ECGR 4101/5101, Fall 2006: Lab 7

Round Robin Scheduling (Version 1.0)

Learning Objectives

This lab will introduce you to using Round Robin Scheduling, UARTs and interrupts on our Renesas board, and new C programming concepts.

General Information

The general steps for this lab are:

1. Generate a new project. Name your new project Lab7.
2. Modify the main.c file and include the appropriate files. Include commenting along the way.
3. Program the lab. Don't forget the necessary include files to get the correct functionality.
4. Compile the code into an .x30 file, and load onto the board.
5. Test the program and repeat sets 4 and 5 until the program works as required.
6. Write your lab report.
7. Demonstrate for a TA and turn in your report.

Laboratory Assignments

In this lab you will be generating two main.c files from scratch. The program will transmit information between boards, read data from one board and display data on a second board's LCD using UARTs.

Steps

1. Follow the steps given in earlier labs for generating a new project.
2. Create the main.c file and include the appropriate files.
3. Build your program slowly, testing along the way. Perform compiles and solve each requirement one at a time.
4. Continue to build and test the program until all of the requirements have been met.
5. If you run into problems, use the break point functionality of KD30 to step through the code until you find the problem.
6. Once all the requirements have been met, ensure that everything works.
7. Finish lab write-up and demonstrate for a TA.

Requirements

Req. 1 – The code generated is written in C for the SKP16C62P.
Req. 2 – The code is well commented and easy to follow
Req. 3 – Board 1 must run with Round Robin Scheduling – you should determine the best priority for the tasks.
Req. 4 – Board 1: Toggle the Green LED every 0.1 seconds.

Req. 5 – Board 1: Toggle the Yellow LED every 0.25 seconds.
Req. 6 – Board 1: Toggle the Red LED every 0.5 seconds.
Req. 7 – Board 1: Configure the board to continually update the thermistor ADC value.
Req. 8 – Board 1: Every 1.0 seconds, read the temperature value, convert it to ASCII, and send the ASCII values (with a newline character at the end) via an RS232-c cable to a PC running HyperTerm.
Req. 9 – Board 1: Every 2.5 seconds, read the temperature value, convert it to ASCII, and send the ASCII values to Board 2.
Req. 10 – Board 1: With the value read from the thermistor, convert it into a temperature as an ASCII string with the format "xxx.x", which is the temperature in Fahrenheit.
Req. 11 – Board 1 & 2: Serial communications must be handled with interrupts and queues.
Req. 12 – Board 2: When ASCII data is received, display it on the LCD.
Req. 13 – Communicate at 9600 baud, odd parity, one stop bit
Req. 14 – Connect two data lines and one ground line between the two boards.
Req. 15 – Connect two data lines and one ground line between Board 1 and the PC.

Lab Report

Turn in a hard copy of the code you wrote and a printout of the map file. Also 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 for this lab…

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

In this lab we learned . . . .