

Running The Demo Program Using HAL/GNU Debugger

Hitachi America provides the HAL/GNU H8/300 Software Tools to program the Hitachi H8/300 series microcontrollers. These software tools consist of C compiler, assembler, linker, librarian, and debugger. The HAL/GNU debugger, **GDB**, can communicate with the H8/300 Series Evaluation Boards to debug and test our program. Hitachi includes some demo programs in the HAL/GNU H8/300 Software Package.

This paper is served as a tutorial on running the demo program, i.e., **Flash.c**, on the H8/300 Series Evaluation Boards. The demo program can be found in the HAL/GNU H8300 directory under the DEMO sub-directory.

Hardware Setup

Connect the H8/300 series Evaluation Board to your PC. Use a cable as documented in the H8/300 Series Evaluation Board user's manual. Attach one end of the cable to the port labeled TERM on the Eval Board and the other end to PC port, i.e., COM1 or COM2. Connect and power up the board. Press reset button on the Eval Board to see if the lights on the eval board are on. It takes about ten seconds to light up all lights on the Eval Board.

The GDB depends on an auxiliary terminate-and-stay-resident program **asynctsr** to communicate with the H8/300 Eval Board through a serial port. We must also use the DOS mode command to set up the serial port on the DOS side. The default communication parameters for GDB and the evaluation board are:

- 9600 baud
- no parity
- 8 data bits
- 1 stop bit
- continue retrying

The following are DOS mode commands to initiate between the H8/300 series Eval Board and COM2:

```
C:\h8300\bin> mode com2:9600,n,8,1,p <enter>
com2:9600,n,8,1,p
C:\h8300\bin> asynctsr 2 <enter>
asynctsr installed
```

Running the Demo Program

Copy the Flash.c demo program in the DEMO directory to the BIN directory. We need to compile the Flash.c program to produce the absolute file. The following command line will compile and produce the absolute file called **Flash.x** with debugging information and optimization:

```
C:\h8300\bin> GCC -o Flash.x -g -O Flash.c
```

where

GCC name of the compiler
-o compiler switch to name the output file as Flash.x

- g compiler switch to generate debugging information
- O compiler switch to optimize the code

We can invoke the GDB Debugger by calling the GDB name. The GDB will prompt us with the prompt (**GDB**). The following commands start and run the GDB:

```
C:\h8300\bin> GDB

(gdb) TARGET HMS COM2

Connected to remote H8/300 HMS system

(gdb) LOAD Flash.x

.text: 0x8000 .. 0x80e0 *
.data: 0x80e0 .. 0x80e4 *
.stack: 0xf000 .. 0xf014 *

(gdb) FILE Flash.x

Reading symbols from flash.x ... done

(gdb) RUN

Starting program: /h8300/bin/flash.x
```

At this point, we can see the lights on the Eval Board are flashing one after another. To stop the program, press the **RESET** button on the Eval Board. The RESET button will:

- interrupt our program
- return to the GDB command prompt after our program finishes normally. The communication protocol provides no other way for GDB to detect program completion.

We can use the **HELP** command at any time in the GDB to find out more about GDB commands. To end the GDB, type the **QUIT** command. We use some of the GDB commands in this tutorial, which are:

- target hms specify cross-debugging to the Hitachi Eval Board
- load download the program to the board
- file reading the symbols from the program
- run execute the program

The information in this document has been carefully checked; however, the contents of this document may be changed and modified without notice. Hitachi America, Ltd. shall assume no responsibility for inaccuracies, or any problem involving a patent infringement caused when applying the descriptions in this document. This material is protected by copyright laws. © Copyright 1995, Hitachi America, Ltd. All rights reserved. Printed in U.S.A.