NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR
Semester 2 AY2010/2011

CG2271/CS2271 – Real Time Operating Systems/Embedded Systems

Apr/May 2011 **Time Allowed: 2 Hours**

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **FOUR** questions and comprises **FIFTEEN (15)** printed pages, including this page. The weightage for each question is as indicated, and total 100 marks. Note that the marks distribution is not equal across questions.

2. Answer **ALL** questions within the boxes provided in this booklet. Whatever is written outside of the box boundaries will be ignored.

3. This is an Open Book examination.

4. Please write your Matriculation Number below.

MATRICULATION NO: _____

This portion is for examiner's use only

Question	Marks	Remarks
Q1	/20	
Q2	/10	
Q3	/35	
Q4	/ 35	
Total	/100	

Question 1 Process Synchronization (20 MARKS)

a. The producer-consumer problem can cause an application that produces data and another that consumes it to stop working completely. Explain why. (5 marks)

b. This code comes from the lecture notes and was used to demonstrate the producer-consumer problem. The notes give a solution using counting semaphores, and another using a monitor. Fix the producer-consumer problem using a THIRD method that does not use counting sempaphores or monitors. (10 marks)

```
#define N 100
int count=0;

void producer()
{
     int item;
     while(1)
     {
          item=produce_item();
          if(count==N)
               sleep();

          insert_item(item);
          count++;
          if(count==1)
               wakeup(consumer);
     }
}

void consumer()
{
     int item;
     while(1)
     {
          if(count==0)
               sleep();

          item=remove_item();
          count--;

          if(count==N-1)
               wakeup(producer);

          consume_item(item);
     }
}
```

c. Explain how the changes you have made solve the Producer-Consumer problem. (5 marks)

Question 2 Tasks and Threads (10 MARKS)

a. List down the similarities between threads, and uC/OS-II tasks. (5 marks)

b. Despite the similarities, uC/OS-II tasks are not threads. Explain why. (5 marks)

Question 3 Scheduling Algorithms (35 MARKS)

a. Can Critical Instance Analysis be used to test schedulability of:

i. Fixed Scheduling.
ii. Earliest Deadline First Scheduling.

In all cases explain why or why not, and the assumptions (if any) you have made. (10 marks)

b. Two assumptions made in Critical Instance Analysis are that i) switching overheads are negligable and ii) There are no inter-task dependencies. Explain the importance of each of these assumptions and what would happen if each assumption did not hold. (10 marks)

c. Suppose now that switching between tasks entails a certain overhead. Detail how you would perform Critical Instance Analysis given:

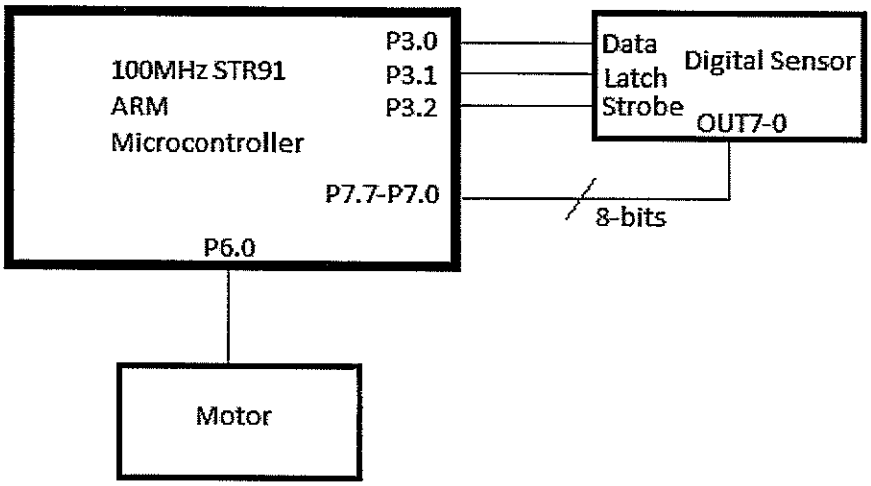
   i. A fixed overhead of n cycles.

   ii. A random overhead of between 1 and n cycles.

   Explain your answer, including assumptions that you have made and limitations of your approach. (15 marks)

Question 4 Microcontroller Programming (35 MARKS)

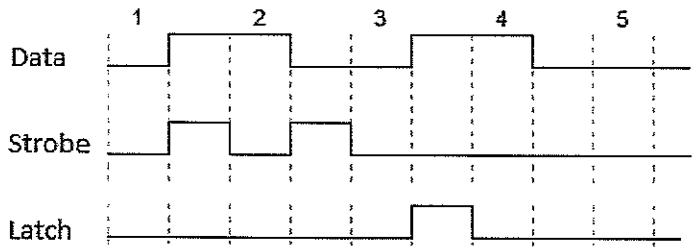
In this question we will consider an embedded system that reads in data from a sensor and sends out PWM commands to a motor. The architecture is shown below:



The system is based on a 100MHz ARM STR91 microcontroller. Three pins of Port 3 are connected to an intelligent digital sensor that sends data back on Port 7. A PWM-controlled motor is connected to pin 0 of port 6 (i.e. pin P6.0).

Pin	Function
P3.0	Data
P3.1	Latch
P3.2	Strobe

To use the sensor, we need to send a 3-bit command over Data, pulsing the Strobe line after the least significant and second bits, and the Latch line after the third (most significant) bit, as shown by the following timing diagram (for command 101 binary).



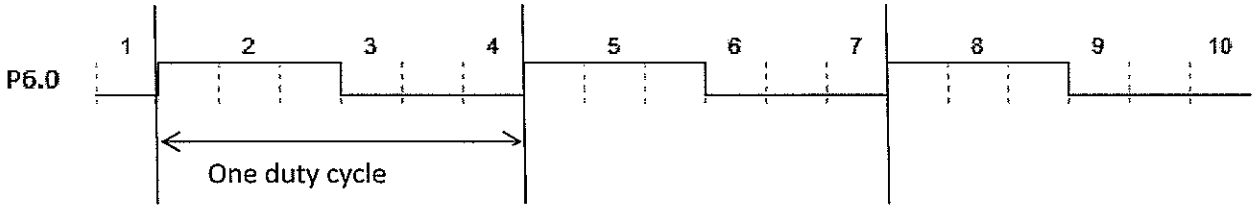
The commands are:

Code (binary)	Command	Comments
000	STANDBY	Switches the sensor to low-power mode.
001	READY	Switches the sensor to READY mode.
010	READ	Reads in the sensor
011	RESET	Resets the sensor in event of an error.
100	ACK	Acknowledges last reading from the sensor.

This sensor has a high power drain when in READY mode and must be switched to STANDBY when not in use. Before reading, the device must be switched back to READY. There must be at least a 5 ms delay between switching to READY and starting a READ.

In READ mode, the sensor sends back an 8-bit reading on P7.7-P7.0 after 20ms after the READ command is sent, and holds the data until the microcontroller acknowledges the reading.

On the output side of things, a motor is connected to pin P6.0, and you need to generate a continuous PWM signal with a duty cycle of 1 ms, as shown below. The data rate is 10,000 bits per second.



a. What is the most appropriate software architecture for this system? Explain your answer. (3 marks)

b. Write the code for "void send_command(int cmd)" which will send the 3 bit command to the digital sensor. (7 marks)

c. Write the code for "int read_digital()", which will get and return one reading from the digital sensor. The sensor is initially in STANDBY mode. (5 marks)

d. What is the duty cycle, in bits, of the PWM signal? (3 marks)

e. The function "generate_pwm" produces a PWM waveform that represents the digital-to-analog conversion of the value stored in a global integer variable called "pwm_out", ranging from 0 to 255. Write the code for "generate_pwm". Note that the waveform should be generated continuously, not just once, as shown on page 10. (10 marks)

(Continues next page)

f. Now write a complete program that reads the digital sensor every 100ms, multiplies the value read in by 2, then writes the result as a PWM signal to the motor. (7 marks)

(Continues next page)

Page is intentionally left blank

END OF PAPER