

# PowerPC による 最小構成 Linux システムの 構築事例

日 櫻 英 孝

ここでは CPU として PowerPC アーキテクチャである MPC5200 を搭載した超小型 CPU ボードを使った Linux システムの構築事例について解説する。CPU ボードについても、Ethernet などのネットワーク機能や HDD (ハード・ディスク装置) などのストレージを持たない、フラッシュ ROM と DDR-SDRAM、そしてコンソール入出力としてのシリアル・インターフェースだけをもった、最小構成ボードとなっている。たったこれだけの CPU ボードでも、Linux が起動する。  
(編集部)

## ターゲット・ハードウェア

今回使用したターゲット・ハードウェアは、DIMM 形状のソケットに挿入して使う CPU モジュール TB0286(タンバック製)です。この CPU モジュールには 64M バイトの DDR-SDRAM と 8M バイトのフラッシュ ROM が実装されており、このモジュールだけで Linux を動かすことができます。

とはいえモジュール単体では扱いにくいので、シリアルを 2 チャンネルと電源コネクタやリセット・ボタンを実装したベース・ボードを用意し、このベース・ボードに CPU モジュールを差し込んで使います(写真 1)。電源として DC (直流)3.3V を供給します。

TB0286 に搭載されている CPU は、MPC5200B( Freescale Semiconductor 社製)です。CPU コアは e300 と呼ばれる 603e シリーズのコアです。603e といえば思い出すのが BeBox です。BeBox に Dual で採用されたのがこの CPU です。MPC5200B はオートモーティブ・コントローラとい

うカテゴリに分類されており、メモリ・コントローラやシリアル、Ethernet、CAN コントローラなども内蔵した自動車産業向けのシステム LSI です。もちろん自動車関係だけでなく、組み込み用の CPU として広く利用されています。CPU の詳細は MPC5200B のデータ・シートを参照してください。データ・シートは Freescale Semiconductor 社の Web サイトからダウンロードできます。

TB0286 には 8 ビット幅のフラッシュ ROM が一つ実装されています。フラッシュ ROM は Intel 社製で、容量は 8M バイトです。また、RAM として 64M バイトの容量の DDR-SDRAM が実装されています。

## 1. U-Boot の基本的な使い方

TB0286 のフラッシュ ROM にはブート・ローダとして U-Boot が書き込まれています。ここではインストールされている U-Boot をそのまま使うので、まずは U-Boot の簡単な使い方を説明します。

### シリアルの接続

Windows から接続する場合は、ハイパーターミナルまたは TeraTerm などの端末ソフトウェアを使います。シリアル・ポートの設定は次のようにします。

- 通信速度：115200bps
- データ長：8 ビット
- パリティ：なし
- ストップ・ビット：1 ビット
- フロー制御：なし

クロス・ケーブルで Windows パソコンと接続して CPU

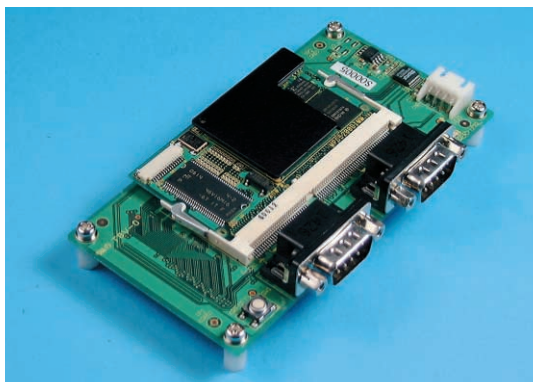


写真 1 LIMM-MPC5200B( TB0286 + ベース・ボード )の外観  
入手先：メディアラボ(株) <http://www.ml.b.co.jp/>

ボードに電源を入れれば、U-Boot が起動してプロンプトが表示されます。

Linux から接続する場合は、端末プログラムとしては cu が一般的です。もちろん minicom のような高機能の端末ソフトウェアを使用してもよいでしょう。Debian の場合は apt-get コマンドで cu をインストールすれば、cu コマンドを使用できます。

Windows の場合の説明と同じように、クロス・ケーブルで Linux パソコンと接続し、

```
# cu -l ttyS0 -s 115200
```

を実行してから CPU ボードの電源を入れます。

筆者が接続しているホスト・マシンは、Windows XP にメディアラボ製の MLD mini4.0D を入れたものです。mini 4.0D はカーネルに coLinux を採用した Windows 上で動作する Debian Sarge ベースの Linux です。コンパイルは coLinux で行い、端末ソフトウェアとしては Windows XP 上で TeraTerm を使っています。ここで解説する内容は、この環境で動作を確認しています。

### MBAR による内部レジスタのアクセス例

MPC5200B の内部レジスタは、MBAR( Memory Base Address Register )というレジスタの内容をベース・アドレスとして、各レジスタのオフセットを加えたアドレスにマップされています。リセット時の MBAR の値は 0x00008000 ですが、U-Boot が起動時に 0x0000f000 に変更します。MPC5200B は MBAR の下位 16 ビットをベース・アドレスの上位 16 ビットとして使うので、実際のベース・アドレスは 0xf0000000 になります。

U-Boot にはメモリをリード (md)、ライト (mw) するコマンドが用意されています。メモリへのアクセスは .l、.w、.b を付加することで 4 バイト単位、2 バイト単位、1 バイト単位でアクセスできます。アドレスは 16 進数で指定します。内部レジスタの先頭をリードしてみると図 1 のようになります。MBAR レジスタ自身はオフセット 0 なので、0xf0000000 番地の値 0x0000f000 が現在の MBAR の内容になります。

0xf0000004 番地から 0xf0000064 番地まではチップ・セレクトの設定レジスタです。0xf0000004 番地には CS0 のスタート・アドレスが、0xf0000008 番地には CS0 の最終アドレスが設定されています。これらのレジスタも MBAR と同じように上位 16 ビットは予約で下位の 16 ビットの値をアドレスの上位 16 ビットとして使います。従って、CS0 は 0xff800000 から 0xfffffff までの 8M バイトとなります。ここには TB0286 のフラッシュ ROM が実装されています。

0xf0000034 番地には SDRAM のチップ・セレクトの設定が書かれています。上位 12 ビットがアドレスで下位の 5 ビットがコード化された SDRAM のサイズです。SDRAM は 0 番地にマップされ、サイズのコードは 0x19 で 64M バイトであることを示します。MPC5200B 用の Linux では、このレジスタの内容をもとにメモリの認識を行っています。もちろんメモリの認識は CPU によって異なるので、CPU が異なる場合は別の方法でメモリを認識しています。

U-Boot が行ったハードウェアの初期設定は、メモリ・マップされたこれらのレジスタを読み出すことで確認でき、変更することも可能です。

MPC5200B の GPIO に LED が接続されていれば、U-Boot で対応する内部レジスタにアクセスすることで、LED の点灯を ON/OFF することができます。しかし残念ながら、TB0286 にはこのような LED は実装されていません。

### flinfo コマンド

flinfo コマンドを使ってフラッシュ ROM のセクタのアドレスを表示することができます。フラッシュ ROM にカーネルを書き込んだり、セクタを消去するときにアドレスを指定しますが、そのときに参考になります。

CS0 の設定レジスタで見たとおり、フラッシュ ROM は 0xff800000 からマップされています。TB0286 の U-Boot は 0xffff0000 からの 2 セクタに書き込まれ、0xffff40000 から 1 セクタが U-Boot の環境変数となっています。アドレスの後ろに (RO) と表示されたセクタはリード・オンリーでライト・プロテクトがかかっていることを示します。U-Boot のセクタは、そのままでは書き換えができないようにライ

図 1  
U-Boot のメモリ・ダンプ・コマンド  
の例

```
# md.l f0000000
f0000000: 0000f000 0000ff80 0000ffff 0000ff00 .....
f0000010: 0000ff7f 0000ffff 0000ffff 0000ffff .....
f0000020: 0000ffff 0000ffff 0000ffff 0000ffff .....
f0000030: 0000ffff 00000019 00000000 0000ffff .....
```