第11章

キーボードやマウス，HDD，ハブなどを標準周辺機器として使う

クラス・ドライバの役割とその基本動作

岡野 彰文

WindowsやLinuxでは，USBにハブやキーボード，HDDを接続すれば，すぐに使用できます．しかも，OSに標準で用意されているドライバが読み込まれるため，別途ドライバを用意する必要がありません．

OSに標準でドライバが用意されているのは，これらのデバイスの仕様がUSB規格(英語のSpecificationは仕様だが，ここでは規格と表現する)でクラスとして標準化されているためです．

1　クラス・ドライバの役割

USBは，パソコンの世界では標準インターフェースと呼んでもよいほど普及しました．これまで周辺機器側のUSBといえば，ペリフェラル(スレーブとして機能する)側の実装が多く紹介されてきました．しかし最近では，組み込み機器でもその処理能力や使用できるリソースの増大と，組み込みシステムに特化した専用チップの使用により，USBホスト機能を比較的容易に実装することが可能になってきました．

USBペリフェラルとして供給されている機器は実に多様で，しかも低価格化が進んでいます．基本的にこれらの機器はパソコン用として供給されているものがほとんどですが，このようなデバイス群を組み込みアプリケーションでも活用しない手はありません．ホスト機能さえ実装すれば，それらのUSB周辺機器が組み込みシステムからも簡単に使えるのです．

さらに進んで，USBは組み込み機器に適したインターフェースであるということもできるでしょう．これは，USB以前のインターフェースで次のような例を考えると明らかです．

キーボードやマウス，ストレージを接続する機器を考えてみましょう．USBが登場する前は，各個別の物理ポート，それをサポートするインターフェース・チップ，さらにそのハードウェア用のドライバがそれぞれに必要でした(**図1**)．ユーザから見れば，これらの周辺機器を使用する際にはシステムを立ち上げる前に接続しておき，稼働中に取り外しや追加はできませんでした．

これをUSBで実現すると，非常に簡単になります(**図2**)．機器のパネルには，単純な1種類の物理ポートを用意するだけで済み，システムが稼働している最中でも必要なときだけ接続して使えます(**図3**)．

もし，このような機器でUSBハブもサポートできれば，一度に接続できる周辺機器の数はポート数に制限を受けることもなくなります．

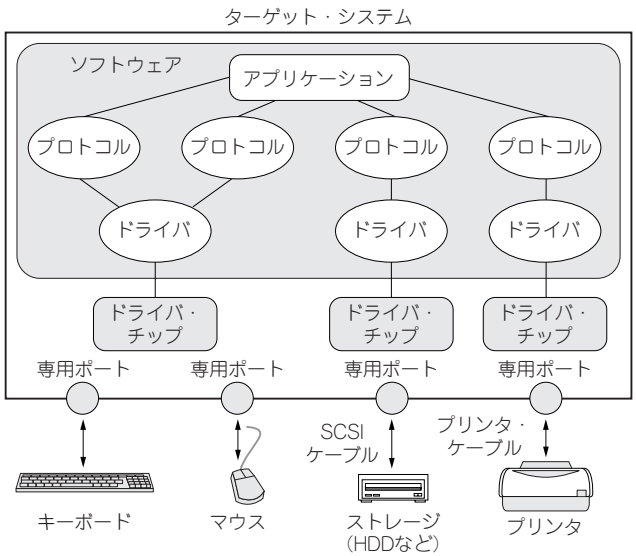

図1　USB以前の周辺機器の接続

第12章

ハブ標準デバイス・クラスの制御

USBハブ・クラス・ドライバの実装と制御手順

岡野 彰文

1　ハブ・クラスとクラス・ドライバの役割

ハブは，USBの中では特殊なクラスと言えるでしょう．なぜなら，ハブ以外のクラスは，各クラスのワーキング・グループによって策定された規格によって定義されているのに対し，ハブ・クラスはUSBの規格の中（USB 2.0規格，第11章）で定義されているからです．

これは，USBの各ペリフェラルを結ぶバス・トポロジ（ネットワーク）を構成するために欠かせない「デバイス・クラス」であるためです．

この特殊性は，ソフトウェアの構造からも次のような特徴として見えます．

ハブ以外のクラスは，通信を行う際により上位のソフトウェア層と共同して動作します（**図1**）．例えば，プリンタやストレージでは，上位のミドルウェアやアプリケーション層と下位の層であるUSBとの間のインターフェースとして動作します．

これに対し，ハブ・クラスは例外的な状況が発生しない限り，上位層とは関係なく下位のバス・ドライバと共同で，USBドライバ・スタックの中の閉じた世界の中で動作します．例外的な状況として，過電流検出などが挙げられます．過電流が発生した場合，接続した周辺機器が動作しないことをユーザに通知するため，アプリケーションやOSにそのイベントの発生を通知します．

しかし，通常の非例外的な動作であれば，クラス・ドライバはポートで発生したイベントをバス・ドライバに通知し，処理を行わせるだけの仕事に専念します（**図2**）．

ハイ・スピード・ハブには，これにもう一つの仕事が加わります．トランザクション・トランスレータと呼ばれるハイ・スピード・ハブ内の機能ブロックが担当する仕事で，そのハブの下位ポートにつながれたフル・スピードまたはロー・スピードのデバイスに対する転送と，上位ポートのハイ・スピードでやり取りされるデータのバッファリングを行います（**図3**）．このバッファリングの制御は，ホスト側システムのハブ・クラス・ドライバで実装されます．

● ルート・ハブはホスト・コントローラに内蔵されてレジスタを介して制御する

USBには，2種類のハブが定義されています．

通常のデバイスとして接続されるハブと，ホスト・コントローラに物理的なポートを提供するために用意

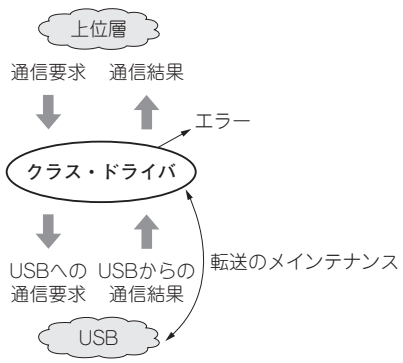

図1　一般的なクラス・ドライバの仕事

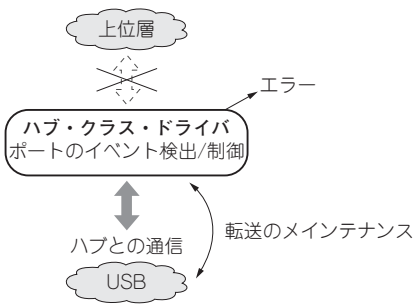

図2　ハブ・クラス・ドライバの仕事

第13章

HID(Human Interface Device)標準デバイス・クラスの制御

キーボードやマウスを USBホストから操作する方法

岡野 彰文

ヒューマン・インターフェース・デバイス(HID)は，USBでもっともよく使われているデバイスかもしれません．USBマウス，USBキーボードなどは，コンピュータの入力機器として，特に多く使われています(**図1**)．HIDクラス・ドライバを備えたシステムは，ほとんどのマウスやキーボードをそのまま用いることができます．これは，クラス・ドライバの特徴がよく活かされている例といえるでしょう．

1　HIDクラス制御の手順

HIDでは，コントロール転送とインタラプト転送の二つのパイプによって通信が行われます．ディスクリプタの要求や設定などはコントロール転送が使われ，デバイスの状態のモニタとデータ取得のためにインタラプト転送が使われます(**図2**，HIDクラス規格4.4参照)．

この接続の方法はハブとよく似ています．ハブも一つのコントロール転送とインタラプト転送の通信を用いて制御していました．このため，どちらかのクラス・ドライバがすでに実装されていれば，エニュメレーションの実装には，そのフレームワークを流用することが可能です．

エニュメレーションそのものは，インタラプト転送を設定するところまではほぼ同じです．

バス・ドライバは，接続されたデバイスがHIDクラスであることを認識し，クラス・ドライバをロードしたあと，それによる初期化を行います．

HIDクラス・ドライバが行うことは，HIDクラス独自のディスクリプタを読み込むことと，その制御に必要な設定を行うことです．HIDクラスにはReportディスクリプタと呼ばれる独自の複雑なデータが存在します．この内容については多くの説明が必要となるため，詳細は割愛します．なぜなら，キーボードやマウスの機能を「手っ取り早く」実現するためにこれらを細部まで理解する必要がないからです．詳細は実装例の中で解説しますが，基本的に多くのキーボード，マウスは決まった型のデータをインタラプト転送で返してくるため，この性質を使って単純化したクラス・ドライバを作成できます．

以下の実装例で，ハブ・クラスの項と重複するものについては多く説明しません．

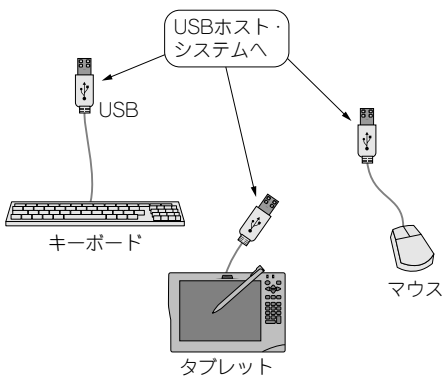

図1　HIDクラス・デバイス

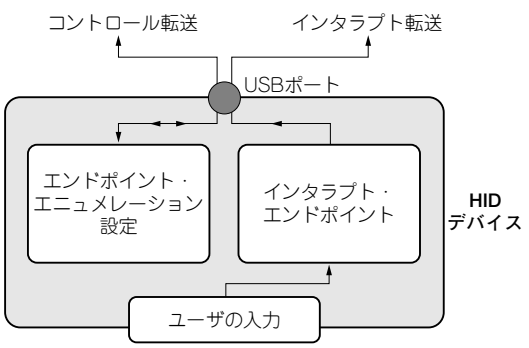

図2　HIDクラスで使われる転送

第14章

マス・ストレージ・クラスの制御

USBメモリやHDDを制御するための実装と制御手順

岡野 彰文

1　マス・ストレージ・クラス制御のプロトコル

USBマス・ストレージ・クラスは，USB経由で記憶装置(ストレージ・デバイス)を制御するために定義されたプロトコルで，SCSIやATA/ATAPIなどのコマンドのラッパ(wrapper)と捉えることができます．

実際に，システムからストレージを操作する場合には，USBストレージ・クラス・ドライバのほかに，SCSIやATAPIなどのストレージ・コマンド・プロトコルやファイルI/O(ファイル・システム)などの実装が必要です(**図1**)．

USBストレージ・クラスを実装するには2種類の転送方式があります．

▶ Bulk-Only転送〔BOT(TはTransfer)と略す〕

▶ Control/Bulk/Interrupt(CBI)転送

BOTは，USBのフラッシュ・メモリ・ストレージや外付けのUSBを使用したハード・ディスク・ドライブ(HDD)，CFやSDなどのメディア・リーダなどでよく使われる方式です．CBIは，主にフロッピーディスクドライブや光学ディスクなどに使われるようです．

本章では，組み込みシステムで特に需要の多い，フラッシュ型メモリ・デバイスのサポートを重要と考え，BOTの詳細について解説します．

● Bulk-Only転送…規格によって定義されるもの

BOTの規格によって定められるのは，どのようなものかを確認します．

具体的な例として，アプリケーションからストレージを操作する場合を考えてみましょう．アプリケーションがファイルの操作を行う場合，ファイルI/Oコール(システム，あるいはライブラリが提供するAPI)を使って操作を行います．ファイルをオープンし，読み書きを行い，クローズするというインターフェースが提供されるわけです．

このAPIの下にはファイル・システムを扱う層が存在し，アプリケーションからの要求をストレージ・デバイスに対する状態の確認や，セクタごとのデータの読み書きといった低レベルの要求に変換します．SCSIやATAPIのコマンド・インターフェースは，その要求に対するインターフェースと各コマンド仕様に合ったプロトコルの実装を行います．

マス・ストレージ・クラスのBOTが定義するのはこの下の部分で，実際にどのようにストレージのコマンドをUSBへ流すのかを定義しています(**図2**)．

● Bulk-Only転送に用いるデバイス

BOTはその名のとおり，Bulk転送のみを用いてストレージにアクセスする規格です．これに対しCBIは，Control(コントロール)とBulk(バルク)，Interrupt(インタラプト)の各転送方法を組み合わせてアクセスする方法です．

BOTを用いるデバイスには，コントロール・エン

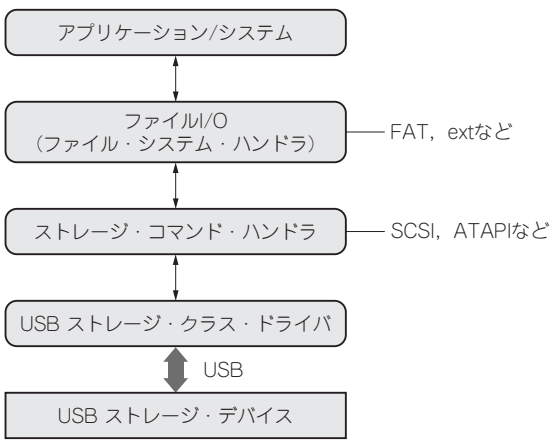

図1　ストレージをサポートするシステム

第15章

プリンタ・クラスの制御

USB対応プリンタを制御する方法

岡野 彰文

1　プリンタ・クラスの概要

● プリンタ・クラスの役割

プリンタ・クラスの制御は，ほかのクラス・ドライバに比べると簡単で実装しやすい部類に入ります．それはこのクラスの定義自体が，ほかのクラスに比べ非常に単純であるためです．

プリンタ・クラスは，ホスト側のシステムからプリンタまでのデータの伝送経路のみを提供します．このため，ストレージ・クラスのようなプロトコルの扱いや，オーディオ・クラスのような機器内の各機能の把握や制御は，このクラス定義には含まれません．プリント作業の設定や制御は，すべてプリンタ・クラス・ドライバより上位の層で行われます．

プリンタ・クラスがこのような構造を採っているのは，プリンタの接続にはUSB以外にもさまざまな方法が使われるためだと考えられます．例えば，パラレル・ポートやシリアル・ポートによる接続，さらにネットワークで接続されるプリンタならEthernetやワイヤレスLANが使われます．古くはSCSIが使われたことや，携帯機器対応を考えてBluetoothで接続するものなどもあります．このように，プリンタには数々の物理的接続方法が存在するため，その制御やデータの転送の物理的な部分と，論理的な部分をうまく切り分けた結果と考えられます．

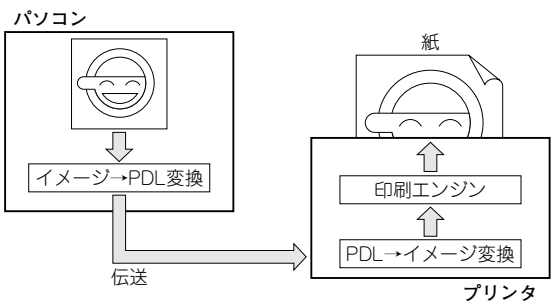

図1　プリンタ制御の概念図

プリンタ接続を抽象化して考えると，**図1**のようなモデルに単純化できます．まず，パソコンなどのプリントするデータを送る側が，プリンタに対するコマンドなどを一連のデータにまとめます．このデータを転送し，受け取った側のプリンタは，このデータを人間の読めるフォーマットに変換して印刷します．

USBのプリンタ・クラスは，この転送を行う部分だけの仕様をまとめたものです．従来は，パラレル・ポートなどで行っていたデータの転送を，USBのバルク転送に置き換えます．

● USB上のデータ

上記の，プリンタにデータ転送を行って印刷を指示するコマンド・データのことを，一般に「ページ記述言語(Page Description Language：以下PDL)」と呼びます．有名なPDLとして，PostScriptやHP-PCLがあります．そのほかにもさまざまなPDLがあり，プリンタの仕様によって，一般的なPDLや独自のPDLが使われます．

プリンタへのデータ転送は，バルクOUT転送を用いて行われます．バルク転送でプリンタに送られるのはこのPDLデータです．しかしこのデータは，必ずしもPDLでなければならないわけではありません．プリンタ・クラスでは，PDLをプリンタ制御プロトコル(IEEE 1284.1など)とともに送出したり，テキスト・データをそのまま流すこともあるとしています．

このほかに，プリンタの状態をモニタするために，オプションとしてバルクIN転送も用意できます．この転送で用いられるデータも，各プリンタで適正化されたPDLや実装によって決められることになっています．

第16章

USBオーディオ・クラスの制御

スピーカやオーディオ機器で使われるアイソクロナス転送の実装方法

岡野 彰文

本章では，USBオーディオ・クラスについて解説します．一般の組み込み機器では，オーディオ・クラスはあまり使われないクラスかもしれません．組み込み機器にとってUSBオーディオは，なじみの薄いアプリケーションです．

本章ではあえてこのオーディオ・クラスを取り上げます．USBヘッドセットを例に，オーディオ・クラス特有のディスクリプタが示す内容について解説し，次にUSBスピーカを例に，ISP1362(ST-NXP Wireless社，以下ST-NXP社)を使用した簡単な転送の実装例とアイソクロナス転送の実際を解説します．

1　オーディオ・クラスの概要

● さまざまなオーディオ機器を想定

オーディオ・クラスの仕様書は，144ページにわたる大部なものです．その中身は，オーディオ機器として想定できる多様な機器とその機能をUSBのインターフェースを通してどのようにホスト側に提供するか，そして制御させる方法がまとめられています．

対象となる機器は，USBスピーカのような単純な出力だけの機器から，楽器やエフェクタ，業務用オーディオ機器に至る高性能で複雑なものまでをカバーしています(**図1**)．

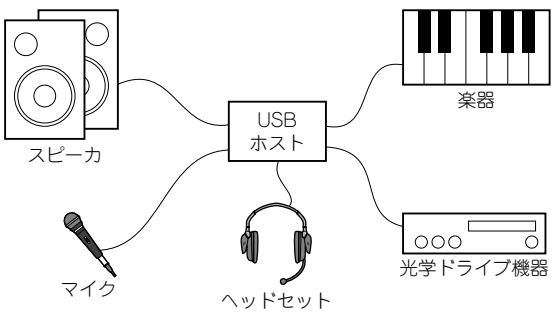

図1　USBを利用したオーディオ機器

また，それらの機器内で実現される細かな機能のブロック，例えば音源の選択スイッチ，ボリューム・コントロール，イコライザ，エコーなどの残響装置，オーディオ・コンプレッサ，ミキサ，果てはAC-3やMPEGのデコーダ，著作権の管理ユニットまでをどのように接続し制御するか，さらにこれらの各機能ブロックに供給するクロック源の定義やその選択を行う方法までを提供しています．

このような仕様は，パソコン上のGUIからオーディオのミキシング・コンソールなどを操作する用途なら，その機能を十分に活用して管理，操作ができるでしょう．パソコンの画面にUSBオーディオ機器の内部状態を詳細に表示し，そこから信号の各制御パラメータを操作することが可能です．

しかし，ここで解説するのはオーディオ・クラスのフル実装ではなく，組み込み機器から制御する簡単なUSBオーディオ出力です．この音声出力をどのように行うかという観点だけから見ると，オーディオ・クラス仕様のすべてを述べるのは非常に大きな回り道です．そこで，比較的単純ですが，面白いディスクリプタを持つUSBヘッドセットを例にして，クラス固有のディスクリプタとそのターゲットの備える特徴を解説します．

その後，オーディオ・クラスに特徴的なアイソクロナス転送についても解説します．アイソクロナス転送はほかの転送方式とは違い，データの実時間生成を最優先する転送方式であり，ホスト・コントローラ内ではほかの転送と違った扱いを受けるのが通例です．

ここでは，ISP1362ホスト・コントローラのサンプル・コードを参考にしながら，具体的な処理方法を紹介します．