1 Introduction

This laboratory is an introduction to the software development process that will be used in this course. A description of this process can be found at: y:\etec274\LectureNotes\Chapter03\SimpleDebug.pdf. You will:

- Edit and assemble a prewritten program using the CodeWarrior development system.
- Download and run the program on the 9S12UB.
- Use and demonstrate the D-Bug12 debugging and memory editing commands.

After finishing this lab you will have completed a typical development cycle and therefore be prepared to write your own programs for the remaining labs.

2 Creating a Project Directory and CodeWarrior Project

The first step in every lab will be to create a project directory and a CodeWarrior project. For this class there is a custom project stationary, ***AbsAsmWWU***, designed to simplify the project so it uses single assembly files with absolute locations defined with the ***org*** directive.

1) Create a new directory called ***lab1*** in your ***et274*** directory.
2) Open the CodeWarrior IDE from the start menu.
3) Select ***File -> New…*** and select ***HCS12 Stationary***, enter the name of the project, and make sure the path points to your new project directory.
4) After selecting ***Ok***, select ***HCS12 Stationary*** and choose ***AbsAsmWWU*** from the list.

3 Accessing Prewritten Source Code

The stationary creates a project that includes the source file ***demo1.a12***, which is in the ***Sources*** folder. This is a simple demonstration program that can be used as a template for future labs. To open the file, double-click on the name in the folder view.

4 Editing the Source Code

In this step, you will rename and edit the demo program.

Procedure. Complete the following items:

1) Open ***demo1.a12***.
2) Use ***Save As…*** to rename it to ***Lab1.a12***.
3) Now, before editing the file, you should configure the Editor so inserts spaces for tabs. To do this, ***SelectEdit→Preferences→Editor→Font & Tabs***. In the dialog box, select ***Tab Inserts Spaces*** and set the ***Tab Size*** to 4. Select ***Apply***. Before leaving this dialog make sure that the font is set to Courier (or Monospace.com) and the size is 9. Select ***Ok***.
4) Find the place in the code that determines the on-time and off-time for the LED. Currently they are set to 250ms on-time and 750ms off-time. Change these value(s) so the program will light the LED every ½ second with a 50% duty cycle.
5) Change the constant that determines the initial value of the pulse counter and change it so the LED only pulses 20 times.
6) Save the file.

5 Assembling the Source File

In this step, you will assemble your program and make a printout of the listing.

Procedure. Complete the following items:

1) To assemble your new source program, click on the ***Make*** button.
2) There should be no errors. If there are, open the source code, and find and correct the errors.
3) To open the listing file (***Lab1.lst***). Look in the ***bin*** folder. If it is not showing, right-click on the ***bin*** folder and select ***Add Files…*** and select the listing. Now that it shows up, click on it to open it up in the edit window.
4) To print the listing file, select ***File->Print…*** You should deselect the ***Wrap Text Lines*** check box when printing listing files.

6 Downloading and Running the Program

Now you will need to download the S-Record file, ***LabBuild.s19***, to the 9S12UB board.

Procedure for Step 5. Complete the following items:

1) Connect the 9S12UB Target serial connector to a PC COMM port.
2) Open the appropriate terminal emulator window and hit the target reset button on the board.
3) You should see the D-Bug12 information and prompt.
4) Download the S-Record file to the 9S12UB by using the ***LOAD*** command.
5) Run the program using the ***G*** command. Verify that the program runs as expected – the LED blinks 20 times at 2Hz.

7 Debugging the Program

Once the program is correctly downloaded, it can be executed and the D-Bug12 debugging commands can be used. Refer to ***D-Bug12 Reference Guide*** for detailed information on the D-Bug12 commands and the ***Basic Software Construction*** document.

Procedure for Step 5. Complete the following items:

1) Follow the debugging procedure described in the ***Basic Software Construction*** document.
2) Practicing using the D-Bug12 commands. In the next few weeks you must be proficient in using the following D-Bug12 commands: ***LOAD***, ***G***, ***BR***, ***NOBR***, ***MD***, ***MM***, ***RD***, ***RM***, ***SO,*** and ***T***.

8 Demonstrate use of Debugging Commands to Instructor

During the lab period, you will be asked to demonstrate the commands used in the debugging process described in the ***Basic Software Construction*** document.

9 Submitting Edited Source Code

Once your edited source code works correctly, you will submit it to the instructor for grading. To submit labs you will email them to: ***Todd.Morton@wwu.edu***.

Procedure. Complete the following items:

1) Send your edited source with ***lab1*** in the subject line. The source code (.a12 file) must be sent as an attachment.

10 Write-Up

A write-up is not required for this lab. The mail sent will be checked off for grading. The program must be loaded onto your board and checked by the instructor before the end of lab on the due date. The source code must be mailed by midnight on the due date.

Due date: April 18, 2008