

Cortex-M/RX/SH/ARM7/ColdFire…
コード・サイズ&機能制約なしでプログラミングOK

どんなCPUでもやり方は同じに! GCCマルチ開発環境の構築

第1回 **現実解!**
Windows上でGNU開発環境を動かすための準備

村井 和夫

表1 GNU開発環境をちゃんと構築できればいろんなCPUを同じ手順で開発できる

本稿では試しに、本誌の歴代付属マイコン基板8種類に共通のGNU開発環境を構築してみる。Coldfireマイコン基板を除く7種類の基板は、拡張コネクタのピン配置に互換性があり、同じサンプル・プログラムで試せる

搭載CPU	付属号
FM3 (Cortex-M3 コア)	2012年6月号
RX	2011年5月号
SH-2A	2010年6月号
LPC (ARM7TDMI コア)	2009年5月号
ColdFire (M68K系コア)	2008年12月号
FR60	2008年5月号
V850	2007年5月号
SH-2	2006年7月号

● コード・サイズ&機能制限なし!

基本が理解できればGCCは簡単で便利

本誌には、何度かマイコン基板を付属したことがあります。各マイコン向けの統合開発環境 (IDE : Integrated Development Environment) を使うと、GUI (Graphical User Interface) により、誰でも比較的簡単に組み込みソフトウェアを作れるようになります。しかしメーカーやCPUファミリごとに専用の開発環境が提供されているため、毎回新しい開発環境をインストールして覚える必要がありました。いろいろなマイコンを使おうとすると、開発環境だらけになってしまいます。しかも初心者でも使えるように、マイコンの詳細を隠しているため、組み込みソフトウェア開発に必要な基礎的なしくみがわかりにくくなっています。

GCC^{注1}を使ったソフトウェア開発方法も、よく紹介されています。しかし、どうしても各CPUに固有の使い方の説明が中心になりがちです。本来C言語で記述すべきところまでアセンブリ言語で書かれている

注1 : 今はGNU Compiler Collectionの意味だが、本稿ではもともとGNU C Compilerの意味で使う。

ことがあるため、初心者には理解しにくく、GCCで開発するときの障壁になっています。このため、せっかくGCCを使ってもCPUごとに説明方法や使い方が変わってしまいます。このため、サポートを受けられないGCCを使った組み込みソフトウェア開発は難しく、実用的でないと思っている方も多いようです。

基礎的なしくみさえ理解すれば、GCCを使って、どのCPUでも同じ方法で簡単に開発ができるようになります。実際、商用の開発環境にもGNU開発環境を使っているものが多くあり、商用ソフトウェアの多くがGCCを使って開発されています。

本連載では、GCCとEclipseを使って、本誌に付属した8種類のマイコン基板をはじめとして、姉妹誌のトランジスタ技術に付属したCortex-M0コア搭載マイコンなど、すべてのCPUに共通のソフトウェア開発環境を構築します。その後、このソフトウェア開発環境と本誌付属のマイコン基板に共通な実験基板 (CQ-BBEL) を使って、CPUに依存しない共通の組み込みソフトウェアの開発方法を紹介します。

Windows用のGNU開発環境 MinGW

● Windowsパソコンをベースに使う理由

GCCを使った組み込みソフトウェアは、ほとんどすべてのOS上で開発することが可能です。しかしオブジェクト (実行形式のプログラム) をマイコンに書き込む (ダウンロードする) ツールは共通化されていません。しかもマイコン・メーカーから手順が公開されていないため、書き込み作業はWindows上で、各マイコン専用の書き込みツールを使わなくてはなりません。このため、書き込みツールの動作環境と同じWindows上に、GNU開発環境を構築することにします。

● 特徴1…最小限の構成で軽い

Windows上のGNU開発環境としては、Cygwinが有名ですが、Windows上のツールとは相性が良くありません。今回は、POSIX (Portable Operating System Interface ; UNIX/Linux互換の標準インター