

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4422359号
(P4422359)

(45) 発行日 平成22年2月24日(2010.2.24)

(24) 登録日 平成21年12月11日(2009.12.11)

(51) Int.Cl.

F I

H O 4 L 12/40 (2006.01)

H O 4 L 12/40 Z

G O 6 F 13/12 (2006.01)

G O 6 F 13/12 3 3 O A

請求項の数 3 (全 19 頁)

(21) 出願番号 特願2001-108265 (P2001-108265)
 (22) 出願日 平成13年4月6日(2001.4.6)
 (65) 公開番号 特開2001-358733 (P2001-358733A)
 (43) 公開日 平成13年12月26日(2001.12.26)
 審査請求日 平成18年10月13日(2006.10.13)
 (31) 優先権主張番号 09/549124
 (32) 優先日 平成12年4月13日(2000.4.13)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 398038580
 ヒューレット・パカード・カンパニー
 HEWLETT-PACKARD COMPANY
 アメリカ合衆国カリフォルニア州パロアル
 ト ハノーバー・ストリート 3000
 (74) 代理人 110000246
 特許業務法人オカダ・フシミ・ヒラノ
 (74) 代理人 100081721
 弁理士 岡田 次生
 (74) 代理人 100105393
 弁理士 伏見 直哉
 (74) 代理人 100111969
 弁理士 平野 ゆかり

最終頁に続く

(54) 【発明の名称】ユニバーサルシリアルバス (USB) をピアツーピアネットワークとして用いる方法

(57) 【特許請求の範囲】

【請求項 1】

ユニバーサルシリアルバス (USB) をピアツーピアネットワークとして用いる方法であって、

ルータを備えるホストプロセッサをUSBに接続するステップと、

それぞれ互いに対して、かつ前記ホストプロセッサに対してピアである複数のデバイスを前記USBに接続するステップと、

前記複数のデバイスまたは前記ホストプロセッサのうちの1つにおいて、メッセージを形成するステップと、

前記メッセージの前にヘッダを付加するステップと、

前記ルータおよび前記USBを介して、前記メッセージを前記デバイスのうちの1つから他の任意のデバイスに、または前記ホストプロセッサに直接伝送するステップと、を含み、

前記ヘッダは、前記デバイスの各々を識別する複数のフィールドを含み、前記デバイスはそれぞれ、前記ヘッダを付加し、前記複数のフィールドを認識するよう構成されたロジックを備える、方法。

【請求項 2】

ユニバーサルシリアルバス (USB) をピアツーピアネットワークとして用いるシステムであって、

ホストメッセージハンドラを含み、メッセージの前にヘッダを付加するよう構成され、

10

20

該ヘッダは、前記デバイスの各々を識別する複数のフィールドを含み、該ホストメッセージハンドラは、該ヘッダを認識するよう構成された、ホストプロセッサと、

前記ホストプロセッサに連係され、それぞれデバイスメッセージハンドラを含み、前記メッセージの前に前記ヘッダを付加するよう構成された複数のデバイスであって、前記デバイスメッセージハンドラも、前記ヘッダを認識するよう構成された、複数のデバイスと、

前記ホストプロセッサに連係された前記複数のデバイスのうちの任意のデバイスと前記ホストプロセッサ間で、ユニバーサルシリアルバス（USB）を介して前記メッセージを直接ルーティングするよう構成された、前記ホストプロセッサに連係されたルータと、を備える、システム。

10

【請求項3】

ユニバーサルシリアルバス（USB）をピアツーピアネットワークとして用いるためのプログラムを記録したコンピュータ読み取り可能媒体であって、

前記プログラムは、

ルータを備えるホストプロセッサをUSBに接続するステップと、

それぞれ互いに対して、かつ前記ホストプロセッサに対してピアである複数のデバイスを前記USBに接続するステップと、

前記複数のデバイスのうちの1つにおいて、メッセージを形成するステップと、

前記メッセージの前にヘッダを付加するステップと、

前記ルータおよび前記USBを介して、前記メッセージを前記デバイスのうちの1つから他の任意のデバイスに、または前記ホストプロセッサに直接伝送するステップと、を実現するよう構成され、

20

前記ヘッダは、前記デバイスの各々を識別する複数のフィールドを含み、前記デバイスはそれぞれ、前記ヘッダを付加し、前記複数のフィールドを認識するよう構成されたロジックを備える、コンピュータ読み取り可能媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、概して、コンピュータネットワーキングに関し、特に、ユニバーサルシリアルバスをピアツーピアネットワークとして用いる方法およびシステムに関する。

30

【0002】

【従来の技術】

ピアツーピアネットワークは、各コンピュータがネットワーク上の任意の他のコンピュータへのメッセージ送信を開始でき、またネットワーク上の任意の他のコンピュータからのメッセージ送信を受信することが可能なように、電子ネットワークで接続された複数のコンピュータと定義される。ローカルエリアネットワーク（LAN）は、ピアツーピアネットワークの一例である。

【0003】

ユニバーサルシリアルバス（USB）は、一般に、パケットまたはメッセージの交換によってホストからデバイスへの通信を可能にするコンピューティングデバイスに関連する通信バスアーキテクチャである。USBメッセージは、通常、ヘッダ情報と、データとを含む。例えば、パーソナルコンピュータについていえば、コンピュータプロセッサをホストと考え、マウス、キーボード、プリンタ、ジョイスティック、ディスクドライブ等をホストと通信するデバイスと考えることができる。従来のUSBプロトコルでは、すべての通信メッセージがUSBホストと該ホストに取り付けられたデバイス間で転送されるよう条件付けられている。統合されたUSBデバイスインタフェースを含む、競争力のある価格が付けられたマイクロプロセッサが利用しやすいために、USBアーキテクチャを採用することは経済的である。

40

【0004】

従来のUSBプロトコルは、また、広い帯域幅、およびプロセッサモジュールを「ホット

50

スワップ(hotswap)」する能力が重要な通信に対しても有用である。「ホットスワップ」という用語は、システムの電源を切ることなく、USBデバイスを動作中のコンピュータシステムからはずし、別のプロセッサと取り替える状況を指す。

【0005】

【発明が解決しようとする課題】

不都合なことに、この従来のプロトコルでは、USBに接続されるデバイス間の通信がすべてホスト-デバイス間通信を介して行われるよう要請されるために、「ピアツーピア」通信とも呼ばれるデバイス間の直接通信を行うことができない。

【0006】

したがって、システムの電源を切ることなく、プロセッサモジュールの取り外しおよび置換が可能であり、かつ通信バスに接続されたデバイス間でのピアツーピア通信が可能な高帯域幅通信プロトコルおよびバスが必要とされている。

【0007】

【課題を解決するための手段】

本発明は、USBに接続されたデバイス間でのピアツーピア通信のための方法およびシステムを提供する。

【0008】

本発明は、USBをピアツーピアネットワークとして用いる方法として概念化することができ、本方法は、ルータを備えるホストプロセッサをUSBに接続するステップと、それぞれ互いに対して、かつホストプロセッサに対してピアである複数のデバイスをUSBに接続するステップと、を含む。本方法はまた、複数のデバイスまたはホストプロセッサのうちの1つにおいて、メッセージを形成するステップと、該メッセージの前にヘッダを付加するステップと、ルータおよびUSBを介して、該メッセージをデバイスのうちの1つから他の任意のデバイス、またはホストプロセッサに直接伝送するステップと、含む。

【0009】

アーキテクチャにおいて、本発明は、USBをピアツーピアネットワークとして用いるためのシステムであり、メッセージハンドラを含むホストプロセッサを備える。ホストプロセッサは、メッセージの前にヘッダを付加するよう構成され、ホストメッセージハンドラは、該ヘッダを認識するよう構成される。本システムはまた、それぞれデバイスメッセージハンドラを含む、ホストプロセッサに連係された複数のデバイスも備える。複数のデバイスはそれぞれ、メッセージの前にヘッダを付加するよう構成される。デバイスメッセージハンドラはまた、ヘッダを認識するよう構成される。本システムは、ホストプロセッサと、ホストプロセッサに連係された複数のデバイスのうちの任意のデバイス間で直接、USBを介してメッセージをルーティングするよう構成された、ホストプロセッサに連係されたルータも備える。

【0010】

特許請求の範囲において定義付けられる本発明について、添付図面を参照することでより良好に理解することができる。図面内の構成要素は、必ずしも互いに対して一定の縮尺ではなく、本発明の原理を明白に示す部分を強調している。

【0011】

【発明の実施の形態】

USBをピアツーピアネットワークとして用いる方法およびシステムの好ましい実施形態について、サーバコンピュータシステムを制御するために、USBを用いて互いに対話する多数のマイクロプロセッサ駆動式デバイスを例として説明するが、本発明は、多数のデバイスがUSBを介して通信するあらゆる状況において適用可能である。

【0012】

USBをピアツーピアネットワークとして用いる方法およびシステムは、ハードウェア、ソフトウェア、ファームウェア、またはそれらの組み合わせにおいて実施することができる。好ましい実施形態において、本発明は、ハードウェアと、ソフトウェアまたはファームウェアとの組み合わせにおいて実施される。ハードウェアまたはファームウェアはメモ

10

20

30

40

50

りに格納し、適した命令実行システムで実行することができる。代替の実施形態におけるように、ハードウェアにおいて実施される場合、本発明は当技術分野で周知の次の技術、すなわち、データ信号に対して論理関数処理を行う論理ゲートを有する離散論理回路、適切な組み合わせ論理ゲートを備えた特定用途向け集積回路（ASIC）、プログラマブル・ゲートアレイ（PGA）、フィールド・プログラマブル・ゲートアレイ（FPGA）等のいずれか一つ、あるいはそれらの組み合わせを用いて実施することができる。

【0013】

USBをピアツーピアネットワークとして用いる方法およびシステムのソフトウェア部分は、論理関数処理を行うための実行可能命令の順序付きリストを含み、コンピュータベースのシステム、プロセッサ包含システム、または媒体から命令をフェッチして実行することのできる他のシステム等のような、命令実行システム、装置、またはデバイスによって、またはこれらと併せて用いるため、任意のコンピュータ読み取り可能媒体において実施することができる。本明細書において、「コンピュータ読み取り可能媒体」とは、命令実行システム、装置、またはデバイスによって、またはこれらと併せて用いるため、プログラムを包含、格納、通信、伝搬、または搬送することが可能な任意の手段でありうる。コンピュータ読み取り可能媒体は、例えば、電気、磁気、光学、電磁、赤外線、または半導体のシステム、装置、デバイス、または伝搬媒体でありうるが、これらに制限されない。コンピュータ読み取り可能媒体のより具体的な例（非網羅的リスト）としては、以下が挙げられる。すなわち、1本または複数本のワイヤを有する電気接続（電気）、ポータブルコンピュータディスク（磁気）、ランダムアクセスメモリ（RAM）（電気）、読み取り専用メモリ（ROM）（電気）、消去可能プログラマブル読み取り専用メモリ（EPROMまたはフラッシュメモリ）（電気）、光ファイバ（光学）、およびポータブルコンパクトディスク読み取り専用メモリ（CD-ROM）（光学）である。なお、コンピュータ読み取り可能媒体は、紙またはプログラムを印刷可能な別の適した媒体であってもよいことに留意する。これは、例えば光学スキャン等により紙または他の媒体上のプログラムを電子的に取り込んでから、必要であればコンパイル、解釈するか、そうでなければ適宜処理し、コンピュータメモリに格納することができるためである。

【0014】

次に、図面に目を向けて、図1は、本発明の概念を示すために用いられる、サーバシステム11を示すブロック図である。なお、サーバシステム11として図示されるが、USBをピアツーピアネットワークとして用いる方法およびシステムは、それぞれUSBインタフェースを有する複数のプロセッサデバイスがピアツーピア構成でUSBを介して通信する任意のシステムに適用可能であることに留意されたい。サーバシステム11は、多数のキャビネットに分割してもよい。例えば、説明のために、サーバ11は、時にキャビネット27および28と呼ばれるノードに分割される。キャビネット27は、メモリ要素12と、プロセッサ14とを備える。キャビネット28は、メモリ要素36と、プロセッサ37とを備える。各キャビネット27および28はまた、メモリ要素12および36と、プロセッサ14および37をそれぞれ制御し、監視するユーティリティサブシステム50および30もそれぞれ備える。

【0015】

ユーティリティサブシステム50は、USB20に接続され、USB20を介して通信する多数のプロセッサデバイスを備え、本明細書では、キャビネットレベルユーティリティ（CLU）デバイス60と、パワーモニタ3（PM3）デバイス70と、システム抽象レイヤインタフェースネットワークコントローラ（SINC）デバイス100と、サービスプロセッサ200と、を備えるものとして図示される。通常の適用においては、各ユーティリティサブシステム50および30は、CLUデバイス、PM3デバイス、およびSINCデバイスの組を繰り返し多数含むと共に、追加のプロセッサおよび制御デバイスを備える。そのため、複数のユーティリティサブシステム50および30が図示されている。しかし、簡略化のために、それぞれCLUデバイス、PM3デバイス、SINCデバイスを一組ずつ備える二組のユーティリティサブシステムのみが図示されている。ユーティリ

10

20

30

40

50

ティサブシステム 50 の場合には、それらに加えてサービスプロセッサ 200 を備える。しかし、サーバ 11 に連係されるサービスプロセッサ 200 は（ユーティリティサブシステム 50 に配置される）一つのみであることに留意されたい。本発明の一形態においては、サービスプロセッサ 200 はホストプロセッサとして機能する。USB プロトコルは一つのホストしか許容しないため、ユーティリティプロセッサおよび関連デバイスの数に関係なく、サービスプロセッサ 200 は一つだけである。本発明の一形態において重要なことは、各ユーティリティサブシステム 50 および 30 内の各デバイスが USB 2.0 を介して通信するということである。

【0016】

SINC デバイス 100 は、接続 21 を介して、メモリ 12 およびプロセッサ 14 と通信し、これらを制御する。CLU デバイス 60 は、接続 22 を介して、バックプレーン要素 16 と通信してこれを制御し、PM3 デバイス 70 は、接続 24 を介して、電源およびファンモジュール 18 と通信し、これらを制御する。同様に、SINC デバイス 31 は、接続 41 を介して、メモリ 36 およびプロセッサ 37 と通信し、これらを制御する。CLU デバイス 32 は、接続 42 を介して、バックプレーン要素 38 と通信してこれを制御し、PM3 デバイス 34 は、接続 44 を介して、電源およびファンモジュール 39 と通信し、これらを制御する。

【0017】

図 2 は、図 1 のユーティリティサブシステム 50 を示すブロック図である。ユーティリティサブシステム 50 は、USB 2.0 を介して、SINC デバイス 100、CLU デバイス 60、および PM3 デバイス 70 と通信するサービスプロセッサ 200 を備える。サービスプロセッサ 200 は、協働して、ユーティリティサブシステム 50 の動作を制御するハードウェア要素とソフトウェア要素の双方で構成される。サービスプロセッサ 200 は、USB ホストコントローラ 226 および USB ホストドライバ 224 を用いて、USB 2.0 を介して通信する。USB ホストコントローラ 226 は、ハードウェア要素（USB コントローラチップ）であり、USB ホストドライバ 224 は、それに対応した、USB 通信の制御に用いられるソフトウェアドライバである。

【0018】

サービスプロセッサ 200 はまた、サービスプロセッサ 200 の動作を制御するソフトウェア要素である、オペレーティングシステム 221 を備える。サービスプロセッサ 200 はまた、USB ルータ 218 も含む。USB ルータ 218 は、サービスプロセッサ 200 と、ユーティリティサブシステム 50 内の各デバイス 60、70、および 100 との間に論理接続を提供する、デバイスドライバ 220 を介して、オペレーティングシステム 221 と通信する。デバイスドライバ 220 については後にさらに詳細に説明される。

【0019】

サービスプロセッサ 200 はまた、各種機能性をサーバシステム 11 に提供する多数のソフトウェアモジュールも備える。例えば、サービスプロセッサ 200 は、スキャンサポートモジュール 208、ファームウェア更新モジュール 211、ならびにサーバシステム 11 の全体設計の一部であり、かつ他の USB 接続されたマイクロプロセッサと通信を行い動作する他のモジュールを備える。基本的に、サービスプロセッサ 200 は、サーバシステム 11 が適切に機能できるようにする、ユーティリティサブシステム 50 内の様々な制御機能を提供する。

【0020】

なお図 2 を参照して、ユーティリティサブシステム 50 は、SINC デバイス 100 と、CLU デバイス 60 と、PM3 デバイス 70 と、をさらに備える。これらデバイスはそれぞれ、互いに対してピアと見なされる。同様に、サービスプロセッサもまた、各デバイス 60、70、および 100 に対してピアである。例えば、後に詳細に説明される本発明によれば、各デバイス（SINC 100、CLU 60、および PM3 70）は、USB 2.0 をピアツーピアネットワークとして用いることで、互いに対して、かつサービスプロセッサ 200 と双方向通信することができる。この例においては、USB は LAN であると考

10

20

30

40

50

えることができる。

【 0 0 2 1 】

本発明の一形態によれば、S I N C デバイス 1 0 0、C L U デバイス 6 0、および P M 3 デバイス 7 0 内に含まれるメッセージロジック 1 5 0 は、デバイスと協働して、各デバイスが特別な (unique) 通信プロトコルヘッダをデータブロックの前に付加することができるようにし、これによって U S B に接続された別のデバイスに直接アドレス指定される通信メッセージを形成する。特別通信プロトコルヘッダおよびデータ (すなわち、通信メッセージ) は、次に、標準 U S B ヘッダおよびトレーラで収容され、U S B を介して転送される。特別通信プロトコルヘッダにより、各デバイスがまるでピアツーピアネットワークに接続されているかのように、各デバイスが他のデバイスと互いに通信することが可能になる。

10

【 0 0 2 2 】

S I N C デバイス 1 0 0 は、エグゼクティブ 1 0 1、メッセージロジック 1 5 0、U S B ドライバ 1 0 2、ハードウェアインタフェース 1 0 4、および U S B デバイスコントローラ 1 0 6 を備える。U S B デバイスコントローラ 1 0 6 は、U S B 2 0 を介してメッセージを通信するために、U S B デバイスインタフェースチップおよび U S B ドライバ 1 0 2 とのインタフェースを備えたハードウェア要素である。ハードウェアインタフェース 1 0 4 により、エグゼクティブ 1 0 1 は、S I N C デバイス 1 0 0 に存在する他のハードウェア要素 (図示せず) と通信可能になる。メッセージロジック 1 5 0 (後により詳細に説明する) は、エグゼクティブ 1 0 1 と協働して、S I N C デバイス 1 0 0 が、上述した特別通信プロトコルヘッダを生成し、該ヘッダをデバイスが送信したいと望むデータの前に付加することによって、U S B 2 0 に接続された任意の他のデバイスにメッセージを送信できるようにする。

20

【 0 0 2 3 】

従来の使用方法では、U S B では、ホストと遠隔的に接続された個々のデバイスとの間のメッセージ通信しかできない。例えば、パーソナルコンピュータ (P C) 環境において、U S B は、プロセッサと、プリンタ、マウス、およびキーボード等遠隔的に接続される周辺装置との間のメッセージ通信に用いられる。例えば、マウスがプリンタと通信する方法はない (理由もない)。従来の適用において、U S B ではホスト - デバイス間の通信しかできなかった。

30

【 0 0 2 4 】

本発明の一形態によると、メッセージロジック 1 5 0 は、U S B ルータ 2 1 8 と協働して、特別通信プロトコルヘッダを解析し、U S B 2 0 に接続された任意のデバイス間のピアツーピア通信を可能にする。それによって、U S B 2 0 はピアツーピアネットワークとして機能できるようになる。

【 0 0 2 5 】

C L U デバイス 6 0 は、エグゼクティブ 6 1、メッセージロジック 1 5 0、U S B ドライバ 6 2、ハードウェアインタフェース 6 4、スキャンインタフェース 6 7、および U S B デバイスコントローラ 6 6 を備える。同様に、P M 3 デバイス 7 0 は、エグゼクティブ 7 1、メッセージロジック 1 5 0、U S B ドライバ 7 2、ハードウェアインタフェース 7 4、および U S B デバイスコントローラ 7 6 を備える。エグゼクティブ 6 1 と 7 1、U S B ドライバ 6 2 と 7 2、ハードウェアインタフェース 6 4 と 7 4、および U S B デバイスコントローラ 6 6 と 7 6 はすべて、S I N C デバイス 1 0 0 における各デバイスと同様に動作する。

40

【 0 0 2 6 】

簡略化のために図 2 からは省略されているが、サービスプロセッサ 2 0 0、S I N C デバイス 1 0 0、C L U デバイス 6 0、および P M 3 デバイス 7 0 はそれぞれ、本発明のソフトウェア部分の実行に必要なプロセッサ、メモリ (ランダムアクセスメモリ (R A M) および読み出し専用メモリ (R O M) を含む)、インタフェース、および接続を備える。

【 0 0 2 7 】

50

本発明のプロトコルスタック概観

図3は、本発明の一形態によるプロトコルスタック300を示す概略図である。サービスプロセッサ200は、USBホストコントローラ226、USBホストドライバ224、およびUSBルータ218を備える。

【0028】

SINCデバイス100は、USBデバイスコントローラ106、USBデバイスドライバ102、およびメッセージロジック150を備える。図3に示すように、USBホストコントローラ226は、物理的接続(USB)20を介して、USBデバイスコントローラ106と通信する。USBホストドライバ224は、論理接続321を介して、USBデバイスドライバ102と論理的に通信する。USBルータ218は、本発明に関連する通信プロトコルを用い、パイプ111、112および114を介して、メッセージロジック150と論理的に通信する。

10

【0029】

図4は、本発明による、メッセージフォーマット330を示す概略図である。メッセージフォーマット330は、USBヘッダ331、通信プロトコルヘッダ355、データ360、およびUSBトレイラ332を含む。本発明の一形態によれば、任意のデバイス、例えば上述したSINCデバイス100、CLUデバイス60、PM3デバイス70、およびサービスプロセッサ200によって、通信プロトコルヘッダ355(後により詳細に説明される)がデータ360の前に付加され(前に取り付けられ)、その結果通信メッセージ350が形成される。通信メッセージ350は、標準USBヘッダ331およびトレイラ332によって収容される。このようにして、USB20に接続された個々の任意のデバイスが、一意に識別されるメッセージを、USB20に接続された他の任意のデバイスに送信することができる。

20

【0030】

システム動作

図5は、図2のサービスプロセッサ200とSINCデバイス100間の接続性を示すブロック図である。図5には、1つのみのデバイス(すなわち、SINCデバイス100)をUSB20を介してサービスプロセッサ200に接続して図示するが、かかる多くのデバイスがサービスプロセッサ200に接続されるものと意図される。本発明の動作を説明するために、かつ例として、ファームウェア更新モジュール211(図2におけるいかなるモジュールもメッセージを送信可能であるが)が、接続266を介して、通信メッセージをUSBルータ218に送信する。接続266を介して送信されるメッセージは、通信プロトコルヘッダ355およびデータ360を含み、このメッセージは、USBルータ218によって受信される。通信プロトコルヘッダ355およびデータ360については、後に図7に関してより詳細に説明する。USBルータ218は、接続266を介して受信したメッセージを検査し、添付された通信プロトコルヘッダ355を解析することで、メッセージをどこに送信すべきかを決定する。例えば、ファームウェア更新モジュール211からのメッセージは、例えば、SINCデバイス100、CLUデバイス60、およびPM3デバイス70等(これらに制限されない)、USBに接続された他の任意のデバイスを宛先とすることができる。USBルータ218は、複数のデバイスドライバ251、252、および254に接続される。デバイスドライバ251は、SINCデバイス100と関連付けられる。デバイスドライバ252は、CLUデバイス60に関連付けられ、デバイスドライバ254はPM3デバイス70に関連付けられる。各デバイスタイプ毎にデバイスドライバを設けることで、異なるデバイスタイプに行くメッセージに優先順位を付けることが可能になる。しかし、デバイスドライバは一つだけでもすべてのデバイスタイプに対応することができる。好ましい実施形態においては、接続されたデバイスタイプ(すなわち、SINCデバイス100、CLUデバイス60、およびPM3デバイス70)それぞれに関連したデバイスドライバが一つずつサービスプロセッサ200に備えられる。これらのデバイスドライバ251、252、および254は、例としてのみ図示されるものである。

30

40

50

【 0 0 3 1 】

接続 2 6 6 を介して受信したメッセージの宛先に応じて、U S B ルータ 2 1 8 は、メッセージが S I N C デバイス 1 0 0、C L U デバイス 6 0、または P M 3 デバイス 7 0 に宛先付けられている場合には、そのメッセージを適切なデバイスドライバ 2 5 1、2 5 2、または 2 5 4 に提供し、サービスプロセッサ 2 0 0 に宛先付けられている場合には、サービスプロセッサ 2 0 0 における別のエンティティに提供する。例として、図 5 は、S I N C デバイス 1 0 0 を備え、これは、あくまで例を示すためのものとして、接続 2 6 6 を介して受信したメッセージがアドレス指定されるデバイスであると仮定する。U S B ルータ 2 1 8 は、接続 2 6 6 を介して受信したメッセージが、S I N C デバイス 1 0 0 に宛先付けられていることを認識し、該メッセージをデバイスドライバ 2 5 1 に送る。メッセージが S I N C デバイス 1 0 0 からサービスプロセッサ 2 0 0 において受信されると、該メッセージもまたデバイスドライバ 2 5 1 に現れる。このような場合、受信したメッセージは、U S B ルータ 2 1 8 に転送され、U S B ルータ 2 1 8 は、本発明の一形態により、該メッセージの前に付加された通信プロトコルヘッダを解析し、該メッセージをどこに送信すべきかを決定する。例えば、もし S I N C デバイス 1 0 0 がメッセージを C L U デバイス 6 0 に送信すると、メッセージはデバイスドライバ 2 5 1 に現れ、U S B ルータ 2 1 8 に転送される。次に、U S B ルータ 2 1 8 は、該メッセージを C L U デバイス 6 0 に関連付けられたデバイスドライバ 2 5 2 に送る。S I N C 1 0 0 から受信したメッセージがサービスプロセッサ 2 0 0 に宛先付けられていた場合、U S B ルータ 2 1 8 は、デバイスドライバ 2 5 1 から受信したメッセージを受け入れ、接続 2 6 2 を介して、該メッセージをサービスプロセッサ入力メッセージハンドラ 2 5 0 に転送する。サービスプロセッサ入力メッセージハンドラ 2 5 0 は、通信プロトコルヘッダ 3 5 5 を除去し、データ 3 6 0 を適切なモジュール、本例では、ファームウェア更新モジュール 2 1 1 に接続 2 6 4 を介して転送する。あるいはまた、サービスプロセッサ入力メッセージハンドラ 2 5 0 は、データ 3 6 0 を接続 2 6 7 を介して、他のアプリケーションモジュール 2 6 8 として表される任意適切な他のアプリケーションモジュールに転送する。

【 0 0 3 2 】

ここで再び、ファームウェア更新モジュール 2 1 1 から S I N C デバイス 1 0 0 に送信されるメッセージについての考察に戻ると、U S B ルータ 2 1 8 は、メッセージをデバイスドライバ 2 5 1 に入れ、該デバイスドライバ 2 5 1 は、次に、メッセージをキュー 2 5 6 に入れる。図 5 のデバイスドライバ 2 5 1、2 5 2、および 2 5 4 は、図 2 のデバイスドライバ 2 2 0 を示し、かつこれに対応する。例示のために説明される本例では、上記タイプ (S I N C、P M 3、および C L U) 毎に 1 つのデバイスがあるものと仮定される。実際には、デバイスタイプ毎に複数のデバイスがありうる。同一のデバイスタイプのデバイスはそれぞれ、そのデバイスタイプ用のデバイスドライバ内に、各自専用のキューを有する。したがって、それぞれドライバの対応するデバイスタイプの特定のデバイスに対応する複数のキューセットを処理することによって、一台のデバイスドライバで各自のデバイスタイプのデバイスを複数扱うことができる。

【 0 0 3 3 】

本発明によれば、論理接続が、サービスプロセッサ 2 0 0 と、S I N C デバイス 1 0 0 等の遠隔的に接続されたデバイスとの間に確立される。サービスプロセッサ 2 0 0 および S I N C デバイス 1 0 0 の双方に示される矢印 1 1 1、1 1 2、および 1 1 4 は、サービスプロセッサ 2 0 0 と S I N C デバイス 1 0 0 の間に存在する論理接続を表す、「パイプ」の概念を示している。矢印 1 1 1 は、バルクアウトパイプを表し、矢印 1 1 2 は、割り込みパイプを表し、矢印 1 1 4 は、バルクインパイプを表す。パイプの概念は、U S B バス通信システムの分野において周知であり、伝送制御プロトコル / インターネットプロトコル (T C P / I P) ネットワークスタックにおける接続ソケットの概念に対応する。

【 0 0 3 4 】

入力メッセージ動作例

本発明の一形態によれば、U S B ホストコントローラ 2 2 6 およびサービスプロセッサ 2

00は、SINCデバイス100内のUSBデバイスコントローラ106を介して、割り込みパイプ112を定期的にポーリングする。SINCデバイス100が、サービスプロセッサ200あるいはUSB20に接続された別のデバイスのいずれかに送信したいデータを有する場合、デバイスによりデータ360の前に付加された通信プロトコルヘッダ355がキュー156に入れられる。キュー156は、デバイス出力メッセージハンドラ152内にあり、デバイス出力メッセージハンドラ152は、通信プロトコルヘッダ355を割り込みパイプ112に入れる。USBホストコントローラ226は、割り込みパイプ112をポーリングすると、割り込みパイプにおける通信プロトコルヘッダ355の存在を認識し、それによって、SINCデバイス100が送信したい情報を有していることを知らせる。次に、通信プロトコルヘッダ355が、デバイス出力メッセージハンドラ152から割り込みパイプ112を介してSINCデバイスドライバ251に転送される。USB業界規格用語において、通信プロトコルヘッダ355は、割り込みデータパケットと呼ばれる。

10

【0035】

SINCデバイスドライバ251は、ヘッダ355の長さフィールド(図7において後述するバイト6および7)を読み出すことで、通信プロトコルヘッダ355にデータフィールド360を足した長さを読み出す。SINCデバイスドライバ251は、SINCデバイス100が送信データをさらに有しているか否かを判断する。これは、通信プロトコルヘッダ355内の特定フィールドを解析することで達成され、これについては、後に図7に関して詳細に説明する。SINCデバイスドライバ251が、SINCデバイス100が送信する情報をさらに有していると判断した場合、SINCデバイスドライバ251は、「USBバルク読み出し」コマンドをSINCデバイス100に送信する。この結果、デバイス出力メッセージハンドラ152からキュー158を通して、次にバルクインパイプ114を通してサービスプロセッサ200に転送される通信メッセージ(すなわち、データ360)の平衡が保たれる。転送はパイプ111、112、および114を介して行われるものとして説明したが、重要なことは、パイプは論理接続であり、情報の転送は、実際にはUSB20を介して行われることであることに留意されたい。

20

【0036】

本発明の本形態において、上記のメッセージ交換が行われた場合、デバイス出力メッセージハンドラ152は、SINC100上のいくつかのソフトウェアモジュール(例えば、新メッセージブロック126、ファームウェア更新ルーチン119、またはリセットルーチン118)のうちの1つからメッセージを受信したものと考えられる。メッセージを発するモジュールは、通信プロトコルヘッダ355をデータ360の前に付加したものと考えられる。デバイス出力メッセージハンドラ152は、次に、通信プロトコルヘッダ355をキュー156に入れ、データ360をキュー158に入れたと考えられる。リセットルーチン118、ファームウェア更新ルーチン119、および新メッセージブロック126を用いて説明されたが、SINCデバイス100から送信されるメッセージは、SINCデバイス100上に存在するが図示されていない種々多数のメッセージ源のうちのいずれから発せられてもよい。

30

【0037】

本発明の一形態によれば、デバイスドライバ251は、「読み出し」呼び出しを用いることで、割り込みパイプ112から読み出しを行う。呼び出しの概念は当業者には周知である。割り込みパイプ112にメッセージがない場合、呼び出しは保留される。USBホストドライバ224が、割り込みパイプ112において通信プロトコルヘッダ355を検出した場合、呼び出しが戻される。次に、デバイスドライバ251は、通信プロトコルヘッダ355を割り込みパイプ112から除去する。そして、通信プロトコルヘッダ355が、付随するデータ360があることを示す場合には、デバイスドライバ251は、読み出し呼び出しをバルクインパイプ114に出す。この読み出し呼び出しは、読み出しが完了するまで保留される。

40

【0038】

50

出力メッセージ動作例

出力メッセージ(すなわち、サービスプロセッサ200からSINCデバイス100等の遠隔デバイスに送信されるメッセージ)の転送について、次に説明する。サービスプロセッサ200によって処理されるメッセージは、サービスプロセッサ200から発せられるか、あるいはUSB20に接続された別のデバイス(SINCデバイス100等)から発せられる。本発明によれば、USBルータ218は、出力メッセージをキュー256、257、または258のうち、メッセージがアドレス指定されているデバイスに関連する1つに入れる。本例において、USBルータ218は、メッセージをデバイスドライバ251に渡し、デバイスドライバ251は、最終的にSINCデバイス100に伝送するために、キュー256にメッセージを入れる。現在SINCデバイス100に送信処理中のメッセージがない場合には、デバイスドライバ251は、すぐにキュー256からメッセージを抽出して、該メッセージをSINCデバイス100に送信する。送信中の前の出力メッセージがある場合には、新しいメッセージはキュー256に保持され、前のメッセージに対する「送信完了」割り込みが発生すると、取り出されてSINCデバイス100に送信される。USBルータ218からバルクパイプ111を通して送信されたメッセージは、次に、デバイス入力メッセージハンドラ151に受信され、キュー154に入れられる。当技術分野で周知の「関数呼び出し」を用いて、エグゼクティブ101は、接続109を介して、デバイス入力メッセージハンドラ151を定期的に呼び出し、すなわちポーリングし、キュー154にメッセージがあるか否かを判断する。エグゼクティブ101はまた、同様に他の関数呼び出しも行う。例えば、エグゼクティブ101は、接続103を介して定期的にハードウェアインタフェース104を呼び出し、SINCデバイス100におけるハードウェアの状態を判断する。ハードウェアインタフェース104は、USB20に接続された他の任意のデバイスに送信する情報を有している場合には、接続108を介してメッセージロジック150と通信することができる。デバイス入力メッセージハンドラ151がキュー154に情報を有している場合、エグゼクティブ101にポーリングされると、デバイス入力メッセージハンドラ151は、キュー154に存在するメッセージを実行するために、適切な関数呼び出しを生成する。

【0039】

例えば、ファームウェア更新モジュール211が、USB20を介してファームウェア更新メッセージをSINCデバイス100に送信したものと仮定した場合、デバイス入力メッセージハンドラ151は、このメッセージをキュー154において検出すると、接続117を介して適切な関数呼び出しを実行して、ファームウェア更新ルーチン119を実行する。このファームウェア更新は、例示目的のためにのみ示されるものである。他のメッセージならば、異なるルーチンを呼び出す。デバイス入力メッセージハンドラ151がファームウェア更新のメッセージの代わりにリセットコマンドを受信したならば、リセットルーチン118に対して接続116を介して関数呼び出しを実行し、リセットルーチン118が、接続127を介して、リセットコマンドをリセットハードウェア128に送信する。

【0040】

SINCデバイス100がメッセージをUSB20に配置された別のデバイスまたはサービスプロセッサ200に送信するには、種々多数のシナリオがある。例えば、ファームウェア更新モジュール211から受信したファームウェア更新メッセージに応答して、ファームウェア更新ルーチン119を実行すると、ファームウェア更新ルーチン119は、メッセージの受信承認と、メッセージにおいて要求されたアクションの成功または失敗を示すメッセージの双方またはいずれかを送信することができる。これは、図7および図8に関して説明する通信プロトコルヘッダ355によって達成することができる。あるいは、非送信請求状態更新メッセージをSINCデバイス100から新メッセージブロック126を介して送信することもできる。別の例として、新メッセージブロック126は、メッセージをパワーモニタであるPM3デバイス70に送信し、SINCが必要とする電力を知らせる。

10

20

30

40

50

【 0 0 4 1 】

メッセージを与えられる方法に関わりなく、デバイス出力メッセージハンドラ 1 5 2 は、通信プロトコルヘッダ 3 5 5 およびデータ 3 6 0 を含むメッセージを解析し、ヘッダ 3 5 5 をキュー 1 5 6 に入れ、データ 3 6 0 をキュー 1 5 8 に入れる。本発明において、通信プロトコルヘッダ 3 5 5 は、メッセージを送信したデバイスによって、データ 3 6 0 の前に付加される。例えば、ファームウェア更新ルーチン 1 1 9 が、ファームウェア更新モジュール 2 1 1 から受信したメッセージに対して返信するよう命令された場合、ファームウェア更新ルーチン 1 1 9 は、以下の説明に従って、メッセージの前に通信プロトコルヘッダ 3 5 5 を付加し、ヘッダおよびメッセージをデバイス出力メッセージハンドラ 1 5 2 に送信する。ファームウェア更新ルーチン 1 1 9 によって前に付加された通信プロトコルヘッダ 3 5 5 は、メッセージの宛先としてサービスプロセッサ 2 0 0 を識別する。ファームウェア更新モジュール 2 1 1 は、ファームウェア更新ルーチン 1 1 9 が現在応答しているメッセージを送信する際、ヘッダのプライベートエリア（図 7 において後に説明されるバイト 8 ~ 1 1）に特別な番号を挿入している。この特別な番号は、ファームウェア更新ルーチン 1 1 9 には認識されないが、サービスプロセッサ入力メッセージハンドラ 2 5 0 がファームウェア更新モジュール 2 1 1 を応答メッセージの所期の受信先として識別できるようにするための識別子である。ファームウェア更新ルーチン 1 1 9 が応答メッセージを作成する際、要求メッセージから応答メッセージにプライベートバイト 3 6 2、3 6 4、3 6 6、および 3 6 7（図 7）をコピーする。メッセージヘッダのプライベートエリアの使用については、図 7 に関して後に説明される。

【 0 0 4 2 】

デバイス出力メッセージハンドラ 1 5 2 は、通信プロトコルヘッダ 3 5 5 を、キュー 1 5 6 に対応する割り込みパイプ 1 1 2 に入れ、もし存在すれば、データ 3 6 0 をキュー 1 5 8 に関連するバルクインパイプ 1 1 4 に入れる。キュー 1 5 6 は、二重バッファキューであり、二重バッファされる U S B デバイスコントローラ 1 0 6 に対応する。

【 0 0 4 3 】

図 6 は、図 5 におけるキューとして与えられる図 3 のパイプを示す略図である。図示のように、ルータ 2 1 8 および S I N C デバイス 1 0 0 は、バルクアウトパイプ 1 1 1、割り込みパイプ 1 1 2、およびバルクインパイプ 1 1 4 を介して接続される。図 6 は、メッセージが交換される個々の経路としてパイプを示している。なお、図 6 には三本の個々のパイプとして示されるが、パイプはすべて U S B 2 0 に存在することに留意されたい。

【 0 0 4 4 】

メッセージヘッダフォーマット

図 7 は、図 4 のメッセージフォーマット 3 3 0 の通信メッセージ 3 5 0 の一例を示す概略図である。通信メッセージ 3 5 0 は、通信プロトコルヘッダ 3 5 5 およびデータ 3 6 0 を含む。通信プロトコルヘッダ 3 5 5 は、1 2 バイトの情報を含む。該バイトは、必ずしも図示した順序である必要はない。宛先キャビネット番号 3 5 1（バイト 0）は、宛先キャビネット番号に関する情報を含む。

【 0 0 4 5 】

宛先アドレス 3 5 2（バイト 1）は、メッセージが宛先付けられたデバイスのキャビネット内のアドレスである。特別なデバイス番号を U S B 2 0 に接続された各デバイスに割り当てることによって、キャビネット番号と宛先アドレス 3 5 2（バイト 1）を合わせて、デバイス（すなわち、S I N C デバイス 1 0 0、C L U デバイス 6 0、および P M 3 デバイス 7 0）が識別される。

【 0 0 4 6 】

ソースキャビネット番号 3 5 4（バイト 2）は、ソースキャビネット番号（すなわち、メッセージを発したデバイスを含むキャビネットの番号）であり、ソースアドレス 3 5 6（バイト 3）は、メッセージを送信した、デバイスのキャビネット内のアドレスである。キャビネット内の宛先アドレスおよびソースアドレスを以下の表 1 に示す。

【 0 0 4 7 】

【表 1】

通信プロトコルデバイス	通信プロトコルアドレス
S I N C 0	0 x 0 0
S I N C 1	0 x 0 1
S I N C 2	0 x 0 2
S I N C 3	0 x 0 3
S I N C 4	0 x 0 4
S I N C 5	0 x 0 5
S I N C 6	0 x 0 6
S I N C 7	0 x 0 7
P M 3	0 x 1 0
C L U	0 x 2 0
サービスプロセッサ	0 x 4 0
予備	0 x 5 0 – 0 x f f

10

20

【 0 0 4 8 】

表 1 に見られるアドレスは、他とは異なる任意の値を用いることができるという点において、自由に決めることのできるものである。名称の欄はあらかじめ定義されており、デバイスに自由に名称をつけることはできない。デバイスが自身の名称（アドレス）および通信先のデバイスの名称（アドレス）を知ることができるように、全般的なネーミング方式を予め決めておくことが望ましい。あるいは、ディレクトリサービスがそのようなネーミング方式によってかわってもよい。

30

【 0 0 4 9 】

コマンドバイト 3 5 7（バイト 4）は、受信モジュールにメッセージの処理の方法を指示する。好ましい実施形態において用いられるコマンドおよび値のいくつかの具体例を以下に示す。

【 0 0 5 0 】

```

BusboyCmmdExecStatus      = 0 x 0 0
BusboyCmmdUnsolicitedStatus = 0 x 0 1、/*CLU/SINC/PM3/PDI to SUB */
BusboyCmmdBootNormal      = 0 x 0 2、/*SUB to SINC */
BusboyCmmdBootType1       = 0 x 0 3、/*SUB to SINC */
BusboyCmmdTocCell         = 0 x 0 4、/*SUB to SINC */
BusboyCmmdResetCell       = 0 x 0 5、/*SUB to SINC */
BusboyCmmdResetYourself   = 0 x 0 6、/*SUB to CLU/SINC/PM3/PDI */
BusboyHdrIdlcmHdrResponse = 0 x f 3、/*SINC to SUB */

```

40

コマンドの意味および値は、本発明に従って構築されたシステムの実施の詳細によって決定される。

【 0 0 5 1 】

フラグバイト 3 5 8（バイト 5）は、フラグを表す。フラグについては、後に図 8 に関してより詳細に説明される。メッセージ長 3 5 9（バイト 6）は、メッセージ長（最下位バイト）を表し、メッセージ長 3 6 1（バイト 7）は、メッセージ長（最上位バイト）を表す。メッセージ長は、メッセージにおけるバイトの数であり、メッセージヘッダ 3 5 5 の

50

12 バイトを含む。入力メッセージハンドラは、メッセージ長が12 バイトよりも大きい
か否かをチェックすることで、メッセージ350のデータ部分360があるか否かを判断
することができる。

【0052】

バイト8 362からバイト11 367は、応答を要する要求を発したソフトウェアモ
ジュール（本例ではファームウェア更新モジュール211）を識別するプライベートデー
タを表す。メッセージを送信する相手のデバイス（例えば、SINCデバイス100）に
送信メッセージに対する返信を期待する場合、送信側のデバイスにおけるモジュール（例
えば、サービスプロセッサ200におけるファームウェア更新モジュール211）は、識別
子をバイト362、364、366、367におけるヘッダのプライベートエリアに挿入することによって、自身の宛て先を知らせる。応答側デバイスにおける応答モジュール（例えば、SINCデバイス100におけるファームウェア更新ルーチン119）は、プライベートエリアの内容を返信メッセージにコピーするよう要求される。

10

【0053】

プライベートエリアにおける識別子は、発信デバイス上の入力メッセージハンドラ（例えば、サービスプロセッサ200上のサービスプロセッサ入力メッセージハンドラ250）が、返信先モジュール（例えば、ファームウェア更新モジュール211応答キュー）として認識することができる値である。識別子は、例えば、元の要求を送信したモジュールにおいて呼び出す関数についての応答キューの識別子、またはアドレスでありうる。サービスプロセッサ入力メッセージハンドラ250は、プライベートエリア（ヘッダ355におけるバイト8からバイト11）の内容に基づいて、すべての返信を指定されたモジュールに送信する。

20

【0054】

アドレス指定を説明するために、図7に示す発信元および宛先バイト351、352、354、356は、デバイスを識別指定するものとする。メッセージが（あるモジュールによって発せられた）コマンドであり、かつ宛先デバイスに2つ以上のコマンド処理モジュールがある場合、コマンドバイト357（バイト4）の内容によって、所与のコマンドを処理する、デバイス内の特定のモジュールを識別することができる。メッセージが（先に受信したコマンドに応答して発せされた）返信メッセージである場合、コマンドバイト357（バイト4）の内容は、プライベートエリアバイト362、364、366、および367（バイト8～バイト11）の内容と合わせて、返信メッセージを渡すべきモジュールを指定する。コマンドバイト357（バイト4）の内容はまた、プライベートエリアバイト362、364、366、および367（バイト8～バイト11）の内容と合わせて、異なるデバイスに対する複数の未処理要求を有するモジュールが、各デバイスからの返信メッセージを未処理の要求と結びつけることができるようにする。

30

【0055】

メッセージフォーマット350はまた、メッセージフォーマット350のオプションコマンドデータバイト368および369（バイト12からバイト長+11まで）を含む、データ360も含む。オプションコマンドデータは、オプションデータを含む。データ360は、0バイト以上のデータを含む可変長フィールドである。このデータは、コマンドバイト357（バイト4）における特定のコマンドに関連するものである。例えば、コマンドが「EPROMセクタプログラム」である場合には、パケットのデータ部分は、EPROMセクタにプログラムされるデータである。コマンドが「状態応答」であり、かつパケットが先の「状態照会」メッセージの結果として送信される場合には、データ部分は、実際のデバイス状態である。

40

【0056】

図8は、図7のフラグバイト358（バイト5）を示す概略図である。図示のように、このフォーマットにおいては、フラグバイト358のビット4～ビット7はメッセージ識別情報を含む。ビット0およびビット1は、USB2.0に接続されたSINC100等、個々のデバイスに要求する応答に応じてセットすることができる。例えば、最下位ビット（

50

LSB) 0 がロジック「0」にセットされる場合は、図5のSINCデバイス100は、送信されたメッセージの受信の確認通知をする必要がないことを示す。LSB 0 がロジック「1」にセットされる場合は、受信デバイス（すなわち、図5に関して先に説明した例では、SINCデバイス100）はメッセージの受信の確認通知をしなければならないことを示す。同様に、フラグバイト358（バイト5）のビット1がロジック「0」にセットされる場合は、メッセージに含まれるコマンドの処理が成功したか失敗したかについて、メッセージの発信元に報告する必要がないことを示す。同様に、ビット1がロジック「1」にセットされる場合は、メッセージに含まれるコマンドの処理が成功したか失敗したかについて、発信デバイスに報告しなければならないことを示す。

【0057】

10

初期化

再度、図5を参照して、標準USB初期化の一部として、USBホストドライバ224は、デバイスがバスに接続されるときに、各デバイス（すなわち、SINCデバイス100、CLUデバイス60等）を列挙する（アドレスを割り当てる）。（なお、これは物理レイヤアドレスであり、表1に示す論理レイヤアドレスとは異なることに留意する）。その際、デバイスは、デバイス記述子をホストに戻す。デバイス記述子によって、デバイスのタイプが識別される。これは、USB通信の分野における当業者に良く知られている。このとき、好ましい実施形態におけるデバイスは、それが、例えば、PM3デバイス70、CLUデバイス60、またはSINCデバイス100であると報告する。デバイスはまた、表1に示す符号化を用いて、自身の固有のアドレス（例えば、キャビネット0におけるSINC100、SINC番号2）も報告する。これにより、USBホストドライバ224は、この特定のデバイスに適したキュー（例えば、キュー256と、パイプ112および114）を初期化することができる。

20

【0058】

表1に示されるデバイスのアドレスは、デバイスの名称と同義であり固定されたものである。これに対して、デバイスタイプおよび番号のアドレスを固定せず、動的に割り当てる方式を考えることもできる。諸タイプのデバイスの組み合わせをより柔軟に行う必要がある場合に、このような方式の方が望ましいことがある。もしそのようなシステムが設計されたとしても、他のエンティティがそのデバイスをアドレス指定することができるように、なお一つのデバイスごとに一つの「名称」を持たせておくべきである。しかし、このよう

30

【0059】

この発明は例として次の実施形態を含む。

【0060】

(1) ユニバーサルシリアルバス(USB)(20)をピアツーピアネットワークとして用いる方法であって、

ルータ(218)を備えるホストプロセッサ(200)をUSB(20)に接続するステップと、

40

それぞれ互いに対して、かつ前記ホストプロセッサ(200)に対してピアである複数のデバイス(60、70、100)を前記USB(20)に接続するステップと、

前記複数のデバイス(60、70、100)または前記ホストプロセッサ(200)のうちの1つにおいて、メッセージ(360)を形成するステップと、

前記メッセージ(360)の前にヘッダ(355)を付加するステップと、

前記ルータ(218)および前記USB(20)を介して、前記メッセージ(360)を前記デバイス(60、70、100)のうちの1つから他の任意のデバイスに、または前記ホストプロセッサ(200)に直接伝送するステップと、を含む、方法。

【0061】

(2) 前記ヘッダ(355)は、前記デバイス(60、70、100)のうちの一つを

50

識別する複数のフィールドをさらに含む、上記（１）記載の方法。

【００６２】

（３） 前記デバイス（６０、７０、１００）はそれぞれ、前記複数のフィールドを認識するよう構成されたロジック（１５０）をさらに備える、上記（１）記載の方法。

【００６３】

（４） 前記ホストプロセッサ（２００）は、前記ルータ（２１８）に関連する第１のデバイスドライバ（２５１）をさらに備え、前記デバイス（６０、７０、１００）はそれぞれ、前記ルータ（２１８）における前記第１のデバイスドライバ（２５１）に対応する第２のデバイスドライバ（１０２）をさらに備え、前記第１のデバイスドライバ（２５１）および第２のデバイスドライバ（１０２）は、前記ルータ（２１８）と前記デバイス（６
0、70、100）それぞれの間に論理接続（３２１）を確立するよう構成された、上記
（１）記載の方法。

10

【００６４】

（５） 前記論理接続（３２１）は、割り込みパイプ（１１２）と、バルクインパイプ（１１４）と、バルクアウトパイプ（１１１）と、を含む、上記（４）記載の方法。

【００６５】

（６） ユニバーサルシリアルバス（ＵＳＢ）（２０）をピアツーピアネットワークを用いるシステムであって、

ホストメッセージハンドラ（２５０）を含み、メッセージ（３６０）の前にヘッダ（３５５）を付加するよう構成され、該ホストメッセージハンドラ（２５０）は、該ヘッダ（３
55）を認識するよう構成された、ホストプロセッサ（２００）と、

20

前記ホストプロセッサ（２００）に連係され、それぞれデバイスメッセージハンドラ（１５０）を含み、前記メッセージ（３６０）の前に前記ヘッダ（３５５）を付加するよう構成された複数のデバイス（６０、７０、１００）であって、前記デバイスメッセージハンドラ（１５０）もまた、前記ヘッダ（３５５）を認識するよう構成された、複数のデバイス（６０、７０、１００）と、

前記ホストプロセッサ（２００）に連係された前記複数のデバイス（６０、７０、１００）のうちの任意のデバイスと前記ホストプロセッサ（２００）間で、ユニバーサルシリアルバス（ＵＳＢ）（２０）を介して前記メッセージ（３６０）を直接ルーティングするよう構成された、前記ホストプロセッサ（２００）に連係されたルータ（２１８）と、
を備える、システム。

30

【００６６】

（７） 前記ヘッダ（３５５）は、前記デバイス（６０、７０、１００）のうちの一つを識別する複数のフィールドをさらに含む、上記（６）記載のシステム。

【００６７】

（８） 前記ホストプロセッサ（２００）は、前記ルータ（２１８）に関連する第１のデバイスドライバ（２５１）をさらに備え、前記デバイス（６０、７０、１００）はそれぞれ、前記ルータ（２１８）における前記第１のデバイスドライバ（２５１）に対応する第２のデバイスドライバ（１０２）をさらに備え、前記第１のデバイスドライバ（２５１）および第２のデバイスドライバ（１０２）は、前記ルータ（２１８）と前記デバイス（６
0、70、100）それぞれの間に論理接続（３２１）を確立するよう構成された、上記
（６）記載のシステム。

40

【００６８】

（９） 前記論理接続（３２１）は、割り込みパイプ（１１２）と、バルクインパイプ（１１４）と、バルクアウトパイプ（１１１）と、を含む、上記（８）記載のシステム。

【００６９】

（１０） 前記デバイスメッセージハンドラ（１５０）は、前記メッセージ（３６０）を前記バルクアウトパイプ（１１１）から受信するよう構成された入力メッセージハンドラ（１５１）と、

前記ヘッダ（３５５）を前記割り込みパイプ（１１２）に挿入するとともに、前記メッセ

50

ージ（３６０）に関連するデータを前記バルクインパイプ（１１４）に挿入するよう構成された出力メッセージハンドラ（１５２）と、
をさらに備える、上記（９）記載のシステム。

【００７０】

本発明の原理から本質的に逸脱せずに、本発明の好ましい実施形態に、多くの変更および変形を行いうることが、当業者には理解されよう。例えば、多数のデバイスがＵＳＢを介して通信する任意のシステムにおいて、本発明を用いることができる。このような変更および変形はすべて、特許請求の範囲に規定される本発明の範囲に包含される。

【図面の簡単な説明】

【図１】 本発明の概念を示すために用いられる、サーバシステムを示すブロック図。

10

【図２】 図１のユーティリティサブシステムを示すブロック図。

【図３】 本発明の一態様によるプロトコルスタック３００を示す概略図。

【図４】 本発明によるメッセージフォーマットを示す概略図。

【図５】 図２のサービスプロセッサと、システム抽象レイヤインタフェースネットワークコントローラ（ＳＩＮＣ）デバイス間の接続性を示すブロック図。

【図６】 図５においてキューとして実施される、図３のパイプを示す略図。

【図７】 図４のメッセージフォーマットの通信メッセージの一例を示す概略図。

【図８】 図７の通信プロトコルフラグバイトを示す概略図。

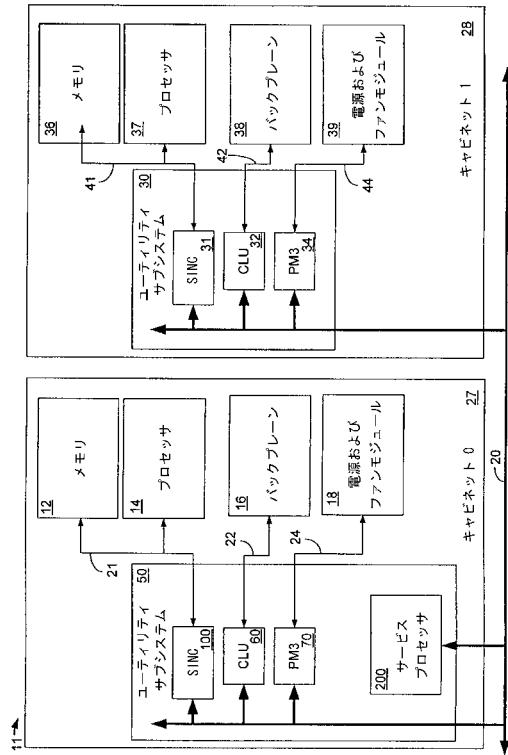
【符号の説明】

２０ ユニバーサルシリアルバス
 ６０、７０、１００ デバイス
 １０２ 第２のデバイスドライバ
 １１１ バルクアウトパイプ
 １１２ 割り込みパイプ
 １１４ バルクインパイプ
 １５０ ロジック、デバイスメッセージハンドラ
 １５１ 入力メッセージハンドラ
 １５２ 出力メッセージハンドラ
 ２００ ホストプロセッサ
 ２１８ ルータ
 ２５０ ホストメッセージハンドラ
 ２５１ 第１のデバイスドライバ
 ３２１ 論理接続
 ３５５ ヘッダ
 ３６０ メッセージ

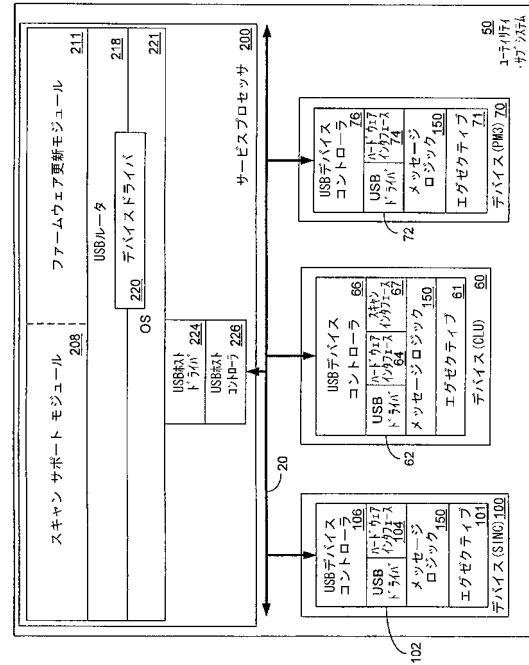
20

30

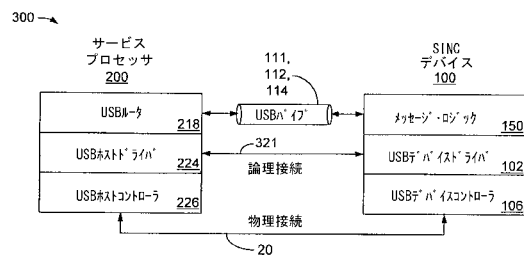
【図 1】



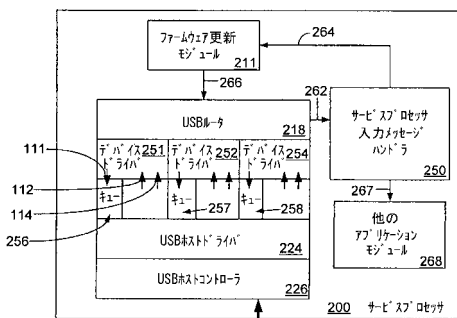
【図 2】



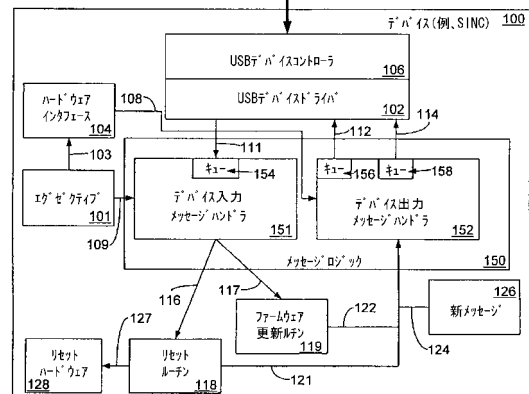
【図 3】



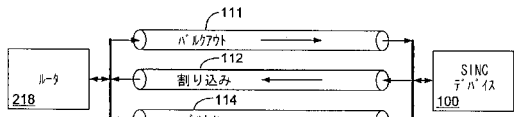
【図 5】



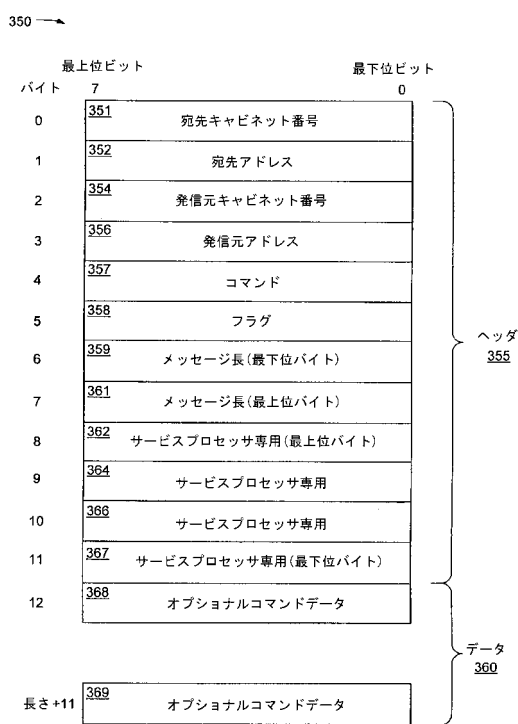
【図 4】



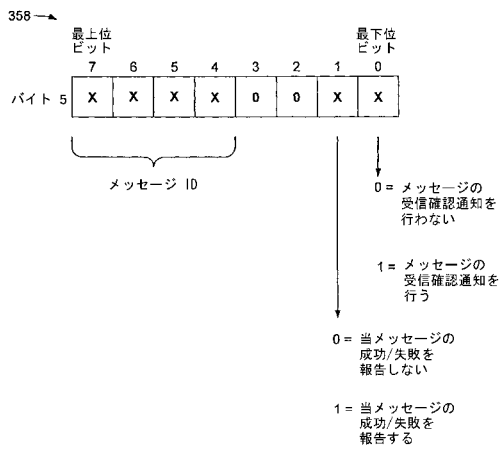
【図 6】



【図 7】



【図 8】



フロントページの続き

- (72)発明者 ポール・エイチ・ブーシェアー
アメリカ合衆国75068テキサス州リトル・エルム、ウッドロウ・サークル 129
- (72)発明者 ロナルド・イー・ギルバート・ジュニア
アメリカ合衆国75093テキサス州プラノ、ミスティ・ヘヴン・レーン 2017
- (72)発明者 クリスチーヌ・コーバー
アメリカ合衆国75287テキサス州ダラス、ホライゾン・エヌ・パークウェイ 4341、ナンバー 131

審査官 矢頭 尚之

- (56)参考文献 特開平11-284636(JP,A)
特開2000-261447(JP,A)
特開2000-315184(JP,A)
特開平10-340199(JP,A)
特表2002-519784(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04L 12/40

G06F 13/12