

RXマイコンでUSBマウスを エミュレーションする



関連データ

芹井 滋喜

本誌2011年5月号には、ルネサス エレクトロニクスのRXマイコン基板が付属しました。実際にこのマイコンで、いろいろなプログラムを作っている方も多いことと思います。

RXマイコンには、USBのホストとファンクションのモジュールが内蔵されています。このモジュールを使いUSB対応デバイスを開発する場合は、USBプロトコル・スタックを利用すると便利です。USBのプロトコル・スタックは、各社からさまざまなものが販売されています。プロトコル・スタックを使用すると、既に用意されているさまざまなデバイス・クラスを容易に使用できます。また、独自デバイスを作る際も、USBデバイスのアクセス用のAPIが利用できるので開発が容易になります。

ルネサス エレクトロニクスからは、USBファンクション用のサンプルがいくつか提供されています。これらのサンプルは、USBプロトコル・スタック「USB Basic Firmware」を使用して作成されています。このスタックは、 μ ITRONとOSなしの環境に対応しています。しかし、OSなしで動作させる場合も、スケジューラを動作させたりする必要があり、またクラス・ドライバも、このUSB Basic Firmwareの形式に合わせる必要があるため、気軽に扱うにはちょっと難しいかもしれません。

そこで、筆者らが現在開発中のUSBファンクション・スタック「Slacker Stack」を、RXマイコンに移植して動作させてみました。

非常にコンパクトなものです。市販の多くのUSBスタックは、リアルタイムOSを前提とした設計となっており、OSなしで使用の際は制限が多い場合があります。けれども、Slacker Stackは、OSを使わない場合でも簡単に利用できるようになっています。

Slacker Stackの主な特徴は次の通りです。

- コード・サイズが小さく、PICマイコンのようにメモリが制限された環境でも使用可能
- ディスクリプタなどの静的データのROM化に対応し、RAMの使用を少なくできる
- マイコンのビット・サイズやエンディアンに依存しないため、使用するマイコンを選ばない
- ハードウェア依存部分が非常に小さく、移植性に優れている
- 構造がシンプルで理解しやすく、また高速に動作する
- CDCやHIDマウス、HIDキーボードのクラス・ドライバを標準で付属
- 単一のUSBコントローラ限定のため、無駄なリソースを食わない
- OS非依存で、OSのない環境でも十分なパフォーマンスで動作する
- リアルタイムOSを使用したマルチタスク環境にも対応
- ロー・コストで、コード・サイズも小さい

● Slacker Stackの構造

図AにSlacker Stackの構造を示します。

スタック構造は、一般的なUSBファンクション・スタックとほとんど同じですが、SETUPドライバが独立している点が大きな特徴でしょう。

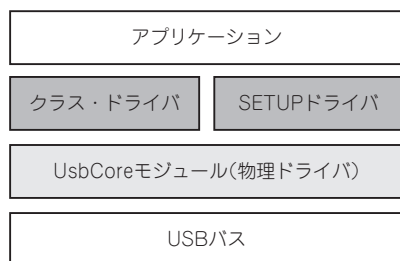
SETUPドライバは、USBの標準リクエストを処理するものですが、多くのUSBファンクション・スタックでは、UsbCoreモジュール(物理ドライバ)で処理しています。標準リクエストの処理はUSBのエミュレーション処理と密接な関係があり、また、マイコン内蔵のUSBモジュールによっては標準リクエストの処理の一部を、USBモジュール内で処理して、ファームウェアの処理の負担を減らすようにしているものもあります。このため、多くのUSBスタックでは、ハードウェア依存部分である物理ドライバの部分でこの処理を行うようになっていくようです。しかしながら、小型のマイコンではUSBのハードウェアの機能が必要最小限に抑えられているケースも多く、このような場合は物理ドライバの移植の際に標準リクエストの処理も書き起こす必要があるため、移植作業の負担が非常に大

1. Slacker Stackについて

● USBファンクション・スタック「Slacker Stack」

Slacker Stackは、ソリトンウェブで開発中のUSBターゲット機器向けファンクション・スタックです。

このUSBファンクション・スタックは、PICマイコンのような比較的小規模なマイコンでも実装できるように設計された、



図A
Slacker Stackの構造