

Linux 用 OHCI USB ホスト・ド
ライバの実装事例

海老原 祐太郎

USB ホストの制御プログラムは高度である。そこで組み込み向け Linux を移植することで、USB ホスト制御プログラムを移植するという方法がある。しかしパソコンと組み込みシステムとではメモリ周りのアーキテクチャが異なるため、OHCI 仕様 USB ホスト・ドライバは、そのまま移植するだけでは正常に動作しない。ここでは SH シリーズ向け Linux における OHCI 仕様 USB ホスト・ドライバの実装について詳しく解説する。(編集部)

USB クライアントのドライバ資産を生かしたい
USB デバイスを利用するにはホスト・ドライバだけではなく、マウスや USB メモリ、USB サウンド機器など、デバイスごとにクライアント・ドライバを用意しなければなりません。しかし、Linux などの既存 OS を利用することによって、クライアント・ドライバを新規に書き起こすことなく組み込み機器で動作させることが可能となります。筆者は Linux (Kernel 2.6) を SH-4 シリーズの一つである SH7760 向けに一部書き直すことで、Linux のクライアント・ドライバをほぼそのまま動作させることに成功しました。

Linux はもともと OHCI をサポートしていますが、Linux 標準の OHCI ドライバは PC アーキテクチャに基づいた設計になっています。そのため、SH7760 でそのまま動作す

るわけではありません。PC アーキテクチャと SH7760 内蔵 OHCI の機能的な差異をソフトウェアで吸収する必要があります。

SH7760 内蔵の OHCI が Linux から利用できれば、Linux の豊富なクライアント・ドライバがそのまま利用可能となります。そのため、Linux のクライアント・ドライバを一切変更せず、ホスト・ドライバの変更だけで USB 周辺機器を動作させることを目標とします。

本稿では SH7760 の OHCI を Linux で利用するうえで必要になった実装について紹介します。後半では CAT760 CPU ボード(写真1,表1)を例に、Linux Kernel 2.6 の USB 機能の使い方について説明します。

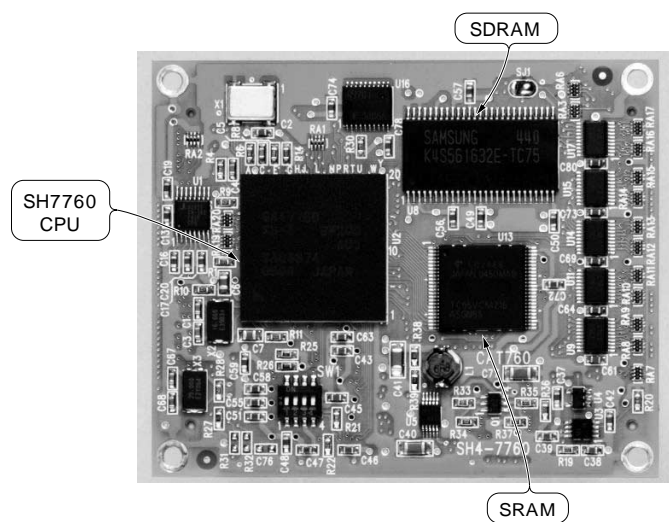


写真1 CAT760 CPU ボード概観

フラッシュ ROM と LAN コントローラは裏面に実装されている

表1 CAT760 CPU ボード仕様

CPU	SH7760(SH-4), 200MHz(バス・クロックは 66MHz)
メモリ	SDRAM 64M バイト(メイン・メモリ) フラッシュ ROM 16M バイト (ブートおよびディスク領域) SRAM 512K バイト (バッテリ・バックアップ・ディスク領域)
Ethernet	100Base-TX x1(AX88796)
シリアル・ポート	RS-232-C レベル x2, ロジック・レベル x1 合計 3 チャンネル (うち 1 チャンネルはコンソールとして使用)
USB ポート	USB1.1 ホスト
I/O ポート	3.3V TTL レベル(42 本)
拡張バス	#RD, #WR 制御型 非同期方式
その他	リアルタイム・クロック IC
液晶コントローラ	320 ピクセル x 240 ピクセル, 16 ビット色対応
サイズ・質量	60mm x 73mm(突起部含まず), 重量 25g
電源	+3.3V 単一電源, 最大 600mA, プロセス待機時 200mA
評価ボード	EB760

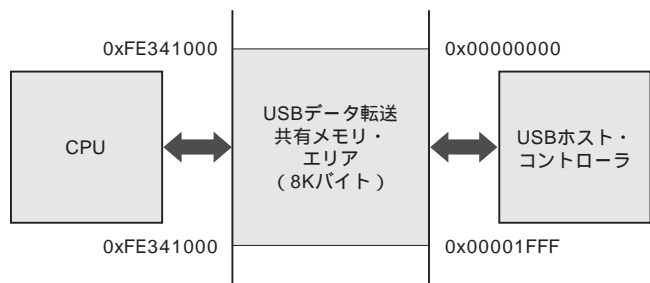


図1 SH7760 内蔵OHCIの構造

SH7760 内蔵のOHCIについて

SH7760は8Kバイトの大きさを持つUSBデータ転送共有メモリ・エリアを持ちます。共有メモリ・エリアはUSBホスト・コントローラとCPUの間で共有されます(図1)。共有メモリ・エリアは8ビット幅, 16ビット幅, 32ビット幅でアクセスが可能で, またデフォルトではリトル・エンディアンのデータ配列になっています。

パソコン向けOHCIとの差異について

OHCIで定められているデータ転送の仕様では, HCCA (Host Controller Communication Area)やED(Endpoint Descriptor), TD(Transfer Descriptor)といったデータ構造体を使って, ソフトウェアとホスト・コントローラの間でのデータ転送を行います。パソコン向けLinuxが想定しているOHCIホスト・コントローラのデータ転送を図2に示します。パソコンでは, メイン・メモリの任意のアドレスにHCCAやED, TDのデータを配置し, ホスト・コントローラはPCIバス・マスタ方式でメイン・メモリ上に配置されたデータを読み書きします。

ところがSH7760内蔵のOHCIでは, 図3のようにHCCAやED, TDを専用の共有メモリ上に配置することになります。このわずか8Kバイトの共有メモリ上にすべてのデータを配置しなければなりません。

パソコン向けに用意されたLinux用OHCIドライバをSH7760向けに移植するにあたっては, 次の二つの問題を解決する必要があります。

- (1)この猫の額のような小さな領域を用いてすべてのデー

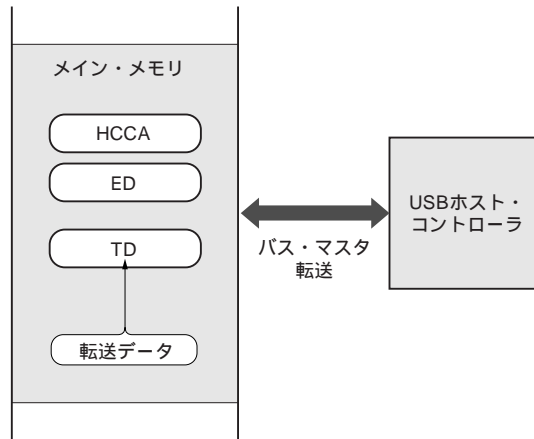


図2 パソコン向けOHCIによるデータ転送

- タを配置するための専用メモリ・アロケータを開発する
- (2)共有メモリとメイン・メモリの間のデータ転送をソフトウェアによって行う

専用メモリ・アロケータ

SH7760内蔵OHCIの共有メモリを“オンチップ・メモリ”と呼ぶことにします。筆者はオンチップ・メモリ専用のメモリ・アロケータ `oc_malloc()` を作りました(リスト1)。各関数の役割を表2に示します。

また, Linuxの各ドライバが呼び出す関数を, オンチップ用の `oc_` シリーズに置き換えるためのマクロをリスト2に示します。このマクロによって, Linuxの各ドライバは8Kバイトの共有メモリ空間上に各データ構造を配置することになります。

専用メモリ・アロケータへ置き換える

上記の `onchipmem.h` ファイルは, 次のインクルード・

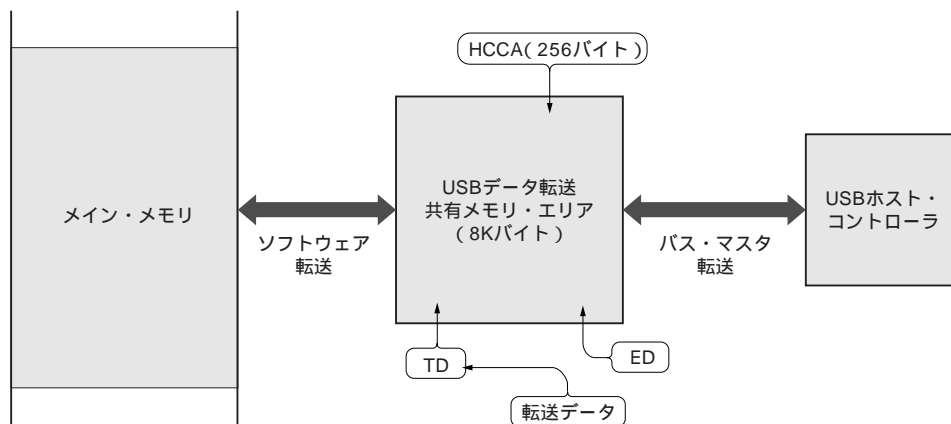


図3 SH7760 内蔵のOHCIによるデータ転送