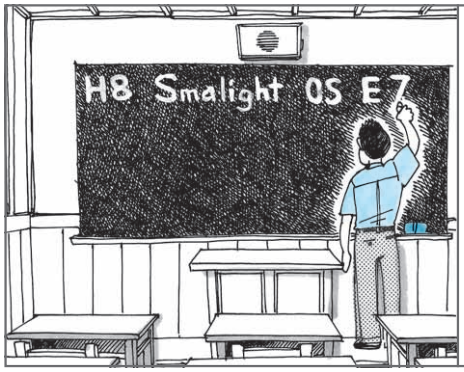

C言語/OS/ICEを使って最先端の開発にチャレンジ

新世紀 マイコン教室

〈第9回〉ファン・コントローラのソフトウェア設計

北野 優
Masaru Kitano

前回までにハードウェア部分の設計が完了しました．今回からはファン・コントローラのソフトウェア部分を設計・制作していきます．

今回は，ソフトウェアの設計について解説します．

ソフトウェアを機能ブロックに分ける

ソフトウェアに実装する機能を知るためには，ファン・コントローラを機能別に各ブロックに分けていく必要があります．以下に各部について解説します．

● 温度計測部

サーミスタによる温度計測回路の電圧を，H8/3694F内蔵のA-Dコンバータを使ってA-D変換し，連載第6回(2004年11月号)で解説したリニアライズを施して温度を算出します．

● ファン制御部

連載第7回，第8回(2005年1月号，2月号)で解説したとおり，H8/3694Fのタイマ V でPWM波形を発生させ降圧チョッパ回路を駆動し，ファンに供給する電圧を変えてファンの回転数を制御します．

● ファン回転数検出部

連載第8回で解説したファン回転数検出回路でファンの回転信号を取り出し，H8/3694Fのタイマ W のインプット・キャプチャ機能を使ってその周期を計測し回転数を算出します．

表9-1　Smalight OSの特徴

- ITRONライクなAPI
- 小容量コンパクト
- タスク・スケジュール方式：ラウンド・ロビン
- 最大タスク数：127個
- サンプルのベクタ・テーブル，ユーザ・タスク，割り込みハンドラを提供
- タスク状態にRun/Ready/Waitをサポート
- コンフィギュレーション・ファイルによる簡単な構築

● LCD，警報表示部

連載第5回(2004年10月号)で解説したLCDモジュール(キャラクタ・ディスプレイ・モジュール)インターフェースとプログラムでファン・コントローラの状態をLCDに表示します．

また，ファンに十分な電圧を供給しているにもかかわらず，ファンの回転が停止したり極端に低い場合，またはファンの回転の有無にかかわらず検出した温度が異常に高い場合には，LED(赤)を点灯させて異常を知らせるようにします．

また，ファン動作インジケータとしてLED(緑)でファンの回転を表示するようにします．

● 設定値入力部

ファン・コントローラの上限設定温度，下限設定温度を設定する二つの可変抵抗器(*VR*)の設定値をA-Dコンバータに入力して，その電圧値から設定温度値を算出します．

Smalight OS体験版の復習

連載第2回(2004年7月号)でLEDを動作させたSmalight OS体験版を使って，ファン・コントローラのソフトウェアの実装を検討します．

復習をかねて，Smalight OSとはどんなソフトウェアか，また体験版の制約はどのようなものか再確認します．

● Smalight OSの特徴と機能

Smalight OSはルネサス北日本セミコンダクタが開発したTinyマイコン(H8/Tinyシリーズ，R8C/Tinyシリーズ，M16C/Tinyシリーズ)用の非常にコンパクトなリアルタイムOSです．**表9-1**にSmalight OSの特徴を再度示します．

Smalight OSはコンパクトに作られているため，ほかのμITRON仕様のOSなど代表的なリアルタイムOSに比べると簡単な機能しかありません．Smalight

表9－2　Smalight OSのサービス・コール一覧

区分	サービス・コール名称	説明	備考
タスク管理	rot_rdq	タスク・スイッチ	
	slp_tsk	自タスクのスリープ	
	wup_tsk	タスクの起床	
	sus_tsk	他タスクのサスペンド	
	rsm_tsk	サスペンドの解除	
割り込み処理支援	INTPUSH	割り込み発生時のレジスタ退避	アセンブラ・マクロ
	INTPOP	割り込み処理の終了とレジスタ復帰	アセンブラ・マクロ
	disp	割り込み処理をディスパッチして終了	
コールバック	uinit	リセット処理	
	stack_init	タスク・スタックの初期化	
	callback_int	割り込み処理	
スタック操作支援	GET_REG	ユーザ・タスクのスタックからのレジスタ参照	マクロ
	SET_REG	ユーザ・タスクのスタックのレジスタ設定	マクロ

OSのサービス・コールを**表9－2**に示します．

● Smalight OS体験版の制約

連載第1回(2004年6月号)に付録として提供いただいたSmalight OS体験版には，体験版としての制限事項がありました．制限事項を**表9－3**に示します．

機能ブロックをタスクとして Smalight OS体験版に実装する

Smalight OS体験版を使って，ファン・コントローラのソフトウェアを実装できるように検討しましょう．

● 高度な機能がないSmalight OS体験版

Smalight OS体験版には高度なタスク間通信機能や時間管理機能がありません．

このような高度な機能はあれば便利なことは確かですが，小型のワンチップ・マイコン程度の応用ソフトウェアでは，なくても大きな不都合がない場合が大多数だと思います．

それよりも，ワンチップ・マイコンのハードウェア的理由からくる，OSを使用することによるメモリや速度上の制約を減らすことのほうにメリットが多い場合があります．

そのような理由により，Smalight OSは必要最小限の機能が非常にコンパクトかつ高信頼に実装されており，ワンチップ・マイコンにターゲットを絞って設計されています．

● もっとも単純なマルチタスク動作

Smalight OSのタスク管理機能を使ってシステムをどのようにタスク分割するかは，システムの事情や設計者の方針などで変わりますが，おおむね1機能くらいを一つのタスクとするのが妥当でしょう．

そこで，今回は機能ごとに分けたブロックをそのままタスクとしてOSに管理させることにします．例えば，温度計測部は温度計測タスクとして単機能プログラムとして設計します．

すなわち，温度計測タスクは起動時に自分の必要な初期化を行った後，ひたすらサーミスタの電圧をA－D変換して変換値から温度を計算して出力するというループを繰り返すプログラムとして設計します．

この単純なプログラムとして記述するタスクは，後述する共有資源のアクセス以外はほかのタスクの影響を受けません．

したがって，プログラムを記述するときもデバッグするときも，例えば温度計測タスクはLCD表示や設定値入力などのことを気にする必要もなく，デバッグ作業を含めてプログラムの作成が簡単にできます．

このような単純なプログラムを集成してシステム全体を構築できるのが，リアルタイムOS導入の最大のメリットでしょう．

表9－3　Smalight OS体験版の制約事項

性能上の制約	メモリ・サイズが正規版に比べて若干大きくなっている
	タスク・スイッチ時間が大きい
機能拡張の制限	アドイン機能を使ってOSの機能を拡張することができない

● タスク間通信は共有メモリで行う

タスクのなかにはほかのタスクの処理結果を必要とするものや，自分が処理した処理結果をほかのタスクに知らせる必要のあるものがあります．

例えば，温度計測タスクの計算した計測温度値は，計測温度と設定値によりファンの制御を行うファン制御タスクや計測温度を表示するLCD，警報表示部タスクで必要とされます．

このようなタスク間のデータの受け渡しをタスク間