第4部 ステップ3…タスクの同期と通信機能

第1章 例えばセンサ制御とデータ処理…優先度をコントロールできる

主に1対1のタスク同期に使われる…起床待ちと起床

豊山 祐一

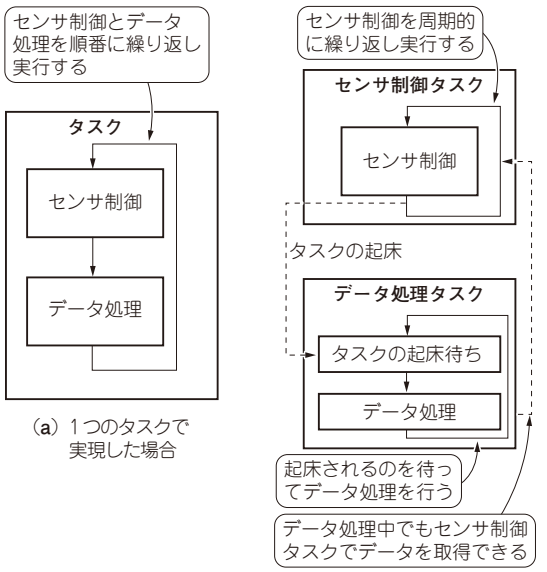

図1 センサ制御とデータ処理の例

第3部ではTry Kernelにマルチタスクの機能を実装し，複数のタスクが同時に実行できるようにしました．ただし，これでは各タスクがばらばらに実行されているだけです．実際のRTOSのアプリケーションは，複数のタスクが連携して動作することにより，アプリケーションの機能を実現していきます．

第4部ではTry Kernelにタスクが連携して動作するための同期と通信の機能を実装します．

● タスクの同期処理の例

タスクの同期が必要となる具体例として，センサを制御してデータを取得し，そのデータに基づいて処理を行うシステムを考えてみます．

このシステムには，センサ制御とデータ処理の2つの仕事があり，その仕事の間には順序があります．

これを1つのタスクで実現すると，**図1**(a)のように2つの仕事を交互に行うことになります．これではデータ処理中にセンサ制御ができませんし，データ処理にかかる時間がセンサ制御の実行開始に影響を与えます．もちろん，両方の仕事にかかる時間が十分に短ければ，1つのタスクで実行しても問題はありませんが，実際にはセンサ制御は実行タイミングが厳しく決められており，一方でデータ処理は処理時間が長くなることがよくあります．

そこで，**図1**(b)のように仕事ごとにタスクを割り当てて並行実行して，これを解決します．センサ制御タスクは決まった周期で実行し，データを取得したらデータ処理タスクを起床します．データ処理タスクは，センサ制御タスクからの起床を待ってデータ処理を行います．

このようにタスクを分割したことにより，それぞれのタスクは独立して動作します．また，センサ制御タスクの優先度をデータ処理タスクより高くしておけば，データ処理タスクが実行中であっても，センサ制御タスクは次のデータを取得できます．

タスク付属同期機能を作成する

● タスクの状態を直接操作するタスク付属同期機能

複数のタスクの動作を合わせて処理を行っていくことをタスクの同期と呼びます．IEEE 2050-2018規格ではタスクの同期を実現するための各種の機能が定められています．その中でも基本的な同期機能は，タスクの実行を一時的に止めたり，再開したりするタスク付属同期機能です．タスク付属同期とはあまり聞きなれない言葉かもしれませんが，タスクの状態を直接操作して同期を行う機能です．

第3部で実装したタスクを時間待ち状態とするAPI tk_dly_tskもタスク付属同期機能の1つです．tk_dly_tskは自タスクを待ち状態にして他のタスクに実行を譲ることができました．

しかし，tk_dly_tskは動作するタスクの指定はできません．これを可能にするのが，本章で実装するタスクの起床待ちAPI tk_slp_tskと起床API