

FPGA部分にオリジナル回路を実装してネットワーク上から制御してみる アルテラSoC評価ボードHelioによるPWM点灯制御のハード&ソフト構築

鳥海 佳孝 Yoshitaka Toriumi

FPGA マガジン No.5で、Cyclone V SoC搭載評価ボードHelio (アルティマ)をターゲット・ボードとして、開発環境の構築からHelioボードでLinuxを起動させるまでの手順について解説しました。しかしCyclone V SoCのFPGA部分はいわゆる空っぽの状態であり、ここを活用しないとアルテラ SoCを使う意味はありません。そこで続編となる今回は、FPGA部分にオリジナルの回路を実装して、ネットワーク上からそれを制御する事例について詳しく解説します。

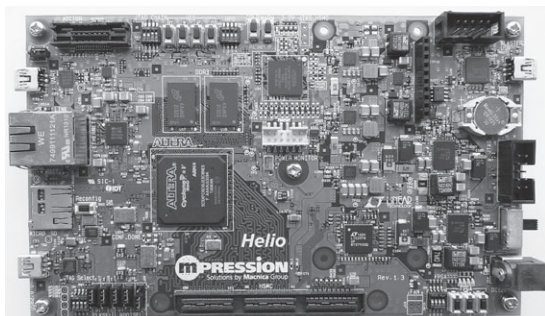


写真1 ボードHelioの外観
問い合わせ先：(株)アルティマ

● 前回までのおさらい

筆者はアルテラ SoC 評価ボード Helio (アルティマ, 写真1) を使い、前回 (FPGA マガジン No.5 特集第2章 アルテラ SoC 評価ボード Helio で Linux を動かそう) までに、次のところまで準備することができました。

- (1) microSD カードに Helio で Linux がブートするのに必要なイメージ・ファイルの書き込み
- (2) VMware による Linux ホスト・マシン (Vine Linux) の構築
 - DHCP サーバの設定
 - Samba サーバの設定
 - FTP サーバの設定
 - NFS サーバの設定
- (3) クロス・コンパイラ (CodeSourcery) のインストール
- (4) クロス・コンパイルしたプログラムの実行
 - Hello World の実行
 - GNU アプリケーション (busybox, thttp) の実行

● 今回の目標

今回は上記の内容が構築されている前提で話を進めていくので、まだここまで準備ができていない方は、前回の内容を構築してから進めてください。取り上げる内容は前回約束したように、FPGA 部分を使用してカスタムのハードウェアを追加し、追加したハードウェアに対してカスタムの Linux デバイス・ドライバ

を作成してのアクセスです。これができないことには、ARM Cortex-A9 デュアルコア内蔵 FPGA のうまみが出せません (というよりここが一番面白いところ!)。それでは早速その目標に向かって進めていきましょう。

1 カーネルの入手とコンパイル

● カーネルのダウンロード

次のコマンドで Helio 用のカーネルを入手します。前回構築した Vine Linux に、次のプログラムをインストールしておいてください。

```
[tori@Vine621 ~]$ sudo apt-get install
git-core texi2html chrpath
```

インストールできたら、Yocto の環境を使って Linux カーネルを入手します。次の手順で行います (詳細は <http://www.rocketboards.org/foswiki/Documentation/GSRDGettingStartedYoctoCopyArrowSoCKitEdition> を参照)。

```
[tori@Vine621 ~]$ mkdir Helio
[tori@Vine621 ~]$ cd Helio
[tori@Vine621 Helio]$ wget http://
releases.rocketboards.org/release/
2013.11/gsrdsrc/linux-socfpga-gsrds
13.1-src.bsx
[tori@Vine621 ~]$ chmod +x linux-
socfpga-gsrds-13.1-src.bsx
[tori@Vine621 ~]$ sudo ./linux-
socfpga-gsrds-13.1-src.bsx
[tori@Vine621 ~]$ /opt/altera-linux/
bin/install_altera_socfpga_src.sh ~/
yocto
[tori@Vine621 ~]$ source ~/yocto/
altera-init ~/yocto/build
[tori@Vine621 ~]$ bitbake virtual/
kernel
```

ダウンロードやインストールにかなりの時間がかか