

短期集中
連載

オープンソースGCC&GDBで試す! まずはLinuxなしから! ARM直系Cortex-A搭載! RZマイコン・プログラミング入門

第1回 RZ用コンパイル&デバッグ環境の構築

村井 和夫

本誌2013年11月号の特集「新定番の研究 ARM×Linux」では高性能Cortex-A搭載でワンチップ・マイコン・ライクに使えるRZマイコンRZ/A1L(ルネサス エレクトロニクス)を紹介しました。10,000円以下で入手できるRZボードCEV-RZ/A1L(コンピューテックス)をターゲットに、LinuxやμITRONなどのOS環境上で動作するプログラムの作成方法が説明されてきました。

Cortex-A コアは、ARM7などの従来のクラシックARMの後継に相当します。ARM7などを使ったことがある人なら似たような感覚でプログラミングできますし、逆にCortex-Aプログラミングができれば従来の定番ARM7などにも通じます。

本稿では、OSが全くない状態から、ハードウェアを制御するRZマイコン用ソフトウェアやOSを、GCC(GNU Cコンパイラ)を使って開発し、GDB(GNUデバッガ)を使ってデバッグする方法について説明します。

● ターゲットRZボードの入手方法

ターゲットとするRZマイコン・ボードCEV-RZ/A1Lは、基本開発環境や関連ドキュメント、μITRON4.0準拠リアルタイムOSお試しサンプルなどを同梱したキットIFKIT CEV-RZ/A1LとしてCQ出版社のウェブ・ショップ(<http://shop.cqpub.co.jp/hanbai/books/I/I000086.html>)から入手できます。

ボードだけであれば、コンピューテックスやマルツパーツ館のウェブ・ショップからも入手できますし、マルチパーツ館では拡張用のセンサ・ボードや無線LANモジュールなども販売しています。

● 専用の書き込みツールを使わないならLinuxパソコンが楽ちん

参考文献(1)(3)で紹介されたWindows上のGNU開発環境MinGWを使用しても、RZボードCEV-RZ/A1Lのソフトウェア開発はできます。メーカー提供のワンチップ・マイコンのプログラム書き込みソフト

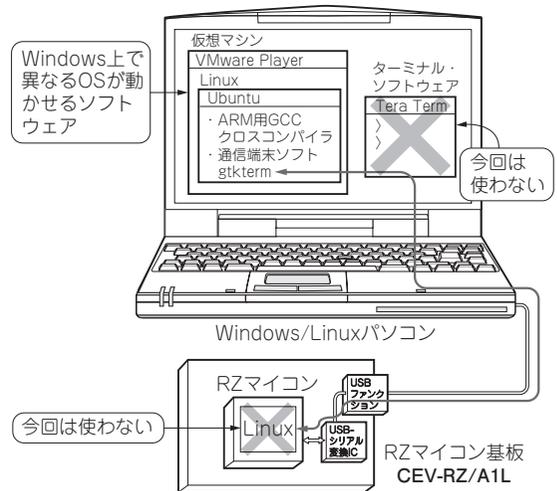


図1 今回構築する開発環境

ウェアが基本的にWindows上で動くので、あえてWindows上に開発環境を構築していました。

しかし筆者が試したときは、RZボードに搭載されているUSB-シリアル変換IC FT232(FTDI社)のWindowsドライバ(<http://www.ftdichip.com/Drivers/VCP.htm>)に問題があるようで、Windows上でシリアル・インターフェースを使ったGDBが実用にならないほど遅くなりました注1。

しかも、MinGWのGDBでは、「Ctrl+C」によるブレークでGDBそのものが終了してしまい、デバッグ対象ソフトウェアを止めることができません。

今回、開発パソコンでは、作成した実行ファイルをRZボードへダウンロードするのにシリアル通信ソフトウェアを使えばよいだけなので、制約の多いWindows環境をわざわざ選ぶ必要はありません。本稿では、RZボードの組み込みソフトウェア開発・デバッグを行う環境として、Ubuntu 12.04 LTS(長期サ

注1: 本誌2009年5月号付属のARM7TDMI基板はCP2102(シリコン・ラボラトリーズ)を使っていて、ドライバの問題は起こっていません。