

## 第2章 一般的な USB ドライバ

本章では USB ドライバの作成にあたって、WDM ドライバで USB に特化する部分について説明する。なお、USB の詳細については USB-IF(Implementers Forum)のページ(<http://www.usb.org>)に各種の仕様書があるので、参考にして頂きたい。

### 2.1 USB の基礎知識

#### 2.1.1 USB デバイスの構成

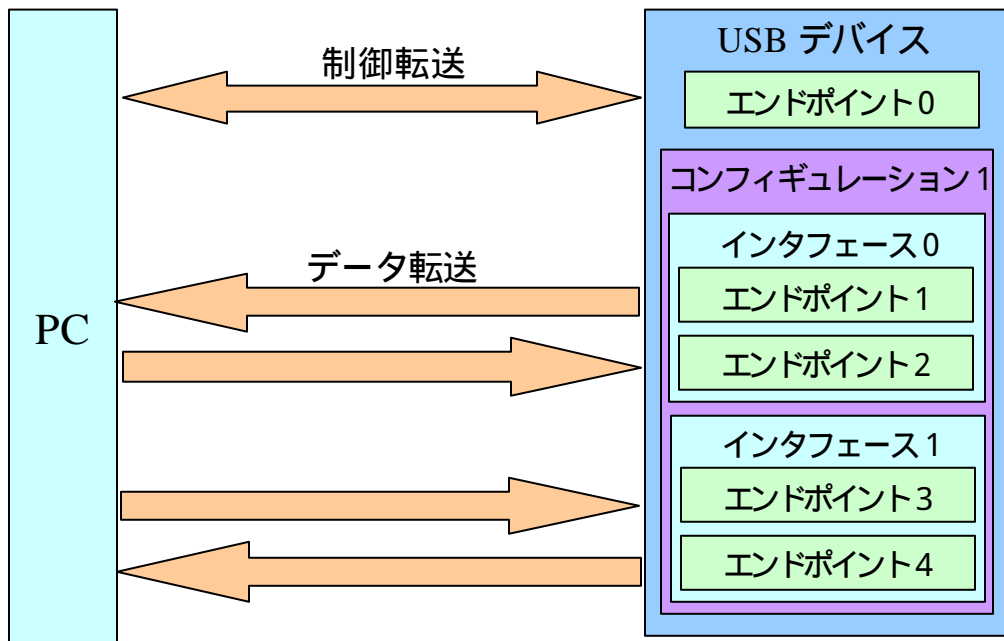


図 2-1 USB デバイスの構成

USB デバイスは図 2-1のように、デバイス内にコンフィギュレーションを必ず 1 つ以上持つことになっている。コンフィギュレーションは 1 つ以上のインタフェースを持っており、インタフェースは 1 つ以上のエンドポイントで構成される。

エンドポイントのうち、エンドポイント 0 はデフォルトエンドポイントと呼

ばれ、制御データの転送を常に行うことができる。その他のエンドポイントはコンフィギュレーションがアクティブにならない限り使用できない。また、エンドポイントとPCの間の論理的な接続路はパイプと呼ばれる。

## 2.1.1.2 ディスクリプタ

USB デバイスはディスクリプタと呼ばれる属性を記述するデータ構造を持っている。ディスクリプタには以下の種類がある。

### 2.1.2.1 デバイスディスクリプタ

デバイスディスクリプタは USB デバイスに 1 つだけとなっており、デバイスのベンダ ID、プロダクト ID、コンフィギュレーションの数の情報を持っている。

### 2.1.2.2 コンフィギュレーションディスクリプタ

コンフィギュレーションディスクリプタはコンフィギュレーションに含まれるディスクリプタ全てのデータバイト数、インタフェース数を含む。多くのデバイスはコンフィギュレーションが 1 つのみとなっている。

コンフィギュレーションディスクリプタの取得要求に対して、インタフェースディスクリプタ、エンドポイントディスクリプタも併せて返却する仕様となっている。

### 2.1.2.3 インタフェースディスクリプタ

インタフェースディスクリプタはデバイスの機能を表すためのディスクリプタである。インタフェースにはインタフェース自身の番号、またそのインタフェースのバリエーションの番号、エンドポイントの数を記述する。インタフェースの概念については後述する。

### 2.1.2.4 エンドポイントディスクリプタ

エンドポイントディスクリプタはエンドポイントのデータ転送種別、転送方向、パケットサイズ、ポーリング時間を記述する。

なお、エンドポイント 0 についてのエンドポイントディスクリプタは存在しない。また、エンドポイントの番号はデータ転送方向によっても異なるため、例えば 1 番のエンドポイントにつき、入力と出力の 2 つを存在させることができる。

### 2.1.2.5 スtring ディスクリプタ

String ディスクリプタはデバイス、インタフェースなどの説明を記述したり、デバイスのシリアル番号文字列を設定できる。

標準的に使用される String ディスクリプタはデバイスディスクリプタ

など、別のディスクリプタからインデックス番号として参照される。

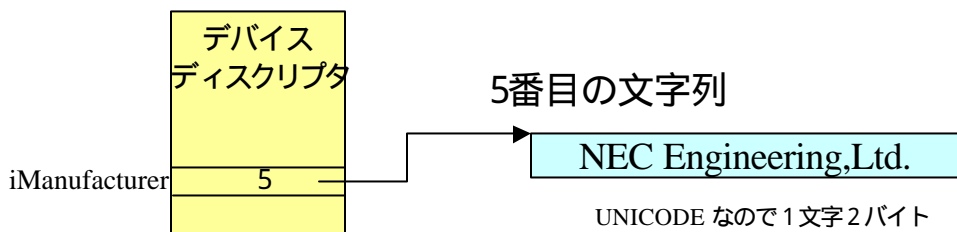


図 2-2 スtringディスクリプタ

Stringディスクリプタはオプションだが、Windows98、2000 では接続時にデバイスの名称など、Stringディスクリプタを元にして表示するため、設定しておくほうが良い。また、Stringディスクリプタは 0 番を指定すると、サポートする言語 ID の配列をデバイスから取得する。Stringの取得はこの言語 ID を使用して取得する。

言語 ID については USB-IF のページに一覧をまとめたものがダウンロード可能となっているので参考にされたい。英語(United States)の 0x0409、日本語の 0x0411 を使うのが一般的と思われる。

#### 2.1.2.6 クラス固有のディスクリプタ

上記以外にも特定のデバイスクラスのみで使用されるディスクリプタが存在する。例えば、HID(Human Interface Device)の HID ディスクリプタ、レポートディスクリプタである。このようなディスクリプタは USB-IF でクラスごとに仕様化されているので、USB-IF のページで該当デバイスの仕様書を参考にしていきたい。

#### 2.1.2.7 USB2.0 のディスクリプタ

USB2.0 では、以下のディスクリプタが追加されている。

- デバイスクオリファイヤディスクリプタ  
High Speed の転送が可能なデバイスが、別の転送でどのように使用されるかを記述したデバイスのディスクリプタ (シリアル番号などは重複するので含まない)
- アザースピードコンフィギュレーションディスクリプタ  
デバイスクオリファイヤディスクリプタで指定した、コンフィギュレーションディスクリプタ
- インタフェースパワーディスクリプタ  
このディスクリプタ自身は USB2.0 固有ではなく、Microsoft の拡張仕様

や Common クラスで策定されようとしており、デバイスの電源管理について詳細の情報を記述する。

### 2.1.3 インタフェース

インタフェースはデバイスを構成する、部分的な単一の機能といえる。

USB1.1 では、デバイスディスクリプタにクラスコードを割り当てるよりも、個々のインタフェースにクラスコードを割り当てるのが推奨される。

例えば、USB で接続される電話があるとする。電話機は複合のデバイスであり、個々の機能に分解すると以下のデバイスクラスのインタフェースを備えるといえる。

つまり、複合のデバイスも個々のインタフェースに分解し、インタフェースごとにドライバを作成すれば良いのである。

表 2-1 電話機のインタフェース要素

機能	デバイスクラス
ボタン	HID クラス
スピーカ	オーディオクラス
マイク	オーディオクラス
ライン制御	コミュニケーションクラス

インタフェースの Alternate 番号に付いては下図のセッティングがこれに相当する。インタフェースはエンドポイントの使用方法などにバリエーションを持たせるために、代替えのセッティングを切り替えられる。

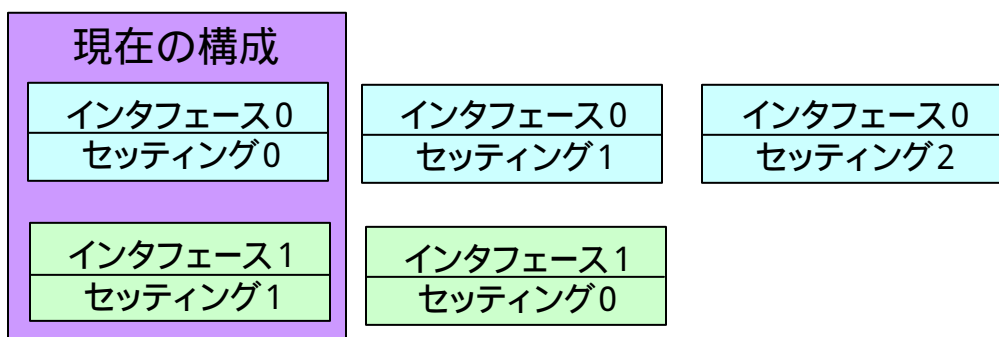


図 2-3 インタフェースのセッティング

例えば、図 2-3 インタフェースのセッティングで、インタフェース 0 に含まれる各エンドポイントのpacketサイズやポーリング間隔をセッティング 0、1、2 でそれぞれ異なった値にすることができる。

このように、異なるインタフェース番号の機能はそれぞれ独立して同時に動作し、セッティングが異なるだけのインタフェースは同時には動作しない。このセッティングの操作には、GET\_INTERFACE、SET\_INTERFACE という標準リクエストが使用される。

## 2.1.4 データ転送種別

### 2.1.4.1 コントロール転送

コントロール転送は制御データの送受信に用いられ、他の転送と異なり双方向の通信で、コマンドの転送とデータ取得が対になっている。

### 2.1.4.2 バルク転送

バルク転送は非同期に大量のデータを転送するために用いられる。フレームの空き時間を利用して転送される。

### 2.1.4.3 割り込み転送

PC の周期的なポーリングに対して、デバイスが任意のタイミングで応答することによって見かけ上、デバイスからの通知が行われたかのように見える。割り込み転送では少量で、いつデータが発生するかタイミングが不定なデータの転送に用いられる。

### 2.1.4.4 アイソクロナス転送

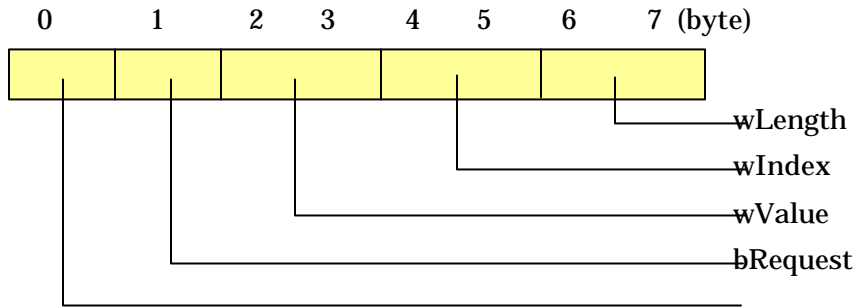
アイソクロナス転送は周期的にデータ転送が行われる。データの信頼性よりも時間が重視されており、エラーによる再送が行われない。音声や CCD カメラの映像などのデータ転送に用いられる。

## 2.1.5 リクエスト

USB のリクエストは標準的に定義されたもの、クラスで定義されたものがあり、そしてデバイスのベンダが独自に定義することも可能となっている。

標準的なリクエストについては、USB 仕様書の 9 章で動作が規定されているため、これらのリクエストにデバイスが準拠しているかどうかのテストプログラムは Chapter9 テストと呼ばれている。

また、リクエストはデータ転送の方向、制御の対象（デバイス、コンフィギュレーションなど）も指定する。



bmRequestType

図 2-4 USB リクエストの構成

USB のリクエストの構成は図のようになっており、bmRequest でリクエストのタイプを指定、bRequest が制御コードに相当する。wValue、wIndex はリクエストのパラメータであり、エンドポイントの番号などを指定する。

wLength はこのリクエストに付随するデータのバイト数である。

リクエストがデバイスにデータを転送するものであれば、このリクエストに続いて wLength バイトのデータ列が転送され、デバイスからのデータを受け取るリクエストであれば、指定データバイト数以下のデータを受け取る。

表 2-2 bmRequestType のビット内容

ビット	内容
7	データの転送方向 0: ホストからデバイス 1: デバイスからホスト
6,5	リクエストの種類 0: 標準 1: クラス 2: ベンダ 3: 予約
4-0	リクエストの対象 0: デバイス 1: インタフェース 2: エンドポイント 3: その他 4: 予約

USB の標準リクエストには以下の種類がある。

表 2-3 標準リクエスト一覧

リクエスト名	機能
<b>CLEAR_FEATURE</b>	特定の機能をクリア。ストール解除など
<b>GET_CONFIGURATION</b>	現在のコンフィギュレーション番号取得
<b>GET_DESCRIPTOR</b>	ディスクリプタを取得
<b>GET_INTERFACE</b>	現在の代替えインタフェース番号を取得
<b>GET_STATUS</b>	エンドポイント等のステータス取得
<b>SET_ADDRESS</b>	デバイスのアドレスを設定
<b>SET_CONFIGURATION</b>	コンフィギュレーション番号設定
<b>SET_DESCRIPTOR</b>	ディスクリプタを変更
<b>SET_FEATURE</b>	リモートウェイクアップなどの機能を設定
<b>SET_INTERFACE</b>	代替えインタフェースを設定
<b>SYNC_FRAME</b>	パターン同期のアイソクロナス転送で使用される、パターン開始番号取得要求

なお、Microsoft 社は電源の状態について USB 仕様の拡張を行っており、SET\_FEATURE の機能に電源状態の遷移を行う機能スイッチを追加している。この拡張スイッチのリクエストはパワーディスクリプタの記述によって発行されるようになる。