

オリジナル仕様ハードウェアの デバイス・ドライバの 書き方

作川 守,
稲岡 一弘

Linuxの世界でデバイス・ドライバを用意する場合、世の中の誰かが作ったソースがないか、まずはネット上を探すことが多い。しかし、組み込みシステム開発評価キットのように、独自仕様のハードウェアの場合は、新たに作らなければならない。とはいえ、一から作るのではなく、仕様の近いコントローラ用のドライバをベースにして、仕様の異なる部分を含ませ込んでいくのが効率的だろう。ここではフレーム・バッファ・デバイス対応のディスプレイ・ドライバと、PS/2 マウス対応のマウス・ドライバを作成する。
(編集部)

プラットフォームはオリジナル仕様ハードウェア

CQ 出版社から発売されている、組み込みシステム開発評価キット(愛称: BLANCA)は、FPGA 搭載のマザーボードで、各コントローラの IP コアが HDL ソース付きで添付されています。そのままの仕様で動かすことや、オリジナル仕様のハードウェアを組み込んで動作させることができます。

CPU アーキテクチャに依存しないプラットフォームになっており、MicroBlaze, M32R といった CPU を使うことができます。近日中にはオプション CPU カードとして SH や

MIPS, PowerPC, Blackfin などが登場する予定とのことです。

その中でも、標準でネットリスト(NGC ファイル)が添付される M32R ソフト CPU コアは、32 ビット RISC CPU に、MMU やキャッシュを含めて一つの FPGA に入る小型の CPU コアです。

BLANCA に標準で実装される LAN コントローラや画面表示コントローラなどの各種周辺コントローラは、市販されている周辺コントローラ LSI の仕様に準拠していない仕様になっています。従って、インターネット上を探して

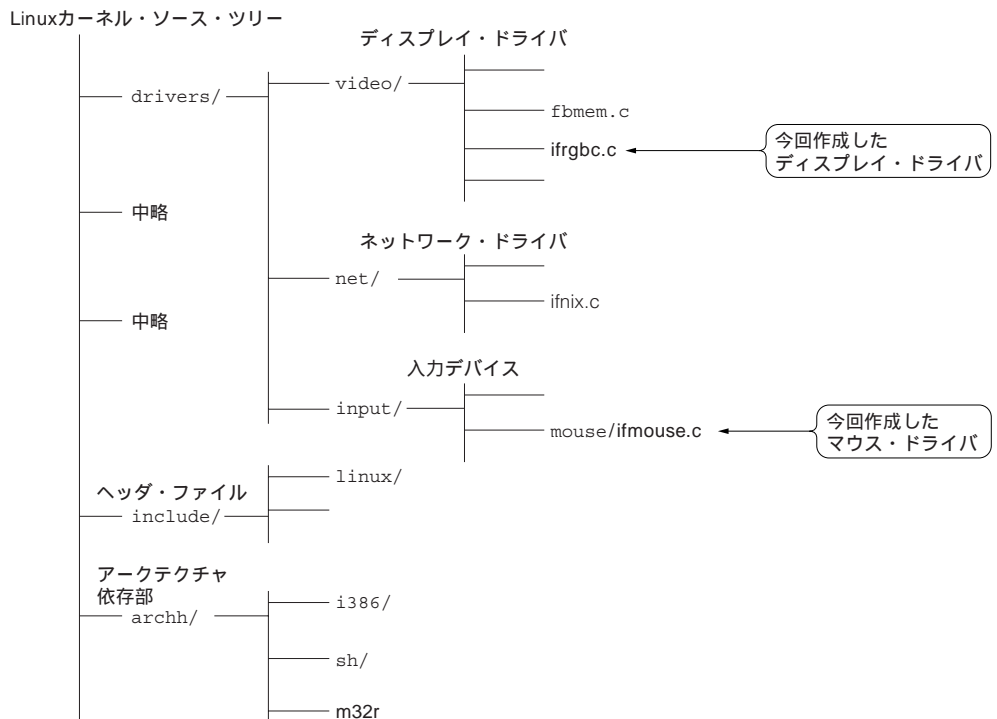


図1 Linux カーネル・ディレクトリ構成

もこれらの仕様に準拠したデバイス・ドライバは基本的に存在しません。これらのコントローラをLinux上から制御できるようにするには、デバイス・ドライバを新たに作成しなければなりません。

筆者らは、2005年12月ごろよりBLANCAにM32R用Linuxのポータリングを始めており、現在はデバイス・ドライバを拡充しています。本稿では、オリジナル仕様のハードウェアをLinux上から制御する場合のポイントについて、Linux/M32Rを例に解説していきます。

1. デバイス・ドライバの組み込み

カーネル・ソースのディレクトリ構成

まずはLinuxのカーネル・ソース・ツリーを見てみましょう。図1はデバイス・ドライバを書く場合に、特に関係するディレクトリを示しています。

デバイス・ドライバを作成するにあたり、まずそのドライバの種類から、関係しそうなディレクトリを探し出します。ネットワーク・ドライバならdrivers/net、ビデオ・ドライバならdrivers/videoのように探し出します。似た仕様のものを探して使うと一番の近道になります。今回紹介するドライバは、BLANCAのシステム・バス上に接続されるIPコア用です。そこで、PCIバスやISAバスのインターフェースを持つチップやカード用のドライバではなく(それをベースにしても構わない)、組み込み用CPU

に内蔵されているものやCPUのローカル・バスに接続されるタイプを探します。そしてそれを修正する方が簡単です。

ドライバをカーネルへ組み込む手順

Linuxカーネルへ新たにドライバを追加するには、下記に記す三つの手順が必要です。

- (1)Kconfigへのコンフィグレーションの追加
- (2)アーキテクチャ依存の初期化処理の追加
- (3)ドライバ本体(アーキテクチャ非依存)の作成

また、作成したデバイス・ドライバを使うためには、カーネルに組み込むという方法と、モジュールとして組み込むという方法の2通りがあります。本稿では、カーネル自身に組み込む方法について説明します。

Kconfigへのコンフィグレーションの追加

カーネルのバイナリを作成する際は、ドライバを使用するか否か、また使用する場合は組み込んで使用するか、モジュール形式にして使用するかを選択できるようにする必要があります。

Linuxではそれぞれのディレクトリにある、Kconfigというファイルを参照してコンフィグレーションを決定します。

まずはKconfigを編集してドライバを追加するところを見ていきます。例として、IFNICというネットワーク・ドライバを追加する場合を挙げます。ネットワーク・ドライバなので、drivers/net/Kconfigに新しいドライバの記述を追加します(リスト1)。

Column 1 クロス開発環境

Linux/M32R用のカーネルのコンパイルは、通常ホスト・パソコン上のクロス開発ツールを使用します。Linux/M32Rは、Debianのパッケージ・システムを利用して、Linux/M32R上で利用するアプリケーションのバイナリやソースを提供しています。クロス開発ツールも同じです。従ってホスト・パソコンにDebian/GNU Linuxシステムを使用していれば、ほかのアプリケーションのインストールと同じようにクロス開発環境をインストールすることができます。

ここではapt-getコマンドを使ったインストール方法を紹介します。

(1)apt-lineに<http://www.linux-m32r.org/>のサイトを登録する

/etc/apt/sources.listに下記の1行を追加する

```
deb http://debian.linux-m32r.org/etch main
```

(2)パッケージの検索先情報を更新する

```
# apt-get update
```

(3)パッケージのインストール

コンパイラの場合

```
# apt-get install gcc-4.1-m32r-linux-gnu
# apt-get install linux-kernel-headers-
m32r-cross
```

デバッグの場合

```
#apt-get install gdb-m32r-linux-gnu
```

以上でその時最新のクロス・ツールをインストールできます。