

FRマイコンを使った独自クラス USBターゲット機器の設計

[芹井 滋喜]

7月号掲載のFRマイコンを使ったUSBターゲットの設計事例は、HIDクラスとしてUSBマウスをエミュレーションするものだったため、ホスト環境とFRマイコンとの間でデータ転送をする用途には使えなかった。そこで、USBの標準クラスには準拠させずに、エンドポイントを独自仕様で使うUSBターゲット機器を設計する。独自クラスの場合は専用ドライバが必要になるが、ここでは本誌でよく取り上げる汎用USBドライバを使った。

(編集部)

本誌2008年5月号付属FRマイコン基板のCPU MB91FV310Aには、USBのホストおよびターゲット・コントローラが内蔵されています(CPUのデータシートにはUSBファンクションと明記されているが、本誌ではUSBターゲットと表記)。また、本誌2008年7月号には、このUSBターゲット・コントローラを使った、USBマウス・エミュレーション・プログラムの作成事例が紹介されました⁽²⁾。マウスの設計事例は、HID(Human Interface Device)デバイスのサンプルとして非常に参考になりますが、HIDデバイスは用途が限られます。そこで今回は、MB91FV310AのUSBターゲット・コントローラを使った、汎用のUSBデバイスの設計事例を紹介します。

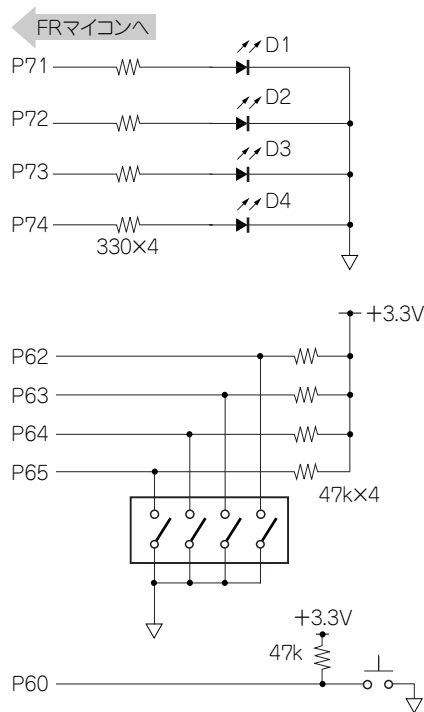


図1 USB ターゲット実験回路

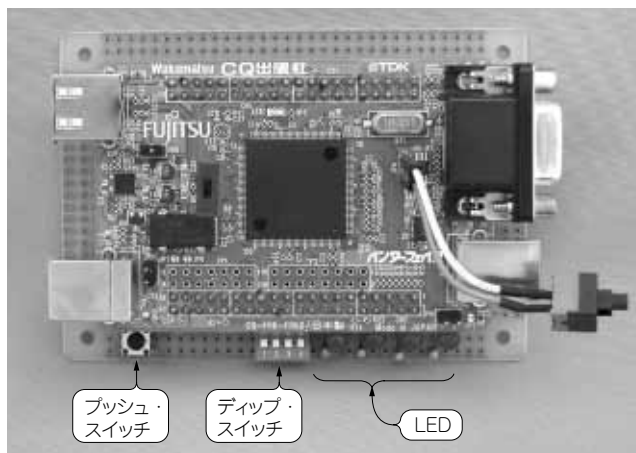
写真1
FRマイコン基板を
使ったUSBターゲット
実験基板

● 設計する USB ターゲット機器の仕様

USBを使ってデータ転送をする場合、ホストからUSBターゲット側への転送にはバルク OUT 転送を、USBターゲットからホスト側への転送にはバルク IN 転送を使います。また、USBターゲットからホストに対して、何らかの状態が変化したことを通知したい場合には、インタラプト IN 転送を使います。

USBにはもう一つ、アイソクロナス転送がありますが、これは転送したデータが途中で化けても次から次へと新しいデータが送られてくるようなストリーム転送用途なので、一般的なデータ転送用途では使いません。

ここでは、バルク転送とインタラプト転送を使ったUSBターゲットを設計します。USBを使ったデータ転送の動作確認を目的とするので、転送するデータは簡単に確認ができるものがよいでしょう。そこで、バルク OUT 転送を使ってLEDの点灯状態を出力し、バルク IN 転送を使ってディップ・スイッチの状態を入力し、インタラプト転送を使ってプッシュ・スイッチが押されたタイミングをホストに通知するようにします。



実は、参考文献(1)の第8章には、EZ-USBシリーズの使い方として、バルクの送受信とプッシュ・スイッチを使った実験が紹介されています。今回の実験回路では、これと同じことをMB91FV310Aで行ってみました。後で詳しく説明しますが、Windowsのドライバやテスト・アプリケーションも、この記事のサンプルをMB91FV310A用に修正して使用しています。興味のある方は、この記事を参照して比較してみるのも面白いと思います。

製作したUSBターゲット実験基板を写真1に、USBターゲット実験回路の回路図を図1に示します。

●MB91FV310AのUSBターゲット・コントローラ

図2にMB91FV310A内蔵のUSBターゲット・コントローラのブロック図を示します。MB91FV310Aの内蔵USBターゲット・コントローラは、表1に示す四つのエンドポイントがあります。

エンドポイント構成で分かるように、MV91FV310AのUSBターゲット・コントローラではエンドポイントが最大4個、使用用途もそれぞれのエンドポイントで固定されている、非常に限定されたハードウェア構成となっています。

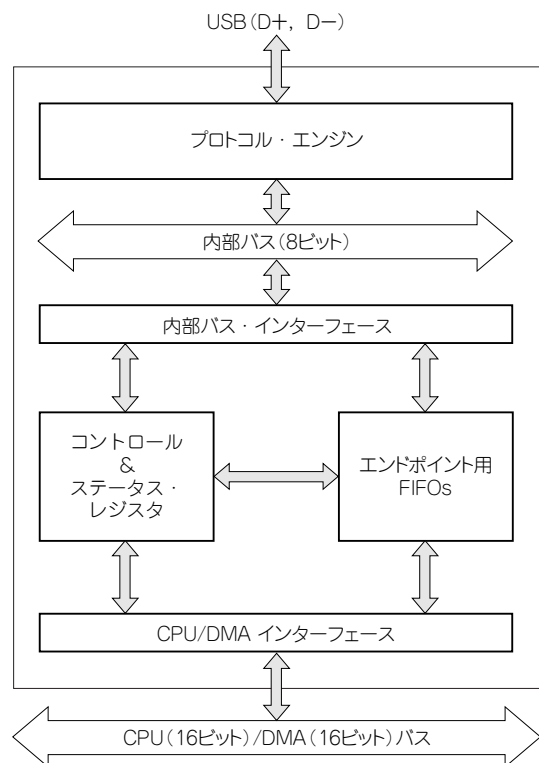


図2 MB91FV310AのUSBターゲット・コントローラのブロック図

ただ、USBターゲット機器では、バルク IN/OUT 転送とインタラプト転送が使えれば十分な場合がほとんどなので、この構成で問題が出ることはあまりないと思います。むしろ、ターゲットの構成を限定することでUSBの制御が非常にシンプルで使いやすくなっており、ユーザにとってバランスのとれた非常に使いやすいUSBターゲット・コントローラとなっています。

●プロトコル・エンジンの概要

USBデバイスでは標準リクエストの処理の実装が必須となり、この部分がUSBデバイスの作成時のハードルを高くしている要因の一つでもあります。標準リクエストでは、デバイスの識別に必要なベンダIDやプロダクトID、エンドポイント構成などのコンフィグレーション情報の取得が行われるので、避けて通ることはできません。

MV91FV310Aのプロトコル・エンジンでは、標準リクエストのほとんどの処理をプロトコル・エンジンが行うため、USBデバイスの開発は必要最小限の実装で済むようになっています。

自動処理を行う標準リクエストは次の通りです。これらのリクエストに関して、CPU側は何もする必要がありません。

- CLEAR_FEATURE
- GET_CONFIGURATION
- GET_INTERFACE
- GET_STATUS
- SET_ADDRESS
- SET_CONFIGURATION
- SET_FEATURE
- SET_INTERFACE

実際にCPU側で処理が必要な標準リクエストは、GET_DESCRIPTOR/SET_DESCRIPTOR/SYNCH_FRAMEの三つです。後の二つは必須ではないため、GET_

表1 MB91FV310Aのエンドポイント構成

エンドポイント	コンフィグレーション	インターフェース	オルタネート	転送タイプ	最大パケット・サイズ (バイト)
0	—	—	—	コントロール転送	8
1	1	0	0	バルク OUT 転送	64
2	1	0	0	バルク IN 転送	64
3	1	0	0	インタラプト IN 転送	8

エンドポイント・アドレスや転送タイプ、最大パケット・サイズなどが固定されている