Ultimate Solutions, Inc.

Tool Talk #03-001

Application Notes for Professional Developers of Embedded Systems

Installing, Configuring and Using GDB for the PowerPC

Introduction

The BDI2000 does not provide built-in source-level debugging capability. Although some target interaction can be accomplished from within the BDI shell, source-level debugging is not possible. This page outlines the steps necessary to use GDB as the front-end debugger.

This page covers installation under RedHat Linux. The tools can of course be installed under any version of Linux, however, the paths used under the various versions may differ. This document will help you to install the tools as built for the PowerPC without overwriting any other version of GDB that may already be installed on your machine.

When the RedHat Linux environment has been installed with the software development tools, the GNU toolset including GDB is available for use. However, the toolset as installed will only work for developing software native to the host. To build and debug software that will run on a PowerPC a cross-development version of the tools must be built. Since this document is focused on the debugger, the building of GCC or other tools in the GNU toolset for the PowerPC is not covered.

One final note: The native GNU toolset must be installed on your system in order to build GDB. If this has not been done please refer to the RedHat documentation for how to install these tools.

Downloading GDB

The first step to building GDB for PowerPC is to download the GDB software. The GDB Home Page is the best place to start in finding the software and suggests mirror site from which it can be downloaded. The following is a suggested sequence to follow in downloading and unpacking the software:

1. Make a directory to download the software to

   > mkdir gdbtools

2. From the GDB download page download the file gdb-<gdb version>.tar.gz to the new directory
3. Unpack the tools using the following command

   > tar zxvf gdb-<gdb version>.tar.gz

   The directory now contains the downloaded file as well as a directory titled gdb-<gdb version>

Building GDB for the PowerPC

Although you can build GDB within the gdb-<gdb version> directory tree, it is recommended that the PowerPC version be built in a separate directory. This approach allows you to build GDB versions for other target types that may be used in the future. Building GDB is a 2-step process: configuration and build. The following is a suggested sequence to configure and build the software. From the gdbtools directory perform the following steps:

1. Make a directory in which to build gdb and cd into the directory :

   > mkdir gdb-powerpc-linux
   > cd gdb-powerpc-linux

2. Run the configuration tool specifying PowerPC as the target-type:

   > ../gdb-<gdb version>/configure --target=powerpc-linux

3. Now build GDB. Note that this will take several minutes:

   > make

Installing GDB for the PowerPC

You can install the cross-GDB in a variety of places. The default location is /usr/local/bin. This location works fine since the location for the native GDB in most Linux environments is /usr/bin. You simply reorder your PATH statement in your login scripts, depending on which version you want to use. The cross-GDB can also be installed in your local account, or virtually any place you want to install it. You can specify the location when performing the configuration mentioned above simply by adding the --prefix=<install location> option to the configure line. To install GDB simply type the following command:
> make install

Using GDB with the BDI2000

Firmware Debugging

The following are some basic steps to get you started using GDB to debug firmware that has been programmed into flash. Use the help command in GDB to find a list of all available commands:

1. Open two console windows
2. In console window 1
   - Open a telnet session to the BDI
   - Reset the target board

     BDI> reset

   - Set the PC to the reset vector

     BDI> rm pc <reset vector>

3. In console window 2
   - Change to the directory where the firmware image was built
   - Run GDB specifying the name of the ELF file associated with the firmware image
   - At the GDB prompt connect to the target

     (gdb) target remote <BDI IP address>:2001

   - Set appropriate breakpoints.
   - Start running the firmware

     (gdb) continue

   Note: the continue command, rather than the run command, must be use to run the program.

Application Debugging

TBD

GUI Front-ends for GDB

GUI front-ends make using GDB easier because they provide immediate visual feedback during the debugging process. Several front-ends are available. The following is a list of some of the more popular ones. Follow the associated links to find out more about each:

DDD
Insight

Authored by: Dan Jozwiak - Independent Contractor
MapleLeaf Software, Inc.
1 Heritage Circle
Hudson, NH 03051
(603) 566-0919

Ultimate Solutions, Inc.
10 Clever Lane
Tewksbury, MA 01876
Toll Free: 866.455.3383 Phone: 978.455.3383
Fax: 978.926.3091 Email: info@ultsol.com
Web: http://www.ultsol.com