

第3部 ARMマイコンの研究

第4章 電源を入れたのに動かない… そんなトラブルに備えて

リセット解除後の マイコンの動きを徹底研究

島田 義人 Yoshihito Shimada

マイコンは電源を投入したら、リセット信号を入力して内部のハードウェアをいったん初期状態に戻す必要があります。リセット信号が解除されると、プログラムにしたがってスタック・ポインタやプログラム・カウンタ動き出します。本章では、マイコンが動き出すまでの流れを研究してみます。 〈編集部〉



スタック・ポインタ、プログラム・カウンタの設定 → SRAMの初期化 → クロックの初期化 → main関数を呼び出す

図1 マイコンはmain関数に到達するまでにさまざまな準備処理を行っている
電源を入れても動き出さないときはマイコンが初期の処理を分析する必要がある

一般的にC言語でプログラムを作る場合は、「main関数からプログラムが始まる」というルールがあります。これはパソコンなどでのC言語プログラミングの常識です。しかしマイコンでは少し違います。

マイコンには「リセットした直後にmain関数を呼び出す」機能はありません。リセット直後に開始した「ブートローダ」と呼ばれるプログラムがmain関数を呼び出しています。そのプログラムは自分が作らなければなりません。実際には、main関数を呼び出すまでに、図1のようにさまざまな処理が必要になります。

とはいえ、LPCXpresso IDEなどの今どきのマイコン用コンパイラでは、CPUの起動からmain関数が呼ばれるまでのブートローダを自動生成してくれます。そのため、ユーザ(使用者)はmain関数から作ればよいだけの状態に準備をしてくれます。

通常であればこれで構わないのですが、電源投入時に発生するような問題に遭遇した場合には、原因を調査する上でCPUに電源を投入してからmain関数が呼ばれるまでの「自動生成される処理」も把握しておく

必要があります。

本章では、電源を投入してリセットが解除されてからmain関数が始まるまでのマイコンのふるまいを詳しく解説します。

CPU周辺の回路構成を頭に入れる

図2に示すのは、LPC1114マイコンのCPU周辺の機能ブロックの模式図です。

フラッシュ・メモリは32 Kバイト搭載されています。メモリは用途に応じていくつかの領域に区切って使用されます。フラッシュ・メモリにはベクタ・テーブル領域、プログラム領域、データ領域があります。ベクタ・テーブル(vector table)とは、マイコンがリセットした時や割り込みが発生したときに対応する処理のアドレスが記述されたデータの配列です。ROMにはブートローダと呼ばれるプログラムが入っていて、フラッシュ・メモリへのプログラムの書き込みなどを制御します。