ECGR4101/5101/6090, Fall 2004: Lab 2

Testing the MSV30262-SKP Board & Building a Simple MSV30262-SKP Program

Learning Objectives

This lab will help you demonstrate how to build a project and load the executable onto the MDECE30262 board.

General Information

The general steps for this lab are:

1. Test your board with the pre-loaded test program. If is not pre-loaded, download the program.
2. Compile the SKP16C26 test program to your board, download and run.
3. Using TM generate a new project. Modify the demo program with the requirements.
4. Build the project and load onto your board. Run the program and observe the operation.
5. Demonstrate for a TA and turn in a lab report.

Prelab Activity

You may use the PCs in Smith 347 or your own PC to do this lab experiment. However, you must load one of the machines in Smith 347 with these software tools. If you want to work on lab assignments on your own PC, then load the tools on your PC to perform this exercise. To load the tools, insert the CD in your machine and follow the instructions. You do not need to load the IAR tools on the machines.

After you load the PC, attach the board to the PC and follow the USB driver installation instructions.

Test the board by attaching your board to the computer with the USB. Power-up the daughter board. Ensure the LCD displays “Renesas Technology”. Turn the right-hand potentiometer and verify the display changes and the LED speed changes.

Prelab Questions

1. Where should your “working directory” be located when using lab computers?
2. How many lines do the development tools allow in a C source file?
3. Where is the first place to look for help with labs in this class?

Laboratory Assignments

1. Follow the instructions on the QuickStart Guide, numbers 4 and 5. Note carefully the steps required to build a program, download, and debug a program. Note the change of behavior of the board.
2. You will next build a new project based on the “demo” (on QuickStart Guide step 4e, choose “demo” instead of “test”). Download and run this new code. Note the change of behavior of the board.

3. In the project editor, choose main_skpdemo.c as the file to edit. You will need to change the program in three areas for these new requirements:
    a. Instead of flashing the lights G -> Y -> R -> Y -> G, change it to having two LED on some of the time with the sequence
       G -> GY -> Y -> YR -> R -> YR -> Y -> GY -> G.
       This will require you to change the state numbers in three areas of the code.
    b. Keep all other code the same.
    c. In the file header comments (bottom of comments), add a few lines stating your name, date changed, and values changed.
    d. In the procedure that changed, add a line or two of comments stating the changes.
4. Save the file.
5. Now use the rebuild button on the toolbar to rebuild the project. This will pull up the builder and show the status as the project is cleaned and rebuilt using the updated files. The memory should be adjusted for the added lines of code.
6. Reload the software onto the boards and test. Verify that the new functionality works as specified.
7. Complete your lab report.
8. Bring the new board to the lab TA and demonstrate the new code (without the HEW application running). When the TA checks your board, he will also take your lab report. You **will not** need to include a printout or soft copy all of the code – just "snippets".

Lab Report

Include in the checkout part of your lab report the lines:

1. LEDs flashing as specified ________
2. Other functionality works as in original code. ________

Include in your lab report observations and procedure like the following:

The general learning objectives of this lab were . . .

The general steps needed to complete this lab were . . .

Some detailed steps to complete this lab were . . . .

1. *Step one*
2. *Step two*
3. *. . . .*

Code generated or modified to complete this lab...

No need to include all the files for the lab. Just include the modified code.

Some important observations while completing/testing this lab were . . .

Here include the memory report given at the end of the compile process.

In this lab we learned . . . .