

ハードウェアを意識して効率良くプログラミング Cによるマイコン操作術

森 久直

Hisanao Mori

第5回

割り込みを使ってLCDとLEDを動かしてみよう

割り込みとは？

例えば、信号機のある横断歩道を渡るとき、信号機が赤のときは待つと思います(信号無視をしなければ)。そして、待っている間、信号機を時折見ても信号機の色が変わったかどうかを確認し、青になったら横断歩道を渡ります。

このように、ある事象の状態を逐一チェックすることをプログラムの世界では**ポーリング**と言います。ポーリングを行い、状態の変化を確認したならば、現在行われている処理から別の処理に移行します。

一方で、信号機が青の間はメロディが鳴るものがあります。この信号機が設置されている場合は、わざわざ信号機を見る必要はありません。信号機が赤から青になったということ、メロディによって知ることができるからです。

このように、ある事象の状態を逐一チェックすることなく、外部からのシグナルに応じて、現在行われている処理から別の処理に移行することをプログラムの世界では**割り込み**と言います。また、スイッチの入力、設定時間への到達、通信データの受信などのように、割り込みのきっかけになる事象のことを**割り込み要因**と言います。

パソコンの操作に使われるキーボードやマウスも、割り込みが使われています。パソコンは、キーボードやマウスからの入力を絶えず見て処理をしているのではなく、入力があったときだけ処理をします。そうすると、パソコンのむだな作業がなくなり、作業効率が上がります。

● プログラムの動きで割り込み処理を確認

割り込み処理を使ったプログラムの動きを、図5-1で確認してみましょう。図5-1では、普段から実行されるmain関数と、割り込み要求があったときだけ実行されるint1関数、int2関数で構成するプログラムが走っているようすを表しています。

割り込み要求がない場合は、main関数の処理内容だけが実行されます。図5-1のように、 t_1 において割り込み要求が行われると、すぐにmain関数の処理からint1関数の処理に移行します。int1関数の処理が終わると、main関数の処理に戻ります。

ところで、もし、図5-1のように t_2 においてint1関数の処理中に割り込み要求が行われたらどうなるのでしょうか。ここでは、割り込み要因の優先度によって2通りの処理に分かれます。 t_1 における割り込み要因よりも、 t_2 における割り込み要因の方が優先度が高い場合は、図5-1のようにint1関数の処理からint2関数の処理に移行します。逆の場合は、int1関数の処理が終わってからint2関数の処理が開始します。

一見すると、main関数の中でint1関数とint2関数をコールしているように見えます。しかし、割り込み関数の最大の特徴は、main関数の中のどこからでも処理を開始できる点です。逆の見方をすると、いつどこで、割り込み関数が開始するか分からないということにもなります。ただし、MPUの割り込み関連レジスタを操作することによって、プログラムの中に割り込み禁止区間を設定することは可能です。

● 重要な処理に割り込みを使う

それでは、割り込みはいったいどのように使われて

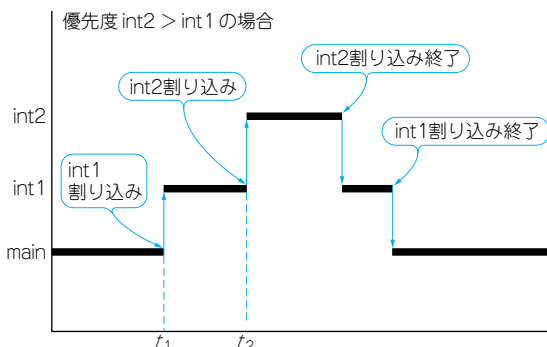


図5-1 割り込み要求があったときの処理の移行



部品セットをご購入いただけます。
詳細は本誌ホームページ(<http://toragi.cqpub.co.jp/>)をご覧ください。

いるのでしょうか。

実際には、MPUの動作の作業効率を上げることが目的ではなく、**どうしても行って欲しい処理を強制的に実行する目的で使われます。**

割り込み処理は、通常の処理に対して優先して実行するしくみになっており、言葉通り「割り込んで」くるのです。

また、割り込み要因ごとに優先度があるので、重要な処理に対して高い優先度を割り当てることで、さまざまな割り込み要求にダイナミックに対応できるシステムを実現することができます。

dsPICの割り込みのしくみ

dsPICの割り込みは、**ベクタ・テーブル**を用いた方式を使っています。従って、dsPICで割り込み処理をするためには、**割り込みベクタ・テーブルと割り込み関連レジスタ**の2点について把握する必要があります。

● 割り込みベクタ・テーブル

割り込みベクタ・テーブルは、割り込み処理関数の開始アドレス一覧を収めたメモリ領域です。

割り込み要求を受け付けたMPUは、このメモリ領域から読み出した割り込み処理関数の開始アドレスをプログラム・カウンタに書き込み、そのアドレスに分岐します。

図5-2に、dsPICの割り込みベクタ・テーブルを示します。割り込みベクタ・テーブルでは、INT0やT1などの割り込み要因ごとに、割り込み処理関数の開始アドレスを格納できる場所が決まっています。開始アドレスは自由に設定できます。

割り込みベクタは62個あり、そのうち8個はマスクできないトラップです。アドレス・エラー・トラップやスタック・エラー・トラップなどがあります。残る54個はマスクできる割り込みです。

dsPICには割り込みベクタ・テーブル(IVT)のほか、代替割り込みベクタ・テーブル(AIVT)が用意されています。これにより、一つの割り込み要因につき、

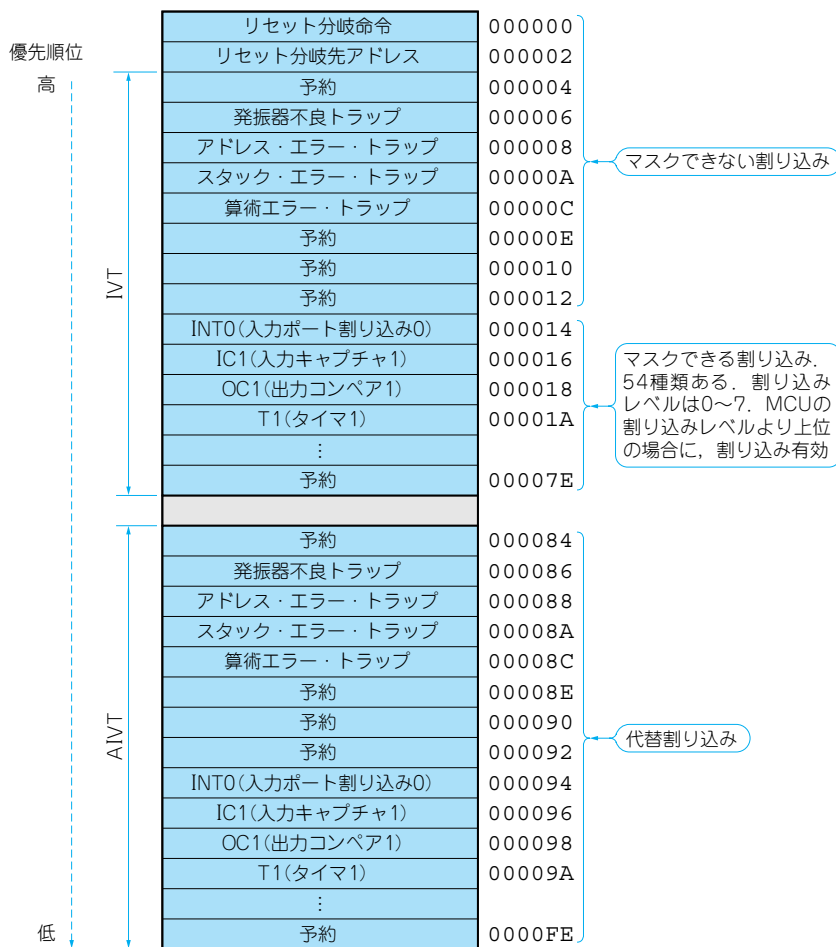


図5-2 割り込みベクタ・テーブル