

第7章 スタックのはたらきがキモ！一時的な処理中に危険アリ

ステップ解説！ハマリポイント②…
関数呼び出し&ローカル変数

永原 柊

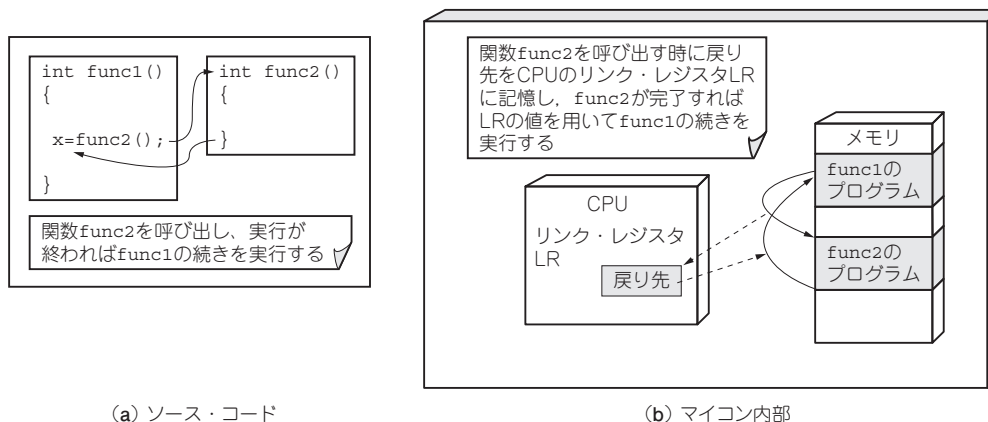


図1 マイコン・プログラミングのハマリポイント②関数呼び出し…基本はリンク・レジスタを使って呼び出し元に戻る
呼び出し元の記憶にリンク・レジスタを使う

関数呼び出しは、実は、ハードウェアを動かすCの書き方は直接関係しません。しかし、メモリの使い方の中で、関数呼び出しを行ったときのマイコン内部の動作について知っておくことは重要と考えます。

関数呼び出しを行ったときにマイコン内で行われている処理を理解すると、メモリの使い方を意識した効率の良いプログラムを作成できるようになります。また、スタックの使いすぎによるメモリ破壊といったデバッグの難しい現象が起こった場合でも、その原因の究明が容易になります。

関数呼び出しのメカニズム

● 呼び出された関数の実行が終われば呼び出し元に戻る

関数を呼び出して、その関数の実行が完了すれば、呼び出し元に戻ってプログラムの実行を継続します。一つの関数を、プログラム内の異なる場所から呼び出しても、関数の実行が終わればそれぞれ呼び出し元に戻ります。これを実現するには、関数の呼び出し元の情報を、関数を実行する間、覚えておく必要があります。

● 戻り先をリンク・レジスタが保持する

ARMマイコンLM4F120H5QR（以下、Stellarisマイコンと省略）では、このしくみをCPUのレジスタであるリンク・レジスタ(Link Register, レジスタR14, またはLRとも記述)で実現します。つまり、関数を呼び出すときに、図1のように戻り先をリンク・レジスタに記憶しておきます。関数の実行が完了したとき、リンク・レジスタが示すところからプログラムを実行すれば、呼び出し元に戻ったこととなります。

● リンク・レジスタはプログラム・カウンタの値を保持する

CPUはプログラム・カウンタが示すメモリにある命令を実行します。命令を実行中は、プログラム・カウンタは次に実行する命令を示します。

関数を呼び出す命令を実行するときには、プログラム・カウンタは関数呼び出しの次に実行する命令を示していることとなります。

関数を呼び出す命令を実行すると、このプログラム・カ