

LED点滅でμITRON  
アプリケーション作成を実践する

中村 建真

そもそも、TOPPERS/JSPの移植はアプリケーションの開発が目的です。ここまでの章で移植が済んだTOPPERS/JSP for LPC2388を使って、簡単なアプリケーションの開発を行いましょう。  
(筆者)



## 1 サンプル・プログラム Sample1

## ● サンプル・プログラム Sample1 のビルド

TOPPERS/JSPには、あらかじめ単純なサンプル・プログラム Sample1 が用意されています。このアプリケーションはキャラクタ端末を通してユーザと対話する小さなプログラムです。ユーザはこのアプリケーションにコマンドを送ることでTOPPERS/JSPの機能を体験できます。Sample1については、すでに簡単に紹介しましたが、自分自身でアプリケーションを開発するきっかけとして、改めてビルドして動かしてみましょう。

Sample1アプリケーションをビルドするために、まずアプリケーションのためのディレクトリを用意しましょう。図1にディレクトリ構成を示します。親ディレクトリがworkspaceになっていますが、もちろんこのディレクトリの名前は好きなものでかまいません。ここに移植を行ったjsp\_studyを置きます。そして、jsp\_studyと並んで新しくsample1ディレクトリを作ります。

次に、TOPPERS/JSPのソース・ツリーjsp\_studyの中に降りて、コンフィギュレータをビルドします。

ビルドが成功したら、今度はsample1の中へ移動して、サンプル・プログラムSample1を作ります。

図1  
Sample1の  
ディレクト  
リ構成

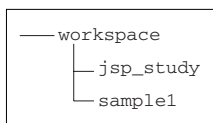


図2  
コンフィギュレータと  
Sample1のビルド

この一連のコマンドの流れを図2に示します。このコマンドの中で呼び出されている../jsp\_study/configureは、コンフィギュレータを呼び出すスクリプトです。パラメータの-cと-sは、それぞれCPU依存部とシステム依存部を指定しています。このスクリプトの使い方の詳しい説明は、jsp\_study/doc/user.txtを参照してください。

注意深く見ればわかることですが、configureスクリプトのパラメータはターゲットとなるシステムを指定しているだけで、アプリケーションのソース・コードを一切指定していません。この場合、configureスクリプトはTOPPERS/JSPのソース・ツリーに含まれているSample1アプリケーションのソース・コードを現在のディレクトリに展開して、コンフィギュレーションを行います。つまり、サンプル・プログラムを自動的に作るわけです。

## 2 コンフィギュレーション

## ● コンフィギュレーションにより資源を自動生成する

コンフィギュレーションとは、タスク、セマフォ、割り込みハンドラなどTOPPERS/JSPアプリケーションが使用

```

● コンフィギュレータのビルド
$ cd jsp_study/cfg
$ make depend
$ make
● Sample1のビルド
$ cd ../..
$ ../jsp_study/configure -C armv4 -S cq_frk_nxp_arm
$ make depend
$ make
  
```

CPU依存部      システム依存部