(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4924472号
(P4924472)
(45)発行日　平成24年4月25日(2012.4.25)　(24)登録日　平成24年2月17日(2012.2.17)

(51)Int.Cl.　　F I
G06F 13/38 (2006.01)　　G06F 13/38 350
G06F 13/12 (2006.01)　　G06F 13/12 330A

請求項の数 3　(全 20 頁)

(21)出願番号　特願2008-44631(P2008-44631)
(22)出願日　平成20年2月26日(2008.2.26)
(65)公開番号　特開2009-205271(P2009-205271A)
(43)公開日　平成21年9月10日(2009.9.10)
審査請求日　平成22年1月18日(2010.1.18)

(73)特許権者　000005267
ブラザー工業株式会社
愛知県名古屋市瑞穂区苗代町１５番１号
(74)代理人　110000578
名古屋国際特許業務法人
(72)発明者　河合　竜次
愛知県名古屋市瑞穂区苗代町１５番１号
ブラザー工業株式会社内
(72)発明者　宇野　文敏
愛知県名古屋市瑞穂区苗代町１５番１号
ブラザー工業株式会社内

審査官　木村　貴俊

最終頁に続く

(54)【発明の名称】ホスト機器

(57)【特許請求の範囲】
【請求項１】
シリアル規格のデータを処理可能な電子機器であるシリアル機器とＵＳＢ規格のデータを処理可能な電子機器であるＵＳＢ機器との間に接続された際に、前記シリアル機器が前記ＵＳＢ機器へ送信したシリアル規格のデータを内蔵するバッファに一時的に蓄積するとともにＵＳＢ規格のデータに変換して前記ＵＳＢ機器からの要求に応じて前記ＵＳＢ機器へ送信し、一方、前記ＵＳＢ機器が前記シリアル機器へ送信したＵＳＢ規格のデータをシリアル規格のデータに変換した後に前記シリアル機器へ送信し、前記シリアル機器と前記ＵＳＢ機器との間のデータ転送を中継するＵＳＢシリアル変換器との間でＵＳＢ規格のデータを送受信可能であり、前記バッファに蓄積されるデータの有無を確認するための確認コマンドを、前記バッファの容量および前記ＵＳＢシリアル変換器との間でデータ転送を行う際の単位時間当たりの最大データ転送量に応じて予め設定された周期にて前記ＵＳＢシリアル変換器へ発行するデバイスドライバと、
前記ＵＳＢシリアル変換器の前記バッファに蓄積されたデータを読み出すための読出しコマンドを前記デバイスドライバを介して前記ＵＳＢシリアル変換器へ発行する通信アプリケーションと、
を備え、前記ＵＳＢ機器として機能するホスト機器であって、
前記デバイスドライバは、前記通信アプリケーションが前記読出しコマンドを発行する周期を計測し、前記確認コマンドを発行する周期を予め設定された周期からその計測した周期へ変更することを特徴とするホスト機器。

【請求項２】
シリアル規格のデータを処理可能な電子機器であるシリアル機器とＵＳＢ規格のデータを処理可能な電子機器であるＵＳＢ機器との間に接続された際に、前記シリアル機器が前記ＵＳＢ機器へ送信したシリアル規格のデータを内蔵するバッファに一時的に蓄積するとともにＵＳＢ規格のデータに変換して前記ＵＳＢ機器からの要求に応じて前記ＵＳＢ機器へ送信し、一方、前記ＵＳＢ機器が前記シリアル機器へ送信したＵＳＢ規格のデータをシリアル規格のデータに変換した後に前記シリアル機器へ送信し、前記シリアル機器と前記ＵＳＢ機器との間のデータ転送を中継するＵＳＢシリアル変換器との間でＵＳＢ規格のデータを送受信可能であり、前記バッファに蓄積されるデータの有無を確認するための確認コマンドを、前記バッファの容量および前記ＵＳＢシリアル変換器との間でデータ転送を行う際の単位時間当たりの最大データ転送量に応じて予め設定された周期にて前記ＵＳＢシリアル変換器へ発行するデバイスドライバと、
前記ＵＳＢシリアル変換器の前記バッファに蓄積されたデータを読み出すための読出しコマンドを前記デバイスドライバを介して前記ＵＳＢシリアル変換器へ発行する通信アプリケーションと、
を備え、前記ＵＳＢ機器として機能するホスト機器であって、
前記通信アプリケーションは、前記シリアル機器との間でデータ転送を行う際の単位時間当たりのデータ転送量であるボーレートを設定可能であり、
前記デバイスドライバは、前記通信アプリケーションが設定したボーレートに対応する周期を算出し、前記確認コマンドを発行する周期を予め設定された周期からその算出した周期へ変更すること
を特徴とするホスト機器。
【請求項３】
シリアル規格のデータを処理可能な電子機器であるシリアル機器とＵＳＢ規格のデータを処理可能な電子機器であるＵＳＢ機器との間に接続された際に、前記シリアル機器が前記ＵＳＢ機器へ送信したシリアル規格のデータを内蔵するバッファに一時的に蓄積するとともにＵＳＢ規格のデータに変換して前記ＵＳＢ機器からの要求に応じて前記ＵＳＢ機器へ送信し、一方、前記ＵＳＢ機器が前記シリアル機器へ送信したＵＳＢ規格のデータをシリアル規格のデータに変換した後に前記シリアル機器へ送信し、前記シリアル機器と前記ＵＳＢ機器との間のデータ転送を中継するＵＳＢシリアル変換器との間でＵＳＢ規格のデータを送受信可能であり、前記バッファに蓄積されるデータの有無を確認するための確認コマンドを、前記バッファの容量および前記ＵＳＢシリアル変換器との間でデータ転送を行う際の単位時間当たりの最大データ転送量に応じて予め設定された周期にて前記ＵＳＢシリアル変換器へ発行するデバイスドライバと、
前記ＵＳＢシリアル変換器の前記バッファに蓄積されたデータを読み出すための読出しコマンドを前記デバイスドライバを介して前記ＵＳＢシリアル変換器へ発行する通信アプリケーションと、
を備え、前記ＵＳＢ機器として機能するホスト機器であって、
前記確認コマンドの発行を停止する旨を入力するための入力手段を備え、
前記デバイスドライバは、前記確認コマンドの発行を停止する旨が入力手段から入力された場合には前記確認コマンドの発行を停止すること
を特徴とするホスト機器。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ＵＳＢシリアル変換器を用いてシリアル機器とホスト機器との間のデータ転送を行う場合に、ＵＳＢシリアル変換器に蓄積されたデータを速やかに読み出しながら、ホスト機器の処理負荷を低減する技術に関する。
【背景技術】
【０００２】

従来より、パーソナルコンピュータ等のホスト機器に接続されて使用されるプリンタやモデム等の電子機器が知られている。このような電子機器は、例えば、ＲＳ－２３２Ｃやセントロニクス等のインタフェースを備えており、所定のケーブルを介してホスト機器と接続され、ホスト機器からの制御等に従って動作する。

【０００３】

近年、ホスト機器に採用されるインタフェース規格として、ＵＳＢ（Universal Serial Bus）が多くなっている。このＵＳＢは、ホスト機器において導入されている一般的なＯＳ（Operating System）にてサポートされており、電子機器の電源を入れたままで接続できるホット・プラグ・インや、電子機器を使用するための設定を自動的に行えるプラグ・アンド・プレイが特徴となっている（例えば、特許文献１参照。）。そして、上述のＲＳ－２３２Ｃ等のような従来型のインタフェースを備えた電子機器と、ＵＳＢ等の最新型のインタフェースを備えた電子機器とを混在して使用する機会が多い。その際、いわゆるＵＳＢシリアル変換器を用いて両者を接続する。

【０００４】

このＵＳＢシリアル変換器は、パーソナルコンピュータ等のホスト機器に接続することでシリアル端子を増設するものであり、ホスト機器のアプリケーション層からのアクセスには旧来のシリアル用ＡＰＩを利用する。また、ＵＳＢシリアル変換器用のドライバについても、旧来のシリアル用のドライバを模擬した動作をすることで、ＡＰＩレベルでは互換性を保っている。このような実装方式によって、ＵＳＢシリアル変換器を利用しても、シリアル通信用の通信ソフトウェア等を利用することができる。

【０００５】

なお、シリアル規格でのデータ転送では双方向且つ非同期にデータ転送が行われ、一方、ＵＳＢ規格でのデータ転送ではフレーム構造のパケット転送が行われる。このため、ＵＳＢシリアル変換器とホスト機器との間のデータ交換プロトコルが問題になる。すなわち、ＵＳＢシリアル変換器では、ＲｘＤからビット列を受信すると、受信したデータをバイト変換してバッファに保持する。この場合、ＵＳＢシリアル変換器内部のバッファの容量は一般的に小さく設定されているため、バッファに空き容量が無くなる前に、バッファが記憶するデータをホスト機器が読み出す必要がある。

【０００６】

そのため、ホスト機器では、ＵＳＢシリアル変換器のバッファにデータが蓄積されているか否かを確認するためにインタラプト転送を行うことやバルク転送（ポーリング）を定期的に行うことが一般的である。

【０００７】

しかし、ホスト機器がインタラプト転送を行う場合には、転送周期を決定して一定量のパケット転送を常に行うため、バッファにデータが蓄積されていないときでもＵＳＢバスを一定幅占有するという問題がある。また、ホスト機器がバルク転送を行う場合には、ホスト機器が制御コマンドを定期的に発行し、それに対してＵＳＢシリアル変換器がデータ返信を行う処理を繰り返すことになり、この場合も同様に、バッファにデータが蓄積されていないときでもＵＳＢバスを一定幅占有するという問題がある。つまり、シリアル方式でのデータ転送では、転送すべきデータが発生する頻度が低く、転送すべきデータが存在しない期間が多いにもかかわらず、転送すべきデータの有無を確認するためのホスト機器やＵＳＢシリアル変換器の負担が大きくなっている。

【０００８】

なお、上述のようにインタラプト転送やバルク転送を行う際の転送周期については、シリアル方式でデータ転送を行う場合の最大転送速度（一般に１１５２００ｂｐｓ）に対して余裕があるものでなければならない。一例を挙げると、ＵＢＳシリアル変換器の制御ＬＳＩが有するデータ受信バッファの容量が、例えば英ＦＴＤＩ社（Future Technology Devices International Ltd.）製「ＦＴ２３２Ｒ」の場合には２５６ｂｙｔｅ程度なので、２２ｍｓｅｃ以下の転送周期でインタラプト転送を行い、２２ｍｓｅｃ以下の周期でバルク転送を行うといった具合である。

【特許文献１】特開２００２－１９７０４８号公報（第２頁、図２）
【発明の開示】
【発明が解決しようとする課題】
【０００９】
しかし、このようなＵＳＢシリアル変換器を用いてシリアル機器とホスト機器との間のデータ転送方法においては、旧来のシリアル通信用のドライバを介してシリアル機器とのデータ転送を行う場合に比べて、ホスト機器が備えるＣＰＵが行う処理の負荷が増大するという問題がある。その理由は、ホスト機器が備えるＵＳＢホストコントローラＬＳＩには、ＰＣＩバス接続として「ＯＨＣＩ」、「ＥＨＣＩ」および「ＵＨＣＩ」の三種類があるが、何れの場合にもＵＳＢ規格ではパケット１個の受信につき１度の割り込み処理を行う必要があり、そのため、転送周期での割込み処理を行う必要があるからである。因みに、旧来のシリアル用のドライバを介してシリアル機器とのデータ転送を行う場合には、データ転送を行う場合のみ上述のような割込み処理を行うこととなるため、転送すべきデータが発生する頻度が低いシリアル機器とのデータ転送においては、割込み処理を行う頻度も著しく少なく、仮に割込み処理を行う周期が短くても、実際にホスト機器が備えるＣＰＵが行う処理の負荷は小さい。
【００１０】
また、シリアル機器の中には、センサ機器や操作端末などのように例えば数分につき数バイト程度のデータ転送を行う機器や、処理負荷の変動幅が大きい機器が多い。さらに、ＵＳＢ規格のデータ転送においては、ＵＳＢシリアル変換器から転送トリガを発行することが規格上できない。そのため、ＵＳＢシリアル変換器では、バッファに空き容量が無くなる前にバッファが記憶するデータをホスト機器が読み出す必要があるために、バルク転送を行う場合にはホスト機器のＣＰＵの負荷が大きくなることを承知の上でその転送周期を可能な限り短く設定している。つまり、転送すべきデータが発生する頻度が低いにもかかわらず、バッファに蓄積されたデータの有無を頻繁に確認するようにしている。
【００１１】
本発明は、このような課題に鑑みなされたものであり、その目的とするところは、ＵＳＢシリアル変換器を用いてシリアル機器とホスト機器との間のデータ転送を行う場合に、ＵＳＢシリアル変換器に蓄積されたデータを速やかに読み出しながら、ホスト機器の処理負担を低減する技術を提供することにある。
【課題を解決するための手段】
【００１２】
上記課題を解決するためになされた請求項１に係るホスト機器は、ＵＳＢシリアル変換器との間でＵＳＢ規格のデータを送受信可能なデバイスドライバと、通信アプリケーションを備える。
【００１３】
なお、ＵＳＢシリアル変換器とは、シリアル規格のデータを処理可能な電子機器である「シリアル機器」とＵＳＢ規格のデータを処理可能な電子機器である「ＵＳＢ機器」と間のデータ転送を中継する機能を有する。具体的には、ＵＳＢシリアル変換器は、シリアル機器とＵＳＢ機器との間に接続された際に、シリアル機器がＵＳＢ機器へ送信したシリアル規格のデータを内蔵するバッファに一時的に蓄積するとともにＵＳＢ規格のデータに変換してＵＳＢ機器からの要求に応じてＵＳＢ機器へ送信し、一方、ＵＳＢ機器がシリアル機器へ送信したＵＳＢ規格のデータをシリアル規格のデータに変換した後にシリアル機器へ送信する。なお、シリアル規格のデータをＵＳＢ規格のデータに変換する時期としては、バッファにデータを蓄積する前であってもよいし、バッファに蓄積したデータを読み出した後であってもよい。
【００１４】
そして、デバイスドライバは、上述のようにＵＳＢシリアル変換器との間でＵＳＢ規格のデータを送受信可能であり、ＵＳＢシリアル変換器に対して各種コマンドを発行可能であり、バッファに蓄積されるデータの有無を確認するための確認コマンドを予め設定され

た周期にてＵＳＢシリアル変換器へ発行する。なお、この予め設定された周期については、バッファの容量およびＵＳＢシリアル変換器との間でデータ転送を行う際の単位時間当たりの最大データ転送量に応じて予め設定される。

【００１５】

また、通信アプリケーションは、読出しコマンドをデバイスドライバを介してＵＳＢシリアル変換器へ発行する。なお、読出しコマンドとは、ＵＳＢシリアル変換器のバッファに蓄積されたデータを読み出すためのコマンドであり、ＵＳＢシリアル変換器のバッファの空き容量が無くならないように設定された周期で発行される。

【００１６】

そして、上述のデバイスドライバは、通信アプリケーションが読出しコマンドを発行する周期を計測し、確認コマンドを発行する周期を予め設定された周期からその計測した周期へ変更する。なお、ＵＳＢシリアル変換器に蓄積されたデータについては、通信アプリケーションが読出しコマンドを発行される度に、ＵＳＢシリアル変換器からホスト機器にデータ転送される。

【００１７】

なお、上述のようなデバイスドライバおよび通信アプリケーションの機能により、ホスト機器は、ＵＳＢ機器として機能することとなる。

このようにすると、通信アプリケーションが読出しコマンドを発行する周期は、デバイスドライバが確認コマンドを発行する周期よりも長いために、変更後の確認コマンドを発行する周期が変更前に比べて長くなる。なお、ＵＳＢシリアル変換器のバッファに蓄積されたデータについては、通信アプリケーションが読出しコマンドを発行する度に、ＵＳＢシリアル変換器からホスト機器にデータ転送されるので、バッファの空き容量が無くなることはない。したがって、ＵＳＢシリアル変換器に蓄積されたデータを速やかに読み出しながら、ホスト機器の処理負担を軽減することができる。

【００１８】

また、上述の確認コマンドを発行する周期を予め設定された周期から通信アプリケーションが設定したデータ転送速度に対応する周期へ変更することが考えられる。具体的には、請求項２のように、通信アプリケーションが、シリアル機器との間でデータ転送を行う際の単位時間当たりのデータ転送量であるＲＳ２３２Ｃのシリアル通信速度、つまりボーレート（ｂｐｓ：Ｂｉｔｓ　Ｐｅｒ　Ｓｅｃｏｎｄ）を設定可能であり、デバイスドライバが、通信アプリケーションによって設定したボーレートに対応する周期を算出し、確認コマンドを発行する周期を予め設定された周期からその算出した周期へ変更することが考えられる。

【００１９】

このようにすると、ボーレートが、予め設定される周期を算出する際に用いられる「単位時間当たりの最大データ転送量」よりも小さいため、変更後の確認コマンドを発行する周期が変更前に比べて長くなる。なお、ＵＳＢシリアル変換器のバッファに蓄積されたデータについては、上述のように、通信アプリケーションが読出しコマンドを発行される度に、ＵＳＢシリアル変換器からホスト機器にデータ転送されるので、バッファの空き容量が無くなることはない。したがって、ＵＳＢシリアル変換器に蓄積されたデータを速やかに読み出しながら、ホスト機器の処理負担を軽減することができる。

【００２０】

また、上述の確認コマンドの発行を停止することが考えられる。具体的には、請求項３のように、確認コマンドの発行を停止する旨を入力するための入力手段を備え、確認コマンドの発行を停止する旨が入力手段から入力された場合には、デバイスドライバが、確認コマンドの発行を停止することが考えられる。なお、確認コマンドの発行を停止する旨を入力手段に入力する方法としては、利用者による手動入力や当該ホスト機器によるコマンド発行などが挙げられる。なお、ＵＳＢシリアル変換器のバッファに蓄積されたデータについては、上述のように、通信アプリケーションが読出しコマンドを発行される度に、ＵＳＢシリアル変換器からホスト機器にデータ転送されるので、バッファの空き容量が無く

なることはない。したがって、ＵＳＢシリアル変換器に蓄積されたデータを速やかに読み出しながら、ホスト機器の処理負担を軽減することができる。

【発明を実施するための最良の形態】

【００２１】

以下に本発明の実施形態を図面とともに説明する。

［第一実施形態］

図１はＵＳＢシリアル変換器を用いてホスト機器とシリアル機器とを接続する例を示す説明図である。また、図２はホスト機器およびＵＳＢシリアル変換器の構成を示すハードブロック図である。また、図３はホスト機器の構成を示すソフトブロック図である。なお、ＵＳＢ機器とは、ＵＳＢ規格のデータを処理可能な電子機器であり、シリアル機器とは、シリアル規格のデータを処理可能な電子機器である。

【００２２】

［ＵＳＢシリアル変換器２０の構成の説明］

ＵＳＢシリアル変換器２０は、ＵＳＢ機器を接続可能なＵＳＢ（Ｂ）コネクタ２１と、ＵＳＢシリアル変換ＬＳＩ２２と、シリアル機器を接続可能なシリアルコネクタ２３と、を備える。なお、ＵＳＢシリアル変換器２０内部では、ＵＳＢ（Ｂ）コネクタ２１とＵＳＢシリアル変換ＬＳＩ２２のＵＳＢデバイスコントローラ回路ブロック２２ａ（後述）とが接続され、ＵＳＢシリアル変換ＬＳＩ２２のシリアル回路ブロック２２ｃ（後述）とシリアルコネクタ２３とが接続されている。

【００２３】

ＵＳＢシリアル変換ＬＳＩ２２は、各種処理を実行するマイコンＣＰＵ回路ブロック２２ｂと、マイコンＣＰＵ回路ブロック２２ｂとＵＳＢ（Ｂ）コネクタ２１に接続されたＵＳＢ機器との間のデータのやり取りを仲介するＵＳＢデバイスコントローラ回路ブロック２２ａと、マイコンＣＰＵ回路ブロック２２ｂとシリアルコネクタ２３に接続されたシリアル機器との間のデータのやり取りを仲介するシリアル回路ブロック２２ｃと、を有する。

【００２４】

このように構成されたＵＳＢシリアル変換器２０は、シリアル機器とＵＳＢ機器と間のデータ転送を中継する機能を有する。具体的には、ＵＳＢシリアル変換器２０は、シリアル機器とＵＳＢ機器との間に接続された際に、シリアル機器がＵＳＢ機器へ送信したシリアル規格のデータを内蔵するバッファ（図示省略）に一時的に蓄積するとともにＵＳＢ規格のデータに変換してＵＳＢ機器からの要求に応じてＵＳＢ機器へ送信し、一方、ＵＳＢ機器がシリアル機器へ送信したＵＳＢ規格のデータをシリアル規格のデータに変換した後にシリアル機器へ送信する。なお、シリアル規格のデータをＵＳＢ規格のデータに変換する時期としては、バッファにデータを蓄積する前であってもよいし、バッファに蓄積したデータを読み出した後であってもよい。

【００２５】

［ホスト機器１０の構成の説明］

ホスト機器１０は、複数のソフトウェアに基づく処理を時分割で並列に実行するＣＰＵ１１と、ＰＣチップセット１２と、ＵＳＢ機器に対してホストとして各種処理を実行するＵＳＢホストコントローラＬＳＩ１３と、ＵＳＢ機器を接続可能なＵＳＢ（Ａ）コネクタ１４と、シリアル機器を接続可能なシリアルコネクタ１５と、を備え、ＵＳＢ機器として機能する。なお、本実施形態のホスト機器１０は、４個のＵＳＢ（Ａ）コネクタ１４および１個のシリアルコネクタ１５を備える。

【００２６】

なお、ホスト機器１０内部では、ＣＰＵ１１とＰＣチップセット１２とが接続され、ＰＣチップセット１２のＰＣＩバスブリッジ回路ブロック１２ｂ（後述）とＵＳＢホストコントローラＬＳＩ１３とがＰＣＩバスを介して接続され、ＰＣチップセット１２のシリアル回路ブロック１２ａ（後述）とシリアルコネクタ１５とが接続され、ＵＳＢホストコントローラＬＳＩ１３とＵＳＢ（Ａ）コネクタ１４とが接続されている。

【００２７】
また、ＰＣチップセット１２には、ＣＰＵ１１とシリアル機器との間のデータのやり取りを仲介するシリアル回路ブロック１２ａや、ＣＰＵ１１とＰＣＩバスに接続される各種構成と間のデータのやり取りを仲介するＰＣＩバスブリッジ回路ブロック１２ｂなどが搭載されている。
【００２８】
また、ＵＳＢホストコントローラＬＳＩ１３は、ＣＰＵ１１に制御されて、ＵＳＢ規格のデータを生成し、その生成したデータをＵＳＢ（Ａ）コネクタ１４を介してＵＳＢ（Ａ）コネクタ１４に接続されるＵＳＢ機器に出力するとともに、ＵＳＢ機器からＵＳＢ（Ａ）コネクタ１４を介して送信されてくる通信信号を受信して、受信したデータをＣＰＵ１１に入力する。
【００２９】
また、ホスト機器１０は、図３に示すように、アプリケーション層に通信アプリケーション１０１を備え、ＡＰＩ層にシリアルＡＰＩ１０２を備え、デバイスドライバ層にＵＳＢシリアル変換器デバイスドライバ１０３、１６５５０シリアルデバイスドライバ１０４、およびＵＳＢホストコントローラデバイスドライバ１０５を備える。なお、これらホスト機器１０が備える各種アプリケーションソフトについては、上述のＣＰＵ１１、ＰＣチップセット１２、およびＵＳＢホストコントローラＬＳＩ１３によって実行される。
【００３０】
デバイスドライバとしてのＵＳＢシリアル変換器デバイスドライバ１０３は、ＵＳＢホストコントローラデバイスドライバ１０５を介してＵＳＢシリアル変換器２０との間でＵＳＢ規格のデータを送受信可能である。また、ＵＳＢシリアル変換器デバイスドライバ１０３は、ＵＳＢシリアル変換器２０に対して各種コマンドを発行可能であり、ＵＳＢシリアル変換器２０のバッファに蓄積されるデータの有無を確認するための確認コマンド（ポーリング）を予め設定された周期にてＵＳＢシリアル変換器２０へ発行する。なお、この予め設定された周期については、バッファの容量およびＵＳＢシリアル変換器２０との間でデータ転送を行う際の単位時間当たりの最大データ転送量に応じて予め設定され、本実施形態では１６ｍｓｅｃに設定される。
【００３１】
［ホスト機器１０のその他の構成について］
なお、本実施形態のホスト機器１０のその他の構成については、公知技術に従っているので、ここでは詳細な説明は省略する。
【００３２】
［従来のホスト機器１０とシリアル機器３０とのデータ送受信プロセスの説明］
次に、ＵＳＢシリアル変換器２０を用いてホスト機器１０とシリアル機器３０とを接続し、ホスト機器１０とシリアル機器３０との間でデータの送受信を行う際の従来のプロセスを（１）ＯＰＥＮコマンド発行、（２）ＲＥＡＤコマンド発行、（３）ＷＲＩＴＥコマンド発行、および（４）ＣＬＯＳＥコマンド発行の順に、図４を参照して説明する。なお、図４は従来のホスト機器１０とシリアル機器３０とのデータ送受信プロセスを説明するシーケンス図である。
【００３３】
（１）ＯＰＥＮコマンド発行
通信アプリケーション１０１とＵＳＢシリアル変換器２０との間でデータの送受信を開始する際には、通信アプリケーション１０１がＯＰＥＮコマンドを発行する。すなわち、通信アプリケーション１０１が、ＵＳＢシリアル変換器２０とのデータの送受信を開始するためにＯＰＥＮコマンドを発行すると、そのＯＰＥＮコマンドはシリアルＡＰＩ１０２を介してＵＳＢシリアル変換器デバイスドライバ１０３に送信される。ＵＳＢシリアル変換器デバイスドライバ１０３では、ＯＰＥＮコマンドを受信するとＯＰＥＮ返値を発行する。このＯＰＥＮ返値は、シリアルＡＰＩ１０２を介して通信アプリケーション１０１へ送信される。また、ＵＳＢシリアル変換器デバイスドライバ１０３では、ＵＳＢシリアル

変換器２０に対してコマンドを発行する。このコマンドは、ＵＳＢホストコントローラデバイスドライバ１０５を介してＵＳＢシリアル変換器２０へ送信される。ＵＳＢシリアル変換器２０では、コマンドを受信するとコマンド返信割込を発行する。コマンド返信割込は、ＵＳＢホストコントローラデバイスドライバ１０５を介してＵＳＢシリアル変換器デバイスドライバ１０３へ送信される。以上のプロセスにより、通信アプリケーション１０１とＵＳＢシリアル変換器２０との間でデータの送受信が開始される。

【００３４】

そして、コマンド返信割込を受信したＵＳＢシリアル変換器デバイスドライバ１０３は、所定周期でポーリングを開始する。具体的には、ＵＳＢシリアル変換器デバイスドライバ１０３は、ＵＳＢシリアル変換器２０のバッファにデータが蓄積されていることを確認するためのコマンドを発行する。このコマンドは、ＵＳＢホストコントローラデバイスドライバ１０５を介してＵＳＢシリアル変換器２０へ送信される。ＵＳＢシリアル変換器２０では、バッファにデータが蓄積しているか否かを確認し、確認結果を示すコマンド（コマンド返信割込）を発行する。コマンド返信割込は、ＵＳＢホストコントローラデバイスドライバ１０５を介してＵＳＢシリアル変換器デバイスドライバ１０３へ送信される。

【００３５】

なお、ＵＳＢシリアル変換器２０が、シリアル機器から送信されたデータを受信し、その受信したデータをバッファに蓄積しているときには、通信アプリケーション１０１がＲＥＡＤコマンドを発行するのを待って、その蓄積するデータを通信アプリケーション１０１へ向けて送信する。

【００３６】

なお、本実施形態では、ポーリングを行う周期の初期値が１６ｍｓｅｃに設定されており、以後この周期にてポーリングが実行される。

（２）ＲＥＡＤコマンド発行

通信アプリケーション１０１では、所定の周期でＲＥＡＤコマンドを発行するように設定されており、この周期にてＲＥＡＤコマンドを発行する。なお、ＲＥＡＤコマンドとは、ＵＳＢシリアル変換器２０のバッファに蓄積されたデータを読み出すためのコマンドであり、ＵＳＢシリアル変換器２０のバッファの空き容量が無くならないように設定された周期で発行される。なお、この周期については通信アプリケーション１０１によって設定される。

【００３７】

まず、通信アプリケーション１０１が、ＵＳＢシリアル変換器２０のバッファに蓄積されるデータを読み出すためにＲＥＡＤコマンドを発行すると、そのＲＥＡＤコマンドはシリアルＡＰＩ１０２を介してＵＳＢシリアル変換器デバイスドライバ１０３に送信される。ＵＳＢシリアル変換器デバイスドライバ１０３では、ＲＥＡＤコマンドを受信するとＲＥＡＤ返値を発行する。このＲＥＡＤ返値は、シリアルＡＰＩ１０２を介して通信アプリケーション１０１へ送信される。また、ＵＳＢシリアル変換器デバイスドライバ１０３では、ＵＳＢシリアル変換器２０に対してコマンド発行する。このコマンドは、ＵＳＢホストコントローラデバイスドライバ１０５を介してＵＳＢシリアル変換器２０へ送信される。ＵＳＢシリアル変換器２０では、コマンドを受信すると、データをバッファに蓄積しているときには、その蓄積するデータをＵＳＢホストコントローラデバイスドライバ１０５を介してＵＳＢシリアル変換器デバイスドライバ１０３へ送信する。ＵＳＢシリアル変換器デバイスドライバ１０３では、受信したデータをＵＳＢ規格に変換した後にシリアルＡＰＩ１０２などを介して通信アプリケーション１０１へ送信する。通信アプリケーション１０１では、受信したデータをバッファに一時保管し、そのデータを必要とするアプリケーションの指示によって処理される。

【００３８】

（３）ＷＲＩＴＥコマンド発行

他のアプリケーションからの指示によってシリアル機器にデータを送信する場合には、通信アプリケーション１０１が、ＵＳＢシリアル変換器２０のバッファにデータを書き込

むためにＷＲＩＴＥコマンドを発行する。すなわち、通信アプリケーション１０１が、ＵＳＢシリアル変換器２０のバッファにデータを書き込むためにＷＲＩＴＥコマンドを発行すると、そのＷＲＩＴＥコマンドはシリアルＡＰＩ１０２を介してＵＳＢシリアル変換器デバイスドライバ１０３に送信される。一方、ＵＳＢシリアル変換器デバイスドライバ１０３では、ＷＲＩＴＥコマンドを受信するとＷＲＩＴＥ返値を発行する。このＷＲＩＴＥ返値は、シリアルＡＰＩ１０２を介して通信アプリケーション１０１へ送信される。そして、通信アプリケーション１０１が、ＵＳＢシリアル変換器２０のバッファに書き込みたいデータをシリアルＡＰＩ１０２などを介してＵＳＢシリアル変換器２０へ送信する。ＵＳＢシリアル変換器２０では、受信したデータをバッファへ一旦保管した後にシリアル規格に変換してシリアル機器へ送信する。

【００３９】

（４）ＣＬＯＳＥコマンド発行

通信アプリケーション１０１とＵＳＢシリアル変換器２０との間でデータの送受信を終了する際には、通信アプリケーション１０１がＣＬＯＳＥコマンドを発行する。すなわち、通信アプリケーション１０１が、ＵＳＢシリアル変換器２０とのデータの送受信を終了するためにＣＬＯＳＥコマンドを発行すると、そのＣＬＯＳＥコマンドはシリアルＡＰＩ１０２を介してＵＳＢシリアル変換器デバイスドライバ１０３に送信される。一方、ＵＳＢシリアル変換器デバイスドライバ１０３では、ＣＬＯＳＥコマンドを受信すると、これまで定期的に行ってきたポーリングを停止するとともにＣＬＯＳＥ返値を発行する。このＣＬＯＳＥ返値は、シリアルＡＰＩ１０２を介して通信アプリケーション１０１へ送信される。

【００４０】

［ポーリング周期変更処理（１）の説明］

以下に、ＵＳＢシリアル変換器デバイスドライバ１０３により実行されるポーリング周期変更処理の処理（１）手順を図５のフローチャートおよび図６に基づいて説明する。なお、図６はポーリング周期変更処理（１）を説明するシーケンス図である。

【００４１】

このポーリング周期変更処理（１）は、ホスト機器１０が起動している場合において他の処理からは独立して繰り返し実行される。

まず、ＵＳＢシリアル変換器デバイスドライバ１０３によるコマンド発行であるドライバコールが開始されると、そのドライバコールの種別に応じてファンクション分岐が行われる。本実施形態では、（Ａ）ドライバコールの種別がＯＰＥＮコマンドである場合、（Ｂ）ドライバコールの種別がＣＬＯＳＥコマンドである場合、（Ｃ）ドライバコールの種別がＲＥＡＤコマンドである場合、（Ｄ）ドライバコールの種別がＷＲＩＴＥコマンドである場合、（Ｅ）ドライバコールの種別がボーレート設定コマンドである場合を説明する。

【００４２】

（Ａ）ドライバコールの種別がＯＰＥＮコマンドである場合

この処理では、設定内容を初期化するよう指示する初期化コマンドを発行する。初期化コマンドが発行されると、タイマ処理の設定を初期化する。本実施形態ではポーリング周期が１６ｍｓｅｃに設定される。そして、ドライバコールを終了する。

【００４３】

なお、ホスト機器１０が搭載するＯＳの機能の一つであるタイマ起動処理が実行されると、上述のタイマ処理が開始され、本実施形態では２５０ｍｓｅｃごとに上述のようなＲＥＡＤコマンドを発行する。そして、タイマ処理を終了する。

【００４４】

（Ｂ）ドライバコールの種別がＣＬＯＳＥコマンドである場合

この処理では、タイマ処理の設定を解除する。このことにより、タイマ処理が実行されなくなり、以後ＲＥＡＤコマンドが発行されなくなる。また、上述のようなＣＬＯＳＥコマンドの発行によってポーリングも停止する。そして、ドライバコールを終了する。

【００４５】
（Ｃ）ドライバコールの種別がＲＥＡＤコマンドである場合
この処理では、ＲＥＡＤデータの出力処理を実行する。具体的には、ＵＳＢシリアル変換器デバイスドライバ１０３内のバッファからデータを読み出す。読み出したデータについては、そのデータを必要とするアプリケーションの指示によって処理される。さらに、タイマ処理の再設定を行う。具体的には、処理を実行する度に内蔵するメモリ（図示省略）に記憶された処理時刻記録を参照して、前回の処理時刻から今回の処理時刻までの時間を計算した計算結果を新しいポーリング周期として設定する。一例を挙げると、前回ＲＥＡＤコマンドを発行した時刻から今回ＲＥＡＤコマンドを発行した時刻までの時間が２５０ｍｓｅｃである場合には、ポーリング周期を２５０ｍｓｅｃに設定すると云った具合である。そして、ドライバコールを終了する。
【００４６】
なお、上述のＵＳＢシリアル変換器デバイスドライバ１０３内のバッファに保管されるデータは、上述のようなＲＥＡＤデータ出力処理にて読み出される一方、コマンド完了割り込み処理が開始された際にも読み出される。具体的には、コマンド完了割り込み処理が開始されると、ＵＳＢシリアル変換器デバイスドライバ１０３内のバッファに保管されるデータが読み出され、その読み出されたデータはＵＳＢシリアル変換器２０へ転送される。ＵＳＢシリアル変換器２０では、転送されたデータをバッファに保管する。なお、このバッファに保管されたデータは、ＵＳＢ規格からシリアル規格へ変換された後にシリアル機器へ送信される。コマンド完了割り込み処理が終了する。
【００４７】
（Ｄ）ドライバコールの種別がＷＲＩＴＥコマンドである場合
この処理では、ＷＲＩＴＥデータの入力処理を実行する。具体的には、ＵＳＢシリアル変換器デバイスドライバ１０３内のバッファにデータが書き込まれるとともに、ＵＳＢシリアル変換器２０のバッファに書き込まれる。そして、ＷＲＩＴＥコマンドを発行し、上述のようにタイマ処理の再設定を行う。そして、ドライバコールを終了する。
【００４８】
（Ｅ）ドライバコールの種別がボーレート設定コマンドである場合
この処理では、ボーレート設定コマンドを発行する。なお、ボーレート（ｂｐｓ：Ｂｉｔｓ　Ｐｅｒ　Ｓｅｃｏｎｄ）とは、ＲＳ２３２Ｃのシリアル通信速度であり、シリアル変換器３０との間でデータ転送を行う際の単位時間当たりのデータ転送量である。このボーレートについては通信アプリケーション１０１によって設定される。そして、ドライバコールを終了する。
【００４９】
［第一実施形態の効果］
（１）このように第一実施形態のホスト機器１０によれば、ＲＥＡＤコマンドを発行する間隔を計算した計算結果を新しいポーリング周期として設定する。このように計算したポーリング周期は、ＵＳＢシリアル変換器２０のバッファの容量およびＵＳＢシリアル変換器２０との間でデータ転送を行う際の単位時間当たりの最大データ転送量に応じて予め設定されたポーリング周期の初期値よりも長いために、変更後のポーリング周期が変更前に比べて長くなる。なお、ＵＳＢシリアル変換器２０のバッファに蓄積されたデータについては、通信アプリケーション１０１がＲＥＡＤコマンドを発行する度に、ＵＳＢシリアル変換器２０からホスト機器１０にデータ転送されるので、ＵＳＢシリアル変換器２０のバッファの空き容量が無くなることはない。したがって、ＵＳＢシリアル変換器２０に蓄積されたデータを速やかに読み出しながら、ホスト機器１０の処理負担を軽減することができる。
【００５０】
［他の実施形態］
以上、本発明の一実施形態について説明したが、本発明は上記実施形態に限定されるものではなく、以下のような様々な態様にて実施することが可能である。

【００５１】
（１）上記実施形態では、ＲＥＡＤコマンドを発行する間隔を計算した計算結果を新しいポーリング周期として設定するが、これには限られず、例えば、通信アプリケーション１０１によって設定したボーレートに対応する周期を新しいポーリング周期として設定してもよい。なお、図７はポーリング周期変更処理（２）を示すフローチャートであり、図８はポーリング周期変更処理（２）を説明するシーケンス図である。具体的には、図７および図８に例示するように、ドライバコールの種別がボーレート設定である場合にボーレートコマンドを発行した後に、タイマ処理の再設定を行う。この際、通信アプリケーション１０１によって設定したボーレートに対応する周期を新しいポーリング周期として設定する。なおこの場合、ドライバコールの種別がＲＥＡＤコマンドである場合にＲＥＡＤデータの出力処理を実行した後、および、ドライバコールの種別がＷＲＩＴＥコマンドである場合にＷＲＩＴＥコマンドを発行した後に実行する、タイマ処理の再設定でＲＥＡＤコマンドを発行する間隔を計算する処理を行わない。このように計算したポーリング周期は、上述のように予め設定されたポーリング周期の初期値よりも長いために、変更後のポーリング周期が変更前に比べて長くなる。なお、ＵＳＢシリアル変換器２０のバッファに蓄積されたデータについては、上述のように、通信アプリケーション１０１がＲＥＡＤコマンドを発行する度に、ＵＳＢシリアル変換器２０からホスト機器１０にデータ転送されるので、ＵＳＢシリアル変換器２０のバッファの空き容量が無くなることはない。したがって、ＵＳＢシリアル変換器２０に蓄積されたデータを速やかに読み出しながら、ホスト機器１０の処理負担を軽減することができる。

【００５２】
なお、前述した、タイマ処理再設定による、通信アプリケーション１０１によって設定したボーレートに対応する周期を新しいポーリング周期の具体的は数値（図７、ポーリング周期Ｘ（ｍｓ））設定の一例を以下に説明する。ここで、「ＳＥＴ」は、通信アプリケーション１０１が設定する通信速度（ボーレート：ｂｐｓ）である。通信アプリケーション１０１が送付するデータ（有効データ）８ｂｉｔに対して、ＲＳ２３２Ｃで転送するデータ（転送データ）は、前記有効データに加えＲＳ２３２Ｃスタートビット（ＳＴＡＲＴ－ｂｉｔ）とストップビット（ＳＴＯＰ－ｂｉｔ）とが加わるので、有効データ８ｂｉｔ（１ｂｙｔｅ）を送付するときのＲＳ２３２Ｃ転送データは（８ｂｉｔ＋１ｂｉｔ＋１ｂｉｔ）となる。なお、本実施形態におけるＵＳＢシリアル変換器２０(例えば、ＦＴ２３２Ｒ）のバッファ容量は２５６（ｂｙｔｅ）である。よって、ポーリング周期Ｘは、Ｘ（ｓｅｃ）＝１÷（ＳＥＴ÷（８＋１＋１）÷２５６）となり、秒からミリ秒へ単位変換すれば、Ｘ＊１０００＝１÷（ＳＥＴｂｐｓ÷（８＋１＋１）÷２５６）＝Ｘ（ｍｓ）となる。

【００５３】
（２）また、ＯＰＥＮコマンドを発行後に、ポーリングを停止するようにしてもよい。なお、図９はポーリング周期変更処理（３）を説明するシーケンス図である。具体的には、図９に例示するように、入力部としてのホスト機器１０のキーボードなどに対するユーザからの指示や他のアプリケーションからの指示により、通信アプリケーション１０１が、ポーリングを停止するためにマニュアルトリガモードに変更する旨のモード変更コマンドを発行する。すなわち、通信アプリケーション１０１が、モード変更コマンドを発行すると、そのモード変更コマンドはシリアルＡＰＩ１０２を介してＵＳＢシリアル変換器デバイスドライバ１０３に送信される。一方、ＵＳＢシリアル変換器デバイスドライバ１０３では、そのモード変更コマンドを受信すると、これまで定期的に行ってきたポーリングを停止するとともにモード変更返値を発行する。このモード変更返値は、シリアルＡＰＩ１０２を介して通信アプリケーション１０１へ送信される。なお、ＵＳＢシリアル変換器２０のバッファに蓄積されたデータについては、上述のように、通信アプリケーション１０１がＲＥＡＤコマンドを発行する度に、ＵＳＢシリアル変換器２０からホスト機器１０にデータ転送されるので、ＵＳＢシリアル変換器２０のバッファの空き容量が無くなることはない。したがって、ＵＳＢシリアル変換器２０に蓄積されたデータを速やかに読み出

しながら、ホスト機器１０の処理負担を軽減することができる。

【図面の簡単な説明】

【００５４】

【図１】ＵＳＢシリアル変換器を用いてホスト機器とシリアル機器とを接続する例を示す説明図である。

【図２】ホスト機器およびＵＳＢシリアル変換器の構成を示すハードブロック図である。

【図３】ホスト機器の構成を示すソフトブロック図である。

【図４】従来のホスト機器とシリアル機器とのデータ送受信プロセスを説明するシーケンス図である。

【図５】ポーリング周期変更処理（１）を示すフローチャートである。

【図６】ポーリング周期変更処理（１）を説明するシーケンス図である。

【図７】ポーリング周期変更処理（２）を示すフローチャートである。

【図８】ポーリング周期変更処理（２）を説明するシーケンス図である。

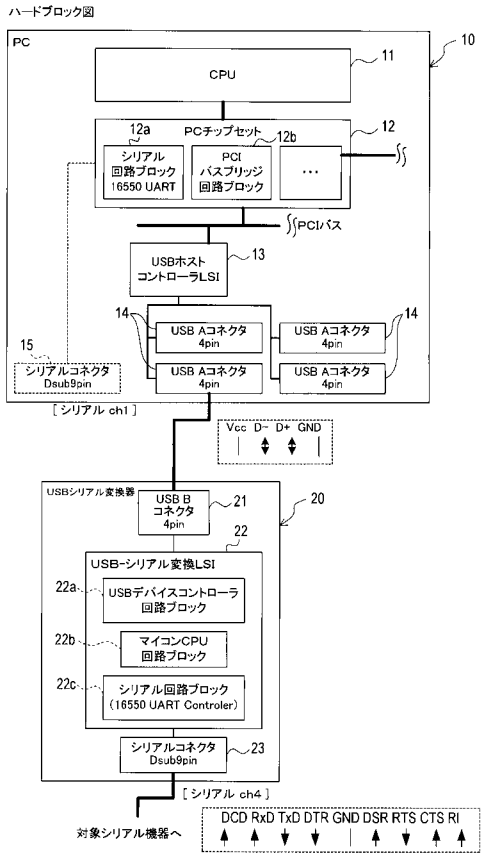
【図９】ポーリング周期変更処理（３）を説明するシーケンス図である。

【符号の説明】

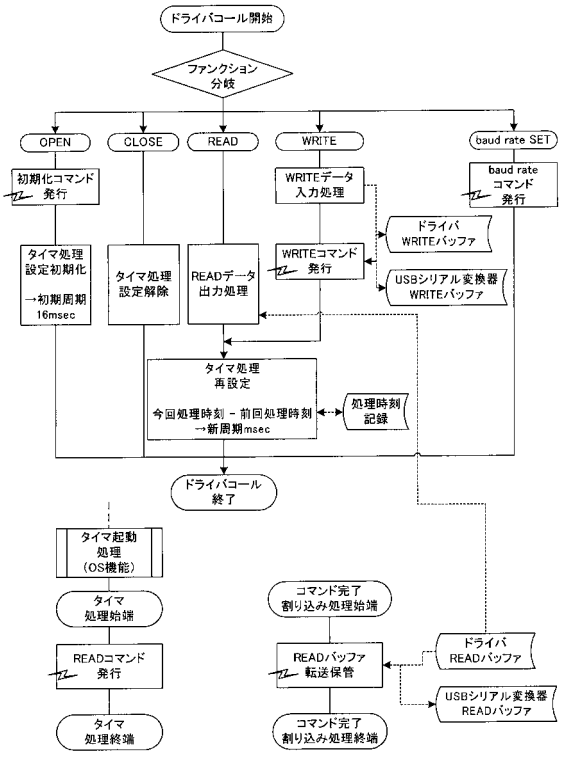
【００５５】

１０…ホスト機器、１１…ＣＰＵ、１２…ＰＣチップセット、１２ａ…シリアル回路ブロック、１２ｂ…ＰＣＩバスブリッジ回路ブロック、１３…ＵＳＢホストコントローラＬＳＩ、１４…ＵＳＢ（Ａ）コネクタ、１５…シリアルコネクタ、２０…ＵＳＢシリアル変換器、２１…ＵＳＢ（Ｂ）コネクタ、２２…ＵＳＢシリアル変換ＬＳＩ、２２ａ…ＵＳＢデバイスコントローラ回路ブロック、２２ｂ…マイコンＣＰＵ回路ブロック、２２ｃ…シリアル回路ブロック、２３…シリアルコネクタ、３０…シリアル機器、１０１…通信アプリケーション、１０２…シリアルＡＰＩ、１０３…ＵＳＢシリアル変換器デバイスドライバ、１０４…シリアルデバイスドライバ、１０５…ＵＳＢホストコントローラデバイスドライバ

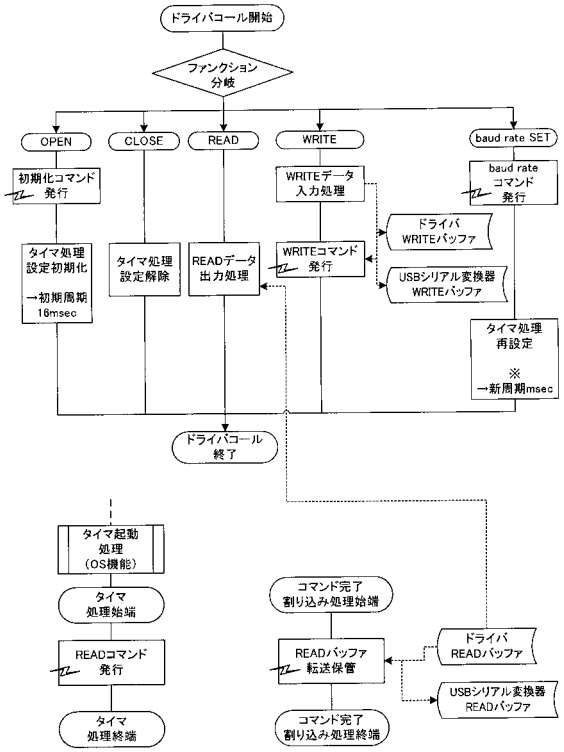
【図２】



【図５】



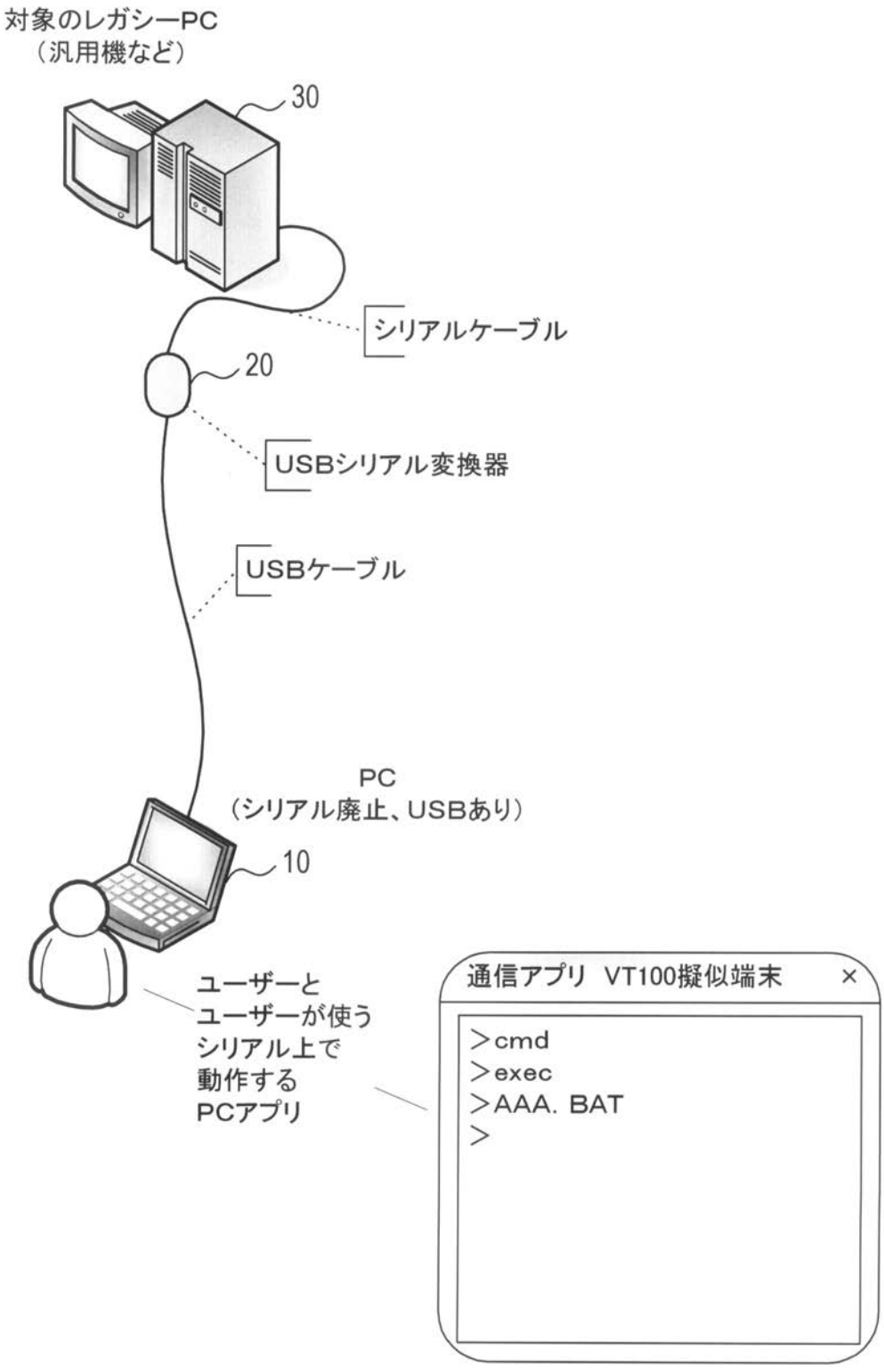
【図７】



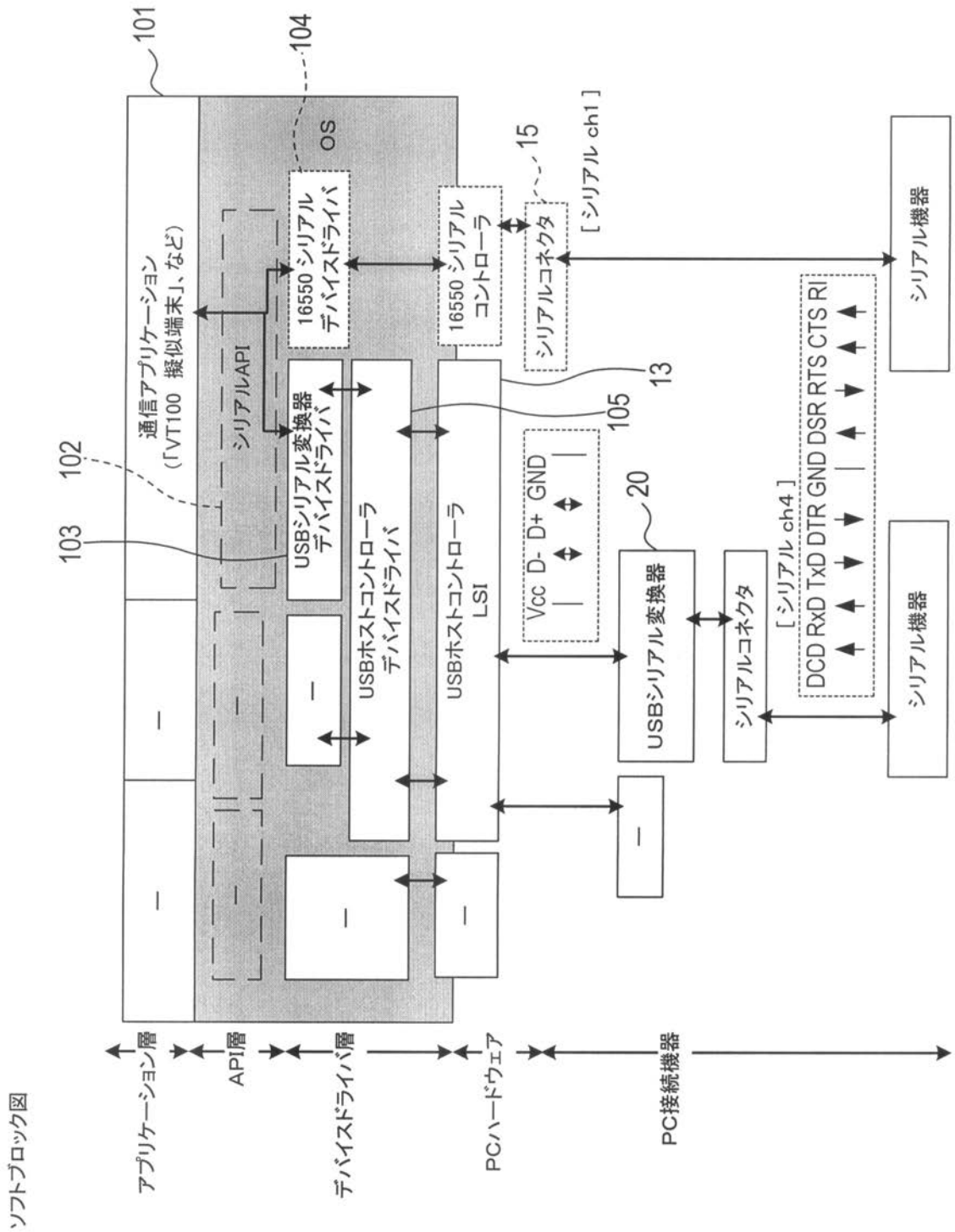
※ 1sec / { SETbps / 8bit(1byte)+2bit(START-bitとSTOP-bit) / 256byte } = Xsec × 1000 = Xmsec

【図１】

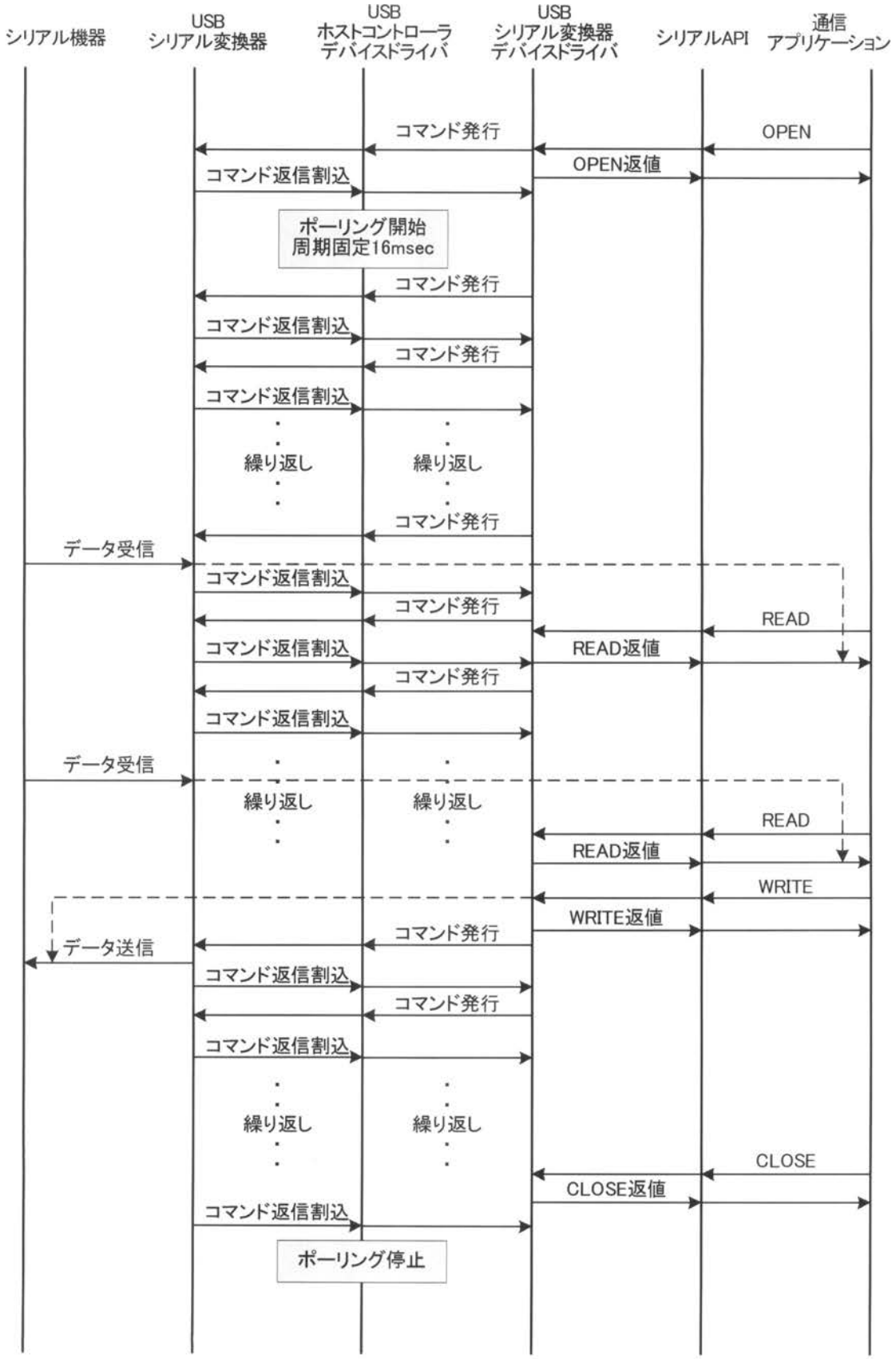
全体構成


対象のレガシーPC
（汎用機など）
30
シリアルケーブル
20
USBシリアル変換器
USBケーブル
PC
（シリアル廃止、USBあり）
10
ユーザーと
ユーザーが使う
シリアル上で
動作する
PCアプリ
通信アプリ VT100擬似端末
×
>cmd
>exec
>AAA. BAT
>

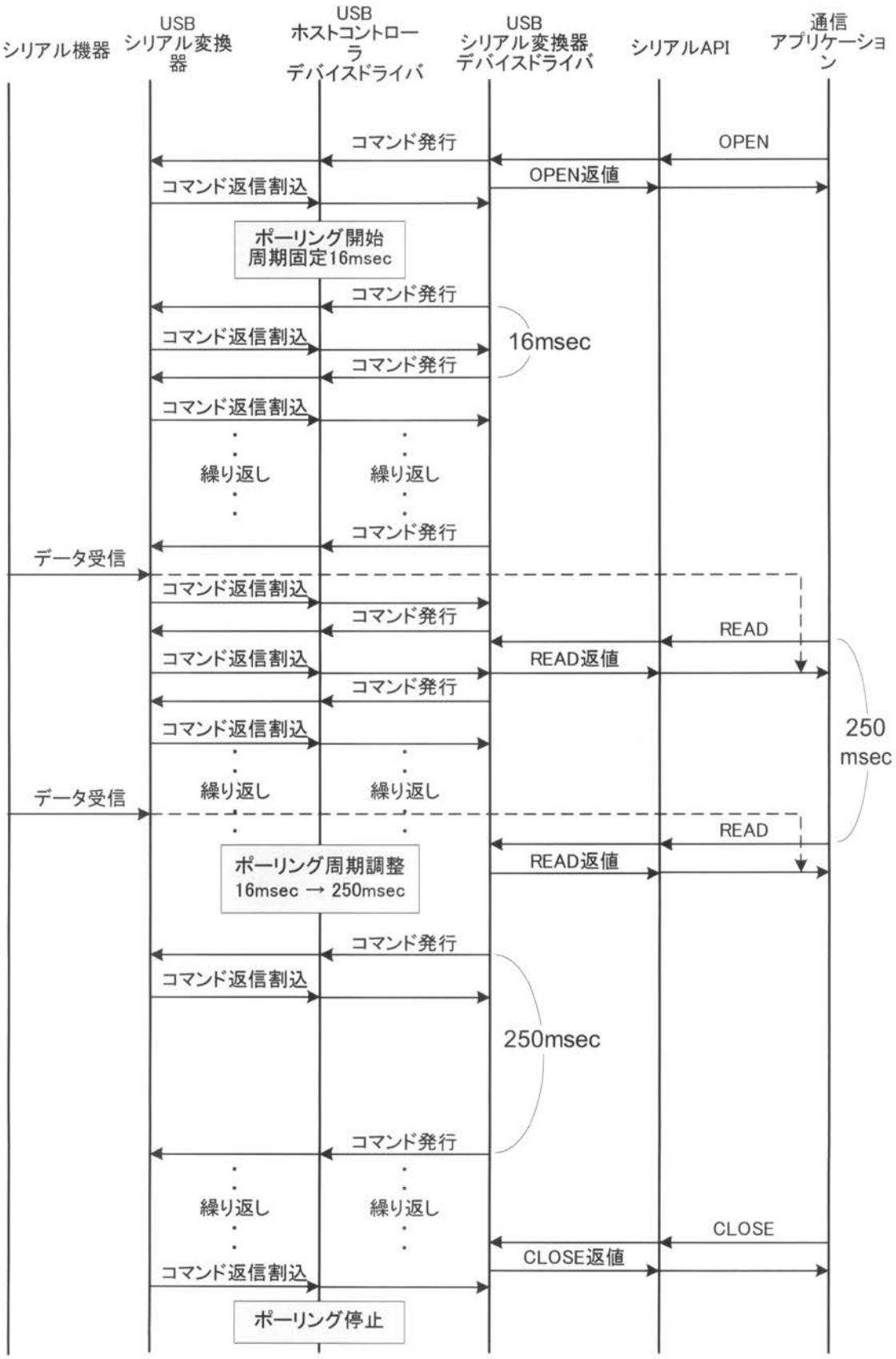
【図３】


ソフトブロック図
アプリケーション層
API層
デバイスドライバ層
PCハードウェア
PC接続機器
101
102
103
104
105
13
15
20
通信アプリケーション
（「VT100 擬似端末」、など）
シリアルAPI
OS
16550 シリアル
デバイスドライバ
USBシリアル変換器
デバイスドライバ
USBホストコントローラ
デバイスドライバ
USBホストコントローラ
LSI
16550 シリアル
コントローラ
シリアルコネクタ
[シリアル ch1]
Vcc D- D+ GND
USBシリアル変換器
シリアルコネクタ
[シリアル ch4]
DCD RxD TxD DTR GND DSR RTS CTS RI
シリアル機器
シリアル機器

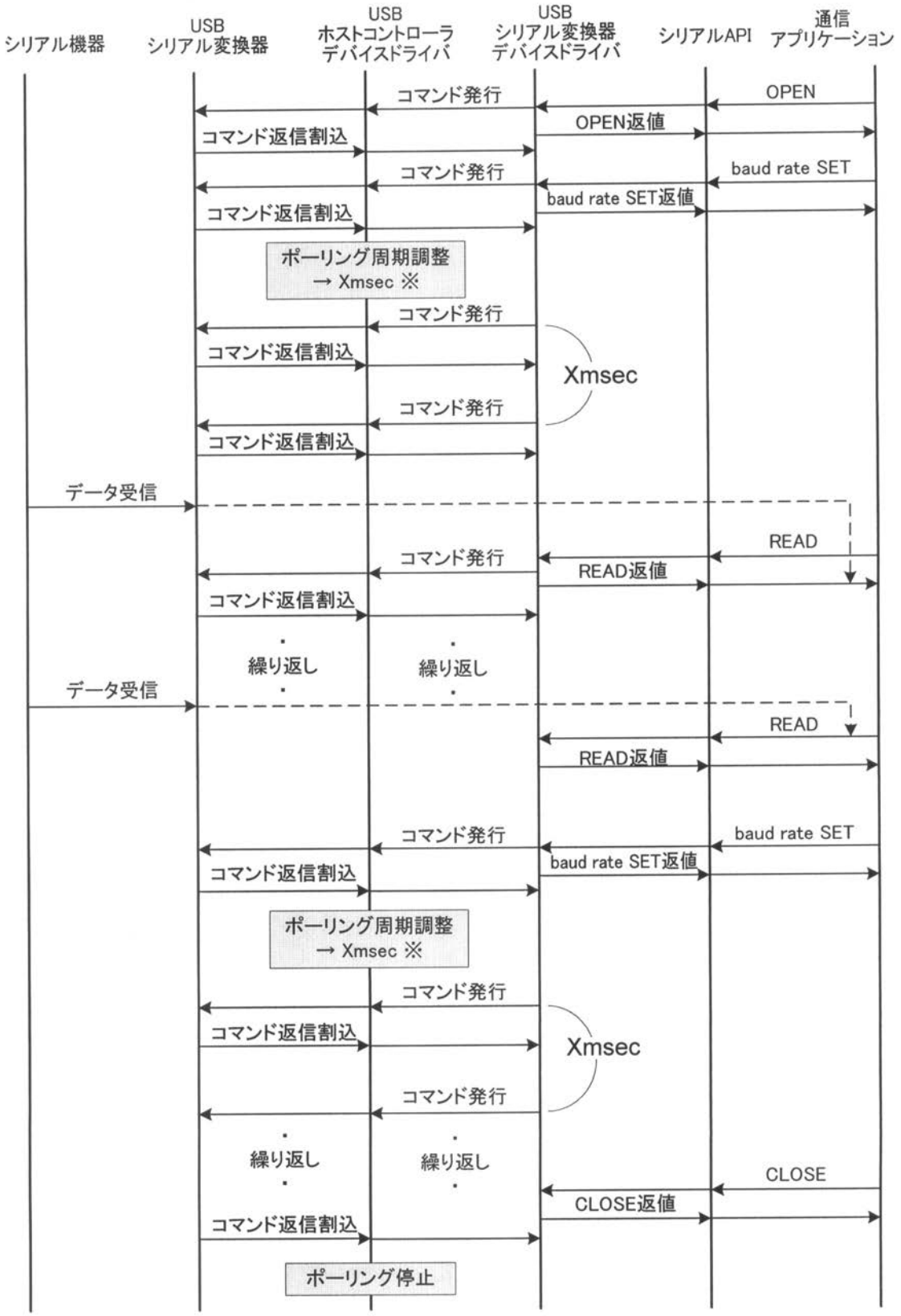
【図４】


シリアル機器
USB
シリアル変換器
USB
ホストコントローラ
デバイスドライバ
USB
シリアル変換器
デバイスドライバ
シリアルAPI
通信
アプリケーション
コマンド発行
OPEN
OPEN返値
コマンド返信割込
ポーリング開始
周期固定16msec
コマンド発行
コマンド返信割込
コマンド発行
コマンド返信割込
繰り返し
繰り返し
コマンド発行
データ受信
コマンド返信割込
コマンド発行
READ
コマンド返信割込
READ返値
コマンド発行
コマンド返信割込
データ受信
繰り返し
繰り返し
READ
READ返値
WRITE
WRITE返値
コマンド発行
データ送信
コマンド返信割込
コマンド発行
コマンド返信割込
繰り返し
繰り返し
CLOSE
CLOSE返値
コマンド返信割込
ポーリング停止

【図６】

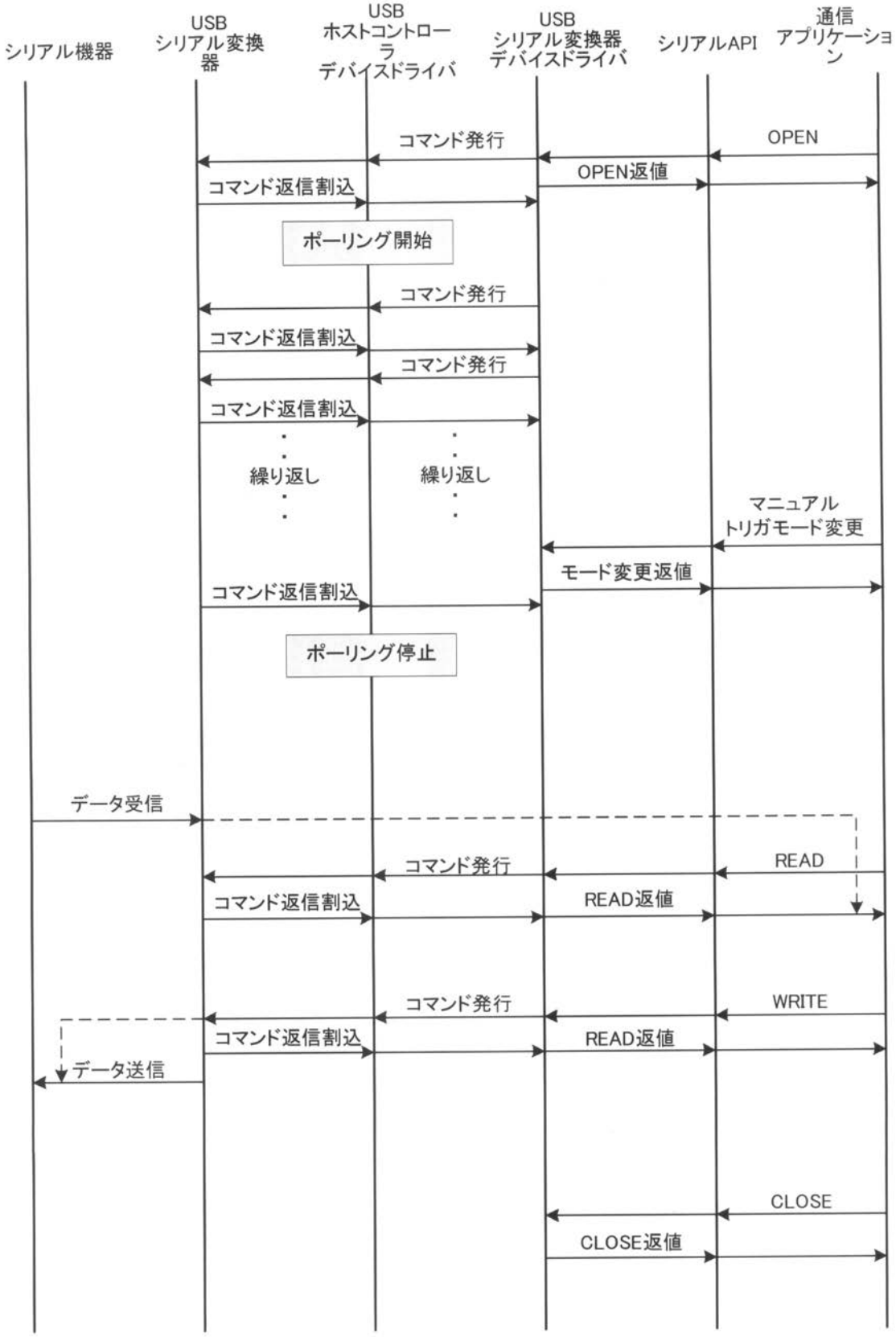

シリアル機器
USB
シリアル変換
器
USB
ホストコントロー
ラ
デバイスドライバ
USB
シリアル変換器
デバイスドライバ
シリアルAPI
通信
アプリケーショ
ン
コマンド発行
OPEN
OPEN返値
コマンド返信割込
ポーリング開始
周期固定16msec
コマンド発行
16msec
コマンド返信割込
コマンド発行
コマンド返信割込
繰り返し
繰り返し
コマンド発行
データ受信
コマンド返信割込
コマンド発行
READ
コマンド返信割込
READ返値
コマンド発行
コマンド返信割込
250
msec
繰り返し
繰り返し
データ受信
READ
ポーリング周期調整
16msec → 250msec
READ返値
コマンド発行
コマンド返信割込
250msec
コマンド発行
繰り返し
繰り返し
CLOSE
CLOSE返値
コマンド返信割込
ポーリング停止

【図８】



※ 1sec / (SETbps / 8bit(1byte)+2bit(START-bitとSTOP-bit) / 256byte) = Xsec × 1000 = Xmsec

【図９】


シリアル機器
USB
シリアル変換
器
USB
ホストコントロー
ラ
デバイスドライバ
USB
シリアル変換器
デバイスドライバ
シリアルAPI
通信
アプリケーショ
ン
コマンド発行
OPEN
OPEN返値
コマンド返信割込
ポーリング開始
コマンド発行
コマンド返信割込
コマンド発行
コマンド返信割込
繰り返し
繰り返し
マニュアル
トリガモード変更
モード変更返値
コマンド返信割込
ポーリング停止
データ受信
コマンド発行
READ
コマンド返信割込
READ返値
コマンド発行
WRITE
コマンド返信割込
READ返値
データ送信
CLOSE
CLOSE返値

フロントページの続き

(56)参考文献 特開２００２－１９７０４８（ＪＰ，Ａ）
特開２００１－０１６２９４（ＪＰ，Ａ）
特開２００１－０８６１４１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１３／１０－４２