

C言語/OS/ICEを使って最先端の開発にチャレンジ

新世紀 マイコン教室

〈第2回〉 シミュレータで体験する
マルチタスク

北野 優
Masaru Kitano

前回(第1回)は、本連載の趣旨と概要について説明しました。今回(第2回)と次回(第3回)は、本誌2004年6月号の付録CD-ROMに収録したSmalight OS(体験版)とE7エミュレータなど、今後使用するツールの感触を確かめます。

連載第4回からは実際の装置を設計、製作して、電子機器の開発過程を皆さんとともに体験して行きます。実際の装置として、簡単な構成でありながらいろいろな要素技術があり、かつ、ある程度は役に立つものとして、サーミスタで温度を計測してファンの回転数をコントロールするファン・コントローラを製作する予定です。

本連載はマイコンを主体にしているのですがどうしてもソフトウェア偏重になりますが、その中でもドキュメント整備やプロジェクト管理、モデリング技術などの概念の一端や、Cコンパイラの性能などについても解説する予定です。

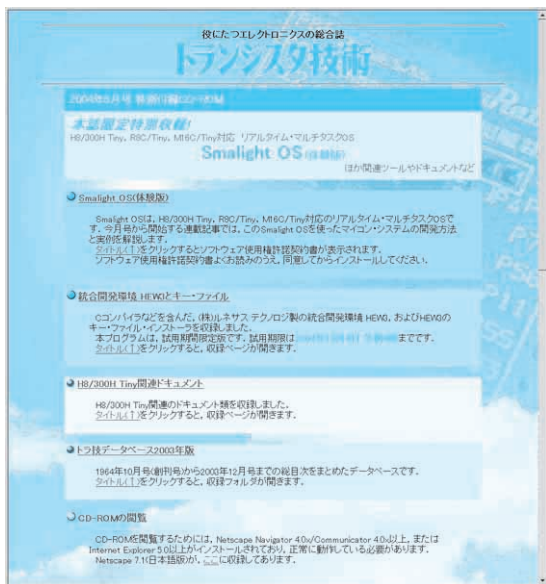


図2-1 2004年6月号の付録CD-ROMのトップ画面

カーネル起動実験

■ 実験準備(インストール)

2004年6月号の付録CD-ROMに収録されているSmalight OS(体験版)を起動してみましょう。

今回は、HEW3試用期間限定版に入っている、パソコン上で動くH8マイコンのシミュレータを使って、Smalight OS(体験版)を動かしてみます。

この方法ならば、ハードウェアのセットアップや不具合などを考えずに、純粋にソフトウェアの動作を確認することができます。

● Smalight OS(体験版)とは

Smalight OS(体験版)は、6月号の付録CD-ROM(図2-1)の説明にあるとおり、(株)ルネサス北日本セミコンダクタから本誌読者の皆さんに向けて特別に提供していただいたものです。

製品版との違いは、6月号の本連載のコラムに記されているとおり、若干の性能と機能に制限があるだけで組み込みOSの入門には何ら不都合のないものです。

● OSの信頼性と性能

OSといえども人間が作ったソフトウェアなので、皆さんお使いのパソコンOSに限らずバグが存在する可能性は否定できません。特定の業界には、RTOSのバグのため〇〇地獄(〇〇にはRTOSの名前が入る)という言葉すらあります。

Smalight OSは、経験豊富なメーカーが限定した機能に絞り込みコンパクトに構築したOSです。したがって、連載で使用するH8/Tinyの少ないRAMでも実装でき、かつ十分信頼できるOSです。

体験版は、拡張性とスピードに若干の制限はありますが、スピードに関してはもともとが極めて高速であるため、本連載で製作する機器ではあまり問題にはならないでしょう。

● Smalight OS(体験版)のインストール

インストールは6月号の付録CD-ROMの表紙である図2-1から「Smalight OS(体験版)」のタイトルをクリックして現れる同意書に同意できる方だけがインストールできます。

インストールすると、図2-2のような三つのフォルダが現れます。これらのフォルダにはそれぞれルネサス テクノロジーのマイコンであるH8用、M16C用、R8C用のSmalight OS(体験版)が収録されています。

今回は2004年4月号付録のH8/Tinyを使用するのでH8用の300Htr_v103をクリックしてsetup.EXEを起動します。あとはインストーラの指示に従ってインストールしてください。

● HEW3のインストール

ルネサス テクノロジーのマイコン開発統合環境のHEW3試用期間限定版が2004年6月号の付録CD-ROMに収録されています。試用期限は2004年10月4日午前9時までです。

今回はシミュレータ・デバッガも使用するので、HEW3のインストール時にいっしょにインストールしてください。

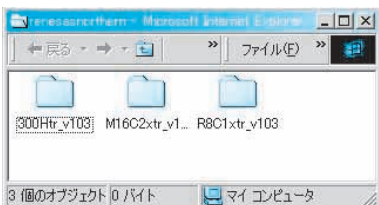


図2-2 Smalight OS体験版格納フォルダ

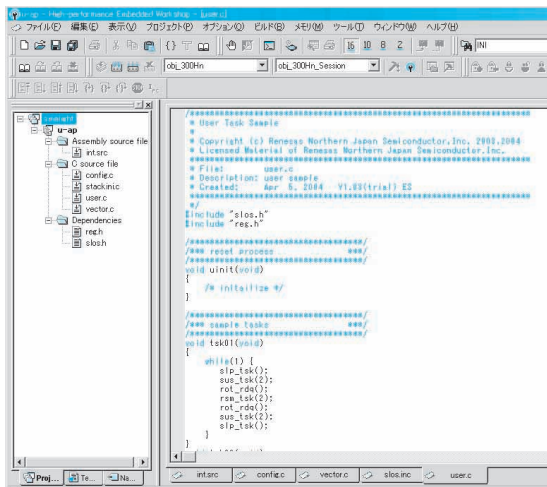


図2-3 サンプル・プロジェクトを開いたようす

リスト2-1 user.cの内容

```

/*****
 * User Task Sample
 *
 * Copyright (c) Renesas Northern Japan
   Semiconductor, Inc. 2003,2004
 * Licensed Material of Renesas Northern Japan
   Semiconductor, Inc.
 *****/
 * File:      user.c
 * Description: user sample
 * Created:   Apr 5, 2004  V1.03(trial)ES
 *****/
*/
#include "slos.h"
#include "reg.h"

/*****
/** reset process      ***/
/*****
void uinit(void)
{
    /* initialize */
}

/*****
/** sample tasks      ***/
/*****
void tsk01(void)
{
    while(1) {
        slp_tsk();
        sus_tsk(2);
        rot_rdq();
        rsm_tsk(2);
        rot_rdq();
        sus_tsk(2);
        slp_tsk();
    }
}

void tsk02(void)
{
    while(1) {
        rot_rdq();
        wup_tsk(1);
        rot_rdq(1);
        rot_rdq();
    }
}

/*****
/** sample interrupt proc      ***/
/*****
/*
 * normal
 */
#pragma nogsave(intProc1)
void intProc1(B *sp)
{
    /* interrupt process */
}

/*
 * return to dispatch
 */
#pragma nogsave(intProc2)
void intProc2(B *sp)
{
    if(cTCB != 0)
        TCBsp[cTCB-1].sp = sp;

    /* interrupt process */

    disp();
}

```

起動時にOSから呼ばれる
イニシャライズ・ルーチン。
初期化に必要なI/Oなどの
処理をここに記述する。

タスク1, 2のサンプル。
イニシャライズ後OSが起
動すると、OSから実行権
を渡される。
各タスク関数がOSなしプ
ログラムのmainに相当す
る。

割り込み処理のサン
プル。
今回は使わないので
放置。
次回で使用。