



知らないとヤバイ！テンプレート頼みで大丈夫？

Web

リセット直後に動く スタートアップ・ルーチン

笹野 貴義

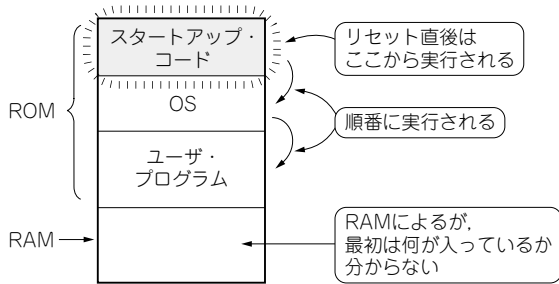


図1 リセット後最初に実行されるスタートアップ・コードで初期設定を行わないといけない

マイコン・メーカー純正開発環境を使っているとあまり意識しなくて済むが、トラブルがあったり、GNU環境を使ったりするときに知らないとヤバイ！

マイコンを使う上で避けて通れないのがスタートアップ・コードです。リセット直後からmain()関数を呼ぶまでに存在し、各種ハードウェアの設定を行うコードです。

マイコン・メーカー純正開発環境などを使う場合はスタートアップ・コードのテンプレートを出力してくれるので、あまり意識する必要はありませんが、GNU環境などを使う場合は自分で記述しなければなりません。このコードの記述方法は、知っておいて損はないでしょう。

このスタートアップ・コードの動作について、实例を挙

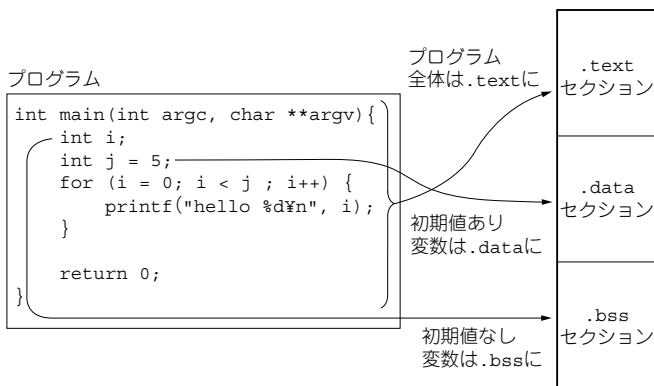


図2 プログラムや変数によって格納されるメモリ領域(セクション)が異なる

げながら解説を行います。

(編集部)

スタートアップ・コードで行うこと

プログラムの記述にアセンブラを使うことが必須に近かった過去のマイコンとは異なり、現在のマイコンはC言語を使用してプログラムの大部分を記述できます。C言語のプログラムは特殊なものでない限りmain()関数から実行を開始します。

しかし、マイコンのプログラムでは、電源を投入してからこの関数の実行を開始するまでの間に、いくつかの下準備が必要です。この下準備を行うプログラムが、図1に示すスタートアップ・コードと呼ばれるものです。

スタートアップ・コードが行う下準備は、プログラムの置かれる環境によってさまざまですが、主な処理は、次の四つです。

- スタック・ポインタの設定
- .bssセクションのゼロ・クリア
- .dataセクションをROMからRAMにコピー
- クロックや周辺機能ON/OFFの設定

● その1：スタック・ポインタの設定

基本的に、スタック・ポインタの設定は最初に行います。多くのマイコンではロード命令を使用してスタック・ポインタの値をロードします。たとえば本誌2012年6月号に付属したFM3マイコンで使用されているCortex-M3の場合は、ROMのベクタ領域(アドレス0x00000000)にスタック・ポインタの初期値を記述するので、これを使用します。

● その2：初期値なし変数を格納するメモリ領域をゼロ・クリアする

図2に示すように、ROM/RAMのメモリ空間を適切に使うために、「セクション」という概念があります。

.dataセクションと.bssセクションはともにRAM上に存在し、C言語のプログラムでは主に変数の値を格納して