

USBホスト・コントローラの製作

Linuxが動作するCPUボードに接続して使う 安井 吏

ここでは、USBホスト・コントローラ的设计事例を紹介する。Linuxが動作するCPUボードに接続して動作させた。USBインターフェース的设计では、ハードウェアのみならずデバイス・ドライバ的设计にも多大な労力が必要になる。そこで今回は既存のLSIと仕様を合わせ、既存のデバイス・ドライバをそのまま使用して評価を行った。(編集部)

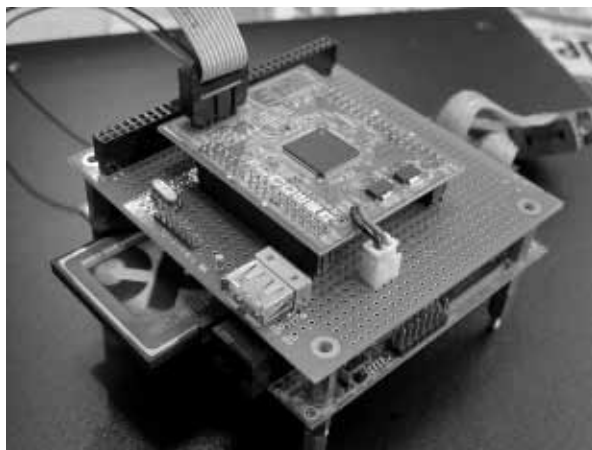
近年では組み込み機器でもUSBやEthernetなど、パソコンと同等のインターフェースを要求されることが多くなりました。しかし、これらの制御用LSIを探してみると、パソコン向けにPCIバスへ接続する仕様のものが多いようです。組み込みマイコンのバスに直結できるものはなかなか見つかりません。また、非PCIバス対応のコントローラが見つかったとしても入手が困難であったり、代替品がないため生産中止の際のリスクを抱えるなど、気軽に使うというわけにもいかない場合が多いのではないのでしょうか。

FPGAは設計者が論理を自由に組み込める便利なLSIです。それならいっそのこと、これらのインターフェース機能をFPGAで実現してみたらよいのではないかと考えました。そこで今回、筆者は、一般的なCPUバスに直結可能なUSBホスト・コントローラを製作し、Linux上で各種USBデバイスを動作させました(写真1)。

USBのプロトコルについて

ここではごく簡単にUSBのプロトコルについて説明します。USBのプロトコルについて説明されているドキュメントはすでに数多くありますので、詳しくはそれらを参考にしてください。

なお、ここではUSB 1.1の仕様に基づいて説明します。USB 2.0では仕様が一部拡張されている部分があります。

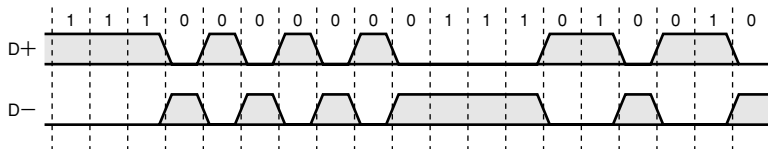


〔写真1〕 製作したUSBホスト・コントローラ・ボード
Linuxの動作するCPUボードに接続したようす。

〔図1〕

NRZI変調

送るべきデータが‘0’のときは1ビット前のデータの状態で反転され、‘1’の場合は変化しない。



●物理層

USBでは、D+信号とD-信号の差動信号としてデータの送受信を行います。このデータ線は1対しかないので、通信は半2重となります。

USBのライン上のデータはNRZI変調が施されています(図1)。これは送るべきデータが‘0’のときは1ビット前のデータの状態が反転され、‘1’の場合は変化しないというものです。ただし、NRZI変調では‘1’のデータが続くとライン上の信号レベルが変化しない状態が続き、受信側ではデータの境目がわからなくなってきます。これを防ぐため、ビット・スタッフという処理が行われます。これは、一定数以上の‘1’が続いた場合、途中にダミーの‘0’を入れ(ゼロ・インサージョン)、受信側では一定数以上‘1’が続いた後の‘0’を無視する(ゼロ・デリーション)処理です。

なお、ライン上のデータはすべてLSBが先に、MSBが後に現れることになっています。

[表1]
パケットの種類

パケット・フォーマットの種類	パケットの種類	PID
トークン	OUT	0001B
	IN	1001B
	SOF	0101B
	SETUP	1101B
データ	DATA0	0011B
	DATA1	1011B
ハンドシェイク	ACK	0010B
	NAK	1010B
	STALL	1110B
特殊	PRE	1100B

●USBのパケット

パケットとは、USBでデータ転送を行う際のデータの最小単位です。パケットはSYNCという特定のデータ(80H)で開始され、EOP(End of Packet)で終了になります(図2)。EOPは、USBラインのD+とD-がともに“L”になり、それが2ビット期間続く状態です。

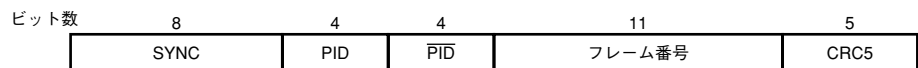
パケットには表1に示すような種類があり、SYNCに続くPID(Packet ID)と呼ばれるフィールドで判別します。また、PREパケットを除くそれぞれのパケットはトークン、データ、ハンドシェイクのどれかに分類され、それぞれパケットのフォーマットが異なります。

トークン・パケットにおける四つのパケットのうち、SOFは特殊で、ホストは1msごとにSOFパケットを送出する必要があります。SOFと次のSOFの間をフレームと呼び、このフレームの時間内で各種の転送が行われます。SOFの送出間隔の許容誤差は±500nsと、ソフトウェアで処理するには厳しいため、ハードウェアで自動的に送出するのが一般的です。

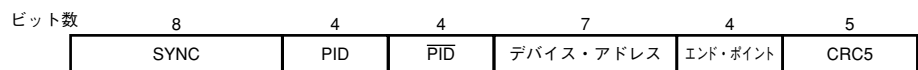
●USBのトランザクション

いくつかのパケットを組み合わせて、実際に意味のある転送を構成するものをトランザクションと言います。ソフトウェアからUSBのデバイスを扱う場合は、このトランザクションのレベルで転送を制御することになります。

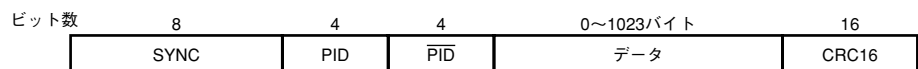
トランザクションにはSETUP, IN, OUTの3種類があり、それぞれどのようにパケットをやり取りするかが決まっています(図3)。例えば、SETUPトランザクションは



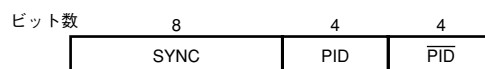
(a) トークン・パケット(SOF)



(b) トークン・パケット(SETUP, IN, OUT)



(c) データ・パケット



(d) ハンドシェイク・パケット

※PIDはPIDのビットを反転したデータ

[図2]

パケットのフォーマット

SYNCという特定のデータ(80H)で開始され、EOP(End of Packet)で終了する。