

ECGR4101/5101, Spring 2010: Lab 6

Round Robin Scheduler

Learning Objectives

You are to write a program that runs on your QSK62P board that uses the Round Robin Scheduler, with interrupts. You will need to use timers and transmit serial data.

Laboratory Assignments

In this lab you will be programming the timers and using interrupts. The LEDs, LCD, the serial port will be used for running the task and the RRS would be used to schedule the tasks to run. This lab must be demonstrated to the TA.

1. The program should rely on Round Robin scheduling and use interrupts for the scheduler and UART only.

Requirements

- Req. 1. The code generated is written in C for the QSK62P or QSK62P Plus.
- Req. 2. The code is well commented and easy to follow
- Req. 3. **You may not use the BNS functions.**
- Req. 4. This board must use Round Robin Scheduling (RRS) for the LEDs and starting the serial communications.
- Req. 5. Check the TASKS tab for information about the tasks that should be run using the RRS
- Req. 6. The RRS timer tick is 1 milliseconds.
- Req. 7. The RRS Priority, high to low, is LED Signaling (Task1), Display on the LCD (Task2), Serial Transmission (Task3).

Tasks

Task 1 – LED Signaling – should run every 2 seconds

- a. Toggle the three LEDs in the following fashion (zero to seven – 3 bit binary)

LED1	LED2	LED3	
OFF	OFF	OFF	
OFF	OFF	ON	
OFF	ON	OFF	
OFF	ON	ON	
...	
...	
ON	ON	ON	
OFF	OFF	OFF	(RESTART)

Task 2 – LED Status Display – should run every 3 seconds

- a. Display the status of the LEDs when task 1 was run last on the board’s LCD display.
- b. BNS functions should not be used
- c. The string should be of the format
 "ON [space] ON [space] ON" or
 "OF [space] OF [space] OF" (or any combination, whatever represents the true state).

Task 3 – Serial Transmission – should run every 5 seconds

- a. Display the status of the LEDs when task 1 was run last on the PC.
- b. Settings – 9600 baud, even parity, one stop bit
- c. Schedule a character string transmission to the PC every 1 second via UART0.
- d. Each string sent to the PC must use interrupts. Each string must end with a CR and LF.
- e. The UART communications is via interrupts, but the string is built and the first character sent due to RRS.
- f. BNS functions should not be used
- g. The string should be of the format
"LED0=ON LED1=ON LED2=ON" or
"LED0=OFF LED1=OFF LED2=OFF" (or any combination, whatever represents the true state).

Test Procedure

See the lab checkout sheet to understand the rubrics of the demonstration.

Lab Report

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. *. . . .*

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

In this lab we learned

Create a single pdf containing:

1. Your lab report
2. Your code (no need to include the sect30.inc, ncr0.a30, LCD, or any .h files). Include all c files that have code that you wrote (but if you are smart, this should be one small file so that the code size is small). Ensure you use an 8 or 9 courier font so that most lines of code take one line of text.
3. The full map file.). Ensure you use an 8 or 9 courier font so that most lines take one line of text.

Upload this pdf to moodle. Name the file xxxxxxxx_yyyyyyy_lab6.pdf, where xxxxxxxx is the last name of one lab partner, and yyyyyyy is the last name of the other lab partner.

Next, upload the code file as text to Moodle. . Name the file xxxxxxxx_yyyyyyy_lab6.c, where xxxxxxxx is the last name of one lab partner, and yyyyyyy is the last name of the other lab partner.

FAILURE TO FOLLOW THESE SIMPLE INSTRUCTIONS COULD RESULT IN THE LOSS OF POINTS.