



LED とスイッチを Linux から制御できる

簡単で基本的なデバイス・ドライバを書いてみよう!



古俣 学

本章では、最も基本的な Linux のデバイス・ドライバの例として、汎用 I/O (以下 GPIO) に接続した LED の点灯状態を変更したり、ディップ・スイッチの状態を取得したり、スイッチを押した瞬間に割り込みを発生させて割り込みの回数をカウントするプログラムを作成してみましょう。(筆者)

1. GPIO サンプル・プログラムを試す

デバイス・ドライバを書く前に、まず GPIO を使ったサンプル・プログラムの動作をシミュレータ Virtual Platform Analyzer (VPA) を使って試してみましょう。なお、GPIO サンプル・プログラムについての詳細は、Appendix 3 を参照してください。

ここで説明するハードウェアの仕様を簡単に説明します。

- LED 点灯用出力：ポート B のビット 21 ~ 18 (負論理)
 - ディップ・スイッチ入力：ポート B のビット 25 ~ 22 (負論理)
 - タクト・スイッチ入力：ポート A のビット 24 (負論理)
- それでは、GPIO サンプル・プログラムを実行してみ

```
load_image gpio.elf
```

図1 VPA 上で GPIO サンプル・プログラムをロードする

ましょう。まず、VPA のインストール・フォルダ内の simulation フォルダに gpio.elf をコピーしてから VPA を起動します。起動したらメニュー・バーの CQmagazine → Sim-Start をクリックして、シミュレータ本体を起動します。VPA に図1のコマンドを入力します。

GPIO サンプル・プログラムをロードしたら、VPA のインストール・フォルダ内の skins¥VPA_GUI フォルダにある ATMEL_ARM9.exe を起動します(図2, 図3)。

実機 BLANCA ARM9 用 GUI アプリケーションが起動したら、VPA の緑色の三角形のアイコンをクリックしてシミュレーションを開始します。

まず、GUI アプリケーションのディップ・スイッチの一番左をチェックした状態でスイッチをクリックすると、クリックした回数をカウント・アップして LED を表示します(図4)。これはスイッチを押すことで GPIO 経由で割り込みが発生し、その割り込みを内部でカウントしつつ、カウント結果を LED に表示しています。



図2 実機 BLANCA ARM9 用 GUI アプリケーション ATMEL_ARM9.exe

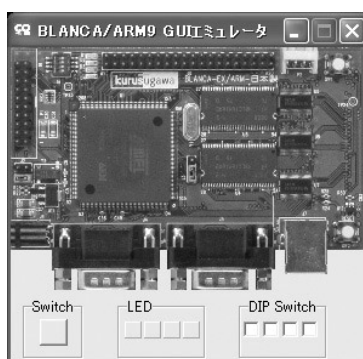


図3 実機 BLANCA ARM9 用 GUI アプリケーション ATMEL_ARM9.exe の起動画面

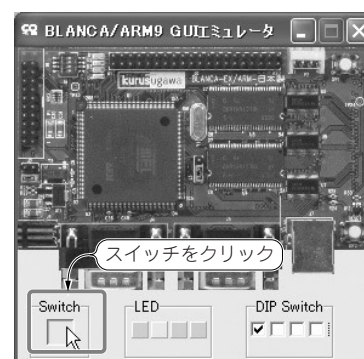


図4 スイッチを押すごとに LED がカウントアップされる