

定番ZYBOボードによるハードウェア制御…その④

ARM FPGAにつきもの! ハードの変更に伴うデバドラの修正

鳥海 佳孝

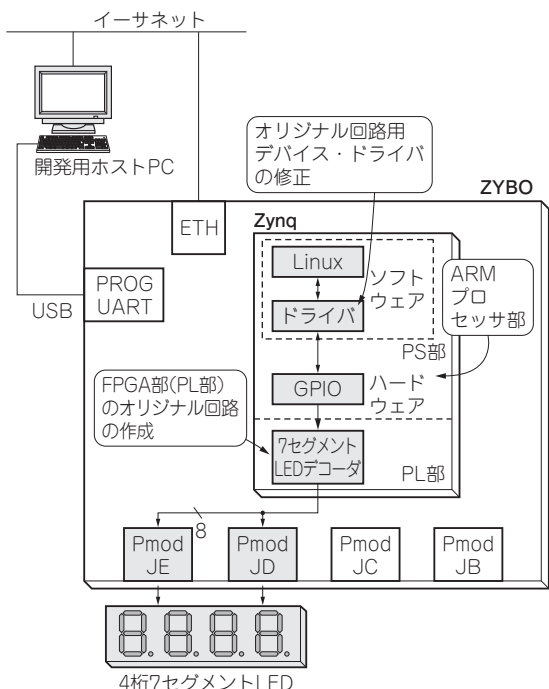


図1 今回解説すること…FPGA部のオリジナル回路を変更してデバイス・ドライバを対応させる

このコーナーでは、ARMプロセッサとFPGA (Field Programmable Gate Array) が1チップになったザイリンクスのZynqと、アルテラのSoC (Cyclone SoCやStratix SoC) を対象に、うまく使う方法や、さまざまな話題を取り上げていきます。

2016年4月号～6月号では、FPGA部を使ったハードウェア設計からLinux用デバイス・ドライバ開発までを説明してきました。今回は、FPGA部のハードウェアを変更した際につきまとうLinuxデバイス・ドライバの修正について説明します。(編集部)

ターゲット回路

今回は、4桁の7セグメントLEDをダイナミック点灯させます(図1)。

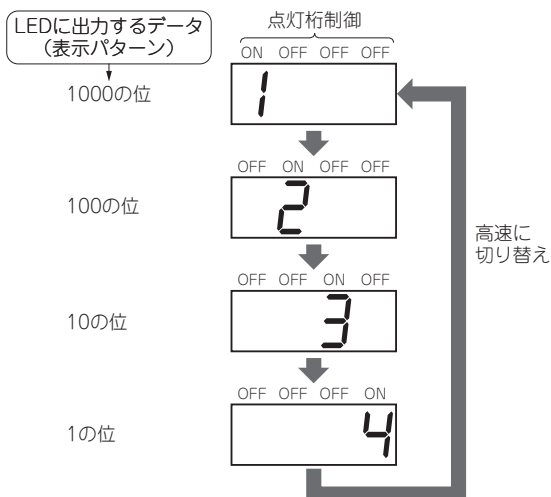


図2 7セグメントLEDのダイナミック点灯
表示する桁を高速に切り替えながら1桁ずつ点灯し続ける。人の目には残像効果ですべての桁が表示しているように見える。少ない信号線で実現できる利点はあるが、常に制御し続ける必要がある

● 連続動作し続ける機能をハードウェア化してプロセッサの負担を減らす

ダイナミック点灯の動作を図2に示します。時間軸の制御(ダイナミックな桁選択)やキャラクタのデコードを、全てハードウェアに任せることによって、ソフトウェア側の負担を減らすことができます。

同じようなことをラズベリー・パイなどでもできますが、GPIOを使ったソフトウェア処理ではプロセッサに負荷がかかりますし、ハードウェアで実現するとLED以外の回路が物理的に必要になるので、システム全体が大きくなってしまいます。しかし、ZynqのようなARM FPGAであれば、チップ内のFPGAブロックを活用できるので、外付け回路を一切追加せずに実現できます。

● さすがARM FPGA! チップ内の配線なら自由度が高い

今回はドライバ側に飛んできたキャラクタ型の値を単に出力するだけにします。ぜいたくにインター