

JTAGデバッガによる 接続事例

邑中 雅樹

本誌 2006 年 6 月号の付録である SH-2 基板に対応した統合開 発環境 PizzaFactory3 を用いた TOPPERS/JSP カーネルのビル ド方法を,本誌 8 月号で紹介しました.今回は TOPPERS カー ネルから少し離れて,フリースタンディングのアプリケーション (OSを使用しないアプリケーション)のデバッグを PizzaFactory3 を用いて行う方法をいくつか示します.

CENESAS

0000

00000000

PizzaFactory3の無償配布版に含まれるデバッガ・サポート には、中小規模組み込み向けの修正が加えられていますが、ユー ザ・インターフェースはCDTのものを踏襲しています.組み込 み向けに Eclipse/CDTを拡張した製品は、PizzaFactory3 以外 にも、Zylin CDT などいくつか存在しています.PizzaFactory3 以外の環境では、コンパイラやデバッガのバイナリを別途用意 しなければならないなどいくつかの手間はかかるはずですが、環 境整備が終わってしまえば同じような手順でデバッグができる はずです.また、PizzaFactory3の無償配布版は、内部でGDB (GNU debugger)を呼び出しています.GDBやInsigntを使っ ている方にも参考になるかと思います。

今回は,本誌8月号付録CD-ROMに含まれるPizzaFactory3 ではなく,PizzaFactory31の α版を用いています.

# 1 Eclipse における C 言語クロス開発環境の現状

開発当時は Java 用の開発ツールとして見られがちだった Eclipse も,しだいに開発環境構築のための標準プラットホー ムとして認識されるようになり,あらゆる分野で利用されるよ



図1 パイナリが生成された状態の プロジェクト うになってきています.組み込み分野も例外ではありません. 多くのベンダから Eclipse をベースとした開発環境がリリース されているのは,読者の皆さんもご承知のことでしょう.

それらの流れの源流である Eclipse プロジェクトをのぞくと, 組み込みソフトウェア開発に関連するプロジェクトとして, CDT と DSDP があります.CDT は,一般的なC 言語開発の環 境を整備するものとしてスタートし,最近は,C 言語以外の make コマンドでビルドするコンパイラもカバーしようとして います.DSDP は複数のサブプロジェクトから成り立っていま す(コラム1参照).多くの組み込み向け開発環境は,この二つ のプロジェクトの成果を何らかの形で応用したものです.

CDT は, プロプライエタリ(商用)のツールをサポートする だけの柔軟な拡張性をもっています<sup>注1</sup>が,公式配布物には, GNU のツール群のためのコードが(想像に難くないが)数多く 入っています.デバッグ環境についても,GDB との併用を念頭 に置いたものとなっています<sup>注2</sup>.PizzaFactory3の無償版では, CDT の環境を可能な限り流用しているため,同じくGDB を内 部で呼び出しています.

## 2 PizzaFactory3 のデバッグ環境

ここで掲載しているスクリーン・ショットは, PizzaFactory3 におけるデバッグ・パースペクティブのものです.原稿執筆時 点でのPizzaFactory3のユーザ・インターフェースは, CDTの ものを踏襲しているので,ほかのCDTベースの環境でも流用 ができると思います.

作業に入る前に,ビルド済みのバイナリが必要です.今回の 目的はデバッグ環境の説明なので,バイナリのビルドの方法は, 別の機会に譲ります.

ここでは図1に示すように, hello プロジェクトに hello. out という SH-2用のバイナリができていることを前提にして

注1:公式配布の中には Microsoft 社の Visual C++ 用のサポートが部分的 に入っている.

注2:独自にほかのデバッガのサポートを追加することももちろん可能. そのようなサポートを追加した Eclipse を配布しているペンダも複数 存在する.

### Eclipse と実機ターゲット・ボードの JTAG デバッガによる接続事例

in-iio hello.out - [ I Debug i i hello.c	Ne <u>w</u> •	<pre>int argc, intf("Hel turn 0; onsole Prop;</pre>	Debug		Debug	
	Open With → Active Build Configuration → Copy Paste X Delete Moye Rename X Import X Expgrt		Create, manage, and functioning type filter text C C/C++ Attach to Local At C C/C++ Local Application C C/C++ Postmortem debug Freestanding C/C++ App	Config Defilter text □ C C/C++ Attach to Local Ap □ C C/C++ Local Application □ C C/C++ Dostmortem debug Freestanding C/C++ Appl	Create, manage, and run com  Comparison (Comparison)  Comparison (Comp	Igurations Interference Interfe
		C 1 Debug Lo		B A - to to to	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	

図2 起動構成の呼び出し

話を進めます.

# 3 シミュレータでのデバッグ方法

実機への接続には,実ボードやスタブなどいくつかのソフト ウェア/ハードウェアが介在します.複雑なシステムになれば なるほど,導入時の予期せぬトラブルに遭遇する可能性が高く なります.そこで,最初は,シミュレータにつないでみます.

いくつかのターゲット用の GDB にはインストラクション(命 令)セット・シミュレータが内蔵されています.インストラク ション・セット・シミュレータは,ほかの CPU の命令セット を解釈して実行するもので,今回の場合は SH-2 のバイナリを x86上で実行することになります.幸い,SH 用の GDB には, シミュレータのサポートがあります.GDB シミュレータとの接 続機能は,PizzaFactory3 の独自拡張なので,ほかの Eclipse ベース環境には用意されていないか,用意されていても手順が 違う可能性があります.

まず,デバッガの起動設定を行います.いくつか方法があり ますが,バイナリを選択した後で,右クリックをして[Debug As][Debug]を選ぶと楽でしょう(図2).

起動構成の画面が表示されたら, "Freestanding C/C++ Application "を選択し<sup>注3</sup>, 左上にある[New ]ボタンを押します (図3).

hello という新しい起動構成ができますが,同時にデバッガが サポートしていない CPU だというエラーが表示されます(図4). これは,適切なデバッガを選択していないために出るエラーで 心配は要りません.「Project」と「C/C++ Application」に表示されている内容に問題がないことを確認したら,「Debugger」タブを選択しましょう.

Debugger タブを見ると, Debugger として gdb/mi が選択さ れています.これは,セルフ環境での使用を想定したデバッガ で,今回の目的には合致しません.そこで,図5のように「GDB Simulator(sh-elf)」を選択します.

選択するとエラーの表示は消えるはずです.さて,実際にデ バッグを実行してみましょう.ダイアログ右下の[Debug]を押

# COLUMN

#### DSDP プロジェクト

DSDP プロジェクトは,組み込みソフトウェアのために特 化したプロジェクトです.CDTは,より一般的なC言語開 発環境の整備を行い,DSDPは組み込みソフトウェア固有の 付加価値の部分を整備するという位置づけになっています.

DSDP には,現在四つのサブプロジェクトが定義されてい ます.

- Device Debugging( CDT のデバッグ環境の強化 )
- Target Management(デバッグ・ターゲットの管理)
- Mobile Tools for Java(J2MEアプリケーション開発)
- Native Application Builder( C++ で記述された組み込み用 GUI の利用 )

DSDPは,日本からの貢献が多いプロジェクトです.たと えば,Native Application Builderには,国産のGUIツール・ キットである WideStudio/MWT が採用されています.

その活躍のわりに国内での盛り上がりは今一つという感が ありましたが,最近になって,Java WG,DSDPワーク ショップ(研究会)の立ち上げなど,日本語圏での活動環境整 備に本腰を入れ始めています.各ベンダによるツールの対応 計画にも影響があるはずなので,Eclipse そのものの開発に 携わっていないエンジニアも動向を注目しておくべきプロ ジェクトといえるでしょう.

注3: PizzaFactory3では「Freestanding C/C++ Application」というタイ プが追加されている.他社の Eclipse ペースの環境では,独自のタイ プが追加されていたり、「C/C++ Attach to Local Application」を流 用したりといったくふうがなされている.