

第6章 マイコンに効率良く仕事をさせるために

最重要機能「割り込み」をマスタする

三好 健文/北野 優
Takefumi Miyoshi/Masaru Kitano

第3章のプログラムでは、点滅と点滅の間の時間を、
`for(i=0;i<40000;i++);`
というように単純な計算を繰り返して作り出していました。その間、マイコンは一所懸命、計算を繰り返しているのですが、その計算結果を外部に出力するわけでもなく、一所懸命に待ち時間を作っているだけなのです。待ち時間を一所懸命に作るというのは、なんともむだなことですね。これは、図1(a)のように、次の処理を延々と待っているようなものです。図1(b)のように、この待ち時間に、気持ちを切り替えて別の処理をすれば、もっと効率良くたくさんの仕事をこなせるはずですよ。

また実際のアプリケーションでは、マイコンは何か一つの処理だけをすれば良いというものではありません。スイッチが押されるなどして外部から信号が入力されたら、別の処理に移るということも必要です。

このようなアプリケーションに有効な機能が、**ハードウェア割り込み機能**です。

ハードウェア割り込みでは、図2のように、特定の端子に変化があった場合に、その変化をトリガとして、現在実行中の処理を退避させた後で、指定したプログ

ラム(以下割り込みプログラム)に移ることができます。ハードウェア割り込みは、マイコン自身も持っている機能です。

割り込み処理の基礎知識

■ H8/3694Fの割り込み機能

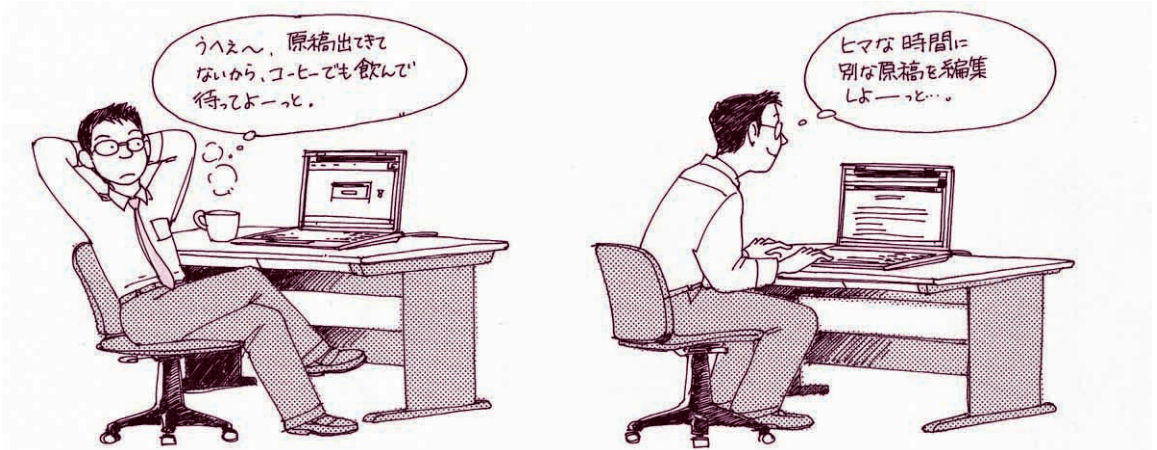
H8/3694Fには、IRQ3～IRQ0の四つの外部割り込み端子があり、それぞれ独立に使用することができます。これらの端子に入力されている信号のレベルの変化に応じて、割り込みプログラムが実行されます。

以下、割り込みの使い方について、CPUに対する手続きと指定方法を説明します。

● 割り込みのしくみ

ハードウェア割り込みが発生してからのマイコンの処理の流れを図3に示します。また、このときのCPU内部の動作を図4に示します。

割り込み要求があり、またその割り込みが有効であれば、**割り込みフラグ・レジスタ**の対応するビットに



(a) 割り込みを使わないと

(b) 割り込みを使えば

図1 割り込み機能の有無の違い

フラグを立てます。

CPUは、現在実行中の命令が終了するのを待って、現在処理中の内容(プログラム・カウンタやスタック・ポインタなど)を退避します。その後、その割り込みに対応する命令を実行します。

この割り込みに対応する命令コードは、マイコンによって決められた番地を書いてある値から生成されます。この割り込みごとに決められた番地を、**割り込みベクタ**と呼びます。

H8/3694Fの割り込みベクタは、メモリ・マップの上位60バイトで、表1のように並んでいます。また、割り込み要求の優先度も定まっており、同時に**割り込みが発生した場合には優先度の低いものは、割り込みコントローラによってその実行を待たされます。**

● 割り込みを使用したプログラム

割り込みを使用したスイッチで、LEDを点灯/消灯

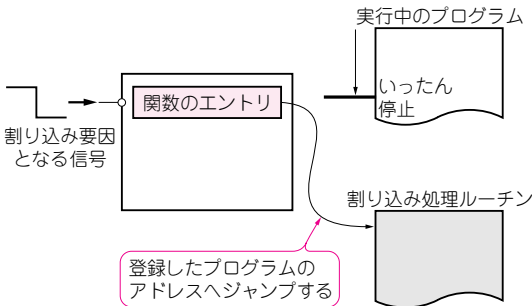


図2 ハードウェア割り込みの概念図

するプログラムをリスト1に示します。詳細は次項で解説します。

main関数の中で割り込みに関する設定を行い、割り込みによって呼ばれる関数 IRQ2 を記述しています。このプログラムは、モニタ上で実行します。

● 割り込みの使用方法

割り込みを使用するまでの手続きを図5に示します。

割り込みを使用するためには、まずほかの機能と共用になっている外部割り込み端子を割り込み用に指定する必要があります。

しかし、割り込み端子をセットしている間に割り込

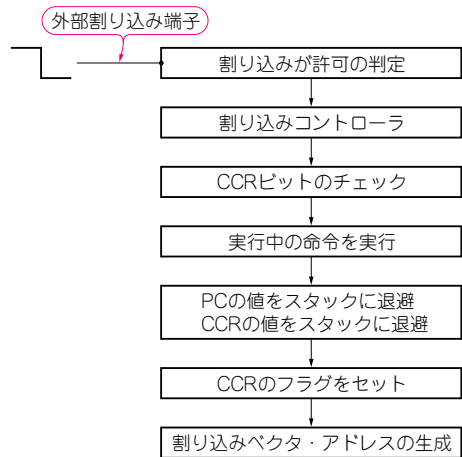


図3 割り込み要求から実行までのフロー

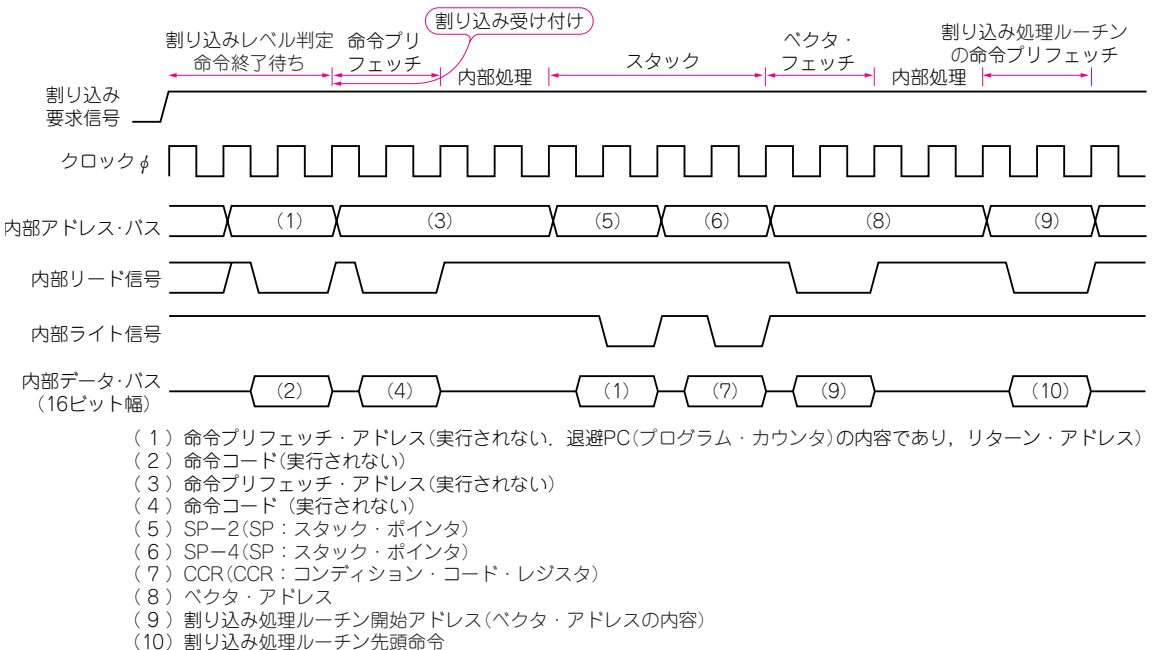


図4(1) 割り込み処理におけるCPUの内部動作