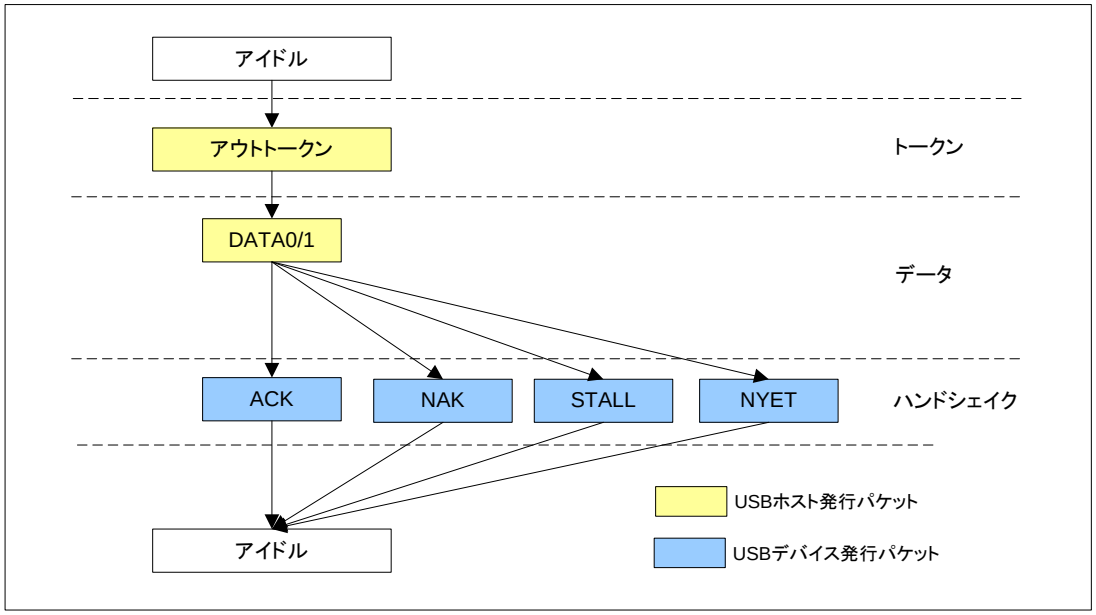

図 2.14　アウトトランザクション

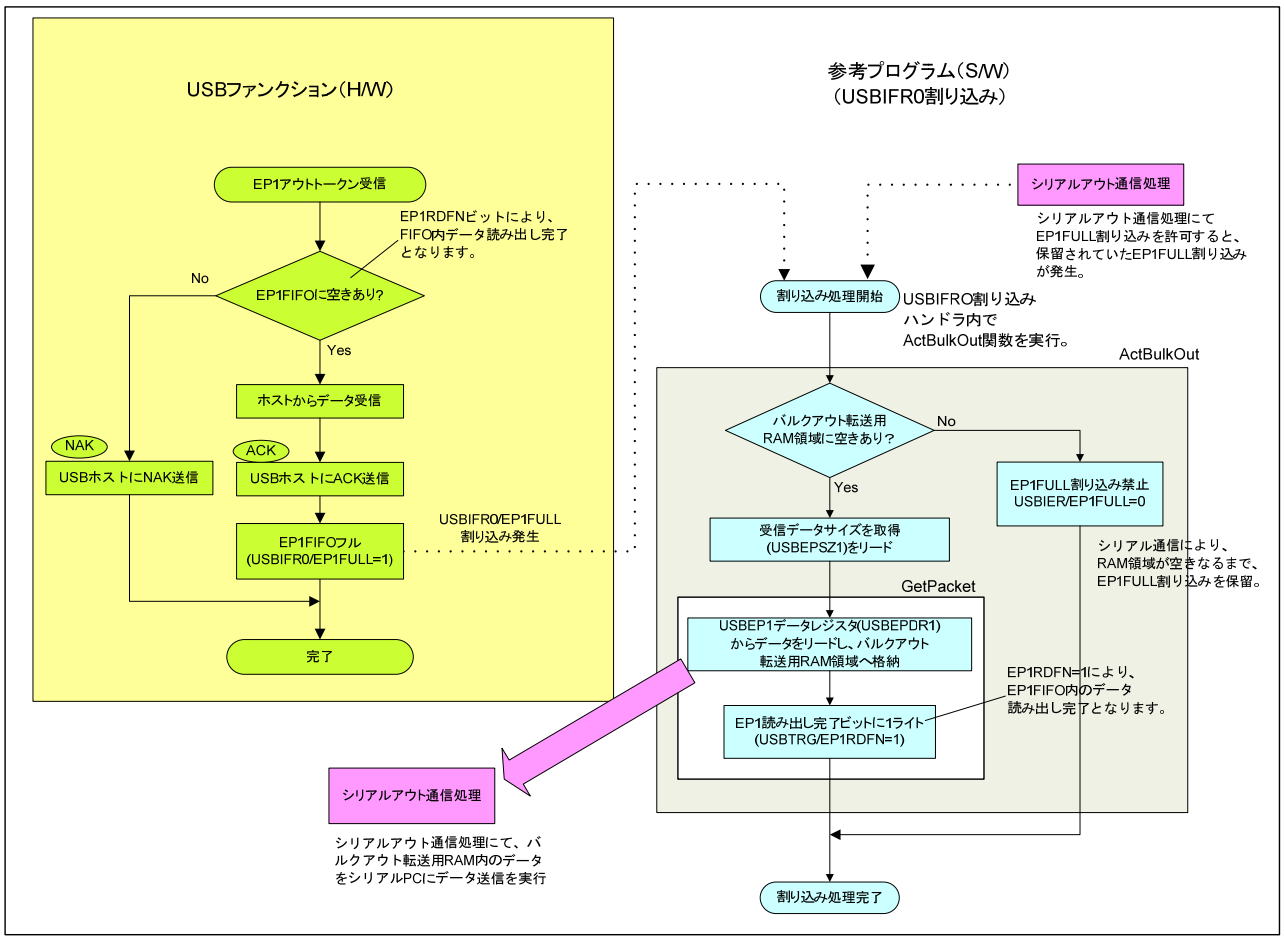

図 2.15　バルクアウト転送

2.5.2　バルクイン転送

バルクイン転送は、1 つまたは複数のイントランザクションにより構成されます。イントランザクションでは、USB ホストがイントークンを送信します。USB デバイスはイントークンを受信すると USB ホストにデータを送信し、USB ホストからの ACK 応答を待ちます。USB デバイスはデータを送信できない状態で、イントークンを受信した場合、USB ホストに NAK を送信します（NAK 応答）。図にイントランザクションのフローを示します。

USB ファンクションモジュールは、EP2FIFO 内に有効なデータない状態でイントークンを受信した場合、ハードウェアが自動的に USB ホストに NAK を送信します。USB ファンクションモジュールは、USBTRG/EP2PKTE により EP2FIFO 内で有効なデータがあるかどうかを判断します。EP2FIFO 内に有効なデータがある状態でイントークンを受信した場合は、EP2FIFO 内のデータを USB ホストに送信し、USB ホストからの ACK 応答を待ちます。ACK 応答を受信すると、データ送信完了フラグ（USBIFR0/EP2EMPTY）を設定します。

参考プログラムは、メインループ処理内で定期的にバルクイン転送処理を行います。他処理（シリアル通信処理）により、USB ホストに送信するべきデータがバルクイン転送用 RAM 内に生成されている場合は、EP2FIFO に空きがあるかどうかを確認します。EP2FIFO に空きがある場合は、そのデータを EP2FIFO に書き込み、PE2PKTE ビットを 1 に設定し、EP2FIFO 内のデータを有効にします。EP2FIFO に空きがない場合は何も行わずに、USB ファンクションモジュールにより、USB ホストへのデータ転送が完了するのを待ちます。

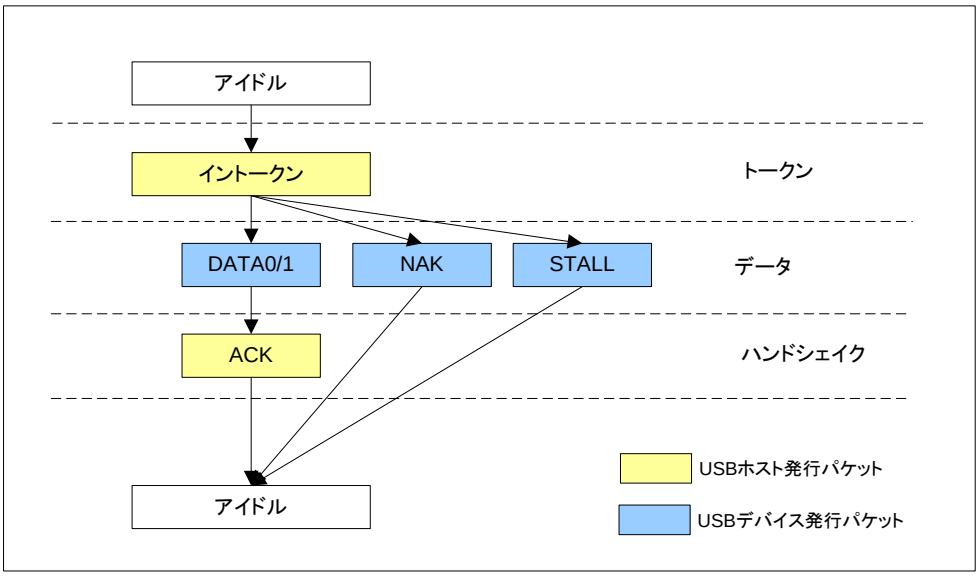

図 2.16　イントランザクション

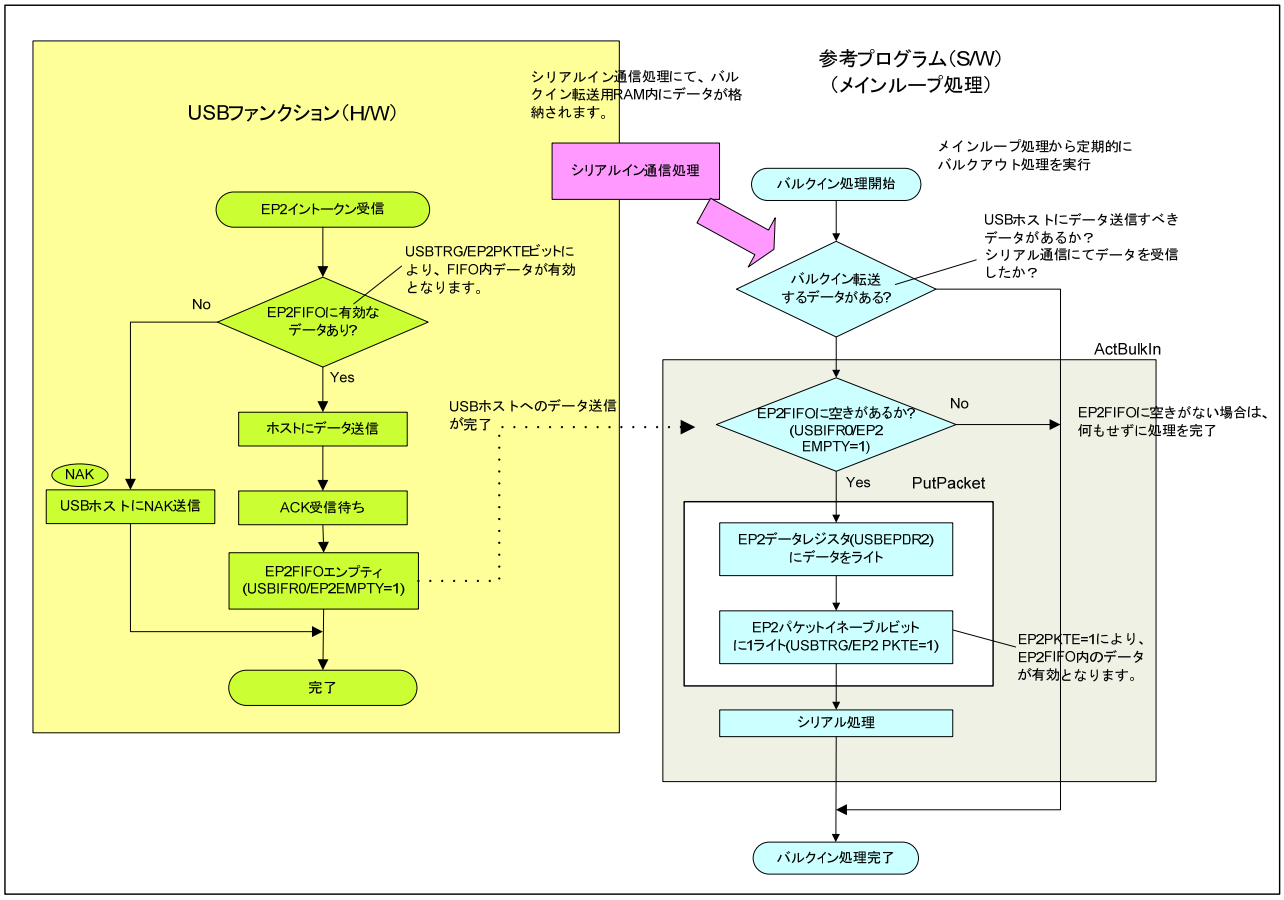

図 2.17 バルクイン転送

3.　USB ファンクションモジュールを使用した USB シリアル変換システム

3.1　　システム概要

参考プログラムでは、SH7285 に内蔵した USB ファンクションモジュールとシリアルコミュニケーションインタフェースを用いて USB シリアル変換機能を実現します。USB ホスト PC とシリアル接続 PC の両者でターミナルソフトを起動することにより、キー入力文字の転送、テキストファイルの転送、および、バイナリファイルの転送が行えます。例えば、USB ホスト PC 側でキー入力することで、その入力文字がシリアル接続 PC へ転送されます。また、シリアル接続 PC 側でキー入力することで、その文字を USB ホスト PC へと転送されます。

図 3.1にUSBシリアル変換のシステム構成例を、表 3.1にUSBシリアル変換の仕様概要を示します。

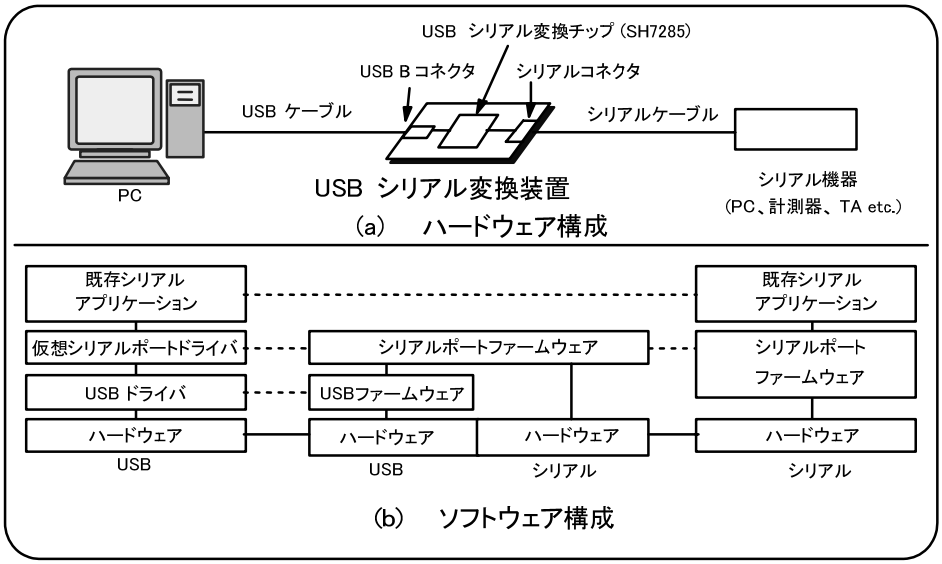

図 3.1　システム構成例

表 3.1　仕様概要

特長	説明
USB ホストへの接続検出	H/W による接続検出後、ポートによる D+端子プルアップ制御を行います。
コントロール転送処理 （USB 標準コマンド）	コントロール転送で USB ホストから送信される USB コマンドに対し、コマンドデコード、データステージおよびステータスステージ処理を実行します。 Get Descriptorコマンドに対し図 3.2～図 3.4のディスクリプタ情報を送信し、COMクラスとしてホストPCと接続します。 USB ホストに送信する各ディスクリプタのサンプルは、SetUsbInfo.h ファイルに記載しています。 図 3.2にデバイス・ディスクリプタサンプルを、図 3.3～図 3.4にコンフィギュレーション・ディスクリプタサンプルを示します。また表 3.2にサンプルのVendorIDおよびProductIDを示します。
コントロール転送処理 （USB COM クラスコマンド）	以下の USB COM クラスコマンド処理を実行します。 Get Line Coding、Set Line Coding、Set Control Line State、Send Break
バルクイン／アウト転送処理	バルクイン／アウト転送処理を実行します。
シリアル通信／USB 通信変換	(1)バルクアウト転送にて USB ホストから受信したデータをシリアルアウト通信にてシリアル機器に送信します。 (2)シリアルイン通信にてシリアル機器から受信したデータをバルクイン転送にて USB ホストに送信します。

```
/*    Descriptor Information */
/*    Device Descriptor      */
      unsigned char          DeviceItem[] = {
              0x12,0x01,0x10,0x01,
              0x02,   /* Communication Class */
              0x00,0x00,0x08,
              0x5B,   /* Vendor = Renesas Technology */
              0x04,   /*      ""Attention"" Please use your company Vendor ID      */
              0x20,   /* Product ID 0x0020 */
              0x00,0x00,0x01,0x00,0x00,
              0x03,   /* Strings Descriptor iSerialNumber */
              0x01
      };

DeviceType           deviceDiscriptorGVar[] = {
      {18,DeviceItem}
};
```

図 3.2　デバイス・ディスクリプタサンプル

【注意事項】

表 3.2にデバイス・ディスクリプタサンプル内のVendor IDおよびProduct IDのデフォルト値を示します。システムに組み込まれまれる場合は、これらの値を必ず変更してください。

表 3.2　Vendor ID および Product ID サンプル

ID	値	内容
Vendor ID	0x045B	Renesas Technology
Product ID	0x0020	Renesas SH7285 Serial-USB

```
/*    Configuration Descriptor      */
      /*     Configuration Descriptor informations , Data length = 67Byte       */
      unsigned char          configurationItem[] = {
             /* Configuration Descriptor */
             0x09,   /*      0:bLength */
             0x02,   /*      1:bDescriptorType */
             0x43,   /*      2:wTotalLength(L) */   /* 43'h (=67) */
             0x00,   /*      3:wTotalLength(H) */
             0x02,   /*  4:bNumInterfaces */
             0x01,   /*  5:bConfigurationValue */
             0x00,   /*  6:iConfiguration */
             0xC0,   /*      7:bmAttributes */      /* self powered */
             0x10,   /*      8:MAXPower */
                     /* Interface Descriptor 1-0-0 [Communication Class] */
                     0x09,   /*      0:bLength */
                     0x04,   /*      1:bDescriptor */
                     0x00,   /*      2:bInterfaceNumber */
                     0x00,   /*      3:bAlternateSetting */
                     0x01,   /*  4:bNumEndpoints */ /* support INT IN (= EP3) */
                     0x02,   /*  5:bInterfaceClass */      /* Communication Interface Class */
                     0x02,   /*  6:bInterfaceSubClass */   /* Abstruct Control Model */
                     0x01,   /*  7:bInterfaceProtocol */   /* Common AT commands */
                     0x00,   /*      8:iInterface */
                     /* Class-Specific Configuration Descriptors */
                     /* Header Functional Descriptor */
                     0x05,   /*  0:length of this desc. */
                     0x24,   /*  1:CS_INTERFACE */
                     0x00,   /*  2:Header Functional Descr. */
                     0x10,   /*  3:bcdCDC Rel. 1.10 */
                     0x01,   /*  4:bcdCDC Rel. 1.10 */
                     /* Abstract Control Management Functional Descriptor */
                     0x04,   /*  0:length of this desc. */
                     0x24,   /*  1:CS_INTERFACE */
                     0x02,   /*  2:Abstr.Contr.Managm. F.D. */
                     0x06,   /*  3:bmCapabilities */       /* support Send Break, Set Line Coding,
                                                             Set Control Line State, Get Line Coding */

                     /* Union Functional Descriptor */
                     0x05,   /*  0:length of this desc. */
                     0x24,   /*  1:CS_INTERFACE */
                     0x06,   /*  2:Union Functional Descr. */
                     0x00,   /*  3:bMasterInterface */
                     0x01,   /*  4:bSlaveInterface0 */
                     /* Call Management Functional Descriptor */
                     0x05,   /*  0:length of this desc. */
                     0x24,   /*  1:CS_INTERFACE */
                     0x01,   /*  2:Call Managment F.D. */
                     0x03,   /*  3:bmCapabilities */
                     0x01,   /*  4:bDataInterface */
                             /* Endpoint Descriptor 1-0-0-0 */
                             0x07,   /*  0:bLength */
                             0x05,   /*  1:bDescriptorType */
                             0x83,   /*  2:bEndpointAddress */     /* IN, number = 3 */
                             0x03,   /*  3:bmAttribute */
                             0x08,   /*  4:wMAXPacketSize_lo */
                             0x00,   /*  5:wMAXPacketSize_hi */
                             0x10,   /*  6:bInterval */
```

図 3.3　コンフィギュレーション・ディスクリプタサンプル(1)

```
                    /* Interface Descriptor 1-1-0 [Data Class Interface] */
                    0x09,   /*  0:bLength */
                    0x04,   /*  1:bDescriptor */
                    0x01,   /*  2:bInterfaceNumber */
                    0x00,   /*  3:bAlternateSetting */
                    0x02,   /*  4:bNumEndpoints */ /* support Bulk OUT (= EP1) and Bulk IN (= EP2) */
                    0x0A,   /*  5:bInterfaceClass */ /* Data Interface Class */
                    0x00,   /*  6:bInterfaceSubClass */   /* unuse */
                    0x00,   /*  7:bInterfaceProtocol */ /* USB */
                    0x00,   /*  8:iInterface */
                           /* Endpoint Descriptor 1-1-0-0 */
                           0x07,   /*  0:bLength */
                           0x05,   /*  1:bDescriptorType */
                           0x01,   /*  2:bEndpointAddress */
                           0x02,   /*  3:bmAttribute */
                           0x40,   /*  4:wMAXPacketSize_lo */
                           0x00,   /*  5:wMAXPacketSize_hi */
                           0x00,   /*  6:bInterval */
                           /* Endpoint Descriptor 1-1-0-1 */
                           0x07,   /*  0:bLength */
                           0x05,   /*  1:bDescriptorType */
                           0x82,   /*  2:bEndpointAddress */
                           0x02,   /*  3:bmAttribute */
                           0x40,   /*  4:wMAXPacketSize_lo */
                           0x00,   /*  5:wMAXPacketSize_hi */
                           0x00    /*  6:bInterval */
        };

        ConfigurationType             configurationDiscriptorGVar[] = {
                {67,configurationItem}
        };
```

図 3.4　コンフィギュレーション・ディスクリプタサンプル(2)

3.2　動作フロー

参考プログラムは、各種初期設定後、メインループに入ります。

参考プログラムは、シリアル機器からシリアルイン通信処理にてデータを受信すると、受信したデータをバルクイン転送用RAM領域に格納します。バルクイン転送用RAMに格納されたデータは、バルクイン転送処理にてUSBホストに送信されます。バルクイン転送処理については「2.5.2 バルクイン転送」を、シリアルイン通信処理については「3.3.2 シリアルイン通信処理」を参照ください。

参考プログラムは、USBホストからバルクアウト転送処理にてデータを受信すると、受信したデータをバルクアウト転送用RAM領域に格納します。バルクアウト転送用RAM領域に格納されたデータは、シリアルアウト通信処理にてシリアル機器にデータを送信します。バルクアウト転送については「2.5.1 バルクアウト転送」を、シリアルアウト通信処理については「3.3.1 シリアルアウト通信処理」を参照ください。

図 3.5に動作フローを、図 3.6にデータフローを示します。

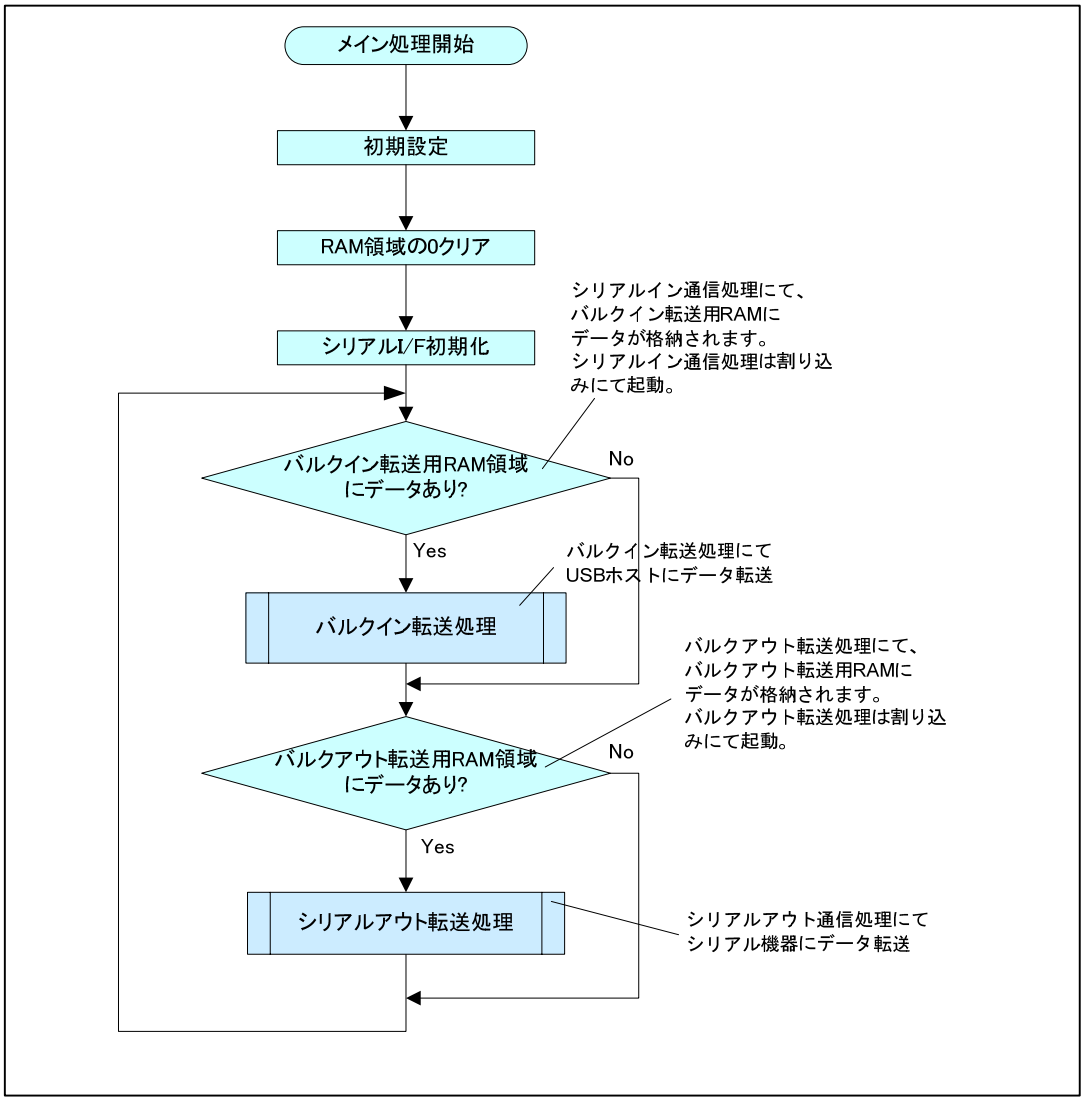

図 3.5　動作フロー

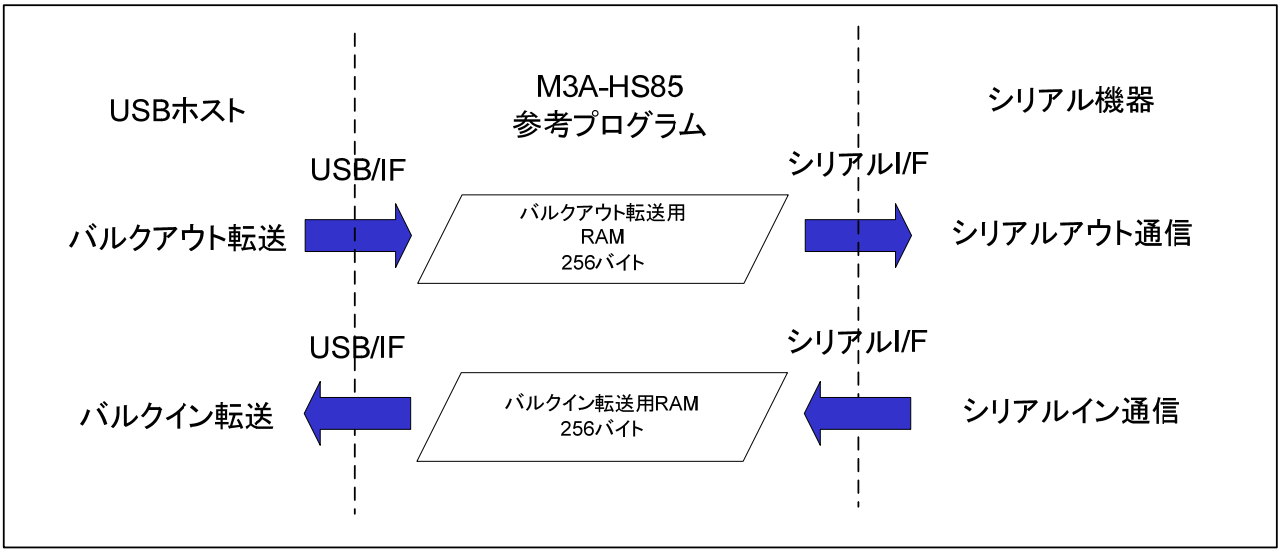
USBホスト
M3A-HS85
参考プログラム
シリアル機器
USB/IF
シリアルI/F
バルクアウト転送
バルクアウト転送用
RAM
256バイト
シリアルアウト通信
USB/IF
シリアルI/F
バルクイン転送
バルクイン転送用RAM
256バイト
シリアルイン通信

図 3.6　データフロー

3.3　シリアル通信処理

参考プログラムでは、シリアル通信のためにSCIFモジュール[1]を使用します。シリアルアウト通信処理はバルクアウト転送用RAMにデータがある場合に実行され、シリアルイン通信処理はシリアル受信割り込みによって実行されます。

3.3.1　シリアルアウト通信処理

シリアルアウト通信処理はActSerialout関数にて実行されます。シリアルアウト通信処理では、バルクアウト転送用RAMのデータをシリアル送信します。シリアル送信処理により、バルクアウト転送用RAM領域に十分な空きを確保できた場合は、EP1FULL割り込みを許可します。EP1FULL割り込みを許可すると、保留されていたEP1FULL割り込みが発生し、割り込み処理にてバルクアウト転送が実行されます。図 3.7にシリアルアウト通信処理の動作フローを示します。

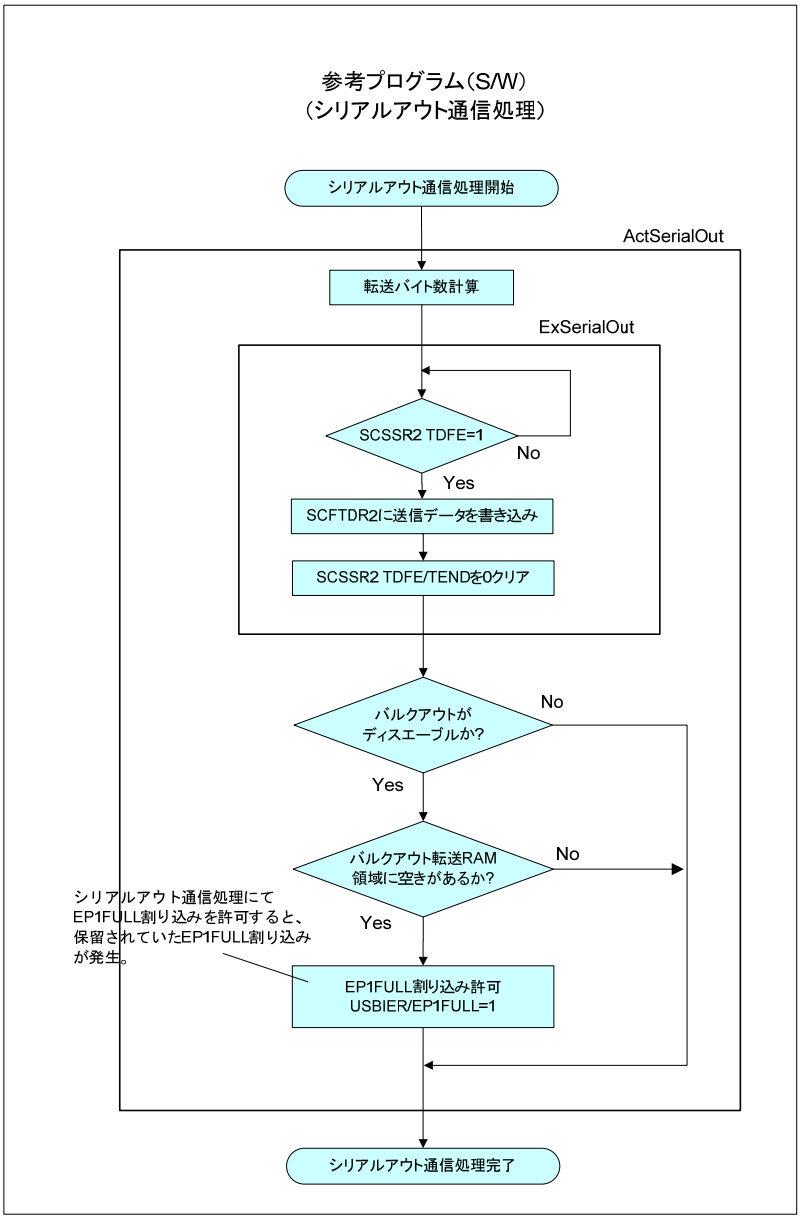

図 3.7　シリアルアウト通信処理

[1] SH7285を搭載したM3A-HS85の場合。SH7286を搭載したM3A-HS87の場合は、SCIモジュールを使用します。

3.3.2　　シリアルイン通信処理

シリアルイン通信処理は、シリアル受信割り込みにて起動し、ActSerialIn関数にて実行されます。シリアルイン通信処理ではシリアル受信データをバルクイン転送用RAM領域に格納します。バルクイン転送用RAMがフルになった場合は、Xoffをシリアル送信しシリアルインをディスエーブルとします。図 3.8にシリアルイン通信処理の動作フローを示します。

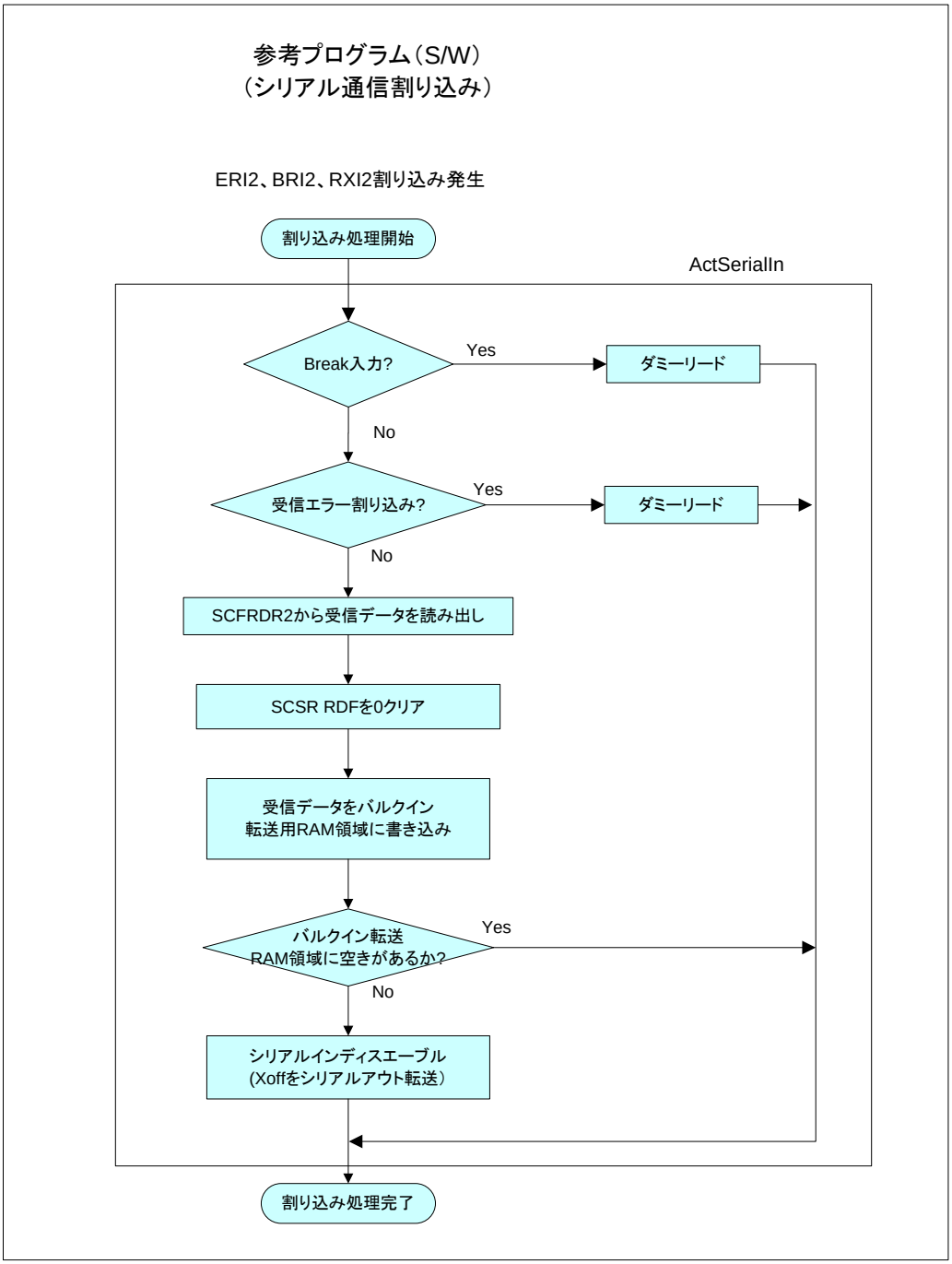

図 3.8　シリアルイン通信処理

3.4　実行環境

M3A-HS85 とPCをRS-232Cシリアルケーブル（クロス接続）で接続し、また、M3A-HS85 とPCをUSBケーブルで接続します。図 3.9に実行環境を示します。

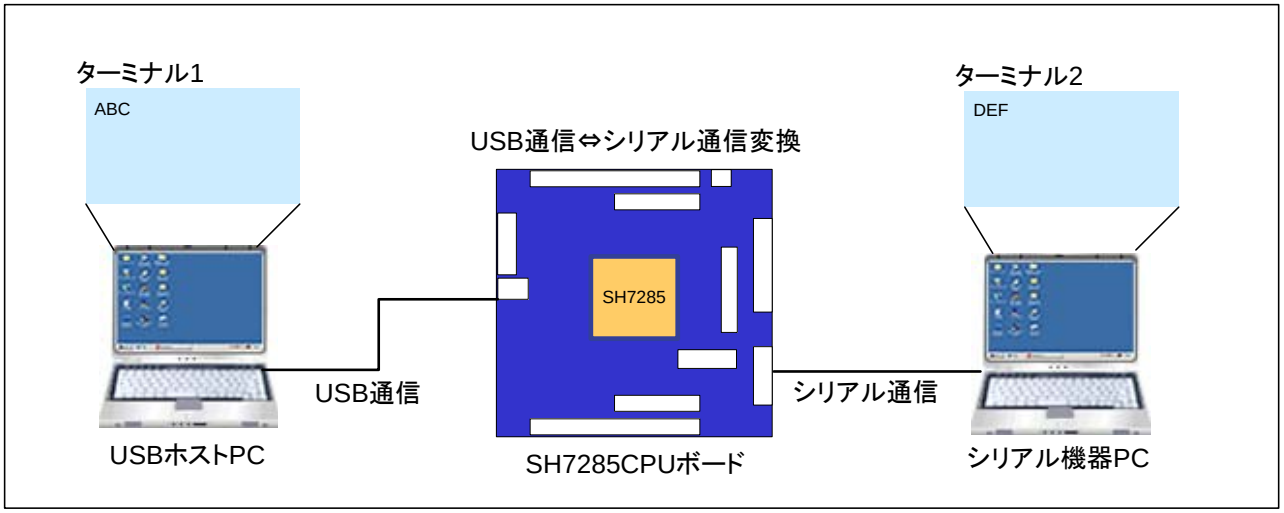

図 3.9　実行環境

3.4.1　USB ホスト PC の設定

参考プログラムをダウンロードしたM3A-HS85 とUSBホストPCをUSBケーブルにて初めて接続した場合、USBホストPCにデバイスドライバをインストールする必要があります。デバイスドライバとして、Windows標準USB COMクラスドライバ（usbser.sys）をインストールします。図 3.10～図 3.12にインストールに使用するWindows用INFファイルの例を示します。

```
;-----------------------------------------------------
;     Renesas Technology Corp
;     USB Serial Ports Driver
;      for Windows2000 and WindowsXP
;      29 September 2003
;-----------------------------------------------------

[Version]
LayoutFile=layout.inf
Signature="$CHICAGO$"
Class=Ports
ClassGuid={4D36E978-E325-11CE-BFC1-08002BE10318}
Provider=%MyCompany%
DriverVer=29/09/2003,1.0.0.0

[DestinationDirs]
DefaultDestDir=12

[Manufacturer]
%MyCompany%=Models

[ClassInstall]
AddReg=PortsClass.AddReg
```

図 3.10　Windows 用 INF ファイル例(1)

```
[PortsClass.AddReg]
HKR,,,,%PortsClassName%

[ClassInstall32.NT]
AddReg=PortsClass.NT.AddReg

[PortsClass.NT.AddReg]
HKR,,,,%PortsClassName%
HKR,,Icon,,"-23"
HKR,,Installer32,,"MsPorts.Dll,PortsClassInstaller"

[ControlFlags]
ExcludeFromSelect=*

;*****************************************
; Please change to your company's VID and PID *
;*****************************************
[Models]
%USB.PnP%=ComPort, USB¥VID_045B&PID_0020     /* USB Vendor ID & Product ID */

[ComPort.NT]
CopyFiles=ComPort.Copy
AddReg=ComPort.AddReg, ComPort.NT.AddReg

[ComPort.NT.HW]
AddReg=ComPort.NT.HW.AddReg

[ComPort.NT.Services]
AddService = usbser, 0x00000002, Serial_Service_Inst,
Serial_EventLog_Inst
AddService = Serenum,,Serenum_Service_Inst

[ComPort.NT.HW.AddReg]
HKR,,"UpperFilters",0x00010000,"serenum"

; -------------- USBSerial Port Driver install sections
[Serial_Service_Inst]
DisplayName    = %Serial.SVCDESC%
ServiceType   = 1            ; SERVICE_KERNEL_DRIVER
StartType     = 3            ; SERVICE_DEMAND_START
ErrorControl   = 1            ; SERVICE_ERROR_NORMAL
ServiceBinary  = %12%¥usbser.sys
LoadOrderGroup = Extended base

; -------------- Serenum Driver install section
[Serenum_Service_Inst]
DisplayName    = %Serenum.SVCDESC%
ServiceType   = 1            ; SERVICE_KERNEL_DRIVER
StartType     = 3            ; SERVICE_DEMAND_START
ErrorControl   = 1            ; SERVICE_ERROR_NORMAL
ServiceBinary  = %12%¥serenum.sys
LoadOrderGroup = PNP Filter

[Serial_EventLog_Inst]
AddReg = Serial_EventLog_AddReg
```

図 3.11　Windows 用 INF ファイル例(2)

```
[Serial_EventLog_AddReg]
HKR,,EventMessageFile,0x00020000,"%%SystemRoot%%¥System32¥IoLogMsg.dll;%
%SystemRoot%%¥System32¥drivers¥usbser.sys"
HKR,,TypesSupported,0x00010001,7

; COM sections
;-----------------------------------------------------------
[ComPort.Copy]
usbser.sys,,,0x20

[ComPort.AddReg]
HKR,,PortSubClass,1,01

[ComPort.NT.Copy]
CopyFiles=ComPort.Copy

[ComPort.NT.AddReg]
HKR,,EnumPropPages32,,"MsPorts.dll,SerialPortPropPageProvider"


;*****************************************
; Please change to your company's information *
;*****************************************
[Strings]
MyCompany="Renesas Technology Corp"
DiskName_Desc="Installation Disk"
PortsClassName = "Ports (COM & LPT)"
Serenum.SVCDESC = "Serenum Filter Driver"
Serial.SVCDESC = "USB Serial Ports Driver"
USB.PnP="USB Serial Ports Driver"
```

図 3.12　Windows 用 INF ファイル例(3)

【注意事項】

VID_045BおよびPID_0020 は、デバイス・ディスクリプタに設定したVendor IDおよびProduct IDに応じて、変更してください。表 3.3にデフォルト値を示します。

表 3.3　Vendor ID および Product ID について

値	意味
VID_045B	Vendor ID = 0x045B であることを示します。
PID_0020	Product ID = 0x0020 であることを示します。

3.4.2　PC 間通信の設定

表 3.4にターミナルソフトウェアに設定する内容を示します。

表 3.4　ターミナルソフトウェア設定

項目	内容
接続方法	RS-232C または USB を接続しているポート番号を選択
ビット／秒（B）	115200 bps
データビット（D）	8
パリティ（P）	なし
ストップビット（S）	1
フロー制御（F）	Xon/Xoff

4.　参考ドキュメント

・ソフトウェアマニュアル
SH-1/SH-2/ SH-DSP ソフトウェアマニュアル　Rev7.00
(最新版をルネサス テクノロジホームページから入手してください)。

・ハードウェアマニュアル
SH7280 グループハードウェアマニュアル　Rev.1.00
(最新版をルネサス テクノロジホームページから入手してください)。

ホームページとサポート窓口

- ルネサステクノロジホームページ
  http://japan.renesas.com/
- お問合せ先
  http://japan.renesas.com/inquiry
  csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.05.13	—	初版発行

本資料ご利用に際しての留意事項

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

D039444