

便利な環境や安価なマイコン・ボードがそろった今こそはじめる

C言語+リアルタイムOSで 初めての本格マイコン開発



第6回 マルチコアCPUでRTOSを動かす(前編)

豊山 祐一

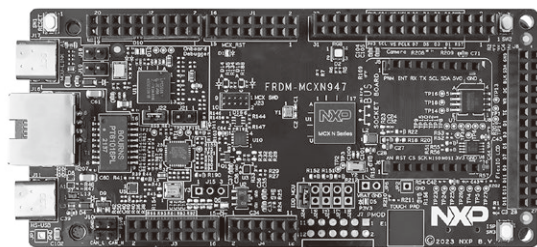


写真1 マルチコア・マイコンを搭載するマイコン・ボード
FRDM-MCXX947 (NXPセミコンダクターズ)

本連載では市販のマイコン・ボードを使用し、最新の開発環境を使ったC言語とリアルタイムOSによるプログラミングを解説していきます。リアルタイムOSには国際標準規格IEEE 2050-2018準拠の μ T-Kernel 3.0を使用します。

今回やること

● マルチコアのマイコンを使ってみる

PCやスマートフォンのプロセッサではマルチコアCPU(以降、マルチコア)は今や当たり前となりました。マルチコアの主な目的は、コンピュータの処理能力を向上させることです。マイコンの主流は今もシングル・コアですが、ハイエンドのマイコンにはマルチコアが採用され、手軽に使えるようになってきました。そこで今回はマルチコアのマイコンを使ってみます。

● マルチコアの2つの制御方式…今回はAMPで使う

マルチコアの制御方式はSMP(Symmetric Multi-Processing: 対称型マルチプロセッシング)とAMP(Asymmetric Multi-Processing: 非対称型マルチプロセッシング)の2つに大きく分けられます。PCではSMPが用いられますが、マイコンで使われるのは主にAMPです。

SMPはOSの機能によりアプリケーション・プログ

ラムを複数のCPUコアで動的に割り当て実行するのに対し、AMPはそれぞれのCPUコアで独立のプログラムを実行するので、OSのサポートがなくても実現できます。

今回はAMPの方式に基づき、各CPUコアで μ T-Kernel 3.0を実行させます。

● μ T-Kernel 3.0のプロジェクトを自分で作成する

今回のもう1つのテーマは、 μ T-Kernel 3.0のプロジェクトの作成方法です。

これまで本連載では、トロンプロジェクトから公開されている μ T-Kernel 3.0 BSP2(Board Support package v2, 以降BSP2)のサンプル・プロジェクトを使って、さまざまな市販のマイコン・ボードで μ T-Kernel 3.0を動かしてきました。

しかし、マイコン・ボードは非常に多くの種類があり、サンプル・プロジェクトが用意されているのはそのごく一部です。また、マルチコア用のサンプル・プロジェクトは提供されていません。

自分で μ T-Kernel 3.0のプロジェクトが作成できれば、より多くのマイコン・ボードや、マルチコアなどの形態に対応できます。マイコンの種類が同じであれば、BSP2が対応していないマイコン・ボードも使用できます。

そこで今回は、 μ T-Kernel 3.0のプロジェクトを自分で作成してマルチコアに対応します。

今回使用するマイコン・ボード …FRDM-MCXX947

今回使用するFRDM-MCXX947(NXPセミコンダクターズ)を写真1に示します。マイコンは、Arm Cortex-M33を2コア搭載したMCXX947VDFです。

BSP2は既にFRDM-MCXX947に対応していますが、シングル・コアのみの対応です。FRDM-MCXX947をシングル・コアで使うのであれば、トロンフォーラムのGitHubリポジトリ⁽¹⁾からサンプル・プロジェクトmtk3bsp2_mcxn947.zipをダウン