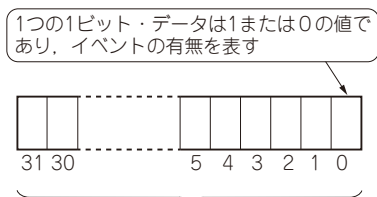


複数のタスク間で複雑な同期もできる…イベント・フラグ

豊山 祐一



イベント・フラグは32個[※]の1ビット・データの集まり

図1 イベント・フラグの構造

注：32ビットCPUの場合

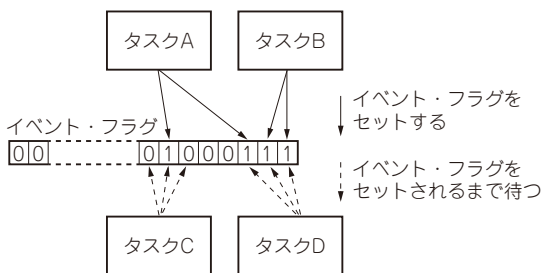


図2 イベント・フラグによる同期の例

第1章では基本的なタスクの同期機能として、タスクの起床待ちと起床のAPIを実装しました。この機能は主に1対1のタスクの同期に使われます。また、複数の情報を伝えたい場合には向きません。

IEEE 2050-2018規格では、複数のタスクの同期や複数の情報の伝達に使用できる同期機能としてイベント・フラグが定められています。本章ではTry Kernelにイベント・フラグを実装します。

イベント・フラグはイベントの有無を表現できる

イベント・フラグは、イベントの有無を示すフラグの集まりです。具体的には、フラグは0と1の2値の値を表す1ビットのデータで、あるイベント(事象)の有無を表現できます。イベントの意味はアプリケーション・プログラムで自由に決めることができます。

1つのイベント・フラグはそのマイコンで一般的な整数の大きさを持ちます。今回ターゲットにしているCortex-Mは32ビットCPUですので、イベント・フラグの大きさは32ビットとなります。つまり1つのイベント・フラグは、32個の1ビット・データの集合となります(図1)。

イベント・フラグの各ビットは、APIを用いてセット(1にする)またはクリア(0にする)できます。そしてタスクは、イベント・フラグの特定のビットがセットされるまで待ち状態となることができます。

● イベント・フラグによる同期

イベント・フラグを使用すると、あるイベントの発生を他のタスクに伝えたり、イベントが発生するまでタスクを待ち状態にしたりできます。

タスクは複数のビットを指定して待ち状態になることができ、指定したビット全てがセットされるまで待つAND待ちと、指定したビットのいずれかがセットされるまで待つOR待ちの選択ができます。

イベント・フラグはタスクから独立したカーネル・オブジェクトですので、複数のタスクが1つのイベント・フラグに対して、セット、クリア、そして待ちなどの操作を行うことができます。従って、イベント・フラグを利用すると複数のタスク間での複雑な同期も実現できます。

図2にイベント・フラグによる複数のタスクの同期の例を示します。タスクAとタスクBは発生したイベントに応じて、イベント・フラグの特定のビットをセットします。一方で、タスクCとタスクDは特定のビットがセットされるのを待っています。

● イベント・フラグとタスク付属同期との違い

タスク付属同期は起床という1つのイベントを通知するだけだったのに対し、イベント・フラグは複数のイベントの通知を行うことができます。

一方で、タスク付属同期でできて、イベント・フラグでできないことは、イベントのカウントです。タスク付属同期は起床要求をカウントできました。しか