

第4章

ルート・ファイル・システムからクロス開発環境の構築、
カーネルのビルドまで共通カスタムLinux
ディストリビューションの開発

石原 ひでみ Hidemi Ishihara

第2章や第3章のFPGA開発では、どうしてもそれぞれAltera社/Xilinx社用のFPGA開発ツールを使わざるを得ませんでした。しかしLinuxカーネルは、どちらも同じものを使うことができます。ここではクロス開発環境の構築からルート・ファイル・システムの作成、U-BootとLinuxカーネルのビルド方法について解説します。もちろんAltera社/Xilinx社の区別なく、どちらでも通用する手順を紹介します。

1. 自前で環境構築する狙い

本章ではDE1-SoC (Cyclone V SoC) とZYBO (Zynq) の両方で動作する、カスタムLinuxディストリビューションとクロス開発環境を構築します。ハードウェアからは逸れた内容になりますが、Cyclone V SoCやZynqを使ったシステムを組むとなると、OSやアプリケーションなどソフトウェアの話を避けては通れません。

● 純正BSPは提供されている

Altera社とXilinx社は、共にBoard Support Package (BSP) という形でLinuxディストリビューションを配布しています。Altera社はRocketboardの、Xilinx社はPeta Linuxが提供しているLinuxディストリビューションが純正BSPに当たります。

Linuxを組み込む場合「○○BSPをダウンロードして…」という説明をよく見ますが、BSPを使う理由は明白です。Linuxディストリビューション環境を使用すると、既に用意された多数のアプリケーションをすぐに使用できるからです。これらの環境を最初から構築しようとする、とてつもない労力を要します。それを解決してくれるのがBSPだからです。

● クロス開発環境も用意されているが

BSPはターゲット・デバイス向けのバイナリやパッケージだけで構成されることが多く、この他にホストPC上でアプリケーションを開発するためのクロス開発環境が必要になります。クロス開発環境はSDK (Software Development Kit) と呼ばれ、Altera社ではSoC EDK, Xilinx社ではVivado SDKが提供されています。

● 純正ツールは使わずに自前で構築

本特集では、Cyclone V SoCとZynqの両方で共有して使用できるカスタムLinuxを実現するため、ベンダが用意しているBSPやクロス開発環境を使わず、

Yocto Projectを使用して、自前でカスタムLinuxディストリビューションを構築し、そのカスタムLinuxディストリビューションに対応したSDKも構築します。

ベンダ提供の純正BSPやSDKではなく、なぜわざわざ独自のBSPやSDKを構築するかというと、Cyclone V SoCとZynqの両方で使えるのはもちろんですが、使用したいソフトウェアの依存関係の解決にあります。

● ライブラリの依存関係

オープン・ソース・ソフトウェアは、さまざまなライブラリを使用して作成されています。これらのソフトウェアは他のソフトウェアのバージョンに依存しているケースが多々あります。BSPは作成された当時のバージョンで構成されることが多く、なかには採用されているソフトウェアのバージョンが古いケースもあります。新しいソフトウェアを追加しようとしたとき、その依存関係を解決しなければインストールできない場合もあります。

例えば、Peta Linuxを使用してZynq用のLinux環境を整えたとしましょう。ここで、Aという新しいソフトウェアをインストールして使うことになったので、Aをビルドしてインストールすることになります。しかしAのビルドのためには「libBのバージョンMM.NNが必要である」とエラーが出力されることがあります。そこでAをビルドする前に、libBのバージョンMM.NNをビルドしてインストールしなければなりません。libBをビルドすると今度は「libCのバージョンPP.QQが必要である」というエラーが発生するといったような、イタチゴッコになった方も多いと思います。

● バージョンが一致したヘッダ・ファイル

さらに、必要なライブラリは既にインストールされているが、そのライブラリのバージョンと一致しているヘッダ・ファイルが必要になる場合もあります。ラ