

割り込み処理のテクニックと タイマの基礎知識

第7章では、I/Oポートに接続されたスイッチの状態を読み込んで、I/Oポートに接続されたLEDに出力するプログラムを作りました。

このように、入力ポートのロジックの状態を判定して、何らかの処理を行ってから出力ポートに出力するという処理は、マイコン・システムの基本的な処理形態です。

I/Oポートだけでなく、タイマ、A-Dコンバータ、シリアル・インターフェースといったマイコン内部の各種機能もほとんどの場合は、関係するレジスタの値を読み取って処理を行うという同様の方法で利用することができます。

8-1 ポーリングとその欠点

8-1-1 ポーリングとは

実際のマイコン・システムではI/Oポートの入出力処理だけでなく、例えばタイマを使ったパルス幅測定、A-Dコンバータを使ったアナログ・デジタル変換、シリアル・インターフェースを使った外部とのシリアル通信など、いくつもの処理を同時に行う必要が生じてきます。

このような処理を行う場合にも、第5章のリスト5-1と同様に、必要とされる処理を並べて、無限ループ(while文)で巡回させることによって一応それなりの処理を行うことはできます。このように、無限ループの中でI/Oポートやレジスタの状態をチェック(リスト5-1ではif文)し処理を振り分ける方法を、一般に、ポーリング(poling)と呼びます。

8-1-2 ポーリングの欠点

ポーリングを使った処理方法は、非常に簡単でわかりやすいという利点がありますが、その一方である処理(リスト5-1ではLEDの点消灯)が行われている間は、ほかの処理が行われなくなってしまうという大きな欠点があります。

例えば、100msおきに発生する事象(イベント)に30msかけて処理を行うプログラムを動かしていたとします。このとき70msはCPUの空き時間になるので、まだまだ処理能力に余裕があります。

ところが、このイベントが発生し対応する処理が開始された直後に、より緊急度の高いイベントが発生したとしても、30ms待たないと処理を開始することができません。

この数値だけを見ているとあまり問題にならないように思えるかもしれませんが、シリアル・インター