Department of Engineering

EE 4710 Lab 3

Title: Cyclic Scheduling and Preemption

Objective: The student should become acquainted with the concept of task preemption and how it is implemented. The student will also construct a preemptive cyclic scheduler.

Parts: 1-C8051FX20-TB Evaluation Board
1-USB Debug Adapter
1-DB-9 Serial cable (USB adapter cable is also ok)

Software: Silicon Laboratories IDE version 3.50.00 or greater. Keil compiler.

Preparation: Write the title and a short description of this lab in your lab book. Make sure the page is numbered and make an entry in the table of contents for this lab.

Download and study the files lab3_main.c, context.h and context.asm from the course website. Pay particular attention to the timer2 interrupt service routine and note how it (1) saves the context of the current task, (2) calls the scheduler (which you will write in C) and (3) restores the context of the task that the scheduler has chosen. Also, carefully analyze what the function yield() does.

Record in your lab book a brief (1/2 to 1 page) explanation of how tasks are created, scheduled, preempted and restored. Write in sufficient detail that you could use it three years from now to write similar code for a different processor.

You are to design and implement a cyclic scheduler that performs 3 periodic tasks and one aperiodic task. Unfortunately, the word “task” in this code anticipates future labs when each task will have its own thread of execution. Do not confuse these tasks with the periodic and aperiodic tasks mentioned earlier. For this exercise, there will only be two such tasks, a cyclic executive, which executes the periodic jobs and a background task, which executes aperiodic jobs.

The three periodic tasks all have periods of 30ms and phases of 0, 10 and 20ms respectively. The first task sends the alphabet A-Z (one letter per job) to the serial port then repeats. The second task sends the numbers 0-9 (one letter per job) to the serial port then repeats. The third task sends a space to the serial port once for every job.

The aperiodic task checks to see if a character has arrived on the serial port and if it has, turns the LED on if it is a ‘1’ or off if it is a ‘0’. Other characters are ignored.

lab3_main.c already initializes the microcontroller, sets up the serial port, creates the cyclic executive task and begins multitasking. In order to meet the requirements of this exercise, you will need to write the scheduler, the cyclic executive and add code to the background thread to handle the aperiodic task. After writing this code and compiling it without error, bring it to your scheduled lab session.

Hint: the scheduler can simply alternate between tasks, and the cyclic executive loop can be implemented something like:

```
for ( ; ; )
{
  jobA();
  yield();
  jobB();
  yield();
  jobC();
  yield
}
```

Procedure: Use a DB-9 serial cable to connect your 8051 board to a computer running a terminal emulator such as puTTY. Configure the terminal emulator for 8 data bits, 1 stop bit, no parity. Compile, link, download and run your code. Verify that letter/number pairs appear on the screen, that whenever a ‘0’ is pressed the LED goes off and that whenever ‘1’ is pressed the LED goes on. Demonstrate this to your lab instructor.

Affix all your source code to your lab book then, if you have not already done so, answer the following questions (in full sentences):

1. Does each task have its own stack or do they all share a common stack?
2. How does yield() cause the processor to suspend the current task and start another?
3. What stack does the scheduler function use?
4. Why does create_task() set PSW to 0?
5. How could you pass parameters to a newly created task?

.

Write a summary or conclusion. Remember to sign or initial then date each page.