

マルチタスク・プログラムなら  
拡張性もリアルタイム性もバッチリ! 作って合点!

# ついに完成! 自作OSで センサとLCDを制御

豊山 祐一

特集の総仕上げとして、作成したTry Kernel上で動くマルチタスク・アプリケーションを作ります。ラズパイPicoにセンサとLCDを接続し、それらのデバイスを制御するタスクを作成します。



## リアルタイム処理でなくても マルチタスクは便利

### ● プログラムを簡潔に記述できる

RTOSのアプリケーションは、複数のタスクから構成されたマルチタスク・プログラムです。従って、PCの一般的なC言語のプログラムと異なり、プログラムは並列に実行するタスクに分割して作成していく必要があります。

このように書くと大変そうに思えるかもしれませんが、マルチタスクは複数の機能を同時に実現しなくてはならない組み込みシステムにおいて、プログラムを簡潔に記述するための手法でもあります。

マルチタスクを使用する第1のメリットは、第1部でも説明したように高いリアルタイム性能の実現にあります。今回作成するセンサ制御アプリケーションは、さほどリアルタイム性は必要としませんがマルチタスク・プログラムとすることにより、個々のセンサ制御をシンプルなコードで記述できます。さらに実行タイミングなどの時間管理はOSにAPIで指示するだけで済みますので、アプリケーション側は手間がいらなくなります。

マルチタスク・プログラムに一度慣れてしまえば、同じ処理をシングルタスクで記述する方が逆に面倒になってくるはずですよ。



## アプリケーション設計… 手順と構成

### ● マルチタスク・プログラムの設計手順

マルチタスクのプログラムを作成するにはまず、タスクの構成やタスク間の制御を決めなくてはなりません。以下の手順で作成していくとよいでしょう。

#### ▶①タスクの分割・構成の設計

アプリケーションを複数のタスクに分割し、タスク構成を決めます。さらにそれぞれのタスクの優先度や実行するタイミングを決めます。

#### ▶②タスク間の同期・通信の設計

タスク間で行う通信や同期処理を検討し、その実現手段を決めていきます。

#### ▶③タスクの実装とテスト

実際にそれぞれのタスクをプログラミングしていきます。タスクはある程度独立に実行できるので、全てのタスクを一気に作って動かすのではなく、1つまたは少数の関連度の高いタスクだけを実行してテストしていくこともできます。

### ● タスク構成の考え方

アプリケーションをどのようなタスク構成とするかは、マルチタスク・プログラムの設計の肝となる部分であり、難しい部分でもあります。まずは以下の複数の観点から考えていくのが良いと思います。

#### ▶①機能面からの検討

アプリケーションの持つ機能ごとにタスクを割り当てていく考え方で、一番分かりやすい考え方だと思いますが、どの程度の機能の粒度でタスクを割り当てるかは、次項②も考慮しつつ決めます。

#### ▶②実行面からの検討

タスクはプログラムの実行単位であり並行実行が可能です。従って、同時に実行したい機能を独立のタスクに割り当てていく考え方は、たとえ機能面で関連の強い処理であっても、別のタスクとして並行実行した方が良い場合もあります。

#### ▶③資源面からの検討

タスク間で共有する資源の観点からの検討です。タスク間で共有する資源は、グローバル変数であったり、I/Oデバイスであったり、さまざまなものが考えられます。共有する資源は、第4部で説明したセマフォなどの機能により排他制御を行って使用できます。しかし、排他制御を行うと、あるタスクの実行が他のタスクの動作に影響を及ぼしますので、使用しない