

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2009-182459
(P2009-182459A)

(43) 公開日 平成21年8月13日(2009.8.13)

(51) Int.Cl.	F I	テーマコード (参考)
HO4L 29/06 (2006.01)	HO4L 13/00 305B	5K027
HO4L 12/56 (2006.01)	HO4L 12/56 300A	5K030
HO4M 1/00 (2006.01)	HO4M 1/00 R	5K034
HO4L 12/66 (2006.01)	HO4L 12/66 Z	

審査請求 有 請求項の数 8 O L (全 23 頁)

(21) 出願番号 特願2008-18121 (P2008-18121)
(22) 出願日 平成20年1月29日 (2008.1.29)

(特許庁注：以下のものは登録商標)

1. Linux
2. リナックス

(71) 出願人 000002185
ソニー株式会社
東京都港区港南1丁目7番1号

(74) 代理人 100095957
弁理士 亀谷 美明

(74) 代理人 100096389
弁理士 金本 哲男

(74) 代理人 100101557
弁理士 萩原 康司

(72) 発明者 藤永 裕幸
東京都港区港南1丁目7番1号 ソニー株式会社内

(72) 発明者 伊藤 力
東京都港区港南1丁目7番1号 ソニー株式会社内

最終頁に続く

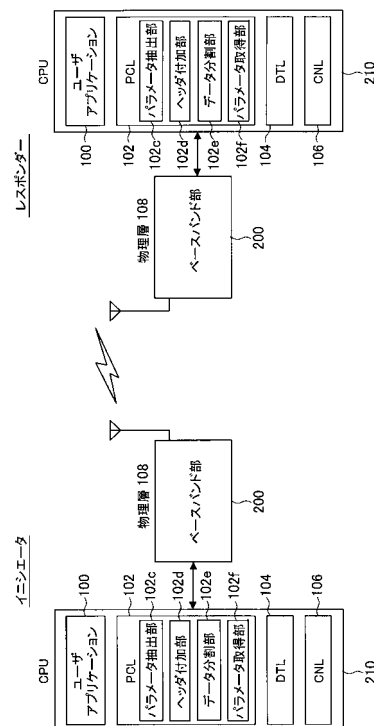
(54) 【発明の名称】 通信装置、通信システム、通信方法及びプログラム

(57) 【要約】

【課題】上位アプリケーションとの通信の互換性を確保するとともに、データ転送効率を向上することができ、且つ処理負荷を最小限に抑えること。

【解決手段】他の通信相手との間で信号の送受信を行う物理層108と、上位のユーザアプリケーション100と物理層108との間を接続するプロトコル変換部(PCL)102とを備え、プロトコル変換部102は、ユーザアプリケーション100から送られた通信開始要求データ(CBW)から、所定のパラメータを抽出するパラメータ抽出部102cと、ユーザアプリケーション100から送られた送信データに抽出したパラメータを含むヘッダを付加するヘッダ付加部102dと、を含み、物理層108は、ヘッダが付加された送信データを送信する。

【選択図】 図8



【特許請求の範囲】**【請求項 1】**

他の通信相手との間で信号の送受信を行う物理層と、
上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備え、
前記プロトコル変換部は、
前記上位アプリケーションから送られた通信開始要求データから、所定のパラメータを抽出するパラメータ抽出部と、
前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加するヘッダ付加部と、を含み、
前記物理層は、前記ヘッダが付加された前記送信データを送信することを特徴とする、
通信装置。

10

【請求項 2】

前記プロトコル変換部は、前記送信データを所定のデータ量毎に分割するデータ分割部を含み、
前記ヘッダ付加部は、前記データ分割部による分割で得られた複数の送信データに対して1つの前記ヘッダを付加することを特徴とする、請求項1に記載の通信装置。

【請求項 3】

前記プロトコル変換部は、前記物理層を介して他の通信相手から受信した受信データから前記パラメータを取得するパラメータ取得部を含むことを特徴とする、請求項1に記載の通信装置。

20

【請求項 4】

前記パラメータに基づいて、前記受信データを記憶領域の所定位置に書き込むデータ書込部を備えることを特徴とする、請求項1に記載の通信装置。

【請求項 5】

前記パラメータは、データの送信方向を示すエンドポイント情報、データのサイズを示すサイズ情報を含むことを特徴とする、請求項1に記載の通信装置。

【請求項 6】

通信装置同士が通信を行う通信システムであって、
前記通信装置は、
通信相手となる通信装置の間で信号の送受信を行う物理層と、
上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備え、
前記プロトコル変換部は、
前記上位アプリケーションから送られた通信開始要求データから、所定のパラメータを抽出するパラメータ抽出部と、
前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加するヘッダ付加部と、を含み、
前記ヘッダが付加された前記送信データを、前記物理層を介して送受信することを特徴とする、通信システム。

30

【請求項 7】

他の通信相手との間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備える通信装置における通信方法であって、
前記上位アプリケーションから送られた通信開始要求データから、所定のパラメータを抽出するステップと、
前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加するステップと、
前記ヘッダが付加された前記送信データを、前記物理層を介して送信するステップと、
を備えることを特徴とする、通信方法。

40

【請求項 8】

他の通信相手との間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備える通信装置におけるプログラムであって、

50

前記上位アプリケーションから送られた通信開始要求データから、所定のパラメータを抽出する手段、

前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加する手段、

前記ヘッダが付加された前記送信データを、前記物理層を介して送信する手段、
としてコンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、通信装置、通信システム、通信方法及びプログラムに関する。

10

【背景技術】

【0002】

従来、例えば下記の特許文献1に記載されているように、移動体の利用者が所有する汎用の携帯端末に移動体の情報を発信できることを意図した移動体通信システムが知られている。

【0003】

【特許文献1】特開2005-191819号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

20

近時においては、通信装置に多種の上位アプリケーションが搭載されることが想定され、通信装置側で各種の上位アプリケーションに対応する必要がある。しかしながら、通信装置が備える物理層と各種の上位アプリケーションとの互換性を持たせることを想定した場合、上位アプリケーションが想定している物理層のデータを、通信装置が備える、転送サイズ等が異なる別の種類の物理層に流す必要が発生する。このため、データの結合、分割等の処理を行う必要が生じ、装置の負荷が増大するという問題が生じる。

【0005】

また、データを受信した装置においても、受信したデータを全て解析した上で、上位アプリケーションに渡すために、データの種別、メモリ上での書き込み位置等を解析する必要が生じ、やはり装置の負荷が増大するという問題がある。

30

【0006】

例えば上位アプリケーションがUSBの場合のケット転送において、通常は、図18に示すように、USB物理層を介してUSBホストとUSBターゲットとの間で通信が行われる。この場合、USBホストからUSBターゲットへの通信要求(CBW)送信、データ(DATA)転送、USBターゲットからUSBホストへの完了通知(CSW)の送信という流れの処理になる。なお、データの転送方向は、CBWの内容に応じて変わり、図18に示すように、USBホストが読み出し動作(READ動作)を行う場合は、USBターゲットからUSBホストへデータが送られる。また、USBホストが書き込み動作(WRITE動作)を行う場合は、USBホストからUSBターゲットへデータが送られる。CBW、CSWは数10バイト程度のデータ量である。一方、USB物理層では、データが512バイトを超える場合は、分割して送信される。512バイトのデータを何回送るかについては、CBWのデータに記載されている。

40

【0007】

図18の場合において、USB物理層の代わりに複数のアプリケーションに対応可能な互換性のある物理層を設けた場合、上位アプリケーション(USB物理層)のデータそのものを、別の転送サイズを持つ物理層に流す必要が生じる。このため、上位アプリケーション(USB物理層)のデータを、互換性のある物理層の転送サイズに合わせて更に結合したり、分割したりすることになる。これにより、処理の負荷が増えると共に、実質的なデータ以外のデータ転送が増加し、転送効率が低下してしまう。また、受信装置側では、受信データを上位アプリケーションに送るために、データの再結合処理を行う必要があり

50

、負荷が増大してしまう。

【0008】

そこで、本発明は、上記問題に鑑みてなされたものであり、本発明の目的とするところは、上位アプリケーションとの通信の互換性を確保するとともに、データ転送効率を向上することができ、且つ処理負荷を最小限に抑えることが可能な、新規かつ改良された通信装置、通信システム、通信方法及びプログラムを提供することにある。

【課題を解決するための手段】

【0009】

上記課題を解決するために、本発明のある観点によれば、他の通信相手との間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備え、前記プロトコル変換部は、前記上位アプリケーションから送られた通信開始要求データから、所定のパラメータを抽出するパラメータ抽出部と、前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加するヘッダ付加部と、を含み、前記物理層は、前記ヘッダが付加された前記送信データを送信する通信装置が提供される。

10

【0010】

上記構成によれば、物理層により他の通信相手との間で信号の送受信が行われ、プロトコル変換部により上位アプリケーションと物理層との間が接続される。パラメータ抽出部により、上位アプリケーションから送られた通信開始要求データから所定のパラメータが抽出され、ヘッダ付加部により、上位アプリケーションから送られた送信データに抽出されたパラメータを含むヘッダが付加される。そして、物理層により、ヘッダが付加された送信データが送信される。従って、通信開始要求より必要最小限のパラメータを抽出してヘッダとして付加することで、転送速度を高速化することが可能になるとともに、データを受信した機器側において、パラメータ解析に要する負荷を大幅に軽減することが可能となる。

20

【0011】

また、前記プロトコル変換部は、前記送信データを所定のデータ量毎に分割するデータ分割部を含み、前記ヘッダ付加部は、前記データ分割部による分割で得られた複数の送信データに対して1つの前記ヘッダを付加するものであってもよい。かかる構成によれば、通信の大部分を占める送信データに、細かいヘッダ情報を1度にまとめることで、転送効率を高めることが可能となる。

30

【0012】

また、前記プロトコル変換部は、前記物理層を介して他の通信相手から受信した受信データから前記パラメータを取得するパラメータ取得部を含むものであってもよい。かかる構成によれば、受信データからパラメータを取得することで、パラメータに基づいて受信データに関する各種情報を取得することが可能となる。

【0013】

また、前記パラメータに基づいて、前記受信データを記憶領域の所定位置に書き込むデータ書込部を備えるものであってもよい。かかる構成によれば、パラメータから受信データの記録位置を取得することで、受信データ全てを解析することなく受信データを所定の位置に書き込むことができる。

40

【0014】

また、前記パラメータは、データの送信方向を示すエンドポイント情報、データのサイズを示すサイズ情報を含むものであってもよい。かかる構成によれば、パラメータに基づいてデータの転送方向又はサイズ情報を取得することが可能となる。

【0015】

また、上記課題を解決するために、本発明の別の観点によれば、通信装置同士が通信を行う通信システムであって、前記通信装置は、通信相手となる通信装置の間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備え、前記プロトコル変換部は、前記上位アプリケーションから送られた通信開始

50

要求データから、所定のパラメータを抽出するパラメータ抽出部と、前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加するヘッダ付加部と、を含み、前記ヘッダが付加された前記送信データを、前記物理層を介して送受信する通信システムが提供される。

【0016】

上記構成によれば、通信装置同士が通信を行う通信システムにおいて、通信装置では、物理層により他の通信相手との間で信号の送受信が行われ、プロトコル変換部により上位アプリケーションと物理層との間が接続される。また、パラメータ抽出部により、上位アプリケーションから送られた通信開始要求データから所定のパラメータが抽出され、ヘッダ付加部により、上位アプリケーションから送られた送信データに抽出されたパラメータを含むヘッダが付加される。そして、物理層により、ヘッダが付加された送信データが送信される。従って、通信開始要求より必要最小限のパラメータを抽出してヘッダとして付加することで、転送速度を高速化することが可能になるとともに、データを受信した機器側において、パラメータ解析に要する負荷を大幅に軽減することが可能となる。

10

【0017】

また、上記課題を解決するために、本発明の別の観点によれば、他の通信相手との間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備える通信装置における通信方法であって、前記上位アプリケーションから送られた通信開始要求データから、所定のパラメータを抽出するステップと、前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加するステップと、前記ヘッダが付加された前記送信データを、前記物理層を介して送信するステップと、を備える通信方法が提供される。

20

【0018】

上記構成によれば、他の通信相手との間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備える通信装置における通信方法において、上位アプリケーションから送られた通信開始要求データから所定のパラメータが抽出され、上位アプリケーションから送られた送信データに抽出されたパラメータを含むヘッダが付加され、ヘッダが付加された送信データが物理層を介して送信される。従って、通信開始要求より必要最小限のパラメータを抽出してヘッダとして付加することで、転送速度を高速化することが可能になるとともに、データを受信した機器側において、パラメータ解析に要する負荷を大幅に軽減することが可能となる。

30

【0019】

また、上記課題を解決するために、本発明の別の観点によれば、他の通信相手との間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備える通信装置におけるプログラムであって、前記上位アプリケーションから送られた通信開始要求データから、所定のパラメータを抽出する手段、前記上位アプリケーションから送られた送信データに前記パラメータを含むヘッダを付加する手段、前記ヘッダが付加された前記送信データを、前記物理層を介して送信する手段、としてコンピュータを機能させるためのプログラムが提供される。

40

【0020】

上記構成によれば、他の通信相手との間で信号の送受信を行う物理層と、上位アプリケーションと前記物理層との間を接続するプロトコル変換部とを備える通信装置におけるプログラムにおいて、上位アプリケーションから送られた通信開始要求データから所定のパラメータが抽出され、上位アプリケーションから送られた送信データに抽出されたパラメータを含むヘッダが付加され、ヘッダが付加された送信データが物理層を介して送信される。従って、通信開始要求より必要最小限のパラメータを抽出してヘッダとして付加することで、転送速度を高速化することが可能になるとともに、データを受信した機器側において、パラメータ解析に要する負荷を大幅に軽減することが可能となる。

【発明の効果】

【0021】

50

本発明によれば、上位アプリケーションとの通信の互換性を確保するとともに、データ転送効率を向上することができ、且つ処理負荷を最小限に抑えることが可能となる。

【発明を実施するための最良の形態】

【0022】

以下に添付図面を参照しながら、本発明の好適な実施の形態について詳細に説明する。なお、本明細書及び図面において、実質的に同一の機能構成を有する構成要素については、同一の符号を付することにより重複説明を省略する。

【0023】

本実施形態の無線通信システムは、一对の機器の間でデータを送受信することを目的とした通信方式であり、近距離の機器間で無線によりデータの送受信を行う。図1は、本実施形態の無線通信システムを構成する2つの機器（通信装置）を示す模式図である。2つの機器は、それぞれレスポnder(Responder)とイニシエータ(Initiator)という役割を有する。イニシエータは「接続要求を出す側」であり、レスポnderは「接続要求を受ける側」であり、本実施形態では1対1(P2P)の通信が行われる。接続の際、イニシエータは接続要求を出し、レスポnderは持ち受け状態となるが、両者は接続の際の役割が異なるのみで、接続に係る機器の構成は同一である。イニシエータとしては例えばパーソナルコンピュータ、携帯機器、電子カードなどが該当し、レスポnderとしてはパーソナルコンピュータ、携帯機器、電子カードなどの機器が該当する。

10

【0024】

図1では、本実施形態の各機器のそれぞれが備える物理層を介して無線通信が行われる様子を模式的に示している。本実施形態では、物理層としてJET物理層と称されるものを例示するが、物理層はこれに限定されるものではなく、通信用の汎用的な物理層に適用することができる。JET物理層は、後述するプロファイルID、CSDU等を用いることで写真、動画などの大容量のデータ通信に特に適したものである。また、本明細書において、イニシエータ、レスポnderの双方の機器を総称してJETデバイス（または単にJET）と称する場合がある。

20

【0025】

図2は、本実施形態に係る無線通信システムにおいて、イニシエータ、レスポnderの各機器の構成を階層構造として示した模式図である。図2に示すように、本実施形態では、上層から順にユーザアプリケーション(User Application)100、PCL(Protocol Conversion Layer)102、DTL(Data Transfer Layer)104、CNL(Connection Layer)106、物理層(Physical Layer)108が構成されている。

30

【0026】

ユーザアプリケーション100は、本実施形態による近距離無線通信が可能な物理層108を搭載する機器において、物理層108の上層のソフトウェアの提供するサービスを用いて、データ通信を行うための上位プロトコル（例えばUSB、TCP/IP、OBEXなど）や、UI(User Interface)等のJETを含めた機器操作を行うアプリケーション（例えばウィンドズ(Windows(登録商標))、リナックス(Linux)などのOS)が該当する。JETデバイスでは、これらの上位プロトコル、またはユーザアプリケーションについては特に規定されるものではなく、機器を構成するユーザ（メーカ）が自由に設定することができる。従って、各機器は、複数の上位プロトコル、またはユーザアプリケーション100を備えていても良い。

40

【0027】

PCL102（プロトコル変換部）は、機器を構成するユーザが使用する任意のプロトコル（USB、TCP/IP、OBEX等）を、JET独自のプロトコルに相互変換する、プロトコル変換(Protocol Conversion)機能をサポートする。これにより、複数の種類のプロトコルをJETの物理層(PHY Layer)108に流すことで、様々なプロトコルをサポートすることが可能である。なお、同じUSBであっても、Windows、LinuxなどのOSの違いによってプロトコル変換が異なる場合がある。PCL102は、上位のユーザアプリケーション100が生成する音声、映像等のコンテンツデータ、その他のプロトコルのデータ、コマンド等を

50

下位のDTL104が扱うことが可能なデータ形式に変換する処理を行う。また、PCL102は、接続、切断、機器認証、動作モード設定、初期化等のJETの通信に必要な処理を行う。

【0028】

図3は、イニシエータ、レスポnderにおけるデータの流れを示す模式図である。図3に示すように、ユーザアプリケーション100はJETによる接続と、データ転送の2種類の制御を行うことになる。JETとしては最上位PCL102でこれらの機能を実現するために必要なサービスを提供し、JET独自のプロトコルへの変換と、接続管理を行う。さらにJET規格に準拠したCSDU (CNL service data unit)を生成するDTL104、CNL106への受け渡しを行う。

10

【0029】

DTL104は、上位のPCL102から受け取ったデータを、所定のケット構造に整形し、下位のCNL106が提供するサービスを用いて、イニシエータ、レスポnder間の送信を行う。また、受信においては、CNL106が受信したデータを解析し、CSDUを抽出し、そのペイロードを上位のPCL102に引き渡す。CSDUには、物理層(PHY Layer)108による通信以外のユーザアプリケーション100で利用可能なステータス情報も含まれており、DTL104は、これらの生成処理、エラー通知等も行う。

【0030】

DTL104は、上位プロトコルの種別に関わらず、上位から入力されたデータをDTLケットに整形して下位のCNL106に渡し、下位からの受信データからDTLケットを抽出し、上位にDTLケットペイロードを受け渡すことが可能である。ただし、DTL104自身は、PCL102からの異なるprotocolから送られてくるデータを受け入れることが可能ではあるが、JETでは異なるプロトコルのデータの送受信は一度セッションの切断を必要とするため、複数プロトコルでのDTLサービスの利用は行わない。

20

【0031】

この制限により、後述する複数のPCLエミュレーションからDTL104へデータの入力が行われたとしても、DTL104はそのデータのMuxを行うことはない。また、CNL106からの受信データに複数のプロトコルが含まれていた場合であっても、そのプロトコルの解析、それぞれのプロトコル内容に合わせたPCL102への配信、またはエラー検知によるセッションの切断等の処理は行わない。

30

【0032】

このため、DTL104によるサービスを利用するPCL102側では、必ず利用するプロトコルを1種類に確定した状態でDTL104によるサービスを利用する必要がある。これらのプロトコル方式を確定させるための判断と、必要な送受信を行うのは後述するPCLコモンの役割であり、プロトコルデータの生成、パースはPCLエミュレーションが行う。複数のプロトコルから同時にDTLサービスを利用できないよう排他処理もPCLコモンの役割である。

【0033】

DTL104はPCLコモンが接続を確立するのに必要なサービス、接続確立後にPCLエミュレーションがデータの送受信を行うのに必要なサービスを提供する。

40

【0034】

また、DTL104は、現在実行されているサービスが、全転送サイズの途中データなのか、最後のデータなのか、もしくは、データではなくパラメータなのかを示すプロファイルID(Profile ID)、サイズをPCL102よりパラメータとして受け取り、下位のCNLサービスを利用して生成するCSDUケットヘッダに挿入する。DTL104は、送信パラメータを、JETがデータを送信する際に生成するCSDUケットの一部に埋め込むことで図6のような複数の論理チャネル(Channel)を1つの物理層(PHY Layer)108上で実現する。

【0035】

DTL104は、JET規格に定義されているCSDUケットを生成する機能を有する。D

50

T L 1 0 4 では、CSDUパケットの種別を理解するためのパラメータをCSDUパケットヘッダに付加する。付加するものはProfile ID , Size , Data Payloadである。

【 0 0 3 6 】

D T L 1 0 4 はCNL 1 0 6 が提供するCSDUの単位でデータ転送を行う。DTL 1 0 4 はCSDU送信時に以下の3種類のプロファイルID (T_DATA, LT_DATA, CNL_DATA)をCSDUに対して付与する。さらに、CSDU受信時は、プロファイルIDの種類に応じた処理を行う。

【 0 0 3 7 】

T_DATA, LT_DATA

DTL 1 0 4 は、ユーザデータを転送するCSDUに対して、T_DATAを付与する。ただし、CSDUペイロードへの分割において、最終のCSDUとなる場合にはLT_DATAを付与する。CSDUのペイロードには、ユーザデータのみが格納され、DTL 1 0 4 がヘッダ情報などを埋め込むことはない。

【 0 0 3 8 】

CNL_DATA

D T L 1 0 4 は、JETシステム固有の制御データを転送するCSDUに対して、CNL_DATAを付与する。制御データの例としては、パラメータ情報などがある。CSDUペイロードにはヘッダ情報(詳細はTBD)が埋め込まれる。D T L 1 0 4 はこのヘッダ情報を解釈し、適切な処理を行う。

【 0 0 3 9 】

C N L 1 0 6 は、上位のD T L 1 0 4 の要求に応じて、物理層1 0 8 のサービスを利用した通信を行う他、物理層1 0 8 の接続の確立、切断、データの連続性の保障などを行う。

【 0 0 4 0 】

物理層1 0 8 は、本実施形態による近距離大容量通信が可能な無線通信システムのJ E T物理層であり、誤り訂正機能、プリアンブルセンス(preamble sense)機能を含む。

【 0 0 4 1 】

図4は、J E Tデバイスを搭載する機器のソフトウェアの役割に基づいて、図2の構成をO S I参照モデルで示したものである。図4に示すように、物理層(第1層)1 0 8 は、データを通信回線に送出するための電気的な変換や機械的な作業を受け持つ。ピンの形状やケーブルの特性なども第1層で定められる。

【 0 0 4 2 】

D T L 1 0 4 、C N L 1 0 6 は、データリンク層(第2層)、トランスポート層(第4層)に対応する。データリンク層は、通信相手との物理的な通信路を確保し、通信路を流れるデータのエラー検出などを行う。また、トランスポート層は、通信相手まで確実に効率良くデータを届けるためのデータ圧縮や誤り訂正、再送制御などを行う。なお、本実施形態のシステムはP2P通信であるため、O S I参照モデルにおけるネットワーク層(第3層)は設けられておらず、システムを簡略化することができる。

【 0 0 4 3 】

P C L 1 0 2 は、セッション層(第5層)とプレゼンテーション層(第6層)が対応する。セッション層は、通信プログラム同士がデータの送受信を行うための仮想的な経路(コネクション)の確立や解放を行う。プレゼンテーション層は、セッション層から受け取ったデータをユーザが分かり易い形式に変換したり、アプリケーション層から送られてくるデータを通信に適した形式に変換するなどの処理を行う。

【 0 0 4 4 】

ユーザアプリケーション1 0 0 は、アプリケーション層(第7層)に対応する。アプリケーション層は、データ通信を利用した様々なサービスを人間や他のプログラムに提供する。

【 0 0 4 5 】

次に、本実施形態の通信装置におけるデータの流れを説明する。図5は、データフローを示す模式図であって、JET機器内の各レイヤーにおけるファイル、データの送受信のデ

10

20

30

40

50

ータフローを示している。なお、PCL102は、PCLコモンとPCLエミュレーションに機能が分かれる。データ転送で利用するのはPCLエミュレーションであるため、図5に示すPCL102による処理はPCLエミュレーションによって実現される機能である。物理層108に入力されるCSDUは、データフォーマットとして規定されており、そのヘッダ情報等の生成、解析を行うDTL104が扱うデータフォーマットも同様である。

【0046】

また、後述するように、共通機能を提供するためのPCLコモンについては規定されているが、PCLエミュレーションは、ユーザプロトコルに準じたデータ変換処理を行うため、それぞれのプロトコルに応じたシステム仕様に依存する。

【0047】

JET通信では、ファイル等のデータだけではなく、PCL102、DTL104内での管理パラメータや、通信先の同一レイヤー間でのデータの送受信が存在する。これらのファイル、パラメータ類は、CNL106によって最終的にCSDUフォーマットに準拠した形式で伝送される。図6は、CSDUによる論理的なチャンネルを示す模式図である。図6に示すように、データの種別を特定するにはProfile IDを用いる。これにより、物理層108のレベルで、複数の伝送Channelを論理的に用いることが可能となる。従って、通信レートを大幅に向上させることができ、特に動画などの大容量のデータ通信に適している。

【0048】

図7は、CSDUがマッピングされる様子を示す模式図である。CSDUは、CNL106とDTL104の間でやり取りされるデータユニットであり、図7に示すように、CNLフレームにマッピングされる。ユーザアプリケーション100が送受信するユーザデータサイズは特に規定されない。PCL102は、データの長さがデータ分割長（最大4096バイト）を超えた場合、複数のCSDUペイロードへ分割する。PCL102はCSDUペイロードの単位で、DTLサービス呼び出してユーザデータの送受信を行う。DTL104はCSDUペイロードにヘッダを追加して下位のCNL106に渡す。CSDUヘッダはProfile IDとCSDUペイロードの長さを示すLengthで構成される。

【0049】

図8は、本システムの機器のハードウェア構成を示す模式図である。図5に示すように、イニシエータとレスポンスのそれぞれは、物理層108を構成するチップ200と、CPU210とを有して構成される。物理層108は、ベースバンド部を含んでいる。上述したユーザアプリケーション100、PCL102、DTL104、CNL106は、ソフトウェア（プログラム）によりCPU210を機能させることによって実現される。ソフトウェアは、イニシエータ、レスポンスを構成する通信装置が備えるメモリ、または通信装置の外部の記録媒体などに格納される。

【0050】

図8に示すように、PCL102は、通信開始要求データ（CBW）から、エンドポイント情報、データサイズなどの所定のパラメータを抽出するパラメータ抽出部102c、送信データに抽出したパラメータを含むヘッダを付加するヘッダ付加部102d、送信データを4Kバイト毎に分割するデータ分割部102e、受信データからパラメータを取得するパラメータ取得部102fを備えている。これらの機能ブロックは、主として後述するPCLエミュレーション102bによって実現される。

【0051】

図9は、各レイヤーが提供するサービスのアクセスポイントと、レイヤー間の関係を示す模式図である。PCL102の上位は、ユーザアプリケーション100となる。PCL102は、下位DTL104を利用したサービスを提供するレイヤーである。PCL102は、上位のユーザアプリケーション100に対しては、制御をPCLコモン102a（共通処理部）が行い、データ転送はPCLエミュレーション102b（変換処理部）が行うというように、役割が分かれるため、PCL102のサービスはそれぞれに対して規定される。

【0052】

10

20

30

40

50

PCLコモン102aによるサービスは、ユーザアプリケーション100の要求に応じて、DTL104の接続/切断/その他の制御のサービスを呼び出すことで下記のサービスを提供する。

- ・接続、切断などの制御サービス
- ・エラーなどのイベント通知サービス
- ・エミュレーション制御サービス

【0053】

PCLエミュレーション102bによるサービスは、対応するプロトコルごとに個別に存在する。個々のPCLエミュレーションはCSDUのペイロード上に汎用プロトコル(USB、TCP/IP、OBEX等)のコマンド、データを乗せて通信することを可能にするプロトコルサービスである。

10

【0054】

PCL102では、PCLエミュレーションサービスにより選択されたプロトコル方式に該当するサービスのみ起動することが許可される。PCLエミュレーションサービス内部では上位プロトコルの要求にしたがって、DTL104によるサービスを利用するためのCSDUペイロードを生成する。PCLエミュレーション102bによるサービスを複数持つことで、1つのJET機器で複数のエミュレーションサービス(Emulation Service)を実現することが可能になる。PCLコモン102aにより、1度のセッションで利用できるエミュレーションサービスは1種類のみであるように管理される。

【0055】

図9に示すように、PCL102は、PCLコモン(PCL Common)102aと、PCLエミュレーション(PCL Emulation)102bにその機能が分割される。PCLコモン102aは、上位のユーザアプリケーション100の要求によって、下位レイヤのサービスの初期化や、接続、切断等の基本機能を提供する。PCLコモン102aでは基本機能の処理を行うため、どのプロトコルが選択された場合においても同様の処理が行われる。一方、PCLエミュレーション102bは、PCLコモン102aにより起動が完了した後、ユーザアプリケーション100が有する任意のプロトコルを下位のDTL104、CNL106が扱うプロトコル形式に変換する。

20

【0056】

上述のように、PCLコモン102aは、初期化、基本通信(接続、切断、機器認証)等の共通機能サービスをユーザアプリケーション100に対して提供する。PCLコモン102aは、全てのJETデバイスで共通に設けられるソフトウェアである。従って、PCL102は、PCLエミュレーション102bのみの構成では動作することができない。

30

【0057】

PCLエミュレーション102bは、PCLコモン102aによって接続が行われた後、ユーザデータ転送を行うもので、ユーザプロトコル(USB、TCP/IP、OBEX等の汎用プロトコルデータ)をDTL104が扱うデータ形式に相互変換する役割を持つ。PCLエミュレーション102bは、ユーザアプリケーション100から送られてくるユーザプロトコルデータを、下位のDTL104が解釈できる形式に変換する役割を有する。PCL102内のエミュレーションブロック(PCLエミュレーション102bの変換モジュール)は、ユーザアプリケーション100から見た場合に、既存のUSB MSC、NFC等のデバイスを制御するのと同様の方式でデータ転送機能を提供するためのサービスを提供する。但し、PCLエミュレーション102bは、機器を構成するユーザー固有のプロトコル数だけ存在する。

40

【0058】

DTL104は、上位の2種類のPCL(PCL Common102a、PCL Emulation102b)に対して、下位CNL106のサービスを利用した機能を、DTLサービスとして提供する。PCLエミュレーション102bは、後で詳細に説明するように、ユーザプロトコルごとに変換モジュール(Protocol A, Protocol B, Protocol C,・・・Protocol Z)を

50

有するが、1度のセッション(接続)で利用できるのは1種類だけであり、その制御はPCLコモン102aによって行われる。例えば、上位プロトコルがUSBの場合、マスタートレイジクラスであるか、あるいは他の方式かによって異なる変換モジュールが用意されている。

【0059】

JETデバイスにおいて、機器を構成するユーザは、上位のプロトコルに対応した変換モジュールを自由に設定してPCLエミュレーション102bを構築することができる。また、変換モジュールの追加、削除もユーザが自由に行うことができる。一方、PCLコモン102aはプロトコル変換の基本機能であるため、全てのJETデバイスにおいて共通であることが義務付けられる。

10

【0060】

図9では、ユーザプロトコルとしてProtocol A~Zが示されており、このうちProtocol Bがアクティブとされ、Protocol Bにより接続が行われている状態を示している。この場合、イニシエータとレスポンドの双方でProtocol Bによる接続が行われる。どのプロトコルで接続するかは、イニシエータとレスポンドとの間のネゴシエーションによって決定される。

【0061】

図10は、本実施形態のシステムにおける状態遷移を示す模式図である。PCL102は、物理層108によるイニシエータとレスポンドの接続状態の変化や、ユーザアプリケーションからPCL Serviceを利用することにより、図10に示すような状態遷移を行う。

20

【0062】

図10において、先ず、レスポンドはイニシエータからの接続待ち状態とされ、イニシエータは接続先のレスポンドをサーチしている状態とされる。イニシエータとレスポンドの接続が開始されると(Start Connection)、イニシエータとレスポンド間でネゴシエーション(Negotiation)が行われる。この状態では、JETデバイス間でソフトウェアのバージョンの確認、エミュレーション方式(互いにどのようなプロトコルを有しているか)の確認が行われる。

【0063】

ネゴシエーションの結果、バージョン、エミュレーション方式が一致した場合は、接続が行われ、物理層108による接続が完了する(Connected)。その後、エミュレーションが開始され(Emulation)、ユーザアプリケーション100間でデータ転送が可能な状態とされる。一方、ソフトウェアのバージョン情報が一致しなかった場合、または双方が保有するプロトコルが一致せず、エミュレーション方式が一致しなかった場合は、接続が行われない(Disconnect)。接続がされなかった場合(Disconnect)、エミュレーションが終了した場合(End Emulation)は、レスポンドが接続待ち状態となる。

30

【0064】

PCLコモン102aは、PCL102のバージョン(Version)情報に基づいて、バージョンチェック(Version Check)及びエミュレーション方式の判別を行うネゴシエーション機能を備える。ネゴシエーションの結果、バージョン及びエミュレーション方式が一致すれば、イニシエータとレスポンドの間で同じプロトコルによる接続が行われる。ネゴシエーションに必要なバージョン管理機能(PCL Version Management)及び、エミュレーション判別機能(PCL Select Emulation)については後で説明する。ネゴシエーションはユーザアプリケーション100に対して提供されるサービスではなく、接続待機中の状態で、接続を検知した際に自動で実行される内部機能である。

40

【0065】

図11は、ネゴシエーションの処理を示す模式図である。図11の処理は、図2の各層のソフトウェアによりCPU210を機能させることで実現できる。一例として、判別自体はレスポンドが行い、イニシエータはその判別結果を待つ。接続先のバージョン情報

50

の取得はCNL106が行い、接続時にバージョン情報が自動的に交換される(ステップS1)。このとき交換されるバージョン情報をJETバージョンと称することとする。

【0066】

PCL102では、下位のCNL106、DTL104からのイベントにより、接続を検知した時点で接続先JETデバイスのJETバージョンの情報を既に取得できている。このため、先ず、図11に示すように、CNL106のソフトウェアのバージョンチェック(CNL Ver check)とDTLのソフトウェアのバージョンチェック(DTL Ver check)が行われる。PCL102の内部で行われるのは、JETバージョン内に含まれるPCLバージョンのチェックによるエミュレーション方式の判別である。

【0067】

エミュレーション方式の判別は、例えばレスポnderが主導して行う。イニシエータとレスポnderの双方でPCL102のバージョン情報が交換され、PCLコモン102aのPCLバージョンチェック機能により、PCL102のソフトウェアのバージョン情報がチェックされる(PCL Ver check)。

【0068】

次に、イニシエータとレスポnderの双方でエミュレーションタイプの情報が交換され、図11に示すように、PCLコモン102aのエミュレーションタイプチェック機能により、エミュレーションのタイプがチェックされる(EMU Type check)。エミュレーションタイプは、互いのJETデバイスが通信可能なエミュレーション方式(プロトコル)を記述したパラメータである。

【0069】

レスポnder側では、イニシエータとレスポnderの互いのエミュレーションタイプの比較を行い、同一のものがあれば接続可能であると判断する。PCL102は、エミュレーションタイプが確定した時点でユーザアプリケーション100に通知を行い、ユーザアプリケーション100がPCL_start_emu serviceを呼び出す(ステップS2)。そして、PCLコモン102aからPCLエミュレーション102bへStart_Emuというコマンドを送る。これによりPCLコモン102aによる起動が完了し、PCLエミュレーション102bによるエミュレーションが開始される(ステップS3)。そして、PCLエミュレーション102bによりユーザアプリケーション100のプロトコルを変換して、下位のDTL104、CTL106と通信を行うことが可能となる。

【0070】

また、イニシエータとレスポnderで同一のエミュレーションタイプを複数有している場合は、ユーザアプリケーション100にその旨を通知する。ユーザアプリケーション100側でこれらの複数のエミュレーションタイプの1つを指定する場合は、その旨の情報がPCL102に送られる。この際、ユーザアプリケーション100側では、予め指定された1つのエミュレーションタイプを指定することができる。また、イニシエータとレスポnderの一方が携帯機器の場合など、高速通信可能なプロトコルを使う必要が比較的低い場合は、通信速度に応じた適切なプロトコルのエミュレーションタイプを選択することができる。これらの仕様は、JETデバイス、またはユーザアプリケーション100を構成するユーザが自由に設定することができる。

【0071】

次に、PCL102によるエミュレーション選択について説明する。エミュレーション選択は、イニシエータからの接続検知を検知したレスポnder側でネゴシエーションを行う際にPCLコモン102aの内部で実行される機能である。接続時にJETデバイス間で交換されるJETバージョン内のPCLバージョン情報から、互いに適合するエミュレーションを持っているかを確認する。

【0072】

図12は、JETバージョン情報を説明するための模式図である。PCL102は、自機のバージョン情報と、接続先JETデバイスのバージョン情報の2つを管理する。自機のバージョン情報は必ず起動時にロードされ、接続検知時、接続中は、接続先デバイスのバ

10

20

30

40

50

ージョン情報を保持する。

【 0 0 7 3 】

図 1 2 に示すように、J E T のバージョン情報は合計 1 0 バイトであり、上層から順にプラットフォーム(Platform)情報(1byte)、J E T ドライババージョン情報(2byte)、C N L バージョン情報(1byte)、D T L バージョン情報(2byte)、P C L バージョン情報(2byte)、リザーブド(2byte)の情報がある。

【 0 0 7 4 】

P C L バージョン情報(2byte)のうち、前半の1バイトは、システムの互換性維持のためのソフトウェアのVersion No (T.B.D)である。後半の1バイトは、装置(JETデバイス)が対応するエミュレーション方式を示している。図 1 2 では、エミュレーション方式としてU S B , T C P / I P , O B E X . . . が例示され、各方式に1ビットのデータが与えられている。そして、ビットが1の場合はそのエミュレーション方式に対応することを表しており、ビットが0の場合はそのエミュレーション方式に対応していないことを表している。1つのJETデバイスで対応するエミュレーション方式の最大値には規定はないが、最低でも1つのエミュレーション方式には対応しなければならない。

10

【 0 0 7 5 】

なお、バージョン情報のチェックは、図 1 1 に示すように、図 1 2 の上層側(プラットフォーム側)の情報から順次に行われる。P C L 1 0 2 のバージョン情報のチェック後、エミュレーションタイプのチェックが行われる。

【 0 0 7 6 】

接続の実行結果として、互いに通信可能なエミュレーション方式を記述したエミュレーションタイプが選択され、選択されたエミュレーションタイプは、レスポnder側からユーザアプリケーション1 0 0 に通知され、また許可待ちをしているイニシエータに通知される。図 1 3 は、エミュレーション選択のシーケンスを示す模式図である。図 1 3 において、選択されたエミュレーションタイプ(EMUTYPE)は、レスポnderのユーザアプリケーション1 0 0 (UsrApp)に対してはPCL_conf_r.indとして、イニシエータ側に対してはPCL_conf_i.indとして通知される(図 1 3 参照)。

20

【 0 0 7 7 】

図 1 4 及び図 1 5 は、P C L エミュレーション1 0 2 b に関するサービスを示す模式図である。以下、各サービスについて説明する。起動、終了はP C L コモン1 0 2 a によって行われるが、データの送信、受信は、ユーザアプリケーション1 0 0 からP C L エミュレーション1 0 2 b に通信を行うことで実行できる。従って、P C L コモン1 0 2 a との間で提供されるサービス(図 1 4)と、ユーザアプリケーション1 0 0 との間で提供されるサービス(図 1 5)に分けられる。

30

【 0 0 7 8 】

Start service (Mandatory ; 図 1 4)

Start serviceは、標準で提供されるサービスであり、エミュレーションを開始する際、P C L コモン1 0 2 a により実行されるP C L エミュレーション1 0 2 b の初期化処理を提供するサービスである。Startが完了した時点で、ユーザアプリケーション1 0 0 から、ユーザプロトコルを用いたデータ送受信が可能となる。

40

【 0 0 7 9 】

End service (Mandatory ; 図 1 4)

End serviceは、標準で提供されるサービスであり、エミュレーションを終了する際、P C L コモン1 0 2 a により実行されるエミュレーションの終了処理を提供するサービスである。Endが完了した時点で、ユーザアプリケーション1 0 0 から、ユーザプロトコルを用いたデータ送受信は不可能となる。P C L コモン1 0 2 a はエミュレーションがStartされていた場合、切断(PCL_Disconnect)を実行する前に、このサービスを実行する。

【 0 0 8 0 】

Open service (Mandatory ; 図 1 5)

Open serviceは、ユーザプロトコル上での通信路のオープン時に必要な処理を提供する

50

サービスである。

【 0 0 8 1 】

Close service (Mandatory ; 図 1 5)

Close serviceは、ユーザプロトコル上での通信路のクローズ時に必要な処理を提供するサービスである。

【 0 0 8 2 】

Read service (Mandatory ; 図 1 5)

Read serviceは、ユーザプロトコル上で、接続先のデータを取得する際に必要な処理を提供するサービスである。

【 0 0 8 3 】

Write service (Mandatory ; 図 1 5)

Write serviceは、ユーザプロトコル上で、接続先にデータを送信する際に必要な処理を提供するサービスである。

【 0 0 8 4 】

以上のように、Open service、Close serviceは、P C L 1 0 2 による上位プロトコルの初期化処理に対応する処理である。また、Read service、Write serviceは、ユーザアプリケーション 1 0 0 によるデータの送信、受信に関する処理である。

【 0 0 8 5 】

User customize service (Option ; 図 1 5)

上述のサービスに該当しないサービスであり、装置を構成するユーザが独自のサービスとして定義できるものである。通信路を“開く(Open)”、“閉じる(Close)”、“データを送る(Write)”、“受ける(Read)”以外の部分は、J E T デバイスを構成するユーザが自由に設定できるカスタマイズ領域とされ、例えばユーザアプリケーション 1 0 0 の種類 (Windows, Linux等) に応じて、ユーザが自由に設定することができる。但し、どのアプリケーションを使用した場合でも、上述のようにP C L コモン 1 0 2 a による“Start”、“End”は必要であり、全てのJ E T デバイスで共通である。

【 0 0 8 6 】

以上のように、本実施形態では、ユーザアプリケーション 1 0 0 のプロトコルを変換するP C L 1 0 2 を設け、各通信装置に共通に設けられたP C L コモン 1 0 2 a により、ユーザアプリケーション 1 0 0 の1又は複数のプロトコルの中から通信相手のプロトコルと一致するプロトコルが選択される。そして、選択されたプロトコルが物理層 1 0 8 で通信を行うためのプロトコルに変換されるため、通信相手のユーザアプリケーション 1 0 0 のプロトコルに応じて通信に最適なプロトコルを選択することが可能となり、通信の互換性を確保することができる。

【 0 0 8 7 】

次に、図 1 6 に基づいて、上位アプリケーションがU S B の場合に、U S B マスストレージクラスのプロトコル上で、イニシエータ側のP C L 1 0 2 (主にP C L エミュレーション 1 0 2 b) で必要なデータをあらかじめ抽出し、効率的なデータの転送を実現する処理について説明する。図 1 6 は、ネゴシエーションが行われた後の処理を示しており、図中に示す“変換”の処理は、主としてP C L エミュレーション 1 0 2 b で行われる。そして、図 1 6 は、U S B 以外の物理層、すなわち本実施形態の通信装置の物理層 1 0 0 で扱うデータ形式に対して、U S B マスストレージのデータ形式を本実施形態に係るJ E T デバイスの通信方式に変換してマッピングする手法を示している。

【 0 0 8 8 】

なお、図 1 6 において、ヘッダ、データは個別のパケットとして送信されるものであり、ヘッダとデータが同じパケットで送信されることはない。また、U S B で転送されるデータはU R B という単位で扱われる。これは、送信側、受信側それぞれの中でのデータを扱う単位なので、U R B 自体の設定値などの互換性は無い。U R B はホスト側で扱うデータサイズによって変わるが、同等のサイズ情報がC B W に記載される。更に、コマンド、データの向きに関するエンドポイント番号 (E P) もC B W に記述される。

10

20

30

40

50

【 0 0 8 9 】

図 1 6 において、U S B ホスト (USBHOST) は、イニシエータの上位アプリケーションを示している。また、U S B ターゲット (USBTGT) は、レスポnder の上位アプリケーションを示している。

【 0 0 9 0 】

図 1 6 では、イニシエータがレスポnder へデータを送り、レスポnder のメモリ、ハードディスク等にデータを書き込む場合 (W r i t e 動作) を示している。図 1 6 に示すように、書き込み用のデータを送る前に、U S B ホスト側から U S B ターゲット側へ通信要求を示す C B W が送られる (ステップ S 3 1) 。 C B W は、C B W 自体のエンドポイントとサイズを示すパラメータ (P A R A M) 、及び C B W のデータ自体から構成される。なお、図中に示す各パラメータは、U S B データを送る前に予めどのようなデータであるかを示すエンドポイント (E P) 番号と、次に送るデータの総量から構成される。エンドポイント番号が 2 の場合は、U S B ホストから U S B ターゲットへ送られるデータであることを示し、エンドポイント番号が 1 の場合は U S B ターゲットから U S B ホストへ返信されるデータを示している。

10

【 0 0 9 1 】

U S B ターゲットは、C B W を受信すると、C B W に付加されたパラメータに基づいてサイズとエンドポイントを認識し、C B W のデータ自体には M S C (Mass Strage Class) 処理が行われる (ステップ S 3 2) 。これにより、U S B ターゲット側では、C B W により次に書き込み用のデータが送られてくることを認識でき、データの待ち受けを開始する。

20

【 0 0 9 2 】

また、U S B ホスト側では、C B W のデータから書き込み用データのエンドポイントと書き込み用データの総サイズを抽出する (ステップ S 3 3) 。また、書き込み用のデータを複数に分割する (ステップ S 3 4) 。そして、C B W から抽出したエンドポイントと総サイズを複数の書き込み用のデータに付加して U S B ターゲットに送る (ステップ S 3 5) 。C B W には、その後送られる書き込み用データに関する情報 (エンドポイント、サイズ等) が含まれる。従って、C B W から必要な情報のみを抽出して書き込み用のデータに付加することで、U S B のデータの全てを J E T デバイスの物理層 1 0 8 に合わせて変換する必要がなく、送信側の装置の処理量を最小限に抑えることができる。また、受信側の装置においても、書き込み用データに付加されたパラメータを参照することで、必要な情報を得ることができるため、送られたデータの全てを解析する必要がなく、処理の負荷を最小限に抑えることが可能である。

30

【 0 0 9 3 】

U S B ホスト側は、データの送信後、U S B ターゲット側からの完了通知待ちうけ状態に遷移する。U S B ターゲット側では、書き込み用のデータを受信すると、付加されているエンドポイントと総サイズを認識し、書き込み用のデータには M S C 一括処理が施される (ステップ S 3 6) 。M S C 一括処理が施されたデータは、エンドポイント、サイズと結合されて上位アプリケーションの記憶領域、またはメモリ等へのデータの書き込みが行われる。

40

【 0 0 9 4 】

U S B ターゲット側での書き込みが完了すると、U S B ターゲット側から U S B ホスト側へ完了通知である C S W が送られる。C S W は、C S W 自体のエンドポイントとサイズを示すパラメータ、及び C S W のデータ自体から構成される。完了通知待ち受け状態の U S B ホストが C S W を受信すると、U S B ホストから U S B ターゲットへのデータの書き込みが完了する。

【 0 0 9 5 】

以上の処理において、C B W 、C S W はデータ量としては小さいものであるので、書き込み用データの送信前に、必要最小限のデータとしての C B W を先に送ることで、待ち受け側の U S B ターゲット側で S W の準備が先行して行うことが可能となり、処理全体のバ

50

パフォーマンスを上げることができる。

【0096】

更に、USBホスト側から書き込み用のデータを送る際には、CBWから書き込み用データのエンドポイント番号、データサイズを抽出し、書き込み用データのパラメータとしてヘッダに付加する。この際、データ送信時に分割されたURBの最初のデータにだけヘッダを付加し、以降の分割URBにはヘッダを付加しない。

【0097】

これにより、転送の大部分を占めるデータ送受信時のパラメータ送信を複数回に一回のトランザクションとすることができ、変換によるパフォーマンス低下を抑止するとともに、トランザクションの増加を大幅に省くことができ、システムの負荷を軽減することができる。

10

【0098】

また、データが書き込まれるUSBターゲット側では、書き込み用のデータに付加されたエンドポイントを参照することにより、バッファ内のどの位置にデータを書き込むかを認識することができる。USBターゲット側では、接続時にUSBホスト側と互いに必要なエンドポイントの情報をやり取りする(USBのエナミュレーションに相当)。そして、この情報のやり取りに基づいて、必要なエンドポイントのバッファ領域を確保し、データのエンドポイント番号を見てどのバッファに書き込むかを判断する。従って、書き込み用データの全てを解析する必要がなく、書き込み用データのヘッダに付加されたパラメータを参照するのみで、書き込み用データを上位のユーザアプリケーション100の所定の位置に送り、記録することが可能である。

20

【0099】

図16では、イニシエータからレスポンスへの書き込み動作を示したが、USBホストからUSBターゲットへの読出し動作(図18参照)についても同様に処理することができる。読出し動作の場合は、イニシエータからレスポンスへデータの読出しを要求するCBWが送られた後、レスポンス側でCBWからエンドポイント番号、サイズ等のパラメータを抽出する。そして、レスポンス側でイニシエータに送るデータを分割して、複数の分割されたデータに対して、抽出したパラメータを含む1つのヘッダを付加して、イニシエータへ送信する。データを受信したイニシエータでは、パラメータに基づいて上位アプリケーションの所定の位置に受信データを送ることができる。

30

【0100】

図17は、図16の処理をより具体的に示した模式図であって、イニシエータ側の処理を示している。イニシエータ側のユーザアプリケーション100であるUSBホストからCBWによる書き込み開始が指示されると(ステップS41)、CBWは一旦USBバッファ(USB BUF)に書き込まれ、下位レイヤ(PC L104)の dongle SW(DONGLE SW)において図16のパラメータを含むヘッダ生成とCBW書き込みが行われ、CBWは、JET Header 64bitとData 31 Byte(CBW)に変換されて、MACバッファ(MAC BUF)に書き込まれる。MACバッファに書き込まれたヘッダ、データは、物理層108を介してUSBターゲット側へ送信(TX)される。

【0101】

次に、USBホスト側のユーザアプリケーションにより書き込みデータ送信が指示されると(ステップS42)、データは一旦USBバッファ(USB BUF)に書き込まれ、下位レイヤ(PC L104)において、CBWからヘッダが生成される。また、書き込み用データが4Kbyteのデータに分割されて、ヘッダ(JET Header 64bit)とともに複数の4Kbyteのデータ(Data 4096Byte)がMACバッファ(MAC BUF)に書き込まれる。このように、書き込み用データは、JETデバイスの物理層108に合わせて4Kbyte毎のデータに分割される。これにより、1つのデータのサイズを最大限に大きくすることができ、装置の負荷を軽減することができる。MACバッファに書き込まれたヘッダ、データは、物理層108を介してUSBターゲット側へ送信(TX)される。なお、図17では、USB物理層とJETデバイスの物理層108との変換を行う機器の例として外部機器(ド

40

50

ングルSW)を用い、PCL104が外部機器に構成され、PCL104よりも上位のユーザアプリケーションがパーソナルコンピュータ等の機器に構成される例を示している。外部機器を用いることなく、パーソナルコンピュータ等の単体機器でイニシエータを構成することも勿論可能である。ここで、ドングルSWは、USBにおけるHUB、カードリーダーなどの機器に相当する。

【0102】

USBターゲット側から受信(RX)したCSWは、レスポンス側での同様の処理によりJET Header 64bitとData 13 Byte(CSW)から構成され、MACバッファに溜められる。このデータはドングルSWにおいてヘッダが解析され、USBバッファにデータが書き込まれる。なお、CBWのデータサイズは31Byteであるが、CSWのデータサイズは13Byteである。ユーザアプリケーション100では、USBバッファのデータを受けて、書き込み要求の完了通知を出す(ステップS43)。以降のステップS44~S46の処理も、ステップS41~S43と同様に行われる。

10

【0103】

以上説明したように本実施形態によれば、USBマスタレイジプロトコルで、通信の大部分を占めるデータ送信時に、細かいヘッダ情報を1度にまとめることで、転送効率を高めることが可能となる。また、CBWより必要最小限のパラメータを抽出することで、ユーザアプリケーション100から異なる物理層108に変換されたデータを受信した機器側において、パラメータ解析に要する負荷を大幅に軽減することが可能となる。なお、上述した例では、USBのマスタレイジクラスのプロトコル変換を示したが、本発明はこれに限定されるものではなく、他のプロトコル変換においても同様の手法を用いることで高速化を達成することが可能である。また、上述した例では、無線通信システムを例に挙げて説明したが、通信システムは有線のものであってもよい。

20

【0104】

以上、添付図面を参照しながら本発明の好適な実施形態について説明したが、本発明に係る例に限定されないことは言うまでもない。当業者であれば、特許請求の範囲に記載された範疇内において、各種の変更例または修正例に想到し得ることは明らかであり、それらについても当然に本発明の技術的範囲に属するものと了解される。

【図面の簡単な説明】

【0105】

30

【図1】本実施形態の無線通信システムを構成する2つの機器を示す模式図である。

【図2】イニシエータ、レスポンスの各機器の構成を階層構造として示した模式図である。

【図3】イニシエータ、レスポンスにおけるデータの流れを示す模式図である。

【図4】図2の構成をOSI参照モデルで示した模式図である。

【図5】機器内の各レイヤーにおけるファイル、データの送受信のデータフローを示す模式図である。

【図6】CSDUによる論理的なチャネルを示す模式図である。

【図7】CSDUがマッピングされる様子を示す模式図である。

【図8】機器のハードウェア構成を示す模式図である。

40

【図9】各レイヤーが提供するサービスのアクセスポイントと、レイヤー間の関係を示す模式図である。

【図10】本実施形態のシステムにおける状態遷移を示す模式図である。

【図11】ネゴシエーションの処理を示す模式図である。

【図12】PCLバージョン情報を説明するための模式図である。

【図13】エミュレーション選択のシーケンスを示す模式図である。

【図14】PCLエミュレーションのサービスを示す模式図である。

【図15】PCLエミュレーションのサービスを示す模式図である。

【図16】USBマスタレイジクラスのプロトコル上で、イニシエータ側のPCLに必要なデータをあらかじめ抽出し、効率的なデータの転送を実現する処理を示す模式図であ

50

る。

【図17】図16の処理をより具体的に示す模式図である。

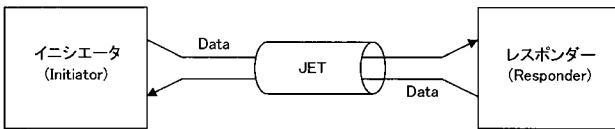
【図18】USB物理層を介してUSBホストとUSBターゲットとの間で通信が行われる様子を示す模式図である。

【符号の説明】

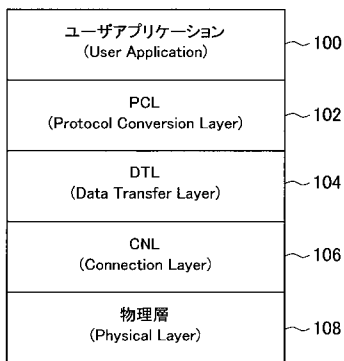
【0106】

- 100 ユーザアプリケーション
- 102 PCL
- 102c パラメータ抽出部
- 102d ヘッダ付加部
- 102e データ分割部
- 102f パラメータ取得部
- 108 物理層
- 210 CPU

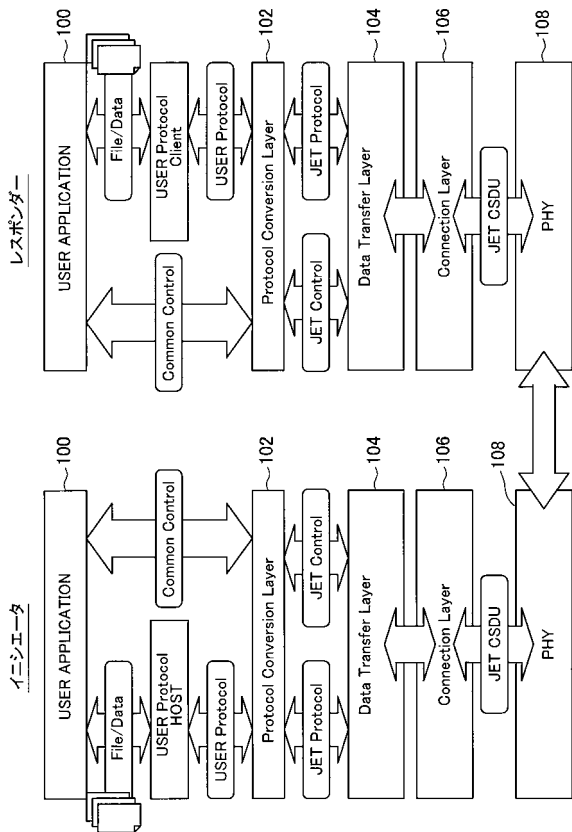
【図1】



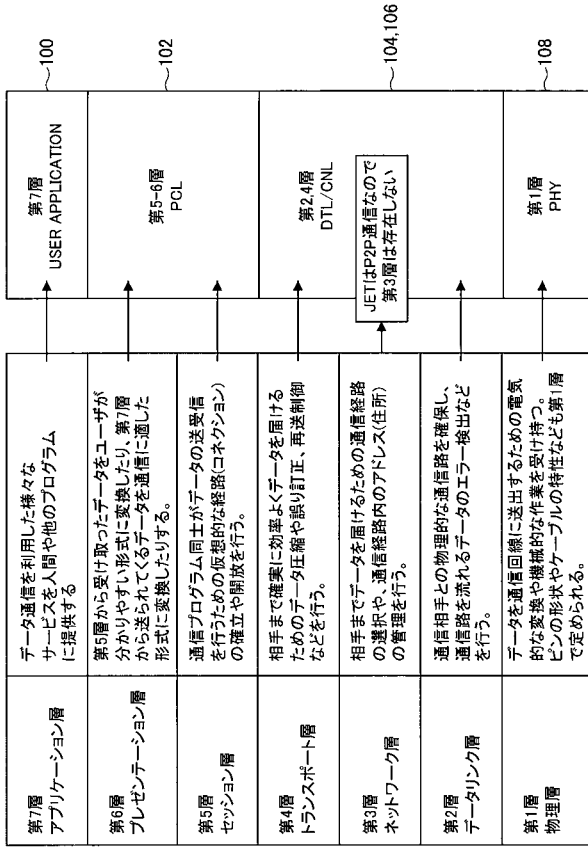
【図2】



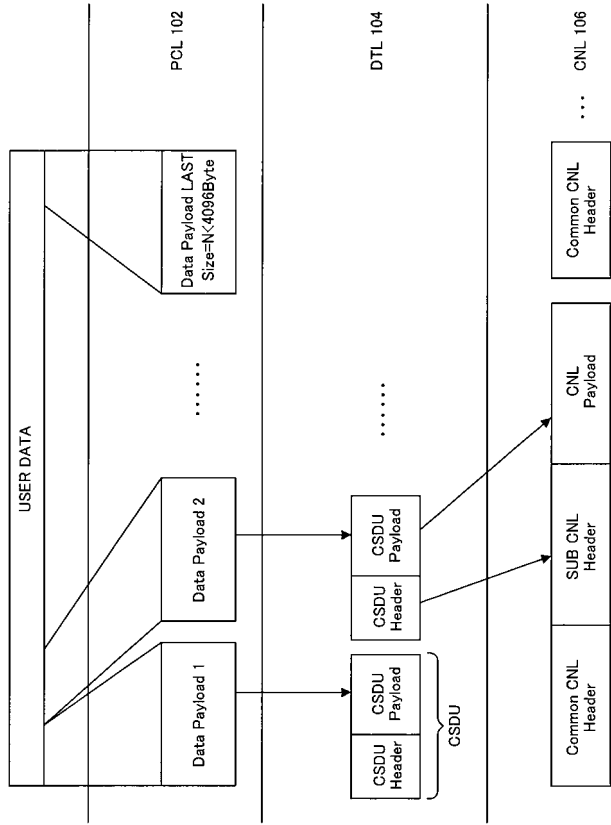
【図3】



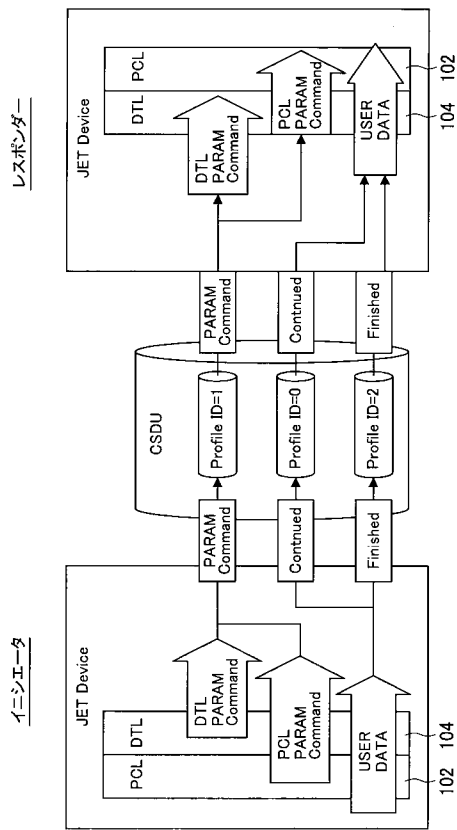
【 図 4 】



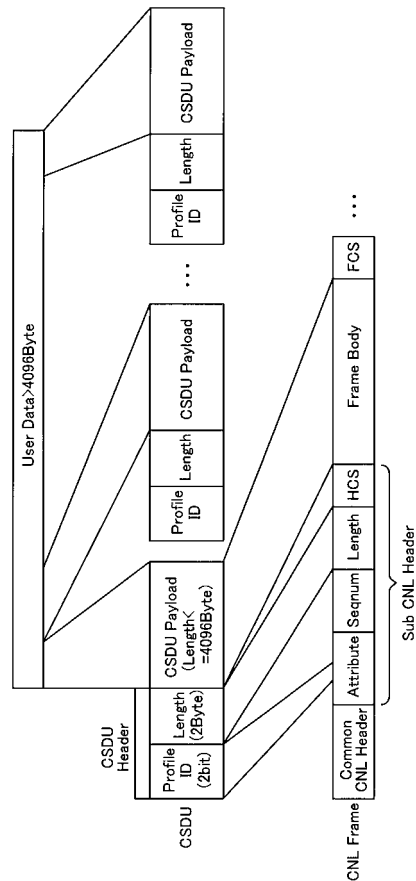
【 図 5 】



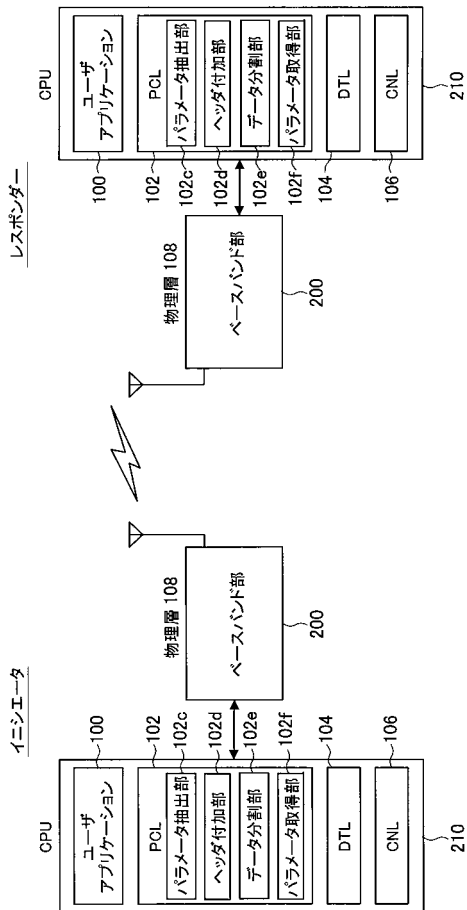
【 図 6 】



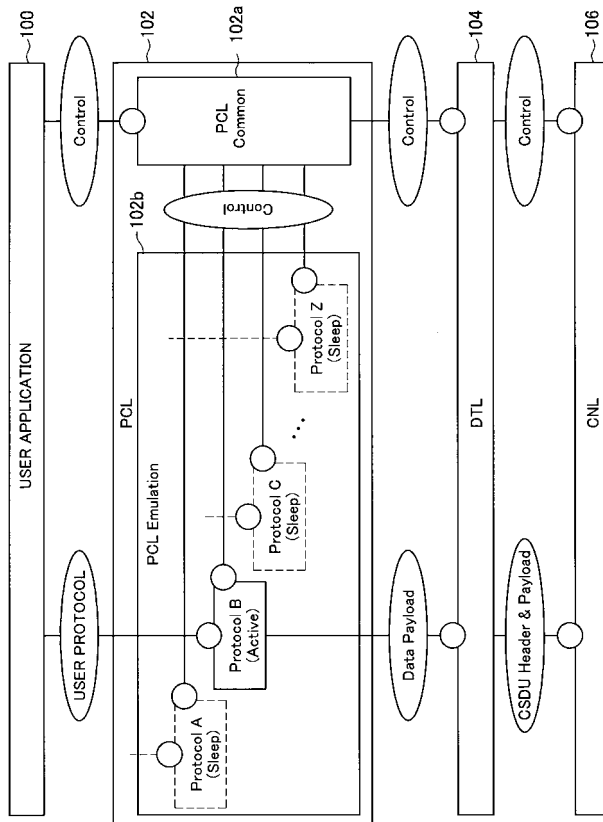
【 図 7 】



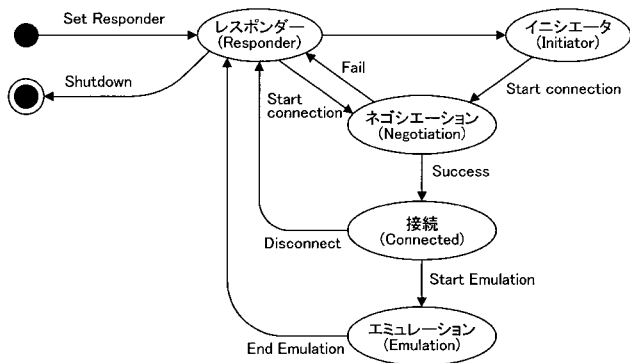
【 図 8 】



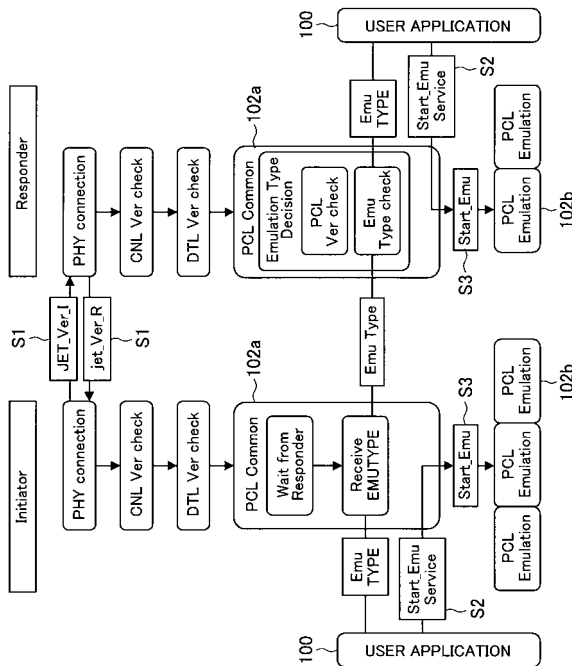
【 図 9 】



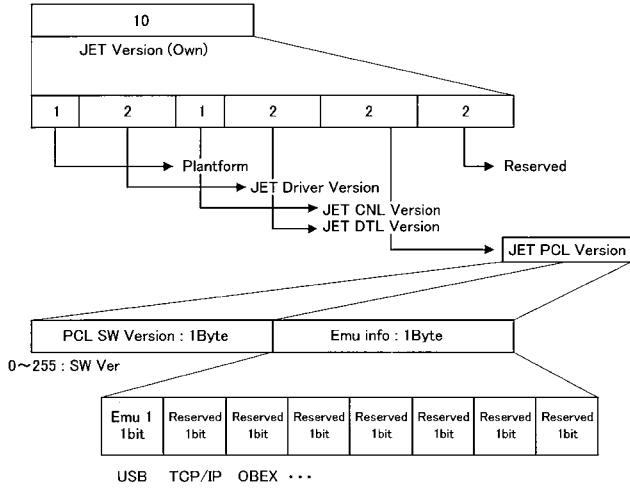
【 図 10 】



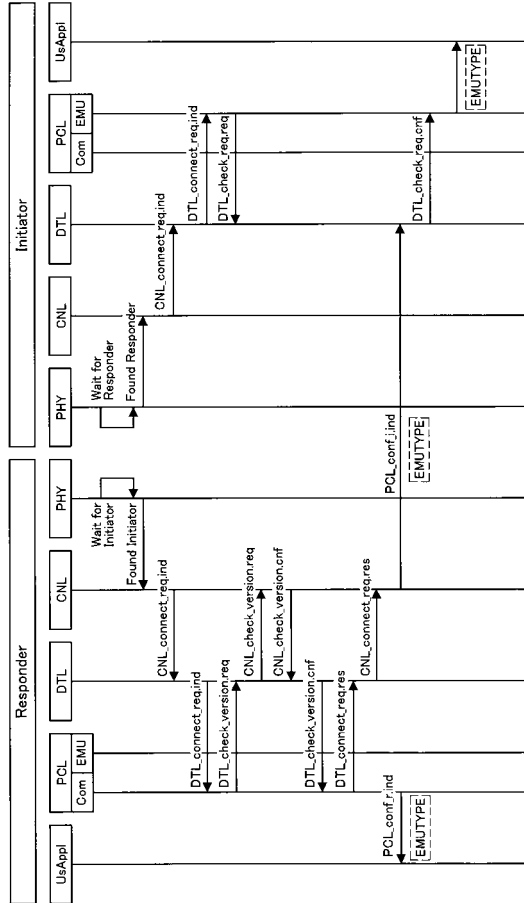
【 図 11 】



【 図 1 2 】



【 図 1 3 】



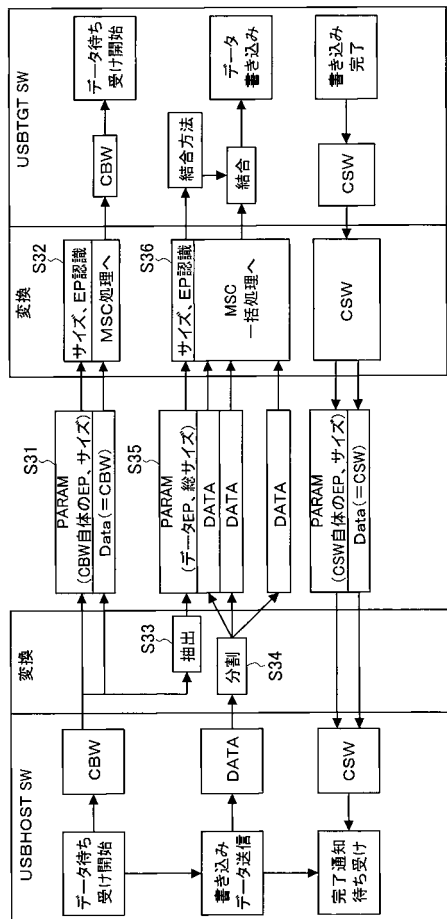
【 図 1 4 】

Service Name	Type				Parameter	Description
	Req	Cnf	Ind	Res		
PCL Common ↔ PCL Emulation						
Start	x	-	-	-	EMU.PARAM	Emulationを開始する
	-	x	-	-	ERROR CODE	
End	x	-	-	-		Emulationを終了する
	-	x	-	-	ERROR CODE	

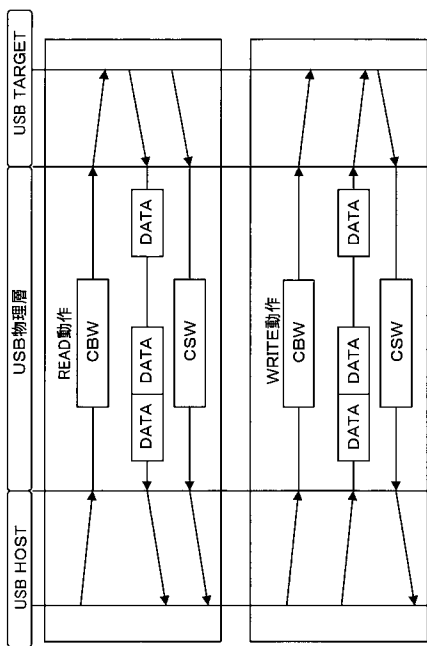
【 図 1 5 】

Service Name	Type				Parameter	Description
	Req	Cnf	Ind	Res		
User Application ↔ PCL Emulation						
Open	x	-	-	-	-	通信を開始する
	-	x	-	-	ERROR CODE	
Close	x	-	-	-	-	通信を終了する
	-	x	-	-	ERROR CODE	
Read	x	-	-	-	SEND DATA	データを受信する
	-	x	-	-	ERROR CODE	
Write	x	-	-	-	RECV DATA	データを送信する
	-	x	-	-	ERROR CODE	
User customize	T.B.D	T.B.D	T.B.D	T.B.D	T.B.D	User Protocolごとの拡張Service

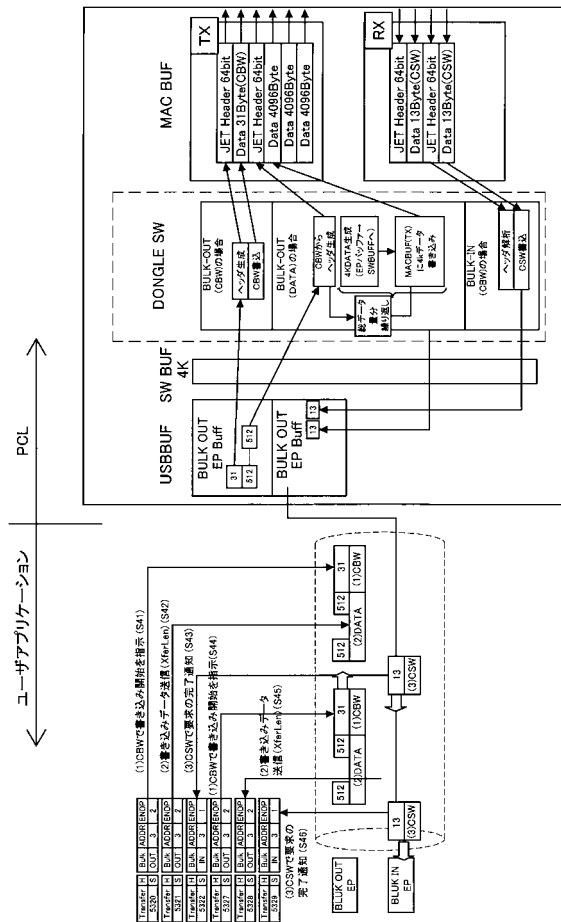
【 図 1 6 】



【 図 1 8 】



【 図 1 7 】



フロントページの続き

Fターム(参考) 5K027 AA11 BB01
5K030 GA03 HA08 HB11 JA05 KA15 LB15
5K034 AA01 AA07 HH62 KK01 MM25 SS00