

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4回 (最終回) GDBを使ったデバッグにトライ

村井 和夫

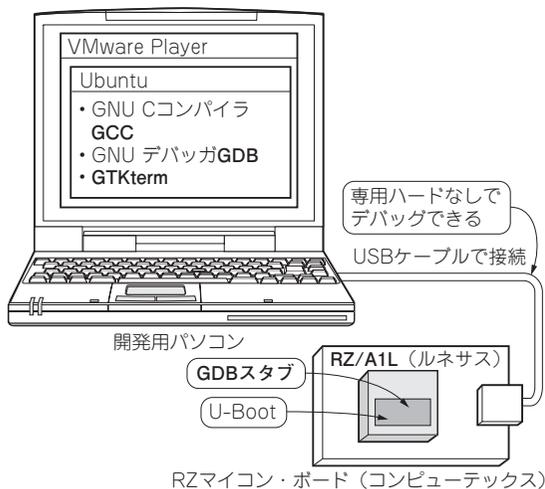


図1 GDBスタブをターゲット・マイコンで動くようにしておけば専用ハード不要のGDBでデバッグできる

本連載では、Cortex-A9搭載RZマイコン・ボードCEV-RZ/A1Lを使って、OSが全くない状態から、ハードウェアを制御する組み込みソフトウェアやOSをGCC & GDBで開発する方法について説明します。

今回は、GDBを使ってデバッグする方法について解説します。メモリ管理やOSの機能を使わずにブレーク・ポイントを設定してデバッグするために、例外処理機能を使っています。

デバッグ環境の構成…GDBスタブを用意する

開発環境の構成を図1に示します。ターゲット・ボード側に、パソコンのGDBと通信するプログラム、GDBスタブを置きます。GDBスタブを使うことで、ターゲット・ボード側にLinuxやJTAGなどのデバッグ環境を持たなくても、GDBを使ったデバッグができるようになります。

● RZボード上のGDBスタブを起動する

筆者が作成したGDBスタブをサンプル・プログラムとして用意しましたので、それを使います。

```

GtkTerm - /dev/ttyUSB0 115200-8-N-1
File Edit Log Configuration Control signals View

=> loads ← stub.srecをダウンロード
## Ready for S-Record download ...
...
## First Load Addr = 0x20200000
## Last Load Addr = 0x202010B7
## Total Size = 0x000010B8 = 4280 Bytes
## Start Addr = 0x20200000
=> go 20200000 ← GDBスタブを実行
## Starting application at 0x20200000 ...

Well come to ARM CortexA9 GDB Stub.
$S04#b7

```

図2 stub.srecをターゲット・ボードにダウンロードする

まず、Ubuntuの端末からGTKtermを起動します。U-Bootのloadsコマンドで、サンプル・プログラムrzGDBsample/gdbstub/にあるstub.srecをターゲット・ボードにダウンロードします(図2)。

```
=> loads
```

と打つと、S形式のオブジェクト・ダウンロード待ちとなるので、GTKtermの[File]-[Send raw file]でstub.srecファイルを選択してダウンロードします。続いて、

```
=> go 20200000
```

で、ダウンロードしたGDBスタブを実行します。このGDBスタブの上で、ターゲット・プログラムのデバッグを行います。

このGDBスタブを実行した状態で、GTKtermを終了します。次に起動するGDBでも通信ポート/dev/ttyUSB0を使いますので、GTKtermを終了しないと、gdbで接続できません。必ず終了させてください。

▶注意! 接続中にUSBを抜いてしまうと…

なお、もう一つ、よく発生する問題があります。GTKtermやGDBで接続中(プログラムを終了する前)にRZボードをUSBから抜いて、また挿してしまうと、プログラムでttyUSB0を使っているのに、次にUSBに挿したときデバイス名がttyUSB1となってしまう、以降戻りません。こうなるとリセットする