

USB 仮想 COM ポート・ デバイスの設計事例

桑野 雅彦

USB はプラグ&プレイ対応とはいえ、OS にドライバが用意されていない場合は、ドライバのインストール作業が必要になる。ドライバは最初に一度だけインストールすればよいが、煩わしい作業であることは確かだ。USB 標準クラスに準拠したデバイスであれば、OS にドライバが用意されているので、初めて接続する場合でもスムーズに使うことができる。ここでは組み込み用途で汎用性の高い USB 仮想シリアル・ポート・デバイスの設計事例について解説する。(編集部)

シリアル・ポートはお手軽通信ポート

ワンチップ・マイコンをパソコンとつないで手軽にやりとりするのに適したインターフェースといえば、なんといっても昔からあるシリアル・ポートでしょう。

シリアル・ポートの電圧レベルは±10V 前後と一般的なロジック電圧レベルとはかなり異なることから、MAX232 などのいわゆるレベル・コンバータを付ける必要があります。それでも、非同期シリアル入出力機能は、多くのマイコンに標準で搭載されています。ホストとの間で最低3本(GND, TxD, RxD)の線を接続すれば、1バイト単位で簡単にデータを送受信できるという点で、非常に手軽です。

ターミナル・ソフトウェアを使えば、スイッチやLEDによるモード切り替えやステータス表示とは比べものにならないほど多くのコマンド指示や情報の表示が行えます。ホスト側のアプリケーションでも、単純にシリアル・ポートをオープンしてリード/ライトするだけで通信が行えます。

しかし、時は流れ、パソコンの周辺機器用ポートは USB に集約されてきました。COM ポートやパラレル・ポートなどのレガシ・ポートのないパソコンも一般的になってしまいました。やむなく FTDI(Future Technology Devices International)社などの USB-シリアル変換 LSI を使って接続することもあります。せっかくのワンチップ・マイコンなのに2チップ構成になってしまうことや、LSI メーカーの専用ドライバが要求される辺りも少々面倒なところがあります。

USB 仮想シリアル・ポートの仕組み

実は USB の仕様の中に、USB コミュニケーション・クラスとして、ネットワークや COM ポートなどを包含した

USB 用の通信インターフェース機器の仕様が標準化されています。Windows でもこのためのドライバを標準で用意しています。USB コミュニケーション・クラスに準拠させておけば、INF ファイルを用意するだけでドライバ本体を作ることなくシリアル・ポートとして見せられます。USB ターゲット・コントローラが内蔵されているマイコンの場合は、コミュニケーション・クラスに準拠した機器として見せかけることができれば、Windows 上からは通常の COM ポートに接続されたものと同じように扱うことができます。

USB ケーブルには、ロジック回路を組むには都合の良い +5V の電源が供給されており、最大で 500mA もとることができます。ケーブル1本で電源供給と通信が可能になるのも USB を使う大きな利点です。

ここでは Cypress Semiconductor 社(以下、Cypress 社)の USB ターゲット・コントローラ内蔵の PSoC マイコンを使った、USB コミュニケーション・クラス準拠の USB 接続機器の製作例を紹介します。

1. USB コミュニケーション・クラスの概要

USB コミュニケーション・クラスの資料

USB コミュニケーション・クラスについては、

http://www.usb.org/developers/devclass_docs#approved

の下にある Class Definitions for Communication Devices 1.1 (ファイル名は usbcidc11.pdf)に資料があります。これを

読むと、このほかにもワイヤレス・モバイル・コミュニケーション・デバイスや Ethernet エミュレーション・モデル・デバイスなども、コミュニケーション・クラスのサブクラスとしてドキュメントが用意されていることがわかります。

コミュニケーション・デバイスは 三つのクラスを持つ

コミュニケーション・クラスの規格は、単なるシリアル・ポートだけでなく、Ethernet や HDLC, ATM など、さまざまな通信インターフェースをサポートしています。そのため、かなり大がかりな仕様になっています。

コミュニケーション・デバイスは、図 1 のように 3 種類のクラスを組み合わせ、機器がどのようなものであるかを規定するようになっています。

- (1) コミュニケーション・デバイス・クラス
- (2) コミュニケーション・インターフェース・クラス
- (3) データ・インターフェース・クラス

(1) のコミュニケーション・デバイス・クラスは、装置全体がコミュニケーション・クラスに準拠して作られていることをホストに対して示すものです。(2) と (3) は、それぞれ独立したインターフェースとして構成されています。

コミュニケーション・インターフェース・クラス

コミュニケーション・インターフェース・クラスのクラス・コードは 02h です。その下にはサブクラス・コードがあり、その装置がどのような通信インターフェースを持つ機器なのかを示すようになっています。サブクラス・コードの一覧を表 1 に示します。今回の USB 仮想シリアル・ポートでは、02h の Abstract Control Model を使用します。

表 1 コミュニケーション・インターフェース・サブクラス・コード一覧

コード	サブクラス
00h	予約
01h	Direct Line Control Model
02h	Abstract Control Model
03h	Telephone Control Model
04h	Multi-Channel Control Model
05h	CAPI Control Model
06h	Ethernet Networking Control Model
07h	ATM Networking Control Model
08h ~ 7Fh	予約(将来の拡張用)
80h ~ FFh	予約(ベンダ定義)



ホスト・パソコン



図 1 USB コミュニケーション・クラス機器のクラス構成

コントロール・プロトコル・コード

正式にはコミュニケーション・インターフェース・クラス・コントロール・プロトコル・コード (Communication Interface Class Control Protocol Codes) というなんとも長ったらしい名称が付いています。要はインターフェース・ディスクリプタのプロトコル・コードのフィールドで、その装置の制御をどのようなプロトコルで行うのかを指定するものです。注意が必要なのは、ここで指定するのは装置を制御するためのプロトコルであって、データ転送のためのプロトコルではないということです。

コミュニケーション・クラス仕様書の Version 1.1 で定義されているのは、表 2 のとおりです。00h と FFh はクラスではなく USB 規格そのものの規定で、なし (00h) およびベンダ定義 (FFh) とされているので、標準化されているといえるのは 01h の AT コマンドだけということになります。

データ・インターフェース・クラス

データ・インターフェース・クラスのクラス・コードは 0Ah です。サブクラス・コードはありません。さらにプロトコル・コードでデータ転送のためのプロトコルを指定します。現状では表 3 のようなものが定義されています。今回は特にデータ転送のためのプロトコルは必要としないので、00h にします。

表 2 コントロール・プロトコル・コード

プロトコル・コード	参照規格	内容
00h	USB 規格書	クラス定義プロトコルはなし
01h	V.25ter	AT コマンド(いわゆる Hayes 互換)
02h ~ FEh	なし	予約(将来使用予定)
FFh	USB 規格書	ベンダ定義